SIEMENS

SINUMERIK
SINUMERIK 808D ADVANCED
功能手册

法律资讯

警告提示系统

为了您的人身安全以及避免财产损失，必须注意本手册中的提示。人身安全的提示用一个警告三角表示，仅与财产损失有关的提示不带警告三角。警告提示根据危险等级由高到低如下表示。

⚠危险
表示如果不采取相应的小心措施，**将会**导致死亡或者严重的人身伤害。

⚠警告
表示如果不采取相应的小心措施，**可能**导致死亡或者严重的人身伤害。

⚠小心
表示如果不采取相应的小心措施，可能导致轻微的人身伤害。

注意
表示如果不采取相应的小心措施，可能导致财产损失。

当出现多个危险等级的情况下，每次总是使用最高等级的警告提示。如果在某个警告提示中带有警告可能导致人身伤害的警告三角，则可能在该警告提示中另外还附带有可能导致财产损失的警告。

合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。 由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用 Siemens 产品

请注意下列说明：

⚠警告
Siemens 产品只允许用于目录和相关技术文件中规定的使用情况。如果要使用其他公司的产品和组件，必须得到 Siemens 推荐和允许。正确的运输、储存、组装、装配、安装、调试、操作和维护是产品安全、正常运行的前提。必须保证允许的环境条件。必须注意相关文件中的提示。

© Siemens AG 2018. 保留所有权利
6FC5397-7EP40-0RA4, 07/2018

前言

适用产品

该手册适用于以下数控系统：

数控系统	软件版本
SINUMERIK 808D ADVANCED T（车削） SINUMERIK 808D ADVANCED M（铣削）	V4.90：PPU16x.3/PPU15x.3，带主轴/进给伺服系统

文档组成与目标读者

最终用户文档	目标读者
编程和操作手册（车削）	车床的编程人员和操作人员
编程和操作手册（铣削）	铣床的编程人员和操作人员
编程和操作手册（ISO 车削/铣削）	车床/铣床的编程人员和操作人员
编程和操作手册（手动机床（MM+），车削）	车床的编程人员和操作人员
诊断手册	机械和电气设计人员，调试工程师，机床操作人员和维修服务人员
制造商/维修文档	**目标读者**
调试手册	安装人员，调试工程师和维修服务人员
功能手册	机械和电气设计人员，技术专家
参数手册	机械和电气设计人员，技术专家
维修手册	机械和电气设计人员、技术专家、调试工程师以及服务和维护人员
自述文件	
第三方软件 - 许可条件和版权说明	

我的文档管理器（MDM）

如何在西门子文档内容的基础上创建自定义文档，请访问以下链接：

www.siemens.com/mdm

标准功能范畴

本手册仅描述了标准功能范畴。机床制造商增添或者更改的功能，由机床制造商资料进行说明。

技术支持

国家	热线 1)	其它服务联系信息：
德国	+49 911 895 7222	• 国际网站： https://support.industry.siemens.com/cs/ww/en/ • 中国网站： http://www.siemens.com.cn/808D
中国	+86 400 810 4288	

1) 如需获取其他热线信息，可查看以上国际网站。

欧盟一致性声明

欧盟指令的符合性声明请访问网址：https://support.industry.siemens.com/cs/ww/en/。

在网页中输入搜索关键字“59843164”或联系您所在地区的西门子办事处。

目录

前言 **2**

1 基本安全说明 **11**

1.1 一般安全说明 11

1.2 应用示例的质保规定 12

1.3 工业安全 12

2 说明 **12**

3 各种接口信号（A2） 13

3.1 概述 13

3.2 PLC 到 NCK 的信号 15
3.2.1 存取权限 15
3.2.2 一般信号 15
3.2.3 数字驱动信号，到进给轴/主轴信号 16

3.3 NCK 到 PLC 的信号 17
3.3.1 一般信号 17
3.3.2 电子驱动信号，来自进给轴/主轴 18

3.4 PLC 到 HMI 的信号 18

3.5 HMI 到 PLC 信号 19

3.6 用户界面 19
3.6.1 概述（OF） 19
3.6.2 PI 通讯 ASUP 20
3.6.3 从 NCK 区读取变量 21
3.6.4 写入 NCK 区的变量 22

3.7 NC 变量 23

3.8 PLC 的信号 25

4 进给轴监控（A3） 25

4.1 监控功能概述 25

4.2 运行监控功能 25
4.2.1 轮廓监控 25
4.2.2 定位监控 26
4.2.3 零速监控 27
4.2.4 夹紧监控 27
4.2.5 转速给定值监控 28
4.2.6 实际速度监控 29

4.3 静态极限监控 29
4.3.1 限位开关监控 29
4.3.2 硬件限位开关 29
4.3.3 软件限位开关 30

4.4 边界条件 31

4.5 数据表 31
4.5.1 机床数据 31
4.5.2 接口信号 32

5 连续路径加工，准停方式和预读（B1） 32

5.1 简要描述 32

5.2 概述 32

5.3 准停 33

5.4 连续路径运行 34
5.4.1 概述 34
5.4.2 按过载系数降低速度 34
5.4.3 降低轨迹速度，以限制过冲 35

5.4.4 机床轴专用的过冲限制 35
5.5 压缩功能 36
5.6 预读 37
5.7 数据表 38
5.7.1 机床数据 38
5.7.2 设定数据 39
5.7.3 接口信号 39
6 加速度（B2） 39
6.1 加速度属性 39
6.2 插补级过冲限制 40
6.3 JOG 方式下的过冲限制 40
6.4 数据表 40
7 龙门轴（G1）（仅用于 PPU16x.3） 41
7.1 概述 41
7.2 “龙门轴”功能 41
7.3 龙门轴的回参考点和同步 44
7.3.1 引言 44
7.3.2 自动同步 47
7.3.3 特点 47
7.4 调试龙门轴 48
7.5 龙门轴中的 PLC 接口信号 51
7.6 龙门轴的其它事项 52
7.7 示例 53
7.7.1 创建龙门联结 53
7.7.2 NCK-PLC 接口设置 54
7.7.3 开始调试 54
7.7.4 设置报警极限值和故障极限值 56
7.8 数据表 56
7.8.1 机床数据 56
7.8.2 接口信号 57
8 轴耦合（M3） 58
8.1 联动 58
8.1.1 简介 58
8.1.1.1 功能 58
8.1.1.2 前提条件 58
8.1.2 一般功能 58
8.1.3 编程 60
8.1.3.1 定义并打开联动轴组（TRAILON） 60
8.1.3.2 关闭（TRAILOF） 61
8.1.4 PLC 接口信号的生效方式 61
8.1.5 耦合状态 62
8.1.6 动态性能限制 62
8.1.7 补充条件 62
8.1.8 示例 63
8.2 联动轴的动态响应 64
8.2.1 动态性能限制的参数设置 64
8.2.2 动态性能限制的程序编写 64
8.2.2.1 编程（VELOLIMA，ACCLIMA） 64
8.2.2.2 系统变量 64
8.3 数据表 65
8.3.1 机床数据 65
8.3.2 接口信号 65
9 手动运行和手轮运行（H1） 66

9.1 JOG 方式下手动操作的一般性能 66
9.2 持续运行 68
9.3 增量运行(INC) 69
9.4 JOG 方式下手轮运行 69
9.5 JOG 运行模式中的固定点运行 71
9.5.1 引言 71
9.5.2 功能性 71
9.5.3 参数设置 73
9.5.4 编程 73
9.5.5 边界条件 73
9.5.6 应用举例 73
9.6 数据表 74
9.6.1 机床数据 74
9.6.2 设定数据 75
9.6.3 接口信号 75
10 输出给 PLC 的辅助功能（H2） 76
10.1 概述 76
10.2 辅助功能编程 76
10.3 传输值和信号到 PLC 接口 77
10.4 辅助功能的分组 77
10.5 程序段搜索动作 79
10.6 辅助功能的描述 79
10.6.1 M 功能 79
10.6.2 T 功能 79
10.6.3 D 功能 79
10.6.4 H 功能 80
10.6.5 S 功能 80
10.7 数据表 80
10.7.1 机床数据 80
10.7.2 接口信号 80
11 运行方式，程序运行（K1） 81
11.1 概述 81
11.2 运行方式 81
11.2.1 运行方式 81
11.2.2 运行方式转换 82
11.2.3 各种运行方式下的功能选择 83
11.2.4 各种运行方式下的监控 83
11.2.5 各种运行方式下的锁定功能 84
11.3 加工零件程序 84
11.3.1 程序运行和零件程序的选择 84
11.3.2 启动零件程序或者零件程序段 85
11.3.3 零件程序中断 85
11.3.4 复位指令 86
11.3.5 程序控制 86
11.3.6 程序状态 87
11.3.7 通道状态 87
11.3.8 事件控制的程序调用 88
11.3.9 异步子程序 (ASUP) 93
11.3.10 操作或程序动作的反应 94
11.3.11 程序运行时间图示例 96
11.4 程序测试 96
11.4.1 程序测试概述 96
11.4.2 程序运行时无进给轴动作(PRT) 96
11.4.3 程序单段运行(SBL) 97
11.4.4 按照空运行(DRY)进给处理程序 98
11.4.5 程序段搜索:处理特定程序段 98

11.4.6 零件程序段跳跃(SKP) ... 100
11.4.7 图形模拟 ... 100
11.5 程序运行时间定时器 ... 101
11.6 工件计数器 ... 102
11.7 数据表 ... 103
11.7.1 机床数据 ... 103
11.7.2 设定数据 ... 104
11.7.3 接口信号 ... 105
12 补偿（K3） ... 106
12.1 概述 ... 106
12.2 间隙补偿 ... 107
12.3 可插补补偿 ... 107
12.3.1 概述 ... 107
12.3.2 SSFK ... 109
12.3.3 角度误差补偿 ... 110
12.3.4 双向丝杠螺距误差补偿 ... 113
12.3.4.1 功能说明 ... 113
12.3.4.2 调试 ... 114
12.3.4.3 示例 ... 116
12.4 跟随误差补偿（前馈控制） ... 119
12.4.1 概述 ... 119
12.4.2 速度前馈控制 ... 120
12.5 摩擦补偿（象限误差补偿） ... 120
12.5.1 一般功能说明 ... 120
12.5.2 边界条件 ... 120
12.5.3 补偿值恒定的摩擦补偿 ... 120
12.5.3.1 功能激活 ... 120
12.5.3.2 调试 ... 121
12.5.4 补偿值和加速度相关的摩擦补偿 ... 124
12.5.4.1 功能说明 ... 124
12.5.4.2 功能激活 ... 125
12.5.4.3 调试 ... 125
12.5.5 适用于短程序段的补偿值 ... 126
12.6 数据表 ... 126
12.6.1 机床数据 ... 126
12.6.2 设定数据 ... 126
12.6.3 接口信号 ... 126
13 动态转换（M1） ... 127
13.1 简要描述 ... 127
13.2 TRANSMIT ... 127
13.2.1 概览 ... 127
13.2.2 配置 TRANSMIT ... 128
13.3 TRACYL ... 131
13.3.1 概览 ... 131
13.3.2 配置 TRACYL ... 133
13.3.3 TRACYL 编程举例 ... 136
13.4 TRANSMIT 和 TRACYL 的特点 ... 138
13.5 数据表 ... 138
13.5.1 机床数据 ... 138
13.5.2 接口信号 ... 139
14 测量（M5） ... 139
14.1 概述 ... 139
14.2 硬件前提条件 ... 139
14.2.1 可使用的测量头 ... 139
14.2.2 测量头连接 ... 140

14.3 通道专用测量 140
14.3.1 测量方式 140
14.3.2 测量结果 141
14.4 测量精度和测试 141
14.4.1 测量精度 141
14.4.2 测量头的功能测试 142
14.5 JOG 方式下刀具测量 143
14.6 数据表 145
14.6.1 机床数据 145
14.6.2 接口信号 145
15 手动刀具测量（带 Y 轴） 145
15.1 简介 145
15.2 手动测量刀具（带 Y 轴）的准备工作 145
15.3 手动测量刀具（带 Y 轴） 147
16 急停（N2） 149
16.1 概述 149
16.2 急停运行 150
16.3 急停响应 150
16.4 数据表 151
16.4.1 机床数据 151
16.4.2 接口信号 151
17 回参考点（R1） 152
17.1 基础部分 152
17.2 在增量测量系统中回参考点 153
17.3 绝对编码器的边界条件 156
17.3.1 校正绝对值编码器 156
17.3.2 使用绝对值编码器回参考点 156
17.4 数据表 156
17.4.1 机床数据 156
17.4.2 接口信号 157
18 主轴（S1） 158
18.1 概述 158
18.2 主轴运行方式 158
18.2.1 主轴运行方式 158
18.2.2 主轴控制方式运行 159
18.2.3 主轴摆动方式运行 160
18.2.4 主轴定位方式运行 161
18.3 同步 164
18.4 变速换档 164
18.5 编程 168
18.6 主轴监控 169
18.6.1 主轴监控 169
18.6.2 轴/主轴停止 169
18.6.3 主轴在给定值范围 169
18.6.4 最大主轴转速 170
18.6.5 变速档的最大/最小转速 170
18.6.6 最大编码器极限频率 170
18.6.7 目标定位监控 171
18.7 模拟主轴 171
18.8 数据表 171
18.8.1 机床数据 171
18.8.2 设定数据 172

18.8.3 接口信号 ... 173
19 进给率（V1） ... 174
19.1 轨迹进给率 F ... 174
19.1.1 轨迹进给率 F ... 174
19.1.2 G33, G34, G35 (螺纹加工)时的进给率 ... 175
19.1.3 G63 (带补偿夹具进行攻丝)时的进给率 ... 176
19.1.4 G331, G332 (不带补偿夹具进行攻丝)时的进给率 ... 176
19.1.5 倒角/倒圆时的进给率： FRC, FRCM ... 177
19.2 快速移动 G0 ... 177
19.3 进给率控制 ... 178
19.3.1 概览 ... 178
19.3.2 进给锁定和进给/主轴停止 ... 178
19.3.3 通过机床控制面板进行进给修调 ... 179
19.4 数据表 ... 180
19.4.1 机床数据/设定数据 ... 180
19.4.2 接口信号 ... 180
20 刀具：刀具补偿（W1） ... 181
20.1 概述：刀具和刀具补偿 ... 181
20.2 刀具 ... 181
20.3 刀具补偿 ... 181
20.4 刀具补偿的特殊处理 ... 182
20.5 数据表 ... 184
20.5.1 机床数据 ... 184
20.5.2 接口信号 ... 184
21 轮廓手轮 ... 184
22 刀具参数：刀具后角 ... 187
23 PLC 轴控制 ... 188
23.1 简介 ... 188
23.2 功能接口 ... 188
23.2.1 用户接口：将 NC 轴作为 PLC 轴 ... 188
23.2.2 用户接口：功能 ... 189
23.2.3 主轴定位 ... 190
23.2.4 主轴旋转 ... 191
23.2.5 主轴往复 ... 192
23.2.6 分度轴 ... 193
23.2.7 定位轴（公制） ... 194
23.2.8 定位轴（英制） ... 195
23.2.9 带手轮倍率的公制定位轴 ... 196
23.2.10 带手轮倍率的英制定位轴 ... 197
23.2.11 通过选择自动变速档旋转主轴 ... 198
23.2.12 通过恒定切削速度[m/min]旋转主轴 ... 199
23.2.13 通过恒定切削速度[feet/min]旋转主轴 ... 200
23.2.14 故障消息 ... 201
23.3 PLC 轴定位示例 ... 202
24 定位轴（P2）（仅用于 PPU16x.3） ... 204
24.1 简要说明 ... 204
24.2 定位轴 ... 206
24.3 运动特性和插补过程 ... 207
24.3.1 轨迹插补器和轴插补器 ... 207
24.3.2 自控单轴运动 ... 207
24.4 定位轴动态响应 ... 211
24.5 编程 ... 212
24.5.1 简介 ... 212

24.5.2 外部编程时的旋转进给率 213

24.6 程序段切换 214
24.6.1 程序段切换 214
24.6.2 可设置的程序段切换时间点 215
24.6.3 程序段搜索时的运动结束条件 219

24.7 通过 PLC 控制 219

24.8 特殊功能下的特性 219
24.8.1 空运行进给率（DRY RUN） 219
24.8.2 单程序段 219

24.9 示例 220

24.10 数据表 220
24.10.1 机床数据 220
24.10.2 设定数据 221
24.10.3 接口信号 221

25 主轴同步（S3） 222

25.1 简要说明 222
25.1.1 功能 222
25.1.2 主轴同步 223
25.1.3 主轴同步的前提条件 227
25.1.4 通过零件程序来激活主轴同步 227
25.1.5 通过零件程序来撤销主轴同步 228
25.1.6 通过 PLC 控制同步主轴耦合 229
25.1.7 监控主轴同步 231

25.2 编程 232
25.2.1 定义（COUPDEF） 232
25.2.2 激活和撤销耦合（COUPON、COUPONC、COUPOF） 234
25.2.3 主轴同步的轴系统变量 235
25.2.4 自动激活和撤销位置闭环控制 235

25.3 定义 236
25.3.1 NC 启动时同步主轴耦合的响应 237
25.3.2 复位时同步主轴耦合的特性 237

25.4 特点 237
25.4.1 主轴同步的一般特点 237
25.4.2 恢复联动主轴的同步 238
25.4.3 主轴同步和 NC/PLC 接口信号 240
25.4.4 引导主轴和联动主轴之间的差速 242
25.4.5 同步补偿中同步信号的特性 245
25.4.6 删除同步补偿及 NC 复位 245
25.4.7 同步主轴耦合功能调试的特殊性 245

25.5 边界条件 248

25.6 示例 248

25.7 数据表 249
25.7.1 机床数据 249
25.7.2 设定数据 250
25.7.3 接口信号 250

26 安全集成（Safety Integrated） 251

26.1 标准和规定 251
26.1.1 概述 251
26.1.1.1 目标 251
26.1.1.2 功能安全 251
26.1.2 欧洲的机械安全 251
26.1.2.1 机械指令 251
26.1.2.2 欧洲协调标准 251
26.1.2.3 控制系统安全设计相关标准 252
26.1.2.4 DIN EN ISO 13849-1 253
26.1.2.5 EN 62061 253
26.1.2.6 系列标准 EN 61508（VDE 0803） 254
26.1.2.7 风险分析/评估 255

26.1.2.8 风险降低 256
26.1.2.9 遗留风险 256
26.1.3 美国的机械安全 256
26.1.3.1 OSHA 的最低要求 256
26.1.3.2 NRTL 清单 256
26.1.3.3 NFPA 79 256
26.1.3.4 ANSI B11 257
26.1.4 日本的机械安全 257
26.1.5 企业设备规定 258

26.2 SINAMICS Safety Integrated 概述 258

26.3 系统特性 258
26.3.1 认证 258
26.3.2 安全说明 258
26.3.3 安全功能的故障概率（PFH 值） 259
26.3.4 响应时间 259
26.3.5 遗留风险 259

26.4 Safety Integrated 基本功能说明 260
26.4.1 安全扭矩停止（STO） 260
26.4.2 强制潜在故障检查 262

27 特殊功能 262

27.1 对机床制造商 HMI 数据的多语言支持 262

27.2 调用在线帮助 264
27.2.1 帮助系统 264
27.2.2 调用机床制造商在线帮助及手册 267

27.3 调用具备辅助功能的标准循环 270

27.4 显示功能 271

27.5 Prog_Event 功能 273

27.6 快速输入/输出 274

27.7 创建用户循环 275
27.7.1 创建用户扩展文本文件 275
27.7.2 创建用户循环的软键索引文件 275
27.7.3 创建用户循环的参数文件 276
27.7.4 创建用户循环文件 277
27.7.5 创建用户循环的告警文件 278
27.7.6 创建用户循环的位图文件 279
27.7.7 传输所需的文件至数控系统 279
27.7.8 调用已创建的用户循环 282
27.7.9 编辑用户循环窗口 284

27.8 使用机床制造商的机床数据描述 284

27.9 使用机床制造商 R 参数名称 285

27.10 通过自定义 EasyXLanguage 脚本创建用户对话框 286
27.10.1 功能范围 286
27.10.2 配置基础 287
27.10.3 配置文件（EasyXLanguage 脚本） 288
27.10.4 配置文件的结构 290
27.10.5 语言相关性 290
27.10.6 XML 标签 290
27.10.6.1 通用结构 290
27.10.6.2 指令/标签说明 291
27.10.6.3 系统变量 311
27.10.6.4 颜色代码 312
27.10.6.5 专用 XML 句法 312
27.10.6.6 运算符 312
27.10.6.7 创建软键菜单和对话框窗口 313
27.10.7 创建用户菜单 336
27.10.7.1 创建加工循环窗口 336
27.10.7.2 替代符号 337
27.10.8 组件地址分配 338
27.10.8.1 PLC 地址分配 338

27.10.8.2 NC 变量地址分配 338
27.10.8.3 在运行时生成 NC/PLC 地址 340
27.10.8.4 驱动组件地址分配 340
27.10.8.5 机床数据和设定数据的地址分配 341
27.10.8.6 用户数据的地址分配 341
27.10.8.7 通过组件地址分配在用户对话框中创建典型菜单 342
27.10.9 预定义函数 347
27.11 快捷键 373
27.12 放映幻灯片 374
27.13 定义服务计划 375
27.14 使用机床制造商开机启动画面和机床商标 377
27.15 配置启动后的操作区域 380
27.16 数控锁功能 381
27.16.1 功能概述 381
27.16.2 前提条件 381
27.16.3 限制 381
27.16.4 防止篡改 382
27.16.5 创建激活文件 383
27.16.6 导入激活文件 384
27.16.7 延长数控锁功能 385
27.16.8 撤销数控锁功能 386
27.16.9 更换损坏的数控系统硬件（PPU）及/或 CF 卡 387
27.16.10 忘记 OEM PIN 388
27.16.11 其它信息 389
27.17 切换几何轴 390
27.18 循环保护 391
27.18.1 加密循环文件 392
27.18.2 加密循环的使用 394
28 SINUMERIK 808D ADVANCED 的许可证 395
28.1 激活可选功能 395
28.2 Internet 链接 397
28.3 重要授权条款 397
A 附录 398

1 基本安全说明

1.1 一般安全说明

⚠警告

未遵循安全说明和遗留风险可引发生命危险

忽视随附硬件文档中的安全说明和遗留风险会导致重伤或死亡。

- 遵守硬件文档中的安全说明。
- 进行风险评估时应考虑到遗留风险。

⚠警告

因参数设置错误或修改参数设置引起机器误操作

参数设置错误可导致机器出现误操作，从而导致人员重伤或死亡。

- 防止恶意访问参数设置。
- 采取适当措施（如驻停或急停）应答可能的误操作。

1.2 应用示例的质保规定

应用示例在组态和配置以及各种突发事件方面对设备没有强制约束力，无需一一遵循。应用示例不会提供客户专用的解决方案，仅在典型任务设置中提供保护。

用户自行负责上述产品的规范运行事宜。应用示例并没有解除您在应用、安装、运行和维护时确保安全环境的责任。

1.3 工业安全

说明

工业安全

西门子为其产品及解决方案提供工业安全功能，以支持工厂、系统、机器和网络的安全运行。

为防止设备、系统、机器和电网受到网络攻击，需执行一个全面的工业安全方案（及持续维护），以符合最新的技术标准。西门子的产品和解决方案只是此类方案的一个组成部分。

用户有防止未经授权访问其设备、系统、机器和电网的责任。系统、机器和组件只能连接至企业网络或互联网并采取相应的保护措施（如使用防火墙和网络分段）。

此外，还须注意西门子针对相应保护措施的建议。更多有关工业安全的信息，请访问：

工业安全 (http://www.siemens.com/industrialsecurity)

有鉴于此，西门子不断对产品和解决方案进行开发和完善。西门子强烈推荐进行更新，从而始终使用最新的产品版本。使用过时或不再支持的版本可能会增大网络攻击的风险。

为了能始终获取产品更新信息，请通过以下链接订阅西门子工业安全 RSS Feed：

工业安全 (http://www.siemens.com/industrialsecurity)

其它信息请上网查找：

工业安全功能选型手册 (https://support.industry.siemens.com/cs/cn/zh/view/108862708/en)

⚠ 警告

篡改软件会引起不安全的驱动状态

篡改软件（如：病毒、木马、蠕虫、恶意软件）可使设备处于不安全的运行状态，从而可能导致死亡、重伤和财产损失。

- 请使用最新版软件。
- 根据当前技术版本，将自动化组件和驱动组件整合至设备或机器的整体工业安全机制中。
- 在整体工业安全机制中要注意所有使用的产品。
- 采取相应的保护措施（如杀毒软件）防止移动存储设备中的文件受到恶意软件的破坏。
- 请激活变频器功能“专有技术保护”，以防止对驱动进行未经授权的改装。

2 说明

注释

本手册中使用以下注释和缩写：

- 可编程逻辑控制器（PLC）接口信号 -> 接口信号“信号名称”（信号数据）
  示例：接口信号“进给倍率”（DB380x.DBB0）
  变量字节表示“进给轴”范围，“x”表示轴：
  - 0 为进给轴 1
  - 1 为进给轴 2
  - n 为进给轴 n+1
- 机床数据 -> MD MD_NR MD_NAME（描述）
  例如：MD30300 IS_ROT_AX（旋转轴）
- 设定数据 -> SD SD_NR SD_NAME（描述）
  例如：SD41200 JOG_SPIND_SET_VELO（主轴的 JOG 速度）

机床数据和设定数据分配在下列范围中：

范围	数据区	含义
200 - 9999	$MM_	显示机床数据
10,000 - 19,999	$MN_	通用机床数据
20,000 - 28,999	$MC_	通道专用机床数据
30,000 - 38,999	$MA_	轴专用机床数据
41,000 - 41,999	$SN_	一般设定数据
42,000 - 42,999	$SC_	通道专用设定数据
43,000 - 43,999	$SA_	轴专用设定数据

技术数据的说明

数据类型：
系统中使用以下的数据类型：

- DOUBLE
  浮点值（64 位值）
  输入范围从 +/-4.19*10^{-307} 到 +/-1.67*10^{308}
- DWORD
  整型值（32 位值）
  输入值极限为-2,147,483,648 到+2,147,483,648（十进制）；作为十六进制值：0000 到 FFFF
- BYTE
  整型值（8 位值）
  输入范围从 -128 到 +127（十进制）；作为十六进制值：00 到 FF
- BOOLEAN
  布尔数值：TRUE (1) 或 FALSE (0)
- STRING
  最多包含 16 个美国信息交换标准码（ASCII）字符（大写字母、数字和下划线）

详细的说明

- 关于使用的机床数据/设定数据和接口信号的详细说明请参见 SINUMERIK 808D ADVANCED 参数手册。
- 有关报警的具体说明，请参考 SINUMERIK 808D ADVANCED 诊断手册。

3 各种接口信号（A2）

3.1 概述

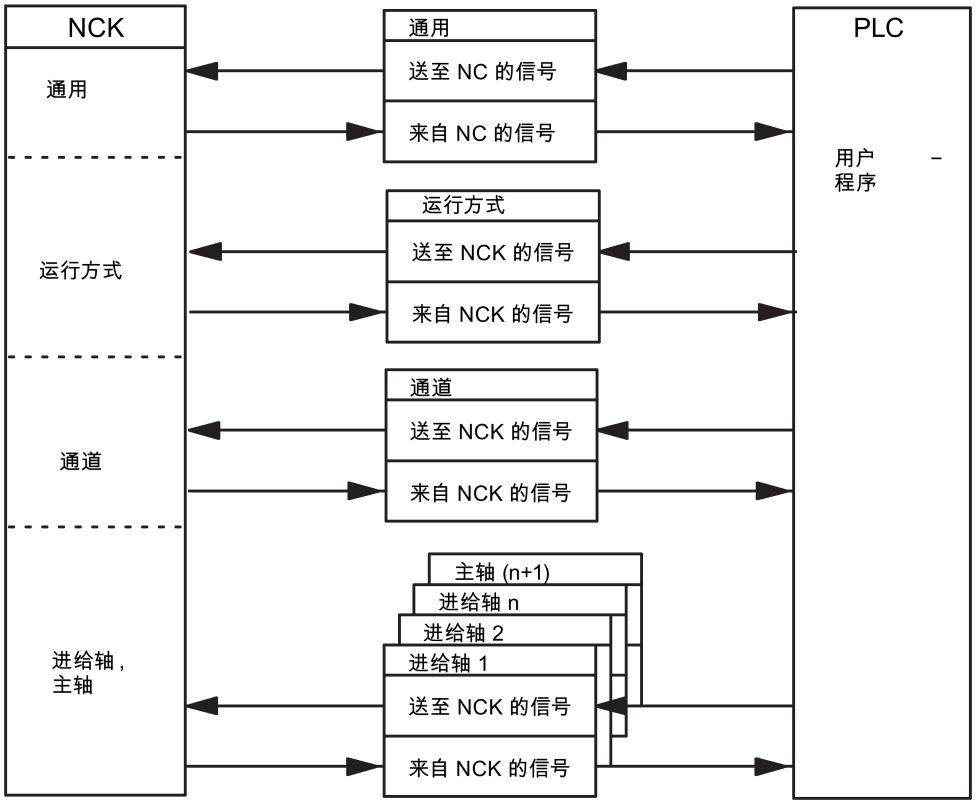
概述

本章描述各种接口信号的一般功能，并且在其它的功能相关章节中没有加以说明。

接口

通过不同的数据区进行 PLC 用户程序和 NCK（数控系统内核）或者 HMI（显示单元）之间信号和数据的交换。PLC 用户程序与数据及信号交换无关。对使用者来说这是自动进行的。

PLC/NCK 接口



循环信号交换

PLC/NCK 控制及其状态信号是循环更新的。

这些信号可以细分为下列组（见上图）：

- 一般信号
- 运行方式信号
- 通道信号
- 轴/主轴信号

PLC 接口信号地址表示方式说明

目前 HMI 上的 PLC 接口信号地址以 V 结构表示，而手册中的 PLC 接口信号地址以 DB 结构表示。

下表给出了这两种表示方法之间的对应关系。

V 结构		**DB 结构**	
存取对象	示例	示例	存取对象
位	V38000002.1	DB3800.DBX2.1	位
字节	VB38000002	DB3800.DBB2	字节
字	VW38000002	DB3800.DBW2	字
双字	VD38000004	DB3800.DBD4	双字

3.2 PLC 到 NCK 的信号

3.2.1 存取权限

存取权限

对程序、数据和功能的访问是通过保护级来实现面向用户并加以控制的。数控系统提供存取等级方案用以释放数据区。从下表中，您可以看到此类信息：

存取级别	默认口令	目标使用人群
西门子（级别 0）	-	保留用于西门子
制造商（级别 1）	SUNRISE	机床制造商
预留（级别 2）	-	-
最终用户（级别 3-6）	CUSTOMER	最终用户
无口令（级别 7）	-	最终用户

这为控制访问权限提供了一个多层级安全概念。

文献：

SINUMERIK 808D ADVANCED 调试手册，章节：设置口令

3.2.2 一般信号

删除剩余行程（DB3200.DBX6.2）

接口信号“删除剩余行程（通道专用）”只适用于路径轴。

当接口信号处于上升沿时，路径轴将按斜率停止；所有路径轴的剩余行程将被删除。然后，开始执行下一个程序段。

进给轴/主轴禁止（DB380x.DBX1.3）

接口信号“进给轴/主轴禁止”可用于测试。

进给轴禁止（用于进给轴）：

发出“轴禁止”信号后，进给轴不再给位置控制器发出位置给定值；因此该轴的运行被禁止。位置控制回路锁闭，剩余的跟随误差将被补偿。如果在轴处于禁止状态时使进给轴运行，则在屏幕上显示实际值位置的地方显示给定位置，显示速度实际值的地方显示给定速度，此时机床坐标轴实际上并没有运行。使用接口信号“复位”（DB3000.DBX0.7）后，位置实际值显示机床的实际位置。进给轴的运行指令继续传送到 PLC。如果取消该接口信号，则相应的进给轴又可正常运行。如果在进给轴运行状态时发出接口信号“轴禁止”，则该轴按照斜坡制动到停止。

主轴禁止（用于主轴）：

如果发出“主轴禁止”信号，则与在进给轴时一样，主轴在受控运行中也不向转速控制器输出转速给定值，处于定位方式时也不向位置控制器输出位置给定值。由此禁止主轴的运行。同样，在转速实际值处显示转速给定值。主轴禁止只有通过“复位”指令或 M2 和程序重新启动消除。如果在主轴旋转时发出“主轴禁止”信号，则主轴按照加速特性曲线制动停止。

取消激活：

只有当进给轴/主轴停下以后（也就是说不再有插补值时），取消“进给轴/主轴禁止”（下降沿变化 1 → 0）才有效。给出新的给定值后开始新的运行。（例如：在“自动”运行方式下运行一段新的程序段）。

说明：模拟轴和实际轴的实际值不同！

跟随模式（DB380x.DBX1.4）

如果坐标轴或主轴位于跟随模式，它的给定值位置将始终跟随当前实际值位置。在跟随模式下，位置给定值不是由插补器定义，而是取决于当前的实际位置。由于轴的位置实际值不断变化，取消跟随模式后无需让轴重新回参考点。

跟随模式下，零速监控、夹紧监控和定位监控无效。

生效：
如果驱动的控制器使能被取消（如通过“驱动使能 = 0”信号或数控系统内部出现故障）或再次发出驱动使能信号，接口信号“跟随模式”才有效。

接口信号“跟随模式” = 1：
如果“驱动使能”信号被取消，轴的位置给定值将连续跟随实际值。此状态通过传输给 PLC 的接口信号“跟随模式有效”

（DB390x.DBX1.3）显示。在一个零件程序生效时，如果再次设置了接口信号“驱动使能”，则在数控系统内部重新定位（REPOSA：所有轴按照直线返回）到上次编程的位置。否则轴运动将从新的实际位置（可能已改变）开始。

接口信号“跟随模式” = 0：
如果信号“驱动使能”被取消，将保留旧的位置给定值。如果轴偏离该位置，在位置给定 值和实际值间将产生一个跟随误差，该跟随误差将在“驱动使能”信号设定时被补偿。轴运动将从“驱动使能”信号取消之前的设定位置开始。在“停止”状态，接口信号“跟随方式有效”（DB390x.DBX1.3）的状态为 0。夹紧监控或零速度监控有效。

位置测量系统 1（DB380x.DBX1.5）

主轴上可以连接一个位置测量系统。这种情况下对于主轴必须要设置该信号。

进给轴始终需要此信号。必须提供位置测量系统。

驱动使能（DB380x.DBX2.1）

在给驱动发出驱动使能时，接通进给轴/主轴的位置控制回路。因此，进给轴/主轴处于位置控制状态。

取消驱动使能后位置控制回路断开，进给轴/主轴的转速控制回路也延迟断开。

接口信号“位置控制器有效”（DB390x.DBX1.5）设置为 0 信号（反馈信号）。

激活：

驱动使能的设置和取消可以有以下几种情况：

1. 通过带接口信号“驱动使能”的 PLC 用户程序（正常情况）
   应用：在夹紧进给轴/主轴之前取消驱动使能。
2. 如果在机床、驱动、位置测量系统以及控制器中出现特定故障时，可以在系统内部取消驱动使能（故障情况）
   应用：在出现故障时，处于运行状态的进给轴必须通过急停使其制动。
3. 在数控系统内部的下列事件中：存在接口信号“急停”（DB2600.DBX0.1）

对于正在运行的进给轴/主轴执行**取消驱动使能**：

- 主轴按照 MD36610 AX_EMERGENCY_STOP_TIME（出错状态时减速斜坡持续时间）中所设定的时间制动到停止。同时给出报警 21612“在运行时复位控制器使能”。
- 断开进给轴/主轴的位置控制回路。“位置控制器有效”（DB390x.DBX1.5）= 0 信号反馈到 PLC。此外，启动用于驱动使能延时的计时器（MD36620 SERVO_DISABLE_DELAY_TIME（驱动使能断开延时））。
- 一旦实际值速度达到停止状态范围时，取消驱动控制器使能。“转速控制器有效”（DB390x.DBX1.6） = 0 信号反馈到 PLC。驱动使能至少在 MD36620 SERVO_DISABLE_DELAY_TIME 中设定的时间结束之后取消。
- 注意：如果将驱动使能延迟时间设置的较低，尽管轴/主轴仍然运行，驱动使能也已被取消。这时，进给轴/主轴突然停止，给定值为 0。
- 进给轴/主轴的位置实际值继续由系统获取。

只有在“复位”之后才可以再次修改进给轴/主轴的位置实际值。

进给轴处于插补方式时：

参加插补的进给轴只要取消了其中**一个**轴的驱动使能，则其它的进给轴也会立即制动停止。

进给轴会如前所述停止。所有具有几何关系的进给轴立即制动停止。此外，给出报警 21612“在运行时复位控制器使能”。NC 程序不能继续运行。

3.2.3 数字驱动信号，到进给轴/主轴信号

速度控制器积分器禁止（DB380x.DBX4001.6）

PLC 用户程序取消了驱动转速控制器的积分器使能。 因此转速控制器从 PI 转换成 P 控制器。

脉冲使能（DB380x.DBX4001.7）

PLC 用户程序给出进给轴/主轴的脉冲使能。 只有当所有的使能信号已出现，才执行驱动模块的脉冲使能。

3.3 NCK 到 PLC 的信号

3.3.1 一般信号

驱动处于循环方式中（DB2700.DBX2.5）

NCK 报告给 PLC，所有的现有驱动已达到快速运行状态，在其中用 NCK 进行循环数据交换。

驱动就绪（DB2700.DBX2.6）

由 NCK 报告给 PLC，即所有的驱动准备就绪。 接口信号“驱动就绪”（汇集信号）来自所有的进给轴/主轴。

NCK 报警有效（DB2700.DBX3.0）

系统报告给 PLC 至少出现一个 NCK 报警。 询问通道专用的接口（DB3300.DBX4.7），是否出现加工停止。

空气温度报警（DB2700.DBX3.6）

触发环境温度监控或风扇监控。

出现通道专用的 NCK 报警（DB3300.DBX4.6）

系统报告给 PLC 通道至少出现一个 NCK 报警。 此时程序中断执行还是终止执行，这可由接口信号“加工停止出现 NCK 报警”（DB3300.DBX4.7）给出。

外部编程语言有效（DB3300.DBX4001.0）

系统发出信号给 PLC，当前有效的零件程序语言不是 SIEMENS 语言。 已经使用 G291 进行了语言转换。

加工停止，NCK 报警（DB3300.DBX4.7）

系统报告给 PLC 通道至少出现一个 NCK 报警。此时程序中断执行或者终止执行(程序停止)。

跟随模式有效（DB390x.DBX1.3）

跟随模式对此轴有效。

参见章节： PLC 到 NCK 的信号，跟随模式 (页 15)（DB380x.DBX1.4）

进给轴/主轴停止（DB390x.DBX1.4）

进给轴当前的实际速度以及主轴的实际转速处于停止范围之内。 该区域的大小由机床数据 MD36060 STANDSTILL_VELO_TOL(“进给轴/主轴停止”时最大速度/转速)确定。

位置控制器有效（DB390x.DBX1.5）

进给轴/主轴的位置控制器接通，位置控制有效。

更多信息请参见控制器使能 (页 15)。

转速控制器有效（DB390x.DBX1.6）

进给轴/主轴的转速控制器接通，转速控制有效。

更多信息请参见控制器使能 (页 15)。

电流控制器有效（DB390x.DBX1.7）

进给轴/主轴的电流调节器接通，电流调节有效。

润滑脉冲（DB390x.DBX1002.0）

一旦进给轴/主轴运行距离较大，超出 MD33050 LUBRICATION_DIST(PLC 润滑移动距离)中设定的位移，就从 NCK 发出信号“润滑脉冲”，并**转换状态**。

3.3.2 电子驱动信号，来自进给轴/主轴

驱动就绪（DB390x.DBX4001.5）

驱动已准备就绪。 这样，进给轴/主轴移动的前提条件已满足。

取消转速控制器积分器使能（DB390x.DBX4001.6）

取消了转速控制器的积分器使能。 因此转速控制器从 PI 转换成 P 控制器。

脉冲使能（DB390x.DBX4001.7）

驱动模块的脉冲使能已给出。 这时进给轴主轴可以开始运行。

启动过程结束（DB390x.DBX4002.2）

信号表明，转速实际值已到达新的转速给定值并考虑了驱动所设定的公差范围。 因此，启动过程结束。 其它因负载变化而导致的转速变化对接口信号无影响。

3.4 PLC 到 HMI 的信号

按键锁定（DB1900.DBX5000.2）

接口信号“按键锁定”用于对操作面板的键盘进行锁定，1 信号时为按键锁，0 信号时取消锁定。

程序号（DB1700.DBB1000）

如果要从 PLC 选择一个 NC 程序，相应的定义的程序号将从 PLC 传输到 HMI。当前选定 的 NC 程序可以通过指令接口（参见 DB1700.DBB1001）进行保护并重新进行选择。

程序是通过它们的名称来管理的（STRING）。在分配列表中，最多可以给 255 个程序分配相应的程序号。

根据不同的保护级可以将程序号分为：

- 1 至 100：用户区（最终用户的保护级）
- 101 至 200：机床厂商（机床制造商的保护级）
- 201 至 255：SIEMENS（SIEMENS 保护级）

“程序号”（DB1700.DBB1000）和以下接口信号相关：

- “程序已经选择”（DB1700.DBX2000.0）
- “错误选择程序”（DB1700.DBX2000.1）

当写入一个 > 0 的程序号时，开始选择 PLC 中的程序。一旦 HMI 发现 > 0 的程序号，立即内部处理该任务并始终将程序号（DB1700.DBB1000）设置为 0。

PLC 等待 HMI 的响应信号：DB1700.DBX2000.0 或 DB1700.DBX2000.1，并立即分析。收到响应信号后，该信号立即提供给 PLC 循环，然后 PLC 系统自动将其删除。

指令（DB1700.DBB1001）

在此从 PLC 传递给 HMI 一个指令任务。

指令	动作
0	无
1	保存所选定程序的名称
2	选择名称被保存的程序

“指令”（DB1700.DBB1001）和以下接口信号相关：

- “执行指令”（DB1700.DBX2001.0）
- “错误的指令执行”（DB1700.DBX2001.1）

当写入一个 > 0 的指令时，开始由 PLC 发出任务。一旦 HMI 发现 > 0 的指令，立即内部处理该任务并始终将指令（DB1700.DBB1001）设置为 0。

PLC 等待 HMI 的响应信号：DB1700.DBX2001.0 或 DB1700.DBX2001.1，并立即开始分析。收到响应信号后，该信号立即提供给 PLC 循环，然后 PLC 系统自动将其删除。

3.5 HMI 到 PLC 信号

程序已经选择（DB1700.DBX2000.0）

HMI 向 PLC 发出信号，表示已经成功地选择了所需的 NC 程序。 此信号给出一个 PLC 循环。 它和 DB1700.DBB1000 相关。

错误选择程序（DB1700.DBX2000.1）

HMI 向 PLC 发出信号，表示选择所需的 NC 程序时出错。 此信号给出一个 PLC 循环。 它和 DB1700.DBB1000 相关。

执行指令（DB1700.DBX2001.0）

HMI 向 PLC 发出信号，表示已经成功地执行了所需的指令。 此信号给出一个 PLC 循环。 它和 DB1700.DBB1001 相关。

错误的指令执行（DB1700.DBX2001.1）

HMI 向 PLC 发出信号，表示执行所需的指令时出错。 此信号给出一个 PLC 循环。 它和 DB1700.DBB1001 相关。

3.6 用户界面

3.6.1 概述（OF）

通过 PLC/NCK 接口“NC 通讯”可以执行通讯任务。因此，存在以下通讯：

- 启动 NCK 区的程序实例通讯（PI 通讯）（例如，异步子程序（ASUP））
- 从 NCK 区读取变量
- 写入 NCK 区的变量

通过接口的全局部分激活各自的通讯。单独的通讯设置在下文中详细说明。

任务，全局部分

可以在某一时刻只启动一种通讯。通过 DB1200.DBX0.1 和 DB1200.DBX0.2 选择通讯。

通讯	DB1200.DBX0.2	DB1200.DBX0.1
启动 NCK 区的 PI 通讯	1	0
从 NCK 区读取变量	0	0
写入 NCK 区的变量	0	1

开始：

设置信号 DB1200.DBX0.0 = 1 后任务启动。只有当前一任务已经结束，即响应信号必须是零时（“任务结束”DB1200.DBX2000.0 和“任务出错”DB1200.DBX2000.1），才可以启动一个新的任务。

任务的执行可能持续几个 PLC 循环，并根据实际负载而有所不同，因此，该功能不是实时功能。

说明

无法中断已经启动的任务。如果在收到应答前错误地复位信号“开始”，则不更新该任务的结果信号。然而仍执行该任务。

任务，全局部分

结果由 PLC 运行系统写入，因此用户只能读曲该信号。

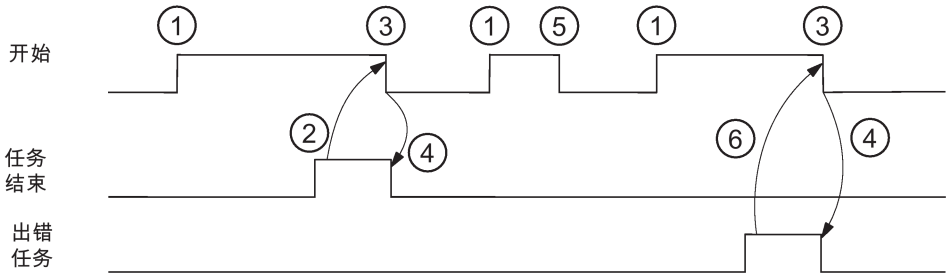
如果顺利执行任务，则信号“任务结束” DB1200.DBX2000.0 设置为 1。如果在执行读写任务时出现了错误，则设置信号“任务出错” DB1200.DBX2000.1。

DB1200.DBB2000 中的结果信号是整个任务的全局位。可能的出错原因有，例如：

- 变量数量（DB1200.DBX1）超出允许的范围
- 变量索引（DB1200.DBX1000）超出允许的范围

在分析完结果后，信号“开始”（DB1200.DBX0.0）由用户复位。接着 PLC 运行系统复位“任务结束”或者“任务出错”。

脉冲示意图：



脉冲示意图的含义：

1. 通过设置“开始”启动任务（“任务结束”和“任务出错”必须已复位）
2. 成功执行任务（还必须分析单个变量的结果）
3. 接收到结果后复位“开始”
4. 通过 PLC 运行系统进行信号转换
5. 如果在接收结果前错误地复位信号“开始”，则不更新输出信号，并且对触发功能的内部运行不产生影响。
6. 任务出错

3.6.2 PI 通讯 ASUP

PI 通讯 ASUP 功能是指通过 PLC 来调用 PLCASUP1.SPF 和 PLCASUP2.SPF 的执行。本数控系统提供了两个默认 ASUP 供 PLC 调用。

初始化

如需自定义这两个 ASUP，可从 PLC 分配中断号 1 和 2 给固定的程序名。前提是在目录 CMA 中必须存在程序 PLCASUP1_SPF 或者 PLCASUP2_SPF。

PI 索引	功能
DB1200.DBB4001 = 1	分配中断 1 给程序 CMA_DIR/PLCASUP1_SPF。 该中断具备优先级 1。
DB1200.DBB4001 = 2	分配中断 2 给程序 CMA_DIR/PLCASUP2_SPF。 该中断具备优先级 2。

初始化时注意以下几点：

- 只允许在重新启动后执行一次 PI 通讯 ASUP，并且此后一直保持。
- 初始化不能在没有激活的通道中进行。
- 如果设计了程序事件“引导启动”，则只有在结束程序事件后才启动初始化。

相关的接口信号

	地址	名称	有效值
任务	DB1200.DBX4000.0	启动	0/1
	DB1200.DBX4000.1	写入变量	0
	DB1200.DBX4000.2	PI 通讯	1
	DB1200.DBB4001	PI 索引	1,2
结果	DB1200.DBX5000.0	任务结束	0/1
	DB1200.DBX5000.1	任务出错	0/1

3.6.3 从 NCK 区读取变量

一个读取任务能够读取 1 到 8 个值（变量 x：0...7）。因此存在一个专用于变量的接口部分：

- 任务：DB120x.DBB1000
- 结果：DB120x.DBB3000

任务，量专用部分

NC 变量：

在变量索引中选择 NC 变量（DB120x.DBB1000）（参见章节“NC 变量 (页 23)”）。

区域号，列索引/行索引（DB120x.DBB1001 ... DB120x.DBB1005）

若干个变量为数组。为进行灵活编址，各个数组索引必须作为列索引和/或行索引输入（例如：R 参数号）。

值：

在读取时，DB120x.DBB1008 ... 11 区并不重要。

结果，变量专用部分

任务中包含的每个变量都将获得一个结果信息。

如果读取顺利结束，则“变量有效”(DB120x.DBX3000.0)设置为 1，而存取结果 DB120x.DBB3001 为 0。

在读取时根据类型输入从 DB120x.DBB3004 开始的数据。

出错时 DB120x.DBX3000.0 = 0，并且存取结果的条目 DB120x.DBB3001:

- 0：无错
- 3：不允许对象存取
- 5：无效地址
- 10：对象不存在

值：

在读取时已经读出的数据位于 DB120x.DBB3004...7 区中，其类型为各个变量特定的数据类型（必要时值从 64 位 转为 32 位 REAL）。

相关的接口信号

	地址	名称	有效值
任务，全局部分	DB1200.DBX0.0	启动	0/1
	DB1200.DBX0.1	写入变量	0
	DB1200.DBX0.2	PI 通讯	0
	DB1200.DBB1	变量数目	1 ... 8
任务，量专用部分	DB120x.DBB1000	变量索引	参见章节“NC 变量 (页 23)”
	DB120x.DBB1001	区域编号	
	DB120x.DBB1002	NCK 变量的行索引	
	DB120x.DBB1004	NCK 变量的列索引	
任务，全局部分	DB1200.DBX2000.0	任务结束	0/1
	DB1200.DBX2000.1	任务出错	0/1
结果，变量专用部分	DB120x.DBX3000.0	变量无效	0/1
	DB120x.DBB3001	存取结果	0/3/5/10
	DB120x.DBB3004/ DB120x.DBW3004/ DB120x.DBD3004	NCK 变量的值，数据类型取决于变量索引	参见章节“NC 变量 (页 23)”

3.6.4 写入 NCK 区的变量

一个写入任务能够写入 1 到 8 个值（变量 x：0...7）。因此存在一个专用于变量的接口部分：

- 任务：DB120x.DBB1000
- 结果：DB120x.DBB3000

任务，量专用部分

NC 变量：

在变量索引中选择 NC 变量（DB120x.DBB1000）（参见章节“NC 变量 (页 23)”）。

区域号，列索引/行索引（DB120x.DBB1001 ... DB120x.DBB1005）

若干个变量为数组。为进行灵活编址，各个数组索引必须作为列索引和/或行索引输入（例如：R 参数号）。

值：

待写入的数值必须输入到 DB120x.DBB1008...11 区中，其类型为各个变量特定的类型。

必要时可以转换数值（例如：NCK 64 位浮点值转为 PLC 32 位格式的值或相反）。从 64 位转换为 32 位 REAL 后数值精度降低。32 位 REAL 数值的最大精度大约为 10^7。

结果，变量专用部分

任务中包含的每个变量都将获得一个结果信息。

如果读取顺利结束，则“变量有效”(DB120x.DBX3000.0)设置为 1，而存取结果 DB120x.DBB3001 为 0。

在读取时根据类型输入从 DB120x.DBB3004 开始的数据。

出错时 DB120x.DBX3000.0 = 0，并且存取结果的条目 DB120x.DBB3001:

- 0：无错
- 3：不允许对象存取
- 5：无效地址
- 10：对象不存在

值：

在写入时，DB120x.DBB3004...07 区并不重要。

相关的接口信号

	地址	名称	有效值
任务，全局部分	DB1200.DBX0.0	启动	0/1
	DB1200.DBX0.1	写入变量	1
	DB1200.DBX0.2	PI 通讯	0
	DB1200.DBB1	变量数目	1 ... 8
任务，量专用部分	DB120x.DBB1000	变量索引	参见章节“NC 变量 (页 23)”
	DB120x.DBB1001	区域编号	
	DB120x.DBB1002	NCK 变量的行索引	
	DB120x.DBB1004	NCK 变量的列索引	
	DB120x.DBB3004/ DB120x.DBW3004/ DB120x.DBD3004	NCK 变量的值，数据类型取决于变量索引	
任务，全局部分	DB1200.DBX2000.0	任务结束	0/1
	DB1200.DBX2000.1	任务出错	0/1
结果，变量专用部分	DB120x.DBX3000.0	变量无效	0/1
	DB120x.DBB3001	存取结果	0/3/5/10

3.7 NC 变量

变量 cuttEdgeParam

刀具带 D 号的补偿值参数和刀沿列表

单个参数的含义取决于各个刀具的类型。 目前为每种刀沿预留了 25 个参数（但只有一部分已赋值）。 为将来可以灵活扩展，不使用 25 个参数中的一个固定值进行计算，而是使用变量值' numCuttEdgeParams'（变量索引 2）进行计算。

刀具参数的详细说明请参见章节“刀具补偿 (页 181)”。

	变量 cuttEdgeParam [r/w]
DB120x.DBB1000	1
DB120x.DBB1001	-
DB120x.DBW1002	(刀沿编号 - 1) * numCuttEdgeParams + 参数编号 (WORD)
DB120x.DBW1004	T 号(1...32000) (WORD)
DB120x.DBD1008	写入： NCK 变量 x 的数据（变量数据类型： REAL)
DB120x.DBD3004	读取： NCK 变量 x 的数据（变量数据类型： REAL)

变量 numCuttEdgeParams

一个刀沿 P 元素的数量

	变量 numCuttEdgeParams [r]
DB120x.DBB1000	2
DB120x.DBB1001	-
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	读取： NCK 变量 x 的数据（变量数据类型： WORD）

变量 linShift

可设定零点偏移（通道专用的可设定框架）的转译

只有当 MD18601 MM_NUM_GLOBAL_USER_FRAMES > 0 时，才有该变量。

框架索引有以下几种：

- 0: ACTFRAME = 当前生成的零点偏移
- 1: IFRAME = 当前可设定的零点偏移
- 2: PFRAME = 当前可编程的零点偏移
- 3: EXTFRAME = 当前外部零点偏移
- 4: TOTFRAME = 当前总零点偏移 = 由 ACTFRAME 和 EXTFRAME 得出的总和
- 5: ACTBFRAME = 当前总基本框架
- 6: SETFRAME = 当前第 1 系统框架（实际值设置，对刀）
- 7: EXTSFRAME = 当前第 2 系统框架（实际值设置，对刀）
- 8: PARTFRAME = 当前第 3 系统框架（可定向刀架的 TCARR 和 PAROT）
- 9: TOOLFRAME = 当前第 4 系统框架(TOROT 和 TOFRAME)
- 10: MEASFRAME = 工件和刀具测量的结果框架
- 11: WPFRAME = 当前第 5 系统框架（工件基准点）
- 12: CYCFRAME = 当前第 6 系统框架（循环）

最大的框架索引为 12。

numMachAxes 的数值包含在索引 4 的变量中。

	变量 linShift [r]
DB120x.DBB1000	3
DB120x.DBB1001	-
DB120x.DBW1002	框架索引 * numMachAxes + 轴号
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBD3004	读取： NCK 变量 x 的数据（变量数据类型： REAL)

变量 numMachAxes

现有通道轴的最高编号

如果没有通道轴间隙，则该值即为通道现有轴的数量。

	变量 numMachAxes [r]
DB120x.DBB1000	4
DB120x.DBB1001	-
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	读取： NCK 变量 x 的数据（变量数据类型： WORD）

变量 rpa

R 参数

	变量 rpa [r/w]
DB120x.DBB1000	5
DB120x.DBB1001	-
DB120x.DBW1002	R 参数号 + 1
DB120x.DBW1004	-
DB120x.DBD1008	写入： NCK 变量 x 的数据（变量数据类型： REAL)
DB120x.DBD3004	读取： NCK 变量 x 的数据（变量数据类型： REAL)

变量 actLineNumber

当前 NC 程序段的行数：

- 0: 程序启动前
- -1: 因故障不可用
- -2: 因 DISPLOF 不可用

	变量 actLineNumber [r]
DB120x.DBB1000	6
DB120x.DBB1001	-
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBD3004	读取： NCK 变量 x 的数据（变量数据类型： DINT)

3.8 PLC 的信号

调试模式

调试模式通过用户接口中的位 0 和位 1(DB1800.DBB1000)报告。

调试模式	DB1800.DBX1000.1	DB1800.DBX1000.0
标准启动	0	0
引导启动使用缺省值	0	1
引导启动使用保存的数据	1	0

4 进给轴监控（A3）

4.1 监控功能概述

监控功能概述

- 运动监控功能
  - 轮廓监控
  - 定位监控
  - 零速监控
  - 夹紧监控
  - 转速给定值监控
  - 实际速度监控
  - 编码器监控
- 静态极限监控
  - 限位开关监控

4.2 运行监控功能

4.2.1 轮廓监控

功能

轮廓监控功能的原理是测量的实际位置值和从 NC 位置给定值计算出的实际位置值进行的比较。为了提前计算出跟随误差，应使用一个模型来模拟包括前馈控制的位置控制的动态特性。

为了使得监控系统在转速轻微变化时不作出响应(由于负载变化而导致的速度变化)，允许使用公差带用于轮廓偏差范围。

如果超出了 MD36400 CONTOUR_TOL (轮廓监控公差带)中定义的允许的实际值偏差，则输出报警，进给轴停止。

有效性

轮廓监控适用于进给轴和位置控制的主轴。

生效

如果轮廓误差过大，会产生下述结果:

- 发出报警 25050 “轮廓监控”。
- 用快速停止功能（位置控制回路断开）通过转速给定值斜坡停止相应的进给轴/主轴。
  制动斜坡时间在 MD36610 AX_EMERGENCY_STOP_TIME（故障状态下的制动斜坡时间）中确定。
- 如果进给轴/主轴和其它的进给轴/主轴插补，随着跟随误差的减少，这些轴快速停止(位置给定值 = 0)。

处理

- 增加 MD36400 中定义的监控功能的公差范围值。
- 实际的“Kv 系数”必须和通过 MD32200 POSCTRL_GAIN（Kv 系数）设置的期望 Kv 系数一致。在模拟主轴上：检查 MD32260 RATED_VELO（给定电机转速）和 MD32250 RATED_OUTVAL（给定输出电压）。
- 检查转速控制器的优化
- 检查轴的运行的灵活性
- 检查用于运行的机床数据(进给修调，加速度，最大速度， …)。

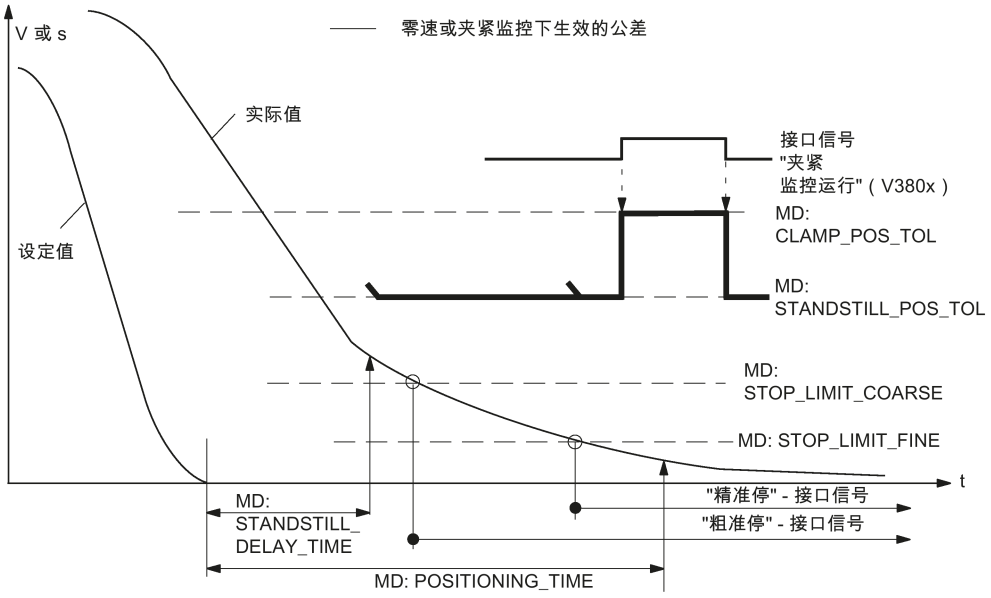
4.2.2 定位监控

功能

为了确保轴在预定的时间内到达指定点，在一个程序段运行结束后(到达给定值位置)启动 MD 36020 POSITIONING_TIME(精准停延迟)里设定的时间，并在这一时间运行结束后检查轴是否到达设定点允许的误差范围内，此误差在 MD 36010 STOP_LIMIT_FINE(精准停)中定义。

关于“粗准停和精准停”请参见章节“连续路径运行、准停和预读 (页 32)”。

定位、零速度和夹紧监控的相互关系：



有效性

定位监控始终在动作程序结束后生效(到达给定值位置)。

定位监控适用于进给轴和一个位置控制的主轴。

取消使能

在达到定义的“精准停极限”或者输出新的给定位置后（例如：定位到“粗准停”并接着转换程序段），定位监控取消使能。

生效

如果在位置监控时间结束后，“精准停”极限值还未达到，会产生下述结果：

- 发出报警 25080 “位置监控”
- 用快速停止功能（位置控制回路断开）通过转速给定值斜坡停止相应的进给轴/主轴。
  制动斜坡时间在
  MD36610 AX_EMERGENCY_STOP_TIME（故障状态下的制动斜坡时间）中确定。
- 如果进给轴/主轴和其它的进给轴/主轴插补，随着跟随误差的减少，这些轴快速停止(局部位置给定值 = 0)。

误差原因/误差消除

- 位置控制增益系数太小 --> 改变位置控制增益系数机床数据
  MD32200 POSCTRL_GAIN (Kv 系数)
- 定位窗口（精准停）、定位监控时间和位置控制增益彼此不匹配 --> 修改机床数据：
  MD36010 STOP_LIMIT_FINE （精准停），
  MD36020 POSITIONING_TIME （精准停延迟时间），
  MD32200 POSCTRL_GAIN （Kv 系数）

通用原则

- 大定位窗口 --> 可以选择相对短的最大定位监控时间
- 小定位窗口 --> 必须选择相对长的最大定位监控时间
- 小的位置控制增益 --> 必须选择相对长的最大定位监控时间
- 大的位置控制增益 --> 必须选择相对短的最大定位监控时间

说明

定位窗口的大小对程序段的切换时间有影响。所选择的范围越小，定位过程则越长，并且在很长时间之后才可以执行下一个指令。

4.2.3 零速监控

功能

结束一个运行程序段后(已到达位置给定值)，将监控在设定的延迟时间（MD36060 STANDSTILL_POS_TOL 零速监控的延迟时间）经过后，进给轴位置与设定位置的距离是否位于 MD36040 STANDSTILL_DELAY_TIME(零速公差)的范围之内。 其他情况下会引发一个报警信号。

有效性

零速监控在“零速监控延迟时间”结束后始终有效，只要没有有效的新的进给命令。

零速监控适用于进给轴和一个位置控制主轴。

生效

监控的响应产生下述结果：

- 发出 25040“零速监控”报警。
- 用快速停止功能（位置控制回路断开）通过转速给定值斜坡停止相应的进给轴/主轴。制动斜坡的持续时间在 MD36610 AX_EMERGENCY_STOP_TIME(出错情况下制动斜坡的持续时间)中定义。
- 如果进给轴/主轴和其它的进给轴/主轴插补，随着跟随误差的减少，这些轴快速停止(局部位置给定值 = 0)。

误差原因/误差消除

- 位置控制增益过大(调节回路振荡) --> 改变控制器增益的机床数据 MD32200 POSCTRL_GAIN(伺服增益系数)
- 零速窗口过小 --> 改变机床数据
  MD36030 STANDSTILL_POS_TOL(零速公差)
- 进给轴由于机械原因偏离自己的位置 --> 排除故障

4.2.4 夹紧监控

功能

如果定位结束后需要将轴夹紧，可以用接口信号“夹紧监控运行”(DB380x.DBX2.3)实现夹紧监控功能。

夹紧监控很必要，因为在夹紧过程中，轴可能被压到离给定点距离超过零速公差的地方。在 MD36050 CLAMP_POS_TOL(用于接口信号“夹紧监控运行”的夹紧公差) 里给出了偏离给定点的距离值。

有效性

通过接口信号“夹紧监控运行”来激活夹紧位置监控功能。在夹紧过程中，夹紧监控取代了零速监控。

夹紧位置监控适用于进给轴和一个位置控制主轴。

生效

如果在夹紧工序中轴离给定点距离超过了夹紧公差，会产生下述结果:

- 发出 26000 “夹紧位置监控”报警。
- 用快速停止功能（位置控制回路断开）通过转速给定值斜坡停止相应的进给轴/主轴。制动斜坡的持续时间在 MD36610 AX_EMERGENCY_STOP_TIME(出错情况下制动斜坡的持续时间)中定义。
- 如果进给轴/主轴和其它的进给轴/主轴插补，随着跟随误差的减少，这些轴快速停止(局部位置给定值 = 0)。

4.2.5 转速给定值监控

功能

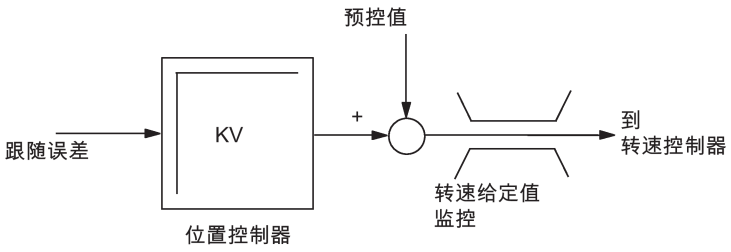
转速给定值监控检查该给定值是否超出了 MD 36210 CTRLOUT_LIMIT(最大转速给定值)中定义的驱动最大允许转速值。如果已超出，则将该值限制为允许的最大值，而且进给轴/主轴停止并产生报警。

对于进给轴，最大转速(百分比)大于 MD32000 MAX_AX_VELO 中定义的最大转速给定值(100%)。这同样也定义了控制余量。

对于模拟主轴，最大输出转速不能大于最大给定输出电压为 10 V(100%)时的转速值。

转速给定值包括位置控制器的转速给定值和前馈控制值(如果前馈控制有效)。

转速给定值监控：



有效性

转速给定值监控始终适用于进给轴和主轴。

生效

如果超过了最大的转速给定值，会产生下述结果:

- 发出 25060“转速给定值极限”报警
- 用快速停止功能（位置控制回路断开）通过转速给定值斜坡停止相应的进给轴/主轴。
  制动斜坡的持续时间在 MD36610 AX_EMERGENCY_STOP_TIME(出错情况下制动斜坡的持续时间)中定义。
- 如果进给轴/主轴和其它的进给轴/主轴插补，随着跟随误差的减少，这些轴快速停止(局部位置给定值 = 0)。

说明

在机床制造商保护级时，可以在 MD36220 CTRLOUT_LIMIT_TIME 中设定延迟时间，延迟时间结束后会产生报警并制动进给轴。该时间的缺省值为零。

转速给定值的初始限值使此伺服回路为非线性。通常，当进给轴在转速给定值范围内停止时，会导致轨迹偏差。因此，必须预留一定的控制余量。

误差原因

- 存在测量回路或驱动故障。
- 设定的设定值过高(加速度、速度、降低系数)
- 工作区域中有障碍物(如接触到工作台)
- 未对模拟主轴正确进行测速发电机调节或存在测量回路误差或驱动器误差。

4.2.6 实际速度监控

功能

此功能监控实际速度是否超过在 MD36200 AX_VELO_LIMIT [n]（速度监控阈值）里给出的极限值。

有效性

如果接口信号“位置测量系统 1”(DB380x.DBX1.5)激活的测量回路产生实际值，即：仍在极限频率下运行，实际速度监控系统就一直生效。

实际速度监控始终适用于进给轴和主轴。

生效

超过“速度监控阈值”时将产生下述结果:

- 发出 25030 “实际速度报警极限”报警
- 用快速停止功能（位置控制回路断开）通过转速给定值斜坡停止相应的进给轴/主轴。 制动斜坡的持续时间在 MD36610 AX_EMERGENCY_STOP_TIME(出错情况下制动斜坡的持续时间)中定义。
- 如果进给轴/主轴和其它的进给轴/主轴插补，随着跟随误差的减少，这些轴快速停止(局部位置给定值 = 0)。

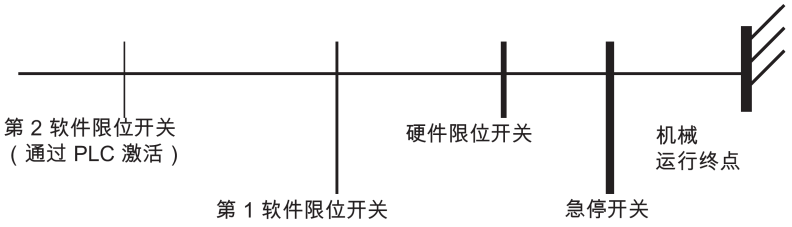
误差消除

- 检查实际值
- 检查位置控制方向
- 检查 MD36200 AX_VELO_LIMIT（速度监控阈值）
- 对于模拟主轴，检查信号给定值电缆

4.3 静态极限监控

4.3.1 限位开关监控

直线轴的限位开关一览：



4.3.2 硬件限位开关

功能

每根轴每个加工方向都有一个硬件（HW）限位开关，以避免滑块脱离滑床。

如果硬件限位开关被逾越，PLC 将通过接口信号“硬件限位开关，正/负”(DB380x.DBX1000.1/.0)通知 NC，并中断所有轴的运动。这种制动方式可以通过在 MD36600 BRAKE_MODE_CHOICE(硬件限位开关的制动方式)中定义来实现。

有效性

在控制器起动后硬件限位开关监控适用于所有的操作方式。

生效

- 在硬件限位开关被超越时，根据方向发出 21614 报警“硬件限位开关 + 或 - ”。
- 根据 MD36600 BRAKE_MODE_CHOICE(硬件限位开关制动模式)的设定，进给轴停止。

- 如果进给轴/主轴和其它进给轴/主轴插补，它们也将根据 MD36600 BRAKE_MODE_CHOICE(硬件限位开关制动模式)中的设定而停止。
- 运行方向的方向键被锁定。

处理

- 复位
- 反向运动（“JOG”方式）
- 程序修改

4.3.3 软件限位开关

功能

它用于控制在正常加工方式下各坐标轴最大的加工范围。

每根轴拥有两对软件限位开关。它们在机床进给轴系统中通过以下机床数据定义：

MD36100 POS_LIMIT_MINUS（负向第 1 软限位开关）

MD36110 POS_LIMIT_PLUS（正向第 1 软限位开关）

MD36120 POS_LIMIT_MINUS2（负向第 2 软限位开关）

MD36130 POS_LIMIT_PLUS2（正向第 2 软限位开关）

有效性

- 回参考点之后，软件（SW）限位开关监控功能在所有操作方式下均有效。
- 轴可以回位到软件限位开关处。
- 第 2 个软件限位开关可通过 PLC 的接口信号“第 2 个软件限位开关正/负”(DB380x.DBX1000.3/.2)激活。设置将立即生效。而第 1 个软件限位开关正/负失效。
- 软件限位开关监控对不停止旋转的回转轴无效，即如果 MD30310 ROT_IS_MODULO = 1。（回转轴和主轴模态转换）

结果/反应

不同的操作方式下会产生不同的反应:

自动，MDA：

- 和软件限位开关有冲突的程序段不能执行。但前一个程序段已正确执行。
- 取消程序执行。
- 输出报警 10720“软件限位开关+或 - ”。

JOG:

- 进给轴停止在软件限位开关位置。
- 输出报警 10621“进给轴已停在软件限位开关 + 或 - ”。
- 运行方向的方向键被锁定。

说明

转换软件限位开关：

软件限位开关转换后，如果当前位置位于新软件限位开关后面，轴将以最大允许的速度来减速如果该轴和其它轴插补，其它轴也将减速。会产生轮廓损坏。

处理

- 复位
- 反向运动（“JOG”方式）
- 程序修改

4.4 边界条件

为了确保监控功能正确运行，必须保证下列机床数据是正确的:

通用数据：

- MD31030 LEADSCREW_PITCH（丝杠螺距）
- 齿轮齿数比（负载变速箱）：
  MD31050 DRIVE_AX_RATIO_DENOM（负载变速箱分母）
  MD31060 DRIVE_AX_RATIO_NUMERA（负载变速箱分子）
  齿轮齿数比（编码器），可用于主轴：
  MD31070 DRIVE_ENC_RATIO_DENOM（测量变速箱分母）
  MD31080 DRIVE_ENC_RATIO_NUMERA（测量变速箱分子）
- MD32810 EQUIV_SPEEDCTRL_TIME
  （用于前馈控制的转速调节回路相关时间常量）
- 编码器分辨率
  MD31020 ENC_RESOL[0]（每转的编码器线数）

仅用于模拟主轴：

- 输出电压/输出转速
  MD32260 RATED_VELO（额定电机转速）
  MD32250 RATED_OUTVAL（额定输出电压）

4.5 数据表

4.5.1 机床数据

序号	名称	名称
进给轴/主轴专用		
30310	ROT_IS_MODULO	回转轴和主轴模态转换
32000	MAX_AX_VELO	最大轴速度
32200	POSCTRL_GAIN[n]	Kv 系数
32250	RATED_OUTVAL	额定输出电压
32260	RATED_VELO	额定电机转速
32300	MAX_AX_ACCEL	轴加速度
32810	EQUIV_SPEEDCTRL_TIME[n]	用于前馈控制的转速调节回路相关时间常量
36000	STOP_LIMIT_COARSE	粗准停
36010	STOP_LIMIT_FINE	精准停
36020	POSITIONING_TIME	精准停延迟时间
36030	STANDSTILL_POS_TOL	停止位置公差
36040	STANDSTILL_DELAY_TIME	停止监控延迟时间
36050	CLAMP_POS_TOL	接口信号“夹紧有效”的夹紧公差
36060	STANDSTILL_VELO_TOL	最大进给轴/主轴停止速度
36100	POS_LIMIT_MINUS	1. 软件限位开关负
36110	POS_LIMIT_PLUS	1. 软件限位开关 正
36120	POS_LIMIT_MINUS2	2. 软件限位开关负
36130	POS_LIMIT_PLUS2	2. 软件限位开关 正
36200	AX_VELO_LIMIT[n]	速度监控极限值
36210	CTRLOUT_LIMIT[n]	最大转速给定值
36300	ENC_FREQ_LIMIT□n□	编码器极限频率
36302	ENC_FREQ_LIMIT_LOW	编码器极限频率重新同步

序号	名称	名称
36310	ENC_ZERO_MONITORING [n]	零标记监控
36400	CONTOUR_TOL	轮廓监控公差带
36500	ENC_CHANGE_TOL	大背隙值/ 实际位置值转换最大公差
36600	BRAKE_MODE_CHOICE	硬件限位开关的制动性能
36610	AX_EMERGENCY_STOP_TIME	出错情况下制动斜坡持续时间
36620	SERVO_DISABLE_DELAY_TIME	伺服无效延迟时间

4.5.2 接口信号

序号	.位	名称
进给轴/主轴专用		
DB380x.DBX1	.5	位置编码器 1
DB380x.DBX2	.3	夹紧运行
DB380x.DBX1000	.0 / .1	硬件限位开关负/硬件限位开关正
DB380x.DBX1000	.2 / .3	2. 软件限位开关负/软件限位开关正
DB390x.DBX0	.2	编码器极限频率超越 1
DB390x.DBX0	.4	回参考点/同步 1

5 连续路径加工，准停方式和预读（B1）

5.1 简要描述

数控系统在执行零件程序时以程序段的形式进行轨迹控制连续加工。 只有在当前程序段执行完毕后，才能执行下一程序段。 不同的加工部件有不同的要求，所要求的程序段转换标准也不同。 对轨迹轴，在程序段转换时有两种处理方式。

第一种“准停”方式意味着在执行下一个程序段之前，所有的轨迹轴必须符合定位精度到达指定目标点。 使用这种方法，就必须在程序段转换时降低轨迹轴速度，这就意味着延长了程序段的转换时间。

使用第二种方式，即“连续路径加工”方式，可以在程序段交界处避免降低轨迹轴速度，尽可能以相同轨迹速度转换到下一程序段。

“预读”是一种方法，它用于连续路径加工方式下对 NC 程序段进行预处理并决定速度控制。

5.2 概述

插补机床轴必须具有相同的动态性能，即在相同的速度下具有相同的跟随误差。

轨迹轴是指所有参与加工工件的进给轴,每个进给轴的运行均由进行轨迹插补计算的插补器控制，从而满足以下要求：

- 所有相关轴同时起动
- 轴相互间以正确的速度关系运行
- 所有相关轴均同时到达编程的目标点

各个轴的加速度会因为轨迹的不同而不同，如圆周轨迹。

轨迹轴可以是几何轴或者辅助轴（例如：参与加工工件的工件旋转轴）。

零循环程序段速度

如果一个程序段的路径长度小于一个插补循环（时间）下编程的进给量，则称之为零循环程序段。出于精度要求，速度将被降到足够小，以便在这段路径中至少需要一个插补循环。 从而使该速度等于或小于程序段路径长度与插补（IPO）循环的比率。

因同步而停止

不管选择的是准停方式还是连续路径加工方式，均可能因为要求同步而推迟程序段转换，从而造成轨迹轴停止。在准停方式下，轨迹轴将停在当前所执行的程序段的结束处。在连续路径加工方式下，轨迹轴将停在最近一个程序段的结束处，该程序段是指轨迹轴在不超出加速度极限值的情况下可以制动到停止的程序段。因为同步而须暂停的情况如下：

- 需要 PLC 进行响应时
  如果在运行结束前/后发出一个辅助功能，需要 PLC 进行响应，轨迹轴将在程序段结束处停止。
- 在后续程序段未出现的情况下
  若未及时准备好后续程序段（比如："外部加工"时），则将在最后一个可启动的程序转换处停止。
- 在清空缓冲存储器的情况下
  如果 NC 零件程序要求预处理和主运行同步(如通过指令 STOPRE 清空缓冲存储器)，则间接导致程序段减速或准停。

为同步而停止不造成任何轮廓误差。但是在连续路径加工方式下，特别不希望出现停止，因为会引起凹凸切削。

5.3 准停

在准确停方式下（G60， G9）轨迹轴将运行到程序段结束处。只有当所有轨迹轴都达到了准确停等级，才可转换程序段。程序段转换时的速度实际为零。

这表明：

- 轨迹轴在程序段结束处没有过冲时可以达到停止状态。
- 由于等待准确停止到位而使加工时间延长。
- 由于等待准确停，可能会产生凹凸切削。

准确停功能适合于沿轮廓精确加工。

在下述情况下准确停方式无意义：

- 为了加快加工速度，允许偏离某一标准(如精准停)的精度范围。
- 要求绝对恒定速度。

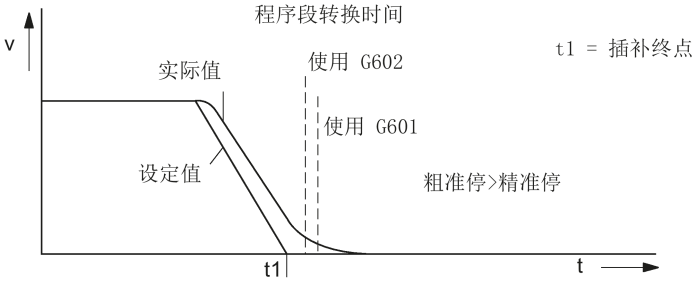
激活准确停功能

在数控系统零件程序中准确停功能可以通过指令 G60 或 G9 选择。G60 为模态方式有效，G9 为非模态。G9 用于中断连续路径加工。两种准确停功能都只能在所给定的准确停精度等级范围内(G601, G602)有效。准确停功能可以通过选择连续路径加工功能 G64 取消。

准停标准

- 精准停：G601
  监控轴实际到达的位置离给定点的距离是否在一定范围内。范围的大小在 MD36010 STOP_LIMIT_FINE（精准停）中给定。
- 粗准停：G602
  功能同精准停一样，其监控窗口大小在 MD36000 STOP_LIMIT_COARSE(粗准停)中给出。为了使程序段转换速度快于精准停，粗准停精度范围参数应相应设置较大。

与精度等级相关的程序段转换：



插补终点

当插补器已经计算出轴的给定值速度在一个插补时钟脉冲内接近为零，则表明到达插补器终点。但是，轨迹轴的实际位置还未到达目标位置(跟随误差)。

和连续路径加工方式和/或准停功能的有效准停等级无关，程序段中包含的辅助功能随同插补器终点一起传输到 PLC，只要这些功能在动作结束后输出。

5.4 连续路径运行

5.4.1 概述

在连续路径加工中，在程序段结束并进行程序段转换时，加工速度不必为了达到定位精度而降低到很小， 从而可以在程序段转换点处避免轨迹轴停止加工，尽可能以相同的速度转到下一个程序段。 为了实现此目标，应激活连续路径加工方式（G64）以及预读功能。

通过连续路径加工方式可以实现：

- 轮廓拐角倒圆。
- 没有为达到准停精度等级而必需的制动和加速过程， 从而缩短了加工时间。
- 平缓的速度变化产生良好的切削状态。

如果需要快速加工轮廓，使用连续路径加工方式是很有效的。

在下列情形下不建议使用连续路径加工方式：

- 轮廓要求准确加工。
- 要求绝对恒定速度。

隐含准停功能

在某些情形下，在连续路径加工方式中需要执行准停功能，以便进行以下操作。 这时轨迹速度被制动为零。

- 辅助功能指令如果在进给轴运行前给出，则前一个程序段只有在到达所选择的准停精度等级后才结束。
- 辅助功能指令如果在进给轴运行后给出，则在程序段插补结束后输出辅助功能。
- 如果所执行的程序段中不包含任何轨迹轴的进给信息，当到达所选择的准停等级时，前一个程序段结束。
- 如果在下一个程序段中执行加速特性的转换 BRISK/SOFT，则当前程序段在插补器终点结束。
- 如果在零件程序中编程“清空缓冲存储器”(STOPRE)功能，则前一个程序段只有在到达所选择的准停精度等级后才结束。

在连续路径加工方式下速度 = 0

在下列情形下，在程序段结束时加工速度被制动为零，这与上述的隐含准停功能无关：

- 在 SPOS 功能中主轴定位所需的时间大于轨迹轴运行时间。 在此情况下，只有在定位主轴“精准停”后才进行程序段的转换。
- 要求同步时（参见章节“概述 (页 32)”）。

运行时的辅助功能输出

如果由于带辅助功能输出的程序段中编程的路径长度和速度，导致运行时间太短，则降低程序段轨迹速度以便可以在一个 PLC 循环中响应辅助功能。

如果在一个 PLC 循环时间内未响应，则不能执行下一个预处理的程序段，并且进给轴立即停止(不考虑加速度极限)。

在长的程序段中，如果在程序段结束时未提供速度响应，并且无需因 PLC 响应时间而降低速度，速度一直保留到程序段结束，然后根据以上介绍降低速度。

如果在制动过程中给出响应，进给轴不制动。

5.4.2 按过载系数降低速度

功能

此功能可以在连续路径方式下降低进给率，以便非正切**程序段转换**可以在一个插补时钟脉冲中超限，并保持加速度极限和考虑过载系数。如果程序段终点的轮廓特性不是正切，速度降低会导致进给轴相关速度的突然改变。速度跃变避免将进给率降低到零。如果当轴速度降低到某一速度值并从该速度值进行加速，从而可以到达新的速度给定值，则发生速度跃变。

可以使用过载系数来控制速度跃变值。由于速度的变化量是进给轴相关的，在程序段转换时应考虑当前有效轴的速度跃变值。对于几乎是正切的程序段转换，如果未超过允许的进给轴加速度，不降低轨迹速度。这可以对轮廓进行很小的超限加工。

过载系数

过载系数限制机床轴在程序段转换时的速度跃变。为了确保速度跃变不超出轴负载，跃变值通过轴的加速度计算得到。过载系数定义了机床轴的加速度在一个 IPO 时钟脉冲内可以超出极限的数量，在 MD32300 MAX_AX_ACCEL（轴加速度）定义了此加速度。

速度进给是以下值的结果：轴加速度 * (过载系数-1) * 插补节拍。过载系数为 1.2。

系数 1.0 表示按不停止速度只能进行正切转换。对于任何其它的转换，速度将根据给定值设定将为零。

选择和取消速度降低

在每个 NC 零件程序中可以使用编程代码 G64（BRISK 有效，非 SOFT），根据过载系数模态选择连续路径加工方式下的速度降低。

连续路径加工方式 G64 可以通过：

- 选择准停 G9 而被中断（程序段方式）。
- 选择准停 G60 而被取消。

5.4.3 降低轨迹速度，以限制过冲

引言

轨迹上过冲限制是控制连续路径加工方式的另一种方法。“按过载系数降低速度”限制速度的变化，而“路径过冲限制”则限制加速度的跃变。

连续加工方式的轮廓加工（如圆弧 - 直线过渡）中，在**程序段转换**时，加速度突然变化。

过冲降低

在具有不同种特性曲线的轨迹的程序段转换时，通过降低轨迹速度来降低过冲。这样可以使轮廓段之间的过渡比较平滑。

过冲限制

使用 MD32432 OATG_TRANS_JERK_LIM(程序段转换时轨迹轴的最大轴特定过冲)定义最大过冲值，这是轨迹轴在程序段转换时允许的最大过冲值。

激活

如果编程了连续路径方式 G64 和 SOFT 加速度特性，轮廓转换时的过冲限制生效。 MD32432 PATH_TRANS_JERK_LIM 必须包含正值。

5.4.4 机床轴专用的过冲限制

功能

轴专用机床数据 MD32431 MAX_AX_JERK[..]可以单独设定每个进给轴的加速度变化，而使用机床数据 MD32300 MAX_AX_ACCEL[..]可以定义加速度极限。

MD32431 MAX_AX_JERK[..]适用于轨迹插补轴以及在**一个程序段**中编程了 **SOFT**（无过冲的加速度变化）功能。

通常，在程序段中的轴加速度曲线和程序段转换时加速度曲线之间有所不同。

优点

使用轴专用机床数据具有以下优点：

- 轴的动态特性直接由插补考虑且可以完全被利用。
- 轴的过冲限制不仅可以编入直线程序段中，也可以编入曲线轮廓中。

“过冲限制”的详细说明请参见章节“加速度 (页 39)”。

5.5 压缩功能

功能

COMPON, COMPCURV

COMPON 和 COMPCURV 压缩功能可将至多十个如下格式的连续线性程序段生成一个多项式程序段："G01 X...Y...Z...F..."。压缩功能的多项式程序段具备如下特性：

- COMPON：连续速度程序段转换
- COMPCURV：连续速度和加速度程序段转换

COMPCAD

COMPCAD 压缩功能理论上可将任意数目的线性和圆弧程序段生成一个多项式程序段。这些多项式程序段在程序段转换时具有恒定速度和加速度。同理来识别并考虑理想的角度。

可通过所有压缩功能的机床数据来指定通向已编程点的计算所得路径的最大允许偏差。与 COMPON 和 COMPCURV 相比，在 COMPCAD 情况下，不在相邻路径的不同方向上使用所指定偏差。实际上，COMPCAD 试图在相似条件下实现与已编程点的相似偏差。

压缩功能的共同目的是，通过实现连续程序段转换以及增加各程序段的路径长度来优化表面质量并提升加工速度。

COMPCAD 在很大程度上是 CPU 时间和内存密集型的。建议仅在通过 CAD/CAM 程序中的方法不能成功优化表面质量的情况下使用 COMPCAD。

概述

- 可按需要实现待压缩程序段中的位置数据，例如 X100、X=AC(100)、X=R1*(R2+R3)。
- 而后，压缩操作因任一其他指令而在待压缩程序段当中以及两个待压缩程序段之间中断。

可用性

在数控系统中，仅在铣削版上支持 NC 程序段压缩功能。

参数设置

在对 NC 程序段压缩功能进行参数设置时，必须设置下列机床数据和设定数据：

通道专用机床数据

编号	名称 $MC_	含义
MD20170	COMPRESS_BLOCK_PATH_LIMIT	待压缩 NC 程序段的最大运行长度
MD20172	COMPRESS_VELO_TOL	压缩过程中路径进给的最大允许偏差
MD20482	COMPRESSOR_MODE	设置压缩器的运行方式

通道专用设定数据

编号	名称$SC_	含义
SD42470	CRIT_SPLINE_ANGLE	COMPCAD 的拐角极限角度
SD42475	COMPRESS_CONTUR_TOL	压缩过程中的最大允许轮廓偏差

说明

拐角极限角度和压缩功能 COMPCAD

通过设定数据 SD42470 $SC_CRIT_SPLINE_ANGLE 设置的 COMPCAD 拐角极限角度仅可对拐角进行近似识别。通过合理性评估，压缩器亦可将较为平坦的程序段转换识别为拐角，并将较大角度识别为外露层。

轴向机床数据

编号	名称$MA_	含义
MD33100	COMPRESS_POS_TOL	压缩过程中的最大允许路径偏差

再生机床数据建议设置

在使用压缩功能时，建议如下设置压缩功能再生机床数据：

编号	名称	建议值
MD18360	$MN_MM_EXT_PROG_BUFFER_SIZE （用于外部执行的 FIFO 缓存大小）	100
MD20490	$MC_IGNORE_OVL_FACTOR_FOR_ADIS （与过载因子无关的 G641/G642）	1
MD28520	$MC_MM_MAX_AXISPOLY_PER_BLOCK （每个程序段轴多项式的最大数目）	3
MD28530	$MC_MM_PATH_VELO_SEGMENTS （用于限制路径速度的存储元件数目）	5
MD28540	$MC_MM_ARCLENGTH_SEGMENTS （用于显示弧长功能的存储元件数目）	10
MD28060	$MC_MM_IPO_BUFFER_SIZE （用于程序段预处理的 NC 程序段数目）	100
MD28070	$MC_MM_NUM_BLOCKS_IN_PREP （用于程序段预处理的程序段数目）	60
MD32310	$MA_MAX_ACCEL_OVL_FACTOR （轴向速度不连续变化的过载因子）	<G64 运行的值>

编程

激活

使用模态 G 指令 COMPON、COMPCURV 或 COMPCAD 来激活压缩功能。

若要进一步优化表面质量，可使用功能 G642（倒圆功能）和 SOFT（过冲限制）。必须在程序开始一并写入这些指令。

示例：

程序代码	注释
PROC ...	
N10 COMPCAD SOFT G642	; 激活 COMPCAD 压缩器
N20 G01 X...Y...Z...F...	; 运行程序段 1 ... n
...	
N1000 COMPOF	; 取消激活压缩器
N1010 RET	

取消激活

使用 COMPOF 指令取消激活所有压缩功能。

文献

关于压缩功能编程的描述，请参考：

SINUMERIK 808D ADVANCED 编程和操作手册（铣削）

5.6 预读

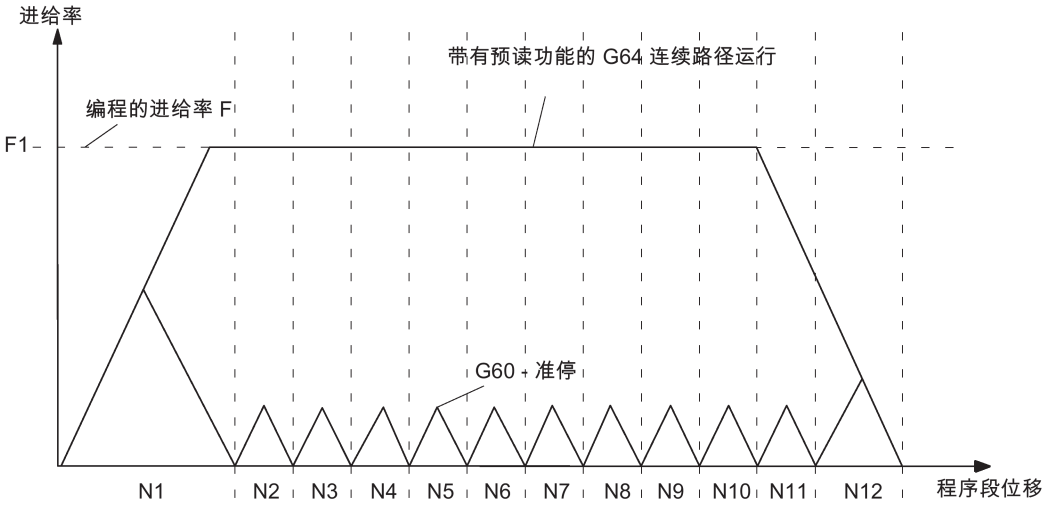
功能

“预读”是一种方法，它用于连续路径加工方式（G64）下对当前程序段之外的若干 NC 程序段进行预处理并决定速度控制。

不使用预读：如果程序段中只编程了较小的路径值，则每个程序段到达一个速度值，该速度值可以让进给轴在程序段终点制动并遵守加速度极限。这说明根本没有到达编程的速度值，尽管出现很多具有接近正切路径转换的预处理程序段。

使用预读：可以在几个接近正切转换的程序段中进行加速和减速过程，这样对于较小路径可以获得更大的进给率。此时，根据速度极限值进行预制动，从而可以避免加速度极限和速度极限的冲突。

G60 和 G64 速度特性比较，短行程程序段：



预读考虑了可以预见的速度极限：

- 程序段中的速度极限
- 程序段中的加速度极限
- 程序段转换时的速度极限
- 程序段转换时同步
- 程序段终点处的准停

操作方法

预读功能只用于轨迹轴，不适用于主轴。

首先每个预处理程序段的终点速度视为零，因为下一个程序段或某个准停程序段可能很小，而且进给轴将在程序段终点处停止。在一些具有高速度给定值和很短的位移值的程序段中，可以根据当前预测的速度值增加各程序段中的速度值，这样可以到达所需的给定值速度。然后降低速度，并且允许最后预处理的程序段终点速度为零。在这种情况下，你会获得锯齿型的速度值，但这可以在预读程序段中通过降低给定值（固定值）避免。

速度特性

处理可以预见的速度极限，预读功能还包括编程的速度。这可以使当前程序段以外的程序段获得较小的速度。

其余程序段速度

速度特性包括对其余程序段速度的检测。根据当前程序段和其余的 NC 程序段的信息，计算出速度特性，并为当前超出值减去所需的速度修调量。速度特性的最大值由最大轨迹速度限制。

此功能通过考虑修调值，可以在当前的程序段中降低速度，这样可以在下一个程序段开始时就到达较小的速度值。如果降低速度需要比当前程序段中的进给更长的时间，则在下一个程序段中继续降低速度。速度控制功能只用于其余的程序段。

选择和取消预读功能

预读功能可以通过选择连续路径加工方式 G64 来激活，通过 G60/G9 取消。

5.7 数据表

5.7.1 机床数据

序号	名称	名称
一般		
18360	MM_EXT_PROG_BUFFER_SIZE	用于外部执行的 FIFO 缓存大小（DRAM）
通道专用		

序号	名称	名称
20170	COMPRESS_BLOCK_PATH_LIMIT	待压缩 NC 程序段的最大运行长度
20172	COMPRESS_VELO_TOL	压缩过程中路径进给的最大允许偏差
20482	COMPRESSOR_MODE	压缩器模式
20490	IGNORE_OVL_FACTOR_FOR_ADIS	与过载因子无关的 G641/G642
28060	MM_IPO_BUFFER_SIZE	IPO 缓存中的 NC 程序段数目（DRAM）
28070	MM_NUM_BLOCKS_IN_PREP	程序段预处理过程中的 NC 程序段数目（DRAM）
28520	MM_MAX_AXISPOLY_PER_BLOCK	每个程序段轴多项式的最大数目
28530	MM_PATH_VELO_SEGMENTS	用于限制程序段中路径速度的存储元件数目
28540	MM_ARCLENGTH_SEGMENTS	用于每个程序段弧长功能显示的存储元件数目
29000	LOOKAH_NUM_CHECKED_BLOCKS	预读程序段数量
进给轴/主轴专用		
32310	MAX_ACCEL_OVL_FACTOR	轴向速度不连续变化的过载因子
32431	MAX_AX_JERK	轨迹运行中的最大轴特定过冲
32432	PATH_TRANS_JERK_LIM	程序段转换时的最大轴特定过冲
33100	COMPRESS_POS_TOL	补偿过程中的最大偏差
36000	STOP_LIMIT_COARSE	粗准停
36010	STOP_LIMIT_FINE	精准停
36020	POSITIONING_TIME	精准停延迟时间

5.7.2 设定数据

序号	名称	名称
一般		
42470	CRIT_SPLINE_ANGLE	拐角极限角度，压缩器
42475	COMPRESS_CONTUR_TOL	压缩过程中的最大轮廓偏差

5.7.3 接口信号

序号	位	名称
通道专用		
DB3300.DBX0004	.3	所有轴停动
进给轴/主轴专用		
DB390x.DBX0000	.6	采用粗准停到达位置
DB390x.DBX0000	.7	采用精准停到达位置

6 加速度（B2）

6.1 加速度属性

跃变式加速度

在普遍使用的 v/t 进给轴直线速度控制中，加速度在该时间段内呈跃变式变化。 由于这种不连续的、跃变式的加速度，无法让轴做到平滑启动和制动，但是可以实现时间优化的速度/时间特性。

加速度，带过冲限制

过冲是指一段时间内加速度的变化。 对于过冲限制的加速度，最大加速度以斜坡方式预设，而不是跃变式。 为了获得更平滑的加速度特性，运行时间将增加，而行程，速度和加速度保持不变。 在一些情况下，这种时间上的损失可以通过给轴设定较高的加速度来补偿。

它具有以下优点：

- 保护机床的机械结构
- 减少机床的高频震动并可以很好地控制

6.2 插补级过冲限制

选择和取消过冲限制加速度

可以使用 MD32431 MAX_AX_JERK（轨迹运行中的最大轴特定过冲）限制各个机床进给轴的加速度变化。 只要 SOFT(平滑)有效，该功能对插补轴有效。 过冲限制加速度只能在插补级上执行。

通过以下方式激活过冲限制的加速度：

在零件程序中编程 **SOFT**。 SOFT 模态有效并取消使用跃变式加速度(BRISK)。 如果在一个程序段中为轨迹轴编程了 SOFT，则前一个程序段以准停结束。

通过以下方式取消过冲限制的加速度(SOFT)：

在零件程序中编程 **BRISK**。 BRISK 模态有效。 如果在一个程序段中为轨迹轴编程了 BRISK，则前一个程序段以准停结束。使用 BRISK 激活 v/t 直线速度控制的跃变式加速度。

有效范围

轨迹相关的过冲限制适用于在“AUTO”和“MDA”运行方式下的插补轴。 加速度属性 SOFT 和 BRISK 可以和运行模式“准停”G9，G60、连续路径模式 G64 和预读一起使用。 该加速度也适用于“空转进给”功能。 触发快速停止的报警会取消该加速度的使能。

有关在连续路径模式下，以及尤其在程序段转变时的速度、加速度和过冲的特性，请参考章节“连续路径运行、准停和预读(B1)”。

说明

我们建议将各个轴的下列机床数据设置为相同的值： MD32431 MAX_AX_JERK 和 MD32432 PATH_TRANS_JERK_LIM（程序段转换时轨迹运行中的最大轴特定过冲）。

6.3 JOG 方式下的过冲限制

“JOG” 方式下，进给轴进行以下操作时过冲限制有效：

- 手动运行
- 手轮运行
- 重新定位

在回参考点时和产生快速停止的报警时过冲限制无效。

可以预设各个轴的过冲限制。加速度特性符合轨迹相关的过冲限制的属性 SOFT。在相应的运行方式下不能取消选择进给轴的限制。

可以在 MD32420 JOG_AND_POS_JERK_ENABLE 中设定需要限制过冲的进给轴。MD32430 JOG_AND_MAX_JERK 中定义了允许的最大轴特定过冲。

6.4 数据表

序号	名称	名称
轴专用		
32300	MAX_AX_ACCEL	轴加速度
32420	JOG_AND_POS_JERK_ENABLE	使能轴特定的过冲限制
32430	JOG_AND_POS_MAX_JERK	轴特定的过冲
32431	MAX_AX_JERK	轨迹运行中最大的轴特定过冲
32432	PATH_TRANS_JERK_LIM	程序段转换中轨迹运行时最大的轴特定过冲

7 龙门轴（G1）（仅用于 PPU16x.3）

7.1 概述

说明

在无有效许可证的情况下激活选件时，会出现报警 8081 “使用了未经许可的 %1 选项”。此时无法正常操作机床。

关于“选件设置”的相关操作，请参见“SINUMERIK 808D ADVANCED 的许可证 (页 395)”章节。

龙门轴

龙门轴是以机械方式联结的机床轴。由于各轴间通过机械方式联结，所以他们总是共同运行。并通过“龙门轴”功能实现运行控制。

实施引导运行的机床轴称为引导轴。与引导轴同步运行的机床轴为同步轴。引导轴和同步轴一起构成了一个龙门轴联结。

会持续监控引导轴和同步轴的实时位置差。若同步轴与引导轴之间的实时位置偏差数值过大，则数控系统会自动停止龙联结的所有运行轴（以避免损坏机床）。

应用

对于带有龙门的大型龙门铣床，为了通过龙门，还必须配备两个进给驱动装置（即：每面含有一个配备位置测量系统的驱动装置）。由于两个驱动装置间通过机械方式强制联结，所有必须对其进行完全同步控制，以避免弯曲机械部件。

配置

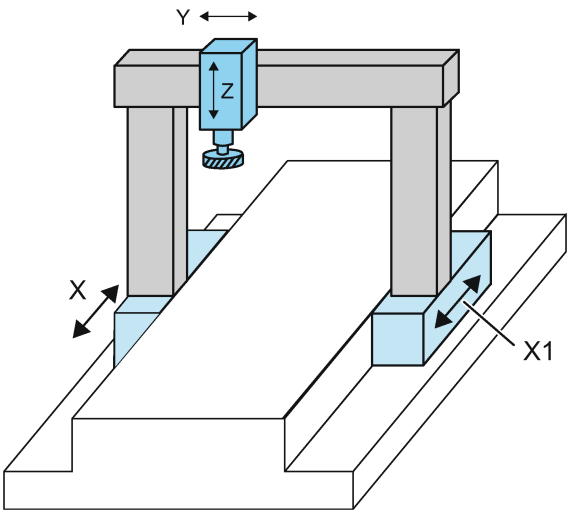
可定义一个龙门轴联结（由一个引导轴和同步轴组成）。

7.2 “龙门轴”功能

应用

对于大型龙门铣床，其不同的机床轴单位（比如龙门或横梁）分别由各个独立的相应驱动装置驱动。每个驱动装置都拥有自己的测量系统并以此构成一个完整的机床轴系统。在这些相互间通过机械方式紧密联结在一起的机床轴的运行过程中，必须**完全同步**控制两个驱动装置，以避免弯曲机械部件（会导致力/力矩传递）。

示例：带一个龙门联结（X 和 X1）的大型龙门铣床



“龙门轴”功能负责引导和监控这些彼此间紧密联结的机床轴。

术语和定义

前文的功能描述中频繁使用了下列术语和概念：

龙门轴

龙门轴由一个龙门轴对、一个引导轴和一个同步轴组成。各轴间以机械方式强制耦合，因此总是由 NC 同时启动。运行期间，会持续监控各轴的实时位置。每个龙门联结的轴或者全是线性轴，或者全是回转轴。

龙门轴联结

一共可定义**一个**龙门联结。一个龙门联结由**一个**引导轴**一个**同步轴组成。

机床数据通过龙门轴联结确定哪个引导轴影响哪个同步轴。引导轴和同步轴分开后，不能运行。

引导轴

引导轴是从操作者和编程者的视角出发存在且可对其如普通 NC 轴一样施加影响的龙门轴。一个龙门联结的所有轴均以引导轴的名称进行标识。

同步轴

同步轴为龙门轴，其应有位置始终由运行的引导轴引导并与之同步。从操作员和编程者的视角观察，同步轴是“不存在的”。

龙门联结的条件

- 其中不能有主轴。
- 不能对同步轴进行转换。
- 同步轴不能为另一轴联结的从动轴。
- 不能将同步轴定义为另一轴联结的引导轴。

说明

设置时，应保证龙门轴联结的每个轴均能随时充当引导轴的角色，即应与速度设置、加速设置和动力设置一致。

数控系统会检查轴定义的合理性。

功能单元

“龙门轴”功能可分为如下几个功能单元：

- 创建同步轴的设定值
- 监控实时数值差
- 引导轴和同步轴的回参考点和同步

创建同步轴的设定值

对于操作人员来说，所有已耦合的龙门轴之运行应该像 NC 中只有一个轴（即引导轴）一样。与之相类似，在零件程序中也仅对引导轴进行编程。因此操作者、PLC 接口或零件程序的指令和运行要求对龙门联结的所有轴均有效。

在所有运行方式中，“龙门轴”功能将直接从引导轴的设定值中创建同步轴的设定值并使其同步运行。

说明

引导轴和同步轴的调节动力设置必须相同。

监控实时数值差

会通过插补脉冲持续比较引导轴和同步轴的实时位置值并检查其是否位于容许的范围之内。

可利用机床数据设置与轴相关的报警和运行关闭的下列极限数值：

龙门报警极限值

若实时位置数值差超过龙门报警极限值，则会出现警告“超过报警极限值”：
MD37110 GANTRY_POS_TOL_WARNING（龙门报警极限值）

除此之外，还会向 PLC 输出如下接口信号：
DB390x.DBX5005.3（超过龙门报警极限值）

当数值低于极限值后，信息和接口信号自动重新消失。

可通过 MD37110 = 0 设置阻止消息。

龙门关闭极限值

当实时位置数值偏离超过机床容许的最大值时，会出现警告 10653“超过故障极限值”：
MD37120 GANTRY_POS_TOL_ERROR（龙门关闭极限值）

为了避免损坏机床设备，制动斜坡会立即关闭龙门轴。

当龙门联结已同步时，可使用 MD37120 中的数值。

必须通过“RESET”功能消除此报警信息。

除此之外，还会向 PLC 输出如下接口信号：
DB390x.DBX5005.2（超过龙门关闭极限值）

若龙门轴联结仍未同步，则会从机床数据中得出龙门关闭极限值：
MD37130 GANTRY_POS_TOL_REF（回参考点时的龙门关闭极限值）



在龙门联结卡住时，也会发出接口信号“超过龙门关闭极限值”（无控制器使能，龙门联结处于“停止”状态）。

龙门联结处于“跟随”状态时，不会对其实施监控。

扩展监控

可通过以下机床数据激活扩展监控：
MD37150 GANTRY_FUNCTION_MASK（龙门功能）

龙门轴的回参考点和同步

从“大型龙门铣床”这一例子中可以看出：无论在何种运行方式下，机床通电后必须对龙门轴进行强制耦合。若引导轴或同步轴安装了增量测量系统，则机器启动后耦合轴必须首先运行至参考点。

当龙门联结的所有轴都运行至参考点后，将对各轴可能出现的倾斜进行平衡和调整（也就是所谓的龙门同步过程）。以上程序结束后，会向 PLC 发出如下接口信号：DB390x.DBX5005.5（龙门联结已同步）

关于龙门轴的回参考点和同步过程在“龙门轴的回参考点和同步 (页 44)”一节中有详细描述。

闭环控制

已耦合龙门轴的调节动力设置必须相同（参见“调试龙门轴 (页 48)”）。如此可保证在正常运行模式下，引导轴和同步轴位置即使在加速期间和制动期间内的位置始终同步。

因此可通过引导轴或同步轴各自的驱动装置解决负载所带来的影响。

故障响应

出现故障时，比如移除控制器使能后导致一个龙门轴停止（例如：急停），则已耦合的其它龙门轴也相应停止。

取消强制耦合

某些情况下（比如某个龙门轴的参考点由于分配故障而丢失）需要在回参考点前补偿或缩小龙门轴的倾斜位置。为此，必须在**无耦合的前提下手动**移动引导轴或同步轴。

可通过如下机床数据设置和随后进行的复位拆分龙门轴的强制耦合：
MD37140 GANTRY_BREAK_UP = 1（拆分龙门联结）

如此设置后，便可手动运行单个龙门轴。此时，报警监控和关闭极限监控无效。

注意
龙门轴的机械耦合 若龙门轴在此种运行状态下仍通过机械方式联结在一起，则运行引导轴或同步轴时会损坏机床！

7.3 龙门轴的回参考点和同步

7.3.1 引言

启动时的倾斜位置

机床启动时，引导轴和同步轴可能会偏离其理想位置（比如龙门的位置倾斜）。通常情况下，此种偏离相对较小，因此各龙门轴仍可以回参考点。

特殊情况下（比如龙门轴由于故障、断电或急停停止时）必须在运行前检查偏移量是否在容许的范围之内，必要时需进行补偿性移动。

此种情况下，必须使用以下机床数据拆分龙门轴联结：
MD37140 GANTRY_BREAK_UP（拆分龙门轴联结）

龙门同步过程

数控系统打开后，所有的龙门轴必须首先回参考点，然后同步。在龙门同步过程中，所有龙门轴**在没有耦合的情况下运行至龙门联结的参考点**。

在龙门轴回参考点时，龙门联结的参考点就是**引导轴的参考点：**
MD34100 REFP_SET_POS（间距编码系统中的参考点数值/目标点）

否则，参考点即是**引导轴的当前实时位置**。

龙门轴的回参考点和同步程序根据特殊的过程表自动进行。

回参考点过程

装有增量测量系统的龙门轴同步过程按下列步骤完成：

第 1 步：

引导轴回参考点

在激活 REF 机床功能的情况下，由 PLC 用户程序发出的引导轴接口信号启动龙门轴特有的轴方式回参考点：
DB380x.DBX0004.7/.6（运行键加/减）

引导轴运行至参考点（此过程与回参考点运行相对应）。

所属同步轴同步运行。引导轴显示接口信号“已回参考点/同步”时，表明已到达参考点。

第 2 步：

同步轴回参考点

一旦引导轴运行至其参考点，则同步轴**自动**回参考点（与回参考点运行相对应）

在此期间，由数控系统内部调节引导轴和同步轴之间的关联性，保证同步轴与引导轴的同步。同步轴显示接口信号“已回参考点/同步”时，表明已到达参考点。此时，之前的龙门轴间的相关性仍然有效。

第 3 步：

龙门同步过程

当龙门联结的所有轴均已回参考点时，必须将其同步至已确定的参考位置。首先会对各同步轴的实时位置与已确定的引导轴的参考位置进行比较。

根据所得的引导轴与同步轴之间的实时位置数值差的不同，将其分为如下几种情况：

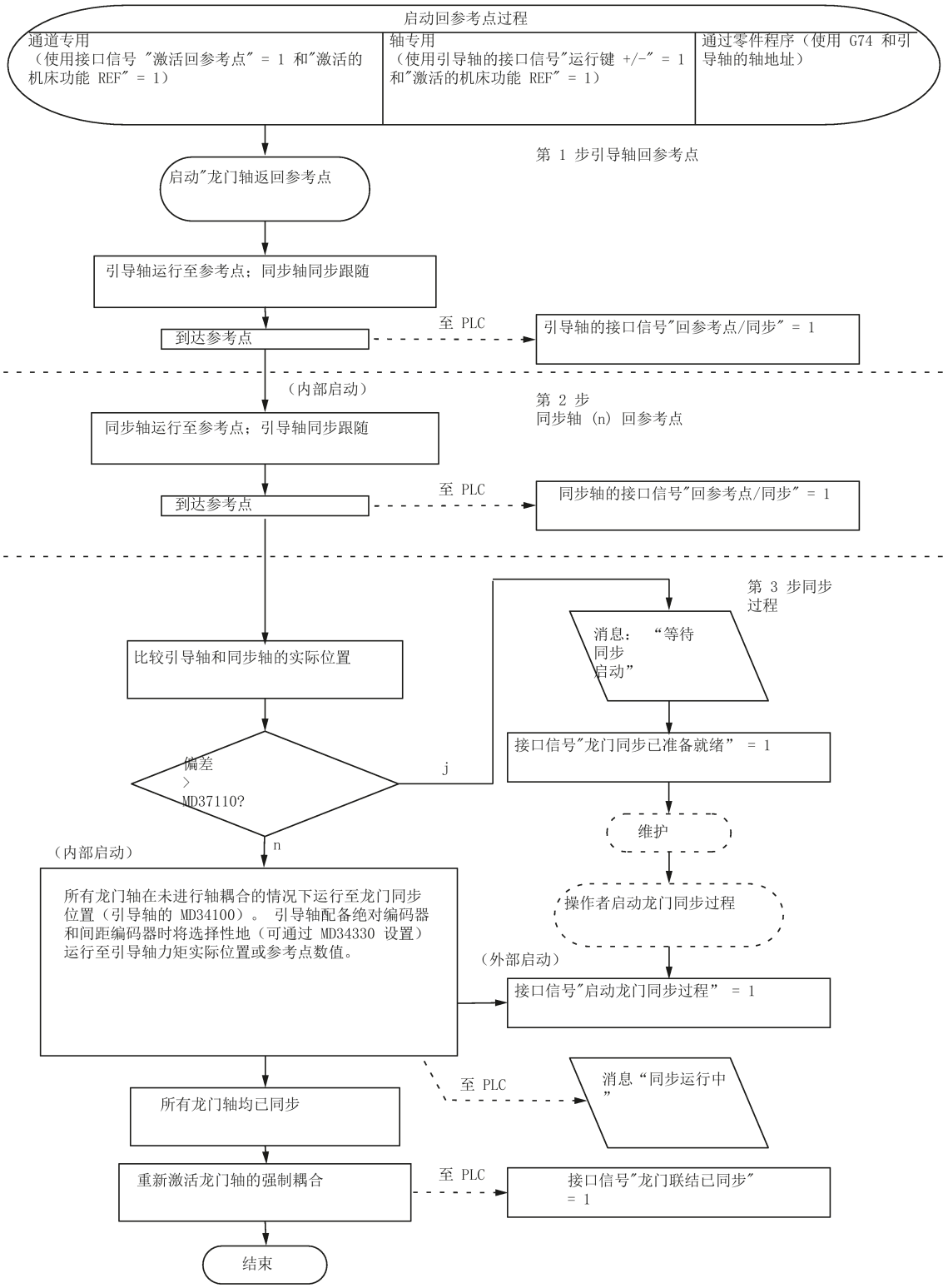
- 差值**小于**龙门报警极限值：
  若未设置接口信号“自动锁定同步”，则**自动**启动龙门同步过程。此时，会显示消息“龙门联结 X 同步中”。
  在**未耦合**的情况下，所有龙门轴均以机床数据中设置的速度运行至特定的位置数值。
  MD34040 REFP_VELO_SEARCH_MARKER（关闭速度）
  在引导轴上确定位置数值：
  MD34100 REFP_SET_POS（增量系统中的参考点）
  当引导轴上有绝对编码器和间距编程编码器时，将选择性地运行至引导轴的当前实时位置或参考点数值，可通过以下机床数据进行设置：
  MD34330 REFP_STOP_AT_ABS_MARKER（无目标点的间距编码长度测量系统）
  此时所使用的速度与回参考点驶入速度相对应。
  MD34070 REFP_VELO_SEARCH_CAM（回参考点驶入速度）

一旦所有的龙门轴达到其目标位置（理想位置），接口信号“龙门联结已同步”将设置为 1 信号且龙门轴耦合再次激活。此时，龙门联结所有轴的实时位置数值必须相同。至此，龙门同步过程结束。

- 同步轴与引导轴间的实时位置数值差**大于**龙门报警极限值时：

  接口信号“龙门同步过程启动准备”将设置为 1 信号并给出消息“等待龙门联结 X 启动同步”。此种情况下，龙门同步过程不会自动启动，而需由操作者或 PLC 用户程序启动。引导轴通过接口信号“启动龙门同步过程”启动同步过程。此信号由引导轴发出。同步过程与上述过程类似。

后面的过程图中有回参考点过程和同步过程的图形显示。



同步过程

以下情形下必须进行同步：

- 龙门联结所属各轴已运行至参考点
- 同步丢失（如下）。

过程中断

若上述回参考点过程由于故障或复位中断，则应如下操作：

- **在步骤 1 或 2 中中断：**
  重新将引导轴运行至参考点（参见第 1 步）
- **第 3 步中断：**
  只要龙门轴仍然处于回参考点状态（接口信号“已回参考点/同步” = 1），则可通过接口信号“同步龙门联结”重新启动龙门同步过程。

重新启动龙门同步过程

只能在下列条件下通过接口信号“启动龙门同步过程”启动龙门轴同步过程：

- 必须激活“REF.POINT”机床功能。必须设置好下列接口信号：
  DB3100.DBX0001.2（激活机床功能 REF）
- DB390x.DBX5005.5 = 0（龙门联结已同步）
- 联结的各轴须位于公差窗口内：
  DB390x.DBX5005.4 = 1（“已做好龙门同步过程启动准备”）
- NC 通道中没有任何轴回参考点
  DB3300.DBX0001.0 = 0（回参考点功能激活）

若不是通过接口信号“启动龙门同步过程”从**回参考点过程中**启动龙门同步过程，则同步轴的目标位置不是 MD34100 中给出的参考位置，而是**引导轴的当前实时位置**，且同步轴应在未进行轴耦合的情况下运行至此位置。

说明

可通过以下接口信号锁定引导轴的自动同步：
DB380x.DBX5005.5（锁定自动同步）

在轴还没有获得使能的情况下，此操作总是有效的。此种情况下，也会由接口信号针对性地启动同步过程：
DB380x.DBX5005.4 = 1（启动龙门同步）

同步丢失

下列情况下，会丢失龙门联结同步（DB390x.DBB5005 = 0）：

- 龙门轴处于“跟随”状态。
- 龙门轴的参考位置丢失。比如在“停车”时（未激活任何测量系统）。
- 某个龙门轴重新回参考点（DB390x.DBB0000 = 0）。
- 已拆分龙门联结（MD37140 = 1）。

若在运行过程中由于故障原因导致龙门联结同步丢失，可直接通过接口信号“启动龙门同步”重新开始龙门同步过程（前提：DB390x.DBB0000 = 1 适用于龙门联结的所有轴）。此种情况下，同步轴未耦合运行至引导轴的当前实时位置。

若在龙门联结运行过程中执行了急停，然后取消，且两个轴彼此的运行距离均小于从动轴的停止位置公差，则龙门联结会自动同步。其将不在进入 BA REF 过程。

参考点选择

为了保证龙门轴回参考点时的运行路径最短，则引导轴和同步轴的参考点数值应该相同（MD34100）。

应通过机床数据考虑各轴的零点标记与参考点之间的距离偏差：

- MD34080 REFP_MOVE_DIST（参考点间距）
- MD34090 REFP_MOVE_DIST_CORR（参考点偏移/绝对偏移）

选择回参考点方向

可通过以下机床数据确定从动轴的零点标记查找方向：
MD37150 GANTRY_FUNCTION_MASK 位 1

位	值	含义
1	0	从动轴零点标记查找方向与 MD34010 REFP_CAM_DIR_IS_MINUS 类似
	1	从动轴零点标记查找方向与引导轴相同

在回参考点过程的同步补偿运行中，引导轴的参考点数值将作为龙门联结各轴的目标位置，并在未进行轴耦合的情况下运行至此位置。当引导轴上有绝对编码器和间距编程编码器时，将选择性地运行至引导轴的当前实时位置或参考点数值，可通过以下机床数据进行设置：
MD34330 REFP_STOP_AT_ABS_MARKER
（无目标点的间距编码长度测量系统）

若引导轴和同步轴只使用了一个回参考点挡块，则必须在 PLC 用户程序中考虑到此点。

7.3.2 自动同步

可在以下模式中运行自动同步：

- “REF.POINT”模式中（参见“引言 (页 44)”一节）
- 通过如下所述的方式。

将龙门联结切换至跟随模式时，对引导轴和同步轴时实位置的监控则处于未激活状态。因此此时联结不再同步。引导轴的接口信号“龙门联结处于同步状态”将设置为“0”，此设置与各轴的位置状态无关：

若将龙门联结从跟随状态切换至位置控制状态，只要引导轴和同步轴之间实时位置监控所得差值小于 MD36030 中确定的数值，则自动恢复同步过程。

此种情况下，将不会进行插补，而为同步轴确定一个新的设定值。之前确定的差值将通过位置控制器进行修正。此过程中，只有同步轴在移动。

同步轴的此种移动过程与其从“停止”状态中切换至位置控制模式后的运行类似。此种情况下，只要停止监控没有在其间触发报警 25040（报警后会进行跟随操作），则位置控制器会对停止前所给出的位置进行重新设置。

这种自动同步模式下会使用与停止监控相同的公差窗口：
MD36030 STANDSTILL_POS_TOL（停止位置公差）

通过以下机床数据根据参数组适用:
MD36012 STOP_LIMIT_FACTOR（粗准停/精准停和停机系数）

说明

除“REF.POINT”模式外，以下接口信号可在各种运行方式中锁定自动同步：
DB380x.DBX5005.5（锁定自动同步）

将 DB380x.DBX5005.5 设置为“0”即可激活自动同步。

随之应将龙门联结的一个轴从跟随模式切换至位置控制运行模式。可通过接口信号实现此操作：

DB380x.DBX0001.4 = 1（跟随运行模式）

DB380x.DBX0002.1 = 1（控制器使能）

7.3.3 特点

通道特有方式回参考点

龙门轴也可使用通道特有方式回参考点：
DB3200.DBX0001.0（激活回参考点功能）

对于通过通道特有方式回参考点的轴顺序将使用引导轴轴向机床数据的数值：
MD34110 REFP_CYCLE_NR（通道特有回参考点的轴顺序）

如上所述，在到达引导轴的参考点后，同步轴将首先回参考点。

利用指令 G74 通过零件程序回参考点

也可使用指令 G74 通过零件程序启动龙门轴的回参考点和同步过程。此方法中只对引导轴的名称进行编程。其过程与通过轴特有方式回参考点类似。

带间距编码参考标记的位置测量系统

为了在回参考点运行中节省路径，只有在龙门轴上使用带间距编码参考标记的位置测量系统。如此可保证测量系统根据较短的运行路径（比如 20 mm）回参考点。龙门轴回参考点的方式与普通增量测量系统中的方式相同。

绝对编码器

在同步补偿运行中，龙门轴联结的各轴在未进行轴耦合的情况下运行至引导轴的参考点数值，可通过以下机床数据确定：
MD34100 REFP_SET_POS（间距编码系统中的参考点数值/目标点）

当引导轴上有绝对编码器和间距编程编码器时，将选择性地运行至引导轴的当前实时位置或参考点数值，可通过以下机床数据进行设置：
MD34330 REFP_STOP_AT_ABS_MARKER
（无目标点的间距编码长度测量系统）

激活轴补偿

引导轴和同步轴上均可激活补偿功能。龙门轴各轴的补偿值分开独立计算。因此，在调试时须确定引导轴和同步轴的补偿值并输入。

只有在轴已回参考点或龙门联结已同步时，数控系统内部的补偿功能才有效。这里：

补偿类型	有效时间	PLC 接口信号
间隙补偿	轴已回参考点	“已回参考点/同步”
LEC	轴已回参考点	“已回参考点/同步”
垂度补偿	龙门联结已同步	“龙门联结已同步”

若同步轴的运行是由于激活了补偿功能，则其会显示一个运行指令，此指令与引导轴无关。

监控有效

龙门轴与普通 NC 轴一样，只有在到达参考点（IS“已回参考点/同步”）后下列监控才有效：

- 工作区域限制
- 软限位开关
- 保护区

同步轴的轴向机床数据值也可作为监控极限值使用。

7.4 调试龙门轴

概述

由于通常情况下龙门轴的引导轴和同步轴之间存在强制耦合，故在调试龙门轴联结时必须像仅操作一个轴单位一样进行。因此，必须共同确定并输入引导轴和同步轴的轴向机床数据。

若同步轴由于动力减小而超载，则会出现报警 10656。

以下将叙述龙门轴调试中**需特别注意的地方**。

轴运行方向

进行龙门轴调试时，必须监控电机的转动方向是否与所需的轴运行方向一致。可通过轴向机床数据进行修正：

MD32100 AX_MOTION_DIR（运行方向）

激活轴联结

MD37100 GANTRY_AXIS_TYPE[a,b]（龙门轴定义）

本机床数据确定龙门轴的下列事项：

- 其所属的龙门联结 (1)、
- 其作为引导轴还是同步轴。

下表给出了 MD37100 中可能的数值：

a	b	龙门轴	龙门联结
-	0	无	-
0	1	引导轴	1
1	1	同步轴	1

调试时应注意：一个龙门联结的各轴只能全部定义为线性轴或全部定义为回转轴：
MD30300 IS_ROT_AX（回转轴/主轴）

输入龙门关闭极限值

为了监控同步轴与引导轴之间的实时位置偏差数值，必须根据机床生产商的相关说明输入引导轴和同步轴的关闭极限数值：

MD37120 GANTRY_POS_TOL_ERROR（龙门关闭极限值）

MD37130 GANTRY_POS_TOL_REF（回参考点时的龙门关闭极限值）

说明

然后关闭数控系统再重新启动，因为龙门轴定义和龙门关闭极限数值必须在重启 (POWER ON) 后才生效。

引导操作和故障回应

龙门轴完全同步运行的前提是：引导轴和同步轴调节回路的**动力参数必须设置一致**。

必须分别**优化**各轴向调节回路（位置、转速和电流控制器），以尽快且有效地排除故障。额定值回路中的**动力匹配**功能可在不减损调节质量的前提下匹配不同类型的轴动力。

必须轴向优化引导轴和同步轴的下列调节参数设置：

- MD32200 POSCTRL_GAIN（KV 系数）
- MD32620 FFW_MODE（前馈控制参数）
- MD32610 VELO_FFW_WEIGHT（速度/转速的前馈控制系数）
- MD32810 EQUIV_SPEEDCTRL_TIME（用于前馈控制的转速控制回路时间常量）

引导轴和同步轴的下列调节参数的设置必须相同：

- MD32400 AX_JERK_ENABLE（轴向过冲限制）
- MD32410 AX_JERK_TIME （用于轴向震动滤波器的时间常量）
- MD32420 JOG_AND_POS_JERK_ENABLE（轴向过冲限制的基础设置）
- MD32430 $MA_JOG_AND_POS_MAX_JERK（轴向急动）

动力匹配

为了实现引导操作，引导轴和已耦合同步轴的动力必须相同。动力相同的意思是：相同转速下其滞后量必须相同。

额定值回路中的动力匹配功能可有效平衡不同动力轴（调节电路）之间的引导操作。动力匹配的时间常量指的是输入与其它相应轴相比动力“最弱”轴的额外时间常量的差值。

示例

在转速前馈控制已激活的前提下，动力基本上由“最慢”转速调节回路的额外时间常量确定。

引导轴：
MD32810 EQUIV_SPEEDCTRL_TIME [n] = 5 ms

同步轴：
MD32810 EQUIV_SPEEDCTRL_TIME [n] = 3 ms

同步轴动力匹配的时间常量：
MD32910 DYN_MATCH_TIME [n] = 5 ms - 3 ms = 2 ms
（动力匹配的时间常量）

必须使用机床数据轴向激活动力匹配功能：
MD32900 DYN_MATCH_ENABLE（动力匹配）

说明

动力匹配监控

若需进行微调，须稍微平衡 K_V 系数或前馈参数，以达到最佳的效果。

龙门轴回参考点

首先，引导轴和同步轴的参考点位置设置应大致相同。

为了使龙门轴同步补偿运行不会自动启动，调试时须在其回到参考点前将龙门报警极限值设置为 0：
MD37100 GANTRY_POS_TOL_WARNING（龙门轴定义）

如此设置后，报警消息在运行过程中无效。

若引导轴和同步轴的瞬时倾斜位置对驱动有过大的附加力矩影响，则必须在运行前较准龙门联结。然后再根据章节“龙门轴回参考点和同步”中的相应说明使龙门轴回到参考点。

引导轴和同步轴均回到参考点后，必须通过在如下操作后进入的屏幕上比较实时位置数值得出两者之间的差值：



且需考虑参考点偏移。

MD34080 REFP_MOVE_DIST（参考点间距）

MD34090 REFP_MOVE_DIST_CORR（参考点偏移/绝对偏移）

龙门轴同步

可通过接口信号“启动龙门同步过程”激活龙门轴同步（参见章节“龙门轴的回参考点和同步 (页 44)”）。同步过程结束后（接口信号“龙门联结已同步”= 1）必须检查引导轴和同步轴之间的偏移量是否都是 0。必要时，使用上述的机床数据进行修正。

输入龙门报警极限值

将引导轴和同步轴的参考点数值进行优化设置后，为了保证龙门轴同步后相互间无倾斜，必须在以下机床数据中为引导轴和同步轴输入报警极限值：

MD37110 GANTRY_POS_TOL_WARNING（龙门报警极限值）

此时，应逐步提高其数值，保证刚好无消息（已超过极限值）出现。当然，首先必须监控加速阶段。

还可通过此极限数值确定在达到何种程度的位置偏差后内部数控系统自动启动龙门同步过程。

得出补偿并激活

若龙门轴需要补偿（垂度补偿或丝杠螺距误差补偿），则必须确定引导轴**和**同步轴的修正数值并将其输入相应的参数或表格中。

请参见“补偿（K3）(页 106)”章节。

函数发生器/测量函数

在同步轴上使用启动工具激活函数发生器和测量函数时，会出现错误消息而中止。

若必须在同步轴上激活此功能（比如为了提高机床性能），则必须事先交换引导轴和同步轴。

说明

通常情况下，在虚拟轴上启动函数发生器、测量函数和 AM 时会出现错误消息而中止。

特殊情况

若需要激活**单个**轴的此功能，则必须事先拆分龙门联结。由于拆分后其它的轴将不再同步运行，所以必须保证已激活轴不再作为位置公差运行。

也就是说，在拆分龙门联结后：

- 始终激活运行范围极限值并将其设置为较小的值（位置公差）
- 尽可能地同步龙门联结，然后执行通电复位操作，**避免**重新回参考点。如此操作后，运行范围极限值总是指的相同位置（通电时的位置）

- 避免阶跃函数。只允许小于公差的位置阶跃存在。
- 与普通轴相反，对于函数发生器和测量函数中的补偿值我们推荐始终设置为 0。
- 函数发生器和测量函数的振幅应较小，保证已激活轴的运行距离少于位置公差。为了实施监控，必须一直激活运行范围极限值（参见上面所述）。

说明

此处多仅以普通方式叙述调试观点和驱动调节动力观点。为了补充此处的不足，我们在“示例 (页 53)”一章中通过具体例子-以机床数据讲述-准备了一整套完整的观点。

龙门联结的调试帮助

函数发生器和测量函数这两个调试功能通过 PI 服务编程。在“JOG”运行模式下，通过 MCP 面板上的如下按键运行已编程的所有轴：



在操作面板“龙门联结的测量函数和函数发生器”中会显示一幅图象，可在其中输入两个振幅数值、一个补偿值和一个带宽值。第一个振幅数值适用于测量轴，另一个数值适用于其它的耦合轴。

7.5 龙门轴中的 PLC 接口信号

龙门轴的特有接口信号

已耦合龙门轴的特有 PLC 接口信号通过引导轴或同步轴的轴向 PLC 接口管理。下表中为全部的龙门特有 PLC 接口信号以及其是否会被引导轴或同步轴使用的标识。

PLC 接口信号	PLC ↔ NCK	地址	引导轴	同步轴
启动龙门同步过程	→	DB380x.DBX5005.4	x	
锁定自动同步	→	DB380x.DBX5005.5	x	
龙门轴	←	DB390x.DBX5005.7	1	1
龙门引导轴	←	DB390x.DBX5005.6	1	0
龙门联结已同步	←	DB390x.DBX5005.5	x	
龙门同步过程已准备就绪	←	DB390x.DBX5005.4	x	
超过龙门报警极限值	←	DB390x.DBX5005.3		x
超过龙门关闭极限值	←	DB390x.DBX5005.2		x

x：与 ... 相关

龙门轴轴向接口信号的作用

a) 从 PLC 至轴的轴向接口信号（PLC → NCK）

一般说来，从 PLC 至轴的轴向接口信号均涉及龙门联结的所有轴。此时，龙门各轴（引导轴和同步轴）处于平等位置。

比如：若将引导轴的如下接口信号设置为“0”，则可同时停止龙门联结的所有轴：

DB380x.DBX0002.1（控制器使能）

下表中为各单个接口信号（从 PLC 至轴）对龙门轴的作用：

PLC 接口信号	地址	作用于	
		引导轴	同步轴
进给轴/主轴锁定	DB380x.DBX0001.3	龙门联结的所有轴	无效
位置编码器 1	DB380x.DBX0001.4	轴向	轴向
调节器使能	DB380x.DBX0002.1	龙门联结的所有轴	
删除剩余行程（轴向）	DB380x.DBX0002.2	轴向	无效
夹紧运行	DB380x.DBX0002.3	轴向	轴向
进给停止	DB380x.DBX0004.4	龙门联结的所有轴	
硬限位开关负/正方向	DB380x.DBX1000.0 /.1	轴向报警：龙门联结各轴的停止命令	
2.硬限位开关负/正方向	DB380x.DBX1000.2 /.3	轴向	轴向
驱动参数组选择	DB380x.DBX4001.0 - .2	轴向	轴向
脉冲使能	DB380x.DBX4001.7	轴向	轴向

在龙门轴中，要么其“跟随”状态（某个龙门轴的接口信号 = 1），要么其“停止”状态（龙门各轴的接口信号 = 0）的激活取决于如下接口信号：

DB380x.DBX0001.4（跟随运行模式）

b) 从轴至 PLC 的轴向接口信号（NCK → PLC）

一般说来，从轴至 PLC 的轴向接口信号由引导轴和同步轴分别设置并发送至 PLC。

例外：
引导轴运行时，也可在同步轴上设置接口信号：
DB390x.DBX0004.6/.7（运行指令负方向/正方向）

7.6 龙门轴的其它事项

手动运行

在“JOG” 运行模式下，不能直接手动运行同步轴。按同步轴的运行键时，会被内部数控系统忽略。同样，旋转同步轴的手动轮时也无效。

手轮倍率

各轴已耦合时，只能在引导轴上通过手轮实现倍率运行。此时，同步轴随同引导轴一起运行。

零件程序中的编程

在零件程序中，只能对龙门轴联结的引导轴进行编程。对同步轴进行编程时，会出现报警，即使龙门轴联结已拆分时仍如此。

PRESET（预设定）

只能在引导轴上使用预设定功能。使用预设定功能后，内部数控系统会对龙门联结的各轴进行重新评估。因此，龙门轴会丢失其参考和同步：

DB390x.DBX5005.5 = 0（龙门联结已同步）

预设定中的默认设置

龙门联结已激活时，会忽略同步轴的下列机床数据参数化操作。

MD30450 IS_CONCURRENT_POS_AX = 1（复位时的默认设置：中性轴/通道轴）

默认为引导轴的状态。用户可通过显示报警 4300 获悉无意义的设计。

位置显示

实时位置显示中会显示引导轴和同步轴的实时位置。系统数据操作区域中的服务显示数值也会进行位置显示。

软限位开关

软限位开关仅适用于引导轴。超过时，龙门联结的所有轴均会停止。

7.7 示例

7.7.1 创建龙门联结

引言

下例中描述了所需的单个步骤：

- 创建一个龙门联结
- 各轴返回参考点
- 出现偏差后的较准
- 同步各参加轴

位置分布

机床轴 1 = 龙门引导轴，增量测量系统

机床轴 3 = 龙门同步轴，增量测量系统

下列机床数据所述为初始值。 以后可根据下列设置修正或补充单个数值。

龙门机床数据

轴 1：

MD37100 GANTRY_AXIS_TYPE = 1

MD37110 GANTRY_POS_TOL_WARNING = 0

MD37120 GANTRY_POS_TOL_ERROR = 1 mm

MD37130 GANTRY_POS_TOL_REF = 100 mm（最大倾斜位置）

MD37140 GANTRY_BREAK_UP = 0

轴 3：

MD37100 GANTRY_AXIS_TYPE = 11

MD37110 GANTRY_POS_TOL_WARNING = 0

MD37120 GANTRY_POS_TOL_ERROR = 1mm

MD37130 GANTRY_POS_TOL_REF = 100mm（最大倾斜位置）

MD37140 GANTRY_BREAK_UP = 0

参考点机床数据

所给出的机床数据数值分别适用于轴 1 和轴 3 的第一个编码器。

MD34000 REFP_CAM_IS_ACTIVE = TRUE

MD34010 REFP_CAM_DIR_IS_MINUS = 例如 FALSE

MD34020 REFP_VELO_SEARCH_CAM =

MD34030 REFP_MAX_CAM_DIST = 与最大运行距离相对应

MD34040 REFP_VELO_SEARCH_MARKER =

MD34050 REFP_SEARCH_MARKER_REVERSE = 例如 FALSE

MD34060 REFP_MAX_MARKER_DIST = 凸轮工作面和零标记之间的差值

MD34070 REFP_VELO_POS =

MD34080 REFP_MOVE_DIST = 0

MD34090 REFP_MOVE_DIST_CORR = 0

MD34092 REFP_CAM_SHIFT = 0

MD34100 REFP_SET_POS = 0

MD34200 ENC_REFP_MODE = 1

7.7.2 NCK-PLC 接口设置

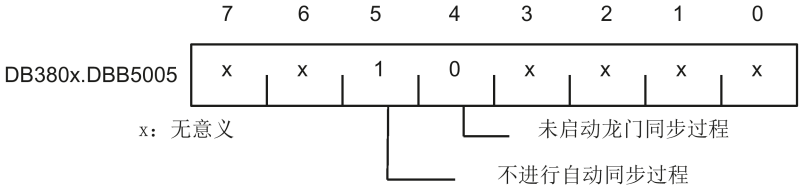
引言

在返回参考点的过程中必须首先阻止自动同步过程，以避免损坏未较准的联结。

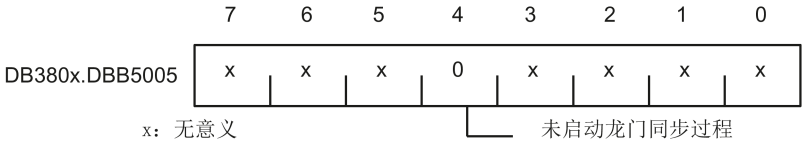
防止自动同步过程

用户 PLC 程序发出下列接口信号：

- 对于引导轴（轴 1）：
  与引导轴相关的 NCK-PLC 接口 DB380x.DBB5005：

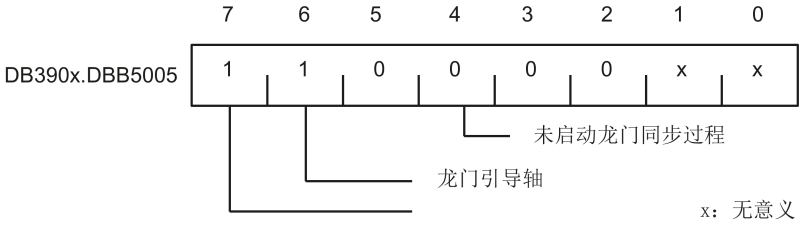


- 对于同步轴（轴 3）：
  与同步轴相关的 NCK-PLC 接口 DB380x.DBB5005：

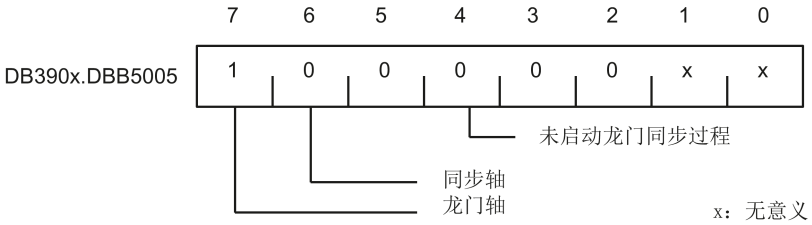


NCK 发出下列接口信号作为确认：

- 对于引导轴（轴 1）：
  与引导轴相关的 NCK-PLC 接口 DB390x.DBB5005：



- 对于同步轴（轴 3）：
  与同步轴相关的 NCK-PLC 接口 DB390x.DBB5005：



7.7.3 开始调试

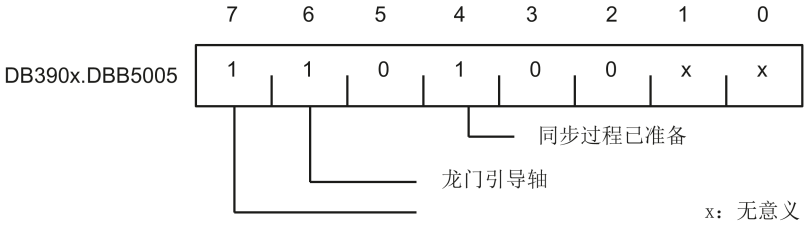
回参考点

应执行下列步骤：

- 选择“REF.POINT”运行方式
- 启动引导轴（轴 1）返回参考点
- 一直等到出现消息“10654 通道 1 等待启动同步过程”。

与此同时，NCK 已将引导轴做好了同步准备。

NCK-PLC 接口 DB390x.DBB5005：引导轴已做好同步准备：



除此之外，还应执行如下操作：

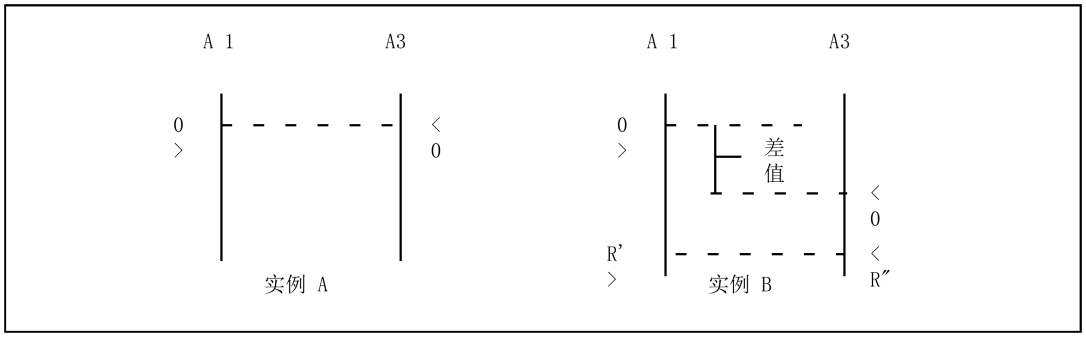
- 复位
- 读出机床坐标系中的数值：
  例如：
  X = 0.941
  Y = 0.000
  XF = 0.000
- 以相反的符号将引导轴（轴 1）的 X 数值填入同步轴（轴 3）的机床数据中：
  MD34090 REFP_MOVE_DIST_CORR = - 0.941

说明

上电后，此机床数据即生效。为了避免需事先通电，也可将此数值输入如下机床数据中：
MD34080 REFP_MOVE_DIST（参考点间距）
此机床数据在复位后即生效。

- 轴 1 根据已修正的机床数据重新返回参考点
- 一直等到出现消息“10654 通道 1 等待启动同步过程”
- 与此同时，NCK 已做好轴 1 的同步准备并发出前图中所示的接口信号。
- 注意机床的实时位置。此时，可能出现 A 或 B 两种情况：

引导轴回参考点后可能出现的结果：



若为情况 A，则可立即启动同步过程。参见步骤“启动同步过程”。若为情况 B，则必须得出偏移“差值”并对其予以考虑：

- 测量差值
- 通过机床底座（图片右侧）两个互为直角的参考点 R 和 R" 可得知 JOG 运行模式中的位置差。然后可读出偏移量，作为位置显示的差值。必须将此偏移差值填入轴 3（同步轴）的如下机床数据中：
  MD34100 REFP_SET_POS
  继续第 1 步（如上所述）。
- 启动同步过程。PLC 发出：
  DB380x.DBX5005.4 = 1（启动龙门同步）

7.7.4 设置报警极限值和故障极限值

一旦设置好龙门联结且已同步后，则必须设置下列机床数据：

MD37110 GANTRY_POS_TOL_WARNING（龙门报警极限值）

MD37120 GANTRY_POS_TOL_ERROR（龙门关闭极限值）

步骤

- 先将各轴的如下机床数据设置为较大数值：
  MD37120 GANTRY_POS_TOL_ERROR（龙门关闭极限值）
- 在如下机床数据中填入一个较小的数值：
  MD37110 GANTRY_POS_TOL_WARNING（龙门报警极限值）
  若此时各轴动力负载较大，则应该会不断发出报警“10652 通道 %1 轴 %2 超过龙门报警极限值”，随之又自动消失。
- 现在增大 MD37110 的数值。
  继续重复以上操作，直到不出现报警消息为止。 接口会显示下面所述的状态。（应显示在有利于生产的窗口中。）
- 在机床数据 MD37120 中填入为报警极限值所得的数值 + 一个较小的安全撞击数值。

故障极限数值

下列机床数据中已填有数值：

MD37110 GANTRY_POS_TOL_WARNING（龙门报警极限值）

MD37120 GANTRY_POS_TOL_ERROR（龙门关闭极限值）

MD37130 GANTRY_POS_TOL_REF（回参考点时的龙门关闭极限值）

匹配过程结束后，这些数值应符合如下大小比例关系：

MD37110 < MD37120 < MD37130

说明

在调试各轴由**线性电动机**和所属的测量系统驱动的龙门联结时，应根据其要求运行。

若接口信号“龙门已同步” 未处于排除等候处理状态，则机床数据 MD37110 和 MD37120 中输入的故障极限值可相互作为引导轴和同步轴之间实时数值差的额外公差（比如报警消除后未重新同步龙门联结）。

7.8 数据表

7.8.1 机床数据

序号	标识符	名称
轴专用		
30300	IS_ROT_AX	回转轴
32200	POSCTRL_GAIN[0]...[5]	K_V 系数
32400	AX_JERK_ENABLE	轴向过冲限制
32410	AX_JERK_TIME	用于轴向震动滤波器的时间常量
32420	JOG_AND_POS_JERK_ENABLE	轴向过冲限制基础设置
32430	JOG_AND_POS_MAX_JERK	轴向过冲
32610	VELO_FFW_WEIGHT[0]...[5]	转速前馈控制系数
32620	FFW_MODE	前馈控制类型
32810	EQUIV_SPEEDCTRL_TIME[0]...[5]	用于前馈控制的转速调节回路相关时间常量
32900	DYN_MATCH_ENABLE	动力匹配
32910	DYN_MATCH_TIME[0]...[5]	动力匹配时间常量
34040	REFP_VELO_SEARCH_MARKER[0]	关闭速度
34070	REFP_VELO_POS	回参考点运行速度
34080	REFP_MOVE_DIST[0]	参考点间距

序号	标识符	名称
34090	REFP_MOVE_DIST_CORR[0]	参考点偏移
34100	REFP_SET_POS[0]...[3]	参考点值
34110	REFP_CYCLE_NR	通道专有回参考点时的轴顺序
34330	REFP_STOP_AT_ABS_MARKER[0]	无目标点的间距编码的长度测量系统
36012	STOP_LIMIT_FACTOR[0]...[5]	粗准停/精准停和停机系数
36030	STANDSTILL_POS_TOL	停止位置公差
37100	GANTRY_AXIS_TYPE	龙门轴定义
37110	GANTRY_POS_TOL_WARNING	龙门报警极限值
37120	GANTRY_POS_TOL_ERROR	龙门关闭极限值
37130	GANTRY_POS_TOL_REF	返回参考点时的龙门关闭极限值
37140	GANTRY_BREAK_UP	拆分龙门轴联结

7.8.2 接口信号

序号	位	名称	引导轴	同步轴
运行模式专有				
DB3100.DBX0001	.2	有效机床功能 REF	-	-
通道专用				
DB3300.DBX0001	.0	回参考点有效	-	-
轴专用				
DB380x.DBX5005	.4	启动龙门同步过程	x	-
DB380x.DBX5005	.5	锁定自动同步	x	-
DB390x.DBX0000	.4	回参考点/同步 1	-	-
DB390x.DBX5005	.2	超过龙门关闭极限值	-	x
DB390x.DBX5005	.3	超过龙门报警极限值	-	x
DB390x.DBX5005	.4	龙门同步过程已准备就绪	x	-
DB390x.DBX5005	.5	龙门联结已同步	x	-
DB390x.DBX5005	.6	龙门引导轴	1	0
DB390x.DBX5005	.7	龙门轴	1	1

x：与 ... 相关

8 轴耦合（M3）

8.1 联动

8.1.1 简介

8.1.1.1 功能

使用“联动”功能，可以在引导轴与联动轴之间建立基于耦合系数的简单轴关联。

联动支持以下特性：

- 任意 NC 轴均可定义为引导轴。
- 任意 NC 轴均可定义为基于指定耦合系数的联动轴。
- 引导轴与联动轴共同形成联动轴组。
- 可为一个引导轴分配任意数量的联动轴。
- 所有联动轴会基于耦合系数跟随引导轴同步运行。
- 在轴耦合生效时，联动轴可脱离引导轴做独立运动。
- 使用零件程序中的编程指令，可定义联动轴组中的引导轴与联动轴，并可打开/关闭轴耦合。
- 亦可在诸如手动、回参考点及手动增量运行等手动模式下实现“联动”功能。

8.1.1.2 前提条件

支持“联动”功能作为 NC 软件的必备组件。

该功能适用以下限制：

- 最多可以同时激活四个联动轴组，联动轴组的最大数目在 MD18450 及 MD18452 中设置。
- 联动轴组可由线性轴与回转轴任意组合或仅由主轴组合。
- 仅可为每个联动轴分配一个引导轴。
- 一联动轴组中的联动轴不可再用作另一联动轴组中的引导轴。

8.1.2 一般功能

可通过“联动”功能定义简单的轴耦合。可在一个引导轴与一或多个联动轴之间实现该轴耦合。可为每个联动轴单独指定耦合系数。

联动轴组

引导轴和所有分配给该轴的联动轴共同形成联动轴组。当引导轴运行时，所有联动轴基于其各自的耦合系数跟随引导轴运行。

联动轴组可由线性轴与回转轴任意组合或仅由主轴组合。

引导轴

包括模拟轴在内的任意 NC 轴均可用作引导轴。

联动轴

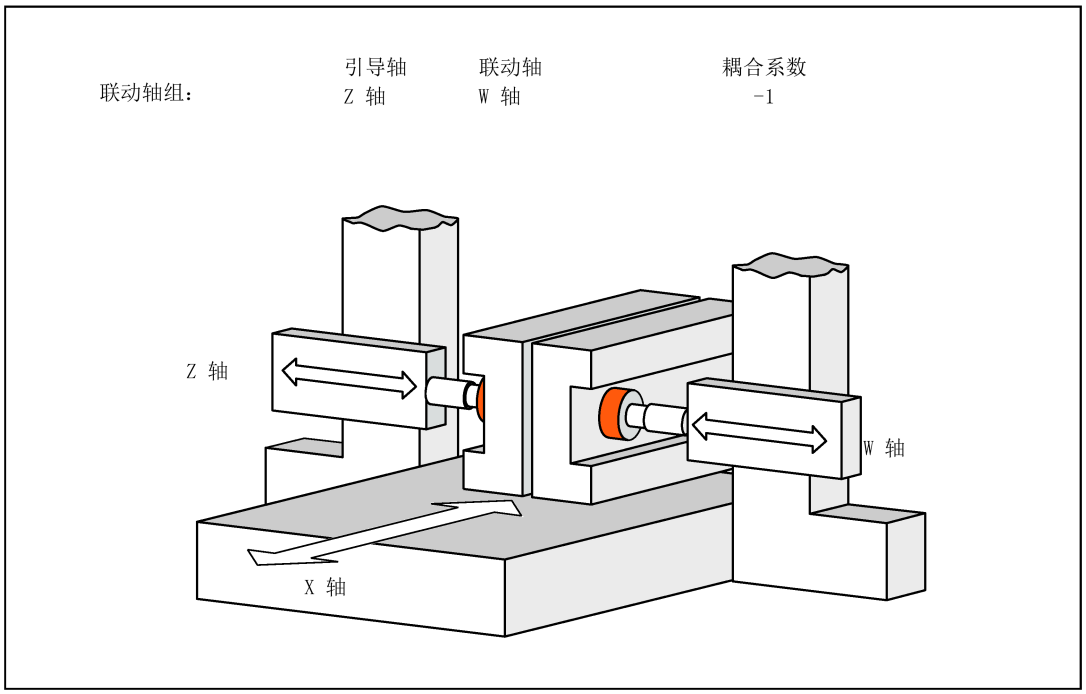
任意 NC 轴均可用作联动轴。

耦合系数

耦合系数即为联动轴与引导轴之间指定的相对运动比例。

耦合系数 K = 联动轴运动 / 引导轴运动

该耦合系数可为负数（联动轴与引导轴的运动方向相反）。



相依联动轴

“相依联动轴”为跟随引导轴运行而运行的联动轴。

独立联动轴

“独立联动轴”为按照直接运行指令运行的联动轴。联动轴的运行结果为其作为“相依”联动轴与“独立”联动轴运行的总和。

坐标系

联动轴在机床坐标系（MCS）或由 MCS 经过动态转换形成的坐标系中运行。

激活/取消

如下所示，仅可通过零件程序来激活/取消联动：

激活：零件程序 → 取消：零件程序

操作模式

联动在“AUTO”、“MDA”及“JOG”模式下有效。

回参考点

在对联动轴组中的轴进行回参考点操作时，请注意：

- 引导轴
  在对联动轴组中的引导轴进行回参考点操作时，引导轴与所有联动轴的耦合继续保持。所有联动轴基于其耦合系数跟随引导轴做同步运动。
- 联动轴：“REF.POINT”模式
  在对联动轴组中的联动轴进行回参考点操作时，引导轴与该联动轴的耦合自动取消。耦合关系取消后，系统显示以下报警：
  报警 16772“通道 *通道号* 程序段 *程序段号* 轴 *轴号* 是随动轴，打开耦合”
  直至“REF.POINT”模式取消后，耦合才会再次被激活。

⚠小心

联动轴脱离耦合会导致人身伤害

在对联动轴进行回参考点操作时，引导轴与该联动轴的耦合自动取消。若此时立即对引导轴进行回参考点操作而不切换“REF.POINT”模式，则联动轴不会跟随引导轴运行，此可能导致人员受伤和财产损失。

- 确保在对联动轴进行回参考点操作之前取消引导轴与联动轴之间的耦合。

- 联动轴：零件程序命令 G74
  不支持使用 G74 编程指令对联动轴组中的联动轴进行回参考点操作。

剩余行程：联动轴

联动轴的剩余行程为其“相依”运行与“独立”运行所剩运行距离的总和。

删除剩余行程：联动轴

删除联动轴的剩余行程仅会终止该联动轴跟随引导轴的独立运行部分。

响应 NC 启动

联动轴组在 NC 启动期间的行为由以下机床数据的设置决定：

MD20112 $MC_START_MODE_MASK（确定 NC 启动后数控系统的初始设置）

位	值	含义
8	0	联动轴组在 NC 启动后保持生效。
	1	联动轴组在 NC 启动后被解除。

响应复位/零件程序结束

联动轴组在复位/零件程序结束后的行为由以下机床数据的设置决定：

MD20110 $MC_RESET_MODE_MASK（确定复位/零件程序结束后数控系统的初始设置）

位	值	含义
8	0	复位/零件程序结束后联动轴组被解除。
	1	复位/零件程序结束后联动轴组仍然保持生效。

8.1.3 编程

8.1.3.1 定义并打开联动轴组（TRAILON）

联动轴组的定义及打开随 TRAILON 零件程序命令的执行而同时实现。

编程

句法：TRAILON (<联动轴>, <引导轴>, [<耦合系数>])

生效方式：模态

参数：

联动轴：类型：AXIS

值范围：通道中所有定义的轴和主轴名称

引导轴：类型：AXIS

值范围：通道中所有定义的轴和主轴名称

耦合系数：联动轴与引导轴的相对运动之比由可选耦合系数指定：

耦合系数 = 联动轴位移 / 引导轴位移

耦合系数为负表明引导轴与联动轴的运动方向相反。

类型：	REAL	
值范围：		± (2,2 * 10-308 … 1,8 * 10+308)
缺省值		+1.0

说明

联动主轴未达到设定转速

主轴实现联动功能时，引导主轴的速度限制为 MD35130（速度控制模式下的最大速度）；联动主轴的速度限制为 MD35135（位置控制模式下的最大速度）。MD35135 的默认值 0 表示 MD35130 值的 90% 为位置控制模式下的最大速度。因此，当耦合系数大于 0.9 时，由于 MD35135 的速度限制，联动主轴无法达到设定的速度值。

- 在此例中，须将 MD35135 设为合适的值，从而使联动主轴达到所需的速度。

示例：

程序代码	注释
`TRAILON(V,Y,2)`	；定义并打开联动轴 `V` 与引导轴 `Y` 之间的耦合。耦合系数为 `2`。

8.1.3.2 关闭（TRAILOF）

联动轴与引导轴之间耦合的关闭通过 `TRAILOF` 零件程序命令来实现。

编程

句法：	`TRAILON`(<联动轴>, <引导轴>)	
	`TRAILOF`(<联动轴>)	
生效方式：	模态	
参数：		
联动轴：	类型： AXIS	
	值范围：	通道中所有定义的轴和主轴名称
引导轴：	类型： AXIS	
	值范围：	通道中所有定义的轴和主轴名称

示例：

程序代码	注释
`TRAILOF(V,Y)`	；取消联动轴 `V` 与引导轴 `Y` 之间的耦合。

8.1.4 PLC 接口信号的生效方式

独立联动轴

联动轴所有相关联的通道及轴专用接口信号均对该联动轴的独立运动生效，例如：

- DB3200.DBX0.3（激活 DRF）
- DB380x.DBB0（进给率补偿）
- DB380x.DBX1.3（进给轴禁用）
- DB380x.DBX2.1（控制器使能）
- DB380x.DBX4.0/1（激活手轮）
- DB380x.DBX4.3（进给停止）
- …

通过这些信号，同时借助“AUTO”及“MDA”模式下使用手轮设定的进给倍率或 DRF 偏移，可以改变联动轴独立运动的速度。

相依联动轴

对于联动轴跟随引导轴所做的相依运动部分，只有那些影响运动终止的联动轴接口信号（例如轴专用信号进给停止、轴禁用、数控系统使能等）生效。

引导轴

当联动轴组生效时，引导轴的接口信号（IS）会借助轴耦合关系作用于适当的联动轴，即

- 引导轴的位置偏移或进给控制动作可基于耦合系数作用于联动轴并在该联动轴上实现相应的位置偏移或进给控制动作。
- 由接口信号（例如，轴专用信号进给停止、轴禁用、伺服使能等）触发的引导轴停止会触发相应的联动轴停止。

位置测量系统 1/2（DB380x.DBX1.5/1.6）

对于生效的联动轴组，仍然可对引导轴及联动轴进行位置测量系统切换。耦合不会被取消。

建议在取消耦合之后再对位置测量系统进行切换。

跟踪（DB380x.DBX1.4）

通过设置下列 NC/PLC 接口信号在 PLC 程序中激活轴跟踪：

DB380x.DBX2.1 = 0（控制器使能）

DB380x.DBX1.4 == 1（跟踪模式）

在对联动轴组激活了跟踪模式后，必须同时为该联动轴组中的所有轴（引导轴及联动轴）设置指定的 NC/PLC 接口信号。

若仅对引导轴激活了跟踪模式，则会使耦合关系中的引导轴与联动轴之间产生永久偏移。

可从下列 NC/PLC 接口信号及系统变量获知轴是引导轴还是联动轴以及哪根轴是引导轴或联动轴的相关信息：

DB390x.DBX5003.0（引导轴/主轴有效）

DB390x.DBX5003.1（联动轴/主轴有效）

$AA_COUP_ACT [轴名称]（参见章节“耦合状态 (页 62)”）

8.1.5 耦合状态

可通过下列系统变量查询轴的耦合状态：

$AA_COUP_ACT [轴名称]

值	含义
0	无耦合有效
8	联动生效

8.1.6 动态性能限制

在零件程序中激活联动轴组后，在引导轴运行时会考虑所有联动轴的动态响应，防止联动轴出现过载。

8.1.7 补充条件

数控系统动态性能

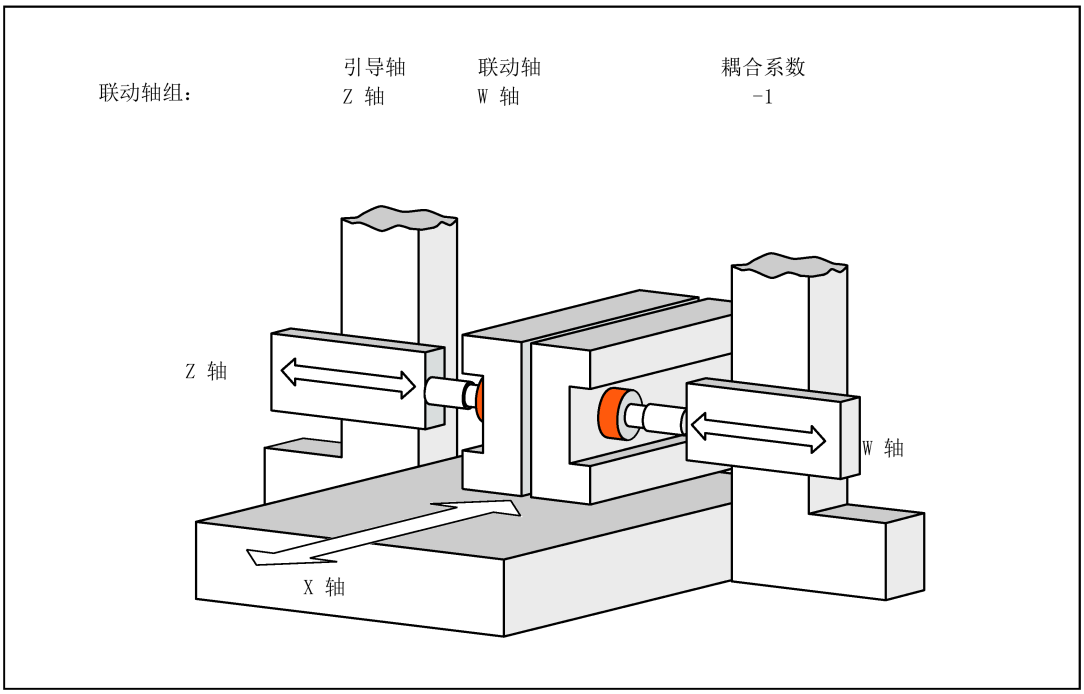
建议对联动轴组中引导轴与联动轴的位置控制参数进行匹配使其一致。

说明

可通过切换参数组来匹配引导轴与联动轴的位置控制参数。

8.1.8 示例

应用示例：双面加工



示例 1

上图所示轴组合的 NC 零件程序示例：

程序代码	注释
TRAILON(W,Z,-1)	; 激活联动轴组
G0 Z10	; Z 轴和 W 轴以相反的轴方向进给
TRAILOF(W,Z)	; 取消联动轴组

示例 2

联动轴的独立与相依运动分量相加所得即为该联动轴的联动运动。相依运动分量可视为联动轴的坐标偏移量。

程序代码	注释
N01 G90 G0 X100 U100	;
N02 TRAILON(U,X,1)	; 激活联动轴组
N03 G1 F2000 X200	; U 轴相依运动，Upos=200，UTrail=100
N04 U201	; 独立移动，Upos=U201，UTrail=100
N05 X250	; U 轴相依运动，Upos=U251，UTrail=150
N06 G91 U100	; 独立移动，Upos=U351，UTrail=150
N07 G90 X0	; U 轴相依运动，Upos=U101，UTrail=-100
N10 TRAILOF(U,X)	

8.2 联动轴的动态响应

8.2.1 动态性能限制的参数设置

通过设置下列机床数据可限制联动轴的动态性能：

MD32000 $MA_MAX_AX_VELO（最大轴速度）

MD32300 $MA_MAX_AX_ACCEL（最大轴加速度）

8.2.2 动态性能限制的程序编写

8.2.2.1 编程（VELOLIMA，ACCLIMA）

降低动态性能限制

可通过**零件程序**来降低由 MD32000 及 MD32300 设定的联动轴（又名跟随轴（FA））的动态性能限制：

命令	含义
VELOLIMA[FA]	降低最大**轴速度**
ACCLIMA[FA]	降低最大**轴加速度**

编写 VELOLIMA[FA] 及 ACCLIMA[FA] 过程中所指定的值为**过程值**。这些值定义了动态性能限制参数设置（MD32000 及 MD32300）所涉及的比例系数：

值范围	含义
1 ≤ 值 < 100	**降低**动态性能限制
100 ≤ 值 < 200	预留

因而可如下所示来计算联动轴的速度及加速度的动态性能限制：

最大轴速度 = MD32000 $MA_MAX_AX_VELO * VELOLIMA[FA]

最大轴加速度 = MD32300 $MA_MAX_AX_ACCEL * ACCLIMA[FA]

说明

降低限制可作用于轴的整体动态性能，即，作用于轴分量在叠加及耦合过程中的运动总和。

上电

上电过程中 VELOLIMA 及 ACCLIMA 的值初始化为 100%。

运行方式转换

动态性能偏移仅在从“AUTO”模式转换为“JOG”模式时保持生效。

复位

（VELOLIMA 及 ACCLIMA）动态性能偏移在复位之后是否保持生效取决于以下通道专用机床数据的设置：

MD22410 $MC_F_VALUES_ACTIVE_AFTER_RESET（复位后 F 功能保持生效）

值	含义
0	复位后 VELOLIMA 及 ACCLIMA 的值恢复为 100%。
1	复位后 VELOLIMA 及 ACCLIMA 的末次程序值仍保持生效。

8.2.2.2 系统变量

对于几何轴、通道轴、机床轴及主轴，零件程序支持下列可读系统变量：

名称	数据类型	描述	单位
预处理			
$PA_ACCLIMA[n]	REAL	通过 ACCLIMA[Ax] 设置的加速度偏移	%

名称	数据类型	描述	单位
$PA_VELOLIMA[n]	REAL	通过 VELOLIMA[Ax] 设置的速度偏移	%
主运行			
$AA_ACCLIMA[n]	REAL	通过 ACCLIMA[Ax] 设置的加速度偏移	%
$AA_VELOLIMA[n]	REAL	通过 VELOLIMA[Ax] 设置的速度偏移	%

说明

读取主运行变量可触发预处理程序停止。

8.3 数据表

8.3.1 机床数据

序号	名称	名称
NC 专用		
18450	MM_NUM_CP_MODULES	CP 模块的最大数量
18452	MM_NUM_CP_MODUL_LEAD	CP 主值的最大数量
通道专用		
20110	RESET_MODE_MASK	上电以及复位/零件程序结束后数控系统的初始设置
20112	START_MODE_MASK	上电以及复位后数控系统的初始设置
22620	START_MODE_MASK_PRT	特殊启动方式下数控系统的初始设置
22621	ENABLE_START_MODE_MASK_PRT	激活机床数据 22620
进给轴/主轴专用		
30130	CTRLOUT_TYPE	设定值的输出类型
30455	MISC_FUNCTION_MASK	各项轴功能
35040	SPIND_ACTIVE_AFTER_RESET	自定义主轴的复位响应

8.3.2 接口信号

序号	位	名称
去向通道的信号		
DB3200.DBX0000	.3	激活 DRF
去向进给轴/主轴的信号		
DB380x.DBB0000	-	进给率补偿
DB380x.DBX1000	.3	轴禁用
DB380x.DBX2000	.1	控制器使能
DB380x.DBX4000	.0/.1	激活手轮
DB380x.DBX4000	.3	进给停止
DB380x.DBX5002	.4	使能联动轴叠加
来自进给轴/主轴的信号		
DB390x.DBX2001	.1	差分速度限制
DB390x.DBX2001	.5	主轴在设定范围内，差分速度
DB390x.DBX2001	.6	超出速度限值，总速度
DB390x.DBX2001	.7	实际旋转方向顺时针，总速度
DB390x.DBX5002	.2	实际值耦合
DB390x.DBX5002	.4	叠加运动
DB390x.DBX5002	.5	速度报警阈值

序号	位	名称
DB390x.DBX5002	.6	加速度报警阈值
DB390x.DBX5003	.0	引导轴/主轴有效
DB390x.DBX5003	.1	联动轴/主轴有效
DB390x.DBX5003	.3	联动轴已加速
DB390x.DBX5003	.5	已达到最大速度
DB390x.DBX5003	.6	已达到最大加速度

9 手动运行和手轮运行（H1）

9.1 JOG 方式下手动操作的一般性能

“JOG”方式

手动移动进给轴/主轴时必须处于“JOG”方式下。生效的运行方式通过信号“生效的运行方式：JOG”（DB3100.DBX0000.2）报告给 PLC，并且可以显示，参见章节“运行方式，程序运行（K1）(页 81)”。

移动选择

可以使用机床控制面板上的移动键(手动移动)或通过连接的手轮(手轮进给)使轴移动。

可以使用移动键使所有机床进给轴同时移动(需使用用户专用的机床控制面板)或者根据安装的手轮数量使用手轮移动进给轴。同时移动几个进给轴时，不执行任何插补。

坐标系

进给轴可以在以下坐标系中移动：

- 机床坐标系(MCS)；可以手动移动每个轴
- 工件坐标系(WCS)；可以手动移动几何轴

机床功能

在**手动运行**下可分为如下几种情况（即机床功能）：

- 连续运行
- 增量运行（INC，预设运行增量的数量）。在制系统中一个增量相当于 0.001 毫米。

机床控制面板接口所提供的机床功能必须通过 PLC 用户程序转换给 PLC/NCK 接口。为此，进给轴专用 NCK/PLC 接口必须用于机床进给轴/主轴，对于几何轴，应使用通道专用 NCK/PLC 接口。对于所有的进给轴/主轴和几何轴：信号位于方式区中(参见下一章节)。

手轮运行

在 MCS 或 WCS 中可以使用手轮移动进给轴。为了计算手轮脉冲，设定增量进给方式（INC...）（参见章节“JOG 方式下手轮运行 (页 69)”）。

几何轴移动

如果加工的工件所在的工件坐标系和机床坐标系不平行(以一角度交错，轮廓中编程的旋转有效)，可以使用移动键或手轮沿工件坐标系中的轴方向移动。这时，当进给轴停止时，应从“AUTO”方式转换到“JOG”方式，并且移动几何轴而不是进给轴。根据工件坐标系旋转是否有效，一到三的进给轴将移动。

机床进给轴移动时，不能另外使用几何轴的移动键进行移动。必须先完成机床轴的移动。否则，输出报警 20062“进给轴已经有效”。手轮 1 和 2 可以同时移动两个几何轴。

说明

几何轴通过单独的通道专用的 PLC 接口定义值来载入。

车床进给轴

几何轴定义为进给轴。如果选择了半径编程(DIAMOF)而非直径编程(DIAMON),在 JOG 方式下移动时，必须遵守以下内容：

- 连续进给：进给轴连续进给时，无变化。
- 增量进给：只进给所选择增量大小的一半位移。
- 手轮进给：对于增量进给，每个手轮脉冲只进给一半的位移。

主轴手动移动

在“JOG”方式下，也可以手动移动主轴。这和手动移动进给轴的条件一样。可以使用移动键/信号“连续”或“INC...“在 JOG 方式下连续移动主轴。通过进给轴/主轴专用 PLC 接口选择和使能此功能。

可以在定位方式(主轴位置控制)和控制方式下手动移动主轴。使用当前变速档的参数记录(机床数据)。

速度

JOG 方式下进给轴/主轴的移动速度可由下述设定数据确定:

- 直线轴使用 SD41110 JOG_SET_VELO(在 G94 时的 JOG 速度)，旋转轴使用 SD41130 JOG_ROT_AX_SET_VELO(旋转轴 JOG 速度)，主轴使用 SD41200 JOG_SPIND_SET_VELO(用于轴的 JOG 速度)。
- 如果 SD 的值为零，则使用 MD32020 JOG_VELO（常规轴速）的值。对于几何轴，机床进给轴赋值如下：X->X1，Y->Y1，Z->Z1（缺省设定）。

快速进给修调

在按方向键时如果再叠加快进键，则进给轴以 MD32010 JOG_VELO_RAPID（“JOG”方式下轴快进速度）确定的快进速度运动。

对于几何轴，机床进给轴赋值如下：X->X1，Y->Y1，Z->Z1（缺省设定）。系统应使用单独的 PLC 接口用于几何轴。

速度修调

只要设置了接口信号“进给修调有效”(DB380x.DBX0001.7)，JOG 方式下进给轴的速度便可以由进给修调开关进行修调。开关位置位于 0%时，即使没有设定“修调有效”，进给轴也不会移动。

对于几何轴，使用通道专用进给修调开关，或者，如果需要快速进给修调，则快速进给修调开关有效。

对于主轴使用有效的主轴修调开关。

加速度

手动方式下也可以按照预定的特性曲线加速，在轴相关 MD32300 MAX_AX_ACCEL 中定义了轴加速度。在“JOG”方式下运行时，可以按预设好的标线设置相应的加速度。在章节“加速度（B2）(页 39)”中对允许的设置进行了说明。

PLC 接口

对于**几何轴**(WCS 中的进给轴)，使用一个单独的 PLC 接口(DB3200.DBB1000,ff 或 DB3300.DBB1000,ff)，此接口所包含的信号和进给轴专用 PLC 接口的相同。

手动移动**主轴时**，NCK 和 PLC 间的 PLC 接口信号和机床进给轴中的信号作用相同。如果主轴在位置控制方式，只设置信号“使用精准停或粗准停到达位置”。

对于纯主轴专用的接口信号，主轴在 JOG 方式移动时必须遵守以下内容：

- 以下用于主轴的 PLC 接口信号无效：
  - 信号“M3/M4 反向”(DB380x.DBX2001.6)
  - 信号“设定逆时针旋转方向”或“设定顺时针旋转方向”（DB380x.DBX2002.7/.6）
  - 信号“摆动速度”（DB380x.DBX2001.5）
- 以下用于主轴的 PLC 接口信号不设置：
  - 信号“顺时针旋转方向”(DB390x.DBX2001.7)
  - 信号“主轴在给定值范围”(DB390x.DBX2001.5)

说明

按复位键将取消手动进给(进给轴/主轴)并产生制动斜坡。

极限

手动进给时，以下极限有效：

- 软件限位开关 1 或 2(轴必须回参考点)
- 硬件限位开关

系统应内部确保一旦到达第一个有效的极限时，终止进给。速度控制可以保证在适当时间内触发制动过程，使轴能够在极限位置准停(如软件限位开关)。如果硬件限位开关响应，进给轴只能使用“快速停止”来制动。

如果到达相应的极限位置，则输出报警信息。系统将停止在此方向的继续移动。进给键和手轮在此方向上无效。

说明

为了确保软件限位开关和工作区极限功能有效，进给轴必须先回参考点。

说明

将已经到达极限位置的进给轴空运行取决于机床制造商。请注意机床制造商的文献资料。请注意机床制造商的文献资料。

关于工作区极限，硬件和软件限位开关的更多说明，参考章节“进给轴监控（A3）(页 25)”。

9.2 持续运行

选择

在选定“JOG”运行方式时，会自动设置有效机床功能信号“持续运行”：

- 几何轴上：DB3300.DBX1001.6，DB3300.DBX1005.6，DB3300.DBX1009.6
- 在机床轴/主轴上：DB390x.DBX0005.6

在“JOG”运行方式下也可以通过 PLC 接口来激活持续运行(接口信号“机床功能：持续运行”)。PLC 决定通过信号“INC 输入在方式组区域生效”(DB2600.DBX0001.0)，INC/持续运行信号在哪个信号区输给 NCK：

DB2600.DBX0001.0 = 1	→	在方式组区域：DB3000.DBB0002, 适用于所有进给轴
DB2600.DBX0001.0 = 0	→	在几何轴/进给轴区域： DB3200.DBB1001，DB3200.DBB1005, DB3200.DBB1009，DB380x.DBB0005

移动键 + / -

正负方向运行功能使相关轴按所需方向运行。

NCK 接口信号上的运行键 PLC 信号：

- 对于几何轴(在 WCS 中进给)：
  DB3200.DBX1000.7/.6，DB3200.DBX1004.7/.6，DB3200.DBX1008.7/.6
- 对于机床进给轴/主轴(在 MCS 中进给):DB80x.DBX004.7/.6

如果同时激活进给轴的两个运行键，将不产生运行动作或进给轴当前的运行状态停止。

运行指令 + / -

一旦进给轴/主轴要求运行（比如：按下轴运行键），根据不同的移动方向，接口信号“运行命令 + ”或“运行命令 - ”会传给 PLC：

- 几何轴上：DB3300.DBX1000.7/.6，DB3300.DBX1004.7/.6，DB3300.DBX1008.7/.6
- 在机床轴/主轴上：DB390x.DBX004.7/.6

“JOG”方式下连续运行

只要运行键一直按着，除非已经到达极限位置，否则进给轴将一直运行。当松开运行键时，轴即被制动，表示动作已经完成。

9.3 增量运行(INC)

增量设置

JOG 方式下进给轴移动距离由所选增量(也称步进量)大小确定。在移动进给轴之前，必须为其设置所期望的增量。

通过机床控制面板来进行设置。某个操作执行完以后，必须根据所需的增量在 PLC 用户程序中设置接口信号“机床功能：INC1 到 INCvar”。PLC 决定通过信号“INC 输入在方式 组区域生效”(DB2600.DBX0001.0)，INC 运行信号在哪个信号区输给 NCK：

DB2600.DBX0001.0 = 1	→	在方式组区域：DB3000.DBB0002, 适用于所有进给轴
DB2600.DBX0001.0 = 0	→	在几何轴/进给轴区域： DB3200.DBB1001，DB3200.DBB1005, DB3200.DBB1009，DB380x.DBB0005

激活的机床功能信号“INC...”从 NCK 传给 PLC：

- 几何轴上：DB3300.DBX1001.0，DB3300.DBX1005.0，DB3300.DBX1009.0 到.5
- 在机床轴/主轴上：DB390x.DBX0005.0 到.5

可设置的增量

操作者可以设置不同的增量:

- **固定增量**大小适用于所有进给轴：INC1、INC10、INC100、INC1000（只通过接口信号：INC10000）。
- **可变增量**(INCvar)。使用通用设定数据 SD41010 JOG_VAR_INCR_SIZE(INC/手轮可变增量大小)可以定义可变增量，它适用于所有的进给轴。

运行键和运行命令

和连续运行相同（参见章节“连续运行 (页 68)”）。

取消运行

如果增量不需完全运行，可用复位键或通过接口信号“取消剩余行程”(DB380x.DBX0002.2)终止运行。

9.4 JOG 方式下手轮运行

选择

“JOG”方式必须有效。使用手轮运行时，操作人员必须设定增量方式 INC1, INC10,..,有效。

最多可连接 2 个手轮。因此可以同时而独立地运行最多 2 个进给轴。

通过接口信号给几何轴或进给轴 (WCS 或 MCS) 分配手轮。

通过以下可以设置转动手轮 1 和 2 使某个进给轴运行：

- 通过 PLC 用户接口使用信号“使能手轮 1 和 2”
  - 对于机床轴(在 WCS 中进给)：DB380x.DBX0004.0 到.2
  - 对于几何轴(在 MCS 中进给)：DB3200.DBX1000.0 至 .2、DB3200.DBX1004.0 至 .2、DB3200.DBX1008.0 至 .2。

  通过 PLC 用户程序连接 PLC 接口的手轮分配。仅通过 PLC 用户程序才可以同时给几个进给轴分配一个手轮。
- 通过菜单引导的操作 (HMI)。
  按下“JOG”方式下主菜单中的如下软键 ，则显示“手轮”窗口：

  在此窗口下，可以给每个轴(WCS 或 MCS)分配一个手轮。

如果要从操作面板(HMI)激活手轮，在 PLC 和 HMI 之间需提供单独的用户接口。由 PLC 基本程序提供的用于手轮 1 和 2 的接口包括以下信息：

- 分配给手轮的轴号信号“手轮 n 轴号”
  (DB1900.DBB1003, ff)
- 机床轴或者几何轴的附加信息
  接口信号“机床轴”(DB1900.DBB1003.7, ff)

PLC 用户程序必须为定义轴设定相应接口信号“使能手轮”值为“0”(取消使能)或“1”(使能)。

定义为路径或速度

转动电子手轮可以使被分配手轮的进给轴在正方向或负方向运行(取决于所需的旋转方向)。

可以通过 MD11346 HANDWH_TRUE_DISTANCE(手轮路径或速度定义)设置手轮移动类型，从而与应用目标相适应。

- MD 值=0 (标准):
  手轮定义是速度定义。当手轮停止时，轴将以最短的位移减速。
- MD 值 = 1：
  手轮定义是路径定义。脉冲不会丢失。由于有最大允许速度极限，因此轴运行时必须遵守，以避免超程。尤其在计算快速手轮脉冲时应考虑这一因素。路径或速度的定义值也可以是 2 或 3。

计算

转动手轮使轴移动，其运行行程/速度的大小取决于如下几点:

- 接口处收到的手轮脉冲数
- 有效增量 (机床功能 INC1, INC10, INC100, ...)
  在制系统中一个增量相当于 0.001 毫米。
- 手轮脉冲计算，由机床数据 MD11320 HANDWH_IMP_PER_LATCH (手轮每个刻度脉冲数)确定。

运行指令 + / -

进给轴运行时，按照不同的运行方向给 PLC 发出接口信号“运行指令+”或“运行指令 - ”。

- 几何轴上：DB3300.DBX1000.7/.6，DB3300.DBX1004.7/.6，DB3300.DBX1008.7/.6
- 在机床轴/主轴上：DB390x.DBX004.7/.6。

如果已经通过操作面板上的方向键使进给轴移动,则不可能再用手轮运行。否则发出报警 20051“不可以进行手轮运行”。

速度

速度取决于手轮所产生的脉冲和计算的脉冲：每个时间段的运动行程。该速度由轴专用的 MD32000 MAX_AX_VELO 值限制。

终止/中断运行

用复位键或通过接口信号“清除剩余行程”(DB380x.DBX0002.2)可以终止进给轴的运行。给定值/实际值之差(即剩余行程)将被清除。

用 NC 停止键仅能中断轴运行，再按 NC 启动键时运行接下去的剩余行程。

反向运行

与机床数据 MD11310 HANDWH_REVERSE 中设定的值有关，反向运行的性能如下：

- MD 值 = 0：
  如果反向转动手轮，则计算由此所产生的行程段，并且尽可能快地回到所计算的终点。在此分两种情况：如果该终点位于轴在当前运行方向所能制动的点之前,则进给轴首先制动，然后在相反方向回终点。其他情况下则直接进行终点回位运行。
- MD 值 > 0：
  如果手轮在反向至少运行机床数据中给出的脉冲数，进给轴将被尽快制动，所有在插补结束之前接收的脉冲都不予处理。这意味着只有在进给轴停止(给定值侧)后才能重新移动（新功能）。

软件限位极限响应

微动模式在“JOG”方式下，进给轴碰到第一个有效限位开关时停止运行，并发出相应的报警。

与机床数据 MD11310 HANDWH_REVERSE 中的设定有关，具有如下响应(轴还未到达给定终点)：

- MD 值 = 0：
  手轮脉冲产生的路径构成一个虚拟终点,供在下面的计算中使用：如果虚拟终点位于限位开关之后(比如 10 毫米之后)，则在进给轴真正运行之前，必须先在反方向运行 10 毫米。如果要在限位开关处使进给轴立即在反向真正运行，则可以通过“清除剩余行程”（DB380x.DBX0002.2）或者取消手轮分配清除虚拟的剩余行程。
- MD 值 > 0：
  所有导致轴运行超过限位开关的手轮脉冲都不予处理。反方向旋转手轮，则轴立即在反向运行，即离开限位位置。

9.5 JOG 运行模式中的固定点运行

9.5.1 引言

功能

“在 JOG 运行模式中运行至固定点” 功能可帮助机床操作者利用机床控制面板中的运行键或手轮将轴运行至机床数据中已定义的轴位置。运行轴到达已定义的固定点后自动停止运行。

应用

典型应用示例：

- 启动 NC 程序前移动至基本位置。
- 运行至刀具更换点、装料点、托盘更换点。

前提条件

- “在 JOG 中运行至固定点”功能只能在“JOG”模式中激活。
  在“JOG-REPOS”、“JOG-REF” 和 JOG 的“AUTO”模式下，不能激活此功能。
- 待运行的轴必须返回参考点。
- 不允许激活车铣复合加工。
- 待运行的轴不能为已进行有效耦合的同步轴。
- 不能进行 ASUB 加工。

使用 G75 运行至固定点

也可通过指令 G75 从零件程序中调用运行至已定义固定点功能。

通过 G75 运行至固定点的更多信息，参见 SINUMERIK 808D ADVANCED 编程和操作手册，章节：“接近固定点”。

9.5.2 功能性

步骤

"在 JOG 运行模式下运行至固定点”的步骤：

- 选择“JOG”方式
- 激活“在 JOG 运行模式下运行至固定点”功能
- 通过运行键或手轮将机床轴运行至固定点

有效性

选择“在 JOG 运行模式下运行至固定点”后，PLC 会发出如下接口信号：
“运行至 JOG 固定点”(DB380x.DBX1001.0-2)

待运行固定点的编号通过二进制数字 0-2 编码。只要此功能有效，NC 将发出如下接口信号进行确认：
“运行至 JOG 固定点已激活”(DB390x.DBX1001.0-2)

过程

通过运行键或手轮启动运行，运行方向为待运行的固定点的方向。

选择的机床轴运行至指定固定点自动停止。

使用“精准停"功能到达固定点后，会发出相应的 NC/PLC 接口信号：
“已到达 JOG 固定点”(DB390x.DBX1001.3-5)

当运行轴通过其它方法（比如 NC 程序、同步等）到达机床坐标系中的额定固定点位置且实时位置位于“精准停”(MD36010 STOP_LIMIT_FINE) 公差窗口内时，也会发出上述显示信号。

反向运行

反向运行，即：朝着指定固定点的反方向运行，可通过如下机床数据中的位 2 的数据进行设置：
MD10735 JOG_MODE_MASK（“JOG”方式设置）

只有设置好位 2 的数据后，才可能进行反向运行。

若位 2 的数据未进行设置，则反方向运行被锁定，尝试使用运行键或手轮沿指定固定点的反方向运行时，会发出如下通道状态消息：

“JOG：此方向已锁定 <轴>”

运行至其它的固定点

在固定点运行过程中选择其它的固定点时，会中断轴运行过程并发出如下消息：
报警 17812“通道 %1 轴 %2 JOG 运行模式中的固定点：固定点已改变”

消息信号“运行至 JOG 固定点已激活”会显示新选择的固定点的编号。若需继续运行，必须重新启动 JOG 运行。

说明

若不想出现报警消息，机床操作者可如下进行：

1.通过消除剩余路径方式中断当前运行。

2.运行轴停止运行后激活其它的固定点，然后启动运行。

离开固定点/取消

若需离开固定点位置，则必须取消“在 JOG 运行模式下运行至固定点”功能。可将激活信号复位为“0”取消上述功能：
DB380x.DBX1001.0-2 = 0

消息信号“运行至 JOG 固定点已激活”和“已到达 JOG 固定点”在轴离开固定点位置时删除。

特殊情况：轴已经位于固定点

若启动运行至固定点功能时运行轴已位于指定的固定点，则轴将不再运行。此种情况下，会发出如下通道状态消息：
“JOG：已到达位置 <轴>”

若需离开固定点位置，则必须取消“在 JOG 运行模式下运行至固定点”功能。

增量方式运行的特点

在增量运行方式中，若在增量已完全运行完毕前已到达固定点，则视为增量已完全运行完毕。此种情况也适用于仅运行整个增量的情形：

MD11346 HANDWH_TRUE_DISTANCE = 2 或 3

模数回转轴中的特点

模数回转轴可沿固定点的两个方向运行（MD10735 中位 2 的数据无任何意义）。不会进行最短路径(DC)运行考虑。

主轴中的特点

激活“在 JOG 运行模式下运行至固定点”功能后，主轴切换至定位运行模式。位置控制功能也因此而激活，可以运行至固定点。

若未发现任何零点，则会像轴运行模式中一样发出如下报警消息：
报警 17810“通道 %1 轴 %2 未回参考点”

由于主轴必须始终是一个模数回转轴，所以在运行方向判断方面同模数回转轴相同（参见段落“模数回转轴中的特点”）。

9.5.3 参数设置

反向运行

反向运行，即：朝着指定固定点的反方向运行，可通过如下机床数据中位 2 的数据进行设置：

MD10735 JOG_MODE_MASK（"JOG"方式设置）

位	值	说明
2	0	**不能**沿反方向运行（基本设置）。
	1	可以沿反方向运行。

固定点位置

通过以下机床数据最多可为每个轴定义 4 个固定点位置：

MD30600 FIX_POINT_POS[n]

有效固定点位置个数

通过此机床数据确定某个轴的有效固定点位置的个数：

MD30610 NUM_FIX_POINT_POS

说明

唯一的例外情况为“使用 G75 运行至固定点”。其可通过设置 (MD30610 = 0) 运行至两个固定点。

9.5.4 编程

系统变量

对于“运行至固定点”功能可使用下列系统变量（这些变量可在零件程序和同步中读取）。

系统变量	说明
$AA_FIX_POINT_SELECTED [<轴>]	指定固定点的编号
$AA_FIX_POINT_ACT [<轴>]	当前轴所处的固定点的编号

9.5.5 边界条件

轴是分度轴

当运行轴是一个分度轴且待运行的固定点与分度位置不一致时，轴不会运行，此时会发出报警。

框架已激活

忽略所有生效的框架。在机床坐标系中运行。

补偿值有效

超越有效的补偿值（外部零点偏移，同步动作偏移 $AA_OFF，在线刀具补偿)。固定点是机床坐标系中的一个点。

若在 JOG 模式下的固定点运行过程中激活了补偿运行（外部零点偏移、同步动作偏移 $AA_OFF、在线刀具补偿），则会发出报警。运行轴将不会运行至机床坐标系中指定固定点的位置，而是运行至假设未激活补偿运行时应达到的位置。 不会发出 NC/PLC 接口信号“已到达 JOG 固定点”(DB390x.DBX1001.3-5)。

9.5.6 应用举例

目标

回转轴（机床轴 4 [AX4]）将通过“在 JOG 运行模式下运行至固定点”功能运行至固定点 2（90 度）。

参数设置

机床轴 4 的“运行至固定点”机床数据参数如下设置：

MD30610 NUM_FIX_POINT_POS[AX4] = 4	为机床轴 4 定义 4 个固定点。
MD30600 FIX_POINT_POS[0,AX4] = 0	AX4 的第 1 个固定点为 0 度
MD30600 FIX_POINT_POS[1,AX4] = 90	AX4 的第 2 个固定点为 90 度
MD30600 FIX_POINT_POS[2,AX4] = 180	AX4 的第 3 个固定点为 180 度
MD30600 FIX_POINT_POS[3,AX4] = 270	AX4 的第 4 个固定点为 270 度

初始状态

机床轴 4 已回参考点且位于位置 0 度上。这与第 1 个固定点匹配，故发出如下 NC/PLC 接口信号：
DB390x.DBX1001.0 = 1（位 0-2 = 1）

运行至固定点 2

数控系统切换至“JOG”方式。

通过如下 NC/PLC 接口信号激活运行至固定点 2 的“运行至固定点”功能：
DB380x.DBX1002.1 = 1（位 0-2 = 2）

然后发出如下 NC/PLC 接口信号对以上激活进行确认：
DB390x.DBX1001.1 = 1（位 0-2 = 2）

通过机床控制面板上的正向运行键持续运行后即可到达固定点 2。

机床轴 4 在 90 度处停止。然后发出如下 NC/PLC 接口信号：
DB390x.DBX1001.4 = 1（位 3-5 = 2）

9.6 数据表

9.6.1 机床数据

序号	标识	名称
一般		
10000	AXCONF_MACHAX_NAME_TAB[n]	机床进给轴名称[n=轴索引]
10735	JOG_MODE_MASK	“JOG”方式设置
11310	HANDWH_REVERSE	定义反方向移动
11320	HANDWH_IMP_PER_LATCH[0]...[2]	手轮每个刻度脉冲数
11346	HANDWH_TRUE_DISTANCE	手轮路径和速度定义
通道专用		
20060	AXCONF_GEOAX_NAME_TAB[n]	通道内的几何轴[n=几何轴索引]
20100	DIAMETER_AX_DEF	具有进给轴功能的几何轴
进给轴/主轴专用		
30600	FIX_POINT_POS[n]	轴的固定点位置
30610	NUM_FIX_POINT_POS	轴固定点位置的个数
32000	MAX_AX_VELO	最大轴速度
32010	JOG_VELO_RAPID	常规快速移动
32020	JOG_VELO	常规轴速度
32300	MAX_AX_ACCEL	轴加速度
32420	JOG_AND_POS_JERK_ENABLE	激活轴特定的过冲限制
32430	JOG_AND_POS_MAX_JERK	轴特定的过冲
35130	GEAR_STEP_MAX_VELO_LIMIT[0]...[5]	变速档/主轴最大转速

9.6.2 设定数据

序号	名称	名称
一般		
41010	JOG_VAR_INCR_SIZE	INC/手轮可变增量大小
41110	JOG_SET_VELO	直线轴 JOG 速度
41130	JOG_ROT_AX_SET_VELO	回转轴下的 JOG 速度
41200	JOG_SPIND_SET_VELO	主轴的 JOG 速度

9.6.3 接口信号

序号	位	名称
HMI 到 PLC 信号		
DB1900.DBX1003	.0 至 .2	用于手轮 1 控制的轴号
DB1900.DBX1004	.0 至 .2	用于手轮 2 控制的轴号
NCK 专用		
DB2600.DBX0001	.0	INC 输入在操作方式区有效
运行方式-专用		
DB3000.DBX0000	.2	“JOG”方式
DB3000.DBX0002	.0 至 .6	运行方式区域机床功能 INC1 连续运行
DB3100.DBX0000	.2	有效方式：JOG
通道专用		
DB3200.DBX1000 DB3200.DBX1004 DB3200.DBX1008	.1、.0 .1、.0 .1、.0	激活手轮 (2, 1) 用于几何轴 1 用于几何轴 2 用于几何轴 3
DB3200.DBX1000 DB3200.DBX1004 DB3200.DBX1008	.4 .4 .4	运行键禁用， 用于几何轴 1 用于几何轴 2 用于几何轴 3
DB3200.DBX1000 DB3200.DBX1004 DB3200.DBX1008	.5 .5 .5	快速运行叠加， 用于几何轴 1 用于几何轴 2 用于几何轴 3
DB3200.DBX1000 DB3200.DBX1004 DB3200.DBX1008	.7 或者.6 .7 或者.6 .7 或者 .6	正或负运行键， 用于几何轴 1 用于几何轴 2 用于几何轴 3
DB3200.DBX1000 DB3200.DBX1004 DB3200.DBX1008	.0 到.6 .0 到 .6 .0 到 .6	机床功能 INC1 连续运行， 用于几何轴 1 用于几何轴 2 用于几何轴 3
DB3300.DBX1000 DB3300.DBX1004 DB3300.DBX1008	.1、.0 .1、.0 .1、.0	激活手轮 (2, 1) 用于几何轴 1 用于几何轴 2 用于几何轴 3
DB3300.DBX1000 DB3300.DBX1004 DB3300.DBX1008	.7 或者.6 .7 或者.6 .7 或者 .6	正向或负向运行指令， 用于几何轴 1 用于几何轴 2 用于几何轴 3
DB3300.DBX1001 DB3300.DBX1005 DB3300.DBX1009	.0 到 .6 .0 到 .6 .0 到 .6	机床功能 INC1 连续运行生效， 用于几何轴 1 用于几何轴 2 用于几何轴 3
进给轴/主轴专用		
DBB380x.DBX0000	-	进给率修调
DB380x.DBX0000	.7	修调有效
DB380x.DBX0002	.2	删除剩余行程
DB380x.DBX0004	.1、.0	激活手轮 (2, 1)
DB380x.DBX0004	.4	移动键锁定

序号	位	名称
DB380x.DBX0004	.5	快速进给修调
DB380x.DBX0004	.7 或 .6	移动键正或负
DB380x.DBX0005	.0 至 .6	轴区域机床功能 INC1 连续运行
DB380x.DBX1002	.0 至 .2	JOG 运行模式中已激活的固定点运行（二进制编码：固定点 1 至 4）
DB390x.DBX0000	.7/.6	粗/精准停到达位置
DB390x.DBX0004	.1、.0	激活手轮 (2, 1)
DB390x.DBX0004	.7 或 .6	运行指令正或负
DB390x.DBX0005	.0 至 .6	有效机床功能 INC1 连续运行
DB390x.DBX1001	.0 至 .2	JOG 运行模式中已激活的固定点运行（二进制编码）
DB390x.DBX1001	.3 至 .5	已到达固定点（二进制编码）

10 输出给 PLC 的辅助功能（H2）

10.1 概述

辅助功能

在机床上加工工件时，数控系统利用零件程序可以事先设定进给轴加工位置和插补方式，也可以设置技术功能(进给、主轴转速、变速档)，另外还可以设置一些对机床附属设备进行控的辅助功能(滑枕向前、夹持器张开、卡盘夹紧)。 将这些不同的类型用“辅助功能”这个总概念来表示。

辅助功能分为：

- 辅助功能 M
- 主轴功能 S
- 辅助功能 H
- 刀具号 T
- 刀具补偿 D
- 进给率 F(SINUMERIK 808D ADVANCED 上，F 不输出给 PLC)

将辅助功能发送给 PLC

在程序加工过程中可以在某一特定的时间激活某一功能，并输出到 PLC，如，PLC 要执行的有关机床工具的特定的辅助功能。 这可以通过将相应的辅助功能，包括它们的参数，输出到 PLC 来实现。

传送的指令和信号由 PLC 应用程序处理。 以下章节中介绍了各种辅助功能的配置，编程和作用方式。

辅助功能组

辅助功能可以归纳为组。

10.2 辅助功能编程

辅助功能的通用结构

字母[地址扩展] = 值

允许用于辅助功能的字母为：**M、S、H、T、D、F**。

地址扩展必须为整数。使用数字值直接定义地址扩展时，括号可以省略。

该值对于单个辅助功能，定义不同：

- INT=整数
- REAL=浮点值

下表介绍了辅助功能编程：

功能	地址扩展（整数）		值			说明	每个程序段个数
	含义	范围	范围	类型	含义		最大值
M	主轴号	1 - 2	0-99	整数	功能	一些数字分配了固定的功能。	5
S	主轴号	1 - 2	0-±3.4028 ex 38	实数	主轴转速		1
H	任意	0 - 99	±3.4028 ex 38	实数	任意	功能对 NCK 没有影响，只能通过 PLC 实现。	3
T	-	-	0-32000	整数	刀具选择		1
D	-	-	0-9	整数	刀具补偿选择	D0 撤销选择，D1 为缺省值	1
F	-	-	0,001-999 999,999	实数	轨迹进给率		1

每个程序段中最多可以编程 10 个辅助功能。如果超出了规定的地址扩展范围或值或者编程了错误的数据类型，则输出报警 14770“编程的辅助功能不正确”。下表列出了 H 功能的编程举例。

如果超出了程序段中允许的辅助功能数量，报警 12010 将输出。

下表给出了 H 功能的编程示例：

编程	H 功能输出到 PLC
H5	H0=5.0
H=5.379	H0=5.379
H17=3.5	H17=3.5
H5.3=21	错误,报警 14770

程序段切换

只有当 PLC 响应了所有的辅助功能后，才可以由 NCK 向 PLC 发出一个新的辅助功能。在用户接口中，辅助功能最少要有一个 PLC 循环的长度。当所编程的运动执行完毕，并且响应了辅助功能之后一个程序段才算结束。对此，为了确保 PLC 用户程序没有丢失辅助功能，NCK 可能会停顿一下零件程序的执行。

10.3 传输值和信号到 PLC 接口

传输时间

只有当所有进给轴和主轴的 SPOS 动作完成后，才在程序段终点输出辅助功能(M2)。

如果在动作程序段中编程了几个具有不同输出类型的辅助功能(动作的前，中，后)，则根据它们特定的输出类型输出各个辅助功能。

在没有进给的程序段中，辅助功能作为一个程序段立即输出。

连续路径运行

在**加工运行期间**输出辅助功能并在轨迹结束以前进行响应，则可以得到连续路径加工。参见章节“轨迹控制运行 (页 34)”。

接口信号

NCK 到 PLC 的信号传输。

10.4 辅助功能的分组

功能

待输出的辅助功能 M，H，D，T 和 S，可以通过机床数据分为不同的功能组。

每个辅助功能只允许属于一个功能组。

每个功能组中只可以编程一个辅助功能。否则会发出报警 14760。

配置

最多可以定义 64 个辅助功能组。在这 64 个辅助功能组中，最多可以分配 64 个辅助功能。缺省值功能组(1 到 3)中的辅助功能将不作考虑。

在机床数据 MD11100 AUXFU_MAXNUM_GROUP_ASSIGN(分配到辅助功能组中的辅助功能个数)中输入功能组中实际所分配的辅助功能个数。为此，必须设置保护级 1 密码。然后系统必须重新上电。现在只有索引 n 大于零的机床数据有效，并可以输入相应的数值。

一个辅助功能由下面的机床数据确定：

- MD22000 AUXFU_ASSIGN_GROUP[n]（辅助功能组）
- MD22010 AUXFU_ASSIGN_TYPE[n]（辅助功能类型）
- MD22020 AUXFU_ASSIGN_EXTENSION[n]（辅助功能扩展）
- MD22030 AUXFU_ASSIGN_VALUE[n]（辅助功能值）

预定义的辅助功能组

组 1：

缺省时，辅助功能 M0、M1 和 M2（M17、M30）属于功能组 1。始终在程序段终点输出。

组 2：

缺省时，辅助功能 M3、M4 和 M5（M70）属于功能组 2。始终在动作执行前输出。

组 3：

缺省时，S 功能属于功能组 3。该功能组和其它功能组和动作同时输出。

用户定义功能组

所有的其它功能组(用户定义)和动作一起输出。

未分组辅助功能

未分组的辅助功能和动作同时输出。

配置举例：

将 8 个辅助功能分配到 7 个功能组中：

组 1：M0，M1，M2(M17,M30)-缺省值，须保留

组 2：M3,M4,M5(M70)-缺省值，须保留

组 3：S 功能 - 缺省值，须保留

组 4：M78，M79

组 5：M80，M81

组 6：H1=10，H1=11，H1=12

组 7：所有 T 功能

保护等级 1 的口令已设置。

输入到 MD11100 AUXFU_MAXNUM_GROUP_ASSIGN=8。

然后系统重新上电或使用相应的软键启动系统编程剩余的机床数据，然后重新启动系统。

下表给出了机床数据的条目示例：

索引 n	MD22000 (GROUP)	MD22010 (TYPE)	MD22020 (EXTENSION)	MD22030 (VALUE)
0	4	M	0	78
1	4	M	0	79
2	5	M	0	80
3	5	M	0	81
4	6	H	1	10

索引 n	MD22000 (GROUP)	MD22010 (TYPE)	MD22020 (EXTENSION)	MD22030 (VALUE)
5	6	H	1	11
6	6	H	1	12
7	7	T	0	-1

10.5 程序段搜索动作

程序段搜索时计算

程序段搜索并计算时，集中功能组中的所有的辅助功能，在搜索结束时并在重新启动程序段(除了功能组 1：M0，M1...)前输出这些功能。在任何情况下，输出功能组中最后的辅助功能。

搜集到的所有辅助功能均作为一般的辅助功能，并在移动前作为一个单独程序段输出。

说明

程序段搜索时所得到的辅助功能必须指同一个功能组中的辅助功能。

10.6 辅助功能的描述

10.6.1 M 功能

应用

机床上的各种开关操作可以通过零件程序中的 M 功能指令激活。

功能范围

- 每个程序段可以有五个 M 指令。
- M 功能的值范围：0 至 99；整数
- 少数几个 M 指令已经由系统生产厂商设置了一些固定功能（参见 SINUMERIK 808D ADVANCED 编程和操作手册）。其它功能供机床生产厂商使用。

10.6.2 T 功能

应用

在某一加工过程中需要某一刀具，这可以由 PLC 通过 T 功能实现。 换刀时是用 T 指令直接进行，还是必须后接 M6 指令才生效，这可以通过机床数据 MD22550 TOOL_CHANGE_MODE 调节。

编程的 T 功能可以作为刀具号，也可以作为位置号。

功能范围

每个程序段可以有 1 个 T 指令

特点

T0 特指以下功能：从刀架中取下当前的刀具，但并不换上新的刀具。

10.6.3 D 功能

使用 D 功能可以为有效的刀具选择刀具补偿。刀具补偿在以下有具体说明：

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

10.6.4 H 功能

应用

H 功能用于将零件程序中的不同值传输到 PLC。功能的定义由用户完成。

功能范围

- 每个零件程序段中可以编程 3 个 H 功能
- H 功能的范围值：浮点值(如算术参数 R)
- 地址扩展值 0 到 99(H0=...到 H99 = ...)

10.6.5 S 功能

S 功能用于设定 M3 或 M4 主轴的速度。对于使用 G96（恒定切削率）的车床，则设定切削率。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

10.7 数据表

10.7.1 机床数据

序号	名称	名称
一般		
11100	AUXFU_MAXNUM_GROUP_ASSIGN	辅助功能组中分配的辅助功能个数
通道专用		
22000	AUXFU_ASSIGN_GROUP[n]	辅助功能组
22010	AUXFU_ASSIGN_TYPE[n]	辅助功能类型
22020	AUXFU_ASSIGN_EXTENSION[n]	辅助功能扩展
22030	AUXFU_ASSIGN_VALUE[n]	辅助功能数值

10.7.2 接口信号

序号	位	名称
通道专用		
DB2500.DBX0000	.0 至 .4	M 功能 1 更改到 M 功能 5 更改
DB2500.DBX0006	.0	S 功能 1 更改
DB2500.DBX0008	.0	T 功能 1 更改
DB2500.DBX0010	.0	D 功能 1 更改
DB2500.DBX0012	.0 至 .2	H 功能 1 更改到 H 功能 3 更改
DB2500.DBD2000		T 功能 1 (DINT)
DB2500.DBD3000		M 功能 1 (DINT)
DB2500.DBB3004		M 功能 1 的扩展地址 (字节)
DB2500.DBD3008		M 功能 2 (DINT)
DB2500.DBB3012		M 功能 2 的扩展地址 (字节)
DB2500.DBD3016		M 功能 3 (DINT)
DB2500.DBB3020		M 功能 3 的扩展地址 (字节)
DB2500.DBD3024		M 功能 4 (DINT)
DB2500.DBB3028		M 功能 4 的扩展地址 (字节)
DB2500.DBD3032		M 功能 5 (DINT)
DB2500.DBB3036		M 功能 5 的扩展地址 (字节)
DB2500.DBD4000		S 功能 1 (REAL 格式)

序号	位	名称
DB2500.DBB4004		S 功能 1 的扩展地址 (字节)
DB2500.DBD4008		S 功能 2 (REAL 格式)
DB2500.DBB4012		S 功能 2 的扩展地址 (字节)
DB2500.DBD5000		D 功能 1 (DINT)
DB2500.DBW6004		H 功能 1 的扩展地址 (字)
DB2500.DBD6000		H 功能 1 (REAL 格式)
DB2500.DBW6012		H 功能 2 的扩展地址 (字)
DB2500.DBD6008		H 功能 2 (REAL 格式)
DB2500.DBW6020		H 功能 3 的扩展地址 (字)
DB2500.DBD6016		H 功能 3 (REAL 格式)
DB2500.DBX1000	.0 - .7	动态 M 功能: M00-M07
DB2500.DBX1001	.0 - .7	动态 M 功能: M08-M15
DB2500.DBX1012	.0 - .7	动态 M 功能: M96-M99
DB370x.DBD0000	-	用于主轴的 M 功能(DINT),进给轴专用
DB370x.DBD0004	-	用于主轴的 S 功能(REAL),进给轴专用

11 运行方式，程序运行（K1）

11.1 概述

程序操作

程序运行是指在“AUTO”或“MDA”方式下执行零件程序或零件程序段。在程序运行期间可以通过 PLC 接口信号和指令影响程序运行。

通道

通道是指在其中可以执行零件程序运行的装置。

系统给每个通道设定一个插补器，进行相应的程序处理。一个插补器适用于一个特定的运行方式。

SINUMERIK 808D ADVANCED 数控系统有一个通道。

11.2 运行方式

11.2.1 运行方式

激活

通过 PLC 变量 DB3000.DBB0000 中的接口信号可以激活所要求的运行方式。同时选择运行方式时，设有优先级顺序：

- JOG（高优先级）：通过操作手轮或者按方向键使进给轴运行。此时通道专用信号和互锁不起作用。
- MDA：可以执行程序段。
- AUTO（低优先级）：自动执行零件程序

应答信号

通过 PLC 变量 DB3100.DBB0000 中的接口信号显示生效的运行方式。

手动方式下允许的机床功能

在“JOG”运行方式下可以选择下面的机床功能：
REF（回参考点）

通过接口信号“REF”（DB3000.DBX0001.2）激活所要求的机床功能。生效的机床功能通过接口信号“有效机床功能 REF”（DB3100.DBX0001.2）显示。

停止

通过下列接口信号可以给出一个“停止”信号

- 接口信号“NC 停止”（DB3200.DBX0007.3）
- 接口信号“NC 停止进给轴和主轴”（DB3200.DBX0007.4）
- 接口信号“程序段结束 NC 停止”（DB3200.DBX0007.2）

通过选择不同的停止信号，可以停止进给轴或者使进给轴与主轴一起停止，或者使进给轴在程序段结束处停止。

复位

通过接口信号“复位”（DB3000.DBX0000.7）可以终止当前程序段的执行。

给出接口信号“复位”之后执行下面的动作：

- 零件程序运行立即停止。
- 进给轴和主轴停动。
- 当前程序段在此时刻尚未输出的辅助功能不再输出。
- 程序段显示器复位到零件程序的开始处。
- 显示器中所有的复位报警清除。
- 接口信号“通道状态复位”（DB3300.DBX0003.7）一设置后复位过程就结束。

就绪

运行准备状态由接口信号“808 就绪”（DB3100.DBX0000.3）显示。

11.2.2 运行方式转换

概述

通过接口信号，请求和激活运行方式的转换。

说明

当**不再**存在“通道状态有效”信号（DB3300.DBX0003.5），系统内部才进行运行方式的转换

在通道处于“复位”状态时（IS：DB3300.DBX0003.7，例如按下“复位”键）可以从任何一种运行方式转换到另外一个运行方式。

在通道处于“中断”状态时（IS：DB3300.DBX0003.6）转换方式需要遵循某些条件（参见下表）。

如果想从“自动”方式转换到“手动”方式，则必须首先回到“自动”方式状态或者按复位键。不可能以顺序 AUTO - JOG - MDA 转换运行方式。同样，对于“MDA”方式也不可能不经过复位状态直接或间接地转换到“自动”方式。

下表中列出了不同运行方式和通道状态所可能进行的运行方式转换（“通道复位”或“通道中断”）。

	从	AUTO		JOG			MDA	
					之前是 AUTO	之前是 MDA		
向		复位	中断	复位	中断	中断	复位	中断
自动				X	X		X	
JOG		X	X				X	X
MDA		X		X		X		

上表中打“x”的位置表示可以进行的方式转换。

运行方式转换时出错

如果一个运行方式转换被系统拒绝，则给出一相应的出错信息。清除故障信息时无需改变通道状态。

禁止方式转换

利用接口信号“禁止方式转换“（DB3000.DBX000.4）可以禁止运行方式的转换。此时，运行方式转换请求被抑制。

11.2.3 各种运行方式下的功能选择

功能概述

在哪些运行方式下，哪些运行状态中可选择哪些功能，可参见下表。

运行方式	自动			JOG						MDA				
功能	1	2	3	1	3	4	3	5	3	1	2	3	6	7
通过”通讯”窗口从外部载入零件程序	sb	sb		sb		sb		sb	sb	sb	sb			
加工零件程序或者零件程序段	s	s	b							s	s	b		
程序段搜索	s	s	b											
通过零件程序指令回参考点（G74）			sb									sb		

s:此状态下可以启用该功能
b:此状态下可加工该功能

1：通道位于复位状态
2：通道中断
3：通道生效
4：在“自动”方式中断期间“手动”方式下的通道中断
5：在“MDA”方式中断期间“手动”方式下的通道中断
6：在“MDA”方式中断期间 MDA 中“手动”方式下的通道中断
7：MDA 中“手动”方式下的通道有效

11.2.4 各种运行方式下的监控

监控功能概述

不同的运行方式下有不同的监控方式生效。

下表介绍了监控功能和互锁：

运行方式	自动			JOG						MDA				
功能	1	2	3	1	3	4	3	5	3	1	2	3	6	7
轴相关监控功能/主轴定位时监控功能生效														
软件限位开关 +			x		x		x		x			x	x	x
软件限位开关 –			x		x		x		x			x	x	x
硬件限位开关 +	x	x	x	x	x	x	x	x	x	x	x	x	x	x
硬件限位开关 –	x	x	x	x	x	x	x	x	x	x	x	x	x	x
粗/精准停	x	x	x	x	x	x	x	x	x	x	x	x	x	x
夹紧公差	x	x	x	x	x	x	x	x	x	x	x	x	x	x
DAC 限制 (模拟主轴)	x	x	x	x	x	x	x	x	x	x	x	x	x	x
轮廓监控			x		x		x		x			x	x	x

主轴相关监控功能															
超出速度限值				x		x		x		x				x		x
主轴是静止的	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
主轴同步			x		x		x		x				x		x
转速在给定值范围内			x												
允许的最大转速			x		x		x		x				x		x
编码器极限频率			x		x		x		x				x		x

主轴相关监控功能	

x：监控在此状态下生效

1：通道位于复位状态
2：通道中断
3：通道生效
4：在“自动”方式中断期间“手动”方式下的通道中断
5：在“MDA”方式中断期间“手动”方式下的通道中断
6：在“MDA”方式中断期间 MDA 中“手动”方式下的通道中断
7：MDA 中“手动”方式下的通道有效

11.2.5 各种运行方式下的锁定功能

锁定功能概述

在不同运行方式下，可以激活不同锁定功能。

在哪些运行方式下，哪些运行状态中可选择哪些锁定功能，可参见下表。

运行方式		自动				JOG							MDA				
功能		1	2	3		1	3	4	3	5	3		1	2	3	6	7
一般锁定功能																	
808D 就绪		x	x	x		x	x	x	x	x	x		x	x	x	x	x
禁止方式转换		x	x	x		x	x	x	x	x	x		x	x	x	x	x
通道专用锁定功能																	
进给停止				x			x		x		x				x	x	x
禁止 NC 启动		x	x	x		x	x	x	x	x	x		x	x	x	x	x
读入禁止		x	x	x		x	x	x	x	x	x		x	x	x	x	x
轴专用锁定功能																	
主轴禁止		x	x	x		x	x	x	x	x	x		x	x	x	x	x
伺服禁止		x	x	x		x	x	x	x	x	x		x	x	x	x	x
进给轴禁止		x	x	x		x	x	x	x	x	x		x	x	x	x	x
主轴专用的锁定功能																	
伺服禁止		x	x	x		x	x	x	x	x	x		x	x	x	x	x
主轴禁止		x	x	x		x	x	x	x	x	x		x	x	x	x	x

x：锁定功能可以在此状态下激活

1：通道位于复位状态
2：通道中断
3：通道生效
4：在“自动”方式中断期间“手动”方式下的通道中断
5：在“MDA”方式中断期间“手动”方式下的通道中断
6：在“MDA”方式中断期间 MDA 中“手动”方式下的通道中断
7：MDA 中“手动”方式下的通道有效

11.3 加工零件程序

11.3.1 程序运行和零件程序的选择

定义

所谓零件程序的执行，是指在“AUTO”方式下加工零件程序，或者在“MDA”方式下加工程序段。

通道控制

程序执行时，PLC 可以通过接口信号控制加工过程。控制可以通过运行方式专用的接口信号进行，或者通过通道专用的接口信号进行。

通过接口信号通道把当前的程序运行状态通知给 PLC。

选择

通道只有处于复位状态时才可以进行零件程序的选择。

可以按如下方式对程序进行选择：

- 操作（加工/程序管理操作区域）
- PLC
  - 通过“程序列表”中程序编号选定一个程序（参见 SINUMERIK 808D ADVANCED 编程和操作手册）
  - 通过 PLC-HMI 接口重新选定一个有效程序（参见章节“HMI 到 PLC 的信号 (页 19)”）

11.3.2 启动零件程序或者零件程序段

启动指令，通道状态

由通道专用的接口信号“NC 启动”（DB3200.DBX0007.1）启动程序，通常情况下该接口信号由如下硬键控制。



只有在“AUTO”和“MDA”方式下才执行启动指令。此时通道必须处于“通道复位状态”（DB3300.DBX0003.7）或“通道中断状态”（DB3300.DBX0003.6）。

必需的信号状态

选择的零件程序只可以通过启动指令使能信号开始执行。可以使用以下使能信号：

接口信号“808D 就绪”（DB3100.DBX0000.3）	必须被设置
接口信号“激活程序测试”（DB3200.DBX0001.7）	不应被设置
接口信号“NC 启动锁定”（DB3200.DBX0007.0）	不应被设置
接口信号“程序段结束 NC 停止”（DB3200.DBX0007.2）	不应被设置
接口信号“NC 停止”（DB3200.DBX0007.3）	不应被设置
接口信号“NC 停止进给轴和主轴”（DB3200.DBX0007.4）	不应被设置
接口信号“急停”（DB2700.DBX0000.1）	不应被设置
进给轴/NCK 报警	不应出现

指令执行

零件程序或零件程序段自动执行，同时接口信号“通道有状态”（DB3300.DBX0003.5）以及“程序运行状态”（DB3300.DBX0003.0）被设置。

程序一直运行下去，直至到达程序结束，或者通道通过停止指令或复位指令中断、终止。

中断

如未满足此前提条件，启动指令无效。并同时出现下述报警之一：10200、10202、10203

11.3.3 零件程序中断

通道状态

只有在其通道处于“通道有效”（DB3300.DBX0003.5）时才可以执行停止指令。

停止指令

有各种指令可以中断程序的执行并使通道状态置为“中断”。

- 接口信号“程序段结束 NC 停止”（DB3200.DBX0007.2）
- 接口信号“NC 停止”（DB3200.DBX0007.3）
- 接口信号“NC 停止进给轴和主轴”（DB3200.DBX0007.4）

- 接口信号“单段方式”（DB3200.DBX0000.4）
- 编程指令“M0”或“M1”以及相应的使能

指令执行

在执行了停止指令之后，接口信号“程序状态：停止”（DB3300.DBX0003.2）和“程序状态：中断”（DB3300.DBX0003.6）被设置。通过重新给出一个新的启动指令，可以使中断的零件程序从中断处恢复执行。

通常，给出停止指令后将进行如下的动作：

- 在下一个程序段交界处停止零件程序的执行(在程序段结束，M0/M1 或单段时 NC 停止)；用其它停止指令时则立即停止程序的执行。
- 当前程序段在此时刻尚未输出的辅助功能不再输出。
- 零件程序停止执行，进给轴停动。
- 程序段指针停留在中断位置。

11.3.4 复位指令

功能

在任一个通道状态都可以执行复位指令（信号“复位”(DB3000.DBX000.7)）。 该指令不可以被其它的指令中断。

通过复位指令可以终止一个零件程序的执行，或者一个零件程序段(MDA 方式)的执行。 执行复位指令后，接口信号“通道复位状态”(DB3300.DBX0003.7)和“程序状态：终止”(DB3300.DBX0003.4)被设置。

零件程序不能在中断点继续运行。 通道中所有的进给轴处于准停状态。

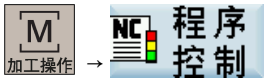
复位指令发出后有如下动作：

- 零件程序运行立即停止。
- 进给轴被制动，必要时主轴也被制动。
- 当前程序段在此时刻尚未输出的辅助功能不再输出。
- 程序段指针返回到零件程序的起始处。
- 除了上电消除的报警(POWER ON)，其它所有报警均从显示器中清除。

11.3.5 程序控制

选择/激活

用户可以通过操作界面对零件程序的执行进行控制。在“AUTO”运行方式下，可以通过操作界面选择某些特定功能，其中某些功能作用于 PLC 的接口信号。



这些接口信号仅仅作为操作界面的选择信号。它们并未激活所选择的功能。

为了使能选择的功能，PLC 用户程序的信号状态必须传送到数据块的另一个区中。由 PLC 进行控制时必须直接设置这些信号（见下表）。

功能	选择信号	使能信号	反馈信号
SKP 程序段跳跃	DB1700.DBX0001.0	DB3200.DBX0002.0	
DRY 空运行进给	DB1700.DBX0000.6	DB3200.DBX0000.6	
ROV 快速移动进给率修调	DB1700.DBX0001.3	DB3200.DBX0006.6	
预选： SBL - 单段粗 SBL - 单段精 单程序段	 - - 用户专用	 - - DB3200.DBX0000.4	
M1 编程的程序停止	DB1700.DBX0000.5	DB3200.DBX0000.5	DB3300.DBX0000.5
PRT 程序测试	DB1700.DBX0000.7	DB3200.DBX0001.7	DB3300.DBX0001.7

11.3.6 程序状态

程序状态

对于某个确定的通道，所选择的程序状态将在“AUTO”和“MDA”运行方式中显示。如果要在程序停止时转换到“JOG”方式，则显示“中断”或者在复位时，显示“终止”程序状态。

数控系统上有如下程序状态：

- 接口信号“程序状态终止”(DB3300.DBX0003.4)
- 接口信号“程序状态中断”(DB3300.DBX0003.3)
- 接口信号“程序状态停止”(DB3300.DBX0003.2)
- 接口信号“程序状态运行”(DB3300.DBX0003.0)

指令/信号的作用

通过不同的指令或激活不同的接口信号可以改变程序状态。下表中列出了通过这些指令所产生的程序状态(假定在指令或信号到来之前的状态为 –> 程序运行状态)。

指令	程序运行状态			
	终止	中断	停止	运行
接口信号 “复位”	X			
信号 “NC 停止”			X	
信号 “程序段结束 NC 停止”			X	
信号“NC 停止进给轴和主轴”			X	
信号“禁止读入”				X
信号“通道专用进给停止”				X
信号“轴专用进给停止”				X
进给倍率 = 0%				X
信号“主轴停止”				X
程序段中 M2 指令	X			
程序段中 M0/M1			X	
信号“单段方式”			X	
向 PLC 输出辅助功能，但尚未响应			X	

11.3.7 通道状态

通道状态

通道的当前状态可以通过接口信号显示出来。通过这一状态显示，用户可以利用 PLC 释放反应或闭锁。在所有的运行方式下均显示通道状态。

通道状态有以下几种：

- 接口信号“通道状态复位”(DB3300.DBX0003.7)
- 接口信号“通道状态中断”(DB3300.DBX0003.6)
- 接口信号“通道状态有效”(DB3300.DBX0003.5)

指令/信号的作用

通过不同的指令或激活不同的接口信号可以改变通道状态。下表中列出了通过这些指令所产生的通道状态(假定在指令或信号到来之前的状态为→通道状态有效)。

在执行零件程序时，或者在“JOG”方式下进给轴运行时处于“通道有效状态”。

下表给出了对通道状态的影响：

指令	通道状态		
	复位	中断	有效
接口信号“复位”	X		
接口信号 “NC 停止”		X	
接口信号 “程序段结束 NC 停止”		X	
接口信号“NC 停止进给轴和主轴”		X	
接口信号“禁止读入”			X
接口信号“通道专用进给停止”			X
接口信号“轴专用进给停止”			X
进给倍率 = 0%			
接口信号“主轴停止”			X
程序段中 M2 指令	X		
程序段中 M0/M1		X	
接口信号“单段方式”		X	
向 PLC 输出辅助功能，但尚未响应			X

11.3.8 事件控制的程序调用

应用

在出现特定事件时隐含式调用某个用户程序，从而使用户可以通过零件程序指令进行功能的初始设置或者初始化功能。

说明

关于该用户程序的调用，软件版本 4.6（本手册所述）与软件版本 4.4 不相兼容。

事件选择

在 MD20108 PROG_EVENT_MASK（事件控制的程序调用）中可以设定，下列哪些事件可以激活用户程序：

- 位 0 = 1:零件程序开始
- 位 1 = 1:零件程序结束
- 位 2 = 1:复位操作面板
- 位 3 = 1:引导启动（NC 数控系统的引导启动）

启动事件的询问

在用户程序中可以通过系统变量 $P_PROG_EVENT 询问，通过哪个事件可以激活零件程序。

事件

零件程序开始

下表给出了启动零件程序的顺序：

过程	指令	边界条件 （必须在指令发出前满足）	附注
1	选择通道：“激活”状态 选择运行方式：AUTO 或者 AUTO 和溢存储或者 MDA	无	选择通道和运行方式
2	NC 启动	无	NCK 启动
3	MD20112 START_MODE_MASK	带分析 的初始化顺序	
4	/_N_CMA_DIR/CYCPE1MA.SPF 和 /_N_CMA_DIR/CYCPE_MA.SPF	作为子程序	作为子程序隐含调用路径名

过程	指令	边界条件 （必须在指令发出前满足）	附注
5		无	作为主程序处理数据部分
6		无	作为主程序处理程序部分

事件

零件程序结束

下表给出了零件程序结束时的处理顺序：

过程	指令	边界条件 （必须在指令发出前满足）	附注
1	选择通道：“激活”状态 选择运行方式：AUTO 或者 AUTO 和溢存储或者 MDA	无	选择通道和运行方式
2	NC 启动	带零件程序结束的程序段	切入程序段
3	MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE	数控系统激活： 带分析的复位顺序	
4	/_N_CMA_DIR/CYCPE1MA.SPF 和 /_N_CMA_DIR/CYCPE_MA.SPF	作为 ASUP	作为 ASUP 隐含调用路径名
5	MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE	数控系统激活： 带分析的复位顺序	通过机床数据继续设置 G 代码复位位置

事件

复位操作面板

下表给出了复位操作面板时的处理顺序：

过程	指令	边界条件 （必须在指令发出前满足）	附注
1	选择通道和运行方式： 任意	初始状态：任意运行方式和通道状态	运行方式/通道状态 关 选择任意状态
2	复位		
3	MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE	数控系统激活： 带分析的复位顺序	
4	/_N_CMA_DIR/CYCPE1MA.SPF 和 /_N_CMA_DIR/CYCPE_MA.SPF	作为 ASUP	作为 ASUP 隐含调用路径名
5	MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE	数控系统激活： 带分析的复位顺序	通过机床数据继续设置 G 代码复位位置

事件

Startup（开机调试）

下表给出了引导启动时的处理顺序：

过程	指令	边界条件 （必须在指令发出前满足）	附注
1	复位	引导启动后	
2	MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE	数控系统激活 引导启动后： 带分析的复位顺序	
3	/_N_CMA_DIR/CYCPE1MA.SPF 和 /_N_CMA_DIR/CYCPE_MA.SPF	作为 ASUP	作为 ASUP 隐含调用路径名

过程	指令	边界条件 （必须在指令发出前满足）	附注
4	MD20110 RESET_MODE_MASK, MD20150 GCODE_RESET_VALUES, MD20152 GCODE_RESET_MODE	数控系统激活： 带分析的复位顺序	通过机床数据继续设置 G 代码复位位置

说明

必须将制造商循环 CYCPE1MA.SPF 和 CYCPE_MA.SPF 放入文件夹 **CMA.DIR**。

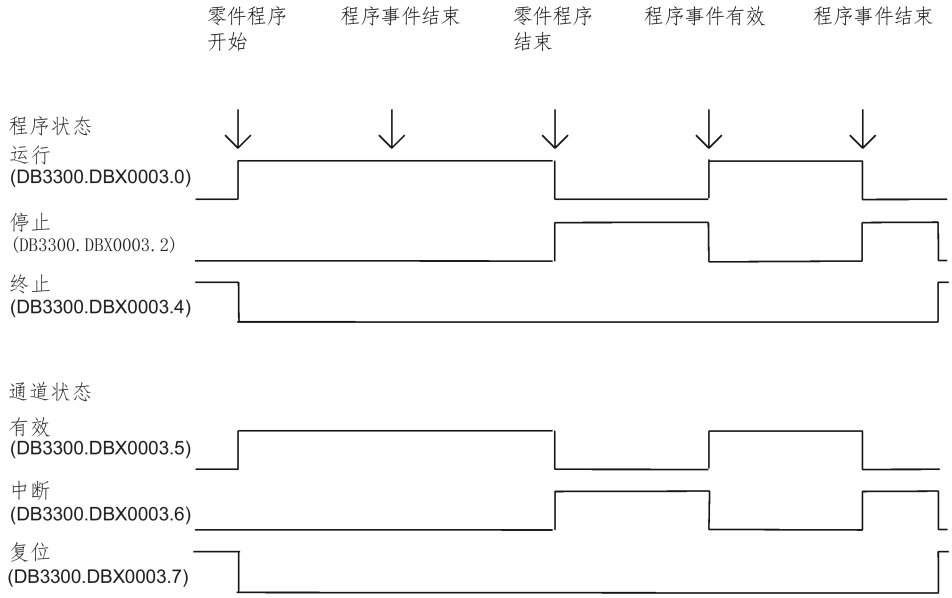
这两个制造商循环即为相应的跳转标记；因而在 PROG_EVENT.SPF 开始时跳转到 CYCPE1MA，在 PROG_EVENT.SPF 结束时跳转到 CYCPE_MA。

时序过程

零件程序开始和零件程序结束时：

处理带事件控制的零件程序调用中，零件程序启动和结束时 VDI 信号 DB3300.DBB0003（“程序状态”和“通道状态”）的时序变化：

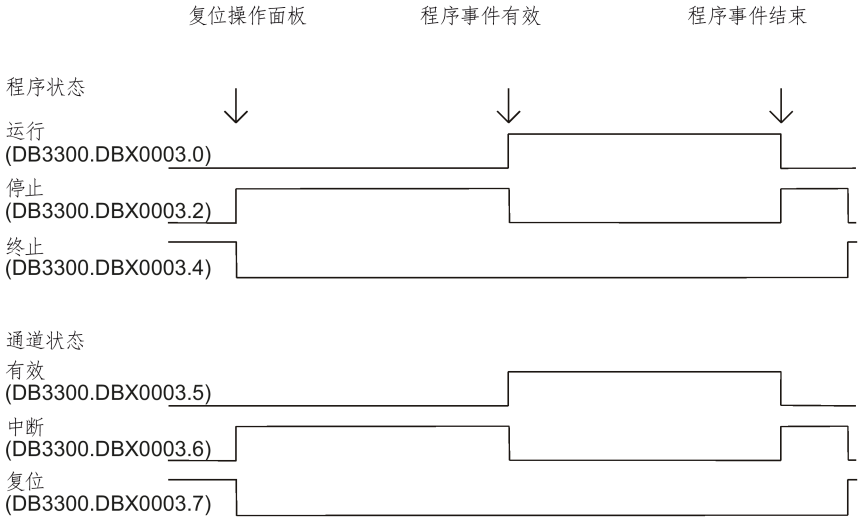
程序状态和通道状态的接口信号的时序(1)：



复位操作面板：

处理带事件控制的零件程序调用中，VDI 信号 DB3300.DBB0003（“程序状态”和“通道状态”）的时序变化：

程序状态和通道状态的接口信号的时序(2)：



说明

CYCPE1MA.SPF 和 CYCPE_MA.SPFF 结束后，才开始接收信号 DB3300.DBX0003.4（“程序状态中断”）和 DB3300.DBX0003.7（“通道状态复位”）。

从程序结束到程序事件开始的这段时间内，既不接收 DB3300.DBX0003.4（“程序状态中断”），也不接收 DB3300.DBX0003.7（“通道状态复位”）。

在复位操作面板和启动程序事件期间也同样如此。

需要注意的地方

用户程序 CYCPE1MA.SPF 和 CYCPE_MA.SPF 必须注意以下几点：

- 如果以最低的优先级执行，则可以由用户 ASUP 中断。
- 通过用户 M 功能 PLC 可以收到 CYCPE1MA.SPF 和 CYCPE_MA.SPF 处理状态的信息。
- 触发事件可以通过 PLC 程序在**接口** 上确定：

  DB3300.DBB4004 包含以下信息：
  0 没有激活的事件

  位 0 = 1 从通道状态 RESET 开始启动零件程序

  位 1 = 1 零件程序结束

  位 2 = 1 复位操作面板

  位 3 = 1 引导启动

  位 4 = 1 搜索后的首次启动

  位 5-7 预留，当前始终为 0

  发至 0 的总询问可以确认是否存在某个事件。如果通过复位使正在运行的事件消失，也会删除接口中相应的显示位。如果事件很短，则相应的位至少保留一个完整的 PLC 循环时间。
- 每次重新设计 MD20108 PROG_EVENT_MASK 后必须载入或者使能 /_N_CMA_DIR/CYCPE1MA.SPF 和 /_N_CMA_DIR/CYCPE_MA.SPF。否则将发出报警 14011 “不存在程序 _N_PROG_EVENT_SPF 或者未使能加工”。
- 可以通过 PROC 指令中 DISPLOF 属性抑制当前程序段中的显示。
- 而单程序段停止可以通过指令 SBLOF 属性或通过
  MD10702 IGNORE_SINGLEBLOCK_MASK（阻止单程序段停止）位 0 关闭。

  读入锁定和单程序段加工的响应可以通过机床数据 MD20106 PROG_EVENT_IGN_SINGLEBLOCK（程序事件忽略单程序段）和 MD20107 PROG_EVENT_IGN_INHIBIT（程序事件忽略读入锁定）分别控制。

  MD20106 PROG_EVENT_IGN_SINGLEBLOCK:
  当出现以下情况时，虽然没有继续启动单程序段，CYCPE1MA.SPF 和 CYCPE_MA.SPF 也会转换程序段：

  设置了位 0 = 1，在零件程序启动事件后

  设置了位 1 = 1，在零件程序结束事件后

  设置了位 2 = 1，在复位操作面板事件后

  设置了位 3 = 1，在引导启动事件后

  设置了位 4 = 1，在程序段搜索完成后第一次启动事件后

  MD 20107:PROG_EVENT_IGN_INHIBIT:

  当出现以下情况时，即使读入锁定，CYCPE1MA.SPF 和 CYCPE_MA.SPF 也会转换程序段：

  设置了位 0 = 1，在零件程序启动事件后

  设置了位 1 = 1，在零件程序结束事件后

  设置了位 2 = 1，在复位操作面板事件后

  设置了位 3 = 1，在引导启动事件后

  设置了位 4 = 1，在程序段搜索完成后第一次启动事件后

  位 0 == 1（零件程序启动后的程序事件）时，下列限制条件生效：
  如果使用零件程序指令“RET”结束程序事件，则 RET 始终引导至可执行的程序段（和 M17 类似）。

  位 0 == 0 时没有新响应，即：RET 在编译器中编译并不会引导至“可执行的程序段”。

零件程序启动/结束的顺序没有完成：

- 如果从复位状态启动一个用户 ASUP，则上述说明的用于零件程序启动/结束的事件顺序不会完成。
- **可设定的程序事件属性**

  在机床数据 MD20109 PROG_EVENT_MASK_PROPERTIES 中可以设置各个通道“事件控制的程序调用”的其他属性。

– 位 0 = 0:当前，从通道状态“复位”中启动一个 ASUP 会引起“事件控制的程序调用”

– 位 0 = 1:从通道状态“复位”中启动一个 ASUP 不会引起“事件控制的程序调用”

零件程序启动时：
/_N_CMA_DIR/CYCPE1MA.SPF 和 /_N_CMA_DIR/CYCPE_MA.SPF 作为子程序被执行。CYCPE1MA.SPF 和 CYCPE_MA.SPF 必须以 M17 或 RET 结束。不允许通过 REPOS 指令返回，否则将导致报警 16020 “不允许再定位”。

复位操作面板时或者**引导启动后**的故障：
如果在复位操作面板时或者引导启动后仍出现急停或者某个操作方式/NCK 故障，则在应答急停或者应答故障后才在通道内处理 CYCPE1MA.SPF 和 CYCPE_MA.SPF。

示例

MD20106 PROG_EVENT_IGN_SINGLEBLOCK = 'H1F'

MD20107 PROG_EVENT_IGN_INHIBIT = 'HC'

MD20109 PROG_EVENT_MASK_PROPERTIES = 'H1'

事件程序

通过所有事件调用的示例

MD20108 PROG_EVENT_MASK = 'H0F'（事件控制的程序调用），
即：在零件程序启动、结束或复位操作面板和引导启动时调用 CYCPE1MA.SPF 和 CYCPE_MA.SPF：

零件程序开始时的处理

```
IF ($P_PROG_EVENT == 1)
N 10 R100 = 0                    ;    加工循环的传输参数
N 20 M17
ENDIF
```

零件程序开始和 **复位操作面板**时的处理

```
IF ($P_PROG_EVENT == 2) OR ($P_PROG_EVENT == 3)
N10 R20 =5
N20 ENDIF
N30 M17
ENDIF
```

引导启动时的处理

```
IF ($P_PROG_EVENT == 4)
N10 $SA_SPIND_S[Ax4] = 0          ;    主轴启动的转速，通过 VDI
N20 ENDIF
N30 M17
ENDIF
M17
```

使用“复位”键的启动

按下复位键会自动启动以下零件程序：

- /_N_CMA_DIR/CYCPE1MA.SPF
- /_N_CMA_DIR/CYCPE_MA.SPF

通过 MD20107 PROG_EVENT_IGN_INHIBIT 控制

如果存在下列机床数据设置：
MD20107 PROG_EVENT_IGN_INHIBIT= 'H04F'
MD20108 PROG_EVENT_MASK= 'H04F'

则使用“复位”键启动的程序将完全处理，直至结束，而不受设置的读入锁定影响。

说明

程序段搜索时(SSL) MD11450 的推荐设置

MD11450 SEARCH_RUN_MODE = 'H7'（搜索参数设置）

位 0 = 1:
程序段搜索后转入最后动作程序段，处理停止并且设置 VDI 信号"最后动作程序段生效"。只有当 PLC 通过设置 VDI 信号"PLC 动作结束"发出请求时，才生成报警 10208。
应用：程序段搜索完成后 PLC 启动 ASUP。

位 1 = 1：
输出动作程序段后自动启动 ASUP。当 ASUP 结束后才生成报警 10208。

位 2 = 1:
抑制动作程序段中辅助功能的输出。将来，例如：在 ASUP 中，可以输出在程序段搜索中搜集的主轴编程。

程序数据保存在以下系统变量中：

- $P_SEARCH_S
- $P_SEARCH_SDIR
- $P_SEARCH_SGEAR
- $P_SEARCH_SPOS
- $P_SEARCH_SPOSMODE

11.3.9 异步子程序 (ASUP)

功能

通过 ASUP 接口范围可以由 PLC 激活两个不同的 ASUP（PLCASUP1_SPF 或者 PLCASUP2_SPF）。PLC 能够启动一个异步子条件(ASUP)的前提条件是，该程序已经通过 NC 程序或者通过 PI 通讯 ASUP 分配了一个中断号（参见 DB1200.DBB4000）。

该类型的 ASUP 可以由 PLC 在任意时刻启动。ASUP 可以中断运行的 NC 程序。

某一时刻只能启动一个 ASUP。如果需要在一个 PLC 循环中两个 ASUP 都设置为逻辑 1，则按照 INT1，然后 INT2 的顺序启动 ASUP。

当 ASUP 结束或者出错时，用户必须将启动信号设置为逻辑 0。

本数控系统提供了两个默认 ASUP 供 PLC 调用。其中 ASUP1 用于手动换刀，ASUP2 用于通过手动机床功能在车床上进行工件的手动加工。

您也可根据需要使用自定义的 ASUP。如要使用自定义 ASUP，必须先将程序（PLCASUP1.SPF 和 PLCASUP2.SPF）放在制造商循环目录（N:\CMA）下，然后将"PI 索引"（DB1200.DBB4001）设为 1（ASUP1）/2（ASUP2）。

说明

在启动 ASUP 前必须结束 PI 通讯 ASUP 的调用。

初始化

通过 PI 通讯 ASUP 进行初始化。

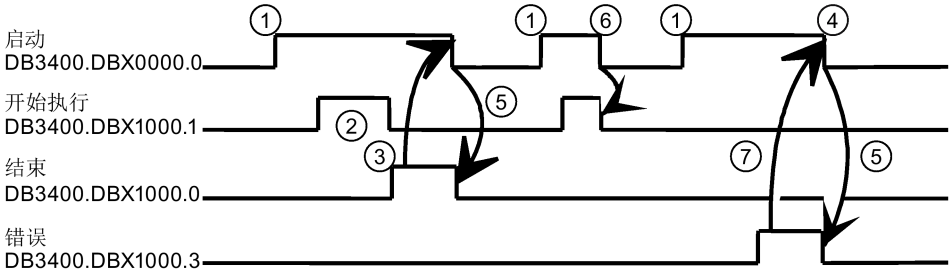
启动 ASUP

在下面的脉冲示意图中，举例显示了 PLCASUP1.SPF 的一个 ASUP 的时间运行过程。参考表格可识别出哪些 NST 与 PLCASUP2.SPF 是相关的。

下表介绍了如何分配信号与脉冲示意图：

信号	地址 - PLCASUP1_SPF	地址 - PLCASUP2_SPF
启动	DB3400.DBX0000.0	DB3400.DBX0001.0
开始执行	DB3400.DBX1000.1	DB3400.DBX1001.1
结束	DB3400.DBX1000.0	DB3400.DBX1001.0
错误	DB3400.DBX1000.3	DB3400.DBX1001.3
未分配中断编号	DB3400.DBX1000.2	DB3400.DBX1001.2

PLCASUP1_SPF 的脉冲示意图：



① 通过启动的上升沿触发功能
② 执行 ASUP
③ 积极应答：ASUP 结束
④ 收到应答后复位功能启动
⑤ 通过 PLC 选择信号
⑥ 不允许。如果在收到应答前复位功能触发，则不更新输出信号，并且对触发的功能运行不产生影响。
⑦ 消极应答：出错

配置

通过以下标准机床数据可以调节 ASUP 的属性：

- MD11602 ASUP_START_MASK（忽略 ASUP 的停止原因）
  该机床数据用于确定 ASUP 启动时忽略哪些停止原因。
  推荐设置：MD11602 = 'H7'
- MD11604 ASUP_START_PRIO_LEVEL（MD11602 开始生效的优先级）
  该机床数据用于确定，从哪个 ASUP 优先级开始可以使用机床数据 MD11602 ASUP_START_MASK。从此处给定的优先级到最高 ASUP 优先级 1 都将考虑到 MD11602 。
  推荐设置：MD11604 = 2
- MD20116 IGNORE_INHIBIT_ASUP（虽然锁定读入，仍处理中断程序）
  虽然设置了读入锁定，但仍为位已设置的中断通道完整处理分配的用户 ASUP。
  位 0 分配给中断通道 1 (PLCASUP1)
  位 1 分配给中断通道 2 (PLCASUP2)
  该机床数据只在 MD11602 ASUP_START_MASK 位 2 = 0 时生效
- MD20117 IGNORE_SINGLEBLOCK_ASUP（虽然为单程序段，但仍完整处理中断程序）
  虽然设置了单程序段处理，但仍通过设置位为中断通道
  完整处理分配的用户 ASUP。
  位 0 分配给中断通道 1 (PLCASUP1)
  位 1 分配给中断通道 2 (PLCASUP2)
  此机床数据只在
  MD10702 IGNORE_SINGLE_BLOCK_MASK 位 1 = 0 时有效

11.3.10 操作或程序动作的反应

反应

下表列出了在某些操作或程序执行后所产生的通道和程序的状态。

表中的左边部分是通道和程序状态，以及初始状态的运行方式，必须在其中选出输入状态。表格的右边部分为一些操作/程序的作用。当相应的动作执行以后，所产生的状态由括号中的状态号表示。

状态	通道状态			程序状态				有效操作方式			操作或程序动作 (动作后状态)
	R	U	A	N	U	开关设备保护	A	A	M	J	
1		x					x	x			复位(4)
2		x					x		x		复位(5)
3		x					x			x	复位(6)
4	x			x				x			NC 启动(13);方式转换 (5 或 6)
5	x			x					x		NC 启动(14);方式转换 (4 或 6)
6	x			x						x	方向键(15);方式转换 (4 或 5)
7		x		x					x		NC 启动(14)
8		x		x						x	NC 启动(15)
9		x			x			x			NC 启动(13);方式转换 (10 或 11)
10		x			x				x		NC 启动(16);方式转换 (9 或 11)
11		x			x					x	方向键(17);方式转换 (9 或 10)
12		x				x		x			NC 启动(13);方式转换 (10 或 11)
13			x				x	x			NC 停止(12)
14			x	x					x		NC 停止(7);程序段结束(5)
15			x	x						x	NC 停止(8);JOG 终点(6)
16			x		x				x		NC 停止(10);程序段结束(10)
17			x		x					x	NC 停止(11);JOG 终点(11)

说明

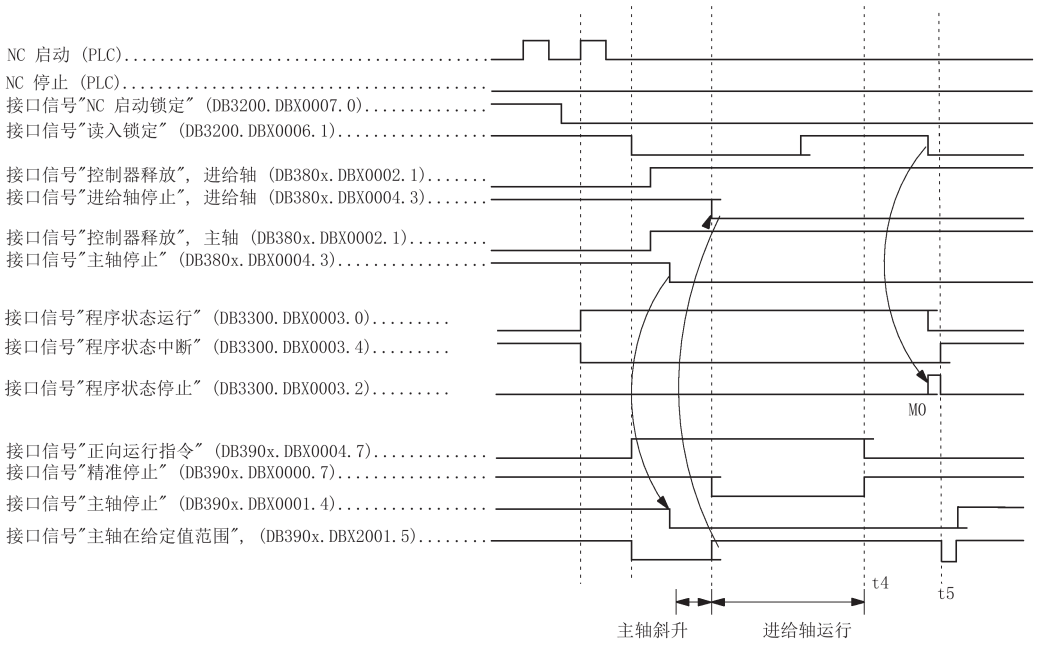
通道状态：
R:中断
U:中断
A:运行

程序状态：
N:中断
U:中断
S:停止
A:运行

运行方式：
A:自动
M:MDA
J:JOG

11.3.11 程序运行时间图示例

程序执行过程中信号举例：



说明：
通过PLC 用户程序生成的控制输入信号，
t4: 通过"读入锁定"阻止程序段切换到 N20，
t5: 使用 RESET 中断程序

程序：
N10 G01 G90 X100 M3 S1000 F1000 M88 N20 M0

11.4 程序测试

11.4.1 程序测试概述

用途

系统具有多种测试功能用来对一个新的零件程序进行测试和运行。 使用测试功能可以减少机床测试阶段运行时发生碰撞的可能性，并可以节省大量的时间。 多个程序测试功能可以同时生效。

在此对以下测试方法给予说明：

- 程序运行时无进给轴动作
- 程序单段运行
- 空运行进给程序运行
- 处理特定程序段
- 程序段跳跃
- 图形模拟

11.4.2 程序运行时无进给轴动作(PRT)

功能

在“程序测试”功能激活的情况下，可以通过接口信号“NC 启动”(DB3200.DBX0007.1)启动并执行 零件程序，并可以输出辅助功能，执行停留时间。只有进给轴和主轴模拟运行。 软件限位开关的安全保护功能也有效。

此时位置控制并没有中断，因此在测试功能关闭后无需使进给轴回参考点。

用户可以由此监控编程的进给轴位置以及辅助功能的输出情况。

说明

不带轴运行的程序处理也可以与“空运行进给”功能一起激活。

选择/激活

通过 HMI 上的如下软键来选择该功能：



此时设置了接口信号“选择程序测试”(DB1700.DBX0001.7)。

通过接口信号“激活程序测试”(DB3200.DBX0001.7)由 PLC 用户程序激活程序测试功能。

显示

程序测试功能激活后反馈信号显示在屏幕中状态栏“PRT”下，同时在 PLC 中设置接口信号“程序测试有效”(DB3300.DBX0001.7)。

11.4.3 程序单段运行(SBL)

功能

用户可以使用此功能逐段执行零件程序并检查各个加工步骤。如果已执行的程序段正确， 可以要求执行下一程序段。 通过接口信号“NC 启动”(DB3200.DBX0007.1)可以启动下一个零件程序。

“单段运行”功能生效时，零件程序运行在每个程序段之后暂停。要注意所接通的单段方式。

单段类型

分为以下两种单段方式：

- 粗略单程序段
  在这种单段方式下所有具有加工功能的程序段(指具有轴运行，辅助功能输出等等)在运 行时均逐段执行。如果有刀具补偿指令(G41,G42)，则在每个插入的中间语句之后停止加工。但若是计算语句则不停止加工，因为计算语句不产生任何动作。
- 精准单程序段
  在这种单段方式下，零件程序**所有**程序段(包括没有运行功能的纯计算语句)均逐段地通过“NC 启动”启动执行。

上电后的缺省设置为“粗略单段方式”。

说明

对于 G33 语句，只有激活了“空运行进给”功能之后单段才有效。

选择/激活

可以直接通过机床控制面板选择单段运行方式。

必须通过接口信号“单段生效”(DB3200.DBX0000.4)从 PLC 用户程序激活此功能。

在“程序控制”窗口中预先选择“粗略单段方式”或“精单段方式”。

显示

单段运行功能生效后，作为反馈信号会在屏幕中相应区域显示“SBL”。

一旦因为设置了单程序段运行而处理一个零件程序段，则：

- 设置以下接口信号：
  - 接口信号“通道状态复位”(DB3300.DBX0003.6)
  - 接口信号“程序状态停止”(DB3300.DBX0003.2)
- 复位以下接口信号：
  - 接口信号“通道状态有效”(DB3300.DBX0003.5)
  - 接口信号“程序状态运行”(DB3300.DBX0003.0)

11.4.4 按照空运行(DRY)进给处理程序

功能

零件程序可以通过接口信号“NC 启动”(DB3200.DBX0007.1)启动。空运行进给功能激活后，在 G1，G2，G3，CIP,CT 指令下编程的速度将被设定数据 SD42100 DRY_RUN_FEED 中的设定值进给率代替。该值同样也代替用 G95 编程的旋转进给率。但是，如果编程的进给率值大于空运行进给率的值，则使用较大值。

注意
工件或刀具损坏 在“空运行进给”功能生效的情况下不得进行工件的加工，因为由于进给率的变化可能会 超出刀具的切削速度而导致工件或机床受损。

选择/激活

在“AUTO”运行方式下进行如下操作，即选择了空运行进给功能：



此时设置接口信号“选择空运行进给”(DB1700.DBX0000.7)。此外还必须在 HMI 上按下如下软键输入所要求的空运行进给率数值：



但此时该功能还没有激活。

通过接口信号“激活空运行进给”(DB3200.DBX0000.4)激活此功能并使用 NC 启动来执行。

程序执行前，必须在 SD42100 DRY_RUN_FEED 中设定空运行进给率。

显示

作为“空运行进给有效”的反馈信息，在屏幕状态栏中显示“DRY”。

11.4.5 程序段搜索:处理特定程序段

功能

如果只需执行一个特定的程序段（目标程序段），可以使用程序段搜索功能。在程序段搜索过程中，可以选择是否执行和通常程序运行方式下的相同的计算。

到达目标程序段后，可以通过接口信号“NC 启动”（给出 2 次）（DB3200.DBX0007.1）启动程序。必要时，执行进给轴到达目标程序段的起点和终点时的自动补偿动作。然后继续执行剩余程序。

说明

必须确保到达的起始位置无碰撞，并提供了相应的技术值以及相应的刀具已经使能！必要时，先使用 JOG 方式手动移动到无碰撞的起始位置。选择目标程序段时须考虑程序段的搜索类型。

选择/激活

在用户界面上的“AUTO”方式下可以选择程序段搜索功能。

使用不同的软键可以激活程序段搜索的不同功能：

- 使用轮廓计算搜索程序段
  这样可以在任何状态下接近轮廓。使用“NC 启动”移动到**目标程序段的起始位置**或者在目标程序段之前的程序段的终点位置。这将运行到终点位置。可以在轮廓处执行该程序段。
- 使用程序段终点计算搜索程序段
  这样可以在任何状态下接近目标位置（如刀具更换位置）。使用目标程序段中有效的插补类型到达**目标程序段的终点**或者下一个编程位置。该操作将不在轮廓处执行。只移动目标程序段中编程的进给轴。

- 不带计算搜索程序段
  用于在主程序中快速搜索。不执行任何计算。系统内部值和程序段搜索前保持不变。是否进行加工，取决于程序且必须由操作人员决定。此程序段搜索类型特别适合于对新程序进行快速语句检查。

接口信号

运行一段时间（见下图）后 PLC 中设置下列接口信号：

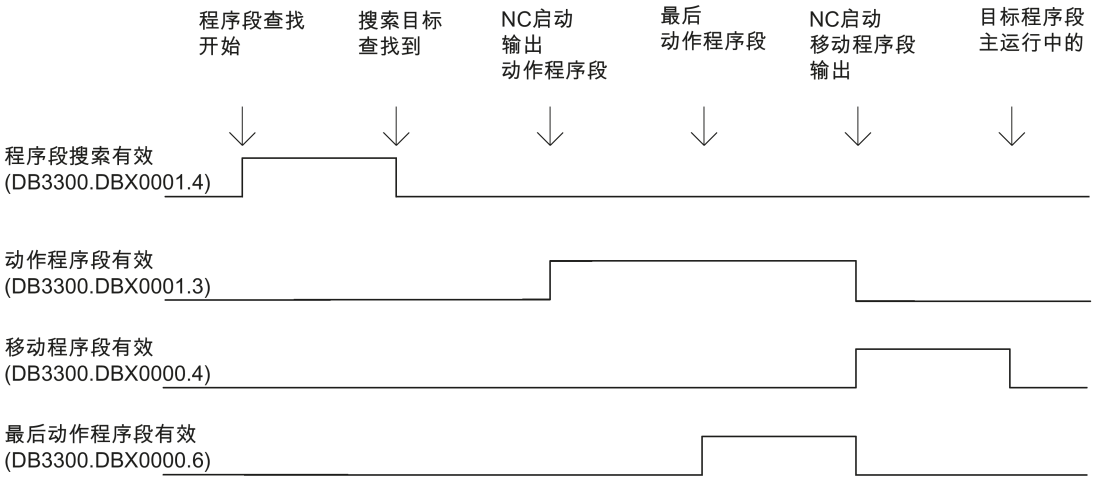
- “程序段搜索有效”（DB3300.DBX0001.4）
- “动作程序段有效”（DB3300.DBX0001.3）
- “移动程序段有效”（DB3300.DBX0000.4）

说明

接口信号“移动程序段有效”只在“使用轮廓计算搜索程序段”功能下设置，因为在“使用程 序段终点计算搜索程序段”功能中没有单段的移动程序段(移动程序段就是目标程序段)。

- “最后动作程序段有效”（DB3300.DBX0000.6）

接口信号的时间顺序



在信号“使用程序段终点计算搜索程序段”之后并且从“最后动作程序段有效”开始直到由 NC 启动继续执行零件程序过程中，不执行自动的重新定位。当前轴位置的起始点是 NC 启动位置，终点取决于零件程序的执行。

动作程序段

动作程序段包含了“带计算的程序段搜索”期间搜集的动作，例如：辅助功能输出、刀具(T, D)、主轴(S)、进给编程等。
在“带计算的程序段搜索”期间（轮廓或程序段终点），动作如 M 功能输出等归入上述动作程序段中。这些程序段的输出是在“找到程序段”之后由“NC 启动”来实现的。

说明

对于动作程序段，其中累计的主轴编程(S 值，M3/M4/M5，SPOS)也将有效。PLC 用户程序必须保证刀具可以使用且通过 NST“主轴复位”（DB380x.DBX0002.2）可以使主轴编程复位。

程序段搜索后的 PLC 动作

这是为了保证所有的动作程序段已执行并可以执行 PLC 或操作动作（如运行方式改变）。例如，PLC 在动作开始之前，可以更换刀具。

缺省时，此时将输出报警 10208。它告诉操作人员需要 NC 启动来继续执行程序。

附加条件

使用目标程序段中有效的插补类型来执行移动动作“使用程序段终点计算搜索程序”。通常应该是 G0 或 G1。使用其它插补类型时，移动动作将终止并输出报警（如使用 G2/G3 导致圆弧终点错误）。

说明

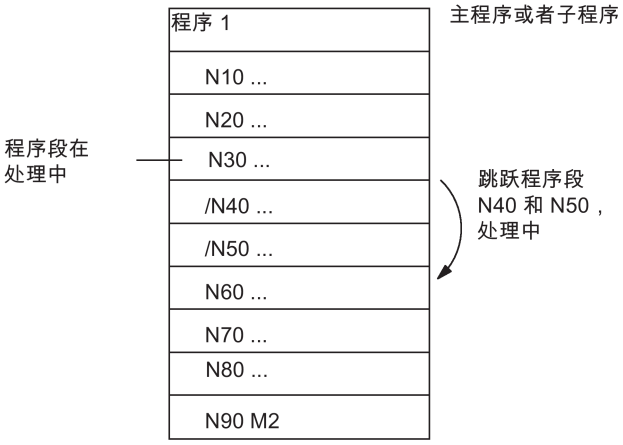
关于“程序段搜索”功能更多的信息，请参考 SINUMERIK 808D ADVANCED 编程和操作手册。

11.4.6 零件程序段跳跃(SKP)

功能

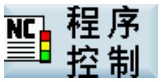
在测试或者运行新程序时，如果可以在程序运行时锁定或跳跃几个特定的程序段将会十分有益。

程序段跳跃：



选择/激活

在按下如下软键后进入的用户界面中选择该跳跃功能：



此时设置接口信号“选择程序段跳跃 “(DB1700.DBX0002.0)。在被跳跃的程序段之前必须要设置一斜线“/”（参图）。但此时该功能还没有激活。

通过接口信号“激活程序段跳跃”(DB3200.DBX0002.0)激活此功能。

显示

作为反馈信号，在程序段跳跃功能激活后将在屏幕状态栏中显示“SKP”。

11.4.7 图形模拟

功能

已选择和打开的程序可以在"AUTO"方式下，在系统的屏幕上进行图形模拟。激活 NC 启动后，编程的进给轴动作以断线图形格式记录。

选择/取消

执行如下操作载入程序并实现对所选程序的图形模拟：



→ 打开程序 →


此时，当退出程序编辑操作区域或在 HMI 上按下如下软键后，接口信号“模拟有效”（DB1900.DBX0000.6）被设置并再次复位：



显示

屏幕上显示了整个工件或只是工件的修改部分以及可以对其进行不同的操作选项。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

PLC 用户程序

在模拟过程中，PLC 用户程序本身必须执行系统所需的动作，例如：

- 通过过渡到程序测试停止进给轴/主轴：设置接口信号“激活程序测试”（DB3200.DBX0001.7）
- 如果设置信号“复位”（DB3000.DBX0000.7）等退出 功能，则当前运行程序被终止。



显示机床数据

用户专用的图形模拟时显示有各种机床数据（MD283 到 MD292）以供使用。

文献：

SINUMERIK 808D ADVANCED 参数手册

11.5 程序运行时间定时器

功能

“程序运行时间”功能提供了可用来监控程序中技术过程或在屏幕中显示的计时器。这些计时器为只读。

存在始终有效的计时器。其他计时器可以通过机床数据停用。

计时器 - 始终有效

- 自最后一次“系统以缺省值上电”起的时间（以分钟为单位）：
  $AN_SETUP_TIME
  每次“系统以缺省值上电”时，计时器自动归零。
- 自最后一次系统上电的时间（以分钟为单位）：
  $AN_POWERON_TIME
  数控系统每次上电后计时器自动归零。

可以取消的计时器

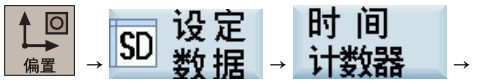
下列定时器通过机床数据来激活（缺省设定）。启动为计时器特定启动。当程序停止或进 给率修调为零，每个有效运行时间的测量将自动中断。

可以使用机床数据指定有效空运行进给率和程序测试的激活计时器行为。

- "AUTO"方式下 NC 程序的总执行时间（以秒为单位）
  $AC_OPERATING_TIME
  在“AUTO”方式下，从数控系统启动到程序结束/复位之间所有程序的运行时间累计值。系统每次上电后计时器自动设为零。
- 选择的 NC 程序的运行时间（以秒为单位）：
  $AC_CYCLE_TIME
  计算所选程序在 NC 启动和程序结束/复位之间的运行时间。当新的 NC 程序启动时，该定时器被删除。
- 刀具切削时间（以秒为单位）：
  $AC_CUTTING_TIME
  快速进给无效而刀具有效时，NC 启动和程序结束/复位之间，在所有 NC 程序中测得的进给轴运行时间。当暂停时间生效时，计算被中断。每次“系统以缺省值上电”时，计时器自动归零。

显示

执行如下操作后 ，显示计时器的内容：



总运行时间	0000 H 00 M 00 S
程序运行时间	0000 H 00 M 00 S
进给时间	0000 H 00 M 00 S
缺省设置后的启动时间	0002 H 18 M
冷启动时间	0002 H 18 M

- **总运行时间** = $AC_OPERATING_TIME
- **程序运行时间** = $AC_CYCLE_TIME
- **进给时间** = $AC_CUTTING_TIME
- **冷启动后的时间** = $AN_SETUP_TIME
- **热启动后的时间** = $AN_POWERON_TIME

程序运行时间也显示在加工操作区域的“AUTO”窗口中的信息行里。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

11.6 工件计数器

功能

“工件计数器”功能提供了可用于计算工件数量的计数器。这些计数器是通道专用的系统变量，可以通过程序或 HMI 来读/写。

可使用以下机床数据来设置计数器写入的保护等级：

- MD9020 USER_CLASS_WRITE_WPC_COUNT（工件计数器的写保护等级）
  该参数范围为 0 至 7，分别对应数控系统的八个保护等级。只有设置了高于或等于 MD9020 设定值的保护等级时，才能对计数器进行写入操作。

说明

工件计数器中数值的输入范围为 0 至 999999999。

按下以下硬键进行确认后，所有输入的数值方可生效：



通过下列通道专用的机床数据可以对计数器激活、归零时刻和计数算法产生影响。

- MD27880 PART_COUNTER（激活工件计数器）
- MD27882 PART_COUNTER_MCODE（使用用户定义的 M 指令进行工件计数）

计数器

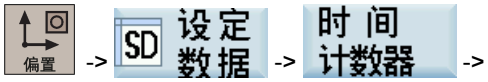
- 所需工件的数量（工件给定值）：
  $AC_REQUIRED_PARTS
  在此计数器中可以定义工件的个数，在到达这个数值之后，实际工件的个数 $AC_ACTUAL_PARTS 归零。
  MD27880 PART_COUNTER（位 0）可用来激活报警 21800 “到达工件给定值”和输出信号“到达工件给定值”（DB3300.DBX40001.1）。

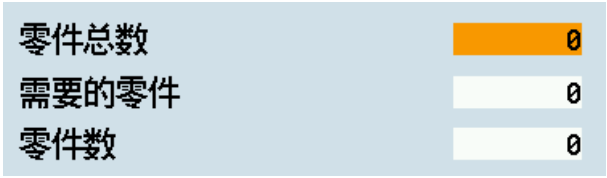
- 生产的工件总数（总的实际数量）：
  $AC_TOTAL_PARTS
  计数器给出所有自开始时刻起所生产的工件数量。
- 当前的工件数量（当前实际值）：
  $AC_ACTUAL_PARTS
  在这种计数器中记录自开始时刻起所生产的所有工件数量。当给定值到达时（$AC_REQUIRED_PARTS），计数器自动置为零（要求 $AC_ REQUIRED_PARTS 值不为 0）。
- 用户定义的工件数:
  $AC_SPECIAL_PARTS
  该计数器允许用户根据自定义来对工件计数。在与 $AC_REQUIRED_PARTS (工件给定值)一致时可以定义报警输出。用户必须将该计数器归零。

启动时，使用将计数器设为零后的第一个用于计数的 M 指令的输出。该 M 指令是为相应的计数器 MD27880 PART_COUNTER 或 MD27882 PART_COUNTER_MCODE 设置的。

显示

执行如下操作后 ，显示计数器的内容：





- **零件总数**= $AC_TOTAL_PARTS
- **需要的零件**= $AC_REQUIRED_PARTS
- **零件数**= $AC_ACTUAL_PARTS
  （$AC_SPECIAL_PARTS 不在屏幕中显示）

零件数也显示加工操作区域的“AUTO”窗口中的信息行里。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

11.7 数据表

11.7.1 机床数据

NC 专用的机床数据

序号	名称	名称
一般		
10702	IGNORE_SINGLEBLOCK_MASK	阻止单程序段停止
11450	SEARCH_RUN_MODE	编程搜索模式
11602	ASUP_START_MASK	忽略 ASUP 的停止原因
11604	ASUP_START_PRIO_LEVEL	ASUP_START_MASK 的优先级

通道的基础机床数据

序号	名称	名称
通道专用		
20050	AXCONF_GEOAX_ASSIGN_TAB[n]	配置几何轴给通道轴 [几何轴号]：0...2

序号	名称	名称
20060	AXCONF_GEOAX_NAME_TAB[n]	通道中的几何轴 [几何轴号]：0...2
20070	AXCONF_MACHAX_USED[n]	通道中有效机床进给轴号 [通道轴号]：0...4
20080	AXCONF_CHANAX_NAME_TAB[n]	通道中通道轴名称[通道轴号]：0...4
20100	DIAMETER_AX_DEF	几何轴具有进给轴功能
20106	PROG_EVENT_IGN_SINGLEBLOCK	程序事件忽略单程序段
20107	PROG_EVENT_IGN_INHIBIT	程序事件忽略读入锁定
20108	PROG_EVENT_MASK	事件控制的程序调用
20109	PROG_EVENT_MASK_PROPERTIES	程序事件的属性
20110	RESET_MODE_MASK	复位时的初始设置
20112	START_MODE_MASK	引导启动后 NC 启动和复位时的初始设置
20116	IGNORE_INHIBIT_ASUP	忽略读入锁定完整处理用户 ASUP
20117	IGNORE_SINGLEBLOCK_ASUP	忽略单程序段处理完整处理用户 ASUP
20700	REFP_NC_START_LOCK	不带参考点的 NC 启动禁用
21000	CIRCLE_ERROR_CONST	圆弧终点监控常量
20150	GCODE_RESET_VALUES	G 功能组的复位位置
20152	GCODE_RESET_MODE	复位时的 G 编码初始设置

通道的辅助功能设置

序号	名称	名称
通道专用		
22000	AUXFU_ASSIGN_GROUP[n]	辅助功能组[通道中的辅助功能号]：0...63
22010	AUXFU_ASSIGN_TYPE[n]	辅助功能类型 [通道中辅助功能号]:0...63
22020	AUXFU_ASSIGN_EXTENSION[n]	辅助功能扩展[辅助功能序号]:0...63
22030	AUXFU_ASSIGN_VALUE[n]	辅助功能数值[辅助功能序号]:0...63
22550	TOOL_CHANGE_MODE	使用 M 功能的新刀具偏移

通道的计时器和计数器

序号	名称	名称
通道专用		
27860	PROCESSTIMER_MODE	程序运行时间计算使能
27880	PART_COUNTER	工件计数器使能
27882	PART_COUNTER_MCODE[n]	工件计数使用 M 指令, n = 0 ... 2

显示机床数据

序号	名称	名称
283 ... 292		显示图形模拟设定值

11.7.2 设定数据

序号	名称	名称
通道专用		
42000	THREAD_START_ANGLE	螺纹加工起始角
42010	THREAD_RAMP_DISP	G33 螺纹加工时进给轴的加速和制动位移
42100	DRY_RUN_FEED	空运行进给率

11.7.3 接口信号

运行方式信号

序号	位	名称
PLC 到 NCK		
DB3000.DBX0000	.0	“AUTO”方式
DB3000.DBX0000	.1	“MDA”方式
DB3000.DBX0000	.2	“JOG”方式
DB3000.DBX0000	.4	禁止方式转换
DB3000.DBX0000	.7	复位
DB3000.DBX0001	.2	机床功能 REF
NCK 到 PLC		
DB3100.DBX0000	.0	有效方式：AUTO
DB3100.DBX0000	.1	有效方式：MDA
DB3100.DBX0000	.2	有效方式: JOG
DB3100.DBX0000	.3	808D 就绪
DB3100.DBX0001	.2	有效机床功能 REF

通道信号

序号	位	名称
PLC 到 NCK		
DB3200.DBX0000	.4	激活单个程序段
DB3200.DBX0000	.5	使能 M01
DB3200.DBX0000	.6	使能空运行进给
DB3200.DBX0001	.0	回参考点激活
DB3200.DBX0001	.7	激活程序测试
DB3200.DBX0002	.0	跳过程序段
DB3200.DBX0006	.0	进给锁定
DB3200.DBX0006	.1	读入禁止
DB3200.DBX0006	.2	删除剩余行程
DB3200.DBX0006	.3	删除子程序循环数
DB3200.DBX0006	.4	程序级终止
DB3200.DBX0006	.6	快速进给修调有效
DB3200.DBX0006	.7	进给率修调有效
DB3200.DBX0007	.0	禁止 NC 启动
DB3200.DBX0007	.1	NC 启动
DB3200.DBX0007	.2	程序结束 NC 停止
DB3200.DBX0007	.3	NC 停止
DB3200.DBX0007	.4	NC 停止进给轴和主轴
DB3200.DBX0007	.7	复位
NCK 到 PLC		
DB3300.DBX0000	.3	动作程序段有效
DB3300.DBX0000	.4	移动程序段有效
DB3300.DBX0000	.5	M00/M01 有效
DB3300.DBX0000	.6	最后动作程序段有效
DB3300.DBX0001	.0	回参考点有效
DB3300.DBX0001	.4	程序段搜索有效

序号	位	名称
DB3300.DBX0001	.5	M2/M30 有效
DB3300.DBX0001	.7	程序测试有效
DB3300.DBX0003	.0	程序状态：运行
DB3300.DBX0003	.2	程序状态：停止
DB3300.DBX0003	.3	程序状态：中断
DB3300.DBX0003	.4	程序状态：终止
DB3300.DBX0003	.5	通道状态：有效
DB3300.DBX0003	.6	通道状态：中断
DB3300.DBX0003	.7	通道状态：复位
DB3300.DBX4001	.1	到达所需工件数量
HMI 到 PLC		
DB1700.DBX0000	.5	M01 已选择
DB1700.DBX0000	.6	空运行进给已选择
DB1700.DBX0001	.3	快速移动进给率修调已选择
DB1700.DBX0001	.7	程序测试已选择
DB1700.DBX0002	.0	“跳跃程序段”已选择
DB1900.DBX0000	.6	模拟有效

ASUP 信号

序号	位	名称	
PLC 到 NCK			
DB3400.DBX0000	.0	INT1 启动	
DB3400.DBX0001	.0	INT2 启动	
DB3400.DBX1000	.0	ASUP 结束	INT1
DB3400.DBX1000	.1	执行 ASUP	
DB3400.DBX1000	.2	未分配中断编号	
DB3400.DBX1000	.3	不能执行 ASUP	
DB3400.DBX1001	.0	ASUP 结束	INT2
DB3400.DBX1001	.1	执行 ASUP	
DB3400.DBX1001	.2	未分配中断编号	
DB3400.DBX1001	.3	不能执行 ASUP	

12 补偿（K3）

12.1 概述

补偿

对于数控系统，可以对进给轴进行以下轴专有补偿：

- 反向间隙补偿
- 可插补补偿
  - 补偿丝杠螺距误差和测量系统误差（LEC）
- 跟随误差补偿（转速前馈控制）
- 摩擦补偿（象限误差补偿）

机床中不同的误差补偿功能可以分别通过不同的机床数据进行设置。

位置显示

标准实际值和设定值位置显示不考虑补偿值并显示“理想机床”的位置值。查看补偿数据的步骤如下：



然后找到“绝对补偿值测量系统 1”一项。

12.2 间隙补偿

影响

带间接测量系统的进给轴/主轴在运行时，由于机械间隙的存在会导致轴移动行程的出错，比如，导致某个进给轴在换向时多走或少走了一个间隙值（见下图）。

补偿

作为对间隙的补偿，在进给轴/主轴每次换向时轴的实际值要由补偿间隙量来修正。

该值的大小可以在系统开机调试时在机床数据 MD32450 BACKLASH (反向间隙)中定义。

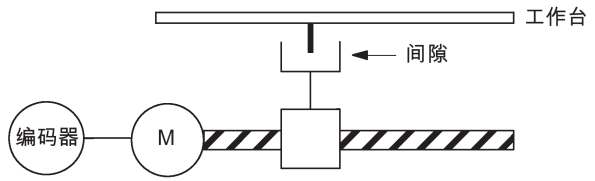
有效性

在回参考点后，间隙补偿在所有的运行方式下有效。

正向间隙

编码器快于机床部件(比如工作台)。因为这样由编码器所获得的实际位置值快于工作台的实际位置，所以工作台少走一个间隙值（见下图）。在此，输入一个**正值**的间隙补偿值（正常状态）。

正向间隙(正常情况)：



编码器实际值超出了真实的实际值（工作台）：
工作台运行时间太短

负向反向间隙

编码器运行落后于机床部件(比如工作台)。工作台多走一个间隙值。输入的补偿值为**负值**。

较大间隙补偿值

用户可以将相关进给轴方向改变时所产生的背隙补偿值可以分为几部分。这可以避免因进给轴给定值过大而产生的特定的轴误差。

使用 MD36500 ENC_CHANGE_TOL 来定义反向间隙补偿值的增量（MD32450 BACKLASH）。必须考虑到间隙补偿只包含在 (n=MD32450/MD36500) 次伺服循环之后的计算。时间间隔过长会产生零速监控警报。如果 MD36500 的值大于 MD32450，即在一个伺服循环中执行补偿。

12.3 可插补补偿

12.3.1 概述

术语和定义

补偿值：编码器测出的轴位置和编程的轴位置(=理想机床的轴位置)间的差异。补偿值通常称为偏移值。

补偿点：轴的位置和相应的补偿值。

补偿表：补偿点表

补偿表

由于滚珠丝杠和测量系统的测量偏差会对工件的加工有直接的影响，必须通过一定的位置相关的补偿值来补偿。补偿值根据测量的误差曲线来确定并在系统启动时，以补偿表的形式输入系统。对于每个补偿关系，必须建立单独的补偿表。

使用特定的系统变量输入补偿值和附加的表参数。

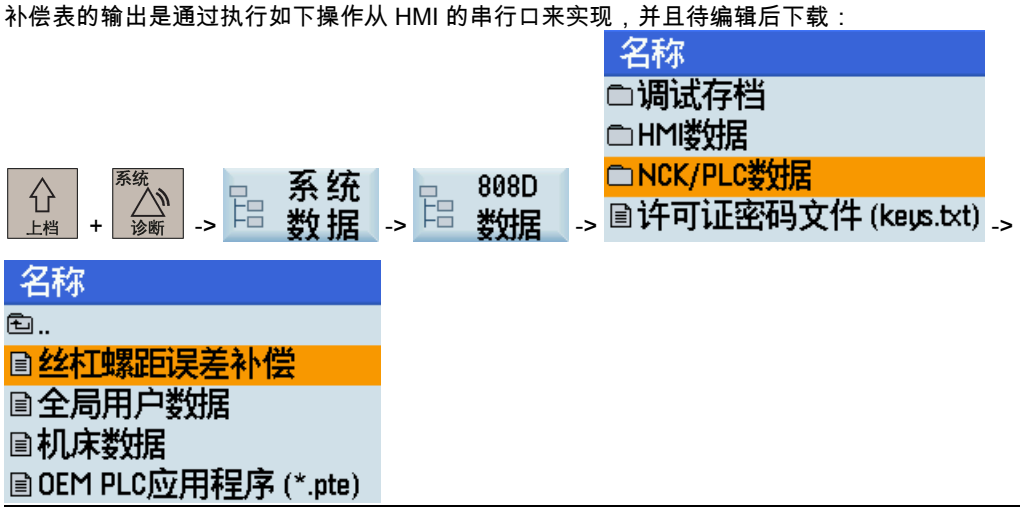
补偿表的输入

有两种方式可以将补偿表载入缓冲的 NC 用户存储器：

- 通过执行包含补偿表的数控系统程序载入补偿值。
- 也可以通过 HMI 的串行口从个人电脑（PC）中传输补偿表来载入补偿数据。

说明

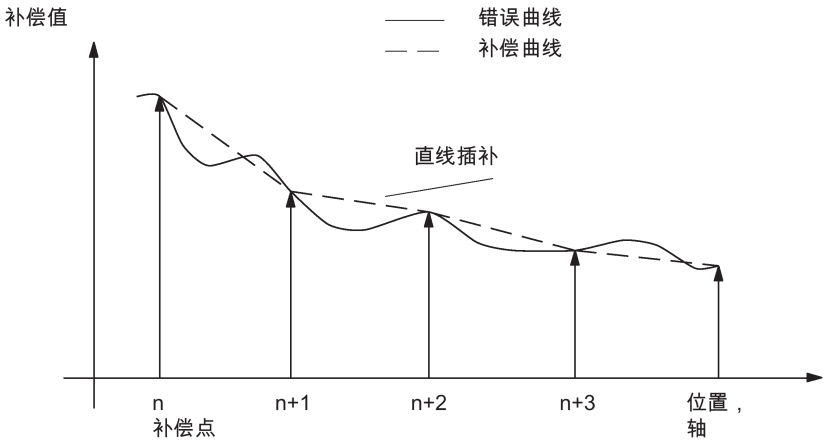
补偿表的输出是通过执行如下操作从 HMI 的串行口来实现，并且待编辑后下载：



补偿点之间的线性插补

由起始位置和终点位置确定的需补偿的进给路径将被分成尺寸相等的单元(单元的数量取决于误差曲线)，参见下图。在下图中，限制各个单元的实际位置就称为“补偿点”。调试时，必须输入每个补偿点（实际位置）的补偿值。两个补偿点间的偏移(补偿)值由相邻的补偿点(即相邻点由直线连接)的补偿值通过**线性插补**获得。

补偿点之间的线性插补：



参考点的补偿值

形成的补偿表必须使参考点的补偿值为“零”。这样可以避免在激活 SSFK 时（在回参考点运行之后）产生位置跃变。

12.3.2 SSFK

功能

丝杠螺距误差补偿/测量系统误差补偿(LEC)是指对进给轴进行的补偿。

在进行 LEC 时，相应的插补循环的补偿值修改进给轴的实际位置值，该轴的实际位置值将由机床进给轴直接运行。补偿值为正值时，进给轴在负方向运行。

补偿值的大小没有极限值，也没有受到监控。因此，为了避免由于补偿而使进给轴的速 度或加速度超过极限值，应选取较小的补偿值。否则，如果所选的补偿值过大，就会引起其它的轴监控功能发出报警(比如，轮廓监控，转速给定值极限)。

有效性

- 补偿值存储在 NC 用户存储器中并在上电后生效。
- 该功能对相关的机床轴生效
  (MD32700 ENC_COMP_ENABLE [0] = 1)。
- 进给轴已经到达参考点（接口信号“已到达参考点/同步 1”DB390x.DBX0.4 已设置）。

如果上述条件满足，进给轴的实际位置值将被相应的补偿值修改，并由进给轴立即运行。

如果参考点丢失（比如因为超出编码器频率，接口信号“回参考点/同步 1” = ‘0’），则补偿功能关闭。

补偿表

在补偿表中，每个进给轴均以系统变量的形式存储了位置相关的补偿值。最多可存储 200 个插补点（N = 0...199）。

为此，补偿表中应确定测量系统的下列参数（参见图“补偿表参数（LEC 的系统变量）”）：

- **补偿表中插补点 N 的补偿值：**
  $AA_ENC_COMP [0,N,AXi]= ...
  其中 Axi=进给轴名，如，X1，Y1，Z1；N = 插补点索引
  表中每个插补点(进给轴位置)填入一个补偿值。补偿值的大小不受限制。

  说明
  起始点补偿值和终点补偿值在整个补偿范围内有效，即如果补偿表无法覆盖整个补偿范围，这些补偿值应为“0”。

- **插补点间距：**$AA_ENC_COMP_STEP[0,AXi]= ...
  插补点间距确定补偿表中补偿值之间的距离(AXi 含义参见前面)。
- **起始位置：**$AA_ENC_COMP_MIN[0,AXi]= ...
  起始位置指相应进给轴在补偿表中开始补偿的进给轴位置(插补点 0)。
  起始位置的补偿值为 $ AA_ENC_COMP[0,**0**,AXi]。
  对于小于起始位置的所有其它位置均使用插补点 0 的补偿值（但不适用于取模的补偿表）。
- **终点位置：**$AA_ENC_COMP_MAX[0,AXi]= ...
  终点位置指相应进给轴在补偿表中结束补偿的进给轴位置(插补点 k<125)。
  终点位置的补偿值为 $AA_ENC_COMP[0,**k**,AXi]
  对于小于起始位置的所有其它位置均使用插补点 k 的补偿值（但不适用于取模的补偿表）。大于 k 的补偿值无效。
- **带取模功能的补偿**：$AA_ENC_COMP_IS_MODULO[0,AXi] = 1
  如果激活带取模功能的补偿，则重复循环执行补偿表，即$AA_ENC_COMP_MAX(=插 补点$AA_ENC_COMP[0,**k**,AXi])的补偿值直接跟随$AA_ENC_COMP_MIN（=插补点 $AA_ENC_COM[0,**0**,AXi]）的补偿值。
  对于取模 360° 的回转轴，建议使用 0 度作为起始位置($AA_ENC_COMP_MIN)和 360°作为终点位置($AA_ENC_COMP_MAX)。这两个补偿值必须直接填写。

说明

在填写补偿值的时候要特别注意，在给定范围内所有的插补点均分配一个补偿值(也就是说不可以留有空档)。否则，在这些插补点就使用以前所填写的补偿值。

说明

在机床数据 MD10240 SCALING_SYSTEM_IS_METRIC=0 时，表中位置参数值单位为英制。

可以通过手动转换来实现位置数据的自动换算。

只有在机床数据 MD32700 ENC_COMP_ENABLE=0 时才可以载入补偿表。当值“1” 时会激活补偿功能并给数据写保护(输出报警 17070)。

示例

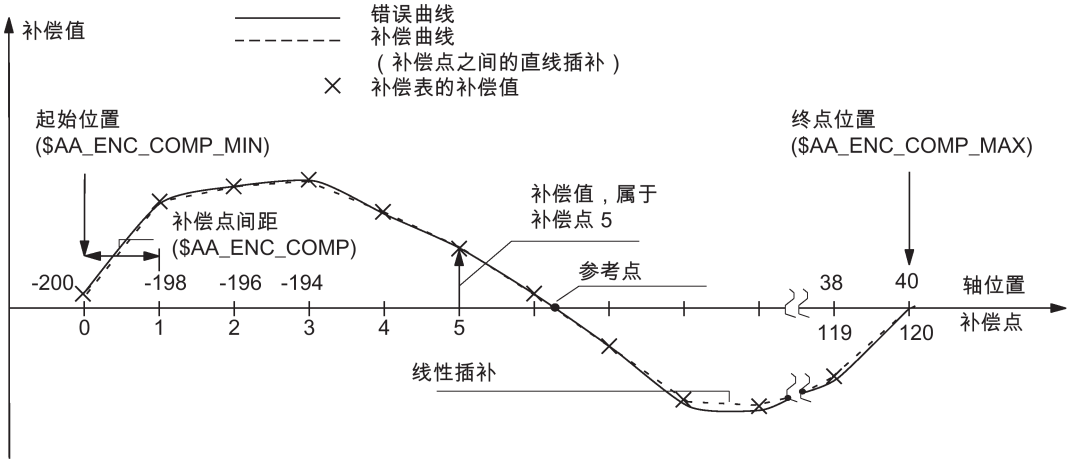
下面通过一个零件程序说明如何规定 X1 轴的补偿值。

```
%_N_AX_EEC_INI
CHANDATA(1)
$AA_ENC_COMP[0,0,X1]=0.0                    ; 1. 补偿值 ( 插补点 0 ) +0 mm
$AA_ENC_COMP[0,1,X1]=0.01                   ; 2. 补偿值 ( 插补点 1 ) +10 mm
$AA_ENC_COMP[0,2,X1]=0.012                  ; 3. 补偿值 ( 插补点 2 ) +12 mm
...
$AA_ENC_COMP[0,120,X1]=0.0                  ;终点补偿值 (插补点 120)

$AA_ENC_COMP_STEP[0,X1]=2.0                 ;插补点间距 2.0 mm
$AA_ENC_COMP_MIN[0,X1]=-200.0               ;补偿起始位置 -200.0 mm
$AA_ENC_COMP_MAX[0,X1] = 40.0               ;补偿终点位置 +40.0 mm
$AA_ENC_COMP_IS_MODULO[0,X1]=0              ;补偿不带取模功能 M17
```

插补点超过 125 将导致报警 12400 “元素丢失”。

补偿表的参数 (SSFK 系统变量) :



12.3.3 角度误差补偿

功能

角度误差补偿是指对车床上旋转定位角度进行的主主轴补偿。

补偿值的大小没有极限值，也没有受到监控。因此，为了避免由于补偿而使进给轴的速 度或加速度超过极限值，应选取较小的补偿值。否则，如果所选的补偿值过大，就会引起其它的轴监控功能发出报警(比如，轮廓监控，转速给定值极限)。

有效性

- 补偿值存储在 NC 用户存储器中并在上电后生效。

- 该功能对主主轴生效（MD32700 ENC_COMP_ENABLE [0] = 1）。
- 主主轴已经到达参考点（接口信号“已到达参考点/同步 1”DB390xDBX0.4 已设置）。

如果上述条件满足，主主轴的实际位置值将被相应的补偿值修改，并由主主轴立即运行。

如果参考点丢失（比如因为超出编码器频率，接口信号“回参考点/同步 1” = ‘0’），则补偿功能关闭。

补偿表

在补偿表中，主主轴以系统变量的形式存储了位置相关的补偿值。最多可存储 200 个插补点（N = 0...199）。

说明

在轴专用机床数据 MD38000 MM_ENC_ENC_COMP_MAX_POINTS[0]中存储最多可用的插补点。

为此，补偿表中应确定测量系统的下列参数：

- **补偿表中插补点 N 的补偿值：**
  $AA_ENC_COMP [0,N,AXi]= ...
  其中 Axi=进给轴名，如，X1，Y1，Z1；N = 插补点索引
  表中每个插补点(进给轴位置)填入一个补偿值。补偿值的大小不受限制。

  说明

  起始点补偿值和终点补偿值在整个补偿范围内有效，即如果补偿表无法覆盖整个补偿范围，这些补偿值应为“0”。

- **插补点间距：**$AA_ENC_COMP_STEP[0,AXi]= ...
  插补点间距确定补偿表中补偿值之间的距离。
- **起始位置：**$AA_ENC_COMP_MIN[0,AXi]= ...
  起始位置指主主轴在补偿表中开始补偿的主主轴位置（插补点 0）。
  起始位置的补偿值为 $ AA_ENC_COMP[0,**0**,AXi]。
  对于小于起始位置的所有其它位置均使用插补点 0 的补偿值（但不适用于取模的补偿表）。
- **终点位置：**$AA_ENC_COMP_MAX[0,AXi]= ...
  终点位置指主主轴在补偿表中结束补偿的主主轴位置（插补点 k<200）。
  终点位置的补偿值为 $AA_ENC_COMP[0,**k**,AXi)]
  对于小于起始位置的所有其它位置均使用插补点 k 的补偿值（但不适用于取模的补偿表）。大于 k 的补偿值无效。
- **带取模功能的补偿**：$AA_ENC_COMP_IS_MODULO[0,AXi] = 1
  如果激活带取模功能的补偿，则重复循环执行补偿表，即$AA_ENC_COMP_MAX(=插 补点$AA_ENC_COMP[0,**k**,AXi])的补偿值直接跟随$AA_ENC_COMP_MIN（=插补点 $AA_ENC_COM[0,**0**,AXi]）的补偿值。
  对于主主轴，建议使用 0 度作为起始位置($AA_ENC_COMP_MIN)和 360°作为终点位置($AA_ENC_COMP_MAX)。这两个补偿值必须直接填写。

  说明

  在填写补偿值的时候要特别注意，在给定范围内所有的插补点均分配一个补偿值(也就是说不可以留有空档)。否则，在这些插补点就使用以前所填写的补偿值。

  说明

  在机床数据 MD10240 SCALING_SYSTEM_IS_METRIC=0 时，表中位置参数值单位为英制。
  可以通过手动转换来实现位置数据的自动换算。
  只有在机床数据 MD32700 ENC_COMP_ENABLE=0 时才可以载入补偿表。当值“1” 时会激活补偿功能并给数据写保护(输出报警 17070)。

示例

主主轴运行自动定位程序与激光干涉仪配合进行位置值采样。

先决条件：

- 角度误差补偿范围：0° 至 360°
- 补偿间隔：30°

```
SPOS=0                          ; 主轴旋转至 0 度位置
M01                             ; 准备启动
SPOS=355                        ; 主轴旋转至 355 度位置
G4F1                            ; 等待 1 秒
SPOS=0                          ; 主轴旋转至 0 度位置
G4F5                            ; 等待 5 秒
SPOS=30                         ; 主轴旋转至 30 度位置
G4F5                            ; 等待 5 秒
SPOS=60                         ; 主轴旋转至 60 度位置
G4F5                            ; 等待 5 秒
SPOS=90                         ; 主轴旋转至 90 度位置
G4F5                            ; 等待 5 秒
SPOS=120                        ; 主轴旋转至 120 度位置
G4F5                            ; 等待 5 秒
SPOS=150                        ; 主轴旋转至 150 度位置
G4F5                            ; 等待 5 秒
SPOS=180                        ; 主轴旋转至 180 度位置
G4F5                            ; 等待 5 秒
SPOS=210                        ; 主轴旋转至 210 度位置
G4F5                            ; 等待 5 秒
SPOS=240                        ; 主轴旋转至 240 度位置
G4F5                            ; 等待 5 秒
SPOS=270                        ; 主轴旋转至 270 度位置
G4F5                            ; 等待 5 秒
SPOS=300                        ; 主轴旋转至 300 度位置
G4F5                            ; 等待 5 秒
SPOS=330                        ; 主轴旋转至 330 度位置
G4F5                            ; 等待 5 秒
SPOS=IC(30)                     ; 主轴旋转 30 度
G4F1                            ; 等待 1 秒
SPOS=IC(5)                      ; 主轴旋转 5 度
G4F1                            ; 等待 1 秒
SPOS=IC(-5)                     ; 主轴旋转 -5 度
G4F1                            ; 等待 1 秒
SPOS=330                        ; 主轴旋转至 330 度位置
G4F1                            ; 等待 1 秒
SPOS=300                        ; 主轴旋转至 300 度位置
G4F1                            ; 等待 1 秒
SPOS=270                        ; 主轴旋转至 270 度位置
G4F1                            ; 等待 1 秒
SPOS=240                        ; 主轴旋转至 240 度位置
G4F1                            ; 等待 1 秒
SPOS=210                        ; 主轴旋转至 210 度位置
G4F1                            ; 等待 1 秒
SPOS=180                        ; 主轴旋转至 180 度位置
```

```
G4F1                                    ; 等待 1 秒
SPOS=150                                ; 主轴旋转至 150 度位置
G4F1                                    ; 等待 1 秒
SPOS=120                                ; 主轴旋转至 120 度位置
G4F1                                    ; 等待 1 秒
SPOS=90                                 ; 主轴旋转至 90 度位置
G4F1                                    ; 等待 1 秒
SPOS=60                                 ; 主轴旋转至 60 度位置
G4F1                                    ; 等待 1 秒
SPOS=30                                 ; 主轴旋转至 30 度位置
G4F1                                    ; 等待 1 秒
SPOS=0                                  ; 主轴旋转至 0 度位置
G4F1                                    ; 等待 1 秒
M30                                     ; 程序结束
```

下面通过一个零件程序 MSP_COM.mpf 说明如何向系统写入主主轴的补偿值。

```
;Machine axis 4                         ; 主主轴地址
$AA_ENC_COMP[0,0,AX4]=0                 ; 补偿值（最小位置 0°）
$AA_ENC_COMP[0,1,AX4]=0.03666666        ; 补偿值（最小位置 0° + 1 个间隔位置）
$AA_ENC_COMP[0,2,AX4]=0.02861111        ; 补偿值（最小位置 0° + 2 个间隔位置）
$AA_ENC_COMP[0,3,AX4]=0.01638888        ; 补偿值（最小位置 0° + 3 个间隔位置）
$AA_ENC_COMP[0,4,AX4]=0.04361111        ; 补偿值（最小位置 0° + 4 个间隔位置）
$AA_ENC_COMP[0,5,AX4]=0.02833333        ; 补偿值（最小位置 0° + 5 个间隔位置）
$AA_ENC_COMP[0,6,AX4]=0.00833333        ; 补偿值（最小位置 0° + 6 个间隔位置）
$AA_ENC_COMP[0,7,AX4]=0.02527777        ; 补偿值（最小位置 0° + 7 个间隔位置）
$AA_ENC_COMP[0,8,AX4]=0.01055555        ; 补偿值（最小位置 0° + 8 个间隔位置）
$AA_ENC_COMP[0,9,AX4]=-0.00311111       ; 补偿值（最小位置 0° + 9 个间隔位置）
$AA_ENC_COMP[0,10,AX4]=0.02194444       ; 补偿值（最小位置 0° + 10 个间隔位置）
$AA_ENC_COMP[0,11,AX4]=0.01194444       ; 补偿值（最小位置 0° + 11 个间隔位置）
$AA_ENC_COMP[0,12,AX4]=0                ; 补偿值（最小位置 0° + 12 个间隔位置）
$AA_ENC_COMP_STEP[0,AX4]=30             ; 插补点间隔角度为 30°
$AA_ENC_COMP_MIN[0,AX4]=0               ; 补偿起始位置为 0°
$AA_ENC_COMP_MAX[0,AX4]=360             ; 补偿结束位置为 360°
$AA_ENC_COMP_IS_MODULO[0,AX4]=1         ; 补偿带取模功能
```

说明

确保在数控系统上于“AUTO”模式下运行以上用于补偿值输入的零件程序。零件程序运行结束后重启数控系统并执行回参考点操作后补偿值生效。

只有在设置了 MD32700 ENC_COMP_ENABLE=0 后才能运行该零件程序。当值“1” 时会激活补偿功能并给数据写保护(输出报警 17070)。

12.3.4 双向丝杠螺距误差补偿

12.3.4.1 功能说明

如果在两个方向上补偿值相差很大、间隙不恒定或对精度要求很高，可能需要采用双向丝杠螺距误差补偿或测量系统误差补偿（采用直接位置测量系统时）。

双向丝杠螺距误差补偿

采用“双向丝杠螺距误差补偿”（“双向 LEC”）时，每根轴会使用两个补偿表。一个补偿表用于正运行方向，另一个用于负运行方向。补偿表中记录了相应补偿点上的偏差（理想设定值和所测得的实际值之间的差值）。中间值的补偿值则由数控系统通过线性插补自动计算。

前提条件/生效方式

仅在满足下列前提条件时，“双向 LEC”才生效：

- 已为相关机床轴（补偿轴）激活了此功能：
  MD32710 $MA_CEC_ENABLE[<AXi>] = 1
- 补偿值存储在静态用户存储器中并在上电后生效。
- 已使能相应补偿表：
  SD41300 $SN_CEC_TABLE _ENABLE[<t>] = 1
- 参照轴和补偿轴的当前测量系统已回参考点：
  DB31, ... DBX60.4 或 60.5 = 1（已回参考点/已同步 1 或 2）

一旦满足上述条件，不管当前处于哪种运行方式，系统都会将根据参照轴的设定位置来按补偿值调整补偿轴的设定位置，并立即使补偿轴移动补偿值。

若接着参考位置由于超出编码器频率等原因丢失（DB31, ... DBX60.4 或 60.5 = 0），则会取消补偿。

可通过比照测量（例如使用激光干涉仪）检验补偿效果，或者在最简单的情形下查看各轴的信息显示对话框。

12.3.4.2 调试

误差值或补偿值的测量

同调试“单向 LEC”时一样，调试“双向 LEC”时也需要采用适宜的测量设备（例如激光干涉仪）测定每根轴的误差曲线。为了执行测量，必须创建一个包含测量点和等待时间的零件程序（参见章节“示例 (页 116)”：程序 “BI_SSFK_MESS_AX1_X.MPF”）。

在实践中，不同的测量仪器在和 SINUMERIK 数控系统配用时提供的功能也不同，因此本手册仅就本系统上常规的测量步骤进行说明。

说明

只有在机床数据中正确设置了 MCS 中轴的运行方向后，才能在首次调试时执行测量，以测定丝杠螺距误差。

执行调试

1. 定义控制点的数量
   必须为每根轴的正运行方向和负运行方向分别指定一个补偿表。必须通过以下机床数据定义控制点的数量：
   MD18342 $MN_MM_CEC_MAX_POINTS[<t>]（垂度补偿中控制点的最大数量）
   其中：<t> = 补偿表的下标
   允许范围：0 ≤ t < 7

   ⚠小心

   用户数据丢失

   改变 MD18342 后输出报警 4400：
   “缓冲重组！”
   为确保自动存储器配置不会使所有输入的数据丢失，在未进行批量调试前**请勿**重启系统。

   示例：
   MD18342 [0] = 11；11 个控制点用于补偿表 1，例如 X 轴正运行方向
   MD18342 [1] = 11；11 个控制点用于补偿表 2，例如 X 轴负运行方向
   MD18342 [2] = 21；21 个控制点用于补偿表 3，例如 Y 轴正运行方向
   MD18342 [3] = 21；21 个控制点用于补偿表 4，例如 Y 轴正运行方向
   …

MD18342 [61] = ...；补偿表 62 的控制点数量

2. 执行批量调试：
   – 创建包含 MD18342 [<t>] 输入的 NC 存档。
   – 回读已创建的 NC 存档。
     提示：此时系统会配置 NC 存储器。

   补偿表现在可用。
3. 以零件程序的形式为各轴及各运行方向创建含补偿值的表格（参见章节“示例 (页 116)”：程序“BI_SSFK_MESS_AX1_X.MPF”）。
4. 在数控系统中执行带补偿值的零件程序。
   “AUTO”方式 > 选择程序 > 循环启动

说明

每次读取补偿表前，以下数据必须总是先置 **0**，然后再置 **1** 用于激活补偿：

MD32710 $MA_CEC_ENABLE[<AXi>]（垂度补偿使能）= **0** → **1**

SD41300 $SN_CEC_TABLE _ENABLE[<t>]（补偿表使能）= **0** → **1**

反向间隙必须始终置 **0**：

MD32450 $MA_BACKLASH [<e>]（反向间隙）= **0**

其中：<e> = 位置测量系统

使用程序模版“BI_SSFK_TAB_AX1_X.MPF”（参见章节“示例 (页 116)”）可自动实现这些步骤。在手动输入机床数据时须注意“激活 MD”或“复位”。

5. 执行上电（热启动）。
6. 现在可使用激光干涉仪进行比照测量。
7. 为了进一步改善补偿结果，可在程序中对各补偿值进行修正。重新读取表格后无需再次上电。

说明

如第 **5** 步中所述，补偿表作为可执行程序载入至程序存储器中，在循环启动后传输至先前配置好的数控系统存储区域。该过程可能需要为每张表格重复。但也可以在一个初始化步骤中一次性载入所有表格。补偿值将在 MD32710[<AXi>] = 1 和强制上电后生效。

表格参数

以系统变量的形式为每个方向定义位置相关补偿和附加表格参数：

- $AN_CEC[<t>,<N>]（补偿表[<t>]的控制点<N>的补偿值）
- $AN_CEC_INPUT_AXIS[<t>]（参照轴）
- $AN_CEC_OUTPUT_AXIS[<t>]（补偿轴）

说明

在“双向 LEC”中，参照轴和补偿轴**相同**。

- $AN_CEC_STEP[<t>]（控制点间距）
- $AN_CEC_MIN[<t>]（起点）
- $AN_CEC_MAX[<t>]（终点）
- $AN_CEC_DIRECTION[<t>]（双向补偿）
  可通过此系统变量设置补偿表[<t>]是否只在正方向或负方向上生效：
  – $AN_CEC_DIRECTION[<t>] = 1：
    补偿表只对参照轴的正运行方向有效
  – $AN_CEC_DIRECTION[<t>] = -1：
    补偿表只对参照轴的负运行方向有效

说明

设置 $AN_CEC_DIRECTION[<t>] = 0（补偿表对参照轴的两个运行方向均有效）在“双向 LEC”中**无**意义。

- $AN_CEC_IS_MODULO[<t>]（带模数功能的补偿）

单位制

切换单位制（修改 MD10240 $MN_SCALING_SYSTEM_IS_METRIC）时，系统会自动对包含位置数据的表参数进行换算。

位置数据总是采用当前生效的单位制。换算必须于外部执行。
可如下配置位置数据的自动换算：

- MD10260 $MN_CONVERT_SCALING_SYSTEM = 1

采用此设置时，以下轴机床数据生效：

- MD32711 $MA_CEC_SCALING_SYSTEM_METRIC（垂度补偿的单位制）

可通过此机床数据为此轴上生效的所有表格确定单位制。所有位置数据和总补偿值一起都可以被换算为设置的单位制。单位制切换时不再需要外部换算位置数据。

监控

为了避免垂度补偿使机床轴的速度和加速度过大，系统会对总补偿值进行监控，并将其限制在最大值以内。
采用垂度补偿时的总补偿值上限通过以下机床数据逐轴定义：

- MD32720 $MA_CEC_MAX_SUM（垂度补偿的最大补偿值）

若所得到的总补偿值大于最大值，则会发出相应报警。程序处理不会因此而中断。作为附加设定值输出的补偿值被限制在最大值以内。
此外，也可以逐轴限制总补偿值的变化速率：

- MD32730 $MA_CEC_MAX_VELO（垂度补偿时的速度变化率）

输入值是系数，是相对于最大轴速度（MD32000 $MA_MAX_AX_VELO）的百分比值。
超出限值时会发出相应报警。程序处理不会因此而中断。一旦补偿值回到限值范围内，系统便会追加运行由于限制而未运行的行程。

12.3.4.3 示例

下面以一台三轴机床为例详细介绍 X 轴和位置相关的补偿表：

配置

补偿控制点的数量：

MD18342 $MN_MM_CEC_MAX_POINTS[**0**] = 11（表 **1**：**X** 轴，**正**运行方向）

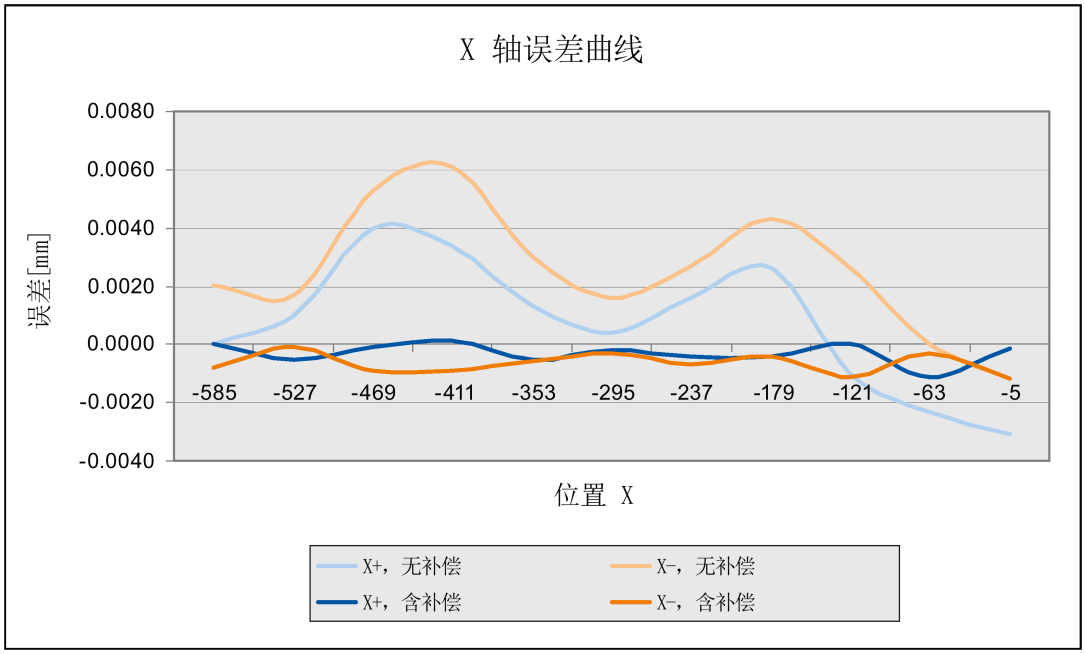
MD18342 $MN_MM_CEC_MAX_POINTS[**1**] = 11（表 **2**：**X** 轴，**负**运行方向）

控制点：

表<t>	[0,<N>]										
控制点数量：	MD18342 $MN_MM_CEC_MAX_POINTS[0] = 11										
控制点<N>	0	1	2	3	4	5	6	7	8	9	10
位置 X	-585	-527	-469	-411	-353	-295	-237	-179	-121	-63	-5

测量

			设定位置	偏差		经过补偿后的对比测量	
	位置	补偿编号	测量位置 [mm]	方向 + [mm]	方向 - [mm]	方向 + [mm]	方向 - [mm]
$AN_CEC_MIN[<t>]	-585	0	-585	0.0000	0.0020	0.0000	-0.0008
		1	-527	0.0010	0.0017	-0.0005	-0.0001
		2	-469	0.0040	0.0053	-0.0001	-0.0009
		3	-411	0.0034	0.0061	0.0001	-0.0009
		4	-353	0.0013	0.0030	-0.0005	-0.0006
		5	-295	0.0004	0.0016	-0.0002	-0.0003
		6	-237	0.0016	0.0027	-0.0004	-0.0007
		7	-179	0.0026	0.0043	-0.0004	-0.0004
		8	-121	-0.0010	0.0026	0.0000	-0.0011
		9	-63	-0.0023	0.0000	-0.0011	-0.0003
$AN_CEC_MAX[<t>]	-5	10	-5	-0.0031	-0.0012	-0.0001	-0.0012



编程

下面的“BI_SSFK_TAB_AX1_X.MPF”程序中包含了 X 轴的两个补偿表（正运行方向和负运行方向）的参数赋值：

```
;双向 LEC
;第 1 轴 MX1
;表 1，正运行方向
;表 2，负运行方向
;------------------------------------------------------------------------------
CHANDATA(1)
```

```
$MA_CEC_ENABLE[AX1]=0                    ; 补偿关闭
$SN_CEC_TABLE_ENABLE[0]=0                ; 禁用表 1
$SN_CEC_TABLE_ENABLE[1]=0                ; 禁用表 2
NEWCONF
;------------------------------------------------------------------------------
$AN_CEC[0,0]=0                           ; 第 1 补偿值（控制点 0）
$AN_CEC[0,1]=0.001                       ; 第 2 补偿值（控制点 1）
$AN_CEC[0,2]=0.004                       ; 第 3 补偿值（控制点 2）
$AN_CEC[0,3]=0.0034                      ; 第 4 补偿值（控制点 3）
$AN_CEC[0,4]=0.0013                      ; 第 5 补偿值（控制点 4）
$AN_CEC[0,5]=0.0004                      ; 第 6 补偿值（控制点 5）
$AN_CEC[0,6]=0.0016                      ; 第 7 补偿值（控制点 6）
$AN_CEC[0,7]=0.0026                      ; 第 8 补偿值（控制点 7）
$AN_CEC[0,8]=-0.001                      ; 第 9 补偿值（控制点 8）
$AN_CEC[0,9]=-0.0023                     ; 第 10 补偿值（控制点 9）
$AN_CEC[0,10]=-0.0031                    ; 终点补偿值（控制点 10）
$AN_CEC_INPUT_AXIS[0]=(AX1)              ; 参照轴
$AN_CEC_OUTPUT_AXIS[0]=(AX1)             ; 补偿轴
$AN_CEC_STEP[0]=58.0                     ; 控制点间距
$AN_CEC_MIN[0]=-585.0                    ; 补偿开始
$AN_CEC_MAX[0]=-5.0                      ; 补偿结束
$AN_CEC_DIRECTION[0]=1                   ; 补偿表对正运行方向生效
$AN_CEC_MULT_BY_TABLE[0]=0               ; 不相乘（此处无意义）
$AN_CEC_IS_MODULO[0]=0                   ; 无模数功能的补偿
;------------------------------------------------------------------------------
$AN_CEC[1,0]=0.002                       ; （控制点 0）
$AN_CEC[1,1]=0.0017                      ; （控制点 1）
$AN_CEC[1,2]=0.0053                      ; （控制点 2）
$AN_CEC[1,3]=0.0061                      ; （控制点 3）
$AN_CEC[1,4]=0.003                       ; （控制点 4）
$AN_CEC[1,5]=0.0016                      ; （控制点 5）
$AN_CEC[1,6]=0.0027                      ; （控制点 6）
$AN_CEC[1,7]=0.0043                      ; （控制点 7）
$AN_CEC[1,8]=0.0026                      ; （控制点 8）
$AN_CEC[1,9]=0.000                       ; （控制点 9）
$AN_CEC[1,10]=-0.0012                    ; （控制点 10）
$AN_CEC_INPUT_AXIS[1]=(AX1)              ; 参照轴
$AN_CEC_OUTPUT_AXIS[1]=(AX1)             ; 补偿轴
$AN_CEC_STEP[1]=58.                      ; 控制点间距
$AN_CEC_MIN[1]=-585.0                    ; 补偿开始
$AN_CEC_MAX[1]=-5.0                      ; 补偿结束
$AN_CEC_DIRECTION[1]=-1                  ; 补偿表对负运行方向生效
$AN_CEC_MULT_BY_TABLE[1]=0               ; 不相乘（此处无意义）
$AN_CEC_IS_MODULO[1]=0                   ; 无模数功能的补偿（仅针对回转轴）
;------------------------------------------------------------------------------
$MA_CEC_ENABLE[AX1]=1                    ; 补偿激活
$SN_CEC_TABLE_ENABLE[0]=1                ; 使能表 1
```

```
$SN_CEC_TABLE_ENABLE[1]=1                    ; 使能表 2
NEWCONF
M17
```

还可创建其他补偿表，例如用于 Y 轴和 Z 轴：

MD18342 $MN_MM_CEC_MAX_POINTS[**2**] = 90（表 **3**：**Y** 轴，正运行方向）

MD18342 $MN_MM_CEC_MAX_POINTS[**3**] = 90（表 **4**：**Y** 轴，负运行方向）

MD18342 $MN_MM_CEC_MAX_POINTS[**4**] = 50（表 **5**：**Z** 轴，正运行方向）

MD18342 $MN_MM_CEC_MAX_POINTS[**5**] = 50（表 **6**：**Z** 轴，负运行方向）

12.4 跟随误差补偿（前馈控制）

12.4.1 概述

轴相关跟随误差

前馈控制可以将跟随误差降到接近零。所以前馈控制也称为“跟随误差补偿”。

特别是当在轮廓弯曲处加速时，如，圆弧和拐角，此跟随误差将导致不期望的速度相关的轮廓误差。数控系统使用前馈控制类型“速度前馈控制”。

在零件程序中激活/取消激活前馈控制

可以使用以下高级语言指令在零件程序中使能/取消前馈控制：

- FFWON - 前馈控制开启
- FFWOF - 前馈控制关闭（系统上电时的缺省位置）

MD32630 FFW_ACTIVATION_MODE 定义是否可以由 FFWON 激活轴的前馈控制功能或由 FFWOF 取消激活此功能。

- 如果设定 MD32630=1，则可以使用 FFWON 和 FFWOF 激活/取消激活所有进给轴/主轴的前馈控制。
- 对于插补轴，也应将 MD32630 设为相同值。

为了避免突变，只能当进给轴/主轴停止时才能开启/关闭前馈控制。编程者必须遵守这一点。

条件

使用前馈控制时，必须遵守以下内容：

- 机床性能稳定
- 了解机床的动态特性
- 没有突变的位置和速度给定值

优化控制回路

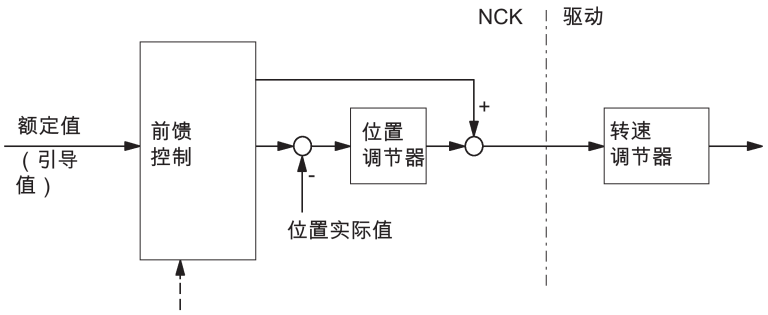
前馈控制是专为进给轴/主轴设置的。首先，必须给相应的进给轴/主轴的电流，速度和位置控制回路设定一个优化值。

参数定义

然后必须在机床数据中（参见下一章节）定义进给轴/主轴前馈控制参数。

12.4.2 速度前馈控制

使用速度前馈控制时，在速度控制器输入端输入速度给定值作为附加设定值（见下图）。



MD 32810: EQUIV_SPEEDCTRL_TIME

参数

为了使速度前馈控制设定正确，必须精确计算速度控制回路的等效时间常量并作为机床数据 MD32810 EQUIV_SPEEDCTRL_TIME （闭合速度控制回路中的等效时间常量）输入。

12.5 摩擦补偿（象限误差补偿）

12.5.1 一般功能说明

除了转动惯量和切削力以外，齿轮和机床导轨中的摩擦力也会影响机床轴的运行。 轴从静止开始加速时，尤其是从静摩擦转换为明显降低的滑动摩擦时，会对轮廓精度产生不良影响。

此时出现的摩擦力跃变会引起短暂的跟随误差增大。 在插补轴（路径轴）上，这会导致明显的轮廓误差。 在加工圆形轮廓时，由于在转换方向时一根轴静止，轮廓误差特别突出，尤其在过象限处。

激活摩擦补偿或象限补偿后，当轴从静止状态开始加速，即从静摩擦转换为滑动摩擦时，系统会作为补偿值加入附加的设定值脉冲。 由此可以几乎完全消除圆形轮廓上过象限处的轮廓误差。

和加速度相关的摩擦补偿

在多数情形下，通过一个恒定幅值的、与轴加速度无关的补偿值即可实现象限误差补偿。 但是，若补偿值与加速度相关，可通过“带自适应的摩擦补偿”激活一条自适应特性曲线，为补偿建模型。

圆度测试

调试摩擦补偿时，最简单的方法是使用圆度测试功能。 此时系统会采集加工圆弧期间各个机床轴实际位置上产生的圆轮廓，然后将该轮廓和编程的轮廓对比，用示意图展示两者的差别，尤其是过象限处的差别。

12.5.2 边界条件

说明

取消设定值相关补偿

以下补偿对位置设定值生效，因此要关闭圆度测试中涉及的轴的这些补偿，然后再开始测试。

双向丝杠螺距误差补偿：

MD32710 $MA_CEC_ENABLE[<轴>] = 0

12.5.3 补偿值恒定的摩擦补偿

12.5.3.1 功能激活

使能

摩擦补偿通过以下机床数据激活：

MD32490 $MA_FRICT_COMP_MODE[<轴>] = 1

激活

补偿值恒定的摩擦补偿通过以下机床数据激活：

- MD32500 FRICT_COMP_ENABLE[<轴>] = 1（摩擦补偿激活）
- MD32510 $MA_FRICT_COMP_ADAPT_ENABLE[<轴>] = 0（自适应关闭）

参数

补偿值恒定的摩擦补偿会考量以下参数：

- MD32520 $MA_FRICT_COMP_CONST_MAX（最大补偿值）
  采用补偿值恒定的摩擦补偿时，参数设置的值作为补偿值激活。
- MD32540 $MA_FRICT_COMP_TIME（摩擦补偿时间常数）
  补偿值通过一个 DT1 滤波器激活。 补偿值在设置的时间段后变为零。

12.5.3.2 调试

圆度测试

如上文所述，调试补偿值恒定的摩擦补偿时建议使用圆度测试。调试的过程可分为如下步骤：

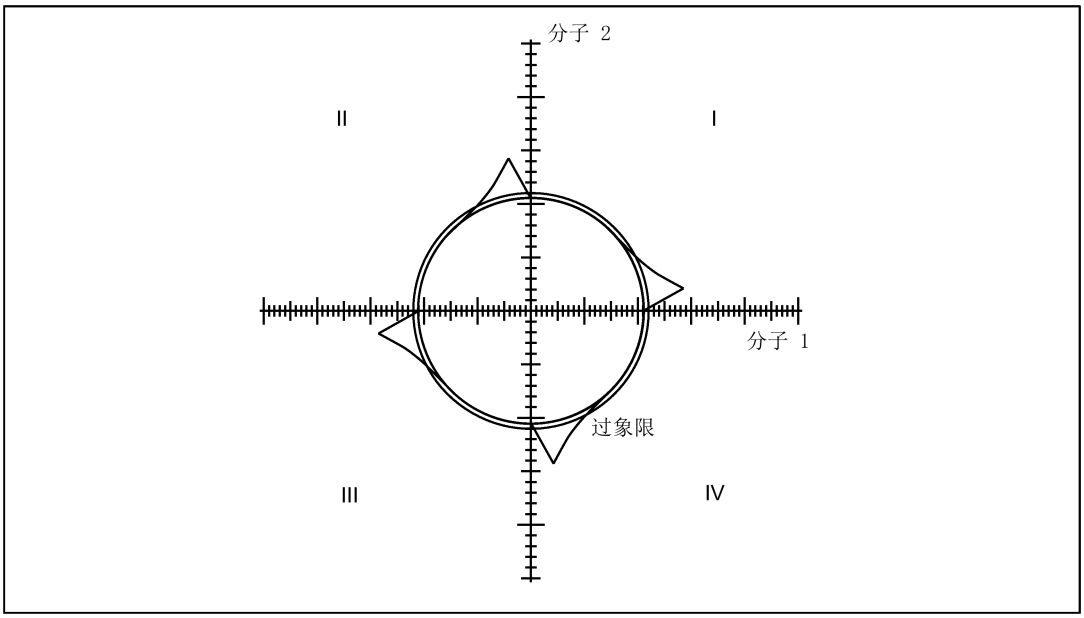
1. 执行**无**摩擦补偿的圆度测试
2. 执行**带**摩擦补偿的、采用低参数值的圆度测试
3. 执行带摩擦补偿的、采用调高的参数值的圆度测试
4. 完成带摩擦补偿的、采用最佳参数值的圆度测试

无摩擦补偿的圆度测试

为了测定过象限处圆弧轮廓的初始质量，必须执行无摩擦补偿的圆度测试。为此必须事先取消摩擦补偿：

MD32500 FRICT_COMP_ENABLE[<轴>] = 0

下图显示了无摩擦补偿时过象限处的典型情形：



之后必须重新启用补偿值恒定的摩擦补偿：

MD32500 FRICT_COMP_ENABLE[<轴>] = 1

带摩擦补偿的、采用低参数值的圆度测试

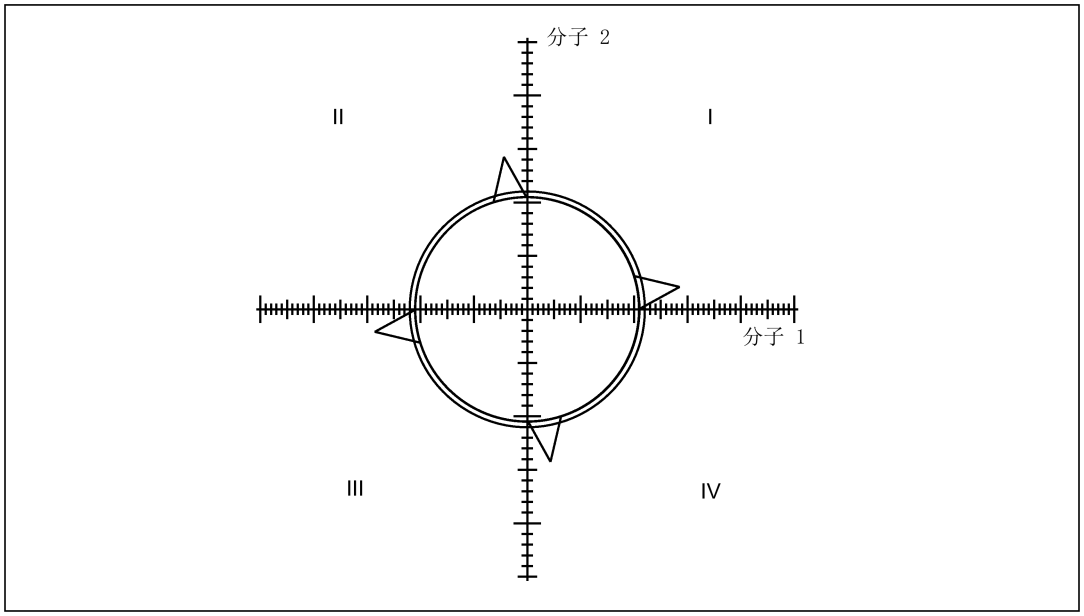
建议使用低补偿值和低时间常数（持续仅几个位置控制器周期）开展测试，例如：

- MD32520 $MA_FRICT_COMP_CONST_MAX[<轴>] = 10 [mm/min]
- MD32540 $FRICT_COMP_TIME[<轴>] = 0.008 [ms]

使用这些参数值进行圆度测试后，可对摩擦补偿的效果进行初步评估。

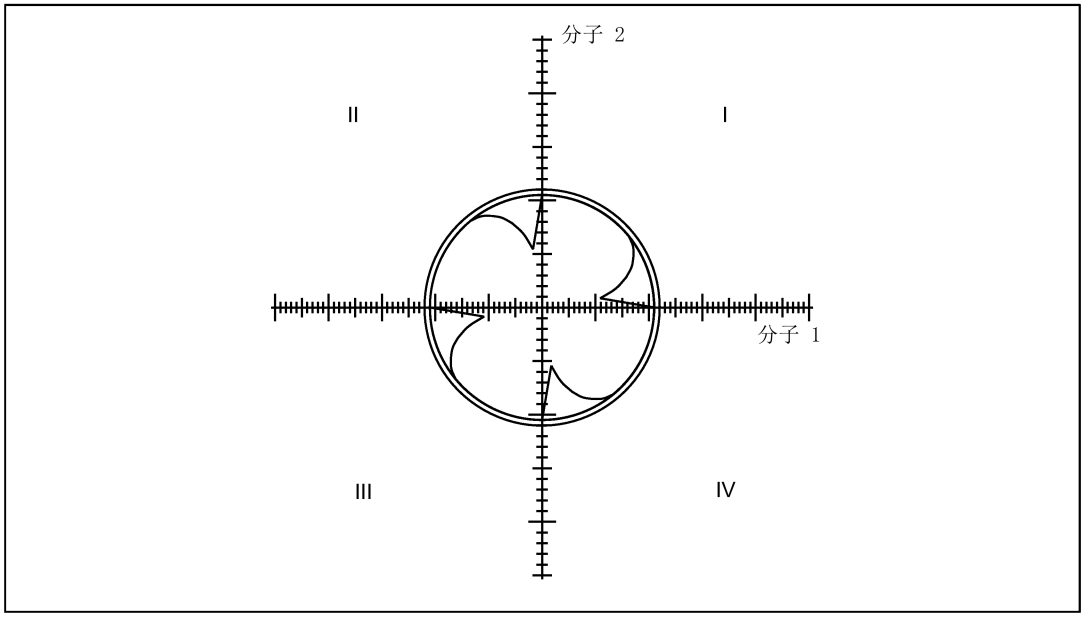
补偿值过小

在圆度测试中，若过象限处的轮廓偏差得到的补偿不足，则表示补偿值（MD32520）过小。



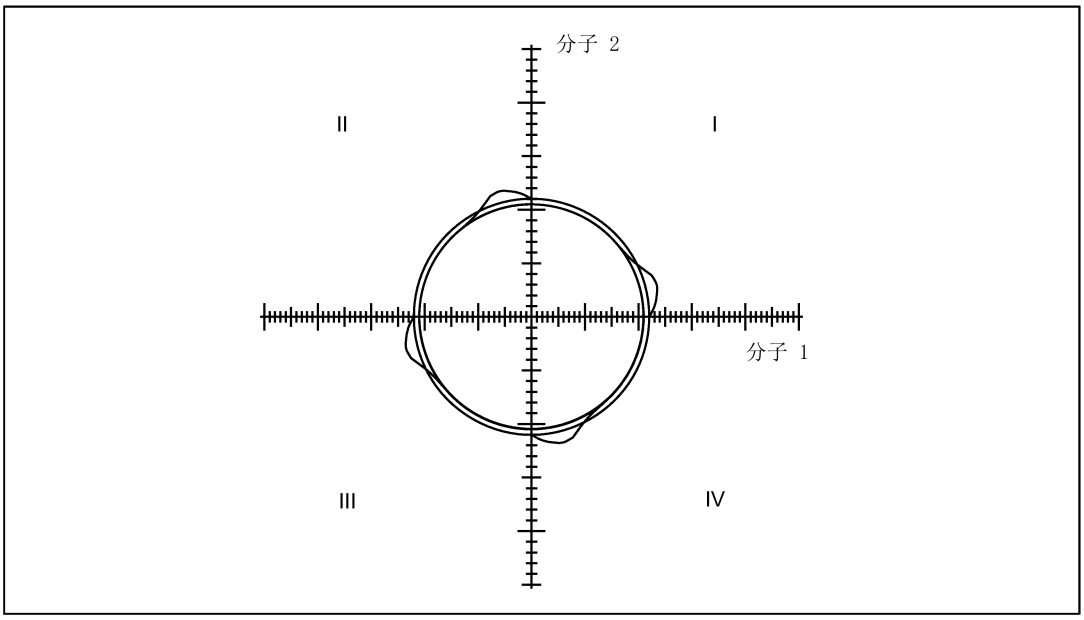
补偿值过大

在圆度测试中，若过象限处的轮廓偏差过补偿，则表示补偿值（MD32520）过大。



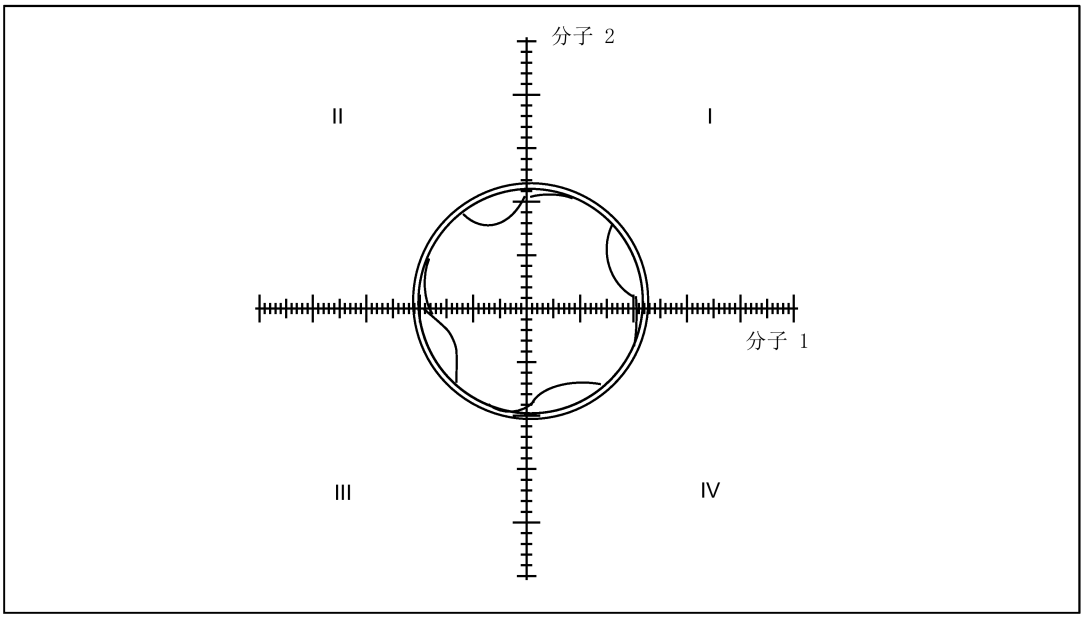
时间常数过小

在圆度测试中，若过象限处的轮廓偏差得到短暂补偿，之后又立即增大，则表示时间常数（MD32540）过小。



时间常数过大

在圆度测试中，如果一开始误差得以补偿，但后续轮廓上出现过度补偿，则表明时间常数（MD32540）设置过大。（前提条件：补偿值用的是最佳补偿值。）它导致补偿值继续作用，从而出现过度补偿。



补偿参数的优化

必须通过微调参数值、多次重复圆度测试来获得最佳的补偿参数。建议用机床加工中各种典型的半径和轨迹速度来进行测试。

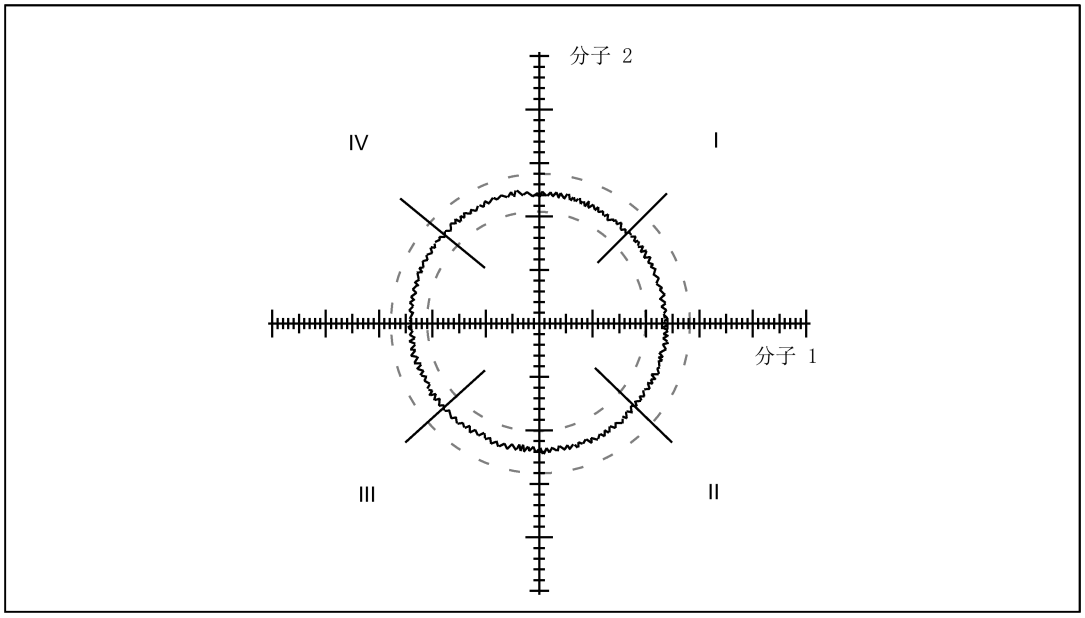
每次调整参数后，都要接着进行圆度测试，并检查和记录测试结果。

求平均值

若采用不同的半径和轨迹速度时所得到的参数值不同，则必须通过求平均值确定最适宜的值。

正确设置的摩擦补偿

摩擦补偿正确设置后，过象限处“不”再观察到轮廓误差。



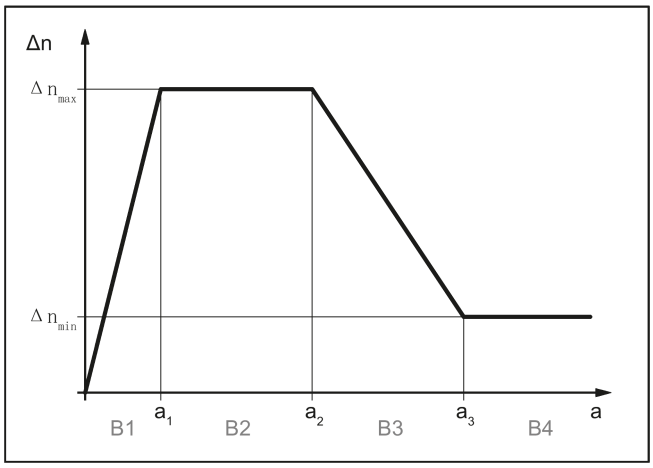
和加速度相关的补偿值

若补偿值为加速度相关，可于随后的步骤中启用下文所述的“带自适应的摩擦补偿”。

12.5.4 补偿值和加速度相关的摩擦补偿

12.5.4.1 功能说明

当补偿值受到加速度的显著影响时，通常情况下高加速度条件下最佳的补偿值对于低加速度条件来说必然太低。这种相关性可以通过下面的自适应曲线来建模。



Δn_{max} 最大补偿值
Δn_{min} 最小补偿值
a_1 加速度值 1
a_2 加速度值 2
a_3 加速度值 3
Bn 加速阶段，其中 n = 1, 2, ... 4
其中： 加速度：a1 < a2 < a3
补偿值：$\Delta n_{max} > \Delta n_{min}$，在特殊情形下也可能 $\Delta n_{max} < \Delta n_{min}$

根据对应的加速阶段 B1 至 B4，补偿值 Δn 如下计算：

范围	其中加速度 a	⇒ 补偿值 Δn
B1	$a < a_1$	$\Delta n = \Delta n_{max} * a / a_1$
B2	$a_1 \leq a \leq a_2$	$\Delta n = \Delta n_{max}$
B3	$a_2 < a < a_3$	$\Delta n = \Delta n_{max} + [(\Delta n_{min} - \Delta n_{max}) / (a_3 - a_2)] * (a - a_2)$
B4	$a \geq a_3$	$\Delta n = \Delta n_{min}$

12.5.4.2 功能激活

使能

摩擦补偿通过以下机床数据激活：

MD32490 $MA_FRICT_COMP_MODE[<轴>] = 1

激活

带自适应特性曲线的摩擦补偿通过以下机床数据激活：

- MD32500 FRICT_COMP_ENABLE[<轴>] = 1（摩擦补偿激活）
- MD32510 $MA_FRICT_COMP_ADAPT_ENABLE[<轴>] = 1（自适应特性曲线激活）

12.5.4.3 调试

为了测定特性曲线参数，需要在指定加速阶段的多个工作点上确定最优补偿值 Δn_{opt}。 相关内容参见章节“调试 (页 121)”。 尤其是在高轨迹速度和小半径的条件下，须确保测量值足够多。

建议用示意图来表示得出的值对用于分析：
$\Delta n_{opt} = f(a)$，其中 Δn_{opt} = 最优补偿值，a = 过象限处的加速度。

之后必须将由测量结果确定的自适应特性曲线的参数输入机床数据。

特性曲线参数

加速度值

换向轴过象限处的加速度如下计算：

$a = v^2 / r$，其中 v = 轨迹速度，r = 圆弧半径

说明

通过进给倍率开关可以简单的方式调整轨迹速度和轴加速度 a。

作为特性曲线参数测定的加速度值 a_1、a_2 和 a_3 必须输入以下机床数据。 此时须遵循以下条件： $a_1 < a_2 < a_3$

- MD32550 $MA_FRICT_COMP_ACCEL1（加速度值 1）
- MD32560 $MA_FRICT_COMP_ACCEL2（加速度值 2）
- MD32570 $MA_FRICT_COMP_ACCEL3（加速度值 3）

补偿值

作为特性曲线参数测定的补偿值 Δn_{min}、Δn_{max} 必须输入以下机床数据：

- MD32520 $MA_FRICT_COMP_CONST_MAX（最大补偿值）
- MD32530 $MA_FRICT_COMP_CONST_MIN（最小补偿值）

说明

若在轨迹速度极小的情况下未获得满意的测量结果，可能须提高计算精度：

- MD10200 $MA_INT_INCR_ PER_MM（线性插补的计算精度）
- MD10210 $MA_INT_INCR_PER_DEG（角度位置的计算精度）

12.5.5 适用于短程序段的补偿值

在运行程序段较短时，测定的象限误差补偿值可能会导致过补偿。 可为此类在一个插补周期内执行完毕的短程序段降低补偿值，以避免过度补偿。 但是每根轴上的降低幅度主要取决于具体的机床工况，因此需根据经验设定。 此值通过以下机床数据以圆度测试中测定的补偿值的百分比设置：

MD32580 $MA_FRICT_COMP_INC_FACTOR（适用于短程序段的补偿值）

12.6 数据表

12.6.1 机床数据

序号	名称	名称
一般		
10200	INT_INCR_PER_MM	线性位置计算精度
10210	INT_INCR_PER_DEG	角度位置计算精度
18342	MM_CEC_MAX_ POINTS[t]	垂度补偿的控制点数量上限
轴专用		
32450	BACKLASH[0]	背隙
32490	FRICT_COMP_MODE	摩擦补偿类型
32500	FRICT_COMP_ENABLE	摩擦补偿生效
32510	FRICT_COMP_ADAPT_ENABLE	摩擦补偿自适应生效
32520	FRICT_COMP_CONST_MAX	最大摩擦补偿值
32530	FRICT_COMP_CONST_MIN	最小摩擦补偿值
32540	FRICT_COMP_TIME	摩擦补偿时间常数
32550	FRICT_COMP_ACCEL1	自适应加速度值 1
32560	FRICT_COMP_ACCEL2	自适应加速度值 2
32570	FRICT_COMP_ACCEL3	自适应加速度值 3
32580	FRICT_COMP_INC_FACTOR	短暂运动中摩擦补偿值的权重系数
32630	ACTIVATION_MODE	通过程序使能前馈控制
32700	ENC_COMP_ENABLE[0]	插补补偿有效
32710	CEC_ENABLE	垂度补偿使能
32810	EQUIV_SPEEDCTRL_TIME[0]...[5]	速度控制回路的等效时间常量
36500	ENC_CHANGE_TOL	间隙补偿间距
38000	MM_ENC_COMP_MAX_POINTS[0]	编码器/主轴补偿螺距插补点数(LEC) (只用于显示)

12.6.2 设定数据

序号	名称	名称
一般		
41300	CEC_TABLE_ENABLE[t]	使能垂度补偿表

12.6.3 接口信号

序号	位	名称
进给轴/主轴专用		
DB390x.DBX0000	.4	回参考点/同步 1

13 动态转换（M1）

13.1 简要描述

使用范围

数控系统将编程的进给指令从笛卡儿坐标系转换到实际坐标系。

TRANSMIT 转换用于在车床上对车削部件进行端面铣削加工(无 Y 进给轴)。

TRACYL 转换用于对圆柱体外表面的加工。 主要用于槽铣削。 其中一个 TRACYL 变量用于车床。 另一个变量用于带 Y 轴的车床或带旋转工作台的铣床。

机床要求

车床必须具有一个可用作 C 轴的主主轴。 第二个主轴必须可以驱动铣刀。 使用 TRACYL 时，铣床必须具有旋转工作台用于和其它轴插补。

可用性

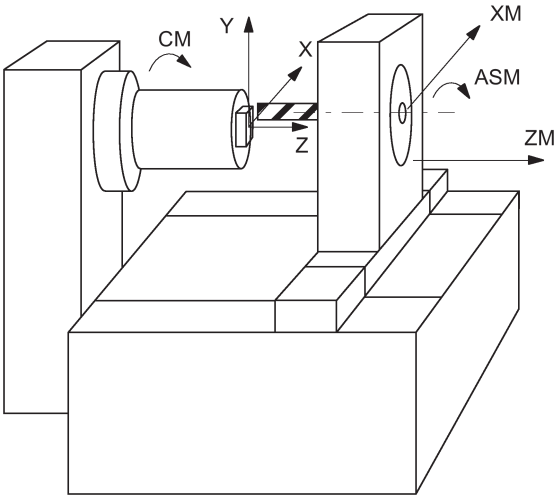
由单独的机床数据对功能 TRANSMIT 和 TRACYL 进行配置并通过程序中的特定语句来激活或取消。

对于 SINUMERIK 808D ADVANCED，最多可以配置两个车铣复合加工（TRANSMIT,TRACYL）；其中一个可以由程序激活。

13.2 TRANSMIT

13.2.1 概览

通过 TRANSMIT 铣削车削件的端面：



X, Y, Z　编程端面加工的笛卡儿坐标系

ASM　第二主轴（用于铣刀，钻头的工作主轴）

ZM　Z 机床进给轴（线性轴）

XM　X 机床进给轴（线性）

CM　C 进给轴（主主轴作为回转轴）

机床运动要求

两个线性轴（XM、ZM）必须相互垂直。回转轴（CM）必须和线性轴 ZM 平行（围绕 ZM 旋转）。线性轴 XM 和回转轴 CM 相交（车削中心）。

激活/取消 TRANSMIT

TRANSMIT 功能可以通过

- TRANSMIT 的单独程序段激活
- TRAFOOF 的单独程序段取消

使用 TRAFOOF 将取消任何有效的转换功能。

编程举例

```
N10 G0 X...Z...SPOS=...                ; 初始位置，主轴在位置控制中
N20 G17 G94 T...                       ; 平面、进给类型、选择铣刀
N30 SETMS(2)                           ; 转换：主主轴现在为铣削主轴
N40 TRANSMIT                           ; 打开 TRANSMIT
N50 G1 G41 F200 X...Y...Z...M3 S...    ; 使用刀具半径补偿铣削端面
...
N90 G40 ...
N100 TRAFOOF                           ;关闭 TRANSMIT
N110 G18 G95 T...                      ; 返回车削加工
N120 SETMS                             ; 主主轴是车削主轴
```

说明：

根据所编程的 X-Y 路径（直线或路径），移动机床进给轴 XM 和 CM，使得在端面上可以通过车削件进行轮廓加工。编程的 Z 轴（进给）仍然作为 Z 轴进给。

13.2.2 配置 TRANSMIT

机床数据

一般机床数据使用的机床进给轴，通道轴和几何轴的名称（$MN_AXCONF...和通道专用的机床数据 $MC_AXCONF...）也同样适用于转换功能。

只当转换功能无效时才可使用在 MD20050 AXCONF_GEOAX_ASSIGN_TAB 中定义的几何轴。如果转换功能有效，需进行特殊定义。

说明

用于转换功能的机床进给轴名称，通道轴名称和几何轴名称必须不同：

- MD10000 AXCONF_MACHAX_NAME_TAB，
- MD20080 AXCONF_CHANAX_NAME_TAB，
- MD20060 AXCONF_GEOAX_NAME_TAB。

使用 TRANSMIT 的例外：
MD20060 和 MD20080 的轴名称（几何轴和通道轴）可以相同，例如：X，Y，Z。如果未激活转换功能，则没有 Y 轴。

转换功能所需的机床数据

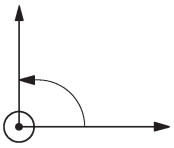
MD24100 TRAFO_TYPE_1	= 256 用于第一转换 TRANSMIT
MD24110 TRAFO_AXES_IN_1[n]	转换 1 的通道轴
MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[n]	转换 1 的几何轴
MD24200 TRAFO_TYPE_2	= 256 用于第二转换 TRANSMIT
MD24210 TRAFO_AXES_IN_2[n]	转换 2 的通道轴
MD24220 TRAFO_GEOAX_ASSIGN_TAB_2[n]	转换 2 的几何轴

机床数据 MD24110/MD24210 中转换功能所需的通道轴分配：

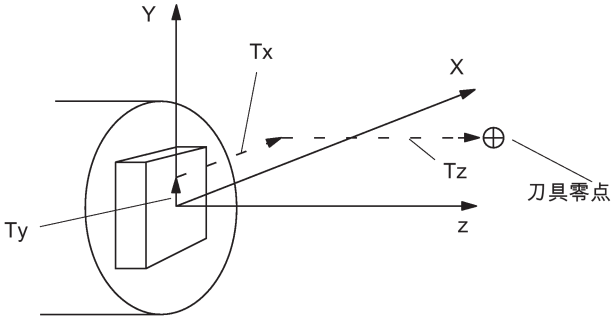
TRAFO_AXES_IN_1/2[0]=	垂直于回转轴的通道轴号
TRAFO_AXES_IN_1/2[1]=	回转轴的通道轴号
TRAFO_AXES_IN_1/2[2]=	平行于回转轴的通道轴号。

TRANSMIT 所需的特殊机床数据

- MD24900 TRANSMIT_ROT_AX_OFFSET_1
  笛卡儿坐标系中 x-y 平面的旋转位置，该位置相对于定义的回转轴零点，单位度（0...<360）。
- MD24910 TRANSMIT_ROT_SIGN_IS_PLUS_1
  如果从 Z 轴的负方向观察，发现在 x-y 平面中的回转轴的旋转方向是逆时针，则将机床数据值设定为 1；否则，设为 0。
  MD 值 = 1 的旋转方向：



- MD24920 TRANSMIT_BASE_TOOL_1
  通知数控系统和相对于 TRANSMIT 使用的坐标系原点的刀具零点位置。此机床数据可以给笛卡儿坐标系中的三个进给轴定义轴元素。
  分配进给轴元素：
  TRANSMIT_BASE_TOOL_1[0]=Tx
  TRANSMIT_BASE_TOOL_1[1]=Ty
  TRANSMIT_BASE_TOOL_1[2]=Tz（参下图）
  对应于笛卡儿坐标系原点的刀具零点的位置（车削中心）：



Tx, Ty, Tz - 位置的轴元素

- MD24911 TRANSMIT_POLE_SIDE_FIX_1 = 0
  连续通过极点

通过极点

在车削中心中，TRANSMIT 平面的 X=0,Y=0 点称为”极点”（X 进给轴与车削中心的交点）。

在极点附近，几何轴 X、Y 位置的较小的变化通常会导致回转轴位置的较大的变化（例外：路径只导致 XM 轴动作）。

因此不建议在极点的附近加工工件，因为在一些情况下，要求进给率迅速变化以防止回转轴过载。如果刀具正好在极点处，不要选择 TRANSMIT 功能。避免 X0/Y0 极点和刀具中心点位移相交。

示例：用于 TRANSMIT 的机床数据设定值

通用设置：轴名称：XM->X1，ZM->Z1，CM->SP1

- 机床轴名称
  MD10000 AXCONF_MACHAX_NAME_TAB[0]="X1"
  MD10000 AXCONF_MACHAX_NAME_TAB[1]="Z1"
  MD10000 AXCONF_MACHAX_NAME_TAB[2]="SP1"
  MD10000 AXCONF_MACHAX_NAME_TAB[3]="SP2"
  MD10000 AXCONF_MACHAX_NAME_TAB[4]=""
- 分配几何轴给通道轴
  MD20050 AXCONF_GEOAX_ASSIGN_TAB[0]=1
  MD20050 AXCONF_GEOAX_ASSIGN_TAB[1]=0
  MD20050 AXCONF_GEOAX_ASSIGN_TAB[2]=2

- 通道中的几何轴名称
  MD20060 AXCONF_GEOAX_NAME_TAB[0]="X"
  MD20060 AXCONF_GEOAX_NAME_TAB[1]="Y"
  MD20060 AXCONF_GEOAX_NAME_TAB[2]="Z"
- 通道内有效的机床轴号
  MD20070 AXCONF_MACHAX_USED[0]=1
  MD20070 AXCONF_MACHAX_USED[1]=2
  MD20070 AXCONF_MACHAX_USED[2]=3
  MD20070 AXCONF_MACHAX_USED[3]=4
  MD20070 AXCONF_MACHAX_USED[4]=0
- 通道内的机床轴名称
  MD20080 AXCONF_CHANAX_NAME_TAB[0]="X"
  MD20080 AXCONF_CHANAX_NAME_TAB[1]="Z"
  MD20080 AXCONF_CHANAX_NAME_TAB[2]="C"
  MD20080 AXCONF_CHANAX_NAME_TAB[3]="SP2"
  MD20080 AXCONF_CHANAX_NAME_TAB[4]=""
- 通道内主主轴的复位位置
  MD20090 SPIND_DEF_MASTER_SPIND=1

转换类型 TRANSMIT：

- 定义通道内的转换 1
  MD24100 TRAFO_TYPE_1=256
- 通道内转换 1 的轴分配
  MD24110 TRAFO_AXES_IN_1[0] = 1
  MD24110 TRAFO_AXES_IN_1[1] = 3
  MD24110 TRAFO_AXES_IN_1[2] = 2
  MD24110 TRAFO_AXES_IN_1[3]=0
  MD24110 TRAFO_AXES_IN_1[4]=0
- 转换 1 的几何轴和通道轴的分配
  MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[0] = 1
  MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[1] = 3
  MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[2] = 2

TRANSMIT 特殊设定值：

- 回转轴的位移
  MD24900 TRANSMIT _ROT_AX_OFFSET_1=0
- 回转轴的符号
  MD24910 TRANSMIT _ROT_SIGN_IS_PLUS_1=1
- 基本刀具的矢量
  MD24920 TRANSMIT_BASE_TOOL_1[0]=0
  MD24920 TRANSMIT_BASE_TOOL_1[1]=0
  MD24920 TRANSMIT_BASE_TOOL_1[2]=0

特殊刀补下的设定数据（仅在必要时）：

- 平面切换时切换刀具长度分量
  SD42940 TOOL_LENGTH_CONST=18
- 刀具长度分量的分配与刀具类型无关
  SD42950 TOOL_LENGTH_TYPE=2

第二主轴的设定（车床的铣削主轴）：

- MD30300 IS_ROT_AX[AX4]=1

- MD30310 ROT_IS_MODULO[AX4]=1
- MD30320 DISPLAY_IS_MODULO[AX4]=1
- MD35000 SPIND_ASSIGN_TO_MACHAX[AX4]=2
- SD43300 ASSIGN_FEED_PER_REV_SOURCE[AX4]=0

说明

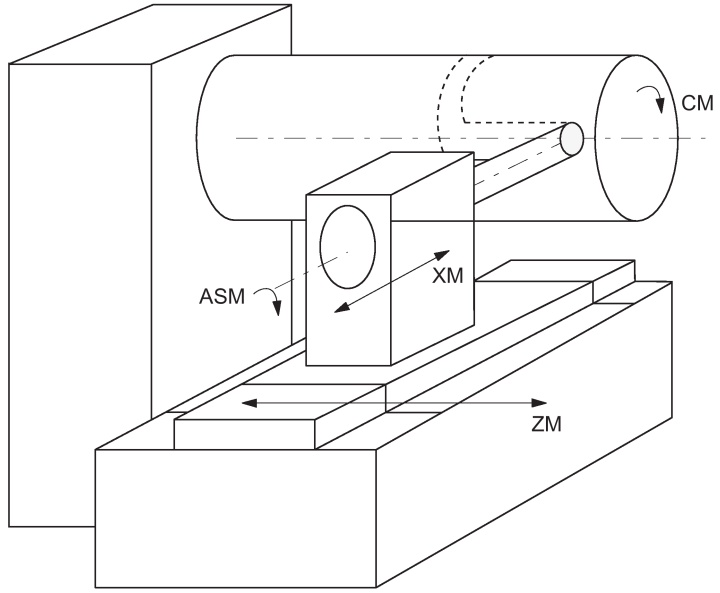
车床上铣刀的特殊处理只适用于长度补偿。

13.3 TRACYL

13.3.1 概览

标准车床（无 Y 进给轴）

使用 X-C-Z 运动在圆柱面加工槽：



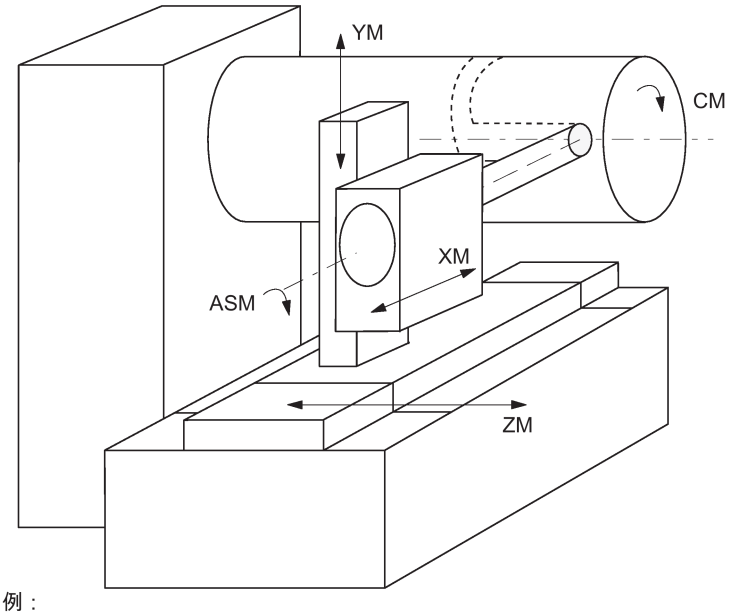
图例：
XM 垂直与回转轴的进给轴
ZM 平行于回转轴的进给轴
CM 回转轴
ASM 加工主轴

机床运动要求

两个线性轴（XM、ZM）必须相互垂直。回转轴（CM）必须和线性轴 ZM 平行（围绕 ZM 旋转）。线性轴 XM 和回转轴 CM 相交（车削中心）。

使用 Y 轴加工

使用 X-Y-Z-C 运动在圆柱面加工槽：



图例：
XM　垂直与回转轴的进给轴
YM　补充轴
ZM　平行于回转轴的进给轴
CM　回转轴
ASM　加工主轴

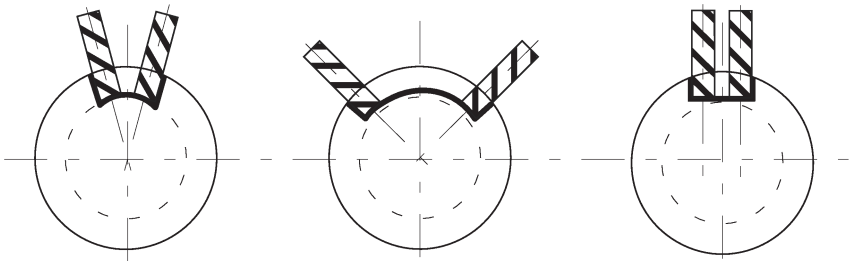
其它机床运动

处理所需的机床运动（参见以上），线性轴 YM 作为附加轴出现。它和 XM 和 ZM 垂直，并形成右手的笛卡儿坐标系。

这些运动主要用于铣床；可以用来加工的槽的槽面和槽低相互垂直 - 如果铣刀直径小于槽直径（槽面补偿）。否则，只有当铣刀直径精确匹配，才能加工这些槽。

槽横截面

有和没有槽壁补偿的键槽：



纵向槽
没有槽壁补偿
（没有 YM 轴）

横向槽

平行受限制的纵向槽
有槽壁补偿
（使用 YM 轴）

激活/取消 TRACYL

TRACYL 功能可以通过

- TRACYL(d) 的单独程序段激活
- TRAFOOF 的单独程序段取消

d - 圆柱体的加工直径，单位是毫米

使用 TRAFOOF 将取消任何有效的转换功能。

编程举例

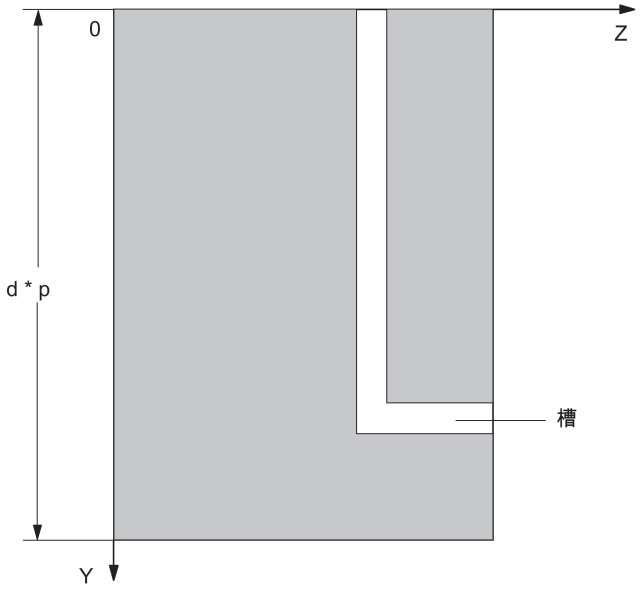
```
                                ; 没有 YM 轴
                                ; 编程几何轴 X、Y、Z
N10 G0 X...Z...SPOS=...         ; 初始位置，主轴在位置控制中
N20 G19 G94 T...                ; 平面、进给类型、选择铣刀
N30 SETMS(2)                    ; 转换：主主轴现在为铣削主轴
N40 TRACYL (24.876)             ; 激活 TRACYL，直径：24.876 毫米
N50 G1 F200 X...M3 S...         ; 进刀，接通铣削主轴
N600 G41 F200 Y...Z...          ; 使用刀具半径补偿铣削圆柱面
...
N90 G40 ...
N100 TRAFOOF                    ; 关闭 TRACYL
N110 G18 G95 T...               ; 返回车削加工
N120 SETMS                      ; 主主轴是车削主轴
```

说明：

根据所编程的 Y-Z 路径（直线或路径），移动机床进给轴 ZM 和 CM，使得在圆柱面上可以通过铣刀进行轮廓加工。编程的 X 轴（进给）仍然作为 X 轴进给。

圆柱面 G19（Y - Z 平面）：

沿着圆柱体的外直径 d 展开，从而得到
一个外表面，编程平面为 Y-Z (G19)。 并确定
G2, G3 时的圆弧旋转方向。



OFFN 定义

槽壁到参考轮廓的距离（参见“编程举例 TRACYL”）

编程：OFFN=... ; 间距，单位 mm

通常要编程槽中心线。使用刀具半径补偿时（G41、G42），OFFN 即定义了槽宽。槽加工结束后，设定 OFFN=0。

13.3.2 配置 TRACYL

一般机床数据

一般机床数据使用的机床进给轴，通道轴和几何轴的名称（$MN_AXCONF...和通道专用的机床数据 $MC_AXCONF...）也同样适用于转换功能。

只当转换功能无效时才可使用在 MD20050 AXCONF_GEOAX_ASSIGN_TAB 中定义的几何轴。如果转换功能有效，需进行特殊定义。

说明

用于转换功能的机床进给轴名称，通道轴名称和几何轴名称必须不同：

- MD10000 AXCONF_MACHAX_NAME_TAB
- MD20080 AXCONF_CHANAX_NAME_TAB
- MD20060 AXCONF_GEOAX_NAME_TAB

TRACYL 的例外：
MD20060 和 MD20080 的轴名称（几何轴和通道轴）用于转换 TRACYL 时可以相同，例如：X，Y，Z），如果未激活转换功能，则没有 Y 轴。这是车床的通常情况。

转换功能所需的机床数据

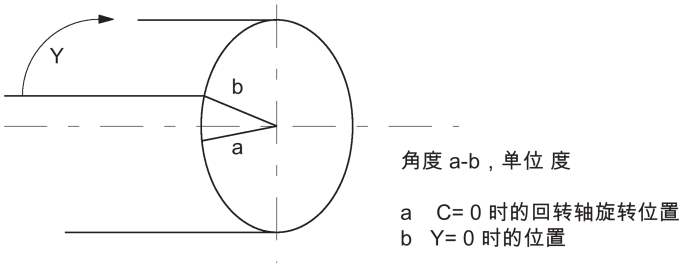
MD24100 TRAFO_TYPE_1	用于第一转换 TRACYL 的定义 *)
MD24110 TRAFO_AXES_IN_1[n]	转换 1 的通道轴
MD24120 TRAFO_GEOAX_ASSIGN_TAB_1[n]	转换 1 的几何轴
MD24200 TRAFO_TYPE_2	用于第二转换 TRACYL 的定义 *)
MD24210 TRAFO_AXES_IN_2[n]	转换 2 的通道轴
MD24220 TRAFO_GEOAX_ASSIGN_TAB_2[n]	转换 2 的几何轴
*) =512/513 带/不带 YM 轴	

机床数据 MD24110 中 TRACYL 转换功能所需的通道轴分配：

没有 YM 轴的配置：	
TRAFO_AXES_IN_1[0]=	回转轴径向轴的通道轴号
TRAFO_AXES_IN_1[1]=	回转轴的通道轴号
TRAFO_AXES_IN_1[2]=	平行于回转轴的通道轴号。
带 YM 轴的配置：	
TRAFO_AXES_IN_1[3]	平行于圆柱面以及垂直于 回转轴（-> YM 轴）的轴的通道轴号。

TRACYL 所需的特殊机床数据

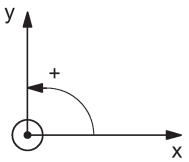
- MD24900 TRACYL_ROT_AX_OFFSET_1
  旋转位置：Y=0 的回转轴位置（0...<360）
  圆柱面中轴的旋转位置：



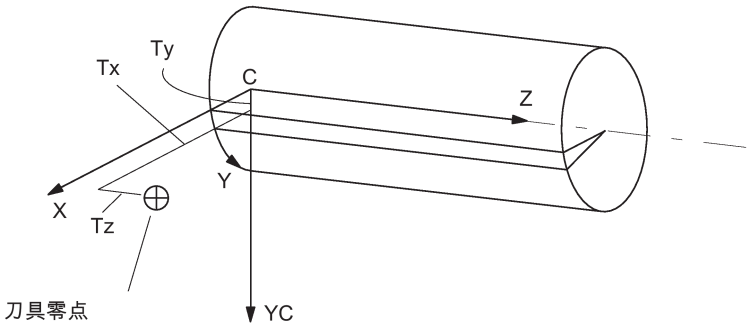
角度 a-b，单位 度

a C= 0 时的回转轴旋转位置
b Y= 0 时的位置

- MD24910 TRACYL_ROT_SIGN_IS_PLUS_1
  如果从 Z 轴的负方向观察，发现在 x-y 平面中的回转轴的旋转方向是逆时针，则将机床数据值设定为 1；否则，设为 0。
  MD 值 = 1 的旋转方向：



- MD24920 TRACYL_BASE_TOOL_1
  通知数控系统和相对于 TRACYL 使用的坐标系原点的刀具零点位置。此机床数据可以给笛卡儿坐标系中的三个进给轴定义轴元素。
  MD24920 中的轴元素分配：
  – TRACYL_BASE_TOOL_1[0]=Tx
  – TRACYL_BASE_TOOL_1[1]=Ty
  – TRACYL_BASE_TOOL_1[2]=Tz（参见下图）
  相对于机床零点的刀具零点位置：



示例：在标准机床上，使用 TRACYL 所需的机床数据设定值

通用设置：轴名称：XM->X1，ZM->Z1，CM->SP1

- 机床轴名称
  MD10000 AXCONF_MACHAX_NAME_TAB[0]="X1"
  MD10000 AXCONF_MACHAX_NAME_TAB[1]="Z1"
  MD10000 AXCONF_MACHAX_NAME_TAB[2]="SP1"
  MD10000 AXCONF_MACHAX_NAME_TAB[3]="SP2"
  MD10000 AXCONF_MACHAX_NAME_TAB[4]=""
- 分配几何轴给通道轴
  MD20050 AXCONF_GEOAX_ASSIGN_TAB[0]=1
  MD20050 AXCONF_GEOAX_ASSIGN_TAB[1]=0
  MD20050 AXCONF_GEOAX_ASSIGN_TAB[2]=2
- 通道中的几何轴名称
  MD20060 AXCONF_GEOAX_NAME_TAB[0]="X"
  MD20060 AXCONF_GEOAX_NAME_TAB[1]="Y"
  MD20060 AXCONF_GEOAX_NAME_TAB[2]="Z"
- 通道内有效的机床轴号
  MD20070 AXCONF_MACHAX_USED[0]=1
  MD20070 AXCONF_MACHAX_USED[1]=2
  MD20070 AXCONF_MACHAX_USED[2]=3
  MD20070 AXCONF_MACHAX_USED[3]=4
  MD20070 AXCONF_MACHAX_USED[4]=0
- 通道内的通道轴名称
  MD20080 AXCONF_CHANAX_NAME_TAB[0]="X"
  MD20080 AXCONF_CHANAX_NAME_TAB[1]="Z"
  MD20080 AXCONF_CHANAX_NAME_TAB[2]="C"
  MD20080 AXCONF_CHANAX_NAME_TAB[3]="SP2"
  MD20080 AXCONF_CHANAX_NAME_TAB[4]=""
- 通道内主主轴的复位位置
  MD20090 SPIND_DEF_MASTER_SPIND=1

用于第二转换的 TRACYL 类型：

- 没有槽壁补偿（没有 YM 轴）
  MD24100 TRAFO_TYPE_2=512
- 通道轴分配
  MD24110 TRAFO_AXES_IN_2[0]=1
  MD24110 TRAFO_AXES_IN_2[1]=3
  MD24110 TRAFO_AXES_IN_2[2]=2
  MD24110 TRAFO_AXES_IN_2[3]=0
  MD24110 TRAFO_AXES_IN_2[4]=0
- 分配几何轴给通道轴
  MD24120 TRAFO_GEOAX_ASSIGN_TAB_2[0]=1
  MD24120 TRAFO_GEOAX_ASSIGN_TAB_2[1]=3
  MD24120 TRAFO_GEOAX_ASSIGN_TAB_2[2]=2

TRACYL 特殊设定：

- 回转轴的位移
  MD24800 TRACYL_ROT_AX_OFFSET_1=0
- 回转轴的符号
  MD24810 TRACYL_ROT_SIGN_IS_PLUS_1=1
- 基本刀具的矢量
  MD24820 TRACYL_BASE_TOOL_1[0]=0
  MD24820 TRACYL_BASE_TOOL_1[1]=0
  MD24820 TRACYL_BASE_TOOL_1[2]=0

特殊刀补下的设定数据（仅在必要时）：

- 平面切换时切换刀具长度分量
  SD42940 TOOL_LENGTH_CONST=18
- 刀具长度分量的分配与刀具类型无关
  SD42950 TOOL_LENGTH_TYPE=2

第二主轴的设定（车床的铣削主轴）：

- MD30300 IS_ROT_AX[AX4]=1
- MD30310 ROT_IS_MODULO[AX4]=1
- MD30320 DISPLAY_IS_MODULO[AX4]=1
- MD35000 SPIND_ASSIGN_TO_MACHAX[AX4]=2
- SD43300 ASSIGN_FEED_PER_REV_SOURCE[AX4]=0

说明

车床上铣刀的特殊处理只适用于长度补偿。

13.3.3 TRACYL 编程举例

使用槽壁补偿加工槽

MD24100_TRAFO_TYPE_1 = 513

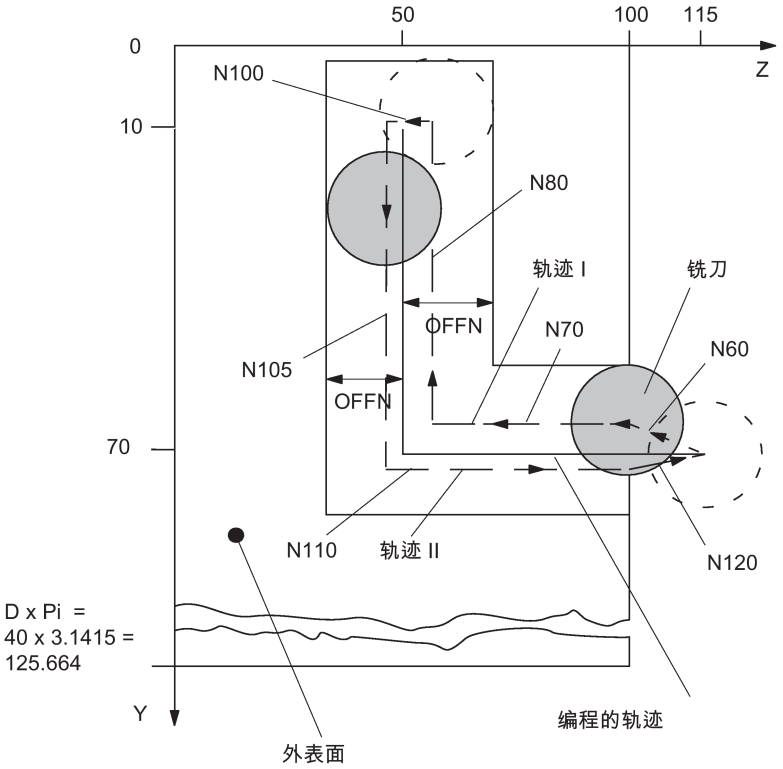
轮廓

如果要加工的槽的宽度大于刀具宽度，可以编程补偿方向（G41、G42）以及通过 **OFFN=...**定义槽壁到相对于编程的参考轮廓间的距离。

刀具半径

使用 G41、G42 自动考虑相对于槽壁的刀具半径。可以提供平面中刀具半径补偿的所有功能（外角和内角的连续转换，以及瓶颈问题的诊断）。

示例：带槽壁补偿的键槽图



TRACYL 用于在圆柱面铣削槽，其中“路径 I” 和“路径 II”的加工分别使用不同的 OFFN 值。

CC 是回转轴的通道轴名，T1、D1 的刀具半径：8.000 mm

```
G54 G94                    ；设置偏移量补偿及进给模式
G1 X50 Z50 F1000           ；移动至安全位置
T11 D1                     ； 刀具选择
DIAMOF                     ； 在半径模式下编程
TRACYL(40)                 ； 转换选择，参考直径
                           ； 对于表面：直径 40 mm
SETMS(2)                   ； 设置第二主轴作为主主轴
S2000M4                    ； 旋转铣削刀具
G19                        ； 加工平面为圆柱形表面 Y/Z
R1=20                      ； 起点
LL:                        ； 循环开始
G1 X=R1 F200               ； X 以 200 mm/min 的进给率移动就位
G41 OFFN=-3                ； 刀具半径补偿，轮廓左边
                           ； 定义槽壁偏移
Y10                        ； Y 移动至位置 10
Z-50                       ； Z 移动至位置 -50
Y70                        ； Y 移动至位置 70
G41 OFFN=-3                ； 刀具半径补偿，轮廓左边
                           ； 定义槽壁偏移
```

```
Y10                                  ; Y 移动至位置 10
Z10                                  ; Z 移动至位置 10
R1=R1-0.2                            ; 每次切削进给的切削深度为 0.2 mm
IF R1>15 GOTOB LL                    ; 切削深度为 5 mm
G40 Y0                               ;关闭补偿功能
TRAFOOF                              ; 关闭 TRACYL 功能
G1 X50 Z50 F1000                     ; 移动至安全位置
M30                                  ; 程序结束
```

13.4 TRANSMIT 和 TRACYL 的特点

POWER ON / RESET

上电或复位（程序结束）后的响应由下列机床数据中的设定值决定：

- MD20110 RESET_MODE_MASK（仅保护级 1 可对该机床数据进行存取）
- MD20140 TRAFO_RESET_VALUE（复位后生效的转换）。

选择功能时，遵守以下内容:

- 确保未选择刀具半径补偿（G40）。
- 数控系统取消选择 TRANSMIT/TRACYL 功能前的动作。（G500）。
- 数控系统将取消受转换功能影响的任何轴的有效工作区域极限（WALIMOF）。
- 连续路径控制和精磨被中断。
- 包含倒角或半径的任何中间的动作程序段将不插入。

取消功能时，遵守以下内容:

- 确保未选择刀具半径补偿（G40）。
- 连续路径控制和精磨被中断。
- 包含倒角或半径的任何中间的动作程序段将不插入。
- TRANSMIT/TRACYL 功能取消后,所有用于车削的零点偏移和设定值必须重新设定。

运行方式，运行方式切换

- 使用 TRANSMIT/TRACYL 功能时，程序在“AUTO”方式下执行。
- 可以中断“AUTO”方式并转换到“JOG”方式。当返回“AUTO”方式时，操作人员必须确保刀具可以毫无问题地重新定位。
- 转换功能有效时，进给轴不能回参考点。

13.5 数据表

13.5.1 机床数据

序号	名称	名称
通道专用		
20110	RESET_MODE_MASK	上电或复位/零件程序结束后，控制器初始位置的定义(存取保护级为 1)
20140	TRAFO_RESET_VALUE	初始化脉冲:复位后转换功能有效
22534	TRAFO_CHANGE_M_CODE	几何轴转换时输出的 M 代码
24100	TRAFO_TYP_1	第 1 转换的类型，可以定义轴顺序
24110	TRAFO_AXES_IN_1	第 1 转换输入时的轴分配
24120	TRAFO_GOEAX_ASSIGN_TAB_1	第 1 转换下的几何轴的分配
24200	TRAFO_TYP_2	第 2 转换的类型，可以定义轴顺序

序号	名称	名称
24210	TRAFO_AXES_IN_2	第 2 转换输入时的轴分配
24220	TRAFO_GOEAX_ASSIGN_TAB_2	第 2 转换下的几何轴的分配
24800	TRACYL_ROT_AX_OFFSET_1	回转轴偏离零点位置的角度(第 1 TRACYL)
24810	TRACYL_ROT_SIGN_IS_PLUS_1	TRACYL 下回转轴符号(第 1TRACYL)
24820	TRACYL_BASE_TOOL_1	刀具零点到几何轴原点距离(第 1 TRACYL)
24900	TRANSMIT_ROT_AX_OFFSET_1	回转轴偏离零点位置的角度 (第 1TRANSMIT)
24910	TRANSMIT_ROT_SIGN_IS_PLUS_1	TRANSMIT 下回转轴符号 (第 1TRANSMIT)
24911	TRANSMIT_POL_SIDE_FIX_1	极点前/后的工作区极限，第 1 转换
24920	TRANSMIT_BASE_TOOL_1	刀具零点到几何轴原点距离(第 1 TRANSMIT)

13.5.2 接口信号

序号	位	名称
通道专用		
DB3800.DBX0001	.6	转换有效

14 测量（M5）

14.1 概述

通道专用测量

在零件程序中，编程了测量循环(带或不带删除剩余行程)。而且，还编程了触发器事件（测量头脉冲沿）来触发测量过程。此测量适用于程序段中编程的所有的进给轴。包含测量过程的程序在“AUTO”方式下执行并可以用于工件测量或刀具测量。

JOG 方式下刀具测量

在数控系统中要完成一提示系统，该提示系统在“JOG”方式下有效并帮助操作人员完成测量过程。该测量过程包括通道专用测量。所需的功能必须集成在 PLC 用户程序中。测量过程结束时，所测的刀具的修正值存储在刀具偏移存储器中。

操作人员应进行哪些操作，参见 SINUMERIK 808D ADVANCED 编程和操作手册。

说明

仅在铣床上提供自动测量功能。

14.2 硬件前提条件

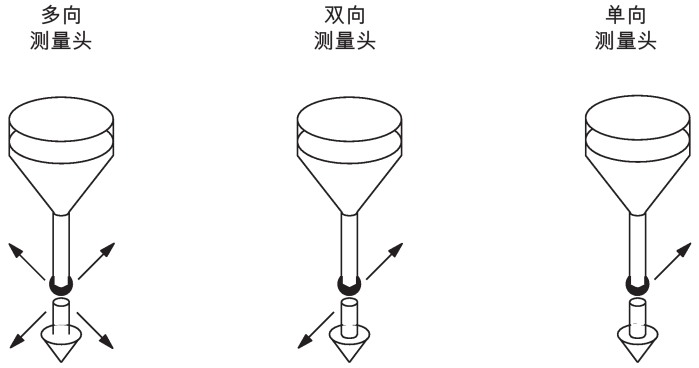
14.2.1 可使用的测量头

概述

为了获取刀具和工件尺寸，需要一个可切换的测量头，该测量头在换向时提供一个恒定的信号（非脉冲信号）。

测量头必须在几乎无振动情况下切换。通常使用可以机械校正的测量头。

市场上不同的制造商所设计的测量头有很多种。根据测量头可以偏移的方向数，分为三种类型（参见下图）。



下表给出了测量头类型分配信息：

测量头类型	铣削和加工中心
	工件测量
多向	X
双向	X
单向	X

对于铣床和加工中心，也可以使用单向测量头测量工件。

多向测量头(3D)

此种测量头可以用于工件测量而不受任何限制。

双向测量头

在铣床和加工中心测量工件时，该类型测量头作为单向测量头使用。

单向测量头

在铣床和加工中心，此类型测量头可以用于工件测量，但有一定的限制。

要在不同的轴方向测量时，必须可以使用 SPOS 功能定位主轴。必须按照所需进行的测量来调节测量头。

切换动作

MD13200 MEAS_PROBE_LOW_ACTIVE[0]用于向系统发送所连接测量头(扩展/无效)的信号级。

14.2.2 测量头连接

通过 X21 的引脚 4（DI1）和引脚 5（DI2）连接用于数控系统的测量头。实际使用哪个引脚可通过编写相关宏指令来确定。从而可以运行轴驱动模块的所有测量输入端，该轴为参与测量的轴。测量头应使用外部电压（24 V），其参考电位在 X21 引脚 10 上。

为获得最佳的抗干扰性，在连接测量头时需要使用导线。

文献：

SINUMERIK 808D ADVANCED 调试手册

14.3 通道专用测量

14.3.1 测量方式

测量指令 MEAS 和 MEAW

由零件程序激活测量过程。程序中编程了触发器和测量方式。有两种测量方式：

- MEAS:测量，带剩余行程删除
  示例：`N10 G1 F300 X300 Z200 MEAS=-1`

触发器事件为测量头 1 的下降沿（-）：从偏转状态到非偏转状态。

- **MEAW**:测量，不带剩余行程删除
  示例：`N20 G1 F300 X300 Y100 MEAW=1`
  触发器事件为测量头 1 的上升沿（+）：从非偏转状态到偏转状态。

当测量头信号到达或位置到达时，测量程序段即完成。使用下键可以取消测量功能：



文献：

SINUMERIK 808D ADVANCED 编程和操作手册

说明

如果在测量程序段中编程几何轴（WCS 中的进给轴），所有几何轴的测量值将被存储。

14.3.2 测量结果

在程序中读取测量结果

使用系统变量，可以在零件程序中读取测量指令的结果。

- 系统变量 **$AC_MEA[1]**
  查询测量任务的状态信号。
  变量在测量开始时被删除。当测量头到达触发器等级时（上升沿或下降沿），即设置变量。这样可以在零件程序中检查测量的结果。
- 系统变量 **$AA_MM[轴]**
  存取机床坐标系中的测量结果。在零件程序中读取。[轴]表示测量轴的名称（X、Y、...）。
- 系统变量 **$AA_MW[轴]**
  存取工件坐标系中的测量结果。在零件程序中读取。[轴]表示测量轴的名称（X、Y、...）。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

PLC 服务显示

执行如下操作可以控制测量信号：



接口信号“测量头 1 已激活”（DB2700.DBX0001.0）。

使用信号“测量有效” 可以显示轴的当前测量状态（测量程序段中包括此轴运行）。

14.4 测量精度和测试

14.4.1 测量精度

精度

测量信号的运行时间由所使用的硬件决定。 延迟时间为微秒数加上测量头的反应时间。

测量不精确度由以下公式得出：
测量不精确度 = 测量信号的运行时间 x 运行速度

只能保证进给速度的结果正确，在此速度时，每个位置控制器循环具有不超过 1 个触发信号到达。

14.4.2 测量头的功能测试

功能检查举例

建议使用 NC 程序执行测量头的功能测试。

```
%_N_PRUEF_MESSTASTER_MPF                    ; 测量头连接的检查程序
N10; R10                                    ; 用于控制状态的标志器
N20; R11:measurement in X axis
N30 T1 D1                                   ; 选择用于测量头的刀具补偿
N40 ANF:G0 G90 X0 F150                      ; 起始位置和测量速度
N50 MEAS=1 G1 X100                          ; X 中的测量输入端 1 处的测量
N60 STOPRE
N70 R10=$AC_MEA[1]                          ; 读取第 1 测量输入处的接通信号
N80 IF R10==0 GOTOF FEHL1                   ; 信号处理
N90 R11=$AA_MW[X]                           ; 读入工件坐标中的测量值
N95 M0
N100 M2
N110 FEHL1:MSG                              ; 未接通测量头！
N120 M0
N130 M2
```

举例: 重复精度

此程序可以计算整个测量系统(加工 - 测量头 - 信号传输)的测量分散带(重复精度或重复性)。

在示例中，在 X 轴上进行十次测量并将测量值接收到工件坐标中。

可以发现“随机尺寸偏差”，它并非是特定的趋势。

```
%_N_CHECK_ACCURATE_MPF
N05; R11                                    ; 接通信号
N06 R12=1                                   ; 计数器
N10; R1 到 R10                              ; MEAS_VAL_IN _X
N15 T1 D1                                   ; 起始条件，预先选择用于测量头的
                                            刀具补偿
N20 ANF:G0 X0 F150                          ; 在测量轴上预先定位
N25 MEAS=+1 G1 X100                         ; 在 X 轴的上升沿
                                            第 1 测量输入处测量
N30 STOPRE                                  ; 停止解码，接着分析
                                            ; 结果 ( 在读取 MEA 时自动
                                            ; 导出 )
N35 R11= $AC_MEA[1]                         ; 读取第 1 测量输入处的接通信号
N37 IF R11==0 GOTOF FEHL1                   ; 检查接通信号
N40 R[R12]=$AA_MW[X]                        ; 将测量值读入到工件坐标中
N50 R12=R12+1
N60 IF R12<11 GOTOB ANF                     ; 重复 10 次
N65 M0
N70 M02
N80 FEHL1:MSG                               ; 未接通测量头
N90 M0
N95 M02
```

选择了参数显示以后，可以读取 R1 到 R10 的测量结果。

14.5 JOG 方式下刀具测量

测量原理

操作人员使用进给键或手轮在“JOG”方式下移动刀具到测量头处。测量程序使用另一个测量头和其它定位过程来控制实际的测量过程。结束时，输入刀具补偿值。

优点：刀具测量前输入的补偿值可以完全和实际值不同。刀具不需要“预先测量”。

说明

不执行磨损测量，但始终重新测量刀具。

“JOG”方式下提供了软键和模板供用户使用。并可以帮助刀具测量。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

说明

PLC 用户程序必须以一定的顺序编制。功能之前不可用。

移动测量头时，需特别小心。因为测量头的行程有限，如果超出了该行程，测量头会损坏或完全毁坏。请按机床制造商的说明进行操作！

特别是，始终降低测量头的移动速度以便可以及时停止。“快速进给修调”不能使用。

显示屏幕格式，测量顺序取决于机床类型。可以测量以下刀具类型：

铣削工艺

- 铣刀（几何长度 1 和几何半径）
- 钻头（几何长度 1）

刀具补偿

屏幕中包含了首先是有限刀具 T，然后是测量结果输入目标的有限补偿号 D。可以通过接口由 PLC 定义不同的刀具或者由操作人员输入不同的 T 号和/或补偿号 D 来实现。

说明

如果输入了非有效值的刀具号或补偿号，在测量后使用此刀具/刀具补偿时，必须通知数控系统，如，通过编程并在“MDA”方式下执行。然后控制单元可以考虑正确的刀具补偿量。

由测量计算得到的**刀具长度补偿**自动输入到有效刀具的有效刀具补偿 D 的组件中。相关的组件“磨损”和“调节器”被删除。

测量**铣刀半径**时，假设在刀尖半径平面的轴中没有其它的偏移存在(值位调节器组件的进给轴中，并且几何长度 2 和 3 等于零)。半径的结果输入到几何组件中。平面中进给轴的相关组件“调节器”和“磨损”被删除。

测量头

用于刀具测量的测量头是一种固定的测量头或使用一机械装置可以在工作区中旋转。如果测量头板是长方形设计，必须在轴向调整边沿。刀具/校准刀具在测量头的反向移动。测量进行前，测量头必须已校准。即系统必须直到确切的相对于机床零点的测量头的切换点。

准备工作，校准测量头





1. 选择“JOG”方式 。
2. 使用该软键，在打开的窗口中输入以下值：
   后退平面，安全间隙，JOG 进给，可变增量尺寸和在 JOG 方式下或刀具测量时的主轴的旋转方向。
3. 按下该软键，在打开的窗口中输入以下值：
   - 在测量程序中测量头自动移动时的进给率。
   - 测量头切换点（校准时自动设定相应的值）。

   如果确切值已知，可以手动输入。因此无需进行校准。





4. 按下这两个软键，出现的窗口可以在校准测量头时提示操作人员。此时，使用的刀具为校准刀具并已输入已知的尺寸值。
   对于铣床，校准刀具是“铣刀”类型的一种。

校准时的内部顺序和测量时的一样。测量结果写入测量头切换点的数据 - 而不是刀具补偿。

说明

用于测量或校准的内部 NC 程序可以编程成当出现上升沿时开始测量。

测量顺序

选择“JOG”方式 。输入测量进给率。测量头已校准和/或已输入准确的测量头切换点。



1. 根据刀具类型，使用该软键和其它软键进行测量。



2. 按下该软键后，信号“JOG 有效时测量”（DB1700.DBX0003.7）从 HMI 传输到 PLC。PLC 可以通过信号“JOG 方式下刀具测量的 T 号”（DB1900.DBD5004）预设不同于有效刀具的其它 T 号，并且将该信号传给 HMI。当所选的进给轴移动时测量头切换，NCK 将输出信号“测量头 1 有效”（DB2700.DBX0001.0）。作为响应，PLC 将设定信号“取消进给”（DB3200.DBX0006.0）并且 NCK 停止动作。进给停止将一直保持，直到在 JOG 方式下按进给键且设置信号“JOG 方式下测量有效”（DB1700.DBX0003.7）。接着 PLC 将给出接口信号“复位”（DB3000.DBX000.7）。JOG 方式下的移动动作即被取消。
3. HMI 将发现测量头已切换，并发出命令到 PLC，要求按下进给键后（使用手轮进给 - - 立即）转换到“AUTO”，接口信号“自动方式”（DB1800.DBX0000.0）。PLC 将此信号传输给 NCK（DB3000.DBX0000.0）。NCK 将使能“AUTO”方式(接口信号“有效方式：自动”（DB3100.DBX0000.0）并在 HMI 屏幕上显示。PLC 将取消接口信号 “停止进给”（DB3200.DBX0006.0）。然后，HMI 给出接口信号“禁止改变运行方式”（DB1800.DBX0000.4）到 PLC。如果 PLC 发现此信号（只在一个 PLC 循环中出现），PLC 将发出接口信号“禁止方式转换“（DB3000.DBX0000.4）给 NCK。HMI 已经将 NC 测量程序载入到 NCK。现在可以将其使能。此测量程序将自动计算测量头的移动方向已经包括安全间隙的进给位移。HMI 将通过接口信号“在 JOG 方式下开始测量”（DB1800.DBX0000.6）向 PLC 发出命令，执行测量程序。在 V1800 范围中的信号只出现在一个 PLC 循环中。随后接口信号“在 JOG 下开始测量”在 PLC 中进行缓冲。通过输出信号“NC 启动”（DB3200.DBX0007.1）给 NCK，从 PLC 开始执行 NC 测量程序。
4. NC 程序将重新定位进给轴，重新接近测量头，测量并返回。HMI 将输出请求给 PLC，要求转换为“JOG”方式（DB1800.DBX0000.2）。作为响应，PLC 将复位接口信号“禁止方式转换”（DB3000.DBX0000.4）。然后 PLC 将输出“JOG”方式（DB3000.DBX0000.2）到 NCK，NCK 便回到接口信号“有效方式：JOG”（DB3100.DBX0000.2）。

测量头测量或校准结束时，使用以下软键取消选择此功能：



这将复位接口信号“在 JOG 方式下测量有效”（DB1700.DBX0003.0）。同样，退出操作区域也将复位此信号。可以使用接口信号“复位”（DB3000.DBX0000.7）取消自动程序，而且可以使用以下软键退出在 JOG 方式下的测量：



这也将清除接口信号“禁止进给”（DB3200.DBX0006.0）和接口信号“禁止方式转换”（DB3000.DBX0000.4），这些信号可能仍然被设置和/或可以清除缓冲区中的所有信号。

PLC 用户程序

用户必须在 PLC 用户程序中按照上面所列的步骤提供所需的功能。

西门子提供的 SINUMERIK 808D ADVANCED 工具盒中的 PLC 库中包含了用户样例程序。您可以使用此程序。使用时，必须调用在 OB1 中的 PLC_INI (SBR32) 和 MCP_NCK (SBR38), 因为它们将子程序 MEAS_JOG (SBR43)的信号传输给 NCK/HMI。

14.6 数据表

14.6.1 机床数据

序号	名称	名称
一般		
13200	MEAS_PROBE_LOW_ACTIVE	测量头的切换动作

14.6.2 接口信号

序号	位	名称
HMI 信号 (从 HMI 到 PLC)		
DB1700.DBX0003	.7	在 JOG 方式下测量有效
DB1800.DBX0000	.0	“AUTO”方式（来自 HMI 请求）
DB1800.DBX0000	.1	“MDA”方式（来自 HMI 请求）
DB1800.DBX0000	.2	“JOG”方式（来自 HMI 请求）
DB1800.DBX0000	.4	禁止方式改变(来自 HMI 请求)
DB1800.DBX0000	.6	在 JOG 方式下启动测量(来自 HMI 请求)
DB1800.DBX0001	.2	机床功能 REF(来自 HMI 请求)
HMI 信号 (从 PLC 到 HMI)		
DB1900.DBD 5004		在 JOG 方式下刀具测量的 T 号(PLC 定义)
一般(从 NCK 到 PLC)		
DB2700.DBX0001	.0	测量头 1 已激活
进给轴/主轴专用(从进给轴到 PLC)		
DB390x.DBX0002	.3	测量有效

15 手动刀具测量（带 Y 轴）

15.1 简介

在一车床上激活一附加轴并将其配置为 Y 轴后，可在该车床上测量刀具的 X 轴、Y 轴及 Z 轴。

有关如何在测量刀具前创建并激活刀具的详细信息，请参见 SINUMERIK 808D ADVANCED 编程和操作手册。

说明

带 Y 轴手动刀具测量的功能仅可用于车床。

15.2 手动测量刀具（带 Y 轴）的准备工作

配置 Drive Bus 总线地址

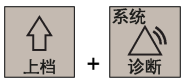
已在相应驱动上正确设置 Drive Bus 总线地址（p0918 = 12）。有关如何配置 Drive Bus 总线地址的详细信息，请参见 SINUMERIK 808D ADVANCED 调试手册。

激活附加轴

已激活至少一个附加轴。更多信息请参见章节“激活可选功能 (页 395)”。

设置 Y 轴参数

按以下步骤设置 Y 轴参数：



1. 选择系统数据操作区域。
2. 通过如下软键操作打开通用机床数据窗口：

3. 设机床数据 10000[1] 为 MY1。
4. 按下该软键打开通道机床数据窗口并设置以下机床数据：

   - 20050[1] = 2
   - 20050[2] = 3
   - 20060[1] = Y
   - 20070[1] = 2
   - 20070[2] = 3
   - 20070[3] = 4
   - 20080[1] = Y
   - 20080[2] = Z
   - 20080[3] = SP
5. 按下该软键激活变更的数据值。请注意，数控系统会进行重启以应用新的数据值，然后附加轴即显示在系统中。

6. 选择系统数据操作区域。
7. 通过如下软键操作打开轴机床数据窗口：

8. 使用软键切换至 MY1 轴的数据集。

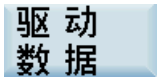
   

9. 使用该软键或光标键搜索以下机床数据并设置相应的值：

   - 30130[0] = 1
   - 30240[0] = 1（增量式编码器）或 4（绝对值编码器）
   - 31020[0] = 编码器分辨率（= 2500：增量式编码器；= 2048：绝对值编码器）
   - 34200[0] = 1（增量式编码器）或 0（绝对值编码器）
10. 按下该软键打开驱动数据列表。

11. 使用软键切换至 MY1 轴的数据集。

12. 通过该软键或光标键搜索驱动参数 p29000。输入连接至 MY1 轴的电机的 ID。电机 ID 可参见电机铭牌。



13. 按下该软键激活变更的数据值。请注意，数控系统会进行重启以应用新的数据值。

14. 如果必要，还可对 Y 轴的驱动性能进行优化。更多信息请参见 SINUMERIK 808D ADVANCED 调试手册。请注意，在开始驱动优化之前，必须先对 Y 轴执行回参考点操作。

15.3 手动测量刀具（带 Y 轴）

操作步骤

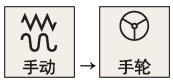
测量刀具的 X 轴



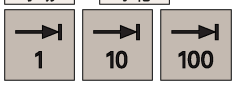
1. 选择加工操作区域。



2. 切换至“MDA”模式并在程序编辑器窗口中输入“SPOS=0”以固定主轴。



3. 切换至手轮控制模式。



4. 选择合适的进给倍率。

5. 使用手轮移动刀具，使刀具逼近工件，然后沿 Y 轴在工件表面切削约 1 mm 的深度。然后沿 Y 轴退刀。

   说明：

   开始切削之后请确保不在 X 轴方向上移动刀具。



6. 切换至“MDA”模式并在程序编辑器窗口中接着输入“SPOS=180”。



7. 按下该键使主轴旋转 180 度。

8. 重复步骤 3 至 5。

9. 使用卡尺测量在上述步骤中切削的两个表面之间的距离。



10. 选择偏移操作区域。

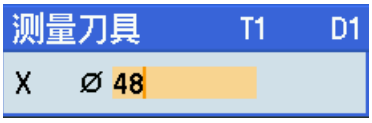


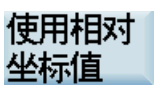
11. 按下字母键<X>或通过以下方法打开在 X 方向上测量刀具的窗口：

    使用光标键将光标移动至长度 X 的输入栏然后按下该软键。



12. 在输入栏中输入第 9 步所测得的距离，例如，48。





如需使用相对坐标值，按下该软键。



13. 按下该软键确认输入。系统自动计算补偿值并将其输入到当前激活刀具的几何输入栏。



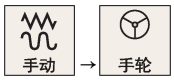
测量刀具的 Y 轴



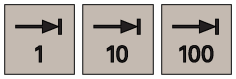
1. 选择加工操作区域。



2. 切换至“MDA”模式并在程序编辑器窗口中输入“SPOS=0”以固定主轴。



3. 切换至手轮控制模式。



4. 选择合适的进给倍率。

5. 使用手轮移动刀具，使刀具逼近工件，然后沿 X 轴在工件表面切削约 1 mm 的深度。然后沿 X 轴退刀。

说明：

开始切削之后请确保不在 Y 轴方向上移动刀具。



6. 切换至“MDA”模式并在程序编辑器窗口中接着输入“SPOS=180”。



7. 按下该键使主轴旋转 180 度。

8. 重复步骤 3 至 5。

9. 使用卡尺测量在上述步骤中切削的两个表面之间的距离。



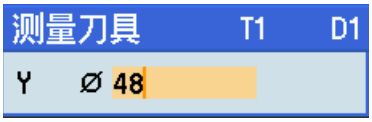
10. 选择偏移操作区域。



11. 按下字母键<Y>或通过以下方法打开在 Y 方向上测量刀具的窗口：
使用光标键将光标移动至长度 Y 的输入栏然后按下该软键。



12. 在输入栏中输入第 9 步所测得的距离，例如，“48”。

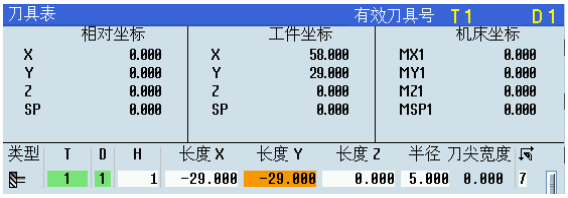




如需使用相对坐标值，按下该软键。



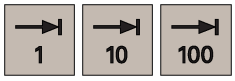
13. 按下该软键确认输入。系统自动计算补偿值并将其输入到当前激活刀具的几何输入栏。



测量刀具的 Z 轴



1. 选择加工操作区域。



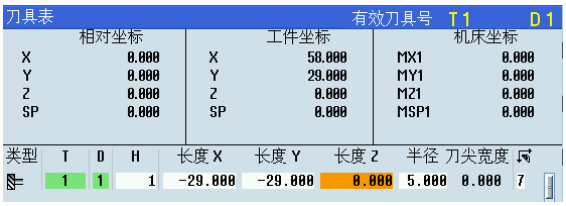
2. 选择合适的进给倍率，然后使用手轮沿 Z 轴移动刀具，使其刮碰到工件边沿（如使用垫块，则使其刮碰垫块边沿）。



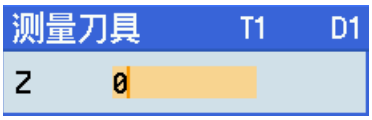
3. 选择偏移操作区域。

4. 按下字母键<Z>或通过以下方法打开在 Z 方向上测量刀具的窗口：
使用光标键将光标移动至长度 Z 的输入栏然后按下该软键。





5. 在输入栏中输入刀尖至工件边沿的距离，例如，“0”。（如使用垫块，则应输入垫块厚度。）

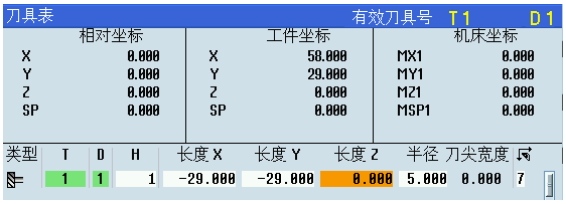


如需使用相对坐标值，按下该软键。





6. 按下该软键确认输入。系统自动计算补偿值并将其输入到当前激活刀具的几何输入栏。



7. 重复以上步骤，完成对所有刀具的测量。

16 急停（N2）

16.1 概述

说明

机床制造商应遵照相应的国际标准和国家标准(参见下面列出的有关标准)。数控系统为机床制造商实现符合该功能说明规定的急停功能提供支持。但急停功能如何具体实现(触发，运行以及应答)，这完全由机床制造商负责。

说明

实现急停功能时请参照以下标准：

- EN ISO 12100-1
- EN ISO 12100-2
- EN 418
- EN 60204-1

系统中急停装置

系统在生产时采取了以下措施，以便机床制造商方便地实现急停的功能：

- 通过 PLC 输入端在 NC 中执行急停。
- 在 NC 中急停时所有的进给轴和主轴都立即制动。
- 急停开关反弹后仍保持急停状态，急停状态不会因此而消失。给控制装置复位，不会导致重新启动。
- 急停状态取消后，无需让进给轴回参考点或让主轴同步(位置已调整)。

16.2 急停运行

前提条件

按急停键产生的信号首先必须作为 PLC 输入信号传送到 PLC 控制器，然后使用 PLC 用户程序通过接口信号“急停”(DB2600.DBX0000.1)继续传送到 NC。

急停键复位后所产生的信号首先必须作为 PLC 输入信号传送到 PLC 控制器。然后使用 PLC 用户程序通过接口信号“急停响应”(DB2600.DBX0000.2)继续传送到 NC。

NC 中的运行情况

EN 418 标准中预定的急停状态内部功能运行表现在数控系统中就是：

1. 零件程序的运行停止。所有进给轴和主轴按照机床数据 MD36610 AX_EMERGENCY_STOP_TIME 定义的制动斜坡进行制动。
2. 接口信号“808D - READY”(DB3100.DBX0000.3)复位。
3. 设置接口信号“急停有效”(DB2700.DBX0000.1)。
4. 设置报警 3000。
5. MD36620 SERVO_DISABLE_DELAY_TIM(控制器使能断开延时)中所设定的时间结束之后关闭调节器使能。此时应注意，给定的 MD36620 必须至少和 MD36610 一样大。

在机床上的运行情况

机床上急停运行的情况只能由机床制造商确定。关于数控系统的运行情况，需要注意：

- 用接口信号“急停”(DB2600.DBX0000.1)启动 NC 的急停运行。在进给轴和主轴停止之后，应根据 EN418 标准断电。
- PLC 外设（数字输出端）不受数控系统运行的影响。如果在急停时让某一输出端处于某一特定状态，则机床制造商必须在 PLC 用户程序中设置这样的功能。

说明

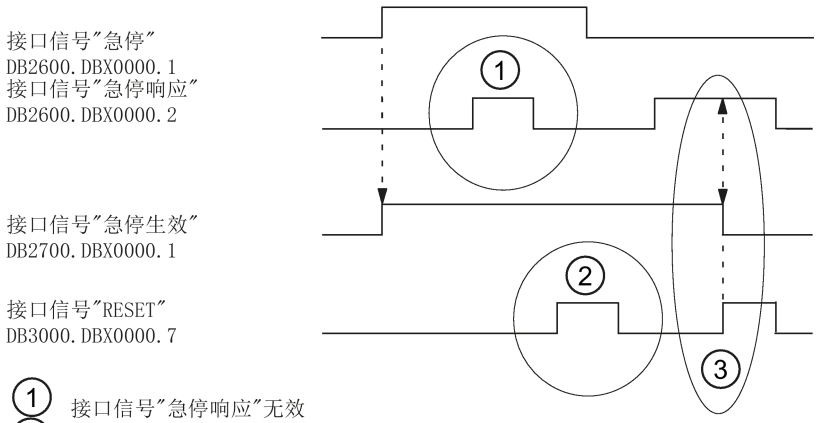
能源供应的切断由机床制造商负责。
如果要使急停时的 NC 运行不按照所规定的方式进行，则在机床制造商通过 PLC 用户程序所确定的急停状态出现之前，不得设置接口信号“急停”(DB2600.DBX0000.1)。只要接口信号“急停”没有设置，并且也没有出现其它报警，则 NC 中所有的接口信号就有效。由此可以使用每一个用户设置的急停状态。

16.3 急停响应

急停响应

只有首先设置接口信号“急停响应”(DB2600.DBX0000.2)，然后设置信号“复位”(DB3000.DBX0000.7)之后，才能重新复位急停状态。在此要注意的是，接口信号“急停响应”和“复位”必须一起设置很长时间，至少必须等到接口信号“急停有效”(DB2700.DBX0000.1)复位之后。

复位急停：



① 接口信号"急停响应"无效
② 接口信号"RESET"无效
③ 接口信号"急停响应"和"RESET"复位"急停生效"

复位急停状态会导致：

- 复位“急停有效”接口信号
- 接通控制器使能
- 设置“位置调节有效”接口信号
- 设置“808D - READY”接口信号
- 取消报警 3000
- 终止零件程序执行

PLC I/O

PLC 外设必须通过 PLC 用户程序重新设置到机床运行时的正常状态。

复位

仅用“复位”(DB3000.DBX0000.7)信号不能使急停状态复位(参见上图)。

电源开/关

除非接口信号 “急停”（DB2600.DBX0000.1）仍处于设置状态，否则可以通过电源断开/接通（POWER ON）操作清除急停状态。

16.4 数据表

16.4.1 机床数据

序号	名称	名称
轴专用		
36610	AX_EMERGENCY_STOP_TIME	出错情况下制动斜坡持续时间
36620	SERVO_DISABLE_DELAY_TIME	伺服无效延迟时间

16.4.2 接口信号

序号	位	名称
概述		
DB2600.DBX0000	.0	急停时停留在轮廓上
DB2600.DBX0000	.1	急停
DB2600.DBX0000	.2	急停响应

序号	位	名称
DB2700.DBX0000	.1	急停生效
运行方式—信号范围		
DB3000.DBX0000	.7	复位

17 回参考点（R1）

17.1 基础部分

为何要回参考点？

为了使系统在开机以后能够立即精确地识别机床零点，必须使系统与进给轴的位置测量系统进行同步。该过程就是所谓的回参考点过程。

主轴的回参考点过程（同步）在章节“主轴（S1） (页 158)”中说明。

位置测量系统

以下位置测量系统可以安装在电机或机床上：

- 增量式旋转测量系统
- 绝对式旋转测量系统

位置测量系统的回参考点可以使用 MD34200 ENC_REFP_MODE（“REF.PIOINT”方式）来设置。

凸轮输出

线性轴的回参考点需要加速档块，它有以下用途：

- 接近零脉冲时（同步脉冲）选择进给方向
- 在需要时选择零脉冲

BERO

可以使用 BERO（感应接近开关）作为编码器用于同步脉冲（代替位置编码器零脉冲）（优先用于旋转轴，主轴）。它可以通过端子 X21 的引脚 6（DI3）连接到数控系统上。

说明

只有在通过 PPU 接口 X54 连接了模拟量主轴驱动（此时没有连接 V70 主轴驱动）且通过 PPU 接口 X60 连接了电机编码器时，才可使用 BERO 接近开关。

文献：

SINUMERIK 808D ADVANCED 调试手册

接口信号“有效机床功能 REF”（DB3100.DBX0001.2）

使用机床功能 REF（信号“有效机床功能 REF”）回参考点。可以在“JOG”方式下选择机床功能 REF（信号“有效机床功能 REF”（DB3000.DBX0001.2））。

进给轴专用回参考点

每个进给轴使用 NST “进给轴键正/负”（DB380x.DBX0004.7 /.6）回参考点。所有轴可以同时回参考点。如果要以一定的顺序回参考点，需遵守以下内容：

- 操作人员必须遵守启动顺序。
- PLC 用户程序必须检查启动顺序或自动定义。
- 顺序定义在 MD34110 REFP_CYCLE_NR（参见通道专用回参考点）

通道专用回参考点

使用接口信号“使能回参考点”（DB3200.DBX0001.0）启动通道专用回参考点运行。系统使用信号“回参考点有效”（DB3300.DBX0001.0）响应成功启动。使用通道专用回参考点时，可以使每个通道轴回参考点（为此，系统内部模拟进给键正/负）。使用轴专用 MD34110 REFP_CYCLE_NR（通道专用的回参考点中的轴顺序）可以确定进给轴按何种顺序回参考点。如果在 MD34110 REFP_CYCLE_NR 中定义的所有轴已经回参考点，则输出信号“所有轴已回参考点”（DB3300.DBX0004.2）。

特点

- 用接口信号“复位”（DB3000.DBX0000.7）可以终止回参考点运行。此时未到达参考点的所有进 给轴被认为未回参考点。同时显示相应的报警 20005 并复位信号“回参考点有效”。
- 工作区极限和软件限位开关监控只对已回参考点的机床轴有效。
- 在回参考点时，在任一时刻均按照给定的加速度运行（有报警时除外）。
- 启动回参考点运行时，方向键仅对 MD34010 REFP_CAM_DIR_IS_MINUS 中设定的方向有效。

零件程序中回参考点

一个或多个进给轴可以同时回参考点，因为这些轴在操作过程中丢失了参考点。操作顺序对应于进给轴专用回参考点，没有使用进给键正/负启动回参考点，而是使用 G74 命令和机床轴标识。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

说明

MD20700 REFP_NC_START_LOCK=1 可以停止执行零件程序（输出报警），除非所有指定轴已回参考点。

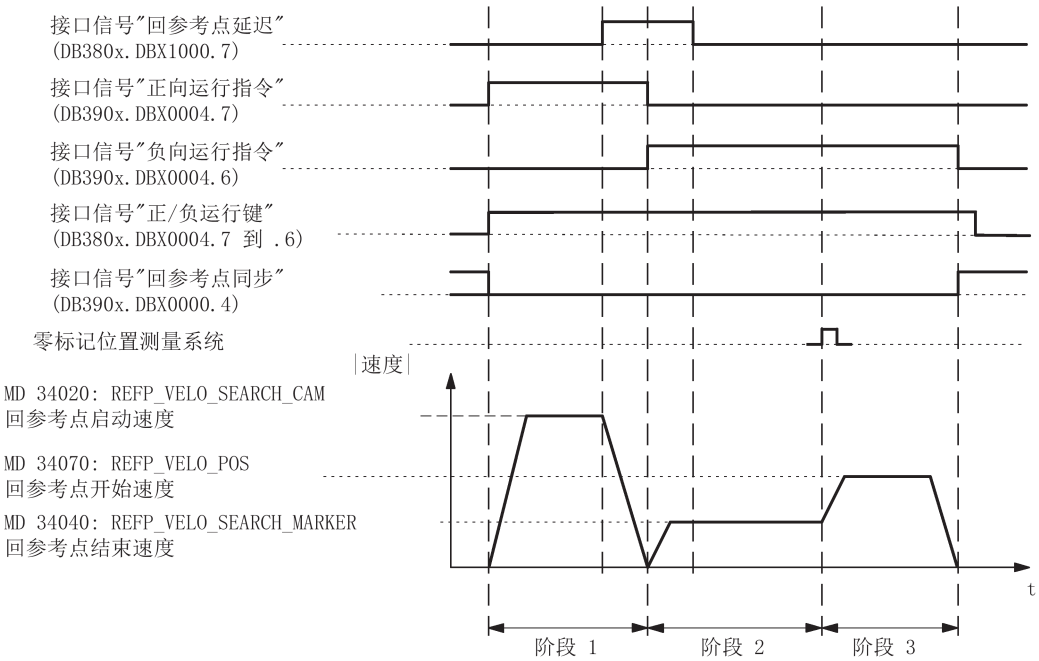
17.2 在增量测量系统中回参考点

时序过程

使用增量测量系统回参考点时，其过程可以分为如下三个阶段：

1. 相位：运行到减速挡块
2. 相位：与零脉冲同步
3. 相位：寻找参考点

使用增量测量系统回参考点时序(示例)：



寻找参考点减速挡块时的特性（阶段 1）

- 进给率修调和进给停止生效。
- 机床轴可以停止/启动。
- 必须在 MD34030 REFP_MAX_CAM_DIST 中定义的进给范围内到达减速挡块。否则输出相应的报警。
- 机床轴必须在到达挡块时停顿。否则输出相应的报警。

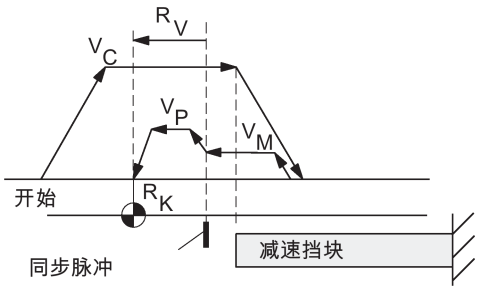
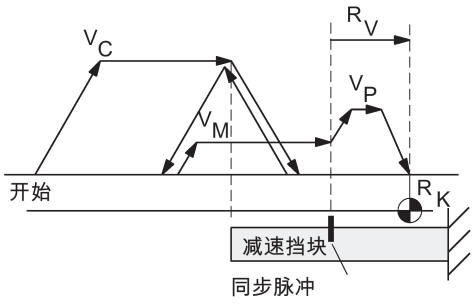
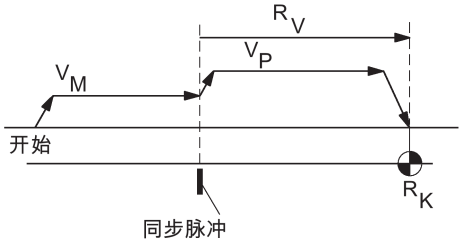
零脉冲同步时的特点（阶段 2）

- 进给修调不生效。使用 100% 进给率修调。如果进给率修调为 0%，移动即被取消。
- 进给停止生效；进给轴将停止并显示相应报警。
- 不能使用 NC 停止/启动键使进给轴停止/启动。
- MD34060 REFP_MAX_MARKER_DIST 监控零脉冲有效。

寻找参考点时的特性（阶段 3）

- 进给率修调和进给停止生效。
- 进给轴可以随 NC 一起停止/启动。
- 如果参考点偏移量小于进给轴从参考点运行速度到静止状态的制动路径，将从反方向回参考点。

回参考点时不同的运行过程：

回参考点方式	同步脉冲 (零脉冲，接近开关 BERO)	运行顺序
带参考减速挡块(MD34000 REFP_CAM_IS_ACTIVE = 1)	同步脉冲在减速挡块之前，参考点坐标在同步脉冲之前 = 无反向： (MD34050 REFP_SEARCH_MARKER_REVERSE = 0)	
	同步脉冲在减速挡块上，参考点坐标在之后 = 反向时： (MD34050 REFP_SEARCH_MARKER_REVERSE = 1)	
不带参考减速挡块(MD34000 REFP_CAM_IS_ACTIVE = 0)	参考点在同步脉冲之后	

回参考点方式	同步脉冲 (零脉冲，接近开关 BERO)	运行顺序
V_C- 参考点返回速度(MD34020 REFP_VELO_SEARCH_CAM) V_M- 参考点关断速度(MD34040 REFP_VELO_SEARCH_MARKER) V_P- 参考点逼近速度(MD34070 REFP_VELO_POS) R_V- 参考点偏移(MD34080 REFP_MOVE_DIST + MD34090 REFP_MOVE_DIST_CORR) R_K- 参考点坐标(MD34100 REFP_SET_POS)		

减速挡块必须至少多长？

该情况举例：同步脉冲在减速挡块之前，参考点在同步脉冲前 = 使用挡块下降沿寻找同步脉冲。

减速挡块必须足够长，使得用“寻找减速挡块速度”寻找减速挡块时可以在挡块上结束制动过程(在挡块上停动)，并且在用“寻找接近开关信号速度”在反方向运行时又再次离开挡块(恒定速度)。

计算挡块的最短长度时请使用其中较大的速度按下列公式计算：

$$\text{最小长度} = \frac{(\text{回参考点开始速度或者结束速度})^2}{2 \times \text{轴加速度(MD 32300: MAX_AX_ACCEL)}}$$

如果机床轴不能在减速挡块上停止（信号“回参考点减速挡块”(DB380x.DBX1000.7)已经复位），则发出报警 20001。如果减速挡块过短，并且在阶段 1 机床轴制动时运行超出减速挡块，则会出现 20001 报警。

如果减速挡块很长，可以达到进给轴的运行范围界限，这也就避免选一个不允许的回参考点起点(在挡块之后)。

减速挡块的调节

减速挡块必须精确调节。以下几点对识别减速挡块的时间性能（信号“参考点运行延迟”）具有影响：

- 减速挡块开关的精度
- 减速挡块开关的时间延迟(常闭接点)
- PLC 输入端时间延迟
- PLC 循环周期时间
- 内部处理时间

实践证明：调节两个同步脉冲信号的中间沿(或零脉冲)以达到同步是最好的方法。过程如下：

- 设置 MD34080 REFP_MOVE_DIST = MD34090 REFP_MOVE_DIST_CORR = MD 34100 REFP_SET_POS = 0。
- 轴回参考点
- 在“JOG”方式下，在两个零脉冲之前使轴移动一半的路径。路径取决于主轴丝杠的螺距 S 和变速系数 n（例如：S=10 毫米/转, n=1:1 得出 5 毫米路径）。
- 调节挡块开关，使它在此位置准确进行开关动作（信号”回参考点延迟”(DB380x.DBX1000.7)）。
- 除了移动挡块开关，还可以修改 MD34092 REFP_CAM_SHIFT 的值。

⚠警告

因减速挡块没有精确调节而无法保护机床

如果减速挡块没有精确调节，则可能会计算一个错误的同步脉冲（零脉冲）。因此控制器接受一个错误的机床零点，并使进给轴运行到错误的位置。这样，软件限位开关就在错误的位置上生效，不能对机床进行保护。

- 精确调节参考点挡块。

17.3 绝对编码器的边界条件

17.3.1 校正绝对值编码器

校正时间

通过校正计算出机床零点和编码器零点之间的偏移量并将它存储在稳定的存储器中。通常，只需在初次开机调试时进行一次校正。然后系统知道该值并可以在任何时候通过编码器绝对值计算出绝对机床位置。可以通过设定 MD34210 ENC_REFP_STATE=2 来标识该状态。

偏移量保存在 MD34090 REFP_MOVE_DIST_CORR 中。

必须在出现以下情况时重复校正过程：

- 拆除/安装或更换编码器或内装有编码器的电机后。
- 电机（带有绝对值编码器）和负载的变速换档后。
- 通常，编码器和负载间的极限连接被断开且还未重新连接时。

说明

系统不能发现需要重新校正的所有情况。如果系统发现某些情况，会设定机床数据 MD34210 的值为 0 或 1。系统能够识别以下情况：变速换档，该变速档在编码器和负载间具有不同的变速比。

在其它情况下，用户自己必须覆盖机床数据 MD34210。

数据备份

数据保存也同时保存 MD34210 ENC_REFP_STATE 的状态。

通过载入该数据记录，表示进给轴已自动校正！

说明

如果数据记录来自其它机床（如批量调试机床时），当数据载入和使能后，必须进行校正。

17.3.2 使用绝对值编码器回参考点

设置

运行使能

机床轴将校准过的绝对值编码器作为生效测量系统时，若为其触发回参考点运行（在 JOG-REF 运行方式下手动触发，或通过零件程序指令 G74 自动触发），那么该轴的运动特性取决于设置的运行使能：

MD34330 $MA_REFP_STOP_AT_ABS_MARKER = <值>

值	含义
0	运行已**使能**。 触发回参考点后，机床轴运行至回参考点位置。到达参考点后，回参考点运行结束。
1	运行**未使能**。 触发回参考点后，机床轴不运行，回参考点运行立即结束。

17.4 数据表

17.4.1 机床数据

序号	名称	名称
通道专用		
20700	REFP_NC_START_LOCK	不带参考点的 NC 启动禁用
轴专用		
30200	NUM_ENCS	编码器数量
30240	ENC_TYP[0]	实际值编码器类型

序号	名称	名称
30330	MODULO_RANGE	模数范围的大小
31122	BERO_DELAY_TIME_PLUS[0]	正向上的 BERO 延迟时间
31123	BERO_DELAY_TIME_MINUS[0]	负向上的 BERO 延迟时间
34000	REFP_CAM_IS_ACTIVE	带减速挡块的轴
34010	REFP_CAM_DIR_IS_MINUS	负向回参考点
34020	REFP_VELO_SEARCH_CAM	回参考点速度
34030	REFP_MAX_CAM_DIST	到达减速挡块的最大位移
34040	REFP_VELO_SEARCH_MARKER[0]	回参考点停止速度
34050	REFP_SEARCH_MARKER_REVERSE[0]	反向寻找零脉冲
34060	REFP_MAX_MARKER_DIST[0]	（至参考标记的最长路段）；至 使用间距编码尺寸的 2 个参考标记的路段
34070	REFP_VELO_POS	回参考点运行速度
34080	REFP_MOVE_DIST[0]	参考点距离/带距离编码系统的目标位置
34090	REFP_MOVE_DIST_CORR[0]	参考点/绝对值偏移，清除编码
34092	REFP_CAM_SHIFT[0]	具有等距离零标记增量测量系统的电子减速档块。
34093	REFP_CAM_MARKER_DIST[0]	参考挡块/参考标记距离
34100	REFP_SET_POS[0]...[3]	参考点值
34110	REFP_CYCLE_NR	通道专有回参考点时的轴顺序
34210	ENC_REFP_STATE[0]	绝对值编码器状态
34220	ENC_ABS_TURNS_MODULO[0]	旋转编码器的绝对值范围
36300	ENC_FREQ_LIMIT[0]	编码器极限频率
36302	ENC_FREQ_LIMIT_LOW[0]	编码器极限频率重新同步
36310	ENC_ZERO_MONITORING[0]	零标记监控

17.4.2 接口信号

序号	位	名称
运行方式-专用		
DB3000.DBX0001	.2	机床功能 REF
DB3100.DBX0001	.2	有效机床功能 REF
通道专用		
DB3200.DBX0001	.0	回参考点激活
DB3300.DBX0001	.0	回参考点有效
DB3300.DBX0004	.2	所有应回参考点的进给轴已回参考点
轴专用		
DB380x.DBX0000	.5	位置编码器 1
DB380x.DBX0004	.6 和 .7	运行键 -/+
DB380x.DBX1000	.7	回参考点延迟
DB390x.DBX0004	.6 和 .7	运行指令 -/+

18 主轴（S1）

18.1 概述

主轴功能

由 NC 控制的主轴根据不同的机床类型有可能具有如下的功能：

- 预置主轴旋转方向(M3,M4)
- 预置主轴转速(S)
- 主轴无定向准停(M5)
- 主轴定位(SPOS=)
  （主轴应为位置控制的主轴）
- 变速换档(M40 到 M45)
- 切削螺纹/攻丝(G33, G34, G35, G331, G332, G63)
- 旋转进给(G95)
- 恒定切削速度(G96)
- 在主轴或电机上可以安装位置测量编码器
- 可以监控主轴转速的极大值和极小值
- 主轴暂停旋转(G4 S)

如果采用的不是控制主轴，而是“级联”主轴， 则主轴转速(S)**不是**通过程序预置，而是通过机床上的手动操作(减速箱)进行控制。 这样，也就不可以编程转速极限。 通过程序可以进行以下设定：

- 预置主轴旋转方向(M3,M4)
- 主轴无定向准停(M5)
- 攻丝(G63)

如果主轴上还有一个位置编码器，则主轴还有其它功能：

- 切削螺纹/攻丝 (G33, G34, G35)
- 旋转进给(G95)

对于级联主轴，不可以通过设定机床数据给主轴输出给定值(MD30130 CTRLOUT_TYPE = 0)。

主轴定义

通过设定以下机床数据可以将一机床进给轴定义为主轴：

- MD30300 IS_ROT_AX
- MD30310 ROT_IS_MODULO
- MD30320 DISPLAY_IS_MODULO
- MD35000 SPIND_ASSIGN_TO_MACHAX.

主轴模式通过接口信号“主轴/无进给轴”(DB390x.DBX0000.0)来显示。

18.2 主轴运行方式

18.2.1 主轴运行方式

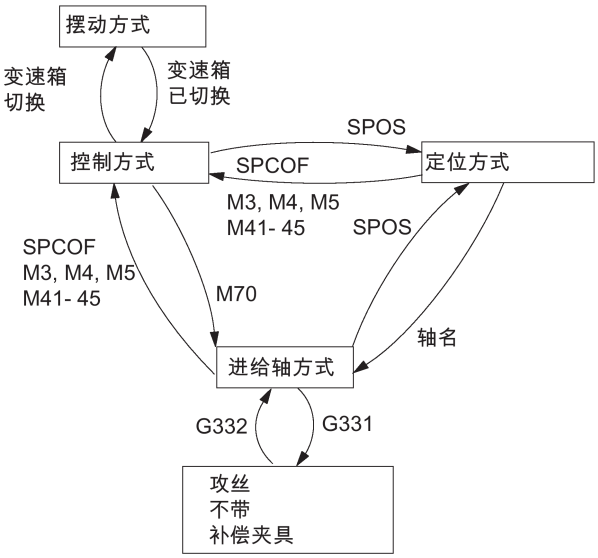
主轴运行方式

主轴可能具有以下运行方式：

- 控制运行参见“主轴控制方式运行 (页 159)”
- 摆动运行参见“主轴摆动方式运行 (页 160)”

- 定位运行参见“主轴定位方式运行 (页 161)”
- 进给轴方式
- 无补偿夹具攻丝(刚性攻丝)参见章节“进给率 (页 174)”

在主轴运行方式之间切换：



在主轴运行方式之间切换

- 控制运行方式 --> 摆动运行方式

  如果通过变速档自动选择(M40)及新的 S 值，或者通过 M41 至 M45 变换为一个新的变速档，则主轴从控制运行方式变换为摆动运行方式。只有当新的变速档不等于当前实际变速档时，才可以从控制运行方式变换为摆动运行方式。
- 摆动运行方式 --> 控制运行方式

  如果已经转换变速档，则接口信号“摆动方式”(DB390x.DBX2002.6)复位，并发出接口信号(DB380x.DBX2000.3)“变速档已经转换”，运行方式变换为控制方式。最后编程的主轴转速(S 值)再次生效。
- 控制运行方式 --> 定位运行方式

  如果要使主轴由旋转状态(M3 或 M4)定位停止，或者从停止状态(M5)取新的方向,则用 SPOS、SPOSA 及 M19 使运行方式变换为定位方式。
- 定位运行方式 --> 控制运行方式

  如果主轴定位完成，则用 SPCOF、M3、M4、M5 及 M41-45 转换到控制方式。最后编程的主轴转速(S 值)再次生效。
- 定位运行方式 --> 摆动运行方式

  如果要结束主轴定向，则可以通过 M41 至 M45 切换到摆动运行方式。换档结束以后，最后编程的主轴转速(S 值)和 M5(控制方式)再次生效。
- 定位控制运行 ---> 攻丝，不带补偿夹具

  无补偿卡盘攻丝（刚性攻丝）使用 G331/G332 激活无补偿卡盘攻丝(螺旋插补)。首先必须使用 SPOS 将主轴转为位置控制模式。

18.2.2 主轴控制方式运行

何时为控制方式？

在执行下述功能时主轴处于控制方式：

- 恒定主轴转速 S, M3/M4/M5 和 G94, G95, G97, G33, G63
- 恒定切削速度 G96 S，M3/M4/M5

前提条件

当旋转进给(G95，F 单位为毫米/转或英寸/转)，恒定切削速度(G96，G97)及加工螺纹(G33)按 M3/M4/M5 运行时，要求必须具备主轴位置实际值编码器。

主轴自身复位

主轴在复位以后或程序结束以后(M2，M30)具有何种特性，可以通过机床数据 MD35040 SPIND_ACTIVE_AFTER_RESET(主轴自身复位)进行设定：

- 如果 MD 值 = 0，则主轴立即以有效的加速度制动到停止。 最后编程的主轴转速和主轴方向被清除。
- 如果 MD 值 = 1(主轴自身复位)，则最后编程的主轴转速(S 功能)和方向(M3,M4,M5)继续保持。 如果在复位之前或者在程序结束之前恒定切削速度 G96 有效，则当前的主轴转速(主轴修调为 100％)在内部作为最后编程的主轴转速。

说明

主轴可以始终用接口信号“剩余行程清除/主轴复位”制动。
但要注意： G94 时程序继续运行！ 在 G95 时若缺少进给率进给轴也会停止，程序运行同样停止(若 G1,G2,...有效)。

18.2.3 主轴摆动方式运行

启动摆动方式

通过摆动方式运行使变速换档变得方便容易。通常，不在摆动方式也可以更换变速档。

如果通过自动变速档选择 (M40) 或者通过 M41 至 M45 预置一个新的运行变速档(接口信号“变速换档”(DB390x.DBX2000.3)已经设置)，则主轴处于摆动运行方式。只有当预置的变速档不同于当前的变速档时，才会设置接口信号“变速换档”。主轴摆动方式运行通过接口信号“摆动速度”(DB380x.DBX202.5)启动。

如果仅设置接口信号“摆动速度”而不预置一个新的变速档，则不会转换到摆动方式运行。

摆动方式通过接口信号“摆动速度”启动。但在执行功能时根据有无信号“通过 PLC 摆动”(DB380x.DBX2002.4)分为：

- 通过 NCK 摆动
- 通过 PLC 摆动

摆动时间

摆动运行时每个方向的摆动时间通过下面的机床数据设定：

- M3 方向的摆动时间（以下称为 t1）
  MD35440 SPIND_OSCILL_TIME_CW
- M4 方向的摆动时间（以下称为 t2）
  MD35450 SPIND_OSCILL_TIME_CCW

通过 NCK 摆动

阶段 1：通过接口信号“摆动速度”(DB380x.DBX2002.5)，主轴电机加速到 MD35400 SPIND_OSCILL_DES_VELO(摆动速度)中设定的速度(使用摆动加速度)。启动方向由 MD35430 SPIND_OSCILL_START_DIR (摆动时启动方向)中的数据决定。

阶段 2：时间 t1(t2)运行结束后，主轴电机在相反的方向加速到 MD35400 SPIND_ OSCILL_DES_VELO(摆动速度)中所设定的速度。开始运行时间 t2 (t1)。

阶段 3：时间 t2(t1)运行结束后，主轴电机又在相反的方向(与阶段 1 的方向一样)进行加速，等等。

通过 PLC 摆动

通过接口信号“摆动速度”(DB380x.DBX2002.5)，主轴电机加速到 MD35400 SPIND_OSCILL_DES_VELO(摆动速度)中设定的速度(使用摆动加速度)。

旋转方向通过“逆时针旋转”和“顺时针旋转”确定 (DB380x.DBX2002.7 或 .6)。

摆动(摆动运行)和两个时间 t1 和 t2 (顺时针方向和逆时针方向运行时间)必须在 PLC 中模拟出来。

摆动运行方式结束

通过信号“已经完成变速换档”(DB380x.DBX2000.3)使 NCK 获悉新的变速档已开始运行(DB380x.DBX2000.0 到 .2“实际变速档”)，摆动运行方式结束。实际变速档应与给定变速档相一致。尽管信号“摆动速度”（DB380x.DBX2002.5）仍设置，但摆动运行方式已经结束。最后编程的主轴转速(S 功能)和方向(M3,M4 或 M5)再次生效。

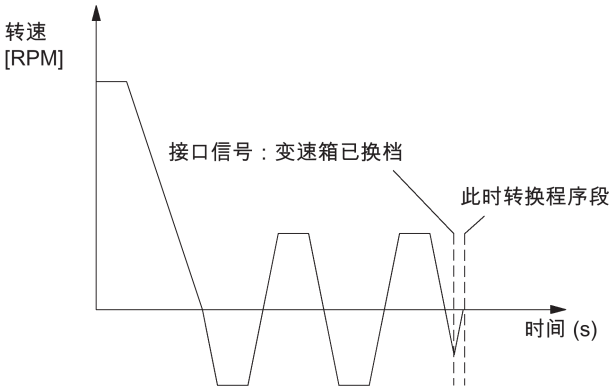
摆动运行方式结束之后主轴再次转入控制方式运行。

在各个变速档，其极值(比如，变速档的最大/最小转速等)的规定均对应着实际变速档的给定值，并在主轴停止时关闭。

程序段切换

如果在摆动方式下接通主轴(设置信号“变速换档”(DB390x.DBX2000.3))，则暂停零件程序执行。不接受新的程序段加工。在出现接口信号“已经完成变速换档”(DB380x.DBX2000.3)，从而结束摆动运行方式后，则恢复零件程序的执行。开始执行新的程序段。

摆动运行方式结束之后进行程序段转换：



特点

- 通过机床数据 MD35410 SPIND_OSCILL_ACCEL (摆动方式时的加速度)确定加速度。
- 如果接口信号“摆动速度” (DB380x.DBX2002.5) 被复位，则停止摆动运行。但摆动运行方式并**没有**结束，处于保持状态。
- 始终通过接口信号“变速档已改变”来结束变速换档。
- 用接口信号“复位”(DB3000.DBX0000.7)**不可以**终止摆动运行方式。
- 使用间接测量系统时同步丢失。越过下一个零标记时，同步恢复。

变速换档时进行复位

主轴在摆动方式并进行变速换档时，并且接口信号“已经完成变速换档” (DB380x.DBX2000.3) 还未出现的情况下，通过接口信号“复位” (DB3000.DBX0000.7) 或“NC 停止” (DB3200.DBX0007.3) 不可以使主轴停止。

在上述情况下如果仍然选择复位，则会出现报警 10640 “变速换档时不能停止”。如果在变速换档后请求复位的指令仍位于接口处，则执行这一指令，并清除报警。

说明

唯一可以停止主轴的方法：设置接口信号“清除剩余行程/主轴复位” (DB380x.DBX0002.2)。

18.2.4 主轴定位方式运行

何时选择定位方式？

在定位方式下，主轴将在定义的位置处停止。定位控制将一直有效直到被取消。在选择了可编程的功能 SPOS =..... 时主轴处于定位方式(参见章节“编程 (页 168)”)。

程序段切换

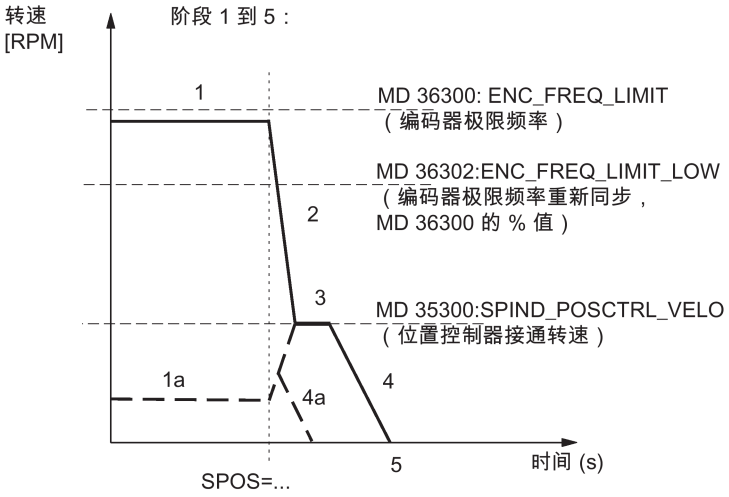
当一个程序段中所有编程的功能均已实现(比如，进给轴结束运行，PLC 已经应答所有的辅助功能)，**并且**主轴已经到位(信号“主轴准停”(DB390x.DBX0000.7))。此时可以接受下一个程序段，进行程序段转换。

前提条件

一定要有主轴实际位置编码器。

主轴从运行状态定位

不同转速下的定位：



过程

阶段 1：主轴以小于编码器极限频率的转速运行。主轴已同步。主轴位于控制方式。接着进行第 2 阶段。

阶段 1a：主轴以低于位置控制器极限速度的转速旋转。主轴已同步。主轴位于控制方式。可以进行 4a。

阶段 1b（未注明）：主轴以大于编码器极限频率的转速运行。主轴未同步，但是等速度达到 MD36302 ENC_FREQ_LIMIT_LOW(MD36300 的百分比值)定义的值时，主轴同步。接着进行第 2 阶段。

阶段 2：随着 SPOS 指令的生效，主轴按照 MD35200 GEAR_STEP_SPEEDCTRL_ACCEL 中定义的加速度开始制动直至达到位置控制极限速度。

阶段 3：到达 MD35300 SPIND_POSCTRL_VELO 中设定的位置控制极限速度后：

- 接通位置控制。
- 计算到目标位置的剩余行程。(最好在阶段 1a 时计算)
- 加速度转换到 MD35210 GEAR_STEP_POSCTRL_ACCEL(位置控制方式加速度)中设定的加速度。（始终在位置控制转速下有效）

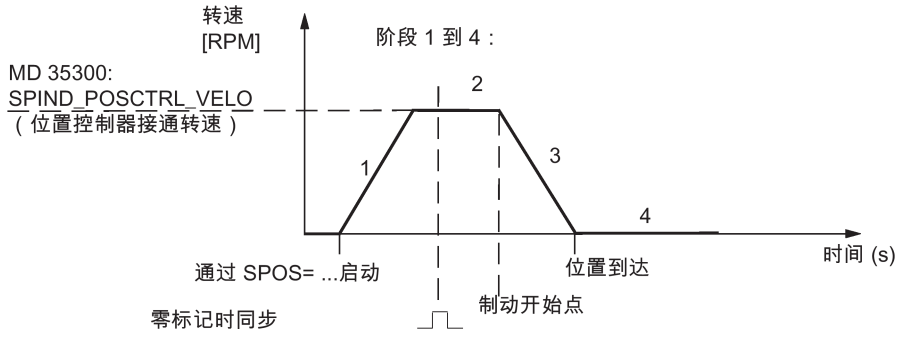
阶段 4：主轴从计算得到的“制动点”开始按照 MD35210 GEAR_STEP_POSCTRL_ACCEL 中设定的加速度制动，直至到达目标位置。

阶段 5：位置控制有效，主轴保持在编程的位置。一旦主轴实际位置与编程位置(给定的主轴位置)之间的距离小于精准停（DB390x.DBX0000.7）和粗准停（DB390x.DBX0000.6）时大小分别在 MD36010 STOP_LIMIT_FINE 和 MD36000 STOP_LIMIT_COARSE 中确定，设置接口信号“精准停”和“粗准停”。

主轴从停止状态进行定位，主轴未同步

数控系统启动后，主轴没有同步。主轴的第一个动作是定位(SPOS=...)。

主轴停止且未同步时定位：



过程

阶段 1：如果编程了 SPOS，主轴按照 MD35210 GEAR_STEP_ POSCTRL_ACCEL(速度控制方式加速度)中所设定的加速度进行加速运行，直至到达 MD35300 SPIND_POSCTRL_VELO(位置控制接通速度)中的速度。

除非在 SPOS 程序(CAN,ACP,IC)中没有预设值，旋转方向由 MD35350 SPIND_POSITIONING_ DIR(从停止状态定位时的旋转方向)设定。主轴与主轴位置实际值编码器的下一个零标记同步。

阶段 2：主轴同步后接通位置控制方式。主轴按照一速度(最大为 MD35300 SPIND_POSCTRL_VELO 设定的速度)运行，一直运行到开始进行减速的起始点为止。控制器通过计算可以得到一起始点，从这点开始按照所确定的加速度可以准确地到达所编程的主轴位置。

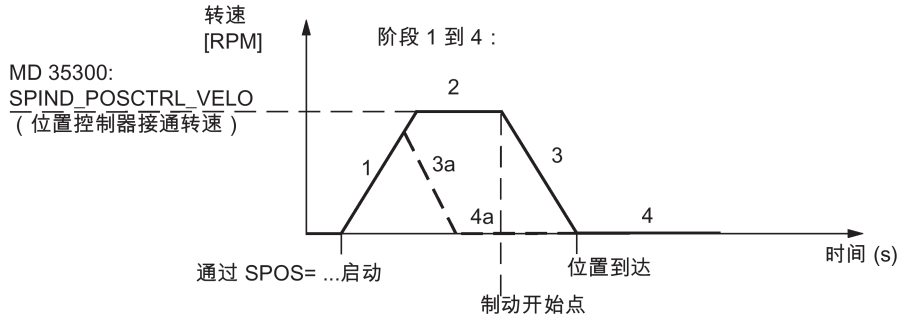
阶段 3：主轴按照 MD35210 GEAR_STEP_POSCTRL_ACCEL(位置控制方式加速度)中设定的加速度，从制动点开始制动到停止。

阶段 4：主轴到达位置并停止。位置控制方式有效，主轴停止在所编程的位置。一旦主轴实际位置与编程位置(给定的主轴位置)之间的距离小于精准停（DB390x.DBX0000.7）和粗准停（DB390x.DBX0000.6）时(MD36010 STOP_LIMIT_FINE 和 MD36000 STOP_LIMIT_COARSE)，设置接口信号“精准停”和“粗准停”。

主轴从停止状态进行定位，主轴已同步

至少一个主轴已使用 M3 或 M4 开始旋转，然后用 M5 停止。

同步主轴停止时定位：



过程

主轴以时间最优的方式运行到所编程的目标位置。根据不同的边界条件可以按照 1 - 2 - 3 - 4 阶段，或者按照 1 - 3a - 4a 的阶段运行。

阶段 1：通过 SPOS 编程将主轴转换到位置控制方式。MD35210 GEAR_STEP_POSCTRL_ACCEL(位置控制模式下的加速度)值有效。主轴的旋转方向由所产生的剩余行程（通过 SPOS 的路径设置类型）确定。

速度不超过 MD35300 SPIND_POSCTRL_VELO(位置控制极限值)的速度值。可以计算出到达目标位置的剩余行程。如果在此阶段可以立即到达目标点，继续执行 3a 和 4a,无需执行阶段 2。

阶段 2：为了到达目标位置，主轴加速运行，但加速后的最大速度为 MD35300 SPIND_POSCTRL_VELO（位置控制转速）中设定的速度。主轴运行到开始进行减速的起始点，从这点开始按照 MD35210 GEAR_STEP_POSCTRL_ACCEL 所确定的加速度可以准确地到达所编程的主轴位置(SPOS=...)。

阶段 3 和阶段 4：“制动”和“位置到达”顺序和未同步主轴的顺序一样。

主轴复位

主轴定位过程可以通过接口信号“清除剩余行程/主轴复位”（DB380x.DBX0002.2）终止。但主轴仍然处于定位方式。

说明

- 主轴修调开关继续有效。
- 定位过程(SPOS)可以用复位或“NC - STOP”终止。

18.3 同步

为何进行同步？

为了使系统在通电以后能够精确地识别出 0 度位置，系统必须与主轴位置测量系统进行同步。只有同步主轴可以进行螺纹切削和定位。

就进给轴而言，此过程称为“回参考点”(参见章节“回参考点（R1） (页 152)”)。

位置测量系统的安装部位

- 直接安装在电机上，且主轴装有 BERO(零脉冲编码器)。
- 直接安装在主轴上。
- 安装在位于测量变速箱上面的主轴且主轴装有 BERO 开关。

说明

只有在通过 PPU 接口 X54 连接了模拟量主轴驱动（此时没有连接 V70 主轴驱动）且通过 PPU 接口 X60 连接了电机编码器时，才可使用 BERO 接近开关。

同步过程

系统上电以后，主轴按如下过程进行同步：

- 主轴以一速度(S 功能)和方向(M3 或 M4)启动，并与位置测量系统或 BERO 信号的下一个零脉冲同步。0 度位置偏移了 MD34080 REFP_MOVE_DIST + MD34090 REFP_MOVE_DIST_CORR - MD34100 REFP_SET_POS。

  说明

  对应 0 度位置的偏移，只使用 MD34080。使用 MD34060 REFP_MAX_MARKER_DIST 的监控应设定成主轴旋转两圈 (720 度)。

- 在不同方式下编程 SPOS = ...（参见章节“主轴定位方式运行 (页 161)”）。
- 在“JOG”方式下，使用方向键启动速度控制方式下的主轴并与位置测量系统的下一个零标记或 BERO 信号同步。

值确认

主轴同步时，通过 MD34100 REFP_SET_POS[0](缺省值 = 0)确认参考点(如果有)的相应的参考点值和偏移量。这些偏移量（机床数据）独立于连接的测量系统有效且在章节“回参考点（R1） (页 152)”中说明。

超出最大编码器频率

主轴在控制方式下运行时，若其转速(编程了一个较大的 S 值)超出编码器极限频率 MD36300 ENC_FREQ_LIMIT(此时不允许超出编码器的最大转速)，则同步丢失。主轴继续旋转，但功能降低。

如果达到的转速小于 MD36302 ENC_FREQ_LIMIT_LOW 中的编码器频率（MD36300 的百分比值）,则主轴自动从下一个零标记处进行同步。可以通过以下方法实现：编程较小的 S 值，修改主轴倍率开关等。

重新进行同步

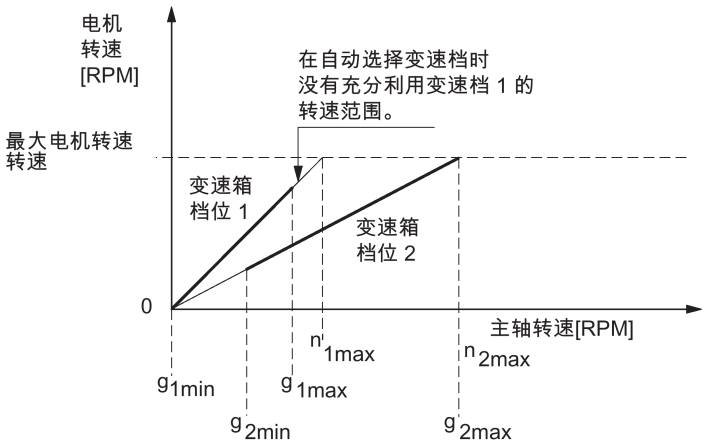
在以下情况中必须重新同步主轴的位置测量系统：位置编码器安装于电机上，在主轴上安装一个接近开关 BERO（用于同步信号的距离传感器），并进行变速换档。当主轴换档以后在新的变速档上旋转时，在内部触发同步信号。

18.4 变速换档

变速档个数

一个主轴可以设置五个变速档。如果主轴直接与电机相连(1:1)，或者主轴到电机的传动比已经固定不变，则机床数据 MD35010 GEAR_STEP_CHANGE_ENABLE(可以进行变速换档)必须置零。

带有变速档选择的变速换档：



通过 MD 预设： n1max ... 第 1 变速档的最大主轴转速
g1min ... 第 1 变速档的最小主轴转速
用于自动变速换档选择
g1max ... 第 1 变速档的最大主轴转速
用于自动变速换档选择
n2max ... 第 2 变速档的最大主轴转速
g2min ... 第 2 变速档的最小主轴转速
用于自动变速换档选择
g2max ... 第 2 变速档的最大主轴转速
用于自动变速换档选择

变速档预选

变速档的预置可以由以下方法进行：

- 通过零件程序(M41 到 M45)
- 通过编程的主轴速度自动进行(M40)

用 M40 自动进行变速换档时主轴必须位于主轴控制方式下并有 S 指令。否则不能执行变速换档，并发出报警 22000“不能换档”。

M41 到 M45

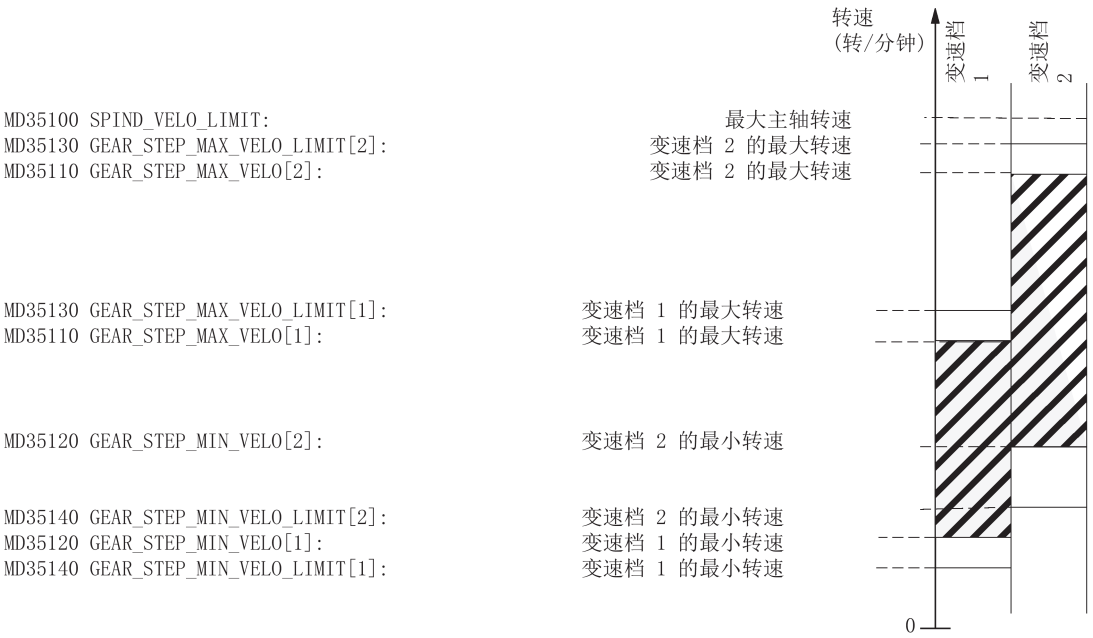
可以在零件程序中用 M41 到 M45 预先确定变速档。如果从当前的变速档转换到 M41 到 M45 所确定的变速档，则设置接口信号“变速换档”(DB390x.DBX2000.3)和信号“给定变速档 A 到 C”(DB390x.DBX2000.0 到.2)。这样，编程的主轴转速(S 功能)就与给定的变速档相关。如果所编程的主轴转速大于变速档的最大转速，则主轴转速只能是变速档的最大转速，并设置接口信号“限制给定转速”(V390x2001.1)。如果所编程的转速小于该变速档的最小转速，则将转速提高到该最小转速。在此对信号“提高额定转速” (DB390x.DBX2001.2)进行设置。

M40

通过零件程序中的 M40 指令，数控系统可以自动确定变速档。此时数控系统确定编程的主轴转速(S 功能)可能位于哪一个变速档上。如果所确定的变速档不是当前的变速档，也就是说，当前的变速档要进行换档，则设置接口信号“变速换档”(DB390x.DBX2000.3)和“设定变速档 A 到 C”(DB390x.DBX2000.0 到.2)。

数控系统在自动选择变速档时按照如下过程进行：编程的主轴转速首先与当前变速档的最小值和最大值进行比较。如果比较结果为正，则不给出新的变速档。如果比较结果为负，则从变速档 1 开始逐节进行比较，直到结果为正。若在变速档 5 时比较结果仍不为正，则不进行变速换档。主轴转速限制为当前变速档的最大转速，或者提高到当前变速档的最小转速，并设置接口信号“限制给定转速”(DB390x.DBX2001.1)或“提高给定转 速”(DB390x.DBX2001.2)。

自动换档时转速范围说明(M40)：



变速换档

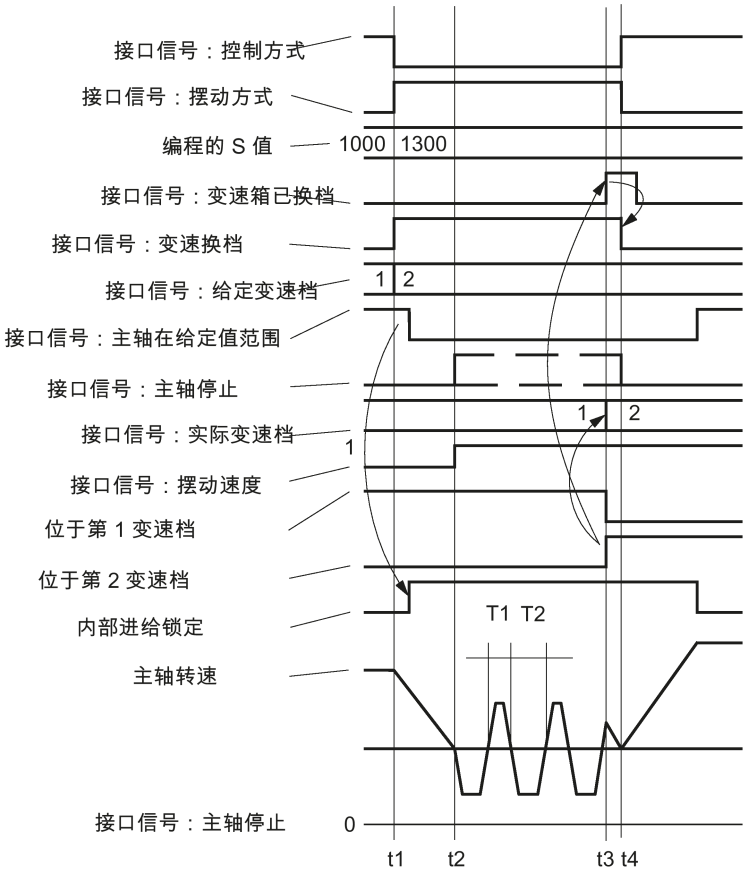
只有在主轴停止时才能切换新的变速档。
如果要求变速换档，则在数控系统内部停止主轴。如果通过 M40 预设新的变速档，或者通过 M41 到 M45 预置主轴转数，则设置接口信号“设定变速档 A 到 C”(DB390x.DBX2000.0 到.2)和信号“变速换档”(DB390x.DBX2000.4)。根据设置接口信号“摆动速度”(DB380x.DBX2002.5)所发生的时间，主轴按照摆动方式运行的加速度，或者按照速度控制方式/位置控制方式加速度制动到停止。

通过 M40 和 S 指令，或者通过 M41 到 M45 使变速换档**之后**，不执行零件程序中的下一个程序段(此时就如同设置了接口信号“禁止读入”(DB3200.DBX0006.1)一样)。

当主轴停止时，（接口信号“进给轴/主轴停止”（DB390x.DBX0001.4））就通过接口信号“摆动速度”（DB380x.DBX2002.5）接通摆动方式运行（参见章节“主轴摆动方式运行 (页 160)”）。在换上新 的变速档后，由 PLC 用户设置接口信号“实际变速档”(DB380x.DBX2000.0 到.2)和接口信号“已经完成变速换档”(DB380x.DBX2000.3)。变速档转换结束（主轴运行方式选定为“摆动运行”），并且转换到新的实际变速档参数程序段上。主轴按新的变速档高速旋转到最后编程的主轴转速（只要 M3 或 M4 有效）。当 PLC 用户必须对信号“变速换档” (DB390x.DBX2000.3)进行复位时，信号“已经完成变速换档” (DB380x.DBX2000.3)通过 NCK 进行复位。零件程序中的下一个程序段可以开始运行。

下图展示了典型的变速换档的时序过程。

主轴停止，变速换档：



t1 通过编程 S1300，NCK 识别新的变速档
（第 2 变速档），设置信号：变速换档并锁定下一
零件程序段的处理。

t2 主轴停止，开始摆动（摆动由 NCK 实现）。
接口信号：摆动转速最迟在 t2 时设置。

t3 确定新的变速档。PLC 用户将新的（实际）
变速档传送到 NCK 并设置接口信号：已经完成变速换档。

t4 然后，NCK 接收接口信号：返回变速档位，结束
摆动方式，释放下一个用于加工的零件程序段并使
主轴加速到新的 S 值（S1300）。

参数组

五个变速档中每个变速档均有一个参数组。相应的参数组由接口信号“实际变速档 A” 到 “...C”(DB380x.DBX2000.0 到.2)激活。它们按如下规则分配：

索引 n	PLC 接口 编码 CBA	数据程序段的数据	目录
0	-	进给轴运行的数据	Kv 系数、监控功能、转速、加速度等
1	000 001	变速档 1 的数据	
2	010	变速档 2 的数据	
3	011	变速档 3 的数据	
4	100	变速档 4 的数据	
5	101	变速档 5 的数据	

在章节“机床数据 (页 171)”中介绍了以上参数组所包含的机床数据。对于每个变速档，它的参数组索引 n(n=1 -> 主轴的第 1 变速档等)中增加了以下机床数据：

- MD35110 GEAR_STEP_MAX_VELO[n]
- MD35120 GEAR_STEP_MIN_VELO[n]
- MD35130 GEAR_STEP_MAX_VELO_LIMIT[n]
- MD35140 GEAR_STEP_MIN_VELO_LIMIT[n]
- MD35200 GEAR_STEP_SPEEDCTRL_ACCEL[n]
- MD35210 GEAR_STEP_POSCTRL_ACCEL[n]
- MD35310 SPIND_POSIT_DELAY_TIME[n]

18.5 编程

功能

主轴可以设置以下功能：

- G95 旋转进给
- G96 S... LIMS=... 恒定切削速度，单位米/分钟，转速上限
- 取消 G97 G96 ，冻结最后主轴转速
- G33，G331，G332 螺纹切削，攻丝
- G4 S... 暂停时间主轴旋转。

M3	顺时针旋转主轴
M4	逆时针旋转主轴
M5	主轴停止，无定向
S...	主轴转速，单位：转/分钟。比如：S300
SPOS=...	主轴定位，例如： SPOS=270 -> 定位到 270 度的位置。 主轴定位以后才更换程序段。
SPOS=DC(Pos)	在移动过程中定位时保持方向不变，并到达该位置。 如果是从静止状 态开始定位，则以最短距离进行定位。
SPOS=ACN(Pos)	始终在负方向到达预定位置。 必要时，在定位之前颠倒移动方向。
SPOS=ACP(Pos)	始终在正方向到达预定位置。 必要时，在定位之前颠倒移动方向。
SPOS=IC(Pos)	给出进给行程。 移动方向取决于路径的符号。 如果主轴已在运行中， 可以偏转移动方向，从而能够按照编程的方向移动。
M40	主轴自动选择变速档
M41 到 M45	为主轴选择变速档 1 到 5
SPCON	开启位置控制
SPCOF	关闭位置控制
M70	开启位置控制
LIMS=...	G96 时的可编程的最大主轴转速

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

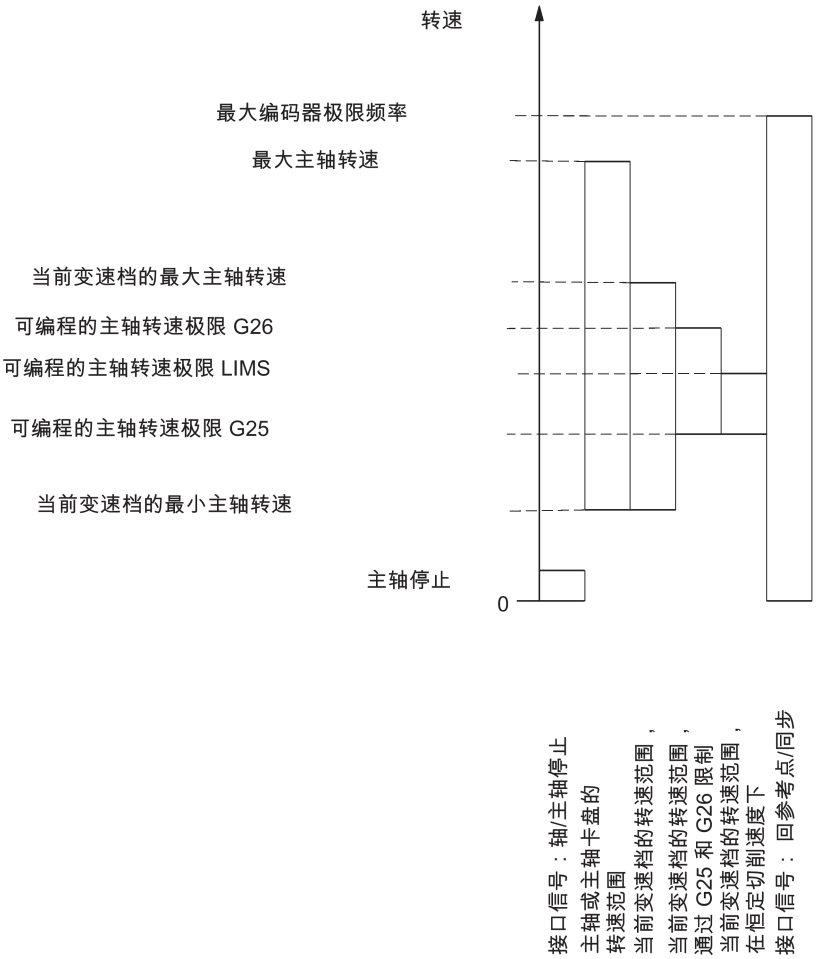
18.6 主轴监控

18.6.1 主轴监控

速度范围

通过主轴监控和当前有效的功能(G94,G95,G96,G33,G331，G332 等)确定允许的主轴转速范围。

主轴监控范围/转速范围：



18.6.2 轴/主轴停止

只有当进给轴/主轴停止时，如当主轴实际转速低于 MD36060 STANDSTILL_VELO_TOL 中规定的数值时，则出现接口信号“进给轴/主轴停 止”(V390x0001.4)。 此时采用用户程序可以执行机床的一些功能，比如：换刀，打开机床门，进给使能等等。

在三种主轴方式下监控均有效。

18.6.3 主轴在给定值范围

主轴监控“主轴在给定范围”功能将监控是否到达编程的主轴转速，主轴是否停止(信号“进给轴/主轴停止”),或者主轴是否还位于加速阶段。

在主轴控制方式下比较实际值和给定值(编程速度主轴修调，考虑有效的极限值)。实际转速和给定转速的偏差如果大于主轴速度公差(由 MD35150 SPIND_DES_VELO_TOL 设定)，则：

- 接口信号“主轴在给定范围”(DB390x.DBX2001.5)置为零。
- 不释放下一加工程序段，当
  设置了 MD35500 SPIND_ON_SPEED_AT_IPO_START。

18.6.4 最大主轴转速

最大主轴转速

主轴转速不得超出主轴监控中所定义的“最大主轴转速”。

在机床数据 MD35100 SPIND_VELO_LIMIT 中设定最大主轴转速。

系统把主轴的转速限制在此速度之内。 如果主轴转速实际值大于最大主轴转速与主轴速 度公差(MD35150 SPIND_DES_VELO_TOL)之和，则驱动出现差错，并设置接口信号“超 出转速极限”(V390x2002.0)。 此外还发出报警 22100，主轴被制动。

18.6.5 变速档的最大/最小转速

最大转速

在机床数据 MD35130 GEAR_STEP_MAX_VELO_LIMIT 中设定变速档转速的最大值。在设定变速档中绝不能超出该给定转速。 在限制编程的主轴转速时设置接口信号“限制给定转速”(DB390x.DBX2001.1)。

最小转速

在机床数据 MD35140 GEAR_STEP_MIN_VELO_LIMIT 中设定变速档转速的最小值。在编程很小的 S 值时不得低于此给定转速。 此时设置接口信号“提高给定转速”(DB390x.DBX2001.2)。

变速档的最小转速仅当主轴在控制方式下起作用，并且只有在下列情况下才可能低于此速度：

- 主轴修调 0 %
- M5
- S0
- 信号“主轴停止”
- 取消信号“控制器使能”
- 接口信号 “复位”
- 接口信号 “主轴复位”
- 接口信号 “摆动速度”
- 接口信号“NC - STOP 进给轴/主轴”
- 接口信号“进给轴/主轴锁定”

18.6.6 最大编码器极限频率

说明

主轴位置实际值编码器的最大极限频率受到控制器的监控(可能会超出)。机床制造商在设计主轴电机部件、减速箱、测量传动装置，以及在选择编码器和设定机床数据时要保 证不会超出主轴位置实际值编码器的最大转速(机械极限转速)。

超出最大编码器极限频率

如果在主轴控制方式运行时速度(编程了一个很大的 S 值)超出了最大编码器极限频率(此 时不得超出编码器的最大机械极限转速)，则同步丢失。但主轴仍继续旋转。

在零件程序中如果编程了以下的某一功能：螺纹切削(G33)、旋转进给(G95)、恒定切削速 度(G96,G97)，则主轴转速会自动降低，直至测量系统恢复工作。

最大编码器极限频率在主轴的“定位模式”和位置控制螺纹加工(G331，G332)时未被超过。

如果超出编码器极限频率，用于测量系统的接口信号“回参考点/同步”(DB390x.DBX0000.4)被复位，且设置了“编码器极限频率 1 超出”(DB390x.DBX0000.2)的接口信号。

如果在超出了最大编码器极限频率之后，速度变化并又低于 MD36302 ENC_FREQ_LIMIT_LOW(MD36300 ENC_FREQ_LIMIT 的百分比值)中定义的极限频率值，则主轴会自动地与下一个零标记或下一个接近开关 Bero 信号同步。

18.6.7 目标定位监控

功能

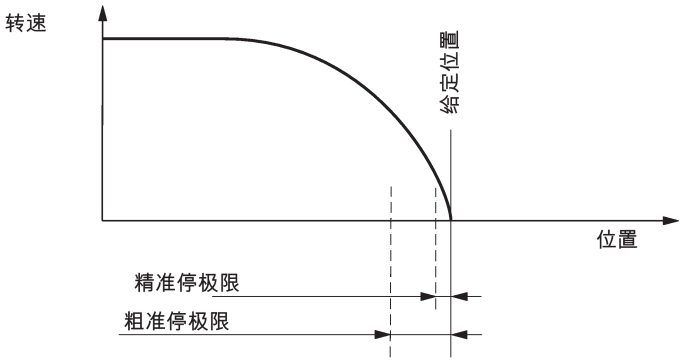
主轴在定位方式进行定位时会监控主轴所定位的实际位置距离编程的给定位置(目标点)有多远。

可以在下列机床数据中预设连个极限值，作为增量位移（从给定位置出发）：

- MD36000 STOP_LIMIT_COARSE （粗准停极限）
- MD36010 STOP_LIMIT_FINE（精准停极限）

主轴定位精度与这两个极限值无关，而是由所连接的主轴测量编码器、间隙及变速传动比决定。

在定位时主轴的准停区域：



接口信号：准停到位

达到 36000 和 MD 36010 中设定的极限时，则将相应的接口信号“粗准停到位”(V390x 0000.6)和信号“精准停到位”(V390x 0000.7)送给 PLC。

SPOS 后程序段转换

通过 SPOS 进行主轴定位后程序段的转换取决于通过接口信号“精准停到位”进行的目标点监控。程序段转换时，要求上一个程序段中的所有功能均已执行完毕(比如轴进给结束， PLC 应答了所有的辅助功能)。

18.7 模拟主轴

功能

数控系统中的模拟主轴设计用于机床运行。通过介于 +10 V 和 -10 V 之间的额定模拟电压和位于 X21-8 和 X21-9 端子的两个信号来控制主轴。该电压在数控系统中拥有对应的输出。

模拟主轴支持增量编码器（TTL 编码器），这种编码器可以直接连接到数控系统。您仅能设定模拟主轴上编码器的参数。当设定步进电机上编码器的参数时，会给出报警 26006。

通过 MD30130 CTRLOUT_TYPE 和 MD30240 ENC_TYPE 可以在模拟主轴和实际主轴间切换额定值输出。对于没有编码器的模拟主轴应将 MD30240 ENC_TYPE[n] 设置为零。

18.8 数据表

18.8.1 机床数据

序号	名称	名称
通道专用		
20090	SPIND_DEF_MASTER_SPIND	主主轴
轴专用		
30134	IS_UNIPOLAR_OUTPUT[0]	设定值输出为单极
30300	IS_ROT_AX	回转轴
30310	ROT_IS_MODULO	模态转换

序号	名称	名称
30320	DISPLAY_IS_MODULO	位置显示
31050 *	DRIVE_AX_RATIO_DENOM[n]	负载变速箱分母
31060 *	DRIVE_AX_RATIO_NUMERA[n]	负载变速箱分子
32200 *	POSCTRL_GAIN[n]	Kv 系数
32810 *	EQUIV_SPEEDCTRL_TIME[n]	速度控制回路的前馈控制的相应时间常量
34040	REFP_VELO_SEARCH_MARKER	回参考点停止速度
34060	REFP_MAX_MARKER_DIST	零标记距离监控
34080	REFP_MOVE_DIST	参考点距离/带距离编码系统的目标位置
34090	REFP_MOVE_DIST_CORR	参考点偏移/绝值偏移行程编码
34100	REFP_SET_POS	参考点值
34200	ENC_REFP_MODE	回参考点模式
35000	SPIND_ASSIGN_TO_MACHAX	主轴定义
35010	GEAR_STEP_CHANGE_ENABLE	可以进行变速换档
35040	SPIND_ACTIVE_AFTER_RESET	主轴复位有效
35100	SPIND_VELO_LIMIT	最大主轴转速
35110 *	GEAR_STEP_MAX_VELO[n]	变速换档的最大转速
35120 *	GEAR_STEP_MIN_VELO[n]	变速档的最小转速
35130 *	GEAR_STEP_MAX_VELO_LIMIT[n]	变速档的最大转速
35140 *	GEAR_STEP_MIN_VELO_LIMIT[n]	变速档的最小转速
35150	SPIND_DES_VELO_TOL	主轴转速公差
35200 *	GEAR_STEP_SPEEDCTRL_ACCEL[n]	转速控制方式下的加速度
35210 *	GEAR_STEP_POSCTRL_ACCEL[n]	位置调节方式下的加速度
35300	SPIND_POSCTRL_VELO	位置控制极限速度
35310	SPIND_POSIT_DELAY_TIME[n]	定位延迟时间
35350	SPIND_POSITIONING_DIR	未同步主轴定位时的旋转方向
35400	SPIND_OSCILL_DES_VELO	摆动速度
35410	SPIND_OSCILL_ACCEL	摆动时加速度
35430	SPIND_OSCILL_START_DIR	摆动时启动方向
35440	SPIND_OSCILL_TIME_CW	M3 方向的摆动时间
35450	SPIND_OSCILL_TIME_CCW	M4 方向的摆动时间
35500	SPIND_ON_SPEED_AT_IPO_START	主轴在设定范围内进给使能
35510	SPIND_STOPPED_AT_IPO_START	主轴在停止时进给使能
36060	STANDSTILL_VELO_TOL	极限速度“进给轴/主轴停止”
36200 *	AX_VELO_LIMIT[n]	速度监控极限值
36300	ENC_FREQ_LIMIT	编码器极限频率
36302	ENC_FREQ_LIMIT_LOW	编码器极限频率重新同步
36720	DRIFT_VALUE	漂移原始值

标有*的机床数据包含在每个变速档的参数组中。

18.8.2 设定数据

序号	名称	名称
概述		
41200	JOG_SPIND_SET_VELO	主轴的 JOG 速度
主轴专用		
43230	SPIND_MAX_VELO_LIMS	编程的 G96 时主轴转速限制

18.8.3 接口信号

序号	位	名称
轴专用		
DB30x.DBD0000	-	用于主轴的 M 功能(DINT),进给轴专用
DB30x.DBD0004	-	用于主轴的 S 功能(REAL),进给轴专用
DB380x.DBB0000	-	进给率修调
DB380x.DBX0001	.7	修调有效
DB380x.DBX0001	.5	位置编码器 1
DB380x.DBX0001	.3	进给轴/主轴锁定
DB380x.DBX0002	.2	主轴复位/删除剩余行程
DB380x.DBX0002	.1	调节器使能
DB380x.DBX2000	.3	变速箱已换档
DB380x.DBX2000	.0 至 .2	实际变速档 A 至 C
DB380x.DBX2001	.4	定位 1 时重新同步(主轴)
DB380x.DBX2001	.6	M3/M4 反向
DB380x.DBX2002	.7	设定逆时针旋转方向
DB380x.DBX2002	.6	设定顺时针旋转方向
DB380x.DBX2002	.5	摆动速度
DB380x.DBX2002	.4	通过 PLC 摆动
DB380x.DBB2003	-	主轴修调
DB390x.DBX0000	.7	采用精准停到达位置
DB390x.DBX0000	.6	采用粗准停到达位置
DB390x.DBX0000	.4	回参考点/同步 1
DB390x.DBX0000	.2	编码器极限频率超越 1
DB390x.DBX0000	.0	主轴/无进给轴
DB390x.DBX0001	.7	当前控制器有效
DB390x.DBX0001	.6	速度控制器有效
DB390x.DBX0001	.5	位置控制器有效
DB390x.DBX0001	.4	进给轴/主轴停止($n < n_{min}$)
DB390x.DBX2000	.3	变速换档
DB390x.DBX2000	.0 至 .2	实际变速档 A 至 C
DB390x.DBX2001	.7	实际旋转方向顺时针
DB390x.DBX2001	.5	主轴在给定值范围
DB390x.DBX2001	.2	增加设定速度
DB390x.DBX2001	.1	设定速度极限
DB390x.DBX2001	.0	超出速度限值
DB390x.DBX2002	.7	有效主轴方式“控制方式”
DB390x.DBX2002	.6	有效主轴方式“摆动方式”
DB390x.DBX2002	.5	有效主轴方式“定位方式”
DB390x.DBX2002	.3	刚性攻丝有效
DB390x.DBX2002	.0	恒定切削率有效(G96)

19 进给率（V1）

19.1 轨迹进给率 F

19.1.1 轨迹进给率 F

功能

进给率 F 是刀具沿着编程的工件轮廓进行加工时的**轨迹速度**。单个轴的速度是刀具轨迹速度在坐标轴上的分量。

进给率 F 在 G1,G2,G3,CIP，CT 插补方式中生效，并且一直有效，直到编程了一个新的 F 值。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

进给率 F 的单位：G94,G95

进给率 F 字的单位由 G 功能确定:

- G94 F 进给率单位为毫米/分钟或英寸/分钟
- G95 F 进给率单位为毫米/转或英寸/转
  (仅在主轴旋转时有意义!)

英制单位系统适用于 G700 或系统的“英制”设定，MD10240 SCALING_SYSTEM_IS_METRIC=0。

G96 和 G97 时进给率 F 的单位

对于**车床**，在 G94 和 G95 功能组中又增加了 G96 和 G97 功能，它们用于**恒定切削速度**(打开/关闭)。这些功能对 S 指令也有影响。

G96 功能生效后，主轴转速随着当前加工的工件直径(端面轴)变化而变化，从而使刀具切削点处编程的切削速度 S 始终保持恒定(主轴转速 x 直径 = 常数)。

从 G96 程序段开始，地址 S 下的转速值作为切削速度处理。G96 为模态有效,直到被 G 功能组中一个其它 G 指令(G94、G95、 G97)替代为止。

在此，进给率 F 的单位始终为毫米/转或英寸/转(如同在 G95 时一样)。

最大轨迹速度

最大轨迹速度取决于相关轴的最大速度(MD32000 MAX_AX_VELO)以及它们在总的进给路径中的进给量。但不可以超出机床数据中设定的最大轴速度。

加工圆弧时的进给率修调 CFC

在用铣刀加工圆弧轮廓时，如果刀尖半径补偿(G41/G42)已经生效，则必须调整刀具中心的进给率，以确保所编程的 F 值在圆弧轮廓处生效。在进给率修调 **CFC** 生效以后，控制器会自动识别内圆加工和外圆加工。

用 **CFTCP** 可以关闭进给率修调。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

接口信号

旋转进给有效时设置接口信号“旋转进给有效”(DB3300.DBX0001.2)。

如果 G96/G332 功能有效，设置主轴接口信号“恒定切削率有效”(DB390x.DBX2002.0)。

报警

- 如果在 G1,G2,G3...的零件程序中没有编程 F 值，则给出报警 10860。进给轴不能运行。注意：SD42110 DEFAULT_FEED!
- 若编程 F0，则给出报警 14800。
- 在 G95 时如果主轴停止，则进给轴也不运行。此时，不发出报警。

说明

- 如果“空运行进给”功能有效且程序已执行，则 G1、G2、G3、CIP 和 CT 下编程的进给率由 SD42100 DRY_RUN_FEED 中的定义值所替代，参见章节“按照空运行进给(DRY)处理程序 (页 98)”。
- 进给轴在“JOG”方式下的进给速度由机床数据/设定数据定义。

19.1.2 G33, G34, G35 (螺纹加工)时的进给率

螺纹攻丝的类型

G33 - 带恒定螺距的螺纹

G34 - 带有（线性）递增螺距的螺纹

G35 - 带有（线性）递减螺距的螺纹

进给轴速度

在用 G33、G34 或者 G35 加工螺纹时，进给轴的速度取决于所调节的主轴转速和所编程的螺距。但其大小也不能超出机床数据 MD32000 MAX_AX_VELO 中所设定的最大速度值。

此时进给率 F 不起作用。尽管如此仍然保留此功能。

加工螺纹时进给轴的速度由所设定的主轴速度 (S) 和所编程的螺距大小 (K) 决定，比如，在加工圆柱螺纹时：

Fz[mm/min]= 转速 S[rev/min]*螺距 K[mm/rev]

说明

在使用 G34 和 G35 时在 F 地址中以毫米/转 ^{2} 对螺距变化进行编程。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

NC 停止，单程序段

NC 停止和单段方式只有在一个螺纹加工过程结束后才生效。

说明

- 在加工螺纹时主轴速度修调开关应保持位置不变。
- 在有 G33、G34、G35 的程序段中，进给修调开关不起作用。

可编程的输入与输出行程：DITS, DITE

在螺纹有必要时需要另外运行输入与输出行程。在这个范围中可以进行轴的快速运行和制动（在锥形螺纹时为两个轴）。该路程取决于螺距，主轴转速以及轴的动力（配置）。

如果可供输入与输出使用的行程受到限制，则根据情况降低主轴转速，直到该行程够用。为了在这类情况下达到较好的切削值和较短的加工时间、或者为了简化处理该问题，可以在程序中额外给定输入与输出行程。

未给定时设置数据 (SD) 的值生效。程序中的数据被写入到 SD42010 THREAD_RAMP_DISP[0] ... [1] 当中。

如果该行程在使用轴加速度的运行过程中不够用的话，则轴会发生加速度过载。对于螺纹导入量会在其后发出报警 22280 “编程的导入行程过短”。报警只是一种信息，它对于零件程序的执行没有影响。

输出行程在螺纹攻丝结束时作为精磨间距起作用。这样可以完成轴运行中的无碰撞变更。

编程

DITS= ...:螺纹的输入行程

DITE= ...:螺纹的输出行程

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

SD42010

在 DITS 和 DITE 下只编程位移，不编程位置。

使用零件程序指令来设定数据 SD42010 THREAD_RAMP_DISP[0], ...[1]，它们在螺纹攻丝中用来确定下面的加速性能（[0]- 输入，[1]- 输出）：

- SD42010 = < 0 到 -1:
  使用定义的加速度启动/制动进给轴。冲击将根据当前编程的 BRISK/SOFT 起作用。
- SD42010 = 0:
  阶梯曲线过后，启动/制动螺纹切削时的进给轴。
- SD42010 = > 0:
  定义了螺纹快速行程/制动行程。为了避免工艺报警 22280，在输入或输出行程很小时要注意轴的加速限制。

说明

DITE 在螺纹结束处作为精加工距离，从而使轴的运行平稳改变。

G34，G35 时的螺纹间距变化率 F

如果已知一个螺纹的起始螺距和最终螺距，那么就可以根据下面的等式计算出编程的螺距变化率：

$$F = \frac{|K_e^2 - K_a^2|}{2 * L_G} \ [mm/rev^2]$$

这些标识符具有下列含义：

K_e 轴目标点坐标的螺距 [mm/rev]

K_a 螺纹起始螺距（在 I, K 下编程）[mm/rev]

L_G 螺纹长度，单位 [mm]

19.1.3 G63 (带补偿夹具进行攻丝)时的进给率

进给率 F

在 G63 中必须编程一个进给率 F。 进给率 F 必须与主轴转速 S(编程值或设定值)和钻头的螺距符合以下关系：

进给率 F[毫米/分钟]=转速 S[转/分钟]x 螺距[毫米/转]

在此，补偿夹具将在一定程度补偿钻削轴所产生的位移差。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

19.1.4 G331, G332 (不带补偿夹具进行攻丝)时的进给率

进给轴速度

在使用 G331/G332 进行攻丝时，进给轴速度由编程的主轴转速 S 和编程的螺距大小决定。 但其大小也不能超出机床数据 MD32000 MAX_AX_VELO 中所设定的最大速度值。

此时进给率 F 不起作用。 尽管如此仍然保留此功能。

接口信号

G331/G332 功能有效时，设置主轴接口信号“不带补偿夹具进行攻丝有效” (DB390x.DBX2002.3)。

说明

只有主轴和进给轴的动态性能能够相互匹配，才可以进行不带补偿夹具的攻丝。 使用 G331/G332 时，进给轴的参数组 n(0...5) 自动生效。同时也对主轴当前的变速档有效(M40、M41 到 M45 - 参见章节“主轴（S1） (页 158)”)。

通常，响应较慢主轴也应和相应的进给轴匹配。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

19.1.5 倒角/倒圆时的进给率： FRC, FRCM

倒角/倒圆

在轮廓角中可以加入倒角（CHF 或 CHR）或倒圆（RND）。 如果希望用同样的方法对若干轮廓拐角连续进行倒圆，那么用“模态倒圆”（RNDM）命令达到。

用于倒角/倒圆的进给率可以用 FRC=... (非模态方式) 或者 FRCM= ... (模态方式)进行编程。如果没有编程 FRC/FRCM，那么一般进给率 F 生效。

编程

FRC=...	用于倒角/倒圆的非模态进给率，	
	值>0:	=在 G94 时进给率以毫米/分为单位，在 G95 时以毫米/转为单位。
FRCM=...	用于倒角/倒圆的模态进给率	
	值>0:	进给率以毫米/分（G94）或者毫米/转（G95）为单位 打开用于倒角/倒圆的模态进给率，
	值 = 0:	关闭用于倒角/倒圆的模态进给率， 在倒角/倒圆时进给率 F 起作用

说明

- 如果以 G0 进行倒角，则 F、FRC、FRCM 无效。 如果倒角/倒圆时进给率 F 生效，则在正常情况下进给率为离开轮廓角程序段中的值。 可以通过机床数据 MD20201 CHFRND_MODE_MASK 来进行另外的设置。
- 在带有倒角/倒圆运行信息的两个程序段之间（平面轴）最多可以放置 3 不带相应信息的程序段。在没有平面内轴数据的多个程序段中，以及在进行供倒角或倒圆使用的说明时，会发出一个报警。

19.2 快速移动 G0

应用

快速移动运动 G0 用于刀具的快速定位，但不适用直接工件加工。可同时运行所有轴。此时产生一条直线轨迹。

每个轴的最大速度(快速移动)均在机床数据 MD32000 MAX_AX_VELO 中确定。如果只移动一个进给轴，则该轴以快速移动速度进行移动。例如当两个进给轴同时移动时，则所得到的轨迹速度为两个进给轴所能产生的最大可能的速度。

如果两个进给轴有最大速度，并且移动相同的轨迹位移，则最后所得到的轨迹速度 = 1.41 * 最大进给轴速度。

进给率 F 在 G0 时不起作用。尽管如此仍然保留此功能。

快速进给修调

在“AUTO”运行方式下可以通过执行如下操作来使能快进修调，使得修调开关对快速移动也生效：

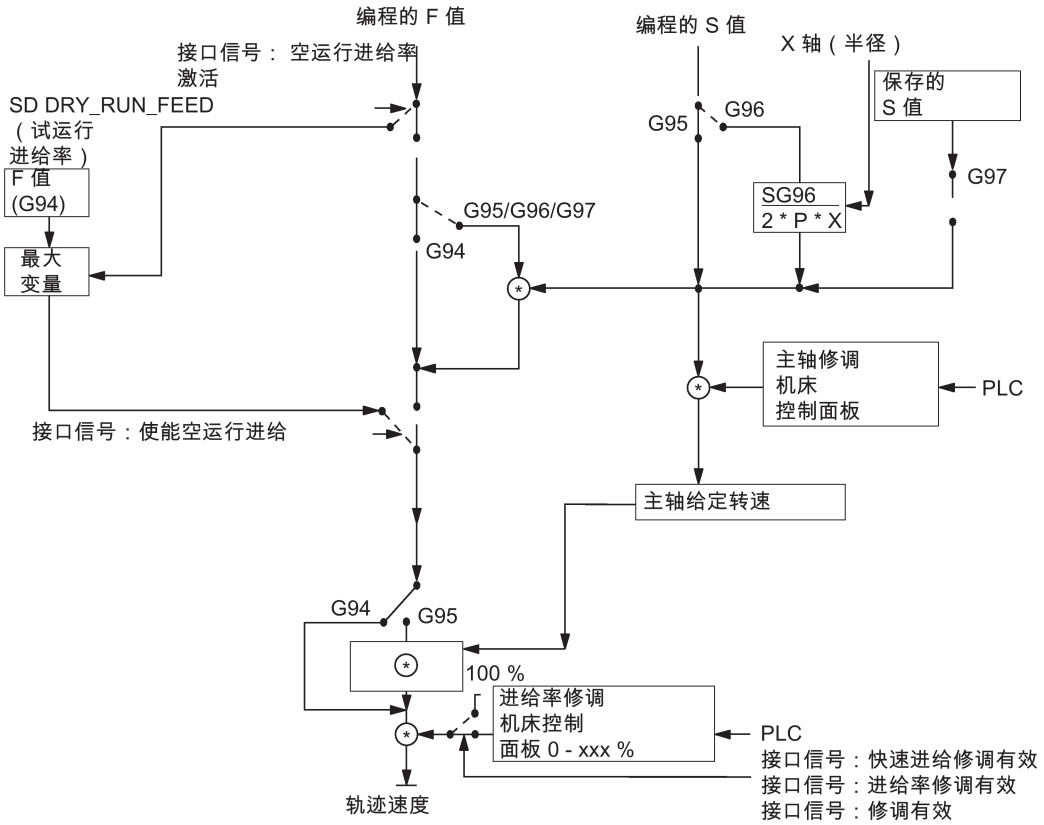


该功能激活后会在状态栏中显示 ROV。在此设置从 HMI 到 PLC 接口信号“选择快进修调”(DB1700.DBX0001.3)。此信号必须由 PLC 用户程序给出到“快速移动修调有效”(DB3200.DBX0006.6)。

19.3 进给率控制

19.3.1 概览

进给率编程和进给控制的方法：



19.3.2 进给锁定和进给/主轴停止

概述

在“进给锁定”或者“进给/主轴停止”时，进给轴停止。轨迹轮廓不变 (例外：G33 语句）。

进给锁定

通过通道专用接口信号“进给锁定”(DB3200.DBX0006.0)，所有运行方式下的轴(几何轴和附加轴)都被制动停止。

在 G33 生效时通道相关的进给禁止**不**起作用，但在 G63,G331,G332 时有效。

在工件坐标系中停止轴进给

通过接口信号“进给停止”（DB3200.DBX1000.3、DB3200.DBX1004.3，和 DB3200.DBX1008.3）制动工件坐标系 中（WCS）“JOG”方式下运行的相应的几何轴（WCS 中的轴）。

进给轴相关的进给停止

通过进给轴相关的接口信号“进给停止”(DB380x.DBX0004.3)分别使各个进给轴制动停止。

在“AUTO”方式下：如果给一个轨迹轴发出“进给停止”信号，则所有在此程序段中运行的进给轴以及参加轨迹运行的进给轴将制动停止。

在“JOG”方式下仅仅停止相关的进给轴。

进给轴专用的“进给停止”在 G33 生效时生效（但是：此处生成的轮廓误差 = 螺丝误差！）。

主轴停止

通过接口信号“主轴停止”(DB380x.DBX0004.3)使主轴制动停止。

“主轴停止”在 G33, G63 时生效。

说明

此处生成的轮廓误差 = 螺丝误差！

19.3.3 通过机床控制面板进行进给修调

概述

操作人员可以借助于进给修调开关在加工现场及时地对轨迹进给进行修调，可以相对于编程的进给率减小或增加一个百分率。进给率和修调值相乘。

轨迹进给 F 的修调范围为 0 到 120%。

程序测试时，使用快速移动修调开关，用于使零件程序运行减缓。

快速移动的修调范围为 0 到 100%。

通过主轴修调开关可以改变主轴转速和切削速度(在 G96 时)。 其修调范围为 50 到 120 %。

速度的修调在不超出机床的加速度极限和速度极限的情况下进行，并且不出现轮廓误差。

在超出极限值之前修调对**所编程的值**起作用。

通道相关的进给修调和快速修调

PLC 接口中分别有一个使能信号和一个字节作为修调系数百分比用于进给修调和快进修调。

- 接口信号“进给修调”(DB3200.DBB0004)
- 接口信号“进给修调有效”(DB3200.DBX0006.7)
- 接口信号“快进修调”(DB3200.DBB0005)
- 接口信号“快进修调有效”(DB3200.DBX0006.6)

各个修调值均通过 PLC 用格雷码给出，由机床控制面板进行控制。

所选择的进给修调值对所有的轨迹轴有效。 而所选择的快进修调值则对所有参加快速移动的进给轴有效。

如果没有单独的快进修调开关，则可以使用进给修调开关。 在此情况下，进给修调值不可以超出 100%，它受到快进修调值的限制，即最大修调只能达到 100%。

修调值的大小通过 PLC 或者通过操作面板进行选择。

通过操作面板进行选择时状态栏中显示 ROV，设置 PLC 接口信号“快进修调已选择”(DB1700.DBX0001.3)，传送到 NCK 的接口信号“快进修调有效”(DB3200.DBX0006.6)。 制面板的接口信号值也由 PLC 传送到 NCK 的接口信号“快进修调”(DB3200.DBB0005)。

在激活了 G33、G63、G331 和 G332 功能后通道相关的进给修调和快进修调不起作用。

进给轴专用的进给修调

对于每个进给轴，PLC 接口提供一个使能信号和一个字节用于进给修调百分率。

- 接口信号“进给修调”(DB380x.DBB0000)
- 接口信号“修调有效”(DB380x.DBX0001.7)

在激活了 G33，G331，G332，G63 后进给轴相关的进给修调不起作用（确保设置值不大于 100%）。

主轴修调

对于每个主轴，PLC 接口提供一个使能信号和一个字节用于主轴修调百分率。

- 接口信号“主轴修调”(DB380x.DBB2003)
- 接口信号“修调有效”(DB380x.DBX0001.7)

通过另外一个接口信号“主轴进给修调有效”(DB380x.DBX2001.0)，PLC 用户程序可以设定接口信号“进给修调”(DB380x.DBB0000)的值有效。

使用 G33 时主轴修调有效，但出于精度要求它并不起作用；这同样适用于 G331，G332。 使用 G63 时，主轴修调率为 100%。

修调有效

设定的修调值在所有的操作方式下以及对于所有机床功能都有效。 只要接口信号“快进修调有效”，“进给修调有效”和“修调有效”均已设置。

0%的修调值等效于进给锁定。

修调无效

修调给无效时(以上的接口信号为 0)，在 NC 内部修调系数等于“1”（除了第 1 个位置外），也就是说修调值为 **100%**。

说明

此值的特性是格雷码接口的第 1 开关位置。 此时如果没有设置“快进修调有效”，“进给修调有效” 和“修调有效”，也会使用第 1 开关位置的修调系数。 因此输出 **0%** 作为进给轴的修调值(和“禁止进给”作用相同)。 如果未设置“修调有效”，则主轴可以使用修调值 **50%**。

19.4 数据表

19.4.1 机床数据/设定数据

序号	名称	名称
通用机床数据		
10240	SCALING_SYSTEM_IS_METRIC	公制尺寸系统
通道专用机床数据		
20201	CHFRND_MODE_MASK	确定倒角/倒圆性能
轴专用机床数据		
32000	MAX_AX_VELO	最大轴速度
35100	SPIND_VELO_LIMIT	最大主轴转速
通道专用设定数据		
42100	DRY_RUN_FEED	空运行进给率
42010	THREAD_RAMP_DISP	螺纹攻丝中进给轴的加速性能
42110	DEFAULT_FEED	进给率缺省值

19.4.2 接口信号

序号	位	名称
通道专用		
DB3200.DBX0000	.6	使能空运行进给
DB3200.DBX0004	-	进给率修调
DB3200.DBX0005	-	快速进给修调
DB3200.DBX0006	.0	进给锁定
DB3200.DBX0006	.6	快速进给修调有效
DB3200.DBX0006	.7	进给率修调有效
DB3200.DBX1000	.3	进给停止，几何轴 1
DB3200.DBX1004	.3	进给停止，几何轴 2
DB3200.DBX1008	.3	进给停止，几何轴 3
DB1700.DBX0000	.6	空运行进给已选择
DB1700.DBX0001	.3	快速移动进给率修调已选择
DB3300.DBX0001	.2	旋转进给有效
进给轴/主轴专用		
DB380x.DBB0000	-	进给率修调

序号	位	名称
DB380x.DBB2003	-	主轴修调
DB380x.DBX0001	.7	修调有效（轴或主轴）
DB380x.DBX2001	.0	主轴进给修调有效
DB380x.DBX0004	.3	进给停止/主轴停止
DB390x.DBX2002	.0	恒定切削率有效 (主轴)
DB390x.DBX2002	.3	刚性攻丝有效(主轴)

20 刀具：刀具补偿（W1）

20.1 概述：刀具和刀具补偿

特性

数控系统具有各种刀具类型的刀具补偿功能（钻头，铣刀，车刀，...）。

- 长度补偿
- 半径补偿
- 刀具数据存储在刀具补偿存储器中
  - 通过 T 号（0 到 32000）标识刀具
  - 一个刀具最多可以定义九个刀沿（补偿数据块）
  - 刀沿通过刀具参数表示：
    刀具类型
    几何量：长度/半径
    磨损量:长度/半径
    刀沿位置（车刀）
- 选择换刀方式：立即通过 T 指令或者通过 M6
- 刀具半径补偿
  - 作用于所有插补方式的补偿：线性和圆弧
  - 外角补偿选择：过渡圆弧（G450）或者等距线交点（G451）
  - 外角/内角的自动识别

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

20.2 刀具

选择刀具

用 T 功能在程序中选择一个刀具。 是否用 T 指令直接选择一个新的刀具或者使用 M6，这要取决于机床数据 MD22550 TOOL_CHANGE_MODE（M 功能下新刀具补偿）的设定。

T 的值范围：

T 功能的值为 T0(没有刀具)到 T32000(刀具号 32000)，整数值。

在数控系统中最多可同时存储 64 个刀具。

20.3 刀具补偿

D 功能刀具补偿

一把刀具可以最多有九个刀沿。九个刀沿分为 D1 到 D9。

数控系统中最多可以同时存储刀具补偿程序段的 128 个数据字段（D 编号）：

刀沿编程从 D1（刀沿 1）到 D9（刀沿 9）。刀沿与当前有效的刀具有关。如果仅有刀沿（D1 到 D9）而无有效的刀具（T0），则此刀沿无效。用 D0 撤销当前刀具的所有刀具补偿。

换刀时选择刀沿

编程一个新的刀具（新的 T 号）并换上之后，有以下的方法用于选择刀沿：

- 编程一个刀沿号
- 不编程刀沿号，D1 自动生效。

激活刀具补偿

用 D1 到 D9 激活当前刀具某个刀沿的刀具补偿。然而刀具长度补偿和刀具半径补偿在不同的时间点生效：

- 刀具长度补偿（WLK）以轴的第一次进刀运动开始进行，在该轴中 WLK 应有效。但该运行必须是线性插补 (G0, G1)。
- 刀具半径补偿（WRK）通过在激活的平面（G17、G18 或者 G19）中编程 G41/G42 而生效。仅允许在程序组中通过 G0（快速运行）或者 G1（线性插补）选择带有 G41/G42 的 WRK。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

20.4 刀具补偿的特殊处理

对于 SINUMERIK 808D/SINUMERIK 808D ADVANCED，刀具补偿具有以下特殊情况。

设定数据的影响

操作者/编程者使用某些设定数据，会影响到如何使用刀具的长度补偿值：

- SD42940 TOOL_LENGTH_CONST
  （几何轴的刀具长度补偿分配）
- SD42950 TOOL_LENGTH_TYPE
  （不考虑刀具类型，刀具长度补偿的分配）

说明

修改的设定数据将在选择新刀沿时生效。

刀具长度和平面变化（SD42940 TOOL_LENGTH_CONST）

设定数据值为 0：

结果取决于标准定义：几何轴和磨损的长度 1 到 3 根据当前有效的 G17 到 G19 和相应的刀具类型作为该平面内第 1 到第 3 轴的长度。如果有效的 G17 到 G19 变化，长度 1 到 3 将分配给进给轴，因为横坐标、纵坐标将分配给其它的几何轴。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

设定数据值不为 0：

根据设定数据值将用于几何轴和磨损的刀具长度 1 到 3 分配给几何轴，并且当改变加工平面时（G17 到 G19），**不**改变刀具长度的分配。

刀具长度 1 到 3 分配给**车刀**（刀具类型 500 到 599）取决于设定数据 SD42940 的值以及以下不同的平面，见下表：

平面/值	长度 1	长度 2	长度 3
17	Y	X	Z
18*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) 不同于以上六种值的其中之一且值不为 0，按照 18 的值处理。

如果值为负，长度 3 的分配始终相同；和正值相比，长度 1 和 2 变化。

下表列出了刀具长度 1 到 3 分配给**钻头/铣刀**（刀具类型 100 到 299）的几何轴的情况：

平面/值	长度 1	长度 2	长度 3
17*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) 不同于以上六种值的其中之一且值不为 0，按照 17 的值处理。

如果值为负，长度 1 的分配始终相同；和正值相比，长度 2 和 3 变化。

说明

表中的说明基于几何轴 1 到 3 用 X、Y、Z 命名。进给轴的补偿值分配不取决于轴的名称，而是轴的顺序（第 1、第 2、第 3 几何轴）。

刀具类型的长度补偿（SD42950 TOOL_LENGTH_TYPE）

设定数据值为 0：

结果取决于标准定义：用于几何轴和磨损的长度 1 到 3 分配给实际的**刀具类型**（铣刀/钻头或车刀）。

文献：

SINUMERIK 808D ADVANCED 编程和操作手册

设定数据值不为 0：

不考虑实际的刀具类型，分配刀具长度：

值 1：长度的分配同铣刀。

值 2：长度的分配始终和铣刀分配一样。

说明

- 以上两个设定数据只影响刀具长度。不影响刀具半径。
- 如果 SD42940 TOOL_LENGTH_CONST 的值不为 0，且 SD42950 TOOL_LENGTH_TYPE 的值为 1 或 2，在 SD42940 表中定义的值适用于此处的刀具类型（铣刀或车刀）。

示例

SD42940 TOOL_LENGTH_CONST =18

SD42950 TOOL_LENGTH_TYPE =2

说明：

有效刀具的长度补偿始终和车刀一样（-> SD 42950 =2）。

G17 到 G19 中的长度分配和 G18 的相同（-> SD42940=18）：

长度 1 -> X 轴

长度 2 -> Z 轴

如果存在 Y 轴：长度 3 -> Y 轴

刀具半径根据实际的刀具类型和当前有效的平面起作用。

20.5 数据表

20.5.1 机床数据

序号	名称	名称
通道专用		
22360	TOOL_PARAMETER_DEF_MASK	刀具参数的定义
22550	TOOL_CHANGE_MODE	使用 M 功能的新刀具补偿

20.5.2 接口信号

序号	位	名称
通道专用		
DB2500.DBX0008	.0	T 功能 1 更改
DB2500.DBX0010	.0	D 功能 1 更改
DB2500.DBD2000	-	T 功能 1
DB2500.DBD5000	-	D 功能 1
DB2500.DBX1000	.6	M6
DB3200.DBX0013	.5	取消工件计数器

21 轮廓手轮

功能

此功能激活时，可在“AUTO”和“MDA”运行方式下通过手轮控制轨迹轴和同步轴的进给速度。

可用性

轮廓手轮功能在 SINUMERIK 808D ADVANCED 系统中作为需要授权的选件提供。

设定方式（行程设定或速度设定）

通过手轮可设定行程或速度：

- **行程设定**
  由于轴最大速度的限制，轴运行可能会滞后于手轮。但轴会走完手轮设定的行程，**脉冲不会丢失**。
- **速度设定**
  通过手轮只设定所需的运行速度。一旦手轮停止，轴也立即停止。若在一个插补周期中无手轮脉冲，则会立即进行制动。这样便不**会导致轴运行滞后于手轮**。手轮脉冲不提供行程设定。

设定方式通过以下机床数据设置：MD11346 $MN_HANDWH_TRUE_DISTANCE（手轮行程或速度输入）。有关该机床数据的更多信息，请参见 SINUMERIK 808D ADVANCED 参数手册。

进给速度

进给速度（单位为 mm/min）**取决于**下列因素：

- 所选手轮在一段时间内发出的脉冲数
- 手轮每转一个刻度发出的脉冲数，通过以下机床数据设置：MD11322 $MN_CONTOURHANDWH_IMP_PER_LATCH（轮廓手轮每转一个刻度发出的脉冲数）
- 激活的增量（INC1, 10, 100, ...）
- 第一可用几何轴的步长：MD31090 $MA_JOG_INCR_WEIGHT（手动断续运行/手轮运行中的步长）
- 程序中编写的进给速度（叠加后得出的速度可能更高）
  实际手轮速度须小于或等于程序中编写的进给速度的 1.26 倍。

进给速度的大小与下列因素**无关**：

- 程序中编写的进给速度单位（mm/min，mm/rev）

- G0 快速移动速度
- 倍率（仅 0% 会影响速度设定，即轴静止）

运行方向

运行方向取决于旋转方向：

- **顺时针方向**
  - 按程序中编写的方向运行

  达到程序段切换标准（插补结束）时，才切换至下一程序段（特性类似 G60）。
- **逆时针方向**
  - 按程序中编写方向的反方向运行

  此时最多只能运行至程序段开始处。手轮继续旋转时脉冲不会累加。在此状况下，系统触发报警 20085，表明移动方向或空运行不允许从语句块开始。

激活功能

此功能可通过 NC 程序或接口信号激活。

通过 NC 程序激活

可在 NC 程序中编写 FD=0 非模态式激活轮廓手轮，也就是说，轮廓手轮前程序段中的进给速度 F... 在轮廓手轮后的程序段中继续有效，**无需**重新编程。

通过 NC 程序激活轮廓手轮的操作步骤如下：



1. 选择加工操作区域。

2. 确保轴已回参考点，然后在“AUTO”或“MDA”方式下打开包含“FD=0”的零件程序，例如：

```
G54 G00 G90 G94 G60¶
G1 X20 Z20 F1200¶
X60 F120¶
X200 Z300 FD=0¶
X0 Z0¶
G1 X240 Z360 F=430¶
X0 Z0¶
M30¶
```



3. 选择“HANDWHEEL”运行方式，将手轮分配给第一根几何轴。有关手轮分配的详细信息，请参见章节“JOG 方式下手轮运行 (页 69)”。



4. 在 MCP 上按下此键。程序段运行至“FD=0”时，即可通过手轮控制路径和同步轴的进给率。

说明

“FD=0”仅对当行有效。轮廓手轮取消后，下一程序段仍按之前编写的进给速度运行。若轮廓手轮前的程序段中未编写进给速度，系统会发出相应报警。

不允许在一个程序段中同时编写“FD”和“F”（否则系统会发出报警）。

通过接口信号激活

通过接口信号激活/取消功能：DB3200.DBX14.0-1（激活轮廓手轮（1, 2））

地址（PLC → NCK）	信号状态	对应于...（NCK → PLC）
DB3200.DBX14.0	=1：激活轮廓手轮 1 =0：取消激活轮廓手轮 1	DB3300 DBX5.0（=1/0：轮廓手轮 1 已激活/未激活）*
DB3200.DBX14.1	=1：激活轮廓手轮 2 =0：取消激活轮廓手轮 2	DB3300 DBX5.1（=1/0：轮廓手轮 2 已激活/未激活）*

* DB3300 的信号状态为 NC 自动标注，不能在 PLC 修改。

轮廓手轮激活时，也可对其进行模拟。通过接口信号 DB3200.DBX14.3（轮廓手轮模拟）激活后，进给速度不再由轮廓手轮定义。而是采用程序中写入的进给速度。

模拟方向也通过接口信号 DB3200.DBX14.4（轮廓手轮模拟负方向）确定。模拟取消或方向切换时，轴会沿制动斜坡制动。

地址（PLC → NCK）	信号状态	对应于...（NCK → PLC）
DB3200.DBX14.3	=1：轮廓手轮模拟开启 =0：轮廓手轮模拟关闭	-
DB3200.DBX14.4	=1：轮廓手轮模拟时，反转运行方向 =0：轮廓手轮模拟时，按编程方向运行	-

您可以直接在默认的 PLC 样例程序中如下修改 PLC 子程序 37 (MCP_NCK) 网络 41 来实现所需的轮廓手轮功能：

输入信号	描述	输出信号	描述
ConHw_Key	定义 MCP 上的轮廓手轮按键	ConHw_LED	定义 MCP 上的轮廓手轮 LED 指示灯
SimConHw_Key	定义 MCP 上的轮廓手轮模拟按键	SimConHw_LED	定义 MCP 上的轮廓手轮模拟 LED 指示灯
NegDirSimConHw_Key	定义 MCP 上的轮廓手轮模拟反转方向按键	NegDirSimConHw_LED	定义 MCP 上的轮廓手轮模拟反转方向 LED 指示灯

例如，如果想要通过 MCP 上的 K7 键来控制轮廓手轮，即可将 ConHw_Key 和 ConHw_LED 的信号地址分别设为 DB1000.DBX1.7 和 DB1100.DBX1.7。

在编辑完默认的 PLC 样例程序并下载至数控系统之后，您就可以通过 MCP 上的 K7 键激活轮廓手轮功能，而后执行包含“FD=0”的零件程序，之后便可使用该功能。

说明

只要 MCP 上定义的轮廓手轮按键处于激活状态，轮廓手轮功能将一直有效。在此情况下，FD=0 始终生效。只有在“FD=0”所在程序段运行完毕，且轮廓手轮按键取消激活后，后续程序才会按照编程的进给速度运行。

倍率像执行 NC 程序时一样生效。

边界条件

- **要求**
  不允许选择固定进给率、空运行进给率、螺纹切削或攻丝。
- **限值**
  轴速度和加速度受机床数据中定义的限值的约束。
- **运行中断**
  进给保持时，此功能保持激活，但是手轮脉冲不会累加，因此变为无效。
  要求：MD32084 $MA_HANDWH_STOP_COND 位 2 = 1
  DRF
  所选的 DRF 功能会设定行程叠加。
- **针对特定通道的剩余行程删除**
  此操作会使通过轮廓手轮触发的运行中断，轴制动，程序从下一个程序段重新启动。之后轮廓手轮重新生效。

22 刀具参数：刀具后角

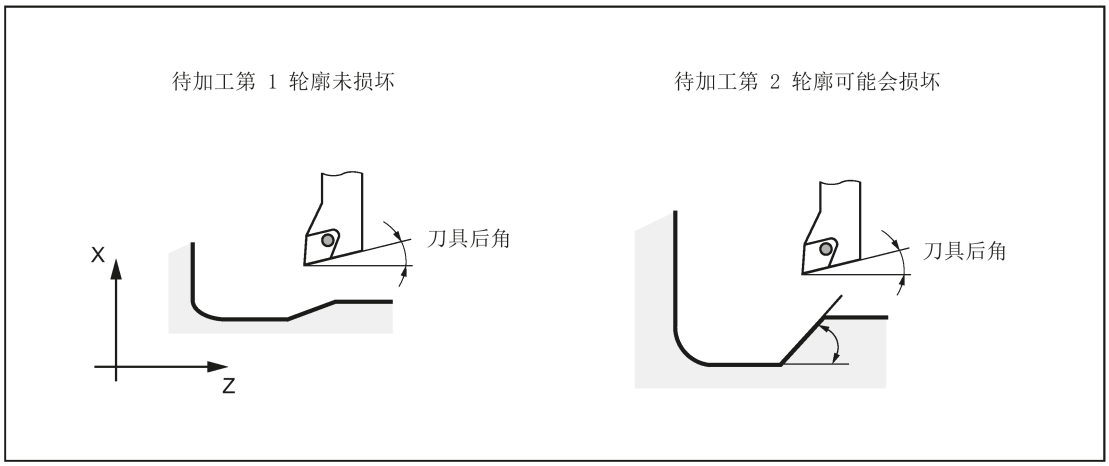
含义

在某些循环中会生成带底切的运动，这些循环会监控当前工作的刀具后角是否会造成轮廓损伤。

取值范围

在该刀具参数中输入角度（0 到 90° 不带符号）作为刀具后角。

底切时车刀的刀具后角：



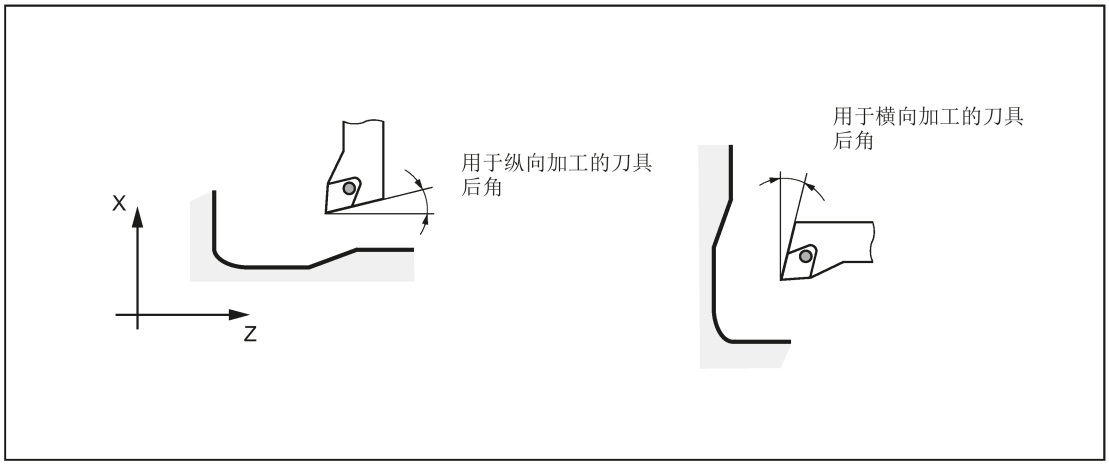
说明

要在 HMI 上激活该刀具参数，须将 MD19730[0] 的位 18 设为 1（MD19730[0] = 40000H）。
目前，刀具后角功能仅在已激活手动机床选件功能的情况下有效。

纵向或横向加工方式

根据纵向或横向加工方式输入不同的刀具后角。如果纵向加工和横向加工需要使用同一把刀具，则后角不同时必须定义两个刀沿。

用于纵向和横向加工的刀具后角：



说明

如果输入零作为刀具后角，则在底切时不会对车削循环进行监控。

23 PLC 轴控制

23.1 简介

使用 PLC 可通过用户接口数据块对轴或主轴进行控制，此时轴/主轴通过 DB 编号指定：

- **DB380x** 接口 PLC → NCK（去向轴/主轴）
- **DB390x** 接口 NCK → PLC（来自轴/主轴）

轴索引 x：0 ≤ x ≤ 最大轴索引；轴索引 = 轴编号 - 1

可以支持以下功能：

- 定位轴
- 主轴定位
- 主轴旋转
- 主轴往复
- 分度轴

更多信息请参见章节“主轴（S1）(页 158)”。

前提条件

待控制的轴必须已被指定给 PLC（PLC 轴）。通过用户接口“*跨通道取轴*”（DB380x.DBB8/DB390x.DBB8）可实现 NC 和 PLC 间的轴交换。

功能启动

每种功能都通过对应“启动”信号的上升沿触发。该信号必须保持在逻辑“1”，直至该功能被积极或消极应答（例如通过*到达位置* = “1”或*故障* =“1”）。信号“定位轴生效” = “1”表示该功能生效且输出字节有效。

终止

无法通过复位启动信号来**终止**功能，而只能通过其他接口信号进行（例如通过轴专用信号*删除剩余行程/主轴复位*，DB380x.DBX2.2）。

轴接口会反馈轴的状态信号，必要时须分析这些信号（例如*准停*、*运行指令*，→ DB390x）。

在调用 PLC 轴控制时，若轴/主轴由于通过 NC 运行被占用（存在运行指令），那么要到该运动结束后才能启动调用。在此情形下系统不会输出故障代码。

轴禁止

设置了**轴禁止**时，由 PLC 控制的轴无法运行。此时只会生成仿真的实际值。（特性与 NC 编程时相同）。

23.2 功能接口

23.2.1 用户接口：将 NC 轴作为 PLC 轴

去向轴/主轴的请求信号（节选）

首先须通过 PLC 请求轴/主轴：

DB380x	去向进给轴/主轴的信号 (PLC → NCK)[r/w]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB8	请求 PLC 轴/主轴			当此字节更改时为激活信号				请求 NC 轴/主轴

修改请求信号（DB380x.DBX8.7 或 8.0）时，必须通过激活信号（DB380x.DBX8.4）的 0→1 脉冲沿通知 NC。PLC 循环结束后必须再次复位激活信号。

来自轴/主轴的状态信号（节选）

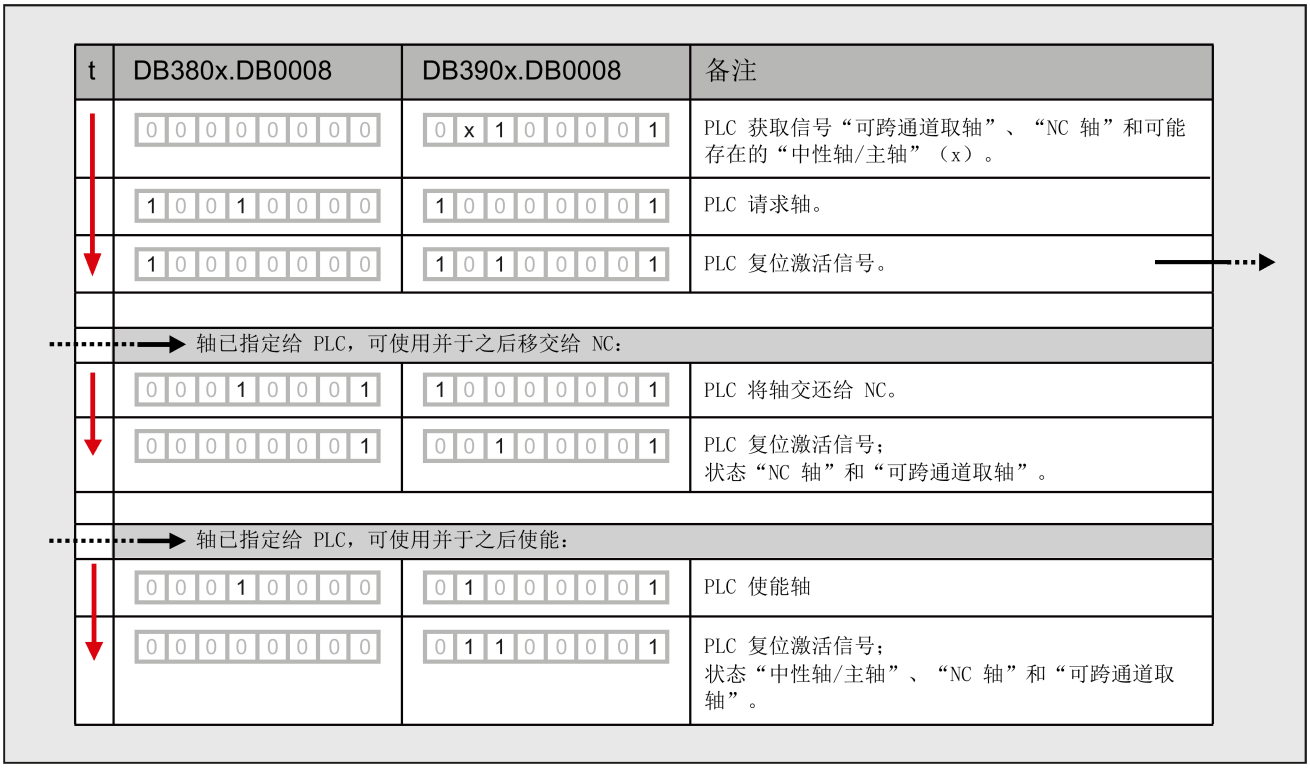
DB390x	来自轴/主轴的信号（NCK → PLC）[r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB8	PLC 轴/主轴	中性轴/主轴	允许轴交换	PLC 要求的新类型				NC 轴/主轴

说明

模拟

如需激活接口信号，必须在模拟时为每个所需轴置位机床数据 MD30350 $MA_SIMU_AX_VDI_OUTPUT。

请求和交还 PLC 轴



23.2.2 用户接口：功能

下面两个表为可供使用的接口信号一览。信号的详细说明及适用范围将在下文中介绍。

去向 PLC 轴的信号

DB380x	去向 PLC 轴的信号（PLC → NCK）[r/w]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB3000	启动定位轴	启动主轴定位	启动主轴旋转	启动主轴往复	-	-	-	-
DBB3001	-	-	停止主轴旋转	停止主轴往复	-	-	-	-
DBB3002	自动变速档选择	恒定切削速度	旋转方向同 M4	-	手轮倍率	运行尺寸，英寸（非公制）	行程条件，最短距离（DC）	行程条件，增量（IC）

DB380x	去向 PLC 轴的信号（PLC → NCK）[r/w]							
DBB3003	分度位置	-	-	-	-	-	行程条件：沿正方向逼近（ACP）	行程条件：沿负方向逼近（ACN）
DBD3004	位置（REAL，对于分度轴：DINT）							
DBD3008	进给速度（REAL），如果 < 0，则从机床数据 POS_AX_VELO 取值							

行程条件的位和方向设定的位用于定义相应的定位或运行模式，只能设置其中一个位：

含义	待设置的行程条件
绝对定位	未设定模式位
增量定位	DBB3002.0 = 1
以最短行程定位	DBB3002.1 = 1
沿正方向绝对定位	DBB3003.1 = 1
沿负方向绝对定位	DBB3003.0 = 1
旋转方向同 M4	DBB3002.5 = 1

其他位用于指定和启动相应功能，这些功能位及地址和速度会在各功能中详细说明。

来自 PLC 轴的信号

DB390x	来自 PLC 轴的信号（NCK → PLC）[r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB3000	定位轴生效	到达位置		-	-	-	运行时出现故障	轴无法启动
DBB3001	-	-	-	-	-	-	-	-
DBB3002	-	-	-	-	-	-		-
DBB3003	故障编号							

为了使用下面列出的功能，必须满足以下前提条件：

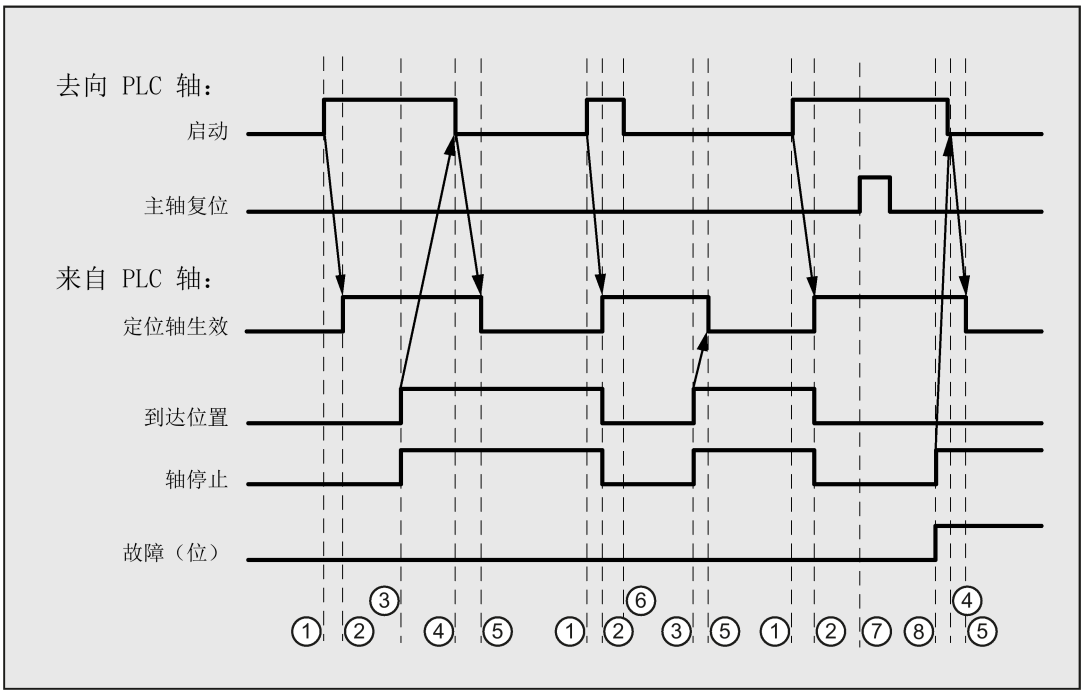
- 轴或主轴按照规定指定给了 PLC。
- 控制器使能和脉冲使能已设置。
- 设置所有控制信号后，启动信号中只有一个在 DB380x.DBB3000 中设置。

23.2.3 主轴定位

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	同一时间始终只能置位四个信号当中的一个。 所有信号 0：绝对定位
DBX3002.1	以最短行程运行	
DBX3003.0	绝对定位，负方向	
DBX3003.1	绝对定位，正方向	
DBD3004	设定位置/设定行程	设定行程，当 DBX3002.0 == 1 时
DBD3008	进给速度	0：以 MD35300 $MA_SPIND_POSCTRL_VELO 中的值运行
DBX3000.6	启动	**提示：**复位信号不会导致停止。
DBX2.2	删除剩余行程，主轴复位	终止信号，用于结束功能

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	启动信号置 1 时为 1。倍率 = 0 或到达设定位置时仍为 1。
DBX3000.6	到达位置	1：通过“精准停”到达设定位置
DBX3000.0	主轴无法启动	
DBX3000.1	运行时出现故障	1：运行时出现故障，故障编号在 DBB3003 中分析。

DB390x	NCK → PLC 状态信号	注释
DBB3003	故障编号	
DBX1.4	轴/主轴停止	1：当 $n < n_{min}$

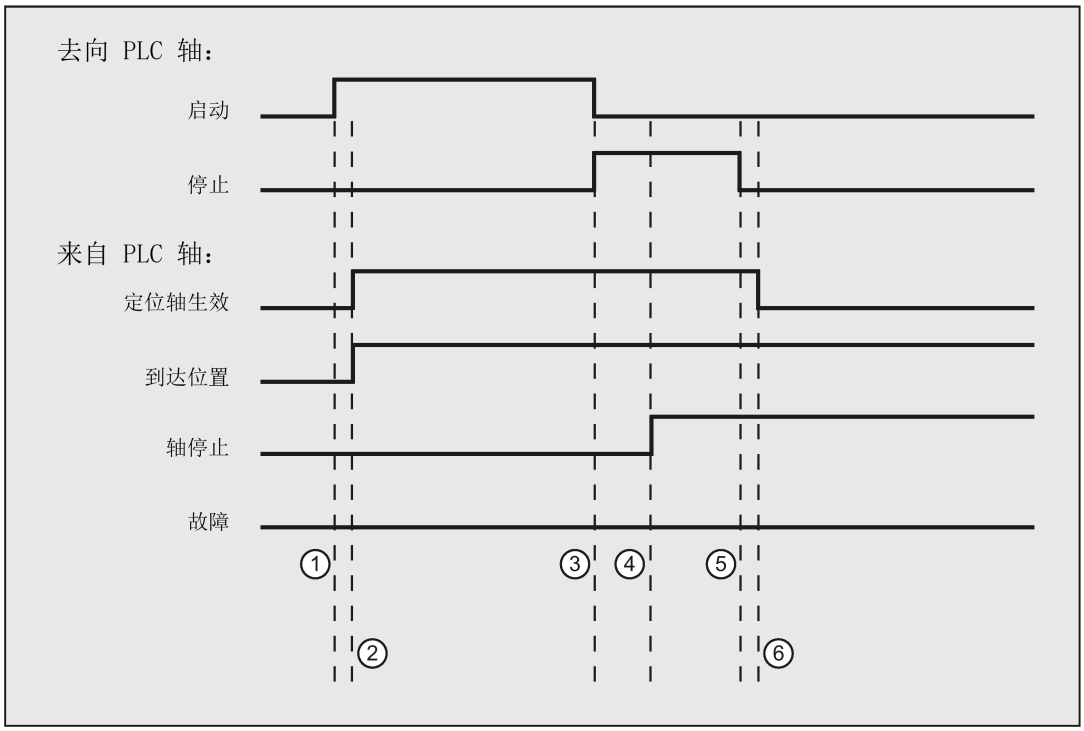


① 用户通过*启动*的上升沿触发功能。
② *定位轴生效*消息显示该功能生效且输出信号有效。*到达位置*和*轴停止*同样会被复位。赋值 = 0 时信号不会被复位。
③ 到达位置时发送*到达位置*信号，*主轴停止*置位。
④ 之后用户复位*启动*。
⑤ 之后信号*定位轴生效*复位。
⑥ 得到*定位轴生效*信号后，用户立即复位*启动*信号。
⑦ 定位通过设置*主轴复位*终止。此信号必须保持至少一个 PLC 周期。
⑧ 主轴进入停止状态（*主轴停止*），信号*故障*置位。（在此情形下系统会输出故障编号 115。）

23.2.4 主轴旋转

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	-
DBX3002.1	以最短行程运行	-
DBX3002.5	旋转方向同 M4	1：旋转方向已通过 M4 设定 0：旋转方向已通过 M3 设定
DBX3003.0	绝对定位，负方向	-
DBX3003.1	绝对定位，正方向	-
DBD3008	进给速度	主轴转速
DBX3000.5	启动主轴旋转	-
DBX3001.5	停止主轴旋转	-

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	1：启动或停止 == 1
DBX3000.6	到达位置	1：该功能已无故障启动
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	-
DBX1.4	轴/主轴停止	-

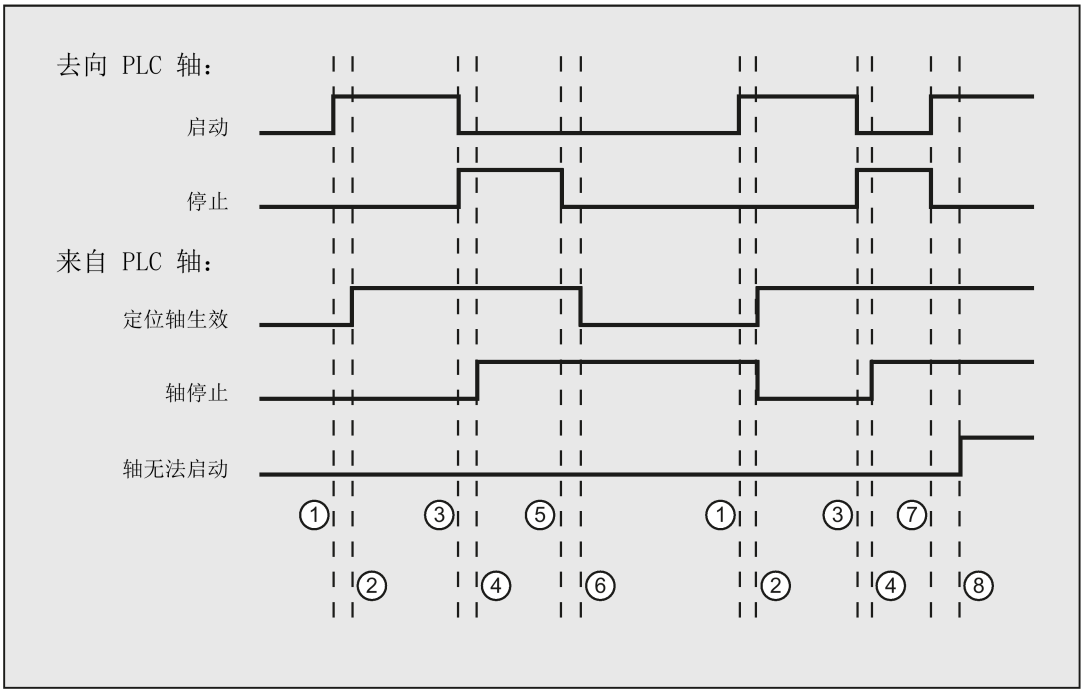


① 用户通过*启动*的上升沿触发功能。
② *定位轴生效*和*到达位置*复位。*到达位置*此处不相关。
③ 用户复位*启动*并设置*停止*，从而停止主轴旋转。
④ 主轴停止，且信号*主轴停止*置位。
⑤ 之后用户复位*停止*。
⑥ *停止*的复位会引起*定位轴生效*的复位。

23.2.5 主轴往复

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量	这些位均不允许置位。
DBX3002.1	最短行程	
DBX3002.5	旋转方向同 M4	
DBX3003.0	绝对定位，负方向	
DBX3003.1	绝对定位，正方向	
DBD3004	设定变速档	MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = 0 0 - 5：往复 MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = 1 0：往复 1：往复运行和变速档切换 M41 2：往复运行和变速档切换 M42 3：往复运行和变速档切换 M43 4：往复运行和变速档切换 M44 5：往复运行和变速档切换 M45
DBD3008	进给速度	在往复运行中无含义。往复转速取自以下机床数据：MD35400 $MA_SPIND_OSCILL_DES_VELO。
DBX3000.5	启动主轴往复	不允许在停止后直接启动。首先须复位停止（均置 0）。
DBX3001.5	停止主轴往复	

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	1：启动或停止 == 1
DBX3000.6	到达位置	1：启动后
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBX3000.0	轴无法启动	1：启动时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	
DBX1.4	轴/主轴停止	



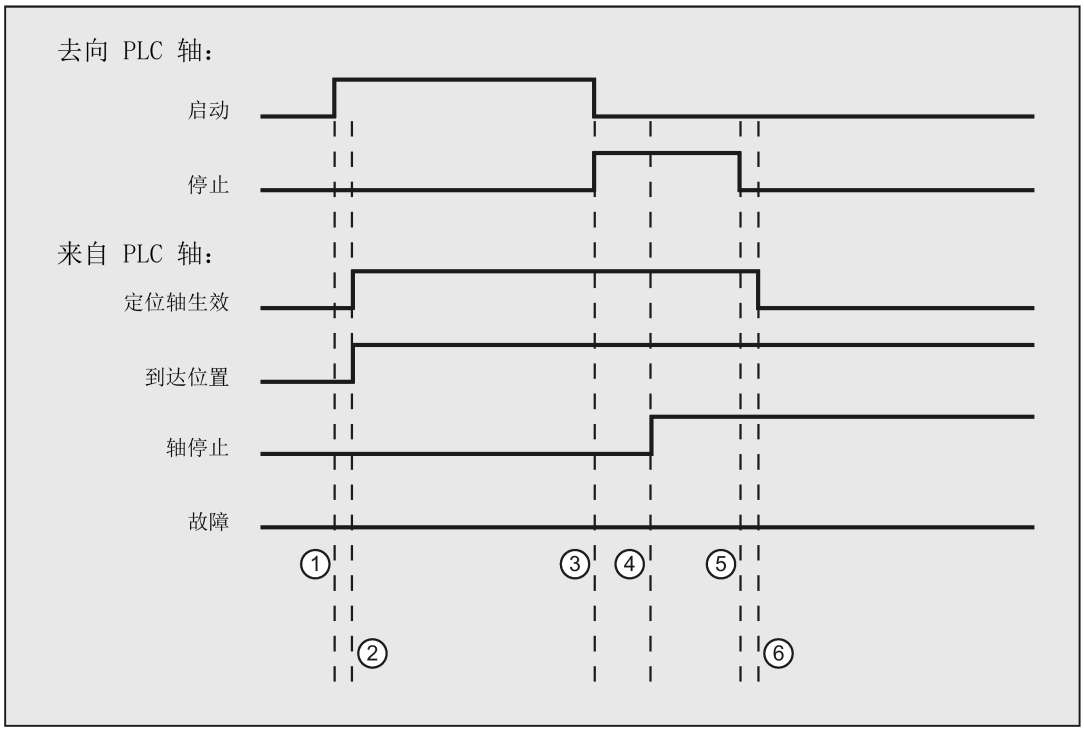
① 用户通过*启动*的上升沿触发功能。
提示：仅在*定位轴生效*信号复位时才可执行。
② *定位轴生效*和*到达位置*复位，*到达位置*此处不相关，不会被考虑。
③ 用户复位*启动*并设置*停止*，从而停止主轴往复。
④ 主轴停止，且信号*主轴停止*置位。
⑤ 之后用户复位*停止*。
⑥ *停止*的复位会引起*定位轴生效*的复位。
⑦ 用户程序中*停止*复位，*启动*在同一 PLC 循环中错误地重新置位。这样*定位轴生效*不会被复位，而是会设置
⑧ 信号*轴无法启动*（故障编号 106）。

23.2.6 分度轴

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	同一时间始终只能置位其中一个信号。 所有信号 0：绝对定位
DBX3002.1	以最短行程运行	
DBX3002.5	旋转方向同 M4	
DBX3003.0	绝对定位，负方向	
DBX3003.1	绝对定位，正方向	
DBX3003.7	分度位置	分度轴 ON
DBD3004	设定位置/设定行程	设定行程，当 DBX3002.0 == 1 时
DBD3008	进给速度	0：以 MD32060 $MA_POS_AX_VELO 中的值运行
DBX3000.7	启动定位轴	**提示：**复位信号不会导致停止。

DB380x	PLC → NCK 控制信号	注释
DBX2.2	删除剩余行程，主轴复位	终止信号，用于结束功能

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	倍率 = 0 或到达设定位置时仍为 1。
DBX3000.6	到达位置	1：通过“精准停”到达设定位置
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	-
DBX1.4	轴/主轴停止	-



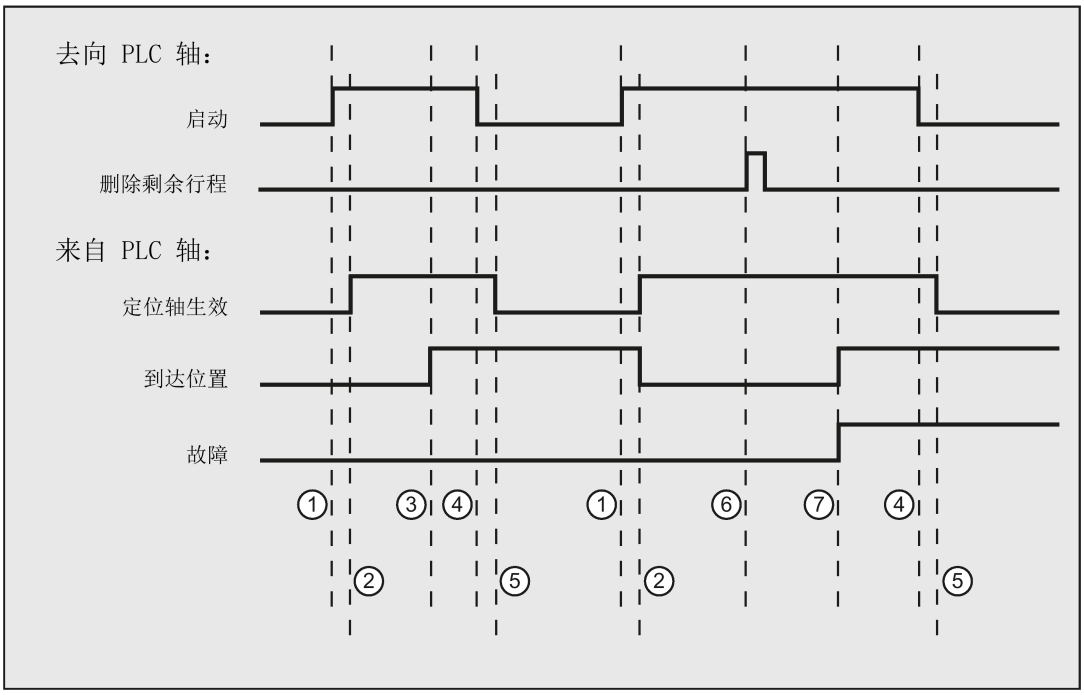
① 用户通过*启动*的上升沿触发功能。
提示：仅在*定位轴生效*信号复位时才可执行。
② *定位轴生效*和*到达位置*复位。*到达位置*此处不相关。
③ 用户复位*启动*并设置*停止*，从而停止主轴往复。
④ 主轴停止，且信号*主轴停止*置位。
⑤ 之后用户复位*停止*。
⑥ *停止*的复位会引起*定位轴生效*的复位。

23.2.7 定位轴（公制）

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	同一时间始终只能置位其中一个信号。 所有信号 0：绝对定位
DBX3002.1	以最短行程运行	
DBX3002.5	旋转方向同 M4	
DBX3003.0	绝对定位，负方向	
DBX3003.1	绝对定位，正方向	
DBD3002.2	英制运行尺寸	0：公制运行尺寸
DBX3002.3	手轮倍率	0：倍率 OFF
DBD3004	设定位置/设定行程	设定行程，当 DBX3002.0 == 1 时
DBD3008	进给速度	0：以 MD32060 $MA_POS_AX_VELO 中的值运行

DB380x	PLC → NCK 控制信号	注释
DBX3000.7	启动定位轴	**提示：**复位信号不会导致停止。
DBX2.2	删除剩余行程，主轴复位	终止信号，用于结束功能

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	倍率 = 0 或到达设定位置时仍为 1。
DBX3000.6	到达位置	1：轴已到达设定位置。
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	-



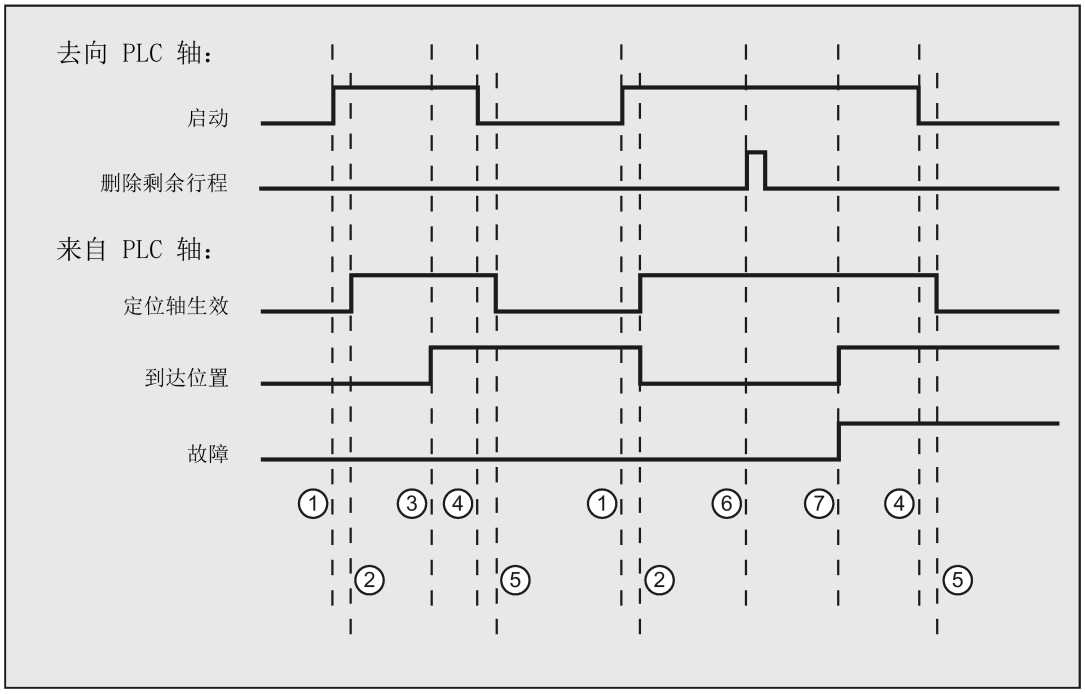
① 先通过*启动*的上升沿触发功能。
② *定位轴生效* = 1 显示该功能生效且输出信号有效。*到达位置*和*轴停止*同样会被复位。
③ 积极应答*到达位置* = 1 且*定位轴生效* = 1。
④ 收到应答后复位功能启动。
⑤ 通过功能进行信号转换。
⑥ 定位通过删除剩余行程终止，信号持续时间至少为 1 PLC 循环。
⑦ *到达位置*和*故障*信号置位，可读取故障编号（此情形下为 30）。

23.2.8 定位轴（英制）

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	同一时间始终只能置位其中一个信号。 所有信号 0：绝对定位
DBX3002.1	以最短行程运行	
DBX3002.5	旋转方向同 M4	
DBX3003.0	绝对定位，负方向	
DBX3003.1	绝对定位，正方向	
DBX3002.2	英制运行尺寸	0：公制运行尺寸
DBX3002.3	手轮倍率	0：倍率 OFF
DBD3004	设定位置/设定行程	设定行程，当 DBX3002.0 == 1 时
DBD3008	进给速度	0：以 MD32060 $MA_POS_AX_VELO 中的值运行
DBX3000.7	启动定位轴	**提示：**复位信号不会导致停止。

DB380x	PLC → NCK 控制信号	注释
DBX2.2	删除剩余行程，主轴复位	终止信号，用于结束功能

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	倍率 = 0 或到达设定位置时仍为 1。
DBX3000.6	到达位置	1：轴已到达设定位置。
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	-

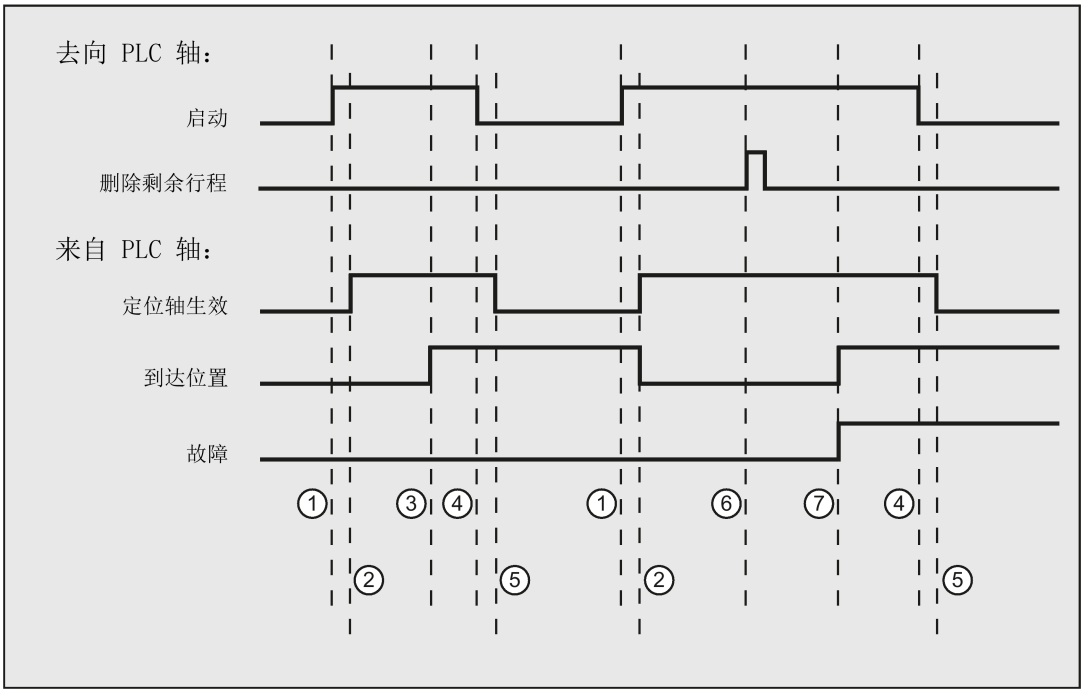


① 先通过*启动*的上升沿触发功能。
② *定位轴生效* = 1 显示该功能生效且输出信号有效。*到达位置*和*轴停止*同样会被复位。
③ 积极应答*到达位置* = 1 且*定位轴生效* = 1。
④ 收到应答后复位功能启动。
⑤ 通过功能进行信号转换。
⑥ 定位通过删除剩余行程终止，信号持续时间至少为 1 PLC 循环。
⑦ *到达位置*和*故障*信号置位，可读取故障编号（此情形下为 30）。

23.2.9 带手轮倍率的公制定位轴

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	同一时间始终只能置位其中一个信号。 所有信号 0：绝对定位
DBX3002.1	以最短行程运行	
DBX3002.5	旋转方向同 M4	
DBX3003.0	绝对定位，负方向	
DBX3003.1	绝对定位，正方向	
DBX3002.2	英制运行尺寸	0：公制运行尺寸
DBX3002.3	手轮倍率	0：倍率 OFF
DBD3004	设定位置/设定行程	设定行程，当 DBX3002.0 == 1 时
DBD3008	进给速度	0：以 MD32060 $MA_POS_AX_VELO 中的值运行
DBX3000.7	启动定位轴	**提示：**复位信号不会导致停止。
DBX2.2	删除剩余行程，主轴复位	终止信号，用于结束功能

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	倍率 = 0 或到达设定位置时仍为 1。
DBX3000.6	到达位置	1：轴已到达设定位置。
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	-

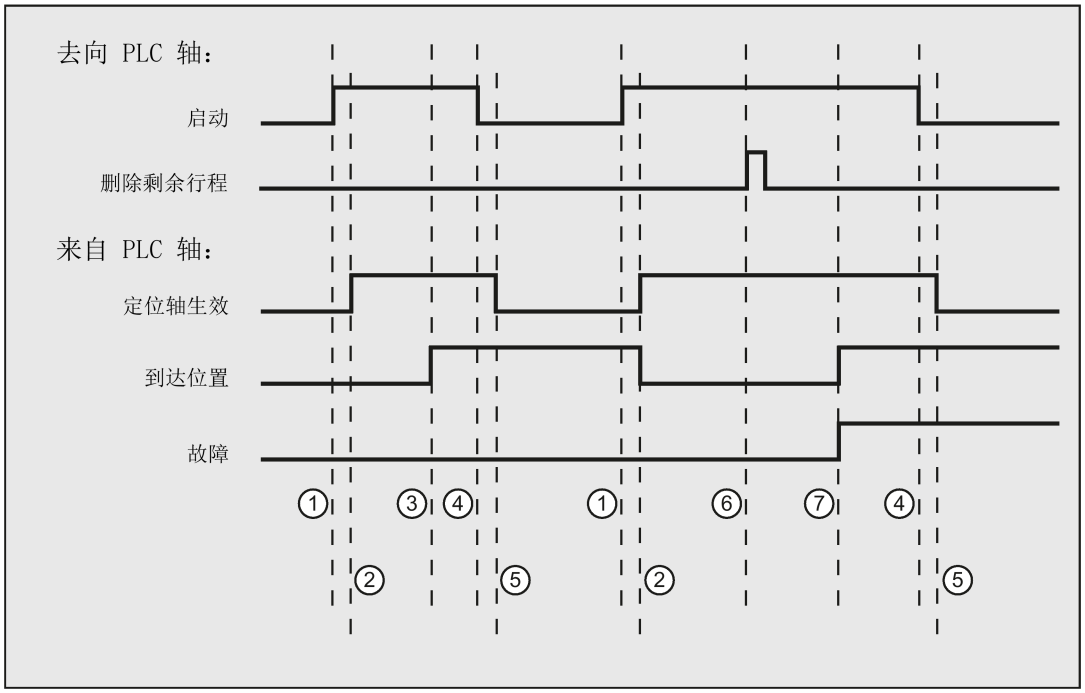


① 先通过*启动*的上升沿触发功能。
② *定位轴生效* = 1 显示该功能生效且输出信号有效。*到达位置*和*轴停止*同样会被复位。
③ 积极应答*到达位置* = 1 且*定位轴生效* = 1。
④ 收到应答后复位功能启动。
⑤ 通过功能进行信号转换。
⑥ 定位通过删除剩余行程终止，信号持续时间至少为 1 PLC 循环。
⑦ *到达位置*和*故障*信号置位，可读取故障编号（此情形下为 30）。

23.2.10 带手轮倍率的英制定位轴

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	同一时间始终只能置位其中一个信号。 所有信号 0：绝对定位
DBX3002.1	以最短行程运行	
DBX3002.5	旋转方向同 M4	
DBX3003.0	绝对定位，负方向	
DBX3003.1	绝对定位，正方向	
DBX3002.2	英制运行尺寸	0：公制运行尺寸
DBX3002.3	手轮倍率	0：倍率 OFF
DBD3004	设定位置/设定行程	设定行程，当 DBX3002.0 == 1 时
DBD3008	进给速度	0：以 MD32060 $MA_POS_AX_VELO 中的值运行
DBX3000.7	启动定位轴	**提示：**复位信号不会导致停止。
DBX2.2	删除剩余行程，主轴复位	终止信号，用于结束功能

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	倍率 = 0 或到达设定位置时仍为 1。
DBX3000.6	到达位置	1：轴已到达设定位置。
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	-



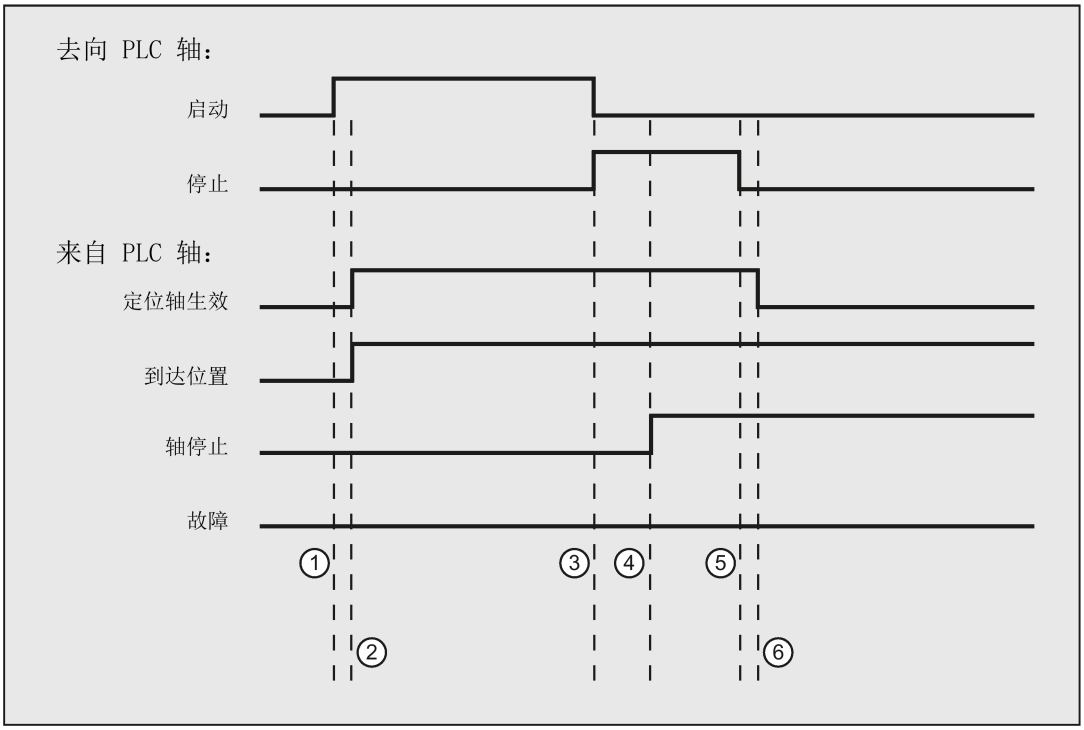
① 先通过*启动*的上升沿触发功能。
② *定位轴生效* = 1 显示该功能生效且输出信号有效。*到达位置*和*轴停止*同样会被复位。
③ 积极应答*到达位置* = 1 且*定位轴生效* = 1。
④ 收到应答后复位功能启动。
⑤ 通过功能进行信号转换。
⑥ 定位通过删除剩余行程终止，信号持续时间至少为 1 PLC 循环。
⑦ *到达位置*和*故障*信号置位，可读取故障编号（此情形下为 30）。

23.2.11 通过选择自动变速档旋转主轴

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	-
DBX3002.1	以最短行程运行	-
DBX3002.5	旋转方向同 M4	1：旋转方向已通过 M4 设定 0：旋转方向已通过 M3 设定
DBX3003.0	绝对定位，负方向	-
DBX3003.1	绝对定位，正方向	-
DBX3002.7	自动选择变速档	1：自动变速档选择 ON
DBD3008	进给速度	主轴转速
DBX3000.5	启动主轴旋转	-
DBX3001.5	停止主轴旋转	-

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	1：启动或停止 == 1
DBX3000.6	到达位置	1：设定转速已输出

DB390x	NCK → PLC 状态信号	注释
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	-
DBX1.4	轴/主轴停止	-



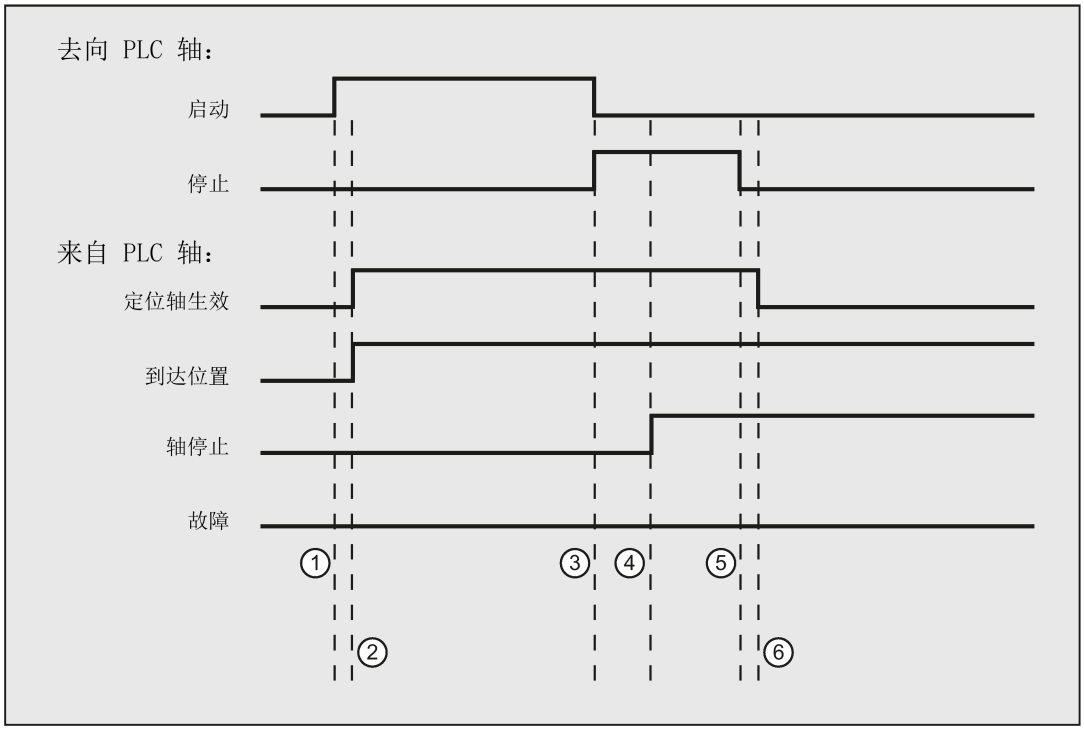
① 用户通过*启动*的上升沿触发功能。
② *定位轴生效*和*到达位置*复位。*到达位置*此处不相关。
③ 用户复位*启动*并设置*停止*，从而停止主轴旋转。
④ 主轴停止，且信号*主轴停止*置位。
⑤ 之后用户复位*停止*。
⑥ *停止*的复位会引起*定位轴生效*的复位。

23.2.12 通过恒定切削速度[m/min]旋转主轴

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	-
DBX3002.1	以最短行程运行	-
DBX3002.5	旋转方向同 M4	1：旋转方向已通过 M4 设定 0：旋转方向已通过 M3 设定
DBX3003.0	绝对定位，负方向	-
DBX3003.1	绝对定位，正方向	-
DBX3002.2	英制运行尺寸	0：公制运行尺寸
DBX3002.6	恒定切削速度	1：恒定切削速度 ON
DBD3008	进给速度	主轴转速
DBX3000.5	启动主轴旋转	-
DBX3001.5	停止主轴旋转	-

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	1：启动或停止 == 1
DBX3000.6	到达位置	1：设定转速已输出

DB390x	NCK → PLC 状态信号	注释
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	-
DBX1.4	轴/主轴停止	-



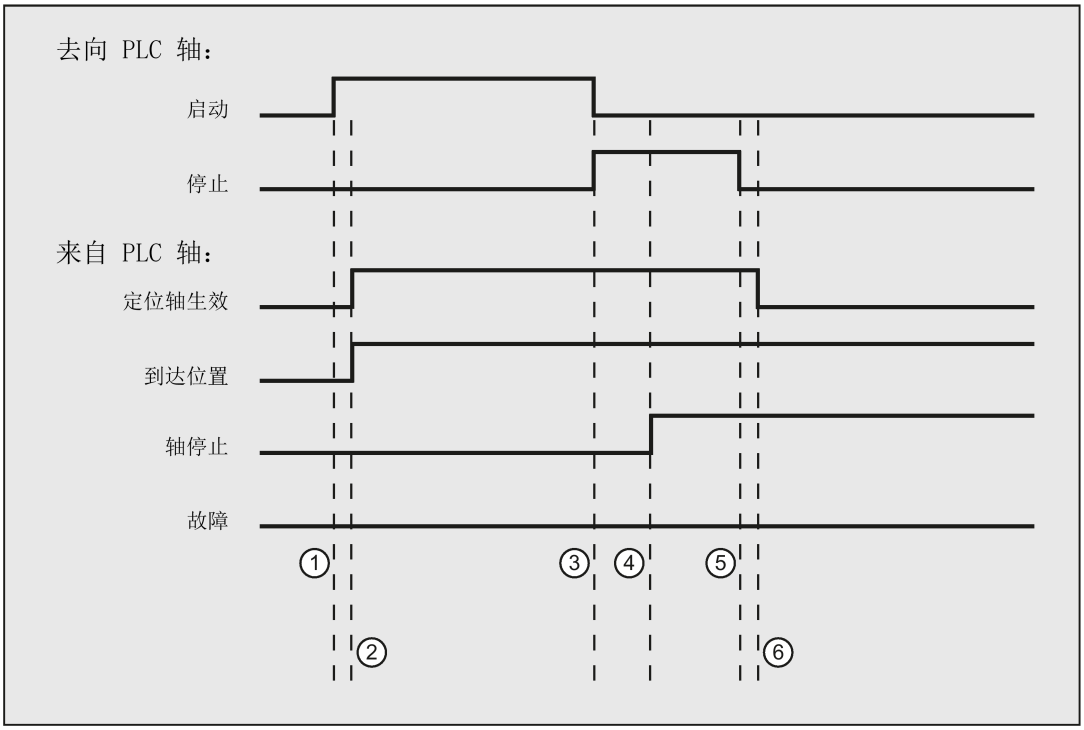
① 用户通过*启动*的上升沿触发功能。
② *定位轴生效*和*到达位置*复位。*到达位置*此处不相关。
③ 用户复位*启动*并设置*停止*，从而停止主轴旋转。
④ 主轴停止，且信号*主轴停止*置位。
⑤ 之后用户复位*停止*。
⑥ *停止*的复位会引起*定位轴生效*的复位。

23.2.13 通过恒定切削速度[feet/min]旋转主轴

DB380x	PLC → NCK 控制信号	注释
DBX3002.0	增量运行	-
DBX3002.1	以最短行程运行	-
DBX3002.5	旋转方向同 M4	1：旋转方向已通过 M4 设定 0：旋转方向已通过 M3 设定
DBX3003.0	绝对定位，负方向	-
DBX3003.1	绝对定位，正方向	-
DBX3002.2	英制运行尺寸	1：英制运行尺寸
DBX3002.6	恒定切削速度	1：恒定切削速度 ON
DBD3008	进给速度	主轴转速
DBX3000.5	启动主轴旋转	-
DBX3001.5	停止主轴旋转	-

DB390x	NCK → PLC 状态信号	注释
DBX3000.7	定位轴生效	1：启动或停止 == 1
DBX3000.6	到达位置	1：设定转速已输出

DB390x	NCK → PLC 状态信号	注释
DBX3000.1	故障	1：运行时出现故障，故障编号在 DBB3003 中分析
DBB3003	故障编号	-
DBX1.4	轴/主轴停止	-



① 用户通过*启动*的上升沿触发功能。
② *定位轴生效*和*到达位置*复位。*到达位置*此处不相关。
③ 用户复位*启动*并设置*停止*，从而停止主轴旋转。
④ 主轴停止，且信号*主轴停止*置位。
⑤ 之后用户复位*停止*。
⑥ *停止*的复位会引起*定位轴生效*的复位。

23.2.14 故障消息

如果功能无法执行，应根据故障置位以下信号：

- DB390x .DBX3000.0 == 1（轴无法启动）
- DB390x.DBX3000.1 == 1（运行中的故障）

详细的故障原因通过以下内容显示：

- DB390x.DBB3003（故障编号）

故障编号		含义
十进制	十六进制	
1	01	已同时激活多个轴/主轴功能
20	14	启动了一个功能，但未到达位置
30	1E	轴/主轴的控制权在运动结束前即已移交给 NC
40	28	轴通过 NC 程序编写，NCK 内部错误
50	32	固定分配的 PLC 轴：运行（JOG）或回参考点
60	3C	固定分配的 PLC 轴：通道状态不允许开始
100	64	为轴/主轴编写的位置错误（报警编号 1) 16830）
101	65	编写的转速过高

故障编号		含义
十进制	十六进制	
102	66	恒定切削速度的取值范围错误（报警编号 1) 14840）
104	68	副主轴：不允许的编程（报警编号 1) 22030）
105	69	无测量系统（报警编号 1) 16770）
106	6A	轴定位尚在生效（报警编号 1) 22052）
107	6B	未找到参考标记（报警编号 1) 22051）
108	6C	从转速闭环控制到位置闭环控制没有过渡（报警编号 1) 22050）
109	6D	未找到参考标记（报警编号 1) 22051）
110	6E	速度/转速为负
111	6F	转速设定值为零
112	70	无效的变速档
115	73	未到达编程的位置
117	75	在 NC 中 G96/G961 未生效
118	76	在 NC 中 G96/G961 仍有效
120	78	该轴不是分度轴（报警编号 1) 20072）
121	79	分度位置故障（报警编号 1) 17510）
125	7D	DC（最短行程）无法启用（报警编号 1) 16800）
126	7E	无法启用负绝对值（报警编号 1) 16820）
127	7F	无法启用正绝对值（报警编号 1) 16810）
128	80	无用于直径编程的平面轴（报警编号 1) 16510）
130	82	软件限位开关正（报警编号 1) 20070）
131	83	软件限位开关负（报警编号 1) 20070）
132	84	工作区域限制正（报警编号 1) 20071）
133	85	工作区域限制负（报警编号 1) 20071）
134	86	框架不可用于分度轴
135	87	配备“端面齿”的分度轴生效（报警编号 1) 17501）
136	88	配备“端面齿”的分度轴生效，且轴未回参考点（报警编号 1) 17503）
137	89	不可为经过转换的主轴/轴启动主轴模式（报警编号 1) 22290）
138	8A	针对轴超出了生效的坐标系专用工作区域正限制（报警编号 1) 20082）
139	8B	针对轴超出了生效的坐标系专用工作区域负限制（报警编号 1) 20082）
200	C8	系统报警编号 1) 450007
1) 详细的报警说明，请参见 SINUMERIK 808D ADVANCED 诊断手册。		

23.3 PLC 轴定位示例

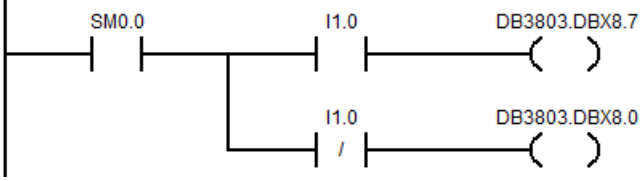
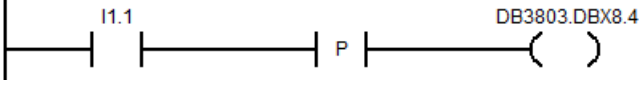
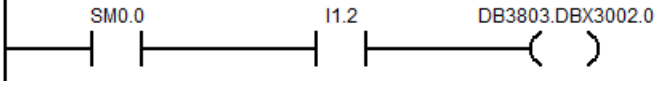
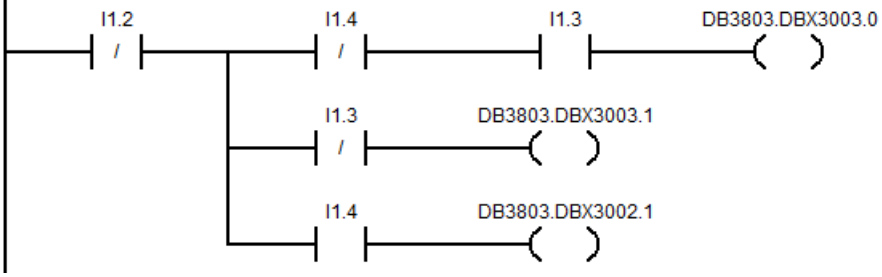
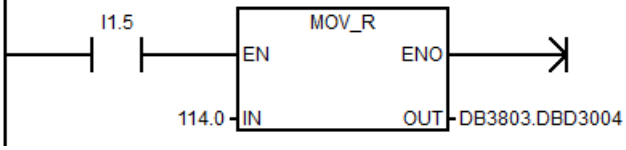
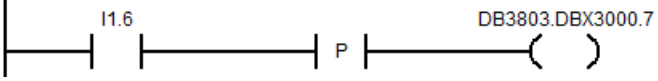
PLC 轴有三种位置控制方式：

- 通过 PLC 用户接口定位
- 利用通用机床数据里预定义的索引表定位
- 利用轴机床数据里定义的值等距控制

前提条件

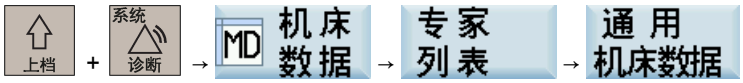
须由 PLC 通过 PLC 用户接口请求 PLC 轴。

通过 PLC 用户接口定位

网络 1 请求PLC/NC轴 	• DB3803.DBX8.7 = 1：请求 PLC 轴 • DB3803.DBX8.0 = 1：请求 NC 轴
网络 2 初始化PLC轴 	
网络 3 选择增里移动 	DB3803.DBX3002.0 = 1：行程条件，增量（IC）
网络 4 绝对方式移动或者最短路径移动（正负） 	• DB3803.DBX3003.0 = 1：行程条件，绝对负方向（ACN） • DB3803.DBX3003.1 = 1：行程条件，绝对正方向（ACP） • DB3803.DBX3002.1 = 1：行程条件，最短距离（DC）
网络 5 请求轴移动的位置 	PLC 轴移动到 114.0 位置。
网络 6 开始移动 	
结果：PLC 轴移动到设定的位置（DB3803.DBD3004 中设定的值）。	

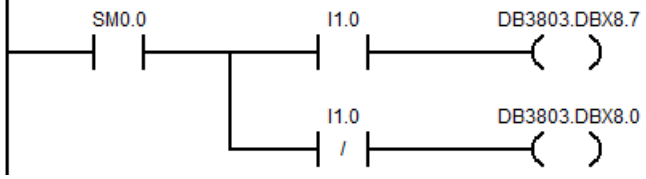
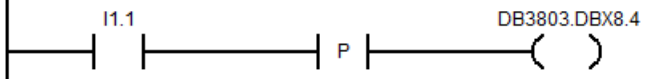
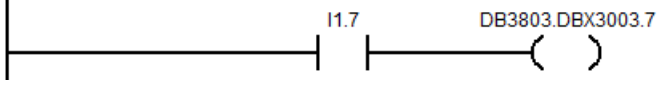
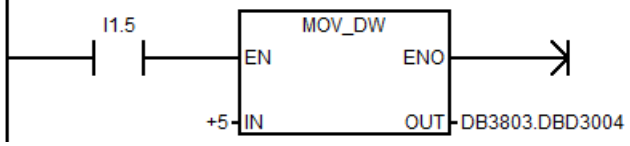
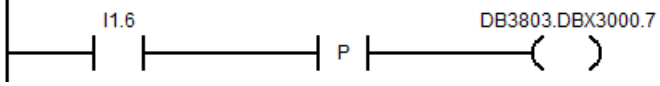
利用预定义的索引表定位

您可通过 PPU 上的如下操作对索引表进行定义：



- 在 MD10900 INDEX_AX_LENGTH_POS_TAB_1 中定义定位点的个数；
- 在 MD10910 INDEX_AX_POS_TAB_1 中输入每个点的位置值。

10900	INDEX_AX_LENGTH_POS_TAB_1	10		re
10910[0]	INDEX_AX_POS_TAB_1	0	mm|de..	re
10910[1]	INDEX_AX_POS_TAB_1	36	mm|de..	re
10910[2]	INDEX_AX_POS_TAB_1	72	mm|de..	re
10910[3]	INDEX_AX_POS_TAB_1	108	mm|de..	re
10910[4]	INDEX_AX_POS_TAB_1	144	mm|de..	re
10910[5]	INDEX_AX_POS_TAB_1	180	mm|de..	re
10910[6]	INDEX_AX_POS_TAB_1	216	mm|de..	re
10910[7]	INDEX_AX_POS_TAB_1	252	mm|de..	re
10910[8]	INDEX_AX_POS_TAB_1	288	mm|de..	re
10910[9]	INDEX_AX_POS_TAB_1	324	mm|de..	re
10910[10]	INDEX_AX_POS_TAB_1	0	mm|de..	re
10910[11]	INDEX_AX_POS_TAB_1	0	mm|de..	re
10910[12]	INDEX_AX_POS_TAB_1	0	mm|de..	re

<table>
<tr><td>网络 1 请求PLC/NC轴

</td><td>• DB3803.DBX8.7 = 1：请求 PLC 轴
• DB3803.DBX8.0 = 1：请求 NC 轴</td></tr>
<tr><td>网络 2 初始化PLC轴

</td><td></td></tr>
<tr><td>网络 3 设置索引表

</td><td>索引表定位激活</td></tr>
<tr><td>网络 4 请求轴移动的位置

</td><td>选择索引表中第五个位置</td></tr>
<tr><td>网络 5 开始移动

</td><td></td></tr>
<tr><td colspan="2">结果：PLC 轴移动到第五个位置，MD10910[4] 中设定的位置。</td></tr>
</table>

24 定位轴（P2）（仅用于 PPU16x.3）

24.1 简要说明

用于辅助运动的轴

现代机床上除了需要配备一些轴执行工件加工外，还需配备一些轴执行辅助运动，例如：

- 用于旋转刀库的轴
- 用于旋转刀塔的轴
- 用于传送工件的轴
- 用于传送托盘的轴
- 用于移动装刀器的轴（也可为多轴）

- 用于移动机械手的轴
- 用于移动套筒/顶尖的轴

执行工件加工的轴被称为轨迹轴。这些轴在通道内由插补器控制，从而同时启动、加速、到达终点和停止。

执行辅助运动的轴则以单独的进给率独立于轨迹轴运行。这类轴过去很多都采用液压驱动，并由零件程序通过辅助功能启动。借助 NC 中的轴闭环控制，可在零件程序中通过名称访问这些轴，以及在屏幕上显示其实际位置。

说明

"附加定位轴/辅助主轴"选件

用于辅助运动的轴必须不是插补轴（"真正的 NC 轴"）。"附加定位轴/辅助主轴"选件提供辅助轴功能，用于执行辅助运动。该可选功能仅适用于 PPU16x.3。

功能限制

被选择作为定位轴/辅助主轴时，轴的功能受到一定限制。此类轴**不**支持以下功能：

- 用作轨迹轴、几何轴或轨迹辅助轴
- 加入几何轴组（GEOAX）
- 螺纹切削和刚性攻丝

调试

缺省设置下，轴被定义作为插补轴：

MD30460 $MA_BASE_FUNCTION_MASK 位 8 = 0

若需将一根轴定义为功能受限的定位轴/辅助主轴运行，必须将位 8 的值设置为"1"：

MD30460 $MA_BASE_FUNCTION_MASK 位 8 = **1**

功能

通过"定位轴"功能可将执行辅助运动的轴更方便地集成至控制系统：

- 编程时：
  定位轴和工件加工轴编写在同一零件程序中，这样便不会浪费宝贵的加工时间。
  有一些针对定位轴的特殊运行指令（POS, POSA）。
- 程序测试/程序试运行时：
  程序测试和程序试运行针对所有轴有效。
- 操作时：
  加工启动以及加工过程监控针对所有轴有效。
- PLC 配置/调试时：
  在 PLC 或外部计算机（PC）中不需要考虑加工轴和辅助轴之间的同步机制。
- 系统配置时：
  不需要另一个通道。

运动和插补

每个通道都有一个轨迹插补器以及至少一个轴插补器，其具有以下插补功能：

- 轨迹插补器：
  直线插补（G1），圆弧插补（G2 / G3），样条插补等。
- 轴插补器：
  编写了一根定位轴时，控制系统中将启动一个轴插补器（直线插补 G1）
- 运动结束条件：
  满足运动结束条件 FINEA、 COARSA 或 IPOENDA，即表明到达编写的定位轴终点位置。
- 快速移动中的轨迹轴：
  快速移动（G0）中可选择通过线性插补或非线性插补运行轨迹轴。
  更多信息请参见章节"连续路径加工，准停方式和预读（B1）(页 32)"。
- 自控单轴运动：
  通过静态同步动作启动的单个 PLC 轴、指令轴或异步往复轴可独立于 NC 进行插补。

此时，通过主运行插补的进给轴/主轴响应不受 NC 程序影响。由程序运行触发的通道特性失效，以便将特定轴/通道的控制权转交给 PLC。

- 通过 PLC 控制：

  通常情况下，所有通道专用信号对轨迹轴和定位轴具有同等作用。

  定位轴还可通过附加的专用信号进行控制。

  PLC 轴由 PLC 通过基本程序中特殊的功能块运行，可以与所有其他的轴异步运行。其运行不受轨迹运动和同步动作的影响。

24.2 定位轴

定位轴作为加工方式编写在零件程序中，并且与轨迹轴即那些用于工件加工的轴一起编程。

其中可将定位轴指令和轨迹轴指令编写在同一个程序段中。但即便处于同一程序段中，轨迹轴和定位轴也不会相互插补，并且不会同时到达程序段终点（无时间耦合）。更多信息请参见章节“运动特性和插补过程 (页 207)”。

定位轴类型和程序段切换

程序段切换时间点取决于所编写的定位轴类型。更多信息请参见章节“程序段切换 (页 214)”。

类型	描述
1	当所有轨迹轴和所有定位轴到达编写的终点时，执行程序段切换。
2	当所有轨迹轴到达编写的终点时，执行程序段切换。 采用类型 2 的定位轴时，可在经过多个程序段后逼近编写的终点。
3	满足轨迹插补的运动结束条件或程序段切换条件时，可在单轴插补的制动斜坡内执行程序段切换。

运动同步

通过定位轴可从同一加工程序中触发辅助运动，并在程序段边界处（类型 1）或通过 WAITP 指令明确设定的位置（类型 2）实现运动同步。

在制动斜坡内进行程序段切换的运动结束条件

针对单轴插补可设置一个运动结束条件，以便在制动斜坡内便可进行程序段切换。

G0 下将轨迹轴作为定位轴运行

快速移动（G0）中每根轨迹轴都可作为定位轴运行。此时所有轴都相互独立地运行至终点。

因此在 G0 下依次编写 X 轴和 Z 轴，可将这两根轴作为定位轴运行。可基于设置的制动斜坡时间点（100-0%）由 X 轴触发向 Z 轴的程序段切换。X 轴运行期间 Z 轴即已启动。两根轴相互独立地运行至终点。

轴类型

定位轴既可为线性轴也可为回转轴。

定位轴还可以是分度轴。

定位轴和轨迹轴之间的独立性

以下因素确保了定位轴和轨迹轴之间的独立性：

- 不相互插补
- 每根定位轴一个单独的插补器
- 每根定位轴一个单独的进给倍率
- 单独的可编程进给率
- 单独的接口信号“删除单轴剩余行程”

关联性

定位轴和轨迹轴之间有以下关联性：

- 共用一个零件程序
- 只能在零件程序中的程序段边界处启动定位轴

- 在快速移动 G0 中将轨迹轴作为定位轴运行时，可选择采用两种不同的模式
- 无快速移动倍率
- 以下接口信号对整个通道生效，因此也对定位轴生效：
  - DB3200.DBX7.1（NC 启动）
  - DB3200.DBX7.3（NC 停止）
  - DB3200.DBX7.7（复位）
  - DB3200.DBX6.1（禁止读入）
- 程序专用和通道专用的报警也会使定位轴停止。
- 程序控制（试运行进给率、程序测试、DRF 等）也对定位轴生效。
- 程序段搜索和单程序段方式也对定位轴生效。
- 搜索中处理的最后一个含运行结束条件的程序段用于保存所有轴的设置。
- G 指令组 1（模态运行指令，包含 G0、 G1、 G2...）不适用于定位轴。

应用

定位轴的典型应用有：

- 单轴装刀器
- 无插补的多轴装刀器（PTP → 点对点运行）
- 送料器和传料器

其他应用还有：

- 在 G0 下送料器和传料器相互独立地运行至终点。
- 有多个加工程序按序列运行的机床：单轴插补的制动斜坡中程序段切换使加工步骤显著减少。

说明

定位轴不适用于需要轴间插补（轨迹插补器）的多轴装刀器。

24.3 运动特性和插补过程

24.3.1 轨迹插补器和轴插补器

轨迹插补器

每个通道都有一个轨迹插补器用于执行各种类型的插补，例如直线插补（G1）、圆弧插补（G2/G3）、样条插补等。

轴插补器

除轨迹插补器外，每个通道还配备有轴插补器。轴插补器的最大数量取决于通道轴的最大数量。

编写了一根定位轴时，控制系统中将启动一个轴插补器用于直线插补 G1。该轴插补器一直独立于轨迹插补器运行，直至到达编写的定位轴终点位置。

轨迹插补器和轴插补器间以及轴插补器相互之间不存在时间上的关联。

定位轴无法采用连续路径运行（G64）。

满足运动结束条件 FINEA、 COARSA 或 IPOENDA，即表明到达编写的定位轴终点位置。

24.3.2 自控单轴运动

功能

通过静态同步动作启动的单个 PLC 轴、指令轴或异步往复轴可独立于 NC 进行插补。此时，通过主运行插补的进给轴/主轴的以下响应不受 NC 程序影响。

- NC 停止

- 报警处理
- 程序控制
- 程序结束
- 复位

边界条件

当前通过 NC 程序运行的进给轴/主轴无法由 PLC 控制。

针对由 PLC 控制的进给轴/主轴，**无法**通过非模态和模态同步动作启动指令轴运动。系统会发出报警 20143。

由 PLC 接收轴控制

顺序说明

1. PLC → NC：请求获得轴控制权
   DB380x.DBX5004.7 = 1（PLC 控制轴）
2. NC：检查轴是主运行轴还是中立轴
3. NC：检查 PLC 是否还控制了其他轴？
4. NC 确认控制权转交：
   - DB390x.DBX3.1 = 1（PLC 控制轴）
   - 系统变量 $AA_SNGLAX_STAT = 1

结果：PLC 获得进给轴/主轴的控制权。

可能出现的情形

初始状态：轴由 PLC 控制。通道停止指令使通道处于“中断”状态。

- 轴状态“未生效” ⇒
  - 取消停止状态。
  - 轴启动指令将直接触发轴运动。
- 轴状态“生效” ⇒
  - **不**取消停止状态。
  - 按照**使用案例 2“停止轴”**生成轴状态。
  - 按照**使用案例 3“继续运行轴”**继续执行轴运动。
- 通道中执行复位 ⇒
  此进程与“由 PLC 接收控制”不同步进行。上述两种情形都有可能出现，或者轴仍指定给通道并被复位。

边界条件

通过 NC 程序运行的进给轴/主轴不能转由 PLC 控制。只有通过静态同步动作运行或者作为往复轴、中立轴、并行定位轴或指令轴运行的进给轴/主轴的控制权才能被转交。

由 PLC 交还轴控制权

顺序说明：

1. PLC → NC：PLC 将控制权交还给 NC
   DB380x.DBX5004.7 = 0（PLC 控制轴）
2. NC：检查是否有轴报警。
3. NC：检查运动是已生效还是尚未完成？如果是，则按照**使用案例 2 “停止进给轴/主轴”**用轴停止来停止运动。
4. NC：按照**使用案例 4“复位进给轴/主轴”**执行轴复位
5. NC 确认控制权转交：
   DB390x.DBX3.0 = 0（复位已执行）
   DB390x.DBX3.1 = 0（PLC 控制轴）
   DB390x.DBX3.2 = 0（轴停止生效）

系统变量 $AA_SNGLAX_STAT = 0

结果：NC 获得进给轴/主轴的控制权。

可能出现的情形

以下情形下，NC 确认控制权转交，但系统内部会针对进给轴/主轴将通道状态设置为“停止”：

- 通道停止指令使通道处于“中断”状态。
- 存在针对通道的停止报警
- 存在针对模式组的停止报警

边界条件

进给轴/主轴必须处于 PLC 控制下。

仅当不存在轴报警时，NC 才会确认获得进给轴/主轴的控制权。

基于使用案例的顺序说明

前提条件

进给轴/主轴处于 PLC 控制下。

相关 NC/PLC 接口信号

可通过以下 NC/PLC 接口信号对 PLC 控制的进给轴/主轴进行控制，而不受 NC 程序的影响：

- DB3200.DBX6.2（删除剩余行程）
- DB380x.DBX5004.1（复位）
- DB380x.DBX5004.2（继续）
- DB380x.DBX5004.6（沿制动斜坡停止）

关于自控单轴运动中 NC/PLC 接口上 NC 和 PLC 间的信号流的详细信息，请参见章节“通过 PLC 控制 (页 219)”。

使用范例 1：终止进给轴/主轴

终止轴时的特性与使用“删除剩余行程”功能相似。

DB3200.DBX6.2 = 1（删除剩余行程）

使用范例 2：停止进给轴/主轴

停止下列由主运行控制的进给轴/主轴运动：

- PLC 轴
- 异步往复轴
- 指令轴（通过静态同步动作）
- 叠加运动：$AA_OFF、DRF 手轮运行、在线刀具补偿和外部零点偏移

进给轴/主轴的从动轴运动不会被停止。

顺序说明：

- PLC → NC：请求停止进给轴/主轴
  DB380x.DBX5004.6 = 1（沿制动斜坡停止）
- NC：沿制动斜坡制动轴
- NC 确认执行：
  - DB390x.DBX0.6 = 1（粗准停）
  - DB390x.DBX0.7 = 1（精准停）
  - DB390x.DBX3.2 = 0（轴停止生效）
  - DB390x.DBX4.6/.7 = 0（运行指令 -/+）
  - 轴状态：系统变量 $AA_SNGLAX_STAT == 3

结果：进给轴/主轴停止。

说明

从动轴运动

从动轴的运动只能通过停止主动轴来中断。

回退运动

通过“扩展停止和退回”功能触发的回退运动无法停止。

文献

功能手册之特殊功能手册，扩展的停止和退回（R3）

使用范例 3：继续执行进给轴/主轴

需要继续执行被**使用案例 2 “停止轴”** 中断的运行。

顺序说明：

- PLC → NC：继续运行轴
  DB380x.DBX5004.2 = 1（继续）
- NC：检查进给轴/主轴是否存在有删除条件“CANCELCLEAR”或“NCSTARTCLEAR”的轴报警？如存在，则清除此报警。
- NC：检查轴运动是否可恢复？如果是，轴将转换至“生效”状态。
- NC 确认执行：
  - DB390x.DBX0.6 = 0（粗准停）
  - DB390x.DBX0.7 = 0（精准停）
  - DB390x.DBX3.2 = 0（轴停止生效）
  - DB390x.DBX4.6/.7 = 1（运行指令 -/+）
  - 轴状态：系统变量 $AA_SNGLAX_STAT == 4.

结果：进给轴/主轴继续运行。

边界条件

以下情形下，继续运行请求会被忽略：

- 进给轴/主轴不再由 PLC 控制。
- 进给轴/主轴不再处于停止状态。
- 存在针对进给轴/主轴的报警。

使用案例 4：复位进给轴/主轴（复位）

顺序说明：

- PLC → NC：请求复位进给轴/主轴
  DB380x.DBX5004.1 = 1（复位）
- NC：进给轴/主轴过渡至“停止”状态。
- NC：中断停止的顺序和发送至 PLC 的信号，该中断与“删除剩余行程”相同。
- NC：进给轴/主轴的内部状态已复位。
- NC：复位时生效的轴专用机床数据开始生效。

  说明

  采用通道复位时，对于由 PLC 控制的轴而言，轴专用机床数据不生效。

- NC 确认执行：
  - DB390x.DBX3.0 = 1（复位已执行）
  - DB390x.DBX3.2 = 0（轴停止生效）
  - 系统变量 $AA_SNGLAX_STAT = 1
- NC 结束此进程。

24.4 定位轴动态响应

速度

定位轴采用为其定义的专用进给率运行。更多信息请参见章节“运动特性和插补过程 (页 207)”。此进给率不受轨迹轴影响。

此进给率作为专用速度以毫米/分钟、英寸/分钟或度/分钟为单位编写。

此进给率通过轴名称固定指定给一根定位轴。

若编写定位轴时未设定进给率，那么控制系统会自动从以下轴专用机床数据取值：

MD32060 $MA_POS_AX_VELO（定位轴速度的初始设置）

编写的进给率在程序结束后失效。

进给倍率

轨迹轴和定位轴使用相互独立的进给倍率。每根定位轴均可通过专用的轴进给倍率进行调整。

快速移动倍率

快速移动倍率仅对轨迹轴生效。定位轴无快速移动插补（只有轴线性插补 G01），因此也无快速移动倍率。

旋转进给率

在 JOG 运行方式下，进给轴/主轴特性也取决于 SD41100 JOG_REV_IS_ACTIVE（JOG 模式下旋转进给率生效）的设置：

- 该设定数据生效时，进给轴/主轴总是视主主轴的情况而定采用以下旋转进给率：MD32050 $MA_JOG_REV_VELO（JOG 旋转进给率）或 MD32040 $MA_JOG_REV_VELO_RAPID（JOG 旋转进给率叠加快速移动速度）。
- 若该设定数据未生效，那么进给轴/主轴的的进给率采用 SD43300 $SA_ASSIG_FEED_PER_REV_SOURCE（定位轴/定位主轴的旋转进给率）。
- 若该设定数据未生效，而几何轴上有含旋转的有效框架命令，则该几何轴的特性取决于通道专用设定数据 SD42600 $SC_JOG_FEED_PER_REV_SOURCE。（JOG 运行方式下，有含旋转的有效框架命令的几何轴的旋转进给率）。

最大单轴加速度

在定位轴运动中，根据设定的定位轴动态模式，下列最大值中的一个生效：

- MD32300 $MA_MAX_AX_ACCEL [0]（动态模式 DYNNORM 中轨迹运动中的最大单轴加速度）
- MD32300 $MA_MAX_AX_ACCEL [1]（动态模式 DYNPOS 中轨迹运动中的最大单轴加速度）

定位轴动态模式在以下 NC 专用机床数据中设置：

MD18960 $MN_POS_DYN_MODE = <模式>

<模式>	含义
0	最大有效单轴加速度：MD32300 $MA_MAX_AX_ACCEL[0]
1	最大有效单轴加速度：MD32300 $MA_MAX_AX_ACCEL[1]

最大单轴加加速度

在加加速度限制生效的情况下运行定位轴时，下列机床数据中的一个作为最大加加速度生效：

- MD32430 $MA_JOG_AND_POS_MAX_JERK（定位轴运动时的最大单轴加加速度）
- MD32431 $MA_MAX_AX_JERK [0] 动态模式 DYNNORM 中轨迹运动中的最大单轴加加速度）
- MD32431 $MA_MAX_AX_JERK [1]（动态模式 DYNPOS 中轨迹运动中的最大单轴加加速度）

启用哪个机床数据取决于所设置的定位轴动态模式：

MD18960 $MN_POS_DYN_MODE = <模式>

<模式>	含义
	定位轴运动时作为最大单轴加加速度生效：
0	MD32430 $MA_JOG_AND_POS_MAX_JERK With active G75（运行到固定点）：MD32431 $MA_MAX_AX_JERK[**0**]
1	MD32431 $MA_MAX_AX_JERK[**1**]

24.5 编程

24.5.1 简介

说明

一个程序段中可编写的定位轴的最大数量取决于存在的通道轴的最大数量。

定义

定位轴通过以下设定确定：

- 轴类型：定位轴类型 1、类型 2 或类型 3
- 终点坐标（绝对坐标或增量坐标）
- 进给率，直线轴以[毫米/分钟]为单位，回转轴以[度/分钟]为单位

示例：定位轴类型 1

程序代码	注释
POS[Q1]=200 FA[Q1]=1000	; 轴 Q1 以 1000 毫米/分钟的进给率运行至位置 200。

示例：定位轴类型 2

程序代码	注释
POSA[Q2]=300 FA[Q2]=1500	; 轴 Q2 以 1500 毫米/分钟的进给率运行至位置 300。

说明

原则上来讲，零件程序内的每根轴都可以是轨迹轴或定位轴。 但是在一条运行程序段中必须为每根轴明确指定一种类型。

在同步动作中编程

也可通过同步动作完全异步于零件程序来进行轴的定位。

示例：

程序代码	注释
ID=1 WHENEVER $R==1 DO POS[Q4]=10 FA[Q3]=990	; 轴进给率采用固定设定。

程序段切换

采用定位轴类型 1 和类型 2 时，可通过以下方式设置程序段切换：

FINEA=<轴名称> 或

FINEA[<轴名称>]

COARSEA=<轴名称> 或

COARSEA[<轴名称>]

IPOENDA=<轴名称> 或

IPOENDA[<轴名称>]

采用定位轴类型 3 时，还可通过以下指令设置在单插补制动斜坡内进行程序段切换：

IPOBRKA=<轴名称> 或

IPOBRKA(<轴名称>[,<时间点*>])

* 程序段切换时间点，是制动斜坡的 %

绝对坐标 / 增量坐标

终点可以绝对坐标（G90）或增量坐标（G91）编写。

示例	含义
	终点坐标编程：
G90 POS[Q1]=200	以绝对坐标
G91 POS[Q1]=AC(200)	以绝对坐标
G91 POS[Q1]=200	以增量坐标
G90 POS[Q1]=IC(200)	以增量坐标

重新编写类型 2 的定位轴

对于类型 2（越过程序段边界运动）的定位轴，必须能在零件程序中确定定位轴是否已到达终点位置。只有这样才能重新编程此定位轴（否则会触发报警）。

若先编写了 POSA ，之后重新编写了 POSA 和 IPOBRKA （在制动斜坡中进行程序段切换），则不会触发报警。更多提示请参见“可设置的程序段切换时间点”章节中的语言指令 IPOBKA

协调（WAITP）

使用用协调指令 WAITP 可以在 NC 程序中标记位置，并在此位置上等待，直到在前一 NC 程序段中用 POSA 编程的轴到达终点。

程序代码	注释
N10 G01 G90 X200 F1000 POSA[Q1]=200 FA[Q1]=500	
N15 X400	
N20 WAITP(Q1)	; 程序处理自动停止，直至 Q1 到达位置。
N25 X600 POS[Q1]=300	; Q1 为定位轴类型 1（采用程序段 N10 中的进给率 FA[Q1]）。
N30 X800 Q1=500	; Q1 为轨迹轴（采用程序段 N10 中的轨迹进给率 F1000）。

WAITP 需要单独占用一个程序段。

需要等待的轴必须明确设定。

示例：

程序代码	注释
N10 G01 G90 X200 F1000 POSA[Q1]=200 FA[Q1]=500	
N15 X400	
N20 WAITP(Q1)	; 程序处理自动停止，直至 Q1 到达位置。
N25 X600 POS[Q1]=300	; Q1 为定位轴类型 1（采用程序段 N10 中的进给率 FA[Q1]）。
N30 X800 Q1=500	; Q1 为轨迹轴（采用程序段 N10 中的轨迹进给率 F1000）。

刀具补偿

定位轴可通过零点偏移进行刀具长度补偿。这例如会使装刀器的定位行程发生变化。因此可将零点偏移代替刀具长度补偿用于以下情形：将装有不同尺寸刀具的装刀器需要越过一个障碍。

程序结束

程序结束（程序状态已选择）会一直延迟，直至所有轴（轨迹轴 + 定位轴）到达编写的终点。

24.5.2 外部编程时的旋转进给率

通过下面两个设定数据，可设定从另一根回转轴/主轴推导出定位轴的旋转进给率：

SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE（定位轴/主轴的旋转进给率）

SD42600 JOG_FEED_PER_REV_SOURCE（JOG 旋转进给率调整）

可以进行下列设置：

值	含义
0	不选择旋转进给率。
>0	旋转进给率从此处设定的机床轴下标对应的回转轴/主轴推导。
-1	旋转进给率从进给轴/主轴所在通道的主主轴推导。
-2	旋转进给率从机床轴下标为 0 的回转轴/主轴推导。
-3	旋转进给率从进给轴/主轴所在通道的主主轴推导。主主轴停止时不选择旋转进给率。

24.6 程序段切换

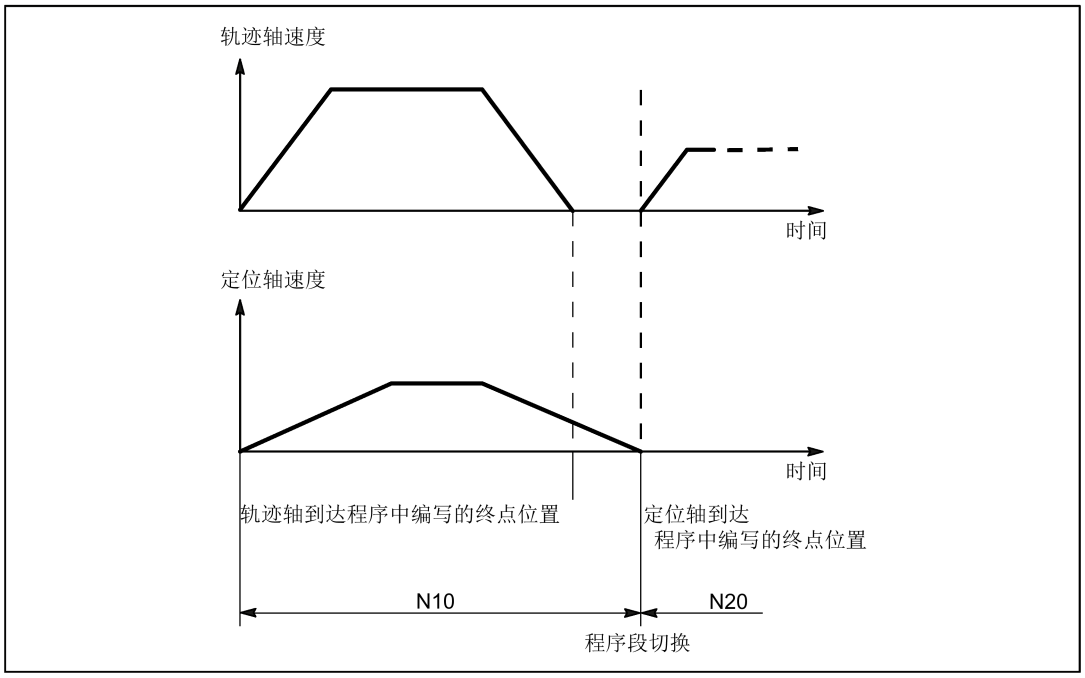
24.6.1 程序段切换

轨迹轴和定位轴相互独立地进行插补，因此它们会在不同的时间点到达编写的终点位置。若将轨迹轴和定位轴编写在同一程序段中，那么程序段切换特性取决于可编写的定位轴类型。

类型 1：程序段相关定位轴

特性：

- **所有轨迹轴和定位轴**符合编写的运动结束条件后，立即进行程序段切换：
  - 轨迹轴：G601, G602, G603
  - 定位轴：FINEA, COARSA, IPOENDA
- 定位轴的编程：POS[<轴>]



说明

连续路径运行

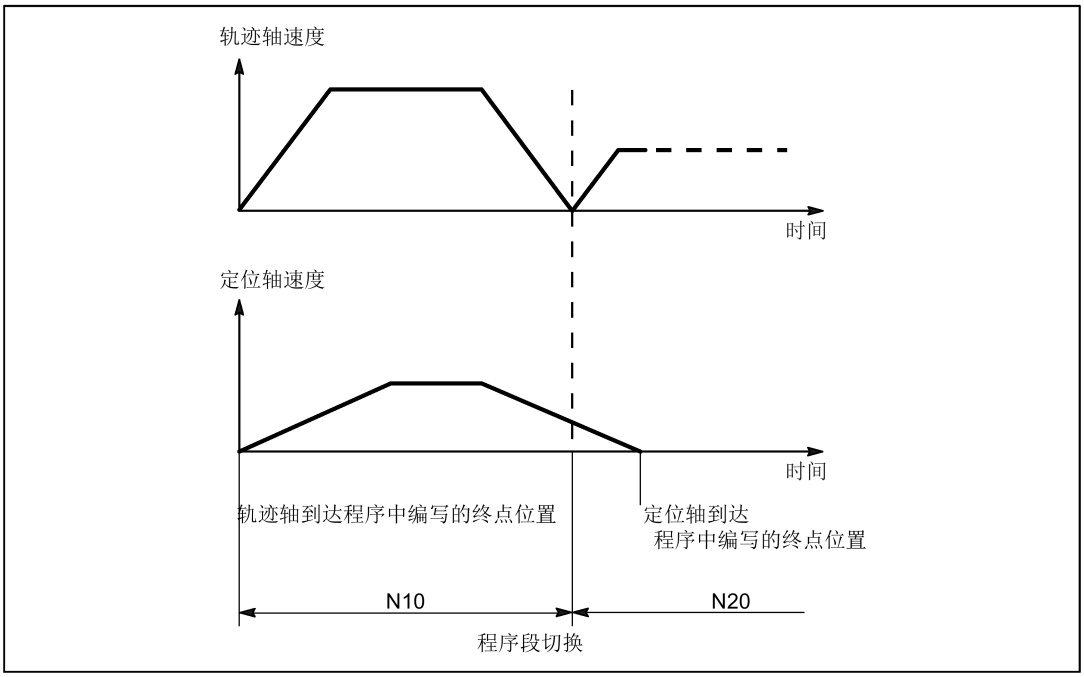
仅当定位轴先于轨迹轴满足运动结束条件时（上图不符合此条件），才能进行跨程序段的连续路径运行（G64）。

类型 2：跨程序段定位轴

特性：

- **所有轨迹轴**符合编写的运动结束条件（G601, G602, G603）后，立即进行程序段切换。
- 定位轴的编程：POSA[<轴>]

- 定位轴跨程序段运行至编写的终点位置。满足运动结束条件前不允许重新编程定位轴。

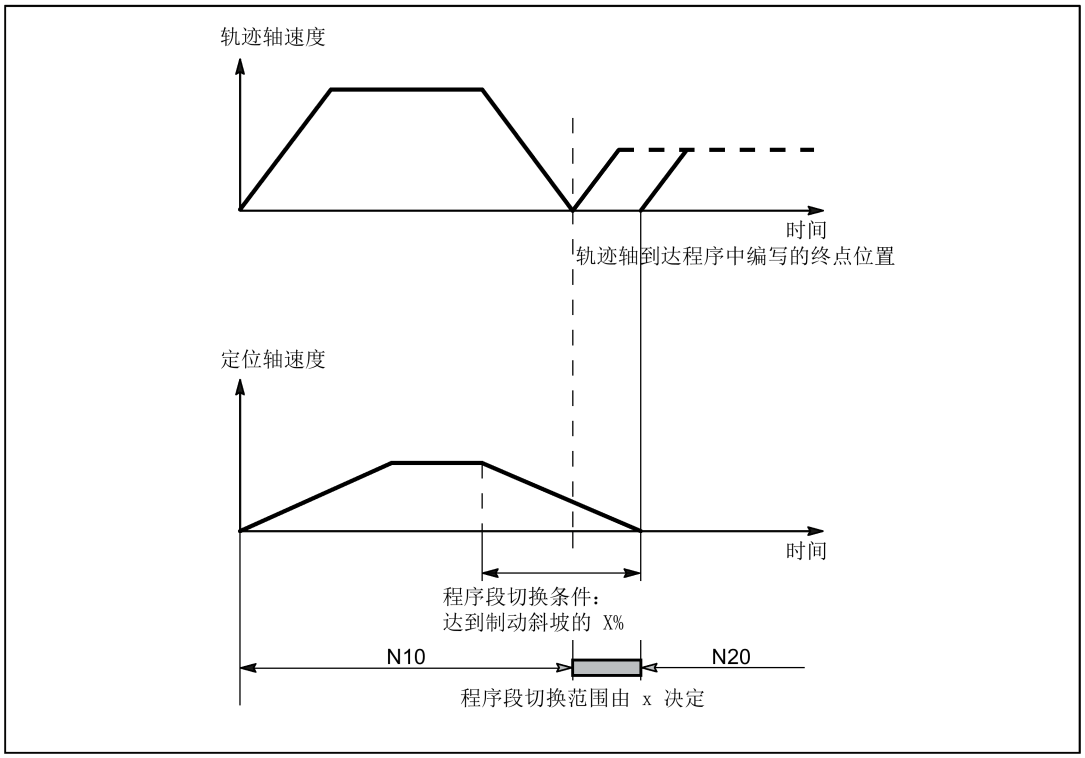


24.6.2 可设置的程序段切换时间点

类型 3：与程序段有条件关联的定位轴

特性：

- **所有轨迹轴和定位轴**符合编写的运动结束条件后，立即进行程序段切换：
  - 轨迹轴：G601, G602, G603
  - 定位轴：IPOBRKA
- 定位轴的编程：
  - N(x) IPOBRK(<轴>[,<时间点>]); 独立程序段
  - N(x+1) POS[<轴>]



程序段切换条件："制动斜坡"（IPOBRKA）

如果激活了程序段切换条件"制动斜坡"，并对可选参数 <时间点> 进行了赋值，那么该值会在下一次定位运动中生效，并和主运行同步地写入设定数据中。没有指定该时间时，设定数据中的当前值生效。

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

可进行程序段切换的时间点以制动斜坡百分比的形式设定：

- 100% = 制动斜坡开始
- 0% = 制动斜坡结束，与程序段切换条件 IPOENDA 含义相同

编程

IPOBRKA(<轴>[,<时间点>])

IPOBRKA:	程序段切换条件：减速斜坡
	生效方式： 模态有效
<轴>：	轴通道名称 (X, Y, ...)
<时间点>:	程序段转换时间，是制动斜坡的 % 值：
	• 100% = 制动斜坡开始
	• 0% = 制动斜坡结束，和 IPOENDA 含义相同
	类型： REAL

重新编写轴运动结束条件（FINEA 、COARSEA、IPOENDA ），对应轴的 IPOBRKA 会被取消。

附加程序段切换条件："公差窗口"（ADISPOSA）

可将一个以程序段终点（可选择实际位置或设定位置）为中心的公差窗口设为附加程序段切换条件。这样程序段切换须满足两个条件：

- 程序段切换条件："制动斜坡"
- 程序段切换条件："公差窗口"

编程

ADISPOSA(<轴>[,<模式>,<窗口大小>])

ADISPOSA：	用作运动结束条件的公差窗口		
	生效方式：	模态有效	
<轴>：	轴通道名称 (X, Y, ...)		
<模式>:	公差窗口的中心		
	类型：	INT	
	取值范围：	0	公差窗口不生效
		1	公差窗口以设定位置为中心
		2	公差窗口以实际位置为中心
<窗口大小>：	公差窗口的大小		
	类型：	REAL	

运动结束条件的系统变量

可通过系统变量 $AA_MOTEND 读取当前生效的运动结束条件。

边界条件

提前程序段切换

在以下情形下无法提前进行程序段切换：

- 往复轴
  在带单次进给的往复运动中，往复运动必须保持生效，直至轴通过单次进给到达终点位置。
- 手轮
  在手轮运行中，最近一次设置的运动结束条件生效。

轴状态更改

如果轴在制动斜坡内进行了程序段切换，那么下一个程序段中只允许在相同的轴状态下重新编写该轴。

轴状态发生切换时，例如从 POS 切换至 SPOS，最近一次编写的运动结束条件 FINEA、COARSEA、IPOENDA 将生效。这也适用于以下情形：

- 定位轴切换为轨迹轴
- 等待定位运动结束时：WAITP、M30、工艺循环技术、预处理停止
- 激活或取消速度叠加时

示例

零件程序中采用程序段切换条件“制动斜坡”

程序代码	注释
	; 缺省设置生效。
N10 POS[X]=100	; X 轴定位到位置 100。
	; 程序段切换条件：“精准停”
N20 IPOBRKA(X,100)	; 程序段切换条件：“制动斜坡”，100% = 制动斜坡开始
N30 POS[X]=200	; 一旦 X 轴开始制动，就立即切换程序段。
N40 POS[X]=250	; X 轴不再制动至位置 200，而是继续运行至 250。一旦 X 轴开始制动就立即切换程序段。
N50 POS[X]=0	; X 轴制动并返回到位置 0， 在位置 0 和“精准停”条件下切换程序段。
N60 X10 F100	; X 轴作为轨迹轴运动到位置 10。
N70 M30	

同步动作中采用程序段切换条件“制动斜坡”

在工艺循环中：

程序代码	注释
FINEA	; 运动结束条件：“精准停”
N10 POS[X]=100	; 当 X 轴到达位置 100 和精准停窗口时，进行工艺循环程序段切换。
N20 IPOBRKA(X,100)	; 激活程序段切换条件“制动斜坡”， 100% = 制动斜坡开始处
N30 POS[X]=200	; 一旦 X 轴开始制动，就立即进行工艺循环程序段转换。
N40 POS[X]=250	; X 轴不再制动至位置 200，而是继续运行至 250。一旦 X 轴开始制动就立即进行工艺循环中的程序段切换。
N50 POS[X]=0	; X 轴制动并返回到位置 0， 在位置 0 和“精准停”条件下切换程序段。
N60 M17	

零件程序中采用程序段切换条件“制动斜坡”和“公差窗口”

程序代码	注释
	; 缺省设置生效。
N10 POS[X]=100	; X 轴定位到位置 100。 程序段切换条件：“精准停”
N20 IPOBRKA(X,100)	; 程序段切换条件：“制动斜坡”，100% = 制动斜坡开始
N21 ADISPOSA(X,1,0.5)	; 公差窗口：1 = 设定位置，公差 = 0.5
N30 POS[X]=200	; 一旦 X 轴开始制动，就立即切换程序段。
N40 POS[X]=250	; X 轴不再制动至位置 200，而是继续运行至 250。一旦 X 轴开始制动就立即切换程序段。
N50 POS[X]=0	; X 轴制动并返回到位置 0， 在位置 0 和“精准停”条件下切换程序段。
N60 X10 F100 N70 M30	

同步动作中采用程序段切换条件“制动斜坡”和“公差窗口”

在工艺循环中：

程序代码	注释
FINEA	; 运动结束条件：“精准停”
N10 POS[X]=100	; 当 X 轴到达位置 100 和精准停窗口时，进行工艺循环程序段切换。
N20 IPOBRKA(X,100)	; 激活程序段切换条件“制动斜坡”，100% = 制动斜坡开始处
N21 ADISPOSA(X,2,0.3)	; 公差窗口：2 = 实际位置，公差 = 0.3
N30 POS[X]=200	; 一旦 X 轴开始制动以及 X 轴的实际位置 >= 199.7，立即进行工艺循环程序段切换。
N40 POS[X]=250	; X 轴不再制动至位置 200，而是继续运行至 250。一旦 X 轴开始制动以及 X 轴的 >= 249.7，立即进行工艺循环程序段切换。
N50 POS[X]=0	; X 轴制动并返回到位置 0， 在位置 0 和“精准停”条件下切换程序段。
N60 M17	

若以下设定数据中已输入了相应值，那么在程序段 N20 中可不设定时间写入 IPOBRKA(X)。

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE[X] == 100

24.6.3 程序段搜索时的运动结束条件

最后一个程序段用作容器

最后一次编写的轴运动结束条件会被记录，并在一个动作程序段中输出。搜索中处理的最后一个包含运动结束条件的程序段用作容器，在其中可设置针对所有通道编写的所有运行结束条件。

示例

两个动作程序段，包含针对三根轴的运动结束条件：

程序代码	注释
N01 G01 POS[X]=20 POS[Y]=30 IPOENDA[X]	; 最后一次编写的运动结束条件 IPOENDA
N02 IPOBRKA(Y, 50)	; 第二个动作程序段，用于 Y 轴 IPOENDA
N03 POS[Z]=55 FINEA[Z]	; 第二个动作程序段，用于 Z 轴 FINEA
N04 $A_OUT[1]=1	; 第一个动作程序段，用于作为数字量输出
N05 POS[X]=100	;
N06 IPOBRKA(X, 100)	; 第二个动作程序段，用于 X 轴 IPOBRKA
...	; 并作为编写的所有运动结束条件的容器
TARGET:	; 程序段搜索目标：

在第一个动作程序段中输出数字量输出：

$A_OUT[1] = 1。

在第二个动作程序段中设置运动结束条件：

X 轴：IPOBRKA / $SA_IPOBRAKE_BLOCK_EXCHANGE[AX1]=100

Y 轴：IPOBRKA / $SA_IPOBRAKE_BLOCK_EXCHANGE[AX2]=50

Z 轴：FINEA。此外为 X 轴标记最后一次编写的运动结束条件 IPOENDA。

24.7 通过 PLC 控制

PLC 轴由 PLC 运行，可与所有其他轴异步运动。其运行不受轨迹运动和同步动作的影响。

更多关于如何通过 PLC 用户界面控制定位轴的信息，请参见章节 “PLC 轴控制 (页 188)”。

24.8 特殊功能下的特性

24.8.1 空运行进给率（DRY RUN）

编写的进给率不大于空运行进给率的情况下，空运行进给率也对定位轴生效。

SD42100 $SA_DRY_RUN_FEED 中输入的空运行进给率的有效性可通过 SD42101 $SA_DRY_RUN_FEED_MODE 控制。参见

更多信息请参见章节“进给率（V1）(页 174)”。

24.8.2 单程序段

定位轴类型 1

采用定位轴类型 1 时，单程序段方式同样生效。

定位轴类型 2

定位轴类型 2 在单程序段方式下仍会越过程序段边界。

定位轴类型 3

定位轴类型 3 在单程序段方式下仍会越过程序段边界。

24.9 示例

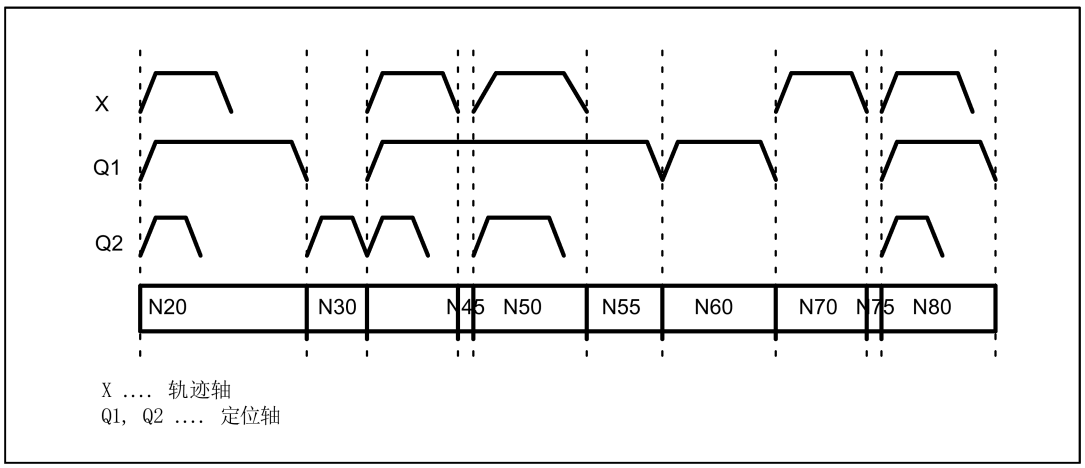
运动特性和插补过程

在此示例中，定位轴 Q1 和 Q2 分别代表两个独立的运动单元。两者不进行插补。此示例中定位轴将作为类型 1（例如 N20 中）和类型 2（例如 N40 中）编程。

程序示例

程序代码

```
N10 G90 G01 G40 T0 D0 M3 S1000
N20 X100 F1000 POS[Q1]=200 POS[Q2]=50 FA[Q1]=500
FA[Q2]=2000
N30 POS[Q2]=80
N40 X200 POSA[Q1] = 300 POSA[Q2]=200] FA[Q1]=1500
N45 WAITP[Q2]
N50 X300 POSA[Q2]=300
N55 WAITP[Q1]
N60 POS[Q1]=350
N70 X400
N75 WAITP[Q1, Q2]
N80 G91 X100 POS[Q1]=150 POS[Q2]=80
N90 M30
```



X .... 轨迹轴
Q1, Q2 .... 定位轴

24.10 数据表

24.10.1 机床数据

序号	标识符	名称
NC 专用		
18960	POS_DYN_MODE	定位轴动态模式
通道专用		
20730	G0_LINEAR_MODE	G0 的插值特性
进给轴/主轴专用		
30460	BASE_FUNCTION_MASK	轴功能
32060	POS_AX_VELO	定位轴的进给率
32300	MAX_AX_ACCEL	最大轴加速度
32430	JOG_AND_POS_MAX_JERK	定位轴运动中的单轴加加速度上限
32431	MAX_AX_JERK	轨迹运动中的单轴加加速度上限

24.10.2 设定数据

序号	标识符	名称
进给轴/主轴专用		
43600	IPOBRAKE_BLOCK_EXCHANGE	程序段切换条件
43610	ADISPOSA_VALUE	制动斜坡公差窗口

24.10.3 接口信号

序号	位	名称
通道专用		
DB3200.DBX0006	.0	进给禁用
DB3200.DBX0007	.1	NC 启动
DB3200.DBX0007	.4	NC 停止进给轴和主轴
DB3200.DBX0007	.7	复位
DB3300.DBX0004	.3	所有轴停止
DB3300.DBX1000	.6	运行指令 -
DB3300.DBX1000	.7	运行指令 +
进给轴/主轴专用		
DB380x.DBB0000	-	轴专用进给倍率
DB380x.DBX0002	.1	控制器使能
DB380x.DBX0002	.2	剩余行程/主轴复位
DB380x.DBX5004	.1	复位
DB380x.DBX5004	.2	继续
DB380x.DBX5004	.6	沿制动斜坡停止
DB380x.DBX5004	.7	PLC 控制轴
DB390x.DBX0000	.6	粗准停
DB390x.DBX0000	.7	精准停
DB390x.DBX0001	.1	轴报警
DB390x.DBX0001	.2	轴运行就绪（AX_IS_READY）
DB390x.DBX0003	.0	复位已执行
DB390x.DBX0003	.1	PLC 控制轴
DB390x.DBX0003	.2	进给轴停止生效
DB390x.DBX0003	.3	进给轴/主轴禁用生效
DB390x.DBX0004	.6	运行指令 -
DB390x.DBX0004	.7	运行指令 +
DB390x.DBX1002	.5	定位轴
DB390x.DBX5002	.7	紧急退回生效

25 主轴同步（S3）

25.1 简要说明

25.1.1 功能

“主轴同步”功能可用于两个主轴间的位置同步或速度同步。其中一个主轴定义为引导主轴（LS），另外一个主轴为联动主轴（CS）。

转速同步：	$n_{CS} = k_{ü} * n_{LS}$	其中 $k_{ü}$ = 1, 2, 3, ...
位置同步：	$\varphi_{CS} = \varphi_{LS} + \Delta\varphi$	其中 $0° \le \Delta\varphi < 360°$

说明

“通用耦合‘CP-BASIC’”选件

激活“通用耦合‘CP-BASIC’”选件后即启用“主轴同步”功能。该可选功能仅适用于车削数控系统。

存储器配置

动态 NC 空间中预留用于“主轴同步”功能的存储器空间通过机床数据定义：

MD18450 $MN_MM_NUM_CP_MODULES（CP 模块的最大数量）

MD18452 $MN_MM_NUM_CP_MODULES（CP 主值的最大数量）

建议在调试过程中就设定好在机床处于最高配置时可预见的最大数值。

应用

背面加工

其中一种应用是在双主轴车床上进行背面加工，即：在引导主轴和联动主轴位置同步后，将工件从引导主轴快速传送至联动主轴（或相反），且引导主轴和联动主轴无须处于静止状态。

多面体加工（多面体车削）

“主轴同步”功能中，可设置引导主轴和联动主轴之间的整数传动比：$k_{ü}$，以此来实现多面体加工。

联动轴数量

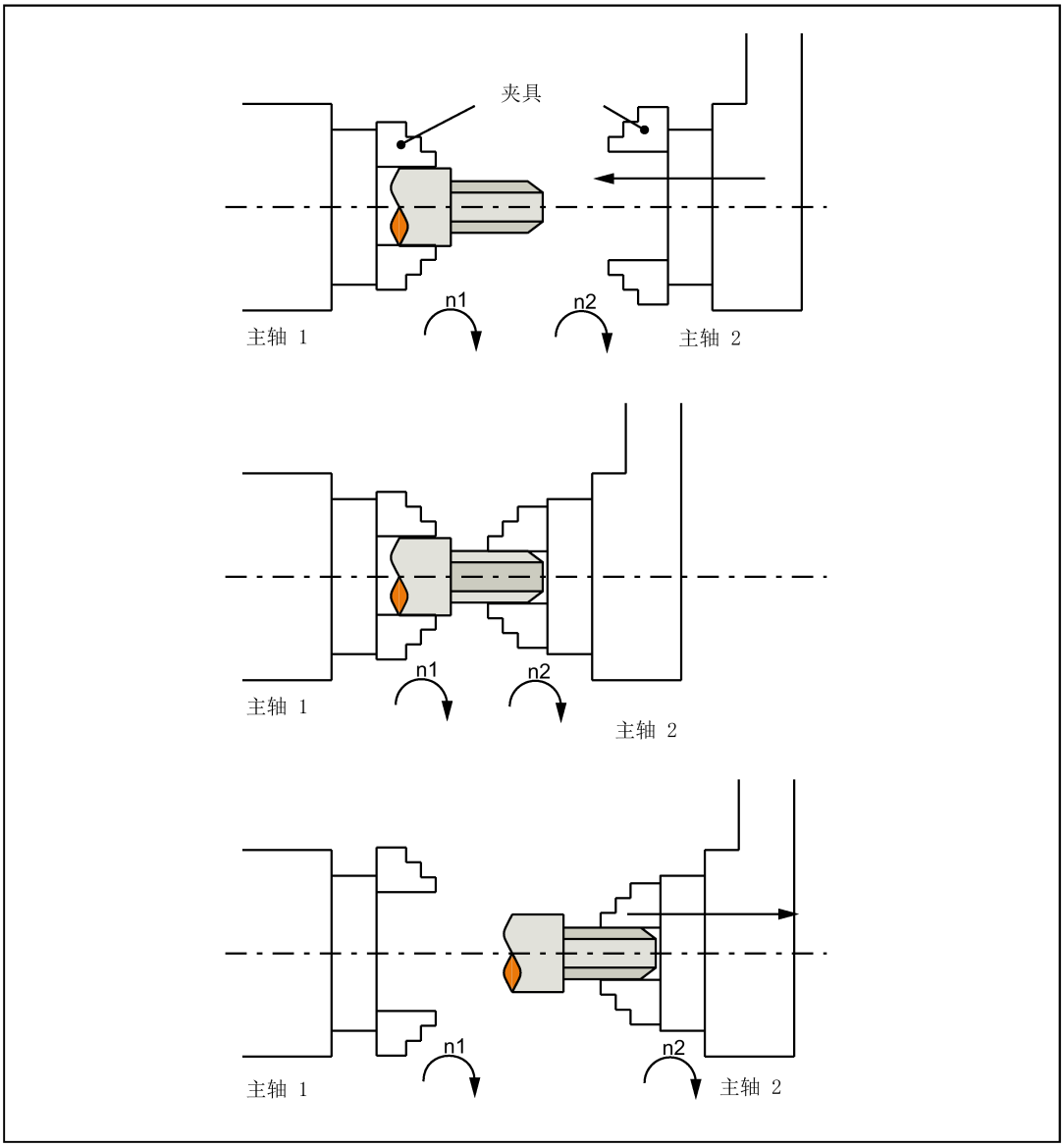
对于 SINUMERIK 808D ADVANCED，NC 通道中仅可同时运行一对同步主轴对。

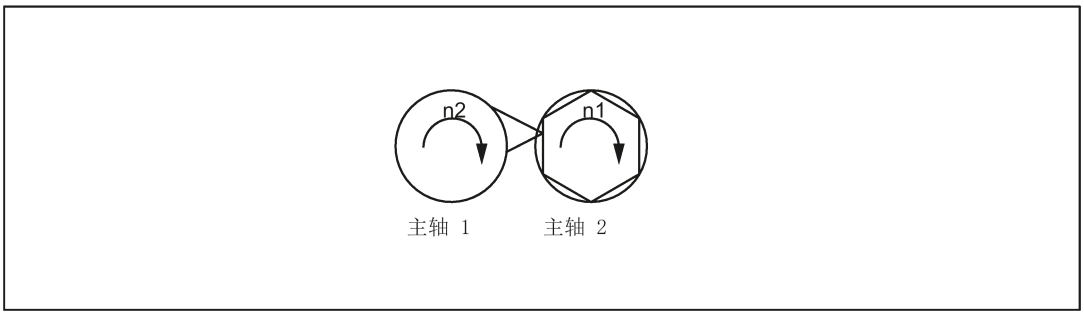
定义

既可以通过通道专用的机床数据来固定设置同步主轴对的引导主轴与联动主轴，也可以通过零件程序指令来灵活定义同步主轴对。

激活/撤销

主轴同步功能是通过零件程序指令来激活/撤销的。





25.1.2 主轴同步

说明

<轴表达式>：	可以是： - 轴名称 - 主轴名称
<轴名称>：	C（主轴在进给轴方式中的名称为“C”。）

<主轴名称>：	Sn, SPI(n) 其中 n = 主轴号
<主轴号>：	1, 2, ...，对应 MD35000 $ MA_SPIND_ASSIGN_TO_MACHAX 中定义的主轴号
（从动轴，引导轴，偏移）：	偏移 = 系统变量读取的联动主轴的可编程偏移
$P_COUP_OFFS[Sn]	程序中编写的同步主轴位置偏移

同步主轴对

在主轴同步中有一个联动主轴和一个引导主轴，即**同步主轴对**。在激活耦合时，联动主轴根据确定的函数关系跟随引导主轴。

主轴同步

主轴同步（也称为同步主轴方式）是另一种主轴运行方式。激活主轴同步前，联动主轴必须转换到位置闭环控制方式下。激活耦合后，联动主轴和引导主轴同步。撤销耦合后，联动主轴进入开环控制方式。

一旦联动主轴的主轴同步激活，NC 便会将以下接口信号报告给 PLC：

IS "主轴同步" (DB390x.DBX2002.4) = 1。

同步主轴数量

对于 SINUMERIK 808D ADVANCED，NC 通道中仅可同时运行一对同步主轴对。

主轴同步方式

主轴同步具有以下功能：

- 联动主轴（CS）和引导主轴（LS）以相同的转速旋转
  （$n_{CS} = n_{LS}$；传动比 $k_{ü} = 1$）
- 联动主轴和引导主轴可以以相同的方向或相反的方向旋转
  （$k_{ü}$ 可确定为正或负）
- 联动主轴（CS）和引导主轴（LS）以不同的转速旋转
  （$n_{CS} = k_{ü} \cdot n_{LS}$；传动比 $k_{ü} \neq 1$）
  应用：多面体车削
- 联动主轴和引导主轴之间的角度可定义（$\varphi_{CS} = \varphi_{LS} + \Delta\varphi$）
  联动主轴和引导主轴以转速同步的方式运行，同时存在指定的角度偏移（位置同步）。
  应用：异型工件
- 既可以在运行状态，也可以在静止状态下激活联动主轴和引导主轴的同步运行。
- 引导主轴可以使用开环控制方式和位置闭环控制的全部功能。
- 主轴同步未激活时，联动主轴和引导主轴可以在任意一种主轴运行方式中运行。
- 在主轴运动时激活主轴同步后也可以改变传动比。
- 可在主轴同步耦合激活后更改联动主轴和引导主轴的偏移量（叠加运动）。

耦合方式

主轴同步耦合可以定义为

- 通过通道专用的机床数据固定定义
  （以下称为**"固定定义的耦合"**）
- 通过零件程序中的语言指令（COUP...）灵活定义
  （以下称为**"用户定义的耦合"**）

.这两途径有以下组合方式：

1. 一个耦合可由机床数据固定定义。第二个耦合可由零件程序灵活定义。
2. 没有通过机床数据定义的耦合。对此，可通过零件程序对耦合进行设置和用户定义。

独立的联动主轴插补器

通过独立的**联动主轴插补器**可以让来自不同通道或另一个 NCU 的多个联动主轴按照用户自定义的方式与唯一的一个引导主轴同步运行。联动主轴插补器可通过

- COUPON 或 COUPONC 激活
- COUPOF 或 COUPOFS 撤销

联动主轴插补器始终位于用 COUPON 和 COUPONC 指令编程联动主轴的通道中。

通过耦合专用的轴 VDI 接口信号，PLC 可以控制一些主轴同步功能。但这些信号只对联动主轴有效，对引导主轴无效（参见章节“通过 PLC 控制同步主轴耦合 (页 229)”）。

定义主轴同步

激活主轴同步之前必须确定需要同步的主轴（联动主轴、引导主轴）。

这可根据应用情况的不同按两种不同的方式进行：

1. 固定定义的耦合：
   需要作为联动主轴和引导主轴生效的机床轴由通道专用的机床数据 MD 21300 $MC_COUPLE_AXIS_1[n] 确定。
   这种耦合配置中确定的引导主轴和联动主轴无法由 NC 零件程序修改。
   耦合参数也可由 NC 零件程序修改。
2. 用户定义的耦合：
   使用语言指令“COUPDEF（联动主轴、引导主轴...）”可以在 NC 零件程序中新建和修改耦合。如果要定义一个新的耦合关系，必须在此之前删除现有的用户定义的耦合，即使用语言指令 COUPDEF（联动主轴、引导主轴）。
   采用引导主轴和联动主轴时，在每个 COUP...语言指令中均需要编写引导主轴和联动主轴的主轴名称 (Sn, SPI(n))。由此可以明确确定同步主轴耦合。
   有效的主轴号必须事先在以下轴专用机床数据中指定了对应的机床轴。
   MD35000 $MA_SPIND_ASSIGN_TO_MACHAX.

NC 会通过对应机床轴的接口信号“联动主轴生效”（DB390x.DBX5003.1）和接口信号“引导主轴生效”（DB390x.DBX5003.0）报告 PLC 这些机床轴是作为引导主轴还是联动主轴。

可选择通过零件程序、PLC 或同步动作对引导主轴进行编程。

传动比

传动比用分子和分母（传动参数）来规定。这样便可以精确规定有理数形式的传动比。

通常情况下下列关系成立：

$k_{Ü}$ = 分子：分母 = $Ü_{分子}$：$Ü_{分母}$

在控制系统内部，传动参数的值范围（$Ü_{分子}$，$Ü_{分母}$）几乎不受限制。

通过机床数据配置的耦合的传动参数可有通道专用的 SD42300 定义：COUPLE_RATIO_1[n]。还可通过语言指令 COUPDEF（联动主轴、引导主轴、 $Ü_{分子}$、$Ü_{分母}$...）来修改传动比。设定数据中的值不可以改写（缺省值）。

对于通过 NC 零件程序定义的同步，只可通过语言指令 COUPDEF (...) 来设置传动比。

新的传动参数会在 COUPDEF 指令执行完毕后立即生效。

更多关于主轴同步耦合的编程指令，请参见章节“编程 (页 232)”。

耦合特性

可为每个主轴同步耦合对设置以下特性：

- **程序段切换特性**
  激活主轴同步时，或在激活耦合的条件下修改传动比或指定的角度偏移时，可确定何时切换到下个程序段。
  - 立即进行程序段切换
  - 在满足“精同步”条件时进行程序段切换
  - 在满足“粗同步”条件时进行程序段切换
  - 在满足 IPOSTOP 条件时进行程序段切换（即：在设定值同步后）
  - 使用 WAITC 等待标记可在任意时间点检查同步条件。

- 联动主轴和引导主轴的**耦合方式**
  既可使用引导主轴的位置设定值作为联动主轴的控制量，也可使用引导主轴的位置实际值作为联动主轴的控制量。因此可选择以下耦合方式：
  - 设定值耦合（DV：设定值）
    在位置闭环模式中使用。此时，两根主轴的动态响应应几乎一致。应优先使用这种耦合方式。
  - 实际值耦合（AV：实际值）
    在没有引导主轴位置闭环控制，或联动主轴与引导主轴的控制方式存在着巨大偏差时使用。联动主轴的设定值从引导主轴的实际值得出。当引导主轴的转速不断变化时，实际值耦合的效果比设定值耦合的效果差。
  - 速度耦合（VV：速度值）
    速度耦合位于设定值耦合内部。它对联动主轴和引导主轴的要求较低。联动主轴和引导主轴无需位置闭环控制和测量系统。
    联动主轴和引导主轴的位置偏移未经定义。

在**固定定义的耦合**中是通过机床数据来定义同步特性的（参见章节“定义 (页 236)”），而在**用户定义的耦合**中是通过语言指令 COUPDEF 来定义耦合特性的（参见章节“定义（COUPDEF） (页 232)”）。

此外，在固定定义的耦合中，“耦合方式”和“程序段切换特性”还可通过语言指令 COUPDEF 修改。

耦合特性设有写保护

使用通道专用的 MD21340 $MC_COUPLE_IS_WRITE_PROT_1 可以确定固定定义的耦合参数（传动比、耦合方式以及程序段切换特性）是否受 NC 零件程序的影响：

0:耦合参数可由 NC 零件程序指令 COUPDEF 加以修改。

1:耦合参数无法由 NC 零件程序修改。一旦尝试，系统会拒绝此次修改并发出报警信息。

叠加运动

主轴同步激活后，联动主轴会按照指定的传动比跟踪引导主轴。

同时，联动主轴可以叠加运动，以修改定义的联动主轴和引导主轴之间的角度位置。

联动主轴的叠加运动可以以不同的形式触发：

- AUTO 和 MDA 模式下，联动主轴的可编程位置偏移：
  - 使用语言指令 COUPON 和 SPOS 激活主轴同步后，可以修改联动主轴和引导主轴之间的相对位置（参见章节“通过零件程序来激活主轴同步 (页 227)”）。
- 联动主轴的手动位置偏移：
  - 在 JOG 模式下（连续 JOG 或断续 JOG）
    主轴同步激活时，使用手轮或正/负移动键给出了联动主轴的叠加运动量。
  - 在 AUTO 和 MDA 模式下
    使用手轮通过 DRF 偏移给出联动主轴的叠加运动量

一旦联动主轴执行叠加运动，接口信号“叠加运动”（DB390x.DBX5002.4）会置为 1 信号。

在编写 COUPON 指令时，联动主轴会以可能的最大速度进行叠加运动。在用 SPOS 修改偏移时，可以用 FA[Sn] 来指定定位速度，并借助倍率（可通过接口信号“主轴进给倍率生效” DB380x.DBX2001.0 选择）对其进行调整。

更多关于如何通过 FA[Sn] 来指定定位速度的信息，请参见章节“主轴（S1） (页 158)”。

设定值补偿

由系统变量 $AA_COUP_CORR[Sn] 读出的设定值偏移如同一个位移值一样，会对联动主轴后续的所有编程指令生效，相当于 MCS 中的 DRF 偏移。

示例：补偿值测定

如果通过 COUPON(....,77) 指令写入了一个 7°的耦合位移，而且由于工件夹具闭合而产生了 81°的机械位置偏移，系统会得出 4°的补偿值。

系统变量为联动主轴提供以下值：

$P_COUP_OFFS[S2] ; 程序中编写的位置偏移 = 77°

$AA_COUP_OFFS[S2] ; 设定位置偏移 = 77°

$VA_COUP_OFFS[S2] ; 实际位置偏移约 77°

$AA_COUP_CORR[S2] ; 补偿值 = 4°

25.1.3 主轴同步的前提条件

激活主轴同步的条件

只有具备以下条件时才可激活同步主轴耦合，否则系统会发出报警信息。

- 必须事先确定好同步主轴耦合（要么通过机床数据进行固定定义，要么借助 COUPDEF 指令通过零件程序进行用户定义）。
- 激活耦合的 NC 通道必须知道哪些主轴需要进行耦合。
  通道专用 MD20070 $MC_AXCONF_MACHAX_USED
  轴专用 MD35000 $MA_SPIND_ASSIGN_TO_MACHAX
- 激活耦合的 NC 通道中必须分到了联动主轴。
  轴专用 MD30550 AXCONF_ASSIGN_MASTER_CHAN 的缺省值
- 设定值耦合和实际值耦合（DV、AV）需要满足的特定条件：
  引导主轴和联动主轴必须至少具备一个位置测量系统进行位置采集，且位置闭环控制必须事先调试完毕。

说明

位置闭环控制激活时，引导主轴的最大设定转速会被限制在最大转速的 90%（控制裕量）以内。NC 会通过接口信号“设定转速受限”（DB390x.DBX2001.1）将这一状态报告给 PLC。

更多信息请参见章节“主轴（S1）(页 158)”。

- 设定值耦合（DV）需要满足的特定条件：
  为确保同步特性更加精确，在激活耦合之前，应编写指令 SPCON 使引导主轴进入位置闭环控制。
- 激活主轴同步之前必须选择合适的引导主轴和联动主轴变速档。主轴同步中无法切换引导主轴和联动主轴变速档，因此也无法进行往复运动。当你要求执行上述操作时，系统会发出报警。

25.1.4 通过零件程序来激活主轴同步

激活 COUPON、COUPONC 耦合

使用语言指令 COUPON 可在零件程序中激活指定主轴之间的耦合从而激活主轴同步，耦合采用上一次的有效参数。耦合可固定定义或用户定义。此时的引导主轴和/或联动主轴既可处于静止状态，也可处于运行状态。

必须具备以下条件才可以激活主轴同步（参见章节“主轴同步的前提条件 (页 227)”）。

语言指令 COUPONC 可使主轴继续采用零件程序中之前写入的方向和转速。但它不能写入角度偏移。

COUPON 编程方式

激活主轴同步有两种方式可选：

1. COUPON 指令可以尽快的激活耦合，但不设定引导主轴（LS）和联动主轴（CS）之间的**角度偏移**。
   COUPON（联动主轴、引导主轴）
2. 使用 COUPON 指令激活耦合，同时引导主轴和联动主轴之间存在**指定的角度偏移** POS_{CS}。选择这种方式时还须写入角度偏移。
   COUPON（联动主轴、引导主轴、POS_{CS}）

程序段切换特性

激活主轴同步之前，应确定使用哪种程序段切换条件（参见章节“定义（COUPDEF）(页 232)”）。

确定当前的耦合状态

通过轴系统变量$AA_COUP_ACT[<轴表达式>]可以在零件程序中确定进给轴/主轴的当前耦合状态（参见章节“主轴同步的轴系统变量 (页 235)”）。一旦联动主轴的耦合激活，位 2 就置 “1”。

修改指定的角度偏移

使用语言指令 COUPON 和 SPOS 可在主轴同步激活时修改指定的角度偏移。作为叠加运动，联动主轴定位到 POS_{CS} 写入的角度偏移位置。在此期间接口信号“叠加运动” (DB390x.DBX5002.4) 会置位。

角度偏移 POS_{CS}

指定的角度偏移 POS_{CS} 是个绝对位置，是在正向上和引导主轴零度位置相差的一个位置值。

位置控制主轴“0°位置”计算方式如下：

- 根据测量系统的零位或接近传感器信号和
- 根据用轴专用机床数据保存的参考值：
  MD34100 $MA_REFP_SET_POS、参考点值、
  在间距编码系统中无任何作用。
  MD34080 $MA_REFP_MOVE_DIST 参考点间距/目标点
  在间距编码系统中，
  MD34090 $MA_REFP_MOVE_DIST_CORR 参考点偏移/间距编码的绝对偏移。

POS_{CS} 的范围：0 ... 359.999°。

更多相关信息，请参见章节“回参考点（R1）(页 152)”。

读取当前的角度偏移

通过轴系统变量可以在 NC 零件系统中读取联动主轴与引导主轴的当前位置偏移。其中可分为：

- 联动主轴与引导主轴的设定值位置偏移
  $AA_COUP_OFFS [<联动主轴的名称>]
- 联动主轴与引导主轴的实际值位置偏移
  $VA_COUP_OFFS [<联动主轴的名称>]

（关于<轴名称>的说明，请参见章节“主轴同步 (页 223)”）

上电后激活

也可以在联动主轴或引导主轴还未回参考点/未同步时，即接口信号“已回参考点/已同步 1 或 2” (DB390x.DBX0.4 or DB390X.DBX0.5 =0) 复位时激活主轴同步。出现这种情况时系统会发出报警。

示例：

上电后，联动主轴与引导主轴已经通过一个工件固定在一起。

25.1.5 通过零件程序来撤销主轴同步

使用 COUPOF 或 COUPOFS 撤销耦合

使用语言指令 COUPOF 可以撤销指定主轴之间的同步。有三种方式可选。

使用指令 COUPOF 可以撤销指定主轴之间的同步，不论该耦合是固定定义还是用户定义的。此时的引导主轴和联动主轴既可处于静止状态，也可处于运行状态。

使用 COUPOF 指令撤销同步时，联动主轴会切换至**开环控制方式**。之前编写的转速值（S 字）不再对联动主轴有效，联动主轴可以像一个普通的主轴一样运行。

使用 COUPOF 撤销耦合时，系统会自动生成一个预处理停止指令，功能与 STOPRE 类似。

使用指令 COUPOFS 既可以无需达到指定位置就快速撤销耦合，也可以到了指定位置时再撤销耦合。

COUPOF 和 COUPOFS 方式

使用 COUPOF 和 COUPOFS 撤销同步有三种方式可选：

1. 尽快撤销耦合
   立即使能程序段切换。
   COUPOF（联动主轴、引导主轴）和 COUPOFS（联动主轴、引导主轴）

2. 等联动主轴越过指定位置 POS_{CS} 时再撤销耦合（联动主轴 = CS；引导主轴 = LS）。
   然后使能程序段切换。
   COUPOFS（联动主轴、引导主轴、POS_{CS}）
3. 等联动主轴和引导主轴都越过指定位置 POS_{CS} 和 POS_{LS} 时再撤销耦合。
   然后使能程序段切换。
   COUPOFS（联动主轴、引导主轴、POS_{CS}、POS_{LS}）

POS_{CS}、POS_{LS}

POS_{CS} 和 POS_{LS} 是联动主轴和引导主轴相对于参考点的实际位置。

POS_{CS}、POS_{LS}0 ... 359,999°。

更多信息请参见章节“回参考点（R1）(页 152)”。

在运行过程中使用 COUPOF 指令

使用 COUPOF 指令可以在运行过程中撤销同步，而联动主轴会以当前的转速继续旋转（n_{CS}）。通过系统变量 $AA_S 可以在零件程序中读取转速。

撤销同步后，从零件程序中使用 M05、SPOS、SPOSA 或相应的 PLC 接口信号可以使联动主轴停止。

COUPOFS 自动停止联动主轴

该指令增加了一个联动主轴停止的特性：

- 无需达到指定位置即尽快撤销耦合，并停止联动主轴。
  然后使能程序段切换。
  COUPOFS（联动主轴、引导主轴）
- 尽快的撤销耦合，同时在指定的位置上停住联动主轴。然后使能程序段切换。

条件：

耦合未激活时，COUPOFS（联动主轴、引导主轴）和 COUPOFS（联动主轴、引导主轴、POS_{CS}）指令没有任何意义。

25.1.6 通过 PLC 控制同步主轴耦合

联动主轴由 PLC 控制

通过耦合专用的轴 VDI 接口信号可以从 PLC 程序中控制联动主轴的同步运动。其中 PLC 可以禁止、抑制或重新激通过编写了偏移的联动主轴同步动作。

这些信号对引导主轴没有任何的作用。可以使用以下耦合专用的 VDI 信号（PLC → NC）：

接口信号**“禁用同步” (DB380x.DBX5007.5)**

“禁用同步”

使用轴接口信号“禁用同步” (DB380x.DBX5007.5) 可以抑制联动主轴的同步动作。

当包含 COUPON（联动主轴、引导主轴、偏移）的零件程序指令进入主运行时，NC 会分析联动主轴的以下接口信号：

接口信号“禁用同步” (DB380x.DBX5007.5)。

- 当接口信号“禁用同步” (DB380x.DBX5007.5) = 0 时，位置偏移如同之前一样生效。
- 当接口信号“禁用同步” (DB380x.DBX5007.5) = 1 时，同步只以恒速进行。联动主轴不会有额外的运动。
  随后的耦合与编写指令 COUPON（<联动主轴>、<引导主轴>）类似。

特点

接口信号“禁用同步” (DB380x.DBX5007.5) 不影响通过以下方式生成的联动主轴的偏移运动：

- SPOS、POS
- 同步动作
- 手动

这些功能可通过 VDI 接口信号“进给停止/主轴停止” (DB380x.DBX4.3) 来控制。

达到同步

一旦达到同步，以下两个 VDI 信号都会置位，不管同步是否禁止：

接口信号“粗同步” (DB390x.DBX5002.1) 和

接口信号“精同步” (DB390x.DBX5002.0)

虽然同步被抑制，但 COUPON 这个指令后续的程序段切换不会被阻止。

示例

COUPON 指令后续程序段切换的特性

程序代码	注释
	; 接口信号"禁用同步"
	; (DB380x.DBX5007.5) = 1，针对 S2 置位
N51 SPOS=10 SPOS[2]=10	; 对应 0° 的偏移
	; 的定位
N52 COUPDEF(S2,S1,1,1,"FINE","DV")	
N53 COUPON(S2,S1,77)	; 保留 0 度的实际偏移
	; 联动主轴不运动
	; VDI 信号
	; 接口信号"粗同步"
	; (DB390x.DBX5002.1) 和
	; 接口信号"精同步"
	; (DB390x.DBX5002.0)
	; 置位，且程序段切换
	; 使能。
N54 M0	
N57 COUPOF(S2,S1)	
N99 M30	

复位及重新激活

复位接口信号“禁用同步” (DB380x.DBX5007.5) 不会对联动主轴偏移产生影响。如果之前通过该 VDI 接口信号抑制了联动主轴的偏移运行，那么该信号复位后，偏移并不会自动生效。

同步恢复如下：

- 在接口信号“禁用同步” (DB380x.DBX5007.5) = 0 时，再次写入零件程序指令 COUPON（联动主轴、引导主轴、偏移）。
  比如可以将 COUPON（联动主轴、引导主轴、偏移）写入 ASUB 中。
- 将接口信号“重新同步” (DB380x.DBX5007.4) 置 1

读取偏移

通过以下系统变量可以从零件程序和同步动作中读取三个不同的联动主轴位置偏移值。变量 $P_COUP_OFFS[Sn] 只可以在零件程序中使用。

描述	NC 变量
程序中编写的同步主轴位置偏移	$P_COUP_OFFS[Sn]
程序中编写的设定值位置偏移	$AA_COUP_OFFS[Sn]
程序中编写的实际值位置偏移	$VA_COUP_OFFS[Sn]

“进给停止/主轴停止”

通过定义 MD30455 MISC_FUNCTION_MASK 中的位 4 可以确定联动主轴接口信号 **“进给停止/主轴停止” (DB380x.DBX4.3)** 的特性。

位 4 = 0 兼容方法：

通过撤销联动主轴的进给使能使耦合组制动。

位 4 = 1：

进给使能只作用于插补分量（SPOS,..），对耦合没有任何影响。

25.1.7 监控主轴同步

精/粗同步

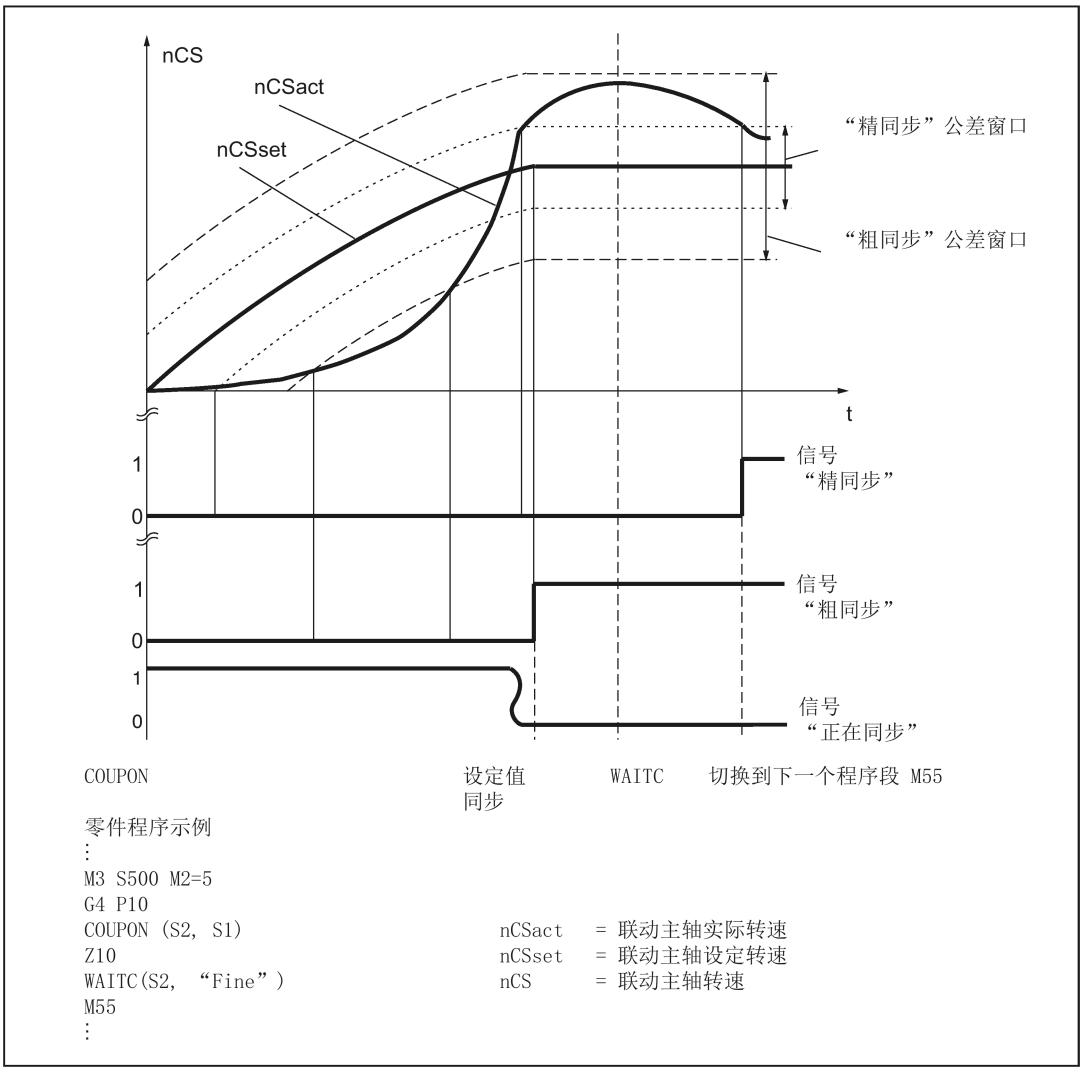
除了常见的主轴监控方式外，还可以在主轴同步中对联动主轴与引导主轴的同步过程进行监控。

在监控中，NC 可通过接口信号“精同步” (DB390x.DBX5002.0) 或接口信号“粗同步” (DB390x.DBX5002.1) 报告 PLC：联动主轴的当前实际位置 (AV、DV) 或实际速度 (VV) 是否在指定公差窗口内。

耦合激活时，“精同步”和“粗同步”信号会在达到设定值同步状态后自动更新。

公差窗口的大小可由联动主轴的机床数据来设置。达到同步运行的过程可由以下系数来控制：

- AV（实际值）、DV（设定值）：联动主轴和引导主轴的位置差
- VV（速度值）：联动主轴和引导主轴的速度差



阈值

联动主轴和引导主轴的位置公差范围和速度公差范围的单位为度或 rpm。

- “粗同步”的阈值
  轴专用 MD37200：AV（实际值）、DV（设定值）：COUPLE_POS_TOL_COARSE
  MD37220：VV（速度值）：COUPLE_VELO_TOL_COARSE

- “精同步”的阈值
  轴专用 MD37210：AV（实际值）、DV（设定值）：COUPLE_POS_TOL_FINE
  MD37230：VV（速度值）：COUPLE_VELO_TOL_FINE

最大转速/加速度

主轴同步中，引导主轴的最大转速和加速度会自动调整，使得在当前的变速档的传动比条件下，联动主轴会跟随引导主轴旋转，且不会超出自身的限值。

比如引导主轴会自动减速，使得联动主轴不会超出自身的最大转速，从而确保同步生效。

25.2 编程

带引导主轴编程的耦合指令

指令	功能
COUPDEF（联动主轴、引导主轴）	定义耦合或修改已定义的耦合
COUPON（联动主轴、引导主轴、POSCS）	激活耦合
COUPONC（联动主轴、引导主轴）	激活耦合并接收当前生效的联动主轴转速
COUPOF（联动主轴、引导主轴）	撤销耦合
COUPOFS（联动主轴、引导主轴、POSCS）	撤销耦合并停止联动主轴
COUPDEL（联动主轴、引导主轴）	删除耦合
COUPRES（联动主轴、引导主轴）	重新激活定义的耦合数据

不带引导主轴编程的耦合指令

指令	功能
COUPOF（联动主轴、引导主轴）	撤销耦合
COUPOFS（联动主轴、POSCS）	撤销耦合并停止联动主轴
COUPDEL（联动主轴）	删除耦合
COUPRES（联动主轴）	重新激活定义的耦合数据

25.2.1 定义（COUPDEF）

可写入的耦合

耦合的数量可根据可用的轴数多次写入。这个数量是由轴/主轴的数量减去一得出。也可以和以前一样通过机床数据来定义耦合。

固定定义的耦合

在固定定义的同步主轴耦合中，通过 NC 零件程序可以修改耦合特性（前提是未设置写保护）和传动比。联动主轴和引导主轴的机床数据不可以更改。

定义新的耦合

使用语言指令 “COUPDEF”可以创建新的耦合（用户定义），并将现有的耦合修改掉。

完整的耦合参数的编程方式为：

COUPDEF（联动主轴、引导主轴、$T_{分子}$、$T_{分母}$、程序段切换特性、耦合方式）

通过“联动主轴”和“引导主轴”可以明确确定采用的是哪个同步主轴耦合。

仅当需要改变其他耦合参数时，才必须编写这些耦合参数。其中未编写的参数采用上次的值。

下文具体介绍每个耦合参数：

- **联动主轴、引导主轴**：联动主轴名称和引导主轴名称
  例如：S1, SPI(1), S2, SPI(2)
  有效的主轴号必须事先在以下轴专用的 MD35000 $MA_SPIND_ASSIGN_TO_MACHAX 中指定了对应的机床轴。
- $T_{分子}$, $T_{分母}$：传动比分子和分母
  传动比用分子和分母（传动参数）来规定（参见章节“主轴同步 (页 223)”）。

其中至少必须规定分子。如果没有规定分母，通常系统假设它为“1.0”。

- **程序段切换特性**
  它可用于激活主轴同步时选择何时进行程序段切换：

NOC：立即使能程序段切换

FINE：在满足“精同步”条件时进行程序段切换

COARSE：在满足“粗同步”条件时进行程序段切换

IPOSTOP：在满足 IPOSTOP 条件时进行程序段切换（即：在设定值同步后）

程序段切换特性以字符串的形式指定（即：带引号）。

程序段转换特性用粗体字母指定即可。其余的字母虽然没有什么含义，却可以使零件程序更加清晰易懂。

如果未指定程序段切换特性，那么在这个时间点生效的特性将继续保持有效。

使用可编程的 **WAITC** 等待标记可将切换新的程序段延迟到达到指定的同步过程时再进行。

- **耦合方式**

DV（设定值）：联动主轴和引导主轴的设定值耦合

AV（实际值）：联动主轴和引导主轴的实际值耦合

VV（速度值）：联动主轴和引导主轴的速度耦合

如果未指定耦合方式，那么在之前的耦合方式继续保持有效。

说明

仅当关闭主轴同步后才能更改耦合方式。

示例

COUPDEF (SPI(2), SPI(1), 1.0, 1.0, "FINE", "DV")

COUPDEF (S2, S1, 1.0, 4.0)

COUPDEF (S2, SPI(1), 1.0)

默认设置

用户定义的耦合有以下缺省设置：

- Ü $_{分子}$=1.0
- Ü $_{分母}$ =1.0
- 程序段切换特性 = **IPOSTOP**（在设定值达到同步后进行程序段切换）
- 耦合方式 = **DV**（设定值耦合）

删除耦合

使用语言指令“COUPDEL”可以删除用户定义的耦合。

COUPDEL（联动主轴、引导主轴）

说明

使用 COUPDEL 指令可撤销已激活的耦合并删除耦合数据。此时，报警 16797 无效。
联动主轴继续采用以前的转速旋转。该特性与 COUPOF（联动主轴、引导主轴）指令的特性相一致。

激活初始耦合参数

通过语言指令“COUPRES”可以再次激活由机床数据定义的耦合参数。

COUPRES（联动主轴、引导主轴）

此时用 COUPDEF 指令更改的参数（包括传动比）丢失。

通过指令“COUPRES”可以激活在机床数据和设定数据中设置的参数（即由机床数据定义的耦合）以及缺省设置（即用户定义的耦合）。

可编程的程序段切换

使用语言指令“WAITC”可以在 NC 程序中标记位置。系统在此位置上等待指定联动主轴达到同步条件，并将切换程序段延迟到达到指定的同步状态后再进行（参见“监控主轴同步 (页 231)”）。

WAITC（联动主轴）

优点：工艺上可能会要求耦合激活后经过一段时间再达到同步状态。

说明

原则上，等待标记 WAITC 可始终写入在程序中。如果指定的主轴不是作为联动主轴生效的，那么该指令对该主轴无效。

在 WAITC 里没有指定同步条件，系统就会检查耦合功能中由程序定义的/机床数据定义的同步条件，起码会检查是否达到设定值同步条件。

示例：

```
WAITC(S2),
:
WAITC(S2, "Fine"),
:
WAITC(S2, ,S4, "Fine")
```

停止及程序段切换

如果在这个时间段内撤销了轴使能而导致引导主轴和联动主轴停止，当伺服驱动再次给出轴使能时，引导主轴和联动主轴会再次运行到**最后**的设定位置上。

可通过程序指令 COUPON 和 WAITC 来控制程序段切换。同时，可通过 COUPDEF 指令或 MD21320 $MC_COUPLE_BLOCK_CHANGE_CTRL_1 来确定程序段切换标准。

25.2.2 激活和撤销耦合（COUPON、COUPONC、COUPOF）

激活主轴同步

使用语言指令 COUPON 可激活耦合和主轴同步。

激活主轴同步有两种方式可选：

1. **COUPON（联动主轴、引导主轴）**
   使用 COUPON 指令可以尽快的激活主轴同步，但不设定引导主轴和联动主轴之间的角度偏移。
2. **COUPON（联动主轴、引导主轴、POS_{CS}）**
   激活主轴同步，同时引导主轴和联动主轴之间存在指定的角度偏移 POS_{CS}。该位置在正向上和引导主轴零度位置相差的一个位置值。根据设置使能程序段切换。POS_{CS} 的范围：0 ... 359.999°。
3. **COUPONC（联动主轴、引导主轴）**
   在用 COUPONC 指令激活时，主轴会继续采用之前编写的转速和方向（M3 S... 或 M4 S...）。此时引导主轴和联动主轴会以差速运行。该指令中不能写入偏移位置。

写入 COUPON（联动主轴、引导主轴、POS_{CS}）或在主轴同步激活时写入 SPOS，可以修改联动主轴与引导主轴之间的角度偏移。

撤销主轴同步

撤销主轴同步有三种方式可选：

- **COUPOF（联动主轴、引导主轴）**
  快速撤销主轴同步。立即使能程序段切换。

如果写入了连续路径运行（G64），控制系统内部会自动生成一个程序段停止。

示例：

```
COUPDEF (S2, S1, 1.0, 1.0, "FINE, "DV")
:
COUPON (S2, S1, 150)
```

:
COUPOF (S2, S1, 0)
:
COUPDEL (S2, S1)

1. **COUPOFS (联动主轴、引导主轴)**
   撤销耦合并停止联动主轴。立即进行程序段切换。
2. **COUPOFS (联动主轴、引导主轴、POS$_{CS}$)**
   联动主轴越过指定位置 (以机床坐标系为参照) 后才使能程序段切换。
   取值范围 0 ... 359.999°。

25.2.3 主轴同步的轴系统变量

确定当前的耦合状态

联动主轴的当前耦合状态可在 NC 件程序中通过下列轴系统变量读取：

$AA_COUP_ACT[<轴表达式>]

关于<轴表达式>的说明参见章节“主轴同步 (页 223)”。

示例：

$AA_COUP_ACT[S2]

读取的值对于联动主轴有以下含义：

值 = 0： 无耦合激活
值 = 4： 同步主轴耦合激活

读取当前的角度偏移

联动主轴的当前耦合状态可在 NC 件程序中通过下列轴系统变量读取：

- 联动主轴与引导主轴的设定值位置偏移：
  $AA_COUP_OFFS[<轴表达式>]
- 联动主轴与引导主轴的实际位置偏移：
  $VA_COUP_OFFS[<轴表达式>]

示例：

$AA_COUP_OFFS[S2]

如果在 COUPON 中写入了一个角度偏移，在达到设定值同步后，读取的值和该角度偏移一致。

读取写入的角度偏移

程序最后写入的联动主轴与引导主轴的位置偏移在 NC 件程序中可以用下列轴系统变量读取：

$P_COUP_OFFS[<轴表达式>]

说明

如果在主轴同步激活后撤销伺服使能，主轴会进入跟踪模式，此后再次给出伺服使能后，会产生一个新的位置偏移值，它和之前的程序写入值不同。

$P_COUP_OFFS 只读取最初的程序写入值。当前值由$AA_COUP_OFFS 和 $VA_COUP_OFFS 读取。使用接口信号 NST DB380x.DBX5007.4 (重新同步) 可再次恢复最初写入的偏移。

25.2.4 自动激活和撤销位置闭环控制

转速开环模式下的特性

在 DV (设定值) 耦合方式中，程序指令 COUPON、COUPONC 和 COUPOF、COUPOFS 会在必要情况下自动激活或撤销引导主轴的位置闭环控制。如果在一根引导主轴上有多根联动主轴，**第一个** DV (设定值) 同步会**激活**引导主轴的位置闭环控制，**最后一个** DV (设定值) 同步会**撤销**引导主轴的位置闭环控制，条件是没有编写 SPCON 指令。

引导主轴和联动主轴不一定要在同一通道中。

在 COUPON 和 COUPONC 指令下自动激活

耦合方式不同，COUPON 和 COUPONC 指令对位置闭环控制的影响也不同：

耦合方式	DV	AV	VV
联动主轴	开启位置控制	开启位置控制	无动作
引导主轴	开启位置闭环控制 [1]	无动作	无动作

[1] 在 DV（设定值）耦合方式下，如果**至少有一个**联动主轴与引导主轴耦合，那么位置闭环控制会在 COUPON 和 COUPONC 指令下自动激活。

在 COUPOF 和 COUPOFS 指令下自动撤销

耦合方式不同，COUPOF 和 COUPOFS 指令对位置闭环控制的影响也不同：

耦合方式	DV	AV	VV
联动主轴	关闭位置闭环控制 [2]	关闭位置闭环控制 [2]	无动作 [2]
引导主轴	关闭位置闭环控制 [3]	无动作	无动作

[2] COUPOF 和 COUPOFS 指令，不指定位置

激活联动主轴转速开环模式。通过带有停止位置的 COUPFS 指令启动位置模式。在写入了 SPCON 指令使联动主轴进入位置闭环控制主轴模式，或在 COUPFS 指令中指定了位置时，位置闭环控制**不会自动关闭**。

[3] 当引导主轴没有其他的联动主轴和它进行 DV（设定值）同步时，COUPOF 指令就会自动**关闭**位置闭环控制。当引导主轴位于定位模式**或**进给轴模式中或是因 SPCON 指令进入位置闭环控制主轴模式时，位置闭环控制**都会继续存在**。

25.3 定义

说明

每个通道都可以定义**一个**同步主轴耦合。

下表给出定义耦合所需的机床数据。

序号	名称：$MC_	功能
MD21300	COUPLE_AXIS_1[<n>]	同步主轴耦合的机床轴： • <n> = 0：联动主轴的机床轴编号 • <n> = 1：引导主轴的机床轴编号 机床轴编号对应：MD20070 $MC_AXCONF_MACHAX_USED（通道中的机床轴） 机床轴编号 == 0：未定义耦合。以下系统数据也变无效。 **注**：为同步主轴耦合对定义的机床轴**无法**通过程序指令修改。
MD21320	COUPLE_BLOCK_CHANGE_CTRL_1	激活主轴同步后使能程序段切换 1)： • 立即 • 达到“精同步” • 达到“粗同步” • 达到“设定值同步” **注：**只要没有设置写保护 1)，就可以通过指令 COUPDEF 修改程序段切换使能。
MD21310	COUPLING_MODE_1	耦合方式 1)： • 实际值耦合 • 设定值耦合 • 速度耦合 **注：**只有没有设置写保护 1)，就可以在耦合**撤销**后通过指令 COUPDEF 修改耦合方式。
MD21330	COUPLE_RESET_MODE_1	NC 启动、NC 停止和复位时同步主轴耦合的特性。
MD21340	COUPLE_IS_WRITE_PROT_1	耦合设有写保护
1) 参见 MD21340 $MC_COUPLE_IS_WRITE_PROT_1		

下表给出定义耦合所需的设定数据。

序号	名称：$SC_	功能
SD42300	COUPLE_RATIO_1[<n>]	转速传动比：联动主轴/引导主轴 = 分子/分母 1)： • <n> = 0：分子（联动主轴） • <n> = 1：分母（引导主轴） **注：**只要没有设置写保护 1)，就可以通过指令 COUPDEF 修改传动比。
1) 参见 MD21340 $MC_COUPLE_IS_WRITE_PROT_1		

25.3.1 NC 启动时同步主轴耦合的响应

NC 启动时同步主轴耦合的特性取决于以下机床数据的设置：

定义的同步主轴耦合

响应	MD21330 $MC_COUPLE_RESET_MODE_1
保持耦合	位 0 = 0
撤销耦合	位 0 = 1
激活定义的数据	位 5 = 1
激活耦合	位 9 = 1

编程的同步主轴耦合

响应	MD20112 $MC_START_MODE_MASK
保持耦合	位 10 = 0
撤销耦合	位 10 = 1

25.3.2 复位时同步主轴耦合的特性

复位/程序结束时主轴同步的特性取决于以下机床数据中的设置：

定义的同步主轴耦合

响应	MD21330 $MC_COUPLE_RESET_MODE_1	MD20110 $MC_RESET_MODE_MASK
保持耦合	位 1 = 0	位 0 = 1
撤销耦合	位 1 = 1	位 0 = 1
激活定义的数据	位 6 = 1	位 0 = 1

编程的同步主轴耦合

响应	MD20110 $MC_RESET_MODE_MASK
保持耦合	位 0 = 1，位 10 = 1
撤销耦合	位 0 = 1，位 10 = 0

25.4 特点

25.4.1 主轴同步的一般特点

动态响应

使用设定值耦合时，联动主轴和引导主轴的位置控制器参数应该一致（例如：K_V 系数）。必要时还要为转速闭环控制和主轴同步激活不同的参数组。与位置闭环控制、前馈控制和参数组不同的是，联动主轴的控制器参数可以通过 MD30455 $MA_MISC_FUNCTION_MASK 来设置，如同未经耦合时（参见章节“同步主轴耦合功能调试的特殊性 (页 245)”）。

前馈控制

为确保主轴同步中的引导主轴和联动主轴具有更好的动态响应性能，前馈控制**始终是激活**的。

通过轴专用的 MD32620 $MA_FFW_MODE 可以撤销联动主轴和引导主轴的前馈控制。当 MD32620 $MA_FFW_MODE 置零时，功能会受限。也就无法通过 SPCON 指令在运动过程中激活位置闭环控制。SPOS、M19 或 SPOSA 指令因此无效。也无法通过 NC 零件程序专用编写 FFWOF 指令来撤销联动主轴和引导主轴的前馈控制。

可以通过轴专用的 MD32620 $MA_FFW_MODE 来确定前馈控制方式（转速还是转矩前馈控制，参见章节“补偿（K3）(页 106)”。

最大转速和加速度

处于同步状态的主轴的最大转速和加速度取决于同步主轴对中“最慢”的主轴。此时系统会考虑当前的变速档、程序写入的加速度和引导主轴当前的位置闭环状态（激活/撤销）。

在控制系统内部，引导主轴的最大转速和加速度是由联动主轴的传动比和最大转速/加速度来决定的。

复合耦合

如果在激活主轴同步时发现了其中有一根联动主轴或引导主轴当前有生效的耦合关系，系统会忽略此激活指令并发出一个报警信息。

复合耦合示例：

- 主轴作为联动主轴用于多个引导主轴
- 耦合级联（一个联动主轴是下一个耦合对的引导主轴）

每个通道中可定义的主轴数量

通道中的每根轴都可定义为主轴。每个通道中的轴数量取决于控制性能。

由 ASUB 启动主轴同步

通过 PLC 启动 ASUB 后，可在“AUTO”或“MDI”模式下激活/撤销或结束同步。

更多信息请参见章节“运行方式，程序运行（K1）(页 81)”。

对报警的响应

如果在主轴同步期间出现了报警，而且该报警会在控制系统内部导致“撤销伺服使能”和“激活跟踪方式”，则控制系统的后续响应和由以下 NC/PLC 接口信号导致的响应一致：

- DB380x.DBX2.1 = 0（控制器使能）
- DB380x.DBX1.4 = 1（跟随模式）

（参见章节“主轴同步和 NC/PLC 接口信号 (页 240)”）

通过 NC/PLC 接口信号重新进行同步：

DB380x.DBX5007.4 = 0 → 1 （重新同步）

可再次恢复已写入的偏移（参见章节“恢复联动主轴的同步 (页 238)”）。

在激活的主轴同步中进行程序搜索

说明

我们建议在主轴同步激活时采用搜索类型 5 “通过程序测试进行程序段搜索”（SERUPRO）。

25.4.2 恢复联动主轴的同步

位置偏移的原因

撤销驱动使能后，在跟踪方式下重新激活耦合时，引导主轴和联动主轴之间可能会出现位置偏移。位置偏移的产生原因可能为：

- 一个零件被夹住或是两根主轴被手动旋转（加工区被打开、驱动未通电）。
- 撤销主轴使能后，两根主轴会因为惯性停转速度不同产生位置偏移。

- 已出现驱动报警（内部跟随模式）：
  DB390x.DBX1.3（跟随模式激活）= 1
  删除报警时不允许从 NC 中触发任何的同步动作。
- 由于联动主轴同步禁用，不可以执行同步动作：
  DB380x.DBX5007.5（禁用同步）

一般步骤

通过以下步骤可再次恢复联动主轴和引导主轴之间已经丢失的同步或还未执行的同步：

1. 设置轴使能，取消同步禁用，如果之前已设置同步禁用。
2. 通过 NC/PLC 接口信号启动联动主轴重新同步：
   DB380x.DBX5007.4（重新同步）
   只有在执行重新同步后才可以再次完整恢复设定值同步。
3. 等待主轴完成同步。

使能重新同步

设置重新同步使能后，联动主轴和引导主轴的实际位置进行重新同步。设置以下两个 NC/PLC 接口信号：

DB390x.DBX5002.1（粗同步）

DB390x.DBX5002.0（精同步）

重新同步须满足以下**要求**：

- 联动主轴获得使能。
- PLC 不能在联动主轴上设置同步禁用：
  DB380x.DBX5007.5（禁用同步）

重新同步联动主轴

一旦 NC 识别到以下接口信号出现一个上升沿，它就会为对应的联动主轴启动重新同步。

DB380x.DBX5007.4（重新同步）

NC 输出以下 NC/PLC 接口信号，报告发现了上升沿：

DB390x.DBX5003.4（正在进行同步）

当出现以下情况时，接口信号“重新同步”会复位：

- 联动主轴和引导主轴的设定值达到同步。
- NST DB380x.DBX5007.4（重新同步）已复位

联动主轴进行附加运动时，同步信号的响应

会计算出用于测定同步信号的叠加分量。

示例

程序代码	注释
N51 SPOS=0 SPOS[2]=90	
N52 OUPDEF(S2,S1,1,1,"FINE","DV")	
N53 COUPON(S2,S1,77)	
N54 M0	; 偏移 = 77°，出现“粗”/“精”同步信号。
N55 SPOS[2]=0 FA[S2]=3600	; 差速，报告“粗”/“精”同步信号
N56 M0	; （注意公差，见上）
	; 偏移 = 0°，出现“粗”/“精”同步信号。
N60 M2=3 S2=500	; 差速，报告“粗”/“精”同步信号
	; 偏移未确定，报告“粗”/“精”同步信号
N65 M0	; （注意公差，见上）

说明

通过撤销轴使能可以中断联动主轴的叠加运行（例如 SPOS）。该运动分量不受 “NC/PLC 接口信号” DB380x.DBX5007.4（重新同步）控制，而是通过 REPOS 过程修复。

附加条件

只有当联动主轴和引导主轴存在**定义的偏移**时，接口信号 DB380x.DBX5007.4（重新同步）才会生效。

偏移的生成方式有：包含位置偏移的 COUPON 指令（COUPON(...,77)）；SPOS、SPOSA、M19 指令生成的偏移。

25.4.3 主轴同步和 NC/PLC 接口信号

说明

在主轴同步中，始终都要考虑到引导主轴（LS）或联动主轴（CS）的各个接口信号对耦合的影响。

主轴倍率（DB380x.DBB2003）

在主轴同步中，只有引导主轴的主轴倍率值生效。

主轴禁用（DB380x.DBX1.3）

LS	CS	耦合	响应
0	0	灭	输出设定值
0	1	灭	不向联动主轴输出设定值
1	0	灭	不向引导主轴输出设定值
1	1	灭	不向引导主轴和联动主轴输出设定值
0	0	ON	输出设定值
0	1	ON	不禁用联动主轴
1	0	ON	禁用联动主轴，不输出设定值
1	1	ON	不向引导主轴和联动主轴输出设定值

控制器使能（DB380x.DBX2.1）

引导主轴：主轴同步过程中复位“控制器使能”

如果在主轴同步过程中复位控制器使能来激活**设定值**耦合，则对**实际值**耦合进行了控制内部切换。如果在引导主轴移动时复位控制器使能，引导主轴将会停止并触发报警。主轴同步依然保持。

引导主轴和联动主轴：没有控制器使能的情况下选择主轴同步

如果激活主轴同步后，未对引导主轴和联动主轴设置“控制器使能”，则控制器使能会被激活。但是，直到引导主轴**和**联动主轴设置控制器使能后引导主轴和联动主轴才会移动。

引导主轴和联动主轴：设置“控制器使能”

设置“控制器使能”后，主轴接接收哪个位置取决于 DB380x.DBX1.4 == <值>（跟随运行）：

<值>	接收的主轴位置
0	撤销控制器使能时的位置
1	当前位置

说明

对于程序段搜索过程中的主轴同步，建议始终对联动主轴**和**引导主轴都写入 DB380x.DBX2.1 接口信号（控制器使能）。如果不这样做，程序段搜索会在联动主轴加工之后停止，因为没有引导主轴的控制器使能。

说明

同步误差

如果不先撤销激活耦合就在主轴停止后取消联动主轴的 DB380x.DBX2.1 接口信号（控制器使能），即使再次激活“控制器使能”也无法补偿外部干预产生的同步误差。这会造成联动主轴与引导主轴之间的已编程的角度关系丢失。

重新同步可以恢复角度关系。DB380x.DBX5007.4 = 1（重新同步）

跟随模式（DB380x.DBX1.4）

跟随运行中会定时将设定位置设为当前实际位置：

DB380x.DBX1.4 == 1（跟随运行）**和** DB380x.DBX2.1 == 0（控制器使能）

⇒ 周期性：设定位置 = 实际位置

说明

DB380x.DBX1.4（跟随运行）只与 DB380x.DBX2.1 == 0（控制器使能）相关

位置测量系统 1/2（DB380x.DBX1.5 和 DB380x.DBX1.6）

主轴同步期间系统不会禁用联动主轴和引导主轴的位置测量系统的切换。切换后耦合仍保持生效。

说明

建议在撤销主轴同步后再切换联动主轴和引导主轴的位置测量系统。

删除剩余行程/主轴复位 (DB380x.DBX2.2)

引导主轴：主轴同步过程中设置主轴复位

设置引导主轴复位后，引导主轴会以设定好的速度减速至静止状态。主轴同步依然保持。

叠加运动（除非有 COUPON/COUPONC 指令）也会尽快结束。

主轴停止（进给停止）（DB380x.DBX4.3）

引导主轴和联动主轴：主轴同步过程中设置主轴停止

设置联动主轴或引导主轴的"主轴停止"后，两根主轴会同步减速至静止状态。主轴同步依然保持。

复位主轴停止

复位了两根主轴的"主轴停止"后，两根主轴会再次加速至最后生效的转速设定值。

应用示例

在主轴同步激活的状态下打开防护门时会使联动主轴和引导主轴停止。

引导主轴和联动主轴的信号曲线：

1. 停止：DB380x.DBX4.3 = 1（主轴停止）
2. 等待静止：DB390x.DBX1.4 == 1
3. 停止：DB380x.DBX2.1 = 0（控制器使能）

删除 S 值（DB380x.DBX2000.7）

引导主轴：主轴同步过程中删除 S 值

引导主轴会通过一个斜坡减速至静止状态。主轴同步依然保持。

联动主轴：主轴同步过程中删除 S 值

接口信号不作用于主轴同步中的联动主轴。

重新同步主轴 1/2（DB380x.DBX2000.4 和 DB380x.DBX2000.5）

引导主轴：主轴同步过程中重新同步位置测量系统。

说明

建议在撤销主轴同步后再重新同步引导主轴的位置测量系统。

重新同步（DB380x.DBX5007.4）

联动主轴：恢复写入的角度偏移

在一个已经丢失或未执行联动主轴和引导主轴同步的系统中，可再次恢复写入的角度偏移。

- 要求：DB380x.DBX5007.4 = 1（重新同步）
- 应答：DB380x.DBX5003.4 == 1（正在同步）

JOG 模式下的移动键（DB380x.DBX4.6 和 DB380x.DBX4.7）

在主轴同步中，系统内部**不会自动禁用** JOG 模式下联动主轴的“正/负移动键”，因此，在按下移动键以后会产生联动主轴的一个叠加运动。

说明

如果需要阻止叠加运动，那么必须由 PLC 用户程序来执行。

NC 停止进给轴和主轴（DB3200.DBX7.4）

在主轴同步中，通过“NC 停止进给轴和主轴”信号可使已同步的主轴在设定的动态响应下制动。此时，主轴同步依然保持。

NC 启动（DB3200.DBX7.1）

（参见章节“NC 启动时同步主轴耦合的响应 (页 237)”）

说明

在给出 NC 停止信号后给出再启动 NC 不会撤销主轴同步。

25.4.4 引导主轴和联动主轴之间的差速

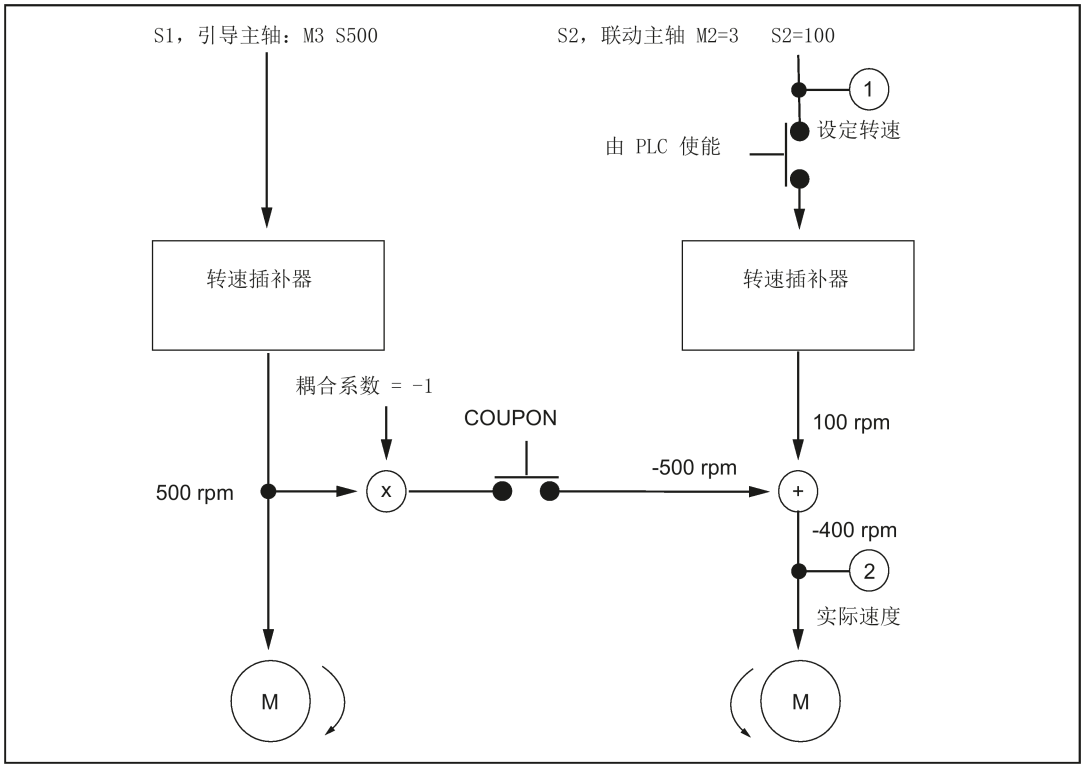
什么时候会出现差速？

在车床应用中，两根主轴相对放置时会产生差速。根据两个转速源之和（带符号），通过引导主轴的耦合系数可导出一个转速分量。此外还可为联动主轴编程设置转速（S...）和旋转方向（M3/M4）。

通常耦合系数设为“-1”即可保持同步。对该系数取反则可为联动主轴生成转速差，而无需进行额外的转速编程。

NC 的典型工作方式会在下图中详细说明。

示例

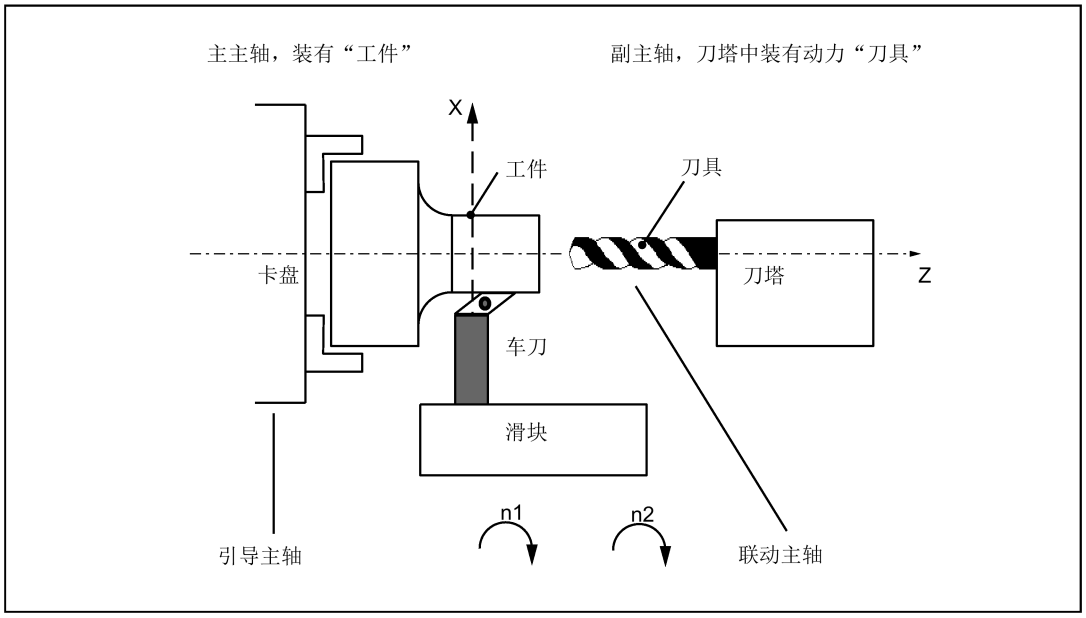


程序代码	注释
N01 M3 S500	; S1 以 500 转/分钟正转
	; 主主轴是主轴 1
N02 M2=3 S2=300	; S2 以 300 转/分钟正转
N05 G4 F1	
N10 COUPDEF(S2,S1,-1)	; 耦合系数 -1:1
N11 COUPON(S2,S1)	; 激活耦合。联动主轴 S2 的转速由引导主轴 S1 的转速和耦合系数得出。
N26 M2=3 S2=100	; 差速编程，S2 为联动主轴。

应用

在定位引导主轴上用动力刀具进行加工时要求联动主轴能够精确的同步。围绕联动主轴旋转的刀塔提供有不同的刀具进行加工。

下图展示的是一个应用示例，示例中的刀具和引导主轴是平行的。



前提条件

差速编程的基本条件：

- 具备主轴同步功能。
- 联动主轴的动态响能力至少要和引导主轴的相同。否则会在进行不带补偿夹具的攻丝 G331/G332 时出现质量问题。
- 差速必须在联动主轴位于的通道内编程。引导主轴可在另一个通道内进行编程。
- 差速必须由 PLC 通过设置联动主轴接口信号“使能叠加”（DB380x.DBX5002.4）来使能。 如果该使能不存在，系统会发出报警 16771 “通道%从动轴%叠加运动未使能”。该报警可通过置位接口信号“使能叠加运动”（DB380x.DBX5002.4）或结束耦合来清除。

说明

定位运动如 SPOS、SPOSA、M19 以及轴运动无需使能。

说明

差速对耦合过程没有影响。

在耦合激活时不可以进行引导主轴和联动主轴的变速换档。

通过 COUPONC 指令激活耦合

激活耦合后，联动主轴和之前一样按传动比加速到引导主轴转速。如果在激活耦合时联动主轴已经在某方向旋转（M3、M4），那么系统会继续采用联动主轴的该转速。

撤销耦合

撤销耦合后，联动主轴仍以两个转速分量之和继续旋转。主轴就像按照已编程的转速和旋转方向旋转。撤销耦合后主轴的工作方式保持不变。

差速

差速的形成途径有：在同步主轴耦合**期间**在转速开环模式中**重新**编程联动主轴的旋转方向和转速，比如：本示例中为 S2=... 或 M2=3，M2=4；另一种方法是通过编写 COUPONC 指令来激活同步主轴耦合，该指令会时使联动主轴继续采用之前编程的转速。

边界条件：

指定旋转方向 M3 或 M4 时，转速 S... 必须要一同重新指定。否则系统会发出报警 16111 “通道%程序段%主轴%未编写转速”。

读取联动主轴的偏移

在写入差速后，当前的偏移会一直发生变化。可通过 $AA_COUP_OFFS[Sn] 读取设定值的偏移，通过 $VA_COUP_OFFS[Sn] 读取实际值的偏移。

可通过变量 $P_COUP_OFFS[Sn] 读取程序写入的最后偏移。

显示差速

在写入了差速后（本例中是 100 rpm），差速作为转速设定值显示。

转速实际值指的是电机转速。在本例中实际转速为 500 rpm * (-1) + 100 rpm = -400 rpm。

接口信号：NC 至 PLC

转速开环模式中的联动主轴

当达到已写入的差速时（参见先前的示例 N26 M2=3 S2=100），NC 会置位联动主轴的接口信号“主轴位于设定值区域内”(DB390x.DBX2001.5)。虽然写入了差速，但是若 PLC 未使能差速，该 VDI 接口信号也不会置位。

编写了差速后，如果现有的耦合有需要，联动主轴可仍然保持在位置闭环控制模式下。

说明

当写入的差速生成了除同步设定值外其他的设定值时，轴 VDI 接口信号 NC → PLC “叠加运动”（DB390x.DBX5002.4）置位。

实际旋转方向为正转（DB390x.DBX2001.7）

接口信号“实际旋转方向为正转”（DB390x.DBX2001.7）指的是当前的电机旋转方向。

接口信号：PLC 至 NC

通过 PLC 接口控制联动主轴

下文说明了各个轴 VDI 接口信号对转速开环模式中联动主轴差速的影响。

删除剩余行程/主轴复位（DB380x.DBX2.2）

已写入的差速和旋转方向可由接口信号“删除剩余行程/主轴复位”（DB380x.DBX2.2）删除。置位接口信号“删除 S 值”（DB380x.DBX2000.7）可删除已写入的转速。

重新同步主轴 1/2（DB380x.DBX2000.4 和 DB380x.2000.5）

接口信号“重新同步主轴 1/2” (DB380x.DBX2000.4/DB380x.2000.5) **未**被禁用。耦合不会自动补偿位置偏移。

M3/M4 取反（DB380x.DBX2001.6）

接口信号“ M3/M4 取反（DB380x.DBX2001.6）”只能使额外写入的联动主轴转速分量取反。

该信号不会影响通过同步主轴耦合生成的转速分量。

主轴倍率（DB380x.DBB2003）

VDI 接口信号"主轴倍率"（DB380x.DBB2003）只会影响为联动主轴额外写入的转速分量。当主轴倍率开关传送至所有的轴输入端时，联动主轴的倍率值会发生**两次**变化：

- 一次因引导主轴转速变化而间接变化
- 另一次是联动主轴本身转速分量的直接变化。

在 PLC 程序中可对倍率值进行相应的调整。

耦合取消

撤销耦合后，联动主轴仍以两个转速分量之和继续旋转。即在撤销同步后主轴转速是恒定的。

在使用 COUPOF 撤销同步时，主轴的转速和旋转方向仍保持不变，仿佛在程序中写入了这两个指令。本例中主轴旋转方向和转速为 M4 S400。

在 COUPOFS 指令下，联动主轴会从当前的转速运行至静止状态。

激活附加功能

联动主轴也可作为主主轴。这种情况下可使用附加功能。

- G95、G96、G97 指令下的旋转进给率。通过指令 G96 S2=... 可以激活联动主轴的"恒定切削速度"功能。
  取决于 X 轴位置的转速是主轴 2 转速插补器的设定转速，计入主轴 2 的总转速计算中。
- 不带补偿夹具的攻丝 G331、G332。

25.4.5 同步补偿中同步信号的特性

同步补偿的作用

通过比较实际值和经过补偿的设定值可以得出新的同步信号。 补偿成功后同步信号应再次出现。

25.4.6 删除同步补偿及 NC 复位

无论在哪种情况下删除了同步补偿值，变量$AA_COUP_CORR[Sn]提供的值都是零：

- 使用 COUPON(..)/COUPONC(..) 指令激活相关联动主轴的同步主轴耦合时，当前的同步补偿会附加到设定位置上。
- 但在 NC 复位时，系统却不会在零件程序结束时将当前的同步补偿附加到设定位置上。同步信号不受此影响。
- 当前的同步补偿不受 M30 的影响。
- 用户也可以提前把变量 $AA_COUP_CORR 设为**零**来删除这个补偿值。同步补偿会立即结束或是在较大值时通过一个斜坡减速至结束。

25.4.7 同步主轴耦合功能调试的特殊性

主轴调试

首先，引导主轴和联动主轴要像一个普通的主轴一样进行一次调试。更多相关步骤的详细信息请参见章节"主轴（S1）(页 158)"。

前提条件

接着可由以下数据对同步主轴对进行定义：

- 引导主轴和联动主轴的机床轴号
  （在使用通道专用机床数据 MD21300 $MC_COUPLE_AXIS_1[n] 固定定义的耦合中）
- 所需的耦合方式（设定值耦合、实际值耦合或速度耦合）
  （在使用通道专用机床数据 MD21310 $MC_COUPLING_MODE_1[n] 固定定义的耦合中）
- 确定联动主轴和引导主轴的变速档
- 在固定定义的同步主轴耦合中还需要确定以下耦合特性：
  - 主轴同步中的程序段切换特性：
    MD21320 $MC_COUPLE_BLOCK_CHANGE_CTRL_1

– 耦合撤销特性：
  MD21330 $MC_COUPLE_RESET_MODE_1
– 耦合参数设有写保护：
  MD21340 $MC_COUPLE_IS_WRITE_PROT_1
– 同步主轴耦合的传动比：
  SD42300 $SC_COUPLE_RATIO_1[n]

设定值耦合中联动主轴和引导主轴的控制性能

为了在使用**设定值耦合**时达到最好的同步效果，联动主轴和引导主轴的动态响应必须相同，以确保良好的控制性能。最好分别**优化**各轴控制回路（位置、转速和电流控制器），以尽快且有效地调节误差。

为了使各轴不同的动态匹配相同且不降低控制性能，系统提供了设定值回路的**动态匹配**（参见章节"补偿（K3）(页 106)"）。必须优化引导主轴和联动主轴的下列控制参数设置：

- K_V 系数 (MD32200 $MA_POSCTRL_GAIN)
- 前馈控制参数
  MD32620 $MA_FFW_MODE
  MD32610 $MA_VELO_FFW_WEIGHT
  MD32650 $MA_AX_INERTIA
  MD32800 $MA_EQUIV_CURRCTRL_TIME
  MD32810 $MA_EQUIV_SPEEDCTRL_TIME

丢失同步时的特性：

- 轴专用 MD32620 $MA_FFW_MODE

我们建议设置 MD32620 = 3，将联动主轴的**前馈方式**设为采用 Tt 对称的转速前馈。

可通过更改轴专用的机床数据对这种前馈方式的对称过程进行进一步优化：

机床数据	含义
MD32810 EQUIV_SPEEDCTRL_TIME	转速环前馈控制的等效时间常量
MD37200 COUPLE_POS_TOL_COURSE	"粗同步"的阈值
MD37210 COUPLE_POS_TOL_FINE	"精同步"的阈值
MD37220 COUPLE_VELO_TOL_COURSE	速度公差"粗"
MD37220 COUPLE_VELO_TOL_FINE	速度公差"精"

这些情况下，同步信号的较高阈值和位置和/或转速公差会让结果更稳定。

动力匹配

联动主轴和引导主轴的动态响应必须相同，以达到最好的同步效果。在任何速度下，联动主轴与引导主轴的跟随误差必须相同。设定值回路中的动力匹配可有效平衡动态响应不同的主轴之间的控制性能。动力匹配的时间常量应当是动态响应"最慢"的轴和其他轴之间等效时间常量的差值。

示例

在转速前馈控制已激活的前提下，动力基本上由"最慢"转速调节回路的额外时间常量确定。

- 引导主轴的等效时间常量：MD32810 $MA_EQUIV_SPEEDCTRL_TIME[<引导主轴>] = 5 ms
- 联动主轴的等效时间常量：MD32810 $MA_EQUIV_SPEEDCTRL_TIME[<联动主轴>] = 3 ms
- 联动主轴的动力匹配时间常量：MD32910 $MA_DYN_MATCH_TIME[<联动主轴>] = 5 ms - 3 ms = 2 ms
- 激活联动主轴的动力匹配：MD32900 $MA_DYN_MATCH_ENABLE[<联动主轴>] = 1

动力匹配设置正确，则联动主轴与引导主轴的跟随误差相同：报警操作区域 >

必要时可稍微调整 K_V 系数或前馈参数，以达到最佳的效果。

位置闭环控制参数组

主轴上，每个变速档对应一个位置闭环控制参数组。通过这些参数组可对主轴同步中的引导主轴和联动主轴的动态响应进行调整。为此，须为转速模式和定位模式及主轴同步各指定另一个变速档。激活各个运行方式前须切换至对应的变速档。

控制参数

引导主轴和联动主轴的下列闭环控制参数的设置必须相同：

- MD33000 $MA_FIPO_TYPE（精插补器类型）
- MD32400 $MA_AX_JERK_ENABLE（最大轴急动度）
- MD32402 $MA_AX_JERK_MODE（用于轴向最大急动度的滤波器类型）
- MD32410 $MA_AX_JERK_TIME（轴向急动滤波器的时间常量）
- MD32412 $MA_AX_JERK_FREQ（轴向急动滤波器的禁用频率）
- MD32414 $MA_AX_JERK_DAMP （轴向急动滤波器的衰减）
- MD32420 $MA_JOG_AND_POS_JERK_ENABLE（使能最大急动度）
- MD32430 $MA_JOG_AND_POS_MAX_JERK（轴向急动）

联动主轴：自动设置控制参数

可通过机床数据按照以下方式来设置联动主轴的闭环控制参数：

MD30455 $MA_MISC_FUNCTION_MASK

位 5=0：同步主轴耦合，联动主轴：

位置闭环控制、前馈控制和参数组的设置和联动主轴上的设置一样。

位 5=1：同步主轴耦合：

控制参数的设置和在没有激活主轴同步的状态下的设置一样。

主轴同步中自动接收引导主轴参数

在主轴同步中，可由主轴接收闭环控制、前馈控制和联动主轴参数组的参数值，以自动执行联动主轴和引导主轴的动力匹配。

MD30455 $MA_MISC_FUNCTION_MASK, 位 5

主轴模式和进给轴模式下不同的动态响应

可通过以下机床数据设置，使得动态响应指令 FA、OVRA、ACC 和 VELOLIMA 对主轴或进给轴单独生效：

MD30455 $MA_MISK_FUNCTION_MASK 位 6=0

随后在上述指令中写入进给轴名或主轴名。例如：VELOLIMA[S1]=50 指令只能使处于主轴模式的主轴的最大转速降到 50%，而 VELOLIMA[C]=50 指令只能使处于进给轴模式中的主轴的最大转速降到 50%。

如果希望 VELOLIMA[S1]=50 和 VELOLIMA[C]=50 指令具有相同的作用，就如下设置该机床数据，使得指令 FA、OVRA、ACC 和 VELOLIM 对轴统一生效，如同之前的指令编程方式。

MD30455 $MA_MISK_FUNCTION_MASK 位 6=1

折线式加速度特性曲线

引导主轴的折线式加速度特性曲线对联动主轴的影响可通过以下轴专用的机床数据来确定：

- MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT（加速度下调系数）和
- MD35230 $MA_ACCEL_REDUCTION_FACTOR（加速度下调）。

只要有数据 MD35242 $MA_ACCEL_REDUCTION_TYPE，就可以对加速度下调方式进行定义。否则加速度会呈双曲线的形式下降。

如果在计算了传动比的条件下，联动主轴的动态响应小于引导主轴的动态响应，系统会自动将引导主轴的动态响应下调至一个合理值。

联动主轴的加速度在整个转速范围内应保持**恒定**。如果通过上述机床数据也为联动主轴定义了一条折线式加速度特性曲线，那系统只会在激活同步的过程中考虑该加速度特性曲线。联动主轴的设定值按定义的折线式加速度特性曲线变化。

更多信息请参见章节“加速度（B2）(页 39)”。

实际值耦合

使用实际值耦合（AV）时，联动主轴的动态响应大大超过引导主轴的动态响应。系统也会根据各轴动态响应来优化设置各个驱动。

实际值耦合只应在特殊情况下使用。

速度耦合

在系统内部是一个设定值耦合，但速度耦合（VV）对联动主轴和引导主轴的动态响应要比设定值耦合（DV）低一些。联动主轴和/或引导主轴不需要位置闭环控制。不需要测量系统。

粗/精同步的阈值

在优化了控制器以及设置了前馈控制后，必须为联动主轴输入粗/精同步的阈值。

- “粗同步”的阈值
  轴专用 MD7200：AV（实际值）、DV（设定值）：COUPLE_POS_TOL_COARSE
  MD37220：VV（速度值）：COUPLE_VELO_TOL_COARSE
- “精同步”的阈值
  轴专用 MD37210：AV（实际值）、DV（设定值）：COUPLE_POS_TOL_FINE
  MD37230：VV（速度值）：COUPLE_VELO_TOL_FINE

联动主轴的上述值应根据机床制造商要求的加工精度来确定，并由 PLC 接口通过信息显示窗口进行检查。

引导主轴/联动主轴的角度偏移

例如：如果在激活主轴同步时需要两者之间存在指定的角度偏移，那就必须使联动主轴和引导主轴的“零度位置”一致。该操作可通过以下机床数据完成：

- MD34100 $MA_REFP_SET_POS
- MD34080 $MA_REFP_MOVE_DIST
- MD34090 $MA_REFP_ MOVE_DIST_CORR

更多信息请参见章节“回参考点（R1）(页 152)”。

显示联动主轴信息

调试主轴同步时，在报警操作区域会显示联动主轴的以下值：

- 联动主轴和引导主轴的设定值偏差
  显示的值：相对于引导主轴（设定值）的位置偏移
  （该值与通过轴变量 $AA_COUP_OFFS 在零件程序中可读的角度偏移一致）
- 联动主轴和引导主轴的实际值偏差
  显示的值：相对于引导主轴（实际值）的位置偏移

25.5 边界条件

“主轴同步”功能的可用性

该功能是一个选件功能（“通用耦合‘CP-BASIC’”），它必须获得授权后方可和硬件配用。

说明

有关通用耦合的不同类型的更多信息，请参见章节“轴耦合（M3）(页 58)”。

25.6 示例

编程示例

程序代码	注释
	; 引导主轴 = 主主轴 = 主轴 1
	; 联动主轴 = 主轴 2
N05 M3 S3000 M2=4 S2=500	; 主主轴转速为 3000 rpm
	; 联动主轴：500 rpm
N10 COUPDEF (S2, S1, 1, 1, "No", "Dv")	; 同步也可以由机床数据
	; 定义

程序代码	注释
N70 SPCON	; 引导主轴进入位置闭环控制
	; (设定值同步)。
N75 SPCON(2)	; 联动主轴进入位置闭环控制。
N80 COUPON (S2, S1, 45)	; 实时同步，两者之间的角度偏移为 45 度。
N200 FA [S2] = 100	; 定位速度 = 100 度/分
N205 SPOS[2] = IC(-90)	; 负方向上叠加转动 90 度
N210 WAITC(S2, "Fine")	; 等待"精同步"
N212 G1 X.., Y..F...	; 加工
N215 SPOS[2] = IC(180)	; 正方向上叠加转动 180 度
N220 G4 S50	; 暂停时间 = 50 转
	; 主主轴
N225 FA [S2] = 0	; 激活由机床数据定义的速度
N230 SPOS[2] = IC (-7200)	; 20 转，以机床数据定义的速度
	; 在负方向上
N350 COUPOF (S2, S1)	; 快速撤销同步，S = S2 = 3000
N355 SPOSA[2] = 0	; 联动主轴停止在零度位置上
N360 G0 X0 Y0	
N365 WAITS(2)	; 等待主轴 2
N370 M5	; 停止联动主轴
N375 M30	

25.7 数据表

25.7.1 机床数据

序号	标识符	名称
NC 专用		
10000	AXCONF_MACHAX_NAME_TAB	机床轴名称
通道专用		
20070	AXCONF_MACHAX_USED	通道内有效的机床轴号
21300	COUPLE_AXIS_1	定义同步主轴对
21310	COUPLING_MODE_1	主轴同步方式
21320	COUPLE_BLOCK_CHANGE_CTRL_1	主轴同步中的程序段切换特性
21330	COUPLE_RESET_MODE_1	同步撤销特性
21340	COUPLE_IS_WRITE_PROT_1	无法修改同步参数
进给轴/主轴专用		
30455	MISK_FUNCTION_MASK	轴功能
32200	POSCTRL_GAIN	伺服增益系数（K_V 系数）
32400	AX_JERK_ENABLE	单轴加加速度上限
32410	AX_JERK_TIME	用于单轴加加速度滤波器的时间常数
32420	JOG_AND_POS_JERK_ENABLE	单轴加加速度上限的缺省设置
32430	JOG_AND_POS_MAX_JERK	单轴加加速度
32620	FFW_MODE	前馈控制类型
32810	EQUIV_SPEEDCTRL_TIME	转速环前馈控制的等效时间常量
34080	REFP_MOVE_DIST	参考点间距
34090	REFP_MOVE_DIST_CORR	参考点偏移
34100	REFP_SET_POS	参考点值

序号	标识符	名称
35000	SPIND_ASSIGN_TO_MACHAX	分配主轴至机床轴
37200	COUPLE_POS_TOL_COARSE	“粗同步”的阈值
37210	COUPLE_POS_TOL_FINE	“精同步”的阈值
37220	COUPLE_VELO_TOL_COARSE	引导主轴与联动主轴之间的“粗”速度公差
37230	COUPLE_VELO_TOL_FINE	引导主轴与联动主轴之间的“精”速度公差
37240	COUP_SYNC_DELAY_TIME	实际值主轴同步的延迟时间

25.7.2 设定数据

序号	标识符	名称
进给轴/主轴专用		
42300	COUPLE_RATIO_1	主轴同步的传动比

25.7.3 接口信号

序号	位	名称
通道专用		
DB1700.DBX0000	.6	空运行进给率已选择
DB1700.DBX0001	.3	快速移动进给倍率已选择
DB3200.DBX0007	.1	NC 启动
DB3200.DBX0007	.4	NC 停止进给轴和主轴
进给轴/主轴专用		
DB380x.DBX0001	.3	进给轴/主轴锁定
DB380x.DBX0001	.4	跟随运行方式
DB380x.DBX0001	.5/6	位置测量系统 1，位置测量系统 2
DB380x.DBX0002	.1	控制器使能
DB380x.DBX0002	.2	剩余行程/主轴复位
DB380x.DBX0004	.3	主轴停止/进给停止
DB380x.DBX0004	.6/.7	JOG 模式下的移动键
DB380x.DBX2000	.4/5	重新同步主轴 1，重新同步主轴 2
DB380x.DBX2000	.7	删除 S 值
DB380x.DBX2001	.0	进给倍率有效
DB380x.DBX2001	.6	M3/M4 反向
DB380x.DBB2003	-	主轴倍率
DB390x.DBX0000	.4/5	已回参考点/已同步 1，已回参考点/已同步 2
DB390x.DBX2002	.4	主轴同步
DB390x.DBX5002	.0	精同步
DB390x.DBX5002	.1	粗同步
DB390x.DBX5002	.2	实际值耦合
DB390x.DBX5002	.4	叠加运动
DB390x.DBX5003	.0	引导主轴有效
DB390x.DBX5003	.1	联动主轴有效

26 安全集成（Safety Integrated）

26.1 标准和规定

26.1.1 概述

26.1.1.1 目标

技术设备和产品的制造商和销售商有责任确保设备和产品的安全性。也就是说，他们应采用最先进的技术设计出尽可能安全的设备、机械及其它技术装置。为此，有关经济合作组织在标准中说明了所有对安全非常重要的先进技术。遵循这些相关标准能够确保设备达到了先进安全技术水平，进而确保设备安装人员和机械/设备制造商履行了相关义务。

安全技术应尽可能避免设备对人和环境造成危险，同时尽可能少地限制工业生产和设备使用（除非是一些必要的限制）。一些全球标准和规定旨在为所有国家的人员和环境提供相同程度的保护。这也避免了由于不同国家对安全要求高低不同而引起的恶性竞争。

在不同的国家和地区，设备安全保障有不同的方案和要求。在法律规定和安全要求中，何时检查设备是否充分安全、采用什么方法检查以及责任分配等也各不相同。
机械制造商和设备安装人员必须确保机械和设备的安全性符合使用地的法律法规。例如，在美国使用的机械的控制系统必须符合美国当地的要求，即使其制造商来自欧洲经济共同体（EEA）。

26.1.1.2 功能安全

从需要保护的对象的角度来看，安全是一个密不可分的整体。但造成危险的原因和避免这些危险的技术措施可能存在很大的差异。因此人们将安全分为不同的类型，例如根据造成危险的原因进行分类。当安全取决于功能正常工作时，该安全便是“功能安全”。
为了保证机械或设备的功能安全，保护装置/控制装置的安全部件必须正常工作。并且在故障状态下能够使设备保持在安全状态或将设备进入安全状态。为此需要使用符合相关标准的专业技术。对功能安全的要求基于以下几个基本目标：

- 避免系统故障
- 控制系统故障
- 控制偶然发生的错误或故障

衡量达到的功能安全的尺度有：危险故障发生的几率、故障公差和系统故障降到最低水平后应达到的质量水平。这些尺度在不同的标准中由不同的术语表示。在 IEC/EN 61508，IEC/EN 62061 中是“安全集成等级”（SIL）；在 EN ISO 13849-1 中是“类别”和“性能等级”（PL）。

26.1.2 欧洲的机械安全

和产品生产相关的欧盟指令以调控自由商品贸易的欧盟条约第 95 条为基础。这些指令是基于一个新的全球措施（“new approach”，“global approach”）形成的：

- 欧盟指令只描述了通用安全目标和定义了基本安全要求。
- 技术详细信息只能由欧洲议会和欧盟理事会委任的标准委员会（CEN、CENELEC）在标准中定义。这些标准与特定指令保持协调，并且公布在欧洲会议和欧盟理事会公报中。立法者不会强制规定要遵守某标准。但是如果遵守了这些协调标准，便可假定为满足了相关指令的所有安全要求。
- 欧盟指令要求成员国之间相互承认彼此的国家规定。

欧盟指令彼此之间具有等同的效力。即一个特定的设备涉及到多个指令时，所有相关指令的要求都生效（例如对于带电气装置的机械，机械指令和低压指令都适用）。

26.1.2.1 机械指令

附件 I 中规定了机械类产品的基本健康和安全要求，必须符合这些要求。

必须尽责地实施保护目标，以符合指令要求。

机械制造商必须出具证明，表明设备符合基本要求。使用协调标准可以简化证明过程。

26.1.2.2 欧洲协调标准

欧洲协调标准由欧盟委员会授权的两个标准机构 CEN（Comité Européen de Normalisation）和 CENELEC（Comité Européen de Normalisation Éctrotechnique）制定，用于对特定产品的欧盟指令要求进行详细描述。这些标准（欧盟标准）随欧洲议会和欧盟理事会公报颁布，之后必须不加更改地纳入各成员国的国家标准中。它们满足基本的安全和健康要求，以及机械指令的附件 I 中所述的保护目标。

在遵循了相关协调标准的情况下，会“自动假定”为遵循了机械指令。即相关的安全要求包含在特定标准中时，制造商在遵循了该标准的情况下可假定遵循了机械指令。但并非所有欧洲标准都为协调标准。协调标准会公布在欧洲议会和欧盟理事会公报中。

欧洲的机械安全标准按等级分类。它可划分为：

- A 类标准（基本标准）
- B 类标准（类别标准）
- C 类标准（产品标准）

A 类标准/基本标准

A 类标准中包含对所有机械的概念和定义。例如 EN ISO 12100-1（原 EN 292-1）“机械安全 - 基本概念，通用设计原则”。

A 类标准主要针对制定 B 类/C 类标准的机构。如果没有相应的 C 类标准，其中的风险最小化措施对制造商也非常有用。

B 类标准/类别标准

B 类标准为涉及了多种机械类别的安全技术标准。B 类标准主要针对制定 C 类标准的机构。如果没有相应的 C 类标准，其中对机械设计和结构的规定对制造商也非常有用。

B 类标准还可以进一步划分为：

- B1 类标准，用于高级安全要素，例如人体工学原则、与危险来源的安全距离，防止身体部位受到撞击的最小距离。
- B2 类标准，用于各种机械类别的安全设备，例如急停设备、双手控制设备、联锁设备、非接触生效防护设备、控制系统的安全部件。

C 类标准/产品标准

C 类标准为特定产品的专用标准，例如机床、木材加工机、升降机、包装机、印刷机等。产品标准为对特定机械的要求。这些要求有时可能会与基本标准和类别标准有所差别。对于机械制造商，C 类标准/产品标准具有最高的优先级。符合该标准时，便可以假定机械制造商符合了机械指令附录 I 的基本要求（自动符合性假设）。如果某种机械无产品标准，则可在机械结构设计时采用 B 类标准。

标准的完整清单以及授权的标准草案都可以从以下网址获取：

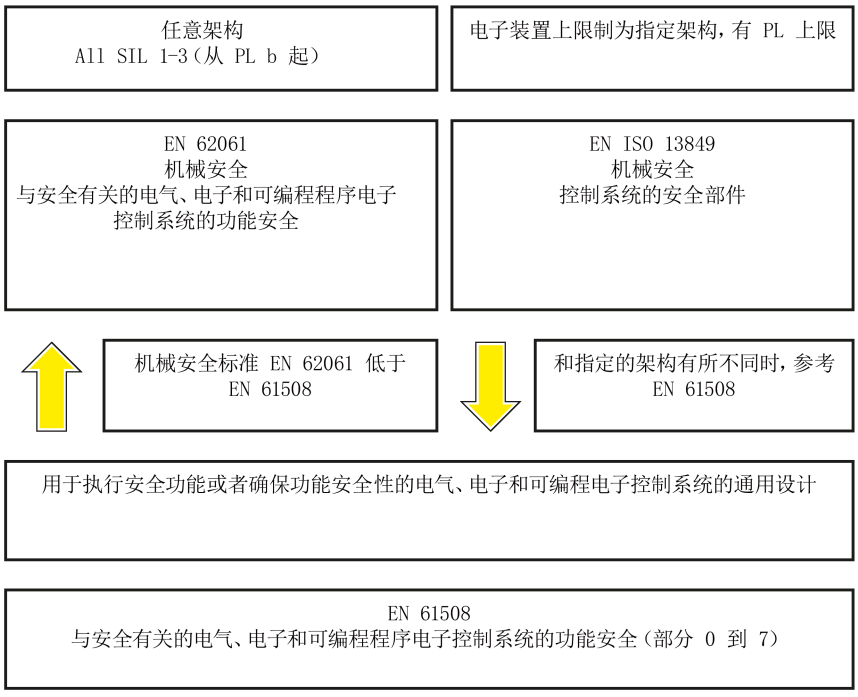
http://www.newapproach.org/

建议：由于科技高速发展，机械标准中的改动也较为频繁，因此在使用这些标准（尤其是 C 类标准）时需要特别注意它的时效性。此外还需要注意，产品不一定符合这些标准，但是一定要达到相关欧盟指令中的所有安全目标。

26.1.2.3 控制系统安全设计相关标准

如果机械的功能安全性能取决于控制系统功能，则在设计控制系统时必须将发生安全功能风险的几率降到足够低。标准 EN ISO 13849-1 和 EN IEC61508 定义了机械控制系统安全设计相关的原则，这些原则符合欧盟机械指令中的所有安全目标。使用此标准即可满足机械指令中的相关安全目标。

控制系统安全设计相关标准：



EN ISO 13849-1，EN 62061，和 EN 61508 的应用范围相近。为了便于用户选择，IEC 协会和 ISO 协会都在其标准的引言中以同一张表格列出了两种标准的应用范围。根据工艺（机械、气动、液压、电气、电子、可编程电子）、风险分级、架构等要素选择使用 EN ISO 13849-1 或 EN 62061。

类型	执行安全相关控制功能的工艺	EN ISO 13849-1	EN 62061
A	非电气（例如液压、气压）	X	未被涵盖
B	电子机械装置（例如继电器和/或简单电子装置）	限制为制定架构（参见注释 1），最大为 PL = e	所有架构，最大到 SIL 3
C	复杂电子装置（例如可编程电子装置）	限制为制定架构（参见注释 1），最大为 PL = d	所有架构，最大到 SIL 3
D	A 标准 + B 标准	限制为制定架构（参见注释 1），最大为 PL = e	X 参见注释 3
E	C 标准 + B 标准	限制为制定架构（参见注释 1），最大为 PL = d	所有架构，最大到 SIL 3
F	C 标准 + A 标准或者 C 标准 + A 标准或者 C 标准 + B 标准	X 参见注释 2	X 参见注释 3

“X”表示标准涵盖了此项。

注释 1：
指定架构在 EN ISO 13849-1 的附录 B 中描述，是定量分析的简单基础。

注释 2：
对于复杂电子装置：使用符合 EN ISO 13849-1，最大为 PL = d 的架构，或者使用符合 EN 62061 的架构。

注释 3：
对于非电气工艺：使用符合 EN ISO 13849-1 的组件作为子系统。

26.1.2.4 DIN EN ISO 13849-1

根据 DIN EN 13849-1 进行的定性分析已不适用于现代控制系统（工艺原因）。例如 DIN EN ISO 13849-1 中没有考虑时间特性（例如时间间隔、循环测试、寿命）。而时间特性是 DIN EN ISO 13849-1 中的可靠性测试基础（每个时间单位发生故障的几率）。
DIN EN ISO 13849-1 以已知的 EN 954-1 的类别为基础。它同样涵盖了所有安全功能以及所有参与了安全功能执行的设备。使用 DIN EN ISO 13849-1 时，除了原先 EN 954-1 中的定性分析，还对安全功能进行定量分析。并基于类别使用性能等级（PL）这一参数。以下安全技术参数是组件/设备必需的：

- 类别（结构要求）
- PL：性能等级
- $MTTF_d$：平均无危险故障时间
- DC：诊断覆盖率
- CCF：共因故障

该标准描述了如何在指定架构的基础上、计算控制系统中安全相关部件的性能等级 PL。与此有偏差时，EN ISO 13849-1 参考 EN 61508。

对于由多个安全相关部件组成的整体系统，此标准用于说明如何计算总 PL。

说明

DIN EN ISO 13849-1 和机械指令

从 2007 年五月起，DIN EN ISO 13849-1 和机械指令统一。

26.1.2.5 EN 62061

EN 62061（等同于 IEC 62061）是 IEC/EN 61508 以下的用于特定领域的标准。它对机械上和安全相关的电气控制系统的设计和制造进行描述，涉及从设计阶段到设备退役的整个生命周期。此标准基于安全功能的定量和定性分析，对于复杂的控制系统，标准采用“Top-Down”的描述方式，即“功能分解（Functional Decomposition）”。此时，它将通过风险分析得到的安全功能划分为子安全功能，并将它们分配给实际设备（子系统和子系统单元）。其中涵盖了硬件和软件。EN 62061 也描述了对应用程序设计的要求。

和安全相关的控制系统由不同子系统组成。子系统通过安全技术参数（SIL 索赔期限和 PFHD）说明。
可编程电子设备（例如 PLC）或者调速驱动器必须符合 EN 61508。它们可作为子系统集成到控制系统中。子系统的制造商必须给出以下安全技术参数。

子系统安全技术参数：

- SIL CL：SIL 索赔期限
- PFHD：每小时发生危险故障的几率
- T1：寿命

简单子系统（例如由电子机械部件组成的传感器或执行器）由相连的不同子系统单元（设备）组成，子系统的 PFHD 值可以由这些单元的参数值计算得出。

子系统单元（设备）的安全技术参数：

- λ：故障率
- B10 值：用于易磨损的单元
- T1：寿命

对于电子机械设备，设备制造商给出的是特定开关次数下的故障率。和时间相关的故障率、寿命必须根据实际应用的开关频率确定。

在结构设计阶段需要为子系统（由子系统单元构成）定义的参数：

- T2：诊断时间间隔
- β：对共因故障的灵敏性
- DC：诊断覆盖率

安全相关控制系统的 PFHD 值由单个子系统的 PFHD 值相加得出。
用户可通过以下方式实现和安全相关的控制系统：

- 使用已经满足了 EN ISO 13849-1、IEC/EN 61508 或 IEC/EN 62061 要求的设备和子系统。在标准中会说明在实现安全功能时如何集成符合要求的设备。
- 开发自己的子系统：
  - 可编程的电子系统或复杂系统：使用 EN 61508 或 EN 61800-5-2。
  - 简单设备和子系统：使用 EN 62061。

EN 62061 中未涵盖对非电气系统的描述。此标准提供了用于设计制造电气、电子和可编程电子控制系统的详细信息。非电气系统必须使用 EN ISO 13849-1。

说明

功能示例

简单子系统的设计、集成信息已作为“功能示例”出版。

说明

EN 62061 和机械指令

在欧洲，IEC 62061 已经批准为 EN 62061，且与机械指令协调。

26.1.2.6 系列标准 EN 61508（VDE 0803）

此系列标准对最先进的技术进行了描述。

EN 61508 与任何欧盟指令都不协调， 因此不会“自动假定”满足了指令中的保护目标。但是根据新的条款，安全相关产品的制造商也可以使用 EN 61508 来满足欧洲指令中的基本要求，例如在以下情况下：

- 在相关应用领域没有适用的协调标准。在此情况下制造商可使用 EN 61508，但是不存在符合性假定。
- 欧洲协调标准（例如：EN 62061、EN ISO 13849、EN 60204-1）中引用了 EN 61508 时。这样可以确保满足指令中的相关要求（“一同生效的标准”）。从引用的角度来说，如果制造商负责地使用了 EN 61508，则自动假定制造商符合了引用该标准的协调标准。

系列标准 EN 61508 包含了使用 E/E/PES 系统（electrical/electronic/programmable electronic System）执行安全功能或者确保功能安全性时需要考虑的全部要素。其它危险例如电击危险将不在标准中描述（与 EN ISO 13849 相似）。

近来 EN 61508 被定位为“国际基本安全出版物”，作为其它行业标准的框架，例如：EN 62061。由于它的国际定位，此标准在世界范围内具有很高的接受度，尤其是在北美和汽车工业领域。现在很多机构要求此标准例如作为 NRTL 清单的基础。

此外，EN 61508 的新发展还包括：从传感器到执行器的全套安全安装的系统设计技术要求；由意外硬件故障造成的危险故障几率的量化，以及为整个 E/E/PES 安全生命周期的每个阶段创建文档。

26.1.2.7 风险分析/评估

由于自身的结构和功能，机械和设备存在风险。因此机械指令要求对每台机械进行风险评估，并在必要时降低风险，使遗留风险小于允许的风险。执行风险评估时必须使用以下标准：

EN ISO 12100-1“机械安全 - 基本概念，通用设计原则”

EN ISO 13849-1“机械安全 - 控制系统安全部件”

EN ISO 12100-1 重点描述了需要分析的风险和风险降低的设计原则.

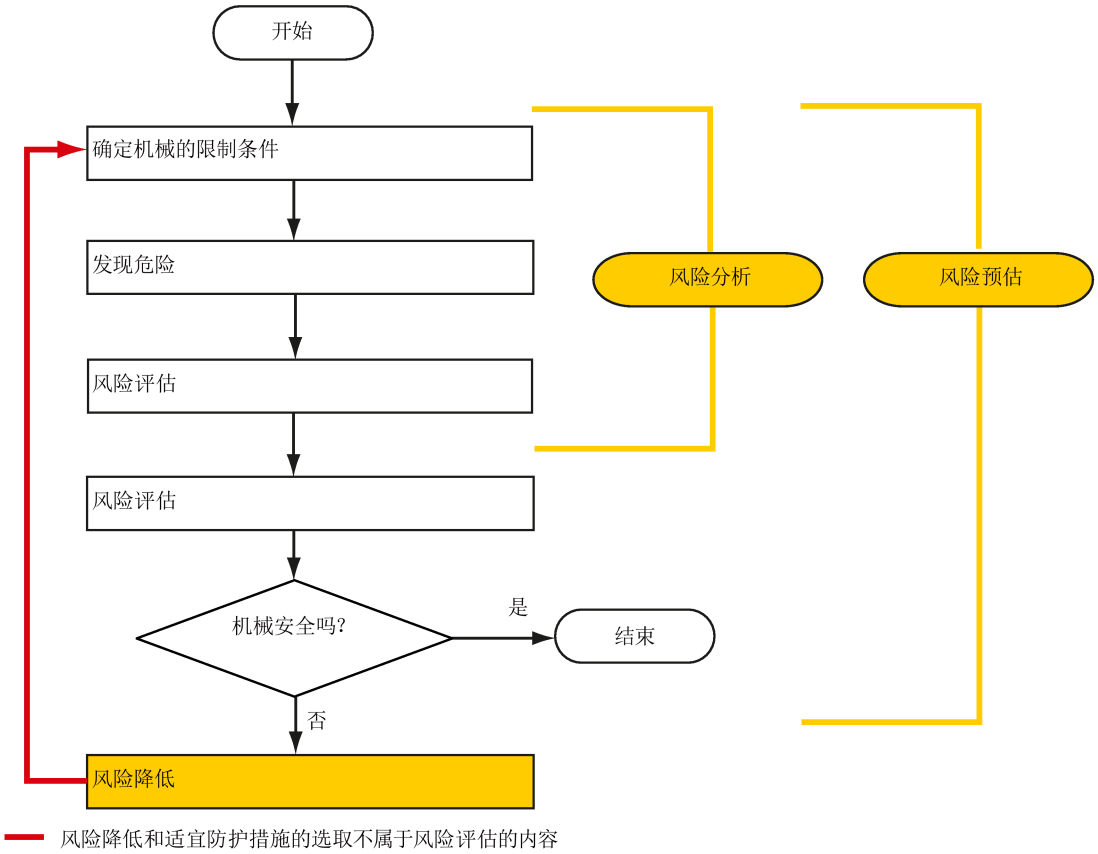
风险评估是指对机械造成的危险进行系统研究的一系列步骤。完成风险评估后要采取相应的降低风险措施。然后再次评估风险、降低风险，由此形成了一个不断重复的过程。这样可以尽可能地消除故障，确保采取了相应的保护措施。

风险评估包括：

- 风险分析
  - 确定机械的限制（EN ISO 12100-1）
  - 风险识别（EN ISO 12100-114）
  - 风险预估（EN 1050 第 7 段）
- 风险评估

根据实现安全性的重复过程，在风险预估后要进行风险评估。此时要决定是否需要降低风险。如果需要继续降低风险，必须选择和使用适当的保护措施。然后必须重复风险评估。

实现安全性的重复过程：



风险降低必须通过适宜的机械设计/制造来实现，例如通过适用于安全功能的控制系统或保护措施。

如果保护措施中包含了联锁和控制功能，则保护措施必须根据 EN ISO 13849-1 设计。除了 EN ISO 13849-1，电气和电子系统也可使用 EN 62061。此时，电子控制系统和总线系统还必须符合 IEC/EN 61508。

26.1.2.8 风险降低

为了实现这些控制系统功能，必须遵循标准中规定的根据风险程度分级的特殊要求。为了实现这些控制系统功能，必须遵循标准中规定的根据风险程度分级的特殊要求。这些要求在 EN ISO 13849-1 中描述，电气系统，尤其是带可编程电子设备的电气系统在 EN 61508 或 EN 62061 中描述。对和安全相关的控制系统部件的要求根据风险程度以及必要的风险降低措施分级。

EN ISO 13849-1 定义了风险矩阵，使用性能等级（Performance Level，PL）取代“类别”。

IEC/EN 62061 使用安全集成等级（Safety Integrity Level，SIL）分级。它是控制系统安全性能的量化标度。必要的 SIL 也根据 ISO 12100（EN 1050）的风险评估原则得出。在标准的附录 A 中描述了确定必要的安全集成等级 SIL 的方法。

不论使用的是哪种标准，在各种情况下都必须确保机械控制系统上所有参与安全功能执行的组件都满足这些要求。

26.1.2.9 遗留风险

在技术高度发展的当今世界，安全只是一个相对的概念。在现实中是无法完全排除风险达到绝对安全的，即所谓的“零风险保障”。遗留风险是指按照先进的经济和技术条件执行了相应的保护措施后仍无法避免的风险。

在机械/设备文档中必须提示遗留风险（用户信息，根据 EN ISO 12100-2）。

26.1.3 美国的机械安全

美国和欧洲对工作环境安全的法律规定的重要区别在于，在美国没有统一的针对机械安全的国家法规来规定制造商/供应商的责任。更多是要求雇主提供安全的工作环境。

26.1.3.1 OSHA 的最低要求

在 1970 年的职业安全法（Occupational Safety and Health Act，OSHA）中规定了雇主必须提供安全的工作环境的要求。OSHA 的核心要求位于第 5 段“责任（Duties）”中。

职业安全法由职业安全与健康管理局（Occupational Safety and Health Administration，通常称为 OSHA）负责管理。OSHA 会安排地区检察员来检查工作环境是否符合规定。

OSHA 中的工作安全相关规定在 OSHA 29 CFR 1910.xxx（“OSHA 法规（29 CFR）部分 1910 职业安全和健康”）中描述。（CFR：Code of Federal Regulations，美国联邦法规）

http://www.osha.gov

在 29 CFR 1910.5“标准适用性（Applicability of standards）”中确定了标准的适用范围。此条款与欧洲的相关规定相似。当特定产品的标准涵盖了相关要素时，则具有比通用标准更高的优先级。满足了此标准时，雇主可以推定满足了标准中相关要素的 OSHA 核心要求。

对于某些特定应用，OSHA 要求所有用于保护雇员的电气设备必须获得 OSHA 授权的国家认可实验室（Nationally Recognized Testing Laboratory，NRTL）的许可。

除了 OSHA 的规定外，还须遵守其它组织如 NFPA 和 ANSI 制订的标准，以及在美国广泛适用的产品责任法。产品责任法颁布后，制造商和销售商不得不为了自己的利益认真遵守法规并采用先进技术。

第三方保险企业通常要求其客户满足标准机构制定的适用的标准的要求。自保企业不受此要求约束，但是在发生事故时必须能证明遵循了普遍公认的安全原则。

26.1.3.2 NRTL 清单

所有在美国使用的电气设备都必须获得 OSHA 授权的国家认可实验室（Nationally Recognized Testing Laboratory，NRTL）的许可，以保障雇员的安全。国家认可实验室有权通过清单、标签或其它方式颁发设备和材料的许可。检测的基础为国家标准例如 NFPA 79，以及国际标准例如 E/E/PES 系统的标准 IEC/EN 61508。

26.1.3.3 NFPA 79

NFPA 79 标准（工业机械电子标准）适用于额定电压小于 600 V 的工业机械电子设备。由多台协同工作的机械所组成的整体也称之为机械。

NFPA 79 中涵盖了对可编程电子系统和通讯总线的基本要求。当这些设备被用于安全相关的功能时，必须列出在清单上。满足了这些要求时，电子控制系统和通讯总线也可用于停止类别 0 和 1 的急停功能（参见 NFPA 79 9.2.5.4.1.4）。同 EN 60204-1 一样，NFPA 79 也不再要求在执行急停功能时通过电子机械方式断开电源。

对可编程电子系统和通讯总线的核心要求为：
系统要求（参见 NFPA 79 9.4.3）。

1.包含了基于软件的控制器的控制系统必须：

- 在发生单个故障时
  - 断开系统使其进入安全状态
  - 防止重新启动直到故障被消除
  - 防止意外的启动
- 具有和硬件式控制系统相同的防护等级
- 根据符合公认的、适用于该系统的标准设计。

2.在附注中将 IEC 61508、IEC 62061、ISO 13849-1、ISO 13849-2 和 IEC 61800-5-2 列为适用的标准。

美国安全检测实验室公司保险商实验所（Underwriter Laboratories Inc. UL）定义了一个特殊的类别“可编程安全控制器”（名称代码 NRGF）用于满足此要求。此类别涵盖了包含软件且设计用于安全功能的控制设备。

对此类别的详细描述以及满足此要求的设备列表可通过以下网址获取：

http://www.ul.com → certifications directory → UL Category code/Guide information → 搜索类别“NRGF”

TUV Rheinland of North America 也是此应用的 NRTL。

26.1.3.4 ANSI B11

ANSI B11 标准是由多个协会，例如美国制造技术协会（Association for Manufacturing Technology，AMT）和机器人工业联合会（Robotic Industries Association，RIA）共同制定的标准。

通过风险分析/评估对机械潜在的危险进行评估。根据 NFPA 79、ANSI/RIA 15.06、ANSI B11.TR-3 和 SEMI S10（半导体），风险分析为一项重要要求。借助风险分析后记录下的结果，并根据实际应用的安全等级可以选择合适的安全技术。

日本的情况与欧洲和美国不同。它没有和欧洲类似的对功能安全性的法律要求。同样，产品责任法的效力也不及美国。

在日本没有对标准应用的法律要求，但是有使用 JIS（日本工业标准）的管理建议：日本借鉴欧洲的方案，将基本标准作为国家标准（见下表）。

ISO/IEC 编号	JIS 编号	注释
ISO12100-1	JIS B 9700-1	原 TR B 0008
ISO12100-2	JIS B 9700-2	原 TR B 0009
ISO14121- 1/EN1050	JIS B 9702	
ISO13849- 1	JIS B 9705-1	
ISO13849- 2	JIS B 9705-1	
IEC 60204-1	JIS B 9960-1	无附录 F 或欧洲前言中的路线图
IEC 61508-0 至 -7	JIS C 0508	
IEC 62061		尚无 JIS 编号

除了指令和标准中的要求外，还须遵循企业特定的要求。特别是大型企业，例如汽车制造商，对自动化部件有很高的要求，并常会将其作为自己的设备规范列出。

和安全相关的事项（例如运行方式、进入危险区域的操作，急停方案等）必须事先与客户说明，确保这些事项纳入到风险评估/降低过程中。

26.1.4 日本的机械安全

日本的情况与欧洲和美国不同。它没有和欧洲类似的对功能安全性的法律要求。同样，产品责任法的效力也不及美国。

在日本没有对标准应用的法律要求，但是有使用 JIS（日本工业标准）的管理建议：日本借鉴欧洲的方案，将基本标准作为国家标准（见下表）。

ISO/IEC 编号	JIS 编号	备注
ISO12100-1	JIS B 9700-1	原 TR B 0008
ISO12100-2	JIS B 9700-2	原 TR B 0009
ISO14121- 1/EN1050	JIS B 9702	
ISO13849-1	JIS B 9705-1	
ISO13849-2	JIS B 9705-1	

IEC 60204-1	JIS B 9960-1	无附录 F 或欧洲前言中的路线图
IEC 61508-0 至 -7	JIS C 0508	
IEC 62061		尚无 JIS 编号

26.1.5 企业设备规定

除了指令和标准中的要求外，还须遵循企业特定的要求。特别是大型企业，例如汽车制造商，对自动化部件有很高的要求，并常会将其作为自己的设备规范列出。

和安全相关的事项（例如运行方式、进入危险区域的操作，急停方案等）必须事先与客户说明，确保这些事项纳入到风险评估/降低过程中。

26.2 SINAMICS Safety Integrated 概述

Safety Integrated 功能 - STO

安全扭矩停止（STO）功能是符合 EN 60204-1:2006 章节 5.4、可防止意外启动的安全功能。

在高要求的运行状态下，STO 功能符合 IEC 61508 安全集成等级 SIL2、ISO 13849-1 3 类、ISO 13849-1:2015 性能等级（PL）d 标准和 IEC 61800-5-2 的功能。

控制 STO 功能

STO 功能可通过终端控制。关于 STO 接线的更多信息，请参见《SINUMERIK 808D ADVANCED 调试手册》中的“连接 24 V 电源/STO - X6”章节。

26.3 系统特性

26.3.1 认证

SINAMICS V70 驱动系统的安全功能满足以下要求：

- ISO 13849-1:2015 的 3 类
- EN ISO 13849-1:2015 性能等级（PL）d
- IEC 61508 安全集成等级 2（SIL 2）

此外，SINAMICS V70 安全功能已经过独立机构认证。最新的认证组件列表可从当地的西门子办事处获取。

26.3.2 安全说明

说明

本章节中未提及的安全提示和遗留风险请参见 SINUMERIK 808D ADVANCED 调试手册。

⚠危险

通过 Safety Integrated 可降低机器与设备的风险

通过 Safety Integrated 可降低机器与设备的风险。
但是只有设备厂商严格遵守下述要求时，配备 Safety Integrated 的机械与设备才能确实安全可靠地运行：

- 机床制造商（OEM）熟读并遵守本用户技术文档，特别是其中指出的先决条件、安全提示和遗留风险。
- 机床制造商（OEM）认真地完成机器与设备的设计、安装和配置工作。安排专业人员严格执行验收测试、记录测试结果。
- 机床制造商（OEM）对系统与机器进行风险分析后，通过 Safety Integrated 功能或其他途径实施并验证分析得出的所有必需安全措施。

请注意，Safety Integrated 功能并不能取代欧盟机械指令中所要求的、应由设备厂商对机械或设备执行的风险评估。
除了使用 Safety Integrated 功能外，其他的风险防范措施也不可或缺。

⚠警告

启动过程中 Safety Integrated 功能不生效会导致生命危险

只有在系统完全启动后才能激活 Safety Integrated 功能。系统启动是存在较大风险的临界运行状态。若发生事故，可能导致人员死亡或重伤。

- 系统启动过程中彻底远离危险区域。
- 对于垂直轴，检查并确保驱动处于无扭矩状态。

⚠警告

EN 60204-1 规定

急停功能必须通过 STO 实现停机。
设备不可在急停后自动重启。
Safety Integrated 功能关闭后，视风险分析的结果而定，必要时刻允许执行自动重启（例外：急停按钮复位时）。例如在防护门关闭后便可自动启动。

⚠警告

在变更或更换硬件和/或软件组件后启动系统会导致生命危险

在变更或更换硬件和/或软件组件后，只有在保护装置关闭后才能启动系统和变频器。未经全面测试的系统变更可能使功能发生意外变化。这可能导致危险区域中的人员发生死亡或重伤。

- 在变更或更换后进行下列测试：
  - 完整验收测试
  - 部分验收测试
  - 简化功能测试
- 在再次进入危险区域前，应在两个方向（+/-）上短暂试运行所有轴，以检测控制性能是否稳定。
- 确保测试过程中危险区域内没有人。
- 开机时，务必确保仅在系统完全启动后才选择 Safety Integrated 功能。

26.3.3 安全功能的故障概率（PFH 值）

故障概率

根据 IEC 61508、IEC 62061 和 ISO 13849-1:2015 的相关规定，安全功能的故障概率必须以 PFH 值（Probability of Failure per Hour 每小时故障概率）的形式指出。安全功能的 PFH 值取决于驱动器器的安全方案、硬件配置以及其它安全功能组件的 PFH 值。

在 SINAMICS V70 驱动器上，PFH 值主要取决于硬件配置（驱动数量、控制方式、使用的编码器的数量等）。它代表了驱动器集成的所有安全功能的故障概率。

您可咨询当地的西门子办事处获取 PFH 值的信息。

26.3.4 响应时间

响应时间是指从终端发出控制指令到响应实际发生之间的时间。STO 功能的最长响应时间为 5 ms。

26.3.5 遗留风险

设备厂商通过故障分析可确定和驱动设备相关的遗留风险。已知的遗留风险如下：

⚠警告

工作原理相关的硬件故障会导致生命危险：PFH 值

在电气系统中，电气工作原理可能会引发一些硬件故障，因此有额外的遗留风险，此风险以 PFH 值的形式表示。

- 在设计机器及需要应用适当对策时考虑这些遗留风险。

⚠ 警告

变频器在两个晶闸管同时故障（过渡层击穿）时发生转动会导致生命危险

逆变器中两个晶闸管（一个在上桥臂，一个在下桥臂）过渡层同时击穿会引起电机短时间运动。若发生事故，可能导致人员死亡或重伤。

- 采取适当措施（例如，使用配有安全监控的制动装置）来防止变频器的意外转动。

26.4 Safety Integrated 基本功能说明

26.4.1 安全扭矩停止（STO）

“Safe Torque Off”（STO）功能可以和设备功能一起协同工作，在故障情况下安全封锁电机的扭矩输出。

选择此功能后，驱动器便处于“安全状态”。“接通禁止”功能将驱动器锁住阻止其重新启动。

该功能的基础是电机模块/功率模块中集成的双通道脉冲清除。

STO 的功能特性

- 该功能为驱动集成功能，即不需要上一级控制。
- 该功能为驱动专用功能，即每个驱动设备都具有该功能，并需要单独调试。
- 在选择 STO 功能后：
  - 可以避免电机意外启动。
  - 通过安全脉冲清除可以安全切断电机扭矩。
  - 在功率单元和电机之间无电气隔离。
- 选择/撤销 STO 会自动应答安全信息。

STO 功能可以用在以下两种场景：驱动需要通过负载扭矩或摩擦力在很短时间内到达静止状态；驱动自由停车不安全。

⚠ 警告

安全功能不生效会导致机器意外转动

电机在电流封锁后可能发生意外转动，例如：电机缓慢停转。

- 采取适当措施以确保电机在该状况下不会意外转动。

说明

抱闸关闭延迟

抱闸关闭信号（低电平）在 STO 触发后的 30 ms 后输出。

使用 STO 功能条件

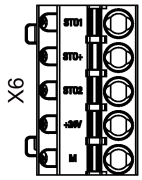
使用 STO 功能时，应满足以下条件：

- 每个监控通道（STO1 和 STO2）都可以通过各自的下电信号通道触发安全脉冲抑制。
- 如果电机连接并配置了抱闸，那么已连接的抱闸是不安全的，因为没有用于抱闸的安全功能，例如安全抱闸。

STO 功能特性

端子		状态	动作
STO1	STO2		
高电平	高电平	安全	伺服驱动上电后伺服电机可正常运行。
低电平	低电平	安全	伺服驱动可以正常启动，但伺服电机不能正常运行。
高电平	低电平	不安全	产生报警，伺服电机自由停车。
低电平	高电平	不安全	产生报警，伺服电机自由停车。

控制电路接口 - 驱动端

类型	示意图	信号	描述
安全扭矩停止（STO）接口		STO 1 STO + STO 2	STO 1：自由停车 STO +：DC 24 V STO 2：自由停车
控制电源输入接口 [1)]		+24 V M	DC 24 V 电源（不带抱闸：-15% 至 +20%，带抱闸：-10% 至 +10%） 0 VDC 电源
最大导线截面：1.5 mm²			

1) 不带抱闸电源和带抱闸电源的最大电流消耗分别为 1 A 和 3 A。

接线

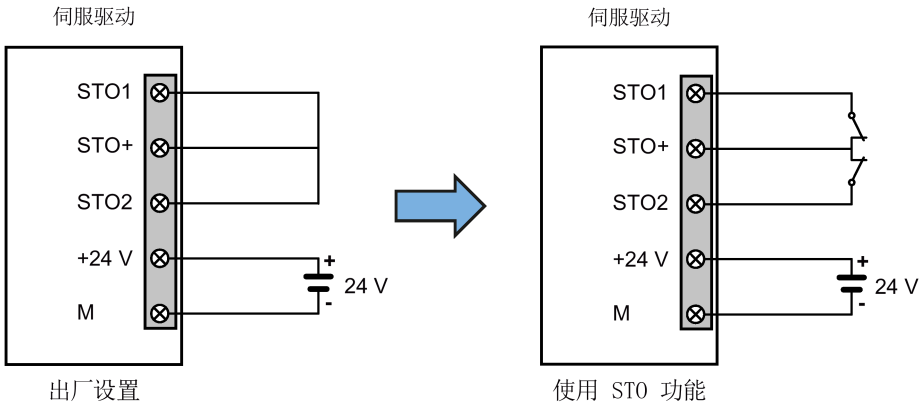
说明

使用 STO 功能

STO1、STO+ 和 STO2 在出厂时是默认短接的。

当需要使用 STO 功能时，连接 STO 接口前必须拔下接口上的短接片。若无需再使用该功能，必须重新插入短接片，否则电机无法运行。

出厂时和使用 STO 功能时的接线如下图所示：



选择/撤销 STO

选择 STO 后会触发以下动作：

- 每个监控通道都通过其断路路径清除脉冲。
- 电机抱闸在连接配置后处于关闭状态。

说明

如果 STO 是在 2 秒内以单通道方式选中并被撤销的，驱动器会清除脉冲而不输出任何信息。

选择 STO 后的驱动器重启

1. 通过输入端子取消各个监控通道的功能。
2. 给出驱动器使能信号。
3. 开启驱动器。
   - 输入信号“ON/OFF1”上输出 0/1 脉冲沿
   - 输入信号“ON/OFF1”上输出 0/1 脉冲沿（接通驱动器）
4. 重新操作驱动器。

STO 的响应时间

STO 功能的最长响应时间为 5 ms。

26.4.2 强制潜在故障检查

Safety Integrated 基本功能的强制潜在故障检查或关机路径测试

断路路径的强制潜在故障检查可以及时识别出两个监控通道中的硬件和软件故障，该功能在选择/取消“Safe Torque Off”时自动执行。

为满足标准 ISO 13849-1:2015 中关于及时发现故障的要求，每隔一段时间就要检查两条关机路径能否正常工作。为此，必须手动或过程自动化地触发强制检查。

定时器可确保强制检查及时执行。

强制检查时间 8760 小时。

此时间届满后驱动器会一直输出相应的报警，只有您完成检查后才会消失。

每次取消选择 STO 时定时器都会复位为设置的值。

假设在运行的设备上已通过相应的安全设施（例如防护门）排除了危险性。因此用户只会收到强制检查到期的报警提示，并被要求在今后的适宜时间执行检查。此报警不会影响设备的运行。

执行强制检查的时间示例：

- 设备上电后驱动器静止时。
- 在防护门打开时。
- 以设定周期。
- 在“AUTO”方式下，根据时间和事件。

说明

如果完成了相应的强制检查，强制检查的定时器也一同清零。相应的报警不会被触发。

STO 的强制检查步骤总是通过终端执行。

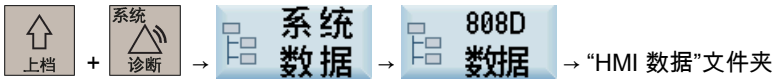
---

27 特殊功能

27.1 对机床制造商 HMI 数据的多语言支持

在数控系统中，可简单实现对机床制造商 HMI 数据的多语言支持。

可通过下列操作步骤访问如下 HMI 数据窗口：



→ “HMI 数据”文件夹

→



对机床制造商 HMI 数据的多语言支持分为下列三种情况：

- 无多语言支持
- 单个文件
- 多个文件

无多语言支持

适用于下列文件：

- 自定义位图
- 用户循环的位图文件
- 用户循环的软键索引文件
- 用户循环的参数文件
- EasyXLanguage 脚本

上述文件不支持多语言，因而在所有语言环境下均使用相同的文件。

单个文件

适用于下列文件：

- 用户循环的告警文件
- 用户扩展文本文件
- OEM 机床数据描述文件
- PLC 报警文本
- OEM R 参数名称文件
- 服务计划任务名称文件

对于上述文件，机床制造商可方便地导入或导出所有语言文件，而无需改变系统语言。不同语言的文件以文件名加以区分。例如，PLC 报警文本的文件名必须为如下格式：

alcu_<LANG>.txt

其中，**<LANG>** 代表实际语言缩写。

说明

文件名中不允许出现大写字母；否则，数控系统无法识别该文件名。

文件名不正确的文件无法被数控系统识别，因而无法在数控系统上生效。

多个文件

适用于下列文件：

- OEM 在线帮助
- OEM 手册
- OEM 幻灯片

对于上述文件，机床制造商可预先在一文件夹中创建某种语言的所有文件，该文件夹名称为 **<LANG>**，其代表实际语言缩写。而后，将创建的文件复制到数控系统上的相应文件夹中。

说明

文件名不正确的文件夹无法被数控系统识别，因而无法在数控系统上生效。

国家代码表

下表给出各种语言代码供参考。

语言	代码	语言	代码
简体中文	chs [1)]	意大利语	ita
繁体中文	cht	韩语	kor
捷克语	csy	荷兰语	nld
丹麦语	dan	波兰语	plk
德语	deu	葡萄牙语	ptb
英文	eng [1)]	罗马尼亚语	rom
西班牙语	esp	俄语	rus
芬兰语	fin	瑞典语	sve
法语	fra	土耳其语	trk
匈牙利语	hun		

1) 交付时已装入数控系统中的默认语言

27.2 调用在线帮助

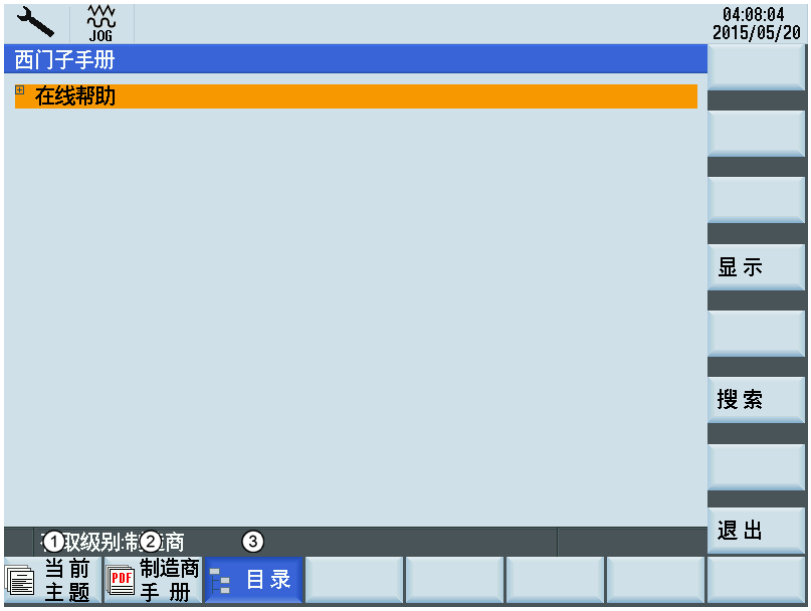
27.2.1 帮助系统

数控系统提供了全面的在线帮助。只要需要您可以从任意操作区域调用帮助系统。

帮助系统



按下此键或<ALT> + <H>组合键可从任意操作区调用帮助系统。如果存在上下文关联的帮助，则窗口“①”打开；否则，窗口“③”打开。



① 调用针对当前主题的上下文关联帮助：

- 当前操作窗口
- 在报警诊断操作区中选中的 NC/V70 报警
- 选中的机床数据或设定数据
- 选中的 V70 参数

② 调用机床制造商编写的 PDF 手册

③ 显示所有可用的帮助信息：

- 西门子帮助手册
- 机床制造商编写的帮助手册（如有该手册）
- 所有可用的 NC/V70 报警
- 所有可用的 V70 参数
- 所有可用的机床数据或设定数据

窗口“①”中的软键说明

	在主题层级中向上导航
	在主题层级中向下导航
到主题	选择交叉索引 交叉索引通过"≫ ... ≪"符号来表示。 **说明：** 该软键仅在当前页面包含交叉索引时显示。
显示	在当前主题窗口中打开选中主题 功能同按下以下键：
	
搜索	在当前主题中搜索特定语句
继续搜索	继续搜索符合搜索条件的下一项
退出	退出帮助系统

窗口“②”中的软键说明

缩放+	当前视图放大
缩放-	当前视图缩小
缩放宽度	缩放当前视图至页面宽度
跳转	跳转到指定页面

搜索	在当前主题中搜索特定语句
继续搜索	继续搜索符合搜索条件的下一项
退出	退出帮助系统

窗口“③”中的按键说明

	扩展主题层级
	收起主题层级
	在主题层级中向上导航
	在主题层级中向下导航
到主题	选择交叉索引 交叉索引通过"≫ ... ≪"符号来表示。 **说明：** 该软键仅在当前页面包含交叉索引时显示。
显示	在当前主题窗口中打开选中主题 功能同按下以下键：
	
搜索	在当前主题中搜索特定语句
继续搜索	继续搜索符合搜索条件的下一项
退出	退出帮助系统

27.2.2 调用机床制造商在线帮助及手册

机床制造商在线帮助

也可在文本文件中创建您自己的在线帮助，并通过 USB 存储器将其上载至数控系统。

创建您自己的在线帮助时，必须使用现有帮助文件格式，如下所示：

#{XE"Chapter 2"} #{HL1} Chapter 2	> 创建将显示于帮助内容列表中的书签。 > 定义深度为 1 的标题。
#{HL2} How to find the Online Help?	> 定义深度为 2 的标题。
You can find the Online Help by pressing the HELP key: #{BITMAP"Images/1.bmp"}	> 标题下面的文本。 >从文件夹"图像"插入位图 1 到文本。
#{HL2} How to open the Online Help? 1. Press the right arrow key, and you can view the Online Help you have created. 2. Press the INPUT key or softkey "Show" to show it.	
#{XREF"Chapter 1"}{text 1}{Go to Chapter 1}	> 创建超链接到章节 1。

说明

必须通过按下回车键来结束文本；否则在线帮助不能正常工作。

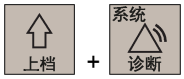
下表提供了关于可用于帮助文本的命令的详细信息：

指令	描述
#{XE "**BookmarkName**"}	创建名为**书签名**的书签。该命令必须跟随一个 HL 命令，该 HL 命令用作帮助索引中的描述。这些书签将显示于帮助内容列表中。
#{HL**{depth}**} {**depth**} = 1 - 5	定义标题。参数{depth}定义了标题深度。
#{NPAGE}	开始新的帮助页面。
#{BOOKMARK "BookmarkName"}	设置名为**书签名**的隐藏书签，该书签出现在帮助索引中。其可用于 **XREF** 命令中来创建一个超链接。
#{XREF "BookmarkName"}{file name}{Display text of hyper-link}	在帮助文本中创建一个超链接。目的文件**书签名**可为通过 **BOOKMARK** 或 **XE** 命令创建的书签。
#{BITMAP "no_ref.bmp"}	在文本中插入位图。
#{SCOLOR **{color}**} {color} = RED, ORANGE, BLACK, BLUE, GREEN, YELLOW, WHITE	将以下文本颜色变为指定颜色。

调用机床制造商在线帮助

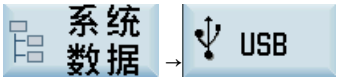
1.
   - 在 PC 上创建在线帮助文件。文件格式可以是 **.txt**, **.png**, 以及 **.bmp**。
   - 将文件按以下目录结构保存于 USB 存储器中：
     - 车削版：...\<LANG>\turning\manual
     - 铣削版：...\<LANG>\milling\manual

     其中，<LANG>代表在线帮助的语言代码。更多信息请参见章节"对机床制造商 HMI 数据的多语言支持 (页 262)"。
2. 将 USB 存储器插入 PPU 前面板上的 USB 接口。



3. 选择系统数据操作区域。

4. 通过以下软键操作打开“USB”窗口。





5. 按下该硬键选择所需<LANG>文件夹，而后按下下述软键复制所选文件夹：

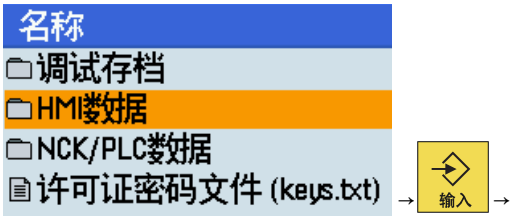


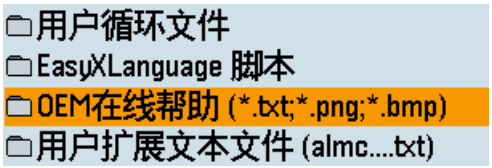


按下该软键打开系统数据窗口。



6. 通过以下操作找到存放在线帮助文件的文件夹，并按该键打开：



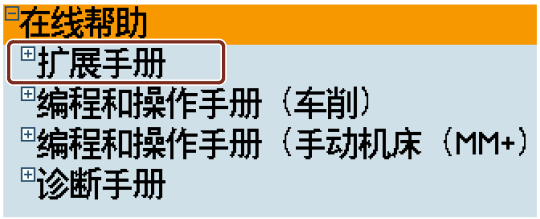


7. 使用该软键粘贴已复制的在线帮助文件。



8. 然后便可通过以下操作查看您自己的在线帮助：





有关此窗口中按键操作的详细信息，请参见“帮助系统 (页 264)”章节。

9. 退出在线帮助。



：返回至调用在线帮助之前所在的操作区域

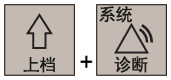


：返回至在线帮助主菜单。

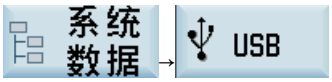
调用机床制造商手册

1.
   - 在 PC 上创建制造商手册的文件。文件名必须为 **oemmanual.pdf**。
   - 将文件按以下目录结构保存于 USB 存储器中：
     – 车削版：...\<LANG>\turning\manual
     – 铣削版：...\<LANG>\milling\manual

     其中，<LANG>代表在线帮助的语言代码。更多信息请参见章节“对机床制造商 HMI 数据的多语言支持 (页 262)”。
2. 将 USB 存储器插入 PPU 前面板上的 USB 接口。



3. 选择系统数据操作区域。
4. 通过以下软键操作打开“USB”窗口。





5. 按下该硬键选择所需<LANG>文件夹，而后按下下述软键复制所选文件夹：

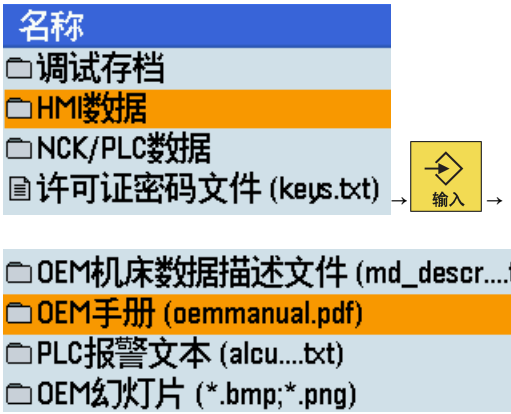




按下该软键打开系统数据窗口。



6. 通过以下操作找到存放制造商手册文件的文件夹，并按该键打开：





7. 使用该软键粘贴已复制的制造商手册文件。
8. 通过以下操作查看您自己的制造商手册：



有关此窗口中按键操作的详细信息，请参见“帮助系统 (页 264)”章节。

9. 退出手册。



注意
系统性能不良 • 不要上载尺寸过大的机床制造商文件；否则，系统性能会降低。

27.3 调用具备辅助功能的标准循环

可以调用具备 M 代码或 T 代码的用户循环。可使用该功能执行换刀等的操作。

说明

用于调用用户循环的 M 代码或 T 代码**不能**存在于同一程序段中。

调用具备"M6"的循环

配置下表中所展示的参数以激活用于调用标准循环的 M 代码：

编号	名称	单位	值	描述
22550	TOOL_CHANGE_MODE	-	1	使用 M 代码来激活刀具参数
22560	TOOL_CHANGE_M_CODE	-	206	用于激活刀具参数的 M 代码
10715	M_NO_FCT_CYCLE[0]	-	6	使用 M06 来调用标准循环
10716	M_NO_FCT_CYCLE_NAME[0]	-	"TOOL"	标准循环的名称

关于标准循环的格式，请参见下例：

```
%_N_TOOL_SPF
;$PATH=/_N_CUS_DIR
PROC TOOL SAVE DISPLOF
IF $P_ISTEST GOTOF _END
IF $P_SEARCH<>0 GOTOF _END
IF $P_TOOLNO==$P_TOOLP GOTOF _NO
G500 D0
G75 Z=0
SPOS=$MN_USER_DATA_FLOAT[0]
MSG("Ready to change tool*** Original tool number: T"<<$P_TOOLNO)
M206
STOPRE
G153 G01 Z0 F2000 ;G153
MSG("Ready to change tool *** Original tool number: T"<<$P_TOOLP)
GOTOF _END
_NO:
MSG("No action *** Reason: programming tool number = spindle tool number")
_END:
M17
```

通过"T"功能调用循环

配置下表中所展示的参数以激活用于调用标准循环的 T 代码：

编号	名称	单位	值	描述
22550	TOOL_CHANGE_MODE	-	0	使用 M 代码来激活刀具参数
10717	T_NO_FCT_CYCLE[0]	-	"TOOL"	使用 M06 来调用标准循环

标准循环的格式与 M 代码的格式相同。用于编程的刀具编号将保存于系统变量 $C_T 中。

频繁使用的系统变量的描述

变量	说明
$P_ISTEST	程序测试状态；布尔变量
$P_SEARCH	程序查找状态；布尔变量
$P_SEARCHL	程序查找状态；实数：1-, 2-, 3-
$P_TOOLNO	主轴刀架中的刀具编号
$P_TOOLP	编程刀具编号
$C_T	编程刀具编号。 当程序代码 T 调用使用 MD10717 定义的换刀循环时 $P_TOOLP 无效。随后用"$C_T"表示刀具编号。
$TC_DP1[Tool number, 1]	刀具类型

变量	说明
$TC_DP3[Tool number, 1]	刀具的几何参数：刀具长度 1
$TC_DP6[Tool number, 1]	刀具的几何参数：刀具半径
$TC_DP12[Tool number, 1]	刀具磨损：长度 1 方向
$TC_DP15[Tool number, 1]	刀具磨损：半径方向
$TC_DP24[Tool number, 1]	刀具尺寸： 0：正常 1：过大
$TC_DP25[Tool number, 1]	刀架编号
_TM[n]	全局用户数据（整型）
_ZSFR[n]	全局用户数据（浮点） **注意：** 由于此数据已用于西门子标准技术循环，所以请务必在使用此数据时确保不存在与技术循环之间的冲突。

27.4 显示功能

显示时间计数器

提供时间计数器用于数控系统计算下列时间段：

时间段	系统变量	描述
总运行时间	$AC_OPERATING_TIME	“AUTO”方式下运行程序的总时间
程序运行时间	$AC_CYCLE_TIME	所选程序的运行时间
进给时间	$AC_CUTTING_TIME	所选程序的进给时间（G01、G02 和 G03）
冷启动后的时间 1)	$AN_SETUP_TIME	从最末次使用默认值启动开始所消耗的时间
热启动后的时间 1)	$AN_POWERON_TIME	从最末次正常启动开始所消耗的时间
剩余时间 2)	-	用于运行当前程序的剩余时间。

1) 冷启动后的时间与热启动后的时间均系在控制器启动之后自动计算。

2) 剩余时间没有相应系统变量，且仅可在零件程序的成功运行周期之后计算。

默认状态下，显示总运行时间，程序运行时间、冷启动后的时间和热启动后的时间。进给时间仅可在通过 MD27860 激活之后计算：

编号	名称	值	描述
27860	PROCESSTIMER_MODE	实际值	计算下列程序运行时间段的激活操作： • 总运行时间 • 程序运行时间 • 进给时间

调用零件计时器的操作如下：



1. 选择偏置操作区域。



→



2. 接连按下这两个软键。系统显示时间计数器，以计算下列时间段：

总运行时间	0000 H 00 M 00 S
程序运行时间	0000 H 00 M 00 S
进给时间	0000 H 00 M 00 S
缺省设置后的启动时间	0002 H 18 M
冷启动时间	0002 H 18 M

或



1. 进入“AUTO”运行方式并选择加工操作区域。



2. 按下该软键后，系统显示时间计数器，以计算下列时间段：

程序	0H00M00S
程序剩余时间	0H00M00S

显示工件计数器

提供工件计数器用于数控系统计算下列零件数目：

零件	**系统变量**	**描述**
需要的零件	$AC_REQUIRED_PARTS	待被计数的所需零件。 通过将 MD27880 位 0 设为 1 来激活： • 位 1 = 0：如果“**零件数**” = “**需要的零件**”，则报警或接口信号 DB3300.DBX4001.1 = 1
零件总数	$AC_TOTAL_PARTS	所计零件总数。 通过将 MD27880 位 4 设为 1 来激活： • 位 5 = 0：M02/M30 将“**零件总数**”增加至“**1**” • 位 5 = 1：由 MD27882 定义的 M 代码将“**零件总数**”增加至“**1**” • 位 6 = 0/1：当“**程序测试**”无效时，计数器不工作。
零件数	$AC_ACTUAL_PARTS	实际所计零件数。 通过将 MD27880 位 8 设为 1 来激活： • 位 9 = 0：M02/M30 将“**零件总数**”增加至“**1**” • 位 9 = 1：由 MD27882 定义的 M 代码将“**零件总数**”增加至“**1**” • 位 10 = 0/1：当“**程序测试**”无效时，计数器不工作。

编号	**名称**	**值**	**描述**
27880	PART_COUNTER	实际值	配置并激活工件计数器
27882	PART_COUNTER_MCODE	实际值	定义用于计数动作的 M 代码：0 至 99

调用工件计数器的操作如下：



1. 选择偏置操作区域。



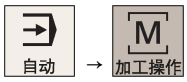
→



2. 接连按下这两个软键。系统显示工件计数器，以计算下列零件数目：



或



1. 进入“AUTO”运行方式并选择加工操作区域。



→



2. 接连按下这两个按键。
系统显示工件计数器，以计算下列零件数目：



27.5 Prog_Event 功能

借助 Prog_Event 功能，名为“CYCPE1MA.SPF”和“CYCPE_MA.SPF”的两个异步子程序被触发以在诸如程序末尾、NC 复位等的状态下执行。必须先创建 CYCPE1MA.SPF 和 CYCPE_MA.SPF 中至少一个文件，而后将其保存在循环目录（N:\CMA）下。

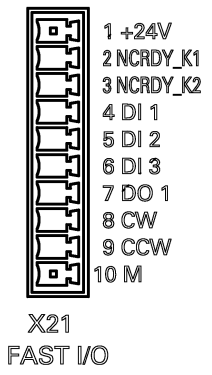
相关参数：

编号	名称	值	说明
11450	SEARCH_RUN_MODE	7H	-
20106	PROG_EVENT_IGN_SINGLEBLOCK	1FH	-
20107	PROG_EVENT_IGN_INHIBIT	CH	-
20108	PROG_EVENT_MASK	实际值	用于 N: \CMA\CYCPE1MA.SPF 和 N: \CMA\CYCPEMA.SPF 的触发模式： • 位 0:在 NC 调试时激活程序事件 • 位 1:在 NC 程序末尾激活程序事件 • 位 2:通过 RESET 键来激活程序事件 • 位 3:在启动 NC 后激活程序事件
20109	PROG_EVENT_MASK_PROPERTIES	1H	-

27.6 快速输入/输出

硬件描述

FAST I/O 接口（X21）提供 3 个数字量输入和 1 个数字量输出：

图示	针脚	信号	说明	变量
	4	DI1	快速输入 1，地址为 DB2900.DBX0.0	$A_IN[1]
	5	DI2	快速输入 2，地址为 DB2900.DBX0.1	$A_IN[2]
	6	DI3	快速输出 3，地址为 DB2900.DBX0.2	$A_IN[3]
	7	DO1	快速输出 1，地址为 DB2900.DBX4.0	$A_OUT[1]

相关参数

MD 号	名称	含义	值
10366	HW_ASSIGN_DIG_FASTIN[0]	快速输入的硬件分配	10101
10368	HW_ASSIGN_DIG_FASTOUT[0]	快速输出的硬件分配	10101

PLC 接口地址

DB2900	来自快速输入和输出的信号							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
0						输入 3	输入 2	输入 1
4								输出 1

快速输入/输出的应用

快速输入

在 PLC 应用程序中，可以从地址 **DB2900.DBX0.0** 直接读取每一位的数值。

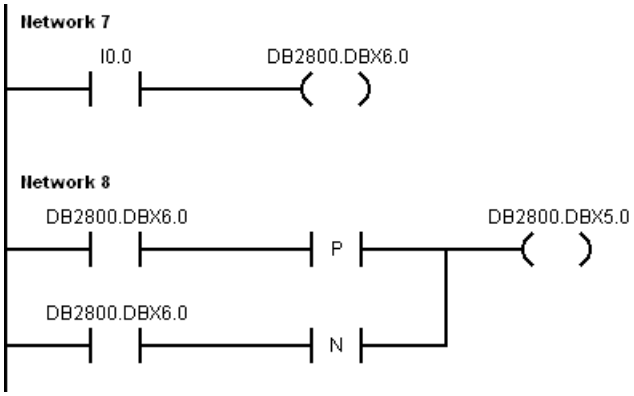
在零件程序中，可以通过相应系统变量从地址 DB2900.DBX0.0 读取每一位的数值。

快速输出

不可从地址 **DB2900.DBX4.0** 分配数值到快速输出；否则，PLC 应用程序会因错误而停止。 然而，可以从地址 **DB2800.DBX5.0** 和 **DB2800.DBX6.0** 分配数值到快速输出。

在 PLC 应用程序中，可以通过地址 **DB2800.DBX6.0** 处的上升沿或下降沿触发地址 **DB2800.DBX5.0**，且因此地址 **DB2900.DBX4.0** 会随地址 **DB2800.DBX6.0** 而变化。

举例而言，如要使用 **I0.0** 来触发或取消地址 **DB2900.DBX4.0** 的设置/重设，则可在 PLC 应用程序中如下进行写入：



在零件程序中，可以通过其相应变量对快速输出进行设置或重设。 系统变量为 **$A_OUT[1]**。

例如：

```
$A_OUT[1]=1  ; 设置 DB2900.DBX4.0 = 1
M30
```

27.7 创建用户循环

数控系统中集成了标准西门子循环。如果必要，您也可以创建自己的循环。

要创建自定义循环，必要准备以下文件：

- 用户循环文件
- 用户循环报警文件
- 用户循环位图文件
- 用户扩展文本文件
- 用户循环软键索引文件
- 用户循环参数文件

27.7.1 创建用户扩展文本文件

显示相应窗口文本、循环消息和软键文本需要用户扩展文本文件。

命名规则

almc_<LANG>.txt

此处“**<LANG>**”表示语言名称，例如，eng。

更多信息请参见章节“对机床制造商 HMI 数据的多语言支持 (页 262)”。

文本定义规则

定义文本时，必须遵循以下规则：

<Identifier> "<Text>" // <lines*chars>

- <Identifier>:在此处用数字定义软键、消息或屏幕的标识符。编号范围为 83000 到 84999。
- <Text>: 在此处定义实际显示的文本，例如软键名、消息文本、参数描述文本。
- <lines*chars>: 在此处设置 GUI 上文本字符数和行数的可用长度。您可以通过插入字符"**%n**"来开始新的一行。软键文本最多允许 2 行，每行各 9 个字符。

示例

83000 "User%nCycles" // 2*9 ⇒ 两行，每行九个字符长度

83001 "CYCLE100" // 9 ⇒ 一行九个字符长度

83020 "DIA short description" // 21 ⇒ 一行 21 个字符长度

27.7.2 创建用户循环的软键索引文件

用户循环的软键索引文件 (cov.com) 须定义用户循环的软键。您可以通过如 WordPad 或 Notepad 之类的文本编辑器来创建 cov.com 文件。

文本定义规则

Sx.y.z\$identifier\bitmap(cycle)

参数	取值范围	含义
x	5	第五个水平软键。
y	1 至 8	第一列的第一到第八个垂直软键。

参数	取值范围	含义
z	1 至 8	第二列的第一到第八个垂直软键。
\$identifier\	-	用户扩展文本文件中定义的软键标识符。
bitmap(cycle)	-	用于循环界面显示的位图。位图名称之后必须带有用户循环的名称，该循环名称会显示在循环界面的左上角。

示例

```
S5.0.0\$83000\ ⇒ 在第五个水平键位置定义软键（标识符为 83000）。
S5.1.0\$83001\CN1(CYCLE100) ⇒ 在按下第五个水平键后的第一级界面的第一个垂直键位置定义软键（标识符为 83001）；CN1.bmp 用作 CYCLE100 的界面显示。
S5.2.0\$83002\CN2(CYCLE101) ⇒ 在按下第五个水平键后的第一级界面的第二个垂直键位置定义软键（标识符为 83002）；CN2.bmp 用作 CYCLE101 的界面显示。
M17
```

27.7.3 创建用户循环的参数文件

用户循环的参数文件 (sc.com) 须定义用户循环的帮助信息和参数。您可以通过如 WordPad 或 Notepad 之类的文本编辑器来创建 sc.com 文件。

文本定义规则

符号"//"表示循环描述的开始。

如果在循环开始时您已经在窗口左侧创建了图形，调用第一行的图形。该图形后面接着带括号的循环名称。

现在可以根据下表中的格式为各个变量定义参数。

位	参数描述	条目
1	变量定义开始	(
2	变量类型	R - REAL I - INTEGER C - CHAR S - STRING
3	分隔符	/
4	• 最小值 + 长度 + 最大值 • * + 可供选择的值	• 最小值 + 长度 + 最大值 • * + 不同的数值（用空格隔开） 注意您还可以为字符定义不同的图片。
5	分隔符	/
6	缺省值	如此处未输入，则循环中不显示缺省值。
7	分隔符	/
8	参数短描述	$ + 标识符（所选参数的短描述，显示在参数界面的左上角；在用户扩展文本文件中定义）
9	变量定义结束	)
10	开始描述	[
11	参数名称	输入栏前的文本；最大长度为五个字符
12	描述结束	]
13	参数图片	/B + bmp 文件名（不带文件扩展名）

说明

分隔符，在开始和结束识别时都必须使用。

位 4、6 和 13 可以空缺。

如果在 $ 标识符中没有保存任何文本，则在窗口的相应区域上会显示三个问号。

示例

```
//CN1(CYCLE100)
(R/0 99999.999//$83020)[DIA]
(R/0 99999.999//$83021)[DIAF]
(R/-9999.999 99999.999//$83022)[STAP]
(R/-9999.999 99999.999//$83023)[ENDP]
(R/0 99999.999//$83024)[MID]
(R/0 99999.999//$83025)[UX]
(I/*0 1 2/0/$83026)[MACH]/B CN1
(R/1 99999.999/1/$83027)[VRT]
M17
```

27.7.4 创建用户循环文件

您可以根据不同的加工功能创建用户循环文件。它是用于调用循环的子程序。

命名规则

CYCLExxx.SPF

此处"xxx"表示循环号。循环号**不能**超过四位数。

说明

用户循环的名称**不可以**与标准西门子循环相同。建议使用 100 到 800 循环编号。

程序举例

通过 wordpad 或 notepad 创建程序。

由于循环窗口总是会将用户循环编号作为调用参数传输给用户循环，所以传输接口要按如下方式定义。

```
PROC CYCLE100(REAL DIA,REAL DIAF,REAL STAP,REAL ENDP,REAL MID,REAL UX,INT MACH,REAL VRT)
SAVE SBLOF DISPLOF
```

PROC 是关键字，后面是循环名称和编号。括号中是窗口的所有传输参数，数据类型和名称用逗号隔开。

```
PROC CYCLE100(REAL DIA,REAL DIAF,REAL STAP,REAL ENDP,REAL MID,REAL UX,INT MACH,REAL VRT)
SAVE SBLOF DISPLOF
DEF REAL VAR1
IF $P_EP[X]<DIA GOTOF LL1
LL3:
IF DIAF>DIA GOTOF END2
START:
IF MACH==0 GOTOF ROUGHING1
IF MACH==1 GOTOF FINISHING
IF MACH==2 GOTOF ROUGHING1
DEF REAL VAR1
ROUGHING1:
R101=(DIA-DIAF)/2-UX
R102=R101/MID
R103=TRUNC(R102)
R104=0
VAR1=DIA
IF R103<=1 GOTOF ROUGHING2
LL2:
SBLON
G90 G0 X=VAR1 Z=STAP+2
G1 Z=ENDP
```

```
G91 X=MID
G0 G91 X=VRT Z=VRT
G90 G0 Z=STAP+2
SBLOF
VAR1=VAR1-2*MID
R104=R104+1
IF R104<=R103 GOTOB LL2
IF R104>R103 GOTOF ROUGHING2
ROUGHING2:
SBLON
G90 G0 X=DIAF+UX
G1 Z=ENDP
G0 G91X=VRT Z=VRT
G90 G0 X=DIA+2
Z=STAP+2
IF MACH==2 GOTOF FINISHING
SBLOF
RET
FINISHING:
SBLON
G0 X=DIAF
G1 Z=ENDP
G1 X=DIA+VRT
G0 G91X=VRT Z=VRT
G90 Z=STAP+2
SBLOF
RET
LL1:
IF $P_EP[Z]<STAP GOTOF END1
GOTOB LL3
END1:
SETAL(65000)
STOPRE
M0
RET
END2:
SETAL(65001)
STOPRE
M0
RET
```

27.7.5 创建用户循环的告警文件

用户循环的告警文件必须显示用户循环的报警号和报警消息。

命名规则

alc_<LANG>.txt

此处“**<LANG>**”表示语言名称，例如，eng。

更多信息请参见章节“对机床制造商 HMI 数据的多语言支持 (页 262)”。

文本定义规则

定义文本时，必须遵循以下规则：

<AlarmNumber> "<Text>" // <lines*chars>

- <AlarmNumber>: 在此处定义报警号。编号范围为 65000 到 69999。
- <Text>: 在此处定义实际报警文本。
- <lines*chars>: 在此处设置 GUI 上文本字符数和行数的可用长度。您可以通过插入字符"**%n**"来开始新的一行。

示例

65000 "Current tool position is incorrect" // 34 ⇒ 一行 34 个字符长度

65001 "DIAF is bigger than DIA" // 23 ⇒ 一行 23 个字符长度

27.7.6 创建用户循环的位图文件

循环图标 **必须**存储为最大像素为 **224 * 224** 的 **16** 色位图文件(*.bmp)。

图标名**必须**以大写/小写的"**C**"开头，且包括文件扩展名在内**不能**超过 **32** 个字符（例如 CN1.bmp）。

说明

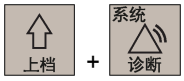
如果 16 色无法满足显示，也可以使用 24 色位图。

27.7.7 传输所需的文件至数控系统

执行以下操作将所需文件传输至数控系统。

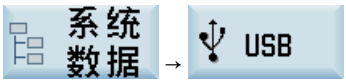
先决条件：将包含所有所需文件的 USB 存储器插入 PPU 前面板上的 USB 接口。

传输用户循环文件



1. 选择系统数据操作区域。

2. 通过以下软键操作打开 USB 存储目录。





3. 选择所需的用户循环文件（例如，CYCLE100.SPF）然后按下该软键复制文件。



4. 按下该软键打开用户循环目录。



5. 按下该软键粘贴已复制的文件。

传输位图文件



1. 按下该软键切换至 USB 存储目录。



2. 选择所需的位图文件（例如，cn1.bmp、cn2.bmp...），然后按下该软键复制文件。
请注意，您还可以使用以下按键选择多个文件：

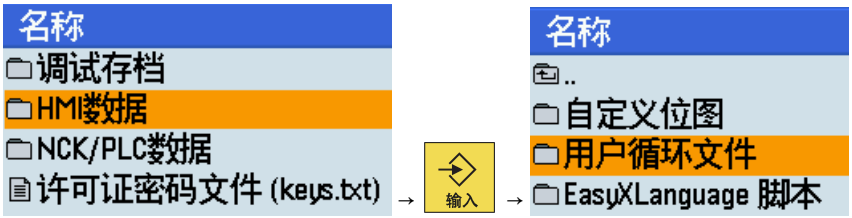


808D 数据

3. 按下该软键打开系统数据窗口。

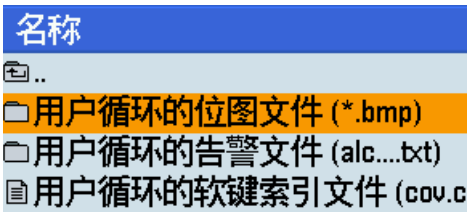
输入

4. 通过以下操作找到存放用户循环文件的文件夹，并按该键打开：



输入

5. 按下该键打开如下用户循环位图文件夹：



粘贴

6. 使用该软键粘贴已复制的用户循环位图文件。

传输用户循环报警文件

USB

1. 按下该软键切换至 USB 存储目录。

复制

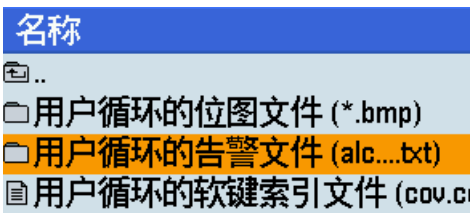
2. 选择用户循环报警文件（例如，alc_eng.txt），然后按下该软键复制文件。

808D 数据

3. 按下该软键打开系统数据窗口。

输入

4. 找到如下用户循环报警文件夹并按该键打开。



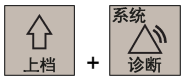
粘贴

5. 使用该软键粘贴已复制的用户循环报警文件。

确认

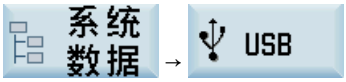
6. 当出现系统消息提示您重启 HMI 时按下该软键确认。HMI 成功重启后即可激活新数据。

传输 cov.com 文件和 sc.com 文件



1. 选择系统数据操作区域。

2. 通过以下软键操作打开 USB 存储目录。





3. 选择 cov.com 和 sc.com 文件，然后按下该软键复制文件。
请注意，您还可以使用以下按键选择多个文件：

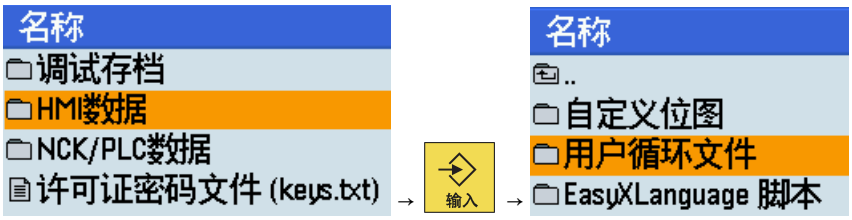




4. 按下该软键打开系统数据窗口。

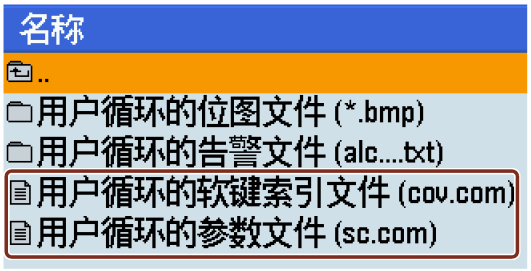


5. 通过以下操作找到存放用户循环文件的文件夹，并按该键打开：





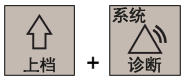
6. 按下该软键以替换如下空文件：





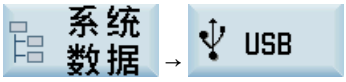
7. 当出现系统消息提示您重启 HMI 时按下该软键确认。HMI 成功重启后即可激活新数据。

传输用户扩展文本文件



1. 选择系统数据操作区域。

2. 通过以下软键操作打开 USB 存储目录。

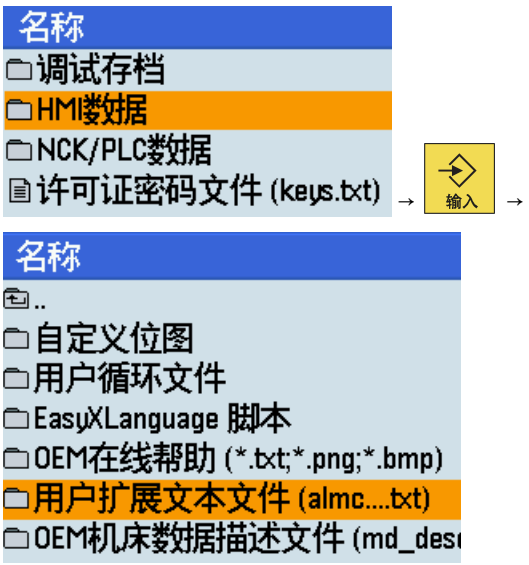




3. 找到用户扩展文本文件（例如，almc_eng.txt），然后按下该软键复制文件。



4. 通过以下操作找到存放用户扩展文本文件的文件夹，并按该键打开：





5. 按下该软键粘贴已复制的用户扩展文本文件。



6. 当出现系统消息提示您重启 HMI 时按下该软键确认。HMI 成功重启后即可激活新数据。

27.7.8 调用已创建的用户循环

将您自己的循环文件都传输至数控系统后，循环即创建成功。随后即可在程序编辑操作区域调用该循环。

按照以下步骤调用已创建的循环，例如，CYCLE100。



1. 选择程序管理操作区域。



2. 选择所需的零件程序，按下该键在程序编辑器中打开该程序。

此时您可以看到在第五个水平软键位置定义的软件，例如：



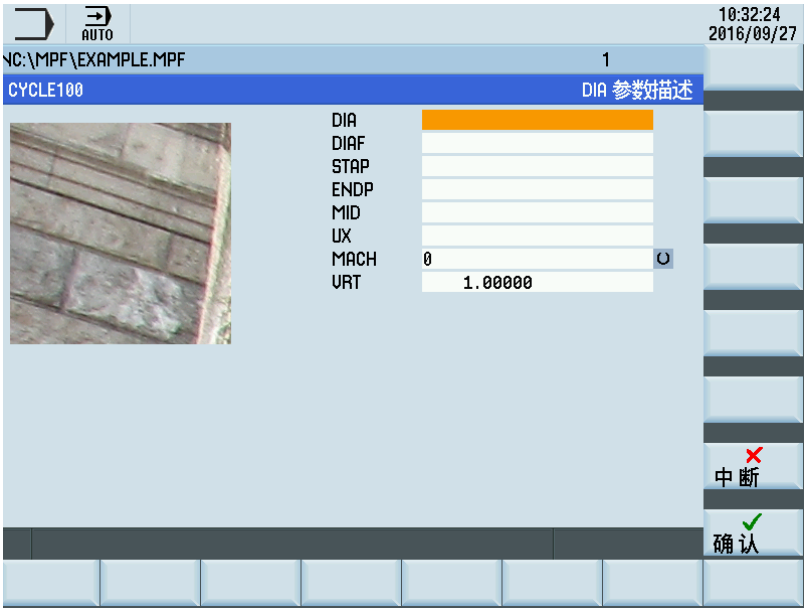
用户循环

3. 按下该软键打开用于调用用户循环的下级菜单，例如：





4. 按下该软键打开 CYCLE100 窗口，例如：





5. 设置所需的参数，然后按下该软键确认并返回程序编辑窗口。此时可看到 CYCLE100 已插入程序中。

27.7.9 编辑用户循环窗口

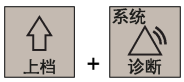
您可以为用户循环编辑软键、标识、位图或参数。

为此可导出相应的文件，并在 PC 上进行编辑。 之后再将其导回至相应的文件夹，然后重启数控系统。

27.8 使用机床制造商的机床数据描述

如果需要，您可以对 PLC 机床数据 14510、14512、14514、14516 和 17400 使用自定义的数据描述。

按以下步骤编辑机床数据描述：



1. 选择系统数据操作区域。



2. 按下该软键，基本 NC 数据窗口默认打开。



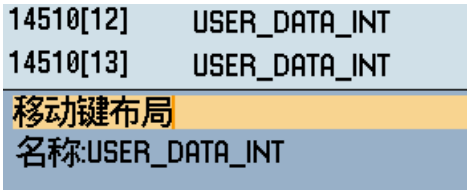
3. 按下该软键，屏幕上会列出所有可编辑的机床数据。



4. 使用光标键选择所需的机床数据。



5. 按下该软键激活屏幕底部的输入框，并输入所需的描述文本，例如：

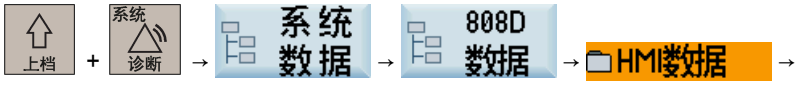




6. 按下该软键或下键确认输入：



在数控系统上修改了至少一个机床数据之后，即可在系统上看到一个 OEM 机床数据描述文件（.txt），该文件中包含在当前系统语言下所编辑的制造商机床数据描述。您可以通过下面的操作查看该文件：





OEM 机床数据描述文件是分语言版本的。您可以使用 USB 存储器将所需语言的文件复制到电脑中进行备份或者对机床数据描述进行批量编辑。

27.9 使用机床制造商 R 参数名称

您可以根据需要自定义 R 参数的名称。

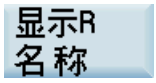
按以下步骤定义 R 参数名称：



1. 选择偏置操作区域。

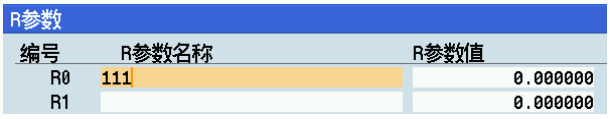


2. 按下该软键打开 R 参数列表。



3. 按下该软键激活所有 R 参数的名称显示。

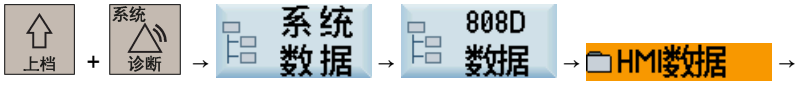
4. 使用光标键选择一个 R 参数，并在名称输入栏输入所需的名称，例如：

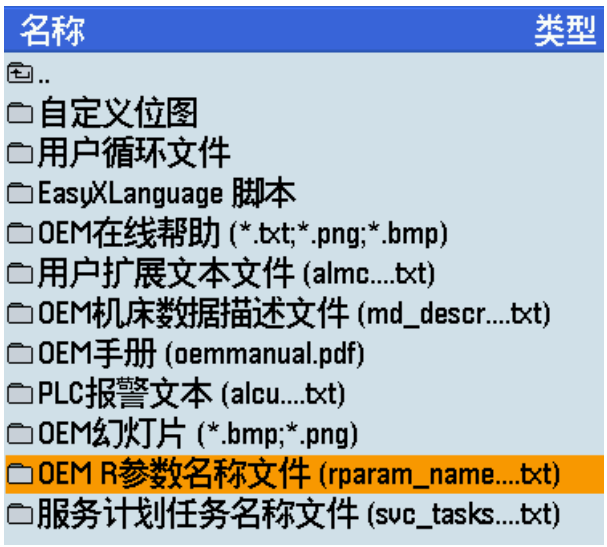




5. 按该键或移动光标确认输入。

在数控系统上定义了至少一个 R 参数名称之后，即可在系统上看到一个 OEM R 参数名称文件（.txt），该文件中包含制造商在当前系统语言下所定义的 R 参数名称。您可以通过下面的操作查看该文件：





OEM R 参数名称文件是分语言版本的。您可以使用 USB 存储器将所需语言的文件复制到电脑中进行备份或者对 R 参数名称进行批量编辑。

27.10 通过自定义 EasyXLanguage 脚本创建用户对话框

27.10.1 功能范围

“创建用户对话框”功能可以提供开放性结构，从而能够让用户在 SINUMERIK 808D ADVANCED 系统上设计应用专用的自定义用户界面。

数控系统提供基于 XML 的脚本语言用于创建用户对话框。



该脚本语言可以在 HMI 上的用户自定义操作区域中显示机床专用菜单和对话框窗口。

所有对话框窗口在设计时均无需考虑语言版本。系统会从一同供货的语言数据库中读取待显示的文本。

使用

已定义的 XML 指令可以实现下列特性：

- 显示对话框并提供：
  - 软键
  - 变量
  - 文本和帮助文本
  - 图形和帮助画面
- 通过以下操作调用对话框：
  - 按下（登入）软键
- 通过以下操作动态重组对话框：
  - 编辑和删除软键
  - 定义并设计变量栏
  - 插入、更换、删除显示文本（和语言相关或无关）

  - 插入、更换、删除图片
- 在进行以下操作时触发动作：
  - 显示对话框
  - 输入数值（变量）
  - 按下软键
  - 关闭对话框
- 交换对话框之间的数据
- 如下操作变量：
  - 读取（NC、PLC 和用户变量）
  - 写入（NC、PLC 和用户变量）
  - 通过数学、比较或者逻辑运算符相连
- 实现以下功能：
  - 子程序
  - 文件功能
  - PI 服务
- 根据用户组考虑保护等级

关于脚本语言的有效元素（标签）的描述可参见章节“XML 标签 (页 290)”。

说明

下列章节关于 XML （Extensible Markup Language 可扩展标记语言）的描述没有完整性要求。更多的信息可查阅相应的专业文献。

27.10.2 配置基础

配置文件

新操作界面的说明存储在配置文件中。这些文件自动编译并显示屏幕上的结果。在供货时并不提供配置文件，因此必须首先创建或下载此文件。

在工具盒的“...\examples\SINUMERIK_808D\easyXL”文件夹中可找到样例配置文件（EasyXLanguage 脚本）。

可以使用 XML 编辑器或其他的文本编辑器来创建配置文件。

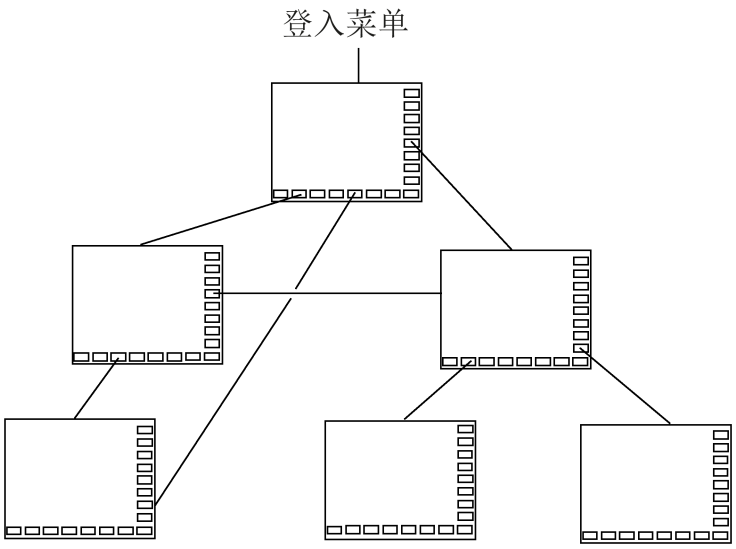
说明

配置文件名不区分大小写。

菜单树的原理

多个相连的对话框构成了一个菜单树。如果能从一个对话框切换入另一个对话框，则表示这两个对话框间存在联系。通过此对话框内重新定义的水平或者垂直软键可以返回上级对话框或者进入任意一个对话框。

可以在登入菜单后面通过配置好的登入软键生成更多的菜单树：



登入菜单

用户自定义

在 PPU 上按下此键即可调用登入菜单，该菜单通过“xmldial.xml”文件中的“main”名称来定义。登入菜单是操作流程自身的输出点。

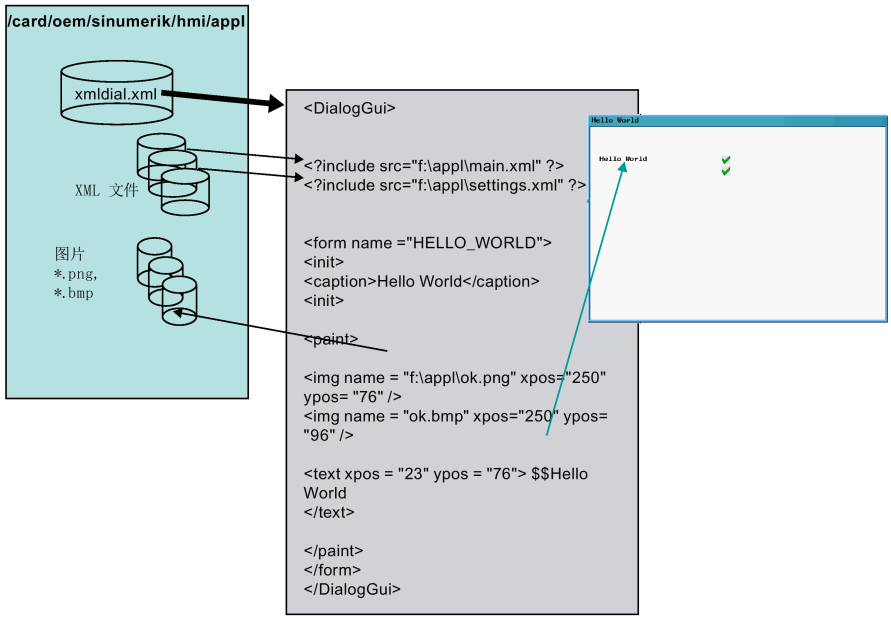
您可以使用“main”菜单将对话框或其他软键栏链接起来，从而可对其进行加载并用其执行更多操作。

27.10.3 配置文件（EasyXLanguage 脚本）

配置用户对话框时需要用到系统中如下制造商文件夹下的文件：

文件类型	文件名称	含义
脚本文件	“xmldial.xml”	该脚本文件使用 XML 标签来控制已配置的软键菜单和对话框窗口在 HMI 上的用户自定义操作区域中如何显示。
位图	例如，“text.bmp”或“text.png”	数控系统支持 BMP 格式和 PNG 格式。
XML 文件，在控制文件“xmldial.xml”中插入了 XML 标签“INCLUDE”。	例如，“machine_settings.xml”	该文件同样包含用来在 HMI 上显示对话框窗口和参数的编程指令。

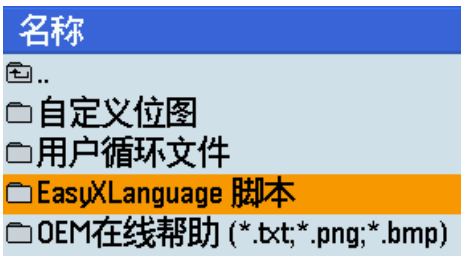
用户对话框配置文件的相关性



载入配置

必须通过 USB 存储器或网络驱动器将生成的配置文件复制到如下路径：





说明

当上述文件夹中存在“xmldial.xml”脚本文件时，即可在用户自定义操作区域中调用登入菜单。

在完成初次复制操作后，必须通过执行如下操作来重启数控系统：



HMI 上用户对话框示例

在调用用户自定义操作区域后可显示已配置的软键菜单。用户可以通过配置好的各个对话框窗口进行操作。



27.10.4 配置文件的结构

概述

配置文件由以下单元组成：

- 对带有登入软键的登入菜单“main”的说明
- 对话框定义
- 变量定义
- 块说明
- 定义软键条

27.10.5 语言相关性

使用和语言相关的文本：

- 软键标记
- 标题
- 辅助文本
- 其它任意文本

27.10.6 XML 标签

27.10.6.1 通用结构

用于对话框配置的脚本文件结构与指令

所有的对话框配置都保存在 **DialogGui** 标签中。

```
<DialogGui>
…
</DialogGui>
```

示例：

```
<?xml version="1.0" encoding="utf-8"?>
<DialogGui>
…
<FORM name ="Hello_World">
<INIT>
<CAPTION>Hello World</CAPTION>
</INIT>
...
</FORM>

</DialogGui>
```

指令

XML 语言提供以下指令用于执行有条件指令和循环控制：

- For 循环
- While 循环
- Do while 循环
- 有条件执行
- Switch 和 case 指令
- 对话框窗口中的操作单元
- 软键说明
- 定义变量

关于指令的详细描述请参见指令/标签说明 (页 291)。

27.10.6.2 指令/标签说明

为了创建对话框和菜单以及处理程序顺序需要定义下列 **XML 标签** ：

说明

应用当前使用的表达式来替换引号"<...>"中的属性值。

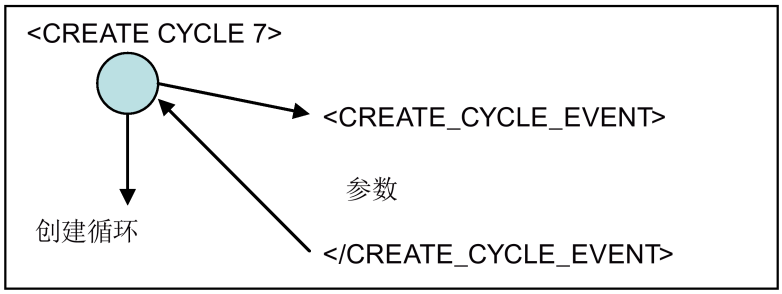
示例：

<DATA_LIST action="read/write/append" id="<list name>">
编程为如下：

<DATA_LIST action="read/write/append" id="my datalist">

标签名称	含义
BREAK	有条件中断一个循环。
CONTROL	该标签用来创建控件元素。
CONTROL_RESET	该标签可以重新启动一个或多个控制器组件。 **句法：** `<CONTROL_RESET resetnc="TRUE" />` **属性：** • RESETNC = "TRUE" 重新启动 NC 组件。

标签名称	含义

CREATE_CYCLE_EVENT	如果分析程序开始处理标签 CREATE_CYCLE，则会首先将 <CREATE_CYCLE_EVENT> 消息发送到有效的 FORM 标签。该消息会在分析程序根据参数表和创建规则生成 NC 指令之前，被用于循环参数的创建准备。
	
	
	句法：
	`<CREATE_CYCLE_EVENT>`
	`…`
	`…`
	`</CREATE_CYCLE_EVENT>`
	
	示例：
	`<SOFTKEY_OK>`
	`…`
	`<CREATE_CYCLE />`
	`…`
	`…`
	`<SOFTKEY_OK>`
	`…`
	`…`
	
	`<FORM>`
	`<NC_INSTRUCTION>MY_CYCLE($P1, $P2)</ NC_INSTRUCTION >`
	`…`
	`…`
	`<CREATE_CYCLE_EVENT>`
	`<type_cast name="P1" type="int"/>`
	
	`<OP> P1 = P1 * 150 </OP>`
	
	`</CREATE_CYCLE_EVENT>`
	`…`
	`…`
	
	`</FORM>`

标签名称	含义
DATA	该标签用于使能 NC、PLC、GUD 和驱动数据的直接写入。 关于地址构成的详细信息可以参见章节“组件地址分配 (页 338)”。 **属性：** • **name** 变量地址 **标签值：** 允许使用所有包括文字和数字的表达方式作为标签值。如要对一个局部变量中的值直接进行写操作，则要使用替代运算符 **$** ，将局部变量名跟在其后。 **句法：** <DATA name="<variable name>"> value </DATA> **示例：** <DATA name = "plc/mb170"> 1 </DATA> … <LET name = "tempVar"> 7 </LET> <!– 局部变量内容"tempVar"写入位存储字节 170 → <DATA name = "plc/mb170">$tempVar</DATA>
DATA_LIST	该标签可以保存或者恢复所列示的驱动和机床数据。 地址逐行列出。关于地址构成的详细信息可以参见章节“组件地址分配 (页 338)”。 可以创建最多 20 个临时数据列表。 **属性：** • **action** *read* – 将列出变量的值存储到临时存储器中 *append* – 将列出变量的值添加到已经存在的列表中 *write* – 将已备份的值复制到相应的机床数据中 • **id** 该标识符用于识别临时存储器 **句法：** <DATA_LIST action="<read/write/append>" id="<list name>"> NC/PLC 地址编译 </DATA_LIST> **示例：** <DATA_LIST action ="read" id="<name>"> nck/channel/parameter/r[2] nck/channel/parameter/r[3] nck/channel/parameter/r[4] $MN_USER_DATA_INT[0] … </ DATA_LIST> <DATA LIST action ="write" id="<name>" />

标签名称	含义
DO_WHILE	Do while 循环 DO 指令 WHILE （测试） **句法：** <DO_WHILE> 指令 … <CONDITION>…</CONDITION> </DO_WHILE> Do while 循环由一个指令程序段和一个条件构成。首先执行指令程序段中的代码，然后运用条件。如果条件为真（true），则再次执行代码部分。一直重复，直至条件为假（false）。 **示例：** <DO_WHILE> <DATA name = "PLC/qb11"> 15 </DATA> <CONDITIION> "plc/ib9" == 0 </CONDITION> </DO_WHILE>
DYNAMIC_INCLUDE	该标签包含一个 XML 脚本文件。 与 INCLUDE 标签相反，读取操作仅在执行相应指令时进行。 在大型项目中该标签的使用能缩短用户区域或循环支持的加载时间。此外，对系统资源的平均需求也会下降，因为在一个会话中不需要始终调用所有的对话框。 **句法：** <DYNAMIC_INCLUDE src="path name"/> **示例：** <SOFTKEY POSITION="3"> <CAPTION>MY_MENU</CAPTION> <DYNAMIC_INCLUDE src/> <NAVIGATION>MY_MENU</NAVIGATION> </SOFTKEY>
ELSE	未满足条件时的指令（**IF（如果）、THEN（则）、ELSE（否则）**）

标签名称	含义
FOR	FOR 循环 for （初始化；测试；继续）指令 **句法：** <FOR> <INIT>…</INIT> <CONDITION>…</CONDITION> <INCREMENT>…</INCREMENT> 指令 … </FOR> For 循环的执行方式如下： 1. 计算表达式**初始化**（INIT）。 2. 将表达式 **测试**（CONDITION）作为布尔型表达式计算。 如果值为假（false），则退出 For 循环。 3. 执行后续的指令。 4. 计算表达式**继续**（INCREMENT）。 5. 从第 2 步继续。 在 INIT、CONDITION 和 INCREMENT 分支中使用的所有变量都要在 FOR 循环外部定义并初始化。 **示例：** <LET name = "count">0</LET> <FOR> <INIT> <OP> count = 0</OP> </INIT> <CONDITION> count <= 7 </CONDITION> <INCREMENT> <OP> count = count + 1 </OP> </INCREMENT> <OP> "plc/qb10" = 1+ count </OP> </FOR>

标签名称	含义
FORM	该标签包含用户对话框的说明。相应的标签在章节创建菜单与对话框窗口中有更多说明。 **句法：** `<FORM name="<dialog name>" color="#ff0000">` **属性：** • **color** 对话框窗口的背景颜色（颜色代码参见章节"颜色代码 (页 312)"） – 了解缺省值 • **name** 窗体的名称 • **type** 允许的值为 *cycle* ，用于识别用户循环窗口 • **xpos** 对话框左上角 X 位置（可选） • **ypos** 左上角 Y 位置（可选） • **width** X 方向上的长度（以像素为单位）（可选） • **height** Y 方向上的长度（以像素为单位）（可选）

FORM 续	**属性：**
	• **truetypefont**
	对话框窗口的字体是否为 true type（可选）
	
	值：
	• **true**
	无论机床数据 MD1113 设为何值，对话框窗口的字体均为 true type
	• **false**
	对话框窗口的字体通过机床数据 MD1113 位 0 设置（默认）
	– 位 0 = 0：文本显示采用固定字体类型（默认值）
	– 位 0 = 1：文本显示采用 true type 字体
	
	示例：
	`<FORM name="dialog name" truetypefont="true">`
	…
	…
	…
	`</form>`

标签名称	含义
FORM 续	该标签可定义在分辨率为 800x600 像素的屏幕上如何显示窗口中的图片，即，通过设置机床数据 MD1113 位 1 将图片调整为屏幕大小. **前提条件：** 只有将脚本文件复制到屏幕分辨率为 800x600 像素的数控系统中才能使用该功能。 **值：** • 位 1 = 0：图片按 1.25 倍系数自动放大（默认值） • 位 1 = 1：不自动放大，保留原有图片尺寸
IF	有条件指令（**IF, THEN, ELSE**） THEN 和 ELSE 标签包含在 IF 标签中。 在 CONDITION 标签中执行的条件处于 IF 标签之后。需要通过其他的指令处理来确定运算结果。如果该功能结果为真，则执行 THEN 分支而跳过 ELSE 分支。如果功能结果为假，则按 ELSE 分支分步程序进行处理。 **句法：** <IF> <CONDITION> Condition != 7 </CONDITION> <THEN> 该情况下的指令：满足条件 </THEN> <ELSE> 该情况下的指令：不满足条件 </ELSE> </IF> 或简化为如下指令（不能用于需带引号的地址）： <IF CONDITION>="Condition != 7"> <THEN> 该情况下的指令：满足条件 </THEN> <ELSE> 该情况下的指令：不满足条件 </ELSE> </IF> **示例：** <IF> <CONDITION> "plc/mb170" != 7 </CONDITION> <THEN> <OP> "plc/mb170" = 7 </OP> … </THEN> <ELSE> <OP> "plc/mb170" = 2 </OP> … </ELSE> </IF>

标签名称	含义
INCLUDE	指令包含有 XML 说明。 （参见该表中的 DYNAMIC_INCLUDE ） **属性：** • **src** 包含路径名称。 **句法：** <?INCLUDE src="<Path name>" ?>
LET	指令以给出的名称创建一个局部变量。 **域：** 使用属性 **dim**（维度），可创建一维或二维域。各个域元素通过域下标定址。 对于二维域，先指定行索引，而后指定列索引。 • 一维域： 索引 0 到 4 0 1 2 3 4 • 二维域： 索引行 0 到 3 以及索引列 0 到 5 0,0 0,1 0,2 0,3 0,4 0,5 1,0 1,1 1,2 1,3 1,4 1,5 2,0 2,1 2,2 2,3 2,4 2,5 3,0 3,1 3,2 3,3 3,4 3,5 **属性：** • **name** 变量名称 • **type** 变量类型可以为整数（INT）、无符号整数（UINT）、双精度浮点数（DOUBLE）、浮点数（FLOAT）、字符串（STRING）或 STRUCT。变量类型也可以是 **typedef** 所定义的结构。如果未给出类型指令，则系统会创建一个整数型变量。 <LET name = "VAR1" type = "INT" /> • **permanent** 如果属性为 **true**，则永久保存变量值。该属性仅对全局变量有效。 • **dim** 必须指定下列数目的域元素。对于二维域，先指定第一维度，再指定第二维度，两者用逗号分开。 通过域索引访问域元素，在变量名称之后于方括号中指定域索引。 name[index] 或 name[row,column] – 一维域：dim="<元素数目>" – 二维域：dim="<行数目>,<列数目>" 非初始化域元素预设为“0”。

标签名称	含义

LET 续	**示例：**
	一维域：
	`<let name="array" dim="10"></let>`
	二维域：
	`<let name="list_string" dim="10,3" type="string"></let>`
	
	预设值：
	可以对变量进行初始化赋值。
	`<LET name = "VAR1" type = "INT"> 10 </LET>`
	
	如果将 NC 或 PLC 变量的值保存在一个局部变量中，则赋值运算会自动将格式与所读入变量的格式进行匹配。
	• 字符串变量的预设值：
	只要将格式化的文本作为值进行传送，就可以为一个字符串变量赋值多行文本。如果一行要使用 line feed <LF>（换行）结尾，则应在行的末尾添加字符 "\\n"。
	
	`<LET name = "text" type = "string"> F4000 G94\\n`
	`G1 X20\\n`
	`Z50\\n`
	`M2\\n`
	`</LET>>`
	
	域（Arrays）：
	`<let name="list" dim="10,3">`
	`  {1,2,3},`
	`  {1,20}`
	`</let>`
	
	`<let name="list_string" dim="10,3" type="string">`
	`  {"text 10","text 11"},`
	`  {"text 20","text 21"}`
	`</let>`
	
	赋值：
	可以使用赋值运算“=”将机床数据或子程序中的值赋值给一个变量。
	变量到上一级 XML 块结束时一直有效。
	全局可用的变量要直接在标签 DialogGUi 后创建。
	在对话框中要注意：
	• 信息处理时会打开相应的标签。
	• 签并在信息处理完后关闭标签。
	• 因此标签中的所有变量都会被删除。

标签名称	含义

LET 续	**变量类型 struct**：
	该变量类型包含可使用结构名称定址的变量组合。一个结构可以包含所有的变量类型和结构。
	
	结构中的变量通过"element"标签进行说明。标签的属性和初始化与 LET 指令的属性和初始化一致。
	
	`<let name="name" type="struct">`
	
	`<element name="<Name>" type="<Variable type>" />`
	
	…
	
	…
	
	…
	
	`</let>`

LET 续	通过结构名称和变量名称访问结构变量。两个名称用点隔开。
	
	`<op>`
	
	`Structure_name.variable_name = value;`
	
	`</op>`
	
	示例：
	`<let name="info" type="struct">`
	
	`  <element name="id" type="int" />`
	
	`  <element name="name" type="string" />`
	
	`  <element name="phone" type="string" />`
	
	`</let>`
	
	`<op>`
	
	`  info.id = 1;`
	
	`  info.name = _T"my name";`
	
	`  info.phone = _T"0034 45634";`
	
	`</op>`
	
	结构初始化：
	可以在创建变量时通过为每个结构元素设置初始值对结构进行初始化。在结构数组中，每个结构间使用尖括号隔开。

MSG	操作组件显示标签中所给出的信息。 **示例：** `<MSG text ="my message" />`

标签名称	含义

MSGBOX	指令会打开一个信息框，其给分支的返回值可供使用。
	
	句法：
	`<MSGBOX text="<Message>" caption="<caption>" retvalue="<variable name>" type="<button type>" />`
	
	属性：
	• **text**
	文本
	• **retvalue**
	其中复制有返回值的变量名称：
	1 – OK
	0 – CANCEL
	• **type**
	应答选项：
	“BTN_OK”
	“BTN_CANCEL”
	“BTN_OKCANCEL”
	
	示例：
	`<MSGBOX text="Test message" retvalue="result" type="BTN_OK" />`

标签名称	含义

OP	该标签执行所列出的运算。
	可以执行章节“运算符 (页 312)”中所列出的各种运算。
	对于 NC/PLC 上的存取和驱动数据要在 **引用字符** 中设置完整的变量名称。关于地址构成的详细信息可以参见章节“组件地址分配 (页 338)”。
	PLC: "PLC/MB170"
	NC: "NC/Channel/…"
	
	示例：
	<LET name = "tmpVar" type="INT"> </LET>
	<OP> tmpVar = "plc/mb170" </OP>
	<OP> tmpVar = tmpVar *2 </OP>
	<OP> "plc/mb170" = tmpVar </OP>
	
	一个运算标签内可以使用多个等式。用分号标记指令结束。
	
	示例：
	<op>
	x = x+1;
	y = y+1;
	</op>
	
	字符串处理：
	运算指令可以对字符串进行处理并将结果字符串赋值给等式中所给出的字符串变量。
	为了标记文本表达式需要在文本前放置标志 _T 。此外还能对变量值进行格式化。格式规定以标志 _F 开始，后面跟有格式指令。接着给定变量地址。
	
	示例：
	<LET name="buffer" type="string"></LET>
	…
	…
	…
	<OP> buffer = _T"unformatted value R0= " +
	"nck/Channel/Parameter/R[0]" + _T" and " + _T"$$85051" + _T" for-
	matted value R1 " + _F%9.3f"nck/Channel/Parameter/R[1]" </OP>

标签名称	含义
OPERATION	**运算** 指令可在等式中移动。 **左移"<<"运算符** << 功能用于向左移位。可以直接设置或通过变量设置移动增量值和数量。如果达到数据格式的限制，则会越过限制移动位而不显示故障信息。 **执行规则** 从左向右执行；'+' 和 '-' 的优先级高于 '<<' 和 '>>' **示例：** <op> idx = idx << 2 </op> <op> idx = 3 + idx << 2 </op> **右移">>"运算符** >> 功能用于向右移位。可以直接设置或通过变量设置移动增量值和数量。如果达到数据格式的限制，则会越过限制移动位而不显示故障信息。 **执行规则** 从左向右执行；'+' 和 '-' 的优先级高于 '<<' 和 '>>' **示例：** <op> idx = idx >> 2 </op> <op> idx = 3+idx >> 2</op>
POWER_OFF	显示信息要求操作者关闭机床。显示文本固定保存在系统中。

标签名称	含义

PRINT	标签会在对话行中输出文本或者将文本复制到给定的变量中。
	如果文本包含有格式化标志，则会将变量值插入到相应的位置上。
	
	句法：
	`<PRINT name="变量名称" text="text %格式化"> Variable, … </PRINT>`
	`<PRINT text="text %格式化"> Variable, … </PRINT>`
	属性：
	• **name**
	其中应当保存文本的变量名称（可选）
	• **text**
	文本
	
	格式化：
	字符“%”会对作为值给定的变量进行格式化。
	%[标记] [宽度] [.小数点后面的位置] 类型
	• 标记：
	用于确定输出格式的可选字符：
	– 右对齐或左对齐（“-” 用于左对齐）
	– 前面加零（“0”）
	– 使用空格符填充
	• 宽度：
	确定一个非负数最小输出宽度的依据。如果待输出值占用的位置少于依据所确定的位置，则使用空格符填充空缺的位置。
	• 小数位：
	使用浮点数时，用可选参数定义小数位。
	• 类型：
	类型字符用来确定打印指令要输出哪些数据格式。该字符必须给定。
	– d:整数值
	– f：浮点数
	– s :字符串

PRINT 续	**值：**
	其值应当插入到文本中的变量数量。
	变量类型必须与格式规定的相应类型标识符一致并使用逗号相互分隔开来。
	
	示例：
	在信息行中输出文本
	`<PRINT text="信息文本" />`
	使用变量格式输出文本
	`<LET name="trun_dir"></LET>`
	`<PRINT text="M%d">trun_dir</PRINT>`
	在字符串变量中使用变量格式输出文本
	`<LET name="trun_dir"></LET>`
	`<LET name="str" type="string" ></LET>`
	`<print name="str" text="M%d ">trun_dir</print>`

标签名称	含义

PROGRESS_BAR	该标签打开或关闭进度条。进度条显示在应用窗口的下方。
	
	句法：
	`<PROGRESS_BAR type="<true/false>"> value </ PROGRESS_BAR>`
	
	属性：
	• **type =** "TRUE" - 打开进度条
	• **type** = "FALSE" - 关闭进度条
	• **min**
	（可选）– 最小值
	• **max**
	（可选）– 最大值
	
	值：
	• **Value**
	进度条的百分比位置
	
	示例：
	`<PROGRESS_BAR type="true" min="0" max="101" >20<`
	`/PROGRESS_BAR>......<PROGRESS_BAR >50<`
	`/PROGRESS_BAR>......<PROGRESS_BAR type="false" >100<`
	`/PROGRESS_BAR>`

SEND_MESSAGE	该标签会向有效的 FORM 标签发送带两个参数的消息，该消息在 message 标签中处理。
	
	句法：
	`<SEND_MESSAGE>p1, p2</SEND_MESSAGE>`
	
	示例：
	`<SOFTKEY POSITION="3">`
	`  <caption>Set%nParameter</caption>`
	`  <send_message>1, 0</send_message>`
	`</SOFTKEY>`
	
	`<FORM>`
	…
	…
	`<MESSAGE>`
	`<SWITCH>`
	`<CONDITION>$message_par1</CONDITION>`
	`<CASE value="1">`
	…
	…
	`</CASE>`
	`</SWITCH>`
	`</MESSAGE>`
	…
	…
	
	`</FORM>`

标签名称	含义

SHOW_CONTROL	使用该标签可设置控件的显示。
	
	句法：
	`<SHOW_CONTROL name="<name>" type="<type>" />`
	
	属性：
	• **name**
	控件的名称
	• **type =** "TRUE" - 控件可见
	• **type** = "FALSE" - 控件不可见（隐藏）
	
	示例：
	`<SHOW_CONTROL name="myEditfield" type="false" />`
	…
	…
	`<SHOW_CONTROL name="myEditfield" type="true" />`

SLEEP	该标签会在指定时间内中断脚本的执行。中断时间为返回值乘以基准时间。
	
	句法：
	`<SLEEP value="Interruption time" />`
	
	示例：
	等待时间，30 * 基准时间。
	`<SLEEP value="30" />`

STOP	解析在此处取消。
SWITCH	SWITCH 指令表示一种多重选择。表达式运用一次并与常数数目进行比较。如果表达式与常数一致，则处理 CASE 指令中的相关指令。 如果没有常数与表达式一致，则处理 DEFAULT 指令。 **句法：** `<SWITCH>` `<CONDITION>` 值 `</CONDITION>` `<CASE value="`常数 1">` 指令 … `</CASE>` `<CASE value="`常数 2">` 指令 … `</CASE>` `<DEFAULT>` 指令 … `</DEFAULT>` `</SWITCH>`
THEN	满足条件时的指令（IF, THEN, ELSE）

标签名称	含义
TYPE_CAST	该标签用于局部变量的数据类型转换。 **句法：** <type_cast name="variable name" type=" new type" /> **属性：** • **name** 变量名称 • **type** 给变量赋值新的数据类型。 • **convert** 给变量赋值新的数据类型。变量值也转换为新的数据类型。
TYPEDEF	通过标签“typedef”可为数据类型定义一个新的名称。这对于结构定义的优势在于只要定义一次数据类型，然后便能在 LET 指令中作为数据类型使用。 名称和类型作为属性。 分析程序仅支持结构定义的确定。 类型定义中的变量通过标签“element”进行说明。标签的属性与 let 指令的属性相符。 <typedef name="<名称>" type="struct"> <element name="<名称>" type="<变量类型>" /> … … … </typedef> 定义后，名称用作 LET 指令的数据类型。 <let name="<变量名称>" type="<名称>"></let> **示例**： <typedef name="my_struct" type="struct"> <element name="id" type="int" /> <element name="name" type="string" /> <element name="phone" type="string" /> </typedef> <let name="info" type="my_struct"></let> … … <op> info.id = 1; info.name = _T"my name"; info.phone= _T"0034 45634"; </op>

标签名称	含义
TYPEDEF 续	某些预定义的功能使用 RECT、POINT 或 SIZE 结构类型变量作为调用参数。这些结构在 struct_def.xml 文件中定义。 RECT: <typedef name="StructRect" type="struct" > <element name="left" type="int">0</element> <element name="top" type="int">0</element> <element name="right" type="int">0</element> <element name="bottom" type="int">0</element> </typedef> POINT: <typedef name="StructPoint" type="struct" > <element name="x" type="int">0</element> <element name="y" type="int">0</element> </typedef> SIZE: <typedef name="StructSize" type="struct" > <element name="width" type="int">0</element> <element name="height" type="int">0</element> </typedef>
WAITING	标签在 NC 或者驱动复位后等待组件重新启动。 **属性：** • WAITINGFORNC = "TRUE" - 系统等待数控系统重启 **句法：** <WAITING WAITINGFORNC = "TRUE"/> **示例：** ... <CONTROL_RESET resetnc = "true"/> <WAITING waitingfornc = "true"/> ...
WHILE	While 循环 WHILE (Test) 指令 **句法：** <WHILE> <CONDITION>…</CONDITION> 指令 … </WHILE> While 循环用于在条件满足时重复执行一系列指令。在处理指令序列前对条件进行检查。 **示例：** <WHILE> <CONDITION> "plc/ib9" == 0 </CONDITION> <DATA name = "PLC/qb11"> 15 </DATA> </WHILE>

标签名称	含义

XML_PARSER	“XML_PARSER”标签可用于解析 XML 文件。
	分析程序解析 XML 文件并调用定义的回调功能。每个回调功能都属于一个预定义事件。编程人员可在该功能内部处理 XML 文件。
	
	预定义事件：
	• start document
	分析程序打开文档并开始分析。
	• end document
	分析程序关闭文档。
	• start element
	分析程序找到了一个元素并创建了一个含所有属性和属性值的列表。
	这些列表被转送给回调功能。
	• end element
	找到元素结尾。
	• characters
	分析程序转送元素的所有字符。
	• error
	分析程序发现了一个句法错误。
	如果出现事件，分析程序会调用回调功能并检查功能的返回值。如果功能返回值为“true”，分析程序会继续进程。
	
	接口
	name 属性值包含 XML 文件的路径。
	如要给回调功能分配事件，须设置以下属性：
	标准
	startElementHandler
	endElementHandler
	charactersHandler
	可选
	errorHandler
	documentHandler
	通过属性值定义回调功能的名称。
	
	示例：
	`<XML_PARSER name="f:\appl\xml_test.xml">`
	`<!-- standard handler -->`
	`<property startElementHandler="startElementHandler" />`
	`<property endElementHandler="endElementHandler" />`
	`<property charactersHandler="charactersHandler" />`
	`<!-- optional handler -->`
	`<property errorHandler="errorHandler" />`
	`<property documentHandler="documentHandler" />`
	`</XML_PARSER>`

标签名称	含义

XML_PARSER 续	此外，分析程序还会提供可使回调功能访问事件数据的变量。
	
	startElementHandler:
	功能参数
	tag_name - 标签名称
	num - 找到的属性数
	系统变量
	$xmlAttribute 包含 0-num 属性名称范围的字符串数组。
	$xmlValue 包含 0-num 属性值范围的字符串数组。
	
	示例：
	`<function_body name="startElementHandler" return="true" parameter="tag_name, num">`
	`<print text="attribute name %s"> $xmlAttribute[0]</print>`
	`<print text="attribute value %s"> $xmlValue [0]</print>`
	`</function_body>`
	
	endElementHandler:
	功能参数
	tag_name - 标签名称
	
	示例：
	`<function_body name="endElementHandler" parameter="tag_name">`
	`<print text="name %s"> tag_name </print>`
	`</function_body>`

标签名称	含义
XML_PARSER 续	charactersHandler: **系统变量** $xmlCharacters 含数据的字符串 $xmlCharactersStart 始终为 0 $xmlCharactersLength 字节数 **示例：** <function_body name="charactersHandler" return="true" > <print text="chars %s"> $xmlCharacters </print> </function_body> documentHandler: **功能参数** state 1 开始文件，2 结束文件 errorHandler: **系统变量** $xmlErrorString 包含无效行（系统变量） **示例：** <function_body name="errorHandler" > <print text="error %s">$xmlErrorString</print> </function_body> **回调结果：** $return 如果为 1（true），分析程序继续分析文件

27.10.6.3 系统变量

系统变量可在 EasyXLanguage 脚本中用作脚本分析程序和程序流之间的数据交换。

下表中列出了可用在相应标签中的自动创建的系统变量。

系统变量	描述	有效的标签
$actionresult	告知脚本分析程序某个事件是否可执行	KEY_EVENT
$focus_name $focus_item_data	包含位于输入焦点的控件的名称 包含赋予控件的 item_data 的数值	FOCUS_IN INDEX_CHANGED EDIT_CHANGED
$return	传输子功能的返回值	FUNCTION_BODY
$message_par1 $message_par2	包含用于调用 SEND_MESSAGE 功能的参数	MESSAGE
$keycode $actionresult	包含带有可执行脚本的键码	KEY_EVENT
$xmlAttribute $xmlValue $xmlCharacters $xmlCharactersStart $xmlCharactersLength	包含已获取变量的属性列表 包含已获取变量的值列表 包含数据流 包含开始下标 包含数据流中保存的字符数量	XML_PARSER - startElementHandler XML_PARSER - charactersHandler

系统变量	描述	有效的标签
$mouse_event.type $mouse_event.x $mouse_event.y $mouse_event.id $mouse_event.buttons $mouse_event.button	包含传输鼠标事件参数的机构	MOUSE_EVENT

27.10.6.4 颜色代码

颜色属性使用 HTML 语言的颜色代码搭配。

颜色数据通过句法由字符“#”（菱形）和六个十六进制系统数字共同组成，而每种颜色通过两个数字代表。

R – 红色

G – 绿色

B – 蓝色

#RRGGBB

示例：

color = "#ff0011"

27.10.6.5 专用 XML 句法

如果要在通用 XML 编辑器中正确显示在 XML 句法中有特殊含义的字符，则必须加以说明。

关系到下列字符：

字符	XML 中的记数法
<	<
>	>
&	&
"	"
'	'

27.10.6.6 运算符

运算指令可以处理下列运算符：

运算符	含义
=	赋值
==	等于
<, <	小于
>, >	大于
<=, <=	小于等于
>=, >=	大于等于
|	位模式或（OR）连接
||	逻辑或（OR）连接
&, &	位模式与（AND）连接
&&, &&	逻辑与（AND）连接
+	加法
-	减法
*	乘法
/	除法
!	否
!=	不等

运算指令由左向右处理。括号可能对于表达式也有意义，可以确定部分表达式的处理优先级。

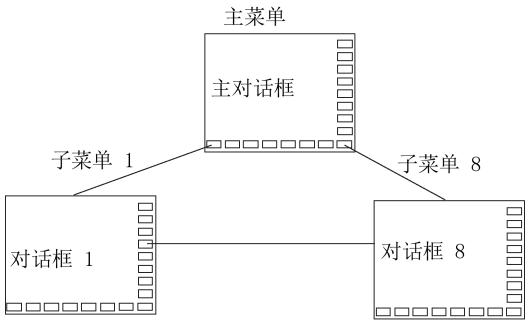
27.10.6.7 创建软键菜单和对话框窗口

要插入用户菜单，则必须在 XML 描述中有名为“main”的主菜单标签。系统会在激活用户自定义操作区后调用该标签。在该标签中可以定义其他的菜单分支以及对话框的激活。

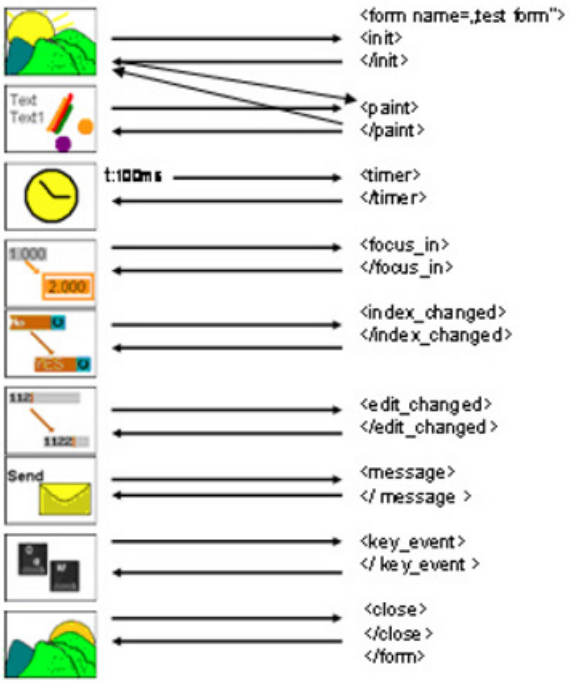
```
<menu name= "MAIN">
<OPEN_FORM name= "main dialogue">
<softkey POSITION="1">
<caption>sub menu 1</caption>
<navigation>sub menu 1</navigation>
</softkey>
<softkey POSITION="8">
<caption>sub menu 8</caption>
<navigation>sub menu 8</navigation>
</softkey>
</menu>

<menu name= "sub menu 1">
<OPEN_FORM name= "dialogue 1">
</menu>

<menu name= "sub menu 8">
<OPEN_FORM name= "dialogue 8“>
</menu>
```



标签名称	含义
FORM	该标签包含用户对话框的描述。 **属性：** • **color** 对话框窗口的背景颜色（颜色代码参见章节“颜色代码 (页 312)”） • **name** 窗体的名称 • **type** *cycle* -该属性定义循环窗口 • **xpos** 对话框左上角的 X 位置（可选） • **ypos** 左上角的 Y 位置（可选） • **width** X 方向上的长度（以像素为单位）（可选） • **height** Y 方向上的长度（以像素为单位）（可选） • **truetypefont** 对话框窗口的字体是否为 true type（可选）

标签名称	含义
FORM 续	对话框信息： • INIT • PAINT • TIMER • CLOSE • FOCUS_IN • INDEX_CHANGED • EDIT_CHANGED • KEY_EVENT • MESSAGE
FORM 续	
FORM 续	句法： <FORM name = "<dialog name>" color = "#ff0000"> 示例： <FORM name = "R-Parameter"> <INIT> <DATA_ACCESS type = "true" /> <CAPTION>R - Parameter</CAPTION> <CONTROL name = "edit1" xpos = "322" ypos = "34" refvar = "nck/Channel/Parameter/R[1]" /> <CONTROL name = "edit2" xpos = "322" ypos = "54" refvar = "nck/Channel/Parameter/R[2]" /> <CONTROL name = "edit3" xpos = "322" ypos = "74" </INIT> <PAINT> <TEXT xpos = "23" ypos = "34">R - Parameter 1</TEXT> <TEXT xpos = "23" ypos = "54">R - Parameter 2</TEXT> <TEXT xpos = "23" ypos = "74">R - Parameter 3</TEXT> </PAINT> </FORM>

标签名称	含义
INIT	**对话框信息** 在对话框生成之后立即执行该标签。在此创建所有的输入元素以及对话框窗口的“热链接”。

KEY_EVENT	对话框信息
	KEY_EVENT 标签可集成在窗体中用来评估键盘事件。如果在 FORM 中存在该标签，则系统会将 MF2 键盘代码发送给有效的 FORM。如果**$actionresult** 变量未设为零，系统会接着处理键盘事件。
	键盘代码作为整数值由**$keycode** 变量提供。
	
	示例：
	输入 **exclude_key** 变量的字符应从输入流中过滤掉。
	
	`<LET name="stream" type="string"/>`
	`<LET name="exclude_key" type="string"/>`
	
	`<FORM name = "keytest_form">`
	`  <INIT>`
	`  <CONTROL name = "p1" xpos = "120" ypos = "84" width ="200" refvar="stream" hotlink="true" />`
	`  <CONTROL name = "p2" xpos = "160" ypos = "104" width ="8" refvar="exclude_key" hotlink="true" />`
	`  </INIT>`
	
	`  <PAINT>`
	`  <text xpos = "8" ypos = "84">data stream</text>`
	`  <text xpos = "8" ypos = "104">exclude key</text>`
	`  </PAINT>`
	
	`  <KEY_EVENT>`
	`  <LET name="excl_keycode" type="string"/>`
	`  <OP>excl_keycode = exclude_key</OP>`
	`  <type_cast name="excl_keycode" convert="int" />`
	`  <PRINT text="%d %d">$keycode, excl_keycode</PRINT>`
	
	`  <IF>`
	`  <CONDITION>$keycode == excl_keycode</CONDITION>`
	`  <THEN>`
	`  <op> $actionresult = 0</op>`
	`  </THEN>`
	`  </IF>`
	
	`  </KEY_IVENT>`
	
	`</FORM>`

标签名称	含义

MESSAGE	**对话框信息**
	如果在脚本中执行 Send_message 指令，分析程序会执行 message 标签。值 P1 和 P2 由变量 $message_par1 和 $message_par2 提供（参见“SEND_MESSAGE”标签）。
	
	句法：
	`<MESSAGE>`
	`</MESSAGE>`
	
	示例：
	`<LET name="user_selection" />`
	
	`<SOFTKEY POSITION="3">`
	`<CAPTION>Set%nParameter</CAPTION>`
	`<SEND_MESSAGE>1, 10</SEND_MESSAGE>`
	`</SOFTKEY>`
	…
	…
	
	`<FORM>`
	…
	…
	`<MESSAGE>`
	`<SWITCH>`
	`<CONDITION>$message_par1</CONDITION>`
	`<CASE value="1">`
	`<OP> user_selection = $message_par2 </OP>`
	…
	…
	`</CASE>`
	`</SWITCH>`
	`</MESSAGE>`
	…
	…
	`</FORM>`

标签名称	含义

SEND_MESSAGE	该标签会向有效的 FORM 标签发送带两个参数的消息，该消息在 message 标签中处理（另请参见 MESSAGE）。
	
	句法：
	`<SEND_MESSAGE>p1, p2</SEND_MESSAGE>`
	
	示例：
	`<LET name="user_selection" />`
	
	`<SOFTKEY POSITION="3">`
	`  <CAPTION>Set%nParameter</CAPTION>`
	`  <SEND_MESSAGE>1, 10</SEND_MESSAGE>`
	`</SOFTKEY>`
	…
	…
	`<FORM>`
	…
	…
	`<MESSAGE>`
	`<SWITCH>`
	`<CONDITION>$message_par1</CONDITION>`
	`<CASE value="1">`
	`<OP> user_selection = $message_par2 </OP>`
	…
	`</CASE>`
	`</SWITCH>`
	`</MESSAGE>`
	…
	…
	`</FORM>`

FOCUS_IN	**对话框信息**
	当系统将焦点置于控件时，调用该标签。为了识别控件，系统将控件名称复制到变量 **$focus_name** 中并将 **item_data** 的属性值复制到变量**$focus_item_data** 中。
	例如可以使用该信息输出与焦点位置有关的图像。
	示例：
	`<focus_in>`
	`<PRINT text="focus on filed:%s, %d">$focus_name,`
	`$focus_item_data </PRINT>`
	`</focus_in>`

PAINT	**对话框信息** 在显示对话框时执行该标签。此处设置需要在对话框中显示的所有文本与图像。 此外，在系统识别出部分对话框需要重新显示时也会执行该标签。例如，当关闭上级窗口时会调用该标签。
TIMER	**对话框信息** 该标签周期性执行。 每个窗体均设有一个定时器。该标签在小于 100 ms 的周期内执行。

CAPTION	该标签包含有对话框的标题。
	应在 INIT 标签内使用该标签。
	
	句法：
	`<CAPTION>Titel</CAPTION>`
	
	示例：
	`<CAPTION>my first dialogue</CAPTION>`

标签名称	含义
CLOSE	**对话框信息** 在关闭对话框之前执行该标签。

CLOSE_FORM	该标签关闭激活的对话框。
	只有通过 MMC 指令打开对话框且为操作者提供了关闭对话框的软键功能时，才需要执行该指令。一般而言，对话框为自动管理，无需特意关闭。
	
	句法：
	`<CLOSE_FORM/>`
	
	示例：
	`<softkey_ok>`
	`<caption>OK</caption>`
	`<CLOSE_FORM />`
	`<navigation>main_menu</navigation>`
	`</softkey_ok>`

CONTROL	该标签用来创建控件元素。 **句法：** `<CONTROL name = "<control name>" xpos = "<X position>" ypos = "<Y position>" refvar = "<NC variable>" hotlink = "true" format = "<format>" />` **属性：** • **name** 域的名称。 名称会同时显示一个局部变量，不得在窗体中多次使用。 • **xpos** 左上角的 X 位置 • **ypos** 左上角的 Y 位置 • **fieldtype** 域类型 如未给定类型，则域被设置为编辑域。 – **edit** 数据可更改 – **readonly** 数据不可更改 **combobox** 域显示相应的名称而非数值。 如果选择了 **combobox** 域类型，则要为域另外设置显示内容。 为此要使用标签 <ITEM> 。 Combobox 将当前选定文本的索引保存在控件所属的变量中（参见属性 **refvar**）。 – **progressbar** 显示值范围在 0 到 100 的进度条。 可使用谷值和峰值属性根据待显示的数据调整值范围。

标签名称	含义

CONTROL 续	• **fieldtype**
	– **graphicbox**
	该域类型创建一个 2 维虚线图形控件。可使用<ITEM>标签将图形元素插入控件。参数 **width** 和 **height** 用来设置框的宽度和高度。
	创建控件之后，可使用 **loadItem**、**addItem** 或 **insertItem** 功能插入其他元素。该控件不对 **itemdata** 参数进行评估。
	示例：
	`<control name= "c_gbox1" xpos = "250" ypos="24" width="240" height="356" fieldtype="graphicbox" refvar="val" hotlink="true"`
	`  <property ORDINATE="Y sinus" />`
	`  <property ABSCISSA="time *100 ms" />`
	`</control>`

CONTROL 续	• **fieldtype** – **listbox** 该域类型可创建一个空的列表框控件。 可使用<ITEM>标签将列表框元素插入列表框。 通过 ITEM 属性 **value** 可赋予该元素唯一值。 例如，该操作可用于识别元素。 参数 **width** 和 **height** 用来设置列表框的宽度和高度。 创建控件之后，可使用 **addItem**、**insertItem** 或 **loadItem** 功能插入其他列表框元素。该控件不对 **itemdata** 参数进行评估。 – **itemlist** 该域类型创建一个静态控件，该控件显示相应的名称而非数值。 可使用<ITEM> 标签赋予该域一个名称。 • **item_data** 可以使用用户专用的整数值为该属性赋值。该值作为 FOCUS_IN 信息中的一部分，用来识别焦点域。 • **refvar** 可链接到该域的参考变量的名称（可选）。 • **hotlink** = "TRUE" 当参考变量数值改变时，该域自动更新（可选）。 • **format** 该属性定义了指定变量的显示格式。 数据格式请参见 print-Tag （可选）。

标签名称	含义
CONTROL 续	**属性：** • font 该属性确定所使用的字体大小。 – 0 – 1 – 2 – 3 – 4 – 5 • color_bk 该属性可设置控件的背景色。 • color_fg 该属性可设置控件的前景色。 颜色代码（参见章节“颜色代码 (页 312)”） • display_format 该属性定义指定变量的处理格式。在存取 PLC 浮点型变量时必须使用该属性，因为要通过双字读取来进行存取。 允许使用下列数据格式： – FLOAT – INT – DOUBLE – STRING 为列表框、图形框或复选框设置内容（例如要显示的文本或图形元素）： **句法：** <ITEM>Expression</ITEM> <ITEM value ="<Value>">Expression</ITEM>

标签名称	含义

CONTROL 续	**示例：**
	`<CONTROL name = "button1" xpos = "10" ypos = "10" fieldtype = " combobox ">`
	`<ITEM>text1</ITEM>`
	`<ITEM>text2</ITEM>`
	`<ITEM>text3</ITEM>`
	`<ITEM>text4</ITEM>`
	`</CONTROL>`
	如果要为显示内容设置任意一个整数值，则要为标签添加属性 **value = "数值"**。
	此时控制变量包含 item 的赋值，而不是连续数字。
	
	示例：
	`<CONTROL name = "button1" xpos = "10" ypos = "10" fieldtype = " combobox ">`
	`<ITEM value = "10">text1</ITEM>`
	`<ITEM value = "20">text2</ITEM>`
	`<ITEM value = "12">text3</ITEM>`
	`<ITEM value = "1">text4</ITEM>`
	</CONTROL>
	进度条示例：
	`<CONTROL name = "progress1" xpos = "10" ypos = "10" width = "100" fieldtype = "progressbar" hotlink = "true" refvar = "nck/Channel/GeometricAxis/actProgPos[1]">`
	`<PROPERTY min = "0" />`
	`<PROPERTY max = "1000" />`
	`</CONTROL>`
	列表框示例：
	`<let name="item_string" type="string"></let>`
	`<let name="item_data" ></let>`
	
	`<CONTROL name="listbox1" xpos = "360" ypos="150" width="200" height="200" fieldtype="listbox" />`
	• 添加元素：
	可使用 **insertitem**、**additem** 或 **loaditem** 功能添加元素。
	• 删除内容：
	可使用 **empty** 功能删除内容。
	
	`<op> item_string = _T"text1\\n" </op>`
	`<function name="control.additem">_T"listbox1", item_string, item_data </function>`
	`<op> item_string = _T"text2\\n" </op>`
	`<function name="control.additem">_T"listbox1", item_string, item data </function>`

CONTROL 续	**示例 itemlist：**
	`<CONTROL name = "itemlist1" xpos = "10" ypos = "10" fieldtype = " itemlist">`
	`<ITEM value = "10">text1</ITEM>`
	`<ITEM value = "20">text2</ITEM>`
	`<ITEM value = "12">text3</ITEM>`
	`<ITEM value = "1">text4</ITEM>`
	`</CONTROL>`

标签名称	含义
CONTROL 续	**创建后更改控件** 该标签用来更改已创建的现有控件的属性。该标签中必须设置待更改的控件名称以及新的属性。其只能在 FORM 标签中执行。可更改下列属性： • xpos • ypos • width • height • color_bk • color_fg • access level • fieldtype • itemdata • min • max • default 参考变量不可更改。如果某个属性需要通过软键事件触发更改，则必须通过 send message 标签将需求传输至 FORM 内容中。 **说明：** 如果使用较早软件版本下的脚本文件，由于控件行为发生变化，因此须检查控件显示是否完整。

CONTROL 续	**示例：**
	`<menu name = "main">`
	`<open_form name = "attrib_form" />`
	`<softkey POSITION="3">`
	`<caption>Set%nro</caption>`
	`<send_message>1, 0</send_message>`
	`</softkey>`
	`<softkey POSITION="4">`
	`<caption>Set%nrw</caption>`
	`<send_message>2, 0</send_message>`
	`</softkey>`
	`</menu>`
	`<form name="attrib_form">`
	`<init>`
	`<control name="c_p0" xpos="60" ypos="70" width="272" />`
	`</init>`
	`<message>`
	`<switch>`
	`<condition>$message_par1</condition>`
	`<case value="1">`
	`<control name="c_p0" fieldtype="static" />`
	`</case>`
	`<case value="2">`
	`<control name="c_p0" fieldtype="edit" />`
	`</case>`
	`</switch>`
	`</message>`
	`</form>`

标签名称	含义

CONTROL 续	下列属性可在指令描述中更改。为此，必须设置控件名称和属性。属性和控件名称之间必须用点隔开。
	• xpos
	• ypos
	• width
	• height
	• color_bk
	• color_fg
	• access level
	• fieldtype
	• itemdata
	• min
	• max
	• default
	• disable
	• tooltip
	• font
	• factor
	
	句法：
	<name>.<property>
	
	示例：
	`<let name="value" />`
	`<let name="w" />`
	`<let name="h" />`
	`<menu name = "main">`
	`<open_form name = "attrib_form" />`
	`<softkey POSITION="3">`
	`<caption>Set%nro</caption>`
	`</softkey>`
	`<op>`
	`c_move.xpos = 300;`
	`value = c_move.xpos;`
	`h = c_move.height;`
	`w = c_move.width;`
	`</op>`
	`</menu>`
	`<form name="attrib_form">`
	`<init>`
	`<control name="c_move" xpos="$xpos" ypos="124" />`
	`</init>`
	`</form>`

标签名称	含义
DATA_ACCESS	该标签用于控制保存用户输入时的对话框窗口的特性。 其特性应在 INIT 标签中定义。 如未使用该标签，则每次输入都会进行缓冲处理。 例外：**hotlink** 属性设置为 **true** 的控件始终会直接进行读写。 **属性：** • **type** = "TRUE" - 不对输入值进行缓冲处理。对话框窗口直接将输入值复制到参考变量中。 • **type =** "FALSE" – 只有通过标签 **UPDATE_CONTROLS** type = "FALSE" 才能将值复制到参考变量中。 **示例：** `<DATA_ACCESS type = "true" />`

EDIT_CHANGED	**对话框信息**
	如果某个编辑控件的内容发生更改即调用该标签。
	为了识别控件，系统将控件名称复制到变量**$focus_name** 中并将 **item_data** 的属性值复制到变量**$focus_item_data** 中。
	
	示例：
	`<EDIT_CHANGED>`
	`<print text="index changed filed:%s, %d"> $focus_name, $focus_item_data </print>`
	…
	…
	`</EDIT_CHANGED>`

INDEX_CHANGED	**对话框信息**
	当操作者更改列表框选项时即调用该标签。
	为了识别控件，系统将控件名称复制到变量
	$focus_name 中并将 **item_data** 的属性值复制到变量**$focus_item_data** 中。
	说明：
	分配给控件的参考变量此时还未与控件变量进行匹配，所含的仍是列表框上一次选择项的索引。
	
	示例：
	`<INDEX_CHANGED>`
	`<print text="index changed filed:%s, %d"> $focus_name, $focus_item_data </print>`
	…
	…
	`</INDEX_CHANGED>`

标签名称	含义
MENU	该标签用来定义菜单，其中包含有软键描述与待打开的对话框。 **属性：** • **name** 菜单名称 **句法：** <MENU name = "<menu name>"> … <open_form …> … <SOFTKEY …> </SOFTKEY> </MENU>
NAVIGATION	该标签用来确定所要调用的菜单。可在软键程序段、菜单程序段和 FORM 中使用。如果给标签分配了一个变量名称作为值，则分析程序会激活保存在变量中的菜单。 菜单程序段中，navigation 位于指令位置。后续指令将不再执行。 **句法：** <NAVIGATION>menu name</NAVIGATION> **示例：** <menu name = "main"> <softkey POSITION="1"> <caption>sec. form</caption> <navigation>sec_menu</navigation> </softkey> </menu> <menu name = "sec_menu"> <open_form name = "sec_form" /> <softkey_back> <navigation>main</navigation> </softkey_back> </menu>

标签名称	含义

OPEN_FORM	该标签可以打开指定名称的对话框窗口。
	
	属性：
	• **name**
	对话框窗口的名称
	
	句法：
	`<OPEN_FORM name = "<form name>" />`
	
	示例：
	`<menu name = "main">`
	`<open_form name = "main_form" />`
	`<softkey POSITION="1">`
	`<caption>main form</caption>`
	`<navigation>main</navigation>`
	`</softkey>`
	`</menu>`
	
	`<form name="main_form">`
	`<init>`
	`</init>`
	
	`<paint>`
	`</paint>`
	
	`</form>`

标签名称	含义

PROPERTY	使用该标签可以为操作控件定义附加属性。
	
	属性：
	• **max** = "<最大值>"
	• **min** = "<最小值>"
	• **default** = "<预设值>"
	• **factor** = "换算系数"
	• **color_bk** = "<背景色代码>"
	• **color_fg** = "<前景色代码>"
	• **password** = "<true>" - 所输入的字符显示为“*”
	• **disable** = ""<true/false>" - 在编辑控件中禁止/允许输入
	• **transparent**= "位图透明色"
	颜色代码（参见章节“颜色代码 (页 312)”）
	• **tooltip** = 当光标置于控件时显示信息文本。
	句法为：<property tooltip="note text" />
	• **abscissa** = "第一坐标轴的名称"（仅允许用于图形框）
	• **ordinate** = "第二坐标轴的名称"（仅允许用于图形框）
	示例：
	`<CONTROL name = "progress1" xpos = "10" ypos = "10" width = "100" fieldtype = "progressbar" hotlink = "true" refvar = "nck/Channel/GeometricAxis/actProgPos[1]">`
	`<PROPERTY min = "0" />`
	`<PROPERTY max = "1000" />`
	`</CONTROL>`
	`<CONTROL name = "edit1" xpos = "10" ypos = "10">`
	`<PROPERTY min = "20" />`
	`<PROPERTY max = "40" />`
	`<PROPERTY default = "25" />`
	`</CONTROL>`

标签名称	含义
SOFTKEY	该标签用于定义软键的属性与反应。 **属性：** • **position** 软键的编号。1-8 为水平软键， 9-16 为垂直软键 有效的软键属性如下： • **type** 定义软件的属性。 user_controled - 该脚本定义软键的显示方式 toggle_softkey - 软键在已按下和未按下状态下切换显示 • **refvar** 只能与 toggle_softkey 结合使用。 参考变量，将当前的软键属性复制到其中。 应将变量类型定义为“String”，其中包含属性“按下”、“未按下”或“锁定”（参见标签 state）。 • **picture** 借助该属性可将位图按左对齐方式显示在软键上。必须给出完整的路径名称。 可显示的文本字符数受位图宽度的限制。 此外还可以在软键程序段中定义如下动作： • **picturealignment** 该属性用来定义图片对齐方向。 默认情况下图片与软键左对齐。 可设置下列值改变对齐方式： – **top** 顶对齐 – **bottom** 底对齐 – **left** 左对齐 – **right** 右对齐 – **center** 居中对齐 • **caption** 软键文本 • **state** 只能与 user_controlled 结合使用。 利用该标签设置软键在系统上所需的显示。 **句法：** <state type="<state>" /> 可设置下列字符串： – **notpressed** 软键显示为未按下状态。 – **pressed** 软键显示为已按下状态。 – **disabled** 软键被锁定并显示为灰色。 • **navigation** • **update_controls** • **function**

标签名称	含义
SOFTKEY 续	**句法：** **标准软键：** <state type="<softkey state>" /> <softkey position = "<1>"> … … </softkey> **或** **脚本控制软键：** <softkey position = "<1>" type="<user_defined>" > <state type="<softkey state>" /> … … </softkey> **或** **切换软键：** <softkey position = "<1>" type="<toggle_softkey>" refvar="<variable name>" > … … </softkey> **示例：** <let name="define_sk_type" type="string">PRESSED</let> <let name="sk_type">1</let> <softkey POSITION="1" type="user_controled" > <caption>Toggle%nSK</caption> <if> <condition>sk_type == 0 </condition> <then> <op> sk_type = 1 </op> <op> define_sk_type = _T"PRESSED" </op> </then> <else> <op> define_sk_type = _T"NOTPRESSED" </op> <op> sk_type = 0 </op> </else> </if> <state type="$$$define_sk_type" /> … … </softkey>

标签名称	含义

SOFTKEY 续	**示例：**
	或
	
	`<let name="curr_softkey_state" type="string">PRESSED</let>`
	`</softkey>`
	
	`<softkey POSITION="3" type="toggle_softkey" refvar="curr_softkey_state">`
	`<caption>Toggle%nSK</caption>`
	…
	
	</softkey>
	
	
	

SOFTKEY_OK	该标签定义了“OK”（确认）软键的响应特性。
	
	
	此外还可以在软键程序段中定义如下动作：
	• **navigation**
	• **update_controls**
	• **function**
	
	句法：
	`<SOFTKEY_OK>`
	…
	…
	`</SOFTKEY_OK>`

SOFTKEY_CANCEL	该标签定义了“Cancel”（撤消）软键的响应特性。
	
	
	此外还可以在软键程序段中定义如下动作：
	• **navigation**
	• **update_controls**
	• **function**
	
	句法：
	`<SOFTKEY_CANCEL>`
	…
	…
	`</SOFTKEY_CANCEL>`

标签名称	含义

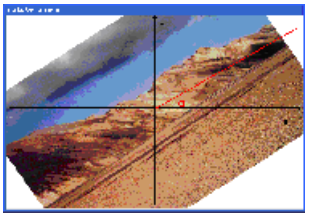
SOFTKEY_BACK	该标签定义了“Back”（返回）软键的响应特性。
	
	
	此外还可以在软键程序段中定义如下动作：
	• navigation
	• update_controls
	• function
	
	句法：
	`<SOFTKEY_BACK>`
	…
	…
	`</SOFTKEY_BACK>`

SOFTKEY_ACCEPT	该标签定义了“Accept”（接收）软键的响应特性。
	
	
	此外还可以在软键程序段中定义如下动作：
	• navigation
	• update_controls
	• function
	
	句法：
	`<SOFTKEY_ACCEPT>`
	…
	…
	`</SOFTKEY_ACCEPT>`

TEXT	该标签用来在指定的位置上显示文本。 **句法：** `<TEXT xpos = "<X position>" ypos = "<Y position>"> Text </TEXT>` **属性：** • xpos 左上角的 X 位置 • ypos 左上角的 Y 位置 • color 文本颜色（请参见章节“颜色代码 (页 312)”） **值：** 待显示的文本

标签名称	含义

IMG	该标签用来在指定的位置上显示图片。支持 BMP 和 PNG 图片格式。
	
	句法：
	`<IMG xpos = "<X position>" ypos = "<Y position>" name = "<name>" />`
	`<IMG … xrot="angle X axis" yrot=" angle Y axis " zrot=" angle Z axis " />`
	
	属性：
	• **xpos**
	左上角的 X 位置
	• **ypos**
	左上角的 Y 位置
	• **name**
	完整路径名称
	• **transparent**
	位图的透明色（参见章节“颜色代码 (页 312)”）

IMG 续	**示例：**
	图片绕 Z 轴旋转 34 度。
	`<img xpos = "5" ypos = "23" name = "f:/appl/pic3.bmp" height="355" width="550" zrot="34" color_bk="#ffffff" ></img>`
	
	可选：
	希望修改图片的原始大小时，可以使用属性 **width** 和 **height** 来定义尺寸。
	• **width**
	宽度（像素）
	• **height**
	高度（像素）
	
	示例：
	`<IMG xpos = "20" ypos = "40" name = "f:/appl/test.bmp" />`
	`<IMG xpos = "5" ypos = "23" name = "f:/appl/test.bmp" height = "355" width = "550"/>`

标签名称	含义
BOX	该标签可以在指定的位置以指定的颜色绘制一个矩形。 **句法：** `<BOX xpos = "<X position>" ypos = "<Y position>" width = "<X extension>" height = "<Y extension>" color = "<Color code>" />` **属性：** • **xpos** 左上角的 X 位置 • **ypos** 左上角的 Y 位置 • **width** X 方向上的长度（以像素为单位） • **height** Y 方向上的长度（以像素为单位） • **color** 颜色代码（参见章节“颜色代码 (页 312)”）

FUNCTION	**函数调用**
	该标签可以执行属性“**name**”中指定的函数体。
	
	属性：
	• **name** = “函数体名称”
	• **return**= “用于保存函数结果的变量名称”
	
	值：
	需要传输到函数体的变量列表。变量之间必须以逗号隔开。最多可以传输 10 个参数。
	此外，还可以将常数或文本内容设置为调用参数。为了识别文本内容，需要在开头放置 _T 标识符。
	
	句法：
	`<FUNCTION name = "<function name>" />`
	调用的函数等待一个返回值
	`<FUNCTION name = "<function name>" return = "<Variablenname>" />`
	参数传递
	`<FUNCTION name = "<function name>"> var1, var2, var3 </FUNCTION>`
	`<FUNCTION name = "<function name>"> _T"Text", 1.0, 1 </FUNCTION>`
	
	示例：
	参见“FUNCTION_BODY”。

标签名称	含义
FUNCTION_BODY	**函数体** 该标签包含有一个子函数的函数体。需在 DialogGui 标签中对函数体进行编程。 **属性：** • **name** = “函数体名称” • **parameter** = “参数列表”（可选） 该属性列出所需传输的参数。参数之间必须以逗号隔开。 调用函数体时，会将函数调用中设置的参数值复制到所列出的传输参数中。 • **return** = "true"（可选） 如果属性为 true，则创建局部变量$return。退出函数时反馈给调用函数的函数返回值应复制到该变量中。 **句法：** **无参数的函数体** <FUNCTION_BODY name = "<function name>"> … … … </ FUNCTION_BODY> **带参数的函数体** <FUNCTION_BODY name = "<function_name>" parameter = "<p1, p2, p3>"> … <LET name = "tmp></LET> <OP> tmp = p1 </OP> … </FUNCTION_BODY> **带返回值的函数体** <FUNCTION_BODY name = "<function_name>" parameter = "<p1, p2, p3>" return = "true"> … <LET name = "tmp></LET> <OP> tmp = p1 </OP> … <OP> $return = tmp </OP> </FUNCTION_BODY>
FUNCTION_BODY 续	**示例：** <function_body name = "test" parameter = "c1,c2,c3" return = "true"> <LET name = "tmp">0</LET> <OP> tmp = c1+c2+c3 </OP> <OP> $return = tmp </OP> </function_body> … … … <LET name = "my_var"> 4 </LET> <function name = "test" return = " my_var "> 2, 3,4</function> <print text = "result = %d"> my_var </print> … …

标签名称	含义

REQUEST	借助该标签可以在周期性读取服务（热链接）中添加一个变量。这样可以缩短对未链接到控件的变量的存取时间。
	如果在值变化时自动调用一个函数，则应在属性中定义该函数的名称。
	该标签仅在 INIT 指令中进行处理。
	
	属性：
	• **name**
	地址标识符
	• **function**
	函数名称
	
	句法：
	`<REQUEST name = "<NC-Variable>" />`
	或
	`<REQUEST name = "<NC-Variable>" function="<function name>"/>`
	
	示例：
	`<request name ="plc/mb10" />`
	
	或
	
	`<function_body name="my_function" >`
	`<print text="value changed" />`
	`</function_body>`
	…
	…
	…
	`<request name ="plc/mb10" function="my function"/>`

UPDATE_CONTROLS	该标签在操作控件和参考变量之间进行比较。
	
	属性：
	• **type**
	该属性用于定义数据比较的方向。
	= TRUE – 从参考变量读取数据并复制到操作控件中。
	= FALSE – 从操作控件读取数据并复制到参考变量中。
	
	句法：
	`<UPDATE_CONTROLS type = "<Direction>"/>`
	
	示例：
	`<SOFTKEY_OK>`
	`< UPDATE_CONTROLS type="false"/>`
	`</SOFTKEY OK>`

27.10.7 创建用户菜单

27.10.7.1 创建加工循环窗口

循环支持功能可以通过 FORM 对话框自动创建和反编译循环调用。

以下标签可用于实现该功能：

- NC_INSTRUCTION
- CREATE_CYCLE

为标记循环窗口，应在 FORM 标签中定义 type 属性值为 cycle。该标记可实现 NC_INSTRUCTION 指令的执行。

示例

```
<FORM name = "cycle100_form" type= "CYCLE">
…
…
</FORM>
```

NC_INSTRUCTION 标签中包含待创建的循环调用。所有循环参数都应使用占位符进行预留处理。

示例

```
<FORM name = "cycle100_form" type= "CYCLE">
<NC_INSTRUCTION refvar= "cyc_string" >Cycle100 ($p1, $p2, $p3)</ NC_INSTRUCTION>
…
…
…
</FORM
```

CREATE_CYCLE 标签提供保存在占位符变量中的数值并生成 NC 指令。

然后复制到指定的变量中。

标签名称	描述
NC_INSTRUCTION	该标签用于定义要创建的 NC 指令。 所有列出的循环参数都会作为 FORM 字符串变量自动创建，并可用于 FORM。 前提条件：FORM 的属性 type 值设为 CYCLE。 **属性：** • refvar 如果向标签分配了一个参考变量，则所有参数都会预设为参考变量中保存的 NC 程序段中的值。 **句法：** <NC_INSTRUCTION> 带占位符的 NC 指令</ NC_INSTRUCTION> **示例：** <let name="cyc_string" type="string"> Cycle100(0, 1000, 5)</let> … … … <FORM name = "cycle100_form" type= "CYCLE"> <NC_INSTRUCTION refvar= "cyc_string" >Cycle100($p1, $p2, $p3)</ NC_INSTRUCTION> .. … … </FORM>

标签名称	描述
CREATE_CYCLE	该标签可生成 NC 程序段，其句法通过 NC_INSTRUCTION 标签的值定义。 在生成 NC 指令之前，分析程序会先调用 FORM 的 CYCLE_CREATE_EVENT 标签。该标签用于计算循环参数。 **句法：** <CREATE_CYCLE/> **选项：** 如果指定了参考变量，指令会将创建的调用复制到该变量中。 <create_cycle refvar="name" /> **属性：** • **refvar** 如果向该标签分配了一个参考变量，则 NC 指令会复制到该变量中。 **示例：** <LET name="cyc_string" type="string"> Cycle100(0, 1000, 5)</LET> <SOFTKEY_OK> <caption>OK</caption> <CREATE_CYCLE /> <close_form /> <navigation>main_menu</navigation> </SOFTKEY_OK> **或** <SOFTKEY_OK> <caption>OK</caption> <CREATE_CYCLE refvar= "cyc_string" /> <close_form /> <navigation>main_menu</navigation> </SOFTKEY_OK>

27.10.7.2 替代符号

系统可以确定有关运行时间的控制特性（属性值）。为了能够使用该函数，需要在一个局部变量中提供所需的特性并使用前置的 **$ 符号** 将变量名称作为属性值传输给标签。

如果标签要以字符串作为属性值或值，则应在变量名称前加上符号 $$$。

示例：

```
<let name="my_ypos">100</let>
<let name="field_name" type="string"></let>

<control name = "edit1" xpos = "322" ypos = "$my_ypos" refvar="nck/Channel/Parameter/R[1]" />

<op>my_ypos = my_ypos +20 </op>

<control name = "edit2" xpos = "322" ypos = "$my_ypos" refvar="nck/Channel/Parameter/R[2]" />
```

```
<print name =" field_name" text="edit%d">3</print>
<op>my_ypos = my_ypos +20 </op>

<control name = "$field_name" xpos = "322" ypos = "$my_ypos"
refvar="nck/Channel/Parameter/R3]" />

<caption>$$$field_name</caption>
```

27.10.8 组件地址分配

为了给 NC 变量、PLC 组件或驱动数据分配地址，地址名称由所需要的数据构成。一个地址由部分路径 **组件名称** 和 **变量地址** 组成。使用斜线作为分隔符。

关于所有系统变量的详细信息，请通过以下链接查阅《SINUMERIK 840D sl/828D 参数手册 - 系统变量》：

https://support.industry.siemens.com/cs/ww/en/ps/14593/man

27.10.8.1 PLC 地址分配

PLC 地址分配以路径部分 **plc** 开始。

允许使用下列地址：

DBx.DB(f)	数据块
I(f)x	输入端
Q(f)x	Output
M(f)x	标志
V(f)x	变量

DBx.DBXx.b	数据块
Ix.b	输入端
Qx.b	Output
Mx.b	标志
Vx.b	变量

数据格式 **f**：

B	字节
W	字
D	双字

在位地址分配时取消数据格式标识。

地址 **x**：

有效的 S7-200 地址名称

位地址分配：

b – 位号

示例：

```
<data name = "plc/mb170">1</data>
<data name = "i0.1"> 1 </data>
<op> "m19.2" = 1 </op>
```

27.10.8.2 NC 变量地址分配

NC 变量地址分配以路径部分 **nck** 开始。

在该部分之后跟有数据地址，其结构可以通过以下链接查阅《SINUMERIK 828D 参数手册 - NC 变量和接口信号》：

https://support.industry.siemens.com/cs/ww/en/ps/14593/man

示例：

```
<LET name = "tempStatus"></LET>
<OP> tempStatus ="nck/channel/state/chanstatus" </OP>
```

常用 NC 变量

下表列出了用于生成用户对话框的常用 NC 变量。

变量地址		含义	备注
Nck/Channel/MachineAxis/actToolBasePos[n]		机械坐标	• n = 0：X 轴 • n = 1：Y 轴 • n = 2：Z 轴 • n = 3：主轴
nck/Channel/GeometricAxis/actprogpos[n]		工件坐标（刀尖在 WCS 中的位置）	
nck/Channel/GeometricAxis/actToolEdgeCenterPos[n]		刀沿中心在 WCS 中的位置	
nck/Channel/State/actTNumber		当前刀具号	
nck/Channel/State/actDNumber		当前刀沿号	
nck/Channel/State/actFeedRateIpo		实际进给速度	
nck/Channel/State/cmdFeedRateIpo		编程进给速度	
nck/Channel/Spindle/actSpeed		实际主轴转速	
nck/Channel/Spindle/cmdSpeed		编程主轴转速	
车床	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+2]	n 号刀具 m 号刀沿 X 长度	• n = 刀具编号 • m = 刀沿编号 • p = (m - 1) x 35
	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+3]	n 号刀具 m 号刀沿 Z 长度	
	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+11]	n 号刀具 m 号刀沿 X 长度补偿	
	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+12]	n 号刀具 m 号刀沿 Z 长度补偿	
铣床	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+2]	n 号刀具 m 号刀沿 X 长度	
	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+3]	n 号刀具 m 号刀沿 Y 长度	
	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+4]	n 号刀具 m 号刀沿 Z 长度	
	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+11]	n 号刀具 m 号刀沿 X 长度补偿	
	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+12]	n 号刀具 m 号刀沿 Y 长度补偿	
	nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+13]	n 号刀具 m 号刀沿 Z 长度补偿	
nck/Tool/Compensation/cuttEdgeParam[u1,cn,p+5]		n 号刀具 m 号刀沿半径	
nck/Tool/Compensation/edgeData[cn,p+14]		n 号刀具 m 号刀沿半径补偿	
nck/Channel/ProgramInfo/block[u1,0]		当前执行程序段的上一行	
nck/Channel/ProgramInfo/block[u1,1]		当前执行的程序段	
nck/Channel/ProgramInfo/block[u1,2]		当前执行程序段的下一行	
nck/Nck/ChannelDiagnose/poweronTime[u1]		系统从上一次正常上电运行的时间（单位：min）	
nck/Channel/State/actProgNetTime[u1]		所选程序的运行时间（单位：s）	
nck/Channel/State/reqParts[u1]		待被计数的所需零件	
nck/Channel/State/actParts[u1]		实际所计零件数	
Nck/Channel/Parameter/R[n]		R 参数	

27.10.8.3 在运行时生成 NC/PLC 地址

可在运行时生成地址标识符。

此时字符串变量的内容会作为地址用在操作指令以及 nc.cap.read 和 nc.cap.write 功能中。

使用这种方式分配地址时须遵循以下说明：

- 变量名称写在引号内。
- 在变量名称前标记三个,$'字符。

句法：

```
"$$$variable name"
```

示例：

```
<PRINT name="var_adr" text="plc/DB9000.DBW%d"> 2000</PRINT>
<OP> "$$$var_adr" = 1 </OP>
```

27.10.8.4 驱动组件地址分配

分配驱动组件地址时须以路径 drive 开头。

然后指定具体的驱动设备：

CU - 控制单元

DC - 驱动控制

该部分要添加待设置的参数。

驱动组件的地址定义如下：

- do1:地址为 11 的驱动
- do2:地址为 12 的驱动
- do3:地址为 13 的驱动
- do4:地址为 14 的驱动
- do5:地址为 15 的驱动

示例：

```
<LET name="r0002_content"></LET>
<LET name="p1460_content"></LET>


<!– 在地址为 15 的驱动的 CU 上读取 r0002 值 ->
<op> r0002_content = "drive/cu/r2[do1]" </op>

<!– 在地址为 15 的驱动的 CU 上读取 r0002 值 ->
<op> r0002_content = "drive/cu/r0002[do5]" </op>

<!– 在地址为 12 的驱动的 DC 上读取 p1460 值 ->
<op> p1460_content = "drive/dc/p1460[do2]" </op>
```

示例：

此外还可以使用**$<变量名称>** "替代字符"从局部变量中读取驱动索引。

for instance DO$局部变量

示例：

```
<DATA name = "drive/dc/p1460[do1]">1</DATA>
```

间接地址分配：

```
<LET name = "driveIndex">1</LET>
<DATA name = "drive/dc/p1460[do$driveIndex]">1</DATA>
```

27.10.8.5 机床数据和设定数据的地址分配

可以使用 $ 字符加数据名称来识别驱动和设定数据。

机床数据：

$Mx_<名称[**索引, AX<轴编号>**]>

设定数据：

$Sx_<名称[索引, AX<轴编号>]>

x：

N – 通用的机床或设定数据

C – 通道专用的机床或设定数据

A – 轴专用的机床或设定数据

索引：

参数在该域中给出数据索引。

AX<轴编号>：

在轴专用数据时对所需要的轴（**<轴编号>**）进行详细说明。

可以选择利用“替代符号” **$<变量名称>** 从一个局部变量中读取轴索引。

例如 AX$ 局部变量

示例：

```
<DATA name ="$MN_AXCONF_MACHAX_NAME_TAB[0] ">X1</DATA>
```

轴的直接地址分配：

```
<DATA name ="$MA_CTRLOUT_MODULE_NR[0, AX1] ">1</DATA>
…
…
```

轴的间接地址分配：

```
<LET name ="axisIndex"> 1 </LET>
<DATA name ="$MA_CTRLOUT_MODULE_NR[0, AX$axisIndex] ">1</DATA>
```

27.10.8.6 用户数据的地址分配

用户数据地址分配以路径部分 **gud** 开始，后面跟有 GUD 名称。

为一个域分配地址，在名称后的方括号中应指定所需域索引。

示例：

```
<DATA name ="gud/syg_rm[0]"
<OP>"gud/syg_rm[0]" 0 10 </op>
```

全局用户数据（GUD）地址分配

地址分配以路径 gud 开头，后面跟有 CHANNEL 范围指定。该地址部分之后还跟有 GUD 范围指定：

GUD 范围	分配
sgud	西门子 GUD
mgud	机床制造商 GUD
ugud	用户 GUD

然后输入 GUD 名称。如要为一个数组分配地址，应在名称后面用方括号标记数组下标。

示例：

```
<data name ="gud/channel/mgud/syg_rm[0]">1</data>
<op>"gud/channel/mgud/syg_rm[0]" = 5*2 </op>
```

27.10.8.7 通过组件地址分配在用户对话框中创建典型菜单

本章节描述了通过生成组件地址分配所需的脚本文件（xmldial.xml）以及插入“xmldial. xml”文件中的 XML 文件（例如，machine_settings.xml）如何在用户对话框中创建典型菜单。

生成脚本文件

“xmldial.xml”控制文件中集成了带有“INCLUDE” XML 标签的 XML 文件，这些文件中包含用来在 HMI 上显示对话框窗口和参数的编程指令。

HMI	句法	使用的标识符及其含义
	<?xml version="1.0" encoding="gb2312"?>	
	<DialogGui type="1">	**DialogGui**：定义所有对话框配置。
	<?include src="f:\appl\Main.xml"?>	**INCLUDE**：包含所需的 XML 描述。
	<?include src="f:\appl\Setting.xml"?>	
	<?include src="f:\appl\Parameter.xml"?>	
	</DialogGui>	**DialogGui**：定义所有对话框配置。

生成用于创建登入菜单的 XML 文件

概述

- 分配地址的组件类型：NC 变量
- HMI 上的菜单功能：显示系统数据

用户对话框及 XML 文件示例



HMI	句法	使用的标识符及其含义
	<menu name="Main">	**MENU**：定义登入菜单，其中包含有软键说明与待打开的对话框。
	<OPEN_FORM name="Main" />	**OPEN_FORM**：打开指定名称下所需的对话框窗口。
①	<softkey position="1" type="user_controlled">	**SOFTKEY**：定义指定软键的特性与反应。
	<state type="notpressed" />	

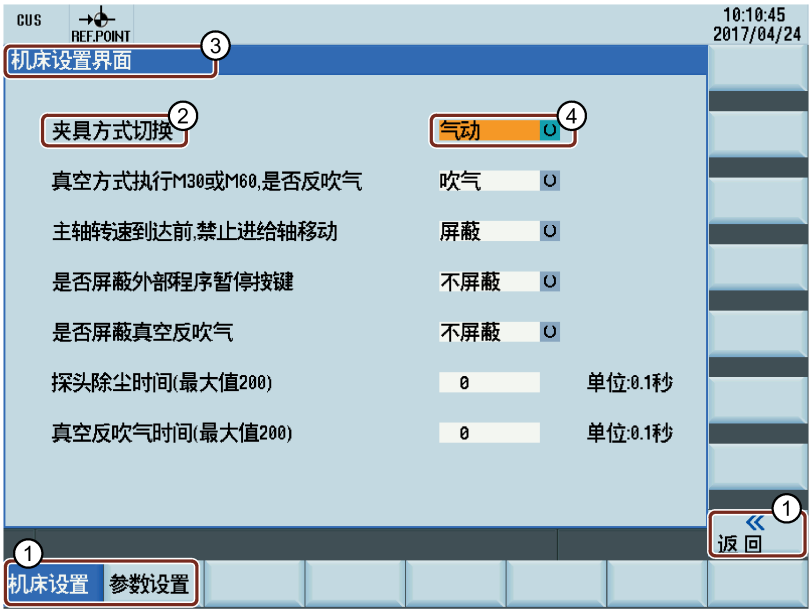
HMI	句法	使用的标识符及其含义
	<caption>机床设置</caption>	
	<navigation>Setting</navigation>	
	</softkey>	
	<softkey position="2" type="user controlled">	
	<state type="notpressed" />	
	<caption>参数设置</caption>	
	<navigation>Parameter</navigation>	
	</softkey>	
	</menu>	MENU：定义登入菜单，其中包含有软键说明与待打开的对话框。
	<form name="Main">	FORM：定义所需用户对话框的描述。
②	</paint>	PAINT：定义对话框显示所需的文本和图片。
	<text xpos="369" color="#000000" ypos="28">T, F, S</text>	
	...	
	</paint>	
	<init>	INIT：定义对话框窗口所需的输入元素和热连接。
③	<caption>用户界面</caption>	CAPTION：定义对话框的标题。
④	<let name="time_array" dim="7" />	LET：定义指定名称下的局部变量。
	<function name ="control.localtime">_T"time_array"</function>	
⑤	<control name="InputField1" xpos="489" ypos="54" height="16" width="40" refvar="nck/Channel/State/actDNumber" hot-link="true" format="%2d" />	CONTROL：定义所需的控制元素。
	...	
	</init>	INIT：定义对话框窗口所需的输入元素和热连接。
	</form>	FORM：定义所需用户对话框的描述。

生成用于创建机床操作菜单的 XML 文件

概述

- 分配地址的组件类型：PLC 数据
- HMI 上的菜单功能：通过对 PLC 数据进行地址分配来进行机床设置从而实现机床操作。

用户对话框及 XML 文件示例



HMI	句法	使用的标识符及其含义
	<menu name="Setting">	MENU：定义登入菜单，其中包含有软键说明与待打开的对话框。
	<OPEN_FORM name="Setting" />	OPEN_FORM：打开指定名称下所需的对话框窗口。
①	<softkey position="1" type="user controlled"> <state type="pressed" /> <caption>机床设置</caption> </softkey> <softkey position="2" type="user controlled"> <state type="notpressed" /> <caption>参数设置</caption> <navigation>Parameter</navigation> </softkey> <softkey back type="user controlled"> <state type="notpressed" /> <navigation>Main</navigation> </softkey back>	SOFTKEY：定义指定软键的特性与反应。
	</menu>	MENU：定义登入菜单，其中包含有软键说明与待打开的对话框。
	<form name="Setting">	FORM：定义所需用户对话框的描述。
②	<paint> <text xpos="15" color="#000000" ypos="60">夹具方式切换</text> ... </paint>	PAINT：定义对话框显示所需的文本和图片。
	<init>	INIT：定义对话框窗口所需的输入元素和热连接。
③	<caption>机床设置界面</caption>	CAPTION：定义对话框的标题。
④	<control name="Toggle1" xpos="346" ypos="60" height="16" width="80" refvar="plc/db1400.dbx9.0" hotlink="true" fieldtype="combobox">	CONTROL：定义所需的控制元素。

HMI	句法	使用的标识符及其含义
	<item value="1">真空</item>	
	<item value="0">气动</item>	
	</control>	
	...	
	</init>	**INIT**：定义对话框窗口所需的输入元素和热连接。
	</form>	**FORM**：定义所需用户对话框的描述。

生成用于创建参数设置菜单的 XML 文件

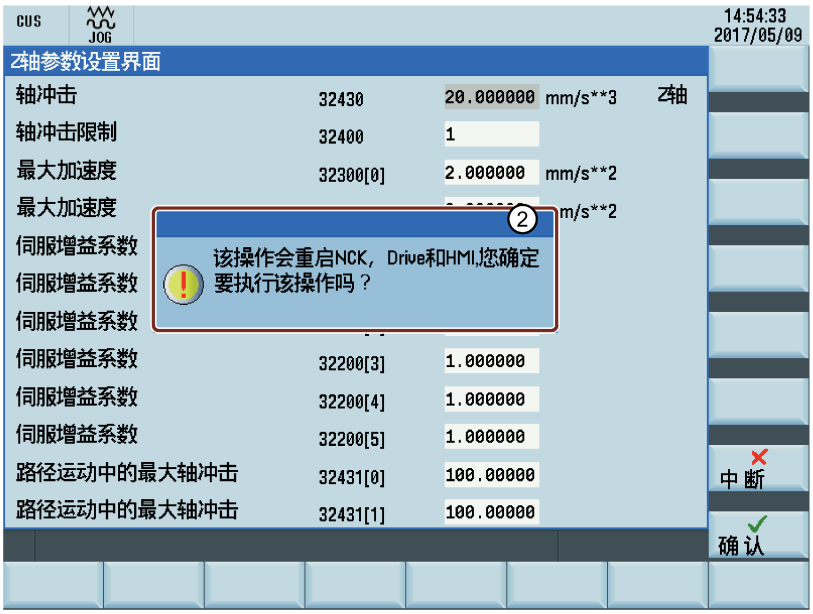
概述

- 分配地址的组件类型：机床数据和设定数据
- HMI 上的菜单功能：对常用的 Z 轴参数进行设置

用户对话框及 XML 文件示例



按下 ⇒



HMI	句法	使用的标识符及其含义
	<let name="var2" type="int"></let>	LET：定义指定名称下的局部变量。
	<menu name="Parameter">	MENU：定义登入菜单，其中包含有软键说明与待打开的对话框。
	<OPEN_FORM name="Parameter" />	OPEN_FORM：打开指定名称下所需的对话框窗口。
①	<softkey position="1" type="user_controlled">	SOFTKEY：定义指定软键的特性与反应。
	<state type="notpressed" />	
	<caption>机床设置</caption>	
	<navigation>Setting</navigation>	
	</softkey>	
	<softkey position="2" type="user_controlled">	
	<state type="pressed" />	
	<caption>参数设置</caption>	
	</softkey>	
	<softkey back type="user_controlled">	
	<state type="notpressed" />	
	<navigation>Main</navigation>	
	</softkey_back>	
	<softkey position="12">	
	<caption>激活</caption>	
②	<msgbox text="该操作会重启 NCK，Drive 和 HMI。您确定要执行该操作吗？" caption="1" retvalue="var2" type="btn_okcancel"/>	MSGBOX：打开所需的消息框，其返回值可用于分支。
	<if>	
	<condition>var2==1</condition>	
	<then>	
	<control_reset resetnc="true"/>	
	<waiting_waitingfornc="true"/>	
	</then>	
	</if>	
	</softkey>	SOFTKEY：定义指定软键的特性与反应。
	</menu>	MENU：定义登入菜单，其中包含有软键说明与待打开的对话框。
	<form name="Parameter">	FORM：定义所需用户对话框的描述。
③	<paint>	PAINT：定义对话框显示所需的文本和图片。
	<TEXT xpos="10" ypos="30" color="#000000">轴冲击</TEXT>	
	...	
	</paint>	
	<init>	INIT：定义对话框窗口所需的输入元素和热连接。
④	<caption>Z 轴参数设置界面</caption>	CAPTION：定义对话框的标题。
⑤	<DATA_ACCESS type="true"/>	CONTROL：定义所需的控制元素。
	<CONTROL name="III" xpos="350" ypos="30" height="20" width="75" fieldtype="edit" refvar="$MA_JOG_AND_POS_MAX_JERK[AX3]" hot-link="true"/>	
	...	
	</init>	INIT：定义对话框窗口所需的输入元素和热连接。
	</form>	FORM：定义所需用户对话框的描述。

27.10.9 预定义函数

脚本语言提供有不同的字符串处理和数学标准函数。后面列出的函数名称是预留的并且不能重载。

函数名称	描述

ncfunc.cap.read	该功能将指定地址中的值复制到局部变量中。如果读取成功，返回变量中的值为零。
	
	与运算指令相反，该功能在出错时不会中断执行脚本指令。
	
	句法：
	`<function name="ncfunc.cap.read" return="error">`
	`lokale variable, "address"</function>`
	
	示例：
	`<let name="error"></let>`
	`<function name="ncfunc.cap.read" return="error"> 3,`
	`"drive/cu/p0009"</function>`
	`<if>`
	`<condition>error != 0</condition>`
	`<then>`
	`<break />`
	`</then>`
	`</if>`

ncfunc.cap.write	该功能向指定变量写入一个值。如果写入成功，返回变量中的值为零。
	
	与运算指令相反，该功能在出错时不会中断执行脚本指令。
	
	句法：
	`<function name="ncfunc.cap.write" return="error">`
	`local variable or constant, "address"</function>`
	
	示例：
	`<let name="error"></let>`
	`<function name="ncfunc.cap.write" return="error">`
	`0, "drive/cu/p0009"</function>`
	`<if>`
	`<condition>error != 0</condition>`
	`<then>`
	`<break />`
	`</then>`
	`</if>`

函数名称	描述
ncfunc.pi_service	使用程序调用（PI）服务可将任务传输至 NCK。 如果服务执行成功，该功能会将值 1 返回到返回变量中。 **刀具列表的操作** _N_CREATO - 创建刀具 _N_DELETO - 删除刀具 _N_CREACE - 创建刀沿 _N_DELECE - 删除刀沿 **激活零点偏移** _N_SETUFR - 激活当前用户框架 _N_SETUDT - 激活当前用户数据 **程序段搜索** _N_FINDBL - 激活程序段搜索 _N_FINDAB -取消程序段搜索 **句法：** `<function name="ncfunc.pi_service" return="return var"> pi name, var1, var2, var3, var4, var5 </function>` **属性：** • name - 功能名称 • return- 用于保存执行结果的变量名称 – 值 == 1 – 成功执行任务 – 值 == 0 – 任务出错 **标签值：** • pi name - PI 服务名称（字符串） • var1 至 var5 - PI 专用自变量

函数名称	描述
ncfunc.pi_service 续	**自变量：** • _N_CREATO var1 - 刀具号 • _N_DELETO var1 - 刀具号 • _N_CREACE var1 - 刀具号 var2 - 刀沿号 • _N_DELECE var1 - 刀具号 var2 - 刀沿号 • _N_SETUFR 无自变量 • _N_SETUDT var1- 待激活的用户数据区域 – 1 - 刀具补偿数据 – 2 - 有效基本框架 – 3 - 有效可设置框架 • _N_FINDBL var1 - 搜索模式 – 2 - 带轮廓计算的搜索 – 4 - 搜索程序段末尾 – 1 - 不带计算的程序段搜索 • _N_FINDAB 无自变量 **示例：** • 创建刀具 - 刀具号 3 <function name="ncfunc.pi_service">_T"_N_CREATO", 3</function> • 删除 5 号刀具的刀沿 1 <function name="ncfunc.pi_service">_T"_N_DELECE", 5, 1</function>

函数名称	描述
ncfunc.chan_pi_service	该功能用于执行通道相关的 PI 服务。通道编号跟在 PI 服务名称后面传输。然后是其他所有调用参数。 **参数：** channel - 通道编号 **句法：** <function name="ncfunc.chan_pi_service" return="error"> _T"_N_SETUFR", channel, …</function> **示例：** <let name="chan" >1</let> <function name="ncfunc.chan_pi_service" return="error"> _T"_N_SETUFR", chan</function> <function name="ncfunc.chan_pi_service" return="error"> _T"_N_SETUDT", chan, _T"016", _T"00000", _T"00000"</function>
ncfunc.displayresolution	该功能提供由数控系统定义的浮点数转换规则。变量必须是字符串变量。 另见显示机床数据 MD203 DISPLAY_RESOLUTION 和 MD204 DISPLAY_RESOLUTION_INCH **句法：** <function name="ncfunc.displayresolution" return="dislay_res" /> **示例：** <let name="dislay_res" type="string"></let> … <function name="ncfunc.displayresolution" return="dislay_res" /> <control name = "cdistToGo" xpos = "210" ypos = "156" refvar="nck/Channel/GeometricAxis/progDistToGo[2]" hotlink="true" height="34" fieldtype="readonly" format="$$$dislay_res" time="superfast" color_bk="#ffffff"/>

函数名称	描述

ncfunc.password	该功能用于设置或删除口令。
	• 设置口令：
	所需口令级别的口令应在参数中设定。
	• 删除口令：
	空字符串可删除口令。
	
	句法：
	`<function name="ncfunc.password">password</function>`
	
	示例：
	`<let name="password"type="string"></let>`
	
	`<function name="ncfunc.password" > password</function>`
	`<function name="ncfunc.password" > _T"CUSTOMER"</function>`
	
	删除口令：
	<function name="ncfunc.password" > _T"" </function>

control.formcolor	该功能通过字符串设置对话框的文本颜色或背景颜色。
	颜色代码（参见章节“颜色代码 (页 312)”）
	
	范围：
	• BACKGROUND – 请求背景颜色值
	• TEXT – 请求文本（前景）颜色值
	
	句法：
	<function name="control.formcolor" return="variable">_T"range"</function>
	
	示例：
	`<let name="bk_color" type="string"></let>`
	
	`<function name="control.formcolor" return="bk color"> T"BACKGROUND"</function>`

函数名称	描述

control.localtime	该功能将本地时间复制到带有 7 个数组元素的域中。
	变量名称用作调用参数。
	以下信息保存在数组元素中：
	• 下标 0 - 年
	• 下标 1 - 月
	• 下标 2 - 星期
	• 下标 3 - 日
	• 下标 4 - 小时
	• 下标 5 - 分钟
	• 下标 6 - 秒
	
	句法：
	`<function name ="control.localtime">_T"time_array"</function>`
	
	示例：
	`<!-- index 0 = Year 1 = Month 2 = Day of week 3 = Day 4 = Hour 5 = Minute 6 = Second --> <let name="time_array" dim="7" /> <function name ="control.localtime"> T"time_array"</function>`

string.cmp	从字典编辑角度对两个字符串进行相互比较。
	当字符串相同时该函数提供返回值零，当第一个字符串小于第二个时返回值小于零或者当第二个字符串小于第一个时返回值大于零。
	参数：
	str1 - 字符串
	str2 - 比较字符串
	
	句法：
	`<function name="string.cmp" return ="<int var>" > str1, str2 </function>`
	
	示例：
	`<let name="rval">0</let> <let name="str1" type="string">A brown bear hunts a brown dog.</let> <let name="str2" type="string">A brown bear hunts a brown dog.</let> <function name="string.cmp" return="rval"> str1, str2 </function>`
	结果：
	rval= 0

函数名称	描述

string.icmp	在不区分大/小写的情况下从字典编辑角度对两个字符串进行相互比较。
	当字符串相同时该函数提供返回值零，当第一个字符串小于第二个时返回值小于零或者当第二个字符串小于第一个时返回值大于零。
	
	参数：
	str1 - 字符串
	str2 - 比较字符串
	
	句法：
	`<function name="string.icmp" return ="<int var>" > str1, str2 </function>`
	
	示例：
	`<let name="rval">0</let>`
	
	`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`
	`<let name="str2" type="string">A brown Bear hunts a brown Dog.</let>`
	`<function name="string. icmp" return="rval"> str1, str2 </function>`
	结果：
	rval= 0

string.left	该函数提取字符串 1 中最初的 n 数量字符并将其复制到返回变量中。
	
	参数：
	str1 - 字符串
	nCount - 字符数量
	
	句法：
	`<function name="string.left" return="<result string>"> str1,  nCount </function>`
	
	示例：
	`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`
	`<let name="str2" type="string"></let>`
	
	`<function name="string.left" return="str2"> str1, 12 </function>`
	结果：
	str2="A brown bear"

函数名称	描述

string.right	该函数提取字符串 1 中最后的 n 数量字符并将其复制到返回变量中。
	
	参数：
	str1 - 字符串
	nCount - 字符数量
	
	句法：
	`<function name="string.right" return="<result string>"> str1, nCount </function>`
	
	示例：
	`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`
	`<let name="str2" type="string"></let`
	
	`<function name="string.right " return="str2"> str1, 10 </function>`
	结果：
	str2="brown dog."

string.middle	该函数从索引 iFirst 开始提取字符串 1 中指定数量的字符并将其复制到返回变量中。
	
	参数：
	str1 - 字符串
	iFirst - 开始索引
	nCount - 字符数量
	
	句法：
	`<function name="string.middle" return="<result string>"> str1, iFirst, nCount </function>`
	
	示例：
	`<let name="str1" type="string">A brown bear hunts a brown dog.</let>`
	`<let name="str2" type="string"></let`
	
	`<function name="string.middle " return="str2"> str1, 2, 5 </function>`
	结果：
	str2="brown"

函数名称	描述
string.length	该函数可以提供字符串的字符数量。 **参数：** **str1** - 字符串 **句法：** <function name="string.length" return="<int var>"> str1 </function> **示例：** <let name="length">0</let> <let name="str1" type="string">A brown bear hunts a brown dog.</let> <function name="string.length" return="length"> str1 </function> **结果：** length = 31
string.replace	该函数可以使用新的字符串替代所有搜索到的部分字符串。 **参数：** **string** - 字符串变量 **find string** - 要替代的字符串 **new string** - 新的字符串 **句法：** <function name="string.replace"> string, find string, new string </function> **示例：** <let name="str1" type="string">A brown bear hunts a brown dog. </let> <function name="string.replace" > str1, _T"a brown dog" , _T"a big salmon"</function> **结果：** str1 = "A brown bear hunts a big salmon!"

函数名称	描述

string.remove	该函数可以删除所有搜索到的部分字符串。
	
	参数：
	string - 字符串变量
	remove string - 要删除的部分字符串
	
	句法：
	`<function name="string.remove"> string, remove string </function>`
	
	示例：
	`<let name="index">0</let>`
	
	`<let name="str1" type="string">A brown bear hunts a brown dog. </let>`
	`<function name="string.remove" > str1, _T"a brown dog" </function`
	结果：
	str1 = "A brown bear hunts"

string.insert	该函数可以在指定的索引位置插入一个字符串。
	
	参数：
	string - 字符串变量
	index - 索引（以零为基准）
	insert string - 要插入的字符串
	
	句法：
	`<function name="string.insert"> string, index, insert string </function>`
	
	示例：
	`<let name="str1" type="string">A brown bear hunts. </let>`
	`<let name="str2" type="string">a brown dog</let>`
	
	`<function name="string.insert" > str1, 19, str2</function>`
	结果：
	str1 = "A brown bear hunts a brown dog"

函数名称	描述

string.delete	该函数可以从指定的启始位置开始删除确定数量的字符。
	
	参数：
	string - 字符串变量
	start index - 启始索引（以零为基准）
	nCount - 要删除字符的数量
	
	句法：
	`<function name="string.delete"> string, start index , nCount </function>`
	
	示例：
	`<let name="str1" type="string">A brown bear hunts. </let>`
	
	`<function name="string.delete" > str1, 2, 5</function>`
	结果：
	str1 = "A bear hunts"

string.find	该函数可以在所传输的字符串中搜索与部分字符串一致的第一处地方。
	如果找到了部分字符串，则该函数会提供到第一个字符的索引（从零开始），否则为 -1。
	
	参数：
	string - 字符串变量
	findstring- 要搜索的字符串
	startindex – 开始下标（可选）
	
	句法：
	`<function name="string.find" return="<int val>"> str1, find string </function>`
	
	示例：
	`<let name="index">0</let>`
	`<let name="str1" type="string">A brown bear hunts a brown dog. </let>`
	`<function name="string.find" return="index"> str1, _T"brown" </function>`
	结果：
	Index = 2
	或
	`<function name="string.find" return="index"> str1, _T"brown", 1 </function>`

函数名称	描述

string.reversefind	该函数可以在所传输的字符串中搜索与部分字符串一致的最后一处地方。
	如果找到了部分字符串，则该函数会提供到第一个字符的索引（从零开始），否则为 -1。
	
	参数：
	string - 字符串变量
	find string- 要搜索的字符串
	startindex – 开始下标（可选）
	
	句法：
	`<function name="string.reversefind" return="<int val>"> str1, find string </function>`
	
	示例：
	`<let name="index">0</let>`
	`<let name="str1" type="string">A brown bear hunts a brown dog. </let>`
	`<function name="string.reversefind" return="index"> str1, _T"brown" </function>`
	结果：
	Index = 21
	或
	`<function name="string.reversefind" return="index"> str1, _T"brown", 10 </function>`
	结果：
	Index = 2

string.trimleft	该函数可以从字符串中删除打头的空格。
	
	参数：
	str1 - 字符串变量
	
	句法：
	`<function name="string.trimleft" > str1 </function>`
	
	示例：
	`<let name="str1" type="string">  test trim left</let>`
	`<function name="string.trimleft" > str1 </function>`
	结果：
	str1 = "test trim left"

函数名称	描述
string.trimright	该函数可以从字符串中删除后续的空格。 **参数：** **str1** - 字符串变量 **句法：** <function name="string.trimright" > str1 </function> **示例：** <let name="str1" type="string"> test trim right </let> <function name="string.trimright" > str1 </function> **结果：** str1 = "test trim right"
sin	该函数可以计算所传输值（以度为单位）的正弦值。 **参数：** **double** - 角度 **句法：** <function name="sin" return="<double val>"> double </function> **示例：** <let name= "sin_val" type="double"></let> <function name="sin" return="sin_val"> 20.0 </function>
cos	该函数可以计算所传输值（以度为单位）的余弦值。 **参数：** **double** - 角度 **句法：** <function name="cos" return="<double val>"> double </function> **示例：** <let name= "cos_val" type="double"></let> <function name="cos" return="cos_val"> 20.0 </function>

函数名称	描述
tan	该函数可以计算所传输值（以度为单位）的正切值。 **参数：** **double** - 角度 **句法：** <function name="tan" return="<double val>"> double </function> **示例：** <let name= "tan_val" type="double"></let> <function name="tan" return="tan_val"> 20.0 </function>
arcsin	该函数可以计算所传输值（以度为单位）的反正弦值。 **参数：** **double** - x 取值范围 -PI/2 至 +PI/2 **句法：** <function name="arcsin" return="<double val>"> double </function> **示例：** <let name= "arcsin_val" type="double"></let> <function name="arcsin" return="arcsin_val"> 20.0 </function>
arcos	该函数可以计算所传输值（以度为单位）的反余弦值。 **参数：** **double -** x 取值范围 -PI/2 至 +PI/2 **句法：** <function name="arcos" return="<double val>"> double </function> **示例：** <let name= "arccos_val" type="double"></let> <function name="arccos" return="arccos_val"> 20.0 </function>
arctan	该函数可以计算所传输值（以度为单位）的反正切值。 **参数：** **double** - 反正切 y/x **句法：** <function name="arctan" return="<double val>"> double </function> **示例：** <let name= "arctan_val" type="double"></let> <function name="arctan" return="arctan_val"> 20.0 </function>

函数名称	描述
文件处理	

doc.readfromfile	该功能可以将指定文件的内容读取到一个字符串变量中。
	可以选择将待读取的字符数量指定为第二参数。
	
	属性：
	return - 局部变量的名称
	
	参数：
	progname - 文件名称
	number of characters - 待读取的字符数量，字节（可选）：
	
	句法：
	`<function name="doc.readfromfile" return="<string var>"> progname, number of characters </function>`
	
	示例：
	`<let name = "my_var" type="string" ></let>`
	
	NC 文件系统
	`<function name="doc.readfromfile" return="my_var">_T"n:\mpf\test.mpf" </function>`
	
	CF 卡
	`<function name="doc.readfromfile" return="my_var">_T"f:\appl\test.mpf" </function>`
	
	或
	
	`<function name="doc.readfromfile" return="my_var">_T".\test.mpf" </function>`

函数名称	描述
doc.writetofile	该函数可以将一个字符串变量的内容写入指定文件中。 **参数：** **progname** - 文件名称 **str1** - 字符串 **句法：** <function name="doc.writetofile" > progname, str1 </function> **示例：** <let name = "my_var" type="string" > file content </let> NC 文件系统 <function name="doc.writetofile">_T"n:\mpf\test.mpf", my_var </function> CF 卡 <function name="doc.writetofile">_T"f:\appl\test.mpf", my_var </function> **或** <function name="doc.writetofile">_T".\test.mpf", my var </function>
doc.remove	该函数可以从目录中删除指定的文件。 **参数：** **progname** - 文件名称 **说明：** CF 卡上的文件要区分大小写。 **句法：** <function name="doc.remove" > progname </function> **示例：** NC 文件系统 <function name="doc.remove">_T"n:\mpf\test.mpf" </function> CF 卡 <function name="doc.remove">_T"f:\appl\test.mpf" </function> **或** <function name="doc.remove">_T".\test.mpf" </function>

函数名称	描述

doc.loadscript	该功能将零件程序中包含的对话框描述复制到指定的局部变量中。
	
	需要定义的调用参数为，程序名称、对话框名称以及用于保存主菜单名称的变量。如果在程序中找到对话框描述的名称，则返回变量中会包含该描述。如果变量内容保存在文件中，可以通过间接调用执行脚本。
	
	系统提供的脚本可从激活的零件程序中提取对话框描述并激活对话框。该脚本可在 MMC 命令中调用，从而激活零件程序相关的界面。
	
	句法：
	`<function name="doc.loadscript" return="<name of script variable>">progname, _T"dialog part name", main menu </function>`
	
	属性：
	return - 保存提取脚本的变量
	
	参数：
	progname - 程序的完整路径。（路径名称可通过 DOS 标记传输至该功能。）
	main menu - 找到的菜单名称会复制到该变量中
	dialog part name - 包含对话框描述的标签名称
	
	示例：
	`<function name="doc.loadscript" return="contents">prog_name, _T"main_dialog", entry</function>`

函数名称	描述
doc.exist	如文件存在，则函数提供值 1。 **参数：** **progname** - 文件名称 **说明：** CF 卡上的文件要区分大小写。 **句法：** <function name="doc.exist" return="<int_var>" > progname </function> **示例：** <let name ="exist">0</let> NC 文件系统 <function name="doc.exist" re- turn="exist">_T"n:\mpf\test.mpf" </function> CF 卡 <function name="doc.exist" re- turn="exist">_T"f:\appl\test.mpf" </function> **或** <function name="doc.exist" re- turn="exist">_T".\test.mpf" </function>
ncfunc.select	该函数可以选择将要处理的程序。程序必须保存在数控系统文件系统中。 **参数：** **progname** - 文件名称 **句法：** <function name="ncfunc.select"> progname </function> **示例：** NC 文件系统 <function name="ncfunc.select"> _T"n:\mpf\test.mpf" </function> CF 卡 <function name="ncfunc.select"> _T"f:\appl\test.mpf" </function> **或** <function name="ncfunc.select"> _T".\test.mpf" </function>

函数名称	描述
ncfunc.bitset	该功能用于设置指定变量的各个位。 可以置位或复位。 **句法：** <function name="ncfunc.bitset" refvar="address" value="set/reset" > bit0, bit1, … bit9 </function> **属性：** **refvar** - 指定变量名称，位组合应写入该变量 **value** - 位的值，范围为 0 和 1 **值：** 以零开头的位号应用作功能值。 每次调用最多可修改 10 个位。 **示例：** <function name="ncfunc.bitset" refvar="nck/Channel/Parameter/R[1]" value="1" > 0, 2, 3, 7 </function> <function name="ncfunc.bitset" refvar="nck/Channel/Parameter/R[1]" value="0" > 1, 4 </function>
control.delete	该功能可删除指定的控件。 **句法：** <function name="control.delete"> control name </function> **属性：** **name** - 功能名称 **值：** **control name** - 控件名称 **示例：** <function name="<control.delete>"> _T"my_editfield" </function>

函数名称	描述
control.additem	该功能可在列表末尾插入新的元素。 说明： 该功能仅适用“listbox”和“graphicbox”类型的控件。 **句法：** <function name="control.additem"> control name, item </function> **属性：** **name** - 功能名称 **值：** **control name** - 控件名称 **item** - 要插入的内容 **itemdata** - 整数值；由用户定义 **示例：** <let name ="itemdata">1</let> … … … <op> item_string = _T"text1" </op> <function name="control.additem">_T"listbox1", item_string, itemdata </function>
control.insertitem	该功能可在指定位置插入新的元素。 说明： 该功能仅适用“listbox”和“graphicbox”类型的控件。 **句法：** <function name="control.insertitem"> control name, index, item, itemdata </function> **属性：** **name** - 功能名称 **值：** **control name** - 控件名称 **index** - 位置从零开始 **item** - 要插入的内容 **itemdata** - 整数值；由用户定义 **示例：** <let name ="itemdata">1</let> … … … <op> item_string = _T"text2" </op> <function name="control.insertitem">_T"listbox1", 1, item_string, itemdata </function>

函数名称	描述
control.deleteitem	该功能可在指定位置删除元素。 说明： 该功能仅适用“listbox”类型的控件。 **句法：** <function name="control.deleteitem"> control name, index </function> **属性：** **name** - 功能名称 **值：** **control name** - 控件名称 **index** - 下标从 0 开始 **示例：** <function name="control.deleteitem">_T"listbox1", 1</function>

函数名称	描述
control.loaditem	该功能可在控件中插入表达式列表。 该功能仅适用“listbox”和“graphicbox”类型的控件。 **句法：** <function name="control.loaditem"> control name, list </function> **属性：** **name** - 功能名称 **值：** **control name** - 控件名称 **list** - 字符串变量 **列表结构：** 列表中包含多个表达式，表达式之间必须用 \n 隔开。 **示例：** <let name="item_string" type="string"></let> <let name="plotlist" type="string"></let> … … <print name ="item_string" text="p; %f; %f; %f; %f\n">s_z, s_x</print> <op>plotlist = plotlist + item_string</op> <print name ="item_string" text="l; %f; %f; %f; %f\n">s_z, s_x, e_z, e_x </print> <op>plotlist = plotlist + item_string</op> <op> s_x = e_x </op> <op> s_z = e_z</op> <op> e_x = s_x + 10 </op> <op> e_z = s_z - 100 </op> <print name ="item_string" text="l; %f; %f; %f; %f\n">s_z, s_x, e_z, e_x </print> <op>plotlist = plotlist + item_string</op> <function name="control.loaditem">_T"gbox", plotlist</function>

函数名称	描述
control.empty	该功能可删除指定列表框或图形框控件中的内容。 **句法：** <function name="control.empty"> control name, </function> **属性：** **name** - 功能名称 **值：** **control name** - 控件名称 **示例：** <function name="control.empty"> T"listbox1"</function>
control.getfocus	该功能提供位于输入焦点的控件的名称。 **句法：** <function name="control.getfocus" re- turn="focus_name" /> **属性：** **name** - 功能名称 **return** - 应指定字符串变量，控件名称会复制到该变量中。 **示例：** <let name>="focus_field" type="string"></let> <function name="control.getfocus" re- turn="focus_field"/>
control.setfocus	该功能可将指定控件设置为输入焦点。 控件名称应作为该功能的文本内容传输。 **句法：** <function name="control.setfocus"> control name </function> **属性：** **name** - 功能名称 **值：** **control name** - 控件名称 **示例：** <function name="control.setfocus" "> T"listbox1"</function>

函数名称	描述

control.getcurssel	该功能可提供列表框中的光标下标。
	控件名称应作为该功能的文本内容传输。
	
	句法：
	`<function name="control.getcurssel" retvar="var"> control name </function>`
	
	示例：
	`<let name>="index"></let>`
	`<function name="control.getcurssel" "> T"listbox1"</function>`

control.setcurssel	该功能可在列表框中将光标放置在相应的行。
	控件名称应作为该功能的文本内容传输。
	
	句法：
	`<function name="control.setcurssel" > control name, index</function>`
	
	示例：
	`<let name>="index">2</let>`
	`<function name="control.setcurssel" "> T"listbox1",index</function>`

control.getitem	该功能可将列表框中所选行的内容复制到指定的变量中。
	应指定字符串变量作为参考变量。
	控件名称应作为该功能的文本内容传输。
	
	句法：
	`<function name="control.getitem" return="var"> control name, index </function>`
	
	示例：
	`<let name>="index">2</let>`
	`<let name>="item" type="string"></let>`
	
	`<function name="control.getitem" return="item" "> T"listbox1",index</function>`

control.getitemdata	该功能可将列表框中用户自定义的元素值复制到指定的变量中。
	在 EDIT 控件中，该功能可将用户自定义的值（item_data）复制到指定的变量中。
	应指定整数变量作为参考变量。
	控件名称应作为该功能的文本内容传输。
	
	句法：
	`<function name="control.getitemdata" return="var"> control name, index </function>`
	
	示例：
	`<let name>="index">2</let>`
	`<let name>="itemdata"></let>`
	
	`<function name="control.getitemdata" return="itemdata" "> T"listbox1",index</function>`

函数名称	描述

bitmap.dim	该功能将位图尺寸复制到结构类型为 SIZE 的变量中。要定义类型，struct_def.xml 文件必须包含在项目中。
	
	句法：
	`<function name="bitmap.dim" >name, variable type </function>`
	
	参数：
	name - 文件路径
	variable type - SIZE 类型变量的变量名称
	
	示例：
	`<let name="bmp_size" type="size" />`
	
	`<function name="bitmap.dim" >_T"test.bmp", bmp_size </function>`

hmi.get_hmi_resolution	该功能将系统的绝对屏幕分辨率复制回 StructSize 结构类型的变量中。要定义类型，struct_def.xml 文件必须包含在项目中。 **句法：** `<function name="hmi.screen_resolution" >varialble type </function>`
hmi.get_hmi_resolution	该功能将 SINUMERIK Operate 使用的屏幕分辨率复制回 StructSize 结构类型的变量中。要定义类型，struct_def.xml 文件必须包含在项目中。 **句法：** `<function name="hmi.get_hmi_resolution" >variable type </function>`
hmi.get_caption_heigt	该功能以像素为单位提供标题栏的高度值。 **句法：** `<function name="hmi.get_caption_heigt" return="<return var>" />` **属性：** **return** - 整数变量
abs	该功能提供给定数值的绝对值。 **句法：** `<function name="abs" return="var"> value </function>`
sdeg	该功能将给定数值换算为度。 **句法：** `<function name="sdeg" return="var"> value </function>`

函数名称	描述
srad	该功能将给定数值换算为弧度。 **句法：** `<function name="srad" return="var"> value </function>`
sqrt	该功能可以计算给定数值的平方根。 **句法：** `<function name="sqrt" return="var"> value </function>`
round	该功能可以将所传输的数值保留指定的小数位数。如果未指定小数位数，则将数值四舍五入至第一位小数位。 **句法：** `<function name="round" return="var"> value, nDecimalPlaces </function>`
floor	该功能提供小于等于所传输值的最大整数值。 **句法：** `<function name="floor" return="var"> value </function>`
ceil	该功能提供大于等于所传输值的最小整数值。 **句法：** `<function name="ceil" return="var"> value </function>`
log	该功能可以计算给定数值的对数。 **句法：** `<function name="log" return="var"> value </function>`
log10	该功能可以计算给定数值的常用（十进制）对数。 **句法：** `<function name="log10" return="var"> value </function>`
pow	该功能计算“a^b”的值。 **句法：** `<function name="pow" return="var"> a, b </function>`
min	该功能会比较所传输的值并将较小的值返回。 **句法：** `<function name="min" return="var"> value1, value2 </function>`
max	该功能会比较所传输的值并将较大的值返回。 **句法：** `<function name="max" return="var"> value1, value2 </function>`

函数名称	描述
random	该功能提供伪随机数。 **句法：** `<function name="random" return="var" </function>`

27.11 快捷键

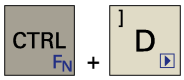
下列功能可通过在 PPU 面板上使用某些键组合来实现：

组合键	描述
<ALT> + <X>	打开加工操作区： 
<ALT> + <V>	打开程序编辑操作区： 
<ALT> + <C>	打开偏置操作区域： 
<ALT> + <B>	打开程序管理操作区： 
<ALT> + <D>	导出操作记录至 USB 存储器
<ALT> + <M>	打开“报警”操作区域： 
• <ALT> + <N> •  + 	打开系统数据操作区域： 
<ALT> + <H>	调用在线帮助系统
<ALT> + <L>	可以输入小写字母，此时在提示区域会显示以下图标： 
<ALT> + <S>	**仅在用户界面语言为中文时使用** 调用中文字符的输入法编辑器
<=>	调用小型计算器。 请注意此功能在“MDA”模式下不适用。
<CTRL> + <B>	在程序段中选择文本
<CTRL> + <C>	复制所选择的文本
<CTRL> + <D>	在屏幕上显示预先定义的页面
<CTRL> + <P>	截屏
<CTRL> + <R>	重新启动 HMI
<CTRL> + <S>	导出调试存档和操作记录至 USB 存储器
<CTRL> + 	增强屏幕背光亮度

组合键	描述
<CTRL> + 	减弱屏幕背光亮度

27.12 放映幻灯片

数控系统具备幻灯片放映功能。系统默认提供西门子产品信息的幻灯片。



在 PPU 上按下此按键组合即可显示幻灯片，再次按下组合键则退出幻灯片。

放映机床制造商幻灯片

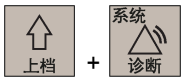
您可以创建自己的幻灯片并在数控系统上放映。系统幻灯片支持“.png”或“.bmp”图片格式。为达到最佳显示效果，推荐图片尺寸为 800*600 像素。

按照以下步骤放映自制幻灯片：

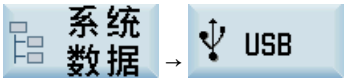
1. 准备好幻灯片图片并按照以下规则给每张图片命名：
   - **slide%u.png**，或
   - **slide%u.bmp**

   其中，“**%u**”代表从 “1” 开始的序数。举例而言，slide1.png、slide2.png、slide3.png...

   如您的图片同时包括 PNG 格式图片与 BMP 格式图片，则分别为这两种图片编号。在显示幻灯片时，PNG 格式的图片要优先于 BMP 格式的图片。
2. 将所有图片保存在 USB 存储器中的文件夹内，并且使用相应的语言代码命名该文件夹，例如，中文为“chs”，英文为“eng”。有关其他语言代码的详细信息，请参见“对机床制造商 HMI 数据的多语言支持 (页 262)”章节。请注意，非标准语言的幻灯片只有在系统已载入该语言的情况下才能生效。
3. 将 USB 存储器插入 PPU 前面板上的 USB 接口。



4. 选择系统数据操作区域。
5. 通过以下软键操作找到 USB 目录下包含幻灯片的文件夹：





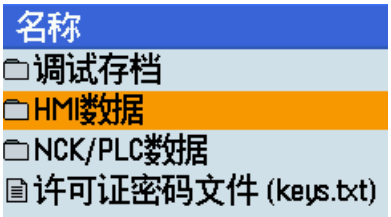
6. 选择文件夹并按此软键执行复制操作。



7. 按下该软键打开系统数据窗口。

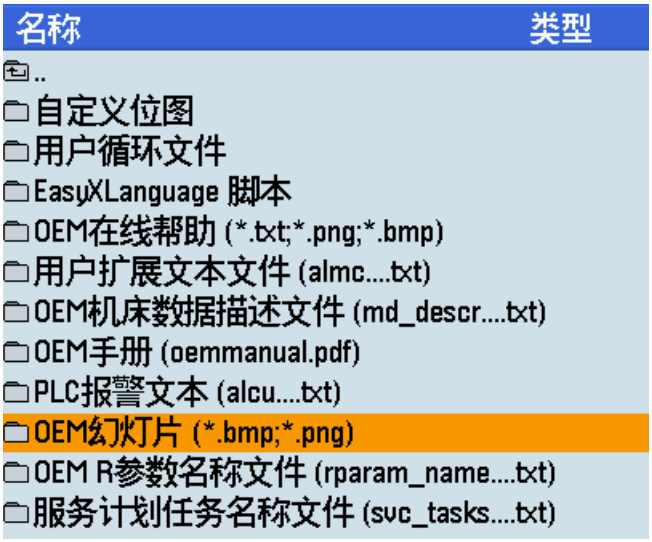


8. 选择 HMI 数据文件夹并按该键打开。





9. 移动光标至如下高亮显示的文件夹：

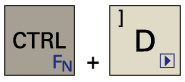




10. 按下该键打开文件夹。



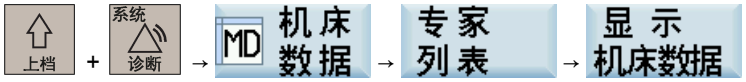
11. 使用该软键粘贴已复制的包含幻灯片的文件夹。



12. 按下该按组合键即可放映幻灯片。

说明

您可以通过 MD9001 TIME_BTWEEN_SLIDES 设置幻灯片放映的间隔时间。通过以下按键操作可找到该参数：



27.13 定义服务计划

使用维护计划功能可以设定维护任务，以特定的时间间隔进行机床维护（例如，补充油料、更换冷却液）。当间隔时间结束时系统会触发报警进行提醒。

在列表内您可以查看已设置的维护任务，以及离设定维护间隔期满的剩余时间。



① PLC 接口中维护任务的位置

② 维护任务的名称

③ 到下一次维护的最长时间，单位为小时

④ 显示首次报警的时间，单位为小时

⑤ 在输出最后一次报警之前操作员可应答的报警次数

⑥ 间隔时间结束前的剩余时间，单位为小时。剩余时间不可编辑。

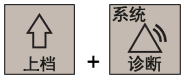
⑦ 指示维护任务的当前状态

:维护任务已启动。

:维护任务已完成。

新建维护任务

按以下步骤创建新的维护任务：



1. 打开系统数据操作区域



2. 按下 PPU 上的该键查看扩展软键。



3. 打开维护任务窗口。



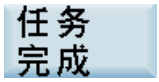
4. 按下该软键打开新建任务对话框。

5. 在相关输入栏中输入所需数据。

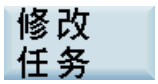


6. 按下该软键确认设置，新建的维护任务即显示在屏幕上。

7. 您还可以根据需要使用维护任务窗口中的垂直软键完成如下操作：



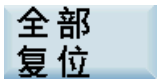
应答当前选中的已经完成的维护任务



编辑所选任务

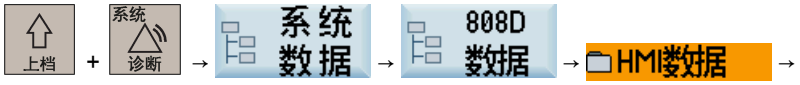


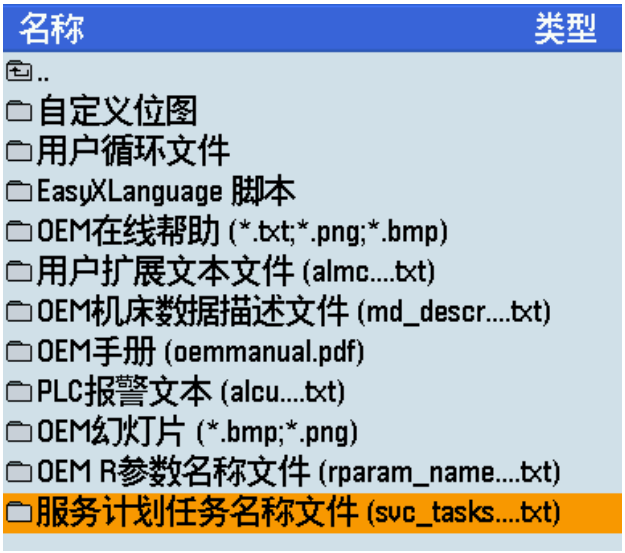
删除所选任务



重置所有任务的剩余时间

在数控系统上创建了至少一个维护任务之后，即可在系统上看到一个任务名称文件（.txt），该文件中包含在当前系统语言下创建的所有维护任务的名称。您可以通过下面的操作查看该文件：





任务名称文件是分语言版本的。您可以使用 USB 存储器将所需语言的文件复制到电脑中进行备份或者对维护任务描述进行批量编辑。

27.14 使用机床制造商开机启动画面和机床商标

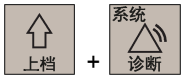
使用机床制造商开机启动画面

数控系统默认使用如下西门子开机启动画面。如果需要，您也可以使用自定义的开机启动画面。



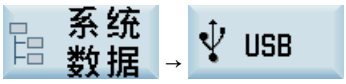
按以下步骤使用自定义开机启动画面：

1. 准备好 BMP 格式的开机启动画面图片。为达到最佳显示效果，推荐图片尺寸为 750*450 像素。将图片命名为“startup.bmp”并保存至 USB 存储器。
2. 将 USB 存储器插入 PPU 前面板上的 USB 接口。



3. 选择系统数据操作区域。

4. 通过以下软键操作找到 USB 目录下的开机启动画面文件：





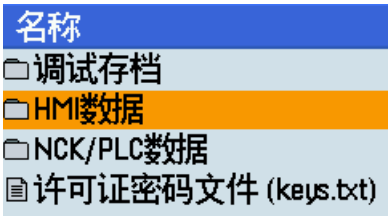
5. 使用光标键选择文件并按下该软键执行复制操作。



6. 按下该软键打开系统数据窗口。



7. 选择 HMI 数据文件夹并按该键打开。





8. 移动光标至如下高亮显示的文件夹：

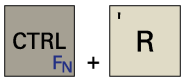




9. 按下该键打开文件夹。



10. 按下该软键使用您自己的开机启动画面文件替换默认文件。



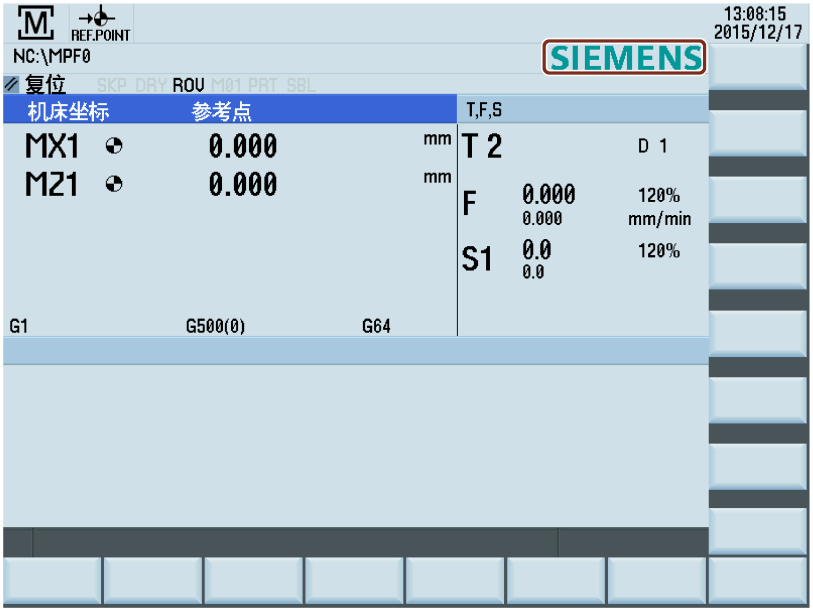
11. 按下这两个键重启 HMI。随后在系统启动过程中您就可以看到自定义的开机启动画面。

说明

要恢复默认的开机启动画面，需从数控系统中删除自定义位图文件（startup.bmp）。

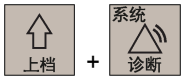
使用机床制造商商标

默认的机床商标显示在加工操作区域，如下所示：如果需要，您可以使用自定义的机床商标。

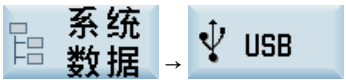


按以下步骤使用自定义机床商标：

1. 准备好 BMP 格式的机床商标图片。为达到最佳显示效果，推荐图片尺寸为 153*24 像素。将图片命名为“mtbico.bmp”并保存至 USB 存储器。
2. 将 USB 存储器插入 PPU 前面板上的 USB 接口。
3. 选择系统数据操作区域。



4. 通过以下软键操作找到 USB 目录下的机床商标文件：



5. 使用光标键选择文件并按下该软键执行复制操作。



6. 按下该软键打开系统数据窗口。



7. 选择 HMI 数据文件夹并按该键打开。







8. 移动光标至如下高亮显示的文件夹：

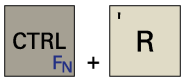




9. 按下该键打开文件夹。



10. 按下该软键使用您自己的产品商标文件替换默认文件。



11. 按下这两个键重启 HMI。随后即可在加工操作区域看到您自定义的机床商标。

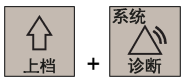
说明

要恢复默认的“SIEMENS”商标显示，需从数控系统中删除自定义位图文件（mtbico.bmp）。

27.15 配置启动后的操作区域

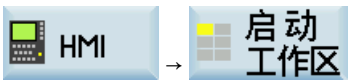
数控系统启动后默认显示加工操作区域。您也可以根据需要选择在系统启动后进入其他操作区域。

操作步骤

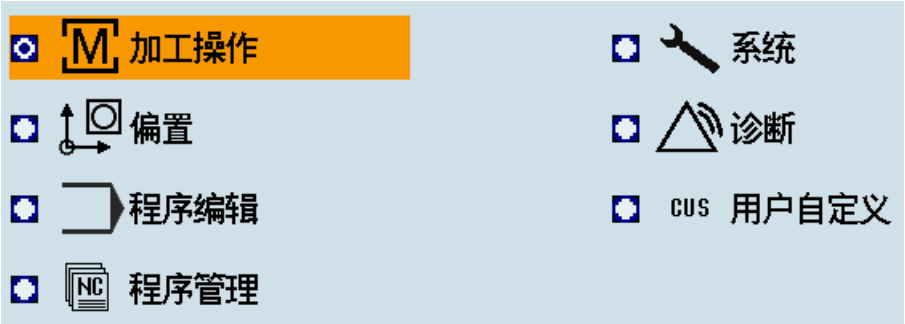


1. 选择系统数据操作区域。

2. 通过如下软键操作打开设置启动操作区域的窗口：



3. 使用光标键在如下窗口中选择所需的操作区域：



4. 按下该软键确认设置，已配置的操作区域会在系统重启后显示。

更 改
启动区域

27.16 数控锁功能

27.16.1 功能概述

说明

数控锁功能是一个需要购买许可证的选件功能（订货号：6FC5800-0AS71-0YB0）。

使用数控禁用功能必须在西门子购买相应的许可证。

说明

机床制造商有责任确保数控锁功能运行的正确性和可靠性。

注意
许可证 启用数控锁功能的单位（机床制造商或代理商）必须留存该选件的许可证（CoL）。 在忘记了 PIN 码的情况下，借助该许可证仍可得到西门子的授权。因此许可证（CoL）的所有者可以将机床解锁。

机床制造商可以使用数控锁功能以及通过调试工具 Access MyMachine（AMM）创建的加密文件来激活数控系统中的锁定日期。这样便可将机床的使用期限限制在锁定日期内。锁定日期届满后，数控系统的 NC 启动功能被禁用。此时正在“AUTO”模式下执行的 NC 程序不会被中断。

通过新的加密文件可以延长或者撤销数控锁功能。当最终用户履行合同约定的义务后，机床制造商负责将该文件提供给用户。

27.16.2 前提条件

使用数控锁功能必须满足以下前提条件：

- 数控锁功能选件须通过许可证密钥激活。有关激活选件功能的更多信息，请参见章节“激活可选功能 (页 395)”。
- 须使用 PLC Programming Tool 将经调整适用于当前机床的 PLC 项目（类型为 808D PPU15x/PPU16x）下载至数控系统。兼容模式不能激活。默认的 PLC 程序不得用于数控锁功能。关于下载 PLC 项目的详细信息，请参见 SINUMERIK 808D ADVANCED 调试手册。
- AMM 工具须安装在电脑上。关于 AMM 安装的详细信息，请参见 SINUMERIK 808D ADVANCED 调试手册。

27.16.3 限制

数控锁功能支持有使用期限的商业模式。以防止在超出规定时间后未经授权使用。直接访问数控系统便可实现功能操作。数控锁功能不具备绝对的防篡改保护功能。通过关闭数控系统的“AUTO”模式防止未经授权使用。由于此时不会中断正在运行的自动程序，因此，可导致运行时间超出锁定日期。SINUMERIK 数控系统的所有其他功能仍然可用。

只有机床制造商提供支持，数控锁功能才能发挥其作用。使用数控锁功能时，必须遵循以下说明和附加条件。

说明

数控锁功能是以连接了 PLC 项目的 SINUMERIK 808D ADVANCED 数控系统为基础的。该数控系统由面板处理单元（PPU）和装有系统软件的 CF 卡组合而成。

补充条件

注意

不当的操作可能导致数控锁功能规避

不当的操作可能导致数控锁功能规避

- 使用已解锁的 PLC 项目
- 使用默认的 PLC 项目
- 重新使用机床的 PLC 项目
- 未对 PLC 项目进行密码保护，或未能对密码保密

为防止功能规避，请遵循以下说明：

- 不得将未存有 OEM PIN 的 PLC 项目提供给用户。
- 不得重新使用 PLC 项目。
- 不得使用数控系统中提供的默认 PLC 项目运行机床。
- 必须对数控系统中的 PLC 程序结构单元进行密码保护（可在 PLC Programming Tool 中激活），这样可以防止用户拷贝机床相关的专有技术并应用在其自己的 PLC 用户程序中，并且防止其用带有数控锁 PLC 密钥的机床制造商的 PLC 用户程序替换用户自己的程序。PLC 项目的密码必须保密。

- 如尝试篡改数控锁功能和/或做出不一致的设置可能会导致机床停机。
- 使用数控锁功能时，机床制造商或代理商可能需要在用户现场请求额外的维修服务和技术支持。
- 必须为每台机床指定唯一的 OEM PIN，以实现更好的保护。
- 首次设置数控锁功能前，调试人员必须正确设置数控系统中的日期和时间。如果设置了过去的日期，机床的运行时间将会根据与实际日期之间的差值相应延长。
- 数控锁功能是以数控系统中的实时时钟为基础的。由于系统采用了免维护设计，因此时间可能会丢失。数控锁功能会对时间执行数字合理性检查。该检查可能会受到实时时钟电力损耗的影响。因此，掉电时间会被忽略。
- 软件功能故障可能导致数控系统被意外锁定。

27.16.4 防止篡改

数控锁功能用于确保只能在设置的时间范围内使用数控系统。尽管现有的保护机制能够防止数控锁功能被篡改，但仍存在漏洞和风险。数控锁功能会周期性检查 PPU、CF 卡和 PLC 项目中所安装的组合。所有三个组件更换时，数控锁功能取消。为了防止篡改，必须强制遵守章节“限制 (页 381)”中的附加条件。

说明

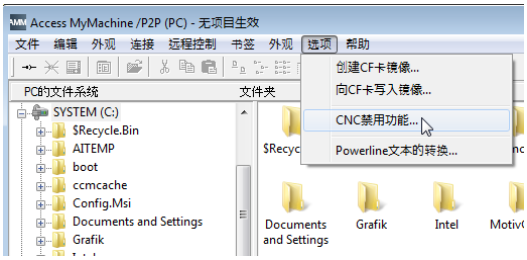
数控锁功能采用的是加密技术。首次投放市场时，所采用的技术符合当时最先进技术水平。随着时间的流逝，该项技术被绕过（篡改）的风险逐渐增加。

27.16.5 创建激活文件

使用数控锁功能必须创建一个与硬件相匹配的加密激活文件（.clc）。该文件可通过 AMM 生成。

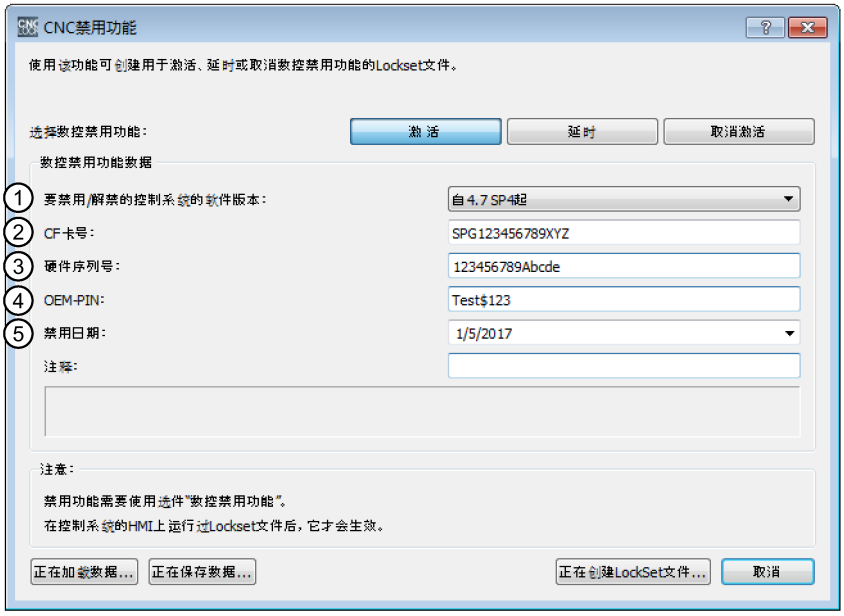
按以下步骤创建用于激活数控锁功能的加密文件：

1. 在电脑上打开 Access MyMachine。
2. 选择如下菜单命令：





3. 在显示的对话框中点击此按钮继续下一步。
4. 在对话框中输入所需的数据，例如：



① 对于 SINUMERIK 808D ADVANCED 数控系统必须选择软件版本 4.7 SP4。

② CF 卡的序列号可通过以下操作在 PPU HMI 上查看：



③ 硬件（PPU）序列号可在 PPU 铭牌上查看。

④ OEM PIN 码长度必须在 8 至 32 个英文字符，并且要包含大写字母、小写字母、数字和特殊字符（不支持空格）。机床制造商须妥善保管 OEM PIN 码并对其保密。

⑤ 机床制造商须在此设置一个禁用数控系统 NC 启动的日期。



5. 点击该按钮将激活数据保存为 .clc 格式的文件。

说明

OEM PIN 可提高数控锁功能的安全性，防止功能被篡改。

OEM PIN 会在系统激活数控锁功能时保存在 PLC 用户程序中。用户无法在 PLC 用户程序中查看、修改或删除 OEM PIN。

27.16.6 导入激活文件

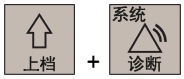
激活文件必须通过以太网连接或存储器（例如，USB 存储器）传输至数控系统。导入操作需要有最终用户或更高访问级别。导入时系统必须处于复位状态。

说明

在向数控系统中导入激活文件之前，机床制造商必须正确设置数控系统的时间和日期，因为激活数控锁功能的时间会保存作为监控的起始值。

按以下步骤导入激活文件：

1.
   - 如要通过 USB 导入文件，须将文件保存在 USB 存储器中，并将其插入 PPU 前面板上的 USB 接口。
   - 如要通过以太网导入文件，须将文件保存在电脑上的共享文件夹（网络驱动器）中，并通过以太网连接网络驱动器。



2. 选择系统数据操作区域。

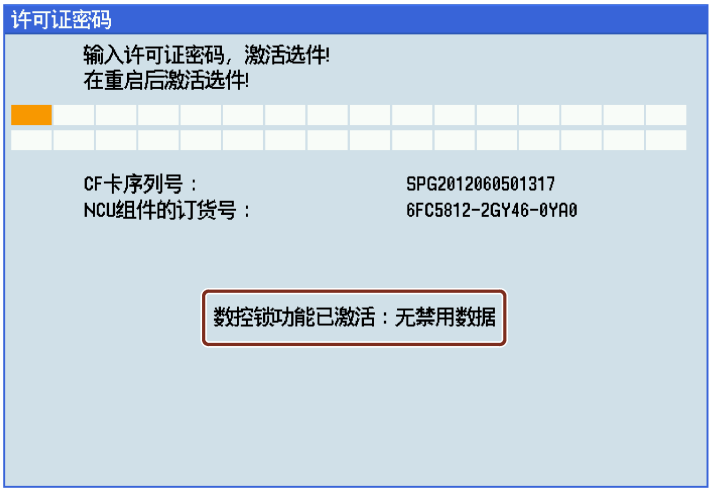


3. 按下该键查看扩展软键。

4. 通过以下软键操作打开许可证密钥对话框：



5. 检查并确保屏幕显示如下，此显示表明数控锁选件已获许可并已正确设置：





6. 按下该软键打开文件选择对话框：

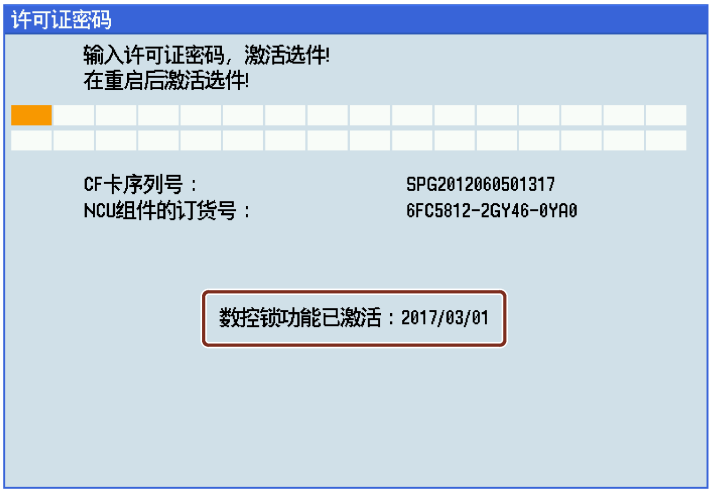


7. 选择目标目录并按下该键进入目录。



8. 找到所需的激活文件并按该软键执行导入。

如果在导入激活文件时未出错，数控系统中的数控锁功能即生效。此时在 HMI 屏幕上会显示具体的锁定日期，例如：



说明

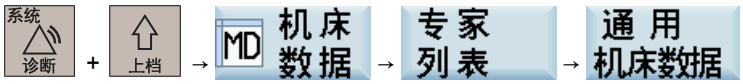
如果在导入激活文件时出错，系统会显示相应的报警。在此情况下数控锁功能的状态保持不变。

说明

建议机床制造商在完成机床调试并且激活数控锁功能之后创建一个涵盖所有系统组件的完整调试存档。这样能够确保数控锁功能数据的一致性。必要时还可借助该调试存档重新调试数控系统，从而可以重新激活数控锁功能而不需要请求维修服务。

设置预警时间

预警时间是指达到锁定日期之前的时间段，在该时间段内每天显示一次报警 8063。该报警提醒用户即将到达锁定日期，并且显示在 NC 启动禁用之前的剩余天数。预警时间通过机床数据 MD17300（默认值 = 30）设置，该机床数据可通过如下操作访问：



27.16.7 延长数控锁功能

要延长数控锁功能，机床制造商必须通过 AMM 工具创建一个**包含新的锁定日期**的激活文件。

创建激活文件

创建新的激活文件需要以下数据：

- CF 卡序列号
- PPU 序列号
- OEM PIN
- **新的锁定日期**

说明

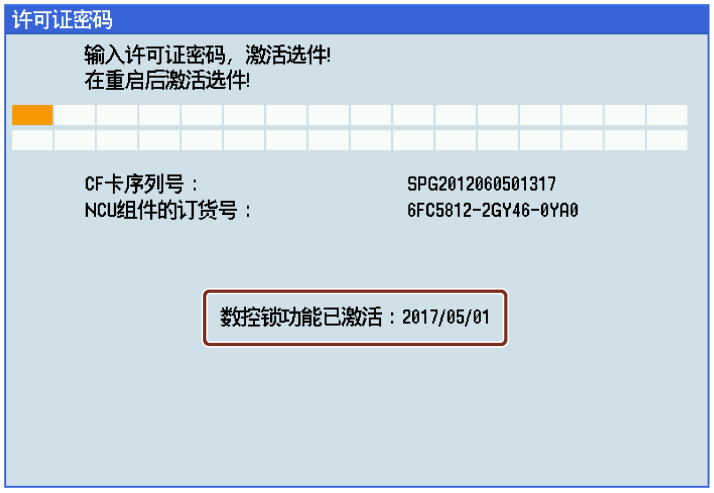
CF 卡和 PPU 的序列号以及 OEM PIN 码必须与初次激活数控锁功能时所使用的值一致。

延长数控锁功能时创建激活文件的步骤与激活数控锁功能基本相同（请参见章节“创建激活文件 (页 383)”），除了第 3 步中必须点击 延时 按钮。

导入激活文件

必须将新的激活文件导入数控系统来延长数控锁功能。可由机床制造商直接导入新的激活文件，或者由机床制造商将文件发送给最终用户由其传输至数控系统。关于如何导入激活文件的更多信息，请参见章节“导入激活文件 (页 384)”。

如果在导入激活文件时未出错，数控系统中的数控锁功能即生效新的锁定日期。此时在 HMI 屏幕上会显示已延长的锁定日期，例如：



说明

如果在导入激活文件时出错，系统会显示相应的报警。在此情况下数控锁功能的状态保持不变。

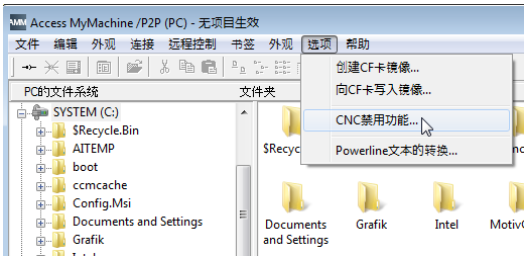
27.16.8 撤销数控锁功能

要撤销数控锁功能，机床制造商必须通过 AMM 工具创建一个**不含锁定日期**的撤销文件。

创建撤销文件

按以下步骤创建撤销文件：

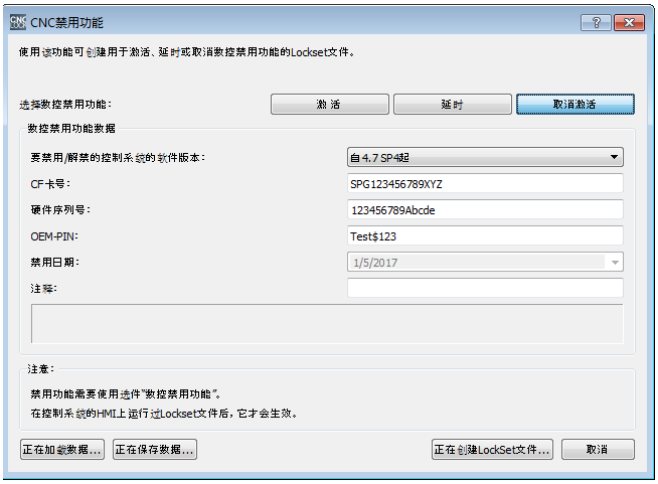
1. 在电脑上打开 Access MyMachine。
2. 选择如下菜单命令：





3. 在显示的对话框中点击此按钮继续下一步。

4. 在对话框中输入所需的数据，例如：



说明：CF 卡和 PPU 的序列号以及 OEM PIN 码必须与初次激活数控锁功能时所使用的值一致。更多信息请参见章节“创建激活文件 (页 383)”。

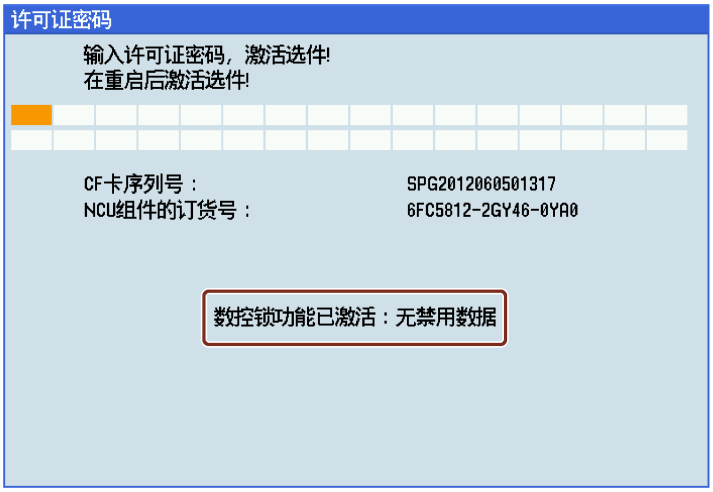


5. 点击该按钮将撤销数据保存为 .clc 格式的文件。

导入撤销文件

必须将撤销文件导入数控系统以撤销数控锁功能。可由机床制造商直接导入撤销文件，或者由机床制造商将文件发送给最终用户由其传输至数控系统。关于如何导入文件的更多信息，请参见章节“导入激活文件 (页 384)”。

如果在导入撤销文件时未出错，数控系统中的数控锁功能即撤销。此时可看到如下所示的数控锁状态：



说明

如果在导入撤销文件时出错，系统会显示相应的报警。在此情况下数控锁功能的状态保持不变。

说明

建议最终用户在撤销数控锁功能之后创建一个涵盖所有系统组件的完整调试存档。必要时可以借助该调试存档重新调试数控系统，而无需再次撤销数控锁功能。

27.16.9 更换损坏的数控系统硬件（PPU）及/或 CF 卡

若遇 PPU 及/或 CF 卡损坏，必须同时更换 PPU 与 CF 卡两者。

更换完成后，原来的许可证密钥不再有效，因而无法再次激活之前购买的选件功能。

此时必须首先联系西门子热线让其解除对原先选件许可证的绑定，而后重新分配许可证并激活这些选件功能。更多信息请参见章节“激活可选功能 (页 395)”。

还原机床状态

方法 1

在首次开启带有新 PPU 及新 CF 卡的数控系统后，必须导入之前在**取消**数控锁功能后创建的调试存档文件。

要再次创建数控锁功能，最终用户必须向机床制造商或代理商索要适用于数控系统的新的激活文件（.clc）。

方法 2

在首次开启带有新 PPU 及新 CF 卡的数控系统后，必须导入之前在**激活**数控锁功能后创建的调试存档文件。在重新开启数控系统后，显示报警 8062 且 NC 启动功能被禁用。报警产生的原因为新数控系统硬件及新 CF 卡的两个新序列号。

要将数控系统解锁，最终用户必须向机床制造商或代理商索要用于取消当前数控系统上数控锁功能的激活文件。在创建该激活文件时，务必使用 PPU 新及新 CF 卡的两个新序列号以及之前分配的 OEM PIN。有关如何使用 AMM 工具取消数控锁功能的更多信息，请参见章节“撤销数控锁功能 (页 386)”。

要再次创建数控锁功能，最终用户必须向机床制造商或代理商索要适用于数控系统的新的激活文件（.clc）。

创建激活文件

创建激活文件需要以下数据：

- **新** CF 卡序列号
- **新**数控系统（PPU）序列号
- OEM PIN
- 锁定日期（视数控锁功能状态可为上次锁定日期或新锁定日期）

说明

OEM PIN 需按照之前所述创建规则重新创建。

关于如何通过 AMM 创建激活文件的更多信息，请参见章节“创建激活文件 (页 383)”。

导入激活文件

必须将新的激活文件导入数控系统来更换损坏的 PPU 及/或 CF 卡。可由机床制造商直接导入新的激活文件，或者由机床制造商将文件发送给最终用户由其传输至数控系统。关于如何导入激活文件的更多信息，请参见章节“导入激活文件 (页 384)”。

如果在导入激活文件时未出错，新数控系统中的数控锁功能即生效。

说明

如果在导入激活文件时出错，系统会显示相应的报警。在此情况下数控锁功能的状态保持不变。

27.16.10 忘记 OEM PIN

如果机床制造商或代理商时忘记了初次激活时设置的 OEM PIN，则无法再为相应的数控系统创建有效的激活文件。

机床解锁

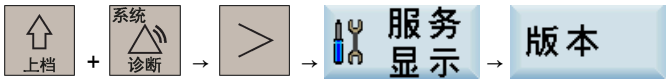
为确保机床制造商或代理商能够再次操作机床，相关技术人员必须联系西门子热线并提供如下信息进行解锁：

- 数控锁功能选件的许可证（CoL）
- CF 卡序列号
- 数控系统（PPU）序列号
- CNC 软件版本

说明

有关如何获取 CF 卡以及数控系统（PPU）序列号的详细信息，请参见章节“创建激活文件 (页 383)”。

CNC 软件的版本号可通过以下操作在 PPU HMI 上查看：



机床制造商或代理商从热线处获取用于解锁机床的激活文件。但解锁仅限于数控系统硬件（PPU）。PLC 项目不会被解锁。因此，必须使用适用于当前机床的原始 PLC 项目。

有关具体操作步骤的更多信息请联系西门子热线。

27.16.11 其它信息

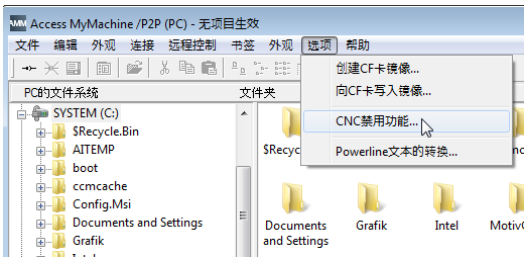
项目文件

您可以通过 AMM 工具创建包含以下数据的非加密项目文件（.ucls）：

- CF 卡序列号
- 硬件（PPU）序列号
- OEM PIN
- 创建日期
- 锁定日期

按以下步骤创建非加密项目文件：

1. 在电脑上打开 Access MyMachine。
2. 选择如下菜单命令：



3. 在对话框中输入所需的数据，例如：



4. 点击该按钮将数据保存为 .ucls 格式的文件。





您还可以点击此按钮重新导入保存在项目文件中的非加密数据。

日期设置错误

如果在数控锁功能激活时设置的日期早于实际日期，系统会在 NC 重启后输出报警 8065 并禁用 NC 启动。在此情况下，必须更正日期并且再次执行 NC 重启以清除报警。

如果在更正日期时无意中设置了未来的日期，系统会输出报警 8066。此时如未执行 NC 重启，则仍可更正日期。执行 NC 重启后，所设的未来日期被视作实际生效日期，无法再重设。

注意
使用期限缩短 执行 NC 重启后，如果所设的未来日期早于锁定日期，则会缩短系统在锁定日期届满之前的使用期限。如果该未来日期等于或晚于锁定日期，系统会输出报警 8064 并禁用 NC 启动。 为此，请确保在执行 NC 重启之前正确设置日期。

其他信息

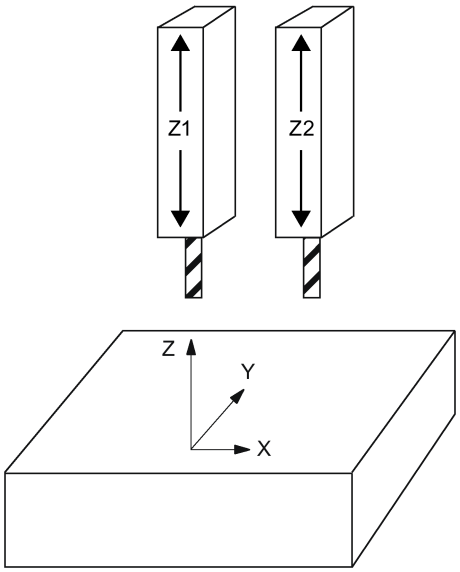
文献：

- 通讯工具 Access MyMachine（AMM）在线帮助
- PLC Programming Tool 在线帮助

27.17 切换几何轴

可使用 GEOAX 指令来指定将哪些通道轴分配给通道的几何轴。

如下图所示，可使用 Z1 轴作为第三几何轴，以几何轴名称“Z”有效，该 Z 轴与 X1 轴和 Y1 轴形成一几何轴坐标系进行插补运动。或者，可使用 Z2 轴作为第三几何轴，以几何轴名称“Z”有效，同样，该 Z 轴与 X1 轴和 Y1 轴形成一几何轴坐标系进行插补运动。这样便可在零件程序中交替使用 Z1 轴与 Z2 轴作为几何轴 Z 轴。



切换几何轴相关的机床数据

工件几何数据通过几何轴构成的坐标系描述。每根几何轴均被指定一根通道轴，每根通道轴均被指定一根机床轴。

切换几何轴的具体分配方式如下：

MD10000 AXCONF_MACHAX_NAME_TAB	机床轴名称
MD20050 AXCONF_GEOAX_ASSIGN_TAB	分配几何轴给通道轴
MD20060 AXCONF_GEOAX_NAME_TAB	通道内的几何轴名称
MD20070 AXCONF_MACHAX_USED	通道内有效的机床轴号
MD20080 AXCONF_CHANAX_NAME_TAB	通道内的机床轴名称

格式

GEOAX 指令的格式如下：

```
GEOAX（n, 通道轴名称）          ；激活几何轴切换功能
GEOAX()                         ；取消几何轴切换功能
```

n = 1 至 3：被指定一根通道轴的几何轴的编号

通道轴名称：被指定给一根几何轴的通道轴的名称

示例

先决条件：

- MD10000[4] = MZ2
- MD20080[0] = X1
- MD20080[1] = Y1
- MD20080[2] = Z1
- MD20080[4] = Z2

下面以上图中所示工件加工过程为例说明如何切换几何轴。

```
GEOAX()                    ; 激活几何轴的基本配置
G0 X0Y0Z0                  ; 默认情况下，使用通道轴 Z1 作为第三几何轴 Z
GEOAX(3, Z2)               ; 使用通道轴 Z2 作为第三几何轴 Z
G0 Z100                    ; 对 Z2 轴调用原先用于 Z 轴的刀具移动指令；而后 Z2 轴移动至 WCS 中的 Z100 位置
GEOAX(3, Z1)               ; 使用通道轴 Z1 作为第三几何轴 Z
G0 Z100                    ; 对 Z1 轴调用原先用于 Z 轴的刀具移动指令；而后 Z1 轴移动至 WCS 中的 Z100 位置
GEOAX()                    ; 取消几何轴切换功能
```

说明

所指定的通道轴名称与几何轴名称必须不同；否则会触发报警 14414“几何轴功能：错误调用”。

- MD20080 AXCONF_CHANAX_NAME_TAB
- MD20060 AXCONF_GEOAX_NAME_TAB

说明

GEOAX 指令用于切换几何轴。对于附加轴，仅在其被指定了一根几何轴之后，才能使用 GEOAX 指令对该附加轴进行切换。

GEOAX 指令仅用于切换在数控系统开机时已经存在的几何轴，而非用于创建新的几何轴。

GEOAX() 指令用于调用几何轴的基本配置，即，还原其在调用 GEOAX 指令前的初始配置状态。如在调用 GEOAX 指令后没有调用 GEOAX() 指令，则几何轴切换操作在执行另一零件程序时仍然保持有效。

27.18 循环保护

概述

说明

使用该功能需要“用户循环保护”选件（订货号：6FC5800-0AP54-0YB0）。

通过循环保护功能，您可以使用 Access MyMachine (AMM) 工具对循环文件（.SPF）进行加密并将生成的加密循环文件（.CPF）传输至数控系统中，从而实现对循环的加密和保护。

为了保护制造商的技术，禁止对受保护循环的内容进行任何形式的显示或查看。如果需要维修作业，机床制造商必须提供未经加密的循环。

说明

采用加密不会违反出口限制或禁运规定。

说明

最终用户

如果使用机床制造商的加密循环，在出现问题的情况下只能联系机床制造商的相关服务人员请求维修。

机床制造商

机床制造商在使用加密循环时，必须确保对原始的、未加密的循环进行存档并采取适当的版本管理。

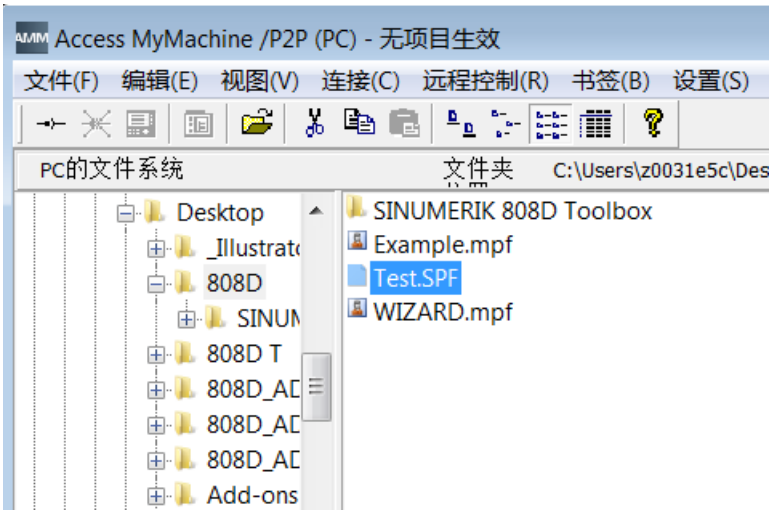
前提条件

- 用户循环保护功能选件必须通过许可证密钥激活。有关激活选件功能的更多信息，请参见章节“激活可选功能 (页 395)”。
- AMM 工具须安装在电脑上。关于 AMM 安装的详细信息，请参见 SINUMERIK 808D ADVANCED 调试手册中的“安装软件工具”章节。

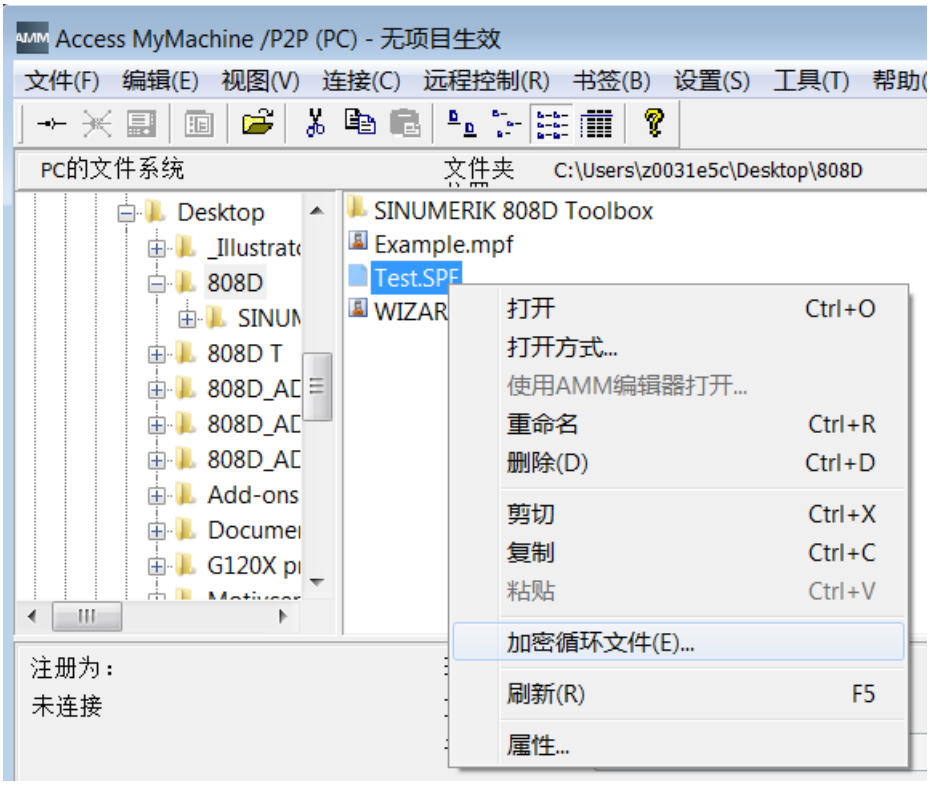
27.18.1 加密循环文件

按以下步骤使用 AMM 工具对循环文件（.SPF）进行加密：

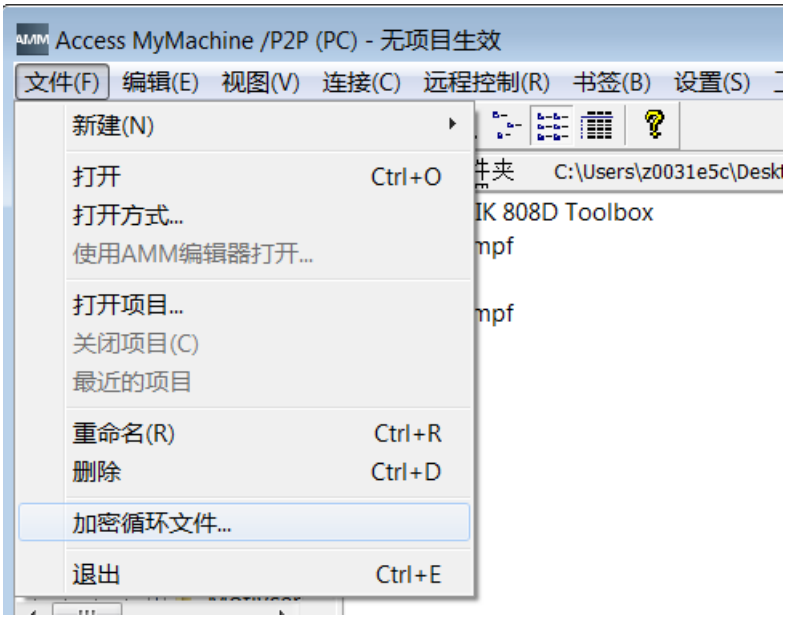
1. 在电脑上打开 Access MyMachine。
2. 从 PC 文件系统中选择所需的一个或多个循环文件（.SPF），例如：



3. 右击文件打开右键菜单，选择循环加密功能，如下图所示：



或者，还可通过主菜单选择该功能：



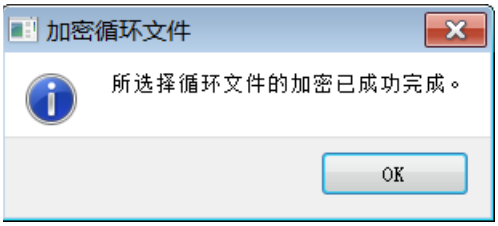
4. 在对话框中选择如下加密选项：





5. 单击该按钮，即开始加密。

加密完成后会显示如下消息，此时在原循环文件的目录下会生成一个加密文件。新生成的文件名为“原始文件名.CPF”。



27.18.2 加密循环的使用

机床制造商须将加密的循环（.CPF 文件）传输至数控系统中，并在执行加密循环前进行上电复位。如未执行上电复位，执行 .CPF 文件会触发 NC 报警 15176。

加密循环的删除和卸载操作同 .SPF 或 .MPF 文件一样。如创建了存档，则会对所有加密的 .CPF 文件进行备份。

执行加密循环

- 加密循环无法直接选中执行。只能从零件程序或直接在 MDI 中调用。
- 加密循环无法通过外部执行功能执行。

在一台或多台机床上使用加密循环

- **仅在一台机床上使用加密循环**
  如需避免在其他机床上使用加密循环，可将循环与一台机床永久绑定。为此可使用机床数据 MD18030 $MN_HW_SERIAL_NUMBER。
  数控系统启动时会将 CF 卡的唯一硬件序列号保存在该机床数据中。如要将循环与一台机床永久绑定，则必须在调用循环时查询 CF 卡的序列号（MD18030）。若循环识别到序列号不匹配，则会在循环中输出报警，从而阻止后续的执行。由于循环代码被加密，因此会形成循环与指定硬件的永久绑定。
- **在多台机床上使用加密循环**
  如要将循环与多台指定机床永久绑定，必须在循环中输入每台机床对应的硬件序列号。输入硬件序列号后的循环必须重新加密。

28 SINUMERIK 808D ADVANCED 的许可证

28.1 激活可选功能

数控系统可以选购以下选件功能：

- 附加轴
- 附加定位轴/辅助主轴 *
- 轮廓手轮
- 双向丝杠螺距误差补偿
- 通用耦合'CP-BASIC'（仅适用于车削数控系统）
- 手动机床（仅适用于车削数控系统）
- 用户循环保护
- 不带 Y 轴的端面转换/柱面转换
- 龙门轴（基础型）*
- 数控锁

*该可选功能仅适用于 PPU16x.3。

说明

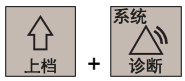
在激活手动机床功能之前，须确保机床参数 MD20050 为默认值。

说明

带 PPU16x.3 的数控系统车削版最多支持三个附加轴，铣削版最多支持两个附加轴（可配置为标准 NC 轴或定位轴）。带 PPU15x.3 的数控系统仅车削版可以支持一个附加轴（可配置为标准 NC 轴）。

如要使用已购买的选件，您必须首先使用许可证密钥在数控系统上激活选件。在购买了选件之后，您可以通过 Web License Manager 获取相应的许可证密钥。

操作步骤

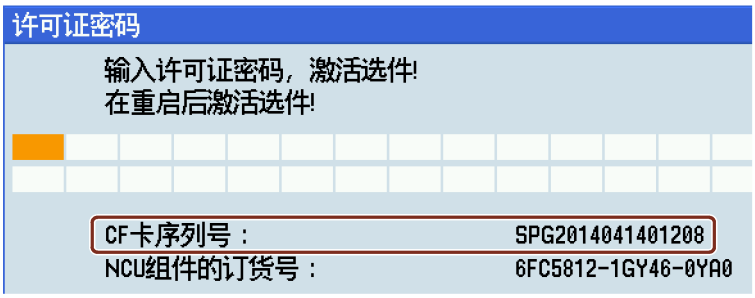




1. 选择系统数据操作区域。
2. 按下该键查看扩展软键。
3. 通过以下按键操作打开输入密钥的对话框：

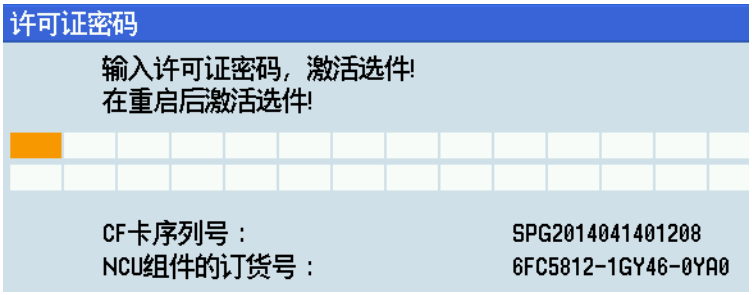


在如下对话框中可查看到 CF 卡序列号：



4. 在带有互联网接入的电脑上打开 网络许可证管理器 (http://www.siemens.com/automation/license)，并且通过"Direct access"登录。

5. 按照 Web License Manager 中的指示进行操作，将已购选件的许可证分配给指定的 PPU。在成功分配许可证后即可生成许可证密钥。
6. 在如下对话框中输入由 Web License Manager 生成的许可证密钥：



7. 按下该软键确认输入。



8. 按下该软键打开可用选件窗口。您可以在此窗口中查看选件的许可状态，例如：





如果有一个或多个附加轴已获得许可，则在许可状态栏中显示相应的许可轴数。对于其他已获许可的选件，在该栏中显示“☑”符号。

说明：该窗口中显示的最后一个选件（见下图）表明了当前数控系统的版本信息和许可状态，例如：



如果您的数控系统的许可状态存在任何问题，请联系西门子维护人员。

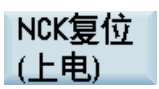
9. 对已获许可的选件进行设置。
   - 对于附加轴，在设置栏中输入所需的轴数并按下键：



   - 对于其他选件，按下键进行选择：



10. 按该软键重新启动 NCK，由此，已获许可的选件即被激活。



说明

如要使用数控锁功能，在激活数控锁选件后还须执行进一步激活操作。更多信息请参见章节“数控锁功能 (页 381)”。

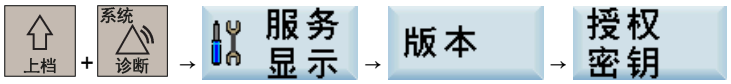
28.2 Internet 链接

所使用的 Internet 链接一览：

编号	主题栏	地址
1	网络许可证管理器	http://www.siemens.com/automation/license
2	西门子 A&D Mall: 客户登录	http://mall.automation.siemens.com
3	下载服务器	http://software-download.automation.siemens.com

28.3 重要授权条款

下面的条款对于理解 SINUMERIK 软件产品的许可证管理非常重要。

术语	描述
软件产品	安装在硬件上、用于处理数据的产品通称为软件产品。在 SINUMERIK 软件产品的许可证管理范围内，使用每个软件产品都需要相应的许可证。
硬件	根据其唯一标识将许可证分配给 SINUMERIK 数控系统的组件，这些组件在 SINUMERIK 软件产品的许可证管理范畴内称为硬件。许可证信息也记忆存储在这些组件上。 • SINUMERIK 808D ADVANCED：系统 CF 卡
许可证	许可证授权于使用软件产品的权限。这些权限的代表有： • CoL（许可证书） • 许可证密钥
CoL （许可证书）	CoL 是对许可证的证明。产品仅允许由许可证的所有者或授权人员使用。CoL 还包含了下列对于许可证管理非常重要的数据： • 产品名称 • 许可证号 • 交货单号 • 硬件序列号 **说明**： 硬件序列号仅位于系统软件的 CoL 上或在许可证捆绑时，即系统软件连同选购件一起订购。
许可证号	许可证号是许可证的标志，通过此标志识别许可证的唯一性。
系统 CF 卡	系统 CF 卡作为所有 SINUMERIK 数控系统附加的数据载体体现了数控系统的一致性。系统 CF 卡还包含了下列对于许可证管理非常重要的数据： • 硬件序列号 • 许可证信息包括授权密钥
硬件序列号	硬件序列号是系统 CF 卡不变的组成部分。通过它识别数控系统的唯一性。硬件序列号可通过下列各项确定： • CoL（参见：许可证证书> “说明”） • HMI 用户界面（在 PPU 上执行如下操作）  • 系统 CF 卡上的印刷
许可证密钥	授权密钥是所有许可证总和的“技术代表”，它分配给一个确定的、通过其硬件序列号唯一标记的硬件。
选件	选件是在基本版中不包含且为了使用必须购买许可证的 SINUMERIK 软件产品。
产品	在 SINUMERIK 软件产品许可证管理范围内，通过下列数据标明产品： • 产品名称 • 订货号 • 许可证号

A 附录

系统变量列表

系统变量	描述
$AA_FIX_POINT_SELECTED [<Axis>]	指定固定点的编号
$AA_FIX_POINT_ACT [<Axis>]	当前轴所处的固定点的编号
$P_PROG_EVENT	事件控制的程序调用有效
$P_SEARCH_S	搜索：转速，切削速率
$P_SEARCH_SDIR	程序段搜索:零件程序中编程的主轴旋转方向
$P_SEARCH_SGEAR	搜索：变速档 M 代码
$P_SEARCH_SPOS	搜索：主轴位置，路径
$P_SEARCH_SPOSMODE	搜索：位置趋近模式
$AA_ENC_COMP_MIN	EEC 表：起始位置
$AA_ENC_COMP_MAX	EEC 表：结束位置
$AA_ENC_COMP	EEC 表：补偿值
$AN_CEC[<t>,<N>]	补偿表[<t>]中插补点<N>的补偿值
$AN_CEC_INPUT_AXIS[<t>]	参照轴
$AN_CEC_OUTPUT_AXIS[<t>]	补偿轴
$AN_CEC_STEP[<t>]	控制点间距
$AN_CEC_MIN[<t>]	初始位置
$AN_CEC_MAX[<t>]	结束位置
$AN_CEC_DIRECTION[<t>]	双向补偿
$AN_CEC_IS_MODULO[<t>]	带模数功能的补偿
$AC_MEA[1]	查询测量任务的状态信号
$AA_MM[axis]	存取机床坐标系中的测量结果
$AA_MW[axis]	存取工件坐标系中的测量结果
$C_T	地址 T 的循环参数
$P_ISTEST	程序测试状态；布尔变量
$P_SEARCH	程序查找状态；布尔变量
$P_SEARCHL	程序查找状态；实数：1-, 2-, 3-
$P_TOOLNO	主轴刀架中的刀具编号
$P_TOOLP	编程刀具编号
$C_T	编程刀具编号 当程序代码 T 调用使用 MD10717 定义的换刀循环时 $P_TOOLP 无效。随后用“$C_T”表示刀具编号。
$TC_DP1[Tool number, 1]	刀具类型
$TC_DP3[Tool number, 1]	刀具的几何参数：刀具长度 1
$TC_DP6[Tool number, 1]	刀具的几何参数：刀具半径
$TC_DP12[Tool number, 1]	刀具磨损：长度 1 方向
$TC_DP15[Tool number, 1]	刀具磨损：半径方向
$TC_DP24[Tool number, 1]	刀具尺寸： 0：正常 1：过大
$TC_DP25[Tool number, 1]	刀架编号
_TM[n]	全局用户数据（整型）

系统变量	描述
_ZSFR[n]	全局用户数据（浮点） **注意**： 由于此数据已用于西门子标准技术循环，所以请务必在使用此数据时确保不存在与技术循环之间的冲突。
$AC_OPERATING_TIME	“AUTO”方式下运行程序的总时间
$AC_CYCLE_TIME	所选程序的运行时间
$AC_CUTTING_TIME	所选程序的进给时间（G01、G02 和 G03）
$AN_SETUP_TIME	从最末次使用默认值启动开始所消耗的时间
$AN_POWERON_TIME	从最末次正常启动开始所消耗的时间
$AC_REQUIRED_PARTS	待被计数的所需零件 通过将 MD27880 位 0 设为 1 来激活： • 位 1 = 0：如果“**零件数**” = “**需要的零件**”，则报警或接口信号 DB3300.DBX4001.1 = 1
$AC_TOTAL_PARTS	所计零件总数 通过将 MD27880 位 4 设为 1 来激活： • 位 5 = 0：M02/M30 将“**零件总数**”增加至“**1**” • 位 5 = 1：由 MD27882 定义的 M 代码将“**零件总数**”增加至“**1**” • 位 6 = 0/1：当“**程序测试**”无效时，计数器不工作。
$AC_ACTUAL_PARTS	实际所计零件数 通过将 MD27880 位 8 设为 1 来激活： • 位 9 = 0：M02/M30 将“**零件总数**”增加至“**1**” • 位 9 = 1：由 MD27882 定义的 M 代码将“**零件总数**”增加至“**1**” • 位 10 = 0/1：当“**程序测试**”无效时，计数器不工作。
$A_OUT	数字量输出
$AA_COUP_ACT [轴名称]	联动轴的/主轴的当前耦合状态
$AA_ACCLIMA	ACCLIMA 所设定的主运行加速补偿
$AA_VELOLIMA	VELOLIMA 所设定的主运行速度补偿
$PA_ACCLIMA	ACCLIMA 所设定的预处理加速补偿
$PA_VELOLIMA	VELOLIMA 所设定的预处理速度补偿
$AC_PLTBB	从基本坐标系统程序段开头起的轨迹长度
$AC_PATHACC	实时事件的轨迹加速度
$AC_PATHJERK	实时事件的轨迹急动度
$AC_PATHN	标准轨迹参数
$AC_PLTEB	从基本坐标系统程序段末尾起的轨迹长度
$AA_SNGLAX_STAT	显示 PLC 控制轴的状态
$AA_MOTEND	单轴插补时的当前运动终端标准
$P_COUP_OFFS [联动主轴]	程序中编写的同步主轴偏移
$AA_COUP_OFFS [联动主轴]	同步主轴位置偏移（设定值）
$VA_COUP_OFFS [联动主轴]	同步主轴位置偏移（实际值）

商标

所有带有标记符号 ® 的都是 Siemens AG 的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
德国

功能手册
6FC5397-7EP40-0RA4, 07/2018