

IndraMotion MTX 14VRS

编程手册

参考手册
R911389415

版本 05



题目	IndraMotion MTX 14VRS 编程手册
文件类型	参考手册
文件类型代号	DOK-MTX***-NC**PRO*V14-RE05-ZH-P
内部存档附注	RS-32b2b2ff1c3483ba0a6846a50090ad76-5-zh-CN-4
更改记录	2018 年 9 月第 13 版 参考 “本文档版本” 第 11 页
版权	© Bosch Rexroth AG 2018 保留所有权利， 也保留包括 任何使用、利用、翻印、编辑、 转让以及申请知识产权的权利。
责任义务	本特定资料仅供产品描述之用， 除非交易契约中明文条列， 否则不得视为特性保证。关于本文件的内容更改和产品可否供货， 本公司保留所有权利。
编辑部	Development Software and Control Firmware – Control Functions CNC, De (SaKi)

目录

	页数
1 关于本文档.....	11
1.1 文档有效性.....	11
1.2 一般信息.....	11
1.3 必需和补充文件 MTX.....	13
1.3.1 选择/编译.....	13
1.3.2 配置.....	13
1.3.3 调试.....	14
1.3.4 操作.....	15
1.3.5 OEM 工程.....	15
1.3.6 附加模块.....	16
1.4 信息表示方法.....	16
1.4.1 安全提示.....	16
1.4.2 所用符号.....	16
1.4.3 名称和缩略语.....	17
1.5 客户反馈意见.....	17
2 重要使用说明.....	19
2.1 预期使用.....	19
2.1.1 简介.....	19
2.1.2 使用领域和应用领域.....	19
2.2 非预期使用.....	20
3 数控编程基础知识.....	21
3.1 简介.....	21
3.2 标准和 CPL 编程的基本信息.....	21
3.2.1 编程选项.....	21
3.2.2 突出显示零件程序 CPL 元素中的	22
3.2.3 手动数据输入期间的 CPL 元素, 缓冲的数控块指定和未链接的子程序.....	23
3.2.4 遵守 CPL 和标准语言元素之间的解释次数.....	23
3.3 链接数控程序.....	23
3.3.1 链接表.....	23
3.3.2 在程序选择或程序运行时进行链接.....	24
3.3.3 链接子程序 - Postlinking.....	24
3.3.4 使用 DIN / CPL ID 影响链接过程.....	25
3.4 基本数控程序组件.....	25
3.4.1 程序块.....	25
3.4.2 命令.....	25
3.4.3 附加条件.....	26
3.5 程序字.....	26
3.5.1 概览.....	26
3.5.2 数控函数的程序字.....	27
3.5.3 作为参数的程序字.....	29
3.5.4 编程属性.....	30

目录

	页数
3.5.5 两个部分字之间使用分隔符.....	31
3.6 程序结束.....	32
3.7 程序字的作用.....	32
3.7.1 模态.....	32
3.7.2 非模态.....	33
3.8 程序设计的特殊程序设计元素.....	34
3.8.1 通道 ID.....	34
3.8.2 块编号.....	34
3.8.3 程序代码中的空行.....	34
3.8.4 零件程序中的注释.....	34
3.8.5 用户界面上的注解.....	35
3.8.6 在程序运行中跳转.....	36
3.9 子程序.....	37
3.9.1 概览.....	37
3.9.2 带 P 地址的子程序调用.....	38
3.9.3 无 P 地址的子程序调用.....	39
3.9.4 本地子程序.....	40
3.9.5 使用 G 和 M 代码的自定义子程序调用.....	42
3.9.6 自定义模态子程序调用.....	42
3.9.7 通过 CALL 命令在 CPL 中调用子程序.....	43
3.9.8 向子程序传递参数.....	44
3.9.9 配置和错误排除.....	49
3.10 宏编程.....	50
3.10.1 概览.....	50
3.10.2 带参数的宏.....	51
3.11 标签编程和跳转指令.....	52
3.11.1 概览.....	52
3.11.2 标准数控块和 CPL 块中的标签.....	52
3.11.3 GoAhead (GOA) 向前跳转至标准数控块	53
3.11.4 GoBack (GOB) 向后跳转至标准数控块.....	53
3.11.5 GoCond (GOC) 有条件跳转到标准 NC 程序块.....	54
3.11.6 GoTo 无条件跳转至标准数控块.....	55
3.11.7 CPL 跳转 (GOTO) 跳转至任何程序块.....	56
3.11.8 RetTo 子程序返回任何程序块.....	56
3.12 决定和分支指示.....	57
3.12.1 概览.....	57
3.12.2 “跳过块”功能.....	58
3.12.3 CPL 指令: CASE-LABEL... LABEL-OTHERWISE-ENDCASE.....	58
3.12.4 CPL 指令: IF-THEN-ELSE-ENDIF.....	59
3.13 重复语句.....	60
3.13.1 概览.....	60
3.13.2 CPL 指令: FOR-STEP-TO-NEXT.....	60
3.13.3 CPL 指令: REPEAT-UNTIL.....	61
3.13.4 CPL 指令: WHILE-DO-END.....	61
3.14 可变编程.....	62
3.14.1 变量名.....	62

目录

页数

3.14.2	变量组.....	63
3.14.3	变量类型.....	66
3.14.4	标准 数控 编程中的变量.....	69
3.14.5	评价 CPL 中功能错误的变量 ERRNO"ERRNO".....	69
3.15	一般系统信息.....	71
3.16	保留命令字.....	71
3.17	CPL 命令.....	72
3.17.1	赋值.....	72
3.17.2	数学运算.....	73
3.17.3	逻辑运算.....	76
3.17.4	数制间的转换.....	77
3.17.5	关系运算" =、>=、>、<>、<=、<".....	77
3.18	更多 CPL 的基本元素.....	78
3.18.1	常量.....	78
3.18.2	代码字符.....	78
3.19	数控块同步命令.....	78
3.19.1	概览.....	78
3.19.2	块准备的同步功能.....	79
3.19.3	块执行时的同步功能.....	82
3.19.4	执行时写入系统数据.....	92
4	模态功能组概述.....	95
5	数控函数，语法符合 DIN 66025（包括扩展）.....	97
5.1	概览.....	97
5.2	G 代码.....	98
5.2.1	快速移动中的直线插补"G0".....	98
5.2.2	进给中的线性插补"G1".....	100
5.2.3	圆弧 / 螺旋 / 螺旋 N 插补 "G2、G3".....	101
5.2.4	停留时间"G4".....	106
5.2.5	切向圆周出口"G12".....	106
5.2.6	切向圆周入口"G5"， "G13".....	107
5.2.7	样条编程"G6".....	108
5.2.8	路径坡度开启"G8"，路径坡度关闭"G9".....	115
5.2.9	急进限制速度控制"G8（形状 ...）"，"G9（形状 ...）"，"G9（ASHAPE ...）"，"G9（X ...，Y...，...）"，"AsynchrShapeOrder, ASO".....	116
5.2.10	无平面 "G16".....	119
5.2.11	平面切换 "G17、G18、G19".....	120
5.2.12	高级平面切换"G17（...），G18（...），G19（...）".....	121
5.2.13	自由平面选择（与 WCS 独立）"G20".....	122
5.2.14	轮廓移动编程"G31".....	123
5.2.15	螺纹切削 "G33".....	125
5.2.16	铣刀路径修正"G40，G41，G42".....	128
5.2.17	铣刀路径修正的轮廓转换：圆弧"G43"，交叉"G44"圆弧.....	130
5.2.18	进给校正刀具接触点"G45"、刀具中心点"G46".....	131

目录

	页数
5.2.19 刀具长度修正"G47、G48".....	132
5.2.20 可编程零点偏置"G52".....	133
5.2.21 零点偏置 (Z0) "G53"、"G53.1-G59.1" 至 "G53.5-G59.5".....	135
5.2.22 准停定位开启 / 关闭"G61、G62".....	137
5.2.23 无补偿卡盘的攻丝"G63".....	139
5.2.24 英制编程 "G70", 本地英制编程 "I 数控 H(...)".....	140
5.2.25 公制编程 "G71".....	141
5.2.26 接近参考点坐标 "G74"	141
5.2.27 接近参考点"G74 (HOME)".....	141
5.2.28 行进至探针"G75".....	142
5.2.29 接近固定的机床轴位置 "G76".....	143
5.2.30 异步子程序: 重新定位各个坐标 "G77".....	144
5.2.31 补偿切换打开 "G78"、补偿切换关闭"G79".....	144
5.2.32 绝对尺寸编程 "G90"、相对尺寸编程 "G91"、局部绝对尺寸编程 "AC(...)"、 局部相对尺寸编程"IC(...)".....	146
5.2.33 时间编程 "G93".....	146
5.2.34 进给编程 (每分钟) "G94".....	147
5.2.35 具有路径比例速度自适应的绝对速度编程 "G94 (LINS) ".....	148
5.2.36 具有时间比例速度自适应的绝对速度编程 "G94 (LINS) ".....	149
5.2.37 增量速度编程 "G94(...)" 加速适应条件下.....	150
5.2.38 绝对速度编程 "G94(...)" 加速适应条件下.....	151
5.2.39 进给编程 (每转) "G95".....	153
5.2.40 增量速度编程"G95 (...)"带加速度调节.....	153
5.2.41 回转轴每转进给编程 "G95(FRA=<Axis>)".....	155
5.2.42 常量切削速度 "G96"、直接速度编程 "G97".....	155
5.2.43 3D 刀具半径补偿 "G140、G141、G142".....	159
5.2.44 定位: 斜面 "G151 - G159.5".....	161
5.3 M 代码.....	167
5.3.1 中断程序 (程序停止) "M0、M00".....	167
5.3.2 有条件程序中断 (Cond. Progr. Stop) "M1、M01".....	167
5.3.3 退出程序 (程序结束) "M2、M02、M30".....	167
5.3.4 主轴顺时针旋转 "M3、M103、M203", 主轴顺时针旋转和冷却液打开"M13、M113、M213".....	168
5.3.5 主轴逆时针旋转 "M4、M104、M204", 主轴逆时针旋转和冷却液打开"M14、M114、M214".....	169
5.3.6 主轴停止 "M5、M105、M205".....	170
5.3.7 定向主轴 / 定位主轴"M19, M119, M219"	170
5.3.8 自动换档步进选择"M40、M140、M240".....	172
5.3.9 手动换档选择 "M41...44、M141...144、M241...244".....	172
5.3.10 惰档选择 "M48、M148、M248".....	173
5.4 进给和速度编程.....	173
5.4.1 速度设置或模态速度设置"<Axis>=S<Value>".....	173
5.4.2 F 地址 "F".....	174
5.4.3 F2 地址"F2".....	175
5.4.4 异步轴的速度 "FA".....	176
5.4.5 Omega 地址 (进给率) "Omega".....	176
5.4.6 编程主轴速度 "S, SSPG".....	176
5.5 刀具修正.....	178

5.5.1	D 修正 "D".....	178
5.5.2	ED 补偿 "ED".....	178

6 使用高级语言语法的 NC 功能..... 181

6.1	概览.....	181
6.2	编程有效半径因子 "ActRadFact、ARF".....	182
6.2.1	效果.....	182
6.2.2	编程.....	182
6.3	调整模态 NC 功能用一般语言句法 "调整, ADJ".....	183
6.4	可编程的与距离有关的信号.....	185
6.4.1	一般信息.....	185
6.4.2	激活搜索标记 "ADSENA".....	185
6.4.3	取消标记搜索 "ADSENA".....	186
6.4.4	设置标记 "ADSLAB".....	186
6.4.5	复位开关信号 "ADSCLR".....	187
6.5	扩展正向回程程序块 "AdvanceAppend, ADVA".....	188
6.6	替换回程正向程序块 "AdvanceSubstitute, ADVS".....	188
6.7	区域监控 "区域, ARA".....	189
6.8	异步子例程.....	193
6.8.1	异步子例程: 登出 "ASPCLR".....	193
6.8.2	异步子例程: 关闭 "ASPDIS".....	194
6.8.3	异步子例程: 打开 "ASPENA".....	194
6.8.4	异步子程序: 定义返回到轮廓点 "ASPRTP".....	195
6.8.5	异步子程序: 注册 "ASPSET".....	195
6.8.6	异步子程序: 通过程序 "ASPSTA"触发.....	196
6.9	赋予逻辑轴名称 "AssLogName, ALN".....	197
6.10	校准轴运动.....	197
6.10.1	后向转换 "ATBWD".....	197
6.10.2	优化参数 "ATCAL".....	198
6.10.3	前向转换 "ATFWD".....	199
6.10.4	从 NC "ATGET"读取参数.....	200
6.10.5	向 NC "ATPUT"写入参数.....	201
6.11	执行所有组的激活辅助功能 "AUXFUNC".....	203
6.12	更改最大轴加速度 "AxAcc, AAC"。缓存轴加速度 "AxAccSave, AAS".....	203
6.13	加载轴误差校正表 "AxCompTab".....	205
6.14	轴耦合 "AxCouple, AXC".....	207
6.15	轴进给耦合 "AxisFeedCouple, AFC".....	210
6.16	更改最大轴过冲 "AxJerk, AJK", 缓存轴过冲 "AxAccSave, AAS".....	213
6.17	关闭主轴的 C 轴模式 "AxisToSpindle, ATS".....	214
6.18	关闭主轴的 C 轴模式并等待 "AxisToSpindleWait, ATSW".....	215
6.19	替换轴 "AxisReplace, AXR".....	215
6.20	改变最大轴加速度 "AxVel, AVE", 保存最大轴速度 "AxVelSave, AVS".....	216
6.21	B 样条逼近的参数 "BsaPar, BAP".....	217
6.22	定位: 工件位置修正 "BcsCorr, BCR".....	217
6.23	使用缓存数控块输入进行下一块监控 "ChkNxtBlk, CNB".....	219
6.24	倒角编程 "ChLength", "CHL", "CHLL", "ChSection", "CHS", "CHSL".....	219

目录

	页数
6.25 自动探测转角和直线 "CLD".....	221
6.26 碰撞 "Collision、CLN".....	222
6.27 G0 插补配置 "ConfG0、CG0".....	224
6.28 选择轴转换 "Coord、CRD".....	225
6.29 锐化内部拐角 "拐角、COR".....	226
6.30 样条耦合表 "CoupleSplineTab、CST".....	227
6.31 曲线参数插补 "CurveParamInterpolation、CPI".....	228
6.32 客户特定的速度参数文件 "CVP".....	229
6.33 启用 D 修正表 "DcTSel、DCS".....	229
6.34 从 MP 应用轴设置 "DefAxis、DAX".....	230
6.35 将通道主轴复位至 MP "DefSpindle、DSP".....	230
6.36 无减速的块过渡 "DefTangTrans、DTT".....	230
6.37 直径编程 "DiaProg、DIA", 半径编程 "RadProg、RAD".....	231
6.38 驱动器错误的延迟响应 "DERDelay".....	232
6.39 样条类型的定义 "SplineDef、SPD"	232
6.40 数字化距离控制 "DistCtrl、DCR".....	233
6.41 触发和退出诊断记录 "DREC_START", "DREC_STOP".....	235
6.42 结束位置耦合 "EndPosCouple、EPC".....	235
6.43 进给计算: 隐藏轴 "FeedAd、FAD".....	237
6.44 前馈 "FeedForward、FFW".....	238
6.45 前馈异步轴 "FFWAxis、FFWA"	239
6.46 进给计算: 轴组 "FeedGrp、FDG".....	240
6.47 进给计算: Feed 组类型 "FeedGrp、FDGT".....	241
6.48 加工中测量 "FlyMeas、FME".....	242
6.49 固定挡块处的测量 "FsProbe、FSP".....	243
6.50 运动至固定挡块 "FsMove、FSM", "FsTorque、FST", "FsReset、FSR".....	245
6.51 直线度和角度误差补偿 "GeoComp、GCT".....	247
6.52 获取轴 "GetAxis、GAX".....	247
6.53 获取通道主轴 "GetSpindle、GSP".....	248
6.54 通过高速信号实现加工中块过渡 "HsBlkSwitch、HSB".....	249
6.55 通过高速信号实现带挡块的块切换 "HsBlkSwitch(..,HSSTOP=..), HSB(..,HSSTOP=..)".....	251
6.56 工件坐标在线修正 "HWOC", "HWOCDis".....	252
6.57 初始化 "加工中测量" "InitMeas、IME".....	253
6.58 逆时间编程 "InvTime、ITIM".....	253
6.59 选择准停定位窗口 "IPS1、IPS2、IPS3".....	254
6.60 确定系统轴以规定过冲限制启用情况下的最佳 形状 过滤器 "JerkAxlList、JAL".....	254
6.61 激活急进限制 "JerkControl、JKC".....	256
6.62 工件坐标中的点动模式 "JogWCSSelect".....	256
6.63 禁用径向和切向加速度的总和产生的速度限制 "KinLimAcc、KLA".....	257
6.64 停用运动数控块分割 "KinSplitDisable、KSD".....	257
6.65 CO 编程 "KvProg、KVP".....	257
6.66 迟滞补偿 "LCP".....	259
6.67 分隔进给块: 路径段长度 "LEN".....	259
6.68 与路径速度相关的激光功率控制器 "LFP", "LFConf、LFC".....	261
6.69 线性样条曲线切换 "LinearToSpline、LTS".....	263
6.70 模态轴 (线性循环轴) 回零 "LinModZp、LMZ".....	263

目录

页数

6.71	速度变化图 (斜坡功能)	264
6.72	主轴切换 "MainSp、MSP"	266
6.73	输入工具: 镜像 "Mirror(...)、MIR(...)"	267
6.74	在块上测量 "MOB"	269
6.75	在轮廓上测量 (MOC)	270
6.76	模数计算, 用于增量行程无限轴 "Modulo, MOD"	272
6.77	模数偏置, 用于无限轴 "ModuloShift, MODS"	273
6.78	更改测头边缘配置 (MPEDGE)	273
6.79	裁料 "Nibble、NIB"	274
6.80	分隔进给块: 路径段数 "NUM"	276
6.81	定向编程 "O()、ROTAX()", "Phi、Theta、Psi"	276
6.81.1	使用 O(...) 的定向编程概述	276
6.81.2	一般描述	277
6.81.3	使用轴编程的线性方向运动	277
6.81.4	使用坐标编程的线性方向运动	278
6.81.5	矢量定向	280
6.81.6	张量定向	282
6.82	进给 100% "OvrDis、OVD", "OvrEna、OVE"	285
6.83	更改最大路径加速度 "PathAcc、PAC"	286
6.84	启用定位表 "PmTSel、PMS"	287
6.85	极坐标编程	287
6.85.1	定义极点 "PolarPol、POP"	287
6.85.2	极坐标编程	288
6.86	定义镜像点 / 旋转点 "PoleSet、PLS"	291
6.87	与位置相关的高速输出 "PosDepHSOut、PHS"	291
6.88	循环轴的定位类型 "PosMode、PMD", 循环轴的局部定位类型 "DC", "ACP", "ACN"	293
6.89	精度编程 "PrecProg、PRP"	295
6.90	取决于块准备的程序处理 "PREP"	296
6.91	冲程释放时间	297
6.91.1	冲压释放时间 (插补终点) "PtBlkEnd、PTE"	297
6.91.2	将冲程释放时间设置为默认值 "PtDefault、PTD"	299
6.91.3	冲程释放时间 (Inpos 窗口) "PtInpos、PTI"	300
6.92	PTP 进给运动 "PtpMove、PTP"	300
6.93	冲孔 "Punch、PUN"	302
6.94	最大径向加速度 "RadialAcc、RAC"	304
6.95	记录作为系统数据的反向向量 "RecordRevVec, RRV, RRV"	305
6.96	扭矩减小 "RedTorque、RDT"	306
6.97	程序取消后通过数控程序重启继续执行程序 "ReentContBlk, RCB"	306
6.98	从轴组移除轴 "RemAxis、RAX"	307
6.99	移除逻辑轴名称 "RemLogName、RLN"	308
6.100	移除通道主轴 "RemSpindle、RSP"	309
6.101	块重复 "REPEAT, RPT"	309
6.102	异步子例程	310
6.102.1	定义重新定位特性 "REPOSDEF"	310
6.102.2	定义异步子例程的路径返回点 "REPOSTP"	310
6.103	回程、RETR	311

目录

	页数
6.104 回程切换锁 LockRetrace, LOCKRETRACE.....	312
6.105 扩展反向回程程序块"AdvanceAppend, ADVA".....	313
6.106 替换反向回程程序块"AdvanceAppend, ADVA".....	314
6.107 输入工具: 旋转 "Rotate(...), ROT(...)".....	314
6.108 转角的修圆.....	316
6.108.1 指定偏差的转角修圆 "RoundEps、RNE".....	316
6.108.2 半径规范的 "RND"修圆转角, "倒圆", "RNDL".....	316
6.109 同步系统轴耦合 "SACSYNC".....	317
6.110 输入工具: Scale"Scale(...), SCL(...)".....	319
6.111 选定的附加坐标耦合.....	321
6.111.1 选定的附加坐标耦合 "SelCrdCouple、SCC".....	321
6.111.2 选定的使用表进行的附加坐标耦合 "SelCrdCoupleTab、SCCT".....	321
6.112 抑制冲突监控 "SCLN".....	322
6.113 设置程序位置 "SetPos、SPS".....	323
6.114 编程的轮廓移位 "Shift、SHT".....	324
6.115 速度限制 "SMin、SMN", "SMax、SMX".....	324
6.116 调整块中的主轴预运行初始化调整程序 "SpAdjSp".....	325
6.117 确定 NC 重启调整程序的主轴语法 "SpAdjStr".....	325
6.118 启动 / 转换保留的主轴 "SpAdmin、SPA".....	326
6.119 定义(激活)耦合主轴组, 移除(取消)耦合主轴组"SpCoupleConfig、SPCC".....	327
6.120 从动主轴的耦合距离 "SpCoupleDist、SPCD".....	328
6.121 同步模式错误窗口 "SpCoupleErrWin、SPCE".....	329
6.122 带活动耦合的角度偏置 "SpCouplePosOffs、SPCP".....	329
6.123 等待角度偏置 "SpCouplePosOffs_Wait、SPCP_WAIT".....	330
6.124 同步模式窗口 "SpCoupleSyncWin、SPCS".....	330
6.125 等待同步模式 "SpCouple_Wait、SPC_WAIT".....	331
6.126 定义/移除主轴组 "SPG...、SPGALL".....	331
6.127 在系统数据队列中写入 "SSDQ".....	332
6.128 系统主轴编程 "SSp.....".....	334
6.128.1 一般信息.....	334
6.128.2 启用/转换保留的主轴 "SSpAdm".....	334
6.128.3 关闭特定通道的主轴运动管理 "SSpAdmOff".....	335
6.128.4 更改档位范围 "SSpGear".....	335
6.128.5 限制最大速度 "SSpMax".....	336
6.128.6 限制最小速度 "SSpMax".....	336
6.128.7 位置 / 速度接口切换 "SSpMode".....	336
6.128.8 运动编程 "SSpMove (SSPM)".....	337
6.128.9 主轴定向 "SSpOri (SSPO)".....	337
6.128.10 速度编程 "SSpSpeed (SSPS)".....	338
6.128.11 特定于刀具的主轴限制"SSpToolLim".....	338
6.129 开启主轴的 C 轴模式 "SpindleToAxis、STA".....	339
6.130 开启主轴的 C 轴模式并等待 "SpindleToAxisWait、STAW".....	339
6.131 在主轴编程结束时等待主轴编程"SpindleWait, SPWAIT".....	340
6.132 选择一个等待策略 "SpindleWaitMode, SWM".....	341
6.133 使用样条的转角修圆.....	341
6.133.1 使用样条的转角修圆(模态) "SplineCornering、SCO".....	341

6.133.2	使用样条的转角修圆（局部） "SplineCorneringLocal、SCOL"	342
6.134	样条类型的定义 "SplineDef、SPD"	342
6.135	可编程路径分割 "Split、SPT"	344
6.136	主轴：位置和速度接口切换 "SpMode、SPM"	345
6.137	静态刀具定向的参数设置 "StatToolOri、STO"	345
6.138	抑制单步模式 "StepModeDisable、SMD"	347
6.139	特定于刀具的主轴限制 "SToolLim"	349
6.140	TCS 中的进给 "TCM(, ,)"	350
6.141	切向刀具控制 "TangTool、TTL"	350
6.142	切向刀具定向 "TangToolOri、TTO"	353
6.143	从攻丝缩回 "TappRet1、TappRet2"	355
6.144	无补偿夹具的攻丝的主轴选择 "TappSp、TSP"	356
6.145	调整轴配置 "TargetAxisConf、TAX"	357
6.146	在程序坐标中定义 TCS "TcsDef、TCS"	357
6.147	特殊螺纹切削函数 "ThreadSet、TST"	359
6.147.1	一般信息	359
6.147.2	配置缩回数据 ThreadSet(...)	360
6.147.3	配置动力学 ThreadSet(DYN)	360
6.147.4	启用缩回 ThreadSet(ROn)	361
6.147.5	切换主轴操作模式 ThreadSet(SPC)	361
6.147.6	影响通道 IF 信号 ThreadSet(TCI)	361
6.148	程序坐标偏置 "Trans、TRS"， 叠加程序坐标偏置 "ATrans、ATR"	361
6.149	体积补偿	363
6.149.1	启用和禁用 "VCPEna"	363
6.149.2	加载表 "VCPLoad"	363
6.150	虚拟驱动器 "VirtAxisPos、VAP"	364
6.151	速度控制的诊断 "VREC_START"， "VREC_STOP"	364
6.152	获取轴，在必要时等待 "WaitAxis、WAX"	367
6.153	使用扩展 Sercos ID 编号写入 Sercos 参数 "WriteId3、WID3"	367
6.153.1	一般信息	367
6.153.2	将参数写入至一个或多个驱动器（用于主轴和异步轴）	368
6.153.3	将参数写入至一个或多个驱动器（用于同步轴）	368
6.153.4	将 ID 列表中的扩展 ID 编号写入驱动器	369
6.153.5	将参数列表中的值写入驱动器	369
6.154	激活零点偏置表 "ZoTSel、ZOS"	369
7	CPL 命令	371
7.1	坐标轴和坐标	371
7.1.1	概览	371
7.1.2	用于读取名称和属性的命令	373
7.1.3	用于读取坐标和轴位置的命令	377
7.1.4	读取测量值的命令	383
7.2	Spindles	386
7.2.1	概览	386
7.2.2	命令读取名称	386
7.3	更正和输入帮助	387

目录		页数
7.3.1	零点偏置.....	387
7.3.2	位置和轮廓偏置.....	393
7.3.3	工件位置校正和位置（斜面）	394
7.3.4	输入帮助.....	399
7.3.5	刀具修正.....	400
7.4	控制数据和过程.....	403
7.4.1	控制版本.....	403
7.4.2	时间记录.....	403
7.4.3	机床参数.....	405
7.4.4	简单系统数据.....	406
7.4.5	结构化类型的系统数据.....	422
7.4.6	Sercos 接口.....	426
7.4.7	PLC 接口.....	431
7.4.8	激活功能.....	433
7.4.9	数控接口（NCS）	434
7.4.10	常规 XML 表.....	485
7.4.11	刀具数据库.....	486
7.4.12	错误和警告.....	495
7.5	文件和目录.....	500
7.5.1	一般信息.....	500
7.5.2	目录.....	501
7.5.3	文件操作.....	504
7.5.4	编辑文件.....	509
7.5.5	文件属性.....	517
7.6	字符串处理.....	519
7.6.1	基本.....	519
7.6.2	类型转换.....	521
7.6.3	更改字符串.....	523
7.6.4	编程示例.....	530
7.7	与客户端通信.....	532
7.7.1	MMC.....	532
8	附录.....	535
8.1	数控函数表格概述.....	535
8.1.1	按字母顺序排列.....	535
8.2	ASCII 字符集.....	560
8.3	其他关键代码.....	563
8.4	优化加工时间.....	564
8.4.1	直接影响加工速度的措施.....	564
8.4.2	间接影响处理速度的措施.....	566
8.4.3	编程错误导致的处理延迟.....	568
9	服务和支持.....	571
	索引.....	573

1 关于本文档

本文档版本

版本	发布日期	注
01	2014-07	第一版
02	2015-04	实施修改
03	2016-05	快速定位激光切削 - 蛙跳
04	2017-01	可中断 ASUP 的一般扩展语言语法: ASPENA、ASPDIS 基础知识的扩展（跳转命令）: RetTo 从子程序返回到主程序或子程序中的标签
05	2018-03	CVP: 切断内部速度控制 KLA: 切断定向和切向加速的同步合并 MOB: 测量工件位置 Hermite 样条: 使用 Hermite 样条规定外部生成的速度参数文件

表格 1-1: 更改记录

1.1 文档有效性

目标组和产品阶段的概述

以下图形指的是本文档的有界活动，产品阶段和目标群体。
目标组“编程人员”可在产品阶段“调试”使用本文档进行“优化和测试”。

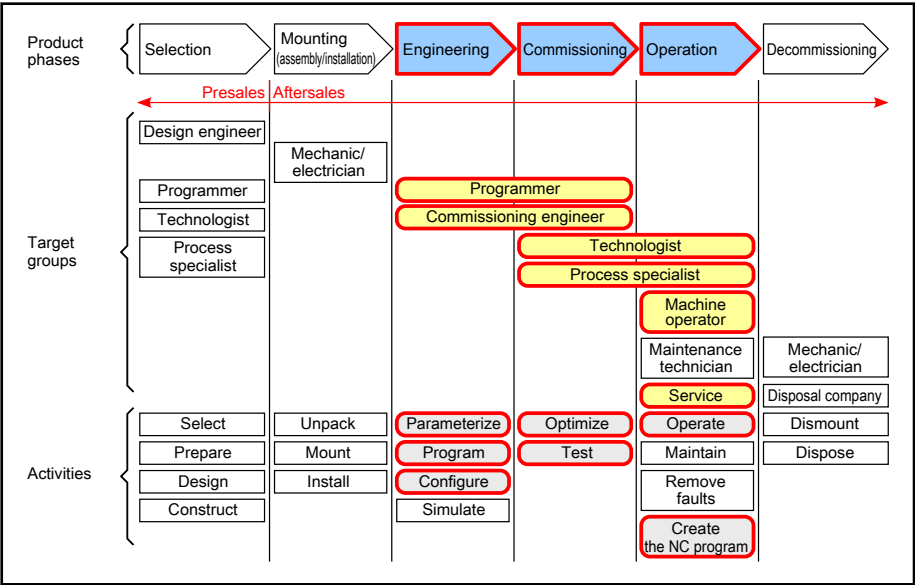


插图 1-1: 将此文档分配给目标组，产品阶段和目标组活动

- 目标群
- 本文档适用于调试 IndraMotion MTX 控制的用户。
- 本文档支持以下操作阶段的用户：
- 调试
 - 操作

1.2 一般信息

本文档描述 Rexroth IndraMotion MTX 控制的标准编程。除了 NC 编程的基本之外，也根据 DIN 66025 描述 NC 功能，并以高层语言语法描述 NC 功能和 CPL 功能。

本手册的目的是在应用 / 激活各种控制功能时提供支持。

关于本文档

手册主要侧重参数设定需要的过程。

这要求 CNC、驱动器、PLC 和电源单元的硬件已根据您的应用文档（包括接线和精度）进行集成和检查。需要特别注意所有安全相应装置，例如 E-STOP 电路、释放触点、限位开关。

理想情况下，所有 Sercos 驱动器的软件已经投入使用。但这并不是必须的，因为控制器可通过参数下载配置启动时连接的任何驱动器。由此控件上的适配初始化文件必须可用。

如需调整初始化文件，须有相应文件（例如所有驱动器所需的 Sercos 参数设置及驱动器文档）。

为了更好地概述，所有功能都已分配不同的上级主题中（请参阅“目录”）。

对于每个功能，提供以下信息：

- 目的或应用领域
- 使用先决条件 / 限制
- 干扰选项。

许多功能同时影响多个域（机器参数，用于编程的 NC 功能，接口信号等）。由此，许多功能也可以在本产品的其他手册中完整或部分描述。



请参阅各自的产品手册。

以下手册补充了本手册：

- 机床参数：“IndraMotion MTX Machine Parameters (IndraMotion MTX 机床参数)”
- NC 和 CPL 功能：“IndraMotion MTX Programming Manual (IndraMotion MTX 编程手册)”
- 接口信号：“IndraMotion MTX PLC Interface (IndraMotion MTX PLC 接口)”

另外，随时查阅驱动器文档。

你需要很好的知识

- 控制器的标准用户界面
- PC 控制面板的 Windows 用户界面
- 用于机器参数配置的工具。

上述工具可变更控制器的配置。因此，您必须对各种需要用到的工具非常熟悉！



小心

由于培训不足或未经培训的人员造成错误或意外使用，可能导致严重的机器损坏，数据丢失甚至人身伤害！

由此，只有经过大量培训并取得资格的人员才能够启动和操作本系统、设置或变更配置参数！

此人还须能够识别更改参数所涉及的风险以及固有的机械、电气或电子设备一般风险。

培训不足或未经培训的员工不正确或意外使用可能会对设备造成严重损坏！Bosch Rexroth 不对由此造成的任何损害和间接损害承担责任。

1.3 必需和补充文件 MTX

1.3.1 选择/编译

文档标题及其类型码和零件号

IndraMotion MTX 14VRS System Description (IndraMotion MTX 14VRS 系统说明) DOK-MTX***-SYS*DES*V14-PRRS-EN-P, R911342640 该文档描述 Rexroth IndraMotion MTX 控制器。它包括控制组件的设计、技术数据、界面和配置。
IndraMotion MTX 14VRS SafeLogic System Overview (IndraMotion MTX 14VRS SafeLogic 系统概述) DOK-MTX***-SL**SYS*V14-PRRS-EN-P, R911341698 本文档描述 IndraMotion MTX 中安全控制器 SafeLogic 的使用。

表格 1-2: MTX 文档概述 - 选择/编译

1.3.2 配置

文档标题及其类型码和零件号

IndraMotion MTX 14VRS Machine Parameters (IndraMotion MTX 14VRS 机床参数) DOK-MTX***-MA*PAR**V14-RERS-EN-P, R911342630 本文档描述可用 Rexroth IndraMotion MTX 参数的设计和修改。还包括 NC 配置器的功能及其操作。
IndraMotion MTX 14VRS PLC Interface (IndraMotion MTX 14VRS PLC 接口) DOK-MTX***-PLC*INT*V14-PRRS-EN-P, R911342622 该文档描述了集成 PLC 的接口信号和程序功能块。
IndraMotion MTX 14VRS Functional Description - Basics (IndraMotion MTX 14VRS 功能描述 - 基本内容) DOK-MTX***-NC*F*BA*V14-RERS-EN-P, R911342626 本文档描述 Rexroth IndraMotion MTX 的基本功能。描述和操作指南中说明了控制器的基本调试步骤和控制功能。
IndraMotion MTX 14VRS Functional Description - Extension (IndraMotion MTX 14VRS 功能描述 - 扩展) DOK-MTX***-NC*F*EX*V14-RERS-EN-P, R911342628 本文档描述 Rexroth IndraMotion MTX 的扩展功能。描述和操作指南中说明了控制器的基本调试步骤和控制功能。
IndraMotion MTX 14VRS Functional Description - Special Functions (IndraMotion MTX 14VRS 功能描述 - 特殊功能) DOK-MTX***-NC*F*SP*V14-RERS-EN-P, R911342624 本文档描述 Rexroth IndraMotion MTX 的特殊功能。描述和操作指南中说明了控制器的基本调试步骤和控制功能。
IndraMotion MTX Free Form Surface Milling (IndraMotion MTX 自由曲面铣削) DOK-MTX***-FREEFORM***-APRS-EN-P, R911341435 本文档描述使用 MTX 控制器进行的自由曲面加工过程。CAD/CAM 系统生成的 CNC 程序是整个加工过程的基础。提供以下主题的概述: 用 CAM 软件生成 CNC 程序时的自由曲面加工 MTX 或 CNC 功能、NC 参数设置、基础条件的描述。
Rexroth IndraMotion MTX Converting MTX Projects (Rexroth IndraMotion MTX 转换 MTX 项目) DOK-MTX***-PROCONV*****-PRRS-EN-P, R911342484 本文档为 IndraMotion MTX 1.x 项目到 MTX 2G 的转换提供支持。描述了转换过程, 指出了转换期间的潜在风险。

表格 1-3: MTX 文档概述 - 配置

关于本文档

1.3.3 调试

文档标题及其类型码和零件号

Rexroth IndraWorks 13VRS Simulation (Rexroth IndraWorks 13VRS 模拟) DOK-IWORKS-SIMU****V13-APRS-EN-P, R911339256 本文档描述模拟组件 View3D、虚拟控制面板、虚拟控制器的功能及其在 IndraWorks 中的操作。
IndraMotion MTX 14VRS Commissioning (IndraMotion MTX 14VRS 调试) DOK-MTX***-STARTUP*V14-CORS-EN-P, R911342620 该文档描述 Rexroth IndraMotion MTX 控制的调试。除了完整的概览，该文档还描述了轴的调试和配置、用户界面以及 PLC 数据。
IndraWorks 14VRS Basic Libraries IndraLogic 2G (IndraWorks 14VRS 基本库 IndraLogic 2G) DOK-IL*2G*-BASLIB**V14-LIRS-EN-P, R911343920 该文档描述了系统综合的 PLC 库。
IndraWorks 14VRS Field Buses (IndraWorks 14VRS 现场总线) DOK-IWORKS-FB*****V14-APRS-EN-P, R911341485 本文档描述 IndraLogic XLC、IndraMotion MLC 和 IndraMotion MTX 系统支持的现场总线和本地外围设备连接。本文档侧重不同外围设备连接的配置、参数设置、调试和诊断。本文档是在线帮助的基础。
IndraWorks 14VRS WinStudio 7.4 DOK-IWORKS-WINSTUD*V14-APRS-EN-P, R911341585 本“用户手册和技术参考手册”提供了与“Rexroth WinStudio”™ 软件有效配合的相关信息。本文档提供创建启用 web 的 HMI / SCADA 程序的技术信息和步骤说明。
IndraWorks 14VRS Software Installation (IndraWorks 14VRS 软件安装) DOK-IWORKS-SOFTINS*V14-CORS-EN-P, R911344286 该文档描述 IndraWorks 安装。
IndraWorks 14VRS Engineering (IndraWorks 14VRS 工程) DOK-IWORKS-ENGINEE*V14-APRS-EN-P, R911343566 该文档描述集成 Rexroth 工程刀具的 IndraWorks 应用。内容包括如何使用 IndraWorks 以及如何操作示波器功能的说明。
Rexroth IndraWorks 12VRS FDT Container (Rexroth IndraWorks 12VRS FDT 容器) DOK-IWORKS-FDT*CON*V12-APRS-EN-P, R911334398 本文档描述 IndraWorks FDT 容器的功能。包括功能在项目中的激活以及与 DTM 的配合。
IndraWorks 14VRS PLC Programming System IndraLogic 2G (IndraWorks 14VRS PLC 编程系统 IndraLogic 2G) DOK-IWORKS-IL2GPRO*V14-APRS-EN-P, R911343571 该文档描述了 PLC 编程工具 IndraLogic 2G 及其应用。包括基本用途、首要步骤、显示、主菜单和编辑器。
IndraWorks 14VRS HMI DOK-IWORKS-HMI*****V14-APRS-EN-P, R911343569 本文档描述 IndraWorks HMI 工程和 IndraWorks HMI 操作的用户界面的功能、配置和操作。

表格 1-4: MTX 文档概述 - 调试

1.3.4 操作

文档标题及其类型码和零件号

Rexroth IndraMotion MTX 09VRS Shop Floor Programming Turning and Milling (Rexroth IndraMotion MTX 09VRS 车间车削和铣削编程) DOK-MTX***-SF*PRO*V09-AWxx-EN-P, R911324377 本文档描述车削和铣削的图形 NC 编程的操作和编程以及工件模拟。
Rexroth IndraMotion MTX 12VRS Block Pre-Run (Rexroth IndraMotion MTX 12VRS 块预运行) DOK-MTX***-BLK*RUN*V12-APRS-EN-P, R911334379 该文档向机床制造商解释如何在机床上为终端用户设置“块预运行”功能。
IndraMotion MTX 14VRS Programming Manual (IndraMotion MTX 14VRS 编程手册) DOK-MTX***-NC**PRO*V14-RERS-EN-P, R911342634 本文档描述 Rexroth IndraMotion MTX 控制的标准编程。除了 NC 编程的基本之外，也根据 DIN 66025 描述 NC 功能，并以高层语言语法描述 NC 功能和 CPL 功能。
IndraMotion MTX 14VRS Standard NC Operation (IndraMotion MTX 14VRS 标准 NC 操作) DOK-MTX***-NC*OP***V14-APRS-EN-P, R911342632 本文档描述 Rexroth IndraMotion MTX 的 NC 控制器的标准用户界面的操作。包括界面、NC 程序开发以及工具管理的操作。
IndraMotion MTX 14VRS Standard NC Cycles (IndraMotion MTX 14VRS 标准 NC 循环) DOK-MTX***-NC*CYC**V14-PRRS-EN-P, R911342638 该文档描述了 Rexroth IndraMotion MTX 控制之不同技术的标准循环应用。
IndraMotion MTX 14VRS NC Simulation (IndraMotion MTX 14VRS NC 模拟) DOK-MTX***-NC*SIM**V14-APRS-EN-P, R911379629 本文档描述 Rexroth IndraMotion MTX 控制器的 NC 模拟。

xx 相应版本

表格 1-5: MTX 文档概述 - 操作

1.3.5 OEM 工程

文档标题及其类型码和零件号

Rexroth IndraMotion MTX 13VRS Automation Interface (Rexroth IndraMotion MTX 13VRS 自动化接口) DOK-MTX***-AUT*INT*V13-APRS-EN-P, R911337274 本文档描述通过自动化接口对 IndraWorks 工程数据进行的基于脚本的访问。描述了不同对象，包括代码实例。本手册还描述了自动化构建器。
Rexroth IndraMotion MTX 14VRS OPC Communication (Rexroth IndraMotion MTX 14VRS OPC 通信) DOK-MTX***-OPC*COM*V14-PRRS-EN-P, R911342636 本文档描述与 Bosch Rexroth 设备的通信的项目的语法和结构。
Rexroth IndraMotion MTX 14VRS OPC UA Communication (Rexroth IndraMotion MTX 14VRS OPC UA 通信) DOK-MTX***-OPC*UA**V14-APRS-EN-P, R911342513 本文档描述 IndraMotion MTX 的 OPC UA 通信。

表格 1-6: MTX 文档概述 - OEM 工程

关于本文档

1.3.6 附加模块

文档标题及其类型码和零件号

Rexroth IndraMotion MTX 11VRS Action Recorder (Rexroth IndraMotion MTX 11VRS 动作记录器) DOK-MTX***-ACR*****V11-APxx-EN-P, R911329943 本文档描述 MTX 动作记录器。包括安装和调试以及接口信号、应用和操作。
Rexroth IndraMotion MTX 14VRS efficiency workbench MTX cta, MTX ega (Rexroth IndraMotion MTX 14VRS 高效工作台 MTX cta , MTX ega) DOK-MTX***-EWB*****V14-APRS-EN-P, R911343916 本文档描述分析工具 IndraMotion MTX cta 和 IndraMotion MTX ega 的操作模式和应用范围。
Rexroth IndraMotion MTX Remote Condition Monitoring (Rexroth IndraMotion MTX 远程条件监控) DOK-MTX***-RCM*****V01-APRS-EN-P, R911334383 本文档描述远程条件监控系统的操作。
Rexroth IndraMotion MTX visIREC User Documentation (Rexroth IndraMotion MTX visIREC 用户文档) DOK-MTX***-VISIREC*V01-APRS-EN-P, R911344242 本文档描述分析工具 visIREC。visIREC 用于优化自由曲面加工过程。路径相关数据的 2D 或 4D 显示。坐标相关数据的 2D 或 4D 显示。分析关键区域（路径和方向偏差）。比较已编程和已插补的 NC 块。

xx

相应版本

表格 1-7:

MTX 文档概述 - 附加模块

1.4 信息表示方法

1.4.1 安全提示

如果当前应用文档中有安全说明，它们包含特定的信号词（危险、警告、小心、注意），有时还包含安全警告符号（根据 ANSI Z535.6-2006）。

信号字提示注意安全说明并指出潜在的风险。

在信号词危险，警告，小心之前的安全警示符号（带三个感叹号的三角形安全反射器）表示对人员有危险。

 小心

若不遵守该安全提示，可造成轻度或中等受伤。

 注意

若不遵守该安全提示，可能造成物质或财产损失。

1.4.2 所用符号

注 注意事项用以下方式表示：



这是对用户的注意事项。

提示 提示用以下方式表示：



这是对用户的一个提示。

1.4.3 名称和缩略语

术语	释义
IWE	Indraworks 工程
IWO	IndraWorks 操作
OWL	光学波导环
NC	数控
OEM	原始设备制造商
Profibus	通讯连接
Sercos	通讯连接

表格 1-8: 所用名称与缩略语

1.5 客户反馈意见

Bosch-Rexroth 高度赞赏客户反馈或改进建议。请将您对文档的反馈发送给 Feedback.Documentation@boschrexroth.de。评论可以插入电子 PDF 文件。然后可以将 PDF 文件发送到上述地址。

2 重要使用说明

2.1 预期使用

2.1.1 简介

Bosch Rexroth 产品根据最先进的技术开发和制造。产品在交付之前已进行测试，以确保操作安全和可靠性。

产品只能按预期使用。如果不按预期使用，会发生导致财产损失或人身伤害的情况。



倘若非预期使用产品而导致损坏，Bosch Rexroth 不承担任何保证，责任或赔偿金；该使用应单独承担非预期使用产品所带来的一切风险。

在使用 Bosch Rexroth 产品之前，必须满足以下要求以保证产品的预期使用：

- 任何人以任何方式处理 Bosch Rexroth 产品者都必须阅读并同意相关安全说明和预期使用。
- 硬件产品不得改动，必须保持原状；即不允许进行结构改变。不允许对软件产品进行反编译或修改源代码。
- 请勿安装或操作已损坏或有故障的产品。
- 必须确保按照相应文件中的说明安装产品。



确保控件中存在的数据或用户输入或读入的数据在应用之前已正确，排除不需要的轴运动。它会是以下无效或旧数据：

- 零件程序
- Z0 表
- 补偿表
- 刀具表
- 永久 CPL 变量
- 残余 PLC 数据
- 永久系统数据

2.1.2 使用领域和应用领域

Rexroth IndraMotion MTX 控制用于

- 工件的程序轮廓和加工技术（路径进给，主轴速度，刀具更改）。
- 沿着编程路径引导机床刀具。

通过 Sercos 接口激活机床的进给驱动，主轴和辅助轴。



此外还需要集成 PLC 的 I/O 组件 - 它与实际的 CNC 一起 - 控制整个加工过程，并在技术安全方面监控该过程。

它只能使用明确指定的硬件组件配置和组合进行操作，并且只能使用正确文档和功能描述指定的软件和固件。

Rexroth IndraMotion MTX 为以下技术的切削和成型提供完美的 CNC 系统解决方案：

- 车削
- 铣削

重要使用说明

- 钻孔
- 磨削
- 弯曲
- 冲压剪切
- 冲孔
- 轮廓切削
- 处理

2.2 非預期使用

Rexroth IndraMotion MTX 在文档和技术数据指定或描述以外的应用领域中的使用被视为“非預期”。

如果...，Rexroth IndraMotion MTX 不得使用

- 它受到不符合指定环境条件的工作条件。禁止在极端温度波动或极端最高温度下在水下操作。
- Bosch Rexroth 并未明确释放预期应用。因此，请阅读一般安全说明提供的信息！
- Rexroth IndraMotion MTX 不能用于通过不安全网络接口连接到因特网的系统或机床。否则，因未经授权的访问可能导致故障或控制失败。

3 数控编程基础知识

3.1 简介

数控控制器通过数控程序（零件程序）接收机床加工工件所需的所有資訊。

由于 数控 程序结构是可变的，因而可采用各种工艺（铣削、车削、磨削等）对几乎任何类型的工件进行加工。零件程序不仅含有描述相对于工件的刀具路径相关运动信息，还含有工艺方面的信息。

运动信息被细分为各种基本轮廓元素（直线型、圆周型、螺旋型、样条型、nurbs 型等）。

如果按照正确顺序并用所有必要的边界条件在数控程序中对所有的加工步骤加以规定，那么，控制器就可在一个加工步骤内执行上述这些简单几何轮廓元素的每种元素运动。此外，必要的边界条件包括工艺功能（转速、速度等）以及各种辅助加工功能（例如，冷却剂和轴夹紧功能）。



关于 数控 程序结构的基本指南，请参阅 DIN 66025。

DIN 66025 “Program Structure for Numerically Controlled Machine Tools”（part 1 and 2）的内容对应于国际标准：ISO/DIS 6983 和 ISO/DP 6983 “机床的数字控制”。

IndraMotion MTX 在控制器中的“文件系统”管理数控程序。也可以与外部驱动相连并从其直接运行程序。

有关的更多信息**文件系统**和**访问权限**及有关创建和编辑零件程序的信息，请参阅 MTX 操作说明。

每个 数控 程序都有一个 数控 程序名，而且必须符合文件名的通用规定。

数控 程序名最多容许使用 28 个字符，包括文件扩展名。所有字母、数字以及特殊字符“.”和“_”都是容许的。

- 数控程序 MTX 的标准文件扩展名为“.npg”。
- 从内部来说，文件名和文件扩展名没有差别。
- 字母区分小写和大写。
- 标识符“.”和“..”不允许用作文件名。
- 文件名在目录内必须是唯一的。但是，具有相同名称的文件可以存在于不同的目录中。
- 在特殊情况下，文件名最多容许 30 个字符（前提是相关程序不得被链接）。

3.2 标准和 CPL 编程的基本信息

3.2.1 编程选项

控制器提供以下两种编程方式：

- 标准或者 DIN 编程
- CPL 编程（CPL：客户编程语言）。

标准或 DIN 编程可用于描述运动顺序及其边界条件（几何学、运动学、动力学、修正等）。标准编程是一种对机床运动进行控制并激活专用机床功能的纯命令语言。

IndraMotion MTX 的语法包括 DIN 66025（G 和 M 代码）中规定的命令，以及有关 G 代码的重要补充和附加的常规类语言语法元素。

标准编程的基本元素也被称为 数控 功能；每个 数控 功能都有相应的编程语法。

可使用分配给 数控 功能的补充参数来设定功能的参数。

示例：

数控 功能：G2 圆周插补，顺时针

参数：I, J, 中心坐标，半径
 K, R

CPL 编程不但与标准 BASIC 通用语言相似，也含有类 Pascal 结构元素。这就是为什么它很容易学习。

CPL 是一种真正的编程语言，因此提供了另一种编程方式。它包括一些机床控制范围之外的程序元素；这些特殊的系统功能允许对控制器的系统数据进行访问。

CPL 编程提供下列选项：

- 带有变量的符号编程
- 字符串处理
- 文件处理
- 数学运算符：
+ , - , * , / ,
以及三角函数，...
- 关系运算符：
= , < , > , ...
- 逻辑运算符：
NOT、AND、OR , ...
- 控制程序运行的控制结构：
REPEAT, WHILE, FOR, IF, CASE, GOTO, ...
- 用来确定内部系统状态的系统功能：
位置、活动功能、刀具数据、接口信号等
- 处理服务：
程序选择、控制器复位、程序启动、操作模式指定

上述选项可以通过可变符号生成和存储各种加工程序。

根据现有的正式输入规定，CPL 指令一般应当采用大写字母书写。

使用 CPL 结果：

- 减少 数控 程序中的重复步骤和相同的程序段，并且
- 由于访问控制系统数据而获得状态程序变量。

标准编程和 CPL 编程之间的最大差异是：读取对应的程序行时，所有的 CPL 部分早已在块准备时间内完成启动。因此，CPL 部分不再存在用于进一步的块准备以及数控块内插时。

3. 2. 2 突出显示零件程序 CPL 元素中的

CPL 编程和标准编程均可在零件程序中使用。由于唯一性的原因，更重要的是为了能提供有效的程序解释，程序的 CPL 部分必须突出显示：

- 完全包含 CPL 元素的程序块通过不带正 / 负号的正值（行号）在块启动时直接突出显示（参见 第 3.8.2 章 “块编号” 第 34 页 ）。

示例：

20 AX1\$="U" 在 CPL 块 20 中，将值 "U" 赋予字符串变量"Ax1\$".

• 如果 CPL 元素被编入**标准数控程序块**（例如，由于符号变量编程），则 CPL 部分的两边将有**方括号**（"["和"]"）。这样就可把可参数化的值赋予数控 功能和 数控 功能的参数。因此，只有能够放在 CPL 内"="号右侧

的表达式（变量、数学表达式、提供对应值的系统功能）才允许放在方括号内。

示例：

N10 G1 Y[PCS("X")]	在数控程序块 N10 中，Y 轴将进给至前一次为 X 轴编制的工件位置。
N20 ROT([360/A+SIN(B)])	在 数控 程序块 N20 中，启动旋转输入工具。旋转角由 CPL 表达式定义。
30 XPOS=100.5	在 CPL 程序块 30 中，先将值 100.5 赋予变量“XPOS”。
N40 G1 X[XPOS] Y10	在数控程序块 N40 中，变量 “XPOS” 的内容被传递为 X 轴的坐标值。

3.2.3 手动数据输入期间的 CPL 元素，缓冲的数控块指定和未链接的子程序

所有 CPL 命令都可以在手动数据输入，缓冲数控块指定或非链接子程序期间使用，不含全局和局部变量。

例外情况是：

- 标签编程和跃变指南(第 3.11 章 “标签编程和跳转指令” 第 52 页)
- 决定和分支指南(第 3.12 章 “决定和分支指示” 第 57 页)
- 重复语句(第 3.13 章 “重复语句” 第 60 页)

3.2.4 遵守 CPL 和标准语言元素之间的解释次数

由于 CPL 语言元素的特殊标识，因此编程人员能立即识别块准备期间已经启动的程序部分。

标准编程元素在块执行之前对机床没有任何外部作用。

示例：

N523 KvProg(X[@KVX])	执行块 N523 时，用于 x 轴的 C0 值 为永久变量 @KVX 在块准备时间时设定的一个值。

但是执行块 N523 之前，变量 @KVX 可能会被设定成一个完全不同的值（例如，由另一通道写入的一个值）！

这种情况下应使用“WAIT”功能以避免不确定性。WAIT 命令会阻止块准备，直到刚刚编程的数控块完全处理完毕。由此，块准备与控制器的活动状态（插补时间）同步。然后，继续对后续编制块的块准备进行处理。此时，不存在其他的准备块；因为这些块已全部处理完毕。

示例：

WAIT	同步块准备
N523 KvProg(X[@KVX])	执行块 N523 时，用于 x 轴的 C0 值 为永久变量 @KVX 在块准备时间时设定的一个值。但是，由于块准备现在已经同步，所以它现在是变量的活动值。

3.3 链接数控程序

3.3.1 链接表

程序被选中后，程序会检查其语法，可能的跳转目标和子程序。此外，为 CPL 变量创建相应的管理结构，运行时控制器将访问这个结构。该过程被称为**链接**（或者“准备”）。

成功链接程序的结果就是生成一个数控程序对应的**链接表**。IndraMotion MTX 所有的链接表都保存在一个特殊的目录下，该目录在机床参数 3080 00004 指定。任何程序链接表的名称就是该零件程序的名称再加上扩展名 “.1”（1: Link）。

在控制器启动时，它会搜索所有现有链接表的相关数控程序。使用参数数控 0 / FileOrg / SrchPathSubProg（MP 3080 00001）中设置的搜索路径进行搜索。没有找到被删除的零件程序链接表。



搜索路径序列的设置应使得包含最多数控程序的路径在参数数控 0 / FileOrg / SrchPathSubProg (MP 3080 00001) 序列的开头尽可能多。

如果已经链接的数控程序被再次选定，则 IndraMotion MTX 会使用现有的链接表，前提是零件程序在此期间未被修改。在程序变更之后，再重新进行链接。

3.3.2 在程序选择或程序运行时进行链接

有两种选择来处理尚未链接的程序：可用于 MTX 12VRS 的程序选择的链接变体以及适用于 MTX 13VRS 和更高版本的程序运行时的链接。

链接程序选择时，执行链接过程。该程序可以在链接后启动。程序运行时的链接在程序选择时启动链接过程，但不会等到链接过程完成（与程序选择时的链接相反）。因此，该程序可以立即开始。

参数数控 0 / FileOrg / ChFileOrg / Ch / KindOfLinking (MP 7070 00030) 定义链接是否可以在程序运行时发生。

如果所选程序在运行时链接，则尚未链接（通过链接处理）的程序部件将被执行。这可能会导致程序执行的短暂延迟。

如果这种延迟在生产过程中造成麻烦，有两种方法可以避免在程序运行时进行链接：

1. 一般链接在程序运行时停用：将相应通道的参数数控 0 / FileOrg / ChFileOrg / Ch / KindOfLinking (MP 7070 00030) 设置为 1。它总是与节目选择挂钩。
2. 程序运行时程序（或子程序）停用链接：程序”（CPL）”在程序/子程序的第一行。包含子程序中调用的子程序的完整子程序是完全链接的。这也可以在程序执行时执行（请参阅 第 3.3.3 章 “链接子程序 - Postlinking” 第 24 页）。

如果程序在程序运行时被链接，那么在程序/子程序的第一行或子程序调用”P <UP> DIN”的 ID 中忽略到目前为止”（DIN）”使用的 ID。

如果程序在选择时被链接，或者由于”（CPL）”ID 而导致子程序/程序完全链接，则”（DIN）”ID 将保持不变。从链接中排除的相应子例程，因此可能不包含请求链接过程的任何 CPL 指令。



对于大型的数控程序，完整的链接会导致程序选择明显延迟。这是要避免的。



”程序运行时的模式，链接，”下创建链接表的优先级较低。在较高的控制负载下，例如，如果有多个通道正在运行，链接表会被延迟创建。如果程序在完成链接表之前中止，链接表生成将中止并因此不保存。



在程序选择时进行链接会导致语法检查。如果在”程序运行时模式的链接”中选择了程序，则在程序选择时不会检查语法。可能的语法错误因此仅在一段时间后才知道。当更改创建的数控程序时，建议通过程序选择将程序完整链接（可以对第一行中的”（CPL）”ID 进行编程）。

3.3.3 链接子程序 - Postlinking

如果在要链接的程序中调用子程序，IndraMotion MTX 会检查是否存在任何现有的链接表。如果有，将不再重新链接上述子程序。这样，将极大地缩短链接步骤。

如果零件程序不含 CPL 部分（跳转、CPL 变量、CPL 表达式等），而只包含数控块（DIN）和子程序调用，则执行前无需对该程序进行明确链接。

在这种情况下，选择程序时可将切换软键“CPL Prog / DIN Prog”设定成“DIN Prog”。如果需要，可在执行程序时重新链接子程序；某些情况下，这样做可能会造成处理进程的延迟。

如果采用 CPL 变量（例如，P[UP\$]）进行子程序调用的编程，则在程序执行前不会执行该子程序的链接程序，因为只有这时才能启用变量名。因此，如果不存在现成的链接表，采用这种方式调用的子程序则将始终要进行重新链接。

3.3.4 使用 DIN / CPL ID 影响链接过程

在上述先决条件下，也可使用零件程序中的命令直接对链接施加影响，而无需考虑切换软键“CPL Prog / DIN Prog”的当前位置。

如需这么做，必须在第一个程序行开始的时候就对关键字“(DIN)”进行编程。同样在这种情况下，如果需要，可在执行该程序时重新链接子程序，

也可以在子程序的第一行对每个 (DIN) 命令进行编程。因此，采用这种方法标识的子程序将不被链接。

另外，可在调用程序中，紧接在子程序调用后写入关键字“DIN”（无圆括号）（也可参见 第 3.9 章“子程序”第 37 页）。

与 (DIN) 命令相似，也可以在第一个程序行开始的时候就对关键字“(CPL)”进行编程。这样，即使已经使用“DIN Prog”设定选定了相应的程序，或者该程序已经作为子程序通过 DIN ID 进行了调用，仍将强制生成该程序的链接表。

3.4 基本数控程序组件

3.4.1 程序块

数控程序至少包含一个**程序块**。

下列适用程序块：

- 每个程序块最多可以编制一个轮廓元素（例如，直线、圆弧等）。
- 程序块前后允许有空行，以提高程序代码的组织/可读取性。
- 程序块所含字符不得超过 512 个。
- 程序块必须以 ASCII 字符<LINEFEED> 结尾。
- 程序块至少包含一个**程序字**（例如，数控函数、功能参数）。

3.4.2 命令

指令就是那些直接或者间接影响刀具路径、程序运行、控制器反应的状态或模式的程序字。所有的数控函数都是典型语句。

可用的数控函数及所需的相应语法规则信息，请参阅章节 第 6 章“使用高级语言语法的 NC 功能”第 181 页。

路径功能在指令中具有特殊的作用：

路径功能描述的是要到达某个位置所采用的方式（例如，直线方式，圆周方式，有 / 无参与轴的插补，进给或快速进给接近运动方式等）。

示例：G0, G1, G2

路径功能通常在相同的块中作为位置、路径或半径规格进行编程。在这种情况下，路径功能始终启用进给运动。

同样，在没有路径功能的程序块中编入的位置或路径说明也始终启用进给运动，这是由于始终有活动的路径功能存在。

示例：带坐标规定的路径功能

程序字	子串 1	部分字 2	示例
数控函数	功能语法	—	AxAcc G0 G52. 1
	功能语法	参数列表	AxAcc (...) G0 (NIPS)
	功能语法	数值	D5 M777

程序字	子串 1	部分字 2	示例
同步命令	命令语法	–	OFFSTOPA
	命令语法	CPL 参数列表	WPV[@9=10]
子程序	UP 语法， UP 名称	–	P UP1 G4711
	UP 语法 UP 名称	CPL 参数列表	P UP2[7,@25]
参数	参数语法	数值	X-23.45 Y=AC(40) S250 F5000
	参数语法	参数列表	O(0,0,1) ROTAX(0,45)

表格 3-1: 部分字的组合



无论是否有部分字 2，都会产生 G 和 M 代码：

- 对于有固定内部功能的 G 和 M 代码，数值为语法的组件，例如 G0, G17, G54, M0, M3, M19, ...。
- 对于自定义功能（子程序调用和辅助功能），则数值不是语法的组件，而是代表功能的值。这些功能将在机床参数中进行配置。

3.5.2 数控函数的程序字

通常，必须符合以下条件：

- 数控函数的语法可包含 G 和 M 代码，以及常用的语言元素。
示例：
"G0", "G41", "G141", "G52.0", "M30", "Mirror(...)"
有关所有可用的数控函数以及相应的语法规则信息，请参阅章节 [第 6 章 "使用高级语言语法的 NC 功能" 第 181 页](#)。
- 数控 功能可包含补充参数，并通过其控制 数控 功能的运行方式。
它分为三种情况：
 - 在数控块中被编程为独立程序字的参数，例如，"G2 X10 I1.3 J2.5 G94 F1000"。这些都是在 DIN 66 025 中定义的主要参数。
 - 被编程为程序字，但在外括圆括号、位于**具有特定语法元素的参数列表**内的参数，例如，"KvProg(X1.2,Y1.2)"。
 - 对于在外括圆括号且位于**无特定语法元素的参数列表**内的参数，只能编入指定位置的值，例如，"Coord(0,1)"、"Rotate(45)"、"GetAxis(Z1,Z,W,REV)"。
- 参数列表是可选的，而且至少含有一个有效参数。如果在一个参数列表中存在数个参数，则需使用逗号使其彼此分隔。



在本手册中，对于无法同时使用的现有参数（可选参数），在两个可选参数之间用"|"符号标出。"|"字符不进行编程。

- 数控 功能的参数是可选的。如果未将此类参数编入，则通常会使用机床参数中永久性编码或储存的默认设置。

示例：

语法规则：TangToolOri({ SYM<s>} ,{ ANG<a>})

根据语法规则，可以看出参数“SYM”和“ANG”是可选参数。编程时，用合适的值替换“SYM”的通配符 <s>，同时用合适的值替换“ANG”的通配符 <a>。通常，语法规则会指定 <s> 和 <a> 的允许值。

例如，上述语法规则的编程选项可以是“TangToolOri(SYM4)”。



在本文档中，确定了以下内容

- 使用大括号标记可选参数，并且
- 使用尖括号标记待编程值的通配符。

尖括号或大括号不用于编程。

无特定语法元素的参数列表：该列表类型中的所有参数都是常量，可将其直接转换为数值或者名称。必须按照精确定义的顺序对其进行编程，因为每个参数的含义都是根据其在参数列表中的位置而单独指定的。

示例：改变参数的顺序可能会产生完全不同的效果：

GetAxis (X1,X,Y3,Y)	系统名称为 X1 和 Y3 的轴被传输到某个通道，在那里它们被赋予通道名“X”和“Y”。
GetAxis (X1,Y3,X,Y)	系统名称为 X1 和 X 的轴被传输到某个通道，在那里它们被赋予通道名“Y3”和“Y”。

无特定语法元素的参数列表中的参数是可选的。

适用于这种情况的内容如下：

- 列表开始和中间处的可选参数不是必需的；但必须将相应的逗号编入程序。只有这样做，列表中的参数位置才会保持唯一性。
- 列表结尾处的可选参数不是必需的；清单会以“) ”结尾。

示例：

GetAxis (<Phy. Ax1> {,<Log. Ax1>} {,<Phy. Ax2> } {,<Log. Ax2>} {,...})

中间没有参数的编程：	GetAxis(X, ,Z)
结尾没有参数的编程：	GetAxis(X)

使用无特定语法元素参数列表中的 CPL 表达式：

如果要将各个元素转换为 CPL 表达式，则每个元素都必须括有方括号。

示例：作为转换参数的变量名

```
10 AX1$="U"           字符串变量 "AX1$" 被赋予 "U"内容。
20 AX2$="V"           将内容"V"赋予字符串变量"AX2$"。
:
N100 G17([AX1$],[AX2$])  G17 使用轴 U 和 V 打开工作平面。
```

带特定语法元素的参数列表：在此列表类型中，所有参数的顺序都无关紧要，因为参数的含义是通过其编程语法来决定的。

示例：

“KvProg (X1.2,Y1.4,Z1.6)” 和

“KvProg (Z1.6,Y1.4,X1.2)”

具有相同的作用，因为可使用轴地址 X、Y 和 Z 对各个 C0 值进行唯一性赋值。

CPL 表达式既可用于语法元素，也可用于数值。但必须采用方括号：“[” 和 “]”。

示例:

示例:

值 20 =1.2	赋值: 1.2 到变量“值”。
N30 KvProg(Y[VALUE],X1.2)	对 X 和 Y 轴的 C0 值进行编程。Y 轴接受包含在变量“值”中的数值, 而 X 轴接受值 1.2 (常量)。

示例:

40 ACHSB\$="X":FACT%=2	将值 "X" 赋予串变量 "AXISB\$", 并将值 2 赋予整数型变量 "FACT%"。
N50 Scale([AXISB\$][FACT%])	对 "X" 轴启用缩放, 缩放系数为 2。 对应编程: Scale(X2)

示例:

60 DIM PARAMETER\$(10)	为最大长度为 10 个字符的字符串创建字符数组。
70 PARAMETER\$="X2"	
N80 AxAcc([Parameter\$])	把 2 m/sm/s ² 的轴加速度赋予 "X" 轴。

3.5.3 作为参数的程序字

带后续值的参数: 一般来说, 下列情况适用于此类程序字:

- 始终以字母开始的地址，且该地址可包含数个字符。
- 具有地址和编号的程序字可用于编程，例如：
 - 轴和坐标名称（例如 X...、Y...、Z...、B...）
 - 半径（R...）和插补参数（I...、J...、K...）
 - 进给率/时间值（F...）
 - 主轴速度和切削速度（S...、S_i =...）
 - 外部刀具补偿（ED...）
 - D 修正（D...）
 - 辅助功能（M...、T...）
- 前导零不一定必须编入程序。
- 非整数采用小数点写入；后缀零可以忽略（例如，“X100.500”相当于“X100.5”）。
- 如果将加号编入程序（或者根本没有），则后续值将始终被视为正值。负号代表负值。

示例： 由地址字母和数值组成的程序字
(这裡：X 轴的坐标值)

X-2407.0458

符号
地址字母
小数点前的值
小数点后的值

数控编程基础知识

- 带后续参数列表的参数：
- 一般来说，下列情况适用于此类程序字：
- 一般来说，与由地址字母和数值组成的程序字相类似的语法，在这里也同样适用。
 - 参数列表须编入在地址之后，并且必须用圆括号括起来。各个元素之间必须用逗号分隔。
 - 参数列表中的元素是常量，可被直接转换为数值或者名称。必须按照精确定义的顺序对其进行编程，因为每个参数的含义都是根据其在参数列表中的位置而单独指定的。
 - 如果要各个元素转换为 CPL 表达式，则每个元素都必须使用方括号括起来。
 - 例如，使用具有地址和参数列表的程序字进行编程
 - 样条系数 (z.B. X(...)..., Y(...)..., Z(...)..., B(...)...)
 - 向量定向 (O(...), ROTAX(...))

3.5.4 编程属性

在对轴和坐标位置进行编程时，也可以为被编程位置值提供编程属性。
在没有指定编程属性的情况下，根据当前的模态机床状态对位置值进行说明。
在有编程属性的情况下，模态状态可根据轴 / 坐标而被本地覆盖。

IndraMotion MTX 能够识别以下属性：

AC(...):	和 G90 / G91 无关，以绝对方式译读已编程位置说明。
IC(...):	和 G90 / G91 无关，以增量方式译读已编程位置说明。
DC(...):	无论“PosMode”功能的编程以及机床参数的设定怎样，均以最短的路径到达为模态轴编制的位置。
ACP(...):	无论“PosMode”功能的编程以及机床参数的设定怎样，均以正旋转方向到达为模态轴编制的位置。
ACN(...):	无论“PosMode”功能的编程以及机床参数的设定怎样，均以负旋转方向到达为模态轴编制的位置。
INCH(...):	无论配置数据中采用何种单位，均以英寸单位来解释为异步轴编制的位置说明。 也可将属性应用于异步进给（FA 地址），从而，无论为异步轴配置何种线性轴单位，均以英寸 / 分钟来解释其编程值。
MM(...):	无论配置数据中采用何种单位，均以毫米来解释为异步轴编制的位置说明。 也可将属性应用于异步进给（FA 地址），从而，无论为异步轴配置何种线性轴单位，均以厘米 / 分钟来解释其编程值。
S(...):	异步循环轴以 rpm 或 MOD / min 进行速率编程

示例：	N10 G90 G1 F1000 X10 Y=IC(15) X 轴完全进给至位置 10，Y 轴进一步增量进给 15 mm。
	N20 G91 X=AC(15) Y5 不管 G91，X 轴端面运动到绝对位置 15，Y 轴以增量方式继续端面移动 5mm。
	N30 B=DC(90) B 轴（循环轴）沿最短路径进给至 90 度位置。
	N40 B=ACP(350) B 轴以正旋转方向移动至 350 度位置（进给路径：260 度）。

N50 B=ACN(0) B 轴以负旋转方向定位在 0 度（进给路径：350 度）。
N100 B=S(1000) 以 1000 rpm 转动异步 B 旋转轴

3.5.5 两个部分字之间使用分隔符

描述: 每个部分字都可包含一个或者多个字符（字符串）。每个可编程字符都可根据下列组进行分类:

- 字母:
 "A" - "Z", "a" - "z"
- 数字（包括小数点!）:
 "0" - "9", "."
- 作为分隔符的字符:
 " ", "=", "+", "-", "(", ")"
- 其他特殊字符（不进一步讨论）

以下情况下必须在两个相邻部分字之间编入分隔符:

- 前面的部分字以字母**或者**数字结尾，并且
- 后面的部分字以字母**或者**数字开头。

例如，下面这种情况:

- 如果没有数值和参数列表的数控函数后面是补充语法
 - 或 -
- 如果以数字和数值结尾的参数语法将被赋值给参数。

示例:

1.N10 OVE FeedForward(...)

合适的分隔符: " "

2.N20 X2=2

合适的分隔符: "="

在第二种情况中，" " 或 "+" 也可以作为分隔符使用；在赋负值的情况下，"-" 也可以使用。

分隔符的使用还与含有特定语法元素的参数列表相关！

轴和坐标指定的特殊功能: 可以在机床参数中指定控制系统的所需轴和坐标标识符:

- 参数 MAIN/Dr/SysDrName (MP 1003 00001) 系统轴名称
- 参数 CHAN/Ch/Coord/Acs/ChAx/ChAxName (MP 7010 00010) 通道轴名称
- 参数 CHAN/Ch/Coord/Acs/OptAxName/OptChAxName (MP 7010 00020) 通道的可选轴名称
- 参数 CHAN/Ch/Coord/Wcs/LinCoord und CHAN/Ch/Coord/Wcs/OriCoord (MP 7080 00010) 通道的坐标名称
- 参数 CHAN / Ch / Coord / Mcs / CartCoord (MP 7080 00020) 通道的笛卡尔坐标名称

轴和坐标名称始终以字母开头，而且可以

- 包含一个或者多个字母，但所得字符串不得与 数控 功能相同。
 示例: "X"、"PALLET"
- 包含一个或者多个字母，并以数字结尾。
 示例: "X1"、"PALLET1"
- 在这种情况下，必须在名称和以下值之间编程一个合适的分隔符或空格: "=", "+", "-".

示例：轴标识符“X”和“X2”被定义。

N40 G1 X10	轴 X 行进到位置 10
N50 G1 X20	轴 X 行进到位置 20
N60 G1 X2.5	轴 X 行进到位置 2.5
N70 G1 X=2.2	轴 X 行进到位置 2.2
N80 G1 X 2.8	轴 X 行进到位置 2.8
N90 G1 X2	轴 X 行进到位置 2
N100 G1 X2 1	轴 X2 行进到位置 1
N110 G1 X2=2.8	轴 X2 行进到位置 2.8
N120 G1 X2+3	轴 X2 行进到位置 3
N130 G1 X2-2.4	轴 X2 行进到位置-2.4

3.6 程序结束

在下列情况下，代表已到达程序和/或子程序的末端：

- 在文件末端
 - 或 -
- 在含有“M2”、“M02”或“M30”的程序块中。
有关所谓的 M 功能的详细信息，请参阅 第 5 章 “数控函数，语法符合 DIN 66025（包括扩展）” 第 97 页 部分。

示例：

:	
N250 ...	N250 是最后一个程序块
M30	程序结束。

在子程序的末端，系统返回到调用程序。所有模态状态都保留（“模态”：参见页面 第 3.7.1 章 “模态” 第 32 页 ）。
在主程序的末端，系统返回到该程序的起点，并等待下一次 “数控 启动”。如果 “M2”、“M02” 或 “M30” 已被用作主程序的末端，则模态状态被设定为初始串的 M30 部分。

3.7 程序字的作用

3.7.1 模态

- 描述：** 程序字可具有“模态”或者“非模态”作用。
“模态”是指在进行以下操作之前，程序字在每个后续的程序块中都保持有效：
- 使用不同的值编制相同的程序字；
 - 编程的另一个程序字，抵消了其作用
 - 或 -
 - 特地关闭该程序字的功能。



术语“自持式”有时用作“模态”的同义词。

数控零件程序示例:

N10 F1000	将进给速度设定为 1000mm / min。 F1000 以模态方式作用。
N20 G0 X0 Y0	在快速进给到位置 X0/Y0 过程中进行直线插补。 G0 以模态方式作用。
N30 Z100	在快速进给到位置 Z100 过程中进行直线插补。
N40 G1(IPS1) X10 Y10	1000 mm / min 直线插补（在进给过程中；采用细缩放定位窗）到 X10/Y10 位置。 G1 取消 G0 的作用。 G1(IPS1) 以模态方式作用。
N50 X20	1000 mm / min 线性插补（在进给过程中；采用细缩放定位窗）到 X20/Y10 位置。
N60 G1(IPS2) X30 Y30	线性插补（在进给中；带有粗略定位窗口），以 1000 mm / min 的速度到 X30 / Y30 的位置。 用不同的值（IPS2）对 G1 进行编程。G1(IPS2) 以模态方式作用。
N70 X40 Y40 F500	线性插补（在进给中；带有粗略定位窗口）以 500 mm / min 的速度到达位置 X40 / Y40。 用不同的值（500）对 F 进行编程。 F500 以模态方式作用。
N80 G0 X0 Y0	在快速进给到位置 X0/Y0 过程中进行直线插补。 G0 取消 G1(...) 的作用。 G0 以模态方式作用。
N90 Scale(X2,Y2) :	激活缩放功能。以模态方式进行缩放。 缩放保持有效。
N200 Scale()	取消缩放功能的激活。

3.7.2 非模态

“非模态”是指仅在被编入的程序块中才起作用的程序字。

 术语“非自持式”或“局部”被用作“非模态”的同义词。

数控零件程序示例:

N10 G1 F1000	激活直线插补。 G1 以模态方式作用。 将进给速度设定为 1000mm / min。F1000 以模态方式作用。
N20 G75.1 X100 Y100	针对第一测量探头行进。(G75.1) 以非模态方式作用。 在这个块中，G75.1 功能超驰模态功能 G1。
N30 Z100	在进给到 Z100 时进行线性插补。 G1 保持模态有效。

3.8 程序设计的特殊程序设计元素

3.8.1 通道 ID

描述: 使用此选项为通道指定了程序可能独占使用的相应程序。若在另一个通道中启动该程序，会产生运行时错误。
频道 ID 写在节目的开头。

语法:

\$<ChannelNumber>

表格 3-2: 频道 ID 语法

示例:	N10 \$2	该程序仅可在通道 2 中运行。
	:	程序指令
	M30	程序结束

3.8.2 块编号

- 程序块可采用编号进行标识，以提高程序代码的可读性。
如果指定了唯一的块编号，则可以将跳转编入这些块编号中。
- 仅含有 CPL 元素的程序块**必须**用块编号进行标识。这类程序块也被称为 **CPL 块**。

以下情况适用:

- 必须始终将块编号编为程序行中的第一个 数控 字。
- 标准数控块**的块编号包含地址字母 “N” 和一个紧跟的正数。
(示例: N10、N10.2)。
请注意: 标准 数控 块中的 CPL 表达式必须用方括号括起来。
- 纯 CPL 块**的块编号只包含没有正 / 负符号的正数。
(示例: 10, 11.9).
在块编号后面编入块的 CPL 命令或者说明。
- 如果某个 CPL 块以 “:” 它必须跟着没有行号的另一个 CPL 块。

3.8.3 程序代码中的空行

空行可用于构造程序，从而提高程序的可读性。空白行被控件跳过。

3.8.4 零件程序中的注释

程序运行时，控制器会跳过注释。

使用注释

- 记载程序代码或者插入说明;
- 为整个计划项目或个别部分提供意见。

程序注释有助于并加速后续的编辑，例如 如果程序需要更改。但是，每个注释字符会将程序文件增加 1 个字节。

为整个程序行提供注释: 。 无论是标准数控块还是 CPL 块，可以完全隐藏任何程序行。

为整行提供注释:

- 在块的开头编入分号 “;”
- 或 -
- 用圆括号 “(” 和 “)” 将整行括起来。

标准数控块中的注释:

可在标准数控块中的任何位置编入备注:

- 编程分号"; "注释开始的地方。从分号到行尾，控制器将程序行解释为注释。
- 用圆括号 "(" 和 ")" 将标准数控块中的整个备注括起来。由此，也可能对"嵌套注释"进行编程，例如在已包含带括号的注释的程序行中隐藏序列。

圆括号不得用在那些其带圆括号的参数列表（例如，G0、G1、G61）能被可选地编辑的功能之后，而是必须用在尚未编制参数列表的功能之后。

示例:

N10 G0 (<comment text>)	无效编程!
N20 G0 (NIPS) (<comment text>)	有效编程

- 在带有可选参数列表的功能后面，如果该列表丢失的话，可用 "/" 来启用备注。这样的注释可以选择以 "\\" 结尾。

CPL 块中的注释:

在 CPL 块中，引入"REM"的注释。从 REM 命令开始到该行结束，控制器将把该程序行视为备注。

语法:

REM <CommentText>

表格 3-3: REM 语法

示例:

```
...
10 REM *** UP 揭露状态字***
...
```

示例:

零件程序中的注释

:	程序代码
; <Comment>	注释行
;N10 <Standard NC block>	隐藏标准数控块
;20 <CPL>	隐藏 CPL 块
(<Comment>)	注释行
(N30 <Standard NC block>)	隐藏标准数控块
(40 <CPL>)	隐藏 CPL 块
N50 G1 X0 Y0 ; <comment>	在标准数控块中注释
N60 G1(IPS) (<comment>) F1000	在标准的数控程序块中完成注释。
N70 X10 (Y10 (<comment>))	在标准数控块中"嵌套注释"。
N80 G0 //<comment>	对可选参数进行编程的功能进行注释。
N90 G0 // <comment> \\ X0 Y0	完成对可选参数编程功能的注释。
100 REM <注释>	在 CPL 块中注释。

3.8.5 用户界面上的注解

使用注解编程在 数控 用户界面上显示注解文本（最多 80 个字符）。因此，例如，可以在程序运行时为机器操作员提供以下内容

- 关于当前程序状态的信息，或者

- 操作指南。
有两种类型的注释:
- **特定通道注释:**
这些注释在程序取消选择时，或者通道复位时，将被删除。
语法: MSG (<note text>)
- **跨通道注解:**
这些注释在全部复位时，将被删除。
语法:
GMSG (<Note text>)

也可以通过编制注释，为机床操作人员提供操作指南。为此，必须在同一行或者下一行编制（例如）“M0”。因此，确保在输出笔记文本之后立即中断程序。该程序的其他部分将在按下“数控 start”之后才能运行。

示例:

:	程序代码
N60 (MSG 探针工件!)	输出通道特定的注释。
N70 M0	等待 数控 启动按钮。
:	程序代码

由于兼容性的原因，也存在一些用于注释编程的可选的语法变量；从功能的角度来讲，它们全部是同等的。

- 特定通道注释的语法变量:
(MSG<Information text>)
(*MSG<Information text>)
(MSG,<Information text>)
(*MSG,<Information text>)
- 跨通道注释的语法变量:
(GMSG<Information text>)
(GMSG,<Information text>)

如果未使用 MSG 或 GMSG 编程注释文本（或空格），则挂起的注释将被删除。MSG 功能删除由 MSG 或 PRN (0, ...) 创建的通道相关音符。GMSG 可以删除与频道无关的音符。

示例:

:	程序代码
N60 MSG (探针工件!)	输出通道特定的注释。
N70 M0	等待 数控 启动按钮。
N80 MSG ()	删除笔记。
:	程序代码

3.8.6 在程序运行中跳转

- 随着程序的不断增大，越来越需要“清楚的编程过程”。
- 这主要包括以下几项:
- 结构化编程，
 - 故障容差，和
 - 软件工效学。
- 一般来说，结构化程序的总体结构更加清晰。将服务于相关用途或者包含常用功能的程序段构建为（参数化）子程序或者单独的跳转目标，并配备一个可识

别的标识符（标签），这样不仅可以提高可读性，还可以提高加工效率，因为这些程序还可以在其他程序中使用。

提供以下选项：

- **子程序调用**
(参见 第 3.9 章 “子程序” 第 37 页)
如果在加工过程中，某个加工段以同样或者类似的形式发生数次，则可使用子程序。
仅对该段进行一次编程（如果需要，也可以转换参数），然后将其保存为程序，需要时只需对其进行调用。
这节省了编程代码和内存空间。此外，这些程序变得更加清晰易于维护。
- **跳转指令**
(参见 第 3.11 章 “标签编程和跳转指令” 第 52 页)
可根据指定事件（例如，计算结果）使用跳转指令，以在当前程序内的其他位置继续运行该程序。
- **决策 / 分支指令**
(参见 第 3.12 章 “决定和分支指示” 第 57 页)
如果要根据一定的条件来执行各个程序块、不同的程序段或者整个子程序时，可使用此类命令。
- **重复指令**
(参见 第 3.13 章 “重复语句” 第 60 页)
如果程序段或者所有的子程序需要重复运行，可使用重复指令。通常，也可以输入重复率。

3.9 子程序

3.9.1 概览

子程序 (SR) 是通过子程序调用所调用的程序。一旦子程序的执行结束，调用程序将从 SR 被调用的那点继续运行。程序的自我调用被称为“递归”子程序调用。

主程序 (MP) 是从它这里执行跳转到第一个子程序级别 (SR 级别) 的程序。形式上来说，子程序和主程序之间没有什么区别。但只有子程序才具有转换参数。

以下情况适用：

- 子程序可包含标准 数控 块和 CPL 块。
- 每个零件程序都能作为子程序被其他程序调用。但是，任何程序都不能作为子程序进行自我调用（递归调用是不可能的）。
- 调用程序能把参数转移到某个子程序。
- 最大的嵌套深度为 24，即，控制器能同时使最多 24 个子程序级别保持打开状态。
- 子例程名称也区分大小写。

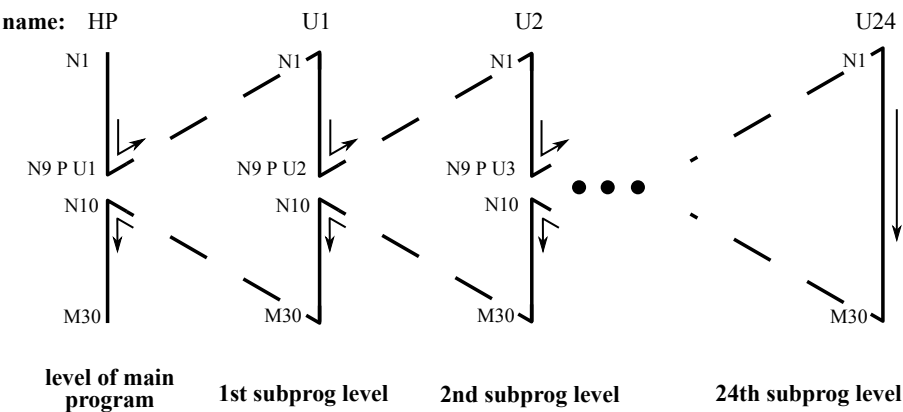


有关终止子程序的信息，请参见 第 3.6 章 “程序结束” 第 32 页。

最大嵌套深度不可直接配置或进行延伸。但是，不同的机床参数能够阻碍可达到的最大嵌套深度。有关更多详细信息，请参见 第 3.9.9 章 “配置和错误排除” 第 49 页。

数控编程基础知识

示例： 允许的子程序嵌套采用以下图表表示。N1 是程序的第一个块，在块 N9 中调用下一个子程序。M30 完成相应子程序。



- 有本地和模态有效的子程序：
- 一般而言，子程序**本地**有效。在这种情况下，子程序在调用点被调用一次。
 - 如果另一方面激活了一个**模态**有效的子程序，则在每次编程的运行运动中再次调用它，直到再次取消选择。例如，该选项可用于钻孔循环。

IndraMotion MTX 提供下列子程序调用的形式：

- 用 P 地址和子程序名称（可选用路径信息）从标准 数控 块中调用。
- 只用没有路径信息和 P 地址的子程序名称从标准 数控 块中调用。
- 子程序作为标准数控块中的自定义 G 或者 M 功能。
- 用自定义语法从标准 数控 块中激活模态子程序。
- 使用 CALL（调用）功能从 CPL 块中调用。

3.9.2 带 P 地址的子程序调用

- 描述：
- 直接在 P 地址后面编入子程序名称。也可以编程子程序所在的目录。
 - 利用 CPL 变量，可在 P 地址后面编入有路径或无路径的子程序名称。因此，CPL 变量必须用方括号括起来。
 - 为提高可读性，可在 P 地址和子程序名称之间编入空格符作为分隔符。
 - 子程序调用必须在程序段结束时编程。
- 在子程序调用之前执行在同一程序段中编程的遍历运动（请参见示例）。
- 一个程序段中只能编写一个以上的子程序。
 - 子程序调用是本地类型（非模态）。

 通过 CPL 变量（如“P [UP \$]”）使用子程序调用会导致扩展编程运行时。

语法：		P{<path>}<Name> {DIN} or P[<CPL variable>] {DIN}	
		其中	
		<Path>	是子程序所在的目录。
		<Name>	是将被调用的程序名称。

<CPL var.>	CPL 字符串变量，包括有路径或无路径的子程序名称作为值。
DIN	可选 防止子程序被链接。 “简单”CPL 指令（请参见 第 3.2.3 章 “手动数据输入期间的 CPL 元素，缓冲的数控块指定和未链接的子程序” 第 23 页）在非链接子例程中是允许的。 子程序调用是允许的。这些也是后链接。 有关更多信息，请参阅 CALL 命令（请参见 第 3.9.7 章 “通过 CALL 命令在 CPL 中调用子程序” 第 43 页）。

表格 3-4: 带有 P 地址的语法

示例:

```
:
N40 P 孔图案           调用程序“钻削模式”。
N50 X100               SR 末端后接块 N50。

:
N140 G0 X10 Y0 PUP1    首先，在快速进给中定位到 X10 / Y0。
N150 Z0                然后调用程序 “UP1”。
:                       SR 末端后接块 N150。

200 DIM UP$ (20)
210 UP $ = “洞型”
N220 P[UP$]
```

3.9.3 无 P 地址的子程序调用

描述: 也可在**没有前导 P 地址**的情况下调用子程序。

- 仅子程序名称直接编定。
- 子程序名也可以用 CPL 变量编程。因此，CPL 变量必须用方括号括起来。
- 无法提供任何路径信息。
- 子程序调用必须在程序段结束时编程。
在子程序调用之前执行在同一程序段中编程的遍历运动（请参见示例）。
- 一个程序段中只能编写一个以上的子程序。
- 子程序调用是本地类型（非模态）。

语法:

<Name> 或 [<CPL var.>]	
其中	
<Name>	是将被调用的程序名称。
<CPL var.>	CPL 字符串变量，包括无路径的子程序名称。

表格 3-5: 无 P 地址的语法



确保这个编程选项不会对正常语法造成任何混淆！
如果要执行的子程序以有效语法开始，则必须使用 P 地址来调用。

因此，始终为子例程分配唯一的名称以避免控制解释器的错误解释。

示例:	
	:
N40 XUP	调用程序 "XSR"。
N50 X100	SR 末端后接块 N50。
	:
100 DIM UP\$ (20)	
110 UP\$= "XUP"	
N120 [UP\$]	
N100 X1UP	注意!
:	这里编制的名称为 "X1UP" 的 UP 会造成语法错误, 因为 "X1" 被解释成名称为 "X" 的某个轴的坐标, 而且名称为 "UP" 的程序并不存在。
:	
:	
:	
:	
N140 G0 X10 Y0 XUP	首先, 在快速进给中定位到 X10 / Y0。
N150 Z0	然后调用程序 "XSR"。
:	SR 末端后接块 N150。

 在用 CPL 变量调用的子例程中, 不可能使用 MTX 块预运行对话框。

3.9.4 本地子程序

- 描述:** 在 MTX 11V02 或更高版本中, 也可以在主程序的文件中定义子程序。本地子例程以命令 **LPS** <SRName> (LocalProgStart) 开始, 并使用命令 **PEND** (ProgramEnd) 完成。在一个程序序列中, 可以使用命令 **LP** <SRName> (LocalProg) 调用本地子例程。
- 要调用本地子例程, 本地子例程的名称直接在 LP 地址后面编程。
 - 子程序名也可以用 CPL 变量编程。因此, CPL 变量必须用方括号括起来。
 - 子程序调用必须在程序段结束时编程。
在子程序调用之前执行在同一程序段中编程的遍历运动 (请参见示例)。
 - 一个程序段中只能编写一个以上的子程序。
 - 本地子程序调用是本地 (不是模态) 有效。
 - 位置或地址参数可以传送到本地子程序。参数按 "常规" 子程序传输。
 - 本地子程序调用 LP 可能包含参数 **DIN**。参数 DIN 表示本地子程序专门包含 DIN 或 "简单的 CPL 指令", 因此不必链接 (请参见 [第 3.2.3 章 "手动数据输入期间的 CPL 元素, 缓冲的数控块指定和未链接的子程序" 第 23 页](#))。
 - 在调用本地子程序时, 不需要编程 DIN / CPL ID, 也可以将此 ID 放在圆括号中并在本地子程序的第一个程序行中编程。
 - 本地子例程不能称为异步子例程 (ASUP)。
 - 本地子程序可以嵌套, 也可以包含普通的子程序调用。
 - 主程序完成后, 本地子程序必须位于文件末尾。
 - 程序结束由 M30 标识。如果没有编程 M30, 则第一个 LPS 编程被隐式解释为程序结束。

- 加工循环的轮廓定义必须位于文件中第一个 LPS 命令的前面。

应用程序：使用本地子程序意味着通过几乎独占的 DIN 编程（例如自由曲面）减少大型零件程序的链接时间。DIN 零件被“发送”到本地子程序中，然后使用 DIN 参数调用该子程序。

语法：	<Name> 或 [<CPL var.>] {DIN}	
	其中	
	<Name>	是将被调用的程序名称。
	<CPL var.>	CPL 字符串变量，包括无路径的子程序名称。
	DIN	可选 防止子程序被链接。 仅当子程序不包含任何 CPL 块时才使用此参数。子程序调用是允许的。如果需要，这些链接被重新链接。

表格 3-6: 本地子例程

示例：	:	
	10 I% = 5	
	N10 G0 X0 F1000	
	20 IF (...	
	:	
	LP Up 1(2,3.4)	使用位置参数 (P1, P2) 调用本地子程序 Up1
	:	
	X10 LP Up2	快速移动 X10 上的运动，然后调用本地子程序 Up2
	:	
	LP Up3(MOD2,X3.4)	使用地址参数 (MOD, X) 调用本地子程序 Up3
	M30	程序结束
	LPS Up1	本地子程序 Up1 的开始
	:	
	PEND	本地子程序 Up1 结束
	LPS Up2	本地子程序 Up2 的开始
	:	
	PEND	本地子程序 Up2 结束
	LPS Up3	本地子程序 Up3 的开始
	(ADR_PARA)	传输地址参数的标签
	:	
	PEND	本地子程序 Up3 结束



在用 CPL 变量调用的子例程中，不可能使用 MTX 块预运行对话框。

3.9.5 使用 G 和 M 代码的自定义子程序调用

除了上述的子程序调用之外，控制器还允许用户：

- 使用 M 地址的“非模态”子程序调用
参见参数数控 P / SubProg / NonModalMFunc / MFunc / MFuncCode (MP 3090 00003) 和参数数控 P / SubProg / NonModalMFunc / MFunc / MFuncSubProg (MP 3090 00004)) 和
- 使用 G 地址的“非模态”子程序调用
(参见参数数控 P / SubProg / NonModGFunc / GFunc / GFuncCode (MP 3090 00001) 和参数数控 P / SubProg / NonModGFunc / GFunc / GFuncSubProg (MP 3090 00002))

各个子程序都可以被赋予最多 64 个自定义的 G 代码和最多 8 个自定义的 M 代码；在编制相应的 G 或者 M 代码时，这些子程序将作为局部（非模态）子程序进行调用。
被配置为子程序调用的 G 和 M 代码不得与永久性定义的 G 和 M 代码相抵触。



有关在特定机床上配置子程序调用的详细信息，请联系相关系统管理员。

以下情况适用于自定义 SR 调用的编程：

- 在零件程序中只能编制相应的 G 或者 M 代码。最终被调用的子程序由机床参数的配置来决定。
- 子程序调用必须在程序段结束时编程。
在子程序调用之前执行在同一程序段中编程的移动运动。
- 一个程序段中只能编写一个以上的子程序。
- 子程序调用是本地类型（非模态）。

3.9.6 自定义模态子程序调用

开始调用后，在标准数控块指定的每次进给运动之后，模态子程序将会继续自动执行。并将一直持续到采用特殊的数控函数将这些子程序取消选定为止。
模态子程序调用（包括关闭功能的语法）在机床参数中进行组态（参见参数数控 P / SubProg / ModaSubProg / SwiOn / SwiOnSynSubProg (MP 3090 00005) 等）。
最多可以设置 63 个模态子程序。模态子程序的名称被分配给可自由定义的语法。此外，必须为可传输至相应子程序的参数指定最大的参数数目。
模态子程序调用的语法不得与永久性定义的数控函数相抵触。



有关在特定机床上配置子程序调用的详细信息，请联系相关系统管理员。

以下情况适用于模态 SR 调用的编程：

- 在零件程序中只编程模态子程序的相应组态语法。最终被调用的子程序由机床参数的配置来决定。
- 在一个块中只能编制 1 个子程序调用。
- 子程序调用是模态类型，即，该子程序在每次已编制的进给运动之后都被再次调用，直到它被取消选择。
- 使用取消选择语法取消激活模态子程序。
- 如果一个模态子程序被编程，而另一个模态子程序已经激活，它将在编程位置执行一次，如非模态子程序。

- 如果相同的模态子程序在没有编程取消选择语法之前被重新编程，则只应用新的参数。该程序以下一个运行运动开始，但带有新参数。这对应于在新模态子程序参数化发生之前仿效取消选择语法的行为。

以下内容适用于模态子程序的参数:

- 程序化参数从 SR 呼叫继承到 SR 呼叫（参数编程一次适用于每个模态子程序呼叫，直到它们被取消选择）。
- 如果这种模态子程序的编程参数被改变一次，则改变的值被继承。
- 最多可以继承 <n> 编程的参数。<n> 是在参数数控 P / SubProg / ModaSubProg / SwiOn / NofParSubProg (MP 3090 00007) 中定义的相应模态子程序的传送参数的数量。如果多于 <n> 个参数被编程，它们只适用于第一个子程序调用。该值在以下模态调用中不再分配。
- 如果一个模态子程序处于激活状态并再次编程（无需预先取消选择），则新编程的参数将另外应用于旧的继承参数。这适用于达到可以继承的参数最大数量（见上文）。

注意: 在调用中未编程的参数值永远不会被继承！

3.9.7 通过 CALL 命令在 CPL 中调用子程序

描述: CALL 命令提供从 CPL 程序中调用子程序的可能性。

适用于编程的情况如下:

- 使用 CALL 命令后，可直接调用已编制的子程序。
- CALL 命令必须在单个 CPL 块中编程。
- 子程序调用为局部类型。

CALL 关键字的后面是程序名称。紧接着是用方括号括起来的转换参数，最后以“DIN”标识符（为影响链接进程）结束指令。



在包含 CALL 指令的 CPL 块中，不得使用 a": "。后续的 CPL 命令必须编入新的 CPL 块。

示例:

```
50 IF A% = 1 THEN
51 CALL P999
52 ENDIF
:
```

调用子程序调用“DIN”标识符帮助影响 LINK 过程（“准备”）:

如果使用 CALL 调用子程序并将“DIN”标识符编程为结束，则控件将从链接过程中调用的子例程排除在外。例如，采用这种方式，可使包含数个 SR 调用的主程序的链接进程大大加快。

示例:

```
50 IF A% = 1 THEN
51 CALL P999 DIN          SR“P999” 未链接。
52 ENDIF
M30
```

在下列情况下，强烈建议将“DIN”标识符放入程序之中:

- 仅由 DIN 块或简单的 CPL 指令组成（参见 第 3.2.3 章 “手动数据输入期间的 CPL 元素，缓冲的数控块指定和未链接的子程序” 第 23 页）和
- 没有调用其他的子程序。

如果包含 CPL 元素的子程序由于“DIN”标识符而从链接中排除，并且包含无效的 CPL 元素，则控件在程序运行时显示相应的错误消息。

还有一种选择，也可以将“DIN”标识符**作为注释**插入要调用子程序的第一行。该控件不会链接该程序。

示例：要调用程序中的“DIN”标识符

示例：要调用程序中的“DIN”标识符

```
N10 (DIN)                SR"P999" 未链接。
N20 ...
:
```

3.9.8

向子程序传递参数

一般信息

有两个选项可将参数传输到子例程：

- 通过 **CPL 参数列表**或
- 通过**地址参数列表**。

CPL 参数列表用于使用局部变量 P1 至 P <n> 访问子例程中的传送值。分配参数值变量名称由调用程序中的编程序列执行。在调用子程序时地址参数列表的情况下，指定子程序中的变量名称。因此，呼叫中的编程序列并不重要，并且可以在呼叫和子程序中使用“通话”变量名称。就其功能来说，这两种方法等同。

地址参数列表

利用**地址参数列表**，可以向子程序传递参数。为此，必须在紧随子程序调用后的列表中编定参数，用圆括号“(”和“)”将列表括起来。各个参数用逗号分隔。

使用参数地址字和参数值的组合（可使用空格或“=”分隔）在需调用程序中编定将被传递的参数。

示例：以下示例中向子程序地址“A”传递实数值“3.4”的效果等同。

```
N10 P子程序 (A 3.4)
N20 P子程序 (A = 3.4)
N30 P子程序 (A 3.4)
```

地址名称的明确性

在 N30 示例中，地址参数和值之间的分隔不明确。在子程序中，值“.4”可以分配给变量“A3”，值“3.4”可以分配给地址（CPL 变量）“A”。控制器寻找最后一个字母数字形式的字符，以区分地址和值，并将剩余部分解释为值。

如果应该将值“.4”赋予变量“A3”，则应将“=”编入“A3”和“.4”之间。

```
N40 P subroutine (A3 .4)。
```

这种方式特别适用于字符串的传递。需要与地址参数区别。

字符串应始终在引号之间进行编程（“”）。如果在反向逗号之间写入字符串，并且在此字符串中需要额外的反向逗号，则额外的反向逗号前面必须加反斜杠（\）。反斜杠则由两个反斜杠表示。

如果字符串非常简单（没有前导或结束空格，没有逗号，没有“括号关闭”），引号可以省略。在这种情况下，反斜杠（\）被解释为正常的反斜杠。

例子

字符串参数

```
N10 P Up (ST ="一个字符串（带圆括号）")
N20 P Up (ST ="带有“Text”的字符串")
```

```
N30 P Up (ST="Pfad: \\mnt\\Test.npg")
N40 P Up (ST = SimpleString)
```

在这些例子中，赋值符号(=)可以被替换为空白。



如需使用地址参数列表向子程序传递参数，则必须在第一行编入关键字“(ADR_PARA)”。额外的 ID“(DIN)”确保子程序没有链接并且没有参数值被传送。

允许的参数地址

- 子程序的 CPL 变量

允许的参数值

- 数字
- 字符串常量
- 算术 CPL 表达式 (CPL 括号 “[” 和 “]” 中)

为了传输多个参数，地址字和参数值对相互之间使用逗号分隔：

```
N50 P subroutine (A3, B5.2)。
```

不允许两个或多个连续逗号。

与 CPL 参数列表相反，参数顺序不重要。子程序中不赋值给非编程地址。

值可以传递至子程序中的每个 CPL 变量。与 CPL 参数列表不同，不必使用 P1、P2% 等特殊变量名。

示例 1:

使用相同变量名

主程序:

```
10 A=5.5

N20 P 子程序 (A = 3.4)

N30 P Subroutine (A=[A])
```

子程序:

```
(ADR_PARA)

N10 G0 X0

N20 G1 X[A]
```

说明:

主程序将值 5.5 赋予局部变量“A”。块 N20 中，子程序被第一次调用，值 3.4 被赋予子程序的局部变量“A”。块 N20 中，子程序使 X 轴从位置 0 运动由子程序局部变量“A”决定的 3.4。块 N30 第二次调用该子程序时，调用程序的局部变量“A”的值被赋予子程序的变量“A”。

一般规则:

左侧定界符之前为子程序的局部变量。定界符右侧方括号中为调用程序的局部变量或其它允许的表达式。

示例 2:

使用多个参数传递字符串

主程序:

```
10 A=5.5
N20 P子程序 (A = 3.4, B [A * 3], UP ="ASubroutine")
...
```

子程序:

```
(ADR_PARA)

10 IF B=NUL THEN
20 B=0
30 ENDIF

N10 G1 X[A] Y[B]

40 DIM UP$ (20)

N20 P [UP$]
```

说明:

主程序将值 5.5 赋予局部变量 "A"。块 N20 中，子程序被调用，值 3.4 被赋予子程序的局部变量 "A"。如果没有对变量 B 编程，则子程序将值 0 赋予变量 B。此后，在 N10 中 X 轴进给至位置 3.4，Y 轴进给至位置 3 * 5.5 = 16.5，因为调用子程序时 3.4 被赋予 A 且 16.5 被赋予 B。第三个参数规定子程序中调用了另一个被称为 "OneSubroutine" 的子程序。



如果选择的调用程序没有链接，并且如果一个字符串被传递到一个子程序中，则必须使用 PDIM 命令（而不是 DIM 命令）（也参考 CPL 参数列表）。

CPL 参数列表

- 描述:
- 利用 **CPL 参数列表**，可以向子程序传递参数。为此，必须在紧随子程序调用后的列表中编制参数，而且该列表必须用方 CPL 括号 "[" 和 "]" 括起来。各个参数值之间必须用逗号相互分隔。
- 在子程序调用过程中被转移的参数始终都是通过变量 P1、P2、P3，根据参数转移的顺序来获得地址的。
- 参数也可以通过 P1TEST、P2XYZ 等方式来寻址。但 P1、P2 等变量后面的大写字母将被忽略（P1 = P1TEST = P1XYZ）。
- 以下参数属于允许范围:
- 数字（对于小于 1 的浮点数，零点必须在该点的前面编程，例如 0,134）。
 - CPL 字符串常量（在引号中: "<CPL string constant>"
 - CPL 变量
 - CPL 算术表达式
- 示例:
- 在 SR 999 中，P1 的值为 2.75;
 - 在参数转移时，P2 的值为变量 X%;

- P3 的值为 0。
- 如果 P2 需在子程序中代表整数型值，只需向 P2 添加 % 字符。这种标识变量类型的模式也可用于其他的变量类型。
- 在该子程序中，可以把各个参数的值赋予补充变量。

主程序：

```
50 IF A% = 1 THEN
51 CALL P999 [2.75, X%, 0]    使用参数传递进行子程序调用
52 ENDIF
M30
```

子程序 P999:

```
1 FACTOR=P1 : XVALUE%=P2% : KORRTAB%=P3%
N1 G1 X[XWERT%*FAKTOR]
N2 G22 K[CORRTAB%]
```

在下列情况下必须使用 PDIM 命令：

- 一个具有字符串常量作为传递参数和的子例程
- 调用程序被选中而没有链接。

语法：

PDIM <ParameterName>(<ArraySize>)	
	如果编制的数组太小或者完全丢失，则控制器将显示零件程序错误“无效变量”。

表格 3-7: PDIM 语法

示例：

```
主程序：
N10 (DIN)
:
N50 P UP ["测试"]
M30
子程序：
10 PDIM P1$(4)
M30
字符串变量 P1$ 的值为 "TEST"。
```

向模态子程序传递参数

模态子程序不能像非模态子程序那样通过传递参数进行参数化。以下内容应考虑在内：

为参数数控 P / SubProg / ModaSubProg / SwiOn / NofParSubProg (MP 3090 00007) 中的每个模态子程序指定要传送给模态子程序的参数数量。在那里设置的值还指定了应该传递给下一个模态子程序调用的参数值的数量。

使用 CPL 参数列表时，变量 P1 .. 在完成模态子程序后，Pn (n 在参数数控 P / SubProg / ModaSubProg / SwiOn / NofParSubProg (MP 3090 00007) 中指定) 被传送并传递到下一个子程序调用。

如果使用地址参数列表，前 n 个编入的地址被传递至下一调用。不传递未编入的地址。

参数列表中的块转换

在一些加工过程的情况下，要求快速移动运动，例如， G0 进给运动不是直接执行，而是例如， 被替代的进给运动替代。例如，对于高度控制的激光切割

数控编程基础知识

机，如果高度控制不足以防止在高进给速度下工具头与工件发生碰撞，那么就是这种情况。必须“提高”高度控制的轴 / 坐标，而不是直线运动。这个例子被称为“Leap Frog”。由于数控程序经常不能被重写，因此建议使用 C 数控支持来涵盖此用例。当然，这个解决方案也可以应用于其他用例。

数控函数“SubPConv”将随后编程的坐标和值的语法转换为地址子程序参数列表。对于所描述的用例，必须开发一个子程序（例如“LeapFrog”），该子程序需要 G0 进给运动的时间作为参数并产生其他运动。为了满足这一要求并避免数控程序必须更改，数控函数必须另外配置，用“SubPConv LeapFrog”代替数控函数“G0”。数控函数“SubPConv”的功能类似于普通的 P 子程序调用 - 子程序名可以在数控函数后面找到，并且应该包含绝对路径规范（以避免由于搜索顺序造成的不确定性）。

为了转换一个典型的“G0 X100 Y220” G0 进给运动，数控字符串“SubPConv LeapFrog X100 Y220”首先在本例中通过替换宏来生成。在第二步中，转换功能将子例程名称后面的字符串转换为有效的子例程程序语法列表：“SubPConv LeapFrog (X=100, Y=220)”。

因此，子程序“LeapFrog”可以访问传送的 CPL 变量“x”和“y”并生成新的遍历动作。

限制：

数控函数“SubPConv”之前编程的轴值或坐标值首先在同一数控块中加工。例如，“X100 Y200 SubPConv LeapFrog”在 (100, 200) 上运行轴，随后调用“LeapFrog”子程序，不传送任何参数。

如果轴或坐标列表被不是轴或坐标的语法元素中断，则参数列表转换在此语法元素之前完成。然后在子程序调用之前执行在该语法元素后编程的轴或坐标命令。示例：“SubPConv LeapFrog X100 G91 Y200”在 Y 中产生相对（G91）约 200 的定位，并在带子参数列表“LeapFrog (X = 100)”的子程序中调用。

如果（例如）“G0”的语法替换由“SubPConv LeapFrog”使用宏功能来应用，则这适用于跨系统。但是，并不总是需要或可能调用子例程（例如，在 InitString 中），以便偶然报告无意的错误消息。确保在关键或敏感情况下不会出现宏搜索文本！

用例 LeapFrog

假设 1：每个用 G0 编程的快速移动进给运动都应该执行一个“LeapFrog”。G00 和 G000 不受影响以保证正常的直线进给运动。

假设 2：过程和机器指定的子程序“LeapFrog”接受 2 个参数：X 和 Y，因为（笛卡尔）进给运动发生在这些坐标中。

宏定义（例如 /usrfep/macro/MacroDef_G0.xml）
程序

```
<MacroDef>
  <Macro Type="DIN" Overlay="Yes">
    <Name>G0</Name>
    <Replace>SubPConv LeapFrog</Replace>
    <Comment>Replace every occurrence of G0 with "SubPConv LeapFrog"</Comment>
  </Macro>
</MacroDef>
```

子程序 / usrfep / 跳蛙
程序

```
(ADR_PARA)

; total retract height
10 RETRACTHEIGHT = 10
; height which has to be moved vertically
20 VLIFT = 2

; read current position (start point)
30 CUR_X = PCS("X")
40 CUR_Y = PCS("Y")
50 CUR_Z = PCS("Z")
```

```
; if no coordinate is given, the endpoint equals the start point
60 IF (X = NUL) THEN
70   X = CUR_X
80 ENDIF

90 IF (Y = NUL) THEN
100  Y = CUR_Y
110 ENDIF

; calculate the length of the desired movement in X/Y plane
130 DX = X - CUR_X
140 DY = Y - CUR_Y
150 LENG = SQRT(DX * DX + DY * DY)

; do some spline calculation (application dependent)
120 L_PARAM=2*(RETRACTHEIGHT - VLIFT)

; external influence: If @1 == 0, then enable Leap Frog,
; else do a linear G0 movement without Leap Frog

160 IF (@1 = 1) THEN
N170   G000 X=[X] Y=[Y]
180 ELSE
      ; Leap-Frog-Logik

190   IF (LENG < 2*RETRACTHEIGHT) THEN
      ; too short for leap frog. Do NOT use rapid mode (as the
      ; height control may not be dynamic enough even for short
      ; movements
N200   G1 X=AC([X]) Y=AC([Y])
210   ELSE
      ; generate 3 movements:
      ;   vertikal upwards
      ;   horizontal
      ;   vertikal downwards
      ; use spline cornering (SCOL) to increase corner velocity
N220   G000 Z=AC([RETRACTHEIGHT + CUR_Z]) SCOL(L1=[L_PARAM], L2=[L_PARAM])
N230   G000 X=AC([X]) Y=AC([Y]) SCOL(L1=[L_PARAM], L2=[L_PARAM])
N240   G000 Z=AC([CUR_Z])
250   ENDIF
260 ENDIF
```

例子

编程示例

N10 G0 X100 Y200 生成（等效）N10 P LeapFrog(X=100, Y=200)

N20 G0 X100 产生（等效）N20 P LeapFrog (X = 100)。Y 坐标保持其位置。

N30 G00 X100 没有改变（G00 未被宏定义文件替代），即没有执行 LeapFrog 子程序。

N40 G0 X100 M992 Y100 产生 N40 P LeapFrog (X = 100) M992 Y100，即轴 Y 首先用原始 G 代码（例如 G1）运行。随后，调用 LeapFrog 子例程。

3.9.9 配置和错误排除

数控 控制的默认设置不允许超过 8 个子程序层次。如果使用 8 个以上子程序层次，则会显示以下错误消息。列出了恢复错误所需的步骤。

通道错误 2085

错误文字：

打开的文件太多

原因：

链接或编辑时打开了所有子程序。由一个机床参数对每个通道所能打开的文件个数进行限制。

排除方法：

提高机床参数 7060 00210。



机床参数应设为主程序所到达最大嵌套深度另加 3，加 CPL 中所打开文件的数量。

通道错误 1927

错误文字:

块准备限于 NNN 块。

原因:

每次子程序跳转使用了一个数控块。如果编定了过多子程序跳转而无进给运动，所用数控块无法及时启动。这造成了所谓的资源缺乏。

排除方法:

- 增加参数 / 数控 0 / LookAh / Ch / NofBlkPrep (MP 7060 00110)
- 或 -
- 使用 G9 禁用相应区域的块前瞻。然后，如必要则使用 G8 启用块预读
- 或 -
- 如果可能，编程子程序跳转之间的遍历运动。

通道错误 2716

错误文字:

运行时间内存不足（堆栈）。

原因:

子程序被调用时，被调用程序的局部 CPL 变量保存于 CPL 堆栈。如果出现该错误，则所要求的 CPL 的堆栈内存高于设定值。

排除方法:

- 参数数控 0 / Mem / ChMemCh / CplStack (MP 7070 00010)，增加索引 1
- 或 -
- 如果可能的话，使用更少或更小的 CPL 变量

通道错误 825 或 465

错误文字:

程序 XXXX 的内存不足

原因:

链接表内存无法满足规定的程序。

排除方法:

参数数控 0 / Mem / ChMem / Ch / CplLinkBuff (MP 7070 00010)，增加索引 2

3. 10 宏编程

3. 10. 1 概览

宏是用于数控块部件的输入工具。

宏包括:

- 名称，
- 替换文本，和
- 属性（DIN 或 CPL）。

在定义文件中对宏进行配置。具体功能和程序说明见《“Rexroth IndraMotion MTX 功能描述”》的“宏编程”一章。

宏一般仅在被配置的环境（DIN 或 CPL）中起作用。此外，宏的使用与 CPL 或 数控 字相似。

例子

使用宏

宏的名称:	G0
替换文本:	ACC (X3.0, Y3.0, Z3.0) G0
编定块:	N10 G0 (NIPS) M5
作用类似:	N10 ACC (X3.0, Y3.0, Z3.0) G0 (NIPS) M5

详细信息参见 “Rexroth IndraMotion MTX 功能描述” 手册。



宏能够超驰数控函数。由此，现有零件程序的反应可能与预期的有所不同。

应避免宏超驰数控函数。

建议为宏规定独特和易识别的名称。

3. 10. 2 带参数的宏

为了以更灵活的形式设计宏，用户能够定义可传输化的宏.. 如果零件程序需调用带参数的宏，参数必须接圆括号中接宏名称。参数使用逗号分隔。如果参数需含有逗号或圆括号，参数必须写在引号之间。如果需在参数文本中传递引号，则需前接反斜线 (“\”)。

例子

用引号括起来使用宏

宏的名称:	注
替换文本:	PRN#(0, {1})
编定块:	10 Note (“\”Value A\$=”, A\$ ”)
作用类似:	10 PRN# (0, ”Value A\$=”, A\$)

参数作为与他们编定时相同形式的字符串，插入替换文本。不检查语法错误。

例子

使用带参数的宏

宏的名称:	Travel_X
替换文本:	G0 X [{1}]
编定块:	N10 Travel_X(@J) M5
作用类似:	N10 G0 X [@J] M5
宏的名称:	移动
替换文本:	G0 (NIPS) ACC ({2}) {1}

编定块:	N10 移动 (X100 Y100, “X3.0, Y3.0, Z3.0”) M5
作用类似:	N10 G0 (NIPS) ACC (X3.0, Y3.0, Z3.0) X100 Y100 M5

详细信息参见 “Rexroth IndraMotion MTX 功能描述” 手册。

3.11

标签编程和跳转指令

3.11.1

概览

跳转指令可用于在某些入口点继续执行程序。

IndraMotion MTX 为跳转指令提供以下功能:

•	标准数控块 (DIN) 的标签编程
•	CPL 块的标签编程
•	GoAhead (GOA): 向前跳转至标准数控块
•	GoBack (GOB): 向后跳转至标准数控块
•	GoCond (GOC): 有条件跳转至标准数控块
•	GoTo: 无条件跳转至标准数控块
•	CPL 跳转 (GOTO): 跳转至任何程序块
•	列岛: 子程序返回到任何程序块



由于跳转目标的地址在逆时针旋转期间已经计算过，所以 CPL 跳转 GOTO 的处理速度要快得多。

然而，跳转目标仅在 DIN 命令的程序运行期间确定。必须在程序中搜索标签，跳转目标远离 GOTO 命令需要的时间越长。

建议使用 CPL 命令。

它可能不会被跳到一个

CASE-LABEL... LABEL-OTHERWISE-ENDCASE 指令使用 CPL 字符串 GOTO 。

标准数控跳转命令 GoAhead, GoBack, GoCond 和 GoTo 不得用于从 CPL 重复命令或分支命令跳转或跳入。

这会影响 CPL 命令:

- 重复直到
- WHILE - DO - END
- FOR STEP TO NEXT
- IF THEN ELSE ENDIF
- CASE - LABEL ... LABEL OTHERWISE ENDCASE

3.11.2

标准数控块和 CPL 块中的标签

标签是跳转指令的跳转标记。IndraMotion MTX 可区分标准数控块 (DIN) 和 CPL 块中的标签。

标准数控块 (DIN) 中的标签编程:

- 跳转目标必须直接编入块的开头。

- 在带有块编号的块中，跳转目标直接位于块编号的后面，并且必须用空格分隔。
- 标签名称可以包含 2 到 32 个字符。字符允许为字母、下划线和数字，但前两个字符不得为数字。字母区分小写和大写。
- 在跳转目标处，必须在标签名称后编入一个冒号。



由于兼容性的原因，DIN 标签应专门在**块编号后面**编程。如果标签直接在程序段的开始处编程，并且以与数控函数相同的字符序列开始，则会发生语法错误。

由于新版本中总是有新的数控函数，因此可能会由于语法错误而导致现有程序不再以新版本运行。

强烈建议在块后面编制 DIN 标签以避免此问题。

CPL 块中的标签编程

- 跳转目标直接位于块编号的后面，并且必须用空格分隔。
- 标签名称由小数点后跟 ASCII 字符组成，以大写字母开头。
- 标签不得是变量。

3.11.3 GoAhead (GOA) 向前跳转至标准数控块

描述：在跳转目标（标签）处无任何条件地继续程序执行。

以下情况适用：

- 必须定义跳转目标。
- 从当前程序块的角度来看，跳转目标必须位于文件结尾的方向。

有关所需标签编程的信息，参见第 第 3.11 章 “标签编程和跳转指令” 第 52 页 页。

语法：

GoAhead <Label>	
缩写形式：GOA	
其中	
<Label>	跳转目标名称 2 到 32 个字符 字符允许为字母、下划线和数字，但前两个字符不得为数字。

表格 3-8: GoAhead 语法

示例：

```
:
N40 GoAhead LABEL1      向前跳转至目标 "LABEL1"。
:
N80 LABEL1:             跳转目标 "LABEL1" 的标签编程。
:
```



标准数控跳转命令 GoAhead 不得用于跳转到 CPL 重复或分支指令，或跳出 CPL 重复或分支指令。

有关相关 CPL 命令的信息，请参阅 第 3.11.1 章 “概览” 第 52 页 。

3.11.4 GoBack (GOB) 向后跳转至标准数控块

描述：在跳转目标（标签）处无任何条件地继续程序执行。

以下情况适用：

- 必须定义跳转目标。
- 从当前程序块的角度来看，跳转目标必须位于文件开始的方向。

有关所需标签编程的信息，参见第 第 3.11 章 “标签编程和跳转指令” 第 52 页 页。

语法：

GoBack <Label>	
缩写形式：GOB	
其中	
<Label>	跳转目标名称 2 到 32 个字符 字符允许为字母、下划线和数字，但前两个字符不得为数字。

表格 3-9: GoBack 语法

 请注意，在编制向后跳转时，可能会意外造成无限循环！

示例：

:	
N40 LABEL1:	跳转目标“LABEL1”的标签编程。
:	
N80 GoBack LABEL1	返回到目标“LABEL1”。
:	如果在 N40 和 N80 之间没有补充的跳转命令，则程序将在 N40 和 N80 之间进行无限循环。

 标准 数控 跳转命令 GoBack 不得用于从 CPL 重复或 CPL 分支指令跳入或跳出。
有关相关 CPL 命令的信息，请参阅 第 3.11 章 “标签编程和跳转指令” 第 52 页 。

3.11.5 GoCond (GOC) 有条件跳转到标准 NC 程序块

描述：如果在程序段准备时“条件跳转”输入信号在通道接口上有效，继续执行指定程序段号的程序。

以下情况适用：

- 必须存在输入的块编号。
- 输入的块编号可位于与当前程序块相关的程序文件中的任何位置。

语法：

GoCond N<Number>	
缩写形式：GOC	
其中	
<Number>	块编号 编号为整数或者实数形式，最多不超过 15 位数。 如果目标块编号采用先导零编制（例如，“N0020 ...”），则这里也必须编入先导零。 （“GoCond N0020”）。

表格 3-10: GoCond 语法

示例：	10 WAIT	在 N20 前的所有块都已经处理之前，停止块准备。
	N20 GoCond N090	如果在准备块 N20 时，接口信号“有条件跳转”有效，则跳转至 N090。
	:	
	:	
	:	
	:	
	N090 ...	
	:	



标准数控跳转命令 GoCond 不得用于跳转到 CPL 重复或分支指令，或跳出 CPL 重复或分支指令。
有关相关 CPL 命令的信息，请参阅 第 3.11 章 “标签编程和跳转指令” 第 52 页 。

- 特殊功能和限制：
- 除非可能是编制的先导块编号，在同一个块中不允许有其他程序字。
 - 在块准备和块执行之间的时间段内，通道接口信号“有条件跳转”的变化不予考虑。

如果相关应用不允许这种行为，则用户必须在以前的程序行中编制“WAIT”功能。



请注意，在向文件开始的方向进行跳转时，可能会意外造成无限循环！

3.11.6 GoTo 无条件跳转至标准数控块

描述： 在任何块编号处无条件地继续程序执行。
以下情况适用：

- 必须存在输入的块编号。
- 输入的块编号可位于与当前程序块相关的程序文件中的任何位置。

语法：	GoTo N N<Number>	
	其中	
	<Number>	块编号 编号为整数或者实数形式，最多不超过 15 位数。 如果目标块编号采用先导零编制（例如，“N0020 ...”），则这里也必须编入先导零。 “GoTo N0020”）。

表格 3-11: GoTo 语法

示例：	:	
	N40 GoTo N080	向前跳转至块 N080。
	:	
	N080 GoTo N40	向后跳转至块 N40。
	:	如果在 N40 和 N080 之间不再编程跳转指令，程序将在 N40 和 N080 之间通过无限循环运行！

特殊功能和限制： 除非可能是编制的先导块编号，在同一个块中不允许有其他程序字。



请注意，在向文件开始的方向进行跳转时，可能会意外造成无限循环！



标准数控跳转命令 `GoTo` 不得用于跳转到 `CPL` 重复或分支指令，或跳出 `CPL` 重复或分支指令。
有关相关 `CPL` 命令的信息，请参阅 [第 3.11.1 章“概览”第 52 页](#)。

3.11.7 CPL 跳转（GOTO）跳转至任何程序块

描述： 在跳转目标处无条件地继续程序执行。
以下情况适用：

- 可将 `CPL` 块编号、标准数控块编号或“标签”（跳转标记）指定为跳转目标。
- 跳转的目标可位于当前程序块程序文件内的任何位置。

语法：

GOTO <Target>

表格 3-12: GOTO 语法

示例：

```
10 GOTO N20           跳转至块 N20
N20 X100
30 GOTO 120           跳转至 CPL 块 120
...
120 GoTo .ZIEL1       跳转到标签.ZIEL1
...
150 .ZIEL1
```



它可能不会被跳到一个 `CASE-LABEL...LABEL-OTHERWISE-ENDCASE` 指令使用 `CPL` 字符串 `GOTO`。

有关标签编程条件的信息，参见第 [第 3.11 章“标签编程和跳转指令”第 52 页](#)。

3.11.8 RetTo 子程序返回任何程序块

描述： 在没有任何条件的情况下执行一个或多个级别的子程序返回，并在请求的目标程序段继续程序序列。
以下情况适用：

- 块编号或跳转标签（标准数控块或单独 `CPL` 块）被指定为目标块
- 目标程序段可以在目标程序文件中关于子程序调用位置的任何位置
- 如果没有指定除目标块之外的其他参数，则返回恰好一个程序级别
- 或者，可以返回到多个程序级别。因此，使用级别参数来指定与当前程序级别的差异（负值）或指定程序级别（正值）作为目标级别

语法：

RetTo (<TargetBlock>{,<TargetLevel>})

其中

<Target block>	块号或跳转标签（分别用于标准数控块或 CPL 块）
<TargetLevel>	绝对或增量程序级别 > 0: 返回到指定的绝对程序级别 = 0: 返回到主程序 < 0: 返回到指定数量的程序级别 如果没有指定目标等级，则等于 -1，即返回一个程序等级

表格 3-13: 语法 RetTo

特殊功能和限制:

RetTo 不被允许

- 在通道初始化程序（"Init 程序"）及其子程序中
- 在主程序中
- 在操作模式"手动数据输入"中作为直接输入或在其中一个程序的第一个程序级中

此外，如果 RetTo 导致错误

- 没有指定目标块
- 没有找到目标块
- 绝对目标水平不低于当前程序水平
- 对于增量目标级别，要返回的程序级别的数量高于当前程序级别

本地子例程是一个单独的程序级别，因此像所有其他子例程一样处理。

示例:	N100 RetTo (N500)	返回一级到标准数控块 N500
	N100 RetTo (500)	返回一级到 CPL 块 500
	N100 RetTo (STD 标签)	使用跳转标签"STD LABEL"将标准数控块返回一级
	N100 RetTo (.CCL 标签)	以跳转标签".CPL LABEL"返回一级到 CPL 块
	N100 RetTo (N500, -1)	返回一级标准数控块 N500（与第一个示例相同）
	N100 RetTo (500, -2)	通过两级返回到 CPL 块 500
	N100 RetTo (STD LABEL, 2)	通过跳转标签"STD LABEL"返回程序级 2 到标准数控块
	N100 RetTo (.CPL LABEL, 0)	用跳转标签".CPL LABEL"返回主程序到 CPL 模块
 使用 RetTo，它可能不会跳转到 CASE-LABEL...LABEL-OTHERWISE-ENDCASE 指令。		
 使用 RetTo，当返回到标准的数控块跳转标签时，它可能不会跳出 CASE-LABEL...LABEL-OTHERWISE-ENDCASE 指令。		

有关标签编程条件的信息，参见第 3.11 章“标签编程和跳转指令”第 52 页。

3.12 决定和分支指示

3.12.1 概览

决策和分支指令用于执行与特定事件相关的单个程序块和部分或整个子例程。

IndraMotion MTX 提供以下可能性:

- 标准数控块 (DIN) 的 "跳过块" 功能
- CPL 指令 IF-THEN-ELSE-ENDIF
- CPL 指令 CASE-LABEL... LABEL-OTHERWISE-ENDCASE

3. 12. 2 "跳过块"功能

描述: 借助于该功能, 控制器可跳过各个标准数控块 (DIN)。为此, 必须在相应程序行的开头编入 "/" 符号。

只有当相应通道的位接口设定了接口信号 "qCh_BlockSlash" (跳过块) 时, 才能跳过已标记的程序块。

示例:

```
:                               接口信号"跳过块"被激活。
:
/N100 ...                      块 N100 被忽略。
:                               接口信号"跳过块"被取消。
:
/N300 ...                      块 N300 被处理。
:
```



"跳过块"功能只能用于标准数控块 (DIN) !

3. 12. 3 CPL 指令: CASE-LABEL... LABEL-OTHERWISE-ENDCASE

描述: 在程序中, 经常需要查询某个整数表达式或者整数型变量两个以上的状态。在这种情况下, 只有采用多重嵌套的 IF 命令才能通过 IF 命令进行查询。这就要求额外的运算时间, 并会削弱程序的可读性和可维护性。

使用 CASE 结构可以避免这些不足:

```
CASE <integer expression> OF
LABEL <int. constant>[ , <int. constant>][ : <instruction>]
<Instruction>
:
LABEL ...
:
[OTHERWISE <command>]
<Instruction>
:]
ENDCASE
```

在 CASE 命令之后, 程序分支到 LABEL 命令, 其中, <integer constants> 的其中一个等于 <integer expression> 的值。现在, 在下一个 LABEL 或者 OTHERWISE 命令前的所有命令都将被执行。程序直接分流到 ENDCASE 指令。

如果没有满足该条件的 LABEL 命令, 则程序将分支到 OTHERWISE 命令或者 (如果 OTHERWISE 没有编程) 接分支到 ENDCASE 命令。

CASE 结构的 <Command> 区域可以包含所有的 CPL 命令。最多可以嵌套 10 个 CASE 结构。

示例:

```
10 CASE A% OF
20 LABEL 0: Y=1
```



```
30 LABEL 2
40 Y=Y*Y
50 LABEL 4: Z=Y*Y
60 Y=Z*Z
70 OTHERWISE Y=0
80 ENDCASE

10 CASE (INT(X/Y)+C%) OF
20 Label 1.2 : X=1 : Y=2
30 Label 4.8
40 X=2 : Y=4
50 LABEL 0
60 X=0 : Y=1
70 OTHERWISE X=0 : Y=0
80 ENDCASE

10 CASE INTFELD%(1,2) OF
20 LABEL 1,2,3 : GOTO .MARK1
30 LABEL 4,5,6 : GOTO .MARK2
40 OTHERWISE GoTo .ENDE
50 ENDCASE
```



标准 数控 跳转命令 GoAhead、GoBack、GoCond 和 GoTo 不得用于从 CASE-LABEL...LABEL-OTHERWISE-ENDCASE 命令中跳入或者跳出。



它可能不会被跳到一个
CASE-LABEL...LABEL-OTHERWISE-ENDCASE 指令使用 CPL 字符串 GOTO
。

3.12.4 CPL 指令：IF-THEN-ELSE-ENDIF

描述: 该功能是一个简单的条件分支指令：
“IF 如果满足某个特定条件，THEN 则执行程序，或者 ELSE 另外执行其他程序。”

语法:

IF <condition> THEN <routine> [ELSE <alternative routine>]	
ENDIF	以下情况适用: • 在同一行中条件作为 “IF”，并且在同一行里以 “THEN” 结尾。 • THEN 和 ELSE 程序都是程序分支，在任何情况下都无需进行处理。 • 如果 ELSE 组件被忽略，而且条件未得到满足时，在处理 ENDIF 命令之后，程序将继续运行。

表格 3-14: IF-THEN-ELSE-ENDIF 句法

与循环命令的中止条件相似，IF 命令的条件可以包含算术、三角和逻辑运算。这里也允许嵌套。

IF 命令必须始终以 ENDIF 命令结尾，否则，将无法识别程序的终点，或者可选择程序的终点。由于 ENDIF 命令的位置取决于程序处理逻辑，因此，控制器有时会无法可靠地检测并解释丢失的 ENDIF 命令。这会导致模糊的错误消息。所以，编程人员要养成审核 IF 命令的完整性的良好习惯。

示例：

```
...
10 X = 1
20 .START
30 IF X>=100 THEN
40 GOTO .ENDE
50 ELSE X=X+2.75
60 GOTO .START
70 ENDIF
...
90 .ENDE
...
```



标准 数控 跳转命令 GoAhead, GoBack, GoCond, GoTo 不得用于从 IF-THEN-ELSE-ENDIF 命令中跳入或者跳出。

3. 13 重复语句

3. 13. 1 概览

如果要对与特定条件相关的一个或者多个程序块进行重复处理，则可借助于 **CPL 重复指令** 来进行编程。程序的多次重复也被称为循环。

IndraMotion MTX 提供以下可能性：

- CPL 命令 FOR-STEP-TO_NEXT
- CPL 命令 REPEAT-UNTIL
- CPL 命令 WHILE_DO_END

3. 13. 2 CPL 指令：FOR-STEP-TO-NEXT

描述：

例如，如果重复指令的终止条件必须是程序进程的直接结果，则必须有一个跟踪计数器。

该计数器不要求为 FOR NEXT 循环进行专门编程。计数变量（integer）由规定的计数起始点和终止点决定。如果计数步长偏离 1，则可以单独指定步长（STEP）。

语法：

FOR <CountingVar.>=<InitialValue> [STEP <StepWidth>]
TO <EndValue><Routine>
NEXT [<Count.Variable>]

表格 3-15: FOR-STEP-TO-NEXT 句法

示例：

```
10 FOR I%=1 TO 18
20 XSINUS (I%)=SIN (I%*10)
30 NEXT I%
```

在循环结束后，计数变量提供的值大于最终值（最大步长）。

在这个例子中，0 到 180 度的 SIN 值被写入 XSINE 域。第 30 行中附在"NEXT"上的"I%"仅用于说明目的，可能会被忽略。

也可以采用**可变步长**来编程 FOR NEXT 循环。在这种情况下，步宽变量应该与计数变量的变量类型相同。

示例：程序

```
10 OPENW(1,"P222",130)
20 STEP%=2 : START%=1 : END%=3500 : NJUST
30 FOR COUNTER%=START% STEP STEP% TO END%
40 STEP%=ROUND(STEP%*SQRT(STEP%))
50 PRN#(1,"COUNTER: ",COUNTER%,"step width: " ,STEP%)
60 NEXT
70 CLOSE(1)
```

在程序执行之后，在"P222"文件中出现以下内容：

示例：程序

```
COUNTER:      1 STEP WIDTH:      3
COUNTER:      4 STEP WIDTH:      5
COUNTER:      9 STEP WIDTH:     11
COUNTER:     20 STEP WIDTH:     36
COUNTER:     56 STEP WIDTH:    216
COUNTER:    272 STEP WIDTH:   3175
COUNTER:   3447 STEP WIDTH: 178902
```



标准 数控 跳转命令 GoAhead, GoBack, GoCond, GoTo 不得用于从 FOR STEP TO NEXT 命令中跳入或者跳出。

3. 13. 3 CPL 指令：REPEAT-UNTIL

描述：如果仅需在第一次程序处理之后对重复指令的终止条件进行查询，则可以使用 REPEAT 循环。

语法：

REPEAT <routine> UNTIL <condition>

表格 3-16: REPEAT-UNTIL 句法

示例：

```
:
30 REPEAT
40 X=X+1                循环直到 X = 100
50 UNTIL X=100
:
```



标准 数控 跳转命令 GoAhead, GoBack, GoCond, GoTo 不得用于从 REPEAT-UNTIL 命令中跳入或者跳出。

3. 13. 4 CPL 指令：WHILE-DO-END

描述：如果在初始循环迭代之前要查询重复指令的中止条件：“当满足条件时，执行程序！”。

WHILE 循环结构如下：

语法：

WHILE <condition> DO <routine> END

表格 3-17: WHILE-DO-END 句法

示例：

```
:
30 WHILE SD(9)=0 DO      等待循环，直到 SD （9）的值为 0。
40 I=I+1
```

50 END
:

表格 3-18:

 标准 数控 跳转命令 GoAhead, GoBack, GoCond, GoTo 不得用于从 WHILE-DO-END 命令中跳入或者跳出。

3. 14 可变频

3. 14. 1 变量名

 变量编程包含在 CPL 语言的范围之内！

CPL 中的变量编程用于设计程序，以便在其中进行参数设置，从而使程序执行与当前条件相适应。

- 变量指满足以下条件的、几个特殊边界条件的符号任意名称：
 - 变量名称必须是唯一的。
 - 变量名称不得与保留的 CPL 命令字相同。
- 虽然标准 数控 和 CPL 编程在形式上是分开的，但在理论上，变量名称可以与 数控 功能或者 数控 功能参数同名；例如，虽然系统中存在名称为“X”的轴，但变量“X”也可以定义。
- 变量名称可由任意顺序的大写字母和数字组成；但首字符必须是大写字母。包含字母“N”且后面只有数字的变量名称不是有效的变量名称。字符串与数控块的标识相对应。

 注意：由于变量名中，只有最前面的 8 个字符被用于名称识别，所以，只有这 8 个字符才是有意义的（除非：可定义的固定变量）。

- 指定变量有效范围的变量组总计有 3 个。可通过在变量名称的开头添加标识符来指定变量组。该字符始终被视为变量名称中的有效字符之一！
 *以下是**变量组**及其对应的标识符：*
 - 本地变量：没有特殊的标识符
 - 全局变量：“#”
 - 永久变量：“@”
- 始终通过在变量名称的结尾添加标识符来指定**变量的类型**。如果变量名称超过有效字符数量，这也同样适用。
 赢取可定义永久变量（“wmhperm.dat” and “anwperm.dat”）的声明文件，此 ID 省略。相应的 CPL **数据类型**将被指定。

以下是变量类型及其对应的标识符：

变量类型	CPL 数据类型	具体的变量识别
整数	INT	“%”
带简单精度的浮点数	实数	无特殊标识
带双精度的浮点数	DOUBLE	“!”
逻辑变量	BOOL	“? ”
字符	CHAR	“\$”

表格 3-19: 变量

示例： 局部、全局和永久变量：

10	NUMBER1% = 1	CPL	型	INT	的局部变量
20	#NUMBER2% = 2	CPL	型	INT	的全局变量
30	@36% = 3	CPL	型	INT	的永久变量
40	@ABCD% = 4	CPL	型	INT	的永久定义变量

3. 14. 2 变量组简介

由于可使用子程序的选择，以及由于不考虑正在执行的程序而将变量值提交给中间存储器的可能要求，因此需要声明变量的有效范围。为此，须对以下变量组之间进行区分：

局部变量

局部变量仅在声明这些变量的程序中有效。当该程序到达程序结尾（EP）时，变量将被删除，从而释放所占用的内存。在子程序调用中，调用程序本地的变量名对子程序“不可见”。因此，也可在子程序中将相同的变量声明为局部变量，从而不会因为它们各自名称的相似性而引起冲突。一旦返回到调用程序，原来的局部变量将重新可用于调用子程序时的值。

全局变量

全局变量通过先导 #（数字符号、门或者哈希）字符进行标识。一旦全局变量被赋值，则可被整个程序的所有其他程序部分的内部进行访问、读取和 / 或修改。全局变量在程序结束（EP）后被删除。

永久变量

通过变量名称前面的前导 @ 字符来标识永久变量。可使用任何活动程序对其寻址。永久变量储存在单独的存储区，并且在程序结束后、控制器复位后，以及激活或者取消激活后始终保留。只能通过直接覆盖的方式加以删除。

CPL 类型 INT 的永久变量可以在 @ 1 到 @ 100 的名称下寻址（有关 CPL 类型 INT 的详细信息，请参阅第 3. 14. 3 章“变量类型”第 66 页）。为提高程序的可读性，此类永久变量的标识也可以通过对数字添加字母来进行增强。

另外，永久一维数组变量 @_R 可使用 100 个 CPL 型双精度元素。@_RES_DOUBLE 和 @_RES_DWORD 这两个永久变量保留用于内部应用，且不应使用。

可定义永久变量

也可以通过变量名称前面的前导 @ 字符来标识可定义永久变量。
其与“永久变量”的差异如下：

- 可定义永久变量不通过系统软件的某个组件自动声明，而必须通过用户进入名为 “wmhperm.dat”（用于机床制造商提供的专有数据）和名为 “anwperm.dat”（用于特定最终用户的专有数据）的文件中进行**手动声明**。声明句法将在 “wmhperm.dat” 和 “anwperm.dat” 的文件结构部分予以讨论。

在系统启动过程中，控制器先在根目录中搜索文件，然后在用户 FEPROM 中搜索，最后在 FEPROM 中进行搜索。

控制器按文件第一次出现的各自文件名对文件进行解释，并使用其内发现的条目来创建“可定义永久变量”，但前提是这些变量之前不存在。上述名称的文件中未被声明的现有“可定义永久变量”将被删除。

可定义的永久变量的可能最大数量受限于可用的内存空间。如果无法再为生成的变量提供内存空间，则控制器将显示相应的错误消息。

数控编程基础知识

- “可定义永久变量”的名称始终以 **@ 字符和字符串** 开始。该字符串由一个大写字母开头，后面是大写字母或者数字的任意组合。
对于“可定义永久变量”，变量名称的前 **16 个字符** 是有效的。如果它们仅与第 17 个字符不同，则 CPL 将它们解释为单个变量！
- 已定义的永久变量可以是 CPL 类型 **INT，实数，双精度，布尔 或 CHAR**。可通过在变量名称的末尾添加标识符来指定变量类型。
必须在零件程序中输入该规范：
 - @ABCD% def. perm.CPL 型 INT 的变量
 - @EFGH def. perm.CPL 型实数类型的变量（无特定标识）
 - @IJKL! def. perm.CPL 型双精度的变量
 - @MNOP? def. perm.CPL 型布尔的变量
 - @QRST\$ def. perm.CPL 型 CHAR 的变量
- 可使用一维和二维的**数组**。
CPL 型的数组变量**最大数组索引** INT，实数，双精度 或布尔 是 **65535**。对于 CPL 型 CHAR 的数组变量，它是 **1024**。

示例：

@WZNR%(1)=4	INT 型的一维数组@WZNR 的第一个变量（指数为 1）被赋值为 4。
@WZKOR(2,2)=0.2	实数二维数组@WZKOR(@TCor)内的变量（指数为 2,2）被赋值为 0.2。

- 估算新可定义永久变量的可用**数量**：
永久变量的总内存空间：100 千字节（102400 字节）

位置	保留用于	以字节为单位的内存空间	注
1	所有永久变量	102400	总内存
其中以下部分保留用于			
2	@1 - @100 (永久变量)	800	数据类型：INT 为提高程序的可读性，此类永久变量的标识也可以通过数字添加字母来进行增强。
3	管理信息	24	
4	所有可定义永久变量	101576	(4) = (1) - (2) - (3)
其中以下部分保留用于			
5	@_R	823	具有 CPL 型双精度的 100 个元素的永久数组变量
6	@_RES_DOUBLE	40	CPL 型双精度的永久变量，保留用于 内部 应用
7	@_RES_DWORD	35	CPL 型 INT 的永久变量，保留用于 内部 应用
8	新可定义永久变量	100678	(8) = (4) - (5) - (6) - (7)

表格 3-20: 所有永久变量的内存容量

位置	保留用于	以字节为单位的内存空间	注
9	可定义永久变量的名称	不超过 16	每字符 1 字节
10	可定义永久变量的值	1, 4 或 8	INT: 4 字节 双精度: 8 字节 实数: 4 字节 布尔: 1 字节
11	管理信息	20	
12	名称长度为 16 个字符的 CPL 型双精度可定义永久变量	44	例如: 最大内存分配 (9) + (10) + (11)

表格 3-21: 可定义永久变量的内存容量
双精度和 INT 型“可定义永久变量”的数量

变量类型	变量数量	注
名称长度最多为 16 个字符的 CPL 型双精度	2288	100678/44 = 2288
名称长度最多为 16 个字符的 CPL 型 INT	2516	100678 / (16+4+20) = 2516
名称长度最多为 8 个字符的 CPL 型 INT	3146	100678 / (8+4+20) = 3146
名称长度最多为 16 个字符的数组变量, CPL 型 INT	25160	(100678-16-20) / 4 = 25160
名称长度最多为 16 个字符的数组变量, CPL 型双精度	12580	(100678-16-20) / 8 = 12580

“wmhperm.dat”和
“anwperm.dat”声明文件的文件结构:

表格 3-22: “可定义永久变量”的数量

这些文件可以只包含“可定义永久变量”的声明。每个变量都在一个单独的行中声明，并且必须用**分号**来完成。行末可以有一个可选的注释。

声明行始终具有以下结构:

DEF <variable type> @ <variable name>; [<comment>]

 分离器”;即使没有注释，也必须写出来！

“wmhperm.dat”和“anwperm.dat”示例

```
DEF INT @CDEF;
DEF INT @ABCD      ; CPL 型 INT 的简单变量
DEF REAL @EFGH     ; CPL 型实数的简单变量
DEF DOUBLE @IJKL    ; CPL 型双精度的简单变量
DEF BOOL @MNOP      ; CPL 型布尔的简单变量
DEF CHAR @PSTR1 (3) ; CPL 型 CHAR 的变量，长度为 3
DEF INT @WZNR (9)   ; 具有 9 个变量的 CPL 型 INT 的一维数组
DEF REAL @WZKOR     ; 具有 18 个变量的 CPL 型 INT 的二维数组
(9,2)
DEF CHAR @PSTR2     ; 带有 9 个部分串、每串两个字符的 CPL 型
(9,2)               CHAR 二维数组
```

数控编程基础知识

永久变量应用示例：

```
10 @1 = 1
15 @2_ZAEHLER = 2
20 @ABCD% = 3
25 @EFGH = 4.1
30 @IJKL! = 5.12345
35 @MNOP? = TRUE
40 @PSTR1$ = "ABC"
45 @WZNR%(2) = 6
50 @WZKOR (3,2) = 7.6
55 @PSTR2$(3) = "DE"
```

结构化变量

结构化变量可以被系统数据 (SD) 替换，但由于兼容性的原因而被保留（参见第 7 章 “CPL 命令” 第 371 页）。

结构化变量的特点是具有以句号 (".") 分隔的各种结构级别。结构化变量始终以 "SV." 开头。

需要以 CPL 命令访问的所有结构化变量都必须通过用户进入名为 **machdef.dat** 的文件进行“手动”声明。在系统启动过程中，控制器先在根目录中搜索文件 “machdef.dat”，然后在用户 FEPRM 中搜索，最后在 FEPRM 中进行搜索。第一个发现的文件将在 数控 启动时被检查，同时将创建管理数据和结构化变量用户数据的内存。每次启动时，都将重新创建内存，并将所有的值都预设为 0。

“machdef.dat”文件 仅用于定义结构化变量。每个定义都占用一个单独行，并以分号结尾。

定义行始终具有以下结构：

```
DEF<Variable type> SV.<Variable name> ; [<Comment>]
```

<Variable type> 和 <variable name> 都是字符串，可以包含小写或者大写字母、数字、破折号和下划线。

 对于所有的变量名称，前 14 个字符是有意义的（没有先前的“SV.”）。如果它们之间的差异超过这一点，CPL 或 PLC 将它们解释为单个变量！

示例：

```
DEF DBT1Rec_t SV.A;    “DBT1Rec_t” 型结构化变量的定义

CPL 命令可以访问 SV.A 和 SV.A 的所有结构组件。各种结构部件彼此分开“.”。

10 SV.A.Hd=DBSEA  (“DBT1",-1,-1,"K1=1",FOUND%)
```

3. 14. 3 变量类型

整数变量 (INT)

CPL 型 INT 变量需要 32 位内存。用添加到变量名称的 “%” 字符来识别。其值范围从 -2.147.483.647 到 +2.147.483.647。

```
10 NUMBER% = 4
```

整数变量

浮点变量 (实数)

如果变量名称没有添加特殊标识，则这个变量将被解释为单精度的实数变量。

在这种情况下，变量占用 32 位的内存空间。值范围为 + / -10³⁸。这对应
有 7 个有效位置。

```
10 PI = 3.141593
```

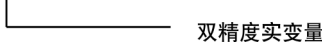


浮点变量（双精度）

如果为变量名称添加一个感叹号 “!”，则该变量将被解释为**双精度**的实数变量。

在这种情况下，变量占用 64 位的内存空间。值范围为 + / - 10³⁰⁸。这对应
有 15 个有效位置。

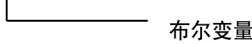
```
10 PI! = 3.141592653589793
```



逻辑变量（布尔）

通过为变量名称添加一个问号 “?” 来标识这类变量。逻辑变量（布尔变量）
的值只能为 **TRUE**（真）或者 **FALSE**（假）。可将其用于储存整个程序执行过
程中所需的逻辑状态或者条件。

```
10 START? = FALSE
```



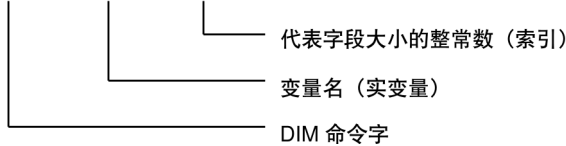
域变量（数组）

数组变量允许在内存空间中以单一的变量名称保留一维或者两维的域（数
组），该域由一个或者多个同类型的变量组成。

数组定义可用于 CPL 型 INT，实数，双精度，布尔和 CHAR 的变量。为
了访问数组中的单个数组元素，除了指定数组变量的名称外，还指定了数组索
引和 / 或索引。

示例： 尺寸标注数组变量

```
10 DIM FELDDVAR (2,3)
```



访问数组变量

```
100 FELDDVAR (1,1) = MCS (1)  
110 FELDDVAR (2,1) = PCS (1)  
120 FELDDVAR (1,2) = MCS (2)  
130 FELDDVAR (2,2) = PCS (2)  
140 FELDDVAR (1,3) = MCS (3)  
150 FELDDVAR (2,3) = PCS (3)
```

在初次访问域变量之前，必须用整数型常量对指数范围和 / 或域大小进行
标注：

- CPL 型 INT 和实数的数组变量的数组大小：
不超过 65536
- CPL 型 CHAR 和实数的数组变量的数组大小：
不超过 1024

```
DIM <VariableName> (<array size1>[,<array size2>])
```



采用 DIM 进行尺寸标注不适用于“可定义的永久变量”。相反，对这些变量的尺寸标注必须在文件 “wmhperm.dat” 或者 “anwpwerm.dat” 中进行。

CPL 型 CHAR 和字符串变量的变量

CPL 型 CHAR 变量由以下的 “\$” 符号识别。这种类型的变量即可以是单一的字符，也可以是完整的字符串。

但是，只有当字符串被存储在 CPL 型 CHAR 变量的一维或者二维数组中时，才能使用字符串命令（参见“处理字符串”章节）。这要求通过 DIM 命令对域进行声明。

则该数组中的每个 CPL 型 CHAR 变量只包含字符串中的一个字符。

由 CPL 型 CHAR 变量组成的一维数组被称为字符串变量。访问一维 CPL 型 CHAR 变量时，不必输入指数。但在访问二维变量时，必须输入指数。

```
示例： 1 REM 字符串变量 AB（长度 10）
        2 DIM AB$(10)
        3 REM 3 字符串变量 CD（每个长度 5）
        4 DIM CD$(3,5)
        5 AB$ = "Z"
        6 CD$(2) = "ABC"
```

变量概述

变量组	变量名	CPL 类型		可能的数组 (X=是)
本地	最多 8 位有效字符	%	INT	X
			实数	X
		!	DOUBLE	X
		?	BOOL	X
		\$	CHAR	X
全局 #	最多 8 位有效字符 (含 “#” 字符)	%	INT	X
			实数	X
		!	DOUBLE	X
		?	BOOL	X
		\$	CHAR	X
永久 @	1 - 100			
可定义永久 @	最多 16 位有效字符	%	INT	X
			实数	X
		!	DOUBLE	X
		?	BOOL	X
		\$	CHAR	X

表格 3-23: 变量概述

3.14.4 标准 数控 编程中的变量

常用的数控函数或带变量的数控函数参数应在标准数控编程（DIN）中的零件程序中进行参数化。

在进行这类赋值时，请注意，标准 数控 编程中的 CPL 表达式必须采用方括号 "[" 和 "]" 括起来。

示例： **标准数控块的参数设定：**

```
10 Angle = 45
20 VAL1 = 1.5
30 VAL2 = 1.5
40 XPOS = 10.2
50 YPOS = 5.73
60 FEEDRATE = 1000
N70 Rotate ([ANGLE])
N80 Scale (X[VAL1],Y[VAL2])
N90 G1 X [POS] Y [POS] F [FEEDRATE]
```



不得使用 CPL 变量对块编号进行参数设定！



调用子程序的地址不用于变量句法！

3.14.5 评价 CPL 中功能错误的变量 ERRNO"ERRNO"

CPL 变量 ERRNO（被调用的功能可在其内返回错误）可作为传递参数被传递到 CPL 功能，该功能在出现错误时会生成运行时错误。

CPL 变量 ERRNO 是可编程 CPL 功能的可选参数，在 CPL 功能中可将其作为参数输入。并且可将其编入功能的任何位置。

如果 ERRNO 未被编入，则在出现错误时，CPL 功能会生成运行时错误或者警告。

如果将 ERRNO 编入，则 CPL 功能**不会**生成运行时错误或者警告。在这种情况下，必须在 CPL 程序中对 CPL 变量 ERRNO 进行评价，而且必须明确生成运行时错误（使用 SETERR）或者警告（使用 SETWARN）。

如果 CPL 功能已经直接执行，则 ERRNO 的值为 0。通过负值确认错误。负值的含义通常是有效的；但并不是所有的值都与 CPL 功能相关。CPL 功能可能的错误值均指定给适当的功能。

通常有效的错误值清单：

0:	访问正常
-1:	参数错误
-2:	坐标 / 轴 / 主轴不存在
-3:	通道中无效的坐标 / 轴
-4:	轴不是伪坐标
-5:	通道不存在
-6:	功能只能在自己的通道中调用
-7:	数据无法读取
-8:	包括路径在内的源文件名太长
-9:	无法访问源文件
-10:	包括路径在内的目标文件名太长

数控编程基础知识

- 11: (源或目标) 文件名无效
- 12: 无法复制
- 13: 数据无法写入
- 14: XML 表已存在
- 15: 未发现 XML 表
- 16: 无效文件扩展名
- 17: 无效表类型
- 18: 插入位置错误
- 19: 超出最大轴数量
- 20: 根标签错误
- 21: 无法访问文件
- 22: 无效键
- 23: 数据块被锁定
- 24: 无权读取该文件
- 25: 无权写入该文件
- 26: 含有无效参数值的命令
- 27: 含有无效参数类型的命令
- 28: 访问 XML 表时的错误
- 29: 数组指数不正确
- 30: 内存不足
- 31: 搜索条件不正确
- 32: 通道中该数据库表的数据块被锁定
- 33: 通道中数据块未锁定
- 34: 赋值右侧的变量未被赋值
- 35: 系统数据访问内部错误
- 36: Sercos 服务通道受阻
- 37: Sercos 标识号不明
- 38: Sercos 标识号无效
- 39: 轴号无效
- 40: 冻结块不存在
- 41: 命令不允许
- 42: 违反限值
- 43: 元素号无效
- 44: 没有可用于轴的测量值
- 45: 无法计算坐标的测量值。对于至少一个属于轴转换成员的轴，没有可用的测量值
- 46: 无法计算坐标的测量值。对于至少一个属于回转的轴，没有可用的测量值

示例: 读取通道轴 X 在第一个通道中的位置。可以编制下列四行中的任何一个（都具有同样的效果）：

```

10 POS = ACS("X", 1, 1, ERRNO)
10 POS = ACS("X", 1, ERRNO, 1)
10 POS = ACS("X", ERRNO, 1, 1)
10 POS = ACS(ERRNO, "X", 1, 1)

```

3. 15 一般系统信息

3. 16 保留命令字

这里列出的关键字必须是单独的，或者用特殊符号对其进行分隔。这将把它们标识为命令字。变量名称的选择不得包含任何保留指令字！

示例：	GOTO 10	跳转至行 10
	GOTO10	如果符号名称（变量）是单独的，则会引起错误消息“运行时错误 2167 = 丢失”，这是因为需要对变量“GOTO10”赋值。

关键字：

A:	ABS ACOS ACS AND APOS ASC ASIN ATAN ATAN2 AXADR\$ AXINF AXO AXP	B:	BCD BIN BITIF	C:	CALL CASE CHR\$ CLOCK CLOSE CLRWARN COF COS CPROBE	D:	DATE DBLOAD DBMOVE DBSAVE DBSEA DBSEAX DBTAB DBTABX DBTABXL DCT DIM DIRCR DIRDEL DIRINF DO DPC
E:	ELSE END ENDIF ENDCASE EOF ERASE ERRNO	F:	假 FILEACCESS FILECOPY FILEDATE FILENO FILEPOS FILESIZE FOR FXC FXCR FXDEL FXINS	G:	GETERR GOTO	I:	IF INP # INSTR INT

数控编程基础知识

L:	LABEL LEN LJUST	M:	MCA MCASE \$ MCODS MCOPS MCS MID \$ MMC	N:	NCF NEXT NJUST 否 NUL	O:	OF OPENR OPENW OTHERWISE 或
P:	PCS PCSPROBE PDIM PLC PMT PMV PPOS PRN # PROBE	R:	REM REPEAT REWRITE ROUND	S:	SCL SCS SCSL SCS3L SD SDLOAD SDR SDSAVE SEEK SETERR SETWARN SIN SPOS SQRT STEP STR\$	T:	TAN TCV THEN 时间 TO TRIM\$ 真
U:	UNTIL	V:	VAL VARINF VERSINF \$	W:	WAIT WCS WHILE	X:	XOR XTAB XTABCR
Z:	ZOCDEL ZOCINS ZOT ZOTCR ZOV						

表格 3-24: 关键词

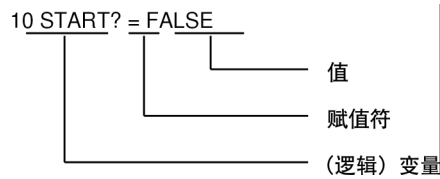
3.17 CPL 命令

3.17.1 赋值

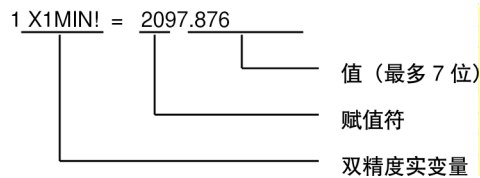
分配

局部以及全局变量可与值相关联。这要通过使用 “=”（等号）进行。

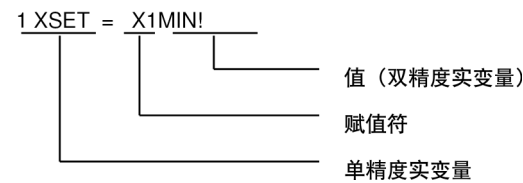
示例：布尔变量的赋值



实数变量的赋值



变量间的赋值



赋值的变量必须位于赋值符号的左边，所对应的值位于右侧。这种声明方法必须小心使用，尤其是在把某个变量的值赋予另一个变量的情况下。

NUL

如果没有对变量赋值，则该变量的值为 NUL。因此，语句 <variable>= NUL 为真。这表明在表示比较或者条件运算的表达式中也可以使用等号。
要删除特定的局部或者全局变量，则对其赋予 NUL 值。永久变量无法被删除，只能被改写。

示例：*删除变量*

```
1 XSOLL = NUL
2 IF XSOLL = NUL THEN
3     PRN#(0,"Variable not assigned.")
4 ENDIF
```



对于没有指定值的变量或通过指定 NUL 删除值，不能将值分配给其他变量！但是，向 NUL 查询变量始终是可能的。

3. 17. 2 数学运算
简单功能

除了以常量表达式（数字）的形式或者作为变量进行赋值以外，还可以把 CPL 表达式的值赋予某个变量。CPL 表达式可以包含使用常量和变量的功能。

基本的四则运算是最简单的功能：

- 加“+”
- 减“-”
- 乘“*”
- 除“/”

数控编程基础知识

一般来说，“乘除运算优先于加减运算”，即，乘除运算优先执行，然后才进行加减运算。也可以将圆括号与简单表达式（不包含功能调用）一起使用，其嵌套深度最多为 7 个。

示例:

```
1 I% = 25: XACTUAL = 10

2 XCOMMAND = 150 / (100-I%) + XCOMMAND 的值为 12。
XACTUAL
```

对变量，常量或 CPL 表达式有影响的算术功能仍然可以被调用。这些功能必须放在紧跟在相应指令字后面的括号中。功能总是指输入值的内部数字表示方法。在程序执行的过程中，可以使用“程序检查”来对此表示方法进行校验。在采用嵌套表达式的情况下，尤其其中包含功能调用时，则必须考虑可能的最大嵌套深度。这取决于圆括号内的表达式在执行期间所要求的内存空间。

ABS

指定输入值的绝对值，即负值变成正值，而正值仍然为正值。

示例:

```
1 I% = -125

2 XVALUE = 2*SQRT (ABS(100+I%)) XWERT 的值为 10
```

EXP

计算指数功能如果没有指定<Basis>（可选参数），则关于任意编程基础或默认基准 e (e = 2.718281828 ...)。

语法:

EXP(<Exponent> [, <Basis>])

表格 3-25: EXP 语法

示例:

```
1 A! = EXP(2) : REM calculates e^2 and assigns the result to A!
2 B! = EXP(3.2) : REM calculates 2^3 and assigns the result to B!
```

INT

INT 通过去除小数点后的位数（四舍五入），把输入值（实数）转变为整数（INTEGER）。。输入值可以是常量或者变量。

示例:

```
1 XVALUE% = INT (10.9) XWERT 的值为 10
```

RANDOM

该功能生成一个随机正整数。默认值介于 0 和 32767 之间。
通过可选参数 <MinValue> 和 <MaxValue> 可指定随机数的值范围。
它必须适用于:

<MaxValue> - <MinValue> <= 32767

语法:

RANDOM ([[<MinValue>] [, <MaxValue>]])	
<MinValue>	可选: 随机数的最小值（正数）

<MaxValue>	可选： 随机数的最大值（正数）
ERRNO	CPL 变量 如果将 CPL 变量 ERRNO 编入参数列表中的任何位置，则命令不产生任何内部运行时错误。该错误由变量的相应返回值返回。如果出现错误，则生成随机数 0。 以下为可能的返回值： 0：访问正常 -1：参数错误 如果未输入 CPL 变量 ERRNO，则出现参数错误时，会产生运行时错误。

表格 3-26: RANDOM 语法

示例：

1 REM 随机数 0 或 1
2 I%=RANDOM (0,1)
3 REM 随机数介于 10 和 100 之间
4 I%=RANDOM (10,100)

ROUND

ROUND 通过四舍五入将输入值转换为整数（INTEGER）。输入值可以是实数表达式。

示例：

1 XVALUE% = ROUND (10.9) XWERT 的值为 11
2 XWERT% = ROUND (5.5) XWERT 的值为 6
3 XVALUE% = ROUND (5.49) XVALUE 的值为 5

SQRT

形成输入值的平方根。输入值不得为负值；这一点未定义。

示例：

1 I% = 44
2 XSET = 4*SQRT (100+I%) XSOLL 的值为 48

SIN、COS、TAN、ASIN、ACOS、ATAN

如果三角函数按常规度处理角度，将这些角度标记为双精度实变量很有用。
三角功能 SIN，COS，TAN 具有作为输入变量的角度，并提供数值作为结果。因此，ASIN，ACOS，ATAN 反功能具有数值作为输入变量和作为结果的角度（以度为单位）。
将角度声明为双精度实变量是有用的。

可以使用以下的三角功能：

•	SIN	正弦功能
•	COS	余弦功能
•	TAN	正切功能
•	ASIN	反正弦功能
•	ACOS	反余弦功能
•	ATAN	反正切功能

数控编程基础知识

示例：			
	1	Angle! = 30	
	2	ERG! = SIN (ANGLE!)	ERG! 的值为 0.5
	3	Angle! = ASIN (ERG!)	ANGLE! 的值为 30
	4	Angle! = ATAN (1/1)	ANGLE 的值为 45 度
	5	ANGLE! = ATAN (1/0)	由于除以零而未定义
	6	ANGLE! = ATAN (0/-1)	ANGLE! 的值为 0 度
	7	Angle! = ATAN (-1/-1)	ANGLE 的值为 45 度

ATAN2

在正确的象限中形成反正切，即结果在 ±180 度的范围内。为了实现这一点，必须用两个参数调用 ATAN2。单圆弧切线（ATAN）不提供这种可能性。其结果总是在 ±90 度的范围内。

ATAN 和 ATAN2 之间存在以下关系：
 $\alpha 1 = \text{atan}(\text{相反腿} / \text{相邻腿}) = \text{atan}(y / x)$
与 $-90 < \alpha 1 < 90$
 $\alpha 2 = \text{atan2}(\text{相对腿}, \text{相邻腿}) = \text{atan2}(y, x)$, $-180 < \alpha 2 < 180$

示例：			
	ANGLE = ATAN2 (1, 1)	ANGLE 的值为 45 度	
	ANGLE = ATAN2 (1, 0)	ANGLE 的值为 90 度	
	Angle = ATAN2 (0, -1)	ANGLE 的值为 180 度	
	ANGLE = ATAN2 (-1, -1)	ANGLE 的值为 -135 度	

3. 17. 3 逻辑运算
一般信息

逻辑操作是二进制的逻辑变量和十进制整数变量。如下图所示，逻辑运算可以用常用的运算符来表达，如 “.” 和 “+” 符号（**但不在 CPL 中**）。这里同样也是“乘除运算优先于加减运算”，即 AND 运算优先于 OR 运算。圆括号的可嵌套深度为 7。

NOT、AND、OR、XOR

CPL 提供四种运算功能类型：

- NOT 功能 NOT
- AND 功能 AND
- OR 功能 OR
- EXCLUSIVE OR 功能 XOR

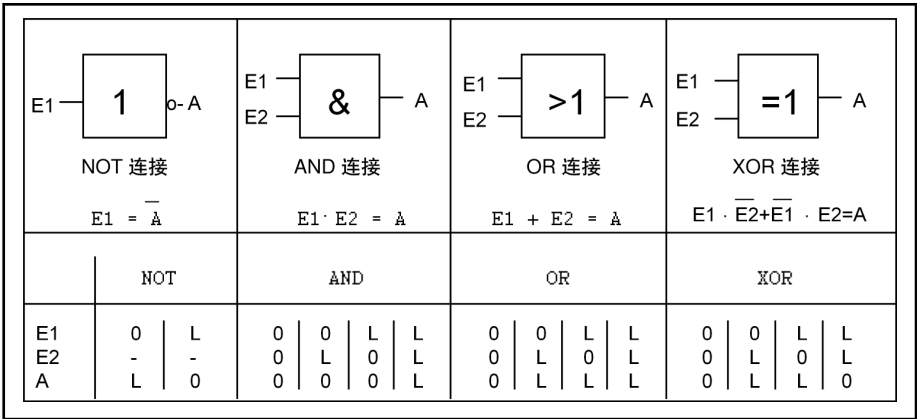


插图 3-1: 逻辑运算
逻辑运算符可用于位掩码。

示例: 位 0 是否设定在 @20?
...
20 IF @20 AND 1 <> 0 THEN GOTO 。设置
30 ELSE GOTO 。UNSET ENDIF
...

3. 17. 4 数制间的转换

BCD

BCD 格式的二进制值的转换。

语法: <BCD value>=BCD(<binary value>)

表格 3-27: BCD 语法

示例: 1 BCD_VALUE = BCD (49) BCD_VALUE 的值为 73

BIN

BCD 编码数字的二进制值的转换。

语法: <binary value>=BIN(<BCD value>)

表格 3-28: BIN 语法

示例: 1 BIN_VALUE = BCD (49) BIN_VALUE 的值为 31

3. 17. 5 关系运算" =、>=、>、<>、<=、<"

允许使用以下关系运算符：

- = 等于
- >= 大于或等于
- > 大于
- <> 不等于
- <= 小于或等于
- < 小于

关系运算用来描述关系（例如用于命令 REPEAT - UNTIL, WHILE - DO - END, IF - THEN - ELSE - ENDIF）的条件（“满足”或者“不满足”）。

3. 18 更多 CPL 的基本元素

3. 18. 1 常量
一般信息

如果为了程序执行而声明数值，并且使其保持不变（常量），则可将此类值作为数值表达式输入命令中。

整数常量（INT）

书写整数时不带小数点。

示例：

NUMBER% = 4

整数常量（CPL 型 INT）

浮点常量（双精度）

使用小数点（浮点）来标识实数数字（小数或者分数）。对于小于 1 的浮点数，零点必须在该点的前面编程，例如 0,134。

示例：

PI = 3.141593

浮点常量（CPL 型实数）

浮点常量总是精确的两倍，即用 15 位数字显示。

字符串常量

字符串常量必须放在引号（"）内。

示例：

EXAMPLE \$ = "This is a character string"

字符串常量

3. 18. 2 代码字符

CPL 使用以下代码字符：

! ? , " [] < - / &
@ % \$: () = > + *

逗号通常用作分隔符。它在字符串内仅用作语法标点符号。在小数中，句号被用作小数点，而对于跳转目标则用作标签。在字符串内，句号只作为语法标点符号来解释。

3. 19 数控块同步命令

3. 19. 1 概览

数控块同步命令可用于：

- 使数控程序与特定事件同步。
- 使块准备和程序执行同步；
- 将预读限制在一定数量的块中。
- 使不同通道中的数控程序相互同步。
- 在块准备时间或在块执行有效时间，使同步功能生效。

同步功能块准备：

- WAIT（无参数）：
 - CPL 功能 WAIT (<idle time>)：
- 暂停块准备，直到之前的所有块都处理完毕。
块准备按指定的时间暂停。

- CPL 功能 WAIT (BITIF(...))：块准备等待 PLC 数控 位接口中的特定信号。
- BlkNmb：将预读限制在编程数量的块中。

同步功能活动时间点：

• WAITA / WAITO：	等待 PLC-数控 位接口的特定状态。
• WPV / WPVE：	等待永久 CPL 变量的值。
• SPV / SPVE：	写入永久 CPL 变量。
• ASTOPA / ASTOPO：	它停止运动，直到达到某些轴位置。
• BSTOPA / BSTOPO：	它停止运动，直到达到基本工件坐标系 (BCS) 中特定的轴位置。
• WSTOPA / WSTOPO：	它停止运动，直到达到当前工件坐标系 (WCS) 中特定的轴位置。
• OFFSTOPA / OFFSTOPO：	取消控制通道中的停止条件。

3. 19. 2 块准备的同步功能

WAIT（无参数）

描述：“WAIT” 功能 **停止块准备**，直到 WAIT 块之前的所有编程块都处理完毕。如果随后要在程序中访问当前机床数据或者基于进程的数据，则这样做是绝对必要的。

按照时间顺序，块准备 – 在单个程序行被分析和解释的阶段 – 总是在机器上执行之前执行。准备和执行之间的时间间隔不是恒定的，而是取决于许多参数（进给，行进距离，预测等）。

因此，如果要使程序对机床或者与进程相关的实际状态（例如当前实际位置、位接口处的信号等）作出反应，则必须使用 WAIT 命令，以确保在评价的时候，上述时间段恰好等于“0”。

示例： WAIT（无参数）

```
N10 X0
N100 (MSG, still running)
N20 X150
30 WAIT                                块执行暂停
40 XPOS = MCS (1)-150
50 IF XPOS < 0.0001 THEN                “抵达位置”输出为 X=150
    (MSG, 已到达位置)
70 ENDIF
```

 在标准数控块（DIN）和 CPL 块中都可以编程“WAIT”功能（无参数）。在包含 WAIT 指令的 CPL 块中，不得编程 a“：”。后续的 CPL 命令必须写入新的 CPL 块中。

CPL 功能：WAIT (,<idle time>)

描述： 停止块准备，直到已编程的等待时间经过。

数控编程基础知识

语法:	WAIT (, <空转时间> [, <ResVar>])	
	其中	
	<Idle time>	等待时间以毫秒为单位，没有小数位。也可将等待时间编程为整数型算术表达式。
	<Event var>	可选整数型变量 等待时间结束后，<ResVar> 被赋值为 1。

表格 3-29: WAIT 语法

示例:	WAIT 及等待时间	
	10 WAIT(, 块处理暂停 1000 毫秒。后续，整数值 "1" 被赋予 E% 变量。 1000, E%)	
	10 WAIT 块处理停止时间取决于整数型变量 TIME% 的内容。无返回值。 (, TIME%)	

特殊功能和限制: 该功能在预运行时被跳过。

CPL 功能: WAIT (BITIF(...))

描述:	停止块准备，直到 PLC 数控 位接口处出现特定状态。
	也可以编程时间期限。在这种情况下，块准备暂停，直到位接口处出现特定的状态，或者该时间期限结束。

语法:	WAIT(<BITIF condition>[, [<timeout>] [, <ResVar>]])	
	其中	
	<BITIF condition>	指定要在 PLC 数控 位接口处检查的条件。这要求使用以下语法: [NOT (] BITF (<Parameter>)] [= <State>]
		<Parameter> 传递 BITIF 功能的参数（有关 BITIF 功能的描述，参见 第 7 章 "CPL 命令" 第 371 页 ）。
		<State> 布尔表达式与 BITIF 功能的结果进行比较。 如果未编制 <State>，则比较结果为 TRUE。
		如果条件满足，块准备继续。
	<Timeout>	等待时间以毫秒为单位，没有小数位。 块准备一直等到 BITIF 条件满足或超时时间经过。
	<Event var>	可选整数变量：返回发生等待事件的时间： 0：呼叫时已经满足等待条件。 1：等待时间已经过，但未满足等待条件 2：在等待时间内符合等待条件

表格 3-30: WAIT 语法

示例:	WAIT (BITIF(...))

10 WAIT (BITIF(1,1,1)=TRUE)	等到第一轴的第二轴相关输入信号设定。
11 WAIT (BITIF(2,0,2)= (E1? OR E2?))	等到第二通道的基于第三通道的输入信号具有逻辑表达式值 (E1? OR E2?)。
12 WAIT (NOT BITIF(3,2,1),,C%)	等到第一主轴的基于第四主轴的输入信号为 FALSE。
20 IF C%=0 THEN	
21 PRN#(0,"BED. OK")	如果在调用 WAIT 时条件已经满足，则变量 C% 提供值"0"。
22 ENDIF	
30 WAIT (BITIF(4,4,1)=E7?,250,ERG%)	等到第一轴的基于第五轴的输入信号设定。等到轴所假定的值为变量 E7?，或者直到 250 毫秒后。
40 IF ERG%=0 THEN	
41 PRN#(0,"Not waited")	如果在调用 WAIT 时条件已经满足，则变量 ERG%提供值"0"。
42 ENDIF	
50 IF ERG%=1 THEN	
51 PRN#(0,">250ms waited")	如果在等待期间条件未满足，变量 ERG% 返回值"1"。
52 ENDIF	

特殊功能和限制：该功能在预运行时被跳过。

BlkNmb (BNB)

描述：使用标准数控函数 "BlkNmb"，可以对块准备功能中块的最大数量进行限制。例如，"BlkNmb"功能可以用来对零件程序中运行时测量结果的进一步处理加以控制。

语法：	BlkNmb<Num>	Restriction ON
	BlkNmb (0)	允许的最大块数取决于参数/ 数控 0 / LookAh / Ch / NofBlkPrep (MP 7060 00110) 。
	缩写形式：BNB... 其中	
	<Disp>	所需块的最大数量 整数 输入范围：大于或等于 0。 0：块的数量取决于 MP 7060 00110。

表格 3-31: BlkNmb (BNB) 句法

- 特殊功能和限制：
- 如果调用该函数时多个<No>块已准备好，则块准备会暂停，直到准备好的块的数量<No>下降为止。
 - 如果在<No>中输入的块数多于控制组态中计划的块数，则块的有效数量将自动减少到 MP 7060 00110 中的设置。

3. 19. 3 块执行时的同步功能

一般信息

IndraMotion MTX 中，每个通道可以运行一个程序。如果各个处理段被分成各个不同的程序，且这些程序在不同的通道上运行，则所有程序的处理顺序可通过块执行时的同步功能来进行控制。

带位事件（REV, SEV, WEV 和 WREV）的程序段执行时间的同步功能用 **DIN 参数列表**编程。



位事件具有系统范围的有效等级（0 或 1），可以修改或使用数控函数 REV (ResetEvent), SEV (SetEvent), WEV (WaitForEvent) 和 WREV (WaitForAndResetEvent)。

块执行时间的所有同步功能都拥有 **CPL 参数列表**（类似于子程序）。其中，CPL 表达式列表必须用方括号括起来：“[” 和 ”]”。

一般来说，以下内容适用：

- 系统范围的有效位事件用于 REV, SEV, WEV 和 WREV。工程师必须确保没有交叉。
- 功能“WPV”, “WPVE”, “SPV”, “SPVE”中使用的永久 CPL 变量在整个系统中均有效。因此，工程师必须确保不正确的使用不会导致意外的交叉。
- 在提供的数控函数框架中，仅允许以下的单一型永久 CPL 变量：
 - INT
 - BOOL
 - 实数
 - DOUBLE

对于数组，只能逐个元素寻址！



功能 WAITA、WAITO、WPV、WPVE 在块结束处具有隐含的下坡效应。错误地设置同步点可能会导致机器损坏。

在实际处理之前，必须对可能会发生同步问题的程序序列进行检查。

等待一些事件：WEV

描述:	使用“WEV”功能时，它会在运行时等待（或阻止连续块切换），直到发生某个位事件。	
语法:	WEV (<EventNo> {, EventNo ...})	
	其中:	
	<EventNo>	事件位的编号。有效 1 - 96。 任何数量的<EventNr>都可以在一个 WEV 中编程。

表格 3-32: WEV 语法

WEV 内的事件是分离的（或）连接的。几个 WEV 可以在一个程序段中编程。所有分开编程的 WEV 都是连接的（和）连接的。一个事件的几个编程不会产生错误。

如果一个事件已经处于待决状态，则简单地跳过 WEV。如果用户等待（边缘反应），则必须先使用 REV 删除该事件。这可以在同一个块中进行。

示例:	N10 X10 WEV (10,12)	首先到达 X10, 然后, 程序等待事件 10 或事件 12 发生。如果事件 10 或事件 12 在前往 X10 或之前已经发生, 则不会在块的末尾等待。(仍然会产生下降斜坡)。
	N10 WEV (10,12) WEV (16, 18) WEV (20)	当事件 (10 或 12) 和 (16 或 18) 和 20 存在时, 块完成。

WEV / WREV / SEV / REV

例如, 通过 “WEV”, “SEV” 和 “REV” 功能, 借助于位事件在运行时使两个通道同步。
如果在一个程序段中编程了多个功能, 则它们按以下顺序生效:

- 1. REV
- 2. SEV
- 3. WEV / WREV

对于块 N10 REV 10) WEV (10) , 因此可以等待事件 10 的边沿。
通道 1 运行以下序列:

通道 1 程序

;Channel 1	
N10 REV(20, 21)	Deletes events 20 and 21
N20 G1 F1000 X100 Y100	Approaches position 1
N30 WEV(20)	
;Waits till event 20 is present. It is synchronized at runtime with channel 2 N30..	
N30 X200 Y200	Approaches position 2
N40 SEV(21)	Triggers event 21, channel 2 may continue

通道 2 运行以下序列:

通道 2 程序

;Channel 2	
N20 G1 F1000 Z100 W100	Approaches position 1
N30 SEV(20)	Triggers event 21, channel 1 may continue
N30 Z200 W200	Approaches position 2
N40 WEV(21)	
;Waits till channel 1 triggers event 21. It is synchronized at runtime with channel 1 N40..	

等待位事件删除: WEV

描述: 使用 “WREV” 功能时, 它会在运行时等待 (或阻止连续块切换), 直到发生某个位事件。如果满足这个条件, 所有等待的事件都被删除。

语法:	WREV (<EventNo> {, EventNo ...})	
	其中:	
	<EventNo>	事件位的编号。有效 1 - 96。 任何数量的<EventNr>都可以在一个 WREV 中编程。

表格 3-33: WREV 语法

关于等待条件, WREV 类似于 WEV (参见 第 4 章 “等待一些事件: WEV” 第 82 页)。满足条件时, 所有在等待条件下编程的位事件都将被删除。等待条件可以由 WEV 和 WREV 组合。



在等待条件满足和删除比特事件的同时，某些通道可能会执行一些块。

示例：通道 1 中编程了 N10 WREV (11)。块 N20 SEV (11) 和 N20 WEV (11) 进入通道。如果执行了 SEV，则 WREV 等待条件在通道 1 中完成并继续。比特事件 11 被删除前经过一段时间。如果 WEV 之前在通道 2 中执行过，则通道 2 不会等待（因为位事件 11 仍处于活动状态）。如果首先执行删除，则通道 2 等待。

在这些情况下，建议通过两个不同的位事件对同步进行编程。

示例：

N10 X10 WREV (10, 12)	首先到达 X10，然后，程序等待事件 10 或事件 12 发生。如果事件 10 或事件 12 在前往 X10 或之前已经发生，则不会在块的末尾等待。（仍然会产生下降斜坡）。然后，位事件 10 和 12 被删除。
N10 WREV (10, 12) WEV (16, 18) WREV (20)	当事件（10 或 12）和（16 或 18）和 20 存在时，块完成。然后，位事件 10, 12 和 20 被删除。

触发位事件：SEV

描述：“SEV”功能在运行时触发（设置）位事件。因此，编程事件的级别设置为 1（不管值是 0 还是 1）。几个 SEV 可以在一个程序段中编程。

语法:	SEV (<EventNo> {, <EventNo> ...})
	其中:
<EventNo>	事件位的编号 有效 1 - 96 任何数量的<EventNr>都可以在一个 SEV 中编程。

表格 3-34: SEV 语法

示例:	N10 SEV (13) 事件 13 已设置。
	N10 SEV (15, 16) 事件 15 和 16 已设置。
	N10 SEV (15) SEV (16) 事件 15 和 16 已设置。

删除位事件：REV

描述：“REV”功能在运行时删除位事件。几个 REV 可以在一个程序段中编程。

语法:	REV (<EventNo> {, <EventNo> ...})
	其中:
<EventNo>	事件位的编号 有效 1 - 96 任何数量的<EventNr>都可以在一个 REV 中编程。 特殊情况：REV (ALL) 或 REV () 删除所有事件位。

表格 3-35: REV 语法

示例:	N10 REV (13) 事件 13 已删除。
	N10 REV (15, 16) 事件 15 和 16 已删除。
	N10 REV (15) REV (16) 事件 15 和 16 已删除。
	N10 REV (全部) 所有事件都被删除。

等待 PLC 数控 位接口处的状态：WAITA / WAITO

- 描述：

功能“WAITA / WAITO ”在运行时启动等待进程，直到最多 16 个接口信号中的一个或多个达到指定值为止。

根据运行状况，可对几个接口信号编制以下内容：

 - **WAITA:**
各个信号的“AND 运算”
等到**所有**接口信号达到指定值。
 - **WAITO:**
各个信号的“或运算”
等到**至少一个**接口信号达到指定值。

语法：

等待 全部 指定信号： WAITA[BITIF(<Parameter>){=<State>}，BITIF(<Parameter>){=<State>}，...，{<Timeout>}]	
等待 一个 指定信号： WAITO[BITIF(<Parameter>){=<State>}，BITIF(<Parameter>){=<State>}，...，{<Timeout>}]	
其中：	
<BITIF>	查询 PLC-数控 接口的 BITIF 功能。 可选择同时查询 2 到 16 个接口信号。
<Parameter>	传送 BITIF 功能的参数 (有关说明，请参阅 BITIF 功能，第 7 章“CPL 命令”第 371 页)
<State>	布尔表达式与 BITIF 功能的结果进行比较。 如果未编制 <State>，则比较结果为 TRUE。 如果条件满足，块准备继续。
<Timeout>	可选时间以 ms 为单位；默认值 = 0。 如果在满足相关条件之前，<Timeout> 已超时，则会生成警告，并且系统将继续等待。 如果未编制 timeout 或者等于 0，则不会产生任何警告。

表格 3-36: BlkNmb (BNB) 句法

- 示例：
- 通道 1 中的程序 1 用于加工车削零件的终止面。

通道 2 中的程序 2 将要在该端面上切削一个槽，而且必须**等待**程序 1 把该车削零件让渡给程序 2。程序 2 通过**设置**特定的接口信号来启用。如果界面信号达到状态，通道 1 把启用状态传送给 2。而程序 2 进行加工，程序 1 等待程序 2，以便继续进行加工任务。

N10 WAITO [BITIF (10,1,1)=FALSE, BITIF (11,1,2)]	主动等待，直到 BITIF(10,1,1) 的值为 0，且 BITIF(11,1,2) 的值为 1。
N10 WAITA [BITIF (10,1,1)=FALSE, BITIF (11,1,2)]	主动等到 BITIF (10,1,1) 的值为 0，且 BITIF (11,1,2) 的值为 1。

以下适用于 **WAITA, WAITO:**

如果 WAITA 和 WAITO 编入同一个数控块，则在两个条件都得到满足之前，块执行将停止。首先评价 WAITO 条件。



功能“WAITA”，“WAITO”，“WPV”，“WPVE”在块结束处具有隐含的下坡效应。

错误地设置同步点可能会导致机器损坏。

在实际处理之前，必须对可能会发生同步问题的程序序列进行检查。

特殊功能和限制： 该功能在预运行时被跳过。

等待永久 CPL 变量值：WPV / WPVE

描述： 功能“WPV / WPVE”在运行时启动等待进程，直到永久 CPL 变量达到特定的比较值。

可在不同的时间确定比较值：

- **WPV:**
比较值是在运行时期间、数控块恰好被激活的时刻所计算出的 CPL 表达式。数控块激活后的比较值变化无效。将比较值与永久变量的值进行比较。
由于在运行时进行评价，所以只允许简单的 CPL 表达式。
- **WPVE:**
比较值是早在准备时就已确定的 CPL 表达式，但在运行时之前并不与永久变量值进行比较。

语法：

在 运行时 确定比较值： WPV[<Perm. CPL variable><ComparisonOperator><Simple CPL Expression>{,<Timeout>}]													
比较值在 准备时 就已确定： WPVE[<Perm. CPL Variable><ComparisonOperator><CPL expression>{,<Timeout>}]													
其中：													
<Perm. CPL Variable>	永久变量，用“@”符号标识，后接变量名称。 如果这些是数组的元素，则只允许一个常量或永久整型变量（CPL 类型 INT）作为 WPV 的索引。对于 WPVE，其指数可以是常量或者任何 CPL 整数型变量。												
<Relational operator>	以下为可能的比较运算符： <table><tr><td>=</td><td>永久 CPL 变量等于 CPL 表达式的值。仅建议用于整数或者布尔值。</td></tr><tr><td>< ></td><td>永久 CPL 变量不等于 CPL 表达式的值。仅建议用于整数或者布尔值。</td></tr><tr><td><</td><td>永久 CPL 变量小于 CPL 表达式的值。</td></tr><tr><td>≤</td><td>永久 CPL 变量小于或等于 CPL 表达式的值。</td></tr><tr><td>></td><td>永久 CPL 变量大于 CPL 表达式的值。</td></tr><tr><td>≥</td><td>永久 CPL 变量大于或等于 CPL 表达式的值。</td></tr></table>	=	永久 CPL 变量 等于 CPL 表达式的值。仅建议用于整数或者布尔值。	< >	永久 CPL 变量 不等于 CPL 表达式的值。仅建议用于整数或者布尔值。	<	永久 CPL 变量 小于 CPL 表达式的值。	≤	永久 CPL 变量 小于或等于 CPL 表达式的值。	>	永久 CPL 变量 大于 CPL 表达式的值。	≥	永久 CPL 变量 大于或等于 CPL 表达式的值。
=	永久 CPL 变量 等于 CPL 表达式的值。仅建议用于整数或者布尔值。												
< >	永久 CPL 变量 不等于 CPL 表达式的值。仅建议用于整数或者布尔值。												
<	永久 CPL 变量 小于 CPL 表达式的值。												
≤	永久 CPL 变量 小于或等于 CPL 表达式的值。												
>	永久 CPL 变量 大于 CPL 表达式的值。												
≥	永久 CPL 变量 大于或等于 CPL 表达式的值。												

<simple CPL expression>	为避免影响运动生成，运行时只对简单的 CPL 表达式进行评价。 一个简单的 CPL 表达式就是一个数学条件，它由 CPL 中允许的永久 CPL 变量、常量以及数学运算组成。 允许的数据类型： • INT • BOOL • 实数 • DOUBLE
<CPL expression>	任何数学条件都必须采用 CPL 编程语言。
<Timeout>	可选时间以 ms 为单位；默认值 = 0。 如果在满足相关条件之前，<Timeout> 已超时，则会生成警告，并且系统将继续等待。 如果未编制 timeout 或者等于 0，则不会产生任何警告。

表格 3-37: WPV / WPVE 句法

示例:	N10 WPV[@9=10]	程序在激活点等待，直到变永久变量 @9 的值达到 10。
	N10 WPVE[@8=(5*#VAR2%)]	在准备时对条件 “5 * #VAR2%” 进行评价。将由此确定的值与运行时的永久变量 @8 进行比较。只要 @8 与确定的值不符，就不会激活新的数控块。

WPV / SPV

例如，通过 “WPV” 和 “SPV” 功能，借助于永久 CPL 变量，可在运行时使两个通道同步。

通道 1 运行以下序列：

程序

```
;Channel 1
N20 G1 F1000 X100 Y100      travel to position 1
N30 WPV(@20=5)              wait till @20=5, here synchronizing at runtime with channel 2
                             N30..
N30 X200 Y200                travel to position 2
N40 SPV(@20=6)               set @20=6, now channel 2 can continue running
```

通道 2 运行以下序列：

程序

```
;Channel 2
N20 G1 F1000 Z100 W100      travel to position 1
N30 SPV(@20=5)               set @20=5, now channel 1 can continue running
N30 Z200 W200                travel to position 2
N40 WPV(@20=6)               wait till @20=6, here synchronizing at runtime with channel 1
                             N40..
```

特殊功能和限制： 该功能在预运行时被跳过。

编写永久 CPL 变量：SPV / SPVE

描述： 使用功能“SPV / SPVE”，可在运行时通过写入的方式将值赋予永久 CPL 变量。

可在不同的时间确定该值:

- SPV:**
被赋予永久变量的这个值只能在运行时确定。
由于在运行时进行评价, 所以只允许简单的 CPL 表达式。
- SPVE:**
被赋予永久变量的这个值在准备时 (CPL 解释时间) 确定, 但只能在运行时赋值给永久 CPL 变量。

语法:	在 运行时 确定被赋予的值: SPV[<Perm. CPL Variable> =<Simple CPL Expression>]	
	在 准备时 确定被赋予的值: SPVE[<Perm. CPL Variable> =<CPL Expression>]	
	其中:	
	<Perm. CPL Variable>	永久变量, 用 “@” 符号标识, 后接变量名称。
	<simple CPL expression>	为避免影响运动生成, 运行时只对简单的 CPL 表达式进行评价。 一个简单的 CPL 表达式就是一个数学条件, 它由 CPL 中允许的永久 CPL 变量、常量以及数学运算组成。 允许的数据类型: - INT - BOOL - 实数 - DOUBLE
	<CPL expression>	任何数学条件都必须采用 CPL 编程语言。

表格 3-38: SPV / SPVE 语法

示例:	N10 SPV[@6=1]	在运行时, 将值 1 赋予永久变量 “@6”。
	N10 SPV[@5=(7*(@PERMVAR1% +5))]	在运行时, 确定条件 (7*(@PERMVAR1% + 5)) 的值, 然后将其赋予 @5。
	N10 SPVE[@5=(7*#VAR1%)]	在准备时, 确定条件 (7 * #VAR1%) 的值, 然后在运行时将其赋予 @5。

运动停止，直到到达某个轴位置：ASTOPA / ASTOP0

- 描述:** 采用 “ASTOPA / ASTOP0” 功能, 可使通道间的运动同步。可根据一个通道中的**轴坐标系统** ACS 的一根或者多根轴的**位置**, 停止或恢复另一通道的同步运动 (特殊情况除外)。
- 限制:**
- 用于同步的轴必须属于除要控制通道之外的通道。否则可能会发生自锁保持。
 - 要控制的通道必须处于“自动”或者“MDI”模式。
 - 如果同时为通道指定了 AND 和 OR 条件, 则在两个功能中的至少一个功能的相应条件得到满足时, 通道将停止。

特殊情况：ASTOP 停止调用通道

- 如果待监控轴未分配给该通道，ASTOPA / ASTOPO 可停止已编程通道。轴必须使用系统轴名称或编号进行编程。

可为每个控制通道指定停止通道的一个或多个条件:

- ASTOPA:**
只要**所有条件**都得到满足，则要控制通道的同步运动即会停止（逻辑 AND 运算）。
- ASTOPO:**
只要**至少一个条件**得到满足，则要控制通道的同步运动即会停止（逻辑 OR 运算）。



如果指定新的 AND 和 OR 条件，那么该通道中所有之前的条件都将无效。

语法:

AND 条件: ASTOPA [<ChannelNumber> , <Cond. 1> { , <Cond. 2> } { .. { , <Cond. 8> } . }]		
OR 条件: ASTOPO [<ChannelNumber> , <Cond. 1> { , <Cond. 2> } { .. { , <Cond. 8> } . }]		
其中:		
<Channel number>	要控制的通道号 (1..n)。整数值或者整数型变量。	
<Bed. 1, Bed. 2... Bed. 8>	1 至 8 条件规范如下: <Axis><Comparison operator><Comparison value> 其中:	
	<Axis>	系统或通道轴名称，作为 CPL 字符串常量或者 CPL 串变量编程。
	<Relational operator>	允许运算符: < , ≤ , > , 。
	<Comparison value>	实数值或者实数 CPL 表达式。 在准备时确定该值，并保持其模态有效。

表格 3-39: 语法



通过 AND / OR 条件可以从一个通道中停止最多 4 个其他通道。

示例:

轴名称和编号的使用:

10 AXISNO% = 2	定义:
20 AXISNAME\$ = "X"	• 轴号
30 STOPCHAN% = 2	• 轴名称
:	• 通道数量
N40 ASTOPO[STOPCHAN%, AXISNO%<10]	
:	
N90 ASTOPO[STOPCHAN%, "Z">20.3]	
:	
N150 ASTOPO[STOPCHAN%, AXISNAME\$<1.5]	

运动停止，直到到达某个基本工件位置：BSTOPA / BSTOP0

描述：采用“BSTOPA / BSTOP0”功能，可使通道间的运动同步。可根据一个通道**基本工件坐标系 BCS** 的一根或多根轴的**位置**，确定另一通道的同步运动是停止还是继续。

限制：

- 用于同步的坐标必须属于除要控制通道之外的通道。否则可能会发生自锁保持。
- 要控制的通道必须处于“自动”或者“MDI”模式。
- 如果同时为通道指定了 AND 和 OR 条件，则在两个功能中的至少一个功能的相应条件得到满足时，通道将停止。

可为每个控制通道指定停止通道的一个或多个条件：

- **BSTOPA:**
只要**所有条件**都得到满足，则要控制通道的同步运动即会停止（逻辑 AND 运算）。
- **BSTOP0:**
只要**至少一个条件**得到满足，则要控制通道的同步运动即会停止（逻辑 OR 运算）。

 如果指定新的 AND 和 OR 条件，那么该通道中所有之前的条件都将无效。

语法：

AND 条件： BSTOPA[<ChannelNumber> ,<Cond. 1> {,<Cond. 2> }{...,<Cond. 8> }.]	
OR 条件： BSTOP0[<ChannelNumber> ,<Cond. 1> {,<Cond. 2> }{...,<Cond. 8> }.]	
其中：	
<Channel number>	要控制的通道号 (1..n) 整数值或变量
<Bed. 1, Bed. 2... Bed. 8>	1 至 8 条件规范如下： <Coordinate><Comparison operator><Comparison value> 其中：
	<Coordinate> 对于 BCS： 坐标名称或者坐标索引，作为 CPL 字符串常量或者 CPL 串变量编程。
	<Relational operator> 允许运算符： < , ≤, > , 。
	<Comparison value> 实数值或者实数 CPL 表达式。在准备时确定该值，并保持其模态有效。

表格 3-40: BSTOPA / BSTOP0 语法

 通过 AND / OR 条件可以从一个通道中停止最多 4 个其他通道。

示例： **激活基本工件坐标的 AND 条件。**

N10 BSTOPA[3,"z"<12.0,"x">15]

只要以下条件适用于控制通道，则通道 3 即会停止：

- 基本工件坐标 z 的位置小于 12mm，并且
- 基本工件坐标 x 的位置大于 15mm

运动停止，直到到达工件位置：WSTOPA / WSTOPO

描述： 采用 “WSTOPA / WSTOPO” 功能，可使通道间的运动同步。可根据一个通道**工件坐标系 WCS** 的一根或多根轴的**位置**，确定另一通道的同步运动是停止还是继续。

- 限制：*
- 用于同步的坐标必须属于除要控制通道之外的通道。否则可能会发生自锁保持。
 - 要控制的通道必须处于“自动”或者“MDI”模式。
 - 如果同时为通道指定了 AND 和 OR 条件，则在两个功能中的至少一个功能的相应条件得到满足时，通道将停止。

可为每个控制通道指定停止通道的一个或多个条件：

- **WSTOPA:**
只要**所有条件**都得到满足，则要控制通道的同步运动即会停止（逻辑 AND 运算）。
- **WSTOPO:**
只要**至少一个条件**得到满足，则要控制通道的同步运动即会停止（逻辑 OR 运算）。



如果指定新的 AND 和 OR 条件，那么该通道中所有之前的条件都将无效。

语法：

AND 条件： WSTOPA[<channel number>,<cond.1>{,<cond.2>}{...{,<cond.8>}.}]	
OR 条件： WSTOPO[<channel number>,<cond.1>{,<cond.2>}{...{,<cond.8>}.}]	
其中：	
<Channel number>	要控制的通道号 (1..n) 整数值或变量

数控编程基础知识

<Bed. 1, Bed. 2... Bed. 8>	1 至 8 条件规范如下： <Coordinate><Comparison operator><Comparison value>	
	其中：	
	<Coordinate>	对于 WCS： 坐标名称或者坐标索引，作为 CPL 字符串常量或者 CPL 串变量编程。
	<Relational operator>	允许运算符： < , ≤, > , 。
	<Comparison value>	实数值或者实数 CPL 表达式 在准备时确定该值，并保持其模态有效。

表格 3-41: WSTOPA / WSTOPO 语法

 通过 AND / OR 条件可以从一个通道中停止最多 4 个其他通道。

示例：**激活工件坐标的 AND 条件：**

```
N10 WSTOPA[3,"Z"<12.0,"X">15]
```

只要以下条件适用于控制通道，则通道 3 即会停止：
通道工件坐标 Z 的位置小于 12 mm，并且工件坐标 X 的位置大于 15 mm

取消停止条件：OFFSTOPA / OFFSTOPO

描述：清除某个控制通道的停止条件。
语法：

OFFSTOPA	清除所有的 AND 停止条件
OFFSTOPO	清除所有的 OR 停止条件

表格 3-42: OFFSTOPA / OFFSTOPO 句法
对于同步功能，以下内容适用：

- ASTOP0、BSTOP0、WSTOP0、OFFSTOP0
 在模态上有效并相互抵消。
- ASTOPA、BSTOPA、WSTOPA、OFFSTOPA
 在模态上有效并相互抵消。

通过程序结束（M30）取消停止条件：

在零件程序的终点（M30），在进行通道复位或者程序重新选择时，该通道中已激活的停止条件将自动取消。

3. 19. 4 执行时写入系统数据

描述 使用功能“SSD / SSDE”，可在运行时通过写入的方式将值赋给系统数据。
 可在不同的时间确定该值：

- SSD:
 赋予系统数据的值在运行时确定。
 由于在运行时进行评价，所以只允许简单的 CPL 表达式。
- SSDE:
 被赋予系统数据的这个值在准备时（CPL 解释时间）确定，但只能在运行时赋值给永久 CPL 变量。除简单数据类型外，还可赋予字符串。

通常建议使用 SSDE。仅在执行时仅须计算被赋值的 CPL 表达式的情况下，使用 SSD 命令。



始终在准备时计算系统数据的目标地址。这意味着如果以域指数的方式指定系统数据，则不应从其他通道改变这些指数。

例子

系统数据中的指数

```
N10 SSDE[SD.Field[@6].Int = 4]
```

如果未在本通道中设置变量 @6，除了通道同步之外，还应在 N10 之前编入 WAIT。

语法:

在 运行时 确定被赋予的值： SSD[<SystemData> = <Simple CPL Expression>]	
在 准备时 确定被赋予的值： SSDE[<SystemData> = <Simple CPL Expression>]	
其中：	
<System date>	系统数据，以 "SD." 标识，后接变量名。
<simple CPL expression>	为避免影响运动生成，运行时只对简单的 CPL 表达式进行评价。 一个简单的 CPL 表达式就是一个数学条件，它由 CPL 中允许的永久 CPL 变量、常量以及数学运算组成。 允许的数据类型： - INT - BOOL - 实数 - DOUBLE
<CPL expression>	任何数学条件都必须采用 CPL 编程语言。 允许的数据类型： - INT - BOOL - 实数 - DOUBLE - CHAR 数组

表格 3-43: SSD / SSDE 句法

示例:

```
N10 SSDE[SD.Testvar=1]
```

在运行时为系统数据"SD.Testvar"分配值 1。

```
N10 SSDE[SD.Array[4].Pos=7.2*(SIN(ANG%)+5)]
```

表达式"(7.2*(SIN(ANG%) +5))" 的值被赋予"SD.Array[4].Pos"。

```
N10 SSDE[SD.Array[J%].Pos=4]
```

运行时将值 4 赋予 "SD.Array[J%].Pos"。

数控编程基础知识

N20 SPV[SD.MeasPos= 3.5*@MPOS! + 12.2]

表达式“3.5*@MPOS! + 12.2”在执行时确定，然后赋予
“SD.MeasPos”。

特殊功能和限制：

- 由于对系统数据的访问，SSD 和 SSDE 的链接时间比类似命令的要长。由此，建议谨慎使用这些命令，以免不必要地延长链接时间（以及“后链接”的情况下）。
- 建议使用 SSDE。仅在执行时仅须计算被赋值的 CPL 表达式的情况下，使用 SSD 命令。

4 模态功能组概述

多个 MTX NC 功能已配置在模态 NC 功能组中。同一个模块组中的不同 NC 功能可以相互取消。所以，任何时候只有一个模块组中的一个函数适用。

模态组	NC 功能
几何	G0、G1、G2、G3、G5、G6、G33
速度控制	G8、G9、CFD、LNU、LND、SNU、SND、S2U、S2D
等级	G16、G17、G18、G19、G20
绝对 / 增量编程	G90、G91
进给编程	G93、G94、G95
主轴速度编程	G96、G97
英制 / 公制	G70、G71
刀具长度修正	G47、G48
2. 5D 刀具径向修正	G40、G41、G42
3D 刀具半径修正	G140、G141、G142
D 修正	D0、D
ED - 修正	ED0、ED
2. 5D 刀具径向修正	G40、G41、G42
2. 5D 刀具径向修正的外角	G43、G44
2. 5D 刀具径向修正的进给	G45、G46
2. 5D 刀具径向修正的碰撞监控	CLN, 关闭**
进给模式下的插补	G61、G62
到位窗口激活	IPS1、IPS2、IPS3
库 1 的零点偏置	G52、G53.1、G54、G55、G56、G57、G58、G59
库 2 的零点偏置	G52.2、G53.2、G54.2 G55.2、G56.2、G57.2、G58.2、G59.2
库 3 的零点偏置	G52.3、G53.3、G54.3 G55.3、G56.3、G57.3、G58.3、G59.3
库 4 的零点偏置	G52.4、G53.4、G54.4 G55.4、G56.4、G57.4、G58.4、G59.4
库 5 的零点偏置	G52.5、G53.5、G54.5 G55.5、G56.5、G57.5、G58.5、G59.5
库 1 的定位倾面	G151.1、G152.1、G153.1、G154.1 G155.1、G156.1、G157.1、G158.1、G159.1
库 2 的定位倾面	G151.2、G152.2、G153.2、G154.2 G155.2、G156.2、G157.2、G158.2、G159.2
库 3 的定位倾面	G151.3、G152.3、G153.3、G154.3 G155.3、G156.3、G157.3、G158.3、G159.3
库 4 的定位倾面	G151.4、G152.4、G153.4、G154.4 G155.4、G156.4、G157.4、G158.4、G159.4
库 5 的定位倾面	G151.5、G152.5、G153.5、G154.5 G155.5、G156.5、G157.5、G158.5、G159.5
定位工件位置修正	BCR, 关闭**
坐标偏置	TRS, 关闭**
附加坐标偏置	ATR, 关闭**

模态功能组概述

模态组	NC 功能
输入刀具“轮廓偏置”	SHT, 关闭**
输入帮助: 缩放	SCL, 关闭**
输入刀具“镜像”	MIR, 关闭**
输入刀具“旋转”	ROT, 关闭**
输入刀具“PoI”	PLS, 关闭**
输入刀具“编程耦合”	PLP、EPC, 关闭**
倒圆和倒角	RND、RNE、CHF、CHL, 关闭**
样条曲线倒圆	SCO, 关闭**
用样条曲线取代直线	LTS, 关闭**
倍率	OVD、OVE
直径编程	DIA、RAD
修正赋值	G78、G79
极坐标的极点	POP, 关闭**
轴速度	AVE, 关闭**
轴加速度	AAC, 关闭**
路径加速度	PAC, 关闭**
Kv 编程	KVP, 关闭**
前馈	FFW, 关闭**
循环轴的定位类型	PMD, 关闭**
编制一个削减扭距	RDT. 关闭**
进给适应	FAD, 关闭**
冲压	PUN、NIB, 关闭**
切向刀具引导	TTL, 关闭**
切向刀具定向	TTO, 关闭*
区域监控	ARA, 关闭**
拆分	分割, 关闭**
轴耦合	AXC, 关闭**
选择性坐标耦合	SCC、SCCT, 关闭**
补偿表	GCT, 关闭**
激光	LEP, 关闭**
切向过渡	DTT, 关闭**
精度编程	PRP, 关闭**
模态子程序	G80、G81、G82、G83、 ...
异步 G0 - 插值	CG0 (ASY1)、CG0 (ASY0)

关闭**

利用 (0) 或 () 编程 NC 函数的关闭功能, 例如, AAC (0)

表格 4-1:

模块 NC 功能组概述

5 数控函数，语法符合 DIN 66025（包括扩展）



不希望的运动！

许多数控函数需要对轴和/或坐标地址进行编程。通常，只有各自的通道轴地址，即同步轴可与坐标地址分开编程。

编程参数列表中异步轴的地址通常会导致语法错误，而在参数列表之外进行编程会导致异步遍历运动！

例外情况：

- G74(Home) VA1:
异步轴 VA 运动到参考点。
- GAX(VA):
异步轴 VA 应用于通道
- FsMove、FsTorque、FsReset:
“移动到固定停止”功能也会影响异步轴。

5.1 概览

该控件提供多个数控函数。除了 DIN 66025 中规定的命令之外，它还包含 G 代码和其他高级语言相似语法元素的重要补充。

控制器的**数控函数**包括：

- **1 位和 2 位 G 代码：**
主要基于 DIN 66025，为其“G 代码词汇”提供适当的补充
 - 在 DIN 中扩展功能组
(例如 G52、G53、G54、...)，或者
 - 引入与现有功能组相关的新功能组。一位数的 G 代码总是可以写成两位数字，即前导零（例如 G0=G00，G1=G01，...）。
- **3 位 G 代码（例外）：**
如果 2D 区域中的现有功能在 3D 区域中以相同的方式应用（例如 G41、G42→G141、G142），则使用该功能。
 - **G 代码与“.”扩展：**
例如它们与直线移动以及零偏置或坐标转换一起使用。这些功能最多可有 5 组使用“.”扩展进行编址 / 编程。
示例：
G1、2 进料中的直线点对点移动
库 2 的 G59.2 G59 零点偏置
- **具有定义的数控函数的 M 功能**
例如 M0、M30
- **使用高级语言语法的 NC 功能：**
(参见 第 6 章 “使用高级语言语法的 NC 功能” 第 181 页)



有关所有数控函数的表格概述，请参阅 第 8 章 “附录” 第 535 页。

另外，所有的数控函数都在索引中。

使用的符号： 在本文档中，以下符号用于数控函数的语法：

数控函数，语法符合 DIN 66025（包括扩展）

字体 "Courier bold" 或 "Courier"：

这种字体中的字符串必须如图所示进行编程。

示例：G0 (POL)

角度括号 < >

指示要编程的表达式/参数的占位符。占位符用斜体表示。

示例：<Axis1>

大括号 { }

表示一个可选的表达式/参数。

这样的语法元素可以但不必被编程。

示例：G0 { ({ POL, } { <Par1> }) }

字符 "|"

分离不能同时使用的可能参数（替代参数）。

示例：G0 { ({ POL, } { NIPS | IPS1 | IPS2 | IPS3 }) }

测量单位：

除非另有规定，数控编程基于以下测量单位：

- 平移位置：
[mm] 用于 G71, [inch] 用于 G70
- 旋转位置：
[度数] 用于回转轴
- 平移速度：
[mm/min] 用于 G71, [inch/min] 用于 G70
- 旋转速度：
[度 / 分钟]
- 平移加速度：
G71 为 [m/s²], G70 为 [1000 inch/s²]
- 旋转加速度：
[1000 度 / s²]

5.2 G 代码

5.2.1 快速移动中的直线插补"G0"

描述 以最大路径速度（=快速移动）在直线上插补编程的位置。

以下情况适用：

- 至少有一个轴以最大速度或加速度运行。控制轴的速度，使所有轴同时到达目标点。
- 如果该功能处于激活状态，“快速移动激活”通道接口（=接口）信号将被输出。
- 该功能为模态方式，删除 G1、G2、G3、G5、G6、G12、G13、G33。
- 块末尾的距离（目标和实际值之间的位移）可以减小到某些目标窗口的大小（=“准确停止功能”）。

数控函数，语法符合 DIN 66025（包括扩展）

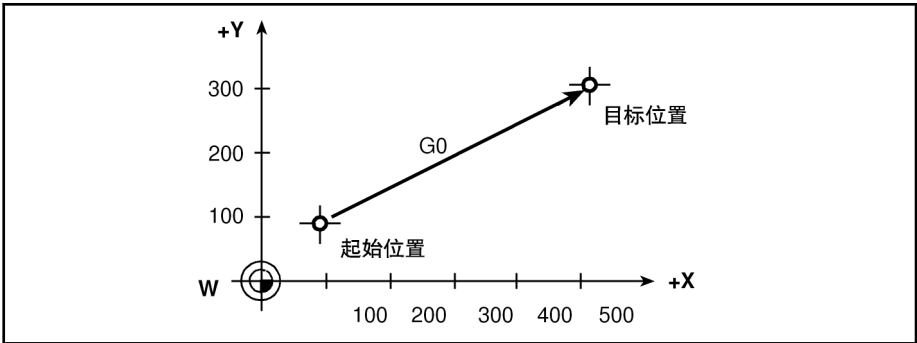


插图 5-1: 快速移动的线性插值

语法：基本功能

G0	快速移动，精确停止 ON 并在笛卡尔坐标系中编程终点位置。考虑到上次为 快速移动精确停止窗口大小 设置的确切停止窗口（默认=精细）。在块终点，控制器首先把路径速度降低到 0。只有在涉及到所有轴的准确停止窗口时，才会遍历下一个程序段。

表格 5-1: 语法 G0

可选参数

G0 ({NIPS IPS1 IPS2 IPS3, } {POL})	
其中	
NIPS	准停定位关闭 除非另有要求，否则在程序段结束时不减速至 v=0。
IPS1	与没有参数的 G0 相同，但是具有 准确停止窗口大小，精细 。
IPS2	与没有参数的 G0 相同，但是具有 准确停止窗口大小，粗糙 。
IPS3	与没有参数的 G0 相同，但是具有 准确停止窗口，无限 。这意味着没有检查定位窗口，而是在块的末尾减速到 v=0。
POL	激活 极坐标编程 并将极角 1 设置为 0 并将极角 2 设置为 90 度。有关极坐标编程，请参阅 第 6.85 章 “极坐标编程” 第 287 页 。

表格 5-2: 可选参数 G0

示例：

```
:
N40 X100 Y100          初始位置
N50 G0 X500 Y300      以快速进给接近目标位置
```

特殊功能和限制：

- 该功能可以在同一个程序段中使用或不使用轴地址进行编程。
- 使用最大轴速度的机器参数确定相应的速度。
- 使用通道接口信号“减少的快速进给”（qCh_RedRap），可以将快速移动速度限制在机床参数中设置的减小值。
- 通过通道接口信号“测试快速移动”（qCh_TestRap）控制的功能“测试快速移动”也可以限制快速移动速度。
- 速度可以根据机器参数进行修改
/CHAN/Ch[]/Path/RapTravOvrdr/EnableRapTravIOvrdr”启用快速进给倍率”（7030 00014）或使用进给或快速移动电位器。
- 使用 G0（ON/OFF，精确停止窗口的大小）进行的精确停止更改仅适用于快速移动。

数控函数，语法符合 DIN 66025（包括扩展）

- 要一般更改精确停止窗口的大小，即快速移动和所有进给功能，请使用数控函数 IPS1、IPS2、IPS3。



有关准确停止功能的更多信息，请参阅本文档中的 G1、G61/62、IPS1、IPS2、IPS3 以及文档 “IndraMotion MTX 功能描述” 章节“精度”，“准确停止”章节。

5.2.2 进给中的线性插补"G1"

描述：在有效进给的情况下以直线插补编程位置（可通过 F—地址编程）。运动的协调方式是所有涉及的轴都同时到达编程的终点。

以下情况适用：

- 编程的进给值（F）用作路径进给。因此，当移动多个轴时，每个轴因此小于 F。
- 该功能为模态方式，删除 G0、G2、G3、G5、G6、G12、G13、G33。

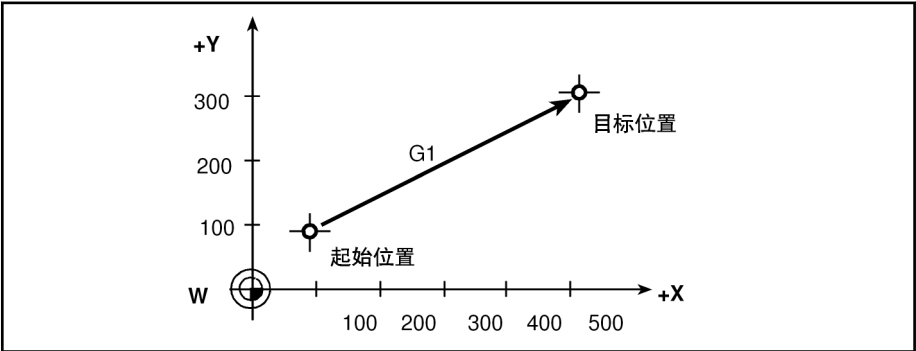


插图 5-2: Feed 中的线性插值

语法：基本功能

G1	根据主动进给运动并以笛卡尔坐标编程结束位置。确切的停止取决于 G61（精确停止 ON）或 G62（精确停止 OFF）是否激活。考虑到上次为进给设置的准确停止窗口大小（默认=精细）。

表格 5-3: 语法 G1

可选参数

G1 ({IPS IPS1 IPS2 IPS3,} {POL}) 其中	
IPS	准确停止 ON，与 G61/G62 无关。 考虑到上次为进给模式设置的准确停止窗口（默认=精细）。在块终止处，控制器将路径速度降低到 v=0。只有在涉及的所有轴都达到此定位窗口时，才会进给移动下一块。此准确停止的激活仅适用于 G1。所有其他进给功能仍然取决于 G61/G62。
IPS1	作为 IPS，但具有准确停止窗口大小，精细。 除了快速移动（G0）外，精确停止窗口大小的定义也适用于所有其他馈送功能，而仅在 G1 上启用精确停止功能。
IPS2	由于 IPS，但准确停止窗口大小，粗糙。 除了快速移动（G0）外，精确停止窗口大小的这一定义也适用于所有其他的进给功能，而仅在 G1 上启用精确停止功能。

数控函数，语法符合 DIN 66025（包括扩展）

IPS3	作为 IPS，但具有 准确停止窗口大小，无限 。 这意味着没有检查定位窗口，而是在块的末尾减速到 v=0。除了快速移动（G0）外，精确停止窗口大小的这一定义也适用于所有其他的进给功能，而仅在 G1 上启用精确停止功能。
POL	激活 极坐标编程 并将极角 1 设置为 0 并将极角 2 设置为 90 度。有关极坐标编程，请参阅 第 6.85 章“极坐标编程”第 287 页。

表格 5-4: 可选参数 G1

示例: 基本功能:

```
:  
N40 X100 Y100           初始位置  
N50 G0 X500 Y300 F100  用 F100 的进给逼近目标位置。
```

特殊功能和限制:

- 该功能可以在同一个程序段中使用或不使用轴地址进行编程。
- 如果没有馈送信号处于激活状态，则必须使用 F—地址在同一行中编程馈送信号。
此提要在被新的 Feed 值覆盖之前保持有效。
- 利用机床参数可以限制所编程的路径速度。
- 路径速度可以通过馈电电位器进行修改。
- 使用数控函数 G61（IPS1）、G61（IPS2）和 G61（IPS3）也可以更改所有进给功能的精确停止窗口的大小。
- 要一般更改精确停止窗口的大小，即快速移动和所有进给功能，请使用数控函数 IPS1、IPS2、IPS3。

 有关准确停止功能的更多信息，请参阅本文档中的 G0、G61/62、IPS1、IPS2、IPS3 以及文档“IndraMotion MTX 功能描述”章节“精度”，“准确停止”章节。

5.2.3 圆弧 / 螺旋 / 螺旋 N 插补 "G2、G3"

一般信息

描述 在带有当前进给的圆形路径上插补（通过 F—地址可编程），在当前工作平面内接近编程的目标位置。

- G2: 顺时针旋转的圆形路径
- G3: 逆时针旋转的圆周路径

此外，编程中还可以包含工作平面外的附加轴。它们的运动是线性插值的。

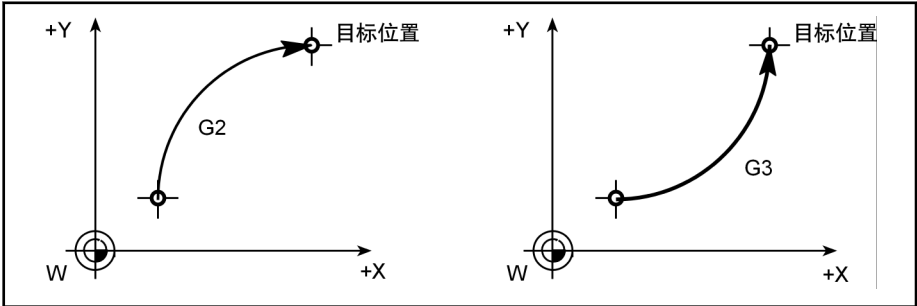


插图 5-3: G2 顺时针旋转的圆形路径; G3 圆弧逆时针旋转

圆弧插补:

仅涉及活动工作平面的轴。

数控函数，语法符合 DIN 66025（包括扩展）

螺旋插补：
除了活动工作平面的轴之外，还涉及**多一个**同步轴。
如果该轴垂直于活动工作平面（例如工件坐标系的其余轴）定位，则会生成具有常量斜率的螺旋路径（螺旋插补）。

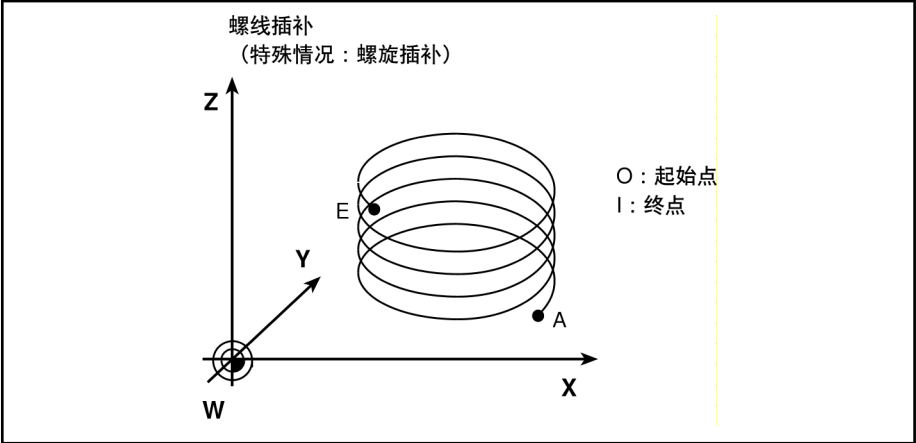


插图 5-4: 螺旋插补

螺旋 N 插补：
编程有效工作平面轴上的**另外一个以上**同步轴（最多 6 个）。

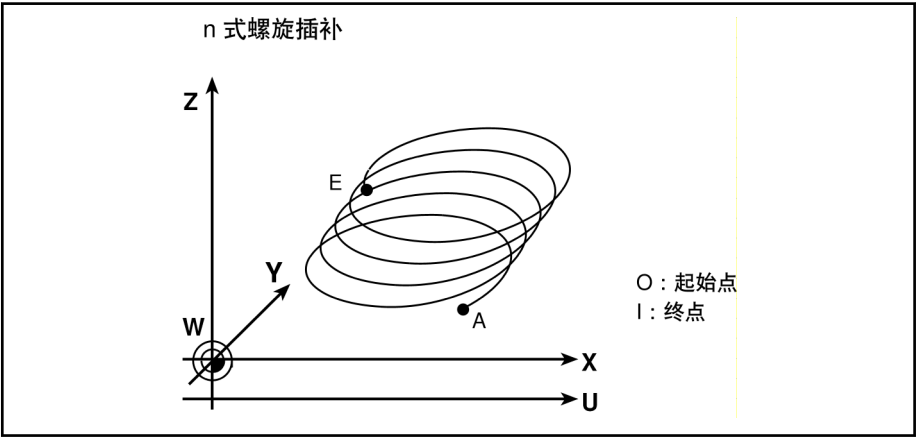


插图 5-5: 螺旋 N 插补

- 以下情况适用：
- 沿圆弧运行的轴由选定的工作平面（G17、G18、G19、G20）明确定义。
 - 编程的进给值（F）作为路径进给，通常指在一个程序段中要运行的所有轴。因此，如果多个轴移动，每个轴都小于 F。
 - 实际的进给量可能受到最大可能的径向加速度的限制。
 - G0、G1、G2、G3、G5、G6、G12、G13、G33 的功能构成模态组，并取消对自己的选择。
 - 圆周路径的编程可以通过以下方式实现：
 - 半径编程或者
 - 中心点编程。

半径编程

- 描述： 使用当前位置作为起点，通过编程设定
- 圆周终点和
 - 圆周半径进行指定圆周运动。

数控函数，语法符合 DIN 66025（包括扩展）

圆周终点可以指定为绝对位置值或增量位置值。

语法：**基本功能：**

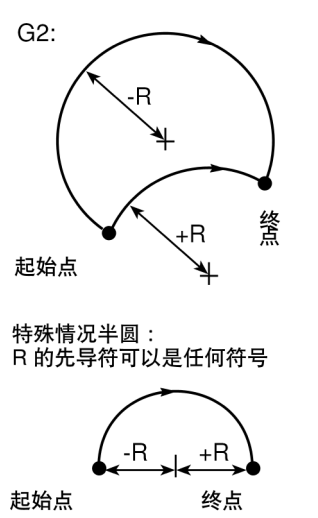
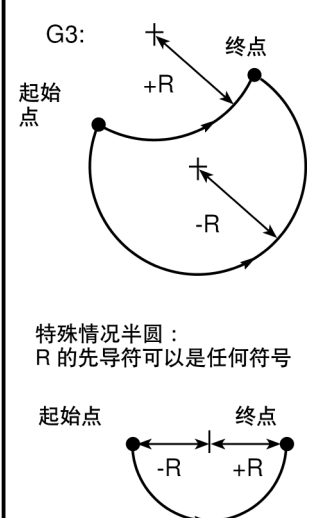
G2<EP> R<value>	顺时针方向的圆周路径
G3<EP> R<value>	逆时针旋转的圆周路径
其中	
<EP>	终点坐标
<Value>	圆周路径的半径 半径必须至少是起点和终点之间距离的一半。 正号 / 负号用于指定在可能的情况下是否可以使用较小或较大的弧来移动路径： 正半径值： 弧度 ≤ 180 度 负半径值： 弧 > 180 度 如果半径恰好等于起点和终点之间距离的一半，则结果就是一个半圆，而且正 / 负号都可以使用。
G2:  特殊情况半圆： R的先导符可以是任何符号 G3:  特殊情况半圆： R的先导符可以是任何符号	

插图 5-6: G2 和 G3 圆周路径的半径

表格 5-5: 语法 G2, 半径编程

- 特殊功能和限制:
- 无法生成整圆。
 - 控制器根据机器参数中设置的公差自动修正不精确的半径输入。否则，会导致运行时错误。

中心点编程

基本功能

- 描述:
- 使用当前位置作为起点，通过编程设定
- 圆周终点和

数控函数，语法符合 DIN 66025（包括扩展）

- 中心点进行指定圆周运动。
- 可以通过渐增或绝对的编程方式编程圆周的中心点。

语法：**基本功能：**

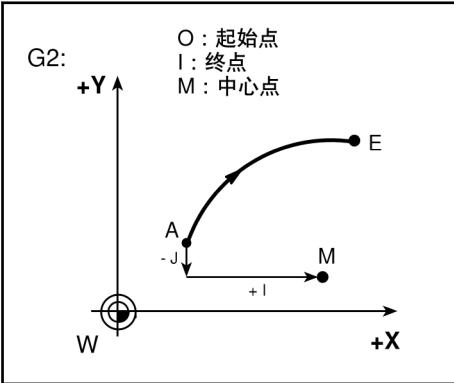
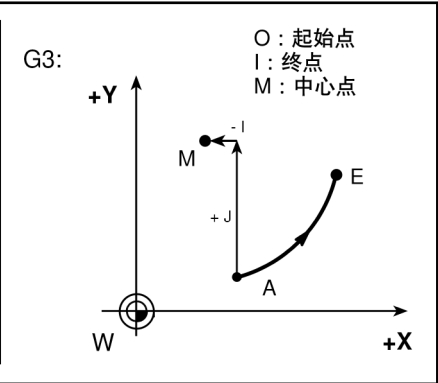
G2<EP> <IP>	顺时针方向的圆周路径
G3<EP> <IP>	逆时针旋转的圆周路径
其中	
<EP>	终点坐标 如果起点和终点在圆平面内相同，则会自动生成一个 整圆 。
<IP>	插值参数 I、J 和 K。 对于局部相对尺寸编程，为每个轴定义圆周起始点 A 和圆周中心点 M 之间的距离。符号是从矢量方向从 A 到 M 的结果。 根据机床参数中坐标的含义来确定各个轴的插补参数。 以下是默认设置： 对于 x 坐标， $I = M(X) - A(X)$ 对于 y 坐标， $J = M(Y) - A(Y)$ 对于 z 坐标， $K = M(Z) - A(Z)$ 使用局部绝对尺寸编程，也可以在绝对坐标下编程圆周中心点。
G2:  O : 起始点 I : 终点 M : 中心点 G3:  O : 起始点 I : 终点 M : 中心点	

插图 5-7: 中心点编程

表格 5-6: 语法 G2，中心点编程

可选参数

语法：	G2/G3 (POL) <EP> R<Value>	用极坐标进行半径编程
	G2/G3 (POL) <EP> <IP>	用极坐标进行中心点编程
其中		
	POL	激活终点坐标的极坐标编程，设定极角 1 为 0，极角 2 为 90 度。有关极坐标编程，请参阅 第 6.85 章 “极坐标编程” 第 287 页 。

表格 5-7: 可选参数 G2、G3

数控函数，语法符合 DIN 66025（包括扩展）

示例： 半径编程（笛卡儿）：

```
：  
N40 G1 X10 Y10 F100          接近初始点  
N50 G2 X38 Y20 R15          圆周插补
```

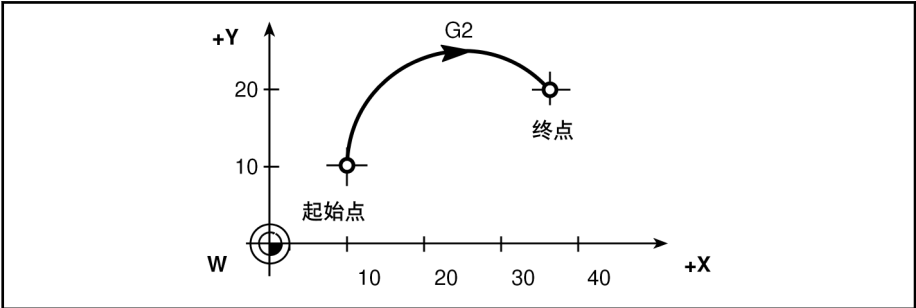


插图 5-8: 半径编程（笛卡儿）

中心点编程（笛卡儿）

```
：  
N80 G1 X100 Y100 F100      接近初始位置  
N90 G90 G17 G3 X350 Y250    x-/-平面内的逆时针圆弧插补。终点坐标  
I200 J-50                  的绝对尺寸编程激活。  
：
```

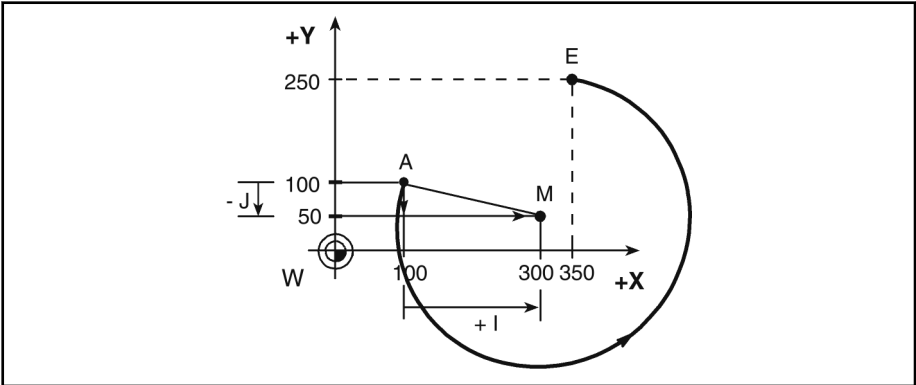


插图 5-9: 中心点编程（笛卡儿）

螺旋编程（螺旋）：

```
10 PITCH = 2                将音调值存储在局部变量中。  
20 转= 10                   将绕组数存储在局部变量中。  
N30 G0 G90 X-10 Y0 Z0      移动到开始位置。  
N50 G2 I10 Z = IC ([ - PITCH]) 指定螺距的单绕组。  
RPT [TURN] F1000
```

特殊功能和限制：

- 如果开始点和结束点相同，则控件会自动生成一个整圆。
- 控制器根据机器参数中设置的公差自动修正不精确的中心点输入。为此，控制器将相应地偏置中心点的位置。
- 如果插补参数以及圆周半径在同一个块中编程，那么只考虑圆周半径（=半径编程）。

数控函数，语法符合 DIN 66025（包括扩展）

- 如果不适用于选定平面的插补参数进行编程，则控制报告运行时错误。
示例：G17 G2 X5 I9 K7（错误：K 与 x / y 平面不匹配）

5.2.4 停留时间"G4"

描述： 停止程序执行。随后编程的程序段仅在编程的停留时间过后才被处理。
停顿时间从数控块之前完成处理开始。
旋转主轴或者进给辅助轴不停止。如果需要，同步轴可以减少滞后。
停留时间可以编程为：

- 秒
- 或 -
- 主轴转速。

为了确定主轴转数，当前的实际转数由主主轴循环建立，计算在此基础上执行的转数。在高度动态的主轴上，实际上等待的编程主轴转速和主轴转速之间存在一定的偏差，因此可能发生在加速或减速阶段。
如果组态的主主轴为模拟主轴（无速度反馈），则用速度指令值代替实际速度值进行计算。

语法：	G4 (F<Value>)	停留时间以秒为单位。
	G4 (S<Value>)	主轴转速停留时间
	其中	
	<Value>	以秒或者主轴转数进行规定 如果输入“0”，G4 块将在内部丢弃。

表格 5-8: 语法 G4

- 特殊功能和限制：**
- 该功能必须在不带路径信息的单独程序段中编程。该块中只允许使用辅助功能和特殊功能。
 - 编程的主轴转数是指机床参数中配置的主主轴或使用“MainSp”功能（请参阅 第 6.72 章 “主轴切换 MainSp、MSP” 第 266 页 ）。

5.2.5 切向圆周出口"G12"

描述： 使用下一个编程的移动运动，控制系统会自动计算 G12 到下一个移动块的切向轮廓过渡。
只有不涉及方向颠倒的过渡才称为“切向”。

 无法连续对多个 G12 块进行编程！

“G0、G1、G2、G3、G5、G6、G12、G13 ”的功能构成模式组，并取消对自己的选择。

注意

在螺旋 / 螺旋 N 插补的情况下，可以在程序段转换中加工标记！

计算的切向过渡只涉及圆平面！空间切线可以在块转换时跳转！

语法：	G12	<EP> [REV]
	其中	

数控函数，语法符合 DIN 66025（包括扩展）

<EP>	圆周终点坐标
REV	可选参数：若已编程，则计算移动方向相反的过渡

表格 5-9: G12 语法

- 特殊功能和限制:
- 在“手动数据输入”模式下不能编程 G12。
 - 带有移动运动的块必须在 G12 之后的同一程序中编程。
 - 在 G12 的下一块中，不得切换当前平面
 - G12 块之后不得跟随另一个 G12 块或 G5 / G13 块

5.2.6 切向圆周入口"G5", "G13"

描述: 使用最新编程的运行运动，控制系统会自动计算 G5 程序段的切线圆入口。只有不涉及方向颠倒的过渡才称为“切向”。



如果连续发生多个 G5 / G13 运动，则第一个入口切线决定所有跟随的 G5 / G13 轮廓元素。

“G0、G1、G2、G3、G5、G6、G12、G13、G33”的功能构成模态组，并取消对自己的选择。

注意

在螺旋 / 螺旋 N 插补的情况下，可以在程序段转换中加工标记！

计算的切向过渡只涉及圆平面！空间切线可以在块转换时跳转！

语法:

G5 G13	<EP> [REV] <EP> [REV]
其中	
<EP>	圆周终点坐标
REV	可选参数：若已编程，则计算移动方向相反的过渡

表格 5-10: 语法 G5、G13

- 特殊功能和限制:
- 在“手动输入”操作模式下不能编程 G5 / G13
 - 带有移动运动的块必须在 G5 / G13 之前的同一程序中编程。
 - 在激活的 G5 / G13 之前或期间，不得直接切换当前平面。

示例:

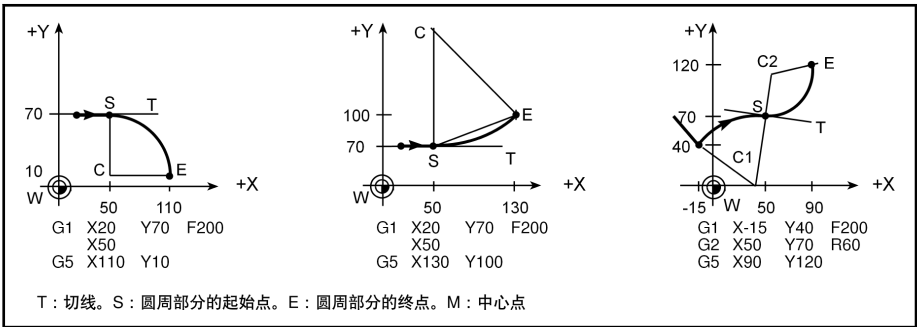


插图 5-10: 切向圆入口的例子

数控函数，语法符合 DIN 66025（包括扩展）

5. 2. 7 样条编程"G6"

概述 – 样条类型

描述: 与线性插值相比，样条插值可提供相同的曲面精度和轮廓精度，尽管其数据点数量较少，因为可在点之间计算常量曲线。

IndraMotion MTX 支持以下样条类型:

- **样条类型 0:**
带有系数编程的样条
(CAD/CAM 系统的多项式系数)
- **样条类型 1:**
具有数据点编程的 C¹ 常量三次样条
(在数据点的切线过渡)
- **样条类型 2:**
具有数据点编程的 C² 常量三次样条
(在数据点处的常量曲率转变)
- **样条类型 3:**
带检查点编程的 B 样条
(靠近数据点的曲线特征)
- **样条类型 4:**
B 样条近似 (压缩机)

使用“SplineDef” (SPD; 参见 第 6.134 章 “样条类型的定义 SplineDef、SPD” 第 342 页) 功能选择并初始化所需的样条类型。之后，使用 G6 函数来激活样条编程。

 有关各个样条类型的更多详细信息，请参阅文档“IndraMotion MTX 功能描述”。

语法:		
	G6	激活“样条”路径类型。

表格 5-11: 语法 G6

根据所需的编程类型，G6 可以针对具有不同模态有效参数的各种样条类型进行编程。

特殊功能和限制: 下列功能**不能**用样条线编程:

- 张量定向
- 2D 路径校正 G41 / G42
- 使用路径分割进行冲压 / 啃
- 倒角和圆角
- 切向刀具引导
- 精度编程
- 样条线后的 G5
- 取消剩余距离

带系数编程的样条（样条类型 0）

坐标 / 轴编程

描述: 每个通道坐标都可以

- 作为样条通过指定多项式系数
- 通过指定终点位置以线性方式运动。

数控函数，语法符合 DIN 66025（包括扩展）

语法：通过指定多项式系数进行样条：

<Coordname>(<c0>,<c1>,...,<cn>)	
	用多项式系数编程各个坐标。

表格 5-12: 运动为样条坐标
- 或 -

通过指定终点位置以线性方式运动：

<Coordname>(<Endpos>)	编程各个坐标 / 轴的终点位置
其中	
<Coordname>	坐标 / 轴的名称
<c0>、<c1>...<cn>	坐标的多项式系数 "n"对应于"SplineDef"中定义的样条度数
<Endpos>	坐标的终点位置

表格 5-13: "线性"坐标运动

示例：SplineDef(3)
G6 X (0.1,1.25,0.5,0.73) Y30 B (0.0,-1.0,0.1,-0.2)

分母多项式编程

语法：

DN(<g ₀ >,<g ₁ >,...,<g _n >)	所有样条坐标的公分母多项式。
其中	
<g ₀ >、<g ₁ >、...、<g _n >	分母多项式的多项式系数。 "n"对应于"SplineDef"中定义的样条度数。

表格 5-14: 分母多项式编程

示例：SplineDef(3)
G6 X (0.1,1.25,0.5,0.73) B (0.0,-1.0,0.1,-0.2) DN (1,0,1)

定向矢量编程

描述：此编程方法需要使用矢量方向（"Coord(..)"）的活动轴转换。

语法：

O1(<o ₁₀ >,<o ₁₁ >,...,<o _{1n} >)	方向矢量的 X 分量
O2(<o ₂₀ >,<o ₂₁ >,...,<o _{2n} >)	方向矢量的 Y 分量
O3(<o ₃₀ >,<o ₃₁ >,...,<o _{3n} >)	方向矢量的 Z 分量
其中	
<o ₁₀ > ,<o ₁₁ > ,...,<o _{1n} >:	样条矢量的 x 分量的样条系数。
<o ₂₀ > ,<o ₂₁ > ,...,<o _{2n} >:	样条向量的 y 分量的样条系数。
<o ₃₀ > ,<o ₃₁ > ,...,<o _{3n} >:	样条矢量 Z 分量的样条系数。 "n"始终对应于"SplineDef"中定义的样条度数。

表格 5-15: 方向矢量编程

数控函数，语法符合 DIN 66025（包括扩展）

示例：程序

```
N00 ;Spline coefficients for vector orientation
001 PI=3.14159:PIH=PI/2:PIHQ=PIH*PIH:PIHC=PIHQ*PIH
N10 G1 F30000 X0 Y0 Z0 B90 C0
N20 SplineDef(3)
N30 Coord(1) ;5-axis transformation with vector orientation on
N40 G6 PL[PIH] ;PL see spline parameter length programming
N50 O1(1,0,-3/PIHQ,2/PIHC) O3(0,1,(3-PI)/PIHQ,(-2+PIH)/PIHC)
N60 O1(0,0,3/PIHQ,-2/PIHC) O3(1,0,(-3+PIH)/PIHQ,(2-PIH)/PIHC)
N70 O(0,1,0) ;normal vector orientation
N80 G1
N90 Coord(0)
```

编程样条参数长度

描述：样条参数长度是为“w”定义的区域长度，“w”在 0 到 w_e 之间激活。在 G6 被取消选定之前，值 w_e 为模态方式作用并对所有数控块保持有效。

PL 必须在 G6 之后的第一个运行程序段中编程：否则，会发生运行时错误。

语法：	<table><tr><td>{ PL<w_e>}</td><td>样条参数长度的可选编程</td></tr><tr><td>其中</td><td></td></tr><tr><td><w_e></td><td>任何大于 0 的值</td></tr></table>	{ PL< w_e >}	样条参数长度的可选编程	其中		< w_e >	任何大于 0 的值
{ PL< w_e >}	样条参数长度的可选编程						
其中							
< w_e >	任何大于 0 的值						

表格 5-16: 编写样条参数长度

示例：

G6 X (0.1,1.25,0.5,0.73) B (0.0,-1.0,0.1,-0.2) PL0.6

($X = 0.1 + 1.25 w + 0.5 w^2 + 0.73 w^3$ 和

$B = 0.0 - 1.0 w + 0.1 w^2 - 0.2 w^3$ ，其中

w 在 0...0.6 之间有效)

Hermite 样条（样条类型 1）

坐标 / 轴编程

描述：可以选择移动每个通道坐标

- 并通过指定位置和一阶导数作为样条
- 或 -
- 通过指定终点位置以线性方式运动。

语法：通过指定位置和一阶导数进行样条：

<Coordname>(<c0>,<c1>)	用位置 ($\Delta c0$) 和一阶导数 ($\Delta c1$) 编程各个坐标。

表格 5-17: 运动为样条坐标

- 或 -

通过指定终点位置以线性方式运动：

<Coordname>(<Endpos>)	编程各个坐标 / 轴的终点位置。

表格 5-18: “线性”坐标运动

其中	
<Coordname>	坐标 / 轴的名称
<c0>,<c1>	坐标的多项式系数， 用 $c0\Delta$ 位置和 $c1\Delta$ 一次导出
<Endpos>	坐标的终点位置

表格 5-19: 协调议案

数控函数，语法符合 DIN 66025（包括扩展）

示例: SplineDef (1143)
G6 X (0.1,1.25) Y30 B (0.0,-1.0)

编程样条参数长度

描述: 样条参数长度是为“w”定义的区间长度，“w”在 0 到 w_e 之间激活。值 w_e 在编程块内局部有效。作为默认值，采用 w_e=1。
在样条序列的开始点（即 G6 之后的第一个遍历块）中，假定导数等于零。

语法:	{ PL<w _e >}	样条参数长度的可选编程
	其中	
	<w _e >:	任何大于 0 的值

表格 5-20: 编写样条参数长度

示例: G6 X (0.1,1.25) B (0.0,-1.0) PL0.6
(w 的范围是 0...0.6)

用户定义的速度参数文件

描述: 使用 NC 功能 CVP 和 CPI，可利用 Hermite 样条指定用户定义的速度参数文件。实际 MTX 速度控制已暂停。对于对应的参数设置，坐标的 <c1> 值与到达 <c0> 位置的轴的速度（单位为 mm/s 或 inch/s）对应。根据机床参数 NC0/VelScaleFact/Ch[]/FactVel（F 值），指定 F1（缩放系数设为 mm/s 或 inch/s）或 F60（缩放系数设为 mm/min 或 inch/min）。

语法:	CVP CPI F60 SplineDef (1143) G6	使用使用定义的速度参数文件启用 Hermite 样条
	在这种情况下，通过样条参数长度 PL 编程的数字值构成块的时长（单位：秒）。	

表格 5-21: 用户定义的速度参数文件

示例: 程序

```
N00010 G0 X150 Y0 Z25
N00020 SplineDef(1143)
N00030 CPI CVP F60
N00040 X(150.0000,0.0000) Y(0.0000,0.0000) Z(24.9167,-24.9999) PL0.01
N00050 X(150.0000,0.0000) Y(0.0000,0.0000) Z(24.5003,-50.0001) PL0.01
N00060 X(150.0000,0.0000) Y(0.0000,0.0000) Z(0.5000,-50.0000) PL0.48
N00070 X(150.0000,0.0000) Y(0.0000,0.0000) Z(0.0833,-25.0001) PL0.01
N00080 X(150.0000,0.0000) Y(0.0000,0.0000) Z(0.0000,0.0001) PL0.01
N00090 X(150.0000,0.00002524) Y(0.0000,-0.00006279) Z(0.0180,9.399888) PL0.06
N00100 ...
...
N50080 X(250.0000,0.0000) Y(0.0000,0.0000) Z(70.0171,-0.00232267) PL 0.018257
N50090 G1 G8 CPI(0)
```

C1-和 C2-常系数三次样条（样条类型 1 和 2）

坐标编程

描述: 编程涉及通道坐标的端点。
“SplineDef”中列出的所有 <Members> 都沿着样条移动。“SplineDef”中未定义的其余坐标沿着一条直线移动。

语法:	<Coordname><Endpos>	编程个别坐标及其值。
	其中	
	<Coordname>	坐标名称（包括方向 0）。
	<Endpos>	坐标的终点位置

表格 5-22: 通道坐标的终点编程

数控函数，语法符合 DIN 66025（包括扩展）

示例：坐标 x、y、z 和方向坐标 phi、theta。
SplineDef (2203,x,y,z,phi,theta)
G6 x10 y20 phi20 theta30
x15 y22 O(1,1,2)

示例：
轴 X、Y、U
SplineDef (1213,X,Y)
G6 X10 Y10 U20 (X 和 Y 作为样条移动，U 直线移动)

样条参数长度

描述：样条参数长度由数控使用指定的数据点计算。样条 ID 中定义的过程（参数设置）用于此目的。如果需要，样条参数长度也可以编程：

语法：	{ PL<w _e >}	如果要覆盖 <Parameterization> 的选择，请参阅样条参数长度的可选编程（请参阅 第 6.134 章“样条类型的定义 SplineDef、SPD” 第 342 页，参数 <Id>）。
	其中	
	<w _e >	任何大于 0 的值

表格 5-23: 样条参数长度的可选编程

开始和结束边界条件

请参阅 第 章“开始和结束边界条件” 第 114 页。

B-样条（NURBS）（样条类型 3）

坐标编程

描述：通道坐标（检查点）的终点被编程。“SplineDef”中列出的所有 <Members> 都沿着样条移动。“SplineDef”中未定义的其余坐标沿着一条直线移动。

语法：	<Coordname><Endpos>	编程个别检查点（坐标）及其值。
	其中	
	<Coordname>	坐标名称（包括方向 0）。
	<Endpos>	坐标的终点位置

表格 5-24: 编程检查点（坐标 / 轴）

示例：坐标 x、y、z 和方向坐标
样条 (3103,x,y,z,O)
G6 x10 y20 z30 O (0.1,0,1.0)
轴 X、Y、U

SplineDef (3102,X,Y)G6 X10 Y10 (X 和 Y 作为样条移动，U 直线移动)
U20

样条参数长度

描述：样条参数长度由数控使用指定的检查点自动计算。样条 ID 中定义的过程（参数设置= 1, 2 或 3）用于此目的。如果需要，样条参数长度也可以编程：

数控函数，语法符合 DIN 66025（包括扩展）

语法:	<table><tr><td>{ PL<w_e>}</td><td>如果要覆盖 <Parameterization> 的选择，请参阅样条参数长度的可选编程（请参阅 第 6.134 章 “样条类型的定义 SplineDef、SPD” 第 342 页，参数 <Id>）。</td></tr><tr><td>其中</td><td></td></tr><tr><td><w_e></td><td>任何大于 0 的值</td></tr></table>	{ PL<w _e >}	如果要覆盖 <Parameterization> 的选择，请参阅样条参数长度的可选编程（请参阅 第 6.134 章 “样条类型的定义 SplineDef、SPD” 第 342 页，参数 <Id>）。	其中		<w _e >	任何大于 0 的值
{ PL<w _e >}	如果要覆盖 <Parameterization> 的选择，请参阅样条参数长度的可选编程（请参阅 第 6.134 章 “样条类型的定义 SplineDef、SPD” 第 342 页，参数 <Id>）。						
其中							
<w _e >	任何大于 0 的值						

表格 5-25: 样条参数长度的可选编程

带有 B-样条的检查点的样条点权重

语法:	<table><tr><td>{ PW<w>}</td><td>点权重的可选编程。 这可以用来修改检查点附近的样条线。</td></tr><tr><td>其中</td><td></td></tr><tr><td><w></td><td>默认: 1 0 < w < 1: 将样条线从检查点推开 w > 1: 将样条拉到检查点</td></tr></table>	{ PW<w>}	点权重的可选编程。 这可以用来修改检查点附近的样条线。	其中		<w>	默认: 1 0 < w < 1: 将样条线从检查点推开 w > 1: 将样条拉到检查点
{ PW<w>}	点权重的可选编程。 这可以用来修改检查点附近的样条线。						
其中							
<w>	默认: 1 0 < w < 1: 将样条线从检查点推开 w > 1: 将样条拉到检查点						

表格 5-26: 点权重的可选编程。

示例: 坐标 x、y、z 和方向坐标
样条 (3103,x,y,z,O)
G6 x10 y20 z30 O (0.1,0,1.0) PW2.3

B-样条逼近（样条类型 4）

坐标编程

描述: 编程涉及通道坐标的端点。只有 <Members> 可以编程。

语法:	<table><tr><td><Coordname><Endpos></td><td>坐标的点编程</td></tr><tr><td>其中</td><td></td></tr><tr><td><Coordname></td><td>坐标的名称（包含方向 0）。</td></tr><tr><td><Endpos></td><td>坐标的终点位置</td></tr></table>	<Coordname><Endpos>	坐标的点编程	其中		<Coordname>	坐标的名称（包含方向 0）。	<Endpos>	坐标的终点位置
<Coordname><Endpos>	坐标的点编程								
其中									
<Coordname>	坐标的名称（包含方向 0）。								
<Endpos>	坐标的终点位置								

表格 5-27: 点编程

示例: 在下面的数控程序示例中，相应的 <Q_i> 表示空间中的编程点，包括方向矢量“x .. y .. z .. 0(..)”

```
N00 SplineDef (4213) ;将后期的 G06 定义为 b- 样条逼近
N01 BAP (E0.01, OE0.05) ; 尺寸公差 0.01 毫米，
                           方向矢量的尺寸公差为 0.05 毫米
..
N100 G0 <Q0> ; b- 样条曲线的起点
N101 G06 <Q1>
N102 <Q2>
N103 <Q3>
..
```

数控函数，语法符合 DIN 66025（包括扩展）

N400 <Q₃₀₀> ; b- 样条曲线的最后一点
G1

逼近的公差值

请参阅 第 章 “B-样条逼近参数” 第 115 页 。

开始和结束边界条件

请参阅 第 章 “开始和结束边界条件” 第 114 页 。

样条参数长度

样条参数长度由数控 自动确定并且无法编程。

开始和结束边界条件

除了类型 0（系数编程）以外的所有样条类型，可以在样条序列内的数控程序段中指定起始和 / 或结束边界条件。

SBC(<BC 类型>{,<值>})	样条序列样条块起点的边界条件
EBC(<BC 类型>{,<值>})	样条序列样条块起点的边界条件
其中	
<BC 类型>	0（样条类型 4） 1（样条类型 1、2、4） 样条序列开始点和 / 或结束点的切线方向说明。对于每个样条成员，必须在 <Values> 列表中指定一个值（方向矢量的三个值）。 10（对于样条类型 1, 2, 3, 4） 在当前样条程序段的起始点连接线方向上的开始和结束切点。 11（样条类型 1、2、4） 使用 SBC：当前样条程序段起点处的切线指向前一个程序段起点终点连接线的方向。 2（对于样条类型 1, 2） 在样条程序段的起点或终点处的样条参数“w”之后的二阶导数的规定。对于每个样条成员，必须在 <Values> 列表中指定一个值（方向矢量的三个值）。 3（仅适用于样条类型 2） De Boor 边界条件将二阶导数与前两个和 / 或最后两个插补点的链接。在大多数情况下，<Values> 是 1。这只能在样条序列的开始或结束时编程。 4（仅适用于样条类型 2） 周期边界条件：样条序列的最后一个点和第一个点是相同的。SBC（4）绝对要求 EBC（4），反之亦然。整个样条序列必须处于预见范围内。否则，会创建运行时错误。
<Values>	仅适用于 BC 类型= 1, 2 和 3 默认值： 0, ..., 0 对于 BC 型= 1,2 和 1, ..., 1, BC 型= 3 BC 类型 10、11 和 4 没有值。

表格 5-28: 开始和结束边界条件

示例:

```
SBC (1, 1.0, 1.0, 0.2) with SplineDef (1213,X,Y,B)
SBC (1, 1.0, 1.0, 0.2, 0.0, 0.5, 0.5) with SplineDef
(2203,x,y,z,0)
EBC (2) with SplineDef (2203,x,y,z,0)
```


数控函数，语法符合 DIN 66025（包括扩展）

EBC (10) with SplineDef (3203, x, y, z, 0)

B-样条逼近参数

语法:

BsaPar (E., OE.)	
缩写形式: BAP(..)	
E (Error)	如果缺少 E, 则 MP 80070030 中的值适用 以毫米或英寸为单位指定尺寸公差
OE (方向错误)	方向矢量的尺寸公差 只有在有效的矢量方向的情况下, OE 才是重要的 如果 OE 缺失, 则接通值 OE = E

表格 5-29: B 样条近似参数的语法

5.2.8 路径坡度开启"G8", 路径坡度关闭"G9"

- 描述:
- 如果没有“路径斜坡”, 控制系统将在运行程序段的开始和结束处执行一个完整的向上和向下斜率(速度斜坡), 以使 $v = 0$ 。
尽管这会降低块过渡处的轮廓偏差, 但却需要更长的处理时间。
 - 使用“路径斜坡”, 控制器尝试生成尽可能常量的速度, 与编程的进给一样高(即使在程序段转换时)。这会减少处理时间。
轮廓可以在角落处“留下”的事实对于某些处理过程(更均匀的表面)也是有利的。

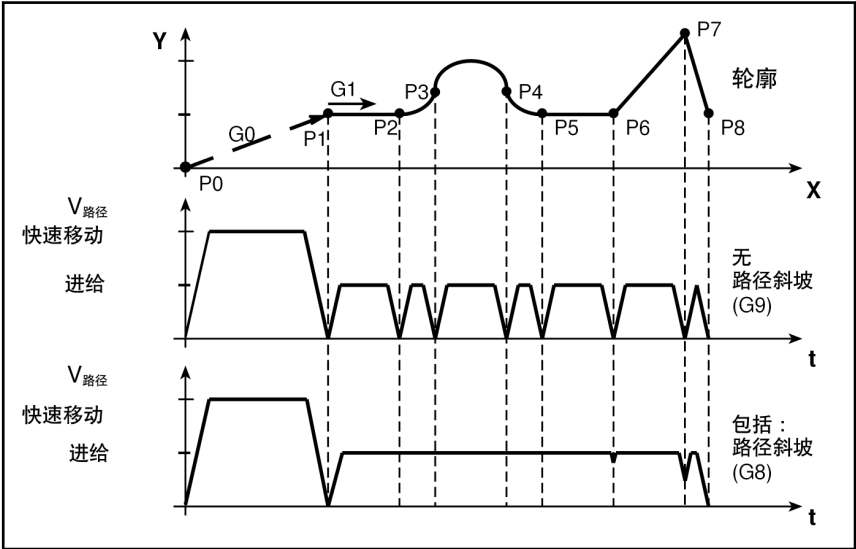


插图 5-11: 路径斜坡

在确定最佳速度曲线时, 控制不仅考虑了轴的跳跃能力值 (MP 1010 00012), 而且还考虑了后续程序段的数量 (程序段预读; MP 7060 00110 – 7060 00130), 因为必须始终确保机器动态范围内的制动。

要限制实际边缘处的轮廓偏差, 最大步进变化不得设置为太高的值。另一方面, 如果最大阶跃变化设置得太低, 则会导致轮廓中轻微膝盖处的不希望有的减速 (几乎是连续的过渡)。

平滑路径加速度可以作为 G8 的默认设置输入机床参数 7050 00320 (另请参阅 第 6.61 章 “激活急进限制 JerkControl、JKC” 第 256 页)。

数控函数，语法符合 DIN 66025（包括扩展）

语法:	G8	路径斜坡开启
	G9	路径斜坡关闭

表格 5-30: G8, G9 语法

- 特殊功能和限制:
- 如果精确停止功能处于激活状态，则无论激活的 G8 如何，控制系统在每个程序段后减速至 $v = 0$ 。
 - 如果 G8 的执行时间（包括确认）不显著低于块的内插时间，辅助功能可以降低 G8 的影响。必要时，必须延长块的运行距离或减少进给量。
 - 路径斜坡只对同步轴（加工轴）有效。
 - 功能 G8 和 G9 与斜坡功能形成一个模态组，从而相互取消选择。

5.2.9

急进限制速度控制"G8（形状 ...）", "G9（形状 ...）", "G9（ASHAPE ...）", "G9（X ..., Y..., ...）", "AsynchrShapeOrder, ASO"

一般描述



SHAPE 功能被轴特定的加加速度限制取代，如果可能的话不再使用（参见“IndraMotion MTX 功能描述”，“轴特定加加速度限制的基础知识”）。我们建议使用轴特定的加加速度限制（即 JerkControl（1））在控制系统内部自动计算 SHAPE。如果停用加加速度限制（即 JerkControl（0）），用户也可以对 SHAPE 进行编程。不能同时使用 SHAPE 功能和轴加加速度限制

对于异步轴，如果设置了 $AX / Dr [i] / Jerk / MaxAxJerk < 0$ ，则所有 SHAPE 数据都将被忽略。

急进限制速度控制使多个插补循环上的路径加速过程中的跳转平滑。插补循环的数量可以事先设定。

这导致平滑的速度转换（加速度限制）。

冲击限制速度控制可用于以下操作模式:

- “路径操作”（在激活 G8 的情况下）
- “定位模式”（G9 有效时）
- “异步模式”

冲击限制速度控制也称为 SHAPE 功能。

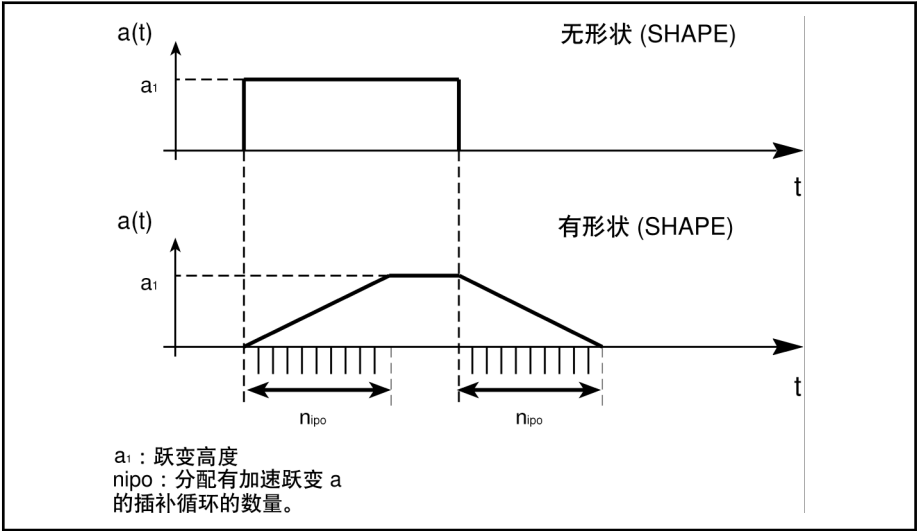


插图 5-12: 带 / 不带 SHAPE 的加速度分布

数控函数，语法符合 DIN 66025（包括扩展）

用于异步轴的形狀

语法:	AsynchShapeOrder <Value>	
	缩写形式: ASO <Value>	
	其中	
	<Value>	控制线性分配加速跳转的插补周期数。 取值范围: 2 ... 100 个循环（整数）。

表格 5-31: AsynchShapeOrder 语法

示例: N10 W150 FA20000 ASO (100)
轴突变结果来自:

$$\frac{MP101000001}{100 * MP903000001 * 10^{-6}} \left[\frac{m}{s^3} \right]$$

如果块中缺少形状定制的信息，则使用参数 1003 00008 “轴的形狀定制”的默认值。

N10 W150 FA1000 AxAcc (W0.5)
轴突变结果来自:

$$\frac{0.5}{MP100300008 * MP903000001 * 10^{-6}} \left[\frac{m}{s^3} \right]$$

该值永远不会低于参数 1003 00007 “轴过冲限制的最小形状定制”。如果未编程 SHAPE 顺序，并且默认和最小 SHAPE 顺序参数未分配 0 或 1，则控制系统移动轴而不受冲击限制。有关异步轴 SHAPE 顺序的信息在块到块的基础上生效，并指所有程序异步轴。

用于路径模式的形狀 - 编程

语法:	G8 { (SHAPE<Value>) }	
	其中	
	<Value>	0 或 1: 用于关闭路径模式的形状。 0 或 1: 2 至 100（整数）: 控制要线性分配跳转的插补周期数（形状顺序）。 没有 SHAPE 参数: 用存储在机床参数 7050 00320 中的 SHAPE 的默认值移动。

表格 5-32: G8 (SHAPE) 语法

 更改形状順序总是会导致 v = 0 的块转换速度。

示例: N30 G8 (SHAPE10) 在轮廓模式下打开 SHAPE 并在 10 个 IPO 循环
:
:
N50 G8 (SHAPE0) 在路径模式下关闭 SHAPE
:
:

数控函数，语法符合 DIN 66025（包括扩展）

N70 G8 (SHAPE1)在路径模式下关闭 SHAPE

:

:

N90 G8用存储在机床参数 7050 00320 中的 SHAPE 的默认值移动

用于定位模式的形状 - 编程

语法:

G9{(SHAPE<Value>)}	打开定位模式的线性 SHAPE。
其中	
<Value>	控制线性分配加速跳转的插补周期数。 取值范围：2 ... 100 个循环（整数）。
没有 SHAPE 参数	用存储在机床参数 7050 00340 中的 SHAPE 的默认值移动。
G9(SIN<Value>)	打开 sine ² -线型形状进行定位模式
其中	
<Value>	控制器以正弦 ² 线型方式分配加速跳转的插补周期数。 以下值有效： 5: 5 IPO 循环 10: 10 IPO 循环 15: 15 IPO 循环 20: 20 IPO 循环 40: 40 IPO 循环
G9(ASHAPE { , TYPE <i> })	
	这将 MP 1003 00008 中相应的轴专用 SHAPE 顺序（IPO 周期数）分配给通道的每个轴，并对该通道的结果块特定 SHAPE 顺序进行块式计算。
G9(<Par ₁ >{, <Par ₂ >, ... } {, TYP<i> })	
	逐轴编程 SHAPE 顺序。如果是非编程轴，则使用来自 MP 1003 00008 的 SHAPE 命令。
其中	
<Par _x >	带有 SHAPE 顺序的逻辑轴名称分配给该轴（最多 100 个）。
示例: N30 G9 (X4,Y6,Z10)	SHAPE 顺序 (x 轴) = 4 SHAPE 顺序 (y 轴) = 6 SHAPE 顺序 (z 轴) = 10
类型 0 或无类型	所有轴的 SHAPE 顺序都是用实际轴加速度与最大轴加速度的比值来计算的。计算出的最高 SHAPE 订单将成为 SHAPE 订单的订单（请参阅下文）。
类型 1	最高轴 SHAPE 顺序变成路径 SHAPE 顺序。

表格 5-33: G9 (SHAPE) 和 G9 (ASHAPE) 的语法

合成形状阶数

描述: 结果路径形状顺序 S_b 是插值中所有轴的有效轴形状顺序 S_i^{eff} 的最大值:

$$S_b = \max \{S_1^{eff}, \dots, S_n^{eff}\}$$

对于 G9 (... , TYPE0) 或没有关于 TYPE 的信息, 有效轴形状定制 S_i^{eff} 使用编程的形状定制通过公式计算:

$$S_i^{eff} = S_i^p \frac{a_i^{eff}}{a_i^{max}}$$

以下适用于 (G9 ... , TYP1)

$$S_i^{eff} = S_i^p$$

在这种情况下:

S_i^p	是用 G9 (...) 编程的 SHAPE 轴顺序
a_i^{eff}	是当前数控块中的有效轴加速度 在线性插补的情况下, 该值取决于轴的当前路径分段。对于其他类型的插值 (例如圆弧, 螺旋), 它对应于通常用 "AxAcc" 编程的轴加速度。 如果使用功能 "斜面" 或 "轴耦合", 与 "AxAcc" 值相比, 有效轴加速度一般会再次下降!
a_i^{max}	最大轴加速度 (参数 AX / Dr / Acc / FeedAxAcc (MP 1010 00001))



"AxAcc" 不会更改最大轴加速度参数 AX / Dr / Acc / FeedAxAcc (MP 1010 00001) !

形状阶数和急进之间的关系

描述: 有了轴形状阶数 S_i^{eff} , r_i^{max} 的最大加加速度 (加速度对时间的导数) 被定义为一个限制, 在任何运动形式下均不可超越此限制。
该过冲由以下项目定义:

$$r_i^{max} = \frac{a_i^{max}}{S_i^p T_{ipo}}$$

T_{ipo} : 内插器的周期时间

示例: x 轴的最大加速度 (参数 AX / Dr / Acc / FeedAxAcc (MP 1010 00001)) 为 10 m/s²。编程的轴 SHAPE 顺序为 5 毫秒, 插补周期为 4 毫秒。
根据上面的公式, 因此定义了 x 轴的最大加加速度 500 m/s³。

5.2.10 无平面 "G16"

描述: 取消激活的平面。
以下情况适用:

- 如果主轴或副轴从通道中移除 (例如与功能 "轴传输" 相关), 控制系统会自动关闭所选平面并激活功能 G16。
只要没有选择有效的平面, 就不能在该通道中执行圆形或螺旋插补。
- 如果既不需要圆周插补, 也不需要螺线插补 (例如, 只有一个加工轴的通道), 则某些应用、机床类型或者处理功能就不需要进行平面配置。

数控函数，语法符合 DIN 66025（包括扩展）

在参数 CHAN / Ch / Coord / Acs / ChAxClass (MP7010 00030)（坐标显着性）中，可以为每个轴输入“0”（没有工作相关的显着性）。

- 如果在开机状态（参数 CHAN / Ch / Ini / ChResetState (MP 7060 00020) 和参数 CHAN / Ch / Ini / NcResetState (MP 7060 00010) 中没有输入平面功能 (G17, G18, G19, G20) ），功能 G16 会自动激活相应的通道。
- 功能 G16, G17, G18, G19 和 G20 组成一个模态组，因此相互取消选择。

语法:

G16

表格 5-34: 语法 G16

特殊功能和限制: 在激活的铣刀轨迹修正 (G41 / G42) 中，不能切换或禁用活动平面。

5. 2. 11 平面切换 "G17、G18、G19"

描述: 使用平面切换功能选择活动平面。该平面由一个主坐标和一个辅助坐标定义，并且必要时还包括一个垂直于该平面定位的进给轴。因此，G17, G18 或 G19 编程时不带参数。

对于典型的笛卡尔标准机床刀具，轴 X, Y 和 Z 跨越笛卡尔工件坐标系（WCS）。使用工件坐标系中的两个坐标轴，可以定义总共三种不同的基准级。

使用平面切换功能选择活动平面。该平面由一个主坐标和一个辅助坐标定义，并且必要时还包括一个垂直于该平面定位的进给轴。因此，G17, G18 或 G19 编程时不带参数。

G17、G18 和 G19 指定这些坐标以及有效平面的过程如下：

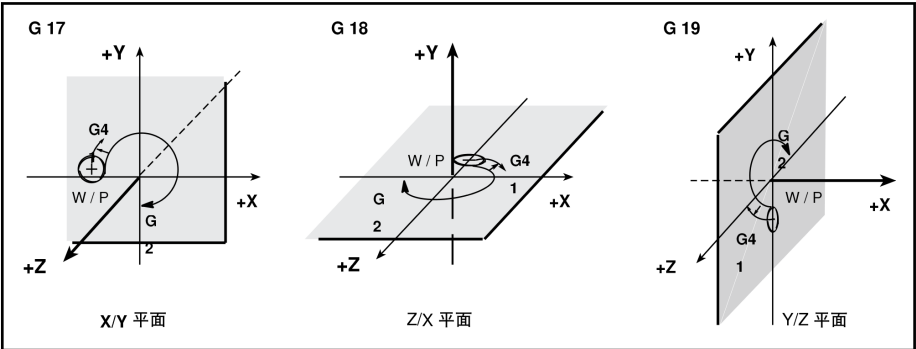


插图 5-13: 平面切换

	主要坐标	辅助坐标	进给坐标	激活的平面
G17	X	Y	Z	XY
G18	Z	X	Y	ZX
G19	Y	Z	X	YZ

表格 5-35: 坐标 G17, G18 和 G19 的概述

语法:

G17

G18

G19

表格 5-36: G17, G18, G19 的语法

- 特殊功能和限制:
- 在激活的铣刀轨迹修正 (G41 / G42) 中，不能切换或禁用活动平面。
 - 以下内容适用于 G2 / G3 和 G33 的插补参数 I、J、K:

数控函数，语法符合 DIN 66025（包括扩展）

- “I”指的是 x 轴
- “J”指的是 y 轴
- “K”指的是 z 轴
- 利用 G47(ActPlane) 将刀具长度修正分配给有效平面，如下所示：
主坐标的 L1
辅助坐标的 L2
横向进给坐标的 L3。
活动平面切换后更新此分配。
- 功能 G16, G17, G18, G19 和 G20 组成一个模态组，因此相互取消选择。

与轴名称无关的指定：轴名称可以在 IndraMotion MTX 中任意设置。通过机床参数 7010 00030（坐标显着性）来选择工件坐标系 WCS 要跨度的轴。与配置的轴名称，该参数指定哪些轴打开工件坐标系的 X, Y 和 Z 坐标：
带“轴影响 X”的轴：WCS 的 X 坐标
带“轴影响 Y”的轴：WCS 的 Y 坐标
带“轴影响 Z”的轴：WCS 的 Z 坐标

一般来说，G17、G18 和 G19 的分配结果如下：

	主要坐标	辅助坐标	进给坐标	激活的平面
G17	X- 坐标 WCS	Y- 坐标 WCS	Z- 坐标 WCS	XY- 坐标 WCS
G18	Z- 坐标 WCS	X- 坐标 WCS	Y- 坐标 WCS	ZX- 坐标 WCS
G19	Y- 坐标 WCS	Z- 坐标 WCS	X- 坐标 WCS	YZ- 坐标 WCS

表格 5-37: 轴名称分配概述

5.2.12 高级平面切换"G17 (...) ， G18 (...) ， G19 (...) "

描述：除了选择有效平面之外，高级平面切换还定义了跨越当前工件坐标系（WCS）的轴。
因此，可以在运行时定义偏离机床参数设置的工件坐标系。基于此坐标系，可以选择当前平面。

根据以下概览进行平面选择：

	主要坐标	辅助坐标	进给坐标	激活的平面
G17	X- 坐标 WCS	Y- 坐标 WCS	Z- 坐标 WCS	XY- 坐标 WCS
G18	Z- 坐标 WCS	X- 坐标 WCS	Y- 坐标 WCS	ZX- 坐标 WCS
G19	Y- 坐标 WCS	Z- 坐标 WCS	X- 坐标 WCS	YZ- 坐标 WCS

表格 5-38: 轴名称指定概述

语法：	G17 (<Axis1>,<Axis2>,<Axis3>)	
	G18 (<Axis1>,<Axis2>,<Axis3>)	
	G19 (<Axis1>,<Axis2>,<Axis3>)	
	其中	
	<Axis1>	工件坐标系（WCS）的“X 坐标”

数控函数，语法符合 DIN 66025（包括扩展）

<Axis2>:	工件坐标系（WCS）的"Y 坐标"
<Axis3>:	工件坐标系（WCS）的"Z 坐标"

表格 5-39: G17 - G19 语法

- 特殊功能和限制:
- 轴可以使用逻辑（基于通道）或物理（系统范围）轴名称进行编程，并且必须彼此不同。直接输入轴或作为 CPL 字符串变量输入。
 - 编程的轴当前必须分配给通道。
 - 编程时，只能输入要重新配置为参数的轴。所有其他轴的 WCS 的相应坐标保持不变。
示例: G17 (, ,W)
WCS 的 Z 坐标通过 W 轴定义。
WCS 的 X 和 Y 坐标保持不变。
有效平面由 WCS 的 X 和 Y 坐标打开。
 - 编程一个空的括号表达式，恢复在 MP7010 00030 中配置的 WCS，然后激活选定的平面。
示例: G17 ()
 - 在铣削铣刀路径修正（G41/G42）有效时，不得编程该功能。
在有效的铣刀路径修正中，不得切换或禁用该平面。
 - 如果轴转换激活，则不能编程高级平面切换，因为 WCS 由空间坐标构成。
 - 功能 G16, G17, G18, G19 和 G20 组成一个模态组，因此相互取消选择。

5.2.13 自由平面选择（与 WCS 独立）"G20"

描述:

如果有效平面必须与当前工件坐标系统 WCS 脱耦，就需要自由选择平面。该功能允许使用任何轴来跨越坐标系。这些轴是否也跨越当前的 WCS 并不重要。

因此，也可以在工件变换（放置，跨度修正）同时参考其他跨越的工件坐标系时当前平面的主坐标，辅助坐标和进给坐标中确定刀具校正（G41，G42，G47）轴。

语法:

G20 (<Axis1>,<Axis2>{,<Axis3>})	
其中	
<Axis1>	该平面的主要坐标将跨越 被分配给插值参数"I"。
<Axis2>:	该平面的辅助坐标将跨越 被分配给插值参数"J"。
<Axis3>:	该平面的横向进给坐标将跨越 如果需要声明刀具补偿"G47 (ActPlane)"的进给坐标，可以进行编程。

表格 5-40: 语法 G20

- 特殊功能和限制:
- 可以采用逻辑（基于通道）或者物理（系统范围）轴名进行轴编程。
 - 铣削铣刀路径修正 G41/G42 参考选定的平面。因此，仅当铣削铣刀路径修正（G40）不处于有效状态时，才可切换有效平面。否则，将产生一个故障报告。
 - 利用 "G47 (ActPlane)" 将刀具长度修正分配给有效平面，如下所示：

数控函数，语法符合 DIN 66025（包括扩展）

主坐标的 L1
辅助坐标的 L2
横向进给坐标的 L3。
活动平面切换后更新此分配。

- 功能 G16, G17, G18, G19 和 G20 组成一个模态组，因此相互取消选择。

5. 2. 14 轮廓移动编程"G31"

描述: 轮廓移动编程是一种特殊类型的轮廓定义，可以将绘图尺寸几乎完全应用为数控参数。轮廓可以由直线和圆弧段组成。

轮廓移动编程支持例如:

- 直线编程
 - 直线点对点，三点和四点移动
 - 连接倒角
 - 与标准编程相比，使用扩展编程选项进行循环编程
 - 对于不完全定义的过渡曲线段的自动相交确定：直线 - 直线，圆 - 圆，直线 - 圆或圆 - 直线
 - 两圆弧之间的圆弧和切向连接圆的切线直线连接
- 可以选择使用数控函数 CHSL, CHLL 和 RNDL 对转换倒角和倒圆进行编程。

G31 轮廓总是指当前有效的平面（G17, G18, G19）。

功能参数

X, Y, Z, ...	坐标名称（如配置）
A	角度（关于极坐标编程的极角 1 的语法;可通过机器参数设置）
I, J, K	圆心点的插补参数（对于 G31，圆心点始终指定为绝对值）
R	圆半径
CW, CCW	圆的旋转方向：顺时针，逆时针
TG	切向过渡或切向连接
CS	连接倒角的倒角段
CL	连接倒角的倒角长度

表格 5-41: 功能 “轮廓移动编程 G31” 的参数

语法:

G31 直线编程的变体:	
X	带编程终点的直线（如 G1）
一个 <Angle>	带角度编程的直线
一个 <Angle> X ... 或 Y ... 或 Z ...	带有角度和编程坐标的直线（直线点对点移动）
G31 A <Angle> X ...Y ... 或 Z ...X ... 或 Y ...Z ...	带有角度规格和完全编程终点的直线
一个 <Angle> CL <Length> X	带角度的直线；编程的长度与编程的终点
一个 <Angle> CS <Section> X	带角度的直线；编程的倒角部份与编程的终点

数控函数，语法符合 DIN 66025（包括扩展）

TG	线性切向连接（例如两个圆之间）
TG X	与编程终点的切线连接

表格 5-42: G31 直线编程的变体

G31 圆编程的变体:	
X I CW 或 CCW	标准圆
X I	没有指定旋转方向的圆
X ... 或 Y ... 或 Z ...I CW 或 CCW	位置规格不完整的圆
I CW 或 CCW	未定义终点的圆
I R <circle radius> CW 或 CCW	未定义的圆
R <Circle radius> TG	切向连接圆
X R <circle radius> TG CW 或 CCW	切线开始的圆

表格 5-43: G31 圆编程的变体

示例:

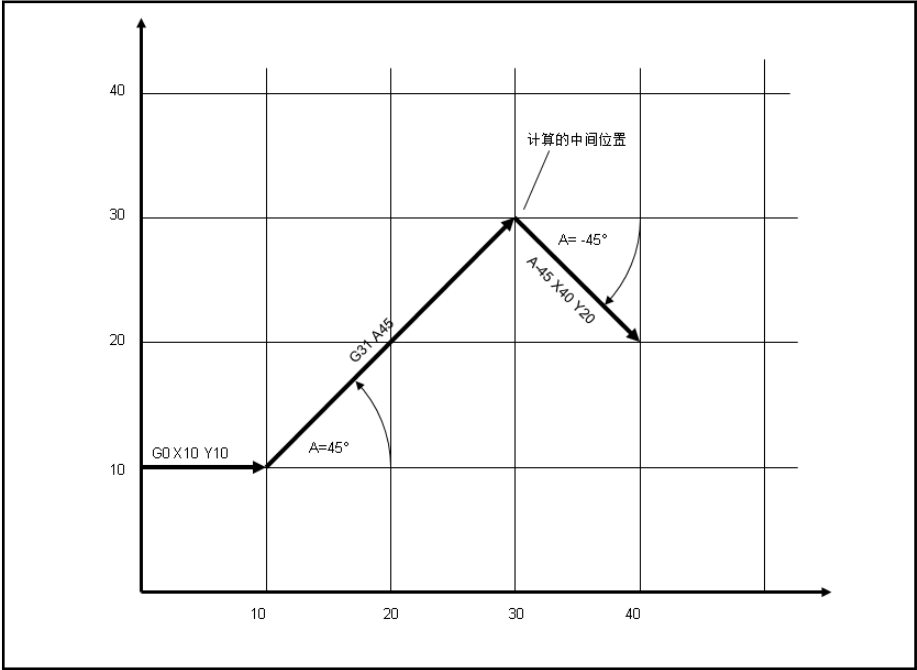


插图 5-14: 直三点移动

程序

```
N10 G0 X0 Y10
N20 G0 X10 Y10
N30 G31 A45 F1000 ; Straight three-point move
N40 A-45 X40 Y20 ; To end position X40 Y20
M30
```

特殊功能和限制:

- 圆心点总是以绝对坐标编程。
- 两个坐标必须总是以圆心点编程：主轴坐标和辅轴坐标。
- 如果在模态倒角或倒圆有效时编程局部倒角和倒圆，局部倒角或倒圆将覆盖模态倒角或倒圆。

数控函数，语法符合 DIN 66025（包括扩展）

5. 2. 15 螺纹切削 "G33"

描述: "螺纹切削"允许切割下列螺纹:

- 纵向螺纹
(切削运动与有效平面的主轴平行)
- 面螺纹
(切削运动与有效平面的辅轴平行)
- 锥形螺纹
(切削运动与有效平面的主轴和辅轴有关)

可以用速度受控主轴和位置受控主轴对 G33 进行编程。

切削运动始终与相关通道上有效的主轴相链接 (请参阅 第 6 章 "使用高级语言语法的 NC 功能" 第 181 页)。

从当前主轴速度和编程的螺距部分 (固定的、可变的, 参见语法) 得出切削运动的进给。

特殊特性:

- 可以加工单头和多头螺纹。
- 可以对固定的和可变的螺纹节距编程。
- 快速回缩运动可以编程。
- 排序的螺纹可以被制造

 进给电位计在 G33 有效时不起作用。

 至于圆弧插补 (G2, G3), 螺纹切削功能也取决于有效平面 (G17 ... G20)。

语法:

G33 <EP> <FixedPitch> {<Var. pitch>} {开始角度}	
其中	
<EP>	有效平面主轴和辅轴的轴坐标通过 G17、G18、G19 或者 G20 定义有效平面。 示例: 通常通过 Z 轴 (主轴) 和 X 轴 (辅轴) 定义 G18 的有效平面。
<fixed pitch>	定义主轴每转一圈, 沿主轴或者辅轴方向所行进的路径 (单位 mm)。通过在对有效平面中有效的插补参数 (I、J 或者 K) 来编程该值。 对于锥形螺纹, 指定的螺距必须总是参考主切削方向。 示例: 对于 G18, 参数 K 被分配给主轴, 而 I 被分配给辅轴。对于纵向螺纹 (节距沿主轴方向), 用 K 地址编程固定螺纹。

数控函数，语法符合 DIN 66025（包括扩展）

<Var. Pitch>	采用地址 DF 的可选参数 定义主轴每转一圈节距的增加 / 减少（单位 mm）。 编程：“DF<Value>” 其中 <Value> 单位为 mm。
<Starting angle>	可选参数 如果没有对 <Starting angle> 编程，就假定为 0 度。 多头螺纹需要起始角（偏置）。未分配给有效平面的插补参数作为位置使用。 示例： 地址 I 和 K 已经分配给 G18 的平面。因此，起始角的位置就是 J。

表格 5-44: G33 语法

示例:

纵向螺纹

G91 G18 G8 M3 S1000	激活相对尺寸编程 激活 Z / X 平面
G0 X-10	切削刀具的横向进给运动 (1)
G33 Z-50 K2	螺纹切削 (2) 终止点: Z- 方向增量-50mm 固定螺纹节距: 2 mm/ 转 插补参数: 这里是 K
G0 X10	回缩切削刀具 (3)

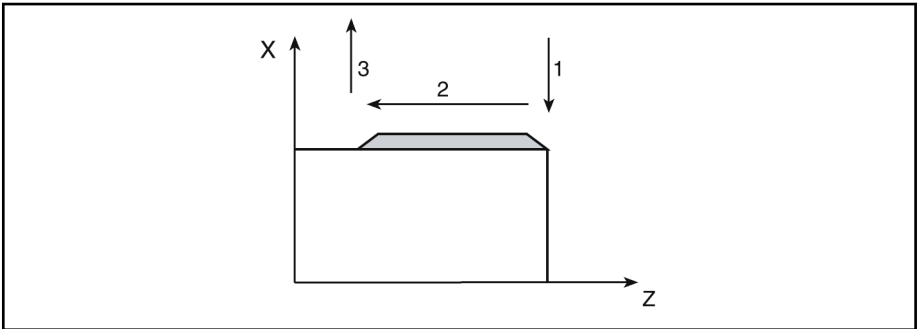


插图 5-15: 纵向螺纹

面螺纹

G91 G18 G8 M3 S1000	激活相对尺寸编程 激活 Z / X 平面
G0 Z-10	切削刀具的横向进给运动 (1)
G33 X40 I2	螺纹切削 (2) 终点: x- 方向增量-40 mm 固定螺纹节距: 2 mm/ 转 插补参数: 这里是 I
G0 Z10	回缩切削刀具 (3)

数控函数，语法符合 DIN 66025（包括扩展）

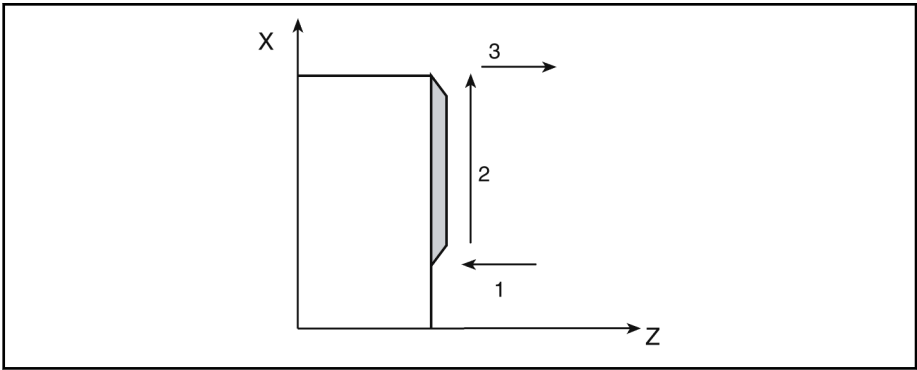


插图 5-16: 面螺纹

锥形纵向螺纹

G91 G18 G8 M3 S1000	激活相对尺寸编程 激活 Z / X 平面
G0 X-20	切削刀具的横向进给运动 (1)
G33 Z-50 X15 K2	螺纹切削 (2) 结束点: Z 方向增量 -50 mm, X 方向增量 +15 mm 固定螺纹节距: 2 mm/ 转 插补参数: 这里是 I
G0 X5	回缩切削刀具 (3)

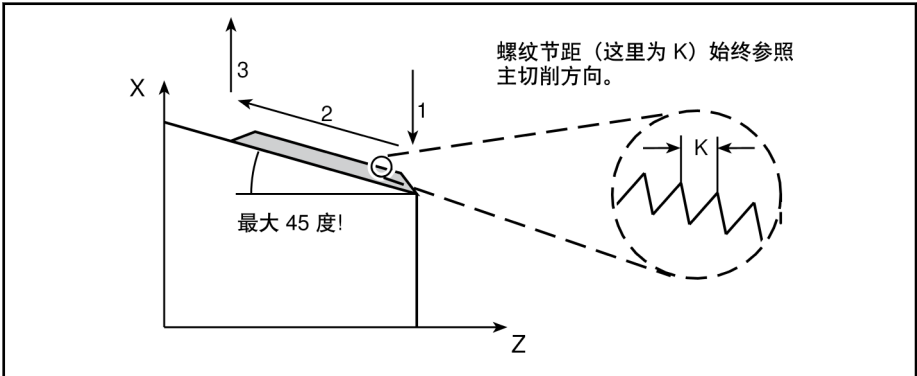


插图 5-17: 锥形纵向螺纹

连续螺纹

- 这些可以从上述所有的螺纹类型中制造。
- 可以用数个连续 G33 块对它们进行编程。

利用每个已编程的 G33 块，数控 检查是否已使用距离（路径）编程后续 G33 块。如果是这种情况，则将其更改为没有轴静止的下一个程序段。

多头螺纹

多头螺纹是通过移位起始角度偏置量来制造的（对于起始角度，请参阅“语法： ” 第 125 页 ）。

四头螺纹由四个切口各 90 度（0, 90, 180, 270）制造而成。

数控函数，语法符合 DIN 66025（包括扩展）

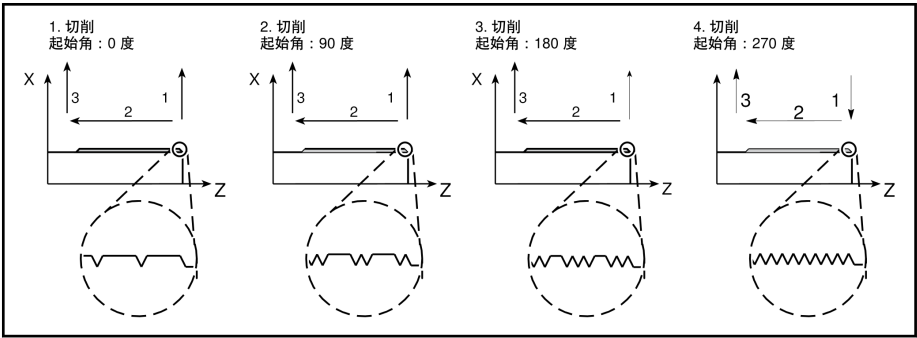


插图 5-18： 四头螺纹

- 特殊特性**
- 在螺纹切削制造的开始和结束时，涉及的轴必须加速或减速。
因此，始终指定足够长的入口路径（以加速切削轴）并保留路径（以停止）。
 - 功能“ThreadSet (...)”
(参见 第 6.72 章 “主轴切换 MainSp、MSP” 第 266 页) 可以和 G33 一起使用。
 - 调整切割动作的动态。
 - 配置通过接口信号激活的回退运动。
 - 切换主轴模式（速度控制，位置控制）。
 - 在通道接口设定信号（可使用机床参数在“有效函数”信号内配置）。

5.2.16 铣刀路径修正"G40, G41, G42"

描述： 铣削铣刀路径校正功能在执行零件特定程序（等距路径=距编程轮廓直角常量距离的路径）时，沿着与编程路径平行的等距路径移动刀具。等距离和编程路径之间的距离取决于有效半径校正值。

下图显示了原理：

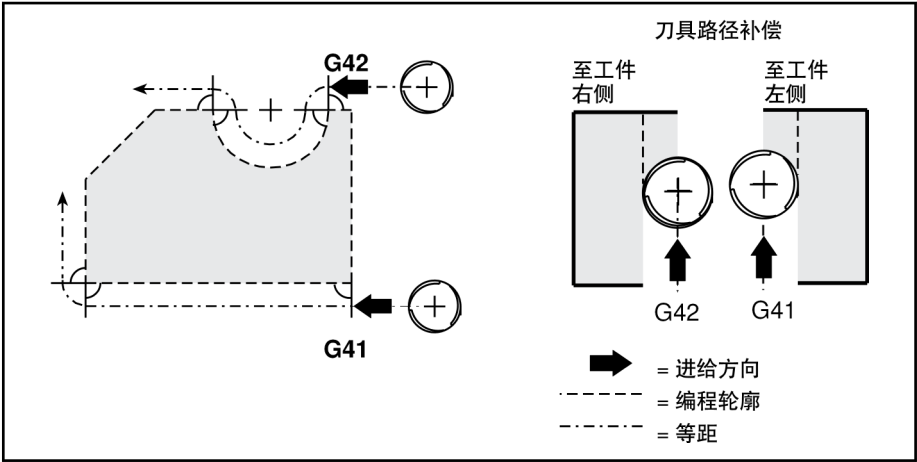


插图 5-19： 铣刀路径校正

语法：

数控函数，语法符合 DIN 66025（包括扩展）

G40	铣刀路径校正关闭（开启状态） G40 程序段无编程运行的行为可通过机床参数设置： - 垂直于前一个运行程序段立即退出修正编程（默认设置）。 - 下一个编程的运行程序段退出修正建立。 如果在 G40 程序段中编程了运行运动，则在运行到编程的运行终点时，控制器会线性停止刀具路径修正。根据是否存在内角或外角，从当前块的前一个垂直线或起始垂直线开始退出校正堆积。如果是外角的话，按照当前轮廓过渡的行进方式（G43/G44），插入一个过渡路径分段。 这种行为也适用于在 G40 程序段之后退出修正编程而不进行运动，如果修正建立只能通过下一个运行程序段退出。
G40（NOM）	铣刀路径校正关闭 编程时不进行遍历运动。 修正只能在下一个运行程序段中收回。可以认为以下编程的增量运动指的是刀具的中心。该刀具被给定的增量运动所遍历。如果使用 G40 直接在一个程序段中编程增量运行运动，则生成的运动会有所不同。
G40（ORTH）	铣刀路径校正关闭 无论在关闭程序段的轮廓过渡处是否存在内角或外角，退出修正建立都会从前一个程序段的垂直末端开始。校正累积沿着到编程的横动运动的终点的路径线性地退出。
G41	工件左侧的铣刀路径校正开启 （对于在加工方向上看到的正的校正值）。 G41 程序段的行为可以通过机床参数进行设置： - 垂直于前一个运行程序段立即开始校正补偿（默认设置）。 - 只有下一个编程的运行程序段才开始修正。 在带有编程运行的 G41 程序段中，修正建立在该程序段中开始。 要在一个运行程序段中开始校正补偿，以下情况适用：校正正在到编程终点的途中线性增加。 如果存在内角，修正建立从下一个编程的运动的起始垂线开始。 如果存在外角，则在开启块的垂直端开始激活修剪路径修正。按照需要，根据当前轮廓过渡的行进方式（G43/G44），插入一个过渡路径分段。
G41（ORTH）	工件左侧的铣刀路径校正开启 无论在随后的轮廓转换中是否存在内角或外角，校正累积都是从后续运行程序段的起始垂线开始的。
G42	工件右侧的铣刀路径校正开启 否则与 G41 相同
G42（ORTH）	工件右侧的铣刀路径校正开启 否则，如 G41（ORTH）

表格 5-45: G40-G42 语法



无需编程单独移动运动，就可直接激活或停用修正值。这会损坏工件或刀具。

因此，请注意本章中的所有信息！

- 特殊功能和限制：
- G40, G41 和 G42 模态操作并取消选择对方。
 - 通过机床参数和功能参数 NOM 和 ORTH 来影响启动和退出修正建立的行为。

数控函数，语法符合 DIN 66025（包括扩展）

- 在 G41 或 G42 有效时，**不允许**进行以下操作：
 - G17 ... G20（平面切换）
 - G63（没有进行卡盘补偿的攻丝）
 - G74（接近参考点坐标）
 - G75（前往探测头）
 - MOB（在块上测量）
 - G76（靠近固定的机床轴位置）
 - G54. x ... G59. x（零点偏置）
 - G154. x ... G159. x（斜面）
- G40, G41 和 G42 不影响任何刀具长度修正。

5.2.17 铣刀路径修正的轮廓转换：圆弧"G43"，交叉"G44"圆弧

描述： 用于有效刀具路径修正的功能（G41、G42）。

该控件在**外角处**实现轮廓过渡，或者是自动生成的

- 自动生成的弧线（G43），或者
- 自动生成的横切等行程路径（G44）。

G43: 弧线

该路径间隙由半径为“r”的切向弧线弥合：

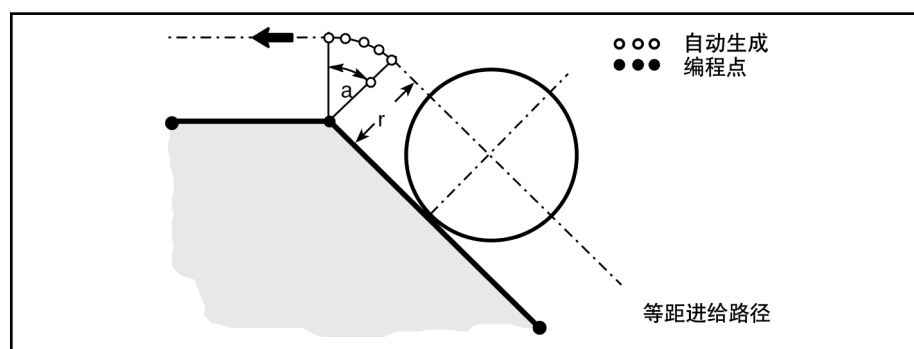


插图 5-20: 弧线 G43

G44: 交叉

控制器尝试用两个等距的交叉来弥合路径间隙。

根据轮廓角“KE”和交叉“S”之间的距离“A”，控制器实现以下操作：

数控函数，语法符合 DIN 66025（包括扩展）

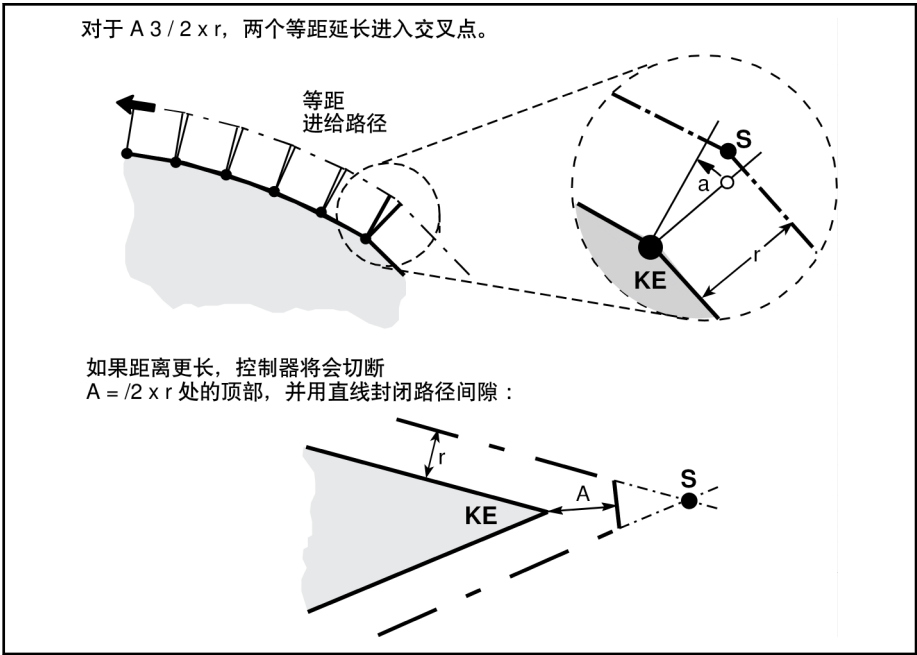


插图 5-21: G44 交叉



如果不存在交叉，那么就采用弧线将线弥合（一如 G43）。

语法:

G43	弧线轮廓过渡
G44	等距交叉轮廓过渡

表格 5-46: G43, G44 语法

以下情况适用:

- G43 和 G44 都是以模态方式作用，而且相互取消。
- 接通状态可在机床参数中进行指定。

特殊功能和限制: 在没有路径条件的情况下编程 G43/G44。

5.2.18 进给校正刀具接触点"G45"、刀具中心点"G46"

描述: 用于有效刀具路径修正的功能（G41、G42）。
在圆弧插补处，定义控制是否保持编程的进给

- 位于刀具接触点（刀具切削路径）或
- 刀具中心点路径上

以常量等级。

数控函数，语法符合 DIN 66025（包括扩展）

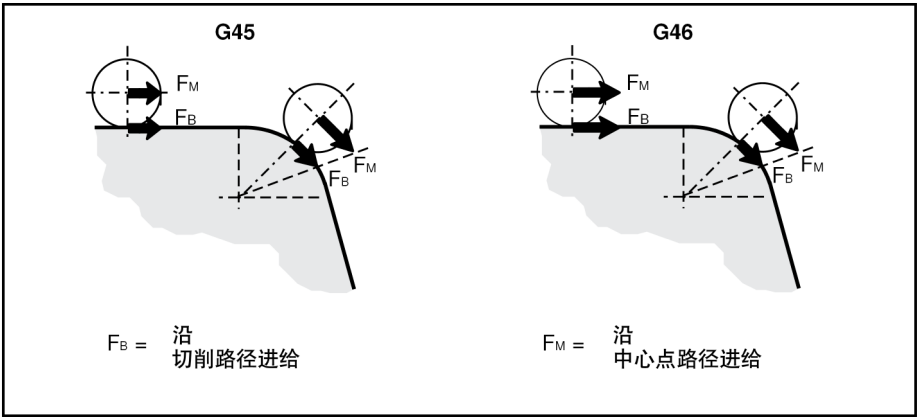


插图 5-22: 进给 G45 和 G46

语法:	G45	以常量的水平，沿切削路径保持进给 F_B 。
	G46	以常量等级沿切削刀具中心点路径保持进给 F_M 。

表格 5-47: G45, G46 语法

以下情况适用:

- G45 和 G46 都是以模态方式作用，而且相互取消。
- 接通状态可以在机床参数中进行指定。

特殊功能和限制: 由于沿圆形轮廓的进给速度可以显著增加，因此只能使用 G45 进行精铣。

5. 2. 19 刀具长度修正"G47、G48"

描述: 该功能

- 激活 / 取消刀具长度修正
- 可选地将长度修正值 L_1 、 L_2 和 L_3 的分配切换为各个坐标（对于 G78 的操作方法；参见 第 5.2.31 章“补偿切换打开 G78、补偿切换关闭 G79”第 144 页）。

当刀具长度修正有效时，以下内容将有效:

- 当前选定的 D 修正设置 (Dxx) 的长度修正值 L_1 、 L_2 和 L_3
- 加上外部刀具修正 (EDxx) 的长度修正值 L_1 、 L_2 和 L_3 ;
- “静态刀具方向”功能（可以使用 ST0 进行参数设定；参见 第 6.137 章“静态刀具定向的参数设置 StatToolOri、ST0”第 345 页）；
- 功能“刀具刃口修正”，如果刀具刃口已经在当前选定的 D 修正块 (Dxx) 输入，铣削刀具路径修正 G41/G42（请参见 第 5.2.16 章“铣刀路径修正 G40, G41, G42”第 128 页）有效。

语法:	G47	刀具长度修正打开
	G47 (ActPlane)	对有效平面主坐标、辅助坐标和垂直坐标的长度修正值 L_1 、 L_2 和 L_3 进行赋值，然后打开刀具长度修正。保持模态有效，并在以后每次平面切换时重新进行自动配置。
	G47 ({ {-} <L1-coord> } {, { {-} <L2-coord> } {, { {-} <L3-coord> } })	将长度补偿值 L_1 、 L_2 和 L_3 的赋值更改为各个坐标，然后激活长度补偿。

数控函数，语法符合 DIN 66025（包括扩展）

G47 ()	根据机床参数 7050 01300 和 7050 01310 的设置，对长度补偿值 L_1 、 L_2 和 L_3 进行赋值，然后激活长度补偿。
G48	刀具长度修正关闭。
其中	
<Li coord>	在有效工件坐标系统中将被赋予补偿 L_i（其中 $i = 1、2、3$）的 WCS 坐标名称。 如果补偿值 L_i（其中 $i = 1、2、3$）将作用于刀具坐标系统（TCS）中的坐标，必须使用坐标识别符 XTR、YTR 和 ZTR，而且相应的轴变换必须已激活。 可选用的负号： 修正方向为负方向。 没有赋予坐标的 L_i 修正值不予以考虑。

表格 5-48: 语法 G47, G48

- 特殊功能和限制:
- 不属于默认通道设置的坐标不会在“G47 ()”语法中考虑。
 - G47 和 G48 9 都是以模态方式作用，而且相互取消。
 - 根据所使用的坐标名称（参阅上述 <coordinate i>），确定 G47 中编程的坐标位置参照的是工件坐标还是刀具坐标。
 - 混合编程如“G47 (X , , ZTR)”是不允许的。
 - 可以与其他路径条件、进给信息或者辅助功能一起对 G47/G48 编程。

示例:	G47 (X , , -Z)	L_1 修正被分配给工件坐标系统（WCS）的 Y 轴，并以负方向计算。并未计算 L_2 修正，原因是未将其分配到任何坐标。 L_3 修正被分配给工件坐标系统（WCS）的 z 轴，并以负方向计算。 后续启动刀具长度修正。
	G47 (, , ZTR)	并未计算 L_1 修正和 L_2 修正，原因是未将其分配到任何坐标。 L_3 修正被分配给工件坐标系统（WCS）的 z 轴，并以正方向计算。 后续启动刀具长度修正。

5. 2. 20 可编程零点偏置"G52"

描述: 可编程零点偏置用来通过在零件程序中编程，实现移动机床坐标系统 MCS。

可编程零点偏置总共可激活 5 个有效零点偏置页（组）。

来自不同零点库的偏置总是叠加的。

可编程的零点偏置是一种允许将零点偏置值直接编程到零件程序中的变量（不使用零点偏置表），从而补充零点偏置量 G54.1 至 G59.5，零点偏置量必须从零偏置量表。

同时，特定库的可编程零点偏置和零点将对应库的 G54.x 至 G59.x 偏置，形成一个模态组。模态组的功能相互取消选择。

使用 G53.x 取消特定库的零点偏置。G53 取消了所有的零点偏置。

数控函数，语法符合 DIN 66025（包括扩展）

语法：	G52.〈Z0 库〉（〈机床坐标〉）〈Z0 库〉的可编程 Z0 开启	
	缩写形式：G52 用于库 1 的可编程 Z0 其中	
	〈Z0 库〉 1...5，整数	

表格 5-49: G52 语法

用下面的方式取消选定可编程零点偏置：

关闭所有修正库 G53 Z0

关闭 〈Z0 库〉的 G53.〈Z0 库〉

G54.〈Z0 库〉...〈Z0 库〉的 G59.〈Z0 库〉零偏置（在表格基础上）开启

以下情况适用：

- G52.1, G53.1, G54.1 ... G59.1
都是模态，且可以相互取消选择。
G52.2, G53.2, G54.2 ... G59.2
都是模态，且可以相互取消选择。
G52.3, G53.3, G54.3 ... G59.3
都是模态，且可以相互取消选择。
G52.4, G53.4, G54.4 ... G59.4
都是模态，且可以相互取消选择。
G52.5, G53.5, G54.5 ... G59.5
都是模态，且可以相互取消选择。
G53 激活功能 G53.1 ... G53.5 和
由此取消对所有零点偏置的选择。
- 如果单独编程，则零点偏置不会导致任何进给运动。只是当前机床坐标系（MCS）被偏置。
- 可与进给运动一起对可编程零点偏置编程。更改后的 Z0 移动当前 MCS，从而也改变当前的 WCS。之后同一块中已经编程的进给运动将参考新的 WCS。

示例：不同 Z0 页的零点偏置的叠加作用

程序：	
N20 G52.1 (X100, Y100)	无进给运动 ACS 位置 X100 Y100 上的零点偏置已经激活。
N30 G52.2 (X300, Y100)	无进给运动 叠加零点偏置已经激活。 结果机床零点现在位于 ACS 位置 X400 Y200。
N40 G90 G1 F1000 X10 Y15	轴进给至 ACS 位置 X410 Y215。
N50 G53.1	零点偏置第 1 库关闭。 机床零点现在位于 X300 Y100。
N60 X10 Y10	轴进给至 ACS 位置 X310 Y110。
N70 G53	所有零点偏置已关闭。

特殊功能和限制：可编程零点偏置参照机床坐标。
可编程零点偏置可在一个程序块中用其他路径条件编程。

数控函数，语法符合 DIN 66025（包括扩展）

在一个块中可以通过零点偏置编程进给运动。

5.2.21 零点偏置 (Z0) "G53"、"G53.1-G59.1" 至 "G53.5-G59.5"

描述: Z0 可用于偏置空间中的机床坐标系统。

 如果需要跨越校正，请参阅功能"BcsCorr"。

通道中机床坐标的偏置距离保存在“零点偏置表”中:

- Z0 表包含 5 个 Z0 页（组），每页有 6 个零点偏置 (Z0)。
- 来自不同 Z0 库的偏置总是叠加的。
- 同一 Z0 库中的偏置彼此覆盖。
- 通道号为 0 的偏置在其激活的每个通道中生效。但是，通道号 > 0 的偏置只能在指定的通道中生效。如果多个输入具有不同通道详细信息的相同机床坐标，则不要使用通道编号 0，因为在此情况下，不考虑此坐标的所有后续通道专用偏置。

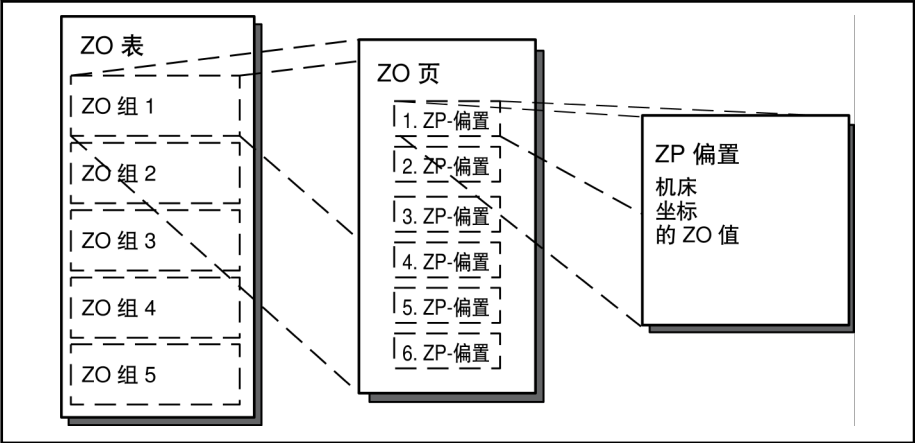


插图 5-23: 零点偏置 (Z0)

 有关编辑 Z0 表格的详细信息，请参阅操作说明。

语法: 利用 "ZoTsel" 激活需要的 Z0 表（请参见 第 6.154 章 "激活零点偏置表 ZoTsel、ZOS" 第 369 页 ）。

对需要的功能编程:

G53	关闭所有修正库 Z0s
G53.<Z0 page>	<Z0 库> 的所有 Z0 关闭
G54.<Z0 库>	1.<Z0 库> 的 Z0 开启
G55.<Z0 库>	2.<Z0 库> 的 Z0 开启
G56.<Z0 页>	3.<Z0 库> 的 Z0 开启
G57.<Z0 库>	4.<Z0 库> 的 Z0 开启
G58.<Z0 库>	5.<Z0 库> 的 Z0 开启
G59.<Z0 库>	6.<Z0 库> 的 Z0 开启
缩写形式: G54 bis G59	对于第 1 库对应的 Z0

数控函数，语法符合 DIN 66025（包括扩展）

其中	
<Z0 bank>	1...5, 整数

表格 5-50: 语法 G53 - G59

以下情况适用:

- G52.1, G53.1, G54.1 ... G59.1
都是模态，且可以相互取消选择。
G52.2, G53.2, G54.2 ... G59.2
都是模态，且可以相互取消选择。
G52.3, G53.3, G54.3 ... G59.3
都是模态，且可以相互取消选择。
G52.4, G53.4, G54.4 ... G59.4
都是模态，且可以相互取消选择。
G52.5, G53.5, G54.5 ... G59.5
都是模态，且可以相互取消选择。
G53 激活功能 G53.1 ... G53.5 和
由此取消对所有零点偏置的选择
- 如果只编程它们，则功能不会产生进给运动。只有当前的机床坐标系被偏置。
- 该功能也可以在一个程序块中用其他路径条件进行编程。在这种情况下，首先激活对应的零点偏置，然后进给到编程位置。
- 指定通道的 Z0 偏置（通道> 0）仅在指定通道中有效。从而可以为每个通道单独使用 Z0 表，激活同一个机床坐标标识符的不同零点。

示例: Z0 表"Z01"在 Z0 第一页中包含以下偏置值:

- 1. Z0: X100, Y100
- 2. Z0: X300, Y100
- 3. Z0: X500, Y100
- 4. Z0: X100, Y450
- 5. Z0: X300, Y450
- 6. Z0: X500, Y450

G54.1 至 G59.1 的功能效果:

数控函数，语法符合 DIN 66025（包括扩展）

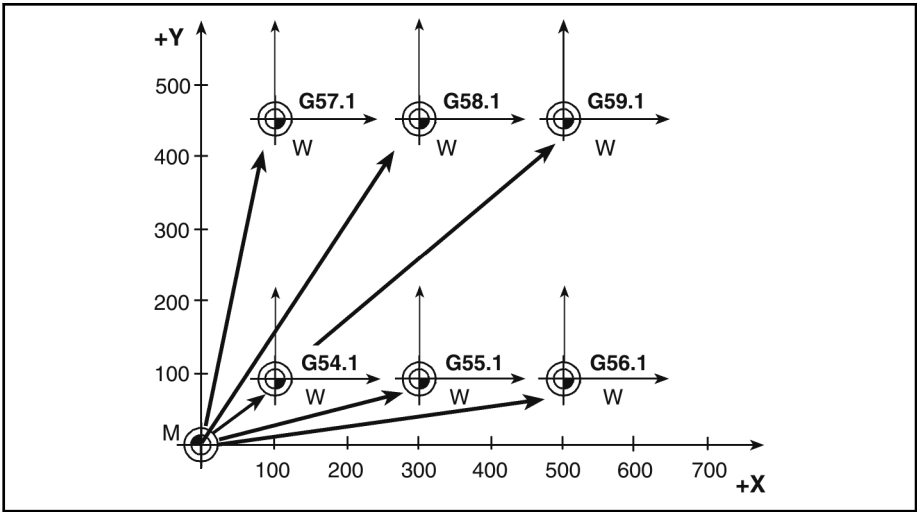


插图 5-24: 功能 G54.1 到 G59.1 的作用

不同 Z0 库的 Z0 叠加效果。

Z0 表“Z01”包含以下偏置值：

Z0 库 1, 1。Z0: X100, Y100

Z0 库 1, 2。Z0: X300, Y100

Z0 库 2, 4。Z0: X200, Y350

:	当前机床位置 0.0
:	无 Z0
:	所有后续的坐标都是机床坐标！
N40 ZOS (Z01)	激活 Z0 表“Z01”
N50 G54.1	无进给运动 激活偏置到 X100、Y100
N60 G55.1 X...Y...	到 X300、Y100 的偏置已激活 后无进给运动
N70 G57.2 X...Y...	激活偏置到到 X500、Y450 后无进给运动
N80 G53	关闭所有仍然有效的偏置

5.2.22 准停定位开启 / 关闭"G61、G62"

描述： 最大程度减少块终止处的跟踪错误。

在刀具运动过程中，由于机床的动力学原因，各个轴的目标和实际值之间会产生时间偏置。这种“滞后效果”在芯片生产时导致跟踪错误。错误大小取决于进给率和 C0 系数（轴动力学）。在轮廓过渡不稳定的情况下（角），由于角“精加工”，此跟踪错误变得明显。

以下情况适用：

- 块终止处的跟随距离（目标和实际值之间的位移）可降低到特定目标窗口大小。
- G61 只对进给运动有效。它对快速移动运动没有影响。
- 与 G1、G61（G0 例外）的精确停车功能相反，对所有的插补功能都有影像（即，还可适用于圆弧/螺线插补和样条插补）。
- G61 和 G62 都是以模态方式作用，而且相互删除。

数控函数，语法符合 DIN 66025（包括扩展）

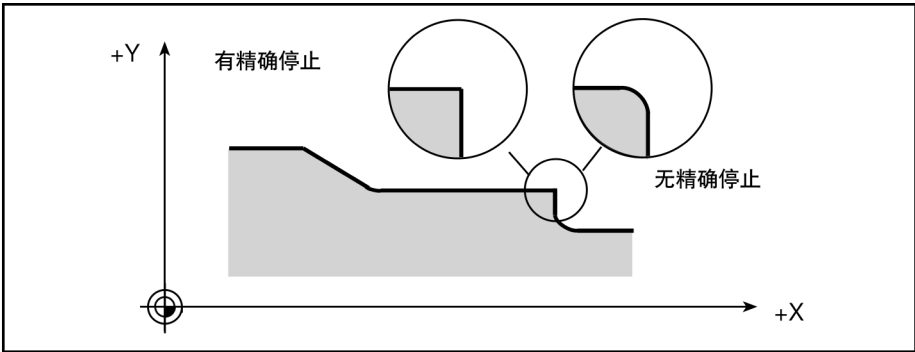


插图 5-25: 准确停止

 当 G61 有效时（准停定位开启），控制器始终在块终止处降速为 $v=0$ 。

语法：基本功能：

G61	准停定位开启
G62	准停定位关闭

表格 5-51: 语法 G61、G62

用于轮廓模式的最新激活定位窗口对于 “准停定位开启” 仍然有效。

默认：定位窗，精细

可选参数

用于选择进给模式的定位窗口。

G61 (IPS1 IPS2 IPS3)	
其中	
IPS1	准停定位开启 等待 精确定位窗口 。 在块终点，控制器首先把路径速度降低到 0。只有在涉及的所有轴都达到此定位窗口时，才会进给移动下一块。
IPS2	准停定位开启 等待 粗略定位窗口 。 通道接口显示“就位范围 2 激活”（还可参见 “《Rexroth IndraMotion MTX PLC 接口》”手册）。在块终点，控制器首先把路径速度降低到 0。只有在涉及的所有轴都达到此定位窗口时，才会进给移动下一块。
IPS3	准停定位开启 在块终止处减速到 $v=0$ 。 在块终止处，控制器将路径速度降低到 $v=0$ 。然后无需检查，把下一块移动到定位窗口。

表格 5-52: 可选参数 G61 语法

 参数“定位窗口，精细”和“定位窗口，粗糙”可在相 3 的 Sercos 文件中确定。
有关 Sercos 文件的更多信息，请参阅“Sercos 初始化”下的“IndraMotion MTX 机床参数”文档。

特殊功能和限制：最迟，G61 / G62 必须在要应用的块中编程。

数控函数，语法符合 DIN 66025（包括扩展）

请参见以下示例：

```
:
N40 G61                准停定位开启
:
N50 Y200              定位
:
N60 G62                准停定位关闭
:
N70 Y0                定位
:
N80 G61 Y200          准停定位开启时进行定位
:
```

5. 2. 23 无补偿卡盘的攻丝"G63"

- 描述：
- 将切换主轴到 C 轴的操作与钻削轴的线性插补同步。因此，不再需要补偿卡盘，因为它用于接受钻头轴和主轴之间的速度差。
 - G63 仅在已编程的块内有效。
 - 在攻丝过程中，可以在通道接口上输出信号（可使用机床参数在“有效功能”信号内配置）。
 - 在攻丝期间只有进给电位计起作用。

 在激活 G63 时，主轴 C 轴必须处于主轴模式。

 有关使用数个主轴攻丝的信息，另请参见 第 6.144 章 “无补偿夹具的攻丝的主轴选择 TappSp、TSP” 第 356 页。
如果为了计算进给需要隐藏轴，则请参见 第 6.43 章 “进给计算：隐藏轴 FeedAd、FAD” 第 237 页。

语法：	G63 (M<3 4> , S<Speed> H<Pitch> , ANG<StartingAngle>) <DrillAxis><InfeedDepth> {F<Feed>}
	G63.2 (M<3 4> , S<Speed> H<Pitch>) <DrillAxis><InfeedDepth> {F<Feed>}

表格 5-53: G63、G63.2 语法
起始角只能与 G63 一起编程。
对于 G63.2，主轴始终在 G63 前一块的结束位置开始启动。
螺距是从路径进给和速度之间的比率（F/S）中得出的。

 钻削和缩回过程中的不同螺纹节距会损坏工件 / 刀具！
因此在对钻削和缩回进行编程时，必须确保螺纹节距相同！

示例： 程序

```
N20 G0 X20 Y15 Z10 F1000      Positioning
;----- Programming the speed -----
;   the pitch is calculated by the control: H=F/S
N30 G63(M3,S500) Z-10 F500     Drilling (DrillAxis Z)
N40 G63(M4,S500) Z10 F500     Retract (DrillAxis Z)
N50 G0 Z20
;----- Programming the pitch -----
```

数控函数，语法符合 DIN 66025（包括扩展）

```
; the speed is calculated by the control: S=F/H
N60 G63(M3,H5) Z-10 F500           Drilling (DrillAxis Z)
N70 G63(M4,H5) Z10                 Retract (DrillAxis Z)
N80 G0 Z100
:
;----- Deep hole drilling-----
N120 G0 X20 Y15 Z10 F1000           Positioning for deep hole drilling
N130 G63(M3,S500) Z-20 F500         Drilling subsection 1
N140 G63(M4,S500) Z5 F500           Retract subsection 1
N150 G63(M3,S500) Z-40 F500        Drilling subsection 2
N160 G63(M4,S500) Z5 F500           Retract subsection 2
:
;----- Drilling with starting angle -----
N220 G0 X60 Y15 Z10                 Positioning
N230 G63(M3,S500,ANG60) Z-20        Drilling with starting angle 60 degrees
F500
N240 M257                           Message to PLC
N250 G63.2(M4,S500) Z5 F500         Retract with G63.2
:
```

- 特殊功能和限制:
- 应在钻削块之后直接对缩回块进行编程。
 - 如果没有其他的进给（地址 F）输入 G63 块，控制器将使用当前有效的路径进给。
 - 地址 “M” 和 “S” 只在已编程的 G63 块内生效。
 - G63 之前不需要“对正主轴”和“主轴停止”。
 - 切换到 C 轴模式会自动进行。
在启动之前，控制器会在内部等待所有相关轴的 “INPOS”。如果有一个轴移出 INPOS 范围，则 G63 就不会启动。
 - 在缩回块之后，主轴再次在主轴模式下自动运行。
 - 使用 CPL 表达式的 G63：
不允许采用 “G63（M[MCODE%]，S[DREHZ]）” 的形式进行编程。
相反地，编程 “G63（M[MCODE\$]，S[DREHZ]）”

5. 2. 24 英制编程 "G70"，本地英制编程 "I 数控 H(...)"

描述:	允许以英制单位指定路径 / 进给信息和加速度。	
语法:	G70	采用英制单位说明路径、进给和加速度方面的信息。
	<Axis>=I 数控 H(<Value>)	采用英制单位解释异步线性轴的路径信息。

表格 5-54: 语法 G70
以下情况适用:

- G70 以模态方式工作，并取消对 G71 的选择。
- 接通状态可在机床参数中进行指定。
- G70 也可在一块中用其他路径条件进行编程。
- G70 参考:
 - 进给距离
 - 进给
 - 加速度
 - 其他几何辅助变量（例如插补参数 I、J、K）。

示例:	<pre>N40 G70 在 N40（包含）处启动，所有路径和进给信息均采用英制单位解释。 : N50 W=I 数控 将异步线性轴 W 进给至位置 2（英制）。 H(2)</pre>

数控函数，语法符合 DIN 66025（包括扩展）

5.2.25 公制编程 "G71"

描述: 允许以公制单位指定路径/进给信息和加速度。.

语法:

G71	采用公制单位解释路径、进给和加速度方面的信息。

表格 5-55: 语法 G71

以下情况适用:

- G71 以模态方式工作，并取消对 G70 的选择。
- 接通状态可在机床参数中进行指定。
- G71 也可在一块中用其他路径条件进行编程。
- **G71 参考:**
 - 进给距离
 - 进给
 - 加速度
 - 其他几何辅助变量（例如插补参数 I、J、K）。

示例:

```
N40 G71 在 N40（包含）处启动，所有路径和进给信息均采用公制单位解释。  
:
```

5.2.26 接近参考点坐标 "G74"

描述: 在与 G74 相同的块中编程轴同时进给至它们的参考位置。
进给速度处于快速进给/进给模式中，取决于 G0/G1。

对于 G74 时，参考点凸轮和标记都不予以考虑。G74 仅仅定位到绝对轴位置（即它也适用于带距离编码编码器的轴）。



在此定位过程中可能不会考虑主动修正！

以下情况适用:

- G74 以块到块的方式作用，并在 G74 块中编程的所有机床轴都达到参考点后撤消。
- 在接近 G74 的参考点时，不会再次设置实际轴的值。
- 对于编程的轴，G74 程序段中不使用任何仍然有效的修正 Z0 等。

语法:

G74<AxisCoordinates>	开始“接近参考点坐标”
其中	
<AxisCoordinates>	必须与一个数字值（如 x1 y1 z1）一起对轴地址编程。此数字值对参考点位置没有作用。它只需要完成这个字。

表格 5-56: 语法 G74

示例:

```
N100 G74 X1 Y1 Z1 x 轴、y 轴和 z 轴同时开始行进到其参考位置并同时  
: 到达它们。
```

特殊功能和限制: 把 G74 与需要进给的轴一起编程在一个单独的块中。辅助和特殊功能也可在同块中进行编程。

5.2.27 接近参考点"G74 (HOME)"

描述:

	G74 (HOME) 功能也可用于异步轴！

数控函数，语法符合 DIN 66025（包括扩展）

使用零件程序触发编程轴的参考。
该功能针对编程轴的驱动器发出 Sercos 命令“驱动器控制的参考”（S-0-0148）。
结果，驱动器从控制器中分离出来并产生其参考的定位规格。驱动器使用 Sercos 参数 S-0-0147（参考参数）、S-0-0041（参考速度）和 S-0-0042（参考加速度）。



有关详细说明，请参阅文档“IndraMotion MTX 功能描述”。

- 以下情况适用：
- 如果在 G74（HOME）块中编程多个轴，则这些轴彼此独立地接近其参考点（无路径模式）。由此，参考点不能同时达到。
 - 同步轴和异步轴**都可以编程。块处理暂停，直到控制器的所有驱动器都确认它们各自的参考点已成功到达。
 - 该功能与操作模式“设置”中的“接近参考点”相同。

语法：	G74 (HOME) <AxisCoordinates>	
		触发已编程轴的 Sercos 命令“驱动器控制的参照”。
	其中	
	<AxisCoordinates>	必须与一个数字值（如 X1 Y1 Z1）一起对轴地址编程。此数字值对参考点位置没有作用。它只需要完成这个字。

表格 5-57: G74 (Home) 语法

示例：

```
N1 G74 (HOME) X1 Y1 将 Sercos 命令“驱动器控制的参照”发送到轴 X, Y 和
Z1 Z 的驱动器。
:
```

5. 2. 28 行进至探针"G75"

- 描述： 数控 具备两个 G75 功能：
- 向第一接触探头行进（G75.1 或 G75）
 - 向第二接触探头行进（G75.2）
- 两个 G75 功能：
- 行驶和测量，这些功能使用变频器的测头功能创建运动。
 - 他们逐块执行或者是本地功能。
 - 他们只使用线性插值。
 - 它们覆盖模块组 G0、G1、G2 ...，正好在它们编程的块中。
 - 他们自动停止块准备。WAIT 不必明确编程。

语法：	G75.1 <轴坐标> 或 G75 <轴坐标>	激活在测头 1 上反应的通道的所有测量轴并将行进轴移动到 <轴坐标>。
	G75.2 <AxisCoordinates>	激活在测头 2 上反应的通道的所有测量轴并将行进轴移动到 <轴坐标>。

表格 5-58: 语法 G75

示例：

```
N100 G75.2 Y250 F500;           以 F500 行进至位置 Y250
110 IF SD(9)=0 THEN;           如果第二接触探头已触发
```

数控函数，语法符合 DIN 66025（包括扩展）

```
120 YPOS=PCSPROBE("Y");      将 y 轴的测量值保存于 YPOS
:
N130 MSG(CONTACT)
140 ELSE
150 SETWARN ("NO CONTACT!")
N160 M0
170 WAIT
180 CLRWARN()
190 ENDIF
N200 ...
```

特殊功能和限制:

- 调试:
G75 功能编程前，必须先对其定制。所有要求的步骤参见“《Rexroth IndraMotion MTX 功能描述》”手册。
- 应至少使用一个轴坐标（即至少一个行进轴）对 G75 进行编程。
- 该坐标确定探头已须触发的最大搜索深度。
- 同一个数控 块中不允许有两个 G75 功能。数控 对此进行检查并造成错误。
- *G75 块之后，创建一个适当的 CPL 程序以*
 - 检查探头是否已触发；
使用 SD(9)
 - 在零件程序中运用测量值；
与 PCSPROBE () 或 PROBE ()
 - 生成所需警告和错误消息；
使用 SETWARN() 等。

5.2.29 接近固定的机床轴位置 "G76"

描述: 在不需要单独取消零件程序中的任何有效校正和转换情况下，允许编程并线性接近至机床坐标系统中的位置。

例如，在更换刀具，检查刀具是否损坏，测量周期或更换托盘时，这可能是必需的。

在快速进给 (G0) 或者进给模式 (G1) 中，并且具有 G93 (时间编程)、G94/G95 (进给编程) 和 F 字的情况下，G76 以块对块基准作用。

以下功能适用于 G76:

- 刀具修正 (G41、G42、G47 Dxx、EDxx)
- 输入帮助 (镜像、缩放、旋转、切换)
- 零点偏置 (G54.1...G59.5)
- 定位 (BcsCorr、G154.1 - G159.5)
- 以机床坐标为基础的轴变换，
例如 5 轴变换
- 相对尺寸编程 (G91)
- 设定程序位置 (SetPos)

数控函数，语法符合 DIN 66025（包括扩展）

语法:	G76<机床坐标>	
	其中	
	<机床坐标>	要接近的机床坐标

表格 5-59: G76 语法

- 特殊功能和限制:
- G76 可和其他路径条件（如 G0、G1、G93、G94、G95 和 F 字）一起写入。
 - 局部相对尺寸编程（"IC (...)" ;请参阅 第 5.2.32 章 "绝对尺寸编程 G90、相对尺寸编程 G91、局部绝对尺寸编程 AC(...)"、局部相对尺寸编程 IC(...)" 第 146 页 ）与 G76 一起无效并导致运行时错误。

5.2.30 异步子程序：重新定位各个坐标 "G77"

描述:

以下内容适用于异步子程序（Asup）：

用编程的坐标 G77 在其各自的调整位置（异步子程序之后的强制坐标位置）上进给。可选地，还可以指定在 G77 块中要进给坐标的调整位置的偏置。

坐标的当前位置也可以指定为调整位置。



有关异步子例程的使用和参数化的全面信息，请参阅文档 "IndraMotion MTX 功能描述"。

语法:	G77<Coord 1><Mode><Coord n><Mode>...F<Value>	
	其中	
	<Coord i>	坐标名称（如"x"）
	<Mode>	1: 将坐标进给至调整位置 0: 确定当前坐标位置到调整位置 =IC(<Offset>): 将坐标进给至调整位置值 + <Offset>
	<Value>	进给到返回轮廓点的位置。

表格 5-60: 语法 G77

- 特殊功能和限制:
- 该函数是为了在异步子例程中使用而设计的。
 - 所有不是通过 G77 编程的坐标都沿直线自动进给至异步子程序终止处的路径返回轮廓点。
 - 在异步子程序中进行的任何校正更改都将自动包含在轮廓所需返回点的内部计算中。
 - 作为替换语法，也可以采用 REPOS 来代替 G77 进行编程。

5.2.31 补偿切换打开 "G78"、补偿切换关闭"G79"

描述:

把几何修正功能的**长度修正**指定给各个坐标。

这些坐标包括

- 当前工件坐标系统（WCS）或
- 刀具坐标系统（TCS）的坐标。

只要刀具与当前工作平面垂直校准，而且在加工过程中，其相对于工作平面的方向保持不变，就能相对于**工件坐标系统（WCS）**的坐标进行赋值。

数控函数，语法符合 DIN 66025（包括扩展）

如果在加工过程中（例如自由曲面的铣削），刀具方向发生了改变，就需要对**刀具坐标系（TCS）**的坐标进行赋值。该修正需要一个已激活的轴变换（如 5 轴变换或 6 轴变换）。校正值在轴转换中确定。

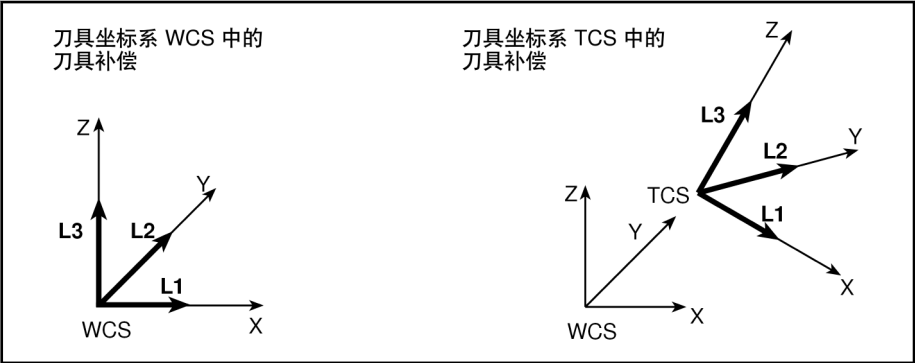


插图 5-26: 工件坐标系 WCS 和 TCS 中的刀具修正

 要激活当前加工修正值，参见 G47（第 5.2.19 章“刀具长度修正 G47、G48”第 132 页）。

语法:

G78 ({-}<coordinate1>), {-}<coordinate2>), {-}<coordinate3>))	
	修正切换打开
其中	
<Coordinate i>	要分配校正 L_i （其中 $i = 1, 2, 3$ ）的 WCS 坐标名称（逻辑 / 物理轴名称）。 如果修正值 L_i 将作用于刀具坐标系（TCS）中的坐标，则必须使用坐标识别符 XTR、YTR 和 ZTR。 可选用的负号：修正方向为负方向。 没有分配坐标的 L_i 修正值不予以考虑。
G78 (ActPlane)	修正值 L_1 、 L_2 和 L_3 分配给特定有效平面的主坐标、辅坐标和正常坐标。
G79	修正切换关闭 机床参数中的设定再次生效。

表格 5-61: 语法 G78、G79

示例:

G78 (X, , ZTR)	将 L_1 补偿分配给工件坐标系 (WCS) 的 x 轴，将 L_3 补偿分配给刀具坐标系 (TCS) 的 z 轴。两个校正值都是以正向计算的。
G78 (, , -Y)	L_3 修正被分配给工件坐标系统 (WCS) 的 Y 轴，并以负方向计算。
G78 (YA, YB)	L_1 修正值被分配给 YA 轴， L_2 修正值被分配给工件坐标系统 (WCS) 的 YB 轴。两个校正值都是以正向计算的。
G79	该通道所有默认坐标的补偿均关闭。

特殊功能和限制:

- G78 和 G79 都是以模态方式作用，而且相互取消。
- G79 中绝不会采用不属于通道默认设置的坐标！
- 在 G78 下编程的坐标地址可以参照工件坐标或者刀具坐标（请参见上述的 <coordinate i>）。
- 可以与其他路径条件、进给信息或者辅助功能一起对 G78/G79 编程。

数控函数，语法符合 DIN 66025（包括扩展）

5. 2. 32 绝对尺寸编程 "G90"、相对尺寸编程 "G91"、局部绝对尺寸编程 "AC(...)"、局部相对尺寸编程"IC(...)"

- 描述：指定控制器是按照绝对值还是相对（增量）值解释轴和坐标的单位信息。
- 绝对尺寸信息参照当前零点（程序坐标系统中的零点）。
 - 相对尺寸信息参照上一个接近的位置。

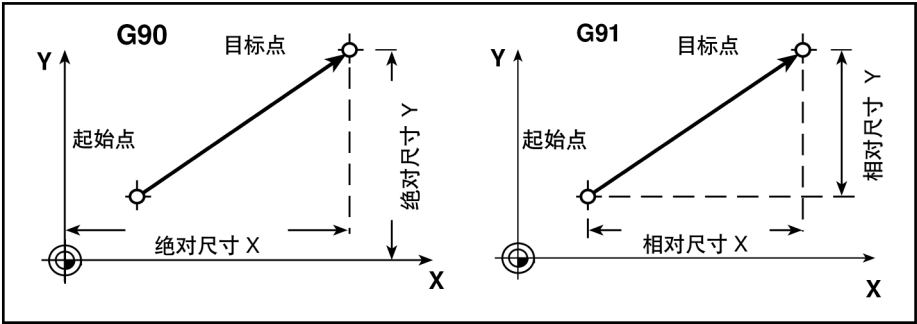


插图 5-27: 绝对尺寸编程 G90 和相对尺寸编程 G91

以下情况适用:

- G90 和 G91 都是以模态方式作用，而且相互取消。
- AC(...) 和 IC(...) 独立于有效的 G90/G91，只对轴的路径信息起作用，而这些轴的位置必须是用 AC(...) 或 IC(...) 编程。

语法：

G90	绝对尺寸编程
G91	相对尺寸编程
<Axis>=AC(<Value>)	局部绝对尺寸编程
<Axis>=IC(<Value>)	局部相对尺寸编程
其中	
<Axis>	逻辑轴地址
<Value>	<axis> 的路径信息

表格 5-62: 语法 G90、G91

示例：

N10 G1 G90	绝对尺寸编程打开
N20 X100 Y100	接近坐标 X100、Y100
N30 G91	相对尺寸编程打开
N40 X100 Y100	接近坐标 X200、Y200
N50 X=AC(50) Y50	x 轴的局部绝对尺寸编程 在机床上进给运动至 X50、Y250

5. 2. 33 时间编程 "G93"

- 描述：控制器根据模态变量 ITIM（参见 第 6.58 章 “逆时间编程 InvTime, ITIM” 第 253 页 ）的值解释 F-值，可以是以秒为单位的加工时间，也可以是以 1/分钟为单位的反时间或以 1/秒为单位编程的轮廓（块持续时间）。
- 默认情况下，周期以秒为单位（即 ITIM = 0）。
- ITIM>0 的单位来源于

数控函数，语法符合 DIN 66025（包括扩展）

$$F = \frac{v \left[\frac{mm}{ITIM \times sec} \right]}{s[mm]} = \frac{v \left[\frac{1}{min} \right]}{s} \Big|_{ITIM=60} = \frac{v \left[\frac{1}{sec} \right]}{s} \Big|_{ITIM=1}$$

这会导致以下单位适用于各个 ITIM 值：

ITIM 值	整数；在 G93 模式下，F 字的单位相对于 ITIM 值发生变化
0	F 值，单位为 [s]
1	F 值，单位为 [1/s]
60	F 值，单位为 [1/min]

以下情况适用：

功能 G93、G94 和 G95 以模态形式作用，并可相互取消选择。



所需的通电状态（G93(如果需要，ITIM)、G94 或 G95）可在机床参数中进行指定（默认值：G94，ITIM=0）。

语法：

G93	切换到时间或反向时间编程

表格 5-63: 语法 G93

示例：

```
N5 G93 G1 X30 Y20 F20 默认：ITIM 未编程或 ITIM = 0
: 编程的线性插补持续 20 秒。
N4 ITIM60 编程的线性插补持续 60/3 = 20 秒。
N5 G93 G1 X30 Y20 F3
:
N4 ITIM1 编程的线性插补持续 200 秒。
N5 G93 G1 X30 Y20 F5
:
```

特殊功能和限制：

- 切换到 G94 或者 G95 时，使用 G93 编程的 F 值仍保留在内部存储器中，并在切换回 G93 时再次激活。
- 在启动或者控制器复位之后，机床参数中指定的 F 字激活（默认值：F0）。
- 根据对应进给块的路径长度和编程的加工时间，控制器在内部计算出所需的进给。
但是，控制器可根据编程的轮廓和参与轴的最大速度对结果有效进给进行限制，从而增加块的持续时间。
- 取决于功能 OvrEna/OvrDis，即使在 G93 条件下，进给倍率仍然保持有效。

5.2.34 进给编程（每分钟）"G94"

描述： 控制器把 F 字（请参见 第 5.4.2 章 "F 地址 F" 第 174 页 和 第 5.4.4 章 "异步轴的速度 FA" 第 176 页 ）或 Omega 字（请参见 第 5.4.5 章 "Omega 地址（进给率）Omega" 第 176 页 ）解释为编程的轮廓进给。

以下情况适用：

- 功能 G93、G94 和 G95 以模态形式作用，并可相互取消选择。

数控函数，语法符合 DIN 66025（包括扩展）

- 以如下方式解释编程的进给：
 - G71 激活时解释为 mm/min
 - 如果 G70 激活，英寸/分
 - 与回转轴相关，度/分

机床参数可用来设定 G70/G71 进给的缩放。

所需的通电状态（G93、G94 或 G95）可在机床参数中进行指定（默认值：G94）。

语法：

G94	切换至进给编程

表格 5-64: 语法 G94

示例：

N10 G71	公制编程打开
N20 G1 G94 X20 Y30	以 200 mm/min 的进给行进
F200	
N30 G4 (F40)	停留时间 40 s。
N40 G70	英制单位编程打开
N60 X300 Y400	进给 F200（单位：mm/min）再次生效
N70 F100	新进给值：100 inch/min
:	

- 特殊功能和限制：
- 在切换到 G93 或者 G95 时，在 G94 期间编程的 F- 或 Omega 值仍保留在内部存储器中，并在切换并在切换回 G94 之后再次激活。
 - 在启动或者控制器复位之后，机床参数中指定的 F 字激活（默认值：F0）。
在启动之后，Omega 值总会自动初始化为“0”。
 - 由于已配置的参与轴的最大速度，控制器可对有效进给进行限制。
 - 根据 OvrEna/OvrDis，进给受到进给倍率的影响。

5. 2. 35 具有路径比例速度自适应的绝对速度编程 "G94（LINS）"

描述：

在默认情况下（无参数编程的 G94），如果存在新编程的 F 值，则会以最快的速度插补到新的进给值。考虑任何有效的加速度值。

在 G94（LINS）的情况下，块内的命令值更改与覆盖距离成比例。总是遵守有效的加速度值。如果可能的话，不管当前的倍率如何，新的指令速度都在块末端完全达到。

以下情况适用：

- G94（LINS）的作用是模态的，并且由 G94 取消选择 [默认情况（再次见上）]。
- 编程的进给不必与 G94（LINS）一起编程在一行中。
- 编程的进给模式和绝对适用于以下块。
- 根据 G70/G71，编程的进给单位是 inch/min 或 mm/min。
- 根据 OvrEna/OvrDis，进给受到进给倍率的影响。

语法：

G94 (LINS)	激活直接供稿编程。以下编程的 F 值会导致与覆盖距离成比例的指令速度变化。

表格 5-65: 语法 G94 (LINS)

数控函数，语法符合 DIN 66025（包括扩展）

示例：	N30 G94 G1 G71 F1000 X250 Y300	块以 1000 mm/min 的进给值进给
	:	
	N50 G94 (LINS) F2000 X300 Y200	命令路径的进给量与距离 2000mm/min 到块端的距离成比例地增加
	:	
	N60 X350	块以 2000 mm/min 的进给值进给
	:	
	N100 F500 X450	命令路径进给与 500 mm/min 到块终止的距离成比例地减少
	N110 G94	G94 在默认情况下：速度的变化以尽可能快的速度插入，以符合有效的加速度值

- 特殊功能和限制：
- 对现有的加速度和减速限制进行监控。因此只能在下一个块中达到最终的速度。
 - 编程较高的速度值会导致编程速度的模态值在当前程序段中增加。但较低速度编程会导致编程速度的模态值在以下块数控中减少。由此，在两种情况下都可以确保，如果由于有效的加速度值，允许在当前块终止数控时分别达到要求的速度。

5. 2. 36 具有时间比例速度自适应的绝对速度编程 "G94 (LINS) "

描述： 在默认情况下（无参数编程的 G94），如果存在新编程的 F 值，则会以最快速度插补到新的进给值。考虑任何有效的加速度值。

在 G94 (LINT) 的情况下，块内的命令值更改结果与时间成比例。总是遵守有效的加速度值。如果可能的话，不管当前的倍率如何，新的指令速度都在块末端完全达到。特别是软速度转换结果（可以与 G94 (DF ...) 比较）。

以下情况适用：

- G94 (LINT) 的效果是模态的，并且由 G94 取消选定 [默认情况（见上）再次生效]。
- 编程的进给不必与 G94 (LINT) 一起编程在一行中。
- 编程的进给模式和绝对适用于以下块。
- 根据 G70/G71，编程的进给单位是 inch/min 或 mm/min。
- 根据 OvrEna/OvrDis，进给受到进给倍率的影响。

语法：	G94 (LINT)	激活直接供稿编程。以下编程的 F-值会导致与时间成比例的指令速度变化。

表格 5-66: G94 (LINT) 语法

示例：	N30 G94 G1 G71 F1000 X250 Y300	块以 1000 mm/min 的进给值进给
	:	
	N50 G94 (LINT) F2000 X300 Y200	命令路径进给与 2000mm/min 到块终止的时间成比例地增加
	:	
	N60 X350	块以 2000 mm/min 的进给值进给
	:	

数控函数，语法符合 DIN 66025（包括扩展）

N100 F500 X450	命令路径进给与 500 mm/min 到块终止的时间成比例地减少
N110 G94	G94 在默认情况下：速度的变化以尽可能快的速度插入，以符合有效的加速度值

- 特殊功能和限制：
- 对现有的加速度和减速限制进行监控。因此只能在下一个块中达到最终的速度。
 - 编程较高的速度值会导致编程速度的模态值在当前程序段中增加。但较低速度编程会导致编程速度的模态值在**以下块数控**中减少。由此，在两种情况下都可以确保，如果由于有效的加速度值，允许在**当前块终止数控**时分别达到要求的速度。

5. 2. 37 增量速度编程 "G94(...)" 加速适应条件下

描述：对于上一个激活的有效值改变进给或速度。G94 (...) 块中的加速度适用于确保在块终止前未达到结果速度或速度。这导致非常修匀的加速行为。

以下情况适用：

- 模块末尾的结果适用于以下模块。
- 根据 G70/G71，增量进给单位是 inch/min 或 mm/min。
- 根据 OvrEna/OvrDis，进给受到进给倍率的影响。

语法：

G94 (DF<F-value>)	程序路径速度（方便差 进给 ）增量
G94 (DS1= <S-value>)	为主轴 1 增量编程主轴速度（方便差 速度 ）
G94 (DF <F-value> ,DS7= <S-value>)	
	以增量方式编程路径速度和主轴 7 的主轴速度
G94 (DSA2=<Axis>,DS2=<S-value>)	
	增量式编程无限轴<Axis>（方便差 速度轴 ）的速度
其中	
<F-value>	增量路径速度 正值会增加当前有效的路径速度，而负值会将之减小。
<S-value>	基于轴分配，S 值表示： - 具有相同索引的通道主轴的增量通道主轴转速 DSi（无轴规格） - 具有相同索引的无限轴 DSAi 的增量速度 DSi - 具有相同指数的馈送耦合环形轴 DSAi 的增量耦合系数 DSi（也参见 AFC） 增加正面价值。负值会降低当前有效的主轴转速。最多 8 个 S 值（DS1 ... DS8）可以同时编程。 无限轴的增量速度编程会导致方向反转。
<Axis>	在速度模式下代表异步无限轴。 对于 8 个及 8 个以下的轴（DSA1 ... DSA8），可同时指定增量 S 值。

表格 5-67: G94 语法 (F 值 / S 值)

数控函数，语法符合 DIN 66025（包括扩展）

示例：		
	N30 G94 (DF100) X250 Y300	在块终止之前，以线性方式按 100 mm/min 增加路径进给
	:	
	N50 G94 (DF=50) X300 Y200	在块终止之前，以线性方式按 50 mm/min 增加路径进给
	:	
	N70 G94 (DS1=100) X25 Y30	在块终止之前，把主轴 1 的速度按 100 转增加
	:	
	在块终止之前，以线性方式按 100 mm/min 增加路径速度，按 150 转/分增加主轴 2 的速度	
	N90 G94 (DF100, DS7=150) X2 Y2	
	:	
	N100 B2=S (100) B3=S (50)	在速度模式下启动无限轴 B2 和 B3
	B2 的速度提高到 200 转	
	B3 的速度减少到 30 转	
	直至块终止点	
	N110 G94 (DSA1=B2, DS1=100, DSA2=B3, DS2=-20) X10	
	:	
	N120 B1=S (500)	以 500 rpm 启动无限轴 B1
	与...连接	
	0.5 转 B2/B1 转	
	N130 AFC (FRA=B1, FA=B2, CF=0.5) B2=S (1)	
	增加到	
	0.8 B2 旋转 / B1 旋转直到块结束	
	N140 G94 (DSA1=B2, DS1=0.3) X10 F1000	

- 特殊功能和限制：
- 对现有的加速度和减速限制进行监控。因此只能在下一个块中达到最终的速度。
 - 计算出的加速度只在 G94 (...) 块中有效。如果该块被取消，将以计算出的加速度进行减速。
 - 绝对和增量进给不得同时在同一块中进行编程。
 - 正值编程的速度会导致编程速度的模态值在当前块中以各自的量增加。但负值编程的速度会导致编程速度的模态值在**以下数控块**中以各自的量减少。调整后的加速度确保在两种情况下，总是在**当前数控程序段结束时**达到最终的实际速度值。
 - 如果程序段先前被中止或缺少位置数据，控制器会直接激活增量速度值

5. 2. 38 绝对速度编程 "G94 (...) " 加速适应条件下

描述：

在多个块之间将进给从当前值连续改变为目标进给。对目标块（新目标进给或完成斜坡）之前的加速度进行调整，使得仅在目标块达到结果速度。这导致非常修匀的加速行为。

数控函数，语法符合 DIN 66025（包括扩展）

相应地，多达八个通道主轴的速度规格可以连续变化，并且可以从当前速度到目标速度的几个数据块进行路径切换。

以下情况适用:

- 尽管 G94 功能是模态的，但是结果进给对于以下块来说并非是模态的。
- 根据 G70/G71，目标进给单位是 inch/min 或 mm/min。
- 根据 OvrEna/OvrDis，进给受到进给倍率的影响。在大多数情况下，100% 的倍率设置不会提供连续的斜坡。

语法:	G94 (TF<F value>)	以绝对方式编写目标路径速度
	G94 (TS1= <S-value>)	以绝对方式编写主轴 1 的目标主轴速度
	G94 (TF <F-value> , TS7 <S-value>)	
		编写目标路径速度和主轴 7 的目标主轴速度
	G94 (TB)	目标进给和目标速度的完成斜坡
	其中	
	<F-value>	目标路径速度
	<S-value>	目标主轴速度

表格 5-68: G94 语法 (TF 值 / TS 值)

示例:	N10 G94 F1000 M3 S1=1000	常量路径进给和常量速度。
	:	
	N30 G94 (TF1500) X50 Y30	将路径进给从 F1000 线性增加至 F1500，直至块 N50。
	:	
	N50 G94 (TF500,TS1=2000) X70 Y50	将路径进给从 F1500 线性增加至 F500，直至块 N70。同时，主轴 1 的速度从 1000 rpm 持续增加到 2000 rpm，直到块 N70。
	:	
	N70 G94 (TB)X90 Y70	从目标块开始，进给为 F500，主轴 1 速度为 2000 rpm。
	:	

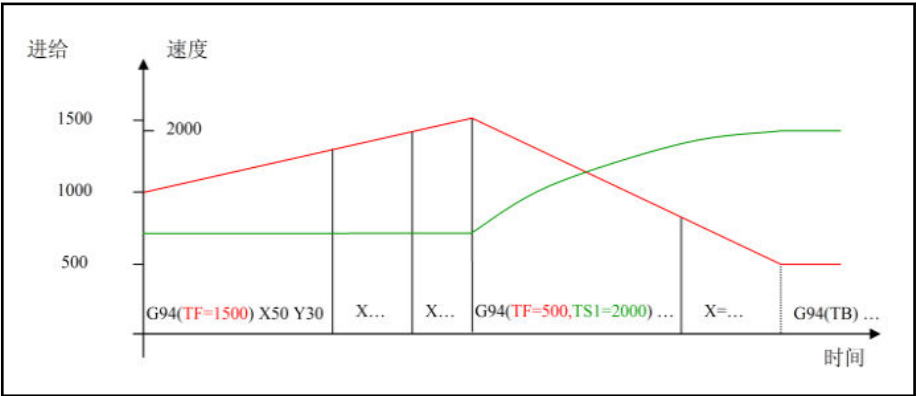


插图 5-28: 连续加速和减速

- 特殊功能和限制:
- 对现有的加速度和减速限制进行监控。因此只能在下一个块中达到最终的速度。
 - 计算出的加速度只在 G94(...) 块中有效。如果该块被取消，将以最大可能加速度进行减速。
 - 在目标进给顺序 (TF-TB) 中，不得编程一个增量进给 (DF) 也不得编程模态进给 (F 字)。

数控函数，语法符合 DIN 66025（包括扩展）

- 在目标速度顺序（TS-TB）中，不得编程一个增量速度（DS）也不得编程模态主轴速度（S 字）。
- 最大可能目标进给 / 目标速度块的数量由块预览参数限制。
数控 0/LookAh/Ch[k]/MaxBlk/MaxBlkMiscFunc 和
数控 0/LookAh/Ch[k]/PercSplit/PercMiscFunc。
- 仅可对于直接主轴编程进行目标速度指定（G97）。

5. 2. 39 进给编程（每转）"G95"

描述: 控制器将 F 字（参见 第 5.4.2 章 "F 地址 F" 第 174 页）解释为进给 / 转。在结合主轴使用时需要用到该特性。

以下情况适用:

- 功能 G93、G94 和 G95 以模态形式作用，并可相互取消选择。
- 以如下方式解释编程的进给：
 - G71 激活时为 mm/转
 - G70 激活时为 inch/转



- 机床参数可用来设定 G70/G71 进给的缩放。
- 所需的通电状态（G93、G94 或 G95）可在机床参数中进行指定（默认值：G94）。
- 关于主轴的定义，请参见 MP 7020 00010 或"MainSp"功能。

语法:

G95	切换到旋转进给编程

表格 5-69: 语法 G95

示例:

N05 G71	公制编程打开
N10 S200 M4	主轴速度 200 rpm，逆时针
N20 G1 G95 X20 Z30 F0.2	以 0.2 mm/min 的进给行进
N30 G4(S20)	停留时间 20 转
N40 G70	英制单位编程打开
N60 X300 Z40	进给 F0.2（单位：mm/转）再次生效
N70 F0.1	新进给 = 0.1 英寸/转
:	

特殊功能和限制:

- G95 需要一个车削主轴。
- 有效进给会受主轴和进给电位计的影响。
- 由于已配置的参与轴的最大速度，控制器可对有效进给进行限制。
- 在切换到 G93 或者 G94 时，在 G95 期间编程的 F 字仍保留在内存中，并在切换回 G95 之后再次激活。
- 在启动或者控制器复位之后，机床参数中指定的 F 字激活（默认值：F0）。

5. 2. 40 增量速度编程"G95 (...)"带加速度调节

描述: 在 G95 (...) 块中加速度适用于确保在块终止前没有达到结果速度或速度。这导致非常修匀的加速行为。

以下情况适用:

- 模块末尾的结果适用于以下模块。

数控函数，语法符合 DIN 66025（包括扩展）

- 根据 G70/G71，目标进给的单位是 inch/min 和 mm/min。
- 根据 OvrEna/OvrDis，进给受到进给倍率的影响。

语法:

G95 (DF<F-value>)	程序路径速度（方便差进给）增量
G95 (DS1=<S-value>)	为主轴 1 增量编程主轴速度（方便差速度）
G95 (DF<F-value>, DS7=<S-value>)	
	以增量方式编程路径速度和主轴 7 的主轴速度
G95 (DSA2=<Axis>, DS2=<S-value>)	
	增量式编程无限轴（Delta 速度轴）的速度
其中	
<F-value>	增量路径速度，单位为毫米 / 转或英寸 / 转 正值会增加当前有效的路径速度，而负值会将之减小。
<S-value>	基于轴分配，S 值表示： • 具有相同索引的通道主轴的增量速度 DSi（无轴规格） • 具有相同索引的无限轴 DSAi 的增量速度 DSi • 具有相同指数的馈送耦合环形轴 DSAi 的增量耦合系数 DSi（也参见 AFC） 正值会增加当前有效的主轴速度，而负值会将之减小。最多 8 个 S 值（DS1 ... DS8）可以同时编程。 无限轴的增量速度编程会导致方向反转。
<Axis>	在速度模式下代表异步无限轴。 对于 8 个及 8 个以下的轴（DSA1 ... DSA8），可同时指定增量 S 值。

表格 5-70: 使用 G95 的增量速度编程语法

示例:

N10 G95 (DF=-0.5) G1 X100	； 递减递减 0.5 毫米 / 转至 至块终点的距离
...	
N20 M3 S1=500	； 以 500 rpm 的转速启动通道 S1
N30 G95 (DS1=200) X10 F0.2	； 将 S1 的速度提高到 700 转 / 分钟，直到阻止 结束
；	
...	
N70 B2=S (100) B3=S (50)	； 启动无限轴 B2 和 B3 启动编辑模式
； B2 的速度提高到 200 转	
； B3 的速度提高到 30 转	
块终止点	
N80 G95 (DSA1=B2, DS1=100, DSA2=B3, DS2=-20) X10	

数控函数，语法符合 DIN 66025（包括扩展）

```
...  
N100 B1=S(500)           ; 以 500 rpm 的转速启动无限轴 B1  
N110 G1 G95(FRA=B1)F0.3  ; 进给参考: 0.3 毫米 / B1 转  
  
                        ;进给联轴器: 0.5 B2 旋转/  
                        ; B1 旋转  
N120 AFC(FA=B2,CF=0.5) B2=S(1)  
  
                        ; 将 F 增加到 0.4 毫米 / B1 的旋转和  
                        ; CF 至 0.7 B2 旋转 / B1 旋转  
                        ; 块终止点  
N130 G95(DF=0.1,DSA1=B2,DS1=0.2)X10
```

- 特殊功能和限制:
- 对现有的加速度和减速限制进行监控。因此只能在下一个块中达到最终的速度。
 - 仅能使用 100% 倍率实现 G95 (DF...) 块中的常量加速度。
 - SPLIT 命令形成每个块段中的编定进给坡度。
 - 如果程序段先前被中止或缺少位置数据，控制器会直接激活增量速度值

5.2.41 回转轴每转进给编程 "G95(FRA=<Axis>)"

- 描述:
- 建立相对系统回转轴或线性循环轴（mm 或 inch / 模态区间）进给参考（单位为 mm 或英寸 / 360°），并改变相对上一次有效的实际值的进给。回转轴参考保持模态。
- 以下情况适用:
- 开启后，将使用 G95 建立主轴参考。
 - 编程的参考轴功能 G95 是模态的。
 - 根据 OvrEna/OvrDis，进给受到进给倍率的影响。

语法:

G95(FRA=<SystemAxisName>)	
	建立相对循环轴的进给参考
G95(FRA=MSP)	建立相对主轴的进给参考 (MainSp)

表格 5-71: G95 回转轴每转进给编程语法

- 示例:
- ```
N10 Z3=S(1000) ; Z3 速度规格 1000rpm
N30 G95(FRA=Z3)F1.0 ; 相对转动轴 Z3 的进给参考
N40 G1 X100 ; 进给: 1000 mm/min at 100% OVR
N50 G95(FRA=MSP) ; 对于有效主轴的进给
```

- 特殊功能和限制:
- 循环轴或当前参考主轴 (MainSp) 仅可用作参考轴 (FRA)。
  - 参考轴不能是通道轴。

5.2.42 常量切削速度 "G96"、直接速度编程 "G97"

- 描述:
- 车削过程中，控制器会将 S 字解释为
- 刀具常量切削速度 (G96) 或
  - 转动工件轴的速度 (G97)。

数控函数，语法符合 DIN 66025（包括扩展）

在给定的主轴速下，切削速度取决于切削刀具和工件回转轴之间的距离。  
为了补偿该速度，功能 G96 会根据切削刀具和工件回转轴之间的距离自动改变回转轴速度：

- 对于 G71:

$$S_{act} \left[ \text{min}^{-1} \right] = \frac{S_{G96} \times 1000}{2 \times \pi \times \left| \text{位置}(\text{坐标, 参照系}) \text{刀具补偿} \right| \left[ \frac{\text{m} \times 1000}{\text{min} \times \text{mm}} \right]}$$

插图 5-29:      G71

- 对于 G70:

$$\left[ \frac{\text{feet} \times 12}{\text{min} \times \text{inch}} \right]$$

插图 5-30:      G70

默认情况下，数控 计算机床坐标系（MCS 参考系）中的距离：

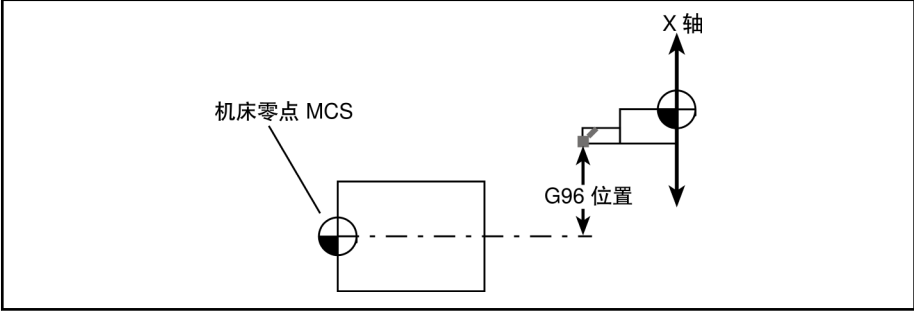


插图 5-31:      参考系 MCS

如果主轴必须位于不同的刀库刀具安装点之间，G96 可以包括涵盖局部坐标系（LCS 参考系）的相应零点偏置：

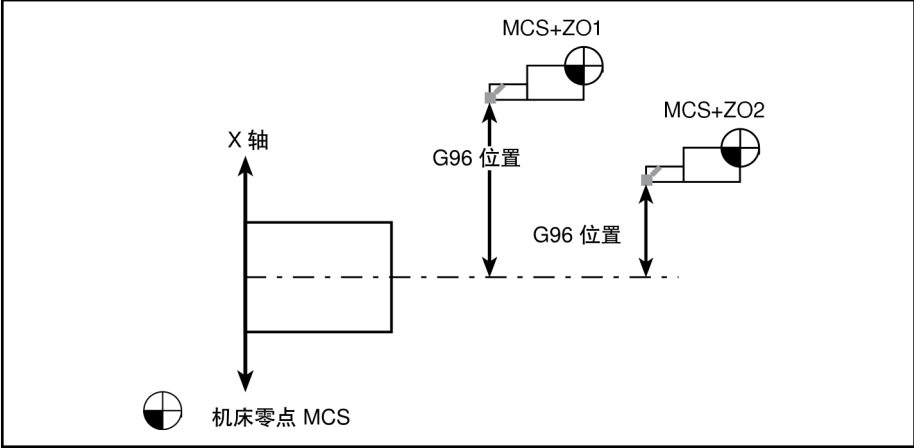


插图 5-32:      参考系 LCS

如果刀具的排布方式相对于主轴并非轴平行，工件零点的偏置量和位置也可以考虑在内（PCS 参考系）：

数控函数，语法符合 DIN 66025（包括扩展）

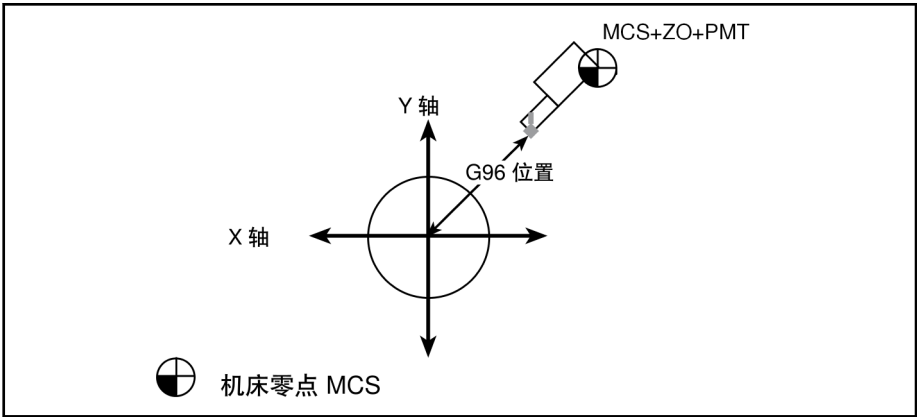


插图 5-33: 参考点 PCS

从而工件的切削速度也就不再受切削刀具和工件回转轴之间距离的影响。  
如果不需要这种操作特性，请使用 G97。工件回转轴的速度只由编程的 S 字决定。

以下情况适用：

- 切削刀具头和工件回转轴之间距离的参考轴由机床参数 7010 00110（默认值）定义。
- 对参考轴的编程可在运行时间更改。
- 参考轴的参考坐标系（坐标变换点）通过机床参数 7010 00120 预设。

可显示以下内容：

- PCS：  
程序坐标系中的位置
- LCS：  
局部机床坐标系中的位置（由 ZO 开始）
- MCS：  
机床坐标系中的位置（默认）
- 参考轴的坐标变换点可在运行时间更改。
- 编程的切削速度解释为
  - G71 激活时解释为 m/min
  - G70 激活时解释为英尺/min（1 英尺 = 12 英寸）。
- 除主轴倍率外，速度限制功能（SMin、SMax；参见 第 6.115 章“速度限制 SMin、SMN， SMax、SMX” 第 324 页）也起作用。

语法：

|                                                                    |                                                             |
|--------------------------------------------------------------------|-------------------------------------------------------------|
| G96 { ( { <ReferenceAxis> { , <PointOfAction> } } ) } ... S<i>=<V> |                                                             |
|                                                                    | 在 G96 块中编程的主轴的 S 字被解释为旋转期间刀具的目标切削速度。<br>其他所有主轴回复为直接速度编程！    |
| G96                                                                | 最新的编程设置激活。如果未设置，则使用默认值。                                     |
| G96 ( ) ...                                                        | 参考轴的默认值和坐标变换点均重新激活。                                         |
| G97 ... S<i>= <Speed>                                              |                                                             |
|                                                                    | 只有其 S 字在 G97 块中编程的主轴才会回复为直接速度编程。对于这些主轴，编程的 S 字被解释为工件回转轴的速度。 |
| G97                                                                | 所有主轴回复为直接速度编程。                                              |

数控函数，语法符合 DIN 66025（包括扩展）

|                    |                                                       |
|--------------------|-------------------------------------------------------|
| 其中                 |                                                       |
| <Reference axis>   | 参考轴的物理或逻辑名称。在设定新值或激活默认值之前，编程会一直保持有效。                  |
| <Points of action> | 参考轴的坐标系：<br>PCS、LCS 或 MCS<br>在设定新值或激活默认值之前，编程会一直保持有效。 |
| <i>                | 主轴索引。<br>1：第一主轴<br>2：第二主轴，等。                          |
| <V>                | 相关主轴的切削速度（单位：m/min 或英尺 /min）。                         |
| <Speed>            | 相关主轴的速度（单位：转 / 分）。                                    |

表格 5-72: G96、G97 语法

示例：（系统中配置 2 根主轴）

|                         |                             |
|-------------------------|-----------------------------|
| N10 G71 S1=500 S2=500   | 在程序中首次定义两根主轴的目标转数。          |
| :                       |                             |
| N60 G96 S1=50           | 为第一主轴启用 G96，速度步长为 50 m/min。 |
| :                       | 2. 主轴仍然按照速度编程运行             |
| :                       | G96 激活                      |
| :                       |                             |
| N100 S1=30              | 将第一主轴的切削速度降低到 30 m/min。     |
| :                       | 第二主轴仍然按照速度编程运行              |
| :                       | G96 激活                      |
| :                       |                             |
| N140 G96 (Y,LCS) S2=100 | 参考轴为 Y，坐标变换点是局部机床坐标系 LCS。   |
| :                       | 启用第二主轴的切削速度 100 m/min       |
| :                       | 第一主轴回退到速度编程状态               |
| :                       | G96 激活                      |
| :                       |                             |
| :                       |                             |
| N180 G97                | 如果尚未出现这种情况，则所有的主轴回复为直接速度编程。 |
| :                       |                             |
| :                       | G97 激活                      |

- 特殊功能和限制：
- 可以通过对序列中相应的 S 字进行编程，在同一个块中一起切换几根主轴（例如：G96 S1=100 S2=1000）。
  - 要更改已经切换到 G96 的主轴的切削速度以执行程序剩余部分，只需对相关主轴的 S 字进行重新编程。
  - 切换到 G97 后，当前有效的主轴切削速度保存在内存中。一旦相应主轴“切换回”G96，它就会重新激活。
  - 从 G96 切换到 G97 时，控制器使用当前速度作为所有主轴（其 S 字未在 G97 块中编程）的新目标速度值。
  - 激活 G96 前，可能需要换档。

数控函数，语法符合 DIN 66025（包括扩展）

- G96 激活时，“自动齿轮档切换”功能不会切换任何齿轮档。
- 如果 G96 激活，控制器会检查参考轴是否会“影响”有效坐标变换点。如果并非如此（例如，如果轴退出通道），则会报告一个运行时错误。

### 5.2.43 3D 刀具半径补偿 "G140、G141、G142"

**描述：** 功能会将一个旋转对称刀具移动至编程路径的左侧或右侧，通过进给激活将其坐标变换点引导至编程路径。

如果是纯粹的方向运动，坐标变换点的当前位置会保持不变，TCP 会移动。

在外部角上，使用自动生成的中间块（弧）关闭路径。内角处计算交点。

**控制器考虑更正：**

- 有效刀具刃口 (ED、DCT) 以及有效 D 补偿 (D、DcTSe1、DCS、DCT) 的刀具半径 "r"
- 可选插入深度 "d"（语法参见 INSDEP）
- 路径的可选偏差，叠加作用于刀具半径（请参见 COFFS 的语法）
- 公差的可选变化率（语法参见 DCOFFS）

下图显示了原理：

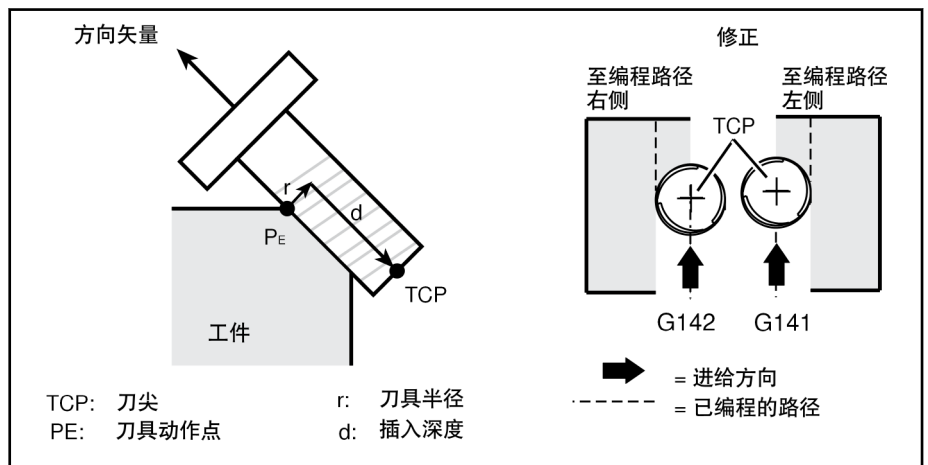


插图 5-34: 3D 刀具半径修正

G141/G142 激活时，插入深度 (INSDEP)、公差 (COFFS)、刀具长度或铣削刀具半径的变动会立即以柔缓方式执行（使用主轴）。

公差的变化率 (DCOFFS) 仅在 G141/G142 有效时有效，但在补偿构建块中无效。变化率的变化立即生效。变化率的变化将立即并在在进给块期间生效，公差将线性地从块开始的起始值到  $\langle \text{Start value} \rangle + \langle \text{Length of the programmed path} \rangle * \langle \text{Change rate of the allowance} \rangle$  给定的块的结束值。在下一个块中，结束值变为起始值，除非其被下一块中的 COFFS 编程覆盖。



有关轴转换功能性的详细信息，请参阅文档“IndraMotion MTX 功能描述”。



无需编程单独移动运动，就可直接激活或停用修正值。这会损坏工件或刀具。

因此，请观察本章和“IndraMotion MTX 功能描述”文档中的所有信息。

数控函数，语法符合 DIN 66025（包括扩展）

|     |            |                                                                                                                                                                                                                                                                     |
|-----|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | G140       | 3D 刀具半径补偿关闭<br><br>(控制启动后的接通状态)<br><br>通过插补方式减小补偿（半径、插入深度），如有必要可使用已编程的进给运动（应用 3 次样条）。有效的刀具长度补偿不会受到 G140 的影响。                                                                                                                                                        |
|     | G141       | <b>路径左侧的</b> 3D 刀具半径补偿打开（可从加工方向看到，并且具有正的补偿值）。<br><br>可在同一块中对进给运动或定位运动进行编程。<br><br>如果在同一块中对激活空间坐标进行编程，则将会缩进半径和插入深度。<br><br><i>否则，控制器构建补偿</i> <ul style="list-style-type: none"><li>只有在 G140 先前有效时才与下一个运行程序段一起运行</li><li>如果预先激活 G141/G142，则立即进行补偿（在激活的进给中）。</li></ul> |
|     | G142       | <b>路径右侧的</b> 3D 刀具半径补偿打开<br><br>否则与 G141 相同                                                                                                                                                                                                                         |
|     | INSDEP<ET> | 定义插入深度 <ET><br><br>起模态作用<br><br>控制启动后的接通状态：0                                                                                                                                                                                                                        |
|     | COFFS<KA>  | 定义公差值 <CA><br><br>以模态形式叠加作用于刀具半径 "r"<br><br>控制启动后的接通状态：0                                                                                                                                                                                                            |
|     | DCOFFS<DA> | 定义公差的变化率 <DA><br><br>起模态作用并沿路径连续改变公差<br><br>控制启动后的接通状态：0                                                                                                                                                                                                            |

表格 5-73: 语法 G140 - G142

- 特殊功能和限制:
- G140、G141 和 G142 都是以模态方式作用，而且相互取消
  - 激活以及取消对任何轮廓（G2、G6 等）的补偿
  - 在 G141 或者 G142 有效时，**不允许**进行以下操作：
    - G17 ... G20（平面切换）
    - G70、G71（英制 / 公制切换）
    - G63（没有进行卡盘补偿的攻丝）
    - G74（接近参考点坐标）
    - G75（前往探测头）
    - MOB（在块上测量）
    - G76（靠近固定的机床轴位置）
    - SetPos（设定程序位置）
    - G54. x ... G59. x（零点偏置）
    - G154. x ... G159. x（斜面）
    - G40 ... G42（刀具半径修正）
    - Coord()（轴变换的打开/关闭切换）
    - 作为当前空间坐标一部分的轴的传送，借助轴 / 坐标转换进行。

数控函数，语法符合 DIN 66025（包括扩展）

- 如果 G141 或 G142 激活，则不会执行碰撞检查。
- 如果程序不是通过 M30 结束，则 G140 在此时必须激活。目前的切槽深度，容差及其变化率已保留。
- 控制器、系统复位或采用 M30 后 G140 会自动激活。目前的切槽深度，容差及其变化率设为 0。

## 5.2.44 定位：斜面 "G151 - G159.5"

**描述：** 定位“斜面”可在空间的任何位置切换和定向工件坐标系。斜面将通过相应通道中“X”、“Y”和“Z”的含义影响坐标。



轴名“X”、“Y”以及“Z”将在下面用于指示含义为 X、Y 和 Z 的轴。

由于存在 3 个定向自由度，因此每个定向都必须用 3 个连续的基础旋转来表达。

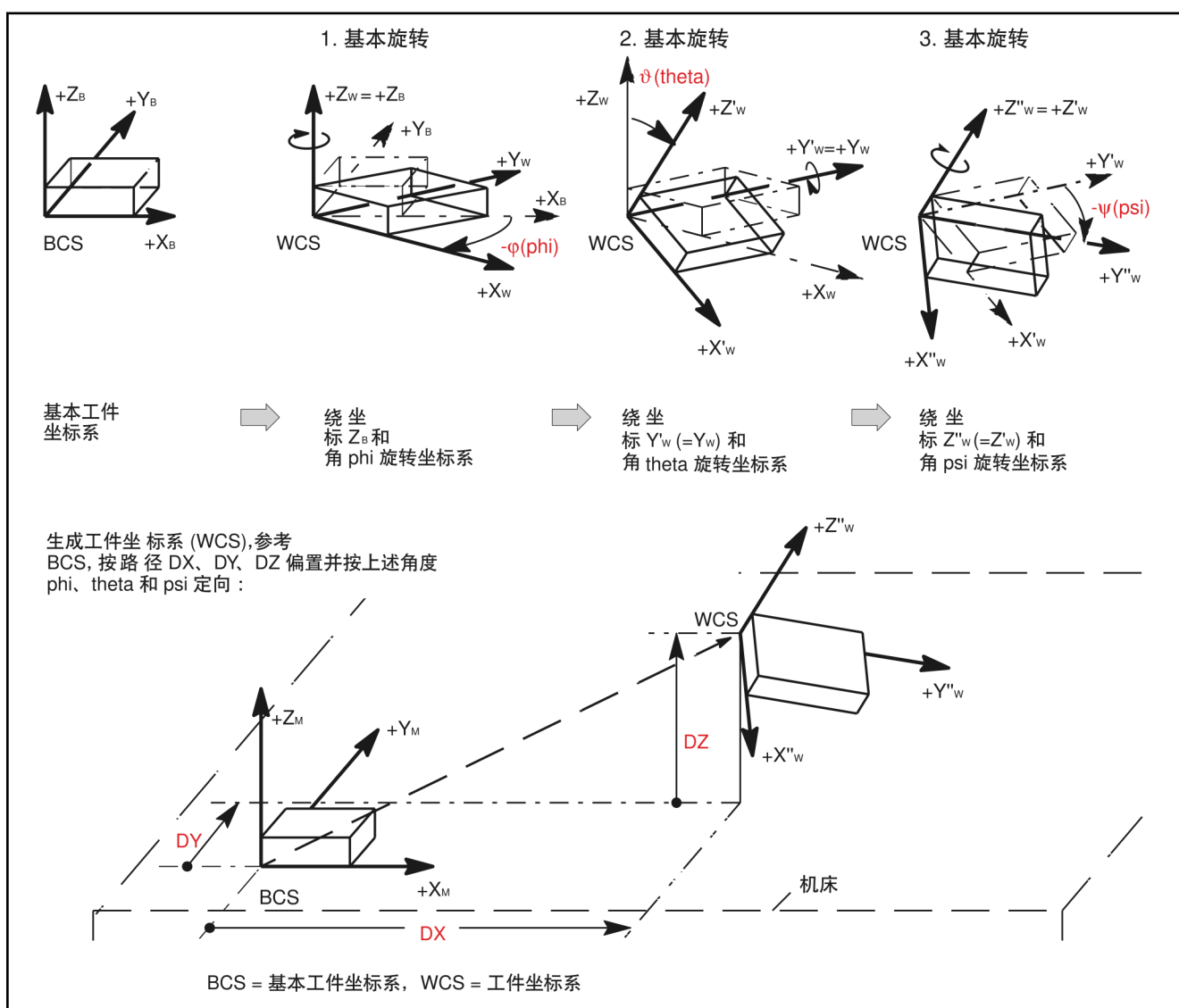


插图 5-35: “斜面”定位:  
“斜面”的结果功能受以下影响

- 五个库（1 至 5）；每个与另一个追加作用。

数控函数，语法符合 DIN 66025（包括扩展）

因此，可以创建多达五个基于彼此的工件坐标系。

- 每个库八个可选集。  
每个设置包含工件坐标系需要的所有偏置和方向数据。  
在一个库内，在任意给定时刻只能激活一个设置（即库的设置可以相互取消）。

斜面可以叠加于定位“工件位置修正”并且因此按照工件位置修正后面的“沉降顺序”：

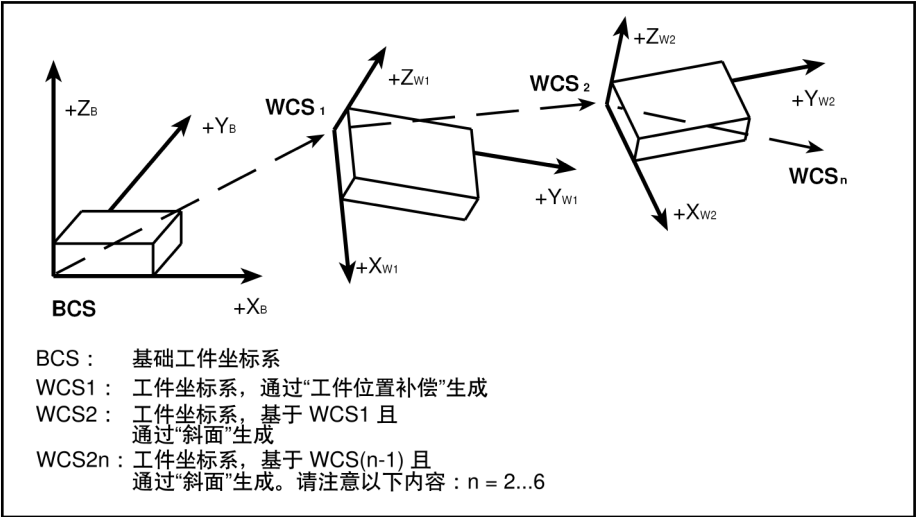


插图 5-36: 定位“工件位置修正”

语法: “设置”是指工件坐标系偏置和定向所需的全部数据。

套件可以如下操作:

- 隐式地，可以根据零件程序中的当前刀具方向（G151 - G151.5）启用这些组，
- 这些组可以编程为零件程序中的功能参数（G152 - G152.5）
- 用户也可以在位置表中指定它们（G154 - G159.5）  
(参见 第 6.84 章 “启用定位表 PmTSe1、PMS” 第 287 页 )  
一张位置表最多可以包含 30 个设置（共 5 库，每库 6 个设置）。

零件程序中的隐式激活（G151 - G151.5）：

如果斜面与刀具垂直，新工件坐标系的位置和方向可以直接通过 TCP 和刀具方向进行定义。



数控函数，语法符合 DIN 66025（包括扩展）

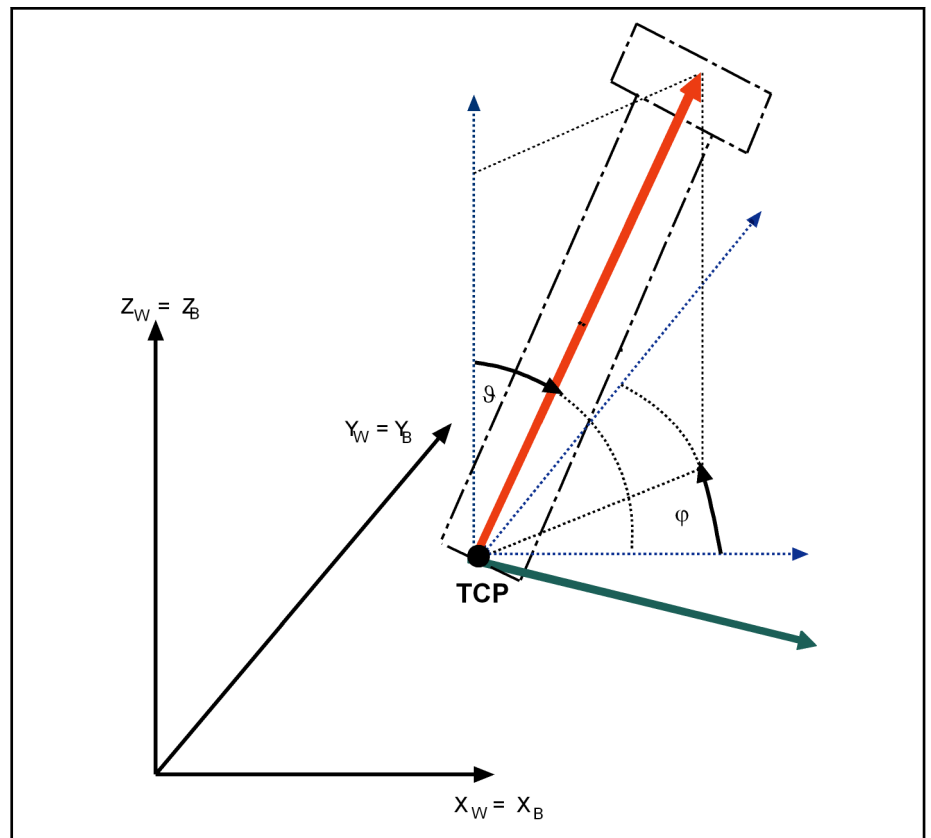


插图 5-37: 倾斜平面与刀具垂直

激活“与刀具垂直的斜面”后，新工件坐标系的原点位于刀具的 TCP 中。

新  $Z_w$  轴即沿着刀具的纵轴排布。

新的  $Y_w$  坐标以与平面平行的方式排布，该平面由前面的  $X_B$  和  $Y_B$  基本坐标定义。

根据数学定义，新  $X_w$  轴是其余两个坐标  $X_w = Y_w \times Z_w$  的向量产品轴结果。

数控函数，语法符合 DIN 66025（包括扩展）

下图显示了激活垂直于刀具的斜面后的配置

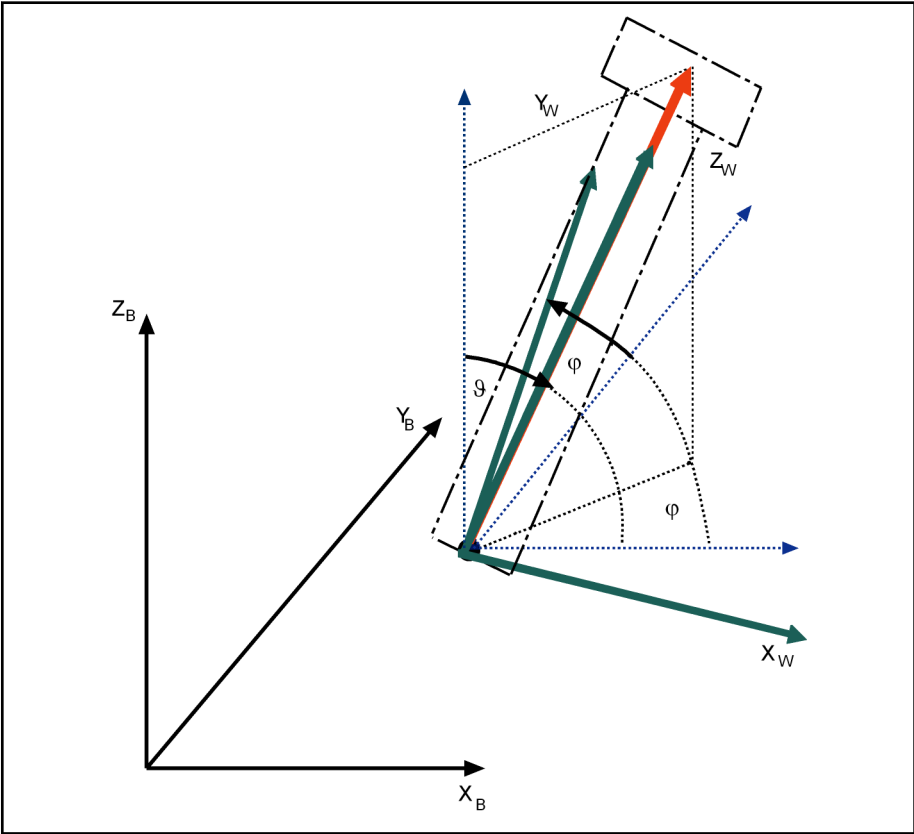


插图 5-38: 垂直于刀具斜面激活后的配置:  
您还可以对旋转角  $\psi$  进行编程，以关于新的  $Z_W$  坐标旋转坐标系的方向。

|                           |                                                                                            |
|---------------------------|--------------------------------------------------------------------------------------------|
| G151.<Bank>{ (<Angle3>) } | 所需库的斜面打开<br>根据当前程序坐标系（PCS）中刀具顶部（TCP）的当前位置得出的偏置值。旋转角度直接从当前刀具方向得出。如果激活的轴变换无法定向，则欧拉角的值各设置为 0。 |
| 缩写形式: G151                | 对应于 G151.1                                                                                 |
| G153.<Bank>               | 编程库的斜面关闭                                                                                   |
| G153                      | 斜面关闭（关闭所有页面）。                                                                              |
| 其中                        |                                                                                            |
| <Bank>                    | 所需页面的编号<br>取值范围: 1 到 5                                                                     |
| <Angle3>                  | 关于新 $Z_W$ 坐标的旋转角度<br>取值范围: $0 \leq \text{<Angle3>} \leq 360$ 度                             |

表格 5-74: 语法 G151, G153

直接在程序行中编程为参数定位（G152 – G152.5）

使用 G152 – G152.5 直接编程斜面时，新的工件坐标系的位置和方向直接由功能的位置参数编程。

数控函数，语法符合 DIN 66025（包括扩展）

|                                                                                                                                           |                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| G152.<Bank>({<X <sub>w</sub> Offset>}{, {<Y <sub>w</sub> Offset>}{, {<Z <sub>w</sub> Offset>}{, {<Angle1>}{, {<Angle2>}{, {<Angle3>}}}}}) |                                              |
|                                                                                                                                           | 所需库的斜面打开，具有编程的偏置 / 方向数据。                     |
| 缩写形式: G152                                                                                                                                | 对应于 G152.1                                   |
| G153.<Bank>                                                                                                                               | 编程库的斜面关闭                                     |
| G153                                                                                                                                      | 斜面关闭（关闭所有页面）。                                |
| 其中                                                                                                                                        |                                              |
| <Bank>                                                                                                                                    | 所需页面的编号<br>取值范围: 1 到 5                       |
| <X <sub>w</sub> offset>                                                                                                                   | 主坐标方向的偏置值                                    |
| <Y <sub>w</sub> offset>                                                                                                                   | 辅坐标方向的偏置值                                    |
| <Z <sub>w</sub> offset>                                                                                                                   | 法向坐标方向的偏置值                                   |
| <Angle1>                                                                                                                                  | 沿着 z 坐标旋转角度<br>取值范围: 0 ≤ <angle1> < 360 度    |
| <Angle2>                                                                                                                                  | 关于新 Y' 坐标的旋转角度<br>取值范围: 0 ≤ <angle2> < 180 度 |
| <Angle3>                                                                                                                                  | 沿着新 Z'' 坐标旋转角度<br>取值范围: 0 ≤ <angle3> < 360 度 |

表格 5-75:        语法 G152, G153

使用定位表编程:

激活所需定位表（请参见 [第 6.84 章 “启用定位表 PmTsel、PMS” 第 287 页](#)）。

对需要的功能编程:

|                   |                        |
|-------------------|------------------------|
| G154.<Bank>       | 编程的库设置 1 的斜面打开         |
| G155.<Bank>       | 编程的库设置 2 的斜面打开         |
| G156.<Bank>       | 编程的库设置 3 的斜面打开         |
| G157.<Bank>       | 编程的库设置 4 的斜面打开         |
| G158.<Bank>       | 编程的库设置 5 的斜面打开         |
| G159.<Bank>       | 编程的库设置 6 的斜面打开         |
| 缩写形式: G152 - G159 | 对应于 G154.1 - G159.1    |
| G153.<Bank>       | 编程库的斜面关闭               |
| G153              | 斜面关闭（关闭所有页面）。          |
| 其中                |                        |
| <Bank>            | 所需页面的编号<br>取值范围: 1 到 5 |

表格 5-76:        G153 - G159 语法

数控函数，语法符合 DIN 66025（包括扩展）

示例：用 G151 钻削锥孔

:

N05 coord(1)

N10 G1 x0 y0 z100 phi90 theta1 F2000

N15 G151

N20 P STC\_G81(SL90, DT80, RL100) F500

N25 G153

N30 phi90 theta10

N35 G151

N40 P STC\_G81(SL90, DT80, RL100) F500

N45 G153

N50 theta45

N55 G151

N60 P STC\_G81(SL90, DT80, RL100) F500

N65 G153

N70 phi0

N75 G151

N80 P STC\_G81(SL90, DT80, RL100) F500

N85 G153

N90 G2 I100 J0 ROTAX(0,0,1) O(-360)

N95 G151

N100 P STC\_G81(SL90, DT80, RL100)

F500

N105 G153

N110 G90G1

N120 coord(0)

M30

轴变换 1 在 FatT0InclinedPlane.npg 主程序中激活。然后，刀具会在空间中的不同点定位和定向。对于每个位置，垂直于刀具的位置被激活，随后使用 MTX 标准钻孔循环执行钻孔。镗孔深度 (DT)，安全平面 (SL) 和收缩高度 (RL) 作为参数传输到子程序。

示例：使用 G152 进行直接放置编程

:

N40 G152.1(100,0,0,90)

:

:

:

:

:

:

N180 G153

库 1 的程序定位  
无进给运动  
零点在 X 方向偏置 100 mm，而 WCS 关于 Z 坐标旋转 90 度。  
关闭所有页面（完全关闭斜面）

示例：使用表格激活展示位置：G154 – G159.5

N40 PMS(Tab1)

N50 G154.1

:

激活定位表 “Tab1”  
使用设置 1 打开第 1 库  
无进给运动

数控函数，语法符合 DIN 66025（包括扩展）

```
N90 G154.2 X1 Y2 Z3 使用设置 2 打开第 1 库
: (添加至库 1)
: 在结果 WCS 中，接近编程的位置 P(1,2,3)。
N120 G153.2 关闭第 2 库 (第 1 库保持有效)
:
N180 G153 关闭所有页面 (完全关闭斜面)
```

- 特殊功能和限制:
- 如果轴变换激活，则不得在同一块中将斜面以移动运动编程。
  - 在控制器复位的情况下，在机床参数 7060 00010 和 7060 00020 中配置上电状态和特性。
  - 所有功能在每个库 (1 – 5) 均为模态，且可相互取消选择。
  - 激活和停用倾斜面会中断块预测。因此，当铣刀路径校正有效时 (G41/G42，请参见 第 5.2.16 章 “铣刀路径修正 G40, G41, G42” 第 128 页)，不得编程。
  - 调用“垂直于刀具的斜面”功能，不对调用的合理性进行检查。因此，用户有责任从正确的放置库中调用该功能。
  - 现在激活斜面定位需要至少两根轴，并且在通道中可使用轴含义 (“X”、“Y”、“Z”)。

5.3 M 代码

5.3.1 中断程序（程序停止）"M0、M00"

- 描述:
- 中断数控程序，
  - 块执行后停止机床运动，并且
  - 发送基于通道的接口信号“程序停止 M0”。
- 当前通道状态切换至 “数控 就绪”。
- 块执行由新的 “NC 启动”继续。



参见 M1/M01，根据基于通道的接口信号“可选停止”来中断程序。

语法: M0 或 M00

特殊功能和限制: “程序终止”可与其它 数控函数一起编程到同一程序块中。只有在其他所有已编程功能执行完毕后，“程序停止”才会生效。

5.3.2 有条件程序中断（Cond. Progr. Stop）"M1、M01"

- 描述:
- 中断数控程序并且
  - 如果基于通道的接口信号“可选停止”是激活的，则会在执行块后停止机床运动。
- 当前通道状态切换至 “数控 就绪”。
- 块执行由新的 “NC 启动”继续。

语法:

M1 或 M01

表格 5-77: M1 语法

特殊功能和限制: “条件的程序终止”可与其它 数控函数一起编程到同一程序块中。只有在其他所有已编程功能执行完毕后，“条件地程序停止”才会生效。

5.3.3 退出程序（程序结束）"M2、M02、M30"

- 描述:
- 终止一个程序。

数控函数，语法符合 DIN 66025（包括扩展）

以下内容适用于作为子程序的程序：

- 发出辅助数控函数（M2、M02 或 M30）
- 数控返回到调用程序
- 数控处理调用程序

 子程序中已改变的模态状态不会复位！

以下内容适用于作为主程序的程序：

- 数控设置基于通道的接口信号“程序结束 M30”
  - 取消基于通道的接口信号“程序运行”
  - 数控取消选择有效的“自动档位选择”（但当前齿轮档保持选定状态）
  - 换至“直接速度编程”（G97）
  - 激活所有在机床参数 7060 00020 “默认状态”中为 “M30” 事件定义的所有状态
  - 返回主程序的开始处，并且
  - 等待下一个 “数控启动”
- 块执行由 “数控启动”时重启。

语法: 

M2 or M02 or M30

特殊功能和限制: 

表格 5-78: M2 语法  
在单独的编程行中对功能进行编程。

 未定义的加电状态会导致损坏！  
退出主程序后如果需要某些状态或功能，则**绝对有必要**在 MP 7060 00020 中正确设定 “M30” 事件初始化字符串的参数！  
初始化字符串必须包含在终止主程序后让 数控 设置为需要/所需的状态。在这种情况下，请记住，即使在终止程序后，以模态形式作用的功能仍然保持激活状态！

 关于各种加电状态的详细信息，请参见“《Rexroth IndraMotion MTX 功能描述》”手册。

5.3.4 主轴顺时针旋转 "M3、M103、M203"，主轴顺时针旋转和冷却液打开"M13、M113、M213"

描述:  所述语法仅适用于机床参数 1040 001xx 和 1040 002xx 的默认设置。可以自由配置语法，因此在您的机床上可能会有所不同！  
因此，MTB 提供的文档优先级始终更高。  
要确定此处所述功能是否能实际应用于您的机床，请联络您的系统管理员！

- 从“刀具到工件”的角度启动主轴顺时针旋转。
- 取消通过“定向/位置主轴”激活的位置控制。
- 保留当前通道的相应主轴。

 只有在上一块或同一块中，为相关主轴/主轴组编程大于 0 的 S-或 SSPG 字时，主轴运动才会开始（请参见 第 5.4.6 章 “编程主轴速度 S, SSPG” 第 176 页 ）。

数控函数，语法符合 DIN 66025（包括扩展）

语法:

|      |                      |
|------|----------------------|
| M3   | 影响第一主轴组              |
| M103 | 作用于第一主轴              |
| M203 | 作用于第二主轴              |
| M13  | 作用于第一主轴组<br>此外，冷却液打开 |
| M113 | 作用于第一主轴<br>此外，冷却液打开  |
| M213 | 作用于第二主轴<br>此外，冷却液打开  |

表格 5-79: M3、M13、M103、M113、M203 和 M213 语法

特殊功能和限制:

- 各自的功能会自动为当前通道保留对应主轴。只有在“主轴停止”功能（请参见第 5.3.6 章“主轴停止 M5、M105、M205”第 170 页）或“SpAdmin”（SPA，请参见第 6.118 章“启动 / 转换保留的主轴 SpAdmin、SPA”第 326 页）用于对应主轴时，才允许另一个通道使用保留的主轴。
- 在编程同一主轴的新运动状态之前（例如，另一个旋转方向、有/无冷却液、“主轴停止”或“定向主轴”），各自的功能保持活动状态。
- 在同一个块中编程的单主轴和主轴组之间的竞争作业将导致运行时错误。（示例：M3 和 M104 不能在同一个块中运行。）
- 换挡之后，之前编程的主轴旋转方向会自动恢复。

5.3.5 主轴逆时针旋转 "M4、M104、M204"，主轴逆时针旋转和冷却液打开"M14、M114、M214"

描述:

 所述语法仅适用于机床参数 1040 001xx 和 1040 002xx 的默认设置。可以自由配置语法，因此在您的机床上可能会有所不同！  
因此，MTB 提供的文档优先级始终更高。  
要确定此处所述功能是否能实际应用于您的机床，请联络您的系统管理员！

- 从“刀具到工件”的角度启动主轴逆时针旋转。
- 取消通过“校准 / 位置主轴”激活的位置控制。
- 保留当前通道的相应主轴。

 只有在上一块或同一块中，为相关主轴/主轴组编程大于 0 的 S-或 SSPG 字时，主轴运动才会开始（请参见第 5.4.6 章“编程主轴速度 S，SSPG”第 176 页）。

语法:

|      |                      |
|------|----------------------|
| M4   | 作用于第一主轴组             |
| M104 | 作用于第一主轴              |
| M204 | 作用于第二主轴              |
| M14  | 作用于第一主轴组<br>此外，冷却液打开 |

数控函数，语法符合 DIN 66025（包括扩展）

|      |                     |
|------|---------------------|
| M114 | 作用于第一主轴<br>此外，冷却液打开 |
| M214 | 作用于第二主轴<br>此外，冷却液打开 |

表格 5-80: M4、M14、M104、M114、M204 和 M214 语法

- 特殊功能和限制:
- 各自的功能会自动为当前通道保留对应主轴。只有在“主轴停止”功能（请参见 第 5.3.6 章 “主轴停止 M5、M105、M205” 第 170 页 ）或 “SpAdmin”（SPA，请参见 第 6.118 章 “启动 / 转换保留的主轴 SpAdmin、SPA” 第 326 页 ）用于对应主轴时，才允许另一个通道使用保留的主轴。
  - 在编程同一主轴的新运动状态之前（例如，另一个旋转方向、有/无冷却液、“主轴停止”或“定向主轴”），各自的功能保持活动状态。
  - 在同一个块中编程的单主轴和主轴组之间的竞争作业将导致运行时错误。（示例：M3 和 M104 不能在同一个块中运行。）
  - 换档之后，之前编程的主轴旋转方向会自动恢复。

5.3.6 主轴停止 "M5、M105、M205"

描述:



所述语法仅适用于机床参数 1040 001xx 和 1040 002xx 的默认设置。可以自由配置语法，因此在您的机床上可能会有所不同！  
因此，MTB 提供的文档优先级始终更高。  
如果无法确保所述功能是否能实际应用于您的机床，请联络您的系统管理员。

- 停止主轴
- 根据活动通道，取消给定主轴所有可能的有效保留。
- 取消通过“定向/位置主轴”激活的位置控制。

语法:

|      |           |
|------|-----------|
| M5   | 作用于主轴组 1。 |
| M105 | 作用于第一主轴   |
| M205 | 作用于第二主轴   |

表格 5-81: M5、M105 和 M205 语法

- 特殊功能和限制:
- 编程同一主轴的新运动状态之前（例如，“主轴，逆时针/顺时针旋转”或“定向主轴”），各自的功能保持活动状态。
  - 在同一个块中编程的单主轴和主轴组之间的竞争作业将导致运行时错误。（示例：M3 和 M105 不能在同一个块中运行。）

5.3.7 定向主轴 / 定位主轴"M19, M119, M219"

描述:



所述语法仅适用于机床参数 1040 001xx 和 1040 002xx 的默认设置。可以自由配置语法，因此在您的机床上可能会有所不同！  
因此，MTB 提供的文档优先级始终更高。  
如果无法确保所述功能是否能实际应用于您的机床，请联络您的系统管理员。

主轴会为当前通道保留，并使用位置控制器定位于可定义位置中。



数控函数，语法符合 DIN 66025（包括扩展）

定位会在以下状态下发生

- 静止状态下发生：  
如驱动器的参数 S-0-0154 中所定义。
- 旋转时发生：  
同时保持旋转方向时。

语法:

|                 |                                                                                                    |
|-----------------|----------------------------------------------------------------------------------------------------|
| M19             | 主轴组 1:<br>将所有相关主轴定位到其参考角度 (= 0 degree)。                                                            |
| M119            | 1. 将主轴定位到其参考角度 (= 0 度)                                                                             |
| M219            | 2. 将主轴定位到其参考角度 (= 0 度)                                                                             |
| M19S<Angle>     | 主轴组 1:<br>将第二主轴定位到 <angle>                                                                         |
| M119 S1=<Angle> | 1. 将主轴定位到 <angle>                                                                                  |
| M219 S2=<Angle> | 2. 将主轴定位到 <angle>                                                                                  |
| <Angle>         | 所需的绝对主轴位置，单位为度<br>取值范围: 0° ≤ 主轴位置 < 360°<br>如果对位置编程的值有所不同，它将自动转换为指定的间隔。<br>如果主轴已经在指定位置，则不需要执行任何运动。 |

表格 5-82: M19、M119 和 M219 语法

示例:

|                     |                                                            |
|---------------------|------------------------------------------------------------|
| N60 M19             | 第一主轴组中的所有主轴定位到 0 度。                                        |
| :                   |                                                            |
| N70 M219            | 只有第二主轴定位至其参考角度。                                            |
| :                   |                                                            |
| N80 M19 S180        | 第一主轴组中的所有主轴定位到 180 度。                                      |
| :                   |                                                            |
| N90 M119 S1=370     | 1. 主轴定位到 10 度。                                             |
| :                   |                                                            |
| N95 M19 S1=10 S2=20 | 如果第一和第二主轴被指定给第一主轴组:                                        |
| :                   | 第一主轴定位到 10 度，<br>第二主轴定位到 20 度。<br>第一主轴组的所有其他主轴关于其参考角度自行定位。 |

特殊功能和限制:

- 主轴定位仅支持 Sercos 主轴。
- 不能在同一块中将各自的功能以竞争主轴功能（“主轴，逆时针/顺时针旋转”、“主轴停止”）对之进行编程。
- 如果需要 S 字，则必须在同一个块中设定它。
- 定位后，主轴会保持位置控制中。位置控制仅通过“主轴，逆时针/顺时针旋转”或“主轴停止”自动取消。
- 在同一个块中编程的单主轴和主轴组之间的竞争作业将导致运行时错误。  
(示例: M3 和 M119 不能在同一个块中运行。)
- 通过设置接口信号 “qSp\_SValueSD”，借助系统数据 “SysSpCmdData[n]/OriPos” 进行位置分配。编程的规格被忽略。

数控函数，语法符合 DIN 66025（包括扩展）

5.3.8 自动换档步进选择"M40、M140、M240"

描述:

 所述语法仅适用于机床参数 1040 001xx 和 1040 002xx 的默认设置。可以自由配置语法，因此在您的机床上可能会有所不同！  
因此，MTB 提供的文档优先级始终更高。  
如果无法确保所述功能是否能实际应用于您的机床，请联络您的系统管理员。

控制器会根据有效速度自动从可用齿轮档中选择合适的齿轮档。  
如果各级别的速度范围互相重叠，控制器将始终选择下一较低水平（包含更高的电机速度）。

语法:

|      |                  |
|------|------------------|
| M40  | 主轴组 1 自动选择齿轮档打开。 |
| M140 | 第一主轴启用自动选择齿轮档。   |
| M240 | 第二主轴启用自动选择齿轮档。   |

- 特殊功能和限制:
- 表格 5-83: M40、M140 和 M240 语法

  - 编程速度 “0” 不会导致当前齿轮档发生变化。
  - M40、M41-M44、M48、M140、M141-M144 和 M148 都是模态，且可以相互取消选择。
  - M240、M241-M244 和 M248 都是模态，且可以相互取消选择。
  - M30 取消选择有效的自动档位选择（但当前齿轮档保持选定状态）
  - 档位切换仅支持 Sercos 主轴。  
对于没有 Sercos 接口的主轴，只有辅助功能被转发给 PLC。

5.3.9 手动换档选择 "M41...44、M141...144、M241...244"

描述:

 所述语法仅适用于机床参数 1040 001xx 和 1040 002xx 的默认设置。可以自由配置语法，因此在您的机床上可能会有所不同！  
因此，MTB 提供的文档优先级始终更高。  
如果无法确保所述功能是否能实际应用于您的机床，请联络您的系统管理员。

选择对应齿轮档。  
如果设定的速度超出了相应档位范围的速度范围，控制器会将其设置为相关齿轮档的最低/最高速度。

语法:

|            |                      |
|------------|----------------------|
| M4<Stage>  | 激活主轴组 1 的齿轮档 <Stage> |
| M14<Stage> | 激活第一主轴的齿轮档。          |
| M24<Stage> | 激活第二主轴的齿轮档 <stage>。  |
| 其中         |                      |
| <Stage>    | 输入范围: 1 ... 4        |

- 特殊功能和限制:
- 表格 5-84: M4、M14 和 M24 语法

  - M40、M41-M44、M48、M140、M141-M144 和 M148

数控函数，语法符合 DIN 66025（包括扩展）

- 都是模态，且可以相互取消选择。
- M240、M241-M244 和 M248  
都是模态，且可以相互取消选择。
  - 档位切换仅支持 Sercos 主轴。  
对于没有 Sercos 接口的主轴，只有辅助功能被转发给 PLC。

5.3.10 惰档选择 "M48、M148、M248"

描述：



所述语法仅适用于机床参数 1040 001xx 和 1040 002xx 的默认设置。可以自由配置语法，因此在您的机床上可能会有所不同！

因此，MTB 提供的文档优先级始终更高。

如果无法确保所述功能是否能实际应用于您的机床，请联络您的系统管理员。

脱离齿轮档。  
档位之后处于空转模式。

语法：

|      |              |
|------|--------------|
| M48  | 脱离主轴组 1 的齿轮档 |
| M148 | 脱离第一主轴的齿轮档   |
| M248 | 脱离第二主轴的齿轮档   |

表格 5-85: M48、M148 和 M248 语法

特殊功能和限制：

- M40、M41-M44、M48、M140、M141-M144 和 M148  
都是模态，且可以相互取消选择。
- M240、M241-M244 和 M248  
都是模态，且可以相互取消选择。
- 档位切换仅支持 Sercos 主轴。  
对于没有 Sercos 接口的主轴，只有辅助功能被转发给 PLC。

5.4 进给和速度编程

5.4.1 速度设置或模态速度设置 "<Axis>=S<Value>"

**描述：** 异步循环轴将加速至指定速度（转 / 分钟或模态区间 / 分钟）。速度设置将一直有效，直至被新的规格代替或被定位运动取消。速度的符号决定旋转方向或运动方向。达到所需速度之后，设置接口信号 iAx\_ProgSpReach = 1。

实时来自速度操作定位运动在运动方向上下一可能的位置将轴停止。轴的定位逻辑和定位属性（DC、ACN 等）不起任何作用。使用 FA=<Feed> 的进给规定可加速定位过程，但无法实现减速。

数控函数，语法符合 DIN 66025（包括扩展）

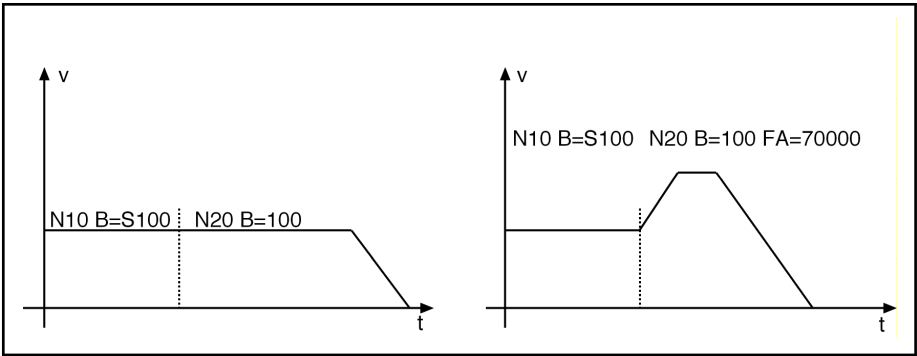


插图 5-39: 飞行定位

语法:

|                               |                        |
|-------------------------------|------------------------|
| <AxisAddress>=S<Value><Value> |                        |
| 其中                            |                        |
| <Value>                       | 速度，单位为 rpm 或模态区间 /min。 |

表格 5-86: S 属性语法

|     |              |                                      |
|-----|--------------|--------------------------------------|
| 示例: | N100 B=S1000 | 将 B 轴加速至 1000 rpm，并常量移动              |
|     | N200 B=-600  | 将 B 轴减速并在反方向上加速至 600 rpm。            |
|     | N300 B=100   | 将 B 轴定位至 100° 绝对位置，速率为增加的定位速度（对      |
|     | N300 B=100   | 应 1000 rpm）                          |
|     | FA=360000    |                                      |
|     | N400 W=S100  | 从 20 mm 模态范围加速线性 W-轴至 100 模态范围 / MIN |
|     |              | (200 mm/min)，并使其常量移动。                |

- 特殊功能和限制:
- 结果速度取决于特定于轴的倍率。最大速度受限于最大允许轴速度 (AX/Dr[i]/Vel/MaxVel/) 和每数控循环时间 (SCS/ScsIf[1..2]/ScsCycTime) 模态范围 (S-0-0103) 的一半。
  - 在激活速度插补的情况下，定位运动的编程属性 ACN、ACP、DC 或 IC 被忽略。
  - 系统复位以允许轴加速度取消速度插补。
  - 过冲限制 (ASO) 仅可在速度插补开始时启用。在速度修改或飞行定位的情况下，规格被忽略。

5.4.2 F 地址 "F"

- 描述:
- 依赖当前有效的 G 功能 G93、G94 和 G95，控制器将 F 地址解释为
- 以秒为单位的停留时间  
(见 G93, 第 5.2.33 章 "时间编程 G93" 第 146 页 )
  - 1 (毫米 / 分钟 (英寸 / 分钟))  
(见 G94, 第 5.2.34 章 "进给编程 (每分钟) G94" 第 147 页 )
  - 以毫米 / 转为单位进给  
(见 G95, 第 5.2.39 章 "进给编程 (每转) G95" 第 153 页 )
- 以下情况适用:
- F 以模态形式作用于 G94 和 G95。

数控函数，语法符合 DIN 66025（包括扩展）



上个 F 字会在加电、控制器复位或复位后改变！  
上面所列事件完成后，在机床参数 7060 00020 或 7060 00010 中定义的 F 字生效（默认值：F0）。有关 G93、G94 或 G95 是否激活的规范也保存在这里（默认值：G94）。  
因此，确保所需进给在加工前已经编程！

语法:

|          |                                 |
|----------|---------------------------------|
| F<Value> |                                 |
| 其中       |                                 |
| <Value>  | 依赖有效的 G 功能，可以将之解释为插补时间、进给或停留时间。 |

表格 5-87: F<value> 语法

特殊功能和限制:

可使用“测试进给”功能禁止编程的路径速度。  
它是通过通道接口信号 “qCh\_TestFeed” 控制的。

5.4.3 F2 地址"F2"

描述

除了当前可用的 F 值（默认进给）之外，还为 G94 模式引入了第二个进给 F2（进给 2）。Feed2 的值可以低于或高于默认的 Feed。根据两个 F 值的最大值考虑速度限制和速度曲线准备（预测）。  
Feed2 由 PLC 通过通道接口的输出信号 13.2（qCh\_Feedrate2）激活和禁用。使用 “qCh\_Feedrate2 = 1” 应用 Feed2。立即应用馈送中的变化，即，路径速度立即变为新的速度，即已经在活动块中。  
使用通道接口的 PLC 输入信号 13.4（iCh\_Feedrate2），数控向 PLC 报告 Feed2 激活。  
如果 G94 有效，则控制器将 F2 地址 “F2” 解释为

- 以 mm/min 或 inch/min 为单位的进给（请参见 G94，第 5.2.34 章 “进给编程（每分钟）G94” 第 147 页）

以下情况适用：  
如果 G94 激活，则 F2 具有模态效果。



控制复位或复位后，最新的当前 F2 值不会改变！  
因此，确保所需进给在加工前已经编程！

语法:

|            |                                |
|------------|--------------------------------|
| F2=<value> | 语法是“F2=”。<br>“F2”和“=”之间不允许有空格。 |
| 其中         |                                |
| 数值         | 如果 G94 激活，则解释为供给               |

表格 5-88: 图 5.115: F2=<value> 语法

示例:

|                           |                                      |
|---------------------------|--------------------------------------|
| N10 G1 G94 X200 Z300 F200 | 同步轴进给:                               |
| F2=1000                   | 1000 毫米 / 分钟, 如果 qCh_Feedrate2 = 1 和 |
|                           | 200 毫米 / 分钟, 如果 qCh_Feedrate2 = 0    |

特殊功能和限制:

可以使用“测试进给”功能禁止编程 Feed2。它是通过通道接口信号 “qCh\_TestFeed” 控制的。  
与默认提要一样，Feed2 也会应用提要替代。可以通过系统日期 SD.SysChCmd [...]。Feedrate20vr 将一个独立的覆盖值（0 到 100% 之间的值）添加到 Feed2 中。效果是乘法。默认情况下，SD.SysChCmd [...]。Feedrate20vr 被初始化为 100 并存在于每个通道中。

数控函数，语法符合 DIN 66025（包括扩展）

在两个馈送值（F↔F2）之间切换也具有与覆盖更改相同的效果。

5.4.4 异步轴的速度 "FA"

**描述:** 在 FA 块中编程的所有异步轴移动运动都会以编程的速度进行进给，而不是快速进给。

要定义线性轴异步进给的解释单位，请使用配置参数 “数控 0/CorrUnit/AsynchrAxUnit”：mm/min 或 inch/min

您可以通过编程属性 MM(...) 和 I 数控 H(...) 在局部切换特定块的单位。

对于回转轴，无论配置中的 FA-进给配置设置如何，也不考虑编程属性，编程的 FA -进给单位会解释为度/分。



错误编程偶尔会造成机床损坏。

给定速度只在当前 FA 块中有效。

在后续块中对异步轴编程而不对 FA 字重新编程会使轴再次以快速模式进给。

语法:

|           |      |
|-----------|------|
| FA<Value> |      |
| 其中        |      |
| <Value>   | 所需速度 |

表格 5-89: FA 语法

示例:

|                           |                                 |
|---------------------------|---------------------------------|
| N10 G1 G94 X200 Z300 F200 | 同步轴进给:<br>200 mm/min            |
| N11 UA400 VA140 FA250     | 异步轴 UA 和 VA 以 250 mm/min 的速度进给。 |
| N12 UA0 WA10              | 异步轴 UA 和 WA 再次以快速进给进行进给。        |

5.4.5 Omega 地址（进给率）"Omega"

**描述:** 如果只有用于进给计算的隐藏轴在块中（见 “FeedAd”，第 6.43 章 “进给计算：隐藏轴 FeedAd、FAD” 第 237 页），它们的进给可通过 “Omega” 地址进行设置。

语法:

|              |       |
|--------------|-------|
| Omega<Value> |       |
| 其中           |       |
| <Value>      | 所需的进给 |

表格 5-90: Omega 语法

5.4.6 编程主轴速度 "S，SSPG"

**描述:** 有效的 G97 定义以下需要的速度

- 单个主轴 (S...) 或者
- 一个完整的主轴组 (SSPG ...)。



只有存在主轴旋转命令（如顺时针旋转：M3；逆时针旋转：M4）存在。

语法:

|                     |            |
|---------------------|------------|
| S<Number>=<Value>   | 编程单个主轴的速度。 |
| SSPG<Group>=<Value> | 编程整组主轴的速度。 |

数控函数，语法符合 DIN 66025（包括扩展）

|          |                                                                                                    |
|----------|----------------------------------------------------------------------------------------------------|
| S<Value> | 编程第一主轴速度的缩写符号。<br>如果第一主轴没有通过 MP 1040 00002 被分配给 <b>任何</b> 主轴组，只对第一主轴有效。<br>否则，会使用该值编程包含第一主轴的整个主轴组。 |
| 其中       |                                                                                                    |
| <Number> | 主轴编号（主轴索引）<br>输入范围：<br>从 1 到已定义的主轴数量（通过 MP1040 00001 定义）<br>整数                                     |
| <Group>  | 主轴组的编号<br>输入范围：1 ... 4                                                                             |
| <Value>  | 需要的主轴速度（默认单位：rpm）。<br>输入值：≥ 0                                                                      |

表格 5-91: S 和 SSPG 语法

- 特殊功能和限制：
- 当 G96 激活时，S 字被解释为切削速度。  
有关语法，请参见 第 5.2.42 章 “常量切削速度 G96、直接速度编程 G97” 第 155 页。
  - 结合功能“校准主轴”时，S 字被解释为定位角。  
有关语法，请参见 第 5.3.7 章 “定向主轴 / 定位主轴 M19, M119, M219 ” 第 170 页。
  - 在同一个块中，可对数个主轴/主轴组的速度进行编程。
  - 结果速度取决于特定特定主轴倍率。
  - 结果速度受到以下因素限制
    - 功能 “SMin” 或 “SMax”  
(参见 第 6.115 章 “速度限制 SMin、SMN, SMax、SMX” 第 324 页)
    - 齿轮档限制值 (MP 1040 00011, 1040 00012)
  - 耦合主轴（同步运行主轴）组的最大允许速度取决于数控循环时间。  
以下情况适用：Smax [rpm] = 14400 / (MP 9030 00001 [ms])
  - 在被同一主轴新速度指定覆盖之前，设置速度将有效。  
控制启动后，应用 S = 0。
  - 通过设置接口信号 “qSp\_SValueSD”，通过系统日期 “SysSpCmdData[n]/Speed” 进行速度指定。通过 S、Sx= 和 SSPGx 编程的速度被忽略。

|                   |                            |
|-------------------|----------------------------|
| 示例：               |                            |
| N10 G97           | 激活速度编程                     |
| N20 SSPG1=1000    | 把第一主轴组的所有主轴的速度设置为 1000 rpm |
| :                 |                            |
| N50 S1=2000 S2=60 | 把第一主轴的速度设置为 2000 rpm       |
| :                 | 把第二主轴的速度设置为 60 rpm         |

数控函数，语法符合 DIN 66025（包括扩展）

N80 S3=2000

把第三主轴的速度设置为 2000 转 / 分钟

:

N90 S1500

把第一主轴的速度设置为 1500 rpm

:

如果第一主轴分配给一个主轴组，则速度规格适用于整个主轴组。

5.5

刀具修正

5.5.1

D 修正 "D"

**描述:** D 修正调用储存在 IndraMotion MTX 的 XML-D-修正表中的刀具修正值。一张校正表最多可包含 99 个数据块。

每个数据块包含以下修正值:

- 3 个刀具长度 L1、L2、L3,
- 刀具刃口半径 RAD,
- 刀具刃口位置 ORI。

D 修正同样适用于钻削、铣削、车床和角头刀具。通过总共三个偏置值（L1、L2 和 L3），可以执行刀具的常量三维刀具偏置以及多达三个不同刀具的平行长度校正。

通过“DeTSe1”（DCS）功能激活一个 D-修正表；然后利用数控命令“D”从 99 个数据块中选择一个修正设置。

如果刀具修正 G47（刀具长度和刀具刃口位置）或者铣削刀具路径修正 G41/G42 或者 G141/G142（刀具半径）有效，则校正值将生效。它们通过激活选择的外部刀具修正 ED 修正值进行追加叠加。

以下情况适用:

- 预选择的刀具集有模态效果。新编程会删除当前预选的刀具集。
- 可同模块中将 D 修正作为其它路径条件、移动运动或辅助功能进行编程。
- 只有在对应数控函数已激活时才能计算刀具修正：G47、G41、G42、G141、G142。

语法:

|                   |                           |
|-------------------|---------------------------|
| D<No. tool set>   | 从有效的 D 修正表中预选定刀具修正块。      |
| D0                | 在不选择新设置情况下，取消选择刀具修正设置。    |
| 其中                |                           |
| <no. tool record> | 校正数据块编号：<br>输入范围：1 ... 99 |

表格 5-92: D 和 D0 语法

**示例:** N10 D7      预选定刀具设置 7 的校正值。

:

5.5.2

ED 补偿 "ED"

**描述:** 外部刀具修正（ED 补偿）最多可为 16 个刀具刃口调用补偿值。PLC 或具有 CPL 命令“DCT”的零件程序可用于通过程序功能块“MT\_TCorr”写入修正值。

总共 16 个数据块，每个数据块包含以下校正值:

- 3 个刀具长度 L1、L2、L3,



数控函数，语法符合 DIN 66025（包括扩展）

- 刀具刃口半径 RAD,
- 刀具刃口位置 ORI

ED 补偿同样适用于钻削、铣削、车削和跨职员刀具。通过总共三个偏置值（L1、L2 和 L3），可以执行刀具的常量三维刀具偏置以及多达三个不同刀具的平行长度校正。

使用数控命令“ED”从 16 个数据块中选择一个校正设置。

如果刀具修正 G47（刀具长度和刀具刃口位置）或者铣削刀具路径修正 G41/G42 或者 G141/G142（刀具半径）有效，则校正值将生效。并通过已激活选定的 D 修正值进行追加叠加。

以下情况适用:

- 预选定的刀具刃口有模态效果。新编程将删除原先有效的预选择刀具刃口。
- 在同一模块中，可将 ED 修正如其它路径条件、移动运动或辅助功能进行编程。
- 只有在对应数控函数已激活时才能计算刀具修正：  
G47、G41、G42、G141、G142。
- 可在对应通道的 PLC 接口上输出预选择刀具刃口：

iCh\_ActFunc1..24

语法:

|                   |                          |
|-------------------|--------------------------|
| ED<No. tool edge> | 预选择校正设置                  |
| ED0               | 在不选择新设置情况下，取消选择校正块。      |
| 其中                |                          |
| <No. tool edge>   | 刀具刃口的编号<br>输入范围：1 ... 16 |

表格 5-93: ED 语法

示例:

|         |                 |
|---------|-----------------|
| N10 ED7 | 预选择刀具刃口 7 的校正值。 |
| :       |                 |



## 6 使用高级语言语法的 NC 功能



**不希望的移动运动！**

许多数控函数需要对轴和/或坐标地址进行编程。通常，只有各自的通道轴地址，即同步轴可与坐标地址分开编程。

**编程参数列表中异步轴的地址通常会导致语法错误，而在参数列表之外进行编程会导致异步遍历运动！**

例外情况：

- G74(Home) VA1：  
异步轴 VA 运动到参考点
- GAX(VA)：  
异步轴 VA 应用于通道
- FsMove、FsTorque、FsReset：  
“移动到固定停止”功能也会影响异步轴

### 6.1 概览

该控件提供多个数控函数。除了 DIN 66025 中规定的命令之外，这些功能还包括 G 代码和本章中描述的其他常规语言类型语法元素的重要补充。

**使用高级语言语法的 NC 功能：**

对于具有通用语言句法的 NC 函数，函数的含义通常可以从名称（例如“scale”：缩放）得出。

以下均为重要的类型：

- 全称，不区分大小写。  
为了提高本文档的可读性，复合部分词的首字母大写。示例：“KvProg”
- 简短形式，由三个大写字母组成（在特殊情况下为四个大写字母）。  
示例：  
KvProg 简称：KVP  
激活的模式 NC 功能的过程数据显示中也显示了对应的缩写形式。

以下语法描述显示了两种变型。



有关所有数控函数的表格概述，请参阅 [第 8 章“附录”第 535 页](#)。

另外，所有的数控函数都在索引中。

**使用的符号：** 在本文档中，以下符号用于数控函数的语法：

字体 “Courier bold” 或 “Courier”：

这种字体中的字符串必须如下所示进行编程。

示例：G0 (POL)

**角度括号 < >**

指示要编程的表达式/参数的占位符。占位符用斜体表示。

示例：<Axis1>

**大括号 { }**

表示一个可选的表达式/参数。

这样的语法元素可以但不必被编程。

使用高级语言语法的 NC 功能

示例： G0{ ({ POL,} {<Par1>} )}

字符 “|”

分离不能同时使用的可能参数（替代参数）。

示例： G0{ ({ POL,} { NIPS| IPS1| IPS2| IPS3} )}

6.2

编程有效半径因子 "ActRadFact、ARF"

6.2.1

效果

进给计算需要计算编程块的路径长度。如果是直线轴/坐标，则为几何距离（根（L1<sup>2</sup> + L2<sup>2</sup> + L3<sup>2</sup> ...））。在旋转轴 / 坐标的情况下，可以定义一个附加系数，通过该系数可以将距离相乘以修改轴对进给的影响（根（L1<sup>2</sup> + L2<sup>2</sup> + L3<sup>2</sup> + （R1 \* F1）<sup>2</sup> + （R2 \* F2）<sup>2</sup> ...））。如果该因数等于 1，则 1° =1mm 适用。这对应于 57.3 毫米的有效半径。

**示例** 线性轴移动 100 毫米，旋转轴移动 100 度，两轴以编程的进给 / 根部（2）以[毫米 / 分钟]和[度 / 分钟]移动，因为两个移动路径被认为是相等的长度。

激活的半径因子可以由机床参数定义“旋转轴速度的通道专用缩放比例因子”（NC0/VelScaleFact / Ch [k]/FactRotInch和NC0 / VelScaleFact/Ch [k]/FactRotMetr）。

可以通过“ActRadFact”在零件程序中指定一个**附加**系数。

除了轴和坐标的有效半径系数之外，还可以为方向指定有效半径系数，但前提是支持有效转换。

6.2.2

编程

语法：

|                                                           |                       |
|-----------------------------------------------------------|-----------------------|
| ActRadFact(<AxisName> <Factor> {,<AxisName> <Factor>}...) |                       |
| 缩写形式：ARF(...)                                             |                       |
| 其中                                                        |                       |
| <Axis name>                                               | 旋转轴的名称，坐标或“0”（方向）     |
| <Factor>                                                  | 下面的块适用的有效半径系数（因子）= 0） |
| ActRadFact<br>( )                                         | 通道的所有有效半径因子都重置为 1。    |

表格 6-1: ActRadFact 语法

- 特殊功能和限制：
- 有效的半径因子可能不是负数。
  - 系统启动后，ActRadFact 的所有有效半径因子都设为 1。再次应用机床参数中已配置的系数。
  - 如果轴从通道中移除，则编程的有效半径系数将丢失。
  - 如果轴 / 坐标的有效半径等于 0，则只能包含该轴/坐标部分的运动可能无法编程。
  - 如果主动变换改变，则应用等坐标的有效半径因子。所有其他有效半径因子都重置为 1。

示例：

|                             |                            |
|-----------------------------|----------------------------|
| G0 X0 A0                    | 起始位置                       |
| N010 G1 G91 F1000 X100 A100 | X→707 [mm/min] A→707 [度/分] |
| N020 ARF(A0.5)              | A 部分分成两半                   |
| N030 X100 A100              | X→894 [mm/min] A→894 [度/分] |
| N040 ARF(A2)                | A 部分重复                     |

使用高级语言语法的 NC 功能

|                |                              |
|----------------|------------------------------|
| N050 X100 A100 | X→447 [mm/min] A→447 [度/分]   |
| N060 ARF(A10)  | A 部分乘以 10 倍                  |
| N070 X100 A100 | X→99 [mm/min] A→99 [度/分]     |
| N080 ARF(A0)   | A 部分隐藏                       |
| N090 X100 A100 | X→1000 [mm/min] A→1000 [度/分] |
| N100 ARF()     | A 的有效半径系数重置为 1               |

6.3 调整模态 NC 功能用一般语言句法 "调整, ADJ"

**描述:** 使用调整功能 "调整"后, 调整程序中的模态 NC 功能可以在块预运行后轻松调整。模态 NC 功能的语法作为调整功能的参数提供。该控制功能在块预运行的目标块之前各自的模态组中启用 NC 功能。

|            |                |                    |
|------------|----------------|--------------------|
| <b>语法:</b> | ADJ (<NCF 名称>) |                    |
|            | <NCF 名称>       | 要调整的组的模态 NC 功能的语法。 |

表格 6-2: ADJ 语法

**示例:** ADJ(G0)  
目标块之前的模态有效几何函数 (G0, G1, G2, G3 ...) 被激活, 从而调整 NC 功能组。

以下列出的用于 NC 功能的功能特定参数可以同时进行调整:

| 模态组          | NC 功能               | 支持的参数                        |
|--------------|---------------------|------------------------------|
| 插值函数         | G0                  | 极坐标编程和准确停止窗口 (NIPS, IPS <n>) |
|              | G1                  | 极坐标编程和准确停止窗口 (IPS <n>)       |
|              | G2, G3              | 极坐标编程                        |
| 路径斜坡         | G8                  | 路径形状                         |
|              | G9                  | 路径形状, 轴形状                    |
| 等级           | G17、G18、G19、G20     | 轴分配                          |
| 刀具长度修正       | G47                 | 长度分配                         |
| 零点偏移 1       | G52.1               | 可编程 Z0 1*)                   |
|              | G53.1               | Z0 1 关闭 *)                   |
|              | G54.1, ..., G59.1   | 激活 Z0 表库 1*)                 |
| ...          |                     |                              |
| 零点偏移 5       | G52.5               | 可编程 Z0 5*)                   |
|              | G53.5               | Z0 5 关闭 *)                   |
|              | G54.5, ..., G59.5   | 激活 Z0 表库 5*)                 |
| 展示位置: "斜面" 1 | G151.1              | 工具定位**)                      |
|              | G152.1              | 可编程斜面 1**)                   |
|              | G153.1              | 斜面 1 关闭**)                   |
|              | G154.1, ..., G159.1 | 激活斜面; 表库 1**)                |
| ...          |                     |                              |

## 使用高级语言语法的 NC 功能

| 模式组          | NC 功能                      | 支持的参数                                                          |
|--------------|----------------------------|----------------------------------------------------------------|
| 展示位置: “斜面” 5 | G151.5                     | 工具定位 5**)                                                      |
|              | G152.5                     | 可编程斜面 5**)                                                     |
|              | G153.5                     | 斜面 5 关闭**)                                                     |
|              | G154.5, ..., G159.5        | 激活斜面; 表库 5**)                                                  |
| 准确停止         | G61                        | 准确停止窗口 (IPS <n>)                                               |
| 刀具长度分配       | G78                        | ActPlane 或长度分配                                                 |
| 路径进给         | G93                        | F 值                                                            |
|              | G94                        | F 值, F2 值                                                      |
|              | G95                        | F 值, 参考轴 (FRA)                                                 |
| 主轴编程         | G96                        | 参考轴, 主轴分配                                                      |
|              | G97                        | 主轴分配                                                           |
| D 修正         | D0                         | D 修正关闭***)                                                     |
|              | D1, ..., D99               | 激活的 D 修正***)                                                   |
| 轴加速度         | AAC                        | 程序轴加速度                                                         |
| 轴突起          | AJK                        | 程序轴跳动                                                          |
| 累加程序坐标偏置     | ATR                        | 程序零点坐标                                                         |
| 最大轴速度        | AVE                        | 轴速度列表                                                          |
| 通道轴耦合        | AXC                        | 块预运行的通道主轴和相应的从轴耦合关系和表格。<br>先决条件:<br>目标块不能包含任何耦合命令。<br>联轴器必须禁用。 |
| 工件位置         | BCR                        | 工件零点坐标, 角度                                                     |
| 倒角 / 倒圆      | CHL<br>CHS<br>RND<br>RNE   | 倒角长度, 倒角段<br>半径, epsilon                                       |
| 自动角点检测       | CLD                        | 角度, 角度系数, 最大点距离, 距离系数, 精度极限                                    |
| 铣削路径碰撞检测     | CLN                        | 监控类型, 预览区域                                                     |
| 轴变换          | COORD                      | 带有变换选项的轴变换 1 和 2                                               |
| 半径 / 直径编程    | DIA<br>RAD                 | 直径坐标                                                           |
| 通道主轴和主轴组     | DSP、GSP、RSP<br>SPGn、SPGALL | 通道主轴和通道组分配 (对应于 SPADJSP)                                       |
| 没有速度减小的块转换   | DTT                        | 过渡角                                                            |
| 进给遮蔽轴        | FAD                        | 遮罩轴列表                                                          |
| 进给坐标组        | FDG<br>FDGT                | 路径坐标列表, 进给组类型                                                  |

| 模态组        | NC 功能 | 支持的参数                                          |
|------------|-------|------------------------------------------------|
| 直线度和角度误差补偿 | GCT   | –                                              |
| 活动轴猛拉      | JAL   | 与路径相关的轴加速度轴列表                                  |
| 冲击限制       | JKC   | –                                              |
| 输入刀具“镜像”   | MIR   | 镜像轴                                            |
| 主轴         | MSP   | 主轴                                             |
| 路径加速度      | PAC   | 路径加速，路径减速                                      |
| 极定义        | PLS   | 镜像 / 转折点                                       |
| 旋转轴定位模式    | PMD   | 定位模式                                           |
| 极定义        | POP   | 极坐标                                            |
| 精度编程       | PRP   | 半径轮廓误差（EPS），落后（DIST）                           |
| 最大径向加速度    | RAC   | 加速度                                            |
| 输入帮助：车削    | ROT   | 角                                              |
| 输入帮助：缩放    | SCL   | 轴心因素                                           |
| 输入帮助：偏置    | SHT   | 轮廓零点坐标                                         |
| 样条曲线定义     | SPD   | 样条曲线类型，样条曲线坐标，起始条件（SBC），结束条件（EBC），样条曲线参数长度（PL） |
| 几何分裂       | SPT   | 模式，长度                                          |
| 程序坐标偏置     | TRS   | 程序零点坐标                                         |
| 攻丝样条曲线     | TSP   | 主轴                                             |

表格 6-3: 可调模态参数化 NC 功能

- \*) 隐式 ZO 表选择（ZOS）
- \*\*) 隐式放置表选择（PMS）
- \*\*\*) 隐式 D-修正表选择（DCS）

- 特殊功能和限制：
- 必要时，必须使用调整程序中的预运行系统数据（InitAdjPrg）或使用操作块来执行上述未提及的模态可参数化 NC 功能的调整。
  - 零点偏移或倾斜平面的每个组必须分别进行调整：ADJ（G54.1），ADJ（G54.2）等
  - 调整后，刀具半径修正 G41/G42 仅在调整后的状态下与接近块一起激活。在调整过程中，内部效应被抑制。校正不计算。

## 6.4 可编程的与距离有关的信号

### 6.4.1 一般信息

在到达编程标记之前，提前距离信号（ADS）信号设置一个信号。信号属性由系统日期 /SysAdvDistSig/Sig [] 指定。



有关更多信息，请参阅 “可编程距离相关信号” 章节中的 “IndraMotion MTX 功能说明特殊功能”。

### 6.4.2 激活搜索标记 "ADSENA"

描述： 提前距离信号启用（ADSENA）

使用高级语言语法的 NC 功能

ADSENA 应用 " /SysAdvDistSig/Sig []" 系统日期的当前内容并激活搜索已编程标记（请参阅 ADSLAB）。

|     |                                  |              |
|-----|----------------------------------|--------------|
| 语法: | ADSENA(<Signal> {,<Signal> ...}) |              |
|     | 其中                               |              |
|     | <Signal>                         | 1 .. 8: 信号数量 |

表格 6-4: 语法 ADSENA

|     |                     |
|-----|---------------------|
| 示例: | <i>SignalConfig</i> |
|-----|---------------------|

```
N00 WAIT
001 SD.SysAdvDistSig.Unit = 0 : REM mm
002 REM
003 REM Configure Signal 3
004 SD.SysAdvDistSig.Sig[3].DistOn = 1.2 : REM 1.2 mm
005 SD.SysAdvDistSig.Sig[3].DistOff = 0.0 : REM 0.0 mm
006 SD.SysAdvDistSig.Sig[3].TimeOn = 0.0 : REM 0.0 msec
007 SD.SysAdvDistSig.Sig[3].TimeOff = 0.0 : REM 0.0 msec
008 SD.SysAdvDistSig.Sig[3].Duration = 40 : REM 40 msec
009 SD.SysAdvDistSig.Sig[3].DistCalc = 0 : REM calc. dist. in active plane
010 SD.SysAdvDistSig.Sig[3].Type = 0 : REM iCh_Custom2
011 SD.SysAdvDistSig.Sig[3].Index = 2 : REM iCh_Custom2
```

PartProg

```
...
N20 ADSENA(3) ; Activate look-ahead for label ADSLAB(3)
N21 G1 X... Y... F...
N22 G2 X... Y...
N23 G1 X... Y...
...
N50 G1 X... Y... ADSLAB(3) ; Set signal iCh_Custom2 1.2 mm before end of block
N51 G1 X... Y...
...
N80 ADSDIS(3) ; Deactivate look-ahead for label ADSLAB(3)
...
```

特殊特性: 控制器复位和 M30 停用功能。

6. 4. 3 取消标记搜索 "ADSENA"

描述: 提前距离信号启用 (ADSENA)  
ADSENA 停用搜索已编程标记（请参阅 ADSLAB）。

|     |                                  |              |
|-----|----------------------------------|--------------|
| 语法: | ADSDIS(<Signal> {,<Signal> ...}) |              |
|     | 其中                               |              |
|     | <Signal>                         | 1 .. 8: 信号数量 |
|     |                                  |              |
|     | ADSDIS ()                        | 停用所有标记的搜索。   |

表格 6-5: 语法 ADSDIS

示例: 参见 "激活搜索标记 "ADSENA" "。

- 特殊特性:
- ADSDIS 不会退出信号输出。
  - 使用 ADSCLR (<Signal>) 功能不能删除开关信号。

6. 4. 4 设置标记 "ADSLAB"

描述 提前距离信号启用 (ADSENA)  
ADSLAB 是搜索的标记。



|          |           |                                  |
|----------|-----------|----------------------------------|
| 语法:      |           | ADSLAB(<Signal> {,<Signal> ...}) |
|          |           | 其中                               |
| <Signal> | 1 .. 8:   |                                  |
|          |           | 信号 "DistOn" + "TimeOn" 已在标记前方设置。 |
|          | -1 .. -8: |                                  |
|          |           | 信号 "DistOn" + "TimeOn" 已在标记前方复位。 |

表格 6-6: 语法 ADSLAB

示例: 请参阅“激活搜索标记“ADSENA””和复位开关信号“”ADSCLR””。

6. 4. 5 复位开关信号 "ADSCLR"

描述: 提前距离信号启用 (ADSCLR)  
复位切换信号。

|           |         |                                  |
|-----------|---------|----------------------------------|
| 语法:       |         | ADSCLR(<Signal> {,<Signal> ...}) |
|           |         | 其中                               |
| <Signal>  | 1 .. 8: | 信号数量                             |
|           |         |                                  |
| ADSCLR () |         | 复位所有切换信号。                        |

表格 6-7: 语法 ADSCLR

示例: *SignalConfig*

```
N00 WAIT
001 SD.SysAdvDistSig.Unit = 0 : REM mm
002 REM
003 REM Configure Signal 3
004 SD.SysAdvDistSig.Sig[3].DistOn = 1.2 : REM 1.2 mm
005 SD.SysAdvDistSig.Sig[3].DistOff = 0.0 : REM 0.0 mm
006 SD.SysAdvDistSig.Sig[3].TimeOn = 0.0 : REM 0.0 msec
007 SD.SysAdvDistSig.Sig[3].TimeOff = 0.0 : REM 0.0 msec
008 SD.SysAdvDistSig.Sig[3].Duration = -1 : REM switching signal
009 SD.SysAdvDistSig.Sig[3].DistCalc = 0 : REM calc. dist. in active plane
010 SD.SysAdvDistSig.Sig[3].Type = 0 : REM iCh_Custom2
011 SD.SysAdvDistSig.Sig[3].Index = 2 : REM iCh_Custom2
```

PartProg

```
...
N20 ADSENA(3) ; Activate look-ahead for label ADSLAB(3)
N21 G1 X... Y... F...
N22 G2 X... Y...
N23 G1 X... Y...
...
...
N50 G1 X... Y... ADSLAB(3) ; Set signal iCh_Custom2 1.2 mm before end of block
N51 G1 X... Y...
N52 G1 X... Y... ADSLAB(-3) ; Reset signal at end of block
...
N80 ADSDIS(3) ; Deactivate look-ahead for label ADSLAB(3)
...
```

“持续 = -1”将信号 3 定义为切换信号。  
如果零件程序“PartProg”被块 N51 中的复位取消，则信号 3 并因此 iCh\_Custom2 保持置位状态。如果 ADSCLR () 命令包含在初始化字符串或 init 程序中，则系统复位会重置信号。

使用高级语言语法的 NC 功能

6.5 扩展正向回程程序块 "AdvanceAppend, ADVA"

**描述:** 对于反向操作，将为回程区域回程（1）.. 回程（0）中的每个零件程序块生成一个正向程序块。功能 "AdvanceAppend" 在当前正向程序块的末尾插入指定的 NC 功能。

|                               |        |
|-------------------------------|--------|
| 语法:                           |        |
| AdvanceAppend (<NCF>...<NCF>) |        |
| 缩写形式: ADVA (...)              |        |
| 其中                            |        |
| <NCF>...<NCF>                 | 数控函数列表 |

表格 6-8: 语法 AdvanceAppend (ADVA)



轮廓更改会导致反向操作中的切换错误。  
不允许 CPL 功能和子例程调用。

|                          |                    |                               |
|--------------------------|--------------------|-------------------------------|
| 示例:                      |                    |                               |
| 零件程序:                    | 正向程序:              | 注意:                           |
| N100 X100 M10            | N1 X=100           | 辅助功能不适用                       |
| N200 X0 ADVA (M125 M126) | N2 X = 0 M125 M126 | 在正向程序中，M125 和 M126 将被输出到 PLC。 |



为了检查目的，将修改后的正向程序保存在文件中（请参阅 第 6.103 章 "回程、RETR" 第 311 页）。

6.6 替换回程正向程序块 "AdvanceSubstitute, ADVS"

**描述:** 对于反向操作，将为回程区域回程（1）.. 回程（0）中的每个零件程序块生成一个正向程序块。"AdvanceSubstitute" 功能用指定的 NC 功能替换当前的正向程序块。

|                                  |        |
|----------------------------------|--------|
| 语法:                              |        |
| AdvanceSubstutue (<NCF>...<NCF>) |        |
| 缩写形式: ADVS (...)                 |        |
| 其中                               |        |
| <NCF>...<NCF>                    | 数控函数列表 |

表格 6-9: 语法 AdvanceSubstitute (ADVS)



轮廓更改会导致反向操作中的切换错误。  
不允许 CPL 功能和子例程调用。

|                          |            |                          |
|--------------------------|------------|--------------------------|
| 示例:                      |            |                          |
| 零件程序:                    | 正向程序:      | 注意:                      |
| N5 X0 F10 ADVS (X0 F100) | N1 X0 F100 | 在前馈中允许更高的进给。始终插入编程的终点坐标！ |



为了检查目的，将修改后的正向程序保存在文件中（请参阅 第 6.103 章 "回程、RETR" 第 311 页）。

## 6.7 区域监控 "区域，ARA"

**描述:** 它定义，激活或关闭多达 32 个矩形，二维死区或具有轴平行边界的工作空间，并通过线性 and 圆形运动监控其交点。监控仅在激活的平面中进行，仅集中于机床和轴坐标。("Area", "ARA") 函数并不用于监控工件坐标。



若要监控工件坐标，例如，在刀具长度修正、坐标旋转或缩放已激活的情况下，必须使用 "IndraMotion MTX 14VRS Functional Description - Special Functions"中所述的监控功能"系统数据的范围监控"。  
所述功能独立于 ("Area", "ARA") 函数！

- **不工作区域:**  
它们在移动运动（包括它们的边界）期间既不可以交叉也不可碰触。
- **工作区:**  
移动运动期间不可退出（包括其边界）。



区域 1-10 的默认数据是使用机床参数组 MP 8002 定义的。  
如果定义了可选的系统日期 SysARACfg，则机床参数组 MP 8002 处于非活动状态：所有区域的默认数据必须在 SysARACfg 中配置/预先分配。

**静态区域:** 默认情况下，区域是静态机器固定的。检测到的区域违规会导致错误并在取消的程序中。只监视活动区域（ ->监视状态）。

对于静态区域，定义（可选）系统日期 SysARACfg。所有预先分配都可以在系统日期完成。由此，不再需要通过机器参数进行配置。

**动态地区:** 在程序处理期间动态区域可以变为活动或不活动（释放），例如，通过机械回溯。  
类似于静态区域监视动态区域。  
只监视活动区域（ ->监视状态）。  
以下（可选）系统数据与动态区域相关，需要定义：

- SysARACfg
- SysARA
- SysARAAdm

"动态" 属性（区域）是一个可选的可配置属性，只能在每个区域 <i> 的 SysARACfg.DynamicArea [<i>]系统数据中指定。

0: 静态区域

- 1: 动态区域，只有在没有其他动态区域无效的情况下才能被无效切换
- 2: 动态区域，只有在没有其他动态区域无效的情况下才能被无效切换

如果动态区域处于活动状态，PLC 必须在执行时通过 SysARA.PlcAreaDisabled [<i>] 系统日期指定：

- 0: 范围已激活
- 1: 范围不活动 / 已释放

检测到的违反动态的区域不会导致取消程序，而是在临界移动运动之前停止运动。输出警告（可用 SysARAAdm.MsgDisable\_5511=1 抑制警告）。停止移动运动，直到受影响范围被释放，PLC 报告消息。

"违反的区域" 状态在系统日期 (SysARA.AreaViolationOccured [i]=1) 中的标识符的关键 NC 块的执行点发出信号。在应用程序 (PLC->系统数据接口 SysARA.PlcAreaDisabled [i]=1) 中启用（不活动）受影响的区域后，将执行

使用高级语言语法的 NC 功能

关键运行动作并重置“违反状态区域”（SysARA.AreaViolationOccured [的<i>] = 0）。

如果在达到临界横移运动之前检测到 SysARA.PlcAreaDisabled [i] = 1，则运动在没有通信的情况下执行。

为了能够在边界停止，如果需要，速度会减速。

建议选择该区域的扩展以避免轴振荡时发生碰撞。

语法：**激活，禁用或释放一个或全部监控区域即刻。**

|                      |                                               |
|----------------------|-----------------------------------------------|
| Area (<BNo> ,<Sta> ) |                                               |
| 缩写形式: ARA (... )     |                                               |
| 其中                   |                                               |
| <BNo>                | 区域编号<br>整数。输入范围: -1、1...32<br>-1: 激活 / 取消所有区域 |
| <Sta>                | 所需的监控状态<br>0: 取消监控<br>1: 激活监控<br>-1: 取消和释放监控  |

表格 6-10:        语法 1 区域 (ARA)

定义、启用、取消和释放一个监控区域。

|                                                                                              |                                                                                                                              |
|----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Area (<BNo> ,<Sta>{ ,<Mod> , {<P1> } , {<P2> } , {<D1> } , {<D2> } , {<Ax1> } , {<Ax2> } } ) |                                                                                                                              |
| 缩写形式: ARA (... )                                                                             |                                                                                                                              |
| 其中                                                                                           |                                                                                                                              |
| <BNo>                                                                                        | 区域编号。<br>整数。输入范围: 1...32。                                                                                                    |
| <Sta>                                                                                        | 所需的监控状态<br>0: 禁用 <BNo> 范围的监控<br>1: 启用 <BNo> 范围的监控<br>-1: 禁用 <BNo> 范围的监控, 释放 <BNo> 范围                                         |
| <Mod>                                                                                        | 0: <BNo> 范围未使用<br>1: <BNo> 范围为不工作区域<br>2: <BNo> 范围在工作范围                                                                      |
| <P1>                                                                                         | 在机床坐标系中的位置<br>指定区域中心点的位置值, 参考区域中涉及的第一个系统轴 (使用 MP 8002 00001 定义 (可选: SysARACfg.Pos1) 或用 Ax1 编程), 固定。<br>请参见以下示例<br>编程单位与轴坐标相同 |

|       |                                                                                                                                  |
|-------|----------------------------------------------------------------------------------------------------------------------------------|
| <P2>  | 在机床坐标系中的位置<br><br>指定区域中心点的位置值，参考区域中涉及的第二个系统轴（使用 MP 8002 00002（可选：SysARACfg.Pos2）或用 Ax2 编程），固定。<br><br>参考下面的示例。<br><br>编程单位与轴坐标相同 |
| <D1>  | 指定区域的长度，参考区域中涉及的第一个系统轴（使用 MP 8002 00001 定义（可选：SysARACfg.Ext1）或用 Ax1 编程），固定。参考下面的示例。<br><br>编程单位与轴坐标相同。                           |
| <D2>  | 指定区域的长度，参考区域中涉及的第二个系统轴（使用 MP 8002 00002 定义（可选：SysARACfg.Ext2）或用 Ax2 编程），固定。参考下面的示例。<br><br>编程单位与轴坐标相同                            |
| <Ax1> | 指定区域的第一个系统轴（可选：SysARACfg.AreaSysAxisNoDim1）<br>系统轴号                                                                              |
| <Ax2> | 指定区域的第二个系统轴（可选：SysARACfg.AreaSysAxisNoDim2）<br>系统轴号                                                                              |

表格 6-11:            语法 2 区域 (ARA)



若程序执行时已输入非编程值，则会保留这些值。  
  
如果在程序运行过程中未指定值，则控制器使用范围 1-10 的机床参数组 8002 的相应值，不分配范围 11-32 的值（可选：预赋值 SysARACfg）。

示例:

|      |      |                                                    |                                                                             |
|------|------|----------------------------------------------------|-----------------------------------------------------------------------------|
| N100 | Area | 取消区域 4，并将该区域的中心点设置为给定的机床坐标 (4,0,,100,200)          | 区域 (100,200)。<br><br>区域的长度保持不变。                                             |
| N120 | 区域   | 激活带有类型工作范围的区域 12，并将区域中心点设置为 (12,1,2,100,200,,,1,3) | 指定的机床坐标 (100,200)。<br><br>区域的长度保持不变。<br><br>该区域的第一个系统轴: 1. 轴。区域的第二系统轴: 3. 轴 |
| N150 | 区域   | (12,-1)                                            | 取消激活区域 12 并将其激活用于其他通道                                                       |

使用高级语言语法的 NC 功能

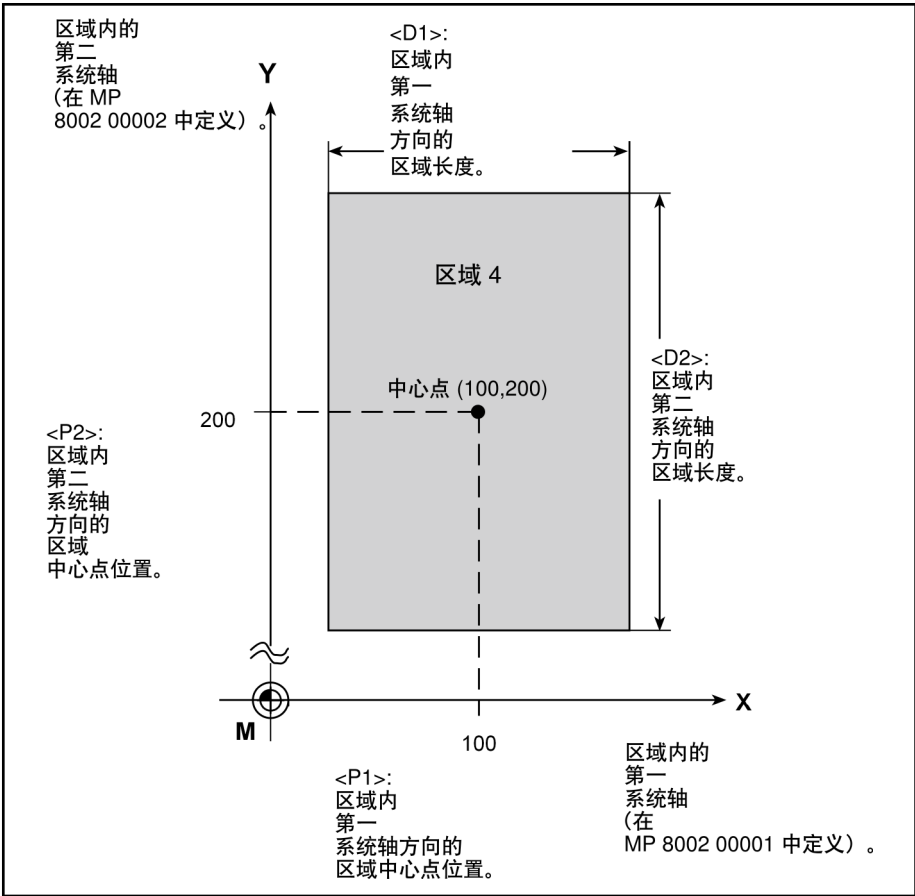


插图 6-1: 区域 (ARA) 示例

特殊功能和限制:

- 仅监控轴和机床坐标。无法监控工件坐标。
- 以下内容适用于 1-10 区域: 涉及各个区域的所有系统轴必须在 MP 8002 00001 和 MP 8002 00002 中进行定义, 或者必须使用 Ax1 和 Ax2 进行编程。
- 以下内容适用于 11-32 区域: 所有属于各个区域的系统轴必须使用 Ax1 和 Ax2 进行编程。
- 要使用区域影响“区域”1-10, 必须相应地设置 MP 8002 00032。
- 该函数需要参照轴。
- 在某个区域激活后, 对应轴必须位于当前通道。
- 通过“区域”影响区域只在“区域”编程的通道中有一个区域。如果一个区域的轴被转移到另一个通道, 则该区域被提供预定义在机器参数组 8002 中的目标通道的默认值, 并且该区域被停用。任何通过源通道中的“区域”编程的值都不会被传输。
- 每个系统都是唯一的, 每个系统只存在一次, 只能通过一个频道进行修改。如果一个区域将被另一个通道使用, 则该区域必须首先由第一通道激活
- 在点动模式中: 不会监控通过手轮移动的轴。
- 在点动模式中: 每次仅会点动一个跨越其中一个不工作区域的轴。
- 点动模式下的区域超出会引发警告。对应轴不会移动, 且仅可朝反方向将之点动。
- 若区域的尺寸在激活平面内, 则会监控线性与圆周运动。
- 若区域的尺寸在激活平面外, 则仅监控该区域的线性运动。

- 若区域尺寸有部分在活动平面外，则仅监控该区域的线性运动。无法监控圆周运动，并生成一条警告消息（可抑制该警告：SysARAAdm.MsgDisable\_3732=1 - 输出警告之前验证系统数据）。
- 包含的轴位于监控通道内的线性进给运动将受到监控。
- 若相关监控区域位于激活平面内，且具有两个轴尺寸，则会监控圆周进给运动。
- 若圆周运动时，监控区域尺寸均位于活动平面外，则不监控该区域。若该区域仅有一个尺寸在激活平面内，则会显示一条警告消息，说明不会监控该运动。

动态保护区的系统数据

保护区的系统数据是易失性的使用 SDDatAra.xml 文件预配置数据。  
如何定义保护区的系统数据：  
将 "feprom/schemas/sdara.xsd\_" 文件复制到 "usrfep/schemas/sdara.xsd" 并将它们集成到活动定义文件（例如 usrfep/SDDefMtb.xml）中。

```
<Variable Dimension="32" Storage="volatile">
<Name>SysARA</Name>
<Type>ARA_t</Type>
```

创建 ARA 系统数据：

```
...
</Variable>
<Variable Dimension="32" Storage="volatile">
<Name>SysARACfg</Name>
<Type>ARACfg_t</Type>
...
</Variable>
<Variable Storage="volatile">
<Name>SysARAAdm</Name>
<Type>ARAAdm_t</Type>
....
</Variable>
```

这些定义的模板存储在 feprom/SDDef0em.xml 中

6.8 异步子例程

6.8.1 异步子例程：登出 "ASPCLR"

描述： 在当前通道中登出异步子例程。  
被登出的子例程将无法启用或取消。



要登录已注销的异步子例程，请参阅函数 "ASPSET"，第 6.8.5 章 "异步子程序：注册 ASPSET" 第 195 页。



有关异步子例程的使用和参数化的全面信息，请参阅文档 "IndraMotion MTX 功能描述"。

使用高级语言语法的 NC 功能

|     |                   |                        |
|-----|-------------------|------------------------|
| 语法: | ASPCLR (<Up-No> ) |                        |
|     | 其中                |                        |
|     | <Up-No>           | 子例程号<br>输入范围: 1...8 整数 |

表格 6-12: ASPCLR 语法

- 特殊功能和限制:
- 须在目标通道中登入对应子例程（参见 ASPSET，第 6.8.5 章 “异步子程序: 注册 ASPSET” 第 195 页）。
  - 若要临时取消某个子例程，请参见功能 “ASPDIS”，第 6.8.2 章 “异步子例程: 关闭 ASPDIS” 第 194 页。

6.8.2 异步子例程: 关闭 "ASPDIS"

描述: 在当前通道中取消异步子例程。  
不会在相关事件发生时调用被取消的子例程。



有关异步子例程的使用和参数化的全面信息，请参阅文档 “IndraMotion MTX 功能描述”。

|     |                                |                                                                                                    |
|-----|--------------------------------|----------------------------------------------------------------------------------------------------|
| 语法: | ASPDIS (<Up-No> {, <IrUp-No>}) |                                                                                                    |
|     | 其中                             |                                                                                                    |
|     | <Up-No>                        | 子例程号<br>输入范围: 1...8 整数                                                                             |
|     | <IrUp-No>                      | 可中断的异步子例程（可选）：<br>未指定：<br>- 锁定零件程序的中断<br>可中断的异步子例程的编号：<br>输入范围: 1..8，整数<br>- 锁定异步子例程 <IrUp-No> 的中断 |

表格 6-13: ASPDIS 语法

- 特殊功能和限制:
- 须在目标通道中登入对应子例程（参见 ASPSET，第 6.8.5 章 “异步子程序: 注册 ASPSET” 第 195 页）。
  - 若要激活某个被取消的子例程，请参见功能 “ASPENA”，第 6.8.3 章 “异步子例程: 打开 ASPENA” 第 194 页。

6.8.3 异步子例程: 打开 "ASPENA"

描述: 在当前通道中激活的异步子例程。  
只有在发生相关事件时才调用激活的子例程。



有关异步子例程的使用和参数化的全面信息，请参阅文档 “IndraMotion MTX 功能描述”。

|     |                                |  |
|-----|--------------------------------|--|
| 语法: | ASPENA (<Up-No> {, <IrUp-No>}) |  |
|     | 其中                             |  |



|           |                                                                                                   |
|-----------|---------------------------------------------------------------------------------------------------|
| <Up-No>   | 子例程号<br>输入范围：1...8 整数                                                                             |
| <IrUp-No> | 可中断的异步子例程（可选）：<br>未指定：<br>- 开锁零件程序的中断<br>可中断的异步子例程的编号：<br>输入范围：1..8，整数<br>- 开锁异步子例程 <IrUp-No> 的中断 |

表格 6-14: ASPENA 语法

- 特殊功能和限制:
- 须在目标通道中登入对应子例程（参见 ASPSET，第 6.8.5 章“异步子程序：注册 ASPSET”第 195 页）。
  - 若要取消某个子例程，请参见功能“ASPDIS”，第 6.8.2 章“异步子例程：关闭 ASPDIS”第 194 页。

6.8.4 异步子程序：定义返回到轮廓点 "ASPRTP"

- 描述:
- 定义在一个可以偶尔中断的运行块的异步子程序之后，控制位置定位到:
- 起始点
  - 终点
  - 中断点

若中断时没有进给块，则控制器将始终停留在最后激活的坐标。



有关异步子例程的使用和参数化的全面信息，请参阅文档“IndraMotion MTX 功能描述”。

语法:

|                            |                                                               |
|----------------------------|---------------------------------------------------------------|
| ASPRTP (<Up-No> , <Point>) |                                                               |
| 其中                         |                                                               |
| <Up-No>                    | 子例程号<br>输入范围：-1; 1...8。整数<br>-1:<br>定义当前通道的所有异步子例程的返回到所需的轮廓点。 |
| <Point>                    | 所需的返回工件加工位置点：<br>1: 起始点<br>2: 终点<br>3: 中断点                    |

表格 6-15: ASPRTP 语法

- 特殊功能和限制:
- 通过复位或者 M30 可以清除所需的返回到轮廓点。
  - 在异步子程序中进行的任何校正更改都将自动包含在轮廓所需返回点的内部计算中。
  - 可以使用“REPOSTP”功能（请参见第 6.102.2 章“定义异步子例程的路径返回点 REPOSTP”第 310 页）临时抑制异步子例程中定义的返回到轮廓点。

6.8.5 异步子程序：注册 "ASPSET"

- 描述:
- 在当前通道中登入异步子例程，并将其激活（对于激活，另请参见功能“ASPENA”，第 6.8.3 章“异步子例程：打开 ASPENA”第 194 页）。

只能使用已经登入并激活的子例程。

 要登出已登入的异步子例程，请参阅函数“ASPCLR”，第 6.8.1 章“异步子例程：登出 ASPCLR”第 193 页。

 有关异步子例程的使用和参数化的全面信息，请参阅文档“IndraMotion MTX 功能描述”。

|     |                                         |                                                                                                   |
|-----|-----------------------------------------|---------------------------------------------------------------------------------------------------|
| 语法： | ASPSET (<Up-No> , <Up-Name>{, <Flags>}) |                                                                                                   |
|     | 其中                                      |                                                                                                   |
|     | <Up-No>                                 | 子例程号<br>输入范围：1...8 整数                                                                             |
|     | <UP name>                               | 子例程名称<br>含有 / 不含路径规范<br>若路径缺失，则会依据参数 NC0/FileOrg/SrchPathSubProg (MP 3080 00001) 来搜索文件（搜索子例程的路径）。 |
|     | <Flags>                                 | 调用后可用于影响行为的开关：<br>00：既无链接，也无自动启动。<br>10：链接，但没有自动启动。<br>01：无链接，只有自动启动。<br>11：链接和自动启动。              |

表格 6-16: ASPSET 语法

示例：

```
:
N30 作为激活通道中的第一个异步子例程，登录程序 ASUP1。在其
ASPSET (1,ASUP1,1 调用之后（例如通过接口信号），该子程序 - 如果需要的话
0) - 自动链接并通过 NC 启动显式启动。
:
```

6.8.6 异步子程序：通过程序 "ASPSTA"触发

描述： 在任意通道中调用（受程序控制的）异步子例程。

 有关异步子例程的使用和参数化的全面信息，请参阅文档“IndraMotion MTX 功能描述”。

|     |                                    |                                                            |
|-----|------------------------------------|------------------------------------------------------------|
| 语法： | ASPSTA (<Up no.>{, <Channel No.>}) |                                                            |
|     | 其中                                 |                                                            |
|     | <Up-No>                            | 子例程号<br>输入范围：1...8 整数                                      |
|     | <Channel number>                   | <Channel no> 将调用 <Up No> 的目标通道<br>如果尚未编程，则在当前通道调用 <Up No>。 |

表格 6-17: ASPRTP 语法

- 特殊功能和限制：
- 不得嵌套异步子例程。
  - 须在目标通道中登入调用的子例程（参见 ASPSET，第 6.8.5 章“异步子程序：注册 ASPSET”第 195 页）。

- 不可在目标通道中取消调用的子例程（参见 ASPDIS，第 6.8.2 章“异步子例程：关闭 ASPDIS”第 194 页或 ASPENA，第 6.8.3 章“异步子例程：打开 ASPENA”第 194 页）。

## 6.9 赋予逻辑轴名称 "AssLogName, ALN"

**描述：**在调用通道中为同步轴赋予新的逻辑轴名称。旧的逻辑轴名称将无效。



有关“轴转换”功能性的详细信息，请参阅文档“IndraMotion MTX 功能描述”。

**语法：**

|                                                                                  |                                                                                                                                                                                                           |
|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AssLogName (<PAN> <PAI> <LAN>, <LANnew><br>{ , <PAN> <PAI> <LAN>, <LANnew>}... ) |                                                                                                                                                                                                           |
| 缩写形式：ALN (...)                                                                   |                                                                                                                                                                                                           |
| 其中                                                                               |                                                                                                                                                                                                           |
| <PAN>                                                                            | 物理轴名称（系统轴名称）<br>指定当前通道中需要重命名的轴。                                                                                                                                                                           |
| <PAI>                                                                            | 物理轴索引（系统轴索引）<br>效果与 <PAN> 相同                                                                                                                                                                              |
| <LAN>                                                                            | 逻辑轴名称（通道轴名称）<br>效果与 <PAN> 相同                                                                                                                                                                              |
| <LANnew>                                                                         | 新逻辑轴名称<br>通过 <PAN>、<PAI> 或 <LAN> 指定的轴，在当前通道中分配的逻辑名为 <LANnew>。<br><LANnew> 必须在参数 CHAN/Ch/Coord/Acs/ChAx/ChAxName (MP 7010 00010)（通道轴名称）或参数 CHAN/Ch/Coord/Acs/OptAxName/OptChAxName (MP 7010 00020)（可选轴名称）。 |

表格 6-18: AssLogName (ALN) 语法

**示例：**

```
:
N030 ALN (YP, X, 向系统轴 YP 分配通道轴名称（逻辑名称）X，向第三系统轴分
3, Y, B, Z) 配通道轴名称 Y。向通道轴 B 分配通道轴名称（逻辑名称）
: Z。
对 B 的后续编程生成一个运行时错误。
```

- 特殊功能和限制：**
- 要重命名的轴必须处于静止状态。否则，控制器会生成错误消息，并终止程序。
  - 相同块中的轴位置必须始终根据“AssLogName (...)”表达式进行编程。

## 6.10 校准轴运动

### 6.10.1 后向转换 "ATBWD"

**描述：**用于以实轴位置（ACS）转换工件坐标系（WCS）的坐标。该过程被称为“后向转换”。

**语法：**

|                            |
|----------------------------|
| AtBwd (<AxCoord>, <Coord>) |
| 其中                         |

使用高级语言语法的 NC 功能

|           |                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------|
| <AxCoord> | 永久、全局或局部 CPL 数组。<br>类型：Double<br>尺寸：<br>至少与当前通道中轴的数量相同。<br>调用子例程后，数组的各变量包含 ACS 中的结果实轴位置。            |
| <Coord>   | 永久、全局或局部 CPL 数组。<br>类型：Double<br>尺寸：<br>至少与轴变换生成的坐标系中通道坐标的数量相同。<br>数组的变量必须包含涉及轴变换的 WCS 中的所有坐标的工件坐标。 |

表格 6-19:        *AtBwd* 语法

**特殊功能和限制:** 在调用时处于激活的轴变换将进行后向变换。所需的长度和角度参数取自机床参数。

6. 10. 2    优化参数 "ATCAL"

**描述:**        结合“校准轴运动”函数，该函数用于优化特定于轴运动的角度和长度参数。这些数据随相应的轴转换类型而变化，并且包含在每个轴转换的参数 TRA/AxTrafo/JointParTrafo (MP 1030 00140) 中。通常，必须在优化之前使用功能“ATGET”（参见 第 6.10.4 章“从 NC ATGET 读取参数”第 200 页）读取，并在使用功能“ATPUT”（参见 第 6.10.5 章“向 NC ATPUT 写入参数”第 201 页）进行优化后返回到 NC。



有关轴运动学校准的全面信息，请参阅文档“IndraMotion MTX”功能描述。

**语法:**

|                                                   |                                                                                                                                                                                                                    |
|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ATCAL(<File>,<OptData>,<Mask>{,<Info>}{,<NumIt>}) |                                                                                                                                                                                                                    |
| 其中                                                |                                                                                                                                                                                                                    |
| <File>                                            | 校准文件的名称含有或不含路径规范。<br>若路径缺失，则会依据参数 NCO/FileOrg/SrchPathSubProg (MP 3080 00001) 来搜索文件（搜索子例程的路径）。<br><File> 包含优化所需的数据。有关更多信息，请参阅文档“IndraMotion MTX 功能描述”。                                                             |
| <OptData>                                         | 永久、全局或局部 CPL 数组。<br>类型：Double<br>尺寸：最小 16<br>计算后，数组会有一个长度和角度参数经过优化的参数块。<br>各个变量（索引 1 至 16）的顺序对应于参数 TRA/AxTrafo/JointParTrafo (MP 1030 00140) 的元素索引。这些值与参数 TRA/AxTrafo/JointParTrafo (MP 1030 00140) 中的各个参数具有相同的单位。 |
| <Mask>                                            | CPL 变量<br>类型：整数<br>用于确定要优化的各个参数的位掩码。<br>示例：                                                                                                                                                                        |

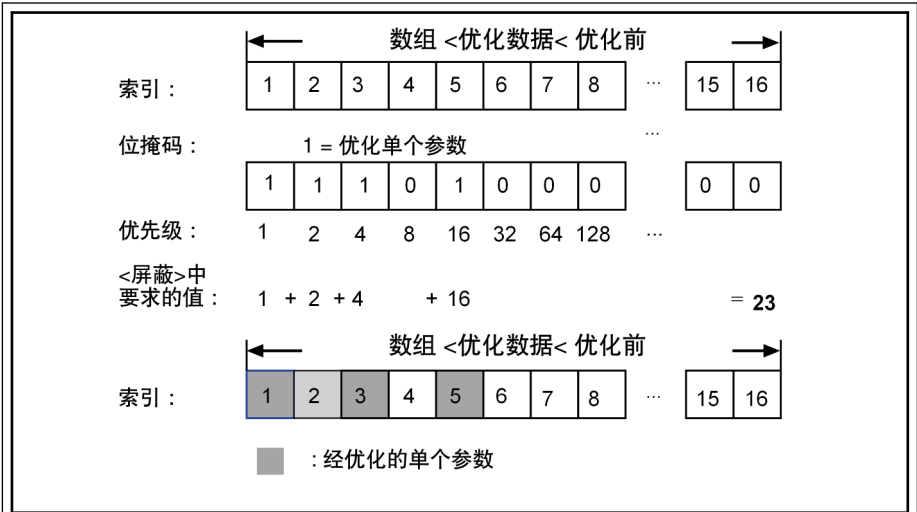


插图 6-2: 位掩码

|         |                                                                                                                                                                                                        |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Info>  | 永久、全局或局部 CPL 数组。<br>类型：双精度型。尺寸：最小 4<br>优化后将包含以下数据：<br><Info>[1]<br>优化前的准则函数值（二次偏差）。<br>单位：mm<br><Info>[2]<br>优化后的准则函数值（二次偏差）。<br>单位：mm<br><Info>[3]<br>优化前的最大偏差（单位 mm）<br><Info>[4]<br>优化后的最大偏差（单位 mm） |
| <NumIt> | 优化迭代步骤的最大数量。<br>若未输入，或其值为“-1”，将仅在两个连续计算的参数块降至足够低的内部 NC 阈值后，才会终止迭代（相关的偏差不再存在）。<br>若输入“1”，则为线性的调整计算。                                                                                                     |

表格 6-20: ATCAL 句法

|                                              |                                               |
|----------------------------------------------|-----------------------------------------------|
| 01 DIM PAR!(16)                              | 使用双精度类型的 16 个数组的局部 CPL 数组尺寸（适用于经过优化的长度/角度参数）。 |
| 02 DIM GA!(4)                                | 使用双精度类型的 4 个数组的局部 CPL 数组尺寸（适用于判别和偏差）。         |
| 03 MASK%=2+4+32                              | 含 2、3 和 6 数组索引的各个参数的优化。                       |
| N4 ATCAL(CL.TXT,[PAR!],<br>[MASK%],[GA!],-1) | 开始优化                                          |

6. 10. 3 前向转换 "ATFWD"

描述： 结合“校准轴运动”功能，本功能用于将实轴位置转换为由轴变换生成的坐标系坐标。该过程也被称为“前向转换”。

使用高级语言语法的 NC 功能



有关轴运动学校准的全面信息，请参阅文档“IndraMotion MTX”功能描述。

语法:

|                                       |                                                                                                                                                                                                                                                                                   |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ATFWD (<Coord>,<Axcoord>{,<ParData>}) |                                                                                                                                                                                                                                                                                   |
| 其中                                    |                                                                                                                                                                                                                                                                                   |
| <Coord>                               | 永久、全局或局部 CPL 数组。<br>类型: Double<br>尺寸:<br>至少与轴变换生成的坐标系中通道坐标的数量相同。<br>转换后数组的各个变量包含结果已变换坐标系中的通道坐标。                                                                                                                                                                                   |
| <AxCoord>                             | 永久、全局或局部 CPL 数组。<br>类型: Double<br>尺寸:<br>至少与当前通道中轴的数量相同。<br>数组的变量必须包含所有涉及轴变换的通道轴的实轴位置。                                                                                                                                                                                            |
| <ParData>                             | 永久、全局或局部 CPL 数组。<br>类型: Double<br>尺寸: 最小 16<br>数组必须包含具有所有长度与角度参数的参数块以用于轴变换。<br>各个变量（索引 1 至 16）的顺序对应于 MP 1030 00140 的元素索引。这些值必须与机床参数 1030 00140 中的各个参数具有相同的单位。<br>如果未如此设定，则使用当前处于活动状态的轴变换的参数块。<br>若控制器中同时有两个轴变换（AT1、AT2；参见功能“Coord”（第 6.28 章“选择轴转换 Coord、CRD”第 225 页），则使用 AT2 的数据。 |

表格 6-21: ATFWD 句法

特殊功能和限制: 若 <ParData> 已经过编程，且无激活的轴变换，会显示运行时错误消息。

6.10.4 从 NC "ATGET"读取参数

描述: 结合“校准轴运动”功能，该功能用于读取特定于轴变换的机床参数。  
对于各类轴变换，该数据会有所不同。通常，必须在优化之前读取（参见“ATGET”，第 6.10.2 章“优化参数 ATCAL”第 198 页），并在使用功能 ATPUT（参见第 6.10.5 章“向 NC ATPUT 写入参数”第 201 页）进行优化后返回到 NC。



有关轴运动学校准的全面信息，请参阅文档“IndraMotion MTX”功能描述。

语法:

|                                               |
|-----------------------------------------------|
| AtGet (<ParData>{,<AxTrafoNo>}{,<MachParNo>}) |
| 其中                                            |

|             |                                                                                                                                                                                                                                  |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ParData>   | 永久、全局或局部 CPL 数组。<br>类型: Double<br>尺寸:<br>最小 8 (MachParNo = 1030 00130) 或<br>最小 16 (MachParNo = 1030 00140) 。<br>执行功能后, 数组包含轴变换 "AxTrafoNo" 的机床参数<br>"MachParNo"的当前有效值。<br>各变量的顺序 (索引 1、2...) 对应着各机床参数的元素索引。其<br>值具有与相应机床参数相同的单位。 |
| <AxTrafoNo> | 要读取其参数的轴变换的编号。<br>若其编号未设定, 则读取当前激活的轴变换的数据。<br>若控制器中同时有两个轴变换 (AT1、AT2; 参见功能 "Coord"<br>( 第 6.28 章 "选择轴转换 Coord、CRD" 第 225 页 ), 则使<br>用 AT2 的数据。                                                                                   |
| <MachParNo> | 读取的机床参数的编号。<br>若该编号未编程, 将使用默认值 1030 00140, 即检索长度和角度参<br>数。                                                                                                                                                                       |

表格 6-22:        AtGet 句法

|     |                                    |                                                                                                                                                                                                             |
|-----|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 示例: | 01 DIM LENPARAM! (16)              | 含长度和角度参数 16 元素的局部 CPL 数组。                                                                                                                                                                                   |
|     | 01 DIM ZEROPOS! (8)                | 含有 8 个元素的局部 CPL 数组适用于通道轴的参考位置。                                                                                                                                                                              |
|     |                                    | 将读取下列值:                                                                                                                                                                                                     |
|     | N1 ATGET (LENPARAM!)               | 当前轴变换的 MP 1030 00140                                                                                                                                                                                        |
|     | N1 ATGET (LENPARAM!, 3)            | 轴变换 3 的 MP 1030 00140                                                                                                                                                                                       |
|     | N1 ATGET (LENPARAM!, 2, 103000140) | 轴变换 2 的 MP 1030 00140                                                                                                                                                                                       |
|     | N1 ATGET (ZEROPOS!,, 103000130)    | 当前轴变换的 MP 1030 00130                                                                                                                                                                                        |
|     | N1 ATGET (ZEROPOS!, 3, 103000130)  | 轴变换 3 的 MP 1030 00130                                                                                                                                                                                       |
|     | 特殊功能和限制:                           | <ul style="list-style-type: none"><li>• McdParamNo 的有效值为 1030 00130 (参考位置的轴位置) 和 1030 00140 (长度和角度参数)。</li><li>• 如果 "AxTrafoNo" 未编程且未激活任何轴变换 (COORD(0, 2) 和 COORD(0, 1)), 则 "ATGET" 命令将导致发生运行时错误。</li></ul> |
|     |                                    |                                                                                                                                                                                                             |

6. 10. 5    向 NC "ATPUT"写入参数

**描述:**        结合 "校准轴运动" 功能, 该功能用于覆盖特定于轴变换的机床参数。  
对于各类轴变换, 该数据会有所不同。通常, 在优化之前必须先读取 (请参阅功能 "ATCAL", 第 6.10.2 章 "优化参数 ATCAL" 第 198 页 ) (请参阅功能 "ATGET", 第 6.10.4 章 "从 NC ATGET 读取参数" 第 200 页 ), 并在使用 "ATPUT"功能进行优化后返回到 NC。



有关轴运动学校准的全面信息, 请参阅文档 "IndraMotion MTX" 功能描述。

使用高级语言语法的 NC 功能

语法:

|                                                   |                                                                                                                                                                                                                   |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AtPut (<ParData> {, <AxTrafoNo>} {, <MachParNo>}) |                                                                                                                                                                                                                   |
| 其中                                                |                                                                                                                                                                                                                   |
| <ParData>                                         | 永久、全局或局部 CPL 数组。<br>类型: Double<br>尺寸:<br>最小 8 (MachParNo = 1030 00130) 或最小 16 (MachParNo = 1030 00140)。<br>该数组必须含有轴变换 “AxTrafoNo” 的机床参数 “MachParNo” 的完整参数块。<br>各变量的顺序 (索引 1、2...) 对应着各机床参数的元素索引。其值具有与相应机床参数相同的单位。 |
| <AxTrafoNo>                                       | 要写入其参数的轴变换编号。<br>若其编号未编程, 则覆盖当前激活的轴变换的数据。<br>若控制器中同时有两个轴变换 (AT1、AT2; 参见功能 “Coord” (第 6.28 章 “选择轴转换 Coord、CRD” 第 225 页)), 则修改 AT2 的数据。                                                                             |
| <MachParNo>                                       | 覆盖的机床参数编号。<br>若该编号未编程, 将使用默认值 1030 00140, 即更改长度和角度参数。                                                                                                                                                             |

表格 6-23: AtPut 句法

示例:

|                                    |                                |
|------------------------------------|--------------------------------|
| 01 DIM LENPARAM! (16)              | 含长度和角度参数 16 元素的局部 CPL 数组。      |
| 01 DIM ZEROPOS! (8)                | 含有 8 个元素的局部 CPL 数组适用于通道轴的参考位置。 |
| 01 LENPARAM! (1) = 123.456         | 赋值                             |
| :                                  |                                |
| 01 ZEROPOS! (8) = 1.0              | 将覆盖以下数值:                       |
| N1 ATPUT (LENPARAM!)               | 当前轴变换的 MP 1030 00140           |
| N1 ATPUT (LENPARAM!, 3)            | 轴变换 3 的 MP 1030 00140          |
| N1 ATPUT (LENPARAM!, 2, 103000140) | 轴变换 2 的 MP 1030 00140          |
| N1 ATPUT (ZEROPOS!,, 103000130)    | 当前轴变换的 MP 1030 00130           |
| N1 ATPUT (ZEROPOS!, 3, 103000130)  | 轴变换 3 的 MP 1030 00130          |

- 特殊功能和限制:
- McdParamNo 的有效值为 1030 00130 (参考位置的轴位置) 和 1030 00140 (长度和角度参数)。
  - 如果 AxTrafoNo 不存在且未激活轴变换 (COORD(0.2) 和 COORD(0.1)), 则 “ATPUT” 命令将导致发生运行时错误。
  - 执行 “ATPUT” 功能需要准许进行机床参数覆盖的 NC 用户级别。
  - 仅当系统复位 和 重新编程 COORD (<AxTrafoNo>) 之后, 才能激活写入的参数块。



## 6.11 执行所有组的激活辅助功能 "AUXFUNC"

**描述:** 执行所有当前激活的现有辅助功能组特定于通道、及跨通道的辅助功能。

该功能与块预运行相关（例如处理中止后），以恢复某个零件程序位置的所有辅助功能状态。

例如，如果零件程序在加工过程中提前终止，则可以使用块预运行重新开始中断时正在处理的块中的处理。

虽然零件程序在开始时使用块预运行重新启动，但是机床的加工将从所定义的块开始。

由于控制器既不插入也不提供块预运行时的编程辅助功能，所有辅助功能状态必须在块预运行结束时恢复。



有关辅助功能的使用和参数设置的全面信息，请参阅文档 "IndraMotion MTX 功能描述"。

**语法:**

|         |
|---------|
| AUXFUNC |
|---------|

表格 6-24: AUXFUNC 句法

**示例:**

|           |               |
|-----------|---------------|
| N100 T102 | 选择刀具 T102     |
| N110 M6   | 更换刀具 T102     |
| :         |               |
| N150 M3   | 1. 主轴 / 主轴组激活 |
| :         | 顺时针旋转         |

在 N160 取消的程序中:

- T102 和 M3 已激活。
- T102 已更改。刀具的位置未更改。

之后将重新选择程序，并借助块预运行，在没有进给运动情况下包括 N150。激活经过编程的辅助功能，但不会输出。

在处理 N160 之前，现在使用 AUXFUNC 来执行当前所有现有辅助功能组的当前激活的通道特定和通道综合辅助功能。

在这个示例中，AUXFUNC 会导致:

- 激活刀具编号 T102（刀具仍然更改）。
- 1. 通过顺时针旋转激活主轴 / 主轴组。

## 6.12 更改最大轴加速度 "AxAcc, AAC"。缓存轴加速度 "AxAccSave, AAS"

**描述:**

- **AxAcc:**  
更改最大轴加速的上限。函数将使用带编程值的机床参数来覆盖最大轴加速的值。
- **AxAccSave:**  
将所有轴的当前最大加速值临时保存至内存。  
一旦程序被选定，就以机床参数的值预初始化该内部存储器。

使用高级语言语法的 NC 功能

|     |                                |                                                                                                                                                                                                      |
|-----|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | AxAccSave                      |                                                                                                                                                                                                      |
|     | 缩写形式:<br>AAS                   | 临时保存当前轴加速值。                                                                                                                                                                                          |
|     | AxAcc 或<br>AxAcc(1)            | 重新激活之前使用 "AxAccSave"保存的轴加速。                                                                                                                                                                          |
|     | AxAcc(<values> {,<reference>}) |                                                                                                                                                                                                      |
|     |                                | 更改轴加速的最大值                                                                                                                                                                                            |
|     | AxAcc() 或<br>AxAcc(0)          | 使用机床参数重新激活轴加速。                                                                                                                                                                                       |
|     | 缩写形式: AAC(..)                  |                                                                                                                                                                                                      |
|     | 其中                             |                                                                                                                                                                                                      |
|     | <Values>                       | 轴名称与加速值<br>各轴的数据以逗号分隔。<br>依赖激活的测量单位 (G71/G70), 控制器会将编程数据解释为 "1000 Inch/s²" 或 "m/s²"。                                                                                                                 |
|     | <Reference>                    | 未编程: <ul style="list-style-type: none"><li>RapidAxAcc &gt; 0.000:<br/>进给加速度发生变化。</li><li>RapidAxAcc = 0.000:<br/>进给和快速移动模式的加速度值为绝对值。</li></ul> <b>RAP:</b> 快速进给加速度被改变。<br><b>ALL:</b> 进给和快速进给加速度被改变。 |

表格 6-25: AxAcc (AAC) 语法

|     |                                                      |                                                                             |
|-----|------------------------------------------------------|-----------------------------------------------------------------------------|
| 示例: | 初始情况:                                                |                                                                             |
|     | 在机床参数中, 轴 X 到 Z 预先分配 8m/s² 的进给加速度和 10 m/s² 的快速横向加速度。 |                                                                             |
|     | :                                                    |                                                                             |
|     | N40                                                  | x-轴的最大进给加速度: 1.0 m/s²                                                       |
|     | AAC(X1.0,Z2.1)                                       | z-轴的最大进给加速度: 2.1 m/s²                                                       |
|     |                                                      | y-轴的最大进给加速度 (8.0 m/s²) 保持不变。                                                |
|     |                                                      | 快速横向加速度值在 10 m/s² 时保持不变。                                                    |
|     | :                                                    |                                                                             |
|     | N80 AxAccSave                                        | 临时缓冲当前所有激活的最大加速值。                                                           |
|     | N90 AAC()                                            | 使用机床参数重新激活值:<br>最大进给加速度值 (X, Y, Z): 8 m/s²<br>最大快速进给加速度值 (X, Y, Z): 10 m/s² |
|     | :                                                    |                                                                             |
|     | N150                                                 | 最大 y-轴的进给和快速进给加速度: 5.0 m/s²                                                 |
|     | AAC(Y5,ALL)                                          | x 轴和 z 轴的最大加速保持不变。                                                          |

:

N200 AAC(1)           重新激活 "AxAccSave"保存的值:

                          x-轴的最大进给加速度: 1.0 m/s<sup>2</sup>

                          y-轴的最大进给加速度: 8.0 m/s<sup>2</sup>

                          z-轴的最大进给加速度: 2.1 m/s<sup>2</sup>

                          所有快速横向加速度值在 10.0 m/s<sup>2</sup> 时保持不变。

- 特殊功能和限制:
- 如果没有为快速移动操作设置单独的加速度值 (AX / Dr [i] / Acc / RapidAxAcc = 0) 并且参数 RAP 未编程, 则 AxAcc / AAC 功能会将加速度值更改为绝对值, 以便进给和快速遍历操作。在这种情况下, 行为与 MTX 版本 V11 或更早版本兼容。



如果括号表达式 "(0)" 或 "(1)" 不代表 NC 功能的参数。因此, 括号内的数值不能通过 CPL 变量传输。

## 6.13      加载轴误差校正表 "AxCompTab"

**描述:**       NC 功能允许更换轴误差校正表 (LSEC 和 CCOMP)。

如果有一个用于轴的轴误差修正的基本表, 并且它被标识为 LOADABLE, 则可以使用 "AxCompTab" 函数加载替代的修正表。



有关使用交叉补偿 (CCOMP) 和主轴螺杆误差补偿 (LSEC) 补偿机械不准确性的更多信息, 请参阅文档 "IndraMotion MTX 功能描述"。

**语法:**       **激活可重载表:**

|                               |                            |
|-------------------------------|----------------------------|
| AxCompTab ( <CAX> , <Table> ) |                            |
| 其中                            |                            |
| <CAX>                         | 补偿轴<br>系统轴的名称或编号。          |
| <Table>                       | CCOMP 或 LSEC 表的名称 (包括路径指定) |

表格 6-26:       AxCompTab 语法 - 可重载表

**激活基本表格:**

|                             |                                                                            |
|-----------------------------|----------------------------------------------------------------------------|
| AxCompTab ( <CAX> , <RAX> ) |                                                                            |
| 其中                          |                                                                            |
| <CAX>                       | 补偿轴<br>系统轴的名称或编号。                                                          |
| <RAX>                       | 参考轴<br>系统轴的名称或编号。<br><br>CAX 和 RAX 是相同的: LSEC 表<br>CAX 和 RAX 是不同的: CCOMP 表 |

表格 6-27:       AxComp 选项卡语法 - 基本表

使用高级语言语法的 NC 功能

激活所有基本表格：

|                     |                   |
|---------------------|-------------------|
| 激活 所有 基本表格：         |                   |
| AxCompTab ( <CAX> ) |                   |
| 其中                  |                   |
| <CAX>               | 补偿轴<br>系统轴的名称或编号。 |

表格 6-28: AxComp 选项卡语法 - 所有基本表

**示例：**基本表 “cnc03r02.tab” 仅包含启用可重新载入表的关键字，而不包含校正  
值。  
如果需要，表格 “Reload0302.tab” 只能通过 NC 功能 “AxCompTab” 激活。  
系统轴 2 的名称为 YA，轴 3 的名称为 ZA。

基本 CCOMP 表：cnc03r02.tab

#STUP

可加载  
空

#ESTUP

可装载的 CCOMP 表：Reload0302.tab

#CONF

CAX 3；ZA 是补偿轴

RAX 2；YA 是参考轴

#ECONF

-20000.0 50000.0 ； 起始值 -20 毫米，步长 50 毫米

7.0 7.0；补偿 -20 毫米

9.0 9.0；补偿 30 毫米

9.8 9.8；补偿 80 毫米

9.0 9.0；补偿 130 毫米

7.0 7.0；补偿 180 毫米

4.0 4.0；补偿 230 毫米

0.0 0.0；补偿 280 毫米

：

N010 CCOMP 表 Reload0302.tab 变为活动状态。  
AxCompTab (ZA,Reload0302.  
tab)

：

N200 AxCompTab (ZA,YA) CCOMP 表 cnc03r02.tab 变为活动状态

：

- 特殊功能和限制：
- 单个通道和异步轴的轴表可以交换。
  - 活动表的名称存储在系统数据 “SysAxActData []”的一个组件中。
  - 更换桌子时发生轴运动！

加载表格时，运动限制为最大 2 毫米。如果需要更大的运动，则会输出运行时错误。

返回基本表时，动作不受限制。

- “AxComTab” 函数 不会在块预运行中执行。

## 6.14 轴耦合 "AxCouple、AXC"

**描述：** 轴耦合用于创建异步“主动轴”运动与一个或多个（最多 7 个）异步“从动轴”间的特定关系。

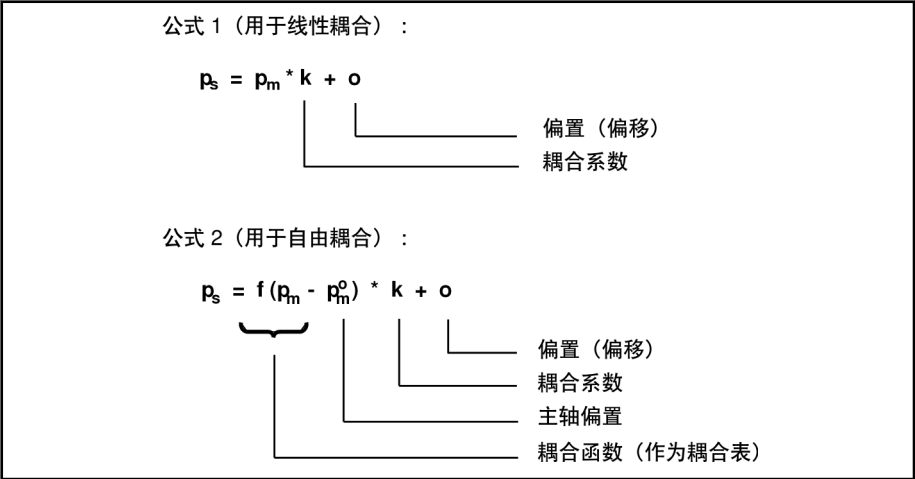
若从动轴在耦合启动时未处于耦合位置，则会以内部生成的线性进给运动方式移动至该位置。激活路径进给和倍率电位计已生效。

若主动轴进行进给，则所有的从动轴将自动根据与主动轴的所定义关系自动移动。主动轴和所有相关从动轴的组合也称为“耦合轴组”。

主轴和从动轴之间的可能关系：

- 主动轴的命令位置通过 **常量偏置** 变换至从动轴的相应命令位置（参见公式 1）。  
此可用于实现从动轴以正向或负向进给方向，相对于主动轴按任何常量路径偏置。
- 主动轴的命令位置通过 **常量耦合系数** 变换至从动轴的相应命令位置（参见公式 1）。  
此可用于相对于主动轴适当地移动从动轴。
- 主动轴的目标位置使用（耦合）表变换至从动轴的 **任何** 相应目标位置。在此将把提供对应主动轴位置的从动轴位置的支点以及主动轴偏置（如果需要保存）保存到表格中（参见公式 2）。

控制器可通过三次样条函数来确认各支点间的位置。





使用高级语言语法的 NC 功能

N100  
AXC (C0,A (ACT),B (ACT,  
2) )

创建 轴组  
C: 主动轴  
A/B: 从动轴  
两个从动轴均以“线性”耦合方式运行。  
A 和 B 耦合至 C, 无进给运动。  
使用耦合系数 2 对 B 进行操作。

N100 AXC (C0,B (,  
0.5,,T\_B))

创建 轴组  
C: 主动轴  
B: 从动轴  
从动轴 B 可以“任何”耦合方式运行。

:

N150 AXC (C1,A (4.5,2) )

扩展一组耦合轴  
A 添加作为从动轴。  
从动轴 A 以“线性”耦合方式运行。  
从动轴 B 仍然可以“任何”耦合方式运行。

:

N200 AXC (C-1,A ( ) )

从 C 轴组移除从动轴 A。

:

N200 AXC (C-1)

整个轴组 c 被删除。

:

另一个示例:  
主动轴 X, 从动轴 Y2  
从轴具有模糊的轴名称 Y2, 与将 y 轴编程为值 2 不同。  
AXC 必须编程如下:  
AXC (X,Y2=(...))

该函数将触发在此块内编程的所有从动轴的进给运动！  
它们会进给至主动轴位置和耦合特性定义的耦合位置（参考值）。

- 特殊功能和限制:
- 耦合轴组中的所有参与轴必须为同步轴，至少在轴进行耦合时为同步轴。不容许将异步轴或 Hirth 轴作为从动轴跟随在一组耦合轴之后。
  - 不允许对从动轴进行进给运动编程，这会导致出现错误消息。
  - 从动轴不可同时在另一组耦合组中作为主动轴使用。
  - 程序在结束时不会自动解散一组耦合的轴。
  - 若主动轴为模轴，则在线性耦合关系下，从动轴也必须为模轴。
  - 若要使轴逐个地进给至参考点，则必须打开耦合轴组。
  - 将从动轴进行耦合可减小允许的主动轴进给范围（例如，从动轴比主动轴更快到达其结束位置，或从动轴的进给范围小于主动轴）。
  - “最弱”轴的最大动力性能决定着整组耦合轴的最大动力性能。
  - 在一组耦合轴活动时禁止锁定轴。
  - 必须在测试模式关闭前解除测试模式中耦合轴的耦合状态。

使用高级语言语法的 NC 功能

## 6.15 轴进给耦合 "AxisFeedCouple, AFC"

**描述:** 根据参考轴或参考主轴的速度来驱动异步轴的进给。例如，该功能允许为异步轴指定每转进给（G95）。该联轴器在定位模式和速度模式下均有效。将结果进给计算如下：

$$\text{Feed}[\text{AsynchronousAxis}] = | \text{Feed}[\text{ReferenceAxis}] | * \text{耦合系数}$$

进给可另外经由轴倍率缩放。

|     |                                                     |                                                                      |
|-----|-----------------------------------------------------|----------------------------------------------------------------------|
| 语法: | AxisFeedCouple, AXISFEEDCOUPLE                      |                                                                      |
|     |                                                     | 激活 / 停用进给耦合                                                          |
|     | 缩写形式: AFC (...)                                     |                                                                      |
|     | AFC (FRA=<ReferenceAxis>,FA=<FeedAxis>,CF=<Factor>) |                                                                      |
|     | 进给耦合参数:                                             |                                                                      |
|     |                                                     | 与系统轴参考进给耦合                                                           |
|     | AFC (FRA=<Axis>,FA=<Axis>,CF=<Factor>)              |                                                                      |
|     |                                                     |                                                                      |
|     |                                                     | 与通道主轴参考点 n = 1..8 进给耦合                                               |
|     | AFC (FRA=Sn,FA=<Axis>,CF=<Factor>)                  |                                                                      |
|     |                                                     |                                                                      |
|     |                                                     | 与系统主轴参考 nn = 00..32 进给耦合                                             |
|     | AFC (FRA=<System spindle>,FA=<Axis>,CF=<Factor>)    |                                                                      |
|     |                                                     |                                                                      |
|     | G95                                                 | 与当前 G95 参考进给耦合                                                       |
|     | AFC (FA=<Axis>,CF=<Factor>)                         |                                                                      |
|     |                                                     |                                                                      |
|     | AFC (FA=<Axis>)                                     | 禁用 <axis> 的馈送耦合                                                      |
|     | AFC ()                                              | 停用所有系统轴的进给耦合                                                         |
|     | 控制器通过编程的进给（S/FA 值）或参数化进给（最大轴速度）继续执行主动轴插补。           |                                                                      |
|     | 其中                                                  |                                                                      |
|     | FRA = <Reference axis>                              | 参考轴可以是系统轴，通道主轴（S1..S8）或系统主轴。<br>如果没有参数，则控制器将存储激活的 G95 参考主轴或 G95 参考轴。 |



|                                 |                                                                                                                                                                                                                                                                                                       |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FA = <Feed axis>                | 异步线性，旋转或无限轴                                                                                                                                                                                                                                                                                           |
| CF=<Factor> 或<br>CF=S(<Factor>) | <p>将参考轴速度缩放到请求的进给速度。旋转轴和主轴的参考轴速度单位为 ° /min，线性轴单位为 mm/min。以下内容适用于组合式旋转/主轴直线轴：</p> <p>1° /分钟对应于 1 毫米/分钟。</p> <p>如果循环参考轴是系统轴联轴器的从动轴，则可以通过 “s (&lt;Factor&gt;)” 属性为每次均匀进给量考虑 “修改后的” 速度。“修改后的” 速度中不包含系统轴联轴器的定位轴或工作台部分。</p> <p>&lt;Factor&gt;必须大于 0。</p> <p>进给耦合系数可以通过 “G94/G95 (DSi =) 增量速度编程” 功能动态改变。</p> |

表格 6-30: 语法 AxisFeedCouple, AFC



如果进给耦合失效，轴可能会发生无意的运动！

停止时连接到参考轴的进给轴可以通过停用联轴器再次移动。

通过在关闭联轴器之前指定 S (0)，完成与系统原点位置的异步运动运动或停止速度模式下的无限轴运动。

示例:

```
; 齿轮整形:
; Z3 行程轴, C1 刀具轴, 进给齿轮 FG [mm/Z3 旋转]
; 一个 Z3 旋转对应于成型工具的双行程 (DS)
 ; 刀具直径 100 毫米
 ; 进给齿轮 0.2 毫米/DS
1 PI = 3.141593: DIA=100 : FW=0.2
 ; 耦合系数= FW/C1 周长

1 FC=FW/
(PI*DIA)

 ; C1 轴的进给耦合
N20 AFC (FRA=Z3, FA=C1, CF=[FC])
N30 C1=S(-1) ; 开始进给运动: C1=S(-(Z3*FC))
...

; 异步线性轴 VA 的每转进给量
N50 G95 F0.5 ; 激活每转通道进给

 ; 异步的每转进给量
 ; VA 轴: 1 mm/rev
N60 VA=10 AFC (FA=VA, CF=[1/360])
...

; 齿轮轴每转 C2 的进给= SAC (B2, X, Z)
```

## 使用高级语言语法的 NC 功能

```

; VA 进给 1 毫米/ C2 转,
; B2 部分但没有 X 和 Z 部分
N80 VA=0 AFC (FRA=C2,FA=VA,CF=S ([1/360]))
...

```

;与路径相关的速度差的进给耦合

```

N100 G1
G95 (FRA=B1) ; 1.5 毫米/ B1 转速的 G95
F1.5

```

; B2 与 B1 的速度相反

;方向 (1: 1 耦合)

```

N110 AFC (FA=B2,CF=1.0) B2=S (-1)

```

;提高 B2 速度直到块终止

; (1: 2 耦合)

```

N120 G95 (DSA1=B2,DS=1.0) X10
...

```

;关闭单轴联轴器

;停止 C1 轴和

```

N150 C1=S (0) ;停用进给耦合
AFC (FA=C1)
...

```

;停用所有系统轴的进给耦合

;停用 B2 和 VA 进给耦合,

```

N170 AFC () ;将 B2 速度更改为 1 rpm

```

**特殊功能和限制:**

- 同时编程的异步进给 (FA) 限制最大耦合进给。
- 如果在速度模式下或在定位过程中停止进给耦合, 则编程的 S 值或 FA 值被激活。运动继续在不同的进给。
- 运动后, 进给联轴器保持模态。如果重新编程, 轴将与现有的联轴器一起运行。
- 如果无级轴在速度模式下运行并且进给耦合激活, 则控制器仅评估旋转方向的 S 值的符号。速度由指定的耦合确定。
- 启用或禁用进给联轴器不会启动或停止进给轴, 而是用允许的轴加速度切换飞行进给。
- 参考主轴的实际速度和参考系统轴的指令速度用于耦合。
- 进给联轴器不能确保联轴的位置同步。
- 进给耦合在设置模式下不起作用。
- 通道进给切换 (G93 / G94 / G95) 对主动轴进给耦合没有影响。

## 6.16 更改最大轴过冲 "AxJerk, AJK", 缓存轴过冲 "AxAccSave, AAS"

- 描述：
- AxJerk:**  
更改最大轴过冲的上限。函数将使用带编程值的机床参数来覆盖最大轴过冲的值。
  - AxJerkSave:**  
将所有轴的当前最大加加速度值临时保存至内部存储器。  
一旦程序被选定，就以机床参数的值预初始化该内部存储器。

语法：

|                                  |                                                                                                                                                                                                            |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AxJerkSave<br>缩写形式: AJS          | 临时缓冲当前的最大加加速度值。                                                                                                                                                                                            |
| AxAcc 或<br>AxAcc (1)             | 使用 "AxJerkSave"再次激活先前保存的急动值。                                                                                                                                                                               |
| AxJerk (<Values> {,<Reference>}) |                                                                                                                                                                                                            |
|                                  | 修改最大轴加加速度值                                                                                                                                                                                                 |
| AxJerk () 或<br>AxJerk (0)        | 从机床参数重新激活轴加加速度值。                                                                                                                                                                                           |
| 缩写形式: AJK (..)                   |                                                                                                                                                                                                            |
| 其中                               |                                                                                                                                                                                                            |
| <Values>                         | 轴名称与过冲值<br>各轴的数据以逗号分隔。<br>依赖当前的测量单位 (G71/G70)，控制器会将编程的数据解释为 "1000 inch/s <sup>3</sup> 或 "m/s" <sup>3</sup> "。                                                                                              |
| <Reference>                      | 未编程: <ul style="list-style-type: none"><li>RapidAxJerk &gt; 0.000:<br/>进给加加速度发生变化。</li><li>RapidAxJerk = 0.000:<br/>进给和快速移动模式的加加速度值为绝对值。</li></ul> <b>RAP:</b> 快速进给加加速度被改变。<br><b>ALL:</b> 进给和快速进给加加速度被改变。 |

表格 6-31: AxJerk、AJK 语法

- 示例：
- 初始情况：
- 在机床参数中，轴 X 到 Z 预先分配 40 m/s<sup>3</sup> 的进给加速度和 60 m/s<sup>3</sup> 的快速横向加速度。
- |                   |                                                                                  |
|-------------------|----------------------------------------------------------------------------------|
| :                 |                                                                                  |
| N10 JKC (1)       | 启用过冲限制                                                                           |
| :                 |                                                                                  |
| N40               | x-轴的最大进给加速度: 30.0 m/s <sup>3</sup>                                               |
| AJK (X30.0,Z20.0) | z-轴的最大进给加加速度: 20.0 m/s <sup>3</sup>                                              |
| :                 | y-轴的最大进给加速度 (40 m/s <sup>3</sup> ) 保持不变。<br>快速横向加速度值在 60 m/s <sup>3</sup> 时保持不变。 |

使用高级语言语法的 NC 功能

|                     |                                               |
|---------------------|-----------------------------------------------|
| N80 AxJerkSave      | 临时缓冲所有当前激活的最大加速度值。                            |
| N90 AJK ()          | 使用机床参数重新激活值：                                  |
| :                   | 最大进给加加速度值 (X, Y, Z) : 40.0 m/s <sup>3</sup>   |
|                     | 最大快速进给加加速度值 (X, Y, Z) : 60.0 m/s <sup>3</sup> |
| N150 AJK (Y35, ALL) | 最大 y-轴的进给和快速进给加加速度: 35.0 m/s <sup>3</sup>     |
| :                   |                                               |
| N200 AJK (1)        | 重新激活 "AxJerkSave"保存的值：                        |
|                     | x-轴的最大进给加速度: 30.0 m/s <sup>3</sup>            |
|                     | y-轴的最大进给加加速度: 40.0 m/s <sup>3</sup>           |
|                     | z-轴的最大进给加加速度: 20.0 m/s <sup>3</sup>           |
|                     | 所有快速横向加速度值在 60.0 m/s <sup>3</sup> 时保持不变。      |

- 特殊功能和限制:
- 如果没有为快速移动操作设置单独的加加速度值 (AX / Dr [i] / Acc / RapidAxAcc = 0) 并且参数 RAP 未编程, 则 AxAcc / AAC 功能会将加加速度值更改为绝对值, 以便进给和快速遍历操作。在这种情况下, 行为与 MTX 版本 V11 或更早版本兼容。
  - 仅在启用过冲限制 (JerkControl/JKC) 的情况下, 过冲限制值才起作用。
  - 若活动通道的物理和逻辑轴在系统内名称相同, 则将会始终影响到逻辑轴的过冲。

6.17 关闭主轴的 C 轴模式 "AxisToSpindle、ATS"

描述: 将处于 C 轴模式 (参见 第 6.129 章 "开启主轴的 C 轴模式 SpindleToAxis、STA" 第 339 页) 的主轴切换至主轴模式。



有关 "轴转换" 功能性的详细信息, 请参阅文档 "IndraMotion MTX 功能描述"。

语法:

|                                              |                                |
|----------------------------------------------|--------------------------------|
| AxisToSpindle (<PAN> <PAI>{,<PAN> <PAI>}...) |                                |
| 缩写形式: ATS (...)                              |                                |
| 其中                                           |                                |
| <PAN>                                        | 物理轴名称<br>指定将由 C 轴模式切换回主轴模式的主轴。 |
| <PAI>                                        | 物理轴索引<br>效果与 <PAN> 相同          |

表格 6-32: AxisToSpindle 语法

示例:

:  
N030 ATS (CH) 物理轴 CH (即轴操作时名称为 CH 的主轴) 切换为主轴操作。  
:

- 特殊功能和限制:
- 给定轴必须处于静止状态, 且不可从属于任何轴组。否则, 控制器会生成错误消息, 并终止程序。
  - 相同块中的轴位置必须始终根据 AxisToSpindle (...) 表达式进行编程。

## 6.18 关闭主轴的 C 轴模式并等待 "AxisToSpindleWait, ATSW"

**描述:** 将处于 C 轴模式的主轴切换至主轴模式并等待直至主轴被启用（参见第 6.129 章 “开启主轴的 C 轴模式 SpindleToAxis、STA” 第 339 页 和第 6.130 章 “开启主轴的 C 轴模式并等待 SpindleToAxisWait、STAW” 第 339 页 ）。



有关 “轴转换” 功能性的详细信息，请参阅文档 “IndraMotion MTX 功能描述”。

**语法:**

|                                             |                                |
|---------------------------------------------|--------------------------------|
| AxisToSpindle(<PAN> <PAI>{,<PAN> <PAI>}...) |                                |
| 缩写形式: ATSW(..)                              |                                |
| 其中                                          |                                |
| <PAN>                                       | 物理轴名称<br>指定将由 C 轴模式切换回主轴模式的主轴。 |
| <PAI>                                       | 物理轴索引<br>效果与 <PAN> 相同          |

表格 6-33: AxisToSpindle 语法

**示例:**

:  
N030 ATSW(CH) 物理轴 CH（即轴操作时名称为 CH 的主轴）切换为主轴操作。  
N040 M3 S1000 主轴被启动。  
:

**特殊功能和限制:**

- 给定的轴不应属于任何轴组。  
否则，控制器会生成错误消息，并终止程序。
- 相同块中的轴位置必须始终根据 AxisToSpindleWait(...) 表达式进行编程。

## 6.19 替换轴 "AxisReplace, AXR "

**描述:** 在 “目标调整程序”中预先运行块后，此参数允许用不同于计算的轴配置的组态输入零件程序，方法是使用功能相似的轴互换轴。

**语法:**

|                    |                                                         |
|--------------------|---------------------------------------------------------|
| AXR(<KAN1>,<PAN2>) | 通道轴名称<CHAN1>的轴从通道中移除，物理轴名称<PA2>的轴被添加到通道并接收通道轴名称<CHAN1>。 |
|--------------------|---------------------------------------------------------|

表格 6-34: AxisToSpindle 语法

**示例:**

**AXR(A, A2)**  
通道轴名称为 A 的轴从通道中移除，物理轴名称为 A2 的轴将添加到通道并接收通道轴名称 A.

**特殊功能和限制:**

- 轴只能相互替换，如果它们属于同一类型的话。
- 具有加工功能的轴不能更换。
- 通道轴名称必须可用且已编程为要替换的轴
- 替换轴功能只能在 “目标调整程序”中编程，并且必须同时配置初始化调整程序。
- 在调用 AXR 之前必须调整轴配置。
- 每条 NC 线只能编程一个 AXR 命令。
- 轴更换后，不再允许编程 WAX, RAX, GAX, RLN 或 ALN 命令。

使用高级语言语法的 NC 功能

- 总共可以更换 4 个轴。

## 6. 20 改变最大轴加速度 "AxVel、AVE"， 保存最大轴速度 "AxVelSave、AVS"

描述：

AxVel：

可临时更改轴速度的上限。函数将使用带编程值的机床参数来覆盖最大轴速度的值。

AxVelSave：

将所有轴的当前最大速度值临时保存至内存。一旦程序被选定，就以机床参数的值预初始化该内部存储器。

语法：

|                       |                        |
|-----------------------|------------------------|
| AxVelSave             | 保存当前激活的最大轴速度值。         |
| 缩写形式： AVS             |                        |
| AxVel (<Values>)      | 更改最大轴速率                |
| AxVel () or AxVel (0) |                        |
|                       | 从机床参数重新激活最大轴速度。        |
| AxVel (1)             | 激活 AxVelSave 保存的轴最大速度。 |
| AxVelSave             | 保存当前激活的最大轴速度值。         |
| 缩写形式： AVE (..)        |                        |
| 其中                    |                        |
| <Values>              | 轴的名称与速度值               |

表格 6-35: AxVel、AxVelSave 语法

各轴的数据以逗号分隔。

根据当前激活测量单位（G71/G70），控制器将编程的线性轴数据解释为“英寸/分”或“mm/min”。

回转轴始终以单位“1000 度/分”指定。

 如果括号表达式“(0)”或“(1)”不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

示例：

初始情况：

机床参数中轴 X 到 Z 的默认值为 50000 mm/s。

```
:
N40 X 轴最大速度: 10000.0 mm/s
AVE (X10000,Z30000
0) Z 轴最大速度: 30000.0 mm/min
: Y 轴的最大速度 (50000.0 mm/min) 保持不变。
:
N80 AVS 临时缓冲所有当前激活的最大速度值。
:
```

N90 AVE()  
:  
N200 AVE  
:  
X 轴最大速度: 10000.0 mm/s  
Z 轴最大速度: 30000.0 mm/min  
Y 轴最大速度: 50000.0 mm/min

使用机床参数重新激活值。  
  
重新激活借助 "AVS/AxVelSave" 保存的值:

6.21 B 样条逼近的参数"BsaPar, BAP"

描述: 设置 B 样条逼近参数。

语法:

|                            |                                                           |
|----------------------------|-----------------------------------------------------------|
| BsaPar(E..,OE..)           |                                                           |
| E                          | 公差范围的半径<br>开启值: 参数 SPL / BSplineApproxTol (MP 8007 00030) |
| OE                         | 定向矢量的公差值<br>开启值: OE = E                                   |
| BsaPar(0)<br>或<br>BsaPar() | 将 B 样条逼近重置为接通值                                            |
| 缩写形式: BAP                  |                                                           |

表格 6-36: BAP 语法

示例: 设置 B 样条逼近的参数

SPD(4233, x, y, z, 0) ; 它定义了一个随后用 G6 激活的样条类型  
GSP(E0.02, OE0.1) ; 它设置 B 样条近似的参数:  
; 公差 0.02 毫米  
; 公差方向 0.1 毫米  
  
G6 x.. y.. z.. phi.. ; B 样条逼近 (样条 4233) 上  
theta..  
...  
BAP () ; 切换回开启值:  
; 来自参数 SPL / BSplineApproxTol 的容差 (MP  
8007 00030)  
...  
G1 x.. y.. z.. phi.. ; B 样条逼近停止  
theta..

6.22 定位: 工件位置修正 "BcsCorr、BCR"

描述: 用作 设置校正。

"工件位置补偿" 的定位可在空间内的任意处对工件坐标系进行进行切换 和 定向。工件位置修正会在对应的通道中, 通过含义 "X"、 "Y" 和 "Z" 来影响坐标。

 轴名 “X”、“Y” 以及 “Z” 将在下面用于指示含义为 X、Y 和 Z 的轴。

由于在跨越后才进行工件位置的测量，并考虑使用工件位置补偿，因此显著简化了设置工作。

由于存在 3 个定向自由度，因此每个定向都必须用 3 个连续的基础旋转来表达。为了便于理解，以下图片仅显示关于 Z 坐标的基本旋转：

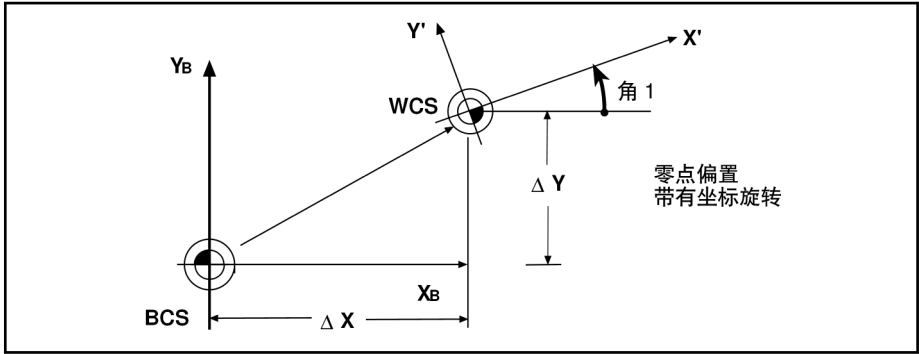


插图 6-4: 绕 Z 坐标的基本旋转

 定位 “斜面”（参见 第 5.2.44 章 “定位：斜面 G151 - G159.5” 第 161 页）在函数方面相同。您可在找到全部三个基本旋转的表示。

附加定位（例如，倾斜平面）具有叠加效果。“计算链”位于斜面前方：

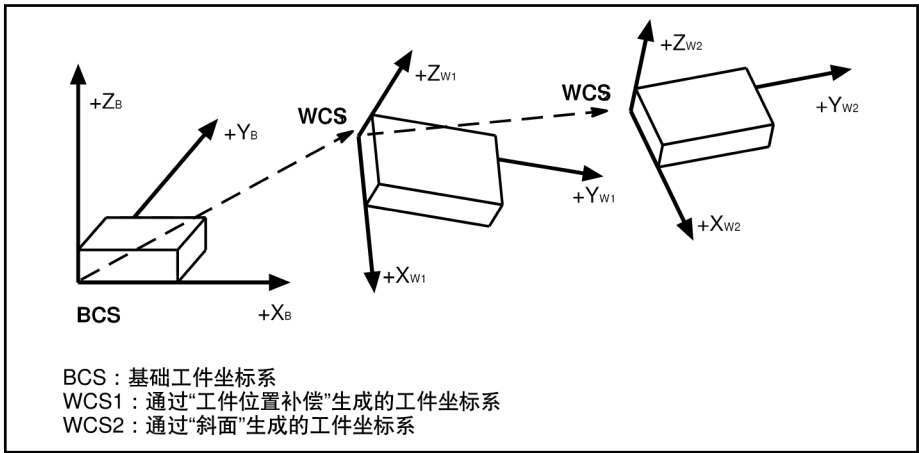


插图 6-5: 计算链

|                         |                                                                                                                                      |          |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------|----------|
| 语法:                     | BcsCorr ( {<X <sub>w</sub> -Offset>} {, {<Y <sub>w</sub> -Offset>} {, {<ZW-Offset>} {, {<Angle1>} {, {<angle2>} {, {<angle3>}}}}}} ) |          |
|                         |                                                                                                                                      | 工件位置修正开启 |
|                         | BcsCorr () 或<br>BcsCorr (0)                                                                                                          | 工件位置修正关闭 |
|                         | 缩写形式: BCR (..)                                                                                                                       |          |
|                         | 其中                                                                                                                                   |          |
| <X <sub>w</sub> offset> | 主坐标 X 方向的偏置值                                                                                                                         |          |
| <Y <sub>w</sub> offset> | 辅助坐标 Y 方向的偏置值                                                                                                                        |          |



|                         |                                              |
|-------------------------|----------------------------------------------|
| <Z <sub>w</sub> offset> | 法向坐标 Z 方向的偏置值                                |
| <Angle1>                | 沿着 z 坐标旋转角度<br>取值范围: 0 ≤ <Angle1> < 360 度    |
| <Angle2>                | 关于新 Y' 坐标的旋转角度<br>数值范围: 0 ≤ <Angle2> < 180 度 |
| <Angle3>                | 沿着新 Z" 坐标旋转角度<br>数值范围 0 ≤ <Angle3> < 360 度   |

表格 6-37: BcsCorr (BCR) 语法

|     |                          |                                                                                                     |
|-----|--------------------------|-----------------------------------------------------------------------------------------------------|
| 示例: | N70 G40                  | 铣刀路径校正关闭                                                                                            |
|     | N80 BCR (50,300,10,1.23) | 工件位置修正开启<br>在 BCS 中, 新工件坐标系统的零点位于 X50 Y300 和 Z10。在 BCS 中, 新工件坐标系的 X 和 Y 坐标轴关于 新的 Z 坐标轴逆时针旋转 1.23 度。 |
|     | :                        |                                                                                                     |
|     | :                        |                                                                                                     |
|     | N200 BCR ()              | 工件位置修正关闭                                                                                            |

- 特殊功能和限制:
- 激活和取消工件位置补偿将中断块预读; 因此, 在刀具路径补偿激活时 (G41/G42, 参见 第 5.2.16 章 “铣刀路径修正 G40, G41, G42” 第 128 页 ), 不得对其进行编程。
  - 在控制器复位的情况下, 在机床参数 7060 00010 和 7060 00020 中配置上电状态和特性。

## 6.23 使用缓存数控块输入进行下一块监控 "ChkNxtBlk、CNB"

描述:

使用缓存数控块输入, ChkNxtBlk 函数能够用于监控数据流。  
将检查缓存的数控块输入是否能够及时提供下一个块。如果不是这种情况, 将输出运行时错误。

 有关 OPC 接口的信息, 请参见 “《Rexroth IndraMotion OPC 通信》”手册。

|     |                              |          |
|-----|------------------------------|----------|
| 语法: | ChkNxtBlk (1)                | 启用下一块监控。 |
|     | ChkNxtBlk (0) 或 ChkNxtBlk () | 停用下一块监控。 |
|     | 缩写形式: CNB (..)               |          |

表格 6-38: Syntax ChkNxtBlk (CNB)

特殊功能和限制:

该功能可通过复位停用。

## 6.24 倒角编程"ChLength"、 "CHL", "CHLL", "ChSection", "CHS", "CHSL"

描述:

“倒角编程” 在两个连续直线或圆周型数控块之间插入一个过渡段, 其长度可以规定为绝对 倒角长度 或作为 倒角段长度。该倒角在激活的工作平面内生成。  
通过模态 (ChLength, CHL, ChSection, CHS) 或逐块 (CHLL, CHSL) 提供 “倒角编程” 功能。在逐块编程时, 可以定义一个本地进给来加工倒角。

可能使用以下的倒角过渡：

• 两条相邻直线间的倒角

该倒角与相邻路径段间的等分线成直角。若与相邻的编程路径段无交叉，会自动修正（减小）倒角的长度。

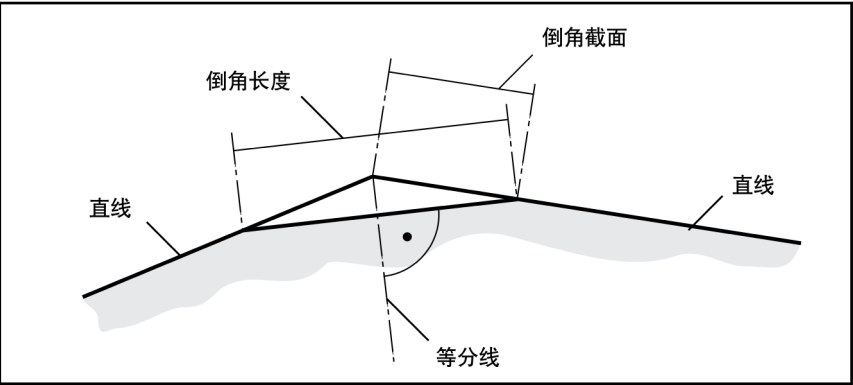


插图 6-6: 两条相邻直线间的倒角

• 两段相邻圆周段间的倒角

若轮廓的过渡包含圆周部分，则斜面的尺寸将与轮廓过渡中所涉及路径段的起始或终止切线相关。所涉及圆周半径等其他因素很大程度上决定了实际结果斜切长度，从而或多或少偏离编程的尺寸。

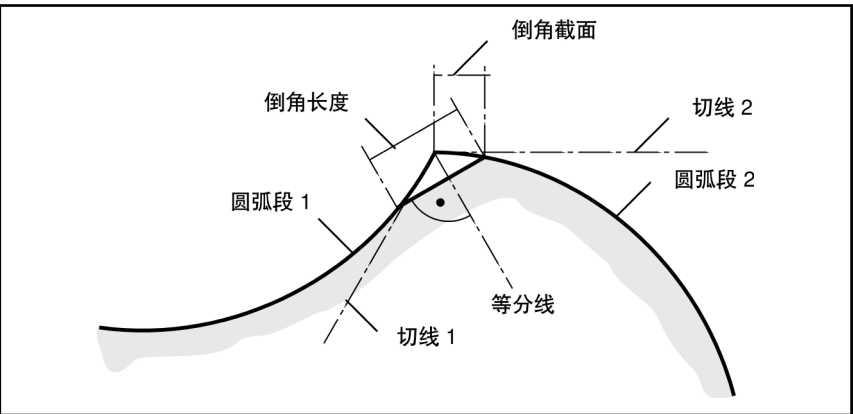


插图 6-7: 两个相邻圆周部分间的倒角

语法:

|                              |                                                          |
|------------------------------|----------------------------------------------------------|
| ChLength (<ChamferLength>)   | 倒角编程打开<br>以单位 mm (G71) 或英寸 (G70) 指定所需的 <Chamfer length>。 |
| ChLength () 或 ChLength (0)   | 倒角编程关闭                                                   |
| 缩写形式: CHL (...)              |                                                          |
| ChSection (<ChamferSegment>) |                                                          |
|                              | 倒角编程打开<br>以单位 mm (G71) 或英寸 (G70) 指定所需的 <ChamferSegment>。 |

|                               |        |
|-------------------------------|--------|
| ChSection() 或<br>ChSection(0) | 倒角编程关闭 |
| 缩 写 形 式 :<br>CHS(...)         |        |

表格 6-39: ChLength (CHL) 和 ChSection (CHS) 语法

|                                    |                                                |
|------------------------------------|------------------------------------------------|
| CHLL(<Length> {,<br><LocFeed> } )  | 用倒角长度指定逐块倒角编程                                  |
| CHSL(<Section><br>{, <LocFeed> } ) | 用倒角段指定逐块倒角编程                                   |
| 其中                                 |                                                |
| <Length>                           | 以单位 mm (G71) 或英寸 (G70) 指定所需的 <Chamfer length>。 |
| <LocFeed>                          | 用于移动过渡倒角的本地进给。                                 |

表格 6-40: CHLL 语法和 CHSL 语法

- 特殊功能和限制:
- 功能 “ChLength”, “ChSection”, “RoundEps” 和 “Rounding” 以**模态行事** 並相互取消选取。
  - 功能 “CHLL” 和 “CHSL” **在本地执行** （逐个块）。
  - 如果倒角或倒圆在模态上是激活的，并且如果同时编程了局部倒角，则局部功能会逐个模块交叉衰减模态功能。
  - 倒角仅与激活的的工作平面（G17、G18、G19、G20）相关。如果在运动中涉及到附加轴，则倒角不受影响。  
由于加工面内轴的编程运行块的坐标由倒角控制，但加工面外的轴的值保持不变，空间直线的方向可以改变。
  - 此函数仅影响 “顺序块”、“单块” 或 “单步”下“行进”的工作范围。  
由于 “程序块” 以手动输入方式作用，倒角编程在此处无效。
  - 打开 / 关闭步骤以及基于控制器复位的特性由机床参数 7060 00010 和 7060 00020 的初始化字符串决定。
  - 若平面、零点偏置或轴变换在两个连续的块之间切换，则不生成任何倒角。

6. 25 自动探测转角和直线 "CLD"

- 描述:
- 根据编程的参数，“自动探测转角和直线”函数将在样条序列中生成转角和直线。若弯角大于已编程的极限角，则保留编程的弯角，例如，该角度不按样条曲线进行倒角。若点距离大于块中编程的极限距离，则生成线性样条。此外，可以定义角度和距离因素（仅样条类型 4）。若角度或距离超出其预读范围中的平均值，则该函数还会生成一个弯角或一条直线。此外，还可以规定转角公差值。如果一条具有该公差的想象修圆样条可插入块过渡，而且至少还剩两侧中一侧的一半，则距离条件满足该样条。此函数仅在 G6（样条插补）活动时有效。除样条系数编程外，对于所有样条类型均有效。
- 语法:
- 可利用 **CornerLineDetection** 语法启用自动检测转角或线条。

|                                                            |                                      |
|------------------------------------------------------------|--------------------------------------|
| CornerLineDetection(ANG<a>, AFACT<fa>, DIST<d>, DFACT<fd>) |                                      |
| 这些参数具有以下含义:                                                |                                      |
| ANG<a>                                                     | "a" 为轮廓中允许的最大弯角。<br>大于 a 的弯角均满足角度条件。 |

使用高级语言语法的 NC 功能

|           |                                                                    |
|-----------|--------------------------------------------------------------------|
| AFACT<fa> | 仅样条类型 4:<br>"fa" 为角度系数<br>借助系数 "fa" 大于平均弯角的弯角均满足角度条件。              |
| DIST<d>   | "d" 为允许的最大点距。大于 d 的距离均满足距离条件。                                      |
| DFACT<fd> | 仅样条类型 4:<br>"fd" 为距离系数<br>借助系数 "fd" 大于点距的点距均满足距离条件。                |
| E<e>      | "e" 是一个轮廓准确度值<br>如果一条具有该公差的想法修圆样条可插入块过渡，而且至少还剩两侧中一侧的一半，则距离条件满足该样条。 |

表格 6-41: CornerLineDetection 语法

参数可以单独省略。由逻辑 "OR" 链接起 DIST、DFACT 和 E 以及 ANG 和 AFACT 条件。使用至少一个参数对 CLD 进行编程后，只要 G6 处于活动状态，函数即可模态打开并处于活动状态（例外：可通过重新编程随时更改这些参数。

缩写为 CLD(...)。

编程时不使用参数 CornerLineDetection() 即可将其关闭。

示例: c2-常系数三次样条

```
SPD (2203, x, y, z, 0) CLD (ANG20, AFACT5, DIST5)

...

G6 x.. y.. z.. phi.. ;样条 2203 打开；绝对距离条件激活；
theta.. ;绝对和相对角度条件激活

x.. y.. z..0(..)

...

x.. y.. z..0() CLD(DIST8) ; 样条关闭
G1 ; 停用角度条件，
新绝对距离条件

...

G6 ;样条打开，最后一个 CLD(DIST8) 有效
```

6. 26 碰撞 "Collision、CLN"

描述: 允许打开或关闭刀具路径补偿 G41/G42

- 的碰撞，
  - 并根据预读范围和碰撞时的特性进行修改以适应应用。
- 以下情况适用:
- 碰撞监控仅在刀具路径补偿 G41/G42 激活时有效，即使补偿半径的值为 "0"。
  - 若当前半径补偿值不允许对单独的轮廓元素进行加工，则控制器将尝试修改相关路径，以避免损坏轮廓。

使用高级语言语法的 NC 功能

- 在加工轮廓时，碰撞监控仅考虑活动工作平面的坐标。若程序成功避免了碰撞（例如，通过更改刀具进给深度等方式），则仍在当前的预读范围内激活碰撞监控。  
在该情况下，可在受影响的加工部分中临时关闭碰撞监控。

 有关“轴转换”功能性的详细信息，请参阅文档“IndraMotion MTX 功能描述”。

 建议在 MP 7060 00010 / MP 7060 00020 中输入所需的碰撞监控上电特性。

语法:

|                                   |                                                                                                                                                                              |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Collision(1) 或<br>CLN(1) 或<br>CLN | 碰撞监控打开<br><br>在碰撞时保留当前特性。<br><br>若在此时尚未编程行为，或仍未在 MP 7060 00010 / MP 7060 00020 中输入，则该特性将对应 Collision (CollErr 0) 的行为。<br><br>CLN (“不加括号的 CLN”) 对应 CLN(1) 和 / 或 Collision(1)。 |
| Collision() 或<br>Collision(0)     | 碰撞监控关闭                                                                                                                                                                       |
| Collision(DEF)                    | 为两个块指定碰撞监控的预读范围预设。                                                                                                                                                           |
| Collision(CollErr<Art>)           |                                                                                                                                                                              |
|                                   | 发生碰撞时，激活碰撞监控并指定其行为。                                                                                                                                                          |
| Collision(DLA<blocks>)            | 指定碰撞监控预读范围的预设。<br><br>在下次编程了 G41/G42 时生效。                                                                                                                                    |
| Collision(LA<blocks>)             | 临时更改预读范围，直至下次编程了 G41/G42。                                                                                                                                                    |
| 缩写形式: CLN(...)                    |                                                                                                                                                                              |
| 其中                                |                                                                                                                                                                              |
| <Art>                             | 指定检测到碰撞时，控制器的行为：<br><br><b>0:</b><br>不发送任何运行时错误或警告。<br><br><b>1:</b><br>运行时错误输出<br>终止加工。<br><br><b>2:</b><br>输出警告<br>不终止加工。                                                  |
| <Blocks>                          | 指定预读范围的大小（块的数量）。整数值。<br><br>建议的预读范围：1 至 10 个块。                                                                                                                               |

表格 6-42: Collision (CLN) 语法

示例:

:  
  
N100 CLN(DLA     在下个 G41/G42 启动的碰撞监控预读范围的预设：5 个块。  
5)  
  
:

使用高级语言语法的 NC 功能

```
N110 G41 D10 铣削刀具路径修正到工件的左侧
N120 X10 向前进给
N130 X20
N140 X30
N150 G42 工件左侧切削刀具路径补偿打开。
: 碰撞监控的预读在块 150 处终止，之后重启。
N160 X20 向后进给
N170 X10
N180 X0
N190 G40 铣刀路径校正关闭
: 碰撞监控由此不再活动，但并未取消！
```

**特殊功能和限制:**

- 预读范围的预设值为 2 个块。
- 预读范围的最大值由机床参数 7060 00110 至 7060 00130 决定。
- 为了在轮廓激活铣刀轨迹修正时不触发碰撞监控功能而向后移动，必须从换向点开始更换主动修正方向（当 G41 有效时编程 G42，G42 有效时编程 G41）。

控制器由此便能够自动在 G41 或 G42 块中终止碰撞监控函数的预读范围，并在之后立即重启预读函数。

6.27 G0 插补配置 "ConfG0、CG0"

**描述:** G0 插补以异步方式配置。轴保持同步；但是，该动作是异步执行的。即，它们按其各自的动态极限值移动到编程的最终位置。轴在不同的时间到达它们的结束位置。

如果所有轴到达了它们的编程结束位置，该数控块关闭。取决于当前激活的 IPS 和 NIPS，到达结束位置与实际或目标位置有关。

该函数包括 NC 内部 WAIT 功能。

|     |                     |                  |
|-----|---------------------|------------------|
| 语法: | ConfG0(ASY<status>) | 状态可能为 "1" 或 "0"。 |
|     |                     |                  |
|     | ConfG0(ASY1)        | 启用异步 G0 插补       |
|     | ConfG0(ASY0)        | 停用异步 G0 插补       |
|     | 缩写形式 CG0            |                  |

异步插补在所有后续 G0 块中有效，直至编入 CG0(ASY0)，即保持有效直至其被停用。

```
示例:
: 启用异步 G0 插补
N10 CG0(ASY1) G0 异步逼近位置 (100,100,50)，等到最弱的轴到达其位置
N20 XA100 YA100 ZA50
N30 XA200 YA300 ZA55 异步逼近位置 (200,300,55)，等到最弱的轴到达其位置
N40 CG0(ASY0)
N50 XA400 YA450 ZA60 停用异步 G0 插补
: 同步移动到位置 (400,450,60)
```

- 运动始终以特定轴的方式进行。通过在接触点 1 或 2 处的激活的轴转换，在内部计算并以异步方式进给至对应坐标的轴结束位置，即坐标中没有运动。
- 如果系统轴耦合启用，则该函数会造成运行时错误
- 该函数与回程不兼容。

6.28 选择轴转换 "Coord、CRD"

描述： 激活或取消机床参数组 1030 中配置的轴变换。



例如在涉及空间坐标编程时，该函数是必需的（参见“《Rexroth IndraMotion MTX 函数说明》”手册），或者在校准轴运动时为可能需要用到（参见第 6.10.4 章“从 NC ATGET 读取参数”第 200 页、第 6.10.5 章“向 NC ATPUT 写入参数”第 201 页和第 6.10.3 章“前向转换 ATFWD”第 199 页）。

一个轴变换最多可在两个坐标变换点上生效：

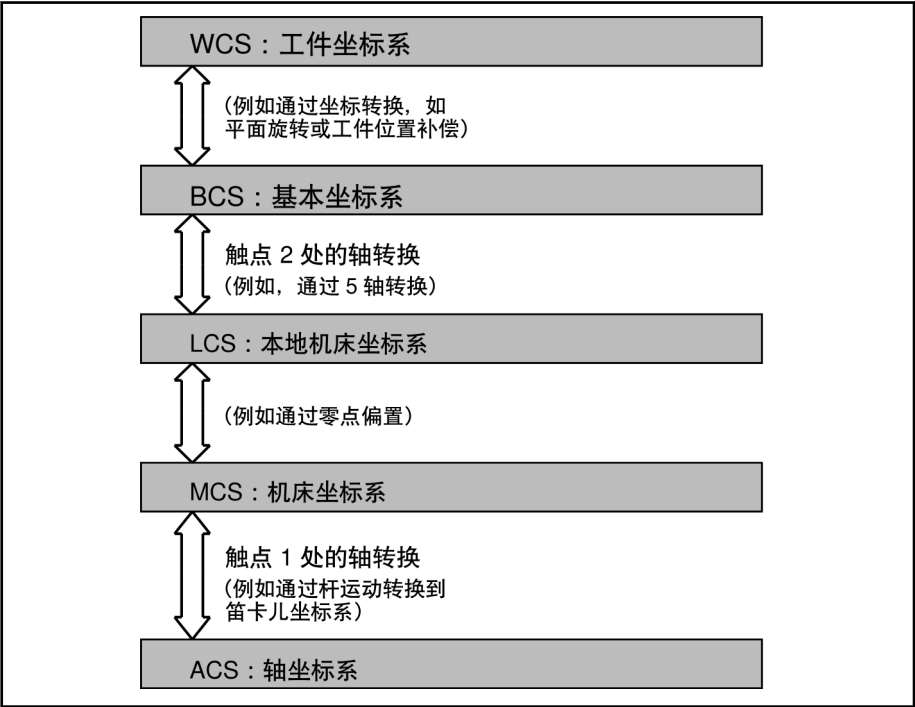


插图 6-8: 轴变换



对于每个可用的轴变换来说，其变换生效的坐标变换点已在内部指定好。

语法：

|                                               |                 |
|-----------------------------------------------|-----------------|
| Coord(<AxTrafoNo>) 或<br>Coord (<[SDTrfName]>) | 轴变换打开           |
| Coord() 或<br>Coord(0)                         | 坐标变换点 2 处的轴变换关闭 |
| Coord(0,<point of<br>action>)                 | 选定坐标变换点的轴变换关闭   |
| 缩写形式: CRD(...)                                |                 |
| 其中                                            |                 |

使用高级语言语法的 NC 功能

|                    |                                                                 |
|--------------------|-----------------------------------------------------------------|
| <AxTrafoNo>        | 轴变换数<br>输入范围：1..20 整数<br>AxTrafoNo 的赋值和轴变换的类型储存于 MP 1030 00110。 |
| <[SDTrfName]>      | SysAxTRafo_t 类型的系统数据名称（括号内）包含了要激活的变换数据。轴变换类型直接包含在系统数据中。         |
| <Points of action> | 需关闭的轴变换的坐标变换动作点<br>输入：1 或 2                                     |

表格 6-43: Coord (CRD) 语法

- 特殊功能和限制:
- 可以在坐标变换点 2 上直接切换不同的轴变换。无需取消之前的轴变换。先前的停用是不必要的。
  - 只有在坐标变换点 2 处无激活的轴变换时，才可在坐标变换点 1 处激活轴变换。



有关可用的轴变换的信息，请参阅文档“IndraMotion MTX 功能描述”。

## 6. 29 锐化内部拐角 "拐角，COR"

- 描述:
- 功能“内角的锐化”可以在使用圆锥工具进行铣削时锐化内角。
- 为了去毛刺和锐化具有三角形横截面的陆地地形轮廓，在激活的刀具半径校正“G41”或“G42”期间，使用具有对应孔径角度的圆锥形刀具并且用陆地的两个边缘进行加工。结果：在土地的尖端的锋利的工具切割边缘。但是，可以在内角找到残余材料（由刀具尖端的边缘半径引起）。因此，在该区域不能加工锋利的刀具刃口。
- 为了加工残余材料，功能“转角，COR”通过考虑拐角前后的轮廓，将工具中心点缩回到拐角的顶端并返回到计算出的轮廓上。抬起刀具尖端，使刀具在轮廓高度前后与轮廓啮合。在角点处，（假定的）刀尖直接位于刀尖上。
- 拐角前后的轮廓线可以由多个任意线性和 / 或圆形 NC 块组成。但是，角落前后相关块的数量在功能“碰撞监视 CLN”的选定预测范围内是很重要的。此外，碰撞监测必须是激活的。同样适用于“轮廓转换 G43 的圆弧”功能。

语法:

|                                                 |                                                                                                                                                        |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cornering( [CAN<can>] [, MAN<man>] [, FF<ff>] ) |                                                                                                                                                        |
| 其中                                              |                                                                                                                                                        |
| <can>                                           | <b>锥角度:</b> 刀具边缘的打开角度，以度为单位。<br>默认值: 90°                                                                                                               |
| <man>                                           | <b>最小角度:</b> 块转换时相邻块的最小角度锐化一条边。<br>默认值: 10°<br><br>该参数仅与“单角”相应。“单角”是由两个相邻轮廓元素单独定义的角。<br><br>在由碰撞监测检测到碰撞的情况下，参数化的最小角度不相关，因为碰撞不一定包含几何角落。如果检测到碰撞，则始终进行锐化。 |



|               |                                                                |
|---------------|----------------------------------------------------------------|
| <ff>          | <b>进给系数</b> ：在削角的情况下，编程进给的进给系数被缩放。<br>默认值：1<br>取值范围：0 < <ff> 1 |
| Cornering(0 ) | 关闭内角的锐化。                                                       |

表格 6-44: 语法转弯

示例:

```
:
N10 G43 在轮廓转换时激活圆弧
N20 CLN(DLA80) 激活碰撞监控；监视范围设置为 80 个块
N30 CON(CAN87.8, 激活并参数化转角锐化：开启角度 87.8°，最小角度
MAN15, FF0.2) 15°，进给系数 20%
N40 G42 ED1 激活刀具半径修正并应用修正块 ED1 中的刀具数据
:
```

- 特殊功能和限制:
- 锐角的功能是模态的。
  - 转角锐化的先决条件：激活“碰撞检测 CLN”。
  - 尖角锐化仅适用于“轮廓转换 G43 处的活动圆弧”功能。
  - 在拐角锐化过程中，刀具的修正半径不能改变。

## 6.30 样条耦合表 "CoupleSplineTab、CST"

描述:

在轴或坐标的耦合中，耦合函数以支点对的形式储存于耦合表中。为了计算数据点间从动轴的位置，块准备会生成一个样条表。

在解释耦合语法时会创建样条表。这些表会以文件的形式储存至链接表目录。样条表为通道轴耦合（AXC，SCCT）自动生成。

*CoupleSplineTab(...)* 也可用于按定义的格式创建样条表：

- CST(TAB(<TabName>, 1))  
明确要求新创建样条表。
- CST(TAB(<TabName>, 0))  
无样条表存在或该表比耦合表更旧时，强制创建样条表。
- CST(TAB(<TabName>{, 0|1}), LOAD)  
将样条表载入系统轴耦合的系统表内存。
- CST(TAB(<TabName>{, 0|1}), UNLOAD)  
将样条表从系统轴耦合的系统表内存中清除。

With CoupleSplineTab(...), 即使没有现有耦合轴组的情况下，仍可创建样条表（例如在手动数据输入模式下）。

有关系统轴联轴器中的表格的应用，请参考手册“IndraMotion MTX 功能说明”，章节“系统轴联轴器”。

样条表的名称由当前激活的耦合表名称再加上扩展名“.s”构成；例如，若耦合表的名称为 **curve.fct** 则样条表的名称为 **curve.fct.s**。



将在当前的搜索路径中搜索耦合表。可在机床参数 3080 00001 中设置搜索路径。

默认的链接表目录为 /usr/lnk。也可使用机床参数 3080 00004 自由定义。

使用高级语言语法的 NC 功能

|                      |  |                                                                                                                                                                                                               |  |
|----------------------|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 语法:                  |  | CoupleSplineTab (TAB (<TabName>{,<1 0>}) {,<LOAD UNLOAD> } {,<Location>})                                                                                                                                     |  |
|                      |  | 生成一个样条耦合表                                                                                                                                                                                                     |  |
| 缩写形式: CST (... ) CST |  |                                                                                                                                                                                                               |  |
| 其中                   |  |                                                                                                                                                                                                               |  |
| <TabName>            |  | 在当前搜索路径中搜索的耦合表名称，同时也是为之创建样条表的耦合表。                                                                                                                                                                             |  |
| <1 0>                |  | 可选:<br><b>0:</b><br>仅在样条表不存在，或比耦合表更旧的情况下创建样条表。(默认)<br><b>1:</b><br>重新生成样条表。                                                                                                                                   |  |
| LOAD UNLOAD          |  | 只能用于系统轴耦合:<br><b>LOAD:</b><br>将样条表载入通道综合表内存 SysAxCoupleTab 以便使用系统轴耦合。<br>如果没有位置说明，则使用条目 “<TabName>” 或者 - 如果不可用 - 则为第一条空闲条目。<br><b>UNLOAD:</b><br>从 SysAxCoupleTab 表内存中清除样条表。<br>如果没有位置说明，则搜索并使用条目 <TabName> 。 |  |
| <Position>           |  | 只能用于系统轴耦合:<br>加载或删除系统表内存位置表 SysAxCoupleTab[<Location>]<br>在面向位置的共享 (UNLOAD) 中，指定的表名将被忽略。<br>位置规范仅与 LOAD / UNLOAD 一起有效。                                                                                        |  |

表格 6-45: CoupleSplineTab (CST) 语法

|     |                              |                                                 |
|-----|------------------------------|-------------------------------------------------|
| 示例: | CST (TAB (curve.fct))        | 在需要时创建样条表 /<Link directory>/curve.fct.s 。       |
|     | CST (TAB (curve.fct,1))      | 创建与样条表 /<link directory>/curve.fct.s 不论日期或是否存在。 |
|     | CST (TAB (curve.fct),LOAD)   | 创建样条表，如必要，则将其载入系统表内存 SysAxCoupleTab (系统数据) 。    |
|     | CST (TAB (curve.fct),UNLOAD) | 从系统表内存中清除该表。                                    |
|     | CST (TAB (curve.fct),LOAD,5) | 将表格 curve.fct 加载到系统表格内存位置 SysAxCoupleTab[5]     |
|     | CST (,UNLOAD,PLACE%)         | 加载或删除系统表内存位置表 SysAxCoupleTab[PLACE%]            |

6. 31

曲线参数插补 "CurveParamInterpolation、CPI"

**描述:** 对于带曲线参数的轮廓元素（即样条类型 0-4 以及修圆样条（语法）SC0），可选择曲线参数插补，而非路径长度插补。通道坐标不同步于距离 s，但与曲线参数 w 同步，w 在 w= 0 与 w = W<sub>e</sub> 之间时处于活动状态。对于样条，W<sub>e</sub>

为样条参数长度。根据所选定的样条参数情况，通过 PL 编程，或自动内部进行计算。

|     |                            |            |
|-----|----------------------------|------------|
| 语法: | CurveParamInterpolation(1) |            |
|     |                            | 开启曲线参数插补   |
|     | CurveParamInterpolation(0) |            |
|     |                            | 返回至路径长度插补。 |
|     | 缩写形式: CPI                  |            |

表格 6-46: CoupleSplineTab (CST) 语法

|          |                     |                      |
|----------|---------------------|----------------------|
| 示例:      | N10 SplineDef(2203) |                      |
|          | N20 G1 F123 X0 Y0   | ;进给为 123 毫米/分钟       |
|          | N30 CPI(1)          | ; 开启曲线参数插补           |
|          | N40 G06 X10 Y10 PL5 | ;块持续时间是 5/123 分钟 (*) |
|          | N50 X20 Y15 PL4 F88 | ;块持续时间是 4/88 分钟 (*)  |
|          | N60 G1 CPI(0)       | ; 返回至路径长度插补          |
| 特殊功能和限制: | (*) 该块中的加速和减速过程除外   |                      |
|          | 控制器启动后，路径长度插补启用。    |                      |

6.32 客户特定的速度参数文件 "CVP"

|     |                                                                   |              |
|-----|-------------------------------------------------------------------|--------------|
| 描述: | CVP 禁用内部速度控制 MTX (G8, G9)。用户可使用 Hermite 样条（样条类型 1）指定用户定义速度参数文件序列。 |              |
| 语法: | CVP                                                               | 禁用 MTX 的速度控制 |
|     | 通过编程 G8 或 G9，取消对 CVP 功能的选择。                                       |              |

表格 6-47: CVP 语法

特殊功能和限制: 此功能仅可用于 Hermite 样条。  
G8、G9 的功能和 CVP 构成模态组，并取消对自己的选择。

6.33 启用 D 修正表 "DcTSeI、DCS"

|     |                                                  |                                                                                                            |
|-----|--------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| 描述: | 启用 D 修正表（几何修正表，GEO 表）。D 修正表以 XML 文件储存在控制器的文件系统中。 |                                                                                                            |
| 语法: | DcTSeI({<path>}<FileName>)                       |                                                                                                            |
|     | 缩写形式: DCS(..)                                    |                                                                                                            |
|     | 其中                                               |                                                                                                            |
|     | <Path>                                           | 有关 <FileName> 存储目录的可选路径指定。<br>如果没有规范，则在 “/数据库” 路径中进行搜索。<br>如果找不到 <FileName>，则为了能在其它目录中找到它，控制器将使用搜索路径搜索子例程。 |
|     | <FileName>                                       | D 修正表的文件名，包括文件扩展名。<br>可以直接通过该编号来激活拥有标准名称的表（DC <Number>.dct），例如 DcTSeI(7) 可以激活表 DC7.dct。                    |

表格 6-48: DcTSeI (DCS) 语法

使用高级语言语法的 NC 功能



有关如何创建和编辑 D 校正表的信息，请参阅控制操作说明。

示例:

```
:

N030 DCS (geotab.dct) 首先在目录 "/database" 中搜索 D 修正表
 "geotab.dct"，若该处没有，则搜索子例程路径。将
: 激活首个发现的，名为 "geotab.dct" 的 D 修正表。

N130 DCS (/mnt/ge.dct) 在目录 "/mnt" 中搜索并激活 D 修正表 "ge.dct"。若
: 该处没有，则会出现错误消息。
```

### 6. 34 从 MP 应用轴设置 "DefAxis、DAX"

描述:

按照 1003 00002 跨通道启用默认轴配置。



有关“轴转换”功能性的详细信息，请参阅文档“IndraMotion MTX 功能描述”。

语法:

|           |
|-----------|
| DefAxis   |
| 缩写形式: DAX |

表格 6-49: DefAxis 语法, DAX

特殊功能和限制:

如果其中一个轴尚未启用，则会导致运行时错误。



由于经常有多个通道参与轴转换的情况，我们建议您在关键字“#SysRes”后的 MP 7060 00020 中输入该函数。  
这样即可始终在“系统复位”后重新生成默认的轴配置。

### 6. 35 将通道主轴复位至 MP "DefSpindle、DSP"

描述:

根据 MP 7020 00020，启用默认主轴设置。

语法:

```
DefSpindle
缩写形式: DSP
```

特殊功能和限制:

如果定义了通道专用整数辅助功能 CSP1 至 CSP8，则会通知 PLC 当前的通道主轴分配。

### 6. 36 无减速的块过渡 "DefTangTrans、DTT"

描述:

“无减速的块过渡”函数限制对最大步长变化对宽过渡角的影响。正确的参数设定可定义行为，从而详细考虑轮廓中的主要弯曲，其中较小的轮廓拐点被视为准继续，并且不会减少路径速度。在真实机床上，此可实现编程轮廓的圆滑。

语法:

|                        |
|------------------------|
| DefTangTrans (<angle>) |
| 缩写形式: DTT (<angle>)    |
| 其中                     |

|                   |                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <Angle>           | 若采用标准化定向向量的轴的一部分的更改在轮廓过渡处大于 sin (<angle>), 将会考虑更改相关轮廓过渡的最大步长。若更改小于 sin (), 则忽略最大步长的更改, 轮廓过渡处无任何减速 (前提是设置的块预读范围中可用的制动路径足够使用)。<br>取值范围: 0 至 50 度 |
| DefTangTrans({0}) | 机床参数 7030 00310 中输入的极限角将生效。                                                                                                                    |

表格 6-50: DefTangTrans (DTT) 语法

- 特殊功能和限制:
- 准停定位函数激活时, 各块过渡中的速度减至 v = 0。
  - 复位后, 启动字符串中的设置生效。
  - 使用 "DefTangTrans" 函数时, 请始终考虑系统进给时所用的速度。由于临时忽略机床参数中配置的极限值, 高速度与大过渡角都可能导致伺服发生错误!

## 6.37 直径编程 "DiaProg、DIA", 半径编程 "RadProg、RAD"

**描述:** 平坦轴的坐标信息 (位于车床上; 通常为 X 坐标) 也可解释为直径或半径。这样, 任何可能存在的尺寸信息可直接传输给零件程序, 无需转换。

若激活 "直径编程", 则直径符号将置于工件位置、剩余行程、终点位置和编程值的平坦轴的轴显示之前。

始终以半径值来显示机床位置、实际轴值以及惯性运转。

 可能存在尺寸信息的错误解释! "DIA" 仅会影响所输入/配置的直径坐标。直径编程对圆周插补参数 I、J、K 无影响。

始终确保仅编程合适的尺寸信息。

语法:

|                                        |                                                                  |
|----------------------------------------|------------------------------------------------------------------|
| DIA {{({ <Coord1> {, ..., <Coord8>})}} |                                                                  |
|                                        | 最多 8 个线性轴 / 坐标的直径编程打开。<br>关闭所有未输入坐标的直径编程。                        |
| 其中                                     |                                                                  |
| <Coordi>                               | 最多可将 8 个线性轴 / 坐标 (i = 1...8) 的距离作为直径信息进行评价。                      |
| DIA                                    | 恢复上次的 RAD 状态。<br>控制器启动后, 将使用默认的机床参数值。                            |
| DIA()                                  | 激活默认值。<br>若机床参数中未定义任何直径坐标, 则分类为 X 的轴将成为直径轴。若该轴也不存在, 则控制器报告运行时错误。 |
| RAD                                    | 所有坐标的半径编程打开                                                      |

表格 6-51: DiaProg (DIA) 和 RadProg (RAD) 语法

示例:

|                |                                      |
|----------------|--------------------------------------|
| N10 DIA()      | 激活已配置的直径坐标                           |
| N20 DIA(Y1,W2) | 给定坐标 Y1 和 W1 成为直径坐标, 所有的非编程坐标成为半径坐标。 |
| N30 RAD        | 取消所有坐标的直径编程。                         |
| N40 DIA        | 坐标 Y1 和 W1 成为直径坐标 (上次的 RAD 状态)。      |
| :              |                                      |

使用高级语言语法的 NC 功能

- 特殊功能和限制：
- 这些函数为模态，并且可相互取消选定。
  - 若轴变换中包含直径坐标，NC 将关闭该轴的直径编程。但若打开轴变换，则坐标可转换为直径编程。
  - 以下内容适用于“手动输入数据”以及“处理”操作模式：  
对于圆周中心坐标、刀具长度和零点偏置的尺寸，平坦轴的坐标将始终解释为半径值。
  - 以下内容适用于“手轮手动操作”以及“点动模式”操作模式：  
可使用轴接口信号“diameter incr. step” (qAx\_JogDia) 来切换直径和半径编程。

6. 38 驱动器错误的延迟响应 "DERDelay"

**描述** 为了避免机器损坏，在某些应用中，NC 需要在短时间内控制驱动器并在发生驱动错误时使机器轴停止坐标。

“DERDelay”函数可以激活 / 关闭 “MAIN / Dr [i] / DelayedErrorReaction” (MP 1001 00012) 中标记为“可编程的”驱动器的延迟错误反应。

 有关更多信息，请参阅“IndraMotion MTX 功能描述”中的“非重大驱动器错误的延迟 NC 错误反应”章节。

语法

|                                                 |                                                                                                 |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------|
| DERDelay(1)                                     | DERDelay 函数可以激活 / 关闭 “MAIN / Dr [i] / DelayedErrorReaction” (MP 1001 00012) 中标记为可编程的驱动器的延迟错误反应。 |
| DERDelay()<br>或<br>DERDelay(0)                  | DERDelay 函数可以激活 / 关闭 “MAIN / Dr [i] / DelayedErrorReaction” (MP 1001 00012) 中标记为可编程的驱动器的延迟错误反应。 |
| DERDelay({ON(<DriveList>)}, {OFF(<DriveList>)}) |                                                                                                 |
|                                                 | 激活 / 取消激活已编程驱动器的延迟错误响应。                                                                         |
| 其中                                              |                                                                                                 |
| <Drive list>                                    | 系统驱动器编号或系统轴名称以逗号分隔的列表。                                                                          |

表格 6-52: DERDelay 语法

**示例** N10 DERDelay(ON(X,B)) ; 激活 x 轴和 b 轴  
...  
N200 DERDelay(OFF(X,B)) ; 关闭 x 轴和 b 轴

- 特殊特性和限制
- NC 功能只能用于机床参数 “MAIN / Dr [i] / DelayedErrorReaction” (MP 1001 00012) 中标记为“可编程”的驱动器。
  - 只有主轴被编程为龙门组。
  - 系统重置会禁任由“DERDelay”功能启用的延迟错误反应。

6. 39 样条类型的定义 "SplineDef、SPD"

**描述：** 可以在所有三个坐标系 Acs（轴坐标），Mcs0（机床坐标（无 Z0））和 Bcs（基本坐标）中计算最小值和最大值。极坐标值分别为每个坐标确定。因此，工作平面的尺寸已经确定。

|     |                             |               |
|-----|-----------------------------|---------------|
| 语法: | DimPlane(1) 或<br>DimPlane   | 启用“工作平面的尺寸”计算 |
|     | DimPlane(0) 或<br>DimPlane() | 禁用“工作平面的尺寸”计算 |
|     | 缩写形式: DimP(..)              |               |

表格 6-53: DimPlane 语法

|                                                                                                                                                                                                                                                                             |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 示例:                                                                                                                                                                                                                                                                         | 程序 |
| <pre>: N100 DimPlane(1)      ;Calculation "Dimension of working plane" on : N500 G1 x11 y22        ;programmed traversing motions : N900 DimPlane(1)      ;Calculation "Dimension of working plane" of N1000 G0               ;target block (e.g. for block search) :</pre> |    |

所有极限值都在块 N100 和 N900 之间确定。

- 特殊功能和限制:
- 极端值在内部存储在易失性通道专用系统日期“/SysDimPlane”中。以下结构元素在模式“sddimp.xsd\_”中定义  
一般数据 (SysDimPlane):

| 结构元素    | 描述                 |
|---------|--------------------|
| MinAcs  | 通道轴的最小值            |
| MaxAcs  | 通道轴的最大值            |
| MinMcs0 | 通道中机床坐标的最小值 (无 Z0) |
| MaxMcs0 | 通道中机床坐标的最小值 (无 Z0) |
| MinBcs  | 通道中机床坐标的最小值        |
| MaxBcs  | 通道中机床坐标的最大值        |

表格 6-54: 结构元素概述 SysDimPlane (通用数据)

- “/SysDimPlane” 系统日期必须由用户创建。模板可在控制系统中找到 (请参阅“工作平面尺寸”一章中的“Rexroth IndraMotion MTX 功能描述”文档)。
- 在控制启动后将值清零后。
- 在激活之后 (即编程“DimPlane(1)”), 所有的值都被初始化: 最小值=无限/最大值= - 无限。
- 系统日期的计算值正在等待, 直到下一次激活。
- “DimPlane(1)” 和 “DimPlane(0)” 之间的粗略插值在块预运行计算期间处于活动状态。然而, 这会减慢块的预运行, 但 NC 可以在没有移动运动的情况下确定最小值和最大值。

## 6. 40 数字化距离控制 "DistCtrl、DCR"

描述: 该函数可在数字化过程中保持被扫描曲面和测量设备 (如激光等) 之间的距离不变。这将确保不会超过测量设备的可用工作范围。



有关轴转换功能性的详细信息, 请参阅文档“IndraMotion MTX 功能描述”。

## 使用高级语言语法的 NC 功能

## 语法:

|                             |                                                                              |                                                                                                                     |
|-----------------------------|------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| DistCtrl(1) 或<br>DistCtrl   | 启动距离控制，并使用曲面和测量设备之间的当前距离作为参照值。<br>通过机床参数 7050 007xx 定义的配置数据已激活。              |                                                                                                                     |
| DistCtrl() 或<br>DistCtrl(0) | 关闭轴距控制，存储当前校正值并停止轴运动。<br>如果在同一个块中作为移动运动进行编程，则在运动完成之前不会禁用轴距距离控制。              |                                                                                                                     |
| DistCtrl(<fct>)             | 在机床参数中禁止一些特定函数的配置数据。                                                         |                                                                                                                     |
| 缩写形式：DCR(...)               |                                                                              |                                                                                                                     |
| 其中                          |                                                                              |                                                                                                                     |
| <Fct>                       | DcAxis(<Axis>,<Corr>)<br>倍率 MP 7050 00702<br>缩写形式：DCA(...)                   |                                                                                                                     |
|                             | <Axis>                                                                       | 要启动轴距控制的通道轴的名称或编号。                                                                                                  |
|                             | <Corr>                                                                       | 计算中应包含校正值的运动方向：<br>+1 或 1：<br>在运动的正方向<br>-1：<br>在运动的负方向                                                             |
| <Fct>                       | DcFilter(<Time>)<br>倍率 MP 7050 00730.<br>缩写形式：DCF(...)                       |                                                                                                                     |
|                             | <Time>                                                                       | 过滤参数化以平滑传感器值。<br>0：<br>过滤器关闭<br>>0：<br>过滤，以毫秒为单位平滑时间                                                                |
| <Fct>                       | DcLimit({<Vel>},{<Acc>})<br>倍率 MP 7050 00740/MP 7050 00741。<br>缩写形式：DCL(...) |                                                                                                                     |
|                             | <Vel>                                                                        | 修正值的最大变化速度。<br>倍率 MP 7050 00740<br>依照测量（G71、G70）的有效单位要求，在输入值时以 mm/min 或英寸/分钟为单位。                                    |
|                             | <Acc>                                                                        | 修正值的最大陡度（加速度）。<br>倍率 MP 7050 00741<br>按照激活的测量单位（G71、G70）要求，输入值使用 m/s <sup>2</sup> 或 1000 英 inch/s <sup>2</sup> 为单位。 |



|           |                                                                                    |                                                                                   |
|-----------|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| <Fct>     | DcMon({<collision>},{<hole>})<br>倍率 MP 7050 00750/MP 7050 00751。<br>缩写形式: DCM(...) |                                                                                   |
|           | <Collision>                                                                        | 容差带用于碰撞检测<br>倍率 MP 7050 00750<br>按照激活的测量单位（G71、G70）要求，输入值使用毫米或英寸为单位。<br>0: 碰撞检测关闭 |
|           | <Hole>                                                                             | 公差带用于检测孔<br>倍率 MP 7050 00751<br>按照激活的测量单位（G71、G70）要求，输入值使用毫米或英寸为单位。<br>0: 孔检测关闭   |
| DcBreak   | 中断轴距离控制。<br>当前修正值保持激活。                                                             |                                                                                   |
| 缩写形式: DCB |                                                                                    |                                                                                   |
| DcCont    | 使用 DCB 中断后恢复轴距离控制。<br>NC 尽可能快地控制与参考值的偏差。                                           |                                                                                   |
| 缩写形式: DCC |                                                                                    |                                                                                   |

表格 6-55: DistCtrl (DCR) 语法



如果括号表达式 "(0)" 或 "(1)" 不代表 NC 功能的参数。因此, 括号内的数值不能通过 CPL 变量传输。

### 6. 41 触发和退出诊断记录 "DREC\_START", "DREC\_STOP"

**描述:** 使用 NC 命令 "DREC\_START" 和 "DREC\_STOP" 时, 能够通过零件程序触发和 / 或退出诊断记录 (轴示波器)。  
若需要在启动记录时加上延迟, 则可在同一数控块中编程 G4 <Delay> 实现。

|     |            |                     |
|-----|------------|---------------------|
| 语法: | DREC_START | 触发来自零件程序诊断记录的启动触发器。 |
|     | DREC_STOP  | 退出诊断记录。             |

表格 6-56: DREC\_START、DREC\_STOP 语法

**示例:**  
N50 DREC\_START G4 F5 ; 以 5 秒的延迟时间触发诊断记录。  
...  
N100 DREC\_STOP ; 退出诊断记录。

### 6. 42 结束位置耦合 "EndPosCouple、EPC"

**描述:** 按照下述关系, 耦合通道中两个当前同步的坐标:  
 $\Delta$  从坐标 = 耦合系数 \*  $\Delta$  主坐标。  
要计算结果从坐标, 主坐标的位置值将总是自动转换为增量路径。  
控制器会使用上面的比率来计算每个数控块所需的从坐标终点位置, 其编程了主坐标的位置值。

使用高级语言语法的 NC 功能

对于其他数控函数，从坐标表现为：如同使用了局部相对尺寸编程（IC），专门为之编写出零件程序中的进给路径。这意味着，例如，当前的进给会影响结果路径运动。

|     |                                               |                |
|-----|-----------------------------------------------|----------------|
| 语法: | EndPosCouple (<M-coord>, <S-coord>, <Factor>) |                |
|     |                                               | 终点位置耦合开启       |
|     | EndPosCouple () 或<br>EndPosCouple (0)         | 终点位置耦合关闭       |
|     | 缩写形式: EPC (...)                               |                |
|     | 其中                                            |                |
|     | <M-coord>                                     | 主坐标名称          |
|     | <S coord>                                     | 从坐标名称          |
|     | <Factor>                                      | 从行驶距离与主行驶距离之比。 |

表格 6-57: EndPosCouple (EPC) 语法

|     |                    |                          |
|-----|--------------------|--------------------------|
| 示例: | N10 G18 G0 Z0      | 激活的平面: z, x<br>快速进给至 z=0 |
|     | N20 G1 F1000 Z3 X1 | 定位至 P1                   |
|     | N30 EPC (Z,X,1)    | 终点位置耦合开启                 |
|     | :                  | 主坐标: Z;                  |
|     | :                  | 从坐标: X                   |
|     |                    | 从行驶距离与主行驶距离相等。           |
|     | N40 Z4             | 定位至 P2                   |
|     | N50 X1             | 定位至 P3                   |
|     | N60 Z5             | 定位至 P4                   |
|     | N70 X3             | 定位至 P5                   |
|     | N80 Z0             | 定位至 P6                   |
|     | N90 EPC ()         | 终点位置耦合关闭                 |
|     | :                  |                          |

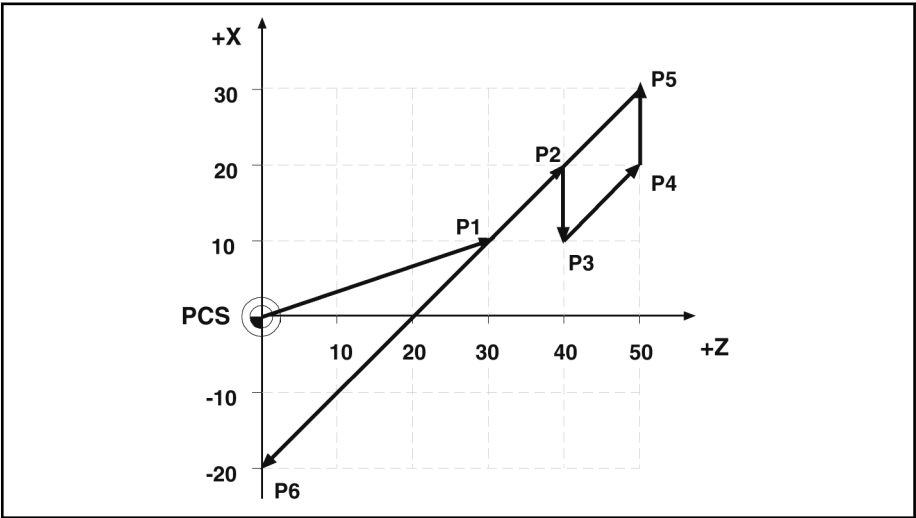


插图 6-9: EndPosCouple (EPC) 的示例

- 特殊功能和限制:
- 终点位置耦合不可结合无终点坐标（这些坐标在其值超过其值域时，会自动转换为定义范围内的值）使用。
  - 激活终点位置耦合后不允许进行极坐标编程。
  - 若终点位置耦合处于活动状态，则不可在同一数控块中对主坐标和轴坐标进行编程。但允许在数控块中仅编程从坐标。
  - 在终点位置耦合活动时，主坐标或从坐标均不可退出通道。不允许进行轴交换以及更改激活的轴变换。
  - 即使打开主坐标的直径编程（参见 第 6.37 章 “直径编程 DiaProg、DIA，半径编程 RadProg、RAD” 第 231 页 ），仍将使用对应的半径值进行耦合。

6.43 进给计算：隐藏轴 "FeedAd、FAD"

描述:

从进给计算隐藏坐标。  
从进给计算中删除所有编程的坐标或 MP 1003 00020 中定义的轴  
被删除的轴在之后会同步移动。  
因此，与编程的 F-值相比，这样可以增加真实路径速度。

语法:

|                         |                                                            |
|-------------------------|------------------------------------------------------------|
| FeedAd(1) 或<br>FeedAd   | 从进给计算中删除机床参数 1003 00020 中定义的轴。                             |
| FeedAd(<Coordinates>)   | 从进给计算中删除编程的坐标。                                             |
| FeedAd(0) 或<br>FeedAd() | 考虑到进给计算的所有坐标。                                              |
| 其中<br><Coordinates>     | 包含坐标名称的列表 方向坐标<br>例如: X, Y, Z, ..., x, y, z, phi, theta, 0 |
| 缩写形式: FAD(..)           |                                                            |

表格 6-58: FeedAd (FAD) 语法

示例:

(在 MP 1003 00020 中输入的 Y 轴)

```
:
N100 G94 G0 X0 Y0 以快速进给定位至 P(0;0)
N110 FeedAd()
N120 X100 Y100 F100 定位至 P(100;100)
: 编程路径速度: 100 mm/min
: 真实路径速度: 100 mm/min
N150 FeedAd 隐藏 Y (通过 MP)
N160 X200 Y200 F100 定位至 P(200;200)
: 编程路径速度: 100 mm/min
: 真实路径速度: 141.42 mm/min
```

使用高级语言语法的 NC 功能

|                |                                                               |
|----------------|---------------------------------------------------------------|
| N190 FeedAd(X) | 用户可读的目标目录定义名称. 本字符串能以 \$ 开头, 此时可在字符串表中查找本地字符串. 在字符串表中定义这些本地设置 |
| X300 Y300 F100 | 定位至 P(300;300)                                                |
| :              | 编程路径速度: 100 mm/min                                            |
| :              | 真实路径速度: 141.42 mm/min                                         |

**特殊功能和限制:** 如果在一个块中仅移动隐藏的用于进给计算的坐标, 则可以通过 “Omega” 地址将它们的进给设置为 “F” 地址。  
(请参阅 第 5.4.5 章 “Omega 地址 (进给率) Omega” 第 176 页 ) 打开。

 如果括号表达式 “(0)” 或 “(1)” 不代表 NC 功能的参数。因此, 括号内的数值不能通过 CPL 变量传输。

## 6.44 前馈"FeedForward、FFW"

**描述:** 该函数可通过适当修正驱动器中的命令插补器, 降低系统相关的惯性运转。因而实现增加的轮廓精度或 “无速度错误运行”。

 根据制造商技术规格, 在驱动器中集成 “前馈”, 且该函数仅由零件程序显示的命令语法激活 / 取消。  
有关 “前馈 ” 函数的详细说明, 请参阅驱动器文档。

 必须使用机床参数 1003 00009 为对应轴释放激活前馈的可能性。

|     |                                   |                                                  |
|-----|-----------------------------------|--------------------------------------------------|
| 语法: | FeedForward(1) 或<br>FeedForward   | 根据 MP 1003 00009, 激活所有已释放轴的前馈 (相关的驱动器切换为辅助模式 1)。 |
|     | FeedForward(<Addr><value>,...)    | 激活 / 取消编程轴的前馈 (效果取决于 <value>)。                   |
|     | FeedForward() 或<br>FeedForward(0) | 取消所有轴的前馈。<br>(所有相关驱动器切换至其主模式。)                   |
|     | 缩写形式: FFW(...)                    |                                                  |
|     | 其中                                |                                                  |
|     | <Addr>                            | 物理或逻辑轴地址                                         |
|     | <Value>                           | <addr.> 的路径信息<br>0:<br>取消前馈<br>不等于 0:<br>激活前馈    |

表格 6-59:      FeedForward (FFW) 语法

|     |             |                                                             |
|-----|-------------|-------------------------------------------------------------|
| 示例: | N10 FFW     | 激活通过参数 AX/Dr/FeedFwd/AxFeedFwd (MP 1003 00009)释放的通道中所有轴的前馈。 |
|     | :           |                                                             |
|     | N50 FFW(Z0) | 禁用 z 的前馈                                                    |
|     | :           | (开关 Z 到主操作模式)                                               |
|     | N90 FFW()   | 禁用通道所有轴的前馈                                                  |
|     | :           | (开关到主操作模式)                                                  |

- 特殊功能和限制:
- 取消前馈后，通道中所有轴均切换至其主模式。
  - 仅可通过写入相关驱动器参数的方式，在驱动器中设置前馈函数的参数。  
可使用“WriteId3”，通过零件程序作为函数来写入 Sercos 参数（参见第 6.153 章“使用扩展 Sercos ID 编号写入 Sercos 参数 WriteId3、WID3”第 367 页）。



如果括号表达式“(0)”或“(1)”不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

## 6. 45 前馈异步轴 "FFWAxis, FFWA"

描述:

该函数可通过适当修正驱动器中的命令插补器，降低系统相关的惯性运转。从而实现增加的轮廓精度或“无速度错误运行”。



根据制造商技术规格，在驱动器中集成“前馈”，且该函数仅由零件程序显示的命令语法激活 / 取消。  
有关“前馈”函数的详细说明，请参阅驱动器文档。



必须使用机床参数 AxFeedFwd (MP 1003 00009)为对应轴释放激活前馈的可能性。

|     |                                             |                                |
|-----|---------------------------------------------|--------------------------------|
| 语法: | FFWAxis({ON(<AxisList>)}, OFF(<AxisList>})) |                                |
|     | 缩写形式: FFWA(..)                              |                                |
|     |                                             | 启用 / 禁用单个通道和异步轴（包括异步 C 轴）轴的前馈。 |
|     | 其中                                          |                                |
|     | <AxisList>                                  | 系统轴名称与系统轴名称以逗号分隔的列表。           |

表格 6-60: Syntax FeedforwardAxis (FFWA)  
或者可选

|                        |                                           |
|------------------------|-------------------------------------------|
| FFWAxis(1)             | 启用/停用单个通道和异步轴的前馈（即 MP 1003 00009 启用的所有轴）。 |
|                        |                                           |
| FFWAxis() 或 FFWAxis(0) | 启用/停用单个通道和异步轴（包括异步 C 轴）轴的前馈。              |

表格 6-61: Syntax FeedforwardAxis (FFWA)

使用高级语言语法的 NC 功能

示例:

N10 FFWA (ON (B, AS10, AS12))

为通道轴 B 和异步轴 AS10, AS12 激活前馈

:

N20 FFWA (OFF (AS10, AS12))

取消激活异步轴 AS10, AS12 的前馈

:

N80 FFWA (1)

激活所有通道轴和所有异步轴的前馈

:

N90 FFWA (0)

取消所有通道轴和所有异步轴的前馈

:

- 特殊功能和限制:
- 龙门从轴只受主轴编程的影响。
  - 主轴 / C 轴切换仅影响轴模式。  
切换 "主轴 - > C 轴"后, 轴处于状态位置界面 (无前馈)。
  - 仅可通过写入相关驱动器参数的方式, 在驱动器中设置前馈函数的参数。
  - 前馈只能在静止状态下切换。  
这意味着 FFWA 块开始时的零件程序必须等待所有编程的异步轴停止。

6. 46 进给计算: 轴组"FeedGrp、FDG"

描述:

确定进给计算坐标。

编程的路径进给是指编程的坐标。

所有未编程的坐标都会同步移动。

因此, 与编程的 F-值相比, 这样可以增加真实路径速度。

语法:

|                                |                                                                                           |
|--------------------------------|-------------------------------------------------------------------------------------------|
| FeedGrp (1)<br>或<br>FeedGrp    | 激活默认的进给组。<br>路径进给是指具有 X, Y, Z 加工功能的坐标。                                                    |
| FeedGrp (<Coordinates>)        | 激活按名称提到的进给组。<br>路径进给是指编程的坐标。                                                              |
| FeedGrp (0)<br>或<br>FeedGrp () | 没有编程的进给组被激活。                                                                              |
| 其中<br><Coordinates>            | 包含坐标名称的列表 方向坐标<br>例如: X, Y, Z, ..., x, y, z, phi, theta, 0<br>phi, theta 和 0 在它们的效果上是相等的。 |
| 缩写形式: FDG (...)                |                                                                                           |

表格 6-62:      FeedGrp 的语法 (FDG)  
例子

FeedGrp (X, Y, Z)

坐标 X、Y 和 Z 是进给计算。

FeedGrp (x, y, z, 0)

坐标 x, y, z 和方向矢量 0 (并且因此方向坐标 phi 和 theta) 是进给计算。

|     |                                                |
|-----|------------------------------------------------|
| 示例: | :                                              |
|     | N100 G94 G0 X0 C0                              |
|     | N110 FeedGrp(X,Y) 坐标 X 和 Y 是进给计算。              |
|     | N120 X100 C50 F100 X 运动用 F100 执行。C 同步移动。       |
|     | N130 C100 F100 没有 XY 运动。                       |
|     | : C 运动是用 F 值执行的。                               |
|     | N200 FeedGrp(1) 默认的 Feed 组 (这里: X, Y, Z) 变得活跃。 |
|     | N210 X200 C150 F100 X 运动用 F100 执行。C 同步移动。      |
|     | N220 C200 F100 没有 XYZ 运动。                      |
|     | : C 运动是用 F 值执行的。                               |
|     | N300 FeedGrp(C) 坐标 C 是进给计算。                    |
|     | N310 X300 C250 F100 C-运动用 F100 执行。X 同步移动。      |
|     | N320 C300 F100 C-运动用 F100 执行。                  |
|     | :                                              |

- 特殊功能和限制:
- 如果移动的唯一坐标是不属于进给组的坐标, 则编程的进给参考这些坐标。
  - 如果缺省进给组 FeedGrp (1) 处于活动状态, 则在编程扩展级别切换 (G17 (...), G18 (...), G19 (...)) 时, 自动调整进给组, 轴坐标转换 (坐标 (...)) 被切换。
  - 如果已命名的进给组处于活动状态, 则在返回到通道后, 临时从通道中移除的轴将再次添加到进给组。

## 6. 47 进给计算: Feed 组类型"FeedGrp, FDGT"

|                                              |                                                                                                                                                                                                                                                                          |                                              |                                      |                                       |                               |                  |  |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|--------------------------------------|---------------------------------------|-------------------------------|------------------|--|
| 描述:                                          | 如果进给组 FeedGrp / FDG 已激活, 则可以在块切换时操作路径速度曲线。                                                                                                                                                                                                                               |                                              |                                      |                                       |                               |                  |  |
| 语法:                                          | <table><tr><td>FeedGrpType (SMOOTH)<br/>或<br/>FeedGrpType (1)</td><td>MTX 根据为 FeedGrp 成员编程的进给 F 计算修改的路径进给。</td></tr><tr><td>FeedGrp (STRICT) 或<br/>FeedGrpType (0)</td><td>编程的进给 F 直接指进给组的成员。剩余的坐标会同步移动。</td></tr><tr><td colspan="2">缩写形式: FDGT (...)</td></tr></table> | FeedGrpType (SMOOTH)<br>或<br>FeedGrpType (1) | MTX 根据为 FeedGrp 成员编程的进给 F 计算修改的路径进给。 | FeedGrp (STRICT) 或<br>FeedGrpType (0) | 编程的进给 F 直接指进给组的成员。剩余的坐标会同步移动。 | 缩写形式: FDGT (...) |  |
| FeedGrpType (SMOOTH)<br>或<br>FeedGrpType (1) | MTX 根据为 FeedGrp 成员编程的进给 F 计算修改的路径进给。                                                                                                                                                                                                                                     |                                              |                                      |                                       |                               |                  |  |
| FeedGrp (STRICT) 或<br>FeedGrpType (0)        | 编程的进给 F 直接指进给组的成员。剩余的坐标会同步移动。                                                                                                                                                                                                                                            |                                              |                                      |                                       |                               |                  |  |
| 缩写形式: FDGT (...)                             |                                                                                                                                                                                                                                                                          |                                              |                                      |                                       |                               |                  |  |

表格 6-63: FeedGrpType 的语法 (FDGT)

|     |                                                                                                                                                                                |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 示例: | A 进给组被激活。编程的进给指进给组的成员。<br><b>FeedGrpType (修匀)</b><br>内部路径进给被调整, 使得用 FeedGrp 编程的坐标随着编程的进给移动。以下速度导致程序通过:<br><i>程序</i>                                                            |
|     | <pre>N10 G0 X0 V0 N20 FeedGrp(X) FeedGrpType(SMOOTH) N20 G1 F1000 X10 V20 ; Fcont = 2236.068, Fx = 1000, Fv = 2000 N30 X20 V100 ; Fcont = 8062.258, Fx = 1000, Fv = 8000</pre> |

在从块 N20 到块 N30 的转换中, 路径进给 Fcont 从 2236 开始增加到 8062。这导致 Fv 的速度相对平滑的变化。

使用高级语言语法的 NC 功能

FeedGrpType (严格)

以下速度导致程序通过:

程序

```
N10 G0 X0 Y0
N20 FeedGrp(X) FeedGrpType(STRICT)
N20 G1 F1000 X10 Y20 ; Fcont = 1000, Fx = 1000, Fy = 2000
N30 X20 Y100 ; Fcont = 1000, Fx = 1000, Fy = 8000
```

在从块 N20 到块 N30 的转换中, 路径进给并未改变。Y 轴的加速余量限制了块切换时的有效路径速度。

特殊功能和限制: 在控制启动期间, SMOOTH 被设置为默认变量。

6. 48 加工中测量 "FlyMeas, FME"

- 描述: "加工中测量"功能 (FME):
- 补充行程命令
  - 允许在行进时进行测量 - 因此可以进行 "加工中测量"
  - 使用变频器的测头功能
  - 以块到块方式起作用或作为一个局部函数
  - (或行进命令) 使用有效模态组的插补类型 G0、G1、G2...
  - 不会停止下一个块准备如有需要, 需对 WAIT 明确编程

|     |                                         |                 |
|-----|-----------------------------------------|-----------------|
| 语法: | FME ( MpiAxis <i> ) <TravelCommand>     |                 |
|     | 或                                       |                 |
|     | FlyMeas ( MpiAxis <i> ) <TravelCommand> |                 |
|     | „MpiAxis “                              | 始终在指数 <i> 之前编程。 |
|     | <i>                                     | 测量轴的系统轴分度       |
|     | <Travel command>                        | 包含行进的 (轴) 坐标。   |

表格 6-64: Syntax FME (FlyMeas)

示例:

```
:
N100 IME(MpiAxis 1); 测量轴的初始化
:
N110 G0 X0 Y0; 行进至 X0, Y0
N120 FME(MpiAxis 1) G1 X20 Y20; 在 X20, Y20 上行进时 FME
:
----- 或 -----
:
N100 IME(MpiAxis 1)
:
N110 G1 X0 Y0
N120 FME(MpiAxis 1) X20 Y20; G1 以模态形式生效。
:
```



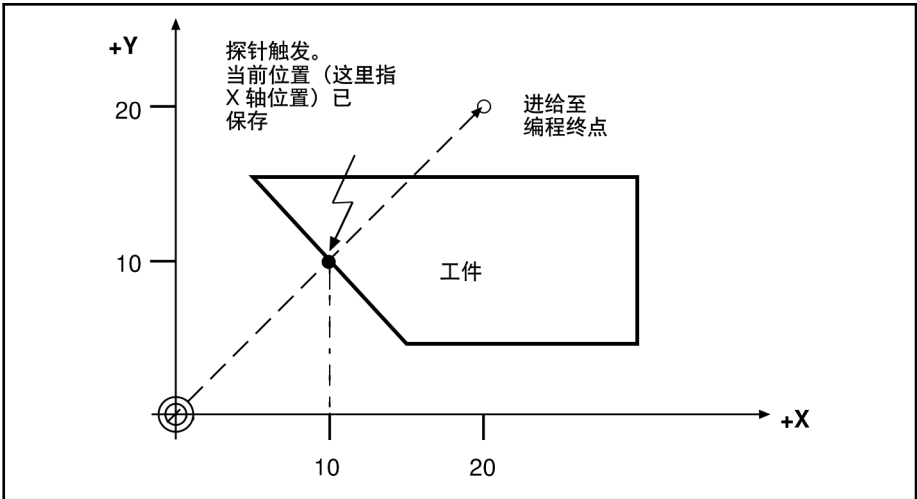


插图 6-10: FME (FlyMeas) 示例

- 特殊功能和限制:
- 调试:

通过 IME 编程或初始化 FME 函数之前,必须先对“加工中测量”(FME)进行操作。所有要求的步骤参见“《Rexroth IndraMotion MTX 功能描述》”手册。
  - 通过 FME 首次对一个测量轴编程之前,应先使用 IME 对其进行初始化。
  - FME 块之后,用户应创建适当的 CPL 程序,将测得的值传递进零件程序;使用 PPOS() 或 PROBE()。
  - 测量值在零件程序中并非立即生效。这不会在 NNC0 / LookAh / Ch[k] /NofBlkPrep 路径中查找块的总数之前执行。

可通过 BlkNmb (BNB) 函数限制规定的块数。如有需要,需对 WAIT 明确编程。
  - 如果已编程的坐标带正号,FME 可用于线性循环轴。使用探头进行反向进给操作时(负坐标的编程)将不会提供唯一值。

6. 49 固定挡块处的测量 "FsProbe、FSP"

描述: 当控制器使用线性插补和提供给已编程终止点的进给来进给所有已编程的同步轴时,将为所选轴监控当前扭矩。

若轴的扭矩在运动时超过可配置的极限值,控制器将执行以下操作:

- 设置轴的 IF 信号“达到固定挡块”
- 存储实际位置
- 使用许可的最大减速度,将路径的运动减速至 v=0
- 清除剩余行程
- 删除“FsProbe”(基于块到块的“FsProbe”)。



为了便于分析,“FsProbe”仅可与 CPL 程序结合使用。

语法:

|                                                           |                                       |
|-----------------------------------------------------------|---------------------------------------|
| FsProbe (MfsAxis<i>) <Coordinates> <Feed>                 |                                       |
|                                                           | 测量轴 <i> 的“固定挡块测量”打开<br>使用线性插补进给编程的坐标。 |
| 或                                                         |                                       |
| FsProbe (MfsAxis (<k>{<threshold>})) <Coordinates> <Feed> |                                       |

使用高级语言语法的 NC 功能

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
|                 | 测量轴 <k> 的“固定挡块测量” 打开<br>若 <threshold> 已编程，则在测量轴 <k> 上使用该值。<br>使用线性插补进给编程的坐标。  |
| 缩写形式： FSP (..)  |                                                                               |
| 其中              |                                                                               |
| <i><br>或<br><k> | 测量轴的系统轴编号<br><br>测量轴的系统轴名或编号                                                  |
| <Threshold>     | 扭矩阈值<br>输入值：以最大扭矩的百分比表示<br>若 <threshold> 未编程，则机床参数 1003 00031（固定挡块处扭矩限制）激活。   |
| <Coordinates>   | 测量时要接近的位置。                                                                    |
| <Feed>          | 所需的路径进给<br>受 MP 1005 00030（“移动至固定挡块处” 的最大进给）和 MP 1005 00002（最大轴速度和快速进给速度）的限制。 |

表格 6-65: FsProbe (FSP) 语法

|     |                           |                            |
|-----|---------------------------|----------------------------|
| 示例： | N100 FSP (MfsAxis (1,30)) | 对第一个物理轴激活 “固定挡块测量”，并使      |
|     | X100 F500                 | 用 F500 接近 X100 的位置。将扭矩阈值设置 |
|     |                           | 为最大扭矩的 30%。                |
|     | 110 IF SD (9) =0 THEN     | 查询是否超过扭矩的阈值。               |
|     | 120 XPOS=PROBE (1)        | 将第一轴（x 轴）切换扭矩中的位置保存至       |
|     | :                         | XPOS 变量。                   |
|     | N130 (MSG, CONTACT)       |                            |
|     | 140 GOTO N180             |                            |
|     | 150 ENDIF                 |                            |
|     | N160 (MSG, NO CONTACT)    |                            |
|     | N170 M0                   | 程序停止                       |
|     | N180 ...                  |                            |

- 特殊功能和限制：
- 在 “FsProbe” 块中不允许使用下列函数：
    - G75（移动至探测头），
    - MOB（在块上测量）
    - InitMeas/FlyMeas（加工中实时测量），
    - RedTorque（扭矩减小）
    - FsMove/FsReset/FsTorque（移动到固定挡块处）。
  - 必须结合至少一个坐标对 FsProbe 编程。其值表示最大搜索深度，在达到该最大搜索深度之前必须达到 “固定挡块”。
  - FsProbe 会暗中暂停后继块的块准备。因此，不一定必须对 WAIT 进行编程。
  - 必须使用 CPL 程序来确定是否达到 “固定挡块”、是否继续程序（出错后）、是否进行安全监控、是否生成错误消息等情况。

- 可使用 SD(9) 来查询是否到达固定挡块处。
- 可以通过 PCSPROBE 或 PROBE 请求固定停止位置。

## 6. 50      运动至固定挡块 "FsMove、FSM",    "FsTorque、FST",               "FsReset、FSR"

描述:

 也可将“运动至固定挡块”函数用于异步轴！

整个“运动至固定挡块”函数包含以下子函数:

1.    **“减少固定挡块的扭矩”: FsTorque, FST**  
      设置开启“移动至固定挡块”后驱动器可使用的最大扭矩。  
      编程值仅对第一个跟随“移动到固定停止”有效。  
      若未使用“FsTorque”，则 MP 1003 00031 生效。
2.    **“移动至固定挡块”: FsMove, FSM**  
      开始固定挡块方向的运动，同时考虑允许的最大扭矩。  
      若在运动中达到允许的最大扭矩（参见第 1 点），控制器将触发以下顺序操作：
  - 发送轴 IF 信号“到达固定挡块”。
  - 使用许可的最大减速，将路径的运动速度减速至 v=0。
  - 将命令位置设置为：  
          实际位置 + 0.1 mm（或实际位置 + 0.1 度）
  - 监控以下内容的轴位置：  
          固定挡块的位置 + 参数 MPS/Dr/MaxDevPoStop (MP 1003 00032)  
          （单位为 mm 或度的固定挡块监控窗口）
  - 停止受影响驱动器上的指定扭矩。
  - 继续处理零件程序。      如果固定挡块尚未达到编程的路径终点（达到指定的扭矩阈值；参见“FsTorque”），控制器会生成错误消息。  
      “FsMove”在“FsMove”块以外仍有效，并且仅可由“FsReset”关闭。
3.    **“复位固定挡块”: FsReset, FSR**  
      释放轴，并在必要时将其从固定挡块移开。
  - 若已在“FsReset”块中编程了同步和/或异步轴，控制器会将所有的轴以指定的进给进给至已编程的终点。各轴容许的最大扭矩应用于从停止处的运动。
  - 若“FsReset”块中未编程任何轴，则仅释放同步轴。在此情况下，只有使用接口信号“复位固定挡块”方可释放“移动至固定挡块”仍然处于活动状态的异步轴。

语法:

|                                              |                                         |
|----------------------------------------------|-----------------------------------------|
| FsTorque(<Addr><Tor>)                        | 激活 <Addr> 轴的最大容许扭矩<Tor>。<br>允许用于同步和异步轴。 |
| 缩写形式: FST ( .. )                             |                                         |
| FsMove..                                     | 激活移动至固定挡块                               |
| <Coord-Syn> <Feed><br><Coord asy> <Feed asy> |                                         |
|                                              | 将同步和异步轴移动至编程的终点位置。                      |

使用高级语言语法的 NC 功能

|                                                           |                                                                                                                                                            |
|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 缩写形式: FSM ..                                              |                                                                                                                                                            |
| FsReset..<br><Coord-Syn> <Feed><br><Coord asy> <Feed asy> |                                                                                                                                                            |
|                                                           | 取消移动至固定挡块<br>将同步和异步轴移动至编程的终点位置。                                                                                                                            |
| 缩写形式: FSR..                                               |                                                                                                                                                            |
| 其中                                                        |                                                                                                                                                            |
| <Addr>                                                    | 物理或逻辑轴地址；可用于异步轴。                                                                                                                                           |
| <Mom>                                                     | 相关轴静止扭矩的最大扭矩（以百分比表示）。<br>取值范围：0 至 500 %。                                                                                                                   |
| <Coord-Syn>                                               | 所需的同步轴终点位置坐标（如 "X100 Y100 Z100"）。各轴进给不考虑 <Feed> 和参数 MPS / Dr / MaxVelAccStop (MP 1010 00030)（最大加速度 "移至固定挡块"）。                                              |
| <Feed>                                                    | 期望的同步轴进给<br>使用 F-地址进行编程，受参数 MPS/Dr/MaxVelPoStop (MP 1005 00030)（"移动至固定挡块"的最大进给）和参数 AX/Dr/Vel/MaxVel (MP 1005 00002)（最大轴和快速速度）的限制。<br>Feed 对每个涉及的同步轴都有个别影响。 |
| <Coord asy>                                               | 请求的异步轴坐标系中的终点位置。考虑到 <FeedAsy>，每个坐标都单独接近和参数 MPS / Dr / MaxVelAccStop (MP 1010 00030)（最大加速度 "移至固定挡块"）。                                                       |
| <Feed asy>                                                | 所需的异步轴进给。<br>使用 "FA"地址进行编程，受参数 MPS/Dr/MaxVelPoStop (MP 1005 00030)（"移动至固定挡块"的最大进给）和参数 AX/Dr/Vel/MaxVel (MP 1005 00002)（最大轴和快速速度）的限制。<br>进给对每个涉及的同步轴都有个别影响。 |

表格 6-66:       FsMove (FSM)、FsTorque (FST) 和 FsReset (FSR) 语法  
转移到固定停止的示例程序

|                             |                                                                |
|-----------------------------|----------------------------------------------------------------|
| N100 FST(X20,Y20)           | ; Restrict torque to 20% for X and Y                           |
| N110 FSM F200 X100 Y=IC(20) | ; Activate moving to fixed stop for X and Y                    |
| ;                           | X travels along the axis position 100, Y travels incrementally |
| ;                           | Both axes each travel with the feed value 200                  |
| :                           |                                                                |
| :                           |                                                                |
| :                           | ; Further part program processing                              |
| N500 FSR                    | ; Deactivate move to fixed stop and                            |
| ;                           | release all synchronous axes                                   |

- 特殊功能和限制:
- 在 "移动至固定挡块" 激活时，在 "FsReset" 块不可使用以下函数：
    - G75（前往探测头）
    - MOB（在块上测量）
    - InitMeas/FlyMeas（加工中实时测量）
    - 通过 PLC 接口减小扭矩
    - FsProbe（固定挡块测量）。
  - 必须结合至少一个终点坐标对 FsMove 编程。其值表示最大搜索深度，在达到该最大搜索深度之前必须达到 "固定挡块"。

- 

用 FsMove 和 FsReset 编程的轴位置始终被解释为绝对位置值，与 G90 / G91 无关！如果要为轴指定增量运行距离，请编程编程属性 IC。
- 

程序轴位置是轴坐标系（ACS）中的位置，而不是工件坐标系（WCS）中的位置！
- 

所有与“行进到固定挡块”有关的程序都是异步执行的。因此，不管 OVD / OVE 如何，轴的速度总是随当前的倍率位置而变化！

## 6. 51 直线度和角度误差补偿"GeoComp, GCT"

**描述:** 激活直线与角度错误补偿，补偿因机械系统误差生成的并且无法由编码器记录的几何错误。

- 

有关“直线度和角度误差补偿”的使用和参数化的全面信息，请参阅文档“IndraMotion MTX 功能描述”。

语法:

|                |              |
|----------------|--------------|
| GeoComp(...)   |              |
| 缩写形式: GCT( ..) |              |
| GCT (1)        | 开启直线度和角度误差补偿 |
| GCT (0)        | 关闭直线度和角度误差补偿 |

表格 6-67: 语法 GCT(1)

- 特殊功能和限制:
- 用户将补偿文件保存为 XML 模式。
  - SCS 模板（文件）存储在控制器的根目录下。文件名可自由选择，文件扩展名必须为“gct”。

- 

如果括号表达式“(0)”或“(1)”不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

## 6. 52 获取轴 "GetAxis、GAX"

**描述:** 将异步轴传递至缩放通道。这会将异步轴变为同步轴。  
然后，可以使用轴的物理或逻辑轴名称在当前通道中对轴进行编程。

- 

有关“轴转换”功能性的详细信息，请参阅文档“IndraMotion MTX 功能描述”。

语法:

|                                                           |                        |
|-----------------------------------------------------------|------------------------|
| GetAxis (<PAN> <PAI>, {<LAN>}{, <PAN> <PAI>, {<LAN>}}...) |                        |
| 缩写形式: GAX (...)                                           |                        |
| 其中                                                        |                        |
| <PAN>                                                     | 物理轴名称<br>指定要变换到当前通道的轴。 |

使用高级语言语法的 NC 功能

|       |                                                                                                                                                                                                            |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <PAI> | 物理轴索引<br>效果与 <PAN> 相同                                                                                                                                                                                      |
| <LAN> | 逻辑轴名称<br>如果对此编程，则要集成到当前通道的轴将被赋予逻辑名称 <LAN>。<br><LAN> 必须在参数 CHAN / Ch / Coord / Acs / ChAx / ChAxName (MP 7010 00010) (通道轴名称) 或参数 CHAN / Ch / Coord / Acs / OptAxName / OptChAxName (MP 7010 00020) (可选轴名称)。 |

表格 6-68:        GetAxis (GAX) 语法

示例:

```
:
N030 接收通道中整合了物理轴 YP 和 ZP。
GAX (YP,ZP,Z) 当 YP 在调用通道中的地址也为 YP 时, ZP 将收到调用通道中的
: 的地址 Z。
```

- 特殊功能和限制:
- 要集成的轴必须处于静止状态。  
否则, 控制器会生成错误消息, 并终止程序。
    - 与函数 "WaitAxis" 相反 (请参阅 [第 6.152 章 "获取轴, 在必要时等待 WaitAxis、WAX" 第 367 页](#)) -
  - 必须始终在 GetAxis(...) 之后, 且仅可在轴变换不处于活动状态时对相同块中轴的位置进行编程。
  - 要转换的轴不得与参与任何激活的监控区域 (参见 [第 6.7 章 "区域监控 区域, ARA" 第 189 页](#))。

6. 53        获取通道主轴 "GetSpindle、GSP"

描述:

将系统主轴传递至调用通道。如此会将系统主轴转变为通道的主轴。其后即可在当前通道中, 使用其逻辑名称和逻辑辅助函数对主轴编程。

语法:

|                                                                           |          |
|---------------------------------------------------------------------------|----------|
| GetSpindle (<SysSpNo>   <SysSpName> , <ChanSpNo>  <ChanSpName> , { ... }) |          |
| 将系统主轴作为逻辑主轴 (通道主轴) 转换至通道中。                                                |          |
| 其中                                                                        |          |
| <SysSpNo>                                                                 | 1..32    |
| <SysSpName>                                                               | 系统主轴名称   |
| <ChanSpNo>                                                                | 1..8     |
| <ChanSpName>                                                              | S1 .. S8 |
| 缩写形式: GSP                                                                 |          |

表格 6-69:        GetSpindle (GSP) 语法

示例:

```
...
N50 GSP (7,2) 应用第七系统主轴, 作为第二通道主轴
N60 M203 S2=500 启动第二通道主轴
...
N110 GSP (SSP12,S2) 应用第十二系统主轴, 作为第二通道主轴
```

- 特殊功能和限制:
- 您可在多个通道中以逻辑的形式表示系统主轴。
  - 若已存在带指定编号的通道主轴, GetSpindle 不会导致运行时错误。

- 例外情况：
- 要替换的主轴正以常量切削速度运行（G96）。
- 如果定义了通道专用整数辅助功能 CSP1 至 CSP8，则会通知 PLC 当前的通道主轴分配（例如，系统主轴 12 被分配给 CSP2 12，即 ChannelSpindle2）。

## 6. 54 通过高速信号实现加工中块过渡 "HsBlkSwitch、HSB"

**描述：**允许使用控制器通道接口的高速输入或用户输入提前实现块过渡。这样就能够根据外部事件，提前终止 **线性** 行进运动。



对已编程路径的修改！

若已编程的“HsBlkSwitch”块终点和其后的块位于不同的直线上，外部事件总会导致无法预测的路径修改！



可用机床参数来配置高速信号。

以下内容适用于加工中块过渡：

- 在没有剩余行程的情况下，于“自动块”、“单块”以及“单步”操作模式中进行删除，
- 在有剩余行程的情况下，于“程序块”和“手动数据输入”操作模式中进行删除，
- 通常不需** 轴静止  
块以当前速度退出（有关例外，请参见“特性和限制”），
- 不检查轴跳转能力的最大可能能力
- 其效果视当前的操作模式而定（自动/单个/程序块、单步、手动数据输入）。参见“特性和限制”。

**语法：**

|                                                    |                                                     |
|----------------------------------------------------|-----------------------------------------------------|
| HsBlkSwitch(HS<x>=<y>) 或<br>HsBlkSwitch(CI<x>=<y>) |                                                     |
|                                                    | 加工中块过渡                                              |
| 缩写形式：HSB(..)                                       |                                                     |
| 其中                                                 |                                                     |
| <x>                                                | 高速信号的编号，或客户所输入的编号（qCh_Custom<x>）。<br>取值范围：1...8，整数。 |
| <y>                                                | 块过渡所需的逻辑信号状态。<br>0：低<br>1：高                         |

表格 6-70: HsBlkSwitch (HSB) 语法

**示例：**

**事件相关进给**

在直线路径上，将根据外部事件减少进给。

对此，需要输入了不同进给信息的三个数控块。若在块过渡处未制动至 v=0，则需要加工中块过渡。

请记住，由于任务的特性（...“在直线路径上”...），所有已编程的终点必须位于同一直线上。

使用高级语言语法的 NC 功能

 之后的块的终点不可与前面的块的终点相同。否则将不会在后面的块中进行进给运动！

 以下块中已编程的行进距离会在块过渡处影响最大可能进给。若进给距离过短，则可能自动降低路径的速度！

：

|                                    |                                           |
|------------------------------------|-------------------------------------------|
| N20 G0 X0 Y0                       | 接近初始位置                                    |
| N30 HSB(HS1=1) G1 X100<br>Y10 F500 | 使用 F500 进给，直至获得高速信号 1 = “高” 或 X100 Y10。   |
| N40 HSB(HS1=0) X110 Y11<br>F100    | 使用 F100 进行进给，直至获得高速信号 1 = “低”，或 X110 Y11。 |
| N50 X120 Y12 F500                  | 使用 F500 执行剩余的行程，直至 X120 Y12               |

：

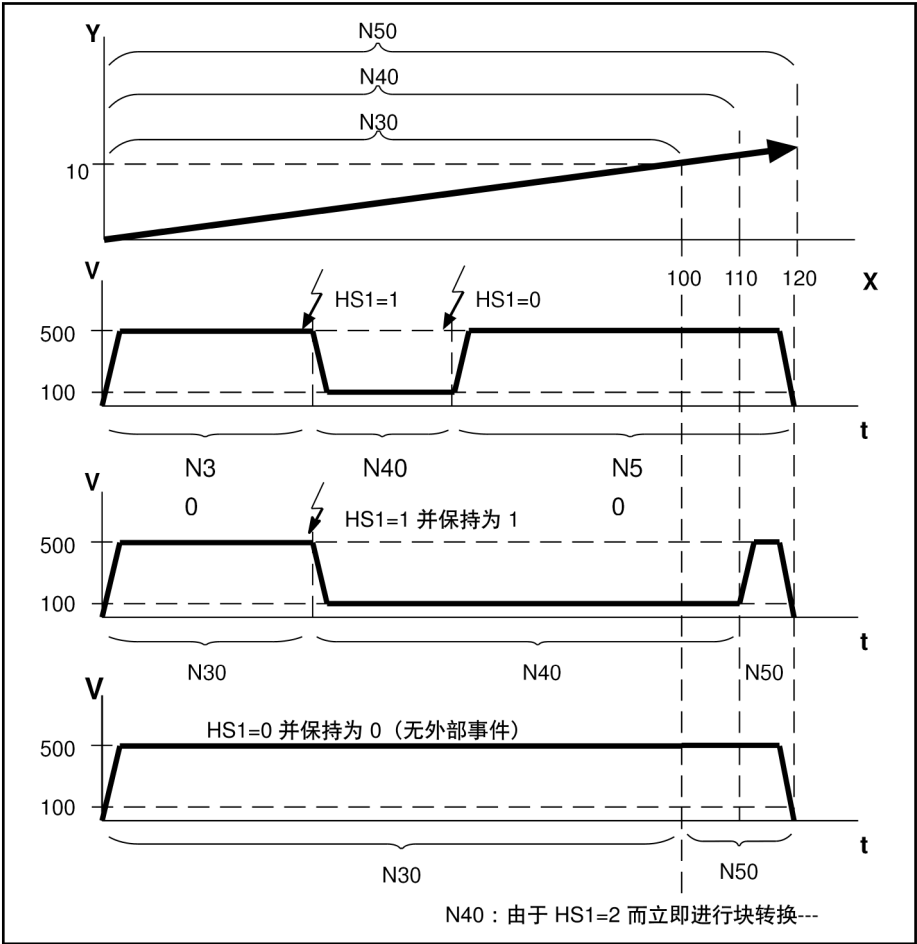


插图 6-11:      HsBlkSwitch (HSB) 示例

特殊功能和限制：

- 此函数需要在 “HsBlkSwitch” 块和以下块中进行线性进给运动。
- 函数必须写入路径信息。它可以与其他路径条件一起写入。
- 在 “继续块”、“单块” 以及 “单步” 操作模式下，未在以下块中编程的轴终点将从提前终止的块中传递。



- 在以下情况下会出现 **带轴静止的“加工中块切换”**：
  - “HsBlkSwitch”块和以下块间的轮廓弯角 > 90°。
  - 准停定位处为活动状态，即 G0(IPS...) 或 G1(IPS...)。
  - 由于附加编程，以下块将以 v=0 启动（例如，若 C0 编程或前馈处于活动状态）。
  - “过冲限制速度导向”（G8(SHAPE...)、G9(SHAPE...) 等）处于活动状态。
  - 操作模式“单块”、“单步”、“程序块”或“手动数据输入”处于活动状态。

6. 55

通过高速信号实现带挡块的块切换  
"HsBlkSwitch(.., HSSTOP=..), HSB(.., HSSTOP=..)"

**描述：**允许使用控制器通道接口的高速输入或用户输入提前实现块切换。这样就能够根据外部事件，提前终止 **线性** 行进运动。



对已编程路径的修改！  
若已编程的“HsBlkSwitch”块终点和以下块位于不同的直线上，外部事件总会导致无法精度预测的路径修改！



可用机床参数来配置高速信号。

以下内容适用于有停止处的块过渡：

- 在所有操作模式中删除剩余距离
- 始终制动至 v=0；  
无论使用速度跳转还是下坡  
即使外部事件不发生，仍在块终点制动至 v=0。

**语法：**

|                                                                            |                                                    |
|----------------------------------------------------------------------------|----------------------------------------------------|
| HsBlkSwitch(HS<x>=<y>, HSSTOP=<z>) 或<br>HsBlkSwitch(CI<x>=<y>, HSSTOP=<z>) |                                                    |
|                                                                            | 有停止处的块过渡                                           |
| 缩写形式：HSB(.., HSSTOP=..)                                                    |                                                    |
| 其中                                                                         |                                                    |
| <x>                                                                        | 高速信号的编号，或客户所输入的编号（qCh_Custom<x>）。<br>取值范围：1...8，整数 |
| <y>                                                                        | 块过渡所需的逻辑信号状态。<br>0：低<br>1：高                        |
| <z>                                                                        | 在事件发生时的制动类型：<br>0：使用最大减速度减速至 v=0<br>-1：速度步长至 V=0。  |

表格 6-71: HsBlkSwitch (HSSTOP) 语法

**示例：**

**事件相关的进给运动停止：**

:

N20 G1 X0 Y0 F1000                      接近初始位置

使用高级语言语法的 NC 功能

|                           |                                            |
|---------------------------|--------------------------------------------|
| N30 HSB (HS1=1,HSSTOP=-1) | 使用 F10 进行 X 轴的进给，直至获得高速信号 1 = “高” 或 X10。   |
| X10 F10                   | 某个事件会触发一个到 v=0 的跳转，并删除剩余的行程。               |
| N40 HSB (HS2=1,HSSTOP=0)  | 使用 F200 进行 X 轴的进给，直至获得高速信号 2 = “高” 或 Y100。 |
| Y100 F200                 | 事件以最大减速度触发减速至 v=0，并删除剩余行程。                 |

- 特殊功能和限制：
- 此函数需要在 “HsBlkSwitch” 块和以下块中进行线性进给运动。
  - 函数必须写入路径信息。它可以与其他路径条件一起写入。
  - 若 “急进限制导向” (G8(SHAPE...), G9(SHAPE...)) 处于活动状态，则在 HSSTOP=-1 时**不会** 发生速度跳转。

6. 56 工件坐标在线修正 "HWOC", "HWOCDIS"

- 描述：
- 在工件坐标系中进行 “在线修正” 时，以下内容与修正值 “重叠”：

  - 通道工件坐标系中的当前位置或方向，或是
  - 刀具长度轴在 TSC Z 方向的当前位置

控制器通过所连接的手轮（常规情况下）、驱动器或 CPL 变量值获取修正范围。

有关轴转换功能性的详细信息，请参阅文档 “IndraMotion MTX 功能描述”。

语法：

|                                                        |                                                                                    |
|--------------------------------------------------------|------------------------------------------------------------------------------------|
| HWOC({CHAN<channel no>},CRDNO <Coord no>,{STEP<incr>}) |                                                                                    |
|                                                        | 在线修正打开                                                                             |
| HWOCDIS{(CHAN<channel no>)}                            |                                                                                    |
|                                                        | 在线修正关闭。<br>当前修正的范围保留在内存内。                                                          |
| HWOC()<br>或<br>HWOC(0)                                 | 在线修正关闭<br>清除相关的修正值。                                                                |
| 其中                                                     |                                                                                    |
| <ChannelNo>                                            | 激活 / 取消在线修正的通道的编号。<br>若未编程 CHAN <channel no>，则当前通道的编号将生效。                          |
| <Coordno>                                              | 在线修正生效的坐标。<br>数值范围：1 至 9 以及 103<br>整数<br>1 至 8：<br>给定通道的坐标编号<br>9 或 103：<br>TCS 坐标 |
| <Incr>                                                 | 每 1 英寸增量所需增量步长的规范。<br>仅在 MP 7050 00926 = 0 时有效。                                    |

表格 6-72: HWOC 和 HWOCDIS 的语法

- 特殊功能和限制:
- 没有针对超出软件限位开关的可能行进，分析由在线修正生成的坐标位置。  
因此，使用在线修正时，需激活驱动器端的终点限位监控。
  - 在“手动”（点动模式）以及“手动进给至参考点”的模式下，无法使用在线修正。
  - G76 接近由当前修正值偏置的位置。
  - CPL 函数 PPOS 以及 PCSPROBE/PCPROBE 不考虑在线修正的修正值。
  - G75 会测量实际的准确位置（包括在线修正）。
  - “FsProbe”会测量实际的准确位置（包括在线修正）。



如果括号表达式“(0)”或“(1)”不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

## 6. 57 初始化“加工中测量” "InitMeas, IME"

- 描述:
- 功能“初始化加工中测量” (IME) :
- 为“加工中测量”(FME) 函数初始化驱动器的探头功能。
  - 以块到块方式起作用或作为一个局部函数
  - 对插补类型或有效模态组的插补类型 G0、G1、G2…无影响

语法:

|                          |                 |
|--------------------------|-----------------|
| IME ( MpiAxis <i> )      |                 |
| 或                        |                 |
| InitMeas ( MpiAxis <i> ) |                 |
| "MpiAxis"                | 始终在指数 <i> 之前编程。 |
| <i>                      | 测量轴的系统轴分度       |

表格 6-73: IME (InitMeas) 语法

- 特殊功能和限制:
- 调试:  
对 IME 函数进行编程之前，应首先操作“加工中测量”。所有要求的步骤参见“《Rexroth IndraMotion MTX 功能描述》”手册，章节“测量功能”。
  - 应在调用 FME (FlyMeas) 函数之前调用 IME (InitMeas) 。

## 6. 58 逆时间编程 "InvTime, ITIM"

- 描述:
- ITIM 仅在 G93 模式下有效。编程的 F 字的单位由 ITIM（参见 第 5.2.33 章“时间编程 G93”第 146 页）定义。

语法:

|              |                                   |
|--------------|-----------------------------------|
| ITIM <Value> |                                   |
| 其中           |                                   |
| <Value>      | 整数；在 G93 模式下，F 字的单位相对于 ITIM 值发生变化 |
| 0            | F 值，单位为 [s]                       |
| 1            | F 值，单位为 [1/s]                     |
| 60           | F 值，单位为 [1/min]                   |

表格 6-74: 反时间规划的语法

ITIM>0 的单位来源于

使用高级语言语法的 NC 功能

$$F = \frac{v \left[ \frac{mm}{ITIM \times sec} \right]}{s[mm]} = \frac{v \left[ \frac{1}{s} \right]}{s \left[ \frac{min}{min} \right]}_{ITIM=60} = \frac{v \left[ \frac{1}{s} \right]}{s \left[ \frac{sec}{sec} \right]}_{ITIM=1}$$

特殊功能和限制:

ITIM 可以在 Init 字符串中设置。  
默认值是 “0” （时间编程以秒为单位）。  
如果通过 ITIM 切换时基，则必须编程一个新的 F 值。

6. 59

选择准停定位窗口 "IPS1、IPS2、IPS3"

描述:

NC 命令 IPS1、IPS2 和 IPS3 可总体预设准停定位窗口的大小，该函数对快速进给和进给模式有效。

|      |            |
|------|------------|
| IPS1 | 精确确切停止窗口大小 |
| IPS2 | 粗略的停止窗口大小  |
| IPS3 | 无限准确停止窗口大小 |

表格 6-75: 可使用 IPS1、IPS2、IPS3 设置的准停定位窗口

若来自模态组 {G0, G1, G2, G3, G5, G6, G33} 的当前激活数控函数的准停定位处于活动状态，则以上预设置将立即生效。

语法:

|      |                                                                          |
|------|--------------------------------------------------------------------------|
| IPS1 | <b>精细准确停止窗口尺寸:</b><br>在块终点，控制器首先把路径速度降低到 0。只有在涉及到所有轴的准确停止窗口时，才会遍历下一个程序段。 |
| IPS2 | <b>粗糙准停定位窗口尺寸:</b><br>否则与 IPS1 特性相同。                                     |
| IPS3 | <b>无限准停定位窗口尺寸:</b><br>在块终点，控制器只把路径速度降低到 0。确切的停止窗口没有被检查。                  |

表格 6-76: IPS1、IPS2、IPS3 的效果（若准停定位对于来自模态组 {G0, G1, G2, G3, G5, G6, G33} 的当前激活的数控函数处于活动状态）

特性和限制:

若来自模态组 {G0, G1, G2, G3, G5, G6, G33} 的当前激活数控函数的准停定位未处于活动状态，可使用 IPS1、IPS2、IPS3 来预设快速进给和进给模式的准停定位窗口大小。但是这个设置不适用。

除对 IPS1、IPS2 和 IPS3 编程外，准停定位窗口的大小可按以下方式更改:

- G0(IPS1)、G0(IPS2)、G0(IPS3)（仅对快速进给有效）
- G1(IPS1)、G1(IPS2)、G1(IPS3)、G61(IPS1)、G61(IPS2)、G61(IPS3)（仅对进给模式有效）。



有关停止函数的详细信息，请参见本手册中的 G0、G1、G61/62，以及“函数说明”手册中的一章的“精确停止”“精度精度”小节。

6. 60

确定系统轴以规定过冲限制启用情况下的最佳 形状 过滤器 "JerkAxList、JAL"

描述:

修改或确定计算活动过冲限制的形状阶数时需考虑的可能系统轴的列表。  
该数控函数也可用作“机床参数”。

如需了解更多详细信息，参见“《Rexroth IndraMotion MTX 函数说明》”手册中“特定轴的过冲限制”。

语法:

|                                                 |                                      |
|-------------------------------------------------|--------------------------------------|
| 在零件程序中进行绝对编程:                                   |                                      |
| JerkAxList() 或<br>JerkAxList(DEF)               | 启用来自 CHAN/Ch[k]/Ini/NcResetState 的默认 |
| JerkAxList(<AbsList>)                           |                                      |
|                                                 | 选择编程系统轴                              |
| 在零件程序中进行相对编程:                                   |                                      |
| JerkAxList({DEF}{,-,<MinusList>}{+,<PlusList>}) |                                      |
|                                                 | 更改轴系统轴列表                             |
| 缩写形式: JAL(..)                                   |                                      |
| 其中                                              |                                      |
| DEF                                             | 默认来自 CHAN/Ch[k]/Ini/NcResetState。    |
| <AbsList><br><MinusList><br><AbsList>           | 系统轴列表, 以逗号分隔。                        |
| 在 CHAN/Ch[k]/Ini/NcResetState 中进行绝对编程:          |                                      |
| JerkAxList(<AbsList>)                           |                                      |
|                                                 | 选择编程系统轴                              |
| 在 CHAN/Ch[k]/Ini/NcResetState 中进行相对编程:          |                                      |
| JerkAxList(-,<MinusList>)                       |                                      |
|                                                 | 移除编程系统轴。                             |
| 缩写形式: JAL(..)                                   |                                      |
| 其中                                              |                                      |
| <AbsList><br><MinusList>                        | 系统轴列表, 以逗号分隔。                        |

表格 6-77: Syntax JerkAxList, JAL

示例:

初始情况:  
第一通道控制三个线性轴 (XA、YA、ZA) 和回转台 (CA) 的加工单元。通道中临时包括 VA 轴。  
回转台 (CA) 仅用于转动工件, 因此与处理无关。  
VA 与处理相关。

输入 CHAN/Ch[k]/Ini/NcResetState (MP 7060 00010) JerkAxList(-,CA)。

或者, 可采用以下类型:  
JerkAxList(XA,YA,ZA,VA)

使用高级语言语法的 NC 功能

- 特殊功能和限制:
- 如果 在 CHAN/Ch[k]/Ini/NcResetState (MP 7060 00010) JerkAxlst(-,CA) 中输入, 该函数作用与机床参数相似。

绝对编程:  
JerkAxlst( <SysAchsList> )

相对编程:  
JerkAxlst( - , <SysAchsList> )

如果未编入该函数, 过冲限制启用情况下计算形状过滤器时, 使用轴被传递时构成或成为一个通道成员的所有轴

6. 61 激活急进限制 "JerkControl、JKC"

描述: 在通道中启用急进限制



有关轴转换功能性的详细信息, 请参阅文档 "IndraMotion MTX 功能描述"。

语法:

|                                   |        |
|-----------------------------------|--------|
| JerkControl 或<br>JerkControl(1)   | 启用过冲限制 |
| JerkControl() 或<br>JerkControl(0) | 停用过冲限制 |
| 缩写形式: JKC(..)                     |        |

表格 6-78: JerkControl、JKC 语法

示例:

N10 JKC(1) 启用过冲限制  
:  
N90 JKC(0) 停用过冲限制。

6. 62 工件坐标中的点动模式 "JogWCSSelect"

- 描述: 对于操作模式 "工件坐标中的点动设置模式", 该函数定义了使用手轮进行点动或进给的坐标。
- 可选择以下的坐标进行点动:
- 若无激活的轴变换, 则选择各机床轴的所有坐标 (伪坐标)。

若存在激活的可定向轴变换, 则选择 (与激活的 WCS 有关的) 所有的线性和定向坐标。

若可定向的轴变换处于活动状态, 则为 TCS 坐标 z。

MP 7050 01010 中的设置决定进给和步宽 (对于增量点动) 是否得自现有机床轴的设置, 或必须明确地对它们编程。



有关轴转换功能性的详细信息, 请参阅文档 "IndraMotion MTX 功能描述"。

语法:

|                                                                                         |                                           |
|-----------------------------------------------------------------------------------------|-------------------------------------------|
| JogWCSSelect {JWSCHAN<ChannelNo>} JWSCoord<CoordNo><br>JWSFEED<F value>{JWSSTEP<Incr>}} |                                           |
| 其中                                                                                      |                                           |
| <ChannelNo><br>>                                                                        | 要在其中点动某个坐标的通道的编号。<br>默认: 要在其中对函数编程的通道的编号。 |

|           |                                                                                   |
|-----------|-----------------------------------------------------------------------------------|
| <Coordno> | 坐标选择<br>取值范围：1 至 8 以及 103<br>整数<br>1 至 8：要点动的坐标的编号。<br>103：要点动的 TCS 坐标。           |
| <F-value> | 点动速度<br>根据激活的测量单位（G71、G70），输入的值以毫米/分钟、英寸/分钟或度/分钟为单位。<br>仅在 MP 7050 01010 = 0 时有效。 |
| <Incr>    | 选择增量点动，以及增量所需步长大小规范。<br>仅在 MP 7050 01010 = 0 时有效。                                 |

表格 6-79: JogWCSSelect 语法

特殊功能和限制：该函数设计用于特殊的坐标选择数控程序（另请参见 MP 7050 01110）。有关更多信息，请参阅文档“IndraMotion MTX 功能描述”。

### 6. 63 禁用径向和切向加速度的总和产生的速度限制 "KinLimAcc, KLA"

描述：KLA 功能抑制了为超驰情况下的速度变化提供的保留加速度。  
提供这种保留加速度时，使用 Hermite 样条和 CVP 指定客户特定的速度参数文件时，可能偶尔无法达到已编程速度值。

|     |               |                       |
|-----|---------------|-----------------------|
| 语法： | KinLimAcc (1) | 禁用径向和切向加速度的总和产生的速度限制。 |
|     | KinLimAcc (0) | 重新激活速度限制。             |
|     | 缩写形式：KLA      |                       |

表格 6-80: KinLimAcc 语法

特殊功能和限制：KLA 功能作用于 CVP 和 Hermite 样条（样条类型 1）。

### 6. 64 停用运动数控块分割 "KinSplitDisable、KSD"

描述：对于时间优化的处理，如有必要，可将数控块分割为多个块，以允许路径进给更好地调整至不同路径相关的最大速度。  
该特性在某些情况下并不需要，例如编程的数控块以缓慢但常量的速度进给。为此，可以将运动块分割停用。

|     |                     |           |
|-----|---------------------|-----------|
| 语法： | KinSplitDisable (1) | 停用运动数控块分割 |
|     | KinSplitDisable (0) | 激活运动数控块分割 |
|     | 缩写形式：KSD            |           |

表格 6-81: KinSplitDisable 句法

特殊功能和限制：控制器启动后，块分割启用。

### 6. 65 C0 编程 "KvProg、KVP"

描述：该函数允许以受程序控制的方式更改各轴的 KV 值。该函数可临时增加轴的刚度（例如用于铣削孔）。  
控制器上电后，NC（包括其他因素）将读取所有耦合驱动器的当前 C0 值，并将其保存。

使用高级语言语法的 NC 功能

在进行 C0 编程时，控制器会向相关驱动器载入编程的 C0 值（参数 S-0-0104）。

C0 编程关闭时，控制器会将之前保存的 C0 值传输回驱动器，以恢复初始状态。

以下情况适用：

- $C0 = (\text{路径速度, 单位: m/min}) / (\text{滞后, 单位: mm})$
- 在包含 C0 开关的各块前，将速度减至  $v=0$ ，这是因为只可在静止时切换驱动器中的 C0 值。
- 触发 C0 切换的命令后，控制器会在内部等待所有相关驱动器的确认。
- 在能够于相同块内进行编程的任何进给运动之前，会执行 C0 切换。

示例：

```
:
N10 G1 F1000
:
N40 X40 使用默认的 C0 值定位
N50 KVP(X2) X50 将 C0 值 2 传输至 X 轴，然后进行进给。
:
```

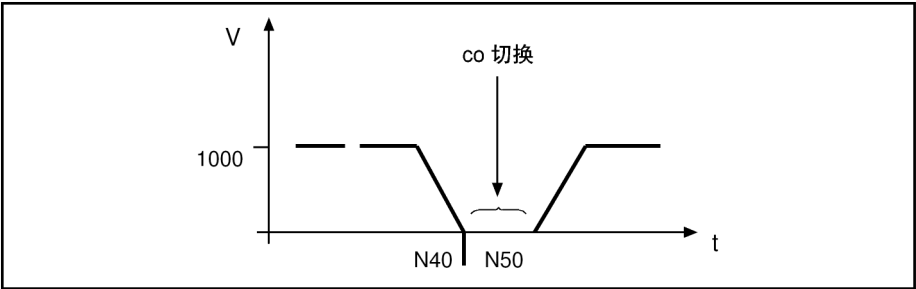


插图 6-12: KvProg (KVP) 的示例 1

|     |                                 |                                                             |
|-----|---------------------------------|-------------------------------------------------------------|
| 语法： | KvProg(<Axis1>{,<Axis2>}{,...}) |                                                             |
|     |                                 | C0 编程开启                                                     |
|     | KvProg({0})                     | C0 编程关闭                                                     |
|     | 缩写形式：KVP(..)                    |                                                             |
|     | 其中                              |                                                             |
|     | <Axis>                          | 物理（系统范围）或逻辑（基于通道）轴的名称，包括所需的 C0 值。<br><br>可编程的最大 C0 值：655.35 |

表格 6-82: KvProg (KVP) 语法

```
示例：
: 主页位置：所有驱动器中均有一个大小为 1.0 的活动 C0 值。
N10 G0 X0 Y0 以 C0 值 = 1.0 进行定位
Z100
N20 将 C0 值 2.1 传输至物理轴 Z 的驱动器。
KVP(Z2.1)
:
```



N30 G1 Z0           以 C0 值 = 2.1 进行定位

N40 KVP ()           取消 C0 编程

Z100                NC 会自动为所有驱动器重新载入一个大小为 1.0 的 C0 值。

:

然后 Z 轴将进行进给。

特殊功能和限制：       仅可对当前分配至通道的轴进行编程。

## 6. 66       迟滞补偿 "LCP"

描述：       "LCP" (LagCompensation) 以系统轴为基础生效。在数控函数中，轴必须由其系统轴名标明地址。

语法：       激活组：

|                                                             |                                                                 |                              |
|-------------------------------------------------------------|-----------------------------------------------------------------|------------------------------|
| N100 LCP (AbsMode, <Slave1>, <OnOff>, <Slave2>, <OnOff>...) |                                                                 |                              |
| 其中                                                          |                                                                 |                              |
| AbsMode                                                     | 如果 "AbsMode" 已编程，则附加轴就被控制在其零点位置周围。为此，当补偿打开时，MLD 将附加轴/轴绘制到 0 位置。 |                              |
|                                                             | 如果尚未编入 "AbsMode"，则不会进行同步运动。附加轴被控制在其实际位置附近。                      |                              |
|                                                             | <Slave i>                                                       | 附加轴的系统轴名称                    |
|                                                             | <OnOff>                                                         | 0:<br>停用耦合<br><br>1:<br>激活耦合 |

表格 6-83:       LCP 语法

通过该调用，可将一个或多个附加轴与配置中为其定义的主轴进行耦合（或去耦）。

停用所有组：

N200 LCP ()

       该函数的具体描述参见 "《Rexroth IndraMotion MTX 功能描述》"手册。

## 6. 67       分隔进给块：路径段长度 "LEN"

描述：       将编程的进给块分隔成数个等长度的部分路径。

       仅在结合 "冲孔"（请参见第 第 6.93 章 "冲孔 Punch、PUN" 第 302 页 页）和 "裁料"（请参见第 第 6.79 章 "裁料 Nibble、NIB" 第 274 页 页）函数使用时才有效。

使用高级语言语法的 NC 功能

|     |             |                                                                                                                                     |
|-----|-------------|-------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | LEN=<value> |                                                                                                                                     |
|     | 其中          |                                                                                                                                     |
|     | <Value>     | 对于线性块：路径段长度<br>对于圆形块：弧长度<br>编程单位与轴坐标相同<br><br><Value> 不必为编程路径长度的整系数。NC 会在内部自动计算激活的 LEN 值，该值小于或等于编程的 LEN 值，其效果是让有效路径段始终是已编程路径长度的整除数。 |

表格 6-84:        LEN 语法

- 示例:
- G90 处于活动状态（绝对尺寸编程）
  - 激活平面：X/Y
  - 当前位置：X=0, Y=0, C=0
  - 冲压 / 步冲关闭

|                                |                                                                                                                              |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| :                              |                                                                                                                              |
| N20 X100 Y100 LEN=15           | 以下的进给块已被分割为多个常规路径区段，每个最大长度为 15 mm。由于冲压 / 步冲关闭，LEN 尚未生效。                                                                      |
| N30 X200 Y200 C180<br>Punch(1) | 冲孔打开<br>LEN 生效<br>路径长度分成 10 个块段。<br>结果冲程位置 (X, Y, C):<br>P1 (110, 110, 18)<br>P2 (120, 120, 36)<br>:<br>P10 (200, 200, 180)。 |
| N40 Y290 C210                  | 路径长度 (90 mm) 分为六个块段。结果冲程位置 (X, Y, C):<br>P11 (200, 215, 185)<br>P12 (200, 230, 190)<br>:<br>P16 (200, 290, 210)。             |
| N50 Punch()                    | 冲孔关闭                                                                                                                         |
| :                              |                                                                                                                              |

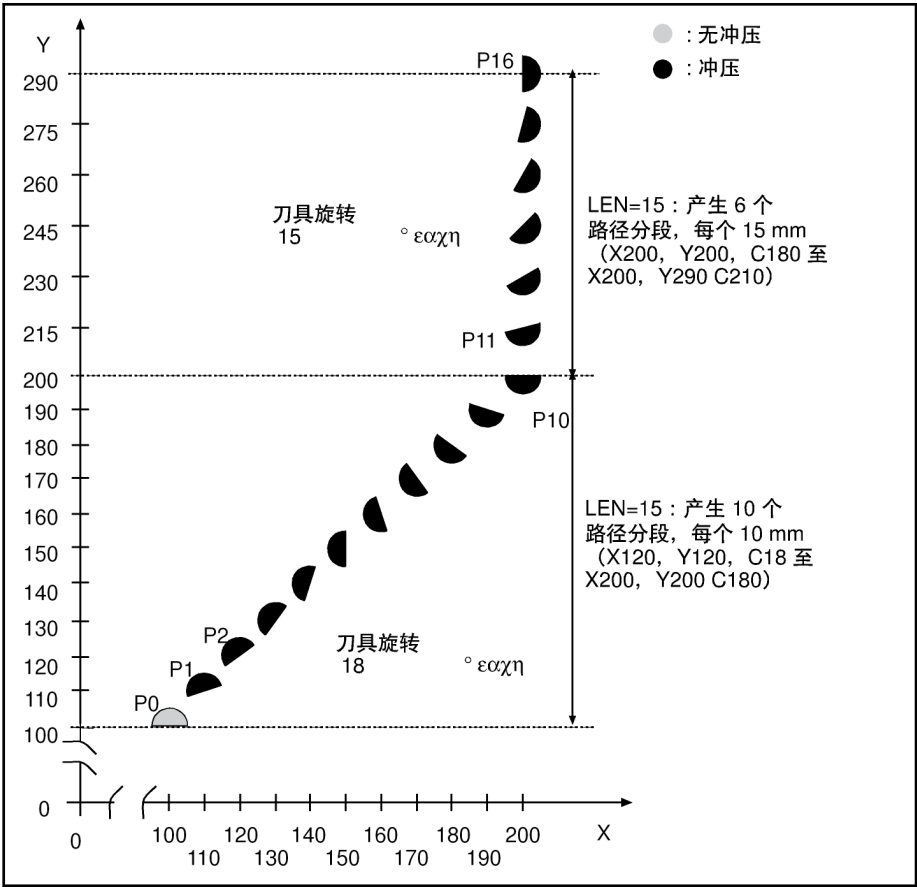


插图 6-13: LEN 的示例

- 特殊功能和限制:
- 若为已分割的圆形块, 则一个冲程到另一个冲程的运动始终为线性。
  - LEN 在冲压 / 步冲函数处于活动状态时始终以模式形式工作。但是在块到块的基础上会被 NUM (参见 第 6.80 章 “分隔进给块: 路径段数 NUM” 第 276 页 ) 覆盖。
  - 在 “冲压 / 步冲” (MP 8001 00010) 已释放时, 可在任意时间对 LEN 进行编程。但是, 只有在 “冲压 / 步冲” 激活后才会块分割。

## 6.68 与路径速度相关的激光功率控制器 "LFP", "LFConf、LFC"

描述: 根据当前的路径速度  $v_{path}$ , 使用模拟电压信号 (0...10 V) 来控制激光的功率。



有关轴转换功能性的详细信息, 请参阅文档 “IndraMotion MTX 功能描述”。

语法:

|                    |                                |
|--------------------|--------------------------------|
| LFP (1) or<br>LFP  | 使用当前的参数设置打开激光功率控制器。            |
| LFP ({LL (... )})) | 使用 LFCont 中描述的附加参数设置打开激光功率控制器。 |
| LFP (0)            | 激光功率控制器关闭                      |

使用高级语言语法的 NC 功能

|                                                                                                             |                                                                                                          |
|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| LFCnf ({LL(<Volta>,<Vmin>)},<br>{UL(<Volta>,<Vmax>)},<br>{PL(<Mode>)},<br>{CD(<Coord 1>{,<Coord n>}...)}),) | 参数化激光功率控制<br>使用 PL(..) 或 CD(..) 选择相关坐标。                                                                  |
| 缩写形式: LFC(..)                                                                                               |                                                                                                          |
| 其中                                                                                                          |                                                                                                          |
| <Volta>                                                                                                     | 以最大输出电压 (10 V) 的百分比表示的电压<br>与 <Vmin> 一起使用:<br>路径速度低于 <Vmin> 时输出。<br>与 <Vmax> 一起使用:<br>路径速度高于 <Vmax> 时输出。 |
| <Vmin>                                                                                                      | 路径速度下限<br>根据激活的测量单位 (G71、G70), 输入的值以毫米/分钟、英寸/分钟或度/分钟为单位。                                                 |
| <Vmax>                                                                                                      | 路径速度上限<br>根据激活的测量单位 (G71、G70), 输入的值以毫米/分钟、英寸/分钟或度/分钟为单位。                                                 |
| <Mode>                                                                                                      | 定义要用来确定 v <sub>路径</sub> 的坐标:<br>APL 当前平面的所有坐标。<br>ASP 当前空间的所有坐标。<br>CFD 根据 MP 7050 00820 而得到的坐标。         |
| <Coord x>                                                                                                   | 定义要使用哪些坐标来确定 v <sub>路径</sub> 。<br>输入值有:<br>相关空间坐标的名称 (若轴变换处于活动状态) 或逻辑轴名称 (伪; 若轴变换未处于活动状态)。               |

表格 6-85: LFP、LFCnf (LFC) 的语法

示例:

|                                 |                                                                                                       |
|---------------------------------|-------------------------------------------------------------------------------------------------------|
| LFP (LL(10,100),<br>UL(60,700)) | 激光功率控制开启<br>附加参数的设置:<br>若路径速度下降至低于 100 mm/min, 则输出电压为 10% (=1V), 若路径速度上升至大于 700 mm/min, 则为 60% (=6V)。 |
| LFP (0)                         | 激光功率控制器关闭                                                                                             |
| LFCnf (PL (APL))                | 仅限参数设定:<br>路径速度来自活动平面内的路径运动。                                                                          |
| LFCnf (CD (X , Z))              | 仅限参数设定:<br>路径速度来自 X 和 z 坐标的运动。                                                                        |
| LFCnf (UL (80,500))             | 仅限参数设定:<br>若路径速度升至 500 mm/min 以上, 则将输出电压限制为 80% (=8V)。                                                |

- 特殊功能和限制:
- 可用的模拟输出会限制可以使用此功能的通道数。
  - 激光控制器的电压信号在以下情况下降至 0 V:

使用高级语言语法的 NC 功能

- 发生错误时（运行时错误、1 类诊断错误）。
- 取消相关驱动器的“运行中的驱动器”信号（无释放，驱动器关闭）时。
- “进给停止”信号变为活动状态。



如果括号表达式“(0)”或“(1)”不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

## 6. 69 线性样条曲线切换 "LinearToSpline、LTS"

**描述：**该函数可将编程的线性插补 G1 改为样条插补编程 G6。

|            |                   |                                      |
|------------|-------------------|--------------------------------------|
| <b>语法：</b> | LinearToSpline(1) | 开启<br>以下 G1 命令将更改为 G6。               |
|            | LinearToSpline(0) | 关闭<br>以下所有 G1 命令保持为 G1。<br>缩写形式为 LTS |

表格 6-86: 语法



如果括号表达式“(0)”或“(1)”不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

**示例：**程序部分

```
N10 SplineDef(2203,X,Y,Z) LTS(1)
N20 G1
与以下内容效果相同
N10 SplineDef(2203,X,Y,Z)
N20 G6
```

## 6. 70 模态轴（线性循环轴）回零 "LinModZp、LMZ"

**描述：**“LinModZp”用于将线性循环轴的当前位置定义为新的程序零点。之后，轴显示将跳至数值“0”。偏移值是线性模数轴的内部控制计算值。该偏移值被添加到所有后续编程的轴位置。

|            |                                                 |                                                                      |
|------------|-------------------------------------------------|----------------------------------------------------------------------|
| <b>语法：</b> | LinModZp{(<LinModAxis>)}<br>缩写形式: LMZ(..)<br>其中 |                                                                      |
|            |                                                 |                                                                      |
|            | <LinModAxis>                                    | 可选参数：需要进行零位调整的线性模轴的系统轴索引、系统轴名称或通道轴名称<br>如果没有指定轴，则将重新为通道的所有线性模数轴设置零点。 |

表格 6-87: LinModZp (LMZ) 语法

**示例：**LinModZp (LMZ)

使用高级语言语法的 NC 功能

|      |         |         |                                              |
|------|---------|---------|----------------------------------------------|
| N50  | LMZ (1) | X20     | 使用系统轴索引 1 将线性模态轴的当前位置定义为新的程序零点。随后移动至 X20 位置。 |
| N90  |         | LMZ (X) | 将 X 轴的当前位置定义为新的程序零点。然后将 X 轴移动至位置 -43.7。      |
| N120 | LMZ     | G91     | 将通道中的所有线性模数轴重置为零，然后将 x 轴正向移动 7.3 mm。         |

- 特殊功能和限制：
- 线性模数轴的运行范围受限于轴的当前零点的极限值“**减模数值**”和“**加模值**”。
  - 编程超出极限值定义的运行范围的位置会导致运行时错误。
  - 线性模数轴可以在这些电流极限值内绝对或递增地运行。
  - 使用 LMZ 命令，将当前轴位置作为新的零点，并且移动范围相应地移动。
  - 通过再次永久设置零点，可以将线性模数轴无限循环到一个方向。由于指向驱动器的指令位置是模数计算的，因此不会导致位置溢出。
  - 即使线性模轴是驱动端的模轴，从程序的角度来看，它就像一个普通的线性轴一样响应模数值，像终端开关一样有效。唯一的特点是可以通过 LMZ 命令来移动零点以及移动范围。
  - 横移运动可以与 LMZ 命令一起编程在一行中。在这种情况下，LMZ 命令不会中断运动序列。



在 Sercos 启动时，使用 Sercos 参数 S-0-0103 在驱动器中定义模态值。更改后的模态值只在 Sercos 重启后才会生效！

## 6.71 速度变化图（斜坡功能）

**描述：** 该函数允许定义加工段，其执行采用单独的 速度曲线。  
为此，可使用以下“单个块”：

- 1 个常量插补器
- 3 个加速与 3 个制动插补器  
(各插补器带线性、正弦、sine<sup>2</sup> 线型速度变化图)。

以下描述的所有函数均以模态形式生效，构成一个带 G8 和 G9 的模态组，并相互取消选定。

以下情况适用：

- 加速插补器 “LinUpFeed、LNU”， “SinUpFeed、SNU”， “Sin2UpFeed、S2U”**  
从速度  $V_0$ （块开关处的速度）开始，控制器在整个编程路径长度上减速，直至目标速度 ( $V_1$ )。  
根据已编程函数，通过线性、正弦曲线或正弦<sup>2</sup> 线型速度增加来实现该目的。  
由此达到目标速度  $V_1$  和编程的终点，同时从编程的进给中得到的结果将作为当前覆盖值的函数。  
其限制因素为
  - 最大路径加速度和
  - 许可的最大路径速度。各路径区段和各数控块的专用控制器会计算这两个量；单块预读的执行与允许的最大速度有关。这可避免在下一块中违反最大轴速度。



若  $V_1$  不大于  $V_0$ ，则忽略当前块中的加速插补器调用。

发生覆盖更改时的特性:

- 在重新计算加速坡度时，增加覆盖结果。
- 如果降低后覆盖最终值低于启动速度  $V_0$  ，
  - 在 “LinUpFeed” 情况下会导致对制动坡度重新计算；此情况会持续至编程的终点。
  - 而在 “SinUpFeed” 和 “Sin2UpFeed” 的情况下会忽略此情况。

2. 效果常量插补器 “ConstFeed、CFD”

控制器考虑当前的倍率值和最大允许路径速度，尝试达到编程的目标速度。  
在每种情况下，会使用许可的加速度在覆盖更改期间进行速度更改。

3. 减速插补器 “LinDownFeed、LND”， “SinDownFeed、SND”， “Sin2DownFeed、S2D”

从速度  $V_0$  （块开关处的速度）开始，控制器在整个编程路径长度上减速，直至静止 ( $V_1=0$ )。  
根据已编程函数，通过线性、正弦曲线或正弦<sup>2</sup>-线型速度减小来实现该目的。  
覆盖更改将没有效果，但以下情况除外：  
若前一块中覆盖设置为 0%，并且之后在块过渡至减速插补器处精度达到目标速度 0，控制器将维持减速插补器的工作，直至覆盖值增加为大于 0 的值！  
使用一个加速步长来增加速度（取决于允许的路径加速度）。  
控制器以从此函数得出的速度值为基础来计算必要的减速坡度。之后，实际覆盖值在达到块终点前将无效。

语法:

|                           |                                      |
|---------------------------|--------------------------------------|
| LinUpFeed<br>缩写形式: LNU    | 通过线性速度增加激活加速插补器。                     |
| SinUpFeed<br>缩写形式: SNU    | 通过正弦速度增加激活加速插补器。                     |
| Sin2UpFeed<br>缩写形式: S2U   | 激活速度增加 sine <sup>2</sup> -线型的加速度插值器。 |
| ConstFeed<br>缩写形式: CFD    | 激活常量插补器                              |
| LinDownFeed<br>缩写形式: LND  | 通过正弦线型速度下降激活加速插补器。                   |
| SinDownFeed<br>缩写形式: SND  | 通过正弦速度降低激活加速插补器。                     |
| Sin2DownFeed<br>缩写形式: S2D | 通过正弦 <sup>2</sup> -线型速度降低激活加速插补器。    |

表格 6-88: LNU、SNU、S2U、CFD、LND、SND 和 S2D 语法

示例:

U 轴摆动循环的速度曲线

使用高级语言语法的 NC 功能

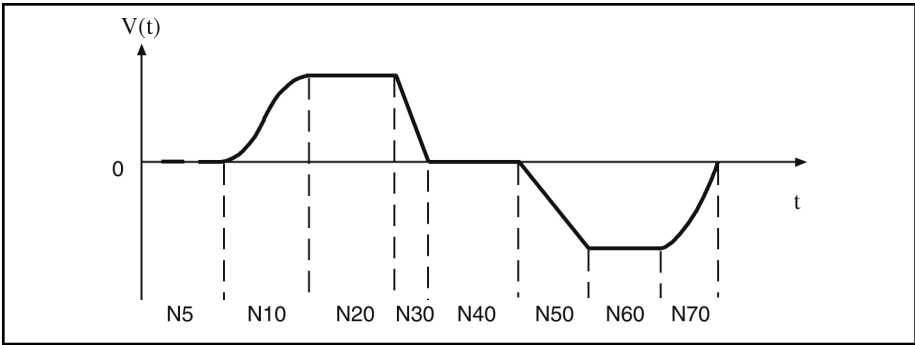


插图 6-14: U 轴摆动循环的速度曲线

|     |              |                                              |
|-----|--------------|----------------------------------------------|
| N5  | G0 U10       | U 轴进给至起始位置 (U=10mm)                          |
| N10 | S2U U17 F500 | Sine² 线型加速至位置 U=17<br>结束点的命令提要: F=500 mm/min |
| N20 | CFD U23      | 保持常量, 直至位置 U=23                              |
| N30 | LND U29      | 线性制动, 直至至位置 U=29<br>最终速度: 0 mm/min           |
| N40 | G4 F0.5      | 反转点的停留时间。                                    |
| N50 | LNU U20      | 线性加速, 直至位置 U=20                              |
| N60 | CFD U17      | 保持常量, 直至位置 U=17                              |
| N70 | SND U10      | 正弦线型减速, 直至位置 U=10。<br>最终速度: 0 mm/min         |

- 特殊功能和限制:
- 必须始终在编程块中输入所需的终点坐标。
  - 仅可在“自动 / 继续块”操作模式下使用所有斜度函数。使用其他操作模式（手动数据输入、单块、单步或程序块）会导致运行时错误。
  - 在极短的进给路径上使用制动插补器可能导致过度减速，从而导致伺服错误。  
因此，在创建零件程序时，请遵守机床动态当前最大可能值。
  - 在常量和加速插补器处于活动状态时，不允许使用辅助函数，或如“准停定位”等函数，否则将可能导致速度突然降低。使用这些功能会导致突然的速度下降！  
禁止使用以下函数：  
G0, G4, "KvProg", G63, G33, G61, "G1(IPS)", G75, "MOB", "G74(HOME)", "HsBlkSwitch", "WriteId".

6. 72 主轴切换 "MainSp、MSP"

|     |                                                                                                                                                                                   |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述: | 与 MP 7020 00010 的设置不同，它将给定主轴定义为当前通道中的主轴。<br>因此，可以在零件程序中确定应使用以下功能的主轴： <ul style="list-style-type: none"><li>• G33（螺纹切削），</li><li>• G95（进给，以 mm/转编程）以及</li><li>• G4（停留时间）</li></ul> |
| 语法: | <div>MainSp(&lt;Num&gt; ) or MainSp(&lt;Spindle&gt;)</div> <div>缩写形式: MSP...</div>                                                                                                |



|           |                                                              |
|-----------|--------------------------------------------------------------|
| 其中        |                                                              |
| <Num>     | 主轴编号（主轴索引）<br>输入范围：-1; 1...8<br>整数<br>-1：根据 MP 7020 00010 主轴 |
| <Spindle> | 与 <Num> 或主轴名称相同（如 S1）                                        |

表格 6-89: MainSp (MSP) 语法

## 6.73 输入工具：镜像 "Mirror(...)、MIR(...)"

**描述：**“镜像”函数是输入工具的一种。

控制器加工编程的轮廓，或者以诸如镜像反转的方式来加工镗孔样式。

镜像始终参照当前的镜像点（参见“PoleSet”函数（第 6.86 章“定义镜像点 / 旋转点 PoleSet、PLS”第 291 页））。若未对此进行明确编程，则当前的程序零点为镜像点。



镜像是一个输入工具，因此它不会更改当前的程序坐标系。输入工具只为程序坐标提供其他输入方法。

镜像也可结合“旋转”和“缩放”使用。

以下情况适用：

- 函数以模态方式作用。其在关闭前保持活动状态。
- 可在同一块中将之作为其他路径条件和输入工具进行编程。

**语法：**

|                                |                                                                                          |
|--------------------------------|------------------------------------------------------------------------------------------|
| Mirror(<Axis 1>1{ ,<Axis n>1}) |                                                                                          |
|                                | 激活给定轴的镜像。                                                                                |
| Mirror(<Axis 1>0{ ,<Axis n>0}) |                                                                                          |
|                                | 取消给定轴的镜像。                                                                                |
| Mirror(0) 或<br>Mirror()        | 取消通道中所有轴的镜像。<br>所接近的轴位置在被重新编程前始终有效。                                                      |
| 缩写形式：MIR(...)                  |                                                                                          |
| 其中                             |                                                                                          |
| <Axis1>, <Axes>                | 由该函数将激活结合数值“1”的轴地址（如 X）：<br>对应轴（如 X100）所有后续编程的路径命令均会在内部乘以数值“-1”。<br>镜像在下一条进给信息前不会处于活动状态。 |

表格 6-90: Mirror(...) 和 MIR(...) 的语法

**示例：** 镜像

使用高级语言语法的 NC 功能

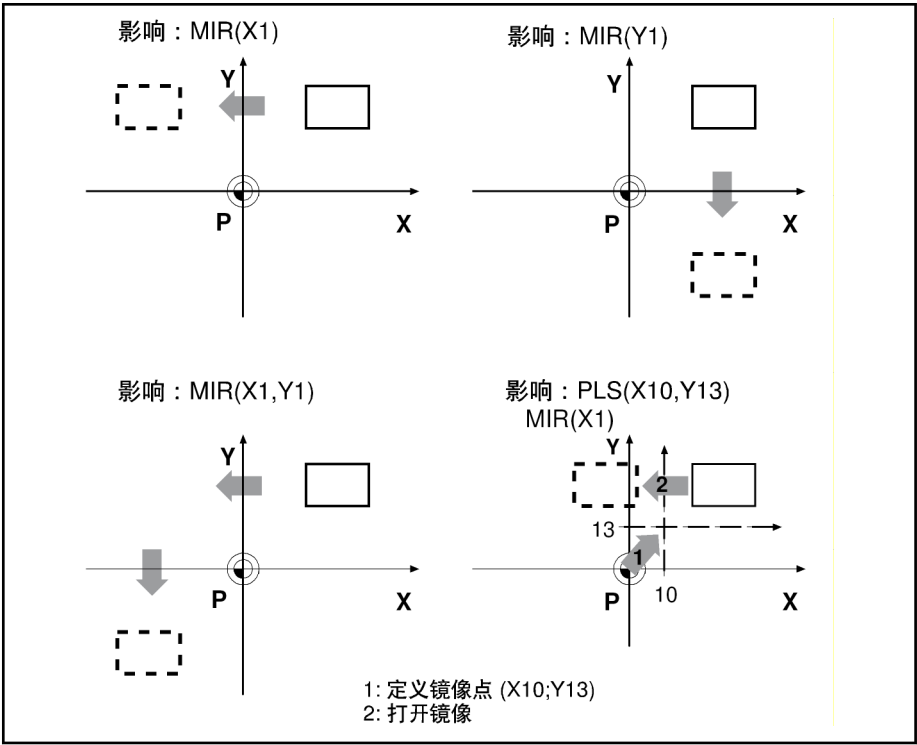


插图 6-15: 镜像示例

特殊功能和限制:

- 该函数在圆周插补的情况下会考虑插补参数
- 它影响着可编程的轮廓切换。  
请参阅 第 6.114 章 “编程的轮廓移位 Shift、SHT” 第 324 页 。
- 它 不 影响:
  - 零点偏置 (G54 – G59.5;  
请参阅 第 5.2.21 章 “零点偏置 (Z0) G53、G53.1–G59.1 至 G53.5–G59.5” 第 135 页 )
  - 程序坐标偏置  
(Trans 或 Atrans; 参见 第 6.148 章 “程序坐标偏置 Trans、TRS, 叠加程序坐标偏置 ATrans、ATR” 第 361 页 ),
  - 设定程序位置  
(“SetPos”; 参见 第 6.113 章 “设置程序位置 SetPos、SPS” 第 323 页 ),
  - 接近参考点坐标  
(G74; 参见 第 5.2.26 章 “接近参考点坐标 G74 ” 第 141 页 ),
  - 接近固定的机床轴位置  
(G76; 参见 第 5.2.29 章 “接近固定的机床轴位置 G76” 第 143 页 )
  - 铣削半径和刀具长度修正值。

定向向量的镜像:

仅可以逐个组件为基础使用以下函数语法对定向向量进行镜像。  
任何可能处于活动状态的缩放，或镜像/旋转点对结果没有任何影响。

语法:

|                             |                     |
|-----------------------------|---------------------|
| Mirror(O(<Sx> ,<Sy> ,<Sz>)) |                     |
|                             | 激活 / 取消定向向量给定组件的镜像。 |
| 缩写形式: MIR(..)               |                     |

|                |                                |
|----------------|--------------------------------|
| 其中             |                                |
| <Sx>、<Sy>、<Sz> | 各向量组件的镜像系数。<br>0: 无镜像<br>1: 镜像 |

表格 6-91: Mirror 语法

以下内容适用于定向向量:

- 无法对极坐标  $\phi$  和  $\theta$  进行镜像。
- 不允许对 MIR(phi1,theta1) 进行编程。

## 6.74 在块上测量 "MOB"

**描述:** MOB 是逐块进行的功能，可用于测量工件位置。  
MOB 与 移动运动 (G1, G6, TcsMove, ...) 编程在一起。

|     |                       |                |
|-----|-----------------------|----------------|
| 语法: | MOB                   |                |
|     |                       | 用探头 1 测量       |
|     |                       |                |
|     | MOB (MP<Touch probe>) |                |
|     |                       | 用探头 1 或 2 测量   |
|     | 其中                    |                |
|     | <Touch probe>         | 选择一个探头 (1 或 2) |

表格 6-92: NC 功能 MOB 的语法

**示例:** 将 MOB 作为 G75.2 的选项

```
..
N100 MOB(MP2) G1 Y250 F100 ;MOB with measuring probe 2
0110 IF SD(9) = 0 THEN
0120 YPOS! = PCSPROBE("Y")
0130 ELSE
0140 SETWARN("NO CONTACT")
N150 M0
0160 ENDIF
..
```

### 在刀具方向测量

```
N010 CRD(2) ;Activate an axis transformation
..
N100 G1 z50 F1000
N110 G1 x50 y70 F1000
..
N200 MOB TCM(, , -20) F100 ;MOB with traversing in TCSz direction
0210 IF SD(9) = 0 THEN
0220 ZPOS! = PCSPROBE("z")
0230 ELSE
0240 SETWARN("NO CONTACT")
N250 M0
0260 ENDIF
..
```

- 特殊功能和限制:**
- 手册 “IndraMotion MTX 功能描述 – 扩展”的“在块上测量 (MOB)”一章中对探头的调试进行了说明。
  - MOB 必须与移动运动编程在一起。

使用高级语言语法的 NC 功能

6.75 在轮廓上测量（MOC）

**描述：** NC 功能 MOC 可用于沿任何所需的路径进行测量。测量路径由运动生成块组成，由一个带参数的 MOC 和一个无括号的 MOC 定义。

 我们建议将测量关断（MOC（））写入“通道重置后的状态”（机床参数 CHAN / Ch [k] / Ini / ChResetState）。否则，测量路径将不会完成，首先不是重置反应（RE3）。

一旦发生测量事件，测量的位置被写入系统数据 / SysMeasPos。此外，G75 的 CPL 功能（如 PROBE，SD（9）...）可用于查询零件程序中的测量位置。

 如果使用 CPL 功能读取测量值，则必须在测量路径末端和 CPL 功能之间设置 WAIT 命令。

如果在测量的起始块中编程运动，则测量即时开始（即没有任何中间停止）。这同样适用于测量路径的末端。  
一次只有一个 MOC 命令处于活动状态。测量可以通过编程另一个测量来完成。

 无法进行多次测量。这意味着，在测量路径内，MOC 逻辑仅在第一次测量事件发生之前被激活，然后关闭。如果 MOC 再次编程，只能进行进一步的测量。

|     |                                                                                                                                                  |                                                                                                                                                                                                                                                    |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法： | MOC (MP<Probe> CI<CustomerInput> HS<HighSpeed>{, ES<Edge>}, RE<Reaction>{, AS<Asup>}{, ER<Error>}{,<ChannelAxisName1>{,<ChannelAxisName2> ...}}) |                                                                                                                                                                                                                                                    |
|     | 其中                                                                                                                                               |                                                                                                                                                                                                                                                    |
|     | <Touch probe>                                                                                                                                    | 选择探头（1 或 2）作为信号源。                                                                                                                                                                                                                                  |
|     | <Customer input>                                                                                                                                 | 选择通道接口（1 ... 8）中的客户输入作为信号源。                                                                                                                                                                                                                        |
|     | <High-speed>                                                                                                                                     | 选择探头（1 或 8）作为信号源。                                                                                                                                                                                                                                  |
|     | <Edge>                                                                                                                                           | 选择下降沿（1），上升沿（2）或两个（3）边缘作为测量触发。如果客户输入或 HighSpeed 输入被选为信号过程，则边沿选择是强制参数。如果我们使用了测头，则选择边缘是可选的。如果没有编程测头，则可以将组态的边沿（参见 IndraMotion MTX 机床参数 MEAS / Dr [i] / Probe1Edge, MEAS / Dr [i] / Probe2Edge 和 MPEDGE, 第 6.78 章“更改测头边缘配置（MPEDGE）”第 273 页）用作 MOC 的触发。 |
|     | <Response>                                                                                                                                       | 测量结果可能会触发以下一种反应：<br><div>1. 切换路径速度<br/>2. 触发一个异步子程序<br/>3. 重置频道<br/>4. 没有反应（加工中实时测量）<br/>5. 跳过剩余的测量路径</div> 有关可能的反应的详细说明，请参见“IndraMotion MTX 功能说明手册”，“轮廓测量”章节。                                                                                     |

使用高级语言语法的 NC 功能

|                    |                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <Asup>             | 测量事件后要执行的已登录异步子程序的编号（仅适用于 RE2）。在调用 MOC 之前，异步子例程必须已经登录到系统（例如，使用 ASPSET，请参阅第 6.8.5 章“异步子程序：注册 ASPSET”第 195 页）。                   |
| <Error>            | 对沿测量路径没有发生的测量事件的反应。<br>潜在的反应是： <ul style="list-style-type: none"><li>0: 没有消息（默认设置）</li><li>1: 输出警告</li><li>2: 输出错误消息</li></ul> |
| <ChannelAxisNameX> | 要测量的所有通道轴的轴名称<br>只有使用测头进行测量时，该数据才有意义。如果未指定轴数据，则测量将针对为测头配置的通道的所有轴执行（请参见文档“IndraMotion MTX 机床参数”MEAS / Dr [i] / EnablProbeFunc）。 |
| MOC ()             |                                                                                                                                |
|                    | 测量路径结束<br>测量关闭，如果未发生测量事件，则根据 ER 生成消息。                                                                                          |

表格 6-93: NC 功能 MOC 的语法

|     |                                                                                      |
|-----|--------------------------------------------------------------------------------------|
| 示例: | N10 ASPSET(3, SAFEPOS01, 11) ;将异步子程序“SAFEPOS01”作为编号 3 登录到系统。                         |
|     | N20 MOC(CI4, ES1, RE2, AS3, X, Z) ;如果客户输入 4 发出下降沿，则执行 Asup 3 (SAFEPOS01) 并测量轴 X 和 Z。 |
|     | N30 X... Y... Z... ; 测量正在进行中                                                         |
|     | N40 X... Y... Z... ; 测量正在进行中                                                         |
|     | N50 X... Y... Z... ; 测量正在进行中                                                         |
|     | N60 X... Y... Z... ; 测量正在进行中                                                         |
|     | N70 MOC() ; 测量已完成（测头已停用）                                                             |

- 特殊功能和限制:
- 只能选择一个信号源（测头/客户输入 / 高速输入）。
  - 如果使用反应 2（触发异步子程序）输入缓冲的 NC 块，则在完成异步子程序后清除块缓冲。在这种情况下，指定的测量路径只能在测量事件发生之前进行。
  - 测量轴可能无法沿着测量路径从通道中移除。
  - 在程序坐标系（PCS）中测量的位置仅在测量通道的所有轴时提供。这是基于在测量时刻激活的（轴）配置。
  - 不允许在测量路径上编程 G75、MOB 或 FlyMeas 命令。不允许用 MPEDGE 重新配置测量边缘。
  - 不允许沿着测量路径改变零点偏移，位置和变换。
  - 跳过的块中的 M 函数变为活动状态;模态功能被切换。
  - 跳过的块中的 CPL 函数将被完全处理。
  - 如果跳过块，则不能启动异步子例程。
  - 如果测量距离端的块不包含任何已编程的运动，则 NC 将使用此时运动功能激活到此块中测量路径的最后位置。

反应 RE5 的特点（跳过剩余的测量路径）:



在跳过的块之后，测量路径结束块（MOC（））中存在运行运动，即使这样的运动没有明确编程。

将运行运动（包括运动功能）编程到测量路径末端块的绝对安全位置非常有用。必须确保轴可以从测量路径的任何点移动到此位置而不会发生碰撞。



我们建议“在通道重置后”（机床参数 CHAN / Ch [k] / Ini / ChResetState）将测量路径末端（MOC（））合并到状态中，否则，如果发生错误，可能会在复位期间发生运行时错误而块被跳过。

使用测头时的特殊功能:

- 不允许编程 3（两个边）进行边选择。
- 所有测量轴的测量边缘必须配置相同，并对应于 ES 的编程值。
- 测量仅涉及指定其通道轴名称的轴。如果未编程通道轴名称，则会测量所有配置为与测头一起使用的通道轴（机床参数 MEAS/Dr[i]/EnablProbeFunc）。
- 如果测量轴未配置为与测头配合使用（机器参数 MEAS / Dr [i] / EnablProbeFunc），则会生成运行时错误。

## 6. 76 模数计算，用于增量行程无限轴 "Modulo, MOD"

**描述:** 如果循环轴增量行进，模数计算会在控制系统中被抑制。这是为了保证真正按照所定义的进给路径进给。仅为驱动器和显示值的目标值使用永久模态计算。对于控制器中的轴值，仅使用下一绝对位置信息进行模态计算。

若模态轴在相同方向上反复进行增量进给，则该轴会最终达到可显示值的内部极限。此由控制器借助限位开关错误进行确认。

若增量移动的总和超过该极限值，则需以协调方式将内部轴内存复位至模态的范围。因此需要使用“模态”函数。

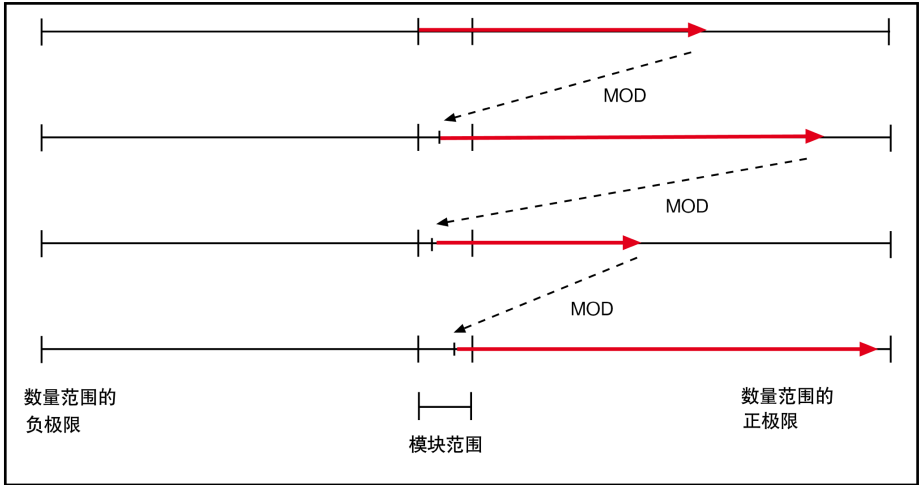


插图 6-16: 模态范围

**语法:** 增量进给循环轴的模态计算:

|           |
|-----------|
| Modulo    |
| 缩写形式: MOD |

表格 6-94: MOD 语法 - 用于单独的无限轴

增量进给循环轴的模态计算:

|                              |                   |
|------------------------------|-------------------|
| Modulo(<AN1>{,<AN1>{, ...}}) |                   |
| 缩写形式: MOD(...)               |                   |
| 其中                           |                   |
| <ANx>                        | 为模轴的轴名（通道或系统轴的名称） |

表格 6-95: MOD 语法 - 用于单独的无限轴

|                                                                                   |                                                                             |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|  | 可在同一块中结合任何进给运动来编程模态命令。模态命令不会使速度突然降低。<br>如果已将模块命令配置为模运行（用于驱动），则该模块命令也可用于回转轴。 |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------|

示例: :  
N030 MOD(A,B) A270 B90 F1000  
:

## 6.77 模数偏置，用于无限轴"ModuloShift, MODS"

描述: 通过指定的旋转次数，可使模块或旋转轴的程序位置与已组态的模块操作的偏移相抵消。

|           |                                                                      |
|-----------|----------------------------------------------------------------------|
| 语法:       | MODS({,ES1 ES2})                                                     |
| 其中        |                                                                      |
| MP1   MP2 | 选择探头 1 或 2 以更改配置                                                     |
| ES1   ES2 | 选择测头的下降（ES1）或上升（ES2）边缘。如果 ES1 和 ES2 都未编程，则 MPEDGE 将接触探头的边缘选择重置回配置设置。 |
| MPEDGE()  |                                                                      |
|           | 将所有测头的边缘重新设置回配置设置。                                                   |

表格 6-96: NC 功能 MPEDGE 的语法

示例: :  
N030 MODS(A1,B-2)  
:

## 6.78 更改测头边缘配置（MPEDGE）

描述: 通过以下机床参数，可定义测头输入的哪个边沿（上升或下降），驱动器应触发单个驱动测量:

MEAS/Dr[i]/Probe1Edge and  
MEAS/Dr[i]/Probe2Edge

该配置可以用 NC 功能 MPEDGE 覆盖。所有使用测头（G75，MOC ...）的功能都会激活新的边沿。

|           |                                                                      |
|-----------|----------------------------------------------------------------------|
| 语法:       | MPEDGE(MP1 MP2{,<ES1 ES2>})                                          |
| 其中        |                                                                      |
| MP1   MP2 | 选择探头 1 或 2 以更改配置                                                     |
| ES1   ES2 | 选择测头的下降（ES1）或上升（ES2）边缘。如果 ES1 和 ES2 都未编程，则 MPEDGE 将接触探头的边缘选择重置回配置设置。 |

使用高级语言语法的 NC 功能

|            |                    |
|------------|--------------------|
| MPEDGE ( ) |                    |
|            | 将所有测头的边缘重新设置回配置设置。 |

表格 6-97: NC 功能 MPEDGE 的语法

|     |                       |                  |
|-----|-----------------------|------------------|
| 示例: | N100 MPEDGE(MP2, ES2) | ;探针 2 对上升沿做出反应   |
|     | ...                   |                  |
|     | N200 MPEDGE(MP1)      | ;探针 1 的边缘返回到配置参数 |
|     | ...                   |                  |
|     | N300 MPEDGE ( )       | ;所有接触探头重置为配置参数   |

- 特殊功能和限制:
- MPEDGE 在块开始处停止（路径速度为零）。
  - 不允许沿 MOC 的测量路径更改配置。
  - 在两个边缘都不可能触发测量。

## 6.79 裁料 "Nibble、NIB"

描述: 开关“裁料”函数。  
若激活裁料，则以下情况中不会释放任何冲程:

- 在由“NUM”（参见第 6.80 章“分隔进给块：路径段数 NUM”第 276 页）或“LEN”函数（参见第 6.67 章“分隔进给块：路径段长度 LEN”第 259 页）生成或编程的每个路径区段的终点，以及
- 第一部分路径的起始位置处（若未在有效平面中编程进给运动或裁料在之前的块中已关闭）。

将总是在冲程完成前启动后续进给运动。



影响冲程释放时间的函数:

- PtDefault  
(参见第 6.91.2 章“将冲程释放时间设置为默认值 PtDefault、PTD”第 299 页)
- PtBlkEnd  
(参见第 6.91.1 章“冲压释放时间（插补终点） PtBlkEnd、PTE”第 297 页)
- PtInpos  
(请参阅第 6.91.3 章“冲程释放时间（Inpos 窗口） PtInpos、PTI”第 300 页) 打开。

|                |                           |      |
|----------------|---------------------------|------|
| 语法:            | Nibble(1) 或<br>步冲         | 裁料打开 |
|                | Nibble(0) 或<br>Nibble ( ) | 裁料关闭 |
| 缩写形式: NIB (..) |                           |      |

表格 6-98: Nibble (NIB) 语法

- 示例:
- G90 处于活动状态（绝对尺寸编程）
  - 激活平面: X/Y
  - 当前位置: X=0, Y=0, C=0



|                                  |                                                                                                 |
|----------------------------------|-------------------------------------------------------------------------------------------------|
| :                                |                                                                                                 |
| N10 LEN=12                       | 使用最大值 12 mm，将正在进给的块分隔为相等的部分路径。<br>必须在打开裁料前对 LEN 进行编程！                                           |
| N20 C10 Nibble(1)<br>G9 (ASHAPE) | 裁料开启。模态 LEN 生效。<br>使用步冲逻辑启用插补器<br>C 轴旋转至 10 度。<br>由于未对 X 和 Y 轴进行编程，所以无冲程。                       |
| N30 X0                           | 由于 X 轴已经位于 0 位置，因此无进给运动。<br>由于 x 轴处于激活的平面内，因此将进行冲程。                                             |
| N40 X110                         | 进给块被分割成 10 个相等的 11 mm 路径区段。在 X11、X22、X33...X99、X110。<br>由于 N30 中无进给运动，因此在位置 X0 进行 <b>附加冲程</b> 。 |
| N50 Y30 NUM=3                    | 覆盖当前块中以模态形式生效的 LEN (N10)。<br>将进给块分割成 3 个相等的路径区段。在 Y10、Y20、Y30 处进行冲程。                            |
| N60 Y90                          | N10 中的 LEN 再次生效。<br>Y42、Y54、Y66、Y78、Y90 处的冲程。                                                   |
| N70 X50 Y50 Nibble()<br>G8       | 裁料关闭<br>路径斜坡开启<br>进给运动至 X=50, Y=50。                                                             |
| :                                |                                                                                                 |

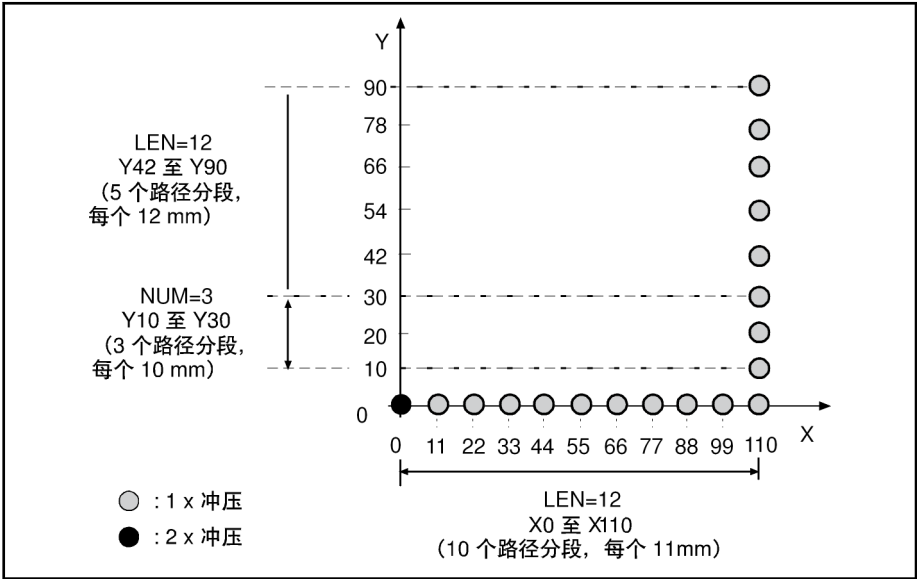


插图 6-17: Nibble (NIB) 示例



要使用步冲功能，必须启用支持 n 步功能的插补器：  
G9(X... , Y... , ...)  
G9 (ASHAPE)  
G9 (SHAPE... , PUN)

特殊功能和限制:

- 必须使用 MP 8001 00010 来启用裁料。
- 必须对“NUM”或“LEN”函数进行编程。控制器由此从已行进的编程距离中自动生成部分路径；将在此类路径段的终点释放冲程。

使用高级语言语法的 NC 功能

- 该函数对同一块中编程的进给运动已有模态效果。
- 如果打开冲孔（参见 第 6.93 章 “冲孔 Punch、PUN” 第 302 页 ），则该函数会取消选定 “裁料”。
- 不包含来自活动平面的轴坐标的块同样不会释放冲程。
- 若 PLC 禁止冲程释放，会在 PCL 释放冲程前将处理保持在冲程释放位置处。

## 6.80 分隔进给块：路径段数 "NUM"

描述： 将编程的进给块分隔成给定数量的等长度路径段。



仅在结合 “冲孔”（请参见第 第 6.93 章 “冲孔 Punch、PUN” 第 302 页 页）和 “裁料”（请参见第 第 6.79 章 “裁料 Nibble、NIB” 第 274 页 页）函数使用时才有效。

语法：

|             |                                             |
|-------------|---------------------------------------------|
| NUM=<value> |                                             |
| 其中          |                                             |
| <Value>     | 路径区段的数量。<br>输入值：整数，大于 0。<br>NUM=1 时不进行任何分隔。 |

表格 6-99: NUM 语法

示例： 参见 第 6.67 章 “分隔进给块：路径段长度 LEN” 第 259 页 中的示例。

特殊功能和限制：

- 若为已分割的圆形块，则一个冲程到另一个冲程的运动始终为线性。
- NUM 仅在编程的块内有效，并且会禁止任何激活的 LEN（参见 第 6.67 章 “分隔进给块：路径段长度 LEN” 第 259 页 ）。
- 仅可在启用 “冲压 / 步冲”（MP 8001 00010）并且其处于活动时对 NUM 进行编程。

## 6.81 定向编程 "O()、ROTAX()", "Phi、Theta、Psi"

### 6.81.1 使用 O(...) 的定向编程概述

可经常使用 O (<parameter list>) 语法在 phi、theta、psi 的位置上对刀具方向进行编程。参数列表可由一个至四个单独参数组成。概述如下表所示：

| 语法             | 描述                                                             |
|----------------|----------------------------------------------------------------|
| O(<β>)         | 刀具绕固定旋转轴旋转角度 β。在同一个块之前，或在同一个块中，必须使用 RotAx 编程旋转轴。<br>用于向量和张量定向  |
| O(<φ>,<θ>)     | 刀具方向位于 phi 和 theta 坐标内。<br>可选地，可编程 phi<φ> theta<θ>。<br>仅用于向量定向 |
| O(<x>,<y>,<z>) | 将刀具方向作为定向向量编程，有三个笛卡尔组件 x，y 和 z。<br>仅用于向量定向                     |

|                                                                                 |                                                                                                                                                                     |
|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $O(\langle \phi \rangle, \langle \vartheta \rangle, \langle \psi \rangle)$      | 刀具方向位于 phi、theta 和 psi 坐标（欧拉角）内。<br>可选地，可编程 $\text{phi} \langle \phi \rangle \text{ theta} \langle \vartheta \rangle \text{ psi} \langle \psi \rangle$ 。<br>仅用于张量定向 |
| $O(\langle w \rangle, \langle x \rangle, \langle y \rangle, \langle z \rangle)$ | 将刀具方向作为四元数 (w, x, y, z) 编程。<br>仅用于张量定向                                                                                                                              |

表格 6-100: O 程序设计概述:

|                                                                                   |                                                                                                  |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
|  | 通过启用类型的第 100 位等于 2 的轴转换，激活向量定向。示例：类型 = 3232201<br>通过启用类型的第 100 位等于 3 的轴转换，激活张量定向。示例：类型 = 3306301 |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|

6. 81. 2 一般描述

以当前程序坐标系（PCS）为基础，在空间内任意处为刀具（铣床、钻头、激光、夹具等）指定方向。有多种类型的定向编程供使用。

|                                                                                   |                                                                            |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------|
|  | 除刀具的方向运动外，若还要移动 TCP，请参见 “《Rexroth IndraMotion MTX 函数说明》” 手册中的 “空间坐标编程” 函数。 |
|-----------------------------------------------------------------------------------|----------------------------------------------------------------------------|

6. 81. 3 使用轴编程的线性方向运动

|     |                                                                                                                                 |
|-----|---------------------------------------------------------------------------------------------------------------------------------|
| 描述: |  仅适用于支持混合线性坐标 / 旋转轴编程的 5 轴转换，例如轴转换类型 3032001。 |
|-----|---------------------------------------------------------------------------------------------------------------------------------|

为了编程刀具定向，需要编程作用于刀具的二个回转轴的位置值（如 B、C）。

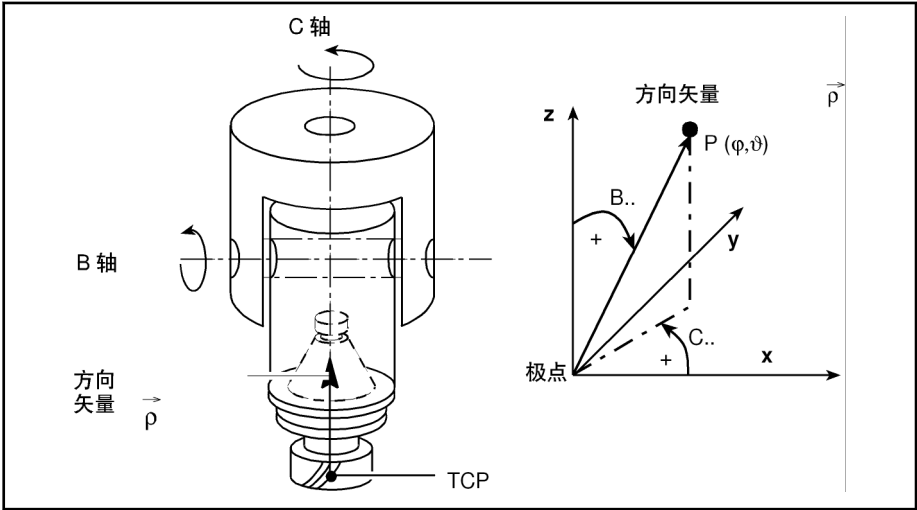


插图 6-18: 使用轴编程的线性方向运动

以下情况适用:

- 仅允许轴运动，对于轴运动，二个回转轴的位置值可在一对一的基础上映射至定向向量的方向坐标（如 phi 和 theta）。
- 在回转轴中会以线性运动的方式执行方向运动。
- 这适用于对称旋转的刀具。

|     |                                                                                       |
|-----|---------------------------------------------------------------------------------------|
| 语法: | 1. 使用 "Coord(...)" 来激活类型为 3032001 的轴变换（参见 ）。(请参阅第 6.28 章 “选择轴转换 Coord、CRD” 第 225 页 )。 |
|-----|---------------------------------------------------------------------------------------|

使用高级语言语法的 NC 功能

2. 使用以下所示语法。

|                                             |                                                                          |
|---------------------------------------------|--------------------------------------------------------------------------|
| {<No>} {<TCP motion>} {<B><Pos>} {<C><Pos>} |                                                                          |
| 其中                                          |                                                                          |
| <No>                                        | 块编号<br>请参阅 第 3.8.2 章 “块编号” 第 34 页 。                                      |
| <TCP mot>                                   | 其他 TCP 运动<br>关于此语法，请参见 ”《Rexroth IndraMotion MTX 函数说明》” 中有关 “空间坐标编程”的章节。 |
| <B>、<C>                                     | 刀具中回转轴的轴地址                                                               |
| <Pos>                                       | 绝对轴位置（单位：度）                                                              |

表格 6-101: “使用轴编程的线性定向运动” 语法

示例：

设置示例：

参数 TRA/AxTrafo/TrafoType (MP 1030 00110) 适用于轴变换 2: 3032001

参数 CHAN/Ch/Coord/Wcs/LinCoord [1] (MP 7080 00010 [1]): x

参数 CHAN/Ch/Coord/Wcs/LinCoord [2] (MP 7080 00010 [2]): y

参数 CHAN/Ch/Coord/Wcs/LinCoord [3] (MP 7080 00010 [3]): z

|                                  |                                                 |
|----------------------------------|-------------------------------------------------|
| N10 G1 X0 Y0 Z0 B0 C0            | 编程逻辑 / 物理轴名称                                    |
| N20 Coord(2)                     | 激活轴变换 2<br>现在已能编程线性坐标<br>x、y、z 以及方向坐标<br>B 和 C。 |
| N30 x100 y200 z300<br>B20 C60    | 使用其他定向运动的线性坐标插补                                 |
| N40 G2<br>x..y..z..I..J..B20 C60 | 带有附加定向运动的 TCP 螺旋运动                              |
| N50 G1 B20 C10                   | 纯方向运动<br>TCP 保持常量。                              |
| N60 coord(0)<br>:                | 轴变换关闭                                           |

- 特殊功能和限制：
- 活动轴转换类型 3032001 必须处于活动状态。
  - 特殊路径搜索逻辑会避免大于 180 度的回转轴旋转。

6. 81. 4 使用坐标编程的线性方向运动

描述：

 适用于类型的第 100 位等于 1 的所有 5 轴转换，例如类型 = 3232101。

使用方向向量的方向坐标（例如，phi 和 theta）实现刀具定向编程。

-

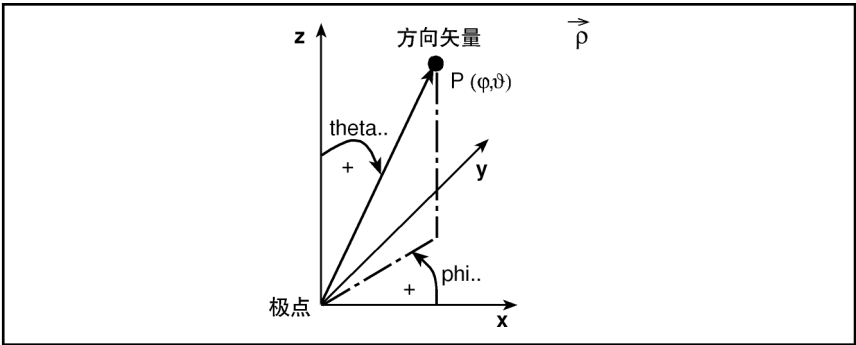


插图 6-19: 定向向量的方向坐标

- 这适用于对称旋转的刀具。
  - 在  $\phi$  和  $\vartheta$  中以线性插补的方式实现方向运动，即在假想的  $\phi/\vartheta$  平面中以直线的方式进行。
- 语法:
1. 使用 “Coord...” 启用轴转换（例如，类型 = **3232101**）（请参阅 第 6.28 章 “选择轴转换 Coord、CRD” 第 225 页）。
  2. 使用以下所示语法。

|                                         |                                                                                      |
|-----------------------------------------|--------------------------------------------------------------------------------------|
| {<No>}{<TCP mot>}{<phi><Φ>}{<theta><Θ>} |                                                                                      |
| 其中                                      |                                                                                      |
| <No>                                    | 块编号<br>参见 第 3.8.2 章 “块编号” 第 34 页                                                     |
| <TCP mot>                               | 其他 TCP 运动<br>关于此语法，请参见 “《Rexroth IndraMotion MTX 函数说明》” 中有关 “空间坐标编程” 的章节。            |
| <phi>                                   | 在参数 CHAN / Ch / Coord / Wcs / OriCoord [1] (MP 7080 00010 [4]) 中输入角度名称。<br>默认: phi   |
| <theta>                                 | 在参数 CHAN / Ch / Coord / Wcs / OriCoord [2] (MP 7080 00010 [5]) 中输入角度名称。<br>默认: theta |
| <Φ>, <Θ>                                | 绝对角度值（单位: 度）                                                                         |

表格 6-102: “使用轴编程的线性方向运动” 语法

示例:

设置示例:

参数 TRA/AxTrafo/TrafoType (MP 1030 00110) 适用于轴变换 2: 3232101

参数 CHAN/Ch/Coord/Wcs/LinCoord [1] (MP 7080 00010 [1]): x

参数 CHAN/Ch/Coord/Wcs/LinCoord [2] (MP 7080 00010 [2]): y

参数 CHAN/Ch/Coord/Wcs/LinCoord [3] (MP 7080 00010 [3]): z

参数 CHAN/Ch/Coord/Wcs/OriCoord [1] (MP 7080 00010 [4]): phi

参数 CHAN/Ch/Coord/Wcs/OriCoord [2] (MP 7080 00010 [5]): theta

|              |                                      |
|--------------|--------------------------------------|
| N10 G1 X0 Y0 | 编程通道或系统轴名称                           |
| Z0 B0 C0     |                                      |
| N20 Coord(2) | 激活轴变换 2                              |
|              | 现已可对线性坐标 x、y、z 和方向坐标 phi、theta 进行编程。 |

使用高级语言语法的 NC 功能

```
N30 x1 y2 z3 使用其他定向运动的线性坐标插补
phi5 theta5

N40 coord(0) 轴变换关闭
```

- 特殊功能和限制:
- 要求轴转换类型的第 100 位 = 1。
  - 特殊路径搜索逻辑会避免大于 180 度的回转轴旋转。

6. 81. 5 矢量定向

描述: 可通过以下方式对刀具方向编程

- 定向向量的定向坐标  
(如 phi 和 theta)。

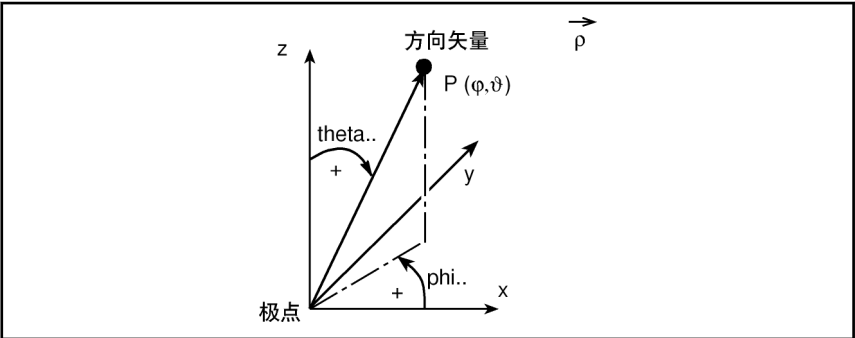


插图 6-20: 定向向量的方向坐标

- 带极角 (φ, θ) 的 "O(..)" 函数，或定向向量的卡笛尔分量 (ρ<sub>x</sub>、ρ<sub>y</sub>、ρ<sub>z</sub>)。

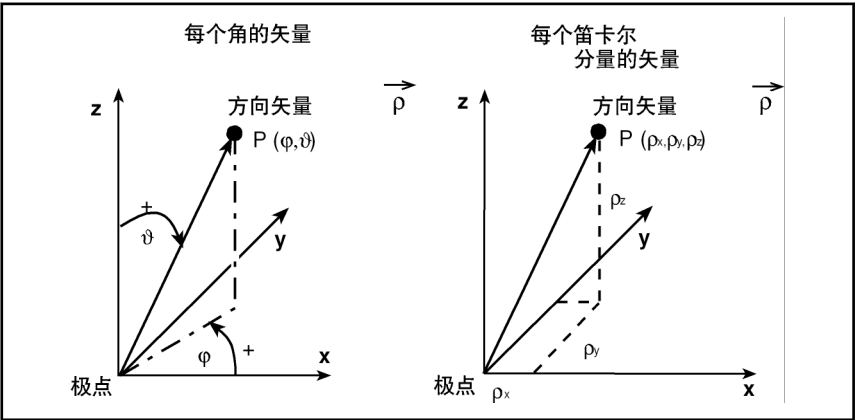


插图 6-21: 带极角 (φ, θ) 的 O(..) 函数，或定向向量的卡笛尔分量 (ρ<sub>x</sub>、ρ<sub>y</sub>、ρ<sub>z</sub>)。

- "ROTAX(..)" 函数用于定义空间中任意位置的轴旋转，从而可使用 "O(..)" 函数让方向张量按特定旋转角度 β 旋转。  
可通过极角 (φ<sub>u</sub>, u) 和笛卡尔分量 (u<sub>x</sub>, u<sub>y</sub>, u<sub>z</sub>) 对回转轴进行编程。

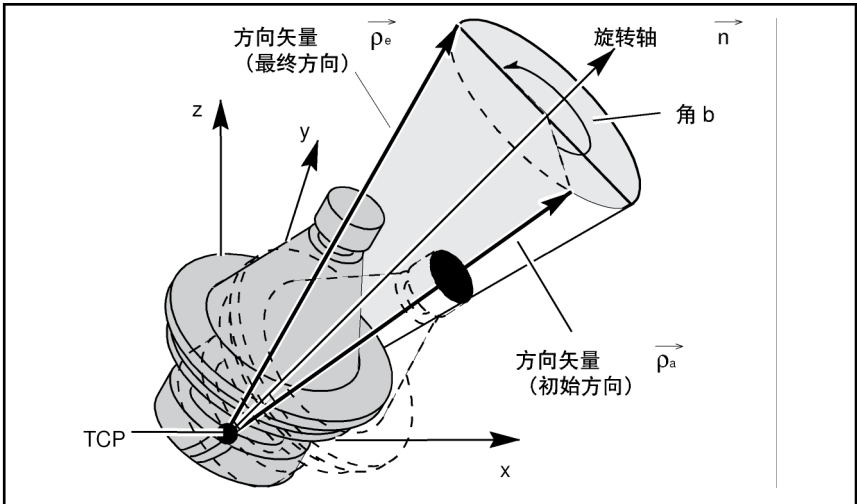


插图 6-22: 可定义空间中任意处定向回转轴的 ROTAX(.) 函数  
以下情况适用:

- 仅可使用第 100 个位置，类型 = 2 的轴变换来编程向量方向。
- 可从编程的初始方向到结束方向，将方向运动作为定向向量的旋转指针运动执行。
- 这适用于对称旋转的刀具。

语法:

1. 用 "Coord(...)"激活允许的轴变换类型（类型是第 100 个位置 = 2）。（请参阅 第 6.28 章 "选择轴转换 Coord、CRD" 第 225 页 ）。
2. 使用以下所示语法。

|                                                                                                                                                                                                                                                                     |                                                                                                |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| {<No>}{<TCP motion>}{<phi><phi>}{<theta><theta>} 或<br>{<No>}{<TCP motion>}O(<phi>,<theta>) 或<br>{<No>}{<TCP motion>}O(<rho x>,<rho y>,<rho z>) 或<br>{<No>}{<TCP motion>}ROTAX(<phi u>,<theta u>) O(<beta>) 或<br>{<No>}{<TCP motion>}ROTAX(<ux>,<uy>,<uz>) O(<beta>) |                                                                                                |
| 其中                                                                                                                                                                                                                                                                  |                                                                                                |
| <No>                                                                                                                                                                                                                                                                | 块编号<br>请参阅 第 3.8.2 章 "块编号" 第 34 页 。                                                            |
| <TCP mot>                                                                                                                                                                                                                                                           | 其他 TCP 运动<br>关于此语法，请参见 "《Rexroth IndraMotion MTX 函数说明》" 中有关 "空间坐标编程" 的章节。                      |
| <phi>                                                                                                                                                                                                                                                               | MP 7080 00010[4] 中输入的角名称。<br>默认: phi                                                           |
| <theta>                                                                                                                                                                                                                                                             | MP 7080 00010[5] 中输入的角名称。<br>默认: theta                                                         |
| <phi>、<theta>、<phi u>、<theta u>                                                                                                                                                                                                                                     | 绝对角度值（单位：度）。<br>取值范围: 0° ≤ phi < 360° ; 0° ≤ theta ≤ 180° .<br>若将 theta 编程为其他的数值，它会自动地转换为给定区间。 |
| O(<phi>,<theta>)                                                                                                                                                                                                                                                    | 使用定向向量的函数 O(.)，以及极角 <phi> 和 <theta> 进行定向。                                                      |





## 使用高级语言语法的 NC 功能

- 可从编程的初始方向到结束方向，将方向运动作为张量定向的旋转指针运动执行。
- 由于 TCS 在此情况下与刀具永久耦合，因此这仅适合 **非** 对称旋转的刀具。

- 语法:**
1. 用 "Coord(...)" 激活允许的轴变换类型（类型是第 100 个位置 = 3）。  
(请参阅 第 6.28 章 "选择轴转换 Coord、CRD" 第 225 页)。
  2. 使用以下所示语法。

|                                                                                             |                                                                                                        |
|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| {<No>} {<TCP>} {<phi><Φ>} {<theta><Θ>} {<psi><Ψ>} 或                                         |                                                                                                        |
| {<No>} {<TCP>} {0(<Φ>, <Θ>, <Ψ>)} or                                                        |                                                                                                        |
| {<No>} {<TCP>} O <sub>x</sub> (<Φ <sub>xxy</sub> (<Φ <sub>yy</sub>                          |                                                                                                        |
| {<No>} {<TCP>} O <sub>x</sub> (<Φ <sub>xxz</sub> (<Φ <sub>zz</sub>                          |                                                                                                        |
| {<No>} {<TCP>} O <sub>y</sub> (<Φ <sub>yyz</sub> (<Φ <sub>zz</sub>                          |                                                                                                        |
| {<No>} {<TCP>} O <sub>x</sub> (<011>, <021>, <031>) O <sub>y</sub> (<012>, <022>, <032>)    |                                                                                                        |
| {<No>} {<TCP>} O <sub>x</sub> (<011>, <021>, <031> ) O <sub>z</sub> (<013>, <023>, <033>) 或 |                                                                                                        |
| {<No>} {<TCP>} O <sub>y</sub> (<012>, <022>, <032>) O <sub>z</sub> (<013>, <023>, <033>) 或  |                                                                                                        |
| {<No>} {<TCP>} 0(<w>, <x>, <y>, <z>) or                                                     |                                                                                                        |
| {<No>} {<TCP>} ROTAX (<Φ <sub>uu</sub>                                                      |                                                                                                        |
| {<No>} {<TCP>} ROTAX (<u <sub>xyz</sub>                                                     |                                                                                                        |
| 其中                                                                                          |                                                                                                        |
| <No>                                                                                        | 块编号<br>请参阅 第 3.8.2 章 “块编号” 第 34 页 。                                                                    |
| <TCP mot>                                                                                   | 其他 TCP 运动<br>关于此语法，请参见 ”《Rexroth IndraMotion MTX 函数说明》” 中有关 “空间坐标编程” 的章节。                              |
| <phi>                                                                                       | 在参数 CHAN / Ch / Coord / Wcs / OriCoord [1]<br>(MP 7080 00010 [4]) 中输入角度名称。<br>默认: phi                  |
| <theta>                                                                                     | 在参数 CHAN / Ch / Coord / Wcs / OriCoord [2]<br>(MP 7080 00010 [5]) 中输入角度名称。<br>默认: theta                |
| <psi>                                                                                       | 在参数 CHAN / Ch / Coord / Wcs / OriCoord [3]<br>(MP 7080 00010 [6]) 中输入角度名称。<br>默认: psi                  |
| <Φ>, <Θ>, <Ψ>                                                                               | 绝对欧拉角（单位：度）。<br>取值范围：<br>0° ≤ Φ < 360° ；<br>0° ≤ Θ ≤ 180° ；<br>0° ≤ Ψ ≤ 360° .<br>此范围以外的值将被自动转换至相应的间隔。 |

使用高级语言语法的 NC 功能

|                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\langle \phi_{..} \rangle, \langle \vartheta_{..} \rangle$                                                                                                                                                                                                                                                                                                                                                    | 向量绝对角度值（单位：度）<br>取值范围：<br>$0^\circ \leq \phi_{..} < 360^\circ$ ;<br>$0^\circ \leq \vartheta_{..} \leq 180^\circ$ .<br>若将 $\vartheta$ 编程为值域以外的数值，它会自动地转换为给定区间。                                                                                                                |
| $O_x(\langle 011 \rangle, \langle 021 \rangle, \langle 031 \rangle)$<br>$O_y(\langle 012 \rangle, \langle 022 \rangle, \langle 032 \rangle)$<br>$O_z(\langle 013 \rangle, \langle 023 \rangle, \langle 033 \rangle)$<br>$O_x(\langle \phi_x \rangle, \langle \vartheta_x \rangle)$<br>$O_y(\langle \phi_y \rangle, \langle \vartheta_y \rangle)$<br>$O_z(\langle \phi_z \rangle, \langle \vartheta_z \rangle)$ | 使用函数 $O_x(..)$ 、 $O_y(..)$ 、 $O_z(..)$ 进行定向<br>$O_x(..)$ 可定义 TCS 的 x 坐标在参考坐标系中的方向（张量方向的第一列）。这同样适用于 $O_y(..)$ 和 $O_z(..)$ 。<br>可使用对应极角 $\langle \phi_{..} \rangle$ 以及 $\langle \vartheta_{..} \rangle$ ，或使用 TCS p 定向张量列向量的笛卡尔分量来指定方向。<br>只允许绝对尺寸。<br>列向量 (o..) 的分量值被自动标准化为 1。 |
| $\langle w \rangle, \langle x \rangle, \langle y \rangle, \langle z \rangle$                                                                                                                                                                                                                                                                                                                                   | 方向四元数的四个组件                                                                                                                                                                                                                                                                   |
| $ROTAX(\langle \phi_u \rangle, \langle \vartheta_u \rangle)$                                                                                                                                                                                                                                                                                                                                                   | 使用极角 ( $\phi_u$ 、 $\vartheta_u$ ) 定义回转轴。                                                                                                                                                                                                                                     |
| $ROTAX(\langle u_x \rangle, \langle u_y \rangle, \langle u_z \rangle)$                                                                                                                                                                                                                                                                                                                                         | 使用绝对尺寸中的笛卡尔分量 $\langle u_x \rangle$ 、 $\langle u_y \rangle$ 、 $\langle u_z \rangle$ 定义回转轴。自动标准化为 1。                                                                                                                                                                          |
| $O(\langle \beta \rangle)$                                                                                                                                                                                                                                                                                                                                                                                     | 声明增量角度 $\langle \beta \rangle$ （单位：度），定向张量将按照该角度围绕回转轴旋转。<br>允许大于 360 度的值。<br>可以使用正 / 负符号来选择旋转方向。                                                                                                                                                                             |

表格 6-104:      *Tensor orientation 语法*

**示例:**      设置示例:

参数 TRA/AxTrafo/TrafoType (MP 1030 00110) 适用于轴变换 2: 3333301

参数 CHAN/Ch/Coord/Wcs/LinCoord [1] (MP 7080 00010 [1]): x

参数 CHAN/Ch/Coord/Wcs/LinCoord [2] (MP 7080 00010 [2]): y

参数 CHAN/Ch/Coord/Wcs/LinCoord [3] (MP 7080 00010 [3]): z

参数 CHAN/Ch/Coord/Wcs/OriCoord [1] (MP 7080 00010 [4]): phi

参数 CHAN/Ch/Coord/Wcs/OriCoord [2] (MP 7080 00010 [5]): theta

参数 CHAN/Ch/Coord/Wcs/OriCoord [3] (MP 7080 00010 [6]): psi

N100 G1 X10 Y20 Z30  $O_x(1,0,0)$   $O_y(0,0.707,-0.707)$

- 或 -

N100 G1 X10 Y20 Z30  $O_x(1,0,0)$   $O_z(0,0.707,0.707)$

- 或 -

N100 G1 X10 Y20 Z30  $O_y(0,0.707,-0.707)$   $O_z(0,0.707,0.707)$

- 或 -

N100 G1 X10 Y20 Z30  $O(0.9239,0.3827,0,0)$

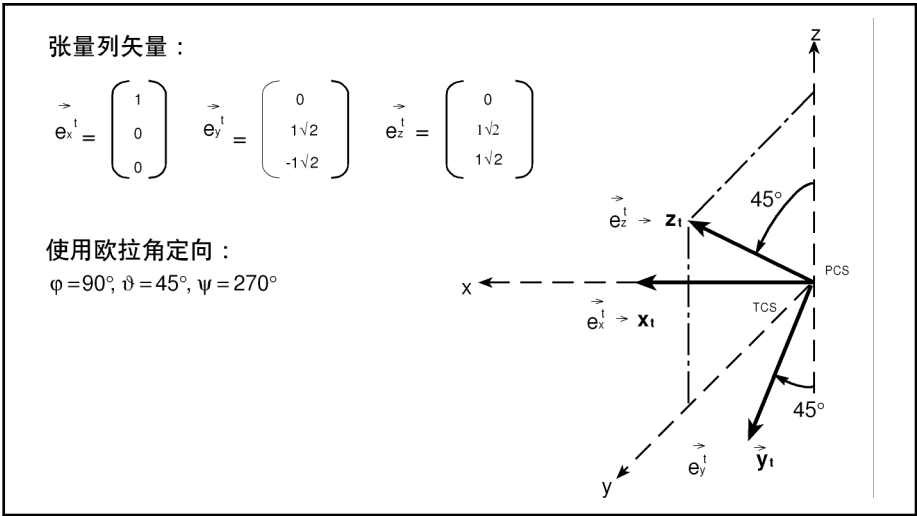


插图 6-24: 张量定向的示例

- 特殊功能和限制：
- 除非已经编程了 ROTAX(..)，否则将始终在朝向最终方向的最短路径上发生方向运动。  
对于 ROTAX(..)，旋转的方向取决于  $\langle \beta \rangle$  的正负号。
  - 要生成唯一的定向张量，需要以下限制：  
适用  $\vartheta = 0^\circ$ ：  
要指定方向， $\phi$  和  $\psi$  的和是必需的。  
适用  $\vartheta = 180^\circ$ ：  
要指定方向， $\phi$  和  $\psi$  的差值是必需的。
  - 若因编程张量列而出现两个平行或逆平行的列向量，则无法计算定向张量。  
显示运行时错误。

6.82 进给 100% "OvrDis、OVD", "OvrEna、OVE"

描述： 程序控制器会影响进给模式和快速进给的通道进给电位计的效果 (qCh\_Override\_xy)。  
这些函数在“手动数据输入”和“处理”操作模式中有效。  
异步轴不受“OVD/OVE”函数影响。

|     |          |                                        |
|-----|----------|----------------------------------------|
| 语法： | OvrDis   | 进给电位计关闭<br>无需考虑进给电位计，进给均设置为编程数值的 100%。 |
|     | 缩写形式：OVD |                                        |
|     | OvrEna   | 进给电位计开启<br>进给取决于进给电位计的位置。              |
|     | 缩写形式：OVE |                                        |

表格 6-105: OvrDis (OVD) 和 OvrEna (OVE) 的语法  
以下情况适用：

- 两个函数均为模态，并相互取消选定。
- 接通状态可在机床参数中进行指定。
- 两个函数可在同一块中结合其他路径条件写入。

使用高级语言语法的 NC 功能

- 使用接口信号 "qAx\_OVERRIDE\_100"可通过特定轴的方式影响异步轴覆盖的效果。

示例:

```
: 进给电位计设为 100%

N40 OVD G1 X5 Z-2 F200 S100 M4

 取消进给电位计

: 将进给电位计切换至 40%，进给保持在 100%

N80 OVE X100 激活进给电位计
Y50 进给更改为 40%
```

6. 83 更改最大路径加速度 "PathAcc、PAC"

描述: 在零件程序中减少下面两项的上限

- 路径加速度以及
- 路径减速度。

可使用机床参数预设加速度和减速度的值，且可在零件程序中共同或分别进行切换。

 编程或预设的路径加速度可由路径中相关轴允许的最大轴过冲限制。

语法:

|                                          |                                                                   |
|------------------------------------------|-------------------------------------------------------------------|
| PathAcc<br>(ACC<value>)                  | 同时设置路径的加速度和减速度。                                                   |
| PathAcc({UP<value1>,<br>{DOWN<value2>}}) | 路径加速度和减速度的分别设置。                                                   |
| PathAcc() 或<br>PathAcc(0)                | 从 MP 7030 00210 和 7030 00220 重新激活加速度值。                            |
| 缩写形式: PAC(..)                            |                                                                   |
| 其中                                       |                                                                   |
| <Value>                                  | 加速度值<br>依据激活的测量单位 (G71/G70)，控制器会将编程数据解释为 "1000 Inch/s²" 或 "m/s²"。 |
| <value1>                                 | 路径加速度的值<br>其他所有方面与 <value> 一致                                     |
| <value2>                                 | 路径减速度的值。<br>其他所有方面与 <value> 一致                                    |

表格 6-106: PathAcc (PAC) 语法

示例:

```
N30 G71 激活公制编程

N40 PAC (UP1.5) 将路径加速度设置为 1.5 m/s²

:

N140 PAC (ACC5) 将路径的加速度和减速度设置为 5 m/s²

:
```

N200 PAC (UP3.5,DOWN2) 将路径加速度设置为 3.5 m/s<sup>2</sup>，路径减速度设置为 2 m/s<sup>2</sup>  
:  
N240 PAC () 将加速度值复位为机床参数设置。  
:

- 特殊功能和限制:
- 加速度的最大可编程值会总是由机床参数的值限制。
  - 我们建议在单个块中对函数进行编程。

## 6.84 启用定位表 "PmTSe1、PMS"

描述: 启用定位表（“斜面”定位修正表）。这类修正表以 XML 文件形式保存于控制器的文件系统中。

语法:

|                             |                                                                                                               |
|-----------------------------|---------------------------------------------------------------------------------------------------------------|
| PmTSe1 ({<path>}<FileName>) |                                                                                                               |
| 缩写形式: PMS (..)              |                                                                                                               |
| 其中                          |                                                                                                               |
| <Path>                      | 有关 <FileName> 存储目录的可选路径指定。<br>如果没有规范, 则在 “/数据库” 路径中进行搜索。<br>如果找不到 <FileName>, 则为了能在其它目录中找到它, 控制器将使用搜索路径搜索子例程。 |
| <FileName>                  | 定位表的文件名, 包括文件扩展名<br>带有标准名称 (PM<Number>.pmt) 的表可直接通过该编号激活, 例如使用 PmTSe1 (3) 激活表 PM3.pmt。                        |

表格 6-107: PmTSe1 (PMS) 语法



请参见“控制器操作说明”，了解创建与编辑定位表的相关信息。

## 6.85 极坐标编程

### 6.85.1 定义极点 "PolarPol、POP"

- 描述:
- 定义极坐标系的原点, 该原点参考活动程序坐标系的活动平面。
  - 将极坐标编程的极角 1 (默认地址: A; 可在 MP 8005 00001 中定义) 设置为 0 度。
  - 将极坐标编程的极角 2 (默认地址: A; 可在 MP 8005 00002 中定义) 设置为 90 度。

语法:

|                                |               |
|--------------------------------|---------------|
| PolarPol (<PCvalue>,<SCvalue>) |               |
|                                | 指定极点          |
| PolarPol (ACTPOS)              | 将当前位置设置为极点    |
| PolarPol () 或 PolarPol (0)     | 将坐标 0.0 设置为极点 |
| 缩写形式: POP (..)                 |               |
| 其中                             |               |

使用高级语言语法的 NC 功能

|           |          |
|-----------|----------|
| <PCvalue> | 主坐标的极点值  |
| <SCvalue> | 辅助坐标的极点值 |

表格 6-108: PolarPol (POP) 语法

示例:

```
:
N10 G18 通过将 Z 作为主要坐标, X 作为辅助坐标, 切换至 ZX 平面。
:
N30 G0 X10 Z25 快速进给至笛卡尔坐标 X10 和 Z25 的定位运动。
:
N40 POP (ACTPOS) 定义极点时使用当前位置值 (PCvalue=25, SCvalue=10)。
:
```

特殊功能和限制:

- 若未使用 “PolarPol”，NC 会将激活的程序坐标系原点用作极点。
- 编程的极点直到下次切换平面才会为当前平面激活（参见 第 5.2.11 章 “平面切换 G17、G18、G19” 第 120 页）。

6.85.2 极坐标编程

描述:

IndraMotion MTX 允许对平面（2D）和空间（3D）中的极坐标进行编程。

使用以下各项输入所需的终点

- 半径值（极点至目标点 P 的距离），
- 极角 1，以及
- 极角 2。

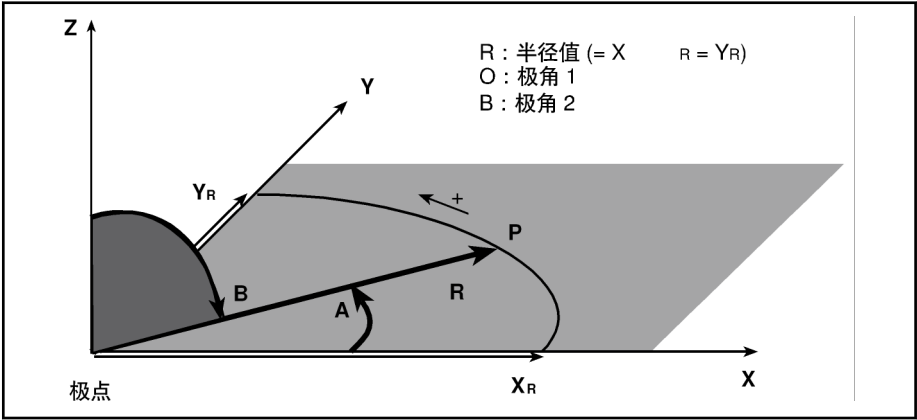


插图 6-25: 示例: 平面中使用激活的 G17 的极坐标

半径值:

会总是使用轴地址来编程当前的工作平面（如处于活动状态 G17 的 “X” 或 “Y”）。该编程的坐标也称为半径坐标。

在绝对尺寸编程的情况下，它表示从极点至目标点的距离；若为增量式编程，则表示从起始点至目标点的距离。

若在同一块内对二个可能的轴地址进行编程，则会发生运行时错误。

极角 1:

- 默认情况下会将此作为地址 “A” 进行编程，且其始终位于当前工作平面内。
- 各个编程的极角 1 涉及 上次编程的 半径坐标轴。正值是指数学上积极意义上的旋转。它在编程新值前保持激活。

如果增量式编程处于活动状态，则会以增量的方式计算编程的角度。

- 使用 POL 激活极坐标编程时，极角 1 的值设为 0 度。
- 极角 2:
- 若绝对尺寸编程处于活动状态，在默认情况下会将其作为地址 “B”进行编程，并且其描述的是（当前工作平面中的）垂直轴与当前极点至目标点向量之间的角度。它在编程新的值前保持有效。  
如果增量式编程处于活动状态，则会以增量的方式计算编程的角度。
  - 使用 POL 激活极坐标编程时，极角 2 的值设为 90 度。  
目标点在此情况下会精度地位于激活的工作平面中。此时的极坐标仅在该平面中有效。工作平面以外所有的其他坐标均按照笛卡尔坐标解释。这一点特别适用于当前垂直轴的编程坐标值！
  - 若极角 2 不等于 90 度，则会生成“实际”的空间极坐标（球面坐标）。编程当前垂直轴的坐标值会导致运行时错误。但编程中可同时包含 3D 空间外的所有坐标。与 2D 的情况类似，同样会按照笛卡尔坐标解释这些坐标。
  - 若极角 2 为 0 度，则极角 1 的值将不再重要，这是由于编程点位于极点上方精度垂直的位置。

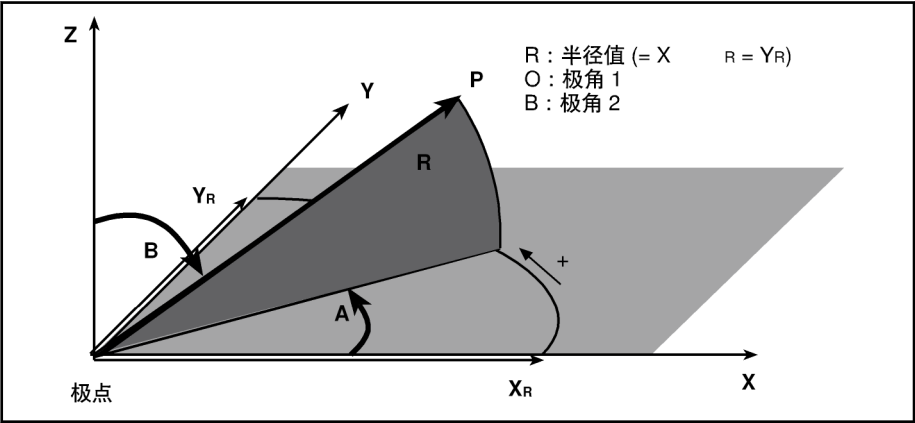


插图 6-26: 示例：空间中使用激活的 G17 的极坐标

 可使用机床参数 8005 0001 和 8005 0002 来更改极角 1 和极角 2 的默认地址。因此它们可能在您的系统中有所不同。因此，它们在您的系统中可能会有所不同。

示例：绝对尺寸极坐标编程

|                |                                                                 |
|----------------|-----------------------------------------------------------------|
| N10 G18        | 平面切换                                                            |
| :              | 此处：通过将 Z 作为主要坐标，X 作为辅助坐标，切换至 ZX 平面。                             |
| N40 POP(25,10) | 在当前平面中的极点定义（这里：用笛卡尔坐标 Z = 25 和 X = 10）。                         |
| :              |                                                                 |
| N50 G1(POL)    | 极坐标中，按照 P2 的路径运动（见图）：                                           |
| Z20 A70        | P2 由从极点开始的绝对距离定义（此处：20 mm）和极角 1（此处：A = 70 度），并基于编程的半径轴（此处：Z-轴）。 |
| :              |                                                                 |

使用高级语言语法的 NC 功能

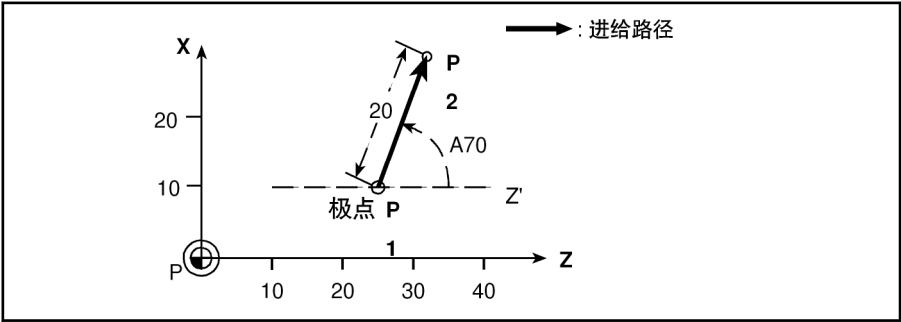


插图 6-27:      极坐标中至 P2 的路径运动

N60 G1 (POL)      极坐标中，按照 P3 的路径运动（见图）：  
X20 A90      P3 由从极点开始的绝对距离定义（此处：20 mm）和极角 1  
:      （此处：A = 90 度），并基于编程的半径轴（此处：X 轴  
:      也可使用以下编程方式：  
:      N60 G1 (POL) Z20 A180  
:      或  
:      N60 Z20 A180

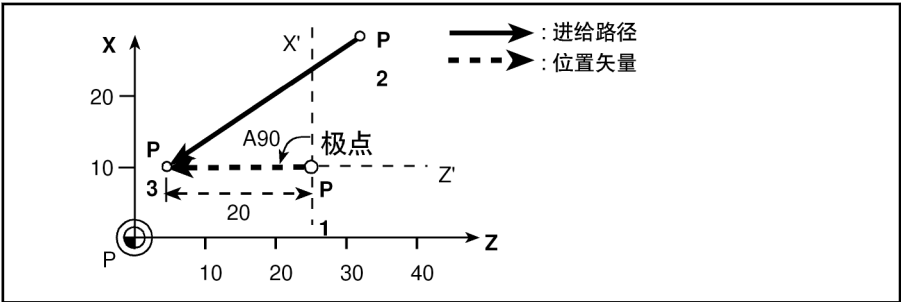


插图 6-28:      极坐标中按照 P3 的路径运动

增量尺寸极坐标编程

N10 G17 G90 G0 X0 Y0      起始位置：P0  
N20 G1 (POL) X200 A0      P1  
N30 G91 Y200 A30      P2  
N40 A60      P3  
N50 A60      P4  
N60 A60      P5  
N70 A60      P6  
N80 A60      P7  
N90 X0      P0



N100 M30

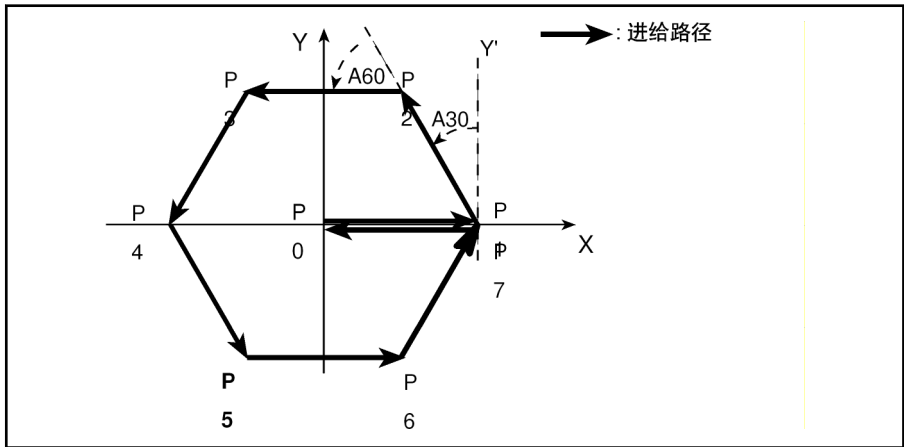


插图 6-29: 增量尺寸中的极坐标编程

## 6.86 定义镜像点 / 旋转点 "PoleSet、PLS"

**描述:** 该函数可指定镜像、缩放和旋转输入工具的极点绝对位置。所定义的极点将在之后成为输入工具的参考点，从而提供镜像/旋转点。



若使用程序零点作为镜像或旋转的参考，则该函数并非必需。

以下情况适用:

- PoleSet 以模态形式生效。极点的位置涉及当前的程序坐标系，该位置在复位为程序坐标原点或重新定义前保持有效。
- 该函数不会引起任何轴进给。
- 可在同一块中将该函数作为其他路径条件和输入工具写入。
- 编程的极点涉及函数“Mirror”、“Scale”和“Rotate”。

|     |                          |                                              |
|-----|--------------------------|----------------------------------------------|
| 语法: | PoleSet (<Coordinates>)  | 将极点设置为给定的坐标。使用逗号分隔给定的坐标（如: PoleSet (X5,Y2)）。 |
|     | PoleSet (0) 或 PoleSet () | 将极点复位至程序坐标系的原点。                              |
|     | 缩写形式: PLS (..)           |                                              |

表格 6-109: PoleSet (PLS) 语法

**示例:**

```
:
N30 PLS (X5,Y2) 将镜像 / 旋转点的位置设置为 X=5 以及 Y=2。
:
N130 PLS () 镜像 / 旋转点复位至程序坐标原点。
:
```

**特殊功能和限制:** 必须以绝对的形式（即根据当前程序零点）输入极点的位置。

## 6.87 与位置相关的高速输出 "PosDepHSOut、PHS"

**描述:** 使用编程的进给运动来影响高速（HS）输出的状态。

使用高级语言语法的 NC 功能

只要 **目标 NC 位置**（涉及当前块的起始点或终点）达到可编程值，即可根据编程来设置或复位 HS 输出。  
在指定事件发生后，可编程输出保持已设定状态的时间。

 使用 MP 4075 00102 设置可用的 HS 输出的参数。

语法:

|                                                    |                                                                                                                                                                                                                                                                              |
|----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PosDepHSOut (<Mode>{, {<distance>} {,<duration>}}) |                                                                                                                                                                                                                                                                              |
| 缩写形式: PHS (...)                                    |                                                                                                                                                                                                                                                                              |
| 其中                                                 |                                                                                                                                                                                                                                                                              |
| <Mode>                                             | 函数运行所需的方法<br>输入值有: 0, 1 或 -1<br><b>0:</b><br>以模态方式将编程的数据保存为 <distance> 和/或 <duration>。当前 HS 输出的状态不会更改。<br><b>1:</b><br>以模态方式将编程的数据保存为 <distance> 和/或 <duration>, 并设置 HS 输出。相关的模态数据值适用于尚未编程的参数。<br><b>-1:</b><br>复位 HS 输出。若已编程 <distance>, 则仅对该块有局部效果。若已编程 <duration>, 则将其忽略。 |
| <Distance>                                         | 到块起始点/终点的距离 (单位: mm 或英寸), 在该距离后要设置信号。<br>默认值: 0<br>0: 位于块终止点<br>大于 0: 与块起点的距离<br>小于 0: 与块终点的距离                                                                                                                                                                               |
| <Duration>                                         | HS 输出的最大开启时长 (单位为 ms)<br>取值范围: 0.5 ... 10000.0<br>编程的值在内部被舍为下一较大的整数, 该整数为 NC 循环时间的倍数。                                                                                                                                                                                        |

表格 6-110:      Syntax PosDepHSOut (PHS)

示例:

```
:
N05 G71
N10 PHS (0,-1.2,40) 配置函数
: 在使用 PHS (1) 语法的后续调用中, 在到达终点前
: 1.2 mm 处, 将 HS 输出开启 40 ms。
N210 G1 G91 F1000
N220 X10 Y23 PHS (1) 根据 N10 中的配置来设置 HS 输出
:
N330 X10 PHS (1,0.1,900) 在起始点位置后的 0.1 mm 处, 最多将 HS 输出开启
: 900 ms
N350 X20 PHS (-1,-0.3) 在终点前 0.3 mm 处复位 HS 输出
:
```

- 特殊功能和限制:
- 每个通道均支持一次 HS 输出。
  - 只有在同一块中编程的函数才对进给运动有效果。
  - 只有在打开时间过后才会复位已设置的 HS 输出。在此时复位控制器不会产生影响。
  - 从数控块的角度来说，<Distance> 始终涉及当前目标位置。  
此处不考虑系统相关的延迟（如由驱动器函数或轴惯性运转引起的延迟）。
  - 若在打开时间中有新的 PHS 作业到来，NC 将删除仍然激活的作业，并执行新的作业。



在两个后续 PHS 作业的情况下，如果 HS 输出时信号未发生更改，则外部硬件通常会无法识别新的作业。

## 6.88 循环轴的定位类型 "PosMode、PMD", 循环轴的局部定位类型 "DC", "ACP", "ACN"

- 描述:
- 指定“循环”轴在定位过程中的旋转方向。
- 以下情况适用:
- 与“PosMode”相反，DC(...)、ACP(...) 以及 ACN(...) 有块到块的作用，在当前块中覆盖相应编程轴的活动定位类型。
  - 结合 DC、ACP 和 ACN 函数编程的位置信息将始终解释为绝对位置值。

语法:

|                             |                                                                                        |
|-----------------------------|----------------------------------------------------------------------------------------|
| PosMode (<Addr><Type>)      | 使用根据 <Type> 的地址 <Addr> 激活轴的定位类型。可在括号中编程数个轴。根据 MP 1003 00005，在当前通道中激活非编程的旋转 / 模态轴的定位类型。 |
| PosMode () 或<br>PosMode (0) | 根据 MP 1003 00005，在当前通道中激活所有轴的定位类型。                                                     |
| 缩写形式: PMD (..)              |                                                                                        |
| <Addr>=DC (<value>)         | 轴 <Addr> 沿着最短路径接近绝对位置 <value>。                                                         |
| <Addr>=ACP (<value>)        | 轴 <Addr> 沿着数学上的正向接近绝对位置 <value>（请参见以下说明）。                                              |
| <Addr>=ACN (<value>)        | 轴 <Addr> 沿着数学意义上的负向接近绝对位置 <value>。                                                     |
| 其中                          |                                                                                        |
| <Addr>                      | 轴地址<br>允许使用“模态”型轴。                                                                     |

使用高级语言语法的 NC 功能

| <Value> | <Addr> 的路径信息                                                                                                                                                                                                                                                                                                                                                                            |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Art>   | 定位类型<br><b>0:</b><br>无定位逻辑<br>行进的方向始终仅由新旧位置间的差异确定。<br><b>1:</b><br>最短的可能路径<br>轴的最大进给路径不超过相关模态值的一半。<br><b>2:</b><br>根据编程的正号 / 负号:<br>"+": 顺时针旋转;<br>"-": 逆时针旋转。<br><b>3:</b><br>活动软件限位开关 (1020 00001/3 和 1020 00002/4) 可定义许可的行进范围以及可能的方向。允许的通过范围从负限位开关一直到正限位开关。参考轴后, 启动限位开关范围。坐标规范在负模和正模范围内均有效 (例如, 20° 对应于-340° ) 坐标符号和局部坐标属性 "DC" 将被忽略。相反, 显示属性 "ACP"、"ACN" 和 "IC"。如果与工作范围冲突, 将导致错误消息。 |

表格 6-111: PosMode (PMD)、DC、ACP 和 ACN 的语法



数学上的正向:  
从坐标轴上以坐标原点的方向查看, 旋转的方向为逆时针。

|     |                                                            |                                                                                                                                |
|-----|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| 示例: | N40<br>B=ACP (-258)<br>:<br>N80 PMD (A1,C2)<br>:<br>:<br>: | 名称为 "B" 的物理轴沿着数学上的正向进给到 258 度的位置, 与 "PosMode" 无关。忽略正 / 负号。<br><br>以下轴的定位类型<br>轴 A: 最短的可能路径<br>B 轴: 根据 MP 1003 00005<br>C 轴: 符号 |
|-----|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|

- 特殊功能和限制:
- 除定位类型 "受限的进给范围" 外, 模态轴的软件限位开关必须始终为隐藏。
  - 只有在 MP 1003 00050 中释放切换后, 才可切换模态轴的定位类型。
  - 可在同一块中编程数个 DC、ACP 和 ACN 函数 (用于不同的轴)。
  - DC、ACP 和 / 或 ACN 函数仅对以下类型的轴有效: "旋转" 或 "模态"。若结合其他轴类型编程它们, 则控制器会忽略它们。
  - DC、ACP 和 ACN 函数位置信息中的负号会被忽略。
  - 以下内容适用于 "有限路径" 的定位类型:  
在正的模态范围内 (0...360° ) 内会始终显示负的终点角。  
坐标属性 DC "最短路径" 会被忽略。相反, 控制器将使用自动方向调整来选择 "允许的路径"。  
不调整使用符号、坐标属性或增量方式 (+/-, ACP/ACN 和 G91/IC) 定位的方向定位; 此类定位仅允许在限位开关范围之内使用。  
仅允许在完整旋转内进行运动。

6.89 精度编程 "PrecProg、PRP"

- 描述：减少轮廓过渡处和圆周路径区段处的进给，从而保证预设的精度。
- 要求的精度也可以通过指定
- 轮廓过渡或圆弧的最大允许 **轮廓或半径误差  $\epsilon$** ，或
  - 经过轮廓过渡进行进给时，不可超过的最大容许 **路径迟滞  $\delta$** （距离转角距离）。

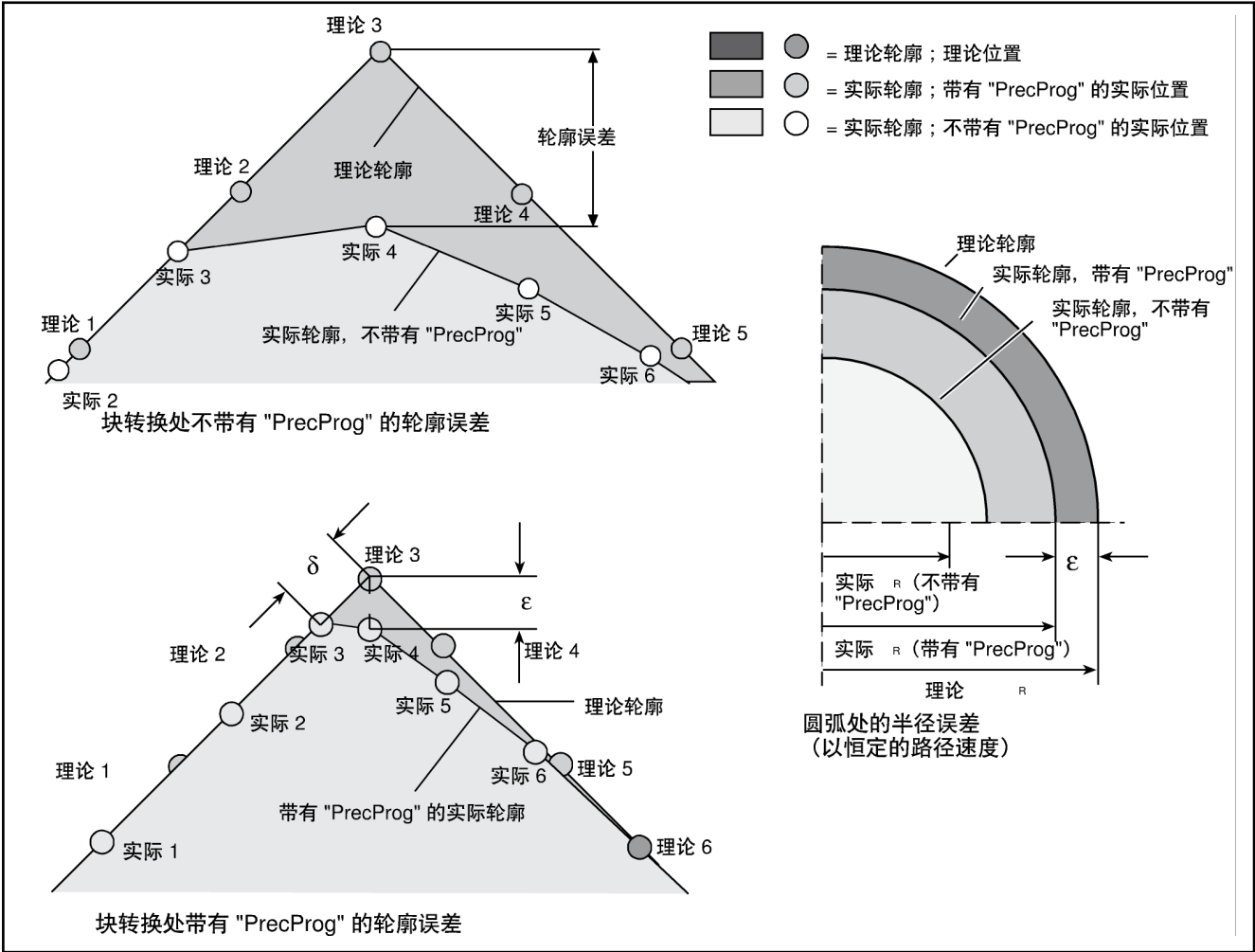


插图 6-30: 精度编程



与“精度”函数（G61/G62；参见第 5.2.22 章“准停定位开启 / 关闭 G61、G62”第 137 页）相比，速度无需在块切换处降至 0。请参阅“特殊功能和限制：”第 296 页。

语法：

|                           |                                                                              |
|---------------------------|------------------------------------------------------------------------------|
| PrecProg 或<br>PrecProg(1) | 精度编程开启<br>根据 MP 8003 00001 的最大容许轮廓 / 半径误差 $\epsilon$ 。                       |
| PrecProg(EPS<e>)          | 精度编程开启<br>允许的最大轮廓/半径误差：< $\epsilon$ >                                        |
| PrecProg(DIST<d>)         | 精度编程开启<br>允许的最大路径惯性运转：< $\delta$ ><br>考虑 MP 8003 00001 的圆弧路径段的 $\epsilon$ 值。 |

|                             |                                           |
|-----------------------------|-------------------------------------------|
| PrecProg() 或<br>PrecProg(0) | 精度编程关闭                                    |
| 缩写形式: PRP(...)              |                                           |
| 其中                          |                                           |
| < ε >                       | 允许的最大轮廓 / 半径误差 (单位: mm 或英寸, 取决于 G70/G71)。 |
| < δ >                       | 允许的最大路径惯性运转 (单位: 毫米或英寸, 取决于 G70/G71)。     |

表格 6-112:      PrecProg (PRP) 语法

- 特殊功能和限制:
- 所有受影响的轴必须使用相同的动态设置。
  - 若 G8 和 G62 处于非活动状态, 则每个块过渡处均减速至 v=0。
  - 只有在能够忽略加速度对惯性运转的影响时, 精度编程才可提供合理的结果。



如果括号表达式 "(0)" 或 "(1)" 不代表 NC 功能的参数。因此, 括号内的数值不能通过 CPL 变量传输。

## 6. 90      取决于块准备的程序处理 "PREP"

描述

如果零件程序中存在关键段 (临时) 非常短的块, 则可能需要确保在处理该段过程之前准备块的缓冲区几乎完全填满。因此, 如果块准备花费比块处理更长的时间, 则不会立即中止 (如果没有完全准备, 则处理无法获得新块)。

只要零件程序的第一个块已经完全准备好, MTX 就开始程序执行也可能不太合适。在这种情况下, 准备缓冲区仍然是空的。如果第一个块非常短, 可能会中止。

最后, MTX 提供了将程序传输到控制器并使用缓冲的 NC 块规范执行的选项。沟通的优先级较低, 可能会延迟。这也可能导致中止。

在三种应用场景中的任何一种情况下, 都可以在临界区之前有目的地停止执行, 直到通过块准备完全准备好足够数量的块为止。因此, 使用 NC 功能 "PREP"。使用此功能编程要准备好的块数。

该功能会将块的执行延迟最多 2 秒的时间。如果在此期间内没有准备好编程的块数, 则会输出运行时错误。编程关键字 "SILENT"可以消除此错误。



完全准备好的块保存在块预读的缓冲区中。  
该缓冲区大小由 NC0 / LookAh / Ch [k] / NofBlkPrep 定义, 并且必须至少具有编程块数的大小。

PREP 功能可能 **不会在异步子程序中编程**。

为了标记要准备的段落的末尾, 可以编程一个 PREP ( )。在这种情况下, 在程序块开始时, 需准备好 PREP (<N>) , 其中有大写的<N>块, 最后编程 PREP ( )。第一个 PREP 之后的第一个块只有在第二个 PREP 完全准备好之前的所有块才执行。

语法:

|                                |                             |
|--------------------------------|-----------------------------|
| PREP ( <BlkCnt> { , SILENT } ) |                             |
| 其中                             |                             |
| <BlkCnt>                       | 要准备好的块数                     |
| 无声                             | 如果关键字已编程, 则在超时的情况下不会生成错误消息。 |

|        |             |
|--------|-------------|
| PREP() |             |
|        | 完成准备的程序部分结束 |

表格 6-113: NC 功能 PREP 的语法

示例: 程序

```
N10 PREP(30)
N20 G1 X10 Y10 F12000 ; Block 1
; Pure comment -> it is not counted.
N30 X10.3 Y10.2 ; Block 2
...
N300 X120.4 Y13.1 ; Block 29
N310 X121.2 Y13.2 ; Block 30, when this block is completely prepared, N20 is executed
```

如果在示例中块前瞻的缓冲区很小，或者如果程序是通过缓冲的 NC 块规范指定的并导致延迟，那么会产生运行时错误，因为无法准备请求的块数。

为了抑制运行时错误，可以编程 N10 PREP (30, SILENT) 块，而不是 N10 行。

特殊功能和限制:

- 块准备缓冲区必须至少与要准备好的程序块数一样大。
- 该功能仅在手动输入，缓冲 NC 块规格和自动模式下等待。它不会等待任何其他操作模式（例如，单个模块）。
- 如果在等待时发现 WAIT 或程序结束，则块将完全准备到那里。然后，执行继续。
- 它不会在重置时等待。
- PREP 命令不能用于异步子程序。如果它是在异步子程序中编程的，则会产生运行时错误。
- 如果 SILENT 开关没有被编程，通讯中止（在缓冲 NC 块指定的情况下）会产生错误信息。
- 没有通道等待状态被激活。
- 要准备的块只包括进入内插器的块。这意味着，例如，CPL 块和空块不会被计算在内。对于因几何形状而分割的块，每个部分都要单独计算。
- 如果使用缓冲的 NC 块规范，则缓冲的 NC 块消息的第一个块可能不包含自动开始标志或 PREP。

6. 91 冲程释放时间

6. 91. 1 冲压释放时间（插补终点） "PtBlkEnd、PTE"

描述:



只有在结合“冲孔”（参见 第 6. 93 章“冲孔 Punch、PUN”第 302 页）和“裁料”（参见 第 6. 79 章“裁料 Nibble、NIB”第 274 页）函数使用时有效。

该函数

- 可定义给定轴的时间参照事件“NC 插补器接近进给运动的终点”，并可
- 指定时间参照事件和冲程释放之间的时间间隔。

这种方式可让冲程释放时间（基于时间参考）根据所需任意偏置。

可实现以下操作:

- 提前冲孔释放  
（例如，补偿由信号处理引起的常量、应用程序相关的等待时间）以及
- 后继冲孔释放  
（例如，使用较低动态增加轴的定位精度，或使用牵制设备提高冲孔的定位精度）。



影响冲程释放时间的其他函数：

- PtBlkEnd（请参阅 第 6.91.1 章 “冲压释放时间（插补终点） PtBlkEnd、PTE” 第 297 页）
- PtInpos（参见 第 6.91.3 章 “冲程释放时间（Inpos 窗口） PtInpos、PTI” 第 300 页）。

语法：

|                                          |                                                                      |
|------------------------------------------|----------------------------------------------------------------------|
| PtBlkEnd(<Axis><Time>{,<Axis><Time>}...) |                                                                      |
| 缩写形式： PTE(..)                            |                                                                      |
| 其中                                       |                                                                      |
| <Axis>                                   | 逻辑轴名称                                                                |
| <Time>                                   | 时间参照事件和冲程释放间所需的时间（单位：ms）。<br>0：事件参考时刻的行程释放<br>负值：早期中风释放<br>正值：延迟卒中释放 |

表格 6-114: PtBlkEnd (PTE) 语法

示例：

- 轴配置：X、Y、C
- Sercos 循环时间：3 ms

设置冲程释放时间：

|               |                                                                |
|---------------|----------------------------------------------------------------|
| :             |                                                                |
| N10 PTE(X-10) | X 轴的时间参照事件：                                                    |
| :             | “NC 插补器接近进给运动的终点”                                              |
| :             | <b>提前</b> 10 ms 释放冲程                                           |
| N20           | Y 轴和 C 轴的时间参照事件：                                               |
| PTI(Y10,C2)   | “已到达 Inpos 窗口”                                                 |
| :             | （有关该函数，请参见 第 6.91.3 章 “冲程释放时间（Inpos 窗口） PtInpos、PTI” 第 300 页）。 |
| :             | Y 轴： <b>延迟</b> 10 ms 释放冲程                                      |
| :             | C 轴： <b>延迟</b> 2 ms 释放冲程                                       |

进给运动时的效果：

|                    |                                                            |
|--------------------|------------------------------------------------------------|
| :                  |                                                            |
| N100 G1 Y20<br>C10 | 由于使用相同的时间参照编程了更长的等待时间，因此 Y 将决定冲程的释放行为（请参见 N20）。            |
| :                  | 由于等待时间被舍入至 Sercos 循环时隙，导致冲程释放实际延迟了 12 ms。                  |
| :                  |                                                            |
| N110 X20 C20       | 由于时间参考为 “Inpos 窗口” 的轴 “弱于” 时间参考为 “插补终点” 的轴，因此 C 将决定冲程释放特性。 |
| :                  |                                                            |
| :                  | 由于等待时间被舍入至 Sercos 循环时隙，导致冲程释放实际延迟了 3 ms。                   |
| :                  |                                                            |



|          |                                                                |
|----------|----------------------------------------------------------------|
| N120 X30 | 由于仅移动了 X，因此 X 将决定冲程的释放。                                        |
| :        | 实际提前释放冲程时间：-9 ms                                               |
| :        |                                                                |
| N130 PTD | 根据 MP 8001 00020 和 MP 8001 00021 来设置系统中所有轴的冲程释放                |
| :        |                                                                |
| :        | (有关该函数，请参见 第 6.91.2 章 “将冲程释放时间设置为默认值 PtDefault、PTD” 第 299 页 )。 |

- 特殊功能和限制:
- 函数 PTD、PTE 和 PTI 以模态的形式作用，并相互取消选定。
  - 必须在 MP 8001 00010 中应用冲孔或裁料。
  - 所有编程的时间值将被舍入，以匹配 Sercos 循环时间空档。
  - 若进给运动中包含多个定义的时间参考和等待期不同的轴，那么 “最弱” 的轴会决定实际的冲程释放特性。
- 以下情况适用:
- 时间参照为 “Inpos 窗口” 的轴 (参见 第 6.91.3 章 “冲程释放时间 (Inpos 窗口) PtInpos、PTI” 第 300 页 ) 比 “时间参照” 为 “终点插补” 的轴要弱。
  - 若时间参照相同，则 “最弱” 轴为等待时间最长的轴。
- 使用 PtDefault 函数 (参见 第 6.91.2 章 “将冲程释放时间设置为默认值 PtDefault、PTD” 第 299 页 ) 可将系统中所有轴的冲程释放复位为默认值。

6.91.2 将冲程释放时间设置为默认值 "PtDefault、PTD"

|     |                                                                                                                                                                                                                                                                                                                                                                          |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述: |  <p>只有在结合“冲孔”（参见 第 6.93 章“冲孔 Punch、PUN”第 302 页）和“裁料”（参见 第 6.79 章“裁料 Nibble、NIB”第 274 页）函数使用时有效。</p>                                                                                                                                                                                   |
| 语法: | <div> <div>PtDefault</div> <div>缩写形式: PTD</div> </div> <p>表格 6-115:      句法 <i>PtDefault</i>, <i>PTD</i></p> <p>请参阅 第 6.91.1 章“冲压释放时间（插补终点） PtBlkEnd、PTE” 第 297 页。</p>                                                                                                                                                                                                   |
| 示例: |  <p>影响冲程释放时间的其他函数:</p> <ul style="list-style-type: none"> <li>           PtBlkEnd<br/>           （参见 第 6.91.1 章“冲压释放时间（插补终点） PtBlkEnd、PTE” 第 297 页）         </li> <li>           PtInpos<br/>           （请参阅 第 6.91.3 章“冲程释放时间（Inpos 窗口） PtInpos、PTI” 第 300 页）打开。         </li> </ul> |
| 限制: | <ul style="list-style-type: none"> <li>函数“PTD”、“PTE”和“PTI”以模态的形式作用，并相互取消选定。</li> <li>必须在 MP 8001 00010 中应用冲孔或裁料。</li> </ul>                                                                                                                                                                                                                                              |

使用高级语言语法的 NC 功能

6.91.3 冲程释放时间（Inpos 窗口） "PtInpos、PTI"

描述:

 只有在结合“冲孔”（参见第 6.93 章“冲孔 Punch、PUN”第 302 页）和“裁料”（参见第 6.79 章“裁料 Nibble、NIB”第 274 页）函数使用时有效。

该函数

- 定义给定轴的时间参考事件“到达 Inpos 窗口”，并
- 指定时间参照事件和冲程释放之间的时间间隔。

这种方式可让冲程释放时间（基于时间参考）根据所需任意偏置（例如，使用较低动态增加轴的定位精度，或使用牵制设备提高冲孔的定位精度）。

 影响冲程释放时间的其他函数：

- PtBlkEnd  
(参见第 6.91.1 章“冲压释放时间（插补终点）PtBlkEnd、PTE”第 297 页)
- PtInpos  
(请参阅第 6.91.3 章“冲程释放时间（Inpos 窗口）PtInpos、PTI”第 300 页) 打开。

语法:

|                                           |                            |
|-------------------------------------------|----------------------------|
| PtInpos (<Axis><Time>{, <Axis><Time>}...) |                            |
| 缩写形式: PTI (...)                           |                            |
| 其中                                        |                            |
| <Axis>                                    | 逻辑轴名称                      |
| <Time>                                    | 时间参照事件和冲程释放间所需的时间（单位: ms）。 |

表格 6-116: PtInpos (PTI) 语法

示例:

请参阅第 6.91.1 章“冲压释放时间（插补终点）PtBlkEnd、PTE”第 297 页。

特殊功能和限制:

- 函数“PTD”、“PTE”和“PTI”以模态的形式作用，并相互取消选定。
- 必须在 MP 8001 00010 中应用冲孔或裁料。
- 所有编程的时间值将被舍入，以匹配 Sercos 循环时间空档。
- 若进给运动中包含多个定义的时间参考和等待期不同的轴，那么“最弱”的轴会决定实际的冲程释放特性。

以下情况适用:

- 时间参照为“Inpos 窗口”的轴（参见第 6.91.3 章“冲程释放时间（Inpos 窗口）PtInpos、PTI”第 300 页）比“时间参照”为“终点插补”的轴要弱。
- 若时间参照相同，则“最弱”轴为等待时间最长的轴。

- 使用“PtDefault”函数（参见第 6.91.2 章“将冲程释放时间设置为默认值 PtDefault、PTD”第 299 页）可将系统中所有轴的冲程释放复位为默认值。

6.92 PTP 进给运动 "PtpMove、PTP"

描述:

轴变换为活动状态时会执行 **点到点**(PTP)的进给运动，即在不取消轴变换的情况下，在 MCS 或 ACS 中执行运动。可以在 WCS、MCS 或 ACS 中指定程序编程的终点位置。控制器中将执行相应的转换。



有关辅助功能的使用和参数设置的全面信息，请参阅文档“IndraMotion MTX 功能描述”。

语法:

|                                                                                     |                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PTP (<EndposSys>, <MveSys>, TB<Mode>) 和 EndposSys 中的终点位置的编程。<br>(可选: PtpMove ( , )) |                                                                                                                                                                                                                                                                                                                                                              |
| 其中                                                                                  |                                                                                                                                                                                                                                                                                                                                                              |
| <EndposSys>                                                                         | 在其中已编程终点位置的坐标系。有效的:<br>WCS: 工件坐标中编程的终点位置<br>MCS: 机床坐标中编程的终点位置<br>AC: 轴坐标中编程的终点位置                                                                                                                                                                                                                                                                             |
| <MveSys>                                                                            | 在其中已执行运动的坐标系。有效的:<br>MCS: 以机器坐标运动<br>ACS: 以轴坐标运动                                                                                                                                                                                                                                                                                                             |
| TB<Mode>                                                                            | TrafoBranch<br>在双分支轴变换的情况下控制变换分支变化。仅适用于 PTP (WCS, MCS) 变体。“模式”可以有以下值:<br>0 (默认值):<br>靠近起始点的位置靠近两个可能的 MCS 位置 MCS (+1) 和 MCS (-1)。如果需要, 转换分支从 +1 更改为 -1, 反之亦然。<br>10:<br>在运动中保留转换分支<br>11:<br>在运动过程中更改转换分支<br>+1:<br>移至变换分支 +1, 即移至位置 MCS (+1)<br>-1:<br>移至变换分支 -1, 即移至位置 MCS (-1)<br>其他属性:<br>在一个分支转换的情况下, TB 被忽略<br>只有 PTP (WCS、MCS) 变体支持 TB 编程。其他变体会生成语法错误。 |

表格 6-117: 语法

在各个情况下, “MCS” 均作为默认值。

示例:

|                          |                          |
|--------------------------|--------------------------|
| PTP (WCS,MCS) x50 y55    | ; 达到 WCS 坐标 x=50, y=55   |
|                          | ; MCS 中的运动               |
| PTP (WCS,ACS) x=IC(50)   | ; WCS 坐标 x 以 50 的增量进行进给  |
|                          | ; ACS 中的运动               |
| PTP (MCS, MCS) X100 Y200 | ; MCS 坐标。到达 X=100, Y=200 |
|                          | ; MCS 中的运动               |
| PTP (MCS, ACS) X101 Y201 | ; MCS 坐标。到达 X=101, Y=201 |
|                          | ; ACS 中的运动               |

使用高级语言语法的 NC 功能

```
PTP (ACS, MCS) XA50 C45 ; ACS 位置。到达 XA=50, C=45
 ; MCS 中的运动

PTP (ACS, MCS) XA51 C46 ; ACS 位置。到达 XA=51, C=46
 ; ACS 中的运动

PTP () 与 PTP (MCS, MCS) 对应
PTP (,) 与 PTP (MCS, MCS) 对应
PTP (WCS,) 与 PTP (WCS, MCS) 对应
PTP (,ACS) 与 PTP (MCS, ACS) 对应
```

使用 5 轴转换进行 TB 编程的示例  
3232201:

```
B10 C0
COORD(1)
G1 phi0 theta30
PTP (WCS,,TB0)
phi180 theta30
或
PTP (WCS) phi180
theta30
PTP (WCS,,TB10) phi0 转换分支不变:
theta30 接近 MCS (-1) = (B-30, C180)

PTP (WCS,MCS,TB11) 转换分支改变:
接近 MCS (+1) = (B30, C0)
WCS 位置仍然是 Phi=0, Theta=30

PTP (WCS,,TB+1) 接近变换分支 +1。
没有运动, 因为转型分支已经是 +1

PTP (WCS,MCS,TB-1) 接近变换分支 -1。
接近 MCS (-1) = (B-30, C180)
WCS 位置仍然是 Phi=0, Theta=30
```

- 特殊功能和限制:
- 该函数为局部非模态函数。
  - PTP 进给运动总是按绝对编程解释, 即忽略 G91。可通过编程属性 "IC" (例如 x=IC(50)) 来执行增量编程。
  - 在 PTP 进给运动期间, 应停用 3D 半径补偿。否则, 将生成运行时错误。
  - 每条 NC 线只能编程一个 AXR 命令。

6.93 冲孔 "Punch、PUN"

**描述:** 开启或关闭 "冲孔"。

当启用打孔时, 如果是, 则在最后触发笔画

- 在各编程进给运动 (单次冲程操作) 的终点触发冲程, 或
- 在由 "NUM" (参见 第 6.80 章 "分隔进给块: 路径段数 NUM" 第 276 页) 或 "LEN" 函数 (参见 第 6.67 章 "分隔进给块: 路径段长度 LEN" 第 259 页) 生成的各路径区段的终点触发冲程

将总是在冲程完成前启动后续进给运动。



- 影响冲程释放时间的函数：
- PtDefault  
(参见 第 6.91.2 章 “将冲程释放时间设置为默认值 PtDefault、PTD” 第 299 页 )
  - PtBlkEnd  
(参见 第 6.91.1 章 “冲压释放时间 (插补终点) PtBlkEnd、PTE” 第 297 页 )
  - PtInpos  
(请参阅 第 6.91.3 章 “冲程释放时间 (Inpos 窗口) PtInpos、PTI” 第 300 页 ) 打开。

语法：

|                         |      |
|-------------------------|------|
| Punch (1) 或<br>Punch    | 冲孔打开 |
| Punch (0) 或<br>Punch () | 冲孔关闭 |
| 缩写形式： PUN (..)          |      |

表格 6-118: Punch (PUN) 语法

示例：

- G90 处于活动状态 (绝对尺寸编程)
- 激活平面：X/Y
- 当前位置：X=0, Y=0, C=0

```
:
N10 C10 冲孔打开
Punch (1) G9 (ASHAPE) 使用步冲逻辑启用插补器
 C 轴旋转至 10 度
 由于未对 X 和 Y 轴进行编程，所以无冲程。

N20 C60 C 轴旋转至 60 度
 由于未对 X 和 Y 轴进行编程，所以无冲程。

N30 X0 由于 X 轴已经位于 0 位置，因此无进给运动。
 由于 x 轴处于激活的平面内，因此将进行冲程。

N40 LEN=12 以下的进给块已被分割为多个常规路径区段，每个最大长度
 为 12 mm。

N50 X110 进给块被分割成 10 个相等的 11 mm 路径区段。
 在 X11、X22、X33...X99, X110

N60 Y30 NUM=3 覆盖当前块中以模式形式生效的 LEN (N40)。
 将进给块分割成 3 个相等的路径区段。
 Y10、Y20、Y30 处的冲程

N70 Y90 N40 中的 LEN 再次生效
 Y42、Y54、Y66、Y78、Y90 处的冲程。

N80 X50 Y50 冲孔关闭
Punch (0) G8 路径斜坡开启
: 进给运动至 X=50, Y=50
```

使用高级语言语法的 NC 功能

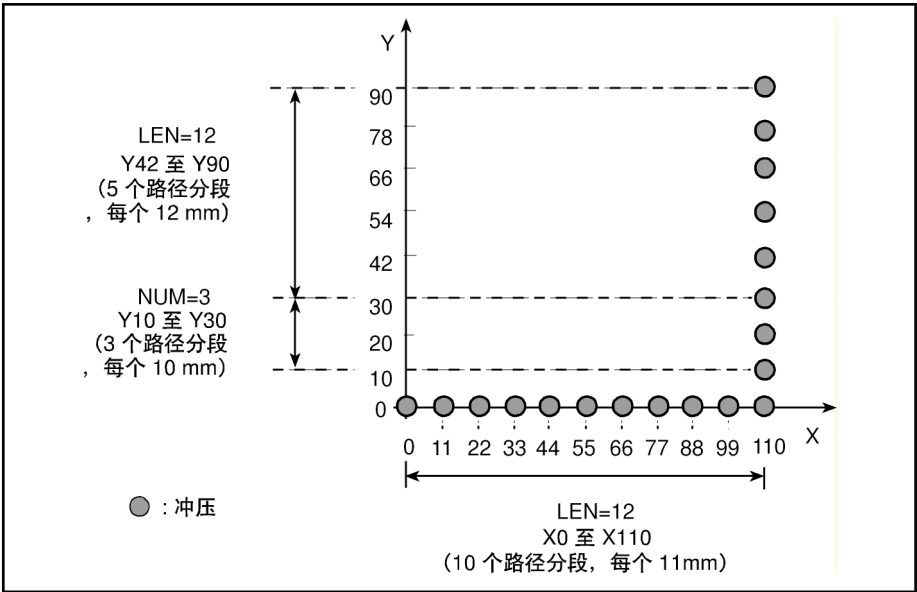


插图 6-31: Punch (PUN) 示例



要使用步冲功能，必须启用支持 n 步功能的插补器：

G9(X..., Y..., ...)  
G9(ASHAPE)  
G9(SHAPE..., PUN)

特殊功能和限制：

- 必须通过 MP 8001 00010 释放冲孔。
- 该函数对同一块中编程的进给运动已有模态效果。
- 控制器可以使用“NUM”或“LEN”函数，自动从行进的编程距离中生成路径区段；在这些路径区段中每一个的终点释放冲程。
- 若开启裁料（参见 第 6.79 章“裁料 Nibble、NIB”第 274 页），则该函数将取消选“定冲孔”。
- 不包含来自活动平面的轴坐标的块同样不会释放冲程。
- 若 PLC 禁止冲程释放，会在 PCL 释放冲程前将处理保持在冲程释放位置处。

6.94 最大径向加速度 "RadialAcc, RAC"

描述： 将样条和弧的路径进给降低，降低程度不超出工件坐标系（WCS）中指定的径向加速度。  
控制启动后，径向加速度限制被禁用。

语法：

|                     |                                                               |
|---------------------|---------------------------------------------------------------|
| RadialAcc (<value>) | 设置最大径向加速度                                                     |
| RadialAcc (0)       | 取消径向加速度限制                                                     |
| 缩写形式：RAC (...)      |                                                               |
| 其中                  |                                                               |
| <Value>             | 加速度值<br>依据激活的测量单位（G71/G70），控制器会将编程数据解释为“1000 Inch/s²”或“m/s²”。 |

表格 6-119: RadialAcc (RAC) 语法

示例:

```
N20 RAC(0.1) ; 将径向加速度限制为 0.1 m/s²。
N30 G1 G8 G90 G94 F5000 ; 路径进给最大值 5000 mm/min。
:
N50 G3 Y20 R10 ; Vmax= 1897 mm/min (Vmax=sqrt(RAC*R10))
N60 G2 Y30 R5 ; Vmax= 1341 mm/min (Vmax=sqrt(RAC*R5))
:
N100 RAC(0) ; 禁用径向加速度限制
```

- 特殊功能和限制:
- 在块的初始位置、中间点和终点随机检查样条的最大曲率
  - 总是对整个样条或圆周路径进行减少，并考虑所有与进给相关的坐标。
  - 只能减少进给。其他受机床参数或零件程序限制的进给正常工作。

## 6.95 记录作为系统数据的反向向量 "RecordRevVec, RRV, RRV"

描述:

开启或关闭“记录作为系统数据的反向向量”。

该数控函数操作的前提条件是存在名为 <name>RevVec</name> 的系统数据且已经由 <type>RevVec\_t</type> 和 dimension="Channel" 定义。

如果执行编入该数控函数的数控块，系统数据中的定向向量和副法线向量将保存为零。但该数控函数编入初始化字符串中的，则不属于该情况。

如果记录已启用，定向向量和副法线向量将在初始化后立即缩放至 1，并保存在系统数据的插补器循环中。

副法线向量与从运动方向和定向向量所创建的平面垂直。例如，在 G41/G42 启用的情况下，该向量为从刀具接触点到刀具中心点的向量。对于 G41，因为刀具位于轮廓左侧，该向量指向左方；对于 G42，该向量指向右方。



相关内容参见“《Rexroth IndraMotion MTX 功能描述》”手册中 G41/G42 和 G141/G142 函数的详细描述。

语法:

|                                     |                                      |
|-------------------------------------|--------------------------------------|
| RecordRevVec(1) 或<br>RecordRevVec() | 启用定向和副法线向量纪录，副法线向量指向左方，除非有效半径修正另有指定。 |
| RecordRevVec(2)                     | 启用定向和副法线向量纪录，副法线向量指向右方，除非有效半径修正另有指定。 |
| RecordRevVec(0) 或<br>RecordRevVec() | 停用定向和副法线向量记录。                        |
| 缩写形式: RRV(..)                       |                                      |

表格 6-120: Syntax RecordRevVec (RRV)

示例:

```
N3 G17(X,Y,Z) X0 Y0 等级选择: 在 X/Y 平面处理。
N5 RRV(2) 启用, 采用副法线向量至右方。
: 定向向量 = [0,0,0], 副法线向量 = [0,0,0]
N7 G41 X8 ED3 副法线向量至左方, G41。
N8 X20 定向向量 = [0,0,1], 副法线向量 = [0,1,0]
```

特殊功能和限制:

如果 G20 已启用，则不记录。

使用高级语言语法的 NC 功能

6.96 扭矩减小 "RedTorque、RDT"

**描述:** "RedTorque" 使用以轴到轴为基础的编程值来禁用机床参数 100300010 的值。  
机床参数本身的值未更改。  
由此, 减小的轴最大扭矩可在轴接口信号 "扭矩减少" (qAx\_TrqLim) 的正坡度不再被永久地限制为机床参数值时生效, 但可 "动态" 进行设置。

|     |                                            |                                                      |
|-----|--------------------------------------------|------------------------------------------------------|
| 语法: | RedTorque (<Axis 1> {, <Axis2> {, ... } }) | 将减小的最大扭矩值设置为编程的值。                                    |
|     | RedTorque (0) 或 RedTorque ()               | 将减小的最大扭矩值设置为机床参数 (MP 1003 00010)。                    |
|     | 缩写形式: RDT (..)                             |                                                      |
|     | 其中                                         |                                                      |
|     | <Axis i>                                   | 物理或逻辑轴名称, 包括轴静止扭矩的相应最大扭矩 (以百分比计)。<br>取值范围: 0 至 500 % |

表格 6-121: RedTorque (RDT) 语法

**示例:** N8 RDT (X5) 使用下个基于相关轴的 "扭矩减少" 信号的正沿, 可将轴 (此处为 X 轴) 的最大扭矩限制为其静止扭矩的 5%。  
:

6.97 程序取消后通过数控程序重启继续执行程序 "ReentContBlk, RCB"

**描述:** 在程序中标出位置, 程序取消后可从该位置继续执行。

|     |                                                                                        |                                                                                                                                     |
|-----|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | ReentContBlk, 缩写形式: RCB<br>RCB (1 {, <Cond 1> {, <Cond 2> {, <Cond 3> } })<br>RCBD (1) |                                                                                                                                     |
|     |                                                                                        | 标记可能继续执行点的位置, 并记忆可选条件的语法和值。<br>参见例 1、3 和 4                                                                                          |
|     | Cond [1, 2, 3]                                                                         | CPL 变量或 CPL 表达式:<br>变量或表达式必须为 Boolean, Int, Real 或 Double 类型。<br>如果变量为字符串或域字符串类型, 字符串或域字符串变量的评价应使字符串变量内容的解释与直接编入函数中情况相似。<br>参见示例 4。 |
|     | RCB (0)                                                                                | 删除最后记忆的位置。参见示例 2。<br>可使用整数变量替换第一个参数。状态可能为 0 或 1。<br>参见示例 5。                                                                         |
|     | RCBD (1)                                                                               | 删除最后记忆的位置。与 RCB (0) 相比, 它在异步子程序中也是有效的。                                                                                              |
|     |                                                                                        |                                                                                                                                     |

表格 6-122: 语法



|     |                                                                                                                                                                                                                                                                                                                                          |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 示例: | <div><div>示例 1:</div><div>N130 RCB(1)            将块 N130 标记为块预运行的可能目标。未规定条件。</div></div>                                                                                                                                                                                                                                                 |
|     | <div><div>示例 2:</div><div>N190 RCB(0)            删除数控程序重启的上一目标，例如例 1 中定义的目标。</div></div>                                                                                                                                                                                                                                                 |
|     | <div><div>示例 3:</div><div><div>310 FOR RUN%=1 TO 10<br/>N320 X[10 * RUN%]<br/>N330 P DrillHole<br/>N340 RCB(1, RUN%)<br/>350 NEXT RUN%</div><div>子例程 "DrillHole" 将在循环中执行 10 次。<br/>例如，如果循环或子例程因为紧急停止在第 7 次循环被取消，通过最后成功处理的 RCB( ) 保存值 "OPERATION% = 6"。<br/>如果按此规定执行数控程序重启，第 6 次循环中会出现目标块 N340，即通过数控程序重启，处理在第 7 次循环中启动。</div></div></div> |
|     | <div><div>示例 4:</div><div><div>500 DIM STRVAR\$(30)<br/>510 STRVAR\$="RUN%&lt;8"<br/>...<br/>N550 RCB(1, [STRVAR\$])<br/>...<br/>STRVAR\$</div><div>为字符串类型。<br/>要求使用方括号，以便 CPL 解释器删除字符串变量，并使用 "Lauf%&lt;8"进行替换。<br/>由此，使用 STRVAR\$="OPERATION%&lt;8" 进行 RCB(1, [STRVAR\$]) 编程与 RCB(1, OPERATION%&lt;8) 完全对应。</div></div></div>           |
|     | <div><div>示例 5:</div><div>N800 RCB([ENABLE%])            第一个参数的参数设定</div></div>                                                                                                                                                                                                                                                          |

**特殊功能和限制:** CPL 变量和表达式仅在块准备期间进行评价。如果已经评价这些表达式，“删除剩余距离”之后和回程模式中，不会再对表达式进行评价。

仅能程序被链接的情况下规定条件。在 DIN 模式中进行选择时，如果对条件进行评价，则会生成运行时错误。

仅可为以下类型对条件进行评价：  
整数、实数、双精度实数、布尔。

此外，不得规定结构化类型和域类型。

取消处理和后续“程序取消后数控程序重启”之间不得修改启用 RCB 调用中所涉及的程序。所涉及的程序为上次执行 RCB 命令的子例程列表中的所有程序，即主程序和至少编入 RCB 的程序。

在异步子程序中 RCB 被忽略而没有任何评论。如果在异步子程序中删除了再入口标记，请使用语法“RCBD(1)”或“RCBDelete(1)”。这是功能 RCB(0) 的功能扩展。

## 6.98 从轴组移除轴 "RemAxis、RAX"

**描述:** 从调用通道中移除同步轴。这会将同步轴变为异步轴。  
然后，可使用该异步轴的物理轴名称在任何通道中对该轴进行编程。

使用高级语言语法的 NC 功能

 有关“轴转换”功能性的详细信息，请参阅文档“IndraMotion MTX 功能描述”。

|     |                                                    |                         |
|-----|----------------------------------------------------|-------------------------|
| 语法： | RemAxis (<PAN> <PAI> <LAN>{,<PAN> <PAI> <LAN>}...) |                         |
|     | 缩写形式： RAX (...)                                    |                         |
|     | 其中                                                 |                         |
|     | <PAN>                                              | 物理轴名称<br>指定要从当前通道中移除的轴。 |
|     | <PAI>                                              | 物理轴索引<br>效果与 <PAN> 相同   |

|       |                       |
|-------|-----------------------|
| <LAN> | 逻辑轴名称<br>效果与 <PAN> 相同 |
|-------|-----------------------|

表格 6-123:      RemAxis (RAX) 语法

|     |                                                                                     |
|-----|-------------------------------------------------------------------------------------|
| 示例： | :                                                                                   |
|     | N030 RAX (XP,        物理轴 XP, 从通道中移除分配了索引 2 的物理轴以及逻辑轴<br>2, Z)                    Z。 |
|     | :                                                                                   |

- 特殊功能和限制：
- 移除轴并不会中断块准备。
  - 无效的轴名称会生成错误消息。
  - 如果在系统中定义了要移除的轴，但该轴在当前通道中已不再存在，则不会发出错误消息。
  - 必须始终根据 RemAxis(...) 来编程同一块中轴的位置。

## 6. 99      移除逻辑轴名称 "RemLogName、RLN"

描述：      从调用通道中删除同步轴的逻辑轴名称。  
该轴仍保留在通道中，但在此处仅可通过其物理轴索引或物理轴索引进行编程。

 有关“轴转换”功能性的详细信息，请参阅文档“IndraMotion MTX 功能描述”。

|     |                                                       |                               |
|-----|-------------------------------------------------------|-------------------------------|
| 语法： | RemLogName (<PAN> <PAI> <LAN>{,<PAN> <PAI> <LAN>}...) |                               |
|     | 缩写形式： RLN (...)                                       |                               |
|     | 其中                                                    |                               |
|     | <PAN>                                                 | 物理轴名称<br>指定当前通道中需要删除其逻辑轴名称的轴。 |
|     | <PAI>                                                 | 物理轴索引<br>效果与 <PAN> 相同         |

|       |                       |
|-------|-----------------------|
| <LAN> | 逻辑轴名称<br>效果与 <PAN> 相同 |
|-------|-----------------------|

表格 6-124:      RemLogName (RLN) 语法

示例:

```
:
```

```
N030 RLN(YP,3,Z) 物理轴 YP 的逻辑名称, 从调用通道中移除第三物理轴和逻辑轴 Z。
```

```
:
```

特殊功能和限制:

- 给定的轴必须为静态。  
否则, 控制器会生成错误消息, 并终止程序。
- 相同块中的轴位置必须始终根据 `RemLogName(...)` 表达式进行编程。

6.100 移除通道主轴 "RemSpindle、RSP"

描述:

从调用通道中移除逻辑主轴。

语法:

|                                                            |          |
|------------------------------------------------------------|----------|
| RemSpindle (<SysSpNo>   <SysSpName>   <ChanSpName , {...}) |          |
| 从通道中移除通道主轴。                                                |          |
| 其中                                                         |          |
| <SysSpNo>                                                  | 1..32    |
| <SysSpName>                                                | 系统主轴名称   |
| <ChanSpName>                                               | S1 .. S8 |
| 缩写形式: RSP                                                  |          |

表格 6-125:      *RemSpindle (RSP) 语法*

示例:

```
...
```

```
N50 RSP(S2) 从通道中移除第二通道主轴。
```

```
N60 M203 S2=500 因为没有可用的第二通道主轴, 生成运行时错误。
```

```
...
```

特殊功能和限制:

- 无法移除通道中以常量切削速度 (G96) 运行的主轴。
- 如果定义了通道专用整型辅助功能 CSP1 至 CSP8, 则会通知 PLC 当前的通道主轴分配 (例如, CSP2 0 不再分配给 ChannelSpindle2)。

6.101 块重复 "REPEAT, RPT"

描述:

NC 函数 REPEAT(<RepeatFactor>) (缩写形式: RPT) 可用于重复 NC 块。

重复因子指定块执行的频率 (1 = 一次)。允许的数值范围是 1 到 65535。

如果包含子程序调用, 则包含子程序的完整块将重复执行。子程序可能包含 RPTs 的更多块。

如果编程了运行运动并且模态子程序处于激活状态, 则将执行包含模态子程序的完整块。

在自动和手动数据输入模式下允许 RPT。

CPL 零件只有在第一次被调用时才会运行。

语法:

```
RPT <Repetition factor>或 REPEAT (<Repetition factor>)
```

- <Repetition factor>指定 NC 块执行的频率。  
取值范围: 1 <= n <= 65535
- CPL 变量有效

示例:

```
...
```

```
N10 G91 X10 Y10 RPT5 ; 该块被执行 5 次
```

使用高级语言语法的 NC 功能

```
...
20 MAXCNT%=10

N20 G91 X5 Y5 ; 包含子程序的完整块被执行 10 次
RPT[MAXCNT%] P Up

...

N10 G81

N20 G91 X20 Y20 ; 该块被执行 8 次。每次调用模态子程序。
RPT(8)

...
```

**特殊功能和限制:** RPT 在 InitString 中或在块预运行的操作块中无效。

## 6. 102 异步子例程

### 6. 102. 1 定义重新定位特性 "REPOSDEF"

**描述:** (在异步子例程中) 局部定义重新定位进给和/或返回工件加工位置点。

语法:

|                                                |                                                           |
|------------------------------------------------|-----------------------------------------------------------|
| REPOSDEF( {F<Feed>   OF<Feed>} , {TP<point>} ) |                                                           |
| 其中                                             |                                                           |
| F<Feed>                                        | 异步子例程结束时，自动重新定位运动的进给。                                     |
| OF<Feed>                                       | 异步子例程结束时，自动重新定位运动的可选进给速度。<br>只有在启动异步子例程时无激活的进给时，才会使用该进给值。 |
| TP<Point>                                      | 1: 起始点<br>2: 终点<br>3: 中断点                                 |

表格 6-126: 语法

**特殊功能和限制:**

- 该函数是为了在异步子例程中使用而设计的。
- 子例程结束后，之前为正在运行的异步子例程使用 ASPRTP（请参见 第 6.8.5 章 “异步子程序：注册 ASPSET” 第 195 页）设置的重新定位点会自动重新生效。

### 6. 102. 2 定义异步子例程的路径返回点 "REPOSTP"

**描述:** 在异步子例程末端编程 "REPOSTP" 函数后，定义控制器是否定位至

- 起始点
- 终点
- 中断点

一个可能被中断的移动块。

若中断时没有进给块，则控制器将始终停留在最后激活的坐标。

REPOSTP 由此禁止在之前为当前运行的异步子例程使用 ASPRTP 设置的路径返回点（参见 第 6.8.4 章 “异步子程序：定义返回到轮廓点 ASPRTP” 第 195 页）。



有关异步子例程的使用和参数化的全面信息，请参阅文档 “IndraMotion MTX 功能描述”。

|         |  |                                            |  |
|---------|--|--------------------------------------------|--|
| 语法:     |  | REPOSTP (<point>)                          |  |
|         |  | 其中                                         |  |
| <Point> |  | 所需的返回工件加工位置点:<br>1: 起始点<br>2: 终点<br>3: 中断点 |  |

表格 6-127: REPOSTP 语法

- 特殊功能和限制:
- 该函数是为了在异步子例程中使用而设计的。
  - 子例程结束后，之前为正在运行的异步子例程使用 ASPRTP（请参见 第 6.8.4 章 “异步子程序：定义返回到轮廓点 ASPRTP” 第 195 页 ）设置的重新定位点会自动重新生效。
  - 在异步子程序中进行的任何校正更改都将自动包含在轮廓所需返回点的内部计算中。

6.103 回程、RETR

描述:

利用“回程”数控函数进行标记（将启用“轮廓上的回程”功能）零件程序中的程序部分。

在该识别标记内，控制器在处理零件程序时生成前进和后退程序。这些程序可以在反向操作中处理。

逐行，前进和后退程序包含零件程序的轮廓生成规格。程序坐标和轮廓相关 NC 功能的一部分自动传送到正向和返回程序。不发送异步轴运动。有关正向和反向程序的特定于应用程序的调整，请参阅 ADVA, ADVS, REVA 和 REVS。

回程本身是通过接口信号控制的。

|     |  |                          |                                                                                                              |
|-----|--|--------------------------|--------------------------------------------------------------------------------------------------------------|
| 语法: |  | 回程(1)                    | 标记轮廓上的回程启用的区域开始。                                                                                             |
|     |  | Retrace([<fwd>][,<bwd>]) | 作为 Retrace (1)，但具有向前和向后程序的数据存储。<br><br>这些文件是回程模式的内部程序的图像。<br><br>程序文件在运行时生成。范围识别结束后 - 回程 (0) - 已处理完毕，可以查看文件。 |
|     |  | 其中                       |                                                                                                              |
|     |  | <fwd>                    | 回程转发程序文件名称                                                                                                   |
|     |  | <bwd>                    | 回程向后程序文件名称                                                                                                   |
|     |  | Retrace(0)               | 标记轮廓上的回程启用的区域结束。                                                                                             |
|     |  | 缩写形式: -                  |                                                                                                              |

表格 6-128: Retrace, RETRACE 语法

下表显示了正转和返回程序中自动发送的 NC 功能:

| 数控函数           | 功能组      |
|----------------|----------|
| G0、G1、G2、G3、G5 | 插值函数*)   |
| G8, G9         | 路径斜坡*)   |
| G17, G18, G19  | 平面预选*)   |
| G40、G41、G42    | 铣削路径修正*) |

使用高级语言语法的 NC 功能

|                          |                  |
|--------------------------|------------------|
| G43、G44                  | 转换路径段            |
| G45、G46                  | 在铣刀接触点 / 铣刀中心点进给 |
| G47、G48                  | 刀具修正*)           |
| G53. x, . . . , G59. x   | 零点偏置             |
| G61、G62                  | 准确停止             |
| G70、G1                   | 英制/公制编程          |
| G90、G91                  | 绝对尺寸, 相对尺寸编程     |
| G93、G94、G95              | 进给编程             |
| G140、G141、G142           | 3D 刀具半径修正        |
| G153. x, . . . , G159. x | 展示位置             |
| CHLL, CHSL               | 局部倒角编程           |
| SC0, SCOL                | 样条模式或局部圆角整合      |
| RNDL                     | 局部修圆             |
| LockRetrace              | 回程切换锁 (开)        |
| NcfLockRetraceOff        | 回程切换锁 (关)        |

\*) 前进和后退程序中没有参数支持  
表格 6-129: 可在回程区域切换 NC 功能

示例:

```
...
N100 Retrace(1)
N110 G1 F1000 X27.571 Y3.958
N120 X38.729 Y26.945
N130 M125 ADVA(M125) REVA(M126)
...
N200 Retrace(0)
...
```

- 特殊功能和限制:
- 回程区域限制回程模式。回程只能在活动区域内进行。这也适用于在零件程序中定义了几个区域的情况
  - 正向和反向程序包含已解析的 NC 块序列, 不包含 CPL 语句, 也不包含子程序调用
  - 前向和后向程序中不能插入子程序调用
  - 向后操作支持线性和圆弧插补
  - 回程中的模式切换仅适用于 列出的功能 表 6-129 格 “可在回程区域切换 NC 功能” 第 311 页

 如果括号表达式 “(0)” 或 “(1)” 不代表 NC 功能的参数。因此, 括号内的数值不能通过 CPL 变量传输。

6. 104 回程切换锁 LockRetrace, LOCKRETRACE

- 描述:
- 使用数控函数 “LockRetrace” 识别回程切换已锁定、产生警告 / 消息的零件程序内的程序部分。可传递以这种方式标记的区。但是, 无法
- 启动带有后续回程的回程模式
  - 更改回程模式下的加工方向
  - 通过继续中止的零件程序, 终止回程模式

|     |                |                   |
|-----|----------------|-------------------|
| 语法: | LockRetrace(1) | 回程切换已锁定的程序区起点的标记  |
|     | LockRetrace(0) | 回程切换已锁定的程序区终点的标记。 |
|     | 缩写形式: -        |                   |

表格 6-130: LockRetrace, LOCKRETRACE 语法

示例:

```
...
N100 LockRetrace(1)
N110 SplineDef(2203,X,Y,Z)
N120 G6
...
N200 LockRetrace(0)
...
```

- 特殊功能和限制:
- 在通道复位的情况下，自动取消可能仍然处于激活状态的回程切换锁。
  - 如果回程切换锁已启用，且在回程区域开始之前未再次禁用过，已锁定区域无法在回程操作期间退出。仅可在通道复位的情况下取消对程序的选择。



如果括号表达式“(0)”或“(1)”不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

## 6.105 扩展反向回程程序块"AdvanceAppend, ADVA"

描述: 对于反向操作，将为回程区域回程（1）.. 回程（0）中的每个零件程序块生成一个反向程序块。功能 AdvanceAppend 在当前正向程序块的末尾插入指定的 NC 功能。

|     |                              |        |
|-----|------------------------------|--------|
| 语法: | ReverseAppend(<NCF>...<NCF>) |        |
|     | 缩写形式: REVA(...)              |        |
|     | 其中                           |        |
|     | <NCF>...<NCF>                | 数控函数列表 |

表格 6-131: 语法 ReverseAppend (REVA)



轮廓更改会导致反向操作中的切换错误。  
不允许 CPL 功能和子例程调用。

|     |                                              |                                        |                                              |
|-----|----------------------------------------------|----------------------------------------|----------------------------------------------|
| 示例: | 零件程序:                                        | 向后程序:                                  | 注意:                                          |
|     | N100 X100 M10<br><br>N200 X0 REVA(M125 M126) | N1 X=10<br><br>N2 X = 100 M125<br>M126 | 辅助功能不适用<br><br>在向后程序中，M125 和 M126 将被输出到 PLC。 |



为了检查目的，将修改后的转发程序保存在文件中（请参阅 [第 6.103 章“回程、RETR”第 311 页](#)）。

使用高级语言语法的 NC 功能

6.106 替换反向回程程序块"AdvanceAppend, ADVA"

**描述:** 对于反向操作，将为回程区域回程（1）.. 回程（0）中的每个零件程序块生成一个反向程序块。“ReverseSubstitute” 功能用指定的 NC 功能替换当前的反向程序块。

|                                 |        |
|---------------------------------|--------|
| ReverseSubstitue(<NCF>...<NCF>) |        |
| 缩写形式: REVS (...)                |        |
| 其中                              |        |
| <NCF>...<NCF>                   | 数控函数列表 |

表格 6-132: 句法 ReverseSubstitute (REVS)



轮廓更改会导致反向操作中的切换错误。  
不允许 CPL 功能和子例程调用。

|            |                                          |                            |                                                  |
|------------|------------------------------------------|----------------------------|--------------------------------------------------|
| <b>示例:</b> | <b>零件程序:</b>                             | <b>向后程序:</b>               | <b>注意:</b><br><br>在向后程序中允许更高的进给。在编程坐标的后退程序中设置起点！ |
|            | N1 X100<br><br>N2 X0 F10 REVS(X100 F100) | N1 X10<br><br>N1 X100 F100 |                                                  |



为了检查目的，将修改后的转发程序保存在文件中（请参阅 第 6.103 章 “回程、RETR” 第 311 页）。

6.107 输入工具：旋转 "Rotate(...), ROT(...)"

**描述:** “旋转” 函数是输入工具的一种。  
控制器在活动平面内（关于平面，请参见 G17、G18、G19 或 G20）旋转编程的轮廓。  
旋转将始终参照当前的旋转点（参见函数 “PoleSet”， 第 6.86 章 “定义镜像点 / 旋转点 PoleSet、PLS” 第 291 页）。若未进行明确编程，则旋转点为当前程序的零点。  
使用此功能，围绕特定角度旋转的循环编程步骤只能编程一次。  
另外，角度工件相对于基本工件坐标系的尺寸不必再被转换；直接从生产图纸中取出并指定相应的旋转角度。其余的控制完成。



旋转是一种输入工具，它不会更改当前的程序坐标系。输入工具只为程序坐标提供其他输入方法。  
“旋转” 也可结合 “镜像” 和 “缩放” 使用。

以下情况适用:

- 函数以模态方式作用。其在关闭前保持活动状态。
- 可在同一块中将之作为其他路径条件和输入工具进行编程。

|            |                             |                                                |
|------------|-----------------------------|------------------------------------------------|
| <b>语法:</b> | Rotate(<angle of rotation>) | 激活围绕所需 <AngleOfRotation>的旋转。                   |
|            | Rotate(0) 或 Rotate()        | 取消通道中所有轴的旋转。将所有的旋转角度设为 “0”。所接近的轴位置在被重新编程前始终有效。 |



|                  |    |                                                                                                                                                 |
|------------------|----|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 缩写形式: ROT(..)    |    |                                                                                                                                                 |
| 其中               |    |                                                                                                                                                 |
| <Angle rotation> | of | >0: 逆时针旋转<br><0: 顺时针旋转<br>=0: 旋转关闭<br>活动平面中所有后续编程的坐标均围绕该旋转点旋转 (参见 PoleSet, 第 6.86 章 “定义镜像点 / 旋转点 PoleSet、PLS” 第 291 页 )。<br>旋转在下一进给信息前不会变为活动状态。 |

表格 6-133: Rotate(...), ROT(...) 的语法

示例:

:

旋转打开

N30 ROT(45)

活动平面中所有后续编程的坐标均围绕某个可能已编程的旋转点逆时针旋转 45 度。

:

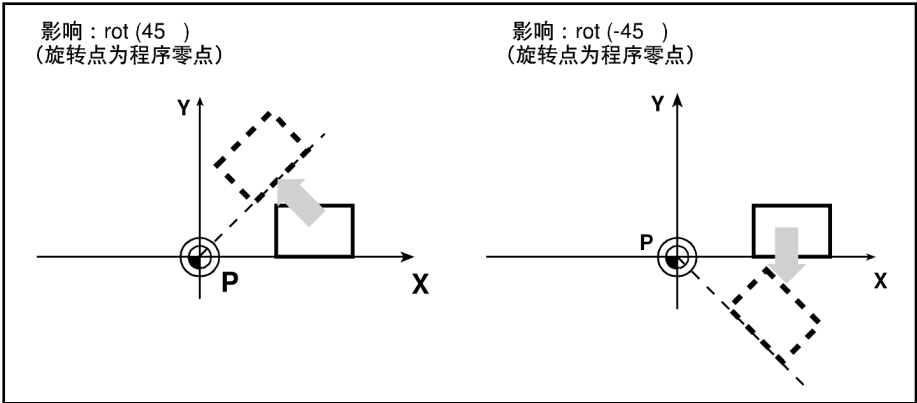


插图 6-32: Rotate(...), ROT(...) 示例

旋转定向向量:

定向向量关于活动平面的法线旋转。  
任何可能处于活动状态的缩放、镜像或旋转点都不会影响结果。  
语法如 “编程” 中描述。

- 特殊功能和限制:
- 该函数在圆周插补的情况下会考虑插补参数
  - 它影响着可编程的轮廓切换。  
请参阅 第 6.114 章 “编程的轮廓移位 Shift、SHT” 第 324 页 。
  - 它 不 影响:
    - 零点偏置 (G54-G59.5; 参见 第 5.2.21 章 “零点偏置 (Z0) G53、G53.1-G59.1 至 G53.5-G59.5” 第 135 页 )、
    - 程序坐标偏置 (Trans 或 ATrans; 参见 第 6.148 章 “程序坐标偏置 Trans、TRS, 叠加程序坐标偏置 ATrans、ATR” 第 361 页 )、
    - 程序位置的设置 (“SetPos”; 参见 第 6.113 章 “设置程序位置 SetPos、SPS” 第 323 页 )、
    - 接近的参考点坐标 (G74; 参见 第 5.2.26 章 “接近参考点坐标 G74 ” 第 141 页 )、
    - 接近的固定机床轴位置坐标 (G76; 参见 第 5.2.29 章 “接近固定的机床轴位置 G76” 第 143 页 )、
    - 铣削半径和刀具长度修正值。

使用高级语言语法的 NC 功能

6. 108 转角的修圆

6. 108. 1 指定偏差的转角修圆 "RoundEps、RNE"

描述: 此功能在主平面中的 2 个线性块之间插入切向过渡圆弧。  
这会导致在这些角上编程轮廓的微小修改，但是在插补过程中会获得连续的速度和加速度特性。

|     |                               |                                                            |
|-----|-------------------------------|------------------------------------------------------------|
| 语法: | RoundEps (<deviation>)        | 激活在 2 个线性块之间的修圆转角。                                         |
|     | RoundEps (DEF)                | 使用 MP 7050 00110 中的默认偏差，在二个线性块间打开边的修圆。                     |
|     | RoundEps (0) 或<br>RoundEps () | 在两个线性块间关闭边的修圆。                                             |
|     | 缩写形式: RNE (..)                |                                                            |
|     | 其中                            |                                                            |
|     | <Deviation>                   | 修改的和编程的轮廓间允许的最大偏差（单位: mm）。<br>小数点数是允许的。<br>控制器计算适合的切向过渡圆弧。 |

表格 6-134: RoundEps (RNE) 语法

- 特殊功能和限制:
- 功能 "ChLength", "ChSection", "RoundEps" 和 "Rounding" 以**模态行事** 並相互取消选取。
  - 控制器 **不** 执行 "修圆转角" 函数如果
    - 两个相邻块中至少有一个不是线性块;
    - 相邻的两个块中至少有一个具有选定的主平面外的路径区段; 或
    - 相邻的两个块至少有一个块存在行进路径, 比 MP 7050 00120 中设置的路径短 (2 至 90 mm, 默认值: 10 mm) 或
    - 若根据机床参数将块过渡视为连续性, 即两个块之间的角度小于 MP 7050 00130 中指定的最大角度 (默认 = 1 度)。
  - 若平面、零点偏置或轴变换在两个连续的块之间切换, 则不生成任何舍入。

6. 108. 2 半径规范的 "RND"修圆转角, "倒圆", "RNDL"

描述: 此功能在主平面中的 2 个线性块之间插入切向过渡圆弧。  
这会导致在这些角上编程轮廓的微小修改，但是在插补过程中会获得连续的速度和加速度特性。  
"具有半径规范的修圆转角" 功能以模态方式 (倒圆, RND) 和逐块 (RNDL) 提供。在逐块编程时, 可以定义一个本地进给来机器处理舍入。

|     |                               |                             |
|-----|-------------------------------|-----------------------------|
| 语法: | Rounding (<Radius>)           | 启用两个线性 / 圆形 / 螺旋块之间角落的模态倒圆。 |
|     | Rounding (0) 或<br>Rounding () | 启用两个线性/圆形/螺旋块之间角落的模态倒圆。     |
|     | 缩写形式: RND (..)                |                             |
|     | 其中                            |                             |
|     | <Radius>                      | 所需的过渡圆弧半径。小数位数是允许的。         |

表格 6-135: 修圆 (RND) 语法

|                                 |                     |
|---------------------------------|---------------------|
| RNDL (<Radius> {, <LocFeed> } ) | 修圆本地转角              |
| 其中                              |                     |
| <Radius>                        | 所需的过渡圆弧半径。小数位数是允许的。 |
| <LocFeed>                       | 用于加工倒圆的本地进给。        |

表格 6-136: RNDL 语法

- 特殊功能和限制:
- 功能 “ChLength”, “ChSection”, “RoundEps” 和 “Rounding” 以**模态行事** 並相互取消选取。
  - “RNDL” 功能 **在本地执行** （逐块）。
  - 如果倒角或倒圆在模态上是激活的，并且如果同时编程了局部舍入，则局部功能会逐个模块交叉衰减模态功能。
  - 控制器 **不执行“修圆转角”函数**如果
    - 相邻的两个块至少有一个块存在行进路径，比 MP 7050 00120 中设置的路径短（2 至 90 mm，默认值：10 mm）或
    - 根据机床参数将块过渡视为连续性，即两个块之间的角度小于 MP 7050 00130 中指定的最大角度（默认 = 1 度）。
  - 若平面、零点偏置或轴变换在两个连续的块之间切换，则不生成任何舍入。
  - 只考虑用于修圆角的活动工作平面中的组件。对于空间直线的情况，这会导致方向在空间中发生改变。类似的适用于螺旋路径段。

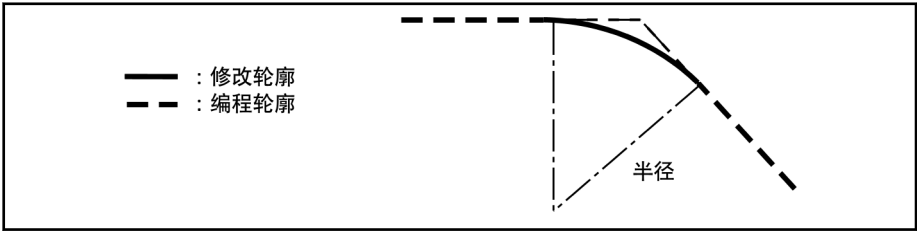


插图 6-33: 以半径指定圆角

6.109 同步系统轴耦合 "SACSYNC"

- 描述:
- 使用系统轴耦合条件 (SAC) 同步数控程序，并将解除耦合位置 (\$目标位置) 传递至工件坐标系，而无需触发轴运动。
- 如果同一块中存在编程的运动，该命令在结束点降低路径速度至零，并在达到结束点后有效果。
- 一旦系统轴耦合位于指定条件，则应用位置并继续程序处理。

语法:

|                                                                      |                                                                  |
|----------------------------------------------------------------------|------------------------------------------------------------------|
| SACSYNC                                                              | 等待所有通道轴的系统轴耦合 (SysAxCoupleSta[i].State=0) 关闭，并应用 NC 坐标系统的解除耦合位置。 |
| SACSYNC (TO=0)                                                       | 应用所有接触耦合通道轴的接触耦合位置而无需等待。                                         |
| SACSYNC (<Axis1>=<SAC status>, ..., <AxisN>=<SAC status>)            |                                                                  |
|                                                                      | 等待直至满足轴的所有 SAC 条件，并应用已解除耦合轴的停用位置。                                |
| SACSYNC (TO=<time>, <Axis1>=<SAC status>, ..., <AxisN>=<SAC status>) |                                                                  |
|                                                                      | 但与之之前一样，仅等待直至时间 <Time> (单位为 ms) 之后。如果等待条件不出现，控制器输入错误且不应用耦合位置。    |

使用高级语言语法的 NC 功能

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 带 <Axis i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 通道轴标识符或<br>系统轴标识符或<br>通道主轴标识符（S1，...，S8）或<br>系统轴标识符                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <p>SAC 状态可以从 SysAxCoupleSta [i] .State 中假设以下值：</p> <p>0: 关闭 (OFF)</p> <p>1: 耦合 (HUNTING)</p> <p>2: 重新耦合 (RECOUPLING)</p> <p>3: 同步 (COUPLED)</p> <p>4: 正在停止 (STOPPING)</p> <p>5: 已停止 (STOPPED)</p> <p>6: 解除耦合 (DISCOUPLE)</p> <p>7: 错误 (ERROR)</p> <p>1x: 放弃 (ABORT-x)，x 从 1-7</p> <p>2x: 实时切换 (PENDING-x)，x 从 1-7 开始</p> <p>以下来自 SysAxCoupleSta [i] .StateAddOn 的值可以作为 SAC 状态中的等待条件被覆盖 (x 从 1-7 开始，y 从 1-2 开始)：</p> <p>1000, 10yx: 同步窗口 1 到达 (INPOS1)</p> <p>2000, 20yx: 同步窗口 2 到达 (INPOS2)</p> <p>3000, 30yx: 同步窗口 1 和 2 到达 (INPOS1+INPOS2)</p> | <p>注释：</p> <p>等候位置应使用静态状态，例如：</p> <p>0: 关闭，</p> <p>3: 同步或</p> <p>5: 已停止</p> <p>因为过渡状态如：</p> <p>1: 解耦，</p> <p>2: 重新耦合，</p> <p>4: 停止或</p> <p>6: 解耦</p> <p>根据轴位置或轴速度可能不会拍摄或仅拍摄。</p> <p>SACSYNC 将等待状态与当前 SAC 状态进行比较，如下所示：</p> <ul style="list-style-type: none"><li>等待条件为 0 或仅包含来自 SysAxCoupleSta [i] .State 的可能状态：<br/>SACSYNC 等待与 SysAxCoupleSta [i] .State 的等待条件完全匹配。</li><li>等待条件仅包含来自 SysAxCoupleSta[i].StateAddOn 的可能状态：<br/>SACSYNC 等待 SysAxCoupleSta [i] .StateAddOn 在等待条件下主动设置的状态匹配。</li><li>等待条件包含来自 SysAxCoupleSta [i] .State 和 SysAxCoupleSta [i] .StateAddOn 的可能状态：<br/>SACSYNC 在 StateAddOn 的等待状态中等待相同的状态设置为与 SysAxCoupleSta [i] .StateAddOn 处于活动状态，并等待状态区域中的等待 状态与 SysAxCoupleSta [i] .State 完全匹配。</li></ul> |

表格 6-137: SACSYNC 语法

示例

```
； 启用 XB 轴耦合位置修正：

1 XB%=10 ； 系统轴索引 10

1 SD.SysAxCoupleCmd[XB%].SyncMode = 0 ； 特定位置的耦合

1 SD.SysAxCoupleCmd[XB%].OnOff = 1 ； 启用耦合

1 SD.SysAxCoupleCtr.Validate = 1 ； 启用数据

SACSYNC (XB=3) ； 等待同步运行

... ； 加工

； 停用 XB 轴耦合，无补偿运动：

1 SD.SysAxCoupleCmd[XB%].SyncMode = 2 ； 速度补偿

1 SD.SysAxCoupleCmd[XB%].OnOff = 0 ； 停用耦合
```

使用高级语言语法的 NC 功能

```
1 SD.SysAxCoupleCtr.Valid = 1 ; 启用数据
SACSYNC ; 等待停用并应用解除耦合位置
```

表格 6-138: SACSYNC 使用示例

更多示例

|                         |                                                    |
|-------------------------|----------------------------------------------------|
| SACSYNC (XB=0)          | ; 等到 xb 轴的耦合关闭。                                    |
| SACSYNC (S1=3, SSP11=3) | ; 等待通道主轴 S1 和系统主轴 SSP11 连接命令值。                     |
| SACSYNC (XB=5)          | ; 等到 xb 轴的耦合停止。                                    |
| SACSYNC (XB=1000)       | ; 等到 be 轴到达 INPOS1 窗口。它是否也到达 INPOS2 窗口并不重要。        |
| SACSYNC (XB=3000)       | ; 等到 xb 轴到达 INPOS1 窗口和 INPOS2 窗口。                  |
| SACSYNC (XB=2003)       | ; 等到 xb 轴被命令值耦合并到达 INPOS2 窗口。它是否也到达 INPOS1 窗口并不重要。 |
| SACSYNC (XB=3003)       | ; 等到 xb 轴为命令值耦合并且到达 INPOS1 窗口和 INPOS2 窗口。          |

表格 6-139: 更多 SACSYNC 呼叫的示例

特殊特性和限制

- 在使用 SACSYNC (<AsyncAxis> = 0) 编程去激活条件之前，异步轴必须完成插补。也可规定插补启期间的其他条件。
- 通过通道或系统复位取消该函数的情况下，如果轴已经通过 SyncMode=2（无补偿运动）解除耦合，则耦合偏置保持启用。
- 为了在复位时应用通道轴带的解除耦合位置（\$目标位置）通道轴带，SACSYNC(TO=0) 可保存在初始化字符串中。复位不会停用已经启用的耦合。
- 只能使用主轴插补器（M3 / M4, M19, SPCC 切换）或驱动参数组切换（C 轴模式，换档等）来减少主轴的耦合偏移。
- 由于使用 PENDING (+20) 直到切换的当前耦合状态为（），因此可以通过 SACSYNC (... <axis> = <state> ...) 在主轴数据的条件激活（SwitchCond> 0）叠加。
- SACSYNC 只等待到达同步窗口，但不能退出它。

6.110 输入工具: Scale"Scale(...), SCL(...)"

描述: “缩放”函数是输入工具的一种。  
控制器会参照程序零点，缩放编程的轮廓。

缩放会始终参照当前的镜像点（参见“PoleSet”函数，第 6.86 章“定义镜像点 / 旋转点 PoleSet、PLS”第 291 页）。若未对此进行明确编程，则当前的程序零点为镜像点。  
“Scale”函数可用在零件程序中，使用固定尺寸（标准尺寸）编程轮廓。之后，在调用此类“标准化”零件程序（例如作为子例程使用）之前，可使用各轴的缩放系数来决定编程轮廓的缩放。  
由此便可容易补偿，例如制造铸件和锻件的模具时工件的收缩。



缩放是一个输入工具，因此它不会更改当前的程序坐标系。输入工具只为程序坐标提供其他输入方法。  
“缩放”也可结合“镜像”和“旋转”使用。

- 以下情况适用:
- 函数以模态方式作用。其在关闭前保持活动状态。
  - 可在同一块中将之作为其他路径条件和输入工具进行编程。

使用高级语言语法的 NC 功能

语法:

|                               |                                                                                                                                             |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Scale(<Axis 1><Factor>{,...}) |                                                                                                                                             |
|                               | 使用编程的系数启用指定轴的缩放。                                                                                                                            |
| Scale(0) 或<br>Scale()         | 取消通道中所有轴的缩放。<br>将所有缩放系数设为值“1”。<br>所接近的轴位置在被重新编程前始终有效。                                                                                       |
| 缩写形式: SCL(...)                |                                                                                                                                             |
| 其中                            |                                                                                                                                             |
| <Axis 1>                      | 要进行缩放的轴的地址（例如，X）。                                                                                                                           |
| <Factor>                      | 使用正系数编程轴地址将激活此函数。<br>由此，对应轴（如 X10）的所有后续编程的路径命令均会在内部乘以该值：<br>因子> 1: 轮廓被放大<br>因子< 1: 轮廓被缩小<br>系数 = 1: 轮廓保持不变<br>缩放本身不会释放移动运动；它在下一次行进信息之前不会激活。 |

表格 6-140:     Scale(...), SCL(...) 的语法

示例:

|                  |                                      |
|------------------|--------------------------------------|
| :                |                                      |
| N30 SCL(X3,Y0.5) | 缩放打开                                 |
| :                | 所有的后续编程的 X 坐标均乘以“3”，同时 Y 坐标均乘以“0.5”。 |

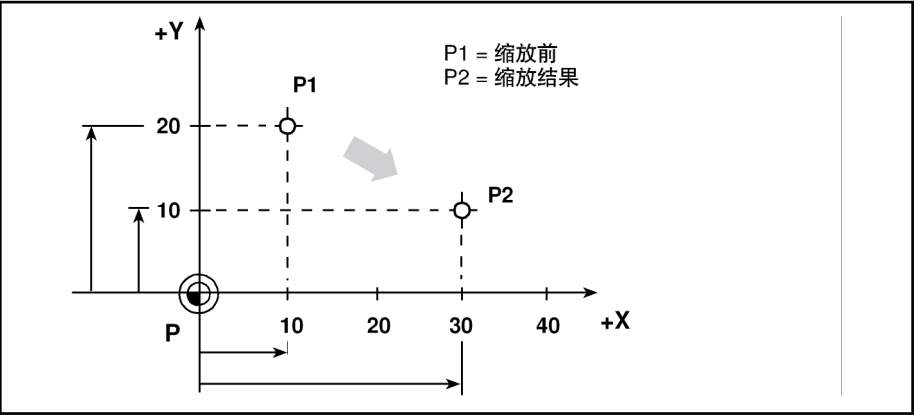


插图 6-34:     Scale(...), SCL(...) 的示例

- 特殊功能和限制:
- 不允使用许负的缩放系数。
  - 圆形平面所有相关坐标的缩放系数对于圆形 / 螺旋 / 螺旋 N 插补必须相同！否则会生成错误消息。
  - 该函数同样会影响 I、J、K 的插补参数，以及 R 地址的量（用于半径编程的）。
  - 它影响着可编程的轮廓切换。
- 请参阅 第 6.114 章 “编程的轮廓移位 Shift、SHT” 第 324 页 。
- 它 不 影响:
    - 进给编程或激活的进给、

- 零点偏置 (G54-G59.5; 参见 第 5.2.21 章 “零点偏置 (Z0) G53、G53.1-G59.1 至 G53.5-G59.5” 第 135 页 )、
- 程序坐标偏置 (Trans 或 ATrans; 参见 第 6.148 章 “程序坐标偏置 Trans、TRS, 叠加程序坐标偏置 ATrans、ATR” 第 361 页 )、
- 程序位置的设置 (“SetPos”; 参见 第 6.113 章 “设置程序位置 SetPos、SPS” 第 323 页 )、
- 接近的参考点坐标 (G74; 参见 第 5.2.26 章 “接近参考点坐标 G74 ” 第 141 页 )、
- 接近的固定机床轴位置坐标 (G76; 参见 第 5.2.29 章 “接近固定的机床轴位置 G76” 第 143 页 )、
- 铣削半径和刀具长度修正值。

## 6.111 选定的附加坐标耦合

### 6.111.1 选定的附加坐标耦合 "SelCrdCouple、SCC"

**描述:** 将当前通道 (目标) 的工件坐标耦合到另一个通道 (源) 的工件坐标。从而源坐标值就对目标坐标值具有叠加效果。

由此, 当前通道中的运动可由另一个通道中编程的运动叠加。

|     |                                                               |                                                  |
|-----|---------------------------------------------------------------|--------------------------------------------------|
| 语法: | SelCrdCouple (SC<Channel>,<br>CL(<Q1>,<Z1>{{,<Qn>,<Zn>}}...)) |                                                  |
|     |                                                               | 选择性叠加坐标耦合打开                                      |
|     | SelCrdCouple (0) 或<br>SelCrdCouple ()                         | 所有激活的坐标耦合关闭                                      |
|     | 缩写形式: SCC ( .. )                                              |                                                  |
|     | 其中                                                            |                                                  |
|     | <Channel>                                                     | <Qn>所在通道的编号。<br>输入值: 整数<br><Channel>必须小于当前频道的号码。 |
|     | <Qn>                                                          | 源<br><Channel> 中坐标的逻辑名称或逻辑编号。                    |
|     | <Zn>                                                          | 目标<br>当前通道中坐标的逻辑名称或逻辑编号。                         |

表格 6-141: SelCrdCouple (SCC) 语法

**示例:**

SCC (SC1,CL (YA, 将当前通道中的坐标 YB 与通道 1 的坐标 YA 进行耦合。  
YB) )

SCC (SC1,CL (1,1 将当前通道中逻辑编号为 1 和 2 的坐标与通道 1 中逻辑编号  
,2,2)) 为 1 或 2 的坐标进行耦合。

### 6.111.2 选定的使用表进行的附加坐标耦合 "SelCrdCoupleTab、SCCT"

**描述:** 将当前通道 (目标) 的工件坐标与另一个或两个通道 (源) 的一个或两个工件或机床坐标耦合。从而源坐标值就对目标坐标值具有叠加效果。由此, 当前通道中的运动可由另一个或其他两个通道中编程的运动叠加。

关于本主题的更多的信息, 参见 “IndraMotion MTX 函数说明” “耦合” 一章 “选定的使用表进行的附加坐标耦合” 一节。

使用高级语言语法的 NC 功能

|     |                                                                                                                                                              |                                               |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| 语法: | SelCrdCoupleTab (DE (<Name>, <Fact>), SO (<Channel>, <Name>, <Fact>, <Tab>, <Offs>, WCS MCS)<br>{, SO2 (<Channel>, <Name>, <Fact>, <Tab>, <Offs>, WCS MCS)}) |                                               |
|     | 缩写形式: SCCT                                                                                                                                                   |                                               |
|     | 其中                                                                                                                                                           |                                               |
|     | DE (目标)                                                                                                                                                      |                                               |
|     | <Name>                                                                                                                                                       | 目标坐标的名称                                       |
|     | <Fact                                                                                                                                                        | 目标坐标的缩放系数 (为 1.0, 另外编程的除外)                    |
|     | SO (源)                                                                                                                                                       |                                               |
|     | <Channel>                                                                                                                                                    | 源通道的编号                                        |
|     | <Name>                                                                                                                                                       | 源坐标的名称                                        |
|     | <Fact>                                                                                                                                                       | 耦合系数 (除非已编程, 否则为 1.0)                         |
|     | <Tab>                                                                                                                                                        | 耦合表的名称                                        |
|     | <Offs>                                                                                                                                                       | 表访问的偏置值 (为 0.0, 另有编订的除外)                      |
|     | WCS MCS                                                                                                                                                      | 源的坐标系:<br>WCS = 工件坐标<br>MCS = 机床坐标<br>默认: WCS |
|     | SO2 (源)                                                                                                                                                      | 于 SO 类似                                       |

表格 6-142: SelCrdCoupleTab (SCCT) 语法  
可为不同的坐标多次编程 SCCT 函数。

|     |                                                                     |                                |
|-----|---------------------------------------------------------------------|--------------------------------|
| 示例: | SCCT (DE (YB), SO (1, YA), YB 取自 YA 位置和 YC 位置编程的总和。<br>SO2 (1, YC)) |                                |
|     | SCCT (DE (YB, 0), SO (1, YA)) by 与 YA 相等。编程的无效。                     |                                |
|     | SCCT (DE (ZB),                                                      | 使用耦合表 "SCCTab.fct", 将通道 1 中的坐标 |
|     | SO (1, ZA, , SCCTab.fct))                                           | ZA 与活动通道的坐标 ZB 耦合。             |
|     | 取消耦合:<br>SCCT ()<br>SCCT () 将打开使用 SCCT ( ... ) 生成的所有耦合。             |                                |

6.112 抑制冲突监控 "SCLN"

|     |                                                                                           |                                                                          |
|-----|-------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| 描述: | 功能 SCLN 局部抑制刀具半径校正的碰撞监测 CLN。因此, 在主动碰撞监控的情况下, 可以处理自包含的几何轮廓回路。如果碰撞监测遇到标记为 SCLN 的块, 则块预见被关闭。 |                                                                          |
| 语法: | SCLN [( <mode> )]                                                                         |                                                                          |
|     | 其中                                                                                        |                                                                          |
|     | <mode>                                                                                    | 1: 在标记的块上取消块碰撞监视的前瞻。<br>2: 要确定闭合轮廓的接近块: 碰撞监测的预测块在与碰撞块碰撞的块中被取消。<br>默认值: 1 |

表格 6-143: 语法 SCLN



|     |                                           |                                              |
|-----|-------------------------------------------|----------------------------------------------|
| 示例: | N100 CLN(DLA10)                           | 使用 10 个程序块进行碰撞监视，阻止预测。                       |
|     | N110 G0 X4.4645 Y-5.5355                  | 定位到开始位置。                                     |
|     | N120 G1 F750 Z5                           | 横向进给刀具轴                                      |
|     | N130 G1 F1250 G42 ED1<br>X6.4645 Y-5.5355 | 激活刀具半径修正                                     |
|     | N140 G2 R5 X5 Y0 SCLN(2)                  | 切向接近圆形至轮廓。碰撞 N190。SCLN(2) 导致在块 N190 中取消碰撞监控。 |
|     | N150 G1 X10 Y0                            | 开始轮廓                                         |
|     | N160 X10 Y10                              |                                              |
|     | N170 X0 Y10                               |                                              |
|     | N180 X0 Y0                                |                                              |
|     | N190 X6 Y0                                | 轮廓终止                                         |
|     | N200 G2 R5 X8.5355 Y-3.5355               | 轮廓的切向离开                                      |
|     | N210 G40 G1 X10.85355<br>Y-5.5355         | 停用刀具半径修正                                     |

特殊功能和限制: SCLN 功能在块到块的基础上有效。

6.113 设置程序位置 "SetPos、SPS"

描述: 在不触发轴运动的情况下，将当前程序的零点（该零点基于当前的程序坐标系和激活的零点）设置为对应的编程值。  
之后，将自动显示新的位置值。

|     |                       |                                        |
|-----|-----------------------|----------------------------------------|
| 语法: | SetPos(<Coordinates>) | 为 <Coordinates> 中编程的轴设置程序零点。           |
|     | SetPos                | 解除 SetPos 启动的所有偏置。                     |
|     | 缩写形式: SPS(..)         |                                        |
|     | 其中                    |                                        |
|     | <Coordinates>         | 将轴的值设置为给定的坐标。使用逗号分隔给定的坐标（例如: (X0,Y0)）。 |

表格 6-144: SetPos (SPS) 语法

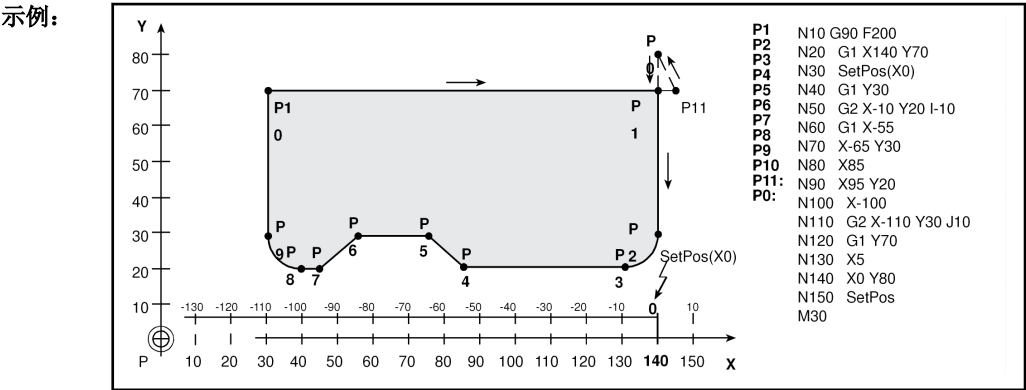


插图 6-35: 示例: 设定程序位置

特殊功能和限制: 可使用机床参数设置各通道在控制器复位后是保留还是清除 SetPos 偏置。

使用高级语言语法的 NC 功能

6.114 编程的轮廓移位 "Shift、SHT"

**描述：**“切换”函数是输入工具的一种。  
该控制器可对与程序坐标系的轴平行的编程的轮廓进行移位。

 “Shift” 是一个输入工具，因此它不会更改当前的程序坐标系。输入工具只为程序坐标提供其他输入方法。

- 以下情况适用：
- 应用编程的偏移坐标，直到它们被新的换挡程序块覆盖或禁用。
  - 若 “Shift” 自行进行编程，则不会产生进给运动；但同一块的编程中可能会包含进给信息。

 错误的编程可能会导致工件和机床损坏！  
“Shift” 受 “Mirror”、“Scale” 和 “Rotate” 的影响；即在 “Shift” 块中进行编程的新轮廓零点的坐标也可进行镜像、缩放或旋转。

|     |                         |                                                 |
|-----|-------------------------|-------------------------------------------------|
| 语法： | Shift (<Coordinates>)   | 轮廓切换                                            |
|     | Shift (0) 或<br>Shift () | 禁用轮廓切换                                          |
|     | 缩写形式：SHT (..)           |                                                 |
|     | 其中                      |                                                 |
|     | <Coordinates>           | 移位的轮廓零点的程序坐标。在存在多个坐标时需使用逗号进行分隔（例如：SHT (x5,y2)）。 |

表格 6-145: shift (SHT) 语法

**示例：**

```
N10 SHT (X10,Y10,Z50) X10, Y10, Z50 上的新轮廓零点
:
N100 G1 X...Y...Z... 考虑偏置的行进轴。
:
N110 SHT (X20,Y20) X20, Y20, Z50 上的新轮廓零点（保留 Z 移位！）
:
N210 SHT () X...Y...Z... 取消移位；轴行进至编程的位置。
:
```

6.115 速度限制 "SMin、SMN", "SMax、SMX"

**描述：**在加工过程中定义范围，在直接速度编程（G97）和常量切削速度（G96；参见第 5.2.42 章 “常量切削速度 G96、直接速度编程 G97” 第 155 页）的情况下，主轴速度在该范围中可能会变化。所定义的速度范围适用于所有档位级别。激活速度限制时，所有速度信息均限制为设定的值。  
系统仅在速度（包括由主轴电位计引起的速度）处于给定的速度范围内时才会执行速度修改。

|     |                  |                            |
|-----|------------------|----------------------------|
| 语法： | SMin (S<speed1>) | 激活 <Speed1>，将其作为许可速度范围的下限。 |
|     | SMin (S0)        | 取消第一主轴或主轴组的下限。             |

|                  |                                       |
|------------------|---------------------------------------|
| 缩写形式: SMN (..)   |                                       |
| SMax (S<speed2>) | 激活 <Speed2>, 将其作为许可速度范围的上限。           |
| SMax (S-1)       | 取消第一主轴或主轴组的上限。                        |
| 缩写形式: SMX (..)   |                                       |
| 其中               |                                       |
| <Speed1>         | 允许的最低速度<br>取值范围: > 0<br>必须小于 <Speed2> |
| <Speed2>         | 允许的最高速度<br>取值范围: > 0<br>必须大于 <Speed1> |

表格 6-146: SMin、SMN、SMax 和 SMX 语法

通用语法的形式如下:

SMin(S1=Value1 {, S2=Value2} ...) 或 SMax(S1=Value1 {, S2=Value2} ...).

|     |                               |                          |
|-----|-------------------------------|--------------------------|
| 示例: | N50 X..Y..SMN(S1500)          | 主轴速度必须在 1500 至 2500 转之间。 |
|     | N60 X..Y..SMX(S2500)          |                          |
|     | :                             |                          |
|     | N90 X..Y..SMN(S0)<br>SMX(S-1) | 停用下上限<br>(= 速度限制关闭)      |

特殊功能和限制: 只有编程的速度极限值在档位范围极限之内时, 速度限制才有效。

### 6.116 调整块中的主轴预运行初始化调整程序 "SpAdjSp"

|          |                                                                                                                                                                                                     |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:      |  有关“轴转换”功能性的详细信息, 请参阅文档“IndraMotion MTX 功能描述”。                                                                   |
| 语法:      | <div>SpAdjSp</div>                                                                                                                                                                                  |
| 示例:      | <p>表格 6-147: Syntax SpAdjSp</p> <p>从 InitAdjPrg 中摘录</p> <pre>; Produce channel spindles and spindle groups ; Also sends aux. functions CSPx to the PLC! N100 SpAdjSp ;</pre>                        |
| 特殊功能和限制: | <ul style="list-style-type: none"><li>功能“SpAdjSp”只能用于 Init 调节程序 (参见 SD.SysSRun.NameOfInitAdjPrg).</li><li>如果定义了通道专用整数辅助功能 CSP1 至 CSP8, 则会通知 PLC 当前的通道主轴分配-就像 Get 主轴、Rem 主轴以及 Def 主轴的情况一样。</li></ul> |

### 6.117 确定 NC 重启调整程序的主轴语法 "SpAdjStr"

|     |                                                                                                                                   |
|-----|-----------------------------------------------------------------------------------------------------------------------------------|
| 描述: |  有关“轴转换”功能性的详细信息, 请参阅文档“IndraMotion MTX 功能描述”。 |
|-----|-----------------------------------------------------------------------------------------------------------------------------------|

使用高级语言语法的 NC 功能

在调整程序中被调用时，函数“SPADJSTR” <SpindleNumber> 填充“SD.SysChSpSRun[]”系统数据的以下组件：

- SpeedStr
- MoveStr
- GearStr

调用后，SpeedStr、MoveStr 和 GearStr 包含块预运行终止时编程主轴的有关辅助功能。

|     |                                             |                 |
|-----|---------------------------------------------|-----------------|
| 语法： | SpAdjStr<spindle>                           |                 |
|     | 系统数据的组件以数控重启结束的有效字符串填充（例如 S1=, M103, M141）。 |                 |
|     | 其中                                          |                 |
|     | <Spindle>                                   | （通道）主轴的编号 1...8 |

表格 6-148: SpAdjStr 语法

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                           |  |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 示例： | 从 InitAdjPrg 中摘录                                                                                                                                                                                                                                                                                                                                                                                                                          |  |
|     | <pre>; ;   Specify speed or cutting velocity ; 0000 DIM SADR\$(8) :REM For spindle syntax      (e.g., S1=) ; 0100  FOR SP% = 1 TO SD.SysChSpSRun.MaxSpNumber 0102    IF SD.SysChSpSRun.Sp[SP%].SpeedNotAdj &lt;&gt; 0 THEN ; N104      SpAdjStr[SP%] ;Load the syntax elements of the current spindle 0106      SADR\$ = SD.SysChSpSRun.SpeedStr ; N108      [SADR\$][SD.SysChSpSRun.Sp[SP%].ProgSpeed] 0110 ENDIF 0112  NEXT SP% ;</pre> |  |

- 特殊功能和限制：
- SpAdjStr 函数仅可用于调整程序（参阅 SD.SysSRun.NameOfAsup）。NameOfInitAdjPrg und SD.SysSRun.NameOfAsup）。
  - 调用“SpAdjStr”时，必须调整主轴配置（参见“SpAdjSp”）。

6.118 启动 / 转换保留的主轴 "SpAdmin、SPA"

描述：

- 启用目前保存于当前通道中的主轴，同时无需停止旋转中的主轴，
- 之后将其转换至其他任意通道。



也可用“主轴停止”函数来启用主轴（参见 第 5.3.6 章“主轴停止 M5、M105、M205”第 170 页）。

|     |                                               |                                |
|-----|-----------------------------------------------|--------------------------------|
| 语法： | SpAdmin (S<Num>=<Mode>{ , S<Num>=<Mode>}... ) |                                |
|     | 缩写形式：SPA (... )                               |                                |
|     | 其中                                            |                                |
|     | <Num>                                         | 主轴编号（主轴索引）<br>输入范围：1...8<br>整数 |
|     | <Mode>                                        | 0: 启用主轴<br>1: 保留并转换已启用的主轴      |

表格 6-149: SpAdmin (SPA) 语法

示例:

N60 SPA (S1=0, S2=0)1 和 2 启用主轴  
:

特殊功能和限制:

只有在 MP7060 00020 中的正确位置输入“主轴停止”命令，才能通过控制器复位和 M30 激活当前通道所保留的主轴。

### 6.119 定义（激活）耦合主轴组，移除（取消）耦合主轴组"SpCoupleConfig、SPCC"



有关更多信息，请参阅“位置同步主轴操作”章节中的“IndraMotion MTX 功能说明基本功能”。

描述:

- 定义一个耦合主轴组，并激活其主轴耦合。  
控制器会自动将相关的主轴驱动器切换至位置接口。
- 将从动主轴添加至现有的耦合主轴组，或将其从现有的耦合主轴组中删除。  
所添加的主轴驱动器会自动切换至位置接口；被删除的主轴驱动器将切换至速度接口（如果耦合前速度接口在受影响的主轴上处于活动状态）。
- 指定从动轴到与主轴耦合的旋转方向。
- 关闭一个耦合组的主轴耦合，从而关闭整个耦合组。  
若速度接口在耦合前的特定主轴上处于活动状态，所有的相关主轴驱动器均将切换至速度接口。

从动主轴位置的耦合规律是：

$$p_s = p_m * k + o_{dist} + o_{offs}$$

其中

|            |                      |
|------------|----------------------|
| $p_s$      | 从主轴指令位置              |
| $p_m$      | 主轴指令位置               |
| $k$        | 从主轴到主主轴的旋转方向（1，-1）   |
| $o_{dist}$ | 主轴和从轴之间的预设耦合距离（SPCD） |
| $o_{offs}$ | 的附加动态偏移量与耦合距离（SPCP）  |

语法:

|                                                                                          |                               |
|------------------------------------------------------------------------------------------|-------------------------------|
| 定义耦合组                                                                                    |                               |
| SpCoupleConfig (CP=<Group>, MA=<Master>, S<Sequence>=1 -1 { {, S<Sequence>=1 -1} ... } ) |                               |
| 向耦合主轴组中添加从动主轴，或从中删除从动主轴                                                                  |                               |
| SpCoupleConfig (CP=<Group>, S<Sequence>=<Mode> { {, S<Sequence>=<Mode>} ... } )          |                               |
| 解决耦合组                                                                                    |                               |
| SpCoupleConfig (CP=<Group>, MA=0)                                                        |                               |
| 缩写形式: SPCC (...)                                                                         |                               |
| 其中                                                                                       |                               |
| <Coupling>                                                                               | 耦合主轴组的编号<br>输入范围: 1...4<br>整数 |

使用高级语言语法的 NC 功能

|          |                                                                                        |
|----------|----------------------------------------------------------------------------------------|
| <Master> | 主轴的编号（主轴索引）<br>输入范围：1...8<br>整数                                                        |
| <结果>     | 从动主轴的编号（主轴索引）<br>输入范围：1...8<br>整数                                                      |
| <Mode>   | 0: 从 <link> 拆下从动主轴<br>1: 将主动主轴旋转方向的从动主轴添加到 <group> 中<br>-1: 在主动主轴的相反旋转方向将从动轴添加到 <link> |

表格 6-150:     *SpCoupleConfig (SPCC) 语法*

示例:     *CoupleAndDecouple*

```
N010 SPCD(S3=0) ;coupling distance 0°
...
N210 SPCC(CP=1, MA=1, S3=1) ;activate coupling 1
N220 SPC_WAIT(CP=1) ;wait for synchronous mode
...
...
N310 M103 S1=2000
...
...
N760 SPCC(CP=1, MA=0) ;deactivate coupling 1
N770 SPC_WAIT(CP=1) ;wait for end of decoupling
...

```

## 6.120     从动主轴的耦合距离 "SpCoupleDist、SPCD"

**描述:**     为“主轴耦合”函数耦合时（耦合主轴组创建时），配置主动主轴和从动主轴间的位置迟滞。

若不在所需的从动主轴上使用该函数，则它到主动主轴的位置延迟为 0 度（耦合时）。



若希望在主轴耦合活动期间取消主动和从动主轴间的附加角位移，请参见函数“SpCouplePosOffs”，第 6.122 章“带活动耦合的角度偏置 SpCouplePosOffs、SPCP”第 329 页。

|     |                                                          |                                                                                                                                                                                               |
|-----|----------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | SpCoupleDist (S<Num>=<Distance>{, S<Num>=<Distance>}...) |                                                                                                                                                                                               |
|     | 缩写形式: SPCD (...)                                         |                                                                                                                                                                                               |
|     | 其中                                                       |                                                                                                                                                                                               |
|     | <Num>                                                    | 从动主轴的编号（主轴索引）<br>输入范围：1...8<br>整数                                                                                                                                                             |
|     | <Distance>                                               | 从动轴的耦合距离: <ul style="list-style-type: none"><li>定位主动和从动主轴间的延迟（单位：度）。<br/>取值范围: 0° ≤ 位置延迟 &lt; 360°<br/>若编程其他值，会自动转换为给定区间。</li><li><b>ACT</b> 用于当前位置的耦合。<br/>这可以避免主轴和从轴停止时的不必要的补偿运动。</li></ul> |

表格 6-151:     *SpCoupleDist (SPCD) 语法*

示例

请参阅“定义联轴器组（激活），解决联轴器组（停用）”。

特殊功能和限制:

仅可使用当前不属于耦合组的主轴来编程此函数。

6. 121 同步模式错误窗口 "SpCoupleErrWin、SPCE"

描述:

为“主轴耦合”函数配置从动主轴的目标与实际值间允许的最大位置偏差。  
若在主轴耦合活动期间，位置偏差未超过所定义的间隔，则发送基于主轴的输出信号“同步模式 2”。  
若未在所需的从动主轴中使用该函数，则其同步模式错误窗口为 +/-10 度。



另请参见函数“SpCoupleSyncWin”，第 6.124 章“同步模式窗口 SpCoupleSyncWin、SPCS”第 330 页。

语法:

|                                                      |                                             |
|------------------------------------------------------|---------------------------------------------|
| SpCoupleErrWin(S<Num>=<Window>{,S<Num>=<Window>}...) |                                             |
| 缩写形式: SPCE(...)                                      |                                             |
| 其中                                                   |                                             |
| <Num>                                                | 从动主轴的编号（主轴索引）<br>输入范围: 1...8<br>整数          |
| <Window>                                             | 与目标值间允许的最大偏差值（单位: 度）。<br>输入范围: 0 至 359.9999 |

表格 6-152: SpCoupleErrWin (SPCE) 语法

示例

请参阅“同步窗口“SpCoupleSyncWin, SPCS””。

特殊功能和限制:

仅可使用当前不属于耦合组的主轴来编程此函数。

6. 122 带活动耦合的角度偏置 "SpCouplePosOffs、SPCP"

描述:

在激活的主轴耦合期间，在主动和从动主轴之间缩回一个角度偏置。可以选择性地编程要取消其角度偏置的主动和从动主轴间的相对速度。



在旋转期间，控制器会复位特定于主轴的输出信号“同步模式 1”。

语法:

|                                                                             |                                                                                         |
|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|
| SpCouplePosOffs(S<num>=<offset>{{,S<num>=<offset>}...}<br>{,POSVEL<Speed>}) |                                                                                         |
| 缩写形式: SPCP(...)                                                             |                                                                                         |
| 其中                                                                          |                                                                                         |
| <Num>                                                                       | 从动主轴的编号（主轴索引）<br>输入范围: 1...8<br>整数                                                      |
| <Offset>                                                                    | 主动和从动主轴间的绝对旋转角度（单位: 度）。<br>取值范围: -3600° ... +3600°                                      |
| <Speed>                                                                     | 要取消其角度偏置的主动和从动主轴间的相对速度。<br>单位和默认值取决于驱动器参数 S-0-0222。<br>输入后，<Speed> 在再次调用函数被修改前，始终储存在内部。 |

表格 6-153: SpCouplePosOffs (SPCP) 语法

示例:

使用高级语言语法的 NC 功能

AngularOffset

```
N010 SPCD(S3=0) ;coupling distance 0°
...
N210 SPCC(CP=1, MA=1, S3=1) ;activate coupling 1
N220 SPC_WAIT(CP=1) ;wait for synchronous mode
...
...
N310 M103 S1=2000
...
...
N410 SPCP(POSVEL20, S3=195) ;rotate S3 in relation to S1
N420 SPCP_WAIT(CP=1) ;wait for end of rotation
...
```

特殊功能和限制： 角度偏置被添加至可能已配置的耦合延迟（参见 第 6.120 章 “从动主轴的耦合距离 SpCoupleDist、SPCD” 第 328 页 ）。

6.123 等待角度偏置 "SpCouplePosOffs\_Wait、SPCP\_WAIT"

描述： 在机床上实际已缩回编程的耦合主轴组角度偏置（SPCP；参见 第 6.122 章 “带活动耦合的角度偏置 SpCouplePosOffs、SPCP” 第 329 页 ）前，暂停零件程序。

|     |                                     |                              |
|-----|-------------------------------------|------------------------------|
| 语法： | SpCouplePosOffs_Wait(CP=<Coupling>) |                              |
|     | 缩写形式：SPCP_WAIT(...)                 |                              |
|     | 其中                                  |                              |
|     | <Coupling>                          | 耦合主轴组的编号<br>输入范围：1...4<br>整数 |

表格 6-154: SpCouplePosOffs\_Wait (SPCP\_WAIT) 语法

示例 “主动耦合时请参阅相位偏移 “SpCouplePosOffs, SPCP””。

6.124 同步模式窗口 "SpCoupleSyncWin、SPCS"

描述： 为 “主轴耦合” 函数配置从动主轴的目标与实际值间允许的最大位置偏差。

- 在执行/更改耦合时的作用（同步阶段的起始处）：  
零件程序在位置偏差处于定义的间隔中前会保持等待状态。
- 执行耦合时的作用：  
若位置偏差在定义的间隔内，则发送基于主轴的输出信号 “同步模式 1”。

若未在所需的从动主轴中使用该函数，则其同步窗口为 +/-1 度。

 有关耦合活动时的其他监控，请参见函数 “SpCoupleErrWin, ” 第 6.121 章 “同步模式错误窗口 SpCoupleErrWin、SPCE” 第 329 页 。

|     |                                                        |  |
|-----|--------------------------------------------------------|--|
| 语法： | SpCoupleSyncWin(S<Num>=<Window>{, S<Num>=<Window>}...) |  |
|     | 缩写形式：SPCS(...)                                         |  |
|     | 其中                                                     |  |



|          |                                     |
|----------|-------------------------------------|
| <Num>    | 从动主轴的编号（主轴索引）<br>输入范围：1...8<br>整数   |
| <Window> | 与目标值间允许的最大偏差值（单位：度）。<br>输入范围：0 至 20 |

表格 6-155:      *SpCoupleSyncWin (SPCS) 语法*

示例: *ConfigureAndCouple*

```
;Configure slave spindle
N010 SPCD(S3=0) ;coupling distance 0°
N020 SPCS(S3=0.5) ;synchronous window 0.5°
N030 SPCE(S3=3.0) ;error window 3°
...
N210 SPCC(CP=1, MA=1, S3=1) ;activate coupling
N220 SPC_WAIT(CP=1) ;wait for synchronous mode
...
...
```

特殊功能和限制:      仅可使用当前不属于耦合组的主轴来编程此函数。

### 6.125      等待同步模式 "SpCouple\_Wait、SPC\_WAIT"

描述:      在成功地创建、重新配置或删除耦合的主轴组之前，零件程序保持等待状态（SPCC；参见 第 6.119 章“定义（激活）耦合主轴组，移除（取消）耦合主轴组 SpCoupleConfig、SPCC”第 327 页）。

|                               |                              |
|-------------------------------|------------------------------|
| 语法:                           |                              |
| SpCouple_Wait (CP=<Coupling>) |                              |
| 缩写形式: SPC_WAIT(...)           |                              |
| 其中                            |                              |
| <Coupling>                    | 耦合主轴组的编号<br>输入范围：1...4<br>整数 |

表格 6-156:      *SpCouple\_Wait (SPC\_WAIT) 语法*

示例      请参阅“定义联轴器组（激活），解决联轴器组（停用）”。

### 6.126      定义/移除主轴组 "SPG. .、SPGALL"

描述:      若已激活“通道主轴”函数（MP 1040 00999 = 1），控制器启动时会自动预设主轴组。

第一通道主轴属于第一主轴组。

主轴组（也称为平行主轴）允许结合用于“档位范围”、“顺时针/逆时针旋转”、“主轴停止”和“主轴定向”的函数，简化多个主轴的编程。因此，单个锭子不必单独编程。



不得将“主轴组”函数与由两个或多个位置受控的主轴组成的“耦合主轴组”混淆。

尽管主轴组中的所有主轴能够以不同的速度运行，而耦合主轴组中的所有主轴始终为同步运行。

|                       |                           |
|-----------------------|---------------------------|
| 语法:                   |                           |
| SPG<Group>(<Numbers>) | 指定当前通道中需要进行组合以构成主轴组的通道主轴。 |
| SPG<Group>(0)         | 将当前通道中的组重置为默认设置。          |

使用高级语言语法的 NC 功能

|                                      |                                           |
|--------------------------------------|-------------------------------------------|
| SPG<Group>(-1)                       | 在当前通道中取消相应的主轴组。                           |
| SPGALL (0)                           | 在当前通道中将所有组复位为默认设置。                        |
| 其中                                   |                                           |
| <Group>                              | 主轴组的编号<br>输入范围：1 到 4<br>整数                |
| <Number>                             | 需要分配给相应主轴组的通道主轴的编号（使用逗号分隔）。<br>输入范围：1...8 |
| SPG(<numbers>) 是 SPG1 (<numbers>)的缩写 |                                           |

表格 6-157: SPG.. 语法 (SPGALL)

|     |                            |                                     |
|-----|----------------------------|-------------------------------------|
| 示例: | GSP (SSP01, S1, SSP07, S2) | 将系统主轴 1 和 7 分配给通道主轴 1 和 2。          |
|     | SPG1 (1,2)                 | 主轴组 1 由 S1 和 S2 组成。                 |
|     | M3                         | 通过顺时针旋转启动 S1 和 S2 (SSP01 和 SSP07) 。 |
|     | GSP (SSP14, S2)            | 将系统主轴 14 分配给主轴 2。                   |
|     | M4                         | 以反转方向启动 S1 和 S2 (SSP01 和 SSP14) 。   |
|     | RSP (S1)                   | 通道主轴 1(S1) 未分配系统主轴。                 |
|     | M3                         | 通过顺时针旋转启动 S2 (SSP14)。               |

- 特殊功能和限制:
- 传送主轴（GSP、RSP、DSP）时，通道主轴到系统主轴的分配将发生改变。此分配更改直接影响主轴组。
  - 在同一个块中编程的单主轴和主轴组之间的同步任务将导致运行时错误（示例：N10 M3 M104）。

## 6. 127 在系统数据队列中写入 "SSDQ"

描述:

“SSDQ”函数在系统数据（一个“Int\_t”系统数据域）中的队列中的 **执行时间** 写入编程的消息（一个整数）。在这种情况下，队列用作环缓冲。如果写入系统数据数组的最后一个元素，则函数在下一次调用时在队列的第一个元素中输入已编程的消息。在所有其他情况下，它被写入后面的元素。

为使用户了解已写入了哪些队列元素，“SSQD”函数同时会修改控制结构。该控制结构也位于系统数据中，并包括两个“Int\_t”元素 **LastFilled** 和 **LastGet**。

如果函数已经向队列写入编程的消息，设置 **LastFilled** 元素到队列中已修改条目的索引。该索引从一开始。

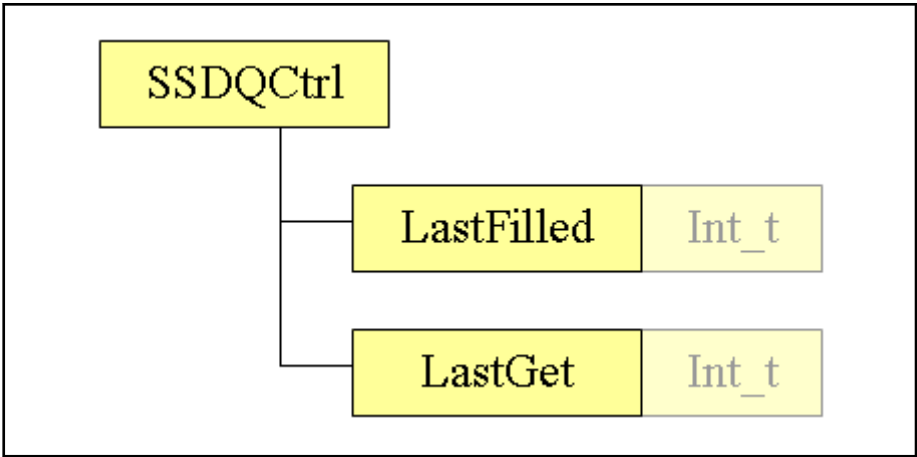


插图 6-36: SSDQ 控制结构的结构

编程人员可指定所用的系统数据队列和控制结构。为此，它们指定相应的结构为 XPath。两种系统数据结构可以是根元素并可以是更大结构的一部分。

 允许以通道相关的形式创建结构。如果在这种情况下省略通道索引，函数自动使用当前通道（零件程序在其中执行）的子结构。

如果未明确指定系统数据结构，则函数尝试使用 “/SSDQueue” 作为队列，“/SSDQCtrl” 作为控制结构。

在写入过程情况下，仅可为控制结构触发修改事件。队列被修改，但是一个改变事件没有被触发。

“SSDQ” 函数执行之后，用户负责从队列中读取消息。控件结构中的 **LastGet** 元素必须设置为最近读取的队列元素的索引。

如果用户读出控制结构且 **LastFilled** 和 **LastGet** 的值不同，则行新消息到达了该队列。如果 **LastFilled** 大于 **LastGet**，新消息所在队列的条目从 **LastGet+1** 至 **LastFilled**（按其被编程的顺序）。如果 **LastFilled** 小于 **LastGet**，队列的环缓冲特性将起作用。在这种情况下，新消息位于 **LastGet+1** 至队列末尾以及从队列开始至 **LastFilled**（按其被编程的顺序）。

如果用户无法按时成功设置 **LastGet**，则队列可能排满且消息会丢失。因此，“SSDQ” 函数检查队列中是否仍存在自由元素（下一写入位置不等于 **LastGet**）。如果不是这种情况，将产生运行时错误。

通过 “SSDQInit” 函数，可以初始化控制结构。对 “SSDQInit” 编程时，**LastGet** and **LastFilled** 在在执行时被设置为 1。

语法:

|                                                |                                             |
|------------------------------------------------|---------------------------------------------|
| SSDQ (M <Message> {, Q <XPath>} {, C <XPath>}) |                                             |
| 其中                                             |                                             |
| M <message>                                    | 整数，将被写入队列。                                  |
| Q <XPath>                                      | 系统数据的 XPath 或作为队列的子结构。<br>默认: “/SSDQueue”   |
| C <XPath>                                      | 系统数据的 XPath 或作为控制结构的子结构。<br>默认: “/SSDQCtrl” |
|                                                |                                             |

使用高级语言语法的 NC 功能

|                        |                                                 |
|------------------------|-------------------------------------------------|
| SSDQInit({C <XPath> }) |                                                 |
| C <XPath>              | 系统数据的 XPath 或被初始化作为控制结构的子结构。<br>默认: "/SSDQCtrl" |

表格 6-158: SSDQ 和 SSDQInit 语法

|                                  |                                                                                                                                     |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| 示例: SSDQ 和 SSDQInit              |                                                                                                                                     |
| N10 SSDQ (M32)                   | 通过控制结构 "/SSDQCtrl"将消息 "32"写入队列"/SSDQueue"                                                                                           |
| N20 SSDQ( M=17, Q / myQueue)     | 通过控制结构 "/SSDQCtrl" 将消息 "17" 写入队列 "/myQueue"                                                                                         |
| N30 SSDQ( M 42, C=/ myCtrl)      | 通过控制结构 "/myCtrl" 将消息 "42" 写入队列 "/SSDQueue"                                                                                          |
| 40 ST\$="/Msg[1,,3]/ Queue"      | 通过控制结构 "/SSDQCtrl" 将 1% 中的消息写入队列 "/Msg[1, CHANNELNO, 3]/Queue"                                                                      |
| N50 SSDQ (M[I%], Q=[ST\$])       | 例如, 如果程序在通道 5 中执行, 函数写入队列 "/Msg[1, 5, 3]/Queue"                                                                                     |
| N60 SSDQInit                     | 使用 <b>LastGet</b> =1 和 <b>LastFilled</b> =1 初始化控制结构 "/SSDQCtrl"                                                                     |
| N70 SSDQInit (C=/ Msg[1,,2]/Ctrl | 使用 <b>LastFilled</b> =1 和 <b>LastGet</b> =1 初始化控制结构 "/Msg[1, CHANNELNO, 2]/Ctrl"<br>例如, 如果块在通道 4 中执行, 函数写入控制结构 "/Msg[1, 4, 3]/Ctrl" |

- 特殊功能和限制:
- 队列必须至少包括三个元素。
  - 消息队列的大小取决于目标的剩余内存空间。
  - 如果通过 SSDQ 块编程的运动的执行短于一个 IPO 循环, 则存在 "过冲"。
  - 由于通过 NCS 至 Windows 的通信仅具有较低的优先权, 消息的读出可能存在延迟。

6. 128 系统主轴编程 "SSp. . . . ."

6. 128. 1 一般信息

以下函数考虑到系统主轴的启用。此类函数使用辅助函数来补充传统的主轴激活。  
缩写 SSp 表示 系统主轴。

6. 128. 2 启用/转换保留的主轴 "SSpAdm"

**描述:** 释放当前通道中由运动或定向设置所保留的一个或多个系统主轴。启动的主轴可转换至当前通道。  
该函数仅在用于对称时存在。我们建议使用 "SpAdmin" 通道函数。

|     |                                                                           |        |
|-----|---------------------------------------------------------------------------|--------|
| 语法: | SSpAdm、SSPADM                                                             |        |
|     | SSpAdm(<SSpNo.>   <SSpName>, <Mode>{ , <SSpNo.>   <SSpName>, <Mode>} ...) |        |
|     | 其中                                                                        |        |
|     | <SSpNo>                                                                   | 1.. 32 |

|           |                            |
|-----------|----------------------------|
| <SSpName> | 系统主轴名称                     |
| <Mode>    | 1 转换已释放的主轴<br>0 有条件地激活一根主轴 |

表格 6-159: SSpAdm 语法

示例:

SSpAdm (SSP05, 0, SSP08, 1)                      启用系统主轴 5，并转换已启用的系统主轴 8

6. 128. 3    关闭特定通道的主轴运动管理 "SSpAdmOff"

描述:        停用特定通道的系统主轴的主轴运动管理。该函数与机床参数 1040 00070 "主轴管理"的效果相同。

语法:

|                                                                                                          |                              |
|----------------------------------------------------------------------------------------------------------|------------------------------|
| SSpAdmOff、SSPADMOFF                                                                                      |                              |
| SSpAdmOff (<SSpNo.>   <SSpName>, <ChannelNo> {, <ChannelNo> } ... )<br>SSpAdmOff (<SSpNo.>   <SSpName> ) |                              |
| 其中                                                                                                       |                              |
| <SSpNo>                                                                                                  | 1..32                        |
| <SSpName>                                                                                                | 系统主轴名称                       |
| <ChannelNo>                                                                                              | 1..60: 通道号<br>0: 为来自位接口的任务启用 |

表格 6-160: SSpAdmOff 语法

示例:

SSpAdmOff (SSP05, 1, 2, 4)                      关闭系统通道 1、2、4 的系统主轴 5 的运动管理

SSpAdmOff (5)                                      将系统主轴 5 的运动管理复位为根据 MP 1040 00070 的状态

- 特殊功能和限制:
- 与其它 "SSp" 功能的模式不同，该功能一次只能影响一根系统主轴。
  - 如果从通道 0 进行指定，则从位接口指定该系统主轴运动管理的功能将关闭。

6. 128. 4    更改档位范围 "SSpGear"

描述:        更改一个或多个系统主轴的档位范围。

语法:

|                                                                                |                             |
|--------------------------------------------------------------------------------|-----------------------------|
| SSpGear、SSPGEAR                                                                |                             |
| SSpGear (<SSpNo.>   <SSpName>, <Level>{ , <SSpNo.>   <SSpName>, <Level>} ... ) |                             |
| 其中                                                                             |                             |
| <SSpNo>                                                                        | 1..32                       |
| <SSpName>                                                                      | 系统主轴名称                      |
| <Stage>                                                                        | 1..4 用于档位级别 1 至 4<br>8 用于空转 |

表格 6-161: SSpGear 语法

示例:

SSpGear (SSP05, 4)                      更改为系统主轴 5 的第 4 个档位范围

- 特殊功能和限制:
- NC 等待执行档位范围更改。

使用高级语言语法的 NC 功能

- 自动档位范围选择是一种通道属性。因此，不可使用“SSpGear”在自动和手动模式之间切换。
- 自动档位范围选择在通道中处于活动状态时不会受“SSpGear”的影响。即  
是说，若“M40”激活，下一个编程的 S 字可能执行换档。
- 档位切换仅支持 Sercos 主轴。

6. 128. 5 限制最大速度 "SSpMax"

描述：限制一根或多根系统主轴的速度上限。

|           |                                                                              |
|-----------|------------------------------------------------------------------------------|
| 语法：       | SSpMax、SSPMAX                                                                |
|           | SSpMax (<SSpNo.>   <SSpName>, <Speed>{ , <SSpNo.>  <SSpName>, <Speed>} ... ) |
|           | 其中                                                                           |
| <SSpNo>   | 1..32                                                                        |
| <SSpName> | 系统主轴名称                                                                       |
| <Speed>   | > 0 速度上限，单位转/分<br>-1 禁用速度限制                                                  |

表格 6-162: SSpMax 语法

- 示例：
- SSpMax (SSp05, 2000, SSp08, 3000)

将系统主轴 5 的速度上限设为 2000，系统主轴 8 的速度下限设为 3000 转/分。
- SSpMax (SSp05, -1, SSp08, -1)

取消系统主轴 5 和 8 的速度上限。

6. 128. 6 限制最小速度 "SSpMax"

描述：限制一根或多根系统主轴的速度下限。

|           |                                                                              |
|-----------|------------------------------------------------------------------------------|
| 语法：       | SSpMin、SSPMIN                                                                |
|           | SSpMin (<SSpNo.>   <SSpName>, <speed>{ , <SSpNo.>  <SSpName>, <speed>} ... ) |
|           | 其中                                                                           |
| <SSpNo>   | 1..32                                                                        |
| <SSpName> | 系统主轴名称                                                                       |

表格 6-163: SSpMin (SSPMIN) 语法

- 示例：
- SSpMin (5, 100, 8, 500)

将系统主轴 5 的速度设置为 100，系统主轴 8 的速度设置为 500 转/分。
- SSpMin (5, 0, 8, 0)

取消系统主轴 5 和 8 的速度下限。

6. 128. 7 位置 / 速度接口切换 "SSpMode"

描述：切换编程系统主轴的驱动器的速度和位置模式。

|         |                                                                              |
|---------|------------------------------------------------------------------------------|
| 语法：     | SSpMode、SSPMODE                                                              |
|         | SSpMode (<SSpNo.>   <SSpName>, <Mode> { , <SSpNo.>  <SSpName>, <Mode>} ... ) |
|         | 其中                                                                           |
| <SSpNo> | 1..32                                                                        |

|           |                        |
|-----------|------------------------|
| <SSpName> | 系统主轴名称                 |
| <Mode>    | 1: 激活位置接口<br>0: 取消位置接口 |

表格 6-164: SSpMode 语法

示例:

SSpMode (5, 1, 8, 0)      打开系统主轴 5 的位置接口，关闭系统主轴 8 的位置接口

特殊功能和限制:      主轴处于静止状态时将总是激活位置接口。

6. 128. 8    运动编程 "SSpMove (SSPM) "

描述:      启动或停止一个或多个系统主轴。

语法:

|                                                                           |                                        |
|---------------------------------------------------------------------------|----------------------------------------|
| SSpMove、SSPMOVE、SSPM                                                      |                                        |
| SSpMove (<SSpNo.>   <SSpName>, <Job>{ , <SSpNo.>  <SSpName>, <Job>} ... ) |                                        |
| 其中                                                                        |                                        |
| <SSpNo>                                                                   | 1..32                                  |
| <SSpName>                                                                 | 系统主轴名称                                 |
| <job>                                                                     | 3 - 主轴顺时针旋转<br>4 - 主轴逆时针旋转<br>5 - 主轴停止 |

表格 6-165: SSpMove (SSPM) 语法

示例:

SSpMove (SSP05, 3,            以顺时针旋转方式启动系统主轴 5，并停止系统主轴  
SSP08, 5)                      8。

6. 128. 9    主轴定向 "SSpOri (SSP0) "

描述:      编程的系统主轴移至编程的角度位置。

语法:

|                                                                              |                 |
|------------------------------------------------------------------------------|-----------------|
| SSpOri、SSPORI、SSPO                                                           |                 |
| SSpOri (<SSpNo.>   <SSpName>, <angle>{ , <SSpNo.>  <SSpName>, <angle>} ... ) |                 |
| 其中                                                                           |                 |
| <SSpNo>                                                                      | 1..32           |
| <SSpName>                                                                    | 系统主轴名称          |
| <Angle>                                                                      | 0.0 .. 359.9999 |

表格 6-166: SSpOri (SSP0) 语法

示例:

SSpOri (5, 20, 8, 0)      将系统主轴 5 的方向定为 20 度，将系统主轴 8 的方向定为 0 度。

特殊功能和限制:

- NC 等待方向运动的执行。
- 与通过 M19, M119, ...实现的方向运动相反，必须始终编程好角度。
- 若已使用 SSpMove 在同一数控块中编程了进给移动的输入，则忽略 SSpMove（无错误消息）。
- 通过设置接口信号 "qSp\_SValueSD"，借助系统数据 "SysSpCmdData[n]/OriPos" 进行位置分配。编程的规格被忽略。

使用高级语言语法的 NC 功能

6. 128. 10 速度编程 "SSpSpeed (SSPS) "

描述： 一个或多个系统主轴的速度设置。

|           |                                                                               |
|-----------|-------------------------------------------------------------------------------|
| 语法：       | SSpSpeed、SSPSPEED、SSPS                                                        |
|           | SSpSpeed(<SSpNo.>   <SSpName>, <Speed>{ , <SSpNo.>  <SSpName>, <Speed>} ... ) |
|           | 其中                                                                            |
| <SSpNo>   | 1..32                                                                         |
| <SSpName> | 系统主轴名称                                                                        |

表格 6-167: SSpSpeed (SSPS) 语法

|     |                                                                                |
|-----|--------------------------------------------------------------------------------|
| 示例： | SSpSpeed(SSp05, 1000, 将系统主轴 5 的速度设置为 1000，系统主轴 SSp08, 2000) 8 的速度设置为 2000 转/分。 |
|-----|--------------------------------------------------------------------------------|

- 特殊功能和限制：
- 默认情况下，编程的速度值解释为 rpm。
  - G96 激活时，所设定的速度将被视为切削速度（m/分钟）。
  - 控制器将始终限制已发送的速度目标值，从而遵从活动档位范围的限定值和编程的速度限制（参见 SMin、SMax、SSpMin 和 SSpMax）。
  - 在使用新的速度进行覆盖前将始终使用已设置的速度值（以模态方式作用）。
  - 通过设置接口信号“qSp\_SValueSD”，通过系统日期“SysSpCmdData[n]/Speed”进行速度指定。编程的规格被忽略。

6. 128. 11 特定于刀具的主轴限制"SSpToolLim"

描述： 在 /SysSpCmdData[<SysSp>]/Tool 系统日期中，在一个或多个系统主轴中启用输入的刀具特定主轴限制。



有关更多信息，请参阅“刀具特定的主轴限制”下“主轴功能章节中的” IndraMotion MTX 功能说明基本内容”。

|           |                                                                             |
|-----------|-----------------------------------------------------------------------------|
| 语法：       | SSpToolLim, SSPTOOLLIM                                                      |
|           | SSpToolLim(<SSpNo.>   <SSpName>, <0 1>{ , <SSpNo.>  <SSpName>, <0 1>} ... ) |
|           | 其中                                                                          |
| <SSpNo>   | 系统主轴编号                                                                      |
| <SSpName> | 系统主轴名称                                                                      |
| <0 1>     | 1 启用 /SysSpCmdData[]/Tool 的工具特定限制<br>0 禁用工具特定的限制                            |

表格 6-168: Syntax SSpToolLim

|     |                                   |
|-----|-----------------------------------|
| 示例： | SSpToolLim(3,1) 启用第三个系统主轴的刀具特定限制。 |
|-----|-----------------------------------|

- 特殊功能和限制：
- SSpToolLim 停止编程的主轴并随后重新启动它们。
  - SSpToolLim 报告 换 档 到 “SpindleWait , SPWAIT” 和 “SpindleWaitMode, SWM” 功能。
  - SSpToolLim 包括 NC 内部 WAIT 功能。



## 6.129 开启主轴的 C 轴模式 "SpindleToAxis、STA"

**描述:** 将在以下机床参数中输入的主轴切换到 C 轴模式:

- 作为 MP 1001 00001 (驱动器函数类型) 中的主轴/C 轴, 以及
- 作为 MP 1040 00001 中的 Sercos 主轴 (主轴类型选择)

对于加工, 该函数会将主轴转换为异步轴。  
其在显示中以异步轴形式出现, 最初位于 0 至 359.9999 度之间的任意位置。

 有关“轴转换”功能性的详细信息, 请参阅文档 “IndraMotion MTX 功能描述”。

|     |                                                |                            |
|-----|------------------------------------------------|----------------------------|
| 语法: | SpindleToAxis (<PAN> <PAI>{ ,<PAN> <PAI>}... ) |                            |
|     | 缩写形式: STA (...)                                |                            |
|     | 其中                                             |                            |
|     | <PAN>                                          | 物理轴名称<br>指定需要转换为 C 轴模式的主轴。 |
|     | <PAI>                                          | 物理轴索引<br>效果与 <PAN> 相同      |

表格 6-169: SpindleToAxis (STA) 语法

**示例:**

```
:
N030 物理轴 "CH" (即在轴运行期间名称为 "CH" 的主轴) 切换为异步轴。
STA (CH)
:
```

**特殊功能和限制:** 相同块中的轴位置必须始终根据 "SpindleToAxis(...)" 表达式进行编程。

## 6.130 开启主轴的 C 轴模式并等待 "SpindleToAxisWait、STAW"

**描述:** 将在以下机床参数中输入的主轴切换到 C 轴模式, 并等待直到 C 轴激活:

- 作为 MP 1001 00001 (驱动器函数类型) 中的主轴/C 轴, 以及
- 作为 MP 1040 00001 中的 Sercos 主轴 (主轴类型选择)

对于加工, 该函数会将主轴转换为异步轴。  
其在显示中以异步轴形式出现, 最初位于 0 至 359.9999 度之间的任意位置。

 有关“轴转换”功能性的详细信息, 请参阅文档 “IndraMotion MTX 功能描述”。

|     |                                                    |  |
|-----|----------------------------------------------------|--|
| 语法: | SpindleToAxisWait (<PAN> <PAI>{ ,<PAN> <PAI>}... ) |  |
|     | 缩写形式: STAW (...)                                   |  |
|     | 其中                                                 |  |

使用高级语言语法的 NC 功能

|       |                            |
|-------|----------------------------|
| <PAN> | 物理轴名称<br>指定需要转换为 C 轴模式的主轴。 |
| <PAI> | 物理轴索引<br>效果与 <PAN> 相同      |

表格 6-170:      *SpindleToAxisWait (STAW) 语法*

示例:

```
:

N030 STAW (CH) 物理轴 "CH"（即在轴运行期间名称为 "CH" 的主轴）切换为
 异步轴。

N040 CH90 异步轴 CH 定位至 90 度。

:
```

特殊功能和限制:

相同块中的轴位置必须始终根据 "SpindleToAxisWait(...)" 表达式进行编程。

6. 131

在主轴编程结束时等待主轴编程"SpindleWait, SPWAIT"

描述:

等待每个块结束时，直到各自的主轴达到所需的状态。

 有关更多信息，请参阅 "路径运动和主轴功能" 下的主轴功能" 章节中的 "IndraMotion MTX 功能说明文档"。

语法:

|                                           |                                                                                                                                            |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| SpindleWait (<Spindle> { , <Condition> }) |                                                                                                                                            |
| 缩写形式: SPWAIT (..)                         |                                                                                                                                            |
| 其中                                        |                                                                                                                                            |
| <Spindle>                                 | 它指定了应该等待的通道主轴。<br>-2: 除主轴外的所有主轴<br>-1 或 MSP: 主轴<br>0 或 ALL: 所有主轴<br>1 .. 8 或 S1...S8: 主轴的编号或名称                                             |
| <Condition>                               | 可选的。<br>等待条件列表，最大内容为 3, 5, 19, 40<br>3: 在 M3 或 M4 等待 "达到速度"<br>5: 等待主轴在 M5 处停止<br>19: 等待主轴在 M19 处停止<br>40: 等待档位切换结束<br>默认: 对于非编程条件，等待所有条件。 |
| SpindleWait                               |                                                                                                                                            |
|                                           | 没有参数列表的 SpindleWait 等待在此块中编程的主轴功能到达。                                                                                                       |

表格 6-171:      *SpindleWait (SPWAIT) 语法*

示例:

```
:

N30 M3 S1000 SPWAIT 它在块尾等待，直到通知 "速度到达"。

:

N260 M203 S2=500 G0 X1000 主轴 2 启动。
```

```
N270 G0 Y10 SPWAIT(S2,3)
N280 G1 ...
:
```

它在块尾等待，直到主轴 2 通知“速度到达”。

### 6.132 选择一个等待策略 "SpindleWaitMode, SWM"

**描述:** 它选择并激活功能“路径运动和主轴功能”的等待策略。



有关更多详细信息，请参见“路径运动和主轴功能”下的“主轴功能”一章中的“IndraMotion MTX 功能说明文档”以及章节“主轴”中的“IndraMotion MTX 机床参数文档”。

**语法:**

|                                                        |                                                                                                  |
|--------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| SpindleWaitMode(<Spindle> , <Switch on> {, <Strategy>) |                                                                                                  |
| 缩写形式: SWM(..)                                          |                                                                                                  |
| 其中                                                     |                                                                                                  |
| <Spindle>                                              | 它指定了策略应该改变的通道主轴。<br>-2: 除主轴外的所有主轴<br>-1 或 MSP: 主轴<br>0 或 ALL: 所有主轴<br>1 .. 8 或 S1...S8: 主轴的编号或名称 |
| <打开>                                                   | 0: 禁用等待策略<br>1: 启用等待策略                                                                           |
| <策略>                                                   | 可选的。<br>1 .. 3: 策略编号<br>1: 预定义策略<br>2 和 3: SP/SpAdmin/BlkSwiCtrl                                 |

表格 6-172: SpindleWaitMode (SWM) 句法

**示例:**

```
:
N010 SpindleWaitMode (MSP, 策略 2 被选择并激活主主轴。
1,2)
:
N150 SpindleWaitMode (MSP,0) 主轴的当前等待策略 (2) 被禁用。
:
N190 SpindleWaitMode (MSP,1) 主轴的当前等待策略 (2) 被启用。
:
```

### 6.133 使用样条的转角修圆

#### 6.133.1 使用样条的转角修圆（模态） "SplineCornering、SC0"

**描述:** 模态数控函数“使用样条的转角修圆”将把所有轮廓过渡修圆（使用样条），此类过渡的弯角等于或大于配置的最小弯角。

使用高级语言语法的 NC 功能

|           |                                    |                                                                                               |
|-----------|------------------------------------|-----------------------------------------------------------------------------------------------|
| 语法:       | SplineCornering (E<lag>)           |                                                                                               |
|           |                                    | 以指定的最大允许公差 E 来“打开圆角加工”。                                                                       |
|           | SplineCornering (D EF)             | 使用 MP 7050 00110 的公差 E 来“打开圆角加工”。                                                             |
|           | SplineCornering (L1=<L1>, L2=<L2>) |                                                                                               |
|           |                                    | 使用指定的前/后拐角距离 L1 或 L2 来“打开圆角加工”。                                                               |
|           |                                    | <b>SplineCornering (L&lt;L&gt;)</b><br>使用指定的前/后拐角距离 $L_1 = L_2 = L$ 来“打开圆角加工”                 |
|           |                                    | <b>SplineCornering (L&lt;L&gt;)</b><br>使用指定的前/后拐角距离 $L_1 = L_2 = L$ 来“打开圆角加工”                 |
|           |                                    | <b>SplineCornering (L&lt;L&gt;, E&lt;distance&gt;)</b><br>使用指定的公差 E 和 L 上同步的前/后拐角距离来“打开圆角加工”。 |
|           | SplineCornering (0 )               | “圆角加工功能关”                                                                                     |
| 缩写形式: SCO |                                    |                                                                                               |

表格 6-173: SplineCornering (SCO) 语法

- 特殊功能和限制:
- 会修圆通道中所有坐标（最多八个）的任何轮廓类型的拐角。
  - 模态数控函数“使用样条的转角修圆”可在局部进行超驰，即通过局部（非模态）数控函数“使用样条进行转角”实现到下一个块的过渡。
  - 相同的机床参数也适用于圆弧圆角加工。

6. 133. 2 使用样条的转角修圆（局部） "SplineCorneringLocal、SCOL"

描述: 对于非切向轮廓过渡，局部（非模态）数控函数“使用样条的转角修圆”使用样条对至下一数控块的过渡实现修圆。

|     |                                    |                                |
|-----|------------------------------------|--------------------------------|
| 语法: | SplineCorneringLocal (E<Distance>) |                                |
|     |                                    | 以指定的最大允许公差 E 进行转角修圆。           |
|     | SplineCornering (DEF)              | 使用 MP 7050 00110 的公差 E 进行转角修圆。 |
|     | SplineCornering (L1=<L1>, L2=<L2>) |                                |
|     |                                    | 使用指定的前/后拐角距离 L1 或 L2 进行转角修圆。   |
|     | SplineCornering (0)                | 至下一个数控块的过渡无转角修圆。               |
|     | 缩写形式: SCOL                         |                                |
|     |                                    |                                |

表格 6-174: SplineCornering (SCOL) 语法

- 特殊功能和限制:
- 会修圆通道中所有坐标（最多八个）的任何轮廓类型的拐角。
  - 模态数控函数“使用样条的转角修圆”可在局部进行超驰，即通过局部（非模态）数控函数“使用样条进行转角”实现到下一个块的过渡。
  - 相同的机床参数也适用于圆弧圆角加工。

6. 134 样条类型的定义 "SplineDef、SPD"

描述: IndraMotion MTX 支持以下样条类型:

- 样条类型 0:

- 带有系数编程的样条  
(CAD/CAM 系统的多项式系数)
- 样条类型 1:  
C1 常系数三次样条, 带支点编程以及切线规定 (如可行)  
(在数据点的切线过渡)
  - 样条类型 2:  
C2 常系数三次样条, 带支点编程  
(在数据点处的常量曲率转变)
  - 样条类型 3:  
B 样条, 带检查点编程  
(靠近支点的曲线形状)。
  - 样条类型 4:  
B 样条逼近  
(数据点周围的指定公差范围)。

使用 "SplineDef" (SDF) 函数选择并初始化所需的样条类型。之后, 使用 "G6" 函数来激活样条编程。

语法:

|                              |                     |
|------------------------------|---------------------|
| SplineDef(<Id>, {<Members>}) |                     |
|                              | 初始化样条类型             |
| 缩写形式: SPD( .. )              |                     |
| 其中                           |                     |
| <Id>                         | 样条类型的整数编号, 最大为 4 位数 |

表格 6-175: SplineDef (SPD) 语法

| 样条类型   | 位置            |                                       |                                                                 |               |
|--------|---------------|---------------------------------------|-----------------------------------------------------------------|---------------|
|        | 1000s         | 100s                                  | 10s                                                             | 1s            |
|        | 样条类型<br>0...4 | 参数化<br>0: 否<br>1: 等距<br>2: 弦<br>3: 向心 | 正切计算或规定<br>0: 否<br>1: Bessel<br>2: Akima<br>3: 错位<br>4: Hermite | 样条次数<br>1...5 |
| 样条类型 0 | 0             | 0                                     | 0                                                               | 1...5         |
| 样条类型 1 | 1             | 1...3                                 | 0...4                                                           | 1...5         |
| 样条类型 2 | 2             | 1...3                                 | 0                                                               | 3             |
| 样条类型 3 | 3             | 1...3                                 | 0                                                               | 1...5         |
| 样条类型 4 | 4             | 2                                     | 3                                                               | 3             |

表格 6-176: 样条类型概述

使用高级语言语法的 NC 功能

|           |                                                                                                                                                                                                                                     |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Members> | <b>不适用于样条类型 0 和/或 Hermite 样条！</b><br>样条运动中相关坐标/轴的名称列表。<br>在 <Members> 中非编程坐标/轴采用线性模式。<br>也可将取向坐标编程为样条。为此，请将定向指定为 "0" 或使用 <Members> 中的极坐标 "phi" 和 "theta"。<br>如果在<Members>中没有编程坐标，则应用以下激活值：<br>在激活的 5 轴转换情况下：x, y, z, 0<br>其它：所有通道成员 |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

表格 6-177: SplineDef (SPD) 语法 (续)

示例：

|                              |                                             |
|------------------------------|---------------------------------------------|
| N10 SplineDef (5)            | 使用样条阶数 5，初始化样条类型 0。                         |
| N15 SplineDef (1143)         | 使用样条阶数 3，初始化样条类型 1 Hermite 样条。              |
| N20 SplineDef (2203,X,Y,Z)   | 使用样条次数 3 和弦参数初始化样条类型 2。其中包括坐标 X、Y 和 Z。      |
| N30 SplineDef (3103,x,y,z,O) | 使用样条角度 3 和等距参数初始化样条类型 3。其中包括坐标 X、Y、Z 和定向坐标。 |
| N30 SplineDef (4233)         | 使用样条次数 3，初始化样条类型 4。涉及所有通道坐标。                |

6.135 可编程路径分割 "Split、SPT"

描述：如果编程的运动块超过一定长度，则将其分隔为数个部分路径。

|                  |                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------|
| 语法：              | Split ({<Mode>} {,<part length>})                                                                                            |
|                  | 缩写形式：SPT (...)                                                                                                               |
|                  | 其中                                                                                                                           |
| <Mode>           | 0: 功能没有效果<br>1: 该函数以模态形式作用<br>2: 该函数仅在已编程的块内有效<br>未编程：作用与为 0 时相同。                                                            |
| <Partial length> | 对于线性块：路径段长度<br>对于圆形块：弧长度<br>编程单位与轴坐标相同<br>如果未编程 <part length>，则使用以下内容：<br>激活的 <part length> 大致对应于可在二个后续插补循环内行进的最大路径，取决于当前进给。 |

表格 6-178: 拆分语法

示例：

|                       |                                            |
|-----------------------|--------------------------------------------|
| :                     |                                            |
| N20 Split (2,10) X100 | 对于 N20 开启，逐块进行有效路径分割                       |
| :                     | N20 中的进给运动被分隔为部分路径，每个的长度为 10 mm。           |
| N120 Split (1)        | 在 N120 处启动的模态有效路径分隔打开                      |
| :                     | 从 N120 开始的进给运动被分隔为路径区段，这些区段可在约 2 个插补循环内进给。 |
| :                     |                                            |

N220 Split(1,3.0)  
:  
N320 Split()  
:

在 N220 处启动的模态有效路径分隔打开  
从 N220 开始的进给运动分隔为路径区段，各区段的长度为 3 mm。  
可编程路径分隔关闭

- 特殊功能和限制:
- 只有长度未超过 <part length> 时，可编程路径分隔才会分隔进给块。
  - 若 <part length> 不是进给路径的整除数，则会单独进给其余量。
  - 若可编程分割在块到块基准激活 (<Mode>=2)，则必须在同一块编程进给运动。

6.136 主轴：位置和速度接口切换 "SpMode、SPM"

描述: 将主轴驱动器在速度和位置模式之间切换。

|        |                                             |
|--------|---------------------------------------------|
| 语法:    | SpMode (S<Num>=<Mode>{, S<Num>=<Mode>}... ) |
|        | 缩写形式: SPM (... )                            |
|        | 其中                                          |
| <Num>  | 主轴编号（主轴索引）<br>输入范围: 1...8<br>整数             |
| <Mode> | 0: 速度模式<br>1: 位置模式                          |

表格 6-179: SpMode (SPM) 语法

示例:

N60 SPM(S1=1,S2=1) 1 和 2. 将主轴切换到位置模式  
:

- 特殊功能和限制:
- 主轴修正停止。
  - 在切换至位置模式时会暂时停止旋转中的主轴。

6.137 静态刀具定向的参数设置 "StatToolOri、ST0"

描述: 使用 G47(...) 或 G78，刀具长度修正 L<sub>1</sub>、L<sub>2</sub> 和 L<sub>3</sub> 给赋值给 L<sub>1</sub>、L<sub>2</sub> 和 L<sub>3</sub> 坐标方向。这些坐标方向构成了一个坐标系，其中的刀具长度修正向量 L 包含分量 L<sub>1</sub>、L<sub>2</sub> 和 L<sub>3</sub>。

可使用 "StatToolOri" 命令，使 L 围绕坐标系的欧拉角 phi、theta 和 psi 旋转。

- 首先按照 Phi 角，以 L<sub>3</sub> 坐标方向旋转，
  - 然后围绕在 前一次旋转期间旋转的 L<sub>2</sub> 坐标方向的角度 Theta
  - 然后围绕角度 Psi 围绕坐标方向 在上一次旋转期间旋转 L<sub>3</sub>.

以下 L 的旋转具有相同的结果:

- 首先按照 Psi 角，以 L<sub>3</sub> 坐标方向旋转，
  - 然后围绕原始角度 Theta 旋转，而不是 旋转 L<sub>2</sub> 坐标方向
  - 然后围绕原始角度 Phi 旋转，而不是 旋转 L<sub>3</sub> 坐标方向

由于两种方法都可能导致相同的结果，请使用对您更有吸引力的方法。

静态刀具定向总是指反向刀具修正向量。因此，连续的静态工具取向不具有可加性。

使用高级语言语法的 NC 功能

|     |                                  |                                                                                                       |
|-----|----------------------------------|-------------------------------------------------------------------------------------------------------|
| 语法: | StatToolOri({<Φ> {,<Θ> {,<Ψ>}}}) |                                                                                                       |
|     | 缩写形式: STO( .. )                  |                                                                                                       |
|     | 其中                               |                                                                                                       |
|     | <Φ>,<Θ>,<Ψ>                      | 欧拉角 (单位: 度)<br>取值范围:<br>0° ≤ Φ < 360° ;<br>0° ≤ Θ ≤ 180° ;<br>0° ≤ Ψ ≤ 360° .<br>此范围以外的值将被自动转换至相应的间隔。 |

表格 6-180: StatToolOri (STO) 语法

围绕 L<sub>1</sub> 坐标方向绕 α 角旋转刀具修正矢量:

|                      |                             |
|----------------------|-----------------------------|
| L <sub>1</sub> 坐标方向: | StatToolOri(-90,<Alpha>,90) |
|----------------------|-----------------------------|

|     |                                                                                                                                                                                                                     |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 示例: | <pre>1 DCT(1,1,0) = 100 1 DCT(2,1,0) = 0 1 DCT(3,1,0) = 0 ; 刀具修正矢量的组件 G0 X0 Y0 Z0 ; (L1,L2,L3) G47(X,Y,Z) ED1 ; (100,0,0) STO(90,0,0) ; (0,100,0) STO(0,0,0) ; (100,0,0) STO(0,90,0) ; (0,0,-100) G48 ED0 M30</pre> |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|     |                                                                                                                                                                                                                                                                                                               |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 示例: | <pre>1 DCT(1,1,0) = 0 1 DCT(2,1,0) = 0 1 DCT(3,1,0) = 100 ; 刀具修正矢量的组件 G0 X0 Y0 Z0 ; (L1,L2,L3) G47(X,Y,Z) ED1 ; (0,0,100) STO(0,90,0) ; (100,0,0) STO(-90,45,90) ; ( 0 , -50*sqrt(2) , 50*sqrt(2) ) STO(0,30,0) ; ( 50 , 0 , 50*sqrt(3) ) STO(45,30,0) ; (25*sqrt(2),25*sqrt(2),50*sqrt(3)) G48 ED0 M30</pre> |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- 特殊功能和限制:
- 需使用“STO”来设置静态刀具定向的参数，并使用“G47”激活该参数（参见 第 5.2.19 章 “刀具长度修正 G47、G48” 第 132 页 ）。
  - 若“G47”在执行 STO 时已处于活动状态，则刀具定向的更改立即生效。



- 如果未使用 G47(...) 或 G78，将坐标名称分配至全部三个刀具长度修正 L<sub>1</sub>、L<sub>2</sub> 和 L<sub>3</sub>，则控制器在刀具定向时仅使用已分配的坐标，并生成警告。



另请参阅 “IndraMotion MTX Standard NC Cycles” 文档中的 “STP”。

## 6.138 抑制单步模式 "StepModeDisable, SMD"

**描述:** 单步抑制用于“单步”，“单块”，“程序块”和“除错步骤”之一的操作模式和单步完全地立行 NC 程序部分处理。这意味着块在没有确认 NC 启动的情况下被处理。

可以完全封装（异步）子程序和循环，即完成（异步）子程序或循环以一次 NC 启动进行处理，即使未在当前操作模式下计划。

单步模式抑制不是操作模式的改变。这意味着几何计算和速度计算在所选择的操作模式下执行（例如单块）。但是，等待 NC 启动信号被抑制。

可以单独控制单块 / 单步，程序块和调试块的功能效果。这意味着在操作模式“调试步骤”保持不动的情况下，可以激活单段 / 单步和程序块的 NC 启动抑制。

**编程:**

单块的抑制被编程为模态有效的 NC 功能。有一个启用和禁用功能。

**语法:**

|                                                                 |                                                                        |
|-----------------------------------------------------------------|------------------------------------------------------------------------|
| StepModeDisable({{Variable={}{Value}}}{},{Variable={}{Value}}}) |                                                                        |
| 缩写形式: SMD {(.)}                                                 |                                                                        |
| 带有可选的可编程变量                                                      |                                                                        |
| SBLK                                                            | 以下内容适用于“单轮操作”以及“单步执行”操作模式:<br>0 - NC 启动未被抑制<br>1 - NC 启动被抑制<br>默认设置: 1 |
| PBLK                                                            | 它适用于操作模式“程序块”<br>0 - NC 启动未被抑制<br>1 - NC 启动被抑制<br>默认设置: 1              |
| DBGS                                                            | 它适用于操作模式“调试步骤”<br>0 - NC 启动未被抑制<br>1 - NC 启动被抑制<br>默认设置: 1             |

使用高级语言语法的 NC 功能

|                                                                                          |                                                                           |
|------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|
| 现在                                                                                       | 它表明抑制是否适用于功能编程的块之后。<br>0 - NC 启动从下一个块开始被抑制<br>1 - NC 启动也于此块被抑制<br>默认设置: 0 |
| 如果一个变量没有编程, 指定的预置生效。<br>如果该值未编程, 则将该值分配给该变量。<br>如果没有编程变量, 括号是可选的。空括号不像禁用功能, 而是像预设值的启用功能。 |                                                                           |

表格 6-181: 语法 StepModeDisable (SMD)

|                                        |
|----------------------------------------|
| StepModeEnable                         |
| 缩写形式: SME                              |
| 没有参数可以编程。块中的 SME 被编程 - 是 NC 启动被禁止的块之一。 |

表格 6-182: StepModeEnable (SME) 语法

示例:

|                                                                         |                                                                               |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| <b>示例 1:</b><br>...<br>N50 SMD<br>N60 X100<br>N70 SME<br>...            | 在所有单步模式下, 块 N60 和 N70 都不能通过 NC 启动进行确认                                         |
| <b>示例 2:</b><br>...<br>N50 SMD X100<br>N60 Y100<br>N70 SME<br>...       | 模拟示例 1                                                                        |
| <b>示例 3:</b><br>...<br>N50 SMD(NOW1) X100<br>N60 Y100<br>N70 SME<br>... | 与示例 1 和 2 相反, 即使存在移动运动, 块 N50 也不会以 NC 启动进行确认。<br>块 N60 和 N70 也不能通过 NC 启动进行确认。 |

|                                                                                                                              |                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <b>示例 4:</b><br><br>...<br>N50 SMD(DBGS0) X100<br>N60 Y100 SME<br>...                                                        | 在“除错步骤”模式下，块 N60 可通过 NC 启动进行确认。NC 启动在“程序块”和“单步/单块”运行模式下被抑制。                                              |
| <b>示例 5:</b><br><b>主程序</b><br><br>...<br>N50 P Up (SR)<br>...<br><b>子例程 SR</b><br>N10 SMD(NOW)<br><br>...<br>...<br>N999 SME | 通过 NC 启动，子程序在主程序的 N50 块中的所有单步模式下完全处理。因此，该子程序不需要 NC 启动。<br><br>请注意，必须使用变量“NOW”在第一块中激活单块抑制 – 并且必须在最后一块中停用。 |

表格 6-183: 单步模式抑制的编程示例

**特殊功能和限制:** 单步模式抑制不会妨碍在锁定操作模式情况下的操作模式的改变。操作模式已更改，但锁定的块仍处理，无 NC 启动。

新 SME 编程覆盖以前有效 SME。只需要一个 SMD。

SME 在控制启动后处于活动状态，并且在控制复位和程序结束时另外隐式激活。这意味着操作模式抑制不会通过程序重启或程序更改进行模态操作。但是，在子程序返回的情况下，SME 通常不会被激活。这也适用于异步子程序。确保“SME”功能被编程为禁用抑制，因为单块模式下 NC 程序块不会由于意外事件而被 NC 启动确认。

每个程序首先使用“SME”功能启动。由于安全原因，无法设置此行为。

6. 139 特定于刀具的主轴限制"SToolLim"

**描述:** 在 /SysSpCmdData[<SysSp>]/Tool 系统日期中，在一个或多个通道主轴中启用输入的刀具特定主轴限制。



有关更多信息，请参阅“刀具特定的主轴限制”下“主轴功能章节中的”IndraMotion MTX 功能说明基本内容”。

|                                                     |                                                                |
|-----------------------------------------------------|----------------------------------------------------------------|
| <b>语法:</b>                                          |                                                                |
| SToolLim, STOOLLIM                                  |                                                                |
| SToolLim(<SpName> = <0 1>{ , <SpName> = <0 1>} ...) |                                                                |
| 其中                                                  |                                                                |
| <SpName>                                            | 主轴名称 (S, S1..S8)                                               |
| <0 1>                                               | <b>1</b> 启用 /SysSpCmdData[]/Tool 的工具特定限制<br><b>0</b> 禁用工具特定的限制 |

表格 6-184: 语法 SToolLim

**示例:** SToolLim(S2=1) 启用当前分配给第二通道主轴的系统主轴的刀具特定限制。

**特殊功能和限制:**

- SToolLim 停止编程的主轴并随后重新启动它们。

使用高级语言语法的 NC 功能

- SSpToolLim 报告换档到 “SpindleWait 、 SPWAIT” 和 “SpindleWaitMode、SWM” 功能。
- SToolLim 包括 NC 内部 WAIT 功能。

6.140 TCS 中的进给 "TCM( , , )"

**描述:** 在刀具坐标系 TCS 和激活的轴变换 2 中进行线性进给运动。  
这是一个局部相对尺寸编程（即，不管 G90/G91 如何，指定的编程位置都是增量解释的）。  
有关辅助功能的使用和 “TcsMove” 参数设置的全面信息，请参阅文档 “IndraMotion MTX 功能描述”。

|     |                                 |                |
|-----|---------------------------------|----------------|
| 语法: | TCM({<xVal>},{<yVal>},{<zVal>}) |                |
|     | (可选: TcsMove( , , ))            |                |
|     | 其中                              |                |
|     | <xVal>, <yVal>, <zVal>          | 刀具坐标系统中增量进给的值。 |

表格 6-185: TCM 语法

|     |                 |                     |
|-----|-----------------|---------------------|
| 示例: | TCM ( , , -100) | ; 在刀具方向上插入 100 [mm] |
|     | TCM ( , , 200)  | ; 在刀具方向上回退 200 [mm] |

6.141 切向刀具控制 "TangTool、TTL"

**描述:** 使用可调节角度，在活动平面内沿着编程的路径旋转或采集坐标。如此可让刀具使用所需的定位到编程路径的定位角随时移动。  
为使刀具轴旋转至对应的所需定位角度，此函数在刀具具有多个刀刃时会考虑所有的现有刀具对称。此外，若在块过渡处的轮廓中存在弯角，控制器可自动插入一个中间块，使刀具轴按照所需的角度的旋转。

以下情况适用:

- 在旋转刀具轴的 0 度位置处，刀具默认的刃口方向与当前平面中主轴的正进给方向平行。

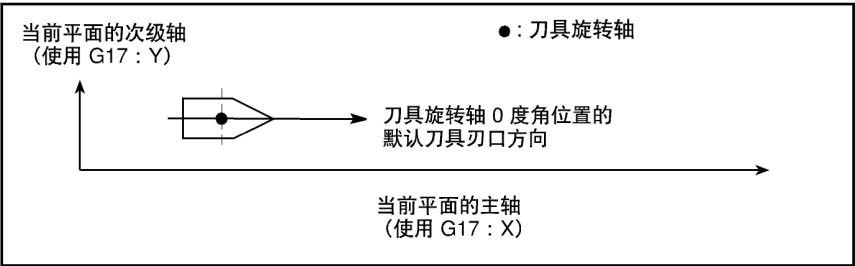


插图 6-37: 刀具回转轴 0 度角位置

- 定位角表示路径和默认刀刃方向间的角度差。  
若默认刀刃方向与编程路径相切，则该角度为 0。

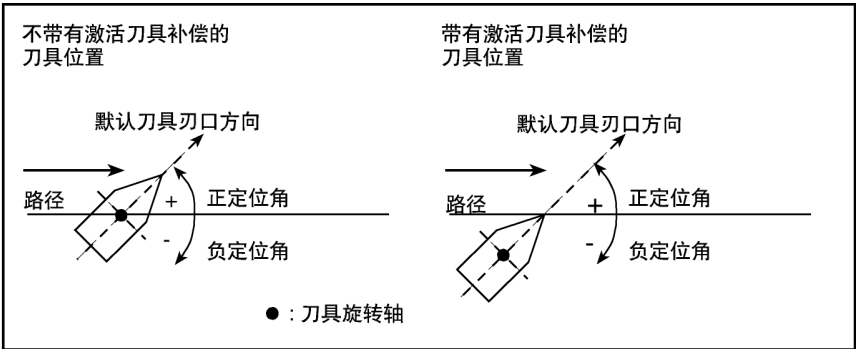


插图 6-38: 设置角度

- 激活时“TangTool”不会引起任何横向运动。  
下一进给运动前不会执行编程的刀具轴。根据参数设定的调整角度，
  - 可预先进给刀具旋转的中间块，或
  - 使刀具在块的开始处跳转至其新位置。
- 圆周路径内给定定位角刀具轴所需旋转的计算在插补器时钟脉冲中发生。由此，在此类轮廓部分中，刀具轴通过使用各插补器时钟脉冲计算的对应定位角保持旋转。



在 Sercos 循环内，最大允许的中间块角度（ZSW）不得大于轴以快速进给速度进给的角度。  
$$ZSW [^\circ] < AXDr[i]/Vel/MaxVel * SCS/ScsIf[1]/ScsCycTime * 1/60.000.000$$

语法:

|                                                                                            |                                                                                     |
|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| TangTool{ ({TAX{=}<Coordinate>}<br>{,SYM{=}<s>}{,ANG{=}<a>}<br>{,IA{=}<zsw>}{,PLC{=}<p>})} |                                                                                     |
|                                                                                            | 切向刀具引导开启                                                                            |
| TangTool() 或<br>TangTool(0)                                                                | 切向刀具引导关闭                                                                            |
| 缩写形式: TTL( .. )                                                                            |                                                                                     |
| 其中                                                                                         |                                                                                     |
| <Coordinate>                                                                               | 使用“切向刀具引导”函数的轴的名称。<br>容许的值:<br>通道坐标或轴的名称或编号。<br>若未对此编程,则 MP 7050 00210 生效。只能输入系统轴号。 |

使用高级语言语法的 NC 功能

|       |                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <s>   | <p>工具对称性（通常是边的数量）。</p> <p>输入值：整数，不等于 0。</p> <p>包含对称值 &lt;s&gt; 的刀具在旋转 360 度/&lt;s&gt; 后，将获得一个技术上相同的位置。</p> <p>示例：</p> <p>矩形刀具：&lt;s&gt;= 2,</p> <p>四角工具：&lt;s&gt;= 4。</p> <p><b>1:</b></p> <p>不对称的刀具，或仅有一个刀刃。</p> <p><b>&gt;0:</b></p> <p>对称的刀具，有数个等距间隔的刀刃。若轮廓中有弯角，刀具仅旋转至允许最近的刀刃使用轮廓在定位角定位的程度。</p> <p><b>&lt;0:</b></p> <p>若方向发生更改（180 度弯角），则无视定位角，不会旋转刀具。否则将与 “&gt;0” 的情况相同。</p> <p>若未对此编程，则 MP 7050 00220 生效。</p> |
| <a>   | <p>设置角</p> <p>取值范围：-180 至 +180 度</p> <p>这表示路径和默认刀刃方向间的角度差。</p> <p>若未对此编程，则 MP 7050 00250 生效。它用于指定旋转刀具轴的当前角度是否用作定位角，或是使用 MP 7050 002540 的值代替。</p>                                                                                                                                                                                                                                                                    |
| <zsw> | <p>中间块角度</p> <p>取值范围：0 至 180 度</p> <p>定义两个块之间的弯角，会在这两个块之间插入一个用来旋转刀具轴的中间块。</p> <p>若轮廓弯角处的角度小于 &lt;zsw&gt;，则不插入用于旋转刀具轴的单独块。而是让刀具在下一个块的开头跳转至其新位置。</p> <p>在 Sercos 循环内，最大允许的中间块角度（ZSW）不得大于轴以快速进给速度进给的角度。</p> <p><b>ZSW [° ] &lt; AXDr[i]/Vel/MaxVel * SCS/ScsIf[1]/ScsCycTime * 1/60.000.000</b></p> <p>若未对此编程，则 MP 7050 00230 生效。</p>                                                                                  |
| <p>   | <p>在中间块的执行期间激活或取消 NC-PLC 通信。</p> <p><b>0:</b></p> <p>NC-PLC 通信关闭。</p> <p>NC 无条件地执行中间块。</p> <p><b>1:</b></p> <p>NC-PLC 通信打开。</p> <p>NC-PLC 通信将控制旋转块的执行。</p>                                                                                                                                                                                                                                                          |

表格 6-186: TangTool (TTL) 语法

|     |                                      |                                         |
|-----|--------------------------------------|-----------------------------------------|
| 示例： | TTL                                  | 使用 MP 7050 00210 中输入的轴的相应机床参数值激活切向刀具引导。 |
|     | TTL (TAX=C, SYM1, ANG90, IA20, PLC0) | 使用逻辑轴名称编程                               |

|                                      |             |
|--------------------------------------|-------------|
| TTL (TAX=theta, ANG45)               | 用坐标名称编程     |
| TTL (TAX3, SYM1, ANG90, IA20, PLC0)  | 用坐标编号编程     |
| TTL (TAX[NAME\$], SYM1, ANG90, IA20) | 使用 CPL 变量编程 |

- 特殊功能和限制:
- 函数“切向刀具引导”不可同时作为“切向刀具定向”处于活动状态（参见第 6.142 章“切向刀具定向 TangToolOri、TT0”第 353 页）。
  - 不得结合进给运动编程“切向刀具引导”。



如果括号表达式“(0)”或“(1)”不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

## 6.142 切向刀具定向 "TangToolOri、TT0"

描述:



仅在结合“冲孔”（请参见第 6.93 章“冲孔 Punch、PUN”第 302 页）和“裁料”（请参见第 6.79 章“裁料 Nibble、NIB”第 274 页）函数使用时才有效。

冲压/步冲开启时，此可确保冲压/步冲刀具在每个冲程过程以可定义角度定位到编程路径上。

对于块初始处的冲程，在必要时可自动插入一个用于旋转刀具的块。

控制器会自动确定最佳的旋转方向（最短路径）。

回转轴通过 MP 7050 00210 指定。

语法:

|                                                                   |          |
|-------------------------------------------------------------------|----------|
| TangToolOri({SYM<s>},{ANG<a>}) 或<br>TangToolOri(1)<br>TangToolOri |          |
|                                                                   | 切向刀具定向开启 |
| TangToolOri()<br>或<br>TangToolOri(0)                              | 切向刀具定向关闭 |
| 缩写形式: TTO...                                                      |          |
| 其中                                                                |          |

使用高级语言语法的 NC 功能

|     |                                                                                                                                                                                                                      |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <s> | 工具对称性（通常是边的数量）<br>输入值：整数，大于 0<br>包含对称值 <s> 的刀具在旋转 360 度/<s> 后，将获得一个技术上相同的位置。<br>示例：<br>矩形刀具：<s>= 2,<br>四角工具：<s>= 4。<br><b>1:</b><br>不对称的刀具，或仅有一个刀刃。<br><b>&gt;1:</b><br>对称的刀具，有数个等距间隔的刀刃。<br>如果 SYM 未编程：效果与 SYM1 相同。 |
| <a> | 定位角至编程的路径<br>取值范围：-180 至 +180 度<br><b>0:</b><br>刀具轴以 0 度角定位到编程路径。<br>如果 SYM 未编程：效果与 ANG0 相同。                                                                                                                         |

表格 6-187: TangToolOri (TTO) 语法

- 示例：
- G90 处于活动状态（绝对尺寸编程）
  - 激活平面：X/Y
  - 刀具轴：C（模态 360）
  - 当前位置：X=0, Y=0, C=0
  - 冲压/步冲已禁用。

|                           |                                                |
|---------------------------|------------------------------------------------|
| :                         |                                                |
| N10 TangToolOri(1)        | 切向刀具定向打开，其效果与 SYN1 和 ANG0 相同。                  |
| N20 G1 G91 X10 Y10        | 由于冲压/步冲禁用，目前 C 轴尚无定向。                          |
| N30 X10 Y10 Punch(1)      | 冲孔打开<br>C 轴旋转至 45°                             |
| N40 Y-10                  | C 轴旋转至 -90°<br>模态计算后：C=270°                    |
| N50 Punch(0)              | 冲孔关闭                                           |
| N60 LEN=30 Nibble(1)      | 裁料打开<br>所需的块段长度：30 mm                          |
| N70 G2 X114.6 I57.3<br>J0 | 具有长为 180 mm 的弧的半圆<br>将块分为 6 个块段，每个段的弧长为 30 mm。 |
| N80 TTO()                 | 切向刀具定向关闭                                       |
| :                         |                                                |



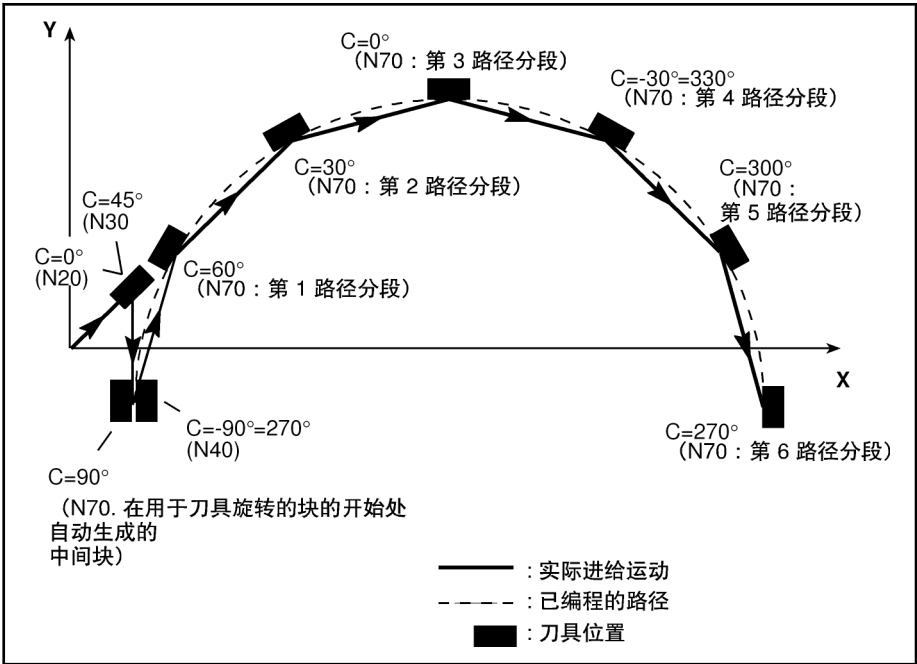


插图 6-39: TangToolOri (TTO) 的示例

- 特殊功能和限制:
- 此函数不可与“TangTool”函数（切向刀具引导，TTL）共同编程。
  - 该函数处于活动状态时，将无法使用“切向刀具引导”（TTL，参见第 6.141 章“切向刀具控制 TangTool、TTL”第 350 页）。

如果括号表达式“(0)”或“(1)”不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

6.143 从攻丝缩回 "TappRet1、TappRet2"

描述:

丝锥仍在运行时，若（通过控制器复位或电压下降）取消“无补偿夹具的攻丝”函数（G63），则可使用“TappRet1”和“TappRet2”将丝锥从攻丝孔中退出。可通过手动输入或在零件程序 / 子例程（循环）中进行编程。

语法:

|               |                                           |
|---------------|-------------------------------------------|
| TappRet1      | 在“控制器复位”（控制器启动）之后，将主轴速度模式下的主轴运行再次切换到位置模式。 |
| TappRet2 F... | 利用编程的 F 值启动实际的回缩运动。                       |

表格 6-188: TappRet1、TappRet2 的语法

必须在回缩之前编程“TappRet1”。

在使用“从攻丝缩回”函数时，必须区分两种情况：

- 控制器复位后缩回（自动缩回）：**  
将保留启动攻丝过程时保存的数据。使用“TappRet1”，将主轴切换至位置模式。若接着编程了“TappRet2”，攻丝将从螺纹内移出，并移至所储存的起始位置。只有“TappRet2”必须与所需的进给（F 值）共同编程。
- 电源故障后的缩回（手动缩回）：**  
攻丝开始时保存的回退信息将丢失。在该情况下，必须将相关参数与“TappRet2”共同编程：  

```
G91 TappRet2 S<speed> F<feed> M3/M4 <boring axis> <incremental distance>
```

使用高级语言语法的 NC 功能

使用 *TappRet1/TappRet2* 的先决条件:

- 必须将相关 C 轴定义为模态回转轴:
  - MP 1001 00004 (/AXSP/Dr[1]/AxFun/ModCalc/ENA\_ModCalc)
  - Sercos 参数 S-0-0076 = 0b1xxxxxxx。
- 在 Sercos 的辅助操作模式 1 (S-0-0033) 下, 用于操作模式的驱动器控制的更改的第 8 位不可设为:  
S-0-0033 = 0b000001011 或 0b000001100。

子程序用于回收线程的应用程序:

为了便于应用“从攻丝回缩”的功能, 请为自动编写一个子程序(循环), 为手动回溯编写一个子程序。这些子程序可以通过以下机器参数分配给任何未分配的 G 代码。

- 3090 00001 (/NCP/SubProg/GFun/Fun[1]/GCode)
- 3090 00002 (/NCP/SubProg/GFun/Fun[1]/SubProg)

6. 144 无补偿夹具的攻丝的主轴选择 "TappSp、TSP"

- 描述:
- 指定 G63 需要参考的主轴。  
“TappSp”在被重新编程前始终有效。它在“控制器复位”后仍然有效!
  - 若未编程“TappSp”, G63 将始终参照第一主轴。

语法:

|                           |                                                                                                |
|---------------------------|------------------------------------------------------------------------------------------------|
| TappSp (CAX<i>)           | 使用主轴编号来激活各主轴的攻丝。                                                                               |
| TappSp (GRP<j>)           | 使用主轴组来激活所有主轴的攻丝。                                                                               |
| TappSp (GRP<j>{, CAX<i>}) |                                                                                                |
|                           | 使用主轴组来激活所有主轴的攻丝, 并使用主轴编号激活附加主轴的攻丝。                                                             |
| 缩写形式: TSP ( .. )          |                                                                                                |
| 其中                        |                                                                                                |
| <i>                       | G63 所参考的 <b>主轴</b> 编号。<br>取值范围: 1...n, n 代表系统中主轴的最高编号(最大为 8)。整数。<br>在编程时有些主轴使用了多个逗号分隔的 CAX<i>。 |
| <j>                       | G63 所参考的 <b>主轴组</b> 编号。<br>取值范围: 1...n; n 代表主轴组的最高编号(最大为 4)。整数。                                |

表格 6-189: TappSp (TSP) 语法

示例:

|                              |                                    |
|------------------------------|------------------------------------|
| N20 TappSp (CAX2)            | 为 G63 选择主轴编号为 2 的主轴。               |
| :                            |                                    |
| N120 TappSp (CAX2,CAX4,CAX7) | 为 G63 选择主轴编号为 2、4 和 7 的主轴。         |
| :                            |                                    |
| N220 TappSp (GRP2)           | 为 G63 选择主轴编号为 2 的主轴组。              |
| :                            |                                    |
| N320 TappSp (GRP3,CAX4)      | 为 G63 选择包含编号 3 主轴组, 以及主轴编号为 4 的主轴。 |
| :                            |                                    |

- 特殊功能和限制：
- 仅可为 G63 激活单个主轴组。
  - 您可以任意方式组合主轴编号，并可在必要时加上一个主轴组。

6. 145 调整轴配置 "TargetAxisConf, TAX"

- 描述：
- 在预运行期间，TargetAxisConf（缩写形式：TAX）NC 功能可以方便地调整调整程序和动作块中的轴名称和轴名称组态。

控制器使用 TAX 命令在块预运行的目标块之前激活轴配置。多余的轴被移除。通道中包含必需的轴。轴名称也适用。

不考虑分配的轴：

  - 不能包含分配给其他通道的轴。
  - 不能包含处于主轴模式下的 C 轴。
  - 作为变换一部分的轴不能被删除。

未考虑的轴随后可能会受到单个命令的影响。必要的信息存档在系统数据（Sys\_ChSRun）中。

可以重复调用 TAX，例如，在转换已经停用之后。
- 语法：
- TAX 或 TargetAxisConf
- 示例：
- 在计算过程中，第二个系统轴从通道中移除，“Y”是分配给第三个系统轴的名称，第五个和第十一个系统轴以“Z”和“U”的名称包含在通道中。

| 计算之前  |    | 计算过程中                | 计算完成后 |    |
|-------|----|----------------------|-------|----|
| SysNo | 名称 | ...                  | SysNo | 名称 |
| 1     | X  | N50 RAX(Y)           | 1     | X  |
| 2     | Y  | N60 ALN(Z, Y)        | 3     | Y  |
| 3     | Z  | N70 WAX(5, Z, 11, U) | 5     | Z  |
| 7     | C  | ...                  | 7     | C  |
|       |    |                      | 11    | U  |

初始调节程序：

```
...
N100 X10 Y10 Z10 C10 ; 1、2、3 和 7。系统轴行进至位置 10
N110 TAX
N120 X20 Y20 Z20 C20 ; 1、3、5、7 和 11。系统轴行进至位置 20
U20
...
```

- 特殊功能和限制：
- TAX 只允许在预运行的调整阶段，即在调整程序和动作块中。

6. 146 在程序坐标中定义 TCS "TcsDef、TCS"

- 描述：
- 激活 6 轴变换的情况下，会生成一个刀具坐标系 TCS<sub>p</sub>，该坐标系可参照上次的 TCS 进行偏置和/或旋转。

给予 TCS<sub>p</sub> 的坐标值会在 NC 内进行转换，并储存于刀具修正内存中。

关闭函数可清除已生成的 TCS<sub>p</sub>。此过程中会重新激活上次的 TCS。
- 语法：
1. 若尚未执行，则激活轴变换类型 **3333301**（参见函数“Coord(...)”，第 6.28 章“选择轴转换 Coord、CRD”第 225 页）。

使用高级语言语法的 NC 功能

2. 使用以下所示语法。

|                                        |                                        |
|----------------------------------------|----------------------------------------|
| TcsDef ({<Position>}, {<orientation>}) |                                        |
|                                        | 定义并激活 TCS <sub>p</sub> 的位置和方向。语法的详情如下。 |
| TcsDef () 或<br>TcsDef (0)              | 删除 TCS <sub>p</sub> ，并重新激活上次的 TCS。     |
| 缩写形式: TCS (...)                        |                                        |

表格 6-190: TcsDef (TCS) 语法

语法详情:

|                                                                                              |                                                                                   |
|----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {<phi><Φ>}, {<theta><Θ>}, {<psi><Ψ>} or            |                                                                                   |
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {O (<Φ> ,<Θ> ,<Ψ> )} or                            |                                                                                   |
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {Ox (<Φ <sub>xxyy</sub>                            |                                                                                   |
| 或                                                                                            |                                                                                   |
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {Ox (<Φ <sub>xxzz</sub>                            |                                                                                   |
| 或                                                                                            |                                                                                   |
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {Oy (<Φ <sub>yyzz</sub>                            |                                                                                   |
| 或                                                                                            |                                                                                   |
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {Ox (<o11>,<o21>,<o31>) Oy (<o12>,<o22>,<o32>)} or |                                                                                   |
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {Ox (<o11>,<o21>,<o31>) Oz (<o13>,<o23>,<o33>)} or |                                                                                   |
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {Oy (<o12>,<o22>,<o32>) Oz (<o13>,<o23>,<o33>)} or |                                                                                   |
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {ROTAX (<Φ <sub>uu</sub>                           |                                                                                   |
| TcsDef ( {<x><px>}, {<y><py>}, {<z><pz>}, {ROTAX (<u <sub>xyz</sub>                          |                                                                                   |
| 其中                                                                                           |                                                                                   |
| <x>                                                                                          | MP 7080 00010[1] 中输入的坐标名称。<br>默认: x                                               |
| <y>                                                                                          | MP 7080 00010[2] 中输入的坐标名称。<br>默认: y                                               |
| <z>                                                                                          | MP 7080 00010[3] 中输入的坐标名称。<br>默认: z                                               |
| <px>、<py>、<pz>                                                                               | 参考当前 PCS，轴地址 <x>、<y> 和 <z> 的绝对笛卡尔坐标。这些值定义了新 TCS <sub>p</sub> 的来源。<br>允许 AC/IC 编程。 |
| <phi>                                                                                        | MP 7080 00010[4] 中输入的角名称。<br>默认: phi                                              |
| <theta>                                                                                      | MP 7080 00010[5] 中输入的角名称。<br>默认: theta                                            |
| <psi>                                                                                        | MP 7080 00010[6] 中输入的角名称。<br>默认: psi                                              |

|                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\langle \phi \rangle, \langle \vartheta \rangle, \langle \psi \rangle$                                                                                                                                                                                                                                                                                                                                  | 与当前 PCS 相关的绝对欧拉角，以度为单位。这些值定义了新 $TCS_p$ 的来源。<br>允许 AC/IC 编程。<br>取值范围：<br>$0^\circ \leq \phi < 360^\circ$<br>$0^\circ \leq \vartheta \leq 180^\circ$<br>$0^\circ \leq \psi \leq 360^\circ$<br>此范围以外的值将自动转换为对应的定义间隔。                                                           |
| $\langle \phi.. \rangle, \langle \vartheta.. \rangle$                                                                                                                                                                                                                                                                                                                                                    | 绝对角度值（单位：度）。<br>取值范围：<br>$0^\circ \leq \phi.. < 360^\circ$<br>$0^\circ \leq \vartheta.. \leq 180^\circ$<br>若将 $\vartheta$ 编程为值域以外的数值，它会自动地转换为给定区间。                                                                                                                          |
| $Ox(\langle o11 \rangle, \langle o21 \rangle, \langle o31 \rangle)$<br>$Oy(\langle o12 \rangle, \langle o22 \rangle, \langle o32 \rangle)$<br>$Oz(\langle o13 \rangle, \langle o23 \rangle, \langle o33 \rangle)$<br>$Ox(\langle \phi_x \rangle, \langle \vartheta_x \rangle)$<br>$Oy(\langle \phi_y \rangle, \langle \vartheta_y \rangle)$<br>$Oz(\langle \phi_z \rangle, \langle \vartheta_z \rangle)$ | 使用函数 $Ox(..)$ 、 $Oy(..)$ 、 $Oz(..)$ 进行定向。<br>例如， $Ox(..)$ 可定义 $TCS_{i_p}$ 的 $x$ 坐标在参考坐标系中的方向。这同样适用于 $Oy(..)$ 和 $Oz(..)$ 。<br>可使用对应极角 $\langle \phi.. \rangle$ 以及 $\langle \vartheta.. \rangle$ ，或使用 $TCS_p$ 定向张量列向量的笛卡尔分量来指定方向。<br>只允许绝对尺寸。<br>列向量 ( $o..$ ) 的分量值被自动标准化为 1。 |
| $ROTAX(\langle \phi_u \rangle, \langle \vartheta_u \rangle)$                                                                                                                                                                                                                                                                                                                                             | 使用极角 ( $\phi_u$ , $\vartheta_u$ ) 定义回转轴。                                                                                                                                                                                                                                    |
| $ROTAX(\langle u_x \rangle, \langle u_y \rangle, \langle u_z \rangle)$                                                                                                                                                                                                                                                                                                                                   | 使用绝对尺寸中的笛卡尔分量 $\langle u_x \rangle$ 、 $\langle u_y \rangle$ 、 $\langle u_z \rangle$ 定义回转轴。自动标准化为 1。                                                                                                                                                                         |
| $O(\langle \beta \rangle)$                                                                                                                                                                                                                                                                                                                                                                               | 指定 $TCS_p$ 定向张量绕回转轴旋转要使用的增量角 $\langle \beta \rangle$ （单位：度），其源自上次 TCS 的定向张量。<br>允许大于 360 度的值。<br>可以使用正 / 负符号来选择旋转方向。                                                                                                                                                        |

表格 6-191: *TcsDef (TCS) 语法的详情*

- 特殊功能和限制：
- 活动轴转换类型 3333301 必须处于活动状态。
  - 如果函数 “Coord (...)” 重新编程（参见 第 6.28 章 “选择轴转换 Coord、CRD” 第 225 页），则自动清除之前生成的  $TCS_p$ ，并重新激活上次的 TCS。
  - 若因编程张量列而出现两个平行或逆平行的列向量，则无法计算定向张量。  
显示运行时错误。



如果括号表达式 “(0)” 或 “(1)” 不代表 NC 功能的参数。因此，括号内的数值不能通过 CPL 变量传输。

## 6.147 特殊螺纹切削函数 “ThreadSet、TST”

### 6.147.1 一般信息

**描述：** “ThreadSet” 允许临时调整 G33 的部分区域。  
在这种情况下，控制器会叠加储存于机床参数中的静态值。

- “ThreadSet” 允许：
- 调整缩进动作的动态。
  - 切换主轴操作模式（速度控制，位置控制）；
  - 在通道接口设定信号（可使用机床参数在“有效函数”信号内配置）。
- 控制器复位或 M30
- 可使用 “ThreadSet”取消叠加设置，
  - 清除 “ThreadSet”设置的 IF 信号，
  - 若之前使用 “ThreadSet(SPC1)”将主轴更改为位置受控的主轴运行，则将其切换会回速度受控的运行。



以下描述的所有部分函数可在一个 “ThreadSet”块中进行联合编程（使用逗号分隔）。

6. 147. 2 配置缩回数据 ThreadSet (..)

语法：

|                                        |                                                                          |
|----------------------------------------|--------------------------------------------------------------------------|
| ThreadSet (RD(<HA-value>, <NA-value>)) |                                                                          |
| 缩写形式：TST(..)                           |                                                                          |
| 其中                                     |                                                                          |
| <HA value>                             | 朝向当前选定平面（G17、G18、G19、G20）主轴的回退路径（增量，单位：毫米）。<br>必须始终编程好该值；但该值仅与纵向和锥形螺纹相关。 |
| <NA 值>                                 | 朝向当前选定平面辅助轴的缩回路径（增量，单位：毫米）。<br>必须始终编程好该值；但该值仅与横向和锥形螺纹相关。                 |
| 或：                                     |                                                                          |
| ThreadSet (RD( , , -1))                |                                                                          |
| -1                                     | 在此情况下，来自 MP 7050 00645 和 MP 7050 00650 的回退数据将再次处于活动状态。                   |

表格 6-192: ThreadSet (TST) 配置回退数据的语法

6. 147. 3 配置动力学 ThreadSet (DYN)

语法：

|                                                               |                                                    |
|---------------------------------------------------------------|----------------------------------------------------|
| ThreadSet (DYN({<jump>}, {<Accel>} {, <Decel>})) 或<br>TST(..) |                                                    |
| 其中                                                            |                                                    |
| <Jump>                                                        | 最大允许跳跃速度，单位为 mm/min<br>输入 “-1” 再次启用 MP 7050 00610。 |
| <Acc>                                                         | 加速度（单位：m/s²）<br>输入 “-1” 再次启用 MP 7050 00615。        |
| <Brake accel>                                                 | 减速度（单位：m/s²）<br>输入 “-1” 再次启用 MP 7050 00620。        |

表格 6-193: ThreadSet (TST) 配置动态的语法

6. 147. 4 启用缩回 ThreadSet (RON)

|                                       |                        |
|---------------------------------------|------------------------|
| 语法:                                   |                        |
| ThreadSet (RON<Status>) 或<br>TST (..) |                        |
| 其中                                    |                        |
| <Status>                              | 0: 取消快速缩回<br>1: 激活快速缩回 |

表格 6-194: ThreadSet (TST) 启用回退的语法

6. 147. 5 切换主轴操作模式 ThreadSet (SPC)

|                                       |                                                                       |
|---------------------------------------|-----------------------------------------------------------------------|
| 语法:                                   |                                                                       |
| ThreadSet (SPC<Status>) 或<br>TST (..) |                                                                       |
| 其中                                    |                                                                       |
| <Status>                              | 0:<br>将主要的主轴切换至速度受控模式。<br>1:<br>根据 7050 00600 [3] 的设置, 将主主轴切换为位置受控模式。 |

表格 6-195: ThreadSet (TST) 切换主轴操作模式的语法

有关主要主轴的信息, 请参见 第 6.72 章 “主轴切换 MainSp、MSP” 第 266 页。

6. 147. 6 影响通道 IF 信号 ThreadSet (TCI)

|                                       |                              |
|---------------------------------------|------------------------------|
| 语法:                                   |                              |
| ThreadSet (TCI<Status>) 或<br>TST (..) |                              |
| 其中                                    |                              |
| <Status                               | > 0: 清除通道接口信号<br>1: 清除通道接口信号 |

表格 6-196: ThreadSet 语法 (TST), 影响通道接口信号



可在“激活函数”信号中使用机床参数配置需要在通道接口处影响的信号。

6. 148 程序坐标偏置 "Trans、TRS", 叠加程序坐标偏置 "ATrans、ATR"

描述: “Trans” 和 “Atrans” 将相对于当前工件坐标系的当前程序坐标系置换。

两个偏移量都可以彼此激活和关闭。如果两者都处于活动状态, 它们会相加作用。

由此, 零件程序可在不进行修改编程轮廓的情况下在工件坐标系内的任意位置执行。基于激活状态的程序坐标系的输入刀具 (例如, 已编程的轮廓偏置、缩放、镜像、旋转) 都独立于程序坐标偏置, 因此无需进行适配。

使用高级语言语法的 NC 功能

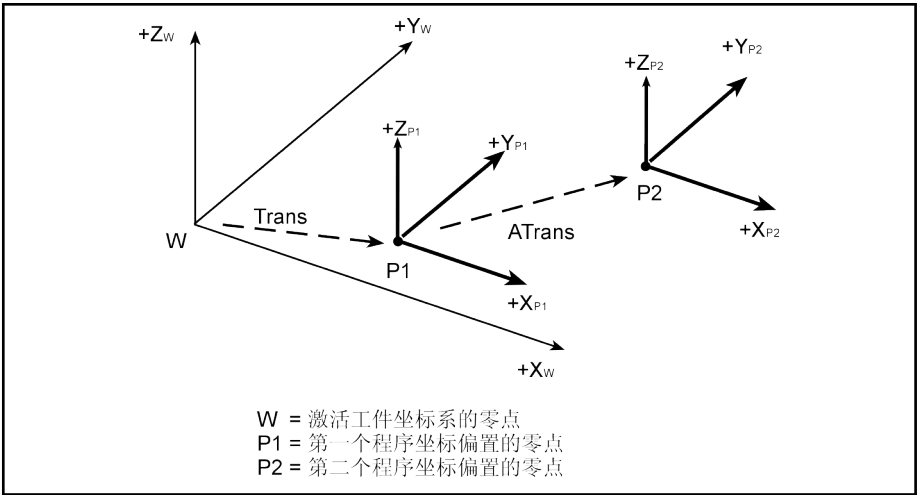


插图 6-40: 程序坐标偏置，叠加的程序坐标偏置

以下情况适用:

编程序坐标偏置不会触发进给运动。

|     |                           |                                                                                                                                  |
|-----|---------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| 语法: | Trans (<Coordinates>)     | 坐标偏置编程开启<br><br>程序坐标系的零点通过编程的<coordinates>移动到工件坐标系。偏置行为累加 “ATrans”。<br><br>可在括号内编程多个坐标轴（使用逗号分隔）。<br><br>（示例：TRS (X100,Y50,Z50)）。 |
|     | Trans () 或<br>Trans (0)   | 所有编程坐标偏置关闭                                                                                                                       |
|     | 缩写形式: TRS (...)           |                                                                                                                                  |
|     | ATrans (<coordinates>)    | 叠加编程坐标偏置开启<br><br>程序坐标系的零点通过编程的<coordinates>移动到工件坐标系。偏移量对 “Trans”增加了作用。<br><br>可在括号内编程多个坐标轴（使用逗号分隔）。<br><br>（示例：ATR (X10,Z70)）。  |
|     | ATrans () 或<br>ATrans (0) | 叠加程序坐标偏置关闭                                                                                                                       |
|     | 缩写形式: ATR (...)           |                                                                                                                                  |

表格 6-197: Trans (TRS) 和 ATrans (ATR) 的语法

|     |                       |                                    |
|-----|-----------------------|------------------------------------|
| 示例: | N10 ATR () TRS ()     | 叠加程序坐标偏置关闭。                        |
|     | :                     | 将程序坐标系的零点设为当前工件坐标系的位置 X0 Y0 Z0。    |
|     | N30 TRS (X10,Y10,Z50) | 将程序坐标系的零点设为当前工件坐标系的位置 X10 Y10 Z50。 |
|     | :                     |                                    |
|     | N50 ATR (X20,Y10)     | 将程序坐标系的零点设为当前工件坐标系的位置 X30 Y20 Z50。 |
|     | :                     |                                    |



N70 ATR (X50, Z30)  
:  
N90 TRS ()  
:  
将程序坐标系的零点设为当前工件坐标系的位置 X60 Y20 Z80。  
  
坐标偏置编制关闭  
  
将程序坐标系的零点设为当前工件坐标系的位置 X50 Y10 Z30。

6. 149 体积补偿

6. 149. 1 启用和禁用 "VCPEna"

描述:

NC 功能启用 / 禁用通道中的体积补偿。在启用补偿之前，一个表格文件将加载 NC 功能 VCPLoad（第 6. 149. 2 章“加载表 VCPLoad”第 363 页）。每个通道的表格文件有两个内存插槽。

语法:

|                         |                                |
|-------------------------|--------------------------------|
| VCPEna (<TabIndex>)     | 启用补偿                           |
| VCPEna                  | 在表 1 中启用补偿                     |
| VCPEna (0)<br>VCPEna () | 禁用补偿                           |
| 其中                      |                                |
| <TabIndex>              | 其表应该用于补偿的内存插槽的编号<br>取值范围：1 或 2 |

表格 6-198: VCPEna 语法

示例 1:

程序

N10 VCPLoad (/Machine.ofm) ; Load table file in memory 1  
N20 VCPEna ; Enable compensation, table from 1  
N30 ... ; Machining  
N40 VCPEna () ; Disable compensation

示例 2:

程序

N10 VCPLoad (/mnt/ToolHead1.ofm) ; Load file in memory 1  
N20 VCPLoad (/mnt/ToolHead2.ofm, 2) ; Load file in memory 2  
N30 VCPEna ; Compensation on, table 1  
N40 ... ; Machining  
N50 VCPEna () ; Disable compensation  
N60 ToolHeadChange (2) ; Change tool head 2  
N70 VCPEna (2) ; Compensation on, table 2  
N80 ... ; Machining  
N90 VCPEna () ; Disable compensation

- 特殊功能和限制:
- 启用和禁用补偿时，校正值在 3s 内行进。因此，机器移动。
  - 运动周期取决于校正值和允许的（安全）速度。
  - 如果所需的轴不在通道中或者指定的轴转换 AT1 未激活，则会在激活时创建警告并且不会进行补偿。
  - 如果轴转换 AT1 被切换或禁用，则补偿将自行关闭并创建警告。
  - 如果从通道中移除补偿的轴，则会输出警告并禁用补偿。
  - 在块预运行之后调整“VCPEna”和“VCPLoad”，可以使用 NC 功能“ADJ（VCP）”。

6. 149. 2 加载表 "VCPLoad"

描述:

NC 功能可用于加载表格文件以进行体积补偿。NC 功能 VCPEna（第 6. 149. 1 章“启用和禁用 VCPEna”第 363 页）启用和禁用补偿。每个通道的表格文件有两个内存插槽。

使用高级语言语法的 NC 功能

|     |                                        |                                     |       |
|-----|----------------------------------------|-------------------------------------|-------|
| 语法: | VCPLoad (<TabName><br>{, <TabIndex> }) |                                     | 加载表文件 |
|     | 其中                                     |                                     |       |
|     | <TabName>                              | 文件名包括要加载的表文件的路径                     |       |
|     | <TabIndex>                             | 其表应该用于补偿的内存插槽的编号<br>取值范围：1 或 2，规格：1 |       |

表格 6-199: VCPLoad 语法

- 示例 1:  
特殊功能和限制:
- 请参阅 VCPLoad ( 第 6.149.1 章 “启用和禁用 VCPEna” 第 363 页 )
- 如果表格当前在指定的存储位置处于活动状态，则会生成警告，并且未加载编程的表格。
  - 为了在关闭之后直接将表格加载到相同的存储位置，“WAIT”和用于重新校正校正的短暂等待时间必须被编程在两者之间用于同步目的。
  - 表格文件可能位于任何地方。
  - 表格文件可用的内存是有限的。如果数据不适合内存，则会生成警告。

6.150 虚拟驱动器 "VirtAxisPos、VAP"

描述: 在当前通道中设定虚拟同步轴的轴位置。当使用 VAPR 语法时，设置 “已知的接口信号参考点”。

 有关轴转换功能性的详细信息，请参阅文档 “IndraMotion MTX 功能描述”。

|     |                                                    |                |
|-----|----------------------------------------------------|----------------|
| 语法: | VirtAxisPos (<Axis 1><value>,<Axis n><value>, ...) |                |
|     | 缩写形式: VAP (...)                                    |                |
|     | 其中                                                 |                |
|     | <Axis x>                                           | 虚拟轴的地址         |
|     | <Value>                                            | <Axis x> 的定位信息 |

表格 6-200: VirtAxisPos (VAP) 语法

示例: VAP (VX150) 将轴 VX 定位为 150 mm

特殊功能和限制: 不允许进行局部增量编程 (IC... )。

6.151 速度控制的诊断 "VREC\_START", "VREC\_STOP"

描述: 许多因素都会直接或间接地影响，使用 G8 速度控制器编辑自由曲面程序结果速度曲线。速度控制的诊断允许用户找出哪个变量对零件程序中指定位置上达到的路径速度具有主要影响。

对于块准备， VREC\_START 函数会记录可影响速度曲线的数据，该数据由速度控制器按块计算得出。激活期间指定日志文件的名称和存储位置（目录）以及应记录数据的物理单位。

可以离线评估创建的诊断文件。借助所记录的数据，可确定必须进行优化的参数，以实施更为有利的速度曲线。

在 VREC 激活期间，输出位于 “VREC 激活时通道轴标题极限” 下的通道中所有轴的有效轴限制值：

使用高级语言语法的 NC 功能

| 列索引: | 列标题:       | 含义:                                             |
|------|------------|-------------------------------------------------|
| 1    | SysAxNo    | 系统轴号                                            |
| 2    | MaxAxVel   | 最大轴速度                                           |
| 3    | MaxAxAccUp | 由于可以使用 NcfAxAccUpSlope NC 功能对路径加速阶段中的最大轴过冲进行编程。 |
| 4    | MaxAxAcc   | 路径减速阶段中的最大轴过冲                                   |
| 5    | MaxAxJerk  | 最大轴过冲                                           |

表格 6-201: 日志文件的结构  
后续，以下数据在标题下逐块记录“结果动态限位:”

| 列索引: | 列标题:       | 含义:                                                                                                                                                                                    |
|------|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | 方式         | NC 块中的插补距离（路径长度）                                                                                                                                                                       |
| 2    | F100       | 包含定向运动的编程速度（在 G93 和 G94 的选定单元中，与 G95 的主轴转数有关）。                                                                                                                                         |
| 3    | Fmax       | 包括定向运动的最大路径速度（有限）                                                                                                                                                                      |
| 4    | F100(TCP)  | 刀具尖端的编程速度（对于所选单元中的 G93 和 G94，对于与主轴转数相关的 G95）                                                                                                                                           |
| 5    | Fmax(TCP)  | 工具提示的最大路径速度（有限）。                                                                                                                                                                       |
| 6    | Fmax 限制器   | 将最大路径速度限制为 Fmax 的系统轴的特性和编号。特性可以是最大轴速度，最大轴加速度（Acc）和最大轴过冲（Jerk）。                                                                                                                         |
| 7    | AccUp      | 刀头的最大路径加速（有限）。如果 NC 函数 NcfAxAccUpSlope 修改了一个或多个最大轴加速度，该值只能偏离 AccDown。                                                                                                                  |
| 8    | AccDown    | 刀头的最大路径减速（有限）。                                                                                                                                                                         |
| 9    | Acc 限制器    | 将最大路径减速限制为 AccDown 的系统轴的特性和编号。特性可以是最大轴加速度（Acc）和最大轴过冲（Jerk）。                                                                                                                            |
| 10   | Vend       | 块末尾的最大路径速度或当前减速坡度，包括定向运动                                                                                                                                                               |
| 11   | Vend(TCP)  | 块末尾的工具提示的最大路径速度或当前减速坡度包括定向运动。                                                                                                                                                          |
| 12   | MaxStepVel | 在转换到下一个块时理论上可能的最大路径速度。                                                                                                                                                                 |
| 13   | 0Vel       | 是：<br>对于 NC 块，该属性已被设置为应在 V=0 时进行后续块转换。该属性被设置为例如要准确停止的块。                                                                                                                                |
| 14   | 预读状态       | 包括当前预读块编号的预读状态：<br>0：<br>虚拟块：无预读<br>1：<br>无 Look_ahead 的进给块，例如使用准停定位的块，或者后续块为 dummy 块等等。<br>2：<br>所设置的 Look_ahead 足够用于制动路径检查。<br>3：<br>所设置的 Look_ahead 足够用于制动路径检查。但在预读范围内未发现 v=0 的块过渡。 |

使用高级语言语法的 NC 功能

| 列索引: | 列标题: | 含义:                                                                             |
|------|------|---------------------------------------------------------------------------------|
| 15   | 插入状态 | 插入块状态:<br>-: 无插入块<br>1: 通用插入块<br>2: 第一分隔块（无插入块）<br>3: 分隔块（插入块）<br>4: 上一分隔块（插入块） |
| 16   | 地址   | 控制器内数控块的内存地址                                                                    |
| 17   | NC 源 | 数控块的源文本（此处: CPL 部分被初始化）                                                         |

表格 6-202: 日志文件的结构

|     |                                              |                                                                                                                                                                                                                                             |
|-----|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | VREC_START({file path/}<file name>[, Units]) |                                                                                                                                                                                                                                             |
|     |                                              | 开始速度控制的诊断记录                                                                                                                                                                                                                                 |
|     | 其中:                                          |                                                                                                                                                                                                                                             |
|     | {File path/}<File name>                      | 应该保存诊断记录的目录和文件名                                                                                                                                                                                                                             |
|     | 单位                                           | 它以下列单位存储:<br>单位 = 0 或未编程:<br>以增量为单位的距离<br>每个 Ipo 周期的增量速度<br>每个 Ipo 周期² 加速<br>每 Ipo 循环³ 的增量<br>单位 = 1 且 G70 有效:<br>距离英寸<br>英寸/分钟的速度<br>in/s² 的加速度<br>加加速度, 单位 in/s³<br>单位 = 1 且 G71 激活:<br>距离 (毫米)<br>毫米/分钟的速度<br>m/s² 的加速度<br>加加速度, 单位 m/s³ |
|     | VREC_STOP                                    | 退出诊断记录                                                                                                                                                                                                                                      |

表格 6-203: VREC\_START、VRECSTOP 的语法

示例:

```
...
N30 VREC_START (/mnt/VDIAG1.txt)
...
N110 VREC_STOP
...
N300 VREC_START (/mnt/VDIAG2.txt, 1)
...
N500 VREC_STOP
```

...  
M30

特殊功能和限制: 使用“速度控制器诊断”函数时，请确保有足够内存用于诊断文件。所需的内存容量通常约为要记录的零件程序相应部分所需内存的 3-4 倍。

6. 152 获取轴，在必要时等待 "WaitAxis、WAX"

描述: 将异步轴传递至调用通道。这会将异步轴变为同步轴。  
然后，可以使用轴的物理或逻辑轴名称在当前通道中对轴进行编程。

 有关“轴转换”功能性的详细信息，请参阅文档“IndraMotion MTX 功能描述”。

|     |                                                        |                                                                                                                                                                                                            |
|-----|--------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | WaitAxis(<PAN> <PAI>,{<LAN>}{,<PAN> <PAI>,{<LAN>}}...) |                                                                                                                                                                                                            |
|     | 缩写形式: WAX(..)                                          |                                                                                                                                                                                                            |
|     | 其中                                                     |                                                                                                                                                                                                            |
|     | <PAN>                                                  | 物理轴名称<br>指定要变换到当前通道的轴。                                                                                                                                                                                     |
|     | <PAI>                                                  | 物理轴索引<br>效果与<PAN>相同                                                                                                                                                                                        |
|     | <LAN>                                                  | 逻辑轴名称<br>如果对此编程，则要集成到当前通道的轴将被赋予逻辑名称 <LAN>。<br><LAN> 必须在参数 CHAN / Ch / Coord / Acs / ChAx / ChAxName (MP 7010 00010) (通道轴名称) 或参数 CHAN / Ch / Coord / Acs / OptAxName / OptChAxName (MP 7010 00020) (可选轴名称)。 |

表格 6-204: WaitAxis (WAX) 语法

示例: N030 WAX(YP,,ZP,Z) 必要时，块准备会等待物理轴 YP 和 ZP，直至其处于静止状态。之后将轴转换至调用通道。  
:  
当 YP 在调用通道中的地址也为 YP 时，ZP 将收到调用通道中的地址 Z。

- 特殊功能和限制:
- 如果要变换的轴尚未处于静止状态，则块准备会等待其到达静止状态。轴随后被转移。  
与函数“GetAxis”相反（参见 第 6.52 章“获取轴 GetAxis、GAX”第 247 页），不会出现任何错误消息，并且程序也不会终止。
  - 必须始终在 WaitAxis(...) 之后，且仅可在轴变换不处于活动状态时对相同块中轴的位置进行编程。
  - 要转换的轴不得与参与任何激活的监控区域（参见 第 6.7 章“区域监控 区域，ARA”第 189 页）。

6. 153 使用扩展 Sercos ID 编号写入 Sercos 参数 "WriteId3、WID3"

6. 153. 1 一般信息

描述: 它写入编程的 Sercos 驱动器参数 (S-x-xxxx-xx.xx, P-x-xxxx.xx.xx) 至一个或多个驱动器。  
除单个值外，还可写入最多 16 个值的完整值列表或最多 8 个 ID 编号的 ID 列表。

使用高级语言语法的 NC 功能

先决条件:

- 该参数必须在 Sercos 阶段 4 中有效且可编辑。
- 参数值必须处于允许的范围内。
- 驱动器必须使用 Sercos 接口连接到控制器；控制器和驱动器之间的循环通讯正在运行。



CPL 命令 “SCS3” 可帮助通过扩展 Sercos ID 编号读取 Sercos 驱动器参数。



使用 Sercos 多路复用通道可加速数据传输。



对 Sercos 传动参数进行不当或随意的修改可能会导致工件和 / 或机器的损坏，并导致危险和不可预测的机器反应。

正确使用驱动器中可用的 Sercos 变频器参数信息。对此，请参见驱动器文档。

6. 153. 2 将参数写入至一个或多个驱动器（用于主轴和异步轴）

语法:

WriteID3 (<Par>, DRIVE (<AA1>, <W1>{<AAn>, <Wn>}...))

缩写形式: WID3 (...)

其中

|               |                                                                |
|---------------|----------------------------------------------------------------|
| <Par>         | 默认参数 (S-x-xxxx. xx. xx)<br>- 或 -<br>特定产品的参数 (P-x-xxxx. xx. xx) |
| <AA1>...<AAn> | Sercos 驱动器的物理轴索引或物理轴名称。使用 DRIVE (...) 语法可最多定义 8 个驱动器。          |
| <W1>...<Wn>   | <Par> 中要写入的值                                                   |

表格 6-205: WriteId3 (WID3) 语法 2

示例:

N100 WID3 (S-0-0104.0.0, DRIVE (1, 2.22))

将数值 2.22 写入至驱动器 1 中的 S-0-0104.0.0

6. 153. 3 将参数写入至一个或多个驱动器（用于同步轴）

语法:

WriteId3 (<Par>, <SA1><W1>{, <SAn><Wn>}...)

缩写形式: WID3 (...)

其中

|               |                                                                |
|---------------|----------------------------------------------------------------|
| <Par>         | 默认参数 (S-x-xxxx. xx. xx)<br>- 或 -<br>特定产品的参数 (P-x-xxxx. xx. xx) |
| <SA1>...<SAn> | Sercos 驱动器的逻辑轴地址                                               |
| <W1>...<Wn>   | <Par> 中要写入的值。                                                  |

表格 6-206: WriteId3 (WID3) 语法 1

示例:

N100 WID3 (S-0-0159.0.0,X10,Y20,Z30)

S-0-0159.0.0 写入:  
在 X 轴的驱动器中为: 10  
在 Y 轴的驱动器中为: 20  
在 Z 轴的驱动器中为: 30

6. 153. 4 将 ID 列表中的扩展 ID 编号写入驱动器

语法:

|                                                 |                                                                       |
|-------------------------------------------------|-----------------------------------------------------------------------|
| WriteID3(<Par>,EID_LIST(<A1>,<ID1>{,<Idn>}...)) |                                                                       |
| 缩写形式: WID3(...)                                 |                                                                       |
| 其中                                              |                                                                       |
| <Par>                                           | 默认参数 (S-x-xxxx.xx.xx)<br>- 或 -<br>包含扩展 ID 列表的特定产品参数 (P-x-xxxx.xx.xx)。 |
| <A1>...<An>                                     | 物理轴索引<br>- 或 -<br>Sercos 驱动器的物理轴名称                                    |
| <ID1>...<Idn>                                   | <Par> 中最多可写入 8 个 ID 编号。                                               |

表格 6-207: WriteId3 (WID3) 语法 4

示例:

N100 WID3  
(S-0-1050.0.20,EID\_LIST(2,P-0-556  
,S-0-1050.0.6))

在轴 2 的驱动器中写入  
S-0-1050.0.20 两个 ID 编号。

- 特殊功能和限制:
- 进行轮廓加工期间不得编程函数。
  - 只能在相关驱动器静止后才可写入编程的参数。

6. 153. 5 将参数列表中的值写入驱动器

语法:

|                                         |                                                                   |
|-----------------------------------------|-------------------------------------------------------------------|
| WriteID3(<Par>,LIST(<A1>,W1{,<Wn>}...)) |                                                                   |
| 缩写形式: WID3(...)                         |                                                                   |
| 其中                                      |                                                                   |
| <Par>                                   | 默认参数 (S-x-xxxx.xx.xx)<br>- 或 -<br>包含数值列表的特定产品参数 (P-x-xxxx.xx.xx)。 |
| <A1>...<An>                             | Sercos 驱动器的物理轴索引或物理轴名称。                                           |
| <W1>...<Wn>                             | <Par> 中最多可写入 16 个数值。                                              |

表格 6-208: WriteId3(WID3) 语法 3

示例:

N100 WID3  
(P-0-4006.0.0, LIST(1, 0.5, 0.2))

将 2 个值写入驱动器 1 中的  
P-0-4006.0.0

6. 154 激活零点偏置表 "ZoTSeI、Z0S"

描述:

激活零点偏置表 (Z0 表)。Z0 表以 XML 文件形式保存在控制器的文件系统中。

使用高级语言语法的 NC 功能

|            |  |                                                                                                            |  |
|------------|--|------------------------------------------------------------------------------------------------------------|--|
| 语法:        |  | ZoTSe1({<Path>}<FileName>)                                                                                 |  |
|            |  | 缩写形式: ZOS(..)                                                                                              |  |
|            |  | 其中                                                                                                         |  |
| <Path>     |  | 有关 <FileName> 存储目录的可选路径指定。<br>如果没有规范,则在 “/数据库” 路径中进行搜索。<br>如果找不到 <FileName>,则为了能在其它目录中找到它,控制器将使用搜索路径搜索子例程。 |  |
| <FileName> |  | Z0 表的文件名,包括文件扩展名<br>含标准名称 (Z0<Number>.zot) 的表可通过编号直接激活,例如<br>ZoTSe1(5) 激活表 Z05.zot                         |  |

表格 6-209: ZoTSe1 (ZOS) 语法

|                       |                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------|
| 示例:                   |                                                                                                |
| :                     |                                                                                                |
| N030 ZOS(npvtab.zot)  | 首先在目录 “/database” 中搜索 Z0 表 “npvtab.zot”,若未在该处找到,则在子例程的搜索路径中搜索。将激活找到的第一个名为 “npvtab.zot” 的 Z0 表。 |
| N130 ZOS(/mnt/np.zot) | 在目录 “/mnt” 中搜索并激活 Z0 表 “np.zot”。若该处没有,则会出现错误消息。                                                |
| :                     |                                                                                                |

**特殊功能和限制:** 激活表时,控制器可有选择地检查通道当前轴配置是否适合表内容 (“严格赋值” 选项;使用表编辑器设置,或在表格首次生成时设置)。  
激活 “严格赋值” 选项后,如果配置不合适,则会终止程序,并生成错误消息。



请参见 “控制器操作说明”,了解创建与编辑 Z0 表的相关信息!



## 7 CPL 命令

CPL 命令的一些基本属性显示在每个命令描述之前的属性字段中。与各个命令相关的字段以灰色突出显示。

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| <type> | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

键:

- **<type>**: CPL 命令返回值的 CPL 数据类型（如果命令提供了返回值）：  
**DOUBLE, REAL, INT, BOOL, CHAR**  
对于具有不同数据类型返回值的命令，属性字段显示 **MULTYP**  
对于提供系统数据结构或子结构作为返回值的命令，属性字段显示 **STRUCT**
- **READ**: CPL 命令转换为读取控制内部日期，例如 位置，更正寄存器或表格元素
- **WRITE**: CPL 命令写入控制内部日期，例如 更正寄存器或表格元素
- **FUNCT(ion)**: CPL 命令触发计算或操作并提供返回值
- **PROC(edure)**: CPL 命令触发一个动作并且不提供任何返回值
- **WAIT**: 例如，使用 CPL 命令需要先前编程的 WAIT 进行同步，因为例如访问激活时间点的数据
- **ERRNO**: CPL 命令支持 ERRNO 参数的编程
- **!**: 使用 CPL 命令时要考虑的功能。以下说明解释了此功能

### 7.1 坐标轴和坐标

#### 7.1.1 概览

CPL 提供访问坐标轴和坐标数据的命令。

以下命令是可区分的:

- 名称和属性  
(AXINF, AXADR\$, AXP)
- 读取坐标和轴位置  
(PCS, WCS, MCS, ACS, SPOS, APOS)
- 读取测头  
(PCSPROBE, PROBE)

在零件程序中，处理程序时内插的坐标始终被编程。轴变换根据当前坐标值计算各个轴的相应命令值（空间和机器坐标）。



有关 “坐标，轴和转换” 的详细信息，请参阅文档 “IndraMotion MTX 功能描述”。

数据流程图

CPL 命令

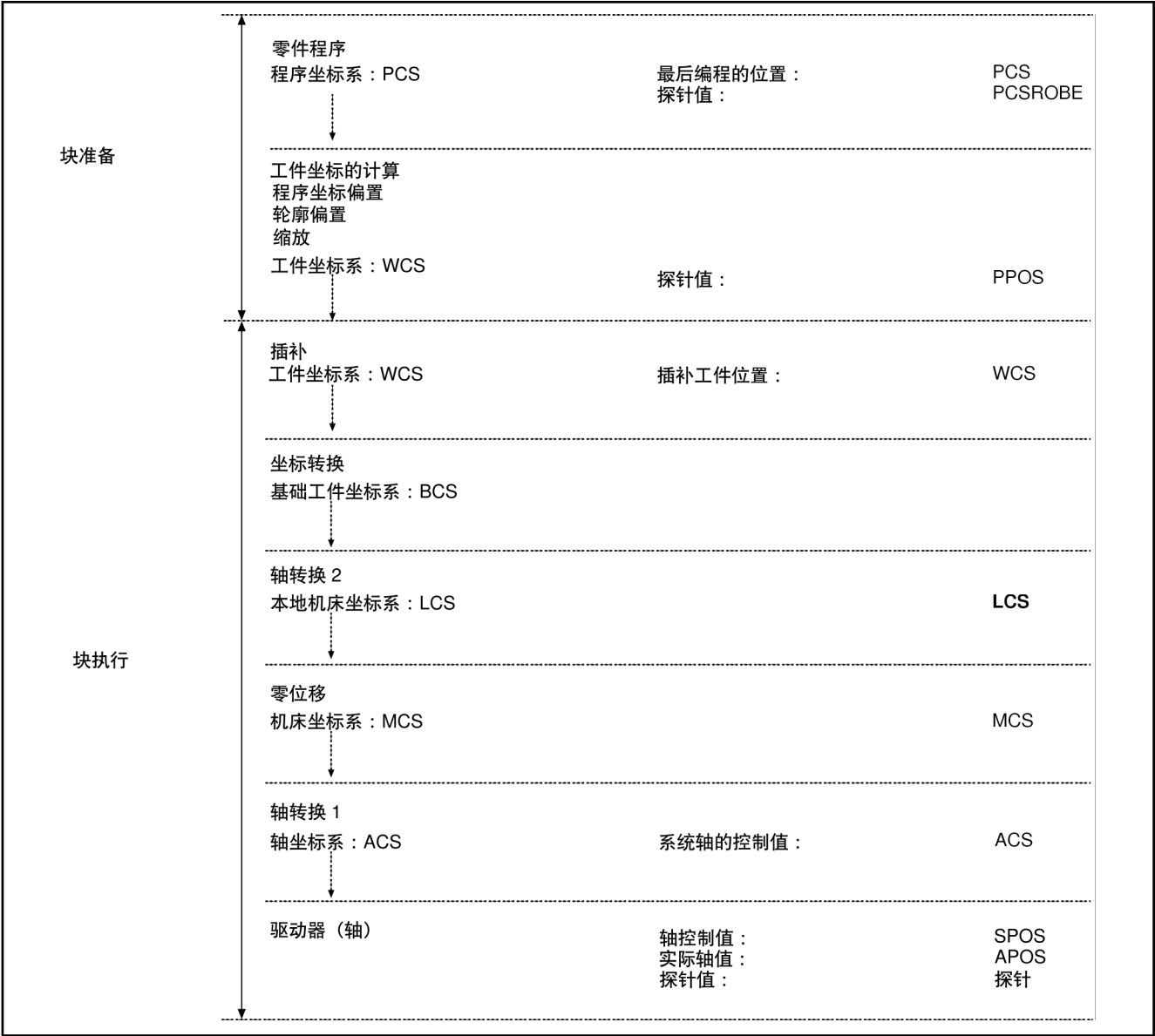


插图 7-1: 数据流程图

提供的轴和坐标位置的测量单位

|                   |                                       |
|-------------------|---------------------------------------|
| 同步线性轴和<br>平移空间坐标： | “毫米” 或 “英寸”；<br>取决于调用通道中的设置（G71，G70）。 |
| 同步旋转轴和<br>旋转空间坐标： | “度”                                   |
| 异步线性轴：            | “mm”                                  |
| 异步回转轴：            | “度”                                   |

示例： 轴名称到通道的配置和分配

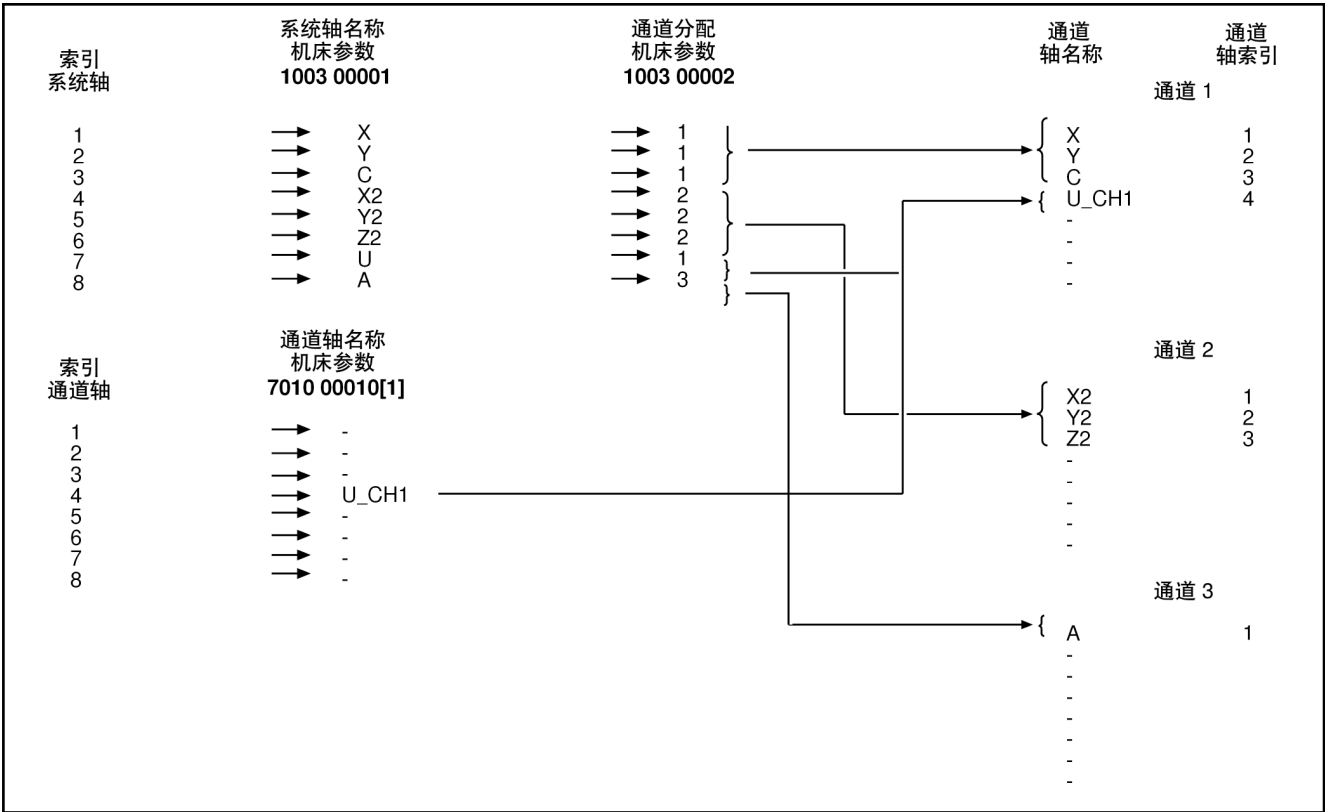


插图 7-2: 轴名称到通道的配置和分配

### 7.1.2 用于读取名称和属性的命令

**AXADR\$**

| CHAR | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

**描述:** 轴地址 - 轴和坐标地址

CPL 命令 **AXADR\$** 返回程序和机器坐标的名称以及通道和系统轴名称作为返回值。

输入参数用于:

- 系统轴索引
- SACI 指数
- 程序坐标名称对应的程序坐标索引（通道坐标索引）
- 机床坐标名称对应机床坐标索引
- 通道和系统轴名称对应的通道轴索引

AXADR\$ 命令支持 ERRNO 参数。

**语法:**

| AXADR\$(<Index> [ , [ Selection type ] [ Channel number ] ] ) |                                                       |
|---------------------------------------------------------------|-------------------------------------------------------|
| <Index>                                                       | 坐标，渠道，系统或 SACI 索引。<br>该名称根据指定的 <Selection type> 进行解释。 |

CPL 命令

|                  |                                                                                                                                                                                                                                                                                                                |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Selection type> | <p><b>可选:</b></p> <p>规定如何解释所传递的索引:</p> <p>0: 系统轴索引<br/>该命令返回相关的系统轴或 SACI 名称</p> <p>1: 通道坐标索引 (默认)<br/>该命令返回相应的程序坐标名称</p> <p>2: 机床坐标索引<br/>该命令返回相应的机器坐标名称</p> <p>3: 通道轴索引<br/>该命令返回相应的通道轴名称</p> <p>4: 通道轴索引<br/>该命令返回链接系统轴的系统轴名称</p>                                                                            |
| <Channel number> | <p><b>可选:</b></p> <p>指定的通道坐标或通道轴索引所指的通道号。</p> <p>-1: 该命令返回中断块中的有效名称 (仅用于块预运行和异步子程序)。</p> <p>0: 当前通道 (默认)</p> <p>通道号仅在与&lt;选择类型&gt; = 1、2、3、4 相关时有效。与 &lt;Selection type&gt; = 0 相关联, 只有在指定了&lt;Index&gt;的值 &lt; 0 时, 才能指定通道号, 即要访问通道中组态的工件或机床坐标名称。</p> <p>如果指定了第三方通道的通道号, 则 AXADR \$命令会返回在指定通道的激活时间点有效的名称。</p> |
| ERRNO            | <p>CPL 变量</p> <p>如果将 CPL 变量 ERRNO 编入参数列表中的任何位置, 则命令不产生任何内部运行时错误。该错误由变量的相应返回值返回。</p> <p><i>以下为可能的返回值:</i></p> <ul style="list-style-type: none"><li>0: 访问正常</li><li>&lt;0: 错误: 有关错误说明, 请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页</li></ul> <p>如果未输入 CPL 变量 ERRNO, 则出现内部访问错误时, 会自动产生运行时错误。</p>          |

表格 7-1: AXADR\$ 变量

语法示例:

|                   |                       |
|-------------------|-----------------------|
| AXADR\$ (2, 0)    | 返回第二个系统轴的系统轴名称        |
| AXADR\$ (3)       | 返回当前通道中第三个程序坐标的程序坐标名称 |
| AXADR\$ (1, 2, 2) | 返回通道 2 中第一个机床坐标的名称    |
| AXADR\$ (4, 3, 4) | 以 4 返回第四个通道轴的通道轴名称    |
| AXADR\$ (5, 4)    | 提供当前通道中第五通道轴的系统轴名称    |

AXINF

描述 轴信息

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

CPL 命令 **AXINF** 将多轴和坐标特定的属性作为返回值返回。

可能的输入参数:

- 系统轴名称
- 系统轴号
- 通道轴名称和（可选）通道索引 - 或 -
- 通道轴索引和（可选）通道索引。

以下坐标轴和坐标特定信息可以通过 *AXINF* 命令确定:

- 一个通道或系统轴的系统轴索引。
- 一个轴的运动类型（线性，旋转，无限线性模）。
- 虚拟轴。
- 轴和坐标含义（X，Y，Z）
- 直径编程

AXINF 命令支持 ERRNO 参数。

语法:

|                                                                    |                                                                                                                                            |
|--------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| AXINF(<Axis> [, Coordinate type [, Channel number, <Selection>]]]) |                                                                                                                                            |
| <Axis>                                                             | 通道轴号或轴名称<br>系统轴名称或通道轴名称可以输入。该名称根据指定的<br><Selection type> 进行解释。                                                                             |
| <Coordinates>                                                      | <b>可选:</b><br>定义是否将编程名称解释为系统轴名称或通道轴名称。<br>0: 系统<br>1: 通道（默认）                                                                               |
| <Channel number>                                                   | <b>可选:</b><br>通道号指的是指定的通道轴索引或通道轴名称。<br>0: 当前通道（默认）<br>如果<Selection type> = 1, 则允许通道号码。<br>如果<Selection type> 为 1 且没有编程通道号, 则定义当前<br>通道的数据。 |

CPL 命令

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Selection> | <p><b>可选:</b></p> <p>选择由 AXINF 提供的轴专用信息:</p> <p>0: 轴和坐标索引 (默认)</p> <p>1: 轴的运动类型</p> <p><i>返回值:</i></p> <ul style="list-style-type: none"><li>1: 线性轴</li><li>2: 模态轴</li><li>3: 旋转轴</li><li>4: 线性模轴</li></ul> <p>2: 虚拟轴</p> <p><i>返回值:</i></p> <ul style="list-style-type: none"><li>0: 实轴</li><li>1: 虚拟轴</li></ul> <p>11: 轴描述</p> <p><i>返回值:</i></p> <ul style="list-style-type: none"><li>0: 没有轴的含义</li><li>1: 含义 X</li><li>2: 含义 Y</li><li>3: 含义 Z</li></ul> <p>12: 直径编程</p> <p><i>返回值:</i></p> <ul style="list-style-type: none"><li>0: 没有直径编程</li><li>1: 直径编程</li></ul> |
| ERRNO       | <p>CPL 变量</p> <p>如果将 CPL 变量 ERRNO 编入参数列表中的任何位置, 则命令不产生任何内部运行时错误。该错误由变量的相应返回值返回。</p> <p><i>以下为可能的返回值:</i></p> <ul style="list-style-type: none"><li>0: 访问正常</li><li>&lt;0: 错误: 有关错误说明, 请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页</li></ul> <p>如果未输入 CPL 变量 ERRNO, 则出现内部访问错误时, 会自动产生运行时错误。</p>                                                                                                                                                                                                                                                                      |

表格 7-2: AXINF 语法

|       |                       |                         |
|-------|-----------------------|-------------------------|
| 语法示例: | AXINF("VA", 0)        | 返回系统轴 "VA" 的系统轴索引       |
|       | AXINF("Z")            | 提供当前通道中通道轴 "Z" 的系统轴索引   |
|       | AXINF("Y", 1, 5)      | 返回通道 5 中通道轴 "ZY" 的系统轴索引 |
|       | AXINF(4, 1, 0, ERRNO) | 返回当前通道中第四个通道轴的系统轴索引     |
|       |                       | 变量 ERRNO 中返回错误代码。       |

AXP

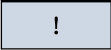
|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** 轴编程 – 变量轴地址  
AXP 命令有利于零件和测量程序的平面独立写入。

**语法**

|                                                                                              |                                                                                                                              |
|----------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| AXP(<Axis number>,<Path information>[,<Axis type>])<br>该命令应用于数控块。它必须在方括号 “[ ]” 中并且被编程而不是地址值。 |                                                                                                                              |
| <Axis number>                                                                                | 工件坐标或系统轴的索引                                                                                                                  |
| <Path information>                                                                           | 路径信息变量或值                                                                                                                     |
| <Axis type>                                                                                  | <b>可选:</b><br>确定如何解释在 <Axis type> 下编程的索引:<br>0:<br>系统轴索引<br>1:<br>通道的工件坐标索引，程序当前在该通道中被处理。<br>如果没有明确的编程，则将 <Axis type> 设置为 1。 |

表格 7-3: 语法 AXP



他只能使用方括号 [ ] 在数控程序段内对 AXP 命令进行编程。在“手动数据输入”操作模式下调用是不可能的！

**示例:** 子程序:

|                                                                            |                             |
|----------------------------------------------------------------------------|-----------------------------|
| 10 A%=P1% : B%=P2%                                                         | 将轴号从 P1% 和 P2% 转换到 A% 和 B % |
| 20 C=P3:D=P4:RA=P5                                                         | 转换 G2 的命令值                  |
| 30 E=0                                                                     | G20 处极点的常量                  |
| N40 G20 [AXP(A%,E)][AXP(B%,E)]                                             | 用 G20 进行平面切换;<br>杆上 0,0     |
| N50 G2 [AXP(A%,C)][AXP(B%,D)]                                              | 用 G2 进行半径编程                 |
| R[RA]] F1000                                                               |                             |
| 用 A% 和 B%定义平面。用 G20 进行后续的平面切换。最后，轴以 F1000 移动通过由变量 C, D （终点）和 RA （半径）定义的圆弧。 |                             |

7. 1. 3

用于读取坐标和轴位置的命令

一般信息

- 注意:*
- 对于空间和机床坐标，通道中的坐标索引总是固定的。
  - 但是，可以使用轴传送数控函数接受坐标或将坐标传送到通道。因此，通道中其他坐标的坐标索引可能会发生变化  
由于可以预设系统轴索引，所以也可以使用固定索引
  - 如果预设的是未配置的系统轴，则会显示运行时错误。
- 所描述的命令分别以相似的参数运行。因此，他们在下面解释。





以下情况适用:

- 如果系统耦合对该轴无效, 则 ACS 结果对应于 “SPOS” 命令。
- 如果在程序段执行时确定返回值, 则必须先在一个程序段中编程 “WAIT” (另请参见 第 3.19.2 章 “块准备的同步功能” 第 79 页 )。  
没有 WAIT, 没有提供唯一可预测的值, 因为不知道块执行“滞后”块的准备程度。
- 访问第三方通道的轴值时, 可能需要采取同步措施来测量已定义的位置。

语法:

|                                              |                                                                                                                   |
|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| ACS (<Axis>[, <SelectionType>[, <Channel>]]) |                                                                                                                   |
|                                              | (关于参数, 请参阅 第 7.1.3 章 “用于读取坐标和轴位置的命令” 第 377 页 ,<br>要编程 ERRNO, 请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页 ) |

表格 7-4: 语法 ACS

语法示例:

|                                                    |                                          |
|----------------------------------------------------|------------------------------------------|
| ACS ("X", ERRNO)                                   |                                          |
| ACS ("X", 1)                                       |                                          |
| ACS ("X", 1, 1)                                    |                                          |
| 通道 2 根据示例配置 (请参阅 插图 7-2 “轴名称到通道的配置和分配” 第 373 页 ) : |                                          |
| N10 G0 G90 X2=150 Y2=100                           |                                          |
| 20 WAIT                                            |                                          |
| 30 X2VALUE=ACS ("X2", ERRNO)                       | 通道轴 X2 的当前插补轴位置分配给 X2VALUE。              |
| 31 IF ERRNO <>0 THEN                               |                                          |
| 32 PRN#(0, "error:", ERRNO)                        |                                          |
| 33 ENDIF                                           |                                          |
| N40 G91 X2=10 Y2=10                                |                                          |
| 50 WAIT                                            |                                          |
| 60 X2VALUE=ACS (1, 1)                              | 为 X2VALUE 分配当前通道 (X2 轴) 的第一个通道轴的当前插补轴位置。 |
| 70 Y2VALUE=ACS ("Y2", 1, 2)                        | 第二个通道的通道轴 Y2 的当前插补轴位置分配给 Y2VALUE。        |
| 80 XVALUE=ACS (1, 0)                               | 为 XVALUE 分配 1 的当前插补轴位置。系统轴已分配。           |

APOS

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

描述:

**实际位置** – 读取实际轴的值  
基于轴坐系统, 提供轴的当前 **实际位置** 。  
所有的系统轴均可使用各自的系统轴名称或系统轴索引提供地址。  
以下情况适用:

- 如果在程序段执行时确定返回值, 则必须先在一个程序段中编程 WAIT (另请参见 第 章 “WAIT (无参数)” 第 79 页 )。  
没有 WAIT, 没有提供唯一可预测的值, 因为不知道块执行“滞后”块的准备程度。

CPL 命令

- 访问第三方通道的轴值时，可能需要采取同步措施来测量已定义的位置。

|     |               |                                  |
|-----|---------------|----------------------------------|
| 语法: | APOS (<axis>) |                                  |
|     | <Axis>        | 系统轴的索引或名称<br>对未组态的轴进行编程会导致运行时错误。 |

表格 7-5: 语法 APOS

无法对 ERRNO 编程。

示例: 根据示例配置的通道（请参阅 插图 7-2 “轴名称到通道的配置和分配” 第 373 页）：

|                    |                                         |
|--------------------|-----------------------------------------|
| 30 AKT4=APOS (4)   | 第四系统轴（通道 2 中的 X2 轴）的当前实际轴值分配给变量 AKT4。   |
| :                  |                                         |
| 50 AKT8=APOS ("A") | 第 8 个系统轴的当前实际轴值（通道 3 中的 a 轴）分配给变量 AKT8。 |
| :                  |                                         |

MCS

| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|
|--------|------|-------|-------|------|------|-------|---|

描述: **机床坐标系** – 读取机器坐标  
为机床坐标提供当前机床位置（MCS），无需进行直线度和角度误差修正。  
第三方渠道坐标也可以访问。

以下情况适用:

- 如果在程序段执行时确定返回值，则必须先在一个程序段中编程 WAIT（另请参见 第 章 “WAIT（无参数）” 第 79 页）。  
没有 WAIT，没有提供唯一可预测的值，因为不知道块执行“滞后”块的准备程度。
- 当访问第三方通道的坐标值时，可能需要采用同步测量来测量已定义的位置。访问第三方通道的坐标值时，可能需要采取同步措施来测量已定义的位置。

|     |                                                      |                                                                                                               |
|-----|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| 语法: | MCS (<Coordinate> [, <Selection type>[, <Channel>]]) |                                                                                                               |
|     |                                                      | （关于参数，请参阅 第 7.1.3 章 “用于读取坐标和轴位置的命令” 第 377 页，<br>要编程 ERRNO，请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页） |

表格 7-6: 语法 MCS

语法示例: MCS ("X", ERRNO)  
MCS ("X", 1)  
MCS ("X", 1, 1)  
通道 2 根据示例配置（请参阅 插图 7-2 “轴名称到通道的配置和分配” 第 373 页）：

```
N10 G0 G90 X2=150 Y2=100
20 WAIT
```

30 X2WERT=MCS ("X2",ERRNO)

31 IF ERRNO <>0 THEN

32 PRN# (0,"error:",ERRNO)

33 ENDIF

N40 G91 X2=10 Y2=10

50 WAIT

60 X2VALUE=MCS (1,1)

70 Y2VALUE=MCS ("Y2",1,2)

80 XVALUE=MCS (1,0)

X2 坐标的当前内插机床位置分配给 X2WERT。

当前通道的第一个机床坐标（X2 坐标）的当前插补机床位置分配给 X2VALUE。

第二个通道的 Y2 坐标的当前内插位置被分配 Y2VALUE。

为 XVALUE 分配 1 的当前插补轴位置。系统轴已分配。

只有在系统轴与机床坐标相同的情况下才允许此访问。

PCS

描述:

**程序坐标系** – 读取编程的坐标  
返回坐标的最后编程值。  
*以下情况适用:*

- 如果在编入坐标后以及查询位置前转换工件坐标系，所提供的值已把新工件坐标系考虑在内。
- 只有自己的通道的坐标可以被查询。

语法:

|                                        |  |  |  |                                                                                                                |  |  |  |
|----------------------------------------|--|--|--|----------------------------------------------------------------------------------------------------------------|--|--|--|
| PCS (<Coordinate> [, <SelectionType>]) |  |  |  |                                                                                                                |  |  |  |
|                                        |  |  |  | (关于参数, 请参阅 第 7.1.3 章 “用于读取坐标和轴位置的命令” 第 377 页 ;关于编程 ERRNO, 请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页 ) |  |  |  |

表格 7-7: 语法 PCS

语法示例:

PCS ("X",ERRNO)

PCS ("X",1)

通道 2 根据示例配置（请参阅 插图 7-2 “轴名称到通道的配置和分配” 第 373 页 ）：

N1 G0 G90 X2=150 Y2=100

02 X2VALUE=PCS (1,ERRNO)

03 IF ERRNO <>0 THEN

04 PRN# (0,"error:",ERRNO)

05 ENDIF

N10 G91 X2=10

011 X2VALUE=PCS (1,1)

N12 X2=5 Y2=10

当前通道中第一个坐标的编程绝对位置（X2VALUE = 150）被分配给 X2VALUE。

当前通道中第一个坐标的编程绝对位置（X2VALUE = 160）被分配给 X2VALUE。



没有 WAIT，没有提供唯一可预测的值，因为不知道块执行“滞后”块的准备程度。

- 当访问第三方通道的坐标值时，可能需要采用同步测量来测量已定义的位置。访问第三方通道的坐标值时，可能需要采取同步措施来测量已定义的位置。

语法:

|                                                     |                                                                                                                   |
|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| WCS (<Coordinate> [, <SelectionType>[, <Channel>]]) |                                                                                                                   |
|                                                     | (关于参数, 请参阅 第 7.1.3 章 “用于读取坐标和轴位置的命令” 第 377 页 ,<br>要编程 ERRNO, 请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页 ) |

表格 7-9: WCS 语法

语法示例:

```
WCS ("X", ERRNO)
WCS ("X", 1)
WCS ("X", 1, 1)
通道 2 根据示例配置 (请参阅 插图 7-2 “轴名称到通道的配置和分配” 第 373 页) :
```

|                           |                                        |
|---------------------------|----------------------------------------|
| 10 WAIT                   |                                        |
| 20 Z2POS=WCS(3,1,2,ERRNO) | 第二个通道 (Z2 轴) 的第三个坐标的当前内插工件位置分配给 Z2POS。 |
| 21 IF ERRNO <>0 THEN      |                                        |
| 22 PRN#(0,"Error:",ERRNO) |                                        |
| 23 ENDIF                  |                                        |
| 30 WAIT                   |                                        |
| 110 YPOS = WCS("Y")       | 当前通道的当前插补的工件位置被分配给 YPOS。               |
| 120 XPOS = WCS(1)         | 当前通道第一个坐标的当前插补工件位置分配给 XPOS。            |

7.1.4 读取测量值的命令

PCSPROBE

| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|
|--------|------|-------|-------|------|------|-------|---|

描述:

程序坐标系 PROBE – 程序坐标中的测量值  
数控函数之后使用 PCSPROBE() :

- G75.1 或 G75 (移动至第一个测量探头。)
- G75.2 (移动至第二个测量探头。)
- FME (飞行测量)
- FSP (在固定挡块处车辆)
- MOB (在块上测量)

在以下情况下 (举例), PCSPROBE() 会生成运行时错误:

- 查询一根异步轴
- 查询第三方通道的同步轴
- 语法错误...

ERRNO 可以编程. 请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页 。

CPL 命令

|     |                                            |                                      |
|-----|--------------------------------------------|--------------------------------------|
| 语法: | PCSPROBE (<Coordinate> [,<SelectionType>]) |                                      |
|     | <Coordinate>                               | 参考 第 7.1.3 章 “用于读取坐标和轴位置的命令” 第 377 页 |
|     | <Selection type>                           |                                      |

表格 7-10: PCSPROBE 语法



在 PCSPROBE 之前，在数控函数“测头”（G75.x）和“固定点停止测量”（FSP）之后，它不能与 WAIT 命令进一步同步。CPL 命令 SD(9) 用于查询测量是否成功以及测量结果是否存在！

示例: 通道 1 根据示例配置（请参阅 插图 7-2 “轴名称到通道的配置和分配” 第 373 页）：

```
N10 G75.1 X100 Y100 Z50

20 IF SD(9)=1 THEN

N30 (MSG, 测头 1 未触发!)

40 GOTO .ERROR

50 ELSE

60 测量位置的 z 坐标值被分配给变量 "ZMEAS"
ZMEAS=PCSPROBE("Z",ERRNO);

61 IF ERRNO <>0 THEN

62 GOTO .Error

63 ENDIF

70 ENDIF
```

特殊特性: PCSPROBE ( ) 检查计算测量值所需的轴的所有测量值是否可用，并考虑以下数控函数。

- 轴转换 (CRD)
- 坐标转换 (BCR, G152 ... G159)
- 旋转 (ROT)

缺失的测量轴值，如果编程，则通过 ERRNO 45 或 46 报告。

PPOS



从 MTX 14VRS 开始，测量的位置总是可以使用 PCSPROBE ( ) 来确定。  
在新的应用程序中，请仅使用 PCSPROBE ( ) 和 PROBE ( ) 命令。

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

描述: 探头位置

数控函数之后使用 PROS()：

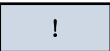
- G75.1 或 G75（移动至第一个测量探头。）
- G75.2（移动至第二个测量探头。）
- FME（飞行测量）
- FSP（在固定挡块处车辆）

例如，PPOS ( ) 会生成运行时错误：

- 查询一根异步轴
  - 查询第三方通道的同步轴
  - 语法错误…
- 无法对 ERRNO 编程。

|                  |  |                                      |
|------------------|--|--------------------------------------|
| 语法:              |  | PPOS (<Axis> [, <SelectionType>])    |
| <Axis>           |  | 参考 第 7.1.3 章 “用于读取坐标和轴位置的命令” 第 377 页 |
| <Selection type> |  |                                      |

表格 7-11: PPOS 语法



在 PROS 之前，在数控函数“测头”（G75.x）和“固定点停止测量”（FSP）之后，它不能与 WAIT 命令进一步同步。CPL 命令 SD(9) 用于查询测量是否成功以及测量结果是否存在！

示例: 通道 3 根据示例配置（请参阅 插图 7-2 “轴名称到通道的配置和分配” 第 373 页）：

```
N10 G75.2 A250 F500
20 IF SD(9) = 1 THEN
N30 (MSG, 测头 2 未触发！)
40 GOTO .ERROR
50 ELSE
60 AMEAS = PPOS(1,1);
70 ENDIF
```

通道的第一个通道轴的测量值被分配给变量“AMEAS”。

- 特殊特性: PPOS 考虑以下更正：
- 轴零点偏置 (G54...G59)
  - 工具更正 (G48, ED)
  - 编程坐标偏置 (Trans, ATrans)
  - 丝杠误差补偿和交叉补偿

以下内容不予考虑：

- 轴变换 (Coord)
- 坐标转换 (BcsCorr, G152 - G159)
- 缩放 (PoleSet, Mirror, Scale, Rotate)

没有任何轴或坐标转换，所提供的值是指最新编程的工件坐标系（WCS）。

PROBE

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

- 描述: PROBE – 驱动器的测量值
- 数控函数之后使用 PROBE()：
- G75.1 或 G75（移动至第一个测量探头。）
  - G75.2（移动至第二个测量探头。）
  - FME（飞行测量）
  - FSP（在固定挡块处车辆）





SPADR\$ 命令支持 ERRNO 参数。

语法:

| SPADR\$( <Index> [ , [ <Selection type> ] [ , <Channel number> ] ] ) |                                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Index>                                                              | 通道主轴或系统主轴索引。<br>该名称根据指定的 <Selection type> 进行解释。                                                                                                                                                                                                                                 |
| <Selection type>                                                     | <b>可选:</b><br>规定如何解释所传递的索引:<br>0: 系统主轴索引<br>该命令返回相应的系统主轴名称<br>1: 通道主轴索引 (默认)<br>该命令返回相应的通道主轴名称<br>2: 通道主轴索引<br>该命令返回链接系统主轴的系统主轴名称                                                                                                                                               |
| <Channel number>                                                     | <b>可选:</b><br>指定的通道主轴索引引用的通道号。<br>-1: 该命令返回中断块中的有效名称 (仅用于块预运行和异步子程序)。<br>0: 当前通道 (默认)<br>频道号码只有在 <SelectionType> = 1 和 2 时才被允许。<br>如果指定了第三方通道的通道号, 则 SPADR\$ 命令会返回在指定通道的激活时间点有效的名称。                                                                                             |
| ERRNO                                                                | CPL 变量<br>如果将 CPL 变量 ERRNO 编入参数列表中的任何位置, 则命令不产生任何内部运行时错误。该错误由变量的相应返回值返回。<br><i>以下为可能的返回值:</i> <ul style="list-style-type: none"><li>0: 访问正常</li><li>&lt;0: 错误: 有关错误说明, 请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页</li></ul> 如果未输入 CPL 变量 ERRNO, 则出现内部访问错误时, 会自动产生运行时错误。 |

表格 7-13: 语法 SPADR\$

语法示例:

|                 |                      |
|-----------------|----------------------|
| SPADR\$ (2,0)   | 返回第二个系统主轴的系统轴名称      |
| SPADR\$ (3)     | 返回当前通道中第三个通道轴的通道主轴名称 |
| SPADR\$ (1,2,4) | 返回通道 4 中第一个通道轴的系统轴名称 |
| SPADR\$ (5,2)   | 提供当前通道中第五通道轴的系统轴名称   |

7.3更正和输入帮助

7.3.1零点偏置

一般信息

以下 CPL 命令允许对 Z0 表进行读取、创建和修改。

CPL 命令

ZOTCR

**描述:** 零偏置表 CReate - 创建 Z0 表

在 MTX 文件系统内没有修正记录的情况下，创建一个 XML 零点偏置表。

|            |                                   |                                                                                                                                                                                                                                                                          |  |  |  |  |  |
|------------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|--|
| <b>语法:</b> | ZOTCR(<Table>,<Channel/Template>) |                                                                                                                                                                                                                                                                          |  |  |  |  |  |
|            | <Table>                           | 文件扩展名为 “.zot”的 XML 表的名称（如果适用）包括路径规范<br>如果没有指定路径，则在根目录中创建该表。                                                                                                                                                                                                              |  |  |  |  |  |
|            | <Channel/<br>template>            | 通道号或 XML 表格模板的名称（如果适用）包括路径规范<br>没有指定路径时，使用根目录中的表。                                                                                                                                                                                                                        |  |  |  |  |  |
|            | ERRNO                             | CPL 变量<br><br>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。<br><i>以下为可能的返回值:</i> <ul style="list-style-type: none"><li>0: 访问正常</li><li>&lt;0: 错误（有关详细的错误描述，请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页）。</li></ul><br>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。 |  |  |  |  |  |

表格 7-14: 语法 ZOTCR

**语法示例:**

```
ZOTCR("/usr/user/tab.zot",1)
```

使用第一个通道的每个轴的列创建一个新的 “/us/user/tab.zot” 表。

ZOCINS

**描述:** Zero Offset Column INSert - 将轴插入 Z0 表中

在 XML 零点偏移表中的现有列前插入一个新轴。新轴包含与表格第一个轴相同数量的修正集。该值被预设为 0.0。

|            |                                                                                                |                                             |  |  |  |  |  |
|------------|------------------------------------------------------------------------------------------------|---------------------------------------------|--|--|--|--|--|
| <b>语法:</b> | ZOCINS(<Table>,<Position>,<AxisName><br>[, [<AxisType>]][, [<AxisChannel>]][, <PosChannel>]]]) |                                             |  |  |  |  |  |
|            | <Table>                                                                                        | XML 表的名称，其在适用情况下包括路径说明<br>没有指定路径时，使用根目录中的表。 |  |  |  |  |  |
|            | <Position>                                                                                     | 插入位置的坐标名称或列索引                               |  |  |  |  |  |
|            | <Axis name>                                                                                    | 新表列的轴名称                                     |  |  |  |  |  |
|            | <Axis type>                                                                                    | 新轴的可选轴类型<br>0: 线性轴（默认值）<br>1: 旋转轴           |  |  |  |  |  |
|            | <Axis Channel>                                                                                 | 新轴的通道号：<br>默认: 0                            |  |  |  |  |  |

|              |                                                                                                                                                                                                                                                                                |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <PosChannel> | 如果指定了坐标名称，则在插入位置处的坐标的通道号。否则，没有意义。                                                                                                                                                                                                                                              |
| ERRNO        | <p>CPL 变量</p> <p>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。</p> <p>以下为可能的返回值：</p> <ul style="list-style-type: none"><li>0: 访问正常</li><li>&lt;0: 错误（有关详细的错误描述，请参阅 第 3.14.5 章“评价 CPL 中功能错误的变量 ERRNOERRNO”第 69 页）。</li></ul> <p>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。</p> |

表格 7-15: 语法 ZOCINS

语法示例：

|                                       |                                                                                |
|---------------------------------------|--------------------------------------------------------------------------------|
| ZOCINS("/usr/user/tab.zot","X","Y")   |                                                                                |
|                                       | 在名称为“X”表列前插入一个名为“Y”的“/usr/user/tab.zot”表中的新轴。                                  |
| <b>具有轴名称的变量</b>                       |                                                                                |
| ZOCINS("/ZOT_Trafo.zot","X","U",,1,1) |                                                                                |
|                                       | 在表“/ZOT_Trafo.zot”的表/ ZOT_Trafo.zot 中插入一个新的线性轴（对于通道 1），列名称“U”位于表列前，名称“X”为通道 1； |
|                                       | 输入此命令后，在表格中搜索名称为“X”和通道 1 的轴。新的直线轴“U”插入通道 1 的位置。                                |

**具有插入位置列索引的变量**

|                                     |                                   |
|-------------------------------------|-----------------------------------|
| ZOCINS("/ZOT_Trafo.zot",1,"U",1,,2) |                                   |
|                                     | 将新回转轴（通道 2 的）插入                   |
|                                     | 表列 1 中列名称为“U”的表“/ZOT_Trafo.zot”中。 |

ZOCDEL

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

描述： 零偏置列 DElete - 删除 ZO 表中的轴  
从 xml 零偏移量表删除一个轴。

|            |                                             |
|------------|---------------------------------------------|
| 语法：        | ZOCDEL(<Table>,<Position>[,<Channel>])      |
| <Table>    | XML 表的名称，其在适用情况下包括路径说明<br>没有指定路径时，使用根目录中的表。 |
| <Position> | 要删除轴的轴名称或列索引                                |

|           |                                                                                                                                                                                                                                                                 |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Channel> | 要删除轴的通道号<br>默认: 0                                                                                                                                                                                                                                               |
| ERRNO     | CPL 变量<br>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置, 则不会产生运行时错误。错误由变量的相应值提供。<br>以下为可能的返回值: <ul style="list-style-type: none"><li>0: 访问正常</li><li>&lt;0: 错误 (有关详细的错误描述, 请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页 )。</li></ul> 如果未指定 CPL 变量 ERRNO, 则访问错误时会产生运行时错误。 |

表格 7-16: 语法 ZOCDEL

语法示例:

```
ZOCDEL("/usr/user/tab.zot","X")
```

删除表 “/usr/user/tab.zot” 中列名为 “X”的轴。

**具有要删除轴轴名称的变量**

```
ZOCDEL("/ZOT_Trafo.zot","X",1)
```

删除表 “/ ZOT\_Trafo.zot” 中列名为 “X”的轴。

输入该命令后, 在表格中搜索名称为 “X” (在通道 1 中) 的轴并将其删除。

**带有要删除轴的列索引的变体**

```
ZOCDEL("/ZOT_Trafo.zot",1)
```

删除表 “/ZOT\_Trafo.zot”中列索引为 1 的轴。

输入该命令后, 表中的第一个轴被删除。

ZOT

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

描述:

**零偏置表** – 读取和写入访问 Z0 表

读取和写入 IndraMotion MTX 文件系统中任何 XML 零点偏置表的访问权限。只能访问各个元素。写入时, 也可以指定增量修改。



ZOT 命令直接写入指定的表中。如果这是一个有效表格, 写入的校正值只有在下一个数控程序段时才会生效。

**使用功能 “异步子程序” (ASUP), “离开和返回轮廓”, “删除待移动距离” 和 “覆盖”:**

所有已准备好的数控程序段都从这些功能的中断点重新准备好。如果已经准备好的数控程序段的修正值在此范围内随 ZOT 发生变化, 则更改的值现在适用于要准备的所有数控程序段。

此外, 已更改的校正值在被调用的异步子程序内或在离开返回轮廓的逼近运动中已处于激活状态。

**如果不需要这种行为, 请在每个 ZOT 之前编程 “WAIT” 命令!**

可选地, 可以使用 /SysCorrRefrDis 系统日期来全局地抑制表和校正寄存器的刷新机制。在此情况下, 更改后的修正值只有在下一次 Z0 编程或下一个表格激活后才会生效。

语法:

|                                                                                             |                                                                                                                                                      |
|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| ZOT (<ColumnSelection>, <ZO Code>[, [<ZO Bank>]][, [<Table>]<br>[, <Unit>] [, <Channel>]]]) |                                                                                                                                                      |
| <Column selection>                                                                          | 机床坐标（轴）的名称；在表中以引号或列索引编程。<br><i>在写作时，这是可能的</i> <ul style="list-style-type: none"><li>覆盖表值或</li><li>当 &lt;Column selection&gt; 以负号开始时，输入附加值。</li></ul> |
| <ZO code>                                                                                   | 54: 1. 零点偏置<br>55: 2. 零点偏置<br>56: 3. 零点偏置<br>57: 4. 零点偏置<br>58: 5. 零点偏置<br>59: 6. 零点偏置                                                               |
| <ZO bank>                                                                                   | ZO 银行指数 (1 ... 5)<br>默认值: 1                                                                                                                          |
| <Table>                                                                                     | 如果需要，可以使用绝对路径信息或增量路径信息（表扩展名 ".zot" 不必编程），ZO 表的名称。<br>如果没有路径信息，则在配置的搜索路径中搜索。<br>对于具有默认名称（例如 Z01 和 Z02）的表格，只能指定数值。<br>如果没有表名，则访问最近激活的表。                |
| <Unit>                                                                                      | 0 或 "MM": mm<br>1 或 "英寸": 英寸<br>在写访问期间，分配的值将以指定的单位进行解释。在读访问期间，该值被转换为指定的单位。<br>默认值: 毫米                                                                |

CPL 命令

|           |                                                                                                                                                                                                                                                                                                             |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Channel> | 通道分配 (0..60)<br><br>仅在 <Column selection> 已被指定为机床坐标的名称时，才<br>可对参数 <Channel> 进行指定。<br><br>如果参数 <ColumnSelection> 包含列索引且指定了 <Channel><br>, 则会显示错误消息。<br><br>可以通过通道分配的方式来区分具有相同名称的 Z0 表的特定<br>通道坐标。<br><br>默认:<br>根据 <ColumnSelection>访问第一列。忽略通道条目。<br>使用 <ColumnSelection> 和 <Channel> 找到的第一列。<br>如果不可用: 错误消息 |
| ERRNO     | CPL 变量<br><br>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不<br>会产生运行时错误。错误由变量的相应值提供。<br><br><i>以下为可能的返回值:</i> <ul style="list-style-type: none"><li>0: 访问正常</li><li>&lt;0: 错误 (有关详细的错误描述, 请参阅 第 3.14.5 章<br/>“评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69<br/>页 )。</li></ul><br>如果未指定 CPL 变量 ERRNO, 则访问错误时会产生运行时错<br>误。            |

表格 7-17: ZOT 语法

|       |                                                    |                                                                     |
|-------|----------------------------------------------------|---------------------------------------------------------------------|
| 语法示例: | ZOT ("X", 54)                                      | 对通道的最后激活 Z0 表中的轴/机床坐标 "X" 的 G54<br>偏置进行访问。                          |
|       | ZOT (2, 55, 2)                                     | 对通道的最后激活 Z0 表中的第二次输入的轴/机床坐<br>标 G55.2 偏置进行访问。                       |
|       | ZOT (3, 57, , "V1")                                | 访问 Z0 表 "V1.zot"中第三个输入的轴/机床坐标的<br>G57 偏置。                           |
|       | ZOT ("Z", 58, 4, 21, 1)                            | 对 Z0 表 "Zo21.zot" 中的轴/机床坐标 "Z"的 G58.4<br>偏置以英制访问。                   |
|       | ZOT ("Y", 59, 5, "/mnt/esmuser/de/HHGENIUS", "MM") | 通过目录 "/mnt/esmuser/de"中 "HHGENIUS"表中的轴/<br>机床坐标 "Y"的 G59.5 偏置的公制访问。 |
|       | ZOT ("Y", 56, 1, "/mnt/ZOF", "MM", 2)              | 通过指令 "/mnt" 中的表 "ZOF" 中分配给通道 2 的轴/<br>机床坐标 "Y" 的 G56.1 偏置的公制访问。     |

ZOV

| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|
|--------|------|-------|-------|------|------|-------|---|

描述: 零偏置值 - 有效的 Z0 值  
提供机床坐标（轴）的最新编程和因此有效的 Z0 值的总和，或者可选地提供  
单个 Z0 库的有效值。

语法:

|                                     |                                                     |
|-------------------------------------|-----------------------------------------------------|
| ZOV(<Axis selection>[, <Z0 bank>] ) |                                                     |
| <Axis selection>                    | 机床坐标（轴）的名称，<br>使用引号编入<br>- 或 -<br>机床坐标（轴）的索引；值 1..8 |
| <Z0 bank>                           | Z0 银行的指数，值为 1..5                                    |

表格 7-18: 语法 ZOV

**语法示例:**

|          |                               |
|----------|-------------------------------|
| ZOV("X") | 返回机床坐标（轴）X 的所有有效（最新编程的）Z0 的总和 |
| ZOV(2,3) | 返回第二个机床坐标（轴）的第三个 Z0 组的有效 Z0   |

### 7.3.2 位置和轮廓偏置

**AXO**

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

**描述:**

## 轴偏置 - 位置偏置

AXO 为块准备时的坐标提供当前 SETPOS 偏置，即提供程序解释时上次激活的偏置。

语法:

|                                      |                                                           |
|--------------------------------------|-----------------------------------------------------------|
| AXO(<Coordinate>[, <SelectionType>]) |                                                           |
|                                      | （关于参数，请参阅 第 7.1.3 章 “用于读取坐标和轴位置的命令” 第 377 页；无法对 ERRNO 编程） |

表格 7-19: AX0 语法



AX0 只允许访问其自己频道的偏置值。因为异步轴没有任何 SETPOS 偏置，所以不允许 AX0 访问异步轴。

示例:

通道 1 根据示例配置（请参阅 插图 7-2 “轴名称到通道的配置和分配” 第 373 页）：

|                       |                                                        |
|-----------------------|--------------------------------------------------------|
| N10 G1 G90 X100 F1000 |                                                        |
| N20 SETPOS (X75,Y125) |                                                        |
| 30 XD = AXO ("X")     | 当前通道的 x 坐标 (XD = 100-75 = 25) 的最新激活的 SETPOS 偏置被分配给 XD。 |
| 40 YD = AXO (2,0)     | 第二个系统轴 (YD = 200-125 = 75) 的最近激活的 SETPOS 偏置被分配给 YD。    |
| 50 X2D = AXO (4,0)    | 运行时错误, 因为第四个系统轴被分配给通道 2。                               |

**COF**

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

**描述:**

### 轮廓偏置 - 轮廓偏置

提供坐标的最后编程轮廓偏置（移位）。

由于编程的轮廓偏置仅影响当前通道中的坐标，因此选择当前通道中不存在的坐标会导致错误消息。

校正值在当前通道的有效测量单位中提供，即以“英寸”为单位的 G70 和以“毫米”为单位的 G71。对于旋转轴/旋转空间坐标，单位始终为“度数”。

CPL 命令

|     |                                        |                                                              |
|-----|----------------------------------------|--------------------------------------------------------------|
| 语法: | COF (<Coordinate> [, <SelectionType>]) |                                                              |
|     |                                        | (关于参数, 请参阅 第 7.1.3 章 “用于读取坐标和轴位置的命令” 第 377 页 ; 无法对 ERRNO 编程) |

表格 7-20: 语法 COF

|     |                 |                               |
|-----|-----------------|-------------------------------|
| 示例: | 10 A=COF (3)    | 返回激活通道中具有第三个坐标索引的坐标的最后编程轮廓偏置。 |
|     | :               |                               |
|     | 20 B=COF ("X")  | 返回激活通道中 x 坐标的最新编程轮廓偏置。        |
|     | :               |                               |
|     | 30 C=COF (2, 0) | 返回激活通道中第二系统轴的最后一个编程轮廓偏置。      |
|     | :               | 只有当系统轴与 WCS 坐标相同时才允许访问。       |
|     | 100 C=COF (0)   | 运行时错误, 因为 0 是无效的坐标索引。         |
|     | :               |                               |

7.3.3

工件位置校正和位置（斜面）

DPC

| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|
|--------|------|-------|-------|------|------|-------|---|

**描述:**     **参照点修正** – 工件位置修正

返回坐标（偏置值和旋转角度）的工件位置修正（BCR）的最新编程参数。

因为工件位置修正仅对当前通道中的坐标起作用, 所以选择当前通道中不存在的坐标会产生错误消息。

以当前通道激活测量单位中提供修正值。即 G70 为 “英寸”, G71 为 “mm”。

对于旋转轴/旋转空间坐标, 单位始终为“度数”。

|     |                                                              |                                                           |
|-----|--------------------------------------------------------------|-----------------------------------------------------------|
| 语法: | DPC (<Coordinate>[, <Selection type>] )                      |                                                           |
|     | (关于参数, 请参阅 第 7.1.3 章 “用于读取坐标和轴位置的命令” 第 377 页 ; 无法对 ERRNO 编程) |                                                           |
|     | <Coordinate>                                                 | 规范的其他作用:<br>"1...N"... "n" 或 "名称": 返回偏置值<br>"0" 返回旋转角 Phi |

表格 7-21: 语法 DPC

|     |                 |                              |
|-----|-----------------|------------------------------|
| 示例: | 10 A=DPC (1)    | 使用通道中的第一坐标索引返回坐标最后编程的工件位置修正。 |
|     | :               |                              |
|     | 15 B=DPC ("X")  | 返回通道中 x 轴/坐标的最后编程工件位置修正。     |
|     | :               |                              |
|     | 20 B=DPC (2)    | 使用通道中的第二坐标索引返回坐标最后编程的工件位置修正。 |
|     | :               |                              |
|     | 25 B=DPC (2, 0) | 返回激活通道中第二系统轴的最后编程工件位置修正。     |
|     | :               |                              |



30 ANGLE=DPC(0) 返回最后编程的旋转角 Phi。

$$\vdots$$

100 C=DPC(9)      运行时错误，因为如果系统中有 8 个轴，则 9 为无效的坐标索引。

$$\vdots$$

**PMV**

|               |      |       |              |      |      |       |   |
|---------------|------|-------|--------------|------|------|-------|---|
| <b>DOUBLE</b> | READ | WRITE | <b>FUNCT</b> | PROC | WAIT | ERRNO | ! |
|---------------|------|-------|--------------|------|------|-------|---|

**描述:** 定位值 - 激活定位值

返回最后编程的、以及因此有效的坐标定位值总和。可选地对单个定位库有效值进行读取。

语法:

|                                        |                                                                  |
|----------------------------------------|------------------------------------------------------------------|
| PMV (<Coord. Selection>[, <Pl. Bank>]) |                                                                  |
| <Coord. Selection>                     | 坐标 (1...6) 的索引或 WCS 坐标固定标识符 "X"、 "Y"、 "Z"、 "PHI"、 "THE"、 "PSI" 。 |
| <Pl. bank>                             | 定位库 (1...5) 的索引                                                  |

表格 7-22: *PMV* 语法

语法示例:

PMV ("Y") 返回所有“Y”坐标的激活(最后编程的)定位总数。由于可能的旋转, 不仅包含各个库的总和, 还包含基本工件坐标系“y”方向中的结果偏置。

PMV (4, 4) 返回第四库的激活定位欧拉角 PHI。

## PMV

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

**描述:** 定位表 - 读取和写入对定位表的访问权限

IndraMotion MTX 的 XML 定位表的读和写访问权限。只能访问单个元素。写入时，也可以指定增量修改。

语法:

|                                                                               |                                                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PMT (<Coord. Selection>, <Pl. Code>[, [<Pl. Bank>] [, [<Table>] [, <Unit>]]]) |                                                                                                                                                                                                                                  |
| <Coord. Selection>                                                            | <p>坐标 (1...6) 的索引或 WCS 坐标固定名称 "X"、"Y"、"Z"、"PHI"、"THE"、"PSI"。</p> <p>可对值进行读或写。</p> <p><i>写访问时，可能会有以下情形：</i></p> <ul style="list-style-type: none"> <li>● 覆盖表值或</li> <li>● 如果 &lt;Coord. Selection&gt; 以负号开头，则将值添加到表值中。</li> </ul> |
| <Pl. code>                                                                    | <p>154: 定位 1</p> <p>155: 定位 2</p> <p>156: 定位 3</p> <p>157: 定位 4</p> <p>158: 定位 5</p> <p>159: 定位 6</p>                                                                                                                            |
| <Pl. bank>                                                                    | <p>定位库 (1...5) 的索引</p> <p>默认值: 1</p>                                                                                                                                                                                             |

CPL 命令

|         |                                                                                                                           |
|---------|---------------------------------------------------------------------------------------------------------------------------|
| <Table> | 定位表的名称；可根据需要使用绝对或增量路径信息（表扩展名 ".pmt" 不必编程）。如果没有路径信息，则在配置的搜索路径中搜索。<br><br>对具有默认名称（如 PM1、PM2 等）的表而言，仅提供数字值。如果没有表名，则访问最近激活的表。 |
| <Unit>  | 0 或 "MM"：mm<br>1 或 "英寸"：英寸<br><br>对三个平移坐标而言，已分配值以写访问指定的单位解释。如果读取访问，值被转换成指定的单位。<br><br>默认值：毫米                              |

表格 7-23:       PMT 语法

|       |                                                |                                                          |
|-------|------------------------------------------------|----------------------------------------------------------|
| 语法示例: | PMT ("Z", 155)                                 | 对通道前一次激活定位表中的坐标 "Z" 的 G155 偏置进行访问。                       |
|       | PMT (5, 157, 3)                                | 对通道最后激活定位表中的 G157.3 的欧拉角 <b>9</b> 进行访问。                  |
|       | PMT ("X", 154, , "P1")                         | 对定位表 P1 中坐标 "X" 的 G154.1 偏置进行访问。                         |
|       | PMT (3, 158, 4, , 1)                           | 对通道最后激活定位表中的坐标 "Z" 的 G158.4 偏置进行英制访问。                    |
|       | PMT ("Y", 159, 5, "/mnt/esmuser/de/PL5", "MM") | 对目录 "/mnt/esmuser/de" 表 "PL5" 中坐标 "Y" 的 G159.5 偏置进行公制访问。 |

ROT2EUL

|     |                                                                                                                                                                                     |                                                    |       |      |      |       |   |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|-------|------|------|-------|---|
| --- | READ                                                                                                                                                                                | WRITE                                              | FUNCT | PROC | WAIT | ERRNO | ! |
| 描述: | <b>旋转至欧拉角</b> – 從沿着轴的旋转计算欧拉角<br><br>ROT2EUL 是计算欧拉角的 CPL 命令。可指定沿着 x、y 轴和/或 z 轴后续旋转。然后将这些转换为欧拉角进行布局。<br><br>与 ABSROT2EUL（第 4 章 "ABSROT2EUL" 第 397 页）不同，它不是沿空间固定的轴旋转，而是沿着新的（旋转的）坐标系旋转。 |                                                    |       |      |      |       |   |
| 语法: | ROT2EUL(<EulerAngle>, <RotaryAxis1>, <Angle1> [, <RotaryAxis2>, <Angle12> [, <RotaryAxis3>, <Angle3>]])                                                                             |                                                    |       |      |      |       |   |
|     | <Euler angle>                                                                                                                                                                       | 三个 DOUBLE 值的数组<br>输入：欧拉角，起始坐标系旋转<br>输出：欧拉角，结果坐标系旋转 |       |      |      |       |   |
|     | <Rotary axis1..3>                                                                                                                                                                   | 要旋转的轴（"x"，"y" 或 "z"）                               |       |      |      |       |   |
|     | <Angle1..3>                                                                                                                                                                         | 要旋转的角度（以度为单位）                                      |       |      |      |       |   |

表格 7-24:       语法 ROT2EUL

|     |                                                                                                                                                                                                                                                             |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 示例: | <b>程序</b><br><br>1 REM Compute Euler angle for rotation along X (20°) and then along Z (60°)<br>2 DIM EULER!(3)<br>3 EULER!(1) = 0 : EULER!(2) = 0 : EULER!(3) = 0<br>4 ROT2EUL(EULER!, "X", 20, "Z", 60)<br>5 REM Result in field EULER! (Phi, Theta, Psi) |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

ABSROT2EUL

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** **绝对旋转至欧拉角** – 從沿着空间固定的轴旋转计算欧拉角

ABSROT2EUL 是计算欧拉角 的 CPL 命令。可以指定沿 BCS（空间固定）的 x、y 和/或 z 轴旋转三圈。然后将这些转换为欧拉角进行布局。

与 ROT2EUL（第 章“ROT2EUL”第 396 页）相比，它不是沿着空间固定的轴旋转，而是沿着基本坐标系的轴旋转。

|                   |                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------|
| <b>语法:</b>        | ABSROT2EUL (<EulerAngle>, <RotaryAxis1>, <Angle1> [, <RotaryAxis2>, <Angle12> [, <RotaryAxis3>, <Angle3>]]) |
| <Euler angle>     | 三个 DOUBLE 值的数组<br>输入: 欧拉角, 起始坐标系旋转<br>输出: 欧拉角, 结果坐标系旋转                                                      |
| <Rotary axis1..3> | 要旋转的轴 (“x”, “y” 或 “z”)                                                                                      |
| <Angle1..3>       | 要旋转的角度 (以度为单位)                                                                                              |

表格 7-25: ABSROT2EUL 语法

**示例:** 程序

```
1 REM Compute Euler angle for rotation along Y (45°) of an already rotated KOS
2 DIM EULER!(3)
3 EULER!(1) = 90 : EULER!(2) = -30 : EULER!(3) = 270
4 ABSROT2EUL(EULER!, "Y", 45)
5 REM Result in field EULER! (Phi, Theta, Psi)
```

PT2EUL

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** **指向欧拉角** – 计算指定三点的倾斜平面的欧拉角

PT2EUL 是计算欧拉角 的 CPL 命令。将位于一个平面上的三个点（如 x、y 和 z 值）传输到该命令。欧拉角是以这种平面跨越新的 xy 平面的方式从点计算出来的。因此，可以对具有三个点的区数组进行采样，并且可以通过 PT2EUL 为相应的布局确定欧拉角。

所有这三点必须有所不同，并且不在一条直线上。即使参数足以确定一个全新的平面，只有计算出的欧拉角被返回。由於 Pt1 等於零點，所以該平面可以直接設置角度和 Pt1。

|               |                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------|
| <b>语法:</b>    | PT2EUL (<EulerAngle>, <Pt1X>, <Pt1Y> ,<Pt1Z>, <Pt2X>, <Pt2Y> ,<Pt2Z>, <Pt3X>, <Pt3Y> ,<Pt3Z>) |
| <Euler angle> | 三个 DOUBLE 值的数组<br>输入: 欧拉角, 起始坐标系旋转<br>输出: 欧拉角, 结果坐标系旋转                                        |
| <Pt1XYZ>      | 新平面的零點與起始坐標係有關                                                                                |
| <Pt2XYZ>      | 通过与起点坐标系相关的 Pt2 的新 x 轴                                                                        |
| <Pt3XYZ>      | Pt3 (与起始坐标系有关) 位于新的 xy 平面中                                                                    |

表格 7-26: 语法 PT2EUL

**示例:** 程序

```
1 REM The probed points are located from PT1! to PT3!
2 DIM EULER!(3)
3 EULER!(1) = 0 : EULER!(2) = 0 : EULER!(3) = 0
```

CPL 命令

```
4 PT2EUL(EULER!, PT1!(1), PT1!(2), PT1!(3), PT2!(1), PT2!(2), PT2!(3), PT3!(1), PT3!(2), PT3!(3))
5 REM Result in field EULER! (Phi, Theta, Psi)
; The placement can be directly set with G152
G152.1([PT1!(1)], [PT1!(2)], [PT1!(3)], [EULER!(1)], [EULER!(2)], [EULER!(3)])
```

VEC2EUL

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** **矢量到欧拉角** - 计算法向量和新 x 轴方向的斜面的欧拉角

VEC2EUL 是计算欧拉角 的 CPL 命令。使用法向矢量（垂直于平面上）指定平面。此外，可以为 x 轴选择编程方向。该命令计算这个平面的欧拉角。

如果第二个矢量不垂直于法向矢量，则使用其垂直投影来确定 x 轴。这两个向量可能不是平行的。如果 x 轴的矢量未编程，则 y 轴与旧 xy 平面平行指定。所包围的角度对应于第二欧拉角（如 G151）。

|               |                                                                                   |
|---------------|-----------------------------------------------------------------------------------|
| <b>语法:</b>    | VEC2EUL(<EulerAngle>, <NormX>, <NormY> ,<NormZ> [, <XAxisX>, <XAxisY> ,<XAxisZ>]) |
| <Euler angle> | 三个 DOUBLE 值的数组<br>输入: 欧拉角, 起始坐标系旋转<br>输出: 欧拉角, 结果坐标系旋转                            |
| <NormXYZ>     | 新平面的法向矢量与起始坐标系有关                                                                  |
| <XAxisXYZ>    | 与起始坐标系相关的新 x 轴的方向矢量                                                               |

表格 7-27: 语法 VEC2EUL

**示例:** *程序*

```
1 REM Compute Euler angle with normal vector
2 DIM EULER! (3)
3 EULER!(1) = 0 : EULER!(2) = 0 : EULER!(3) = 0
4 VEC2EUL(EULER!, 1, 0.5, 0)
5 REM Result in field EULER! (Phi, Theta, Psi)
```

WCS2BCS

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** **WCS 到 BCS** - 将一个点从工件坐标系（WCS）转换到基准坐标系（BCS）

WCS2BCS 是一个转换点的 CPL 命令。工件坐标系（WCS）中的转换点被转换为基准坐标系（BCS）。两个点之间的连接位置不同。

展示位置数据可以选择传输到命令中。如果此数据未编程，则当前有效的布局（无 BCR）用于计算。

|             |                                                     |
|-------------|-----------------------------------------------------|
| <b>语法:</b>  | WCS2BCS(<Point> [, <Placement>])                    |
| <Point>     | 三个 DOUBLE 值的数组<br>输入: 指向 WCS<br>输出: 指向 BCS          |
| <Placement> | 包含六个 DOUBLE 值的数组<br>定位数据 [X, Y, Z, Phi, Theta, Psi] |

表格 7-28: WCS2BCS 语法

**示例:** *程序*

```
; Set a placement
G151.1(10,20,30, 45,30,0)
WAIT
; The zero point of a new placement is to be located on the current placement related to (10,0,50)
```

|     |      |       |       |             |      |       |   |
|-----|------|-------|-------|-------------|------|-------|---|
| --- | READ | WRITE | FUNCT | <b>PROC</b> | WAIT | ERRNO | ! |
|-----|------|-------|-------|-------------|------|-------|---|

|             |                                                     |                                  |
|-------------|-----------------------------------------------------|----------------------------------|
| 语法:         |                                                     | BCS2WCS(<Point> [, <Placement>]) |
| <Point>     | 三个 DOUBLE 值的数组<br>输入: 指向 BCS<br>输出: 指向 WCS          |                                  |
| <Placement> | 包含六个 DOUBLE 值的数组<br>定位数据 [X, Y, Z, Phi, Theta, Psi] |                                  |

示例： 程序

```
; Set a placement
G151.1(10,20,30, 45,30,0)
WAIT
; previously measured point in the field PT! is present in the BCS and is now to be approached
2 BCS2WCS(PT!)
; Result in the field PT!
; Approach point in new coordinate system
G1 F500 X[PT!(1)] Y[PT!(2)] Z[PT!(3)]
```

**SCL**

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

|             |                                                   |              |
|-------------|---------------------------------------------------|--------------|
| 语法:         | SCL(<Selection>[,<Coordinate>[,<SelectionType>]]) |              |
| <Selection> | 0:                                                | 主平面最后编程的旋转角  |
|             | 1:                                                | 通道轴最后编程的极点   |
|             | 2:                                                | 通道轴最后编程的缩放系数 |

CPL 命令

|                  |                                                                                                                                                    |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <Coordinate>     | 坐标的索引或名称：<br><b>名称</b> 被解释为坐标名称。只有不存在对应的坐标名称时才将该名称解释为通道或系统轴名称。<br><b>索引</b> 根据指定的 <SelectionType> 进行解释。<br>如果未对可选变量 ERRNO 编程，则对未配置的坐标/轴编程会导致运行时错误。 |
| <Selection type> | <b>可选：</b><br>确定如何解释在 <Coordinate> 下编程的索引：<br>0：系统轴索引<br>1：坐标索引（默认）<br>如果没有 <Column selection>，则索引被解释为坐标索引！                                        |

表格 7-30: SCL 语法

|     |    |              |                            |
|-----|----|--------------|----------------------------|
| 示例： | 10 | W=SCL(0)     | 将最后编程的旋转角写入变量 W。           |
|     | :  |              |                            |
|     | 20 | P=SCL(1,2)   | 将通道中具有第二坐标索引的坐标极点写入变量 P。   |
|     | :  |              |                            |
|     | 30 |              | 将通道中具有第二坐标索引的坐标缩放系数写入变量 F。 |
|     |    | F=SCL(2,2,1) |                            |
|     | :  |              |                            |
|     | 40 |              | 将激活通道中 x 坐标的缩放系数写入变量 D。    |
|     |    | D=SCL(2,"X") |                            |
|     |    |              |                            |

7.3.5

刀具修正

一般信息

以下 CPL 命令允许对内部刀具修正数据（D 修正）和外部刀具修正数据（ED 修正）进行访问，PLC 可对这些数据进行访问。

TCV

| DOUBLE | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|
|--------|------|-------|-------|------|------|-------|---|

**描述：** ToolCorrValues - 应用的最新刀具校正值  
提供最后编程的刀具修正值，该值可以是总和（D 修正 + 外部修正内存）或单值。

语法:

|                                                 |                                                                                                                                                                                                                                          |
|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TCV (<ValueSelection>[, <CorrectionSelection>]) |                                                                                                                                                                                                                                          |
| <Value selection>                               | 0: 最后激活的 D/ED 修正<br>1 或 "L1": L1 修正长度<br>2 或 "L2": L2 修正长度<br>3 或 "L3": L3 修正长度<br>4 或 "RAD": 刀具/刀具刃口半径<br>5 或 "ORI": 刃口位置<br>6 或 "PHI": 欧拉角 $\phi$ (仅限 ED 修正)<br>7 或 "THE": 欧拉角 $\theta$ (仅限 ED 修正)<br>8 或 "PSI": 欧拉角 $\psi$ (仅限 ED 修正) |
| <Correction selection>                          | 1 或 "D": D 修正 (内部)<br>2 或 "E": ED 修正 (外部)<br>默认: D 和 ED 修正的总和                                                                                                                                                                            |

表格 7-31: TCV 语法

语法示例:

|              |                            |
|--------------|----------------------------|
| TCV (0, "D") | 提供最新的编程 D 校正。              |
| TCV ("L1")   | 读取最后编程的 D 和 ED 修正 L1 值的总和。 |
| TCV (4, 1)   | 读取最后编程的 D 修正的刀具半径          |
| TCV (2, "E") | 读取最后编程的 ED 修正的 L2 刀具长度。    |

DCT

描述:

**D 校正表** – 访问 D 校正表和 ED 校正寄存器  
对任何 D 修正表中的刀具修正值以及外部修正值 (ED 修正) 进行读和写访问。写入时, 也可以指定增量修改。



DCT 命令直接写入指定的表或校正寄存器。如果这是一个激活表格, 写入的校正值只有在下一个数控程序段时才会生效。

**使用功能 “异步子程序” (ASUP), “离开和返回轮廓”, “删除待移动距离” 和 “覆盖”:**

所有已准备好的数控程序段都从这些功能的中断点重新准备好。如果使用 DCT 更改已经准备好的数控程序段范围内的修正值, 则更改的值现在适用于要准备的数控程序段。

此外, 已更改的校正值在被调用的异步子程序内或在离开返回轮廓的逼近运动中已处于激活状态。

**如果不需要这种行为, 请在每个 DCT 之前编写 “WAIT” 命令!**

可选地, 可以使用 /SysCorrRefrDis 系统日期来全局地抑制表和校正寄存器的刷新机制。在这种情况下, 校正值更改只有在下一次校正或下一个表格激活或边缘选择时才会激活。

CPL 命令

|                   |                                                                                                                                                                                                                                                                                                                                                                                        |  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 语法:               | DCT(<ValueSelection>,<DataBlock>[, [<Table>]],[<Unit>]])                                                                                                                                                                                                                                                                                                                               |  |
| <Value selection> | 1 或 "L1": L1 修正长度<br>2 或 "L2": L2 修正长度<br>3 或 "L3": L3 修正长度<br>4 或 "RAD": 刀具/刀具刀口半径<br>5 或 "ORI": 刀口位置<br>6 或 "PHI": 欧拉角 $\phi$ (仅限 ED 修正)<br>7 或 "THE": 欧拉角 $\theta$ (仅限 ED 修正)<br>8 或 "PSI": 欧拉角 $\psi$ (仅限 ED 修正)<br>可对值进行读或写。<br>写访问时,可能有: <ul style="list-style-type: none"><li>覆盖表值或</li><li>如果 &lt;Value selection&gt; 以负号开头,则将值添加到表值中。</li></ul> 不可能对刀口位置 ( 5 或 "ORI") 进行增量写入。 |  |
| <Data set>        | 1...99: 针对 D 修正<br>1...16: ED 修正                                                                                                                                                                                                                                                                                                                                                       |  |
| <Table>           | D 修正表的名称,<br>可根据需要使用绝对或增量路径信息 (无表扩展名 ".dct")。<br>如果没有路径信息,则在配置的搜索路径中搜索。<br>对具有默认名称 (如 DC1、DC2 等) 的表而言,仅提供数值。<br>没有最近访问的 D 校正表被使用。<br>0: 访问当前通道的 ED 修正。<br>-1: 访问通道 1 的 ED 修正。<br>-2: 访问通道 2 的 ED 修正。<br>-3: 访问通道 3 的 ED 修正。<br>...<br>-60: 访问通道 60 的 ED 修正。                                                                                                                            |  |
| <Unit>            | 仅与 D 校正表相关:<br>0 或 "MM": mm (默认)<br>1 或 "英寸": 英寸<br>在写访问期间,分配的值将以指定的单位进行解释。在读访问期间,该值被转换为指定的单位。<br>进行 ED 校正时,该值始终按照本机 (在主叫通道 G70/G71 中选择的最后一个单元) 进行解释或在本机中提供。                                                                                                                                                                                                                           |  |

表格 7-32: DCT 语法

|       |                 |                                 |
|-------|-----------------|---------------------------------|
| 语法示例: | DCT (1,10,"K4") | 访问 D 修正表 "K4"中数据块 10 的 "L1"修正值。 |
|       | DCT ("RAD",7)   | 对前一次激活的表数据块 7 的刀具半径进行增量写入。      |
|       | DCT (3,1,0)     | 访问通道外部刀具校正的 "L3" 校正。            |



DCT("L2",16,1,"INCH") 对表 DC1.dct 的数据块 16 的 "L2" 修正进行英制访问。

DCT(4,7,-3) 访问第三通道的修正设置 7 的刀具半径。

-  当从第三方通道读 / 写刀具修正数据时，请注意，不存在不同通道之间的竞争访问。
-  进行 ED 校正时，该值始终按照本机（在主叫通道 G70/G71 中选择的最后一个单元）进行解释或在本机中提供。

7.4 控制数据和过程

7.4.1 控制版本

VERSINF\$

| CHAR | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

描述:

VERSION 信息 - 控制版本

VERSINF\$ 允许在 CPL 程序中查询 IndraMotion MTX 的管理数据。该命令提供了一个“String”类型的值。

语法:

|                                |                                                                                                                                                                                  |             |          |            |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|----------|------------|
| VERSINF\$(<Index1>[,<Index2>]) |                                                                                                                                                                                  |             |          |            |
| <Index1>                       | 整数类型的常量<br>命令模式:<br>1: 读取软件版本<br>2: 读取硬件版本                                                                                                                                       |             |          |            |
| <Index2>                       | 整数型。附加命令模式取决于参数值 <Index1>。<br><Index1> = 1: 无其它模式可用<br><Index1> = 2: 附加模式:<br><table><tr><td>1: 硬件类型（默认）</td></tr><tr><td>2: 矢量组编号</td></tr><tr><td>3: 矢量组编号索引</td></tr></table> | 1: 硬件类型（默认） | 2: 矢量组编号 | 3: 矢量组编号索引 |
| 1: 硬件类型（默认）                    |                                                                                                                                                                                  |             |          |            |
| 2: 矢量组编号                       |                                                                                                                                                                                  |             |          |            |
| 3: 矢量组编号索引                     |                                                                                                                                                                                  |             |          |            |

示例:

10 DIM SYS\_INFO\$(50)  
20 SYS\_INFO\$ = VERSINF\$(1)  
30 PRN#(0,"软件版本: ", SYS\_INFO\$)  
M30

表格 7-33:      VERSINF\$ 语法

7.4.2 时间记录

时钟

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

描述

时钟 - 系统时间

读取控制器内部的 系统时间或内部插补周期计数器。两个测量结果的差异可用于确定程序段的运行时间。

CPL 命令

|     |               |                                                                                                               |
|-----|---------------|---------------------------------------------------------------------------------------------------------------|
| 语法: | CLOCK(<Mode>) |                                                                                                               |
|     | <Mode>        | 可选（默认：0）<br><br>0:<br>系统时间输出（以毫秒为单位）。<br><br>1:<br>系统时间输出（以微秒为单位）。<br><br>2:<br>以毫秒为单位指示插补周期计数器的当前状态，并设置插补周期。 |

表格 7-34:      *CLOCK* 语法

示例:

```
:
20 WAIT
30 STARTTIME% = CLOCK(2)
N4 G1 X50 Y70
40 WAIT
50 ENDTIME%=CLOCK(2) : DIFF%=ENDTIME%-STARTTIME%
:
```

执行块 N4 前后，插补周期计数器的当前状态已分配给变量“STARTZEIT%”或“ENDZIET%”。可众这两个变量之间的差值确定 N4 的块执行时间（毫秒）。

请注意“WAIT”命令是绝对必需的！



具有控制内部系统时间的程序段的运行时间测量不会在 MTX 控制仿真中提供有用的结果！

测量 Ipo 周期的数量导致 MTX 控制仿真和实际控制几乎相同的运行时间，只要不存在无准备数控块的延迟（例如，通过块循环时间问题或长 CPL 段落。

DATE

| CHAR | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

描述:

**DATE** – 日期

提供有关当前日期的数据。

|     |                |                                                                                                                                                                    |
|-----|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | DATE[(<Mode>)] |                                                                                                                                                                    |
|     | <Mode>         | 可选（默认：0）<br><br>0:<br>以格式输出 “DD. MM”。<br><br>1:<br>以格式输出 “DD. MM. YY”。<br><br>2:<br>星期的输出：1 =周一，2 =周二等。<br><br>3:<br>一年中的当天 “40 日” 为 2 月 9 日。<br><br>4:<br>一年的日历周。 |

表格 7-35:      *DATE* 语法



如果 DATE 在程序中被使用若干次，则必须确定对应结果变量维度。否则，DATE 命令最后读取值被赋予所有未确定维度的变量，该变量包含 DATE 分配结果。

示例：

```
:
10 DIM A$(5) 字符串变量 A$ 的维度。
20 DIM B$(8) 字符串变量 B$ 的维度。
30 DIM C$(1) 字符串变量 C$ 的维度。
40 DIM D$(3) 字符串变量 D$ 的维度。
50 DIM E$(2) 字符串变量 E$ 的维度。
60 A$ = DATE 日期以 "DD.MM" 形式分配给字符串变量 A$。
70 B$ = 日期以 "DD.MM.YY" 形式分配给字符串变量 B$。
DATE(1)
80 C$ = 平日数字值 "1" - "7" 分配给字符串变量 C$。
DATE(2)
90 D$ = 平日数字值 "1" - "366" 分配给字符串变量 D$。
DATE(3)
100 E$ = 一年的日历周数字值 "1" - "53" 分配给字符串变量 E$
DATE(4)
:
```

时间

| CHAR | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

描述：

**TIME** - 时间  
提供时间的当前值。

示例：

```
:
40 B$ = TIME 日期以 "HH.MM.SS" 形式分配给字符串变量 B$。
:
```



如果 TIME 在程序中被使用若干次，则必须确定对应结果变量维度。否则，TIME 命令最后读取值被赋予所有未确定维度的变量，该变量包含 TIME 分配结果。

7. 4. 3

机床参数

MCA

| MULTYP | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|
|--------|------|-------|-------|------|------|-------|---|

描述：

**MAchine COnfiguration DAta** - 机床参数  
传输单个 MACODA 参数的内容。根据数据的类型，该值可以是以下类型 "INT"、"REAL"、"DOUBLE" 或 "CHAR"。要存储转移值的变量的类型必须相同！  
程序运行期间检测转移值和目标变量之间的类型冲突，并以错误消息的形式确认。

CPL 命令

|     |                                                   |                                                                                                                                         |
|-----|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | MCA (<BlockId>, <ParameterIndex>[, <ClassIndex>]) |                                                                                                                                         |
|     | <BlockId>                                         | MACODA 参数的数量<br>一个 MACODA 可能包含多个单个 MACODA 参数（参数列表）。<br>如果编程不存在的参数编号，则会输出运行时错误。                                                          |
|     | <Parameter index>                                 | 单一 MACODA 参数的索引，以 "0" 开头。<br>如果编写了不存在的索引号，则会输出运行时错误。                                                                                    |
|     | <Class index>                                     | 可选：仅适用于通道，主轴和变换特定的机器参数。<br>该参数指定必须读取机床参数值的通道，主轴或变换索引。<br>如果没有编程，并且在通道参数的情况下，该命令提供当前执行 CPL 程序的通道的 MACODA 单一参数。<br>如果編寫了無效的類索引，則會輸出運行時錯誤。 |

表格 7-36: 语法 MCA

示例: **MCA 命令**

```
10 BLOCKNO%=100300004
20 ERG%=MCA (BLOCKNO%, 0)
```

机床参数 100300004 的第一个单参数的内容被分配给整数变量 ERG%。

7. 4. 4 简单系统数据  
SD（一般描述）

| INT                                                                            | READ                                             | WRITE          | FUNCT | PROC | WAIT | ERRNO | ! |
|--------------------------------------------------------------------------------|--------------------------------------------------|----------------|-------|------|------|-------|---|
| 描述: <b>系统数据</b> – 整数格式的简单系统内部数据<br>从 <b>整数格式</b> 的 IndraMotion MTX 系统内部数据集中读取。 |                                                  |                |       |      |      |       |   |
| 语法:                                                                            | SD (<Group>[, <Index1>[, <Index2>[, <Index3>]]]) |                |       |      |      |       |   |
|                                                                                |                                                  | SD 命令提供 整数型 值。 |       |      |      |       |   |

表格 7-37: SD 语法

SD (2, ...) – 当前倍率位置

| 语法                                   | 返回值                               |
|--------------------------------------|-----------------------------------|
| SD (2, 1)                            | 提供当前通道中进给倍率的当前位置。                 |
| SD (2, 2)                            | 提供当前通道中快速进给倍率的当前位置。               |
| SD (2, 3)<br>( $\Delta$ SD (202, 1)) | 返回主轴 1 的倍率当前位置。<br>如果没有应用主轴，则为 0。 |
| SD (2, 4)<br>( $\Delta$ SD (202, 1)) | 提供主轴 2 的倍率当前位置。<br>如果没有应用主轴，则为 0。 |

表格 7-38: SD (2, ...)

SD (5, ...) – 速度和速度

| 语法           | 返回值                                    |
|--------------|----------------------------------------|
|              | 激活值:                                   |
| SD (5, 1, 1) | 返回以毫米/分钟或英寸/分钟四舍五入为整数值的激活进给速度（100% 值）。 |

| 语法                                           | 返回值                                                     |
|----------------------------------------------|---------------------------------------------------------|
| SD(5, 2, 1)                                  | 返回以毫米/分钟或英寸/分钟四舍五入为整数值的激活快速进给速度（100% 值）。                |
| SD(5, 3, 1)<br>( $\triangleq$ SD(205, 1, 2)) | 以 rpm 为单位返回第一个主轴的当前主轴速度（命令值）（包括倍率评估）。<br>如果没有应用主轴，则为 0。 |
| SD(5, 4, 1)<br>( $\triangleq$ SD(205, 1, 2)) | 以 rpm 为单位返回第二个主轴的当前主轴速度（命令值）（包括倍率评估）。<br>如果没有应用主轴，则为 0。 |
|                                              | 已编程值：                                                   |
| SD(5, 1, 2)                                  | 以每分钟输入单位返回最后编程的输入。                                      |
| SD(5, 3, 2)<br>( $\triangleq$ SD(205, 2, 1)) | 返回第一个主轴通道中编程的最后一个主轴转速（指令值），单位 rpm。<br>如果没有应用主轴，则返回 0。   |
| SD(5, 4, 2)<br>( $\triangleq$ SD(205, 2, 2)) | 返回第二个主轴通道中编程的最后一个主轴速度（命令值），单位 rpm。<br>如果没有应用主轴，则返回 0。   |
|                                              | 实际值：                                                    |
| SD(5, 3, 3)<br>( $\triangleq$ SD(205, 3, 1)) | 返回第一个主轴的实际主轴速度。                                         |
| SD(5, 4, 3)<br>( $\triangleq$ SD(205, 3, 2)) | 返回第二个主轴的实际主轴速度。                                         |

表格 7-39: SD(5, ...)

## SD(8) – 通道查询

| 语法    | 返回值         |
|-------|-------------|
| SD(8) | 返回调用通道的通道号。 |

表格 7-40: SD(8)

## SD(9) – 接触探头

| 语法    | 返回值                                                                                                                                                                                                                                                                                       |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 返回 G75、G75.2 或 MOB 之后是否触发了接触探头，或者“在固定停止点测量” FSP 或 FsProbe 是否激活的信息：                                                                                                                                                                                                                        |
| SD(9) | <ul style="list-style-type: none"><li>0:<ul style="list-style-type: none"><li>至 G75、G75.2 或 MOB：接触探头切换。</li><li>至 FSP (FsProbe)：已执行固定挡块的测量：</li></ul></li><li>1:<ul style="list-style-type: none"><li>至 G75、G75.2 或 MOB：接触探头未切换。</li><li>至 FSP (FsProbe)：仍未执行固定挡块的测量：</li></ul></li></ul> |

表格 7-41: SD(9)

CPL 命令

SD(10,...)- 应用 L1、L2 或 L3 的坐标索引

| 语法                                            | 返回值                                                                                                                                                                                   |                                                                   |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
|                                               | 如果校正在工件坐标系中已计算，则 SD（10，...）提供应用 L1、L2 或 L3 修正轴的通道轴索引：<br>如果在工件坐标系 TCS，SD(10,...）中计算修正值，则返回值如下：<br>101: TCS ≙ XTR 在 x 方向的计算。<br>102: TCS ≙ YTR 在 y 方向的计算。<br>103: TCS ≙ ZTR 在 z 方向的计算。 |                                                                   |
| SD(10,1,<Corr>)<br>SD(10,1)<br>(≙ SD(10,1,3)) | 在块准备的时间点。                                                                                                                                                                             |                                                                   |
| SD(10,2,<Corr>)<br>SD(10,2)<br>(≙ SD(10,2,3)) | 在激活的时间点。                                                                                                                                                                              |                                                                   |
|                                               | <Corr>                                                                                                                                                                                | 1: 返回适用 L1 修正的坐标索引。<br>2: 返回适用 L2 修正的坐标索引。<br>3: 返回适用 L3 修正的坐标索引。 |

表格 7-42:SD(10,...)

SD(11,...)- 主轴，辅助轴和进给轴的通道轴索引

| 语法              | 返回值                       |                             |
|-----------------|---------------------------|-----------------------------|
|                 | 根据激活平面提供主轴，辅助轴和进给轴的通道轴索引： |                             |
| SD(11,<Axis>,1) | 在块准备的时间点。                 |                             |
| SD(11,<Axis>,2) | 在激活的时间点（块执行）。             |                             |
|                 | <Axis>                    | 1: 主轴<br>2: 次级轴<br>3: 横向进给轴 |

表格 7-43:SD(11,...)

## SD (12, ...) – 主轴旋转的方向

| 语法                                         | 返回值                                                                                                                               |
|--------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| SD(12, 1)<br>( $\triangleq$ SD(212, 1, 1)) | 主轴旋转激活方向:<br>0: 主轴停止<br>3: 主轴正转<br>4: 主轴逆时针<br>19: 定向主轴<br>-1: 主轴未应用。                                                             |
| SD(12, 2)<br>( $\triangleq$ SD(212, 2, 1)) | 最后编程的主轴旋转方向:<br><ul style="list-style-type: none"> <li>如主轴旋转的“激活方向”命令 (SD (12, 1))。</li> <li>不考虑每个接口信号的激活改变旋转方向!</li> </ul>       |
| SD(12, 3)<br>( $\triangleq$ SD(212, 1, 2)) | 激活的主轴旋转方向 (第二主轴):<br><ul style="list-style-type: none"> <li>“激活的主轴旋转方向”等命令 (SD(12, 1))。</li> </ul>                                |
| SD(12, 4)<br>( $\triangleq$ SD(212, 2, 2)) | 最后编程的主轴旋转方向 (第二主轴):<br><ul style="list-style-type: none"> <li>如主轴旋转的“激活方向”函数 (SD(12, 1))。</li> <li>不考虑每个接口信号的激活改变旋转方向!</li> </ul> |

表格 7-44: SD(12, ...)

## SD (13) – 解释时的处理方法

| 语法     | 返回值                                                                            |
|--------|--------------------------------------------------------------------------------|
| SD(13) | 在解释时处理时的处理方法:<br>0: 单块, 单步<br>1: 继续块<br>2: 程序块<br>11: 用连续块阻止预运行<br>(仍未对所选块解释)。 |

表格 7-45: SD(13)

## SD (14) – 国家语言

| 语法     | 返回值                                   |
|--------|---------------------------------------|
| SD(14) | 返回激活国家语言的编号 (国家代码) (机床参数 6010 00010)。 |

表格 7-46: SD(14)

## SD (15) – 无运动测试

| 语法     | 返回值                                                                                            |
|--------|------------------------------------------------------------------------------------------------|
| SD(15) | 返回“无运动测试”的信息处于激活状态:<br><ul style="list-style-type: none"> <li>0 = 没有</li> <li>1 = 是</li> </ul> |

表格 7-47: SD(15)

CPL 命令

SD(17,...)- 具有指定含义

| 语法               | 返回值                                  |                                        |
|------------------|--------------------------------------|----------------------------------------|
| SD(17,<Class>,1) | 返回块准备时含义 <Class> 的坐标的通道相关索引（1..8）。   |                                        |
| SD(17,<Class>,2) | 返回激活时间点（块执行）中含义为 <Class> 的坐标的通道相关索引。 |                                        |
|                  | <Class>                              | 坐标描述：<br>1 = "X"<br>2 = "Y"<br>3 = "Z" |

表格 7-48:SD(17,...)

SD(20,...)- 呼叫通道中的轴数

| 语法                      | 返回值            |
|-------------------------|----------------|
|                         | 返回调用通道中同步轴的数量： |
| SD(20)<br>或<br>SD(20,1) | 在块准备的时间点。      |
| SD(20,2)                | 在激活的时间点（块执行）。  |

表格 7-49:SD(20,...)

SD(21,...)- 通道中的轴数

| 语法                                    | 返回值           |              |
|---------------------------------------|---------------|--------------|
|                                       | 返回通道的同步轴数量：   |              |
| SD(21,<Chan>)<br>或<br>SD(21,<Chan>,1) | 在块准备的时间点。     |              |
| SD(21,<Chan>,2)                       | 在激活的时间点（块执行）。 |              |
|                                       | <Chan>        | 1..60 = 通道索引 |

表格 7-50:SD(21,...)



如果指定的通道未激活，则此通道的轴可能已被分配（它们当前在另一通道中处于激活状态）。不过，轴属于指定的通道。

示例：轴 X2 属于通道 2（未激活），X2 当前在通道 1 中同步运行。在指令 SD（21,2，...）中，轴 X2 被视为通道 2 的一部分。



SD (22, ...) – 呼叫信道的系统轴的通道轴号

| 语法                                               | 返回值                   |                                    |
|--------------------------------------------------|-----------------------|------------------------------------|
|                                                  | 它提供调用通道的系统轴的通道轴号或 -1: |                                    |
| SD (22, <SysAxis>)<br>或<br>SD (22, <SysAxis>, 1) | 在块准备的时间点。             |                                    |
| SD (22, <SysAxis>, 2)                            | 在激活的时间点（块执行）。         |                                    |
|                                                  | <SysAxis>             | 1..250 =系统轴索引<br>或<br>系统轴名称以引号括起来。 |

表格 7-51: SD (22, ...)

SD (23, ...) – 调用通道的通道轴的系统轴号

| 语法                                                 | 返回值                 |                                  |
|----------------------------------------------------|---------------------|----------------------------------|
|                                                    | 返回调用通道的通道轴的系统轴号或-1: |                                  |
| SD (23, <ChanAxis>)<br>或<br>SD (23, <ChanAxis>, 1) | 在块准备的时间点。           |                                  |
| SD (23, <ChanAxis>, 2)                             | 在激活的时间点（块执行）。       |                                  |
|                                                    | <ChanAxis>          | 1..8 =通道轴索引<br>或<br>通道轴名称以引号括起来。 |

表格 7-52: SD (23, ...)

SD (24, ...) – 通道轴的系统轴号

| 语法                                                                 | 返回值             |                                  |
|--------------------------------------------------------------------|-----------------|----------------------------------|
|                                                                    | 返回通道轴的系统轴号或 -1: |                                  |
| SD (24, <ChanAxis>, <Chan>)<br>或<br>SD (24, <ChanAxis>, <Chan>, 1) | 在块准备的时间点。       |                                  |
| SD (24, <ChanAxis>, <Chan>, 2)                                     | 在激活的时间点（块执行）。   |                                  |
|                                                                    | <ChanAxis>      | 1..8 =通道轴索引<br>或<br>通道轴名称以引号括起来。 |
|                                                                    | <Chan>          | 1..60: 通道索引                      |

表格 7-53: SD (24, ...)



如果指定的通道未激活，则此通道的轴可能已被分配（它们当前在另一通道中处于激活状态）。不过，轴属于指定的通道。

示例：轴 X2 属于通道 2（未激活），X2 当前在通道 1 中同步运行。在 SD (24, 2, ...) 指令中，轴 X2 被视为属于通道 2。

SD (25, ...) – 系统轴的通道

| 语法                 | 返回值                            |                                    |
|--------------------|--------------------------------|------------------------------------|
| SD (25, <SysAxis>) | 在激活时间点返回系统轴的通道。对于异步轴，返回 “-1” 。 |                                    |
|                    | <SysAxis>                      | 1..250 =系统轴索引<br>或<br>系统轴名称以引号括起来。 |

表格 7-54: SD (25, ...)

SD (30) – 初始化程序调用的触发事件

| 语法      | 返回值                                                                                                                                                                                    |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SD (30) | 返回触发事件以调用相应通道中特定于通道的 init 程序： <ul style="list-style-type: none"><li>1: 主程序结束后 (M30)</li><li>2: 通道复位后</li><li>3: 系统复位后</li><li>0: 不被称为 init 程序，而被称为数控程序</li></ul> SD (30) 旨在用于 init 程序。 |

表格 7-55: SD (30)

SD (60, ...) – 样条数据

| 语法                            | 返回值                                                                                                            |                                                                                                            |
|-------------------------------|----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| SD (60, 1, <Chan>, <BlkType>) | 用 SplineDef (<Spline ID>) 编程的样条线 ID。<br>另请参阅 SDR (60)<br>值 >= 0: 无错误<br>值 = -1: BlkTyp 错误<br>值 = -2: 数控程序段无法确定 |                                                                                                            |
|                               | <Chan>                                                                                                         | 可选:<br>通道索引<br>或<br>-1 为自己的频道<br>默认: -1                                                                    |
|                               | <BlkType>                                                                                                      | 可选:<br>数控块选择:<br>1: 块准备<br>2: 阻止执行<br>-1: 重新进入块<br>-2: 中断块<br>默认:<br>1 为 Chan = 自己的频道<br>2 代表 Chan = 第三方渠道 |

表格 7-56: SD (60, ...)



BlkType = 1 不得用于第三方频道。返回值是 -1  
BlkType = -1、-2 仅返回“取消要走的距离”，“块预运行”和“Asup”的合理返回值。否则，-2 结果。

SD (68, ...) – 程序坐标偏置总和

| 语法                       | 返回值                                 |             |
|--------------------------|-------------------------------------|-------------|
|                          | 为一个通道坐标提供程序坐标偏置的总数 (Trans + ATrans) |             |
| SD (68, 1, <CoordIndex>) | 在块准备时间点 (最新的编程值)。                   |             |
| SD (68, 2, <CoordIndex>) | 在激活的时间点 (块执行)。                      |             |
|                          | <CoordIndex>                        | 1..8 = 坐标索引 |

表格 7-57: SD (68, ...)

CPL 命令

### SD (77) – 异步子程序返回轮廓的点

| 语法     | 返回值                                                                                                                |
|--------|--------------------------------------------------------------------------------------------------------------------|
| SD(77) | <p>返回当前设置的异步子例程的接近点:</p> <ul style="list-style-type: none"> <li>1 = 起点</li> <li>2 = 终点</li> <li>3 = 中断点</li> </ul> |

表格 7-58: SD(77)

### SD(131,...)– 切向刀具导向

| 语法         | 返回值         |
|------------|-------------|
| SD(131, 1) | 返回旋转轴的通道轴编号 |
| SD(131, 2) | 返回工具对称性。    |

表格 7-59: SD(131,...)

### SD(168,...)– 程序坐标偏置

| 语法                       | 返回值                      |
|--------------------------|--------------------------|
|                          | 返回一个通道坐标的程序坐标偏置（Trans）的值 |
| SD(168, 1, <CoordIndex>) | 在块准备时间点（最新的编程值）。         |
| SD(168, 2, <CoordIndex>) | 在激活的时间点（块执行）。            |
|                          | <CoordIndex> 1..8 = 坐标索引 |

表格 7-60: SD(168,...)

### SD(200,...)– 工作区数组和死区

| 语法                  | 返回值                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------|
| SD(200, 1)          | 返回通道中激活区数组的数量。                                                                                                  |
| SD(200, 1, <Area>)  | <p>频道中的区数组状态</p> <ul style="list-style-type: none"> <li>1: 通道中的区数组处于激活状态。</li> <li>0: 通道中的区数组处于激活状态。</li> </ul> |
| SD(200, 3, <Area>)  | <p>范围类型:</p> <ul style="list-style-type: none"> <li>0: 未定义</li> <li>1: 不工作区域</li> <li>2: 工作区域</li> </ul>        |
| SD(200, 11, <Area>) | 编程单位中的区数组中心点（第一个区数组轴）                                                                                           |
| SD(200, 12, <Area>) | 编程单位中的区数组中心点（第二个区数组轴）                                                                                           |
| SD(200, 21, <Area>) | 编程单位中的区数组扩展（第一个区数组轴）                                                                                            |
| SD(200, 22, <Area>) | 编程单位中的区数组扩展（第二个区数组轴）                                                                                            |
| SD(200, 31, <Area>) | 系统轴号（第一区域轴）                                                                                                     |
| SD(200, 32, <Area>) | 系统轴号（第二区域轴）                                                                                                     |
|                     | <Area> 1..32 =区数组索引                                                                                             |

表格 7-61: SD(200,...)

**SD (202, ...) – 主轴倍率**

| 语法                      | 返回值           |              |
|-------------------------|---------------|--------------|
| SD (202, <ChanSpindle>) | 返回通道主轴的有效覆盖值。 |              |
|                         | <ChanSpindle> | 1..8 =通道主轴编号 |

表格 7-62: SD (202, ...)

**SD (205, ...) – 通道主轴的速度**

| 语法                         | 返回值             |              |
|----------------------------|-----------------|--------------|
| SD (205, 1, <ChanSpindle>) | 返回包含重写的有效命令值速度。 |              |
| SD (205, 2, <ChanSpindle>) | 返回最后编程的速度。      |              |
| SD (205, 3, <ChanSpindle>) | 返回当前的实际速度。      |              |
|                            | <ChanSpindle>   | 1..8 =通道主轴编号 |

表格 7-63: SD (205, ...)

**SD (211, ...) – 主轴组的通道主轴编号**

| 语法                     | 返回值                     |            |
|------------------------|-------------------------|------------|
|                        | 将块准备时的主轴组的当前配置作为压缩列表返回。 |            |
| SD (211, <Group>, 0)   | 返回组的成员数 (0 .. 8)。       |            |
| SD (211, <Group>, <N>) | 返回组的 N. 成员的通道主轴号。       |            |
|                        | <Group>                 | 1..4 =组号   |
|                        | <N>                     | 1..8 =会员号码 |

表格 7-64: SD (211, ...)

**SD (212, ...) – 通道主轴的运动功能通道主轴的**

| 语法                         | 返回值           |              |
|----------------------------|---------------|--------------|
|                            | 返回通道主轴的运动功能   |              |
| SD (212, 1, <ChanSpindle>) | 在块准备的时间点。     |              |
| SD (212, 2, <ChanSpindle>) | 在激活的时间点。      |              |
|                            | <ChanSpindle> | 1..8 =通道主轴编号 |

表格 7-65: SD (212, ...)

**SD (222, ...) – 系统主轴的通道主轴编号**

| 语法                                                       | 返回值                   |                       |
|----------------------------------------------------------|-----------------------|-----------------------|
|                                                          | 返回当前通道的系统主轴的通道主轴号或-1。 |                       |
| SD (222, <SysSpindle>)<br>或<br>SD (222, <SysSpindle>, 1) | 块准备时的价值。              |                       |
| SD (222, <SysSpindle>, 2)                                | 在激活时间点的价值。            |                       |
|                                                          | <SysSpindle>          | 系统主轴名称或 1..60 =系统主轴编号 |

表格 7-66: SD (222, ...)

CPL 命令

SD (223, ...) – 通道主轴的系统主轴编号

| 语法                                                         | 返回值                    |                              |
|------------------------------------------------------------|------------------------|------------------------------|
|                                                            | 返回激活通道的通道主轴的系统主轴号或 -1。 |                              |
| SD (223, <ChanSpindle>)<br>或<br>SD (223, <ChanSpindle>, 1) | 块准备时的价值。               |                              |
| SD (223, <ChanSpindle>, 2)                                 | 在激活时间点的价值。             |                              |
|                                                            | <ChanSpindle>          | “S1” ... “S8” 或 1..8 =通道主轴编号 |

表格 7-67:      SD (223, ...)

SD (230, ...) – 系统主轴的系统轴号

| 语法                     | 返回值              |                       |
|------------------------|------------------|-----------------------|
| SD (230, <SysSpindle>) | 返回系统主轴的系统轴号或 -1。 |                       |
|                        | <SysSpindle>     | 系统主轴名称或 1..60 =系统主轴编号 |

表格 7-68:      SD (230, ...)

SD (231, ...) – 通道主轴的系统轴编号

| 语法                      | 返回值              |                              |
|-------------------------|------------------|------------------------------|
| SD (231, <ChanSpindle>) | 返回通道主轴的系统轴号或 -1。 |                              |
|                         | <ChanSpindle>    | “S1” ... “S8” 或 1..8 =通道主轴编号 |

表格 7-69:      SD (231, ...)

SD (232, ...) – 系统轴的系统主轴编号

| 语法                  | 返回值              |                                    |
|---------------------|------------------|------------------------------------|
| SD (232, <SysAxis>) | 返回系统轴的系统主轴号或 -1。 |                                    |
|                     | <SysAxis>        | 1..250 =系统轴索引<br>或<br>系统轴名称以引号括起来。 |

表格 7-70:      SD (232, ...)

SD (268, ...) – 累加程序坐标偏置

| 语法                        | 返回值                     |             |
|---------------------------|-------------------------|-------------|
|                           | 返回通道坐标的加性坐标偏置（ATrans）的值 |             |
| SD (268, 1, <CoordIndex>) | 在块准备时间点（最新的编程值）。        |             |
| SD (268, 2, <CoordIndex>) | 在激活的时间点（块执行）。           |             |
|                           | <CoordIndex>            | 1..8 = 坐标索引 |

表格 7-71:      SD (268, ...)

**SD(328, ...) – 精确编程**

| 语法                         | 返回值                         |
|----------------------------|-----------------------------|
| SD(328, 1)<br>或<br>SD(328) | 数控函数 "PrecProg" 的最后编程精度限制。  |
| SD(328, 2)                 | 数控函数 "Prec-Prog" 的最后编程拐角距离。 |

表格 7-72: SD(328, ...)

**SD(581, ...) – 通道轴耦合**

| 语法                   | 返回值                                                                                                                                |          |                   |         |                   |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------|---------|-------------------|
| SD(581, 0, <Master>) | <ul style="list-style-type: none"><li>通道轴编号 (&lt;Master&gt;)，如果 &lt;Master&gt; 是主轴。</li><li>0: 如果 &lt;Master&gt; 不是主轴。</li></ul>   |          |                   |         |                   |
| SD(581, <Slave>, 0)  | <ul style="list-style-type: none"><li>主轴的通道轴号，&lt;Slave&gt; 是从属轴。</li><li>0: 如果 &lt;Slave&gt; 不是从站轴。</li></ul>                     |          |                   |         |                   |
| SD(581, <Slave>, 1)  | <ul style="list-style-type: none"><li>编程单位中的 &lt;Slave&gt; 轴编程的从轴偏置。</li><li>0: 如果 &lt;Slave&gt; 不是从站轴。</li></ul>                  |          |                   |         |                   |
| SD(581, <Slave>, 2)  | <ul style="list-style-type: none"><li>程序化的耦合系数用于 &lt;Slave&gt; 轴。</li><li>0: 如果 &lt;Slave&gt; 不是从站轴。</li></ul>                     |          |                   |         |                   |
| SD(581, <Slave>, 3)  | <ul style="list-style-type: none"><li>编程单位中的 &lt;Slave&gt; 轴编程的主轴偏置。</li><li>0: 如果 &lt;Slave&gt; 不是从站轴。</li></ul>                  |          |                   |         |                   |
|                      | <table><tr><td>&lt;Master&gt;</td><td>主索引的通道轴索引 (1..8)。</td></tr><tr><td>&lt;Slave&gt;</td><td>从索引的通道轴索引 (1..8)。</td></tr></table> | <Master> | 主索引的通道轴索引 (1..8)。 | <Slave> | 从索引的通道轴索引 (1..8)。 |
| <Master>             | 主索引的通道轴索引 (1..8)。                                                                                                                  |          |                   |         |                   |
| <Slave>              | 从索引的通道轴索引 (1..8)。                                                                                                                  |          |                   |         |                   |

表格 7-73: SD(581, ...)

**SD(594, ...) – 进给组**

| 语法                  | 返回值                                                                                                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | 提供有关进给组 (FeedGrp) 和进给适应 (FeedAd) 的信息。                                                                                                                                                        |
| SD(594, <Coord>, 0) | <b>FeedGrp and FeedAd:</b> <ul style="list-style-type: none"><li>2: 坐标是当前进给组的一部分。</li><li>1: 坐标同步移动。</li><li>0: 坐标是从坐标 (例如, AXC 的从坐标)。</li><li>-1: 通道中不存在坐标。</li></ul> E. g. SD(594, "Y", 0) |
| SD(594, <Coord>, 1) | <b>FeedGrp:</b> <ul style="list-style-type: none"><li>1: FeedGrp 处于激活状态, 坐标是已编程的成员。</li><li>0: 没有成员或 FeedGrp 处于激活状态。</li><li>-1: 通道中不存在坐标。</li></ul> e. g. SD(594, "Y", 1)                   |

CPL 命令

| 语法                | 返回值                                                                                                                                                                                                        |                                               |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| SD(594,<Coord>,2) | <b>FeedAd:</b> <ul style="list-style-type: none"><li>2: FeedGrp 处于激活状态, 坐标是已编程的成员。</li><li>1: FeedAd 处于激活状态, 坐标是通过 MP 的成员。</li><li>0: FeedAd 不活动或不是成员。</li><li>-1: 通道中不存在坐标。</li></ul> e. g. SD(594,"Y",2) |                                               |
|                   | <Coord>                                                                                                                                                                                                    | 通道轴索引 (1..8)<br>或<br>坐标名称<br>或<br>9 (=方位坐标 0) |

表格 7-74: SD(594,...)

SD – 编程实例

示例:

```
30 A% = SD(2,1) A% 包含百分比形式的进给电位计的激活位置。
:
40 B% = SD(5,1,1) B% 包含激活进给速率。
:
```

程序示例:

**SD (9)**

```
N4 G75.2 X120
60 IF SD(9) = 1 THEN
N7 (MSG, 测头 2 未触发)
80 GOTO .ERROR
90 ELSE
100 XMESS = PPOS(1)
110 ENDIF
```

在编程示例 “SD (9)”中, X 轴以指定的位置方向进给。如果到达位置并且测头未触发, 则会输出一条消息 (第 N7 行), 并将其跳至标签 .ERROR。如果触发探头被触发, 可以保存当前位置, 这个位置基于 XMESS 中的程序坐标系。

SDR (概述)

| 实数                                                                                                        | READ | WRITE | FUNCT | PROC             | WAIT | ERRNO | ! |
|-----------------------------------------------------------------------------------------------------------|------|-------|-------|------------------|------|-------|---|
| <b>描述:</b> 系统数据实时 - Real_Format 中的系统内部数据<br>从 <b>实数型格式</b> 的 IndraMotion MTX 系统数据集中读取。该命令的格式和应用对应于 SD 命令。 |      |       |       |                  |      |       |   |
| <b>语法:</b> SDR (<Group>[,<Index1>[,<Index2>]])                                                            |      |       |       |                  |      |       |   |
|                                                                                                           |      |       |       | SDR 命令输出 REAL 值。 |      |       |   |

表格 7-75: 语法 SDR



**SD(1,...) – 块预运行后的目标位置**

| 语法                  | 返回值                                                                               |             |
|---------------------|-----------------------------------------------------------------------------------|-------------|
| SD(1, <CoordIndex>) | 在程序段预运行 / 重新运行中计算的所有坐标的工件处理。<br>如果没有预运行块，则返回 “0” 。<br>如果寻址未应用的轴，坐标或辅助轴，则会导致运行时错误。 |             |
|                     | <CoordIndex>                                                                      | 1..8 = 坐标索引 |

表格 7-76: SDR(1,...)

**SDR(2,...) – 电流覆盖位置**

| 语法        | 返回值                                     |
|-----------|-----------------------------------------|
| SDR(2, 1) | 以百分比的形式返回当前通道中进给倍率的当前位置。                |
| SDR(2, 3) | 以百分比形式返回主轴 1 的倍率当前位置。<br>如果没有应用主轴，则为 0。 |
| SDR(2, 4) | 以百分比形式返回主轴 2 的倍率当前位置。<br>如果没有应用主轴，则为 0。 |

表格 7-77: SDR(2,...)

**SDR(5,...) – 速度**

| 语法           | 返回值                                      |
|--------------|------------------------------------------|
| SDR(5, 1, 1) | 返回以毫米/分钟或英寸/分钟四舍五入为整数值的激活进给速度（100% 值）。   |
| SDR(5, 2, 1) | 返回以毫米/分钟或英寸/分钟四舍五入为整数值的激活快速进给速度（100% 值）。 |
| SDR(5, 1, 2) | 以每分钟输入单位返回最后编程的输入。                       |

表格 7-78: SDR(5,...)

**SDR(60,...) – 样条数据**

| 语法                 | 返回值                                                                               |                                         |
|--------------------|-----------------------------------------------------------------------------------|-----------------------------------------|
| SDR(60, 2, <Chan>) | 用 BAP (E ..) 编程的 B 样条逼近的公差值                                                       |                                         |
| SDR(60, 3, <Chan>) | 用 BAP (OE ..) 编程的 B 样条逼近的方位公差值                                                    |                                         |
|                    | 返回值是指自身通道中的准备块和第三方通道中的激活块<br>另请参阅 SD(60, 1)<br>值 >= 0: 无错误<br>值 = -2.0: 数控程序段无法确定 |                                         |
|                    | <Chan>                                                                            | 可选:<br>通道索引<br>或<br>-1 为自己的频道<br>默认: -1 |

表格 7-79: SDR(60,...)

## CPL 命令

## SDR(68,...)– 程序坐标偏置

| 语法                       | 返回值                                 |             |
|--------------------------|-------------------------------------|-------------|
|                          | 为一个通道坐标提供程序坐标偏置的总数 (Trans + ATrans) |             |
| SDR(68, 1, <CoordIndex>) | 在块准备时间点（最新的编程值）。                    |             |
| SDR(68, 2, <CoordIndex>) | 在激活的时间点（块执行）。                       |             |
|                          | <CoordIndex>                        | 1..8 = 坐标索引 |

表格 7-80: SDR(68,...)

## SDR(77,...)– 在 Asup 后返回轮廓

| 语法                                       | 返回值                                                             |        |
|------------------------------------------|-----------------------------------------------------------------|--------|
|                                          | 返回 Asup 后可能的接近点的坐标：<br>返回的位置允许使用 "G76" 在一个异步子程序中重新定位所需接近点附近的坐标。 |        |
| SDR(77, <CoordIndex>, 0)<br>或<br>SDR(77) | 根据中断块中机床坐标的 ASPRTP / REPOSTP 返回当前接近点。                           |        |
| SDR(77, <CoordIndex>, 1)                 | 返回中断块中机床坐标的起点。                                                  |        |
| SDR(77, <CoordIndex>, 2)                 | 返回中断块中机床坐标的终点。                                                  |        |
| SDR(77, <CoordIndex>, 3)                 | 返回中断程序段中机床坐标的中断点。                                               |        |
|                                          | <CoordIndex>                                                    | 机床坐标索引 |

表格 7-81: SDR(77,...)

## SD(131,...)– 切向刀具导向

| 语法          | 返回值             |
|-------------|-----------------|
| SDR(131, 3) | 以度数返回定位角度       |
| SDR(131, 4) | 以度为单位返回中间程序段角度。 |

表格 7-82: SDR(131,...)

## SD(168,...)– 程序坐标偏置

| 语法                        | 返回值                        |             |
|---------------------------|----------------------------|-------------|
|                           | 返回一个通道坐标的程序坐标偏置 (Trans) 的值 |             |
| SDR(168, 1, <CoordIndex>) | 在块准备时间点（最新的编程值）。           |             |
| SDR(168, 2, <CoordIndex>) | 在激活的时间点（块执行）。              |             |
|                           | <CoordIndex>               | 1..8 = 坐标索引 |

表格 7-83: SDR(168,...)

**SDR (202, ...) – 主轴倍率**

| 语法                       | 返回值           |               |
|--------------------------|---------------|---------------|
| SDR (202, <ChanSpindle>) | 返回通道主轴的有效覆盖值。 |               |
|                          | <ChanSpindle> | 1..8 = 通道主轴编号 |

表格 7-84: SDR (202, ...)

**SDR (205, ...) – 通道主轴速度**

| 语法                          | 返回值             |               |
|-----------------------------|-----------------|---------------|
| SDR (205, 1, <ChanSpindle>) | 返回包含重写的有效命令值速度。 |               |
| SDR (205, 2, <ChanSpindle>) | 返回最后编程的速度。      |               |
| SDR (205, 3, <ChanSpindle>) | 返回当前的实际速度。      |               |
|                             | <ChanSpindle>   | 1..8 = 通道主轴编号 |

表格 7-85: SDR (205, ...)

**SDR (219, ...) – 定向通道主轴的位置**

| 语法                       | 返回值                                |               |
|--------------------------|------------------------------------|---------------|
| SDR (219, <ChanSpindle>) | 在 M19 上次编程的主轴位置（如果 M19 未编程，则返回 0）。 |               |
|                          | <ChanSpindle>                      | 1..8 = 通道主轴编号 |

表格 7-86: SDR (219, ...)

**SDR (268, ...) – 累加程序坐标偏置**

| 语法                         | 返回值                         |             |
|----------------------------|-----------------------------|-------------|
|                            | 返回通道坐标的累加程序坐标偏置 (ATrans) 的值 |             |
| SDR (268, 1, <CoordIndex>) | 在块准备时间点（最新的编程值）。            |             |
| SDR (268, 2, <CoordIndex>) | 在激活的时间点（块执行）。               |             |
|                            | <CoordIndex>                | 1..8 = 坐标索引 |

表格 7-87: SDR (268, ...)

**SDR (300) – 北极点周围的 Epsilon 环境**

| 语法        | 返回值                                                                                |
|-----------|------------------------------------------------------------------------------------|
| SDR (300) | 活动 5 轴转换的北极点周围的活动 epsilon 环境的大小（单位为度）。<br>激活值可大于 EpsAxTrafo (MP 1030 00160) 中设置的值。 |

表格 7-88: SDR (300)

CPL 命令

SDR (328, ...) – 精确编程

| 语法                             | 返回值                         |
|--------------------------------|-----------------------------|
| SDR (328, 1)<br>或<br>SDR (328) | 数控函数 “PrecProg” 的最后编程精度限制。  |
| SDR (328, 2)                   | 数控函数 “Prec-Prog” 的最后编程拐角距离。 |

表格 7-89: SDR (328, ...)

SDR (581, ...) – 通道轴耦合

| 语法                     | 返回值                                                                                                                              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| SDR (581, 0, <Master>) | <ul style="list-style-type: none"><li>通道轴编号 (&lt;Master&gt;)，如果 &lt;Master&gt; 是主轴。</li><li>0: 如果 &lt;Master&gt; 不是主轴。</li></ul> |
| SDR (581, <Slave>, 0)  | <ul style="list-style-type: none"><li>主轴的通道轴号，&lt;Slave&gt; 是从属轴。</li><li>0: 如果 &lt;Slave&gt; 不是从站轴。</li></ul>                   |
| SDR (581, <Slave>, 1)  | <ul style="list-style-type: none"><li>编程单位中的 &lt;Slave&gt; 轴编程的从轴偏置。</li><li>0: 如果 &lt;Slave&gt; 不是从站轴。</li></ul>                |
| SDR (581, <Slave>, 2)  | <ul style="list-style-type: none"><li>程序化的耦合系数用于 &lt;Slave&gt; 轴。</li><li>0: 如果 &lt;Slave&gt; 不是从站轴。</li></ul>                   |
| SDR (581, <Slave>, 3)  | <ul style="list-style-type: none"><li>编程单位中的 &lt;Slave&gt; 轴编程的主轴偏置。</li><li>0: 如果 &lt;Slave&gt; 不是从站轴。</li></ul>                |
|                        | <Master> 主索引的通道轴索引 (1..8)。                                                                                                       |
|                        | <Slave> 从索引的通道轴索引 (1..8)。                                                                                                        |

表格 7-90: SDR (581, ...)

7. 4. 5 结构化类型的系统数据

一般信息

结构化类型的系统数据（简称 SD）形成整体系统中具有以下属性的一类数据：

- 个人 SD 可以位于易失性或非易失性存储器（永久性 SD）中。
- 数量，大小和结构可以在现有存储器中自由定义。

永久系统数据在每次系统启动时再次初始化易失性系统数据时再次初始化。这些值可以在初始化文件中指定。如果没有指定，则值被预设为“0”。

结构化变量“SV”可以用SD代替。但是，由于兼容性的原因，它们被保留。

有关系统数据使用和定义的全面信息，请参阅文档“Rexroth IndraMotion MTX 功能描述”。

访问系统数据

描述:

关键字 SD 用于在 CPL 中进行访问。与 XPath 寻址（对结构化数据的标准化访问）不同的是，使用点运算符“.”来替换分隔符“/”。使用 [ ] 对数组进行寻址，如：  
XPath: /MyArrVar[2, 3, 4]/subcomponent  
CPL: SD.MyArrVar[2, 3, 4].subcomponent

示例:

10 SD.MyChanVar=DBSEA("/dbt1/Rec",-1,-1,"Key1=1",I%)

对于频道特定的 SD，频道索引可以省略，例如：  
SD.MyArrVar[, 3, 4] 或者  
SD.MyChanVar 没有[] 用于一维通道 SD。  
任何提供整数值的 CPL 表达式都可以在方括号内编程。  
具有相同结构类型的 SD 可以相互分配。  
以下分配规则适用于基本类型：

| 名称              | 字节长度              | 描述              | C 类型           | IndraLogic 类型 | CPL    |
|-----------------|-------------------|-----------------|----------------|---------------|--------|
| 字符串             | 2 * maxLength + 1 | 字符串 (UTF-8 格式)  | char [ ]       | String()      | CHAR   |
| IsoLatin1String | maxLength + 1     | 字符串 (Latin1 格式) | char [ ]       | String()      | CHAR   |
| Byte_t          | 1                 | 有符号 8 位整数       | char           | SINT          | INT    |
| Short_t         | 2                 | 有符号 16 位整数      | short          | INT           | INT    |
| Int_t           | 4                 | 有符号 32 位整数      | int            | DINT          | INT    |
| UnsignedByte_t  | 1                 | 无符号 8 位整数       | unsigned char  | USINT         | INT    |
| UnsignedShort_t | 2                 | 无符号 16 位整数      | unsigned short | UINT          | INT    |
| UnsignedInt_t   | 4                 | 无符号 32 位整数      | unsigned int   | UDINT         | INT    |
| Float_t         | 4                 | 32 位实数          | float          | 实数            | 实数     |
| Double_t        | 8                 | 64 位实数          | double         | LREAL         | DOUBLE |
| Boolean_t       | 1                 | true、false、1、0  | char           | BOOL          | BOOL   |

表格 7-91: 基本类型的分配规则

VARINF

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

描述: 变量信息 - 变量信息

可使用 **VARINF** 命令向零件程序提交请求，查询它是否含有结构化系统数据或者永久变量。如果輸入的變量存在，該命令提供 1 作為結果。如果变量不存在，则命令为 0。

语法:

|                        |                  |
|------------------------|------------------|
| VARINF(<VariableName>) |                  |
| 其中                     |                  |
| <Variable name>        | 结构化系统日期或永久变量的名称。 |

表格 7-92: VARINF 语法

示例:

10 DIM A\$(20)  
10 I% = VARINF("@\_RES\_DOUBLE")  
...  
40 A\$ = "SD.SysSRun[1].Active"  
50 J% = VARINF(A\$)  
...

VARVAL

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| MULTYP | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

描述:

可變數值 - 變量值

CPL 命令

CPL 命令返回系統數據變量或永久變量的內容。變量名稱作為字符串表達式傳輸。因此，變量可以被間接處理。此外，變量不一定存在以鏈接程序。

如果指定的變量不存在或具有無效類型，則會生成運行時錯誤。要預先檢查系統數據變量或永久變量是否存在，請使用 VARINF（第 423 頁）命令。

VARVA 命令支持 ERRNO 参数。

|     |                        |                                                                                                                                                                                                                                                                            |
|-----|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | VARVAL(<VariableName>) |                                                                                                                                                                                                                                                                            |
|     | 其中                     |                                                                                                                                                                                                                                                                            |
|     | <Variable name>        | 结构化系统日期或永久变量的名称。                                                                                                                                                                                                                                                           |
|     | ERRNO                  | <p>CPL 变量</p> <p>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。</p> <p>以下为可能的返回值：</p> <ul style="list-style-type: none"><li>0：访问正常</li><li>&lt;0：错误（有关详细的错误描述，请参阅第 3.14.5 章“评价 CPL 中功能错误的变量 ERRNOERRNO”第 69 页）</li></ul> <p>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。</p> |

表格 7-93: VARVAL 语法

|     |                                                                                                                                                                               |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 示例: | <pre>程序 1 DIM A\$(30) : DIM B\$(30) : IDX%=4 1 A\$="SD.MySDVar[" : B\$="].Active" 1 RES% = VARVAL(A\$+STR\$(IDX%)+B\$) ; in RES% is the content of SD.MySDVar[4].Active</pre> |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|     |                                                                                                                                                                                                                                   |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 限制: | <ul style="list-style-type: none"><li>只支持系统数据变量和永久变量（全局或本地 CPL 变量）。</li><li>传输的字符串是变量名称，而不是要计算的表达式。</li><li>如果访问数组变量，则所有索引必须可用作常量（数字），而不能用作表达式。</li><li>只能读取简单的类型（INTEGER, REAL, DOUBLE, BOOLEAN, STRING）（既不是数组也不是结构体）。</li></ul> |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

SDLOAD

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

|     |                                                                                                                                                       |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述: | <p><b>系统数据 LOAD</b> – 加载结构化系统数据</p> <p>CPL 命令 SDLOAD 可用于从文件中读取系统数据。可以指定几个根元素。指定的子组件被忽略。始终读取从根元素开始的所有数据。</p> <p>可以完全读取根数组元素（不包含数组索引）或单独读取（使用数组索引）。</p> |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------|

|    |                                                            |                                                   |
|----|------------------------------------------------------------|---------------------------------------------------|
| 语法 | SDLOAD(<FileName>,<Root-Element_1>[,...,<Root-Element_n>]) |                                                   |
|    | <FileName>                                                 | <p>目标文件的目录路径和文件名</p> <p>如果没有指定路径，则从当前目录中读取文件。</p> |

|                |                                                                                                                                                                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Root element> | 要读取数据组的根元素                                                                                                                                                                                                                                                                                                                                               |
| ERRNO          | <p>CPL 变量</p> <p>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。</p> <p>错误由变量的相应值提供。</p> <p>以下为可能的返回值：</p> <p>0：访问正常</p> <ul style="list-style-type: none"> <li>-1：参数错误</li> <li>-7：数据无法读取</li> <li>-9：无法访问源文件</li> <li>-10：包括路径在内的目标文件太长</li> <li>-20：根标签错误</li> <li>-21：无法访问文件</li> <li>-35：系统数据访问内部错误</li> </ul> <p>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。</p> |

表格 7-94: SDLOAD 语法

**语法示例：**

```
SDLOAD("/usrfep/SDDatUsr.xml","DgnUsr[1]",ERRNO)
```

從文件“SDDatUsr.xml”中讀取用戶特定系統數據“DgnUsr [1]”並將其加載到“/usrfep”（用戶FEPROM）。

## SDSAVE

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** 系统数据 SAVE - 保存存结构化系统数据

使用 CPL 命令 SDSAVE，系统数据保存在**新创建**的文件中。已存在的同名文件将被覆盖。可以指定几个根元素。指定的子组件被忽略。始终保存从根元素开始的所有数据。

可以保存根数组元素（不包含数组索引）或单独保存（使用数组索引）。

## 语法

SDSAVE (<FileName>, <Root-Element\_1>[, ..., <Root element n>])

|            |                                                    |
|------------|----------------------------------------------------|
| <FileName> | <p>目标文件的目录路径和文件名</p> <p>如果没有路径指定，则会在当前目录中搜索文件。</p> |
|------------|----------------------------------------------------|

|                  |             |
|------------------|-------------|
| 〈Axis selection〉 | 系统轴索引或系统轴名称 |
|------------------|-------------|



| <EID no> | 扩展 Sercos ID 编号 (S-x-xxxx.xx.xx, P-x-xxxx.xx.xx)                                                                                                                                                                                                                                                                                             |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ERRNO    | <p>CPL 变量</p> <p>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。</p> <p>以下为可能的返回值：</p> <p>0：访问正常</p> <p>-1：参数错误</p> <p>-7：数据无法读取</p> <p>-10：包括路径在内的目标文件太长</p> <p>-21：无法访问文件</p> <p>-27：含有无效参数类型的命令</p> <p>-30：内存不足</p> <p>-36：服务通道被阻止</p> <p>-37：未知标识号</p> <p>-38：无效标识号</p> <p>-39：轴号无效</p> <p>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。</p> |

表格 7-96: SCS3 语法  
提供没有单位或缩放的参数内容。

 此命令只用于带扩展 ID 编号的 Sercos 驱动器

 无法读取包含列表（用逗号隔开的若干值）的参数。在这种情况下，控制器将发布错误消息。

 使用 Sercos 多路复用通道可加速数据传输。

如果驱动数据在 Sercos 驱动器报文中，则从报文读取此数据（请参阅 Sercos 驱动器的《参数手册》）。否则，将直接在驱动器读取驱动数据。

如果 **其他** 应用程序访问驱动数据，则不能 **同时** 对这种驱动数据进行访问。参数 ERRNO 可用来响应零件程序中的这个错误。重复访问可提供所需的驱动数据。

 持续访问驱动器数据可防止访问其他应用程序！

可通过评价错误变量 EERRNO 来提供对零件程序中错误的响应。

示例:

```
10 LAGE%=SCS3(1,"S-0-51.0.0",ERRNO)
 第一轴的实际位置值被分配到整数型变量 LAGE%。

12 IF ERRNO = 0 THEN
 误差估算

13 REM***正确读出实际位置***

14 ELSE
```

CPL 命令

```
15 REM***无法正确读出实际位置***
16 ENDIF
```

SCS3L

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** SerCoS 3 列表 - Sercos - 读取参数列表

某些 Sercos 参数存储为列表。这些列表可以使用命令“SCS3L”读取。由于列表的长度（所需内存）未知，因此所读取的列表元素被存储在 ASCII 文件中。后续可使用 CPL 文件命令处理读取的数据。

通过 SCS3L 命令，如果命令中指定的文件不存在，则创建该文件。现有文件的内容将被覆盖。

|              |                                                                                                                                                                                                                                                                                                      |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>语法:</b>   | SCS3L(<AxisIndex>,<EID no>,<FileName>)]                                                                                                                                                                                                                                                              |
| <Axis index> | 系统轴索引或系统轴名称                                                                                                                                                                                                                                                                                          |
| <EID no>     | 扩展 Sercos ID 编号 (S-x-xxxx.xx.xx, P-x-xxxx.xx.xx)                                                                                                                                                                                                                                                     |
| <FileName>   | 要储存所读取列表的 ASCII 文件的名称。                                                                                                                                                                                                                                                                               |
| ERRNO        | CPL 变量<br><br>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。<br><br>以下为可能的返回值：<br>0: 访问正常<br>-1: 参数错误<br>-7: 数据无法读取<br>-10: 包括路径在内的目标文件太长<br>-21: 无法访问文件<br>-27: 含有无效参数类型的命令<br>-30: 内存不足<br>-36: 服务通道被阻止<br>-37: 未知标识号<br>-38: 无效标识号<br>-39: 轴号无效<br><br>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。 |

表格 7-97: SCS3L 语法

如果其他应用程序在访问驱动数据，则在某些时候无法对驱动数据进行访问。参数 <ERRNO> 可用于响应零件程序中的这个错误。重复访问可提供所需的驱动数据。

 此命令只用于带扩展 标识号的 Sercos 驱动器

 持续访问驱动器数据可妨害访问其他应用程序！

**示例:**

```
10 SCS3L(1,"P-0-0021","OUTFILE.txt")
```

将驱动器 1 的轴示波器的标识号列表写入文件“OUTFILE.txt”。

WSCS

描述:

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

便于写入 Sercos 驱动器参数。

除个别值外，还可以编写最多 16 个值的值列表。

先决条件:

- 该参数必须在 Sercos 阶段 4 中有效且可编辑。
- 参数值必须处于允许的范围内。
- 驱动器必须使用 Sercos 接口连接到控制器；控制器和驱动器之间的循环通讯正在运行。

- 

调用 WSCS 时，不能处理数控块。这可以通过使用 WAIT 命令来实现。
- 

WSCS 可选用于数控函数 "WriteID3(WID3)"。
- 

要读取 Sercos 变频器参数，可以使用 CPL 命令 "SCS3"。
- 

对 Sercos 传动参数进行不当或随意的修改可能会导致工件和 / 或机器的损坏，并导致危险和不可预测的机器反应。

正确使用驱动器中可用的 Sercos 变频器参数信息。对此，请参见驱动器文档。

语法:

|                                                                      |                                                                     |
|----------------------------------------------------------------------|---------------------------------------------------------------------|
| WSCS (<Axisselection>, <ID-No>, <Arraylength>, <Value(s)>, {<Unit>}) |                                                                     |
| <Axis selection>                                                     | 系统轴索引或系统轴名称                                                         |
| <ID no>                                                              | Sercos 标识号 (S-x-xxxx, P-x-xxxx, S-x-xxxx. xx. xx, P-x-xxxx. xx. xx) |
| <Field length>                                                       | 0<br>写入个别参数<br>1..16<br>写入列表参数                                      |
| <Value (s)>                                                          | 对于数组长度= 0:<br>CPL 表达式<br>对于数组长度= 1..16:<br>数组变量的名称                  |

CPL 命令

|          |                                                                                                                                                                                                                                                                                                       |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {<Unit>} | 可选的。<br><br>1: （默认）通过考虑单位和缩放进行书写<br><br>0: 没有单位和缩放的书写                                                                                                                                                                                                                                                 |
| ERRNO    | CPL 变量<br><br>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。<br><br>以下为可能的返回值：<br><br>0: 访问正常<br>-1: 参数错误<br>-13: 数据无法写入<br>-26: 含有无效参数值的命令<br>-27: 含有无效参数类型的命令<br>-36: 服务通道被阻止<br>-37: 未知标识号<br>-38: 无效标识号<br>-39: 轴号无效<br>-42: 违反限值<br>-43: 元素号无效<br><br>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。 |

表格 7-98: WSCS 语法

**写入个别参数:** 使用<ArrayLength> 0，写入各个参数。  
<Value (s)>可以是任意的 CPL 表达式。

第 5 个系统轴的参数用值写入。在分配的服务通道的情况下，等待 100 毫秒并重复写入操作:

```
...
N10 WAIT
000 .WRTORQUE
001 WSCS (5,"S-0-92",0,400.0,ERRNO)
002 IF ERRNO <> 0 THEN
003 IF ERRNO = -36 THEN
004 WAIT(,100)
005 GOTO .WRTORQUE
006 ELSE
007 REM error handling
009 ENDIF
010 ENDIF
...
```

插图 7-3: 写单个参数 (例 1)  
通道轴 Z 的参数写入:

```
...
N10 WAIT
000 SAX% = SD(23,"Z")
001 WSCS (SAX%,"S-0-92",0,400.0,ERRNO)
002 IF ERRNO <> 0 THEN
007 REM error handling
010 ENDIF
...
```

插图 7-4: 写单个参数 (例 2)

**写参数列表** 参数列表是通过传输字段变量的使用元素的名称和编号来编写的。  
<Field length> (1 至 16) 指定要写入的列表元素的编号。

<Value (s)>必须是一维数组变量的名称。

在下面的程序部分中，字段变量 VALUE! 是用值创建和写入的。

字段变量的名称和要使用的元素的数量被传送给 WSCS。

程序

```
...
N10 WAIT
000 DIM VALUE!(5)
001 VALUE!(1) = 3.5
002 VALUE!(2) = 0.2
003 WSCS("X", "P-0-4006", 2, VALUE!, ERRNO)
004 IF ERRNO <> 0 THEN
005 REM error handling
006 ENDIF
...
```

插图 7-5: 写参数列表

特殊特性: 使用 {<unit>} = 0, "SCS3" 读取的参数值可以再次写入驱动器。

7.4.7

PLC 接口

BITIF

|      |      |       |       |      |      |       |   |
|------|------|-------|-------|------|------|-------|---|
| BOOL | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|

描述: **BIT 接口** - 位接口

此命令允许访问 数控和 PLC 之间的数字接口:

- 所有的输入和输出都可以被读取。
- 也可以写入通道接口中的 CPL 客户输出 Ch\_Cp101 - 16。

语法:

BITIF(<BitSignal>[,<Index>[,<IF Unit>]])

其中

|              |                                                                                                                                                  |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <Bit signal> | 位偏置或符号地址<br>位偏置:<br>0...31 全局接口<br>0...111 通道接口<br>0...95 轴接口<br>0...95 主轴接口<br>0...7 高速接口<br>有关符号地址, 请参阅 "《Rexroth IndraMotion MTX PLC 接口》" 手册。 |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------|



|                                                                 |                                                                                                                                                                                                        |
|-----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Address>                                                       | 起始于区数组开始处的相关字节地址<br>控制器根据激活的 PLC 检查参数。                                                                                                                                                                 |
| <Size>                                                          | 数据类型的大小：<br>1: 字节<br>2: 字<br>4: 双字                                                                                                                                                                     |
| PLC (<Type>, <EmptyParam. >, <Address>, <Size> [, <DataType>] ) |                                                                                                                                                                                                        |
| <Data type>                                                     | <b>可选:</b><br>访问 PLC 上操作数的类型<br>0: INTEGER （默认）<br>1: 实数                                                                                                                                               |
| ERRNO                                                           | CPL 变量<br>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。<br>以下为可能的返回值<br>0: 访问正常<br><0: 错误（有关详细的错误描述，请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页）<br>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。 |

表格 7-100:      PLC 语法



使用 <Data type> 参数时，访问（读、写）相同操作数时必须使用相同类型。

示例:                   :  
30 REM 从输入 10 开始读取 2 个字节  
40 I% = PLC(1,,10,2)  
:

7. 4. 8      激活功能

NCF

| CHAR | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

描述:      **数控函数** – 激活数控函数和模态辅助功能

返回模态组或者辅助功能组 <NC function> 内最新编程的数控函数的语法。可以查询控制的所有模态组以及辅助功能组中定义的辅助功能。

可选地，中断块的模态也可以在由参数控制的异步子程序中检索。

*中断块是*

- – 关于预先运行的块 – 选择目标块的前一块
- – 关于正常的异步子程序 – 一个从它跳转到 Asup 的数控块。

其中结果将被存储的变量必须是类型 “定维字符数组”。

在程序运行时检测到类型冲突，并通过错误消息进行确认。



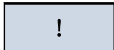




所有读取值都是指块准备处理 CPL 块的时间。

如果调用程序也包含数控块，则块处理通常在机器处理之前执行。如果程序执行受到确定当前机床数据或状态的功能的影响，请消除程序段准备和当前机床状态之间的“时间偏置”。但是，这仅适用于访问其执行通道的函数。

在这种情况下，请在呼叫前的行中使用“WAIT”命令。这会停止块准备，直到“WAIT”之前所有的块完全处理完毕。



**“MCODS”命令不能与块预运行一起使用！原因是 MCODES 访问激活点时间数据（插补）。插补在块预运行时被抑制。因此，当调整开始时，激活状态仍然与块预运行之前的时间点的状态相同。**

对于大量的“MCODS”命令，存在常见的 CPL 功能。这些特殊的 CPL 命令提供相同和相似的数据。它也被称为 CPL 命令，在列表下面的列表中“也请参阅”

这些命令提供了一个返回值（请参阅 第 章“错误返回值”第 482 页）。

语法：

|                                                                                      |                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MCODS (<Type>,<Channel>,<Version>,<Buffer>,<Size>,<AxisNumber>,<IdentNumber>[,<Pl>]) |                                                                                                                                                                                                                                                                  |
| <Type>                                                                               | 整数型表达式<br>指出要执行的函数。<br>下表列出所有可用函数。                                                                                                                                                                                                                               |
| <Channel>                                                                            | 整数型表达式<br>指出受函数支配的通道。                                                                                                                                                                                                                                            |
| <Version>                                                                            | 已初始化的整数型或实数型变量（ <b>非</b> 常数！）。<br>如果调用该命令时变量内容为 0，由 <Type> 指定的函数会立即将请求的数据写入 <Buffer>。<br>另外，该命令返回 <Version> 中提供的数据的版本 ID。如果此版本 ID 在下次调用该命令时仍处于变量中，则该命令不会立即写入所请求的数据，而只会在下次修改 <Buffer> 中的数据后才执行。<br>因此，程序循环可以运行，直到某个通道达到某个状态为止。但是，在循环中包含超时条件（例如计数器或经过的时间段）以防止无限循环！ |
| <Buffer>                                                                             | 该命令在 <Buffer> 中返回所请求的数据值。<br><i>根据数据类型，&lt;Buffer&gt; 必须是</i> <ul style="list-style-type: none"><li>一个“整数”类型的简单变量，“实数”，“Double”</li><li>“整数型”、“实数型”、“双精度型”数组变量</li><li>字符串变量（一维字符数组）。</li></ul>                                                                  |

表格 7-102: MCODES 语法



对于数组或字符串变量，只能输入不带维度/索引的变量名称！

CPL 命令

|                         |                                                                                              |
|-------------------------|----------------------------------------------------------------------------------------------|
| <Size>                  | 整数型表达式<br>指定 <Buffer> 的数组大小。<br>如果 <Buffer> 不是数组变量，而是简单的“整数型”、“实数型”或“双精度”变量，则为 <Size> 输入值 1。 |
| <Axis number>           | 整数型表达式<br>指定系统轴的轴号。                                                                          |
| <Identification number> | 整数型表达式<br>为所有轴提供来自循环轴报文的 <ident number> 值。                                                   |

表格 7-103: MCODS 参数



数组变量的大小必须先通过 DIM 命令定义。参数 <Size> 不能超过！  
在 MCODS 命令之前和同一个文件中标注这些变量！

|      |                  |
|------|------------------|
| <P1> | 取决于 <Type>的可选参数。 |
|------|------------------|

表格 7-104: MCODS 语法 3

命令概述 MCODS(...)函数概述  
MCODS(...)

| 命令提供:            | 语法:            | 另请参阅:         |
|------------------|----------------|---------------|
| 位置               |                |               |
| 轴命令位置            | MCODS (1. .)   | SPOS          |
| 轴命令位置            | MCODS (2. .)   | SPOS          |
| 驱动器的实际轴位置        | MCODS (35. .)  | APOS          |
| CNC 的实际轴位置       | MCODS (147. .) | –             |
| 实际轴值（工件坐标系）      | MCODS (38. .)  | –             |
| 滞后               | MCODS (4. .)   | –             |
| 轴程序值（程序坐标系统）     | MCODS (37. .)  | PCS           |
| 包括偏置的已编程终点位置     | MCODS (16. .)  | WCS           |
| 不包括偏置的已编程终点位置    | MCODS (23. .)  | PCS           |
| ACS1 中的轴坐标       | MCODS (73. .)  | –             |
| ACS0 中的轴坐标       | MCODS (109. .) | ACS           |
| 工件坐标系（WCS）中的剩余行程 | MCODS (83. .)  | –             |
| 轴坐标系 ACS1 中的剩余行程 | MCODS (85. .)  | –             |
| 速率和速度            |                |               |
| 路径进给             | MCODS (3. .)   | SDR (5, 1, 2) |
| 已编程的路径进给         | MCODS (24. .)  | SDR (5, 1, 2) |
| 已编程的路径进给         | MCODS (142. .) |               |
| 已激活编程的路径进给       | MCODS (143. .) |               |
| 点动速度             | MCODS (27. .)  | –             |

| 命令提供:           | 语法:             | 另请参阅:                           |
|-----------------|-----------------|---------------------------------|
| 主轴命令速度、主轴切削速度   | MCODS (5..)     | SD (205, 2, ...)                |
| 实际主轴速度          | MCODS (36..)    | SDR (205, 3, ...)               |
| 最大主轴速度          | MCODS (19..)    | –                               |
| 最小主轴速度          | MCODS (20..)    | –                               |
| 已编程的主轴速度        | MCODS (25..)    | SDR (205, 2, ...)               |
| 当前轴速率           | MCODS (119..)   | –                               |
| 当前轴速度           | MCODS (120..)   | –                               |
| 编程轴速度           | MCODS (121..)   | –                               |
| 最大轴速度           | MCODS (131..)   | –                               |
| 轴速度的峰值和最大值      | MCODS (145..)   | –                               |
| 轴加速度的峰值和最大值     | MCODS (146..)   | –                               |
| <b>状态</b>       |                 |                                 |
| “InPos”状态       | MCODS (6..)     | BITIF (“IAX_INPOS”, ...)        |
| “接近参考点”状态       | MCODS (26..)    | BITIF (“IAX_REFKNOWN”, ...)     |
| 状态 “测试模式” (空运行) | MCODS (29..)    | BITIF (“ICH_DRYRUN”, ...)       |
| SAV 和 IPO 状态    | MCODS (32..)    | BITIF (“ICH_STATE”, ...)        |
| “停留时间激活”状态      | MCODS (39..)    | –                               |
| “服从确认激活的辅助函数”状态 | MCODS (40..)    | –                               |
| 状态 “启用的块传输”     | MCODS (41..)    | BITIF (“QCH_TRANSFERLOCK”, ...) |
| “行程命令”状态        | MCODS (47..)    | BITIF (“IAX_TRVCMD”, ...)       |
| “进给禁止”状态        | MCODS (49..)    | BITIF (“QAX_DRVLOCK”, ...)      |
| 通道等待状态          | MCODS (87..)    | –                               |
| 通道复位; 等待        | MCODS (129, ..) | –                               |
| <b>电位计</b>      |                 |                                 |
| 进给电位计的值         | MCODS (7..)     | SDR (2, 1)                      |
| 主轴电位计的值         | MCODS (8..)     | SDR (202, ...)                  |
| 轴电位计的值          | MCODS (50..)    | –                               |
| <b>修正</b>       |                 |                                 |
| 激活刀具半径修正数       | MCODS (11..)    | TCV (0)                         |
| 激活刀具半径修正        | MCODS (12..)    | TCV (4)                         |
| 激活刀具修正表的名称      | MCODS (13..)    | –                               |
| 激活轴 / Z0 表的名称   | MCODS (14..)    | –                               |
| 激活轴 Z0 值        | MCODS (15..)    | ZOV                             |
| 激活刀具修正          | MCODS (54..)    | –                               |
| 激活数据块定位         | MCODS (112..)   | –                               |
| 激活定位表的路径和名称     | MCODS (113..)   | –                               |

CPL 命令

| 命令提供:                        | 语法:            | 另请参阅:                       |
|------------------------------|----------------|-----------------------------|
| 刀具位置修正                       | MCODS (114. .) | DPC                         |
| 总定位                          | MCODS (115. .) | PMV                         |
| 激活程序坐标偏置                     | MCODS (116. .) | SDR (168)                   |
| 激活叠加程序坐标偏置                   | MCODS (117. .) | SDR (268)                   |
| 激活程序坐标偏置总和                   | MCODS (118. .) | –                           |
| 在线修正值                        | MCODS (89. .)  | –                           |
| 在线修正状态                       | MCODS (90. .)  | –                           |
| 当前温度补偿                       | MCODS (132. .) | –                           |
| 当前丝杠误差补偿                     | MCODS (133. .) | –                           |
| 当前交叉补偿                       | MCODS (134. .) | –                           |
| 当前补偿偏置                       | MCODS (135. .) | –                           |
| 操作模式                         |                |                             |
| 通道模式                         | MCODS (31. .)  | BITIF (“ICH_OPMODE”, . . .) |
| 轴模式                          | MCODS (48. .)  | –                           |
| 系统结构                         |                |                             |
| 进给轴、辅助轴、主轴的数目;<br>移动类型、驱动器类型 | MCODS (34. .)  | –                           |
| 通道的数目                        | MCODS (44. .)  | –                           |
| 轴数                           | MCODS (45. .)  | –                           |
| 轴名                           | MCODS (33. .)  | AXINF                       |
| 激活通道轴名称                      | MCODS (59. .)  | AXADR\$                     |
| 通道轴名称 (默认设置)                 | MCODS (60. .)  | –                           |
| 轴-通道分配                       | MCODS (43. .)  | SD (25, . . .)              |
| 轴-通道默认分配                     | MCODS (58. .)  | –                           |
| 来自循环轴报文的 ID 编号               | MCODS (62. .)  | –                           |
| 坐标名称                         | MCODS (78. .)  | AXINF\$                     |
| 坐标数目                         | MCODS (82. .)  | –                           |
| 零点偏置的当前数据块                   | MCODS (99. .)  | ZOV                         |
| 零点偏置的当前总和                    | MCODS (100. .) | ZOV                         |
| 机床坐标名称                       | MCODS (106. .) | –                           |
| 机床坐标数目                       | MCODS (107. .) | –                           |
| 测量单元                         |                |                             |
| 轴单位 (默认设置)                   | MCODS (61. .)  | –                           |
| 轴单位                          | MCODS (53. .)  | –                           |
| 编程类型 (英寸/公制)                 | MCODS (18. .)  | NCF (“G70”)                 |
| 直径编程                         | MCODS (91. .)  | NCF (“DIA”)                 |

| 命令提供:          | 语法:            | 另请参阅:                             |
|----------------|----------------|-----------------------------------|
| <b>辅助功能</b>    |                |                                   |
| 辅助函数的组号        | MCODS (94. .)  | –                                 |
| 激活辅助函数组的语法     | MCODS (65. .)  | –                                 |
| <b>主轴</b>      |                |                                   |
| 主轴的运动功能        | MCODS (63. .)  | SD (212, 2)                       |
| 主轴的档位范围        | MCODS (64. .)  | BITIF ("ISP_GEAR1..4SELECT", ...) |
| 通道主轴的组分配       | MCODS (65. .)  | –                                 |
| 切换档位是否激活的信息    | MCODS (66. .)  | –                                 |
| 自动 / 手动档位范围选择  | MCODS (67. .)  | –                                 |
| 主轴名称           | MCODS (102. .) | –                                 |
| 主轴数目           | MCODS (103. .) | –                                 |
| 主轴系统编号         | MCODS (104. .) | –                                 |
| 主轴轴编号          | MCODS (105. .) | –                                 |
| 实际主轴位置         | MCODS (108. .) | –                                 |
| 主轴扭矩           | MCODS (110. .) | –                                 |
| 主轴性能           | MCODS (111. .) | –                                 |
| <b>驱动器</b>     |                |                                   |
| 制造版本           | MCODS (55. .)  | –                                 |
| 控制设备类型         | MCODS (56. .)  | –                                 |
| 电机类型           | MCODS (57. .)  | –                                 |
| <b>其他</b>      |                |                                   |
| 零件程序中的消息       | MCODS (28. .)  | –                                 |
| 主程序的路径和名称      | MCODS (30. .)  | –                                 |
| 回归轮廓策略和慢跑运动的记录 | MCODS (46. .)  | –                                 |
| 特定客户数据         | MCODS (42. .)  | –                                 |
| 可选停止 (已激活)     | MCODS (68. .)  | BITIF ("ICH_BLKSLASH", ...)       |
| 跳过块 (激活)       | MCODS (69. .)  | BITIF ("ICH_OPTSTOP", ...)        |
| 程序自动重选激活       | MCODS (70. .)  | BITIF ("QCH_RESELOFF", ...)       |
| 旋转角            | MCODS (122. .) | SCL (0)                           |
| 镜射和转动极位置       | MCODS (123. .) | SCL (1, ...)                      |
| 偏置坐标           | MCODS (124. .) | COF                               |
| 激活缩放系数         | MCODS (125. .) | SCL (2, ...)                      |
| 坐标镜射, 激活       | MCODS (126. .) | SCL (2, ...)                      |
| <b>阻止预运行</b>   |                |                                   |

CPL 命令

| 命令提供:       | 语法:            | 另请参阅: |
|-------------|----------------|-------|
| 调整状态        | MCODS (128,..) | –     |
| 块预运行重新标记的状态 | MCODS (127,..) | –     |

表格 7-105: MCODS 的命令概述



在下表中，语法中部分指定的参数是整数常量。除了这些常量，整数变量也可以被编程。但是，它们在命令被调用时的指定值必须分配给那个。

MCODS (1, ...) – 轴命令位置

|                 |                                                                                                                                                                |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中以升序和通道无关的顺序提供 <b>系统中所有进给轴和辅助轴</b> 的指令位置: <ul style="list-style-type: none"><li>– 对于直线轴，单位为毫米</li><li>– 旋转轴，单位角度</li></ul> “设置实际值”（例如 SetPos）包含在值中。 |
| 更新:             | 循环运行                                                                                                                                                           |
| <Buffer> 的数据类型: | 双精度、数组                                                                                                                                                         |
| 语法:             | <div>MCODS (1, -1, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                                                                        |

表格 7-106: 语法 MCODES (1..)

MCODS (2, ...) – 轴命令位置（整数）

|                 |                                                                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中以升序和通道无关的顺序提供 <b>系统中所有进给轴和辅助轴</b> 的指令位置: <ul style="list-style-type: none"><li>线性轴以 0.0001 毫米计量</li><li>回转轴以 0.0001 度计量</li></ul> “设置实际值”（例如 SetPos）包含在值中。 |
| 更新:             | 循环运行                                                                                                                                                                   |
| <Buffer> 的数据类型: | 整数、数组                                                                                                                                                                  |
| 语法:             | <div>MCODS (2, -1, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                                                                                |

表格 7-107: 语法 MCODES (2..)

MCODS (3, ...) – 路径进给

|                 |                                                                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中，按照毫米/分钟递增的顺序返回三个值，其中 <Channel> 的当前路径馈送（包括馈送电位计）以毫米/分钟为单位: <ol style="list-style-type: none"><li>指令速度外部分配给插补器。</li><li>实际插补器速度 (=当前路径速度)。</li><li>插补器内部指令速度。由于应用程序（例如进给适应功能），它可能与外部分配的相比发生了变化。</li></ol> 以毫米/转（G95）进给编程时，它以毫米/分钟为单位提供路径进给。 |
| 更新:             | 循环运行                                                                                                                                                                                                                                                   |
| <Buffer> 的数据类型: | 实数、数组                                                                                                                                                                                                                                                  |
| 语法:             | <div>MCODS (3, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, 3)</div>                                                                                                                                                                              |

表格 7-108: 语法 MCODES (3..)

MCODS (4, ... ) – Lag

|                 |                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 按 <Buffer> 中的升序返回所有轴的滞后: <ul style="list-style-type: none"><li>线性轴的单位为毫米</li><li>回转轴的单位为度</li></ul> 如果滞后传输（通过 Sercos 参数）不被驱动器支持，则返回 0.0。 |
| 更新:             | 循环运行                                                                                                                                     |
| <Buffer> 的数据类型: | 实数、数组                                                                                                                                    |
| 语法:             | <div>MCODS (4, -1, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                                                  |

表格 7-109:      语法 MCODES (4. . )

MCODS (5, ... ) – 主轴指令速度，主轴切削速度

|                 |                                                                                                                                                                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 按 <Buffer> 中的升序 (S 或 S1, S2, S3 等) 返回 <Channel> 中所有主轴的主轴指令速度或切削速度。如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的数据。否则，提供通道主轴数据。使用有效的 G96 时，切削速度以米/分钟为单位。 否则，当前主轴指令速度以 rpm 提供。否则，当前主轴指令速度以 rpm 提供。计算中包含电位计，速度限制 (SMin, SMax) 和由档位范围引起的限制。如果不存在主轴，则 0.0 在相应位置的 <Buffer> 中返回。 |
| 更新:             | 循环运行                                                                                                                                                                                                                                                                   |
| <Buffer> 的数据类型: | 实数、数组                                                                                                                                                                                                                                                                  |
| 语法:             | <div>MCODS (5, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                                                                                                                                                                   |

表格 7-110:      语法 MCODES (5. . )

MCODS (6, ... ) – InPos

|                 |                                                                                                                                                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中，为每个 <b>进给和辅助轴</b> 提供值 0 或 1 作为其 InPos 信号，它们以上升且与通道无关的顺序排列:<br>轴就位: 1<br>轴未就位: 0<br><br>如果轴在参数化的 InPos 窗口 (MP 1015 00100) 中，及没有行进命令待定，则轴已就位。（另参见 MCODES (47. . )）。 |
| 更新:             | 循环运行                                                                                                                                                                           |
| <Buffer> 的数据类型: | 整数、数组                                                                                                                                                                          |
| 语法:             | <div>MCODS (6, -1, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                                                                                        |

表格 7-111:      语法 MCODES (6. . )

MCODS (7, ... ) – 进给电位器的值

|                 |                                                                           |
|-----------------|---------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中，提供以 1/100 % 计量的 <Channel> 的进给率电位计的当前值。                       |
| 更新:             | 循环运行                                                                      |
| <Buffer> 的数据类型: | 实数                                                                        |
| 语法:             | <div>MCODS (7, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, 1)</div> |

表格 7-112:      语法 MCODES (7. . )

MCODS (8, ... ) – 主轴电位计的值

|     |                                                                  |
|-----|------------------------------------------------------------------|
| 描述: | 在 <Buffer> 中，提供以 1/100% 计量的进给率电位计的当前值，它们以升序 (S 或 S1、S2、S3 等) 排列。 |
|-----|------------------------------------------------------------------|

CPL 命令

如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的数据。否则，提供通道主轴数据。

更新: 循环运行  
<Buffer> 的数据类型: 实数、数组  
语法:

```
MCODS(8,<Channel>,<Version>,<Buffer>,<Size>)
```

表格 7-113: 语法 MCODES(8,...)

MCODS(9,...)

描述: 不适用于 MTX

MCODS(10,...)

描述: 不适用于 MTX

MCODS(11,...) – 有效的刀具半径校正编号

描述: 在 <Buffer> 中，提供在 <Channel> 中激活的刀具半径修正数。  
如果半径修正未激活，返回 -1。  
更新: 每个块切换后  
<Buffer> 的数据类型: 整数  
语法:

```
MCODS(11,<Channel>,<Version>,<Buffer>,1)
```

表格 7-114: 语法 MCODES(11..)

MCODS(12,...) – 激活刀具半径修正

描述: 在 <Buffer> 中，提供在 <Channel> 中激活的刀具半径修正。  
如果半径修正未激活，返回 0.0。  
更新: 每个块切换后  
<Buffer> 的数据类型: 实数  
语法:

```
MCODS(12,<Channel>,<Version>,<Buffer>,1)
```

表格 7-115: 语法 MCODES(12..)

MCODS(13,...) – 激活刀具修正表的名称

描述: 在 <Buffer> 中，提供在 <Channel> 中激活的刀具修正表名称。  
如果未激活，则返回 3 个空格作为字符串。  
更新: 有了块切换  
<Buffer> 的数据类型: 字符、数组  
语法:

```
MCODS(13,<Channel>,<Version>,<Buffer>,<Size>)
```

表格 7-116: 语法 MCODES(13..)

MCODS(14,...) – 激活轴 Z0 表的名称

描述: 在 <Buffer> 中，提供在 <Channel> 中激活的轴 Z0 表名称。  
如果未激活，则返回 3 个空格作为字符串。  
更新: 有了块切换  
<Buffer> 的数据类型: 字符、数组  
语法:

```
MCODS(14,<Channel>,<Version>,<Buffer>,<Size>)
```

表格 7-117: 语法 MCODES(14..)



MCODS (15, ...) – 激活轴 Z0 值

|                 |                                                                                                                                                                                                                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中返回所有机床坐标的前三个库的 <Channel> 中以毫米或度为单位的 Z0 值。<br>如果没有偏置是激活的，则返回 0.0。<br>以下顺序适用: <ul style="list-style-type: none"><li>库 1 中第一个机床坐标的偏置</li><li>库 1 中第二个机床坐标的偏置</li><li>库 1 中第 n 个机床坐标的偏置</li><li>库 2 中第一个机床坐标的偏置</li><li>:</li><li>库 2 中第 n 个机床坐标的偏置</li><li>:</li><li>库 3 中第 n 个机床坐标的偏置</li></ul> 所有 5 个偏置库的值由命令 MCODES (99, ...) 提供 |
| 更新:             | 每个块切换后                                                                                                                                                                                                                                                                                                                                       |
| <Buffer> 的数据类型: | 实数、数组                                                                                                                                                                                                                                                                                                                                        |
| 语法:             | <div>MCODS (15, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                                                                                                                                                                                                                                        |

表格 7-118: 语法 MCODES (15..)

MCODS (16, ...) – 包括输入帮助的当前块的结束位置

|                 |                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中，提供涉及工件坐标的、所有 进给率和辅助轴 激活块的终点位置，它们以上升且与通道无关的顺序排列：<br>线性轴的单位为毫米<br>旋转轴，单位角度<br>该计算已包括了所有的偏置值。<br>这些值 不 包括“设置实际值”（如 SetPos）。 |
| 更新:             | 有了块切换                                                                                                                                 |
| <Buffer> 的数据类型: | 实数、数组                                                                                                                                 |
| 语法:             | <div>MCODS (16, 0&lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                                                 |

表格 7-119: 语法 MCODES (16..)

MCODS (17, ...)

描述: 不适用于 MTX

MCODS (18, ...)– 英制/公制编程

|                 |                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 返回 <Buffer> 中的 <Channel> 中轴的编程类型: <ul style="list-style-type: none"><li>0: 英寸</li><li>1: 公制</li><li>2: 度</li><li>3: 轴不存在</li></ul> |
| 更新:             | 有了块切换                                                                                                                              |
| <Buffer> 的数据类型: | 整数                                                                                                                                 |
| 语法:             | <div>MCODS (18, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, 1)</div>                                                         |

表格 7-120: 语法 MCODES (18..)

CPL 命令

MCODS (19,...) – 最大主轴速度

|                 |                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中，以 rpm 为单位返回最大容许主轴速度，它们以升序排列（S 或 S1、S2、S3 等）。<br><br>如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的数据。否则，提供通道主轴数据。<br><br>该计算包括速度限制。 |
| 更新:             | 有了块切换                                                                                                                                        |
| <Buffer> 的数据类型: | 实数、数组                                                                                                                                        |
| 语法:             | <div>MCODS (19,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div>                                                            |

表格 7-121: 语法 MCODES (19..)

MCODS (20,...) – 最小主轴速度

|                 |                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中，以 rpm 为单位返回最小容许主轴速度，它们以升序排列（S 或 S1、S2、S3 等）。<br><br>如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的数据。否则，提供通道主轴数据。<br><br>该计算包括速度限制。 |
| 更新:             | 有了块切换                                                                                                                                        |
| <Buffer> 的数据类型: | 实数、数组                                                                                                                                        |
| 语法:             | <div>MCODS (20,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div>                                                            |

表格 7-122: 语法 MCODES (20..)

MCODS (21,...)

描述: 不适用于 MTX

MCODS (22,...)

描述: 不适用于 MTX

MCODS (23,...) – 编程的结束位置无偏置

|                 |                                                                     |
|-----------------|---------------------------------------------------------------------|
| 描述:             | 如 MCODES (16...), 但没有偏置。                                            |
| 更新:             | 有了块切换                                                               |
| <Buffer> 的数据类型: | 实数、数组                                                               |
| 语法:             | <div>MCODS (23.0,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div> |

表格 7-123: 语法 MCODES (23..)

MCODS (24,...) – 编程的路径进给

|                 |                                                                        |
|-----------------|------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中以毫米/分钟返回编程的 <Channel> 的路径进给。                               |
| 更新:             | 每个块切换后                                                                 |
| <Buffer> 的数据类型: | 实数                                                                     |
| 语法:             | <div>MCODS (24,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,1)</div> |

表格 7-124: 语法 MCODES (24..)

MCODS (25,...) – 编程的主轴速度

描述: 在 <Buffer> 中，以 rpm 为单位返回已编程主轴速度，它们以升序排列（S 或 S1、S2、S3 等）。

如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的数据。否则，提供通道主轴数据。

更新: 每个块切换后  
<Buffer> 的数据类型: 实数数组

语法: 

MCODS (25, <Channel>, <Version>, <Buffer>, <Size>)

表格 7-125: 语法 MCODES (25..)

MCODS (26, ...) – 接近参考点

描述: 在 <Buffer> 中，为每个 **进给和辅助轴** 返回 0 或 1 作为“接近的参考点”信号，它们以上升且与通道无关的顺序排列：

接近的参考点: 1  
未接近的参考点: 0

更新: 改变后  
<Buffer> 的数据类型: 整数、数组

语法: 

MCODS (26, -1, <Version>, <Puffer>, <Size>)

表格 7-126: 语法 MCODES (26..)

MCODS (27, ...) – 点动速度

描述: 在 <Buffer> 中以升序和通道无关的顺序返回 **系统中所有进给轴和辅助轴** 的当前点动速度：

线性轴的单位为毫米/分  
回转轴的单位为 rpm

更新: 改变后  
<Buffer> 的数据类型: 实数、数组

语法: 

MCODS (27, -1<Version>, <Buffer>, <Size>)

表格 7-127: 语法 MCODES (27..)

MCODS (28, ...) – 注解(消息)

描述: 通过 <Buffer> 中的 MZA 命令，MSG 命令或 PRN# (0, ...) 返回最新的编程注解。

<Channel> 参数选择综合通道或通道特定注解：  
-1 返回最新的通道综合注解（MSG）。  
0..60 返回指定通道中 MSG 和 PRN# 命令的最新注解。

更新: 改变后  
<Buffer> 的数据类型: 字符、数组

语法: 

MCODS (28, <Channel>, <Version>, <Buffer>, 80)

表格 7-128: 语法 MCODES (28..)

MCODS (29, ...) – 测试模式

描述: 如果测试模式未激活，则在 <Buffer> 中返回 1。否则 0。

更新: 改变后  
<Buffer> 的数据类型: 整数

语法: 

MCODS (29, 0, <Version>, <Buffer>, 1)

表格 7-129: 语法 MCODES (29..)

MCODS (30,...)– 主程序的路径和名称

|                 |                                                                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中， 返回 <Channel> 中编址的主程序路径， 包括 <Buffer> 中作为字符串的名称。<br>调用命令时，<Version> 中的值被忽略。<br>文件名的长度加上路径的长度在 <Size> 中指定。在内部文件系统中， 包含文件名的路径不能超过 100 个字符。<br>对于存储在已装入文件系统中的文件，<Size> 中的值取决于外部文件系统支持的最大允许字符数以及路径和文件名。 |
| 更新:             | 改变后                                                                                                                                                                                                             |
| <Buffer> 的数据类型: | 字符、数组                                                                                                                                                                                                           |
| 语法:             | <div>MCODS (30,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div>                                                                                                                               |

表格 7-130: 语法 MCODES (30..)

MCODS (31,...)– 通道模式

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述: | 在 <Buffer> 中返回 <Channel> 中的激活 <ul style="list-style-type: none"><li>主模式</li><li>次级模式</li></ul> 以下值是为主模式定义的: <ul style="list-style-type: none"><li>0: 无操作模式。因此没有激活的过程。</li><li>1: 点动模式<br/>轴可以点动 (+/-)。</li><li>2: 接近参考点<br/>使用信号 Manual+ / Manual- 可启动轴。</li><li>3: 保留</li><li>4: 手动数据输入<br/>可指定加工各个数控块。</li><li>5: 自动（继续块）<br/>零件程序已处理完毕。</li><li>6: 自动（程序块）。<br/>对零件程序的各个块按顺序进行处理。准备好每个单个块，并用数控启动使其启动。</li><li>7: 自动（单步）<br/>在某些情况下， 数控会从零件程序中的单个数控块中生成并准备若干块。在这种操作模式下， 数控启动总是用来将单个块传给插补器， 以进行处理。</li><li>8: 教学<br/>轴可以点动 (+/-) 或手轮进给。可指定加工各个数控块。</li><li>9: 保留</li><li>10: 自动（单块）<br/>使用数控启动， 将已在零件程序中单个数控块生成并准备好的所有块都传给插补器， 以进行处理。</li><li>11: 返回到轮廓<br/>可手动将轴从轮廓移开， 并可以自动或手动将其重启。</li><li>12: CPL 调试器（程序块模式）</li></ul> |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

与零件程序一样处理单块。

- 13: CPL 调试器（继续块模式）  
对下一个中断点前的所有块进行处理。

以下值是为次级模式定义的:

- 0: 无辅助轴激活
- 1: 覆盖激活。可指定加工各个数控块。

更新: 改变后

<Buffer> 的数据类型: 整数、数组

语法:

|                                               |
|-----------------------------------------------|
| MCODS (31, <Channel>, <Version>, <Buffer>, 2) |
|-----------------------------------------------|

表格 7-131: 语法 MCODES (31..)

MCODS (32, ...) – SAV 和 IPO 状态

描述: 返回 <Channel> 的 <Buffer> 中

- SAV 状态（块准备）和
- 内插器状态。

为 SAV 状态定义以下值:

- 1: 操作模式未激活。可对进程进行选择。
- 2: 操作模式已就绪。进程可以启动。
- 3: 操作模式已激活。正在处理程序或数控块。
- 4: 保留
- 5: 保留
- 6: 操作模式出错。该错误只能通过“控制复位”或“程序取消选择”清除
- 7: 保留
- 8: 复位当前已执行
- 9: 选定一个程序并已准备好（例如已链接）
- 10: “取消剩余行程”被触发，并且仍未完成。
- 11: 操作模式激活并再次处理现有缓冲区
- 12: 操作模式已就绪。进程处于程序开头且可以启动。
- 13: 处理缓存的数控块规格中的所有块。等待下一个规范。

为 IPO 状态定义以下值:

- 1: 插补器运行中
- 2: 由于进给保持，插补器关闭
- 3: 插补器停止轴

更新: 改变后

<Buffer> 的数据类型: 整数、数组

语法:

|                                               |
|-----------------------------------------------|
| MCODS (32, <Channel>, <Version>, <Buffer>, 2) |
|-----------------------------------------------|

表格 7-132: 语法 MCODES (32..)

MCODS (33, ...) – 轴名称

描述: 如果 <Channel> = -1, 则 <Buffer> 返回所有配置系统轴 1-100 的名称，它们由 “0”（0 字节）分开并以升序排列，每个名称 9 字节。  
如果 <Channel> = -2, 则 <Buffer> 返回所有配置系统轴 101-200 的名称，它们由 “0”（0 字节）分开并以升序排列，每个名称 9 字节。

CPL 命令

如果 <Channel> = -3, 则 <Buffer> 返回所有配置系统轴 201-250 的名称, 它们由 "0" (0 字节) 分开并以升序排列, 每个名称 9 字节。

如果 <Channel> = 现有通道号, 则 <Buffer> 提供所有 **指定通道中的轴** 的名称, 它们由 "0" (0 字节) 分开以升序排列。

对于 8 字符以下的名称, 填充第 8 个已填充字符之前的空间。

<Buffer> 的大小在 <Size> 中指定。对于 100 个轴, 最多可以是 (9\*100) 900 个字节。

有关示例, 请参阅 [第 4 章 "编程示例" 第 483 页](#)。

更新: 控制启动后

<Buffer> 的数据类型: 字符、数组

语法: 

MCODS (33, <Channel>, <Version>, <Buffer>, <Size>)

表格 7-133: 语法 MCODES (33..)

MCODS (34, ...)

描述: 不适用于 MTX

MCODS (35, ...) – 驱动器的实际轴位置

描述: 提供驱动器通过 Sercos 接口以 <Buffer> 中的升序和通道无关顺序传输到 CNC 的所有 **进给轴和辅助轴** 的实际位置:

- 对于直线轴, 单位为毫米
- 旋转轴, 单位角度

这些值 **不** 包括"设置实际值" (如 SetPos)。

 由 MCODES (35 ..) 提供的值与 MCODES (147 ..) 的不同之处在于轴错误更正 LSEC, CCOMP 和温度补偿。

更新: 循环运行

<Buffer> 的数据类型: 实数、数组

语法: 

MCODS (35, -1<Version>, <Buffer>, <Size>)

表格 7-134: 语法 MCODES (35..)

MCODS (36, ...) – 主轴列表速度

描述: 在 <Buffer> 中, 以 rpm 为单位返回实际主轴速度, 它们以升序排列 (S 或 S1、S2、S3 等)。如果传输给 <Channel> 的是 -1 或 0, 则 <Buffer> 提供系统主轴的数据; 否则, 将返回通道主轴数据。计算中包含电位计, 速度限制 (SMin, SMax) 和由档位范围引起的限制。如果不存在主轴, 则 0.0 在相应位置的 <Buffer> 中返回。

更新: 循环运行

<Buffer> 的数据类型: 实数、数组

语法: 

MCODS (36, <Channel>, <Version>, <Buffer>, <Size>)

表格 7-135: 语法 MCODES (36..)

MCODS (37, ...) – 编程的坐标值

描述: 在 <Buffer> 中, 为 **系统中的进给率轴** 提供与工件相关的插补器目标位置, 它们以上升且与通道无关的顺序排列:

- 对于直线轴, 单位为毫米
- 旋转轴, 单位角度

"设置实际值" (例如 SetPos) 和轴 Z0 (G54.x ... G59.x) **不包含**在值中。

|                 |                                                      |
|-----------------|------------------------------------------------------|
| 更新:             | 循环运行                                                 |
| ⟨Buffer⟩ 的数据类型: | 双精度、数组                                               |
| 语法:             | <div>MCODS (37, -1⟨Version⟩, ⟨Buffer⟩, ⟨Size⟩)</div> |

表格 7-136:      语法 MCODES (37..)

MCODS (38, ...) – 工件坐标系中的实际轴值

|                 |                                                                                                                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 返回 ⟨Buffer⟩ 中的 “实际值” 。这些值与 ⟨Channel⟩ 的相应（笛卡尔）工件坐标系有关。<br><br>这些值是使用逐个通道机床特定的运动学正向变换（轴变换）从实际轴位置计算得出的。如果没有运动轴转换激活，则 MCODES（38）将返回与 MCODES（147）相同的值。<br><br>应用 MCODES（38）需要相应设置机器参数 9030 00002。此机床参数用于组态工件坐标系中是否计算实际值以及计算实际值的频率。 |
| 更新:             | 循环运行                                                                                                                                                                                                                          |
| ⟨Buffer⟩ 的数据类型: | 双精度、数组                                                                                                                                                                                                                        |
| 语法:             | <div>MCODS (38, ⟨Channel⟩, ⟨Version⟩, ⟨Buffer⟩, ⟨Size⟩)</div>                                                                                                                                                                 |

表格 7-137:      语法 MCODES (38..)

MCODS (39, ...) – 暂停时间激活

|                 |                                                          |
|-----------------|----------------------------------------------------------|
| 描述:             | 如果 ⟨Channel⟩ 中的停留时间处于激活状态，则在 ⟨Buffer⟩ 中返回 1。否则 0。        |
| 更新:             | 改变后                                                      |
| ⟨Buffer⟩ 的数据类型: | 整数                                                       |
| 语法:             | <div>MCODS (39, ⟨Channel⟩, ⟨Version⟩, ⟨Buffer⟩, 1)</div> |

表格 7-138:      语法 MCODES (39..)

MCODS (40, ...) – 需要确认激活的辅助功能

|                 |                                                          |
|-----------------|----------------------------------------------------------|
| 描述:             | 如果 ⟨Channel⟩ 中的辅助功能正在等待确认，则返回 ⟨Buffer⟩ 中的是 1。否则 0。       |
| 更新:             | 改变后                                                      |
| ⟨Buffer⟩ 的数据类型: | 整数                                                       |
| 语法:             | <div>MCODS (40, ⟨Channel⟩, ⟨Version⟩, ⟨Buffer⟩, 1)</div> |

表格 7-139:      语法 MCODES (40..)

MCODS (41, ...) – 已启用块传输

|                 |                                                          |
|-----------------|----------------------------------------------------------|
| 描述:             | 如果 ⟨Channel⟩ 中设置数控输入信号 “块传输禁止” ，则在 ⟨Buffer⟩ 中返回 1。否则 0。  |
| 更新:             | 改变后                                                      |
| ⟨Buffer⟩ 的数据类型: | 整数                                                       |
| 语法:             | <div>MCODS (41, ⟨Channel⟩, ⟨Version⟩, ⟨Buffer⟩, 1)</div> |

表格 7-140:      语法 MCODES (41..)

MCODS (42, ...) – 特定于客户的数据

|     |                                                                                                  |
|-----|--------------------------------------------------------------------------------------------------|
| 描述: | 从 ⟨Buffer⟩ 中的 ⟨Channel⟩ 返回客户特定的数据。<br><br>在⟨P1⟩中，一个介于 0 和 65535 之间的整数值可以传送给客户服务器，以便在命令调用时选择某些数据。 |
|-----|--------------------------------------------------------------------------------------------------|

CPL 命令

该命令旨在用于“数控内核”领域的客户特定内部开发。

- 更新: 改变后
- <Buffer> 的数据类型: 取决于客户特定数据的类型
- 语法:

```
MCODS (42, <Channel>, <Version>, <Buffer>, <Size>, <Pl>)
```

表格 7-141: 语法 MCODES (42..)

MCODS (43, ...) – 轴分配: 通道

- 描述: 在 <Buffer> 中, 返回每个系统轴的以下默认分配:
- ≥0: 同步轴的通道号
- 1: 轴为异步轴
- 2: 轴为主轴
- 3: 轴未定义

- 更新: 改变后
- <Buffer> 的数据类型: 整数、数组
- 语法:

```
MCODS (43, -1<Version>, <Buffer>, <Size>)
```

表格 7-142: 语法 MCODES (43..)

MCODS (44, ...) – 通道数

- 描述: 在 <Buffer> 中, 按照升序返回以下内容
- 可用的用户通道数
  - 接口的通道数
  - 内部和外部通道数

- 更新: 控制启动后
- <Buffer> 的数据类型: 整数、数组
- 语法:

```
MCODS (44, -1, <Version>, <Buffer>, 3)
```

表格 7-143: 语法 MCODES (44..)

MCODS (45, ...) – 轴数

- 描述: 在 <Buffer> 中, 按照升序返回以下内容
- 系统中所有驱动器（轴和主轴）的最大索引。这对于在所有驱动器上进行磨削是必不可少的。如果不存在间隙, 则其为系统中的驱动器数。
  - 系统中轴的最大索引。例如, 这对于轴接口的大小是必不可少的。如果不存在间隙, 则其为系统中的轴数。
  - 系统中主轴的最大数。例如, 这对于主轴接口的大小是必不可少的。

- 更新: 控制启动后
- <Buffer> 的数据类型: 整数、数组
- 语法:

```
MCODS (45, -1, <Version>, <Buffer>, 3)
```

表格 7-144: 语法 MCODES (45..)

MCODS (46, ...) – 返回轮廓策略和点动运动

- 描述: 按指定 <Channel> 的升序, 在 <Buffer> 中返回以下内容:
- 返回轮廓模式
  - 返回轮廓点
  - 记录点动运动的状态。



以下数值适用于 返回轮廓模式:

- 1: 自动返回轮廓
- 2: 单块返回轮廓
- 3: 手动返回轮廓

对于 返回轮廓的点:

- 1: 返回到起点
- 2: 返回至结束点
- 3: 返回至中断点

对于 记录状态:

- 0: 记录未激活
- 1: 记录激活

更新: 改变后  
<Buffer> 的数据类型: 整数, 改变  
语法:

```
MCODS (46, <Channel>, <Version>, <Buffer>, 3)
```

表格 7-145: 语法 MCODES (46..)

MCODS (47,...)- 行程命令

描述: 对于<Channel> = -1, <Buffer>按照升序返回所有系统轴的行程命令信号。  
对于<Channel> =现有通道号, <Buffer>返回 指定通道中所有轴的行程命令信号, 然后按照升序排列异步轴 的行程命令信号。  
目前行程命令: 1  
行程命令不存在: 0  
只要一个轴通过手动输入或输入零件程序来执行运动运动, 就始终设置一个运动指令。

更新: 循环运行  
<Buffer> 的数据类型: 整数、数组  
语法:

```
MCODS (47, <Channel>, <Version>, <Buffer>, <Size>)
```

表格 7-146: 语法 MCODES (47..)

MCODS (48,...)- 轴模式

描述: 对于<Channel> = -1, <Buffer>按照升序返回所有系统轴的操作模式。  
对于<Channel> =现有通道号, <Buffer>返回 指定通道中所有轴的操作模式, 然后按照升序排列异步轴 的操作模式。

可能的操作模式返回值:

- 0: 无操作模式。因此没有处理是激活的
- 1: 点动模式  
轴可以点动 (+/-)。
- 2: 接近参考点  
使用信号 Manual+ / Manual- 可启动轴。
- 3: 保留
- 4: 手动数据输入  
可指定加工各个数控块。
- 5: 自动 (继续块)  
零件程序已处理完毕。

CPL 命令

- 6:自动（程序块）。  
对零件程序的各个块按顺序进行处理。准备好每个单个块，并用数控启动使其启动。
- 7:自动（单步）  
在某些情况下，数控会从零件程序中的单个数控块中生成并准备若干块。在这种操作模式下，数控启动总是用来将单个块传给插补器，以进行处理。
- 8:保留
- 9:保留
- 10:自动（单块）  
使用数控启动，将已在零件程序中单个数控块生成并准备好的所有块都传给插补器，以进行处理。
- 11:返回到轮廓  
可手动将轴从轮廓移开，并可以自动或手动将其重启。

更新:改变后

<Buffer> 的数据类型:整数、数组

语法:

MCODS (48,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-147: 语法 MCODES (48..)

MCODS (49,...)– 进给禁止

- 描述:对于 <Channel> = -1，<Buffer> 提供所有系统轴的进给禁止信号，它们以升序排列。  
对于 <Channel> = 现有通道号，则 <Buffer> 提供所有 指定通道中的轴（其后为异步轴的进给禁止信号） 的进给禁止信号，它们以升序排列。  
1: 进给禁止存在  
0: 进给禁止不存在

更新:改变后

<Buffer> 的数据类型:整数、数组

语法:

MCODS (49,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-148: 语法 MCODES (49..)

MCODS (50,...)– 轴电位计

- 描述:对于 <Channel> = -1，<Buffer> 提供所有系统轴的（轴电位计或通道电位计）各轴相关倍率的值，按升序排列（以 0.01% 计量）。  
对于 <Channel> = 现有通道号，<Buffer> 为每个 指定通道中的轴 提供通道电位计的值（ 其后为所有异步轴的轴电位计的值 ），它们以升序排列，以 0.01% 计量。

更新:改变后

<Buffer> 的数据类型:实数、数组

语法:

MCODS (50,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-149: 语法 MCODES (50..)

MCODS (51,...)

- 描述:不适用于 MTX

MCODS (52, ...)

描述: 不适用于 MTX

MCODS (53, ...)- 轴的测量单位

描述: 对于 <Channel> = -1, <Buffer> 返回所有系统轴的测量单位（公制、英制、度），它们以升序排列。

对于 <Channel> = 现有通道号, 则 <Buffer> 为每个 **指定通道中的轴（其后为所有异步轴的测量单位）** 提供测量单位，它们以升序排列。

在异步线性轴处于“点动”和“进给至参考点”轴操作模式的情况下, 由轴接口来确定其单位。如果未输入轴操作模式, 则返回“公制”为其单位。

在同步线性轴处于“点动”和“进给至参考点”轴操作模式的情况下, 由轴接口来确定其单位。在其它操作模式下, 根据公制/英制通道单位（G70/G71）来确定其单位。

对于旋转轴和主轴, 单位是度。对于具有位置编程的 Hirth 轴, 提供对应测量单位。

单位可能的返回值:

- 0: 英寸
- 1: 公制
- 2: 度
- 3: 轴不存在
- 4: 位置可编程的 Hirth 轴

更新: 改变后

<Buffer> 的数据类型: 整数、数组

语法: 

MCODS (53, <Channel>, <Version>, <Buffer>, <Size>)

表格 7-150: 语法 MCODES (53..)

MCODS (54, ... )激活刀具修正

描述: 在 <Buffer> 中, 返回 <Channel> 中激活的刀具半径修正数。

顺序:

- 半径修正
- L3 长度修正
- L1 长度修正
- L2 长度修正
- 刃口位置
- 修正类型

以下为已定义的修正类型:

- 0: 未修正
- 1: 钻孔刀具
- 2: 铣刀
- 3: 车刀
- 4: 弯头刀具

更新: 块切换后

<Buffer> 的数据类型: 实数、数组

CPL 命令

语法:

MCODS(54,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-151: 语法 MCODES(54..)

MCODS(55,...) – 制造商版本

描述: 返回驱动器的制造商版本。  
通过输入系统轴在参数 <axis number> 中选择轴。（编号 0 也返回第一轴）。制造商版本对应于 Sercos 参数 S-0-0030。参数 <Buffer> 中提供了最多 40 个字符的数组。

更新: 控制启动后

<Buffer> 的数据类型: 字符、数组

语法: MCODES(55,-1,<Version>,<Buffer>,<Size>,<AxisNumber>)

表格 7-152: 语法 MCODES(55..)

MCODS(56,...) – 控制装置类型

描述: 输出驱动器的控制设备类型。  
通过指定系统轴在参数 <Axis number> 中选择轴（0 号也提供第一个轴）。控制器设备类型对应于 Sercos 参数 S-0-0140。参数 <Buffer> 中提供了最多 40 个字符的数组。

更新: 控制启动后

<Buffer> 的数据类型: 字符、数组

语法: MCODES(56,-1,<Version>,<Buffer>,<Size>,<AxisNumber>)

表格 7-153: 语法 MCODES(56,...)

MCODS(57,...) – 电机类型

描述: 返回驱动器的电机类型。  
通过输入系统轴在参数 <axis number> 中选择轴。（编号 0 也返回第一轴）。电机类型对应于 Sercos 参数 S-0-0141。参数 <Buffer> 中提供了最多 40 个字符的数组。

更新: 控制启动后

<Buffer> 的数据类型: 字符、数组

语法: MCODES(57,-1,<Version>,<Buffer>,<Size>,<AxisNumber>)

表格 7-154: 语法 MCODES(57..)

MCODS(58,...) – 默认轴分配: 通道

描述: 在 <Buffer> 中，返回每个系统轴的以下默认分配:  
≥0: 同步轴的通道号  
-1: 轴为异步轴  
-2: 轴为主轴  
-3: 轴未定义  
在 250 轴的情况下，<Buffer> 必须具有 <Size> 250（整数）。

更新: 启动后

<Buffer> 的数据类型: 整数

语法: MCODES(58,-1,<Version>,<Puffer>,<Size>)

表格 7-155: 语法 MCODES(58..)

MCODS (59, ...) – 激活通道轴名称 MCODES

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | <p>对于 &lt;Channel&gt; = -1, &lt;Buffer&gt; 返回所有帶系統轴索引 1-100 的<b>激活通道轴</b>, 它们由 "0" (0 字节) 分开並以升序排列, 每个名称 9 字节。</p> <p>对于 &lt;Channel&gt; = -2, &lt;Buffer&gt; 返回所有帶系統轴索引 101-200 的<b>激活通道轴</b>, 它们由 "0" (0 字节) 分开並以升序排列, 每个名称 9 字节。</p> <p>对于 &lt;Channel&gt; = -1, &lt;Buffer&gt; 返回所有帶系統轴索引 201-250 的<b>激活通道轴</b>, 它们由 "0" (0 字节) 分开並以升序排列, 每个名称 9 字节。</p> <p>如果 &lt;Channel&gt; = 现有通道号, 则 &lt;Buffer&gt; 提供所有 <b>指定通道中的轴</b> 的名称, 它们由 "0" (0 字节) 分开以升序排列。</p> <p>对于 8 字符以下的名称, 填充第 8 个字符之前的空间。</p> <p>&lt;Buffer&gt; 的大小在 &lt;Size&gt; 中指定。对于 100 个轴, 最多可以是 (9*100) 900 个字节。</p> <p>有关示例, 请参阅 <a href="#">第 章 "编程示例" 第 483 页</a> 。</p> |
| 更新:             | 改变后                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <Buffer> 的数据类型: | 字符、数组                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 语法:             | <div>MCODS (59, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

表格 7-156: 语法 MCODES (59..)

MCODS (60, ...) – 通道轴名称的默认设置

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | <p>对于 &lt;Channel&gt; = -1, &lt;Buffer&gt; 返回所有帶系統轴索引 1-100 的<b>激活通道轴</b>, 它们由 "0" (0 字节) 分开並以升序排列, 每个名称 9 字节。</p> <p>对于 &lt;Channel&gt; = -2, &lt;Buffer&gt; 返回所有帶系統轴索引 101-200 的<b>激活通道轴</b>, 它们由 "0" (0 字节) 分开並以升序排列, 每个名称 9 字节。</p> <p>如果 &lt;Channel&gt; = -3, 则 &lt;Buffer&gt; 返回所有帶系統轴索引 201-250 的<b>激活通道名称</b>, 它们由 "0" (0 字节) 分开並以升序排列, 每个名称 9 字节。</p> <p>如果 &lt;Channel&gt; = 现有通道号, 则 &lt;Buffer&gt; 提供所有 <b>指定通道中的轴</b> 的名称, 它们由 "0" (0 字节) 分开以升序排列。</p> <p>对于 8 字符以下的名称, 填充第 8 个已填充字符之前的空间。</p> <p>&lt;Buffer&gt; 的大小在 &lt;Size&gt; 中指定。对于 100 个轴, 最多可以是 (9*100) 900 个字节。</p> <p>有关示例, 请参阅 <a href="#">第 章 "编程示例" 第 483 页</a> 。</p> |
| 更新:             | 控制启动后                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <Buffer> 的数据类型: | 字符、数组                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 语法:             | <div>MCODS (60, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

表格 7-157: 语法 MCODES (60..)

MCODS (61, ...) – 轴的测量单位的默认设置

|     |                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述: | <p>如果 &lt;Channel&gt; = -1, 则 &lt;Buffer&gt; 返回所有系统轴的测量单位 (公制、英制、度), 按默认设置。</p> <p>如果 &lt;Channel&gt; = 现有通道号, 则 &lt;Buffer&gt; 返回所有 <b>通道指定轴</b> 的测量单位 (公制、英制、度), 它们按默认设置。</p> <ul style="list-style-type: none"><li>对于 <b>异步</b> 线性轴, 使用 "公制" 单位。</li><li>对于 <b>同步</b> 线性轴, 测量单位取决于启动后的参数 CHAN / Ch / Ini / NcResetState (MP 7060 00010) ): "公制/英制" (G70/G71)</li><li>对于旋转轴, 单位是度。对于具有位置编程的 Hirth 轴, 提供对应单位。</li></ul> |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

CPL 命令

单位可能的返回值：

- 0:英寸
- 1:公制
- 2:度
- 3:轴不存在
- 4:位置可编程的 Hirth 轴
- 在 250 轴的情况下，<Buffer> 必须具有 <Size> 250（整数）。

更新:控制启动后

<Buffer> 的数据类型:整数、数组

语法:

```
MCODS (61,<Channel>,<Version>,<Buffer>,<Size>)
```

表格 7-158: 语法 MCODES (61..)

MCODS (62,...)– 来自循环轴报文的标识号

描述: 为所有轴提供来自循环轴报文的 <ident number> 值。

如果 <标识号> 不在循环报文中，则输出值 NCS\_MCO\_NOT\_IN\_CYCL\_AT\_C (-2147483648)。

该值在 Sercos 缩放中为“整数”。

在 250 轴的情况下，<Buffer> 必须具有 <Size> 250（整数）。

更新:循环运行

<Buffer> 的数据类型:整数、数组

语法:

```
MCODS (62,-1,<Version>,<Buffer>,<Size>,<ident number>)
```

表格 7-159: 语法 MCODES (62..)

MCODS (63,...)– 主轴的运动功能

描述: 返回主轴的运动函数。

运动函数的编码:

0:主轴未定义

1:顺时针旋转，无冷却液

2:顺时针旋转，有冷却液

3:逆时针旋转，无冷却液

4:逆时针旋转，有冷却液

5:主轴停止

6:定向主轴

在 32 主轴的情况下，<Buffer> 必须具有 <Size> 32（整数）。

如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的数据。否则，提供通道主轴数据。

更新:改变后

<Buffer> 的数据类型:整数、数组

语法:

```
MCODS (63,<Channel>,<Version>,<Buffer>,<Size>)
```

表格 7-160: 语法 MCODES (63..)

MCODS (64,...)– 档位级别

描述: 返回主轴的齿轮档。

编码齿轮档：

- 40: 自动齿轮档选择
- 41: 齿轮档 1
- 42: 齿轮档 2
- 43: 齿轮档 3
- 44: 齿轮档 4
- 48: 脱开齿轮

在 32 主轴的情况下，<Buffer> 必须具有 <Size> 32（整数）。

如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的数据。否则，提供通道主轴数据。

更新：  
<Buffer> 的数据类型：  
语法：

```
MCODS (64, <Channel>, <Version>, <Buffer>, <Size>)
```

表格 7-161: 语法 MCODES (64..)

MCODS (65, ...) – 通道主轴组的分配

描述: 返回通道主轴的组分配:

组分配:

- 0: 无激活的组
- 1 .. 4: 组编号

例如，在 8 通道主轴的情况下，<Buffer> 必须具有 <Size> 8（整数）。

如果为 <Channel> 返回 -1，则 <Buffer> 返回激活通道的数据。否则，返回所选通道的数据。

更新：  
<Buffer> 的数据类型：  
语法：

```
MCODS (65, <Channel>, <Version>, <Buffer>, <Size>)
```

表格 7-162: 语法 MCODES (65..)

MCODS (66, ...) – 齿轮切换

描述: 返回切换档是否激活的信息:

- 0: 切换档位未激活
- 1: 切换档位已激活

在 32 主轴的情况下，<Buffer> 必须具有 <Size> 32（整数）。

如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的数据。否则，提供通道主轴数据。

更新：  
<Buffer> 的数据类型：  
语法：

```
MCODS (66, <Channel>, <Version>, <Buffer>, <Size>)
```

表格 7-163: 语法 MCODES (66..)

MCODS (67, ...) – 自动 / 手动齿轮切换

描述: 提供自动或手动换档选择:

- 0: 手动
- 1: 自动

CPL 命令

在 32 主轴的情况下，<Buffer> 必须具有 <Size> 32（整数）。  
如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的数据。否则，提供通道主轴数据。

|                 |                                                                                   |
|-----------------|-----------------------------------------------------------------------------------|
| 更新:             | 改变后                                                                               |
| <Buffer> 的数据类型: | 整数、数组                                                                             |
| 语法:             | <div>MCODS (67,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div> |

表格 7-164: 语法 MCODES (67..)

MCODS (68,...)– 跳过块

|                 |                                                                        |
|-----------------|------------------------------------------------------------------------|
| 描述:             | 返回 <Buffer> 中数控输出信号 <b>激活跳过块</b> 和 <Channel> 的输入信号 <b>跳过块</b> 的状态。     |
| 更新:             | 改变后                                                                    |
| <Buffer> 的数据类型: | 整数、数组                                                                  |
| 语法:             | <div>MCODS (68,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,2)</div> |

表格 7-165: 语法 MCODES (68..)

MCODS (69,...)– 可选停止

|                 |                                                                        |
|-----------------|------------------------------------------------------------------------|
| 描述:             | 返回 <Buffer> 中数控输出信号 <b>选择停止有效</b> 和 <Channel> 的输入信号 <b>选择停止</b> 的状态。   |
| 更新:             | 改变后                                                                    |
| <Buffer> 的数据类型: | 整数、数组                                                                  |
| 语法:             | <div>MCODS (69,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,2)</div> |

表格 7-166: 语法 MCODES (69..)

MCODS (70,...)– 自动程序重新选择激活

|                 |                                                                        |
|-----------------|------------------------------------------------------------------------|
| 描述:             | 在<Buffer>中返回是否在指定的通道中应用 <b>自动程序重新选择</b> :<br>0: 函数未应用<br>1: 函数已应用      |
| 更新:             | 请求后立即                                                                  |
| <Buffer> 的数据类型: | 整数                                                                     |
| 语法:             | <div>MCODS (70,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,1)</div> |

表格 7-167: 语法 MCODES (70..)

MCODS (71,...)– 工件坐标

|                 |                                                                                   |
|-----------------|-----------------------------------------------------------------------------------|
| 描述:             | 返回<Buffer>中指定通道的 <b>工件坐标</b> 值（WCS）：首先是所有空间坐标，然后是通道的伪坐标。                          |
| 更新:             | 循环运行                                                                              |
| <Buffer> 的数据类型: | 双精度、数组                                                                            |
| 语法:             | <div>MCODS (71,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div> |

表格 7-168: 语法 MCODES (71..)

MCODS (72,...)– 基本坐标的命令值

|     |                                                              |
|-----|--------------------------------------------------------------|
| 描述: | 返回 <Buffer> 中指定通道的 <b>基本坐标命令值</b> （BCS）：首先是所有空间坐标，然后是通道的伪坐标。 |
| 更新: | 循环运行                                                         |



|                                 |                                                           |
|---------------------------------|-----------------------------------------------------------|
| ⟨Buffer⟩ 的数据类型:                 | 实数、数组                                                     |
| 语法:                             | <div>MCODS (72,⟨Channel⟩,⟨Version⟩,⟨Buffer⟩,⟨Size⟩)</div> |
| 表格 7-169:      语法 MCODES (72..) |                                                           |

MCODS (73,...)- 在 ACS1 中的轴坐标

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| 描述:                             | 在 ⟨Buffer⟩ 中, 提供指定通道的 <b>轴坐标值</b> (ACS1):<br>通道 = -1: 所有轴坐标<br>0 < 通道 ≤ 最大通道: 指定通道的数据 |
| 更新:                             | 循环运行                                                                                  |
| ⟨Buffer⟩ 的数据类型:                 | 实数、数组                                                                                 |
| 语法:                             | <div>MCODS (73,⟨Channel⟩,⟨Version⟩,⟨Buffer⟩,⟨Size⟩)</div>                             |
| 表格 7-170:      语法 MCODES (73..) |                                                                                       |

MCODS (74,...)- 加工坐标

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| 描述:                             | 在 ⟨Buffer⟩ 中, 提供指定通道的 <b>加工坐标值</b> (MCS):<br>通道 = -1: 所有轴坐标<br>0 < 通道 ≤ 最大通道: 指定通道的数据 |
| 更新:                             | 循环运行                                                                                  |
| ⟨Buffer⟩ 的数据类型:                 | 实数、数组                                                                                 |
| 语法:                             | <div>MCODS (74,⟨Channel⟩,⟨Version⟩,⟨Buffer⟩,⟨Size⟩)</div>                             |
| 表格 7-171:      语法 MCODES (74..) |                                                                                       |

MCODS (75,...)- 基本坐标的实际值

|                                 |                                                                |
|---------------------------------|----------------------------------------------------------------|
| 描述:                             | 返回 ⟨Buffer⟩ 中指定通道的 <b>基本坐标实际值</b> (BCS): 首先是所有空间坐标, 然后是通道的伪坐标。 |
| 更新:                             | 循环运行                                                           |
| ⟨Buffer⟩ 的数据类型:                 | 实数、数组                                                          |
| 语法:                             | <div>MCODS (75,⟨Channel⟩,⟨Version⟩,⟨Buffer⟩,⟨Size⟩)</div>      |
| 表格 7-172:      语法 MCODES (75..) |                                                                |

MCODS (76,...)- 编程的坐标终点

|                                 |                                                           |
|---------------------------------|-----------------------------------------------------------|
| 描述:                             | 返回 ⟨Buffer⟩ 中指定通道的 <b>编程的坐标终点</b> : 首先是所有空间坐标, 然后是通道的伪坐标。 |
| 更新:                             | 每个块切换后                                                    |
| ⟨Buffer⟩ 的数据类型:                 | 实数、数组                                                     |
| 语法:                             | <div>MCODS (76,⟨Channel⟩,⟨Version⟩,⟨Buffer⟩,⟨Size⟩)</div> |
| 表格 7-173:      语法 MCODES (76..) |                                                           |

MCODS (77,...)- 坐标终点

|                 |                                                          |
|-----------------|----------------------------------------------------------|
| 描述:             | 返回包括⟨Buffer⟩中偏置指定通道的 <b>坐标终点</b> : 首先是所有空间坐标, 然后是通道的伪坐标。 |
| 更新:             | 每个块切换后                                                   |
| ⟨Buffer⟩ 的数据类型: | 实数、数组                                                    |

CPL 命令

语法:

MCODS(77,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-174: 语法 MCODES(77..)

MCODS(78,...)– 坐标名称

描述:

返回 <Buffer> 中指定通道的 **激活坐标名称**：首先是所有空间坐标，然后是通道的伪坐标。  
  
<Buffer> 的大小在 <Size> 中指定。对于 8 个轴，最多可以是 (8\*9) 72 个字节。

更新:

改变后

<Buffer> 的数据类型:

字符、数组

语法:

MCODS(78,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-175: 语法 MCODES(78..)

MCODS(79,...)– 用于坐标的 INPOS 状态

描述:

返回 <Buffer> 中指定通道的 **坐标 INPOS 状态**：首先是所有空间坐标，然后是通道的伪坐标。  
  
轴信号的 AND 操作状态是针对空间坐标生成的。

更新:

循环运行

<Buffer> 的数据类型:

整数、数组

语法:

MCODS(79,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-176: 语法 MCODES(79..)

MCODS(80,...)– 坐标参考

描述:

在 <Buffer> 中，返回指定通道的 **坐标的参考点状态**：  
  
轴信号的 AND 操作状态是针对空间坐标生成的。

更新:

改变后

<Buffer> 的数据类型:

整数、数组

语法:

MCODS(80,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-177: 语法 MCODES(80..)

MCODS(81,...)– 用于坐标的测量单位

描述:

返回 <Buffer> 中指定通道的 **坐标测量单位**：首先是所有空间坐标，然后是通道的伪坐标。  
  
**单位可能的返回值:**

- 0: 英寸
- 1: 公制
- 2: 度
- 3: 坐标不存在

更新:

改变后

<Buffer> 的数据类型:

整数、数组

语法:

MCODS(81,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-178: 语法 MCODES(81..)

MCODS (82, ...) – 坐标数

|                 |                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中, 提供指定通道的 3 个元素中的 <u>坐标/轴数目</u> :                                                                        |
|                 | <ul style="list-style-type: none"><li>1. 值: 通道中轴的总数。</li><li>2. 值: 空间坐标数 + 通道的伪坐标。</li><li>3. 值: 通道中的伪坐标数。</li></ul> |
| 更新:             | 改变后                                                                                                                  |
| <Buffer> 的数据类型: | 整数、数组                                                                                                                |
| 语法:             | <div>MCODS (82, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div>                                |

表格 7-179: 语法 MCODES (82..)

MCODS (83, ...) – 工件坐标系統中坐标剩余距离 (WCS)

|                 |                                                                                       |
|-----------------|---------------------------------------------------------------------------------------|
| 描述:             | 返回 <Buffer> 中指定通道的 <u>工件剩余距离</u> : 首先是所有空间坐标, 然后是通道的伪坐标。                              |
| 更新:             | 循环运行                                                                                  |
| <Buffer> 的数据类型: | 实数、数组                                                                                 |
| 语法:             | <div>MCODS (83, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div> |

表格 7-180: 语法 MCODES (83..)

MCODS (84, ...)

描述: 不适用于 MTX

MCODS (85, ...) – 轴坐标系統中轴的剩余距离

|                 |                                                                                       |
|-----------------|---------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中, 提供指定通道的 <u>轴坐标系統中轴的剩余距离</u> (ACS)。                                      |
| 更新:             | 循环运行                                                                                  |
| <Buffer> 的数据类型: | 实数、数组                                                                                 |
| 语法:             | <div>MCODS (85, &lt;Channel&gt;, &lt;Version&gt;, &lt;Buffer&gt;, &lt;Size&gt;)</div> |

表格 7-181: 语法 MCODES (85..)

MCODS (86, ...)

描述: 不适用于 MTX

MCODS (87, ...) – 频道等待状态

|     |                                                                                                         |
|-----|---------------------------------------------------------------------------------------------------------|
| 描述: | 在<Buffer>中提供 <u>通道</u> 的 <u>等待状态</u> 。<br>该命令指示通道进入等待状态的原因。<br>激活等待状态是位编码的。<br>以下常量定义从最低值开始的第一个整数值的对应位: |
| 0:  | 停留时间                                                                                                    |
| 1:  | 服从确认的辅助函数                                                                                               |
| 2:  | 块传送抑制                                                                                                   |
| 3:  | 通道中的进给等于 0                                                                                              |
| 4:  | 使用 M0/M1 停止程序                                                                                           |
| 5:  | 通道中的进给暂停                                                                                                |
| 6:  | 通道或通道轴中的进给禁止                                                                                            |
| 7:  | 由客户指定的块转换禁止                                                                                             |
| 8:  | 通道间的同步运动停止 (ASTOP, ...)                                                                                 |

CPL 命令

- 9: 在轴交换时等待轴（WAX）
- 10: 等待永久变量（WPV）
- 11: 等待激活时间点的接口信号（WAITA, ...）
- 12: 在块准备时，等待接口信号（WAIT(BITIF...)）或固定区间（WAIT(, ZEIT%)）
- 13: 运动控制数据服务（MCODS(...)）
- 14: 等待通道轴的 inpos 消息
- 15: G95: 等待主轴和旋转轴速度
- 16: G33: 等待主轴速度
- 17: 无返回工件加工位置速度例如点动速度
- 18: 等待主轴运动
- 19: 可编程的事件 (SEV/WEV)
- 20: 等待内部资源的份额（内存）
- 21: G63: 等待 C-轴切换
- 参考 第 4 章“编程示例”第 483 页 中的示例。

更新: 改变后

<Buffer> 的数据类型: 整数

语法:

MCODS (87, <Channel>, <Version>, <Buffer>, 1)

表格 7-182: 语法 MCODES (87..)

MCODS (88, ...)

描述: 不适用于 MTX

MCODS (89, ...) – 在线修正值 (WCS)

描述: 在 <Buffer> 中，返回指定通道的在线修正（WCS）值。首先是所有的空间坐标，然后是通道的伪坐标。

更新: 改变后

<Buffer> 的数据类型: 实数、数组

语法:

MCODS (89, <Channel>, <Version>, <Buffer>, <Size>)

表格 7-183: 语法 MCODES (89..)

MCODS (90, ...) – 在线修正状态 (WCS)

描述: 在 <Buffer> 中，返回指定通道的在线修正当前状态（WCS）。首先是所有的空间坐标，然后是通道的伪坐标。

0: 没有激活

1: 激活

更新: 改变后

<Buffer> 的数据类型: 整数、数组

语法:

MCODS (90, <Channel>, <Version>, <Buffer>, <Size>)

表格 7-184: 语法 MCODES (90..)

MCODS (91, ...) – 直径编程

描述: 对于直径编程的轴，返回 1 <Buffer>。

如果为通道指定了值 Ncs\_MCoNoChannel\_C (-1) ，则会返回所有系统轴的值。

如果指定了通道号，则会返回通道轴的值。

更新: 每个块切换后  
<Buffer> 的数据类型: 整数、数组

语法: 

MCODS (91, <Channel>, <Version>, <Buffer>, <Size>)

表格 7-185: 语法 MCODES (91..)

MCODS (92, ...) – 刀尖的基本坐标 (BCS-Tcp)

描述: 在 <Buffer> 中激活 3D 半径修正的情况下，返回基本坐标系 (BCS Tcp) 中刀尖的指令值。首先是空间坐标，然后是指定通道的伪坐标。  
如果 3D 半径校正关闭，则这些值与 MCODES (72, ...) 相同。  
更新: 循环运行  
<Buffer> 的数据类型: 实数、数组

语法: 

MCODS (92, <Channel>, <Version>, <Buffer>, <Size>)

表格 7-186: 语法 MCODES (92..)

MCODS (93, ...)

描述: 不适用于 MTX

MCODS (94, ...) – 辅助功能的组号

描述: <GroupId> 中提供在指定通道的 <Syntax> 中指定的辅助功能的 辅助功能组号。  
<Syntax> 包含 辅助功能 ，它们由空格隔开并以 CHR\$(0) 结束，请参阅 “示例: ” 第 473 页 。  
更新: 循环运行  
<Buffer> 的数据类型: 字符、数组

语法: 

MCODS (94, <Channel>, <Version>, <GroupId>, <Size>, <Syntax>)

表格 7-187: 语法 MCODES (94..)

MCODS (95, ...) – 激活辅助函数数组的语法

描述: 在 <Syntax> 中为指定通道的 <GroupId> 中指定的 辅助功能 组编号提供激活语法。  
在 <GroupId> 中传输一个数组（由 CHR \$ (0) 完成）所需的 辅助功能 组号；请参考下面的示例。  
更新: 循环运行  
<Buffer> 的数据类型: 字符、数组

语法: 

MCODS (95, <Channel>, <Version>, <Syntax>, <Size>, <GroupId>)

表格 7-188: 语法 MCODES (95..)

MCODS (96, ...)

描述: 不适用于 MTX

MCODS (97, ...)

描述: 不适用于 MTX

MCODS (98, ...)

描述: 不适用于 MTX

CPL 命令

MCODS (99,...) – 激活零点偏置数据

**描述:** 在 <Buffer> 中, 返回机床坐标的零点偏置的激活数据块。

对于每个坐标, 可为 Z0 库 1..5 的激活偏置值提供最多五个”双精度”值。如果没有偏置是激活的, 则返回 0.0。

按以下将数据存储到 <Buffer>:

- 偏置 1. 坐标, 库 1
- 偏置 1. 坐标, 库 2
- ...
- 偏置 1. 坐标, 库 5
- 第 n 坐标的偏置, 库 1
- ...
- 第 n 坐标的偏置, 库 5

**更新:** 改变后

**<Buffer> 的数据类型:** 双精度、数组

**语法:**

MCODS (99,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-189: 语法 MCODES (99..)

MCODS (100,...) – 零点偏置的总和

**描述:** 返回 <Buffer> 中机床坐标的有效零点偏置的总和。对于每个坐标, 都提供了一个“双精度”值。

如果没有偏置是激活的, 则返回 0.0。

**更新:** 改变后

**<Buffer> 的数据类型:** 双精度、数组

**语法:**

MCODS (100,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-190: 语法 MCODES (100..)

MCODS (101,...) – 轴坐标终点

**描述:** 在 <Buffer> 中, 提供指定通道的 **轴坐标的终点** (ACS):

0 < 通道 ≤ 最大通道: 指定通道的数据

**更新:** 每个块切换后

**<Buffer> 的数据类型:** 实数、数组

**语法:**

MCODS (101,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-191: 语法 MCODES (101..)

MCODS (102,...) – 主轴名称

**描述:** 在 <Buffer> 中, 返回指定通道的激活主轴名称。如果为 <Channel> 提供 -1 或 0, 则 <Buffer> 提供系统主轴的数据。否则, 提供通道主轴数据。

**更新:** 改变后

**<Buffer> 的数据类型:** Char, 数组

**语法:**

MCODS (102,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-192: 语法 MCODES (102..)

MCODS (103,...)- 主轴数

**描述:** 在 <Buffer> 中, 返回指定通道的主轴号。如果为 <Channel> 提供 -1 或 0, 则 <Buffer> 提供系统主轴的数据。否则, 提供通道主轴数据。

**更新:** 改变后

**<Buffer> 的数据类型:** 整数

**语法:**

```
MCODS (103,<Channel>,<Version>,<Buffer>,<Size>)
```

表格 7-193: 语法 MCODES (103..)

MCODS (104,...)- 主轴系统编号

**描述:** 在 <Buffer> 中, 返回主轴的系统编号 (系统主轴编号), 它们以升序排列 (S 或 S1、S2、S3 等)。如果为 <Channel> 提供 -1 或 0, 则 <Buffer> 提供系统主轴的数据。否则, 提供通道主轴数据。

**更新:** 改变后

**<Buffer> 的数据类型:** 整数、数组

**语法:**

```
MCODS (104,<Channel>,<Version>,<Buffer>,<Size>)
```

表格 7-194: 语法 MCODES (104..)

MCODS (105,...)- 主轴轴编号

**描述:** 在 <Buffer> 中, 返回主轴的轴编号, 它们以升序排列 (S 或 S1、S2、S3 等)。如果为 <Channel> 提供 -1 或 0, 则 <Buffer> 提供系统主轴的数据。否则, 提供通道主轴数据。

**更新:** 改变后

**<Buffer> 的数据类型:** 整数、数组

**语法:**

```
MCODS (105,<Channel>,<Version>,<Buffer>,<Size>)
```

表格 7-195: 语法 MCODES (105..)

MCODS (106,...)- 机床坐标名称

**描述:** 返回 <Buffer> 中指定通道激活机床的名称: 首先是所有空间坐标, 然后是机床坐标系中通道的伪坐标。

<Buffer> 的大小在 <Size> 中指定。对于 8 个轴, 最多可以是 (8\*9) 72 个字节。

**更新:** 改变后

**<Buffer> 的数据类型:** 字符、数组

**语法:**

```
MCODS (106,<Channel>,<Version>,<Buffer>,<Size>)
```

表格 7-196: 语法 MCODES (106..)

MCODS (107,...)- 机床坐标数

**描述:** 在 <Buffer> 中, 返回指定通道的 3 个元素中的坐标/轴数:

- 1. 值: 通道中轴的总数。
- 2. 值: 空间坐标数加上机床坐标系中通道的伪坐标数。
- 3. 值: 机床坐标系中通道的伪坐标数

CPL 命令

|                            |                                                                                   |
|----------------------------|-----------------------------------------------------------------------------------|
| 更新:                        | 改变后                                                                               |
| <Buffer> 的数据类型:            | 整数、数组                                                                             |
| 语法:                        | <div>MCODS(107,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div> |
| 表格 7-197: 语法 MCODES(107..) |                                                                                   |

MCODS(108,...)- 实际主轴位置

|                            |                                                                                                     |
|----------------------------|-----------------------------------------------------------------------------------------------------|
| 描述:                        | 在 <Buffer> 中，返回指定通道的激活主轴实际位置。<br>如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的实际位置。<br>否则，提供通道主轴的实际位置。 |
| 更新:                        | 循环运行                                                                                                |
| <Buffer> 的数据类型:            | 实数、数组                                                                                               |
| 语法:                        | <div>MCODS(108,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div>                   |
| 表格 7-198: 语法 MCODES(108..) |                                                                                                     |

MCODS(109,...)- 在 ACS0 中的轴坐标

|                            |                                                                                      |
|----------------------------|--------------------------------------------------------------------------------------|
| 描述:                        | 在 <Buffer> 中，提供指定通道的 <b>轴坐标值</b> (ACS0):<br>通道 = -1: 所有轴坐标<br>0 < 通道 ≤ 最大通道: 指定通道的数据 |
| 更新:                        | 循环运行                                                                                 |
| <Buffer> 的数据类型:            | 实数、数组                                                                                |
| 语法:                        | <div>MCODS(109,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div>    |
| 表格 7-199: 语法 MCODES(109..) |                                                                                      |

MCODS(110,...)- 实际主轴扭矩

|                            |                                                                                                                                                                             |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:                        | 在 <Buffer> 中，返回指定通道的激活主轴的最小实际扭矩（百分比 [%]）。<br>如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的扭矩。否则，提供通道主轴的扭矩。<br>为每个主轴传递两个值。<br>也可参阅文档 “Rexroth IndraMotion MTX 标准数控操作”，“技术显示”章节。 |
| 更新:                        | 循环运行                                                                                                                                                                        |
| <Buffer> 的数据类型:            | 双精度、数组                                                                                                                                                                      |
| 语法:                        | <div>MCODS(110,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div>                                                                                           |
| 表格 7-200: 语法 MCODES(110..) |                                                                                                                                                                             |

MCODS(111,...)- 主轴性能

|     |                                                                                                                                                                              |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述: | 在 <Buffer> 中，返回指定通道的激活主轴的最大实际性能（单位，瓦 [W]）。<br>如果为 <Channel> 提供 -1 或 0，则 <Buffer> 提供系统主轴的性能。否则，提供通道主轴的性能。<br>为每个主轴传递两个值。<br>也可参阅文档 “Rexroth IndraMotion MTX 标准数控操作”，“技术显示”章节。 |
| 更新: | 循环运行                                                                                                                                                                         |



|                              |                                                           |
|------------------------------|-----------------------------------------------------------|
| ⟨Buffer⟩ 的数据类型:              | 整数、数组                                                     |
| 语法:                          | <div>MCODS(111,⟨Channel⟩,⟨Version⟩,⟨Buffer⟩,⟨Size⟩)</div> |
| 表格 7-201: 语法 MCODES(111,...) |                                                           |

MCODS(112,...)- 激活定位数据

|                            |                                                                                                                                                                          |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:                        | 返回 ⟨Buffer⟩ 中定位的激活数据块。为每个级别或库提供六个“双精度”值。如果没有激活定位，则返回 0.0。总共提供了 30 个“双精度”值。<br>按以下将数据存储到 ⟨Buffer⟩:                                                                        |
|                            | <ul style="list-style-type: none"><li>库 1 [X, Y, Z, Phi, Theta, Psi]</li><li>库 2 [X, Y, Z, Phi, Theta, Psi]</li><li>...</li><li>库 5 [X, Y, Z, Phi, Theta, Psi]</li></ul> |
| 更新:                        | 改变后                                                                                                                                                                      |
| ⟨Buffer⟩ 的数据类型:            | 双精度、数组                                                                                                                                                                   |
| 语法:                        | <div>MCODS(112,⟨Channel⟩,⟨Version⟩,⟨Buffer⟩,⟨Size⟩)</div>                                                                                                                |
| 表格 7-202: 语法 MCODES(112..) |                                                                                                                                                                          |

MCODS(113,...)- 激活定位表格的路径和名称

|                            |                                                                |
|----------------------------|----------------------------------------------------------------|
| 描述:                        | 在 ⟨Buffer⟩ 中，提供在 ⟨Channel⟩ 激活的定位表格名称。<br>如果未激活，则返回 3 个空格作为字符串。 |
| 更新:                        | 每个块切换后                                                         |
| ⟨Buffer⟩ 的数据类型:            | 字符、数组                                                          |
| 语法:                        | <div>MCODS(113,⟨Channel⟩,⟨Version⟩,⟨Buffer⟩,⟨Size⟩)</div>      |
| 表格 7-203: 语法 MCODES(113..) |                                                                |

MCODS(114,...)- 工件位置修正

|                            |                                                                                                                                                                            |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:                        | 在 ⟨Buffer⟩ 中，返回 ⟨Channel⟩ 中激活的工件位置位置。<br>提供六个”双精度”值：<br>[X <sub>w</sub> offset, Y <sub>w</sub> offset, Z <sub>w</sub> offset, Phi, Theta, Psi].<br>如果没有激活工件位置修正，则始终返回 0.0。 |
| 更新:                        | 改变后                                                                                                                                                                        |
| ⟨Buffer⟩ 的数据类型:            | 双精度、数组                                                                                                                                                                     |
| 语法:                        | <div>MCODS(114,⟨Channel⟩,⟨Version⟩,⟨Buffer⟩,⟨Size⟩)</div>                                                                                                                  |
| 表格 7-204: 语法 MCODES(114..) |                                                                                                                                                                            |

MCODS(115,...)- 定位总数

|                 |                                                                                                                                                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 返回 ⟨Channel⟩ 中激活的定位总数和 ⟨Buffer⟩ 中的工件位置修正。<br>提供六个”双精度”值：<br>[X <sub>w</sub> offset, Y <sub>w</sub> offset, Z <sub>w</sub> offset, Phi, Theta, Psi].<br>如果没有激活工件位置修正及定位，返回 0.0。 |
| 更新:             | 改变后                                                                                                                                                                            |
| ⟨Buffer⟩ 的数据类型: | 双精度、数组                                                                                                                                                                         |

CPL 命令

语法:

MCODS(115,Channel>,<Version>,<Buffer>,<Size>)

表格 7-205: 语法 MCODES(115..)

MCODS(116,...)- 激活程序坐标偏置

描述: 返回 <Buffer> 中的 <Channel> 中激活的程序坐标偏置。  
为每个坐标提供“双精度”值。  
如果没有程序坐标偏置被激活，则返回 0.0。

更新: 改变后

<Buffer> 的数据类型: 双精度、数组

语法: MCODES(116,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-206: 语法 MCODES(116..)

MCODS(117,...)- 激活累加程序坐标偏置

描述: 返回 <Buffer> 中的 <Channel> 中激活累加程序坐标偏置。  
为每个坐标提供“双精度”值。  
如果没有程序坐标偏置被激活，则返回 0.0。

更新: 改变后

<Buffer> 的数据类型: 双精度、数组

语法: MCODES(117,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-207: 语法 MCODES(117..)

MCODS(118,...)- 激活程序坐标偏置总和

描述: 返回 <Buffer> 中的 <Channel> 中激活的程序坐标偏置总和。  
为每个坐标提供“双精度”值。  
如果没有程序坐标偏置被激活，则返回 0.0。

更新: 改变后

<Buffer> 的数据类型: 双精度、数组

语法: MCODES(118,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-208: 语法 MCODES(118..)

MCODS(119,...)- 当前轴速度

描述: 在 <Buffer> 中，提供指定通道的 当前轴速度 。  
如果 <channel> 指定为 -1，则将提供所有系统轴的轴速率。  
否则，该函数提供指定通道的当前轴速率。

更新: 循环运行

<Buffer> 的数据类型: 双精度、数组

语法: MCODES(119,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-209: 语法 MCODES(119..)

MCODS(120,...)- 当前轴速度

描述: 在 <Buffer> 中，提供所有系统轴的 当前轴速度 。  
轴速度只用于异步、循环回转轴。

更新: 每个块切换后

<Buffer> 的数据类型: 双精度、数组

语法:

MCODS(120,-1,<Version>,<Puffer>,<Size>)

表格 7-210:      语法 MCODS(120..)

MCODS(121,...)- 已编程的轴速度

描述:      在 <Buffer> 中, 提供所有系统轴的 **编程轴速度** 。  
轴速度只用于异步、循环回转轴。

更新:      每个块切换后

<Buffer> 的数据类型:      双精度、数组

语法:      MCODS(121,-1,<Version>,<Buffer>,<Size>)

表格 7-211:      语法 MCODS(121..)

MCODS(122,...)- 旋转角度

描述:      在 <Buffer> 中, 提供在 <Channel> 激活平面的 **旋转角** 。  
提供”双精度”值。  
如果无旋转角被激活, 则返回 0.0。

更新:      改变后

<Buffer> 的数据类型:      Double

语法:      MCODS(122,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-212:      语法 MCODS(122..)

MCODS(123,...)- 镜像和旋转的极位置

描述:      在 <Buffer> 中的 <Channel> 中提供 **已编程的极点数据** 。  
为每个指定的坐标提供“双精度”值。  
如果未编程极点数据, 则返回 0.0。

更新:      改变后

<Buffer> 的数据类型:      双精度、数组

语法:      MCODS(123,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-213:      语法 MCODS(123..)

MCODS(124,...)- 激活偏置坐标

描述:      在 <Buffer> 中的 <Channel> 中提供 **激活偏置坐标** 。  
为每个坐标提供“双精度”值。  
如果无偏置坐标被激活, 则返回 0.0。

更新:      改变后

<Buffer> 的数据类型:      双精度、数组

语法:      MCODS(124,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-214:      语法 MCODS(124..)

MCODS(125,...)- 激活缩放系数

描述:      在 <Buffer> 中的 <Channel> 中提供 **激活缩放系数** 。  
将双精度型值提供给每个系数。  
如果无缩放系数被激活, 则返回 1.0。

更新:      改变后

<Buffer> 的数据类型:      双精度、数组

CPL 命令

语法:

MCODS(125,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-215: 语法 MCODES(125..)

MCODS(126,...)- 激活坐标镜像

描述:

在”缓冲区”中的 <Channel> 中提供 **激活坐标镜像** 。  
将整数值提供给每个坐标。  
如果没有激活镜像，则返回 0。否则，返回 1。

更新:

改变后

<Buffer> 的数据类型:

整数、数组

语法:

MCODS(126,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-216: 语法 MCODES(126..)

MCODS(127,...)- 块预运行再入口标记的状态

描述:

提供 <Buffer> 中 **块预运行重新标记** 中存在的轴状态。  
如果生成或更改了块预运行重新标记，则返回 1。如果它被删除，则返回 0。

更新:

改变后

<Buffer> 的数据类型:

整数

语法:

MCODS(127,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-217: 语法 MCODES(127..)

MCODS(128,...)- 块预运行调整状态

描述:

在 <Buffer> 中提供当前执行的 **块预运行调整** 状态。  
返回以下调整状态:

- 0:
- 块预运行未激活
- 5:
- 计算激活
- 10:
- 找到目标块; 等待 NC 启动
- 20:
- 初始化调整程序激活
- 30:
- 手动数据输入有效 (块预运行后重写)
- 40:
- 激活作用块
- 60:
- 目标块调整程序激活
- 80:
- 接近动作激活

更新:

改变后

<Buffer> 的数据类型:

整数

语法:

MCODS(128,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-218: 语法 MCODES(128..)

MCODS(129,...)- 通道复位

描述:

等待 <Channel> 中执行的 **通道复位** 。初始数据无法读取。这意味着，该命令仅在指定通道的下一次重置时返回。  
重置通道的通道号在 <Buffer> 中返回。

更新:

改变后

<Buffer> 的数据类型:

整数

语法:

MCODS(129,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-219: 语法 MCODES(129..)

MCODS(131,...)- 最大轴速度

描述: 在 <Buffer> 中, 提供指定通道的 **最大轴速度**。  
如果 <channel> 指定为 -1, 则将提供所有系统轴的轴速率。  
否则, 该命令提供指定通道的轴速率。  
最大轴速度仅适用于点动模式和异步轴的轴

更新: 循环运行

<Buffer> 的数据类型: 双精度、数组

语法:

MCODS(131,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-220: 语法 MCODES(131..)

MCODS(132,...)- 当前温度补偿

描述: 在 <Buffer> 中提供 **当前的温度补偿**  
如果 <Channel> 指定为 -1, 则将提供所有系统轴的温度补偿。  
否则, 该命令提供指定通道的温度补偿。

更新: 循环运行

<Buffer> 的数据类型: 双精度、数组

语法:

MCODS(132,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-221: 语法 MCODES(132..)

MCODS(133,...)- 电前丝杠误差补偿

描述: 在 <Buffer> 中提供 **当前的丝杠补偿**  
如果 <Channel> 指定为 -1, 则将提供所有系统轴的丝杠误差补偿。  
否则, 该命令提供指定通道的丝杠误差补偿。

更新: 循环运行

<Buffer> 的数据类型: 双精度、数组

语法:

MCODS(133,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-222: 语法 MCODES(133..)

MCODS(134,...)- 当前交叉补偿

描述: 在 <Buffer> 中提供 **当前的交叉补偿**  
如果 <Channel> 指定为 -1, 则将提供所有系统轴的交叉补偿。  
否则, 该命令提供指定通道的交叉补偿。

更新: 循环运行

<Buffer> 的数据类型: 双精度、数组

语法:

MCODS(134,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-223: 语法 MCODES(134..)

MCODS(135,...)- 当前补偿偏置

描述: 在 <Buffer> 中提供 **当前补偿偏置**  
对于补偿偏置, 它是激活的有效补偿总和。  
如果 <Channel> 指定为 -1, 则将提供系统轴的偏置。

CPL 命令

否则，该命令提供指定通道的轴偏置。

更新:循环运行

<Buffer> 的数据类型:双精度、数组

语法:

MCODS(135,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-224: 语法 MCODES(135..)

MCODS(142,...) 编程的路径进给

描述:某些数控功能可以使用两个输入进行编程，例如 G94。MCODS(142,...) 可用于确定各个进给。

源从通道 <Channel> 中读取，并返回到 REAL 数组 <Buffer> 中。<Size> 提供数组中元素的数量。

MCODS(143,...) 提供了当前适用的编程馈送索引。MCODS(24,...) 总是仅提供第一个编程的进给，即，也提供有 MCODES(142,...) 的第一个元件。

更新:每个块切换后

<Buffer> 的数据类型:实数、数组

语法:

MCODS(142,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-225: 语法 MCODES(142..)

MCODS(143,...) – 激活编程的路径进给

描述:该命令提供当前激活的编程路径进给索引。该路径进给数组可以用 MCODES(142,...) 确定。由于可通过 PLC 接口信号切换激活的编程进给，因此即使在单个程序段中，编程的路径进给的索引也可以改变（多次）。

更新:改变后

<Buffer> 的数据类型:整数

语法:

MCODS(143,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-226: 语法 MCODES(143..)

MCODS(145,...) – 轴速度的峰值和最大值

描述:返回 <Buffer> 中指定通道的当前轴速度的峰值，最大值和标称值。如果为 <Channel> 指定了 -1，则会提供所有系统轴的轴速度值。否则，该命令将提供指定通道的当前轴速度值。为每个轴传输由峰值，最大值和标称值组成的三个值。将峰值和最大值除以标称值以确定峰值和最大值的百分比值。

 如果通过系统数据输入（/SysNcDgnCtrl/AxesLoadDgn/VelLoadDgn/DgnMode = 2,.../AccLoadDgn /DgnMode = 2）激活这些诊断，则控制器将首先提供速度和加速度轴负载诊断的值。

 如果不再需要这些诊断信息，则使用系统数据输入 (... /VelLoadDgn/DgnMode = 0,.../AccLoadDgn/DgnMode = 0) 关闭速度和加速度负载诊断！这阻碍插补器中断的不必要负载。

更新:循环运行

<Buffer> 的数据类型:实数、数组

语法:

MCODS(145,<Channel>,<Version>,<Buffer>,<Size>)

表格 7-227: 语法 MCODES(145..)

MCODS (146, ...) – 轴加速度的峰值和最大值

|                 |                                                                                                                                                                                                                       |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 返回<Buffer>中指定通道的当前轴加速度的峰值，最大值和标称值。如果为<Channel>指定了-1，则会提供所有系统轴的轴加速度值。否则，该命令将提供指定通道的当前轴加速度值。为每个轴传输由峰值，最大值和标称值组成的三个值。将峰值和最大值除以标称值以确定峰值和最大值的百分比值。                                                                         |
|                 | <div> 如果通过系统数据输入（ /SysNcDgnCtrl/AxesLoadDgn/VelLoadDgn/DgnMode = 2, .../AccLoadDgn /DgnMode = 2）激活这些诊断，则控制器将首先提供速度和加速度轴负载诊断的值。</div> |
|                 | <div> 如果不再需要这些诊断信息，则使用系统数据输入（... /VelLoadDgn/DgnMode = 0, .../AccLoadDgn/DgnMode = 0）关闭速度和加速度负载诊断！这阻碍插补器中断的不必要负载。</div>              |
| 更新:             | 循环运行                                                                                                                                                                                                                  |
| <Buffer> 的数据类型: | 实数、数组                                                                                                                                                                                                                 |
| 语法:             | <div>MCODS (146,&lt;Channel&gt;,&lt;Version&gt;,&lt;Buffer&gt;,&lt;Size&gt;)</div>                                                                                                                                    |

表格 7-228: 语法 MCODES (146,...)

MCODS (147, ...) – CNC 的实际轴位置

|                 |                                                                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 描述:             | 在 <Buffer> 中以升序和通道无关的顺序提供 系统中所有进给轴和辅助轴 的实际位置: <ul style="list-style-type: none"><li>- 对于直线轴，单位为毫米</li><li>- 旋转轴，单位角度</li></ul> 这些值 不 包括“设置实际值”（如 SetPos）。            |
|                 | <div> 由 MCODES (147 ..) 提供的值与 MCODES (35 ..) 的不同之处在于轴错误更正 LSEC、CCOMP 和温度补偿。</div> |
| 更新:             | 循环运行                                                                                                                                                                 |
| <Buffer> 的数据类型: | 实数、数组                                                                                                                                                                |
| 语法:             | <div>MCODS (147,-1,,&lt;Version&gt;,&lt;Puffer&gt;,&lt;Size&gt;)</div>                                                                                               |

表格 7-229: 语法 MCODES (147..)

MCODS 编程示例

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 示例: | <div>辅助功能</div> <div><pre>: N10 M3 S1234 T5678 WAIT 10 DIM GROUPID\$(64) 15 DIM SYNTAX\$(256) 17 ERSION=0 18 CHAN%=1 20 REM find group numbers of M3 and T 25 SYNTAX\$="T M3" +CHR\$ 30 ERR_VAR%=MCODS (94,CHAN%,VERSION,GROUPID\$,64,SYNTAX\$) 31 PRN#(0,"94 Error:", ERR_VAR%) 32 PRN#(0,"T M3:",ACS(MID\$(GROUPID\$,1,1),     ACS(MID\$(GROUPID\$,2,1))) 33 REM read groups 2 and 1 36 GROUPID\$=CHR\$(2)+CHR\$(1)+CHR\$(0): 37 REM read groups 2 and 1 50 VERSION=0 58 CHAN%=1 60 ERR_VAR%=MCODS (95,CHAN%,VERSION,SYNTAX\$,256,GROUPID\$) 65 PRN#(0,"95 Error:", ERR_VAR%)</pre></div> |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

CPL 命令

```
70 RN#(0,"Group 2 1 : ", SYNTAX$)
71 REM SYNTAX$ contains e.g. "S1234 M3"
```

MCODS（一般零件）

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

**描述:** **运动控制处理服务** – 运动控制处理服务  
通过 CPL 调用 NCS 的运动控制处理服务。由此可对 CNC 中的通道进行控制。

|               |                                                                   |
|---------------|-------------------------------------------------------------------|
| <b>语法:</b>    | MCOPS (<Fct>,<Channel>[[ , [<P1>] [ , [<P2>] , [<P3>] ] ] ,<P4>]) |
| <Fct>         | 整数型表达式<br>指出要执行的函数。下表描述所有可用函数。                                    |
| <Channel>     | 整数型表达式<br>指出受函数支配的通道。                                             |
| <P1> ... <P4> | 取决于 <Fct> 的可选参数。逗号序列是允许的。但是，在右括号之前的逗号是不允许的。                       |

表格 7-230: MCOPS 语法

 在下表中，语法中指定的一些参数是整数常量。除了这些常量，整数变量也可以被编程。但是，它们在命令被调用时的指定值必须分配给那个。

MCOPS(1,...)– 取消剩余距离

**描述:** **运动控制流程服务 (1,...)** – 运动控制流程服务 (1,...)：取消剩余距离  
调用 NCS 的运动控制处理服务 “取消剩余距离”。

- 触发 “取消剩余行程”之后，所有准备好的数控块，包括当前块的余量将取消并再次准备好。

**未考虑 CPL 块或 CPL 零件:**

**示例:**  
CPL 变量 POS 在准备时的值为 10：将数控字 X[POS] 解释为以下 X10 “取消剩余距离” 即使 POS 当时有完全不同的值。

*考虑任何可能的修正值:*

- 在显示中，指示的终点被设置为当前位置，并导致同时删除剩余距离。然后 <Channel> 切换到 NC 就绪状态 (iCh\_NCReady PLC 输入信号)。
- 在 NC 启动之后 (PLC 输出信号 qCh\_NCStart)，考虑到新修正值，程序在中断点继续。

使用 MCOPS 的例子 (1, <Channel>)：修正表更改后，新值也适用于之前准备的块。

|            |                      |
|------------|----------------------|
| <b>语法:</b> | MCOPS(1,<Channel>)   |
| <Channel>  | 整数型表达式<br>指出受函数支配的通道 |

表格 7-231: 语法 MCOPS(1,...)

MCOPS(2,...)– 复位

**描述:** **运动控制流程服务 (2,...)** – 运动控制流程服务 (2,...)：复位



调用 NCS “复位”的运动控制处理服务。

要启动整体复位: <Channel> = -2

- 通道暂时不再接受任何任务，例如程序选择或操作模式切换。
- 插补器已停止。
- 通道的任务 - 未处理命令 - 被取消。
- 主程序被取消选择。
- 修改的不需要启动的机床参数已被接受，例如， 参数 CHAN/Ch/Ini/ChResetState (MP 7060 00020)（软件限位开关）。
- 该启动通道的错误和警告被删除。
- 插补器重启。
- 复位的启动状态（机床参数 7060 00020）被接受，即，对应模态状态变成激活。
- 通道发送接口信号 0.2 “复位已执行”，并再次做好接受新任务的准备。

语法:

| MCOPS (2, <Channel>[, <ResetVariant>]) |                                                                                                                                                                                               |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Channel>                              | 整数型表达式<br>指出受函数支配的通道。<br><Channel> = -2: 总重置启动。                                                                                                                                               |
| <ResetVariant>                         | 整数型表达式<br>指定函数的行为。以下列表包含所有定义的行为模式。每个模式提前都有一个 ID。要设置某种行为，必须在 <Reset variant> 中提供对应 ID。如果要组合多个行为模式，则必须将 ResetVariant 中所有对应 ID 的总数传送给该函数。<br>到目前为止，该列表仅包含一个元素：<br>代码号：<br>2: 如果自动程序重新选择已激活，则被禁用。 |

表格 7-232: 语法 MCOPS (2, ...)

示例:

10 ERR\_VAR=MCOPS (2, 2, 2)  
;在没有自动程序重选的情况下复位第二个通道。

MCOPS3, ...)- 搜索块

描述:

**运动控制流程服务 (3, ...)** - 运动控制流程服务 (1, ...)：搜索块调用 NCS 的运动控制处理服务 “以搜索块”:

- <Starting block> 和 <End block> 作为字符串表达式传递。  
搜索 <Starting block> 和 <End block> 时适用以下约定：
- 处理从 <Starting block> 开始并以 <End block> 结束。  
如果 <Starting block> 缺失或未找到该块，则从程序开始时开始处理。  
如果 <End block> 缺失或未找到该块，则在程序结束时停止处理。
- 数控块状态在 “搜索块” 后更改为 READY。

语法:

| MCOPS (3, <Channel> [, [<StartingBlock>] [, <EndBlock>]]) |                       |
|-----------------------------------------------------------|-----------------------|
| <Channel>                                                 | 整数型表达式<br>指出受函数支配的通道。 |

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Starting block> | 字符串表达式                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <End block>      | <ul style="list-style-type: none"><li>忽略数控块开始处的空格、&lt;Tab&gt;、&lt;LF&gt;。</li><li>如果 &lt;StartingBlock&gt; 或 &lt;EndBlock&gt; 以数字开头，并且如果在搜索程序中未发现表达式，则系统将重启其表达式搜索，但这次前面有字符 “N” 。<br/>由此例如， “50” 还将找到数控块 “N50X100”。</li><li>如果 &lt;Starting block&gt; 或 &lt;Ending block&gt; 以数值结束，则只有在第一个，例如， “G1X10” <b>没有</b> 找到数控块 “G1X100”之后立即添加其他数值时才能找到程序中的表达式。</li><li>如果 &lt;Starting block&gt; 或 &lt;End block&gt; 以一个字母结尾，则只有在空格紧跟后才能找到搜索程序中的表达式。<br/>例如， “50A” 发现 Nc 块 “50A = 1” 但不是 “50A=1”。</li></ul> |

表格 7-233: 语法 MCOPS(3,...)

示例: 10 ERR\_VAR=MCOPS(3,2,"N50","N100")  
;启动通道 2 中的“搜索块”。主程序从 N50 开始处理，直到 N100（含）。

MCOPS(4,...)– 选择程序或手动数据输入

描述: 运动控制流程服务（4，...）– 运动控制流程服务（4，...）：选择程序或手动数据输入  
调用 NCS 的运动控制过程服务来选择程序或手动数据输入：  
在编程的 <Channel> 中，它使用手动数据输入模式选择一个程序或一个字符串进行处理。  
该程序或字符串自动选择，无需链接运行。

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法:       | MCOPS(4,<Channel>[[,<String>][,<StartingBlock>],<EndBlock>]],<SelectionType>)]                                                                                                                                                                                                                                                                                                                                                                                            |
| <Channel> | 整数型表达式<br>指出受函数支配的通道。                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <String>  | 字符串表达式<br>根据 <Selection type>, 系统将参数解释为: <ul style="list-style-type: none"><li>要选择零件程序的路径名称（包括零件程序名称）（最多 100 个字符），或</li><li>如果在 &lt;Selection type&gt; 中指定 32 作为要使用 “手动数据输入” 操作模式处理的 NC 程序段（最大 512 字节，包括终止 0 字节），或者</li><li>如果在 &lt;Selection type&gt; 中指定了 32 + 4096 作为多个待使用 “手动数据输入” 模式处理的 NC 块。<br/>多个 NC 块由 NewLine (“\n”, Hex 0x0A) 分隔。所有 NC 块的最大尺寸不得超过 4096 字节，包括终止 0 字节。</li><li>如果在 &lt;Selection type&gt; 中指定了 32768: 名称或路径名称是针对异步子例程的 “SETINT” 任务提供的。</li></ul> |

## CPL 命令

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Starting block> | 字符串表达式                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <End block>      | <p>&lt;Starting block&gt; 和 &lt;End block&gt; 指定零件程序中的启动和结束块，以进行处理。如果系统将 &lt;String&gt; 解释为手动数据输入，则 &lt;Start block&gt; 和 &lt;End block&gt; 将被忽略。</p> <ul style="list-style-type: none"><li>忽略数控块开始处的空格、&lt;Tab&gt;、&lt;LF&gt;。</li><li>如果 &lt;StartingBlock&gt; 或 &lt;EndBlock&gt; 以数字开头，并且如果在搜索程序中未发现表达式，则系统将重启其表达式搜索，但这次前面有字符“N”。</li></ul> <p>由此例如，“50”还将找到数控块“N50X100”。</p> <ul style="list-style-type: none"><li>如果 &lt;Starting block&gt; 或 &lt;Ending block&gt; 以数值结束，则只有在第一个，例如，“G1X10”<b>没有</b>找到数控块“G1X100”之后立即添加其他数值时才能找到程序中的表达式。</li><li>如果 &lt;Starting block&gt; 或 &lt;End block&gt; 以一个字母结尾，则只有在空格紧跟后才能找到搜索程序中的表达式。</li></ul> <p>例如，“50A”发现 Nc 块“50A = 1”但不是“50A=1”。</p> <p>如果 &lt;Selection type&gt;：异步子程序可以使用规范：= 32768. 他们可以登录，关闭，重新启动，删除和启动。</p> <ul style="list-style-type: none"><li>对于异步子程序（&lt;SelectionType&gt; = 32768）的规定，数字（<math>1 \leq \text{Number} \leq 8</math>）在 &lt;StartBlock&gt; 中以字符串（例如“1”）提供，在该任务中，则在 &lt;EndBlock&gt; 中提供。</li></ul> <p>异步子例程的任务：</p> <p>“SETINT” 登录</p> <p>“DISABLE” 关闭</p> <p>“ENABLE” 再次开启</p> <p>“CLRINT” 删除</p> <p>“START” 启动</p> |
| <Selection type> | <p>整型表达式</p> <p>指定函数的行为。以下列表包含所有定义的行为模式。每个模式提前都有一个 ID。如果要设置某个行为，则必须在 &lt;Selection type&gt; 中提供此 ID。</p> <p>如果要组合多个行为模式，则必须将 &lt;Selection type&gt; 中的所有相应 ID 的总和传送给该命令。</p> <p>如果 &lt;SelectionType&gt; 2（等待 NC 状态切换为 READY）被输入并且要选择的程序不存在或不能执行，则返回错误消息 6。在所有其他情况下，不检查程序是否可以执行。该命令返回 0（没有发生错误）。</p> <p>对应运行时错误仅由后续链接生成。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 选择类型的值：          | 含义：                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 1                | 在选择期间，系统执行链接操作。如果链接不存在，则为选定的主程序生成链接表。程序中存在子程序调用或 CPL 命令时，链接表是必不可少的。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 2                | 当 NC 状态切换到 READY 时，系统确认选择的有效性。通常情况下，选择无需等待 NC 状态准备就绪。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 32               | 系统将 <String> 解释为手动数据输入块。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 64               | 在命令选择指定程序或手动数据输入块之前，取消选择活动程序或手动数据输入块。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

CPL 命令

|        |                                                                                                                                                                                                                                                                                                                                                                           |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 256    | 必须在“点动”模式下进给轴或者在“工件坐标模式下点动”执行运动。                                                                                                                                                                                                                                                                                                                                          |
| 512    | 必须在“接近参考点”模式中启动轴。                                                                                                                                                                                                                                                                                                                                                         |
| 2048   | 一个已经激活的程序被新选择的程序取代。结果，所有的模态状态仍然存在。在手动数据输入中，旧字符序列被新的字符序列替换。                                                                                                                                                                                                                                                                                                                |
| 4096   | 已缓存 NC 规范用代码号 32 激活。尽管前面的块仍然执行，但其他块已经可以指定。                                                                                                                                                                                                                                                                                                                                |
| 32768  | 使用该标志可以控制异步子程序（参见〈Starting block〉和〈End block〉）。                                                                                                                                                                                                                                                                                                                           |
| 65536  | 阻止预运行<br>可以使用此标志设置块预运行的设置。数据在 StartBlock 中被指定为 XML 字符串。它们被〈SearchRun〉和〈/SearchRun〉包围。因此，为了输入目标位置 100，传送以下 XML 字符串：<br>〈SearchRun〉〈DesPos〉 100 〈/DesPos〉 〈/SearchRun〉                                                                                                                                                                                                      |
| 131072 | 其他块如下。<br>与手动数据输入（32）和缓冲 NC 块（4096）的输入一起有效。如果块缓冲区为空（并生成向下斜坡），块预测不会停止，但会一直等到块再次输入为止。                                                                                                                                                                                                                                                                                       |
| 262144 | 重写：<br>如果在激活的程序中存在进给保持，NC 程序段将被传送到〈String〉中的通道。通道切换到覆盖模式，可以指定和处理更多的块。如果另外指定了 2048，则可以修改，删除已指定的块或重置语法错误。<br>该标志可以指定数控块进行覆盖。<br>它区分：<br><ul style="list-style-type: none"> <li>在自动模式下覆盖<br/>使用第一个规格时，如果存在进给保持，它会在自动模式下切换到内部覆盖模式。然后可以指定更多的数控块。</li> <li>在 Init 调整程序结束时在块预运行后重写<br/>如果数控在 Init 调节程序结束时处于“等待手动数据输入”状态，则可以使用该 ID 指定用于手动调节的数控块。该命令只能与块预运行（65536）一起使用。</li> </ul> |

表格 7-234: MCOPS(4,...)

**示例:**

```

10 ERR_VAR=MCOPS(4,1,"sekt.cnc","N50","N100",1)
;程序选择通道 1 中的程序 "sekt.cnc"，包括程序段搜索和链接。
20 ERR_VAR=MCOPS(4,1,/usr/"user/p1.cnc")
;程序选择通道 1 中的 "p1.cnc"，不进行块搜索和链接。
30 ERR_VAR=MCOPS(4,1,"F1000G1X500",,,32)
;使用手动数据输入选择通道 1 中的块 "F1000G1X500"
40 ERR_VAR=MCOPS(4,1,"S1000",,,262144)
;在辅助模式“覆盖”中激活进给保持并选择块 "S1000"。它用数控启动进行处理。
50 Flags=262144 + 2048
60 ERR_VAR=MCOPS(4,1,"STA(SP1)",,,Flags)
一个不正确的或已经指定的块被重置并且块 "STA (SP1)" 被选中。

```



如果未通过选择选择类型明确指定，则不会使用链接运行进行选择。

MCOPS (5, ...) – 取消选择程序

**描述：** 运动控制流程服务（5，...） – 运动控制流程服务（5，...）：取消选择程序调用 NCS 运动控制处理服务取消选择程序：  
在已编程的 <Channel> 中取消选择选定的程序或手动数据输入块。

|                                            |                                                                                                                                                                                                                                  |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MCOPS (5, <Channel>, <DeselectionVariant>) |                                                                                                                                                                                                                                  |
| <Channel>                                  | 整数型表达式<br>指出受函数支配的通道。                                                                                                                                                                                                            |
| <DeselectionVariant>                       | 整数型表达式<br>指定函数的行为。<br><br>以下列表包含所有定义的行为模式。每个模式提前都有一个 ID。要设置某种行为，必须在 <Deselection variant> 中提供对应 ID。如果要组合多个行为模式，则必须传输 <Deselection Variant> 中所有对应 ID 的总数。<br><br>到现在为止，该列表仅有一个 ID：<br><br>2: 如果自动程序重新选择处于活动状态，则可以取消该程序中的值 2 来抑制它。 |

表格 7-235: 语法 MCOPS (6, ...)

**示例：** 10 ERR\_VAR=MCOPS (5, 2, 2)  
;通道 2 中的程序取消选择没有自动程序重选。

MCOPS (6, ...) – 启动程序

**描述：** 运动控制流程服务（6，...） – 运动控制流程服务（9，...）：启动程序调用 NCS 的运动控制处理服务启动程序：  
启动已编程 <Channel> 中选定的程序或手动数据输入块。

|                      |                       |
|----------------------|-----------------------|
| MCOPS (6, <Channel>) |                       |
| <Channel>            | 整数型表达式<br>指出受函数支配的通道。 |

表格 7-236: 语法 MCOPS (6, ...)

MCOPS (7, ...) – 指定操作模式

**描述：** 运动控制流程服务（7，...） – 运动控制流程服务（7，...）：指定操作模式调用 NCS 的运动控制处理服务以指定操作模式：  
由此，仅对编程通道指定一种操作模式。  
操作模式只能在以下条件下更改：

- 通道接口信号 “qCh\_OpModePlc”（PLC 操作模式）可能未设置
- 数控没有选择程序或块
  - 或 -

仅在自动模式，连续程序段，程序段，单步或单程序段之间切换。

CPL 命令

|     |                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法： | MCOPS(7, <Channel>, <OperationMode>) |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|     | <Channel>                            | 整数型表达式<br>指出受函数支配的通道。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|     | <Operation mode>                     | 整数型表达式<br><div><div>1. 点动模式：<br/>轴可以点动 (+/-)。<br/>也可参阅 &lt;Selection type&gt; 下的 MCOPS(4..)：特性值 256。</div><div>2. 接近参考点：<br/>使用信号 "Manual+ " 或 "Manual-"可启动轴。<br/>也可参阅 &lt;Selection type&gt; 下的 MCOPS(4,...)：ID 512。</div><div>3. 块预运行：<br/>零件程序可在没有运动的情况下处理到目标块。</div><div>4. 手动数据输入：<br/>可指定加工各个块。</div><div>5. 自动（继续块）：<br/>零件程序已处理完毕。</div><div>6. 自动（程序块）：<br/>零件程序的程序段单独准备并通过 "数控启动"进行处理。</div><div>7. 程序块（单步）：<br/>零件程序的程序段是相互关联的，但是通过 "数控启动"进行单独处理。即使是内部生成的数控块也会被单独处理。</div><div>8. 教学：<br/>轴可以点动 (+/-) 或手轮进给。可指定加工各个块。</div><div>9. 未分配</div><div>10. 程序块（单块）：<br/>与单步相反，由于编程的 NC 程序段而在内部生成的所有程序段都将传送到插补器，以便在每次 NC 启动时进行处理。</div><div>11. 返回到轮廓：<br/>可手动将轴从轮廓移开，并后续接近。</div><div>12. CPL 调试器：<br/>与零件程序在零件程序中处理单块。</div><div>13. CPL 调试器：<br/>对下一个断点前的所有块进行处理。</div><div>14. 工件坐标系的点动：<br/>轴可以在工件坐标中点动。</div></div> |

表格 7-237: 语法 MCOPS(7,...)

示例： 10 ERR\_VAR=MCOPS(7, 2, 5)  
;第二个通道的操作模式更改为自动（继续块）。

MCOPS(8,...)- 更改返回轮廓策略

描述： 运动控制流程服务（8,...）- 运动控制流程服务（8,...）：改变返回到路径策略

调用 NCS 的运动控制处理服务以返回轮廓策略：  
由此，指定在已编程的 <Channel> 中返回路径策略。

|     |                                               |                                                                                                             |
|-----|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 语法: | MCOPS (8, <Channel>, <Mode>, <ApproachPoint>) |                                                                                                             |
|     | <Channel>                                     | 整数型表达式<br>指出受函数支配的通道。                                                                                       |
|     | <Mode>                                        | 整数型表达式<br>选项: <ul style="list-style-type: none"><li>1: 自动返回轮廓</li><li>2: 单块返回轮廓</li><li>3: 手动返回轮廓</li></ul> |
|     | <Approach point>                              | 整数型表达式<br>选项: <ul style="list-style-type: none"><li>1: 到起始点</li><li>2: 到终止点</li><li>3: 到中断点</li></ul>       |

表格 7-238: 语法 MCOPS (8, ...)

MCOPS (9, ...) – 完成返回轮廓录制

**描述:** 运动控制流程服务 (9, ...) – 运动控制流程服务 (9, ...)：完成返回轮廓录制  
  
调用 NCS 的运动控制处理服务以指定返回轮廓录制：  
完成在已编程的 <Channel> 中返回轮廓录制。从而不再记录点动运动。

|    |                      |                       |
|----|----------------------|-----------------------|
| 语法 | MCOPS (9, <Channel>) |                       |
|    | <Channel>            | 整数型表达式<br>指出受函数支配的通道。 |

表格 7-239: 语法 MCOPS (9, ...)

MCOPS (10, ...)

**描述:** 不适用于 MTX

MCOPS (11, ...) – 通道重启

**描述:** 运动控制流程服务 (11, ...) – 运动控制流程服务 (11, ...)：通道复位  
调用 NCS 的运动控制处理服务“重启通道”。该功能在所选主程序的开始处重新启动通道。没有重置。因此，开启状态也不会复位时应用。  
如果程序例如等待永久变量 (“WPV”)，则等待被取消。



模态不会被修改！例外情况：模态有效的子程序 / 循环。  
在主程序中，程序中使用的所有模态功能必须复位，因为所有激活的模态 NC 功能在通道重启时保持有效。

命令 MCOPS (11, ...) 仅在通道重启后或发生错误时才会返回。

**错误消息** 错误消息作为返回值返回。可能的错误信息在 第 4 章 “错误返回值” 第 482 页 中指定。  
  
错误代码 2: 通道重启任务已经在该通道中执行。  
错误代码 37: 由于错误，通道重启失败。

CPL 命令

错误代码 38：当前的操作模式不允许重新启动通道。有效的只是选择主程序的操作模式。这意味着（例如）手动数据输入和缓冲 NC 块指定无效。异步子程序也可能不活动。

- 顺序：
- 通道暂时不再接受任何任务，例如程序选择或操作模式切换。
  - 插补器已停止。
  - 通道的任务 - 未处理命令 - 被取消。
  - 激活程序重置为主程序的开始。
  - 本地和全局 CPL 变量被重置。
  - 模态子程序被禁用。
  - 插补器重启。
  - 通道切换到“数控未就绪”状态（PLC 输入信号 iCh\_NCReady）。

语法：

|                     |                       |
|---------------------|-----------------------|
| MCOPS(11,<Channel>) |                       |
| <Channel>           | 整数型表达式<br>指出受函数支配的通道。 |

表格 7-240: 语法 MCOPS(11,...)

- 示例：
- 10 ERR\_VAR=MCOPS(11, 1)  
； 在第一个通道重新启动通道。

错误返回值

描述：

所有命令调用都会为检查目的和错误处理提供返回值。该值可以分配给“整数”或“实数”变量。

语法示例：

ERR\_VAR% = MCOPS(...)  
ERR\_VAR% = MCODS(...)



错误的程序反应可能！

如果调用的命令返回一个错误代码，那么对于进一步执行程序来说非常重要的操作要么不被执行，要么没有完全执行

我们强烈建议您在调用命令后检查程序（例如使用 CASE），以确定命令是否可以正确执行。剩余的程序行为取决于错误的类型和严重程度。

定义了以下返回值：

- 0:
- 无错误发生
- 1:
- 指定的通道不存在。
- 2:
- 命令无法执行，因为指定的通道当前被分配（当前状态不允此动作）。
- 3:
- 在指定通道中已经启动的初始化复位还没有完成。
- 4:
- 指定的程序名称太长（当前未使用）。
- 5:
- 命令要求接近的参考点。
- 6:
- 指定的程序不存在或无法执行。
- 7:
- 用缓冲的数控块规范写入缓冲区被中断。同时试图写入缓冲器的第二种情况。
- 8:
- 命令无法在当前操作模式下执行。
- 9:
- 通道无法启动，因为状态未就绪。
- 10:
- 命令无法执行，因为未选择程序。
- 11:
- 无法选择指定的程序，因为通道状态不允许（例如，块准备和插补器处于“运行”状态）。
- 12:
- 目前未使用
- 13:
- 无法切换操作模式，因为通道状态不允许。
- 14:
- 未找到“搜索块”目标。



- 15: 虽然通道状态已就绪 – 且已经启动主程序处理（如程序在 M0 处），但无法执行“搜索块”。
- 16: 轴数对指定外部零点偏置来说太高。
- 17: Z0 轴组数对指定外部零点偏置来说太高。
- 18: 指定的语法未知。
- 19: 指定外部刀具更正时无效的索引。
- 20: 指定外部刀具修正时的修正次数过高（可能与修正指数一起）。
- 21: 提供外部工具更正时格式规格无效。
- 22: 指定外部刀具更正的边缘位置无效。
- 23: 修正组无效
- 24: 寻址轴不可用。
- 25: 将自动启动输入数控块时，检测到运行时错误，例如语法错误。
- 26: 指定缓冲数控块时，缓冲区溢出。
- 27: 坐标过滤器规范不正确。
- 28: 虚拟数控规范是不允许的，因为状态没有被冻结。
- 29: 服务器缺少内存。请求的数据无法存储。
- 30: 为 Ncs 函数指定了一个无效值。因此，服务器无法提供数据。
- 31: 异步子例程的规范不正确。
- 32: 块预运行的规范不正确。
- 33: 由于内插器尚未应用最后的校正值，所以校正值不能被传送。
- 34: 传递的客户端缓冲区太小。
- 35: 处理的呼叫链无效。
- 36: 为刀具校正编程了一个值。数值量要么太高，要么太低，以至于不再合理。
- 37: 由于错误，通道重启失败。
- 38: 由于当前操作模式和当前通道状态，通道重启无效。
- 100: 报文的幻数不正确。
- 101: NCS 通信不适当。
- 102: 此软件版本中没有指定的功能。
- 103: 发生内部错误（目前未使用）。



如果调用程序也包含数控块，则块处理通常在机器处理之前执行。如果在块准备时请求过程服务或使用 MCOPS 或 MCODS 命令查询机床状态，则机床上的先决条件尚未满足。

但是，这个问题仅适用于访问它们自己执行的通道的命令。

在这种情况下，请在呼叫前的行中使用“WAIT”命令。这会停止块准备，直到“WAIT”之前所有的块完全处理完毕。

编程示例

示例 1: 立即请求通道 2 的 SAV 状态和插补器

10 DIM BUF% (2) 创建数组

20 VERSION=0 立即提供数据

30 ERR\_VAR%=MCODS (32,2,VERSION,BUF%,2) 命令调用

BUF% (1) 包含 SAV 状态。BUF% (2) 包含 IPO 状态。

数据的当前版本号显示在 VERSIO N 中（例如重要的是 2）。

示例 2: “等待直至通道 2 的 SAV 状态切换为未激活”

<示例 1 的代码>

## CPL 命令

## 程序

---

```

10 INAKTIV = 1
20 WHILE BUF%(1) <> INAKTIV DO
30 ERR_VAR% = MCODS(32,2,VERSION,BUF%,2)
40 END

```

---



---

**示例 3:**     在 MSG 窗口中显示轴名称

---

```

:
30 VERSION=0
40 DIM AXNAME$(512)
50 REM Request all axis names
60 ERR=MCODS(33,-1,VERSION,AXNAME$,512)
70 IF ERR=0 THEN
80 REM Determine number of axes
90 DIM AXNMB%(3)
100 VERSION=0
110 ERR=MCODS(45,-1,VERSION,AXNMB%,3)
120 ANZ=AXNMB%(2)
130 ENDIF
140 IF ERR<>0 THEN
150 PRN#(0,"Error occurred: ",ERR)
160 ELSE 170 REM Displaying the axis names
180 FOR I%=0 TO (ANZ-1)
190 NAME$=MID$(AXNAME$,I%*9+1,8)
200 IF ASC(NAME$)<>0 THEN
210 REM Axis name defined
220 PRN#(0,I%+1,". Axis name: ",NAME$)
230 ENDIF
240 NEXT
250 ENDIF
N260 M30

```

---



---

**示例 4:**     通道/等待状态

---

```

10 CHAN%=1
20 VERSION%=0
30 STATES%=0
40 ERR=MCODS(87,CHAN%,VERSION%,STATES%,1)
50 IF ERR=0 THEN
60 MASKE%=1
70 WHILE MASKE% <= STATES% DO
80 CASE (STATES% AND MASKE%) OF
90 LABEL 1:PRN#(0,"Dwelling time")
100 LABEL 2:PRN#(0,"Acknowledgement required for help function")
110 LABEL 4:PRN#(0,"Read in blocker")
120 LABEL 8:PRN#(0,"Feed in channel equal to 0")
130 LABEL 16:PRN#(0,"Program Stop using M0/M1")
140 LABEL 32:PRN#(0,"Feed Stop in channel")
150 LABEL 64:PRN#(0,"Feed blocker in channel or in channel axis")
160 LABEL 128:PRN#(0,"Read in blocker from customer")
170 LABEL 256:PRN#(0,"Synchronized motion stop between channels (ASTOP, ...)")
180 LABEL 512:PRN#(0,"Waiting for axis during axis exchange (G511)")
190 LABEL 1024:PRN#(0,"Waiting for permanent variable (WPV)")
200 LABEL 2048:PRN#(0,"Waiting for interface signal to active point of time (WAITA, ...)")
210 LABEL 4096:PRN#(0,"Waiting for If signal (WAIT(BITIF(...))) or")
220 PRN#(0," (WAIT(,ZEIT%)) in block preparation")
230 LABEL 8192:PRN#(0,"Motion Control data service (MCODS(...))")
240 ENDCASE
250 MASKE%=MASKE%*2
260 END
270 ENDIF
M30

```

---



---

**示例 5:**     轴-通道分配

---

```

10 REM Program queries number of axes in system
15 REM and number of axis of channel axis. By means of the

```

```
20 REM information, the channel axes are set to
30 REM position 0 and then traversed to position <channel number>
40 REM
50 CHAN%=SD(8) : REM Individual channel number
60 IDCHAX%=43 : REM Function type for assignment axis - channel
70 IDMAXAX%=45 : REM Function type for number of axes
80 DIM BUF%(16): REM Buffer for assignment axis - channel
90 SIZE%=16
100 ANZ%=0 : REM max. index of system axes
120 REM Determine channel axes of active channel
130 VERSION=0
140 ERR=MCODS(IDCHAX%,CHAN%,VERSION,BUF%,SIZE%)
150 IF ERR=0 THEN
160 VERSION=0
170 ERR=MCODS(IDMAXAX%,CHAN%,VERSION,ANZ%,1)
180 IF ERR=0 THEN
190 FOR I%=1 TO ANZ%
200 IF BUF%(I%) = CHAN% THEN
N210 F1000 [AXP(I%,0,0)]; Traverse channel axes to 0
220 ENDIF
230 NEXT
240 FOR I%=1 TO ANZ%
250 IF BUF%(I%) = CHAN% THEN
N260 M0
N270 WAIT
N280 F1000 [AXP(I%,CHAN%,0)]; Traverse channel axes to CHAN%
290 ENDIF
300 NEXT
310 ENDIF
330 ENDIF
N310 M30
```

7.4.10 常规 XML 表  
XTAB

|        |      |       |       |      |      |       |   |
|--------|------|-------|-------|------|------|-------|---|
| STRUCT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|

**描述:** Xml TABLE - 读取/写访问任何 XML 表。  
与用于 Z0, D 校正和放置表的特殊命令相比, 如果定义了相应的数据类型, 该命令还可以访问表的任何子结构。

|                               |                                               |
|-------------------------------|-----------------------------------------------|
| XTAB (<Table>,<Substructure>) |                                               |
| <Table>                       | XML 表的名称, 其在适用情况下包括路径信息<br>没有指定路径时, 使用根目录中的表。 |
| <Substructure>                | 所需子结构, 被提供为 XPATH 表达式。                        |

表格 7-241: XTAB 语法

**示例:**

```
10 X!=XTAB("/database/PM1.pmt","/PMT/Set[1]/G154/Corr/Trans/XWCS")
20 XTAB("/database/PM1.pmt","/PMT/Set[1]/G154/Corr/Trans/XWCS")=X!+1.0
```

使表中的值增量 1。

XTABCR

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** Xml TABLE 创建 - 创建一个通用的 XML 表

|                                           |                                               |
|-------------------------------------------|-----------------------------------------------|
| XTABCR (<Table>,<Type>[,<TableTemplate>]) |                                               |
| <Table>                                   | XML 表的名称, 其在适用情况下包括路径说明<br>没有指定路径时, 使用根目录中的表。 |

CPL 命令

|                  |                                                                                                                                                                                                                      |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Type>           | 表类型（XML 根标签）<br>以下为支持的表类型： <ul style="list-style-type: none"><li>• /ZOT：零点偏置表</li><li>• /DCT：D 修正表</li><li>• /PMT：定位表</li></ul>                                                                                      |
| <Table template> | 模板 XML 表的名称（如果适用）包括路径。指定的 <Type> 必须与 <Table template> 的类型对应。<br>没有指定路径时，使用根目录中的表。                                                                                                                                    |
| ERRNO            | CPL 变量<br>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。<br>以下为可能的返回值：<br>0：访问正常<br>-1：参数错误<br>-13：数据无法写入<br>-14：XML 表已存在<br>-15：未发现 XML 表<br>-16：无效文件扩展名<br>-17：无效表类型<br>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。 |

表格 7-242: XTABCR 语法  
在 MTX 文件系统中创建一个没有数据的 XML 表。

语法:

示例:

```
10 XTABCR("/database/Z02.zot","/ZOT","/database/Z01.zot")
```

零点偏置表 Z02.zot 在目录 / 数据库中创建。表/database/Z01.zot 用作表模板。

7. 4. 11

刀具数据库

DBLOAD

|     |                                                                                                            |          |       |      |      |       |   |
|-----|------------------------------------------------------------------------------------------------------------|----------|-------|------|------|-------|---|
| --- | READ                                                                                                       | WRITE    | FUNCT | PROC | WAIT | ERRNO | ! |
| 描述: | <b>数据库负载</b> – 加载数据库<br>数据库表中的部分或来自文件的完整数据库表可以读入数据库。<br><DataSelection> 参数允许控制是否写入数据块的所有数据或仅读取特定刀具或特定位置数据。 |          |       |      |      |       |   |
| 语法: | DBLOAD(<DbTabelle>,<Key1>,<Key2>),<Filename>[,<DataSelection>])                                            |          |       |      |      |       |   |
|     | <DbTable>                                                                                                  | 数据库表的根标签 |       |      |      |       |   |
|     | <Key1>                                                                                                     | 数据库键 1   |       |      |      |       |   |
|     | <Key2>                                                                                                     | 数据库键 2   |       |      |      |       |   |





|        |                                                                                                                                                                                                                           |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Mode> | 可选（默认：0）<br><br>0：<br>数据被写入文件“FileName”。如果该文件已经存在，则其内容将被覆盖<br><br>1：<br>数据附加到现有文件“FileName”。                                                                                                                              |
| ERRNO  | CPL 变量。如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。<br>以下为可能的返回值：<br>0：访问正常<br>-1：参数错误<br>-7：数据无法读取<br>-13：数据无法写入<br>-20：根标签错误<br>-21：无法访问文件<br>-22：无效键<br>-23：数据块被锁定<br><br>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。 |

表格 7-245: DBSAVE 语法

如果两个键的值均为 -1，则将所有数据块写入文件。如果第一个键的值不等于 -1，第二个键的值为 -1，则将第一个键具有指定值的所有数据块写入文件。

语法示例:

```
DBSAVE("/DBT1",1,1,"/dbdaten.txt")
```

数据集 (1,1) 的内容保存到“dbdaten.txt”文件中。

DBSEA

| STRUCT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|
|--------|------|-------|-------|------|------|-------|---|

描述:

**数据库 SEArch - 搜索数据块**  
CPL 命令返回与搜索条件相对应的第一个数据块的标题。在这种情况下，变量 <SearchRes> 返回 1。  
数据块搜索从使用定义的数据块开始  
<StartKey1>和<StartKey2>  
定义的数据块。  
如果两个启动键其中一个键的值为 -1，则搜索将在数据库表的第一数据块中启动。

语法:

|                                                                                        |             |
|----------------------------------------------------------------------------------------|-------------|
| DBSEA (<DbTable>, <StartKey1>, <StartKey2>, <SearchCond>, <SearchRes>, [<br><ResVar>]) |             |
| <DbTable>                                                                              | 数据库表的名称     |
| <StartKey1>                                                                            | 起始数据块的 Key1 |
| <StartKey2>                                                                            | 起始数据块的 Key2 |

CPL 命令

|              |                                                                                                                                                             |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <SearchCond> | 以字符串搜索条件<br>(参见 第 章 “DBSEAX” 第 490 页 )                                                                                                                      |
| <SearchRes>  | 搜索结果，整数型变量类型<br>0:<br>找不到与搜索标准相对应的数据块。<br>1:<br>找到的数据块对应于搜索条件。                                                                                              |
| <ResVar>     | 整数类型的变量<br>如果输入 <ResVar>，则在发生访问错误时不会导致运行时错误。<br>以下为可能的返回值：<br>0: 访问正常<br>1: 访问时出错<br>2: 内存不足<br>3: 无效变量类型<br>4: 搜索标准错误<br>如果未指定 <ResVar>，则在发生访问错误时会导致运行时错误。 |

表格 7-246: DBSEA 语法

示例:

```
10 SD.DBRec.Hd=DBSEA("DBT1",-1,-1,"K1=1",FOUND%)
20 WHILE FOUND%=1 DO
30 SD.DBRec.Hd=DBSEA("DBT1", SD.DBRec.Hd.K1, SD.DBRec.Hd.K2, "K1=1",FOUND%)
40 END
```

以下系统日期是必需的:

程序

```
<Variable Storage="volatile">
 <Name>DBRec</Name>
 <Type>DBT1Rec_t</Type>
</Variable>
```

DBSEAX

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** **数据库 SEArch 扩展** - 扩展搜索数据块

在工具数据库表中搜索一个或多个数据块。该命令提供了与搜索条件相对应的数据块标题列表。报头被存储在系统数据数组 <HeaderArr> 中，按 K1 和 K2 排序。参数 <ErgSize> 用于指定要搜索的数据块头的最大数量。DBSEAX 的返回值提供了找到的数据块的数量。

数据块搜索从使用定义的数据块开始

<StartKey1>和<StartKey2>

定义的数据块。

如果两个启动键其中一个键的值为 "-1"，则搜索将在数据库表的第一数据块中启动。



语法:

|                                                                                      |                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DBSEAX (<DbTabelle>, <StartKey1>, <StartKey2>, <SearchCond>, <HeaderArr>, <ArrSize>) |                                                                                                                                                                                                                                            |
| <DbTable>                                                                            | 数据库表的名称                                                                                                                                                                                                                                    |
| <StartKey1>                                                                          | 起始数据块的 Key1                                                                                                                                                                                                                                |
| <StartKey2>                                                                          | 起始数据块的 Key2                                                                                                                                                                                                                                |
| <SearchCond>                                                                         | 搜索条件作为字符串（见下文）                                                                                                                                                                                                                             |
| <HeaderArr>                                                                          | 类型数据块标题的系统数据<br>（例如 DBT1Hd_t）<br><br>如果要搜索几个数据块，它必须是一个数组。该数组通过递增 SD 中的最后一个索引来索引，例如<br>SD. HdArr[1, 1]<br>- HdArr 中的第二个索引增加。<br>SD. SDArr[1, 1]. RecArr[1]. Hd<br>- RecArr 的索引增加。                                                           |
| <ArrSize>                                                                            | 已搜索数据块标题的最大数量<br><b>注意:</b> <HeaderArr> 必须具有足够的尺寸以接受这个数量的标题。                                                                                                                                                                               |
| ERRNO                                                                                | CPL 变量<br><br>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。错误由变量的相应值提供。<br><br>以下为可能的返回值：<br><br>0: 访问正常<br><br><0: 错误（有关详细的错误描述，请参阅 <a href="#">第 3.14.5 章“评价 CPL 中功能错误的变量 ERRNOERRNO”第 69 页</a> ）<br><br>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。 |

表格 7-247: DBSEAX 语法

搜索条件 (<<SearchCond>)

维护以下语法:

- 提供了必须满足的个别条件/标准的列表。
- 正常的单一条件如下所示:  
    <标签名称> <关系运算符> <值>与
  - <关系运算符>:
    - "=" 等于
    - ">" 大于
    - ">=" 大于等于
    - "<" 小于
    - "<=" 小于等于
    - "<>" 不等于

只有 SKQ 作为 SKQ 的关系运算符: "=" and "<>" ,  
例如 SKQ=4711
- 单个位掩码条件如下所示:  
    <标签名称> = <值>: <掩码>

CPL 命令

- <Value>: 请注意所需的相关位。
- <Mask>: 定义相关位。

例如: BQ1=0x307000:0xF0FF00 with:

只有关于平等的疑问才有可能。

解释值和掩码:

- 如果它们是以 "0x" 或 "0X"开头的十六进制数字,
- 作为八进制数, 如果它们以 "0"开头
- 作为所有其他情况下的十进制数。

在 <Value> 中, 不会分析前面或后面的空格。如果要指定一个字符串, 其中这些空格是重要的, 则该子字符串必须包含反向逗号 (例如: SKQ = ' ' )。

要在以引号括起来的子字符串中使用反转逗号, 字符 "\" 必须在反向逗号前面 (例如: SKQ=" Achim 的\" S' )

- 边界条件
- 个别条件由 a ", "分隔。
  - 列表中的所有单个条件必须引用不同的 <Tag> 名称。
  - 只能使用预定义的标签名称。

语法示例:

```
K1=1
IKQ2=3
BQ2=0x1:0x1
```

必须找到满足搜索条件的所有数据块。为此, 搜索必须经常重复, 以确保找到的数据块的数量低于搜索数据块的最大数量。对于每次新搜索, 必须将起始密钥设置为最新找到的数据块的密钥。

程序

```
10 KEY1%=-1
10 KEY2%=-1
30 ARRSIZE%=10
31 REPEAT
33 FOUND% = DBSEAX("DBT1",KEY1%,KEY2%,"K1>0",SD.HdArr[1],ARRSIZE%,ERRNO)
35 IF ERRNO=0 THEN
37 IF FOUND%>=1 THEN
59 FOR I%= 1 TO FOUND%
43 PRN#(0,Sector: ",SD.HdArr[I%].K1,", Place: ",SD.HdArr[I%].K2)
46 NEXT I%
49 KEY1%= SD.HdArr[FOUND%].K1
51 KEY2%= SD.HdArr[FOUND%].K2
53 ENDIF
55 ELSE
57 PRN#(0,"Access error on tool database!")
61 ENDIF
63 UNTIL (FOUND%<ARRSIZE%) OR (ERRNO<>0)
```

以下系统日期是必需的:

程序

```
<Variable Dimension="10" Storage="volatile">
 <Name>HdArr</Name>
 <Type>DBT1Hd_t</Type>
</Variable>
```

DBTAB

| STRUCT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|
|--------|------|-------|-------|------|------|-------|---|

描述:

**数据库 TABLE** - 只读或写访问数据库

可以将完整的数据集或刀具数据库表的子结构读取到 CPL 变量中, 或从变量写入数据集。

|           |  |                                                                                                                                                  |  |
|-----------|--|--------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 语法:       |  | DBTAB (<DbTable>, <key1>, <key2>[, <ResVar>])                                                                                                    |  |
| <DbTable> |  | 包含子结构的数据库表名 (Xpath 字符串)。一个 "." 或 "/" 可以用作结构组件之间的分隔符。                                                                                             |  |
| <Key1>    |  | 数据库键 1                                                                                                                                           |  |
| <Key1>    |  | 数据库键 2                                                                                                                                           |  |
| <ResVar>  |  | 整数类型的变量<br>如果输入 <ResVar>, 则在发生访问错误时不会导致运行时错误。<br>以下为可能的返回值:<br>0: 访问正常<br>1: 访问时出错<br>2: 内存不足<br>3: 无效变量类型<br>如果未指定 <ResVar>, 则在发生访问错误时会导致运行时错误。 |  |

表格 7-248: DBTAB 语法

示例:

```
10 SV.A=DBTAB ("DBT1.Rec",1,1)
20 D!=DBTAB ("DBT1.Rec.UD.Ed.Geo.L1",1,1)
30 SV.A.UD.Ed.Geo.L1 = 10
40 DBTAB ("DBT1.Rec",1,1) = SV.A
```

DBTABX

| STRUCT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|--------|------|-------|-------|------|------|-------|---|
|--------|------|-------|-------|------|------|-------|---|

描述:

**数据库表扩展** - 扩展访问数据库表

可以将完整的数据集或刀具数据库表的子结构读取到 CPL 变量中, 或从变量写入数据集。参数 <DataSelection> 可用于控制是写入数据块的所有数据还是仅写入特定于工具或地点的数据。

|           |  |                                                         |  |
|-----------|--|---------------------------------------------------------|--|
| 语法:       |  | DBTABX (<DbTabelle>, <Key1>, <Key2>[, <DataSelection>]) |  |
| <DbTable> |  | 包含子结构的数据库表名 (Xpath 字符串)。一个 "." 或 "/" 可以用作结构组件之间的分隔符。    |  |
| <Key1>    |  | 数据库键 1                                                  |  |
| <Key2>    |  | 数据库键 2                                                  |  |



|                 |                                                                                                                                                                                                            |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <DataSelection> | 在写访问的情况下，不管<DataSelection>如何，锁都被删除。<br>以下值是可能的：<br>0：数据块的所有数据都被写入<br>1：仅写入特定刀具数据<br>2：只写入地点特定的数据<br>3：没有数据被写入<br>如果未指定 <DataSelection>，则在写入访问的情况下，将使用 0 作为默认值。<br>在只读访问的情况下，<DataSelection> 不相关。           |
| ERRNO           | CPL 变量<br>如果将 CPL 变量 ERRNO 指定在参数列表中的任何位置，则不会产生运行时错误。<br>错误由变量的相应值提供。<br>以下为可能的返回值：<br>0：访问正常<br><0：错误（有关详细的错误描述，请参阅 第 3.14.5 章 “评价 CPL 中功能错误的变量 ERRNOERRNO” 第 69 页）。<br>如果未指定 CPL 变量 ERRNO，则访问错误时会产生运行时错误。 |

表格 7-250: DBTABXL 语法

示例:

```
10 SV.A = DBTABXL("DBT1.Rec",1,1)
20 D! = DBTABXL("DBT1.Rec.UD.Ed.Geo.L1",1,1)
30 SV.A.UD.Ed.Geo.L1 = 10
40 DBTABXL("DBT1.Rec",1,1) = SV.A
```

7. 4. 12

错误和警告

SETERR

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

描述:

**SET ERRor** - 输出运行时错误

在对 ERRNO 进行分析后，CPL 命令 SETERR 可用于生成运行时错误。此时取消零件程序。输入到参数 <ErrorString> 中的文本总是与输出语言无关地输出。

使用可选参数 <Mode>，可以在当前通道中控制它，无论零件程序名称和零件程序行是否另外输出。对于不同通道的输出，仅输出指定的文本

语法:

|                                                 |                                               |
|-------------------------------------------------|-----------------------------------------------|
| SETERR(<ErrorString> [, [<Channel>] [,<Mode>]]) |                                               |
| 其中                                              |                                               |
| <Error string>                                  | 文本输出错误，错误号为 3371。                             |
| <Channel>                                       | 要输出错误的通道的通道号。<br>如果未指定 <Channel>，则将当前频道用作默认值。 |

CPL 命令

|        |                                                                                                        |
|--------|--------------------------------------------------------------------------------------------------------|
| <Mode> | 指定零件程序名称和零件程序行是否在当前通道中输出。<br>0:<br>没有输出零件程序信息<br>1:<br>零件程序名称和零件程序行的输出。<br>如果未指定 <Mode>, 则将使用 0 作为默认值。 |
| ERRNO  | CPL 变量可以在括号中的任何位置进行编程。<br>当 ERRNO 编程时, 不输出运行时错误。<br>返回值是:<br>0: 访问正常<br>-1: 参数错误<br>-5: 通道不存在          |

表格 7-251:     SETERR 语法

示例:

```
读取通道轴 X 在第一个通道中的位置
10 POS = ACS("X",1,1,ERRNO)
20 IF ERRNO <> 0 THEN
30 SETERR("position could not be read")
40 ENDIF
```

SETWARN

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

描述:

**SET WARNING** – 输出运行时警告

在对 ERRNO 进行分析后, CPL 命令 SETWARN 可用于生成警告。零件程序继续运行。不管语言设置如何, 参数 <WarningString> 中指定的文本总是被输出。

使用可选参数 <Mode>, 可以在当前通道中控制它, 无论零件程序名称和零件程序行是否另外输出。对于不同通道的输出, 仅输出指定的文本。

语法:

|                                                           |                                             |
|-----------------------------------------------------------|---------------------------------------------|
| SETWARN(<WarningString> [, [<Channel>]][, <Mode>]])<br>其中 |                                             |
| <Warning string>                                          | 文本输出为警告错误号 3372。                            |
| <Channel>                                                 | 要输出警告的通道号。<br>如果未指定 <Channel>, 则将当前频道用作默认值。 |

|        |                                                                                                        |
|--------|--------------------------------------------------------------------------------------------------------|
| <Mode> | 指定零件程序名称和零件程序行是否在当前通道中输出。<br>0:<br>没有输出零件程序信息<br>1:<br>零件程序名称和零件程序行的输出。<br>如果未指定 <Mode>, 则将使用 0 作为默认值。 |
| ERRNO  | CPL 变量可以在括号中的任何位置进行编程。<br>当 ERRNO 编程时, 不输出运行时错误。<br>返回值是:<br>0: 访问正常<br>-1: 参数错误<br>-5: 通道不存在          |

表格 7-252:     SETWARN 语法

CLRWARN

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:**     **清空警告 - 删除警告**  
使用 CLRWARN 命令清除由 SETWARN 生成的通道的所有警告。

|            |                                                                                               |
|------------|-----------------------------------------------------------------------------------------------|
| <b>语法:</b> | CLRWARN([<Channel>] )<br>其中                                                                   |
| <Channel>  | 需要将通道中由 SETWARN 生成的所有警告清除的通道号。<br>如果未指定 <Channel>, 则将当前频道用作默认值。                               |
| ERRNO      | CPL 变量可以在括号中的任何位置进行编程。<br>当 ERRNO 编程时, 不输出运行时错误。<br>返回值是:<br>0: 访问正常<br>-1: 参数错误<br>-5: 通道不存在 |

表格 7-253:     CLRWARN 语法

**示例:**  
10 CLRWARN (,ERRNO)  
20 IF ERRNO <> 0 THEN  
30 SETWARN("warnings not cleared")  
40 ENDIF

GETERR

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:**     **GET ERROR - 查询错误**  
该命令查询 CPL 程序中的当前错误。它包含当前错误号, 错误的通道号和相应的错误类别。  
任何发生的错误及其元素都将保存到数组中。阵列中要接受的最大错误数量受参数<错误编号>的尺寸标注 (DIM) 限制。

CPL 命令

GETERR 命令提供以下返回值:

- 1: 命令无法执行
- ≥0: <Channel> 中待处理错误的数量

语法:

|                                                         |                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GETERR (<Channel>, [<Category>], <ErrorNo>[, <Number>]) |                                                                                                                                                                                                                                                                                                                                                                              |
| <Channel>                                               | 被查询通道的通道编号<br>-1: 所有通道<br>> 0: 通道数量                                                                                                                                                                                                                                                                                                                                          |
| <分类>                                                    | 0: 所有警告和错误 (默认)<br>1: 低优先级系统错误<br>2: 控制或驱动错误<br>3: 插值器错误<br>4: 硬件错误<br>5: ICL 错误<br>6: 部件程序错误<br>7: 运行时警告<br>8: MZA 消息: 错误<br>9: MZA 消息: 警告<br>10: MZA 消息: 注解                                                                                                                                                                                                                |
| <Error number>                                          | 结果变量:<br>在第二维中至少有三个元素的二维整数数组 (DIM <ErrorNo> % (x, 3) ), 默认值: 0.<br>该命令按照时间递减顺序从 <通道> 提供当前错误编号。<br>第二维的三个要素的描述: <ul style="list-style-type: none"><li>&lt;Error no. &gt;(x, 1): 错误编号</li><li>&lt;Error no. &gt;(x, 2): 错误频道<br/>(-1 = 跨通道)</li><li>&lt;error no. &gt;(x, 3): 错误类别 (如果由 DIM 命令声明)。<br/>0 = 未知类别<br/>除 &lt;category&gt; 以外的值</li></ul> 示例: DIM ERRNO% (100, 3). |

表格 7-254: GETERR 语法 1



只能输入没有尺寸 / 指数的变量名称。



|          |                                                                                                                                                                                                                        |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Number> | <p>整型变量（默认值：1）</p> <p>指定要读取的错误数量。</p> <p>1:<br/>默认值</p> <p>&gt; 0:<br/>不检查参数值的有效性，即如果要读取的数据数量超出数组维度，则不会生成零件程序错误。</p> <p><b>示例:</b></p> <p>&lt;Number&gt; = 120，但是 DIM ERRNO% (100, 3)。</p> <p>在这种情况下，有 20 个错误无法读取。</p> |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

表格 7-255: GETERR 语法 2

**示例:** 查询通道 2 中的上一个部件程序错误

|                                                                                                                                                                                                                                                                                                                                 |  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <pre>10 DIM ERR_FELD%(5,3): REM integer array with 5 elements 20 REM queries last parts program error in channel 2 30 CHAN%=2: CATEGORY%=6 40 ERG%=GETERR (CHAN%,CATEGORY%,ERR_FELD%,1)</pre>                                                                                                                                   |  |
| <p>5 个数组元素的结果分析</p>                                                                                                                                                                                                                                                                                                             |  |
| <pre>10 DIM ERR_FELD%(5,3): REM integer array with 5 elements 20 CHAN%=2: CATEGORY%=0 25 REM queries warnings and error in channel 2 30 ERG%=GETERR (CHAN%,CATEGORY%,ERR_FELD%,5) 40 FOR I%=1 TO RES%     50 IF ERR_FELD%(I%,3)=6 THEN         60 PRN#(0,"parts program error: ",ERR_FELD%(I%,1))     70 ENDIF 80 NEXT I%</pre> |  |

CPL 命令

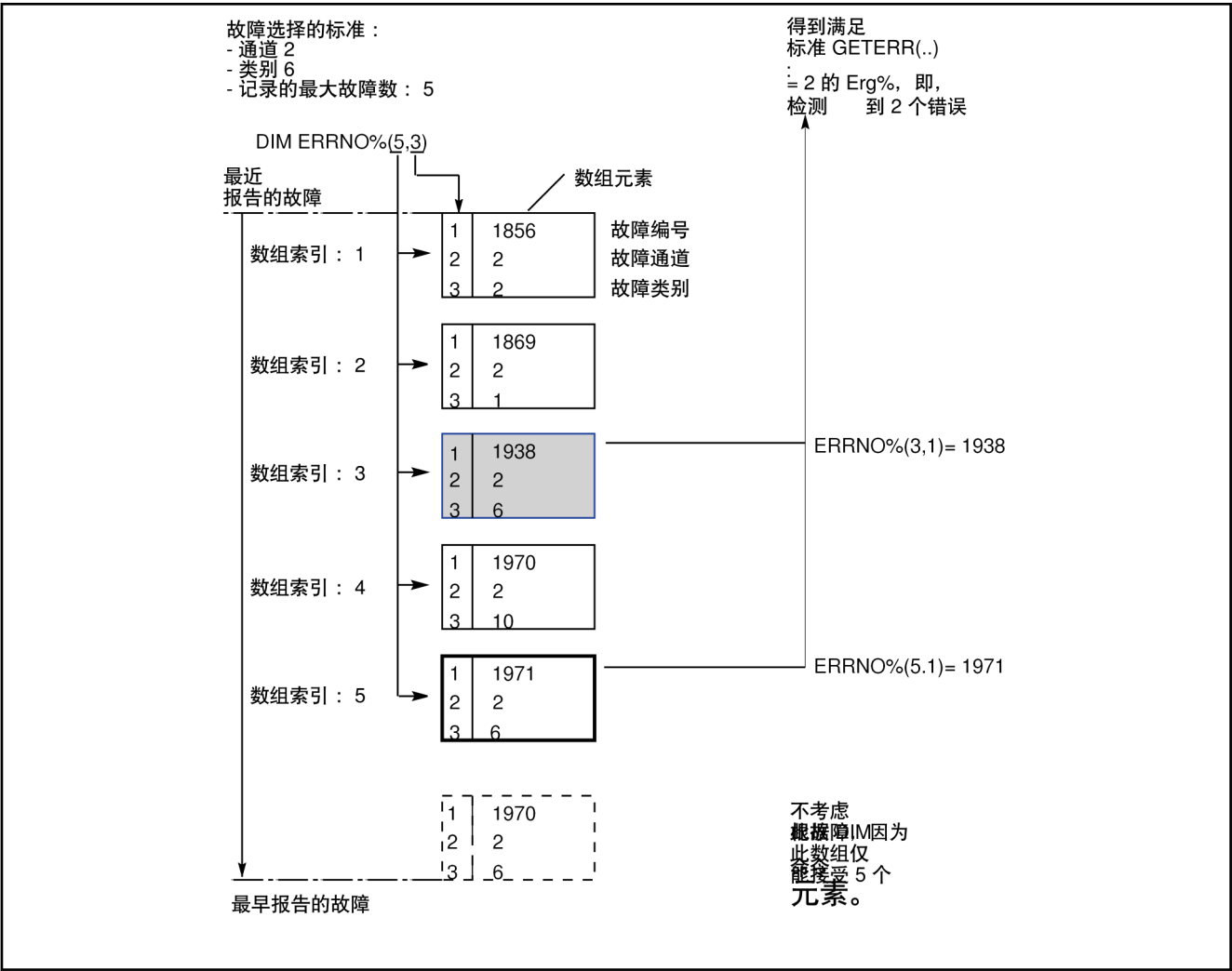


插图 7-6: 示例

结果，零件程序错误 1938 和 1971 显示在 MSG 窗口中。变量 Erg% 包含值 2。

 GETERR 可用于记录一个或多个错误的时间顺序发生，从而可以追踪错误的实际原因。

7.5 文件和目录

7.5.1 一般信息

描述

文件是数据的容器。在 CPL 程序中，数据可以从文件读取或写入文件。因此，测量值可以先保存，然后显示或打印。在 CNC 中，文件在文件系统中进行管理。他们按照等级结构组织起来。单个文件可以使用目录路径访问。文件可以使用各种操作模式进行管理为了提供对数据的只读或写入访问，必须首先打开相应的文件（参见 OPENW、OPENR 命令）。只要不再需要访问数据，文件就会关闭（请参阅 CLOSE 命令）。

文件名称

描述： 请遵守文件名的以下惯例：

- 最大长度 = 30 个字符。  
名称和任何可能的文件扩展名之间没有区别。所有字母，数字以及特殊字符 “.” 和 “\_” 是有效的。  
零件程序的文件名不得超过 28 个字符，因为在连接期间，NC 会生成一个文件，其名称由原始文件名和两个附加字符组成。



内部生成的文件仅使用特殊字符 “\$”。外部文件名称（通过用户界面或 DNC）不得包含 “\$”。

- 字母区分小写和大写。  
示例：文件名称  
P123456789. PRG  
P12\_Data\_Dial  
P12\_DATA\_DIAL
- 文件名称 “.” 和 “..” 无效，因为它们已经在内部使用。
- 文件名在目录内必须是唯一的。但是，具有相同名称的文件可以存在于不同的目录中。

随机文件结构

**描述:** 随机文件具有**固定的，定义长度的组件**（记录）。这允许对文件中的任何元素进行随机、直接访问。可按固定长度将随机文件分成记录，这允许对某条记录进行直接访问。至于顺序文件，数据保存为 ASCII 字符。这允许使用编辑器进行熟悉的访问以及读取随机文件。

随机文件的优点是可以更快地访问所需的数据。另外，可以处理和/或修改记录的数据而不改变剩余文件的结构。没有完全填充数据的记录将填充空白，直到定义的长度。

如果试图将一个 STRING 变量插入一个随机文件中，并且该变量的长度大于该记录的长度，则该记录会用 STRING 变量的第一个字符填充到定义的长度，其余字符将被忽略。

读取文件时，其结尾由 “EOF” 标识。REWRITE 和 CLOSE 指令用于顺序文件。

随机文件也可以顺序访问。

顺序文件结构

**描述:** 顺序文件具有一系列 **长度可变** 的 **组件**（记录）。如果在顺序文件中搜索某个记录，则搜索从文件的开始处开始。直接访问是不可能的。当在顺序文件中更改记录长度时，必须移动所有后续记录。

与随机文件相反，顺序文件包含可变的长度（最大长度 = 1024 个字符）记录。记录的结尾用 <LF> 标记，该 <LF> 不计为长度的一部分。在文件的最后一个记录之后，插入代表 EOF 指针的 <ETX> <LF>。EOF 指针指示文件中可用数据的结尾（<ETX>）。

7.5.2

目录

DIRINF

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** **目录信息** – 管理目录

DIRINF 允许在 CPL 程序中查询 IndraMotion MTX 文件系统的管理数据。该命令提供了一个 “整数” 类型的值（对于负值结果，实际结果大于最大整数值，即 2.147.483.647）。

语法根据参数 <Index1> 的值而变化。

CPL 命令

|     |                                                                                                                                                                                                                                                       |                                                                                                                                                            |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | <b>&lt;Index1&gt; = 1、2 或 3:</b><br>DIRINF (<Index1>[, <Index2>])<br><b>for &lt;Index1&gt; = 4 or 6:</b><br>DIRINF (<Index1>[, <Index2>], <ResVar>[, <FileName>])<br><b>&lt;Index1&gt; = 5、7 或 8:</b><br>DIRINF (<Index1>[, <Index2>], <ResultVar>[]) |                                                                                                                                                            |
|     | <索引 1>                                                                                                                                                                                                                                                | 整数类型的常量<br>命令模式:<br>1: 有效内存<br>2: 分配的内存<br>3: 目录中的文件数量<br>4: 索引<文件编号>目录中文件的名称<br>5: 活动主程序的目录名称<br>6: 具有索引<文件编号>的装载目录的名称<br>7: 活动主程序或子例程的目录名称<br>8: 调用程序的名称 |
|     | <索引 2>                                                                                                                                                                                                                                                | 字符串类型<br>附加模式:<br>目录名称 (默认: 当前目录)                                                                                                                          |
|     | <Event var>                                                                                                                                                                                                                                           | 键入尺寸标注的字符串变量, 仅适用于命令模式 4 - 8。<br>提供指定目录中文件的名称或命令执行后的目录名称。                                                                                                  |
|     | <文件编号>                                                                                                                                                                                                                                                | "整数" 类型, 仅用于命令模式 4 和 6<br>在<FileName>中, 指定目录中文件的序列号。它的名字在<ResVar>中提供。<br>如果序列号无效, 则 "NUL" 以 <ResVar>返回。                                                    |

表格 7-256: DIRINF 语法

|      |                  |                                             |
|------|------------------|---------------------------------------------|
| 返回值: | <Command mode 1> | 空闲内存以字节为单位                                  |
|      | <Command mode 2> | 以字节为单位指定的内存                                 |
|      | <Command mode 3> | 目录中的文件数量                                    |
|      | <Command mode 4> | 0 (<Result var>包含文件名) - 或 -<br>1 (<事件变量>为零) |
|      | <Command mode 5> | 0 (<ResultVar>包含目录名称) - 或 -<br>1 (<事件变量>为零) |
|      | <Command mode 6> | 0 (<ResultVar>包含目录名称) - 或 -<br>1 (<事件变量>为零) |
|      | <Command mode 7> | 0 (<ResultVar>包含目录名称) - 或 -<br>1 (<事件变量>为零) |
|      | <Command mode 8> | 0 (<Result var>包含文件名) - 或 -<br>1 (<事件变量>为零) |

示例: ; 中的当前目录的所有文件名

```
; 显示 MSG 窗口
10 DIM FILENAME$(30)
20 DIM DIRNAME$(80)
30 LJUST
40 ERG%= DIRINF(5,, DIRNAME$)
45 IF ERG%= 0 THEN
50 ANZ_FILES%= DIRINF(3, DIRNAME$)
60 FOR LNR%= 1 TO ANZ_FILES%
70 ERG%= DIRINF(4, DIRNAME$, FILENAME$, LNR%)
80 PRN#(0, LNR%, "：", FILENAME$)
90 NEXT LNR%
95 ENDIF
M30
```

示例：; 读取控件的所有安装目录

```
10 LJUST
20 DIM T$(100)
30 FOR I%= 1 TO 100
40 ERG%= DIRINF(6,, T$, I%)
50 IF ERG%=0 THEN
60 PRN#(0, I%, "：", T$)
70 ELSE
80 PRN#(0, "没有其它安装目录可用！")
90 I%=100
100 ENDIF
110 NEXT I%
M30
```

DIRCR

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

描述：创建目录 - 创建目录  
可使用 DIRCR 在 CPL 程序中创建新目录。该函数提供 “整数”型值。

|     |                    |                                                      |  |  |  |  |  |
|-----|--------------------|------------------------------------------------------|--|--|--|--|--|
| 语法： | DIRCR(<directory>) |                                                      |  |  |  |  |  |
|     | <Directory>        | 具有作为字符串表达式的完整路径的目录名称。<br>如果未提供路径，则当前目录被定位在字符串表达式的前面。 |  |  |  |  |  |

表格 7-257: DIRCR 语法

返回值：0: 目录已创建。  
1: 不可能创建目录

示例：10 I% = DIRCR("/usr/user/test")  
M30

DIRDEL

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

CPL 命令

描述:

目录创建 - 删除目录

DIRDEL 允许删除 CPL 程序中的空目录。该函数提供 “整数”型值。

语法:

|                     |                       |
|---------------------|-----------------------|
| DIRDEL(<Directory>) |                       |
| <Directory>         | 具有作为字符串表达式的完整路径的目录名称。 |

表格 7-258: DIRDEL 语法

返回值:

0: 目录已删除

1: 不可能删除目录 DIRCR 格式化。

示例:

10 I% = DIRDEL("/usr/user/test")

M30

7. 5. 3

文件操作

FILENO

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

描述:

文件编号 - 逻辑文件编号

逻辑文件号码是访问文件所必需的。这些可以设定 1 到 9 之间的值（即最多可以通过 CPL 同时打开 9 个文件）。命令 FILENO 允许查询下一个有效的逻辑文件编号。如果该命令返回 “-1”，则没有可用的逻辑文件编号可用。返回值是 “整数”类型。

示例:

10 LOG\_NO%= FILENO

20 IF (LOG\_NO% <> (-1)) THEN

30 OPENW(LOG\_NO%,"/user/usr/Test")

40 PRN#(LOG\_NO%,"Date: ",DATE)

50 CLOSE (LOG\_NR%)

60 ELSE

70 PRN#(0,"No free log. file number available!")

80 ENDIF

M30

OPENW, OPENR

| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

描述:

OPEN 写入 - 打开文件进行写入访问

OPEN 读取 - 打开文件以进行只读-访问

用于打开文件的命令取决于所需的访问类型:

- 写访问: OPENW
- 只读访问: OPENR

如果要打开的文件不存在，则在打开时创建它。

已经打开写入的文件也可以打开，通过 “OPENR” 命令进行只读访问。已打开的文件不能为了写访问而再次打开。

可使用其他参数打开随机文件，其以字节为单位将记录长度返回到文件中（1 字节 = 字符的长度）。在所有其他方面，命令结构对应于顺序文件中的一个。

OPENR 命令之后，文件指针被定位在第一个文件记录上，然后该记录可被读取。OPENW 命令之后，文件指针被定位在 EOF 指针上，即在文件的最后记录之后。

语法:

|                                                                                                       |                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| OPENW (<n>,<FileName>[, [<EmptyParam>][,<RecordLength>]])<br>OPENR (<n>,<FileName>[, <RecordLength>]) |                                                                                                                                                          |
| <n>                                                                                                   | 用于解决文件的逻辑编号。<br>值从 1 至 9 都可被选择。逻辑编号必须编程为 INTEGER 表达式。不可以将逻辑编号分配用于同时对文件进行读取和写入。可同时打开最多九个不同的文件。<br>如果不属于该值的范围，则会出现以下错误消息：“INVALID LOGICAL NUMBER”（无效逻辑编号）。 |
| <FileName>                                                                                            | 字符串表达式：文件名；可选地具有完整路径。<br>如果没有路径指定，则会在当前目录中搜索文件。                                                                                                          |
| <empty parameter>                                                                                     | （未分配）                                                                                                                                                    |
| <Record length>                                                                                       | 记录的字节数<br>取值范围：1..1024<br>如果不属于该值范围，则会出现错误消息 “INVALID COMPONENT LENGTH” 。                                                                                |

表格 7-259:      读取访问和写入访问的语法

示例:

```
50 OPENW(1,"P500","This is my best program")
40 A$="P500" : B1$="This is my best program"
50 OPENW(9,A$,B1$)
50 OPENW(7,"PDaten_Mes.DAT","Save measurement data")
```

当为写访问而打开文件时，需要对其进行检查以确保随机结构被保持。如果结构已被编辑器破坏，会出现以下错误消息：“INVALID COMPONENT LENGTH”（无效元素长度）。

```
10 OPENW(2,"P200",,10)
20 FOR I% = 1 TO 3
30 PRN#(2,"TEST")
40 NEXT I%
50 CLOSE(2)
```

结果: "P2"

```
TESTE <LF>
TESTE <LF>
TESTE <LF>
<ETX><LF>
```

当为读取访问而打开文件时，需要对其进行检查以确保随机结构被保持。所有组件必须具有已在 OPENR 命令中指定的相同长度。

```
1 OPENW(2,"P200",,"TEST",10)
2 PRN#(2,"ABC")
3 CLOSE(2)
4 OPENR(1,"P2",5)
5 CLOSE(1)
```

检查文件 "P2" 的记录长度是否是 5。但是，这个文件的记录长度是 10。

CPL 命令

```
P1:
N10 G1F10000X1000Y1000Z1000
1 A$="01234567890123456789"
2 B$="TEST"
N20 X0
M30
P2:
1 OPENW(1,"PMess_PRG",,"RANDOMDATEI",10)
2 OPENR(2,"P1")
3 DIM A$(30)
4 FOR I% = 1 TO 5
5 INP#(2,A$)
6 PRN#(1,A$)
7 NEXT
8 CLOSE(2)
9 CLOSE(1)
结果: PMess_PRG:
N10 G1F100<LF>
1 A$="0123<LF>
2 B$="TEST<LF>
N20 X0 <LF>
M30 <LF>
<ETX><LF><LF>
```

如果结构已被编辑器破坏，会出现以下错误消息：“INVALID COMPONENT LENGTH”（无效元素长度）。

REWRITE

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** RE WRITE - 覆盖文件内容

如果数据已经存在于打开的文件中，当写入时，新数据通常被附加到现有数据。但是，可通过“REWRITE”来重写文件，而无需特别地删除不再需要的内容。覆盖时，保留在 OPENW 命令中的范围在零件程序内存中仍然可用。

|               |                     |
|---------------|---------------------|
| REWRITE (<n>) |                     |
| <n>           | 逻辑文件编号（值范围 1 ... 9） |

表格 7-260: REWRITE 语法

在覆盖文件之前，请打开它。

CLOSE

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** “CLOSE” - 关闭文件。



可同时打开最多九个文件。如果打开九个文件并且需要再访问一个文件，请首先关闭一个文件。

因此，打开的文件通常在完成只读或写入操作后立即关闭。

|     |             |                        |
|-----|-------------|------------------------|
| 语法: | CLOSE (<n>) |                        |
|     | <n>         | 1 至 9:<br>要关闭的文件的逻辑编号。 |

表格 7-261: 关闭语法

|     |                                              |
|-----|----------------------------------------------|
| 示例: | :                                            |
|     | 90 DIM A\$(35)                               |
|     | 100 XPOS = MCS(1)                            |
|     | 110 YPOS = MCS(2)                            |
|     | 120 OPENW(1,"P5",500,"ACHSPOS")              |
|     | 130 REWRITE(1)                               |
|     | 140 PRN#(1,"X-axis",XPOS,YPOS,"Y-axis",YPOS) |
|     | 150 CLOSE(1)                                 |
|     | 160 OPENR(1,"P5")                            |
|     | 170 INP#(1,A\$)                              |
|     | 180 CLOSE(1)                                 |
|     | :                                            |

在示例中，X 轴和 Y 轴的当前位置被传送给变量（行 90 至 110）。之后，文件 1 被打开，并被存储为零件程序 P5（行 120）。

随后写入或重写文件，然后将文件关闭（行 140 至 150）。之后，文件被打开以进行读取，同时内容被分配给 A\$ 变量。只读访问后，文件再次被关闭（行 160 至 180）。

ERASE

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

描述: ERASE- 删除文件

删除当前目录中的文件。

|     |                    |                                                                  |
|-----|--------------------|------------------------------------------------------------------|
| 语法: | ERASE (<FileName>) |                                                                  |
|     | <FileName>         | 具有作为字符串表达式的完整路径的文件名。<br>如果没有路径指定，则会在当前目录中搜索文件。CPL 命令以整数值形式提供返回值： |

表格 7-262: 删除语法

“ERASE”命令可为已分配整数变量或为循环或查询（WHILE、IF 等）以整数值形式返回以下值：

0: 文件被删除

1: 文件未被删除，因为它不存在

2: 文件未被删除，因为该文件受删除保护

3: 文件未被删除，因为该文件已激活

4: <文件名称>未被删除，因为<文件名称>不是文件，而是目录

5: 文件未被删除，因为该文件开启

如果无法删除文件，则会对此发布警告，并继续执行程序。

CPL 命令

示例:

```
10 IF ERASE("P1") <> 0 THEN ...
10 I% = ERASE("P1")
10 WHILE ERASE("P1") <> 0 DO ...

10 OPENW(1,"P2",200)
11 OPENW(2,"P3",200)
20 PRN#(1,"TEST1 FUER ERASE")
21 PRN#(2,"TEST2 FUER ERASE")
31 CLOSE(1) 32 CLOSE(2)
40 ERASE("P2")
43 A$="P3"
44 ERASE(A$)
```

FILECOPY

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

描述:

**FILE COPY** - 复制文件

FILECOPY 用于复制文件至 CPL 程序。

语法:

|                                     |                                                |                |
|-------------------------------------|------------------------------------------------|----------------|
| FILECOPY(<FileSource>,<FileTarget>) |                                                |                |
| <SourceFile>                        | 具有作为字符串表达式的源文件完整路径的文件名。如果没有路径指定，则会在当前目录中搜索文件。  |                |
| <TargetFile>                        | 具有作为字符串表达式的目标文件完整路径的文件名。如果没有路径指定，则会在当前目录中创建文件。 |                |
| ERRNO                               | 可在任何位置编入的 CPL 变量。当 ERRNO 编程时，不输出运行时错误。其返回值为：   |                |
|                                     | 0:                                             | 访问正常           |
|                                     | -8:                                            | 包括路径在内的源文件太长   |
|                                     | -9:                                            | 无法访问源文件        |
|                                     | -10:                                           | 包括路径在内的目标文件名太长 |
|                                     | -11:                                           | （源或目标）文件名无效    |
|                                     | -12:                                           | 复制不可能          |

表格 7-263: FILECOPY 语法

示例:

程序

```
10 FILECOPY("/usr/user/Test.txt", "/usr/user/Test.bak", ERRNO)
20 IF ERRNO = 0 THEN
30 PRN#(0,"copying ok")
40 ENDIF
```

7.5.4

编辑文件

PRN#

描述

**PRiNt#** - 写入文件

变量类型可以自由选择。索引变量和字符数组也可以使用。双精度实数表达式也可以编程为任何 CPL 表达式。

如果要输出表达式的结果以提供格式，则至少有一个表达式必须是“STRING”类型。格式可以用“#”和“.”在这种格式字符串指定。结果将输入到用“#”指定的格式指令的位置。STRING 表达式中的第一个格式指令指的是可以用格式规范输出的第一个后续表达式。布尔表达式不能被格式化。所有编程格式条目的数量必须小于或等于要输出的表达式数量。如果这个条件没有满足，那么显示不必要的“#”。如果没有格式说明，则表达式将以标准格式输出。

|            | 默认格式     | 地点数目   |
|------------|----------|--------|
| INTEGER 数字 | 9 个地方    | 最大 9。  |
| 单精度实数      | 四位前三位小数位 | 最大 7。  |
| 双精度实数      | 九位前六位小数位 | 最大 15。 |

如果表达式的输出超过 1024 个字符，则会显示以下错误消息：“BLOCK EXCEEDS 1024 BYTES”（块超过 1024 字节）。

如果不能以指定格式输出结果，则返回警告“PRN 格式不正确”返回，并且输出“\*”字符而非错误格式。

如果要在文件本身中创建“#”字符，则在 PRN# 指令中的字符串后不会跟有可格式化的表达式。也可用 CHR\$(35) 来执行“#”字符的输出。

可在使用 CHR\$(13) 执行输出期间启动换行，即，PRN# 命令的进一步输出在下一行（下一记录）中继续。

可用 CHR\$() 函数转换其他控制字符，如当通过串行接口输出时。

语法:

| PRN# (<n>, [<Expression>][, <Expression>][, <Expression>][, ... ] [;]) |                                                                                                                                                                                                                     |
|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <n>                                                                    | 1 至 9:<br>要用于写入的逻辑文件编号。<br><br>0:<br>输出被转移到监视器（如通过 MSG 命令编程的消息）。                                                                                                                                                    |
| <表达式>                                                                  | 要对其内容进行保存/显示的任何字母数字字符（引号中的文本）、格式化字符串或变量。                                                                                                                                                                            |
| ;                                                                      | 禁止 <CR><LF> 的自动附加。<br><br>如果通过 PRN# 命令对记录进行了重写，则以下内容适用：<br><b>带有分号的 PRN# 命令:</b><br><br>如果要写入的新数据的长度小于旧数据的长度，则新数据被插入并且旧数据的剩余部分被保留。<br><br><b>不带分号的 PRN# 命令:</b><br><br>如果要写入的新数据的长度小于旧数据的长度，则新数据被插入并且旧数据的剩余部分被空格重写。 |

表格 7-264: PRN# 语法

示例: 带分号的 PRN# 命令

## CPL 命令

```

1 OPENW(2,"PProg123.PRG",200,35)
2 PRN#(2,"TEST1 FOR PRN COMMAND WITH SEMICOLON")
3 PRN#(2,"TEST2 FOR PRN COMMAND WITH SEMICOLON")
4 PRN#(2,"TEST3 FOR PRN COMMAND WITH SEMICOLON")
6 SEEK(2,1)
7 PRN#(2,"OVERWRITE";)
8 CLOSE(2)

```

PProg123.PRG 中的结果:

```

OVERWRITE COMMAND WITH SEMICOLON <LF>
TEST2 FOR PRN COMMAND WITH SEMICOLON <LF>
TEST3 FOR PRN COMMAND WITH SEMICOLON <LF>
<ETX><LF>

```

**不带分号的 PRN# 命令**

```

1 OPENW(2,"P2",1000,36) 1 REWRITE(2)
2 PRN#(2,"TEST1 FOR PRN COMMAND WITHOUT SEMICOLON")
3 PRN#(2,"TEST2 FOR PRN COMMAND WITHOUT SEMICOLON")
4 PRN#(2,"TEST3 FOR PRN COMMAND WITHOUT SEMICOLON")
6 SEEK(2,1)
7 PRN#(2,"OVERWRITE")
8 CLOSE(2)

```

P2 中的结果:

---

```

OVERWRITE <LF>
TEST2 FOR PRN COMMAND WITHOUT SEMICOLON <LF>
TEST3 FOR PRN COMMAND WITHOUT SEMICOLON <LF>
<ETX><LF>

```

<ETX><LF> 被插入文件最后一个块的后面。

如果块的长度超过 1024 个字符，则会出现以下错误消息：“BLOCK EXCEEDS 1024 BYTES”（块超过 1024 字节）。

如果顺序文件被写并且到达文件末端，则文件会自动复制，并且保留范围会增加被分配的长度，如果零件程序内存中有足够的内存可以使用。

由于这会极快地占用大量内存，因此，使用“OPENW”创建文件时，建议保留足够大的文件长度。

```

1 OPENW(1,"P2",300,"TEST PRN COMMAND")
2 A$="TEST"
3 B$="FOR"
4 C$="PRN COMMAND"
5 PRN#(1,A$)
6 PRN#(1,B$)
7 PRN#(1,C$)
8 PRN#(1,A$;)
9 PRN#(1,B$;)
10 PRN#(1,C$;)

```

```
11 CLOSE (1)
结果:
P2:
TEST<LF>
FOR<LF>
PRN COMMAND<LF>
TESTFORPRN COMMAND<LF><ETX><LF>

10 DIM E$(50)
20 OPENW (1,"P2",300,"TEST2")
30 A% = 5000
40 R = 1.231
50 B! = 4/3
60 D$ = "ABCDE"
70 E$ = "CDEFGHI"
80 PRN# (1,"10";)
90 PRN# (1,"#####","###.###","#.#####",A%,R,B!,D$,E$)
95 CLOSE (1)
P2:
10 5000 1.2311.33333ABCDECDEFGHI<LF>
 A% R B! D$ E$
```

LJUST, NJUST

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** Left JUSTify - 左对齐格式化  
无对齐 - 无格式化  
有 LJUST , 它被切换到左调整数据输出。一直到程序运行结束, 所有的数据输出都是有效的。  
通过 NJUST, 可以将其切换回格式化输出。  
数据输出到文件时, “实数”数据类型最多可用 7 位 (小数点前 4 位和小数点后 3 位), “实数”数据类型最多可用 9 位。前导零和后缀零被禁止。这也适用于左调整输出。  
由于数控地址和值之间的空格被禁止, 所以 LJUST 可利用 CPL 直接创建数控程序, 该程序可在 “PROCESSING” 模式下执行。

INP#

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** INPut# - 从文件中读取  
用 INP# 指令, 可以记录格式对打开文件中的 ASCII 数据进行读取, 并将其分配给一个或多个变量。该命令仅对用 “OPENR(..)” 打开的文件有效。

|                                                        |                         |
|--------------------------------------------------------|-------------------------|
| <b>语法:</b> INP# (<n>,<Variable>[,<Variable>][,...][;]) |                         |
| <n>                                                    | 1 至 9:<br>要用于读取的文件逻辑编号。 |

CPL 命令

|            |                                                                               |
|------------|-------------------------------------------------------------------------------|
| <Variable> | 用于对读取数据进行储存的变量。                                                               |
| ;          | 如果编程分号，则文件指针会在记录中一直保持到到达记录末端。下一个记录随后打开。然而，阅读不会自动发生。<br>如果没有编程分号，它会自动切换到下一条记录。 |

表格 7-265: INP# 语法

变量类型可以自由选择。索引变量和字符数组也可以使用。如果将除 TRUE 或 FALSE 之外的值分配给逻辑变量，则该变量的值为 "NUL"。

变量类型整数型或实数型（简单 - 或双精度）、字符 "0" .. "9"、前导符号 "-"、 "+"、 前导零或空格均被转换成整数型或实数型值。如果将其他字符分配给整数型或实数型变量，则变量被分配为 "NUL" 。如果将 "NUL" 分配给变量，则文件内的位置不会更改。

如果分配给整数型或实数型变量的值太高，则会出现以下对应错误消息：

"INVALID INTEGER VALUE"

"INVALID FLOAT VALUE"

示例： INP# 指令

```
P2:
ABC 123456789 ABC
P3:
1 OPENR(2,"P2") 2 DIM C$(3) 3 DIM D$(3) 4 INP#(2,I
%,J,L?,C$,K%,D$) 5 CLOSE(2)
结果:
I% = NUL
J = NUL
L? = NUL
C$ = "ABC"
K% = 123456789
D$ = "ABC"
```

从文件读取记录

```
1 OPENW(1,"P2",200,"Test",22)
2 PRN#(1,"-12TEST1.23V12ABCD2.4A")
3 PRN#(1,"-12TEST1.23V12ABCD2.4A")
4 PRN#(1,"-12TEST1.23V12ABCD2.4A")
5 CLOSE(1)
6 DIM A$(3)
7 DIM C$(5)
8 DIM D$(4)
9 DIM E$(4)
10 DIM G$(25)
11 DIM H$(7)
12 DIM I$(7)
13 DIM J$(25)
14 DIM R(1,2)
```

```
15 OPENR(2,"P2",22)
16 INP#(2,B%,D$,R(1,1),MID$(E$,1,1),R(1,2),A$,C$)
17 INP#(2,G$)
18 INP#(2,H$;)
19 INP#(2,I$;)
20 INP#(2,J$)
21 CLOSE(2)
```

结果:

```
B% = -12
D$ = "TEST" , 因为字符数组的最大长度 = 4
R(1,1) = 1.230
E$ = "V"
R(1,2) = 12.000
A$ = "ABC" , 因为字符数组的最大长度 = 3
C$ = "D2.4A"
G$ = "-12TEST1.23V12ABCD2.4A"
H$ = "-12TEST" , 因为字符数组的最大长度 = 7
I$ = "1.23V12" , 因为字符数组的最大长度 = 7
J$ = "ABCD2.4A"
```

SEEK

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| --- | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|

**描述:** **SEEK** - 位置文件指针

定位打开文件的某个位置处的文件指针。文件可以是顺序文件或随机文件。对于顺序文件，必须用命令 "OPENR(..)" 打开。对于随机文件，也可以使用命令 "OPENW(..)" 。

|            |                                                                                      |
|------------|--------------------------------------------------------------------------------------|
| <b>语法:</b> | SEEK(<n>,<k>[,<o>])                                                                  |
|            | <n> 要在该处定位文件指针的文件的逻辑编号。<br>取值范围: 1 到 9<br>如果不属于该值范围，则会出现错误消息 "INVALID FILE NUMBER" 。 |

CPL 命令

|     |                                                                                                                                                                                                                         |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <k> | 随机文件的记录编号或顺序文件的字节数。文件指针定位在 <k> 上。<br>取值范围：<br>0 至最后存在的记录或<br>0 至 最后存在的字节。<br>具有 EOF 指针的记录成为最后存在的记录。在 0 处，定位在 EOF 指针上。<br>如果不属于该值的范围或如果指定记录不存在，则会出现错误消息 “INVALID COMPONENT”（无效元素）。                                       |
| <o> | 记录偏置<br>指示记录中的哪个字节将被定位。<br>取值范围：<br>1 ... 记录长度 + 1<br>如果没有为随机文件对记录偏置编程，文件指针定位在记录 <k> 的第一个字节处。<br>如果不属于该值范围，则会出现错误消息 “INVALID PARAMETER”。<br>该参数仅允许用于随机文件。如果该参数已编程，但其为顺序文件（已被打开用于读取），则会出现错误消息 “INVALID PARAMETER”（无效参数）。 |

表格 7-266:     *SEEK* 语法

示例:     **SEEK 和顺序文件**

```
1 DIM A$(1):LJUST:OPENW(1,"P271",130,"TEST"):FOR I%=1
 TO 10:
 PRN#(1,"!/-/!/-/!/-/!/-/!/-/!/-/!/-/!/"):NEXT:
 CLOSE(1):OPENR(2,"P271"):FOR I%=1 TO
 FILESIZE(2,2)-28: IF NOT (EOF(2)) THEN SEEK(2,I
 %):INP#(2,A$) ENDIF: IF (EOF(2)) THEN PRN#(0,"###",I
 %,".BYTE: <EOF>"): ELSE PRN#(0,"###",I%,".BYTE: <","A
 $,">") ENDIF: NEXT I%:CLOSE(2)

M30
```

**SEEK 和随机文件**

```
1 OPENW(1,27272,200,"TEST",1024):LJUST
2
3 FOR I%= 1 TO 10
4
5 PRN#(1,I%,".Record")
6
7 NEXT
8
9 SEEK(1,3,4) : 第四个字节位于第三个记录中的 REM
10 PRN# (1, "用第 4 个字节开始覆盖第三条记录")
11
12 SEEK(1,11):PRN#(1,"11.Record")
13
14 SEEK(1,11,5):PRN#(1,"@@")
15
16 SEEK(1,0):PRN#(1,"<EOF>")
17
18 SEEK(1,0,1):PRN#(1,"新的 <EOF>")
19
20 CLOSE(1)
```

FILEPOS

|     |      |       |       |      |      |       |   |
|-----|------|-------|-------|------|------|-------|---|
| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|



**描述:** **FILE POSition** – 确定文件指针位置

“FILEPOS()”命令返回随机文件的当前记录号。该记录可以随后访问。也可以确定随机文件当前记录中的记录偏置，或确定顺序文件中可访问的当前字节的偏置。因此，文件可以是顺序文件或随机文件。

偏置是指从文件头一直到文件当前字节的字节数。记录偏置指定在记录中进行定位时该处的字节。记录偏置以值 1 (= 记录中的第一个字节) 开头，并且其最大值为记录长度 + 1 (该记录最后一个字节为 <LF>)。在 EOF 指针处，将返回值 1。

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FILEPOS (<n>[, <Mode>]) |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <n>                     | 1 至 9:<br>要在该处文件指针位置的文件逻辑编号。<br><br>如果不属于该值范围，则会出现错误消息 “INVALID FILE NUMBER”。                                                                                                                                                                                                                                                                                                                                                  |
| <mode>                  | <b>对于随机文件:</b> 值范围 1 to 3<br><br><mode> = 1:<br>提供可读写当前字节的偏置。<br><br><mode> = 2:<br>提供可读写的当前记录的记录编号。在 EOF 指针处，其结果如下：<br>记录编号 + 1。<br><br><mode> = 3:<br>提供可读写的当前记录的记录偏置。记录偏置以值 1 (⇒ 该记录中的第一个字节) 开头，并且其最大值为记录长度 + 1 (⇒ 该记录中最后一个字节为 <LF>)。<br><br>1 被返回并且如果位于 EOF 指针上，则从该文件读取是无效的。<br><br><mode> 未编程:<br>提供可读写的当前记录的记录编号。<br><br>在 EOF 指针处，其结果如下：记录编号 + 1。<br><br><b>对于顺序文件:</b> 取值范围 1<br><br><mode> = 1 或未编程:<br>提供可读写当前字节的偏置。 |

表格 7-267: FILEPOS 语法



如果不属于 <mode> 的值范围，则会出现以下错误消息：“INVALID PARAMETER”（无效参数）。

**示例:** **FILEPOS 和顺序文件**

```
1 OPENW (1, "P2", 200, "TEST")
2 FOR I%= 1 TO 10
3 PRN# (1, "TEST FOR FILEPOS")
4 NEXT
5 CLOSE (1)
6 OPENR (1, "P2")
7 SEEK (1, 3)
8 POS% = FILEPOS (1)
9 POS1% = FILEPOS (1, 1)
```

CPL 命令

```
11 SEEK(1,0) : REM 定位于文件结束处
12 POS2% = FILEPOS(1)
13 POS3% = FILEPOS(1,1)
14 CLOSE(1)
```

结果:

|             |       |
|-------------|-------|
| POS% = 3    | → 字节数 |
| POS1% = 3   | → 字节数 |
| POS2% = 171 | → 字节数 |
| POS3% = 171 | → 字节数 |

FILEPOS 和随机文件

程序

```
1 OPENW(1,"P2",200,"TEST",1024)
2 FOR I% = 1 TO 10
3 PRN#(1,"TEST FOR FILEPOS")
4 NEXT
5 SEEK(1,3,2)
6 POS% = FILEPOS(1)
7 POS1% = FILEPOS(1,1)
8 POS2% = FILEPOS(1,2)
9 POS3% = FILEPOS(1,3)
10 PRN#(1,"Overwrite 3rd record from byte 2 with this text")
11 SEEK(1,0) : REM Positioned at end of file
12 POS% = FILEPOS(1)
13 POS1% = FILEPOS(1,1)
14 POS2% = FILEPOS(1,2)
15 POS3% = FILEPOS(1,3)
16 CLOSE(1)
```

结果:

|              |               |
|--------------|---------------|
| POS% = 3     | → 当前位置记录的记录编号 |
| POS1% = 258  | → 字节数         |
| POS2% = 3    | → 当前位置记录的记录编号 |
| POS3% = 2    | → 在第三个记录中的位置  |
| POS% = 11    | → 当前位置记录的记录编号 |
| POS1% = 1281 | POS1% = 1281  |
| POS2% = 11   | → 当前位置记录的记录编号 |
| POS3% = 1    | → 在第三个记录中的位置  |

EOF

|      |      |       |       |      |      |       |   |
|------|------|-------|-------|------|------|-------|---|
| BOOL | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|

描述:

文件末端 - 文件终止 ID  
“EOF” 命令可用于查询是否到达文件末尾。  
读取时，如果到达文件末端，则 EOF 函数返回逻辑值 “TRUE”。否则，将返回 “FALSE”。

示例:

: 9 DIM A\$(10)  
10 OPENR(1,"P",444) : I%=0  
11 WHILE NOT (EOF(1)) DO

```
12 INP#(1,A$)
13 I%=I%+1
14 END
15 CLOSE(1)
M30
```

7.5.5 文件属性

FILEACCESS

**描述:** FILE ACCESS - 访问权限文件

使用 FILEACCESS，可以在 CPL 程序中确定文件是否存在以及分配了哪些访问权限。

语法:

| FILEACCESS (<FileName>) |                                                                  |
|-------------------------|------------------------------------------------------------------|
| <FileName>              | 具有作为字符串表达式的完整路径的文件名。<br>如果没有路径指定，则会在当前目录中搜索文件。CPL 命令以整数值形式提供返回值: |
|                         | -1 : 文件不存在                                                       |
|                         | 0: 文件没有访问权限                                                      |
|                         | 否则: 二进制编码访问权限:                                                   |
|                         | Bit0 : 可以执行 (X)                                                  |
|                         | Bit1 : 允许写入 (W)                                                  |
|                         | Bit2 : 允许读取 (R)                                                  |
|                         | Bit3 : 文件是目录 (D)                                                 |
|                         | Bit4 : 文件是激活的程序 (A)                                              |

表格 7-268: FILEACCESS 语法

**激活程序** 是一个文件

- 作为通中的程序执行
- 作为程序的子程序执行并且
- 通过 CPL 命令打开。

如果使用 CPL 命令 "FILEACCESS"对零点偏置或刀具修正表（用于运行的零件程序）的 **访问权限** 进行查询，则 Bit5 未 被设置。

**示例:** 10 I% = FILEACCESS ("/usrfep/test.cnc")

FILEDATE

**描述:** FILE DATE - 日期

在 CPL 程序中，可用 FILEDATE 确定文件的日期。

访问错误不会生成零件程序错误。该命令提供了一个空字符串。

CPL 命令

|     |                                         |                                                                     |
|-----|-----------------------------------------|---------------------------------------------------------------------|
| 语法: | FILEDATE (<FileName>[,<DataSelection>]) |                                                                     |
|     | <FileName>                              | 具有作为字符串表达式的完整路径的文件名。如果没有路径指定，则会在当前目录中搜索文件。CPL 命令提供字符串表达式作为返回值。      |
|     | <DataSelection>                         | 命令模式的整数型变量（默认 = 1）：<br>1 = 文件日期，格式：tt:mm:jj<br>2 = 文件时间，格式：hh:mm:ss |

表格 7-269: FILEDATE 语法

|     |    |                                             |
|-----|----|---------------------------------------------|
| 示例: | 10 | DIM DATUM\$ (10)                            |
|     | 20 | DATUM\$ = FILEDATE ("/usr/user/Test.txt",1) |
|     | 30 | IF LEN (DATE\$)>0 THEN                      |
|     | 40 | PRN# (0,"File date: ",DATUM\$)              |
|     | 50 | ENDIF                                       |
|     |    |                                             |

FILESIZE

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

**描述:** FILE SIZE - 文件大小

提供文件大小或已经写入文件的限制。文件可以是顺序文件或随机文件。该命令仅对用 "OPENR(..)" 打开的文件有效。

|     |                      |                                                                                                                                                                                                                                                                                                                                               |
|-----|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 语法: | FILESIZE (<n>[,<k>]) |                                                                                                                                                                                                                                                                                                                                               |
|     | <n>                  | 1 到 9<br>要确定其大小文件的逻辑编号。如果不属于该值的范围，则会出现以下错误消息："NVALID FILE NUMBER"。                                                                                                                                                                                                                                                                            |
|     | <k>                  | <b>对于随机文件:</b><br>值范围 1 to 4<br><b>对于顺序文件:</b><br>值范围 1 to 2<br><k> = 1:<br>文件使用的总内存区大小（以字节计）。<br><k> = 2:<br>从数据范围开始到 EOF 指针所使用的内存区大小（单位字节）（不包括 EOF 指针的大小）。<br><k> = 3:<br>文件中记录的最大数目。这个结果取决于用来打开文件的记录长度。<br><k> = 4:<br>从文件开始到 EOF 指针的记录数目。这个结果取决于用来打开文件的记录长度。<br><k> 未编程：如<k> = 1。<br>如果不属于 <k> 的值范围，则会出现以下错误消息："INVALID PARAMETER"（无效参数）。 |

表格 7-270: FILESIZE 语法

**示例:** FILESIZE 和顺序文件

整数型变量 C% 包括值: 300

```

1 OPENW(1,"P2",1000,10)
2 FOR I%= 1 TO 20
3 PRN#(1,"TESTE FILESIZE")
4 NEXT
5 CLOSE(1)
6 OPENR(2,2,10)
7 A%=FILESIZE(2)
9 B%=FILESIZE(2,1)
10 C%=FILESIZE(2,2)
10 D=FILESIZE(2,3)
10 E%=FILESIZE(2,4)
11 CLOSE(2)

```

整数型变量 E% 的值为: 20

**DIM**

**描述:** 标注 - 尺寸标注字符串

CPL 命令

要创建字符数组，必须使用 指令对字符变量编索引。可以创建多达 1,024 个字符的字符数组（索引值范围：1 至 1024）。如果不属于该值范围，则会出现错误消息 “INVALID ARRAY LIMIT”。

**示例：**  
1 DIM VWX\$(14)  
在该示例中，创建了包括 14 个单独字符变量的字符数组 VWX\$。因此，可将长度最多为 14 个字符的字符串储存在 VWX\$ 中。

|                  |                        |
|------------------|------------------------|
| 1 DIM ABC\$(1)   | 最大长度为 1 个字符的字符串的字符数组。  |
| 2 DIM BCDE\$(10) | 最大长度为 10 个字符的字符串的字符数组。 |

INSTR

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

**描述** 在字符串中 – 在字符串中搜索  
使用指定的字符序列搜索字符串。

|            |                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>语法：</b> | <div>INSTR (&lt;CharacterString&gt;,&lt;StringExpression&gt;[,&lt;StartingPoint&gt;])<br/><br/>从 &lt;StartingPoint&gt; 开始在&lt;StringExpression&gt;中搜索 &lt;CharacterString&gt;，并将 &lt;StringExpression&gt; 中找到的 &lt;CharacterString&gt; 的第一个字符的位置返回为 INTEGER 值。<br/>如果 &lt;CharacterString&gt; 未找到，则返回值 0。The &lt;CharacterString&gt; 可以编程为一个字符串表达式。<br/>第三个参数的值范围介于 1 到 1,024 之间。如果不属于该值范围，则会出现错误消息 “INVALID PARAMETER”。</div> |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

表格 7-271: INSTR 语法

**示例：**  
1 DIM A\$(8)  
2 DIM B\$(16)  
3 A\$="A" : MID\$(A\$,2)="UVWXYZ"  
4 B\$="ABCDEF UVWXYZ GH"  
5 POS1%=INSTR(MID\$(A\$,2),B\$,4)      整数型变量 POS1% 包括值 8。  
6 POS2%=INSTR(MID\$(A\$,2,4),B\$,10)      整数型变量 POS2% 包括值 0。  
7 POS3%=INSTR(MID\$(A\$,2),B\$)      整数型变量 POS3% 包括值 8。

LEN

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

**描述：** 长度 – 字符串的长度

|            |                                                                                                                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>语法：</b> | <div>LENLEN (&lt;StringExpression&gt;)<br/><br/>提供 &lt;StringExpression&gt;中的字符数目。<br/>结果为 整数 值。<br/>如果 &lt;StringExpression&gt; 是空的，LEN 返回 0。<br/>如果 &lt;StringExpression&gt; 未定义，LEN 返回 -1。</div> |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

表格 7-272: LEN 语法

**示例：**  
1 DIM XYZ\$(10)  
2 XYZ\$="ABC"

```
3 I%=LEN(XYZ$) 整数型变量 I% 的值为 3。
4 XYZ$=""
5 J%=LEN(XYZ$) 整数型变量 J% 的值为 0。
6 XYZ$=NUL
7 K%=LEN(XYZ$) 整数型变量 K% 包括值 -1。
```

比较字符串表达式

**描述:** 也可以对 STRING 表达式以及 STRING 常量、STRING 变量和字符数组进行相互比较。

**示例:**

```
1 DIM A$(10)
2 DIM B$(15)
3 A$="ANTON"
4 B$= "WILLI"
5 C$="ABCDE"
6 D$="VWXYZ"
7 IF A$ < B$ THEN ...
8 IF MID$(A$,2,3) = MID$(B$,1,3) THEN ...
9 Z?=A$ <> "TEST"
10 IF "A" <= "C" THEN ...
11 IF C$ > D$ THEN ...
12 IF A$ = C$ THEN ...
13 IF "TE" < MID$(D$,2,2) THEN ...
```

根据个别字符的 ASCII 代码，对字符串表达式内容的字母顺序进行检查。

7. 6. 2 类型转换  
ASC

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

**描述:** ASCi i – 确定一个字符的 ASCII 码

|     |  |                                                                                                                                 |
|-----|--|---------------------------------------------------------------------------------------------------------------------------------|
| 语法: |  | ASC(<CharacterString>)                                                                                                          |
|     |  | 将 <CharacterString> 的第一个字符的序数（ASCII 代码）提供为整数型值。<br>如果 <String> 为空或未定义，则 ASC 返回值 -1。<br><String>必须是一个字符串表达式。<br>ASC 是 CHR\$ 的逆转。 |

表格 7-273: ASC 语法

**示例:**

```
10 DIM A$(1)
20 A$ = "ABC"
30 B$ = "BCD"
40 I% = ASC(A$) 整数型变量 I% 包括值 65。
50 J% = ASC(B$) 整数型变量 J% 包括值 66。
60 A$ = ""
```

CPL 命令

```
70 K% = ASC (A$) 整数型变量 K% 包括值 -1。
80 A$ = NUL
90 L% = ASC (A$) 整数型变量 L% 包括值 -1。
```

CHR\$

| CHAR | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

描述: CHaRacter - 确定 ASCII 字符  
CHR\$ 是 ASC 的回动命令。

|     |                                         |
|-----|-----------------------------------------|
| 语法: | CHR\$ ( <IntegerExpression> )           |
|     | 将 <integer expression> 转换成各自的 ASCII 字符。 |

表格 7-274: CHR\$ 语言  
所有 ASCII 字符十进制优先级在本手册附录的“ASCII 字符块”中均有描述。

示例:

```
10 DIM A$ (1)
20 I% = 65
40 A$ = CHR$ (I%) 整数型变量 A% 包括值 65。
```

STR\$

| BOOL | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

描述: STRing - 将数字转换为字符串  
语法:

|                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STR\$ ([<FormatString>,<Value>)                                                                                                                                                      |
| 把数值表达式 <Value> 转变为字符串，而且只能分配给字符串组。如果分配给字符串变量，则会导致运行时错误。<br><Value> 可以是单精度或双精度的整数型或实数型表达式。<br>如果 <Format string> 已编程，则可用格式化的形式提供字符串。符号“#”表示数字而“.”表示小数点。<br>以不带 <FormatString>的标准格式提供。 |

表格 7-275: STR\$ 语法

|            | 默认格式     | 地点数目   |
|------------|----------|--------|
| INTEGER 数字 | 9 个地方    | 最大 9。  |
| 单精度实数      | 四位前三位小数位 | 最大 7。  |
| 双精度实数      | 九位前六位小数位 | 最大 15。 |

示例:

```
10 DIM A$ (50)
20 DIM B$ (21)
30 A$=STR$ ("Zahl=##.###", 37/3) 字符数组 A$ 的 REM 内
 容: "Number=12.333"
40 B$=STR$ (2.5) 字符数组 B$ 的内容: "2.500"
```



VAL

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

描述

VALue – 将数字转换为字符串

语法:

|                         |                                                                                                                                                                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VAL(<StringExpression>) |                                                                                                                                                                                                                                                           |
|                         | 返回 <StringExpression>的数值。如果字符串除了前导空格、前导 “+” 或 “-” 号、数字 0 至 9 或小数点 “.” 之外，还包含字符，则将对这个（其它）字符以及之前的字符进行转换。为了形成值，前导空格和零将被忽略。<br><br>如果上述字符中并未出现，则返回 “NUL”（空字符）。<br><br>如果字符串包含小数点，则所得结果只能被分配给实数型或双精度实数型变量。在这种情况下，如果将其分配给 整数型 变量，则会导致错误消息 “INVALID ASSIGNMENT”。 |

表格 7-276: VAL 语法

示例:

```
1 I% = VAL("1.23DE")
2 K% = VAL("123DE")
3 J% = VAL("ABC")
4 R = VAL("-1.23DE")
5 Z = VAL("+ 000001234TEST4365")
6 X = VAL("ABC1.23DE")
7 D! = VAL("1234567.234567")
```

行 1 导致错误消息，这是因为执行整数型变量的分配。  
实数型变量 K% 的值为 123。  
数字 1, 2, 3 被转换成整数型数字。  
字符 “D” 退出转换，因为它不属于 整数型 数字。未对以下字符进行进一步处理。  
整数型变量 “J%” 的值为 NUL，即变量未被分配。字符 “A” 中断 <StringExpression>的处理。  
实数型变量 “R” 的值为 -1.23。字符 “-” 被检测为实数型数字的正/负号。数字 1、字符 “.” 以及数字 2 和 3 被转换成实数型数字。字符 “D” 退出转换，因为它不属于实数型数字。未对字符 “E” 进行进一步处理。  
实数型变量 “R” 的值为 1234。字符 “+” 被检测为实数型数字的正/负号。为了形成值，后面的空格以及前导零将被忽略。数字 1、 2、 3 和 4 被转换成实数型数字。字符 “T” 退出转换，因为它不属于实数型数字。未对剩余字符进行进一步处理。  
实数型变量 “X” 为 NUL，即未被分配。当识别为字符 “A” 时，转换中断。双精度实数型变量 “D!” 的值为 1234567.234567。

7.6.3 更改字符串

将字符串表达式分配给字符数组

描述:

如果字符串表达式包含的字符少于字符数组的选定范围，则将剩余范围视为未分配。该剩余范围不包括在字符数组长度内。

示例:

```
1 DIM XYZ$(16)
2 XYZ$=" " Content of the character array XYZ$: " "
```

字符数组的长度 XYZ\$: 1

CPL 命令

如果在分配过程中，字符串表达式的长度超过字符数组的最大长度，则多余的字符会被放弃。

```
1 DIM XYZ$ (3)
2 XYZ$="ABCDEF"
字符字段的内容 XYZ$: "ABC"
字符数组的长度 XYZ$: 3 ---> 最大长度
```

```
1 DIM XYZ$ (16)
2 A$="DAS "
3 B$="IS A TEST"
4 C$="An Egg"
5 MID$ (XYZ$,1,4)=A$
```

**字符数组的内容**  
*程序*

|            |   |   |   |                   |   |   |   |   |    |    |    |    |    |    |    |
|------------|---|---|---|-------------------|---|---|---|---|----|----|----|----|----|----|----|
| 1          | 2 | 3 | 4 | 5                 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| D          | A | S |   |                   |   |   |   |   |    |    |    |    |    |    |    |
| length = 4 |   |   |   | field is not used |   |   |   |   |    |    |    |    |    |    |    |

```
6 MID$ (XYZ$,5,6)=B$
字符数组的内容
程序
```

|             |   |   |   |   |   |   |   |   |    |                   |    |    |    |    |    |
|-------------|---|---|---|---|---|---|---|---|----|-------------------|----|----|----|----|----|
| 1           | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11                | 12 | 13 | 14 | 15 | 16 |
| D           | A | S |   | I | S | T |   | E | I  |                   |    |    |    |    |    |
| length = 10 |   |   |   |   |   |   |   |   |    | field is not used |    |    |    |    |    |

```
7 MID$ (XYZ$,5,12)=B$
字符数组的内容
程序
```

|                                       |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |
|---------------------------------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| 1                                     | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| D                                     | A | S |   | I | S | T |   | E | I  | N  |    | T  | E  | S  | T  |
| length = 16; field is used completely |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |

```
8 MID$ (XYZ$,9,8)=C$
字符数组的内容
程序
```

|  |                                       |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |
|--|---------------------------------------|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|
|  | 1                                     |  | 2 |  | 3 |  | 4 |  | 5 |  | 6 |  | 7 |  | 8 |  | 9 |  | 10 |  | 11 |  | 12 |  | 13 |  | 14 |  | 15 |  | 16 |  |
|  | D                                     |  | A |  | S |  |   |  | I |  | S |  | T |  |   |  | E |  | I  |  | N  |  |    |  | T  |  | E  |  | S  |  | T  |  |
|  | length = 16; field is used completely |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |

9 XYZ\$=MID\$(XYZ\$,1,4)

字符串组的内容

程序

|  |                                   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |
|--|-----------------------------------|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|
|  | 1                                 |  | 2 |  | 3 |  | 4 |  | 5 |  | 6 |  | 7 |  | 8 |  | 9 |  | 10 |  | 11 |  | 12 |  | 13 |  | 14 |  | 15 |  | 16 |  |
|  | D                                 |  | A |  | S |  |   |  |   |  |   |  |   |  |   |  |   |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |
|  | length = 4      field is not used |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |

字符串组被剪切。

链接 STRING 表达式

**描述:** 可通过 “+” 加字符连接多个 STRING 表达式。结果必须分配给字符串组。链接 STRING 表达式的嵌套深度为 3。如果超过此深度，则显示以下错误消息: “RUNTIME ERROR 2153 - NESTING TOO DEEP” (运行时错误 2153 - 嵌套太深)。

**示例:** 还可以在 CPL 命令中链接

```
1 DIM A$(3)
2 DIM B$(3)
3 A$ = "ABC"
4 B$ = "DEF"
5 C$ = "GH"
6 D$ = "JKL"
7 OPENW(1,"P2",130,"TEST LINKING",10)
8 PRN#(1,A$+B$)
9 PRN#(1,A$+C$)
10 PRN#(1,C$+D$)
11 PRN#(1,A$+C$+"TEST")
12 PRN#(1,"UVW"+"XYZ")
13 CLOSE(1)
```

文件 P2 的内容:

|           |               |
|-----------|---------------|
| ABCDEF    | <LF>          |
| ABCGH     | <LF>          |
| GHJKL     | <LF>          |
| ABCGHTEST | <LF>          |
| UVWXYZ    | <LF><ETX><LF> |

使用 STRING 表达式连接文本

```
10 DIM A$(100)
```

CPL 命令

```
20 DIM B$(100)
30 DIM C$(10)
40 DIM D$(20)
51 DIM E$(30)
52 DIM F$(30)
53 DIM G$(30)
54 DIM H$(30)
55 DIM I$(30)
60 A$="DAS "
70 B$="IST EIN TEST"
80 MID$(C$,1,6)=A$ + B$
90 MID$(D$,1,10)=MID$(A$,1,1) + MID$(B$,1,2)
92 E$=A$ + MID$(B$,1)
93 X$="ABC"
94 Y$="DE"
95 F$=X$ + Y$ 96 G$=X$ + A$
97 H$=X$ + A$ + "TEST"
98 I$="TES" + "T1"
```

|          |                  |       |
|----------|------------------|-------|
| A\$ 的内容: | "THIS "          | 长度 4  |
| B\$ 的内容: | "IS A TEST"      | 长度 12 |
| C\$ 的内容: | "THIS IS"        | 长度 6  |
| D\$ 的内容: | "DIS"            | 长度 3  |
| E\$ 的内容: | "THIS IS A TEST" | 长度 16 |
| F\$ 的内容: | "ABCDE"          | 长度 5  |
| G\$ 的内容: | "ABCDAS "        | 长度 7  |
| H\$ 的内容: | "ABCDAS TEST"    | 长度 11 |
| I\$ 的内容: | "TEST1"          | 长度 5  |

以下编程会导致错误:

|                                                |                     |
|------------------------------------------------|---------------------|
| 1 DIM A\$(3):A\$ = "ABC":B\$ = "CD":C\$ = "EF" |                     |
| 2 D\$ = A\$ + B\$                              | 无效赋值给无量纲的 STRING 变量 |
| 3 D\$ = B\$ + C\$                              |                     |
| 4 D\$ = A\$ + B\$ + "TEST"                     |                     |
| 5 D\$ = "TEST" + "TEST1"                       |                     |

STR\$

```
1 DIM A$(50) : DIM B$(21)
2 A$ = STR$("A$ = ##.###", (37/3)) : B$ = STR$(2.5)
```

|               |                 |           |         |
|---------------|-----------------|-----------|---------|
| 字符数组 A\$ 的内容: | "A\$ = 12.333"; | 字符数组 B\$: | "2.500" |
|---------------|-----------------|-----------|---------|

## VAL

```
1 DIM FOLGE$(20) : FOLGE$="X-VALUE -0001.234 MM"
2 XR=VAL(MID$(FOLGE$,7)) : Z%=VAL(MID$(FOLGE$,7,6))
3 Y%=VAL(MID$(FOLGE$,15,5)) : X%=VAL(MID$(FOLGE$,18))
```

---

|               |        |
|---------------|--------|
| 实数型变量 XR 的内容: | -1.234 |
| 整数型变量 Z% 的内容: | -1     |
| 整数型变量 Y% 的内容: | 34     |
| 整数型变量 X% 的内容: | NUL    |

## LEN

```
1 DIM Z$(10)
2 Z$ = "TEST"
3 S$ = "TEST"
4 A% = LEN("TEST")
5 B% = LEN(Z$)
6 C% = LEN(S$)
7 D% = LEN("TEST"+Z$+S$)
```

---

|               |    |
|---------------|----|
| 整数型变量 A% 的内容: | 4  |
| 整数型变量 B% 的内容: | 4  |
| 整数型变量 C% 的内容: | 4  |
| 整数型变量 D% 的内容: | 12 |

## 只读访问的 MID\$ 命令

```
10 DIM A$(4)
20 DIM B$(10)
30 DIM C$(10)
40 DIM D$(10)
50 DIM E$(10)
55 DIM F$(10)
60 A$ = "ABCD"
```

---

|                         |                 |
|-------------------------|-----------------|
| 70 B\$ = MID\$(A\$,2,2) | --> B\$ = "BC"  |
| 80 C\$ = MID\$(A\$,2,5) | --> C\$ = "BCD" |
| 95 E\$ = MID\$(A\$,5,1) | --> E\$ = NUL   |
| 97 F\$ = MID\$(A\$,2)   | --> F\$ = "BCD" |
| 98 F\$ = MID\$(F\$,1,1) | --> F\$ = "B"   |

## 写访问的 MID\$ 命令

---

```
10 DIM A$(4)
20 DIM B$(10)
```

CPL 命令

```
30 DIM C$ (10)
40 DIM D$ (10)
60 A$ = "ABCD"
70 B$ = "1234567890"
80 C$ = "EFGHIJKLMN"
85 D$ = A$ --> D$ = "ABCD"
90 MID$(D$,2,3) = B$ --> D$ = "A123"
95 MID$(D$,5,1) = C$ --> D$ = "A123E"
97 MID$(D$,4) = B$ --> D$ = "A121234567"
```

TRIM\$

```
1 DIM XYZ$ (16)
2 XYZ$ = "XWERT = 0.123 "
3 A$ = MID$(XYZ$,8)
4 B$ = TRIM$(MID$(XYZ$,8))
5 C$ = TRIM$(MID$(XYZ$,8),"L")
6 D$ = TRIM$(MID$(XYZ$,8),"R")
```

|                |          |
|----------------|----------|
| 字符串变量 A\$ 的内容: | " 0.123" |
| 字符串变量 B\$ 的内容: | "0.123"  |
| 字符串变量 C\$ 的内容: | "0.123"  |
| 字符串变量 D\$ 的内容: | " 0.123" |

TRIM \$

| CHAR | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

**描述**     **修剪** – 删除主导或拖尾空格

将字符串范围分配给字符串变量或字符数组时，“TRIM\$()” 返回没有前（→ 索引 L）或后（→ 索引 R）空格的字符串。没有索引的“TRIM”命令隐藏前导和拖尾空格。如果在 TRIM 命令中出现链接（如 TRIM\$(A\$+B\$)），则结果仅能分配到字符数组。

|            |                                                                                                    |
|------------|----------------------------------------------------------------------------------------------------|
| <b>语法:</b> | TRIM\$ (<CharacterString>)<br>TRIM\$ (<CharacterString> , "L")<br>TRIM\$ (<CharacterString> , "R") |
|------------|----------------------------------------------------------------------------------------------------|

表格 7-277:     TRIM\$ 语法

**示例:**

```
1 A$ = " ABCDEF "
2 B$ = TRIM$(A$,"L")
3 C$ = TRIM$(A$,"R")
4 D$ = TRIM$(A$)
5 PRN# (1,">",A$,"<")
6 PRN# (1,">",B$,"<")
7 PRN# (1,">",C$,"<")
8 PRN# (1,">",D$,"<")
```

导致逻辑编号为 1 的文件打开，用于编辑：

```
> ABCDEF <
>ABCDEF <
> ABCDEF<
>ABCDEF<
```

MID\$

| CHAR | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

**描述：** 读取字符串中的 **中间** 字符  
*结果可以传送到度量的或无度量的字符变量：*

- 定维的字符变量可接收由 MID\$ 命令定义的完整的部分字符串。
- 进行尺寸标注的字符变量仅接收已定义部分字符串的起始地址和长度。如果被移除部分字符串的字符串表达式发生改变，则进行尺寸标注的字符变量也相应地进行改变。

如果在 MID 命令中出现链接（如 MID\$ (A\$+B\$,2,3) ），则所得结果仅能分配到字符数组。

|                     |  |                                                                 |
|---------------------|--|-----------------------------------------------------------------|
| <b>语法：</b>          |  | MID\$ (<STR Expression>,<StartingPoint>[,<NumberOfCharacters>]) |
| <STR expression>    |  | 被部分移除的字符串表达式。                                                   |
| <起始点>               |  | 指定在字符数组 <StringExpression> 中的位置，该字符数组的字符被移除。                    |
| <no. of characters> |  | 确定被移除字符的数目。如果 <NumberOfCharacters> 未编程，则字符数组长度末端之前的所有字符都将被移除。   |

表格 7-278: MID\$ 语法

第二个和第三个参数的值范围包含 1 到 1,024 之间的整数值。如果不属于该值范围，则会出现错误消息 "INVALID PARAMETER"。

如果访问尚未分配的字符数组的一部分，则返回"NUL" 。

**示例：** *程序*

```
1 DIM A$(10)
2 DIM B$(5)
3 A$="ABCDEFGH I J"
4 B$=MID$(A$,2,5)
5 C$=MID$(A$,2,5)
6 REM The variables B$ and C$ both have the following content: BCDEF
7 A$="QRSTUVWXYZ"
8 REM The variable B$ has the following content: BCDEF
 The variable C$ has the content: RSTUV
```

MCASE\$

| CHAR | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|------|------|-------|-------|------|------|-------|---|
|------|------|-------|-------|------|------|-------|---|

**描述：** **混合案例** – 转换大写和小写字母  
该命令将字符串表达式转换为大写和小写字母。转换结果通过返回值提供给命令。

|                    |  |                                               |
|--------------------|--|-----------------------------------------------|
| <b>语法：</b>         |  | MCASE\$ (<StringExpression>, <DataSelection>) |
| <StringExpression> |  | 待转换的字符串表达式。                                   |

|                 |                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <DataSelection> | 字符串表达式<br>"U": 转换为大写字母<br>"L": 转换为小写字母                                                                                                                   |
| ERRNO           | CPL 变量<br>如果将 CPL 变量 ERRNO 编入参数列表中的任何位置, 则命令不产生任何内部运行时错误。该错误由变量的相应返回值返回。<br>以下为可能的返回值:<br>0: 访问正常<br>-1: 参数错误<br>如果未输入 CPL 变量 ERRNO, 则出现参数错误时, 会产生运行时错误。 |

表格 7-279: MCASE\$ 语法

示例:

```
1 DIM A$(20)
2 DIM B$(20)
3 A$="Hello world"
4 B$=MCASE$(A$, "U")
5 REM B$ 的内容如下: "HALLO WELT"
```

特殊功能和限制:

向未标注尺寸的字符参数赋予反馈值会导致运行时错误。

7. 6. 4 编程示例

描述:

可将字符串表达式分配给字符串变量。

示例:

**编程字符串变量（之前并进行尺寸标注）**

```
1 A$="ABCDE"
2 B$=CHR$(10)
```

字符串变量（没有先前的标注）的行为与引用或指针类似:

程序

```
10 DIM B$(10)
20 A$ = "Blue"
30 B$ = "Yellow"
40 PRN#(0,A$,B$) : REM output: YellowBlue
50 A$ = B$: REM A$ refers to B$
60 B$ = "XXXX"
70 PRN#(0,A$,B$) : REM output: XXXXXXXX
```

只读访问期间, 可通过 MID\$ 命令访问字符串变量的一部分:

```
1 A$="ABCDEFGHIJKLMN"
2 B$=MID$(A$,2,1)
3 C$=MID$(A$,4,4)
以下编程会导致错误:
4 MID$(A$,1,4)="ABCD"
4 A$=MID$(A$,1,3) + MID$(A$,4,1)
4 A$=B$ + A$
```

要继续处理定维的字符数组, 必须对一个或多个相关字符进行特别访问。只有这样才能将字符数组或字符数组的一部分分配给字符串变量或其他字符数组。

通过 MID\$ 命令执行对字符数组的一部分的只读和写入访问。如果仅指定了字符数组名称, 则会填写完整的字符数组。



读取字符数组

如果要对字符数组的第 <n> 个字符进行访问，则按以下内容进行（n 小于或等于字符数组的长度和该数组中的字符数目）：

```
1 DIM VWX$(13)
2 VWX$="TEST TEST TES"
3 A$ = MID$(VWX$,12,1)
4 I%=12
5 A$=MID$(VWX$,I%,1)
```

字符数组 VWX\$ 的第 12 个字符（"E"）被分配给 A\$ 字符串变量。

写字符数组

如果要将字符串变量的内容分配到字符数组或字符数组的一部分，则必须对分配进行改变。

示例：部分写入字符数组

```
1 DIM XYZ$(15)
2 B$="ABCDE"
3 MID$(XYZ$,1,5)=B$
4 MID$(XYZ$,6,5)=B$
```

1 到 10 B \$ 字符串变量的内容被分配给 XYZ \$字符数组的字符。

以下编程会导致错误消息 "CHARACTER FIELD NOT ASSIGNED" 因为字符数组的第 1 至 第 5 字符未被分配：

```
1 DIM XYZ$(15)
2 B$="ABCDE"
4 MID$(XYZ$,6,5)=B$
```

示例：部分写入字符数组

```
1 DIM XYZ$(100)
2 B$="ABCDE"
3 MID$(XYZ$,1,10)=B$
```

|                        |         |
|------------------------|---------|
| 字符串变量 B\$ 的内容：         | "ABCDE" |
| 数组变量 XYZ\$ 的内容：        | "ABCDE" |
| 数组可变长度是 5。剩余 95 字符未分配。 |         |

如果字符串变量的长度小于字符数组，则仅以字符串变量的长度写入字符数组 XYZ\$ 。将该字符数组分配到字符串变量时，不是将其分配到整个字符数组（由 DIM 指令定义），而是分配到先前写入的范围（→字符数组的长度）。

示例：

```
1 DIM XYZ$(100)
3 MID$(XYZ$,1,10)="ABCDE"
4 MID$(XYZ$,6,3)="T"
```

块 3 之后的 XYZ\$ 数组变量的内容："ABCDE"。数组可变长度是 5。剩余 95 个字符未分配，因此不是长度的一部分。

块 4 之后的 XYZ\$ 数组变量的内容："ABCDET"。数组可变长度是 6。剩余 94 个字符未分配，因此不是长度的一部分。

CPL 命令

**示例： 覆盖字符数组**

```
1 DIM XYZ$(100)
3 MID$(XYZ$,1,10)="1234567890"
4 MID$(XYZ$,3,3)="T"
```

块 3 之后的 XYZ\$ 数组变量的内容为 "1234567890"。数组可变长度是 10。

块 4 之后的 XYZ\$ 数组变量的内容为 "12T4567890"。

数组可变长度是 10。字符 "3" 被 "T 覆盖"。字符 "4" 和 "5" 被保留。

**示例： 禁止访问字符数组**

```
1 DIM XYZ$(100)
3 MID$(XYZ$,1,6)="ABCDEF"
5 MID$(XYZ$,9,5)="TESTE"
```

块 3 之后的 XYZ\$ 数组变量的内容为 "ABCDEF"。

数组可变长度是 6。

块 5 之后，试图将常量分配给字符数组的第 9 至 第 13 组件。然而，这会导致错误消息 "CHARACTER FIELD NOT ASSIGNED" 因为第 7 和第 8 元素未被分配。

如果要访问整个字符数组，则输入变量名称即可。

7.7 与客户端通信

7.7.1 MMC

| INT | READ | WRITE | FUNCT | PROC | WAIT | ERRNO | ! |
|-----|------|-------|-------|------|------|-------|---|
|-----|------|-------|-------|------|------|-------|---|

**描述：** 人机交流

将程序运行时间的信息从子程序发送到客户端，并等待此客户端的结果。借助于 CPL 变量执行此操作，该变量可以从零件程序发送值，也可以将值返回到零件程序。

运行时在接收 MMC 命令处停止零件程序。

处理中选项如下：

- 如果没有客户端报告其能够处理 MMC 命令，则分配相应的返回值 (=1)，并继续处理零件程序。
- 如果有适合的客户端，其能够用于处理 MMC 命令的数据，则在零件程序和客户端之间进行分配。客户端发送回复之后，相应的返回值被设置，并继续执行零件程序。

MMC 命令最多可拥有 20 个作为参数的 CPL 变量。这些变量的名称和值被传送到客户端。

|                                                   |                                                          |
|---------------------------------------------------|----------------------------------------------------------|
| MMC (<CPL-Var1>[,<Cpl-Var2>...[,<Cpl-VarN>]....]) |                                                          |
| <CPL-Var1>...                                     | CPL variables, N = max. 20                               |
| <CPL-VarN>                                        | 客户端可将新变量写入已在 MMC 命令中声明的 CPL 变量。MMC 命令中声明的 CPL 变量可用于零件程序。 |

表格 7-280: MMC 语法

**MMC 命令提供以下返回值作为结果：**

0: 确定

1: 客户端不存在

2: 客户端错误

## 9: 客户端已完成

---

示例:

```
10 DIM PROGNAMES$ (50)
20 PROGNAMES$="WinProg"
30 INTPAR%=1
40 REALPAR=1.1
50 I%=MMC (PROGNAMES$, INTPAR%,REALPAR)
```

;将 CPL 变量 PROGNAMES\$, INTPAR% 和 REALPAR 与它们的值一起提供给客户端。

零件程序的块准备在行 60 处不再继续执行，直到出现相应的“已完成”消息。

```
60 IF I%=0 THEN
70 IF INTPAR%=2 THEN
80 ...
90 ELSE
100 ...
110 ENDIF
120 ENDIF
```



## 8 附录

### 8.1 数控函数表格概述

#### 8.1.1 按字母顺序排列

| 数控函数名称     |      | 组<br>(“ - ”表示：块状激活) | 说明/解释                                                                                                |
|------------|------|---------------------|------------------------------------------------------------------------------------------------------|
| 全称         | 缩写形式 |                     |                                                                                                      |
| ( )        | -    | -                   | 注释：跳过括号中的内容                                                                                          |
| (MSG       | -    | -                   | 输出消息文本                                                                                               |
| /          | -    | -                   | 跳过块                                                                                                  |
| //         | -    | -                   | 注释                                                                                                   |
| ;          | -    | -                   | 注释：跳过行的其余部分                                                                                          |
| ABS        | -    | CPL                 | ABS(<Input value>)                                                                                   |
| AC(...)    | -    | -                   | 本地绝对尺寸编程，例如 X = AC (10)                                                                              |
| ACOS       | -    | CPL                 | <Function value> = ASIN(<input value>):<br>反余弦函数应用于 <input value>。                                   |
| ACS        | -    | CPL                 | ACS(<axis>[, <selection type>[, <channel>]]):<br>提供轴的当前指令位置。                                         |
| ActRadFact | ARF  |                     | ActRadFact(<axis name> <factor> {, <axis name> <factor>}...)<br>编制有效半径因子                             |
| 调整         | ADJ  | -                   | ADJ(<NCF-Name>):<br>调整模态数控函数组。                                                                       |
| AND        | -    | CPL                 | <expression1> AND <expression2>:<br>两个 BOOLEAN 或 INTEGER 表达式的二进制操作                                   |
| APOS       | -    | CPL                 | APOS(<axis selection>):<br>根据机器零点传输当前的实际轴值。                                                          |
| 区          | ARA  | -                   | Area(<BNr>, <Sta>{, <Mod>, {<P1>}, {<P2>}, {<D1>}, {<D2>}}):<br>定义，激活或禁用多达 10 个矩形，二维死区或具有轴平行边界的工作范围。 |
| ASC        | -    | CPL                 | ASC(<character string>)                                                                              |
| ASIN       | -    | CPL                 | <Function value> = ASIN(<input value>):<br>反余弦函数在 <input value> 上的应用。                                |
| ASPCLR     | -    | -                   | ASPCLR (<Up no>)<br>异步子程序：登录。                                                                        |
| ASPDIS     | -    | -                   | ASPDIS (<Up no. >)<br>异步子程序：关闭                                                                       |
| ASPENA     | -    | -                   | ASPENA (<Up no. >)<br>激活异步子程序                                                                        |

## 附录

| 数控函数名称     |      | 组              | 说明/解释                                                                                                                                    |
|------------|------|----------------|------------------------------------------------------------------------------------------------------------------------------------------|
| 全称         | 缩写形式 | (“ - ”表示：块状激活) |                                                                                                                                          |
| ASPRTP     | -    | -              | ASPRTP(<Up no. >, <Point>)<br>异步子程序：定义返回轮廓的点                                                                                             |
| ASPSET     | -    | -              | ASPSET(<Up no. >, <Up name>{, <Flags>})<br>异步子程序：注册                                                                                      |
| ASPSTA     | -    | -              | ASPSTA(<Up no. >{, <channel no. >})<br>异步子例程：触发器                                                                                         |
| AssLogName | ALN  | -              | ALN(...):<br>分配逻辑轴名称                                                                                                                     |
| ASTOPA     | -    | -              | ASTOPA[<ChannelNo. >, <Cond. 1> {, <Cond. 2>} {... {, <Cond. 8>}...}]<br>通道同步命令：<br>只要满足 <b>所有</b> 条件，就要停止要控制的通道。条件是指轴位置<br>例如 “Z” > 20。 |
| ASTOPO     | -    | -              | ASTOPO[<ChannelNo. >, <Cond. 1> {, <Cond. 2>} {... {, <Cond. 8>}...}]<br>通道同步命令：<br>只要满足 <b>一个</b> 条件就会停止要控制的通道。条件涉及轴位置，例如，<br>“Z” > 20  |
| ATAN       | -    | CPL            | <Function value> = ATAN(<input value>):<br>反正切函数在<input value>上的应用。                                                                      |
| ATAN2      | -    | CPL            | <Function value> = TAN2(input value1, input value2)<br>该功能可用于确定正确象限内的角度                                                                  |
| ATBWD      | -    | -              | AtBwd(<AxCoord>, <Coord>)<br>校准轴运动：向后转型                                                                                                  |
| ATCAL      | -    | -              | ATCAL(<File>, <OptData>, <Mask> {, <Info>} {, <NumIt>})<br>校准轴运动：优化参数。                                                                   |
| ATFWD      | -    | -              | ATFWD(<Coord>, <AxKoord>{, <ParData>})<br>校准轴运动：转换参数                                                                                     |
| ATGET      | -    | -              | ATGET(<ParData>{, <ATrafNo>})<br>校准轴运动：从 NC 读取参数                                                                                         |
| ATPUT      | -    | -              | ATPUT(<ParData>{, <ATrafNo>})<br>校准轴运动：将参数写入 NC                                                                                          |
| ATrans     | ATR  | 程序坐标偏置         | ATR(X..., Y..., Z):<br>叠加编程坐标偏置开启<br>ATR():<br>附加程序坐标偏移 OFF                                                                              |
| AUXFUNC    | -    | -              | 执行所有组的有效辅助函数。                                                                                                                            |

| 数控函数名称            |      | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                            |
|-------------------|------|-------------------------|----------------------------------------------------------------------------------|
| 全称                | 缩写形式 |                         |                                                                                  |
| AxAcc             | AAC  | 轴加速度                    | AAC(X..., Y..., Z): 开启<br>AAC(1): 开启保存的设置<br>AAC(): 关闭<br>轴加速编程                  |
| AxAccSave         | AAS  | -                       | 保存当前的轴加速度                                                                        |
| AxCompTab         | -    | -                       | AxCompTab(<CAX>,<table>):<br>交换轴误差补偿表 (LSEC 和 CCOMP)                             |
| AxCouple          | AXC  | 轴耦合                     | AXC(<Master>,<Slave1>(...),<Slave2>(...),...):<br>激活轴耦合<br>AXC():<br>停用轴耦合       |
| AXADR\$           | -    | CPL                     | AXADR\$(<Index> [ , [ <selection type> ] [ , <channel<br>number> ] ] )<br>轴和坐标地址 |
| AXINF             | -    | CPL                     | 为通道或系统轴提供系统轴索引作为返回值。                                                             |
| AxisReplace       | AXR  |                         | AXR(<KAN1>,<PAN2>)<br>更换轴                                                        |
| AxisToSpindle     | ATS  | -                       | ATS(<Axisname>):<br>将轴切换到主轴模式。                                                   |
| AxisToSpindleWait | ATSW | -                       | AxisToSpindleWait(<PAN> <PAI>{,<PAN> <PAI>}...)<br>关闭主轴的 C 轴模式并等待                |
| AxJerk            | AJK  | -                       | AxJerk(<values>)<br>更改最大轴突变<br>AxJerk() 或 AxJerk(0)<br>再次从机床参数激活轴加速度             |
| AXO               | -    | CPL                     | AXO((<axis selection>[,<selection type>]):<br>传输坐标的活动 SetPos 偏移量。                |
| AXP               | -    | CPL                     | AXP(axis number>,<path information>):<br>该功能允许平面独立的零件和测量程序 (程序 AXP 代替地址值)。       |
| AxVel             | AVE  | -                       | AxVel(<values>):<br>更改最大轴速率                                                      |
| AxVelSave         | AVS  | -                       | 缓冲最大轴速度                                                                          |
| BCD               | -    | CPL                     | <BCD value> = BCD(<binary value>):<br>从 BCD 转换为二进制格式。                            |

## 附录

| 数控函数名称    |      | 组              | 说明/解释                                                                                                                                                                                              |
|-----------|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 全称        | 缩写形式 | (“ - ”表示：块状激活) |                                                                                                                                                                                                    |
| BcsCorr   | BCR  | 工件位置修正         | BCR({<XW offset>} {, {<YW offset>}<br>{, {<ZW offset>} {, {<Angle1>}<br>{, {<Angle2>} {, {<Angle3>}}}}}): 开启<br>BCR( ): 关闭<br>定位: 刀具位置修正。                                                          |
| BIN       | -    | CPL            | <Binary value> = BIN(<BCD value>):<br>从二进制转换为 BCD 格式。                                                                                                                                              |
| BITIF     | -    | CPL            | BITIF(<BitSignal>[, <Index>[, <IF Unit>]]):<br>访问 NC 和 PLC 之间的数字接口。                                                                                                                                |
| BlkNmb    | BNB  | -              | BlkNmb(<No>):<br>限制通过块准备读入和查看的程序块的数量。                                                                                                                                                              |
| BSTOPA    | -    | -              | BSTOPA[<ChannelNo. >, <Cond. 1> {, <Cond. 2>} {... {, <Cond. 8>}...}]<br>通道同步命令: 只要满足所有条件, 就要停止要控制的通道。条件是指基本的工件位置, 例如“Z”> 20。                                                                      |
| BSTOPO    | -    | -              | BSTOPA[<ChannelNo. >, <Cond. 1> {, <Cond. 2>} {... {, <Cond. 8>}...}]<br>通道同步命令: 只要满足一个条件就会停止要控制的通道条件是指基本的工件位置, 例如“Z”> 20。                                                                         |
| CALL      | -    | CPL            | CALL <Program number><br>[<Transfer parameter1>, ...] [DIN]:<br>从一个 CPL 程序调用子程序                                                                                                                    |
| CASE      | -    | CPL            | CASE <integer expression> OF<br>LABEL <Int. constant><br>[, <other int. constant>]<br>[: <instruction>]<instruction><br>标签 ...<br>OTHERWISE <instruction><instruciton><br>ENDCASE<br>有条件的选择几个备选方案。 |
| ChkNxtBlk | CNB  | -              | ChkNxtBlk (1) 开启连续块监视<br>ChkNxtBlk (0) 关闭连续块监视                                                                                                                                                     |
| ChLength  | CHL  | 倒角/倒圆          | CHL(<chamfer length>): 开启<br>CHL( ): 关闭<br>插入具有确定倒角长度的过渡倒角。                                                                                                                                        |
| CHR\$     | -    | CPL            | CHR\$(<integer expression>):<br>提供一个字符, 其 ASCII 码表中的序号等于通过参数 <INTEGER expression> 传送的值。                                                                                                            |



| 数控函数名称                  |      | 组              | 说明/解释                                                                                           |
|-------------------------|------|----------------|-------------------------------------------------------------------------------------------------|
| 全称                      | 缩写形式 | (“ - ”表示：块状激活) |                                                                                                 |
| ChSection               | CHS  | 倒角/倒圆          | CHS(<chamfer segment>): 开启<br>CHS(): 关闭<br>使用已定义的倒角段插入过渡倒角。                                     |
| CLOCK                   | -    | CPL            | <Function value> = CLOCK:<br>时间计数器的查询以毫秒为单位。                                                    |
| CLOSE                   | -    | CPL            | CLOSE(<n>):<br>读取或写入操作完成后关闭开启的文件。                                                               |
| CLRWARN                 | -    | CPL            | 删除使用 SETWARN 生成的通道的警告。                                                                          |
| COF                     | -    | CPL            | COF(<axis selection>[, <selection type>]):<br>提供当前通道坐标的最新编程轮廓移位。                                |
| COFFS                   | -    | -              | 3D 半径修正的轮廓偏置                                                                                    |
| 碰撞                      | CLN  | 冲突监控           | CLN(1): 开启<br>CLN(CollErr...   LA...   DLA...   DEF):<br>开启参数设置<br>CLN(): 关闭<br>2D 铣刀路径修正的碰撞监控。 |
| ConfG0                  | CG0  | 插补             | ConfG0(ASY<status>)<br>ConfG0(ASY1)<br>启用异步 G0 插补<br>ConfG0(ASY0)<br>停用异步 G0 插补                 |
| ConstFeed               | CFD  | V-profile      | 恒定速度                                                                                            |
| CornerLineDetection     | CLD  | -              | 自动检测拐角和直线                                                                                       |
| Coord                   | CRD  | -              | CRD(x): 开启轴转换<br>CRD(): 关闭轴转换, 例如端面改造                                                           |
| COS                     | -    | CPL            | <Function value> = COS(<input value>):<br>余弦函数用于 <input value>。                                 |
| CoupleSplineTab         | CST  | -              | CST(STAB(...))<br>生成花键联轴器表                                                                      |
| CurveParamInterpolation | CPI  | 插补             | CurveParamInterpolation(1)<br>开启曲线参数插补<br>CurveParamInterpolation(0)<br>切换回路长度插值                |
| D                       | D    | D 修正           | Dx: 选择 D 修正 (1-99 刀具)<br>D0: 取消选择 D 修正                                                          |

## 附录

| 数控函数名称     |      | 组              | 说明/解释                                                                                               |
|------------|------|----------------|-----------------------------------------------------------------------------------------------------|
| 全称         | 缩写形式 | (“ - ”表示：块状激活) |                                                                                                     |
| DATE       | -    | CPL            | <String variable> = DATE:<br>将日期作为 DD.MM 分配给<STRING variables>。                                     |
| DBLOAD     | -    | CPL            | DBLOAD(<DbTable>, <Key1>, <Key2>),<br><File name>[, <Mode>]):<br>数据库表中的部分或来自文件的完整数据库表可以读入数据库。       |
| DBMOVE     | -    | CPL            | DBMOVE(<DbTab>, <SrcKey1>, <SrcKey2>,<br><DestKey1>, <DestKey2>[, <Mode>]):<br>在数据库表中实现数据集。         |
| DBSAVE     | -    | CPL            | DBSAVE(<DbTab>, <Key1>, <Key2>,<br><File name>[, <Mode>]):<br>使用 DBSAVE 将数据库表完全或部分保存到文件中。           |
| DBSEA      | -    | CPL            | DBSEA (<DbTab>, <Key1>, <Key2>,<br><SearchCond>, <SearchRes>[, <ResVar>]):<br>搜索数据库表中的数据集。          |
| DBSEAX     | -    | CPL            | DBSEAX(<DbTable>, <Key1>, <Key2>,<br><SearchCond>, <ErgArr>, <ErgSize>):<br>在刀具数据库表中搜索与搜索条件相对应的数据集。 |
| DBTAB      | -    | CPL            | DBTAB(<DbTable>, <Key1>, <Key2><br>[, <ResVar>])<br>将完整的数据集或数据库表的子结构读取到 CPL 变量或从此变量写回。              |
| DBTABX     | -    | CPL            | DBTABX(<DbTable>, <Key1>, <Key2><br>[, <Mode>]):<br>可以将完整的数据集或刀具数据库表的子结构读取到 CPL 变量中，或从变量写入数据集。      |
| DBTABXL    | -    | CPL            | DBTABXL(<DbTable>, <Key1>, <Key2>[, <Mode>]):<br>可以将完整的数据集或刀具数据库表的子结构读取到 CPL 变量中，或从变量写入数据集。         |
| DCOFFS     | -    | -              | 3D 半径修正的轮廓偏置的变化率                                                                                    |
| DCT        | -    | CPL            | DCT(<value selection>, <Data block><br>[, [, <Table>] [, <Unit>]]):<br>读取和写入任何 D 修正表格或外部刀具修正值。      |
| DcTSe1     | DCS  | -              | DCS({<Path>}<FileName>):<br>选择一个 D 修正表                                                              |
| DefAxis    | DAX  | -              | 创建默认轴配置                                                                                             |
| DefSpindle | DSP  | -              | 把通道主轴复位到 MP                                                                                         |

| 数控函数名称       |      | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                                  |
|--------------|------|-------------------------|--------------------------------------------------------------------------------------------------------|
| 全称           | 缩写形式 |                         |                                                                                                        |
| DefTangTrans | DTT  | 过渡角                     | DTT(<Transition angle>): 激活中<br>DTT(): 停用<br>切向轮廓过渡的定义                                                 |
| DiaProg      | DIA  | 直径/半径                   | 直径编程                                                                                                   |
| DIM          | -    | CPL                     | DIM <Variable name>(<array size1><br>[, <Array size2>]):<br>使用 INTEGER 常量指定 ARRAY 变量的数组大小（标注尺寸）。       |
| DIRCR        | -    | CPL                     | DIRDEL(<Directory>)<br>创建一个新目录。                                                                        |
| DIRDEL       | -    | CPL                     | DIRDEL(<Directory>)<br>删除一个空目录。                                                                        |
| DIRINF       | -    | CPL                     | DIRINF(4, [<Index2>], <ResultVar><br>[, <file number>]):<br>调用文件系统的管理数据                                |
| DistCtrl     | DCR  | -                       | DCR(1): 根据机器参数进行数字化 ON 的距离控制。<br>DCR(<Fct>): 使用单独配置数据开启<br>DCR(): 距离控制关闭                               |
| DPC          | -    | CPL                     | DPC(<Selection>[, <Selection type>]):<br>提供当前通道的坐标（偏移值和转角）的工件位置修正“BcsCorr”的<br>最新编程参数。                 |
| ED           | ED   | Ed-修正                   | ED x: 选择外部刀具修正（1..16 个刀具/边缘）<br>ED0: 取消选择外部刀具修正                                                        |
| EndPosCouple | EPC  | -                       | EPC(<M-Coord>, <S-Coord>, <Factor>):<br>终点位置耦合开启<br>EPC(): 终点位置耦合关闭                                    |
| EOF          | -    | CPL                     | EOF(<n>): 检查文件终点                                                                                       |
| ERASE        | -    | CPL                     | ERASE(<File name>):<br>擦除文件                                                                            |
| ERRNO        | -    | CPL                     | 各种 CPL 功能的错误信息。                                                                                        |
| EXP          | -    |                         | EXP(<Exponent> [, <Basis> ] )<br>如果未指定 <Basis>（可选参数），则计算关于任意编程基准或默认<br>基准 e（e = 2.718281828 ...）的指数函数。 |
| F            | F    | -                       | G93、G94、G95 的 F 地址                                                                                     |
| F2 地址        | F2   |                         | F2=<value><br>语法是“F2=”。“F2”和“=”之间不允许有空格。                                                               |
| FA           | FA   | -                       | 异步轴的速度                                                                                                 |

## 附录

| 数控函数名称      |      | 组              | 说明/解释                                                                                                                  |
|-------------|------|----------------|------------------------------------------------------------------------------------------------------------------------|
| 全称          | 缩写形式 | (“ - ”表示：块状激活) |                                                                                                                        |
| 假           | -    | CPL            | <Boolean variable> = FALSE:<br>布尔变量的真值                                                                                 |
| FeedAd      | FAD  | Feed 广告。       | FAD(1) : 从供稿计算中删除轴<br>FAD( ) : 进给计算中的轴                                                                                 |
| FeedForward | FFW  | 前馈             | FFW(X..., Y..., Z ) : 开启<br>FFW( ) : 关闭<br>启用前馈                                                                        |
| FeedGrp     | FDG  | 进给计算           | FeedGrp (1) 或<br>FeedGrp<br>进给计算：进给组                                                                                   |
| FeedGrpType | FDGT | 进给计算           | FeedGrpType (SMOOTH) 或<br>FeedGrpType(1)<br>进给计算：进给组类型                                                                 |
| FILEACCESS  | -    | CPL            | FILEACCESS(<file name>) :<br>确定文件是否存在及其访问权限。                                                                           |
| FILECOPY    | -    | CPL            | 复制文件                                                                                                                   |
| FILEDATE    | -    | CPL            | FILEDATE(<File name>[, <Mode>]) :<br>确定文件的日期/时间。                                                                       |
| FILENO      | -    | CPL            | 提供下一个有效的逻辑文件编号。                                                                                                        |
| FILEPOS     | -    | CPL            | FILEPOS(<n>[, <mode>]) :<br>提供当前记录的记录编号和随机文件的记录偏移量。<br>为顺序文件提供文件指针的当前字节位置。                                             |
| FILESIZE    | -    | CPL            | FILESIZE(<n>[, <k>]) :<br>提供文件大小或已经写入文件的限制。                                                                            |
| FlyMeas     | FME  | 飞行测量           | FME(MpiAxis< i>) X...Y...Z... :<br>与飞行测量一起旅行                                                                           |
| FOR NEXT    | -    | CPL            | FOR <counting var.>=<start count> [ STEP <step size>] TO <end count><routine>NEXT [<count variable>]:<br>自动更新计数器的循环结构。 |
| FsMove      | FSM  | -              | 转到固定停止<br>(Fs = fixed stop)                                                                                            |
| FsProbe     | FSP  | -              | 在固定站点测量                                                                                                                |
| FsReset     | FSR  | -              | 取消固定停止                                                                                                                 |
| FsTorque    | FST  | -              | 转矩用于固定停止                                                                                                               |
| G           | -    | -              | 用户定义的本地子例程                                                                                                             |
| G0          | G0   | 插补             | 线性插值（快速移动）                                                                                                             |

| 数控函数名称        |        | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                                                                                                                                |
|---------------|--------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 全称            | 缩写形式   |                         |                                                                                                                                                                                                      |
| G0(..)        | G0(..) | 插补                      | 使用选项快速进给<br>可选参数：<br>- NIPS: 没有确切的停止<br>- IPS1: 准确停止窗口 1<br>- IPS2: 准确停止窗口 2<br>- IPS3: 准确停止窗口 3 (仅减速至 V=0)<br>- POL / POLAR: 用极坐标编程，例如 G0 (POL) X50 A45 B10，其中 A\B =极角 1/2                          |
| G1            | G1     | 插补                      | 线性插值 (feed)                                                                                                                                                                                          |
| G2            | G2     | 插补                      | 顺时针圆弧插补 (包括螺旋 N)                                                                                                                                                                                     |
| G2 (POL)      | G2(..) | 插补                      | 带极坐标编程的顺时针圆弧插补 (包括螺旋 N)                                                                                                                                                                              |
| G3            | G3     | 插补                      | 逆时针圆弧插补 (包括螺旋 N)                                                                                                                                                                                     |
| G3 (POL)      | G3(..) | 插补                      | 带极坐标编程的逆时针圆弧插补 (包括螺旋-N)                                                                                                                                                                              |
| G4            | G4     | -                       | G4(F..): 以秒为单位的停留时间<br>G4(S...): 主轴旋转停留时间。参考主轴是主主轴 (MainSp)                                                                                                                                          |
| G5            | G5     | 插补                      | 切向圆圈入口 (包括螺旋 N)                                                                                                                                                                                      |
| G6            | G6     | 插补                      | 样条\nurbs                                                                                                                                                                                             |
| G0T010        | G0T010 | 路径斜坡                    | 根据机器参数，路径斜坡与路径 SHAPE 开启                                                                                                                                                                              |
| G8 (SHAPE...) | -      | 路径斜坡                    | 路径斜坡与编程设置<br>路径形状                                                                                                                                                                                    |
| G0T010        | G0T010 | 路径斜坡                    | 路径坡度 OFF (无 SHAPE)                                                                                                                                                                                   |
| G9(..)        | G9(..) | 路径斜坡                    | G9(X..., Y):<br>路径坡度关闭，逐轴编程 SHAPE 命令<br>G9(ASHAPE):<br>根据机床参数，按照轴的 SHAPE 顺序关闭路径坡度<br>G9(SHAPE ...):<br>路径坡度关闭，编程路径 SHAPE<br>G9(SIN ...):<br>路径坡度关闭与 $\sin^2$ -线型 SHAPE (可能的 SHAPE 命令: 5、10、15、20、40) |
| G140          | -      | 3D 半径修正                 | 3D 半径修正关闭                                                                                                                                                                                            |
| G141          | -      | 3D 半径修正                 | 轮廓左侧的 3D 半径修正                                                                                                                                                                                        |
| G142          | -      | 3D 半径修正                 | 3D 轮廓右侧的半径修正                                                                                                                                                                                         |
| G152.1        | G152   | IP 库 1                  | 定位: 倾斜的平面，可编程，库 1                                                                                                                                                                                    |
| G152.2-G152.5 | -      | IP 库 2-5                | 定位: 斜面，可编程，库 2-5                                                                                                                                                                                     |
| G153          | -      | SE                      | 定位: 斜面，全部关闭                                                                                                                                                                                          |
| G153.1-G153.5 | -      | IP 库 1-5                | 定位: 倾斜的平面，库 1-5 OFF                                                                                                                                                                                  |

## 附录

| 数控函数名称                       |      | 组              | 说明/解释                                                                                                                                  |
|------------------------------|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 全称                           | 缩写形式 | (“ - ”表示：块状激活) |                                                                                                                                        |
| G154.1                       | G154 | IP 库 1         | 定位：1. 斜面，库 1                                                                                                                           |
| G154.2-G154.5                | -    | IP 库 2-5       | 定位：1. 斜面，库 2-5 开启                                                                                                                      |
| G155.1                       | G155 | IP 库 1         | 定位：2. 斜面，库 1                                                                                                                           |
| G155.2-G155.5                | -    | IP 库 2-5       | 定位：2. 斜面，库 2-5 开启                                                                                                                      |
| G156.1                       | G156 | IP 库 1         | 定位：3. 斜面，库 1                                                                                                                           |
| G156.2-G156.5                | -    | IP 库 2-5       | 定位：3. 斜面，库 2-5 开启                                                                                                                      |
| G157.1                       | G157 | IP 库 1         | 定位：4. 斜面，库 1                                                                                                                           |
| G157.2-G157.5                | -    | IP 库 2-5       | 定位：4. 斜面，库 2-5 开启                                                                                                                      |
| G158.1                       | G158 | IP 库 1         | 定位：5. 斜面，库 1                                                                                                                           |
| G158.2-G158.5                | -    | IP 库 2-5       | 定位：5. 斜面，库 2-5 开启                                                                                                                      |
| G159.1                       | G159 | IP 库 1         | 定位：6. 斜面，库 1                                                                                                                           |
| G159.2-G159.5                | -    | IP 库 2-5       | 定位：5. 斜面，库 2-5 开启                                                                                                                      |
| G16                          | -    | 平面选择           | 没有平面                                                                                                                                   |
| G17                          | -    | 平面选择           | 平面选择 XY                                                                                                                                |
| G17(...), G18(...); G19(...) | -    | 平面选择           | 高级平面切换<br>G17/18/19(<axis1>, <axis2>, <axis3>):<br>圆括号中的轴开启 WCS 并接收 X、Y 和 Z 的含义。随后，选择编程的平面。<br>G17/18/19( ):<br>将工件坐标系重置为默认设置并选择编程的平面。 |
| G18                          | -    | 平面选择           | 平面选择 ZX                                                                                                                                |
| G19                          | -    | 平面选择           | 平面选择 YZ                                                                                                                                |
| G20                          | -    | 平面选择           | 免费飞机选择（独立于 WCS）<br>配售影响 WCS。飞机的定义与 WCS 无关。                                                                                             |
| G33                          | -    | 线              | 螺纹切削                                                                                                                                   |
| G40                          | -    | 路径修正           | 铣刀路径校正关闭                                                                                                                               |
| G41                          | -    | 路径修正           | 铣削刀具路径修正到工件的左侧                                                                                                                         |
| G42                          | -    | 路径修正           | 铣削刀具路径修正到工件右侧                                                                                                                          |
| G43                          | -    | 插入策略           | 插入策略：弧                                                                                                                                 |
| G44                          | -    | 插入策略           | 插入策略：交集                                                                                                                                |
| G45                          | -    | 铣刀速度           | 铣刀接触点的速度                                                                                                                               |
| G46                          | -    | 铣刀速度           | 铣刀中心的速度                                                                                                                                |
| G47                          | -    | 刀具修正           | 开启刀具长度修正                                                                                                                               |
| G47(..)                      | -    | 刀具修正           | G47(<L1-coord>, <L2-coord>, <L3-coord>) G47(ActPlane):<br>切换修正分配时刀具长度修正开启。<br>G47( )：刀具长度修正开启，根据机器参数进行修正分配。                            |

## 附录

| 数控函数名称       |      | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                |
|--------------|------|-------------------------|--------------------------------------------------------------------------------------|
| 全称           | 缩写形式 |                         |                                                                                      |
| G48          | -    | 刀具修正                    | 刀具长度修正关闭                                                                             |
| G53          | -    | -                       | 所有零点偏置关闭                                                                             |
| G53.1-G53.5  | -    | Z0 库 1-5                | 零点偏移，库 1-5 OFF                                                                       |
| G54.1        | G54  | Z0 库 1                  | 1. 零点偏移，库 1 开启                                                                       |
| G54.2-G54.5  | -    | Z0 库 2-5                | 1. 零点偏移，库 2-5 开启                                                                     |
| G55.1        | G55  | Z0 库 1                  | 2. 零点偏移，库 1 开启                                                                       |
| G55.2-G55.5  | -    | Z0 库 2-5                | 2. 零点偏移，库 2-5 开启                                                                     |
| G56.1        | G56  | Z0 库 1                  | 3. 零点偏移，库 1 开启                                                                       |
| G56.2-G56.5  | -    | Z0 库 2-5                | 3. 零点偏移，库 2-5 开启                                                                     |
| G57.1        | G57  | Z0 库 1                  | 4. 零点偏移，库 1 开启                                                                       |
| G57.2-G57.5  | -    | Z0 库 2-5                | 4. 零点偏移，库 2-5 开启                                                                     |
| G58.1        | G58  | Z0 库 1                  | 5. 零点偏移，库 1 开启                                                                       |
| G58.2-G58.5  | -    | Z0 库 2-5                | 5. 零点偏移，库 2-5 开启                                                                     |
| G59.1        | G59  | Z0 库 1                  | 6. 零点偏移，库 1 开启                                                                       |
| G59.2-G59.5  | -    | Z0 库 2-5                | 6. 零点偏移，库 2-5 开启                                                                     |
| G61          |      | 准确停止                    | 准确停止开启                                                                               |
| G61 (IPS...) | -    | 准确停止                    | 在准确停止窗口 IPS1、IPS2 或 IPS3 准确停止 ON                                                     |
| G62          | -    | 准确停止                    | 准停定位关闭                                                                               |
| G63          | -    | -                       | G63 (M3/M4, S.../H...) F... Z:<br>攻丝无补偿卡盘                                            |
| G70          | -    | 英制/公制                   | 英制编程会影响编程的路径和长度，进给和加速度。                                                              |
| G71          | -    | 英制/公制                   | 公制单位编程                                                                               |
| G74          | -    | -                       | G74 X1 Y1 Z1 ...:<br>接近参考点坐标                                                         |
| G74 (Home)   | -    | -                       | G74 (HOME) X1 Y1 Z1:<br>逼近参考点（实际参考，也适用于异步轴）                                          |
| G75          | -    | -                       | 移动探头（运动中止）                                                                           |
| G76          | -    | -                       | 固定机器位置（机器坐标）的方法                                                                      |
| G77          | -    | -                       | G77 <coord 1><Mode> <coord n><Mode> ... F<value><br>其他语法：REPOS<br>异步子程序：<br>重新定位个别坐标 |
| G78          | -    | -                       | G78({{-}<Coordinate1>}, {{-}<Coordinate2>},<br>{{-}<Coordinate3>}):<br>修正开启          |
| G79          | -    | -                       | 修正关闭                                                                                 |

## 附录

| 数控函数名称                       |      | 组              | 说明/解释                                                                                   |
|------------------------------|------|----------------|-----------------------------------------------------------------------------------------|
| 全称                           | 缩写形式 | (“ - ”表示：块状激活) |                                                                                         |
| G90                          | -    | 绝对/相对          | 绝对尺寸编程                                                                                  |
| G91                          | -    | 绝对/相对          | 相对尺寸编程                                                                                  |
| G93                          | -    | 进给编程           | 时间编程                                                                                    |
| G94                          | -    | 进给编程           | Feed 编程（每分钟）                                                                            |
| G94(...)                     | -    | 进给编程           | G94({DF <value>}, { DS1 <value>, DS2 <value>, ...}):<br>带加速度适应的增量速度编程。                  |
| G95                          | -    | 进给编程           | Feed 编程（每转）                                                                             |
| G96                          | -    | 主轴编程           | G96({(<reference axis>{,<actuation point>}}}):<br>恒定的切割速度                               |
| G97                          | -    | 主轴编程           | 直接速度编程                                                                                  |
| GCT(0)                       | -    | -              | 关闭直线度和角度误差补偿                                                                            |
| GCT(1)                       | -    | -              | 开启直线度和角度误差补偿                                                                            |
| GetAxis                      | GAX  | -              | GAX(...):<br>将自由轴应用于通道                                                                  |
| GETERR                       | -    | CPL            | GETERR(<channel> [, <category> ],<br><Error>-No> [, <Number>]):<br>为当前错误提供错误号，通道号和错误类别。 |
| GetSpindle                   | GSP  | -              | GetSpindle (<SysSpNo>   <SysSpName>, <ChanSpNo>  <br><ChanSpName>, {...}):<br>创建通道主轴    |
| GMSG                         | -    | -              | (GMSG<Information text>)<br>将注释编入用户界面                                                   |
| GoAhead                      | GOA  | -              | 向前跳                                                                                     |
| GoBack                       | GOB  | -              | 向后跳                                                                                     |
| GoCond                       | GOC  | -              | 有条件跳转                                                                                   |
| GOTO                         | -    | CPL            | GOTO <Jump target>:<br>无条件程序跳转到行号，程序段号或标签。                                              |
| GoTo                         | -    | -              | 无条件跳转                                                                                   |
| HsBlkSwitch                  | HSB  | -              | HSB(HS<X>=<Y>):<br>通过高速信号实时阻止切换                                                         |
| HsBlkSwitch(..HS<br>STOP=..) | HSB  | -              | HSB(..., HSSTOP=..):<br>通过高速信号终止切换                                                      |



## 附录

| 数控函数名称                 |      | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                                                                                 |
|------------------------|------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 全称                     | 缩写形式 |                         |                                                                                                                                                       |
| HWOC                   | -    | -                       | HWOC({CHAN<channel no>}, CRDNO <coordno>, { STEP<Incr>}):<br>工件坐标在线修正 ON<br>HWOCDIS({CHAN<channel no>}):<br>在线修正 OFF, 保存值<br>HWO( ):<br>在线修正 OFF, 删除值 |
| IC(...)                | -    | -                       | 本地相对维度编程<br>例如 X=IC(5)                                                                                                                                |
| IF ENDIF               | -    | CPL                     | IF <condition> THEN <routine><br>[ELSE <Optional routine>] ENDIF:<br>有条件跳转到例程或替代例程。                                                                   |
| InitMeas               | IME  | -                       | IME(MpiAxis i):<br>初始化“飞行测量”                                                                                                                          |
| INP#                   | -    | CPL                     | INP#(<n>, <Variable>[, <Variable>] [,...][;]):<br>读取文件的访问权限                                                                                           |
| INSDEP                 | -    | -                       | 3D 半径修正切入深度                                                                                                                                           |
| INSTR                  | -    | CPL                     | INSTR(<string sequence>, <string expression>[, <starting point>]):<br>在<string sequence>之后从<starting point>搜索<String expression>, 并将其初始位置输出为整数值。      |
| INT                    | -    | CPL                     | <Integer number>=INT(<Real expression>):<br>通过跳过小数来将 <real expression> 转换为 <integer number>。                                                          |
| InvTime                | ITIM |                         | ITIM <value><br>逆时间编程                                                                                                                                 |
| GOTO10<br>ISP2<br>ISP3 | -    | -                       | IPS1: 精细确切停止窗口大小<br>IPS2: 粗略的停止窗口大小<br>IPS3: 无限准确停止窗口大小                                                                                               |
| JerkAxList             | JAL  | -                       | 确定系统轴以指定在激活冲击限制情况下的最佳 SHAPE 过滤器                                                                                                                       |
| JerkControl            | JKC  | -                       | 在工件坐标中慢跑                                                                                                                                              |
| JogWCSSelect           | -    | -                       | 选择坐标以在工件坐标中进行点动（设置模式）。                                                                                                                                |
| KinSplitDisable        | KSD  | -                       | KinSplitDisable(0)<br>启用运动数控块分割<br>KinSplitDisable(1)<br>停用运动数控块分割                                                                                    |
| KvProg                 | KVP  | C0 编程                   | COP(Y..., Y..., Z): C0 编程开启<br>KVP( ): C0 编程关闭                                                                                                        |
| LCP                    | -    | -                       | 迟滞补偿<br>一个或多个附加的轴/轴与配置中定义的主轴耦合（或分离）。                                                                                                                  |

## 附录

| 数控函数名称                                 |        | 组              | 说明/解释                                                      |
|----------------------------------------|--------|----------------|------------------------------------------------------------|
| 全称                                     | 缩写形式   | (“ - ”表示：块状激活) |                                                            |
| LEN                                    | -      | CPL            | LEN(<String expression>):<br>提供 STRING 表达式中的字符数。           |
| LEN                                    | -      | -              | LEN=<value>:<br>将横移块分成几段等长的段。                              |
| LFCnf                                  | LFC    | -              | LFC({LL(...)}):<br>参数化激光功率控制                               |
| LFP                                    | LFP    | -              | LFP, LFP(1), LFP({LL(...)}): 激光功率控制开启<br>LFP(0): 激光功率控制器关闭 |
| LinDownFeed                            | LND    | V-profile      | 线性减速                                                       |
| LinearToSpline                         | LTS    | -              | 线性样条切换<br>LinearToSpline(1): 开启<br>LinearToSpline(0): 关闭   |
| LinModZp                               | LMZ    | -              | LMZ({LinModAxis i}):<br>将线性模数轴设置为零<br>(Zp = zero point)    |
| LinUpFeed                              | LNU    | V-profile      | 线性加速                                                       |
| LJUST                                  | -      | CPL            | 切换到左对齐的数据输出；直到程序运行结束。                                      |
| M                                      | M      | -              | 用户定义的本地子例程                                                 |
| GOT010                                 | GOT010 | -              | 程序停止                                                       |
| GOT010                                 | GOT010 | -              | 有条件的程序停止                                                   |
| M19, M119, M219                        | -      | -              | 定向主轴                                                       |
| GOT010                                 | GOT010 | -              | 主程序终点                                                      |
| M3、M103、M203、<br>M13、M113、M213         | -      | -              | 主轴正转<br>冷却液开启时主轴顺时针旋转                                      |
| M30                                    | M30    | -              | 主程序终点                                                      |
| M4,<br>M104, M204, M14,<br>M114, M214  | -      | -              | 主轴逆时针<br>冷却液开启时主轴逆时针旋转                                     |
| M40、M140                               | -      | -              | 自动档位范围选择                                                   |
| M41、...M44、<br>M141、...M241、..<br>M244 | -      | -              | 手动档位范围选择                                                   |
| M48、M148、M248                          | -      | -              | 脱开齿轮                                                       |
| M5、M105、M205                           | -      | -              | 主轴停止                                                       |
| MainSp                                 | MSP    | -              | 选择主轴，例如 为 G33 和 G95                                        |
| MCA                                    | -      | CPL            | MCA(<Block>, <Index>[, <channel>]):<br>传输单个 MACODA 参数的内容。  |

## 附录

| 数控函数名称 |      | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                                                                                                                                                             |
|--------|------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 全称     | 缩写形式 |                         |                                                                                                                                                                                                                                   |
| MCODS  | -    | CPL                     | MCODS(<type>, <channel>, <version>, <buffer>, <size>[, <P1>]):<br>调用 NCS 的运动控制数据服务。因此, 可以从 NC 读取数据和状态。                                                                                                                            |
| MCOPS  | -    | CPL                     | MCOPS(<Fct>, <channel>[[, [<P1>][, [<P2>], [<P3>]]], <P4>))<br>调用 NCS 的运动控制过程服务。这可以控制 NC 中的通道。                                                                                                                                    |
| MCS    | -    | CPL                     | MCS(<coordinate>[, <selection type>[, <channel>]]):<br>为机床坐标提供当前机床位置 (MCS), 无需进行直线度和角度误差修正。                                                                                                                                       |
| MID\$  | -    | CPL                     | MID\$(<string expression>, <starting point><br>[, <no. of characters>]):<br>接受 STRING 表达式的一部分并将其输出为文本。<br>结果可以传输到 STRING 变量或适当尺寸的字符数组。<br>MID\$(<character field>, <starting point>[, <number of<br>characters>]):<br>覆盖字符数组的一部分。 |
| Mirror | MIR  | 镜像                      | MIR(X1, Y1, Z1,...): 镜像坐标 ON<br>MIR(): 镜像坐标关闭                                                                                                                                                                                     |
| MMC    | -    | CPL                     | MMC(<CPL-Var1>[, <CPL-Var2>...<br>[, <CPL-VarN>]...]):<br>将程序运行时间的信息从子程序发送到客户端, 并等待此客户端的结果。                                                                                                                                       |
| MOC    |      |                         | MOC(MP<probe> CI<customer input> HS<HighSpeed>{, ES<edge>},<br>RE<response>{, AS<Asup>}{, ER<error>}{, <channel axis<br>name1>{, <channel axis name2> ...}))<br>测量轮廓                                                              |
| 模      | MOD  | -                       | 用于递增横越无限轴的模数计算                                                                                                                                                                                                                    |
| MPEDGE |      |                         | MPEDGE(MP1 MP2{, ES1 ES2})<br>更改测头的边缘配置                                                                                                                                                                                           |
| MSG    | -    | -                       | (MSG<Information text>)<br>用户界面中的程序笔记                                                                                                                                                                                             |
| NCF    | -    | CPL                     | NCF(<NC function>)<br>传送 NC 模式组中的激活数控函数的语法<br><NC-function>.                                                                                                                                                                      |
| Nibble | NIB  | 冲压                      | NIB( NUM...): 下料流程开启<br>NIB(): 半字节处理关闭                                                                                                                                                                                            |
| NJUST  | -    | CPL                     | 之前从左对齐数据输出重置为格式化输出。                                                                                                                                                                                                               |
| 否      | -    | CPL                     | 不 <expression>:<br>否定 BOOLEAN 表达式或 INTEGER 表达式的按位否定。                                                                                                                                                                              |
| NUL    | -    | CPL                     | <Variable> = NUL:<br>删除变量                                                                                                                                                                                                         |

## 附录

| 数控函数名称   |      | 组              | 说明/解释                                                                                                                                          |
|----------|------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 全称       | 缩写形式 | (“ - ”表示：块状激活) |                                                                                                                                                |
| NUM      | -    | -              | NUM=<value>:<br>将运行块分割成定义数量的等长段。                                                                                                               |
| 0        | 0    | -              | 0(...): 刀具定位运动                                                                                                                                 |
| OFFSTOPA | -    | -              | 通道同步命令:<br>删除控制通道 (ASTOPA、BSTOPA、WSTOPA) 中编程的“AND 停止条件”。                                                                                       |
| OFFSTOP0 | -    | -              | 通道同步命令:<br>删除控制通道中已编程的“OR 停止条件” (ASTOP0、BSTOP0、WSTOP0)。                                                                                        |
| Omega    | -    | -              | Omega 地址                                                                                                                                       |
| OPENR    | -    | CPL            | OPENR(<n>,<Prog-Name>[,<Record length>]):<br>开启文件进行以下只读访问。                                                                                     |
| OPENW    | -    | CPL            | OPENW(<n>,<Progr-Name>[, [<empty param.>][,<Comment>][,<Record length>]]):<br>开启文件以进行以下写入访问。                                                   |
| OR       | -    | CPL            | <expression> OR <expression>:<br>用 OR 操作对两个 BOOLEAN 或 INTEGER 表达式进行二进制操作。                                                                      |
| OvrDis   | OVD  | 进给 100%        | 进给电位计关闭                                                                                                                                        |
| OvrEna   | OVE  | 进给 100%        | 进给电位计开启                                                                                                                                        |
| P        | P    | -              | 子程序调用 (也可以直接没有 P)                                                                                                                              |
| PL       | -    | -              | 编写样条参数长度                                                                                                                                       |
| PathAcc  | PAC  | 路径加速度          | PAC(ACC...   UP..., DOWN ...): 开启<br>PAC(): 关闭<br>路径加速编程, 分别用于加速和减速或组合。                                                                        |
| PCS      | -    | CPL            | PCS(<coordinate>[,<selection type>]):<br>提供坐标的最新编程绝对位置。                                                                                        |
| PCSPROBE | -    | CPL            | PCSPROBE (<coordinate>[,<selection type>]):<br>读取每个坐标的测量值                                                                                      |
| PDIM     | -    | CPL            | PDIM <ParameterName>(<ArraySize>)<br>必须在子程序中使用:<br><ul style="list-style-type: none"> <li>是用一个字符串常量作为调用参数调用和</li> <li>调用程序被选中而没有链接。</li> </ul> |
| PLC      | -    | CPL            | PLC(<type>,<empty param.>,<address>,<size>):<br>该功能用于访问 PLC 的操作数。                                                                              |

| 数控函数名称      |      | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                           |
|-------------|------|-------------------------|-------------------------------------------------------------------------------------------------|
| 全称          | 缩写形式 |                         |                                                                                                 |
| PMT         | -    | CPL                     | PMT(<coord. selection>,<Pl code><br>[, [<PlBank>][, [<Table>][, <Unit>]]]):<br>读取/写入 xml 展示位置表。 |
| PmTSel      | PMS  | -                       | PMS(<Placement table name>):<br>选择放置表                                                           |
| PMV         | -    | CPL                     | PMV(<coord. selection>[, <Pl bank>]):<br>为坐标提供有效的位置值。                                           |
| PolarPol    | POP  | PolarPol                | POP(X..., Y..., Z...): 设定极点<br>POP(): 原点极点<br>极坐标编程的极点。                                         |
| PoleSet     | PLS  | 设定极点                    | PLS(X..., Y..., Z...): 设定极点<br>PLS(): 原点极点<br>镜像/旋转输入帮助杆。                                       |
| PosDepHSOut | PHS  | -                       | PHS(<mode>{, {<distance>}{, <duration>}}):<br>可编程的与位置有关的高速输出                                    |
| PosMode     | PMD  | 定位类型                    | PMD(A..., B...): 开关定位类型<br>PMD(): 定位类型 acc. 加工参数<br>循环轴的定位类型                                    |
| PPOS        | -    | CPL                     | PPOS(<axis selection>[, <axis type>]):<br>查询测头切换点的轴位置。                                          |
| PrecProg    | PRP  | 精度编程                    | PRP(DIST...   EPS...): 精度编程开启<br>PRP(): 精度编程关闭                                                  |
| PREP        | -    | -                       | PREP(<BlkCnt> { , SILENT } )<br>程序处理取决于块准备                                                      |
| PRN#        | -    | CPL                     | PRN#(<n>[, <expression>][, <expression>]<br>[, <expression>][, ...][;]):<br>在文件上写入访问权限。         |
| PROBE       | -    | CPL                     | PROBE(<axis selection>[, <axis type>]):<br>查询与 MCS 的轴零点坐标相关的测头切换点的轴位置。                          |
| PtBlkEnd    | PTE  | 冲压中风                    | PTE(X..., Y..., Z):<br>在程序段结束时触发行程触发器                                                           |
| PtDefault   | PTD  | 冲压中风                    | 根据参数化触发行程触发                                                                                     |
| PtInpos     | PTI  | 冲压中风                    | PTI(X..., Y..., Z):<br>Inpos 的冲程触发 (Pt =冲床时间)                                                   |
| PtpMove     | PTP  | -                       | 在主动轴转换的情况下, 执行点对点运动 (PTP)。                                                                      |
| Punch       | PUN  | 冲压                      | PUN( NUM...): 打孔处理开启<br>PUN(): 打孔处理关闭                                                           |

## 附录

| 数控函数名称       |      | 组              | 说明/解释                                                                             |
|--------------|------|----------------|-----------------------------------------------------------------------------------|
| 全称           | 缩写形式 | (“ - ”表示：块状激活) |                                                                                   |
| RadProg      | RAD  | 直径/半径          | 半径编程                                                                              |
| RadialAcc    | RAC  | —              | RadialAcc (<value>)：设置最大径向加速度<br>RadialAcc(0)：取消径向加速度限制                           |
| RecordRevVec | RRV  | —              | RRV(...)<br>记录缩回向量                                                                |
| RedTorque    | RDT  | —              | RDT(<axis1>{, <axis2>{, ...}}):<br>扭矩减小的扭矩规格                                      |
| ReentContBlk | RCB  | —              | 程序中中止后继续程序块预运行                                                                    |
| REM          | —    | CPL            | REM <CommentText><br>程序注释                                                         |
| RemAxis      | RAX  | —              | RAX(...):<br>从通道中清除轴                                                              |
| RemLogName   | RLN  | —              | RLN(...):<br>清除逻辑轴名称                                                              |
| RemSpindle   | RSP  | —              | RemSpindle (<SysSpNo>   <SysSpName>   <ChanSpName> , {...})<br>清除通道主轴             |
| REPEAT       | —    | CPL            | REPEAT <Routine> UNTIL <Condition>:<br>第一次处理程序后，查询中止条件的循环结构。                      |
| REPEAT       | RPT  |                | RPT<repetition factor><br>阻止重复                                                    |
| REPOSDEF     | —    | —              | 在本地定义返回到轮廓进给和/或返回到轮廓的点（在异步子程序中）。                                                  |
| REPOSTP      | —    | —              | REPOSTP(<point>)<br>异步子程序：在异步子程序中定义返回轮廓点。                                         |
| 列岛           | —    | —              | 子程序返回到任何程序块                                                                       |
| REV          | —    |                | REV(<EventNo> { , <EventNo> ...}):<br>运行时删除一个事件。                                  |
| REWRITE      | —    | CPL            | REWRITE(<n>):<br>覆盖现有文件。                                                          |
| 旋转           | ROT  | 旋转             | ROT(<angle>): 坐标旋转开启<br>ROT( )：坐标旋转关闭                                             |
| ROTAX        | —    | —              | 为矢量方向定义旋转轴                                                                        |
| ROUND        | —    | CPL            | <Integer number> = ROUND(<Real expression>):<br>向上或向下舍入，将 REAL 表达式转换为 INTEGER 数字。 |
| RoundEps     | RNE  | 倒角/倒圆          | RNE(<value>): 倒圆开启<br>RNE( )：倒圆关闭<br>以确定的轮廓移位进行舍入。                                |

| 数控函数名称          |      | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                                           |
|-----------------|------|-------------------------|-----------------------------------------------------------------------------------------------------------------|
| 全称              | 缩写形式 |                         |                                                                                                                 |
| 倒圆              | RND  | 倒角/倒圆                   | RND(<value>)：倒圆开启<br>RND()：倒圆关闭<br>以定义的倒圆半径倒圆                                                                   |
| S               | S    | -                       | S<number>=<value><br>S<value>：<br>编程主轴速度                                                                        |
| 规模              | SCL  | 缩放                      | SCL (X ..., Y ..., Z)：坐标缩放开启<br>SCL ()：坐标缩放关闭                                                                   |
| SCL             | -    | CPL                     | SCL(<SCL selection>[, <selection><br>[, <selection type>]]):<br>为当前通道提供 PLS 和 ROT 功能（极坐标，缩放因子和旋转角度）<br>的最新编程参数。 |
| SCS3            | -    | CPL                     | SCS3(<Axis index>, <EID-No><br>[, <ERRNO>]):<br>只读访问 Sercos 使用扩展标识号驱动参数。                                        |
| SCSL3           | -    | CPL                     | SCSL3(<Axis index>, <EID-No><br><File name>[, <ERRNO>]):<br>使用扩展标识号为 Sercos 参数列表创建文件。                           |
| SD              | -    | CPL                     | SD(<group>[, <Index1>[, <Index2><br>[, <Index3>]]]):<br>读取 NC 控制的有效系统数据                                         |
| SDLOAD          | -    | CPL                     | SDLOAD(<FileName>, <Root-Element_1>[, ..., <Root-Element_n>])<br>从文件读取系统数据                                      |
| SDR             | -    | CPL                     | SDR(<group>[, <Index1>[, <Index2>]]):<br>以 REAL 格式读取 NC 控制系统的有效系统数据。                                            |
| SDSAVE          | -    | CPL                     | SDSAVE(<FileName>, <Root-Element_1>[, ..., <Root-Element_n>])                                                   |
| SEEK            | -    | CPL                     | SEEK(<n>, <k>[, <o>]):<br>将文件指针放在随机文件的第 <k> 个记录或顺序文件的第 k 个字节<br>上。                                              |
| SelCrdCouple    | SCC  | -                       | SCC(SC<channel>, CL(Q1>, <Z1>, {...})):<br>选择附加坐标耦合 ON<br>SCC()：所有坐标耦合关闭                                        |
| SelCrdCoupleTab | SCCT | -                       | 使用表的选择性叠加坐标耦合                                                                                                   |
| SETERR          | -    | CPL                     | 在评估 ERRNO 后生成运行时错误。                                                                                             |
| SetPos          | PLC  | -                       | 设定程序位置                                                                                                          |
| SETWARN         | -    | CPL                     | 在评估 ERRNO 后生成警告。                                                                                                |
| SEV             | -    |                         | SEV(<EventNo> {, <EventNo> ...}):<br>一个位事件在运行时被触发（设置）。                                                          |

## 附录

| 数控函数名称               |           | 组              | 说明/解释                                                                     |
|----------------------|-----------|----------------|---------------------------------------------------------------------------|
| 全称                   | 缩写形式      | (“ - ”表示：块状激活) |                                                                           |
| Shift                | SHT       | —              | SHT(X..., Y..., Z): 轮廓切换开启<br>SHT(): 轮廓切换关闭                               |
| Sin2DownFeed         | S2D       | V-profile      | Sine <sup>2</sup> 线型的减速                                                   |
| Sin2UpFeed           | S2U       | V-profile      | Sine <sup>2</sup> 线型的加速                                                   |
| SinDownFeed          | SND       | V-profile      | 正弦曲线型的减速                                                                  |
| SinUpFeed            | SNU       | V-profile      | 正弦曲线型加速                                                                   |
| Smax                 | SMX       | —              | SMX(<value>):<br>主轴的最大速度。                                                 |
| Smin                 | SMN       | —              | SMN(<value>):<br>主轴的最小速度。                                                 |
| SpAdjSp              | —         | —              | SpAdjSp<br>在块预运行初始化调整程序中调整主轴                                              |
| SpAdjStr             | —         | —              | SpAdjStr<Spindle><br>确定程序段预运行调整程序的主轴语法                                    |
| SpAdmin              | SPA       | —              | SPA(Si=0 1):<br>通过另一个通道有条件释放或应用保留的主轴。                                     |
| SpCouple_Wait        | SPC_WAIT  | —              | SPC_WAIT(CP=1..4):<br>等待指定的偶数组的同步模式。                                      |
| SpCoupleConfig       | SPCC      | —              | SPCC(CP=1..4, MA=<Master>, Si=1, Sj=1, ...):<br>定义耦合组, 删除或添加跟随主轴, 取消耦合组。  |
| SpCoupleDist         | SPCD      | —              | SPCD(S1=<distance>, S2=<distance>, ...):<br>定义主轴联轴器的联轴距离。                 |
| SpCoupleErrWin       | SPCE      | —              | SPCE(S1=<window>, S2=<window>, ...):<br>定义同步运行错误窗口<br>定义主轴联轴器。            |
| SpCouplePosOffs      | SPCP      | —              | SPCP(S1=<offset>, S2=<offset>, ... {POSVEL<speed>}):<br>为主动主轴联轴器组定义角度偏移。  |
| SpCouplePosOffs_Wait | SPCP_WAIT | —              | SPCP_WAIT(CP=1..4):<br>等待编程的联轴器组的角度偏移。                                    |
| SpCoupleSyncWin      | SPCS      | —              | SPCS(S1=<window>, S2=<window>, ...):<br>定义同步窗口主轴联轴器。                      |
| SPG                  | —         | —              | SPG<group>(<numbers>)<br>SPGALL(0):<br>定义/取消主轴组                           |
| SplineCornering      | SCO       | —              | SplineCornering(E<lag>)<br>SplineCornering(L1=<L1>, L2=<L2>):<br>使用样条修圆转角 |



附录

| 数控函数名称               |        | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                                            |
|----------------------|--------|-------------------------|------------------------------------------------------------------------------------------------------------------|
| 全称                   | 缩写形式   |                         |                                                                                                                  |
| SplineCorneringLocal | SCOL   | -                       | 使用样条（本地）修圆转角                                                                                                     |
| SplineDef            | SPD    | -                       | 规定样条变形                                                                                                           |
| SpindleToAxis        | STA    | -                       | STA(...)<br>从主轴切换到轴模式。                                                                                           |
| SpindleToAxisWait    | STAW   | -                       | SpindleToAxisWait(<PAN> <PAI>{ , <PAN> <PAI>}...)<br>开启主轴的 C 轴模式并等待                                              |
| 对分                   | SPT    | -                       | Split({<Mode>} {, <partial length>}):<br>如果超过一定的长度，则将运动块分割成若干部分份。                                                |
| SpindleWait          | SPWAIT | -                       | SpindleWait(<Spindle> { , <Condition>})<br>在每个程序段结束时等待，直到各自的主轴达到所需的状态。                                           |
| SpindleWaitMode      | SWM    | -                       | SpindleWaitMode(<Spindle>, <Switch on> {, <Strategy>):<br>功能等待策略“路径运动和主轴功能”                                      |
| SpMode               | SPM    | -                       | SPM(S1=0 1, S2=0 1, ...):<br>在速度模式和位置模式之间手动切换主轴驱动接口                                                              |
| SPOS                 | -      | CPL                     | SPOS(<axis selection>):<br>转移某个物理轴的当前轴命令值。                                                                       |
| SPV                  | -      | -                       | SPV [ <Perm. CPL-Var.> =<br><simple CPL expression> ]<br>通道同步命令：<br>永久 CPL 变量在运行时写入。                             |
| SPVE                 | -      | -                       | SPV E[ <Perm. CPL var.> = <CPL expression> ]<br>通道同步命令：<br>永久 CPL 变量在运行时写入。<br>CPL 表达式可以包括 CPL 命令，但在块准备时间期间已经终止。 |
| SQRT                 | -      | CPL                     | <Function value> = SQRT(<input value>):<br>将平方根函数应用于 <input value>。                                              |
| SSD                  | -      | CPL                     | SSD[<SystemDate> =<simple CPL expression>]<br>运行时确定将赋予的值。                                                        |
| SSDE                 | -      | CPL                     | SSDE[<System date> =<CPL expression>]<br>在准备时确定将被赋予的值：                                                           |
| SSDQ                 | -      | -                       | SSDQ(M <Message> {, Q <XPath>} {, C <XPath>})<br>写入系统数据队列                                                        |
| SSpAdm               | -      | -                       | SSpAdm(<SSpNo.>   <SSpName>, <Mode><br>{ , <SSpNo.>   <SSpName>, <Mode>} ...):<br>释放/使保留的主轴生效                    |

## 附录

| 数控函数名称          |      | 组               | 说明/解释                                                                                                                                        |
|-----------------|------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 全称              | 缩写形式 | (“ - ”表示: 块状激活) |                                                                                                                                              |
| SSpAdmOff       | -    | -               | SSpAdmOff(<SSpNo.>   <SSpName>,<br><Channel no.> {, <Channel no.> } ... )SSpAdmOff(<SSpNo.>   <SSpName>):<br>关闭指定通道的主轴运动管理。                  |
| SSPG            | -    | -               | SSPG<group>=<value>:<br>设定主轴组速度, 另请参见 S<value>                                                                                               |
| SSpGear         | -    | -               | SSpGear(<SSpNo.>   <SSpName>, <step>{, <SSpNo.>   <SSpName>,<br><step>} ... ):<br>改变档位范围                                                     |
| SSpMax          | -    | -               | SSpMax(<SSpNo.>   <SSpName>, <speed>{, <SSpNo.>   <SSpName>,<br><speed>} ... ):<br>限制最大速度                                                    |
| SSpMin          | -    | -               | SSpMin(<SSpNo.>   <SSpName>, <speed>{, <SSpNo.>   <SSpName>,<br><speed>} ... ):<br>限制最小速度                                                    |
| SSpMode         | -    | -               | SSpMode(<SSpNo.>   <SSpName>, <Mode> {, <SSpNo.>   <SSpName>,<br><Mode>} ... ):<br>在位置/速度接口之间切换                                              |
| SSpMove         | SSPM | -               | SSpMove(<SSpNo.>   <SSpName>, <task>{, <SSpNo.>   <SSpName>,<br><task>} ... ):<br>运动编程                                                       |
| SSpOri          | SSPO | -               | SSpOri(<SSpNo.>   <SSpName>, <angle>{, <SSpNo.>   <SSpName>,<br><angle>} ... ):<br>主轴定向                                                      |
| SSpSpeed        | SSPS | -               | SSpAdmOff(<SSpNo.>   <SSpName>,<br><Speed> {, <SSpNo.>   <SSpName>,<br><Speed>} ... ):<br>速度编程                                               |
| StatToolOri     | STO  | -               | StatToolOri(({<Φ>} {, {<Θ>} {, <Ψ>}}))<br>静态刀具定位参数设置                                                                                         |
| StepModeDisable | SMD  | -               | StepModeDisable(({Variable(=) {Value}} {, } {Variable(=) {Value}}))):<br>用于在一个操作模式下一步处理 NC 程序的各个部分: 单步, 单块, 程序块和调试步骤。这意味着块在没有确认 NC 启动的情况下被处理 |
| StepModeEnable  | SME  | -               | StepModeEnable<br>没有参数可以编程。块中的 SME 被编程 - 是 NC 启动被禁止的块之一。                                                                                     |

| 数控函数名称         |      | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                                                                                                                      |
|----------------|------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 全称             | 缩写形式 |                         |                                                                                                                                                                                            |
| STR\$          | -    | CPL                     | STR\$([<Format string>,<value>):<br>把数值表达式 <Value> 转变为字符串, 而且只能分配给字符数组。<br>如果 <Format String> 已编程, 则可以用格式化形式对该字符串进行输出。                                                                     |
| TAN            | -    | CPL                     | <Function value> = TAN(<input value>):<br>将切线函数应用于 <input value>。                                                                                                                          |
| TangTool       | TTL  | 刀具定向。                   | TTL(TAX..., SYM..., ANG..., IA..., PLC...):<br>切向刀具引导开启<br>TTL(): 切向刀具引导关闭                                                                                                                 |
| TangToolOri    | TTO  | 切向刀具定向。                 | TTO(SYM..., ANG...): 切向刀具定向开启<br>TTO(): 切向刀具定向关闭                                                                                                                                           |
| TappRet1       | -    | -                       | 在“控制器复位”(控制器启动)之后, 将主轴速度模式下的主轴运行再次切换到位置模式。                                                                                                                                                 |
| TappRet2       | -    | -                       | 利用编程的 F 值启动实际的回缩运动。                                                                                                                                                                        |
| TappSp         | TSP  | -                       | TSP(CAXi, ..., GRPj, ...):<br>无补偿卡盘的攻丝的主轴选择。                                                                                                                                               |
| TargetAxisConf | TAX  |                         | TAX 或 TargetAxisConf<br>轴配置调整                                                                                                                                                              |
| TCM( , , )     | -    | -                       | TCM({<xVal>}, {<yVal>}, {<zVal>}):<br>在 TCS 中的端面运动                                                                                                                                         |
| TcsDef         | TCS  | -                       | 明确直接规定刀具坐标系统的位置(结合对应轴变换进行固有刀具长度修正)。                                                                                                                                                        |
| TCV            | -    | CPL                     | TCV(<value selection>[, <correction selection>]):<br>提供最新的编程刀具修正值                                                                                                                          |
| ThreadSet      | TST  | -                       | 螺纹切削, 附加功能                                                                                                                                                                                 |
| 时间             | -    | CPL                     | <String variable> = TIME:<br>将时间分配给一个 STRING 变量, 如 HH、MM、SS。                                                                                                                               |
| Trans          | TRS  | 程序坐标偏置                  | TRS( X ..., Y ..., Z ):<br>坐标偏置编程开启<br>TRS():<br>坐标偏置编制关闭                                                                                                                                  |
| TRIM\$         | -    | CPL                     | TRIM\$(<CharacterString>)<br>TRIM\$(<character string>," L")<br>TRIM\$(<character string>," R").<br>为了将字符数组范围赋值给 STRING 变量或字符数组, 它提供了一个没有前面(索引“L”)或后面(索引“R”)空间的字符串。如果没有索引, 则前面和后面的空格都是隐藏的。 |

## 附录

| 数控函数名称      |      | 组              | 说明/解释                                                                                                                                        |
|-------------|------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 全称          | 缩写形式 | (“ - ”表示：块状激活) |                                                                                                                                              |
| 真           | -    | CPL            | <Boolean variable> = TRUE:<br>布尔变量的真值                                                                                                        |
| VAL         | -    | CPL            | VAL(<string expression>):<br>返回 STRING 表达式的数值。                                                                                               |
| VARINF      | -    | -              | VARINF(<Variable name>)<br>结构化系统日期或永久变量的名称。                                                                                                  |
| VERSINF\$   | -    | CPL            | VERSINF\$(<Index1>[, <Index2>]):<br>呼叫管理数据, 例如 软件版本。                                                                                         |
| VirtAxisPos | VAP  | -              | 设置当前通道中虚拟同步轴的轴位置。                                                                                                                            |
| VREC_START  | -    | -              | 开始速度控制的诊断记录                                                                                                                                  |
| VREC_STOP   | -    | -              | 退出诊断记录。                                                                                                                                      |
| WAIT        | -    | -              | 在数控段中等待 (如 CPL WAIT)                                                                                                                         |
| WAIT        | -    | CPL            | WAIT (无参数):<br>停止块准备, 直到处理完 WAIT 之前编程的所有块为止。<br>等待 (<idle time>):<br>停止阻止处理, 直到经过指定的时间。<br>WAIT (BITIF(...)):<br>停止块准备, 直到 PLC-NC 位接口发生某种状态。 |
| WAITA       | -    | -              | WAITA[BITIF(<Parameter>){=<State>} {,...}{,<Timeout>}]<br>通道同步命令:<br>程序在运行时, 等待所有查询的接口信号给出或超时。                                               |
| WaitAxis    | WAX  | -              | WAX(...):<br>等到轴释放, 然后将其应用到通道。                                                                                                               |
| WAITO       | -    | -              | WAITO[BITIF(<Parameter>){=<State>} {,...}{,<Timeout>}]<br>通道同步命令:<br>程序在运行时, 等待查询到的接口信号之一或超时。                                                |
| WCS         | -    | CPL            | WCS(<coordinate>[, <selection type>[, <channel>]]):<br>提供当前工件位置, 而不用在线修正坐标值。                                                                 |
| WEV         | -    |                | WEV(<EventNo> {, EventNo ...}):<br>通过 WEV 功能, 在运行时等待 (或连续模块切换受阻), 直到发生某个位事件。                                                                 |
| WHILE       | -    | CPL            | WHILE <condition> DO <routine>END<br>在第一次迭代循环之前查询中止条件的循环构造。                                                                                  |
| WPV         | -    | -              | WPV[<Perm.CPL var.><Comparison op.><br><simp.CPL expression>{,<Timeout>}]<br>通道同步命令:<br>程序在运行时, 请等到表达式满足或超时。                                 |

| 数控函数名称   |      | 组<br>(“ - ”表示：块<br>状激活) | 说明/解释                                                                                                                                   |
|----------|------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 全称       | 缩写形式 |                         |                                                                                                                                         |
| WPVE     | —    | —                       | WPVE[<Perm.CPL var.><Comparison op.><br><CPL expr.>{,<Timeout>}]<br>通道同步命令：<br>程序在运行时，请等到表达式满足或超时。<br>CPL 表达式可以包括 CPL 命令，但在块准备时间期间已经终止。 |
| WriteId3 | WID3 | —                       | WID3(S-0-0104.0.0, X..., Y..., Z):<br>使用扩展标识号编写 Sercos 参数                                                                               |
| WSTOPA   | —    | —                       | WSTOPA[<channel no.>,<cond.1>{,<cond.2>} {...,<cond.8>}...]<br>通道同步命令：<br>只要满足所有条件，要控制的通道就停止。条件是指工件位置，<br>例如 “Z” > 20.                  |
| WSTOPO   | —    | —                       | WSTOPO[<channel no.>,<cond.1>{,<cond.2>} {...,<cond.8>}...]<br>通道同步命令：<br>只要满足其中一个条件，要控制的通道就会停止。条件涉及工件位置，<br>例如 “Z” > 20.               |
| XOR      | —    | CPL                     | <expression> XOR <expression>:<br>使用 EXCLUSIVE OR 操作对两个 BOOLEAN 或 INTEGER 表达式进行二进制操作。                                                   |
| XTAB     | —    | CPL                     | XTAB(<Table>,<Substructure>):<br>对 DCT、ZOT、PMT 或 GCT 类型的任何 XML 表进行读写访问。                                                                 |
| XTABCR   | —    | CPL                     | XTABCR(<Table>,<Type><br>[,<TableTemplate>]):<br>在 MTX 文件系统中创建一个没有数据的 XML 表。                                                            |
| ZOCDEL   | —    | CPL                     | ZOCDEL(<table>,<position>[,<channel>]):<br>从 xml 零偏移量表中删除一个轴。                                                                           |
| ZOCINS   | —    | CPL                     | ZOCINS(<table>,<position>,<axis name><br>[, [<AxisType>][,<AxisChannel>][,<PosChannel>]]):<br>在 XML 零点偏移表中的现有列前插入一个新轴。                  |
| ZOT      | —    | CPL                     | ZOT(<column selection>,<ZO code><br>[, [<ZOBank>][,<Table>][,<Unit>]]):<br>读取和写入任何 xml 零点偏移表的访问权限。                                      |
| ZOTCR    | —    | CPL                     | ZOTCR(<table>,<channel/template>):<br>在 MTX 文件系统中创建一个没有修正集的 XML 零点偏移表。                                                                  |

附录

| 数控函数名称 |      | 组                    | 说明/解释                                                    |
|--------|------|----------------------|----------------------------------------------------------|
| 全称     | 缩写形式 | ( “ - ” 表示：块<br>状激活) |                                                          |
| ZoTSe1 | ZOS  | –                    | ZOS(<ZO name>) :<br>选择零点偏移表。                             |
| ZOV    | –    | CPL                  | ZOV(<axis selection>[.<ZO bank>]):<br>为轴/机器坐标提供有效的 ZO 值。 |

表格 8-1:            按字母顺序排列

8.2        ASCII 字符集

ASCII 代表“美国信息交换标准代码”。字符 00 至 31 和字符 127 是控制信号

| 十进制 | 十六进制 | ASCII |
|-----|------|-------|
| 0   | 00   | NUL   |
| 1   | 01   | SOH   |
| 2   | 02   | STX   |
| 3   | 03   | ETX   |
| 4   | 04   | EOT   |
| 5   | 05   | ENQ   |
| 6   | 06   | ACK   |
| 7   | 07   | BEL   |
| 8   | 08   | BS    |
| 9   | 09   | HT    |
| 10  | 0A   | LF    |
| 11  | 0B   | VT    |
| 12  | 0C   | FF    |
| 13  | 0D   | CR    |
| 14  | 0E   | SO    |
| 15  | 0F   | SI    |
| 16  | 10   | DLE   |
| 17  | 11   | DC1   |
| 18  | 12   | DC2   |
| 19  | 13   | DC3   |
| 20  | 14   | DC4   |
| 21  | 15   | NAK   |
| 22  | 16   | SYN   |
| 23  | 17   | ETB   |
| 24  | 18   | CAN   |
| 25  | 19   | EM    |

附录

| 十进制 | 十六进制 | ASCII |
|-----|------|-------|
| 26  | 1A   | SUB   |
| 27  | 1B   | ESC   |
| 28  | 1C   | FS    |
| 29  | 1D   | GS    |
| 30  | 1E   | RS    |
| 31  | 1F   | US    |
| 32  | 20   | SP    |
| 33  | 21   | !     |
| 34  | 22   | “     |
| 35  | 23   | #     |
| 36  | 24   | \$    |
| 37  | 25   | %     |
| 38  | 26   | &     |
| 39  | 27   | ‘     |
| 40  | 28   | (     |
| 41  | 29   | )     |
| 42  | 2A   | *     |
| 43  | 2B   | +     |
| 44  | 2C   | ,     |
| 45  | 2D   | -     |
| 46  | 2E   | .     |
| 47  | 2F   | /     |
| 48  | 30   | 0     |
| 49  | 31   | 1     |
| 50  | 32   | 2     |
| 51  | 33   | 3     |
| 52  | 34   | 4     |
| 53  | 35   | 5     |
| 54  | 36   | 6     |
| 55  | 37   | 7     |
| 56  | 38   | 8     |
| 57  | 39   | 9     |
| 58  | 3A   | :     |
| 59  | 3B   | ;     |
| 60  | 3C   | <     |
| 61  | 3D   | =     |
| 62  | 3E   | >     |
| 63  | 3F   | ?     |

## 附录

| 十进制 | 十六进制 | ASCII |
|-----|------|-------|
| 64  | 40   | @     |
| 65  | 41   | A     |
| 66  | 42   | B     |
| 67  | 43   | C     |
| 68  | 44   | D     |
| 69  | 45   | E     |
| 70  | 46   | F     |
| 71  | 47   | G     |
| 72  | 48   | H     |
| 73  | 49   | I     |
| 74  | 4A   | J     |
| 75  | 4B   | K     |
| 76  | 4C   | L     |
| 77  | 4D   | M     |
| 78  | 4E   | N     |
| 79  | 4F   | O     |
| 80  | 50   | P     |
| 81  | 51   | Q     |
| 82  | 52   | R     |
| 83  | 53   | S     |
| 84  | 54   | T     |
| 85  | 55   | U     |
| 86  | 56   | V     |
| 87  | 57   | W     |
| 88  | 58   | X     |
| 89  | 59   | Y     |
| 90  | 5A   | Z     |
| 91  | 5B   | [     |
| 92  | 5C   | \     |
| 93  | 5D   | ]     |
| 94  | 5E   | ↑ (^) |
| 95  | 5F   | → ( ) |
| 96  | 60   | ‘     |
| 97  | 61   | a     |
| 98  | 62   | b     |
| 99  | 63   | c     |
| 100 | 64   | d     |
| 101 | 65   | e     |



| 十进制 | 十六进制 | ASCII |
|-----|------|-------|
| 102 | 66   | f     |
| 103 | 67   | g     |
| 104 | 68   | h     |
| 105 | 69   | i     |
| 106 | 6A   | j     |
| 107 | 6B   | k     |
| 108 | 6C   | l     |
| 109 | 6D   | m     |
| 110 | 6E   | n     |
| 111 | 6F   | o     |
| 112 | 70   | p     |
| 113 | 71   | q     |
| 114 | 72   | r     |
| 115 | 73   | s     |
| 116 | 74   | t     |
| 117 | 75   | u     |
| 118 | 76   | v     |
| 119 | 77   | w     |
| 120 | 78   | x     |
| 121 | 79   | y     |
| 122 | 7A   | z     |
| 123 | 7B   | {     |
| 124 | 7C   |       |
| 125 | 7D   | }     |
| 126 | 7E   | ~     |
| 127 | 7F   | DEL   |

以黄色突出显示 读入时默认跳过的字符。  
表格 8-2: ASCII 字符组

### 8.3 其他关键代码

| 关键代码（十进制） | 描述       |
|-----------|----------|
| 134       | 光标向上     |
| 135       | 光标向下     |
| 136       | 光标向右     |
| 137       | 光标向左     |
| 139       | 平面返回     |
| 141       | SOFTKEY1 |
| 142       | SOFTKEY2 |

| 关键代码（十进制） | 描述       |
|-----------|----------|
| 143       | SOFTKEY3 |
| 144       | SOFTKEY4 |
| 145       | SOFTKEY5 |
| 146       | SOFTKEY6 |
| 147       | SOFTKEY7 |
| 148       | SOFTKEY8 |

表格 8-3: 补充关键代码

8.4

优化加工时间

8.4.1

直接影响加工速度的措施

简介

这些措施直接影响处理速度。

准确停止关闭的路径坡度 G8/G9 的"G62"

没有“路径坡度”，（G9），控制系统在运行程序段的起始和结束处执行一个完整的向上和向下斜坡（速度斜坡）到  $v=0$ 。

使用“路径斜坡”（G8），即使在程序段转换时，控制器也会尝试生成尽可能恒定的速度，与编程进给一样高。这会减少处理时间。

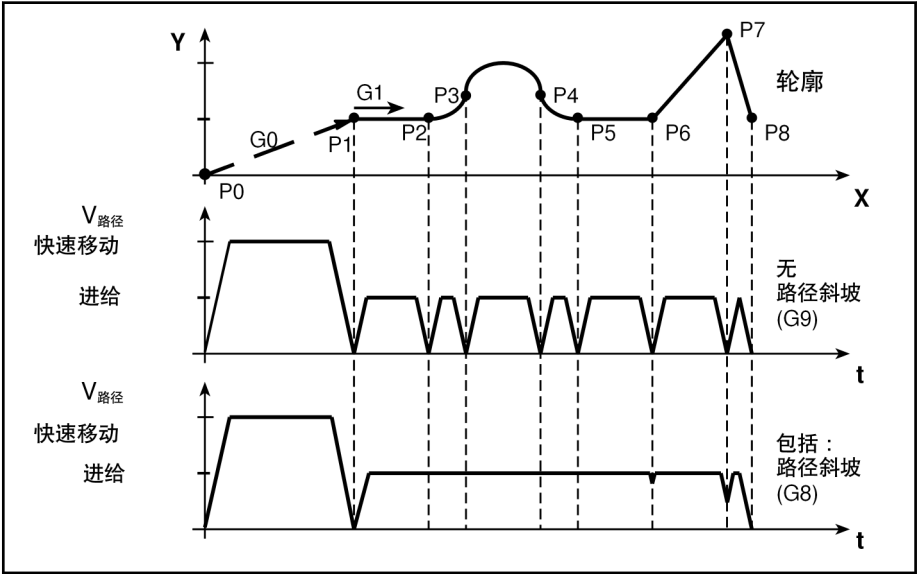


插图 8-1: 路径斜坡

为了使 G8 激活，必须关闭准确停止（G62）。

准停定位开启的 Inpos 窗口"G61"

“准停定位开启”仅用于真正需要等待达到确切位置的地方。

请注意，块转换需要更多时间才能到达 inpos 窗口，inpos 范围越小，相应位置控制回路的控制质量就越低。

|                          |
|--------------------------|
| G61 (IPS1   IPS2   IPS3) |
| 其中                       |

|            |                                                                                                                                                        |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| GOTO1<br>0 | 准确停止开启<br>等待 <b>精确定位窗口</b> 。<br><br>在块终点，控制器首先把路径速度降低到 v=0。只有在涉及的所有轴都达到此定位窗口时，才会进给移动下一块。                                                               |
| GOTO1<br>0 | 准确停止开启<br>等待 <b>粗略定位窗口</b> 。<br><br>“Inpos 范围 2 激活”显示在通道接口上（另见文档“Rexroth IndraMotion MTX PLC 接口”）。在块终点，控制器首先把路径速度降低到 v=0。只有在涉及的所有轴都达到此定位窗口时，才会进给移动下一块。 |
| GOTO1<br>0 | 准确停止开启<br><br>在块终止处减速到 v=0。<br><br>在块终止处，控制器将路径速度降低到 v=0。然后无需检查，把下一块进给运动到定位窗口。                                                                         |

表格 8-4:            可选参数 G61  
用于轮廓模式的最新激活定位窗口对于“准确停止开启”仍然有效。

在 G0 上的 Inpos 窗口

G61 仅影响进给中的运动。它对快速移动运动没有影响。  
G0 的 inpos 窗口 G0（NIPS）可以被禁用。

连续块转换

平滑，连续的转换比具有转角的转换更快。使用以下功能可以实现平滑过渡：

- SC0（使用样条修圆角）
- RND（使用半径修圆角）

数控段

一般情况下，即使没有编程运行运动（至少有一个插补周期），每个编程的数控段都会被传送到插补器并需要一定的执行时间。  
没有必要在在一个块中的进给信息中编入非进给信息。  
*例子*

```
减少插补虚幻
取代
N10 G91
N20 G1 F1000
N30 X100 Y50
应写入
N10 G91 G1 X100 Y50 F1000
```

Sercos 通讯

每个识别号码写入（）都需要处理时间并导致插值减速到 0。

自由曲面加工

自由曲面的数控段通常具有非常短的路径长度。为避免路径速度过度降低，建议使用机器参数启用“跳过数控段”功能。  
参数最”大数控块缓冲区的块数”（706000120）和”数控块缓冲区（706000130）的数量百分比”指定要跳过的块的最大数量。控制器通过窗口使

用设定的块数量得出各个块的平均速率。如果在一个窗口内至少存在一个具有较高最大速度的较长程序段，则该程序段会增加较短相邻程序段的最大速度，但前提是不超过极限值。

8. 4. 2 间接影响处理速度的措施

简介

如果在块循环时间限制处执行处理，这些度量仅影响处理速度。

短语法

零件程序中的每个编程字符都需要一定的运行时间。因此，短语法在块解释方面要比长语法形式要优越。

例子

短语法和长语法

FFW < FeedForward  
SCCT < SelCrdCoupleTab

模态函数

模态函数仅在其发生变化时进行编程。  
只要有可能，模态功能的修改将在一个数控段中编入运行信息。

例子

模态功能

N10 G90  
N20 1 X50 F1000  
N30 G90 G1 X70 Y20 F1000  
N40 G90 G1 X100 Y50 F1000  
N50 G91  
N60 G1 X-100 F1000  
N70 G91 G0 Y-50

例子

理想模态函数

N10 G90 G1 X50 F1000  
N20 X70 Y20  
N30 X100 Y50  
N40 G91 X-100  
N50 G0 Y-50

注释

- 使用 GOTO 对较长的注释段优化处理时间及跳过处理。  
10 GOTO .HEADER  
;=====  
; 在执行 NC 程序之前请参考文档  
;  
; ...  
; PRG 标题

- ```
;  
=====
```
- 20 .HEAD
 - 如果单行要标注为注释，则 a";"将在一行的开始处使用。

WAIT 命令

一般情况下，如果 CPL 部分访问插值的当前值（查询当前位置），则只能使用“WAIT”将 CPL 部分与前面的 DIN 部分分开。

如果只读取编程值或将数据写入 CPL 部分，则不需要额外的“WAIT”，因为 CPL 在块准备中以同步方式运行。

如果 CPL 数据被写入稍后由 DIN 部分处理（例如刀具修正），则在 CPL 部分之后也不需要“WAIT”。

以下函数具有内部 WAIT 功能

ATBWD	校准轴运动：向后转型
ATCAL	校准轴运动：优化参数
ATFWD	校准轴运动：前向转换
ATGET	校准轴运动：从 NC 读取参数
ATPUT	校准轴运动：将参数写入 NC
DCR(0)	禁用数字化的距离控制
EPC(0)	停用终点位置耦合
FSM	转到固定停止
FSP	在固定站点测量
FSR	重置“移至固定点”
G74(HOME)	接近参考点
G75、G75.1、G75.2	接触探头
HSB	通过高速信号终止切换
HWOC(0)	停用工件坐标的在线修正
LCP	迟滞补偿
SCC(0)	停用选择性叠加坐标耦合
SCCT(0)	停用使用表的选择性叠加坐标耦合
SCPOWAIT	新的主轴耦合偏置已启用
SCWAIT	等待同步主轴耦合
SPC_WAIT	等待同步模式
SPCP_WAIT	等待角度偏置
TappRet1、TappRet2	从窃听中退出
TSP	启用各主轴的攻丝
VAP	虚拟驱动器

文件访问

对已装入文件的访问始终比访问控制系统内部文件系统中的数据要慢。

在“usr/用户”下存储经常调用的程序或循环。

附录

等待永久 CPL 变量

如果等待某个事件，请先检查 CPL 循环的内部，然后使用命令“WPV”和“SPV”。

例子

WPV、SPV

10 WPV[@1=1]

20 SPV[@2=3]

"MCODS"版本参数

如果 MCODES 调用中的版本变量内容为 0，该函数立即提供请求的数据。每次数据修改都会增加版本。如果调用 MCODES 的版本内容 <> 0，该函数等待以下数据修改。

例子

MCODES 调用立即提供数据

10 VERSION = 0

20 重复

30 ERR = MCODES (32, 1, VERSION, SAVSTATE%, 1)

40 UNTIL SAVSTATE%=6

8.4.3 编程错误导致的处理延迟

简介

特殊编程语法能够造成无法预期的编程延误。由于编程功能的时间行为，即使程序在另一个通道中等待此事件，也可能已处理事件。该程序不会继续。

。通常，当从“主轴切换到轴”（STA）或反之（ATS）时，根据各自的命令，WAIT 将被编程为不带参数（例如 N10 等待）。

例子

应用主轴作为轴

N10 STA(C)

N20 WAIT

N30 WAX(C)

通过速度规格编程回转轴

通过速度指定编程旋转轴并返回到通道。需在 WAX 之前输入 WAIT。

例子

带速度规定的回转轴

N10 RAX(C1)

N20 C1=S(500)

N30 G4 F5

N40 C1=S(0)

N50 WAIT

N60 WAX(C1)

CPL 循环

由外部事件中止的 CPL 循环将通过“WAIT (,<WaitingTime>)”指令提供。由此，阻止优先级较低的任务（位置显示）。

例子

CPL 循环

10 重复

20...

30 WAIT(, 200)

40 UNTIL SD(9)=0

9 服务和支持

我们的全球服务网络可为您提供优化的高效支持。如果您有任何疑问，可以咨询我们的专家，他们将为您提供建议和帮助。一天 24 小时、每周 7 天，请随时与我们联系。

德国服务总部 我们在德国洛尔的技术解决方案支援中心负责所有与电动驱动及控制方面的相关服务事宜。

请通过以下联系方式联系**服务热线**和**服务帮助台**：

电话：	+49 09352 40 50 60
传真：	+86 (0)21 2218 6766
电子邮件：	svf@boschrexroth.com.cn
因特网：	www.boschrexroth.com.cn/faservice

关于服务、维修（例如送交地址）和训练的详细信息，可在我们的网站上查找。

全球服务 如果您不在德国，请首先联系您当地的服务办公室。关于热线电话号码，请参考因特网上的销售办公室地址。

准备信息 为了能够更迅速有效地帮助您，请预先备妥下列信息：

- 故障及其发生情形的详细描述
- 请键入受损产品铭牌上的规格，尤其请键入编码和序号
- 您的联系资料（电话和传真号码，以及您的电子邮件地址）

索引

标记

<Axis>=S<Value>.....	174
\$<ChannelNumber>	34

0 ... 9

2 个部分字之间的分隔符	
轴和坐标指定.....	31
3D 刀具半径修正.....	159

A

AAC	204
AAS.....	204
ABS.....	74
ABSR0T2EUL.....	397
AC(...).....	146
ACN.....	293
ACOS.....	75
ACP.....	293
ACS.....	378
ActRadFact.....	182
ADJ.....	183
ADR_PARA).....	45
ADSCLR.....	187
ADSDIS.....	186
ADSENA.....	186
ADSLAB.....	187
ADVA.....	188
AdvanceAppend.....	188
AdvanceSubstitute.....	188
ADVS.....	188
AFC.....	210
AJK.....	213
AJS.....	213
ALN.....	197
AND.....	76
anwperm.dat.....	65
APOS.....	379, 380
ARA.....	190
Area.....	190
ARF.....	182
ASC.....	521
ASCII 字符集.....	560
ASIN.....	75
ASPCLR.....	194
ASPDIS.....	194
ASPENA.....	194
ASPRTP.....	195
ASPSET.....	196
ASPSTA.....	196
AssLogName.....	197
ASTOPA.....	89
ASTOPO.....	89
AsynchrShapeOrder, ASO.....	116
AsynchShapeOrder.....	117
ATAN.....	75

ATAN2.....	76
ATBWD.....	197
ATCAL.....	198
ATFWD.....	200
ATGET.....	200
ATPUT.....	202
ATR.....	362
ATrans.....	362
ATS.....	214
ATSW.....	215
AUXFUNC.....	203
AVE.....	216
AVS.....	216
AxAcc.....	203
AxAccSave.....	203
AXADR \$.....	373
AXADR\$.....	373
AXC.....	208
AxCompTab.....	205
AxCouple.....	208
AXINF.....	374, 375
AxisFeedCouple.....	210
AxisToSpindle.....	214
AxisToSpindleWait.....	215
AxJerk.....	213
AxJerkSave.....	213
AXO.....	393
AXP.....	376, 377
AXR.....	215
AxVel.....	216
AxVelSave.....	216

B

B-样条 (NURBS)	112
检查点的样条点权重.....	113
样条参数长度.....	112
坐标编程.....	112
B-样条逼近.....	113
参数长度.....	114
公差值.....	114
坐标编程.....	113
B-样条逼近参数.....	115
BAP.....	115
BC 类型.....	114
BCD.....	77
BCR.....	218
BCS2WCS.....	399
BcsCorr.....	218
BIN.....	77
BITIF.....	431
BlkNmb.....	81
BNB.....	81
BsaPar.....	115
BSTOPA.....	90
BSTOPO.....	90

索引

- 半径编程..... 102, 231
- 帮助台..... 571
- 保留命令字..... 71
 - 关键字..... 71
- 本地英制编程..... 140
- 本地子程序..... 40
- 逼近参考点..... 142
- 比较 STRING 表达式..... 521
- 编程
 - 用于定位模式的形状..... 118
 - 用于路径模式的形状..... 117
- 编程错误导致处理延迟..... 568
- 编程的轮廓移位..... 324
- 编程示例—删除前和/或后空格..... 530
 - 编程字符串变量（之前并进行尺寸标注）.. 530
 - 读取字符数组..... 531
 - 覆盖字符数组..... 532
 - 写字符数组..... 531
- 编程属性..... 30
- 编程样条参数长度..... 110, 111
- 编程主轴速度..... 176
- 编辑文件
 - 格式..... 511
 - 确定文件指针..... 514
 - 位置文件指针..... 513
 - 文件终止 ID..... 516
- 变量编程
 - 变量类型..... 66
 - 变量名..... 62
 - 变量组..... 63
 - 标准 数控 编程中的变量..... 69
 - 评价错误的 CPL 变量 ERRNO..... 69
- 变量间的
 - 赋值..... 73
- 变量类型..... 66
 - CHAR 变量..... 68
 - 浮点变量（实数）..... 66
 - 浮点变量（双精度）..... 67
 - 概述..... 68
 - 逻辑变量（布尔）..... 67
 - 域变量（数组）..... 67
 - 整数变量（INT）..... 66
- 变量名..... 62
- 变量组..... 63
 - 结构化变量..... 66
 - 局部变量..... 63
 - 可定义永久变量..... 63
 - 全局变量..... 63
 - 永久变量..... 63
- 标签编程
 - CPL 块中（DIN）..... 53
 - 标准数控块中（DIN）..... 52
- 标签编程和跳转指令..... 52
 - CPL 跳转（GOTO）跳转至任何程序块..... 56
 - 标准和 CPL 块中的标签..... 52
 - 无条件跳转至标准数控块 GoTo..... 55
 - 向后跳转至标准数控块 GoBack（GOB）..... 53
 - 向前跳转至标准数控块 GoAhead（GOA）..... 53
 - 有条件跳转到标准 NC 程序段 GoCond（GOC）..... 54
 - 子程序返回任何程序块 RetTo..... 56
- 标准编程..... 21
- 标准和 CPL 块中的标签..... 52
- 标准数控编程中的变量..... 69
- 标准数控块（DIN）的跳过块..... 58
- 标准数控块中的注释..... 35
- 补偿，体积..... 363
- 布尔变量..... 67
- 步冲..... 274
- C**
 - C1-和 C2-常量三次样条
 - 开始和结束边界条件..... 112
 - C1-和 C2-常数三次样条
 - 样条参数长度..... 112
 - 坐标编程..... 111
 - C1-和 C2-常系数三次样条..... 111
 - CALL 命令..... 43
 - CASE..... 58
 - CASE-LABEL... LABEL-OTHERWISE-ENDCASE..... 58
 - CFD..... 265
 - CGO..... 224
 - CHAR 变量..... 68
 - ChkNxtBlk..... 219
 - CHL..... 220
 - ChLength..... 220
 - CHLL..... 221
 - CHR\$..... 522
 - CHS..... 221
 - ChSection..... 220
 - CHSL..... 221
 - CLD..... 222
 - CLN..... 223
 - CLOCK..... 403
 - CLOSE..... 506, 507
 - CLRWARN..... 497
 - CNB..... 219
 - COF..... 393, 394
 - COFFS..... 160
 - Collision..... 223
 - ConfG0..... 224
 - ConstFeed..... 265
 - Coord..... 225
 - CornerLineDetection..... 221
 - COS..... 75
 - CoupleSplineTab..... 227
 - CPL
 - 命令坐标和轴位置..... 377
 - CPL 编程..... 22
 - CPL 变量 ERRNO..... 69
 - CPL 参数列表..... 46
 - CPL 程序部分..... 22
 - CPL 块中的标签..... 53
 - CPL 块中的注释..... 35
 - CPL 命令..... 72, 371

变量轴地址.....	376	浮点常量（双精度）.....	78
刀具修正.....	400	整数常量（INT）.....	78
读取测量值.....	383	字符串常量.....	78
控制数据和过程.....	403	常量切削速度.....	155
零点偏置.....	387	程序代码中的空行.....	34
文件和目录.....	500	程序结束.....	32, 167
与客户端通信.....	532	程序块 - 数控程序.....	25
轴和坐标地址.....	373	程序取消后通过数控程序重启继续执行程序... 306	
轴信息.....	374	程序设计	
主轴.....	386	在程序运行中跳转.....	36
主轴地址.....	386	程序设计，元素.....	34
字符串处理.....	519	CPL 块中的注释.....	35
坐标轴和坐标.....	371	标准数控块中的注释.....	35
CPL 跳转至任何程序块.....	56	程序代码中的空行.....	34
CPL 循环.....	569	块编号.....	34
CPL 语言元素.....	23	零件程序中的注释.....	34
CPL 元素.....	22	通道 ID.....	34
CPL 指令		为程序行提供注释.....	34
赋值.....	72	程序设计元素	
工件位置校正位置（斜面）.....	394	用户界面上的注解.....	35
关系运算.....	77	程序停止.....	167
逻辑运算.....	76	程序运行中的跳转.....	36
数学运算.....	73	决策 / 分支指令.....	37
数制间的转换.....	77	跳转指令.....	37
CPL 指令：CASE-LABEL... LABEL-OTHERWISE- ENDCASE.....	58	重复指令.....	37
CPL 指令：FOR-STEP-TO-NEXT.....	60	子程序调用.....	37
CPL 指令：IF-THEN-ELSE-ENDIF.....	59	程序字.....	26
CPL 指令：REPEAT-UNTIL.....	61	编程属性.....	30
CPL 指令：WHILE-DO-END.....	61	部分字.....	26
CPL 重复指令.....	60	部分字之间的分隔符.....	31
CPL 指令：REPEAT - UNTIL.....	61	程序字 - 效果	
CPL 指令：WHILE-DO-END.....	61	非模态.....	33
CPL 重复指令能		程序字一作用.....	32
CPL 指令：FOR-STEP-TO-NEXT.....	60	模态.....	32
CPL 命令.....	393	迟滞补偿.....	259
输入帮助.....	399	齿轮档选择	
CPL 中通过 CALL		手动.....	172
带 DIN 的 Link 过程.....	43	自动.....	172
CRD.....	225	冲程释放时间.....	297
CST.....	228	冲程释放时间（Inpos 窗口）.....	300
CurveParamInterpolation.....	229	将冲程释放时间设置为默认值.....	299
裁料.....	274	冲程释放时间（Inpos 窗口）.....	300
采用依照 DIN 66025 的语法的数控函数		冲程释放时间（插补结束时间）	
刀具修正.....	178	提前冲孔释放.....	297
进给和速度编程.....	173	冲程释放时间（插补终点）.....	297
采用依照 DIN66025 的语法的数控函数		后继冲孔释放.....	297
M 代码.....	167	设置冲程释放时间.....	298
参数传输到子程序		冲孔.....	302
数控转换为地址参数列表.....	47	冲压释放时间	
操作，逻辑.....	76	冲压释放时间（插补终点）.....	297
测量单位		初始化“加工中测量”.....	253
公制.....	141	处理速度，间接影响.....	566
提供的轴和坐标位置.....	372	“MCODS”版本参数.....	568
英制.....	140	WAIT 命令.....	567
常规 XML 表		等待永久 CPL 变量.....	568
访问.....	485	短语法.....	566
常量.....	78	模态函数.....	566
		文件访问.....	567

索引

注释.....	566
处理速度, 直接影响	
准确停止关闭 G62 的路径坡度 G8/G9.....	564
准确定位开启的 Inpos 窗口.....	564
处理延迟, 编程错误	
通过速度规格编程回转轴.....	568
处理延迟, 错误的编程	
CPL 循环.....	569
触发位事件.....	84
触发诊断记录.....	235
从 MP 应用轴设置.....	230
从动主轴	
带活动耦合的角度偏置.....	329
耦合距离.....	328
从动主轴的耦合距离.....	328
从攻丝缩回.....	355
从攻丝缩回.....	355
电源故障后的缩回.....	355
从轴组移除轴.....	307
错误和错误类别 "GETERR"	
5 个数组元素的结果分析.....	499
错误和错误类别 "GETERR"	
查询通道 2 中的上一个部件程序错误.....	499
错误和警告	
删除警告.....	497

D

D.....	178
D 修正.....	178
DATE.....	404
DAX.....	230
DBLOAD.....	486
DBMOVE.....	487
DBSAVE.....	488
DBSEA.....	489
DBSEAX.....	490, 491
DBTAB.....	493
DBTABX.....	493
DBTABXL.....	494
DC.....	293
DCB.....	235
DcBreak.....	235
DCC.....	235
DcCont.....	235
DCOFFS.....	160
DCR.....	234
DCS.....	229
DCT.....	401, 402
DcTSel.....	229
DefAxis.....	230
DefSpindle.....	230
DefTangTrans.....	230
DERDelay.....	232
DIA.....	231
DiaProg.....	231
DIM.....	519, 520
DIN 编程.....	21

DIN 和 CPL 编程选项.....	21
DIN 语言元素.....	23
DIRCR.....	503
DIRDEL.....	503
Directories	
Create.....	503
DIRINF.....	501, 502
DistCtrl.....	234
DN.....	109
DPC.....	394
DREC_START.....	235
DREC_STOP.....	235
DSP.....	230
DTT.....	230
达到轴位置前运动停止.....	88, 90, 91
打开文件	
OPENR.....	504
OPENW.....	504
代码字符.....	78
带 P 地址的子程序调用.....	38
带参数的	
宏编程.....	51
带参数的宏.....	51
带常用语言语法的 NC 函数	
在块预调整初始化程序中调整主轴.....	325
带常用语言语法的数控函数	
替换轴.....	215
带活动耦合的角度偏置.....	329
带系数编程的样条	
定向矢量编程.....	109
带一般语言句法的数控函数	
轴进给耦合.....	210
带有 b-样条的检查点的样条点权重.....	113
带常用语言语法的 NC 功能	
赋予逻辑轴名称.....	197
带常用语言语法的数控函数	
B 样条逼近.....	217
初始化“加工中测量”.....	253
带高级语言语法的 NC 功能	
反向回程序块扩展.....	313, 314
正向回程序块扩展.....	188
追溯正向程序块替换.....	188
带加速度自适应的速度编程	
增量.....	153
带系数编程的样条.....	108
编程样条参数长度.....	110
分母多项式编程.....	109
轴编程.....	108
坐标编程.....	108
带一般语言语法的数控功能	
抑制单步模式.....	347
单步抑制.....	347
刀具补偿	
D 修正.....	178
刀具接触点 - G45.....	131
刀具数据库	
动数据块.....	487
负载.....	486

扩展搜索.....	490	张量定向.....	282
使用上锁的扩展访问.....	494	定向矢量编程.....	109
搜索数据块.....	489	定义（激活）耦合主轴组.....	327
只读或写入权限.....	492	定义/移除主轴组.....	331
刀具数据库		定义极点.....	287
扩展权限.....	493	定义镜像点 / 旋转点.....	291
刀具修正.....	178, 400	定义异步子例程的返回工件加工位置点.....	310
ED 补偿 "ED".....	178	定义重新定位行为.....	310
刀具修正表和修正寄存器.....	401	读取测量值.....	383
激活刀具修正值.....	400	驱动器的测量值.....	385
刀具引导, 切向.....	350	在程序坐标中的测量值.....	383
刀具长度修正.....	132	轴坐标中的测量值.....	384
刀具中心点 - G46.....	131	读取文件	
刀具坐标系 TCS 中的线性移动进给.....	350	INP# 指令.....	512
刀具坐标系统 (TCS)		从文件读取记录.....	512
刀具修正.....	145	读取文件指针位置	
倒角编程.....	219	FILEPOS 和顺序文件.....	515
倒角段长度.....	219	FILEPOS 和随机文件.....	516
倒角长度.....	219	短语法.....	566
两段相邻圆周段间的倒角.....	220	对正主轴 / 定位主轴.....	170
两条相邻直线间的倒角.....	220	惰档选择.....	173
地址参数列表.....	44	E	
地址名称的明确性.....	44	EBC.....	114
使用多个参数.....	45	ED.....	179
使用相同变量名.....	45	ED 补偿.....	178
允许的参数地址.....	45	Edit file	
允许的参数值.....	45	Read.....	511
等待 PLC 数控 位接口处的状态		Write.....	509
WAITA.....	85	EndPosCouple.....	236
WAITO.....	85	EOF.....	516
等待 PLC 数控 位接口处的状态:		EPC.....	236
WAITA / WAITO.....	85	ERASE.....	507
等待角度偏置.....	330	ERRNO 变量.....	69
等待块结束时的主轴编程.....	340	Errors and warnings	
等待删除位事件.....	83	Query.....	497
等待同步模式.....	331	Runtime error.....	495
等待位事件.....	82	Runtime warning.....	496
等待永久 CPL 变量.....	568	F	
等待永久 CPL 变量的值.....	86	F.....	175
WPV.....	86	F-地址.....	174
WPVE.....	86	FA.....	176
定位.....	164	FAD.....	237
定位: 工件位置校正.....	217	FDG.....	240
计算链.....	218	FDGT.....	241
定位: 工件位置修正.....	162	FeedAd.....	237
绕 Z 坐标的基本旋转.....	218	FeedForward.....	238
定位: 斜面.....	161	FeedGrp.....	240
零件程序中的固有激活.....	162	FeedGrpType.....	241
倾斜平面与刀具垂直.....	163	FFW.....	238
使用定位表编程:.....	165	FFWA.....	239
直接在程序行中编程为参数.....	164	FILEACCESS.....	517
定位文件指针		FILECOPY.....	508
SEEK 和顺序文件.....	514	FILEDATE.....	517, 518
SEEK 和随机文件.....	514	FILENO.....	504
定向编程.....	276, 277	FILEPOS.....	514, 515
使用轴编程的线性方向运动.....	277		
使用坐标编程的直线方向运动.....	278		
向量定向.....	280		

索引

FILESIZE.....	518	局部相对尺寸编程.....	146
FlyMeas.....	242	具有路径比例速度自适应的绝对速度编程..	148
FME.....	242	具有时间比例速度自适应的绝对速度编程..	149
FOR-STEP-TO-NEXT.....	60	快速移动时的直线插补.....	98
FSM.....	246	每转进给编程.....	153
FsMove.....	245	平面切换.....	120
FSP.....	244	切向圆周出口.....	106
FsProbe.....	243	时间编程.....	146
FSR.....	246	无补偿卡盘的攻丝.....	139
FsReset.....	246	无平面.....	119
FST.....	245	相对尺寸编程.....	146
FsTorque.....	245	行进至探针.....	142
反馈.....	17	异步子程序：重新定位个别坐标.....	144
访问 XML 表.....	485	英制编程.....	140
访问操作数.....	432	直接速度编程.....	155
非模态 - 程序字.....	33	自由平面切换.....	122
局部.....	33	G 代码	
非模态-程序字		C1-和 C2-常数三次样条.....	111
非自持式.....	33	带系数编程的样条.....	108
非预期使用		停留时间.....	106
后果，无责任.....	19	G- 代码	
非预期使用.....	20	轮廓过渡用于铣削铣刀路径校正.....	130
非自持式.....	33	铣削铣刀路径修正.....	128
分隔进给块：路径段数量	276	准停定位开启 / 关闭	137
分隔进给块：路径段长度.....	259	G-代码	
分母多项式编程.....	109	AsynchrShapeOrder.....	116
分配字符串表达式.....	523	B-样条（NURBS）（样条类型 3）.....	112
服务热线.....	571	B-样条逼近参数.....	115
浮点变量（实数）.....	66	高级平面切换.....	121
浮点变量（双精度）.....	67	公制编程.....	141
浮点常量（双精度）.....	78	急进限制速度控制.....	116
附加条件.....	26	加速适应条件下增量速度编程	150
程序字.....	26	接近参考点坐标.....	141
含附加条件的路径信息.....	26	进给编程.....	147
赋予逻辑轴名称.....	197	开始和结束边界条件.....	114
赋值.....	72	可编程零点偏置.....	133
NUL.....	73	路径坡度关闭.....	115
布尔变量.....	72	路径斜坡开启.....	115
赋值.....	72	轮廓移动编程.....	123
实数变量.....	73	螺纹切削.....	125
G		样条编程.....	108
G 编码		G0.....	99
圆弧 / 螺旋 / 螺旋 N 插补.....	101	G0 插补配置.....	224
G 代码.....	98	G0 上的 Inpos 窗口.....	565
3D 刀具半径补偿.....	159	G1.....	100
补偿切换.....	144	G2，顺时针圆形路径.....	101
常量切削速度.....	155	G3，逆时针旋转的圆形路径.....	101
带加速度调节的增量编程.....	153	G4.....	106
刀具长度修正.....	132	G5.....	107
定位：倾斜平面.....	161	G6.....	108
回转轴每转进给编程.....	155	G8.....	116
加速适应条件下绝对速度编程.....	151	G8 (SHAPE.....)	117
进给到固定的机床轴位置.....	143	G9.....	116
进给校正.....	131	G9 (ASHAPE.....)	118
进给至参考点.....	141	G9 (SHAPE.....)	118
局部绝对尺寸编程.....	146	G9 (X..., Y..., ...)	116
		G12.....	106
		G16.....	120

索引

G17.....	120	G95 (F 值 / S 值)	154
G17 (...).	121	G95 (FRA=<Axis>)..	155
G18.....	120	G96.....	155
G18 (...).	121	G97.....	155
G19.....	120	G140.....	160
G19 (...).	121	G141.....	160
G20.....	122	G142.....	160
G33.....	125	G151.....	164
G40.....	129	G152.....	165
G41.....	129	G153.....	164
G42.....	129	G153 - G159.....	165
G43.....	131	GAX.....	247
G43 - 弧线.....	130	GCT.....	247
G44.....	131	GCT(0).....	247
G44 - 交叉.....	130	GCT(1).....	247
G45.....	132	GeoComp.....	247
G46.....	132	GetAxis.....	247
G47.....	132	GETERR.....	498
G48.....	133	GetSpindle.....	248
G52.....	134	GMSG.....	36
G53 至 G59.....	135	GOA.....	53
G53.1.....	136	GoAhead.....	53
G53.5.....	136	GOB.....	54
G54.1.....	136	GoBack.....	54
G54.1 至 G59.1.....	136	GOC.....	54
G54.5.....	136	GoCond.....	54
G55.1.....	136	GOTO.....	55, 56
G55.5.....	136	GOTO 命令.....	566
G56.1.....	136	GSP.....	248
G56.5.....	136	G 代码	
G57.1.....	136	Hermite 样条.....	110
G57.5.....	136	进给中的线性插补.....	100
G58.1.....	136	切向圆周入口.....	107
G58.5.....	136	高级平面切换.....	121
G59.1.....	136	轴名称指定.....	121
G59.5	136	高级语言句法.....	181
G61.....	138, 564	高速输出	
G62.....	138, 564	与位置相关.....	291
G63.....	139	跟踪错误.....	137
G63.2.....	139	更改档位范围.....	335
G70.....	140	更改字符串	
G71.....	141	读取字符串中的字符.....	529
G74.....	141	分配字符串表达式.....	523
G74 (HOME)	142	链接 STRING 表达式.....	525
G75.....	142	删除空格.....	528
G76.....	143	转换大写和小写字母.....	529
G77.....	144	更改最大路径加速度.....	286
G78.....	145	工件位置校正和位置 (斜面)	
G79.....	145	转换点.....	398, 399
G90.....	146	工件位置校正位置 (斜面)	394
G91.....	146	工件位置修正.....	394
G93.....	147	工件位置修正和定位 (斜面)	
G94.....	148	计算欧拉角度.....	396, 397, 398
G94 (F-value/S-value).....	150	工件位置修正和位置 (斜面)	
G94 (LINS).....	148	工件位置修正.....	394
G94 (LINT).....	149	工件坐标	
G94 (TF value/TS value).....	152	点动.....	256
G95.....	153	工件坐标系 (WCS)	

索引

刀具修正.....	144
工件坐标中的点动模式.....	256
工作平面尺寸.....	233
工作平面的尺寸.....	232, 233
工作位置修正和定位(斜面)	
定位表.....	395
激活定位值.....	395
公制编程.....	141
攻丝.....	139
固定挡块处的测量.....	243
关闭特定通道的主轴运动管理.....	335
关闭主轴的 C 轴模式.....	214
关闭主轴的 C 轴模式并等待.....	215
关键代码.....	563
关键字.....	71
关系运算.....	77
关于本文档.....	11
文档有效性.....	11
信息表示方法.....	16

H

Hermite 样条

编程样条参数长度.....	111
用户定义的速度参数文件.....	111
Hermite 样条.....	110
轴编程.....	110
坐标编程.....	110
HSB.....	249
HSB(., HSSTOP=.).....	251
HsBlkSwitch.....	249
HsBlkSwitch(., HSSTOP=.).....	251
HWOC.....	252
HWOCDIS.....	252
合成形状阶数.....	119
宏.....	50
宏编程.....	50
后链接.....	24
后向转换.....	197
回程.....	311
回程反向程序扩展.....	313
回程反向程序替换.....	314
回程切换锁.....	312
回程正向程序扩展.....	188
回程正向程序替换.....	188
回转轴每转进给编程.....	155
获取通道主轴.....	248
获取轴.....	247
获取轴, 在必要时等待.....	367

I

I 数控 H.....	140
IC(...).....	146
IF-THEN-ELSE-ENDIF.....	59
IME.....	253
InitMeas.....	253
INP#.....	511
INSDEP.....	160

INSTR.....	520
INT.....	74
InvTime.....	253
IPS1.....	254
IPS2.....	254
IPS3.....	254
ITIM.....	253

J

JAL.....	255
JerkAxList.....	255
JerkControl.....	256
JKC.....	256
JogWCSSelect.....	256
基本 CPL 元素	
常量.....	78
代码字符.....	78
激光功率控制器	
与路径速度相关的.....	261
激活功能	
NCF.....	433
激活急进限制.....	256
激活零点偏置表.....	369
激活主轴的 C 轴模式并等待.....	339
极坐标编程.....	288
半径值.....	288
定义极点.....	287
极角 1.....	288
极角 2.....	289
极坐标编程.....	288
绝对尺寸.....	289
增量尺寸.....	290
急进.....	119
急进限制速度控制.....	116
合成形状阶数.....	119
形状阶数和急进之间的关系.....	119
用于定位模式的形状.....	118
用于路径模式的形状.....	117
记录反向向量.....	305
加工速度, 直接影响.....	564
Sercos 通讯.....	565
连续块转换.....	565
数控段.....	565
在 G0 上的 Inpos 窗口.....	565
自由曲面加工.....	565
加工中测量.....	242
初始化.....	253
加加速度限制的速度限制	
用于异步轴的形状.....	117
加速适应条件下速度编程	
绝对.....	151
增量.....	150
简单的系统数据	
SD.....	418
简单数学功能.....	73
简单系统数据	
SD.....	406

建议.....	17
将 ID 列表中的扩展 ID 编号写入驱动器 WriteId3, WID3.....	369
将参数传输到子例程.....	44
将参数列表中的值写入驱动器 WriteId3, WID3.....	369
将参数写入驱动器	
同步轴 (WriteId3, WID3).....	368
主轴和异步轴 (WriteId3, WID3).....	368
将冲程释放时间设置为默认值.....	299
将通道主轴复位至 MP.....	230
将主轴切换到轴.....	568
将主轴作为轴应用于通道.....	568
将主轴作为轴应用于通道处理延迟, 错误编程 将主轴作为轴应用于通道.....	568
将字符串表达式分配给字符串.....	523
接近参考点坐标.....	141
接近机床固定的轴位置.....	143
接近轴位置, 机床固定.....	143
结构化变量.....	66
结构化类型的系统数据	
访问.....	422
加载结构化系统数据.....	424
可变信息.....	423
结束位置耦合.....	235
結構化類型的系統數據	
保存結構化系統數據.....	425
可變信息.....	423
解除/转移保留主轴.....	326
解释次数.....	23
进给.....	100
进给 100%.....	285
进给编程 (按分钟).....	147
进给和速度编程.....	173
F 地址.....	174
F2 地址 F2.....	175
异步轴的速度.....	176
进给计算: 进给组.....	240
进给计算: 进给组类型.....	241
进给计算: 隐藏轴.....	237
进给校正.....	131
刀具接触点 G45.....	131
刀具中心点 G46.....	131
进给至探头.....	142
进给中的线性插补.....	100
基本功能.....	100
可选参数.....	100
禁用径向和切向加速度的总和产生的速度限制.....	257
禁用运动数控块分割.....	257
精度编程.....	295
静态刀具定向的参数设置.....	345
镜像.....	267
局部.....	33
局部变量.....	63
局部绝对尺寸编程.....	146
局部相对尺寸编程.....	146
具有常用语言语法的数控函数	

对驱动错误的延迟响应.....	232
进给计算: 进给组类型.....	241
具有高电位语言语法的数控函数	
回程切换锁 LockRetrace, LOCKRETRACE ...	312
具有通用语言语法的 NC 功能	
根据块准备进行程序处理.....	296
具有通用语言语法的 NC 功能	
异步子例程.....	193, 310
决策和分支指令.....	57
决定和分支指令	
CPL 指令: IF-THEN-ELSE-ENDIF.....	59
跳过块.....	58
决定和分支命令	
CPL 命令: CASE-LABEL... LABEL- OTHERWISE-ENDCASE.....	58
绝对尺寸编程.....	146
K	
KinLimAcc.....	257
KinSplitDisable.....	257
KV 编程.....	257
KVP.....	258
KvProg.....	257, 258
开始和结束边界条件.....	114
可编程零点偏置.....	133
可编程路径分割.....	344
可变编程.....	62
可定义永久变量.....	63
可选参数	
进给中的线性插补.....	100
快速移动的线性插补.....	99
圆 / 螺旋 / 螺旋 N 插补.....	104
可用函数, NS 耦合	
MCODS, 函数概述.....	436
客户反馈.....	17
客户特定的速度参数文件.....	229
控件数据	
常规 XML 表.....	485
刀具数据库.....	486
控制版本	
VERSINF\$.....	403
控制数据.....	403
PLC 接口.....	431
Sercos 接口.....	426
错误和警告.....	495
机床参数.....	405
激活辅助功能.....	433
激活功能.....	433
简单系统数据.....	406
时间记录.....	403
控制数据和过程	
数控接口 (NCS).....	434
控制数据據	
控制版本.....	403
块编号.....	34
CPL 块.....	34
标准数控块.....	34

索引

块的执行

DIN - CPL 语言部分..... 23

块执行

CPL 元素..... 23

块执行时的同步功能..... 82

达到基本工件位置前运动停止..... 90, 91

达到轴位置前运动停止..... 88

等待 PLC 数控 位接口处的状态..... 85

等待永久 CPL 变量的值..... 86

取消停止条件..... 92

通过程序结束 (M30) 取消停止条件..... 92

写入永久 CPL 变量..... 87

块准备的同步功能..... 79

BlkNmb (BNB)..... 81

WAIT (, <idle time>)..... 79

WAIT (BITIF(...))..... 80

WAIT (无参数)..... 79

快速移动的线性插补

可选参数..... 99

快速移动中的直线插补 “G0”..... 98

L

LagCompensation..... 259

LCP..... 259

LCS 参考系..... 156

LEN..... 260, 520

LFC..... 262

LFConf..... 262

LFP..... 261

LinDownFeed..... 265

LinearToSpline..... 263

LinModZp..... 263

LinUpFeed..... 264

LJUST..... 511

LJUSTi..... 511

LMZ..... 263

LND..... 265

LNU..... 264

LTS..... 263

类型转换

ASCII 码..... 521

ASCII 字符..... 522

数字转换..... 523

字符转换..... 522

连接 STRING 表达式

使用 STRING 表达式连接文本..... 525

连续的块转换..... 565

链接

链接表..... 23

链接 STRING 表达式..... 525

还可以在 CPL 命令中链接..... 525

写访问的 MID\$ 命令..... 527

只读访问的 MID\$ 命令..... 527

链接, 数控程序

后台的链接..... 24

链接, 子程序..... 24

与 DIN / CPL ID 链接..... 25

链接, 子程序..... 24

后链接..... 24

链接数控程序..... 23

两个部分字之间的分隔符..... 31

零点偏置..... 135, 387

读取和写入访问 Z0 表..... 390

激活 Z0 值..... 392

在 Z0 表中插入轴..... 388, 389

零件程序中的注释..... 34

路径坡度关闭..... 115

路径坡度开启..... 115

轮廓移动编程..... 123

逻辑变量 (布尔)..... 67

逻辑操作..... 76

逻辑运算

AND 功能 AND..... 76

EXCLUSIVE OR 功能 XOR..... 76

NOT 功能 NOT..... 76

OR 功能 OR..... 76

螺距..... 139

螺纹切削..... 125

多头螺纹..... 127

连续螺纹..... 127

面螺纹..... 126

锥形端面螺纹..... 127

纵向螺纹..... 126

螺线 N 插补..... 102

螺旋插补..... 102

M

M 代码

Omega 地址 (进给)..... 176

编程主轴速度..... 176

惰档选择..... 173

模态速度设置..... 173

手动换档选择..... 172

速度设置..... 173

退出程序..... 167

有条件程序中断..... 167

中断程序..... 167

主轴逆时针旋转..... 169

主轴逆时针旋转和冷却液打开..... 169

主轴顺时针旋转..... 168

主轴顺时针旋转和冷却液打开..... 168

主轴停止..... 170

自动换档步进选择..... 172

M-代码..... 167

M0..... 167

M00..... 167

M1..... 167

M01..... 167

M2..... 168

M02..... 168

M3..... 169

M4..... 169

M5..... 170

M13..... 169

M14.....	169	刀具修正表的名称.....	442
M19.....	171	点动速度.....	445
M30.....	168	电机类型.....	454
M40.....	172	电前丝杠补偿.....	471
M41... 44.....	172	定位的总数.....	467
M48.....	173	返回轮廓.....	450
M103.....	169	峰值和轴加速度的最大值.....	473
M104.....	169	峰值和轴速度的最大值.....	472
M105.....	170	辅助功能的组号.....	463
M113.....	169	工件位置修正.....	467
M114.....	170	工件坐标.....	458
M119.....	171	机床坐标名称.....	465
M140.....	172	机床坐标数.....	465
M141... 144.....	172	基本坐标, 命令值.....	458
M148.....	173	基本坐标, 实际值.....	459
M203.....	169	激活 Z0 值.....	443
M204.....	169	激活半径修正.....	442
M205.....	170	激活程序坐标偏置.....	468
M213.....	169	激活程序坐标偏置总和.....	468
M214.....	170	激活刀具修正.....	453
M219.....	171	激活的路径进给.....	472
M240.....	172	激活定位表格的路径和名称.....	467
M241... 244.....	172	激活定位数据.....	467
M248.....	173	激活辅助函数组的语法.....	463
machdef. dat.....	66	激活累加程序坐标偏置.....	468
MACHINE 参数		激活零点偏置数据.....	464
MCA.....	405	激活偏置坐标.....	469
MainSp.....	266	激活缩放系数.....	469
MCA.....	405, 406	激活坐标镜像.....	470
MCADS		加工坐标.....	459
接近的参考点.....	445	进给电位器.....	441
MCASE\$.....	529	进给禁止.....	452
MCOD		可选停止.....	458
编程的主轴速度.....	444	控制装置类型.....	454
当前补偿值.....	471	块的预运行再入口标记的状态.....	470
MCODS.....	434, 435, 474	块预运行调整状态.....	470
A CS 中轴的剩余距离.....	461	来自循环轴报文的标识号.....	456
CNC 的实际轴位置:.....	473	零点偏置的总和.....	464
InPos.....	441	路径进给.....	440
IPO 模式.....	447	默认轴分配: 通道.....	454
Lag.....	441	频道等待状态.....	461
SAV 模式.....	447	驱动器的实际轴位置:.....	448
WCS 中的在线修正.....	462	实际主轴扭矩.....	466
WCS 中的坐标剩余距离.....	461	实际主轴位置.....	466
Z0 表的名称.....	442	特定于客户的数据.....	449
半径修正编号.....	442	跳过块.....	458
包括输入帮助结束位置.....	443	通道复位.....	470
编程的结束位置.....	444	通道模式.....	446
编程的路径进给.....	444, 472	通道数.....	450
编程的坐标值.....	448	通道轴名称的默认设置.....	455
编程的坐标终点.....	459	通道主轴: 组分配.....	457
编程示例.....	473	行程命令.....	451
测试模式.....	445	需要确认激活的辅助功能.....	449
当前交叉补偿.....	471	旋转角度.....	469
当前温度补偿.....	471	已编程的轴速度.....	469
当前轴速度.....	468	已启用块传输.....	449
档切换, 自动 / 手动.....	457	英制/公制.....	443
刀尖的基准坐标 (BCS-Tcp).....	463	用于坐标的 INPOS 状态.....	460

索引

用于坐标的测量单位.....	460	MIR.....	267, 268
在 ACS1 中的轴坐标.....	459	mirror.....	267
在 ACS0 中的轴坐标.....	466	MMC.....	532
在 WCS 中的实际轴值.....	449	MOB.....	269
在 WCS 中的在线修正.....	462	MOC.....	270
暂停时间激活.....	449	MOD.....	272
直径编程.....	462	Modulo.....	272
制造商版本.....	454	MPEDGE.....	273
轴的测量单位.....	453	MSG.....	36
轴的测量单位（默认设置）.....	455	MSP.....	266
轴电位计.....	452	M 代码	
轴分配：通道.....	450	定向 / 定位主轴.....	170
轴名称.....	447	每转进给编程.....	153
轴命令位置.....	440	命令 - 数控程序.....	25
轴命令位置（整数）.....	440	模态 - 程序字	
轴模式.....	451	自持式.....	32
轴数.....	450	模态程序字.....	32
轴坐标终点.....	464	模态功能组 - 概述.....	95
主程序：名称和路径.....	446	模态函数.....	566
主轴：档切换.....	457	模态速度指定.....	173
主轴档位级别.....	456	模态轴（线性循环轴）回零.....	263
主轴电位计.....	441	模态轴回零.....	263
主轴列表速度.....	448	模态子程序.....	47
主轴名称.....	464	默认轴配置.....	230
主轴数.....	465	目标组.....	11
主轴系统编号.....	465	目录	
主轴性能.....	466	目录信息.....	501
主轴运动功能.....	456	删除.....	503
主轴轴编号.....	465		
主轴转速.....	441	N	
注解(消息).....	445	NC 功能	
自动程序重新选择.....	458	调整模态 NC 功能.....	183
最大轴速度.....	471	NC 功能，具有高级语言语法	
最大主轴速度.....	444	可编程的距离相关信号.....	185
最小主轴速度.....	444	NC 功能，具有通用语言语法	
坐标参考.....	460	增量式行程无限轴模数计算.....	272
坐标名称.....	460	NC 功能具有常用语言语法	
坐标数.....	461	加工中测量.....	242
坐标终点.....	459	NC 函数具有常用语言语法	
MCODS 的版本参数.....	568	加载轴误差校正表.....	205
MCODS 调用.....	568	NCF.....	433, 434
MCODSC		NCS 耦合	
当前轴速度.....	468	编程示例- 立即请求通道 2 的 SAV 状态和插	
MCOPS.....	474	补器.....	483
复位.....	474	编程示例- 等待直至通道 2 的 SAV 状态切换为	
改变返回轮廓策略.....	480	未激活.....	483
改变返回轮廓录制.....	481	编程示例- 通道/等待状态.....	484
启动程序.....	479	编程示例- 在 MSG 窗口中输出轴名称.....	484
取消剩余距离.....	474	编程示例- 轴通道分配.....	484
取消选择程序.....	479	错误返回值.....	482
搜索块.....	475	NC 函数具有常用语言语法	
通道重启.....	481	逆时间编程.....	253
选择程序.....	476	NC 接口（NCS）	
选择手动数据输入.....	476	编程示例.....	483
指定操作模式.....	479	NIB.....	274
MCS.....	380	NJUST.....	511
MCS 参考系.....	156	NOT.....	76
MID\$.....	529		

索引

NUL..... 73
NUM..... 276
NURBS..... 112
扭矩减小..... 306

O

O()..... 281
OFFSTOPA..... 92
OFFSTOPO..... 92
Omega..... 176
Omega 地址 (进给)..... 176
OPENR..... 505
OPENW..... 505
OR..... 76
OVD..... 285
OVE..... 285
OvrDis..... 285
OvrEna..... 285

P

P..... 38
PAC..... 286
PathAcc..... 286
PCS..... 381
PCS 参考系..... 156
PCSPROBE..... 383, 384
PDIM..... 47
PHS..... 292
PL..... 110, 111, 112, 113
PLC..... 432
PLC 接口..... 431
 位接口..... 431
PLC 接口
 操作访问..... 432
PLS..... 291
PMD..... 293
PMS..... 287
PMT..... 395
PmTSel..... 287
PMV..... 395
PolarPol..... 287
PoleSet..... 291
POP..... 287
PosDepHSOut..... 292
PosMode..... 293
PPOS..... 384, 385
PrecProg..... 295
PREP..... 296
PRN#..... 509
PROBE..... 385, 386
PRP..... 296
PT2EUL..... 397
PtBlkEnd..... 298
PTD..... 299
PtDefault..... 299
PTE..... 298
PTI..... 300

PtInpos..... 300
PTP..... 301
PTP 进给运动..... 300
PUN..... 303
Punch..... 303
PW..... 113
配置动力学..... 360
配置缩回数据..... 360
碰撞监控..... 222
批評..... 17
平面切换..... 120
 指定与轴无关..... 121
平面切换, 高级..... 121
平面选择, 自由..... 122

Q

启用 / 转换保留的主轴..... 334
启用 D 修正表..... 229
启用定位表..... 287
启用缩回..... 361
前馈..... 238, 239
前向转换..... 199
切换主轴操作模式..... 361
切向刀具定向..... 353
切向刀具控制
 刀具控制的位置..... 350
切向刀具引导..... 350
 设置角度..... 351
切向圆周出口..... 106
切向圆周入口..... 107
区域监控..... 189
 不工作区域..... 189
 定义、启用、取消和释放一个监控区域..... 190
 工作区..... 189
 激活, 停用并释放一个或所有监控区域即
 刻..... 190
曲线参数插补..... 228
驱动器错误的延迟响应..... 232
取消停止条件..... 92
 OFFSTOPA..... 92
 OFFSTOPO..... 92
去除小数点后的位数..... 74
全局变量..... 63
确定 NC 重启调整程序的主轴语法..... 326
确定文件大小
 FILESIZE 和顺序文件..... 518
 FILESIZE 和随机文件..... 519
确定系统轴以规定过冲限制启用情况下的最佳形状
过滤器..... 254

R

RAC..... 304
RAD..... 231
RadialAcc..... 304
RadProg..... 231
RANDOM..... 74
RAX..... 308

索引

RCB.....	306
RDT.....	306
RecordRevVec.....	305
RedTorque.....	306
ReentContBlk.....	306
REM.....	35
RemAxis.....	308
RemLogName.....	308
RemSpindle.....	309
REPEAT.....	309
REPEAT-UNTIL.....	61
REPOSDEF.....	310
REPOSTP.....	311
RetTo.....	56
REV.....	84
REVA.....	313
ReverseAppend.....	313
ReverseSubstitute.....	314
REVS.....	314
REWRITE.....	506
RLN.....	308
RND.....	316
RNDL.....	317
RNE.....	316
ROT(...)	315
ROT2EUL.....	396
Rotate(...)	314
ROTAX()	282
ROUND.....	75
RoundEps.....	316
Rounding.....	316
RRV.....	305
RSP.....	309
热线.....	571
锐化内角	226

S

S.....	176
S 属性.....	174
S2D.....	265
S2U.....	264
SACSYNC.....	317
SBC.....	114
Scale(...)	320
SCC.....	321
SCCT.....	322
SCL.....	399
SCL(...)	320
SCO.....	342
SCOL.....	342
SCS3.....	426
SCS3L.....	428
SD.....	406, 418
Block 预运行: 目标位置.....	419
Init 程序, 触发事件.....	412
倍率.....	406
编程实例.....	418

程序坐标偏置 (Trans + ATrans) ...	413, 414
处理方法.....	409
当前通道.....	407
当前通道中的轴数.....	410
当前通道中通道轴的系统轴号.....	411
返回轮廓的点.....	414
工作区和死区.....	414
国家语言.....	409
接触探头.....	407
进给组.....	417
精确编程.....	417
累加程序坐标偏置 (Trans + ATrans) ...	416
平面配置 - 分配的通道轴.....	408
切向刀具导向.....	414
速度.....	406
通道中的轴数.....	410
通道轴的系统轴号.....	411
通道轴耦合.....	417
通道主轴的速度.....	415
通道主轴的系统轴编号.....	416
通道主轴的系统主轴编号.....	416
通道主轴的主轴编号.....	415
无运动测试.....	409
系统轴的通道.....	412
系统轴的通道轴编号.....	411
系统轴的系统主轴编号.....	416
系统主轴的系统轴号.....	416
样条数据.....	413
运动功能.....	415
长度 修正 - 已分配的坐标.....	408
主轴倍率.....	415
主轴旋转的方向.....	409
主轴组的通道主轴编号.....	415
坐标含义的坐标的通道索引.....	410

SDA

精确编程.....	422
SDLOAD.....	424

SDR

北极点周围的 Epsilon 环境.....	421
程序坐标偏置 (Trans + ATrans)	420
程序坐标偏置 (trans)	420
覆盖.....	419
累加程序坐标偏置 (Trans + ATrans) ...	421
切向刀具导向.....	420
速度.....	419
通道轴耦合.....	422
样条数据.....	419
在 Asup 之后返回轮廓.....	420
主轴倍率.....	421

SDSAVE.....	425
SEEK.....	513
SeICrdCouple.....	321
SeICrdCoupleTab.....	322

Sercos 接口

读取 Sercos III 参数.....	426
读取 Sercos 参数列表.....	428
写入 Sercos 参数.....	429
Sercos 通讯.....	565

索引

SETERR.....	495	SRD	
SetPos.....	323	正向通道主轴的位置.....	421
SETWARN.....	496	SRS	
SEV.....	84	通道主轴速度.....	421
Shift.....	324	SSD.....	93
SHT.....	324	SSDE.....	93
SIN.....	75	SSDQ.....	332
Sin2DownFeed.....	265	SSDQInit.....	333
Sin2UpFeed.....	264	SSpAdm.....	334
SinDownFeed.....	265	SSpAdmOff.....	335
SinUpFeed.....	264	SSPG.....	176
SMax.....	325	SSpGear.....	335
SMD.....	347	SSPM.....	337
SME.....	348	SSpMax.....	336
SMin.....	324	SSpMin	
SMN.....	325	SSPMIN.....	336
SMX.....	325	SSpMode.....	336
SND.....	265	SSpMove.....	337
SNU.....	264	SSP00.....	337
SPA.....	326	SSpOri.....	337
SpAdjStr.....	326	SSPS.....	338
SpAdmin.....	326	SSpSpeed.....	338
SPADR\$.....	386, 387	SSpToolLim.....	338
SPC_WAIT.....	331	STA.....	339
SPCC.....	327	StatToolOri.....	346
SPCD.....	328	STAW.....	339
SPCE.....	329	StepModeDisable.....	347
SpCouple_Wait.....	331	StepModeEnable.....	348
SpCoupleConfig.....	327	ST0.....	346
SpCoupleDist.....	328	SToolLim.....	349
SpCoupleErrWin.....	329	STR\$.....	522
SpCouplePosOffs.....	329	SWM.....	341
SpCouplePosOffs_Wait.....	330	删除前和/或后空格	
SpCoupleSyncWin.....	330	编程示例.....	530
SPCP.....	329	删除位事件.....	84
SPCP_WAIT.....	330	设置 B 样条逼近的参数.....	217
SPCS.....	330	设置 B 样条逼近的参数.....	217
SPD.....	343	设置程序位置.....	323
SPG.....	331	设置校正.....	217
SPGALL.....	332	生成随机正整数.....	74
SpindleToAxis.....	339	时间编程.....	146
SpindleToAxisWait.....	339	时间记录.....	403
SpindleWait.....	340	系统时间.....	403
SpindleWaitMode.....	341	时间录制	
SplineCornering.....	342	时间.....	405
SplineCorneringLocal.....	342	时钟.....	403
SplineDef.....	343	实数变量.....	66
Split.....	344	使用 G 和 M 代码的自定义子程序调用.....	42
SPM.....	345	使用常用语言语法的 NC 功能	
SpMode.....	345	测量轮廓.....	270
SPOS.....	382	校准轴运动.....	197
SPS.....	323	使用常用语言语法的 NC 函数	
SPT.....	344	块重复.....	309
SPV.....	88	使用常用语言语法的数控函数	
SPV 命令.....	568	C0 编程.....	257
SPVE.....	88	G0 插补配置.....	224
SPWAIT.....	340	PTP 进给运动.....	300
SQRT.....	75	TCS 中的进给.....	350

索引

半径编程.....	231	确定系统轴以规定过冲限制启用情况下的最佳形	
保存最大轴速度.....	216	状过滤器.....	254
裁料.....	274	设置程序位置.....	323
程序取消后通过数控程序重启继续执行程序.....	306	使用缓存数控块输入进行下一块监控.....	219
程序坐标偏置.....	361	使用样条的转角修圆.....	341
迟滞补偿.....	259	数字化距离控制.....	233
冲程释放时间.....	297	速度变化图（斜坡函数）.....	264
冲孔.....	302	速度控制的诊断.....	364
从 MP 应用轴设置.....	230	速度限制.....	324
从攻丝缩回.....	355	特殊螺纹切削函数.....	359
从轴组移除轴.....	307	停用运动数控块分割.....	257
带活动耦合的角度偏置.....	329	通过高速信号实现带挡块的块过渡.....	251
倒角编程.....	219	通过高速信号实现加工中块过渡.....	249
等待角度偏置.....	330	同步模式窗口.....	329, 330
等待块结束时的主轴编程.....	340	同步系统轴耦合.....	317
等待同步模式.....	331	退出诊断记录.....	235
调整轴配置 TargetAxisConf, TAX.....	357	无补偿夹具的攻丝的主轴选择.....	356
叠加程序坐标偏置.....	361	无减速的块过渡.....	230
定向编程.....	276	系统主轴编程.....	334
定义（激活）耦合主轴组.....	327	线性样条曲线切换.....	263
定义/移除主轴组.....	331	写入 Sercos 参数.....	367
定义镜像点 / 旋转点.....	291	选定的附加坐标耦合.....	321
改变最大轴速度.....	216	选择一个等待策略.....	341
更改最大路径加速度.....	286	选择轴转换.....	225
工件坐标在线修正.....	252	选择准停定位窗口.....	254
工件坐标中的点动模式.....	256	循环轴的定位类型.....	293
固定挡块处的测量.....	243	样条类型的定义.....	232, 342
关闭主轴的 C 轴模式.....	214	样条耦合表.....	227
关闭主轴的 C 轴模式并等待.....	215	移除（取消）耦合主轴组.....	327
获取通道主轴.....	248	移除逻辑轴名称.....	308
获取轴.....	247	移除通道主轴.....	309
获取轴，在必要时等待.....	367	与路径速度相关的激光功率控制器.....	261
激活急进限制.....	256	与位置相关的高速输出.....	291
激活零点偏置表.....	369	运动至固定挡块.....	245
极坐标编程.....	287	在程序坐标中定义 TCS.....	357
记录反向向量.....	305	在系统数据队列中写入.....	332
将通道主轴复位至 MP.....	230	直径编程.....	231
结束位置耦合.....	235	轴耦合.....	207
进给 100%.....	285	主轴切换.....	266
精度编程.....	295	转角的修圆.....	316
静态刀具定向的参数设置.....	345	自动探测转角和直线.....	221
开启主轴的 C 轴模式.....	339	最大径向加速度.....	304
开启主轴的 C 轴模式并等待.....	339	使用高级语言语法的 NC 功能	
可编程路径分割.....	344	编程的轮廓移位.....	324
模态轴（线性循环轴）回零.....	263	从动主轴的耦合距离.....	328
扭矩减小.....	306	定位：工件位置修正.....	217
碰撞监控.....	222	分隔进给块：路径段数量.....	276
启动/转换保留的主轴.....	326	分隔进给块：路径段长度.....	259
启动诊断记录.....	235	更改最大轴过冲.....	213
启用 D 修正表.....	229	更改最大轴加速度.....	203
启用定位表.....	287	缓存最大轴过冲.....	213
前馈.....	238	缓存最大轴加速度.....	203
切向刀具控制.....	350	回程、RETR.....	311
区域监控.....	189	进给计算：进给组.....	240
曲线参数插补.....	228	进给计算：隐藏轴.....	237
确定 NC 重启调整程序的主轴语法.....	325	禁用径向和切向加速度的总和产生的速度限制 ..	257
		客户特定的速度参数文件.....	229
		前馈异步轴.....	239

切向刀具定向.....	353	数控接口 (NCS)	
锐化内部拐角.....	226	过程服务.....	474
输入工具: 镜像.....	267	数据服务.....	434
输入工具: 缩放.....	319	数控块同步.....	78
输入工具: 旋转.....	314	活动时间点.....	79
特定于刀具的主轴限制.....	338, 349	块准备.....	78
体积补偿.....	363	块准备的同步功能.....	79, 82
虚拟驱动器.....	364	执行时写入系统数据.....	92
抑制冲突监控.....	322	数学运算.....	73
在块上测量.....	269	ABS.....	74
执行所有组的激活辅助功能.....	203	ACOS.....	75
直线度和角度误差补偿.....	247	ASIN.....	75
主轴: 在位置/速度接口之间切换.....	345	ATAN.....	75
使用缓存数控块输入进行下一块监控.....	219	ATAN2.....	76
使用通用语言语法的 NC 功能		COS.....	75
更改测头边缘配置.....	273	EXP.....	74
使用样条的转角修圆		INT.....	74
局部.....	342	RANDOM.....	74
模态.....	341	ROUND.....	75
使用轴编程的线性方向运动.....	277	SIN.....	75
使用坐标编程的直线方向运动.....	278	SQRT.....	75
事件相关进给.....	249	TAN.....	75
手动齿轮档选择.....	172	简单功能.....	73
输入帮助		去除小数点后的位数.....	74
SCL.....	399	三角功能.....	75, 76
输入工具: 编程的轮廓切换.....	324	输入值的绝对值.....	74
输入工具: 镜像.....	267	输入值的平方根.....	75
镜像定向向量.....	268	四舍五入.....	75
输入工具: 缩放.....	319	指数功能.....	74
输入工具: 旋转.....	314	数制间的转换.....	77
旋转定向向量.....	315	BCD 格式.....	77
输入值的绝对值.....	74	BCD 格式的二进制值.....	77
输入值的平方根.....	75	BCD 编码数字的二进制值.....	77
数据处理		BIN 格式.....	77
顺序文件结构.....	501	数字化	
文件名.....	500	距离控制.....	233
数控 编程.....	21	数字化距离控制.....	233
数控程序		数字接口访问.....	431
程序块.....	25	数组变量.....	67
附加条件.....	26	双精度变量.....	67
基础组件.....	25	顺序文件结构.....	501
命令.....	25	四舍五入.....	75
数控段.....	565	速度编程.....	338
数控功能, 具有通用语言语法		速度变化图 (斜坡函数)	
模数偏置用于无限轴.....	273	常量进给插补器, 效果.....	265
数控函数 - 基础知识.....	21	速度控制的诊断.....	364
数控函数, 语法符合 DIN 66025.....	97	速度曲线	
数控函数程序字		定义单独.....	264
参数列表中的 CPL 表达式.....	28	速度曲线 (斜坡函数).....	264
带特定语法元素的参数列表.....	28	U 轴摆动循环.....	265
数控函数的		加速插补器.....	264
程序字.....	27	减速插补器.....	265
数控函数的程序字.....	27	速度限制.....	324
参数列表中的参数.....	28	速度指定	
无特定语法元素的参数列表.....	28	飞行定位.....	174
数控函数语法符合 DIN 66025		速率规定.....	173
G 代码.....	98	随机文件结构.....	501
数控函数 - 概述.....	535	缩略语.....	17

索引

缩放..... 319

T

TAN..... 75

TangTool..... 351

TangToolOri..... 353

TappRet1..... 355

TappRet2..... 355

TappSp..... 356

TCM(...)..... 350

TCS..... 358

TCS 中的行进..... 350

TcsDef..... 358

TCV..... 400, 401

ThreadSet..... 360

ThreadSet(...)..... 360

ThreadSet(DYN)..... 360

ThreadSet(ROn)..... 361

ThreadSet(SPC)..... 361

ThreadSet(TCI)..... 361

Time recording

Date..... 404

Tool database

Save..... 488

Trans..... 362

TRIM \$..... 528

TRIM\$..... 528

TRS..... 362

TSP..... 356

TST..... 360

TTL..... 351

TTO..... 353

特殊螺纹切削函数..... 359

配置动力学..... 360

配置缩回数据..... 360

启用缩回..... 361

切换主轴操作模式..... 361

影响通道 IF 信号..... 361

体积补偿..... 363

条件跳转到标准数控块..... 54

跳转指令..... 52

停留时间..... 106

通道 ID..... 34

通道号..... 34

通道名称..... 34

通过 CALL 在 CPL 中调用子程序..... 43

通过调用程序中的 CALL

DIN 标识符在 CPL 中调用子程序..... 44

通过高速信号实时阻止切换..... 249

通过高速信号实现带挡块的块过渡

事件相关的进给运动停止..... 251

通过高速信号实现带挡块的块切换..... 251

通过高速信号实现加工中块过渡..... 249

通过速度指定编程的旋转轴..... 568

通用语言语法

编程有效半径因子 ActRadFact、ARF..... 182

同步功能在程序段执行时间

触发位事件..... 84

等待位事件..... 82

删除位事件..... 84

位事件..... 83

同步模式窗口..... 329, 330

同步模式错误窗口..... 329

同步系统轴耦合..... 317

投诉..... 17

退出程序..... 167

退出诊断记录..... 235

V

VAP..... 364

VARINF..... 423

VARVAL..... 423, 424

VCPEna..... 363

VCPLoad..... 364

VEC2EUL..... 398

VERSINF\$..... 403

VirtAxisPos..... 364

VREC_START..... 364

VREC_STOP..... 366

W

WAIT..... 79

WAIT 及等待时间..... 80

WAIT 命令..... 567

WAIT (, <空转时间>)..... 80

WAIT(BITIF(...))..... 80

WAIT (无参数)..... 79

WAITA..... 85

WaitAxis..... 367

WAITO..... 85

WAX..... 367

WCS..... 382, 383

WCS2BCS..... 398

WEV..... 82

WHILE-DO-END..... 61

WID 命令..... 565

WID3..... 368, 369

wmhperm.dat..... 65

WPV..... 86

WPV 命令..... 568

WPVE..... 86

WREV..... 83

WriteId3..... 368

WriteID3..... 368, 369

WSCS..... 429

WSTOPA..... 91

WSTOPO..... 91

为程序行提供注释..... 34

为主轴启用 C 轴模式..... 339

位置 / 速度接口切换..... 336

位置和轮廓偏置

位置偏置..... 393

位置和坐标偏置..... 393

文件操作

打开.....	504
复制.....	508
覆盖.....	506
关闭.....	506
逻辑文件号.....	504
删除.....	507
文件访问.....	567
文件访问权限.....	21
文件和目录.....	500
编辑文件.....	509
目录.....	501
随机文件结构.....	501
文件操作.....	504
文件属性.....	517
文件名.....	500
文件属性	
访问权.....	517
日期.....	517
文件大小.....	518
文件系统和文件保护.....	21
无 P 地址的	
子程序调用.....	39
无 P 地址的子程序调用.....	39
无补偿夹具的攻丝	
主轴选择.....	356
无补偿夹具的攻丝的主轴选择.....	356
无补偿卡盘的攻丝.....	139
无减速的块过渡.....	230
无平面.....	119
无条件跳转至标准数控块.....	55
X	
XOR.....	76
XTAB.....	485
XTABCR.....	485
铣削铣刀路径校正.....	128
轮廓过渡.....	130
铣削铣刀路径校正的轮廓过渡	
G43 - 弧线.....	130
铣削铣刀路径校正轮廓过渡	
G44 - 交叉.....	130
系统数据	
结构化类型的系统数据.....	422
系统数据, 结构化类型	
基本类型的分配规则.....	423
系统信息.....	71
系统主轴编程.....	334
速度编程.....	338
系统主轴编程 SSp... ..	
更改档位范围.....	335
关闭特定通道的主轴运动管理.....	335
启用 / 转换保留的主轴.....	334
位置 / 速度接口切换.....	336
限制最大速度	336
限制最小速度	336
运动编程.....	337
主轴定向.....	337

线性插补.....	100
线性样条曲线切换.....	263
限制最大速度.....	336
限制最小速度.....	336
相对尺寸编程.....	146
向后跳转至标准数控块 k GoBack (GOB).....	53
向量定向.....	280
0(...).....	280
phi、theta、psi.....	280
ROTAX(...).....	280
定向坐标.....	280
向模态子程序传递参数.....	47
向前跳转至标准数控块.....	53
向子程序传递参数	
CPL 参数列表.....	46
地址参数列表.....	44
向子程序传递参数.....	47
校准轴运动.....	197
从 NC 读取参数.....	200, 201
后向转换.....	197
前向转换.....	199
优化参数.....	198
斜面.....	161
垂直于刀具的斜面激活后的配置.....	164
写入 Sercos 参数	
WSCS.....	429
将 ID 列表中的扩展 ID 编号写入驱动器..	369
将参数列表中的值写入驱动器.....	369
将参数写入至一个或多个驱动器（同步	
轴）.....	368
将参数写入至一个或多个驱动器（用于主轴和异	
步轴）.....	368
写入 Sercos 参数	
WriteId3, WID3.....	367
写入永久 CPL 变量.....	87
SPV.....	88
SPVE.....	88
信息表示方法	
安全提示.....	16
名称与缩略语.....	17
所用符号.....	16
修正切换打开.....	145
修正切换关闭.....	145
虚拟驱动器.....	364
旋转.....	314
旋转轮廓.....	314
选定的附加坐标耦合.....	321
SelCrdCouple、SCC.....	321
使用表进行.....	321
选定的附加坐标耦合 SelCrdCouple, SCC.....	321
选定的使用表进行的附加坐标耦合.....	321
选择一个等待策略.....	341
选择轴线转换	
动作点.....	225
选择轴转换.....	225
选择准停定位窗口.....	254
循环轴的定位类型.....	293

索引

Y

样条	
C1-常系数三次.....	111
C2-常系数三次.....	111
样条编程.....	108
样条参数长度.....	112
样条类型.....	108, 342
样条类型 0.....	108
样条类型 1.....	110
样条类型 1 和 2.....	111
样条类型 3.....	112
样条类型 4.....	113
样条类型的定义.....	342
样条类型定义.....	342
样条耦合表.....	227
一般的 XML 表格	
创建.....	485
移除（取消激活）耦合组.....	327
移除逻辑轴名称.....	308
移除通道主轴.....	309
移动至固定挡块	
移动至固定挡块.....	245
以 C 轴模式切换主轴.....	339
异步轴	
速度.....	176
异步轴速度.....	176
异步子程序	
定义返回到轮廓的点.....	195
异步子程序：重新定位各个坐标.....	144
异步子例程	
登出.....	193
登入.....	195
定义异步子例程的路径返回点.....	310
定义重新定位行为.....	310
关闭.....	194
开启.....	194
通过程序触发.....	196
抑制碰撞监测.....	322
英制编程.....	140
影响通道 IF 信号.....	361
永久变量.....	63
用户定义的速度参数文件.....	111
用户界面的注解	
通道 - 独立.....	36
用户界面上的注解.....	35
用户界面中的注释	
特定通道.....	36
用例 LeapFrog	
用例 LeapFrog.....	48
用于无限轴的模数偏置.....	273
用于铣削铣刀路径校正的轮廓过渡.....	130
用于异步轴的形状.....	117
优化加工时间.....	564
有条件程序暂停.....	167
有条件程序中断.....	167
与 DIN / CPL ID 链接.....	25
与客户端通信.....	532
与路径速度相关的激光功率控制器.....	261
与位置相关的高速输出.....	291
预期使用	
简介.....	19
应用领域.....	19
用例.....	19
域变量（数组）.....	67
圆 / 螺旋 / 螺旋 N 插补	
可选参数.....	104
圆 / 螺旋 / 螺旋 N 插值	
半径编程.....	102
圆弧 / 螺旋 / 螺旋 N 插补.....	101
圆弧/螺旋/螺旋 N 插补	
中心点编程.....	103
圆弧插补.....	101
圆周入口，切向.....	107
运动编程.....	337
运动停止，直到达到基本工件位置	
BSTOPA.....	90
BSTOP0.....	90
运动停止，直到到达工件位置	
WSTOPA.....	91
WSTOP0.....	91
运动停止，直到到达某个轴位置	
ASTOPA.....	89
ASTOP0.....	89
运动信息.....	21
运动至固定挡块.....	245
扭矩减小固定挡块.....	245
Z	
Zero 点偏置	
创建 Z0 表.....	388
Z0.....	135
ZOCDEL.....	389
ZOCINS.....	388
ZOS.....	370
ZOT.....	390, 391
ZOTCR.....	388
ZoTSel.....	370
ZOV.....	392, 393
在程序坐标中定义 TCS.....	357
在工件坐标系中进行在线修正.....	252
在文件上写入	
不带分号的 PRN# 命令.....	510
带分号的 PRN# 命令.....	509
在系统数据队列中写入.....	332
LastFilled.....	332
LastGet.....	332
增量进给循环轴的模态计算	
模态范围.....	272
用于通道中的单根循环轴.....	272
用于通道中的所有循环轴.....	272
增量行程循环轴的模数计算.....	272
张量定向.....	282
0(...).....	283
phi、theta、psi.....	282

ROTAX(...)	282	类型转换	521
列向量 (0x, 0y, 0z)	282	字符串处理的基础	
整数变量	66	标注	519
整数变量 (INT)	66	字符串处理	520
整数常量 (INT)	78	字符串处理基础	
支持	571	字符串处理搜索	520
执行时写入系统数据	92	自持式	32
SSD	92	自定义模态子程序调用	42
SSDE	92	自动齿轮档选择	172
执行所有组的活动辅助功能	203	自动探测转角和直线	221
直接速度编程	155	自动探测转角或线条	
直径编程	231	c2-常量系数三次样条	222
直线度和角度误差补偿	247	自动语法转换为地址参数列表	47
启用	247	自由平面选择 (与 WCS 无关)	122
指令—数控程序		自由曲面加工	565
带坐标规定的路径功能	25	子程序	37
路径功能	25	本地有效	38
指数功能	74	错误排除	49
中断程序	167	带 P 地址的调用	38
中心点编程	103	模态有效	38
重新定位各个坐标	144	使用 G 和 M 代码的自定义调用	42
轴编程	108, 110	通过 CALL 命令在 CPL 中调用	43
轴进给耦合	210	向子程序传递参数	44
轴耦合	207	自定义模态调用	42
常量耦合系数	207	子程序额	
常量偏置	207	嵌套深度	49
主轴	386	子程序返回任何程序段	56
主轴: 位置和速度模式之间切换	345	子程序结束	32
主轴定向	337	子例程	
主轴逆时针旋转	169	本地子程序	40
主轴逆时针旋转和冷却液打开	169	最大径向加速度	304
主轴耦合		最大轴过冲	
从动主轴的耦合距离	328	变更	213
带活动耦合的角度偏置	329	改变	203
等待角度偏置	330	缓冲	203, 213
等待同步模式	331	最大轴速度	
同步模式窗口	330	改变	216
同步模式错误窗口	329	缓冲	216
主轴切换	266	作为参数的	
主轴顺时针旋转	168	程序字	29
主轴顺时针旋转和冷却液打开	169	作为参数的程序字	29
主轴停止	170	带后续参数列表的参数	30
注释	566	带后续值的参数	29
转角修圆		坐标编程	108, 110, 111, 112
指定半径规范	316	坐标和轴位置	
指定偏差	316	Axis 指令位置	378
指定偏差的	316	读取程序坐标	381
准确停止关闭 G62 的路径坡度 G9	564	读取工件位置	382
准确停止关闭 AUS G62 的路径斜坡 G8	564	读取机器坐标	380
准确定位开启	564	读取实际轴值	379
准确定位开启 / 关闭	137	读取系统轴的命令位置	382
准确定位开启 G61 的 Inpos 窗口	564	坐标和轴值	
字符串变量	68	测量单位	372
字符串常量	78	坐标轴和坐标	371
字符串处理			
比较字符串表达式	521		
更改字符串	523		
基本	519		

笔记

笔记

Bosch Rexroth AG

Electric Drives and Controls

P.O. Box 13 57

97803 Lohr, Germany

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Germany

Phone +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911389415