

SIEMENS

SINUMERIK

SINUMERIK 808D 功能手册

操作说明

前言	1
各种接口信号	2
进给轴监控	3
连续路径加工，准停方式和 预读	4
加速度	5
手动运行和手轮运行	6
输出给 PLC 的辅助功能	7
运行方式，程序运行	8
补偿	9
测量	10
急停	11
回参考点	12
主轴	13
进给	14
刀具：刀具补偿	15
特殊功能	16
SINUMERIK 808D 的许可证	17

适用于：
SINUMERIK 808D 车削（软件版本：V4.4.2）
SINUMERIK 808D 铣削（软件版本：V4.4.2）

目标使用人群：
设计人员、机床制造商的技术人员、数控系统/机床的
调试人员和编程人员




12/2012

6FC5397-2EP10-0RA0

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自带的文件说明，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。

商标

所有带有标记符号®的都是西门子股份有限公司的注册商标。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。

前言

SINUMERIK 808D 文档

SINUMERIK 808D 的文档包括：

- 操作说明
 - 机械安装手册
 - 电气安装手册
 - PLC 子程序库手册
 - 功能手册
 - 参数手册
- 诊断手册
- 调试手册
- 编程和操作手册（车削）
- 编程和操作手册（铣削）
- Manual Machine Plus（车削）
- 操作编程在线帮助（车削）
- 操作编程在线帮助（铣削）
- Manual Machine Plus（车削）在线帮助

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

目标使用人群

设计人员、机床制造商的技术人员、数控系统/机床的调试人员和编程人员。

使用

从而使目标客户可以运行相关功能。

标准功能范畴

在现有文献中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。

数控系统有可能执行本文献中未描述的某些功能。但是这并不意味着在提供系统时必须带有这些功能，或者为其提供有关的维修服务。

同样，因为只是概要，所以该文献不包括全部类型产品的所有详细信息，也无法考虑到安装、运行和维修中可能出现的各种情况。

技术支持

热线:	+86 400-810-4288
服务与支持	<ul style="list-style-type: none">• 中国: www.siemens.com.cn/808D• 国际: http://support.automation.siemens.com

欧盟一致性声明

访问 <http://support.automation.siemens.com> 获取 EMC 指令的欧盟一致性声明。

在网页中输入搜索关键字 **15257461** 或联系您所在地区的西门子办事处。

许可证条款

SINUMERIK 808D 软件受国家和国际版权法律法规保护。未经授权地复制及销售整个软件或部分软件将受到法律制裁。这种行为会触犯刑法与民法，情节严重的可以追究其刑事责任以及/或者损害赔偿责任。

SINUMERIK 808D 软件采用的是开源软件。工具箱 DVD 光盘中的软件受相关授权法则保护。

目录

前言	3
1	13
2 各种接口信号	15
2.1 概述	15
2.2 PLC 到 NCK 的信号	16
2.2.1 存取权限	16
2.2.2 一般信号	17
2.2.3 数字驱动信号，到进给轴/主轴信号	19
2.3 NCK 到 PLC 的信号	20
2.3.1 一般信号	20
2.3.2 电子驱动信号，来自进给轴/主轴	21
2.4 PLC 到 HMI 的信号	22
2.5 HMI 到 PLC 信号	23
2.6 用户接口	24
2.6.1 概述 (OF)	24
2.6.2 PI 通讯 ASUP	26
2.6.3 从 NCK 区读取变量	27
2.6.4 写入 NCK 区的变量	29
2.7 NC 变量	31
2.8 PLC 的信号	34
3 进给轴监控	35
3.1 监控功能概述	35
3.2 运行监控功能	35
3.2.1 轮廓监控	35
3.2.2 定位监控	37
3.2.3 零速监控	39
3.2.4 夹紧监控	40
3.2.5 转速给定值监控	41
3.2.6 实际速度监控	42
3.3 静态极限监控	43
3.3.1 限位开关监控	43
3.3.2 硬件限位开关	43

3.3.3	软件限位开关.....	44
3.4	边界条件.....	46
3.5	数据表	47
3.5.1	机床数据.....	47
3.5.2	接口信号.....	48
4	连续路径加工, 准停方式和预读.....	49
4.1	概述.....	49
4.2	概述.....	49
4.3	准停.....	50
4.4	连续路径运行.....	52
4.4.1	概述.....	52
4.4.2	按过载系数降低速度.....	53
4.4.3	降低轨迹速度, 以限制过冲.....	54
4.4.4	机床轴专用的过冲限制.....	55
4.5	预读.....	55
4.6	数据表	57
4.6.1	机床数据.....	57
4.6.2	接口信号.....	58
5	加速度.....	59
5.1	加速度属性.....	59
5.2	插补级过冲限制.....	59
5.3	JOG 方式下的过冲限制.....	60
5.4	数据表	60
6	手动运行和手轮运行.....	61
6.1	JOG方式下手动操作的一般性能.....	61
6.2	持续运行.....	65
6.3	增量运行(INC).....	66
6.4	JOG方式下手轮运行.....	67
6.5	JOG 运行模式中的固定点运行.....	70
6.5.1	引言.....	70
6.5.2	功能性.....	71
6.5.3	参数设置.....	73
6.5.4	编程.....	74
6.5.5	边界条件.....	74
6.5.6	应用举例.....	75

6.6	数据表	76
6.6.1	机床数据	76
6.6.2	设定数据	77
6.6.3	接口信号	77
7	输出给 PLC 的辅助功能	79
7.1	概述	79
7.2	辅助功能编程	80
7.3	传输值和信号到 PLC 接口	81
7.4	辅助功能的分组	82
7.5	程序段搜索动作	84
7.6	辅助功能的描述	84
7.6.1	M 功能	84
7.6.2	T 功能	84
7.6.3	D 功能	85
7.6.4	H 功能	85
7.6.5	S 功能	85
7.7	数据表	86
7.7.1	机床数据	86
7.7.2	接口信号	86
8	运行方式, 程序运行	89
8.1	概述	89
8.2	运行方式	89
8.2.1	运行方式	89
8.2.2	运行方式转换	91
8.2.3	各种运行方式下的功能选择	92
8.2.4	各种运行方式下的监控	93
8.2.5	各种运行方式下的锁定功能	94
8.3	加工零件程序	95
8.3.1	程序运行和零件程序的选择	95
8.3.2	启动零件程序或者零件程序段	96
8.3.3	零件程序中断	97
8.3.4	复位指令	98
8.3.5	程序控制	98
8.3.6	程序状态	99
8.3.7	通道状态	100
8.3.8	事件控制的程序调用	101
8.3.9	异步子程序 (ASUP)	110
8.3.10	操作或程序动作的反应	112
8.3.11	程序运行时间图示例	114

8.4	程序测试.....	115
8.4.1	程序测试概述.....	115
8.4.2	程序运行时无进给轴动作(PRT).....	115
8.4.3	程序单段运行(SBL).....	116
8.4.4	按照空运行(DRY)进给处理程序.....	117
8.4.5	程序段搜索:处理特定程序段.....	118
8.4.6	零件程序段跳跃(SKIP).....	121
8.4.7	图形模拟.....	122
8.5	程序运行时间定时器.....	123
8.6	工件计数器.....	125
8.7	数据表.....	127
8.7.1	机床数据.....	127
8.7.2	设定数据.....	129
8.7.3	接口信号.....	129
9	补偿.....	133
9.1	概述.....	133
9.2	间隙补偿.....	133
9.3	可插补补偿.....	135
9.3.1	概述.....	135
9.3.2	SSFK.....	137
9.4	数据表.....	140
9.4.1	机床数据.....	140
9.4.2	接口信号.....	140
10	测量.....	141
10.1	概述.....	141
10.2	硬件前提条件.....	142
10.2.1	可使用的测量头.....	142
10.2.2	测量头连接.....	143
10.3	通道专用测量.....	144
10.3.1	测量方式.....	144
10.3.2	测量结果.....	145
10.4	测量精度和测试.....	146
10.4.1	测量精度.....	146
10.4.2	测量头的功能测试.....	146
10.5	JOG 方式下刀具测量.....	148
10.6	数据表.....	152
10.6.1	机床数据.....	152
10.6.2	接口信号.....	152

11	急停	153
11.1	概述	153
11.2	急停运行	154
11.3	急停响应	155
11.4	数据表	156
11.4.1	机床数据	156
11.4.2	接口信号	156
12	回参考点	157
12.1	基础部分	157
12.2	在增量测量系统中回参考点	159
12.3	有间距编码参考标记时回参考点	163
12.3.1	概述	163
12.3.2	基础参数设置	163
12.3.3	时序过程	165
12.3.4	阶段 1: 同步驶过参考标记	166
12.3.5	阶段 2: 运行至目标点	168
12.4	绝对编码器的边界条件	170
12.5	数据表	171
12.5.1	机床数据	171
12.5.2	接口信号	172
13	主轴	173
13.1	概述	173
13.2	主轴运行方式	174
13.2.1	主轴运行方式	174
13.2.2	主轴控制方式运行	176
13.2.3	主轴摆动方式运行	177
13.2.4	主轴定位方式运行	180
13.3	同步	184
13.4	变速换档	186
13.5	编程	191
13.6	主轴监控	192
13.6.1	主轴监控	192
13.6.2	轴/主轴停止	193
13.6.3	主轴在给定值范围	193
13.6.4	最大主轴转速	193
13.6.5	变速档的最大/最小转速	194
13.6.6	最大编码器极限频率	195

13.6.7	目标定位监控.....	195
13.7	模拟主轴.....	196
13.8	数据表	197
13.8.1	机床数据.....	197
13.8.2	设定数据.....	198
13.8.3	接口信号.....	199
14	进给	201
14.1	轨迹进给率 F.....	201
14.1.1	轨迹进给率 F.....	201
14.1.2	G33, G34, G35 (螺纹加工)时的进给率	203
14.1.3	G63 (带补偿夹具进行攻丝)时的进给率.....	205
14.1.4	G331, G332 (不带补偿夹具进行攻丝)时的进给率.....	206
14.1.5	倒角/倒圆时的进给率: FRC, FRCM.....	207
14.2	快速移动 G0.....	208
14.3	进给率控制	209
14.3.1	概览.....	209
14.3.2	进给锁定和进给/主轴停止.....	210
14.3.3	通过机床控制面板进行进给修调.....	211
14.4	数据表	213
14.4.1	机床数据/设定数据	213
14.4.2	接口信号.....	213
15	刀具: 刀具补偿	215
15.1	概述: 刀具和刀具补偿	215
15.2	刀具.....	216
15.3	刀具补偿.....	216
15.4	刀具补偿的特殊处理.....	217
15.5	数据表	220
15.5.1	机床数据.....	220
15.5.2	接口信号.....	220
16	特殊功能.....	221
16.1	调用在线帮助.....	221
16.2	调用具备辅助功能的标准循环.....	228
16.3	显示功能.....	230
16.4	Prog_Event 功能	234
16.5	快速输入/输出.....	235
16.6	创建用户循环.....	237

16.6.1	创建用户扩展文本文件	237
16.6.2	创建用户循环的软键索引文件.....	238
16.6.3	创建用户循环的参数文件.....	239
16.6.4	创建用户循环文件.....	241
16.6.5	创建用户循环的告警文件.....	243
16.6.6	创建用户循环的位图文件.....	243
16.6.7	传输所需的文件至数控系统	244
16.6.8	调用所创建的用户循环	246
16.6.9	编辑用户循环窗口.....	246
16.7	通过自定义 EasyXLanguage 脚本创建用户对话框.....	247
16.7.1	功能范围.....	247
16.7.2	配置基础.....	249
16.7.3	配置文件 (EasyXLanguage)	250
16.7.4	配置文件的结构	252
16.7.5	语言相关性	252
16.7.6	XML 标签.....	252
16.7.6.1	通用结构.....	252
16.7.6.2	指令/标签说明.....	254
16.7.6.3	颜色代码.....	266
16.7.6.4	专用 XML 句法	266
16.7.6.5	运算符	267
16.7.7	组件地址分配.....	267
16.7.7.1	PLC 地址分配.....	268
16.7.7.2	NC 变量地址分配.....	269
16.7.7.3	机床数据和设定数据的地址分配	269
16.7.7.4	用户数据的地址分配	270
16.7.8	创建用户菜单.....	270
16.7.8.1	创建软键菜单和对话框窗口	270
16.7.8.2	替代符号.....	287
16.7.9	预定义函数	288
17	SINUMERIK 808D 的许可证	301
17.1	SINUMERIK 808D 的许可证	301
17.2	网络许可证管理器.....	302
17.2.1	网络许可证管理器.....	302
17.2.2	分配许可证	302
17.3	激活可选功能.....	304
17.4	Internet 链接.....	306
17.5	重要授权条款.....	306
	索引	309

注释

本说明书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“进给修调”（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（说明）
 例如：MD30300 IS_ROT_AX（回转轴）
- 设定数据 -> SD SD_NR SD_NAME（说明）
 例如：SD41200 JOG_SPIND_SET_VELO（主轴的 JOG 速度）
- 章节的标题增加了带括号的标识（如章节 1：急停（N2））。这些简短标识用于与其它章节之间进行交叉引用。

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_	一般机床数据
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)
输入极限为 $\pm 4.19 \times 10^{-307}$ 到 $\pm 1.67 \times 10^{308}$
- **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)
输入值极限为 -128 到 +127 (十进制); 作为十六进制值: 00 到 FF
- **BOOLEAN**
布尔值: 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 参数手册。
- 有关报警的具体说明, 请参考 SINUMERIK 808D 诊断手册。

参见

各种接口信号 (页 15)

各种接口信号

2.1 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。

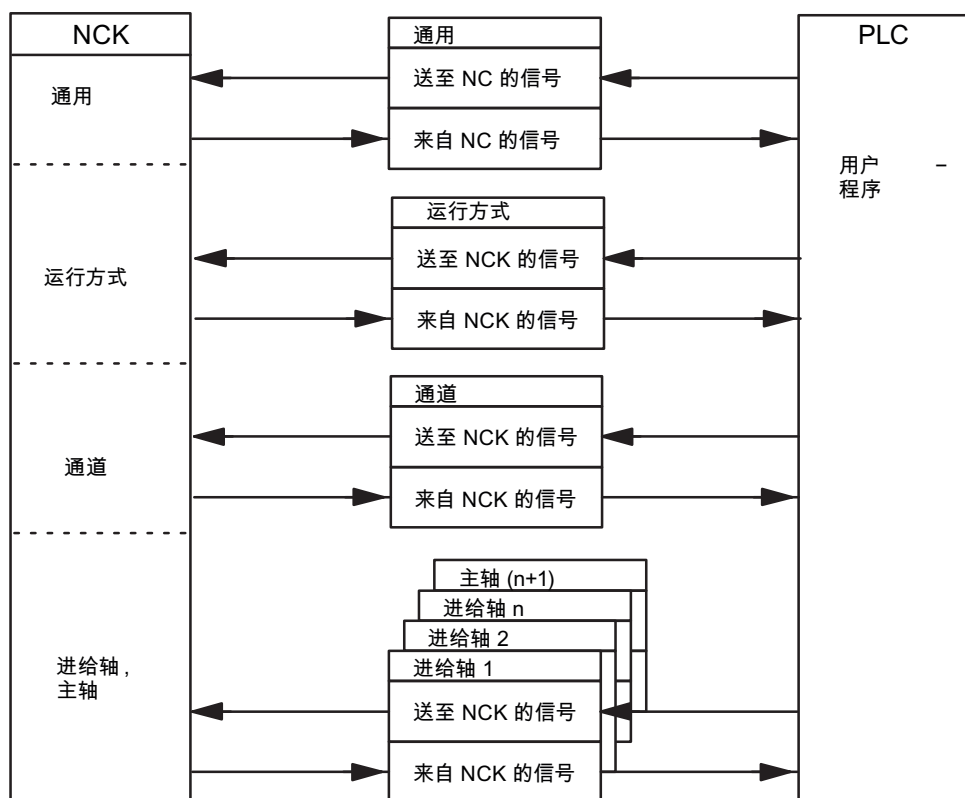


图 2-1 PLC/NCK 的接口

2.2 PLC 到 NCK 的信号

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

2.2 PLC 到 NCK 的信号

2.2.1 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。在 SINUMERIK 808D 中有一个存取等级方案用来释放数据区。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
制造商（保护级 2）	SUNRISE	OEM
用户（保护级 3）	CUSTOMER	最终用户
未设置口令	-	-

这为控制访问权限提供了一个多层级安全概念。

文献：

调试手册，章节：存取级别

2.2.2 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。

进给轴/主轴锁定 (DB380x.DBX1.3)

在测试中使用此接口信号“进给轴/主轴禁止”。

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”操作模式下运行一段新的程序段）。

注意：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：

如果驱动的控制器的使能被取消（如通过“控制器使能=0”信号或控制系统内部出现故障）或再次分配控制器的使能，接口信号“跟随模式”才有效。

接口信号“跟随模式”= 1：

如果“控制器使能”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“控制器使能”，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“控制器使能”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“控制器使能”信号设定时被补偿。轴运动将从“控制器使能”信号取消之前的设定位置开始。在“停止”状态，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。

控制器使能 (DB380x.DBX2.1)

在给驱动发出控制器使能时，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。

取消控制器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：

驱动控制器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“控制器使能”的 PLC 用户程序（正常情况）

应用：在夹紧进给轴/主轴之前取消控制器使能。

2. 如果在机床、驱动、位置测量系统以及控制器中出现不同的故障时，可以在系统内部取消控制器使能（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. 在控制系统内部的下列事件中：存在接口信号“急停”（DB2600.DBX0.1）

取消一个正在运行的进给轴/主轴的控制器使能：

- 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，启动延时调节器使能的计时器（MD36620 SERVO_DISABLE_DELAY_TIME（控制器使能断开延时））。
- 一旦实际值速度达到停止状态范围时，取消控制器使能。“转速控制器有效”（DB390x.DBX1.6）= 0 信号反馈到 PLC。驱动的控制器的使能至少在 MD36620 SERVO_DISABLE_DELAY_TIME 中设定的时间结束之后取消。
- 注意：如果将控制器使能延迟时间设置的较低，尽管轴/主轴仍然运行，控制器的使能也已被取消。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采集。

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控制器的使能，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。

2.2.3 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。

2.3 NCK 到 PLC 的信号

2.3.1 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 17) (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 17)。

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 17)。

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**。

2.3.2 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。

2.4 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。

在 SINUMERIK 808D 上，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。

2.5 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。

2.6 用户接口

2.6.1 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，因此用户只能读取该信号。

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”。

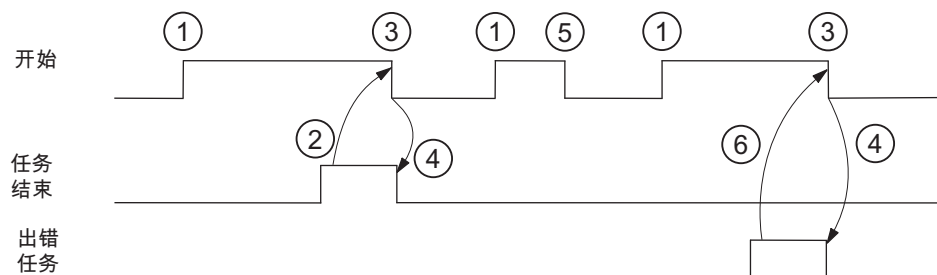


图 2-2 脉冲示意图

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

2.6.2 PI 通讯 ASUP

初始化

通过 PI 通讯 ASUP 可以从 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。
- 初始化不能在未激活的通道中进行。
- 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

2.6.3 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 31)

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， 变量专用 部分	DB120x.DBB100 0	变量索引	参见章节 NC 变量 (页 31)
	DB120x.DBB100 1	区域编号	
	DB120x.DBB100 2	NCK 变量的行索引	
	DB120x.DBB100 4	NCK 变量的列索引	
任务， 全局部分	DB1200.DBX200 0.0	任务结束	0/1
	DB1200.DBX200 0.1	任务出错	0/1
结果， 变量专用 部分	DB120x.DBX300 0.0	变量无效	0/1
	DB120x.DBB300 1	存取结果	0/3/5/10
	DB120x.DBB300 4/ DB120x.DBW300 4/ DB120x.DBD300 4	NCK 变量的值，数据类型 取决于变量索引	参见章节 NC 变量 (页 31)

2.6.4 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 31)

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， 变量专用 部分	DB120x.DBB100 0	变量索引	参见章节 NC 变量 (页 31)
	DB120x.DBB100 1	区域编号	
	DB120x.DBB100 2	NCK 变量的行索引	
	DB120x.DBB100 4	NCK 变量的列索引	
	DB120x.DBB300 4/ DB120x.DBW300 4/ DB120x.DBD300 4	NCK 变量的值，数据类型 取决于变量索引	
任务， 全局部分	DB1200.DBX200 0.0	任务结束	0/1
	DB1200.DBX200 0.1	任务出错	0/1
结果，变量专用部分	DB120x.DBX300 0.0	变量无效	0/1
	DB120x.DBB300 1	存取结果	0/3/5/10

2.7 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 216)”。

	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）

2.8 PLC 的信号

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)

2.8 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

进给轴监控

3.1 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

3.2 运行监控功能

3.2.1 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，...）。

3.2.2 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 49)”。

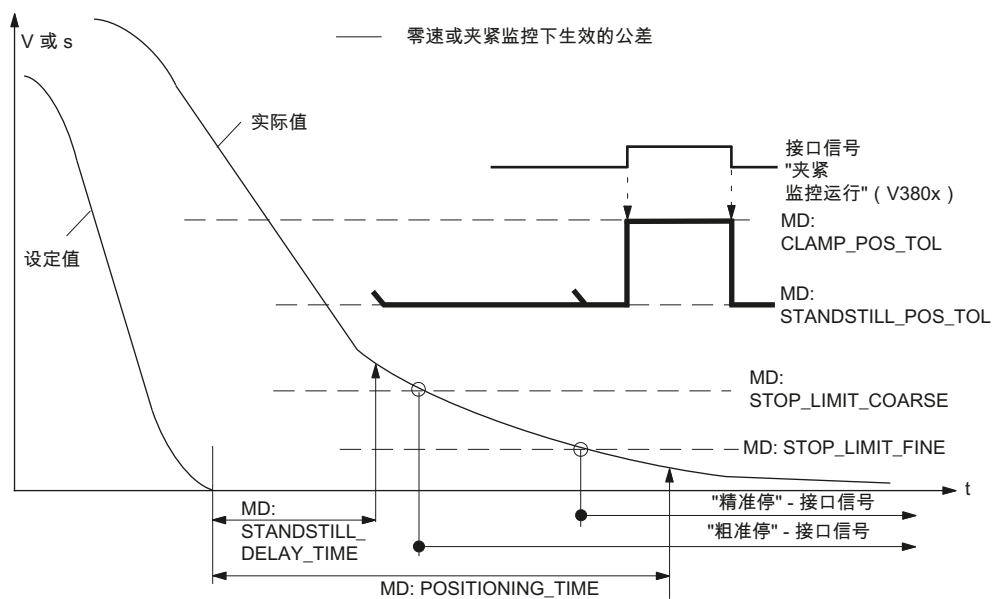


图 3-1 定位、零速度和夹紧监控的相互关系

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。

3.2.3 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

3.2 运行监控功能

3.2.4 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)。

3.2.5 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)。

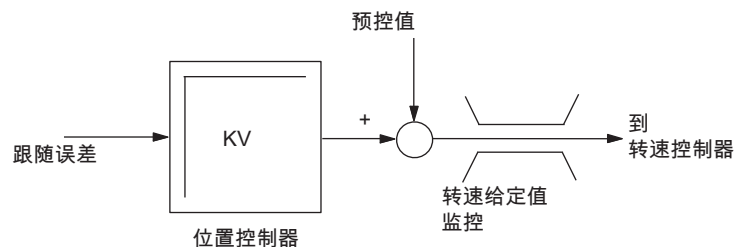


图 3-2 转速给定值监控

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/主轴和别的进给轴/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

在存取级“专家模式”（保护级 1）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。

3.2 运行监控功能

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。

3.2.6 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

3.3 静态极限监控

3.3.1 限位开关监控

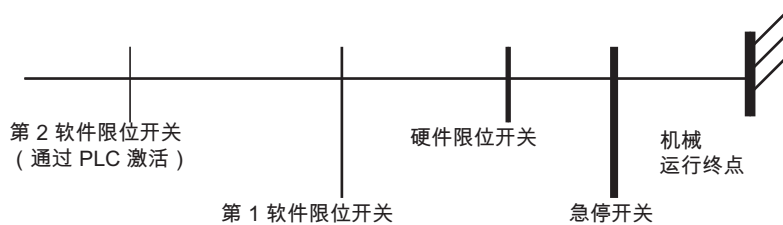


图 3-3 线性轴的限位开关一览

3.3.2 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。

3.3 静态极限监控

- 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

3.3.3 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

3.4 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：

- MD31030 LEADSCREW_PITCH (丝杠螺距)
- 变速比(减速箱):
MD31050 DRIVE_AX_RATIO_DENOM (负载变速箱分母)
MD31060 DRIVE_AX_RATIO_NUMERA (负载变速箱分子)
变速比(编码器)，只用于主轴：
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
(用于前馈控制的速度控制回路相关时间常量)
- 编码器分辨率
MD31020 ENC_RESOL[0] (每转的编码器线数)

相应的机床数据参见章节“速度、给定值/实际值系统、闭环控制(G2)”。

只适用于模拟主轴：

- 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)

3.5 数据表

3.5.1 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
36310	ENC_ZERO_MONITORING [n]	零标记监控

3.5 数据表

序号	名称	名称
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

3.5.2 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

连续路径加工，准停方式和预读

4.1 概述

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。

4.2 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。

4.3 准停

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。

4.3 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。在程序段转换时, 速度近似为零。

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。

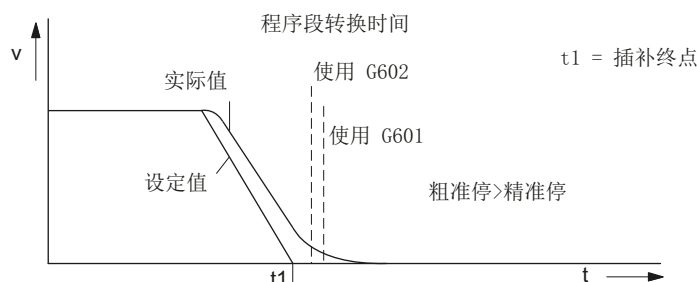


图 4-1 与精度等级相关的程序段转换

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。

4.4 连续路径运行

4.4 连续路径运行

4.4.1 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 49)”）。

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。

4.4.2 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）定义了此加速度值。

4.4 连续路径运行

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)，根据过载系数模式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。

4.4.3 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。

4.4.4 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 59)”。

4.5 预读

功能

预读是连续路径加工方式(G64)下的一个功能，它可以预先决定当前程序段以后的几个 NC 零件程序段的速度控制。

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。

4.5 预读

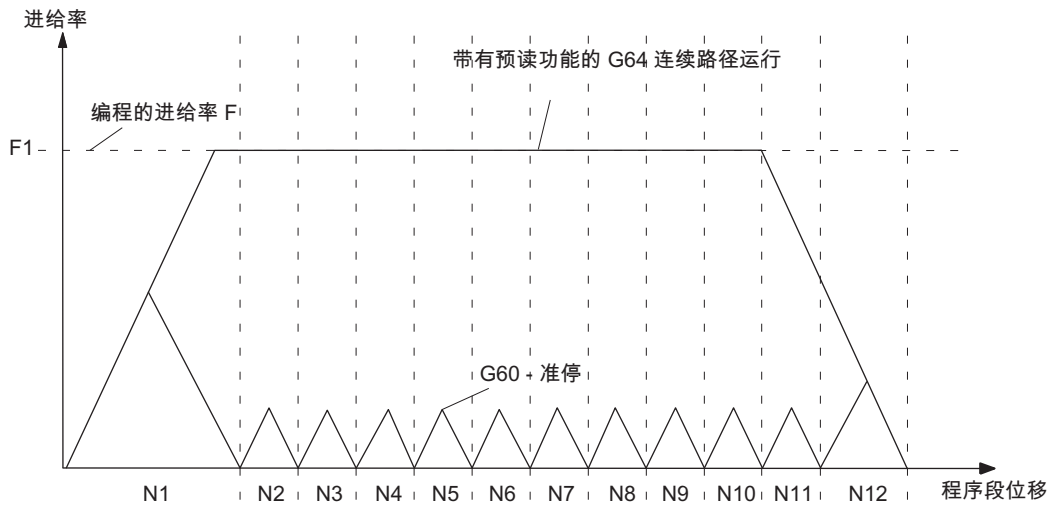


图 4-2 G60 和 G64 速度特性比较, 短行程序段

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。

4.6 数据表

4.6.1 机床数据

序号	名称	名称
通道专用		
29000	LOOKAH_NUM_CHECKED_BLOCK S	预读程序段数量
进给轴/主轴专用		
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
36000	STOP_LIMIT_COARSE	粗准停
36010	STOP_LIMIT_FINE	精准停
36020	POSITIONING_TIME	精准停延迟时间

4.6 数据表

4.6.2 接口信号

序号	位	名称
通道专用		
DB3300.DBX 0004	.3	所有轴停动
进给轴/主轴专用		
DB390x.DBX 0000	.6	采用粗准停到达位置
DB390x.DBX 0000	.7	采用精准停到达位置

加速度

5.1 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

5.2 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“自动”和“MDA”操作模式下的插补轴。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）。

5.3 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

在下面情况下过冲限制无效：

- 回参考点时。
- 产生快速停止的报警时。

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。

5.4 数据表

机床数据

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轴特定过冲

手动运行和手轮运行

6.1 JOG 方式下手动操作的一般性能

运行方式 JOG

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 (页 89)”。

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)。

6.1 JOG 方式下手动操作的一般性能

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 67)”)。

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“自动”方式转换到“手动”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 方式下轴快速速度)确定的快速速度运动。

对于几何轴, 采用机床进给轴的赋值: 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 (页 59)”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”。

6.1 JOG 方式下手动操作的一般性能

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！

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 (页 35)”。

6.2 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. → 在方式组区域：DB3000.DBB0002,
0 = 1

适用于所有进给轴

DB2600.DBX0001. → 在几何轴/进给轴区域：
0 = 0

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。

6.3 增量运行(INC)

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。

6.3 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. → 在方式组区域： DB3000.DBB0002,
0 = 1

适用于所有进给轴

DB2600.DBX0001. → 在几何轴/进给轴区域：
0 = 0

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 65)”)。

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。

6.4 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.DBX0000.0 到.2, DB3200.DBX0004.0 到.2, DB3200.DBX0008.0 到.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 方式下主菜单中的软键“手轮”,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。

6.4 JOG 方式下手轮运行

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

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。

6.5 JOG 运行模式中的固定点运行

6.5.1 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 运行模式中运行至固定点”功能只能在 "JOG" 运行模式中激活。
在 JOG-REPOS、JOG-REF 和 JOG "自动"运行模式下，不能激活此功能。
- 待运行的轴必须返回参考点。
- 不允许激活运动转换。
- 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 编程和操作手册，章节：“接近固定点”。

6.5.2 功能性

步骤

"在 JOG 运行模式下运行至固定点"的步骤：

- 选择 JOG 运行模式
- 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 运行模式设置)

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”）。

6.5.3 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 运行模式设置）

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]

6.5 JOG 运行模式中的固定点运行

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。

6.5.4 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

6.5.5 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)。

6.5.6 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 运行模式。

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）

6.6 数据表

6.6 数据表

6.6.1 机床数据

序号	名称	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 运行模式设置
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

6.6.2 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

6.6.3 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	运行方式 JOG
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	.1, .0	激活手轮 (2, 1) 用于几何轴 1
DB3200.DBX1004	.1, .0	用于几何轴 2
DB3200.DBX1008	.1, .0	用于几何轴 3
DB3200.DBX1000	.4	运行键禁用, 用于几何轴 1
DB3200.DBX1004	.4	用于几何轴 2
DB3200.DBX1008	.4	用于几何轴 3
DB3200.DBX1000	.5	快速运行叠加, 用于几何轴 1
DB3200.DBX1004	.5	用于几何轴 2
DB3200.DBX1008	.5	用于几何轴 3
DB3200.DBX1000	.7 或者 .6	正或负运行键, 用于几何轴 1
DB3200.DBX1004	.7 或者 .6	用于几何轴 2
DB3200.DBX1008	.7 或者 .6	用于几何轴 3

6.6 数据表

序号	位	名称
DB3200.DBX1000	.0 到 .6	机床功能 INC1 连续运行， 用于几何轴 1
DB3200.DBX1004	.0 到 .6	用于几何轴 2
DB3200.DBX1008	.0 到 .6	用于几何轴 3
DB3300.DBX1000	.1, .0	激活手轮 (2, 1) 用于几何轴 1
DB3300.DBX1004	.1, .0	用于几何轴 2
DB3300.DBX1008	.1, .0	用于几何轴 3
DB3300.DBX1000	.7 或者 .6	正向或负向运行指令， 用于几何轴 1
DB3300.DBX1004	.7 或者 .6	用于几何轴 2
DB3300.DBX1008	.7 或者 .6	用于几何轴 3
DB3300.DBX1001	.0 到 .6	机床功能 INC1 连续运行生效， 用于几何轴 1
DB3300.DBX1005	.0 到 .6	用于几何轴 2
DB3300.DBX1009	.0 到 .6	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
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）

输出给 PLC 的辅助功能

7.1 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制的辅助功能(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 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。

7.2 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

表格 7-1 编程的辅助功能一览表

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		
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。

表格 7-2 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。

7.3 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 52)”。

接口信号

NCK 到 PLC 的信号传输。

7.4 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 2 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 2 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。

表格 7-3 示例：机床数据的条目

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
5	6	H	1	11
6	6	H	1	12
7	7	T	0	-1

7.5 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。

7.6 辅助功能的描述

7.6.1 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 功能的范围值：0 到 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（参见编程和操作手册）。其它功能供机床生产厂商使用。

7.6.2 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。

7.6.3 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：

编程和操作手册

7.6.4 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=...)

7.6.5 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：

编程和操作手册

7.7 数据表

7.7 数据表

7.7.1 机床数据

序号	名称	名称
一般		
11100	AUXFU_MAXNUM_GROUP_ASSIGN N	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

7.7.2 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)

序号	位	名称
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)
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

7.7 数据表

运行方式，程序运行

8.1 概述

程序操作

程序运行是指在"自动"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 数控系统有一个通道。

8.2 运行方式

8.2.1 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：

- **手动**（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。
- **自动**（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。

8.2 运行方式

手动方式下允许的机床功能

在“手动”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。

MDA 方式下允许的机床功能

在“MDA”方式下可以选择下面的机床功能：

TEACH IN（插入程序段）

通过接口信号“TEACH IN”（DB3000.DBX0001.0）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 TEACH IN”（DB3100.DBX0001.0）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。

8.2.2 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”)。

表格 8-1 运行方式转换取决于通道状态

	从	AUTO		JOG			MDI	
					之前是 AUTO	之前 是 MDI		
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
之前		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。

8.2 运行方式

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。

8.2.3 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。

表格 8-2 各种运行方式下的功能选择

运行方式	自动			JOG					MDI					
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		
<p>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</p>														

8.2.4 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。

表格 8-3 监控和互锁

运行方式	自动			JOG					MDI					
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														
功能	1	2	3	1	3	4	3	5	3	1	2	3	6	7
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
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

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

8.2 运行方式

8.2.5 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					MDI					
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														

8.3 加工零件程序

8.3.1 程序运行和零件程序的选择

定义

所谓零件程序的执行, 是指在自动方式 "自动" 下加工零件程序, 或者在 "MDA" 方式下加工程序段。

影响

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:

- 操作员输入 ("加工"操作区) /程序管理器
- PLC
 - 通过“程序列表”中程序编号选定一个程序 (参见编程和操作手册)
 - 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 23)”)

8.3.2 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机床控制面板上的<循环启动>键控制。

只有在“自动”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

8.3.3 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。

8.3 加工零件程序

8.3.4 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。

8.3.5 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"程序控制"窗口("自动"方式，"加工"操作区)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。

表格 8-4 程序控制

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	

功能	选择信号	使能信号	反馈信号
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

8.3.6 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"自动"和"MDA"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。

SINUMERIK 808D 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)。

表格 8-5 对程序状态的影响

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

8.3 加工零件程序

指令	程序运行状态			
	终止	中断	停止	运行
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	

8.3.7 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“通道状态复位”(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”。

表格 8-6 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		
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

指令	通道状态		
	复位	中断	有效
信号“轴专用进给停止”			X
进给修调 = 0 %			
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

8.3.8 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。

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)

通常, 用户程序保存在路径 /_N_CMA_DIR/_N_PROG_EVENT_SPF 下。在 MD11620 PROG_EVENT_NAME 中可以给定其他的程序名。

8.3 加工零件程序

其他程序名

在 MD11620 PROG_EVENT_NAME (PROG_EVENT 的程序名) 中给定名称。随后在下列目录中按给出的顺序查找用户程序：

- /_N_CUS_DIR/ 用户循环
- /_N_CMA_DIR/ 制造商循环

在出现设计的事件时将激活第一个查找到的、带该名称的程序。

和适用于循环的情况类似，保护系统对该程序生效（读写的保护级）。

在模拟中忽略 MD20108 PROG_EVENT_MASK。

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

表格 8-7 零件程序开始时的处理顺序

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 MDA 或者 TEACH IN	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/_N_PROG_EVENT_ SPF 或者 MD11620 中的名称	作为子程序	作为子程序隐含调用路径名
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

表格 8-8 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 MDA 或者 TEACH IN	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/_N_PROG_EVENT_ SPF 或者 MD11620 中的名称	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

表格 8-9 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	

8.3 加工零件程序

过程	指令	边界条件 (必须在指令发出前满足)	附注
4	/_N_CMA_DIR/_N_PROG_EVENT_SPF 或者 MD11620 中的名称	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)

表格 8-10 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/_N_PROG_EVENT_SPF 或者 MD11620 中的名称	作为 ASUP	作为 ASUP 隐含调用路径名
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

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:

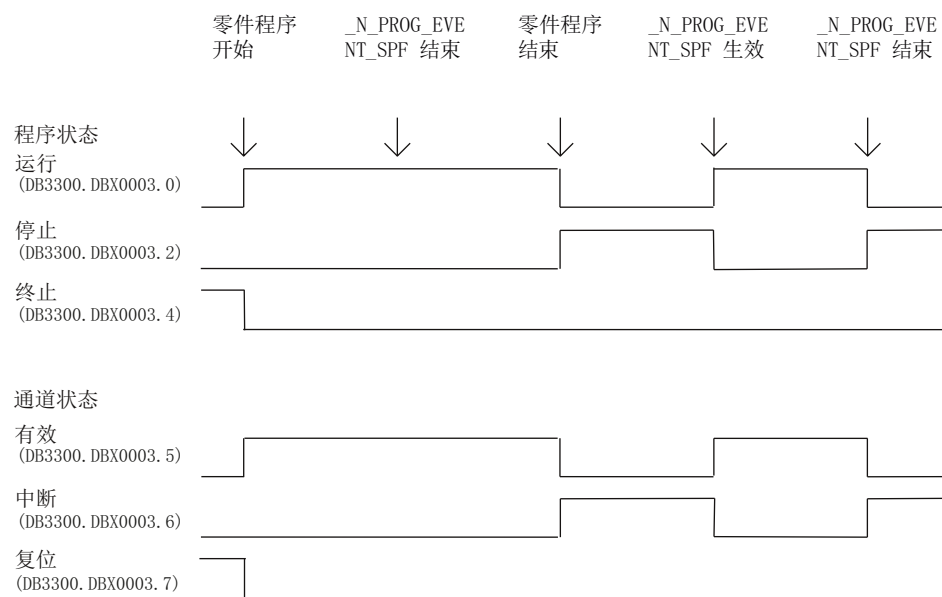


图 8-1 程序状态和通道状态的接口信号的时序(1)

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:

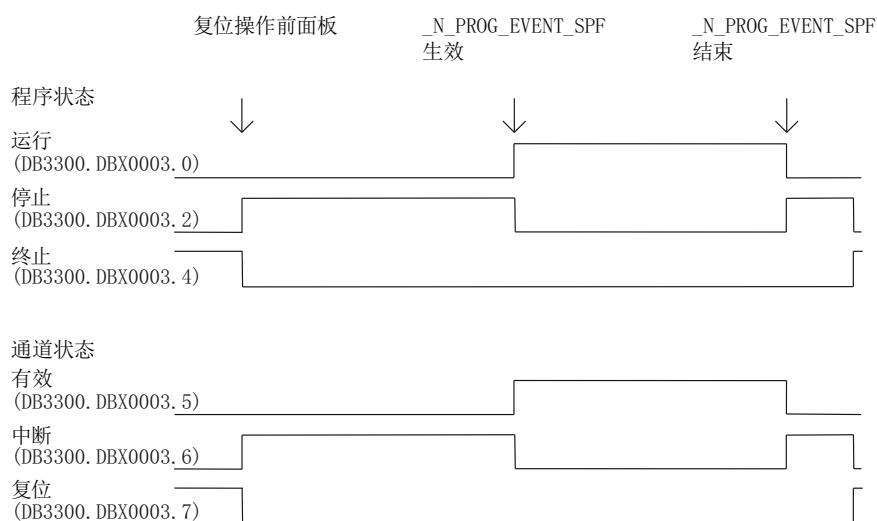


图 8-2 程序状态和通道状态的接口信号的时序(2)

说明

_N_PROG_EVENT_SPF 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。

特点

用户程序 _N_PROG_EVENT_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 _N_PROG_EVENT_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/_N_PROG_EVENT_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, `_N_PROG_EVENT_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, `_N_PROG_EVENT_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”

8.3 加工零件程序

零件程序启动时：

/_N_CMA_DIR/_N_PROG_EVENT_SPF 作为子程序被执行。_N_PROG_EVENT_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 _N_PROG_EVENT_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
_N_PROG_EVENT_SPF：

```
PROC PROG_EVENT DISPLOF
```

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:

- 该程序的名称位于 MD11620 PROG_EVENT_NAME (程序事件的程序名) 中并且保存在目录 /_N_CUS_DIR/ 或者 _N_CMA_DIR/ 下
- _N_PROG_EVENT_SPF (预设)

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 (参见 MD11620 PROG_EVENT_NAME)。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
-

8.3.9 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。

说明

在启动 ASUP 前必须结束 PI 通讯 ASUP 的调用。

初始化

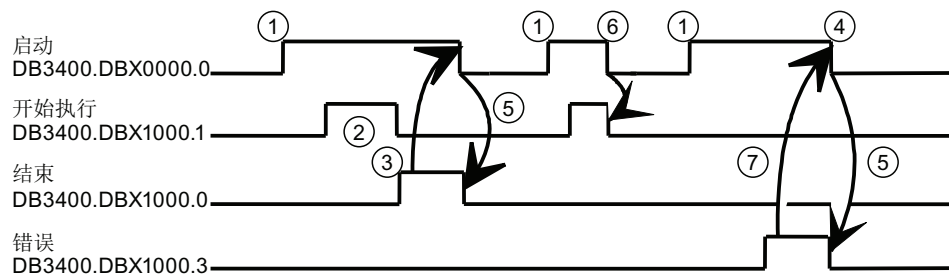
通过 PI 通讯 ASUP 进行初始化。

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。

表格 8- 11 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



- 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

图 8-3 PLCASUP1_SPF 的脉冲示意图

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

8.3 加工零件程序

- 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

8.3.10 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。

表格 8- 12 操作或程序动作的反应

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)

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	
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:

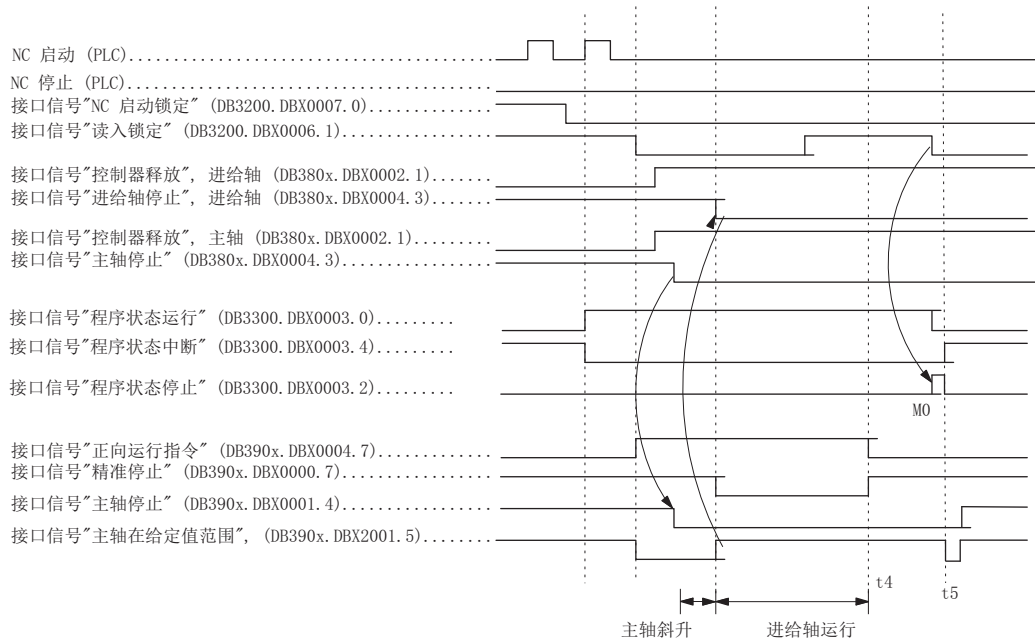
A:自动

M:MDA

J:JOG

8.3 加工零件程序

8.3.11 程序运行时间图示例



说明：
 通过PLC 用户程序生成的控制输入信号。
 t_4 : 通过“读入锁定”阻止程序段切换到 N20，
 t_5 : 使用 RESET 中断程序

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

图 8-4 程序执行过程中信号举例

8.4 程序测试

8.4.1 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

8.4.2 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

在操作面板“程序控制”窗口中选择程序测试功能。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。

8.4 程序测试

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)。

8.4.3 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：

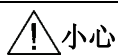
- 粗略单程序段

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”。



小心
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)

8.4.4 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“加工”操作区下按软键“程序控制”(“自动”方式), 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“设定数据”窗口中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”。

8.4.5 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 "自动" 模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。

- 无计算搜索程序段

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（参图）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)

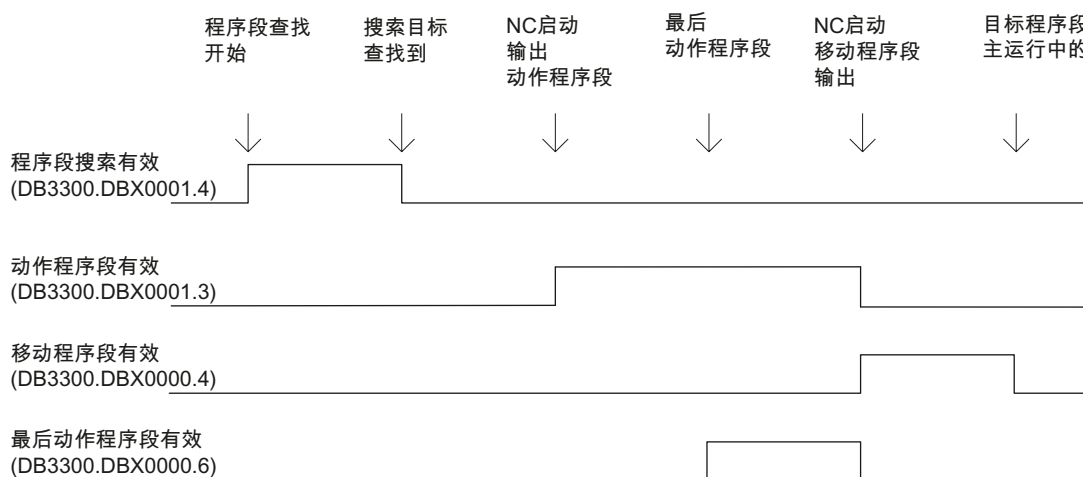


图 8-5 接口信号的时间顺序

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。

8.4 程序测试

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

如果要在程序段搜索之后激活 PLC 动作，应设置信号“最后动作程序段有效”。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。

边界条件

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 编程和操作手册。

8.4.6 零件程序段跳跃(SKIP)

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。

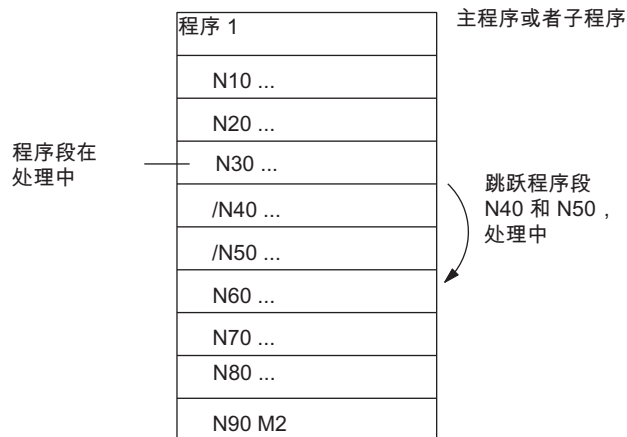


图 8-6 程序段跳跃

选择/激活

在操作面板“程序控制”窗口中选择程序段跳跃功能。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“SKIP”。

8.4 程序测试

8.4.7 图形模拟

功能

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

选择/取消

选择"编程"操作区，载入程序并使用“模拟”软键可以实现对所选程序的图形模拟。同时，设置接口信号“模拟有效”(DB1900.DBX0000.6)且当退出“程序”操作区或切换到“编辑”时，接口信号被复位。

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 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)等退出模拟功能，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 参数手册

8.5 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。

其中有些计时器始终有效。而其它计时器需要由机床数据激活。

计时器 - 始终有效

- 自最后一次“以默认值控制上电起的时间”（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。

- 自最后一次控制上电起的时间（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。

可以取消的定时器

下列定时器通过机床数据来激活(缺省设定)。开始是计时器专用的。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。

当空运行进给和程序测试功能有效时，时间测量的使能可以由机床数据定义。

- 数控系统程序在“自动”方式下总的运行时间(以秒为单位)：

\$AC_OPERATING_TIME

在“自动”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。

8.5 程序运行时间定时器

- 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

“偏置”操作区-> 软键“设定数据”-> 软键“时间计数器”, 显示计时器的内容:

- 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“AUTO”窗口中的信息行里。

文献:

编程和操作手册

8.6 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，可以通过程序或操作面板来读/写(遵守写的保护级)。

值范围： 0 到 999 999 999。

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)一致时可以定义一个报警输出。用户必须自行将该计数器归零。

8.6 工件计数器

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

“偏置”操作区-> 软键“设定数据”-> 软键“时间计数器”，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”也显示在“加工”操作区“Auto”窗口的信息行中。

文献：

编程和操作手册

8.7 数据表

8.7.1 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
11620	PROG_EVENT_NAME	程序事件的程序名称

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
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

8.7 数据表

序号	名称	名称
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

8.7.2 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

8.7.3 接口信号

运行方式信号

序号	位	名称
PLC 到 NCK		
DB3000.DBX0000	.0	自动方式
DB3000.DBX0000	.1	MDA 模式
DB3000.DBX0000	.2	运行方式 JOG
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: 自动方式
DB3100.DBX0000	.1	有效方式: MDA
DB3100.DBX0000	.2	有效方式: JOG

8.7 数据表

序号	位	名称
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

序号	位	名称
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
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

8.7 数据表

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	

补偿

9.1 概述

补偿

对于 SINUMERIK 808D，可以对进给轴进行以下轴专有补偿：

- 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）

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。补偿值显示在操作区域“系统”->“服务显示”->“轴信息”下的“绝对补偿值测量系统 1”中。

9.2 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位移行程的出错。比如，一进给轴在换向时少走或多走了一个间隙值（参下图）。

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。

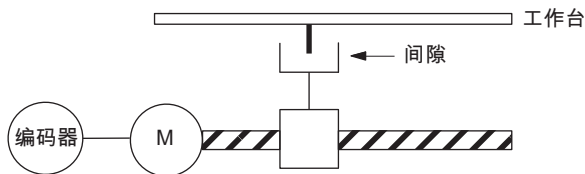
该值的大小可以在系统开机调试时在机床数据 MD32450 BACKLASH (反向间隙)中定义。

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（参下图）。在此，输入一个**正值**的间隙补偿值（正常状态）。



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

图 9-1 正向间隙(正常情况)

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。

9.3 可插补补偿

9.3.1 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 HMI 的串行口从操作区"系统" -> "系统数据" / NCK/PLC 数据：螺距误差补偿 来实现并在编辑后再载回。

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。

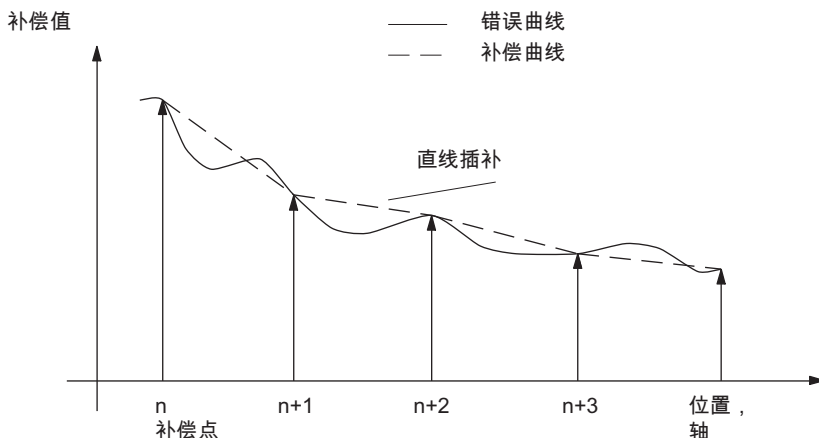


图 9-2 补偿点之间的线性插补

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 SSK 时（在回参考点运行之后）产生位置跃变。

9.3.2 SSKF

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.DBX0000.4 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 125 个插补点(N = 0...124)。

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：
 $\$AA_ENC_COMP [0,N,AXi]= \dots$

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：** $\$AA_ENC_COMP_STEP[0,AXi]= \dots$

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：** $\$AA_ENC_COMP_MIN[0,AXi]= \dots$

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：** $\$AA_ENC_COMP_MAX[0,AXi]= \dots$

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 $\$AA_ENC_COMP[0,0,AXi])$ 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。



小心

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”。

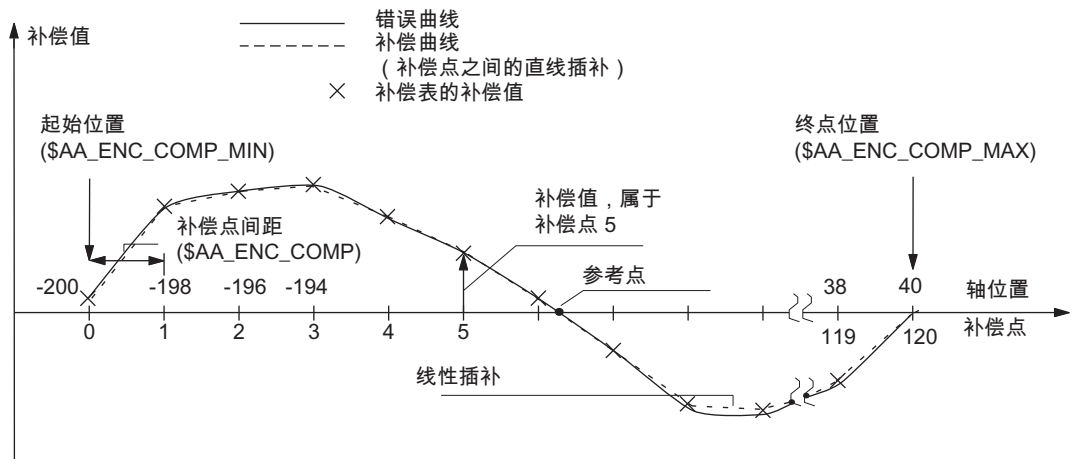


图 9-3 补偿表的参数(SSFK 系统变量)

9.4 数据表

9.4.1 机床数据

序号	名称	名称
轴专用		
32450	BACKLASH[0]	背隙
32630	ACTIVATION_MODE	通过程序使能前馈控制
32700	ENC_COMP_ENABLE[0]	插补补偿有效
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)

9.4.2 接口信号

序号	位	名称
进给轴/主轴专用		
DB390x.DB X0000	.4	回参考点/同步 1

测量

10.1 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"自动"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 SINUMERIK 808D 中要完成一提示系统，该提示系统在"手动"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 编程和操作手册。

说明

仅在铣床上提供自动测量功能。

10.2 硬件前提条件

10.2.1 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）。

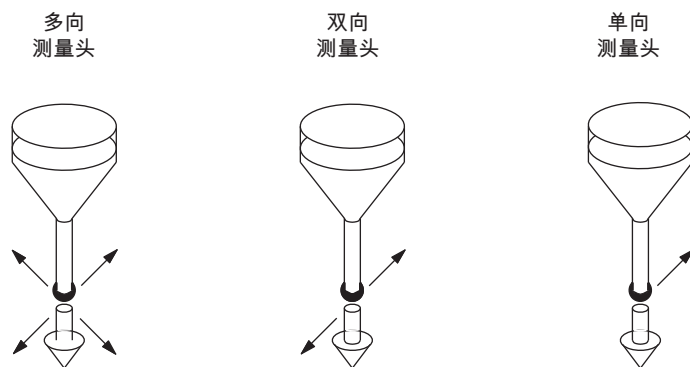


图 10-1 测量头类型

表格 10-1 测量头类型分配

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。

10.2.2

测量头连接

通过 X21 的引脚 4 (DI1) 和引脚 5 (DI2) 连接用于 SINUMERIK 808D 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 电气安装手册

10.3 通道专用测量

10.3.1 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 的下降沿(-)：从偏转状态到非偏转状态。

当测量头信号到达或位置到达时，测量程序段即完成。使用“复位”键可以取消测量功能。

文献：

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。

10.3.2 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:

编程和操作手册

PLC 服务显示

通过操作区域 "SYSTEM" → "PLC" 中的 "PLC 状态" 菜单可以控制测量信号:

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)。

10.4 测量精度和测试

10.4.1 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。

10.4.2 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: MESSWERT_IN_X
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。

10.5 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" 模式中提供了软键和模板供用户使用。并可以帮助刀具测量。

文献：

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。请遵守制造商文献！

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。然后数控系统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。

测量按键

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

- 选择 "JOG" 运行方式。
- 使用"设定"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。

- 按软键"测量头数据"，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。

- 按软键"测量工具"-> "调节测量头"，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一而不是刀具补偿。

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。

- 根据刀具类型，使用软键"测量刀具"和其它软键进行测量。
- 按下"测量刀具"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。
- 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)转换到自动方式(信号"自动方式"DB1800.DBX0000.0)。PLC 将此信号传输给 NCK(DB3000.DBX0000.0)。

NCK 将使能自动方式(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 方式下开始测量"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。

- 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)。
- 使用软键"下一步"，选择下一个移动方向/进给轴退回到测量头。以下的步骤是类似的一直到所有的方向/进给轴已进给。

测量头测量或校准结束时，使用软键“回退”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“回退”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 工具箱中的 PLC 库中包含了用户示例。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。

10.6 数据表

10.6.1 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

10.6.2 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	自动方式(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

急停

11.1 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)。SINUMERIK 808D 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)。

急停按钮

蘑菇形按键(具有一副常开触点和一副常闭触点的紧急停止开关), 即急停开关, 安装在西门子 808D 机床控制面板(MCP)上。

11.2 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 NC 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。

11.3 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(参见图 1-1)。

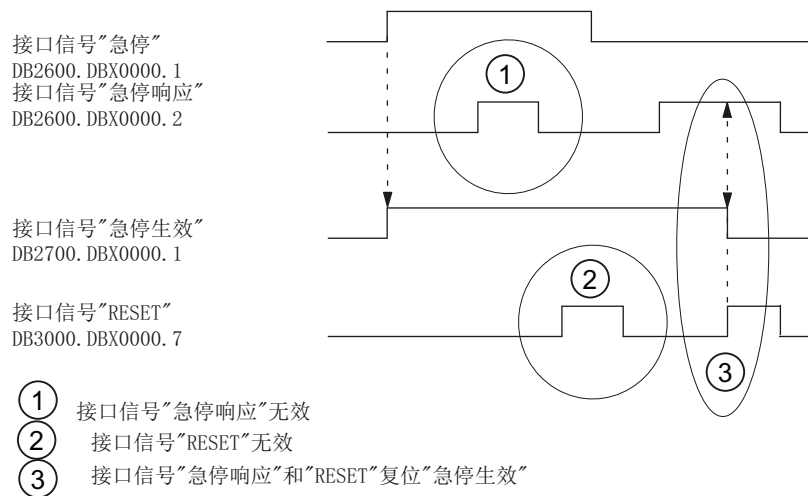


图 11-1 复位急停

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)。

11.4 数据表

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。

11.4 数据表

11.4.1 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

11.4.2 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
DB2700.DBX0000	.1	急停生效
运行方式—信号范围		
DB3000.DBX0000	.7	复位

回参考点

12.1 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 (页 173)”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(回参考点方式)来设置。

输出凸轮

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 808D 上。

文献：

电气安装手册

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：

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。

12.2 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

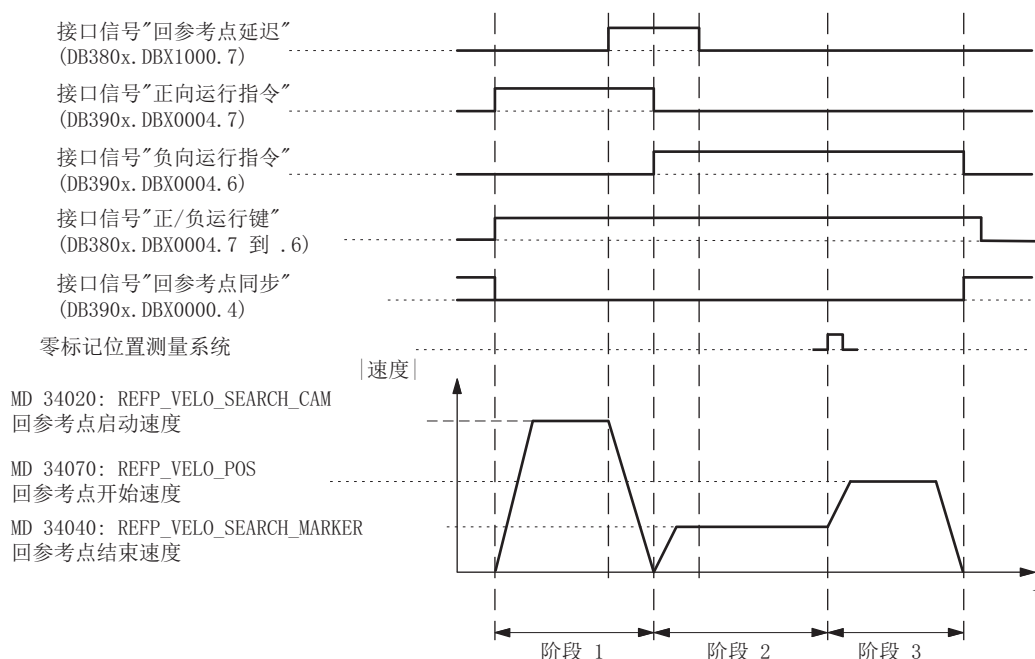


图 12-1 使用增量测量系统回参考点时序(示例)

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)	

回参考点方式	同步脉冲 (零脉冲, 接近开关 BERO)	运行顺序
不带参考减速挡块 (MD34000 REFP_CAM_IS_ACTIVE = 0)	参考点在同步脉冲之后	
<p>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)</p>		

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。 以下几点对于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 = MD34100 REFP_SET_POS = 0。
- 轴回参考点
- 在 JOG 方式，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

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。

12.3 有间距编码参考标记时回参考点

12.3.1 概述

间距编码参考标记

带间距编码参考标记的测量系统由两个并行的标度信号构成：

- 增量光栅
- 参考标记信号

应定义相邻每两个参考标记之间的间距。当运行超过两个相邻参考标记后，可由此确定机床轴的绝对位置。例如：假设参考标记间距大约为 10 mm，则机床轴回参考点时仅需大约 20 mm 的运行路径即可。

可从任意的轴位置出发向正方向或者负方向返回参考点（例外：运行范围终点）。

12.3.2 基础参数设置

线性测量系统

对于线性测量系统的参数化需设置下列数值：

- 机床零点和长度测量系统第一个参考标记位置之间的绝对偏移量：
MD34090 REFP_MOVE_DIST_CORR（参考点偏移/绝对偏移）
参见：确定绝对偏移量
- 长度测量系统相对于机床坐标系的方向设置（相同或相反）：
MD34320 ENC_INVERS（长度测量系统与机床系统方向相反）

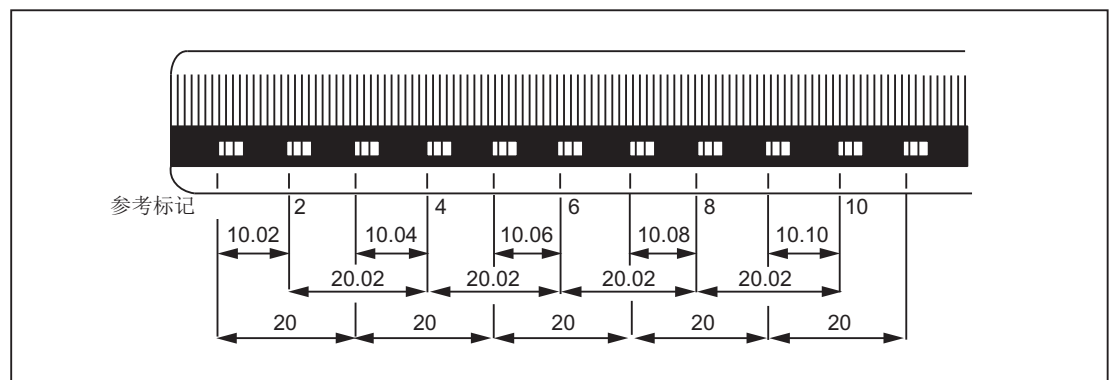


图 12-2 带间距编码参考标记的 DIADUR 精密线纹尺
(长度单位 mm，分度距离 20 mm)

旋转测量系统

旋转测量系统的规则与线性测量系统相同（参见上文）。

确定绝对偏移量

确定机床零点和机床轴第一个参考点位置之间的绝对偏移量时，建议如下操作：

1. 将绝对偏移量数值设置为零：

```
MD34090 REFP_MOVE_DIST_CORR = 0
```

2. 执行返回参考点操作。

说明

返回参考点运行应该在机床的某个位置执行，可在此位置使用如激活干扰测试仪等工具轻松测得机床轴相对于机床零点的准确位置。

3. 通过操作面板获得机床轴的实时位置。
4. 测量当前机床轴相对于机床零点的位置。
5. 计算出绝对偏移量并输入 MD34090 中。

绝对偏移量的计算与测量系统相对于机床坐标系的方向（相同或相反）有关：

测量系统的方向	绝对偏移量
相同	所测得的位置 + 显示的实时位置
相反	所测得的位置 - 显示的实时位置



警告

得到绝对偏移量并填入 MD34090 后，必须重新启动机床轴的回参考点运行。

返回参考点模式

在有间距编码参考标记时，可选择两种模式回到参考点：

- 分析计算相邻**两个**参考标记：

```
MD34200 ENC_REFP_MODE (回参考点模式) = 3
```

优点：

- 运行路径较短

- 分析计算相邻**四个**参考标记：

```
MD34200 ENC_REFP_MODE = 8
```

优点:

- 可通过 NC 进行合理性检查
- 提高回参考点结果的安全性

12.3.3 时序过程

时序过程

有间距编码参考标记时，返回参考点的时序过程分为 2 个阶段：

- 阶段 1：同步驶过参考标记
- 阶段 2：运行至固定目标点

DB380x.DBX1000.7 (回参考点运行延迟)

DB390x.DBX0004.6 (负向运行键)

DB380x.DBX0004.7 和 .6 (正负运行键)

DB390x.DBX0000.4 (回参考点/同步 1)

长度测量系统参考标记

速度

MD34070 REFP_VELO_POS
(回参考点开始速度)

MD34040 REFP_VELO_SEARCH_MARKER
(回参考点结束速度)

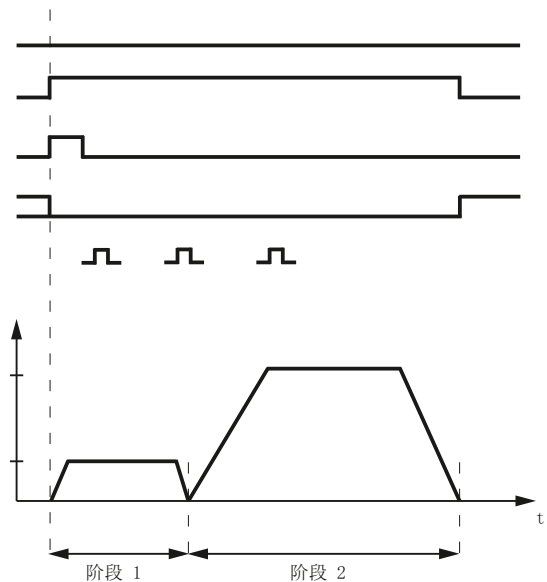


图 12-3 间距编码参考标记

12.3.4 阶段 1：同步驶过参考标记

阶段 1：启动

关于启动返回参考点运行的相关操作请参见“轴专有返回参考点 (页 157)”和“通道专有返回参考点 (页 157)”。

参考凸轮

在有间距编码参考标记的测量系统中，返回参考点本来无需任何参考凸轮。出于功能性考虑，在通道专有返回参考点运行和使用零件程序返回参考点运行中（G74）须在机床轴的运行范围末端安装参考凸轮。

阶段 1：过程

未触碰到参考凸轮的过程

启动返回参考点运行后，机床轴即加速到已编程的参考点停止速度：

MD34040 REFP_VELO_SEARCH_MARKER（参考点停止速度）

运行完已编程的参考标记个数后，机床轴再次停止，其实时数值系统同步至 NC 获得的绝对位置。

在参考凸轮上启动的过程

若启动返回参考点运行时机床轴位于参考凸轮上，则其将加速至已编程的参考点停止速度，运行方向与已编程的参考点运行方向相反：

MD34040 REFP_VELO_SEARCH_MARKER（参考点停止速度）

MD34010 CAM_DIR_IS_MINUS（沿负方向返回参考点）

如此可保证机床轴在未行驶完已编程的参考标记个数前不会撞上运行范围限制点。

运行完已编程的参考标记个数后，机床轴再次停止，其实时数值系统同步至 NC 获得的绝对位置。

返回参考点途中触碰到参考凸轮的过程

启动返回参考点运行后，机床轴即加速到已编程的参考点停止速度：

MD34040 REFP_VELO_SEARCH_MARKER（参考点停止速度）

机床轴在运行完已编程的参考标记个数前即触碰到了参考凸轮。此种情况下，机床轴会反向运行并沿相反的方向重新启动参考标记寻找过程。

运行完已编程的参考标记个数后，机床轴再次停止，其实时数值系统同步至 NC 获得的绝对位置。

参考标记间距合理性检查

若 NC 在返回参考点运行过程中探测到两个相邻参考标记间的间距大于已编程参考标记间距的两倍，则会出现错误：

MD34300 ENC_REFP_MARKER_DIST（参考标记距离）

此种情况下，机床轴会以已编程的参考点停止速度 (**MD34040**) 的一半沿反方向运行并重新启动参考标记寻找过程。

若又再次发现参考标记间距有误，则机床轴会停止并中断返回参考点运行过程（报警 20003 “测量系统”故障）。

中断标准

若在已编程的路段中未发现已编程的参考标记个数，机床轴会停止并中断返回参考点运行过程：

MD34060 REFP_MAX_MARKER_DIST（至参考标记的最长路段）

阶段 1 的特点

第 1 阶段成功结束后，机床轴实时数值系统即完成同步。

12.3.5 阶段 2：运行至目标点

阶段 2：启动

若第 1 阶段未发生报警且成功完成，则自动启动第 2 阶段。

初始情况：

- 机床轴直接停在已编程参考标记个数的最后一个上。
- 机床轴的实时位置系统已同步。

阶段 2：过程

机床轴在结束了返回参考点运行后将在第 2 阶段中运行至已定义的目标位置（参考点）。若需要缩短参考点的运行路径，也可阻止此过程进行：

MD34330 STOP_AT_ABS_MARKER

值	含义
0	运行至目标位置
1	不运行至目标位置

运行至目标位置（通常情况）

机床轴加速至已编程的参考点驶入速度并运行至已编程的目标点（参考点）：

MD34070 REFP_VELO_SEARCH_CAM（回参考点驶入速度）

MD34100 REFP_SET_POS（参考点数值）

机床轴已到达参考点。NC 发出相应的接口信号进行识别：

DB390x.DBX0000.4 = 1（已到达参考点/同步 1）

不运行至目标位置

机床轴现在已到达参考点。NC 发出相应的接口信号进行识别：

DB390x.DBX0000.4 = 1（已到达参考点/同步 1）

阶段 2 的特点

根据机床轴是否编程有参考点凸轮的不同，阶段 2 呈现出不同的特点。

无参考点凸轮的机床轴

MD34000 REFP_CAM_IS_ACTIVE (带有参考点凸轮的轴) = 0

特性:

- 进给补偿有效。
- 进给停止 (通道专用和轴专用) 生效。
- NC-STOP 和 NC-START 均有效。

有参考点凸轮的机床轴

MD34000 REFP_CAM_IS_ACTIVE (带有参考点凸轮的轴) = 1

特性:

- 进给补偿无效。
内部按进给补偿 = 100% 运行。
若进给补偿预设设为 0%，则中断运行。
- 进给停止 (通道专用和轴专用) 生效。
- NC-STOP 和 NC-START 均无效。
- 离开参考凸轮后，若未在已编程的路段中发现已编程的参考标记的个数，机床轴则停止：

MD34060 REFP_MAX_MARKER_DIST (至参考标记的最长路段)

旋转测量系统的特点

在旋转且有间距编码的测量系统中，只能在旋转范围内确定唯一的绝对位置。由于编码器机械结构的差别，硬件方面的绝对位置溢出并不总是与回转轴的运行范围吻合。

模数回转轴中的特点

在模数回转中参考点位置位于已编程的模数范围内：

MD30330 MODULO_RANGE (模数范围的大小)

MD30340 MODULO_RANGE_START (模数范围的启动位置)

说明

也可通过轴功能“在已设计的模数范围内确定旋转间距编码器的参考点位置”将参考点位置设置在已编程的（虚拟的）模数范围内。

MD30455 MISC_FUNCTION_MASK (轴功能)，位 1 = 1

12.4 绝对编码器的边界条件

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。

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！



警告

如果数据记录来自其它机床(如串行调试机床时)，当数据载入和使能后，必须进行校正。

12.5 数据表

12.5.1 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
30330	MODULO_RANGE	模数范围的大小
30340	MODULO_RANGE_START	模数范围的启动位置
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]	回参考点位移/带清除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]	参考凸轮与参考标记间距离
34100	REFP_SET_POS[0]...[3]	参考点值
34110	REFP_CYCLE_NR	通道相关轴顺序回参考点

12.5 数据表

序号	名称	名称
34200	ENC_REFP_MODE[0]	回参考点方式
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
34330	REFP_STOP_AT_ABS_MARKER[0]	无目标点的间距编码的长度测量系统
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

12.5.2 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.DBX0000	.4	回参考点, 同步 1
DB390x.DBX0004	.6 和 .7	运行指令 -/+

主轴

13.1 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。

13.2 主轴运行方式

13.2.1 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 176)”
- 摆动运行参见“主轴摆动方式运行 (页 177)”
- 定位运行参见“主轴定位方式运行 (页 180)”
- 进给轴方式
- 无补偿夹具攻丝(刚性攻丝)参见章节“进给率 (页 201)”

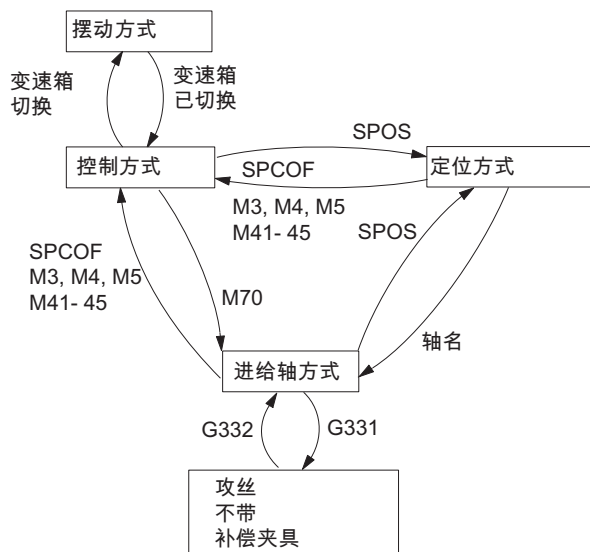


图 13-1 在主轴运行方式之间切换

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 或 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 或 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。

13.2.2 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)。

13.2.3 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“已经完成变速换挡”(DB380x.DBX2000.3)使 NCK 获悉新的变速档已运行 (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“变速换挡”(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“已经完成变速换挡”(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。

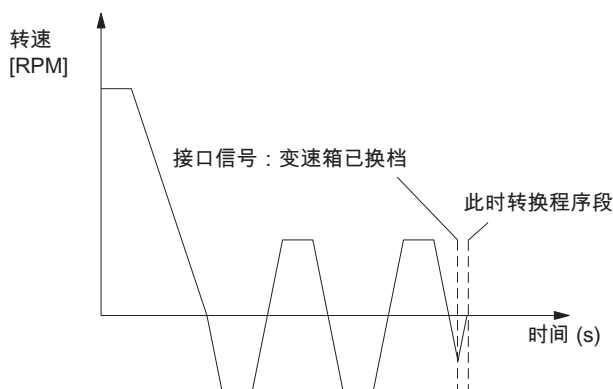


图 13-2 摆动运行方式结束之后进行程序段转换

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)。

13.2.4 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 191)”)。

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

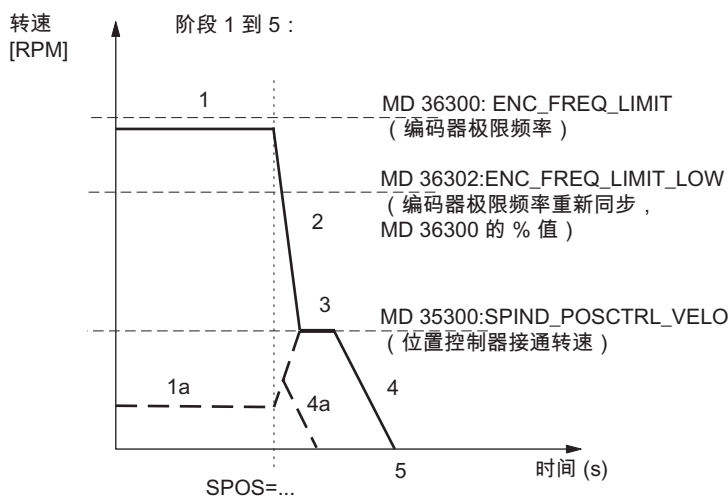


图 13-3 不同转速下的定位

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): 主轴以高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=...).

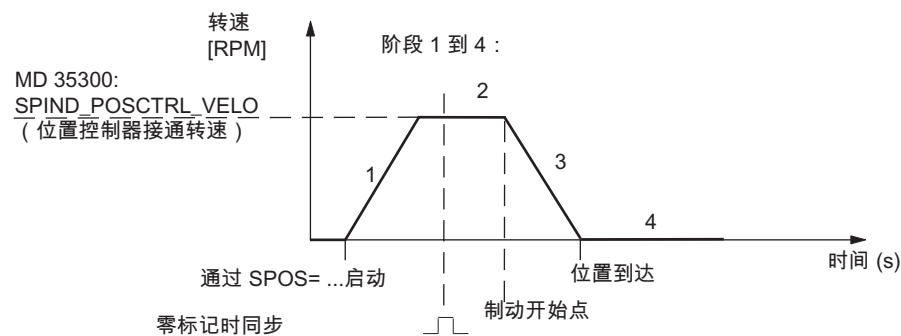


图 13-4 主轴停止且未同步时定位

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。

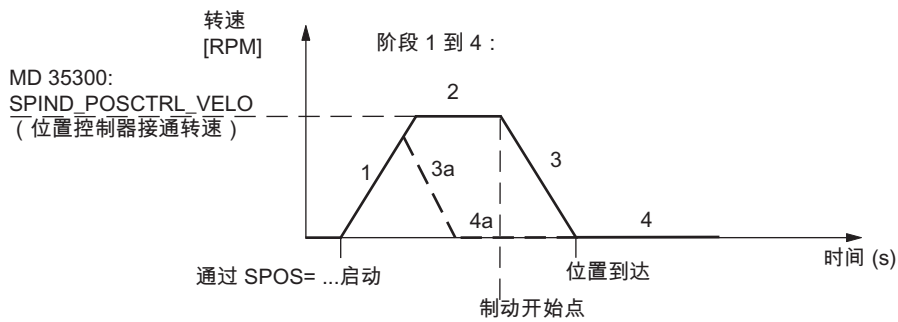


图 13-5 同步主轴停止时定位

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。

13.3 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”)。

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。

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 180)”)。
- 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”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

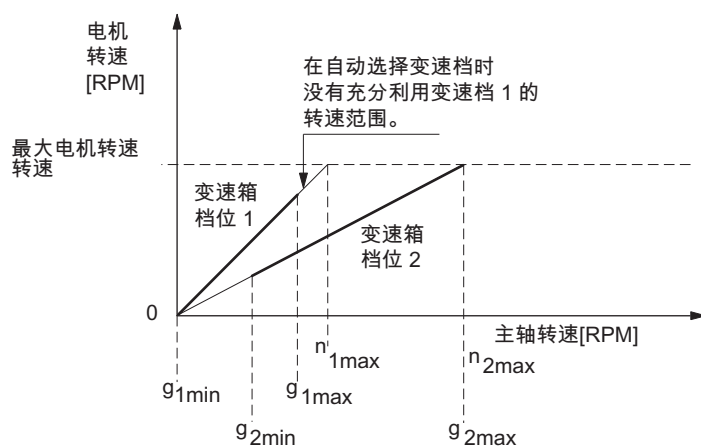
在以下情况中必须重新同步主轴的位置测量系统：

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。

13.4 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。



通过 MD 预设：
 n_{1max} ... 第 1 变速档的最大主轴转速
 g_{1min} ... 第 1 变速档的最小主轴转速
 用于自动变速换档选择
 g_{1max} ... 第 1 变速档的最大主轴转速
 用于自动变速换档选择
 n_{2max} ... 第 2 变速档的最大主轴转速
 g_{2min} ... 第 2 变速档的最小主轴转速
 用于自动变速换档选择
 g_{2max} ... 第 2 变速档的最大主轴转速
 用于自动变速换档选择

图 13-6 带有变速档选择的变速换档

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，控制器可以自动确定变速档。此时控制器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)。

控制器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)。

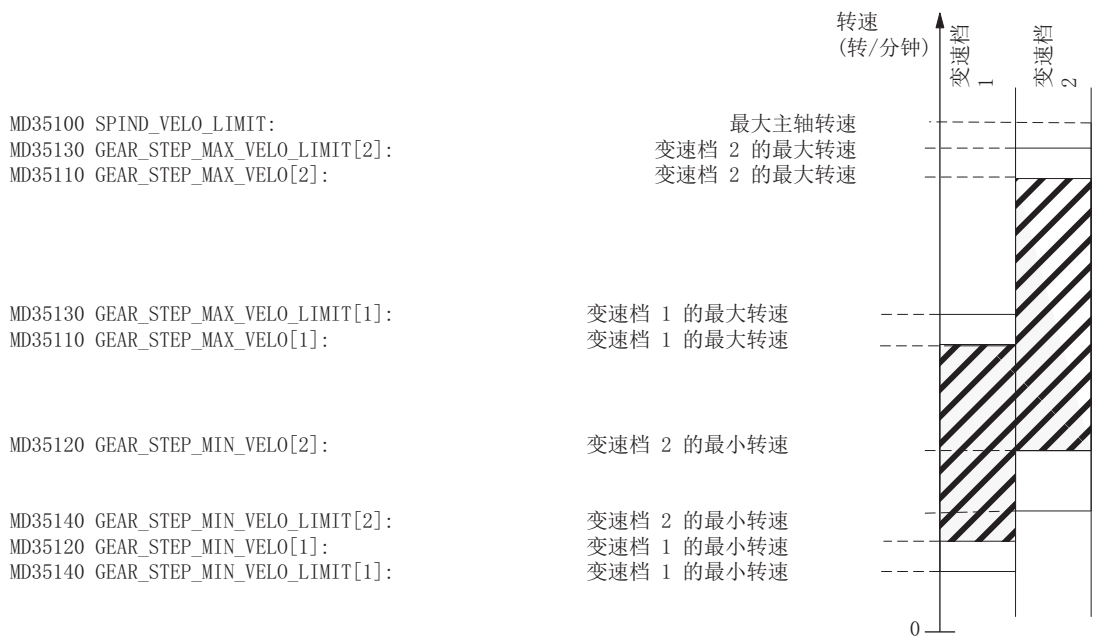


图 13-7 自动换档时转速范围说明(M40)

变速换档

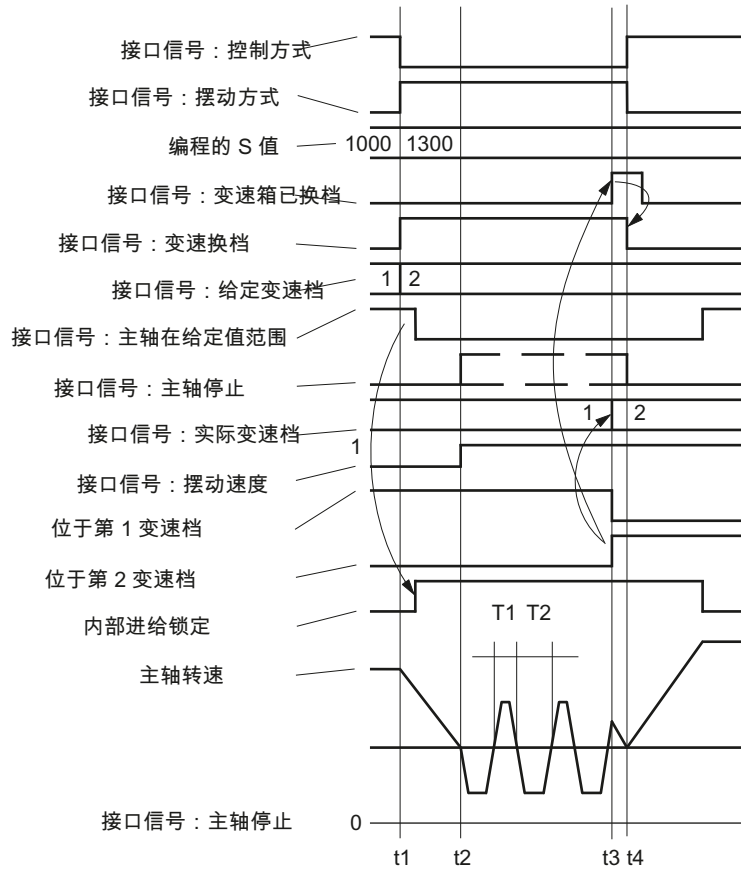
只有在主轴停止时才能切换新的变速档。

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 177)”)。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。

典型的变速换档的时序过程：



- 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）。

图 13-8 主轴停止，变速换档

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 197)”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]

13.5 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：

编程和操作手册

13.6 主轴监控

13.6.1 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。

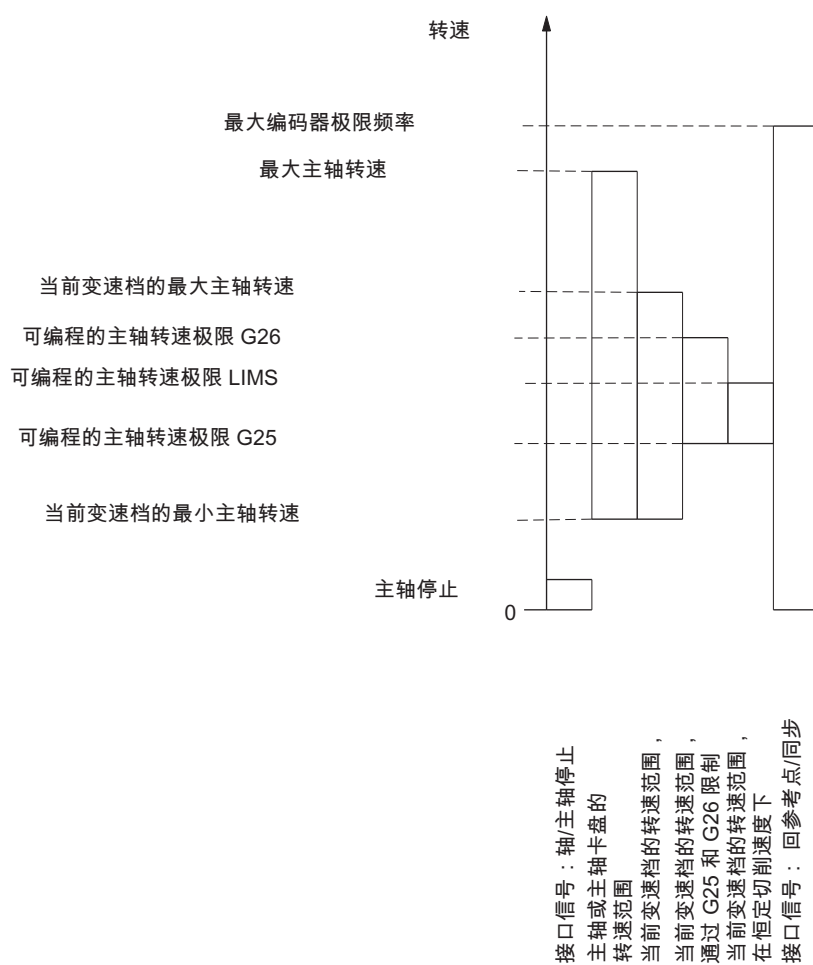


图 13-9 主轴监控范围/转速范围

13.6.2 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。

13.6.3 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”),或者主轴是否还处于加速阶段。

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。

13.6.4 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。

13.6.5 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”

13.6.6 最大编码器极限频率



警告

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。

13.6.7 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。

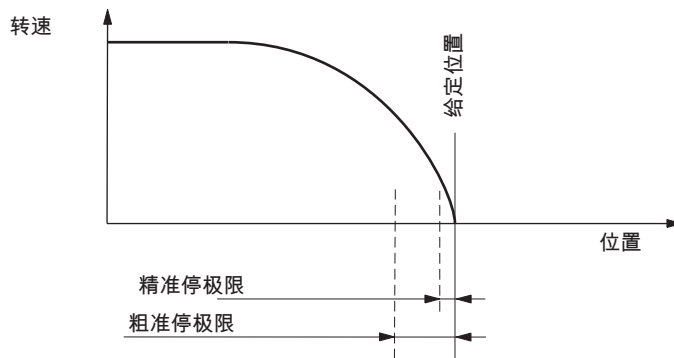


图 13-10 在定位时主轴的准停区域

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)。

13.7 模拟主轴

功能

808D 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。

13.8 数据表

13.8.1 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
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

13.8 数据表

序号	名称	名称
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。

13.8.2 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

13.8.3 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($n < n_{min}$)

序号	位	名称
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)

进给

14.1 轨迹进给率 F

14.1.1 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:

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:

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 117)”。
- 进给轴在 JOG 方式下的进给速度由机床数据/设定数据定义。

14.1.2 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：

$$Fz[\text{mm/min}] = \text{转速 } S[\text{rev/min}] * \text{螺距 } K[\text{mm/rev}]$$

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：

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”。该报警仅用于提供信息对零件程序执行没有影响。

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:

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：

$$F = \frac{|K_e^2 - K_a^2|}{2 * L_G} \text{ [mm/rev}^2\text{]}$$

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]

14.1.3 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：

$$\text{进给率 F[毫米/分钟]} = \text{转速 S[转/分钟]} \times \text{螺距[毫米/转]}$$

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：

编程和操作手册

14.1.4 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 (页 173)”)。

通常，响应较慢主轴也应和相应的进给轴匹配。

文献:

编程和操作手册

14.1.5 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，以及在供倒角或倒圆使用的说明时，会发出一个报警。

14.2 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“自动”方式下可以通过操作区“加工”->软键“程序控制”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)。

14.3 进给率控制

14.3.1 概览

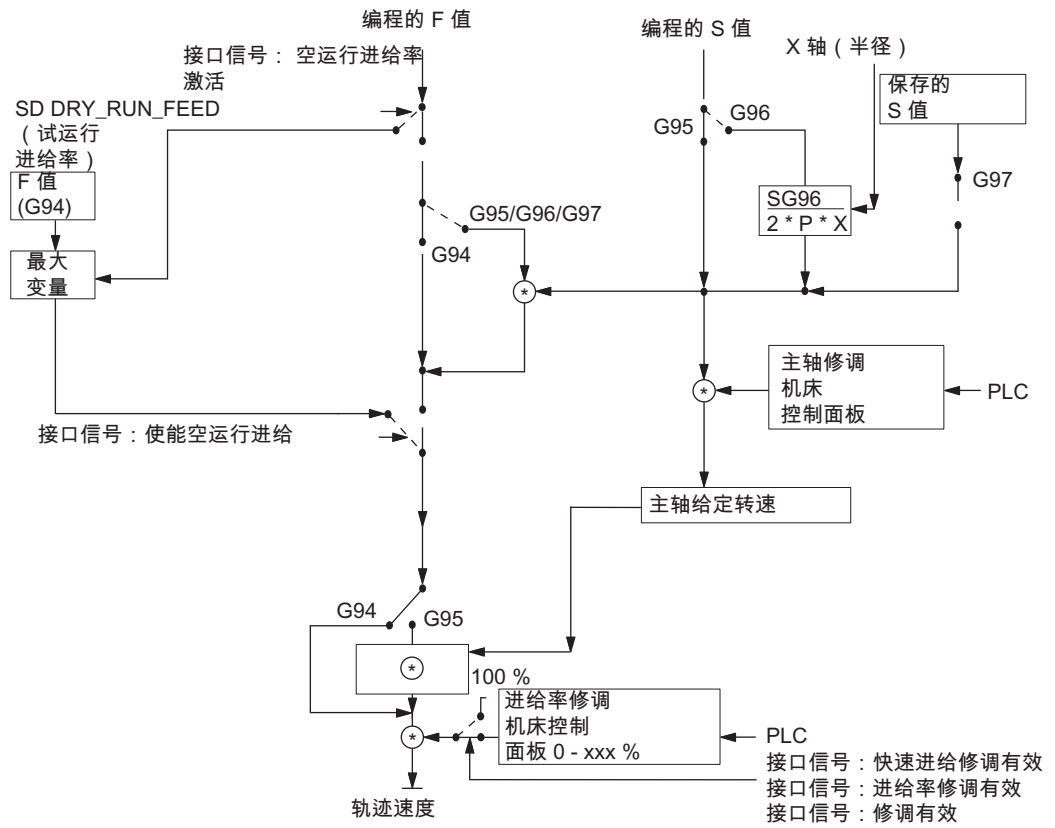


图 14-1 进给率编程和进给控制的方法

14.3.2 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 "自动" 模式下：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！

14.3.3 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%。

14.4 数据表

14.4.1 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

14.4.2 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

序号	位	名称
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
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)

刀具：刀具补偿

15.1 概述：刀具和刀具补偿

特性

SINUMERIK 808D 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

详细说明：

文献：

编程和操作手册

15.2 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。

15.3 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。

有关刀具补偿的具体说明，参见：

文献：

编程和操作手册

15.4 刀具补偿的特殊处理

对于 SINUMERIK 808D，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：

编程和操作手册

设定数据值不为零：

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

15.4 刀具补偿的特殊处理

-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:

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 内所有平面的长度补偿和 G18 内的补偿一样 (-> 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。

15.5 数据表

15.5 数据表

15.5.1 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

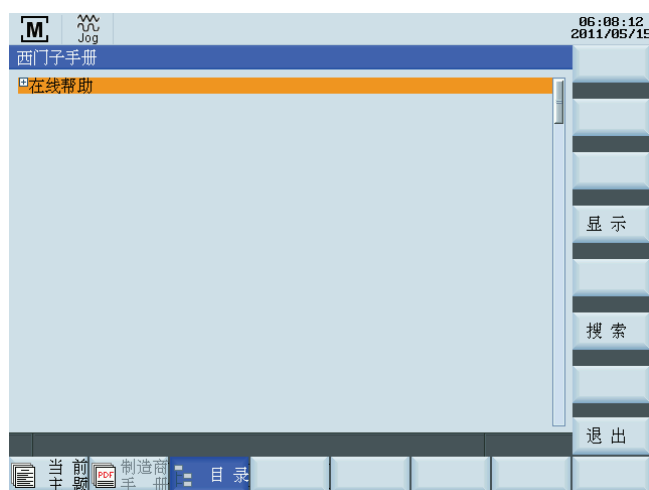
15.5.2 接口信号

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

特殊功能

16.1 调用在线帮助

提供西门子在线帮助供您参考。可按下<帮助> 键来调用该帮助：



按下 PPU 上的 <输入> 键或软键 "显示" 来显示该帮助，或按下 "退出" 来退出在线帮助屏幕。

OEM 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。例如：

<pre>#{XE"Chapter 2"} #{HL1} Chapter 2 #{HL2} How to find the Online Help? You can find the Online Help by pressing the HELP key: #{BITMAP"Images/1.bmp"} #{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}</pre>	<ul style="list-style-type: none"> > 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。 > 定义深度为 2 的标题。 > 标题下面的文本。 > 从文件夹“图像”插入位图 1 到文本。 > 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	说明
<code>#{XE "BookmarkName"}</code>	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。
<code>#{HL{depth}}</code> <code>{depth} = 1 - 5</code>	定义标题。参数{depth}定义了标题深度。
<code>#{NPAGE}</code>	开始新的帮助页面。
<code>#{BOOKMARK "BookmarkName"}</code>	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。
<code>#{XREF "BookmarkName"}{file name}{Display text of hyperlink}</code>	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。
<code>#{BITMAP "no_ref.bmp"}</code>	在文本中插入位图。
<code>#{SCOLOR {color}}</code> <code>{color} = RED, ORANGE, BLACK, BLUE, GREEN, YELLOW, WHITE</code>	将以下文本颜色变为指定颜色。

通过 USB 存储器上载 OEM 在线帮助

通过 USB 存储器上载 OEM 在线帮助的操作如下：

1. 创建自己的在线帮助文件并保存到 USB 存储器中。

文件格式可以是 .txt, .png, 以及 .bmp。由于 OEM 在线帮助支持多语言，需要首先创建不同语言的文件夹。可以在 USB 存储器中创建，例如，以下文件夹结构。

第一级：

名称	类型	日期	时间
..			
chs	DIR	11/10/11	00:19:18
eng	DIR	11/10/11	00:19:18
ptb	DIR	11/10/11	00:19:18
rus	DIR	11/10/11	00:19:18

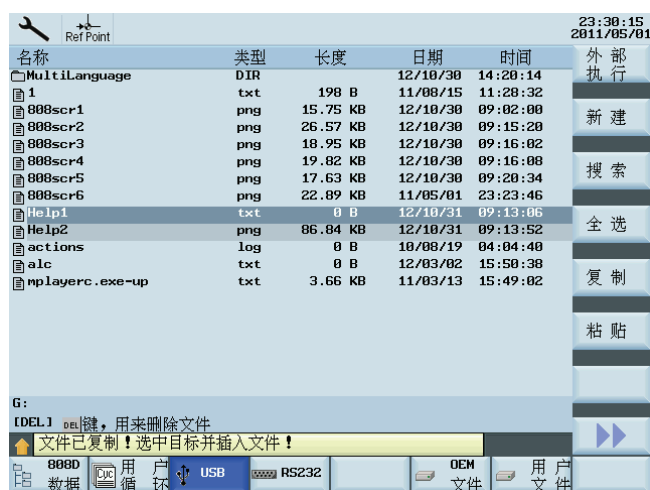
第二级：

名称	类型	日期	时间
..			
milling	DIR	11/10/11	00:19:18
turning	DIR	11/10/11	00:19:18

第三级：

名称	类型	日期	时间
..			
manual	DIR	11/10/11	00:20:33

2. 将四个第一级文件夹复制到"OEM 在线帮助 (*.txt; *.png; *.bmp)"文件夹中。参见以下步骤可找到该文件夹。
3. 将 USB 存储器插入 PPU 前面板上的 USB 接口。
4. 在"系统"操作区里，按下"系统数据"软键。
5. 按下"USB"软键。
6. 通过<选择>键多选在线帮助文件，并按下“复制”软键。

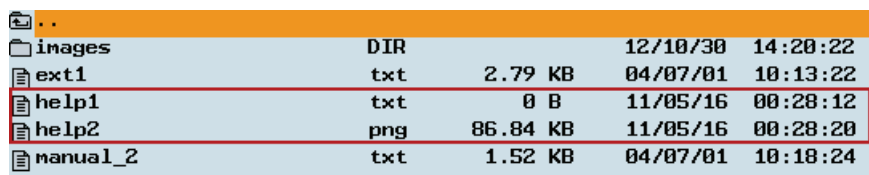


16.1 调用在线帮助

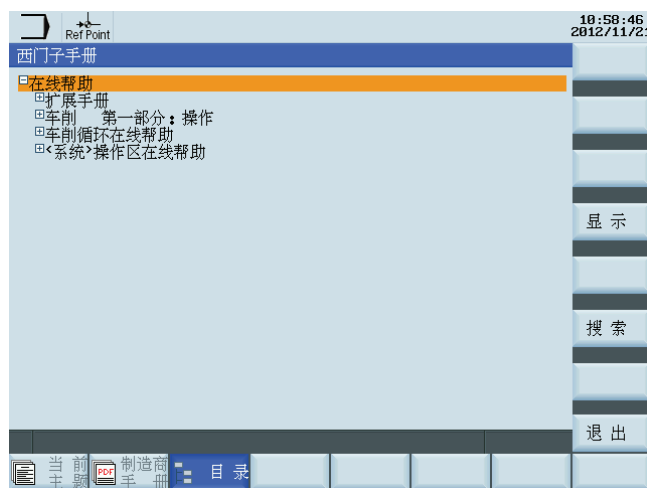
7. 按下 "808D 数据" 软键，并按下 <输入> 进入 "HMI 数据" 文件夹。然后使用上下光标键选择 "OEM 在线帮助 (*.txt; *.png; *.bmp)" 文件夹。



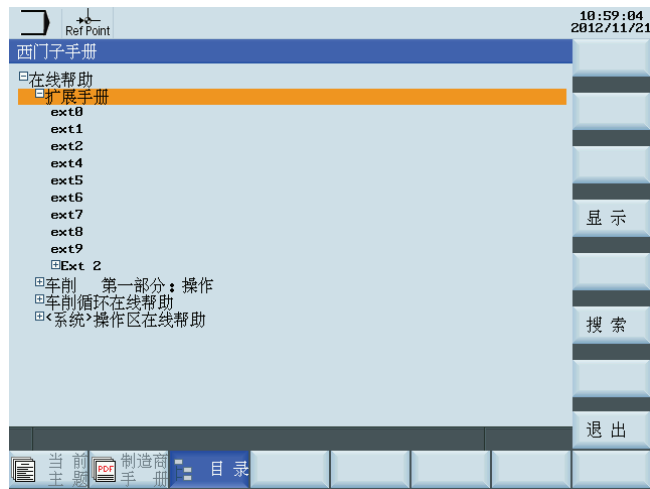
8. 按下 <输入> 进入 "OEM 在线帮助 (*.txt; *.png; *.bmp)" 文件夹，然后进入 "manual" 文件夹，即，上述三级文件夹。按下 "粘贴" 软键可将所复制的文件粘贴到该文件夹下。



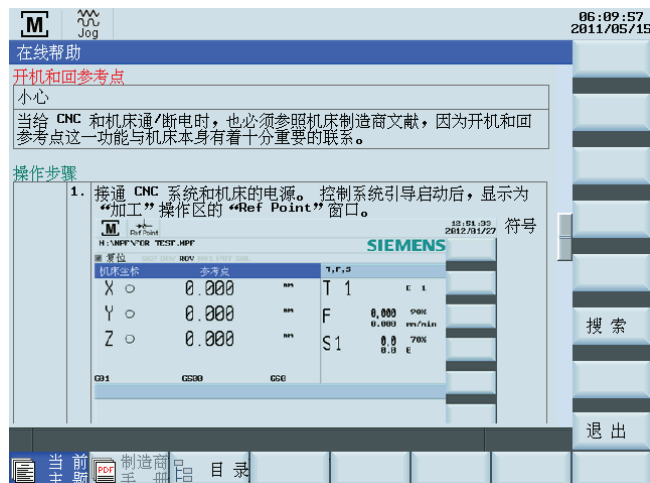
9. 按下<帮助>键并按下向右箭头键。然后可以查看如下所示的您自己的在线帮助。



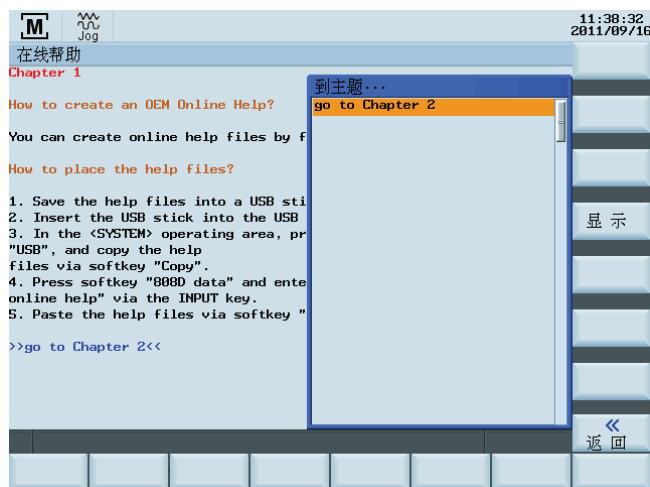
可通过按下向右箭头键来查看帮助内容列表。



10. 直接按下 <输入> 键或者软键 "显示" 来显示您的在线帮助，或者选择某一帮助内容，然后通过 <输入> 或者软键 "显示" 来查看。



11. 如已在在线帮助文本中创建了超链接，则使用软键“到主题”，接着按下软键“显示”来转到所链接的目标。



12. 按下“退出”来退出在线帮助或者按下软键“目录”来返回到在线帮助主菜单。

通过 USB 存储器上载 OEM 手册

通过 USB 存储器上载 OEM 手册的操作如下：

1. 创建自己的 OEM 手册并保存到 USB 存储器中。

文件格式必须是 **oemmanual.pdf**。由于 OEM 在线帮助支持多语言，需要首先创建不同语言的文件夹。可以在 USB 存储器中创建，例如，以下文件夹结构。

第一级：

..				
chs	DIR		11/10/11	00:19:18
eng	DIR		11/10/11	00:19:18
ptb	DIR		11/10/11	00:19:18
rus	DIR		11/10/11	00:19:18

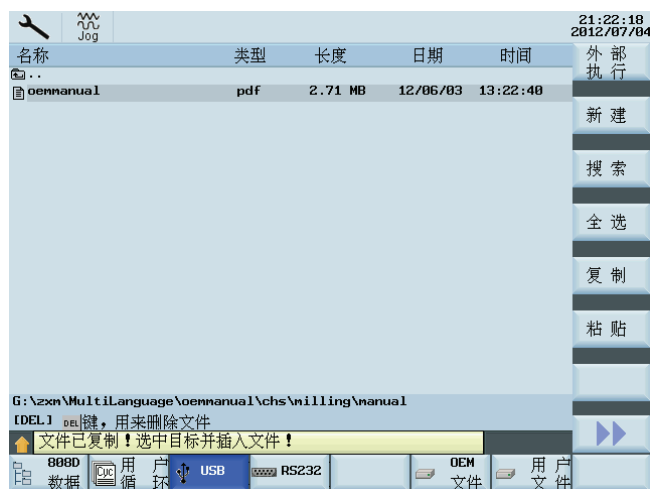
第二级：

..				
milling	DIR		11/10/11	00:19:18
turning	DIR		11/10/11	00:19:18

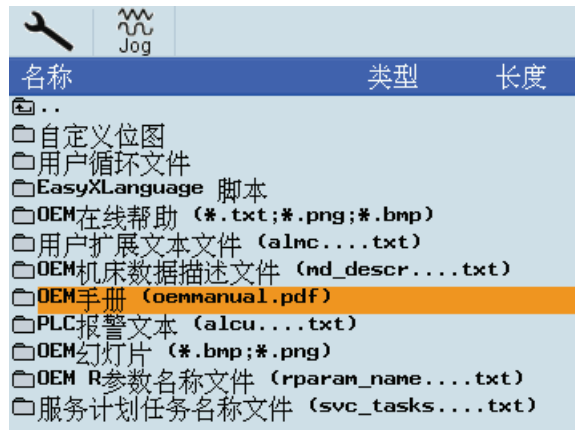
第三级：

..				
manual	DIR		11/10/11	00:20:33

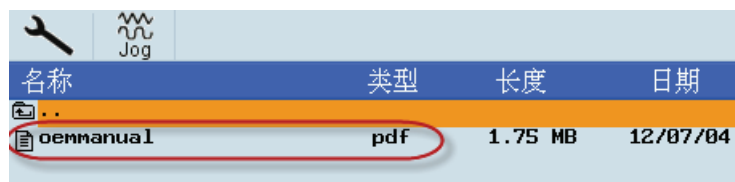
2. 将四个第一级文件夹复制到“OEM 手册 (oemmanual.pdf)”文件夹中。参见以下步骤可找到该文件夹。
3. 将 USB 存储器插入 PPU 前面板上的 USB 接口。
4. 在“系统”操作区下，按“系统数据”软键。
5. 按下“USB”软键。
6. 通过<选择>键多选 OEM 手册，并按下“复制”软键。



7. 按下 "808D 数据" 软键，并按下 <输入> 进入 "HMI 数据" 文件夹。然后使用上下箭头键选择 "OEM 手册 (oemmanual.pdf)" 文件夹。



8. 按下 <输入> 进入 "OEM 手册 (oemmanual.pdf)" 文件夹，然后进入 "manual" 文件夹，即，上述第三级文件夹。按下 "粘贴" 软键可将所复制的文件粘贴到该文件夹下。



9. 按下 <帮助> 并按下 "OEM 手册" 软键。然后即可查看您自己的 OEM 手册。

10. 按下 "退出" 键退出 OEM 手册。

注意

系统性能不良

不要上载一个或多个尺寸过大的 OEM 文件；否则，系统性能会降低。

16.2 调用具备辅助功能的标准循环

对于 SINUMERIK 808D 数控系统，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	说明
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，请参见下列：

```

%_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	说明
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
\$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。

16.3 显示功能

显示零件计时器

提供零件计时器用于 SINUMERIK 808D 以计算下列时间段：

时间	相应系统变量	说明
总运行时间	\$AC_OPERATING_TIME	自动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。

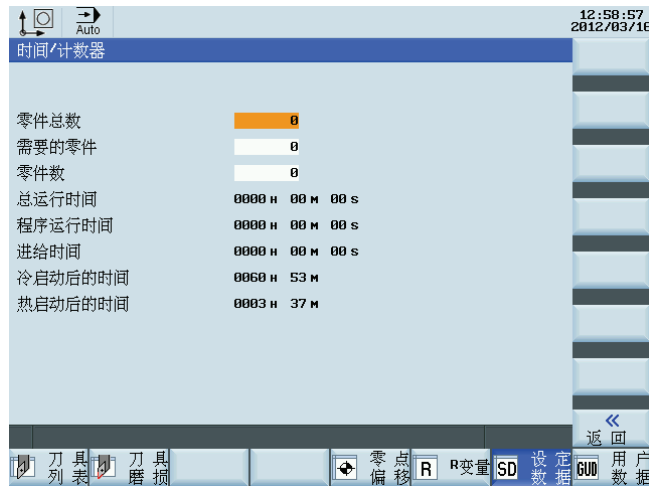
¹⁾ 剩余时间没有相应系统变量，且仅可在零件程序的成功运行周期之后计算。

²⁾ 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。

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：

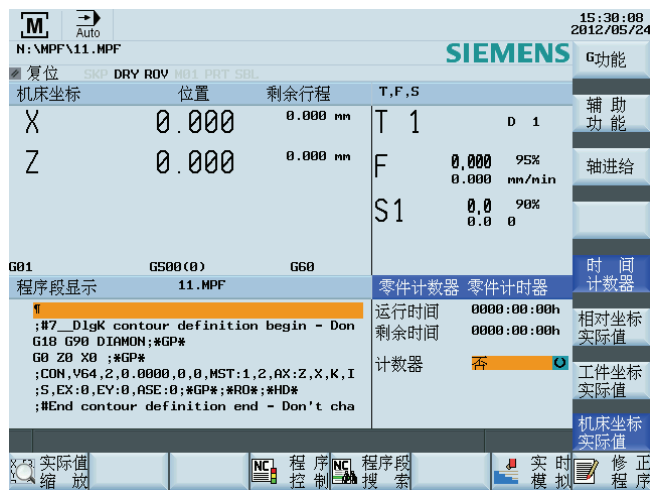
编号	名称	值	说明
2786 0	PROCESSTIMER_MODE	实际值	计算下列程序运行时间的激活操作： <ul style="list-style-type: none"> • 总运行时间 • 程序运行时间 • 进给时间

在“偏置”操作区，按下水平软键“设定数据” > 垂直软键“时间计数器”，打开“时间/计数器”窗口：



在“加工”操作区的“自动”模式下，时间计数器（垂直软键“时间计数器”）也可计算下列时间：

- 程序运行时间
- 剩余时间



显示零件计数器

提供零件计数器用于 SINUMERIK 808D 以计算下列信息：

时间	相应系统变量	说明
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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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： <ul style="list-style-type: none"> • 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。

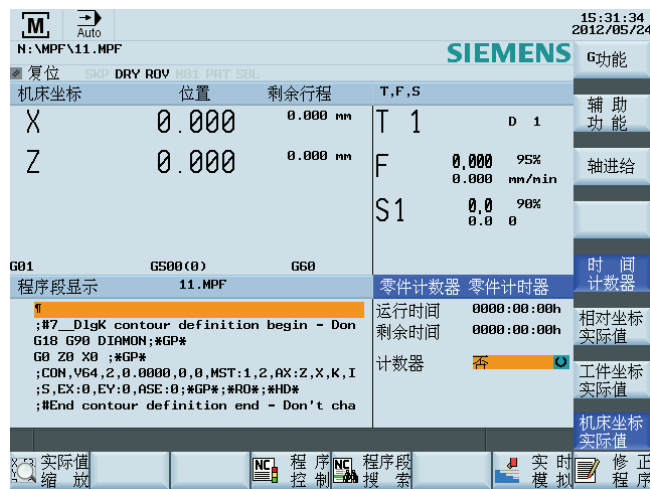
表格 16-1 相关参数

编号	名称	值	说明
2788 0	PART_COUNTER	实际值	配置并激活零件计数器
2788 2	PART_COUNTER_MCOD E	实际值	定义用于计数动作的 M 代码：0 至 99

在“偏置”操作区，可显示零件计数器（水平软键“设定数据” > 垂直软键“时间计数器”）来记录下列信息：



在“加工”操作区中的自动模式下，亦可显示零件计数器（垂直软键“时间计数器” > <选择> 键）来计算下列信息：



说明

所有已输入的数目必须通过<输入> 键进行确认。

16.4 Prog_Event 功能

借助 Prog_Event 功能，名为“PROG_EVENT.SPF”的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。必须将 PROG_EVENT.SPF 文件保持在循环目录（N:\CMA）下。

表格 16-2 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\PROG_EVENT.SPF 的触发模式： <ul style="list-style-type: none"> • 位 0: 在 NC 调试时激活 PROG_EVENT.SPF • 位 1: 在 NC 程序末尾激活 PROG_EVENT.SPF • 位 2: 通过 RESET 键来激活 PROG_EVENT.SPF • 位 3: 在启动 NC 后激活 PROG_EVENT.SPF
20109	PROG_EVENT_MASK_PROPERTIES	1H	-

16.5 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
 <p>1 +24V 2 NCRDY_K1 3 NCRDY_K2 4 DI 1 5 DI 2 6 DI 3 7 DO 1 8 CW 9 CCW 10 M</p> <p>X21 FAST I/O</p>	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。

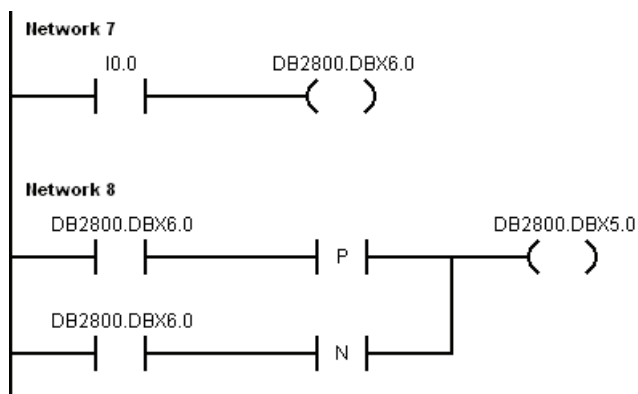
在零件程序中，可以通过相应系统变量从地址 **DB2900.DBX0.0** 读取每一位的数值。

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
```

16.6 创建用户循环

SINUMERIK 808D 整合了标准西门子循环。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
- 用户扩展文本文件
- 用户循环的告警文件
- 用户循环的软键索引文件
- 用户循环的参数文件
- 用户循环的位图文件

16.6.1 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_XXX.txt

"XXX"表示语言名称，例如，eng。

文本定义规则

定义文本时，必须遵循以下规则：

<Identifier> "<Text>" // <# chars & lines>

- **<Identifier>**: 在此处定义带编号的标识。编号范围为 83000 到 84999。
- **<Text>**: 在此处定义实际文本。
- **<# chars & lines>**: 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。

举例

83000 "User%nCycles" // 2*9 ⇒ 两行。每行为九个字符长度。

83001 "CYCLE10" // 9 ⇒ 每行为九个字符长度

16.6.2 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)

表格 16-3 参数定义

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。
Z	1 至 8	第二列的第一到第八个垂直软键。
\\${identifier}\	-	在循环文本文件中定义。
bitmap(cycle)	-	用于用户循环的位图。位图名称之后必须接着用户循环的名称。

举例

```
S5.0.0\83000\ > 在第 5 个水平键定义软键 (标识: 83000)。  
S5.1.0\83001\CN1(CYCLE100) > 当按下第 5 个水平键时,在第一层界面的第一个垂直键定义软键  
M17 (标识: 83001)。
```

16.6.3 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。

行	参数描述	条目
1	开始变量声明	(
2	变量类型	R - REAL I - INTEGER C - CHAR S - STRING
3	分隔符	/
4	<ul style="list-style-type: none"> • 最小值 + 长度 + 最大值 • * + 不同的选择值 	<ul style="list-style-type: none"> • 最小值 + 长度 + 最大值 • * + 不同字符（使用空格分开不同的字符。） <p>注意您还可以为字符定义不同的图片。</p>
5	分隔符	/
6	缺省值	在未进行输入时，该值会被传输到循环中。
7	分隔符	/
8	帮助信息	\$ + 在循环文本文件中定义的标识
9	变量声明结束)
10	开始描述	[
11	缩写	文本以（在循环文本文件中定义的）参数窗口形式显示。
12	分隔符	/

16.6 创建用户循环

行	参数描述	条目
13	窗口中的文本	输入窗口前的文本：最大长度为 5 个字符。
14	描述结束]
15	行定义的图片	/B 图片名.bmp

说明

分隔符，在开始和结束识别时必须使用。

4 行、6 行和 15 行可以空缺。

如果在\$标识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//83002)[83003/DIA]
(R/0 99999.999//83004)[83005/DIAF]
(R/-9999.999 99999.999//83004)[83004/STAP]
(R/-9999.999 99999.999//83025)[83005/ENDP]
(R/0 99999.999//83026)[83006/MID]
(R/0 99999.999//83027)[83007/UX]
(I/*0 1 2/0/83028)[83008/MACH]/B CN1
(R/1 99999.999/1/83029)[83009/VRT]
```


16.6.4 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。循环号不可以超过四位数。

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。

16.6 创建用户循环

```
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
```

16.6.5 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_xxx.txt

"xxx"表示语言名称，例如，eng。

文本定义规则

定义文本时，必须遵循以下规则：

<AlarmNumber> "<Text>" // <# chars & lines>

- **<AlarmNumber>**: 在此处定义报警号。编号范围为 65000 到 79999。
- **<Text>**: 在此处定义实际报警文本。
- **<# chars & lines>**: 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。

举例

65000 “当前刀具位置不正确” // 34 ⇒ 每行为三十四个字符长度

65001 “DIAF 大于 DIA” // 23 ⇒ 每行为二十三个字符长度

16.6.6 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**以小写/大写的"**C**"开头，且包括后缀名在内**不能**超过 **32** 个字符（例如 CN1.bmp）。



说明

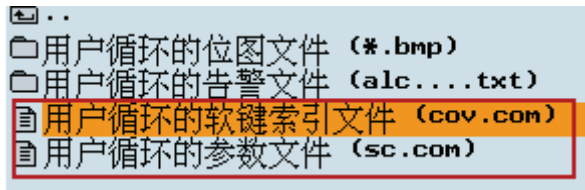
如果 16 色无法满足显示，也可以使用 24 色位图。

16.6.7 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 SINUMERIK 808D 数控系统。

导入 cov.com 文件和 sc.com 文件

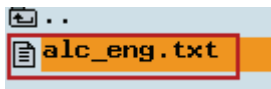
1. 将所需文件保存在 USB 闪存盘上。
2. 将 USB 闪存盘插入 PPU 前面板上的 USB 接口。
3. 按下  +  切换至“系统”操作区。
4. 按下软键"USB"并通过<选择>多选 cov.com 文件和 sc.com 文件，然后通过软键"Copy"复制这两个文件。
5. 按下软键“808D 数据”，然后进入“HMI 数据”>“用户循环文件”文件夹。



通过软键“粘贴”替换两个空文件。

导入用户循环的告警文件

1. 按下软键 "USB" 并通过<选择>键选择用户循环的告警文件（例如，alc_eng.txt）。按下 "复制"。
2. 按下"808D 数据"软键。按下 <输入> 进入 "HMI 数据"> "用户循环文件"> "用户循环的告警文件 (alc....txt)" 文件夹并按下 "粘贴"。



导入位图文件

1. 按下软键 "USB" 并通过<选择>键选择位图（例如，cn1.bmp）。按下“复制”。
2. 按下软键“808D 数据”。按下 <输入> 进入 "HMI 数据" > "用户循环文件" > "用户循环的位图文件 (*.bmp)" 文件夹并按下 "粘贴"。

名称	类型	长度	日期	时间
..				
ico1024	DIR		12/02/29	01:43:06
ico1280	DIR		12/02/29	01:43:06
ico1600	DIR		12/02/29	01:43:06
ico640	DIR		12/02/29	01:43:06
ico800	DIR		12/02/29	01:43:06
cn1	bmp	253.66 KB	12/03/05	10:57:18

导入用户循环文件

1. 按下软键"USB"并选择用户循环文件（例如，CYCLE100）。按下“复制”软键。
2. 按下“用户循环”软键，而后按下“粘贴”。

名称	类型	长度	日期	时间
CYCLE100	SPF	944 B	12/03/05	10:55:00

导入用户扩展文本文件

1. 按下软键“系统数据” > “USB”。选择用户扩展文本文件(例如，almc_eng.txt)并通过软键"复制"来复制该文件。
2. 按下软键“808D 数据”。按下<输入>进入"HMI 数据" > "用户扩展文本文件 (almc....txt)" 文件夹，然后按下"粘贴"。

..
almc_eng.txt

说明

按下“粘贴”导入文件 cov.com, sc.com, alc_xxx.txt 和 almc_xxx.txt 后，会出现系统消息提醒您按下"确认"重启 HMI。然后按下"确认"重启 HMI，这样新数据即可用。

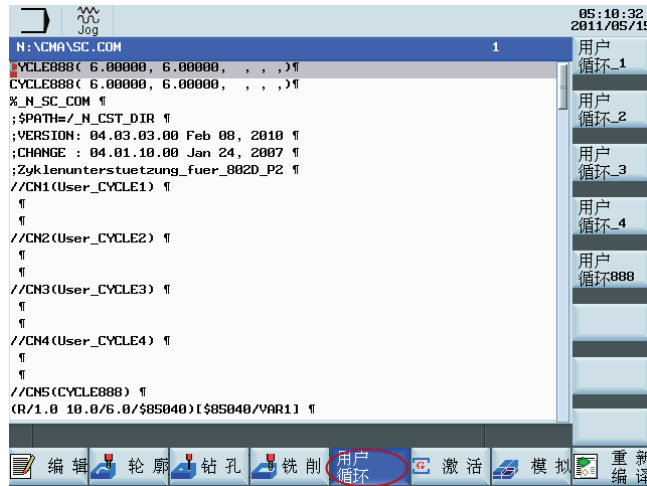
16.6 创建用户循环

16.6.8 调用所创建的用户循环

将您自己的循环必备的文件都传递至数控系统后，循环创建成功。然后即可使用"程序"操作区的循环。

如下调用所创建的循环，例如，CYCLE888。

1. 按下 PPU 上的<程序>，水平软键"用户循环" 出现。按下“用户循环”。



2. 按下垂直软键“用户 Cycle888”打开"CYCLE888"的窗口。



3. 根据需要设置参数，并按下"确认"保存所作设置。或者，按下"取消"退出循环。

16.6.9 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。

16.7 通过自定义 EasyXLanguage 脚本创建用户对话框

16.7.1 功能范围

概述

使用功能“创建用户对话框”可以保证开放性，它能够让用户设计出客户专用和应用专用的 HMI 界面。

SINUMERIK 808D 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 上的操作区“用户自定义”中显示机床专用菜单和对话框窗口。

使用

已定义的 XML 指令可以实现下列特性：

1. 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
2. 通过以下方法调用对话框：
 - 按下相应软键
3. 动态重组对话框
 - 修改、删除软键
 - 定义并设计变量栏
 - 显示、更换、删除显示文本（和语言相关或无关）
 - 显示、更换、删除图形

16.7 通过自定义 *EasyXLanguage* 脚本创建用户对话框

4. 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
5. 对话框间的数据交换
6. 变量
 - 读取（NC 变量、PLC 变量、用户变量）
 - 写入（NC 变量、PLC 变量、用户变量）
 - 和数学、比较或者逻辑运算符相连
7. 执行下列功能：
 - 子程序
 - 文件功能
 - PI 服务
8. 根据用户组考虑保护等级

关于脚本语言的有效单元（标签）的描述可参见章节“XML 标签 (页 252)”。

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。

16.7.2 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。在工具箱的“...\examples\easyXL”文件夹中可找到配置文件（EasyXLanguage 脚本）。

可以使用 XML 编辑器或其他文本编辑器来创建配置文件。

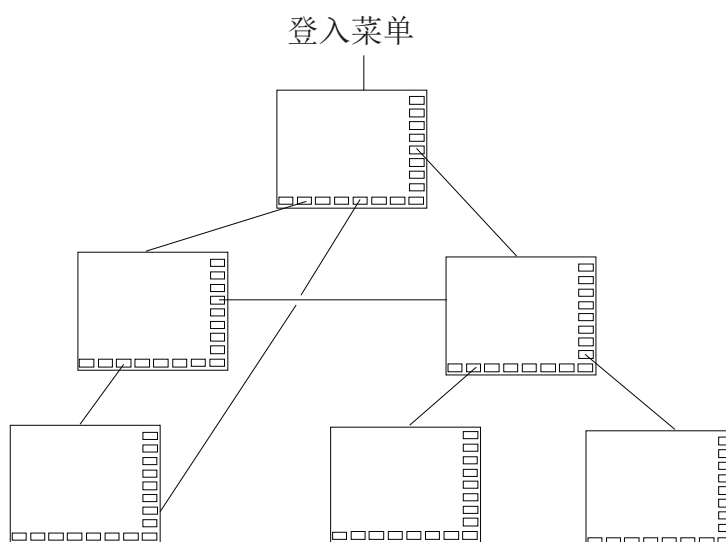
说明

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

在文件“xmldial.xml”中使用名称“main”来定义登入菜单。登入菜单是操作流程自身的输出点。

使用主菜单可以与加载自身对话框或其他软键条连接在一起。而使用这些软键条又可以执行其他动作。

16.7 通过自定义 EasyXLanguage 脚本创建用户对话框

返回到标准应用程序

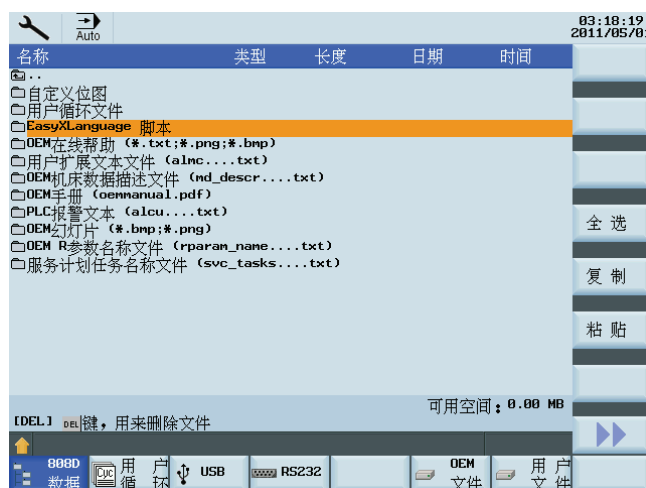
可以通过按下 PPU 上的下列键或组合键退出新建的用户界面并返回到标准应用程序。



16.7.3 配置文件 (EasyXLanguage)

载入配置文件

必须将创建的配置文件从 USB 存储器复制到“系统”操作区 > “系统数据” > “808D 数据” > “HMI 数据” 菜单 > “EasyXLanguage 脚本”文件夹。参见下屏：



用于配置的文件

为了进行用户对话框的配置需要使用数控系统中“EasyXLanguage 脚本”文件夹内的下列文件：

文件类型	文件名称	含义
脚本文件	“xmldial.xml”	该脚本文件通过 XML 标签控制已配置软键菜单的映像以及 HMI 上操作区“用户自定义”中的对话框屏幕。
文本文件	“almc.txt”	该文本文件包含有用于单个语言菜单和对话框屏幕的文本。

文件类型	文件名称	含义
位图	"*.bmp" (例如, "text.bmp") "*.png" (例如, "text.png")	带位图的文档。 控制器支持 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。

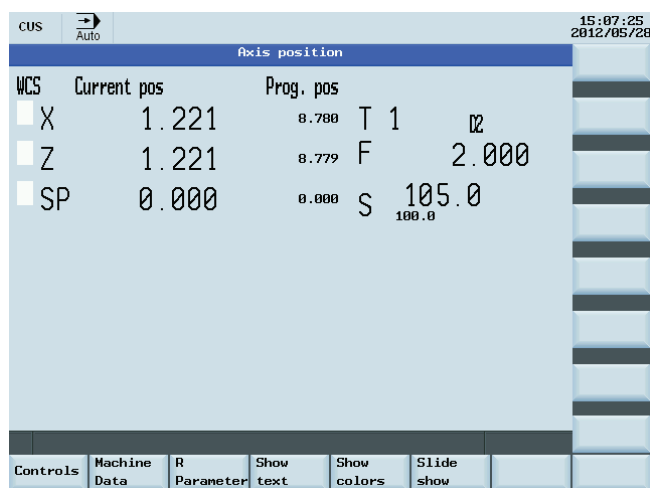
说明

一旦"xmldial.xml"脚本文件存在于"EasyXLanguage 脚本"文件夹中, 便可在"用户自定义"操作区中起始该用户对话框。

初始复制过程结束后, 必须通过"系统"操作区 > "调试" > "NC" > "正常引导启动"来复位数控系统。

HMI 上用户对话框示例

在调用操作区"自定义"时显示配置好的软键菜单。用户可以通过配置好的各个对话框窗口进行操作。



说明

如果需要同时使用配置好的且已编程的对话框, 则必须通过脚本语言调用已编程对话框。为此所需的功能在章节"预定义函数 (页 288)"中有所说明。

16.7.4 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

16.7.5 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

与语言相关的文本保存在文本文件（*almc.txt*）中。

16.7.6 XML 标签

16.7.6.1 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

语言为完成有条件指令与完成程序循环提供有下列指令：

- For loop 循环
- While loop 循环
- Do with loop 循环
- 有条件执行
- Switch 指令和 Case 指令
- 对话框窗口中的操作单元
- 软键说明
- 定义变量

指令的详细说明参见章节“指令/标签说明 (页 254)”。

16.7.6.2 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。

实例：如下对

```
<DATA_LIST action="read/write/append" id="<list name>">
```

进行编程：

```
<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_RESET	<p>该标签可以重新启动一个或多个控制器组件。</p> <p>句法： <CONTROL_RESET resetnc="TRUE" /></p> <p>属性：</p> <ul style="list-style-type: none"> • RESETNC = "TRUE" 数控系统组件重新启动
DATA	<p>该标签启用待直接写入的 NC、PLC 和 GUD 数据。</p> <p>地址构成可以查阅章节“组件地址分配 (页 267)”。</p> <p>属性：</p> <ul style="list-style-type: none"> • name 变量地址 <p>标签值：</p> <p>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，则使用替代运算符 \$，将局部变量名跟在其后。</p> <p>句法： <DATA name="<variable name>"> value </DATA></p> <p>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></p>

标签名称	含义
DATA_LIST	<p>该标签可以保存或者恢复所列示的机床数据。</p> <p>可以创建最多 20 个临时数据列表。</p> <p>属性:</p> <ul style="list-style-type: none"> • action <ul style="list-style-type: none"> <i>read</i> – 将列出变量的值存储到一个临时存储器中 <i>append</i> – 将列出变量的值添加到已经存在的列表中 <i>write</i> – 将保存的值复制到相应的机床数据中 • id <ul style="list-style-type: none"> 名称用于识别临时存储器 <p>句法:</p> <pre><DATA_LIST action="<read/write/append">" id="<list name>"> NC/PLC 地址编译 </DATA_LIST></pre> <p>示例:</p> <pre><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>" /></pre>
ELSE	满足条件时的指令（ IF （如果）、 THEN （则）、 ELSE （否则））

标签名称	含义
FORM	<p>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。</p> <p>句法: <code><FORM name="<dialog name>" color="#ff0000"></code></p> <p>属性:</p> <ul style="list-style-type: none"> • color 对话框窗口的背景颜色（颜色代码参见章节 颜色代码 (页 266)） – 了解缺省值 • name 形状的名称 • 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）
HMI_RESET	<p>该标签触发 HMI 重启。 解释在该操作之后中断。</p>

标签名称	含义
IF	<p>有条件指令 (IF, THEN, ELSE)</p> <p>THEN 和 ELSE 标签包含在 IF 标签中。</p> <p>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。</p> <p>句法:</p> <pre><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></pre> <p>示例:</p> <pre><IF> <CONDITION> "plc/mb170" != 7 </CONDITION> <THEN> <OP> "plc/mb170" = 7 </OP> ... </THEN> <ELSE> <OP> "plc/mb170" = 2 </OP> ... </ELSE> </IF></pre>
INCLUDE	<p>指令包含有 XML 说明。</p> <p>(参见该表中的 DYNAMIC_INCLUDE)</p> <p>属性:</p> <ul style="list-style-type: none"> • src 包含路径名称。 <p>句法:</p> <pre><?INCLUDE src="<Path name>" ?></pre>

标签名称	含义
LET	<p>指令以给出的名称创建一个局部变量。</p> <p>域:</p> <p>使用属性 dim (维度)，可创建一维或二维域。域索引用于定址单个域元素。</p> <p>对于二维域，先指定行索引，而后指定列索引。</p> <ul style="list-style-type: none"> • 一维域: 索引 0 到 4 • 二维域: 索引行 0 到 3 以及索引列 0 到 5 <p>属性:</p> <ul style="list-style-type: none"> • name 变量名称 • type 变量类型可以为整数型 (INT)、双精度型 (DOUBLE)、浮点数值型 (FLOAT) 或者字符串型 (STRING)。如果未给出类型指令，则系统会创建一个整数型变量。 <code><LET name = "VAR1" type = "INT" /></code> • 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] <ul style="list-style-type: none"> - 一维域: dim="<元素数目>" - 二维域: dim="<行数目的>,<列数目>" 非初始化域元素预设为“0”。

标签名称	含义
LET 续	<p>示例：</p> <p>一维域： <code><let name="array" dim="10"></let></code></p> <p>二维域： <code><let name="list_string" dim="10,3" type="string"></let></code></p> <p>预设值：</p> <p>可以对变量进行初始化赋值。 <code><LET name = "VAR1" type = "INT"> 10 </LET></code></p> <p>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。</p> <ul style="list-style-type: none"> 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。如果一行要使用换行 line feed <code><LF></code> 结尾，则可以在行的末尾添加字符 <code>"\n"</code>。 <pre><LET name = "text" type = "string"> F4000 G94\n G1 X20\n Z50\n M2\n </LET>></pre> <p>域 (Arrays) :</p> <pre><let name="list" dim="10,3"> {1,2,3}, {1,20} </let></pre> <pre><let name="list_string" dim="10,3" type="string"> {"text 10","text 11"}, {"text 20","text 21"} </let></pre> <p>赋值：</p> <p>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。</p> <p>全局可用的变量要直接在标签 DialogGui 后创建。</p> <p>在对话框中要注意：</p> <ul style="list-style-type: none"> 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
MSG	<p>操作组件显示标签中所给出的信息。</p> <p>如果使用报警编号，则对话框会显示为该编号所保存的文本。</p> <p>示例：</p> <pre><MSG text ="my message" /></pre>
MSGBOX	<p>指令会打开一个信息框，其给分支的返回值可供使用。</p> <p>句法：</p> <pre><MSGBOX text="<Message>" caption="<caption>" retvalue="<variable name>" type="<button type>" /></pre> <p>属性：</p> <ul style="list-style-type: none"> • text 文本 • caption 标题 • retvalue 其中复制有返回值的变量名称： 1 – OK 0 – CANCEL • type 应答可能： "BTN_OK" "BTN_CANCEL" "BTN_OKCANCEL" <p>如果对属性"text"或"caption"使用报警编号，则消息框会显示为该编号所保存的文本。</p> <p>示例：</p> <pre><MSGBOX text="测试消息" caption="信息" retvalue="result" type="BTN OK" /></pre>

标签名称	含义
OP	<p>该标签执行所列出的运算。</p> <p>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。</p> <p>PLC: "PLC/MB170" NC: "NC/Channel/..."</p> <p>示例:</p> <pre><LET name = "tmpVar" type="INT"> </LET> <OP> tmpVar = "plc/mb170" </OP> <OP> tmpVar = tmpVar *2 </OP> <OP> "plc/mb170" = tmpVar </OP></pre> <p>字符串处理:</p> <p>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。</p> <p>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。</p> <p>示例:</p> <pre><LET name="buffer" type="string"></LET> <OP> buffer = _T"unformatted value R0= " + "nck/Channel/Parameter/R[0]" + _T" and " + _T"\$\$85051" + _T" formatted value R1 " + _F"9.3f"nck/Channel/Parameter/R[1]" </OP></pre>
PASSWORD	<p>标签打开用于输入密码的对话框。</p> <p>输入之后可以使用所给出参考变量中的字符串。</p> <p>句法:</p> <pre><PASSWORD refVar ="<variable name>" /></pre> <p>属性:</p> <ul style="list-style-type: none"> • refVar 参考变量的名称 <p>示例:</p> <pre><PASSWORD refvar="plc/mw107" /></pre>
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	<p>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。</p> <p>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。</p> <p>句法:</p> <pre><PRINT name="变量名称" text="text %格式化"> Variable, ... </PRINT> <PRINT text="text %格式化"> Variable, ... </PRINT></pre> <p>属性:</p> <ul style="list-style-type: none"> • name 其中应当保存文本的变量名称（可选） • text 文本 <p>格式化:</p> <p>字符“%”会对作为值给定的变量进行格式化。</p> <p>%[标记][宽度][.小数点后面的位置] 类型</p> <ul style="list-style-type: none"> • 标记: 用于确定输出格式的可选字符： <ul style="list-style-type: none"> - 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。 • 小数点后面的位置: 使用浮点数时优化参数用来确定小数点后面的位置数。 • 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。 <ul style="list-style-type: none"> - d: 整数值 - f: 浮点数 - s: 字符串

标签名称	含义
PRINT 续	<p>值：</p> <p>其值应当插入到文本中的变量数量。</p> <p>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。</p> <p>示例：</p> <p>在信息行中输出文本</p> <pre><PRINT text="信息文本" /></pre> <p>使用变量格式输出文本</p> <pre><LET name="trun_dir"></LET> <PRINT text="M%d">trun_dir</PRINT></pre> <p>在字符串变量中使用变量格式输出文本</p> <pre><LET name="trun_dir"></LET> <LET name="str" type="string" ></LET> <print name="str" text="M%d ">trun_dir</print></pre>
STOP	解释在该位置中断。
SWITCH	<p>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。</p> <p>如果没有常数与表达式一致，则处理 DEFAULT 指令。</p> <p>句法：</p> <pre><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></pre>
THEN	满足条件时的操作 (IF, THEN, ELSE)

标签名称	含义
<p>FOR</p>	<p>FOR loop 循环 for (初始化; 测试; 增量) 指令</p> <p>句法: <FOR> <INIT>...</INIT> <CONDITION>...</CONDITION> <INCREMENT>...</INCREMENT> 指令 ... </FOR></p> <p>按下列方式执行 For 循环:</p> <ol style="list-style-type: none"> 1. 运用表达式初始化 (INIT) 。 2. 运用表达式测试 (CONDITION) 作为布尔型表达式。 如果值为假 (false) , 则结束 For 循环。 3. 执行后续的指令。 4. 运用表达式增量 (INCREMENT) .. 5. 继续使用 2。 <p>所有在 INIT、ONDITION 和 INCREMENT 分支中使用的变量都要在 FOR loop 循环外部创建。</p> <p>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></p>
<p>WAITING</p>	<p>标签在数控系统复位后等待组件重新启动。</p> <p>属性:</p> <ul style="list-style-type: none"> • WAITINGFORNC = "TRUE" - 系统等待数控系统重启 <p>句法: <WAITING WAITINGFORNC = "TRUE"/></p>

标签名称	含义
WHILE	<p>While 循环</p> <pre>WHILE (Test) 指令</pre> <p>句法:</p> <pre><WHILE> <CONDITION>...</CONDITION> 指令 ... </WHILE></pre> <p>While loop 循环用于按顺序多次执行指令，只要满足条件。在处理指令序列前对条件进行检查。</p> <p>示例:</p> <pre><WHILE> <CONDITION> "plc/ib9" == 0 </CONDITION> <DATA name = "PLC/qb11"> 15 </DATA> </WHILE></pre>
DO_WHILE	<p>Do-While 循环</p> <pre>DO 指令 WHILE (测试)</pre> <p>句法:</p> <pre><DO_WHILE> 指令 ... <CONDITION>...</CONDITION> </DO_WHILE></pre> <p>Do while loop 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），则重新执行代码部分。一直重复，直至条件为假（false）。</p> <p>示例:</p> <pre><DO_WHILE> <DATA name = "PLC/qb11"> 15 </DATA> <CONDITION> "plc/ib9" == 0 </CONDITION> </DO_WHILE></pre>

16.7.6.3 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"

16.7.6.4 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
<	<
>	>
&	&
"	"
'	'

16.7.6.5 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。

16.7.7 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。

16.7.7.1 PLC 地址分配

PLC 地址分配以路径部分 **plc** 开始。

表格 16-4 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

表格 16-5 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>
```

16.7.7.2 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，其结构可以查阅 SINUMERIK 808D 参数手册。

示例：

```
<LET name = "tempStatus"></LET>
<OP> tempStatus ="nck/channel/state/chanstatus" </OP>
```

16.7.7.3 机床数据和设定数据的地址分配

可以使用 \$ 符号标记设定数据，后面跟有数据名称。

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>
```

16.7 通过自定义 EasyXLanguage 脚本创建用户对话框

16.7.7.4 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>
```

16.7.8 创建用户菜单

16.7.8.1 创建软键菜单和对话框窗口

要合成用户菜单必须在 XML 说明中有名为“main”的主菜单标签。在激活操作区“用户自定义”后由系统调用该标签。在标签之中可以定义其他的菜单分支以及激活一个对话框。

```

<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	<p>该标签包含用户对话框的说明。</p> <p>属性:</p> <ul style="list-style-type: none">• color 对话框窗口的背景颜色（颜色代码参见章节“颜色代码”）• name 窗体的名称• 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） <p>对话框信息:</p> <ul style="list-style-type: none">• INIT• PAINT• TIMER• CLOSE• FOCUS_IN

标签名称	含义
FORM 续	<p>句法:</p> <pre><FORM name = "<dialog name>" color = "#ff0000"></pre> <p>示例:</p> <pre><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></pre>
INIT	<p>对话框信息</p> <p>在对话框创建之后立即处理该标签。在此创建所有的输入单元以及对话框窗口的“热连接”。</p>
FOCUS_IN	<p>对话框信息</p> <p>当系统将焦点设置到控制单元时，调用该标签。为了识别控制单元，系统将控制单元的名称复制到变量 \$focus_name 中并将 item_data 的属性值复制到变量 \$focus_item_data 中。系统自动创建变量。</p> <p>例如可以使用该信息，来输出与焦点位置有关的图像。</p> <p>示例:</p> <pre><focus_in> <PRINT text="focus on filed:%s, %d">\$focus_name, \$focus_item_data </PRINT> </focus_in></pre>
PAINT	<p>对话框信息</p> <p>在显示对话框时处理该标签。在此要给定需要在对话框中显示的所有文本与图像。</p> <p>此外，在系统识别出部分对话框需要重新显示时处理该标签。例如，通过关闭高层窗口来触发该操作。</p>

标签名称	含义
TIMER	<p>对话框信息</p> <p>周期性处理该标签。</p> <p>每一窗体均具备一个定时器，该定时器大约每隔 100 ms 触发一次定时器标签的处理。</p>
CAPTION	<p>该标签包含有对话框的标题。</p> <p>在 INIT 标签内使用该标签。</p> <p>句法: <CAPTION>Titel</CAPTION></p> <p>示例: <CAPTION>my first dialogue</CAPTION></p>
CLOSE	<p>对话框信息</p> <p>在关闭对话框之前处理该标签。</p>
CLOSE_FORM	<p>该标签关闭活动对话框。</p> <p>该指令仅当其涉及用于程序编辑器区域中的循环对话框时有用。一般而言，对话框受自动管理，无需特意关闭。</p> <p>句法: <CLOSE_FORM/></p> <p>示例: <softkey_ok> <caption>OK</caption> <CLOSE_FORM /> <navigation>main_menu</navigation> </softkey_ok></p>

标签名称	含义
CONTROL	<p>该标签用来创建控制单元。</p> <p>句法:</p> <pre><CONTROL name = "<control name>" xpos = "<X position>" ypos = "<Y position>" refvar = "<NC variable>" hotlink = "true" format = "<format>" /></pre> <p>属性:</p> <ul style="list-style-type: none"> • name 域的名称 名称会同时显示一个局部变量，且无法在窗体中多次使用。 • 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。 <ul style="list-style-type: none"> - edit 并可对数据进行修改 - readonly 无法对数据进行修改 <p>combobox 域显示相应的名称来代替数值。 如果选择了域类型“combobox”，则要为域另外补充显示表达式。 为此要使用标签 <ITEM>。 复选框将当前选定文本的索引储存在控制单元所属的变量中（参见属性 refvar）。</p> <ul style="list-style-type: none"> - progressbar 在值域 0 到 100 内显示进度条。 值域与待显示数据的最小值和最大值相匹配。

标签名称	含义
CONTROL 续	<ul style="list-style-type: none"> • fieldtype <ul style="list-style-type: none"> – listbox <p>该域类型创建一个空的列表框控制项。</p> <p>可使用标签 <ITEM> 将列表框元素插入列表框中。</p> <p>ITEM 属性 value 允许赋予该元素唯一值。</p> <p>该操作，例如，可用于识别元素。</p> <p>参数 width 和 height 指定列表框的宽度和高度。</p> <p>创建控制项之后，可使用函数 AddItem、InsertItem 或 LoadItem 插入其他列表框元素。</p> – graphicbox <p>该域类型创建一个 2 维虚线图形控制项。</p> <p>可使用标签 <ITEM> 将图形元素插入控制项。</p> <p>参数 width 和 height 指定框的宽度和高度。</p> <p>注意： 该控制项不关联“剪切”功能。</p> <p>因而，其他元素可覆盖该控制项。</p> <p>创建控制项之后，可使用函数 AddItem 或 InsertItem 插入其他元素。对该控制项不进行参数 itemdata 评估。</p> – itemlist <p>该域类型创建一个静态控制项，该控制项显示相应的名称来代替数值。</p> <p>可使用 <ITEM> 标签赋予该域一个名称。</p> • item_data <p>可以使用用户专用的整数值为该属性赋值。该值被提供给 FOCUS_IN 信息，用来识别焦点域。</p> • refvar <p>可连接到该域的参考变量的名称（可选）。</p> • hotlink = "TRUE"当参考变量数值改变时，该域自动更新（可选）。 • format <p>属性定义了指定变量的显示格式。</p> <p>格式化数据，参见 print-Tag（可选）。</p>

标签名称	含义
CONTROL 续	<p>属性:</p> <ul style="list-style-type: none"> • time 给定数据的刷新率（可选）。 可以使用下列数据： <ul style="list-style-type: none"> - super fast 刷新时间 < 100 ms - fast 刷新时间约 100 ms - normal 刷新时间约 200 ms - slow 刷新时间约 500 ms • font 该属性确定所使用的字体大小。 <ul style="list-style-type: none"> - 0: 8*8 - 1: 16*8 - 2: 24*16（仅用于数字） - 3: 8*8 两倍字符高度 - 4: 16*8 两倍字符高度 - 5: 24*16 两倍字符高度（仅用于数字） • color_bk 此属性可设置控制项的背景色。 • color_fg 此属性可设置控制项的前景色。 （“颜色代码”参见章节“颜色代码 (页 266)”） • display_format 属性定义了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： <ul style="list-style-type: none"> - FLOAT - INT - DOUBLE - STRING <p>向列表框、图形框或复选框分配表达内容（例如要显示的文本或图形元素）： 句法: <ITEM>表达式</ITEM></p>

标签名称	含义
CONTROL 续	<p>示例:</p> <pre><CONTROL name = "button1" xpos = "10" ypos = "10" fieldtype = " combobox "> <ITEM>text1</ITEM> <ITEM>text2</ITEM> <ITEM>text3</ITEM> <ITEM>text4</ITEM> </CONTROL></pre> <p>如果要为表达式分配任意一个整数值，则要为标签添加属性 value = “数值”。控制变量含有已赋值的对象值，用来代替连续的编号。</p> <p>示例:</p> <pre><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></pre> <p>渐进条形图示例</p> <pre><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></pre> <p>列表框示例:</p> <pre><let name="item_string" type="string"></let> <let name="item_data" ></let></pre> <pre><CONTROL name="listbox1" xpos = "360" ypos="150" width="200" height="200" fieldtype="listbox" /></pre> <ul style="list-style-type: none"> • 添加元素: 可使用函数 additem 或 loaditem 添加元素。 • 清除内容: 可使用函数 empty 清空内容。 <pre><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></pre>

标签名称	含义
<p>CONTROL 续</p>	<p>图形框示例:</p> <pre><CONTROL name= "graphic" xpos = "8" ypos="23" width="300" height="352" fieldtype="graphicbox" /></pre> <ul style="list-style-type: none"> • 添加元素: 可使用函数 additem 或 loaditem 添加元素。 可以使用以下 2 维元素: <ul style="list-style-type: none"> - 直线 - l(inc) - 圆弧 - c(circle) - 点 - p(oint) 元素结构: <Elementtyp>; 坐标 • 行: l; xs; ys; xe, ye l - 直线符号 Xs - X 起始位置 Ys - Y 起始位置 Xe - X 结束位置 Ye - Y 结束位置 • 圆弧: C, xs, ys, xe, ye, cc_x, cc_y, r C - 圆弧符号 Xs - X 起始位置 Ys - Y 起始位置 Xe - X 结束位置 Ye - Y 结束位置 Cc_x - 圆心的 X 坐标 Cc_y - 圆心的 Y 坐标 • 半径: R • 点: P, x, y P - 点的符号 X - X 坐标 Y - Y 坐标 • 清除图形: 可使用函数 empty 清空内容。

标签名称	含义
CONTROL 续	<p>示例:</p> <pre><let name="item_string" type="string"></let> <let name="s_z" type="double">100</let> <let name="s_x" type="double">50</let> <let name="itemdata"></let> <control name= "gbox" xpos = "6" ypos="24" width="328" height="356" fieldtype="graphicbox" /> <print name ="item_string" text="p; %f; %f">s_z, s_x</print> <function name="control.additem">_T"gbox", item_string, itemdata</function> ... </pre> <p>示例 itemlist:</p> <pre><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> </pre>
HELP_CONTEXT	<p>该标签用来确定所要调用的帮助主题。在 INIT 初始化数据块中进行编程。在属性中给定的名称会加上前缀 <code>XmlUserDlg_</code> 并转发到帮助系统中。这样可以在线帮助主题中提取出帮助文件的关联结构。</p> <p>激活帮助系统的过程:</p> <ol style="list-style-type: none"> 1. 按下“信息”键。 2. 对话画面会显示“my_dlg_help”。 3. 分析程序会将表达式转换为“XmlUserDlg_my_dlg_help”。 4. 帮助系统激活。 5. 传递查找关键字“XmlUserDlg_my_dlg_help”。 <p>句法:</p> <pre><HELP_CONTEXT name="<context name>" /> </pre> <p>示例:</p> <pre>... <INIT> ... <CAPTION>my dialogue</CAPTION> <HELP_CONTEXT name="my_dlg_help" /> ... </INIT> </pre>

标签名称	含义
DATA_ACCESS	<p>该标签在储存用户输入数据时用来控制对话框窗口的特性。</p> <p>在 INIT 初始化标签中确定其特性。</p> <p>如未使用标签，则总是使用输入数据中间存储器。</p> <p>例外：属性 hotlink 设置为 true。</p> <p>属性：</p> <ul style="list-style-type: none"> • type = "TRUE" - 不对输入值进行缓存。对话框窗口直接将输入值复制到参考变量中。 • type = "FALSE" - 值只通过标签 UPDATA_DATA type = "FALSE" 复制到参考变量中。 <p>示例：</p> <pre><DATA_ACCESS type = "true" /></pre>
MENU	<p>该标签定义一个菜单，其中包含有软键说明与待打开的对话框。</p> <p>属性：</p> <ul style="list-style-type: none"> • name 菜单名称 <p>句法：</p> <pre><MENU name = "<menu name>"> ... <open_form ...> ... <SOFTKEY ...> </SOFTKEY> </MENU></pre>
NAVIGATION	<p>该标签用来确定所要调用的菜单。只能在一个软键程序块内使用该标签。</p> <p>句法：</p> <pre><NAVIGATION>menu name</NAVIGATION></pre> <p>示例：</p> <pre><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></pre>

标签名称	含义
OPEN_FORM	<p>该标签可以打开名称下指令的对话框窗口。</p> <p>属性:</p> <ul style="list-style-type: none"> • name 对话框窗体的名称 <p>句法: <code><OPEN_FORM name = "<form name>" /></code></p> <p>示例:</p> <pre> <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> </pre>

标签名称	含义
<p>PROPERTY</p>	<p>使用该标签可以确定操作单元的附加特性。</p> <p>属性:</p> <ul style="list-style-type: none"> • 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>” • font = “<字体编号>” • password = “<true>” - 所输入的字符显示为“*” • multiline = “<true>” - 在 EDIT 控制项中允许多行输入 • disable = “<true/false>” - 在 EDIT 控制项禁止/允许输入 <p>示例:</p> <pre><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></pre>
<p>SOFTKEY</p>	<p>该标签定义了软键的特性与反应。</p> <p>属性:</p> <ul style="list-style-type: none"> • position 软键的编号。1-8 水平软键， 9-16 垂直软键 <p>可以在软键程序块中确定下列其他动作。</p> <ul style="list-style-type: none"> • caption • navigation • update_controls • function <p>句法:</p> <pre><softkey position = "<1>"> </softkey></pre>

标签名称	含义
TEXT	<p>该标签用来在指定的位置上显示文本。</p> <p>如果使用报警编号，则对话框会显示为该编号所保存的文本。</p> <p>句法: <code><TEXT xpos = "<X 坐标>" ypos = "<Y 坐标>"> Text </TEXT></code></p> <p>属性:</p> <ul style="list-style-type: none"> • xpos 左上角的 X 位置 • ypos 左上角的 Y 位置 • color 文本颜色（颜色代码） <p>值: 待显示的文本</p>
IMG	<p>该标签用来在指定的位置上显示图像。支持 BMP 和 PNG 图像格式。</p> <p>句法: <code><IMG xpos = "<X 坐标>" ypos = "<Y 坐标>" name = "<名称>" /></code></p> <p>属性:</p> <ul style="list-style-type: none"> • 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”） <p>可选: 希望修改图片的原始大小时，可以使用属性 width 和 height 来确定尺寸。</p> <ul style="list-style-type: none"> • width 宽度（像素） • height 高度（像素） <p>示例: <code></code> <code></code></p>

标签名称	含义
BOX	<p>该标签可以在指定的位置以指定的颜色绘制一个填满的矩形。</p> <p>句法: <code><BOX xpos = "<X 坐标>" ypos = "<Y 坐标>" width = "<X 长度>" height = "<Y 长度>" color = "<颜色代码>" /></code></p> <p>属性:</p> <ul style="list-style-type: none"> • 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（颜色代码参见章节“颜色代码”）
FUNCTION	<p>函数调用</p> <p>该标签可以执行属性“name”中给出的。</p> <p>属性:</p> <ul style="list-style-type: none"> • name = “函数体的名称” • return = “用于储存函数结果的变量名称” <p>值:</p> <p>需要将其传输到函数体的变量列表。变量相互之间以逗号分隔。最多可以传输 10 个参数。</p> <p>此外，还可以将常数或文本表达式指定为调用参数。为了标记文本表达式需要在文本前放置标志 <code>_T</code>。</p> <p>句法: <code><FUNCTION name = "<function name>" /></code> 待调用的函数等待一个返回值 <code><FUNCTION name = "<function name>" return = "<Variablename>" /></code></p> <p>参数传递</p> <code><FUNCTION name = "<function name>" var1, var2, var3 </FUNCTION></code> <code><FUNCTION name = "<function name>" _T"Text", 1.0, 1 </FUNCTION></code> <p>示例:</p> <p>参见“FUNCTION_BODY”</p>

标签名称	含义
FUNCTION_BODY	<p>函数体</p> <p>该标签包含有一个子函数的函数体。D 要在对话框 DialogGui 标签中对函数体进行编程。</p> <p>属性:</p> <ul style="list-style-type: none"> • name = “函数体的名称” • parameter = “参数列表”（可选） <p>属性列出了所需传输参数。参数相互之间以逗号分隔。</p> <p>调用函数体时将函数调用中给定参数的值复制到所列出的传输参数中。</p> <ul style="list-style-type: none"> • return = "true" <p>如果属性为 true，则创建局部变量 \$return。将函数的返回值复制到其中，在放弃该函数时可以将其继续传输给待调用的函数。</p> <p>句法:</p> <p>无参数的函数体</p> <pre><FUNCTION_BODY name = "<function name>"> </ FUNCTION_BODY></pre> <p>带参数的函数体</p> <pre><FUNCTION_BODY name = "<function_name>" parameter = "<p1, p2, p3>"> ... <LET name = "tmp"></LET> <OP> tmp = p1 </OP> ... </FUNCTION_BODY></pre> <p>带返回值的函数体</p> <pre><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></pre>

标签名称	含义
FUNCTION_BODY 续	<p>示例:</p> <pre><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></pre>
REQUEST	<p>借助该标签可以在周期性读取服务（hotlink 热连接）中使用一个变量。这样可以缩短对未与控制项相连的变量的存取时间。</p> <p>如果在值变化时自动调用一个函数，则应将函数的名称作为其它属性加以定义。</p> <p>该标签仅在 INIT 初始化指令中进行处理。</p> <p>属性:</p> <ul style="list-style-type: none"> • name 地址识别符 <p>句法:</p> <pre><REQUEST name = "<NC-Variable>" /></pre>
UPDATE_CONTROLS	<p>该标签在操作单元和参考变量之间进行补偿。</p> <p>属性:</p> <ul style="list-style-type: none"> • type 该属性可以确定数据比较的方向。 = TRUE – 从参考变量读取数据并复制到操作单元中。 = FALSE – 从操作单元读取数据并复制到参考变量中。 <p>句法:</p> <pre><UPDATE_CONTROLS type = "<方向>" /></pre> <p>示例:</p> <pre><SOFTKEY_OK> < UPDATE_CONTROLS type="false" /> </SOFTKEY_OK></pre>

16.7.8.2 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>
```

16.7.9 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	含义
String.cmp	<p>从字典编辑角度对两个字符串进行相互比较。</p> <p>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。</p> <p>参数:</p> <p>str1 - 字符串</p> <p>str2 - 比较字符串</p> <p>句法:</p> <pre><function name="string.cmp" return ="<int var>" > str1, str2 </function></pre> <p>示例:</p> <pre><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></pre> <pre><function name="string.cmp" return="rval"> str1, str2 </function></pre> <p>结果:</p> <p>rval= 0</p>

函数名称	含义
String.icmp	<p>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。</p> <p>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。</p> <p>参数:</p> <p>str1 - 字符串 str2 - 比较字符串</p> <p>句法:</p> <pre><function name="string.icmp" return = "<int var>" > str1, str2 </function></pre> <p>示例:</p> <pre><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></pre> <p>结果:</p> <p>rval= 0</p>
String left	<p>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。</p> <p>参数:</p> <p>str1 - 字符串 nCount - 字符数量</p> <p>句法:</p> <pre><function name="string.left" return="<result string>"> str1, nCount </function></pre> <p>示例:</p> <pre><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></pre> <p>结果:</p> <p>str2="A brown bear"</p>

函数名称	含义
String.right	<p>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。</p> <p>参数:</p> <p>str1 - 字符串 nCount - 字符数量</p> <p>句法:</p> <pre><function name="string.right" return="<result string">"> str1, nCount </function></pre> <p>示例:</p> <pre><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></pre> <p>结果:</p> <p>str2="brown dog."</p>
String middle	<p>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。</p> <p>参数:</p> <p>str1 - 字符串 iFirst - 开始索引 nCount - 字符数量</p> <p>句法:</p> <pre><function name="string.middle" return="<result string">"> str1, iFirst, nCount </function></pre> <p>示例:</p> <pre><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></pre> <p>结果:</p> <p>str2="brown"</p>

函数名称	含义
String.length	<p>该函数可以提供字符串的字符数量。</p> <p>参数:</p> <p>str1 - 字符串</p> <p>句法:</p> <pre><function name="string.length" return="<int var>"> str1 </function></pre> <p>示例:</p> <pre><let name="length">0</let></pre> <pre><let name="str1" type="string">A brown bear hunts a brown dog.</let> <function name="string.length" return="length"> str1 </function></pre> <p>结果:</p> <p>length = 31</p>
Strings.replace	<p>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。</p> <p>参数:</p> <p>string - 字符串变量</p> <p>find string - 要替代的字符串</p> <p>new string - 新的字符串</p> <p>句法:</p> <pre><function name="<string.replace>"> string, find string, new string </function></pre> <p>示例:</p> <pre><let name="str1" type="string">A brown bear hunts a brown dog. </let></pre> <pre><function name="string.replace" > str1, _T"a brown dog" , _T"a big salmon"</function></pre> <p>结果:</p> <p>str1 = "A brown bear hunts a big salmon!"</p>

函数名称	含义
String.remove	<p>该函数可以删除所有搜索到的部分字符串。</p> <p>参数:</p> <p>string - 字符串变量</p> <p>remove string - 要删除的部分字符串</p> <p>句法:</p> <pre><function name="string.remove"> string, remove string </function></pre> <p>示例:</p> <pre><let name="index">0</let></pre> <pre><let name="str1" type="string">A brown bear hunts a brown dog. </let></pre> <pre><function name="string.remove" > str1, _T"a brown dog" </function</pre> <p>结果:</p> <pre>str1 = "A brown bear hunts"</pre>
Strings.insert	<p>该函数可以在指定的索引位置插入一个字符串。</p> <p>参数:</p> <p>string - 字符串变量</p> <p>index - 索引（以零为基准）</p> <p>insert string - 要插入的字符串</p> <p>句法:</p> <pre><function name="string.insert"> string, index, insert string </function></pre> <p>示例:</p> <pre><let name="str1" type="string">A brown bear hunts. </let></pre> <pre><let name="str2" type="string">a brown dog</let></pre> <pre><function name="string.insert" > str1, 19, str2</function></pre> <p>结果:</p> <pre>str1 = "A brown bear hunts a brown dog"</pre>

函数名称	含义
String delete	<p>该函数可以从指定的起始位置开始删除确定数量的字符。</p> <p>参数:</p> <p>string - 字符串变量</p> <p>start index - 起始索引（以零为基准）</p> <p>nCount - 要删除字符的数量</p> <p>句法:</p> <pre><function name="string.delete"> string, start index , nCount </function></pre> <p>示例:</p> <pre><let name="str1" type="string">A brown bear hunts. </let></pre> <pre><function name="string.delete" > str1, 2, 5</function></pre> <p>结果:</p> <pre>str1 = "A bear hunts"</pre>
String.find	<p>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。</p> <p>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。</p> <p>参数:</p> <p>string - 字符串变量</p> <p>findstring- 要搜索的字符串</p> <p>句法:</p> <pre><function name="string.find" return="<int val>"> str1, find string </function></pre> <p>示例:</p> <pre><let name="index">0</let></pre> <pre><let name="str1" type="string">A brown bear hunts a brown dog. </let></pre> <pre><function name="string.find" return="index"> str1, _T"brown" </function></pre> <p>结果:</p> <pre>Index = 2</pre>

函数名称	含义
String.reversefind	<p>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—处地方。</p> <p>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。</p> <p>参数:</p> <p>string - 字符串变量</p> <p>find string- 要搜索的字符串</p> <p>句法:</p> <pre><function name="string.reversefind" return="<intval>"> str1, find string </function></pre> <p>示例:</p> <pre><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></pre> <p>结果:</p> <p>Index = 21</p>
String.trimleft	<p>该函数可以从字符串中删除打头的空格。</p> <p>参数:</p> <p>str1 - 字符串变量</p> <p>句法:</p> <pre><function name="string.trimleft" > str1 </function></pre> <p>示例:</p> <pre><let name="str1" type="string"> test trim left</let> <function name="string.trimleft" > str1 </function></pre> <p>结果:</p> <p>str1 = "test trim left"</p>

函数名称	含义
String.trimright	<p>该函数可以从字符串中删除后续的空格。</p> <p>参数:</p> <p>str1 - 字符串变量</p> <p>句法:</p> <pre><function name="string.trimright" > str1 </function></pre> <p>示例:</p> <pre><let name="str1" type="string"> test trim right </let> <function name="string.trimright" > str1 </function></pre> <p>结果:</p> <p>str1 = "test trim right"</p>
sin	<p>该函数可以计算所传输值（以度为单位）的正弦值。</p> <p>参数:</p> <p>double - 角度</p> <p>句法:</p> <pre><function name="sin" return="<double val>"> double </function></pre> <p>示例:</p> <pre><let name= "sin_val" type="double"></let> <function name="sin" return="sin_val"> 20.0 </function></pre>
cos	<p>该函数可以计算所传输值（以度为单位）的余弦值。</p> <p>参数:</p> <p>double - 角度</p> <p>句法:</p> <pre><function name="cos" return="<double val>"> double </function></pre> <p>示例:</p> <pre><let name= "cos_val" type="double"></let> <function name="cos" return="cos_val"> 20.0 </function></pre>

函数名称	含义
tan	<p>该函数可以计算所传输值（以度为单位）的正切值。</p> <p>参数:</p> <p>double - 角度</p> <p>句法:</p> <pre><function name="tan" return="<double val>"> double </function></pre> <p>示例:</p> <pre><let name= "tan_val" type="double"></let> <function name="tan" return="tan_val"> 20.0 </function></pre>
arcsin	<p>该函数可以计算所传输值（以度为单位）的反正弦值。</p> <p>参数:</p> <p>double - x 取值范围 $-\pi/2$ 至 $+\pi/2$</p> <p>句法:</p> <pre><function name="arcsin" return="<double val>"> double </function></pre> <p>示例:</p> <pre><let name= "arcsin_val" type="double"></let> <function name="arcsin" return=" arcsin_val"> 20.0 </function></pre>
arccos	<p>该函数可以计算所传输值（以度为单位）的反余弦值。</p> <p>参数:</p> <p>double - x 取值范围 $-\pi/2$ 至 $+\pi/2$</p> <p>句法:</p> <pre><function name="arccos" return="<double val>"> double </function></pre> <p>示例:</p> <pre><let name= "arccos_val" type="double"></let> <function name="arccos" return=" arccos_val"> 20.0 </function></pre>

函数名称	含义
arctan	<p>该函数可以计算所传输值（以度为单位）的反正切值。</p> <p>参数:</p> <p>double - 反正切 y/x</p> <p>句法:</p> <pre><function name="arctan" return="<double val>"> double </function></pre> <p>示例:</p> <pre><let name= "arctan_val" type="double"></let> <function name="arctan" return="arctan_val"> 20.0 </function></pre>
dll.load	<p>该函数可以在存储器中装载其他的用户 DLL。</p> <p>参数:</p> <p>dll_name - DLL name</p> <p>class_name - “函数等级的名称”</p> <p>句法:</p> <pre><function name="dll.load"> dll_name, class_name </function></pre> <p>示例:</p> <pre><function name="dll.load"> _T"customer.dll", T"customer" </function></pre>
dll.function	<p>该函数可以调用用户 DLL 中的函数。将参数 ID 后面列出的所有参数传输给所调用的函数。</p> <p>参数:</p> <p>class_name - 函数等级的名称</p> <p>id - 函数标识</p> <p>parameter - 最多七个函数参数（字符串变量）</p> <p>句法:</p> <pre><function name="dll.function"> class_name, id, parameter1, parameter2</function></pre> <p>示例</p> <pre><function name="dll.function"> _T"customer", 290, T"par1", T"par2"</function></pre>

函数名称	含义
文件处理	
doc.readfromfile	<p>该函数可以将指定文件的内容装载到一个字符串变量中。</p> <p>属性:</p> <p>Return - 局部变量的名称</p> <p>参数:</p> <p>Programe - 文件名称</p> <p>句法:</p> <pre><function name="doc.readfromfile" return="<string var>"> programe </function></pre> <p>示例:</p> <pre><let name = "my_var" type="string" ></let></pre> <pre><function name=" doc.readfromfile " return="my_var"> T"\spf\test.mpf" </function></pre>
doc.writetofile	<p>该函数可以将一个字符串变量的内容写入指定文件中。</p> <p>参数:</p> <p>programe - 文件名称</p> <p>str1 - 字符串</p> <p>句法:</p> <pre><function name="doc.writetofile" > programe, str1 </function></pre> <p>示例:</p> <pre><let name = "my_var" type="string" > file content </let></pre> <pre><function name="doc.writetofile">_T"\spf\test.mpf", my var </function></pre>
doc.remove	<p>该函数可以从目录中删除指定的文件。</p> <p>参数:</p> <p>programe - 文件名称</p> <p>句法:</p> <pre><function name="doc.remove" > programe </function></pre> <p>示例:</p> <pre><function name="doc.remove">_T"\mpf\test.mpf" </function></pre>

函数名称	含义
doc.exist	<p>如文件存在，则函数提供值 1。</p> <p>参数：</p> <p>programe - 文件名称</p> <p>句法：</p> <pre><function name="doc.exist" return="<int_var>" > programe </function></pre> <p>示例：</p> <pre><let name ="exist">0</let></pre> <pre><function name="doc.exist" return="exist"> T"\mpf\test.mpf" </function></pre>
ncfunc.select	<p>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。</p> <p>参数：</p> <p>programe - 文件名称</p> <p>句法：</p> <pre><function name="ncfunc.select"> programe </function></pre> <p>示例：</p> <pre><function name="ncfunc.select"> _T"\mpf\test.mpf" </function></pre>

16.7 通过自定义 *EasyXLanguage* 脚本创建用户对话框

SINUMERIK 808D 的许可证

17.1 SINUMERIK 808D 的许可证

SINUMERIK 808D 许可证

数控系统 PPU 上的 PPU 软件从出厂时就已经获得授权。

厂方会按照具体要求为下列加工类型发放许可证：

- SINUMERIK 808D 车削版
- SINUMERIK 808D 铣削版

也可为 SINUMERIK 808D 车削版购买可选功能 **"Manual Machine Plus"** 和 **"Additional Axis"**。必须通过 HMI 用户界面在数控系统上激活这两个功能。

说明

随后可以从网络许可证管理器 (<http://www.siemens.com/automation/license>) 获取许可证。

17.2 网络许可证管理器

17.2.1 网络许可证管理器

通过网络许可证管理器 (<http://www.siemens.com/automation/license>), 您可以在标准网络浏览器上将许可证分配给硬件。若要结束分配过程, 则必须通过 HMI 用户界面手动将授权密钥输入到数控系统上。

17.2.2 分配许可证

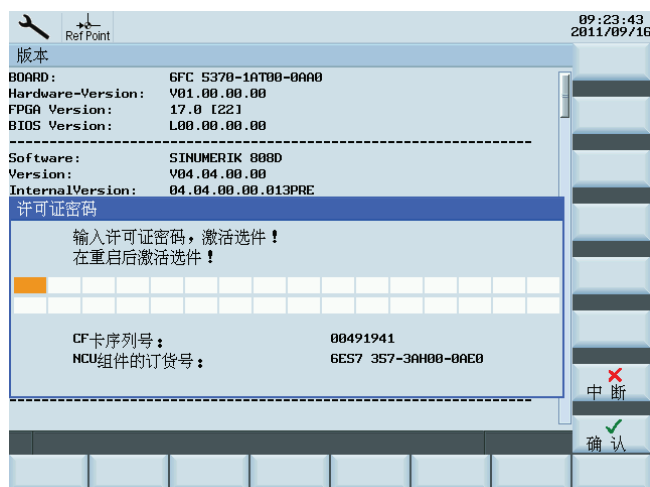
前提条件

必须满足下列前提条件, 以能够通过直接访问和 HMI 操作界面将许可证分配给硬件。

- 数控系统引导启动。
- 有用于直接访问 (例如通过 CoL) 的登录数据:
 - 许可证号
 - 派遣单号
- 数控系统类型
- 系统 CF 卡的“CF 卡序列号”

操作步骤

1. 在“系统”操作区中, 按下“服务显示” > “版本” > “授权密钥”来获得“CF 卡序列号”。



说明

确保所显示的“CF 卡序列号”是您想要进行分配的那个真正的序列号。许可证与硬件的分配可以通过网页许可证管理器进行，不得撤销。

2. 转到网络许可证管理器 (<http://www.siemens.com/automation/license>):

3. 通过“直接访问”登录:

- 许可证号
- 派遣单号

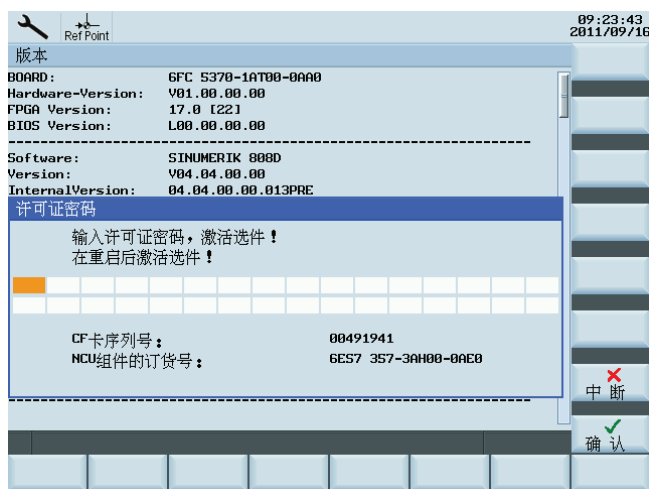
4. 在网络许可证管理器中，遵循下列指令并逐步进行操作。

5. 连接之后网络许可证管理器会显示出授权密钥。

此时能够进行:

- 记下密码。
- 将其储存为 PDF 文件。

6. 在完成分配过程之后，将显示在网络许可证管理器上的授权密钥输入到 HMI 用户界面的授权对话框中（在“系统”操作区中，按下“服务显示” > “版本” > “授权密钥”）。



7. 输入新的许可证密钥并按下软键“确定”。

8. 激活可选功能 **Manual Machine Plus** 和 **Additional Axis**。

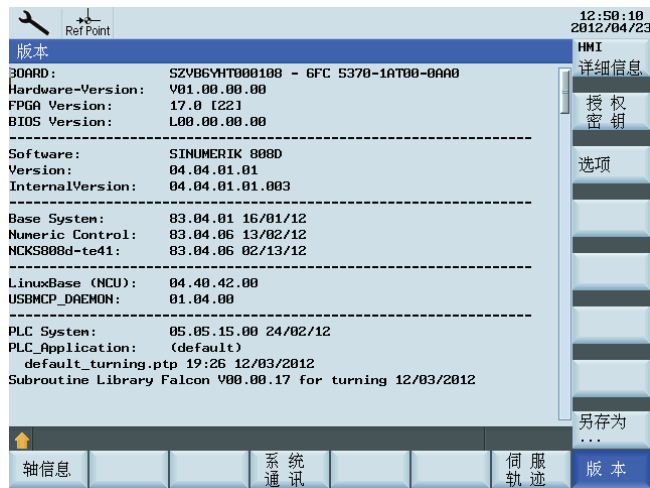
说明

关于激活可选功能的详细信息，请参见章节“激活可选功能 (页 304)”。

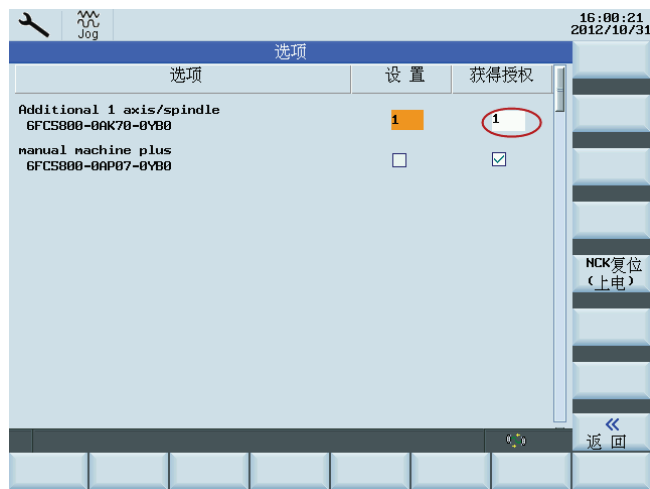
17.3 激活可选功能

欲激活 SINUMERIK 808D 车削版的可选功能 **Manual Machine Plus** 和 **Additional Axis**，可如下操作：

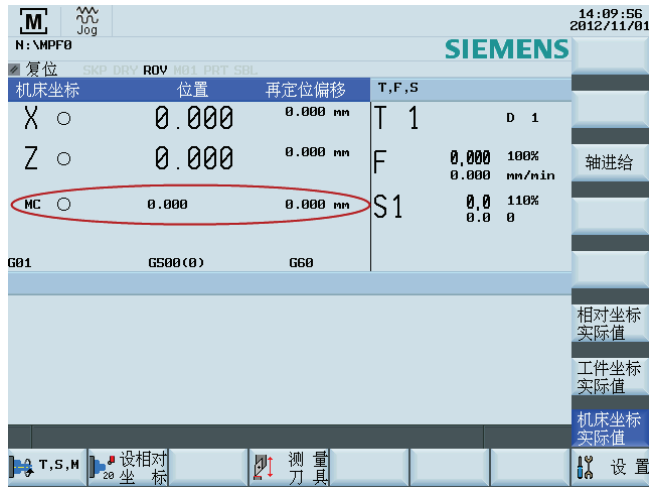
1. 在“系统”操作区中，按下“服务显示”>“版本”。显示如下（示例）：



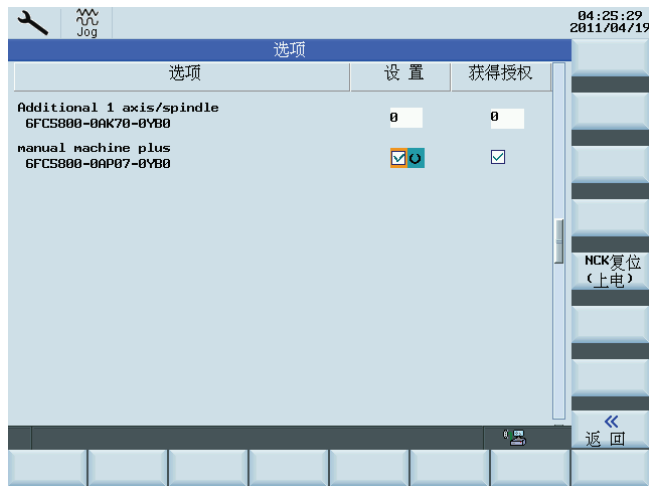
2. 按下“选项”。如果您已通过输入许可证密钥激活 **Additional Axis** 功能，获得授权一栏的值为 **1**。然后即可将设置一栏设为 **0** 或 **1**，分别表示 0 或 1 根附加轴已设置。



分配附加轴并正确设置相关机床数据后，附加轴可在一些操作区域显示。例如，“加工”操作区会如下显示：



或者，按下<选择>键选择软键选项 **Manual Machine Plus**。



3. 按下"NCK 复位（上电）"。数控系统上开始进行热启动。系统重启后，相关软件功能在数控系统中激活。

说明

如两个可选功能未经授权，报警 **8081** 出现在屏幕上：“设置了 1 个或多个未经授权密钥许可的选项”。必须输入有效许可证密码，否则无法正常操作机床。

激活后如要使用附加轴功能，首先须设置以下机床数据：

- MD10000[1]: 将"Y"改为另一个符号，除大写字母 **C**，例如，"**MC**"。
- MD20070[3]: 将"0"改为"**2**"。
- MD20080[3]: 将该值设为"**C**"。
- 必要时，设置以下机床数据，其中 MD30300 决定附加轴作为回转轴或线性轴。
 - MD30300
 - MD30310
 - MD30320

17.4 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

17.5 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	说明
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。 <ul style="list-style-type: none"> • SINUMERIK 808D: 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： <ul style="list-style-type: none"> • CoL（许可证书） • 授权密钥
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： <ul style="list-style-type: none"> • 产品名称 • 许可证号 • 交货单号 • 硬件序列号 注意： 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。

术语	说明
系统 CF 卡	<p>系统 CF 卡作为所有 SINUMERIK solution line 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：</p> <ul style="list-style-type: none"> • 硬件序列号 • 许可证信息包括授权密钥
硬件序列号	<p>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：</p> <ul style="list-style-type: none"> • CoL（参见：许可证证书>“提示”） • HMI 操作界面（“系统”操作区 > “服务显示” > “版本”> “授权密钥”） • 系统 CF 卡上的印刷
授权密钥	<p>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。</p>
选件	<p>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。</p>
产品	<p>在 SINUMERIK 软件产品许可证管理范围内，通过下列数据标明产品：</p> <ul style="list-style-type: none"> • 产品名称 • 订货号 • 许可证号

索引

A

ASUP

初始化, 26

启动, 110

C

CF 卡, 307

CoL, 306

D

D功能, 216

J

JOG

运行至固定点, 70

P

PLC 到 NCK 的信号, 16

PLC 服务显示, 145

PLC/NCK 的接口, 15

Prog Event, 101

S

SINUMERIK 808D 许可证, 301

T

TEACH IN, 90

T功能, 84

X

XML

运算符, 267

语法, 266

XML 函数

arccos, 296

arcsin, 296

arctan, 297

cos, 295

string.delete, 293

doc.remove, 298

dll.function, 297

doc.exist, 299

string.find, 293

string.reversefind, 294

dll.load, 297

doc.readfromfile, 298

ncfunc.select, 299

sin, 295

string.cmp, 288, 289

string.icmp, 289

string.insert, 292

string.left, 289

string.length, 291

string.middle, 290

string.remove, 292

- string.replace, 291
 - string.right, 290
 - tan, 296
 - doc.writetofile, 298
 - XML 标签
 - BOX, 284
 - BREAK, 254
 - CAPTION, 273
 - CLOSE, 273
 - CLOSE_FORM, 273
 - CONTROL, 274
 - CONTROL_RESET, 254
 - DATA, 254
 - DATA_ACCESS, 280
 - DATA_LIST, 255
 - DO_WHILE, 265
 - ELSE, 255
 - FOCUS_IN, 272
 - FOR, 264
 - FORM, 256, 271
 - FUNCTION, 284
 - FUNCTION_BODY, 285
 - HEPL_CONTEXT, 279
 - HMI_RESET, 256
 - IF, 257
 - IMG, 283
 - INCLUDE, 257
 - INIT, 272
 - LET, 258
 - MENU, 280
 - MSG, 260
 - MSGBOX, 260
 - NAVIGATION, 280
 - OP, 261
 - OPEN_FORM, 281
 - PAINT, 272
 - PASSWORD, 261
 - POWER_OFF, 261
 - PRINT, 262
 - PROPERTY, 282
 - REQUEST, 286
 - SOFTKEY, 282
 - STOP, 263
 - SWITCH, 263
 - TEXT, 283
 - THEN, 263
 - TIMER, 273
 - UPDATE_CONTROLS, 286
 - WAITING, 264
 - WHILE, 265
- ## 二划
- 刀具, 216
- T 的值范围: , 216
 - 刀具补偿, 216
 - 选择, 216
 - 换刀时选择刀沿, 216
 - 激活刀具补偿, 216
- ## 四划
- 分组辅助功能, 82
 - 分配许可证, 302
 - 手轮
 - JOG 方式下的运行, 67
 - 移动轴, 61
- ## 五划
- 丝杠螺距误差补偿数据 (LEC) , 137
 - 主轴
 - 主轴监控, 192

同步, 184
 变速换档, 186
 主轴运行方式, 174
 主轴修调系数, 212
 加速度, 63
 加速度属性, 59
 加速度, 带过冲限制, 59
 跃变式加速度, 59
 可插补补偿
 补偿表, 135
 线性插补, 136

六划

产品, 307
 合理性检测, 167
 回参考点
 带增量测量系统, 159
 轴专用, 158
 通道专用, 158
 在 TP 中读取测量结果, 145
 许可证, 306
 许可证号, 306
 轨迹进给率 F, 201
 G33 螺纹加工的进给率, 203
 G63 (带补偿夹具进行攻丝)时的进给率, 205
 报警, 202
 轨迹轴, 49
 过冲降低, 54
 过冲限制, 54
 过载系数, 53

七划

位置显示, 133
 位置控制增益, 38
 快速进给修调, 63

快速进给修调开关, 211
 补偿表, 137
 运动监控功能, 35
 运行方式
 功能, 94
 运行方式转换时出错, 91
 监控, 93
 禁止方式转换, 92
 运行方式转换, 91
 运行至固定点
 在 JOG 运行模式中, 70
 使用 G75, 71
 进给/主轴停止, 210
 进给轴监控
 夹紧, 40
 定位监控, 37
 实际速度, 42
 转速给定值, 41
 零速, 39
 进给倍率开关, 211
 进给率
 主轴修调, 212
 轨迹进给率 F, 201
 进给/主轴停止, 210
 进给轴专用的进给修调, 212
 进给率修调, 211
 进给率控制, 208
 进给锁定, 209, 210
 带补偿夹具的攻丝, G63, 205
 螺纹切削 G33, 203
 进给率倍率, 63
 进给锁定, 210
 间隙补偿, 133

八划

事件控制的程序过程

- 复位操作面板, 105
- 零件程序开始和零件程序结束, 105
- 其余程序段速度, 57
- 单程序段, 116
- 固定点位置, 73
- 定位窗口, 38
- 所用互联网链接, 306
- 空运行进给率, 117
- 线性轴
 - 限位开关监控, 43
- 轮廓损坏, 45
- 软件产品, 306
- 软件限位开关, 64

九划

- 复位, 155
- 急停
 - 过程, 154
 - 应答, 155
 - 接口, 154
- 急停按钮, 153
- 持续运行, 65
- 指令 MEAS, MEAW, 144
- 按过载系数降低速度, 53
- 测速发电机调节, 42
- 测量头, 142
- 测量头连接, 143
- 测量头的功能测试, 146
- 测量系统误差补偿, 137
- 测量精度, 146
- 轴相关的过冲限制, 60
- 选件, 307

十划

- 准停, 50

- 准停标准, 51
- 通道, 89
- 通道状态, 100
- 速度降低, 54
- 配置文件, 249
- 预读, 52, 55

十一划

- 授权密钥, 307
- 菜单树, 249
- 辅助功能, 79
- 辅助功能输出
 - 程序段切换, 81
 - 程序段搜索, 84
- 隐含准停功能, 52

十二划

- 循环信号交换, 16
- 插补终点, 51
- 登入软键, 249
- 硬件, 306
- 硬件序列号, 307
- 硬件限位开关, 64
- 程序状态, 99
- 程序运行
 - 启动零件程序或者零件程序段, 96
 - 复位指令, 98
 - 程序状态, 99
 - 零件程序中断, 97
 - 零件程序选择, 95
- 程序段搜索, 84, 118
- 程序测试
 - 处理特定程序段, 118
 - 程序单段运行(SBL), 116
 - 程序段搜索, 118

零件程序段跳跃, 121

程序控制, 98

程序操作, 89

十三划

路径相关的过冲限制, 59

零件程序中断, 97

零件程序段跳跃, 121

十五划

增量运行, 66

十六划

激活可选功能, 304

