

SIEMENS

SINUMERIK

SINUMERIK 828D 调试

开机调试手册

前言

基本安全说明

1

CNC 调试

2

SINUMERIK Operate

3

Easy XML

4

SINUMERIK Integrate

Access MyMachine /OPC UA

5

SINUMERIK Integrate

Run MyScreens

6

附件

A

适用于：

数控软件 V4.7 SP1




01/2015

6FC5397-3DP40-5RA2

法律资讯

警告提示系统

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

 危险
表示如果不采取相应的小心措施， 将会 导致死亡或者严重的人身伤害。
 警告
表示如果不采取相应的小心措施， 可能 导致死亡或者严重的人身伤害。
 小心
表示如果不采取相应的小心措施，可能导致轻微的人身伤害。
注意
表示如果不采取相应的小心措施，可能导致财产损失。

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


合格的专业人员

本文件所属的产品/系统只允许由符合各项工作要求的**合格人员**进行操作。其操作必须遵照各自附带的文件说明，特别是其中的安全及警告提示。

由于具备相关培训及经验，合格人员可以察觉本产品/系统的风险，并避免可能的危险。

按规定使用Siemens 产品

请注意下列说明：

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

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 文档

SINUMERIK 文档分为以下几个类别：

- 通用文档
- 用户文档
- 制造商/维修文档

其它信息

访问网页Hotspot-Text (www.siemens.com/motioncontrol/docu)可以查看以下内容：

- 订购文档/查看印刷品一览
- 进入下载文档的链接
- 使用在线文档（查找搜索手册/信息）

如果您对技术文档有疑问（例如：建议或修改），请发送一份电子邮件到下列地址：
(<mailto:docu.motioncontrol@siemens.com>)

我的文档管理器（MDM）

点击下面的链接，您可以在西门子文档内容的基础上创建自己的机床文档：**MDM**
(www.siemens.com/mdm)

培训

提供的培训课程有：

- SITRAIN (www.siemens.com/sitrain) - 西门子自动化产品、系统以及解决方案的培训
- SinuTrain (www.siemens.com/sinustrain) - SINUMERIK 培训软件

常见问题

常见问题（FAQ）请点击“产品支持”Hotspot-Text
(www.siemens.com/automation/service&support)，然后点击右侧的“支持”。

SINUMERIK

SINUMERIK 的信息点击Hotspot-Text (www.siemens.com/sinumerik):

目标用户

该手册供调试人员使用。

设备/系统已经安装连接完毕，随时可以使用。

本调试手册包含了后续步骤，如：单个组件配置和组态所需的所有信息或者必要说明。

使用

阅读本调试手册后，相关人员可以专业、安全地检测并调试系统或者设备。

适用阶段： 安装和调试阶段

标准功能范畴

本文档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。

控制系统有可能执行本文档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，所以本文档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所有情况。

技术支持

各个国家技术咨询的电话号码请点击Hotspot-Text (www.siemens.com/automation/service&support)，然后点击“联系”。

欧盟一致性声明

EMC 指令的欧盟一致性声明请点击 (<http://support.automation.siemens.com/WW/view/de/10805517/134200>)。

目录

前言	3
1 基本安全说明	7
1.1 一般安全说明	7
1.2 操作静电敏感元器件 (ESD).....	10
1.3 工业安全	11
1.4 驱动系统（电气传动系统）的遗留风险	11
2 CNC 调试.....	15
3 SINUMERIK Operate	393
4 Easy XML	817
5 SINUMERIK Integrate Access MyMachine /OPC UA	971
6 SINUMERIK Integrate Run MyScreens	1001
A 附件	1259
A.1 SINUMERIK 828D 文档一览.....	1259

基本安全说明

1.1 一般安全说明



危险

接触带电部件和其他能源供给会引发生命危险

接触带电部件会造成人员重伤，甚至死亡。

- 只有专业人员才允许在电气设备上作业。
- 在所有作业中必须遵守本国的安全规定。

通常有六项安全步骤：

1. 做好断电的准备工作，并通知会受断电影响的组员。
2. 断开设备电源。
 - 关闭设备。
 - 请等待至警告牌上说明的放电时间届满。
 - 确认导线与导线之间和导线与接地线之间无电压。
 - 确认辅助电压回路已断电。
 - 确认电机无法运动。
3. 检查其他所有危险的能源供给，例如：压缩空气、液压、水。
4. 断开所有危险的能源供给，措施比如有：闭合开关、接地或短接或闭合阀门。
5. 确定能源供给不会自动接通。
6. 确保正确的设备已经完全闭锁。

结束作业后以相反的顺序恢复设备的就绪状态。




警告

连接了不合适的电源所产生的危险电压可引发生命危险


接触带电部件可能会造成人员重伤或死亡。

- 所有的连接和端子只允许使用可以提供 SELV(Safety Extra Low Voltage: 安全低压) 或 PELV(Protective Extra Low Voltage: 保护低压) 输出电压的电源。





 警告
接触损坏设备上的带电压部件可引发生命危险 未按规定操作设备可能会对其造成损坏。 设备损坏后，其外壳或裸露部件可能会带有危险电压，接触外壳或这些裸露部件可能会导致重伤或死亡。 <ul style="list-style-type: none">• 在运输、存放和运行设备时应遵循技术数据中给定的限值。• 不要使用已损坏的设备。



 警告
电缆屏蔽层未接地可引起电击从而导致生命危险 电缆屏蔽层未接地时，电容超临界耦合可能会出现致命的接触电压。 <ul style="list-style-type: none">• 电缆屏蔽层和未使用的电缆芯线至少有一侧通过接地的外壳接地。



 警告
未接地可引起电击从而导致生命危险 防护等级 I 的设备缺少安全接地连接或连接出错时，在其裸露的部件上会留有高压，接触该部件会导致重伤或死亡。 <ul style="list-style-type: none">• 按照规定对设备进行接地。

 警告
外壳大小空间不足可引起火灾从而导致生命危险 明火和烟雾可引起重大人员伤亡或财产损失。 <ul style="list-style-type: none">• 没有保护外壳的设备应安装在金属机柜中（或采取相同效果的措施进行保护），以避免设备接触明火。• 确保烟雾只能经所设安全通道排出。

**警告****使用移动无线电装置或移动电话时机器的意外运动可引发生命危险**

在距离本组件大约 2 m 的范围内使用发射功率大于 1 W 的移动无线电设备或移动电话时，会导致设备功能故障，该故障会对设备功能安全产生影响并能导致人员伤亡或财产损失。

- 关闭设备附近的无线电设备或移动电话。

**警告****通风不足会引起过热、引发火灾，从而导致生命危险**

通风空间不足会导致过热，产生烟雾，引发火灾，从而造成人身伤害。这可能会造成人员重伤或死亡。此外，设备/系统故障率可能会因此升高，使用寿命缩短。

- 组件之间应保持规定的最小间距，以便通风。

**警告****无效的安全功能可导致生命危险**

无效的或不适合的安全功能可引起机器功能故障，可能导致重伤或死亡。

- 调试前请注意相关产品文档中的信息。
- 对整个系统和所有安全相关的组件进行安全监控，以确保安全功能。
- 进行适当设置，以确保所使用的安全功能是与驱动任务和自动化任务相匹配并激活的。
- 执行功能测试。
- 在确保了机器的安全功能能正常工作后，才开始投入生产。

说明**Safety Integrated 功能的重要安全说明**

使用 Safety Integrated 功能时务必要注意 Safety Integrated 手册中的安全说明。

1.2 操作静电敏感元器件 (ESD)



警告

因参数设置错误或修改参数设置引起机器误操作可引发生命危险

参数设置错误可导致机器出现误操作，从而导致人员重伤或死亡。

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

1.2 操作静电敏感元器件 (ESD)

静电敏感元器件 (ESD)

是可被静电场或静电放电损坏的元器件、集成电路、电路板或设备。



注意

电场或静电放电可损坏设备

电场或静电放电可能会损坏单个元件、集成电路、模块或设备，从而导致功能故障。

- 仅允许使用原始产品包装或其他合适的包装材料（例如：导电的泡沫橡胶或铝箔）包装、存储、运输和发运电子元件、模块和设备。
- 只有采取了以下接地措施之一，才允许接触元件、模块和设备：
 - 佩戴防静电腕带
 - 在带有导电地板的防静电区域中穿着防静电鞋或配带防静电接地带
- 电子元件、模块或设备只能放置在导电性的垫板上（带防静电垫板的工作台、导电的防静电泡沫材料、防静电包装袋、防静电运输容器）。

1.3 工业安全

说明

工业安全

西门子为其产品及解决方案提供工业安全功能，以支持工厂、解决方案、机器、设备和/或网络的安全运行。这些功能是整个工业安全机制的重要组成部分。

有鉴于此，西门子不断对产品和解决方案进行开发和完善。西门子强烈建议您定期了解产品更新和升级信息。

此外，要确保西门子产品和解决方案的安全操作，还须采取适当的预防措施（例如：设备单元保护机制），并将每个组件纳入先进且全面的工业安全保护机制中。

可能使用的所有第三方产品须一并考虑。更多有关工业安全的信息，请访问网址 (<http://www.siemens.com/industrialsecurity>)。

要及时了解有关产品的更新和升级信息，请订阅相关产品的时事通讯。

更多相关信息请访问网址 (<http://support.automation.siemens.com>)。



警告

篡改软件会引起不安全的驱动状态从而导致危险

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

- 请使用最新版软件。

相关信息和新闻请访问网址 (<http://support.automation.siemens.com>)。

- 根据当前技术版本，将自动化组件和驱动组件整合至设备或机器的整体工业安全机制中。

更多相关信息请访问网址 (<http://www.siemens.com/industrialsecurity>)。

- 在整体工业安全机制中要注意所有使用的产品。

1.4 驱动系统（电气传动系统）的遗留风险

驱动系统的控制组件和传动组件允许用于工业电网内的工业和商业场合。

在民用电网中使用时，要求采取特殊设计或附加措施。

这种组件只允许在封闭的壳体或控制柜内运行，并且必须安装保护装置和保护盖。

只有经过培训、了解并遵循组件和用户手册上指出的所有安全注意事项的专业技术人员，才可以在组件上开展工作。

1.4 驱动系统（电气传动系统）的遗留风险

机器制造商在依据相应的本地指令（比如：欧盟机械指令）对机器进行风险评估时，必须注意驱动系统的控制组件和驱动组件会产生以下遗留风险：

1. 调试、运行、维护和维修设备时，被驱动的机器部件意外运行，原因可能有：
 - 编码器、控制器、执行器和连接器中出现了硬件故障和/或软件故障
 - 控制器和传动设备的响应时间
 - 运行和/或环境条件不符合规定
 - 凝露/导电杂质
 - 参数设置、编程、布线和安装出错
 - 在控制器附近使用无线电装置/移动电话
 - 外部影响/损坏
2. 在出现故障时，变频器内外部出现异常温度、明火以及异常亮光、噪音、杂质、气体等，原因可能有：
 - 零件失灵
 - 软件故障
 - 运行和/或环境条件不符合规定
 - 外部影响/损坏

防护等级为“开放式类型/IP20”的设备必须安装在金属机柜中（或采取相同效果的措施进行保护），以避免变频器内外部接触明火。
3. 出现危险的接触电压，原因可能有：
 - 零件失灵
 - 静电充电感应
 - 静充电感应
 - 运行和/或环境条件不符合规定
 - 凝露/导电杂质
 - 外部影响/损坏

4. 设备运行中产生的电场、磁场和电磁场可能会损坏近距离的心脏起搏器支架、医疗植入体或其它金属物。
5. 当不按照规定操作以及/或违规处理废弃组件时，会释放破坏环境的物质并且产生辐射。

说明

必须采取措施防止导电异物进入各组件，例如：将组件装入符合 EN 60529 IP54 防护等级或符合 NEMA 12 的控制柜中。

如果安装地点排除了导电异物，则使用较低防护等级的控制柜。

其它有关驱动系统组件产生的遗留风险的信息见用户技术文档的相关章节。

1.4 驱动系统（电气传动系统）的遗留风险

SIEMENS

SINUMERIK

SINUMERIK 828D CNC 调试

开机调试手册

供货范围和前提条件	1
数据级的导入与应用	2
操作软件的设置	3
NC 机床数据设置	4
调试驱动	5
优化驱动	6
维护管理器	7
Easy Extend	8
刀具管理	9
Easy Archive	10
附录	A




适用于:

数控软件 V4.7 SP1

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。

目录

1	供货范围和前提条件.....	9
1.1	系统概述.....	9
1.2	调试工具和服务工具.....	11
1.3	调试过程.....	13
1.3.1	手册编排结构.....	13
1.4	启动控制系统.....	17
1.5	建立和控制系统的通讯.....	20
1.5.1	通过 X130 建立和控制系统的通讯.....	20
1.5.2	通过编程工具建立和控制系统的通讯的步骤.....	22
1.5.3	这样通过 Access MyMachine 与控制系统进行通讯.....	23
2	数据级的导入与应用.....	27
2.1	NCK 中的数据级.....	28
2.2	PLC 中的数据级.....	31
2.3	操作软件中的数据级.....	33
3	操作软件的设置.....	37
3.1	存取级别.....	37
3.2	设置和修改口令的步骤.....	39
3.3	设置日期和时间的步骤.....	40
3.4	设置操作软件的语言.....	41
3.4.1	提供的标准系统语言和扩展语言.....	41
3.4.2	删除其他语言.....	42
3.4.3	使用输入法编辑器输入亚洲字符。.....	42
3.4.4	输入汉字的步骤。.....	44
3.4.5	输入韩语字符的步骤.....	45
3.5	检查和输入许可证.....	47
3.5.1	输入许可证密码的步骤.....	48
3.5.2	检查缺少的许可证/选件的步骤.....	49
3.5.3	许可证管理的相关定义.....	50
3.6	定义 PLC 报警.....	53
3.6.1	PLC 报警的结构.....	53
3.6.2	创建 PLC 报警的步骤.....	55
3.6.3	配置报警记录.....	57
3.6.4	配置记录选项的步骤.....	57

3.7	创建 OEM 专用在线帮助.....	60
3.7.1	配置文件的结构和句法.....	61
3.7.2	帮助手册的结构和句法.....	63
3.7.3	在线帮助采用的句法的描述.....	64
3.7.4	示例：创建 OEM 专用在线帮助手册的步骤.....	68
3.7.5	示例：创建针对用户 PLC 报警的在线帮助的步骤.....	72
3.7.6	示例：创建针对 NC /PLC 变量的在线帮助的步骤.....	74
3.7.7	示例：创建编程在线帮助的步骤.....	77
4	NC 机床数据设置.....	81
4.1	机床数据的分类.....	81
4.2	执行外部 CNC 系统的零件程序.....	84
4.3	用 Advanced Surface 加工自由形状.....	85
4.4	Advanced Surface 的机床数据.....	86
4.5	用于模拟量主轴的机床数据.....	90
4.6	禁用控制系统（数控锁功能）.....	94
4.7	管理数据.....	96
4.7.1	在控制系统内传输数据的步骤.....	97
4.7.2	保存和装载数据的步骤.....	98
4.7.3	比较数据的步骤.....	99
4.8	带自适应特性曲线的摩擦补偿.....	100
4.8.1	带速度接通脉冲的摩擦补偿.....	101
4.8.2	带力矩接通脉冲的摩擦补偿.....	103
4.8.3	数据列表.....	104
4.8.4	示例：线形轴的摩擦补偿.....	104
5	调试驱动.....	109
5.1	调试驱动.....	109
5.1.1	配置示例 - 概览.....	109
5.1.2	自动首次调试.....	111
5.1.3	示例：检查电源配置（S120 Combi）.....	114
5.1.4	示例：配置主轴（DO3）.....	115
5.1.5	示例：配置轴（DO4）编码器.....	124
5.1.6	示例：检查轴（DO6）配置.....	128
5.1.7	示例：配置电源（S120 书本型）.....	133
5.1.8	更换 SINAMICS S120 组件.....	139
5.1.9	示例：包含 TM120 的并联.....	140
5.1.10	交换过程数据.....	142
5.2	轴指定.....	143
5.2.1	示例：分配轴的步骤.....	143
5.2.2	示例：设置进给轴/主轴的机床数据.....	146

5.3	调试辅助轴.....	149
5.3.1	配置示例 - 概览.....	149
5.3.2	示例: 配置辅助轴的步骤.....	151
5.4	DRIVE-CLiQ 拓扑规则.....	154
5.4.1	S120 Combi 的拓扑规则.....	154
5.4.2	S120 书本型的拓扑规则.....	156
5.4.3	SMC40 的拓扑结构规定.....	160
5.5	端子布局.....	161
5.5.1	X122 和 X132 的引脚布局.....	161
5.5.2	X242 和 X252 的端子布局.....	162
5.5.3	示例: CU 和电源接触器的连接.....	164
5.5.4	输入/输出(对话框).....	166
5.5.5	互联(对话框).....	168
5.5.6	连接测量头.....	169
5.6	配置数据组.....	171
5.6.1	数据组一览.....	171
5.6.2	添加数据组.....	172
5.6.3	删除数据组.....	178
5.6.4	修改数据组.....	181
6	优化驱动.....	183
6.1	控制性能.....	183
6.2	自动伺服优化.....	186
6.2.1	启动前的选项.....	186
6.2.2	选择优化方案.....	187
6.2.3	启动自动伺服优化的步骤.....	192
6.2.4	开始轨迹插补.....	195
6.3	优化轴急动度.....	199
6.3.1	检查轴急动度.....	199
6.3.2	轴急动度的测试程序.....	200
6.3.3	选择跟踪信号.....	201
6.3.4	优化急动度设置.....	205
6.4	转矩使用率.....	208
6.5	优化主轴.....	212
6.5.1	设置主轴的机床数据.....	212
6.5.2	主轴: 检查转速控制器的方式.....	214
6.5.3	主轴: 检查位置控制器的方式.....	220
6.6	圆度测试.....	223
6.6.1	圆度测试: 功能.....	223
6.6.2	圆度测试: 执行测量.....	224
6.6.3	圆度测试: 示例.....	226

6.6.4	圆度测试：保存数据.....	229
7	维护管理器.....	233
7.1	PLC 用户程序中的接口.....	235
7.2	操作软件接口.....	240
7.3	导入和导出维护任务的步骤.....	243
7.4	应答维护任务.....	246
8	Easy Extend.....	247
8.1	功能概述.....	247
8.2	PLC 用户程序中的配置.....	249
8.3	机床制造商和经销商的选件位.....	251
8.4	操作界面上的显示画面.....	254
8.5	建立与语言有关的文本.....	255
8.6	示例.....	257
8.6.1	调试支持参数示例.....	257
8.6.2	控件示例.....	259
8.6.3	功率部件示例.....	260
9	刀具管理.....	263
9.1	基础部分.....	263
9.1.1	刀具管理 - 概览.....	263
9.1.2	刀具管理的结构.....	264
9.1.3	刀具管理的组件.....	266
9.1.4	多刀.....	268
9.1.5	手动刀具.....	269
9.2	刀具管理的机床数据.....	271
9.3	NC ↔ PLC 用户接口.....	276
9.3.1	换刀、卸刀、装刀、刀库定位.....	278
9.3.2	换刀.....	286
9.3.3	传输步骤表和应答步骤表.....	295
9.4	PLC 程序块.....	298
9.4.1	应答过程.....	298
9.4.2	应答类型.....	298
9.4.3	应答状态.....	300
9.4.4	配置步骤表.....	305
9.4.5	配置应答步骤.....	307
9.4.6	调整 PLC 用户程序.....	309
9.4.7	刀库刀位信息.....	310
9.4.8	PI 服务：TMMVTL.....	312

9.5	示例：装刀/卸刀.....	314
9.6	示例：更换手动刀具.....	316
9.7	车床应用示例.....	320
9.7.1	示例：车床，带转塔刀库（MAG_CONF_MPF）.....	320
9.7.2	示例：应答步骤（车床）.....	325
9.7.3	示例：车床的换刀循环.....	326
9.7.4	示例：程序段搜索和 MCALL.....	328
9.7.5	示例：车床，带副主轴.....	329
9.7.6	示例：空缓冲区测试.....	330
9.7.7	示例：将刀具从缓冲区传输到刀库.....	331
9.7.8	示例：重复“准备换刀”任务.....	331
9.8	铣床应用示例.....	333
9.8.1	示例：铣床，带链式刀库和双夹爪(MAG_CONF_MPF).....	333
9.8.2	流程图：换刀.....	338
9.8.3	示例：应答步骤（铣床）.....	348
9.8.4	示例：铣床的换刀循环.....	350
10	Easy Archive.....	353
10.1	备份和存档数据.....	353
10.2	存档中会保存哪些数据？.....	354
10.3	仅保存变更的机床数据.....	357
10.4	创建调试存档的步骤.....	358
10.5	导入调试存档的步骤.....	360
10.6	示例：数据文档“Easy Archive”（Use case）.....	361
10.7	设置串行接口参数.....	363
A	附录.....	365
A.1	文件名中的语种缩写表.....	365
A.2	缩略语列表.....	367
A.3	关于使用的第三方软件的说明.....	370
	索引.....	373

供货范围和前提条件

1.1 系统概述

和 SINAMICS S120 书本型的配置

下面展示了一个系统和 SINAMICS S120 书本型配置的典型示例:

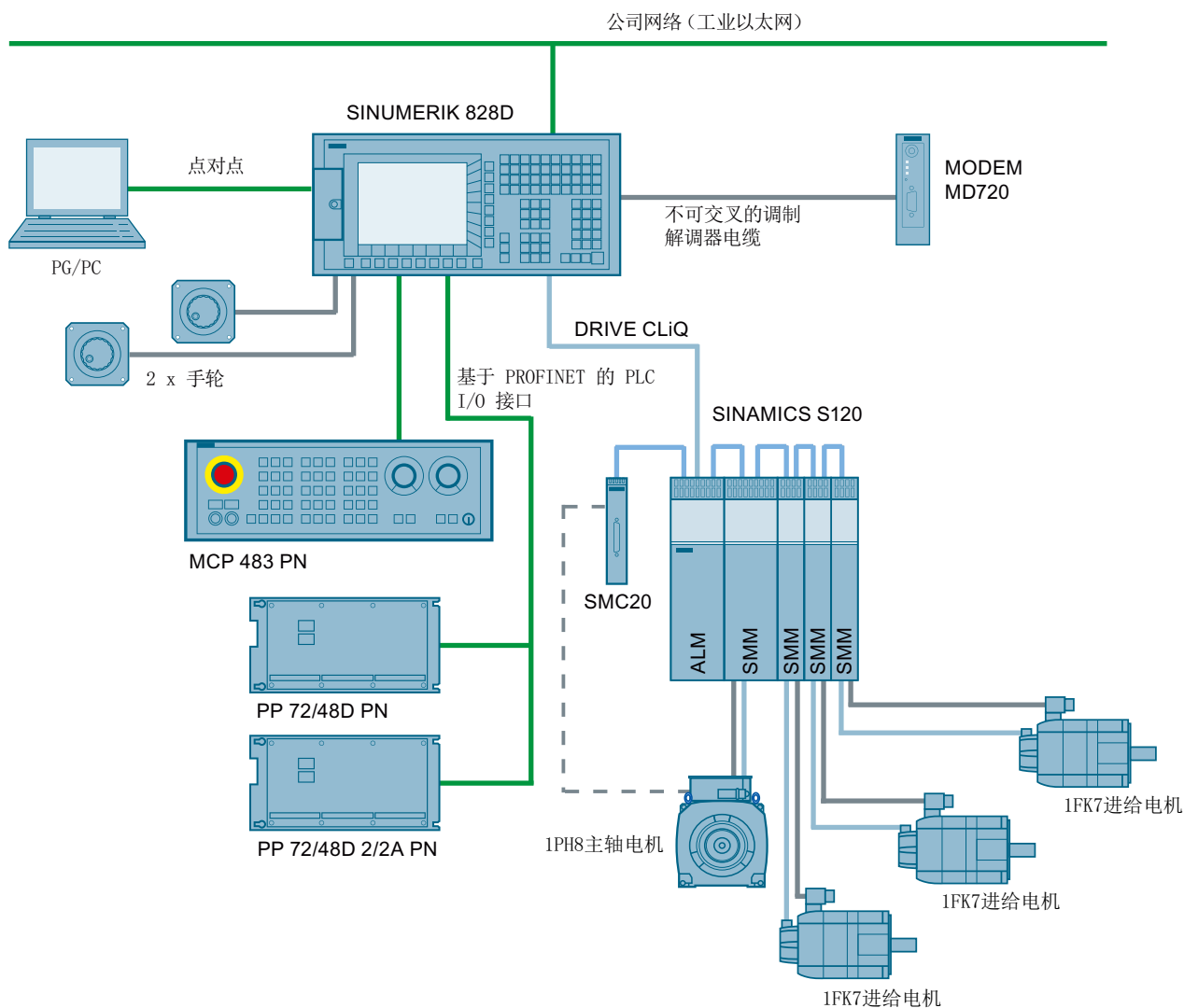


图 1-1 配置示例 1: 4 轴的基本扩展

1.1 系统概述

和 SINAMICS S120 Combi 的配置

下面展示了系统和 SINAMICS S120 Combi 配置的示例：

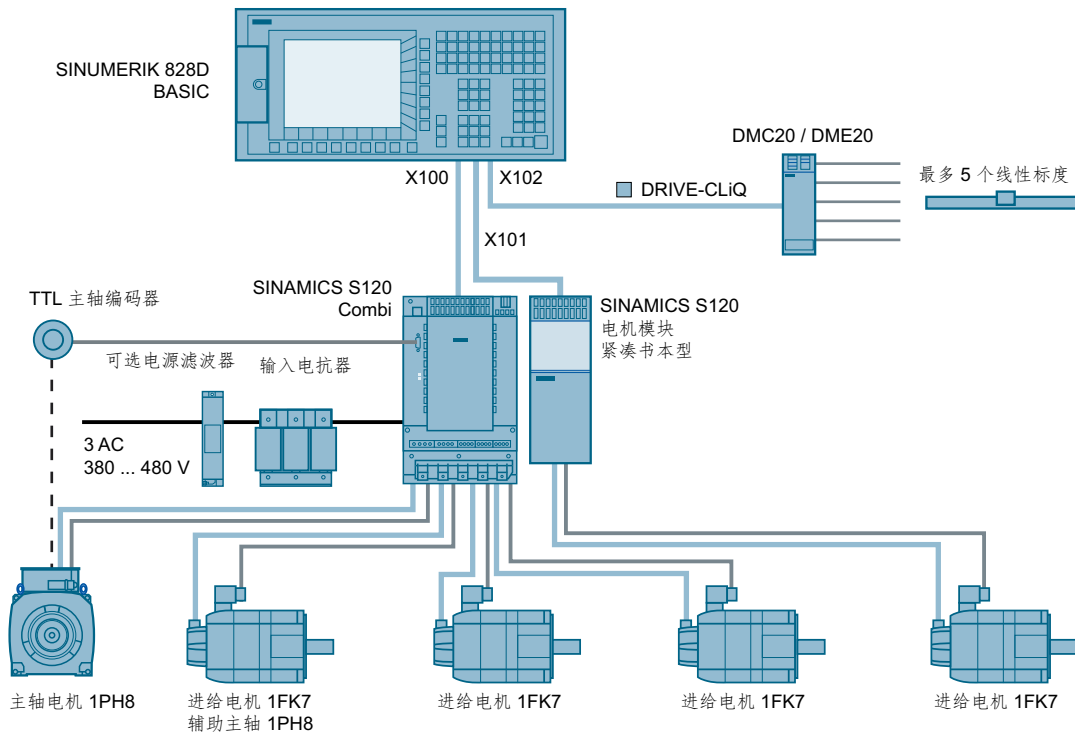


图 1-2 配置示例 2：SINUMERIK 828D BASIC 和 S120 Combi

1.2 调试工具和服务工具

Toolbox CD 光盘

SINUMERIK 828D Toolbox CD 光盘（工具箱）中包含的工具具有：

- 配置数据
- PLC 编程工具
- Access MyMachine /P2P
- 循环保护程序
- 示例
- 许可证

PLC 编程工具

对于 PLC 编程有下列工具可供使用：用于集成 PLC 的 PLC 编程工具。下文简称为“编程工具”。

SINUMERIK Integrate Access MyMachine /P2P

Access MyMachine /P2P 是一种调试工具，调试人员利用它可以非常方便地在 PC 和控制系统之间传输文件。

此时，PC 直接连接到设备前面的以太网接口上。在点对点的连接方式中，您不需要花费大量时间设置以太网接口。所有的设置都由 Access MyMachine /P2P 自动执行。

Access MyMachine /P2P 也可以通过公司网络连续访问多个 NCU。

下文简称为“Access MyMachine”。

SINUMERIK Integrate Create MyConfig

Create MyConfig 可用于机床制造商创建和执行机床自动化调试的项目。也可配置和自动执行在最终用户端升级数控系统。优点在于：

- 大大减少所需的调试或升级时间
- 通过实现过程步骤的结构化准备和自动化避免调试或升级过程中出现故障
- 自动调试或升级过程的可重复性

交付时，针对该工具的说明与软件一并保存在随附的 CD 上。

驱动软件/调试软件 STARTER

SINUMERIK 828D 的驱动调试可以借助驱动软件/调试软件 STARTER 完成。

- 一些简单的、通常由现场服务人员执行的调试步骤，例如：激活直接测量系统等，可以直接在 SINUMERIK 828D 的操作界面上执行。
- 而一些复杂的、通常在机床生产中执行的调试步骤，则可以在 SINAMICS S120 调试软件的离线模式下完成。

1.3 调试过程

1.3.1 手册编排结构

过程

下表列出的调试步骤是西门子推荐的步骤，非常有效，在 SINUMERIK 培训课程中也会介绍这些步骤。方括号的代码是培训资料中各个步骤的编号。右列指出了本手册中对该步骤进行详细描述的段落。

概述

<B060> 系统概述	调试手册 “CNC 调试” (页 9)
<B002> 调试工具和服务工具	调试手册 “CNC 调试” (页 11)

基础部分

<B061> 菜单结构	测量循环编程手册
<B021> 访问等级	调试手册 “CNC 调试” (页 37)
<B021> 日期和时间	调试手册 “CNC 调试” (页 40)
<B041> 设置机床数据	调试手册 “CNC 调试” (页 81)
<B032> SINUMERIK Integrate Access MyMachine	在线帮助
<B044> 检查和输入授权	调试手册 “CNC 调试” (页 47)
<B045> 网络配置 → 设置驱动器	操作手册，车削版/铣削版/磨削版/通用版
<B007> 电子日志	调试手册 “SINUMERIK Operate”

1.3 调试过程

<B017> 数据管理	调试手册 “CNC 调试” <ul style="list-style-type: none">• 数据级的导入与应用 (页 27)• Easy Archive (页 353)
<B006> 电气柜设计	设计手册 “电磁安装指南”

建立连接和诊断

<B010> PPU 连接的诊断 <ul style="list-style-type: none">• 系统概述• 数字量输入/输出	设备手册 “PPU 和组件”
<B030> 数字量输入/输出 → 端子布局 手轮和探头	调试手册 “CNC 调试” (页 161)
<B051> 电源模块、电机模块	SINAMICS S120 设备手册
<B053> 诊断 <ul style="list-style-type: none">• 驱动系统诊断• 诊断与服务 → 轴	调试手册 “SINUMERIK Operate”
<B008> PROFINET 连接的诊断: <ul style="list-style-type: none">• MCP 310C PN / MCP 483C PN• MCP 310 USB / MCP 483 USB• PP 72/48D PN 和 PP 72 / 48D 2/2A PN	设备手册 “PPU 和组件”

PLC

<B026> 创建 PLC 程序	在线帮助 “PLC 编程工具”
<B019> S7-200 PLC 指令	在线帮助 “PLC 编程工具”
<B033> PLC 功能	功能手册 “基本功能”(P4)
<B031> PLC 报警和信息 → 功能接口: PLC 用户功能手册 “基本功能”(P4) 报警	

<B058> PLC 板载诊断	操作手册，车削版/铣削版/磨削版/通用版
	功能手册“基本功能”(P4)
<B000> PLC 接口结构	参数手册“NC 变量和接口信号”，章节“接口信号一览”

调试流程

<B068> 调试过程 → 调试驱动	调试手册 “CNC 调试” (页 109)
<B084> 回参考点，编码器校准	功能手册“基本功能”(R1)
<B101> 驱动优化	调试手册 “CNC 调试” (页 183)
<B102> 刀具管理	调试手册 “CNC 调试” (页 263)
<B009> Easy Extend	调试手册 “CNC 调试” (页 247)
<B0 ... > Easy XML	编程手册 “Easy XML”
<B011> SINUMERIK Integrate Run MyScreens	编程手册 “SINUMERIK Integrate Run MyScreens”
<B020> 维护管理器	调试手册 “CNC 调试” (页 233)
<B018> Easy Message	操作手册，车削版/铣削版/磨削版/通用版
<B016> 安装附加轴	---
<B029> 模拟量主轴	调试手册 “CNC 调试” (页 90)

1.3 调试过程

前提条件

本手册中介绍的调试步骤以下述条件为前提：

- 设备的机械和电气安装必须已结束。
- 目测设备的下列项目：
 - 机械安装是否正确、电气连接是否稳固
 - 电源是否接好
 - 屏蔽层和接地是否接好
- 接通控制系统，在“Normal startup”的模式中启动系统：
显示操作软件的初始画面后，表明控制系统启动完成。

资料

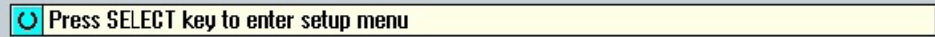
关于各个循环的调试细节请参考 SINUMERIK Operate 828D 调试手册，章节“配置循环”。

1.4 启动控制系统

启动控制系统

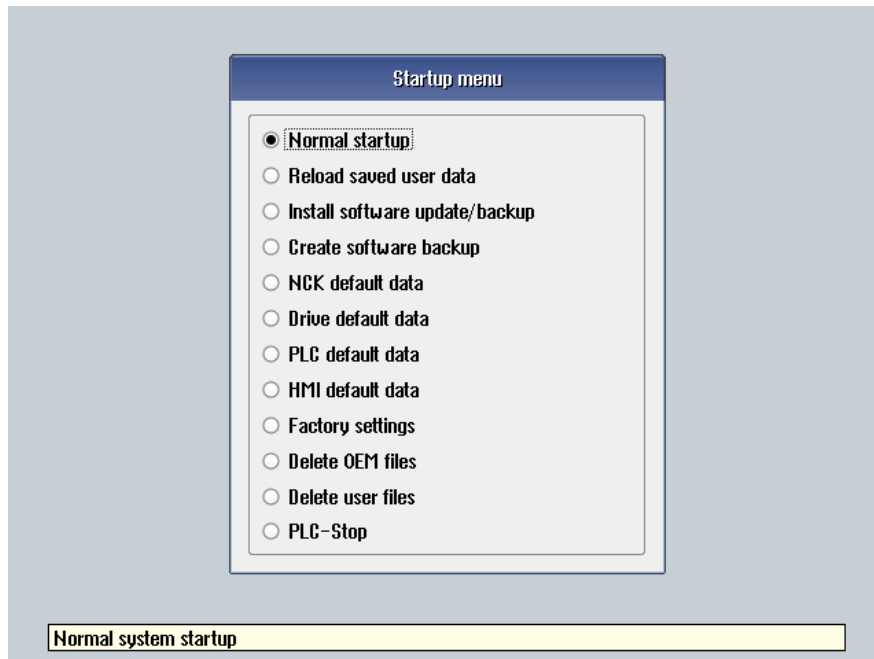
操作步骤:

1. 控制系统上电启动后会出现以下提示:



Press SELECT key to enter setup menu

2. 在 3 秒钟内，按下<SELECT>键。
3. 然后依次按下：
菜单回调键，**HSK2**（水平软键 2），**VSK2**（垂直软键 2）
4. “Startup menu” 显示在屏幕上，“Normal startup” 被选为缺省设置。



控制系统的启动方式

选择	功能
Normal startup	系统正常启动。
Reload saved user data	系统从系统 CF 卡上载入其中保存的用户数据（相当于软键“数据存储”的功能）。
Install software update/ backup	系统从用户 CF 卡或 U 盘上安装系统 CF 卡的升级程序。
Create software backup	系统将系统 CF 卡的数据备份到用户 CF 卡或 U 盘上。

1.4 启动控制系统

选择	功能
NCK default data	系统载入出厂时设置的西门子 NCK 数据，并删除 PLC 上的永久数据。
Drive default data	系统删除 SINAMICS 用户数据。
PLC default data	系统清零 PLC，载入缺省 NOP PLC 程序。
HMI default data	有两个方案： No [方案 1]/ Yes [方案 2] <ul style="list-style-type: none"> ● 方案 1: 系统删除目录“/user”下的用户数据。 ● 方案 2: 和方案 1 相同，另外还： 删除目录“/oem”和“/addon”中的数据，除了日志数据。
Factory settings	有两个方案： No [方案 1]/ Yes [方案 2] <ul style="list-style-type: none"> ● 方案 1: 系统删除 SINAMICS 用户数据。 载入西门子 NCK 缺省数据。 系统清零 PLC，载入缺省 NOP PLC 程序。 并保存目录“/user”下的用户数据。 ● 方案 2: 和方案 1 相同，另外还： 删除目录“/oem”和“/addon”中的数据，除了日志数据。
Delete OEM files	其他所有机床制造商数据都会被删除。
Delete user files	
PLC stop	系统停止 PLC。

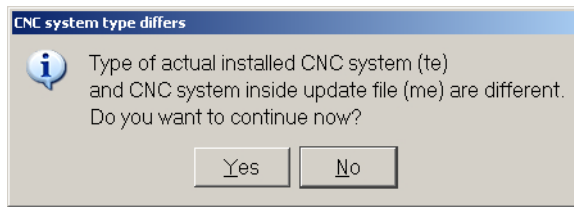
选择磨削工艺

控制系统启动时，启动菜单会提供选项，可以选择载入外圆磨削或者平面磨削配置数据。

如果要切换磨削工艺，则需要在启动菜单中再次选择磨削工艺。之后会载入标准数据以及所需的工艺。许可证和选项保持不变。

硬件检测

如果使用硬件 PPU 2xy.3 会在启动时检测工艺。系统会根据硬件信息块中找到的产品编号检测硬件型号，并将结果和当前的 NC 性能等级进行比较。如果系统发现有和该硬件不配套的软件，会发出以下报告：



说明

更换不同 PPU 的系统 CF 卡

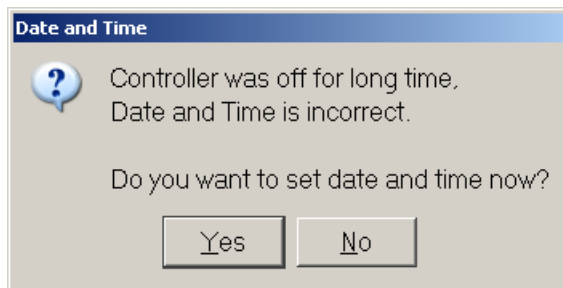
SINUMERIK 828D 的系统 CF 卡和 SRAM 在数据管理方面相互关联，因此应把系统 CF 卡看成一个焊住的 EEPROM，不能擅自更换！

如果不得不更换系统 CF 卡，系统在启动时会根据保存的序列号检测出这一情况。

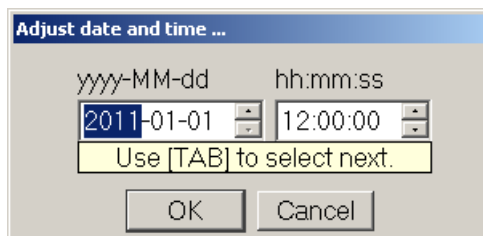
此时，如果在此之前已使用软键“数据存储”手动备份了数据，系统会载入备份数据。如果没有找到备份数据，系统会自动以“NCK default data”的模式启动。

空的 RTC 电容

RTC 电容为空时，系统启动时会显示以下信息：



然后可以设置时间和日期：



在控制系统成功启动后，RTC 电容再次充电。

1.5 建立和控制系统的通讯

建立连接

控制系统和 PG/PC 通过一条 Ethernet 电缆连接。控制系统上有以下 Ethernet 接口可使用：

- **X127**（在正面盖板后）

电缆类型：交叉 Ethernet 电缆

在 X127 接口上，控制系统是缺省的 DHCP 服务器，它为直接连接法（对等网络）提供 IP 地址 192.168.215.1。

- **X130**（背面）：

电缆类型：非交叉 Ethernet 电缆

X130 接口是连接到公司网络上的接口。此处 PG/PC 作为 DHCP 客户端获得的 IP 地址由 DHCP 服务器从公司网络确定，或由手动输入的固定 IP 地址确定。

1.5.1 通过 X130 建立和控制系统的通讯

连接到公司网络上

NCU 通过 Ethernet 接口 X130 连接到公司网络中。通过公司网络可以访问网络驱动器。

在操作区“诊断”中，按下菜单扩展键，选择软键“TCP/IP 总线”→“TCP/IP 诊断”→“详细”，设置 X130 通讯所需的参数。

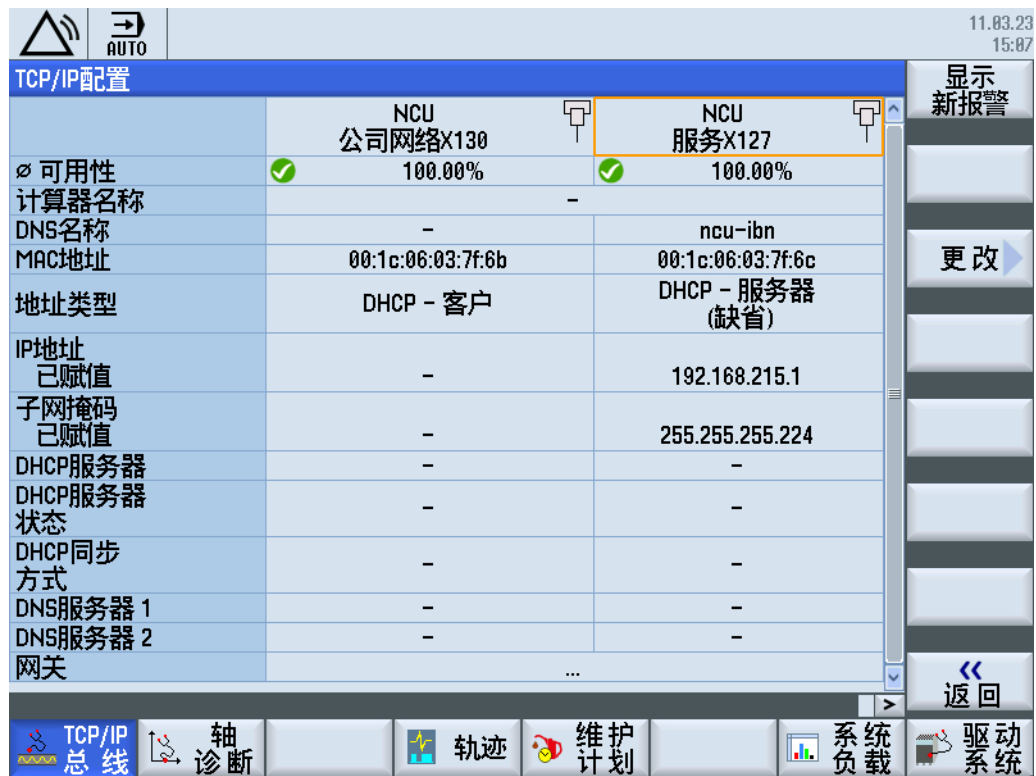







图 1-3 网络设置

公司网络 X130

-  白色 网线已插入
-  红色 网线未插入

可用性

可用性指测出的错误数据占总数据量的比例。公司网络中出现的故障，例如：无法访问逻辑驱动器，IP 地址重复等，以及启动中的起振时间可能会影响可用性的稳定性：

-  绿色 大于 95 %
-  黄色 50 - 95 %
-  红色 小于 50 %

说明

所有不存在的信息由表格项中的连字符“-”表明。

1.5.2 通过编程工具建立和控制系统的通讯的步骤

在 PLC 编程工具中建立通讯接口

操作步骤：

1. 通过下面的快捷方式或“开始”菜单启动 PLC 编程工具：



2. 点击浏览栏中的“通讯”按钮，或选择菜单指令“检视”→“通讯”。
3. 在左侧的“通讯参数”下输入 X127 的 IP 地址 192.168.215.1。
4. 双击右上方的按钮“TCP/IP”。
5. 在打开的“PG/PC 接口”对话框中选择 PG/PC 的 TCP/IP 协议。通常是 PC 的网卡。

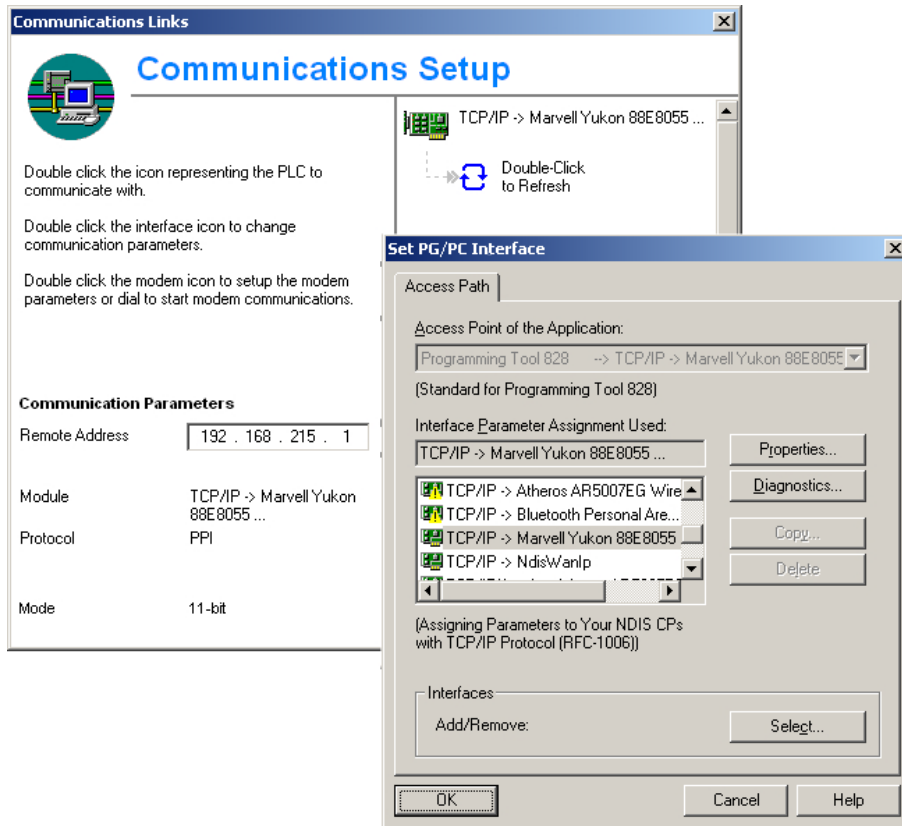


图 1-4 TCP/IP 通讯设置

6. 按下“确认”。

7. 双击按钮“双击刷新”，建立连接。成功建立连接后，按钮周围会出现绿色边框：

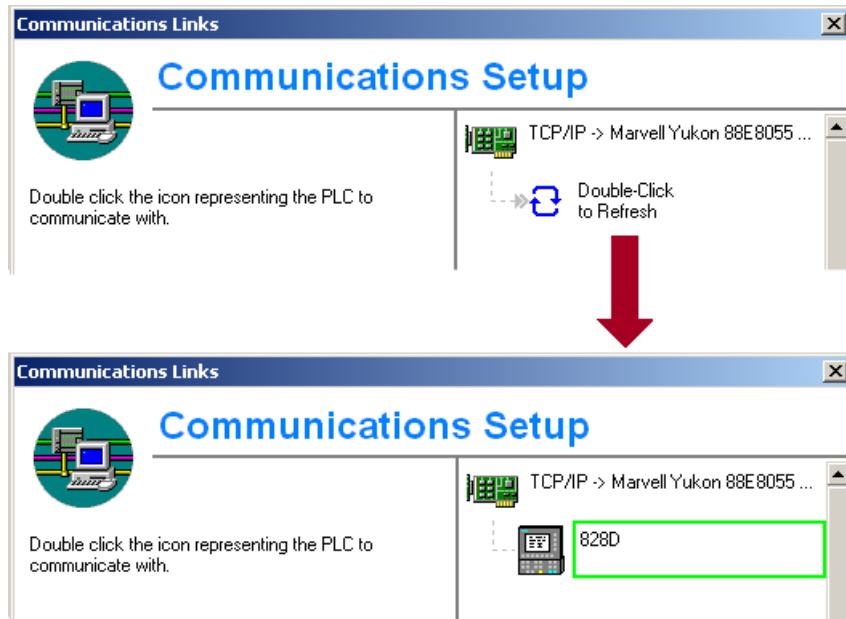


图 1-5 在线连接

如果建立连接失败，请检查“网络连接”→“本地连接”下的设置。

8. 按下“确认”，重复第 7 步。

1.5.3 这样通过 Access MyMachine 与控制系统进行通讯




连接方法

在“Access MyMachine”中，有下列方法可以连接控制系统：

- 直接连接（对等网络）
- 网络连接

连接的当前状态会显示在 Access MyMachine 的下方状态栏中。

各个按钮的含义：

- | | |
|---|------|
|  | 连接 |
|  | 断开 |
|  | 远程操作 |

说明

一般只允许**一条连接**，也就是说，不能同时连接到多个控制系统：这样可以防止通过“Access MyMachine”交换两个控制系统之间的数据。

直接连接

建立直接连接的步骤：

1. 在对话框“Settings” → “Connetion” → “Direct connection” 中输入登录数据：

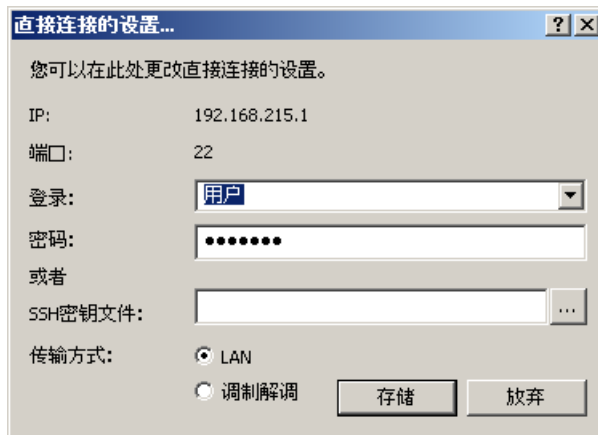


图 1-6 对话框：输入直接连接的登录数据

2. 选择菜单“Connection” → “Connect” → “Direct connection”，或者点击按钮“Connect”。屏幕上显示以下对话框：



图 1-7 对话框：直接连接

3. 上次选择的直接连接被选中。按下按钮“Connect”，建立和 IP 地址 196.168.215.1 的连接。通过菜单选择直接连接时，不会显示该对话框。

网络连接

建立网络连接的步骤:

1. 选择菜单“Settings” → “Connection” → “Direct connection”，或者点击按钮“Connect”。

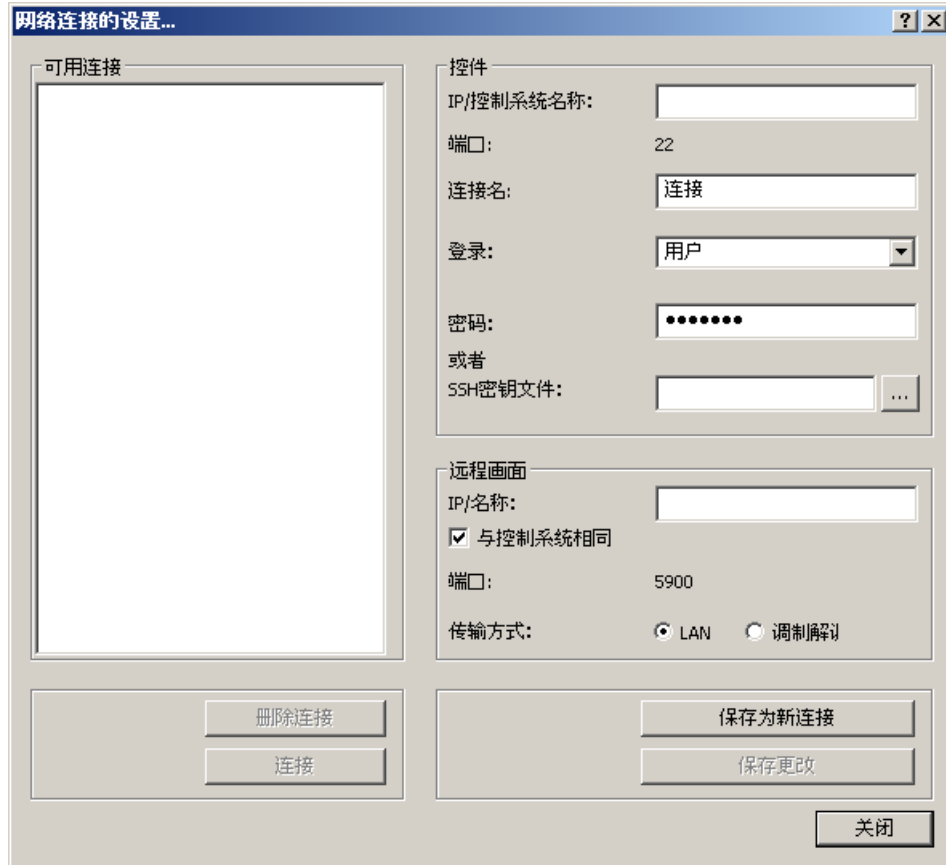


图 1-8 对话框：网络连接

2. 选择菜单“Connection” → “Connect” → “Network connection”，或者选择上次使用的连接。
3. 连接到设置的控制系统。

说明

SSH-Key File

除了密码外，用户还可以使用 SSH 码通过验证。详细信息请参见“在线帮助”。

1.5 建立和控制系统的通讯

数据级的导入与应用

目标

数据级这一概念是为了明确区分系统数据（软件安装时产生的数据）、OEM 数据（样机调试时产生的数据）以及用户数据（用户在使用时产生的数据）。OEM 数据又可细分为适用于一个机床系列的数据（同一系列所有机床上的数据都一致）和只适用于特定机床的个性化数据。

在创建存档时您需要选择数据级，即选择一部分数据进行存档：是 OEM 数据、机床特定数据还是用户数据。这样可以方便调试和设备升级。

2.1 NCK 中的数据级

归类

与控制系统有关的所有数据都会被划分到数据级中，为了方便日后要进行的调试、软件升级、软件更新以及部件更换。这些数据会被划分为 4 个数据级：M（制造商）、I（个性化）、U（用户）以及 S（系统）

系统（S）数据级

此数据级包括了西门子目录下的数据以及 CF 卡上的系统数据，并且在其他文件中以“S”标识。

S 数据级的标识/属性

此数据会在首次系统开启或者在初始化时自动加载，并进行激活。

“S”级数据不能进行保存，并以“只读”标识。

数据类型	示例
HMI:	西门子标准操作界面及其语言扩展
NCK:	西门子提供的标准循环、测量循环以及 JobShop 循环，SGUD 和 SMAC 定义以及其他所有 S 数据级的 NCK 有效数据。
PLC:	硬件配置
驱动:	编码与数据（数据说明、报警说明、宏）

制造商（M）数据级

此数据级包括了，机床制造商(OEM)在同一系列机床首次调试时得出的全部数据，并在其他文件中以“M”标识。

M 数据级的标识/属性

M 级的数据在一个单独的存档“制造商”中储存。

数据类型	示例
HMI:	OEM 对话框（Easy Screen）和循环支持、报警和信息文本。
NCK:	制造商循环，MGUD 和 MMAC 定义以及 M 数据级的全部 NC 有效数据（无标准机床数据的存储器）

数据类型	示例
PLC:	PLC 程序, 此数据级数据块的内容
驱动:	全部

个性化 (I) 数据级

此数据级包括的数据为, 针对特定设备的, 在调试时由 OEM 或此后由销售商生成的数据。此数据级在其他文件中以“I”标识。

I 数据级的标识/属性

I 级数据在一个单独的存档“INDIVIDUAL”中储存。

数据类型	示例
HMI:	Easy Extend 与 Service Planner
NCK:	参考点尺寸、间隙补偿、补偿数据、刀架数据、I 数据级的全部 NC 有效数据
PLC:	此数据级的数据块内容
驱动:	目前未设置数据。

用户 (U) 数据级

此数据级包括了所有用户数据以及在机床运行时产生的数据, 例如维护间隔时间。此数据级在其他文件中以“U”标识。

U 数据级的标识/属性

U 级的数据在一个单独的存档“用户”中储存。

数据类型	示例
HMI:	---
NCK:	刀具数据、设定数据、零件程序、子程序和用户循环, UGUD 与 UMAC 的定义, 但无程序代码 (例如 PLC)
PLC:	此数据级的数据块内容
驱动:	目前未设置数据。

GUD 与宏

按下表进行数据级定义文件（GUD、宏）的归类：

定义文件	数据级
GUD4	个性化 (I)
MGUD	制造商 (M)
MMAC	制造商 (M)
UGUD	用户 (U)
UMAC	用户 (U)

数据级属性

数据级属性是隐含设置的。数据级的默认属性可以进行显式更改。

- 通常从有效文件系统中导出的数据，会根据其属性被划分到一个**数据级**的特定文件中。
示例：补偿数据（CEC、EEC 或 QEC 导出的数据）被划分到“个性化”数据级。如果数据由定义产生（GUD 和宏），那么该数据会自动进入到定义文件中的数据级。
UGUD.DEF 中的 GUD 划分到“用户”数据级。

- 数据定义和数据内容要进入**不同的数据级**，那么必须在数据定义中对其进行说明。
示例：对测量头进行说明的 GUD 定义应保存到“制造商”数据级中，因为其对于制造商循环的执行是必需的。

因机床与机床的测量头型号可能有区别，所有数据的数值要划分到“个性化”数据级。此外，要输入附加的变量属性“DC”（DataClass）：

```
MGUD.DEF （数据级“制造商”）
DEF CHAN DCI INT CALIPER
```

数据内容的数据级定义，只能比数据定义的数据级“低或相同”。对于其他不同的设置，会发出报警。

此外，采用以下数据级等级：**M > I > U**

2.2 PLC 中的数据级

PLC 中的数据级

程序模块的数据级划分用于 SINUMERIK 控制系统的升级或调试。分为三个数据级：

- **制造商 (M)** 由机床制造商在设计时分配。
- **个性化 (I)** 由机床制造商在首次调试时分配（这一具体机床的数据）。
- **用户 (U)** 由最终用户设置（各个过程的数据）。不允许把主程序和子程序划分为不同的数据级。

模块	缺省设置	可修改
主程序 (OB1)	MANUFACTURER	否
子程序 (SBR、INT ...)	MANUFACTURER	否
数据块	INDIVIDUAL MANUFACTURER USER	是

示例

在“属性”对话框中为模块分配数据级。

1. 通过菜单“检视”→“数据块”来选择一个数据块。
2. 在菜单“检视”中选择“属性”，修改数据级。

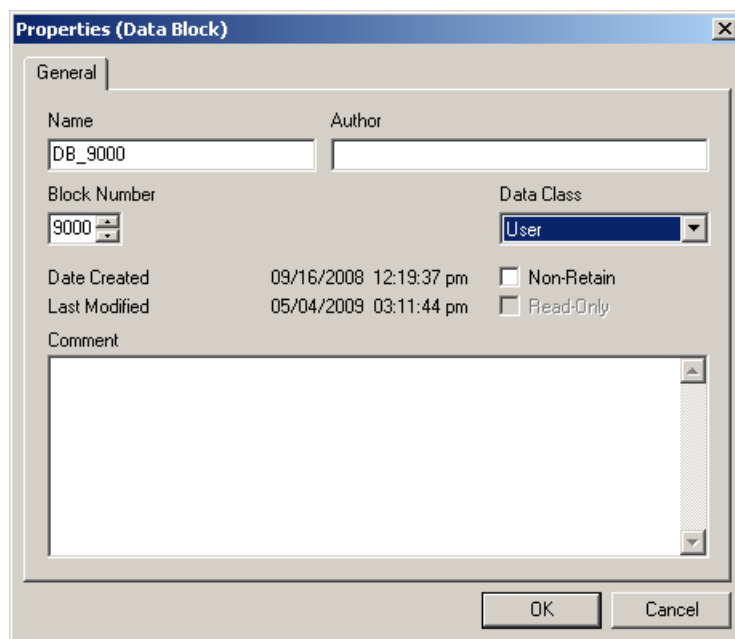


图 2-1 选择数据级

2.3 操作软件中的数据级

操作软件中的数据级显示

数据级基本上已由 CF 卡的目录结构决定：

目录	数据级
system	SYSTEM
siemens	
addon	MANUFACTURER
oem	MANUFACTURER, INDIVIDUAL
individual	MANUFACTURER, INDIVIDUAL
user	USER

目录“system”和“siemens”在存档时没有含义，因为其是在安装 SINUMERIK 软件时建立的并且不能通过安装和配置以及在以后的使用中进行修改。系统更新或升级通常是在这些目录下进行的。这些目录下的存档因此只能以回滚（rollback）备份方式在后台进行。

这些目录再分为 SINUMERIK NCK / PLC / HMI 和 SINAMICS 涵盖了上述的数据区并处于所有已提到的目录（数据级）中。

CF 卡	数据区
SINUMERIK	NCK
	PLC
	HMI
SINAMICS	驱动

CF 卡的结构

文件系统的最高一级目录由“addon”、“individual”、“oem”、“siemens”和“user”组成。这些目录的结构基本上相同。

以下为配置相关的目录结构部分：

siemens 目录	
/siemens/sinumerik	
	/hmi

2.3 操作软件中的数据级

siemens 目录		
	/appl	应用（操作区）
	/base	基础系统组件
	/cfg	全部配置文件
	/data	版本数据
	/hlp	在线帮助文件
	/hlps	打包在线帮助文件和版本文件
	/ico	图标文件
	/ico640	分辨率 640x480 的图标
	/ico800	分辨率 800x600 的图标
	/ico1024	分辨率 1024x768 的图标
	/ico1280	分辨率 1280x1024 的图标
	/ico1600	分辨率 1600x1240 的图标
	/lng	文本文件
	/lngs	打包文本文件和版本文件
	/osal	
	/ace	ACE/TAO
	/qt	Qt
	/proj	Easy Screen 配置
	/template	各种模板
	/cfg	配置文件的模板
	/lng	文本文件的模板
	/tmpp	临时数据存储位置

addon 目录		
/addon/sinumerik		
	/hmi	
	/appl	应用（操作区）
	/cfg	配置文件
	/data	版本数据
	/hlp	打包在线帮助文件和版本文件
	/ico	图标文件

addon 目录		
	/ico640	分辨率 640x480 的图标
	/ico800	分辨率 800x600 的图标
	/ico1024	分辨率 1024x768 的图标
	/ico1280	分辨率 1280x1024 的图标
	/ico1600	分辨率 1600x1240 的图标
	/lng	文本文件
	/lngs	打包文本文件和版本文件
	/proj	Easy Screen 配置
	/template	不同的模板

oem 目录		
/oem/sinumerik		
	/data	版本数据
	/archive	制造商文档
	/hmi	
	/appl	应用（操作区）
	/cfg	配置文件
	/data	版本数据
	/hlp	在线帮助文件
	/hlps	打包在线帮助文件和版本文件
	/ico	
	/ico640	分辨率 640x480 的图标
	/ico800	分辨率 800x600 的图标
	/ico1024	分辨率 1024x768 的图标
	/ico1280	分辨率 1280x1024 的图标
	/ico1600	分辨率 1600x1240 的图标
	/lng	文本文件
	/lngs	打包文本文件和版本文件
	/proj	Easy Screen 配置
	/template	不同的模板

2.3 操作软件中的数据级

user 目录		
/user/sinumerik		
	/data	版本数据
	/archive	用户专用文档
	/prog	用户专用程序
	/hmi	
	/cfg	配置文件
	/data	版本数据
	/hlp	在线帮助文件
	/ico	图标文件
	/ico640	分辨率 640x480 的图标
	/ico800	分辨率 800x600 的图标
	/ico1024	分辨率 1024x768 的图标
	/ico1280	分辨率 1280x1024 的图标
	/ico1600	分辨率 1600x1240 的图标
	/lng	文本文件
	/log	日志文件
	/md	机床数据视图
	/proj	Easy Screen 配置

操作软件的设置

3.1 存取级别

访问功能和机床数据

这样便可以控制对各个功能和数据区域的读写。有 0 至 7 个存取级别，0 表示最高等级，7 表示最低等级。存取级别 0 至 3 通过口令锁定，4 至 7 通过钥匙开关位置锁定。

存取级别	锁定方法	范围	数据级
1	口令: SUNRISE	制造商	Manufacturer (M)
2	口令: EVENING	服务	Individual (I)
3	口令: CUSTOMER	用户	User (U)
4	钥匙开关位置 3	程序员, 调试员	User (U)
5	钥匙开关位置 2	合格的操作员	User (U)
6	钥匙开关位置 1	受过培训的操作员	User (U)
7	钥匙开关位置 0	学过相关内容的操作员	User (U)

口令一直保持如此设置，直至用软键“删除口令”复位为止。口令可在激活后更改。

如果忘记口令，必须重新初始化，即以“NCK default data”模式启动。此时所有的口令都恢复为缺省设置（见上表）。上电不会复位口令。

注意

更改口令缺省设置

滥用口令可能存在很大的安全风险。

对于简单调试，默认采用记录的口令。建议在循环加工过程中定期更改口令。

钥匙开关

存取级别 4-7 要求在机床控制面板上有一个相应的钥匙开关位置。因此有三个不同颜色的钥匙开关。每个钥匙仅可许可特定的区域。

3.1 存取级别

各个钥匙开关位置的含义：

存取级别	开关位置	钥匙颜色
4-7	0 至 3	红色
5-7	0 至 2	绿色
6-7	0 和 1	黑色
7	0 = 插拔位置	未插入钥匙

钥匙开关位置始终须由 PLC 用户程序编辑，并连接到相应的接口上。

文档

口令也可由 PLC 删除 → PI 服务：LOGOUT

功能手册 基本功能，章节“SINUMERIK 828D 的 PLC”(P4)

3.2 设置和修改口令的步骤

设定口令

需要切换存取级别时，首先选择操作区“启动”。

1. 按下“口令”软键。
2. 按下“设定口令”软键，打开下面的对话框：

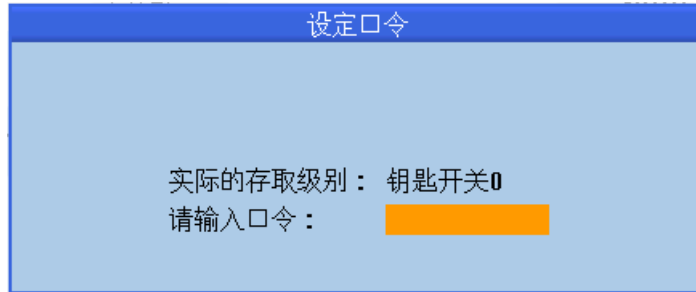


图 3-1 设定口令

3. 输入一条口令，按下“确认”结束输入，或按下<Input>键。
有效口令会置入系统，并显示当前有效的存取级别。无效口令被拒绝。
4. 如果需要为低于当前存取级别的访问级设置口令，必须首先删除原先的口令。
按下“删除口令”软键后，上次生效的口令被删除。当前的钥匙开关位置随后生效。

修改口令

修改口令的步骤为：

1. 按下“修改口令”软键，打开下面的对话框：

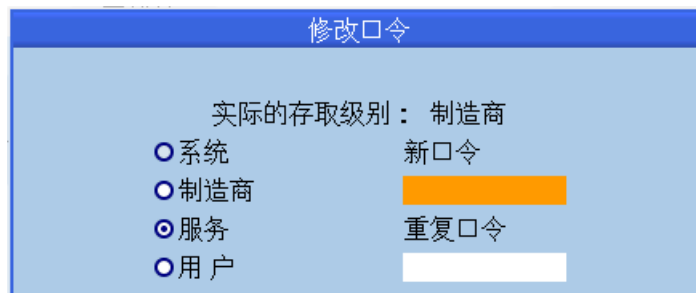


图 3-2 修改口令

2. 在两个输入栏中输入新口令，然后按下“确认”。如果这两个口令一致，新口令便被视为有效口令，置入系统中。

3.3 设置日期和时间的步骤

前提条件

只有在具有充足的访问权限（从“用户”起）时，才可修改日期/时间。

设置日期和时间

操作步骤：

1. 选择操作区域“调试”。
2. 按下“HMI”软键。
3. 按下“日期时间”软键。
打开窗口“日期/时间”。
4. 在“格式”栏中选择需要的日期和时间的显示格式。
5. 按下软键“确认”结束输入。
新的日期或时间被接收并在“当前”栏的第一行中显示。

说明

时间的同步

SINAMICS 驱动系统没有实时时钟。SINAMICS 时钟显示的时间和日期按照 10 秒的周期和 SINUMERIK 的实时时钟同步。

这种同步方式导致只有 SINUMERIK 的时间和/或日期的修改经过 10 秒后，SINAMICS 驱动系统的时间和/或日期才会更新。

如果在这 10 秒的间隔内出现了 SINAMICS 报警（报警号 200000 - 299999），该 SINAMICS 报警获得的是没有更新的时间/日期。与此相反，SINAMICS 报警引发的 SINUMERIK 报警（报警号 < 200000 或 > 300000）则已经获得了新的时间/日期。

3.4 设置操作软件的语言

3.4.1 提供的标准系统语言和扩展语言

系统语言

在标准供货范围中，SINUMERIK 828D 具有以下系统语言：

- 英语
- 法语
- 德语
- 意大利语
- 韩语
- 葡萄牙语（巴西）
- 简体中文
- 西班牙语
- 繁体中文

在出厂时，SINUMERIK 828D 已安装了所有系统语言，这样便可以直接在操作界面上切换语言，无需再次载入系统语言数据。

其他语言

其他语言以 DVD 光盘的形式提供。DVD 光盘上有针对软件版本的安装说明及兼容性说明。

注意

升级时的其他语言

系统软件升级后仅保留与当前软件版本相兼容的系统语言和其他语言。系统会自动删除不再兼容的其他语言。

3.4 设置操作软件的语言

3.4.2 删除其他语言

前提条件

通常只能删除用户后来安装的其他语言，而不能删除系统语言。

删除语言

操作步骤：

1. 选择操作区域中的“调试”，然后按下软键“修改语言”。
2. “语言选择”窗口打开。有效的语言被选中。
3. 选择要删除的语言。
4. 按下软键“删除语言”。
5. 按下“确认”。

3.4.3 使用输入法编辑器输入亚洲字符。

输入法编辑器 (IME)

在程序编辑器和 PLC 报警文本编辑器中可以编辑亚洲字符。对于以下亚洲国家语言，可以使用输入法编辑器：

- 简体中文
- 繁体中文（台湾）
- 韩语

通过组合按键<Alt>+<S> 开启编辑器。

带有主动学习功能的编辑器

对于简体中文和繁体中文，系统提供了字典：

- 在操作系统上编辑字典
- 在操作系统上导入字典

如果输入的音标与操作系统中所储存的数据没有符合项时，编辑器就会打开学习功能。这个功能可以将音节和单词进行匹配，在保存后可以一直使用。编辑器将汉字排列在拼音旁边显示。在完整地显示出来后，使用 <Input> 键保存汉字。同时，也输入到了相应的输入栏中。

编辑字典

如果激活了该功能，则会另外显示由字符及其拼音的组成的一行文本。编辑器根据此发音提供不同的字符，键入相应的数字 (1 ... 9) 可以选择所需的字符。按下<TAB>键可以在拼音栏和拼音输入栏之间切换光标。



如果光标位于上一栏，则操作员可以按下<Backspace>键取消显示的字符组合。



按下<Select>键则保存显示的字符。



按下<Delete>键则从字典删除显示的字符组。

导入字典

通过任何一个 **Unicode** 编辑器都可以制作字典，来添加带有拼音的汉字。如果一个发音包含多个汉字，那么该行不允许再包含其它配对。如果一个发音有多种配对，那么要分行进行录入。否则，每行中会出现多个汉字。

创建的文件为 **UTF8** 格式，使用文件名 **chs_user.txt**（简体中文）或 **cht_user.txt**（繁体中文）进行存储。

行结构：

拼音 <TAB> 汉字 <LF>

或

拼音 <TAB> 汉字 1<TAB> 汉字 2 <TAB> ... <LF>

<TAB> - 制表符

<LF> - 换行

将制作的字典保存在下面其中一个路径下：

```
../user/sinumerik/hmi/ime/
```

```
../oem/sinumerik/hmi/ime/
```

在下次开启中文编辑器时，字典的内容就会添加到系统字典中。

示例：

3.4 设置操作软件的语言

ai	哎	哀	唉	埃	挨
caise	彩色				
hongse	红色				
huise	灰色				
heli	河裹				
zuihaowan	最好玩				

说明

选择了“HMI 数据”时，也可将字典保存在调试存档中（另见：存档中会保存哪些数据？(页 354)）

3.4.4 输入汉字的步骤。

输入汉字

通过音标（拼音）选择汉字： 音标由拉丁字母组成。

1. 输入由拉丁字母组成的音标。
编辑器显示一系列该拼音对应的汉字。
2. 使用光标键选择需要的汉字。

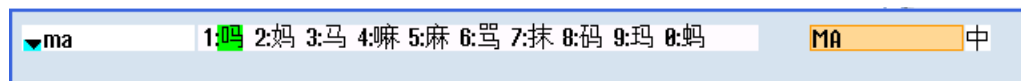


图 3-3 示例：简体中文

通过<SELECT> 键把选择区域转换为拉丁字符输入，就会回到打开中文编辑器之前所在的输入栏处。

繁体汉字的注音输入

此外，对于繁体汉字可以使用以下输入方法：

1. 为建立单个音节，可以使用键盘的数字键区。
每个数字都对应了几个注音符号，可以通过一次或多次点击数字键进行选择。
2. 可以通过使用<INPUT> 键或输入另外的数字来确认注音输入栏中所显示的选择。

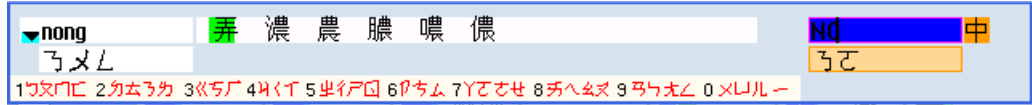


图 3-4 示例：注音方法

3.4.5 输入韩语字符的步骤

通过矩阵输入字符

如果只能使用控制系统上的键盘，就要采用矩阵输入，这只需要使用数字键就可以：

1. 通过第一次敲数字键在行中进行选择：所选中的行颜色会突出显示。
2. 通过第二次敲数字键在列中进行选择：字符短时高亮显示并接收到“字符”栏中。

通过<SELECT>键可以在韩语和拉丁字母输入之间进行切换。

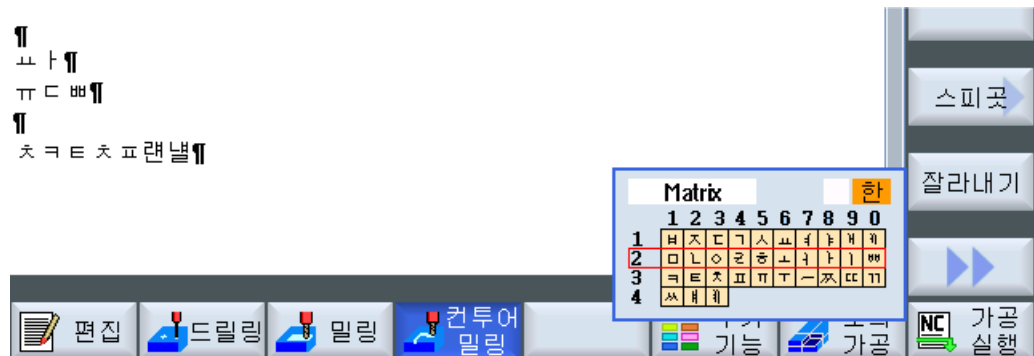


图 3-5 示例：程序编辑器

可以在“Matrix”和“Beolsik”之间切换输入模式。

3.4 设置操作软件的语言

使用韩语键盘

需要输入韩语字符时，操作员需要使用以下布局的键盘。此键盘的布局和一个英文 QWERTY 键盘（标准的传统键盘）类似，其中包含的事件必须归结为音节。

	1	2	3	4	5	6	7	8	9	0			Backspace ←
Tab ↔	Ⓜ Q ㅊ	Ⓜ W ㅌ	Ⓜ E ㄷ	Ⓜ R ㄹ	Ⓜ T ㅅ	Ⓜ Y ㅕ	Ⓜ U ㅇ	Ⓜ I ㅑ	Ⓜ O ㅓ	Ⓜ P ㅗ			Enter ↵
Caps Lock	A ㅇ	S ㄴ	D ㅇ	F ㅇ	G ㅇ	H ㅇ	J ㅇ	K ㅇ	L ㅇ				
↑		Z ㅇ	X ㅇ	C ㅇ	V ㅇ	B ㅇ	N ㅇ	M ㅇ				↑	
Ctrl		Alt											Ctrl

示例： 韩语键盘

韩语字母(Hangeul)由 24 个字母组成： 14 个辅音和 10 个元音。 元音和辅音构成一个音节。

3.5 检查和输入许可证

需授权的选件

在使用 SINUMERIK 控制系统上安装的系统软件和激活的选件前，需要首先将购买的许可证分配给硬件。在分配期间，系统软件和选件的许可证号以及硬件序列号会生成一个 **License Key**（许可证密钥）。此时会通过互联网访问由西门子管理的许可证数据库。接着将包含有许可证密钥的许可信息传输到硬件上。

许可证数据库的访问由 **Web License Manager**（网络许可证管理器）执行。

网络许可证管理器

通过网络许可证管理器可以在一个标准网络浏览器中将许可证分配给硬件。若要结束分配，必须通过操作界面手动将许可证密钥输入到控制系统上。

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

说明

SINUMERIK 软件产品

如果某个 SINUMERIK 软件产品没有许可证密钥，或没有有效的许可证密钥，控制系统便发出报警 8081，并且不能输出 NC-START。

激活测试许可证

如果要使用没有授权的选件，可以通过测试许可证在短时间内激活选件，以便使用。仅在存取级别大于或等于 2“服务”时，才可激活。测试许可证允许的时间段为 150 个小时且可由机床制造商用于调试和测试。

另见

许可证管理的相关定义 (页 50)

3.5.1 输入许可证密码的步骤

前提条件

使用激活的选件需要相应的许可证。在通过网络许可证管理器给选件授权后，您会获得一个许可证密钥，它包含了所有需要获得授权的选件，并且只作用于系统 CF 卡。

只有存取级别等于或大于“制造商”时，才可以设置或复位选件。

输入或读入许可证密钥

操作步骤：

1. 选择操作区域“调试”。
2. 按下菜单扩展键。
3. 按下“许可证”软键。
4. “授权”窗口打开
5. 选择输入行，输入新的许可证密钥。

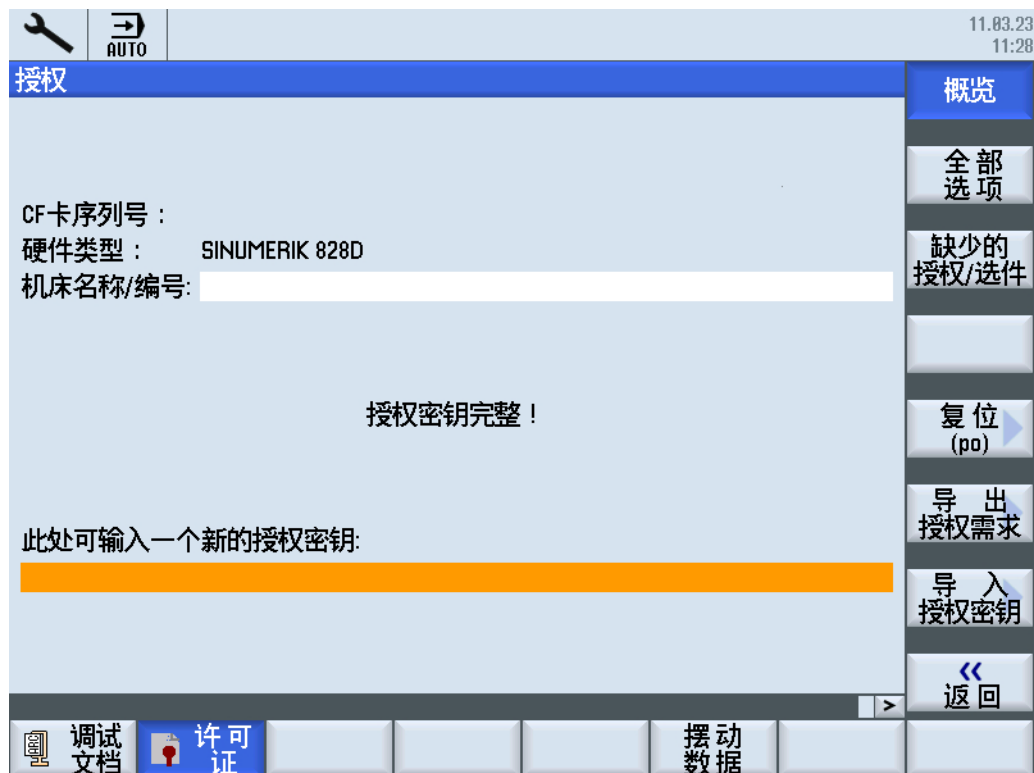


图 3-6 授权

其它操作：

- 软键“全部选项”：显示所有选项。
- 软键“缺少的许可证/选件”：显示缺少的许可证和选件：检查缺少的许可证/选件的步骤 (页 49)
- 软键：“导出许可需求”：将缺少的许可导出到一个文件中并可以保存到存储设备上。
- 软键：“读入许可证密钥”：从许可证文件中读入许可证密钥。
该文件可随产品一起交付，或通过 Web License Manager (<http://www.siemens.com/automation/license>)获取。

3.5.2 检查缺少的许可证/选件的步骤

确定授权需求

操作步骤：

1. 按下软键“全部选件”，列出控制系统可以选用的全部选件。
2. 在“已设置”栏中激活或取消激活所需的选件：
 - 勾选复选框
 - 输入选件数量

3.5 检查和输入许可证

3. 点击软键“缺少的许可证/选项”，显示所有已授权的选项。在“已设置”栏中可以取消不需要的选项。



图 3-7 许可 (示例)

4. 按下软键“根据许可证设置选项”，激活许可证密钥所涵盖的所有选项。用“确认”键确认接下来的安全询问。
5. 按下软键“复位(po)”使刚刚激活的选项生效。出现一条安全询问。
 1. 按下软键“确认”，开始热启动。
-或者-
 2. 按下软键“取消”，中断操作过程。

3.5.3 许可证管理的相关定义

产品

产品处于 SINUMERIK → 软件产品的许可证管理范围内，通过以下数据标记：

- 产品名称
- 货号
- → 许可证号

软件产品

“软件产品”通常指安装在 → 硬件上、用于处理数据的产品。在 SINUMERIK 软件产品的许可证管理范围内，使用每款软件产品都需要相应的 → 许可证。

许可证书 (CoL)

CoL 是获得 → 许可证的证明。产品仅允许由 → 许可证所有者或授权人员使用。CoL 还包含了下列对于许可证管理非常重要的数据：

- 产品名称
- → 许可证号
- 交货单号
- → 硬件序列号

硬件

SINUMERIK 控制系统的组件，基于其唯一标识向其分配 → 许可证，这些组件在 SINUMERIK → 软件产品的许可证管理范畴内被称为硬件。这些组件上也会留存有许可证信息，例如位于 → CF 卡上。

CF 卡

CF 卡用于存储 SINUMERIK solution line 控制系统的所有留存数据，可代表此控制系统。CF 卡是一张存储卡，可以从外插入 → 控制单元中。CF 卡上还包含了以下和许可证管理相关的数据：

- → 硬件序列号
- 许可证信息，包括 → 许可证密钥

硬件序列号

硬件序列号是 → CF 卡固定的组成部分。它相当于控制系统的 ID，具有惟一性。硬件序列号可通过以下方式获取：

- → 许可证书
- 操作界面
- → CF 卡上的印刷标签

3.5 检查和输入许可证

许可证

许可证即使用 → 软件产品的权限。这些权限的代表有：

- → 许可证书 (CoL)
- → 许可证密钥

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。

许可证密钥

许可证密钥是所有 → 许可证总和的“技术代表”，其被分配给一个特定的、通过 → 硬件序列号标记为唯一的 → 硬件。

选件

选件是在基本规格中不包含，必须购买 → 许可证才能使用的 SINUMERIK → 软件产品。

3.6 定义 PLC 报警

3.6.1 PLC 报警的结构

创建和载入 PLC 报警

700000 - 700247 范围内的 PLC 报警由机床制造商定义。只有具有“制造商”存取级别以及对应口令，才可以读写这些报警。

选择操作区域“调试”→“HMI”→“报警文本”，在操作界面上输入制造商 PLC 报警。随后可以选择：

报警文本类型	xml 文件的名称
制造商循环报警	oem_alarms_cycles
制造商 PLC 报警	oem_alarms_plc
制造商子程序信息文本	oem_partprogram_messages

只有在启动时才载入报警文本文件。

- 属性“报警”：红色，显示在“报警清单”中。
- 属性“信息”：黑色，显示在“信息”中。

为了载入报警文本，请重新启动系统。

PLC 报警的结构

用户 PLC 报警具有以下结构：

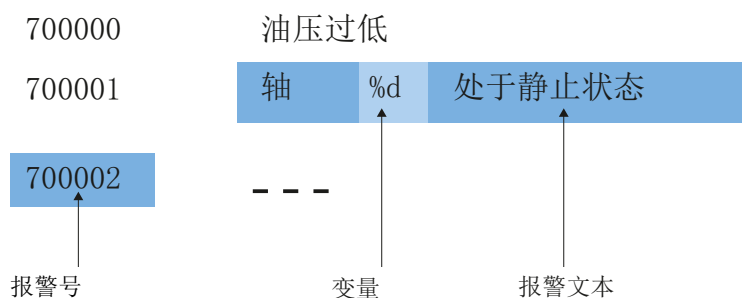


图 3-8 报警结构

3.6 定义 PLC 报警

下表展示了 PLC 报警的生效方式。

- PLC 信号使具有对应编号的报警触发并输出。
- 报警文本可自由选择，最长不超过 255 个字符。
- 触发报警时 NC 的响应在 MD14516 \$MN_USER_DATA_PLC_ALARM 中定义。
- 带有多个通道的控制系统可以通过 MD14518 \$MN_USER_DATA_PLC_ALARM_ASSIGN 进行通道分配。
- 如果为某个报警定义了一个变量，该变量的值会显示在 PLC 变量指定的数据字中。

报警号码	PLC 信号	PLC 变量	报警响应	报警文本
700000	DB1600.DBX0. 0	DB1600.DBD1000	MD14516[0]	报警 1
700001	DB1600.DBX0. 1	DB1600.DBD1004	MD14516[1]	报警 2
700002	DB1600.DBX0. 2	DB1600.DBD1008	MD14516[2]	报警 3
700003	DB1600.DBX0. 3	DB1600.DBD1012	MD14516[3]	报警 4
700004	DB1600.DBX0. 4	DB1600.DBD1016	MD14516[4]	报警 5
700005	DB1600.DBX0. 5	DB1600.DBD1020	MD14516[5]	报警 6
700006	DB1600.DBX0. 6	DB1600.DBD1024	MD14516[6]	报警 7
...				
700246	DB1600.DBX30 .6	DB1600.DBD1987	MD14516[246]	报警 247
700247	DB1600.DBX30 .7	DB1600.DBD1988	MD14516[247]	报警 248

定义带变量的报警文本

报警文本中的变量可以是以下数据类型：

变量	含义
%b	二进制 32 位值
%d	整数十进制值
%f	4 字节浮点数
%i	整数十进制数，带符号
%o	整数八进制数
%u	十进制数，无符号
%x	整数十六进制数

文档

关于 PLC 报警的更多详细信息请参见以下手册：

- SINUMERIK 840D sl/828D：功能手册之基本功能分册：SINUMERIK 828D 的 PLC（P4），章节“PLC 用户报警”
- SINUMERIK 828D 诊断手册，章节“系统响应”

3.6.2 创建 PLC 报警的步骤

编辑提示

在编辑文件时应注意：

- 应在外部 PG/PC 上使用文本编辑器（如：记事本）或者一个 xml 编辑器编辑文件。文件结构请勿修改。
- 创建的报警文本文件应复制到 CF 的目录中：oem/sinumerik/hmi/lng
- 报警文本文件名称应以小写格式写入，这样系统才能识别。
- 报警文本文件在系统启动时转换格式：重新启动系统后，创建的报警才会生效。

操作步骤

首先直接在控制系统上创建 2 个或 3 个报警，这样可以同时编辑多个报警。您可以使用具有正确结构的“文档模板”，向其中添加更多的报警，最后生成文件“oem_alarms_plc_xxx.ts”。缩写“xxx”代表文件创建语言。

1. 选择操作区域“调试”。
2. 按下“HMI”软键。
3. 按下“报警文本”软键。“选择文件”窗口打开。
4. 选择“oem_alarms_plc”，创建制造商 PLC 报警文本。
5. 在“报警号”栏中输入报警号，在“报警文本”栏中输入所需报警文本。报警号和对应的文本可以不连续。如果触发了一个没有报警文本的报警，则只输出报警号。

在报警文本范围内搜索

需要搜索文本或字符串时：

1. 按下软键“搜索>”。“搜索”窗口打开，垂直软键条上显示了一个新菜单。
2. 在“文本”栏中输入待查找的关键词。
3. 将光标移动到“方向”栏中并按下<SELECT>键选择搜索方向（向前、向后）。
4. 如果您需要区分输入文本的大小写，请选中复选框“区分大小写”。
5. 按下软键“搜索+替换”。“搜索和替换”窗口打开。
6. 按下软键“确认”，开始进行搜索。
7. 按下软键“取消”，中断搜索。

其他浏览选项为：

- “转到开始”软键
光标跳至所选报警文本文件的第一个条目上。
- “转到结束”软键
光标跳至所选报警文本文件的最后一个条目上。

另见

文件名中的语种缩写表 (页 365)

示例：创建针对用户 PLC 报警的在线帮助的步骤 (页 72)

3.6.3 配置报警记录

记录

报警记录在操作区域“诊断”中配置。

所有报警和信息及其出现/清除的时间戳都按照时间顺序记录在系统中。但 NC 零件程序“msg”类型的信息除外。所有在显示报警记录时不再生效的报警或信息（即历史事件）也得以保留。

报警记录的形式为环形缓冲器（缺省设置）。在以下情况下，新的事件会覆盖最早的事件：

- 超出最大报警数量（允许范围：0 - 32000）。0 - 32000).
- 事件发生在上次系统通电之前。

安全备份

缺省设置：不备份报警记录。

报警记录的备份会写入 CF 卡上的 `card/user/sinumerik/hmi/log/alarm_log` 目录中。为限制报警记录中事件的数量，请设置筛选条件并选择所需范围：

- 时间间隔
- 报警号范围

注意

CF 卡的使用寿命

对于 CF 卡的使用寿命所给出的写入数据量是一个建议值，不是实际可写入的数据量。如果按照每天平均 60 MB 的数据量来计算，可以使用超过 10 年。

3.6.4 配置记录选项的步骤

配置记录选项

操作步骤：

1. 选择操作区“诊断”。
2. 按下软键“报警记录”。
3. 按下软键“设置”。

3.6 定义 PLC 报警

4. 在“项数”栏中输入所需数字，修改出现/删除的事件的最大数量。
缺省设置为 500 个事件，允许的取值范围是 0 - 32000。
5. 在“文件写入模式”下选择记录方式：
 - “关”，即事件不写入文件。
 - “对每一事件”，即每个事件都写入文件。
 - “时间控制的”，即在特定的时间后重写文件。您可以在补充输入栏“写间隔”中规定时间，单位：秒。
6. 按下“保存记录”，保存报警记录。

这些设置只有在系统重新启动后才会生效。

编辑配置文件

操作步骤：

1. 从目录 /siemens/sinumerik/hmi/template/cfg 中复制配置文件“oem_alarmprot_slaesvconf.xml”。
2. 将文件粘贴到目录 /oem/sinumerik/hmi/cfg 或/user/sinumerik/hmi/cfg 中。
3. 命名此文件为“slaesvconf.xml”
4. 在编辑器中打开用户专用文件“slaesvconf.xml”。
5. 在标签 <Records type ... /> 中输入需要输出事件的数量。
缺省值是 500。允许的取值范围是 0 ... 32000。

或者

直接在操作界面上规定需要输出的事件数量和记录方式：

1. 在“诊断”操作区域中按下软键“报警记录”→“设置>”。
修改了缺省设置后，会自动在目录 /user/sinumerik/hmi/cfg 下创建文件“slaesvconf.xml”。
2. 在标签 <DiskCare type="int" value="-1"/> 中输入持久备份的模式。允许的值有：
 - 1: 不备份报警记录（缺省设置）。
 - 0: 每个报警事件都会使报警记录立即备份。
 - >0: 备份报警记录的时间间隔，单位秒：
一旦出现修改，报警记录会按时间开始备份，即每 n 秒（n > 0）备份。
3. 在标签 <Filter> 中按照记录类型来筛选。
通常：
 - 只有当符合筛选标准时，报警事件才会输入记录中。
 - 定义了多个筛选条件时，请通过逻辑运算符 OR 或 AND 连接这些条件。

这些设置只有在系统重新启动后才会生效。

说明

事件数量

每个报警或信息的出现和删除事件都会占用一个条目，即使它们属于同一条报警或同一条信息。

此外报警记录中还包含有应答事件。它同样需要占用一个条目，即使在目前的记录显示中也无法识别。

示例

满足以下条件时，记录所有报警：

- **CLEARINFO ≠ 15**，即没有零件程序信息：

```
<CONFIGURATION>
  <Protocol>
    <Filters>
      <Siemens_Filter_01 type="QString" value="CLEARINFO NOT 15" />
    </Filters>
  </Protocol>
</CONFIGURATION>
```

- **“SEVERITY 大于 10” 并且 “小于 500”**：

```
<CONFIGURATION>
  <Protocol>
    <Filters>
      <Filter_01 type="QString" value= "SEVERITY HIGHER 10
                                     AND SEVERTY LOWER 500" />
    </Filters>
  </Protocol>
</CONFIGURATION>
```

3.7 创建 OEM 专用在线帮助

概述

除系统中已有的各种在线帮助外，还可创建制造商专用在线帮助，并将它添加到操作软件中。

这种在线帮助使用 HTML 格式，即：由相互链接的 HTML 文档组成。可在特定的窗口中通过目录或关键词索引查找所需主题。与文档浏览器（比如 Windows 浏览器）相似，在窗口的左边部分显示选择概览，点击所需的主题后，会在右边窗口部分显示相应的说明。

这种在线帮助不是上下文相关的帮助。

一般操作步骤：

1. 创建 HTML 文件
2. 创建帮助手册
3. 将在线帮助添加到操作软件中
4. 将帮助文件保存到目标系统中

其他使用情况

可以建立以下 OEM 专有扩展的在线帮助，可以对 SINUMERIK Operate 在线帮助系统进行补充：

- 机床制造商循环与/或 M 功能的在线帮助，可扩展 SINUMERIK 控制系统的编程方式。该在线帮助通过 SINUMERIK Operate 在线帮助“编程”来调用。
另见： 示例：创建编程在线帮助的步骤 (页 77)
- 机床制造商 OEM 专用变量的在线帮助。此在线帮助从 SINUMERIK Operate 的变量角度进行调用。
另见： 示例：创建针对 NC /PLC 变量的在线帮助的步骤 (页 74)

3.7.1 配置文件的结构和句法

“slhlp.xml”文件的句法描述

将帮助手册添加到操作界面上现有的在线帮助系统时，需要使用配置文件“slhlp.xml”。

标签	数量	含义	
CONFIGURATION	1	XML 文档的根元素 用于标识配置文件。	
OnlineHelpFiles	1	引入多个帮助手册的段落。	
<help_book>	*	引入一个帮助手册的段落。	
EntriesFile	1	包含目录和关键词索引的帮助手册的文件名。 属性:	
		value	XML 文件的名称
		type	数值的数据类型 (QString)
III-Technology	0,1	指定该帮助手册适用的工艺。 "All" 表示适用于所有工艺。 若一个帮助手册适用于多个工艺，则工艺之间用逗号隔开。 可能的值: All, Universal, Milling, Turning, Grinding, Stroking, Punching 属性:	
		value	工艺值
		type	数值的数据类型 (QString)
DisableSearch	0,1	取消帮助手册的关键词搜索。 属性:	
		value	true, false
		type	数值的数据类型 (bool)
DisableFullTextSearch	0,1	取消帮助手册的全文搜索。 属性:	
		value	true, false
		type	数值的数据类型 (bool)

3.7 创建 OEM 专用在线帮助

标签	数量	含义	
DisableIndex	0,1	取消帮助手册的关键词索引。	
		属性:	
		value	true, false
		type	数值的数据类型 (bool)
DisableContent	0,1	取消帮助手册的目录。	
		属性:	
		value	true, false
		type	数值的数据类型 (bool)
DefaultLanguage	0,1	若帮助手册中有当前国家的语言, 则会显示此语言的缩写。	
		属性:	
		value	chs, cht, deu, eng, esp, fra, ita, kor, ptb ...
		type	数值的数据类型 (QString)

“数量”列: * 表示 0 或者更多

文件“slhlp.xml”示例

下面的示例展示了帮助手册“hmi_myhelp.xml”的配置, 其中关键词索引功能没有激活:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE CONFIGURATION>
  <CONFIGURATION>
    <OnlineHelpFiles>
      <hmi_myhelp>
        <EntriesFile value="hmi_myhelp.xml" type="QString"/>
        <DisableIndex value="true" type="bool"/>
      </hmi_myhelp>
    </OnlineHelpFiles>
  </CONFIGURATION>
```

3.7.2 帮助手册的结构和句法

帮助手册的句法

帮助手册是一个 XML 文件，可在其中决定在线帮助的结构。文件名可自由选择，例如：`hmi_myhelp`。可在此文件中定义：

- HTML 文档
- 目录和关键词索引

标签	数量	含义	
HMI_SL_HELP	1	XML 文档的根元素	
BOOK	+	帮助手册命名。手册名称可自由选择，只要不是手册中预定义的名称即可，例如： <code>sinumerik_alarm_plc_pmc</code> 。 属性：	
		ref	命名 HTML 文档，其将作为帮助手册的主页显示。
		titel	显示在目录中的帮助手册的标题。
		helpdir	包含帮助手册的在线帮助的目录。
ENTRY	*	在线帮助的章节 属性：	
		ref	命名 HTML 文档，其将作为章节的主页显示。
		titel	显示在内容目录中的章节标题。
INDEX_ENTRY	*	待显示的关键词 属性：	
		ref	命名 HTML 文档，将根据关键词条目跳至此文档。
		titel	在关键词索引中显示的关键词的标题。

“数量”列：

* 表示 0 或者更多

+ 表示 1 或者更多

设定索引的格式

一共有三种关键词索引格式：

- 一级条目： `<INDEX_ENTRY ...title="index"/>`

- 两个二级条目，每个标题都有一个主条目和子条目。

各个条目之间用逗号隔开。

```
<INDEX_ENTRY ...title="mainIndex_1,subIndex_1 with
mainIndex_1"/>
```

- 二级条目，其中第一个标题为主条目，第二个标题为子条目。

各个条目之间用分号隔开。

```
<INDEX_ENTRY ...title="mainIndex_2;subIndex_2
without mainIndex_1"/>
```



图 3-9 示例：二级索引

3.7.3 在线帮助采用的句法的描述

创建 HTML 文件的规定

创建 HTML 格式的帮助用户。此时，可以将所有信息保存在一个 HTML 文件中，或者分开保存在多个 HTML 文件中。

在文件命名时应注意：

- HTML 文件之间的参照关系必须始终以相对路径指定。只有如此才能保证参照关系在开发计算机和目标系统中均有效。

- 若需要在 HTML 文件中的特定点上通过链接进行跳转，必须定义所谓的锚点。

HTML 锚点示例：

```
<a name="myAnchor">这就是锚点</a>
```

另见： 示例： 创建针对用户 PLC 报警的在线帮助的步骤 (页 72)

- HTML 文档的内容必须保存为 UTF-8 编码格式。这样才能保证 HTML 文档在所有支持的语言中都可正确显示。

HTML 标签

可支持如下 HTML 功能范围内的子集：

标签	说明	注释
a	锚点或链接	所支持的属性： href 和 name
address	地址	
b	黑体	
big	大字体	
blockquote	文字缩排	
body	正文	所支持的属性： bgcolor (#RRGGBB)
br	换行	
center	居中段落	
cite	内嵌引用	与标签 i 的效果相同
code	代码	与标签 tt 的效果相同
dd	定义数据	
dfn	定义	与标签 i 的效果相同
div	文档分隔	支持标准程序段属性
dl	定义列表	支持标准程序段属性
dt	定义术语	支持标准程序段属性
em	强调	与标签 i 的效果相同
font	字体大小，语系，颜色	所支持的属性： 大小，字符体和颜色 (#RRGGBB)
h1	1 级标题	支持标准程序段属性
h2	2 级标题	支持标准程序段属性
h3	3 级标题	支持标准程序段属性
h4	4 级标题	支持标准程序段属性
h5	5 级标题	支持标准程序段属性
h6	6 级标题	支持标准程序段属性
head	文档页眉	
hr	水平线	所支持的属性： 宽度（可以是绝对数值或者相对数值）
html	HTML 文档	
i	斜体	

3.7 创建 OEM 专用在线帮助

标签	说明	注释
img	图像	所支持的属性: src , 宽度, 高度
kbd	用户输入文本	
meta	元信息	
li	列表条目	
nobr	不准许折行的文本	
ol	有序列表	支持列表的标准属性
p	段落	支持标准程序段属性 (默认设置: 左对齐)
pre	预先设好格式的文本	
s	删除线	
samp	示例代码	与标签 tt 的效果相同
small	小字体	
span	分组元素	
strong	加强	与标签 b 的效果相同
sub	下标	
sup	上标	
table	表格	所支持的属性: 边框, bgcolor (#RRGGBB) , 单元格间距, 单元格填充, 宽度 (绝对或者相对), 高度
tbody	表格正文	无效
td	表格数据单元	支持表格单元格的标准属性
tfoot	表格页脚	无效
th	表格页眉单元	支持表格单元格的标准属性
thead	表格页眉	打印表格时使用, 可扩展至多页
title	文档标题	
tr	表格行	所支持的属性: bgcolor (#RRGGBB)
tt	打字字体	
u	下划线	
ul	无序列表	支持列表的标准属性
var	变量	与标签 tt 的效果相同

程序段属性

标签 `div`、`dl`、`dt`、`h1`、`h2`、`h3`、`h4`、`h5`、`h6` 和 `p` 支持如下属性：

- 对齐（左对齐，右对齐，居中，垂直居中）
- `dir` (`ltr`, `rtl`)

列表的标准属性

标签 `ol` 和 `ul` 支持如下属性：

- 类型（1, a, A, 方形，圆形，环形）

表格的标准属性

标签 `td` 和 `th` 支持如下属性：

- 宽度（绝对，相对，无数值）
- `bgcolor` (`#RRGGBB`)
- 列间距
- 行间距
- 对齐（左对齐，右对齐，居中，垂直居中）
- 对齐（上对齐，居中，下对齐）

CSS 属性

如下表格中为所支持的 CSS 功能范围：

属性	参数值	说明
<code>background-color</code>	<code><color></code>	元素的背景颜色
<code>background-image</code>	<code><uri></code>	元素的背景图像
<code>color</code>	<code><color></code>	文本的背景颜色
<code>text-indent</code>	<code><length>px</code>	像素中某段第一行的缩进空格
<code>white-space</code>	<code>normal</code> <code>pre</code> <code>nowrap</code> <code>pre-wrap</code>	确定如何处理 HTML 文档中的空白符
<code>margin-top</code>	<code><length>px</code>	像素中上部段落边缘的宽度
<code>margin-bottom</code>	<code><length>px</code>	像素中下部段落边缘的宽度

3.7 创建 OEM 专用在线帮助

属性	参数值	说明
margin-left	<length>px	像素中左边段落边缘的宽度
margin-right	<length>px	像素中右边段落边缘的宽度
vertical-align	baseline sub super middle top bottom	垂直较准文本（在表格中只支持数值的居中、上对齐和下对齐）
border-color	<color>	文本表格的边框颜色
border-style	none dotted dashed dot-dash dot-dot-dash solid double groove ridge inset outset	文本表格的边框样式
background	[<'background-color'> <'background-image'>]	背景属性的缩写
page-break-before	[auto always]	在段落/表格前分页
page-break-after	[auto always]	在段落/表格后分页
background-image	<uri>	元素的背景图像

所支持的 CSS 选择器

除所谓的虚拟选择器等级（如 :first-child, :visited 和 :hover）外，支持所有的 CSS 2.1 选择器等级。

3.7.4 示例：创建 OEM 专用在线帮助手册的步骤

前提条件

创建以下文件：

- 配置文件： "slhlp.xml"

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE CONFIGURATION>
  <CONFIGURATION>
    <OnlineHelpFiles>
      <hmi_myhelp>
```

```
<EntriesFile value="hmi_myhelp.xml" type="QString"/>
<DisableIndex value="false" type="bool"/>
</hmi_myhelp>
</OnlineHelpFiles>
</CONFIGURATION>
```

- 帮助手册定义: "hmi_myhelp.xml"

```
<?xml version="1.0" encoding="UTF-8"?>
<HMI_SL_HELP language="en-US">
  <BOOK ref="index.html" title="Easy Help" helpdir="hmi_myhelp">
    <ENTRY ref="章节_1.html" title="第 1 章">
      <INDEX_ENTRY ref="章节_1.html" title="关键词 1"/>
      <INDEX_ENTRY ref="章节_1.html" title="关键词 2"/>
    </ENTRY>
    <ENTRY ref="章节_2.html" title="第 2 章">
      <INDEX_ENTRY ref="章节_2.html" title="关键词 2"/>
    </ENTRY>
    <ENTRY ref="章节_3.html" title="第 3 章">
      <INDEX_ENTRY ref="章节_3.html" title="关键词 3"/>
      <ENTRY ref="章节_31.html" title="第 3.1 章">
        <INDEX_ENTRY ref="章节_31.html" title="测试;第 3.1 章"/>
      </ENTRY>
      <ENTRY ref="章节_32.html" title="第 3.2 章">
        <INDEX_ENTRY ref="章节_32.html" title="测试;第 3.2 章"/>
      </ENTRY>
    </ENTRY>
  </BOOK>
</HMI_SL_HELP>
```

将帮助文件保存到目标系统中

下面举例说明一个名为"Easy Help" 帮助手册的结构，它包含目录和关键词索引。

3.7 创建 OEM 专用在线帮助

操作步骤:

1. 将配置文件“slhlp.xml”复制到目录:
/oem/sinumerik/hmi/cfg
2. 在路径/oem/sinumerik/hmi/hlp 下创建一个目录用于所需语言的在线帮助:
其中, 请使用章节文件名中的语种缩写表 (页 365)中规定的语种缩写。

说明

输入方法

目录的名称必须为小写。

例如, 需要添加英语帮助手册时, 创建文件夹“eng”。

-
3. 将帮助手册如“hmi_myhelp.xml”移到文件夹“eng”中。
/oem/sinumerik/hmi/hlp/eng/hmi_myhelp.xml
 4. 将帮助文件复制到目录:
/oem/sinumerik/hmi/hlp/eng/hmi_myhelp/

这些设置只有在系统重新启动后才会生效。

说明

更新或更改

在显示帮助手册的目录和关键词索引时, 为了进行快速处理, 可在目录/siemens/sinumerik/sys_cache/hmi/hlp 下将帮助文件保存为二进制格式
slhlp_<Hilfebuch>_*_<lng>.hmi 。

示例: slhlp_hmi_myhelp_*_eng.hmi

必须首先删除这些文件, 在线帮助中的修改才会生效, 并得以显示。

结果

帮助手册由包含子章节的三个章节组成。



图 3-10 示例：OEM 在线帮助

关键字索引中的条目：

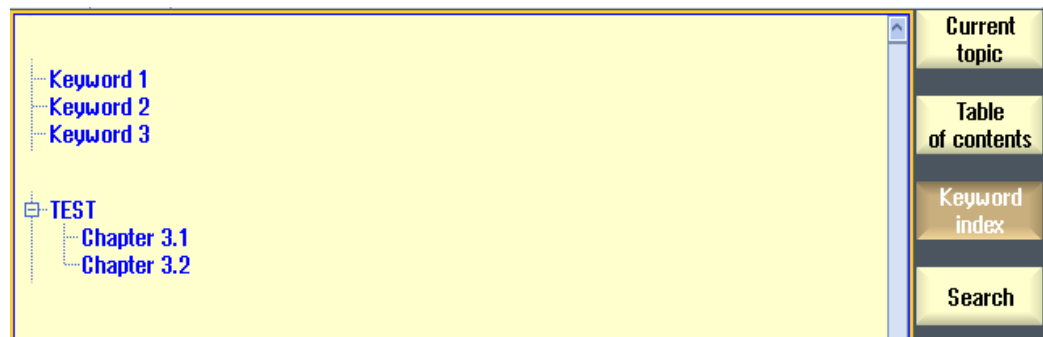


图 3-11 示例：索引

3.7.5 示例：创建针对用户 PLC 报警的在线帮助的步骤

概述

在一个用户 PLC 报警触发后，可以创建针对该报警、上下文相关的在线帮助，这些帮助可以包含详细说明和消除办法。这些针对用户 PLC 报警的在线帮助文本由以下文件管理：“sinumerik_alarm_oem_plc_pmc.html”

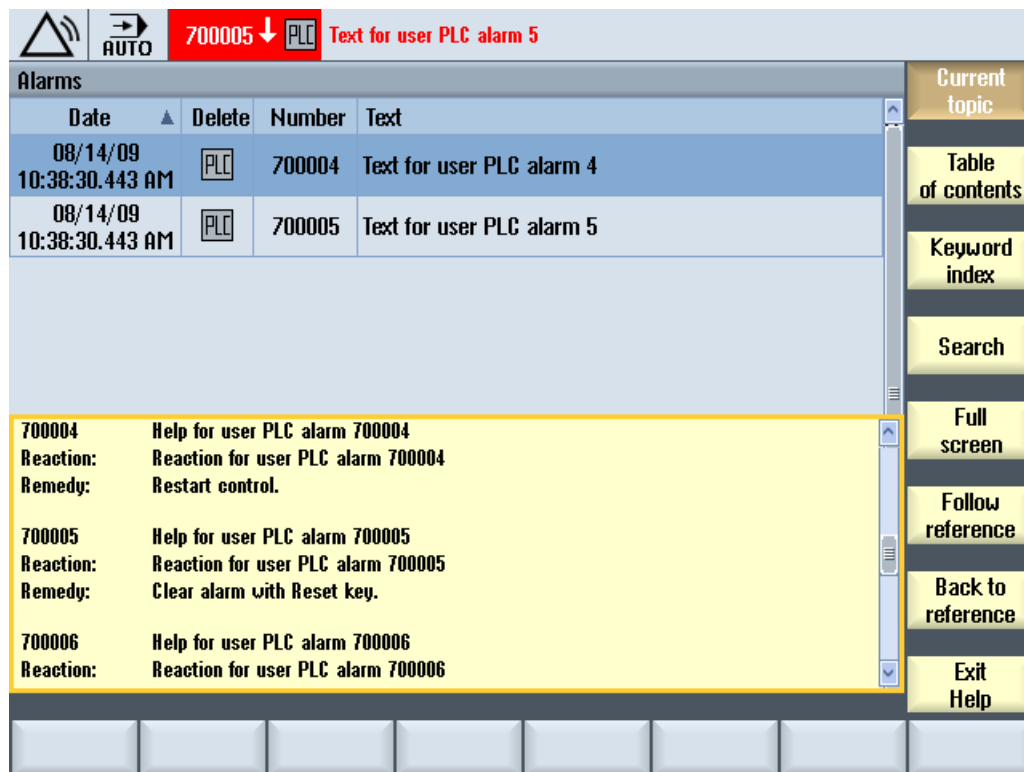


图 3-12 示例：针对用户 PLC 报警的在线帮助

帮助文件的结构

帮助文件中可以使用的条目有：

条目	含义
AlarmNr	报警编号超级链接
	相应报警的帮助文本
<td width="85%">.....</td>	显示在“注释”或“消除方法”栏后面的文本。

创建帮助文件

文件名称和语种不相关，必须为：

```

sinumerik_alarm_oem_plc_pmc.html
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE html PUBLIC>

<html>
<head><title></title></head>
<body>
<table>
...
<tr>
<td width="15%">
<b><a name="700004">700004</a></b></td>
<td width="85%"><b>针对用户 PLC 报警 700004 的帮助 </b></td></tr>
<tr><td valign="top" width="15%"><b>响应: </b></td>
<td width="85%">针对用户 PLC 报警 700004 的响应 </td></tr>
<tr><td valign="top" width="15%"><b>消除办法: </b></td>
<td width="85%">重新启动控制系统。 </td>
</tr>
<br>

<tr>
<td width="15%">
<b><a name="700005">700005</a></b></td>
<td width="85%"><b>针对用户 PLC 报警 700005 的帮助 </b></td></tr>
<tr><td valign="top" width="15%"><b>响应: </b></td>
<td width="85%">针对用户 PLC 报警 700005 的响应 </td></tr>
<tr><td valign="top" width="15%"><b>消除办法: </b></td>
<td width="85%">按下“RESET”键清除报警。 </td>
</tr>
<br>

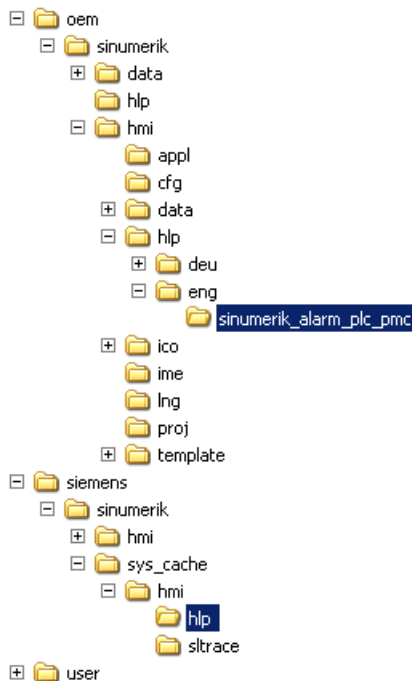
...
</table>
<p></p>
</body>
</html>

```

3.7 创建 OEM 专用在线帮助

操作步骤:

1. 将文件复制到以下一个目录中:
/oem/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_plc_pmc/
/user/sinumerik/hmi/hlp/<lng>/sinumerik_alarm_plc_pmc/
<lng> 代表语种标识。



2. 删除目录下的所有文件:
/siemens/sinumerik/sys_cache/hmi//hlp
这些设置只有在系统重新启动后才会生效。

3.7.6 示例：创建针对 NC /PLC 变量的在线帮助的步骤

概述

为能像下面示例一样，为 NC /PLC 变量或系统变量创建一个上下文敏感的在线帮助，须在和语言相关的 html 文件中管理描述文本。

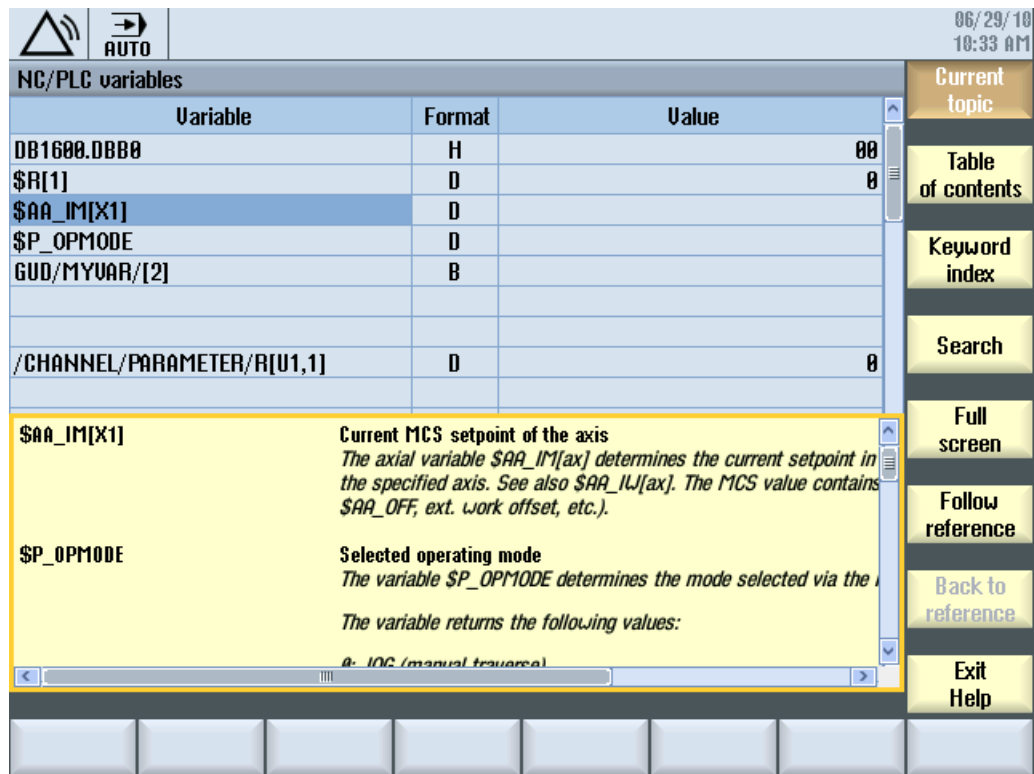


图 3-13 示例：用户变量的在线帮助

在线帮助的建立

对于在线帮助，需要以下文件：

文件	含义
sldgvarviewhelp.ini	配置文件，用于管理 html 文件或多个 html 文件
<lng>/<name>1.html	在线帮助 html 文件的内容是与语言相关的，并且储存在各个语言目录 <lng>中。
<lng>/<name>2.html	
. . .	
<lng>/<name>n.html	

配置文件的建立

文件和语种不相关，必须为：

```
sldgvarviewhelp.ini
[HelpBindings]
```

3.7 创建 OEM 专用在线帮助

sldgvarviewhelp.ini

```

/BAG/STATE/OPMODE = var1_help.html#var1
$AA_IM[X1] = var1_help.html
$R[1] = var1_help.html#var2
/Channel/Parameter/R[u1,1] = var2_help.html#var2
DB1600.DBB0 = var2_help.html#var1
GUD/MyVar[2] = var2_help.html

```

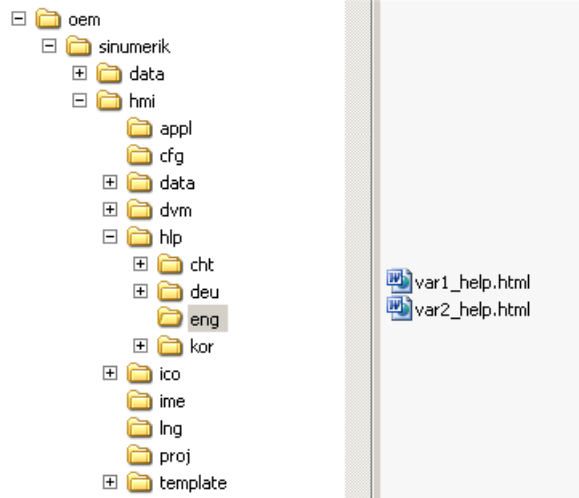
说明

html 文件可以通过任意的 html 编辑器来创建。在配置文件中定义，哪些 html 文件属于在线帮助。

这个描述可以由一个或多个 html 文件构成：例如，每个变量有一个 html 文件或多个同类型的变量在一个文件中。

操作步骤：

1. 将配置文件复制到以下目录中：
/oem/sinumerik/hmi/cfg/sldgvarviewhelp.ini
2. 将 html 文件复制到以下任何一个目录中：
/oem/sinumerik/hmi/hlp/<lng>/
/user/sinumerik/hmi/hlp/<lng>/
<lng> 代表语种标识。



3. 删除目录下的所有文件：
/siemens/sinumerik/sys_cache/hmi//hlp

这些设置只有在系统重新启动后才会生效。

3.7.7 示例：创建编程在线帮助的步骤

前提条件

创建以下文件：

- 配置文件：“prog_help.ini”

```
[milling]
CYCLE1=cycle1_help.html
CYCLE2=cycle2_help.html#TextAnchor1
CYCLE3=cycle3_help.html
CYCLE4=cycle4_help.html

[turning]
CYCLE3=cycle2_help.html
CYCLE4=cycle3_help.html
```

- 帮助手册“slhlp.xml”的配件文件（可选）

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE CONFIGURATION>
<CONFIGURATION>
  <OnlineHelpFiles>
    <hmi_prog_help>
      <EntriesFile value="hmi_prog_help.xml" type="QString"/>
      <DisableIndex value="true" type="bool"/>
    </hmi_prog_help>
  </OnlineHelpFiles>
</CONFIGURATION>
```

- 帮助手册“hmi_prog_help.xml”的配件文件（可选）

```
<?xml version="1.0" encoding="UTF-8"?>
<HMI_SL_HELP language="en-US">
```

3.7 创建 OEM 专用在线帮助

```
<BOOK ref="index.html" title="OEM_CYCLES" helpdir="hmi_prog_help">
  <ENTRY ref="cycle1_help.html" title="Cycle1"></ENTRY>
  <ENTRY ref="cycle2_help.html" title="Cycle2"></ENTRY>
  <ENTRY ref="cycle3_help.html" title="Cycle3"></ENTRY>
  <ENTRY ref="cycle4_help.html" title="Cycle4"></ENTRY>
  <ENTRY ref="cycle_help.html" title="OEM_Cycles"></ENTRY>
</BOOK>
</HMI_SL_HELP>
```

- 与语言相关的文件“<prog_help_eng>.ts”：此文件名是固定的。

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<!DOCTYPE TS>
<TS>
  <context>
    <name>oem_cycles</name>
    <message>
      <source>CYCLE1</source>
      <translation>short description for cycle 1</translation>
      <chars>*</chars>
    </message>
    <message>
      <source>CYCLE2</source>
      <translation>short description for cycle 2</translation>
      <chars>*</chars>
    </message>
    <message>
      <source>CYCLE3</source>
      <translation>short description for cycle 3</translation>
      <chars>*</chars>
    </message>
  </context>
</TS>
```

将帮助文件保存到目标系统中

操作步骤：

1. 将配置文件“prog.ini”复制到目录：
/oem/sinumerik/hmi/cfg
2. 将帮助手册的文件“slhlp.xml”拷贝到以下目录中：
/oem/sinumerik/hmi/cfg

3. 在路径 `/oem/sinumerik/hmi/hlp/<lng>` 下创建一个目录用于所需语言的在线帮助，并把它拷贝到文件中去 `hmi_prog_help.xml`
对此，使用文件名语言识别表中规定的语言符号。目录的名称必须为小写。
 4. 把用于简短描述的、和语言有关的文件 `prog_help_<lng>.ts` 拷贝到以下路径：`/oem/sinumerik/hmi/<lng>/prog_help_<lng>.ts`
 5. 把对 OEM 循环描述的 `html` 文件拷贝到以下目录中：
`/oem/sinumerik/him/hlp/<lng>/hmi_prog_help/cycle<n>_help.html`
- 这些设置只有在系统重新启动后才会生效。

参见

示例：创建 OEM 专用在线帮助手册的步骤 (页 68)

3.7 创建 OEM 专用在线帮助

NC 机床数据设置

4.1 机床数据的分类

存取机床数据的权限

输入或修改机床数据至少需要制造商存取级别的口令。



警告

对人员和机床构成危险

机床数据更改对机床会有重大影响。

错误的参数设置可能危及人员生命，并导致机床损坏。

机床数据的分类

机床数据可以分为：

- 通用机床数据（\$MN）
- 通道专用机床数据（\$MC）
- 轴专用机床数据（\$MA）
- 显示机床数据（\$MM）
- 通用设定数据(\$SN)
- 通用专用设定数据(\$SC)
- 轴专用设定数据(\$SA)
- SINAMICS 机床数据（控制单元和驱动机床数据）：
 - r0001 ... r9999（只读参数）
 - p0001 ... p9999（读写参数）

说明

用于车铣工艺的机床数据已经设置了缺省值，只有在特殊情况下才需要修改。

每个机床数据分类都有一个单独的清单，按下下面的软键可以查看或修改：

4.1 机床数据的分类

通用 机床数据	通道 机床数据	轴 机床数据	用户 视图		控制单元 参数		
			通用 设定数据	通道 设定数据	轴 设定数据	显示 机床数据	

图 4-1 软键栏

从左向右显示机床数据的下列特性：

- 机床数据编号，方括号中是数组索引
- 机床数据名称
- 机床数据值
- 机床数据单位
- 机床数据的有效性
- 数据级

机床数据的物理单位

机床数据的物理单位显示在输入区的右侧：

显示	单位	测量值
m/s**2	m/s ²	加速度
U/s**3	U/s ³	旋转轴加速度的变化速率
kg/m**2	kgm ²	转动惯量
mH	mH（毫亨利）：	电感
Nm	Nm（牛顿米）：	扭矩
us	μs（微秒）：	时间
uA	μA（微安）：	电流强度
uVs	μVs（微伏特秒）：	磁通量
userdef	用户自定义：	单位由用户确定。

对于没有单位的机床数据，单位栏为空。

如果未提供数据，则用“#”代替数值进行显示。如果值以“H”结尾，则为十六进制值。

机床数据的生效条件

在右栏中显示什么时候机床数据的修改生效：

cf	机床数据修改只在按下软键“机床数据有效”后才生效。
po	机床数据修改只在在对 NCK 执行上电复位操作后才生效。
re	机床数据修改只在按下“RESET”键后才生效。
so	机床数据修改立即生效。

用户视图

用户视图汇集了用户专用的各种机床数据。它可以提供某个操作状态中来自不同操作区的各种相关机床数据用于处理。

用户视图保存在 CF 卡的路径：

```
user/sinumerik/hmi/template/user_views
```

系统中提供了以下用户视图模板：

- Electrical_Startup
- Mechanical_Startup
- Optimizing_Axis

另见

数据级说明：功能手册 基本功能(P4)

机床数据的详细说明可以参见参数手册和功能手册中的相关章节。

4.2 执行外部 CNC 系统的零件程序

说明

缺省设置

SINUMERIK 828D 已经预先为车削或铣削工艺设置了以下配套的机床数据。这些机床数据是不可以更改的，不显示在系统屏幕上。

激活 ISO 语言功能

通过机床数据 MD18800 \$MN_EXTERN_LANGUAGE 可以激活外部语言。ISO 语言 M 或 T 语言的选择通过 MD10880 \$MN_EXTERN_CNC_SYSTEM 实现。

西门子编程语言到 ISO 编程语言的转换通过 G 功能组 47 中的两个 G 指令实现：

- G290: 西门子数控编程语言生效
- G291: ISO 数控编程语言生效

语言转换后，当前刀具、刀具补偿和零点偏移仍保持不变。必须在单独的 NC 程序段中写入 G290 和 G291。

在 SINUMERIK 828D 的供货范围中包含了外部编程语言。MD10712: \$MN_NC_USER_CODE_CONF_NAME_TAB 只用于西门子数控编程语言指令。

资料

功能手册 ISO 语言

4.3 用 Advanced Surface 加工自由形状

精优曲面加工 (Advanced Surface)

在“高速切削(HSC)”区内执行 CAM 程序时，控制系统必须以最短程序句、最快进给率加工工件。在这种加工中，进给率一方面要超过 10 米/分钟，另一方面工件必须达到微米级的精度、具有良好的表面质量。

借助 CYCLE832 提供的多种加工方案，系统很好地协调两者的关系：

- 在粗加工时刀具磨削出工件轮廓，此时重点放在速度上。
- 在精加工时重点则放在精度上。

在这两种情况下系统都通过设定的公差来控制加工轮廓的尺寸，以达到所需的表面质量。

检查 Advanced Surface 的设置

在 Toolbox 828D 安装成功后，目录\Examples\AS_checkprograms 下会生成多个 MDC3AXVSPF 类型的测试程序：

另外，自述文件 readme.txt 还推荐了使用测试程序检查 Advanced Surface 机床数据时的设置。在启动程序后，系统会在目录 SPF.DIR 下生成文件 MDADVS.SPF，该文件以列表的形式列明检查结果。

示例：

(仅文件的部分选段，非完整文件)

```
Machine Data setting different to required Advanced Surface
Setting
Machine Data setting different to recommended Advanced Surface
Setting
N20470 $MC_CPREC_WITH_FFW=1, recommended: 3
N20485 $MC_COMPRESS_SMOOTH_FACTOR[2]=0, recommended: 0.0001
N20485 $MC_COMPRESS_SMOOTH_FACTOR[3]=0, recommended: 0.0001
N20485 $MC_COMPRESS_SMOOTH_FACTOR[4]=0, recommended: 0.0001
N20486 $MC_COMPRESS_SPLINE_DEGREE[2]=3, recommended: 5
N20486 $MC_COMPRESS_SPLINE_DEGREE[3]=3, recommended: 5
N20486 $MC_COMPRESS_SPLINE_DEGREE[4]=3, recommended: 5
N20487 $MC_COMPRESS_SMOOTH_FACTOR_2[2]=0, recommended: 0.5
N20487 $MC_COMPRESS_SMOOTH_FACTOR_2[3]=0, recommended: 0.5
N20487 $MC_COMPRESS_SMOOTH_FACTOR_2[4]=0, recommended: 0.5
```

4.4 Advanced Surface 的机床数据

Advanced Surface 的机床数据

为了在模具加工中加工出高表面质量的模具，我们建议对以下机床数据进行适当设置：

通用机床数据：

机床数据编号	名称	缺省设置	建议	数据级	参数
10200	\$MN_INT_INCR_PER_MM	100000	100000	M	--
10210	\$MN_INT_INCR_PER_DEG	100000	100000	M	--
10682	\$MN_CONTOUR_SAMPLING_FACTOR	1	1	M	--

通道专用机床数据：

机床数据编号	名称	缺省设置	建议	数据级	参数
20150[3]	\$MC_GCODE_RESET_VALUES	3	3	M	FIFOCTRL
20150[44]	\$MC_GCODE_RESET_VALUES	2	2	M	UPATH
20170	\$MC_COMPRESS_BLOCK_PATH_LIMIT	20	20	M	--
20172	\$MC_COMPRESS_VELO_TOL	60000	1000	M	--
20443[0]	\$MC_LOOKAH_FFORM	0	0	M	DYNNORM
20443[1]	\$MC_LOOKAH_FFORM	0	0	M	DYNPOS
20443[2]	\$MC_LOOKAH_FFORM	0	1	M	DYNROUGH
20443[3]	\$MC_LOOKAH_FFORM	0	1	M	DYNSEMIFIN
20443[4]	\$MC_LOOKAH_FFORM	0	1	M	DYNFINISH
20455	\$MC_LOOKAH_FUNCTION_MASK	0	0	M	--
20480	\$MC_SMOOTHING_MODE	0	0	M	--
20482	\$MC_COMPRESSOR_MODE	100	100	M	--
20485	\$MC_COMPRESS_SMOOTH_FACTOR [0-4]	0	0	M	--
20486	\$MC_COMPRESS_SPLINE_DEGREE[0-4]	3	3	M	--

机床数据编号	名称	缺省设置	建议	数据级	参数
20490	\$MC_IGNORE_OVL_FACTOR_FOR_A DIS	1	1	M	--
20550	\$MC_EXACT_POS_MODE	3	3	M	--
20600[0]	\$MC_MAX_PATH_JERK	10000	10000	M	DYNNORM
20600[1]	\$MC_MAX_PATH_JERK	10000	10000	M	DYNPOS
20600[2]	\$MC_MAX_PATH_JERK	10000	10000	M	DYNROUGH
20600[3]	\$MC_MAX_PATH_JERK	10000	10000	M	DYNSEMIFIN
20600[4]	\$MC_MAX_PATH_JERK	10000	10000	M	DYNFINISH
20602[0]	\$MC_CURV_EFFECT_ON_PATH_ACC EL	0	0	M	--
20602[1]	\$MC_CURV_EFFECT_ON_PATH_ACC EL	0	0	M	--
20602[2]	\$MC_CURV_EFFECT_ON_PATH_ACC EL	0,6	0,65	M	--
20602[3]	\$MC_CURV_EFFECT_ON_PATH_ACC EL	0,6	0,6	M	--
20602[4]	\$MC_CURV_EFFECT_ON_PATH_ACC EL	0,6	0,5	M	--
20603[0]	\$MC_CURV_EFFECT_ON_PATH_JERK	0	0	M	--
20603[1]	\$MC_CURV_EFFECT_ON_PATH_JERK	0	0	M	--
20603[2]	\$MC_CURV_EFFECT_ON_PATH_JERK	0	0	M	--
20603[3]	\$MC_CURV_EFFECT_ON_PATH_JERK	0	0	M	--
20603[4]	\$MC_CURV_EFFECT_ON_PATH_JERK	0	0	M	--
20605[0]	\$MC_PREPDYN_SMOOTHING_FACTO R	0	1	M	--
20605[1]	\$MC_PREPDYN_SMOOTHING_FACTO R	0	1	M	--

4.4 Advanced Surface 的机床数据

机床数据编号	名称	缺省设置	建议	数据级	参数
20605[2]	\$MC_PREPDYN_SMOOTHING_FACTOR	0	1	M	--
20605[3]	\$MC_PREPDYN_SMOOTHING_FACTOR	0	1	M	--
20605[4]	\$MC_PREPDYN_SMOOTHING_FACTOR	0	1	M	--
20606[0]	\$MC_PREPDYN_SMOOTHING_ON	0	0	M	DYNNORM
20606[1]	\$MC_PREPDYN_SMOOTHING_ON	0	0	M	DYNPOS
20606[2]	\$MC_PREPDYN_SMOOTHING_ON	0	1	M	DYNROUGH
20606[3]	\$MC_PREPDYN_SMOOTHING_ON	0	1	M	DYNSEMIFIN
20606[4]	\$MC_PREPDYN_SMOOTHING_ON	0	1	M	DYNFINISH
28060	\$MC_MM_IPO_BUFFER_SIZE	150	150	M	--
28070	\$MC_MM_NUM_BLOCKS_IN_PREP	80	80	M	--
28302[0]	\$MC_MM_PROTOCOL_NUM_ETP_STD_T YP	28	28	M	--
28520	\$MC_MM_MAX_AXISPOLY_PER_BLOCK	3	5	M	--
28530	\$MC_MM_PATH_VELO_SEGMENTS	5	5	M	--
28533	\$MC_MM_LOOKAH_FFORM_UNITS	18	18	M	--
28540	\$MC_MM_ARCLENGTH_SEGMENTS	10	10	M	--

进给轴机床数据:

机床数据编号	名称	缺省设置	建议	数据级	参数
32300	\$MA_MAX_ACCEL[0-3]	1	1	M	--
32300	\$MA_MAX_ACCEL[4]	1	1	M	--
32310[0]	\$MA_MAX_ACCEL_OVL_FACTOR	1.2	1.2	M	--
32310[1]	\$MA_MAX_ACCEL_OVL_FACTOR	1.2	1.2	M	--
32310[2]	\$MA_MAX_ACCEL_OVL_FACTOR	1.2	1.2	M	--
32310[3]	\$MA_MAX_ACCEL_OVL_FACTOR	1.2	1.2	M	--
32310[4]	\$MA_MAX_ACCEL_OVL_FACTOR	1.2	1.2	M	--

机床数据编号	名称	缺省设置	建议	数据级	参数
32400	\$MA_AX_JERK_ENABLE	0	0	M	--
32402	\$MA_AX_JERK_MODE	2	2	M	--
32410	\$MA_AX_JERK_TIME	0,001	0,001	M	--
32431[0]	\$MA_MAX_AX_JERK	1000000	1000000	I	DYNNORM
32431[1]	\$MA_MAX_AX_JERK	1000000	1000000	I	DYNPOS
32431[2]	\$MA_MAX_AX_JERK	1000000	1000000	I	DYNROUGH
32431[3]	\$MA_MAX_AX_JERK	20	20	I	DYNSEMIFIN
32431[4]	\$MA_MAX_AX_JERK	20	20	I	DYNFINISH
32432[0]	\$MA_PATH_TRANS_JERK_LIM	1000000	1000000	I	DYNNORM
32432[1]	\$MA_PATH_TRANS_JERK_LIM	1000000	1000000	I	DYNPOS
32432[2]	\$MA_PATH_TRANS_JERK_LIM	1000000	1000000	I	DYNROUGH
32432[3]	\$MA_PATH_TRANS_JERK_LIM	20	20	I	DYNSEMIFIN
32432[4]	\$MA_PATH_TRANS_JERK_LIM	20	20	I	DYNFINISH
32620	\$MA_FFW_MODE	3	3	M	DYNNORM
32640	\$MA_STIFFNESS_CONTROL_ENABLE	1	1	M	DSC
33120	\$MA_PATH_TRANS_POS_TOL	0,005	0,005	U	--

设定数据:

机床数据编号	名称	缺省设置	建议	数据级	参数
42470	\$SC_CRIT_SPLINE_ANGLE	36°	36°	U	--
42471	\$SC_MIN_CURV_RADIUS	1	1	M	--
42500	\$SC_SD_MAX_PATH_ACCEL	10000	10000	U	--
42502	\$SC_IS_SD_MAX_PATH_ACCEL	0	0	U	--
42510	\$SC_SD_MAX_PATH_JERK	10000	10000	U	--
42512	\$SC_IS_SD_MAX_PATH_JERK	0	0	U	--

4.5 用于模拟量主轴的机床数据

编码器实际值的连接

在主轴上连接实际值编码器的前提是带有数字量驱动的轴（SINAMICS 轴）只有一个编码器。编码器模块（例如：SMC30）和直接安装在主轴上的实际值编码器相连后，作为“第二编码器”连接到该数字量驱动轴上。SINAMICS 轴的 PROFIdrive 报文应设计用于两个编码器实际值。取决于 MD11240 \$MN_PROFIBUS_SDB_NUMBER 的设置，SINUMERIK 828D 的标准 SDB 支持西门子报文 116 或 136。

功能方面的限制：

- 由于在模拟量主轴上不存在设定值报文，因此基于报文的相应功能也不存在，例如：DSC 和转矩前馈等。
- 信号“Drive Ready”由驱动软件作为“置位”信号模拟。
- 由于无法访问驱动参数，因此与之相关的功能也受到限制（例如：主轴负载显示、自动转速调节器优化和自动位置调节器优化等）。在“服务一览”画面中，“模拟量主轴”一栏中缺少的值显示为灰色：“该轴不相关”。

主轴设定值的连接

模拟量主轴连接到接口 X252 上。取决于模拟量主轴的运行方式，它会输出以下信号：

主轴类型	信号名称	含义
双极主轴	AOUT	模拟量输出端 +/-10 V
	AGND	模拟量输出端 0 V 基准信号
	DOUT11	调节器使能
单极主轴，具有单独的使能信号和方向信号	AOUT	模拟量输出端 +10 V
	AGND	模拟量输出端 0 V 基准信号
	DOUT12	负运行方向（在缺少调节器使能时也保持置位）
单极主轴，使能信号和方向信号相关联	AOUT	模拟量输出端 +10 V
	AGND	模拟量输出端 0 V 基准信号

主轴类型	信号名称	含义
	DOUT11	调节器使能和正运行方向
	DOUT12	调节器使能和负运行方向

主轴设定值的数模转换以 14 位的分辨率进行。

相关的机床数据

以下机床数据用于设置模拟量主轴：

- 激活模拟量主轴：

MD30100 \$MA_CTRLOUT_SEGMENT_NR = 0

将本地总线设为总线的一部分，便可以激活用于设定值输出的模拟量主轴。

- 选择模拟量主轴的运行方式：

MD30134 \$MA_IS_UNIPOLAR_OUTPUT	
=0	双极输出端(+/-10 V)
=1	单极主轴，具有单独的使能信号和方向信号
=2	单极主轴，使能信号和方向信号相关联

- 额定输出电压必须和模拟量驱动的额定转速相匹配：

MD32250 \$MA_RATED_OUTVAL

MD32260 \$MA_RATED_VELO

- 如果希望模拟量主轴不带实际值编码器运行，需要将“编码器数量”设为 0。

MD30200 \$MA_NUM_ENCS = 0

- 在使用一个直接安装在主轴上的实际值编码器时，需要选择 PROFIBUS 实际值检测的类型：

MD30240 \$MA_ENC_TYPE = 1 或 4

- 主轴实际值编码器必须定义为 SINAMICS 轴的“编码器 2”。因此，实际轴指定给的驱动应该和连接了 SMC30 的轴相同。

MD30220[0] \$MA_ENC_MODULE_NR[0] = SINAMICS 轴的 MD30220[0]

- 实际值的输入应该设为 SINAMICS 轴的“编码器 2”：

MD30230[0] \$MA_ENC_INPUT_NR[0] = 2

4.5 用于模拟量主轴的机床数据

- 在连接了主轴实际值编码器后，可以激活“自动漂移补偿”：
MD36700 \$MA_DRIFT_ENABLE
MD36710 \$MA_DRIFT_LIMIT
- 漂移原始值和是否存在主轴实际值编码器无关，它始终作为附加的转速设定值接入：
MD36720 \$MA_DRIFT_VALUE

示例

3 根轴和一根带实际值编码器的模拟量主轴（SMC30 作为“第二编码器”连接到 Y 轴上）。

MD 名称	X	Y	Z	SP	含义
MD30100 CTRLOUT_SEGMENT_NR	5	5	5	0	总线段
MD30110 CTRLOUT_MODULE_NR	1	2	3	1	指定模块
MD30120 CTRLOUT_NR	1	1	1	1	指定输出端
MD30130 CTRLOUT_TYPE	1	1	1	1	输出方式
MD30134 IS_UNIPOLAR_OUTPUT	0	0	0	0	“0”表示单极
MD30200 NUM_ENCS	1	1	1	1	编码器数量
MD30220[0] ENC_MODULE_NR[0]	1	2	3	2	SMC30 连接到 Y 轴上
MD30230[0] ENC_INPUT_NR[0]	1	1	1	2	“第二编码器”的输入
MD30240 ENC_TYPE	1/4	1/4	1/4	1/4	编码器类型

MD 名称	SP	含义
MD32250 RATED_OUTVAL	80 (10 V 的 80%)	额定输出电压
MD32260 RATED_VELO	3000 (8 V)	电机额定转速
MD36700 DRIFT_ENABLE	0	取消自动漂移补偿。
MD36710 DRIFT_LIMIT	0	自动漂移补偿的限幅
MD36720 DRIFT_VALUE	0	漂移原始值

文档


其它信息请参见： 功能手册之基本功能分册， 主轴（S1）

4.6 禁用控制系统（数控锁功能）

前提条件

前提条件如下：

- 需要许可证的选件，以使用禁用功能：

	软件选件 使用该功能需要下列选件：数控锁功能
---	----------------------------------

- **Access MyMachine**，以激活禁用功能。
- 编程工具，以通过 **PLC** 保护禁用功能。

使用范围

机床制造商在决定出售机床前会考虑其是否带有数控锁功能以及西门子是否通过 **CF** 卡的序列号为机床的控制系统授权了数控锁功能。为此，机床制造商可借助调试工具 **Access MyMachine** 将一个日期（**NC** 启动自该日期起禁用）保存到控制系统中。通过 **PLC** 保护数控禁用功能。

注意

用于解除禁用的数据丢失

如果用于解除禁用的数据丢失，必须更换控制系统和 **CF** 卡。

一般操作顺序

调试机床时必须在 **PLC** 项目中激活该功能并连同项目一并加载到控制系统中。

激活该功能所需的步骤如下：

- **Access MyMachine**：生成禁用文件。
- 在控制系统上：
 - 设置正确的日期和时间。
 - 读取文件并重启操作软件。
 - 创建调试存档。
- **Access MyMachine**：生成并保存解除禁用文件。

取消该功能所需的步骤如下：

- 读取控制系统上的解除禁用文件并重启操作软件。

通过 MD17300 \$MN_CNC_LOCK_WARNING_TIME 可以设置一个预警时间（缺省设置：30 天）。从该时间点起，系统每天输出一个报警。

文档

有关该功能的详细信息请参考：

- 功能手册之基本功能，“SINUMERIK 828D 的 PLC” 一章 (P4)
- SINUMERIK Integrate Access MyMachine /P2P 的在线帮助

4.7 管理数据

应用

“管理数据”功能用于简化调试，并可以保存、加载和比较以下数据：

- 机床数据
- 设定数据
- 驱动数据
- 补偿数据

与调试存档不同，此处只会将单个控制系统对象（轴、通道、SERVO、电源等）保存为 ASCII 格式（*.tea）。此文件可编辑，并可传输至其他同类型的控制系统对象。“管理数据”功能也可用于在 SINAMICS 驱动上传输驱动对象。

管理数据

通过“管理数据”功能可进行以下操作：

- 在控制系统内传输数据
- 将数据保存至文件
- 将数据载入至文件
- 比较数据

可备份以下数据，并将其存放至 CF 卡上的下列绝对路径：

- `user/sinumerik/hmi/data/backup/ec` 用于补偿数据
- `user/sinumerik/hmi/data/backup/md` 用于机床数据
- `user/sinumerik/hmi/data/backup/sd` 用于设定数据
- `user/sinumerik/hmi/data/backup/snx` 用于 SINAMICS 参数

可通过“调试”→“机床数据”→“管理数据”执行这些功能。

“在控制系统内传输数据”的示例

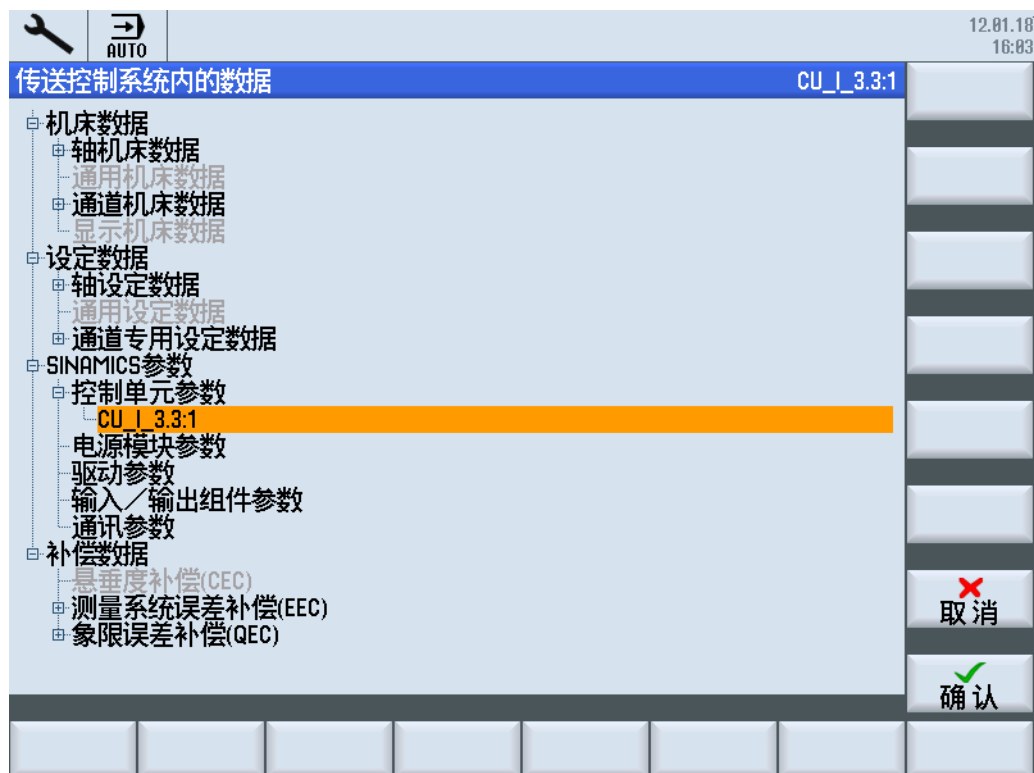


图 4-2 管理数据

4.7.1 在控制系统内传输数据的步骤

在控制系统内传输数据

注意
机床保护
出于安全原因，只能在使能被禁用的情况下传输机床数据和设定数据。

步骤：

1. 选择“在控制系统内传输数据”选项。
2. 在数据结构中选择源数据，并按下“确定”进行确认。
3. 在选择列表中选择数据需要传输至的对象，例如另一个轴或另一个驱动对象，并按下“确定”进行确认。

4.7 管理数据

4. 请遵循安全提示，并检查机床以及驱动上的使能。
5. 对于驱动数据，数据通过软键“装载”传输至目标对象。

4.7.2 保存和装载数据的步骤

将数据保存至文件

步骤:

1. 选择“将数据保存至文件”选项。
2. 在数据结构中选择需要保存至文件的数据，并按下“确定”进行确认。
3. 选择目录或 USB 存储媒介作为存储路径，并输入名称。

说明

SINAMICS 参数

保存时始终会生成 ASCII 文件(*.TEA)。

保存驱动数据时会生成以下类型的三个文件:

- 一个二进制文件 (*.ACX)，不可读。
 - 一个 ASCII 文件 (*.TEA)，可在 ASCII 编辑器中读取或编辑。
 - 一个日志文件 (*.log)，在故障情形下包含信息文本，存储成功时则为空。
-

从文件加载数据

注意
机床保护
出于安全原因，只能在使能被禁用的情况下传输机床数据和设定数据。

步骤:

1. 选择“从文件加载数据”选项。
2. 在数据结构中选择保存的数据，并按下“确定”进行确认。
3. 在选择列表中选择数据需要传输至的对象，例如另一个轴或另一个驱动对象，并按下“确定”进行确认。
4. 请遵循安全提示，并检查机床以及驱动上的使能。
5. 对于驱动数据，数据通过软键“装载”传输至目标对象。

4.7.3 比较数据的步骤

比较数据

进行数据比较时，可选择不同的数据源：控制系统上的当前数据，或保存在文件中的数据。

步骤：

1. 选择“比较数据”选项。
2. 在数据结构中选择需要进行比较的数据。
3. 按下软键“接收至列表”，将数据接收至列表下方的显示区域。
4. 通过软键“从列表删除”重新删除数据。
5. 若列表中有超过 2 个数据对象，可通过激活复选框对列表中的 2 个或多个数据进行比较。
6. 按下软键“比较”启动比较。参数列表中包含较多参数时，比较结果的显示可能要持续一定时间。
7. 通过软键“图例”显示或隐藏图例。显示预设如下：
 - 显示不同参数。
 - 不显示相同参数。
 - 显示非常用参数。

4.8 带自适应特性曲线的摩擦补偿

前提条件

在控制系统的“调试”操作区域调用此功能：软键“NC” → 软键“摩擦补偿”。对话框“摩擦补偿”打开。

机床数据	含义
MD32500 \$MA_FRICT_COMP_ENABLE = 1	使能轴摩擦补偿
带自适应特性曲线的摩擦补偿：	
MD32490 \$MA_FRICT_COMP_MODE = 3	接通时间点取决于速度设定值
MD32490 \$MA_FRICT_COMP_MODE = 4	接通时间点取决于位置控制器输出



软件选件

使用该功能需要下列选件：摩擦补偿

备份测量结果

若在该选件未经过授权的情况下为轴激活摩擦补偿，则会触发报警。

系统会将测量结果写入机床数据。测量序列中测量结果如下变化：

- 按下“启动”软键启动测量序列时，机床数据中设置的摩擦补偿值会被复位。
- 按下“确认”软键结束测量序列时，机床数据中为新的摩擦补偿值。
- 通过“取消”软键终止测量序列时，机床数据中将重新启用之前的摩擦补偿值。

若在执行测量序列期间发生了掉电，那么新的摩擦补偿值和之前的旧值均会丢失。

为了防止数据丢失，建议创建一个调试存档，将上一次完整测量序列的轴机床数据备份于其中。不需要选择“含补偿数据”选项。在 SINUMERIK 828D 上创建一个“INDIVIDUAL”类型的数据存档，将上一次完整测量序列的补偿值备份于其中。

文档

更多详细信息参见功能手册之扩展功能中的补偿章节（K3）。

4.8.1 带速度接通脉冲的摩擦补偿

在 SINUMERIK Operate 上操作

通过此功能可选择运行一根或两根轴。测量结果以圆弧图的形式显示。所输入的摩擦补偿值仅对标题行中选定的轴生效。

若未手动输入缩放值，系统将自动调整图形显示的缩放比例。可在一个测量过程中执行最多九次采用不同进给率的测量。在 1 轴和 2 轴运行中，优化针对上下的反转点。测量由初始化阶段和测量阶段本身组成。右下方的状态行中会显示当前的测量进度。

如果测量少于九次，则在不需要的进给率处填写零。需要满足下列边界条件：

- 进给率只能在测量序列结束时可为零。
- 须同时进行所有步骤。

步骤：

1. 通过软键“轴 +”或“轴 -”选择摩擦补偿已激活的轴。
测量的半径、进给率和旋转方向均已在下列设定数据中预设，适用于通道中的所有轴。可通过软键“数据列表”加以修改：

名称	单位	机床数据
半径（线性轴）	mm	SD55820 \$SCS_FRICT_OPT_RADIUS
半径（回转轴）	°（度）	SD55821 \$SCS_FRICT_OPT_RADIUS_ROT
进给率（线性轴）	[mm/min]	SD55822 \$SCS_FRICT_OPT_FEED[0...8]
进给率（回转轴）	[°/min]	SD55823 \$SCS_FRICT_OPT_FEED_ROT[0...8]
旋转方向（仅在有 2 根轴时生效）	---	SD55828 \$SCS_FRICT_OPT_DIR

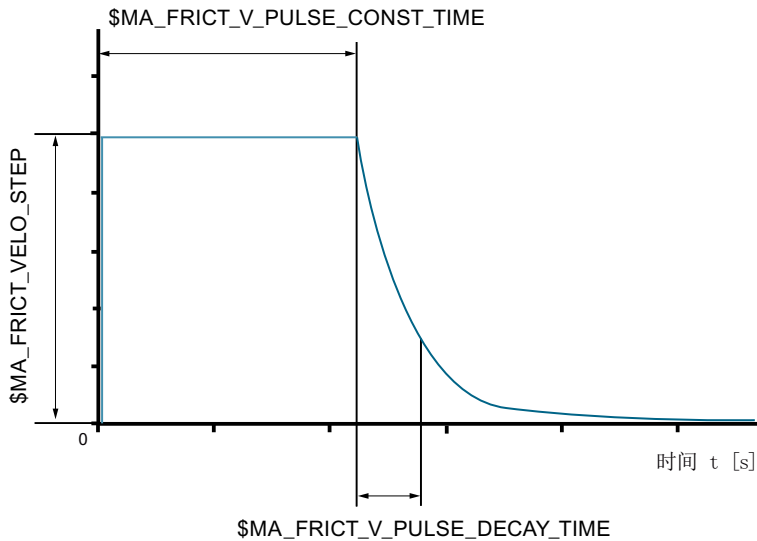
2. 按下软键“开始”启动测量过程。此时会自动创建一个零件程序并由 NC 选中，在该程序中调用 CYCLE790，此循环用于以预设的进给率反转轴。
在启动零件程序前，系统会自动检查，轴的动态响应是否能达到输入的进给率。如果达不到，会输出消息，要求降低进给率。
3. 之后系统会打开测量序列的零件程序，此时可额外添加指令。按下“确认”关闭窗口并继续测量。

4.8 带自适应特性曲线的摩擦补偿

4. 此时第一步测量会被选中，其采用的是测量序列的首个进给率。请将摩擦补偿值输入以下输入栏：

输入栏	单位	机床数据
振幅	[mm/min] 或 [°/min]	MD32571 \$MA_FRICT_VELO_STEP
失效时间	[s]	MD32574 \$MA_FRICT_V_PULSE_DECAY_TIME
生效时间	[s]	MD32573 \$MA_FRICT_V_PULSE_CONST_TIME

若需依据运行方向为轴输入不同的补偿值，请选择“分别设置换向点”选项。否则两个运行方向会使用相同的值。



5. 按下软键“下一步”，以下一个进给率启动新测量，并输入新值。重复测量时，系统会依据上次的测量建议和进给率匹配的值。

在最后一步完成后，按下“确认”结束测量并保存数据。

说明

增量修改数值

若需增量修改补偿值，请使用下列按键组合：

- 使用 <INSERT> 键激活选择的输入栏，此时其颜色会发生变化。
- 使用 <Page Up> / <Page Down> 键及光标键 ↑↓ 以不同的增量步距修改数值。
- 按下 <INPUT> 键完成输入。

⇒ 其它操作:

- 软键“轴 +”、“轴 -”，用于选择摩擦补偿生效的轴。
- 软键“启动”，用于启动测量。
- 软键“下一步”，用于选择下一个进给率。
- 软键“上一步”，用于回到之前的进给率。
- 软键“数据列表”，用于切换至机床数据列表。
- 软键“取消”，用于终止记录并放弃输入。
- 软键“确认”，用于接收在摩擦补偿中测定的值。

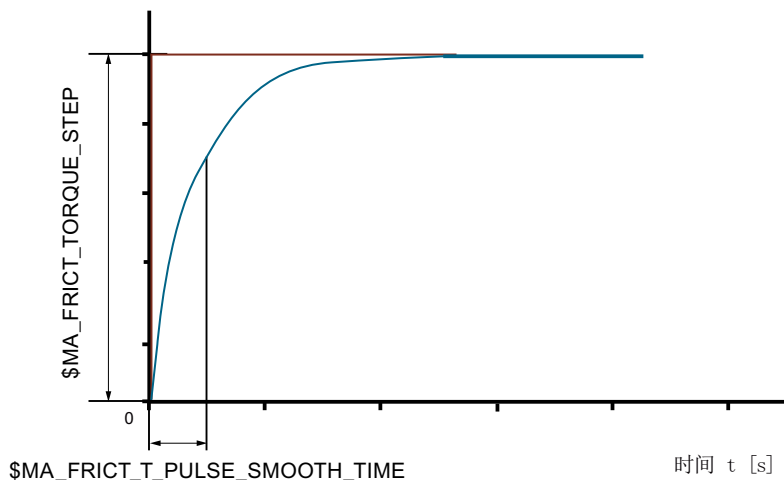
4.8.2 带力矩接通脉冲的摩擦补偿

参数设置机床数据

为了使用取决于加速度的摩擦补偿力矩接通脉冲，必须参数设置下列机床数据：

名称	单位	机床数据
取决于加速度的力矩接通脉冲振幅	---	MD32576 \$MA_FRICT_TORQUE_STEP
力矩接通脉冲的延时	[s]	MD32577 \$MA_FRICT_T_PULSE_DELAY_TIME
力矩接通脉冲的提升时间	[s]	MD32578 \$MA_FRICT_T_PULSE_SMOOTH_TIME
力矩接通脉冲振幅的权重系数	---	MD32588 \$MA_FRICT_T_STEP[0...9]

4.8 带自适应特性曲线的摩擦补偿



请在 MD32588 \$MA_FRICT_T_STEP[0...9] 为每个加速度值输入一个 -1.0 至 1.0 之间的系数，用于对力矩接通脉冲进行加权。

4.8.3 数据列表

数据列表中会显示带自适应特性曲线的摩擦补偿所涉及的机床数据和设定数据。

- **机床数据：**所有机床数据均可编辑。
- **设定数据：**半径、进给率和旋转方向**只能**在启动测量序列前修改，测量序列执行期间无法输入。

⇒ 其它操作：

软键“<<”，用于返回至“摩擦补偿”对话框。

4.8.4 示例：线形轴的摩擦补偿

前提条件

已经设置了 X 轴上用于激活摩擦补偿的机床数据：

- MD32490 = 3
- MD32510 = 1

根据以下设置测量 X 轴：

- SD55820 \$SCS_FRICT_OPT_RADIUS = 2 mm 半径
- SD55822[0...8] \$SCS_FRICT_OPT_FEED 进给速度见下表

X 轴测量

步骤:

1. 通过软键“NC”和“摩擦补偿”选择优化对话框。
2. 选中 X 轴，然后按下软键“启动”。
3. 不再输入时，按下“确认”软键确认程序编辑器。X 轴不与分开的反向点一同测量。
4. 修改参数，直到对补偿满意。

4.8 带自适应特性曲线的摩擦补偿

5. 按下“下一步 >” 切换至下一步。

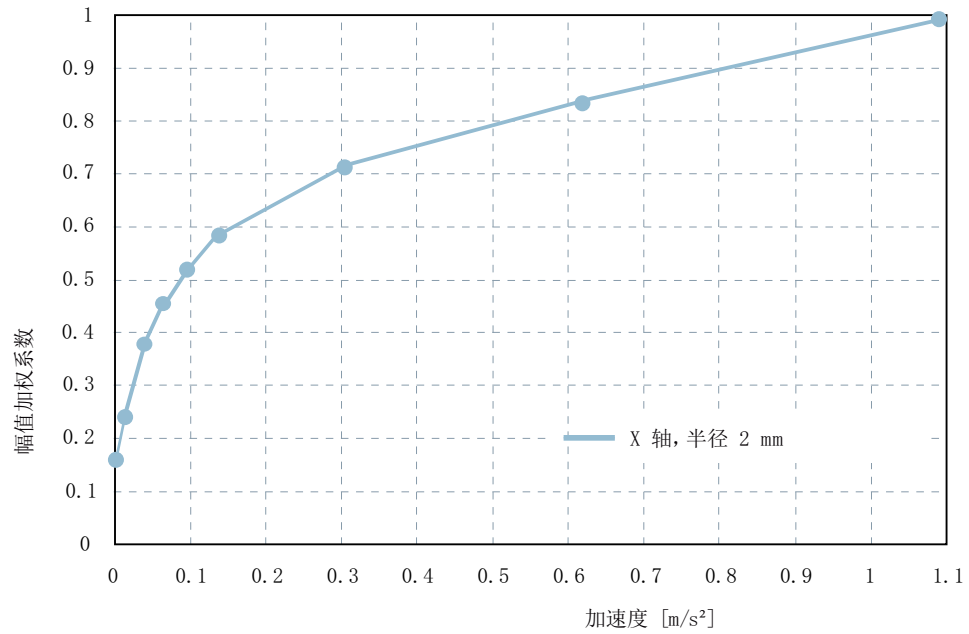
步骤 1/9: 1000 mm/min	振幅: 17,25 mm/min 失效时间: 0.043 s 生效时间: 0.016 s
步骤 2/9: 800 mm/min	振幅: 15,2 mm/min 失效时间: 0.046 s 生效时间: 0.014 s
步骤 3/9: 600 mm/min	振幅: 14.050 mm/min 失效时间: 0.026 s 生效时间: 0.016 s
步骤 4/9: 400 mm/min	振幅: 11.250 mm/min 失效时间: 0.030 s 生效时间: 0.016 s
步骤 5/9: 200 mm/min	振幅: 7.500 mm/min 失效时间: 0.035 s 生效时间: 0.016 s
步骤 6/9: 100 mm/min	振幅: 4.900 mm/min 失效时间: 0.050 s 生效时间: 0.016 s
步骤 7/9: 1500 mm/min	振幅: 22.175 mm/min 失效时间: 0.027 s 生效时间: 0.018 s
步骤 8/9: 2100 mm/min	振幅: 25.000 mm/min 失效时间: 0.015 s 生效时间: 0.018 s
步骤 9/9: 2800 mm/min	振幅: 30.000 mm/min 失效时间: 0.010 s 生效时间: 0.016 s

6. 重复该过程，直到 9 个步骤都完成。

7. 按下软键“确认” 并保存数据。

结果

下图为适配振幅值与各个加速度之间的单调递增函数图：



4.8 带自适应特性曲线的摩擦补偿

调试驱动

5.1 调试驱动

5.1.1 配置示例 - 概览

配置概览

使用 SINUMERIK Operate 最基本的调试流程对于 SINAMICS S120 Combi 和 SINAMICS S120 书本型来说是相同的。

以下示例显示了配置步骤：

示例 1： 不带 NX 的 SINAMICS S120 Combi:

- 自动配置电源。
- 配置配备 1PH8 电机的主轴。
- 带 2 个编码器的 DO4：此处还需配置编码器 2：“用户自定义”
- 带 2 个编码器的 DO6，两者自动识别。

示例 2： 配置 SINAMICS S120 书本型的电源

配置驱动

DRIVE-CLiQ 连接必须符合“DRIVE-CLiQ 拓扑规则 (页 154)”一节中的拓扑规则。在缺省设置中，DRIVE-CLiQ 的连接顺序和 SINAMICS 驱动对象号的顺序一致：

轴	SINAMICS 驱动对象	
	编号 (DO)	名称
---	1	Control Unit: 控制单元
---	2	电源模块
MSP1	3	主轴
MX1	4	伺服 1
MY1	5	伺服 2
MZ1	6	伺服 3

轴	SINAMICS 驱动对象	
	编号 (DO)	名称
MA1	7	伺服 4
---	11	集线器模块
---	9	TM54F 主站

配置示例

以下示例以以下配置为基准：

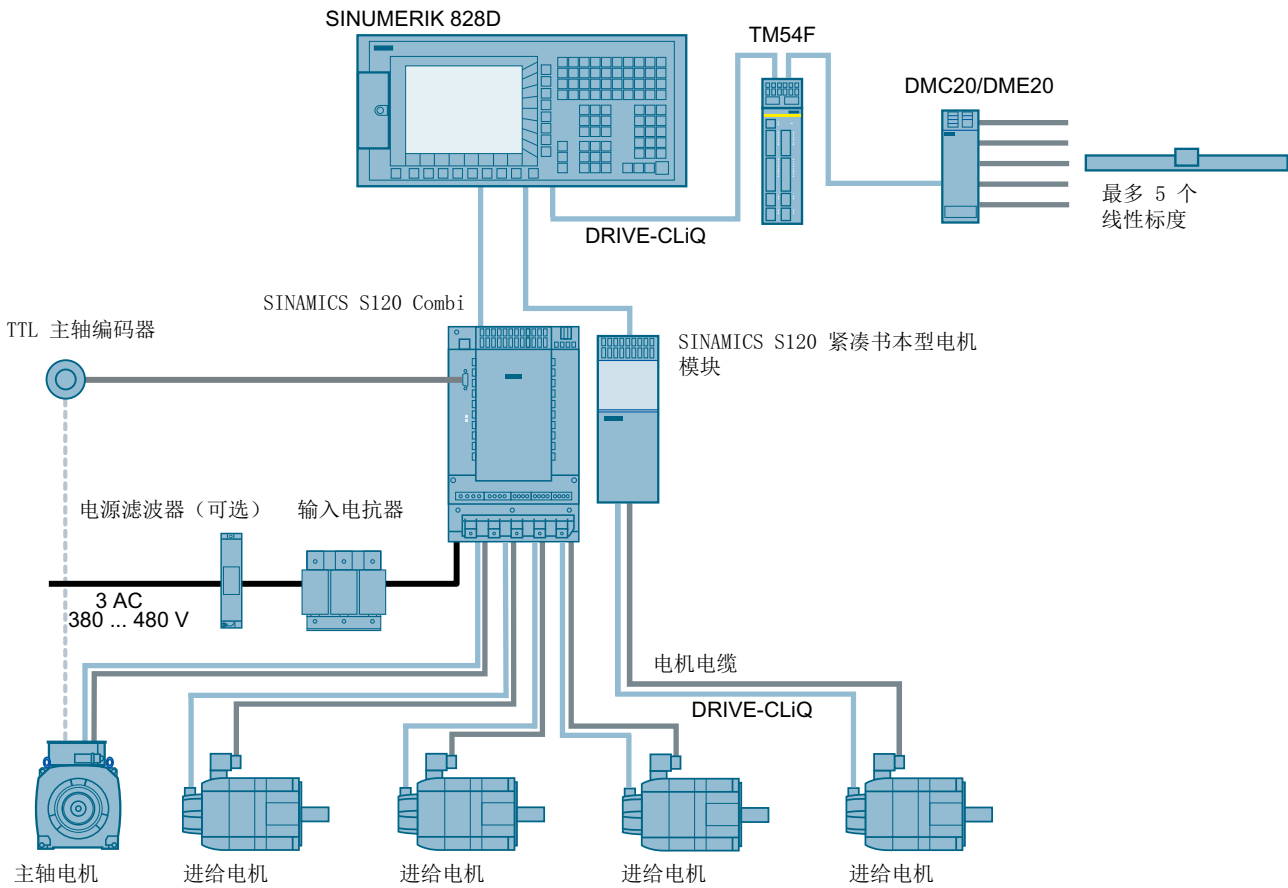


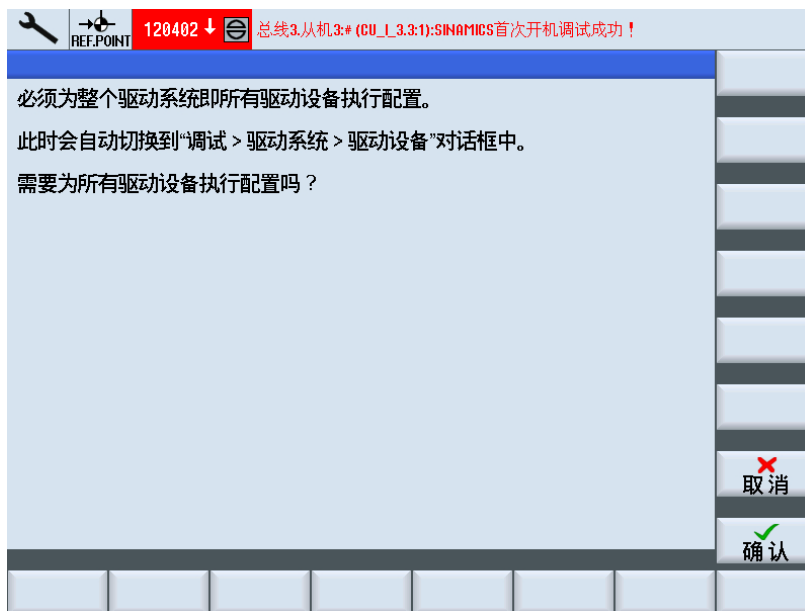
图 5-1 调试示例

5.1.2 自动首次调试

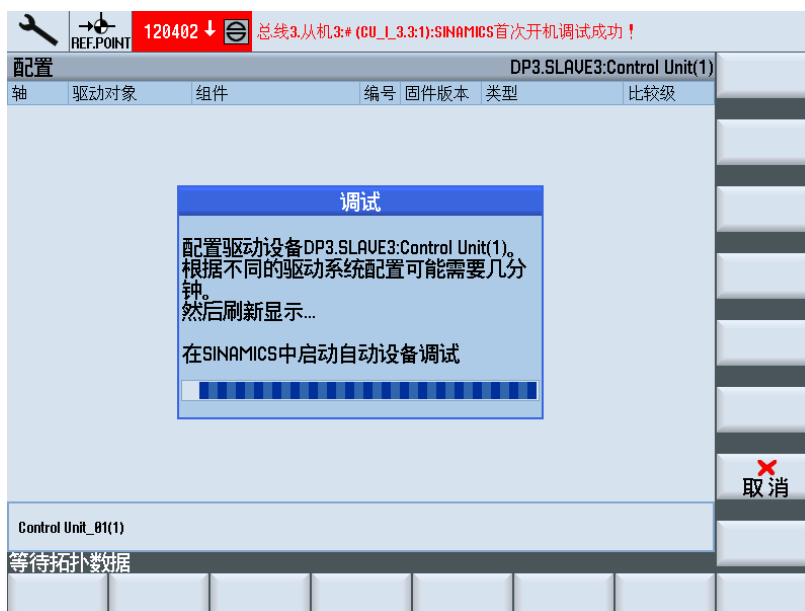
自动首次调试

操作步骤：

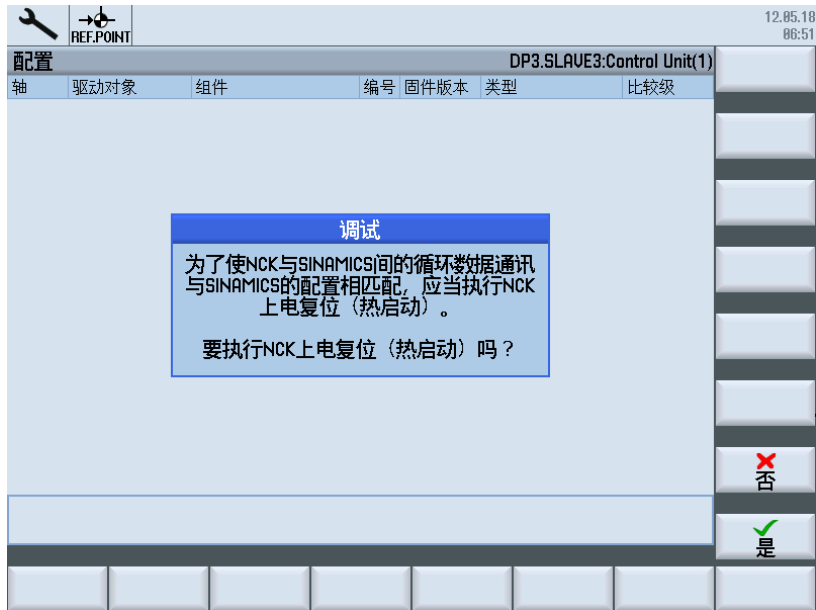
1. 在整个驱动系统启动完毕后，显示以下用于自动设备配置的对话框：“所有驱动系统（所有驱动设备）必须进行设备配置。”



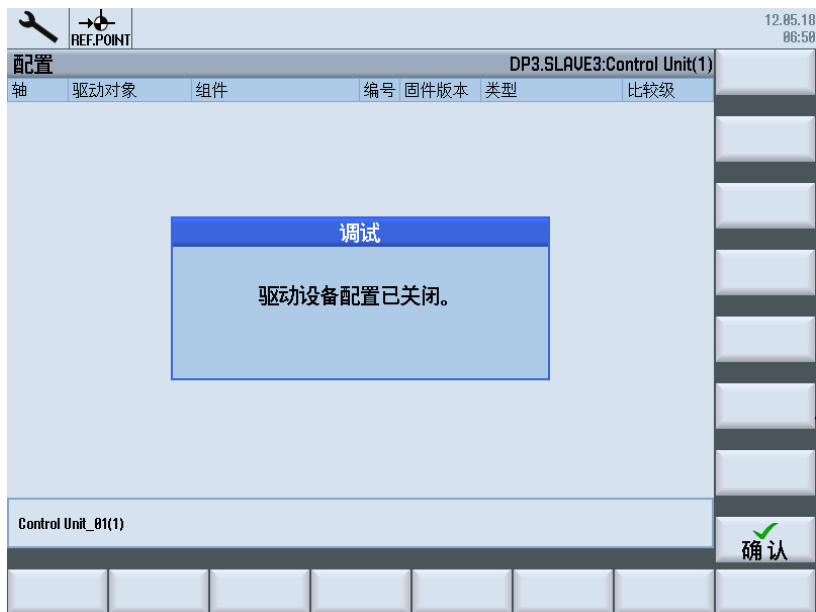
2. 按下“确认”，开始进行设备的自动配置。之后，系统将自动读取拓扑结构。该过程可能要持续几分钟。



3. 然后系统会要求热启动。点击“是”确认。该过程可能要持续几分钟。



4. 热启动后会得到拓扑结构概览，点击“确认”，继续进行调试。



说明

创建定义的输出状态

如果无法开始进行自动设备配置，则表明已经执行了配置：加载“出厂设置”，以创建定义的状态或再启动一次。然后，定义的输出状态便再次可用。

首次调试后的配置

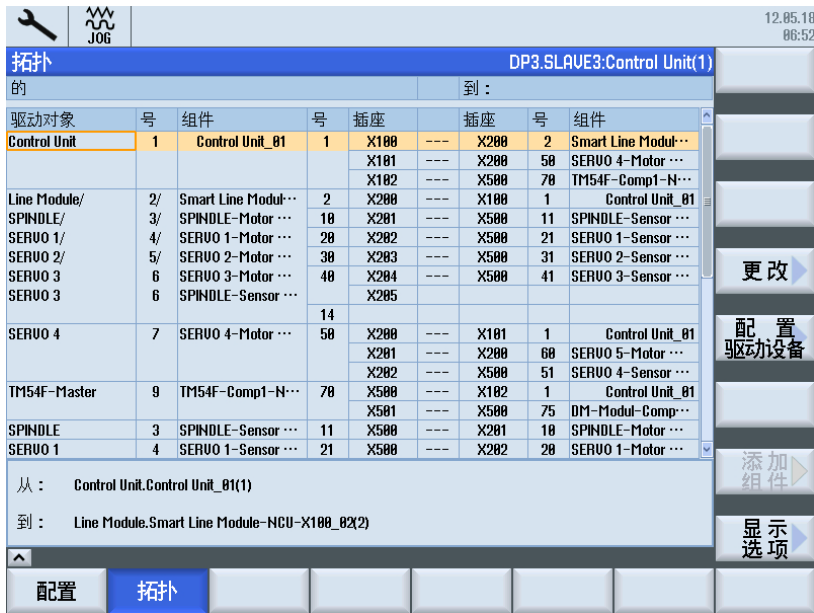
可在“配置”窗口中获得已选择的驱动设备的已连接的组件列表。您可以从中看出各组件与驱动对象间的分配关系，比如电机模块、电机和编码器分配至驱动对象以及分配至 NC 加工轴（根据轴分配）。

轴	驱动对象	组件	编号	固件版本	类型	比较级
	Control Unit	Control Unit_01	1	99887766	028.20	高
	Line Module	Smart Line Module-NC...	2	4583802	Smart Line Module f...	高
	SPINDLE	SPINDLE-Motor Modul...	10	4583802	MM_COMBI_4AXIS...	高
	SPINDLE	SPINDLE-Sensor Modul...	11	4583802	SMx module sin/cos	高
	SPINDLE	SPINDLE-Sensor Modul...	14	4583802	SMC30	高
	SERVO 1	SERVO 1-Motor Modul...	20	4583802	MM_COMBI_4AXIS...	高
	SERVO 1	SERVO 1-Sensor Modul...	21	1322530	SMI20/DQI	高
	SERVO 1	SERVO 1-Sensor Modul...	24	4583802	SMx module sin/cos	高
	SERVO 2	SERVO 2-Motor Modul...	30	4583802	MM_COMBI_4AXIS...	高
	SERVO 2	SERVO 2-Sensor Modul...	31	1322530	SMI20/DQI	高
	SERVO 2	SERVO 2-Sensor Modul...	34	4583802	SMx module sin/cos	高
	SERVO 3	SERVO 3-Motor Modul...	40	4583802	MM_COMBI_4AXIS...	高
	SERVO 3	SERVO 3-Sensor Modul...	41	1322530	SMI20/DQI	高
	SERVO 3	SERVO 3-Sensor Modul...	44	4583802	SMx module sin/cos	高
	SERVO 4	SERVO 4-Motor Modul...	50	4583802	MM_1AXIS_DCAC	高
	SERVO 4	SERVO 4-Sensor Modul...	51	1322530	SMI20/DQI	高
	SERVO 4	SERVO 4-Sensor Modul...	54	4583802	SMx module sin/cos	高
	SERVO 5	SERVO 5-Motor Modul...	60	4583802	MM_1AXIS_DCAC	高
	SERVO 5	SERVO 5-Sensor Modul...	61	1322530	SMI20/DQI	高
	SERVO 5	SERVO 5-Sensor Modul...	64	4583802	SMx module sin/cos	高

检查首次调试后的拓扑结构

可在“拓扑结构”窗口中显示已选驱动设备组件的 DRIVE-CLiQ 布线。组件已连接的接口分类显示在表格中。因此，可追踪每条 DRIVE-CLiQ 连接线，从控制单元或扩展模块 NX（比如至编码器）开始。

驱动对象的布线“from/to”（从/至）显示在表格的下方。



5.1.3 示例：检查电源配置（S120 Combi）

检查电源配置

驱动对象已进行了调试。以下概览（两页）中显示了自动计算出的特性数据：电源滤波器数据、电压测量模块（VSM）数据和电源电压数据。



图 5-2 S120 Combi 电源（第 1 页）

显示在“电源数据”对话框中输入的设备连接电压：该电压为报警阈值和断路阈值电源电压监控（p0281 - p0283）的基础。

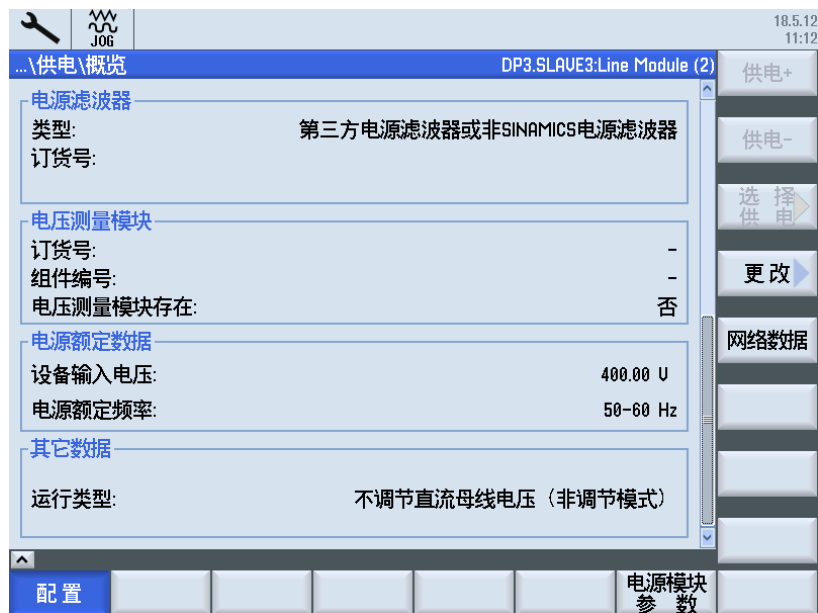


图 5-3 S120 Combi 电源（第 2 页）

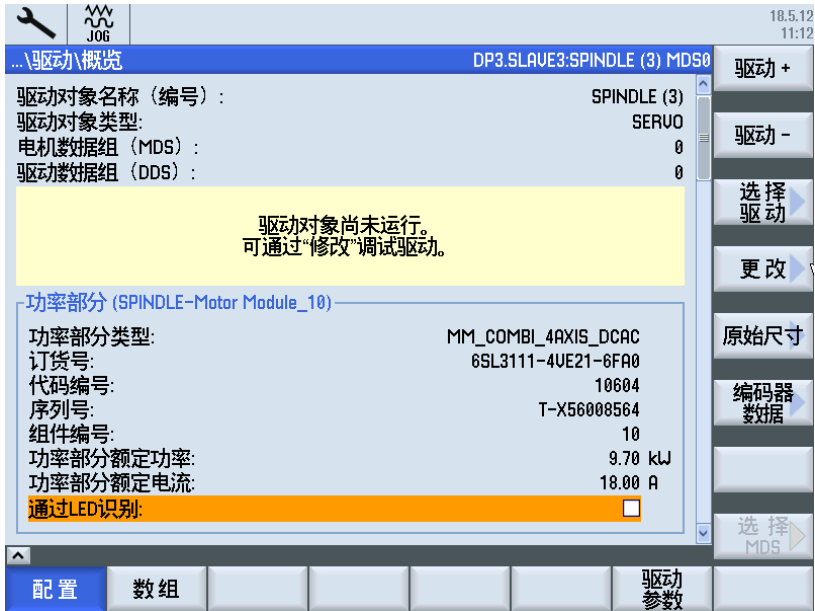
5.1.4 示例：配置主轴（DO3）

主轴配置

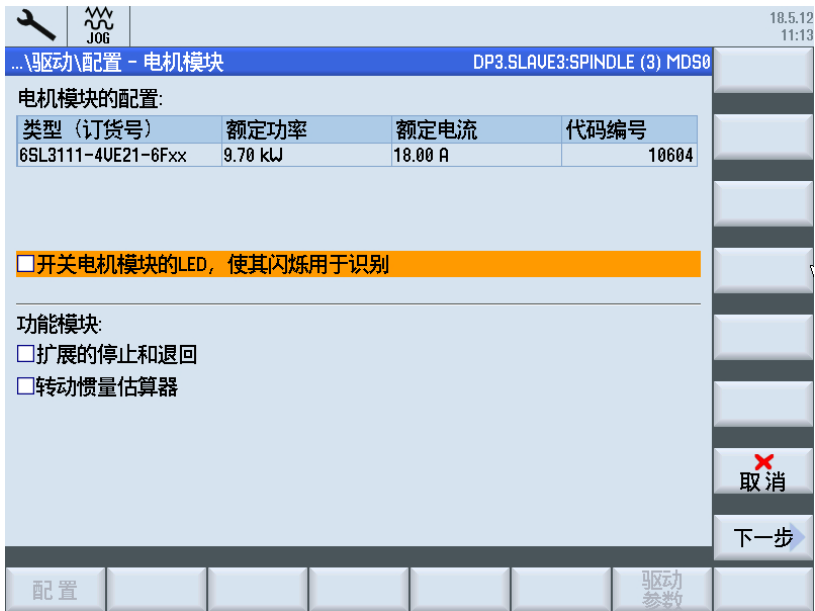
由于主轴（驱动对象 DO3）不是由 SMI 电机驱动的，因此需要手动调试，也不能自动检测电机系统和测量系统。

操作步骤：

1. 选择“修改”，进行主轴配置：



2. 显示驱动对象（DO3）自动设备配置的特性数据：

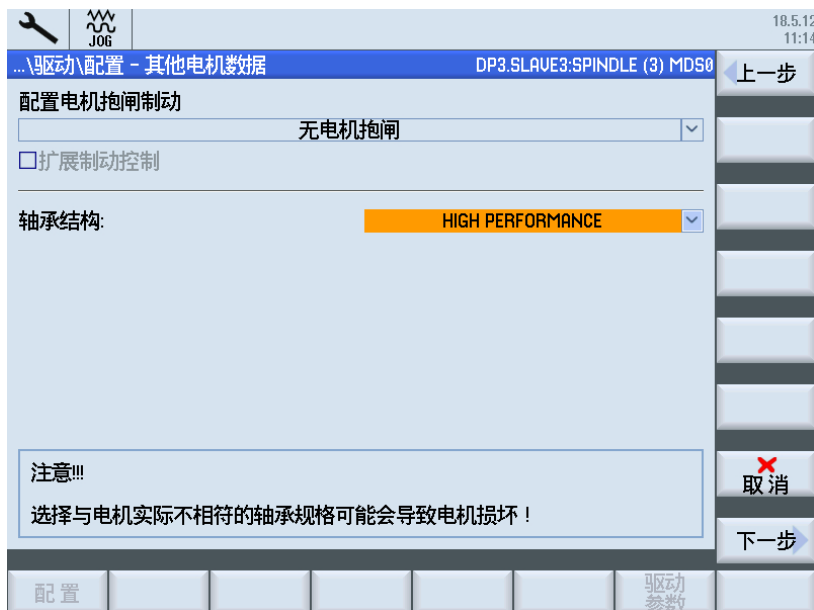


或者您也可以激活需要部分授权的功能模块。此时各个功能模块调试所需的其他参数生效。

3. 选择“下一步”>。选择“从列表中选择标准西门子电机”，进行电机配置。
首先选择电机类型。为了限制电机选择列表，要在“搜索”中输入文本，作为筛选条件。*
作为占位符。文本适用于列表的所有列内容，单位和连接方式除外。

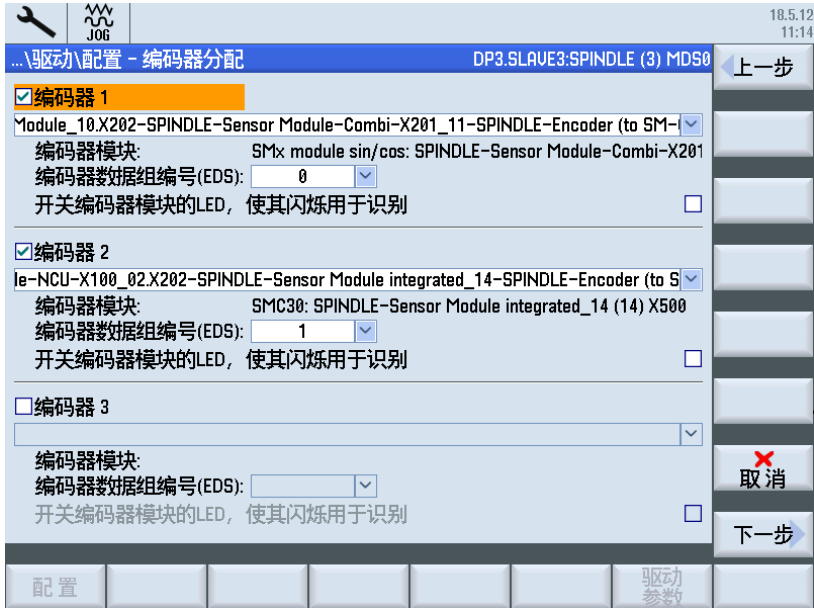


4. 选择“下一步”>。只有电机具有不同的轴承规格时，“轴承规格”选项才可选择。
在这里可设置轴承类型：这样可将转速限制在轴承的最大转速以内，从而避免发生机械损坏。选择“无说明”，不监控轴承的最高转速。

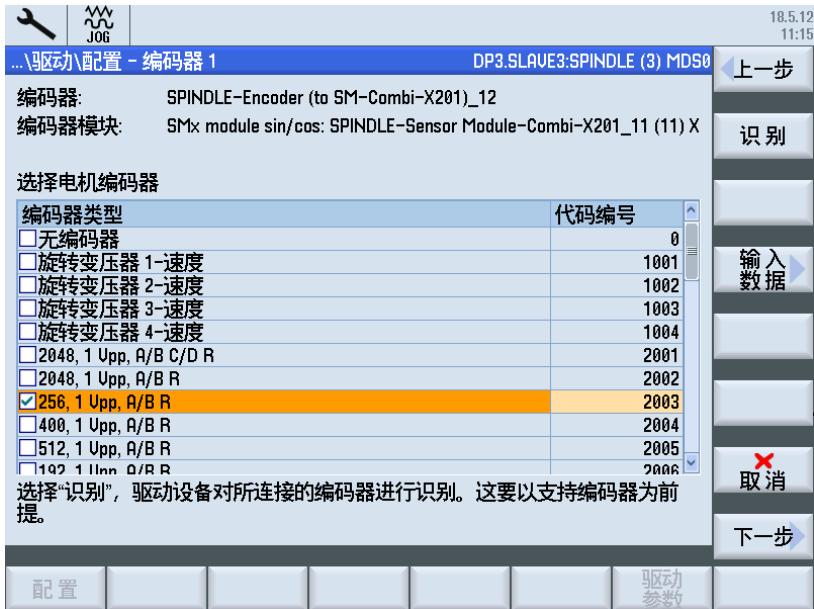


5.1 调试驱动

- 选择“下一步”>。此时可为所选的 SERVO 电机最多分配三个编码器：
如果 SERVO 电机上的编码器模块（SMx）已经正确插入，则会自动进行编码器分配。也可以自动分配多个测量系统。



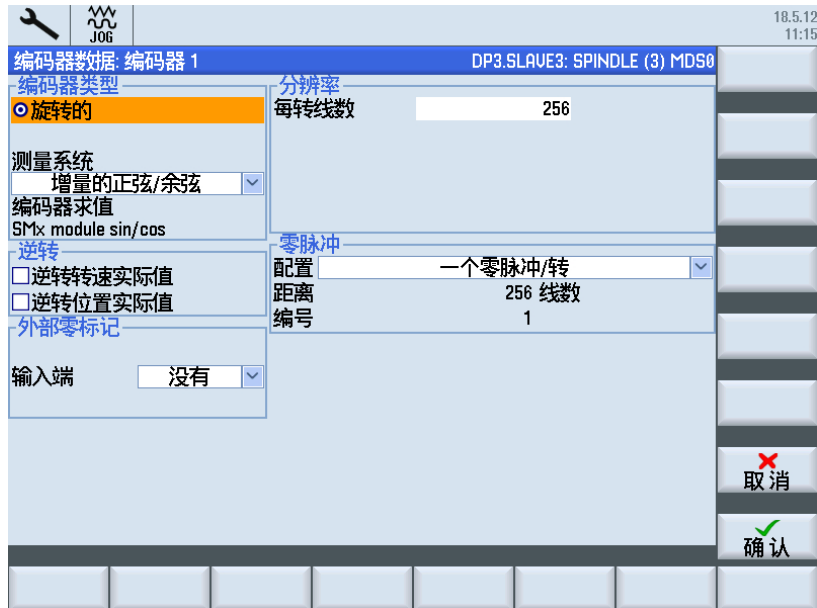
- 选择“下一步”>。



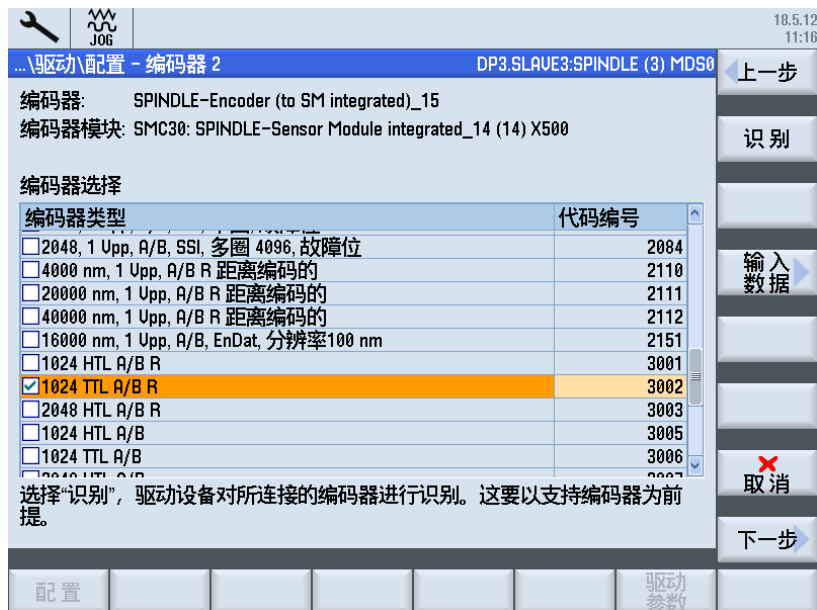
此处显示编码器名称和编码器模块，通过该编码器模块连接编码器。

- **方案 1:** 如使用 EnDat 和 DRIVE-CLiQ 则自动选择，无需其他设置。要检查预设置，请按下“输入数据”软键。
- **方案 2:** 未自动识别编码器。如果列表中存在已连接的编码器，则根据名称和编号选择编码器。否则，选择一个类似的编码器，以便输入编码器数据。

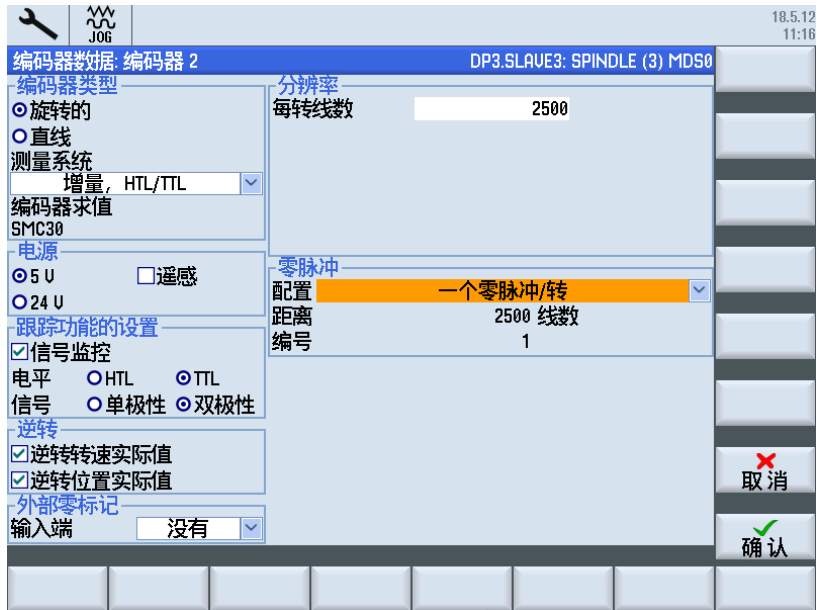
7. 选择“输入数据 >”。



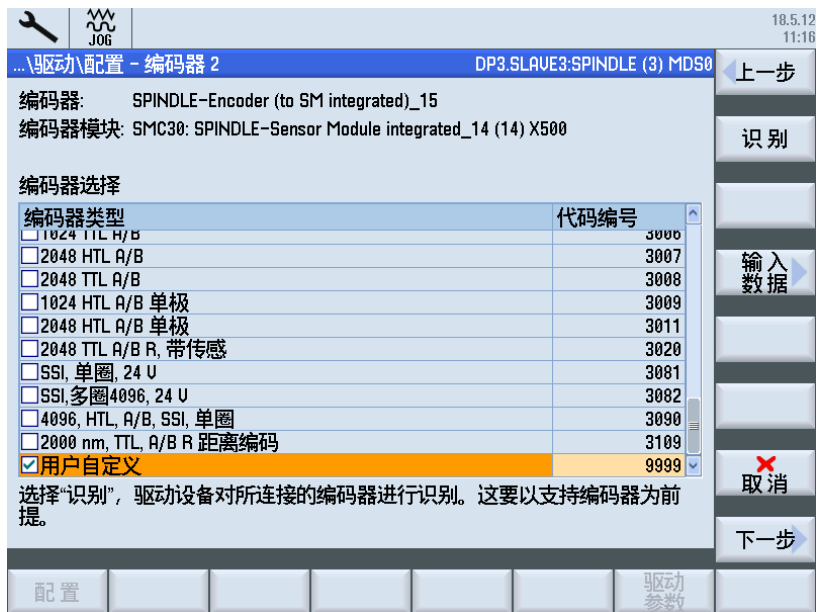
8. 按下“确认”，确认编码器数据。



9. 选择“输入数据 >”。

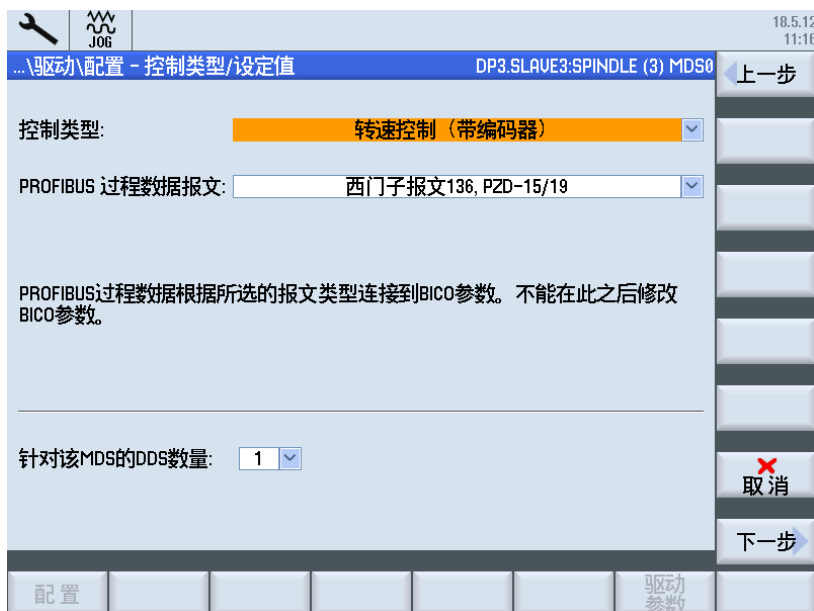


10. 按下“确认”，确认编码器数据。



11. 选择“下一步”>。

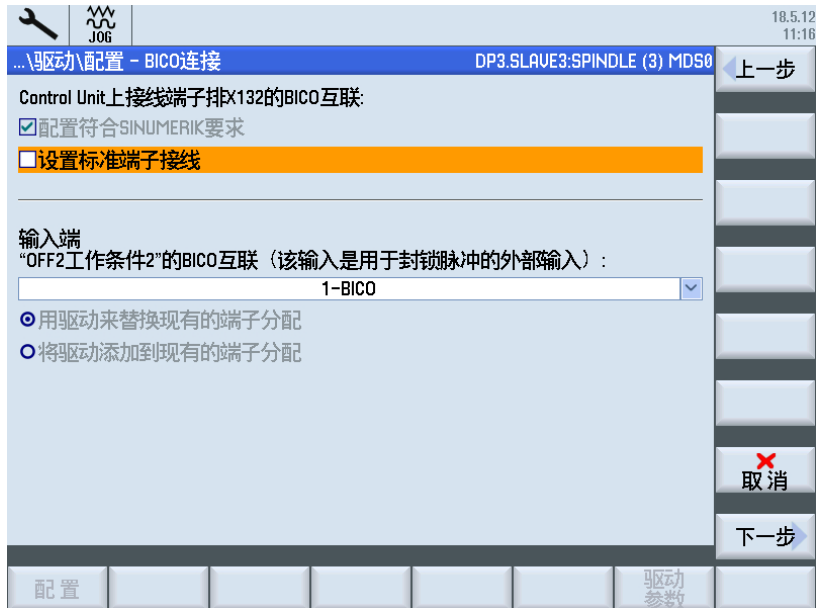
缺省设置：控制方式、PROFIBUS 报文和 DDS 的数量一般已正确设置。



在此设置控制方式、PROFIBUS 报文类型以及驱动数据组 DDS 的数量。

- 控制方式：
 - 如果 SERVO 电机未分配到电机编码器，则系统自动设置为“转速控制（不带编码器）”。
 - 如果 SERVO 电机已分配到电机编码器，则系统自动设置为“转速控制（带编码器）”。
- 报文类型：默认设置是报文 136：修改报文类型时还必须调整 PLC 中的设置。
- 数据组：DDS 的数量默认设为 1。当要进行驱动数据组切换时，可根据需要创建多个 DDS。需要注意的是：DDS 数量过高会降低 SINAMICS 的系统启动速度。以后也可以在数据组向导中创建其他 DDS 和 MDS。

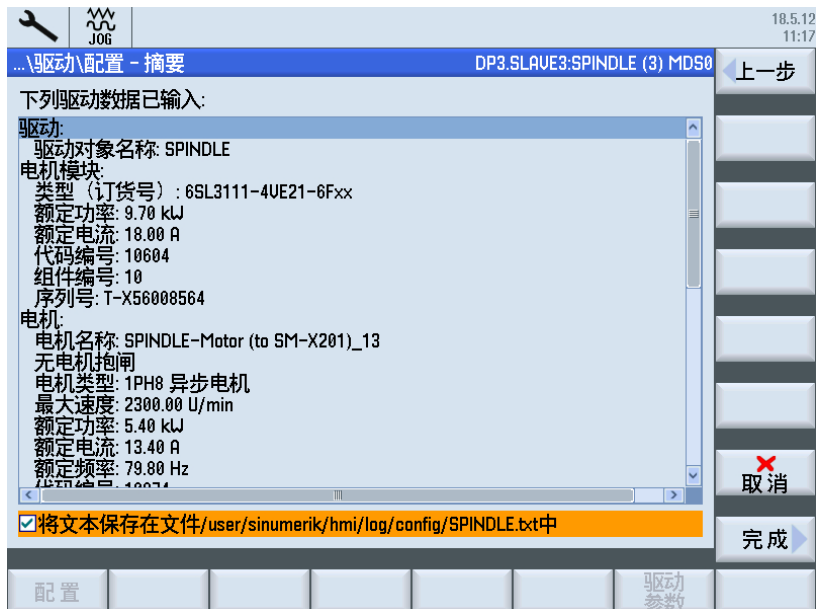
12. 选择“下一步”>。



以下两个复选框可进行如下选择：

- **缺省设置：“符合 SINUMERIK 的互联，用于端子布线”**
即：端子布线的 BICO 互联设置能确保所有使能和反馈信息能正确通讯。
如未勾选该复选框，则所有对缺省互联作出的修改生效。如果不希望这些修改生效，可以勾选“设置缺省端子布线”，恢复符合 SINUMERIK 的互联。
- **选项“设置用于标准端子布线的 BICO 互联”**
激活该选项，设置标准端子布线。即：配置符合 SINUMERIK 的互联。

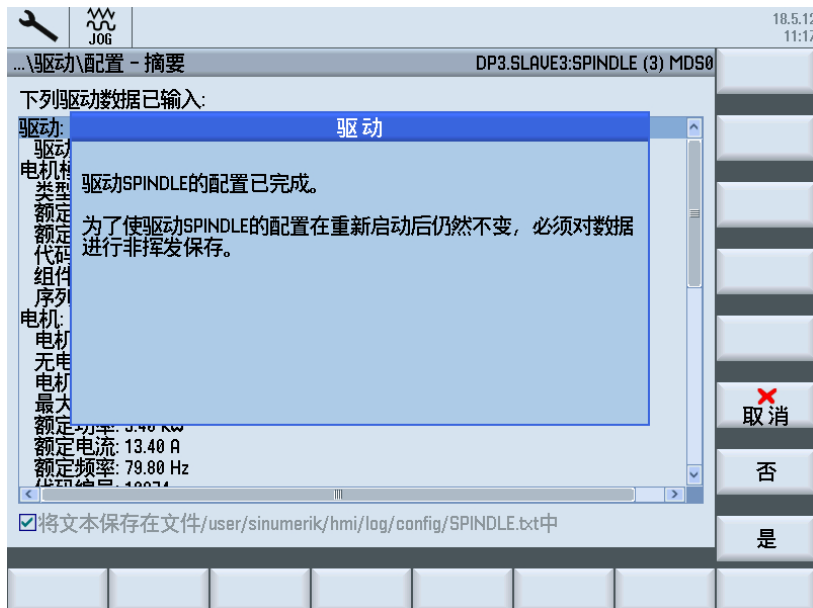
13. 选择“下一步”>。



概要中显示了所有用于配置驱动对象的数据。

选项：勾选“将文本保存至文件”，将配置数据保存到文本文件中。

14. 选择“完成 >”，结束驱动对象的调试。修改时会弹出一个询问是否保存的对话框。



15. 按下“是”，数据在上电/断电后仍会保留。
数据保存在 CF 卡上，保存在含有驱动数据的调试存档中。



由此结束主轴调试。

5.1.5 示例：配置轴（DO4）编码器

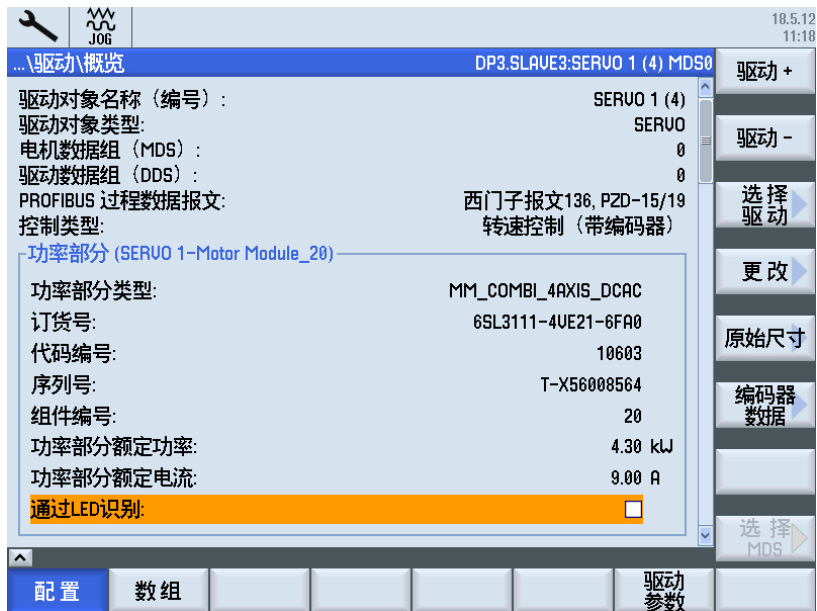
配置轴编码器 2

驱动对象 DO4 在自动设备配置时已进行了调试。

- 已检测到编码器 1。
- 编码器 2 不在选择列表中，还必须进行配置。

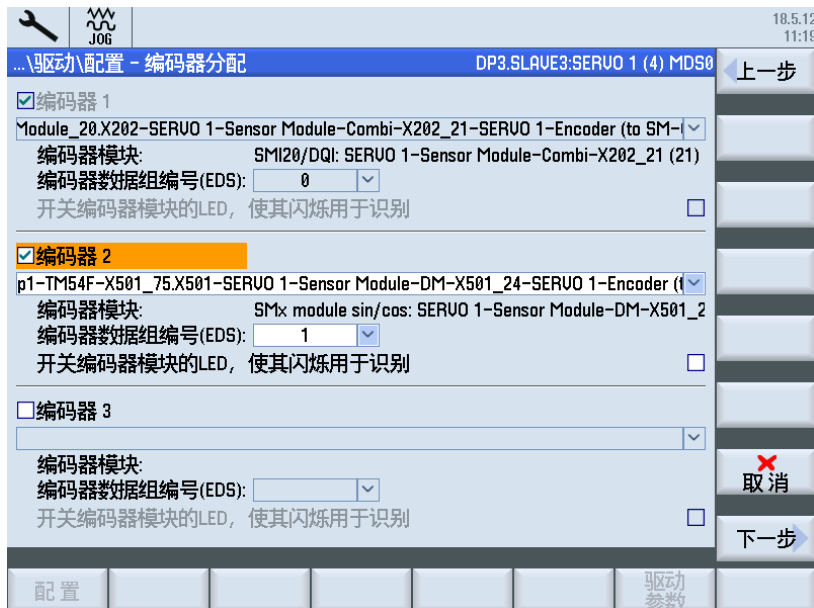
操作步骤：

1. 参见以下示例：

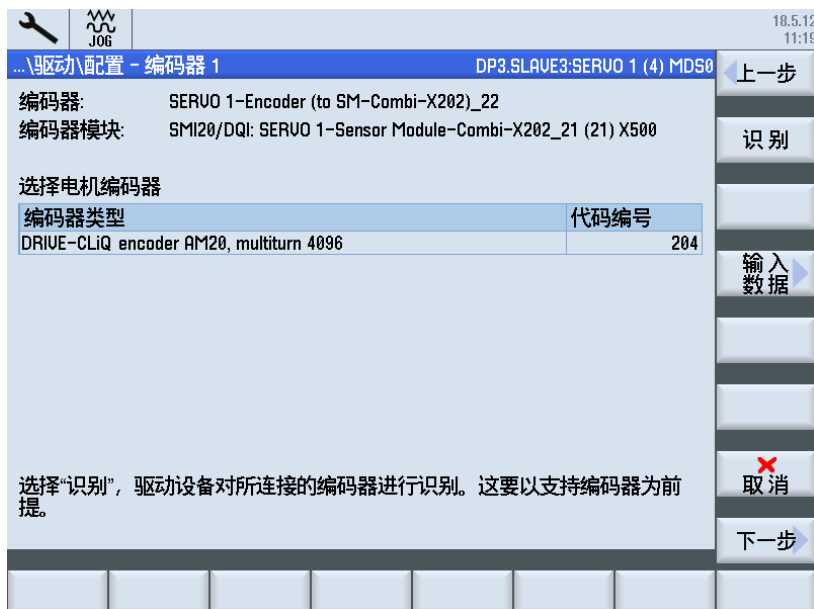


2. 选择“修改 >”。

3. 选择“下一步 >”直至出现“编码器分配”对话框：



4. 选择“下一步 >：已检测到编码器 1”。
此处显示编码器名称和编码器模块，通过该编码器模块连接编码器。



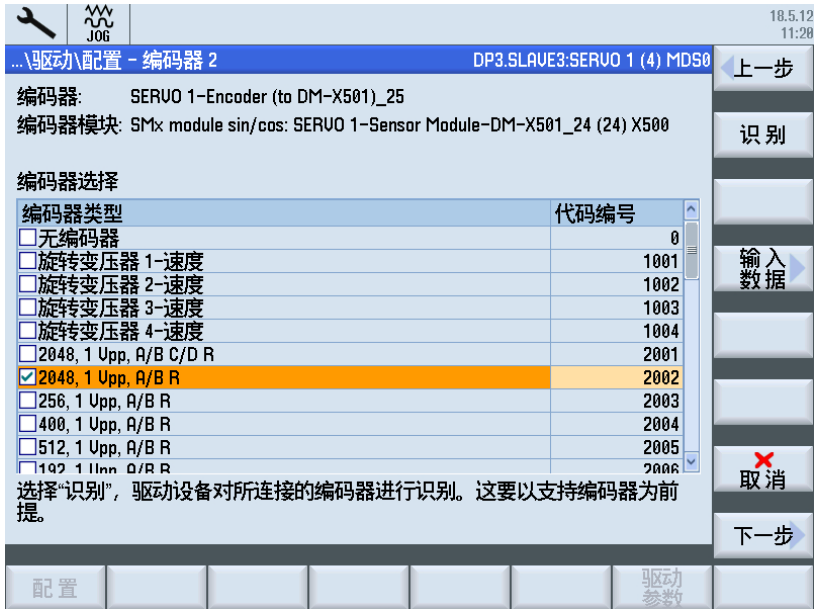
说明

编码器识别

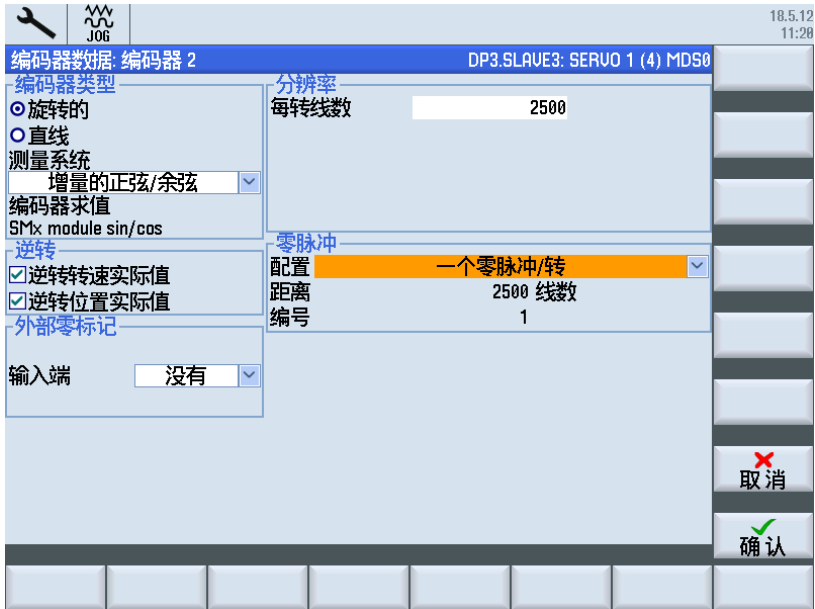
通过软键“识别”重新进行编码器识别，以检测所显示的编码器是否与实际连接的编码器吻合。只能识别到绝对值编码器或 DRIVE-CLiQ 编码器。

5.1 调试驱动

- 选择“下一步”>。
编码器 2 未能自动检测（“无编码器”）必须进行配置：
如果列表中存在已连接的编码器，则根据名称和编号选择编码器。否则，从列表选择一个类似的编码器并按下软键“输入数据”，以便输入编码器数据。
- 选择以下编码器类型：编号 2002。

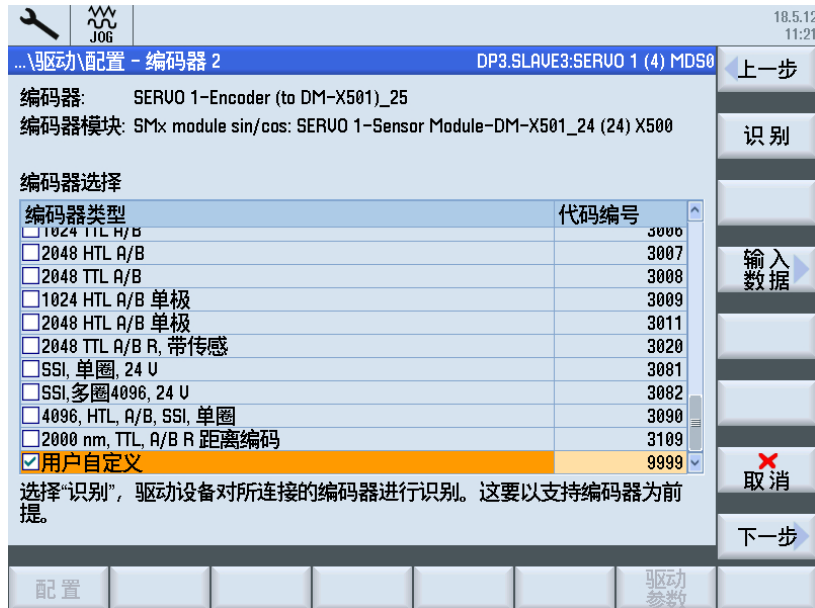


- 选择“下一步 >”并输入编码器数据：

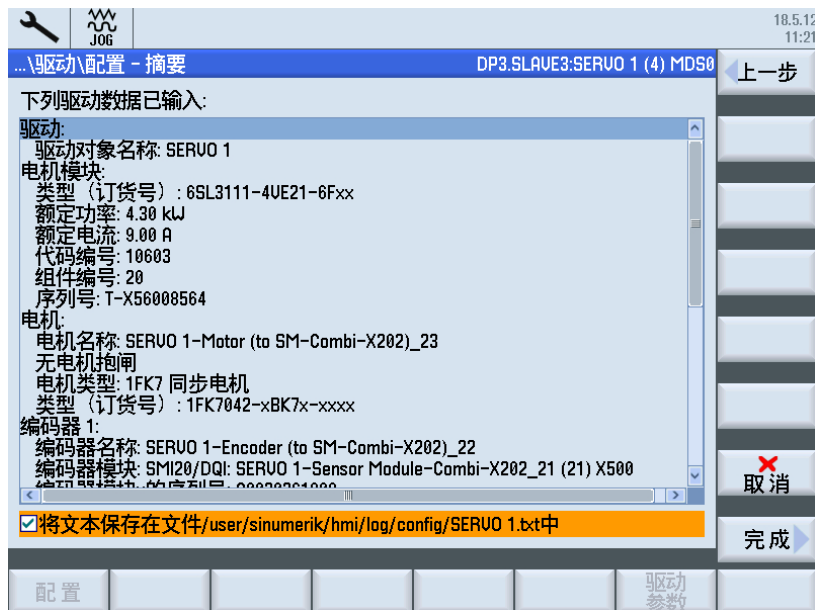


取反：转速和位置实际值取反。
线数/转：2500
按下“确认”。

8. 编码器 2 已配置并显示为“用户自定义”。

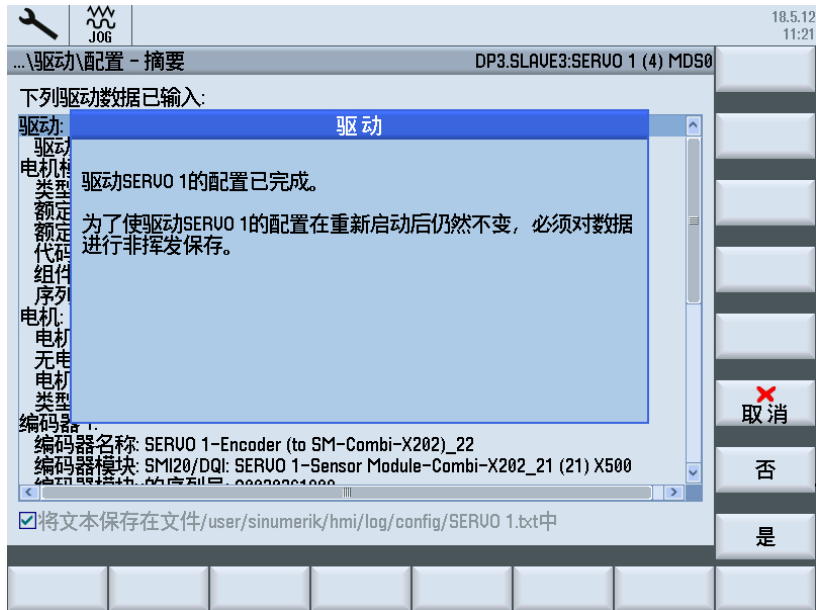


9. 选择“下一步>”，直至显示如下概要：



选项：勾选“将文本保存至文件”，将配置数据保存到文本文件中。

10. 选择“完成>”，结束配置。



11. 选择“是”，对配置数据进行非易失性存储。

由此结束轴调试。

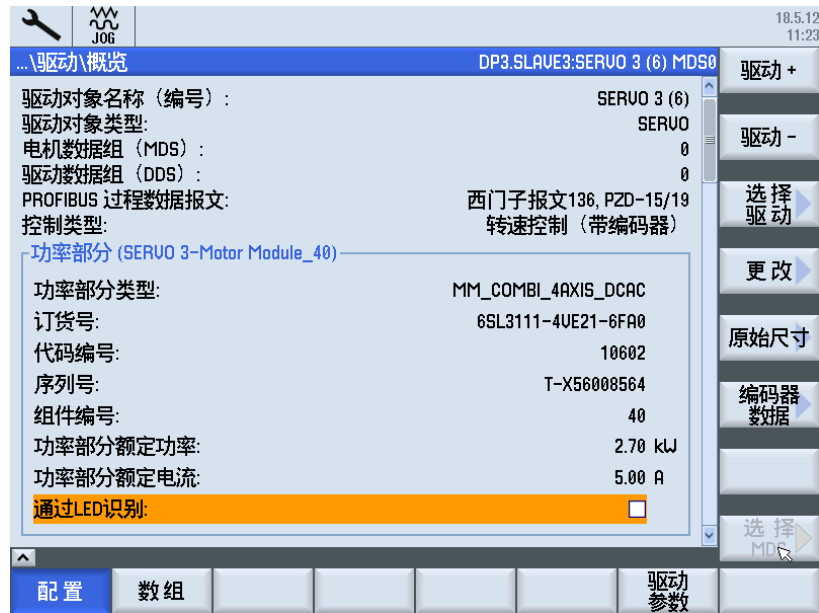
5.1.6 示例：检查轴（DO6）配置

SERVO DO6: 检测编码器

驱动对象 DO6 在自动设备配置时已进行了调试。两个编码器均在选择列表中，可以识别。

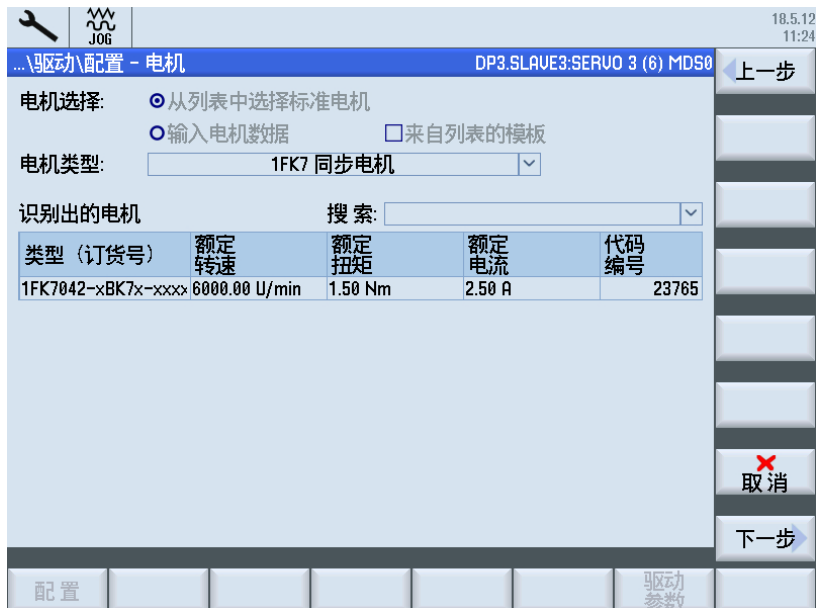
操作步骤:

1. 参见以下示例:

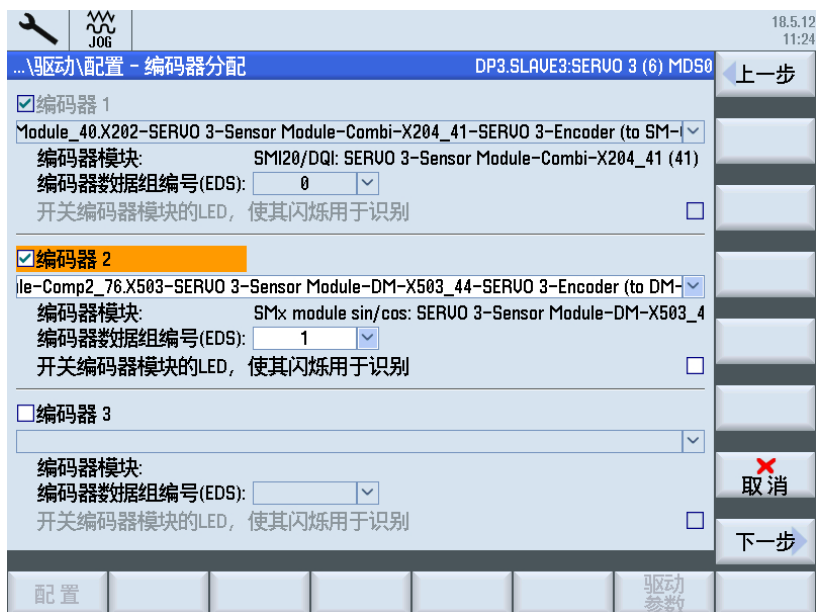


5.1 调试驱动

1. 选择“修改 >”。



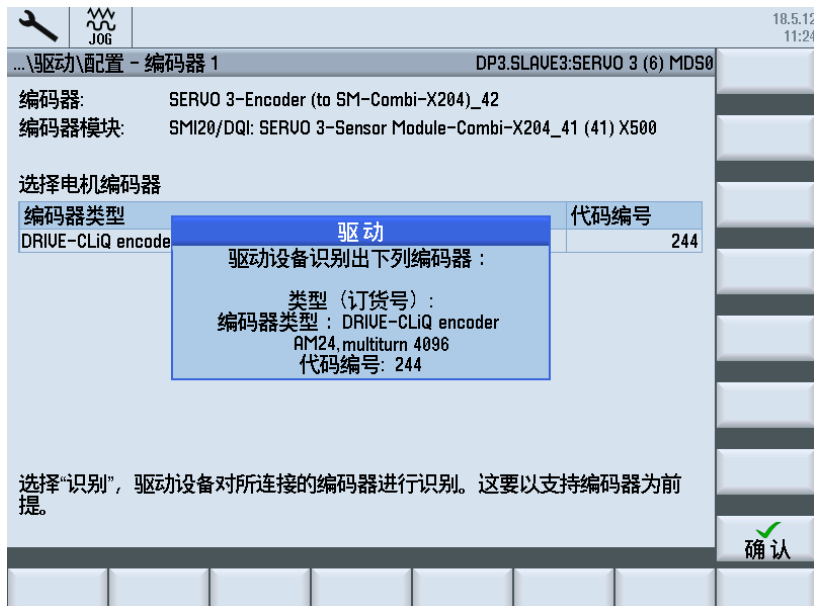
2. 选择“下一步 >”。



此时，SERVO 电机上的编码器模块（SMx）已经正确插入，会自动进行编码器分配。也可以自动分配多个测量系统。

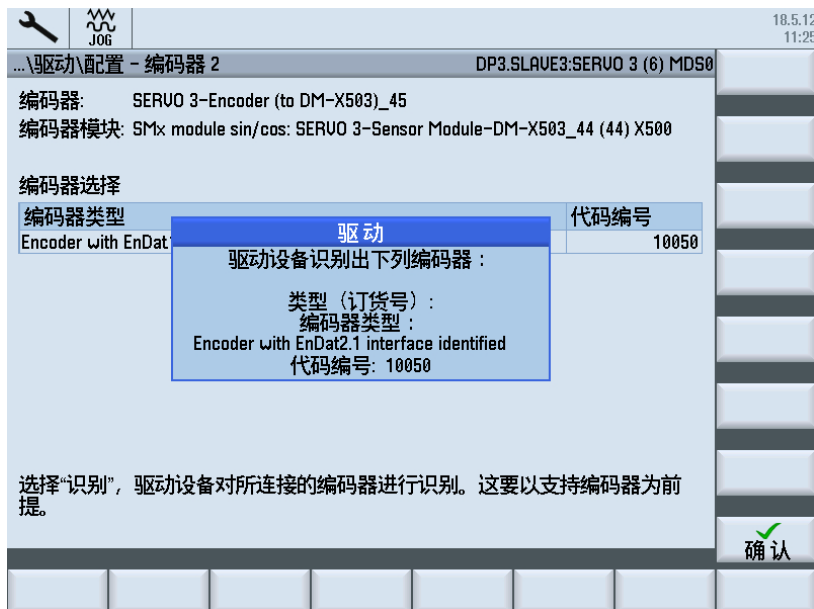
3. 选择“下一步 >”。

4. 按下编码器 1 的软键“识别”。

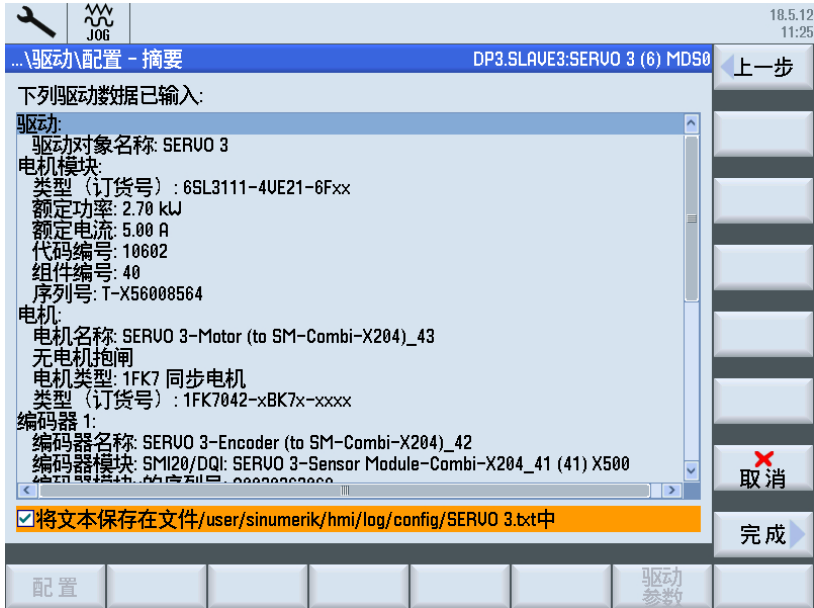


5. 选择“下一步 >”。

6. 按下编码器 2 的软键“识别”。

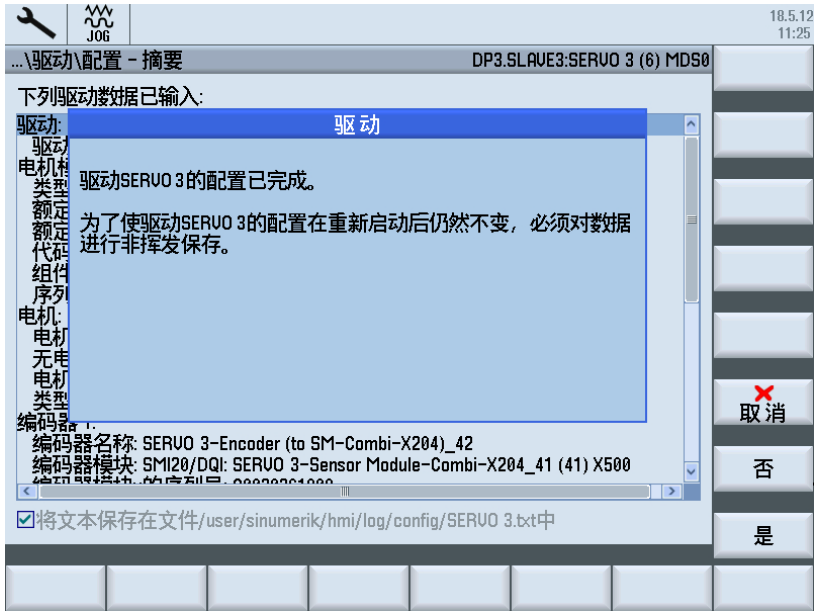


7. 按下“确认”，选择“下一步>”，直至显示如下概要：



选项：勾选“将文本保存至文件”，将配置数据保存到文本文件中。

8. 选择“完成”，结束配置。



由此结束轴调试。

5.1.7 示例：配置电源（S120 书本型）

配置电源（S120 书本型）

操作步骤：

1. 选择“修改”，进行电源配置：

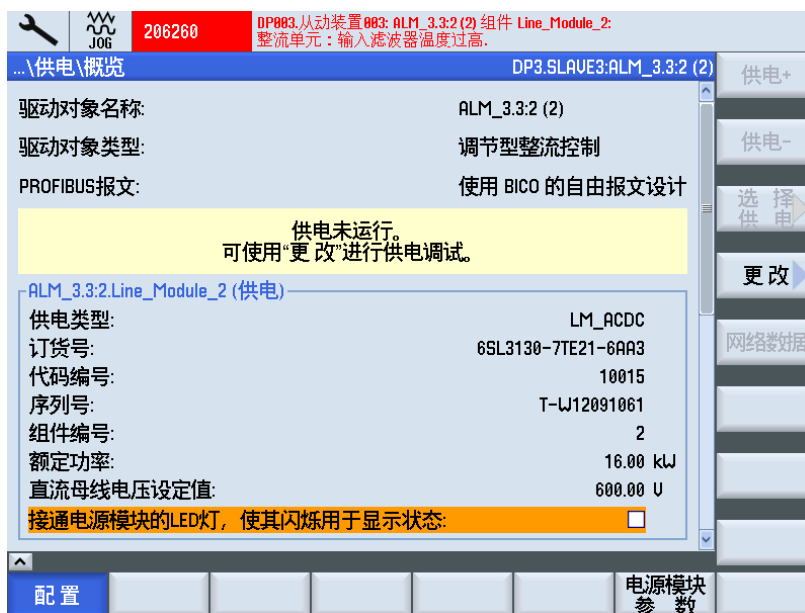


图 5-4 电源 > 概览

2. 显示自动设备配置时检测到的电源的特性数据：

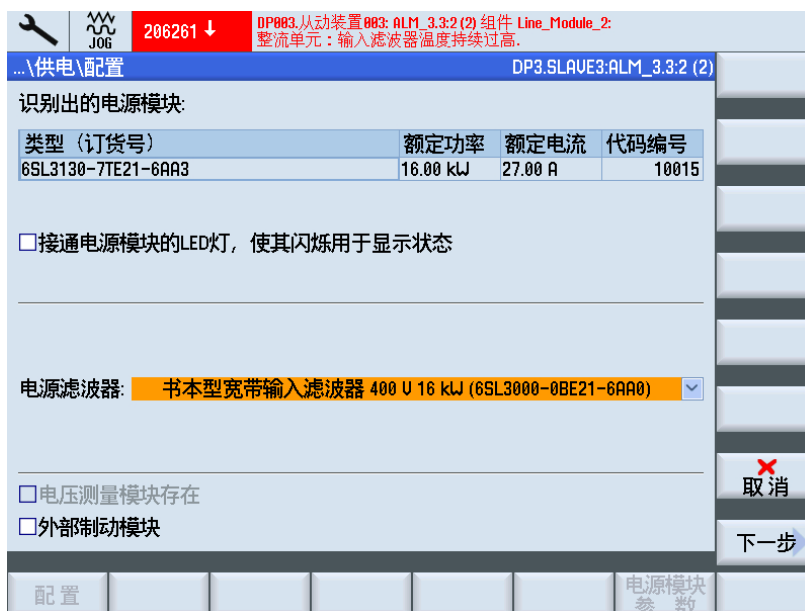


图 5-5 电源 > 配置

此外，还可激活以下功能：

- 勾选了选项“检测 LED 闪烁”后，所选择的电源 LED 会红绿交替闪烁。
- 在“电源滤波器”的列表中选择连接在电源前的滤波器类型。如果是带有温度监控（AIM = 调节型接口模块）的滤波器，则选择后会自动激活该滤波器（缺省设置）。如果在连接 AIM 之前已经因消除干扰发射和消除无线电干扰原因而连接了一个基础电源模块，则必须激活该选项。
- 检测到了电压测量模块后，该选项被选择。
- 如果激活了外部制动模块选件，则制动模块监控激活。此时也必须存在相应的端子布线（S120 书本型端子 X21.4 上的 p3866），否则系统会输出报警 206900。
- 如果设置了选项“激活外部功能模块”，则制动模块的功能模块激活，可用于模块监控。

此时需要设置相应的端子布线（S120 书本型端子 X21.4 上的 p3866），否则系统会输出报警 206900。此外，还需建立电源到 NCU 上一个可用端子的 BICO 互联。此处必须互联的参数参见参数手册 SINAMICS S120 中的制动模块功能图。

3. 选择“下一步 >”。

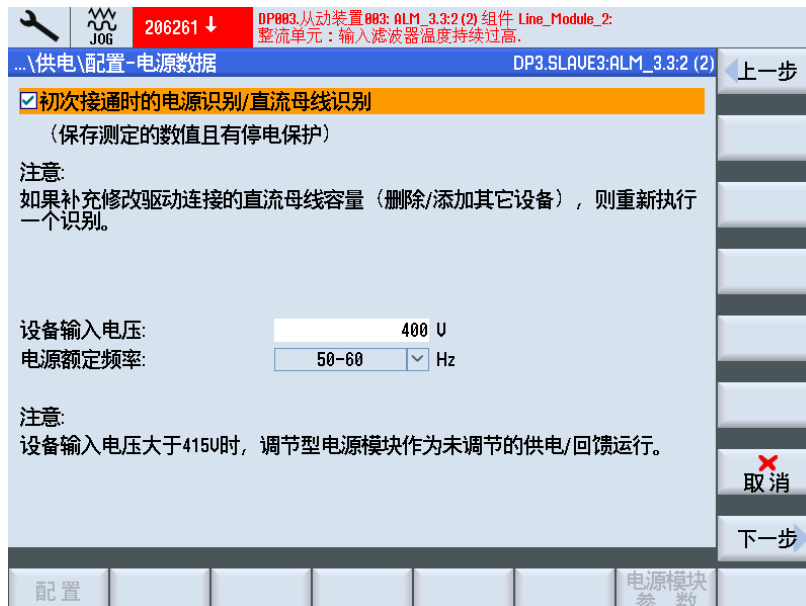


图 5-6 电源 > 电网数据

此处对以下电网数据进行配置：

- 选中此复选框时，在电源脉冲使能后会激活电网/直流母线识别（p3410）。之后电源进入运行状态。

说明

直流母线识别

如果电源环境或直流母线上的组件发生变化（例如：用户安装设备后或驱动组扩展后），则必须重新设置此复选框：因此“电网数据”软键也处于概览中，用于重新启动电网/直流母线识别。

若在调试存档中备份了 $p3410 = 5$ ，则会在读取存档后采用驱动数据自动启动电网/直流母线识别。

这样才能确保电源采用最佳的控制器设置工作。

- 输入设备输入电压：这是电源电压监控（p0281 - p0283）的基础，在该监控中会在过压或欠压时触发报警。（报警阈值和断路阈值）。实际电源电压会自动测算，并以此值为基准进行补偿。
- 电源的实际电源频率会自动测算。
- 请在参数 p0284 和 p0285 中设置阈值，超出阈值时会再次触发报警（监控预设：45 Hz 至 65 Hz）。

4. 选择“下一步 >”。



图 5-7 电源 > 端子布线

使用电网接触器，以确保驱动组和直流母线与电源电网之间的电流隔离。

电源的复选框是通过缺省设置选定的。即：首次调试时的 BICO 互联（ON/OFF，双向）是根据缺省设置设定的。勾选该复选框，便可以重新进行端子布线缺省设置。

如果在使用电源之前已经使用了一个外部电网接触器，则可通过 SINAMICS 自动对其进行控制。为此需勾选复选框“SINAMICS 内部电网接触器控制”。

或者通过 PLC 进行控制。通过所显示的预留端子进行连接。

– **X132.12:** 电网接触器控制输出端子电源

– **X132.4:** 电网接触器反馈信息输入端子电源

不可能进行 NCU 上驱动对象内部的电网接触器控制，因为电网接触器必须由外部 PLC 或硬件控制器进行控制。

5. 选择“下一步 >”。概要中显示了所有用于配置电源的数据。

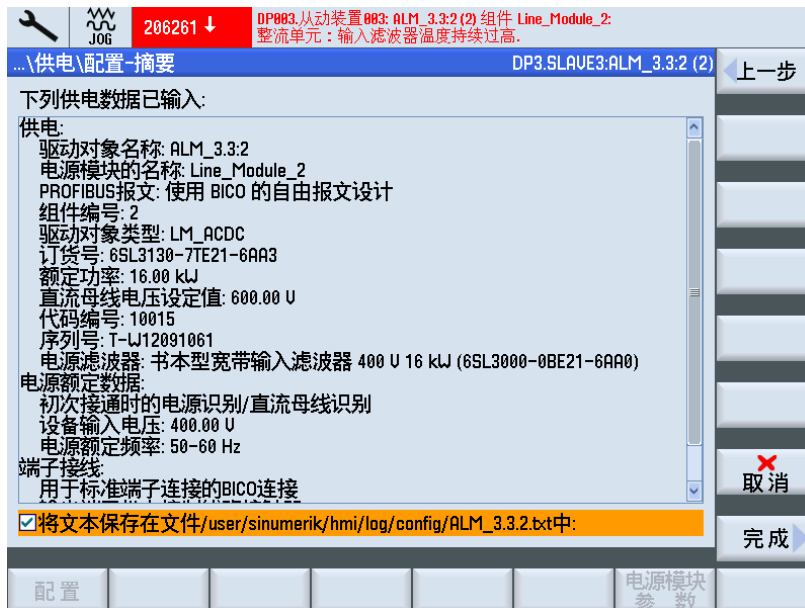


图 5-8 电源 > 概要

其它操作：

- 选项：将概要保存至文本文件。
- 软键“< 上一步”：返回上一步。
- 软键“完成”：结束调试。

6. 选择“完成”。修改后总是出现询问是否保存对话框：如果保存数据，则上电/断电后仍会保留该数据。数据保存在 CF 卡上，保存在含有驱动数据的调试存档中。

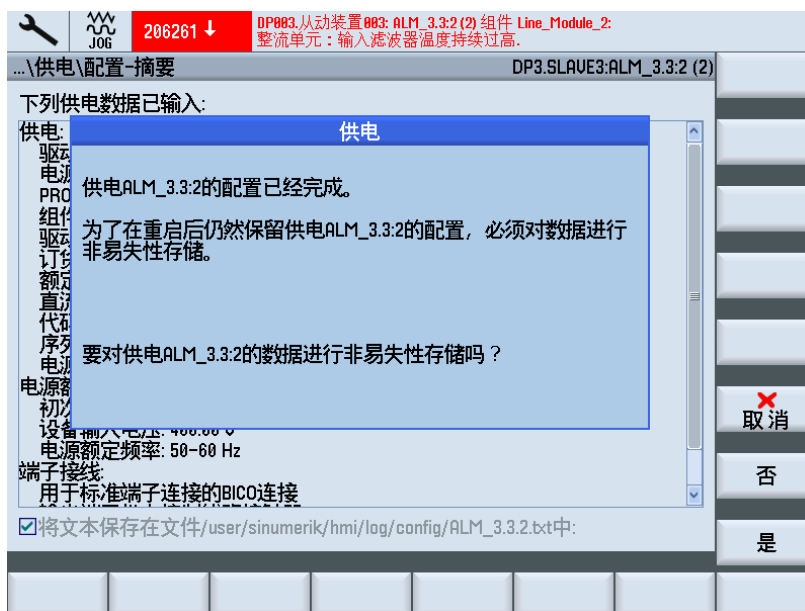


图 5-9 保存配置

7. 选择“是”

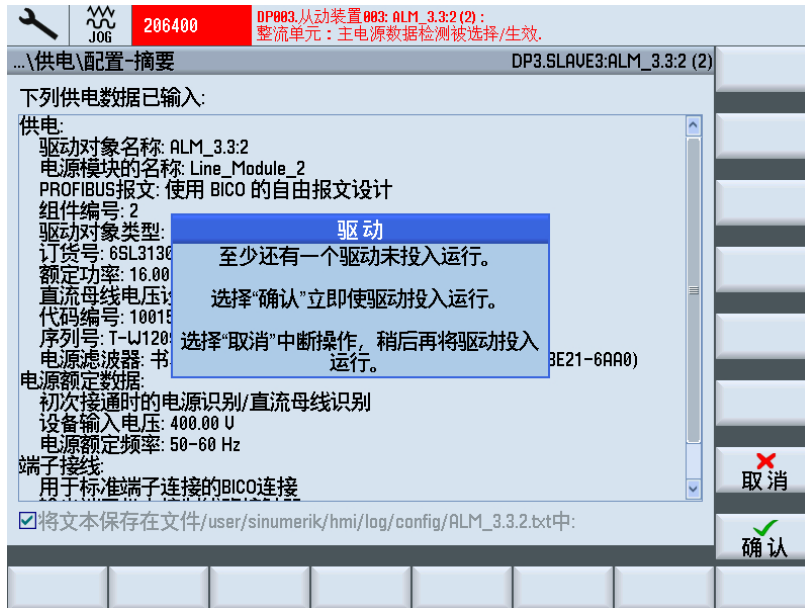


图 5-10 驱动图片

8. 选择“取消”：检测电源概览。

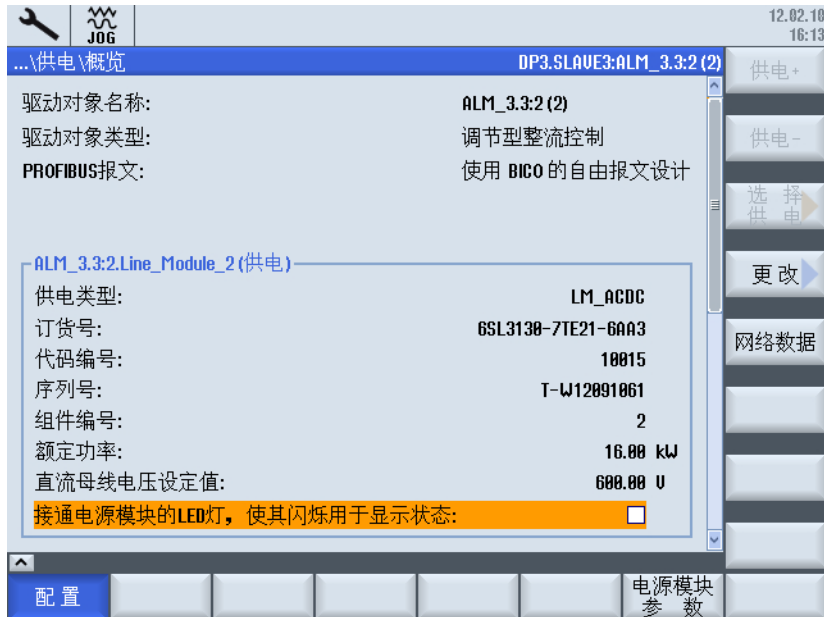


图 5-11 电源概览（第 1 页）

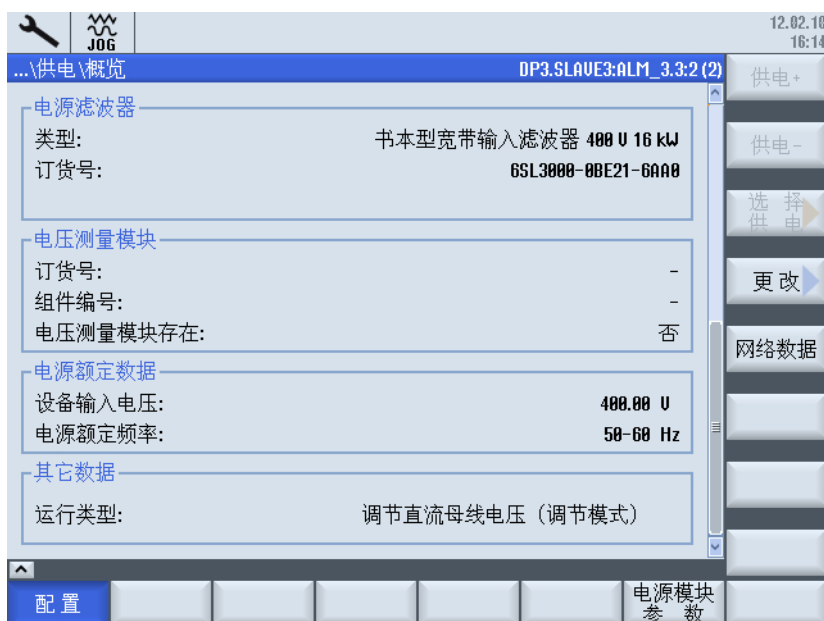


图 5-12 电源概览（第 2 页）

5.1.8 更换 SINAMICS S120 组件

更换电机模块

首次调试后需要更换组件时，例如将功率为 16 kW 的 ALM 更换为 50 kW 的 ALM 时，必须重新读入配置数据并更新驱动数据。

说明

固件升级

只有在驱动组件断电状态下才能进行正常的固件升级。同样，**只有在断电状态下才能插入驱动组件。**

如果在首次调试后需要将驱动组件更换为一个不同货号的组件，例如替换为功率更大的模块，请遵循以下操作过程：

操作步骤：

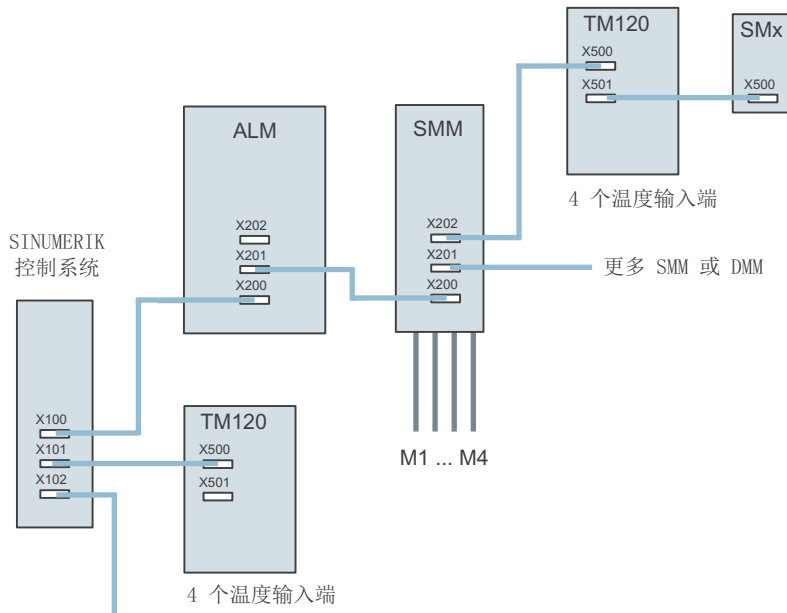
1. 选择“驱动设备”→“拓扑”→“更改”。
2. 选择模块并选择“取消驱动对象”。
3. 更换电柜中的模块。
4. 选择模块并选择“激活驱动对象”。

5. 这样可读入新模块的配置数据。
6. 将新的配置数据更新到调试存档中的驱动数据中。

5.1.9 示例：包含 TM120 的并联

拓扑结构

应用：4 电机并联



M1 ... M4 电机 1 ... 电机 4

每台电机上连接 1 个 KTY 和一个由 3 PTC 组成的串联电阻。

SMx 编码器模块（电机编码器）

ALM 调节型电源模块

DMM 双轴电机模块

SMM 单轴电机模块

TM120 端子模块

显示的拓扑需要 4 个 KTY 传感器和 4 个 PTC 传感器：

- 每个一次侧部分有 1 个 KTY 传感器（Temp-F）和 3 个串联的 PTC 传感器（Temp-S）

需要两个 TM120：

- 一个 TM120 自动串接至电机模块和编码器模块 SMx 之间。
- 另一个 TM120 直接插入电源模块：此处需要由调试人员手动进行连接。

措施

根据拓扑的不同，TM120 上应采取以下措施：

1. 电机模块和编码器模块 SMx 之间的 TM120

通过此 TM120 分析 4xKTY => 必须通过 Servo-p4610/TM-p4100 选择编码器类型。相应温度通过 Servo-r4620/TM120-r4105 输出。

2. TM120 直接插入电源模块

通过此 TM120 分析 4xPTC => 必须通过 TM-p4100 选择编码器类型。相应温度通过 Servo-r4105 输出。

PTC 关联菜单中的阈值设置：

– TM120-p4102[x]=251 => 分析关闭

– TM120-p4102[x]=120 => 分析开启

通过故障传播将 TM120 响应分配至驱动 => 设置传播：Servo-p0609=BICO:<对象编号>TM120:4105.0

将报警分配至电机

所出现的故障信息如下分配至电机：

- 温度通道“TM120 与电机模块和编码器串联”：

报警 207015 <路径>驱动：电机温度传感器报警

报警 207016 <路径>驱动：电机温度传感器故障

报警 235920 <路径>TM：故障 温度传感器通道 0

在此情况下会输出基于电机和温度组件的信息，所涉及电机在信息中可见。

- 温度通道“TM120 直接插入电源模块”：

报警 235207 <路径>TM：超出通道 0 温度故障/报警阈值

PTC 的特殊性：

TM120-r4105 = -50 温度低于额定响应温度

TM120-r4105 = 250 温度高于额定响应温度

在此情况下信息基于温度组件输出。

通过至 TM120 的信息可确定所涉及的电机。

5.1.10 交换过程数据

驱动的逻辑输入/输出地址

关于输入/输出地址，要对 NC 给出明确的数据区，以此完成与驱动和控制单元的循环过程数据交换：

- 驱动的输入/输出地址
- 控制单元的输入/输出地址
- 报文类型

将 PLC 中定义的输入/输出地址输入到以下的机床数据中：

不带 NX 模块：		
MD13120[0]	\$MN_CONTROL_UNIT_LOGIC_ADR ESS	= 6500
MD13120[1]	\$MN_CONTROL_UNIT_LOGIC_ADR ESS	= 0
带 NX 模块：		
MD13120[0]	\$MN_CONTROL_UNIT_LOGIC_ADR ESS	= 6500
MD13120[1]	\$MN_CONTROL_UNIT_LOGIC_ADR ESS	= 6516

报文类型

在预设的系统数据模块 (SDB) 中将驱动报文预设为类型 136。调试时可设置组件 NC (MD13060 \$MN_DRIVE_TELEGRAM_TYPE) 和 SINAMICS S120 (p0922 PROFIdrive PZD 报文选择) 的该报文类型。通过 MD11240 \$MN_PROFIBUS_SDB_NUMBER = n 选择系统数据块。

文档

有关过程数据的详细信息请参考以下文献：

- SINUMERIK 828D 参数手册，“机床数据”一章
- SINAMICS S120 功能手册，“通讯”一章
- SINAMICS S120/S150 参数手册，“功能图”一章

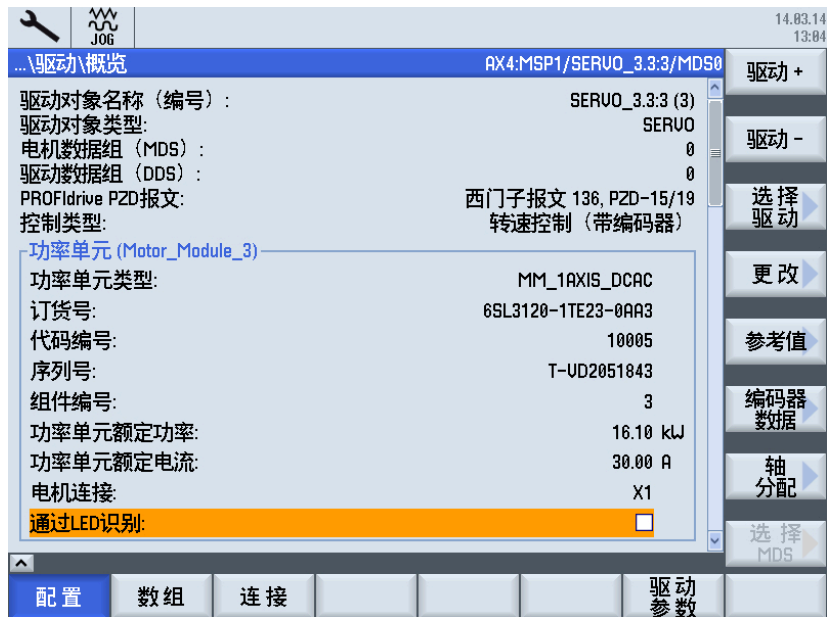
5.2 轴指定

5.2.1 示例：分配轴的步骤

轴分配

操作步骤：

1. 在保存了配置数据后，切换至以下概览：



2. 选择软键“轴分配”，为每个逻辑驱动指定一个真实的轴：



3. 选中最上方的选择栏“设定值：轴 → 驱动”。
4. 从选择列表中选择“AX1:MSP1 ”用于主轴。
5. 选中下一个选择栏“实际值：编码器 → 轴”并且按照表格来分配测量系统。
6. 按下垂直软键“驱动 +”和“驱动 -”选择下一个模块，进行其他轴分配：

轴	驱动	I/O 地址
MSP1	SERVO_3.3:3	4100
MX1	SERVO_3.3:4	4140
MY1	SERVO_3.3:5	4180
MZ1	SERVO_3.3:6	4220

7. 按下“确认”确认分配并将数据写入 NC。要使数据生效，必须对 NC 执行复位操作。

接收设置

在重新启动后，控制系统上会显示以下机床配置：



图 5-13 机床配置：所有轴经过调试

步骤 5 “调试驱动系统” 结束。

数据备份

按下垂直软键“数据存储”，调试后的配置数据会保存在非易失存储器中：

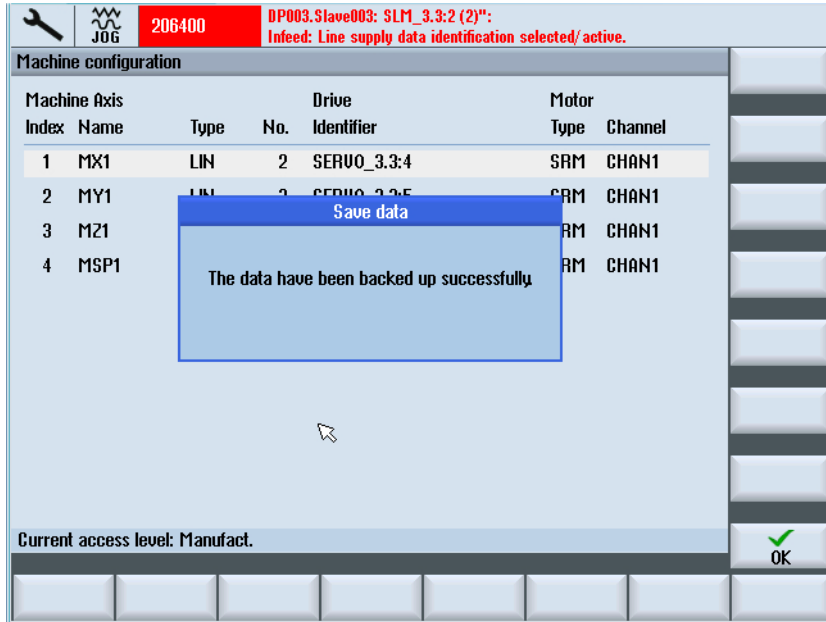


图 5-14 提示保存数据的信息

5.2.2 示例：设置进给轴/主轴的机床数据

进给轴机床数据

在结束上面示例中的调试后，进给轴机床数据被设为下列值：

进给轴机床数据		X	Y	Z	SP	A
MD30200	\$MA_NUM_ENCS	1	1	1	1	1
MD30230	\$MA_ENC_INPUT_NR	1	1	1	2	1
MD30240	\$MA_ENC_TYPE	1	1	1	1	4
MD31020	\$MA_ENC_RESOL	2048	2048	2048	1024	512
MD34200	\$MA_ENC_REFP_MODE	1	1	1	1	0

主轴机床数据

请重新设定定义下机床数据的值，以便在结束上面的驱动调试后在 JOG 模式中移动主轴。

主轴机床数据		缺省设置	新值
MD32000	\$MA_MAX_AX_VELO	10000	3000
MD32010	\$MA_JOG_VELO_RAPID	10000	100
MD32020	\$MA_JOG_VELO	2000	50
MD35100	\$MA_SPIND_VELO_LIMIT	10000	3000
MD35110[0]	\$MA_GEAR_STEP_MAX_VELO[0]	500	3000
MD35110[1]	\$MA_GEAR_STEP_MAX_VELO[1]	500	3000
MD35130[0]	\$MA_GEAR_STEP_MAX_VELO_LIMIT[0]	500	3150
MD35130[1]	\$MA_GEAR_STEP_MAX_VELO_LIMIT[1]	500	3150
MD36200[0]	\$MA_AX_VELO_LIMIT[0]	11500	3300
MD36200[1]	\$MA_AX_VELO_LIMIT[1]	11500	3300

最大转速驱动参数

在 DDS 切换后，例如 DDS0 → DDS1，可只使用 p1082[0]中的转速来运行主轴。如果在 p1082[0]中输入了比 p1082[1...n]小的值，则主轴转速会限制在 p1082[0]的值之内。

说明

应在参数 p1082[0]中输入最大转速：

- 如果将参数值设置得比其他的参数都大，则可使用最大转速运行主轴。
- 如果此后将参数 p1082[0]更改为一个较小的值，并接着进行 NCK 复位或重新上电，则又只能以 p1082[0]中该较小的值来运行主轴，其他所有的值都会被忽略。

诊断显示

需要浏览进给轴/主轴状态的其他信息时，请在操作区域“诊断”中按下菜单扩展键：

- 软键“轴诊断”打开“服务概览”。
- 软键“轴服务”打开窗口“服务 - 进给轴 / 主轴”。

说明

DSC 模式

前提条件：报文 ≥ 116 ；驱动参数 SINAMICS：

- p1192[0] 编码器选择
- p1193[0] 编码器调整系数

下列组合的参数设定是不允许的：\$MA_STIFFNESS_CONTROL_ENABLE=1 和
\$MA_ENC_FEEDBACK_POL=-1

编码器取反必须由 SINAMICS 驱动的参数 p0410 位 1 设置：

- DSC 模式是电机测量系统的缺省模式。
 - 对于外部测量系统，必须手动激活 DSC 模式。
-

5.3 调试辅助轴

5.3.1 配置示例 - 概览

通过 PROFINET 连接辅助轴

使用辅助轴可以执行辅助运动。不需要插补其他轴，所以几乎不影响系统的效率。辅助轴典型的应用就是例如换刀和换工件。

在 SINUMERIK 828D 上可以通过 PROFINET 连接最多 2 根辅助轴。对于连接辅助轴需要使用 CU310-2 PN 或者 CU320-2 PN，参见下面的配置示例。

PROFINET 的连接电缆必须按下列规定插接：

- PN1 连接到 SINUMERIK 828D
- X150 P1 连接到 CU310-2 PN 或者 CU320-2 PN



软件选件

要基于 PROFINET 通过 PLC I/O 接口来运行辅助轴，需要选件“定位轴/辅助主轴”。

配置示例

下列配置展示的是带 2 根辅助轴的示例，是基于 PROFINET 通过 PLC I/O 接口连接到 SINUMERIK 828D 上的：

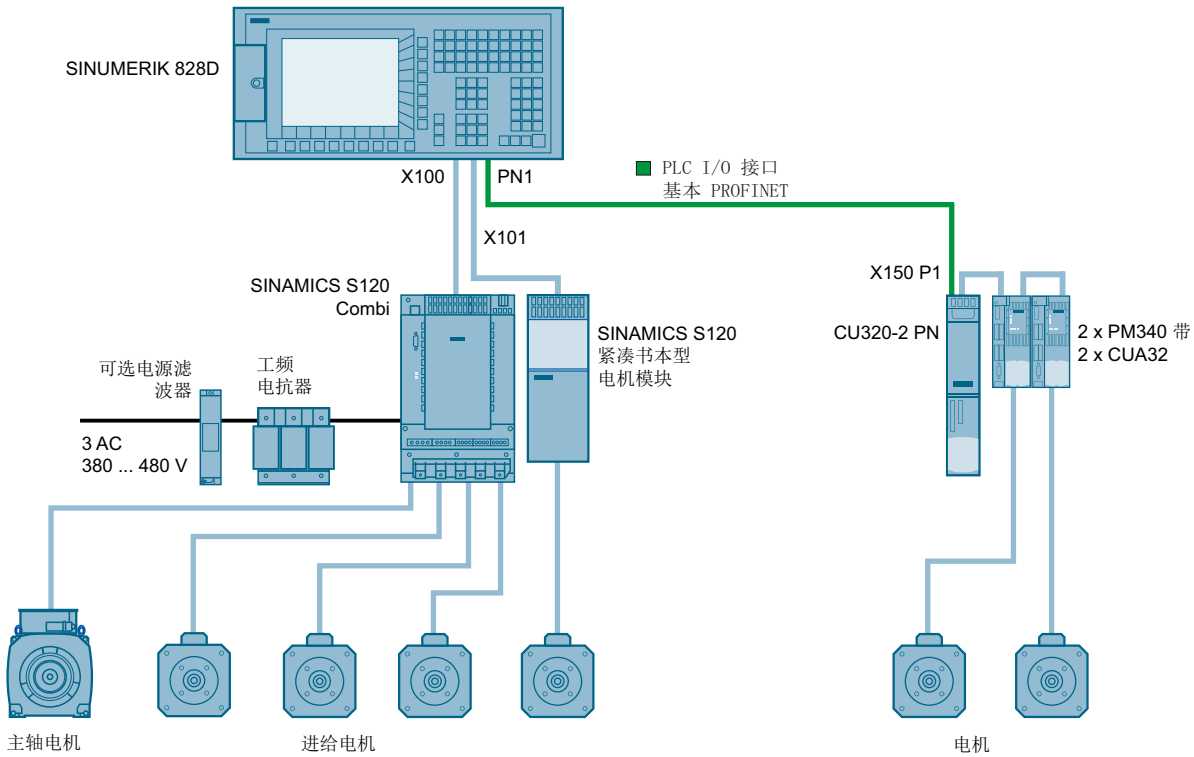


图 5-15 配置示例 1：2 根辅助轴带 CU320-2 PN，2 x PM340 和 2 x CUA32

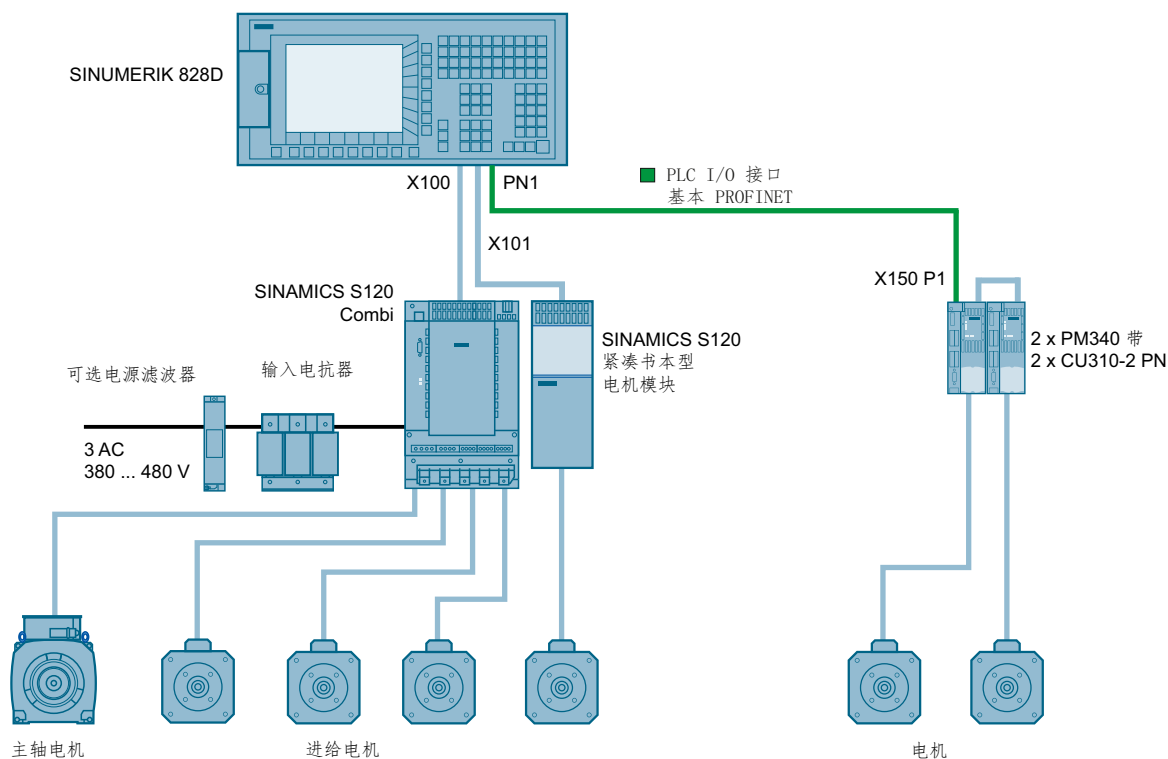


图 5-16 配置示例 2：2 根辅助轴带 2 x CU310-2 PN 和 2 x PM340

5.3.2 示例：配置辅助轴的步骤

CU310-2 PN/CU320-2 PN 的设备名称

CU310-2 PN/CU320-2 PN 的设备名称必须和系统中定义的名称相同。如果名称不相同，则不能通过 PROFINET 连接到控制系统的驱动，且不能运行驱动。

名称可以如下设定：

- 如果 CU310-2 PN/CU320-2 PN 还未命名，则系统会自动将其设备名称定义为“MQ2”。
- 如果 CU310-2 PN/CU320-2 PN 已命名，则名称必须和配置的设备名称相同，以保证能够正常运行。

激活辅助轴

可以激活：

- 通道中的辅助轴“MQ2”和“MQ3”通过 MD20070 \$MC_AXCONF_MACHAX_USED[] 进行设置。
- “MQ2”和“MQ3”的辅助轴定义通过 MD30460 \$MA_BASE_FUNCTION_MASK 的位 8 进行设置。
- 合适的 SDB 设置通过 MD11240 \$MN_PROFIBUS_SDB_NUMBER 进行，参见下表。

更改的设置要执行“上电复位”后才能生效。

Control Unit: 控制单元	SDB 编号	模块编号	轴名称	IP 地址
1x CU320-2 PN	(2,-1,2,2)	1. 轴 = 30 2. 轴 = 31	1. 轴 = MQ2 2. 轴 = MQ3	192.168.214.19
1x CU310-2 PN	(4,-1,4,4)	1. 轴 = 30	1. 轴 = MQ2	192.168.214.19
2x CU310-2 PN	(5,-1,5,5)	1. 轴 = 30 2. 轴 = 31	1. 轴 = MQ2 2. 轴 = MQ3	CU1: 192.168.214.19 CU2:192.168.214.18

调试驱动

此处描述的调试和轴分配参考了第一个配置示例“2 根辅助轴带 CU320-2 PN, 2 x PM340 和 2 x CUA32”。第二个配置示例的步骤也可以进行相应操作。

如果通过 PROFINET 可以连接到 SINUMERIK 828D 的控制单元 CU320-2 PN, 则可以调试驱动。

在进行自动设备配置时识别出连接的辅助轴并对其进行配置，参见自动首次调试 (页 111)。

这样检查/更改驱动配置：

1. 要打开窗口“驱动”，需要选择“调试 → 驱动系统 → 驱动”。
2. 选择“更改”，配置驱动。
3. 保存更改后的配置数据。

执行驱动分配/轴分配

驱动调试结束后执行驱动分配/轴分配。

操作步骤:

1. 通过“调试 → 驱动系统 → 驱动 → 轴分配”启动驱动分配/轴分配。
2. 给 I/O 地址为“5260”的驱动“DR30”分配轴“MQ2”。
I/O 地址“5260”在机床数据 MD13050 \$MN_DRIVE_LOGIC_ADDRESS [29] 中。

轴	驱动	I/O 地址
MQ2	DR30	5260
MQ3	DR31	5300

3. 给 I/O 地址为“5300”的驱动“DR31”分配轴“MQ3”。
I/O 地址“5300”在机床数据 MD13050 \$MN_DRIVE_LOGIC_ADDRESS [30] 中。

驱动和轴已完成调试。

5.4 DRIVE-CLiQ 拓扑规则

5.4.1 S120 Combi 的拓扑规则

DRIVE-CLiQ 拓扑规则

S120 Combi 有固定的 DRIVE-CLiQ 拓扑规则。必须遵守这些规则，否则会显示报警。

DRIVE-CLiQ 接口的布局

表格 5-1 S120 Combi 上 DRIVE-CLiQ 接口的布局

DRIVE-CLiQ 接口	用于连接
X200	PPU 的 X100
X201	主轴电机编码器
X202	进给轴 1 电机编码器
X203	进给轴 2 电机编码器
X204	进给轴 3 的电机编码器 → 仅限 4 轴型的 S120 Combi 在 3 轴型 S120 Combi 上，接口空置
X205	可选：用于主轴的第 2 个正弦/余弦直接编码器（通过 SMx20 连接） ¹⁾ 通过 X220 连接 TTL 直接主轴编码器时，接口空置

¹⁾ 在这种情况下，TTL 编码器接口 X220 空置

表格 5-2 SINUMERIK 828D (PPU) 上 DRIVE-CLiQ 接口的布局

DRIVE-CLiQ 接口	用于连接
X100	S120 Combi 上的 X200
X101	单轴电机模块或双轴电机模块上的 X200
X102	端子模块 TM54F 上的 X500 集线器模块 DMx20 上的 X500 ¹⁾

¹⁾ 在使用 TM54F 时，DMx20 通过 TM54F 上的 DRIVE-CLiQ 接口 X501 串联在一起

表格 5-3 扩展轴上 DRIVE-CLiQ 接口的布局

DRIVE-CLiQ 接口	用于连接
第一个单轴电机模块	
X200	PPU 的 X101
X201 ¹⁾	第二个单轴电机模块上的 X200
X202	用于进给轴 1、扩展轴的电机编码器（通过编码器模块连接）
第二个单轴电机模块	
X200	第一个单轴电机模块上的 X201
X201	空置
X202	用于进给轴 2、扩展轴的电机编码器（通过编码器模块连接）
双轴电机模块	
X200	PPU 的 X101
X201	空置
X202	用于进给轴 1、扩展轴的电机编码器
X203	用于进给轴 2、扩展轴的电机编码器

¹⁾ 只使用一个单轴电机模块时，接口空置

表格 5-4 DMx20 上 DRIVE-CLiQ 接口的布局，用于为进给轴指定直接测量系统

DRIVE-CLiQ 接口	进给轴
X500	TM54F 上的 X501 PPU 上的 X102（当不使用 TM54F 时）
X501	用于连接 S120 Combi 进给轴 1 的直接编码器
X502	用于连接 S120 Combi 进给轴 2 的直接编码器
X503	S120 Combi 上进给轴 3 的直接编码器 → 仅限 4 轴型的 S120 Combi 在 3 轴型 S120 Combi 上，接口空置
X504	电机模块上的进给轴 1、扩展轴
X505	电机模块上的进给轴 2、扩展轴

5.4 DRIVE-CLiQ 拓扑规则

表格 5-5 TM54F 上 DRIVE-CLiQ 接口的布局

DRIVE-CLiQ 接口	
X500	控制系统(PPU)上的 X102
X501	DMx20 上的 X500 不使用 DMx20 时，接口空置。

5.4.2 S120 书本型的拓扑规则

引言

DRIVE-CLiQ 组件有以下拓扑规则/建议。其中，分为**强制性 DRIVE-CLiQ 拓扑规则**和**拓扑建议**，遵守拓扑建议，在扩展拓扑时便无需再次费力修改。

DRIVE-CLiQ 组件的最大数量、拓扑类型主要取决于以下几点：

- 强制性 DRIVE-CLiQ 拓扑规则
- 所激活驱动的数量和类型以及相应控制单元的功能
- 相应控制单元的计算效率
- 所设置的处理周期和通讯周期

除了强制性拓扑规则和补充的拓扑建议外，PPU 手册中还介绍了几个 DRIVE-CLiQ 拓扑示例。

这些示例中，组件可以被删除、更换或添加。如果组件被替换为其它类型的组件或者添加了额外的组件，则应使用工具 SIZER 来检查该拓扑。

说明

经过 SIZER 确认的每个拓扑都是正确的、可以投入运行的（SINAMICS S120 功能手册/FH1/）。

DRIVE-CLiQ 拓扑规则

下列的拓扑规则针对的是标准周期时间（伺服为 125 μ s）。低于该周期时间时，控制单元的计算效率会产生限制。

以下规则在一般情况下都适用，除非受到固件版本的限制。

- 每排最多允许有 8 个 DRIVE-CLiQ 节点。每排都是从控制单元的角度观察的。
在配置基于驱动的 Safety Integrated 功能时，数量减少到每排最多有 4 个 DRIVE-CLiQ 节点。
- 控制单元的一条 DRIVE-CLiQ 线路上最多允许有 14 个 DRIVE-CLiQ 节点。
- 不允许环形拓扑。
- 组件不允许重复连接。

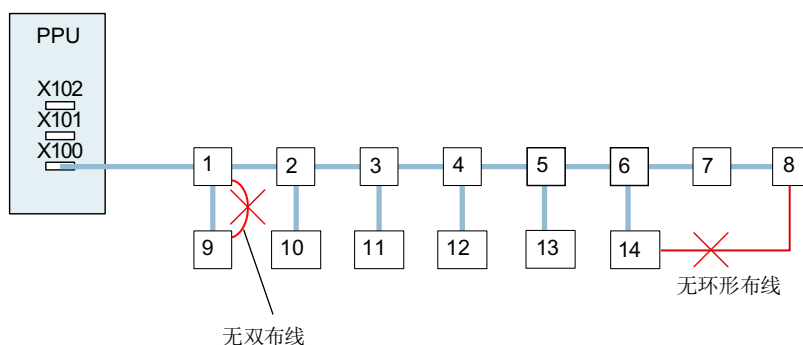


图 5-17 示例：端子 X100 上的 DRIVE-CLiQ 支路（无 Safety Integrated）

- 对于书本型结构类型：
 - 在伺服控制和矢量 V/f 控制运行方式中，控制单元上只可连接一个电源模块。在矢量控制运行方式中则可再并联三个电源模块（即总共 4 个电源模块）。
 - 在伺服控制运行方式中，一个电源模块和若干电机模块可连接至相同的 DRIVE-CLiQ 支路。
 - 在矢量控制运行方式中，一个电源模块和若干电机模块必须连接至不同的 DRIVE-CLiQ 支路。
 - 使用书本型结构类型时，不允许并联电源模块或电机模块。
- 针对装机装柜型组件：电源模块（基本型、调节型、非调节型）和电机模块必须连接到同一条 DRIVE-CLiQ 支路上。
- 允许修改预设的采样时间。
- NX10.3 / NX15.3 必须连接到 PPU 的 X102 上。

- 在“PPU + NX10.3 / NX15.3”的组合上可以最多连接 16 / 20 个测量系统：
 - 示例 1：6 轴型 PPU 可连接 6 个电机测量系统和 6 个直接测量系统；NX10.3 上连接 2 个电机测量系统和 2 个直接测量系统。
 - 示例 2：5 轴型 PPU 可连接 5 个电机测量系统和 5 个直接测量系统；NX10.3 上连接 3 个电机测量系统和 3 个直接测量系统。
 - 示例 3：6 轴型 PPU 可连接 6 个电机测量系统和 6 个直接测量系统；NX15.3 上连接 4 个电机测量系统和 4 个直接测量系统。
- 每个 PPU 和 NX10.3 / NX15.3 只允许连接一个 TM54F。
- 书本型调节型电源模块和书本型电机模块可以连接在一条 DRIVE-CLiQ 线路上。
- 装机装柜型电源模块和电机模块串联在一起。
- 下列模块应连接在相应的调节型电源模块/电机模块的空置 DRIVE-CLiQ 端口上，从而在调试时自动分配（设备识别）。
 - 电压监控模块 (VSM)
 - 端子模块 TM120

说明

若连接 TM120 时不按照这条规则，就必须通过 BICO 技术指定温度通道以及调试驱动。

- 在一条 DRIVE-CLiQ 线路上的所有组件的采样时间（p0115[0] 和 p4099）必须能够互相整除。如果必须将一个 DO 上的电流调节采样时间修改到另一最小单位，而 DRIVE-CLiQ 线路上的其它 DO 与最小单位并不匹配，则可以按以下方式设置：
 - 将 DO 换插到另一个单独的 DRIVE-CLiQ 线路上。
 - 将其他 DO 上的电流调节采样时间、输入输出端采样时间修改为和新的最小单位匹配的值。

说明

一个双轴电机模块，一个 DMC20 和一个 TM54F 相当于两个 DRIVE-CLiQ 节点。只配置了一个驱动的双轴电机模块也是如此。

为了使用“自动配置”功能将编码器指定给驱动，请遵守以下拓扑建议：

拓扑建议：

- 控制单元的 DRIVE-CLiQ 电缆须按以下方法连接：
 - 连接在首个书本型功率部件的 X200 上
 - 连接在首个装机装柜型功率部件的 X400 上
- 功率部件间的 DRIVE-CLiQ 电缆应分别从接口 X201 连接到下一个组件的 X200 上，或者相应地从 X401 连接到 X400 上。

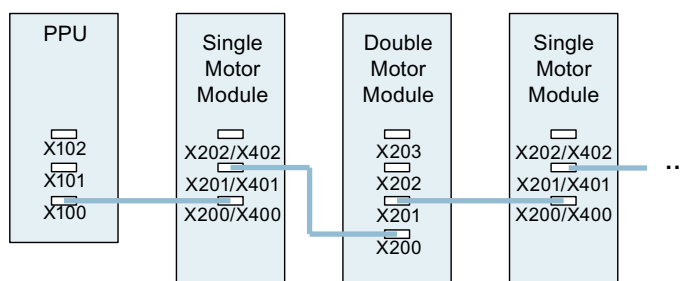


图 5-18 示例：DRIVE-CLiQ 线路

- 电机编码器必须与对应的功率部件相连接。

组件	通过 DRIVE-CLiQ 连接电机编码器
书本型单轴电机模块	X202
书本型双轴电机模块	<ul style="list-style-type: none"> 电机接线端子 X1: 编码器连接到 X202 电机接线端子 X2: 编码器连接到 X203
装机装柜型单轴电机模块	X402
装机装柜型功率模块	X402

说明

如果在电机模块上连接了一个额外的编码器，它将作为编码器 2 自动指定给该驱动。在双轴电机模块上，X201 上的编码器作为第 2 测量系统指定给进给轴 2 使用。

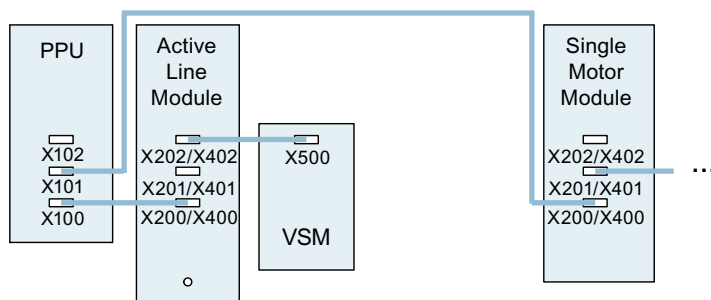


图 5-19 示例：书本型和装机装柜型组件带有 VSM 的拓扑

组件	VSM 连接
调节型电源模块（书本型）	X202
调节型电源模块（装机装柜型）	X402
功率模块	不支持 VSM。

5.4.3 SMC40 的拓扑结构规定

使用 SMC40

机柜安装式编码器模块 SMC40 用于将 EnDat 2.2 绝对值编码器的信号转换成 DRIVE-CLiQ 信号。SMC40 上可以连接两个 EnDat 2.2 编码器系统，这两个系统发出的信号分别转换成一路 DRIVE-CLiQ 信号。

连接条件

为进行**拓扑结构的基本调试**时系统能自动识别 SMC40，务必注意以下规定：

- 通过 DRIVE-CLiQ 电缆将 SINAMICS 设备连接到 SMC40 上的至少一个 DRIVE-CLiQ 接口：X500/1 或 X500/2。
- 将 EnDat 编码器连接到对应的编码器接口：X520/1（和 X500/1 对应）或 X520/2（和 X500/2 对应）。
- 只在星形拓扑结构中运行 SMC40。DRIVE-CLiQ 接口 X500/1 和 X500/2 不允许用于串联。

注意

“拓扑结构”对话框中显示

如果 DRIVE-CLiQ 插口 X500/x 和配套的编码器接口 X520/x 被占用，则 SMC40 将接收在实际拓扑结构中。

如果没有连接编码器，SMC40 也不会传送到拓扑结构中。

5.5 端子布局

5.5.1 X122 和 X132 的引脚布局

X122 PPU 数字量输入/输出端

引脚	信号名称	状态	含义	
1	DI0	DI0	r722.0	数字量输入端 0
2	DI1	DI1	r722.1	数字量输入端 1
3	DI2	DI2	r722.2	数字量输入端 2
4	DI3	DI3	r722.3	数字量输入端 3
5	DI16	DI16	r722.16	数字量输入端 16
6	DI17	DI17	r722.17	数字量输入端 17
7	M2	MEXT2	---	引脚 1....6 的接地
8	P1	P24EXT1	---	+24 V 电源
9	IO8	DI/DO8	r722.8 / r747.8	数字量输入/输出端 8
10	IO9	DI/DO9	r722.9 / r747.9	数字量输入/输出端 9
11	M1	MEXT1	---	引脚 9、10、12、13 的接地
12	IO10	DI/DO10	r722.10 / r747.10	数字量输入/输出端 10
13	IO11	DI/DO11	r722.11 / r747.11	数字量输入/输出端 11
14	M1	MEXT1	---	引脚 9、10、12、13 的接地

X132: PPU 数字量输入/输出端

引脚	信号名称	状态	含义	
1	DI4	DI4	r722.4	数字量输入端 4
2	DI5	DI5	r722.5	数字量输入端 5
3	DI6	DI6	r722.6	数字量输入端 6
4	DI7	DI7	r722.7	数字量输入端 7
5	DI20	DI20	r722.20	数字量输入端 20
6	DI21	DI21	r722.21	数字量输入端 21
7	M2	MEXT2	---	引脚 1....6 的接地

5.5 端子布局

引脚	信号名称	状态	含义	
8	P1	P24EXT1	---	+24 V 电源
9	IO12	DI/DO12	r722.12 / r747.12	数字量输入/输出端 12
10	IO13	DI/DO13	r722.13 / r747.12	数字量输入/输出端 13
11	M1	MEXT1	---	引脚 9、10、12、13 的接地
12	IO14	DI/DO14	r722.14 / r747.14	数字量输入/输出端 14
13	IO15	DI/DO15	r722.15 / r747.15	数字量输入/输出端 15
14	M1	MEXT1	---	引脚 9、10、12、13 的接地

5.5.2 X242 和 X252 的端子布局

X242 的引脚布局

引脚	信号名称	NC 变量	含义	
1	未连接			
2	未连接			
3	IN1	DIN1	\$A_IN[1]	NC 数字量输入端 1
4	IN2	DIN2	\$A_IN[2]	NC 数字量输入端 2
5	IN3	DIN3	\$A_IN[3]	NC 数字量输入端 3
6	IN4	DIN4	\$A_IN[4]	NC 数字量输入端 4
7	M4	MEXT4		引脚 3...6 的接地
8	P3	P24EXT3		+24 V 电源
9	O1	DOUT1	\$A_OUT[1]	NC 数字量输出端 1
10	O2	DOUT2	\$A_OUT[2]	NC 数字量输出端 2
11	M3	MEXT3		引脚 9、10、12、13 的接地
12	O3	DOUT3	\$A_OUT[3]	NC 数字量输出端 3
13	O4	DOUT4	\$A_OUT[4]	NC 数字量输出端 4
14	M3	MEXT3		引脚 9、10、12、13 的接地

X252 的引脚布局

引脚		信号名称	NC 变量	含义
1	AO	AOUT		模拟量输出端（用于模拟主轴的电压）
2	AM	AGND		模拟接地
3	IN9	DIN9	\$A_IN[9]	NC 数字量输入端 9
4	IN10	DIN10	\$A_IN[10]	NC 数字量输入端 10
5	IN11	DIN11	\$A_IN[11]	NC 数字量输入端 11
6	IN12	DIN12	\$A_IN[12]	NC 数字量输入端 12
7	M4	MEXT4		引脚 3...6 的接地
8	P3	P24EXT3		+24 V 电源
9	O9	DOUT9	\$A_OUT[9]	NC 数字量输出端 9
10	O10	DOUT10	\$A_OUT[10]	NC 数字量输出端 10
11	M3	MEXT3		引脚 9、10、12、13 的接地
12	O11	DOUT11	--	<ul style="list-style-type: none"> 不带模拟量主轴： NC 数字量输出端 11 带模拟量主轴： 模拟量主轴的伺服使能（根据 MD30134 \$MA_IS_UNIPOLAR_OUTPUT）
13	O12	DOUT12	--	<ul style="list-style-type: none"> 不带模拟量主轴： NC 数字量输出端 12 带模拟量主轴： 模拟量主轴的运动方向（根据 MD30134 \$MA_IS_UNIPOLAR_OUTPUT）
14	M3	MEXT3		引脚 9、10、12、13 的接地

参见

用于模拟量主轴的机床数据 (页 90)

5.5.3 示例：CU 和电源接触器的连接

示例

示意图中展示的连接方法针对的是上一章节中介绍的端子布局。

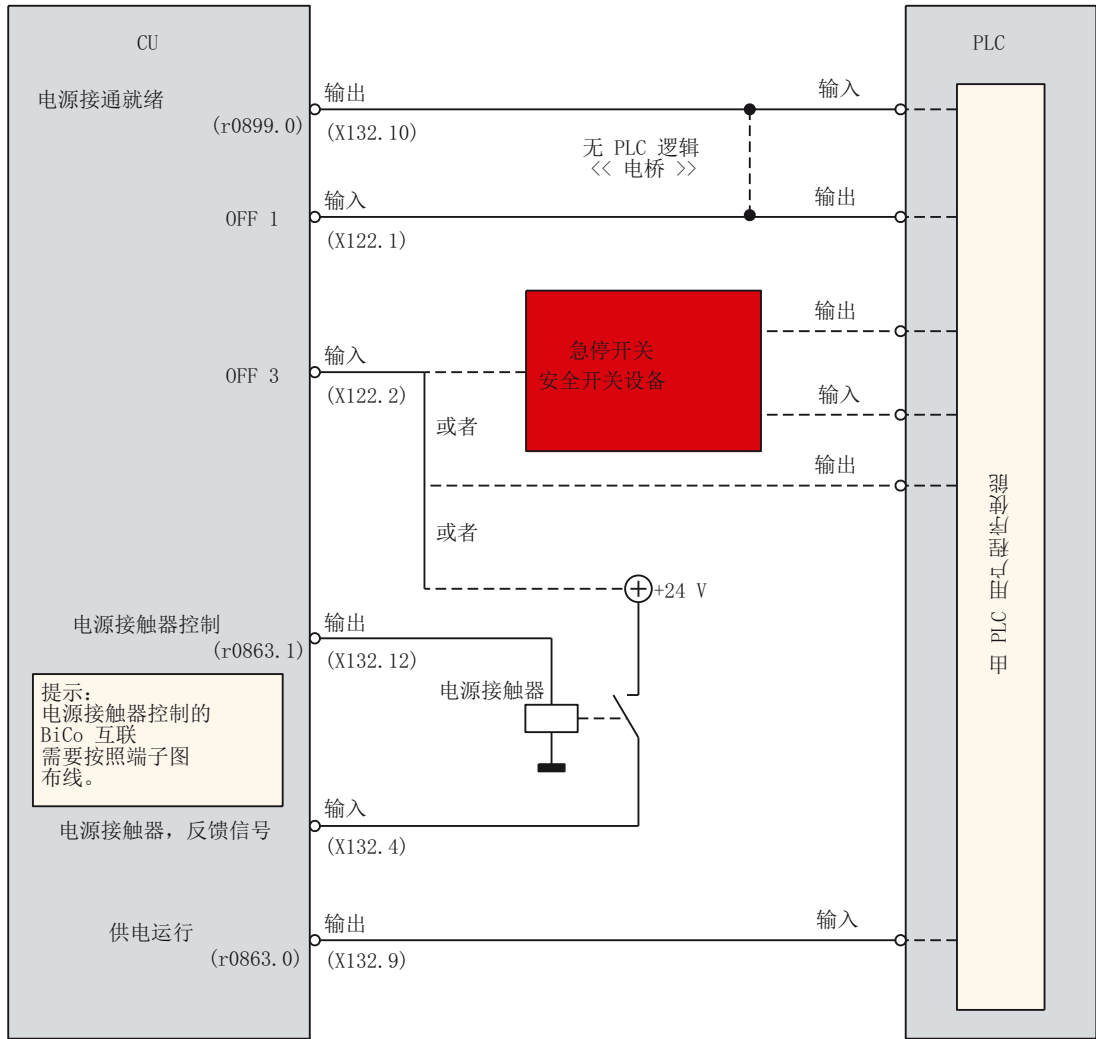
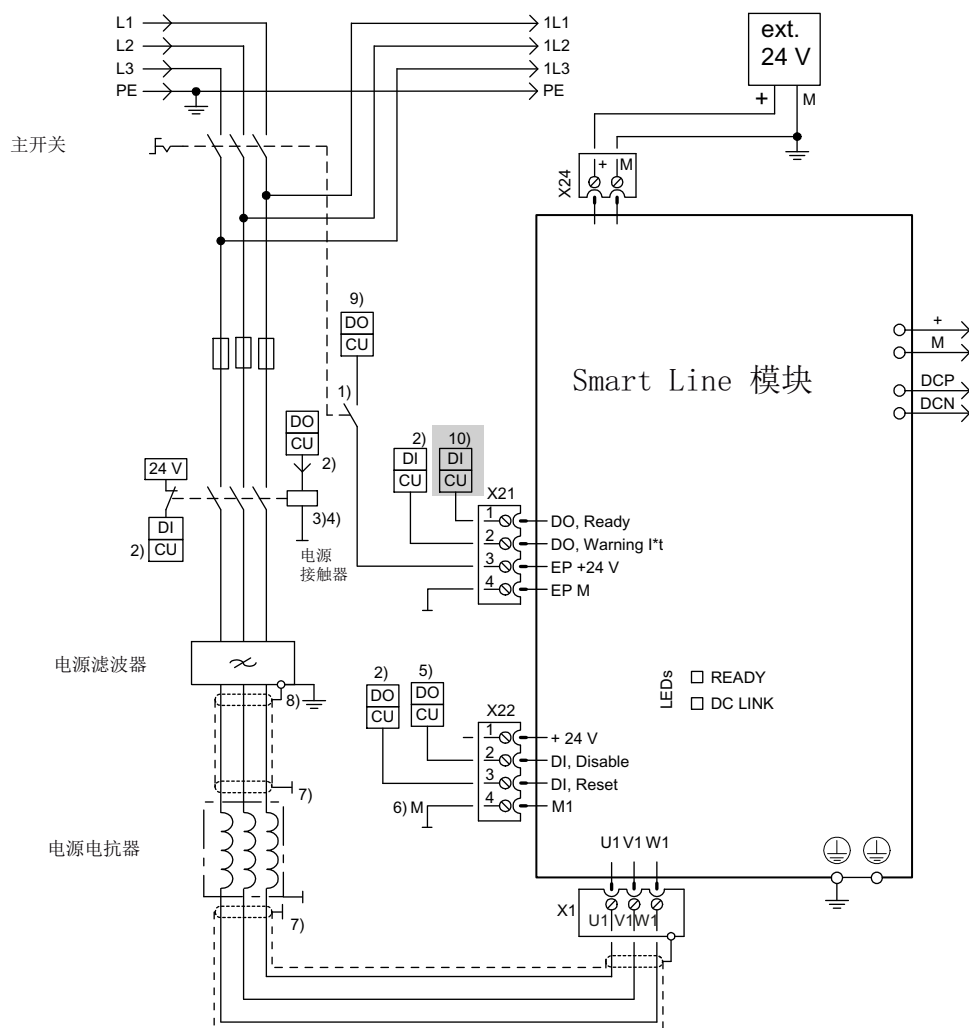


图 5-20 控制单元和电源接触器的连接

连接调节型电源模块

用于连接数字量输入输出 X122 和 X132 的接口位于控制系统的背面。



- ① 1) 提前打开的触点 $t > 10\text{ ms}$, 运行时必须设置 DC 24 V 和接地。
- ② DI/DO 由控制单元控制。
- ③ 电源接触器后面不能另外连接其他用电设备。
- ④ 此时请注意 DO 的载流能力, 必要时必须使用输出耦合元件。
- ⑤ DO high: 回馈功能被取消, 需要长时间取消时, 应在 X22 引脚 1 和引脚 2 之间插入跳线。
- ⑥ X22 引脚 4 必须接地(外部 24 V)。
- ⑦ 按照 EMC 安装指令通过安装后壁或屏蔽母线接地
- ⑧ 5 kW 和 10 kW 的电源滤波器通过屏蔽端子接地
- ⑨ 控制系统上的信号输出, 防止 DC 24 V 电源反作用于 EP 端子。
- ⑩ 通过 BICO 互联到参数 p0864 → X122.1

图 5-21 示例: 连接 SLM

5.5 端子布局

使能 ON/OFF1: 连接非调节型电源模块引脚 X21.1 → X122.1 SINUMERIK 828D

连接其他的输入输出信号和 PLC 外设:

- DI → PLC 输入
- DO → PLC 输出

另见

其它信息请参见:

- SINUMERIK 828D 设备手册 PPU
- SINAMICS S120 书本型功率部件手册

5.5.4 输入/输出(对话框)

端子布局支持

下图显示了 SINAMICS 驱动组中所包含驱动设备的端子布局。

操作区域 “调试” → “驱动系统” → “驱动设备” → “输入/输出” :



图 5-22 输入/输出的端子布局

轴/主轴测试运行

驱动:

参数/端子	含义
p0840	ON/OFF1
p0844	1. OFF2
p0845	2. OFF2
p0848	1. OFF3
p0849	2. OFF3
p0852	运行使能
X21.3 (+24 V) 和 X21.4 (接地)	EP 端子使能 (脉冲使能)
p0864	电源使能
p1140	斜坡函数发生器使能
p1141	斜坡函数发生器启动
p1142	设定值使能

电源:

参数/端子	含义
p0840	ON/OFF1
p0844	1. OFF2
p0845	2. OFF2
p0852	运行使能
X21.3 (+24 V) 和 X21.4 (接地)	EP 端子使能 (脉冲使能)

另见

关于驱动的其他文献:

- SINAMICS S120 调试手册
- 书本型功率部件手册

5.5.5 互联（对话框）

互连

在每个驱动设备中都存在大量的可互联输入/输出参数以及闭环控制的内部参数。利用 BICO 互联技术(Binector Connector Technology)，您可以对驱动设备功能进行调整，以满足各种应用的要求。

可通过 BICO 参数任意连接的数字和模拟信号，其参数名预设为 BI、BO、CI 或 CO。这些参数在参数列表或功能图中也具有相应的标记：

- 二进制接口（digital）：
 - BI： 数字接口输入
 - BO： 二进制信号源输出端
- 模拟量接口（analog）：
 - CI： 模拟量互联输入
 - CO： 模拟输出端

控制系统显示

必须将输入参数（信号汇点）分配给所需的输出参数（信号源），以连接两个信号。在以下对话框中检测或修改 SINAMICS 驱动组中组件的互联：

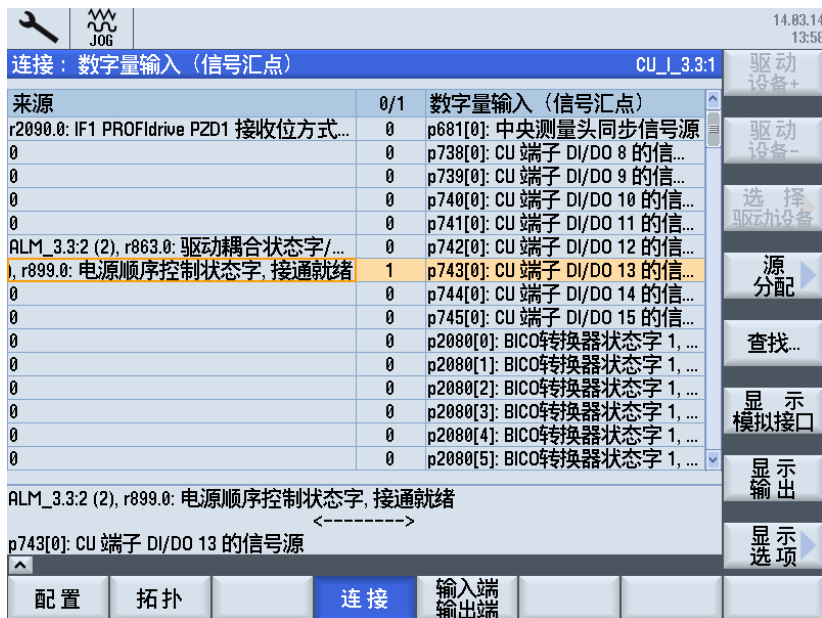


图 5-23 示例：互联

5.5.6 连接测量头

连接测头

测头不仅可以连接到 SINUMERIK 828D CU 上，也可以连接到 NX 上：

1. NX 上 X122 引脚 13 / X122 引脚 13 端子的测头
2. NX 上 X132 引脚 13 / X122 引脚 12 端子的测头

说明

在 SINUMERIK 828D 上开始测量的前提条件是：设置了分布式（本地）测量功能。

SINUMERIK 828D 不支持中央测量。

机床数据

检查以下机床数据，必要时修改数据：

- **通用机床数据：**

MD13200[0] \$MN_MEAS_PROBE_LOW_ACTIVE = 0 或 1

MD13200[1] \$MN_MEAS_PROBE_LOW_ACTIVE = 0 或 1

值 0 = 偏转状态 24 V（缺省设置）

值 1 = 偏转状态 0 V

MD13210 \$MN_MEAS_TYPE = 1 分布测量

- **轴专用机床数据：**

MD30244[0] \$MA_ENC_MEAS_TYPE = 1，用于所有轴

MD30244[1] \$MA_ENC_MEAS_TYPE = 1，用于所有轴

说明

机床数据 MD13210 和 MD30244 的缺省设置是 1，不能修改！（数据级：S）

- **控制单元参数:**

p0680[0] 中央测头输入端子 = 0

p0680[1] 中央测头输入端子 = 0

p0680[2] 中央测头输入端子 = 0

测头 1 连接到 SINUMERIK 828D 的端子 X122 引脚 13 上；测头 2 连接到 X132 引脚 13 上，连接了 NX10 模块时，连接到该模块的端子 X122 引脚 12 上。

前提是，已经将引脚 X132.13 从输出端切换为输入端。

设置 CU 输入端或输出端 = 将 p0728 位 15 设为 0(DI/DO X132.13)

- **驱动参数:**

p0488[0] 测头 1 输入端子: 编码器 1 = 3 → 引脚 X122.13

p0488[1] 测头 1 输入端子: 编码器 2 = 3 → 引脚 X122.13

p0488[2] 测头 1 输入端子: 编码器 3 = 0 → 不使用

p0489[0] 测头 2 输入端子: 编码器 1 = 6 → 引脚 X132.13

p0489[1] 测头 2 输入端子: 编码器 2 = 6 → 引脚 X132.13

p0489[2] 测头 2 输入端子: 编码器 3 = 0 → 不使用

在所有 NX 模块的轴上，必须将测头 2 的输入端子设为 X122.12。

p0489[0] 测头 2 输入端子: 编码器 1 = 6 → 引脚 X122.12

p0489[1] 测头 2 输入端子: 编码器 2 = 6 → 引脚 X122.12

p0489[2] 测头 2 输入端子: 编码器 3 = 0 → 不使用

说明

请设置所有驱动。

5.6 配置数据组

5.6.1 数据组一览

前提条件

说明

相应驱动必须已经过调试。

数据组

通过操作区域“调试”→“驱动系统”→“驱动”→“数据组”执行数据组配置。在系统的引导下分步执行以下步骤：

- “添加数据组”
- “删除数据组”，若已创建另一个数据组的情况下。
- “修改数据组”

可配置数据组的数量：

- 电机数据组 → MDS0...3（最多 4 个）
- 驱动数据组 → DDS0...31（每个 MDS 最多 8 个）
- 编码器数据组 → EDS0...2（最多 3 个）

说明

对数据组有影响的驱动参数

接口模式 SINAMICS 的缺省设置为 $p2038 = 0$ 。

在通过 $p0922$ 选择报文后，可通过参数 $p2038$ 针对设备具体条件来设置控制字和状态字中的各个位。

$p0922 = 100 \dots 199$ 时会自动设置 $p2038 = 1$ ，并禁止修改 $p2038$ 。因此，在使用报文时接口模式固定设为“SIMODRIVE 611U”，每个 MDS 有 8 个 DDS。

应用

数据组向导(DSA)是供机床厂商(OEM)使用的简单、高效的向导程序，它可以用于复制和修改数据，以配置驱动、电机和编码器的特性。DSA 中的数据可重复使用，因此当需要为协调各组件的运行而生成机床数据时，所需配置时间可降到最低。

5.6 配置数据组

驱动系统各组件的特性是通过电机数据组（Motor Data Set，简称 MDS）、驱动数据组（Drive Data Sets，简称 DDS）以及编码器数据组（Encoder Data Set，简称 EDS）来配置的。数据可重复使用，因此当需要协调各组件的机械动态响应时，配置步骤得以减少。

文档

其它信息请参见：

SINUMERIK 840D sl/828D 功能手册之基本功能分册：章节“各种 NC/PLC 接口信号与功能(A2)”

5.6.2 添加数据组

添加数据组的操作步骤

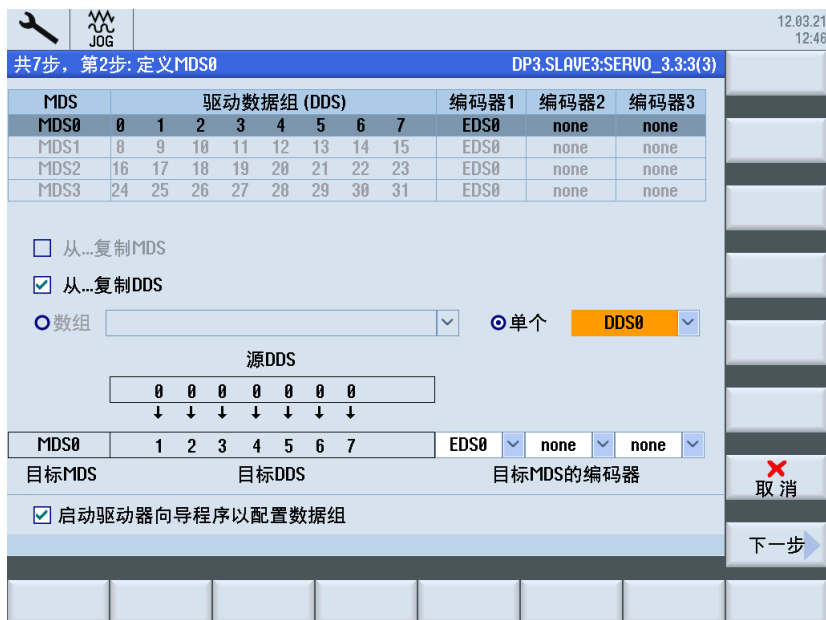
出厂设置为一个电机数据组 MDS0、一个驱动数据组 DDS0 和一个编码器数据组 EDS0：



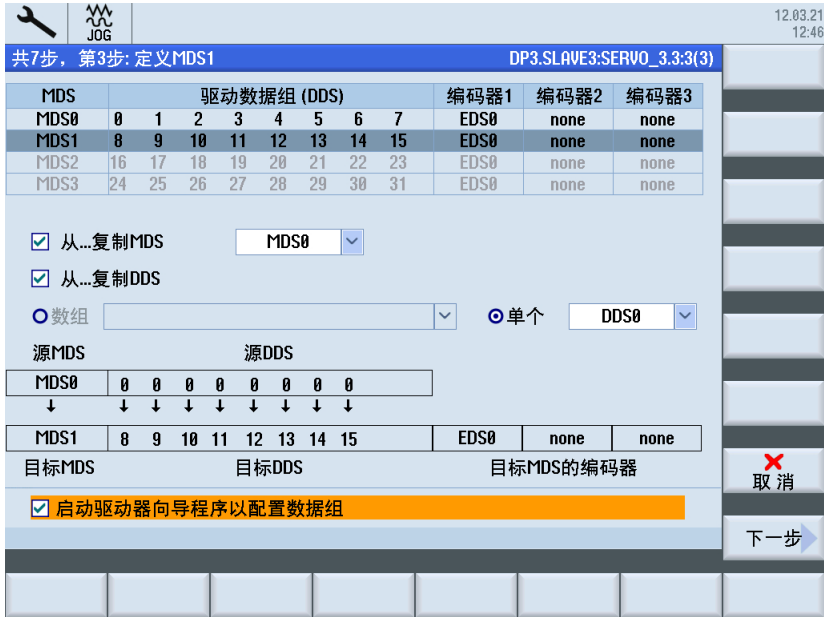
1. 按下“添加数据组”。此示例中创建 4 个 MDS（最大数量）。



2. 按下“下一步 >”。
 第一步在 MDS0 中创建驱动数据组 DDS1 至 DDS7。



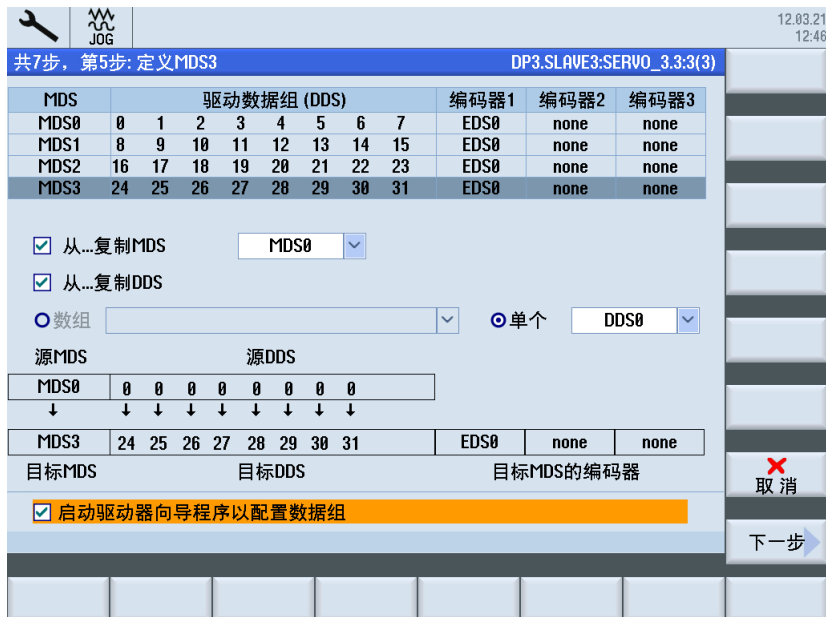
3. 按下“下一步 >”。创建电机数据组 MDS1。



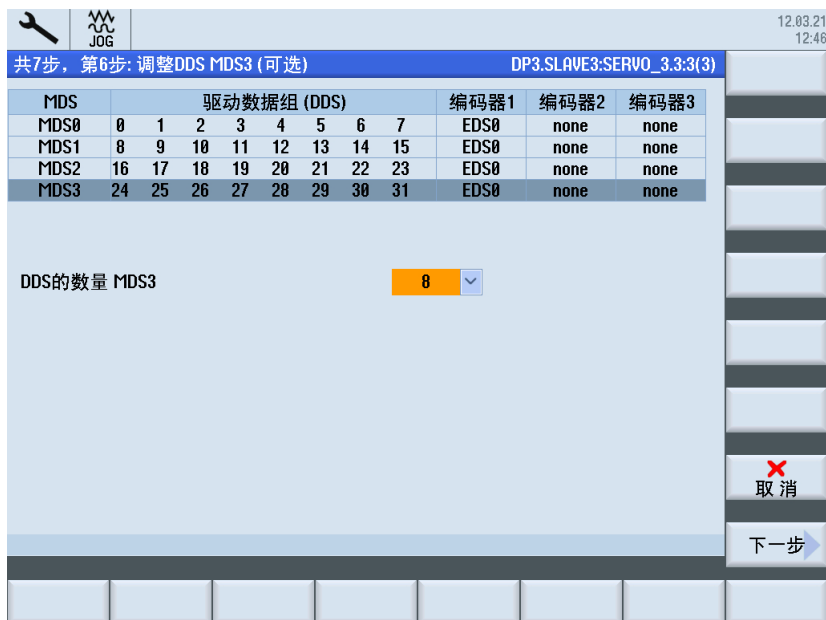
4. 按下“下一步 >”。创建电机数据组 MDS2。



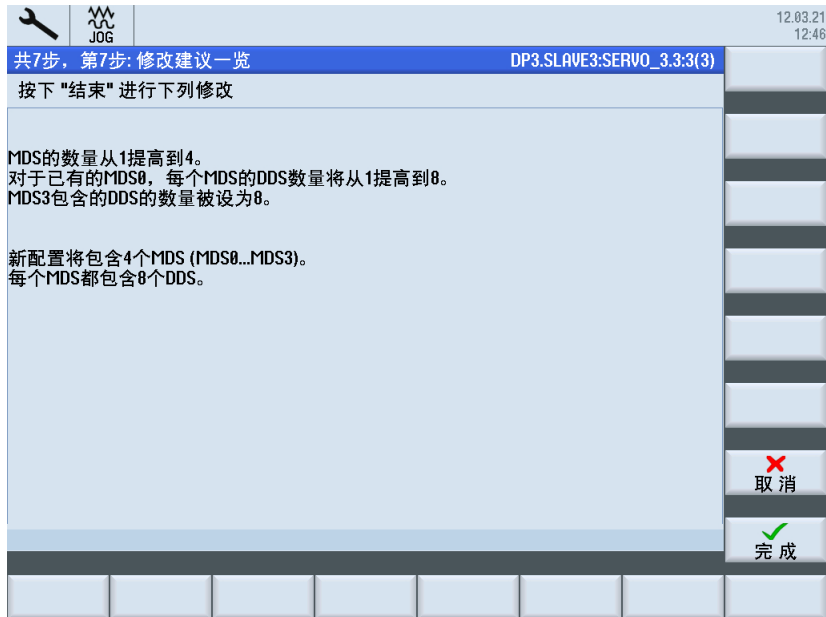
5. 按下“下一步 >”。创建电机数据组 MDS3。



6. 按下“下一步 >”。如下图所示，创建一个完整的 MDS 或者选择输入某个 MDS 的 DDS 的数量。



7. 按下“下一步 >”，在摘要中检查需要执行的修改。

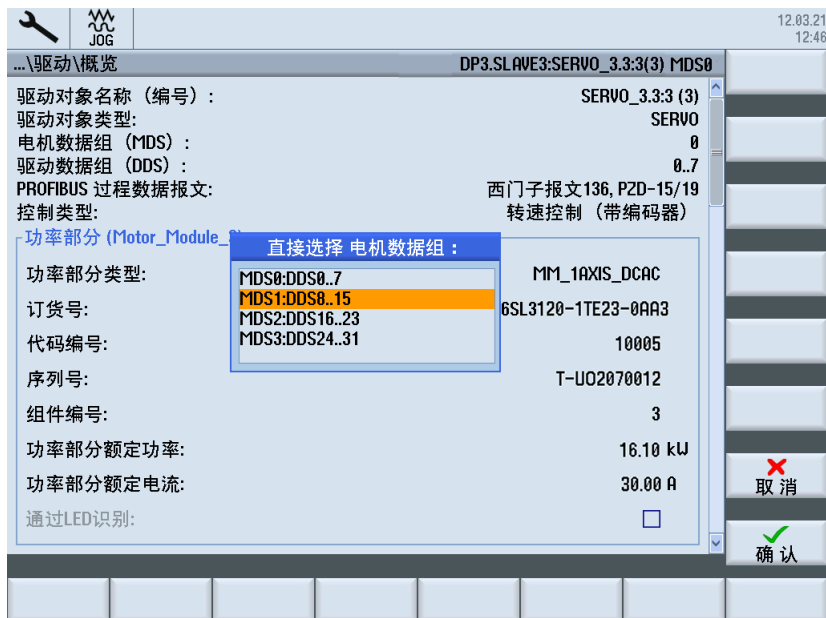


8. 按下“完成 >”接收修改。



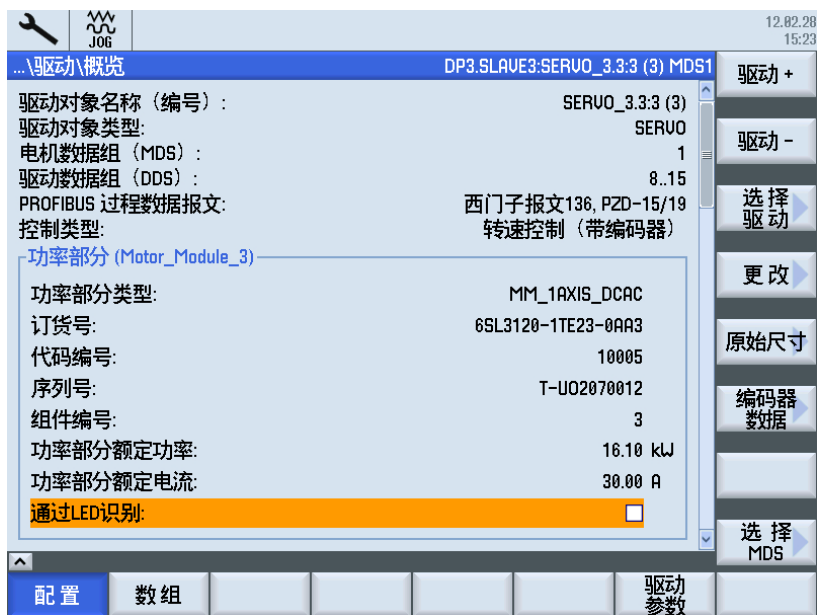
9. 点击“确定”，对数据进行非易失性存储。
参数的保存/写入可能需要持续几分钟。

10. 显示驱动对象对话框“驱动”→“概览”。现在可使用“选择 MDS”软键。按下“选择 MDS...”。



在直接选择中选择电机数据组，此处例如 MDS1。

11. 按下“确认”，显示驱动对象的新电机数据组。



现在便完成了驱动调试。

5.6.3 删除数据组

前提条件

执行删除数据组操作须满足以下前提条件：

- MDS 中 DDS 数量 > 1 \Rightarrow 可删除 DDS。
- MDS 数量 > 1 \Rightarrow 可删除 MDS。

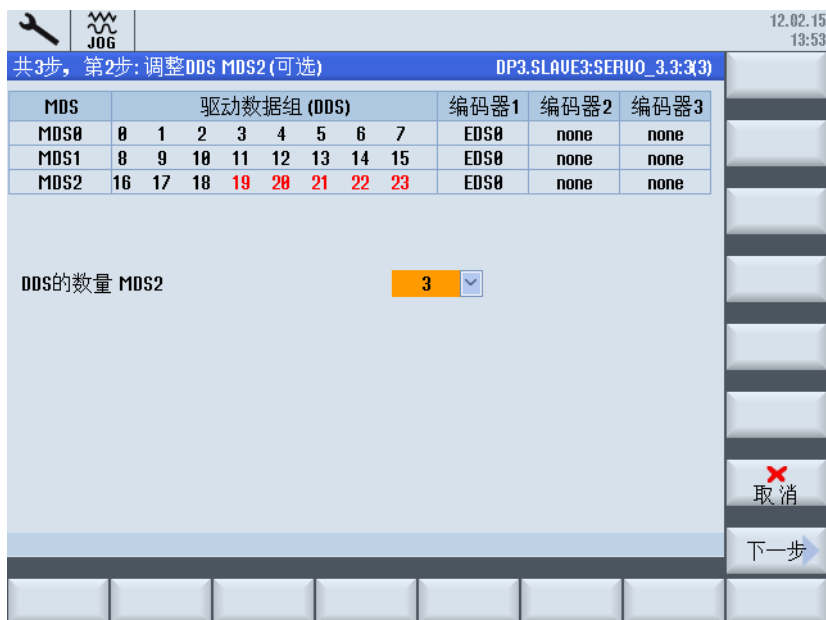
删除数据组的操作步骤

1. 选择“删除数据组”。



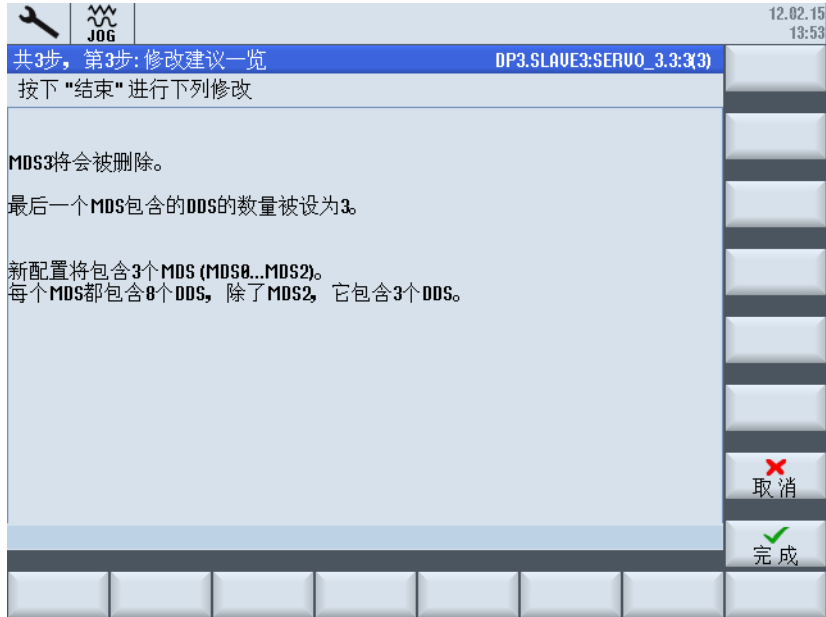
在第一列中通过钩号标记相应行，从而选中需要删除的 MDS。可选择多个数据组。

2. 按下“下一步 >”。



可选择输入 MDS 中需要保留的 DDS 驱动数据组的数量，从而将剩余的 DDS 从 MDS2 删除：在 MDS 中输入 3 DDS 则会删除 5 DDS。

3. 按下“下一步 >”。



按下“取消”中断此步骤，并显示包含 MDS0 ... MDS3 的数据组概览。

4. 按下“完成 >”接收修改。
数据会被写入参数。之后显示结果：



退出此对话框时按下“是”确认询问，从而对数据进行非易失性存储。

5.6.4 修改数据组

前提条件

若对驱动对象（SERVO）分配了超过一个编码器，则创建附加的编码器数据组（EDS0...2）。

修改数据组

选择“修改数据组”软键，修改针对相应编码器的编码器数据组分配：



这样便激活了编辑功能，用于对针对编码器的 EDS 分配进行后续调整：

5.6 配置数据组

12.03.21
12:46

修改 数据组 DP3.SLAVE3.SERVO_3.3:3(3)

MDS	驱动数据组 (DDS)	编码器1	编码器2	编码器3
MDS0	0	EDS0	EDS1	none
MDS1	1	EDS0	EDS1	none

EDS0
EDS1

取消

确认

优化驱动

6.1 控制性能

功能一览

使用功能“自动伺服优化”可以为电流环、转速环和位置环加前馈，进行优化。目标是使用 SINUMERIK Operate: 来精整进给轴或主轴。

- 频率响应测量
- 转速环计算
- 电流设定值滤波器设定
- 位置环计算
- 验证优化后控制环的实际特性
- 前馈控制类型：等转速控制时间
- 轨迹插补

闭环控制

被控对象上有三个相连的控制闭环：

- 电流环
- 转速环
- 位置环

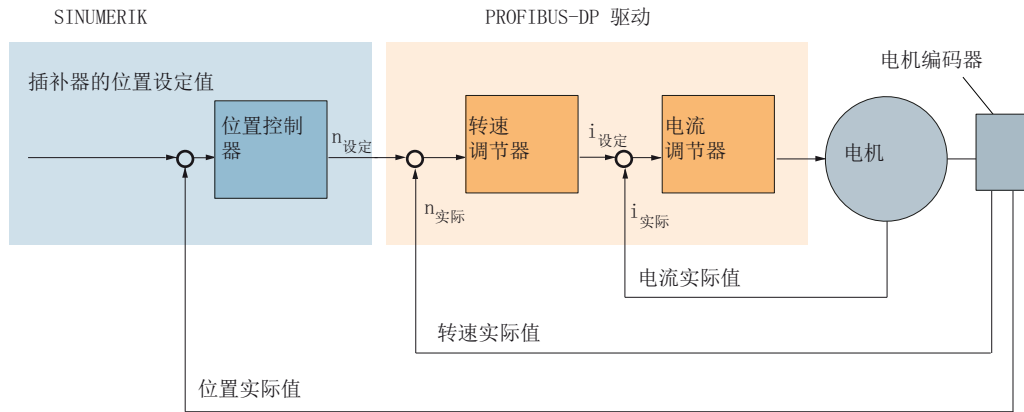


图 6-1 被控对象

控制环是依次优化的：首先是电流控制器，然后是转速控制器，最后是位置控制器。速度、加速度和急动度的动态限幅在插补器中设置。转速环通常由电机编码器来构成闭环；而位置环既可由电机编码器，也可由直接测量系统来构成闭环。

前提条件

在开始自动优化前，须设置激活动态刚性控制：

MD32640[0] \$MA_STIFFNESS_CONTROL_ENABLE = 1

位置环因此可以更短的周期时间工作。使用动态刚性控制时还须设置以下机床数据：

MD32110 \$MA_ENC_FEEDBACK_POL = 1

需要反转极性时，必须在驱动中通过参数 **p0410** 设置编码器实际值取反：

- **p0410[0]=1** 转速实际值取反
- **p0410[1]=1** 位置实际值取反

在主轴上使用“自动伺服优化”功能时，可能会缺少伺服使能 PLC 信号。机床厂商应向用户提供使能该信号的方法，以使用户能进行优化。比如：可以通过一个特殊的按键或者设置一个 PLC 状态标记来给出使能。

在优化过程中需要修改特定机床数据。参数值右侧显示了如何激活机床数据修改。

- po** 机床数据修改只在在对 NCK 执行复位操作后才生效。
- re** 机床数据修改只在按下“RESET”键后才生效。
- cf** 机床数据修改只在按下软键“机床数据有效”后才生效。
- so** 机床数据修改立即生效。

保存驱动数据

在驱动上电或执行了驱动复位操作后，SINAMICS S120 的数据会从保存文件载入对应的驱动。必须对这些数据执行保存操作，以便在下次上电或执行驱动复位操作后驱动数据修改仍能永久保存。

说明

NCK 复位操作对驱动数据没有影响。

操作步骤：

1. 按下软键“驱动参数”，然后按下软键“保存/复位”。
2. 按下软键“保存”，最后按下软键“驱动系统”。

屏幕左下方会显示提示，指出当前保存进度。保存操作结束后，屏幕上会显示以下提示：“所有驱动对象已保存”。

文档

更多信息请参见以下手册：

- SINUMERIK Operate 调试手册 (828D_IH9)，章节“跟踪”
- 功能手册之基本功能分册；章节“速度，设定值/实际值系统，闭环控制（G2）”
- 功能手册之扩展功能分册；章节“补偿（K3）”

另见

下文详细介绍了用 SINUMERIK Operate 优化控制环的各个步骤：

- 自动伺服优化 (页 186)
- 轴急动度 (页 199)
- 转矩使用率 (页 208)
- 主轴优化 (页 212)
- 圆度测试 (页 223)
- 摩擦补偿 (页 100)

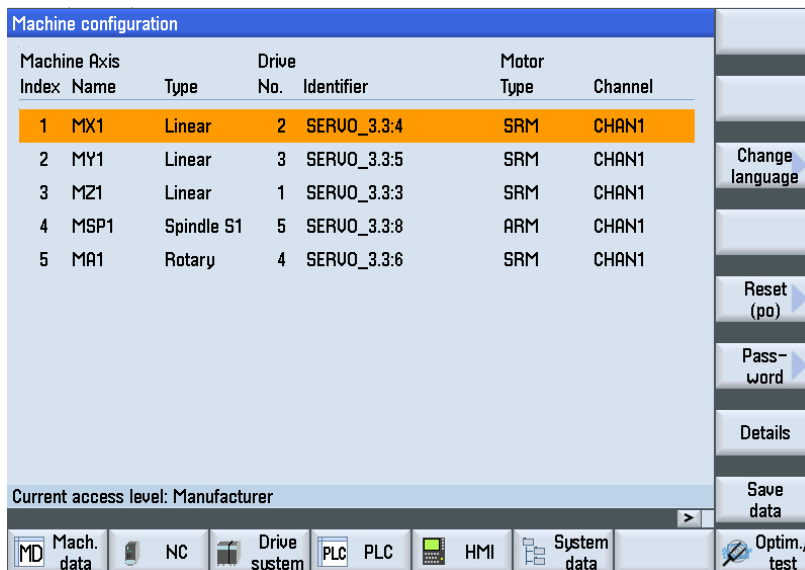
6.2 自动伺服优化

6.2.1 启动前的选项

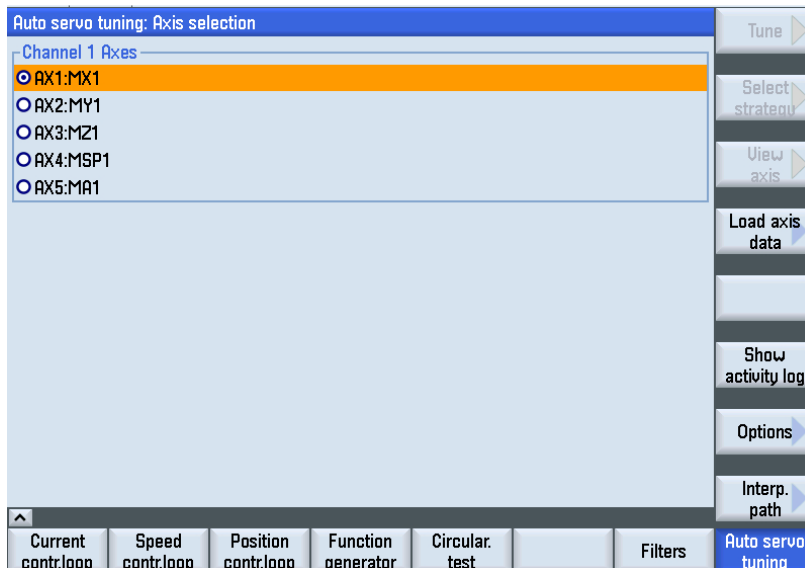
启动功能

步骤:

1. “自动伺服优化”功能在操作区“调试”中按下软键“优化/测试”启动。



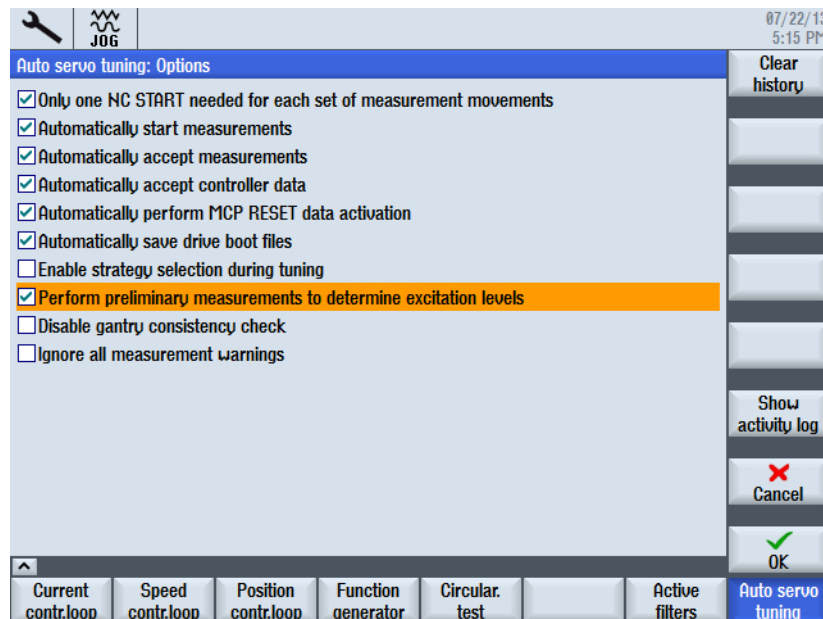
2. 用方向键浏览，然后按下<SELECT>键选中轴，此处选择 X1 轴：



对话框会显示，该轴是否经过优化。经过优化的轴会标有优化时间和日期。

选择选项

按下软键“选项”，检查是否已经设置了所有自动优化所需的选项。设置了所有选项后，只需按下一个软键，便可自动完成优化。按照屏幕上的提示操作。



选项“允许在优化期间选择方案”只建议专家使用。

6.2.2 选择优化方案

“轴”方案的缺省设置

通过软键“选择方案 >”选择用于轴、转速控制器和位置控制器的优化方案。推荐使用方案 102、303 和 203:

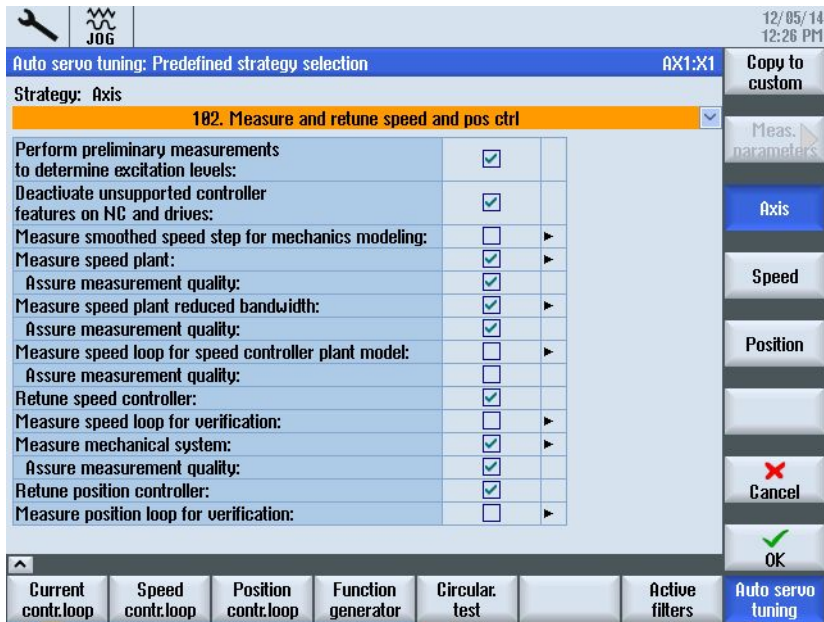


图 6-2 “方案：轴” - 缺省设置 102

在“轴”方案中选择需要优化的控制器。此时也可对经过优化的闭环进行测量，以检查结果。选择了“用户自定义的方案（108）”时，所有选项都激活并可进行设置：

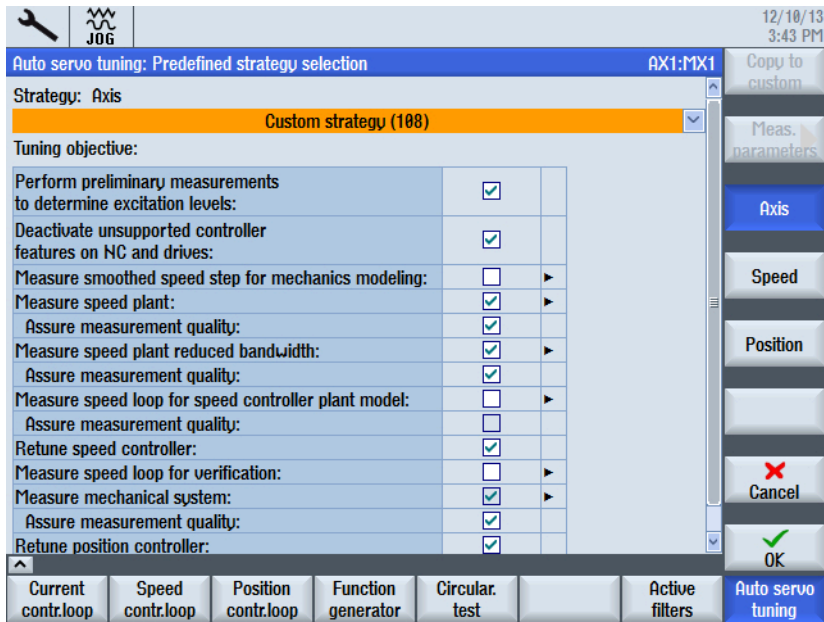


图 6-3 “方案：轴” - 示例 (108)

选择列表中还有其他方案并可与优化目标进行组合。通过“调整方案”可以激活并设置各个选项。

“转速” 示例：方案 303

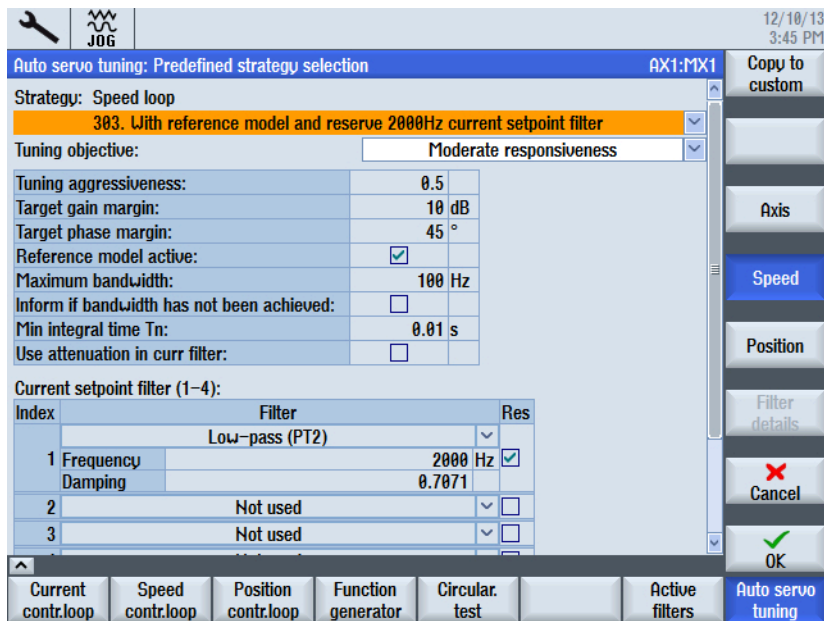


图 6-4 “方案：转速/速度控制器” 示例

重要设置：优化程度和最小积分时间 T_n 。

- 优化程度：

该参数基于稳定性限值确定了 K_p 和 T_n 的设置。在该设置下，会预设相位裕度和振幅裕度。

 - 缺省值 = 0.5
 - 最小值 = 0 [最大稳定性]
 - 最大值 = 1 [最小稳定性]
- 最小积分时间 T_n ：

该参数可避免自动伺服优化将转速环的积分时间设置过低。如果自动伺服优化设置的 T_n 时间比该参数更低，系统将采用该参数的值。

如果可能出现质量或转动惯量的变化，较高的积分时间有助于更稳定的控制。这适用于直接驱动和车床的主主轴。

 - 缺省值 = 10 ms
 - 最小值 = 0.5 ms
 - 最大值 = 100 ms

“位置” 示例：方案 203

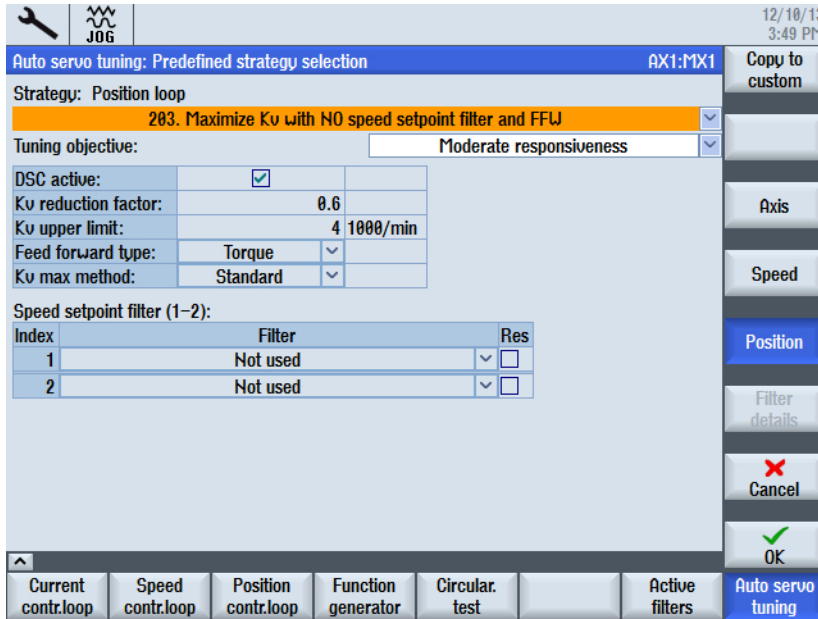


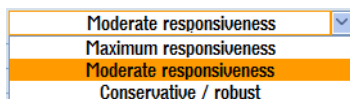
图 6-5 “方案：位置环” 示例

Kv 降低系数和 Kv 上限值取决于所选的优化目标。

- **Kv 降低系数：**
 该参数会降低由自动伺服优化计算出的最大 Kv。在位置环中不可以有过冲。最大 Kv 是尚不会导致过冲的值。为使调整更稳固（防止出现机械变化），可以降低可能的最大 Kv 值。“正常控制”优化目标下设置的裕度是 40%。此外，Kv 被限制在 4 mm/min 内。
 - 缺省值 = 0.6
 - 最小值 = 0.1
 - 最大值 = 1（无降低）
- **前馈控制类型：** 建议采用转矩前馈控制。
 使用报文 136 时，转矩前馈控制可实现最高轮廓精度。

优化目标

通过选择优化目标调整转速控制器和位置控制器的预设置。



- **快速控制:**

转速控制器和位置控制器以最大增益系数和最低稳定性进行优化。

应用：需要最大程度地降低阻力的高速加工（阻力比如有：摩擦力、传动带齿轮、强大的切削力，例如：钛加工中）。高速加工推荐采用直线电机。

前提条件：机床必须具有刚性结构；运动质量不能有明显变化。

- **正常控制 (= 缺省设置) :**

对于大多数机床和应用，该控制器动态响应已经足够。设置比“快速控制”更耐用。

应用：轴的转矩或负载量只需改变一点点，因为该设置适用于很多应用。

- **稳定控制:**

仅选择较小的控制器增益，以确保最高的稳固性。

对转速控制器进行优化，使其达到最大阻尼来避免振动并达到良好的位置控制器增益。

应用：推荐用于轴机械或负载量变化较大的机床。也适用于响应振动的轴，例如：车削机床上的主主轴或负载量较大的大型轴。

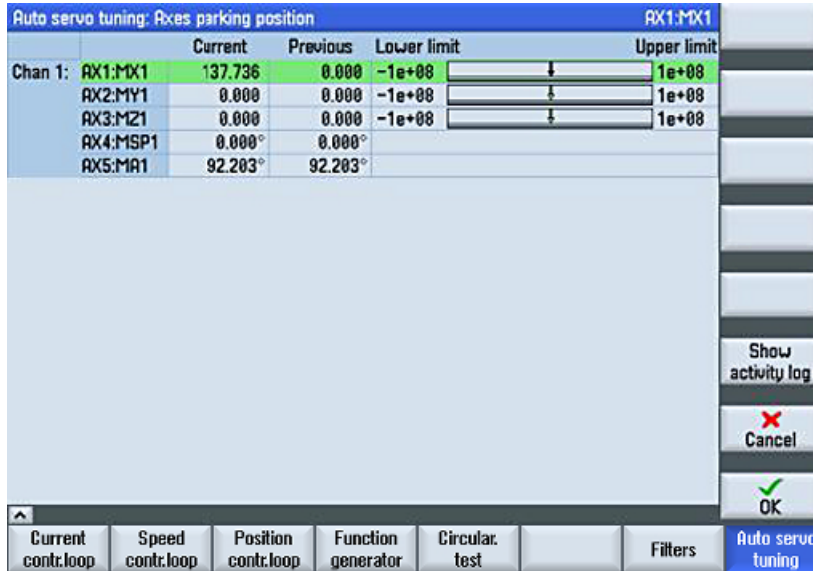
推荐用于对加工时间要求较低的应用。

6.2.3 启动自动伺服优化的步骤

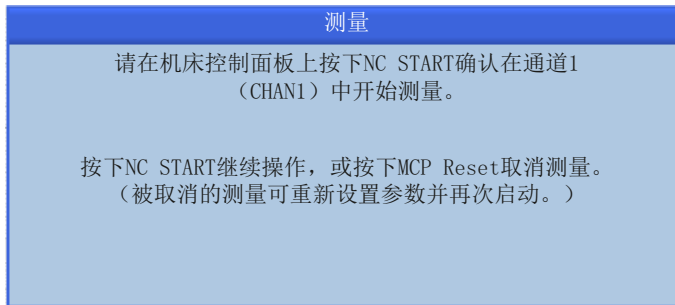
启动自动伺服优化

操作步骤：

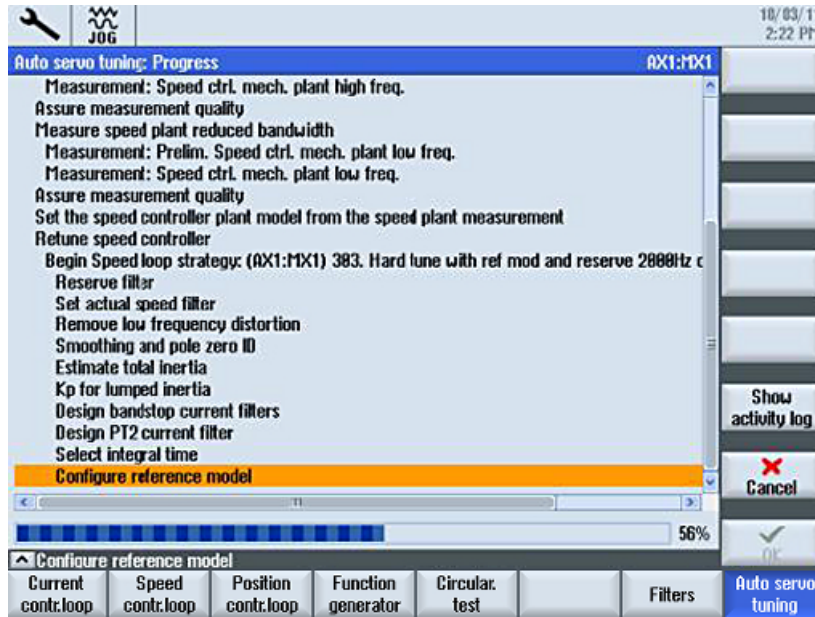
1. 确认需要优化的轴处于安全位置。如果不处于安全位置，在 JOG 模式中将轴移动到其行程中心：



2. 按下“确认”。之后界面上会弹出提示，要求按下机床控制面板上的“CYCLE START”键：



3. 在自动伺服优化的后续过程中还会弹出此类提示。需要多次按下“CYCLE START”键以开始下一个测量。



6.2 自动伺服优化

4. 在优化结束后会显示以下画面，在其中可以检查优化后的新值和优化前的原始值。
 示例：位置控制器

Auto servo tuning: Controller review AX1:MX1

Position loop
 Channel: 1 Drive data set: DDS0
 Axis: AX1:MX1 Axis param set: 1

	Manual	Auto tuned	Original
Ku factor:	17.78	17.78	2 1000/min
Feed forward type:	Speed	Speed	Speed
Equivalence time for FFW:	6.284e-4	6.284e-4	8e-3 s
Inertia for torque FFW:	0	0	0 kgm ²
DSC active:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Estimated dead time:	0	0	0 s

Manual filters:

	Frequency	Damping	Numerator frequency	Numerator damping	Res	EN
SS PT1	3999				<input type="checkbox"/>	<input type="checkbox"/>
SS PT1	3999				<input type="checkbox"/>	<input type="checkbox"/>

SS=Speed setpoint
 Auto tuned filters: None
 Original filters: None

Buttons: Speed ctrl, Bode plots, Machine data, Show activity log, Cancel, Accept, Auto servo tuning

示例：转速控制器

Auto servo tuning: Controller review AX1:MX1

Speed loop
 Channel: 1 Drive data set: DDS0
 Axis: AX1:MX1 Axis param set: 1

	Manual	Auto tuned	Original
Proportional gain Kp:	0.1611	0.1611	0.008612 Nms/rad
Integral time Tn:	5e-3	5e-3	0.01 s
Reference model active:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Reference model frequency:	290.2	290.2	0.5 Hz
Reference model damping:	0.7071	0.7071	1
Reference model time:	0	0	0 s
Act speed filter time const:	0	0	0 s
Estimated total inertia:	9.065e-5	9.065e-5	9.065e-5 kgm ²
Motor inertia:	2.8e-5	2.8e-5	2.8e-5 kgm ²

Manual filters:

	Frequency	Damping	Numerator frequency	Numerator damping	Res	EN
CS PT2	2003	0.707			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
CS 2nd order	1533	0.6156	1538	0.0277	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CS PT2	1993	0.7			<input type="checkbox"/>	<input type="checkbox"/>
CS PT2	1993	0.7			<input type="checkbox"/>	<input type="checkbox"/>

Buttons: Position ctrl, Bode plots, Machine data, Show activity log, Cancel, Accept, Auto servo tuning

按下软键“接收”后，采用新值。

5. 以下信息会提示还有其他轴需要进行优化，以获得良好的插补结果：

自动伺服优化

提示：所有轴的优化完成后，还需采取额外措施确保轴在轨迹插补中相互协调。

更多详细信息参见记录。

优化结果

接着显示优化的时间和日期：

Auto servo tuning: Axis selection		AX1:MX1
-Channel 1 Axes		
<input checked="" type="radio"/>	AX1:MX1	Tuned: 11/15/11 9:53:37 AM
<input type="radio"/>	AX2:MY1	Untuned
<input type="radio"/>	AX3:MZ1	Untuned
<input type="radio"/>	AX4:MSP1	Untuned
<input type="radio"/>	AX5:MA1	Untuned

其他轴可按同样方式优化：

Auto servo tuning: Axis selection		AX5:MA1
-Channel 1 Axes		
<input type="radio"/>	AX1:MX1	Tuned: 11/15/11 9:53:37 AM
<input type="radio"/>	AX2:MY1	Tuned: 11/15/11 10:07:30 AM
<input type="radio"/>	AX3:MZ1	Tuned: 11/15/11 10:11:39 AM
<input type="radio"/>	AX4:MSP1	Tuned: 11/15/11 10:27:50 AM
<input checked="" type="radio"/>	AX5:MA1	Tuned: 11/15/11 10:31:00 AM

6.2.4 开始轨迹插补

引言

插补表明参与轮廓加工的轴必须具有相同的跟随误差。在对单轴进行优化后，各个轴具有不同的控制特性，这种特性只对于轴本身而言是最佳的。为确保各轴有相同的跟随误差，必须以轴组中的“最慢轴”为准来调整控制特性。这其中包含 K_v 系数和前馈控制方式。

插补轴组的优化方案选择

在此为轨迹插补确定方案。

缺省设置 1103: “不降低转速控制器增益的调整”

- **(1103) 不降低 Kp 的轨迹优化 (推荐)**

特别推荐采用此方案: 这样便可实现对所有前馈控制类型的标准调整。轴优化的转速控制器比例增益 Kp 不会降低。

其中会调整以下参数:

- Kv 系数 (MD32200)
- 前馈控制的等效时间常数 (MD32800 或 MD32810)
- 积分时间 (p1462)
- 参考模型 (p1433 ff.)
- 所有转速设定值筛选条件
- 所有位置设定值筛选条件 (MD32400 和 MD32900) 失效。
- 前馈控制方式 (MD32620) 调整为最简单的模式。

如果一根轴在插补组中采用的是转速前馈控制, 其他轴采用的是转矩前馈控制, 则系统会将所有轴切换为转速前馈控制。如果插补组中没有为轴选择前馈控制的话, 也同样执行上述操作。然后所有轴只能设置为位置控制模式。

- **(1104) 用户自定义 Kp 降低幅度的轨迹优化**

所有调整与 1103 中一样。此外, 用户可以输入转速控制器比例增益 Kp 的降低系数。

- 调整有效 Kp: ✓✓
- Kp 最小的降低系数: 根据需要才可降低 Kp 的系数。
- 主轴可能会限制其他轴: 只要主主轴是动态性能最弱的轴, 主主轴便可导致伺服轴 Kp 值降低。
- 其他轴可能会限制主轴: 如果伺服轴是动态性能最弱的轴, 也会导致主轴 Kp 值降低。

一般设置

“轨迹优化”的设置:

- 动态刚性控制(DSC)

可为所有轴激活或取消动态刚性控制

(MD32640 \$MA_STIFFNESS_CONTROL_ENABLE, 简称 DSC)。激活 DSC 后, 便可使用更高的 Kv 系数。

- 前馈控制方式

- 如果所有轴的优化都加了转矩前馈, 则在轨迹插补优化中该设置会保留。
- 如果各轴的前馈模式设置不同, 有的是 3, 有的是 4, 则所有轴统一使用 MD32620 \$MA_FFWMODE = 3。
- 如果某根轴的优化中没有加前馈, 则所有轴统一使用 MD32620 \$MA_FFWMODE = 0。

“单轴最佳”的设置：

- 计算 K_v 系数，所有轴统一使用其中的最低值：始终调整 K_v 系数！
- 比例增益 K_p /转动惯量，用于轴的后期调整。
- 所有轴都调整积分时间 T_n 。
- 参考模型（转速控制器）激活，即：调整参考模型。
- 计算转速环的等效时间常数，所有轴统一使用其中的最高值。

MD32402 \$MA_AX_JERK_MODE = 0 在单轴急动度限制激活时不激活滤波器类型。

轨迹插补

下一步是选择需要进行轨迹插补的轴。

操作步骤：

1. 按下软键“轨迹插补”。一起插补的轴属于“通道 1”。
2. 按下软键“编辑&优化”。
3. 只能选择可以一起插补的轴。

本例选择 X 轴、Y 轴、Z 轴和主轴：

Auto servo tuning: Interpolation paths	
Channel 1 Path	
<input checked="" type="checkbox"/> AX1:MX1	Tuned: 11/15/11 9:53:37 AM
<input checked="" type="checkbox"/> AX2:MY1	Tuned: 11/15/11 10:07:30 AM
<input checked="" type="checkbox"/> AX3:MZ1	Tuned: 11/15/11 10:11:39 AM
<input checked="" type="checkbox"/> AX4:MSP1	Tuned: 11/15/11 10:27:50 AM
<input type="checkbox"/> AX5:MA1	Tuned: 11/15/11 10:31:00 AM

4. 系统会计算出新设置并显示在屏幕上供检查:

Auto servo tuning: Interpolation path review				
Channel 1 Path Path optimal:				
	AX3:MZ1	AX2:MY1	AX4:MSP1	AX1:MX1
Kv factor:	12.79	12.79	12.79	12.79 1/min
Proportional gain Kp:	0.1987	0.1498	5.938	0.1611 Nms/rad
Kp/inertia:	1329	1651	591	1777 rad/s
Integral time Tn:	5e-3	5e-3	0.00618	5e-3 s
Reference model active:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Reference model frequency:	215.1	215.1	149.3	215.1 Hz
DSC active:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Feed forward type:	Speed	Speed	Speed	Speed
Equivalence time for FFL:	0.00156	0.00156	0.00156	0.00156 s
Axis optimal:				
	AX3:MZ1	AX2:MY1	AX4:MSP1	AX1:MX1
Kv factor:	15.63	18.23	12.79	17.78 1/min
Proportional gain Kp:	0.1987	0.1498	5.938	0.1611 Nms/rad
Kp/inertia:	1329	1651	591	1777 rad/s
Integral time Tn:	5e-3	5e-3	0.00618	5e-3 s
Reference model active:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Reference model frequency:	215.1	276.4	149.3	298.2 Hz

5. 按下“接收”和“确认”，采用新设置。

Auto servo tuning: Interpolation paths	
Channel 1 Path	Tuned: 11/15/11 11:17:18 AM
AX1:MX1	Tuned: 11/15/11 9:53:37 AM
AX2:MY1	Tuned: 11/15/11 10:07:30 AM
AX3:MZ1	Tuned: 11/15/11 10:11:39 AM
AX4:MSP1	Tuned: 11/15/11 10:27:50 AM
AX5:MA1	Tuned: 11/15/11 10:31:00 AM

自动优化到此结束。

6.3 优化轴急动度

6.3.1 检查轴急动度

急动度限制

下一步是设置急动度限制，优化轴的加速过程：急动度限制通过设置一段加速度变化时间，来平滑加速特性和响应特性。

急动度的单位是“距离单位/ s^3 ”，即加速度的时间变化率：加速度（距离单位/ s^2 ）除以时间（s）。

相应地，加速度变化时间的计算公式为： $t[s] = \text{加速度（距离单位}/s^2\text{）除以急动度（距离单位}/s^3\text{）}$ 。

各轴的急动度不一定要相同。设置急动度限制后，速度曲线上加速阶段开始和结束处的曲线会更加平滑，运动因此更加平稳。

- 在程序中写入 **SOFT** 后，急动度限制激活。
- 在程序中写入 **BRISK** 后，急动度限制关闭。

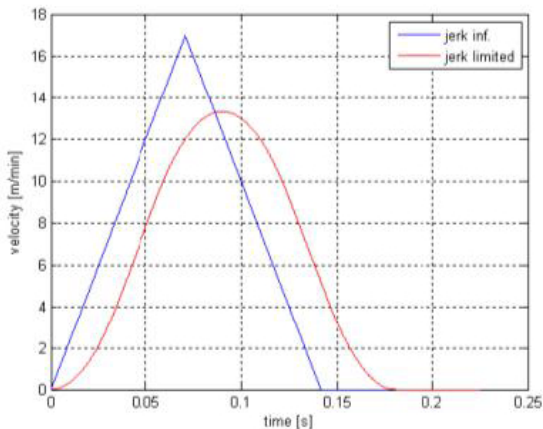
急动度的相关机床数据

建议采用 $MD20150[20] = 2$ 的缺省设置，即“SOFT”。

- $MD20150[20] = 1$: “BRISK” 在通电后激活（刚性加速模式）。
- $MD20150[20] = 2$: “SOFT” 在通电后激活（软加速模式）。

示例：

6.3 优化轴急动度



蓝色 速度曲线，不带急动度限制

红色 速度曲线，带急动度限制

轴的定位特性

编写一个包含多个进给率、多个目标位置的测试程序，然后执行测试程序，再通过“跟踪”功能便可检查轴的定位特性。

- 如果使用的是光栅尺，可以非常方便地观察出滑块的响应特性。
- 但如果使用的是电机编码器来构成位置闭环（间接反馈），则无法用跟踪功能来观察位置闭环中滑块的响应特性。

在配备 SINUMERIK 828D 的机床上，常用的急动度在 20 到 100 之间。编写测试程序时要注意编写一段足够长的行程，这样轴才能达到程序中的进给速度。

此外，程序中还应编写轴达到进给速度和目标位置后的一段停留时间，比如：0.5 秒。程序中的第一个进给速度应为最大进给速度的 50%。应在所有进给倍率值下检查轴的定位特性，直到轴达到最大进给速度。

6.3.2 轴急动度的测试程序

测试程序

下面的数控程序可以在自动模式或 MDA 模式中使用，取决于机床的具体配置。程序包含了用于激活/关闭跟踪触发器的指令：

```
FFWON
```

```
SOFT
```

```
$AN_SLTRACE=0; Reset Start Servo-Trace-Trigger  
  
LAB:  
  
G01 Y210 F10000  
  
G04 F0.5  
  
$AN_SLTRACE=1; Start Servo-Trace-Trigger  
  
Y260  
  
G04 F0.5  
  
GOTOB LAB  
  
M30
```

查看跟踪结果:

选择 <MENU SELECT>，选择操作区“诊断”，按下菜单扩展键、软键“跟踪”，按下 <CYCLE START>:

轴开始移动，系统会跟踪选中的信号 5 秒钟。之后显示结果。

机床数据:

为检查不受位置控制器影响的定位特性，应将位置控制器增益系数 K_v 设为 0。并调整定位公差，避免在测试时系统报错:

- MD32200 \$MA_POSCTRL_GAIN = 0
- MD36012 \$MA_STOP_LIMIT_FACTOR = 100
- MD36400 \$MA_CONTOUR_TOL = 20

初始化跟踪功能

按下软键“启动跟踪”后，跟踪功能启动并初始化。初始化过程取决于对话框“设置”中的选择。在本例中，跟踪功能是由程序指令启动的。按下软键“启动跟踪”，来初始化跟踪。

跟踪成功初始化，一旦通过程序设置了触发器便立即启动。

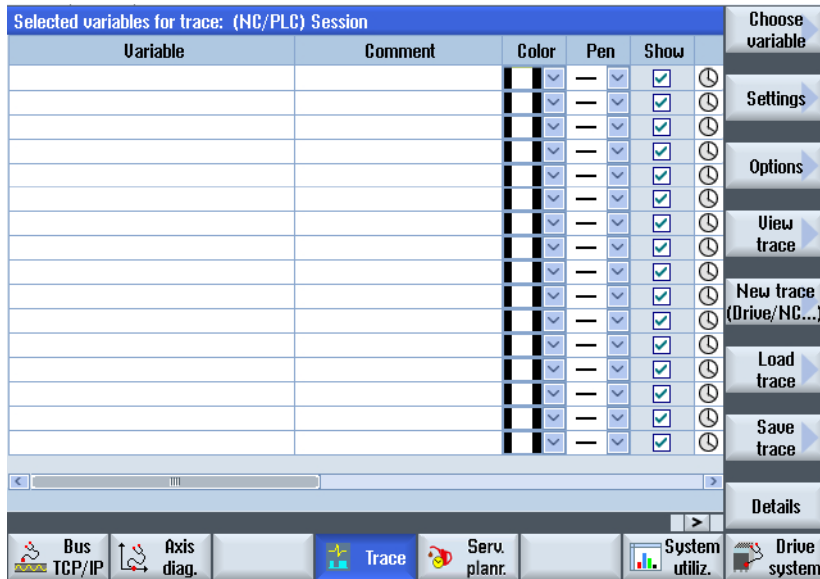
6.3.3 选择跟踪信号

选择跟踪信号

现在，需要跟踪位置设定值和位置实际值。

操作步骤：

1. 用以下按钮调用跟踪功能：
 <MENU SELECT → 操作区 “诊断” → 菜单扩展键 → 软键 “跟踪”。



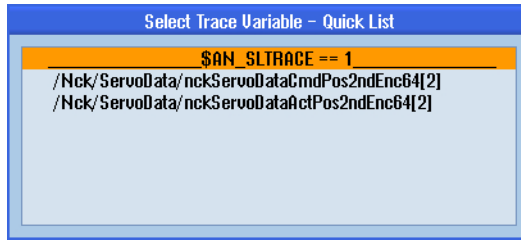
2. 选择所需变量：软键 “选择变量”。
3. 按下软键 “筛选/搜索”，以筛选变量。有以下筛选条件：
 - 系统变量
 - NC
 - 轴
 - 通道
 - 操作方式组（BAG）
 - 伺服
 - 全局用户数据(GUD)
 - PLC
 - PLC 报警
 - 全部（无筛选）
4. 设置筛选条件 “伺服”，从中选择两个变量，比如：Y 轴位置设定值和位置实际值：

Selected variables for trace: (NC/PLC) Session					
Variable	Comment	Color	Pen	Show	
ervoDataCmdPos2ndEnc64[2]	Position setpoint (64 bit)	Yellow	—	<input checked="" type="checkbox"/>	
ervoDataActPos2ndEnc64[2]	Position actual value meas.system	Green	—	<input checked="" type="checkbox"/>	
		Black	—	<input checked="" type="checkbox"/>	

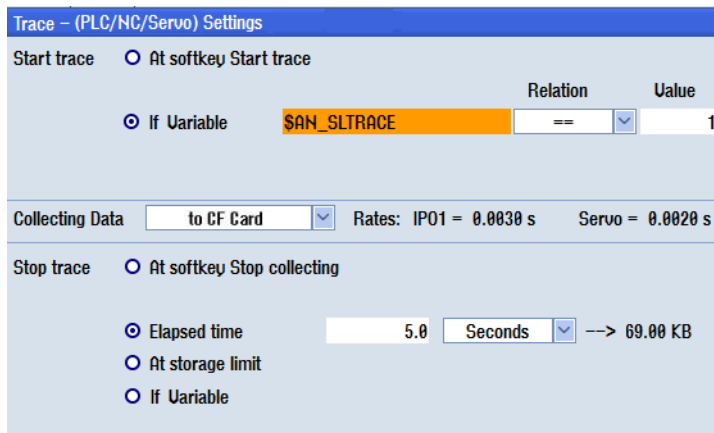
跟踪时长和触发条件

跟踪功能可以手动或自动启动。另外还可设置跟踪时长。在本例中，跟踪功能是由程序指令启动的。变量可通过软键 “快捷表” 选择。

选择选项“\$AN_SLTRACE==1”，用一个变量来触发跟踪：

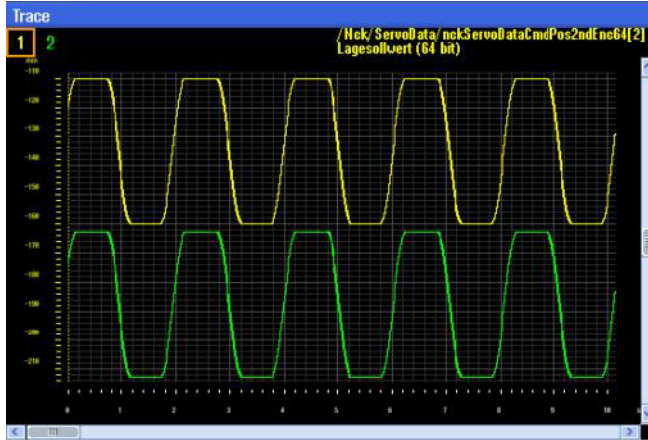


另外，跟踪还可以由选中信号的某个值来触发，或由一个通过软键“插入变量”插入的系统变量来触发。



跟踪结果

轴开始移动，系统会跟踪选中的信号 5 秒钟。之后显示结果。

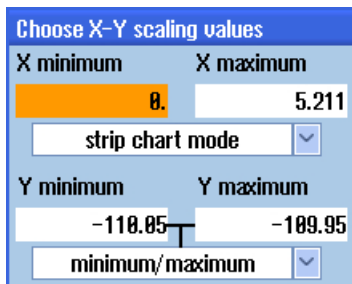


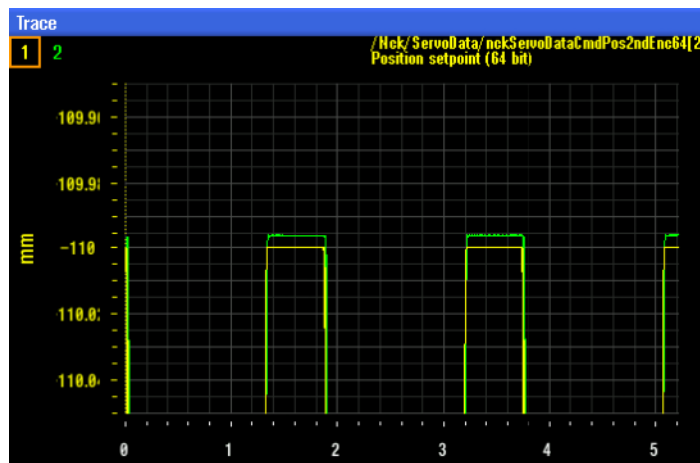
设置跟踪的显示比例

现在需要观察的是轴移动到目标位置这一阶段的跟踪结果。为观察细节，需要调整显示比例：软键“显示跟踪”→软键“比例”

为观察 -110 毫米处轴的定位过程，本例的比例设置如下。该设置得出的窗口为：上下各 100 毫米，以目标位置 -110 毫米为中心。根据机床/轴的具体情况，比例设置可与此不同。

必要时也可以调整 X 轴最小值和最大值。





细节图:



目标位置（黄线）和实际位置（绿线）不重合，这是因为位置控制器没有激活。

6.3.4 优化急动度设置

优化目标

急动度优化的目标是：轴不越过实际位置。当急动度设的过高时会出现此情况。

但急动度设的过低对机床来说也是缺点，因此这表明轴需要很长的时间才能完成定位。不加位置控制器时轴的定位过程无超调。

不加位置控制器时的定位会导致定位误差，此时轴仅仅根据进给信号移动。定位结束后会出现超调，这是由机械装置导致的，可以通过使用急动度限制排除。

首先检查以下机床数据是否正确设置：

- MD32200 \$MA_POSCTRL_GAIN = 0
- MD32610 \$MA_VELO_FFW_WEIGHT = 1
- MD32620 \$MA_FFW_MODE = 3
- MD32810 \$MA_EQUIV_SPEEDCTRL_TIME = 优化得出的结果

下例展示了不同的急动度对轴定位的影响。

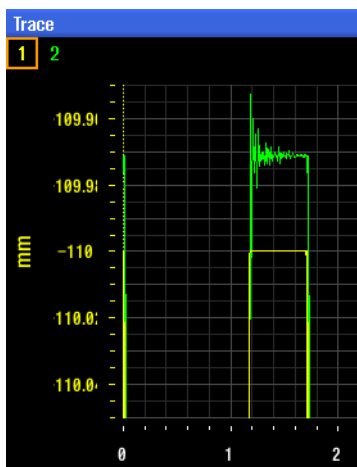
示例

急动度由轴机床数据 MD32431 设置。

示例 1: MD32431 \$MA_MAX_AX_JERK = 600 ⇒ 急动度过高。

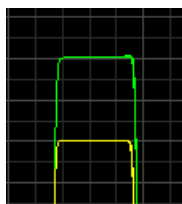
- 1. 重新启动跟踪，然后执行测试程序。
- 2. 按下软键“启动跟踪”，然后按下<RESET>和<CYCLE START>。
- 3. 调高 MAX_AX_JERK，排除超调。

下面的示意图展示了急动度不断升高，直到机械装置导致超调这一过程。



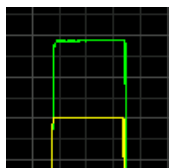
示例 2: MD32431 \$MA_MAX_AX_JERK = 2 ⇒ 急动度过低。

急动度过低时定位轮廓有圆角。



示例 3: MD32431 \$MA_MAX_AX_JERK = 65 = 优化后的急动度。

急动度经过优化后，超调极少或根本没有，定位轮廓比较尖锐。



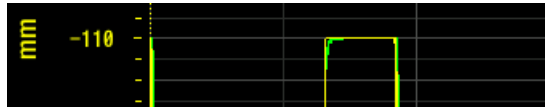
示例：加入了前馈和位置控制器的定位

急动度经过优化后，必须激活位置控制器和前馈。

必要时设置以下机床数据：

- MD20150[23] = 2 : 前馈作为缺省设置自动激活。
- MD32200 \$MA_POSCTRL_GAIN = 优化得出的结果
- MD32610 \$MA_VELO_FFW_WEIGHT = 1.0
- MD32620 \$MA_FFW_MODE = 3
- MD32810 \$MA_EQUIV_SPEEDCTRL_TIME = 优化得出的结果
- MD32431 \$MA_MAX_AX_JERK = 65
- MD36012 \$MA_STOP_LIMIT_FACTOR = 原始值
- MD36400 \$MA_CONTOUR_TOL = 原始值

对前馈和急动度进行正确优化后，可以获得以下没有超调的定位过程：



6.4 转矩使用率

检查最大轴加速度

在优化了 Kv 系数和前馈后，还需要检查最差条件下的转矩使用率，以避免超出转矩限值：

- 高速条件
- 重载条件

“跟踪”功能可用于跟踪有效测量系统的速度设定值和转矩使用率，其中转矩使用率等于“转速设定值除以转速设定值限值”。如果跟踪信号“转矩使用率” = 100%，则表明驱动达到电流限值。修改跟踪功能中的测量参数，便可查看转矩使用率。

MD32300 \$MA_MAX_AX_ACCEL 定义了轴的最大加速度。缺省设置是 [1 m/s²]（公制单位）、[39.37 inch/s²]（英制单位）和[2.77 rev/s²]（回转轴）。每根轴可以有不同的加速度设置。

如果机床厂商允许，MD32300 可设为一个适当值，使最差条件下加速阶段的转矩使用率一直保持在限值的 80% 到 90% 之间，具体百分比取决于期望的安全系数。

说明

机床的机械结构对最大加速度有一定限制。机床厂商必须规定理想的设置值！如果机床厂商没有规定该值，使用缺省值；如果机床厂商允许，可相应地调高该值。

示例：测试程序

在该测试程序中，轴要移动一段足够长的行程（300 mm），以达到程序中的快速移动速度。对测试程序作如下更改：

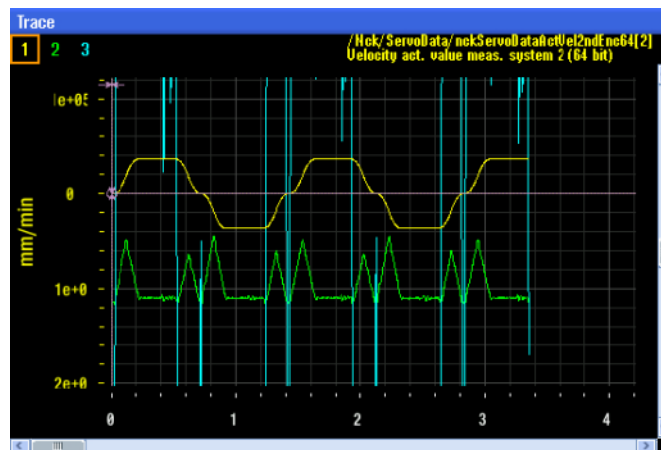
```
SOFT
$AN_SLTRACE=0; Reset Start Servo-Trace-Trigger
LAB:
G0 X10
$AN_SLTRACE=1; Start Servo-Trace-Trigger
X310
GOTOB LAB
M30
```

选择以下信号进行跟踪：

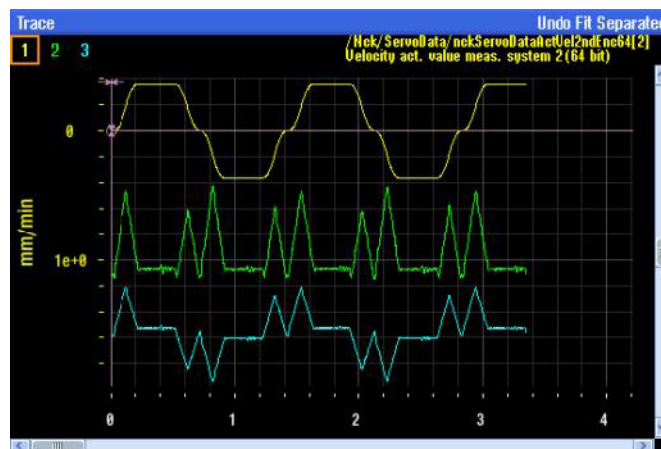
- 测量系统 2 提供的速度实际值
- 转矩使用率
- 转矩电流实际值 $i(q)$

Selected variables for trace: (NC/PLC) Session					
Variable	Comment	Color	Pen	Show	
/nckServoDataActVel2ndEnc64[2]	Velocity act. value meas. system	Yellow	—	<input checked="" type="checkbox"/>	
/nckServoDataDrvLoad64[2]	Load (m_set/m_set,limit) (64 bit)	Green	—	<input checked="" type="checkbox"/>	
a/nckServoDataActCurr64[2]	Torque-prod. Current act. Val. $i(q)$	Cyan	—	<input checked="" type="checkbox"/>	
		Black	—	<input checked="" type="checkbox"/>	

按下软键“启动跟踪” → <RESET> → <CYCLE START>:



按下软键“平铺显示”，显示所有三幅曲线图：



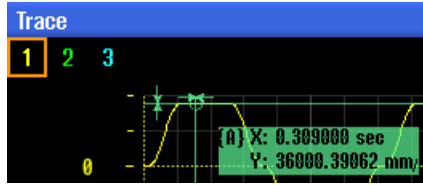
按下软键“指针”，将指针移动到需要标出最大值的曲线图上：软键“指针” → “指针 A”。

6.4 转矩使用率

按下软键“指针”，将指针移动到需要查看最大值的曲线图上：

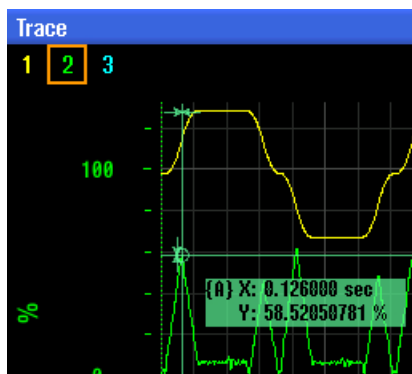
- 速度

跟踪曲线图上的指针表明，达到的最大速度为 36 m/min：



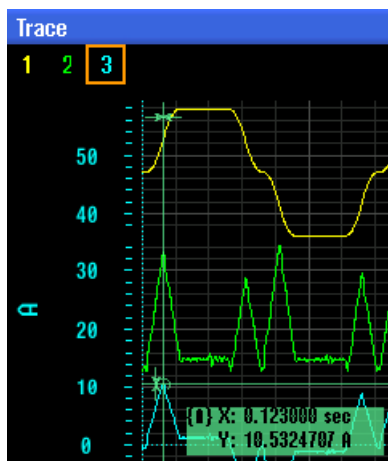
- 使用率

跟踪曲线图上的指针表明达到的最大使用率为 58.52 %：



- 电流

跟踪曲线图上的指针表明达到的最大电流为 10.53 A：



概要

达到最大使用率 58.52% 时，达到最大速度 36 m/min。信号“转矩使用率”是电流达到电流限值的百分比值。电流限值为：p0640 = [有效值 A]

p0640 被限制在 p0338（电机电流限值，有效值 A）以下，或被限制在 r0209（驱动电流限值，有效值 A）以下，取决于其中的较低值。

本例中 p0640 = 18 A。

电流值因此为 $18 \text{ A} \times 58,52/100 = 10,53 \text{ A}$ 。

本测试是在工作台上有夹具的铣床 Y 轴上进行的。58% 的使用率表明可继续增加重量。使用率可最高达到 85%。可修改加速度来控制使用率。

6.5 优化主轴

6.5.1 设置主轴的机床数据

检查机床数据设置

参考下表中的机床数据来调整用于主轴/进给轴插补的预设置的机床数据以及全部相关参数。本例为不带机械齿轮箱的主轴电机，转速为 8000 rpm。

机床数据可以设为表中指出的推荐值。可根据实际应用调整该值。

序号	标识	值	含义
MD30300	\$MA_IS_ROT_AX	1	回转轴和主轴
MD30310	\$MA_ROT_IS_MODULO	1	回转轴/主轴的模数转换
MD30320	\$MA_DISPLAY_IS_MODULO	1	回转轴和主轴的 360 度模数显示
MD32000	\$MA_MAX_AX_VELO	8000	最大轴速度
MD32010	\$MA_JOG_VELO_RAPID	60	JOG 快速移动速度
MD32020	\$MA_JOG_VELO	10	JOG 轴速度
MD32040	\$MA_JOG_REV_VELO_RAPID	60	JOG 带快速移动速度的旋转进给率
MD32050	\$MA_JOG_REV_VELO	10	JOG 旋转进给率
MD32200[0]	\$MA_POSCTRL_GAIN	x	K _v 系数：优化得出的结果
MD32200[1]	\$MA_POSCTRL_GAIN	x	
MD32620	\$MA_FFW_MODE	3	前馈控制方式
MD32640	\$MA_STIFFNESS_CONTROL_ENABLE	1	动态刚性控制
MD32810[0]	\$MA_EQUIV_SPEEDCTRL_TIME	x	转速环前馈控制的等效时间常数：优化得出的结果
MD32810[1]	\$MA_EQUIV_SPEEDCTRL_TIME	x	
MD33000	\$MA_FIPO_TYPE	3	精插补类型
MD34000	\$MA_REFP_CAM_IS_ACTIVE	0	带减速档块的坐标轴
MD34020	\$MA_REFP_VELO_SEARCH_CAM	30	回参考点速度
MD34020	\$MA_REFP_VELO_SEARCH_CAM	30	

序号	标识	值	含义
MD34040[0]	\$MA_REFP_VELO_SEARCH_MARKER	30	关闭转速
MD34040[1]	\$MA_REFP_VELO_SEARCH_MARKER	30	
MD34060[0]	\$MA_REFP_MAX_MARKER_DIST	370	到参考脉冲的最大距离
MD34060[1]	\$MA_REFP_MAX_MARKER_DIST	370	
MD35000	\$MA_SPIND_ASSIGN_TO_MACHAX	1	将主轴指定为机床轴
MD35100	\$MA_SPIND_VELO_LIMIT	8000	最大主轴转速
MD35110[0]	\$MA_GEAR_STEP_MAX_VELO	8000	自动齿轮换档时的主轴最大转速
MD35110[1]	\$MA_GEAR_STEP_MAX_VELO	8000	
MD35130[0]	\$MA_GEAR_STEP_MAX_VELO_LIMIT	8000	各个齿轮档的最大转速
MD35130[1]	\$MA_GEAR_STEP_MAX_VELO_LIMIT	8000	
MD35200[0]	\$MA_GEAR_STEP_SPEEDCTRL_ACCEL	x	转速控制方式下的加速度：优化得出的结果
MD35200[1]	\$MA_GEAR_STEP_SPEEDCTRL_ACCEL	x	
MD35210[0]	\$MA_GEAR_STEP_POSCTRL_ACCEL	x	位置控制方式下的加速度：优化得出的结果
MD35210[1]	\$MA_GEAR_STEP_POSCTRL_ACCEL	x	
MD35500	\$MA_SPIND_ON_SPEED_AT_IPO_START	2	主轴达到规定转速后使能进给
MD35550[0]	\$MA_DRILL_VELO_LIMIT	4000	攻丝时主轴最大转速
MD35550[1]	\$MA_DRILL_VELO_LIMIT	4000	
MD36000	\$MA_STOP_LIMIT_COARSE	0,4	粗准停
MD36010	\$MA_STOP_LIMIT_FINE	0,1	精准停

6.5 优化主轴

序号	标识	值	含义
MD36030	\$MA_STANDSTILL_POS_TOL	5	静态公差
MD36040	\$MA_STANDSTILL_DELAY_TIME	1	停止监控延迟时间
MD36050	\$MA_STOP_ON_CLAMPING	1	夹紧公差
MD36060	\$MA_STANDSTILL_VELO_TOL	2	阈值速度/转速 “进给轴/主轴静止”
MD36200	\$MA_AX_VELO_LIMIT	8800	速度监控极限值
MD36200	\$MA_AX_VELO_LIMIT	8800	
MD36300[0]	\$MA_ENC_FREQ_LIMIT	1000000	编码器极限频率
MD36300[1]	\$MA_ENC_FREQ_LIMIT	1000000	
MD36400	\$MA_CONTOUR_TOL	30	轮廓监控公差带
p1433	SPEED_CONTROLLER_REFERENCE_MODEL_NATURAL_FREQUENCY	x	转速控制器参考模型固有频率：优化得出的结果

6.5.2 主轴：检查转速控制器的方式

检查转速控制器的加速度

现在要检查转速控制器的加速度。首先必须规定一段加速时间，在该时间内，NC 不控制加速度。

设置以下机床数据：

- MD35200[0] \$MA_GEAR_STEP_SPEEDCTRL_ACCEL = 9999
- MD35200[1] \$MA_GEAR_STEP_SPEEDCTRL_ACCEL = 9999
($t=V/a$ $t=9000/60s/9999$ $t=15ms$ ，因此没有影响)。

电机技术数据表

示例：异步电机 1PH8089-1VM02-0MG1

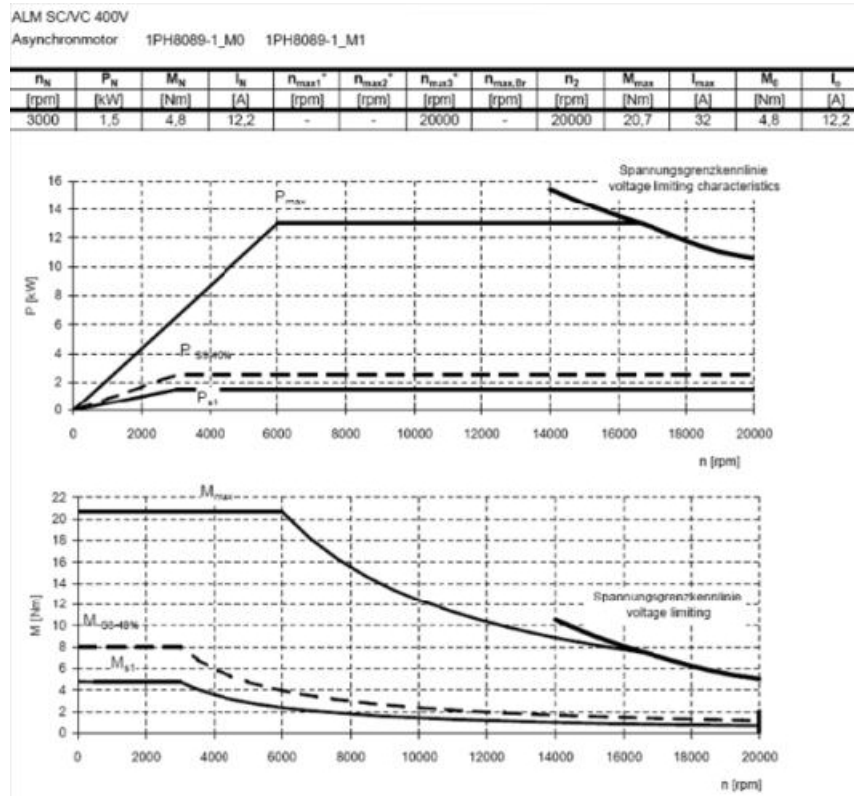


图 6-6 1PH8 异步电机

电机的初始加速和减速方式符合 S1 曲线。

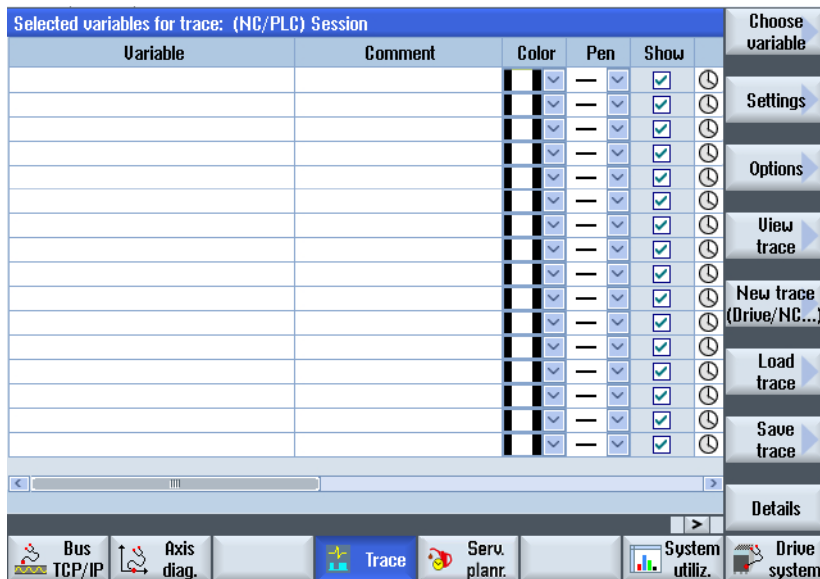
序号	名称	值
p0640	电流限值 (p0305 电机额定电流 x 1.5)	17.7 A
p1520	CO: 电动式转矩上限	4.8 Nm
p1521	CO: 再生式转矩下限	-4.8 Nm
p1530	电动式功率限值	1.5 kW
p1531	再生式功率限值	-1.5 kW

记录跟踪

操作步骤：

1. 用以下按键调用跟踪功能：

<MENU SELECT> → 操作区 “诊断” → 菜单扩展键 → 软键 “跟踪”。



2. 按下软键 “选择变量”，选择所需变量。

- 电机转速/速度实际值
- 有功功率
- 转矩电流实际值 i(q)
- 转矩/推力实际值（限幅后）

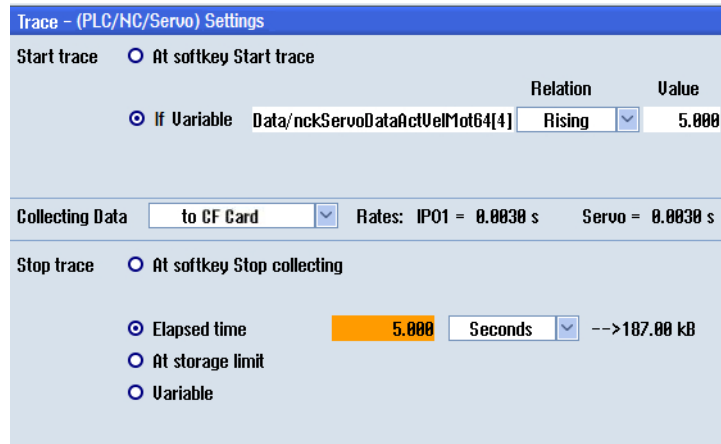
3. 选择所需变量：软键 “选择变量”

4. 按下软键 “筛选/搜索”，以筛选变量。

5. 确保在插入变量前已选择主轴：

Variable	Comment	Color	Pen	Show
nckServoDataActVelMot64[4]	Speed / velocity act. Value motor	Yellow	—	<input checked="" type="checkbox"/>
nckServoDataActPower64[4]	Active power (64 bit)	Green	—	<input checked="" type="checkbox"/>
a/nckServoDataActCurr64[4]	Torque-prod. Current act. Val. i(q)	Cyan	—	<input checked="" type="checkbox"/>
nkServoDataCmdTorque64[4]	Torque / force setpoint (limited) (f	Magenta	—	<input checked="" type="checkbox"/>
			—	<input checked="" type="checkbox"/>

6. 按下软键“设置”。



7. 按下“确认”。

8. 在 MDA 方式中输入以下程序：

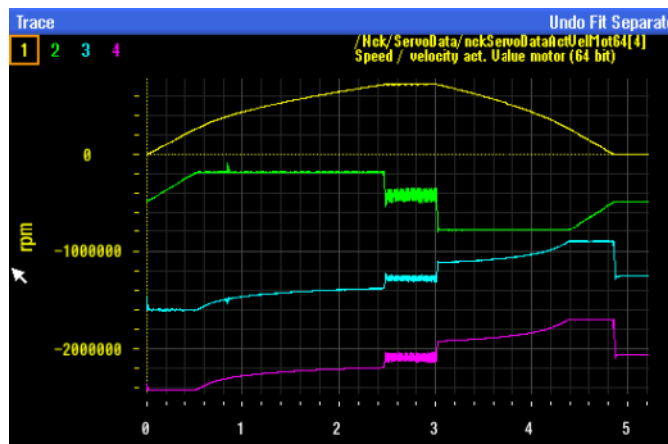
```
S8000 M03
G04 F5
M30
```

9. 初始化跟踪，执行程序：

软键“跟踪”→软键“显示跟踪”→软键“启动跟踪”→按键<RESET>→按键<CYCLE START>

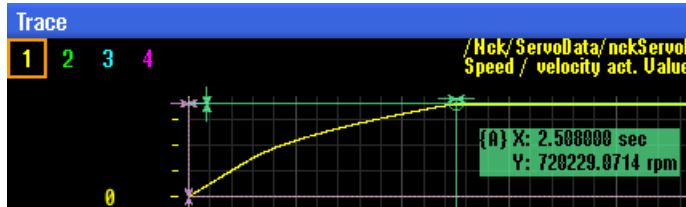
分析

在跟踪结束后，按下软键“平铺显示”：

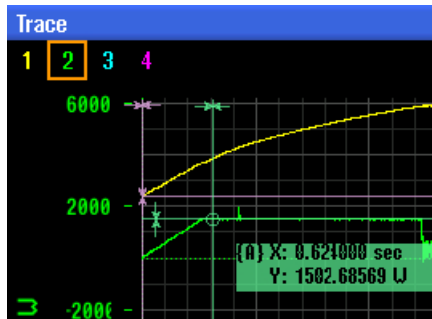


用方向键调整加速时间。

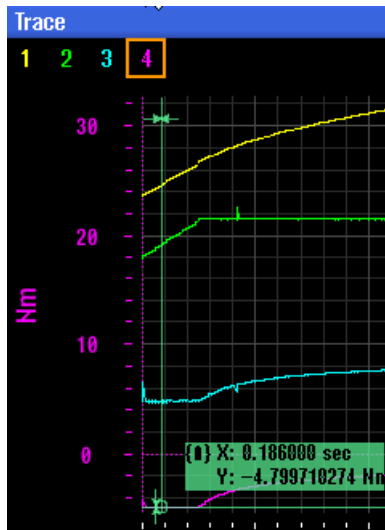
- **跟踪选择：实际转速（跟踪 1）：**
指针会移动到曲线图开头，并定位在达到设定转速的点上。
加速到设定转速需要 2.5 秒。



- **跟踪选择：实际转速（跟踪 2）**
电机加速度所需的功率在“跟踪 2”中查看：
最大可用功率 = 1.5 kW = S1 曲线



- **跟踪选择：转矩设定值（跟踪 4）**
转矩设定值可在“跟踪 4”中查看：
转矩设定值 = 4.8 Nm = S1 曲线



检查加速度

现在可以根据电机技术数据表在最大值范围内修改加速度/减速度，但也要考虑具体的应用。应输入和具体应用匹配的数值。在本例中，要求比较短的加速/减速时间。

序号	名称	值
p0640	电流限值 (p0305 电机额定电流 x 1.5)	32 A
p1520	CO: 电动式转矩上限	20,7 Nm
p1521	CO: 再生式转矩下限	-20,7 Nm
p1530	电动式功率限值	13 kW
p1531	再生式功率限值	-13 kW

在对应的驱动数据中输入电机技术数据表中的最大值：比如：异步电机 1PH8089-1_M0, 1PH8089-1_M1。

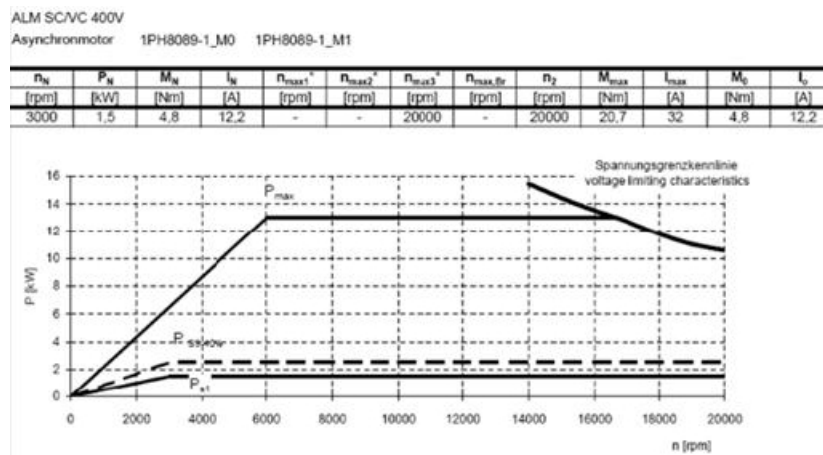
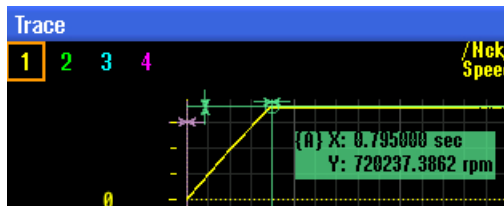


图 6-7 1PH8 异步电机

记录新的跟踪

操作步骤:

1. 执行某项主轴功能，然后执行跟踪功能。现在可以检查主轴加速到设定转速所需的时间。
选择跟踪：实际转速



2. 适当地调整 MD35200，使其满足机床厂商对主轴加速度的要求：
MD35200[0] \$MA_GEAR_STEP_SPEEDCTRL_ACCEL
MD35200[1] \$MA_GEAR_STEP_SPEEDCTRL_ACCEL

6.5.3 主轴：检查位置控制器的方式

测试位置控制器

现在也可以执行一项测试，来设置位置控制的加速度和急动度。该测试在刚性攻丝中进行。

操作步骤:

1. 检查 MD35550 \$MA_DRILL_VELO_LIMIT[0] und [1]；攻丝时的最大速度。
2. 检查设定数据 SD55484 \$SCS_DRILL_TAPPING_SET_MC[0] = 1；（在攻丝期间主轴保持在定位模式中）。
使用以下程序，在同一位置上钻出 10 个孔：

```

SUPA D0 G0 G90 Z-200
FFWON
M19
SOFT
LAB:
G331 Z-214 S4000 K1
G332 Z-200 S4000 K1
REPEAT LAB P=9
M30
        
```

3. 选择下面的变量：
 - 位置设定值（64 位）
 - 测量系统提供的位置实际值（64 位）
 - 轮廓偏差（64 位）
 - 转矩使用率（64 位）

Selected variables for trace: (NC/PLC) Session					
Variable	Comment	Color	Pen	Show	
ervoDataCmdPos2ndEnc64[4]	Position setpoint (64 bit)	Yellow	—	<input checked="" type="checkbox"/>	
ServoDataActPos1stEnc64[4]	Position actual value meas.system	Green	—	<input checked="" type="checkbox"/>	
/nckServoDataContDev64[4]	Contour deviation (64 bit)	Cyan	—	<input checked="" type="checkbox"/>	
/nckServoDataDrvLoad64[4]	Load (m_set/m_set,limit) (64 bit)	Magenta	—	<input checked="" type="checkbox"/>	
		Black	—	<input checked="" type="checkbox"/>	
		Black	—	<input checked="" type="checkbox"/>	

说明

确保在监控测量系统时选择了正确的信号，比如：测量系统 1 或 2。

4. 按下软键“设置”，设置跟踪触发条件等项目。本例选择的触发条件是：轴越过实际位置“5 度”。跟踪过程持续 10 秒。

Trace – (PLC/NC/Servo) Settings

Start trace At softkey Start trace

Relation Value

If Variable

Collecting Data Rates: IP01 = 0.0030 s Servo = 0.0030 s

Stop trace At softkey Stop collecting

Elapsed time -->190.00 kB

At storage limit

Variable

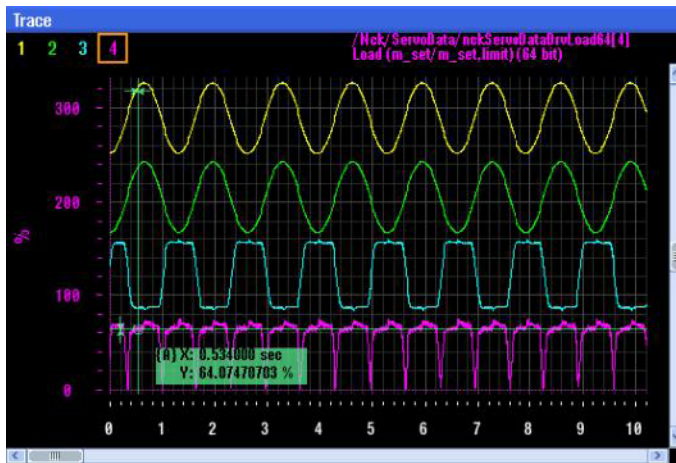
5. 启动跟踪，按下机床面板上的“CYCLE START”。

示例

本例中使用率为 64%，以提高位置环的加速度。但在攻丝期间，主轴运行可能会超过 S1 曲线，因此必须确保使用率不会达到 100%，否则会输出轮廓监控报警。

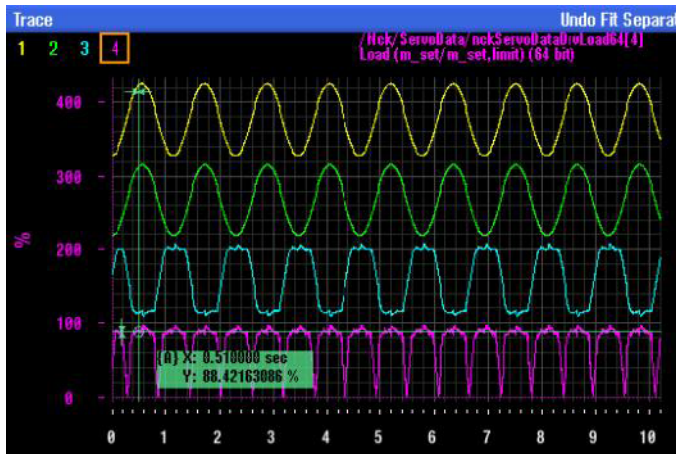
1. 调整机床数据 MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL 和 MD32431 \$MA_MAX_AX_JERK，以获得最佳结果。
2. 另外检查主轴的可调范围 p1464, p1465。

主轴：使用率 64%



本例展示了位置环加速度从 120 提高到 160 产生的影响，使用率因此从 64% 提高到 88%。如果机床是一个纯粹的攻丝机，则可以钻出 100 个螺纹孔来进行测试。接着跟踪便会重复进行，以确保使用率绝不会达到 100%。

主轴：使用率 88%



6.6 圆度测试

6.6.1 圆度测试：功能

圆度测试用于设置并评价插补轴的动态特性，以及用于分析通过摩擦补偿（常规象限误差补偿）所达到的象限过渡处（圆形轮廓）的轮廓精度。

圆度测试可用于检查共同机床轴的插补。该功能可测量以机床或直接测量系统为参照的圆。结果并不考虑机床机械系统的校准。这可帮助调试人员清楚地区分控制器优化问题与机械问题。

以下轴机床数据和参数会在该操作中进行检查：

- MD32200, MD32400, MD32402, MD32410, MD32490, MD32500, MD32510, MD32520, MD32540 MD32620, MD32640, MD32810, MD32900, MD32910, MD32930, MD32940
- p1421 bis p1426, p1400, p1433, p1434

当执行该操作时，应取消以下的位置误差补偿：

- MD32450 无反向补偿
- MD32500 摩擦补偿生效
- MD32700 编码器/主轴误差补偿
- MD32710 垂度补偿使能
- MD32750 温度补偿类型

说明

MD32450“无反向补偿”应通过外部设备，如圆度测试或千分表，进行匹配。

示例：

用于 X-Y 轴测量的 NC 测试程序：

```
FWON  
SOFT  
G90 G01 F3000 X400 Y200 Z500  
LAB:  
G91 G64 G02 X0 Z0 I10  
GOTOB LAB
```

6.6 圆度测试

M30

位置、进给率和有效平面应根据机床进行匹配！

结果

当圆度测试结果与轴组合的插补之间在实际尺寸、形状和最小 p/p 方面的差异都在合理范围内时，即达到了最佳的轮廓结果。

MDA 运行方式下的 NC 程序和圆度测试功能都可用于测量和评价该结果。圆半径和轨迹速度的“最差情况”应能实现机床能够达到的实际径向加速度。

机床制造商一般都会对测试圆的半径和进给率进行说明。

一般情况下机床制造商在圆度测试时采用的半径为 100 mm 或 150 mm，进给速度由机床制造商确定。机床制造商定义可接受结果的标准。

高速加工一般对在高速铣床上进行的圆度测试要求较高，圆半径为 10 - 25 mm，进给率为 5 - 10 m/min。对于高速铣床，如果 p/p 误差小于等于 0.010 mm 并且圆的实际尺寸与程序设定的半径相同，最差情况轨迹速度与设定相同，一般就认为结果是可接受的。

6.6.2 圆度测试：执行测量

设置参数

Circularity test measurement			
Measurement			
Axis:	Meas. system:	Absolute position:	Status:
MX1	1 Active	0.00000 mm	Inactive
MZ1	3 Active	0.00000 mm	Inactive
Parameter		Representation:	
Radius:	10.00000 mm	Resolution	0.01000 mm/grad.
Feedrate:	3000.00000 mm/min	Represent:	Mean radius
Multiplier:	1.00000		
Meas. time:	1257 ms		
Current contr.loop Speed contr.loop Position contr.loop Function generator Circular. test Filters Auto servo tuning			

欲执行测量，请输入以下参数：

- “测量”：选择要测量的两个轴和测量系统。不提供“已停止的编码器”选项。
- “参数”：在输入栏“半径”和“进给率”中设置时，应根据是否启用进给补偿的情况，输入用于控制轴进行圆周运动的零件程序中的相应数值。
- “显示”：用于图形显示的参数
 - 图表轴的“分辨率”（比例），单位[mm/Skt]
 - “显示”基于平均半径或程序设定的半径

执行测量

步骤：

1. 按下操作区域“调试”中的软键“优化/测试”→“圆度测试”。
2. 使用<SELECT>键或通过软键“轴 +”/“轴 -”选择要测量的轴。
3. 设置用于测量的“参数”：“半径”和“进给率”
在显示栏“测量时间”中会显示出由“半径”和“进给率”数值所计算出的、圆周运行时用于记录位置实际值的测量持续时间：
如果测量时间不充足，则只会显示圆的一部分。通过降低进给率可增加测量时间。即使是从静止状态启动圆度测试也同样适用。
4. 设置用于图形显示的参数：
如果计算出的测量时间超过可显示的时间范围（最大测量时间 = 位置控制器周期 * 2048），则记录时会计算对应的采样时间（n * 位置控制器周期），这样就可以显示一个完整的圆。

其它操作：

- 按下软键“启动”来开始测量。
- 按下软键“停止”来终止测量。
- 欲执行对优化的其他调整，请按下软键“优化”。
- 通过新显示的软键条，您可以**直接**进入以下区域：
 - “诊断”操作区域中的“服务轴”
 - “轴机床数据”
 - “驱动机床数据”
 - “用户视图”
- 欲保存测量参数，请按下软键“保存参数”。
- 欲以相同的参数重复执行测量，请点击软键“装载参数”。

显示图形

欲以图形方式显示测量结果，请按下软键“图形”。

6.6.3 圆度测试：示例

轴急动度限制 MD32400 \$MC_AX_JERK_ENABLE 通过一个时间常量来设置并且始终生效。

位置设定值滤波器的机床数据：

- 推荐设置 MD32402 \$MC_AX_JERK_MODE = 类型 2，类型 1 是缺省值，以保持和旧版本的兼容性。强烈不推荐设置单纯的带阻滤波器。
- MD32402 \$MA_AX_JERK_MODE（滤波器类型）和 MD32410 \$MA_AX_JERK_TIME > 0 只在 MD32400 \$MA_AX_JERK_ENABLE = 1 时生效。

优化示例 1

轴优化结束后的机床数据：

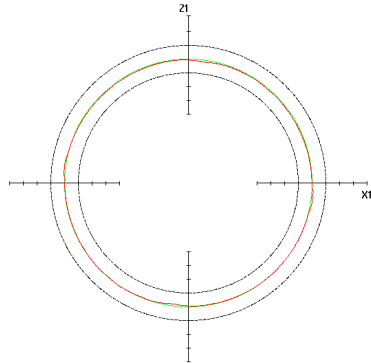
参数/机床数据		X 轴	Z 轴
MD32200	\$MC_POSCTRL_GAIN	8.500	8.500
p1460	SPEEDCTRL_GAIN1	3.01	3.89
p1462	SPEEDCTRL_INTEGRATOR_TIME_1	6.18	6.18
p1463	SPEEDCTRL_REF_MODEL_FREQ	106.3	106.3
p1440	NUM_SPEED_FILTERS	0	0
MD32610	\$MC_VELO_FFW_WEIGHT	1.0	1.0
MD32620	\$MC_FFW_MODE	4	4
MD32810	\$MC_EQUIV_SPEEDCTRL_TIME	0.0022	0.0022
MD32400	\$MC_AX_JERK_ENABLE	0	0

一般在使用优化的进给前馈时，半径的实际值都会过大。这可使用时间常数 MD32410 \$MC_AX_JERK_TIME 来修正。需要时，所有轴可使用统一的一个时间常数。

圆会显示进给前馈经过优化后的结果。但平均半径会大 0,0019 mm:

X1: 激活的测量系统

Z1: 激活的测量系统



参数

半径: 10.00000 mm

进给率: 3000.00000 mm/min

测量时间: 1257 ms

X1: 激活的测量系统

Z1: 激活的测量系统

显示

分辨率: 0.01000 mm

显示 平均半径

半径: 10.00190 mm

Delta R: 4.02698 μm

参数/机床数据		X 轴	Z 轴
MD32200	\$MC_POSCTRL_GAIN	8.500	8.500
p1460	SPEEDCTRL_GAIN1	3.01	3.89
p1462	SPEEDCTRL_INTEGRATOR_TIME_1	6.18	6.18
p1463	SPEEDCTRL_REF_MODEL_FREQ	106.3	106.3
p1440	NUM_SPEED_FILTERS	0	0
MD32610	\$MC_VELO_FFW_WEIGHT	1.0	1.0
MD32620	\$MC_FFW_MODE	3	3
MD32810	\$MC_EQUIV_SPEEDCTRL_TIME	0.0022	0.0022
MD32400	\$MC_AX_JERK_ENABLE	1	1
MD32402	\$MC_AX_JERK_MODE	2	2
MD32410	\$MC_AX_JERK_TIME	0.012	0.012

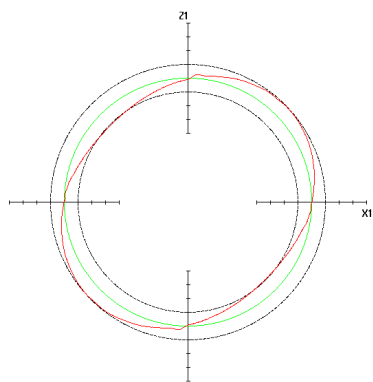
优化示例 2

该圆会显示出略有差异的时间常数对轴急动度滤波器的影响。为修正该类误差，应调整时间常数：

参数/机床数据		X 轴	Z 轴
MD32400	\$MC_AX_JERK_ENABLE	1	1
MD32402	\$MC_AX_JERK_MODE	2	2
MD32410	\$MC_AX_JERK_TIME	0.012	0.0125

X1: 激活的测量系统

Z1: 激活的测量系统



参数

半径: 10.00000 mm

进给率: 3000.00000 mm/min

测量时间: 1257 ms

X1: 激活的测量系统

Z1: 激活的测量系统

显示

分辨率: 0.01000 mm

显示: 平均半径

半径: 10.00029 mm

Delta R: 25.47002 μm

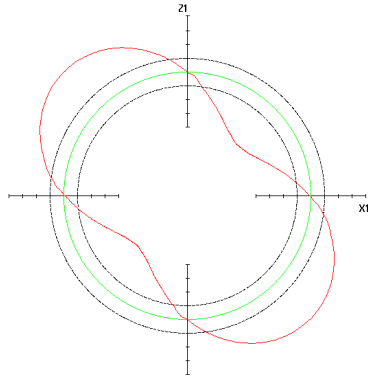
优化示例 3

该圆会显示出差异明显的时间常数对轴急动度滤波器的影响。为修正该类误差，应调整时间常数：

参数/机床数据		X 轴	Z 轴
MD32400	\$MC_AX_JERK_ENABLE	1	1
MD32402	\$MC_AX_JERK_MODE	2	2
MD32410	\$MC_AX_JERK_TIME	0.015	0.012

X1: 激活的测量系统

Z1: 激活的测量系统



参数

半径: 10.00000 mm

进给率: 3000.00000 mm/min

测量时间: 1257 ms

X1: 激活的测量系统

Z1: 激活的测量系统

显示

分辨率: 0.01000 mm

显示: 平均半径

半径: 9.98971 mm

Delta R: 75.67665 μ m

6.6.4 圆度测试：保存数据

在圆度测试时可以保存以下数据：

- **保存参数：** 所输入的参数可以保存为文件。
默认的保存路径为：user/sinumerik/hmi/log/optimization/cicular/<name>.sup
- **保存图形：** 如果图形被正确保存，则会显示信息“数据已被保存”，否则会输出一条报警“保存文件时出错”。
默认的保存路径为：user/sinumerik/hmi/log/optimization/cicular/<name>.sud
- **打印图形：** 图形会被保存为 PNG 格式的像素图形。输入一个名称<name>。名称可自由选择。默认的保存路径为：user/sinumerik/hmi/log/optimization/cicular/<name>.png

保存参数

文件格式的结构如下（[]中为附注）

H: CstPar [圆度测试参数的标识]

V: 5.0 [文件格式的版本号]

关于测量

P 1: 1 [第一根轴的编号]

P 2: 2 [第二根轴的编号]

P 3: 0 [测量系统 轴 1 - 0: 有效轴; 1: 第一根轴; 2: 第二根轴]

6.6 圆度测试

P 4: 0 [测量系统 轴 2 - 0: 有效轴; 1: 第一根轴; 2: 第二根轴]

关于参数

P 10: 30 [半径]

P 11: 3000 [进给率]

P 12: 3770 [测量时间]

P 14: 1 [乘法器]

关于显示

P 20: 10 [分辨率]

P 21: 8 [平均/程序设定的半径 - 8 = 平均半径; 9 = 程序设定的半径]

保存图形

文件格式的结构如下 ([]中为附注):

H: CstPic [圆度测试图形的标识]

V: 5.0 [文件格式的版本号]

关于参数

P 1: 30 [半径]

P 2: 3000 [进给率]

P 3: 3770 [测量时间]

P 4: 0 [测量系统 轴 1 - 0: 有效轴; 1: 第一根轴; 2: 第二根轴]

P 5: 0 [测量系统 轴 2 - 0: 有效轴; 1: 第一根轴; 2: 第二根轴]

关于显示

P 10: 10 [分辨率]

P 11: 9 [平均/程序设定的半径 - 8 = 平均半径; 9 = 程序设定的半径]

P 12: X1 [轴 1]

P 13: Z1 [轴 2]

关于中间值

P 20: 15.6632 [最大半径测量值]

P 21: 10.9326 [最小半径测量值]

P 22: 13.6694 [平均半径测量值]

P 23: 1886 [测量值数量]

关于辅助值

P 30: 1000 [精度 (1/P30)]

关于物理单位

P 40: 5370 [半径单位的文本编号]

P 41: 5381 [进给率单位的文本编号]

P 42: 6165 [分辨率单位的文本编号]

P 43: 5346 [Delta 半径单位的文本编号]

P 44: 0 [新建: 运算: 基本长度单位]

关于横坐标

Ai: [横坐标值 i : 0..P23]

关于纵坐标

Oi: [纵坐标值 i : 0..P23]

关于半径

Ri: [半径值 i : 0..P23]

维护管理器

概述

维护管理器的作用在于，通过操作界面上的维护管理器窗口或编程工具，为需要处理的任务（通常是机床维护任务）设置时间间隔和报警，具体操作包括：编辑、启动、取消和激活。

任务的数字数据以数据块的格式管理，在用户接口中提供给 PLC 用户程序、HMI 和编程工具使用。各个任务的名称，由操作软件管理、编辑，并与数字数据一同显示。

PLC 固件访问用户接口中的数据块，处理这些数据并将结果以剩余时间、警告和报警的形式再次返回到数据块中。维护管理器在 PLC 固件中每分钟处理一次。在关闭控制系统后，维护任务的实际数据会被冻结。下次接通系统后便恢复到这些持久保存的值。

然后 PLC 用户程序分析这些实际数据，并生成数字格式的报警或警告信息（包含或不包含 Power-OFF 状态）。报警处理器随后通过对应的 PLC 报警文件“oem_alarm_plc_<lng>.ts”，将这些信息转换成操作界面上显示给操作员的信息（<lng>为当前设置的语言），必要时也可记录这些信息。

接口和配置

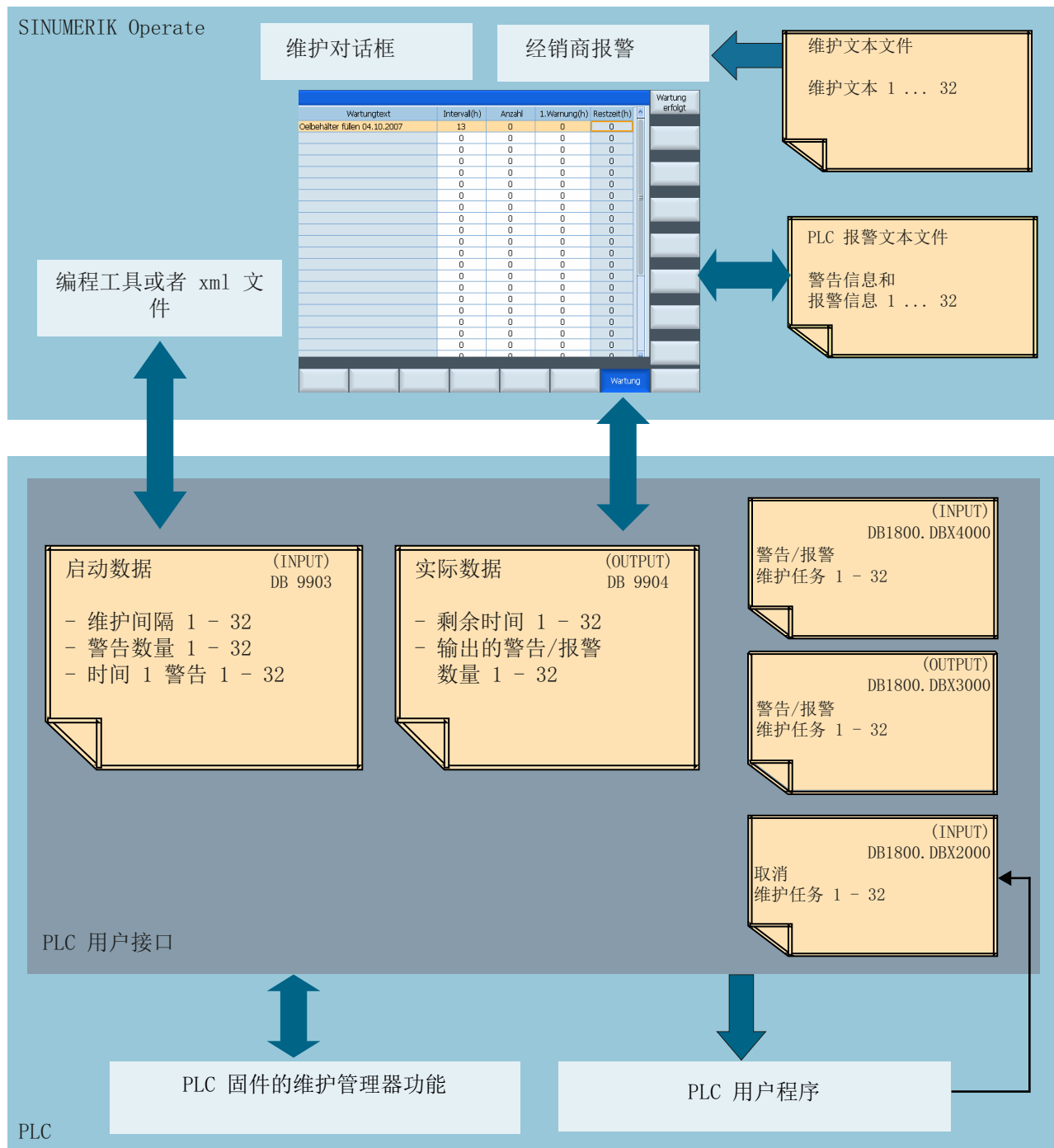


图 7-1 维护管理器 - 接口

7.1 PLC 用户程序中的接口

PLC 用户程序

PLC 用户程序用于操作用户接口。其中尤其是 DB1800 的位接口和 DB9904 中剩余时间的计算。请确保，在出现警告和报警时会显示正确的信息。

在设计显示信息时，可以选择报警响应，例如：在出现 Power-OFF 信息后机床处于“禁止”状态。

报警信息应根据 SINUMERIK 828D 的 PLC 信息规定来设计。文本由报警文本编辑器输入。之后，可以使用操作软件中的文本：

- 分析报警数据和实际数据，以便生成 PLC 警告和 PLC 报警。其他信号可以添加到分析逻辑中。
- 选择性连接禁止位和位内存或 I/O 信号。

说明

在 PLC 功能库中也包含了示例程序。机床制造商可以根据实际需要修改示例程序。

PLC 接口

在用户接口中，可以使用以下数据块：

数据块	含义
DB9903	初始数据
DB9904	实际数据
DB1800.DBB2000	取消任务
DB1800.DBB3000	报警
DB1800.DBB4000	应答
DB1800.DBB5000	应答禁止

DB9903: 初始数据

DB9903	初始数据表 [r16]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBW0	间隔 1 [h]							

7.1 PLC 用户程序中的接口

DB9903	初始数据表 [r16]
DBW2	首次警告时间 1 [h]
DBW4	待输出警告的数量 1
DBW6	保留 1
DBW8	间隔 2 [h]
DBW10	首次警告时间 2 [h]
DBW12	待输出警告的数量 2
DBW14	保留 2
...	...
DBW248	间隔 32 [h]
DBW250	首次警告时间 32 [h]
DBW252	待输出警告的数量 32
DBW254	保留 32

间隔： 执行维护的时间间隔（小时）。 经过该时间段后，任务的警告/报警位最后一次置位。

第一次输出警告的时间： 第一次输出警告的时间间隔（小时）。 该时间必须大于或等于维护间隔。

待输出警告的数量： 在发出报警前输出的警告的数量 n。 报警位最多会置位 n + 1 次，即： n 次作为警告，1 次作为报警。

预留： 用于将来扩展。

示例：

间隔 = 100

第 1 次输出警告的时间 = 80

待输出警告的数量 = 2

在任务启动后，经过 80 小时，警告/报警位第一次置位，再经过 10 小时，也就是总共经过了 90 小时后，第二次置位；总共经过 100 小时后，最后一次置位。

DB9904: 实际数据

DB9904	实际数据表 [r16]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBW0	剩余时间 1 [h]							
DBW2	待输出的警告的数量 1							

DB9904	实际数据表 [r16]
DBW4	保留_1 1
DBW6	保留_2 1
DBW8	剩余时间 2 [h]
DBW10	待输出的警告的数量 2
DBW12	保留_1 2
DBW14	保留_2 2
...	...
DBW248	剩余时间 32 [h]
DBW250	待输出的警告的数量 32
DBW252	保留_1 32
DBW254	保留_2 32

剩余时间：任务启动离期满还剩下的时间（小时）。

剩余时间 $\neq 0$ ，对应的报警位置位：警告

剩余时间 = 0，对应的报警位置位：报警

待输出的警告的数量：待输出的警告的数量 n 。如果间隔完全届满，则输出值为 $(n+1)$ ：

n = “待输出的警告的数量”

1 = 间隔届满时的报警

保留_1, ~_2: 用于将来扩展。

示例：

间隔 = 100, 第 1 次输出警告的时间 = 80, 待输出警告的数量 = 2

在任务启动后，剩余时间每小时递减。

- 在经过 80 小时后，剩余时间为 20 小时，输出警告从 0 个增加到 1 个。
- 再经过 10 小时后，也就是总共经过了 90 小时，剩余时间为 10 小时，输出的警告从 1 个增加到 2 个。
- 总共经过 100 小时后，剩余时间为 0，输出了 3 个警告 (= 2 个警告 + 1 个报警)。

DB1800: 应答

DB1800	应答 [r/w]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB4000	应答 8	应答 7	应答 6	应答 5	应答 4	应答 3	应答 2	应答 1

7.1 PLC 用户程序中的接口

DB1800	应答 [r/w]							
DBB4001	应答 16	应答 15	应答 14	应答 13	应答 12	应答 11	应答 10	应答 9
DBB4002	应答 24	应答 23	应答 22	应答 21	应答 20	应答 19	应答 18	应答 17
DBB4003	应答 32	应答 31	应答 30	应答 29	应答 28	应答 27	应答 26	应答 25

应答 n: 指定给任务 n 的应答位。

如果对应的应答禁止位没有置位, 则应答位的置位会重新启动任务, 特别是会设置任务的实际数据:

剩余时间: = 间隔

待输出的警告的数量 := 0

在 PLC 扫描周期结束后, 该位会自动复位。

示例:

间隔 = 100, 第 1 次输出警告的时间 = 80, 待输出警告的数量 = 2

如果应答禁止位没有置位, 在对应的应答位置位后, 剩余时间便恢复为间隔时间, 待输出警告的数量变为零。

DB1800: 报警

DB1800	维护/报警 [r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB3000	报警 8	报警 7	报警 6	报警 5	报警 4	报警 3	报警 2	报警 1
DBB3001	报警 16	报警 15	报警 14	报警 13	报警 12	报警 11	报警 10	报警 9
DBB3002	报警 24	报警 23	报警 22	报警 21	报警 20	报警 19	报警 18	报警 17
DBB3003	报警 32	报警 31	报警 30	报警 29	报警 28	报警 27	报警 26	报警 25

报警 n: 指定给任务 n 的报警位。

该位针对某个 PLC 循环分别置位为警告 (剩余时间 ≠ 0) 和报警 (剩余时间 = 0)。

DB1800: 取消任务

DB1800	取消任务[r/w]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB2000	取消 8	取消 7	取消 6	取消 5	取消 4	取消 3	取消 2	取消 1
DBB2001	取消 16	取消 15	取消 14	取消 13	取消 12	取消 11	取消 10	取消 9

DB1800	取消任务[r/w]							
DBB2002	取消 24	取消 23	取消 22	取消 21	取消 20	取消 19	取消 18	取消 17
DBB2003	取消 32	取消 31	取消 30	取消 29	取消 28	取消 27	取消 26	取消 25

取消 n: 指定给任务 n 的取消位。

如果该位由 SINUMERIK Operate 或 PLC 用户程序置位，则任务 n 的当前状态被冻结，无法再处理。

TRUE:取消任务

FALSE:激活任务

这样便可以根据模块的实际运行时间修改维护间隔。

DB1800: 应答禁止

DB1800	应答禁止 [r/w]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB5000	应答禁止 8	应答禁止 7	应答禁止 6	应答禁止 5	应答禁止 4	应答禁止 3	应答禁止 2	应答禁止 1
DBB5001	应答禁止 16	应答禁止 15	应答禁止 14	应答禁止 13	应答禁止 12	应答禁止 11	应答禁止 10	应答禁止 9
DBB5002	应答禁止 24	应答禁止 23	应答禁止 22	应答禁止 21	应答禁止 20	应答禁止 19	应答禁止 18	应答禁止 17
DBB5003	应答禁止 32	应答禁止 31	应答禁止 30	应答禁止 29	应答禁止 28	应答禁止 27	应答禁止 26	应答禁止 25

应答禁止 n: 指定给任务 n 的应答禁止位。

如果该位由 SINUMERIK Operate 或 PLC 用户程序置位，即使应答位置位，也不应答任务。

TRUE:禁止应答任务

FALSE: 允许应答任务

这样便可以在 PLC 用户程序中添加一些传感器，这些传感器可发出信号表示维护任务已完成，必要时还允许应答任务。

7.2 操作软件接口

概述

编辑维护任务有以下两种方式：

- 通过编程工具管理 PLC 用户程序中的数据。
- 通过 XML 接口建立维护任务。

区分这两种方式的目的在于，当通过编程工具或调试存档更改 PLC 用户程序中的 PLC 数据块时可以避免 PLC 数据块和 XML 脚本产生冲突。

配置

您最多可以定义 32 个维护任务。在配置模式下，系统会显示以下栏：

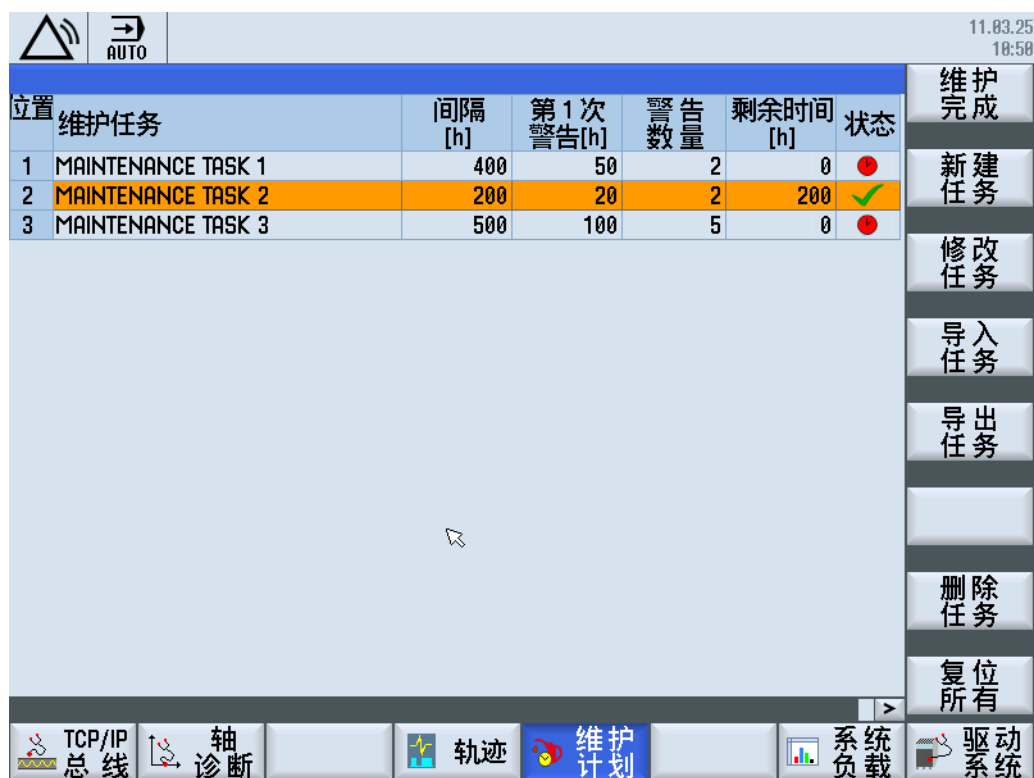
栏目名称	含义
维护任务	维护任务的名称
间隔 [小时]	指离下一次维护相距的最长时间，单位为小时；只要不为 0，PLC 就会将它作为有效的维护任务接收。
第 1 次警告[h]	指出现第一个警告的时间，单位为小时，该值必须小于上面的间隔。
警告数量	指由 PLC 输出、PLC 在间隔期满（剩余时间为 0）后最后一次置位报警位前的报警数量。
剩余时间 [小时]	到间隔期满的时间，单位为小时
状态	<ul style="list-style-type: none"> ● 绿色对号表示离下次维护还有一段时间。 ● 红色时钟表示目前有维护任务需要完成。

窗口中显示的内容取决于不同的存取级别。

配置模式

存取级别 2：服务

在该模式下，您可以创建、修改和删除维护任务。另外还可以应答维护任务。所有栏目都显示在屏幕上。您可以按下<Tab>或者左右箭头键在各栏之间浏览。



The screenshot shows the Maintenance Manager software interface. At the top right, the date and time are 11.03.25 18:58. The main area contains a table with the following data:

位置	维护任务	间隔 [h]	第 1 次 警告[h]	警告 数量	剩余时间 [h]	状态
1	MAINTENANCE TASK 1	400	50	2	0	●
2	MAINTENANCE TASK 2	200	20	2	200	✓
3	MAINTENANCE TASK 3	500	100	5	0	●

On the right side of the table, there is a vertical sidebar with the following buttons: 维护完成, 新建任务, 修改任务, 导入任务, 导出任务, 删除任务, 复位所有. At the bottom of the interface, there is a navigation bar with icons and labels: TCP/IP 总线, 轴 诊断, 轨迹, 维护计划, 系统 负载, 驱动 系统.

说明

编号

在创建任务后，任务会自动编号。如果您不希望自动编号，必须为该任务手动编号。

建议采用手动编号的情况有：因不小心删除了某个任务（编号为 m）而需要创建相同编号的任务。这一点是由 PLC 用户程序内计算方式决定的。

标准模式

存取级别 3：用户

在标准状态中系统会显示出维护任务的最新状态。下面这些栏目处于只读状态，无法编辑：“维护任务”、“间隔”“剩余时间”以及“状态”

The screenshot displays a software interface for maintenance management. At the top right, the date and time are shown as 11.03.24 and 16:51. The main area contains a table with the following data:

位置	维护任务	间隔 [h]	剩余时间 [h]	状态
1	MAINTENANCE TASK 1	400	0	🔴
2	MAINTENANCE TASK 2	200	0	🔴
3	MAINTENANCE TASK 3	500	500	✅

To the right of the table is a vertical sidebar with a header '维护完成' (Maintenance Complete) and several rectangular buttons. At the bottom of the interface is a navigation bar with icons and labels: TCP/IP 总线, 轴 诊断, 轨迹, 维护 计划 (highlighted in blue), and 系统 负载.

7.3 导入和导出维护任务的步骤

概述

对话框中有两个软键，以此可以读入或读取维护任务的配件文件：

- 导入维护任务
- 导出维护任务

设计维护任务

操作步骤：

1. 根据下面描述的结构，建立一个配置文件。
文件名称最好使用小写字母。
2. 最少建立一个文件 `oem_maintenance_<lng>.ts`，维护任务的名称为与语言有关的名称。
3. 把这两个文件拷贝到存储设备上（CF 卡或 USB FlashDrive）。
4. 把存储设备插到前端相应的插口中。
5. 按下软键“导入维护任务”。
6. 按下“确认”。
文件被复制。可以马上读取和语言有关的文本。

当需要进行变更或补充时，点击软键“维护任务导出”。这样，把文件拷贝到储存设备上，就可以在外接编辑器上进行处理了。

xml 配置文件的建立

可以使用下列符号：

标识符	含义
<MAINTENANCE_TASK>	主要标记
<TASK_ID>	维护任务符号
<INTERVALL>	维护间隔时间
<FIRST_WARNING>	到第一次警报，给出的时间间隔。
<NUMBER_OF_WARNING>	警报次数

7.3 导入和导出维护任务的步骤

文件名随意选择，例如 `task.xml`。

```
task.xml
<MAINTENANCE>
  <MAINTENANCE_TASK>
    <TASK_ID>0</TASK_ID>
    <INTERVALL>3</INTERVALL>
    <FIRST_WARNING>1</FIRST_WARNING>
    <NUMBER_OF_WARNING>1</NUMBER_OF_WARNING>
  </MAINTENANCE_TASK>
  <MAINTENANCE_TASK>
    <TASK_ID>1</TASK_ID>
    <INTERVALL>3</INTERVALL>
    <FIRST_WARNING>2</FIRST_WARNING>
    <NUMBER_OF_WARNING>1</NUMBER_OF_WARNING>
  </MAINTENANCE_TASK>
  ...
  <MAINTENANCE_TASK>
    <TASK_ID>2</TASK_ID>
    <INTERVALL>3</INTERVALL>
    <FIRST_WARNING>2</FIRST_WARNING>
    <NUMBER_OF_WARNING>1</NUMBER_OF_WARNING>
  </MAINTENANCE_TASK>
</MAINTENANCE>
```

文件 `oem_maintenance_<lng>.ts` 的结构

该文件后缀名为“.ts”，包含了在窗口中输入的、和语言相关的维护任务文本。在下次运行系统后，可以使用二进制格式 (*.qm) 的文件。

目标目录： `/oem/sinumerik/hmi/lng`

```
oem_maintenance_<lng>.ts
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS>
<TS>
  <context>
    <name>maintenance</name>
    <message>
      <source>1</source>
      <translation>维护任务 1</translation>
      <chars>44</chars>
    </message>
```

```
oem_maintenance_<lng>.ts
<message>
  <source>2</source>
  <translation>维护任务 2</translation>
  <chars>44</chars>
</message>
. . .
<message>
  <source>32</source>
  <translation>维护任务 32</translation>
  <chars>44</chars>
</message>
</context>
</TS>
```

编辑文本条目

在维护窗口中，可以输入维护任务文本、间隔、第一次输出警告的时间和警告数量。另外，如果在配置文件“oem_alarms_config.xml”中有所需条目，还可以在报警文本编辑器中编辑 *.ts 文件。

和现有的语言方案的集成性

在启动时，维护管理器按照语言选择菜单中设置的语言读取文件 oem_maintenance_<lng>.ts。如果没有该文件，则读取调试必备的英语版本。

7.4 应答维护任务

应答维护任务

在结束维护作业后，您可以通过 PLC 用户程序或在操作软件对话框中的“维护完成”软键应答维护任务。

通过对话框应答维护任务后，维护任务对应的应答位置位，PLC 删除实际数据中的“输出的报警/警告数量”，重新载入剩余时间和间隔时间。用户可以从这一点看出维护任务已经成功应答。

在不同时间应答维护任务产生的影响也有所不同：

- **在间隔期满前应答任务**
维护任务可以随时应答。提前应答意味着提前开始新的间隔。
- **在间隔期满后应答任务**
应答维护任务后重新启动任务。

说明

应答所需的存取级别

应答维护任务所需的存取级别由 MD51235 \$MNS_ACCESS_RESET_SERV_PLANNER 确定。

缺省设置：存取级别 2“服务”

Easy Extend

8.1 功能概述

目标

使用“Easy Extend”能够非常简单地调试、激活、取消激活或测试附加设备。控制系统中有一张显示所有可用设备和设备状态的清单。系统可以最多管理 64 个设备。设备的激活或取消由软键操作实现。

功能“Easy Extend”位于“参数”操作区域 → 扩展软键栏 → “Easy Extend” 中。

设计

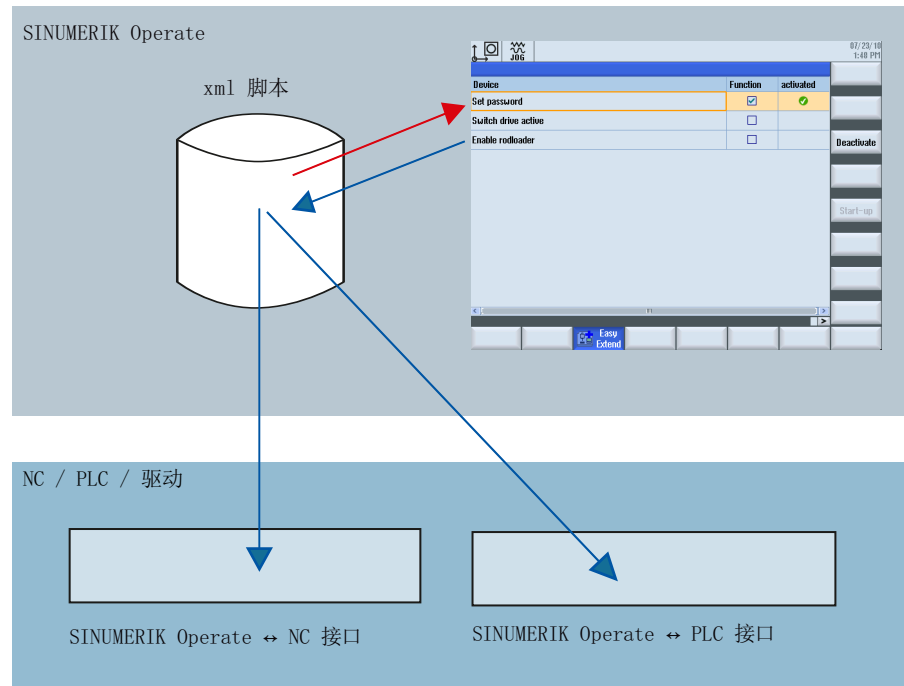


图 8-1 “Easy Extend” 的功能

机床制造商首先定义以下功能，才可以使用“Easy Extend”：

- **PLC 接口 ↔ SINUMERIK Operate**
可选设备是通过操作界面和 PLC 之间的接口来管理的。
- **脚本编辑**
机床制造商应在一个指令脚本中定义如何测安装、激活、取消和测试设备的流程。
- **参数窗口（可选）**
参数窗口用于显示在脚本文件中定义的设备信息。

文件保存

和“Easy Extend”相关的文件保存在系统 CF 卡的目录“oem”（机床制造商）和“oem_i”（个人）中。

文件	名称	目标目录
文本文件	oem_aggregate_XXX.t s	/oem/sinumerik/hmi/lng/ /oem_i/sinumerik/hmi/lng/
脚本文件	agm.xml	/oem/sinumerik/hmi/dvm /oem_i/sinumerik/hmi/dvm
存档文件	任意	/oem/sinumerik/hmi/dvm/archives /oem_i/sinumerik/hmi/dvm/archives
PLC 用户程序	任意	PLC

文档

脚本语言的指令范围与 Easy XML 一致，详细信息参见 SINUMERIK 840D sl/828D Easy XML 编程手册。

8.2 PLC 用户程序中的配置

载入配置

首先将制造商创建的配置和脚本文件、文本文件一起传送到控制系统的制造商目录中。另外，还需要载入对应的 PLC 用户程序。

设备的编程

操作组件和 PLC 之间是通过 PLC 用户程序中**数据块 DB9905** 通讯的，该数据块中有 128 个字预留，最多可用于 64 个设备的管理。

每个设备可以占用四个字节，含义如下：

字节	位	说明	
0	0	== 1	设备已经启动（HMI 反馈信息）
	1	== 1	请激活设备（HMI 请求）
	2	== 1	请取消激活设备（HMI 请求）
	3-7	保留	
1	0-7	保留	
2	0	== 1	设备已激活（PLC 反馈）
	1	== 1	设备出错
	2-7	保留	
3	0-7	设备的唯一标识	

添加轴

在需要为机床增加机床数据时，请按照固定顺序添加驱动对象(DO)，因为调试存档中包含了机床制造商参考机床的结构，一旦顺序被改动，便无法使用。

我们推荐为控制系统组件选择以下设置：

- NC 数据
- PLC 数据
- 驱动数据
 - ACX 格式（二进制）

说明

调试存档

- 为了在 Easy Extend 脚本中使用调试存档，请创建**不带 HMI 数据**的存档！

修改机床配置

- 需要修改驱动机床数据时，首先在控制系统中执行修改。然后在所有设置和装置上再次修改。
-

另见

创建调试存档的步骤 (页 358)

8.3 机床制造商和经销商的选件位

选件位的划分

在 16 位区中管理选件位。按照缺省设置 3 区为经销商预存。这样对于制造商可以使用 48 个选件，经销商有 16 个选件。

Easy Extend 功能采用以下机床数据，对选件进行管理：

MD14510 \$MN_USER_DATA_INT[0] 到 MD14510 \$MN_USER_DATA_INT[3]

可以对地址的分类在脚本中用 **OPTION_MD** 进行重新定义，这样经销商就可以说明自己的地址信息了。对于一个机床型号，如果经销商想提供多于 16 个选件的话，可以检查尚未使用的区。但是，对此要与机床制造商取得联系。

为了使经销商区变得明了，按照脚本中的说明对 3 区进行编程。此外，对于任何设备，都应给出一个固定的设备索引。

示例，对任意一个区进行重新定义：

```
<option_md name= "数据的地址标识符" 索引="<index>"/>
索引 - 分区索引标识符：
0: 设备 0 到 16 (缺省设置)
1: 设备 17 到 32
2: 设备 33 到 48
3: 设备 49 到 64
```

在 PLC 接口上构建设备

为使设备与选件位以及 PLC 接口之间形成一个固定的对应关系，给每个设备都匹配了单一的索引。用属性 **option_bit** 完成固定对应关系。如果匹配不成功，可以通过索引确定位和接口，对设备如下表进行匹配：

索引	MD14510	数据块	设备名称
0	\$MN_USER_DATA_INT[0] 位 0	DB9905.DBB0	设备 1
1	\$MN_USER_DATA_INT[0] 位 1	DB9905.DBB4	设备 2
2	\$MN_USER_DATA_INT[0] 位 2	DB9905.DBB8	设备 3
3	\$MN_USER_DATA_INT[0] 位 3	DB9905.DBB12	设备 4
....
47	\$MN_USER_DATA_INT[2] 位 15	DB9905.DBB188	设备 48

8.3 机床制造商和经销商的选件位

索引	MD14510	数据块	设备名称
48	\$MN_USER_DATA_INT[3] 位 0	DB9905.DBB192	设备 49
49	\$MN_USER_DATA_INT[3] 位 1	DB9905.DBB196	设备 50
....
62	\$MN_USER_DATA_INT[3] 位 14	DB9905.DBB248	设备 63
63	\$MN_USER_DATA_INT[3] 位 15	DB9905.DBB252	设备 64

固定匹配的示例:

```

<device option_bit= "位号">
...
</device>
    
```

示例



按区进行匹配:

- ① DB9905.DBB0 设备 1
- ② DB9905.DBB4 设备 2
- ...
- ③ DB9905.DBB192 设备 49
- ④ DB9905.DBB196 设备 50

在新区上进行匹配:

- ① DB9905.DBB0 设备 1
- ② DB9905.DBB4 设备 2
- ...
- ③ DB9905.DBB72 设备 19

- ④ DB9905.DBB76 设备 20
- ...

PLC 的反馈报告

如果经销商不能置入自己的 PLC 逻辑，就必须通过脚本输入 PLC 的反馈信号。

8.4 操作界面上的显示画面

操作界面上的窗口

以下窗口用于 Easy Extend:

- 控制系统提供了一个**自定义窗口**，其中可显示可用设备。
- 如果还没有开展首次调试，系统会打开**调试窗口**。

如果已经为设备编写了一个调试流程（XML 指令：“START_UP”），而设备还没有经过调试，系统会启动调试流程。

此时，在读取脚本文件中保存的调试存档前，系统会完整备份数据。允许的存档为标准或数据级存档：*.arc 和 *.ard。

- 出错时调试人员可以决定，是否取消调试，或手动排除可能出现在机床配置中的错误。
- 按下“取消”，可以中断调试。系统随后会将取消调试数据的备份。

调试成功结束后需要断开机床时，可以编写 XML 指令“POWER_OFF”，控制系统上便会输出对应的信息。

8.5 建立与语言有关的文本

和语言相关文本的替代符号

系统可以确定有关运行时间的 **CONTROL** 特性（属性值）。需要使用该功能时，应在一个局部变量中提供所需属性，并将带有前置 **\$** 符号的变量名称作为属性值传递给标签。

示例：

```
<let name="caption_device_variable" type="string">my_device 3</let>
<DEVICE option_bit="8">
  <!-- 直接：显示输入的表达式。 -->
  <NAME>my device 1</NAME>
</DEVICE>
<DEVICE option_bit="10">
  <!-- 间接：显示为 caption_id_my_device2 输入的文本。 -->
  <NAME>$$caption_id_my_device2</NAME>
  <form>
    <init>
      <!-- 间接：显示为
form_caption_device2 输入的文本。 -->
      <caption>$$form_caption_device2</caption>
    </init>
  </form>
</DEVICE>
<!-- 间接：显示 caption_device_variable 变量的内容。 -->
<DEVICE option_bit="11">
  <NAME>$$$caption_device_variable</NAME>
</DEVICE>
```

文本文件结构

包含和语言相关的文本的 **xml** 文件应以 **UTF8** 格式创建：

示例：oem_aggregate_eng.ts

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
<context>
<name>EASY_EXTEND</name>
  <message>
```

8.5 建立与语言有关的文本

```
<source>form_caption_id_my_device2</source>
<translation>my_device2</translation>
<chars>64</chars>
</message>
<message>
  <source>form_caption_device2</source>
  <translation>my_input_form</translation>
  <chars>64</chars>
</message>
</context>
</TS>
```

8.6 示例

8.6.1 调试支持参数示例

包含附加参数的对话框

输入区显示选择的驱动参数。

```
<DEVICE>
  <list_id>3</list_id>
  <name> "测试形式" </name>
  <form>

  <init>
    <caption>设备管理器</caption>
    <control name = "编辑 1" xpos = "400" ypos = "34" refvar = "drive/dc/
p105[DO5]" />
    <control name = "编辑 1" xpos = "400" ypos = "54" refvar =
"$MC_AXCONF_MACHAX_USED[4]" />
    <control name = "编辑 1" xpos = "400" ypos = "74" refvar = "drive/dc/
p971[DO5]" />
    <control name = "编辑 1" xpos = "400" ypos = "94" refvar = "drive/dc/
r2[DO5]" />
  </init>

  <paint>
    <text xpos = "40" ypos = "34">dc[DO5]/p105</text>
    <text xpos = "40" ypos = "54">$MC_AXCONF_MACHAX_USED[4]</text>
    <text xpos = "40" ypos = "74">dc[DO5]/p971</text>
    <text xpos = "40" ypos = "94">dc[DO5]/r2</text>
  </paint>
</form>

</DEVICE>
```

包含下拉框的对话框

```
<form>

<init>
```

```
<caption>选择的机床数据</caption>
<DATA_ACCESS type="true" />
<!-- 激活对 NC 变量的直接访问 -->
<control name = "编辑 1" xpos = "322" ypos = "34"
refvar="$MN_AXCONF_MACHAX_NAME_TAB[0]" />
<control name = "编辑 2" xpos = "322" ypos = "54"
refvar="$MN_AXCONF_MACHAX_NAME_TAB[1]" />
<control name = "编辑 3" xpos = "322" ypos = "74"
refvar="$MN_AXCONF_MACHAX_NAME_TAB[2]" />
<control name = "编辑 4" xpos = "322" ypos = "94"
refvar="$MN_AXCONF_MACHAX_NAME_TAB[3]" />

<control name = "编辑 5" xpos = "322" ypos = "114" refvar="$MA_IS_ROT_AX[AX1]"
hotlink="true" />
<control name = "编辑 6" xpos = "322" ypos = "134" refvar="$MA_IS_ROT_AX[AX2]"
hotlink="true" />
<control name = "编辑 7" xpos = "322" ypos = "154" refvar="$MA_IS_ROT_AX[AX3]"
hotlink="true" />

<!-- 使用下拉框控件类型显示旋转轴的值 -->
<control name = "编辑 5" xpos = "322" ypos = "194" refvar="$MA_IS_ROT_AX[AX1]"
fieldtype = "combobox" hotlink="true" >
<item value= "0" >no</item>
<item value= "1" >yes</item>
</control>

<control name = "编辑 6" xpos = "322" ypos = "214" refvar="$MA_IS_ROT_AX[AX2]"
fieldtype = "combobox" hotlink="true" >
<item value= "0" >否</item>
<item value= "1" >是</item>
</control>

<control name = "编辑 7" xpos = "322" ypos = "234" refvar="$MA_IS_ROT_AX[AX3]"
fieldtype = "combobox" hotlink="true" >
<item value= "0" >否</item>
<item value= "1" >是</item>
</control>

</init>

<paint>
```



```

<text xpos = "23" ypos = "34">AXCONF_MACHAX_TAB[0]</text>
<text xpos = "23" ypos = "54">AXCONF_MACHAX_TAB[1]</text>
<text xpos = "23" ypos = "74">AXCONF_MACHAX_TAB[2]</text>
<text xpos = "23" ypos = "94">AXCONF_MACHAX_TAB[3]</text>
<text xpos = "23" ypos = "114">是旋转轴 1</text>
<text xpos = "23" ypos = "134">是旋转轴 2</text>
<text xpos = "23" ypos = "154">是旋转轴 3</text>

<text xpos = "23" ypos = "174">使用下拉框</text>

<text xpos = "23" ypos = "194">是旋转轴 1</text>
<text xpos = "23" ypos = "214">是旋转轴 2</text>
<text xpos = "23" ypos = "234">是旋转轴 3</text>

</paint>

</form>

```

8.6.2 控件示例

下拉框示例:

如果选择了字段类型“combobox”，则需要另外定义显示在窗口中的表达式。为此必须使用标签 `<item>`。下拉框将当前所选文本的索引保存到“CONTROL”的变量中（变量名称）。索引从 1 开始。

```

<control name = "button1" xpos = "10" ypos = "10" fieldtype = "combobox">
<item>文本 1</item>
<item>文本 2</item>
<item>文本 3</item>
<item>文本 4</item>
</control>

```

8.6 示例

赋值示例

如果要为表达式指定任意一个整数值，则要为该标签添加属性 `value=“数值”`。现在控制变量获得 `item` 的赋值，而不是连续编号。

```
<control name = "按钮 1" xpos = "10" ypos = "10" fieldtype = "combobox">  
<item value = "10" >文本 1</item>  
<item value = "20" >文本 2</item>  
<item value = "12" >文本 3</item>  
<item value = "1" >文本 4</item>  
  
</control>
```

说明

属性“hotlink”

属性“hotlink”表示周期性更新对应的控件。这意味着输入的值会被之后的更新周期覆盖。需要避免该特性时，请通过标签“DATA_ACCESS”立即保存输入值。

另一种方法是在表达式中增加 `SOFTKEY_OK` 标签。此标签在关闭对话框时执行。在此段落中可使用 `UPDATE_CONTROLS` 指令执行控制变量和参考变量之间的数据校验。

8.6.3 功率部件示例

说明

两个示例也会随附在 `SINUMERIK 828D Toolbox → Examples` 中。

激活驱动对象

需要激活的驱动对象已经过调试，并由机床制造商重新取消激活，用于将轴作为选件投入市场。

必须执行以下步骤激活轴：

- 通过 `p0105` 激活驱动对象。
- 2. 在通道机床数据中激活轴。

- 通过 p0971 备份驱动机床数据。
- 等待数据写入。
- 重启 NC 和驱动。

```

<DEVICE>
  <list_id>1</list_id>
  <name> "激活驱动" </name>
  <SET_ACTIVE>
    <data name = "drive/dc/p105[DO5]">1</data>
    <data name = "$MC_AXCONF_MACHAX_USED[4]">5</data>
    <data name = "drive/dc/p971[DO5]">1</data>
    <while>
      <condition> "drive/dc/p971[DO5]" !=0 </condition>
    </while>
    <control_reset resetnc = "true" resetdrive = "true"/>
  </SET_ACTIVE>
  <SET_INACTIVE>
    <data name = "drive/dc/p105[DO5]">0</data>
    <data name = "$MC_AXCONF_MACHAX_USED[4]">0</data>
    <data name = "drive/dc/p971[DO5]">1</data>
  </while>
    <condition> "drive/dc/p971[DO5]" !=0 </condition>
  </while>
    <control_reset resetnc = "true" resetdrive = "true"/>
  </SET_INACTIVE>
</DEVICE>

```

激活由 PLC 控制的设备

通过用户程序激活或禁止设备。

```

<AGM>
<OPTION_MD NAME= "$MN_USER_DATA_INT[0]" />
  <DEVICE>
    <LIST_ID>1</LIST_ID>
    <NAME> "PLC Device" </NAME>
    <SET_ACTIVE>
    <SET_ACTIVE>
      <DATA NAME= "PLC/M100.0"> 1 </DATA>
    </SET_ACTIVE>

```

8.6 示例

```
<SET_INACTIVE>  
  <DATA NAME= "PLC/M100.0"> 0 </DATA>  
</SET_INACTIVE>  
</DEVICE>  
</AGM>
```

刀具管理

9.1 基础部分

9.1.1 刀具管理 - 概览

刀具管理 (TM)

刀具管理功能 (TM) 确保在机床上任意时候正确的刀具都位于正确的刀位上。

机床、刀库、装载位和刀具缓冲区 (例如主轴、夹具) 共同构成了对刀具进行保存和输送的设备系统。刀具管理持续向 NC 提供当前刀具位置的信息, 并且记录通过零件程序, PLC 或 SINUMERIK Operate 触发的刀具移动。

在调试刀具管理功能时, 控制系统中会对具体设备系统的结构进行映射。例如会创建一个或多个可在其刀位上接收刀具的刀库。刀具的“停留位置”在控制系统中以值对形式 (刀库号和刀位号) 描述。

说明

供货范围

刀具管理包含在所有控制系统的供货范围中。功能“用于刀具管理的备用刀具”是一个需要许可证的选件。

备用刀具

如果使用选件“用于刀具管理的备用刀具”, 则可以生成并使用备用刀具。

相同刀具名称的刀具组成一个刀具组, 以不同的 Duplo 号来区分。一个刀具组的最大备用刀具数量可以通过 MD17500 \$MN_MAXNUM_REPLACEMENT_TOOLS 来设置。在零件程序中编程刀具组的名称。如果刀具组包含多个刀具, 则通常情况下会使用状态为“有效”的刀具。如果该刀具禁用, 则会使用同一刀具组中的另一把刀具。

文档

刀具参数和刀库参数以及数据结构的详细说明请参见：

- **SINUMERIK 840D sl 功能描述： 刀具管理**
本手册的 NC 相关部分（基本上包含在功能范围中）也适用于 SINUMERIK 828D。本手册中 PLC 功能的说明和 NC 和 PLC 之间的通讯不适用于 SINUMERIK 828D。
- **SINUMERIK 828D 参数手册：“机床数据”**
- **SINUMERIK 828D 参数手册：“NC 变量和接口信号”**
- 刀具管理操作界面的设置请参见：
828D SINUMERIK Operate 调试手册

9.1.2 刀具管理的结构

功能结构

控制系统的软件组件在刀具管理功能中有以下任务：

- **NC:**
刀具管理对刀库刀位进行管理。这些刀位可为空、配备刀具、或者由邻接刀位上的过尺寸刀具占据。空刀位可装载其他刀具。通过刀具管理，机床制造商可实现对刀具和刀库刀位的最优管理。使用刀库管理还可执行装刀、卸刀、刀具定位等扩展功能。此外还能使用刀具、刀库刀位的搜索功能，以及备用刀具的搜索方案。
使用监控功能时，监控生效时刀具被禁用且无法再使用。之后会自动采用未禁用的同种刀具（Duplo 刀具，如果存在的话）继续执行加工。
 - **TO 区域：** 所有刀具管理所必需的数据（刀具数据和刀库数据）都在 NC 的 TO 区域中。
 - **TO 单元 (TOA):** 每个通道都可以设置最多一个自有 TO 单元，作为 NC TO 区域的子集。TO 单元提供了一个独立的刀具管理区域。TO 单元之间没有联系。可以将多个通道分配给一个 TO 单元。但不能将多个 TO 单元分配给一个通道。双通道系统预设为，给每个通道都分配了自有 TO 单元 (MD28085 \$MC_MM_LINK_TOA_UNIT)。

- **PLC:**
 - 执行换刀
 - 在刀库中移动刀具
 - 夹爪控制
 - 必要时进行刀库控制
 - 安全联锁
 - 以传输步骤表提供刀具移动结构
 - 通过应答步骤表对刀具移动进行应答
- **SINUMERIK Operate:**
 - 刀具数据显示, 输入/输出
 - 刀库数据显示, 输入/输出
 - 换位对话框, 装刀/卸刀

PLC 用户程序

PLC 用户程序用于执行刀具管理任务和对刀具（和刀库）的所有位置变化进行应答。监控和避免可能的碰撞是 PLC 用户程序的专有任务，例如：

- 多个主轴使用同一个刀库。
- 同时进行的任务的运行轨迹相交。
- 较大刀具处于换刀机构中时，不可移动链。

PLC 固件

PLC 固件的功能

- 将刀具管理任务传输至 PLC 用户程序。
- 将 PLC 用户程序应答发送至刀具管理
- 将对每条应答的反馈信息（应答故障时与故障编号一起，或者应答正常）传输至 PLC 用户程序。
- 此外：记录任务状态

9.1.3 刀具管理的组件

刀具列表、刀库、刀库列表

可对转塔刀库和链式刀库进行管理。其他刀库类型都映射为上面两种类型。必须将装载位或者装载站用作装刀和卸刀的刀库类型。刀库缓冲区包含了所有其他可放置刀具的刀位（主轴，夹爪）。

刀具库的缺省设置：

因为总是必须要有缓冲区和加载位，所以需管理的真实刀库较少。NC 可管理的刀库数量由系统设定：

- PPU 24x.3/26x.3: 刀库数量 = 3，其中 1 个真实刀库
- PPU 28x.3: 刀库数量 = 4，其中 2 个真实刀库
使用 PPU 28x.3 的特殊情况 (TE822): 刀库数量 = 6，其中每个通道 1 个真实刀库，如果通道 1 使用 TO 单元 1 且通道 2 使用 TO 单元 2（缺省设置）。如果两个通道共同使用 TO 单元 1，则可以设置最多 4 个真实刀库。

刀库

系统提供刀库中所有刀位的内容和状态信息。

刀具的位置通过一个刀库标识符和一个刀位标识符描述。刀库有标识符和编号，刀库刀位仅有编号。在真实刀库（链式、转塔等）中刀具位置通过调试时设定的刀库号和此刀库中的刀位标记。

示例：刀库 1 中刀位 7 的刀具的 T 号：\$TC_MPP6[1.7]

刀具列表

刀具列表包含了所有 NC 已知的刀具。这包括刀库中的刀具，以及已卸载的、需要保留数据的刀具。刀具管理运行时使用刀具列表中装载的刀具。

刀库列表

刀库列表是刀库、夹爪和主轴的针对刀位的映射。刀具管理仅使用刀库列表中的刀具运行。但是在换刀时可选择未分配至刀位的刀具。此时必须手动将刀具添加至机床，并且在加工后再手动将其卸载（手动刀具）。

装载刀库

装载刀库是第 1 内部刀库，其刀库号为 9999。在装载刀库中包含用于刀具装载和卸载的装载位。

在设定刀位时其中一个为固定分配，所有其他刀位可自由设定。固定分配的为装载刀库中的刀位 1。刀位 1 预留，用于装刀/卸刀至所有主轴/刀套。

此外所有至任意刀位（非装载位）的定位任务和换位任务也通过刀位 1 进行。与特定装载刀位相关的任务将在此装载刀位的接口上输出。在调试期间将装载刀位分配至刀库（\$TC_MDP1）。装载位是至刀库的开放接入点，在此可手动向刀库添加或从刀库删除刀具。

缓冲区

缓冲区为第 2 内部刀库。缓冲区包含了主轴、刀套、夹爪、装载器和传输刀位。缓冲区在刀库号 9998 下管理。每个缓冲区单元都会被分配一个唯一的刀位。刀位号可任意设定。建议从编号 1 起递增地为所有主轴或刀套编号。至真实刀库或主轴/刀套间至其他缓冲区的分配在调试时进行（\$TC_MDP2, \$TC_MLSR）。

链式刀库

MD22550 中的设置：对于此刀库类型，\$MC_TOOL_CHANGE_MODE 中的赋值必须为 1。

使用链式刀库时，通常有附加的缓冲区用于刀库和主轴之间的传输。在这些附加缓冲区中可能临时存在刀具。

缓冲区和装载位的名称：

刀库	刀位	含义
1	xx	真实刀库 1（链式、盘式、盒式），位置 xx
9998	1	主轴
9998	2	夹爪
9998	3	夹爪
9998	4	抓刀器
9998	5	换刀机构
9999	1	主轴、手动刀具的装载位
9999	2	刀库装载位

转塔刀库

通常 MD22550 \$MC_TOOL_CHANGE_MODE 设置为 0。

使用转塔刀库时不需要附加的缓冲区用于从刀库到主轴的传输。使用转塔刀库时刀具不会被真正地传输至主轴，而是通过旋转转塔被移动到定义的位置，从而可使用特定的刀具进行加工。至主轴或刀套的传输仅为数据方式执行。将刀具传输至缓冲区 9998/1（主轴）用于通知刀具管理：带有所需刀具的转塔已旋转至加工位置。

编程指令“T = 标识”会触发换刀。也可编程“T = 刀位”。使用“T = 刀位”时，此刀位上可没有刀具。

缓冲区和装载位的名称：

刀库	刀位	含义
1	xx	真实刀库 1（转塔），位置 xx
9998	1	刀套
9999	1	刀套、手动刀具的装载位

对于 MD22550 中的转塔：\$MC_TOOL_CHANGE_MODE 的设置值为 1 时，则与链式刀库说明相同。

邻接刀位观察

邻接刀位观察在使用过尺寸刀具时很有用。在空刀位搜索（装刀，换刀）时，会分析刀库刀位参数 \$TC_MPP4 中的位 4 ... 11（半刀位占用/预留）。

9.1.4 多刀

车削工艺中的多刀：

多刀是一个迷你转塔，在车削工艺中带有数量一定的刀具，如同普通刀具一样可以将其装载进刀库以及从刀库卸载。零件程序中的 T 选择在搜索刀具时也会识别并检查多刀刀具。

简要说明：

- 使用多刀可在一个刀库位置上安装多个刀具。
- 多刀包含了定义数量的多刀刀位，用于安装刀具。
- 在装载至刀库和从刀库卸载时，多刀被视为一个刀具。
- 多刀是作为一个整体进行手动安装的，并且作为整体装载、卸载或者移动。在多刀上安装刀具或者将刀具从多刀上移除都不会在 PLC 接口上产生刀具管理任务。

- 多刀可识别参数“刀位数量”和“刀位间距定义方式”。
- 间距方式可以通过多刀刀位号或者通过多刀刀位角度来定义。
- 多刀名称同刀具名称和刀库名称一样来源于相同的名称范围。
- 不同的多刀必须具有不同的名称，且不定义 Duplo 号。

该功能对下列接口有影响：

- NC 语言（用于刀具选择、换刀、刀具移动的数据定义）
- NC 变量（数据的定义和可视化）
- NC/PLC 接口

编程以及多刀数据的说明参见下列文档。

文档

SINUMERIK 840D sl 功能手册 工具管理

9.1.5 手动刀具

手动刀具

手动刀具是指数据已经存在于 NC 中，但还未加载至刀库的刀具。在 MD22562 \$MC_TOOL_CHANGE_ERROR_MODE 位 1 中设置设备中的手动刀具是否可用。必须手动将自动选择的刀具添加至机床，并且在加工后再手动将其卸载。

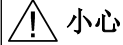
手动刀具包含以下类型的刀具：

- 过尺寸刀具
- 不可在刀库中保存的刀具
- 不允许通过设备的夹爪系统运行的刀具

操作人员的职责

机床操作人员必须确保

- 插在主轴上的刀具的数据组存在于 NC 中。
- 根据 NC 中保存的数据组选择合适的主轴刀具。



小心

机床的安全

机床操作人员必须通过 PLC 用户程序确保操作符合安全规程。

手动刀具的唯一标识

手动刀具会根据 \$TC_TP8[T_Nr] 位 15 = 1 获取唯一标识。设置了该状态后，刀具在刀库中便不会找到空刀位，该（旧）刀具在换刀时直接传输至虚拟刀库位 9999/1。换刀子程序中会以位的方式根据 UND 逻辑关系进行标识识别：

```
IF $TC_TP8[_TOOL_IN_SPINDLE] B_AND 'B1000000000000000' GOTOF HANDTOOL
```

“手动刀具” 状态设置：

如满足以下条件，便设置了刀具的“手动刀具”状态：

- 该机床数据设置为：MD22562 \$MC_TOOL_CHANGE_ERROR_MODE 位 1 = 1
- 在换刀框架中选择一个未加载的换刀位。

“手动刀具” 状态复位：

在以下情况下刀具的“手动刀具”状态会复位：

- 刀具已卸载，即：在当前刀库中不占刀位。可通过以下过程实现：
 - 在刀具更换框架中更换旧刀具。
 - 将手动刀具传输至加载位置（通过 SINUMERIK Operate 卸载）。
 - 选择或切换手动刀具的指令失效，刀具还未运行到刀库刀位。
- 刀具被放至实际刀库或加载刀库的刀位上。

说明

手动刀具报警

在执行有手动刀具参与的换刀时，用户总是会接收到报警（17212、17214 或 17216）。在 PLC 用户程序发出换刀应答后，该报警删除。

报警是可以隐藏的：MD11415 \$MN_SUPPRESS_ALARM_MASK_2 位 16 = 1

通过 SINUMERIK Operate 载入刀具

通过 SINUMERIK Operate 载入至主轴的刀具并不是手动刀具，因此，使用 T0/M06 转入刀库。

9.2 刀具管理的机床数据

机床数据（缺省设置）

以下数据已预设用于刀具管理，或者在“default data”模式下启动时设置：必要时可修改设置。

序号	名称	缺省设置
MD10715[0]	\$MN_M_NO_FCT_CYCLE	6
MD10716[0]	\$MN_M_NO_FCT_CYCLE_NAME	L6
MD10717	\$MN_T_NO_FCT_CYCLE_NAME	TCHANGE
MD17500	\$MN_MAXNUM_REPLACEMENT_TOOLS	1
MD20124	\$MC_TOOL_MANAGEMENT_TOOLHOLDER	1
MD20310	\$MC_TOOL_MANAGEMENT_MASK	181400F / 180400F
MD22550	\$MC_TOOL_CHANGE_MODE	0 / 1
MD22560	\$MC_TOOL_CHANGE_MCODE	206
MD22562	\$MC_TOOL_CHANGE_ERROR_MODE	0（用于手动刀具）

MD20270:\$MC_CUTTING_EDGE_DEFAULT

未编程时刀具刀沿的基本设置（DWORD）：如果在换刀后未编程刀沿，则会使用 \$MC_CUTTING_EDGE_DEFAULT 中设置的刀沿号。

MD20270:\$MC_CUTTING_EDGE_DEFAULT	
> 0	使用 M206 选择的刀沿号。即使之后采用 D 编程，该刀沿选择也生效。
= 1	缺省设置
= 0	在换刀后最先无刀沿生效。在换刀前生效的刀具补偿被取消（对应 D0!）。在 D 编程时刀沿选择才生效。
= -1	旧刀具的刀沿选择也适用于新刀具
= -2	旧刀具的刀沿补偿继续生效，直至编程 D。

MD20270 的设置会对 NC 的程序段处理产生影响。为了避免换刀指令的调用导致预处理停止（该停止一直持续到换刀应答），必须在换刀子程序中执行 NC 功能（无刀具补偿），例如运行轴或输出辅助功能。

示例： MD20270: \$MC_CUTTING_EDGE_DEFAULT= 0 或 = -2

在调用 M206 进行换刀后，可不等待换刀应答继续运行轴，并执行无刀具补偿的运行程序段。运行只在写入了补偿选择（D 号）的程序段中停止，直至 PLC 发送换刀结束的信息。

零件程序：

```

N10 T="钻头 18"           ; 换刀准备
N20 M6                   ; 调用换刀子程序
换刀子程序 L6:
N10 M206                 ; 换刀
N20 D0                   ; 取消补偿
N40 Y150 M79             ; 运行机床轴
N50 G01 D1 X10           ; 激活刀具补偿
                           ; 检查换刀是否完成
                           ; 预处理停止一直保持直至换刀准备完成。 主运行在 N50 (D1) 中等待，直至
                           ; 执行了换刀并对其进行了应答。
    
```

MD20310: \$MC_TOOL_MANAGEMENT_MASK

激活刀具管理功能：

MD20310: \$MC_TOOL_MANAGEMENT_MASK		
位 0	= 1	刀具管理生效： 当前通道的刀具管理功能已使能。
位 1	= 1	刀具管理监控功能生效： 刀具监控（使用寿命和件数）功能已使能。
位 2	= 1	OEM 功能生效
位 3	= 1	相邻刀位观察生效
位 14	= 1	根据以下机床数据的设置选择刀具和补偿： MD20110 \$MC_RESET_MODE_MASK MD20112 \$MC_START_MODE_MASK
位 23	= 1	在选择补偿时与主运行不同步。
位 24	= 1	可通过 201 应答状态进行异步传输，将一把刀具移动到另一刀具预留的刀位上（即预留用于缓冲区中的刀具的刀位）。 为此会在执行移动前删除刀位预留，“预留用于新的待装载刀具”（位值="H8"）保持生效。

MD20310: \$MC_TOOL_MANAGEMENT_MASK		
此外仅对于 T 版本:		
位 16	= 1	T 刀位号生效。

如果禁用的刀具在编程的刀位上，则勾选了选项“刀具管理的备用刀具”时，刀具管理会将姊妹刀具（如果有）的刀位作为任务输出。

MD52270: \$MCS_TM_FUNCTION_MASK

刀具管理功能屏幕

MD52270: \$MCS_TM_FUNCTION_MASK	
位 0:	不允许在刀库刀位上创建刀具。 只能在刀库以外创建刀具。
位 1:	机床不处于复位状态时，禁止装刀/卸刀。 只有在相应的通道处于复位状态中时，才能执行装刀和卸刀。
位 2:	急停时，禁止装刀/卸刀。 只有在急停未生效时，才能执行装刀和卸刀。
位 3:	禁止向主轴装刀或从主轴卸刀。 刀具不能装入主轴或从主轴卸出。
位 4:	刀具直接装入主轴。 刀具只能直接装入主轴。
位 5:	保留
位 6:	保留
位 7:	通过 T 号创建刀具。 在创建刀具时必须输入刀具的 T 号。
位 8:	隐藏“刀具换位”。 “刀具换位”功能在操作界面中隐藏。
位 9:	隐藏“刀库定位”。 “刀库定位”功能在操作界面中隐藏。
位 10:	重新激活刀具，并定位刀库。 在重新激活前将刀具定位至装载位。

MD52270: \$MCS_TM_FUNCTION_MASK	
位 11:	<p>在所有监控类型中重新激活刀具。</p> <p>在重新激活刀具时，会重新激活 NC 中所有使能用于此刀具的监控类型，其中包括未设置用于此刀具而只是处于后台中的监控类型。</p>
位 12:	<p>隐藏“刀具重新激活”。</p> <p>“刀具重新激活”功能在操作界面中隐藏。</p>

车削工艺的缺省设置

对于车削工艺，由下列机床数据定义功能：

- MD22550: \$MC_TOOL_CHANGE_MODE = 0**

转塔刀库的设置：立即用 T 功能进行换刀。不需再使用 M 指令。不会区分“准备换刀”和“执行换刀”。

在此情况下“手动刀具”功能不使能。
- MD20310: \$MC_TOOL_MANAGEMENT_MASK = 81400F (Bit 16=1)**

通过位 16 设置刀具的编程方式：
 T = "x"，其中 x 为刀具名称
 Tx，其中 x 为刀具在刀库中的刀位号。

此功能生效时，使用 T1 选择的是刀位号 1 上的刀具，而不是名称为“1”的刀具。然后确定此刀位上的刀具名称（例如“精修刀”）。之后的步骤都与编程 T="精修刀" 时相同。

使用 T = 刀位号时，此刀位上可没有刀具。
- MD10717: \$MN_T_NO_FCT_CYCLE_NAME = TCHANGE**

用于替换 T 功能的换刀循环的名称

相关描述请参见章节示例：车床的换刀循环 (页 326)

车削工艺中的 ISO 语言：

刀具补偿编程的缺省设置

MD10888 \$MN_EXTERN_DIGITS_TOOL_NO = 0
 MD10889 \$MN_EXTERN_DIGITS_OFFSET_NO = 2
 MD10890 \$MN_EXTERN_TOOLPROG_MODE = 4

T101 与 T0101 的编程结构相同：T1 H01

文档：SINUMERIK 828D/840D sl 功能手册：ISO 语言

- “刀具选项上通过替代循环进行的替代”的说明，适用于 ISO 模式（G291）。

- “刀具更换与刀具补偿”的说明，适用于刀具补偿编程。

配置刀库

在修改刀库配置后须执行 NC 上电。在 NC 重启后，才显示修改过的配置。

另见

带双夹爪的链式刀库的配置 (页 333)

转塔刀库的配置： (页 320) 该程序可以在 SINUMERIK 828D Toolbox 中的“Examples”目录下获取。

9.3 NC ↔ PLC 用户接口

概述

刀具管理从零件程序或 SINUMERIK Operate 接收准备换刀和执行换刀任务（T 指令，M06），刀具移动任务（MVTOOL）或者刀库定位任务（POSM）。刀具管理从这些任务中获取刀具需要进行的刀位变化，并将任务交给 PLC。

用户接口提供单独的数据块，一方面用于卸载、装载、移位和刀库定位，另一方面用于换刀。

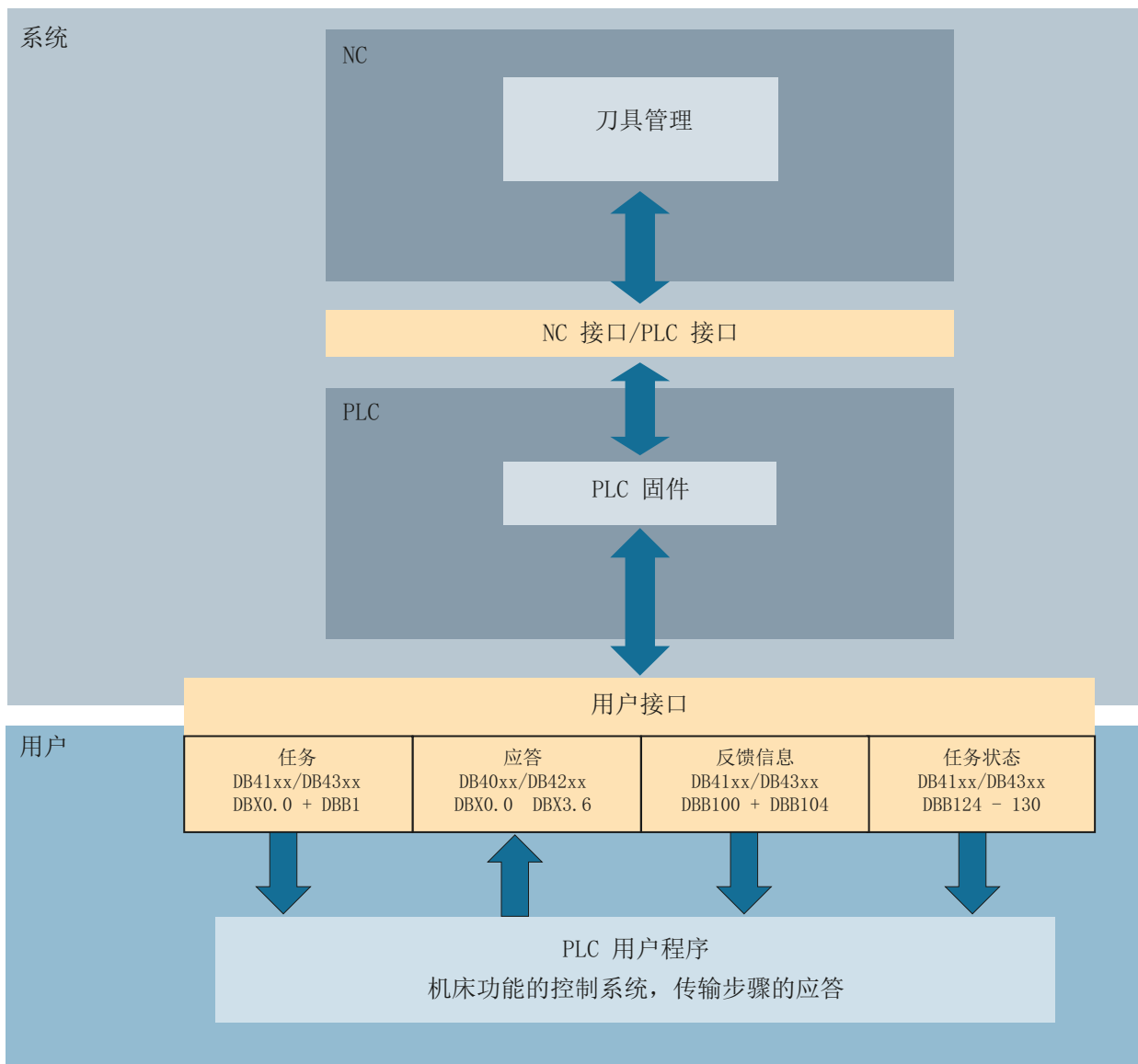


图 9-1 刀具管理的接口

装载/卸载和移位

装载/卸载和移位：通过零件程序或者 SINUMERIK Operate 进行刀具移动 (MVTOOL) 或者刀库定位 (POSM)

xx: 装载位的索引

y: TO 单元的索引

- 40xx y000 用于装载/卸载和移位，刀库定位的 PLC 用户程序的应答
- 41xx y000 用于装载/卸载和移位，刀库定位的刀具管理任务
- 41xx y100 刀具管理的反馈信息
- 41xx y120 任务状态

换刀

换刀 (T 指令, M06)

xx: 主轴的索引

y: TO 单元的索引

- 42xx y000 用于准备和执行换刀的 PLC 用户程序的应答
- 43xx y000 用于准备和执行换刀的刀具管理任务
- 43xx y100 刀具管理的反馈信息
- 43xx y120 任务状态

传输及应答步骤表:

- 9900 0000 固定传输步骤表 (可配置)
- 9901 0000 可变传输步骤表 (可配置并且可更改)
- 9902 0000 应答步骤表 (可配置)

示例:

对于每个 TO 单元在各个数据块中都有自有数据区域可供使用:

- 刀套 1 在 TO 单元 1: DB4x00.0xxx
- 刀套 2 在 TO 单元 1: DB4x01.0xxx
- 刀套 2 在 TO 单元 1: DB4x00.1xxx
- 刀套 2 在 TO 单元 2: DB4x01.1xxx

9.3.1 换刀、卸刀、装刀、刀库定位

刀具移动和刀库移动

每个转载位各有一个接口用于：

- 装刀、卸刀、刀具换位 (MVTOOL) 和刀库定位 (POSM) 任务。

发送至 PLC 用户程序的信息：

- 任务生效
- 任务定义
- 任务描述

任务显示在需要更换刀具的装载位的接口。

- PLC 用户程序的**应答**

对一个任务的所有应答必须在同一个装载位的接口中进行。 应答故障同样必须在此接口中复位。

- 至 PLC 用户程序的刀具管理**反馈信息**

发送至 PLC 用户程序的信息：

- 应答状态
- 故障状态
- 应答位映射

- **任务状态**

保存从最后一个中间应答或结束应答选择的数据。PLC 固件需要这些数据用于下一次对刀具管理的应答，且数据可读用于诊断。PLC 用户程序可使用这些数据进行中断（例如在换刀期间复位）后的重启。

对一个任务的所有应答必须在同一个装载位的接口中进行。

接口信号	含义
DB40xx.DBXy000.0 - DBXy003.6	PLC 用户程序： 对装刀、卸刀、刀具换位或刀库定位的应答
DB40xx.DBXy009.0	PLC 用户程序： 复位信息“应答故障”（DB41xx.DBX100.1）和反馈接口中的诊断信息。
DB41xx.DBXy000.0	刀具管理： 装刀、卸刀、刀具换位或刀库定位任务
DB41xx.DBBY001	刀具管理： 任务定义
DB41xx.DBWY006 - DBWY034	任务描述

接口信号	含义
DB41xx.DBXy100:0	正常反馈：应答状态，应答正常，保持 1 个 PLC 周期
DB41xx.DBXy100.1	非正常反馈：应答状态，应答故障，静态保持
DB41xx.DBBy104	刀具管理：反馈信息故障状态
DB41xx.DBXy108.0 - DB41xx.DBXy111.6	对装刀、卸刀、刀具换位或刀库定位的应答的映射。此映射属于正常/非正常反馈信息并与其一同保持生效。
DB41xx.DBWy124 - DBWy130	任务状态

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 装载位

任务：接口说明

DB4100...41x x	从刀具管理发出的信号 [r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBBy000	多刀：多刀区域的数据							任务仍然有效。
DBBy001		多刀：定位是 NC 特性	多刀：定位	定位是 NC 特性	定位	换位	卸刀	装刀
DBBy002	预留							
DBBy003	预留							
DBBy004	预留							
DBBy005	预留							
DBWy006	源刀库号 (INT)							
DBWy008	源刀位号 (INT)							
DBWy010	目标刀库号 (INT)							
DBWy012	目标刀位号 (INT)							
DBWy014: HMI → PLC								装载/卸载，不运行刀库
DBWy016								
...								

DB4100...41x x	从刀具管理发出的信号 [r]
DBWy056	多刀: 类型 (INT)
DBWy058	多刀: 刀位数量 (INT)
DBWy060	多刀: 刀具间距 (FLOAT)
DBWy062	
DBWy064	多刀: 编号 (INT)
DBWy066	多刀: 刀位号 (INT)
DBWy068	多刀: 主轴编号/刀套编号 (INT)

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 装载位

任务: 信号描述

信号描述:

- 任务:
接口包含一个任务。尚未通过结束应答完成任务执行。在将结束应答传输至刀具管理后此信号复位。
- 多刀: 多刀区域的数据
接口包含任务中给定的多刀的信息。
- 装刀:
需要通过参数“源刀位”的装载位向参数“目标刀位”的刀库刀位执行装刀。
- 卸刀:
需要将参数“源刀位”的刀库刀位中的刀具卸载至参数“目标刀位”的卸载位。
- 换位:
需要将参数“源刀位”的刀库刀位中的刀具换位至参数“目标刀位”的刀库刀位。
- 定位:
需要将参数“源刀位”的刀库刀位定位至参数“目标刀位”的换刀/装载/卸载位。刀具保留在其刀库刀位中。
- 定位是 NC 特性:
定位任务由零件程序发出。
- 多刀: 定位
多刀上的刀具会移动到加工位置。多刀保留在刀套/主轴上。

- 多刀：定位是 NC 特性：
定位任务由零件程序发出。
- 源刀位：
运行的或需要定位至换刀/装载位的刀具的刀库号和刀位号。
- 目标刀位：
刀具需移动到或刀库刀位需定位到的刀库号和刀位号。
- 装刀/卸刀，不运行刀库：
在读入装配数据或功能“卸载全部刀具”时，除了 HMI 的装刀或卸刀任务外还要设置信号。信号可在 PLC 用户程序中用于抑制意外的刀库运行（例如只设置全部应答、无其他操作）。信号可通过有效任务结束应答返回到系统的刀具管理中。
- 多刀：类型：
间距编码方式：1 = 刀位号，2 = 长度，3 = 角度
- 多刀：刀位数量
多刀中已有刀具刀位的数量
- 多刀：刀具间距
间距编码值：多刀中刀具的刀位号或者多刀内部刀具的角度或间距
- 多刀：序号
多刀的 T 编号
- 多刀：刀位号
多刀中连续编号的刀位
- 多刀：主轴编号/刀套编号
从属于多刀的主轴编号/刀套编号

应答：接口说明

DB4000...40x x	发送至刀具管理的信号 [r/w]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBBy000	应答步骤 7	应答步骤 6	应答步骤 5	应答步骤 4	应答步骤 3	应答步骤 2	应答步 骤 1	总应答
DBBy001	应答步骤 15	应答步骤 14	应答步骤 13	应答步骤 12	应答步骤 11	应答步骤 10	应答步 骤 9	应答步骤 8
DBBy002	应答步骤 23	应答步骤 22	应答步骤 21	应答步骤 20	应答步骤 19	应答步骤 18	应答步 骤 17	应答步骤 16

DB4000...40x x	发送至刀具管理的信号 [r/w]							
DBBy003	预留	应答步骤 30	应答步骤 29	应答步骤 28	应答步骤 27	应答步骤 26	应答步 骤 25	应答步骤 24
DBWy004	多刀： 刀位号（INT）							
DBBy006	预留							
DBBy007	预留							
DBBy008	预留							
DBBy009								复位应答 故障
DBBy010	预留							

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 装载位

应答：信号说明

信号说明：

- 总应答：

在 0/1 脉冲沿时会向当前任务发送状态为 99 的结束应答（任务完成，已到达所有目标位置）。此信号存在时，不可修改此接口的数据！

在将应答传输至刀具管理后，此信号由 PLC 固件复位。
- 应答步骤 1 到 30：

在 0/1 脉冲沿时会将应答步骤表中的相应应答步骤发送至刀具管理。此信号存在时，不可修改此接口的数据和可变传输步骤表的数据！

在将应答传输至刀具管理后，此信号由 PLC 固件复位。
- 多刀： 刀位号

PLC 用户程序将有效刀位号的信息反馈给 NC 的刀具管理。
- 复位应答故障：

复位信息“应答故障”（DB41xx.DBX100.1）和反馈接口中的诊断信息。

反馈信息：接口说明

DB4100...41x x	从刀具管理发出的信号 [r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBBy100							应答故障	应答正常
DBBy101	预留							
DBWy102	预留							
DBWy104	故障状态 (WORD)							
DBBy106	预留							
DBBy107	预留							
DBBy108	应答步骤 7	应答步骤 6	应答步骤 5	应答步骤 4	应答步骤 3	应答步骤 2	应答步骤 1	应答步骤 0
DBBy109	应答步骤 15	应答步骤 14	应答步骤 13	应答步骤 12	应答步骤 11	应答步骤 10	应答步骤 9	应答步骤 8
DBBy110	应答步骤 22	应答步骤 22	应答步骤 21	应答步骤 20	应答步骤 19	应答步骤 18	应答步骤 17	应答步骤 16
DBBy111	预留	应答步骤 30	应答步骤 29	应答步骤 28	应答步骤 27	应答步骤 26	应答步骤 25	应答步骤 24
DBWy112	多刀：刀位号 (INT)							

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 装载位

反馈信息： 信号描述

信号描述：

- **应答正常 (DB41xx.DBXy100.0):**

PLC 用户程序的应答（区域 DB40xx.DBB0 至 DBB3）已无故障传输至刀具管理。此信号在一个 PLC 周期后复位。

- **应答故障 (DB41xx.DBXy100.1):**

非正常反馈应答状态。PLC 用户程序的应答（区域 DB40xx.DBB0 至 DBB3）出错。故障原因在“故障状态”中给出。

如果 PLC 固件认为应答正常而接收应答，但刀具管理在应答的刀具传输中识别出故障并发送信息时（例如刀具传输的目标刀位已被占用），位“应答故障”也会置位。

在 PLC 程序尚未应答时，刀具管理在 NC 中直接检测到的故障将不会引起位 100.1 的置位。

如果故障会阻碍应答向刀具管理的传输（故障状态 1 至 7），则其仅在刀具管理的接口上输出，而不通过 NC 输出（无 NC 报警!）。

必要时在 PLC 用户程序中通过用户 PLC 报警发送此故障信息。

此信号“静态保持”直至用户应答故障（设置位“复位应答故障”

DB40xx.DBX9.0）。位“应答故障”待处理时，DB40xx.DBB0 至 DBB3 中的接口被禁用。输入应答位不通过 PLC 固件分析，且在位“复位应答故障”置位时被删除。

- **多刀： 刀位号**

多刀： PLC 用户程序将其作为有效刀位号在应答时输入应答接口的刀位号。

故障状态

故障时在 DB41xx.DBBY104 中显示一个不等于零的诊断编号，含义如下：

诊断编号	含义
0	无故障
1	同时存在多个应答信号
2	无任务应答
3	无效的传输步骤编号
4	定位设定无任务
5	状态不允许更改刀位（已使用应答状态 0）
7	使用了不允许的应答状态
其他值：	该值相当于此次传输造成的、NC 中刀具管理的故障信息。

故障状态在用户应答故障后复位为零。

应答的映射 (DB41xx.DBBy108 至 DBBy111)

PLC 用户程序最后设置的应答 (DB40xx.DBB0 至 DBB3) 在此由 PLC 固件与位“应答正常”或“应答故障”一起置位和复位。根据在故障状况下静态保持的位, 用户可识别出哪个应答步骤触发了故障。如果通过 PLC 用户程序错误地设置了多个应答位, 则这些位也输入映射。

任务状态

DB4100...41xx	从刀具管理发出的信号 [r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBWy120	预留							
DBWy122	预留							
DBWy124	刀具的当前刀库号 (INT)							
DBWy126	刀具的当前刀位号 (INT)							
DBWy128	刀具的目标刀库号 (INT)							
DBWy130	刀具的目标刀位号 (INT)							
DBWy132	预留							
DBWy134	预留							
DBWy136	预留							
DBWy138	预留							
DBWy140	多刀类型 (INT)							
DBWy142	多刀刀位数量 (INT)							
DBWy144	多刀刀具间距 (FLOAT)							
DBWy146								
DBWy148	多刀刀位号 (INT)							

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 装载位

9.3.2 换刀

接口描述

每个刀套/主轴各有一个接口用于：

- “准备换刀”和“执行换刀”任务。
 发送至 PLC 用户程序的信息：任务生效；任务说明和任务描述。
 任务显示在需要更换刀具的刀套（主轴）的接口。
- PLC 用户程序的应答
 对一个任务的所有应答必须在同一个刀套（主轴）的接口中进行。应答故障同样必须在此接口中复位。
- 至 PLC 用户程序的刀具管理反馈信息
 发送至 PLC 用户程序的信息：应答状态，故障状态，应答位映射。
- 任务状态
 保存从最后一个中间应答或结束应答选择的数据。PLC 固件需要这些数据用于下一次对刀具管理的应答，且数据可读用于诊断。PLC 用户程序可使用这些数据进行中断（例如在换刀期间复位）后的重启。

换刀的结束应答

对“准备换刀”和“执行换刀”可使用共同的（在使用转塔刀库时通过 T 指令换刀）或独立的（不同程序段中的 Txx 和 M206，铣削缺省设置）结束应答。在进行了相应的机床数据设置时，对“准备换刀”任务的结束应答允许 NC 预处理继续运行。

用于定义程序段预处理、主处理特性以及各种响应特性的机床数据在章节刀具管理的机床数据 (页 271)中描述。

通过对“执行换刀”任务的结束应答可继续进行 NC 的主处理。因此应尽早进行结束应答。这表示在旧刀具进入刀库之前就可进行结束应答（例如新刀具在主轴上，旧刀具在抓刀器中）。此时将旧刀具送到刀库的剩余步骤的信息必须异步发送。为此必须使用与同步应答相同的接口。

接口信号	含义
DB42xx.DBXy000.0 - DBXy003.6	PLC 用户程序： “换刀准备”和“换刀执行”应答
DB42xx.DBXy009.0	PLC 用户程序： 复位信息“应答故障”（DB43xx.DBX100.1）和反馈接口中的诊断信息。

接口信号	含义
DB43xx.DBXy000.0	刀具管理：“准备换刀”和“执行换刀”任务
DB43xx.DBBy001	刀具管理任务说明
DB43xx.DBWy006 - DBWy034	任务说明
DB43xx.DBXy100.0 DB43xx.DBXy100.1	正常反馈：应答状态，应答正常，保持 1 个 PLC 周期 非正常反馈：应答状态，应答故障，静态保持
DB43xx.DBXy100.0 DB43xx.DBXy100.1	正常反馈：应答状态，保持 1 个 PLC 循环 非正常反馈：应答状态，静态保持
DB43xx.DBXy100.0	刀具管理反馈信息：应答状态
DB43xx.DBBy104	刀具管理反馈信息：故障状态
DB43xx.DBXy108.0 - DB43xx.DBXy111.6	对换刀应答的映射：此映射属于正常/非正常反馈信息并与其一同保持生效。
DB43xx.DBWy124 - DBWy138	任务状态

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 主轴/刀套索引

任务：接口说明

DB4300...43xx	从刀具管理发出的信号 [r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBBy000	多刀：多刀区域的数据	--	--	--	--	--	--	任务仍然有效。
DBBy001	刀具停留在主轴	卸载手动刀具	装载手动刀具	无旧刀具	T0	准备换刀	执行换刀（触发：M06）	固定位置编码
DBBy002	预留							
DBBy003	预留							
DBBy004	预留							
DBBy005	预留							
DBWy006	新刀具的源刀库号（INT）							
DBWy008	新刀具的源位置号（INT）							

DB4300...43xx	从刀具管理发出的信号 [r]								
DBWy010	预留								
DBWy012	预留								
DBWy014	预留								
DBWy016	预留								
DBWy018	旧刀具的目标刀库号 (INT)								
DBWy020	旧刀具的目标位置号 (INT)								
DBWy022	刀位类型 (INT)								
DBWy024	左侧尺寸 (INT)								
DBWy026	右侧尺寸 (INT)								
DBWy028	预留								
DBWy030	预留								
DBBy032	新刀具的状态:								
	刀具已使用	刀具固定位置编码	刀具处于更换中	达到预警极限	刀具测量	刀具已禁用	刀具已使用	激活的刀具	
DBBy033	新刀具的状态:								
	手动刀具	1:1 换刀	--	原刀具	待装载	待卸载	禁用	周转区刀具的标识	
DBWy034	新刀具: NC 的内部 T 号 (INT)								
DBWy036	预留								
DBWy038	预留								
DBWy040	预留								
DBWy042	预留								
DBWy044	自由参数 1 (DWORD)								
DBWy046									
DBWy048	自由参数 2 (DWORD)								
DBWy050									
DBWy052	自由参数 3 (DWORD)								
DBWy054									
DBWy056	多刀: 类型 (INT)								
DBWy058	多刀: 刀位数量 (INT)								
DBWy060	多刀: 刀具间距 (FLOAT)								
DBWy062									

DB4300...43xx	从刀具管理发出的信号 [r]
DBWy064	多刀：新刀具的编号（源）(INT)
DBWy066	多刀：新刀具的刀位号（源）(INT)
DBWy068	多刀：旧刀具的编号（目标）(INT)
DBWy070	多刀：旧刀具的刀位号（目标）(INT)
DBWy072	刀位类型 (INT)
DBWy074	左侧尺寸 (INT)
DBWy076	右侧尺寸 (INT)
DBWy078	上部尺寸 (INT)
DBWy080	下部尺寸 (INT)
DBWy082	新刀具的刀具状态 (INT)
DBWy084	刀具 T 编号 (INT)
DBWy086	主轴编号/刀套编号 (INT)
DBWy088	多刀：原始刀库号 (INT)
DBWy090	多刀：原始刀位号 (INT)

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 刀套

任务：信号说明

信号说明：

- 任务：
接口包含一个任务。尚未通过结束应答完成任务执行。在将结束应答传输至刀具管理后此信号复位。
- 多刀：多刀区域的数据
接口包含任务中给定的多刀的信息。
- 固定刀位编码：新刀具是固定刀位编码的。
- 执行换刀：
需要将新刀具装载至刀套/主轴。而将旧刀具送回刀库位置。此任务总是需要结束应答。
- 准备换刀：
准备新刀具。必要时将旧刀具的刀库位置定位至换位置。此任务需要独立的结束应答；在同时进行“执行换刀”任务时可省略对“准备换刀”的应答。

- T0: T0 已编程（空运行刀套/主轴）。
- 无旧刀具：
向尚为空的刀套/主轴执行换刀。
- 装载手动刀具：
需装载手动刀具。在 SINUMERIK Operate 上显示需要换入的刀具。
- 卸载手动刀具：
需手动卸载刀具。
- 刀具停留在主轴：
在从“刀套 → 主轴”切换为“刀套 → 主轴”时此位置位。例如复位启动模式或程序段搜索都可能触发该动作。
- 新刀具的源位置：
新刀具原来的刀库号和位置号（通常为真实刀库中的位置）。
- 旧刀具的目标位置：
旧刀具需要移动至的刀库号和位置号（通常为实际刀库中的位置）。
- 新刀具的来源：
 - 内部 T 号：新刀具的内部 T 号
 - 刀具状态：新刀具的状态
 - 刀位类型：新刀具的刀位类型
 - 大小：新刀具的尺寸（右侧、左侧）
- 自由参数：3 个从零件程序传输至 PLC 用户程序的自由参数。
- 多刀：类型：
间距编码方式：1 = 刀位号，2 = 长度，3 = 角度
- 多刀：刀位数量
多刀中已有刀具刀位的数量
- 多刀：刀具间距
间距编码值：多刀中刀具的刀位号或者多刀内部刀具的角度或间距
- 多刀：编号（源）/多刀：新刀具的刀位号（源）
新多刀原来的刀库号和刀位号（通常为真实刀库中的刀位）。
- 多刀：编号（目标）/多刀：旧刀具的刀位号（目标）
旧多刀需要移动至的刀库号和刀位号（通常为实际刀库中的刀位）。
- 刀位类型：新刀具的刀位类型
- 左侧尺寸：刀库中多刀所需的左侧刀位尺寸
- 右侧尺寸：刀库中多刀所需的右侧刀位尺寸

- 上部尺寸：刀库中多刀所需的上部刀位尺寸
- 下部尺寸：刀库中多刀所需的上部刀位尺寸
- 刀具状态
- 刀具的 T 编号
- 主轴编号/刀套编号
- 多刀：原始刀库号
- 多刀：原始刀位号

应答：接口说明

DB4200 ... 42xx	发送至刀具管理的信号 [r/w]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBBy000	应答步骤 7	应答步骤 6	应答步骤 5	应答步骤 4	应答步骤 3	应答步骤 2	应答步骤 1	总应答
DBBy001	应答步骤 15	应答步骤 14	应答步骤 13	应答步骤 12	应答步骤 11	应答步骤 10	应答步骤 9	应答步骤 8
DBBy002	应答步骤 23	应答步骤 22	应答步骤 21	应答步骤 20	应答步骤 19	应答步骤 18	应答步骤 17	应答步骤 16
DBBy003	预留	应答步骤 30	应答步骤 29	应答步骤 28	应答步骤 27	应答步骤 26	应答步骤 25	应答步骤 24
DBWy004	多刀：刀位号 (INT)							
DBBy006	预留							
DBBy007	预留							
DBBy008	预留							
DBBy009							全部应答时的状态 3	复位应答故障
DBBy10	预留							

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 刀套

应答：信号说明

信号说明：

- 总应答：
在 0/1 脉冲沿时会向当前任务发送状态为 99 的结束应答（任务完成，已到达所有目标位置）。此信号存在时，不可修改此接口的数据！
在将应答传输至刀具管理后，此信号由 PLC 固件复位。
- 应答步骤 1 到 30：
在 0/1 脉冲沿时会将应答步骤表中的相应应答步骤发送至刀具管理。此信号存在时，不可修改此接口的数据和可变传输步骤表的数据！
在将应答传输至刀具管理后，此信号由 PLC 固件复位。
- 多刀：刀位号
PLC 用户程序将有效刀位号的信息反馈给 NC 的刀具管理。
- 复位应答故障：
复位信息“应答故障”（DB43xx.DBX100.1）和反馈接口中的诊断信息。
- 全部应答时的状态 3
用于中断换刀任务的位：
 - 使用等同于配置的应答步骤状态 = 3。
 - 仅由 PLC 用户程序中的机床制造商负责使用。

反馈信息：接口说明

DB4300...43xx	从刀具管理发出的信号 [r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBBy100							应答故障	应答正常
DBBy101	预留							
DBWy102	预留							
DBWy104	故障状态 (WORD)							
DBWy106	预留							
DBBy108	应答步骤 7	应答步骤 6	应答步骤 5	应答步骤 4	应答步骤 3	应答步骤 2	应答步骤 1	应答步骤 0
DBBy109	应答步骤 15	应答步骤 14	应答步骤 13	应答步骤 12	应答步骤 11	应答步骤 10	应答步骤 9	应答步骤 8
DBBy110	应答步骤 23	应答步骤 22	应答步骤 21	应答步骤 20	应答步骤 19	应答步骤 18	应答步骤 17	应答步骤 16

DB4300...43xx	从刀具管理发出的信号 [r]							
DBBy111	预留	应答步骤 30	应答步骤 29	应答步骤 28	应答步骤 27	应答步骤 26	应答步骤 25	应答步骤 24
DBBy112	多刀: 刀位号 (INT)							

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 刀套

反馈信息: 信号描述

信号描述:

- 应答正常 (DB43xx.DBX100.0): 反馈” 应答状态正常”。**
 PLC 用户程序的应答 (区域 DB42xx.DBB0 至 DBB3) 已顺利传输至刀具管理。此信号在一个 PLC 周期后复位。
- 应答故障 ((DB43xx.DBX100.1): 非正常反馈应答状态。**
 PLC 用户程序的应答 (区域 DB42xx.DBB0 至 DBB3) 出错。故障原因在“故障状态”中给出。
 如果 PLC 固件认为应答正常而接收应答, 但刀具管理在应答的刀具传输中识别出故障并发送信息时 (例如刀具传输的目标位置已被占用), 位“应答故障”也会置位。
 在 PLC 程序尚未应答时, 刀具管理在 NC 中直接检测到的故障将不会引起位 100.1 的置位。
 如果故障会阻碍应答向刀具管理的传输 (故障状态 1 至 7), 则其仅在刀具管理的接口上输出, 而不通过 NC 输出 (无 NC 报警!)。
 必要时在 PLC 用户程序中通过用户 PLC 报警发送此故障信息。
 此信号“静态保持”直至用户应答故障 (设置位“复位应答故障” DB4200.DBX9.0)。位“应答故障”待处理时, DB42xx.DBB0 至 DBB3 中的接口被禁用。输入应答位不通过 PLC 固件分析, 且在位“复位应答故障”置位时被删除。
- 多刀: 刀位号**
 多刀: PLC 用户程序将其作为有效刀位号在应答时输入应答接口的刀位号。

故障状态

故障时在 DB43xx.DBB104 中显示一个不等于零的诊断编号, 含义如下:

诊断编号	含义
0	无故障
1	同时存在多个应答信号

诊断编号	含义
2	无任务应答
3	无效的传输步骤编号
4	定位设定无任务
5	状态不允许更改刀位（已使用应答状态 0）
7	使用了不允许的应答状态
其他值:	该值相当于此次传输造成的、NC 中刀具管理的故障信息。

故障状态在用户应答故障后复位。

应答的映射（DB43xx.DBB108 至 DBB111）

PLC 用户程序最后设置的应答（DB42xx.DBB0 至 DBB3）在此由 PLC 固件与位“应答正常”或“应答故障”一起置位和复位。根据在故障状况下静态保持的位，用户可识别出哪个应答步骤触发了故障。如果通过 PLC 用户程序错误地设置了多个应答位，则这些位也输入映射。

任务状态

DB4300 ... 43xx	从刀具管理发出的信号 [r]							
	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBWy120	预留							
DBWy122	预留							
DBWy124	新刀具的实际刀库号 (INT)							
DBWy126	新刀具的实际位置号 (INT)							
DBWy128	新刀具的目标刀库号 (INT)							
DBWy130	新刀具的目标位置号 (INT)							
DBWy132	新刀具的实际刀库号 (INT)							
DBWy134	新刀具的实际位置号 (INT)							
DBWy136	旧刀具的目标刀库号 (INT)							
DBWy138	旧刀具的目标位置号 (INT)							
DBWy140	多刀: 类型 (INT)							
DBWy142	多刀: 刀位数量 (INT)							

DB4300 ... 43xx	从刀具管理发出的信号 [r]
DBWy144	多刀：刀具间距 (FLOAT)
DBWy146	
DBWy148	多刀：刀位号 (INT)

y [= 0 或 1]: TO 单元 1 或 TO 单元 2

xx: 刀套

9.3.3 传输步骤表和应答步骤表

可配置步骤表

在数据块 TM_CTS (DB9900)、TM_VTS (DB9901) 和 TM_ACK (DB9902) 中提供用于描述刀具移动过程的可配置表格。

接口信号	名称	含义
DB9900	TM_CTS	固定传输步骤表 (可配置)
DB9901	TM_VTS	可变传输步骤表 (可配置且可通过 PLC 用户程序写入)
DB9902	TM_ACK	应答步骤表 (可配置)

数据块 DB40xx、41xx、42xx 和 43xx 为系统模块，且由控制系统自动创建。

可在编程工具中通过库/特殊数据块使用数据块 DB9900、DB9901 和 DB9902。数据块尚未填充必要的数，必须由用户复制到 PLC 项目并进行编辑。

传输步骤表

单独的刀具移动被定义为刀具从刀库刀位 x/y 到刀库刀位 m/n 的传输步骤。可使用这些传输步骤定义应答步骤。DB9900 包含了固定配置的传输步骤 (固定传输步骤表)。DB9901 可通过 PLC 用户程序修改，例如用于应答中间步骤，如换刀准备时的刀库位置 (可变传输步骤表)。

DB9900	固定传输步骤表 [r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBW0	传输步骤 1 源刀库号 (INT)							
DBW2	传输步骤 1 源刀位号 (INT)							

DB9900	固定传输步骤表 [r]
DBW4	传输步骤 1 目标刀库号 (INT)
DBW6	传输步骤 1 目标刀位号 (INT)
DBW8	传输步骤 2 源刀库号 (INT)
DBW10	传输步骤 2 源刀位号 (INT)
DBW12	传输步骤 2 目标刀库号 (INT)
DBW14	传输步骤 2 目标刀位号 (INT)
	...
DBW504	传输步骤 64 源刀库号 (INT)
DBW506	传输步骤 64 源刀位号 (INT)
DBW508	传输步骤 64 目标刀库号 (INT)
DBW510	传输步骤 64 目标刀位号 (INT)

DB9901	可变传输步骤表 [rw]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBW0	传输步骤 101 源刀库号 (INT)							
DBW2	传输步骤 101 源刀位号 (INT)							
DBW4	传输步骤 101 目标刀库号 (INT)							
DBW6	传输步骤 101 目标刀位号 (INT)							
DBB 8	传输步骤 102 源刀库号 (INT)							
DBW10	传输步骤 102 源刀位号 (INT)							
DBW12	传输步骤 102 目标刀库号 (INT)							
DBW14	传输步骤 102 目标刀位号 (INT)							
	...							
DBW504	传输步骤 164 源刀库号 (INT)							
DBW506	传输步骤 164 源刀位号 (INT)							
DBW508	传输步骤 164 目标刀库号 (INT)							
DBW510	传输步骤 164 目标刀位号 (INT)							

应答步骤表

每一条记录指示两个传输步骤（新刀具和旧刀具），并给出已达到的相应状态。DB9902 中的应答步骤表既用于装载位的接口上的应答，也用于刀套接口上的应答。

DB9902	应答步骤表 [r]							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB0	应答步骤 1 新刀具的传输步骤 (BYTE)							
DBB1	应答步骤 1 旧刀具的传输步骤 (BYTE)							
DBB2	应答步骤 1 应答状态 (BYTE)							
DBB3	应答步骤 1 预留							
DBB4	应答步骤 2 新刀具的传输步骤 (BYTE)							
DBB5	应答步骤 2 旧刀具的传输步骤 (BYTE)							
DBB6	应答步骤 2 应答状态 (BYTE)							
DBB7	应答步骤 2 预留							
	...							
DBB116	应答步骤 30 新刀具的传输步骤 (BYTE)							
DBB117	应答步骤 30 旧刀具的传输步骤 (BYTE)							
DBB118	应答步骤 30 应答状态 (BYTE)							
DBB119	应答步骤 30 预留							

参见

PLC 程序块 (页 298)

9.4 PLC 程序块

9.4.1 应答过程

发送至刀库管理的信息

刀具管理等待对其任务的应答，用于对实际刀具位置进行实时跟踪。每个任务至少需要一个应答。在很多应用中满足此要求即可。

应答可按照 DB9902 中定义的表格进行；或者在一个步骤中的刀具管理任务完成后，通过用户接口上相应位的 0/1 脉冲沿发出总应答（DB40xx/42xx DBX0.0）。

应答信号存在时，不可修改此接口的数据！在将应答传输至刀具管理后，此信号由 PLC 固件复位。复位可能在多个 PLC 周期后进行。

如果也向刀具管理提供中间位置的信息，则有以下优势：

- **中间位置的信息：**

如果刀具管理已知所有中间位置，也可询问缓冲区刀库的分配情况。这样，在系统重新上电或被指令（如：复位）中断运行后，可以更便捷地启动。如果需要使用刚刚卸载的刀具，可不将其保存至刀库，而直接将其从缓冲区刀位重新装载至主轴。

- **刀库位置的信息：**

如果刀具管理已知哪个刀库刀位处于传输位（至主轴的换刀位，装载位）上，则其可确定刀库中的最短距离用于搜索空刀位或选择新刀具。在任务中，刀具管理通常可从中间应答（例如真实刀库和缓冲区之间的刀具传输）或结束应答（例如“刀库定位”任务完成）识别刀库位置。如果未通过刀具管理任务，而是由 PLC 用户程序自行进行了刀库定位（例如通过 HMI 或机床键），必须通过异步信息将定位告知刀具管理。

9.4.2 应答类型

刀具移动和刀库移动

刀具管理分为同步应答和独立于任务的异步信息。

同步应答

- 任务的中间步骤的应答（刀具管理记录当前的刀具位置更改，零件程序须等待）
中间应答可以向刀具管理提供任务的中间步骤的信息。中间应答只提供中间步骤的目标位置。源位置在任务或最后的中间应答中提供。在换刀时也可同时对两把刀具（新和旧）进行应答。中间应答只能在结束应答之前进行。
- 对任务的结束应答（零件程序可继续执行）
每个任务都需要一个结束应答。结束应答允许零件程序继续执行，并使能新任务的任务接口。结束应答应尽早进行（例如在新刀具刚刚位于主轴中且不希望再发生碰撞时）。在对任务进行结束应答后，可将后续步骤异步发送给刀具管理（例如旧刀具返回刀库的距离）。

独立于任务的异步信息

提示刀具或刀位的位置改变的“异步信息”，例如 PLC 通过机床控制面板更改了刀具位置，而不是通过换刀任务。

使用异步信息可独立于任务向刀具管理提供刀具移动或刀库移动的信息。一条异步信息必须总是包含源位置（从）和目标位置（向）。

刀库中的刀具移动（刀具换位）只能通过实际进行了分配的刀库刀位执行。不允许进行空传输。在一条信息中可执行两个异步传输。在此情况下必须使用换刀接口 DB42xx。

应答产生的影响

应答对任务和零件程序产生的影响：

- 中间应答和结束应答与任务同步进行。
 - 零件程序必须等待。
 - 不会有新任务。
- 异步传输的信息：
 - 零件程序继续运行。
 - 信息完全独立于任何任务。

9.4.3 应答状态

应答状态及其含义:

下表中给出了相应应答类型所指示的状态:

应答		含义
同步结束应答	1	任务在设定的位置结束: 刀具处于设定位置。零件程序可继续运行。
	3	任务取消: 任务取消,保留之前应答的刀具位置更改。取消指令自身不会触发刀具管理中的位置应答或更改。
	7	重复“准备换刀”任务: 换刀之前已发送了一个新的刀具位置信息。需考虑此位置重新计算“准备换刀”任务。仅在尚未应答准备指令时允许。
	99	总应答: 任务结束,到达所有位置。 所有涉及的刀具都处于任务中设定的位置上。零件程序可继续运行。达到了任务中的所有目标位置。
同步中间应答	105	刀具的中间位置: 刀具已从任务中设定的源位置或者上一次应答的中间位置向设定的目标位置移动。
发送异步传输信息	201	发送刀具移动信息: 刀具已从源位置向设定的目标位置移动。在从真实刀库刀位移动至缓冲区刀位时,为该刀具保留了原先的刀位。 对此,还要参考 MD20310,位 24 (页 271)。
	204	发送刀库位置信息: 刀库刀位处于设定目标刀位所属的换刀/装载/卸载位上。

使用的表格参数一览

应答状态		1	3	7	99	105	201	204
新刀具	传输步骤	x	-	-	-	x	x	x
	● 从刀库	-	-	-	-	-	xx	xx
	● 从刀位	-	-	-	-	-	xx	xx
	● 向刀库	xx	-	-	-	xx	xx	zz
	● 向刀位	xx	-	-	-	xx	xx	zz
旧刀具	传输步骤	x	-	-	-	x	x	-
	● 从刀库	-	-	-	-	-	xx	-
	● 从刀位	-	-	-	-	-	xx	-
	● 向刀库	xx	-	-	-	xx	xx	-
	● 向刀位	xx	-	-	-	xx	xx	-

图例:

- 数据不相关

x: 传输步骤表中的传输步骤编号 (1 到 n)

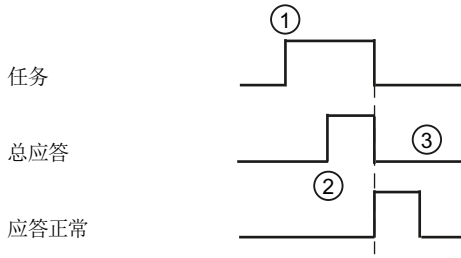
xx: 刀具的刀库号、刀位号

zz: 装载/卸载或换刀位的刀库号、刀位号

状态一览提供以下信息:

- 状态 1、6、105、201 和 204 必须在应答步骤表中根据其含义与传输步骤组成有意义的应答步骤。
- 如果状态 1 以两个传输步骤号 = 0 编码, 则此应答步骤用作当前通过中间应答达到的状态的结束应答。
- 如果已将刀具从真实刀库向缓冲区移动 (换位, MVTOOL), 则在应答状态 6 或总应答时为此刀具预留源刀位 (\$TC_MPP4 位 1 和位 2)。其特性与换刀时从刀库卸载刀具时相同。状态为 1 时, 在执行换位或 MVTOOL 时不预留源刀位。
- 状态 3 和 7 只需在应答步骤表中编码一次, 因为不会对传输步骤进行分析。
- 状态 99 不需进行编码, 而是通过“总应答”的位设定。

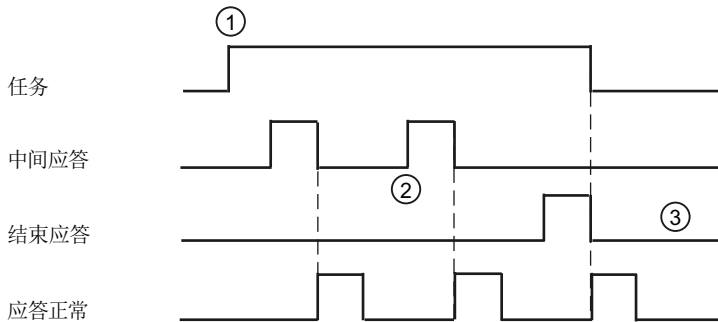
任意任务处理的典型时序图，采用总应答



信号描述:

- ① PLC 用户程序通过信号 DB43xx.DBX0.0 (任务) 的 0/1 脉冲沿识别出刀具管理已发出新任务。
- ② PLC 用户程序使 DB42xx.DBX0.0 (总应答) 中的应答信号置位。触发 0/1 脉冲沿时, PLC 固件开始向刀具管理传输应答。
- ③ 在成功将应答传输至刀具管理后, PLC 固件将信号“应答正常”设置为 1 (持续一个 PLC 周期), 并同时 will 将信号“任务”与应答位复位为 0。

任意任务处理的典型时序图，采用中间应答和结束应答



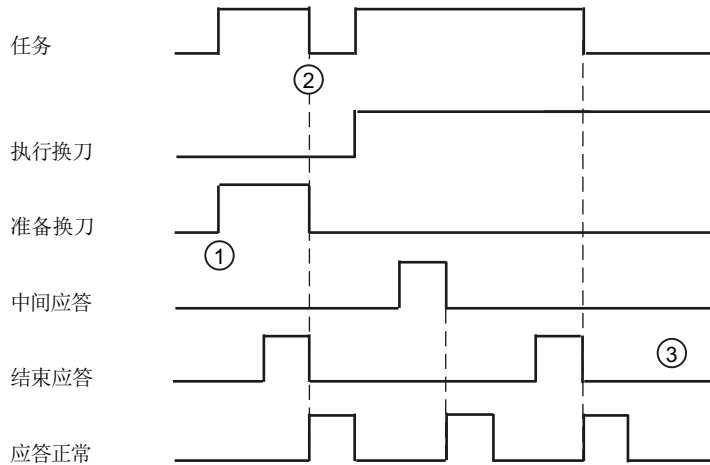
信号描述:

- ① PLC 用户程序通过信号 DB43xx.DBX0.0 (任务) 的 0/1 脉冲沿识别出刀具管理已发出新任务。
- ② PLC 用户程序通过应答状态 105 应答 DB9900, DB9901 和 DB9902 中配置的传输步骤。借助应答的传输步骤, 刀具管理将对刀具位置进行更新。
- ③ PLC 用户程序通过应答状态 1 通知任务已完成。在成功将应答传输至刀具管理后, PLC 固件将信号“应答正常”设置为 1 (持续一个 PLC 周期), 并同时 will 将信号“任务”与应答位复位为 0。

刀具管理的时序图，采用程序段分割（铣削的机床数据设置）

Txx M6 ;

使用 M6 调用程序 L6（初始设置）



信号描述:

- ① PLC 用户程序接收到新任务。“准备换刀”和“执行换刀”任务连续发出。Txx 和 M206 已编程在单独的 NC 程序段中。在 DB43xx 中的接口中仅存在任务“准备换刀”。“执行换刀”任务只有发出“准备任务”结束应答后才输出。
- ② 应答“准备换刀”任务时，位 DB43xx.DBX0.0（任务）复位。一旦换刀指令（M206）已在 NC 主运行中执行完毕，之后立即会在接口上输出新任务。
- ③ “执行换刀”任务被作为普通任务应答。将“应答正常”的信息反馈给刀具管理，同时复位该任务位。

任务描述（“执行换刀”和“准备换刀”）不会被复位。DB43xx 的字节 1 在下一个任务时才重新写入。

说明

MD20270, MD20310:

B43xx.DBB1 的接口特性以及 NC 程序段执行，会受到 MD20270 :

\$MC_CUTTING_EDGE_DEFAULT 以及 MD20310:

\$MC_TOOL_MANAGEMENT_MASK 位 5、6、7 和 8 中设置的影响。

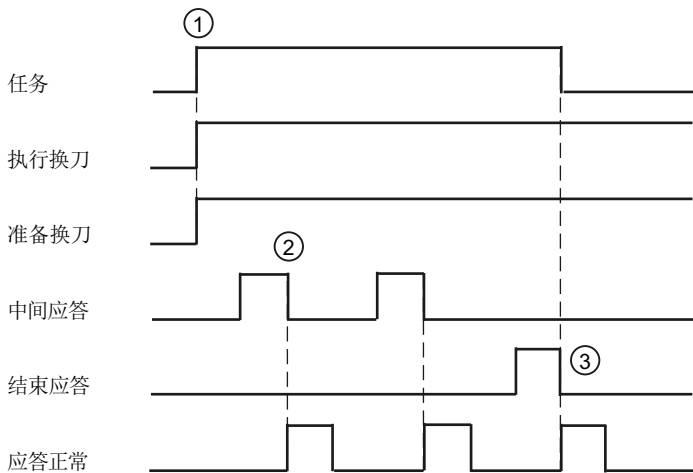
这里描述的时序是机床数据的缺省设置。

刀具管理的时序图，不采用程序段分割（铣削的机床数据设置）

Txx Myy ;

Myy 是 MD22560 : \$MC_TOOL_CHANGE_M_CODE 中的设置

不建议使用此编程方式!



信号描述:

- ① PLC 用户程序接收到新任务。“准备换刀”和“执行换刀”任务一起发出。Txx 和 M206 已编程在同一 NC 程序段中。
- ② 会对多个中间步骤进行应答。任务状态保持不变。刀具位置由刀具管理根据应答的传输步骤不断更新。
- ③ 将“应答正常”的信息反馈给刀具管理，同时复位该任务位。
任务描述（“执行换刀”和“准备换刀”）不会被复位。DB43xx 的字节 1 在下一个任务时才重新写入。

说明

MD20310: \$MC_TOOL_MANAGEMENT_MASK

DB43xx.DBB1 中接口的特性受 MD20310 位 10 的设置的影响。
这里描述的时序是机床数据的缺省设置。

9.4.4 配置步骤表

配置步骤表

在编程工具中通过菜单“库”→“特殊数据块”使用传输步骤表（TM_CTS, DB9900 和 TM_VTS, DB9901）和应答步骤表（TM_ACK, DB9902）。通过双击将模块复制到项目中。

数据块的结构为固定设定。

数据块尚未填充必要的数，必须由用户通过编程工具中的菜单指令“检视”→“数据块”编辑。在编程工具中写入数据块的初始值即可配置固定步骤表（TM_CTS, DB9900 和 TM_ACK, DB9902）。

数据块的初始值会与 PLC 用户程序一同装载至控制系统。修改过的初始值在 PLC 重启后才生效。

传输步骤配置

通过 PLC 用户程序更改刀具和刀库位置后，必须通知刀具管理。在此提供一个包含了所有需要应答/通知的单个机械运动表作为辅助。表格包含了刀具传输时的起始和目标位置，以及将刀库刀位定位至传输位（换刀/装载/卸载位）时的刀库位置和传输位名称。

- 传输步骤 1 到 64 在 TM_CTS（DB9900）中固定配置，只可以通过重新载入修改。
- TM_VTS（DB9901）中的传输步骤 101 到 164 可通过 PLC 用户程序完全或局部覆盖（例如通过输入当前刀库刀位）。

任务中的位置编码

在固定传输步骤表中，真实刀库的刀位不是以其实际值（例如 1/14 对应刀库 1 刀位 14），而是以符号值 (0/1) 或 (0/2) 标记。否则，大刀库的传输步骤表会很大。

符号值的含义如下：

刀库/刀位	含义
(0/1)	需要使用任务中单个刀具或新刀具的源位置。
(0/2)	需要使用任务中旧刀具的目标位置。
(0/3)	需要使用任务中单个刀具或新刀具的目标位置。

这种符号标记法只能用于同步中间应答和结束应答，因为异步信息无任务作为参考。

示例：固定传输步骤表

传输步骤	地址 DB9900	名称	起始值	注释
1	0.0	SrcMag_1	0	传输步骤的源刀库号
	2.0	SrcPos_1	1	传输步骤的源位置号
	4.0	DstMag_1	0	传输步骤的目标刀库号
	6.0	DstPos_1	1	传输步骤的目标位置号
2	8.0	SrcMag_2	0	传输步骤的源刀库号
	10.0	SrcPos_2	1	传输步骤的源位置号
	12.0	DstMag_2	9998	传输步骤的目标刀库号
	14.0	DstPos_2	2	传输步骤的目标位置号

整个步骤的示例

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
1	0	1	0	1	准备换刀：将刀库定位至新刀具的换刀位。
2	0	1	9998	2	换刀：将刀具从刀库移动至爪 1
3	9998	1	9998	3	换刀：将刀具从主轴移动至爪 2
4	9998	2	9998	1	换刀：将刀具从爪 1 移动至主轴
5	9998	3	0	2	换刀：将刀具从爪 2 移动至刀库

示例：可变传输步骤表

传输步骤	地址 DB9901	名称	起始值	注释
101	0.0	SrcMag_101	1	传输步骤的源刀库号
	2.0	SrcPos_101	0	传输步骤的源位置号
	4.0	DstMag_101	9998	传输步骤的目标刀库号
	6.0	DstPos_101	1	传输步骤的目标位置号

传输步骤	地址 DB9901	名称	起始值	注释
102	8.0	SrcMag_102	1	传输步骤的源刀库号
	10.0	SrcPos_102	0	传输步骤的源位置号
	12.0	DstMag_102	9998	传输步骤的目标刀库号
	14.0	DstPos_102	2	传输步骤的目标位置号

整个步骤的示例

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
101	1	0	9998	1	“准备换刀”：将刀库刀位定位至换刀位。必须通过 PLC 用户程序输入源位置。
102	1	0	9998	2	“准备换刀”：将刀具从刀库移动至缓冲区。必须通过 PLC 用户程序输入源位置。

9.4.5 配置应答步骤

应答步骤配置

PLC 31 提供一些应答步骤用于确认刀具和刀库移动，这些步骤通过用户接口上相应的位激活。这些应答步骤（特例为应答步骤 9：总应答）的数据保存在应答步骤表 TM_ACK (DB9902) 中。应答步骤中包含旧刀具和新刀具的传输步骤（传输步骤表的传输步骤编号）和应答状态。

重要的是此表中包含状态 3 的应答步骤，从而可在必要时复位故障。此应答步骤中传输的编号为 0。

传输步骤 0 的特殊含义

只执行指定给应答步骤的传输步骤。如果仅指定了一个传输步骤或未指定传输步骤，当刀具的传输步骤 = 0 时，不会执行刀具传输。刀具不存在或停留在原来的刀位。

示例：应答步骤表

应答步骤	地址 DB9902	名称	起始值	注释
1	0.0	TsNewT_1	0	新刀具的传输步骤编号
	1.0	TsOldT_1	0	旧刀具的传输步骤编号
	2.0	State_1	3	发送至 NCK 的状态
2	4.0	TsNewT_2	1	新刀具的传输步骤编号
	5.0	TsOldT_2	0	旧刀具的传输步骤编号
	6.0	State_2	1	发送至 NCK 的状态

整个步骤的示例

应答步骤	传输步骤		应答状态	注释
	新刀具	旧刀具		
1	0	0	3	“取消任务”指令
2	2	0	1	准备刀具：将刀具从刀库（换刀位）移动至夹爪 1。

通过在用户接口中设置相应的位进行应答：

- DB40xx 用于装刀/卸刀，刀具换位或刀库定位
- DB42xx 用于“准备换刀”和“执行换刀”

在结束任务后，PLC 固件复位应答位，复位持续一个 PLC 周期。

在执行应答的同一个数据块中，位 100.0 中输出反馈“应答正常”；位 100.1 中输出静态信号“应答故障”，字节 104 中输出故障状态，字节 DBB108~DBB111 中输出上次设置的应答位，这些输出都持续一个 PLC 周期。用户可根据这些位识别哪些应答位触发了故障。如果通过 PLC 用户程序错误地设置了多个应答位，则这些位也 1:1 输入映射。用户在 DB40xx.DBX9.0 或 DB42xx.DBX9.0 中应答故障可以复位故障状态。

9.4.6 调整 PLC 用户程序

调整 PLC 用户程序

PLC 用户程序的任务是：控制机械步骤、监控和避免可能发生的碰撞、应答刀具的位置变化。

另见

在工具箱光盘上的 PLC 项目中，有带转塔刀库的车床，和带链式刀库和双夹爪的铣床的传输步骤及应答示例。

- 铣床应用示例 (页 333)
- 车床应用示例 (页 320)

这些程序段是应答不同刀具管理任务的功能示例。

应答任务

很多刀具移动或刀具管理任务可直接通过接口位 0.0 上的总应答来应答，而不需进行中间应答。

例如：

- 转塔刀库的旋转
- 装刀/卸刀（仅在无附加缓冲区的系统中，如装卸系统，装刀器等）
- 更换手动刀具
- 刀库定位

应答时的规则

如果中间步骤有意义，则在应答时必须遵守下面几个规则：

PLC 用户程序必须确保所有应答正确地传输至刀具管理。

- 一次只能向刀具管理发送一个应答信号。
- 仅在待处理任务中允许同步应答。
- 只能使用有效的传输步骤编号（1 - 64, 100 - 164）。对于异步信息，在状态 201 时至少须设定一个传输步骤，在状态 204 时至少须设定一个新刀具的传输步骤。
- 传输步骤中的编码位置只能用于同步应答，且只能使用值 0/1、0/2 或 0/3。

- 不允许使用非法应答状态。
- 通过任务定位刀库只能接收同步应答（结束应答）。必须通过异步信息将中间位置发送至刀具管理
- 必须由 PLC 基本程序复位应答信号。在设置了应答位后，用户接口不可更改，直到 DB41xx/DB43xx DBB100 中输出反馈！
- 带两个应答步骤的异步信息必须在换刀接口（DB42xx）上应答。

9.4.7 刀库刀位信息

概述

通过 DB1200 的“NC 通讯”接口，可以在一个任务中读取最多 8 个 NC 变量。

变量 \$TC_MPP2，下标 7: 刀位类型

参数设置：

从 NC 读取变量	地址	信号	有效值
任务	DB1200.DBX0.0	启动	0/1
	DB1200.DBX0.1	写入变量	0
	DB1200.DBB1	变量数量	1 ... 8
参数	DB120x.DBW1000	变量下标	7
	DB120x.DBW1002	刀位号	1 ... 31999
	DB120x.DBW1004	刀库号	1 ... 9999
结果	DB1200.DBX2000.0	任务结束	0/1
	DB1200.DBX2000.1	任务出错	0/1
	DB120x.DBX3000.0	变量有效	0/1
	DB120x.DBB3001	访问结果	0/3/5/10
	DB120x.DBW3004	NC 变量的数据	n

图例：

n > 0: 虚拟刀位的刀位类型

n = 0: “match all”（缓冲区）

n = 9999: 未定义（非虚拟刀位）

变量 \$TC_MPP4, 下标 8: 位置状态

参数设置:

从 NC 读取变量	地址	信号	有效值
任务	DB1200.DBX0.0	启动	0/1
	DB1200.DBX0.1	写入变量	0
	DB1200.DBB1	变量数量	1 ... 8
参数	DB120x.DBW1000	变量下标	8
	DB120x.DBW1002	刀位号	1 ... 31999
	DB120x.DBW1004	刀库号	1 ... 9999
结果	DB1200.DBX2000.0	任务结束	0/1
	DB1200.DBX2000.1	任务出错	0/1
	DB120x.DBX3000.0	变量有效	0/1
	DB120x.DBB3001	访问结果	0/3/5/10
	DB120x.DBW3004	NC 变量的数据	n

图例:

n = 1 禁用

n = 2 未占用 (<> 占用)

n = 4 预留用于缓冲区中的刀具

n = 8 预留用于待装载的刀具

n = 16 左半刀位占用

n = 32 右半刀位占用

n = 64 上半刀位占用

n = 128 下半刀位占用

变量 \$TC_MPP6, 下标 9: 该位置上刀具的 T 编号

参数设置:

从 NC 读取变量	地址	信号	有效值
任务	DB1200.DBX0.0	启动	0/1
	DB1200.DBX0.1	写入变量	0
	DB1200.DBB1	变量数量	1 ... 8

从 NC 读取变量	地址	信号	有效值
参数	DB120x.DBW1000	变量下标	9
	DB120x.DBW1002	刀位号	1 ... 31999
	DB120x.DBW1004	刀库号	1 ... 9999
结果	DB1200.DBX2000.0	任务结束	0/1
	DB1200.DBX2000.1	任务出错	0/1
	DB120x.DBX3000.0	变量有效	0/1
	DB120x.DBB3001	访问结果	0/3/5/10
	DB120x.DBW3004	NC 变量的数据	n

n = 设置的刀位上的刀具的 T 号

故障（用于所有变量）

发生故障时保持 DB120x.DBX3000.0 = 0，并且在访问结果中增加记录：

DB120x.DBB3001 中的值	
0	无故障
3	不允许访问对象
5	无效地址
10	对象不存在

9.4.8 PI 服务：TMMVTL

功能

通过程序级服务 TMMVTL，可由 PLC 触发刀具换位任务。在顺利执行“PI Start”后，刀具管理为设定源刀位上的刀具在目标刀库中搜索空刀位。之后 PLC 接收刀具换位任务（用户接口 DB41xx.DBB0）。

参数设置

启动 NC 区域的 PI 服务：

PI 服务	地址	信号	有效值
任务	DB1200.DBX4000.0	启动	0/1
	DB1200.DBB4001	PI 索引	5
参数	DB1200.DBW4004	刀具号 (内部 T 号)	1 ... 31999
	DB1200.DBW4006	源刀位号	1 ... 31999
	DB1200.DBW4008	源刀库号	1 ... 31999
	DB1200.DBW4010	目标刀位号	-1
	DB1200.DBW4012	目标刀库号	1 ... 32000
结果	DB1200.DBX5000.0	任务结束	0/1
	DB1200.DBX5000.1	任务出错	0/1

可以有选择的对刀具，通过 T 编号，或者通过位置以及刀库编号进行说明。未使用的规格数值为 1。

通过参数设置目标刀位号为-1，就是按照设定的查找方式，在整个刀库中为刀具寻找一个空刀位。如果对目标刀位进行了说明，那么就会自动检测，这个刀位是否与刀具所描述的目标刀位号相对应。

在输入目标刀库号为 -1 时，就会按照 \$TC_MDP2 给出的分类，对所有缓冲区的刀具进行查找。

示例：

- 在从缓冲区送回刀具时（例如抓刀器和/或换刀机构），在异步送回期间可能需要在刀库中进行明确的空刀位搜索。在这种情况下 PLC 不需要标示源刀位，PI 服务会搜索适合的刀位。
- 刀具需从后台刀库移动至前台刀库。

9.5 示例：装刀/卸刀

编程

在装刀时刀库直接插入刀库或主轴，卸刀时刀具直接从刀库移除。通常操作人员或用户程序的一次应答（刀套重新闭合）足够作为步骤结束的信息。此处不需要配置传输步骤。可设置 DB40xx.DBX0.0 中的总应答。

向刀具管理发送应答：

应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
xxx	DB4000.DBX0.0	--	--	(99)

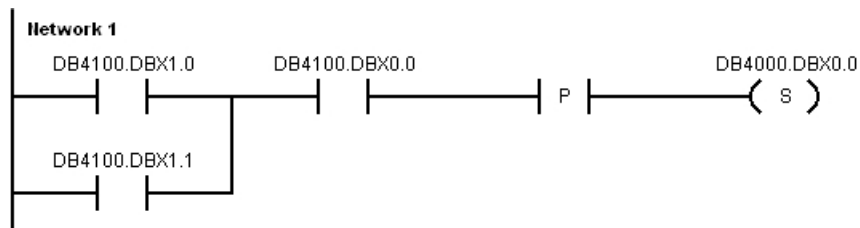


图 9-2 PLC 用户程序的编程

采样装卸系统的各个装载步骤、或者从主轴向刀库的刀具传输由更多的异步信息执行。

说明

手动刀具

使用手动刀具 (页 269) 时，请注意其特性。

加载至主轴或刀库中的刀具并不是手动刀具，因此，使用 T0/M06 转入刀库。

以下章节说明了加载过程的其他步骤。

通过主轴装刀，预选刀库刀位

可直接在所需的空刀库刀位上创建新刀具，或将刀具列表中、刀库中没有的刀具通过“装刀”对话框装入刀库刀位。

1. 此时总是在装载位的接口上输出刀具管理任务。必须应答此任务。
2. 之后使用 Txx M6 或通过异步传输将尚不在刀库中的刀具换至主轴。
3. 然后手动将刀具插入主轴，并且使用 T0 M6 转入刀库。

刀具可过尺寸或固定刀位编码。

通过主轴装刀，不预选刀库刀位

可直接在主轴创建新刀具，或将刀具列表中、刀库中没有的刀具通过“装刀”对话框装入主轴。

1. 此时总是在装载位的接口上输出刀具管理任务。必须应答此任务。
2. 然后手动将刀具插入主轴，并且使用 T0 M6 转入刀库。刀具管理会选择一个可放入刀具的空刀位。

刀具可过尺寸或固定刀位编码。

直接装刀至刀库

将所需的刀库刀位定位至装载位。可直接在所需的空刀库刀位上创建新刀具，或将刀具列表中、刀库中没有的刀具通过“装刀”对话框装入刀库刀位。

1. 此时总是在装载位的接口上输出刀具管理任务。必须应答此任务。
2. 现在将刀具插入刀库。

该方法不受限制，无前提条件。

9.6 示例：更换手动刀具

编程

在机床数据 MD22562: \$MC_TOOL_CHANGE_ERROR_MODE 位 1=1 时，由 NC 零件程序选择了未分配刀库的附加刀具。所选择的刀具必须手动装入机床，加工结束后仍需手动卸除（“手动刀具”）。

操作人员应注意，主轴中所装刀具的数据组存在于 NCK 中，或者针对 NCK 中所存储的数据组向主轴中装入适合的刀具。

说明

用户必须通过 PLC 用户程序确保换刀符合安全规程。

执行换刀任务时，刀具管理通过 DB43xx.DBX1.5 和 DBX1.6 向 PLC 用户程序通知手动刀具是否参与。操作人员会收到换刀请求：报警 17212：“通道 %1，手动刀具 %2，Duplo 号 %3，装入刀套 %4”，或报警 17214：“卸除主轴/刀套上的手动刀具”。

在换刀结束、PLC 发出应答后，这些报警复位。

初始位置 1

主轴上的手动刀具应更换为其它手动刀具。

刀具管理任务发送给 PLC 用户程序（换刀）：

DB4300.DBX0.0, DBX1.2, DBX1.5 和 DBX1.6（“准备换刀”）

DB43xx.DBW6	源刀库号	9999
DB43xx.DBW8	源刀位号	1
DB43xx.DBW10	目标刀库号	9999
DB43xx.DBW12	目标刀位号	1

应答“准备换刀”

应答步骤	应答位	新刀具传输步骤	旧刀具传输步骤	状态
xxx	DB4200.DBXx.x	0	0	1

因此接口上出现：DB4300.DBX0.0 /1.1, DBX1.5 和 DBX1.6（“执行换刀”）

否则任务保持不变，刀具仍停留在初始位置。为了对卸除旧刀具进行应答，可以加入一个中间步骤。

同步应答

刀具已不在主轴上。

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
6 DB9900.DBW40	9998	1	9999	1	从主轴上卸除刀具

应答步骤	应答位	新刀具传输步骤	旧刀具传输步骤	状态
xxx	DB4200.DBXx.x	0	6	105

PLC 用户程序发出中间应答“主轴为空”、主轴中装入新刀具后，总应答发出，结束换刀。

应答步骤	应答位	新刀具传输步骤	旧刀具传输步骤	状态
	DB4200.DBX0.0			(99)

初始位置 2

手动刀具在主轴上，需要和刀库中的刀具互换。

刀具管理任务发送给 PLC 用户程序（换刀）：

DB4300.DBX0.0, DBX1.2 和 DBX1.6（“准备换刀”）

DB43xx.DBW6	源刀库号	1
DB43xx.DBW8	源刀位号	6
DB43xx.DBW10	目标刀库号	9999
DB43xx.DBW12	目标刀位号	1

应答“准备换刀”：

应答步骤	应答位	新刀具传输步骤	旧刀具传输步骤	状态
xxx	DB4200.DBXx.x	0	0	1

因此接口上出现：DB4300.DBX0.0 /1.1 和 1.6（“执行换刀”）

否则任务保持不变，刀具仍停留在初始位置。

9.6 示例：更换手动刀具

同步应答

旧刀具已不在主轴上。

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
6 DB9900.DBW40	9998	1	9999	1	异步通知，从主轴上卸除刀具

应答步骤	应答位	新刀具传输步骤	旧刀具传输步骤	状态
xxx	DB4200.DBXx.x	0	6	105

主轴现在为空，旧刀具在刀库外。

下一步骤

同步应答 新刀具到夹爪 1

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
3 DB9900.DBW16	0	1	9998	2	新刀具到夹爪 1

应答步骤	应答位	新刀具传输步骤	旧刀具传输步骤	状态
xxx	DB4200.DBXx.x	3	0	105

任务保持不变。

下一步骤

同步应答 新刀具从夹爪 1 到主轴

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
4 DB9900.DBW24	9998	2	9998	1	新刀具从夹爪 1 到主轴

应答步骤	应答位	新刀具传输步骤	旧刀具传输步骤	状态
xxx	DB4200.DBXx.x	4	0	105

此时刀具移动完成。

结束应答

应答步骤	应答位	新刀具传输步骤	旧刀具传输步骤	状态
xxx	DB4200.DBXx.x	0	0	1

“新刀具从夹爪 1 到主轴”这一步骤可以省略并用结束应答替代。这时同样会通知刀具管理，全部刀具都移动到了目标位置。

应答步骤	应答位	新刀具传输步骤	旧刀具传输步骤	状态
	DB4200.DBX0. 0			(99)

9.7 车床应用示例

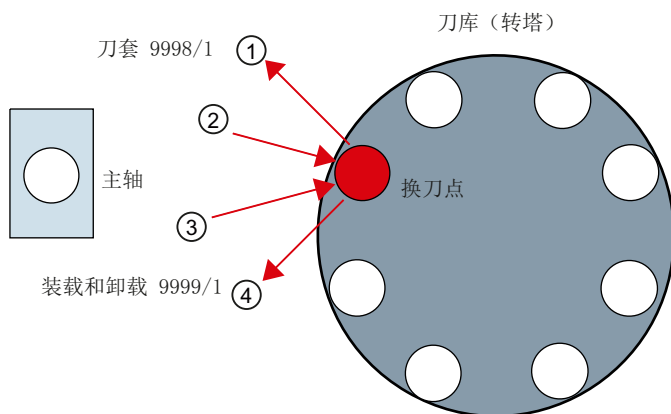
9.7.1 示例：车床，带转塔刀库（MAG_CONF_MPF）

示例文件

刀库配置程序在工具盒中。

该程序可读入到控制系统中并根据具体设备加以调整。

配置



1 ... 4 传输步骤

图 9-3 车床，带转塔刀库

程序描述

开始时删除所有旧的刀库定义和刀具。在下面的程序流程中，通过写入刀库参数重新创建和定义所有的刀库和缓冲区。

在 N70 中可以确定刀具和刀位的查找策略。

对转塔刀库而言，对所有的刀位进行固定编码是很有必要的。对于刀库类型 3，是在 N320 $\$TC_MAP3[NUM_MAG] = 81$ (位 6 = 1) 中进行设置的。

链式刀库的刀位在 N430 ~ N500 中设置。刀位类型 = 0 表示刀位可以装载不同刀位类型的刀具。

从 N520 开始可设置缓冲区。

从 N920 起可以指定缓冲区为主轴/刀套和刀库。

在刀库配置程序完成运行后，重新启动 NC（NCK Reset）。

另见

所用参数的详细描述见 SINUMERIK 840D sl 刀具管理功能手册。

示例 MAG_CONF_MPF

1. 设备配置：

- 1 个转塔刀库带 8 个刀位（可在 N40 中设置）
- 1 个装载位
- 3 个缓冲刀位（可在 N50 中设置，从 N540 开始分配）

2. 零件程序：

```

;MAG_CONF_MPF
N10 def int NUM_MAG,MAG_TYPE, LOCATIONS,
PLACE, NUM_BUFFER, NUM_LOAD, PLACE_SEARCH
;
N20 NUM_MAG = 1 ; 刀库号
N30 MAG_TYPE = 3 ; 刀库类型（1：链式，3：转塔，5：盒式刀库）
N40 LOCATIONS = 8 ; 刀位数
N50 NUM_BUFFER = 1 ; 缓冲区数量（主轴，夹爪）
N60 NUM_LOAD = 1 ; 装载位数量
N70 PLACE_SEARCH = 257 ; 查找策略的类型
; = 257 位 13=0 刀位上的旧刀具不更换为新刀具
; 用于选择刀库的设置
; = 12289 位 13=1 刀位上的旧刀具更换为新刀具
; 链式刀库的设置
N80;
N90;
; 检查参数
N100 STOPRE
N110 if ((NUM_MAG==0) or (LOCATIONS==0))
N120 Err1:STOPRE
N130 MSG("Wrong Parameter --> Cancel")
N140 G04 F4
N150 STOPRE
N160 M0
N170 GOTOB Err1
N180 endif
N190; 刀库配置
N200;

```

```

N210;
N220; 当创建了刀库 1 时，删除旧的数据
N230 if NUM_MAG ==1
N240 $TC_MAP1[0]=0 ; 删除刀库
N250 $TC_DP1[0,0]=0 ; 删除刀具
N260 STOPRE
N270 endif
; 配置
;
N280 $TC_MAMP2= PLACE_SEARCH ; 查找策略的类型
;
; 刀库
; 建立刀库
N290 $TC_MAP1[ NUM_MAG ]= MAG_TYPE
N300 $TC_MAP2[ NUM_MAG ]="刀库"<<NUM_MAG
N310 if MAG_TYPE == 3
N320 $TC_MAP3[ NUM_MAG ]=81 ; 刀库状态，转塔刀库中的全部刀位已固定编码
N330 else
N340 $TC_MAP3[ NUM_MAG ]=17 ; 刀库状态
N350 endif
N360 $TC_MAP4[ NUM_MAG ]=-1
N370 $TC_MAP5[ NUM_MAG ]=-1
N380 $TC_MAP6[ NUM_MAG ]=1 ; 刀库的行数
N390 $TC_MAP8[ NUM_MAG ]=0
N400 $TC_MAP9[ NUM_MAG ]=0
N410 $TC_MAP7[ NUM_MAG ]=LOCATIONS ; 刀位数
N420 $TC_MAP10[ NUM_MAG ]=PLACE_SEARCH
;
; 刀位
N430 for PLACE=1 to LOCATIONS
N440 STOPRE
N450 $TC_MPP1[ NUM_MAG, PLACE ]=1 ; 刀位类型
N460 $TC_MPP2[ NUM_MAG, PLACE ]=0 ; 刀位类型，0 表示可接受任何刀位类型的刀具
N470 $TC_MPP3[ NUM_MAG, PLACE ]=1 ; 邻近刀位监控打开（关闭时为 0）
N480 $TC_MPP4[ NUM_MAG, PLACE ]=2 ; 刀位状态
N490 $TC_MPP5[ NUM_MAG, PLACE ]=PLACE ; 刀位类型索引
N500 endfor
N510 STOPRE
;
N520; 缓冲区刀库定义（始终为编号 9998）
;
N530 $TC_MAP1[9998]=7 ; 刀库类型 7: 缓冲区
N540 $TC_MAP2[9998]="BUFFER"<<NUM_MAG
N550 $TC_MAP3[9998]=17 ; 刀库状态
    
```



```

N560 $TC_MAP6[9998]=1 ; 行数
N570 $TC_MAP7[9998]=NUM_BUFFER ; 刀位数
;
; 缓冲区刀位
; 主轴
N580 $TC_MPP1[9998,1]=2 ; 刀位类型 (此处为主轴)
N590 $TC_MPP2[9998,1]=0 ; 刀位类型 (此处始终为 0)
N600 $TC_MPP3[9998,1]=0 ; 邻近刀位监控关闭
N610 $TC_MPP4[9998,1]=2 ; 刀位状态
N620 $TC_MPP5[9998,1]=1 ; 刀位类型索引
;
N630; 夹爪
N640 FOR PLACE=2 to NUM_BUFFER
N650 STOPRE
N660 $TC_MPP1[9998,PLACE]=3 ; (此处为夹爪)
N670 $TC_MPP2[9998,PLACE]=0 ; (此处始终为 0)
N680 $TC_MPP3[9998,PLACE]=0 ; 邻近刀位监控关闭
N690 $TC_MPP4[9998,PLACE]=2 ; 刀位状态
N700 $TC_MPP5[9998,PLACE]=PLACE ; 刀位类型索引
N710 endfor
N720 STOPRE
;
;
N730; 装载刀库的定义 (始终为编号 9999)
;
N740 $TC_MAP1[9999]=9 ; 刀库类型 9: 装载刀库
N750 $TC_MAP2[9999]="装载刀库"<<NUM_MAG
N760 $TC_MAP3[9999]=17 ; 刀库状态
N770 $TC_MAP4[9999]=-1
N780 $TC_MAP5[9999]=-1
N790 $TC_MAP6[9999]=1 ; 行数
N800 $TC_MAP7[9999]=NUM_LOAD ; 刀位数
N810 STOPRE;
;
N820; 装载刀库的刀位
;
N830 for PLACE=1 to NUM_LOAD
N840 STOPRE
N850 $TC_MPP1[9999,PLACE]=7 ; 刀位类型 装载位
N860 $TC_MPP2[9999,PLACE]=0 ; 刀位类型 (此处始终为 0)
N870 $TC_MPP3[9999,PLACE]=0 ; 邻近刀位监控关闭
N880 $TC_MPP4[9999,PLACE]=2 ; 刀位状态
N890 $TC_MPP5[9999,PLACE]=PLACE ; 刀位类型索引
N900 endfor

```

9.7 车床应用示例

```
N910 STOPRE  
;  
;  
N920; 偏移 (间距); 与刀库的间距  
;  
; 缓冲区  
N930 for PLACE=1 to NUM_BUFFER  
N940 $TC_MDP2[1,PLACE]=0  
N950 endfor  
N960 STOPRE  
;  
; 装载位  
N970 for PLACE=1 to NUM_LOAD  
N980 stopre  
N990 $TC_MDP1[1,PLACE]=0  
N1000 endfor  
  
N1010 M30 ; 结束
```

操作软件中的显示

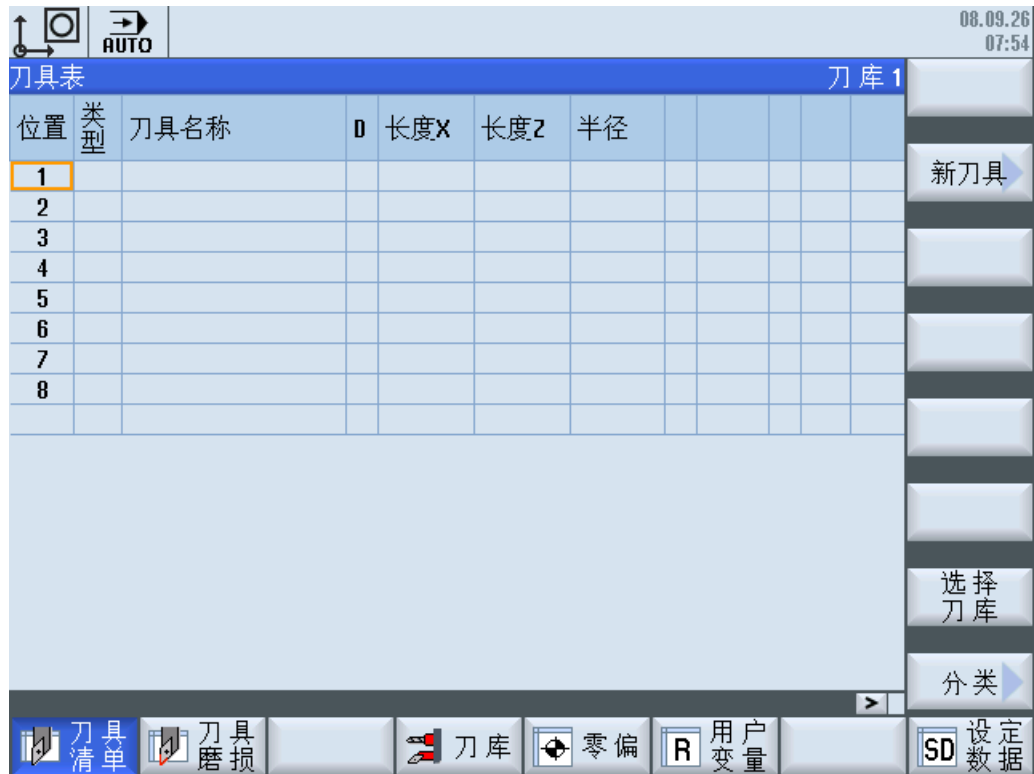


图 9-4 车床刀具列表

9.7.2 示例：应答步骤（车床）

应答步骤

一般而言车床的机械流程比铣床的简单。在前面章节所描述的配置（无附加缓冲区）中，换刀会在机械运动后、无传输步骤时被应答。PLC 用户程序应识别任务接口上到达的任务并执行机械运动。

向刀具管理发送总应答：

应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
--	DB4000.DBX0.0	--	--	(99)
--	DB4200.DBX0.0	--	--	(99)

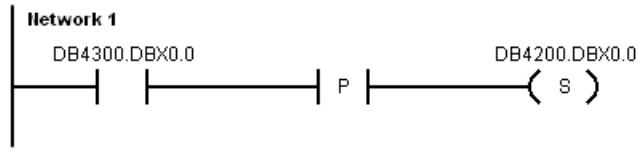


图 9-5 PLC 用户程序的编程

对于无刀具管理任务的刀库移动，可以使用异步通知。

向刀具管理发送应答：

应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
xxx	DB4000.DBXx.x	101	0	204

传输步骤	来源		目标		注释
	刀库	刀位	刀库	刀位	
101 DB9901.DBW0	1	n	9998	1	刀库 1 中的可变刀位在与主轴的切换点上。

n 在此处是 PLC 用户程序实际向可变传输表中输入的刀位号(n ≠ 0)。

9.7.3 示例：车床的换刀循环

T 替代循环的传输变量：

选项“用于刀具管理的备用刀具”无效。

前提是每个刀库移动都已通知给刀具管理。

使用机床数据 MD10717: \$MN_T_NO_FCT_CYCLE_NAME（替代 T 功能的循环的名称，如“TCHANGE”）可设置，通过调用 T 指令来运行循环。循环在每次 T 调用时都会运行，无论是调用新刀具还是已激活的刀具。在该循环中，转塔的位置可定位在所编程的刀具处 (POSM)。这在选刀后对转塔手动定位并且未取消刀具时是很必要的。这种情况下 NC 不会在接口上发出新的换刀任务。

编程

在编程示例中会使用 POSM 在接口上发出刀库定位任务。在执行刀库定位时 NC 程序段执行由 PLC 用户程序控制。建议在大多数情况下在此期间设置读入禁止或进给禁止。只要一达到任务给定的设定位置（刀库位置异步通知），就会使用结束应答来结束任务。

```

PROC L6 SAVE SBLOF DISPLOF
  IF $C_T_PROG==1                ; T 是数字
    IF $C_T==0                    ; T=0
      T=0
    ENDIF
    IF $C_T>0
      IF $C_T<=$TC_MAP7[1]        ; 刀位存在吗?
        POSM($C_T)                ; 定位刀库
      ENDIF
      T=$C_T                       ; T 编程 刀位号
    ENDIF
  ENDIF

  IF $C_TS_PROG==1                ; T 是标识符
    _TNO_NEW=GETT($C_TS,1)         ; 询问 T 号
    IF _TNO_NEW>0                  ; T 编号存在吗?
      _TL_NEW=$A_MYMLN[_TNO_NEW]   ; 询问刀位号
    ENDIF
    IF _TL_NEW>0                   ; 刀具在刀库中吗?
      POSM(_TL_NEW)                ; 定位刀库
    ENDIF
    T=$C_TS                         ; T 编程 无地址扩展名
  ENDIF
M17

```

测量刀具时的刀具调用

在特定应用，如 JOG 模式下的测量中，需要将特定刀具（例如 Fraeser_15, Duplo 号 2）装入主轴或刀套，而不考虑刀具状态，例如：不管刀具是否禁用。现在，在重新使能前应对刀具进行测量。刀具调用指令的替代指令，不考虑刀具状态，选中 Duplo 号和刀套号，对此可以使用 NC 指令 TCA，同时在刀具测量时，在 JOG 模式下运行该指令。NC 指令 TCA（“刀具名称”、Duplo 号、刀架号）与 T 指令在 PLC 上的警报和命令发出形式相似。

边界条件

对于有效的 T 指令替代 (= 铣削工艺机床数据)，要注意以下边界条件：

- 在使用 TCA 指令时，MD10717 \$MN_T_NO_FCT_CYCLE_NAME 中定义的、在 T 调用时运行的循环不启动。
为确保在 TCA 编程时也能启动更换刀具循环，应重新定义 _TCA 中的语言指令 TCA (缺省设置 NC)。
- TCA 可以用一个替代循环来替代。替代循环的名称在 MD15710 \$MN_TCA_CYCLE_NAME 中输入。该替代循环是从重新定义的 TCA 功能循环中调用的 (Toolbox 中的 TCA.SPF)。
- T 替换和 TCA 替换可以在共同的替代循环中实现。

Toolbox 里有两个刀具调用替换的循环示例，这两个替换在共同的替代循环中。

说明

相应 MD20270 \$MC_CUTTING_EDGE_DEFAULT 的正确选择和 T 指令操作大致相同。
TCA 和 D 不可以在同一个组中进行编程。

以下示例对流程进行了描述：

```

PROC TCA (STRING [64] _TOOL_NAME, INT _DUPLO, INT _TH_NO)
; 版本: 01.00.07.2009 年 10 月 22 日
; 变更: 01.00.07.2009 年 10 月 22 日
TCA (_TOOL_NAME, _DUPLO, _TH_NO) ; NC 上的刀具调用
; 在此插入用于更换的机床功能
M17
    
```

文档

该功能可以独立于刀具管理来使用并在下列文档有详细描述：

功能手册之基本功能分册 (K1)，章节“通过子程序替换功能”

9.7.4 示例：程序段搜索和 MCALL

前提条件

在换刀子程序中通过模态调用循环 (例如：钻削循环) 进行程序段搜索时要考虑到模态调用仍会保留，并且要确保使用正确的刀具。

带计算的程序段搜索后，在搜索 ASUP 中调用换刀子程序 (L6)。只有当 L6 不是从 ASUP 中调用时，才允许取消模态调用。

编程示例

```
L6
N100 IF $P_MC == TRUE ; MCALL 有效
N110 IF $AC_ASUP <> 0 ; 通过搜索 ASUP 或者覆盖进行调用 → 不取消 MCALL
N140 ELSE
N150 MCALL
N160 ENDIF
N170 ENDIF
```

机床制造商同样必须确保允许带有轴运动的真实的换刀以及模态循环调用。如果编程的刀具在程序段搜索后和主轴的刀具不同，则必须禁止换刀。

文档

SINUMERIK Operate 828D 调试手册，章节“钻削的工艺循环”

9.7.5 示例：车床，带副主轴

刀库配置

在刀库配置中，刀库具有的刀位数量应为实际存在的两倍，例如 12 个刀位的转塔应配置 24 个刀位。刀位 1-12 为主主轴，刀位 13-24 为副主轴。

PLC 用户程序定位刀库时，例如在刀位 1 和刀位 13 时会到达相同的位置。每个真实的刀库刀位都对应一个虚拟的主主轴刀位和一个虚拟的副主轴刀位。

9.7.6 示例：空缓冲区测试

操作步骤

读取夹爪 1 和夹爪 2 中刀具的 T 号：

1. 在 PLC 用户程序中输入 DB1200 中的参数。
2. 在 DB1200.DBX0.0 中设置启动读取刀位状态。

如果成功执行了任务，结果从 DB1200.DBB3000 开始：

从 NCK 中读取变量	地址	信号	参数值
参数	DB1200.DBW1000	变量索引	9
	DB1200.DBW1002	刀位号	2
	DB1200.DBW1004	刀库号	9998
	DB1201.DBW1000	变量索引	9
	DB1201.DBW1002	刀位号	3
	DB1201.DBW1004	刀库号	9998
任务	DB1200.DBX0.1	写入变量	0
	DB1200.DBX0.2	PI 服务	0
	DB1200.DBB1	变量数量	2
	DB1200.DBX0.0	启动	→ 1
结果	DB1200.DBB2000.0	任务结束	1
	DB1200.DBX2000.1	任务出错	0
	DB1200.DBX3000.0	变量有效	1
	DB1200.DBB3001	访问结果	0
	DB1200.DBW3004	NCK 变量的数据	n
	DB1201.DBX3000.0	变量有效	1
	DB1201.DBB3001	访问结果	n
	DB1201.DBW3004	NCK 变量的数据	0

9.7.7 示例：将刀具从缓冲区传输到刀库

操作步骤

刀具应从缓冲区（例如夹爪）转入刀库中。使用 PI 服务 TMMVTL 从夹爪 1（刀库 9998，刀位 2）开始执行刀具的空刀位查找并产生刀具的换位任务。

在 PLC 用户程序中输入 DB1200 中的参数并在 DB1200.DBX4000.0 中设置 PI 服务的启动信号。

在 NCK 区域中启动 PI 服务	地址	信号	数值
参数	DB1200.DBW4004	刀具号	0
	DB1200.DBW4006	源刀位号	2
	DB1200.DBW4008	源刀库号	9998
	DB1201.DBW4010	目标刀位号	-1
	DB1201.DBW4012	目标刀库号	1
任务	DB1200.DBB4001	PI 下标	5
	DB1200.DBX4000.0	启动	→ 1
结果	DB1200.DBB5000.0	任务结束	1
	DB1200.DBX5000.1	任务出错	0

在 DB410x.DBX0.0 中得到缓冲区刀具的换位任务。刀库 1 中的目标刀位在 DB4100.DBW12 中。这样 PLC 用户程序可以执行必要的流程。

9.7.8 示例：重复“准备换刀”任务

工作流程：重复任务

在带有抓刀器和换刀机构的铣床上异步传输通知和任务可交叉进行：

- 对于旧刀具，已将抓刀器作为目标位置，用结束应答向刀具管理进行应答。
- 零件程序使用新刀具继续执行并需要在很短的时间后再次使用之前（旧）的刀具。
- 刀具管理会产生下一个准备换刀任务，将抓刀器作为下个新刀具的源位置。
- 同时 PLC 用户程序已将刀具从抓刀器中接收到换刀机构中。

- PLC 用户程序会异步通知从抓刀器到换刀机构的刀具移动并开始接着向刀库中传输。
- 在下一个时钟周期中用户程序会识别到新任务，将刀具从抓刀器中转换到主轴上。

在抓刀器中不再存在刀具！PLC 用户程序应识别该情况（监控：抓刀器或换刀机构中含有刀具吗？）。现在可以取消将刀具送回刀库。接着刀具管理可以请求重复任务（状态 7）：

- 刀具管理此时会得到通知，刀具在换刀机构中。
- 它会产生一个新的准备任务，将换刀机构作为新刀具的源位置。

说明

对重复任务“准备换刀”的应答只允许在任务“准备换刀”的结束应答之前进行！

向刀具管理发送应答：

应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
xxx	DB4000.DBXx.x	0	0	7

9.8 铣床应用示例

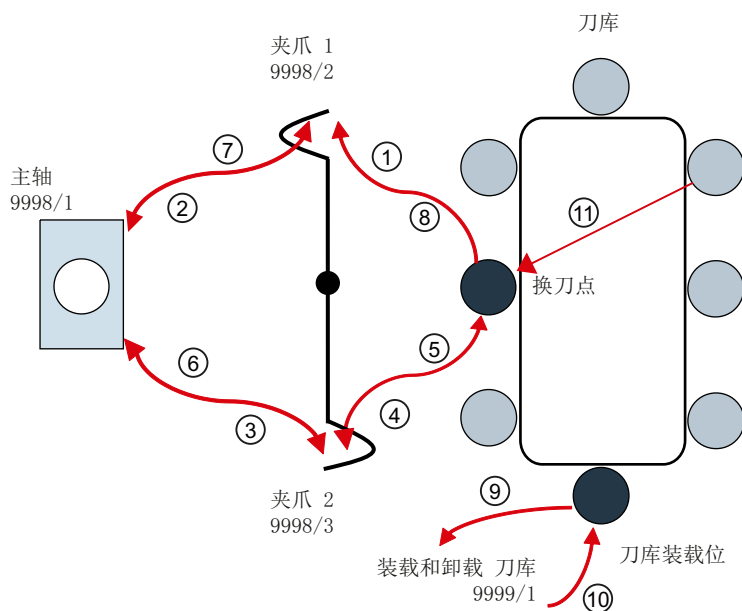
9.8.1 示例：铣床，带链式刀库和双夹爪(MAG_CONF_MPF)

示例文件

刀库配置程序在工具箱中。

该程序可读入到控制系统中并根据具体设备加以调整。

配置



1 ... 11 传输步骤

图 9-6 铣床，带链式刀库

缓冲区和装载位的名称：

刀库	刀位	含义
1	xx	真实刀库（链式，盘式，盒式），位置 xx
9998	1	主轴
9998	2	夹爪 1

刀库	刀位	含义
9998	3	夹爪 2
9999	1	刀库装载位

程序描述

开始时删除所有旧的刀库定义和刀具。在下面的程序流程中，通过写入刀库参数重新创建和定义所有的刀库和缓冲区。

在 N70 中可以确定刀具和刀位的查找策略。在此处确定刀具在移出主轴时是否可以直接移动到新刀具的刀位上。这样就可以在一个机械流程中进行换刀，可以达到较短的换刀时间。该过程不能用于选择刀库。

链式刀库的刀位在 N430 ~ N500 中设置。刀位类型 = 0 表示刀位可以装载不同刀位类型的刀具。

从 N520 开始可设置缓冲区。如有附加缓冲区（抓刀器，换刀机构...），应修改 N50 中的数量。

类似地，如有附加装载位在 N60 中修改。

从 N920 起可以指定缓冲区为主轴/刀套和刀库。

在刀库配置程序完成运行后，重新启动 NC（NCK Reset）。

另见

详细信息：

- 固定刀位编码的盘式刀库的配置参考转塔刀库的配置：示例：车床，带转塔刀库（MAG_CONF_MPF）（页 320）
- 所用参数的详细描述见 SINUMERIK 840D sl 刀具管理功能手册。

示例 MAG_CONF_MPF

1. 设备配置：
 - 1 个链式刀库带 8 个刀位（可在 N40 中设置）
 - 1 个装载位
 - 3 个缓冲刀位（可在 N50 中设置，从 N540 开始分配）
2. 零件程序：

```

;MAG_CONF_MPF
    
```

```

N10 def int NUM_MAG,MAG_TYPE, LOCATIONS,
PLACE, NUM_BUFFER, NUM_LOAD, PLACE_SEARCH
;
N20 NUM_MAG = 1 ; 刀库号
N30 MAG_TYPE = 1 ; 刀库类型 (1: 链式, 3: 转塔, 5: 盒式刀库)
N40 LOCATIONS = 8 ; 刀位数
N50 NUM_BUFFER = 3 ; 缓冲区数量 (主轴, 夹爪)
N60 NUM_LOAD = 1 ; 装载位数量
N70 PLACE_SEARCH = 12289 ; 查找策略的类型
; = 257 位 13=0 刀位上的旧刀具不更换为新刀具
; 用于选择刀库的设置
; = 12289 位 13=1 刀位上的旧刀具更换为新刀具
; 链式刀库的设置
N80;
N90;
; 检查参数
N100 STOPRE
N110 if ((NUM_MAG==0) or (LOCATIONS==0))
N120 Err1:STOPRE
N130 MSG("Wrong Parameter --> Cancel")
N140 G04 F4
N150 STOPRE
N160 M0
N170 GOTOB Err1
N180 endif
N190; 刀库配置
N200;
N210;
N220; 当创建了刀库 1 时, 删除旧的数据
N230 if NUM_MAG ==1
N240 $TC_MAP1[0]=0 ; 删除刀库
N250 $TC_DP1[0,0]=0 ; 删除刀具
N260 STOPRE
N270 endif
; 配置
;
N280 $TC_MAMP2= PLACE_SEARCH ; 查找策略的类型
;
; 刀库
; 建立刀库
N290 $TC_MAP1[NUM_MAG]= MAG_TYPE
N300 $TC_MAP2[NUM_MAG]="刀库"<<NUM_MAG
N310 if MAG_TYPE == 3
N320 $TC_MAP3[NUM_MAG]=81 ; 刀库状态, 转塔刀库中的全部刀位已固定编码

```

9.8 铣床应用示例

```

N330 else
N340 $TC_MAP3[NUM_MAG]=17 ; 刀库状态
N350 endif
N360 $TC_MAP4[NUM_MAG]=-1
N370 $TC_MAP5[NUM_MAG]=-1
N380 $TC_MAP6[NUM_MAG]=1 ; 刀库的行数
N390 $TC_MAP8[NUM_MAG]=0
N400 $TC_MAP9[NUM_MAG]=0
N410 $TC_MAP7[NUM_MAG]=LOCATIONS ; 刀位数
N420 $TC_MAP10[NUM_MAG]=PLACE_SEARCH
;
; 刀位
N430 for PLACE=1 to LOCATIONS
N440 STOPRE
N450 $TC_MPP1[NUM_MAG,PLACE]=1 ; 刀位类型
N460 $TC_MPP2[NUM_MAG,PLACE]=0 ; 刀位类型, 0 表示可接受任何刀位类型的刀具
N470 $TC_MPP3[NUM_MAG,PLACE]=1 ; 邻近刀位监控打开 (关闭时为 0)
N480 $TC_MPP4[NUM_MAG,PLACE]=2 ; 刀位状态
N490 $TC_MPP5[NUM_MAG,PLACE]=PLACE ; 刀位类型索引
N500 endfor
N510 STOPRE
;
N520; 缓冲区刀库定义 (始终为编号 9998)
;
N530 $TC_MAP1[9998]=7 ; 刀库类型: 7: 缓冲区
N540 $TC_MAP2[9998]="BUFFER"<<NUM_MAG
N550 $TC_MAP3[9998]=17 ; 刀库状态
N560 $TC_MAP6[9998]=1 ; 行数
N570 $TC_MAP7[9998]=NUM_BUFFER ; 刀位数
;
; 缓冲区刀位
; 主轴
N580 $TC_MPP1[9998,1]=2 ; 刀位类型 (此处为主轴)
N590 $TC_MPP2[9998,1]=0 ; 刀位类型 (此处始终为 0)
N600 $TC_MPP3[9998,1]=0 ; 邻近刀位监控关闭
N610 $TC_MPP4[9998,1]=2 ; 刀位状态
N620 $TC_MPP5[9998,1]=1 ; 刀位类型索引
;
N630; 夹爪
N640 FOR PLACE=2 to NUM_BUFFER
N650 STOPRE
N660 $TC_MPP1[9998,PLACE]=3 ; (此处为夹爪)
N670 $TC_MPP2[9998,PLACE]=0 ; (此处始终为 0)
N680 $TC_MPP3[9998,PLACE]=0 ; 邻近刀位监控关闭

```

```

N690    $TC_MPP4[9998,PLACE]=2          ; 刀位状态
N700    $TC_MPP5[9998,PLACE]=PLACE      ; 刀位类型索引
N710  endfor
N720  STOPRE
;
;
N730;  装载刀库的定义 (始终为编号 9999)
;
N740  $TC_MAP1[9999]=9                  ; 刀库类型: 9:  装载刀库
N750  $TC_MAP2[9999]="装载刀库"<<NUM_MAG
N760  $TC_MAP3[9999]=17                 ; 刀库状态
N770  $TC_MAP4[9999]=-1
N780  $TC_MAP5[9999]=-1
N790  $TC_MAP6[9999]=1                  ; 行数
N800  $TC_MAP7[9999]=NUM_LOAD           ; 刀位数
N810  STOPRE;
;
N820;  装载刀库的刀位
;
N830  for PLACE=1 to NUM_LOAD
N840    STOPRE
N850    $TC_MPP1[9999,PLACE]=7          ; 刀位类型 装载位
N860    $TC_MPP2[9999,PLACE]=0          ; 刀位类型 (此处始终为 0)
N870    $TC_MPP3[9999,PLACE]=0          ; 邻近刀位监控关闭
N880    $TC_MPP4[9999,PLACE]=2          ; 刀位状态
N890    $TC_MPP5[9999,PLACE]=PLACE      ; 刀位类型索引
N900  endfor
N910  STOPRE
;
;
N920;  偏移 (间距); 与刀库的间距
;
; 缓冲区
N930  for PLACE=1 to NUM_BUFFER
N940  $TC_MDP2[1,PLACE]=0
N950  endfor
N960  STOPRE
;
; 装载位
N970  for PLACE=1 to NUM_LOAD
N980    stopre
N990    $TC_MDP1[1,PLACE]=0
N1000 endfor

```

N1010 M30 ; 结束

操作软件中的显示



图 9-7 铣床刀具列表

9.8.2 流程图：换刀

换刀程序流程(PLC)

此处所述流程描述刀库与主轴之间的换刀。手动换刀以及装刀和卸刀不涉及。该流程详见：

- 示例：装刀/卸刀 (页 314)
- 示例：更换手动刀具 (页 316)

机床数据的预设置为，使用 T 指令编程时在接口上触发“准备换刀”任务：

N10 T = “刀具名” M6

程序段预处理不会中断。同时使用 M6 启动子换刀程序(L6)。只要“准备换刀”任务一被应答并且在子换刀程序中达到所发出换刀任务的 M 代码，就会在接口上发出任务“执行换刀”（程序段分割）。

对于换刀指令(M206)必须始终先执行一个准备换刀指令。未提前执行“准备换刀”任务的换刀指令不会触发刀具管理任务。

在程序流程中会出现 1:1 换刀的表达方式。其含义是在一个循环中执行换刀。从主轴上移出的刀具（旧刀具）将移动到新刀具的刀位上。不需要进行额外的刀库定位。此时在刀具管理任务中，旧刀具的目标刀位与新刀具的源刀位相同（DB43xx.DBW6 和 DBW8 与 DBW18 和 DBW20 相同）。

1:1 换刀在以下情况下不能进行：

- 刀具的刀位类型不同
- 刀具尺寸不同
- 固定刀位的刀具

在 NC 程序中进行 T0 编程时会触发无新刀具的换刀。只将主轴的刀具传输到刀库中（主轴空运行）。

① ... ⑭ 是下面流程图中的各步骤。

可以从各个步骤中看到不同的应答方式。任务相关的应答以及异步通知都在使用。已编程的传输步骤请参见本章节中的表格 示例： 应答步骤（铣床）（页 348）

“换刀”流程图

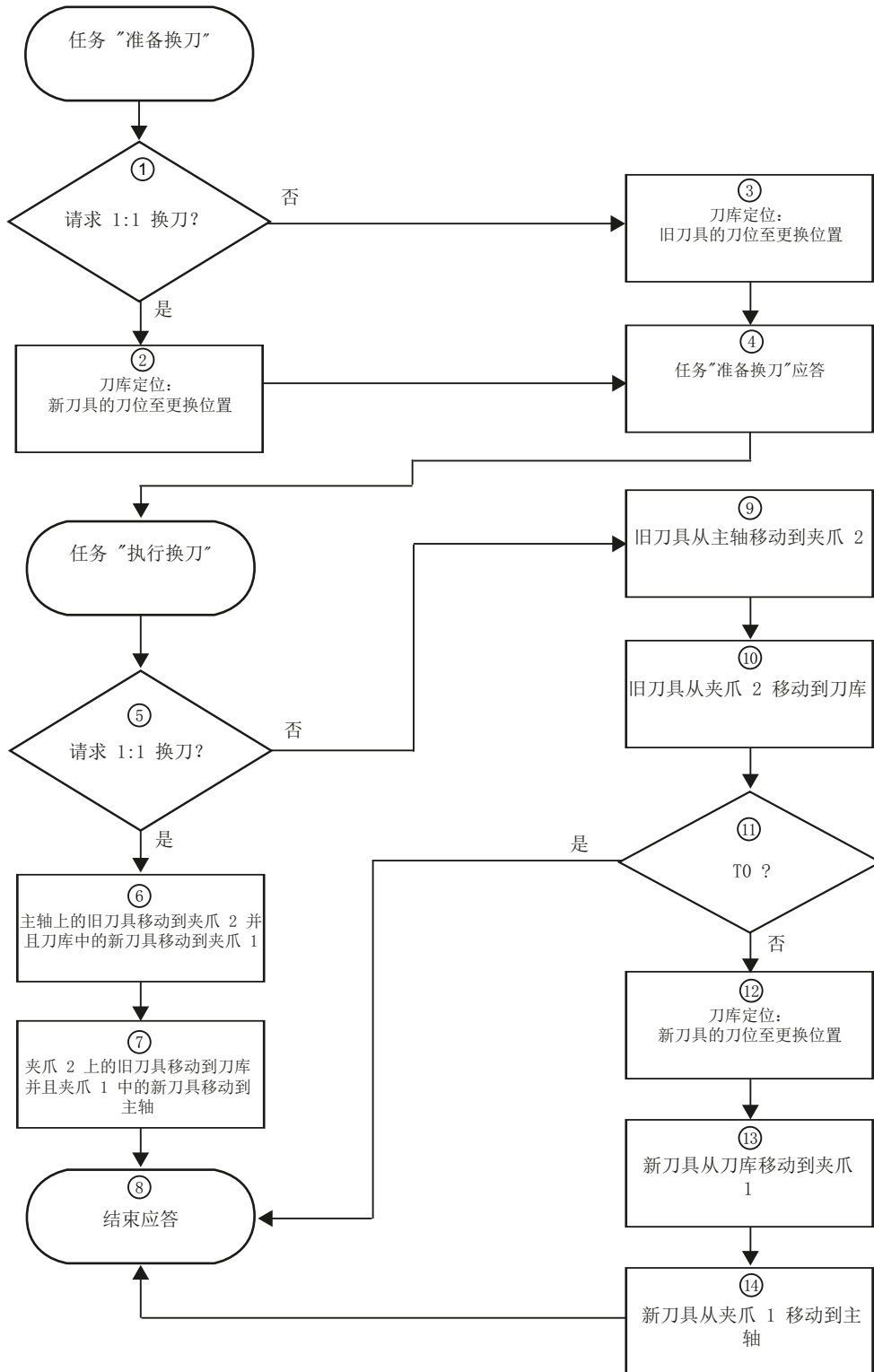


图 9-8 流程图

流程描述:

- NC 程序:
T 指令或带有同步换刀调用的 T 指令(M6)
- 接口信号:
刀具管理任务: DB43xx.DBX0.0 (任务位) 和
DB43xx.DBB1 (指令位): 准备换刀 (DB43xx.DBX1.2)
- 要移动刀具的刀库号和刀位号:
DB43xx.DBW6 至 DBW20: 新刀具的源刀位, 旧刀具的目标刀位

步骤 1: 请求 1:1 换刀

询问: 新刀具的源刀位 == 旧刀具的目标刀位

- 通常情况: 刀套中存在一个刀具, 并且要求一个新刀具。两个刀具在刀库列表中的刀位类型和刀具尺寸都相同, 刀具不是固定刀位编码的。
旧刀具会直接换位到新刀具的刀库刀位上 (1:1 换刀)。如果刀套中没有旧刀具 (DB4300.DBX1.4), 也执行同样的流程。此时刀库会将新刀具移动到换刀位。
→ 继续执行步骤 2

询问: 新刀具的源刀位 >< 旧刀具的目标刀位, 特殊情况:

- 刀具管理通过 DB43xx.DBX1.3 通知, 激活的刀具是否应从刀套中移出(T0)。这时刀库将旧刀具 (仍在主轴上) 的存放刀位定位为换刀位。
如果旧刀具无法存放到新刀具的刀位上 (无法进行 1:1 换刀), 旧刀具的刀库刀位也会定位为换刀位。原因可能是刀位类型或刀具尺寸不同, 或者是固定刀位编码的刀具。
这时换刀会分为两步进行。首先将旧刀具存放到刀库中, 接着将新刀具传输到主轴上。
→ 继续执行步骤 3

步骤 2: 刀库将新刀具的刀位定位为换刀位。

← 上一步: 第 1 步

刀库移动可以通过由 PLC 控制的 NC 回转轴来进行。移动应报告给刀具管理。刀库的位置因此会在操作界面上的刀具和刀库列表中更新。 发送给刀具管理的应答: 从 DB4300.DBW8 (新刀具的刀位号 - 源) 中读取目标位置并写入到 DB9901.DBW2 (传输步骤 101) 中。当目标位置的刀库重合时, 步骤会被异步应答:				
应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态

9.8 铣床应用示例

4	DB4200.DBX0. 4	101	0	204
刀库 1 中的可变刀位在与主轴的切换点上。				
传输步骤	从		向	
	刀库	刀位	刀库	刀位
101 DB9901.DBW0	1	n	9998	1
n: 在此处是 PLC 用户程序实际向可变传输表中输入的刀位号(n ≠ 0)。				

→ 继续执行步骤 4

步骤 3: 刀库将旧刀具的刀位定位为换刀位

← 上一步: 第 1 步

与步骤 2 类似，但是这里的刀库目标刀位从 DB4300.DBW20 中获取（旧刀具的刀位号 - 目标）。				
发送给刀具管理的应答：				
应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
4	DB4200.DBX0. 4	101	0	204
刀库 1 中的可变刀位在与主轴的切换点上。				
传输步骤	从		向	
	刀库	刀位	刀库	刀位
101 DB9901.DBW0	1	n	9998	1

n: 在此处是 PLC 用户程序实际向可变传输表中输入的刀位号(n ≠ 0)。

→ 继续执行步骤 4

步骤 4：准备和应答换刀任务

← 上一步：第 2 步或第 3 步

发送给刀具管理的应答： <ul style="list-style-type: none"> • NC 程序： M206 触发执行换刀任务 • 接口信号： 刀具管理任务：DB43xx.DBX0.0（任务位） DB43xx.DBB1（指令位）：执行换刀（DB43xx.DBX1.1） • 要移动刀具的刀库号和刀位号： DB43xx.DBW6 至 DBW20：新刀具的源刀位，旧刀具的目标刀位 				
应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
1	DB4200.DBX0.1	0	0	1
在很多系统上这时换刀准备已完成。				

→ 继续执行步骤 5

步骤 5：请求 1:1 换刀

← 上一步：无上一步骤，“执行刀具管理”任务步骤序列的进入点

进行与步骤 1 类似的区分，是执行直接换刀还是分两步换刀：

- 可以 1:1 换刀：→ 继续执行步骤 6
- 不可以 1:1 换刀：→ 继续执行步骤 9

步骤 6：主轴上的旧刀具移动到夹爪 2 并且刀库中的新刀具移动到夹爪 1

← 上一步：第 5 步

如果机械移动完成并已在 PLC 用户程序中应答，该应答会发送给刀具管理。 发送给刀具管理的应答：				
应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
5	DB4200.DBX0.5	1	2	105
新刀具从刀库移动到夹爪 1 → 步骤 6 或 13				

9.8 铣床应用示例

传输步骤	从		向	
	刀库	刀位	刀库	刀位
1 DB9900.DBW0	0	1	9998	2
旧刀具从主轴移动到夹爪 2 → 步骤 6 或 9				
传输步骤	从		向	
	刀库	刀位	刀库	刀位
2 DB9900.DBW8	9998	1	9998	3

→ 继续执行步骤 7

步骤 7: 夹爪 2 上的旧刀具移动到刀库并且夹爪 1 上的新刀具移动到主轴

← 上一步: 第 6 步

如果机械移动完成并已在 PLC 用户程序中应答, 该应答会发送给刀具管理。 发送给刀具管理的应答:				
应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
7	DB4200.DBX0. 7	3	4	105
新刀具从夹爪 1 移动到主轴 → 步骤 7 或 14				
传输步骤	从		向	
	刀库	刀位	刀库	刀位
3 DB9900.DBW1 6	9998	2	9998	1
旧刀具从夹爪 2 移动到刀库 → 步骤 7 或 10				
传输步骤	从		向	
	刀库	刀位	刀库	刀位
4 DB9900.DBW2 4	9998	3	0	2

→ 继续执行步骤 8

步骤 8： 结束应答

← 上一步： 第 7 步或第 14 步

刀具切换到初始位置或进入机床可继续加工的状态，就会发出结束应答。在能够结束换刀之前，可能此处仍然会有要执行的机械移动。

发送给刀具管理的应答：

应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
1	DB4200.DBX0.1	0	0	1

→ 继续执行步骤 9

步骤 9： 旧刀具从主轴移动到夹爪 2

← 上一步： 第 5 步

如果机械移动完成并已在 PLC 用户程序中应答，该应答会发送给刀具管理。

发送给刀具管理的应答：

应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
8	DB4200.DBX1.0	0	2	105

旧刀具从主轴移动到夹爪 2 → 步骤 6 或 9

传输步骤	从		向	
	刀库	刀位	刀库	刀位
2 DB9900.DBW8	9998	1	9998	3

→ 继续执行步骤 10

步骤 10: 旧刀具从夹爪 2 移动到刀库

← 上一步: 第 9 步

如果机械移动完成并已在 PLC 用户程序中应答, 该应答会发送给刀具管理。 发送给刀具管理的应答:				
应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
9	DB4200.DBX1.1	0	4	105
旧刀具从夹爪 2 移动到刀库 → 步骤 7 或 10				
传输步骤	从		向	
	刀库	刀位	刀库	刀位
4 DB9900.DBW2 4	9998	3	0	2

→ 继续执行步骤 11

步骤 11: T0 ?

← 上一步: 第 10 步

询问: 换刀任务中设置 T0 了吗? → DB43xx.DBX1.3

- 若只需将刀套空运行, 则可结束换刀。 → 继续执行步骤 8
- 需要将新刀具传送到刀架上吗? → 继续执行步骤 12

步骤 12: 刀库将新刀具的刀位定位为换刀位

← 上一步: 第 11 步

流程同步骤 2 发送给刀具管理的应答:				
应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
4	DB4200.DBX0.4	101	0	204
刀库 1 中的可变刀位在与主轴的切换点上。				

传输步骤	从		向	
	刀库	刀位	刀库	刀位
101 DB9901.DBW0	1	n	9998	1

n: 在此处是 PLC 用户程序实际向可变传输表中输入的刀位号(n ≠ 0)。

→ 继续执行步骤 13

步骤 13: 新刀具从刀库移动到夹爪 1

← 上一步: 第 12 步

PLC 用户程序控制机床功能，使用这些功能进行夹爪移动、刀具夹紧等。 发送给刀具管理的应答：				
应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态
10	DB4200.DBX1. 2	1	0	105
新刀具从刀库移动到夹爪 1 → 步骤 6 或 13				
传输步骤	从		向	
	刀库	刀位	刀库	刀位
1 DB9900.DBW0	0	1	9998	2

→ 继续执行步骤 14

步骤 14: 新刀具从夹爪 1 到主轴

← 上一步: 第 13 步

PLC 用户程序控制机床功能，使用这些功能进行夹爪移动、刀具夹紧等。如果机械移动完成并已在 PLC 用户程序中应答，该应答会发送给刀具管理。 可以结束换刀。 发送给刀具管理的应答：				
应答步骤	应答位	新刀具的传输步骤	旧刀具的传输步骤	状态

9.8 铣床应用示例

11	DB4200.DBX1. 3	3	0	105
新刀具从夹爪 1 移动到主轴 → 步骤 7 或 14				
传输步骤	从		向	
	刀库	刀位	刀库	刀位
3 DB9900.DBW1 6	9998	2	9998	1

→ 继续执行步骤 8

9.8.3 示例：应答步骤（铣床）

固定传输步骤表

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
1 DB9900.DBW0	0	1	9998	2	新刀具从刀库移动到夹爪 1 步骤 ⑥ 或 ⑬
2 DB9900.DBW8	9998		9998	3	步骤 ⑥ 或 ⑨
3 DB9900.DBW16	9998		9998	1	步骤 ⑦ 或 ⑩
4 DB9900.DBW24	9998	3	0	2	步骤 ⑦ 或 ⑭
5 DB9900.DBW32	0	2	9998	1	旧刀具的存放刀位
6 DB9900.DBW40	0	1	9998	1	带有新刀具的刀库刀位到换刀位 步骤 ② 或 ⑫
7 DB9900.DBW48	--	--	--	--	

可变传输步骤表

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
101 DB9901.DBW0	1	n	9998	1	刀库 1 中的可变刀位在与主轴的切换点上。
102 DB9901.DBW8	--	--	--	--	

n: 在此处是 PLC 用户程序实际向可变传输表中输入的刀位号($n \neq 0$)。

应答步骤表

应答步骤	传输步骤		应答状态	注释
	旧刀具	新刀具		
1 DB9902.DBW0	0	0	1	结束应答 步骤 ④ 和 ③
2 DB9902.DBW4	0	0	3	取消任务
3 DB9902.DBW8	0	0	105	下面任务的中间应答, 步骤 ④
4 DB9902.DBW12	101	0	204	可变刀库刀位到换刀位
5 DB9902.DBW16	1	2	105	中间应答 步骤 ⑥
6 DB9902.DBW20	0	5	105	中间应答 步骤 ③
7 DB9902.DBW24	3	4	105	中间应答 步骤 ⑦
8 DB9902.DBW28	0	2	105	中间应答 步骤 ⑨
9 DB9902.DBW32	0	4	105	中间应答 步骤 ⑩

9.8 铣床应用示例

应答步骤	传输步骤		应答状态	注释
	旧刀具	新刀具		
10 DB9902.DBW36	1	0	105	中间应答 步骤 ⑬
11 DB9902.DBW40	3	0	105	中间应答 步骤 ⑭
12 DB9902.DBW44	--	--	--	

提示： 步骤编号 ① ... ⑭ 参考章节流程图：换刀 (页 338)中的流程图

9.8.4 示例： 铣床的换刀循环

程序示例

```

PROC L6 SAVE DISPL0F
;-----
; 机床制造商换刀循环示例
;-----
DEF INT  _WZ_IN_SP, _WZ_VOR
DEF REAL  _SPP= ...           ; 主轴位置
;
  IF (NOT $P_SEARCH)           ; 若无程序段查找
    _WZ_IN_SP=$TC_MPP6[9998,1] ; 主轴上的刀具
    GETSELT(_WZ_VOR)           ; 之前选择的刀具
;
  IF (_WZ_IN_SP<>_WZ_VOR)      ; 如果是其它刀具
    SPOS=_SPP                  ; 定位主轴
    GO                          ; 回到换刀点
    G75 Z=0
    WAITS(1)
  ENDIF
ELSE
  ENDIF
;
; 换刀： 刀具管理和 PLC
M206
M17
    
```

```
;-  
; ENDE  
;-
```

9.8 铣床应用示例

10.1 备份和存档数据

概述

根据数据类的划分，可以分别为每个数据区和每个数据类创建一个存档。

“系统”数据类是一个特例：这些数据是固定设置的并在首次安装或默认初始化时生效。由于这个原因不要求对系统数据备份，因为该数据类中不含调试时或机床运行时的数据。

注意

系统数据的保护

HMI, NC, PLC 和驱动区域中的所有系统数据和“系统”数据类中的数据内容都不得进行修改。

系统数据无法通过零件程序、子程序或者循环的执行或写入，或者通过读入存档来进行修改。

使用 USB 设备存储用户数据时，数据大小不可超过 4 MB!

存档的标识

每个存档都含有以下标识：

- 数据类： M、I、U
- 控制系统类型： 例如 828D ME
- 时间标记： 存档创建的日期和时间。
- 版本名称： 创建该存档所用软件的版本。
- 系统 CF 卡的序列号。

这些标识使得在读取时可以设置过滤器，定义允许读取哪种类型系统上的哪些存档，根据其数据类和软件版本。

参见

数据级的导入与应用 (页 27)

10.2 存档中会保存哪些数据？

何时备份调试数据？

建议在以下时刻执行数据备份：

- 调试后
- 更改机床专用设置后
- 更换硬件组件后
- 软件升级之前

备份和恢复数据

需要备份和恢复数据时选择“调试”操作区：

- “数据存储”软键，用于整个存储器的内部数据备份。
- 软键“IBN 存档”用于创建调试存档或读取调试存档。

注意
USB 设备 USB 设备不适宜作为永久保存媒介。

数据区

在调试存档中保存了以下数据区：

组件	数据
NC 数据	<ul style="list-style-type: none"> ● 编译循环 ● 标准循环和用户循环 ● 定义和宏 ● 机床数据 ● 设定数据 ● 选件数据 ● 全局 (GUD) 和本地 (LUD) 用户数据 ● 刀具和刀库数据 ● 保护区域数据 ● R 参数 ● 零点偏移 ● 补偿数据 ● 零件程序 ● 子程序 ● 工件
PLC 数据	<ul style="list-style-type: none"> ● 主程序(MAIN) ● 数据块(DB) ● 用户程序
驱动数据	二进制格式或 ASCII 格式
HMI 数据	<ul style="list-style-type: none"> ● 文本： 机床制造商的 PLC 报警文本、循环报警文本、零件程序信息文本 ● 模板： 单个模板、工件模板 ● 软件应用程序，比如： 机床制造商的应用程序 ● 设计 ● 配置： 配置，包括显示机床数据 ● 帮助： 在线帮助文件 ● 版本数据 ● 日志： 比如： 操作记录、截屏 ● 程序列表 ● 字典： 简体中文和繁体中文 (IME) ● 数据备份： 通道数据、轴数据等， ASCII 格式 ● 本地驱动器上的程序： 保存在 CF 卡上用户存储区中的程序。

10.2 存档中会保存哪些数据？

说明

存档数据

调试存档会根据数据类 (*.ard) 进行保存。驱动数据作为二进制数据保存，不能进行编辑。

PLC 数据兼容性

SINUMERIK 802D sl 的 PLC 存档可以通过编程工具传输给 SINUMERIK 828D 系列控制系统。

存档的存储区域

存档可以保存在以下存储区域中：

- 在用户 CF 卡上：任意目录。
- CF 卡的目录：
/user/sinumerik/data/archive
或者
/oem/sinumerik/data/archive
- USB 设备上

10.3 仅保存变更的机床数据

前提条件

必须具有“服务”存取级别。

差异-只存储修改的机床数据

通过常用的 MD11210 \$MN_UPLOAD_MD_CHANGES_ONLY 可以设定，在文件中保存全部的机床数据，还是保持仅与缺省设置有偏差的机床数据。

MD11210 \$MN_UPLOAD_MD_CHANGES_ONLY	
只存储修改的机床数据	
= FFH (缺省设置)	
位 0	
...	
位 7	

说明

对调试存档的影响

机床数据 MD11210 \$MN_UPLOAD_MD_CHANGES_ONLY 只作用于“*.arc”型的存档。

创建存档

操作步骤：

1. 将存储设备（CF 卡 或 USB FlashDriv）插到操作系统正面相应的接口中。
2. 点击<Ctrl> + <Alt> + C 键，建立 *.arc 型文件。

结果：

在存储设备上建立了文件：

文件名称构成为：CompleteArchive_Date_Time.arc

示例：

CompleteArchive2010-08-11_08-36-15.arc 表示在 2010 年 8 月 11 日 8:36:15 时生成的文件。

10.4 创建调试存档的步骤

概述

可以选择单个或整个控制系统组件进行备份。建议为每个组件分别创建单独的调试存档，这样就可以单独导入存档。

前提条件

必须具有“服务”存取级别。

创建调试存档

操作步骤：

1. 选择操作区域“调试”。
2. 按下菜单扩展键和软键“调试存档”。“调试”窗口打开。
3. 选择选项“创建调试存档”，按下“确认”。“创建调试存档”窗口打开：

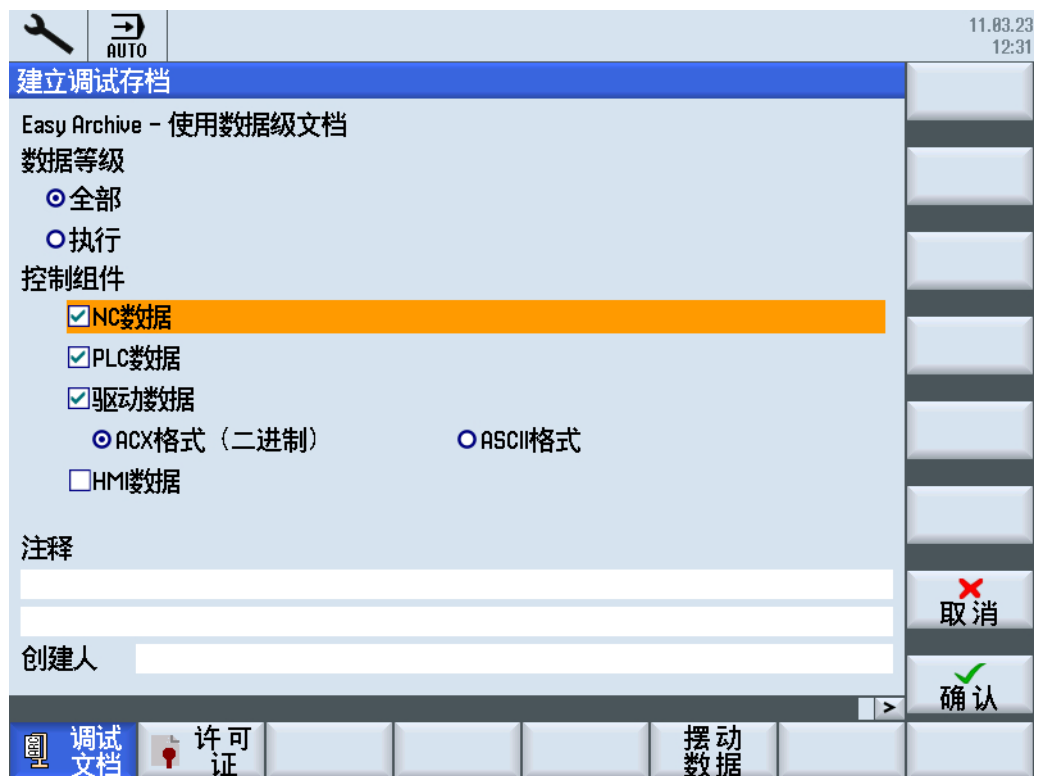


图 10-1 创建调试存档

4. 用 <SELECT> 按键选择, 应“忽略”还是“考虑”数据级:
 - 选择“忽略”, 会对系统组件相关的全部数据进行存档。
 - 选择“考虑”, 则只会将“选择”项下所选择的数据级的数据写入存档。
5. 选中用于存档的系统组件。
6. 可以输入存档的注释和创建人。
7. 按下软键“确认”创建存档。
窗口“创建存档: 选择存储位置”打开。
8. 选择目录或者按下软键“新建目录”, 创建新的子目录。
窗口“新建目录”打开。
9. 输入名称并按下“确认”键确定。
在所选文件夹下创建该目录。
窗口“创建存档: 名称”打开。
10. 输入名称并按下“确认”键确定。
在所选目录下建立存档文件。

10.5 导入调试存档的步骤

前提条件

导入存档必须具有存取级别“用户”。

导入调试存档

操作步骤：

1. 选择操作区域“调试”。
2. 按下菜单扩展键和软键“调试存档”。
“调试”窗口打开。
3. 选择选项“导入调试存档”，按下“确认”。
“选择调试存档”窗口打开。
4. 选择存档并按下“确认”键确定。
5. 按下“确认”应答安全询问，导入存档。
窗口“导入文档”打开，以进度条显示导入进度。
6. 按下软键“取消”，中断导入。

10.6 示例：数据文档“Easy Archive”（Use case）

Easy Archive

带有“Easy Archive”的 SINUMERIK 828D 具有完全不同的数据存档流程。该流程已严格根据批量机床制造的需要进行了裁减。“Easy Archive”基于严格地划分 SINUMERIK 系统软件，OEM 调整数据（机床数据，制造商循环）和操作人员数据（零件程序，刀具补偿）。调整数据还划分为一类在所有类型的机床上都相同的数据，和另一类机床专用的调整数据。

举例进行说明：

用户示例

某一机床制造商建造了一系列垂直加工中心。调整数据会在创建在原型机床上。该原型机床的调整数据组稍后会传输给所有的系列机床（复制）。数据传输后，在每个机床上会进行各自的设置。

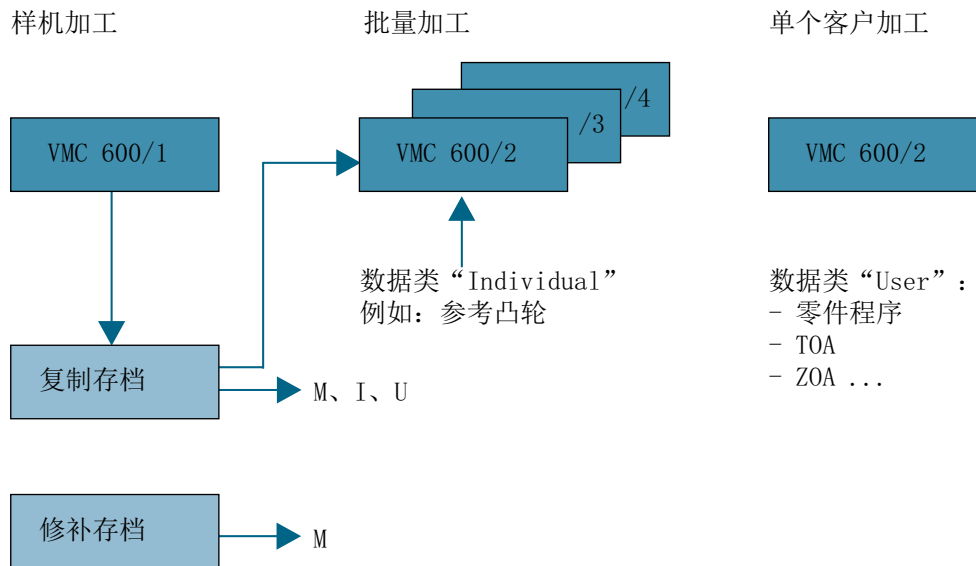


图 10-2 用户示例

例如，测量参考挡块以及滚珠丝杠并作为各自的调整数据输入。如果现在最终用户出错，则错误会在原型机床上再现并消除。如果原型机床的整个文档都传输给相关机床，则该机床独立的调整数据都会被原型机床的独立调整数据所覆盖。

对于 SINUMERIK 828D，机床制造商的调整数据（未进行自定义修改）会单独进行存档。如果该文档传输给相关机床，则可以保证调整数据和最终用户数据保持不变。这样机床制造商的升级过程会大大简化。

10.6 示例：数据文档“Easy Archive” (Use case)

优点

“Easy Archive”的优点是文档设置直接在 SINUMERIK 828D 的操作界面上进行。无需另外用于存档的 PC。

由于将系统数据划分为调整数据和用户数据，SINUMERIK 828D 的系统升级可以完全在 OEM 方面进行，不需要修改自定义数据。系统升级可由最终用户在短时间内自行完成。

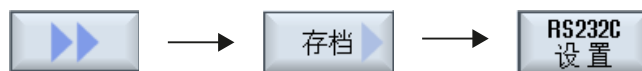
10.7 设置串行接口参数

数据交换

您可以从以下操作区出发设置通过串行接口的数据交换：

- 操作区“程序管理器”
- 操作区“调试” → 软键“系统数据”

按下以下软键，设置接口参数：



说明

如果接口已被占用，例如已经连接了调制解调器，则无法再通过串行接口进行数据交换，系统会输出相应提示。

激活 GSM 调制解调器

使用以下机床数据激活 GSM 调制解调器：

- MD51233 \$MNS_ENABLE_GSM_MODEM = 0 (Voreinstellung)
允许 V24 接口进行数据交换。
- MD51233 \$MNS_ENABLE_GSM_MODEM = 1
该设置激活了 GSM 调制解调器。数据传输相关的软键不在显示。

10.7 设置串行接口参数

附录

A

A.1 文件名中的语种缩写表

支持的语言

系统语言：

语言	文件名中的缩写
简体中文	chs
繁体中文	cht
德语	deu
英语	eng
西班牙语	esp
法语	fra
意大利语	ita
韩语	kor
葡萄牙语（巴西）	ptb

其他语言：

语言	文件名中的缩写
捷克语	csy
丹麦语	dan
芬兰语	fin
匈牙利语	hun
印度尼西亚语	ind
日语	jpn
马来西亚语	msl
荷兰语	nld
波兰语	plk
罗马尼亚语	rom
俄语	rus

A.1 文件名中的语种缩写表

语言	文件名中的缩写
斯洛伐克语	sky
斯洛文尼亚语	slv
瑞典语	sve
泰语	tha
土耳其语	trk
越南语	vit

A.2 缩略语列表

缩写	含义	说明
ALM	Active Line Module	
ASCII	American Standard Code for Information Interchange	美国信息互换标准码
AUTO	运行方式“Automatic”	
BAG	运行方式组	
BTSS	操作面板接口	
BERO	带有反馈振荡的非接触式限位开关	
BICO	Binector Connector	应用在驱动上的互联技术
CEC	交叉误差补偿	
CNC	Computerized Numerical Control	计算机数字控制
dbSI	drive based Safety Integrated	驱动内部集成的安全功能
DB	PLC 中的数据块	
DBB	PLC 中的数据块“字节”	
DBW	PLC 中的数据块“字”	
DBX	PLC 中的数据块“位”	
DDE	Dynamic Data Exchange	动态数据交换
DDS	Drive Data Set	驱动数据组
DIN	Deutsche Industrie Norm	
DO	Drive object	驱动对象
DRAM	Dynamic Random Access Memory	动态存储器
DRF	Differential Resolver Function	微分旋转变压器功能（手轮）
DRY	Dry Run	空运行进给
EDS	Encoder Data Set	编码器数据组
ESR	扩展的停止和退回	
FIFO	First In - First Out	数据如何保存在存储器以及如何重新调用的过程
GUD	Global User Data	全局用户数据
HD	Hard Disk	硬盘
HW	硬件	
HSC	高速切削	

A.2 缩略语列表

缩写	含义	说明
IME	Input Method Editor	亚洲字符输入法
INC	Increment	增量尺寸
INI	Initializing Data	初始化数据
IGBT	Insulated Gate Bipolar Transistor	
IPO	插补器	
IRT	Isochronous Real Time	等时同步通讯
ISO	International Standardization Organisation	国际标准化组织
JOG	运行方式“JOGGING”	使用方向键进行点动
LEC	Leadscrew Error Compensation	丝杠螺距误差补偿
LED	Light Emitting Diode	发光二极管
LUD	Local User Data	本地用户数据
MB	兆字节	
MCP	Machine Control Panel	机床控制面板
MD	机床数据	
MDA	运行方式“Manual Data Automatic”	手动数据输入
MDS	Motor Data Set	电机数据组
MCS	机床坐标系	
MLFB	产品订货号	
MPF	Main Program File	主程序（NC 零件程序）
MAIN	Main program	主程序 (OB1, PLC)
MPI	Multi Point Interface	多端口接口
NCK	Numerical Control Kernel	数字控制中央单元
NCU	Numerical Control Unit	NCK 硬件单元
NPV	零点偏移	
OEM	Original Equipment Manufacturer	
PCU	Programmable Control Unit	
PI	程序等级	
PG	编程器	
PLC	Programmable Logic Control	存储器可编程控制
POE	程序组织单元	在 PLC 用户程序中
PPU	Panel Processing Unit	面板型控制单元

缩写	含义	说明
PZD	驱动的过程数据	
QEC	Quadrant Error Compensation	象限误差补偿
REF POINT	在 JOG 运行方式下“回参考点”	
REPOS	在 JOG 运行方式下“重新定位”	
RPA	R-Parameter Active	NCK 中 R 参数号的存储区
RTC	Real Time Clock	实时时钟
SBL	Single Block	单程序段
SBR	Subroutine	子程序(PLC)
SD	设定数据	
SDB	系统数据块	
SEA	Setting Data Active	设定数据标识（文件类型）
SK	软键	
SLM	Smart Line Module	
SPF	Subprogram file	子程序(NC)
SRAM	Static Random Access Memory	静态存储器
SW	软件	
TEA	Testing Data Aktive	机床数据标识
TO	Tool Offset	刀具补偿
TOA	Tool Offset Active	刀具补偿标识（文件类型）
VPM	Voltage Protection Module	
VSM	Voltage Sensing Module	
WCS	工件坐标系	
WZMG	刀库管理	
WZV	刀具管理	
ZOA	Zero Offset Active	零点偏移数据标识（文件类型）

A.3 关于使用的第三方软件的说明

Copyright 1995 Sun Microsystems, Inc.

Printed in the United States of America.

All Rights Reserved.

This software product (LICENSED PRODUCT), implementing the Object Management Group's "Internet Inter-ORB Protocol", is protected by copyright and is distributed under the following license restricting its use. Portions of LICENSED PRODUCT may be protected by one or more U.S. or foreign patents, or pending applications.

LICENSED PRODUCT is made available for your use provided that you include this license and copyright notice on all media and documentation and the software program in which this product is incorporated in whole or part. You may copy, modify, distribute, or sublicense the LICENSED PRODUCT without charge as part of a product or software program developed by you, so long as you preserve the functionality of interoperating with the Object Management Group's "Internet Inter-ORB Protocol" version one. However, any uses other than the foregoing uses shall require the express written consent of Sun Microsystems, Inc.

The names of Sun Microsystems, Inc. and any of its subsidiaries or affiliates may not be used in advertising or publicity pertaining to distribution of the LICENSED PRODUCT as permitted herein. This license is effective until terminated by Sun for failure to comply with this license. Upon termination, you shall destroy or return all code and documentation for the LICENSED PRODUCT.

LICENSED PRODUCT IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE WARRANTIES OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

LICENSED PRODUCT IS PROVIDED WITH NO SUPPORT AND WITHOUT ANY OBLIGATION ON THE PART OF SUN OR ANY OF ITS SUBSIDIARIES OR AFFILIATES TO ASSIST IN ITS USE, CORRECTION, MODIFICATION OR ENHANCEMENT. SUN OR ANY OF ITS SUBSIDIARIES OR AFFILIATES SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY LICENSED PRODUCT OR ANY PART THEREOF.

IN NO EVENT WILL SUN OR ANY OF ITS SUBSIDIARIES OR AFFILIATES BE LIABLE FOR ANY LOST REVENUE OR PROFITS OR OTHER SPECIAL, INDIRECT AND CONSEQUENTIAL DAMAGES, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

SunOS, SunSoft, Sun, Solaris, Sun Microsystems and the Sun logo are trademarks or registered trademarks of Sun Microsystems, Inc.

SunSoft, Inc.

2550 Garcia Avenue

Mountain View, California 94043

Copyright (c) 1991 by AT&T.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR AT&T MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

This product includes software developed by the University of California, Berkeley and its contributors.

QLocale's data is based on Common Locale Data Repository v1.6.1.

说明

关于使用的第三方软件的详细说明可以参见产品 DVD 光盘中的自述文件 `Readme_OSS`。

索引

\$

\$MC_TOOL_CHANGE_ERROR_MODE
(MD22562), 269, 316
\$MC_TOOL_CHANGE_MODE (MD22550), 267
\$MN_UPLOAD_MD_CHANGES_ONLY
(MD11210) , 357
\$MN_USER_DATA_INT (MD14510), 251
\$MNS_ACCESS_RESET_SERV_PLANNER
(MD51235), 246

A

Access MyMachine /P2P, 11
Advanced Surface, 86

C

CF 卡, 51
CYCLE832, 85

D

DB1800, 237, 238
DB9903, 235
DB9904, 236
DRIVE-CLiQ
 拓扑规则, 156

E

Easy Archive, 361
Easy Extend, 247
 选件位, 251
EE_IFC (DB9905), 249
Ethernet 接口, 20

G

GUD, 30

I

ISO 语言, 84

J

JOG 模式下的测量, 327

M

MD10717
 \$MN_T_NO_FCT_CYCLE_NAME, 274
MD11210
 \$MN_UPLOAD_MD_CHANGES_ONLY, 357
MD14510
 \$MN_USER_DATA_INT[i], 251
MD17300 的预警时间, 95
MD22550
 TOOL_CHANGE_MODE, 267
MD22562
 TOOL_CHANGE_ERROR_MODE, 269, 316
MD51235
 ACCESS_RESET_SERV_PLANNER, 246
MD52270
 TM_FUNCTION_MASK, 273

N

NC 变量, 310

P

PI 服务, 312
PLC 报警, 53
PLC 固件, 265
PLC 用户程序, 265
 调整, 309

R

RTC 电容, 19

T

TCP/IP, 20
TMMVTL (PI 服务) , 312
Toolbox, 11

U

USB 设备, 354

V

V24

接口, 363
取消, 363

报

报警

定义变量, 55
记录, 57
结构, 53
筛选事件, 57

报文类型, 142

备

备用刀具, 263

被

被控对象, 183

编

编程工具 (Programming Tool) , 11

变

变量

T 编号, 311
刀位类型, 310
位置状态, 311

参

参数

运行使能/运行禁止, 122

测

测试许可证, 47

测头, 169

初

初始数据表, 235, 236, 237

传

传输步骤, 305

传输步骤表
固定, 295, 296
可变, 296

串

串行接口, 363

存

存档

存储区域, 356
存取级别, 37

刀

刀具调用, 327

刀具管理, 263
程序段分割, 339

刀具列表, 266

刀库, 266, 305

配置, 329

刀库列表, 266

电

电机数据组直接选择, 177

电源

概述, 114, 138

配置, 133

首次调试的修改, 133

电源模块

保存配置, 137

电网数据, 135

端子布线, 136

概要, 137

定

定义文件, 30

端

端子布局, 166

多

多刀, 268

反

反馈信息, 283

辅

辅助轴

 激活, 152

 配置, 149

高

高速切削 (HSC), 85

更

更换电机模块, 139

公

公司网络, 20

轨

轨迹插补, 183, 197

轨迹优化

 不降低 Kp, 196

 降低 Kp, 196

过

过程数据, 142

宏

宏, 30

互

互联 (对话框), 168

缓

缓冲区, 267

换

换刀, 286, 338, 339

 结束应答, 286

机

机床数据, 81

 单位, 82

 生效条件, 83

激

激活 GSM 调制解调器, 363

结

结束应答, 299, 300

开

开机调试档案文件, 354

 创建, 358

 读入, 360

口

口令

 设置, 39

 修改, 39

链

链式刀库, 267

邻

邻接刀位, 268

模

模拟主轴, 91

摩

摩擦补偿

力矩接通脉冲, 103

示例, 107

速度接通脉冲, 101

其

其他语言, 365

切

切换语言, 41

驱

驱动

BICO 互联, 122

编码器分配, 118, 125

编码器类型选择, 126

参数, 167

端子布线, 122

互联, 164

控制方式/额定值, 121

驱动软件/调试软件 STARTER, 12

驱动设备

配置, 113

拓扑结构, 113

任

任务状态, 294

删

删除语言, 42

设

设定数据, 81

设置日期/时间, 40

示

示例

车床, 320

更换手动刀具 (1), 316

更换手动刀具 (2), 317

零件程序 车削, 321

零件程序 铣削, 334

铣床, 333

手

手动刀具, 269

输

输入/输出(对话框), 166

输入法编辑器 (IME), 42

数

数据备份, 354

数据级, 28

数据区, 354

维护管理器, 235

数据组

编码器 (EDS), 171

电机 (MDS), 171

驱动 (DDS), 171

删除, 179

添加, 172

修改, 181

数控锁功能 (选件), 94

数字量输入/输出端 X122, 161

数字量输入/输出端 X132, 161

拓

拓扑规则 DRIVE-CLiQ, 156

网

网络连接, 23

维

维护管理器
 标准模式, 241
 配置模式, 240
 维护周期, 246

位

位置控制器方案, 190

系

系统语言, 41, 365

信

信息, 异步, 299

许

许可证, 48, 52
 许可证号, 52
 许可证密钥, 48, 52
 许可证书 (CoL), 51

选

选件, 48, 52
 选择方案
 轨迹插补, 195

引

引脚布局
 数字量输入/输出端, 162

应

应答
 WZV, 281
 存取级别, 246
 规则, 309
 过程, 298
 同步, 299
 维护任务, 237, 238, 246
 状态, 300

应答步骤, 307
 应答步骤表, 297
 应答禁止, 239

硬

硬件序列号, 51

用

用户接口, 276
 用户示例, 361
 用户视图, 83

优

优化方案, 187
 优化目标
 快速控制, 191
 稳定控制, 191
 正常控制, 191

语

语种缩写, 365

圆

圆度测试
 保存参数, 229
 保存图形, 230
 设置参数, 225
 位置设定值滤波器的示例: , 226
 优化示例 1, 227
 优化示例 2, 228
 优化示例 3, 228
 执行测量, 225

直

直接连接, 23

制

制造商 PLC 报警, 53

中

中间应答, 299, 300

轴

轴方案, 187

转

转速控制器方案, 189

转塔刀库, 268

装

装载刀库, 267

字

字典

编辑, 43

导入, 43

自

自动伺服优化, 186

总

总应答, 300

SIEMENS

SINUMERIK

SINUMERIK 828D SINUMERIK Operate

开机调试手册

通用设置	1
机床数据和设定数据	2
连接驱动器	3
配置网络	4
调整操作区域“加工”	5
模拟和同步记录	6
配置循环	7
主轴功能	8
刀具管理	9
诊断与维修	10
附录	A




适用于:

数控软件 V4.7 SP1

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。

目录

1	通用设置	11
1.1	访问等级	11
1.1.1	定义存取权限	11
1.1.2	编辑存取权限的口令	11
1.1.3	程序的访问等级	13
1.1.4	文件的访问权限	15
1.1.5	软键的访问等级	16
1.1.6	确定软键的访问等级的步骤	17
1.2	设置日期/时间	19
1.3	切换语言	21
1.4	设计键盘特性	22
1.5	添加用户自定义引导启动画面	23
1.6	生成屏幕拷贝	23
2	机床数据和设定数据	25
2.1	用户视图	25
2.1.1	创建用户视图	26
2.1.2	加工用户视图	27
2.1.3	删除用户视图	29
2.2	机床数据和设定数据的明码文本	29
3	连接驱动器	33
3.1	概览	33
3.2	设置驱动器	33
3.3	设置 EXTCALL	33
3.4	设置 EES	34
3.5	设置全局零件程序存储器	35
3.6	程序选择	36
3.6.1	显示软键	36
4	配置网络	39
4.1	显示网络一览	39
4.2	公司网络设置	40
4.3	保存网络设置	41

4.4	基于子站的网络诊断.....	42
4.4.1	显示 PPU 的网络接口.....	42
4.4.2	TCP/IP 诊断和配置.....	43
4.4.3	以太网诊断.....	46
4.4.3.1	网络诊断和 DHCP 服务器.....	46
4.4.3.2	可访问节点.....	47
5	调整操作区域“加工”.....	51
5.1	设置实际值显示的字体大小.....	51
5.2	插入用户专用商标.....	51
5.3	设置 G 代码组的显示.....	52
5.4	配置通道运行信息.....	54
5.5	取消程序测试.....	58
5.6	激活“示教”功能.....	59
5.7	程序段搜索.....	60
5.7.1	激活程序段搜索模式.....	60
5.7.2	由外部处理的加速程序段搜索过程.....	61
5.8	手动加工.....	62
5.9	多通道支持.....	64
6	模拟和同步记录.....	67
6.1	模拟概述.....	67
6.2	设置用于模拟的坐标系.....	69
6.3	同步记录概述.....	72
6.4	毛坯夹具.....	74
7	配置循环.....	79
7.1	激活车削/铣削/钻削/磨削工艺.....	79
7.2	用于钻削的工艺循环.....	84
7.3	制造商循环.....	86
7.3.1	制造商循环.....	86
7.3.2	标准循环 PROG_EVENT.SPF.....	87
7.3.3	用于换刀的制造商循环 CUST_T 和 CUST_M6.....	88
7.3.4	制造商循环 CUST_TECHCYC.SPF.....	89
7.3.5	用户循环 CUST_MULTICHAN.....	93
7.3.6	通过 programSYNC 调试多通道机床.....	93
7.4	铣削.....	101
7.4.1	常规.....	101
7.4.2	用于铣削的工艺循环.....	102

7.4.3	设置用于铣削的 ShopMill 循环.....	103
7.4.4	柱面转换 (TRACYL)	106
7.4.5	示例: 带有轴配置 XYZ-AC 的铣床.....	107
7.4.6	用于多工件加工的 ShopMill 循环.....	111
7.5	车削.....	114
7.5.1	常规.....	114
7.5.2	设置用于车削的循环.....	114
7.5.3	设置用于车削的 ShopTurn 循环.....	119
7.5.4	设置副主轴.....	121
7.5.5	设置 ShopTurn 中的副主轴.....	124
7.5.6	用于车削的工艺循环.....	126
7.5.7	车床的轴配置.....	130
7.5.8	柱面转换 (TRACYL)	132
7.5.9	端面加工 (TRANSMIT)	136
7.5.10	斜置 Y 轴 (TRAANG)	139
7.5.11	X 零点不位于旋转中心.....	143
7.6	磨削.....	145
7.7	回转.....	147
7.7.1	用于回应的工艺循环.....	147
7.7.2	用于机床运动识别的检查列表 CYCLE800.....	155
7.7.3	调试回转数据 (运动链)	156
7.7.4	回转开机调试机床运动的示例.....	165
7.7.5	Herstellerzyklus CUST_800.SPF.....	175
7.7.6	CYCLE996 工件 - 运动测量.....	182
7.8	高速切削 (HSC).....	184
7.8.1	配置高速设定 CYCLE832.....	184
7.8.2	制造商循环 CUST_832.SPF.....	186
7.9	测量循环和测量功能.....	186
7.9.1	测量循环和测量功能, 通用.....	186
7.9.2	制造商与用户循环 CUST_MEACYC.SPF.....	190
7.9.3	在 JOG 运行方式下测量.....	192
7.9.3.1	在铣床上测量工件.....	195
7.9.3.2	在铣床上测量刀具.....	197
7.9.3.3	在车床上测量刀具.....	202
7.9.3.4	在 JOG 方式中不用电子探头进行测量.....	203
7.9.4	在 AUTOMATIK 运行方式中测量.....	204
7.9.4.1	测量工件, 概述	206
7.9.4.2	在铣床上测量工件.....	209
7.9.4.3	在铣床上测量刀具.....	210
7.9.4.4	在车床上测量工件.....	218
7.9.4.5	在车床上测量刀具.....	219
7.10	比较循环版本.....	220

7.10.1	显示循环版本.....	220
7.10.2	指定循环版本.....	221
7.11	循环保护（选项）.....	223
7.11.1	循环保护概述.....	223
8	主轴功能.....	227
8.1	主轴控制.....	227
8.2	模拟主轴.....	229
8.3	主轴丝杆螺距误差补偿.....	229
8.4	主轴诊断.....	231
8.4.1	主轴诊断.....	231
8.4.2	温度.....	233
8.4.3	电机温度传感器.....	235
8.4.4	附加温度传感器.....	235
8.4.5	温度柱形图.....	236
8.4.6	转速/转矩.....	236
8.4.7	夹紧系统.....	237
8.4.8	夹紧系统：转速限值.....	239
8.4.9	夹紧系统：诊断统计.....	240
8.4.10	夹紧系统：夹紧时间统计.....	241
8.4.11	查看主轴的物流数据.....	241
9	刀具管理.....	243
9.1	用于刀具管理的机床数据.....	243
9.1.1	有/无刀库管理的设置.....	243
9.1.2	配置刀具管理的访问级别.....	245
9.1.3	其他设置.....	247
9.2	配置操作界面.....	251
9.2.1	一般设置.....	253
9.2.2	配置刀具列表.....	259
9.2.2.1	刀具参数标识.....	264
9.2.2.2	刀沿参数的标识.....	267
9.2.2.3	监控参数的标识.....	269
9.2.2.4	磨削参数标志.....	270
9.2.2.5	刀位参数的标识.....	271
9.2.2.6	刀库位置适配器参数的标记.....	272
9.2.2.7	多刀参数的标识.....	273
9.2.2.8	多刀刀位参数的标识.....	275
9.2.3	配置列表参数.....	275
9.2.4	刀具类型列表.....	280
9.2.5	配置刀具类型.....	283
9.2.6	配置“其他数据”窗口.....	285
9.2.7	配置窗口“新刀具 - 收藏”.....	286

9.2.8	配置窗口“新建刀具”	286
9.2.9	配置新刀具的默认值.....	287
9.2.10	配置“详细信息”窗口.....	292
9.2.11	TO 单元和刀套至通道的分配.....	293
9.2.12	命名刀库和刀库位置.....	296
9.2.13	设定刀库刀位类型的名称.....	298
9.2.14	确定刀库和通道的对应关系.....	299
9.2.15	冷却剂和刀具专用功能.....	301
9.2.16	重新激活时的换刀原因.....	305
9.2.17	配置载码器连接.....	309
9.3	创建 OEM 文本.....	312
9.3.1	创建 OEM 文本.....	312
9.3.2	标准文本的标识.....	313
9.3.3	OEM 文本举例.....	316
9.4	示例.....	318
9.4.1	示例：配置 OEM 刀具列表.....	318
9.4.1.1	修改配置文件.....	318
9.4.1.2	修改用户文本文件.....	321
9.4.2	示例：配置刀库位置类型及名称.....	323
9.4.2.1	修改配置文件.....	323
9.4.2.2	修改用户文本文件.....	325
9.5	替换 T、M 或 D 功能.....	327
9.6	双刀架加工.....	331
9.7	使用多刀.....	334
9.7.1	激活多刀显示.....	334
9.7.2	装卸多刀.....	335
9.7.3	多刀中的换刀.....	337
9.8	刀具和刀库管理编辑器.....	340
9.8.1	概述.....	340
9.8.2	调用刀具管理.....	341
9.8.3	编辑刀具管理.....	341
9.8.4	设置刀具管理参数.....	344
9.8.4.1	创建刀具单元.....	344
9.8.4.2	定义缓存位置.....	345
9.8.4.3	定义装载位置.....	345
9.8.4.4	设置刀库配置.....	346
9.8.4.5	链接分配.....	348
10	诊断与维修.....	351
10.1	NC/PLC 变量.....	351
10.1.1	显示和编辑 PLC 变量和 NC 变量.....	351
10.1.2	保存和加载屏幕.....	355

10.2	显示维修概览.....	355
10.2.1	选择轴和驱动.....	357
10.2.2	轴诊断.....	359
10.2.3	驱动信息.....	365
10.3	PROFINET 诊断.....	373
10.3.1	显示 PROFINET 配置.....	373
10.3.2	显示 PROFINET 设备的详细信息.....	375
10.4	跟踪.....	377
10.4.1	一般步骤.....	378
10.4.2	跟踪会话.....	379
10.4.2.1	创建跟踪会话文件.....	379
10.4.2.2	保存跟踪会话文件.....	379
10.4.2.3	加载跟踪会话文件.....	381
10.4.3	用于跟踪的变量.....	382
10.4.3.1	筛选/搜索变量.....	382
10.4.3.2	选择变量的属性.....	384
10.4.3.3	显示变量的详细信息.....	387
10.4.4	跟踪设置.....	388
10.4.4.1	跟踪设置 (PLC、NC、伺服)	388
10.4.4.2	跟踪设置 (驱动)	389
10.4.4.3	跟踪的选项.....	390
10.4.4.4	启动跟踪.....	390
10.4.5	分析跟踪.....	391
10.4.5.1	设置跟踪的视图.....	391
10.4.5.2	选择变量.....	393
10.4.5.3	设置曲线图的比例.....	393
10.4.5.4	缩放曲线图.....	394
10.4.5.5	定位光标.....	396
10.4.5.6	获取测量值.....	396
10.5	驱动系统.....	397
10.5.1	显示驱动状态.....	397
10.5.2	显示驱动对象的详细信息.....	398
10.6	操作日志.....	400
10.6.1	设置操作日志.....	400
10.6.2	显示记录文件	402
10.6.3	在记录文件中搜索.....	402
10.6.4	保存记录.....	403
10.6.5	日志文件的结构.....	404
10.6.6	扩展设置.....	407
10.7	HMI 跟踪.....	408

A	附录.....	411
A.1	关于使用的第三方软件的说明.....	411
A.2	缩略语.....	413
	索引.....	417

通用设置

1.1 访问等级

1.1.1 定义存取权限

针对不同用户对程序、数据和功能的访问需求，系统将访问权限分为 7 个等级。它们分为

- 3 个口令，供制造商、服务人员和用户使用
- 4 个钥匙开关位置，供用户使用

下表列出了访问级别 1 到 7，其中

- 1 表示最高等级而
- 7 表示最低等级。

表格 1-1 访问级别的设计方案

访问级别	保护方式	区域
1	口令：SUNRISE（缺省口令）	制造商
2	口令：EVENING（缺省口令）	服务
3	口令：CUSTOMER（缺省口令）	用户
4	钥匙开关位置 3	程序员，调试员
5	钥匙开关位置 2	具有专业资质的操作员
6	钥匙开关位置 1	受过培训的操作员
7	钥匙开关位置 0	掌握基本内容的操作员

1.1.2 编辑存取权限的口令

一览

您可以在操作界面上激活受口令保护的访问级别。

1.1 访问等级

可以选择下列方式编辑口令：

- 设置新口令
- 修改口令
- 删除口令

设定口令



1. 选择操作区域“调试”。
2. 按下软键“口令”。
3. 按下软键“设定口令”。
“设定口令”窗口打开。
4. 输入一个允许的默认口令，按下软键“确认”结束输入。
有效的口令会成功设置，屏幕上显示有效的访问级别。而无效的口令会被拒绝。

修改口令

您可以修改默认口令，提高访问安全性。



1. 按下软键“修改口令”。
“修改口令”窗口打开。
2. 选中需要设置新口令的区域（如用户）。
3. 在输入栏“新口令”和“重复口令”中输入新口令。
4. 按下软键“确认”，结束输入。
只有当两次的输入一致时，新的有效口令才会被接收。

删除口令



按下软键“删除口令”。访问权限再次被取消。

说明

上电不会自动删除访问权限。

在执行 NC 清零时，口令恢复为缺省口令。

1.1.3 程序的访问等级

目标

设置访问权限一方面可以确保信息安全，另一方面可以防止控制系统或机床被非法操作或被蓄意损坏。

在文件系统的某个文件和目录上设置访问权限，便可以控制各种用户对该文件或目录的操作。这里的操作指：

- 读文件内容(READ)，复制文件或生成文件存档。
- 修改文件内容(WRITE)，或删除文件。
- 执行文件(EXECUTE)。
- 列出并显示某个目录中的某个文件(SHOW)。
- 在某个目录内新建文件。
- 显示或读取某个目录的内容。
- 新建或删除某个目录。

功能说明

上述单项操作的访问权限的设置针对 NCK 文件系统中的以下目录生效：

- 零件程序
- 子程序
- OEM 循环目录
- 用户循环目录
- 多工件目录
- 工件目录

访问权限的设置还针对其中的主程序文件(*.mpf)、子程序文件或循环文件(*.spf)生效。

1.1 访问等级

另外，访问权限还可以扩大到本地驱动器上的以下目录：

- 零件程序
- 子程序
- 多工件目录
- 工件目录

说明

远程访问时的访问权限

在远程访问控制系统时，例如通过 WinSCP 或者 RCS Commander 访问，在新建或者编辑文件和目录方面享有和直接在控制系统上访问一样的权限。

通过机床数据设置访问权限

可以用下面的机床数据来**统一 设置访问权限**：

MD51050 \$MNS_ACCESS_WRITE_PROGRAM

默认设置为 4（即访问级别为用户）。用户）

说明

设置冲突时访问权限的效力

如果不仅设置了单项操作的访问权限，还通过“MD51050 \$MNS_ACCESS_WRITE_PROGRAM”设置了统一访问权限，一旦两者冲突，则其中限制性更强的设置生效。

文件 slfsfileattributes.ini 中的设置

文件 slfsfileattributes.ini 是一份模板文件，位于以下目录中：

../siemens/sinumerik/hmi/template/cfg

值 0 至 7 对应访问级别，例如密码也适用。4 位数的数字表示根据以下权限的顺序：

从左至右的含义：R W X S		
R	READ	读文件和目录
W	WRITE	修改、删除或者创建文件和目录
X	EXECUTE	执行程序。指文件的访问权限，如果是目录，则始终为 7。
S	SHOW	输出文件和目录的列表

将文件 `slfsfileattributes.ini` 复制到以下一个目录中：

`../oem/sinumerik/hmi/cfg` 或者 `../user/sinumerik/hmi/cfg`

重新启动操作软件后，该文件修改才能生效。

示例

路径由数据类型命名符组成。在下面的例子中，“`\wks.dir*.wpd*.mpf`”是位于工件目录“`wks.dir`”下工件“`*.wpd`”的零件程序“`*.mpf`”。

所有工件的新建零件程序的默认访问权限为 `7777`。

所有条目都位于段落[ACCESSMASKS]中，格式如下：

```
[ACCESSMASKS]
```

```
\wks.dir\*.wpd\*.mpf = 7777
```

```
\mpf.dir\*.mpf = 6577
```

```
\cus.dir\*.spf = 6577
```

参见

定义存取权限 (页 11)

1.1.4 文件的访问权限

修改访问权限

在“属性”窗口中显示了执行、写入、列明和读取文件的权限。

- 执行：选择文件加以执行。
- 写入：设置修改/删除目录/文件的权限。

可为每个 **NC** 文件单独设置访问权限：从钥匙开关 **0** 到当前的访问权限。

如果文件/目录所需的访问权限高于您当前的访问权限，则无法进行设置。

1.1 访问等级

对于外部文件而言，比如本地驱动器上的文件，此处只向您显示机床制造商为这些文件设置的访问权限。您无法在属性窗口中进行任何修改。

说明

设置目录和文件的访问权限

通过配置文件和 MD51050 您可以对 NC 存储器和用户存储器（本地驱动器）上的目录和文件的访问权限进行修改和预设。

配置文件“slfsfileattributes.ini”

在 NC 中创建文件时，配置文件“slfsfileattributes.ini”和 MD51050 中设置的访问权限开始起作用，您可以随后在程序管理器的功能“属性”中单独修改。在 NC 中，访问权限是每个文件的单独属性。

在访问 NCU CF 卡中的外部数据或者访问 PCU/PC 的硬盘时，配置文件“slfsfileattributes.ini”和 MD51050 中设置的访问权限对所有指定目录和文件起作用，访问权限通过操作无法加以修改。外部数据的访问权限只能统一设置，不能单独设置。

1.1.5 软键的访问等级

使用

软键的显示和操作不仅可以由 OEM 隐藏，也可以由用户隐藏，以便使操作软件更加符合实际功能范围的需要，使操作界面更加清晰简单。

为此需要对控制系统的功能范围加以限制，以取消对操作软件某些功能的访问，降低误操作几率。

访问权限的等级

只有当前访问等级能够查看并操作的软键才可以被重新设置访问等级。

示例：一个软键只能由访问等级“制造商”访问，因此它的访问等级不能被“钥匙开关 3”修改。

前提条件

为了确保系统的稳定运行和可操作性，有些软键的访问等级是禁止被修改的，例如：垂直软键条的软键“<<”，或软键“调整软键”本身，以便可以再次恢复软键显示/操作。

另外，以下软键也禁止被隐藏或修改访问等级：

- 所有“确认”软键
- 所有“取消”软键
- 所有“接收”软键
- 所有“<<”和“>>”软键

循环窗口中软键的访问等级也可以被修改，但“Easy Screen”窗口中的软键不可以。

说明

软件升级

由于软键访问等级的所有改动都通过软键 ID 加以指明，因此即使在可能引起软键更新的软件升级后，也仍然保持有效。

对于在访问时由机床数据如：MD51045 \$MNS_ACCESS_TEACH_IN 加以保护的软键，在设置新的访问等级后，就只由新的访问等级保护。系统不会再考虑该机床数据。

所有通过这种方法隐藏的软键都可以暂时通过系统默认的访问权限显示，按下软键“显示所有软键”即可。只要当前还有软键保持旧访问等级，该软键就一直有效。重启系统后，软键访问等级的设置才生效。

1.1.6 确定软键的访问等级的步骤

一般操作顺序

按照以下步骤修改软键的访问等级：

- 激活软键调整模式。
- 修改软键的访问等级。
- 接收并确认修改。

前提条件

对软键“调整软键”的访问受到通用机床数据 MD51073 \$MNS_ACCESS_SET_SOFTKEY_ACCESS 的保护，该机床数据的默认访问等级是 3，也就是说：从口令“维修”开始起才可以访问软键“调整软键”。访问等级为“用户”的操作人员如需访问“调整软键”，必须将该数据设为大于 3 的值。

调整软键的访问等级时，需要使用一个外部鼠标。

修改软键的访问等级

操作步骤：

1. 按下操作区域“调试”中的软键“HMI”。
2. 按下“>>”，按下“调整软键”。
接着会弹出一个对话框，给出详细的操作提示。
1. 按下“确认”，激活“调整软键访问等级”模式。随即显示一张列表。
2. 选择一个操作区域。
3. 将鼠标光标移到一个软键上，点击鼠标右键打开右键菜单“确定访问等级”：
 - 当前显示的访问等级是默认等级。
 - 点击“始终隐藏”，使该软键隐藏。
 - 点击“恢复设置”，再次清除额外指定的访问等级。该选项只有在已经设置了单个访问等级后才会出现。
4. 返回“调试”操作区域，按下软键“结束调整”。现在，便退出了“调整软键访问等级”模式。
按下“确认”确认弹出的询问，接收修改，修改在系统重启后生效。
5. 如果希望取消软键的调整，可以选择一行，然后按下软键“恢复设置”。随即会在“新访问等级”列中加入标识“恢复设置”，结果如同删除新访问等级一样，在系统重启后，该条目消失。”

示例

通过上述步骤修改了以下访问等级：

软键	新访问等级
SISuMainScreenMenuHu::0::3	制造商
SIMaJogMillMenuHU::jogHuMore::6	用户
SISuPaUserDataMenu::::4	维修

在“软键”列中记录了选择软键时软键的标识和位置。

结果

一旦接收了软键调整，系统便会生成一份名为“slsoftkeyaccess.xml”的配置文件，它位于文件系统的以下目录下：../user/sinumerik/hmi/cfg。在/oem下也可以保存一些在系统启动时参与计算的文件。/oem目录的优先级要高于/user目录。

说明**编辑文件“slsoftkeyaccess.xml”的注意事项**

为了避免 xml 句法错误和软键标识错误，只允许通过操作软件按照上文说明的步骤来编辑文件“slsoftkeyaccess.xml”，而不允许直接在文件系统的“系统数据”中编辑文件，否则可能会导致无法编译整个文件或个别软键条目。

1.2 设置日期/时间

您可以手动设置 SINUMERIK 控制器的日期和时间。其中您还可以选择不同的显示格式。

日期显示的格式

格式	示例
d.M.yy	3.1.09
dd.MM.yy	03.01.09
dd.MM.yyyy	03.01.2009
d/M/yy	3/01/09
dd/MM/yy	03/01/09
dd/MM/yyyy	03/01/2009
M/d/yy	1/3/09
MM/dd/yy	01/03/09
MM/dd/yyyy	01/03/2009
yyyy/MM/dd	2009/01/03

这里：

日期	输入说明
d	每个月的一位或者二位数日期 (1 - 31)。
dd	每个月的两位数日期，第一位默认为零 (01 - 31)。
M	每年的一位或者两位数月份 (1 - 12)。
MM	每年的两位数月份，第一位默认为零 (01 - 12)。
yy	两位数年份 (00 - 99)。
yyyy	四位数年份 (比如 2009)。

时间显示的格式

格式	示例	小时制
H:m:s AP	1:2:4 PM	24 小时制，显示“AM”/“PM”
h:m:s ap	1:2:8 pm	12 小时制，显示“am”/“pm”
HH:mm:ss AP	13:02:08 PM	24 小时制，显示“AM”/“PM”
hh:mm:ss ap	01:02:09 pm	12 小时制，显示“am”/“pm”。
H:mm:ss AP	1:02:09 PM	24 小时制，显示“AM”/“PM”
h:mm:ss ap	1:02:09 pm	12 小时制，显示“am”/“pm”
h:mm:ss	13:02:09	24 小时制
hh:mm:ss	13:12:04	24 小时制

这里：

时间符号：	输入说明	
h	用一位或者两位数表示小时（0 - 23 或者 1 - 12 am/pm）。	
hh	用两位数表示小时，第一位默认为零（00 - 23 或者 01 - 12 am/pm）。	
H	用一位或者两位数表示小时（0 - 23 AM/PM）。	
HH	用两位数表示小时，第一位默认为零（00 - 23 AM/PM）。	
m	用一位或者两位数表示分钟（0 - 59）。	
mm	用两位数表示分钟，第一位默认为零（00 - 59）。	
s	用一位或者两位数表示秒数（0 - 59）。	
ss	用两位数表示秒，第一位默认为零（00 - 59）。	
AP	白天时间的显示：	AM 或者 am = 上午 PM 或者 pm = 下午




步骤



1. 选择操作区域“调试”。



2. 按下软键“HMI”。

- | | |
|---|--|
|  | 3. 按下软键“日期时间”。
窗口“设置日期和时间”打开。 |
|  | 4. 按下<SELECT>键，在“格式”栏选择需要的日期和时间的显示格式。 |
|  | 5. 在“新”后面的输入栏输入当前的日期和时间。
6. 按下软键“确认”结束输入。
采用新的日期和时间。 |

1.3 切换语言

标准配置中，操作软件有多种语言。

选择语言

可通过如下显示机床数据确定语言选择模式。

MD9100 \$MM_CHANGE_LANGUAGE_MODE	语言选择模式
= 1	通过窗口“语言选择”确定界面语言（缺省值）。



1. 选择操作区域“调试”。
 2. 按下软键“Change language”。
“语言选择”窗口打开。上次设置的语言被选中。
 3. 请将光标定位在需要的语言上。
 4. 按下 <INPUT> 键。
- 或者-
- 按下软键“确认”。
操作软件将切换到所选语言。

1.4 设计键盘特性

确定两种语言

MD9100 \$MM_CHANGE_LANGUAGE_MODE	语言选择模式
= 2	在两种语言之间进行选择。在运行期间可以通过软键“Change language”切换语言。



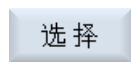
1. 选择操作区域“调试”。



2. 按下软键“HMI”。



3. 按下软键“>>”。



4. 按下软键“语言选择”。
“语言选择”窗口打开。

在“第一种语言”和“第二种语言”栏中将显示全部已安装的语言。

5. 分别选择一种语言。



6. 按下 <INPUT> 键。

-或者-



按下软键“确认”。



在运行中可以在两种语言间切换，通过按下软键“Change language”。

1.4 设计键盘特性

通过 CAPSLOCK（大写锁定）可以锁定外部键盘的小写输入，一直采用大写输入。

按键属性通过显示机床数据设定。

设置

MD9009 \$MM_KEYBOARD_STATE	
= 0	CAPSLOCK 关闭（缺省值）
= 2	CAPSLOCK 打开

在更改机床数据后必须重新启动系统。

1.5 添加用户自定义引导启动画面

OEM 自定义的引导启动画面

西门子引导启动画面可以替换成自定义的引导启动画面。

步骤:

1. 命名自定义引导启动画面“logo.png”。

说明

文件名只能由小写字母构成。

2. 将自定义引导启动画面保存在下面其中一个目录下:

`/user/sinumerik/hmi/ico/ico640` 或
`/oem/sinumerik/hmi/ico/ico640` 或
`/addon/sinumerik/hmi/ico/ico640`

说明

画面的大小没有影响，分辨率始终设为 640x480 像素。

1.6 生成屏幕拷贝

可通过当前操作界面生成截屏。

每个截屏将保存为文件，并存放在如下文件夹中:

`/user/sinumerik/hmi/log/screenshot`

步骤

Ctrl + P 按下快捷键 <Ctrl + P>，

从当前操作界面创建格式为 .png 的截屏。

文件名由系统按升序给定，从“SCR_SAVE_0001.png”到“SCR_SAVE_9999.png”。最多可以创建 9999 幅图。

复制文件



1. 选择操作区域“调试”。



2. 按下软键“系统数据”。

3. 打开上述文件夹并选择必要的截屏。



4. 按下软键“复制”。

-或者-



按下软键“剪切”。



5. 打开需要的存储位置目录，例如在 U 盘上，并按下软键“粘贴”。

机床数据和设定数据

2.1 用户视图

功能的用途

在“用户视图”窗口中，您可以将某个功能的所有相关机床数据综合在一起，并加以调整，这些自定义的机床数据组合可以大大简化您的工作。

您还可以在创建或编辑用户视图时，添加注释。

在“参数”操作区域中显示用户视图

按照以下步骤，同时在“参数”操作区域中显示用户视图：

1. 选择操作区域“调试”。
2. 按下软键“机床数据”。
3. 按下软键“用户视图”。
4. 按下软键“管理视图”。
5. 按下软键“新视图”。
6. 若需要在“参数”→“设定数据”→“数据列表”中一并显示用户视图，则应勾选复选框“在参数/设定数据下也进行显示”。
7. 命名用户视图。
8. 按下“确认”。
9. 添加在“参数”区域显示的机床数据。

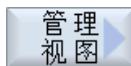
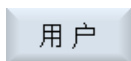
2.1 用户视图

2.1.1 创建用户视图

步骤



1. 选择操作区域“调试”，按下软键“机床数据”。



2. 按下软键“用户视图”，“管理视图”和“新视图”。

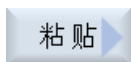
窗口“新视图”打开。



3. 命名用户视图。

按下软键“确认”。

若与已有的用户视图同名，会出现安全消息询问是否覆盖已有的视图。



4. 按下软键“插入数据”。

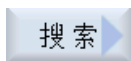
“插入数据”选择窗口打开。



5. 按下<INSERT>键打开下拉菜单，选择所需的机床数据区并按下<INPUT>键。



通过光标键或软键“搜索”，选择需要的机床数据。



6. 按下软键“行前插入”或“行后插入”，将所选择的机床数据加入用户视图中所需的位置上。

提示:

在打开的用户视图中，您可以用鼠标浏览，选择所需的数据行，无需关闭选择窗口。



7. 按下软键“返回”关闭选择窗口，返回到待编辑的用户视图中。
输入值会被自动储存。



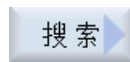
8. 按下软键“插入文本”，可以向打开的用户视图中插入任意文本。
输入窗口“插入文本”打开。



9. 输入所需的文本以及与插入文本相关的注释说明，按下软键“行前插入”或“行后插入”，插入文本。



10. 按下软键“返回”，保存输入内容并返回到待编辑的用户视图中。

选择特定的机床数据

借助搜索对话框可以明确查找到目标数据。




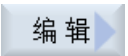





2.1.2 加工用户视图


按下软键“属性”，您可以查看并修改注释、机床数据的来源（通道、轴、驱动设备）以及数据说明，具体情况取决于所选的数据行。


- 机床数据
 - 可以为机床数据选择固定或可变的数据源，视数据类型而定。
 - 选项“可变(*)”表示使用垂直软键 1 - 3 (+、- 和 直接选择) 可以切换相应的数据源（通道、轴、驱动设备），并且显示值会随即变为刚刚选取的数据源。
 - 也可以选择固定的数据源，这样在切换到其他数据源时显示值不会发生变化。
- 注释文本
 - 文本和说明


2.1 用户视图

步骤

- | | |
|--|--|
| 











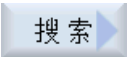
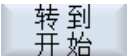
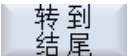


 | <ol style="list-style-type: none"> 1. 选择操作区域“调试”，按下软键“机床数据”。 2. 按下软键“用户视图”。
“用户视图”窗口打开。 3. 按下软键“编辑视图”。编辑用户视图时，还可使用其它的软键。 4. 如果需要添加其他数据，按下软键“插入数据”。 5. 借助光标键选择所需的机床数据。 6. 按下软键“向下”或“向上”来移动所选中的行。

-或者-
按下软键“删除行”，从视图中删除所选中的行。此处不会弹出安全询问，而是直接删除数据。

-或者-
按下软键“属性”，以读取或更改注释文本。
“属性”窗口打开。取决于设置，机床数据的说明或机床数据可能显示在窗口左下方。 7. 按下软键“返回”保存修改。 |
|--|--|

选择特定的机床数据

- | | |
|---|---|
| 

 | <p>借助搜索对话框可以明确查找到目标数据。</p> <p>按下软键“转到开始”，从第一个数据开始搜索。</p> <p>按下软键“转到结尾”，从最后一个数据开始搜索。</p> <p>如果找到的数据不符合您的需要，按下软键“继续搜索”。</p> |
|---|---|

2.1.3 删除用户视图

步骤



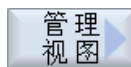
1. 选择操作区域“调试”。



2. 按下软键“机床数据”。



3. 按下软键“用户视图”。



4. 按下软键“管理视图”。

5. 在选择列表中选择所需视图。



6. 按下软键“删除视图”。



或者



随即弹出一条安全询问，要求按下“是”确认，或按下“否”取消操作。

之后会弹出对话框“用户视图”。

还有一种方法是，在以下路径的“系统数据”下修改用户视图：`/user/sinumerik/hmi/template/user_views`。

2.2 机床数据和设定数据的明码文本

对于机床数据和设定数据，可以按照特征[索引]来配置用户专用的、与语言相关的明码文本。

明码文本的显示

可通过以下显示机床数据设置明码文本的显示：

MD9900 \$MM_MD_TEXT_SWITCH	替代机床数据名称的明码文本
= 1	明码文本在表格中显示
= 0	明码文本在对话框中显示

文件

文件	机床数据/设定数据
nctea_xxx.txt	通用机床数据/设定数据
chtea_xxx.txt	通道专用机床数据/设定数据
axtea_xxx.txt	轴专用机床数据/设定数据
hmitea_xxx.txt	显示机床数据

对于“xxx”必须使用规定的语言标识，例如：德语为“deu”，英语为“eng”等
 请确保写入文件名时使用的是小写字母。

步骤

1. 根据需要修改哪种机床数据或设定数据的明码文本，创建相应的文件，例如“nctea_xxx.txt”。
2. 采用 UTF-8 编码保存文件，从而使变音字符和特殊字母能在 SINUMERIK Operate 中正确显示。
3. 在 SINUMERIK Operate 中将文件保存在以下目录：
 System CF-Card/user/sinumerik/hmi/Ing 或
 System CF-Card/oem/sinumerik/hmi/Ing 或
 System CF-Card/addon/sinumerik/hmi/Ing

说明

搜索文件时，会依次搜索“user”、“oem”和“addon”目录。然后对最先搜索到的文件进行分析。例如文本文件位于“user”和“oem”目录中时，会对“user”目录中的文件进行分析。

说明

如果一个机床数据有不同的索引，而明码文本中未指定索引，则对机床数据的所有索引显示相同的明码文本。
 参数和索引必须为升序排列。

使用德语文本的文件示例

```
14510 Mein Integer-Text für PLC_MD 14510
14510[1] Mein Integer-Text für MD 14510[1]
14512[1] Mein Text für HEX-Datum 14512[1]
```

显示的结果:

在操作区域“调试”、“通用 MD”中，明码文本在表格中或在对话框中显示。

机床数据	文本
14510	Mein Integer-Text für PLC_MD 14510
14510[1]	Mein Integer-Text für MD 14510[1]
14512[1]	Mein Text in HEX-Datum 14512[1]

2.2 机床数据和设定数据的明码文本

连接驱动器

3.1 概览

在 HMI 中最多可设置 21 个与所谓的逻辑驱动器（数据存储器）的连接。在操作区域“程序管理器”和“调试”中可以访问该驱动器。

可配置以下逻辑驱动器：

- USB 接口
- CF 卡
- 网络驱动器



软件选件

如需通过以太网管理附加驱动器，需要选件“管理网络驱动器”。

3.2 设置驱动器

操作区域“调试”中有一个“设置驱动器”窗口，用于设置程序管理器中的软键。

操作步骤和配置信息参见下列文档：

参考文档

通用/车削/铣削/磨削操作手册，章节“程序管理”>“设置驱动器”

文件

生成的设置数据保存在文件“logdrive.ini”中。文件保存在目录 /user/sinumerik/hmi/cfg 下。

3.3 设置 EXTCALL

使用指令 EXTCALL 可实现从零件程序中调用外部程序存储器上的程序。

不需要关闭 EXTCALL。

3.4 设置 EES

打开功能

可以通过以下显示机床数据激活或取消 EXTCALL 调用命令的执行。

MD9106 \$MM_SERVE_EXTCALL_PROGRAMS	
= 0	HMI 忽略 EXTCALL 指令并通过 PLC 选择
= 1	HMI 执行 EXTCALL 指令并通过 PLC 选择（缺省值）
= 2	HMI 执行 EXTCALL 指令并忽略通过 PLC 的选择
= 3	HMI 忽略 EXTCALL 指令并执行通过 PLC 的选择

说明

通过外部程序存储器选择主程序的情形下，若上电时该程序存储器继续可用，且 MD9106 中激活了通过 EXTCALL 调用加工，则系统会自动重新选择该程序。

3.4 设置 EES

通过 EES 功能（从外部存储器执行），可直接处理驱动器上已激活的用于 EES 的零件程序，而无需下载缓冲器。该行为是数控零件程序存储器上的执行进程，且没有适用于“外部执行”的限制。



软件选项

使用该功能需要用到软件选项
“从外部存储器执行（EES）”。

功能 EES 激活时可在系统 CF 卡上启动一个用户内存（100 MB）。因此，即使没有 PPU 的用户 CF 卡也可使用 EES 功能。

说明

有了 EES 功能，EXTCALL 调用则是多余的。现有的 EXTCALL 调用可以在需要时切换。未切换的 EXTCALL 调用也可在激活 EES 时执行，但仅作为 EXTCALL 执行。

诊断

如需查看当前的 EES 设置，可通过以下设置显示 EES 模式：

MD18045 \$MN_EES_MODE_INFO	EES 功能在此模式下工作。
位 0 = 1	本地 EES 激活（PPU 的本地驱动）

位 1 = 1	全球 EES 激活（所有可用的驱动器（USB，包括本地 EES））
位 2 = 1	在外部存储器上设置了全局零件程序存储器。 在搜索路径中，首先在 NC 的被动文件系统搜索程序，然后在外部存储器的相应目录下搜索。

EES 功能的详细说明请参见以下文档：

参考文档

功能手册之基本功能分册： BAG、通道、程序运行、复位特性 (K1)

章节“EES（Execution from External Storage: 从外部存储器执行）”

3.5 设置全局零件程序存储器

通过全局零件程序存储器(GDIR)提供一个类似 NC 被动文件系统的结构。它集成在 NC 搜索路径中，用于子程序调用。如设置了全局零件程序存储器(GDIR)，则系统的全部节点都可访问同一个逻辑驱动。各节点可直接从共享的程序存储器中执行零件程序。GRID 替代或扩展了 NC 零件程序存储器。

说明

GDIR 只用于 EES 功能。建立 GDIR 对于 EES 操作而言并不是必须的。

可能的步骤

1. 在“设置驱动器”窗口中设定全局零件程序存储器。
2. 将本地程序存储器中 SPF、MPF 和 WKS 目录下的内容移动到外部驱动器上，如 U 盘。
3. 将备份的目录复制到全局零件程序存储器中。

关于程序存储器管理的更多信息参见下列文档：

参考文档

编程手册工作准备部分：文件和程序管理

章节“程序存储器”

3.6 程序选择

3.6 程序选择

3.6.1 显示软键

操作界面上的软键可以由通用的配置机床数据激活，这些软键可以调用程序列表。

设计软键

MD51041 \$MNS_ENABLE_PROGLIST_USER	
0	无软键
1	显示软键“Prog.-liste”（程序列表）

MD51043 \$MNS_ENABLE_PROGLIST_MANUFACT	
0	无软键
1	显示软键“MANUFACTURER”（制造商）

前提条件

创建程序列表需要下列权限：

- 操作区域“程序管理器”：访问等级 3（口令：用户）或者
- 操作区域“调试”：访问等级 1（口令：制造商）

程序列表

您可以创建一个程序列表，列表包含了从 NC 数据中提取的所需程序，例如：工件、零件程序、子程序。然后通过 PLC 可选择该程序进行处理。

程序列表	目录
plc_proglist_user.ppl	/user/sinumerik/hmi/plc/programlist
plc_proglist_manufacturer.ppl	/oem/sinumerik/hmi/plc/programlist

步骤



1. 如果访问权限等级是 1（口令：制造商），则选择操作区域“调试”。

-或者-



如果访问权限等级是 3（口令：用户），则选择操作区域“程序管理器”。



2. 按下菜单扩展键和软键“程序列表”。
窗口“程序列表”打开。



3. 如果要创建一个程序列表，该列表可用于特定系列的所有机床，按下软键“Manufacturer”。

4. 将光标定位到所需的行中（程序编号）。



5. 按下软键“选择程序”。

窗口“程序”打开。会显示出带有“工件目录”、“零件程序目录”和“子程序目录”的 NC 存储器的文件树形图。



6. 将光标定位到所需的程序上并按下软键“确认”。
所选定的程序及其路径信息被置于列表的第一行。

-或者-

直接在列表中输入程序名称。

在手动输入时要注意准确的路径信息（例如 //NC/WKS.DIR/MEINPROGRAMM.WPD/MEINPROGRAMM.MPF）。

有时需要加上后缀名（.MPF）。

提示：

系统不检查句法。

对于工件请注意，在工件中主程序或工作列表的名称相同。



7. 要从列表中删除程序，可以将光标定位到相应的行上并按下软键“删除”。

-或者-



如要从程序列表中删除所有的程序，可以按下软键“全部删除”。

3.6 程序选择

配置网络

4.1 显示网络一览

在“网络设置一览”窗口中列出了所有可用的网络接口及其设置。此处会显示公司网络的 IP 地址、子网掩码和 MAC 地址并且都能编辑。

步骤



1. 选择操作区域“调试”。



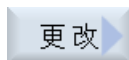
2. 按下菜单扩展键。



3. 按下软键“网络”。



“网络设置一览”窗口打开。



4. 按下软键“更改”，对设置进行修改。

参见

公司网络设置 (页 40)

保存网络设置 (页 41)

4.2 公司网络设置

窗口**公司网络设置**中列出了公司网络接口的配置，当前生效的协议以及防火墙特例。

设置	说明
网关	如果该值不为空，则其中给出的主机作为缺省网关来使用，也就是说，所有不能直接递送的 IP 包都会发送到那里用来转发。
DNS 服务器	如果在此处给定了 DNS 名称服务器（最多 3 个），则该服务器用于分辨主机符号名称，即：在大多数需要 IP 地址的位置上也可以使用主机名称。 名称服务器设置也会通过 NCU 的 DHCP 服务器传递到 DHCP 用户 (TCU, PG)，从而使这些用户也能够用符号名称来工作。
时间服务器	在此处最多可以输入 3 个 NTP 服务器 (UDP/123)，NCU 上的 NTPD 通过此服务器进行时间同步。
主机名	此处可以确定一个本地主机的名称。该手动给出的名称相对于其他所有名称有优先权。 主机名依照以下顺序来决定： <ul style="list-style-type: none"> • 该条目（与文件 <code>basesys.ini</code> 中的主机名一致） • 一个由 DHCP 服务器获取的名称（反向进行 DNS 查找，即哪个名称属于哪个获取的 IP 地址） • 缺省名称 ("NONAME_...") 由于主机名也用作 DNS 名称，因此它必须满足相应的 RFC 要求： <ul style="list-style-type: none"> • ASCII 字母(a-z, A-Z)、数字(0-9)和“-” • 最多 63 个字符
生效的协议	确定公司网络中应使用哪条协议，有 DCP (Discovery and Basic Configuration Protocol) 和 LLDP (Link Layer Discovery Protocol) 两种协议可选。
防火墙例外	此处可以输入在外部网络接口的防火墙中使能的端口。端口 TCP/22 (SSH 协议) 始终使能。

设置	说明
<ul style="list-style-type: none"> • S7 通讯(TCP/102) 	S7 通讯端口
<ul style="list-style-type: none"> • VNC 访问(TCP/5900) 	VNC 访问端口
<ul style="list-style-type: none"> • 更多端口 	<p>此处可以使能更多端口。此时，先输入协议名称 (TCP 或 UDP)，然后输入斜杠 '/' 和待使能的服务的数字端口号。多个端口数据用空格隔开。</p> <p>示例（使能 VNC 服务器）：TCP/5900 TCP/5904 TCP/5905</p>




注意

安全风险

请注意，打开防火墙的端口潜在有安全隐患。请只有在确实需要的情况下才打开防火墙端口。

请注意，每个可访问的服务也可能有安全漏洞。

步骤

- | | |
|---|---------------------------------|
|  | 1. 打开网络一览。 |
|  | 2. 按下软键“公司网络”。
“公司网络设置”窗口打开。 |
|  | 3. 按下软键“更改”，对设置进行修改。 |

4.3 保存网络设置

NCU 的设置可保存为文件。

可以在窗口**保存网络设置：选择保存位置**中选择一个目标文件夹,将网络配置保存在该文件夹中,或在该文件夹中选中一个已有文件，用新设置覆盖该文件。

4.4 基于子站的网络诊断

步骤



1. 网络一览已打开，您已完成了网络设置。
2. 按下软键“保存 NCU 设置”。
“保存网络设置：选择保存位置”窗口打开。
3. 选择一个用于保存网络配置的文件夹。如果已经有一个文件，则将其覆盖掉。
-或者-
按下软键“新建目录”，创建新目录。
-或者-
按下软键“搜索”，找到网络配置文件的保存位置。
4. 如果找到的文件不是所需的配置文件，则按下软键“继续搜索”。
5. 按下软键“取消”，放弃设置。
-或者-
按下软键“确认”，保存网络配置。

4.4 基于子站的网络诊断

借助基于子站的网络诊断可以找出并分析出错的、错误设置的或未连接的网络接口和总线接口。

SINUMERIK 828D 的网络/总线接口有：

- 公司网络 (X130)
- 维修 (X127)
- PROFINET (PLC I/O Interface PN1, PN2)
- PROFIBUS (集成的)

4.4.1 显示 PPU 的网络接口

在窗口“TCP/IP 诊断”中，以树形图的形式显示了网络接口（公司网络 X130 和维修 X127）及其可用性。

详细信息显示

可显示所选接口的详细信息，例如：IP 地址、DNS 名称等。

步骤



1. 选择操作区域“诊断”。



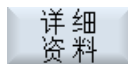
2. 按下菜单扩展按键和软键“TCP/IP 总线”。



3. 按下软键“TCP/IP 诊断”。

窗口“TCP/IP 诊断”打开，显示网络连接的当前可用性。

4. 需要了解接口的详细信息时，将光标移动到该组件上。



5. 按下软键“详细”，显示配置的网络连接的所有可用参数。



6. 按下软键“详细”，退出详细信息画面。



7. 按下软键“返回”，退出窗口“TCP/IP 诊断”，回到窗口“PROFINET/PROFIBUS 诊断”。

4.4.2 TCP/IP 诊断和配置

窗口“TCP/IP 配置”中显示了网络接口的配置。

维修

调试和诊断时使用维修接口 X127。

PG/PC 通过该接口与控制器进行点对点通讯。维修接口的 IP 地址固定设置为 192.168.215.1。

公司网络

通过公司网络可以访问网络驱动器。

PPU 通过以太网接口 X130 连接到公司网络中。

4.4 基于子站的网络诊断




网络连接的可用性

网络适配器接口



可用性

可用性指在所有已发送和接收的数据包中，错误数据包所占的百分比。
公司网络中出现的故障或启动中的起振时间可能会影响可用性的稳定性。

-  绿色 大于 95 %
-  黄色 50 - 95 %
-  红色 小于 50 %

网络连接的详细信息

- **计算机名称**
控制器的计算机名称。
- **DNS 名称**
网络适配器的名称，正如保存在 DNS 服务器 (Domain Name Service) 上的名称。
- **MAC 地址**
网络适配器的物理地址。
- **地址类型**
配置文件 `basesys.ini` 中网络适配器配置的相关信息：
 - DHCP: DHCP 对该网络适配器生效。
连字符“-”后的数据说明了 DHCP 的工作方式：
关 - DHCP 不对该适配器生效
客户端 - 接口上运行的 DHCP 客户端，它由服务器获取 IP 地址和其它数据。
服务器 - 在该接口上，SINUMERIK 控制器提供 DHCP 服务器，它为该网络上的客户端提供 IP。
如果在配置文件中没有对网络适配器进行任何修改，则还输出信息“默认”。
 - 手动
按下软键“修改”，可以手动修改设置，参见章节“可修改参数”和“地址类型”。

说明

在“修改”模式中可以切换“手动”和“DHCP”（仅适用于公司网络：X130）。

- **分配的 IP 地址**
网络适配器的当前 IP 地址。
- **分配的子网掩码**
网络适配器的子网掩码。
- **DHCP 服务器**
DHCP 服务器的 IP 地址（地址类型“DHCP”）。
- **DHCP 服务器状态**
同步 DHCP 服务器的状态：
 - 有效：在所有的 NCU 中，当前 NCU 处于模式“同步服务器”并分配 IP 地址。
 - 待机：NCU 并不是作为 DHCP 服务器激活的。当激活的 DHCP 服务器故障时，该 NCU 可以接收 DHCP 服务器的任务。）
- **DHCP 同步方式**
同步 DHCP 服务器的优先级：
低级，高级，主站
- **DNS 服务器 1, DNS 服务器 2**
DNS 服务器的 IP 地址。
- **网关**
设置为网关的主站 IP 地址。

说明

所有不存在的信息由表格项中的连字符“-”表明。

可修改的参数

以下公司网络接口（X130）参数可以设置。

- 地址类型
- IP 地址
- 子网掩码
- DNS 服务器 1 和 2
- 网关（适用于维修接口和公司网络）

4.4 基于子站的网络诊断

地址类型

- DHCP
DHCP 服务器根据“basesys.ini”中的条目自动设定 IP 地址、子网掩码、DNS 服务器 1、DNS 服务器 2 和网关的可设定参数值。
- 手动
可以手动设定 IP 地址、子网掩码、DNS 服务器 1、DNS 服务器 2 和网关的可设定参数值，这些值随后保存到“basesys.ini”中。

步骤

- | | |
|---|---------------------------------|
|  | 1. 选择操作区域“诊断”。 |
|  | 2. 按下菜单扩展按键和软键“TCP/IP 总线”。 |
|  | |
|  | 3. 按下软键“TCP/IP 诊断”和“TCP/IP 配置”。 |
|  | “TCP/IP 诊断”窗口打开。 |
|  | 4. 按下“更改”软键，修改特定参数。 |
| | 可被编辑的栏显示为白色背景。 |
|  | 5. 输入所需的修改并按下软键“确认”，结束输入。 |
|  | 6. 按下“刷新”软键，刷新显示。 |

4.4.3 以太网诊断

4.4.3.1 网络诊断和 DHCP 服务器

按下软键“网络诊断”，显示网络节点的数据和可用性。诊断结果中包含了以下信息：

- 硬件 ID 号
- TCP/IP 地址/ MAC 地址
- DHCP 服务器 ID
- 节点状态

您可以检查实际拓扑结构和设定拓扑结构是否一致。

步骤



1. 选择操作区域“诊断”。



2. 按下菜单扩展按键和软键“TCP/IP 总线”。



3. 按下软键“TCP/IP 诊断”。



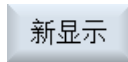
窗口“TCP/IP 诊断”打开，显示网络连接的当前可用性。



4. 按下软键“网络诊断”，扫描整个网络。

“网络状态”窗口打开，显示设备网络 PN 和公司网络 (X130) 的地址。

网络无法访问时，窗口中对应的一栏显示为空。



5. 按下“刷新”软键，刷新显示。

-或者-



按下软键“DHCP 服务器”，显示 DHCP 服务器的信息和状态。

“DHCP 服务器”窗口打开。

窗口上方以列表形式显示了每个激活的 DHCP 服务器的状态、角色（优先级）、MAC 地址、TCU 版本和租赁版本。窗口下方显示了目标 IP 及其地址范围。

和目标不符的数据标为红色。

4.4.3.2 可访问节点

按下软键“可访问节点”，系统会生成一张通过 PROFINET 可访问节点的列表并将它显示在屏幕上。该列表可作为实际拓扑结构和设定拓扑结构相比。

说明

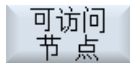
系统没有将手持单元列入“可访问节点”列表中，而是将它们作为“更多节点”插入到拓扑结构中。

4.4 基于子站的网络诊断

步骤



1. “网络状态”窗口打开。



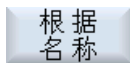
2. 按下软键“可访问节点”。

窗口“可访问节点”打开。

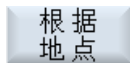
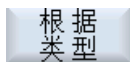
屏幕上以表格形式显示了每个节点的名称、IP 地址、硬件 ID /型号、状态和地点。



3. 按下软键“排序”。



4. 按下软键“根据名称”、“根据 IP”、“根据类型”或“根据地点”来排序。



-或者-



3. 如果选择的节点有进一步信息，可按下软键“详细”。

“详细”窗口打开并显示更多信息。



4. 按软键“缩放+”或“缩放-”，直至达到所需的显示大小为止。



更多节点



3. 按下软键“更多节点”，查找没有在“可访问节点”中显示的节点。

窗口“查找更多节点”打开。必要时您可以修改地址范围，按下“确认”。



此时系统找到的更多节点被添加到表格中，输入栏“硬件 ID/型号”、“地点”、“联系人”、“工作站”和“说明”变为可编辑状态。您可勾选第 1 列的复选框，然后点击“保存”，将节点插入到设定拓扑结构中。

对比



3. 按下软键“对比”，您可以对当前拓扑结构与之前保存的厂商拓扑结构、设定拓扑结构或 xml 格式的备份拓扑结构中的某个结构进行对比。

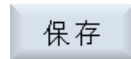


窗口“对比对象”打开。将光标移动到所需选项上，按下“确认”。

两个拓扑结构之间的不同以一定颜色突出显示：

- 红色： 和设定拓扑结构相比，实际拓扑结构中缺少组件或有组件不可访问。
- 粉色： 和设定拓扑结构相比，实际拓扑结构中某些组件的属性不同
- 蓝色： 和设定拓扑结构相比，实际拓扑结构中有新组件

保存



3. 按下软键“保存”，保存实际拓扑结构。
窗口“另存为”打开。



4. 将光标移动到所需选项上，输入文件名称和必要的注释，按下“确认”。

说明

只有至少具有访问级别 3（用户）才能创建厂商拓扑结构和设定拓扑结构，这两份文件始终保存在有效的 DHCP 服务器上。

缺省保存目录：

- 实际拓扑结构 (.xml 或.csv):
根据配置的不同提供有下列保存目录：
 - 网络拓扑结构（保存目录：HMI 数据/数据备份/网络拓扑结构）
 - 用户 CF 卡
 - 网络驱动器
 - USB在 csv 格式中，分号“;” 用作分隔符，可以在 sldgconfig.xml 中加以修改。
- 设定拓扑结构：
/user/common/tcu/\$soll/\$soll.xml
- 厂商拓扑结构：
/user/common/tcu/\$soll/\$oem_soll.xml

4.4 基于子站的网络诊断

调整操作区域“加工”

自动切换

通过“MD51040 \$MNS_SWITCH_TO_MACHINE_MASK”可以设置，在哪种条件下系统可以自动切换到操作区域“加工”。

MD51040 \$MNS_SWITCH_TO_MACHINE_MASK		0 (缺省设置)
位 0 = 1	在“程序管理器”操作区中选择程序时不会自动切换到“加工”操作区。	
位 1 = 1	在“程序管理器”操作区中选择程序时不会自动切换到“加工”操作区。 在通过机床操作面板切换操作方式时，不会自动切换到“加工”操作区。	
位 2 = 1	在“程序”操作区中选择程序时不会自动切换到“加工”操作区。	
位 3 = 1	在“程序”操作区中选择程序或执行程序时，不会自动启动程序段搜索。	

5.1 设置实际值显示的字体大小

设置字体大小

在操作区“加工”中，MCS 或 WCS 实际值窗口可以通过两种不同的字体大小显示。

通过通道机床数据可以确定用大字体显示的实际值数量。

MD52010 \$MCS_DISP_NUM_AXIS_BIG_FONT		大字体的实际值数量
= 3	大字体的实际值数量 (缺省设置)	

5.2 插入用户专用商标

OEM 商标

在操作区域“加工”中可以在标题行中看到西门子的商标。

5.3 设置 G 代码组的显示

该西门子商标可以替换成自己的商标。

步骤:

1. 命名自己的商标“logo.png”。

说明

文件名只能由小写字母构成。

2. 根据位图大小，在如下目录中保存自己的商标：**/oem/sinumerik/hmi/ico** 或 **user/sinumerik/hmi/ico**

目录	位图大小
/lco640	122 x 19
/lco800	160 x 24
/lco1024	199 x 30

5.3 设置 G 代码组的显示

在操作区域“加工”中可以看到“G 功能”窗口。

窗口中最多可以显示 16 个 G 代码组。在文件“slmagcodeconfig.ini”中设置，哪个 G 代码组在哪个位置显示。

设置 G 代码组

段落	含义
Selected G-Groups	该区域用于在西门子模式下运行控制系统。
Selected G-Groups ISO-Mode	该区域用于在 ISO 模式下运行控制系统。

按以下步骤输入 G 代码组和位置：

SelectedGGroupN = M	
N	G 代码组 M 的显示位置。 如果位置需要为空，则不输入任何编号 (M)，例如 “SelectedGGroup1 = ”
M	在 N 位置显示的 G 代码组。

文件结构

```
[Selected G-Groups]
SelectedGGroup1 = 1
SelectedGGroup2 = 2
SelectedGGroup3 = 3
SelectedGGroup4 = 4
SelectedGGroup5 = 5
SelectedGGroup6 = 6
SelectedGGroup7 = 7
SelectedGGroup8 = 8
SelectedGGroup9 = 9
SelectedGGroup10 = 11
SelectedGGroup12 = 12
SelectedGGroup13 = 13
SelectedGGroup14 = 14
SelectedGGroup15 = 15
SelectedGGroup16 = 16
[Selected G-Groups ISO-Mode]
... (与西门子区相同)
```

步骤

1. 从目录
`/siemens/sinumerik/hmi/template/cfg` 中复制文件“`slmagcodeconfig.ini`”。
2. 将文件保存在以下目录：`/oem/sinumerik/hmi/cfg` 或者 `/user/sinumerik/hmi/cfg`。
只要该文件位于用户特定的目录中，则该文件的条目优先于西门子的文件。如在用户专用文件中缺少注册信息，则使用西门子文件中的。
3. 打开文件并将相应 G 代码的编号输入到需要的位置“`SelectedGGroup1 = 5`”。
即在第 1 个位置显示第 5 个 G 代码组。

5.4 配置通道运行信息

概览

在操作区“加工”中，通道运行信息显示在标题的第三行中。具备两种通道运行信息：

1. 要求操作以继续处理程序的通道运行信息，例如：缺少进给使能时，M01生效。
该通道运行信息通过感叹号加以标注。
2. 要求不进行操作以继续处理程序的通道运行信息。停止条件长时间(>1s)生效，由程序处理中断，例如：停留时间生效，等待换刀。
该通道运行信息通过钟表符号加以标注，特定时间届满后不再显示。

文件

相应的设置在文件“slmahdconfig.ini”中确定。

配置通道运行信息

段落	含义
CondStopIcons	确定和特定通道运行信息一同显示的符号。 如果没有为特定的通道运行信息规定符号，则显示“DefaultIcon”中确定的符号。
ConDelayTime	确定特定信息显示前应持续的时间段（毫秒为单位）。 <ul style="list-style-type: none"> ● 如果没有为特定的通道运行信息规定时间，则使用“ConDelayTime”下给定的时间。 ● 如果需要立即显示通道运行信息，则必须将该时间设为“0”。

步骤

1. 从目录 /siemens/sinumerik/hmi/template/cfg 中复制文件“slmahdconfig.ini”。
2. 将文件保存在以下目录： /oem/sinumerik/hmi/cfg 或者 /user/sinumerik/hmi/cfg。
3. 打开该文件并执行所需的设置。

文件“slmahdconfig.ini”

标准设计

[CondStopIcons]

DefaultIcon = condwait.png

- 1 = condstop.png ; 没有 NC 就绪
- 2 = condstop.png ; 没有 BAG 就绪
- 3 = condstop.png ; 急停激活
- 4 = condstop.png ; 带有停止的报警激活
- 5 = condstop.png ; M0/M1 激活
- 6 = condstop.png ; 单程序段中的程序段已结束
- 7 = condstop.png ; NC 停止激活
- 8 = condstop.png ; 无读入许可
- 9 = condstop.png ; 无进给许可
- 10 = condwait.png ; 停留时间激活
- 11 = condwait.png ; 无帮助功能确认
- 12 = condstop.png ; 无轴许可
- 13 = condwait.png ; 没有达到准确停止
- 14 = condwait.png ; 等待定位轴
- 15 = condwait.png ; 等待主轴
- 15 = condwait.png ; 等待其它通道
- 17 = condstop.png ; 等待进给倍率
- 18 = condstop.png ; NC 程序段有错
- 19 = condstop.png ; 等待外部 NC 程序段
- 20 = condwait.png ; 等待同步动作
- 21 = condwait.png ; 程序段运行过程激活
- 22 = condstop.png ; 无主轴许可
- 23 = condstop.png ; 轴进给倍率
- 24 = condwait.png ; 等待刀具更换确认
- 25 = condwait.png ; 齿轮换档
- 26 = condwait.png ; 等待位置调节
- 27 = condwait.png ; 等待螺纹切削
- 28 = condwait.png ; 备用
- 29 = condwait.png ; 等待冲孔

5.4 配置通道运行信息

30 = condwait.png	; 等待安全运行
31 = condwait.png	; 无通道准备
32 = condstop.png	; 摆动激活
33 = condwait.png	; 轴更换已激活
34 = condwait.png	; 轴容器旋转
35 = condwait.png	; AXCT: 随动轴激活
36 = condwait.png	; AXCT: 引导轴激活
37 = condwait.png	; AXCT: 跟踪激活
38 = condwait.png	; AXCT: 内部状态切换
39 = condwait.png	; AXCT: 进给轴/主轴禁用
40 = condwait.png	; AXCT: 补偿运动激活
41 = condwait.png	; AXCT: 轴更换已激活
42 = condwait.png	; AXCT 轴插补器激活
43 = condwait.png	; 等待编译循环
44 = condwait.png	; 访问系统变量
45 = condstop.png	; 搜索目标已找到
46 = condwait.png	; 快速后退已触发
47 = condwait.png	; AXCT: 等待主轴停止
48 = condwait.png	; 机床数据补偿
49 = condwait.png	; 轴更换: 轴联动
50 = condwait.png	; 轴更换: 快速提升激活
51 = condwait.png	; 轴更换: 新配置激活
52 = condwait.png	; 轴更换: AXCTSW 激活
53 = condwait.png	; 轴更换: Waitp 激活
54 = condwait.png	; 轴在另一个通道中
55 = condwait.png	; 轴更换: 轴是 PLC 轴
56 = condwait.png	; 轴更换: 轴是摆动轴
57 = condwait.png	; 轴更换: 轴是 Jog 轴
58 = condwait.png	; 轴更换: 指令轴
58 = condwait.png	; 轴更换: 轴是 OEM 轴
60 = condwait.png	; 联动的随动轴
61 = condwait.png	; 联动的随动轴
62 = condwait.png	; 联动的从动轴


```

63 = condstop.png ; 在 M0 循环结束处停止
64 = condstop.png ; 在 M1 循环结束处停止
65 = condwait.png ; 等待轴位于固定点
66 = condwait.png ; 主从功能转换有效
67 = condwait.png ; 轴更换： 轴是单轴
68 = condstop.png ; 停止： 搜索后到达目标
69 = condwait.png ; 同步运行 同步主轴
70 = condwait.png ; 关闭位置同步主轴
71 = condwait.png ; 等待转换轴使能
72 = condstop.png ; 由于可能发生的碰撞等待
73 = condstop.png ; Jog： 位置到达
74 = condstop.png ; Jog： 方向禁止
75 = condwait.png ; 制动请求
76 = condwait.png ; 等待： G4 : U
77 = condstop.png ; PLC 轴向进给禁止
78 = condstop.png ; 等待轴向进给释放
79 = condwait.png ; 同步动作轴向进给禁止
80 = condwait.png ; 等待主主轴转速
81 = condwait.png ; 等待参数段切换
82 = condwait.png ; 等待变压器转换前的运行结束
83 = condwait.png
84 = condwait.png
85 = condwait.png

```

```
[CondDelayTime]
```

```
DefaultDelayTime=1000
```

```

1 = 0 ; 没有 NC 就绪
2 = 0 ; 没有 BAG 就绪
3 = 0 ; 急停激活
4 = 0 ; 带有停止的报警激活
5 = 0 ; M0/M1 激活
6 = 0 ; 单程序段中的程序段已结束
7 = 0 ; NC 停止激活

```

5.5 取消程序测试

- 8 = 0 ; 无读入许可
- 9 = 0 ; 无进给许可
- 10 = 0 ; 停留时间激活
- 12 = 0 ; 无轴许可
- 16 = 0 ; 等待其它通道
- 17 = 0 ; 等待进给倍率
- 18 = 0 ; NC 程序段有错
- 19 = 0 ; 等待外部 NC 程序段
- 22 = 0 ; 无主轴许可
- 23 = 0 ; 轴进给倍率
- 32 = 0 ; 摆动激活
- 45 = 0 ; 搜索目标已找到
- 76 = 0 ; 等待: G64 及 U

5.5 取消程序测试

有多种程序测试的方法可用于测试和试运行新的零件程序，这些方法可以使用“程序测试”功能来运行。

标准设置下，此功能已激活。可在“加工”操作区域中的“程序控制”窗口通过复选框“PRT”来选择程序测试。

取消程序测试

可使用如下配置机床数据取消程序测试。取消后，状态栏中不再显示“PRT”。

MD51039 \$MNS_PROGRAM_CONTROL_MODE_MAS		机床程序控制的选项
K		
位 0 = 0	程序测试功能不可用	

关于测试的详细描述可参见如下文献：

文献

功能手册 基本功能；BAG、通道、程序运行、复位特性 (K1)

5.6 激活“示教”功能

打开示教功能

此外，可通过通用配置机床数据打开示教功能。

设置

MD51034 \$MNS_TEACH_MODE	
位 0 = 1	显示软键“程序示教”。按下软键“接收”后，程序将获得已经到达的位置值，缺省值。
位 1 = 1	示教程序段的接收可以由 PLC 禁止。 DB1700.DBX1000.0 = 0 接收程序段 DB1700.DBX1000.0 = 1 不接收程序段

此外，该功能的访问还受存取权限影响。

MD51045 \$MNS_ACCESS_TEACH_IN	示教保护级别
= 7	确定示教时将运行过程写入 MDA 缓冲器/零件程序的保护等级，缺省值。

MD51050 \$MNS_ACCESS_WRITE_PROGRA M	写入零件程序保护级别
= 7	确定写入零件程序的保护等级，缺省值。

设置通道专用机床数据：

MD20700 \$MC_REFP_NC_START_LOCK	不带参考点的 NC 启动禁用
= 0	

激活示教功能

1. 设置机床数据并执行复位。
2. 切换到“AUTO”和“MDA”运行方式。
3. 选择机床功能 TEACHIN。
4. 按下软键“程序示教”。

5.7 程序段搜索

示教未激活

若软键“程序示教”处于未激活状态，则原因可能如下：

- 该程序不处于“复位”、“停止”或“中断”状态。
- 不是所有必须回参考点的轴都已回参考点。

接收 STAT 和 TU

若需将机床的支轴位置和回转轴的位置导入至运行程序段中，则须点击复选框“STAT”和“TU”。

可通过设置下列通用机床配置数据来显示上述复选框：

MD51032 \$MNS_STAT_DISPLAY_BASE	数字基准显示：支轴位置 STAT
= 1	显示、支轴位置 STAT

MD51033 \$MNS_TU_DISPLAY_BASE	数字基准显示：回转轴位置 TU
= 1	显示、回转轴位置 TU

5.7 程序段搜索

5.7.1 激活程序段搜索模式

设置程序段查找方式

通过下列机床数据您可以确定 G 代码程序中程序段的查找方式：

MD51028 \$MNS_BLOCK_SEARCH_MODE_MAS K	设置可用查找方式的位标记
位 0 = 1	程序段查找，带计算，不逼近断点
位 1 = 1	程序段查找，带计算，逼近断点
位 4 = 1	程序段查找，不带计算
位 5 = 1	程序段查找，带试车

通过下列机床数据您可以确定 JobShop 程序(ShopMill, ShopTurn)中程序段的查找方式:

MD51024 \$MNS_BLOCK_SEARCH_MODE_MAS K_JS	设置可用查找方式的位标记 (ShopMill, ShopTurn 为单通道查找)
位 0 = 1	程序段查找, 带计算、不逼近断点 (缺省设置)
位 1 = 1	程序段查找, 带计算, 逼近断点 说明: ShopMill、ShopTurn 程序上只能通过 G 代码进行程序段搜索。
位 3 = 1	跳过 EXTCALL 程序
位 5 = 1	程序段查找, 带试车

说明

只有在机床数据 MD51024 和 MD51028 中定义了不止一种程序段查找方式时 (例如: “程序段查找, 带计算, 不逼近断点” 和 “程序段查找, 带计算, 逼近断点”) 才会显示 “查找方式” 软键。

注意, “查找方式” 窗口中有效设置的查找方式也是基于程序的。

5.7.2 由外部处理的加速程序段搜索过程

可通过加速程序段搜索运行零件程序, 直至到达程序的所需位置。

设置

若已激活了“跳过 EXCALL”功能, 则必须设置下列通用配置机床数据:

MD51028 \$MNS_BLOCK_SEARCH_MODE_MAS K	可使用搜索模式的位掩码
位 3 = 1 (33H 至 3BH)	

对于外部主程序而言, 不包含计算的程序段搜索速度总是较快的。只会加载自搜索位置起的程序段。

5.8 手动加工

简介

“Manual Machine”功能使人员可以利用丰富多样的循环进行手动加工，而无需编程。



软件选件

使用该功能需要下列软件选件：

“ShopMill/ShopTurn”

激活

MD52212 \$MCS_FUNCTION_MASK_TEC		工艺通用的功能屏幕
H		
位 8	Manual Machine(ShopMill/ShopTurn)	

使能

MD52214 \$MCS_FUNCTION_MASK_MILL		铣削功能屏幕
位 5	使能刀具主轴的控制	
= 0	通过 NC/PLC 接口	
= 1	通过操作界面	
位 6	使能旋转主轴的控制	
= 0	通过 NC/PLC 接口	
= 1	通过操作界面	

MD52218 \$MCS_FUNCTION_MASK_TURN		车削功能屏幕
位 3	使能主主轴的控制	
= 0	通过 NC/PLC 接口	
= 1	通过操作界面	
位 4	使能副主轴的控制	
= 0	通过 NC/PLC 接口	
= 1	通过操作界面	

位 5	使能刀具主轴的控制
= 0	通过 NC/PLC 接口
= 1	通过操作界面

SD43200 \$SA_SPIND_S	通过 NC/PLC 接口信号给出主轴启动转速；通过操作软件输出数值
SD43202 \$SA_SPIND_CONSTCUT_S	通过 NC/PLC 接口信号给出主轴启动切削速度；通过操作软件输出数值
SD43206 \$SA_SPIND_SPEED_TYPE	通过 NC/PLC 接口信号给出主轴启动转速类型；通过操作软件输出数值

更多设置

推荐以下更多设置：

手轮	
MD11346 \$MN_HANDWH_TRUE_DISTANCE = 3	手轮设定行程还是设定速度
MD11602 \$MN_ASUP_START_MASK 位 0, 3 = 1	忽略 ASUP 的停止原因
MD20150 \$MC_G_CODE_RESET_VALUE [7] = 2	G 功能组的初始化设置；G54 引导启动有效
MD20624 \$MC_HANDWH_CHAN_STOP_COND = 0x7FFF	确定通道专用的手轮运行性能
MD32084 HANDWH_STOP_COND = 0x7FFF	手轮运行性能
MD52212 FUNCTION_MASK_Tech 位 9 = 1	工艺通用的功能屏幕；通过软键设置零点偏移

说明

如果机床数据 MD52212 FUNCTION_MASK_Tech 的位 9 被置位，则会在窗口“零点偏移 – G54 ... G599”中显示软键“NPV 选择”。

5.9 多通道支持

工作区域限制	
MD28600 \$MC_MM_NUM_WORKAREA_CS_GROUPS > 0	坐标系专用的工作区域限制数量
MD20150 \$MC_GCODE_RESET_VALUES[59] = 2	G 功能组的初始化设置；激活功能组的工作区域限制

说明

如果机床数据 MD20150 \$MC_GCODE_RESET_VALUES[59] = 2 的位被置位，则会在窗口“Manual Machine”中显示软键“挡块”。前提是 MD28600 \$MC_MM_NUM_WORKAREA_CS_GROUPS 的值也被置位为 1。

进给率	
SD42600 \$SC_JOG_FEED_PER_REV_SOURCE = -3	JOG 方式下的控制系统转速进给
SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE [X 轴] = -3	定位轴 X 的旋转进给率
SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE [Z 轴] = -3	定位轴 Z 的旋转进给率

说明

要注意轴专用的设置数据 SD43300 SA_ASSIGN_FEED_PER_REV_SOURCE 必须相同。

参考文档

- 操作：
操作手册，车削版/铣削版，章节“Manual Machine 操作”
- 编程：
编程手册，基本原理，章节“工作区域限制”

5.9 多通道支持

在操作区域“加工”中可为一个运行方式组中的任意两个通道（n 中的 2 个）显示双通道试视图。

通过使用多通道编辑器，可在多通道机床上同时打开、修改或创建多个时间上相互独立的程序。此时相关程序会依次显示在编辑器中。

在实际加工前还可在控制系统上进行程序模拟。

此外可根据同步点校正程序。

说明

只有带以下 CNC 软件功能的车削和磨削工艺才适用多通道支持：

- PPU 280/281 System Software Turning advanced
 - PPU 280/281 System Software G-Tech advanced
-

前提条件

- 对于多通道机床，以下数据必须在所有通道中都相同：
 - 尺寸单位
 - 零点偏移（例如 G54）
 - 零点偏移 Z 值（可选）
 - 毛坯
 - 转速限制
- 此外下列 JobShop 数据必须在所有通道中相同：
 - 回退平面（分为主主轴和副主轴）
 - 尾座（分为主主轴和副主轴）
 - 换刀点（分为主主轴和副主轴）
 - 安全距离
 - 顺铣和逆铣

创建多通道程序

进行多通道加工时，需要包含以下数据的工作列表：

- 程序名称
- 程序类型
- G 代码
- ShopTurn 程序
- 通道分配

5.9 多通道支持

**软件选件**

需要选件“ShopMill/ShopTurn”用于创建和编辑 ShopTurn/ShopMill 程序。

在创建了一个工作列表后，可在“多通道数据”参数屏幕中将 G 代码和 JobShop 数据进行汇总，之后再将其作为循环调用保存在工作列表中。只要工作列表中至少有一个 ShopTurn 程序时，就会显示“多通道数据”参数屏幕。

仅编程 G 代码程序

如果在工作列表中只有 G 代码程序，可以取消“多通道数据”功能。为此设置以下机床数据：

MD51228 \$MNS_FUNCTION_MASK_TE CH	工艺通用的功能屏幕
位 0	G 代码编程，无多通道数据
= 1	在只有 G 代码程序的工作列表中不提供多通道数据。不显示软键“多通道数据”。

模拟和同步记录

6.1 模拟概述



软件选件

记录功能需要使用选件
“3D 模拟 1（成品件）”。

SINUMERIK Operate 中集成了一个单独的程序编译器(SNCK)和一个单独的模拟数据环境，为加工模拟提供支持。SNCK 编译器可编译 SINUMERIK 系列产品的所有句法，并通过和 NCK 数据环境的对比纳入了机床上安装的特殊用户选件。在必要时，操作软件会一直比较模拟数据和 NCK 数据环境（初始化数据、宏指令、用户数据、刀具数据和加工循环），或只在刀具数据或加工循环有所变化时进行数据对比。

在具有特定运动结构的机床上，可以在系统界面上在工件坐标系中模拟加工工艺的整个过程，重点是钻削、铣削、车削和磨削工艺。

- 成品的模拟是用真实 NC 数据进行的。
- 在每次更换零件程序时系统会自动对比 NC 数据。
- 模拟功能可以动态展示整个加工过程，甚至支持 5 轴转换(TRAORI) 和回转功能。
- 铣床上的车削
- 传统铣床加工通道模拟。
- 在带 B 轴的车床上，可最多模拟 4 个加工通道。
- 以智能的方式确定程序段处理时间和程序处理时间。
- 连续不断地绘制工件，模拟画面更新更加迅速
- 最佳分辨率，无论截取哪个部分的画面。
- 可模拟任何切削。
- 从 NCU 720.x 起，在加工一个工件的同时可模拟另一个工件的加工过程。
- 模拟 3D 图（可选）。

6.1 模拟概述

模拟存储器配置

- 模拟画面更新非常迅速，但这也意味着动态分辨率比较低。
- 在复杂的工件上，绘图程序过慢，在绘制某些非常复杂的工件时甚至无法工作。
- “Quickview”，用于模具制造零件程序的快速查看。
- 在使用某些带特殊运动结构的编译循环时无法使用模拟功能。
- 仅部分支持编译循环。
- 无法读出毛坯的 CAD 数据，比如在铸造加工中。
- 无法模拟轴的耦合过程。（联动、轴的主值耦合、电子齿轮箱）：
 - 联动：TRAILON(<从动轴>, <主动轴>, <耦合系数>)
 - 轴的主值耦合：LEADON(从动轴, 主动轴, 曲线表 n)，实际值耦合和设定值耦合
 - 电子齿轮箱：EGDEF(FA, LA1, 耦合 1, LA2, 耦合 2, ... LA5, 耦合 5), EGON, EGONSYN, EGONSYNE
- 在模拟中参考 G74 无效。
- //不支持
 - 轴容器
 - 对零件程序和 PLC 的依赖性
 - 多工位加工
 - CNC 功能同步动作，用于 NC 的安全可编程逻辑（SPL）
- 仅部分支持不同通道的不同技术和不同的轴坐标系统。
- 在磨削工艺中不可使用 3D 模拟。
- 不评估软件限位开关：

在一个刀具切换的生产周期里，机床数据“软件限位开关+Z 轴”作为切换位置使用：
G0 Z=\$MA_POS_LIMIT_PLUS[Z]-0.1

在模拟中该位置不可用，而是机床数据的预定设置= $1e8 = 100.000.000$ 。这些预定的位置在模拟中不处理。因此，模拟很慢，加工时间激增，例如计算的时间为 555 小时，而不是 2 分钟。

- 模拟中的程序启动
在一个通道中设置了 MD22622 \$MCS_DISABLE_PLC_START 时，此通道在模拟中不会自动启动。只能通过零件程序指令“START”从另一个通道启动。该功能不适用于 SINUMERIK 828D。
- 模拟中的报警显示
在模拟功能中报警的输出方式和正常程序处理过程基本一样，只是它最多显示五条报警。报警位于图形上方，以便和机床报警加以区分。
如果同步记录和模拟具有不同的故障报告动作（例如：句法错误），则可能是在编程中定义了不同的硬件访问方式：模拟不访问硬件，只是内部读取输入 0，然后结束。而同步记录则出现报警。

6.2 设置用于模拟的坐标系

车削工艺

例如对于机床运动学：

- 使用两个几何轴的传统车削
- 三个主轴： 主主轴，副主轴，变量
- 副主轴滑块，尾座为 NC 轴。
- B 轴： 较准刀具主轴上的车刀
- 使用几何轴的铣削： TRANSMIT, TRACYL, TRAANG

用于车削工艺：

MD52200 \$MCS_TECHNOLOGY = 1	车削工艺
MD52000 \$MCS_DISP_COORDINATE_SYSTEM = 34 (举例)	坐标系的位置 (可变)

铣削工艺

例如对于机床运动学：

- 5 轴铣削： 回转/TRAORI
- 回转头切换

6.2 设置用于模拟的坐标系

用于铣削工艺:

MD52200 \$MCS_TECHNOLOGY = 2	铣削工艺
MD52000 \$MCS_DISP_COORDINATE_SYSTEM = 0	坐标系的位置 (可变)

磨削工艺

适用于磨削

MD52200 \$MCS_TECHNOLOGY = 3	外圆磨削工艺
MD52200 \$MCS_TECHNOLOGY = 4	平面磨削工艺
MD52000 \$MCS_DISP_COORDINATE_SYSTEM = 0	使用操作台的外圆磨削和平面磨削
MD52000 \$MCS_DISP_COORDINATE_SYSTEM = 6	使用工作台的平面磨削

轴的含义

MD52206 \$MCS_AXIS_USAGE[n]	通道中轴的含义 [n]为通道轴号
= 0	没有特定含义
= 1	刀具主轴 (动力刀具)
= 2	辅助主轴 (动力刀具)
= 3	主主轴 (车削)
= 4	主主轴的 C 轴 (车削)
= 5	副主轴 (车削)
= 6	副主轴的 C 轴 (车削)
= 7	副主轴的直线轴 (车削)
= 8	尾座 (车削)
= 9	中心架 (车削)

请通过以下通道专用的机床数据输入旋转轴的旋转方向，该旋转轴没有在 ToolCarrier 或 5 轴转换中配置。

MD52207 \$MCS_AXIS_USAGE_ATTRIB[n]	轴的属性 [n]通道轴号
位 0	围绕第 1 几何轴旋转（旋转轴）
位 1	围绕第 2 几何轴旋转（旋转轴）
位 2	围绕第 3 几何轴旋转（旋转轴）
位 3	旋转方向逆转（适用于旋转轴）

机床数据 MD52290 \$MCS_SIM_DISPLAY_CONFIG 仅在 OP019 上起作用。对于 SINUMERIK 828D 没有任何意义。

MD52290 \$MCS_SIM_DISPLAY_CONFIG	通道的状态栏在模拟窗口中的位置。
位 0	左上角
位 1	右上角
位 2	左下角
位 3	右下角

在以下通道专用机床数据中，在每个通道中至少为一根几何轴设定一个不为 0.0 的值，例如 0.001。当值为 0.0 时，系统会认为此参数尚未设置。

MD53230 \$MCS_SIM_START_POSITION	启动模拟时的轴位置
只有在为至少一根几何轴设置了不为 0 的值时，才能进行模拟。	

取消模拟

设置以下机床数据关闭模拟功能：

MD51226 \$MNS_FUNCTION_MASK_SIM	模拟的功能标记
位 1 = 1	取消模拟
位 10 = 1	隐藏零点符号

6.3 同步记录概述



软件选件

而“同步记录”功能需要使用选件“同步记录（实时模拟）”。



软件选件

显示尾座需要选件“ShopMill/ShopTurn”。



软件选件

同步记录功能 3D 图需要使用选件“3D 模拟 1（成品件）”。

在机床开展加工的同时，系统可在屏幕上同步绘制出刀具的运动轨迹，以 3 面视图或 3D 视图显示。工件的外观、视图与图形化的“模拟”功能一致。

说明

为避免遗漏某些显示，在启动系统前必须提前激活同步记录功能。

示例：激活主轴卡盘数据中的尾座图

步骤：

1. 通过在程序中选择使尾座可见。
2. 给出尾座尺寸：
在操作区域内选择“参数”→“设置数据”→“主轴卡盘数据”并输入尾座长度和直径参数“ZR”和“XR”。
在下列机床数据中写入尾座的尺寸：

MD52218 \$MCS_FUNCTION_MASK_TURN	车削功能屏幕
位 2 = 1	激活尾座： 在程序中选择“尾座”后，尾座即可见。

MD53240 \$MAS_SPINDLE_PARAMETER[i]	主轴卡盘数据，单位：mm
[0]: 卡盘尺寸	
[1]: 挡块尺寸	
[2]: 卡爪尺寸	

MD53241 \$MAS_SPINDLE_CHUCK_TYPE	主轴卡盘类型
0 = 从外面夹紧	
1 = 从里面夹紧	

MD53242 \$MAS_TAILSTOCK_PARAMETER[i]	尾座尺寸，单位：mm
[0]: 尾座直径	
[1]: 尾座长度	

激活处理时间的显示

在功能“程序运行时间”下提供了计时器，它相当于系统变量。NCK 专用的计时器会一直保持工作，以测量上次控制系统启动后经过的时间，而该通道专用的计时器必须通过下述通道专用的机床数据启动：

MD27860 \$MC_PROCESS_TIMERMODE	程序运行时间计算使能。
位 0 = 1	计算所有程序的总运行时间有效 (\$AC_OPERATING_TIME)。
位 1 = 1	计算当前程序运行时间有效 (\$AC_CYCLE_TIME)。

6.4 毛坯夹具

位 4 = 1	空运行进给有效时计算。
位 5 = 1	程序测试时计算。

6.4 毛坯夹具

毛坯夹具功能可用来选择并确定毛坯的夹具位置。

实行该功能要求机床能满足以下前提条件之一或多个：

前提条件

- 机床具有多种夹具方法可供选择。
- 机床具有一种不会默认显示在 Z 方向上（例如：钻头）的夹具方法。

以下情况中可以在机床数据 MD53220 和 MD52207 中注明回转轴的位置和方向：

- 回转轴的位置和方向朝着 MCS 零点相反的方向移动，但不能通过 Toolcarrier 或 TRAORI 确定。
- 车床 X 轴或 Z 轴上的主轴或副主轴的基准点朝着 MCS 零点相反的方向移动。
- 副主轴不能借助于滑块运动，而是位于一个固定的位置上。

必须通过以下机床数据指定机床坐标轴中的位置，才能在模拟/同步记录时将旋转轴显示在正确的位置：

MD53220 \$MAS_AXIS_MCS_POSITION[n]	MCS 中轴的位置 [n] 0 = X, 1 = Y, 2 = Z
------------------------------------	--------------------------------------

有关回转轴转换的其他信息参见章节“配置循环 (页 79)”。

功能

设定以下通道专用的机床数据：

MD52207 \$MCS_USAGE_ATTRIB[n]	通道中轴的属性 [n] 通道轴号
位 0	围绕第 1 几何轴旋转（旋转轴）
位 1	围绕第 2 几何轴旋转（旋转轴）

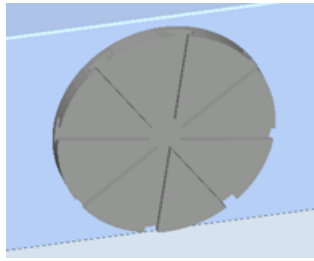
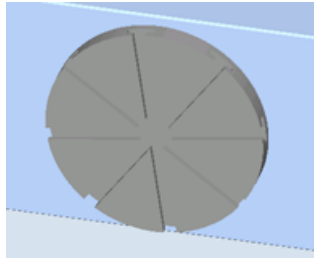
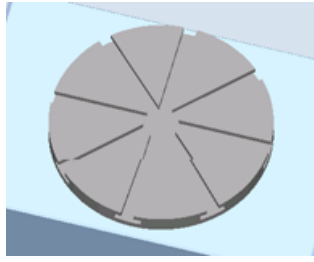
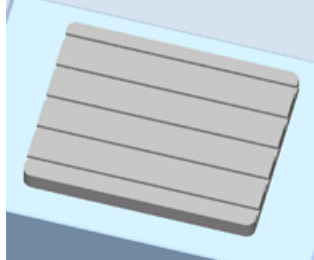
位 2	围绕第 3 几何轴旋转（旋转轴）
位 8	提供用于毛坯夹具的旋转轴（铣削机床上）

MD52214 \$MCS_FUNCTION_MASK_MILL	铣削功能屏幕
位 1	提供用于毛坯夹具的固定工作台（铣床上）

从示例中获取 MD20080 的机床配置。

功能可能性

程序头中可选择毛坯夹具 A、B、C 并设置工作台：

夹具	视图	机床数据
A 轴		MD52207 \$MCS_USAGE_ATTRIB 位 0 和 位 8 = 1
B 轴		MD52207 \$MCS_USAGE_ATTRIB 位 1 和 位 8 = 1
C 轴		MD52207 \$MCS_USAGE_ATTRIB 位 2 和 位 8 = 1
工作台		MD52214 \$MCS_FUNCTION_MASK_MILL 位 1 = 1

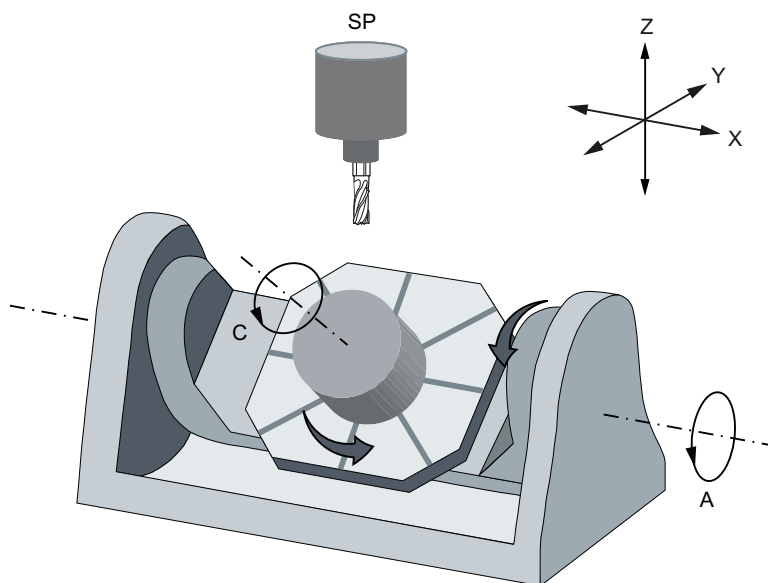
6.4 毛坯夹具

说明

为了在程序头上保留选择可能性，必须为毛坯件夹紧装置定义两个旋转轴。

机床配置示例

A 轴围绕 X 轴旋转的机床配置示例：



MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[通道内的通道轴名称
]		
[0]	X	
[1]	Y	
[2]	Z	
[3]	主轴	
[4]	A	
[5]	C	

MD52207 \$MCS_AXIS_USAGE_ATTRIB[4]		通道 4 中旋转轴 A 的属性
= 181H		

位 0 = 1	围绕第 1 几何轴旋转（旋转轴）
位 8 = 1	提供用于毛坯夹具的旋转轴（铣削机床上）

参考文档

操作手册 铣床版

- 章节“创建 ShopMill 程序”>“程序头”
- 章节“创建 ShopMill 程序”>“修改程序设置”

6.4 毛坯夹具

配置循环

7.1 激活车削/铣削/钻削/磨削工艺

在控制系统启动（上电）时会自动载入循环。

循环是由循环机床数据和循环设定数据配置的，具体而言，这些数据有：

- 通用循环机床数据
- 通用专用循环机床数据
- 轴专用循环机床数据
- 通用循环设定数据
- 通道专用循环设定数据
- 轴专用循环设定数据

说明

确定工艺

通过订购的软件包确定 SINUMERK 828D 的工艺（车削、铣削或磨削）。工艺不可通过机床日期设置。

调整操作界面

借助下列通道专用的配置机床数据您可以设置软键显示，该软键用于在程序编辑器中选择循环支持：

MD52200 \$MCS_TECHNOLOGY	工艺
= 0	车削设置
= 1	车削工艺的设置
= 2	铣削工艺的设置
= 3	外圆磨削工艺的设置
= 4	平面磨削工艺的设置

7.1 激活车削/铣削/钻削/磨削工艺

MD52201 \$MCS_TECHNOLOGY_EXTENSI ON	带有多种工艺的联合机床上的工艺扩展
= 1	车削工艺
= 2	铣削工艺
例如：采用铣削工艺的车床 MD52200 \$MCS_TECHNOLOGY = 1 MD55201 \$MCS_TECHNOLOGY_EXTENSION = 2	
= 3	扩展的外圆磨削工艺的设置
= 4	扩展的平面磨削工艺的设置

文档：磨削

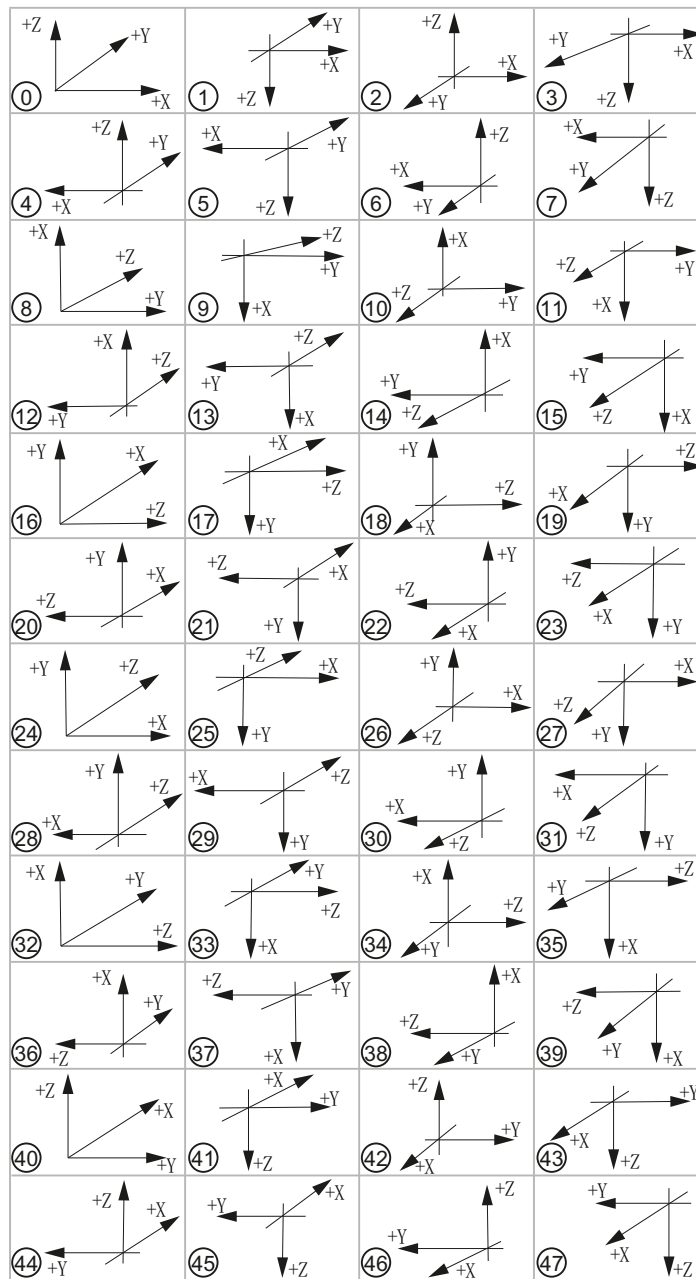
关于磨削工艺的信息可以参见下列文档：

- 章节“磨削 (页 145)”
- 磨削操作手册
- 编程手册之工作准备分册；章节“循环外部编程”：
 - CYCLE495 - 成型
 - CYCLE4071 - CYCLE4074: 纵向磨削循环
 - CYCLE4075 - CYCLE4079: 平面磨削循环
- 功能手册之扩展功能分册；章节“补偿 (K3)” > “插补补偿” > “液压缸故障补偿”

设置坐标系

MD52000 \$MCS_DISP_COORDINATE_SYSTE M	坐标系的位置
= 0	缺省设置
使用该机床数据可进行操作界面坐标系位置与机床坐标系的匹配。在操作界面上会自动根据所选位置更新所有的辅助图、过程图、模拟和与旋转方向有关的输入栏。	

坐标系有以下设置选项：



轴的含义

MD52206 \$MCS_AXIS_USAGE[n]		通道中轴的含义 [n]为通道轴号
= 0	没有特定含义	
= 1	刀具主轴（动力刀具）	

7.1 激活车削/铣削/钻削/磨削工艺

= 2	辅助主轴（动力刀具）
= 3	主主轴（车削）
= 4	主主轴的独立 C 轴（车削）
= 5	副主轴（车削）
= 6	副主轴的独立 C 轴（车削）
= 7	副主轴的直线轴（车削）
= 8	尾座（车削）
= 9	中心架（车削）
= 10	B 轴（车削）。
= 12	副主轴上的 B 轴（车削）
= 13	副主轴的横轴 X 轴（车削）

确定旋转方向

对于没有在 ToolCarrier 或 5 轴转换中设计的回转轴，通过以下通道专用机床数据输入其旋转方向。

MD52207 \$MCS_USAGE_ATTRIB[n]		通道中轴的属性 [n]为通道轴号
位 0	围绕第 1 几何轴旋转（旋转轴）	
位 1	围绕第 2 几何轴旋转（旋转轴）	
位 2	围绕第 3 几何轴旋转（旋转轴）	
位 3	旋转方向为逆时针方向（适用于旋转轴/ C 轴）	
位 4	显示的主轴/ C 轴旋转方向 M3 为逆转	
位 5	M3/M4 逆转（在主轴时） 该节的设置类似于 PLC-Bit DB38xx.DBX2001.6。	
位 6	显示旋转轴作为测量的补偿目标	
位 7	提供位置模式中的旋转轴	
位 8	提供用于毛坯接收的旋转轴（铣床上）	
位 9	主轴没有定位能力	
位 10	回转轴围绕第 1 几何轴旋转（仅用于位置模式）	
位 11	回转轴围绕第 2 几何轴旋转（仅用于位置模式）	
位 12	回转轴围绕第 3 几何轴旋转（仅用于位置模式）	

其他设置

MD52005 \$MCS_DISP_PLANE_MILL		铣削平面选择
= 0	在操作界面中选择平面（在“programGUIDE G-Code”下编程时的循环支持）	
= 17	G17 平面（缺省值）	
= 18	G18 平面	
= 19	G19 平面	

MD52006 \$MCS_DISP_PLANE_TURN		车削平面选择
= 17	G17 平面	
= 18	G18 平面（缺省值）	
= 19	G19 平面	

MD52212 \$MCS_FUNCTION_MASK_Tech		工艺通用的功能屏幕
位 0	通过 ToolCarrier 使能回转	
= 0	不使能平面回转、刀具回转	
= 1	使能平面回转、刀具回转	
位 1	沿软件限位开关的优化运行	
= 0	没有沿软件限位开关的优化运行	
= 1	沿软件限位开关的优化运行	
位 5	调用程序段搜索功能 SERUPRO。	
= 0	在程序段搜索循环 PROG_EVENT.SPF 中不调用循环 CYCLE207。	
= 1	在程序段搜索循环 PROG_EVENT.SPF 中调用用于 SERUPRO（CYCLE207）的循环。	

MD52240 \$MCS_NAME_TOOL_CHANGE_P ROG		用于 G 代码步骤的换刀程序
= 程序名	为换刀调用相应的程序。	

7.2 用于钻削的工艺循环

钻削工艺

使用下列通道专用的配置机床数据和循环设定数据，您可以对钻削工艺进行设置。

MD52216 \$MCS_FUNCTION_MASK_DRILL		钻削功能屏幕
位 0	攻丝 CYCLE84，工艺输入栏	
= 0	隐藏输入栏	
= 1	显示输入栏	
位 1	攻丝 CYCLE840，工艺输入栏	
= 0	隐藏输入栏	
= 1	显示输入栏	

SD55216 \$SCS_FUNCTION_MASK_DRILL_SET		钻削功能屏幕
位 1	镗孔 CYCLE86：主轴定位时考虑刀具平面旋转 注： 必须根据 DIN 设置主轴旋转方向 M3/ M4 和回转轴旋转方向。SPOS=0 时，显示平面的第 1 轴 + 方向的刀沿（G17 中为 X+）。	
= 0	主轴定位（SPOS）时不考虑刀具平面旋转。	
= 1	主轴定位（SPOS）时考虑刀具平面旋转。	
位 2	镗孔 CYCLE86：进行主轴定位时要考虑回转台系统（Toolcarrier）	
= 0	进行主轴定位时，不考虑回转台运动系统的分量（通过 CYCLE800，Toolcarrier 回转）。 注： 仅考虑旋转工件的机床运动系统的回转轴。在运动系统的初始位置中，此回转轴必须平行于刀具轴定向。 例如绕 Z 轴（G17）旋转的回转轴 C，回转数据组的回转轴矢量 V2xyz = 0, 0, -1。	
= 1	进行主轴定位时，考虑回转台运动系统的分量（通过 CYCLE800，Toolcarrier 回转）。	
位 3	攻丝 CYCLE84：主轴机床数据 MD31050 \$MA_DRIVE_AX_RATIO_DENOM 和 MD31060 \$MA_DRIVE_AX_RATIO_NUMERA 的监控	

位 4	攻丝 CYCLE84， 设置主轴机床数据 MD31050 \$MA_DRIVE_AX_RATIO_DENOM 和 MD31060 \$MA_DRIVE_AX_RATIO_NUMERA 的监控
= 0	无监控
= 1	监控
位 6	镗孔 CYCLE86： 调整主轴位置， 使之与刀具定向一致 可在通过 TOOLCARR 或 TRAORI 调用循环之前进行刀具定向。
= 1	位 1 和 位 2 不再生效。
位 7	镗孔 CYCLE86： 根据激活的镜像指令调整平面上的退刀方向
= 0	平面上的退刀按照编程进行。
= 1	镜像指令激活时， 平面上的退刀方向在循环内部自动根据该指令调整。

攻丝（CYCLE84 和 CYCLE840）

当通过通道专用配置机床数据 MD52216 \$MCS_FUNCTION_MASK_DRILL 隐藏工艺屏幕时， 下列通道专用循环设定数据中的设置生效：

SD55481 \$SCS_DRILL_TAPPING_SET_GG12[0]	设置攻丝 G 代码组 12： 精准停时程序段切换的属性
= 0	准停特性， 与循环调用之前相同（缺省值）。
= 1	G601
= 2	G602
= 3	G603

SD55482 \$SCS_DRILL_TAPPING_SET_GG21[0]	设置攻丝 G 代码组 21： 加速属性
= 0	加速特性， 与循环调用之前相同（缺省值）。
= 1	SOFT
= 2	BRISK
= 3	DRIVE

7.3 制造商循环

SD55483 \$SCS_DRILL_TAPPING_SET_GG24[0]	设置攻丝 G 代码组 24: 前馈控制
= 0	前馈控制, 与循环调用之前相同 (缺省值)。
= 1	FFWON
= 2	FFWOF

攻丝 (CYCLE84)

SD55484 \$SCS_DRILL_TAPPING_SET_MC[0]	MCALL 时的主轴运行
= 0	MCALL 时重新激活主轴运行 (缺省值)。
= 1	MCALL 时保持位置控制的主轴运行。

7.3 制造商循环

7.3.1 制造商循环

在循环软件包中, 我们还为您提供如下自定义循环:

CUST_TECHCYC.SPF	用来调整工艺循环的制造商循环。
CUST_800.SPF	用来调整平面回转与刀具回转功能 (CYCLE800) 的制造商循环, 参见章节 “Herstellerzyklus CUST_800.SPF (页 175)”。
CUST_832.SPF	用来调整高速设定功能 (CYCLE832) 的制造商循环, 参见章节 “制造商循环 CUST_832.SPF (页 186)”。
CUST_MEACYC.SPF	用来调整测量功能的制造商循环, 参见章节 “制造商与用户循环 CUST_MEACYC.SPF (页 190)”。
CUST_T	该循环用于在 SERUPRO 后跟踪 T 就绪状态。
CUST_M6	该循环用于 SERUPRO 后的换刀状态。
CUST_MULTICHAN	用于多通道车床的制造商循环
CUST_CLAMP	夹具的制造商循环

复制制造商循环

执行修改前，将循环从目录 NC 数据/循环/标准循环中复制到目录 NC 数据/循环/用户循环中。只有在该目录下循环修改才生效。



1. 选择操作区域“调试”。
2. 按下软键“系统数据”。
3. 打开目录“NC 数据/ 循环/标准循环”，选定制造商循环。
4. 按下软键“复制”，打开目录“NC 数据/ 循环/制造商循环”。
5. 按下软键“粘贴”。
该循环可供个性化匹配。

7.3.2 标准循环 PROG_EVENT.SPF

适用于 PROG_EVENT 的规定

PROG_EVENT.SPF 是一个标准循环，无法修改（无法访问）。

- 在制造商循环或用户循环目录中保存一个自己写入的循环 PROG_EVENT.SPF 时，该循环不会生效。
- 制造商专用的 prog_events 必须由以下循环映射，这些循环由 PROG_EVENT 调用：
 - CYCPE_MA.SPF 在内部 PROG_EVENT.SPF 结束时调用。
注释：最好使用 CYCPE_MA.SPF。
 - CYCPE1MA.SPF 在内部 PROG_EVENT.SPF 开始时调用。
- 程序段搜索：
缺省设置：换刀、主轴旋转方向/转速、回转轴的跟踪由内部 PROG_EVENT 执行，所需的机床数据会相应地作出缺省设置。制造商不能为上述操作使用 CYCPE_MA.SPF。
程序段搜索在 PROG_EVENT 应用中的特殊性：

机床数据		位
MD11450	\$MN_SEARCH_RUN_MODE	位 1=1
MD52212	\$MCS_FUNCTION_MASK_TECH	位 3=1

说明

CYCPE_MA.SPF 中的 \$P_PROG_EVENT==5 时，由于 PROG_EVENT.SPF 已经存在，不应编程 REPOSA，否则会在 SSL 上计算时出现错误。

7.3.3 用于换刀的制造商循环 CUST_T 和 CUST_M6**功能**

两个循环用于跟踪刀具准备（CUST_T）和 SERUPRO 后的换刀（CUST_M6）。此时针对特定刀套输出以下信息：

- 最后编程的换刀
- 最后编程的准备

步骤

两个循环在“SERUPRO-ENDE-ASUP”中调用，其还提供两个传输参数。

参数	含义
_THnr	已经为换刀或准备而设置过的刀套号或主主轴号。
_MTHnr	编程时生效的刀套或主主轴的编号。

循环自身不包含“机械逻辑”。只进行单纯的数据运算。

生效的刀套或主主轴会被保存。

通过传输参数测定 T 编程或 M6 编程时的情况。此情况决定，输出 T 和/或 M06，或者进入制造商循环。在输出 T 和 M6 后，或者从制造商循环返回后，会重新激活最初保存的刀套。

当存在相关询问（例如程序测试、程序段搜索等）时，可不受限制地调用换刀循环。

换刀循环

使用换刀循环时，必须注意以下提示：

说明

循环 CUST_T 使用语言指令 TCA (ToolChangeAbsolut) 调用刀具，因此可换入已被禁用的刀具。在 828D 上，可在 _TCA 中重新配置语言指令 TCA。可用 MD15710 \$MN_TCA_CYCLE_NAME 替换 TCA。

在机床上使用换刀操作时，必须注意以下事项：

- 必须明确地在规定位置上调用刀具准备循环。
- 如果通过循环 CUST_T 进行调用，则会跳过参数分析 (\$C_TS_PROG, ...)。或者可重新配置原始的语言指令 TCA。

参考文档

更多详细信息请参见“功能手册之基本功能”，章节：SERUPRO-Ende-ASUP

7.3.4 制造商循环 CUST_TECHCYC.SPF

功能

当机床制造商需要根据具体机床进行调整时（例如激活特定的 M 指令），可从工艺循环以及 ShopMill 循环和 Shop Turn 循环中调用 CUST_TECHCYC 循环。该循环比如可用于，在车床上进行切断加工时，驶入/驶出接料箱；对主轴进行定位或夹紧（仅限加工车间）。

因此在编程或执行必要的进程时可以使用 CUST_TECHCYC.SPF 循环。

调整 OEM 循环“CUST_TECHCYC.SPF”

利用参数_MODE 可以执行以下操作：

标记	操作	
_M1	主轴：	切换到 C 轴模式
_M2		切换到主轴模式
_M3		夹紧 C 轴
_M4		松开 C 轴
_M5		冲洗卡盘
_M6		闭合卡盘
_M7		主轴静止时打开卡盘
_M8		主轴旋转时打开卡盘
_M9		关闭冲洗卡盘
_M11	刀具主轴：	切换到 C 轴模式
_M12		切换到主轴模式
_M13		夹紧 C 轴
_M14		松开 C 轴
_M21	副主轴：	切换到 C 轴模式
_M22		切换到主轴模式
_M23		夹紧 C 轴
_M24		松开 C 轴
_M25		冲洗卡盘
_M26		闭合卡盘
_M27		主轴静止时打开卡盘
_M28		主轴旋转时打开卡盘
_M29		关闭冲洗卡盘
_M30	4. 在程序段查找结束后进行轴的定位（比如：副主轴、尾座和中心架）	
_M41	拖动动力刀具	

标记	操作	
_M42	脱开动力刀具	
_M61	切换到旋转加工	
_M62	切换到 C 柱面加工	
_M63	切换到 C 端面加工	
_M64	切换到 Y 柱面加工	
_M65	切换到 Y 端面加工	
_M66	切换到 B 端面加工（回转）	
_M67	撤销 B 端面加工（回转）	
_M68	在程序段查找结束后跳转到回转前的 ST 程序段	
_M100	接料箱：	在切断工件前就位
_M101		在切断工件时移出
_M102		在切断工件后归位
_M103	结束切断工件	
_M110	换刀	
_M111	在换刀周期结束后（有换刀）	
_M112	在换刀前回到返回平面	
_M113	在重新选择刀沿前	
_M114	在换刀周期结束后（无换刀）	
_M115	刀具预选后（ShopTurn）	
_M116	在换刀点上（ShopTurn）	
_M117	换刀 T 指令前	
_M120	定义: 主主轴与副主轴的耦合	
_M121	执行 COUPOF 后关闭位置控制	
_M122	定义: 主主轴/副主轴耦合	
_M123	执行 COUPOF 后关闭位置控制	
_M125	切换到主主轴	
_M126	切换到副主轴	

7.3 制造商循环

标记	操作	
_M131	程序头 (ShopTurn 程序)	
_M132	程序头结束 (ShopTurn 程序)	
_M135	程序循环末尾 (ShopTurn 程序)	
_M136	程序尾 (ShopTurn 程序)	
_M140	程序段搜索	换刀前
_M141		换刀后
_M142		换刀后 (ShopTurn 的新刀具)
_M150	2 通道切削:	主通道在“Balance Cutting” 前
_M151		主通道在“Balance Cutting” 后
_M152		副通道在“Balance Cutting” 前
_M153		副通道在“Balance Cutting” 后
_M154		在主通道中开始加工
_M155		在主通道中结束加工
_M156		在副通道中开始加工
_M157		在副通道中开始加工
_M160	多通道编程开始	
_M170	带主轴的开始程序块	
_M171	带主轴的结束程序块	
_M231	程序头 (ShopMill 程序)	
_M232	程序头结束 (ShopMill 程序)	
_M235	程序循环末尾 (ShopMill 程序)	
_M236	程序尾 (ShopMill 程序)	
_M300	镗孔 CYCLE86 根据刀具位置调整主轴 (SD55216 \$SCS_FUNCTION_MASK_DRILL_SET 位 6 = 1)	

7.3.5 用户循环 CUST_MULTICHAN

功能

循环 CUST_MULTICHAN 用于车床上的多通道编程。在含程序块的编程中，总是在程序块开始处调用此循环。

在循环中例如会将主主轴设置为程序块开始处设定的主轴。通过返回值 (= 1) 可设置跳过整个程序块。

CUST_MULTICHAN

参数	含义
_S_NR	主轴号为为主主轴设置的编号。
_RET	返回值
= 0	根据试运行模式执行程序块。
= 1	跳过整个程序块。

7.3.6 通过 programSYNC 调试多通道机床

本节描述的是您在通过 programSYNC（选件）和 ShopTurn 调试多通道机床时须注意的特性。



软件选件

使用 programSYNC 功能需要以下选件: "programSYNC".

参考文档

功能手册之基本功能分册

- 章节“H2: 向 PLC 输出的辅助功能” > “辅助功能”
- 章节 "K1: 运行方式组、通道、程序运行、复位特性" > “程序段搜索”
- 章节 "K2: 轴、坐标系、框架">“框架”

设置多通道视图

采用 AUTO 运行方式时，在“设置”→“多通道功能”下选择并排显示的通道。



图 7-1 多通道视图对话框

定义零点偏移

对于多通道机床，建议在贯穿所有通道的特定零点偏移中保留此类特殊轴。

两个通道同时加工一个工件，其前边缘值 Z 通常为 0 并通过可设定零点偏移定义。

副主轴上的零点位置也是通过可设定零点偏移定义的。但该位置同样取决于副主轴的运动轴 Z3 的位置且只存在一次。

通常，应在零点偏移中指定以下框架：

MD10616 \$MN_MAPPED_FRAME_MASK	使能框架映射
位 0 = 1	用于实际值设定和对刀的系统框架 (\$P_SETFR)

位 12 = 1	通道专用基准框架 (\$P_CHBFR)
位 13 = 1	可设定框架、零点偏移 (\$P_UIFR)

通过以下机床数据确定轴连接，其在贯穿所有通道的零点偏移中拥有同一个值：

MD32075 \$MA_MAPPED_FRAME[AXn]	轴框架映射
--------------------------------	-------

示例：

轴 AX2 = Z1（第一个通道中的 Z 轴）

轴 AX6 = Z3（副主轴的运动轴）

轴 AX11 = Z2（第二个通道中的 Z 轴）

MD32075 \$MA_MAPPED_FRAME[AX6] = "AX6"

MD32075 \$MA_MAPPED_FRAME[AX11] = "AX2"

副主轴的运动轴自连接 (Z3 = Z3)。通道的两根 Z 轴相连 (Z2 = Z1)。只能在一个方向上创建轴连接。

程序试运行

为支持程序试运行，“打开程序”→“试运行”下提供以下对话框：

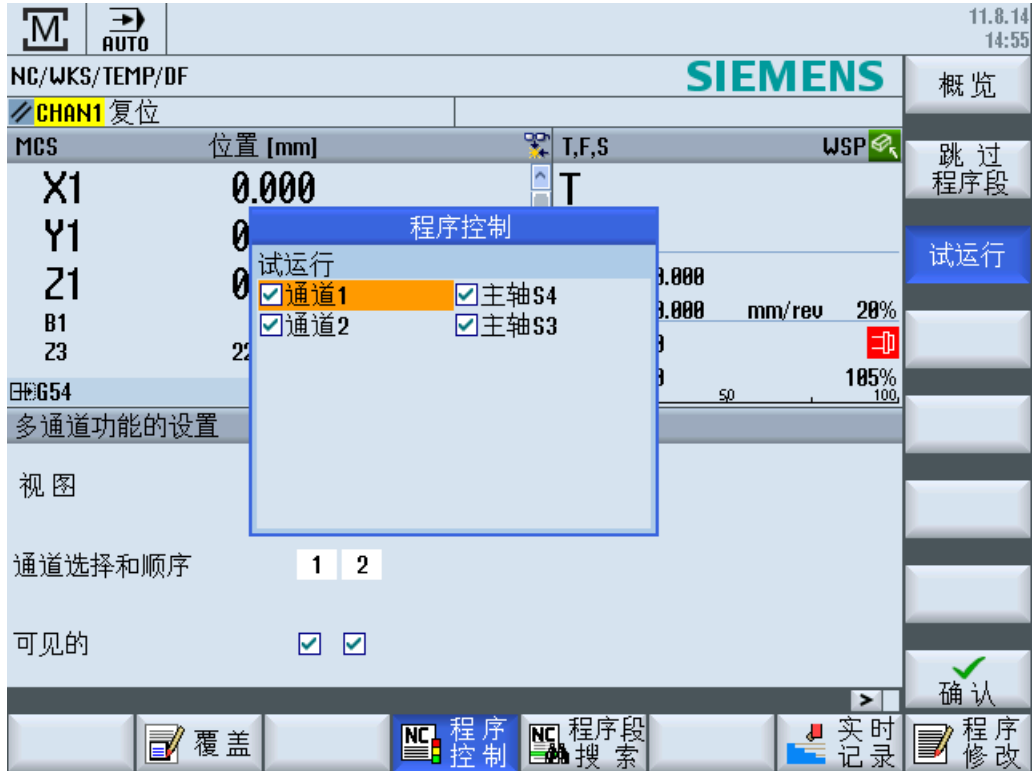


图 7-2 程序试运行

在该对话框中指定程序试运行所用的通道和主轴。

试运行小勾选框在 PLC 中按如下方式显示：

(小勾选框 = 加工 = 无试运行 = 0)

- 通道 1...n: DB330x.DBX1.7
- 主轴 n: DB380x.DBX1.3

说明

启动功能的调试示例见 SINUMERIK 828D 工具箱。

必须对所需的全部 M 指令进行分组。标准 M 指令（例如：M3、M4、M5、...）已经预定义了几个分组，参见以下机床数据：

MD22040 \$MC_AUXFU_PREDEF_GROUP	预定义辅助功能的分组指定
------------------------------------	--------------

用户自定义 M 指令必须在以下机床数据中定义：

MD22000 \$MC_AUXFU_ASSIGN_GROUP	辅助功能组
------------------------------------	-------

位 9、10、11 和 15 在以下机床数据中都有一个特定含义：

MD22035 \$MC_AUXFU_ASSIGN_SPEC	用户自定义辅助功能的输出特性说明
MD22080 \$MC_AUXFU_PREDEF_SPEC	预定义辅助功能的输出特性说明
位 9	在程序段搜索类型 5 (SERUPRO) 期间采集
位 10	在程序段搜索类型 5 (SERUPRO) 期间不输出
位 11	跨通道辅助功能 (SERUPRO)
位 15	在程序试运行（空运行）期间不输出

如果选中了一根主轴，系统会输出与该主轴相关的 M 指令（主轴旋转方向、主轴卡盘打开/闭合等）。PLC 中必须明确分辨（例如：通过 DB380x.DBX1.3 取消激活主轴）。

如果选中了一个通道，系统会输出与该通道相关的 M 指令（冷却液等）。为此，请使用位 15。也可在 PLC 中选择分辨率。

跨通道 M 指令（例如：M3、M4、M5）的典型设置：

位 9 = 1，位 10 = 1，位 11 = 1，位 15 = 0

通道专用 M 指令（例如：M7、M8、M9）的典型设置：

位 9 = 1，位 10 = 1，位 11 = 0，位 15 = 1

计算程序运行时间

为计算程序运行时间，可为以下机床数据中的 M 指令指定一个平均时间：

MD22037 \$MC_AUXFU_ASSIGN_SIM_TIME	辅助功能的应答时间，单位：毫秒
------------------------------------	-----------------

MD22090 \$MC_AUXFU_PREDEF_SIM_TIME	辅助功能的应答时间，单位：毫秒
---------------------------------------	-----------------

可通过以下指令增加时间：

- \$AC_PRTIME_M_INC，运行时间
- \$AC_PRTIME_A_INC，静止时间

此时，可单独考虑已知时间。

程序段搜索，带 SERUPRO

为进行快速程序段搜索，须确定以下值：

MD22600 \$MC_SERUPRO_SPEED_MODE	程序段搜索类型 5 时的速度
= 0	程序测试采用搜索速度/空运行速度：

MD22601 \$MC_SERUPRO_SPEED_FACTOR	程序段搜索类型 5 时的速度系数
= 10000	空运行进给与其相乘的系数。

模拟时也需用到这些值。

为在程序段搜索（带 SERUPRO）期间不输出 M 指令，必须将以下机床数据中的位 10 设为“1”。

MD22035 \$MC_AUXFU_ASSIGN_SPEC	用户自定义辅助功能的输出特性说明
位 10 = 1	在程序段搜索类型 5 (SERUPRO) 期间不输出

MD22080 \$MC_AUXFU_PREDEF_SPEC	预定义辅助功能的输出特性说明
位 10 = 1	在程序段搜索类型 5 (SERUPRO) 期间不输出

为启动零件程序，应设置以下机床数据：

MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK	忽略单程序段停止
位 16 = 1	单程序段与 SERUPRO 组合未生效

MD10708 \$MN_SERUPRO_MASK		程序段搜索模式
位 0 = 1	M0 与 SERUPRO 组合未生效	
位 4 = 1	倍率与 SERUPRO 组合未生效	

为调用用于 SERUPRO 过程的循环，应设置以下位：

MD52212 \$MCS_FUNCTION_MASK_Tech		工艺通用的功能屏幕
位 5 = 1	调用程序段搜索功能 SERUPRO	

可在以下用于 programGUIDE 和单通道 ShopTurn 的机床数据中单独设置用于搜索模式的选项：

MD51024 \$MNS_BLOCK_SEARCH_MODE_MASK_JS		用于可用搜索模式的位掩码 (ShopMill、单通道 ShopTurn)
位 0 = 1	程序段查找，带计算，不逼近断点	

MD51028 \$MNS_BLOCK_SEARCH_MODE_MASK		用于可用搜索模式的位掩码
= 33H		
位 0 = 1	程序段查找，带计算，不逼近断点	
位 1 = 1	程序段查找，带计算，逼近断点	
位 3 = 0	跳过 EXTCALL 程序	
位 4 = 1	程序段搜索，不带计算（仅针对 programGUIDE）	
位 5 = 1	程序段搜索，带试运行	

程序段搜索: 定位夹紧轴

在循环 PROGEVENT 结束时通过 REPOSA 自动定位所有轴。因此，夹紧轴的夹具必须自动通过 PLC 打开或由机床制造商事先在循环 CYCPE_MA 中对轴进行定位。

示例：

```
IF ($P_PROG_EVENT==5)AND($P_SEARCHL==5); 程序段搜索，带 SERUPRO
  IF ($AC_RETPOINT[B]<>$AA_IW[B]); 回转轴不在位置上
    M... ; 打开夹具
  GO B=$AC_RETPOINT[B]
```

7.3 制造商循环

```

M... ; 重新夹紧轴
ENDIF
ENDIF

```

程序段编程

进行多通道编程时建议使用程序段。为创建一个结构明确的程序，可将加工程序段组合在一起。

可将程序段分配给一根主轴。缺省设置下，主轴位于主主轴程序段的开头。该特性在循环 CUST_MULTICHAN 中配置且可由机床制造商根据需要进行补充。

坐标系

在工件上使用刀具时，刀具通常会有一个 180° 的偏移。由此得出用于通道的不同坐标系：

- 通道 1

MD52000 \$MCS_DISP_COORDINATE_SYSTEM	坐标系的位置
= 34	Z 轴向右, X 轴向上

- 通道 2

MD52000 \$MCS_DISP_COORDINATE_SYSTEM	坐标系的位置
= 33	Z 轴向右, X 轴向下

用于第二个通道的 C 轴的基本零点偏移中也可固定设置该 180° 偏移。因此，0° 时，在两个通道中的同一个工件位置上钻孔。

刀具列表

在刀具列表中，刀具位置是按照各个通道显示的。如果只有一个 TOA，未换入的刀具始终显示在第一个通道的坐标系中。

主主轴和副主轴

必须在两个通道中同时分配主轴。如果第一个通道中的主主轴是“左”主轴，则须确定第二个通道中的主主轴。

在以下机床数据中定义通道中轴的含义（例如：3 = 主主轴）：

MD52206 \$MCS_AXIS_USAGE	通道中轴的含义
= 3	主主轴（车削）

多通道数据

通过 programSYNC 编程时会创建一个任务列表。该任务列表中包含多通道数据，其中包括跨通道信息。多通道数据通过毛坯报告模拟信息。多通道数据中包含一个 WAIT 窗口，用于确保所参与通道中的所有程序在程序开头同步。

- 在 ShopTurn 程序中，系统会自动在程序头中调用多通道数据。
- 在 G 代码程序中，系统会在生成任务列表后自动通过 CYCLE208 添加多通道数据的相关调用。
如果无法自动添加该调用，可通过 MD51228 \$MNS_FUNCTION_MASK_TECH Bit 0 = 1 强行操作。之后，操作人员必须调用 CYCLE208 或通过毛坯报告模拟信息。

程序启动

启动通道中的程序有两种方法可选：

- 按下按键“NC 启动”，自动启动所有通道。
- 也可只启动第一个通道，然后再启动其余参与的通道。
针对此种方法，必须在制造商循环 CUST_TECHCYC 中对标记“_M160”进行相应的调整。

7.4 铣削

7.4.1 常规

标准循环的机床数据和设定数据的所有设置均适用于 ShopMill 铣削和 ShopTurn 车削。

7.4.2 用于铣削的工艺循环

铣削功能屏幕

SD52214 \$SCS_Function_MASK_MILL		铣削功能屏幕
位 0	使能圆柱体表面转换 (ShopMill)	
位 1	提供用于夹装毛坯的工作台 (铣床上) (只有机床上存在用于夹装毛坯的旋转台和固定工作台时, 才允许置位该位。)	
位 2	多边形圆盘铣刀/锯的特殊处理 (对于此类刀具, 初次进刀时应确保刀具的上缘正好碰到参考点 Z0)。	
位 3	使能内外加工 (ShopTurn)	
位 4	使能主轴夹紧 (C 轴) (ShopTurn)	
位 5	使能界面上刀具主轴的控制	
位 6	使能界面上旋转主轴的控制	

SD55214 \$SCS_FUNCTION_MASK_MILL_SET		铣削功能屏幕
位 0	同向运行铣削循环的基本设置。	
位 2	有和没有安全距离的铣削循环深度计算。	
= 0	在参考平面 + 安全距离与深度之间进行铣削循环的深度计算。	
= 1	进行深度计算时不包括安全距离 位 2 在下列铣削循环中有效: CYCLE61, CYCLE71, CYCLE76, CYCLE77, CYCLE79, CYCLE899, LONGHOLE, SLOT1, SLOT2, POCKET3, POCKET4。	

轮廓铣削 (CYCLE63)

SD55460 \$SCS_MILL_CONT_INITIAL_RAD_FI N		精加工逼近圆半径 精加工轮廓腔时控制逼近圆的半径。
= 0	选择半径, 使得起始点与精加工余量保持安全距离 (缺省值)。	
> 0	选择半径, 使得起始点与精加工余量保持由该通道专用设定数据所确定的值。	

多边形（CYCLE79），圆位置模式（HOLES2），圆弧槽（SLOT2）

SD55230 \$SCS_CIRCLE_RAPID_FEE D	以毫米/分钟为单位的快速进给率，用于圆弧槽之间或轮廓元素之间圆弧轨迹上的定位。
= 10000	

7.4.3 设置用于铣削的 ShopMill 循环

读入用于铣削的 ShopMill 循环

控制系统启动时，ShopMill 循环会自动读入以下目录： ../siemens/sinumerik/cycles/sm/prog

设置坐标系

MD52000 \$MCS_DISP_COORDINATE_SYSTE M	坐标系的位置
= 0	

示例：

=0 用于立式铣床

=16 用于卧式铣床，卧式镗床

MD52000 的详细说明请参见章节：

激活车削/铣削/钻削/磨削工艺 (页 79)

请根据以下机床数据设置铣削用的平面：

MD52005 \$MCS_DISP_PLANE_MILL	铣削平面选择
= 17	
= 0	在循环支持中
= 17	G17（缺省值）
= 18	G18
= 19	G19

模拟和同步记录

为了正确显示模拟和同步记录，可以按照下面章节中的说明设置机床数据：

设置用于模拟的坐标系 (页 69)

确定旋转方向

为了能够在 ShopMill 界面中正确显示旋转方向并且在编程 ShopMill 功能时执行正确的旋转方向，必须进行一些彼此协调相关的设置。此处的设置必须与机床上轴的实际旋转方向一致。

MD52207 \$MCS_USAGE_ATTRIB[n]		通道中轴的属性 [n] 为通道轴号
位 4	显示的旋转方向 M3 为逆时针旋转（主轴）	
位 5	M3 相当于回转轴负向旋转（主轴） 必须按照 PLC-Bit DB38xx.DBX2001.6 设置该位！	

其他通道专用的机床数据

MD52216 \$MCS_FUNCTION_MASK_DRILL		钻削功能屏幕
位 2	攻丝，带/无编码器 设置取决于机床上是否存在用于刀具主轴的编码器。	
= 0	只用于带编码器的攻丝	
= 1	攻丝，带/无编码器	
位 3	基于中心孔直径的恒定切削速度	
= 0	基于刀具直径的恒定切削速度	
= 1	基于中心孔直径的恒定切削速度	

MD52229 \$MCS_ENABLE_QUICK_M_CODES		使能 M 指令
= 0		
位 0	冷却液关闭（缺省值）	
位 1	冷却液 1 开	
位 2	冷却液 2 开	
位 3	冷却液 1 和 2 开	

MD52230 \$MCS_M_CODE_ALL_COOLANTS_ON FF	用于全部冷却液关闭的 M 代码
= 9	确定用于关闭冷却液的 M 功能，在换刀时输出。

MD52231 \$MCS_M_CODE_COOLANT_1_ON	用于冷却液 1 打开的 M 代码
= 8	确定用于冷却液 1 的 M 功能，在换刀时输出。

MD52232 \$MCS_M_CODE_COOLANT_2_ON	用于冷却液 2 打开的 M 代码
= 7	确定用于冷却液 2 的 M 功能，在换刀时输出。

MD52233 \$MCS_M_CODE_COOLANT_1_AND_2_ON	用于两个冷却液打开的 M 代码
= -1	确定用于冷却液 1 和 2 的 M 功能，在换刀时输出。

MD52281 \$MCS_TOOL_MCODE_FUNC_ON[]	刀具专用功能打开的 M 代码
= -1	不输出 M 功能。 功能的两个 M 指令“= -1”，则在界面中不显示相应的栏。
[0]	用于激活刀具专用功能 1 的 M 代码
[1]	用于激活刀具专用功能 2 的 M 代码
[2]	用于激活刀具专用功能 3 的 M 代码
[3]	用于激活刀具专用功能 4 的 M 代码

MD52282 \$MCS_TOOL_MCODE_FUNC_OFF[]	用于关闭刀具专用功能的 M 代码
= -1	输出 M 功能。 功能的两个 M 指令“= -1”，则在界面中显示相应的栏。
[0]	用于关闭刀具专用功能 1 的 M 代码
[1]	用于关闭刀具专用功能 2 的 M 代码
[2]	用于关闭刀具专用功能 3 的 M 代码
[3]	用于关闭刀具专用功能 4 的 M 代码

通道专用循环设定数据

SD55212 \$SCS_FUNCTION_MASK_Tech_SET	工艺通用的功能屏幕
= 6	
位 0	<p>刀具预选有效。</p> <p>在换刀后立即准备下一个刀具。</p> <p>说明：使用转塔刀架时必须将设定数据的值设置为“0”。</p>
位 1	在公制螺纹时自动计算螺纹深度。
位 2	从列表接收螺纹直径和螺纹深度。

7.4.4 柱面转换 (TRACYL)



软件选件

使用该功能需要下列软件选件：

“Transmit 与外表面转换”

前提条件

- 机床上至少要有一根回转轴。
- 铣刀必须在待加工圆柱体上径向定向。

功能

使用柱面转换功能可以进行下列槽加工：

- 圆柱体上的纵向槽
- 圆柱体上的横向槽
- 圆柱体上任意形状的槽

槽的形状要在展开的平坦柱面上编程。可以通过直线/圆弧，钻削或铣削循环或者通过轮廓铣削（自由轮廓编程）来进行编程。

圆柱体表面转换有两种表现方式：

1. 有槽壁补偿（打开）
2. 没有槽壁补偿（关闭）

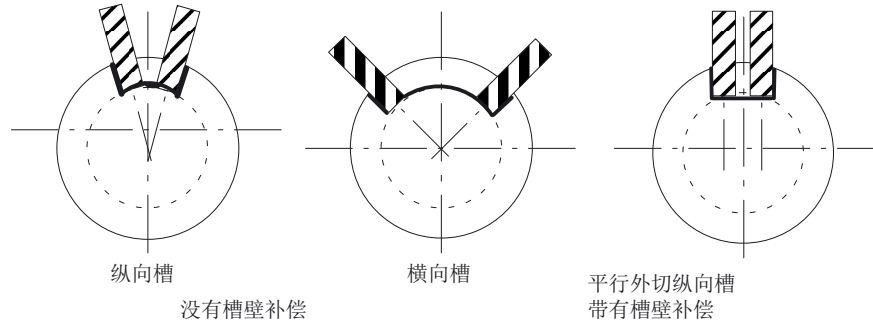


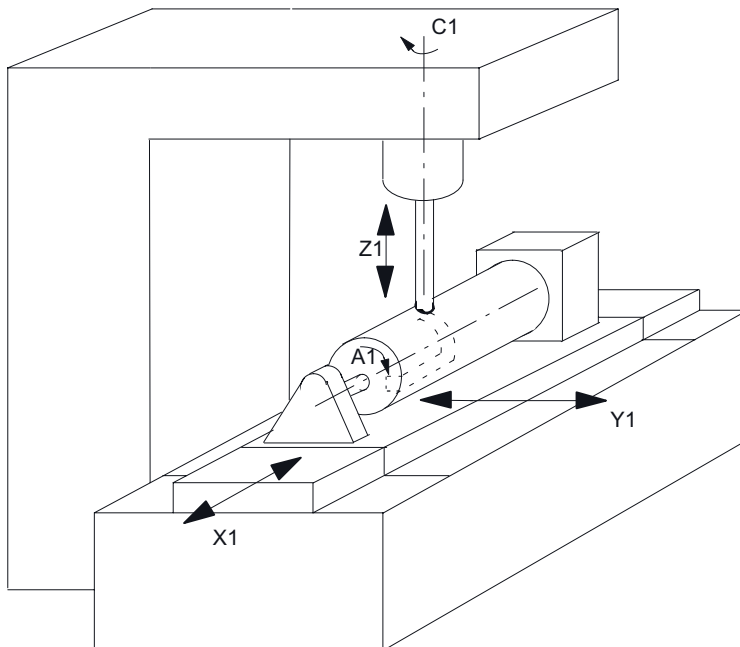
图 7-3 有和没有槽壁补偿的键槽

7.4.5 示例：带有轴配置 XYZ-AC 的铣床

示例：带轴配置 X1Y1Z1-A1C1 的铣床

以下述机床为例说明轴的配置方式：

轴配置



- X1 1. 加工平面的轴平行于回转轴
- Y1 2. 加工平面的轴
- Z1 进给轴（刀具轴），垂直于（径向）回转轴
- A1 回转轴
- C1 工作主轴

上述机床必须使用下列机床数据对两个数据组进行配置：

MD20070 \$MC_AXCONF_MACHAX_USED[4]	通道内有效的机床轴号
]	
= 5	通道轴的数量

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[]	通道内的通道轴名称
[0] = X1	通道轴 X1， 相当于通道轴 1
[1] = Y1	通道轴 Y1， 相当于通道轴 2
[2] = Z1	通道轴 Z1， 相当于通道轴 3
[3] = A1	通道轴 A1， 相当于通道轴 4
[4] = C1	通道轴 C1， 相当于通道轴 5

MD28082 \$MC_MM_SYSTEMFRAME_MASK	系统框架 (SRAM)
缺省设置, 不可修改	

用于转换的通用设置

MD10602 \$MN_FRAME_GEOAX_CHANGE_MO DE	切换几何尺寸标记时的框架
= 1	在几何轴切换时 (选择、取消 TRACYL) 重新计算当前的总框架 (零点偏移)。

通道内转换 1 的轴数据组

MD24100 \$MC_TRAFO_TYPE_1	定义通道内的转换 1。
= 512	转换类型 (512 = 没有槽壁补偿时的柱面转换)

MD24110 \$MC_TRAFO_AXES_IN_1	通道内转换 1 的轴分配	
[0] = 3	通道轴: 进给轴 (刀具轴), 垂直于 (径向) 回转轴	Z
[1] = 4	通道轴: 回转轴	A
[2] = 1	通道轴: 加工平面内的轴 1, 平行于回转轴	X
[3] = 2	通道轴: 加工平面的轴 2	Y

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1	转换 1 时几何轴与通道轴的分配。	
[0] = 1	通道轴: 第 1 几何轴	X
[1] = 4	通道轴: 第 2 几何轴	A
[2] = 3	通道轴: 第 3 几何轴	Z

MD24800 \$MC_TRACYL_ROT_AX_OFFSET_1	
= 0	用于 TRACYL 转换 1 的回转轴偏移。

MD24805 \$MC_TRACYL_ROT_AX_FRAME_1	
= 1	用于 TRACYL 转换 1 的回转轴轴向偏移。

MD24810 \$MC_TRACYL_ROT_SIGN_IS_PLUS_1	
= 1	用于 TRACYL 转换 1 的回转轴符号。

MD24820 \$MC_TRACYL_BASE_TOOL_1[]	
[0] = 0	用于 XYZ 中 TRACYL 转换 1 的基本刀具矢量
[1] = 0	
[2] = 0	

通道内转换 2 的轴数据组

MD24200 \$MC_TRAFO_TYPE_2	
= 513	转换类型（513 = 有槽壁补偿时的柱面转换）

MD24210 \$MC_TRAFO_AXES_IN_2		
[0] = 3	通道轴： 横向进给轴垂直于（径向）回转轴	Z
[1] = 4	通道轴： 回转轴	A
[2] = 1	通道轴： 加工平面内的轴 1，平行于回转轴	X
[3] = 2	通道轴： 加工平面的轴 2	Y

MD24220 \$MC_TRAFO_GEOAX_ASSIGN_TAB_2		
[0] = 1	通道轴： 第 1 几何轴	X
[1] = 4	通道轴： 第 2 几何轴	A
[2] = 3	通道轴： 第 3 几何轴	Z

MD24850 \$MC_TRACYL_ROT_AX_OFFSET_2	
= 0	用于 TRACYL 转换 2 的回转轴偏移

MD24855 \$MC_TRACYL_ROT_SIGN_IS_FRAME_2	
= 1	用于 TRACYL 转换 2 的轴向偏移

MD24860 \$MC_TRACYL_ROT_SIGN_IS_PLUS_2	
= 1	用于 TRACYL 转换 2 的回转轴符号

MD24870 \$MC_TRACYL_BASE_TOOL[]	
[0] = 0	用于 XYZ 中 TRACYL 转换 2 的基本刀具矢量
[1] = 0	
[2] = 0	

说明

转换功能所需的数据组

对于这两个数据组可以使用所提供转换（MD24100 \$MC_TRAFO_TYPE_1, MD24200 \$MC_TRAFO_TYPE_2 等）中的任意两个转换。这两个数据组不一定要前后相随。

数据组 1 必须始终用于“**没有**槽壁补偿的柱面转换”（= 512），而数据组 2 用于“**有**槽壁补偿的柱面转换”（= 513）。

7.4.6 用于多工件加工的 ShopMill 循环



软件选件

使用该功能需要下列软件选件：“ShopMill/ShopTurn”

功能

多工件加工功能可以连续加工多个工件，优化换刀过程。首先它可以节省加工准备时间。其次可以缩短换刀时间，因为只有在完成所有工位上工件的加工后，才会换入下一把刀具。

不同工位上可以运行同一个程序，也可以运行不同的程序。如果机床上使用的是平面夹具，则不需要进行其他设置。

如果是旋转式夹具，您需要根据夹具条件调整循环，使得在加工完一个工件后夹具可以将下一个工件旋转到加工位置，或者在多夹具机床上加工一个工件的同时使另一个夹具将下一个工件旋转到加工位置。

调试步骤

1. 修改目录 `cycle/sc/prog/templates_deu` 或 `cycle/sc/prog/templates_eng` 中的循环 `CLAMP.SPF`。
2. 将循环复制到用户或制造商目录中。

示例

此处使用包含 4 个工位的旋转式夹具（交换工作台）。回转轴 A 每旋转一次，便加工一个工件：

工件 1:	A = 0°
工件 2:	A = 90°
工件 3:	A = 180°
工件 4:	A = 270°

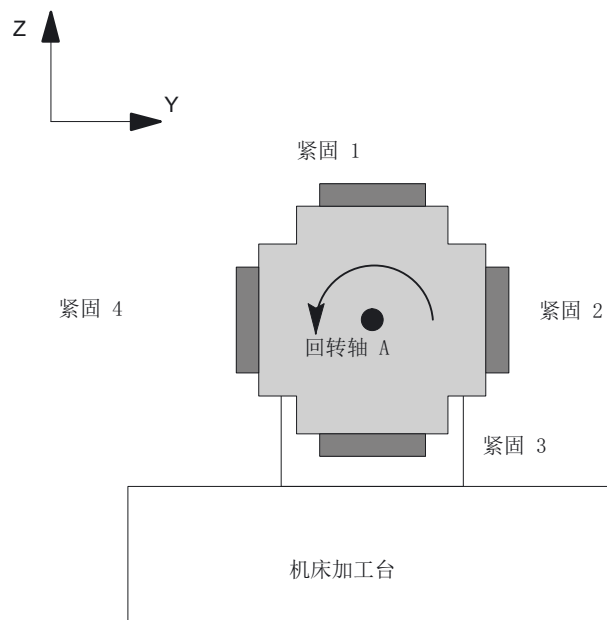


图 7-4 旋转式夹具（交换工作台）

接着必须按如下方式调整循环 CLAMP.SPF:

...

```
DEF INT _NV ; 帮助变量
```

```
-----
```

```
; 调整
```

```
;
```

```
IF _ACT==1
```

```
G0 A=DC(0)
```

```
ENDIF
```

```
;
```

```
IF _ACT==2
```

```
G0 A=DC(90)
```

```
ENDIF
```

```
;
```

```
IF _ACT==3
```

```
G0 A=DC(180)
```

7.5 车削

```

ENDIF
;
IF _ACT==4
GO A=DC(270)
ENDIF
;
;-----
_NV=_NPV+_ACT ; 计算当前零点偏移
N10 G[8]=_NV ; 此外不许进行计算
RET
    
```

7.5 车削

7.5.1 常规

标准循环的机床数据和设定数据的所有设置均适用于 ShopMill 铣削和 ShopTurn 车削。

7.5.2 设置用于车削的循环

设置坐标系

MD52000 \$MCS_DISP_COORDINATE_SYSTEM	坐标系的位置
= 0	缺省设置

示例:

0: 立式车床 (旋转车床)

19: 卧式车床, 在旋转中心前方进行加工

34: 卧式车床, 在旋转中心后方进行加工 (倾斜式床身车床)

MD52000 的详细说明请参见章节: “激活车削/铣削/钻削/磨削工艺 (页 79)”。

车削功能屏幕

通过下列通道专用的配置机床数据，您可以在车削功能屏幕中激活各种功能。

MD52218 \$MCS_FUNCTION_MASK_TURN		车削功能屏幕
位 0	手动测量刀具时激活放大镜	
位 1	激活切断工件时的工件托盘	
位 2	激活尾座	
位 3	激活界面上主主轴的控制	
位 4	激活界面上副主轴的控制	
位 5	使能界面上刀具主轴的控制	
位 6	使能双通道式切削的“Balance Cutting”	
位 7	使用 G1 进行轮廓切削时的退刀	
位 8	程序中主轴卡盘数据输入	
位 9	程序中附加的尾座数据输入	
位 12	关闭螺纹同步	
位 13	使用 CYCLE95 进行轮廓切削（828D programGUIDE，不带 Advanced Technology	

轴的含义

在机床数据 MD52206 \$MCS_AXIS_USAGE[n]中输入轴的含义。

MD52206 \$MCS_AXIS_USAGE[n]	通道中轴的含义 [n]为通道轴号

MD52206 的详细说明请参见章节：“激活车削/铣削/钻削/磨削工艺 (页 79)”

运行方向

进行以下设置，使操作界面和循环了解主轴和回转轴的旋转方向。

通过 MD32100 \$MA_AX_MOTION_DIR 设置 NC 回转轴的旋转方向。

MD32100 \$MA_AX_MOTION_DIR	运行方向
-1	方向反转
0, 1	无方向反转

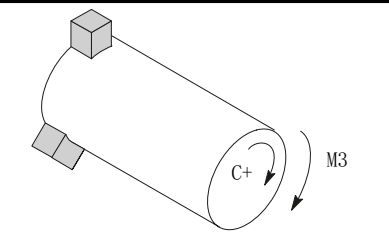
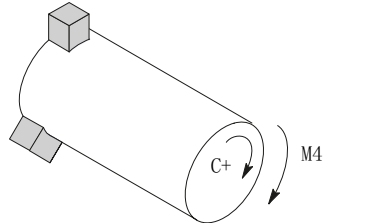
确定旋转方向

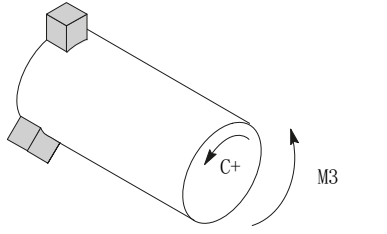
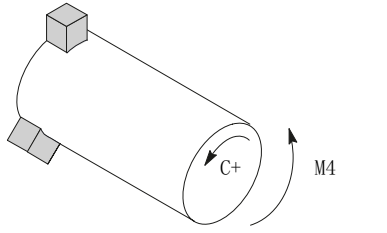
为了使主轴和 C 轴的旋转方向正确显示在操作界面中，并且在编程时会执行正确的旋转方向，一些设置必须相互协调，此处的设置必须与机床上主轴/C 轴的实际旋转方向一致。

在下列机床数据中进行设置：

MD52207 \$MCS_AXIS_USAGE_ATTRIB[n]	轴的属性 [n]通道轴号
位 3	显示的正旋转方向是逆时针旋转（回转轴） 旋转方向是从机床内部观察回转轴得出的方向（相当于操作员视角）。
位 4	显示的旋转方向 M3 为逆时针旋转（主轴） 该位可以从操作员的视角出发设置或者根据 DIN66025 设置。
位 5	旋转方向 M3 为回转轴负向（主轴）。 该节的设置类似于 PLC-Bit DB38xx.DBX2001.6。

由此可以得出下列用于主主轴的设置选项：

主主轴旋转方向	52207[n] 位 3 =	52207[n] 位 4 = *)	52207[n] 位 5 =	DB38xx DBX2001.6
	0	0	0	0
	0	1	1	1

主轴旋转方向	52207[n] 位 3 =	52207[n] 位 4 = *)	52207[n] 位 5 =	DB38xx DBX2001.6
	1	1	0	0
	1	0	1	1

说明

MD52207[n]位 4 = *)

位 4 可以从操作员视角出发设置（值如上），也可以依据 DIN（0 和 1 互调）设置。

尺寸

为了确定副主轴运行时的参考点，必须知道副主轴的尺寸。既可以在下列轴专用循环机床数据中，也可以在菜单“参数->设定数据->主轴卡盘数据”中输入尺寸。机床数据中的修改会被自动接收到菜单中，反之亦然。

MD53240 \$MAS_SPINDLE_PARAMETER[]		主轴卡盘数据
[0]	卡盘尺寸	
[1]	挡块尺寸	
[2]	卡爪尺寸	

MD53241 \$MAS_SPINDLE_CHUCK_TYPE[]		主轴卡盘类型
= 0	外部夹紧	
= 1	内部夹紧	

其他设置

在下列机床数据中 M 代码例如 M34 或 M1 = 34 用来定义主轴卡盘。制造商循环 CUST_TECHCYC.SPF 也使用下列机床数据中的 M 功能：

MD52250 \$MCS_M_CODE_CHUCK_OPEN[]		主轴静止时用于打开卡盘的 M 代码
[0]	主主轴	
[1]	副主轴	

MD52251 \$MCS_M_CODE_CHUCK_CLOSE_OPEN_ROT[]		主轴旋转时用于打开卡盘的 M 代码
[0]	主主轴	
[1]	副主轴	

MD52252 \$MCS_M_CODE_CHUCK_CLOSE[]		用于闭合卡盘的 M 代码
[0]	主主轴	
[1]	副主轴	

MD52253 \$MCS_M_CODE_TAILSTOCK_FORWARD[]		用于前面套筒的 M 代码
[0]	与主主轴相对的套筒	
[1]		

MD52254 \$MCS_M_TAILSTOCK_BACKWARD[]		用于后部套筒的 M 代码
[0]	与主主轴相对的套筒	
[1]	与主主轴相对的套筒	

收紧轮廓

SD55582 \$SCS_TURN_CONT_TRACE_ANGLE	车削轮廓：指定收紧轮廓的最小角度
= 5	指定刀沿与轮廓的夹角值，在车削轮廓时从此开始收紧轮廓，将剩余的材料清除（缺省值）。

7.5.3 设置用于车削的 ShopTurn 循环

MD52210 \$MCS_FUNCTION_MASK_DISP	显示功能屏幕
位 0 = 1	程序始终采用系统的单位制（缺省）
位 1 = 1	车削时的端面视图（缺省设置）
位 4 = 1	在 T,F,S 窗口中显示下一把刀具

MD52212 \$MCS_FUNCTION_MASK_TECH	工艺通用的功能屏幕
位 2	阶梯钻头的逼近逻辑
位 3	程序段搜索 - 用于 ShopMill/ShopTurn 的循环
= 0	在程序段搜索循环 PROG_EVENT.SPF 中，不调用循环 E_ASUP 或 F_S_ASUP。
= 1	在程序段搜索循环 PROG_EVENT.SPF 中，未调用循环 E_ASUP（在 ShopMill 中）或 F_S_ASUP（在 ShopTurn 中）。
位 4	通过循环的逼近逻辑
位 5	在程序段搜索循环 PROG_EVENT.SPF 中调用用于 SERUPRO（CYCLE207）的循环。
位 6	零点偏移值不可输入
位 7	检测程序中刀具寿命的过程（ShopMill / ShopTurn）
位 8	手动加工(ShopMill/ShopTurn)
位 9	通过软键选定/取消零点偏移补偿
位 11	切断钻削刀具和铣削刀具位置检查
位 12	正向 Z 偏移时的报警

MD52214 \$MCS_FUNCTION_MASK_MILL		铣削功能屏幕
位 3	在 ShopTurn 屏幕中激活了“内部/后部”加工，它们可以自行确定加工平面。	
位 4	如果想通过制造商循环 CUST_TECHCYC.SPF 实现“夹紧/松开主轴”功能，可以使用该机床数据在钻削与铣削屏幕中激活参数“夹紧/松开主轴”。	
= 0	参数“夹紧/松开主轴”不显示在钻削与铣削屏幕中。 当对加工有意义时，ShopTurn 会自动夹紧主轴。	
= 1	参数“夹紧/松开主轴”不显示在钻削与铣削屏幕中。 可以自行决定在哪种加工时夹紧主轴。	

MD52216 \$MCS_FUNCTION_MASK_DRILL		钻削功能屏幕
位 3	基于中心孔直径的恒定切削速度	
= 0	基于刀具直径的恒定切削速度	
= 1	基于中心孔直径的恒定切削速度	

MD52218 \$MCS_FUNCTION_MASK_TURN		车削功能屏幕
位 2	激活尾座	
位 5	使能界面上刀具主轴的控制	
位 6	使能双通道式切削的“Balance Cutting”	

MD52229 \$MCS_ENABLE_QUICK_M_CODES		使能快速 M 功能
位 0	冷却液 关	
位 1	冷却液 1 开	
位 2	冷却液 2 开	
位 3	冷却液 1 和 2 开	

MD52230 \$MCS_M_CODE_ALL_COOLANTS_OFF		用于全部冷却液关闭的 M 代码
= 9	使用该机床数据可以定义用于冷却液关闭的 M 功能，在换刀时输出该功能。	

MD52231 \$MCS_M_CODE_COOLANT_1_ON		用于冷却液 1 打开的 M 代码
= 8	使用该机床数据可以定义用于冷却液 1 的 M 功能，在换刀时输出该功能。	

MD52232 \$MCS_M_CODE_COOLANT_2_ON	用于冷却液 2 打开的 M 代码
= 7	使用该机床数据可以定义用于冷却液 2 的 M 功能，在换刀时输出该功能。

MD52233 \$MCS_M_CODE_COOLANT_1_AND_2_ON	用于两个冷却液打开的 M 代码
= -1	使用该机床数据可以定义用于冷却液 1 和 2 的 M 功能，在换刀时输出该功能。

SD55505 \$SCS_TURN_ROUGH_O_RELEASE_DIST	指定轮廓切削中加工外部时的回退距离
= 1	指定距离值，刀具在切削外角时按该值从轮廓退回。这不适用于轮廓切削（缺省值）。
= -1	距离为内部定义。

SD55506 \$SCS_TURN_ROUGH_I_RELEASE_DIST	指定轮廓切削内部加工时的回退距离。
= 0.5	指定距离值，刀具在切削内角时按该值从轮廓退回。这不适用于轮廓切削（缺省值）。
= -1	距离为内部定义。

7.5.4 设置副主轴



软件选件

为了在机床上使用副主轴，需要使用下列软件选件：

“通用耦合 CP-STATIC”或更高级耦合

功能

如果车床有副主轴，不必手动重新卡持工件，即可使用车削、钻削和铣削功能加工工件的正面和背面。

开始在背面加工之前，副主轴必须夹紧工件，将其从主轴上卸下，并移动到新的加工位置。

轴的含义

MD52206 \$MCS_AXIS_USAGE[n]	通道中轴的含义 [n]为通道轴号
-----------------------------	---------------------

MD52006 的详细说明请参见章节：“激活车削/铣削/钻削/磨削工艺 (页 79)”。

运行方向

进行以下设置，使操作界面和循环了解主轴和回转轴的旋转方向。

通过 MD32100 \$MA_AX_MOTION_DIR 设置 NC 回转轴的旋转方向。

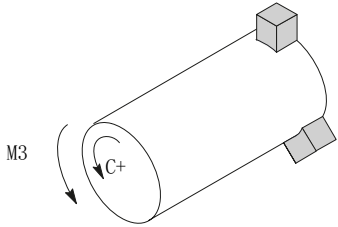
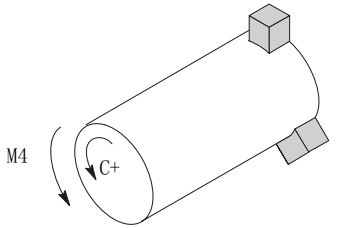
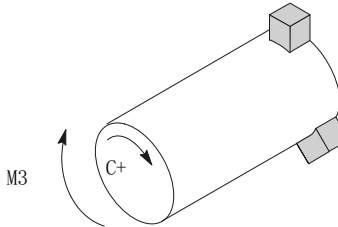
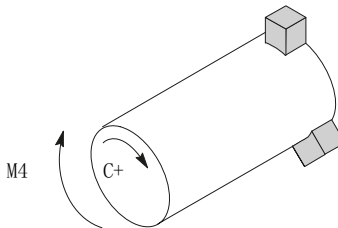
MD32100 \$MA_AX_MOTION_DIR	运行方向（非调节方向）
-1	方向反转
0, 1	无方向反转

确定旋转方向

在以下机床数据中设置平面的旋转方向：

MD52207 \$MCS_AXIS_USAGE_ATTRIB[n]	通道中轴的属性 [n] 通道轴号
位 3	显示的正旋转方向是逆时针旋转（回转轴） 旋转方向是从机床内部观察回转轴得出的方向（相当于操作员视角）。
位 4	显示的旋转方向 M3 为逆时针旋转（主轴） 该位可以从操作员的视角出发设置或者根据 DIN 设置。
位 5	旋转方向 M3 为回转轴负向（主轴）。 该节的设置类似于 PLC-Bit DB38xx.DBX2001.6。

由此可以得出下列用于副主轴的设置选项：

副主轴旋转方向	52207[n] 位 3 =	52207[n] 位 4 = *)	52207[n] 位 5 =	DB38xx. DBX2001.6
	1	1	0	0
	1	0	1	1
	0	0	0	0
	0	1	1	1

说明

MD52207[n]位 4 = *)

位 4 可以从操作员视角出发设置（值如上），也可以依据 DIN（0 和 1 互调）设置。

主轴和副主轴的旋转方向设置必须相同，要么都从操作员视角出发设置，要么都依据 DIN。

尺寸

为了确定副主轴运行时的参考点，必须知道副主轴的尺寸。既可以在下列轴专用循环机床数据中，也可以在菜单“参数”→“设定数据”→“主轴卡盘数据”中输入尺寸。机床数据中的修改会被自动接收到菜单中，反之亦然。

MD53240 \$MAS_SPINDLE_PARAMETER[]		主轴卡盘数据
[0]	卡盘尺寸	
[1]	挡块尺寸	
[2]	卡爪尺寸	

MD52241 \$MCS_SPINDLE_CHUCK_TYPE		主轴卡盘类型
= 0	外部夹紧	
= 1	内部夹紧	

7.5.5 设置 ShopTurn 中的副主轴



软件选件

为了在机床上使用副主轴，需要使用下列软件选件：

“通用耦合 CP-STATIC”或更高级耦合

副主轴在程序开始时运行到的目标位置，在通道专用的循环设定数据中确定：

SD55232 \$SCS_SUB_SPINDLE_REL_PO S	副主轴的 Z 轴退回位置
---------------------------------------	--------------

当固定挡块运行被激活时，以下各通道的设置数据才会起作用。

SD55550 \$SCS_TURN_FIXED_STOP_DIS T	运行到固定点的路径
SD55551 \$SCS_TURN_FIXED_STOP_FEE D	运行到固定点的进给率
SD55552 \$SCS_TURN_FIXED_STOP_FO RCE	运行到固定点的力，以 % 为单位。

在运行到固定点与夹紧之间副主轴可以回退一小段，用来抵消工件中的压缩应力。

SD55553 \$SCS_TURN_FIXED_STOP_RETRACTI ON	夹紧前至固定点的回退行程。
--	---------------

夹紧后可以切断工件。副主轴可以事先带着工件回退一小段，用来加大工件张应力。借此在切断时去除刀具负荷。

SD55543 \$SCS_TURN_PART_OFF_RETRACTION	切断之前的回退。
--	----------

切断之后可以执行出料检查并在车削时使用“运行至固定点”功能。可以通过下列通道专用循环设定数据设置或激活出料检查：

SD55540 \$SCS_TURN_PART_OFF_CTRL_DIST	用于出料检查的行程。
SD55541 \$SCS_TURN_PART_OFF_CTRL_FEED	用于出料检查的进给率
SD55542 \$SCS_TURN_PART_OFF_CTRL_FORC E	用于出料检查的力，以 % 为单位。

如果运行到固定点失败，则需要进行切断。会输出下列报警：

报警	报警文本
20091	轴 % 1 未到达固定点。
20094	轴 % 1 功能已被中断。

可以通过下列机床数据关闭报警显示：

MD37050 \$MA_FIXED_STOP_ALARM_MASK	激活到达固定点报警。
= 2	抑制报警 20091 和 20094。

可以在“机床数据”窗口的操作区“刀具零点”中设置这个轴专用的机床数据。

如在出料检查时未达到给定的力（即运行到固定点成功），则发出报警 61255 “切断故障：刀具受损？”。

说明

主轴夹紧时也可以使用“运行到固定点”功能（参见上面）。如果在这种情况下运行到固定点失败，当然也会输出报警。但不是报警 20091 和 20094，而会显示报警 61254 “运行到固定点时故障”。

机床制造商循环

如要执行下列动作之一，必须对机床制造商循环 CUST_TECHCYC.SPF 进行匹配。

- 主轴和 C 轴运行时在主主轴和副主轴间切换。
- 打开，关闭，清洗卡盘（主主轴/副主轴）。
- 修改用于主主轴与副主轴耦合的预设置。

7.5.6 用于车削的工艺循环

螺纹切削（CYCLE99）

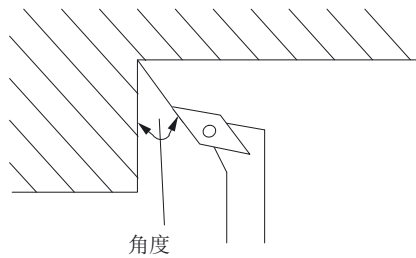
螺纹的正确加工需要设置机床数据 MD52207 \$MCS_AXIS_USAGE_ATTRIB[n] 位 3，参见章节：设置副主轴 (页 121)。

轮廓槽式车削 CYCLE930、轮廓车削 CYCLE950、拐角车削 CYCLE951、轮廓车削 CYCLE952

SD55500 \$SCS_TURN_FIN_FEED_PERCENT	在进行粗加工与精加工的综合加工时输入精加工进给率。百分比值等于参数 F（进给率）中的输入值。
= 100	100 % 精加工进给率

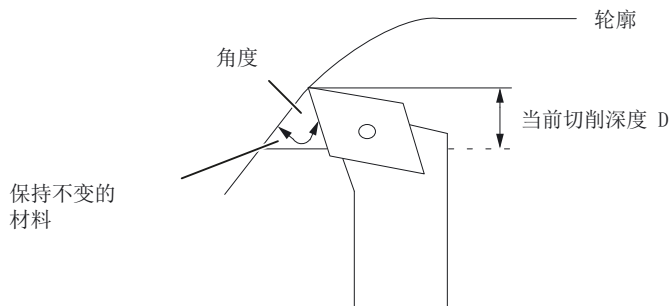
SD55510 \$SCS_TURN_GROOVE_DWELL_TIME	工艺“槽式车削”中刀具切入与撤回之间的暂停时间，在“切槽”中，底部的退回时间
= > 0	暂停时间，单位秒
= < 0	暂停时间，单位：主轴转数

SD55580 \$SCS_TURN_CONT_RELEASE_ANGLE	“轮廓车削”中粗加工时，刀具按该角度退刀。
= 45	退回角度为 45 度



SD55581 \$SCS_TURN_CONT_RELEASE_DISTANCE		粗加工轮廓时，采用退刀角度 SD55580 \$SCS_TURN_CONT_RELEASE_ANGLE 的退刀量。 这些数据都适用于轮廓车削、槽式车削和往复车削。
= 1	退刀量为 1 毫米或 1 英寸	

SD55582 \$SCS_TURN_CONT_TRACE_ANGLE		刀沿与轮廓的夹角，在车削轮廓时从此开始收紧轮廓将剩余材料清除。 如果该角度大于设定数据中所确定的角度，则刀具开始紧跟轮廓。
= 5	角度为 5 度	



SD55583 \$SCS_TURN_CONT_VARIABLE_DEPTH		轮廓车削中可变切削深度百分比值。 在“轮廓车削”和“轮廓车削余料”中，您可以选择可变的切削深度。
= 20	20 % 可变切削深度	

SD55584 \$SCS_TURN_CONT_BLANK_OFFSET	绕行障碍物时的安全距离，以免在走刀和退刀时出现碰撞。 这些数据都适用于轮廓车削、槽式车削和往复车削。
= 1	毛坯余量为 1 毫米或 1 英寸

SD55585 \$SCS_TURN_CONT_INTERRUPT_TIME	车削轮廓时的进给中断时间。 这些数据都适用于轮廓车削、槽式车削和往复车削。
= > 0	进给中断时间，以秒为单位
= < 0	进给中断时间，以转为单位
= 0	无中断

说明

只有当 SD55586 \$SCS_TURN_CONT_INTER_RETRACTION = 0 时，才分析通道专用循环设定数据 SD55585。

SD55586 \$SCS_TURN_CONT_INTER_RETRACTION	车削轮廓时进给中断的回退行程。 这些数据都适用于轮廓车削、槽式车削和往复车削。
= > 0	进给中断的回退行程 SD55585 \$SCS_TURN_CONT_INTERRUPT_TIME 无效。
= 0	无回退行程

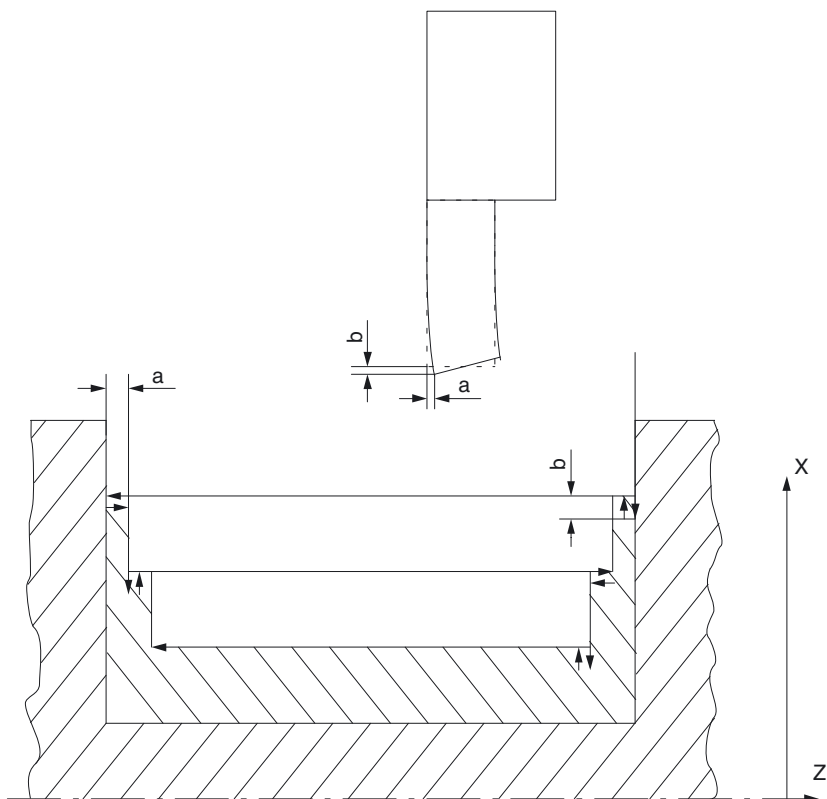
SD55587 \$SCS_TURN_CONT_MIN_REST_MAT_A X1	轮廓车削中轴 1（G18 为加工平面时为 Z 轴）方向上清理余料的极限值。 这些数据都适用于轮廓车削余料、槽式车削和往复车削。
50	如果待切削材料少于精加工余量的 50%，则不进行余料加工的粗加工步骤。在精加工时才开始切去材料。

示例：如果设定数据的设置为 50% 并且精加工余量为 0.5 毫米，则小于 0.25 毫米的余量在余料加工时不会被去除，而是在精加工时进行切除。如果某个加工步骤中的余料比设定数据中规定的尺寸小，系统会报告故障“没有材料”。也就是说，这一余料粗切步骤被跳过，因为没有余料可被切除。

SD55588 \$SCS_TURN_CONT_MIN_REST_MAT_A X2	轮廓车削中轴 2（G18 为加工平面时为 X 轴）方向上清理余料的极限值。 这些数据都适用于轮廓车削余料、槽式车削和往复车削。
50	如果待切削材料少于精加工余量的 50%，则余料加工时不进行粗加工步骤。在精加工时才开始切去材料。

由于“槽式车削”中的刀具弯曲，可能会在“轮廓车削”中无法完全逼近轮廓，下一次切削中需要缩短的侧面间距，在以下通道专用的循环设定数据中定义。

SD55595 \$SCS_TURN_CONT_TOOL_BEND_RET R	刀具弯曲产生的退刀距离
0.1	退刀距离为 0.1 毫米或 0.1 英寸



a 和上一次切削相隔的间距，SD55595

b “槽式车削”和“轮廓车削”之间的退刀，SD55596

由于“槽式车削”中的刀具弯曲，可能会在“轮廓车削”中形成更深的切削。“槽式车削”和“轮廓车削”之间的退刀距离可以在下列通道专用循环设定数据中定义。

SD55596 \$SCS_TURN_CONT_TURN_RETRACTI ON	车削前的退刀深度
= 0.1	退刀深度为 0.1 毫米或 0.1 英寸

7.5.7 车床的轴配置

通用配置

当车床上提供的是传动铣刀时，可以在该机床上另外设置下列功能：

- 柱面转换（TRACYL）（页 132）
- 端面加工（TRANSMIT）（页 136）

带 X1 和 Z1 轴、主主轴与刀具主轴的车床

对于带 X1 和 Z1 轴以及主主轴 (C1) 与刀具主轴 (SP1) 的车床可以象下面这样配置通道专用机床数据:

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[]		通道内的通道轴名称
[0] = X1	通道轴 X1	
[1] = Z1	通道轴 Z1	
[2] = C1	主主轴 C1	
[3] = SP1	刀具主轴 SP1	

带 X1 和 Z1 轴、主主轴、刀具主轴与副主轴的车床

对于带 X1 和 Z1 轴以及主主轴 (C1), 刀具主轴 (SP1) 与主主轴 (C2) 的车床可以象下面这样配置机床数据:

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[]		通道内的通道轴名称
[0] = X1	通道轴 X1	
[1] = Z1	通道轴 Z1	
[2] = C1	主主轴 C1	
[3] = SP1	刀具主轴 SP1	
[5] = C2	副主轴 C2	

带 X1 和 Z1 轴、主主轴、刀具主轴与 Y1 轴的车床

对于带 X1、Z1 和 Y1 轴以及主主轴 (C1) 与刀具主轴 (SP1) 的车床可以象下面这样配置机床数据:

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[]		通道内的通道轴名称
[0] = X1	通道轴 X1	
[1] = Z1	通道轴 Z1	
[2] = C1	主主轴 C1	
[3] = SP1	刀具主轴 WSP1	
[5] = Y1	通道轴 Y1	

7.5.8 柱面转换（TRACYL）



软件选件

使用该功能需要下列软件选件：

“传输与外表面转换”

功能

使用柱面转换（TRACYL）功能可以加工车削件的外表面。

设置

用于柱面转换的通用设置：

MD10602 \$MN_FRAME_GEOAX_CHANGE_M ODE	几何轴切换时的框架。
= 1	在几何轴切换时（选择、取消 TRACYL）重新计算当前的总框架（零点偏移）。

MD24040 \$MC_FRAME_ADAPT_MODE	匹配有效的框架
位 0 = 1	
位 1 = 1	
位 2 = 1	

MD28082 \$MC_MM_SYSTE M_FRAME_MASK	系统框架 (SRAM)
缺省设置，不可修改	

B 在设置功能时可以考虑下列通道专用机床数据：

MD24300 \$MC_TRAFO_TYPE_3	主轴柱面转换： 转换 3
= 512	没有槽壁补偿（无 Y1 轴）
= 513	有槽壁补偿（带 Y1 轴）：
= 514	有槽壁补偿与 Y1 补偿

MD24400 \$MC_TRAFO_TYPE_4	副主轴柱面转换： 转换 4
= 512	没有槽壁补偿（无 Y1 轴）
= 513	有槽壁补偿（带 Y1 轴）
= 514	有槽壁补偿与 Y1 补偿

说明

对于单个转换还必须分别设置其他机床数据。

设置 ShopTurn 中的柱面转换

另外要设置下列通道专用配置机床数据：

MD52214 \$MC_FUNCTION_MASK_MILL	铣削功能屏幕
位 3	在 ShopTurn 屏幕中激活“内部/后部”加工，它们可以自行确定加工平面。
位 4	如果想通过机床制造商循环 CUST_TECHCYC.SPF 实现“夹紧/松开主轴”功能，可以使用该机床数据在钻削与铣削屏幕中激活参数“夹紧/松开主轴”。
= 0	参数“夹紧/松开主轴”不显示在钻削与铣削屏幕中。 当对加工有意义时，ShopTurn 会自动夹紧主轴。
= 1	参数“夹紧/松开主轴”显示在钻削与铣削屏幕中。 操作员可以决定，在何种加工时夹紧主轴。

没有槽壁补偿的柱面转换

MD24300 \$MC_TRAFO_TYPE_3	在通道中定义第 3 个转换
= 512	TRACYL 主主轴。

MD24310 \$MC_TRAFO_AXES_IN_3[]	用于转换 3 的轴分配
[0] = 1	垂直于回转轴 X1
[1] = 3	回转轴（主主轴） C1
[2] = 2	平行于回转轴 Z1

MD24320 \$MC_TRAFO_GEOAX_ASSIGN_TAB_3[]	用于转换 3 的几何轴与通道轴的分配。
[0] = 1	第 1 通道轴 X1
[1] = 3	第 2 通道轴 Y1
[2] = 2	第 3 通道轴 Z1

MD24800 \$MC_TRACYL_ROT_AX_OFFSET_1	用于第 1 个 TRACYL 转换的回转轴偏移。
= 0	

MD24805 \$MC_TRACYL_ROT_AX_FRAME_1	在 TRACYL 时考虑回转轴的轴向偏移。
= 2	

MD24810 \$MC_TRACYL_ROT_SIGN_IS_PLUS_1	用于第 1 个 TRACYL 转换的回转轴符号
= 1	

MD24820 \$MC_TRACYL_BASE_TOOL_1[]	用于第 1 个 TRACYL 转换的基本刀具矢量。
[0] = 0	
[1] = 0	
[2] = 0	

有槽壁补偿的柱面转换

MD24300 \$MC_TRAFO_TYPE_3	在通道中定义第 3 个转换
= 513	TRACYL 主主轴

MD24310 \$MC_TRAFO_AXES_IN_3[]	用于转换 3 的轴分配
[0] = 1	垂直于回转轴 X1

[1] = 3	回转轴（主主轴） C1
[2] = 2	平行于回转轴 Z1
[3] = 6	平行于圆柱体外表面并垂直于回转轴 Z1

MD24320 \$MC_TRAFO_GEOAX_ASSIGN_TAB_3[]	用于转换 3 的几何轴与通道轴的分配。
[0] = 1	1. 通道轴 X1
[1] = 3	2. 通道轴 Y1
[2] = 2	3. 通道轴 Z1

MD24800 \$MC_TRACYL_ROT_AX_OFFSET_1	用于第 1 个 TRACYL 转换的回转轴偏移。
= 0	

MD24805 \$MC_TRACYL_ROT_AX_FRAME_1	在 TRACYL 时考虑回转轴的轴向偏移。
= 2	

MD24810 \$MC_TRACYL_ROT_SIGN_IS_PLUS_1	用于第 1 个 TRACYL 转换的回转轴符号
= 1	

MD24820 \$MC_TRACYL_BASE_TOOL_1[]	用于第 1 个 TRACYL 转换的基本刀具矢量。
[0] = 0	
[1] = 0	
[2] = 0	

7.5.9 端面加工（TRANSMIT）



软件选件

使用该功能需要下列软件选件：

“传输与外表面转换”

功能

使用端面加工（TRANSMIT）功能可以加工车削件的端面。

设置

用于转换的通用设置可以参见章节：“柱面转换（TRACYL）（页 132）”。

其他设置在下列通道专用机床数据中执行：

MD24100 \$MC_TRAFO_TYPE_1		主主轴端面加工：转换 1
= 256	无 Y1 轴的加工	
= 257	带 Y1 轴的加工	

MD24200 \$MC_TRAFO_TYPE_2		副主轴端面加工：转换 2
= 256	无 Y1 轴的加工	
= 257	带 Y1 轴的加工	

MD24110 \$MC_TRAFO_AXES_IN_1[]		通道内转换 1 的轴分配
[0] = 1	垂直于回转轴 X1	
[1] = 3	回转轴（主主轴） C1	
[2] = 2	平行于回转轴 Z1	

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[]		转换 1 时几何轴与通道轴的分配
[0] = 1	1. 通道轴 X1	

[1] = 3	2. 通道轴 Y1
[2] = 2	3. 通道轴 Z1

MD24900 \$MC_TRANSMIT_ROT_AX_OFFSET_1	
= 0	用于第 1 个 TRANSMIT 转换的回转轴偏移。

MD24905 \$MC_TRANSMIT_ROT_AX_FRAME_1	
= 2	在 TRANSMIT 1 时考虑回转轴的轴向偏移。

MD24910 \$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1	
= 0	用于第 1 个 TRANSMIT 转换的回转轴符号。

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1	
= 1	极点前/后的工作区极限，第 1 个 TRANSMIT 转换。

MD24920 \$MC_TRANSMIT_BASE_TOOL_1[]	用于第 1 个 TRANSMIT 转换的基本 刀具矢量。
[0] = 0	
[1] = 0	
[2] = 0	

带真正 Y 轴的 TRANSMIT 转换

MD24100 \$MC_TRAFO_TYPE_1	
= 257	定义通道内的转换 1： 主主轴 TRANSMIT 转换。

MD24110 \$MC_TRAFO_AXES_IN_1[]	通道内转换 1 的轴分配。
[0] = 1	垂直于回转轴 X1

[1] = 3	回转轴 C1
[2] = 2	平行于回转轴 Z1

MD24120 \$MC_TRAFO_GEOAX_ASSIGN_TAB_1[]	转换 1 时几何轴与通道轴的分配。
[0] = 1	1. 通道轴 X1
[1] = 3	2. 通道轴 Y1
[2] = 2	3. 通道轴 Z1

MD24900 \$MC_TRANSMIT_ROT_AX_OFFSET_1	
= 0	用于第 1 个 TRANSMIT 转换的回转轴偏移。

MD24905 \$MC_TRANSMIT_ROT_AX_FRAME_1	
= 2	在 TRANSMIT 1 时考虑回转轴的轴向偏移。

MD24910 \$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1	
= 0	用于第 1 个 TRANSMIT 转换的回转轴符号。

MD24911 \$MC_TRANSMIT_POLE_SIDE_FIX_1	
= 1	极点前/后的工作区极限，第 1 个 TRANSMIT 转换。

MD24920 \$MC_TRANSMIT_BASE_TOOL_1[]	用于第 1 个 TRANSMIT 转换的基本刀具矢量。
[0] = 0	
[1] = 0	
[2] = 0	

ShopTurn 中的端面加工

如要使用 ShopTurn 中的端面加工功能，还要另外设置下列通道专用配置机床数据：

MD52214 \$MCS_FUNCTION_MASK_MILL		功能屏幕 ShopTurn
位 3	在 ShopTurn 屏幕中激活“内部/后部”加工，它们可以自行确定加工平面。	
位 4	如果想通过制造商循环 CUST_TECHCYC.SPF 实现“夹紧/松开主轴”功能，可以使用该机床数据在钻削与铣削屏幕中激活参数“夹紧/松开主轴”。	
= 0	参数“夹紧/松开主轴”不显示在钻削与铣削屏幕中。当对加工有意义时，ShopTurn 会自动夹紧主轴。	
= 1	参数“夹紧/松开主轴”显示在钻削与铣削屏幕中。操作员可以决定，在何种加工时夹紧主轴。	

说明

端面加工自动集成在循环中，直线或圆弧除外。

对于这两个循环可以在“程序”操作区的“直线”或“圆弧”中选择刀具功能。

参考文档

关于端面加工的更多信息，请参见：

功能手册扩展功能；运动转换（M1）：TRANSMIT

7.5.10 斜置 Y 轴（TRAANG）



软件选件

为了在机床上使用“斜置 Y 轴”功能，需要使用下列软件选件：

“斜置轴”或“固定角斜置轴”

功能

如果车床具有“斜置 Y 轴”功能（即该轴不垂直于 X 轴和 Z 轴），就可以继续在直角坐标系中对加工进行编程。控制器借助斜置轴（TRAANG）功能将直角坐标转换为斜置轴的运行动作。

设置

此外必须通过机床数据对斜置轴（TRAANG）功能进行设置。

ShopTurn 中的斜置轴

另外，需要注意：

说明

在操作界面中斜置轴在设置后会自动集成到循环中。您可在斜置轴加工时在屏幕中选择加工平面“端面 Y1”或“柱面 Y1”并输入直角坐标。

参考文档

功能手册扩展功能：运动转换（M1）：TRAANG

示例

对于带 X1、Z1 和斜置 Y1 轴以及主主轴 (C1) 与刀具主轴 (SP1) 的车床可以象下面这样配置机床数据：

MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB	分配几何轴给通道轴。
[]	
[0] = 1	第 1 实际几何轴 X1 轴。
[1] = 0	第 2 实际几何轴 Y1 轴不存在。
[2] = 2	第 3 实际几何轴 Z1 轴。

MD20110 \$MC_RESET_MODE_MASK	
位 0 = 1	TRAANG 在引导启动后保持不变。
位 7 = 0	

MD20112 \$MC_START_MODE_MASK	
位 7 = 1	TRAANG 在“循环启动”后保持不变。

MD20118 \$MC_GEOAX_CHANGE_RESET	
= 1	允许自动更换几何轴。

MD20140 \$MC_TRAFO_RESET_VALUE	
= 5	TRAANG 在复位时始终有效。

MD20144 \$MC_TRAFO_MODE_MASK	
位 0 = 1	TRAANG 在后台运行（持续）并且不显示在操作界面中。

MD20070 \$MC_AXCONF_MACHAX_USED[4]	
= 5	通道轴 Y1 = 第 5 机床轴

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB[]	通道内的通道轴名称。
[0] = X1	第 1 通道轴 X1
[1] = Z1	第 2 通道轴 Z1
[2] = C1	第 3 通道轴 C1
[3] = SP1	第 4 通道轴 SP1
[4] = Y1	第 5 通道轴 Y1

用于斜置轴的数据程序段：

MD24430 \$MC_TRAFO_TYPE_5	
= 1024	转换 5： TRAANG

MD24432 \$MC_TRAFO_AXES_IN_5[]	用于转换 5 的轴分配。
[0] = 5	第 1 转换轴 = 通道轴， Y1
[1] = 1	第 2 转换轴 = 通道轴 X1
[2] = 2	第 3 转换轴 = 通道轴 Z1

MD24434 \$MC_TRAFO_GEOAX_ASSIGN_TAB_5[]		用于转换 5 的几何轴与通道轴的分配。
[0] = 1	第 1 轴 = 通道轴 X1	
[1] = 5	第 2 轴 = 通道轴 Y1	
[2] = 2	第 3 轴 = 通道轴 Z1	

MD24436 \$MC_TRAFO_INCLUDES_TOOL_5	
= 0	第 5 转换有效时的刀具处理

MD24700 \$MC_TRAANG_ANGLE_1	
= 55	第 1 和第 2 转换轴之间的角度 用于链接 (TRACON) 主主轴端面加工 (TRANSMIT) 和斜置轴 (TRAANG) 的数据组

用于链接 (TRACON) 主主轴端面加工 (TRANSMIT) 和斜置轴 (TRAANG) 的数据组:

MD24440 \$MC_TRAFO_TYPE_6	
= 8192	通道中第六个提供的转换类型

MD24444 \$MC_TRAFO_GEOAX_ASSIGN_TAB_6[]		用于转换 6 的几何轴与通道轴的分配。
[0] = 1	第 1 轴 = 通道轴 X1	
[1] = 3	第 2 轴 = 通道轴 Y1	
[2] = 2	第 3 轴 = 通道轴 Z1	

MD24995 \$MC_TRACON_CHAIN_1[]		转换链接
[0] = 1	用于链接的转换编号 TRANSMIT (主主轴)。	
[1] = 5	用于链接的转换编号 TRAANG 用于链接 (TRACON) 主主轴柱面转换 (TRACYL) 与斜置轴 (TRAANG) 的数据组。	

用于链接（TRACON）主主轴柱面转换（TRACYL）与斜置轴（TRAANG）的数据组：

MD24450 \$MC_TRAFO_TYPE_7	
= 8192	通道 TRACON 中转换 7 的类型。

MD24454 \$MC_TRAFO_GEOAX_ASSIGN_TAB_7[]		用于转换 7 的几何轴与通道轴的分配。
[0] = 1	第 1 轴 = 通道轴 X1	
[1] = 3	第 2 轴 = 通道轴 Y1	
[2] = 2	第 3 轴 = 通道轴 Z1	

MD24996 \$MC_TRACON_CHAIN_2[]		转换链接
[0] = 3	用于链接的转换编号 TRACYL（主主轴）。	
[1] = 5	用于链接的转换编号 TRAANG。	

7.5.11 X 零点不位于旋转中心

简介

在某些机床上，机床坐标系 (MCS) 中的 X 轴的零点不位于旋转主轴的中心。

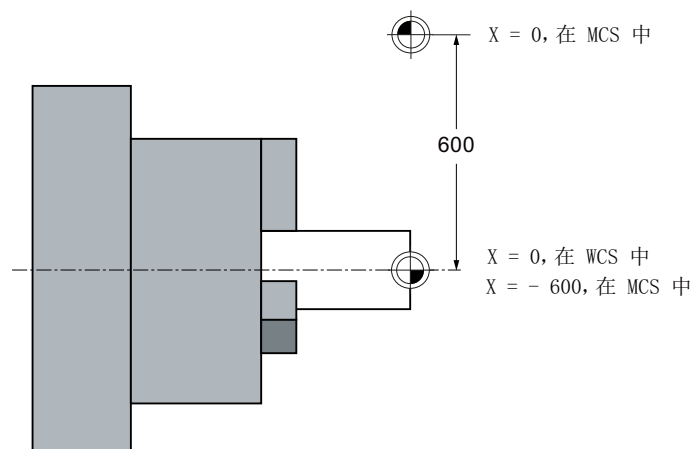


图 7-5 X 零点不位于旋转中心

如果机床坐标系 (MCS) 中的 X 轴的零点不位于旋转主轴的中心，则需进行一次特定的调试。

必须将主主轴的位置保存到以下机床数据中：

MD53220 \$MAS_AXIS_MCS_POSITION[0] [HSP]	MCS 中轴的位置
= [MCS 位置]	例如： - 600

此外，还须通过回转轴的位置报告转换：

MD24906 \$MC_TRANSMIT_BASE_TOOL_COMP_1	框架 TRANSMIT 1 中 BASE_TOOL 的补偿
位 0 = 1	通过 \$P_TRAFRAME 对 MD24920 \$MC_TRANSMIT_BASE_TOOL_1[0] 进行补偿。

MD24920 \$MC_TRANSMIT_BASE_TOOL_1 [0]	用于第 1 个 TRANSMIT 转换的基本刀具 矢量
= - 600	值，单位：毫米

必要时为其它转换输入相同的值 (TRANSMIT、TRACYL)。

也要注意以下机床数据：

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK	定义刀具参数的作用。
位 3 = 0	平面轴框架中的零点偏移不算作直径值。

副主轴的 Z 位置中的偏移

如果副主轴的零点在 MCS = 0 时不位于旋转中心，必须将偏移输入到以下机床数据中：

MD53220 \$MAS_AXIS_MCS_POSITION[2] [GSP]	MCS 中轴的位置
= [MCS 位置]	例如： 1000

固定副主轴

如果副主轴没有运行轴可用，必须将副主轴的位置输入到以下机床数据中：

MD53220 \$MAS_AXIS_MCS_POSITION[2] [GSP]	MCS 中轴的位置
= [MCS 位置]	例如：800

任意回转轴

如果机床上有一根没有通过 Toolcarrier 进一步定义的任意回转轴，必须将位置输入到以下机床数据中：

MD53220 \$MAS_AXIS_MCS_POSITION[n] [RAX]	MCS 中轴的位置 [n] 0 = X, 1 = Y, 2 = Z
---	--------------------------------------

7.6 磨削

常规

磨具的刀具类型为 400、410，校正刀具的刀具类型为 490。“新刀具”功能提供了这些类型选项。刀具列表中列出了刀具类型“砂轮”和“校正刀具”，按刀具类型编排。为生成砂轮补偿，设置人员会创建一个校正程序。根据复杂程度，可访问外部或内部代码发生器。之后，该程序被分配给一个砂轮。

基本型

校正是在 JOG 和 AUTO 运行方式下进行的，其中，在 AUTO 运行方式下，校正从零件程序中调用的。可通过手动模式或程序编辑器中的诊断屏幕设置校正过程。

基本包中包含预定义的砂轮轮廓。随附的屏幕和系统循环都支持校正。这些系统循环可与加工循环包分开使用。

往复循环

这些磨削循环可以通过磨削机床上的两根几何轴加工工件。

由于通过循环只能生成在每台机床上都可任意使用的摆动运动，因此，不设定机床类型。前提是有适用于程序加工的控制系统 SINUMERIK 及快速输入和输出。

循环软件包包含以下循环：

- CYCLE4071:反向点处带进给的纵向磨削
- CYCLE4072: 反向点处带进给以及测量控制器的纵向磨削
- CYCLE4073:带连续进给的纵向磨削
- CYCLE4074: 带连续进给以及测量控制器的纵向磨削
- CYCLE4075: 反向点处带进给的平面磨削
- CYCLE4077: 平面磨削：带测量控制器的切槽磨削
- CYCLE4078:带连续进给的平面磨削
- CYCLE4079:带间歇进给的平面磨削

参考文档

关于摆动循环的更多信息参见下列参考文献：

编程手册之工作准备分册，章节“循环外部编程”>“技术循环”

磨削功能屏幕

MD51840 \$MNS_GRIND_FUNCTION_MASK		磨削循环功能屏幕
位 0	修整器/砂轮采集时的参考点为零点偏移	
= 0	撤销作为参考点的零点偏移	
= 1	选择作为参考点的零点偏移	

MD52840 \$MCS_GRIND_FUNCTION_MASK		磨削循环功能屏幕
位 7	成型磨削使用 G1 退回	
= 0	成型磨削使用快速移动速度（G0）退回	
= 1	成型磨削使用进给率（G1）退回	

SD54840 \$SNS_GRIND_FUNCTION_MASK		磨削循环功能屏幕
SD55840 \$SCS_GRIND_FUNCTION_MASK		磨削循环功能屏幕

成型磨削 (CYCLE495)

SD55880 \$SCS_GRIND_CONT_RELEASE_ANGLE	回退角度
	该设定数据确定了成型磨削中砂轮从轮廓退出的角度。

SD55881 \$SCS_GRIND_CONT_RELEASE_DIST	回退量
	该设定数据确定了成型磨削中砂轮在两根轴上的退回量。

SD55884 \$SCS_GRIND_CONT_BLANK_OFFSET	毛坯余量
	该设定数据确定了成型磨削中砂轮从 G0 变为 G1 的位置点 (间距)。

7.7 回转

7.7.1 用于回转的工艺循环

前提条件

仅适用于铣削工艺。

为了确保回转功能 (CYCLE800) 正确运作, 必须进行机床运动链的调试。运动链储存在刀具参数 \$TC_CARR1 bis \$TC_CARR65 中。

说明

可以使用运动测量功能 (CYCLE996) 测定运动链的矢量。

为激活回转功能, 828D 上提供了一个刀架。

MD18088 \$MN_MM_NUM_TOOL_CARRIER	最大可定义刀架数量
	缺省设置, 不可修改。

MD28082 \$MC_MM_SYSTEM_FRAME_MASK	系统框架 (SRAM)
缺省设置，不可修改。	

激活回转功能

在操作界面上可以通过下列通道专用机床数据使能回转功能：

MD52212 \$MCS_FUNCTION_MASK_TE CH	工艺通用的功能屏幕
位 0 = 1	使能回转

配置输入窗口

可以通过下列通道专用循环设定数据来配置回转的输入屏幕。设定数据对所有相关的回效数据组有效。

SD55221 \$SCS_FUNKTION_MASK_SWIVEL_ SET	回转功能屏幕 CYCLE800
位 0	输入栏“回转，不”
= 0	隐藏
= 1	显示
位 1	空运行时的选择文本“Z”、“ZXY”或“固定点 1/2”
= 0	显示文本 Z = “Z”，显示文本 ZXY = “ZXY”
= 1	空运行时的选择文本“固定点 1”或“固定点 2”。 如果通过制造商循环“CUST_800.SPF”修改了退回方案“Z”或“ZXY”，则会显示中性文本“固定点 1”和“固定点 2”。
位 2	允许勾选“撤销”回转数据
= 0	如果不允许取消选择，在输入屏幕“回转”中不会显示选择栏“回转数据组”(TC)。
= 1	允许取消 另参见回转数据组 \$TC_CARR37 亿位
位 3	在“JOG 中回转”下显示当前有效的回转面。回转功能标记中的设置会针对所有的回转数据组生效。

位 4	运动初始设置（极位置）中的定位
= 0	机床运动极位置中输入值的分析
= 1	兼容性（在极位置回转时无输入值的分析）
位 6	JOG 中回转下不提供回转模式“直接”
位 7	回转平面：运动初始设置中的方向选择

其他设置

如要使用回转功能，至少要像下面一样设置下列机床数据，与缺省值有所偏差：

MD10602 \$MN_FRAME_GEOAX_CHANGE_MO DE	几何轴切换时的框架
= 1	几何轴切换时重新计算当前总框架（零点偏移），同时新几何轴的偏移、比例缩放和镜像生效。原有几何轴的旋转保留不变。

MD11450 \$MN_SEARCH_RUN_MODE	程序段搜索的设置
位 1 = 1	在程序段搜索之后激活 PROG_EVENT.SPF。这样在程序段搜索时可以对当前有效回转数据组回转轴进行预定位。

MD20110 \$MC_RESET_MODE_MASK	复位与零件程序结束后的控制系统初始设置
位 14 = 1	“基准框架”的系统特性

MD20112 \$MC_START_MODE_MASK	零件程序启动后的控制系统初始设置。
400H	

MD20126 \$MC_TOOL_CARRIER_RESET_VAL UE	复位时的有效刀架。
= 0	复位时无刀架生效。
= > 0 (n)	复位时编号为 n 的刀架生效。 在 CYCLE800 中描述 MD20126 CYCLE800() 与取消刀架选择相符（MD20126 = 0）。

MD20150 \$MC_GCODE_RESET_VALUES[]	G 功能组初始化设置
[41] = 1	TCOABS 上 G 功能组 42 的初始化设置
[51] = 2	PAROTOF 上 G 功能组 52 的初始化设置
[52] = 1	TOROTOF 上 G 功能组 53 的初始化设置
[52] = > 1	TOROT、TOROTY 或 TOROTX 上 G 功能组 53 的初始化设置 机床运动类型为“T”和“M”时的应用。 参见参数 \$TC_CARR34

说明

如需在 NC 复位后计算刀具方向的框架，可以将 MD20150 \$MC_GCODE_RESET_VALUES[52] 的值设置为 > 1。

应用：带切端面齿的机床运动

对带切端面齿运动（回转头/混合运动）的说明：

以有效的平面（G17, G18, G19）为基准，在 NCU 中为了计算 CYCLE800 切端面齿时补偿框架，编程指令 TOROT（或者 TOROTX、TOROTY）（G 功能组 53）。如果编程的旋转由于切端面齿而与回转轴可能的位置相偏离，在回转头运动中和混合运动中会产生一个补偿框架 \$P_TOOLFRAME（参见 HMI 有效的零偏/刀具基准详细信息）。

如果在复位或零件程序结束后需要保留补偿框架，则必须在通道专用机床数据中输入下列数值：

MD20150 \$MC_GCODE_RESET_VALUES[52]	G 功能组的复位属性
= 2	在 G17 时（TOROT）
= 3	在 G18 时（TOROTY）
= 4	在 G19 时（TOROTX）

MD20196 \$MC_TOCARR_ROTAX_MODE	可定向刀架的回转模式
位 0 = 1	带一根回转轴的回转数据组 应用于带 C 轴的回转台
位 1 = 1	带两根回转轴的回转数据组 应用于标准回转

MD20360 \$MC_TOOL_PARAMETER_DEF_MAS K		刀具参数的设置
位 10 = 1	在 T0 或 D0（无刀具）时方向矢量保持不变。 机床运动类型为“T”和“M”时的应用。 参见参数 \$TC_CARR34	
位 19 = 1	在 T0 或 D0（无刀具）时更改的方向矢量保持不变。 位 19 仅在位 10 = 1 时有效。	

MD21186 \$MC_TOCARR_ROT_OFFSET_FROM _FR		由回转轴零偏得出的可定向刀架回转轴偏移。
= 0	在 CYCLE800 中为回转轴零偏（NPV）中的数值重新计算工件坐标系 WCS。 如果在当前激活的回转数据组中输入了回转轴的零点偏移，回转数据组的初始设置可能会发生变化，工件坐标系的初始设置也可能因此变化。	
= 1	回转轴零偏中的数值被用作可定向刀架的偏移。工件坐标系 WCS 保持不变。	

不允许在程序中通过调用 CYCLE800 改写 MD21186。

如果每个通道有几个回转数据组，并且在更换回转头或者回转台时要求使能机床功能，则可以在 PLC 程序中在回转数据组更换时触发一个 M 指令。

MD22530 \$MC_TOCARR_CHANGE_M_CODE		用于回转数据组切换的 M 代码
= 0	无回转数据组切换	
< 0	M 代码 + 用于回转数据组切换的回转数据组的编号	

示例

输出 M 指令后，PLC 可以限制或取反主轴转速，或者夹紧或松开主轴。

通道 1 中回转数据组的数量	= 2
MD22530 \$MC_TOCARR_CHANGE_M_CODE	= -800
编程回转数据组 1（TCARR=1）	= M801
编程回转数据组 2（TCARR=2）	= M802

设置工件、刀具和回转台参考

使用下列机床数据可以设置系统框架的工件、刀具和回转台参考或者可以控制系统框架的特性。

应用：复位或上电后系统框架仍有效，这样可以将例如钻头在不发生碰撞的情况下从回转位置退回。

MD24006 \$MC_CHSFRAME_RESET_MASK		复位后有效的系统框架。
位 4	系统框架工件参考	
= 0	无效	
= 1	仍有效	

MD24007 \$MC_CHSFRAME_RESET_CLEAR_MASK		在复位后删除系统框架
位 4	系统框架工件参考	
= 0	不删除	
= 1	删除	

在测量或 JOG 手动方式回转时，工件参考在复位时必须有效而不能删除（级联测量）。

MD24006 \$MC_CHSFRAME_RESET_MASK		复位后有效的系统框架。
位 4 = 1	系统框架工件参考在复位后仍有效	

MD24007 \$MC_CHSFRAME_RESET_CLEAR_MASK		在复位后删除系统框架
位 4 = 0	系统框架工件参考在复位后不删除	

MD24008 \$MC_CHSFRAME_POWERON_MASK		系统框架在上电后重置。
位 2	系统框架回转台参考（PAROT）	
= 0	不重置	
= 1	重置	
位 3	系统框架刀具参考（TOROT, ..）	
= 0	不重置	
= 1	重置	

位 4	系统框架工件参考
= 0	不重置
= 1	重置

MD24080 \$MC_USER_FRAME_POWERON_MASK	用于可设定框架的设置。
位 0	
= 0	可设定零点偏移上电后无效。
= 1	最后有效的可设定零点偏移在上电后仍有效，当 MD20152 \$MC_GCODE_RESET_MODE[7] = 1 时。

应用：如果零点偏移为 G5xx，包括所有旋转在上电后都仍有效。

MD28082 \$MC_MM_SYSTEM_FRAME_MASK	设置系统框架（SRAM）。
位 2 = 1	旋转台参考（\$P_PARTFRAME）
位 3 = 1	刀具参考（\$P_TOOLFRAME）
位 4 = 1	工件参考（\$P_WPFRAME）

MD28083 \$MC_MM_SYSTEM_DATAFRAME_MASK	设置系统框架（SRAM）数据管理
位 2 = 1	旋转台参考（\$P_PARTFRAME）
位 3 = 1	刀具参考（\$P_TOOLFRAME）
位 4 = 1	工件参考（\$P_WPFRAME）

MD28085 \$MC_MM_LINK_TOA_UNIT	TO 单元至通道的分配 (SRAM)
<p>设置了多个通道时，可在机床数据中进行以下设置： 对应每个通道的 TO 单元的刀架数量（参见 MD18088）。</p> <p>-或者-</p> <p>将设置的所有刀架分配给一个通道。</p>	

用于回转数据组模数回转轴的轴向机床数据

MD30455 \$MA_MISC_FUNCTION_MASK	轴功能
位 0	模数回转轴编程

= 0	无模数回转轴编程（例如 0 到 359.999 度）
= 1	模数回转轴编程（例如 -180 到 + 180 度）
位 2	回转轴定位
= 0	和编程一样
= 1	按最短行程 应用：例如在 G90 带 DC 时回转轴 C 使用设置“位 2 = 1”按最短行程运行。 附加信息请参见章节“Herstellerzyklus CUST_800.SPF (页 175)”。

MD32010 \$MA_JOG_VELO_RAPID[AX] AX = 轴名称	JOG 方式下快速移动， 回转轴与加工轴，在 JOG 方式下回 转时运行。
= 10000	用于 JOG 回转的 JOG 运行方式下快速移动

SD42980 \$SC_TOFRAME_MODE	TOROT、PAROT 时的框架定义设 置
= 2000	回转（缺省值）

SD42974 \$SC_TOCARR_FINE_CORRECTION	精偏移 TCARR（回转数据组）。
= 0	回转数据组的矢量没有精偏移。
= 1	回转数据组的矢量有精偏移。 回转数据组的参数从 \$TC_CARR41[n] 起生效，n...回转数据组的编号。

在 JOG 运行方式下回转

通过以下循环机床数据可以显示或隐藏循环报警 62186 和 62187：

MD55410 \$MC_MILL_SWIVEL_ALARM_M ASK	激活故障分析 CYCLE800
位 0	激活故障 61186
= 0	隐藏故障 61186 “有效零点偏移 G%4 和基准（基本参考）包括旋转”（缺省 设置）。
= 1	显示故障 61186
位 1	激活故障 61187

= 0	隐藏故障 61187 “有效基准和基本参考 (G500) 包括旋转” (缺省设置)。
= 1	显示故障 61187

参考文档

- 功能说明 基本功能；刀具补偿 (W1)：使用 3 + 2 轴进行斜置加工
- 功能手册之特殊功能分册：多次转换 (F2)
- 测量循环编程手册：CYCLE996

7.7.2 用于机床运动识别的检查列表 CYCLE800

说明

按照 DIN 66217 或 ISO 841-2001 识别机床运动 (运动链)
检查列表不要求完整性。

- **转换有效的机床上，3 根线性轴是否构成一个直角坐标系？** 几何轴 XYZ
- **机床有多少回转运动？**
始终由 2 (或 1) 根回转轴与 3 根线性轴构成。
- **这是哪一种运动类型？**
回转头、回转台或回转头与回转台混合运动。
- **运动的回转轴叫什么名称？**
手动回转轴不需要也不一定要在 NC 中约定。
- **什么是回转数据组的第 1 或第 2 回转轴？**
规则：回转轴 2 是基于回转轴 1 设立的。混合运动时刀具所定向的轴始终为回转轴 1。
- **性线性轴和回转轴的运行方向正确吗？** 右手规则
规则：如线性轴或回转轴向工件移动，则轴的运动方向与回转轴矢量都要变化。

- 运动的初始化设置是什么样的？

从初始化设置得出刀具方向和平面 G17, G18, G19。

- 哪些回转轴绕着哪些坐标轴或机床轴旋转？

可以由此得出运动的回转轴矢量。

示例 1:

头运动, 回转轴 2 绕 Y 轴旋转 → 回转轴矢量 $V_{2xyz} = 0,1,0$

示例 2:

台运动, 回转轴 1 绕 X 轴旋转 → 回转轴矢量 $V_{1xyz} = -1,0,0$

7.7.3 调试回转数据（运动链）

定义回转数据组

必须给每个回转头、回转台或者每个回转头/回转台的组合创建一个回转数据组。

回转数据组由以下参数组成: $\$TC_CARR1[n]$ 到 $\$TC_CARR65[n]$

n = 回转数据组号

可以在“调试”操作区读入和读出回转数据组的参数 ($\$TC_CARR1[n]$ 到 $\$TC_CARR65[n]$)。也可以在 NC 程序（制造商循环）中使用相应的赋值进行编程。程序启动后回转数据组的参数立即生效。

偏移矢量 I1 到 I4

这些矢量始终包含三个部分, 它们是机床轴 (X, Y, Z) 的基准。运动链的位置由机床制造商测量, 它们对回转头/回转台（回转数据组）非常重要。偏移矢量 I1 到 I4 针对的是回转轴尚未回转时的状态（机床运动初始位置）。

不必完全实现所使用的机床运动。必须注意的是, 回转平面中的运行区域可能受限。如果要求一个机床运动仅带一个回转轴, 则必须讲它定义为第 1 个回转轴。

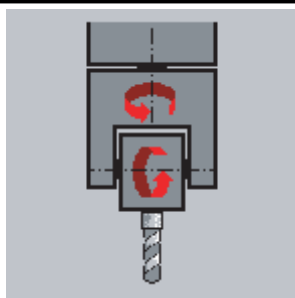
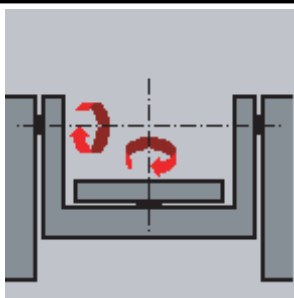
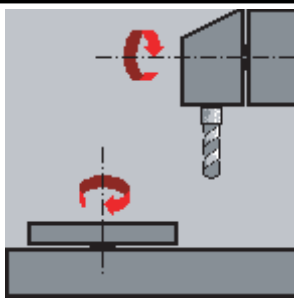
$\$TC_CARR1[n]$,	$\$TC_CARR2[n]$,	$\$TC_CARR3[n]$	偏移矢量 I1xyz
$\$TC_CARR4[n]$,	$\$TC_CARR5[n]$,	$\$TC_CARR6[n]$	偏移矢量 I2xyz
$\$TC_CARR15[n]$,	$\$TC_CARR16[n]$,	$\$TC_CARR17[n]$	偏移矢量 I3xyz
$\$TC_CARR18[n]$,	$\$TC_CARR19[n]$,	$\$TC_CARR20[n]$	偏移矢量 I4xyz

回转轴矢量 V1 和 V2

\$TC_CARR7[n],	\$TC_CARR8[n],	\$TC_CARR9[n]	回转轴矢量 V1xyz
\$TC_CARR10[n],	\$TC_CARR11[n],	\$TC_CARR12[n]	回转轴矢量 V2xyz

运动类型 \$TC_CARR23[n]

可以选择下列运动类型：

回转头（类型 T）	回转台（类型 P）	回转头+回转台（类型 M）
		
偏移矢量 I1	偏移矢量 I2	偏移矢量 I1
回转轴矢量 V1	回转轴矢量 V1	回转轴矢量 V1
偏移矢量 I2	偏移矢量 I3	偏移矢量 I2
回转轴矢量 V2	回转轴矢量 V2	偏移矢量 I3
偏移矢量 I3	偏移矢量 I4	回转轴矢量 V2
		偏移矢量 I4

回转数据组名称



图 7-6 用于输入回转数据组参数的对话框

每个回转数据组都有一个名称：\$TC_CARR34[n]

回转数据组名称中仅可以使用 NC 编程所允许的字符：A...Z，0...9 和 _ !

偏移矢量 I1~I4 \$TC_CARR1[n] ... \$TC_CARR20[n]

这些矢量始终包含三个部分，它们是机床轴 (X, Y, Z) 的基准。运动链的位置由机床制造商测量，它们对回转头/回转台 (回转数据组) 非常重要。

偏移矢量 I1 到 I4 针对的是回转轴尚未回转的状态 (机床运动初始位置)。不必完全实现所使用的机床运动。必须注意的是，回转平面中的运行区域可能受限。如果要求一个机床运动仅带一个回转轴，则必须讲它定义为第 1 个回转轴。

偏移矢量和回转轴矢量的符号由按照 ISO 841-2001 或者 DIN 66217 (右手规则) 的定义的轴方向得出。在使工件运动 (回转工作台) 的机床运动中，轴的方向反转。

不必强制这些偏移矢量指向回转轴的旋转中心。重要的是，这些矢量指向旋转方向的一个点 (交点)。

可以手调的回转轴（模式：手动）可以带/不带测量系统，在“简单机床”中可以使用。

- 回转头

I3	从刀具托架到第 2 个回转轴的旋转中心/交点的距离
I2	从回转轴 2 的旋转中心/交点到回转轴 1 的旋转中心/交点的距离
I1	如果回转头不可更换，矢量链关闭 $I1=-(I2+I3)$

- 回转台

I2	从机床参考点到第 1 回转轴的旋转中心/交点的距离
I3	从回转轴 1 的旋转中心/交点到回转轴 2 的旋转中心/交点的距离（或者到刀具安装的参考点）
I4	如果回转头不可更换，矢量链关闭 $I4=-(I2+I3)$

- 回转头/回转台（混合运动）

I2	从刀具托架到第 1 个回转轴的旋转中心/交点的距离
I1	如果回转头不可更换，矢量链关闭 $I1=-I2$
I3	从机床参考点到第 2 个回转轴的旋转中心/交点的距离（或者到刀具安装参考点）
I4	如果回转头不可更换，矢量链关闭 $I4=-I3$

偏移矢量的精偏移：

在调用回转功能 CYCLE800 或 NC 功能 TCARR=n 时，精偏移会对相应的基本矢量起到附加作用。

\$TC_CARR41[n] 到 \$TC_CARR60[n]

分配基本矢量到精偏移矢量：

I1	\$TC_CARR1...3[n]	到	\$TC_CARR41...43[n]
I2	\$TC_CARR4...6[n]	到	\$TC_CARR44...46[n]
I3	\$TC_CARR15...17[n]	到	\$TC_CARR55...57[n]
I4	\$TC_CARR18...20[n]	到	\$TC_CARR58...60[n]

激活精偏移要通过设定数据：

SD42974 \$SC_TOCARR_FINE_CORRECTIO N = 1	打开/关闭精偏移 TCARR
= 0	在激活可定位的刀架时不会考虑到精偏移值。
= 1	在激活可定位的刀架时会考虑到精偏移值。

回转轴名称



图 7-7 用于输入回转轴参数的对话框

优先选择以下的名称作为回转轴名称：

- 围绕加工轴 X 旋转的回转轴 → A
- 围绕加工轴 Y 旋转的回转轴 → B
- 围绕加工轴 Z 旋转的回转轴 → C

对于自动回转轴，必须在 $\$TC_CARR37[n]$ 的十位中输入对应 NC 回转轴的通道名称，在百位中输入自动模式。对于手动（手动调节）和半自动回转轴，可以使用任意轴名称（最多 6 个字符或数字）。

$\$TC_CARR35[n]$ 回转轴 1 名称

$\$TC_CARR36[n]$ 回转轴 2 名称

系统变量 $\$TC_CARR37[n]$

在操作区域“程序”→“其它”中的以下软键分配到了回转功能：

铣削工艺 → “回转平面” → “调整铣刀”

小数位的含义

为了显示回转的数值输入错误/选择错，可以使用下列显示方案：

$\$TC_CARR37[n]$	用于 CYCLE800 的输入屏幕显示方案	
小数位	含义	
个位	选择回转模式	
	0 =	轴方式
	1 =	轴方式 + 投影角度
	2 =	轴方式 + 投影角度 + 立体角度
	3 =	轴方式 + 直接
	4 =	轴方式 + 投影角度 + 直接
	5 =	轴方式 + 投影角度 + 立体角度 + 直接
十位	回转轴 1	
	0 =	自动
	1 =	手动
	2 =	半自动
百位	回转轴 2	
	0 =	自动
	1 =	手动
	2 =	半自动
千位	方向选择栏： 回转轴的方向参考	
	0 =	不显示仅有一个方案的运动的方向参考。方向选择（_DIR）“负”在循环调用 CYCLE800 中生成。

\$TC_CARR37[n]	用于 CYCLE800 的输入屏幕显示方案	
小数位	含义	
	3 =	回转轴 1 方向参考，方向选择“负”在运动的初始设置中。
	4 =	回转轴 2 方向参考，方向选择“负”在运动的初始设置中。
	5 =	不显示仅有一个方案的运动的方向参考。方向选择（_DIR）“正”在循环调用 CYCLE800 中生成。
	8 =	回转轴 1 方向参考，方向选择“正”在运动的初始设置中。
	9 =	回转轴 2 方向参考，方向选择“正”在运动的初始设置中。
		不允许值 1, 2, 6 和 7。
万位	选择栏“刀尖跟踪”	
	0 =	没有用于跟踪刀尖的输入栏显示。
	1 =	借助 TRAORI 跟踪刀尖。
	2 =	不跟踪刀尖。
	3 =	刀尖跟踪：该功能以选件“5 轴转换（TRAORI）”为前提条件。
十万位	预留	
百万位/千万位	回退选择栏	
	00 =	无回退
	01 =	回退 Z
	02 =	回退 Z, XY
	03 =	回退 Z 或 Z, XY
	04 =	刀具方向上回退，最大
	...	
	08 =	刀具方向上回退，增量
	...	
	15 =	回退 Z 或 Z, XY 或 刀具方向上最大值或者刀具方向上增量值
	\$TC_CARR38[n]	回退位置 X
	\$TC_CARR39[n]	回退位置 Y

\$TC_CARR37[n]	用于 CYCLE800 的输入屏幕显示方案		
小数位	含义		
		\$TC_CARR40[n]	回退位置 Z
亿	使能回转数据组 设置回转数据组的切换 只有在 ShopMill 或 ShopTurn 中需要手动或自动切换刀具 (→ 参见: CUST_800.spf, 标志_M2 到_M13)。 无论如何都必须使能回转数据组, 即设为大于等于 4 的值。		
	0 =	不使能回转数据组	
	4 =	使能回转数据组 自动切换回转数据组和刀具	
	5 =	使能回转数据组 自动更换回转数据组, 手动更换刀具	
	6 =	使能回转数据组 手动更换回转数据组, 自动更换刀具	
	7 =	使能回转数据组 手动切换回转数据组和刀具	

回转之前几何轴的空运行

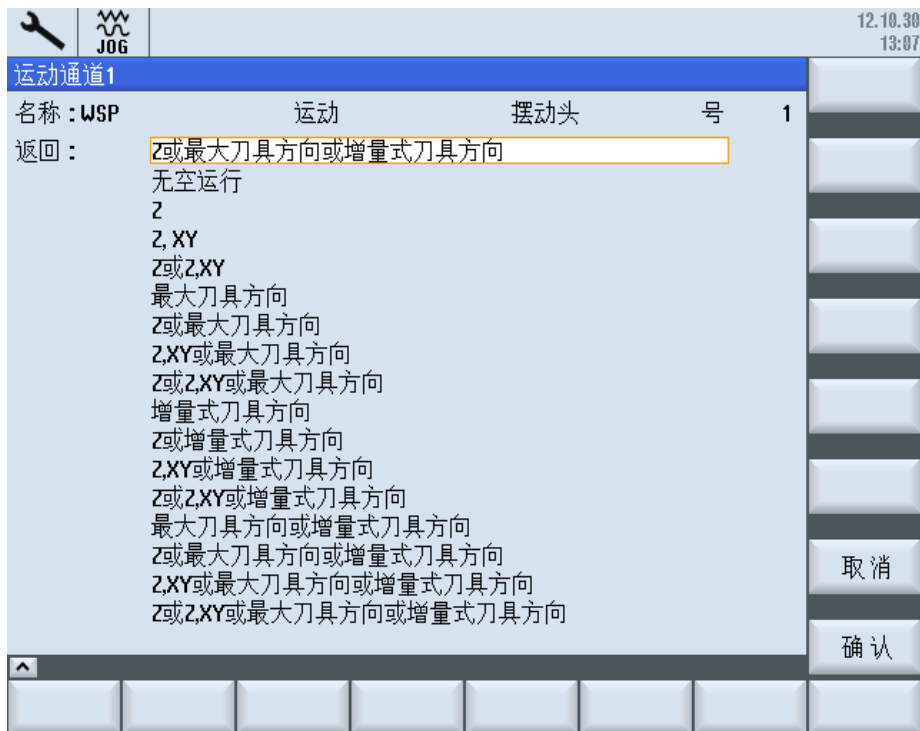


图 7-8 选择空运行方式的对话框

在输入栏中选择回退方式。该方式在系统变量的百万位和千万位中确定。

- Z 轴的回退
- Z 或 Z, XY 轴的回退
- 刀具方向的最大或增量回退

注意**碰撞监测**

在运行刀具轴时要注意，避开回转时刀具和工件碰撞。

Z 轴的回退或者 Z, XY 轴的回退以绝对加工位置运行到参数 \$TC_CARR38[n] bis \$TC_CARR40[n] 中的数值上。

\$TC_CARR38[n] 回退位置 X

\$TC_CARR39[n] 回退位置 Y

\$TC_CARR40[n] 回退位置 Z

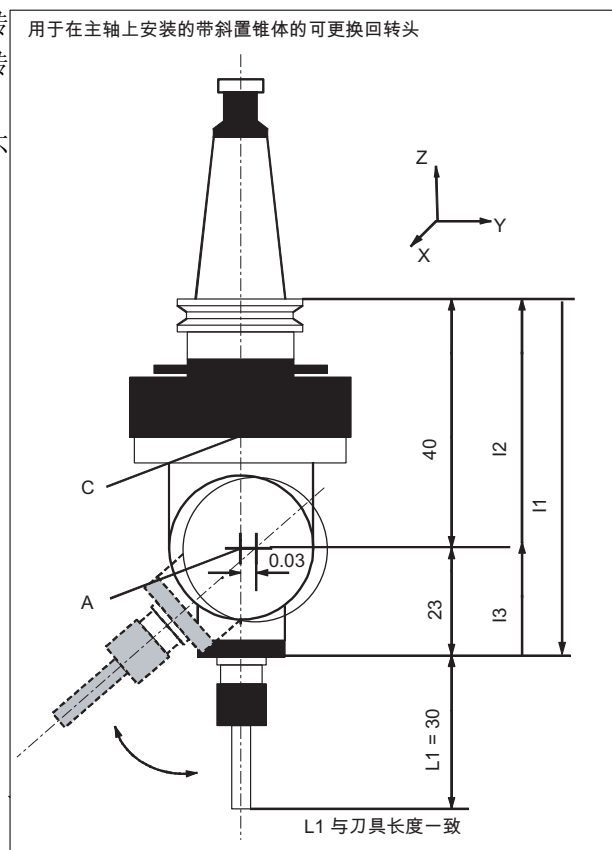
在 Herstellerzyklus CUST_800.SPF (页 175) 中修改空运行的类型。

7.7.4 回转开机调试机床运动的示例

示例 1: 回转头 1“HEAD_1”

- 手动回转轴 1 (C)，围绕 Z 轴旋转
- 手动回转轴 2 (A)，围绕 X 轴旋转
- 可手调并可更换的手动回转头

矢量取决于运动的初始位置（图纸不按照比例）



表格 7-1 调试软键“回转”，运动（示例 1）

运动	回转头		HEAD_1
退回	Z		
	X	Y	Z
			200.000
偏移矢量 I1	0.000	0.030	-63.000
回转轴矢量 V1	0.000	0.000	1.000
偏移矢量 I2	0.000	0.000	40.000
回转轴矢量 V2	1.000	0.000	0.000
偏移矢量 I3	0.000	-0.030	23.000
显示方案			
回转模式	轴方式		
方向参考	回转轴 2		
跟踪刀具	否		
回转轴			
回转轴 1	C	模式	手动
角度范围	0.000		360.000
回转轴 2	A	模式	手动
角度范围	-15.000		100.000

示例 2：回转头 2“HEAD_2”

回转轴矢量 V1: 回转轴 B 围绕 Y 旋转

回转轴矢量 V2: 回转轴 C 围绕 Y 和 Z 旋转

偏移矢量 I1: 在固定安装回转头上闭合矢量链
 $I1 = -(I2 + I3)$

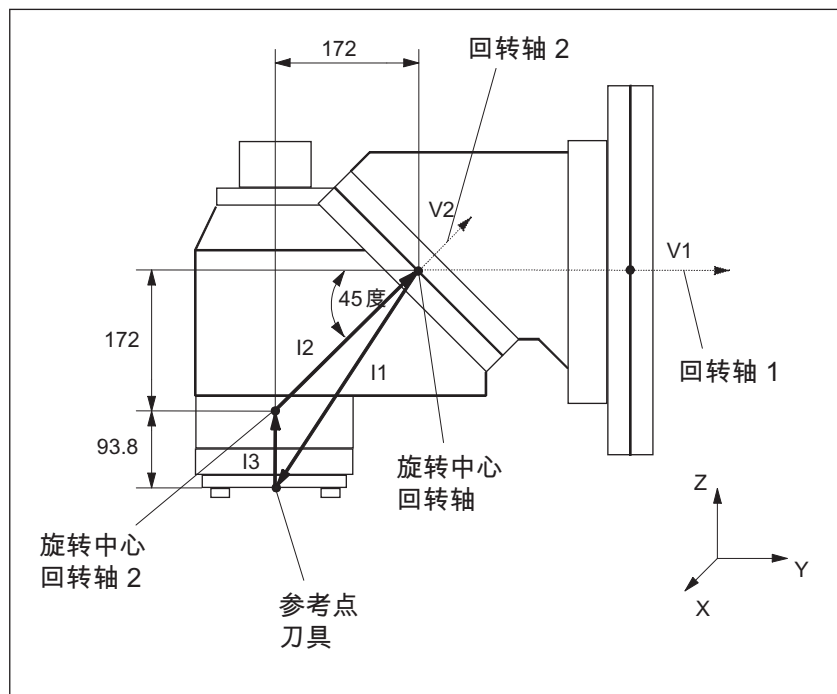
偏移矢量 I2: 回转轴 1 的旋转中心和回转轴 2 的旋转中心之间的距离

偏移矢量 I3: 刀具参考点和回转轴 2 的旋转中心之间的距离

带切端面齿的万向回转头（可手调）

矢量取决于运动初始位置

如果已固定安装回转头，则矢量链闭合（参见 I1）



表格 7-2 调试软键“回转”，运动（示例 2）

运动	回转头		HEAD_2
退回	Z	刀具方向	最大+增量
	X	Y	Z
			200.000
偏移矢量 I1	0.000	-172.000	-265.800
回转轴矢量 V1	0.000	1.000	0.000
偏移矢量 I2	0.000	172.000	172.000
回转轴矢量 V2	0.000	1.000 ¹⁾	1.000 ¹⁾
偏移矢量 I3	0.000	0.000	93.800
显示方案			
回转模式	轴方式		
方向参考	回转轴 2		
回转轴			
回转轴 1	B	模式	手动
角度范围	0.000		360.000

运动	回转头		HEAD_2
运动偏移值	0.000		
切端面齿	是	角度单位	1.000
回转轴 2	C	模式	手动
角度范围	0.000		180.000
运动偏移值	0.000		

回转轴 1 和 2 的旋转中心基准点可以移动到旋转线，不必与机械旋转中心一致。

1) 计算回转轴矢量 V2: 45 度角

$$V2Y = \sin(45) = 0.7071$$

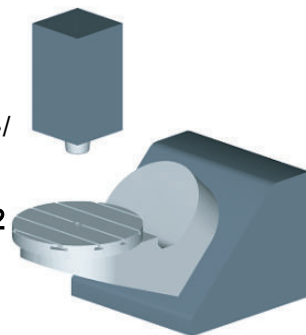
$$V2z = \cos(45) = 0.7071$$

V2Y 和 V2z 可以标定为 1。

示例 3: 万向工作台“TABLE_45”

矢量取决于运动初始位置

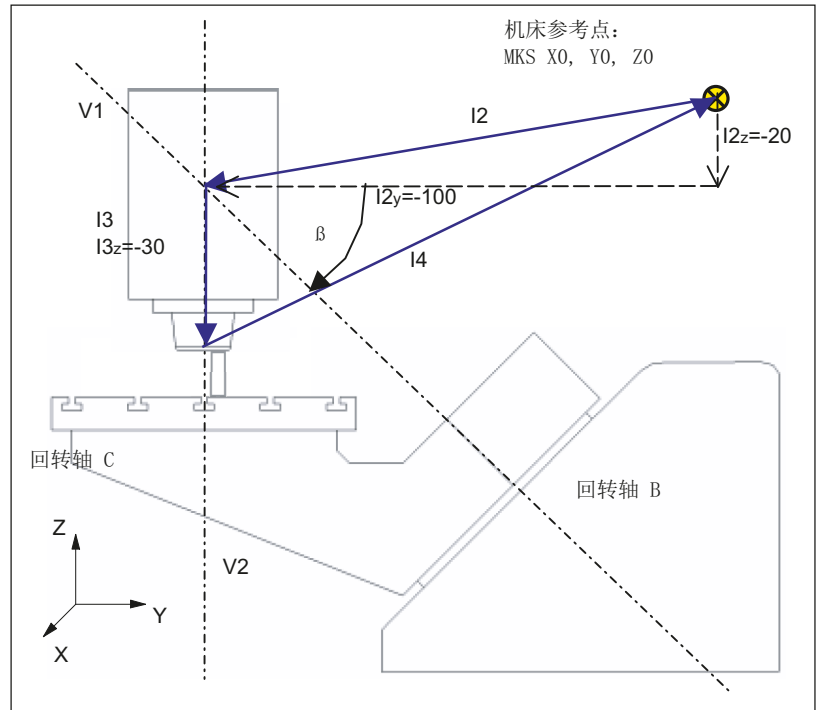
- 回转轴矢量 V1: 回转轴 B 围绕 Y 和 Z 旋转
- 回转轴矢量 V2: 回转轴 C 围绕 Z 轴旋转
- 偏移矢量 I2: 从机床基准点到**回转轴 1**的旋转中心/交点的距离
- 偏移矢量 I3: **回转轴 1**的旋转中心/交点和**回转轴 2**的旋转中心/交点之间的距离
- 偏移矢量 I4: 结束矢量链 $I4 = -(I2 + I3)$



机床侧视图

主轴定位到工作台上
方（回转轴 C）或者
工作台中心上方的定
位块上。

借助主轴上的测量杆
确定回转台的旋转中
心。



运动	回转台		TABLE_45
	X	Y	Z
偏移矢量 I2	0.000	-100.000	-20.000
回转轴矢量 V1	0.000	-1.000 ¹⁾	1.000 ¹⁾
偏移矢量 I3	0.000	0.000	-30.000
回转轴矢量 V2	0.000	0.000	-1.000
偏移矢量 I4	0.000	100.000	50.000
显示方案			
回转模式	轴方式		
方向参考	回转轴 2		
跟踪刀具	否		
回转轴			
回转轴 1	B	模式	自动方式
角度范围	0.000		180.000
回转轴 2	C	模式	自动方式
角度范围	0.000		360.000

1) 计算回转轴矢量 V1: $\beta = -45$ 度

$$V1Y = \sin(-45) = -0.7071$$

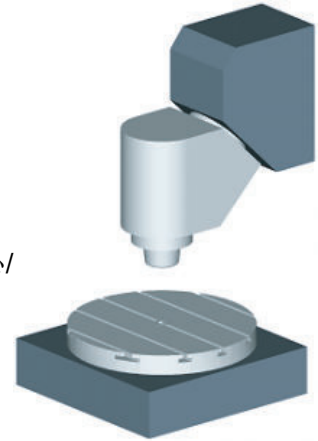
$$V1z = \cos(-45) = 0.7071$$

V1Y 和 V1z 可以定标为 -1 和 1。

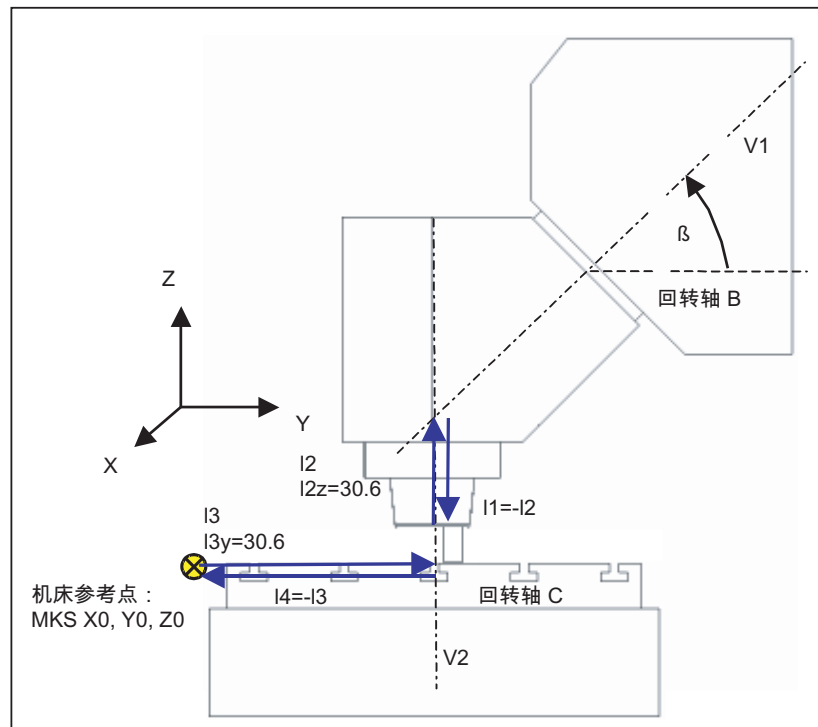
示例 4： 回转头/回转台“MIXED_45”

矢量取决于运动初始位置

- 回转轴矢量 V1: 回转轴 B 围绕 Y 和 Z 旋转
- 回转轴矢量 V2: 回转轴 C 围绕 Z 轴旋转
- 偏移矢量 I2: 从刀架基准点到
回转轴 1 的旋转中心/交点的距离
- 偏移矢量 I1: 结束矢量链 I1=-I2
- 偏移矢量 I3: 从机床基准点到回转轴 2 的旋转中心/
交点的距离
- 偏移矢量 I4: 结束矢量链 I4=-I3



机床侧视图:



主轴定位到工作台上方（回转轴 C）或者工作台中心上方的定位块上。借助主轴上的测量杆确定回转台的旋转中心。

表格 7-3 调试软键“回转”，运动（示例 4）

运动	混合式运动转换		MIXED_45
	X	Y	Z
偏移矢量 I1	0.000	0.000	-30.600
回转轴矢量 V1	0.000	1.000 ¹⁾	1.000 ¹⁾
偏移矢量 I2	0.000	0.000	30.600
偏移矢量 I3	300.000	150.000	0.000
回转轴矢量 V2	0.000	0.000	-1.000
偏移矢量 I4	-300.000	-150.000	0.000
显示方案			
回转模式	轴方式		
方向	回转轴 1		
跟踪刀具	是		
回转轴			
回转轴 1	B	模式	自动方式
角度范围	0.000		180.000
回转轴 2	C	模式	自动方式
角度范围	0.000		360.000

1) 计算回转轴矢量 V1: $\beta = 45$ 度

$$V1Y = \sin(45) = -0.7071$$

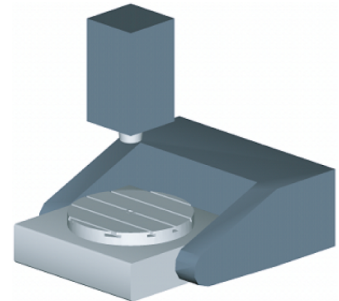
$$V1z = \cos(45) = 0.7071$$

V1Y 和 V1z 可以定标为 1。

示例 5：回转台“TABLE_5”

矢量取决于运动初始位置

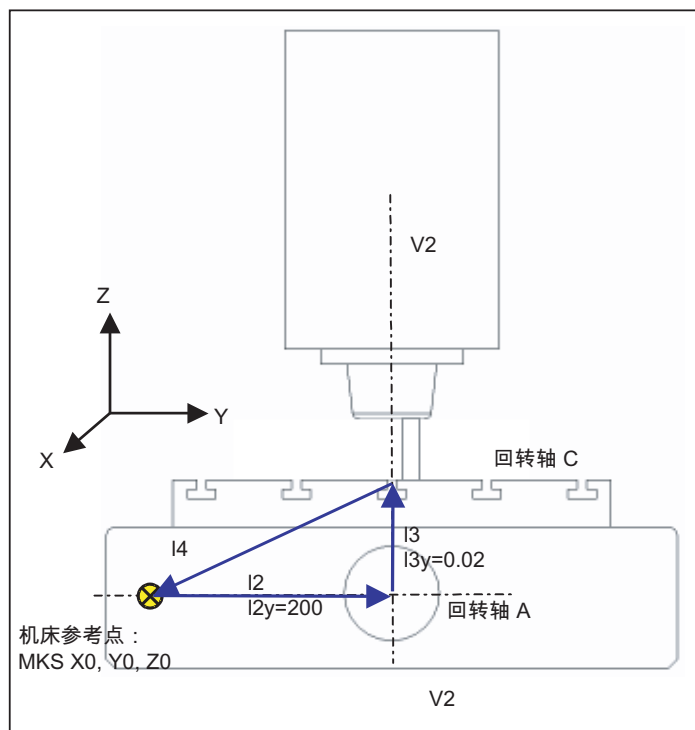
- 回转轴矢量 V1: 回转轴 A 围绕 X 轴旋转
 回转轴矢量 V2: 回转轴 C 围绕 Z 轴旋转
 偏移矢量 I2: 从机床基准点到**回转轴 1**的旋转中心/交点的距离
 偏移矢量 I3: 从回转轴 1 的旋转中心/交点到**回转轴 2**的旋转中心/交点的距离
 偏移矢量 I4: 结束矢量链 $I4 = -(I2 + I3)$



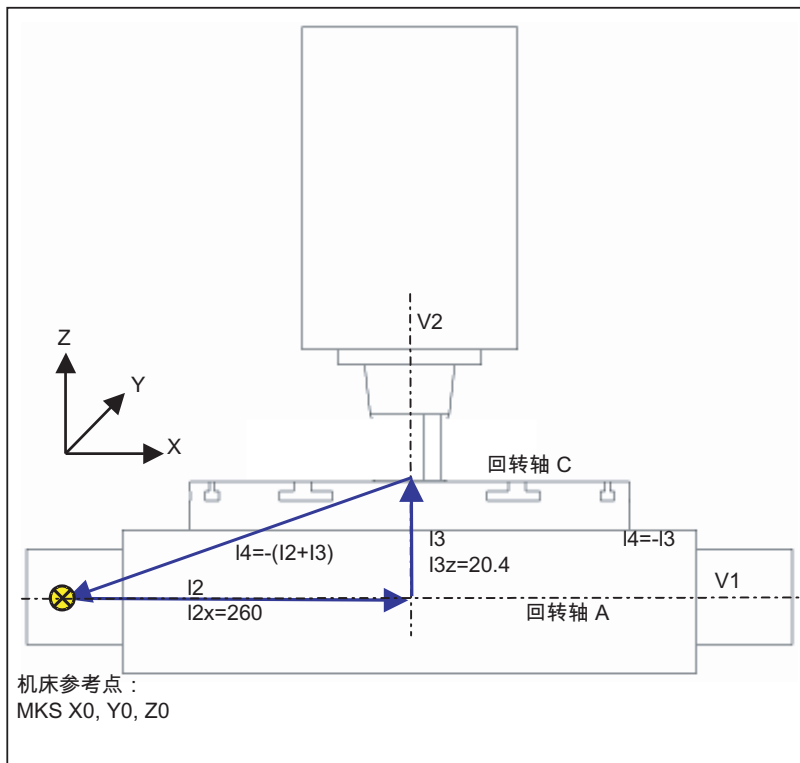
自 X 方向的机床侧视图

主轴定位到工作台上（回转轴 C）或者工作台中心上方的定位块上。

借助主轴上的测量杆确定回转台的旋转中心。



自 Y 方向的
机床前视图



表格 7-4 调试软键“回转”，运动（示例 5）

运动	回转台		TABLE_5
	X	Y	Z
偏移矢量 I2	260.000	200.000	0.000
回转轴矢量 V1	-1.000	0.000	0.000
偏移矢量 I3	0.000	0.020	20.400
回转轴矢量 V2	0.000	0.000	-1.000
偏移矢量 I4	-260.000	-200.020	-20.400
显示方案			
回转模式	轴方式		
方向	回转轴 1		
跟踪刀具	否		
回转轴			

运动	回转台		TABLE_5
回转轴 1	A	模式	自动方式
角度范围	-90.000		90.000
回转轴 2	C	模式	自动方式
角度范围	0.000		360.000

7.7.5 Herstellerzyklus CUST_800.SPF

功能

回转中的所有轴位置是通过循环 CUST_800.SPF 定义的。这些位置只能从回转循环 CYCLE800 中调用或者从循环

E_TCARR (ShopMill) 或 F_TCARR (ShopTurn) 中调用。

在循环 CUST_800.SPF 中提供并记录了功能标记 (_M2: 到 _M59)。也可参见后面的“结构程序 CYCLE800”。

说明

修改 CUST_800 时要注意，在 CUST_800 的末尾要再次加入所用 G 功能组的 G 指令（型号 S_G_1 到 S_G_5!）。

参数

CUST_800 (INT _MODE, INT _TC1, REAL _A1, REAL _A2, INT _TC2, REAL _T_POS)

_MODE	跳过标记 _M2 到 _M59
_TC1	回转头/回转台编号
_A1	回转轴 1 的角度
_A2	回转轴 2 的角度
_TC2	JOG 模式中回转的进给率系数(%)以及 ShopMill 中新换入的回转头/回转台的编号
_T_POS	沿刀具方向退回时的增量位置（参见标记 _M44, _M45）

回转前的退回

如果循环 CUST_800.SPF 没有被修改，在进行回转运动前 Z 轴（标记_M41）首先退回 MCS 中的某个位置，然后是 X 轴和 Y 轴（标记_M42）退回。在系统变量 \$TC_CARR38[n] 到 \$TC_CARR40[n] 中确定了这些位置值。在轴退回时，之前选中的刀沿(D0) 被撤销，在退回后再次被选中。

在确定了沿刀具方向退回时，刀具会退回到软限位上或者从工件出发退回一定的增量行程。此时系统会适当地考虑刀具长度。

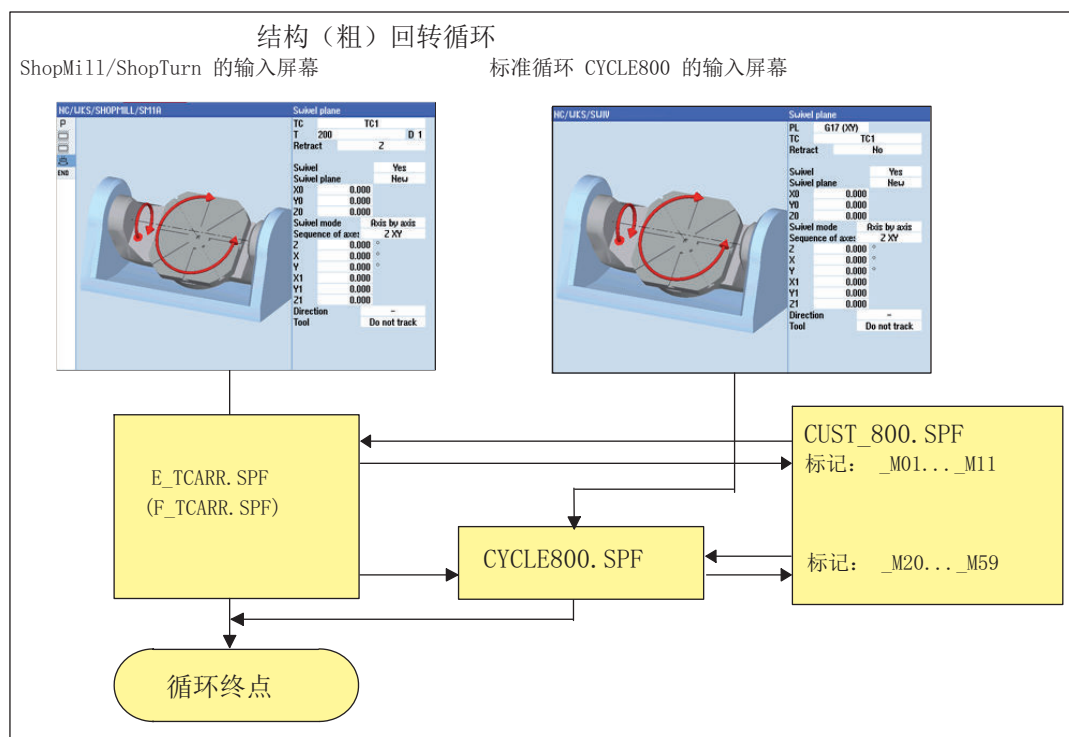


图 7-9 回转循环的大致结构

结构程序 CYCLE800

自动运行方式中的工作流程

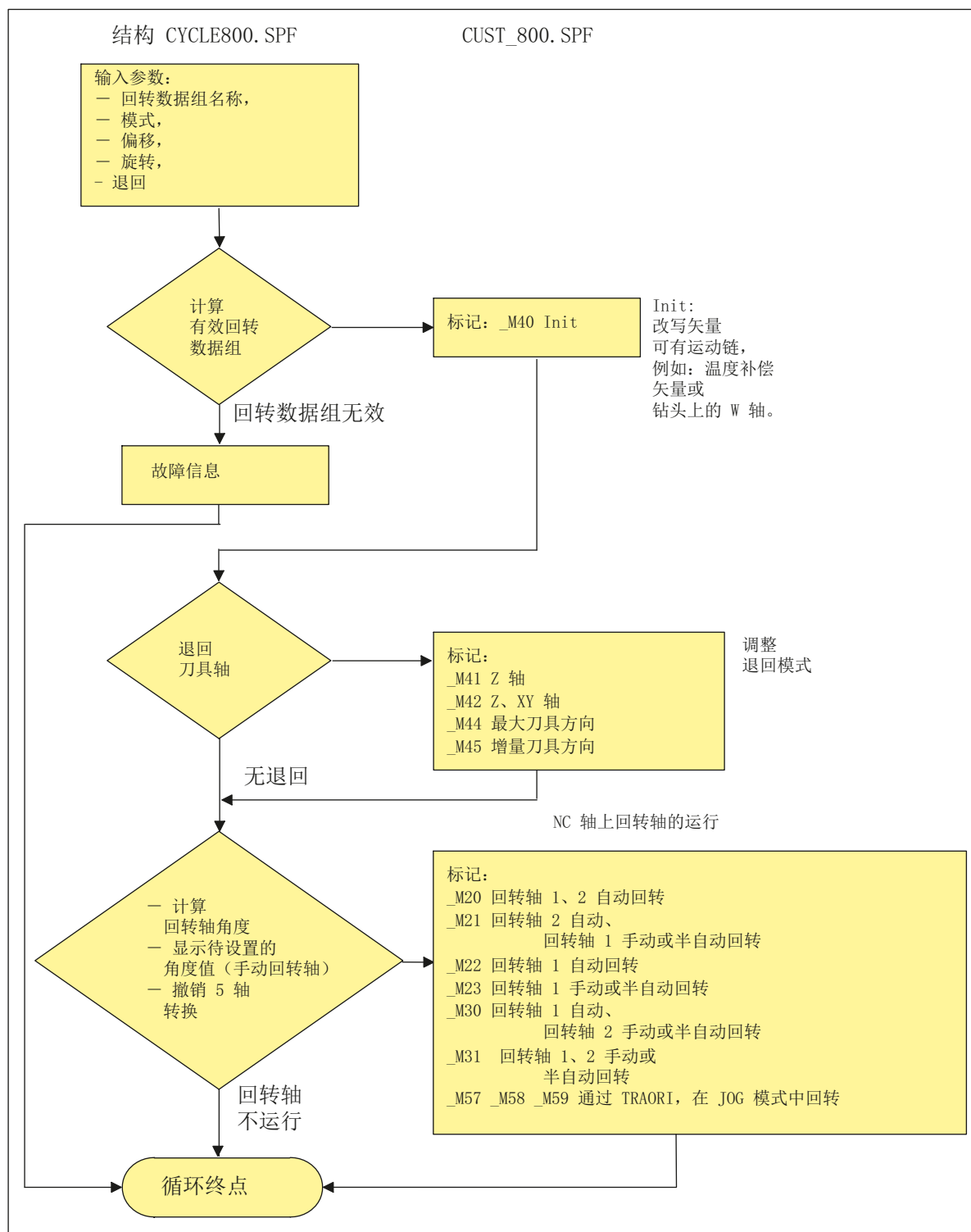


图 7-10 结构：CYCLE800.SPF / CUST_800.SPF

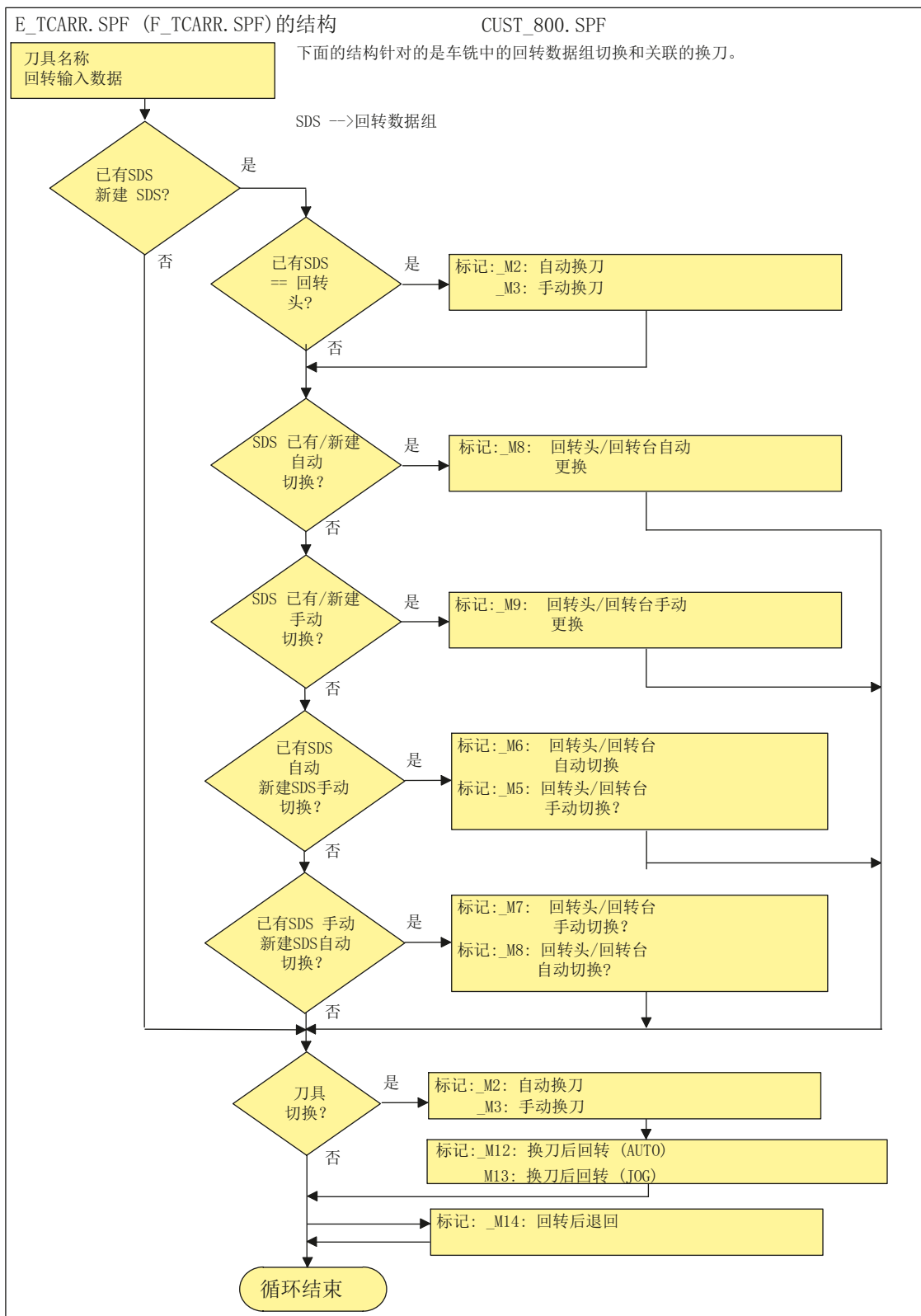


图 7-11 结构: E_TCARR.SPF (F_TCARR.SPF) / CUST_800.SPF

对于标记 **_M2** 到 **_M13** 的说明

在切换回转数据组(SDS)或换刀时，直线轴以之前的退回模式（模态有效）退回。

如果在车削/铣削时不需要此特性，则用分号（；）标注相关的调用。在制造商循环 CUST_800.SPF 中，车削/铣削（参见标记 **_M2** 到 **_M9**）调用的是循环 **E_SWIV_H** 或者 **F_SWIV_H**。

参数 **E_SWIV_H**（参数 1、参数 2、参数 3）

- 参数 1：回转数据组号（**_TC1**）
- 参数 2：回转轴 1 的角度
- 参数 3：回转轴 2 的角度

修改示例

如果在切换回转数据/换刀时不要求定位回转轴（回转头/回转台），可以在对应的标记上对循环 **E_SWIV_H** 进行注释。如果要求回转轴运行到指定的位置，则可以传送一个角度值到参数 2 和参数 3 中。

对于标记 **_M14**，**_M15** 的说明

根据退回平面和程序中的回转平面，程序段搜索结束后，直线轴开始移动，从当前位置移动到一同被回转的退回平面内的软限位上。为了避免该问题，在回转结束后将调用 CUST_800.SPF 中的标记 **_M14**。其中缺省的循环 **E_SP_RP(30)** 会使轴移动到铣削退回平面上，必要时移动到软限位。可以在标记 **_M15** 上设置一段程序段结束后的退回动作。

对标记 **_M16** 的说明

希望在切换回转数据组或撤销回转数据时回转轴运动到零点，可以相应地修改标记 **_M16** 的句法，例如：写入程序段 **N8006 E_SWIV_H(_TC1,0,0)**。

对标记 **_M20** 到 **_M31** 的说明

标记 **_M20** 到 **_M31** 的区别在于机床运动带有两个或一个回转轴。此外，区别还在于自动回转轴（已为 NCU 所知）和手动（半自动）回转轴。带有有效回转数据组回转运动只有一个标记有效。

对标记 **_M35** 的说明

在程序段搜索和带手动回转轴的回转数据组中使用标记 **_M35**。

对标记 `_M40` 的说明

该标记作为用户 `Init` 在调用和激活功能 `Toolcarrier` 前激活。 在一些特殊的运动过程中，借助该标记可以更新或改写有效回转数据组中的矢量。

钻头上的应用示例： 在平行轴上，`Z` 方向上的矢量随 `W` 轴的实际值更新。

在该标记处安装了一个 `STOPRE`。

制造商可以测试 `STOPRE` 在 `CUST_800.spf` 中的标记 `M40` 是否满足其功能要求。 此时要在中断条件(`RESET`)的速度（无 `STOPRE`）和功能之间进行权衡。 该功能也与机床运动类型有关。

对于标记 `_M41` 到 `_M45` 的说明

在标记 `_M41` 到 `_M45` 上，可以修改回转轴在执行回转前的退回模式。

在考虑刀具长度组件的情况下（不取消选择刀具切削）退回到 `MKS` 位置 `Z` 或 `Z, XY`。

对标记 `_M46` 的说明

标记 `_M46` 可以设置程序段搜索后、回转前的退回模式。 如果是铣削工艺程序，则变量 `_E_VER` 为 1。

对于标记 `_M57` 到 `_M59` 的说明

标记 `_M57` 到 `_M59` 用于 `JOG` 回转和 5 轴转换（`TRAORI`）。

对“跟踪刀具”的说明

跟踪刀具的前提是，已对与相应的回转数据组相对应的 5 轴转换进行调试。 用于跟踪刀具的程序部分集成在标记 `_M20`，`_M21`，`_M22` 和 `_M30` 中。 首个 5 轴转换用 `TRAORI(1)` 调用。

关于换刀 + 回转的说明

总得来说，一台机床上回转 (`CYCLE800`) 功能和换刀功能是相互独立的功能。 因此在一个需要使用多个刀具进行加工的工艺流程中（例如：先定中心、再钻削和攻丝），工作平面在回转后可以保持不变。

在换刀的机械过程中，有效回转数据组回转轴是否参与或者必须要退回，都需要在换刀程序中加以考虑。 换刀后回转轴退回到换刀前的位置。 如果有线性轴（几何轴）参与了换刀过程，则不能删除 `NC` 中的旋转（回转框架），而是要用指令 `G153` 或 `SUPA` 将线性轴作为机床轴进行定位。

关于不带有效刀具补偿的回转的说明

如果在没有有效刀沿(D0)条件下回转轴无法回转，您可以修改循环 CUST_800.SPF:

```
_M40 :  
IF ((NOT $P_TOOL) AND _TC1)  
  LOOP  
  MSG (“无刀沿有效”)  
  M0  
  STOPRE  
  ENDLOOP  
ENDIF  
GOTOF _MEND
```

7.7.6 CYCLE996 工件 - 运动测量

功能

使用功能运动测量可以通过测量空间内的曲面位置计算几何矢量，这些矢量用于定义 5 轴转换（TRAORI 和 TCARR）。

一般，通过每个回转轴的工件探头接触测量球的三个位置进行测量。然后根据用户预设的机床几何比例确定球面位置。球面位置通过各个待测量的回转轴的一次重新定位调节。

使用 CYCLE996 无需对于机床的基础机械构造有详细的了解。执行测量也无需机床的尺寸图和结构图。

使用范围

功能运动测量(CYCLE996) 可以确定回转轴运动转换（TRAORI, TCARR）功能相关的数据。

可能的情况：

- 重新确定回转数据组
 - 机床的调试
 - 作为 TCARR 可回转的夹具
- 检查回转数据组
 - 碰撞后的维修
 - 检查加工过程中的运动

手动轴（可手动调节的回转台、可回转的夹具等）的运动测量和 NC 控制的回转轴的运动测量一样。

启动 CYCLE996 时必须为回转数据配置基础数据（运动类型）。而测量本身无需运动转换激活即可执行。

如果在设定数据 SD55740 \$MNS_MEA_FUNCTION_MASK 中置位了位 7，则根据设置，运动测量基于计算出的回转轴矢量，或基于以保存在回转数据组中的回转轴矢量。

前提条件

使用 CYCLE996（运动测量）时必须满足以下前提条件：

- 安装了西门子测量包
- 校准了工件探头
- 安装了校准球
- 定向刀架已设置（MD18088 \$MN_MM_NUM_TOOL_CARRIER > 0）
- 机床基本几何数据：相互垂直、可返回参考点(X, Y, Z)
垂直针对刀具主轴，可通过测量杆进行控制。
- 参与转换的回转轴的位置已定义
- 参与转换的轴定义的运行方向符合标准 ISO 841-2001 或 DIN 66217（右手准则）
- 定义的日志文件大小（MD11420 \$MN_LEN_PROTOCOL_FILE = 100）

详细的测量与编程步骤包括示例都在下面的手册中有所说明：

参考文档

测量循环编程手册：CYCLE996

7.8 高速切削 (HSC)

7.8.1 配置高速设定 CYCLE832

功能

使用高速设定功能 (CYCLE832) 可以预先指定加工任意形状的表面所需的最佳数据。CYCLE832 的调用包含三个参数：

- 公差
- 加工方式 (工艺)
- 输入定向公差
第 3 个参数只用于带多方向转换的机床(TRAORI)。它的缺省值为 1。

公差/定向公差

线性轴 (几何轴) 的公差通过 NC 指令 CTOL 传送给数控系统。CTOL = 3 的平方根 * 公差值。

公差为 0：公差值 (CTOL) 来自以下设定数据：

SD55446 \$SCS_MILL_TOL_VALUE_ROUGH H	粗加工 CYCLE832 的公差值
SD55447 \$SCS_MILL_TOL_VALUE_SEMIFI N	初精整 CYCLE832 的公差值
SD55448 \$SCS_MILL_TOL_VALUE_FINISH	精加工 CYCLE832 的公差值

示例：CYCLE832(0,_ROUGH,1) → 公差来自 SD55446。

加工方式和工艺 G 功能组 59

在 CYCLE832 中或在 CUST_832.SPF 中为 G 功能组 59 固定分配工艺加工方式：

加工方式	简明文本	工艺 G 功能组 59	数组索引
取消	_OFF	DYNNORM	0
粗加工	_ROUGH	DYNROUGH	2
初精整	_SEMIFIN	DYNSEMIFIN	3
精加工	_FINISH	DYNFINISH	4

加工方式	简明文本	工艺 G 功能组 59	数组索引
粗加工 ^{*)}	_ORI_ROUGH	DYNROUGH	2
初精整 ^{*)}	_ORI_SEMIFIN	DYNSEMIFIN	3
精加工 ^{*)}	_ORI_FINISH	DYNFINISH	4

^{*)} 含定向公差输入

使用工艺 G 功能组可以根据各加工过程调整动态参数。使用工艺 G 功能组 59 指令可以激活下列还有相应**数组索引**的通道专用与轴专用机床数据：

MD20600 \$MC_MAX_PATH_JERK[0..4]	和轨迹相关的最大加速度变化速率
MD20602 \$MC_CURV_EFFECT_ON_PATH_ACCEL[0..4]	轨迹曲率对轨迹进给率和路径加速度的影响
MD20603 \$MC_CURV_EFFECT_ON_PATH_JERK[0..4]	轨迹曲率对路径加速度变化率的影响
MD32300 \$MA_MAX_AX_ACCEL[0..4]	最大轴加速度
MD32310 \$MA_MAX_ACCEL_OVL_FACTOR[0..4]	用于限制速度跳跃的过载系数
MD32431 \$MA_MAX_AX_JERK[0..4]	路径运动中的最大轴冲击
MD32432 \$MA_PATH_TRANS_JERK_LIM[0..4]	连续轨迹模式中程序段过渡处的最大轴冲击

说明

对机床轴进行优化时，要正确设置上述提到的机床数据值（注意数组索引）。

取消加工方式选择

在取消 CYCLE832 后，G 功能组会在程序运行期间设为机床数据 D20150 \$MC_GCODE_RESET_VALUES[] 中定义的值。涉及到的 G 功能组有：1、4、10、12、21、24、30、45、59、61。

轮廓公差与定向公差关闭（CTOL=-1，OTOL=-1）。

7.8.2 制造商循环 CUST_832.SPF

功能

如果因 CYCLE832.SPF 而与设置（G 功能）有所不同，您可以在制造商循环 CUST_832.SPF 中对该设置进行修改。

在 CUST_832.SPF 中定义了以下标记：

<code>_M_NORM:</code>	用于撤销 <code>_OFF CYCLE832</code>
<code>_M_FINISH:</code>	用于精加工 <code>_FINISH</code> 和 <code>_ORI_FINISH</code>
<code>_M_SEMIFINISH:</code>	用于初精整 <code>_SEMIFIN</code> 和 <code>_ORI_SEMIFIN</code>
<code>_M_ROUGH:</code>	用于粗加工 <code>_ROUGH</code> 和 <code>_ORI_ROUGH</code>

在 CUST_832.SPF 中对下列 G 功能进行编程：

```
SOFT  
COMPCAD  
G645  
FIFOCTRL  
UPATH  
;FFWON
```

根据加工方式：

```
DYNNORM  
DYNFINISH  
DYNSEMIFIN  
DYNROUGH
```

7.9 测量循环和测量功能

7.9.1 测量循环和测量功能，通用

前提条件

测量基本上需要使用两类电子测头：

- 测量工件的测头
- 测量刀具的测头

电子测头必须连接到 PPU 或 NX 模块上。

下文中电子测头简称为测头。

设置开关动作

可以通过下列通用机床数据设置所连接测头的极性：

MD13200 \$MN_MEAS_PROBE_LOW_ACTIVE		测头极性反转
= 0	测头非偏转状态为 0 V，缺省值 测头偏转状态为 24 V	
= 1	测头非偏转状态为 24 V 测头偏转状态为 0 V	

MD13210 \$MN_MEAS_TYPE		分布式驱动上的测量方式
= 1	缺省值，不可设定	

检查测头功能

可以通过手动偏转测头与控制下列 PLC 接口信号来检查测头的开关功能：

DB2700	来自 NC 的通用信号 NC -> PLC 接口							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB0	-	-	-	--	-	-	急停 生效	-
DBB1	系统 英制 单位	-	-	-	-	-	操作测头	
							测头 2	测头 1

例如可以使用 NC 检查程序借助下列 NC 命令来检查测头的开关动作与测量值的传输：

MEAS	测量，带剩余行程删除
\$AC_MEA[n]	检查开关动作，n = 测量输入编号
\$AA_MW[轴名称]	在工件坐标中轴的测量值
\$AA_MM[轴名称]	在机床坐标中轴的测量值

检查程序示例

程序代码	注释
%_N_PRUEF_MESSTASTER_MPF	
\$PATH=/_N_MPF_DIR	; 测头连接的检查程序
N00 DEF INT MTSIGNAL	; 开关状态检查标志
N05 G17 G54 T="3D_Taster" D1	; 选择用于测头的刀具几何值
N10 M06	; 激活刀具
N15 G0 G90 X0 F150	; 起始位置和测量速度
N20 MEAS=1 G1 X100	; X 中的测量输入端 1 处的测量
N30 MTSIGNAL=\$AC_MEA[1]	; 测量输入 1 上进行了开关动作, 是/否
N35 IF MTSIGNAL == 0 GOTOF _FEHL1	; 信号分析
N40 R1=\$AA_MM[X]	; 将机床坐标的测量值保存到 R1
N45 R2=\$AA_MW[X]	; 将工件坐标的测量值保存到 R2
N50 M0	; 检查 R1/R2 中的测量值
N55 M30	
N60 _FEHL1: MSG ("测头未开关!")	
N65 M0	
N70 M30	

测量循环的通用功能设置

MD51740 \$MNS_MEA_FUNCTION_MASK		测量循环的功能屏幕
位 0	激活自动测量标定状态监控工件测量	
位 1	工件测头长度基准, 工艺铣削	
= 0	刀具长度 L1, 取决于测头球体中心的位置	
= 1	刀具长度 L1, 取决于测头球体范围	
位 2	工件测量时支持可定向刀架, 带刀具补偿	
位 4	工件测头固定在机床上不可定位	
位 15	工件测量, 带轨迹特性 G60 精准停的测量程序段运行至程序段末尾	
位 16	刀具, 车削工艺, 考虑刀具	

说明

该机床数据修改后, 必须重新补偿 (标定) 测头。

MD52740 \$MCS_MEA_FUNCTION_MASK		测量循环的功能屏幕
位 0	工件测头测量输入	
= 0	CNC 测量输入 1 上的工件测头，激活 - 缺省设置	
= 1	CNC 测量输入 2 上的工件测头，激活	
位 1	第三几何轴（Y 轴）工件测量功能特性，工艺中以 G18 加工平面旋转！	
位 2	计算校正角度，带/不带激活的定向转变	
位 16	刀具测头测量输入	
= 0	CNC 测量输入 1 上的刀具测量头，激活 - 缺省设置	
= 1	CNC 测量输入 2 上的工件测头，激活	

MD54740 \$SNS_MEA_FUNCTION_MASK		测量循环的功能屏幕
位 0	超出尺寸差值（参数_TDIF）和/或允许范围（参数_TSA）后重复工件测量	
位 1	重复工件测量，带 M0 上报警和循环停止	
位 2	测量工件：公差报警 62304 尺寸余量、62305 尺寸不足、62306 超出允许的尺寸差值时的 M0	
位 3	将标定工件测头半径接收至刀具数据中。	
位 4	车削/铣削工艺中标定/测量平面不同。	
位 16	超出尺寸差值（参数_TDIF）和/或允许范围（参数_TSA）后重复刀具测量	
位 17	重复刀具测量，带 M0 上报警和循环停止	
位 18	测量工件：M0，超出允许尺寸差值后	
位 19	铣刀刀具测量：在最后一次接触时主轴转速减小	

MD55740 \$SCS_MEA_FUNCTION_MASK		测量循环的功能屏幕
位 0	中间定位时通过工件测头的碰撞监控	
位 1	激活平面上通过坐标系旋转的主轴校准耦合，自动运行方式中多测头工件测量时	
位 2	工件测量时根据激活平面上通过坐标系旋转的有效主轴校准耦合的主轴定位旋转方向	
位 3	测头未关闭时工件测量的重复次数	
位 4	工件测量进给	

7.9 测量循环和测量功能

位 5	工件测量时从测量位置的退回速度
位 6	工件测量，主轴定位时的测头激活/撤销
位 7	运动测量，定向矢量的标准化基础
位 8	运动测量，通过激活的 TRAORI 或 TCARR (CYCLE800) 测量
位 15	带已知或未知中心钻孔中的工件测量标定，JOG 测量
位 16	刀具测量时的碰撞监控
位 17	测头未关闭时刀具测量的重复次数
位 18	刀具测量进给
位 19	刀具测量时从测量位置的退回速度

7.9.2 制造商与用户循环 CUST_MEACYC.SPF

功能

循环“CUST_MEACYC.SPF”为测量循环功能的一部分，在每个测量循环中都会在执行测量任务前后调用该文件。它对于 JOG 方式下的测量和 AUTO 方式下的测量同样有效。

借助 CUST_MEACYC.SPF，可以编程并执行测量前后必要的步骤（例如激活/禁用测头）。

说明

执行修改前，将 CUST_MEACYC 循环从目录“NC 数据/循环/标准循环”中复制到目录“NC 数据/循环/用户循环”中。只有在该目录下 CUST_MEACYC 修改才生效。

CUST_MEACYC 结构程序

AUTO 方式下的步骤，以钻孔测量为例，带测头类型 712 的 CYCLE977。

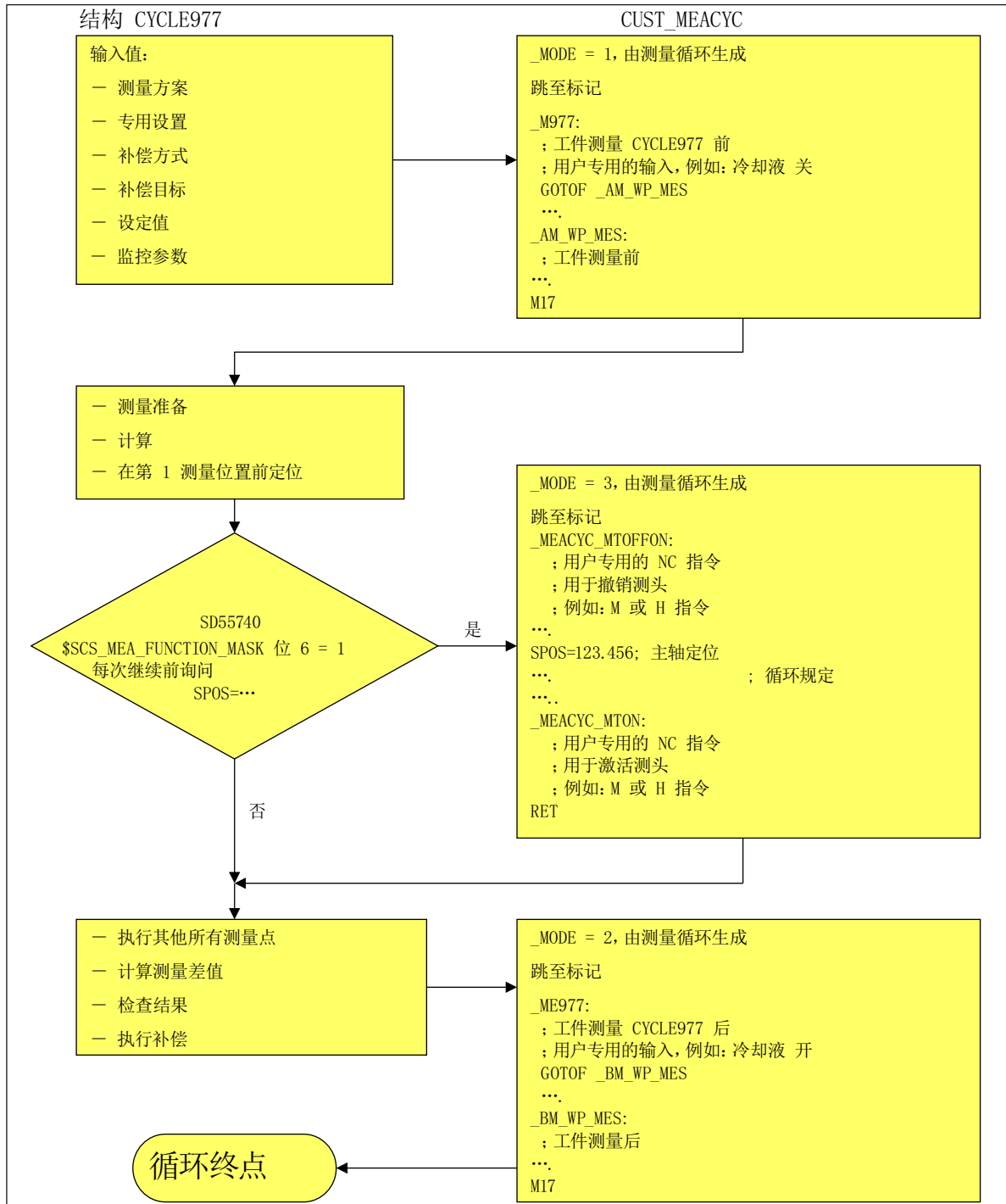


图 7-12 结构: CYCLE977 / CUST_MEACYC

7.9.3 在 JOG 运行方式下测量

前提条件

已执行了之前章节“测量循环和测量功能，通用 (页 186)”中描述的设置。

测量工件

用于铣削工艺：

- 将测量头换入刀具主轴。
- 在刀具列表中选择类型 7xx（测量头）。
- 将测量头作为当前 NC 通道中的刀具激活。

刀具测量

进行刀具测量时必须在机床空间内加装相应的测量头，确保它能够到达主轴上的刀具处。

刀具测量支持下列刀具类型：

- 铣削工艺： 刀具类型 1xx 和 2xx
- 车削工艺： 刀具类型 5xx, 1xx, 2xx

可以测量上述刀具类型的刀具长度和刀具半径。

设置通用数据

SD54782 \$SNS_J_MEA_FUNCTION_MASK_TOOL		输入屏幕“JOG 刀具测量”的设置
位 2	激活功能“自动刀具测量使能”。	
位 3	选择刀具测量头标定数据组，使能	
位 10	选择单片刀沿测量	
位 11	选择在平面内标定测头时的主轴是否旋转	

MD51784 \$MNS_J_MEA_T_PROBE_APPR_AX_DIR [i]		在 JOG 方式下测量： 选择在平面内逼近刀具测头的方向
-2	第 2 测量轴负向	
-1	第 1 测量轴负向（缺省值）	

0	
1	第 1 测量轴正向
2	第 2 测量轴正向

MD52750 \$MCS_J_MEA_FIXPOINT	相对固定点测量的 Z 值
[0]	第 1 轴的定位
[1]	第 2 轴的定位

MD52751 \$MCS_J_MEA_MAGN_GLAS_POS	刀具测量的放大位置
[0]	第 1 轴的定位
[1]	第 2 轴的定位

检查并设置通用机床数据

MD11450 \$MN_SEARCH_RUN_MODE	搜索的参数设置
位 1 = 1	输出动作程序段后自动启动 当 ASUB 结束后才输出报警 10208（另见 MD11620 \$MN_PROG_EVENT_NAME）。

MD11604 \$MN_ASUP_START_PRIO_LEVEL	优先级从 ASUP_START_MASK 起有效。
= 1 - 64	用于 ASUP_START_MASK 的优先级。

设置通道专用机床数据

MD20050 \$MC_AXCONF_GEOAX_ASSIGN_T AB[]	几何轴和对应的通道轴
[0]	前提条件是，在 JOG 运行方式下测量时所有几何轴可用。特别是 XYZ。
[1]	
[2]	

MD20110 \$MC_RESET_MODE_MASK	确定复位/零件程序结束后控制器的初始设置。
= 4045H	最小值
位 0 = 1	确定上电和复位后控制器的初始设置。
位 2 = 1	
位 6 = 1	
位 14 = 1	

MD20112 \$MC_START_MODE_MASK	确定零件程序启动后控制器的初始设置。
= 400H	最小值
位 6 = 0	确定 NC 启动后控制器的初始设置。

MD20310 \$MC_TOOL_MANAGEMENT_MASK	激活刀具管理功能。
= 4002H	最小值
位 1 = 1	刀具管理和监控功能有效。
位 14 = 1	在复位和启动时自动换刀。

MD28082 \$MC_SYSTEM_FRAME_MASK	系统框架（SRAM）。
1	缺省设置，不可修改。

只有将通道专用的机床数据 MD28082 \$MC_SYSTEM_FRAME_MASK 的位 0 设为 1 后，即设置用于实际值设定和对刀的系统框架，下列机床数据才生效。

MD24006 \$MC_CHSFRAME_RESET_MASK	复位后有效的系统框架。
位 0 = 1	实际值设定时系统框架的复位特性。

MD24007 \$MC_CHSFRAME_RESET_CLEAR_MASK	复位时删除系统框架。
位 0 = 0	复位时系统框架的删除特性。

说明

如果满足了本章中所述的前提条件，并设置或控制了机床设定数据，就可以在 JOG 运行方式下使用工件测量头在铣床上测量工件！

在 JOG 运行方式下可以使用刀具测量头在铣床或车床上测量刀具！

在后续章节中将说明，可以开展哪些设置，使测量过程符合机床上的特殊要求。

7.9.3.1 在铣床上测量工件

您可以通过以下通道专用的通用机床数据、循环设定数据，来根据特定要求调整“加工”区中的测量。

通用配置机床数据

SD55630 \$SCS_MEA_FEED_MEASURE		JOG 和 AUTO 方式中“工件测量” 用进给率
300	缺省值	

说明**“测量工件”时的测量进给率**

所有的测量循环都使用补偿（标定刀具测头）之后保存在 SD54611 中的值作为测量进给率。每一个标定字段[n]可以有不同的测量进给率。

在标定测头时，要么使用来自“SD55630 \$SCS_MEA_FEED_MEASURE”设置的测量进给率，要么使用在校准时输入屏幕窗口中的测量进给率。为此必须将“SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE”的位 4 设为 1。

MD52207 \$MCS_AXIS_USAGE_ATTRIB[n]		轴的属性 [n]通道轴号
位 6	指定回转轴为测量的补偿目标。	
	= 0	缺省值
	=1	指定回转轴为补偿目标，显示在“JOG 测量”窗口中。

7.9 测量循环和测量功能

MD51751 \$MNS_J_MEA_M_DIST_MANUELL	指定测量点前后的测量路径，以毫米为单位。
= 10	缺省值
MD51757 \$MNS_J_MEA_COLL_MONIT_FEED	指定碰撞监控有效时工作平面中的进给率，以毫米/分钟为单位。
= 1000	缺省值
MD51758 \$MNS_J_MEA_COLL_MONIT_POS_FEED	指定碰撞监控有效时进给轴中的进给率，以毫米/分钟为单位。
= 1000	缺省值
MD51770 \$MNS_J_MEA_CAL_RING_DIAM[n]	指定标准环直径的预设，单位毫米，专用于标定数据组
= -1	缺省值
MD51772 \$MNS_J_MEA_CAL_HEIGHT_FEEDA X[n]	指定进给轴方向标准高度的预设，单位毫米，专用于标定数据组
= -99999	缺省值

通用循环设定数据

SD54780 \$SNS_J_MEA_FUNCTION_MASK_PIE CE	配置操作区“加工”JOG方式下“测量”功能的输入对话框。
= 512	缺省值
位 2 = 1	激活带电子测头的测量。
位 3 = 1	使能测头标定数据的选项。
位 6 = 1	使能基准参考（SETFRAME）中的零偏补偿选项。
位 7 = 1	使能通道专用基准框架中的零偏补偿选项。

位 8 = 1	使能全局基准框架中的零偏补偿选项。
位 9 = 1	使能可设置框架中的零偏补偿选项。

7.9.3.2 在铣床上测量刀具

JOG 和 AUTO 方式中“刀具测量”用进给率

SD55628 \$SCS_MEA_TP_FEED_MEASURE	校准刀具测量头或主轴静止时测量刀具采用的进给率。
= 300	缺省值

说明

测量刀具时的进给率

所有的测量循环都使用补偿（标定）刀具测头之后保存在 SD54636 或 SD54651 中的进给率。每一个标定字段[n]可以有不同的测量进给率。

在标定测头时，要么使用来自“SD55628 SCS_MEA_TP_FEED_MEASURE”设置的测量进给率，要么使用在“标定”屏幕窗口中输入的测量进给率。为此必须将“SD54762 \$SNS_MEA_FUNCTION_MASK_TOOL”的位 4 设为 1。

在下面的设定数据中，索引 [k] 代表测头当前数据组的编号（即“测头编号 -1”）。

SD54633 \$SNS_MEA_TP_TYPE[k]	指定测头类型：立方体/圆片
= 0	立方体（缺省值）
= 101	XY 中的圆片，工件平面 G17
= 201	ZX 中的圆片，工件平面 G18
= 301	YZ 中的圆片，工件平面 G19

使用下列设定数据能够确定，可以在哪根轴的哪个方向上对刀具测头进行标定。

SD54632 \$SNS_MEA_T_PROBE_ALLOW_AX_DIR[k]	指定参与标定的轴和方向。
= 133	缺省值

小数位		
个位	1. 轴	
	= 0	该轴无标定
	= 1	仅有负方向
	= 2	仅有正方向
	= 3	两个方向
十位	2. 轴	
	= 0	该轴无标定
	= 1	仅有负方向
	= 2	仅有正方向
	= 3	两个方向
百位	3. 轴	
	= 0	该轴无标定
	= 1	仅有负方向
	= 2	仅有正方向
	= 3	两个方向

示例

如果通用设定数据 SD54632 \$SNS_MEA_T_PROBE_ALLOW_AX_DIR[k] 的值为 123，则在 G17 平面中按如下方式标定刀具测头：

- X 轴两个方向
- 仅 Y 轴仅正方向
- 仅 Z 轴负方向

SD54631 \$MNS_MEA_T_PROBE_DIAM_LENGTH[k]	指定刀具测头的有效直径，进行长度测量。
= 0	缺省值

刀具测头的通用循环机床数据/设定数据

MD51780	\$MNS_J_MEA_T_PROBE_DIAM_RAD[k]	指定刀具测头的有效直径，用于半径测量。
= 0	缺省值	

SD54634	\$SNS_MEA_TP_CAL_MEASURE_DEPTH[n]	刀具测头上边沿和铣刀下边沿之间的距离
= 2	缺省值	

MD51752	\$MNS_J_MEA_M_DIST_TOOL_LENGTH	指定刀具长度测量时的测量行程。
= 2	缺省值	

MD51753	\$MNS_J_MEA_M_DIST_TOOL_RADIUS	指定刀具半径测量时的测量行程。
= 1	缺省值	

MD51786	\$MNS_J_MEA_T_PROBE_MEASURE_DIST	指定“标定刀具测头”和“用静止主轴测量刀具”时的测量行程。
= 10	缺省值	

用旋转主轴进行测量时的通用循环设定数据

SD54670	\$SNS_MEA_CM_MAX_PERI_SPEED[0]	指定允许的待测刀具的最大圆周速度。
= 100	缺省值	

SD54671	\$SNS_MEA_CM_MAX_REVOLUTIONS[0]	指定允许的待测刀具的最大转速。一旦超过该转速，则转速自动减少。
= 1000	缺省值	

SD54672 \$SNS_MEA_CM_MAX_FEEDRATE[0]	指定允许的待测刀具接触测头时的最大进给率。
= 20	缺省值

SD54673 SNS_MEA_CM_MIN_FEEDRATE[0]	指定待测刀具首次接触测头的最小进给率。这样可以避免刀具半径较大时首次接触测头的进给率太小。
= 1	缺省值

SD54674 \$SNS_MEA_CM_SPIND_ROT_DIR[0]	指定测量刀具时主轴的旋转方向。
= 4	主轴旋转方向和 M4 一样（缺省值）

说明

如果调用测量循环时主轴已经旋转，则会保持该旋转方向不变，不管上述数据的设置如何。

SD54675 \$SNS_MEA_CM_FEEDFACTOR_1[0]	指定进给系数 1
= 10	缺省值
= 0	仅使用由循环计算出的进给率进行一次接触。但进给率至少为“SD54673[0] \$SNS_MEA_CM_MIN_FEEDRATE”指定的最小进给率。
>= 1	使用进给率第一次接触。但进给率起码是“SD54673[0] \$SNS_MEA_CM_MIN_FEEDRATE”x “SD54675[0] \$SNS_MEA_CM_FEEDFACTOR_1” 的值。

SD54676 \$SNS_MEA_CM_FEEDFACTOR_2[0]	指定进给系数 2
= 0	使用由测量循环计算出的进给率进行二次接触。仅在 SD54673[0] \$SNS_MEA_CM_FEEDFACTOR_1 > 0 时有效，缺省值
>= 1	使用由 SD54673[0] \$SNS_MEA_CM_MIN_FEEDRATE 计算出的进给率进行第二次接触时的进给系数 2。 使用由计算出的进给率进行第三次接触。

说明

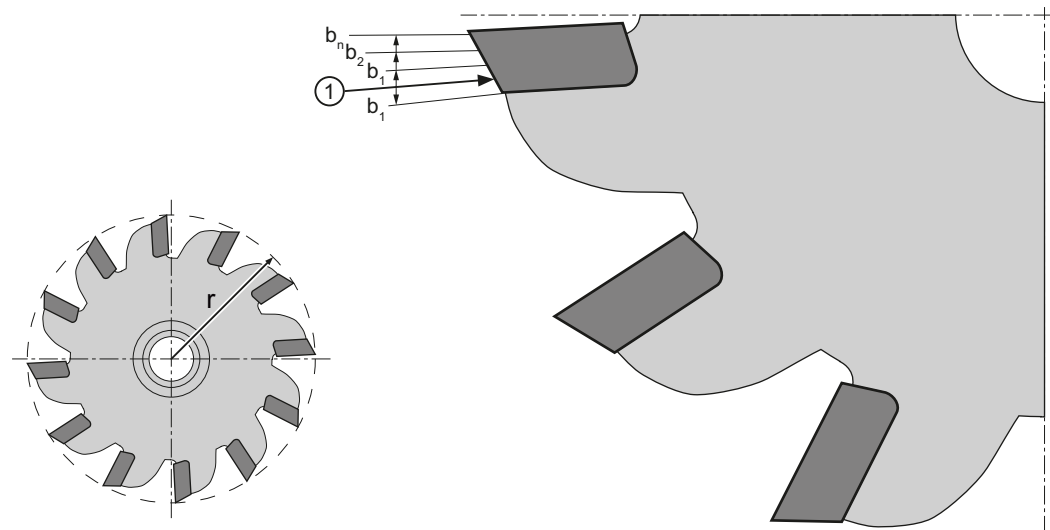
进给系数 2 应该小于进给系数 1。

SD54677 \$SNS_MEA_CM_MEASURING_ACCURACY[0]	指定要求的测量精度。 测量精度指刀具最后一次接触测头时的精度。
= 0.005	缺省值

SD54692 \$SNS_MEA_T_CIRCULAR_ARC_DIST	测量单片刀沿的半径时的弧长
= 0.25 ^{*)}	缺省值

SD54693 \$SNS_MEA_T_MAX_STEPS	测量单片刀沿的半径时的接触次数
= 10 ^{*)}	缺省值

^{*)} 值为 0 时，系统不查找最长刀沿上的最高点，而是采用接触旋转主轴得出的测量值。



- r 有效刀具半径
b 弧长 = SD54692
n 最大接触次数 = SD54693
① 1. 测量点

图 7-13 单片刀沿测量

使用旋转主轴进行测量时借助补偿列表进行补偿的通用循环设定数据

SD54691 \$SNS_MEA_T_PROBE_OFFSET	激活测量结果补偿
= 0	不指定，缺省值
= 1	指定循环内部补偿。仅当 SD54689 \$SNS_MEA_T_PROBE_MANUFACTURER>0 时有效。
= 2	指定通过用户自定义补偿表进行补偿。

SD54689 \$SNS_MEA_T_PROBE_MANUFACTURER	激活一些刀具测头型号预先制订的补偿表（用户专用）
= 0	不指定，缺省值
= 1	TT130 (Heidenhain)
= 2	TS27R (Renishaw)

7.9.3.3 在车床上测量刀具

车削工艺中刀具测量用进给率

SD55628 \$SCS_MEA_TP_FEED_MEASURE	校准刀具测量头或主轴静止时测量刀具采用的进给率。
= 300	缺省值

说明

测量刀具时的进给率

所有的测量循环都使用补偿（标定）刀具测头之后保存在 SD54636 或 SD54651 中的进给率。每一个标定字段[n]可以有不同的测量进给率。

在标定测头时，要么使用 SD55628 设置的测量进给率，要么使用在“标定”屏幕窗口中输入的测量进给率。为此必须将“SD54762 \$SNS_MEA_FUNCTION_MASK_TOOL”的位 4 设为 1。

通用循环机床数据/设定数据，用于指定测量行程/测量进给率

SD42950 \$SC_TOOL_LENGTH_TYP		指定刀具长度补偿，和刀具类型无关。
= 0	测量车刀类型 5xx（缺省值）	
= 2	测量车刀类型 5xx，钻头和铣刀类型 1xx，2xx	

MD51786 \$MNS_J_MEA_T_PROBE_MEASURE_DIST		指定“标定刀具测头”和“用静止主轴测量刀具”时的测量行程。
= 10	缺省值	

SD55628 \$SCS_MEA_TP_FEED_MEASURE		指定主轴静止时标定刀具测头的进给率。
= 300	缺省值	

标定刀具测头

可通过以下通用循环机床数据设置显示软键“标定测头”的最低访问级别。前提条件是已通过设定数据 SD54782（位 2=1）使能了自动刀具测量。

MD51070 \$MNS_ACCESS_CAL_TOOL_PROBE		指定访问“标定测头”的访问级别。
= 7	访问权限：访问级别 7（钥匙开关 0）	
= 4	读取访问权限：访问级别 4（钥匙开关 3），缺省值	
= 3	写访问权限：访问级别 3（用户）	

7.9.3.4 在 JOG 方式中不用电子探头进行测量

测量工件

操作方式 JOG 中的手动测量功能要求主轴运转，同时：

7.9 测量循环和测量功能

- 在接收测量值前通道必须处于复位状态。
- 主轴可以通过 T,S,M 菜单或者 DB3800.DBB5006 中的位移动或定位。
- 在固定转速时使用工艺功能 DB3800.DBB5006 需要满足以下前提条件：

机床数据的设置		
MD10709 \$MN_PROG_SD_POWERON_INIT_TAB [0] = 0		指定需要初始化的设定数据
MD35035 \$MA_SPIND_FUNCTION_MASK	Bit 4 = 0 Bit 5 = 1	主轴功能
SD43200 \$SA_SPIND_S		指定转速

说明

上电后，“SD43200 \$SA_SPIND_S”中的设置保持不变。

该设置不会再影响通过“SD41200 \$SN_JOG_SPIND_SET_VELO”设置的手动模式主轴速度。

7.9.4 在 AUTOMATIK 运行方式中测量

前提条件

**软件选件**

“在自动运行方式下测量”的功能需要下列软件选件：“测量循环”

已执行了之前章节“测量循环和测量功能，通用 (页 186)”中描述的设置。

测量工件

铣削工艺中的顺序：

1. 在刀具列表中选择类型 7xx（测头）。
2. 将测头换入刀具主轴。
3. 在当前的 NC 通道中激活探头。

车削工艺中的顺序:

1. 选择刀具“类型 580（3D 车削测头）”。
2. 在当前的 NC 通道中激活刀具。

刀具测量

进行刀具测量时必须在机床空间内加装相应的测头，确保它能够到达主轴上的刀具处。

刀具测量支持下列刀具类型：

- 铣削工艺： 刀具类型 1xx 和 2xx
- 车削工艺： 刀具类型 5xx, 1xx, 2xx

可以测量上述刀具类型的刀具长度和刀具半径。

通道专用循环设定数据

使用以下通道专用循环设定数据，可以在“程序”操作区中按特殊要求调整**工件测量**和**刀具测量**，但通常这两个测量不需要进行任何更改。

SD55613 \$SCS_MEA_RESULT_DISPLAY	指定测量结果的图形画面
= 0	没有测量结果的图形画面（缺省值）
= 1	有测量结果的图形画面，持续 8 秒
= 3	测量循环在内部机床数据上停止时，测量结果的图形画面保持不变！ 按下“NC 启动”继续运行时，图形画面被撤销。
= 4	仅在循环报警 61303、61304、61305、61306 时出现测量结果的图形画面。 按下“NC 启动”继续运行时，图形画面被撤销。

SD55614 \$SCS_MEA_RESULT_MRD	测量结果图形显示 MRD 的程序控制的设置
= 0	关闭测量结果图形显示
= 1	打开测量结果图形显示

SD55623 \$SCS_MEA_EMPIRIC_VALUE[n]	经验值
= 0	缺省值

SD55618 \$SCS_MEA_SIM_ENABLE	测量循环模拟
= 0	在测量循环和“SINUMERIK Operate”组合使用、开展 ShopMill 模拟或 ShopTurn 模拟时，会跳过测量循环（缺省值）。
= 1	默认设置： 测量循环和“SINUMERIK Operate”组合使用、开展 ShopMill 模拟或 ShopTurn 模拟时，会执行测量循环，但不进行补偿，也不进行记录。测量结果没有图形画面。

关于 SD55618 的更多信息参见下列参考文献：

测量循环编程手册，章节“在程序段搜索、试运行、程序测试、模拟时的特性”

配置程序编辑器中测量循环的输入屏幕

通过以下设定数据可以配置自动方式下刀具/工件测量的显示画面和补偿方式：

- SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE 测量工件：测量循环位于铣削区域，参见章节“测量工件，概述 (页 206)”
- SD54762 \$SNS_MEA_FUNCTION_MASK_TOOL 测量刀具：测量循环位于铣削区域，参见章节“在铣床上测量刀具 (页 210)”

说明

如果满足了本章中所述的前提条件，设置并检查了机床设定数据，就可以在自动方式下使用工件/刀具测头在机床上进行测量了！

如果希望调整车床上的工件测量过程，则需要进一步地进行设置。相关设置请参见章节“在车床上测量工件 (页 218)”。

如果希望调整刀具测量过程，同样需要进一步设置：相关设置请参见章节“在车床上测量刀具 (页 219)”或“在铣床上测量刀具 (页 210)”。

在后续章节中将说明，可以开展哪些设置，使测量过程符合机床上的特殊要求。

7.9.4.1 测量工件，概述

使用下面的通用和通道专用循环设定数据，可以在“程序”操作区中调整工件测量过程，使其符合特殊要求。

前提条件

已执行后续章节中的设置：

另见： 测量循环和测量功能，通用 (页 186)

设置

通用循环设定数据

SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE	输入屏幕的设置，在自动运行方式下测量，工件测量。
位 1	显示选择软键“3D 测量”
位 3	选择“使能测头校准数据组”
位 4	选择“校准测量进给率输入” ¹⁾
位 6	使能基准参考 (SETFRAME) 中的零偏补偿选项
位 7	使能通道专用基准框架中的零偏补偿选项
位 8	使能全局基准框架中的零偏补偿选项
位 9	使能可设置框架中的零偏补偿选项
位 10	使能零偏补偿选项 (粗和精)
位 11	几何和磨损刀具补偿选项
位 12	刀具补偿未取反和取反选项
位 13	刀具补偿 L1、R 或 L1、L2、L3 R 选项
位 14	刀具补偿零点补偿 (_TZL) 选项
位 15	刀具补偿尺寸差值检查 (_TDIF) 选项
位 16	带主轴旋转的工件测量选项
位 17	在切换方向上标定工件测头选项
位 18	测量次数 (_NMSP) 选项
位 19	带平均值计算 (_TMV) 的补偿选项 ¹⁾
位 20	经验值 (_EVNUM) 选项
位 21	设置补偿和选项
位 22	在未知中心点和已知中心点上标定的选项
位 24	带/不带位置偏移标定选项
位 25	使能主轴角度测量时的零点补偿选项

通道专用循环设定数据

SD55630 \$SCS_MEA_FEED_MEASURE	指定测量进给率 [毫米/分钟]
= 300	指定补偿（标定）工件测头时的测量进给率

说明

测量工件时的测量进给率

所有的测量循环都使用补偿（标定）刀具测头之后保存在“SD54611 \$SNS_MEA_WP_FEED[n]”中的值作为测量进给率。

每一个标定字段[n]可以有不同的测量进给率。

在标定测头时，要么使用来自“SD55630 \$SCS_MEA_FEED_MEASURE”的测量进给率，要么使用“标定”屏幕窗口中输入的测量进给率。为此必须将“SD54760 MEA_FUNCTION_MASK_PIECE”的位 4 设为 1。

SD55632 \$SCS_MEA_FEED_RAPID_IN_PERCENT	指定减速百分比。
= 50	在不带碰撞监控的循环内部中间定位时，快速移动速度的减速百分比（缺省值）。 通道专用循环设定数据“SD55600 SCS_MEA_COLLISION_MONITORING”必须设为 0。

SD55634 \$SCS_MEA_FEED_PLANE_VALUE	指定碰撞监控有效时工作平面中的中间定位进给率。
= 1000	缺省值

SD55636 \$SCS_MEA_FEED_FEEDAX_VALUE	指定碰撞监控有效时进给轴中的中间定位进给率。
= 1000	缺省值

SD55638 \$SCS_MEA_FEED_FAST_MEASURE	指定快速测量进给率。
= 900	缺省值

SD55640 \$SCS_MEA_FEED_CIRCLE	指定沿圆弧轨迹的进给率。
= 1000	缺省值

SD55642 \$SCS_MEA_EDGE_SAVE_ANG	指定“测量拐角”(CYCLE961)时的安全角度
= 0	度

该角度与工件参考边和正方向上激活的坐标系中的第 1 轴之间的角度用户说明共同生效并涉及到自动运行中的角度测量 (CYCLE961)。

角度用户说明在角度测量参数屏幕中的符号为“α0”。兼容程序中参见符号_STA1。

说明:

兼容程序中需要设置 \$SCS_MEA_EDGE_SAVE_ANG=10。

7.9.4.2 在铣床上测量工件

在“程序”操作区中，您可以通过通道专用的循环设定数据来自定义测量，使其符合您的要求，

设置

SD55625 \$SCS_MEA_AVERAGE_VALUE[n]	平均值的数量。
= 0	缺省值

通过带有“测量头补偿”（校准）功能的测量循环描述下列循环设定数据。在此无需设置用户参数。当然可以在测量头校准过程后检查这些数值并计算测量头的质量，例如在位置偏差时没有数值 > 0.1 毫米。另外可以重新机械校准测量头。

校准时的测量进给率会被应用于后续的测量循环。

说明

请注意测量头制造商的提示说明。

SD54600 \$SNS_MEA_WP_BALL_DIAM[n]	工件测量头球的有效直径。
SD54601 \$SNS_MEA_WP_TRIG_MINUS_DIR_AX1[n]	负方向触发点， 平面中的测量轴 1。
SD54602 \$SNS_MEA_WP_TRIG_PLUS_DIR_AX1[n]	正方向触发点， 平面中的测量轴 1。
SD54603 \$SNS_MEA_WP_TRIG_MINUS_DIR_AX2[n]	负方向触发点， 平面中的测量轴 2。
SD54604 \$SNS_MEA_WP_TRIG_PLUS_DIR_AX2[n]	正方向触发点， 平面中的测量轴 2。
SD54605 \$SNS_MEA_WP_TRIG_MINUS_DIR_AX3[n]	负方向触发点， 刀具方向上的测量轴 3。
SD54606 \$SNS_MEA_WP_TRIG_PLUS_DIR_AX3[n]	正方向触发点， 刀具反方向上的测量轴 3。缺省情况下 = 0。
SD54607 \$SNS_MEA_WP_POS_DEV_AX1[n]	位置偏差， 平面中的测量轴 1。
SD54608 \$SNS_MEA_WP_POS_DEV_AX2[n]	位置偏差， 平面中的测量轴 2。
SD54609 \$SNS_MEA_WP_STATUS_RT	轴定位的标定状态
SD54610 \$SNS_MEA_WP_STATUS_GEN[n]	校准状态
SD54611 \$SNS_MEA_WP_FEED[n]	校准时的测量进给率

7.9.4.3 在铣床上测量刀具

设置

SD54762 \$SNS_MEA_FUNCTION_MASK_TOOL	输入屏幕的设置，在自动运行方式下测量， 刀具测量
位 3	输入屏幕的设置，在自动运行方式下测量， 刀具测量；使能刀具测量头标定数据组选项
位 4	校准测量进给率输入 ¹⁾ 选项

位 5	接触时的进给输入和主轴转速选项
位 7	MCS 和 WCS 中测量选项
位 8	绝对测量和增量测量选项
位 9	几何和磨损刀具补偿选项
位 10	单片刀沿选项
位 11	选择在平面内标定测头时的主轴是否旋转
位 12	测量次数 (_NMSP) 选项
位 13	经验值 (_EVNUM) 选项

1) 测量进给输入适用于 AUTO 和 JOG 模式

刀具探头在机床坐标系中的校准数据

在开始标定前，必须将刀具探头在机床坐标系（MCS）中的位置输入到下列通用循环设定数据中。此时的参考点为主轴上有效刀具的外直径或刀具长度。如果主轴上没有刀具，则参考点为主轴中心点和主轴上的刀具参考点。

说明

校准探头

如果在 JOG 运行方式下已经校准了刀具探头，则校准数据会记录到以下设定数据中：

SD54632 \$SNS_MEA_TP_AX_DIR_AUTO_CAL[k]

在 AUTOMATIC 运行方式下不用再次校准刀具测头。

索引[k] 代表当前数据组的号码（即探头编号 -1）。

SD54625 \$SNS_MEA_TP_TRIG_MINUS_DIR_AX1[k]	指定测量轴 1 负方向上的触发点。
SD54626 \$SNS_MEA_TP_TRIG_PLUS_DIR_AX1[k]	指定测量轴 1 正方向上的触发点。
SD54627 \$SNS_MEA_TP_TRIG_MINUS_DIR_AX2[k]	指定测量轴 2 负方向上的触发点。
SD54628 \$SNS_MEA_TP_TRIG_PLUS_DIR_AX2[k]	指定测量轴 2 正方向上的触发点。
SD54629 \$SNS_MEA_TP_TRIG_MINUS_DIR_AX3[k]	指定测量轴 3 负方向上的触发点。

7.9 测量循环和测量功能

SD54630 \$SNS_MEA_TP_TRIG_PLUS_DIR_AX3[k]	指定测量轴 3 正方向上的触发点。
SD54631 \$SNS_MEA_TP_EDGE_DISK_SIZE[k]	指定刀具探头的边沿长度/圆片直径。
SD54632 \$SNS_MEA_TP_AX_DIR_AUTO_CAL[k]	指定在自动方式下参与探头校准的轴和方向
SD54634 \$SNS_MEA_TP_CAL_MEASURE_DEPTH[k]	指定刀具探头上沿与刀具下沿之间的距离（测量铣刀半径时指定校准深度和测量深度）。
SD54635 \$SNS_MEA_TPW_STATUS_GEN[k]	校准状态
SD54636 \$SNS_MEA_TPW_FEED[k]	指定校准时的测量进给率

在通用循环设定数据 SD54632 \$SNS_MEA_TP_AX_DIR_AUTO_CAL 中能够确定，可以在哪根轴的哪个方向上对刀具探头进行校准。

小数位		
个位	第 1 轴	
	= 0	该轴无标定
	= 1	仅有负方向
	= 2	仅有正方向
	= 3	两个方向
十位	第 2 轴	
	= 0	该轴无标定
	= 1	仅有负方向
	= 2	仅有正方向
	= 3	两个方向
百位	第 3 轴	
	= 0	该轴无标定
	= 1	仅有负方向
	= 2	仅有正方向
	= 3	两个方向

示例

如果通用循环设定数据 SD54632 \$SNS_MEA_TP_AX_DIR_AUTO_CAL 的值为 123，则在 G17 平面中按如下方式校准刀具探头：

- X 轴两个方向
- 仅 Y 轴仅正方向
- 仅 Z 轴负方向

SD54633 \$SNS_MEA_TP_TYPE[k]	指定探头的规格
= 0	立方体，缺省值。
= 101	XY 中的圆片，工件平面 G17。
= 201	ZX 中的圆片，工件平面 G18。
= 301	YZ 中的圆片，工件平面 G19。

刀具探头在工件坐标系中的校准数据

开始校准探头前，必须将刀具探头在工件坐标系（WCS）中的粗略位置输入到下列通用循环设定数据中。此时的参考点为主轴上有效刀具的外直径或刀具长度。如果主轴上没有刀具，则参考点为主轴中心点和主轴上的刀具参考点。

说明

在测量刀具时要注意，可设定零点偏移或基本参考的数据总是与校准时的数据一致（在工件坐标系 WCS 中测量！）。

始终使用相同的可设定零点偏移进行测量或校准。

SD54640 \$SNS_MEA_TPW_TRIG_MINUS_DIR_AX1[k]	指定测量轴 1 负方向上的触发点。
SD54641 \$SNS_MEA_TPW_TRIG_PLUS_DIR_AX1[k]	指定测量轴 1 正方向上的触发点。
SD54642 \$SNS_MEA_TPW_TRIG_MINUS_DIR_AX2[k]	指定测量轴 2 负方向上的触发点。
SD54643 \$SNS_MEA_TPW_TRIG_PLUS_DIR_AX2[k]	指定测量轴 2 正方向上的触发点。
SD54644 \$SNS_MEA_TPW_TRIG_MINUS_DIR_AX3[k]	指定测量轴 3 负方向上的触发点。

7.9 测量循环和测量功能

SD54645 \$SNS_MEA_TPW_TRIG_PLUS_DIR_AX3 [k]	指定测量轴 3 正方向上的触发点。
SD54646 \$SNS_MEA_TPW_EDGE_DISK_SIZE[k]	指定刀具探头的边沿长度/圆片直径。
SD54647 \$SNS_MEA_TPW_AX_DIR_AUTO_CAL[k]	指定自动校准刀具探头，自动使能轴/方向。
SD54648 \$SNS_MEA_TPW_TYPE[k]	指定探头的规格
= 0	立方体（缺省值）
= 101	XY 中的圆片，工件平面 G17。
= 201	ZX 中的圆片，工件平面 G18。
= 301	YZ 中的圆片，工件平面 G19。
SD54649 \$SNS_MEA_TPW_CAL_MEASURE_DEPTH[k]	指定刀具探头上沿与刀具下沿之间的距离（测量铣刀半径时指定校准深度和测量深度）。
SD54650 \$SNS_MEA_TPW_STATUS_GEN[k]	校准状态
SD54651 \$SNS_MEA_TPW_FEED[k]	指定校准时的测量进给率

使用下列通用的循环设定数据 SD54647 \$SNS_MEA_TPW_AX_DIR_AUTO_CAL 能够确定，可以在哪根轴的哪个方向上对刀具探头进行校准。

小数位		
个位	1. 轴	
	= 0	该轴无标定
	= 1	仅有负方向
	= 2	仅有正方向
	= 3	两个方向
十位	2. 轴	
	= 0	该轴无标定
	= 1	仅有负方向
	= 2	仅有正方向
	= 3	两个方向
百位	3. 轴	
	= 0	该轴无标定

小数位		
	= 1	仅有负方向
	= 2	仅有正方向
	= 3	两个方向

示例

如果通用循环设定数据 SD54647 \$SNS_MEA_TPW_AX_DIR_AUTO_CAL 的值为 123，则在 G17 平面中按如下方式校准刀具探头：

- X 轴两个方向
- 仅 Y 轴仅正方向
- 仅 Z 轴负方向

用旋转主轴测量时的监控

SD54670 \$SNS_MEA_CM_MAX_PERI_SPEED[0]	指定允许的待测刀具的最大圆周速度。
= 100	缺省值

SD54671 \$SNS_MEA_CM_MAX_REVOLUTIONS[0]	指定允许的待测刀具的最大转速。一旦超过该转速，则转速自动减少。
= 1000	缺省值

SD54672 \$SNS_MEA_CM_MAX_FEEDRATE[0]	指定允许的待测刀具接触测头时的最大进给率。
= 20	缺省值

SD54673 \$SNS_MEA_CM_MIN_FEEDRATE[0]		指定待测刀具首次接触测头的最小进给率。这样可以避免刀具半径较大时首次接触测头的进给率太小。
= 1	缺省值	

SD54674 \$SNS_MEA_CM_SPIND_ROT_DIR[0]		指定测量刀具时主轴的旋转方向。
4 = M4	缺省值	

说明

如果调用测量循环时主轴已经旋转，则会保持该旋转方向不变，不管上述数据的设置如何。

SD54675 \$SNS_MEA_CM_FEEDFACTOR_1[0]		指定进给系数 1
= 10	缺省值	
= 0	仅使用由循环计算出的进给率进行一次接触。但进给率至少为“SD54673[0] \$SNS_MEA_CM_MIN_FEEDRATE”指定的最小进给率。	
≥ 1	使用进给率第一次接触。但进给率起码是“SD54673[0] \$SNS_MEA_CM_MIN_FEEDRATE”x “SD54675[0] \$SNS_MEA_CM_FEEDFACTOR_1”的值。	

SD54676 \$SNS_MEA_CM_FEEDFACTOR_2[0]		指定进给系数 2
= 0	使用由循环计算出的进给率进行二次接触。仅在“SD54673 \$SNS_MEA_CM_FEEDFACTOR_1[0] > 0”时有效，缺省值	
≥ 1	使用由“SD54673 \$SNS_MEA_CM_MIN_FEEDRATE[0]”和“进给系数 2”相乘得出的进给率进行第二次接触。 使用由此计算出的进给率进行第三次接触。	

说明

进给系数 2 应该小于进给系数 1。

SD54677 \$SNS_MEA_CM_MEASURING_ACCURACY[0]	指定要求的测量精度。 测量精度指刀具最后一次接触探头时的精度。
= 0.005	缺省值

用旋转主轴测量： 借助补偿表进行测量值补偿

SD54691 \$SNS_MEA_T_PROBE_OFFSET	激活测量结果补偿。
= 0	不指定，缺省值
= 1	指定循环内部补偿。仅当 SD54690 \$SNS_MEA_T_PROBE_MANUFACTURER>0 时有效。
= 2	指定通过用户自定义补偿表进行补偿。

SD54689 \$SNS_MEA_T_PROBE_MANUFACTURE R	激活一些刀具探头型号预先制订的 补偿表（用户专用）。
= 0	不指定，缺省值
= 1	TT130 (Heidenhain)
= 2	TS27R (Renishaw)

用户补偿值

当通用循环设定数据 SD54691 \$SNS_MEA_T_PROBE_OFFSET= 2 时，下列设置有效：

SD54695 到 SD54700	补偿值用于半径测量。	参见后面的通用循环设定数据。
SD54705 到 SD54710	补偿值用于长度测量。	参见后面的通用循环设定数据。

SD54695 \$SNS_MEA_RESULT_OFFSET_TAB_RAD 1[n] SD54705 \$SNS_MEA_RESULT_OFFSET_TAB_LEN 1[n]	半径测量 长度测量
= 0	0
= 1	1. 半径
= 2	2. 半径

7.9 测量循环和测量功能

= 3	3. 半径
= 4	4. 半径

7.9.4.4 在车床上测量工件

车削用通用设置

MD52740 \$MNS_MEA_FUNCTION_MASK	测量循环的功能屏幕
位 1	工件测量，第三几何轴（Y，车削上）上的测量
= 0	测量循环不支持已有的第三几何轴（Y 应用）！
= 1	设定值说明和参数（SETVAL, _TUL, _TLL, SZO）涉及第三几何轴（Y 轴）然而刀具长度或零点偏移的补偿发生在第二几何轴（X 轴，纵坐标）激活的部分（即：Y 轴上测量；X 轴上补偿）。补偿目标会受参数_KNUM 影响！
位 16	刀具测头测量输入

刀具测头在机床坐标系（MCS）中的标定数据

在开始标定前，必须将刀具测头在机床坐标系（MCS）中的位置输入到下列通用循环设定数据中。

SD54615 \$SNS_MEA_CAL_EDGE_BASE_AX1[n]	指定标准槽底部在测量轴 1 上的位置。
SD54617 \$SNS_MEA_CAL_EDGE_PLUS_DIR_AX1[n]	指定标准槽边缘在测量轴 1 正方向上的位置。
SD54618 \$SNS_MEA_CAL_EDGE_MINUS_DIR_AX1[n]	指定标准槽边缘在测量轴 1 负方向上的位置。
SD54619 \$SNS_MEA_CAL_EDGE_BASE_AX2[n]	指定标准槽底部在测量轴 2 上的位置。
SD54620 \$SNS_MEA_CAL_EDGE_UPPERE_AX2[n]	指定标准槽上边缘在测量轴 2 上的位置。
SD54621 \$SNS_MEA_CAL_EDGE_PLUS_DIR_AX2[n]	指定标准槽边缘在测量轴 2 正方向上的位置。
SD54622 \$SNS_MEA_CAL_EDGE_MINUS_DIR_AX2[n]	指定标准槽边缘在测量轴 2 负方向上的位置。

说明

在带有轴 X 轴和 Z 轴（G18）的标准车床上，测量轴 1 为 Z 轴而测量轴 2 为 X 轴。

7.9.4.5 在车床上测量刀具**刀具探头在机床坐标系（WCS）中的校准数据**

如果要在机床坐标系（MCS）中校准刀具探头，就必须在下列通用循环设定数据中输入刀具探头在 MCS 中的位置。

SD54625 \$SNS_MEA_TP_TRIG_MINUS_DIR_AX1[k]	指定测量轴 1（G18 平面内为 Z 轴）负方向上的触发点。
SD54626 \$SNS_MEA_TP_TRIG_PLUS_DIR_AX1[k]	指定测量轴 1（G18 平面内为 Z 轴）正方向上的触发点。
SD54627 \$SNS_MEA_TP_TRIG_MINUS_DIR_AX2[k]	指定测量轴 2（G18 平面内为 X 轴）负方向上的触发点。
SD54628 \$SNS_MEA_TP_TRIG_PLUS_DIR_AX2[k]	指定测量轴 2（G18 平面内为 X 轴）正方向上的触发点。

刀具探头在工件坐标系中的校准数据

如果要在工件坐标系（WCS）中校准刀具探头，就必须在下列通用循环设定数据中输入刀具探头在 WCS 中的大致位置。此时的参考点为主轴上有效刀具的外直径或刀具长度。

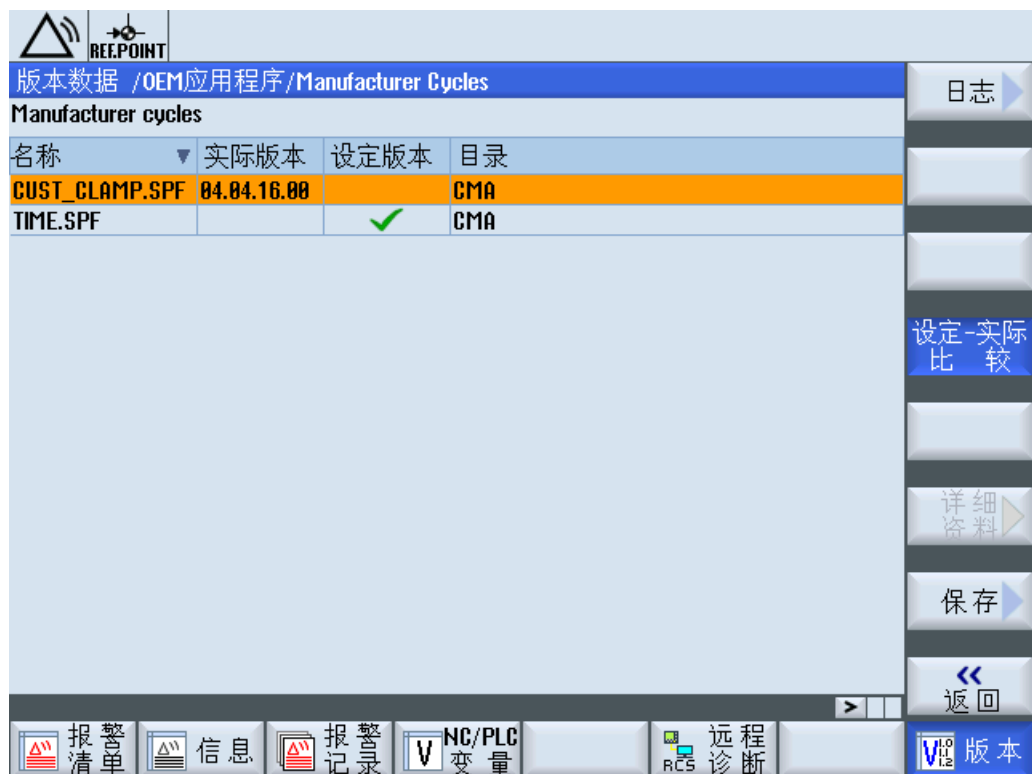
索引[k] 代表当前数据组的号码（即探头编号 -1）。

SD54640 \$SNS_MEA_TPW_TRIG_MINUS_DIR_AX1[k]	测量轴 1（G18 上为 Z 轴）负向上的触发点。
SD54641 \$SNS_MEA_TPW_TRIG_PLUS_DIR_AX1[k]	测量轴 1（G18 上为 Z 轴）正向上的触发点。
SD54642 \$SNS_MEA_TPW_TRIG_MINUS_DIR_AX2[k]	测量轴 2（G18 上为 X 轴）负向上的触发点。
SD54643 \$SNS_MEA_TPW_TRIG_PLUS_DIR_AX2[k]	测量轴 2（G18 上为 X 轴）正向上的触发点。

7.10 比较循环版本

7.10.1 显示循环版本

如果循环中配有版本说明，可以在版本视图中显示该版本。



前提条件

版本说明通过以下方式录入循环文件：

;VERSION: <Version> ;DATE: <YYYY-MM-DD>

示例：

;VERSION: 05.05.05.00 ;DATE:2012-11-30

步骤



1. 选择操作区域“诊断”。



2. 按下软键“版本”。

获取版本信息的过程较慢，对话框中会显示一个进度条和说明性文字。



3. 选中“OEM 应用/制造商循环”区并按下软键“详细”。

“制造商循环”窗口打开。

软键“设定-实际比较”默认选中。

7.10.2 指定循环版本

简介

可在版本视图中查看当前循环版本和目标版本的对比表。

版本数据 / OEM应用程序/Manufacturer Cycles

Manufacturer cycles

版本：05.05.05.00

名称	实际版本	设定版本	目录
CUST_CLAMP.SPF	04.04.16.00	05.05.05.00	CMA
TIME.SPF		05.05.05.00	CMA

右侧软键：日志、设定-实际比较、详细资料、保存、返回

底部软键：报警清除、信息、报警记录、NC/PLC 变量、远程诊断、版本

文件概览

需要以下文件：

名称	路径	含义
versions.xml	/siemens/sinumerik/hmi/data/version/oem/cma	版本文件
versions.xml	/oem/	版本路径文件

设置版本文件

循环版本说明中需要手动设置版本文件。

1. 从如下目录中复制模板文件“versions.xml”：
/siemens/sinumerik/hmi/data/version/oem/cma
2. 将模板文件保存/oem 下的自定义目录下。
此处的默认目录为 /oem/sinumerik/hmi/data/version/oem/cma。
3. 打开文件。
4. 在标签<info>“defaultFileVersion”中输入循环文件的目标版本，在标签<Version>中输入总版本说明。
5. 关闭文件，保存修改。

示例：

```
<info defaultFileType=" *.spf *.cpf" defaultFileVersion="05.05.00.00"
linkname="CMA" linkpath="//NC/CMA.DIR" EffDirOrder="CUS CMA">
<Name>Manufacturer cycles</Name>
<Version>5.5.0.0</Version>
<Link>
<Name>CUS</Name>
<Path>//NC/CUS.DIR</Path>
</Link>
</info>
```

设置版本路径文件

如果版本文件在默认目录下且当前为 oem\versions.xml，则无需进行任何更改。在所有其他情况下可以手动输入路径：

1. 打开 /oem 下的文件 versions.xml。
2. 使用自己的版本文件路径替换标签<Path>“Manufacturers Cycles”中的默认路径。
3. 关闭文件，保存修改。

标准：

```
<Component>  
<Name>Manufacturer Cycles</Name>  
<Path>/siemens/sinumerik/hmi/data/version/oem/cma</Path>  
</Component>
```

示例：

```
<Component>  
<Name>Manufacturer Cycles</Name>  
<Path>/oem/sinumerik/hmi/data/version/oem/cma</Path>  
</Component>
```

7.11 循环保护（选件）

7.11.1 循环保护概述

使用

使用循环保护可对循环进行加密，之后将受保护的循环存储在控制系统中。这种加密循环在 NC 中的执行不受限制。

**软件选件**

使用该功能需要下列选件“Lock MyCycles”。

说明

采用加密不会违反出口限制或禁运规定。

7.11 循环保护 (选件)

为了保护制造商的技术，会阻止所有对受保护的循环的查阅。在开展服务时机床制造商必须提供未加密的循环。

说明

最终用户

在使用机床制造商的加密循环时，在出现问题的情况下只能向机床制造商请求维修。

机床制造商

机床制造商在使用加密循环时，必须确保按版本对原始的、未加密的循环进行了存档。

说明

在控制系统外部通过 Programm SinuCom Protector 对循环加密。

加密

调用程序时，加密 NC 循环在与原始循环相同的目录中生成：

- 若需启用强效加密，请在调用程序时递交选件“-e2”。
- 若需启用与较旧的软件版本相兼容的加密，请在调用程序时递交选件“-e1”。

OEM 将加密的 NC 循环载入至控制系统，并通过 NC 热启动激活。

注意
加密的兼容性
将通过“-e2”选件加密的循环载入较旧的软件版本时，系统会劝阻该操作。

西门子循环供货时始终与软件版本相匹配。

不允许混合使用通过“-e1”和“-e2”加密的循环。

复制加密的循环

可复制加密的循环，然后将其用于其他机床。

- **仅在一台机床上使用加密循环**

如需避免在其他机床上使用加密循环，可将循环与机床绑定。为此可使用机床数据 MD18030 \$MN_HW_SERIAL_NUMBER。

控制系统启动时会将 CF 卡的唯一硬件序列号保存在该机床数据中。如需将循环与一台机床绑定，必须在循环的调用指令开头查询 CF 卡的具体序列号 (MD18030 \$MN_HW_SERIAL_NUMBER)。若循环识别出了不匹配的序列号，则会在循环中输出报警并阻止后续处理。循环的代码被加密，因此形成了循环与指定硬件的绑定。

- **在多台机床上使用加密循环**

若需将循环绑定至定义的多台机床，必须在循环中输入每个硬件序列号。循环必须使用这些硬件序列号重新加密。

加密循环的使用

同 _SPF 或 _MPF 文件一样，可删除或卸载 _CPF 文件。如创建了存档，则会对所有加密的 _CPF 文件进行备份。

- 加密循环无法直接选中执行。该类循环只能从一个程序或直接在 MDA 中调用。
- 加密循环无法通过“外部执行”功能执行。

7.11 循环保护 (选件)

主轴功能

8.1 主轴控制

主轴控制可用以下设置：

程序结束

下列 M 功能用于程序结束：

MD10714 \$MN_M_NO_FCT_EOP 中的 M 功能	主程序结束	主轴继续运行
M2, M30	主程序结束并返回程序开头	主轴停止
M17	子程序结束并返回主程序	主轴继续运行

如果机床数据 MD10714 \$MN_M_NO_FCT_EOP 不等于零，则 M2 / M30（NC 程序结束）和 MD10714 \$MN_M_NO_FCT_EOP 中设置的 M 功能之间会有区别。

示例

MD10714 \$MN_M_NO_FCT_EOP = 32 表示“M32”，即运行方式“JOG”或“MDA”下创建的程序的结尾。

此外还需要此功能，用于在手动运行中持续启动主轴（例如用于对刀）。

配置按键

通过机床操作面板的按键手动控制主轴时，需在主轴数据块中使用下列接口信号：

DB3800.DBX5006.0	主轴停止
DB3800.DBX5006.1	主轴顺转
DB3800.DBX5006.2	主轴逆转

8.1 主轴控制

可在以下状态中启动和停止主轴：

DB3300.DBX3.7 = 1	通道状态: 复位
DB3300.DBX3.6 = 1	通道状态: 中断
DB3300.DBX3.3 = 1	程序状态: 中断

说明

需要在程序运行的情况下停止主轴时，请在用户 PLC 中设置接口信号，DB380x.DBX4.3，“进给轴停止/主轴停止”。

其他设置：

MD11450 \$MN_SEARCH_RUN_MODE	编程搜索模式
位 1 = 1	<p>输出动作程序段后自动启动 (PROG_EVENT)。当 ASUB 结束后才输出报警 10208。</p> <p>说明： 在 SINUMERIK 828D 中循环 PROG_EVENT 不可更改。制造商特定的输入仅在循环 CYCPE_MA 和 CYCPE_US 中可行。</p>

参考文档

主轴配置的详细描述请参见：

功能手册之基本功能分册：主轴 (S1)

旋转方向（仅适用于 ShopTurn）

在 ShopTurn 界面中，确保主轴和 C 轴的旋转方向显示正确，并且在编程 ShopTurn 功能时执行了正确的旋转方向。此处的设置必须与机床上主轴/ C 轴的实际旋转方向一致。

- 使用柱面转换和端面加工功能前，必须确保已正确调试了功能。
参见柱面转换（TRACYL）(页 132)
- 此外可以通过机床数据定义，在 ShopTurn 界面中使用 M3 时显示哪些主轴/ C 轴旋转方向：
参见：定义旋转方向的章节：设置用于车削的 ShopTurn 循环 (页 119)
- 通过接口信号 DB38xx.DBX2001.6 可以指定 C 轴正向旋转是哪个主轴旋转方向（M3/ M4）。该位定义了 M3 和 C+ 是同向（= 0）还是逆向（= 1）。相应的设置方法参见下列章节：
 - 副主轴旋转方向（设置 ShopTurn 中的副主轴 (页 124)）
 - 主主轴旋转方向（设置用于车削的 ShopTurn 循环 (页 119)）

8.2 模拟主轴

表示模拟主轴的机床轴索引

没有指定驱动（VSA、HSA、SLM）的主轴的功率显示可由 PLC 控制。

您需要在以下机床数据中输入模拟主轴的轴索引，以便操作软件将主轴判定为模拟主轴。

MD51068 \$MNS_SPIND_DRIVELOAD_FROM _PLC1	主轴 1 的机床轴索引 负载显示来自 PLC
	主轴（模拟）的机床轴索引，T,F,S 窗口中该主轴的负载显示来自 PLC (DB1900.DBB5006)。

8.3 主轴丝杆螺距误差补偿

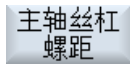
您可以通过输入屏幕输入所有轴的补偿数据、检查或修改补偿数据，而不通过零件程序或将 ini 文件导入 NC 来输入补偿数据。

经过修改的数据在 NC 中统一激活。必要的机床数据会自动设置。

步骤

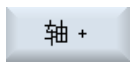


1. 选择操作区“调试”，按下软键“NC”。



2. 按下软键“主轴螺距误差”

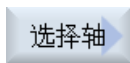
窗口“补偿表数值”打开，其中显示了某根轴预先设置的补偿表，当前还没有补偿值时，显示使用的测量系统。



3. 按下软键“轴 +”或“轴 -”，选择所需轴。



-或者-



按下软键“选择轴”。



窗口“轴直接选择”打开。将光标移动到所需轴上，按下“确认”。



4. 按下软键“配置”，编辑所选轴的补偿值。

窗口“补偿表配置”打开。

5. 输入以下项目：

- 补偿方式：EEC 还是 CEC。
- 起点位置、终点位置和控制点间距。
- 只限 EEC：
使用的测量系统；有多个测量系统时，要从下拉菜单中选择一个测量系统。
- 只限 CEC：
 - 补偿表(-)：从大到小的表序号。
 - 补偿表(+): 从小到大的表序号。



6. 按下软键“激活”结束输入。

接着屏幕上会弹出提示，要求重启 NC，以激活补偿数据。按下“确认”。



窗口“补偿表数值”打开。

7. 为每个控制点输入所需补偿值。



8. 再次按下软键“激活”，将补偿值传送给 NC。

参考文档

功能手册，扩展功能 (K3)

8.4 主轴诊断

8.4.1 主轴诊断



软件选件

“主轴诊断”功能需要选件“S-Monitor”（主轴监测器）。

利用各种状态信号可对主主轴的驱动器进行监控。系统通过主轴上的 DRIVE CliQ 接口读入驱动器内的这些状态信号，系统会对状态信号进行检测，使主轴运行状态的诊断更加高效：借助软件选件“S-Monitor（主轴监测器）”，操作界面上会显示以下信息：

- 运行小时
- 温度
- 转速/转矩
- 夹紧系统
- 物流数据

前提条件

- 主轴配有编码器模块 SMI24：
r0459，位 14 = 1。
- 为主轴配置了驱动报文 139。
- 机床轴具有主轴功能，即
机床数据 35000 \$MA_SPIND_ASSIGN_TO_MACHAX > 0
该数据的值即主轴号。

文档

更多信息参见下列文档:

- 功能手册之基本功能 (S1); 章节“配备 SMI24 的主轴”
- 有关应用说明“启用一根配备 SMI24 的混合主轴”的详细信息请访问:
开始运转混合主轴, 带 SMI24 (<http://support.automation.siemens.com/>), 条目 ID 67598396

数据一览

在窗口“主轴诊断 S1”中显示以下信息:

参数	值
运行小时 ● 主轴处于闭环控制中 ● 主轴旋转	h
夹紧次数 (即换刀次数) 夹紧循环定义: 夹紧状态: 已松开 [3] → 夹紧状态: 已夹紧 [7、8 或 10] → 夹紧状态: 已松开 [3]。	
电机温度	°C
夹紧状态 (传感器 1)	V
活塞松开 (传感器 4)	
例如: 定义的轴旋转角位 (传感器 5)	是/否
最大转速	rpm

步骤

1. 选择操作区域“诊断”。
2. 按下菜单扩展键 2 次。
3. 按下软键“主轴诊断”, 显示数据一览表。
窗口“主轴诊断 S1”打开。

说明

如果有多个主轴在工作, 可通过软键“主轴+”或“主轴-”选择所需主轴。

8.4.2 温度

检测和显示的温度取决于安装在主轴中的传感器的数量和安装位置（例如：传感器 S6 用于测量前轴承温度）。如果温度传感器类型的参数 p0601 = 0 或 1，则不显示电机温度。

说明

温度限值

此处所显示的限值针对的是主轴监控，不会影响在驱动中设置的适用于主轴停止响应的驱动参数值。温度限值在 SMI24 中为缺省设置，无法在该窗口中修改。必要时还须调整驱动的关闭限值。

最多可以检测 3 个温度：

信号	传感器	列/含义	说明
电机温度 (页 235)	KTY	实际值	当前电机温度。如果超温保护存在且响应，则显示值 250 °C。
电机温度报警		限值	保存的电机温度阈值，超出该阈值会因 KTY 电机温度过高而出现报警。
		超限时间	显示累积的超限时间（单位：小时）。
		最后一次超限	显示最后一次超限时的时间戳[日期]和[时间]。
		超限次数	累积的超限次数。
电机温度故障		限值	保存的电机温度阈值，超出该阈值会因 KTY 电机温度过高而出现故障。
		超限时间	显示累积的超限时间（单位：小时）。
		最后一次超限	显示最后一次超限时的时间戳[日期]和[时间]。
		超限次数	累积的超限次数

8.4 主轴诊断

信号	传感器	列/含义	说明
超温保护	PTC	实际值	如果超温保护响应，则显示“激活”（红色）。“实际值”一列中显示的电机温度值为 250 °C。
		限值	保存的电机温度阈值，超出该阈值会因 PTC 绕组电机温度过高而出现故障。
		最后一次超限	显示最后一次超限时的时间戳[日期]和[时间]。
		超限次数	累积的超限次数。
附加温度 (页 235) (例如：前轴承温度)	S6 (KTY)	实际值	测量点上的当前温度。
附加温度报警		限值	保存的温度阈值，超出该阈值会因轴承温度过高而出现报警。
		超限时间	显示累积的超限时间（单位：小时）。
		最后一次超限	显示最后一次超限时的时间戳[日期]和[时间]。
		超限次数	累积的超限次数。
附加温度故障		限值	保存的温度阈值，超出该阈值会因轴承温度过高而出现故障。
		超限时间	显示累积的超限时间（单位：小时）。
		最后一次超限	显示最后一次超限时的时间戳[日期]和[时间]。
		超限次数	累积的超限次数。

8.4.3 电机温度传感器

停止响应

SMI24 中存有电机温度传感器的报警和故障阈值，在操作界面上显示为限值。这两个值既可用于 SMI24 的统计分析，也可用于驱动的停止响应。超出保存在 SMI24 中的报警阈值直到触发故障的延时设置不影响显示的值。

注意：p0604、p0605 的电机参数值（报警/故障阈值）和延迟时间 p0606 是预先定义的，不可修改。

p0606 中的延迟时间不影响操作界面上所显示的值。也就是说：超出报警阈值以及 p0606 中设置的延迟时间届满时，信号“电机温度故障”中所显示的“最后一次超限”值不变，“超限次数”也不会增加。

p0604、p0605 的参数值（报警/故障阈值）和 p0603（延迟时间）会影响主轴的停止响应。超出故障阈值或者超出报警阈值及 p0603 中设置的延迟时间届满后，主轴停止。

当前，延迟时间不由 SMI24 给定。p0603 的参数值显示为出厂设置 (240 s)。

报警：

- 207015 = 电机温度传感器报警
- 207016 = 电机温度传感器故障

8.4.4 附加温度传感器

停止响应

SMI24 中存有附加温度传感器的报警和故障阈值，在操作界面上显示为限值。这两个值仅用于 SMI24 的统计分析。超出保存在 SMI24 中的报警阈值直到触发故障的延时设置不影响显示的值。

注意：p4102[0..1] 的参数值（报警/故障阈值）和 p4103（延迟时间）是预先定义的，不可修改。

修改 p4102[0..1] 的参数值（报警/故障阈值）和 p4103（延迟时间）不影响操作界面上所显示的值。

p4102[0..1] 的参数值（报警/故障阈值）和 p4103（延迟时间）会影响主轴的停止响应。超出故障阈值或者超出报警阈值及 p4103 中设置的延迟时间届满后，主轴停止。为防止主轴意外停止，可对参数值进行相应的调整。

8.4 主轴诊断




当前，延迟时间不由 SMI24 给定。p4103 的参数值显示为出厂设置 (0 s)。也就是说：延时段被取消。

报警：

- 207017 = 超出了附加温度的报警阈值
- 207018 = 超出了附加温度的故障阈值

8.4.5 温度柱形图

柱形图中会显示电机温度和可选的附加温度（例如：轴承温度或机壳温度）。通过颜色来区分以下温度范围：

颜色	含义
	第一限值以内的温度范围。
	第一和第二限值之间的温度范围。
	第二限值以外的温度范围。

单个条形格的高度对应的是设置的温度测量点的温度范围。

指定给单个测量点（传感器）温度范围的时间即主轴旋转时间。

条形格下方会显示当前测量的温度。

8.4.6 转速/转矩

系统可在一条柱形图中显示特定工作小时后主轴的转速和转矩。每种颜色指出了一种状态：

- 淡绿色：轻载
- 绿色： 额度负载
- 黄色： 过载
- 橙色： 重载

单个条形格的高度对应的是基于主轴旋转时间的时间百分比（停留时间）。在各个条形格的内部或上方是相应转速或转矩范围的累积主轴旋转时间次数。条形格的宽度对应的是评估的转速或转矩范围。

柱形图分析

- 转速柱形图可用于分析转速对轴承和回转接头产生的负载。
- 转矩柱形图可用于分析总切削力对轴承产生的负载，但是转矩和径向力之比受到刀具直径（力臂）、铣削弧（各个切削刃产生的主切削力的叠加）和切削力系数（总切削力由主切削力、进给力 and 背向力组成 - 但是系统的转矩图中只显示主切削力）的影响。

8.4.7 夹紧系统

利用“夹紧系统诊断”功能，您可以：

- 快速可靠地查看主轴的夹紧状态
- 监测主轴的夹紧过程和夹紧系统
- 主轴中的传感器 S1 和 S4 用于提供夹具的信息。
- 传感器 S5 用于识别位置，例如：定义的轴旋转角位。

夹紧状态

上半部分窗口中会显示针对三个夹紧状态所设置的限值。

夹紧状态是针对给定的传感器 S1 模拟输出信号的电压范围而定义的。

“最小电压”和“最大电压”两列中的值显示了相应夹紧状态下传感器 S1 的模拟信号所对应的电压范围。

缺省转速限值“最大转速”指定了夹紧状态所允许的最大转速并由驱动对此加以限制。

信号一览

信号一览中会显示传感器、刀具夹头、设置的设定转速和有效转速限值的当前状态值。

信号	传感器	说明	值	含义
状态	S1	夹紧状态	3	已松开
			7 或 8	已夹紧，带刀具
			10	已夹紧，无刀具
活塞松开	S4	活塞询问	是	释放活塞未与轴接触。
			否	释放活塞与轴接触。
轴进入切换位置	S5	位置询问	是	例如：轴位于定义的旋转角位中。
			否	例如：轴不位于定义的旋转角位中。
设定转速		设定转速		由用户设定转速。

信号	传感器	说明	值	含义
状态		夹紧状态	0	传感器 S1 不存在或状态值无效。
			1	状态初始化中。
			2	已松开，发出消息（故障状态）
			3	已松开
			4	夹紧，带刀具
			5	松开，带刀具
			6	松开，无刀具
			7	已夹紧，带刀具且 S4 = 0（否）
			8	已夹紧，带刀具且 S4 = 1（是）
			9	夹紧，无刀具
			10	已夹紧，无刀具
			11	已夹紧，发出消息（故障状态）

状态条形图

夹紧状态以状态条形图显示。绿色区域是已夹紧“带或无刀具”和“刀具已松开”的状态区域。箭头标记显示的是传感器 S1 的当前模拟电压。

文档

有关对应接口信号 (NC → PLC) 的信息请参考以下文档：

- 参数手册之 NC 变量和接口信号；章节“轴/主轴专用信号”

步骤

1. 窗口“主轴诊断 S1”打开。
在没有软件选项“S-Monitor（主轴监视器）”的情况下，窗口“夹紧系统 S1”打开。
2. 按下软键“夹紧系统”。
“夹紧系统 S1”窗口打开，显示系统获得的数据。

转速限值

3. 按下软键“转速限值”，根据夹紧状态调整转速限值。
窗口“转速限值 S1”打开。

4. 按下软键“修改”，根据夹紧状态输入所需的转速限值。
按下“确认”。
-或者-
3. 按下软键“统计”，查看主轴的夹紧时间并对夹紧系统进行诊断。
“夹紧系统统计 S1”窗口打开，显示系统获取的数据。
-或者-
3. 按下软键“返回”，回到数据一览表。

8.4.8 夹紧系统：转速限值

出于安全因素和保护主轴的目的，某些夹紧状态下只允许使用最大转速。驱动会将各个夹紧状态下的主轴转速限定为输入的转速。

夹紧状态	含义
已松开	刀具夹紧系统与释放活塞接触。注意：释放活塞会对夹紧系统施加释放力。
夹紧	刀具夹紧系统从“已松开”状态过渡到“已夹紧，带刀具”状态。
松开，从 “已夹紧，带刀具”状态过渡 而来	刀具夹紧系统从“已夹紧，带刀具”状态过渡到“已松开”状态。
松开，从 “已夹紧，无刀具”状态过渡 而来	刀具夹紧系统从“已夹紧，无刀具”状态过渡到“已松开”状态。
已夹紧，带刀具	刀具夹紧系统处于“已夹紧，带刀具”状态。
夹紧，无刀具	刀具夹紧系统从“已夹紧，带刀具”状态过渡到“已夹紧，无刀具”状态。
已夹紧，无刀具	刀具夹紧系统处于“已夹紧，无刀具”状态。

可通过软键“修改”按照您的要求修改缺省转速限值。

注意
转速过高可损坏主轴
修改缺省转速限值可导致运行状态发生变化，影响安全，进而损坏主轴。

8.4.9 夹紧系统：诊断统计

该窗口中会显示夹紧过程和转速超限的统计数据。

统计数据	含义
正确的夹紧循环次数	累积的正确夹紧循环（带或无刀具）次数。 正确的夹紧循环是：夹紧状态“已松开” [3] → 夹紧状态“已夹紧” [7、8 或 10] → 夹紧状态“已松开” [3]。正确的夹紧循环是指夹紧过程和之后的松开过程是通过相应的夹紧状态（无“状态回退”）进行的。从一个夹紧状态开始，下一个可达到的夹紧状态不可以是前一个夹紧状态。此时不存在时间关系。
夹紧时间（无刀具）	夹紧状态“刀具已松开”和“已夹紧，无刀具”之间的夹紧过程内的时间。
<ul style="list-style-type: none"> 在公差范围内 	设定时间内累积的夹紧过程（无刀具）次数。
<ul style="list-style-type: none"> 在公差范围外 	设定时间外累积的夹紧过程（无刀具）次数。
<ul style="list-style-type: none"> 设定 	夹紧过程（无刀具）时间的出厂设定。
<ul style="list-style-type: none"> 最新 	测出的上一次执行的夹紧过程（无刀具）的时间。
已夹紧（无刀具）	夹紧状态“已夹紧，无刀具”期间的诊断
<ul style="list-style-type: none"> 转速超限 	累积的主轴转速大于出厂时设置的限值的事件次数。
<ul style="list-style-type: none"> 转速阈值 	夹紧状态“已夹紧，无刀具”下的主轴转速限值（出厂值）。 通过诊断一览“夹紧系统 S1” → “转速限值” → “修改” 修改该转速限值不会影响此处给定的值。
夹紧/松开过程错误（整体）	累积的非连续运行夹紧状态：夹紧状态“已松开” [3] → 夹紧状态“已夹紧” [7、8 或 10] → 夹紧状态“已松开” [3] 的夹紧过程次数。 错误的夹紧循环是指：从一个夹紧状态开始，下一个达到的夹紧状态又是前一个夹紧状态（“状态回退”）。

8.4.10 夹紧系统：夹紧时间统计

该窗口中会显示测出的夹紧时间的统计数据。

统计数据	含义
上一次测出的夹紧时间	测出的上一次执行的夹紧过程的时间。 夹紧过程定义：夹紧状态“已松开”[3] → 夹紧状态“已夹紧，带刀具”[7 或 8]。夹紧状态 3 = 已松开；夹紧状态 7 = 已夹紧，带刀具及 S4 = 0（否）；夹紧状态 8 = 已夹紧，带刀具及 S4 = 1（是）
参考夹紧时间（带刀具）	在主轴调试阶段，前 100 个夹紧过程由 SMI24 测定并确定最小值、最大值和平均值。
● 最小值	从前 100 个夹紧过程中测出的最小夹紧时间。
● 最大值	从前 100 个夹紧过程中测出的最大夹紧时间。
● 平均值	从前 100 个夹紧过程中测出的平均夹紧时间。
平均夹紧时间（带刀具）	系统会以前 100 个测出的夹紧时间为参考计算其余测出的夹紧时间的平均值。
● 24 小时	最后 24 小时内测出的夹紧循环的平均夹紧时间。
● 10 天	最后 10 天内测出的夹紧循环的平均夹紧时间。
● 100 天	最后 100 天内测出的夹紧循环的平均夹紧时间。

如果在调试阶段重启了 SMI24，系统会删除迄今为止所测出的值。前 100 个夹紧过程的分子被置零，可以重新测定夹紧过程（最多 100 个）的参考时间。调试阶段（前 100 个夹紧过程）结束之后，参考夹紧时间值被“冻结”。重启 SMI24 后，值仍被保留且不可修改。

8.4.11 查看主轴的物流数据

前提条件

只有“服务”访问级别才可查看主轴数据。

主轴数据

通过 DRIVE-CLiQ 可以查看主轴制造商的下列主轴数据:

- 厂商名称
- 图纸号
- 序列号
- 生产日期
- 服务数据和服务信息(1...4):
服务信息即 WEISS 主轴技术有限责任公司服务报告的过程编号。

刀具管理

9.1 用于刀具管理的机床数据

9.1.1 有/无刀具管理的设置

无刀具管理的设置

如果要采用无 NC 刀具管理的刀具管理，可以进行下列设置：

MD18080 \$MN_MM_TOOL_MANAGEMENT_MA SK	逐步为刀具管理预留存储器(SRAM)。
= 02H	
位 1	提供监控数据。

MD20310 \$MC_TOOL_MANAGEMENT_MASK	激活刀具管理功能。
= 02H	
位 1	刀具管理监控功能生效。

MD17530 \$MN_TOOL_DATA_CHANGE_COUNT ER	标识 HMI 上的刀具数据变更。
= 1FH	
位 0	刀具状态变更
位 1	刀具剩余件数变更
位 2	刀具数据的变更服务
位 3	变更服务中的刀库数据。
位 4	变更服务中的 ISO 数据。

9.1 用于刀具管理的机床数据

MD28450 \$MC_MM_TOOL_DATA_CHG_BUFF_SIZE	刀具数据变更的缓冲器 (DRAM)。
= 100	用于刀具数据 BTSS 变更服务的缓冲器中的条目数量。将使用动态存储器。只有在 MD17530 \$MN_TOOL_DATA_CHANGE_COUNTER 的位 2 或位 3 置位后，才能创建此缓冲器。

MD19320 \$ON_TECHNO_FUNCTION_MASK	
=2000000H	
位 25	使能无刀库管理的刀具管理。

带刀库管理的设置

如果要采用带 NC 刀库管理的刀具管理，可以进行下列设置：

MD18080 \$MN_MM_TOOL_MANAGEMENT_MASK	逐步为刀具管理预留存储器(SRAM)。
= 03H	
位 0	准备刀具管理数据。
位 1	提供监控数据。

MD20310 \$MC_TOOL_MANAGEMENT_MASK	激活刀具管理功能。
= 03H	
位 0	激活刀库管理。
位 1	刀具管理监控功能生效。

MD17530 \$MN_TOOL_DATA_CHANGE_COUNTER	标识 HMI 上的刀具数据变更。
= 1FH	
位 0	刀具状态变更
位 1	刀具剩余件数变更
位 2	刀具数据的变更服务

位 3	变更服务中的刀库数据。
位 4	变更服务中的 ISO 数据。

MD28450 \$MC_MM_TOOL_DATA_CHG_BUFF_SIZE	刀具数据变更的缓冲器 (DRAM)。
= 100	用于刀具数据 BTSS 变更服务的缓冲器中的条目数量。将使用动态存储器。只有在 MD17530 \$MN_TOOL_DATA_CHANGE_COUNTER 的位 2 或位 3 置位后, 才能创建此缓冲器。

MD19320 \$ON_TECHNO_FUNCTION_MASK	
= 10H	
位 4	使能带刀库管理的刀具管理。

9.1.2 配置刀具管理的访问级别

所选参数的访问级别

MD51199 \$MNS_ACCESS_WRITE_TM_GRIND	在刀具管理中写入磨削数据的访问级别
MD51200 \$MNS_ACCESS_WRITE_TM_GEO	在刀具管理中写入几何数据的访问级别
MD51201 \$MNS_ACCESS_WRITE_TM_WEAR	在刀具管理中写入磨损数据的访问级别
MD51202 \$MNS_ACCESS_WRITE_TM_WEAR_DELTA	写入部分刀具磨损数据的访问级别
MD51203 \$MNS_ACCESS_WRITE_TM_SC	在刀具管理中写入总补偿的访问级别
MD51204 \$MNS_ACCESS_WRITE_TM_EC	在刀具管理中写入设置补偿的访问级别
MD51205 \$MNS_ACCESS_WRITE_TM_SUPVIS	在刀具管理中写入监控数据的访问级别
MD51206 \$MNS_ACCESS_WRITE_TM_ASSDNO	在刀具管理中写入唯一的 D 号的访问级别

9.1 用于刀具管理的机床数据

MD51207 \$MNS_ACCESS_WRITE_TM_WGROUP	在刀具管理中写入磨损组（刀库位置/刀库）的访问级别
MD51208 \$MNS_ACCESS_WRITE_TM_ADAPT	在刀具管理中写入刀具适配器几何数据的访问级别
MD51209 \$MNS_ACCESS_WRITE_TM_NAME	在刀具管理中写入刀具名称和双号的访问级别
MD51210 \$MNS_ACCESS_WRITE_TM_TYPE	在刀具管理中写入刀具类型的访问级别

各个参数的访问级别可以通过条目“AccessLevel”中的标签
<PARAMETERCONFIGURATION>加以修改，参见章节配置列表参数 (页 275)。

窗口“其他详细信息”的访问级别

MD51198 \$MNS_ACCESS_READ_TM_ALL_PARAMETERS	在刀具管理中读取“详细信息 - 全部参数”的保护级别 “其他详细信息”按钮会根据机床数据显示或隐藏。
MD51215 \$MNS_ACCESS_WRITE_TM_ALL_PARAMETERS	在刀具管理中写入“详细信息 - 全部参数”的访问级别

针对特定刀具管理功能的访问级别

MD51216 \$MNS_ACCESS_TM_TOOL_CREATE	在刀具管理中创建刀具的访问级别
MD51217 \$MNS_ACCESS_TM_TOOL_DELETE	在刀具管理中删除刀具的访问级别
MD51218 \$MNS_ACCESS_TM_TOOL_LOAD	在刀具管理中载入刀具的访问级别
MD51219 \$MNS_ACCESS_TM_TOOL_UNLOAD	在刀具管理中卸载刀具的访问级别
MD51220 \$MNS_ACCESS_TM_TOOL_MOVE	在刀具管理中移动刀具的访问级别
MD51221 \$MNS_ACCESS_TM_TOOL_REACTIVATE	在刀具管理中重新激活刀具的访问级别
MD51222 \$MNS_ACCESS_TM_TOOL_MEASURE	在刀具管理中测量刀具的访问级别
MD51223 \$MNS_ACCESS_TM_TOOL_EDGE_CREATE	在刀具管理中创建刀沿的访问级别

MD51224 \$MNS_ACCESS_TM_TOOLEEDGE_DEL ETE	在刀具管理中删除刀沿的访问级别
MD51225 \$MNS_ACCESS_TM_MAGAZINE_POS	在刀具管理中定位刀库的访问级别

9.1.3 其他设置

其他设置

通过以下机床数据/设定数据可以使能操作界面上的其他功能：

MD52270 \$MCS_TM_FUNCTION_MASK	
位 0	不允许在刀库位置上创建刀具。
位 1	机床未处于复位状态时，禁止装刀卸刀。
位 2	急停时，禁止装刀卸刀。
位 3	禁止刀具装入主轴、从主轴卸出或移动刀具。
位 4	直接装入主轴。
位 5	使用磨削配置文件。
位 6	所有设置了禁止（见位 3），但仍允许刀具装入主轴、从主轴卸出或移动刀具。
位 7	通过 T 号创建刀具。
位 8	隐藏“刀具移位”。
位 9	隐藏“定位刀库”。
位 10	刀具重新激活，带刀库定位。
位 11	重新激活刀具的所有监控类型。
位 12	隐藏“刀具重新激活”。

SD54215 \$SNS_TM_FUNCTION_MASK_SET	
位 0	在以下类型的刀具上，刀沿参数 6（刀沿半径）和 15（磨损半径）不用半径值，而用直径值显示：型号 100 至 299、580、710、711、712、713、714 和 725。切割参数 7（外径）在以下刀具类型中显示为直径值，而不是半径值：刀具 140 和 714。
位 1	所有车刀的标准转动方向为 M4。创建车刀时，转动方向默认为 M4。
位 2	在创建刀具时不提供建议名称。

9.1 用于刀具管理的机床数据

位 3	禁止输入已装入的刀具的名称和类型。 刀具装入后，不能再修改它的名称和类型。
位 4	通道未处于复位状态时，禁止输入已装入的刀具。
位 5	累计计算刀具磨损值： 输入的磨损数值累加在已有磨损数值上。
位 6	在“刀具名称”栏中只允许输入数字。
位 7	隐藏刀具监控参数。 在操作界面上隐藏刀具监控参数。
位 8	横轴几何尺寸为直径值。 在机床数据 MD20100 \$DIAMETER_AX_DEF 中定义了一根横轴、设定数据 SD42940 \$TOOL_LENGTH_CONST 设为 18 且 SD42950 \$TOOL_LENGTH_TYPE 设为 2 时，横轴的几何值为直径值。
位 9	横轴磨损量为直径值。 在机床数据 MD20100 \$DIAMETER_AX_DEF 中定义了一根横轴、设定数据 SD42940 \$TOOL_LENGTH_CONST 设为 18 且 SD42950 \$TOOL_LENGTH_TYPE 设为 2 时，横轴的磨损量为直径值。
位 10	使能至缓冲区刀位的刀具装载/换位。 可在装载对话框中输入刀库编号。 可使用刀库编号 9998 访问缓冲区。
位 11	禁止在抓刀器刀位上创建新刀具。
位 12	在功能“全部卸载”中不卸载测量刀具。

关联性

设定数据 SD54215 \$TM_FUNCTION_MASK_SET 和机床数据 MD20360 \$TOOL_PARAMETER_DEF_MASK 存在以下关联：

关联性		说明	建议
SD54215	↔	MD20360	
位 9	↔	位 0	我们不建议使用 MD20360 位 0 和位 1。 建议您使用 SD4215 位 8 和位 9。
位 8	↔	位 1	MD20360 位 0 和位 1 只能用于车刀和磨削刀具。 这两个位不能在车铣加工中心上使用，因为车刀在横轴上的长度分量是直径值，而铣刀在横轴上的长度分量是半径值。 而在使用设定数据时，只有屏幕上显示的是直径值，NC 内部实际上是半径值。

关联性		说明	建议
SD54215	↔ MD20360		
位 0	↔ 位 11	MD20360 位 11 置位后，SD54215 位 0 对于刀沿参数 6（刀沿半径）来说无效，因为 NC 已经将刀沿半径作为直径值提供。	我们不建议使用 MD20360 位 11 和位 12。 建议设置 SD54215 位 0。 MD20360 位 11 和位 12 可用于所有类型的刀具，其中也包括车刀。该设置在车床/铣床是不可用的，因为车刀的切割半径不可能显示为直径。 在使用该设置数据时仅显示为直径，NC 的内部值保持为半径。
位 0	↔ 位 12	MD20360 位 12 置位后，SD54215 位 0 对于刀沿参数 15（磨损半径）来说无效，因为 NC 已经将磨损半径作为直径值提供。	

使能图形化的刀具显示和刀库显示

如需图形显示刀具和刀库，请做以下设置：

MD52271 \$MCS_TM_MAG_PLACE_DISTANCE	单个刀库位置间隔的距离，单位采用 NC 的单位制。
> 0	刀具和刀库位置以图形显示在屏幕上。
= 0	刀具和刀库位置不以图形显示在屏幕上。

在操作界面上进行以下设置：

1. 在“刀具列表”窗口按“设置”键。
“用户确认”窗口打开。
2. 激活“显示工具提示”的复选框。
只有当机床数据 MD52271 \$MCS_TM_MAG_PLACE_DISTANCE 的值大于 0 时才出现此复选框。

边界条件：

- MD52200 \$MCS_TECHNOLOGY = 2
- MD52201 \$MCS_TECHNOLOGY_EXTENSION = 0

装刀卸刀以及换刀的默认设置

MD52272 \$MCS_TM_TOOL_LOAD_DEFAULT_MAG	装刀的默认刀库。
= 0	无默认刀库，标准值

9.1 用于刀具管理的机床数据

MD52273 \$MCS_TM_TOOL_MOVE_DEFAULT_MAG	换刀的默认刀库。
= 0	无默认刀库，标准值

MD52274 \$MCS_TM_TOOL_LOAD_STATION	装刀站编号。
= 0	考虑所有配置站，标准值

磨损量的输入限制

MD51212 \$MNS_TM_WRITE_WEAR_ABS_LIMIT	最大刀具磨损量。
MD51213 \$MNS_TM_WRITE_WEAR_DELTA_LIMIT	和最大刀具磨损量之间的最大差值
MD51214 \$MNS_TM_WRITE_LIMIT_MASK	刀具磨损量输入限制的生效范围

手动刀具功能设置

MD22562 \$MC_TOOL_CHANGE_ERROR_MODE	刀具切换故障时的操作。
位 1 = 1	可换入手动刀具。
位 2 = 1	激活的 D 号 > 0 且激活的 T 号 = 0 时会发出报警信息。 激活的 D 号 > 0 且激活的 D 号 = 0 时会发出报警信息。

说明

如果设置了机床数据 MD22562 \$MC_TOOL_CHANGE_ERROR_MODE 位 2 = 1，则可以通过程序中的补偿选择（D 号）暂停机床，直到 PLC 报告刀具切换已结束。前提条件是要在设置机床数据 MD20270 \$MC_CUTTING_EDGE_DEFAULT = 0。

9.2 配置操作界面

可调整的设置

可以进行以下配置：

- 一般设置
- 配置刀具列表
- 配置列表参数
- 配置刀具类型
- 配置“其它数据”窗口
- 配置窗口“新建刀具 - 收藏”
- 配置窗口“新建刀具”
- 配置新刀具的默认值
- 配置“详细信息”窗口
- 命名刀库和刀库位置
- 设定刀库位置类型的名称
- 确定刀库和通道的对应关系
- 冷却剂和刀具专用功能
- 重新激活时的换刀原因
- 配置载码器连接

下面将详细介绍如何开展上述配置。

使用磨削工艺时需要满足的前提条件

选择工艺“通用”并设置：

MD52270 \$MN_TM_FUNCTION_MASK Bit 5 = 1。

现在在操作界面上只显示磨削刀具供选择。

步骤

1. 打开目录： **/siemens/sinumerik/hmi/template/cfg.**
2. 复制模板文件“oem_sltmlistconfig.xml”。

说明

用于磨削工艺的其他模板

文件“sltmaddgrindinglistconfig.xml” 用于在车削配置中增加磨削刀具。 从以下目录中复制该文件： **/siemens/sinumerik/hmi/template/cfg.**

3. 将文件复制到目录“/oem/sinumerik/hmi/cfg” 下。
或者 **/user/sinumerik/hmi/cfg.**
4. 根据设置的工艺修改文件名称：

文件名	工艺
sltmlistconfig.xml	铣削工艺
sltmturninglistconfig.xml	车削工艺
sltmplclistconfig.xml	PLC 刀具管理 "TRANSLINE 2000"
sltmcirculargrindinglistconfig.xml	外圆磨削工艺
sltmsurfacegrindinglistconfig.xml	平面磨削工艺

5. 在模板中删除示例的注释， 以便进行自定义。

配置示例

模板“oem_sltmlistconfig.xml” 中包含了所有可能出现的配置示例， 下面一一介绍这些示例。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
- <CONFIGURATION>
<!-- *****-->
<!-- enter your configuration behind this comment -->
<!-- ***** -->
```

配置示例

```
<!-- ***** -->
<!-- enter your configuration ahead this comment -->
<!-- ***** -->
</CONFIGURATION>
```

利用下面的 xml 标签来输入配置：

标签<tag>	作用于以下设置：
<SETTINGS>	一般设置 (页 253)
<LISTCONFIGURATION>	配置刀具列表 (页 259)
<PARAMETERCONFIGURATION>	配置列表参数 (页 275)
<TOOLTYPECONFIGURATION>	配置刀具类型 (页 283)
<MOREDATACONFIGURATION>	配置“其他数据”窗口 (页 285)
<NEWTOLFAVORITECONFIGURATION>	配置窗口“新刀具 - 收藏” (页 286)
<NEWTOLCONFIGURATION>	配置窗口“新建刀具” (页 286)
<NEWTOLDEFAULTVALUECONFIGURATION>	配置新刀具的默认值 (页 287)
<DETAILSCONFIGURATION>	配置“详细信息”窗口 (页 292)
<MAGAZINEPLACENAMECONFIGURATION>	命名刀库和刀库位置 (页 296)
<PLACETYPECONFIGURATION>	设定刀库刀位类型的名称 (页 298)
<CHANNELMAGAZINEASSIGNMENT>	确定刀库和通道的对应关系 (页 299)
<TOOLCHANGEREASON>	重新激活时的换刀原因 (页 305)
<IDENTCONNECTIONCONFIGURATION>	配置载码器连接 (页 309)

9.2.1 一般设置

标签<SETTINGS>

标签<SETTINGS>可以定义“刀具管理”操作界面的通用设置。

条目	含义
ReactivateWithMagPos	true - 重新激活，并将刀具装入刀具刀库的装载位。 false - 缺省设置，重新激活，不带刀具刀库定位。
ReactivateAllMonitorModes	true - 复位所有在 NC 中设置的监控类型的实际值。 false - 缺省设置，复位已激活监控类型的实际值。
ReactivateEnabled	true - 缺省设置，“重新激活”使能。 false - “重新激活”禁用。

条目	含义
CreateNewToolDialog	<p>true - 显示窗口“新刀具”。在该窗口中，可在实际创建刀具前输入刀具名称和刀具大小（超大刀具）。更多输入方法可通过“新刀具”窗口配置确定。</p> <p>false - 缺省设置，不显示窗口“新刀具”。输入名称后直接在刀具表中创建刀具。</p>
CreateNewMultiToolDialog	<p>true - 缺省设置，显示窗口“新建多刀”。在此窗口中可在实际创建多刀前进行如下配置：</p> <ul style="list-style-type: none"> ● 多刀名称 ● 多刀刀位数量 ● 间距输入方式 ● 每个多刀刀位的刀位间距值 <p>false - 不显示窗口“新建多刀”。输入名称后直接在刀具表中创建多刀。</p>
MagazineSelectionDialog	<p>true - 出现显示刀具缓冲区、刀具刀库和 NC 刀具的对话框。每项都有标记，使用此标记可在列表显示中隐藏或显示相应的项。通过软键“转到”可直接选择列表显示中的项。</p> <p>false - 无刀库选择对话框。在刀具缓冲区、刀具刀库和 NC 存放区之间切换。</p> <p>如果标签<Settings>中缺少条目 MagazineSelectionDialog，则在刀库配置含多个刀具刀库时（不含系统刀库）会自动出现对话框。如果仅有一个刀具刀库，使用切换功能。</p>
NewToolFavoritesOnly	<p>涉及到功能“新建刀具”。</p> <p>true - 只提供收藏中配置的刀具类型。</p> <p>false - 缺省设置，提供所有刀具类型。</p>
SortationInFirstEtcLevel	<p>true - 排序功能将分布在第一个软键级中。功能“其他”取消。</p> <p>false - 缺省设置，排序功能在所有软键级中。</p>
ToolBufferOnceOnTop	<p>true - 缺省设置，在列表的开始处显示刀具缓冲区。</p> <p>false - 显示每个刀库的缓冲区。其中，只显示分配到该刀库的刀位。</p>

条目	含义
UnloadToolFromLoadPlaceEnabled	<p>true – 可通过软键“卸载”来移除装载位上的刀具。</p> <p>false - 缺省设置。不在位于装载位上的刀具处显示软键“卸载”。</p>
AutoHNumberDisplayEnabled	<p>刀沿参数列“IsoHNoDPH - ISO H 号”取决于机床的配置。</p> <p>true - 缺省设置，自动显示此列。</p> <p>false - 不会自动显示此列。</p>
AutoUniqueDNumberDisplayEnabled	<p>刀沿参数列“UniqueDNo”取决于机床的配置。</p> <p>true - 缺省设置，自动显示此列。</p> <p>false - 不会自动显示此列。</p>
AccessLevelDeleteAllDNos	<p>该条目确定从哪个访问级别起系统会提供功能“删除所有 D 号”。</p> <p>缺省设置：访问级别 3（用户），设置了“唯一性 D 号”时，该条目无效。</p>
MagazineNumberOfDCheckAndActivate	<p>D-Check（检查唯一性 D 号）和 Acitvate（激活磨损数据组）作用于的刀库号。</p> <p>缺省设置：-2</p> <p>含义：所有和刀夹或主轴相连的刀库。</p>
AutoFixedPlaceDisplayEnabled	<p>刀具参数列“ToolStateFixed”（刀具状态 - 刀具为固定刀位编码）取决于刀库配置生效</p> <p>true - 缺省设置，在配置的所有刀库都为固定刀位编码时，自动隐藏列。</p> <p>false - 不自动隐藏列。</p>
MagPlaceAdapterDisplayEnabled	<p>在“刀库”窗口中显示“刀库位置适配器参数”一列。</p> <p>false - 缺省设置，不显示参数。</p>
AdaptTransDataViewEnabled	<p>使能经过刀库位置适配器转换后的刀沿参数显示。该显示可以在刀具表的设置对话框中使能和关闭。</p> <p>false - 缺省设置，不使能经过刀库位置适配器转换后的刀沿参数显示。</p>
ActiveCompensationsEnabled	<p>使能功能“加工补偿”。该功能取决于机床配置。唯一性 D 号、刀库位置适配器和刀库管理必须先设置。</p> <p>false - 缺省设置，不使能功能“加工补偿”。</p>

条目	含义
M2NLockEnabled	在执行刀具管理功能时，封锁“M 到 N”的切换。 true - 缺省设置，封锁切换。
AccessLevelWriteDetailsAllParameter	该条目可确定从哪个访问级别起可在窗口“详细信息 - 所有参数”中写入数据。 缺省设置：访问级别 7（钥匙开关位置 0），可以通过 MD51215 加以修改。
AccessLevelReadDetailsAllParameter	该条目确定从哪个访问级别起在“详细信息”窗口中提供功能“其他详细信息”。 缺省设置：访问级别 7（钥匙开关位置 0），可以通过 MD51198 加以修改。
FixNumberOfMultiToolPlaces	利用固定刀位数创建多刀。创建多刀时不弹出询问消息。
FixKindOfDistInMultiTools	利用固定的刀位间距创建多刀。创建多刀时不弹出询问消息。 允许的输入值：3，多刀刀位的间距，为角度值。
DIDisplayMode	<ul style="list-style-type: none"> • DisplayDistributed - 缺省设置，设置补偿 EC 显示在刀具表中，而总补偿 SC 显示在刀具磨损表中。 • EclnWearList - 设置补偿 EC 显示在刀具磨损表中，而总补偿 SC 不显示。
CombinedToolMonitoring	确定下拉菜单中提供的组合监控方式的选项： <ul style="list-style-type: none"> • Time/Count - 寿命监控和件数监控 • Time/Wear - 寿命监控和磨损监控 • Count/Wear - 件数监控和磨损监控 • Time/Count/Wear - 寿命监控、件数监控和磨损监控

条目	含义
FiltersCombinedWithOr	<p>确定在筛选窗口中逻辑“或”连接的筛选条件。所有筛选条件的默认设置为逻辑“与”。允许的输入为:</p> <ul style="list-style-type: none"> • “StandBy” - 用于可用的刀具 • “Active” - 用于带激活标识的刀具 • “Locked” - 用于禁用的刀具 • “Prewarn” - 用于预警极限 • “RemainingPieces” - 用于剩余工件数目 • “RemainingTime” - 用于剩余寿命 • “MarkedToUnload” - 用于卸刀标识 • “MarkedToLoad” - 用于装刀标识
MagazineMoveMessage	<p>true - 显示“刀库移动”或者“刀库结束移动”的信息。</p> <p>false - 缺省设置，没有显示信息。</p>
ToolLoadErrorMessage	<p>涉及 PLC 故障时故障信息的输出。</p> <p>true - 缺省设置，显示故障信息。</p> <p>false - 不显示故障信息。</p>

下列设置和列表中的标记相关:

true - 显示标记。

false - 不显示标记（缺省设置）。

如果多个标记适用于一个刀位，则显示最重要的信息。比如，在刀具已禁用且此刀具的“刀具位置”和“夹角参考方向”不匹配时，会显示红色的叉，表示刀具已禁用。不会看见表示出现冲突的黄色三角形。

条目	含义
ShowSymbolActivePlace	<p>标记加工位置上的刀位。</p> <p>true - 缺省设置适用于“刀塔”型刀库。</p> <p>false - 所有其它刀库类型的设置。</p>
ShowSymbolActiveMultiToolPlace	<p>标记加工位置上的多刀刀位。多刀刀位只在多刀位于主轴或刀夹中时才显示。</p> <p>true - 缺省设置。</p>

条目	含义
ShowSymbolLoadPlace	<p>标记装载位置上的刀位。</p> <p>true - 适用于所有刀库类型的缺省设置（除了刀塔之外，都是 true 值）。</p> <p>false - 适用于“刀塔”型刀库的设置。</p>
ShowSymbolChangeInSpindlePlace	<p>标记换刀位置上的刀位。</p> <p>true - 适用于所有刀库类型的缺省设置（除了刀塔之外，都是 true 值）。</p> <p>false - 适用于“刀塔”型刀库的设置。</p>
ShowSymbolActiveTool	<p>标记生效的刀具。</p> <p>false - 缺省设置。</p>
ShowSymbolProgrammedTool	<p>标记后续刀具。</p> <p>true - 缺省设置。</p>
ShowSymbolActiveDNo	<p>标记生效的刀沿。</p> <p>false - 缺省设置。</p>
ShowSymbolActiveDNoInTool	<p>通过标记生效的刀沿标记生效的刀具。</p> <p>true - 缺省设置，不带刀库管理的配置。</p> <p>false - 带刀库管理配置的设置。</p>
ShowSymbolActiveDINo	<p>标记生效的 DI 号。</p> <p>true - 缺省设置。</p>
ShowConflictPositionAndDirection	<p>标记冲突，即：刀具位置和切削方向不相符。</p> <p>true - 缺省设置。</p>
ShowConflictActiveToolWarning	<p>标记冲突，即“刀塔”型刀库和生效的刀具不在加工位置上。</p> <p>true - 缺省设置。</p>
ShowSymbolActiveIsoHNoL1	<p>标记 ISO 刀具列表中生效的长度 1 的 H 号。</p> <p>true: 缺省设置。</p>
ShowSymbolActiveIsoHNoL2	<p>标记 ISO 刀具列表中生效的长度 2 的 H 号。</p> <p>true - 缺省设置。</p>
ShowSymbolActiveIsoHNoL3	<p>标记 ISO 刀具列表中生效的长度 3 的 H 号。</p> <p>true - 缺省设置。</p>

条目	含义
ShowSymbolActiveIsoDNo	标记 ISO 刀具列表中生效的 D 号。 true - 缺省设置。
ShowSymbolActiveIsoHDNo	标记 ISO 刀具列表中生效的 HD 号。 true - 缺省设置。

示例

```
<SETTINGS>
  <ReactivateWithMagPos value="false" type="bool" />
  <MagazineMoveMessage value="false" type="bool" />
  <CreateNewToolDialog value="true" type="bool" />
  <MagazineSelectionDialog value="false" type="bool" />
  <AccessLevelWriteDetailsAllParameter value="4" type="int" />
</SETTINGS>
```

9.2.2 配置刀具列表

标签<LISTCONFIGURATION>

在标签<LISTCONFIGURATION>中有用于以下列表视图配置的输入：

- 窗口“刀具表”，标签<SlTmToollistForm>，见示例 1。
- 窗口“刀具磨损”，标记<SlTmToollistForm>
- 窗口“刀库”，标记<SlTmToolmagazinForm>
- 窗口“刀具数据 OEM”，标记<SlTmTooloemForm>
在标准配置中，窗口被隐藏。只有通过<Enabled>使能，才显示该窗口，见示例 3。
- 窗口“刀具选择”，标记<SlTmToolselectionForm>：可在程序输入屏幕中显示此窗口，用于选择刀具。

条目	含义
Enabled	true - 显示窗口。 false - 隐藏窗口。
CAPTION	确定列表名称。 参见示例 1：对于标记“TM_CAPTION_TOOLLIST”在操作界面上显示“刀具列表”。
COLUMNX	列选择，X 表示列号。
TOOLTYPE_ALL	列设置适用于所有刀具类型。
TOOLTYPE_XXX	列设置适用于某一特定刀具类型。XXX 表示刀具类型号。
EMPTY_MAG_PLACE	列设置适用于空位。空位出现在按刀库排序的列表中。
Item	列表参数的标记。 参见示例：对于标记“ToolInMagInPlace”将创建一列，用于显示刀库号和位置。 刀具参数标识 (页 264) 刀沿参数的标识 (页 267) 监控参数的标识 (页 269) 刀位参数的标识 (页 271)
DELETE_COLUMN	删除列。
INSERT_COLUMN	插入列。

说明

输入按以下顺序生效（与当前编程无关）：

1. COLUMNX
2. DELETE_COLUMN
3. INSERT_COLUMN

说明

Enabled 条目不适用于“刀具选择”窗口，其他条目生效。

示例 1: 配置刀库列表 - 刀库位置和刀具位置类型

说明

刀库列表配置中的列不能每次都来自界面上的显示，因为有些列是自动添加的，没有加入配置的计数中。

型式 1:

列 10 和列 11 被覆写。如果没有列 10 和列 11，新的列会加入到现有列中。

```
<LISTCONFIGURATION>
  <!-- **** 刀库列表 **** -->
  <SlTmToolmagazinForm>
    <!-- **** 列 10 **** -->
    <COLUMN10>
      <!-- **** 用于所有类型的刀具 **** -->
      <TOOLTYPE_ALL>
        <!-- **** 显示刀库位置类型 **** -->
        <Item value="MagPlaceType" type="QString" />
      </TOOLTYPE_ALL>
    </COLUMN10>
    <!-- **** 列 11 **** -->
    <COLUMN11>
      <!-- **** 用于所有类型的刀具 **** -->
      <TOOLTYPE_ALL>
        <!-- **** 显示刀具位置类型 **** -->
        <Item value="ToolPlaceSpec" type="QString" />
      </TOOLTYPE_ALL>
      <!-- **** 用于所有多刀 **** -->
      <TOOLTYPE_9997>
        <!-- **** 显示所有多刀位置类型 **** -->
        <Item value="MultiToolPlaceSpec" type="QString" />
      </TOOLTYPE_9997>
    </COLUMN11>
  </SlTmToolmagazinForm>
</LISTCONFIGURATION>
```

型式 2:

新列插入到标准配置的列“刀库位置被禁止”和列“刀具过大”之间。

```

<LISTCONFIGURATION>
  <!-- **** 刀库列表 **** -->
  <SlTmToolmagazinForm>
    <!-- **** 列已插入 **** -->
    <INSERT_COLUMN>
      <!-- **** 列 7 **** -->
      <COLUMN7>
        <!-- **** 用于所有类型的刀具 **** -->
        <TOOLTYPE_ALL>
          <!-- **** 显示刀库位置类型 **** -->
          <Item value="MagPlaceType" type="QString" />
        </TOOLTYPE_ALL>
      </COLUMN7>
      <!-- **** 列 8 **** -->
      <COLUMN8>
        <!-- **** 用于所有类型的刀具 **** -->
        <TOOLTYPE_ALL>
          <!-- **** 显示刀具位置类型 **** -->
          <Item value="ToolPlaceSpec" type="QString" />
        </TOOLTYPE_ALL>
        <!-- **** 用于所有多刀 **** -->
        <TOOLTYPE_9997>
          <!-- **** 显示所有多刀位置类型 **** -->
          <Item value="MultiToolPlaceSpec" type="QString" />
        </TOOLTYPE_9997>
      </COLUMN8>
    </INSERT_COLUMN>
  </SlTmToolmagazinForm>
</LISTCONFIGURATION>

```

示例 2：配置刀库列表 - 删除列“刀具过大”**说明**

刀库列表配置中的列不能每次都来自界面上的显示，因为有些列是自动添加的，没有加入配置的计数中。

```
<LISTCONFIGURATION>
  <!-- **** 刀库列表 **** -->
  <SlTmToolmagazinForm>
    <!-- **** 删除列 7 **** -->
    <DELETE_COLUMN>
      <COLUMN7 />
    </DELETE_COLUMN>
  </SlTmToolmagazinForm>
</LISTCONFIGURATION>
```

示例 3：使能 OEM 刀具数据列表

```
<SlTmTooloemForm>
  <Enabled value="true" type="bool" />
</SlTmTooloemForm>
```

结果

- 使能窗口“OEM 刀具数据”。 屏幕上自动显示下图展示的列。
- 出现软键“OEM 刀具”，并可操作。



9.2.2.1 刀具参数标识

刀具参数

在配置文件中使用了以下刀具参数的标记:

标记	参数和属性	系统变量*)
Empty	空栏	-
ToolNo	刀具编号	-
AdaptNo	夹具号分配	\$TC_MPP8
DuploNo	双编号	\$TC_TP1
DuploNoRW	双号, 可读写	\$TC_TP1
NumCuttEdges	刀具刀沿数量	\$P_TOOLND
ToolIdent	刀具名称	\$TC_TP2
ToolIdentRO	刀具名称,只读	\$TC_TP2

标记	参数和属性	系统变量 ^{*)}
ToolInfo	刀具信息	\$TC_TP11
ToolInMag	刀具位于的刀库	\$A_TOOLMN
ToolInPlace	刀具位于的刀库位置	\$A_TOOLMLN
ToolInMagInPlace	刀库编号/位置	\$A_TOOLMN / \$A_TOOL_MLN
ToolMonTCW	刀具监控类型, 寿命, 件数和磨损。如果磨损监控未通过机床数据激活, 则该参数和 ToolMonTC 一样。	\$TC_TP9
ToolMonTC	刀具监控类型, 寿命和件数	\$TC_TP9
ToolPlaceSpec	位置类型	\$TC_TP7
ToolPlaceSpecIdent	应显示为文本的刀位类型。前提条件是已为刀位类型配置了文本。	\$TC_TP7
ToolSearch	替换刀具的刀具查找类型	\$TC_TP10
ToolMyMag	刀具所属的刀库	\$A_MYMN
ToolMyPlace	刀具所属的刀库位	\$A_MYMLN
ToolSizeLeft	左半位的刀具尺寸	\$TC_TP3
ToolSizeRight	右半位的刀具尺寸	\$TC_TP4
ToolSizeUpper	上半位的刀具尺寸	\$TC_TP5
ToolSizeDown	下半位的刀具尺寸	\$TC_TP6
ToolOverSize	刀具尺寸为固定设置 - 两个左半位、两个右半位、一个上半位、一个下半位	\$TC_TP3 - \$TC_TP6
ToolState	以十六进制方式表示的刀具状态	\$TC_TP8
ToolStateActiv	刀具状态 - 刀具当前有效	\$TC_TP8, 位 0
ToolStateEnabled	刀具状态 - 刀具已释放	\$TC_TP8, 位 1
ToolStateLocked	刀具状态 - 刀具禁止	\$TC_TP8, 位 2
ToolStateLockedRO	刀具状态 - 刀具禁止, 只读	\$TC_TP8, 位 2
ToolStateMeasured	刀具状态 - 刀具已测量	\$TC_TP8, 位 3
ToolStatePrewarn	刀具状态 - 刀具已达到预警极限	\$TC_TP8, 位 4
ToolStatePrewarnRO	刀具状态 - 刀具已达到预警极限, 只读	\$TC_TP8, 位 4
ToolStateInChange	刀具状态 - 换刀中	\$TC_TP8, 位 5
ToolStateFixed	刀具状态 - 刀具固定在编码刀位上	\$TC_TP8, 位 6
ToolStateUsed	刀具状态 - 刀具已使用	\$TC_TP8, 位 7

标记	参数和属性	系统变量 ^{*)}
ToolStateAutoReturn	刀具状态 - 自动返回	\$TC_TP8
ToolStateIgnoreLocked	刀具状态 - 忽视禁用	\$TC_TP8, 位 9
ToolStateMarkedToUnload	刀具状态 - 刀具标记为待卸载	\$TC_TP8, 位 10
ToolStateMarkedToLoad	刀具状态 - 刀具标记为待装载	\$TC_TP8, 位 11
ToolStatePermanent	刀具状态 - 刀具为主刀具	\$TC_TP8, 位 12
ToolState1To1Exchange	刀具状态 - 一对一交换	\$TC_TP8, 位 14
ToolStateHandTool	刀具状态, 手动刀具	\$TC_TP8, 位 15
ToolProtAreaFile	确定保护区域的刀具描述文件的名称	\$TC_TP_PROTA
ToolMaxVelo	最大转速	\$TC_TP_MAX_VELO
ToolMaxAcc	最大加速度	\$TC_TP_MAX_ACC
ToolInMultiTool	刀具位于的多刀	\$A_TOOLMTN
ToolInMultiToolPlace	刀具位于的多刀刀位	\$A_TOOLMTLN
ToolMyMultiTool	刀具所属的多刀	\$A_MYMTN
ToolMyMultiToolPlace	刀具所属的多刀刀位	\$A_MYMTLN
ToolAlarmsExtended	PLC 刀具状态 - 功能“报警延迟”有效	-
ToolAlarmLimit	PLC 刀具状态 - 达到报警极限	-
ToolExtAlarmLimit	PLC 刀具状态 - 达到“报警延迟”极限	-
ToolUser_1, ... ToolUser_10	OEM 刀具参数 1 到 10	\$TC_TPC1, ... \$TC_TPC10
ToolApp_1, ... ToolApp_10	西门子应用刀具参数 1 到 10	\$TC_TPCS1, ... \$TC_TPCS10

*) 参数和系统变量一致，换句话说，系统变量构成了参数的基础。

说明

刀具名称“ToolIdent”

刀具名称的最大长度为 31 个 ASCII 字符。

当使用亚洲字符或 Unicode 字符时字符数要相应减少。

不允许使用下列特殊字符： | # "

9.2.2.2 刀沿参数的标识

在配置文件中使用了以下刀沿参数的标记:

标记	参数和属性	系统变量 ^{*)}
EdgeNo	D 号	-
EdgeNoIdx	刀沿号	-
ToolType	刀具类型, 刀沿参数 1	\$TC_DP1
CuttEdgePos	刀沿位置, 刀沿参数 2	\$TC_DP2
GeoLength1	长度 1, 刀沿参数 3	\$TC_DP3
GeoLengthGeoAx1	几何轴 1 长度, 刀沿参数 3, 例如: 长度 X	\$TC_DP3
GeoLength	长度刀沿参数 3	\$TC_DP3
GeoLength2	长度 2, 刀沿参数 4	\$TC_DP4
GeoLengthGeoAx3	几何轴 3 长度, 刀沿参数 4, 例如: 长度 Z	\$TC_DP4
GeoLength3	长度 3, 刀沿参数 5	\$TC_DP5
GeoLengthGeoAx2	几何轴 2 长度, 刀沿参数 5, 例如: 长度 Y	\$TC_DP5
GeoRadius	半径, 刀沿参数 6	\$TC_DP6
GeoCornerRadius	倒圆半径, 刀沿参数 7	\$TC_DP7
GeoOutsideRadius	外圆半径, 刀沿参数 7	\$TC_DP7
GeoBoomLength	吊臂长度, 刀沿参数 7	\$TC_DP7
GeoLength4	长度 4, 刀沿参数 8	\$TC_DP8
PlateLength	刀片长度, 刀沿参数 8	\$TC_DP8
GeoLength5	长度 5, 刀沿参数 9	\$TC_DP9
GeoWidth	宽度, 刀沿参数 9	\$TC_DP9
PlateWidth	刀片宽度, 刀沿参数 9	\$TC_DP9
GeoPitch	螺距, 刀沿参数 9	\$TC_DP9
BoreRadius	钻孔半径, 刀沿参数 9	\$TC_DP9
GeoAngle1	角度 1, 刀沿参数 10	\$TC_DP10
HolderAngle	夹角, 刀沿参数 10	\$TC_DP10
CorrectionAngle	用于探头校准的补偿角度, 刀沿参数 10	\$TC_DP10
GeoAngle2	角度 2, 刀沿参数 11	\$TC_DP11
AngleConicalMillTool	圆锥形铣削刀具角, 刀沿参数 11	\$TC_DP11
CuttDirection	夹角参考方向, 刀沿参数 11	\$TC_DP11

9.2 配置操作界面

标记	参数和属性	系统变量 ^{*)}
WearLength1	磨损长度 1, 刀沿参数 12	\$TC_DP12
WearLengthGeoAx1	几何轴 1 磨损长度, 刀沿参数 12, 例如: Δ 长度 X	\$TC_DP12
WearLength	磨损长度, 刀沿参数 12	\$TC_DP12
WearLength2	磨损长度 2, 刀沿参数 13	\$TC_DP13
WearLengthGeoAx3	几何轴 3 磨损长度, 刀沿参数 13, 例如: Δ 长度 Z	\$TC_DP13
WearLength3	磨损长度 3, 刀沿参数 14	\$TC_DP14
WearLengthGeoAx2	几何轴 2 磨损长度, 刀沿参数 14, 例如: Δ 长度 Y	\$TC_DP14
WearRadius	磨损半径, 刀沿参数 15	\$TC_DP15
WearCornerRadius	倒圆半径磨损, 刀沿参数 16	\$TC_DP16
WearLength4	磨损长度 4, 刀沿参数 17	\$TC_DP17
WearLength5	磨损长度 5, 刀沿参数 18	\$TC_DP18
WearAngle1	磨损角度 1, 刀沿参数 19	\$TC_DP19
WearAngle2	磨损角度 2, 刀沿参数 20	\$TC_DP20
AdaptLength1	夹具长度 1, 刀沿参数 21	\$TC_DP21
AdaptLengthGeoAx1	几何轴 1 夹具长度, 刀沿参数 21, 例如: 夹具长度 X	\$TC_DP21
AdaptLength	夹具长度, 刀沿参数 21	\$TC_DP21
AdaptLength2	夹具长度 2, 刀沿参数 22	\$TC_DP22
AdaptLengthGeoAx3	几何轴 3 夹具长度, 刀沿参数 22, 例如: 夹具长度 Z	\$TC_DP22
AdaptLength3	夹具长度 3, 刀沿参数 23	\$TC_DP23
AdaptLengthGeoAx2	几何轴 2 夹具长度, 刀沿参数 23, 例如: 夹具长度 Y	\$TC_DP23
ReliefAngle	后角, 刀沿参数 24	\$TC_DP24
PlateAngle	刀片角度, 刀沿参数 24 和刀沿参数 10 的组合	\$TC_DP24 与 \$TC_DP10
NoseAngle	刀尖角, 刀沿参数 24	\$TC_DP24
CuttRate	切削速度, 刀沿参数 25	\$TC_DP25
SpindleDirection	在旋转刀具上为刀具主轴的旋转方向。在车刀上为主主轴的旋转方向。	\$TC_DP25 位 8 和位 9

标记	参数和属性	系统变量 ^{*)}
Coolant1	冷却液 1	\$TC_DP25 位 10
Coolant2	冷却液 2	\$TC_DP25 位 11
MFunction1	M 功能 1	\$TC_DP25 位 0
MFunction2	M 功能 2	\$TC_DP25 位 1
MFunction3	M 功能 3	\$TC_DP25 位 2
MFunction4	M 功能 4	\$TC_DP25 位 3
IsoHNoDPH	ISO H 号	\$TC_DPH
OrientNo	刀沿定向	\$TC_DPV
OrientV1	刀沿定向矢量 1	\$TC_DPV3
OrientV2	刀沿定向矢量 2	\$TC_DPV4
OrientV3	刀沿定向矢量 3	\$TC_DPV5
OrientVGeoAx1	几何轴 1 刀沿定向矢量, 刀沿定向参数 3, 例如: 矢量 X	\$TC_DPVN3
OrientVGeoAx3	几何轴 3 刀沿定向矢量, 刀沿定向参数 4, 例如: 矢量 Z	\$TC_DPVN4
OrientVGeoAx2	几何轴 2 刀沿定向矢量, 刀沿定向参数 5, 例如: 矢量 Y	\$TC_DPVN5
TeethCount	刀齿数, 刀沿参数 34	\$TC_DPNT
UniqueDNo	唯一性 D 号	\$TC_DPCE
EdgeUser_1, ..., EdgeUser_10,	OEM 刀沿参数 1 到 10	\$TC_DPC1, ... \$TC_DPC10
EdgeAppl_1, ..., EdgeAppl_10x	西门子应用刀沿参数 1 到 10	\$TC_DPCS1, ... \$TC_DPCS10

^{*)} 参数和系统变量一致, 换句话说, 系统变量构成了参数的基础。

9.2.2.3 监控参数的标识

在配置文件中使用了以下监控参数的标记:

标记	参数和属性	系统变量 ^{*)}
SupWarning	预警极限, 参考当前监控类型。	-
SupRemaining	实际值, 参考当前监控类型	-
SupDesired	设定值, 参考当前监控类型	-

标记	参数和属性	系统变量 ^{*)}
SupWarningTime	刀具寿命预警	\$TC_MOP1
SupRemainingTime	实际刀具寿命	\$TC_MOP2
SupWarningPieces	工件数预警限值	\$TC_MOP3
SupRemainingPieces	实际工件数	\$TC_MOP4
SupDesiredTime	刀具额定寿命	\$TC_MOP11
SupDesiredPieces	工件设定值	\$TC_MOP13
SupWarningWear	磨损预警极限	\$TC_MOP5
SupRemainingWear	磨损实际值	\$TC_MOP6
SupDesiredWear	磨损设定值	\$TC_MOP15
SupExtendedAlarm	PLC 刀具管理： 延长式报警	-
SupWarningPiecesIncremental	PLC 刀具管理： 工件数预警限值	\$TC_MOP13 - \$TC_MOP3
SupActualPieces	PLC 刀具管理： 实际工件数	\$TC_MOP13 - \$TC_MOP4
EdgeSupUser_1, ... EdgeSupUser_10	OEM 刀沿监控参数 1 到 10	\$TC_MOPC1, ... \$TC_MOPC10
EdgeSupAppl_1, ... EdgeSupAppl_10	西门子应用刀沿参数 1 到 10	\$TC_MOPCS1,\$TC_MOPCS10

^{*)} 参数和系统变量一致，换句话说，系统变量构成了参数的基础。

9.2.2.4 磨削参数标志

在配置文件中使用了以下磨削参数标志：

标志/参数	含义	系统变量 ^{*)}
GrindingSpindleNo	主轴号码	\$TC_TPG1
GrindingConnectionRule	级联规则	\$TC_TPG2
MinimalDiscRadius	最小砂轮半径	\$TC_TPG3
ActualDiscRadius	当前砂轮半径（只读）	-
MinimalDiscWidth	最小砂轮宽度	\$TC_TPG4
ActualDiscWidth	当前的砂轮宽度	\$TC_TPG5
MaximalDiscSpeed	最大砂轮转速	\$TC_TPG6

标志/参数	含义	系统变量 ^{*)}
MaximalDiscPeripheralVelocity	最大砂轮圆周速度	\$TC_TPG7
AngleBevelDisc	斜砂轮的角度	\$TC_TPG8
ParamForRadiusCalculation	半径计算用参数	\$TC_TPG9
	修整程序目录	\$TC_TPG_DRSPATH
	修整程序名称	\$TC_TPG_DRSPROG

^{*)} 参数和系统变量一致，换句话说，系统变量构成了参数的基础。

下面的参数用于磨削刀具的修整：

标志/参数	含义
GrindingMonitoring	磨削专用的刀具监控
GrindingUseBaseLength	在计算砂轮半径时考虑基本尺寸

9.2.2.5 刀位参数的标识

在配置文件中使用了以下刀库位置参数的标记：

标记	参数和属性	系统变量 ^{*)}
MagPlaceKind	位置方式	\$TC_MPP1
MagPlaceType	位置类型	\$TC_MPP2
MagPlaceTypeIdent	位置类型为文本	\$TC_MPP2
MagPlaceTNo	该位置上的刀具号	\$TC_MPP6
MagPlaceWatchNeighbour	邻接位置监控	\$TC_MPP3
MagPlaceStateLocked	禁止刀库位置	\$TC_MPP4, 位 0
MagPlaceStateEmpty	空刀库位置	\$TC_MPP4, 位 1
MagPlaceStateResInterMag	为缓冲区中的刀具预留	\$TC_MPP4, 位 2
MagPlaceStateResLoadTool	为待装刀具预留	\$TC_MPP4, 位 3
MagPlaceStateOccupiedLeft	占用左半位	\$TC_MPP4, 位 4

9.2 配置操作界面

标记	参数和属性	系统变量 ^{*)}
MagPlaceStateOccupiedRight	占用右半位	\$TC_MPP4, 位 5
MagPlaceStateOccupiedUpper	占用上半位	\$TC_MPP4, 位 6
MagPlaceStateOccupiedDown	占用下半位	\$TC_MPP4, 位 7
MagPlaceStateReservedLeft	预留左半位	\$TC_MPP4, 位 8
MagPlaceStateReservedRight	预留右半位	\$TC_MPP4, 位 9
MagPlaceStateReservedUpper	预留上半位	\$TC_MPP4, 位 10
MagPlaceStateReservedDown	预留下半位	\$TC_MPP4, 位 11
MagPlaceMagazineNo	刀库号	-
MagPlaceTypeIdx	位置类型索引	\$TC_MPP5
MagPlaceWearGroup	磨损组号	\$TC_MPP5
MagPlaceAdaptNo	夹具号	\$TC_MPP7
MagPlaceNo	刀库位置号	-
MagNoMagPlaceNo	刀库编号/刀库位置号	-
MagPlaceUser_1, ... MagPlaceUser_10	OEM 刀库位置参数 1 到 10	\$TC_MPPC1, ... \$TC_MPPC10
MagPlaceAppl_1, ... MagPlaceAppl_10	西门子刀库位置参数 1 到 10	\$TC_MPPCS1, ... \$TC_MPPCS10

^{*)}参数和系统变量一致，换句话说，系统变量构成了参数的基础。

9.2.2.6 刀库位置适配器参数的标记

在配置文件中使用了以下刀库位置适配器参数的标记：

标记	参数和属性	系统变量 ^{*)}
MagPlaceAdapterLength1	刀库位置适配器长度 1	\$TC_ADPT1
MagPlaceAdapterLengthGeoAx1	刀库位置适配器长度，几何轴 1	\$TC_ADPT1

标记	参数和属性	系统变量 ^{*)}
MagPlaceAdapterLength2	刀库位置适配器长度 2	\$TC_ADPT2
MagPlaceAdapterLengthGeo Ax3	刀库位置适配器长度, 几何轴 3	\$TC_ADPT2
MagPlaceAdapterLength3	刀库位置适配器长度 3	\$TC_ADPT3
MagPlaceAdapterLengthGeo Ax2	刀库位置适配器长度, 几何轴 2	\$TC_ADPT3
MagPlaceAdapterTrafoNo	刀库位置适配器转换号	\$TC_ADPT4

^{*)} 参数和系统变量一致, 换句话说, 系统变量构成了参数的基础。

9.2.2.7 多刀参数的标识

在配置文件中使用了以下多刀参数的标记:

标记	参数和属性	系统变量 ^{*)}
MultiToolNo	多刀编号	-
MultiToolNumberOfPlaces	多刀刀位数量	\$TC_MTPN
MultiToolNumberOfPlacesRO	多刀刀位数量, 只读	\$TC_MTPN
MultiToolIdent	多刀名称	\$TC_MTP2
MultiToolInMag	多刀位于的刀库	\$A_TOOLMN
MultiToolInPlace	多刀位于的刀库位置	\$A_TOOLMLN
MultiToolInMagInPlace	刀库编号/位置	\$A_TOOLMN / \$A_TOOLMLN
MultiToolMyMag	多刀所属的刀库	\$A_MYMN
MultiToolMyPlace	多刀所属的刀库位置	\$A_MYMLN
MultiToolPlaceSpec	位置类型	\$TC_MTP7
MultiToolPlaceSpecIdent	位置类型为文本	\$TC_MTP7
MultiToolSizeLeft	多刀占据的左半位数目	\$TC_MTP3
MultiToolSizeRight	多刀占据的右半位数目	\$TC_MTP4
MultiToolSizeUpper	多刀占据的上半位数目	\$TC_MTP5
MultiToolSizeDown	多刀占据的下半位数目	\$TC_MTP6
MultiToolOverSize	刀具尺寸为固定设置 - 两个左半位、 两个右半位、一个上半位、一个下半 位	\$TC_MTP3 - \$TC_MTP6

标记	参数和属性	系统变量 ^{*)}
MultiToolPosition	Multitoolposition	\$TC_MTP_POS
MultiToolProtAreaFile	确定保护区的多刀描述文件的名称	\$TC_MTP_PRO TA
MultiToolKindOfDist	多刀刀位间距的输入方式	\$TC_MTP_KD
MultiToolKindOfDistRO	多刀刀位间距的输入方式, 只读	\$TC_MTP_KD
MultiToolDistAngle	多刀刀位间距为角度值	\$TC_MTP_KD
MultiToolState	以十六进制方式表示的多刀状态	\$TC_MTP8
MultiToolStateEnabled	多刀状态 - 多刀已使能	\$TC_MTP8, 位 1
MultiToolStateLocked	多刀状态 - 多刀已禁用	\$TC_MTP8, 位 2
MultiToolStateLockedRO	多刀状态 - 多刀禁用,只读	\$TC_MTP8, 位 2
MultiToolStateInChange	多刀状态 - 多刀换刀中	\$TC_MTP8, 位 5
MultiToolStateFixed	多刀状态 - 多刀采用固定刀位编码	\$TC_MTP8, 位 6
MultiToolStateUsed	多刀状态 - 多刀已使用	\$TC_MTP8, 位 7
MultiToolStateAutoReturn	多刀状态 - 自动返回	\$TC_MTP8, 位 8
MultiToolStateIgnoreLocked	多刀状态 - 忽视禁用	\$TC_MTP8, 位 9
MultiToolStateMarkedToUnload	多刀状态 - 已选中多刀待卸载	\$TC_MTP8, 位 10
MultiToolStateMarkedToLoad	多刀状态 - 已选中多刀待装载	\$TC_MTP8, 位 11
MultiToolStatePermanent	多刀状态 - 多刀为主刀具	\$TC_MTP8, 位 12
MultiToolState1To1Exchange	多刀状态 - 一对一交换	\$TC_MTP8, 位 14
MultiToolStateHandTool	多刀状态 - 手动刀具	\$TC_MTP8, 位 15
MultiToolStateLockMtlfToolLock	多刀状态 - 若多刀中的一把刀具被禁用, 则禁用多刀	\$TC_MTP8, 位 16
MultiToolUser_1, ... MultiToolUser_10	OEM 多刀参数 1 到 10	\$TC_MTPC1, ... \$TC_MTPC10
MultiToolAppl_1, ... MultiToolAppl_10	西门子应用多刀参数 1 到 10	\$TC_MTPCS1, .. \$TC_MTPCS10

^{*)}参数和系统变量一致, 换句话说, 系统变量构成了参数的基础。

9.2.2.8 多刀刀位参数的标识

在配置文件中使用了以下多刀刀位参数的标记：

标记	参数和属性	系统变量 ^{*)}
MultiToolDist	多刀刀位间距，考虑当前的多刀间距输入类型	-
MultiToolDistLength	多刀刀位间距，长度	\$TC_MTPPL
MultiToolDistAngle	多刀刀位间距，角度	\$TC_MTPPA
MultiToolPlaceType	位置类型	\$TC_MTPP2
MultiToolPlaceTypeIdent	位置类型为文本	\$TC_MTPP2
MultiToolPlaceState	以十六进制方式表示的多刀状态	\$TC_MTPP4
MultiToolPlaceStateLocked	多刀刀位被禁用	\$TC_MTPP4, 位 0
MultiToolPlaceStateEmpty	多刀刀位为空	\$TC_MTPP4, 位 1
MultiToolPlaceTNo	此多刀刀位上的刀具号	\$TC_MTPP6
MultiToolPlaceAdaptNo	夹具号	\$TC_MTPP7
MultiToolPlaceNo	多刀刀位号	-
MultiToolPlaceUser_1, ... MultiToolPlaceUser_10	OEM 多刀刀位参数 1 到 10	\$TC_MTPPC1, ... \$TC_MTPPC10
MultiToolPlaceAppl_1, ... MultiToolPlaceAppl_10	西门子应用多刀刀位参数 1 到 10	\$TC_MTPPCS1, ... \$TC_MTPPCS10

^{*)}参数和系统变量一致，换句话说，系统变量构成了参数的基础。

9.2.3 配置列表参数

标签<PARAMETERCONFIGURATION>

在标签<PARAMETERCONFIGURATION>中可以进行：

1. 更改列表参数。
2. 在已有列表参数的基础上创建新的列表参数。

修改列表参数

可以修改系统中包含的所有参数（直至 ISO 参数）。

刀具参数标识 (页 264)

刀沿参数的标识 (页 267)

监控参数的标识 (页 269)

磨削参数标志 (页 270)

刀位参数的标识 (页 271)

刀库位置适配器参数的标记 (页 272)

多刀参数的标识 (页 273)

多刀刀位参数的标识 (页 275)

条目	含义
HeadLine	列标题。所输入的字段将作为标题显示。 示例： 字段“半径” 将作为标题显示（默认）。
ToolTip	工具栏中显示的文本。 示例： 字段“几何半径” 显示在工具栏中（默认）。
ShortText	在窗口“其它数据”中显示参数时出现的文本。 示例： 在窗口“其他数据”中显示字段“半径”（默认）。
DetailsText	在窗口“详细信息”中显示参数时出现的文本。 示例： 在窗口“详细信息”中显示文本“半径”（默认）。
Width	基于 640x480 分辨率的列宽度，以像素计。 参见下例： 默认列宽度更改为 53 像素。
DisplayMode	参数以该值进行显示。 参见下例： 缺省设置改为“DoubleMode”。 也接受以下各值：

条目	含义	
	AnyMode	全部字符
	IntegerMode	整数值
	UnsignedIntegerMode	无符号的整数值
	DoubleMode	包含小数点的数值。 在“DecimalPlaces”中确定小数位的位数。
	UnsignedDoubleMode	不带正负号、包含小数点的数值 在“DecimalPlaces”中确定小数位的位数
	Length	长度
	Angle	角度
	LinearFeedPerTime	直线进给 毫米/分
	LinearFeedPerRevolution	直线进给 毫米/转
	LinearFeedPerTooth	直线进给 毫米/齿
	RevolutionSpeed	转速
	ConstantCuttingSpeed	恒定切削速度
DecimalPlaces	当在 DisplayMode 中选择“DoubleMode”或“UnsignedDoubleMode”时，小数位的位数。 参见示例：输入了 2 个小数位。	
ItemType	数值显示区的类型	
	TextField	数值和文字的输入/输出栏
	TextFieldReadOnly	数值和文字的输出栏
	CheckBox	状态的输入/输出栏
	CheckBoxReadOnly	状态的输出栏
BitMask	从数值中显示数位的位掩标记。位标记作为整数值输入。位 0 -> 1、位 1 -> 2、位 2 -> 4	
AccessLevel	从哪个访问级别起可在操作界面中使用该参数，值范围 1 至 7	
UpperLimit	参数在操作界面中的输入上限值， 只针对数值输入型参数。	
LowerLimit	参数在操作界面中的输入下限值， 只针对数值输入型参数。	

示例

下面的示例中会使用参数<GeoRadius>。

只给定已修改的数据。

```
<PARAMETERCONFIGURATION>
  <GeoRadius>
    <Width value="53" type="int" />
    <DisplayMode value="DoubleMode" type="QString" />
    <DecimalPlaces value="2" type="int" />
  </GeoRadius>
</PARAMETERCONFIGURATION>
```

在已有列表参数的基础上配置新的列表参数

请给定新的参数名称并只写入修改的数据：

条目	含义
	分配新参数名称 参见示例 1： 文本“NewGeoRadius”
Base	新参数参考的参数名称 参见示例 1： 参数“GeoRadius”用作模板。
	此处也只输入已修改的数据。所有其他数据被现有参数采纳 参见示例 1： 修改过的列宽度改为 46 像素。小数位的位数改为 1。

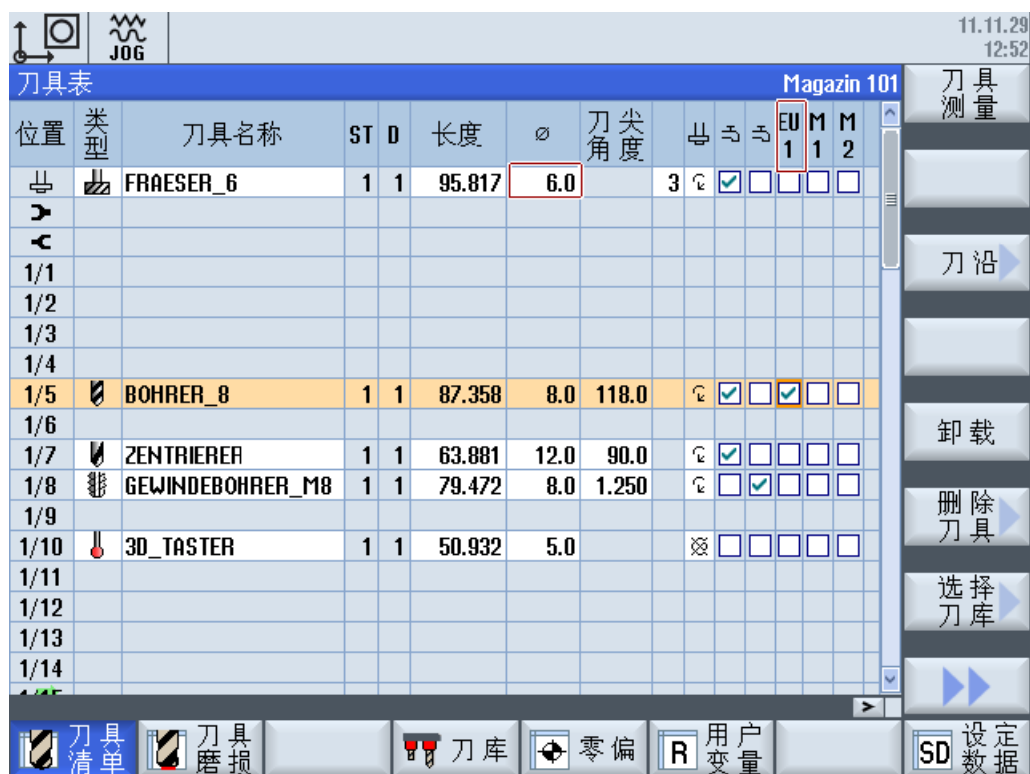
示例 1

```
<PARAMETERCONFIGURATION>
  <NewGeoRadius>
    <Base value="GeoRadius" type="QString" />
    <Width value="46" type="int" />
    <DisplayMode value="DoubleMode" type="QString" />
    <DecimalPlaces value="1" type="int" />
  </NewGeoRadius>
</PARAMETERCONFIGURATION>
```

示例 2

条目	含义
	命名新的刀沿参数 在下面的示例中参数名称为“EdgeUser_1_Bit0”。
Base	新参数参考的参数名称 下例中将参数“EdgeUser_1”用作模板。
	此处也只输入已修改的数据。所有其他数据从已有刀沿参数中接收。
	修改下列条目： 数组类型：状态的输入/输出栏 数值输出：全部字符 位标记：位 0 列宽度改为 17 像素。 标题改为“TM_HL_EDGE_USER_1_Bit0” 工具栏中的文本改为“TM_TT_EDGE_USER_1_Bit0”

```
<PARAMETERCONFIGURATION>
  <EdgeUser_1_Bit0>
    <Base value="EdgeUser_1" type="QString" />
    <ItemType value="CheckBox" type="QString" />
    <DisplayMode value="AnyMode" type="QString" />
    <BitMask value="1" type="int" />
    <Width value="17" type="int" />
    <HeadLine value="TM_HL_EDGE_USER_1_Bit0" type="QString" />
    <ToolTip value="TM_TT_EDGE_USER_1_Bit0" type="QString" />
  </EdgeUser_1_Bit0>
</PARAMETERCONFIGURATION>
```



9.2.4 刀具类型列表

铣刀类型代码

类别组 1xy（铣刀）：

100	符合 CLDATA (DIN 66215)的铣刀
110	圆柱形球头模具铣刀
111	圆锥形球头模具铣刀
120	无倒角立铣刀
121	带倒角立铣刀
130	无倒角角头铣刀
131	带倒角角头铣刀
140	面铣刀
145	螺纹铣刀
150	盘形铣刀

151	锯
155	无倒角截锥铣刀
156	带倒角截锥铣刀
157	圆锥形模具铣刀
160	螺纹钻铣刀

钻头类型代码

类别组 2xy (钻头) :

200	麻花钻
205	整体钻头
210	镗杆
220	中心钻
230	尖头铰钻
231	平头铰钻
240	螺纹丝锥
241	细螺纹丝锥
242	惠氏螺纹丝锥
250	铰刀

磨具类型代码

类别组 4xy (磨具) :

400	外圆磨削砂轮
410	平面磨削砂轮
490	修整器
494	修整轮
495	带辅助轴的修整轮
496	修整轮
497	带回转轴的修整轮

9.2 配置操作界面

磨具类型 401, 402, 403 以及 411, 412, 413 不是刀具类型的选项。可以通过单独的参数设置“磨削专用的刀具监控”和“计算砂轮半径时考虑基本尺寸”，这两个参数作用于常规形状的砂轮类型。

车刀类型代码

类别组 5xy（车刀）：

500	粗车刀
510	精车刀
520	车槽刀
530	切断刀
540	螺纹车刀
550	成形车刀
560	旋转钻头（ECOCUT）
580	定向测头
585	标准刀具

特种刀具类型代码

组类型 7xy（特种刀具）：

700	槽锯
710	3D 测头
711	寻边测头
712	单向测头
713	L 形测头
714	星形测头
725	标准刀具
730	定位挡块
731	顶尖套筒
732	中心架
900	辅助刀具

9.2.5 配置刀具类型

标签<TOOLTYPECONFIGURATION>

在标签<TOOLTYPECONFIGURATION>中提供了配置刀具类型的条目：

条目	含义
TOOLTYPE_XXX	XXX 表示刀具类型的编号。 刀具类型的分类和编号参见章节“刀具类型列表 (页 280)”。
Tooltype	刀具类型号(XXX)
Name	刀具名称的文本标记。 刀具名称在以下窗口中显示： <ul style="list-style-type: none"> “新建刀具 - 收藏” “新建刀具 - 铣刀 100-199” “新建刀具 - 钻头 200-299” “新建刀具 - 磨削刀具 400-499” “新建刀具-特种刀具和辅助刀具 700-900” 参见示例： 对于文本标记“TM_PAR_SHANK_END_CUTTER”在操作界面上显示名称“立铣刀”。
Shortname	刀具名称的文本标记。 刀具名称在刀具列表中显示。 参见示例： 如果指定文本标记“TM_PPTT_SHANK_END_CUTTER”，则在操作界面上显示名称“铣刀”。
Tooltip	刀具名称的文本标记。 刀具名称显示在气泡悬浮框中。 参见示例： 如果指定文本标记“TM_TTTT_SHANK_END_CUTTER”，则在操作界面上显示名称“立铣刀”。
Icon9 - 适用于铣削工艺	表示刀具类型的符号。 符号显示在以下窗口中： <ul style="list-style-type: none"> “刀具列表”中的“类型”列 “新建刀具 - 收藏”中的“刀具位置”列 “新建刀具 - 铣刀 100-199”中的“刀具位置”列 “新建刀具 - 钻头 200-299”中的“刀具位置”列 “新建刀具 - 特种刀具 700-900”中的“刀具位置”列 符号的文件格式为“.png”。 按屏幕分辨率保存文件到以下目录中： /oem/sinumerik/hmi/ico/ico640, 或 ico800, 或 ico1024 /user/sinumerik/hmi/ico/ico640, 或 ico800, 或 ico1024

条目	含义
IconX - 适用于车削和磨削工艺	表示刀具类型的符号（参见 Icon9 - 适用于铣削工艺） 特殊之处：在车削和磨削工艺中支持刀具位置。 X 表示用图符显示的刀具位置。位置 9 是未经定义的位置，用打叉符号表示。
Iconorder	由符号显示的刀具位置的顺序。

示例

```

<TOOLTYPECONFIGURATION>
  <TOOLTYPE_120>
    <Tooltype value="120" type="uint" />
    <Name value="TM_PAR_SHANK_END_CUTTER" type="QString" />
    <Shortname value="TM_PPTT_SHANK_END_CUTTER" type="QString" />
    <Tooltip value="TM_TTTT_SHANK_END_CUTTER" type="QString" />
    <Icon9 value="to_poly_shank_end_cutter_down.png" type="QString" />
  </TOOLTYPE_120>
</TOOLTYPECONFIGURATION>
    
```



9.2.6 配置“其他数据”窗口

标签<MOREDATACONFIGURATION>

在标签<MOREDATACONFIGURATION>中提供了配置窗口“其它数据”的条目。对于每个刀具类型，可以在多个行和列中显示各个其他数据。每个待显示的数据由列表参数的输入定义：

刀具参数标识 (页 264)

刀沿参数的标识 (页 267)

监控参数的标识 (页 269)

在窗口中显示参数的简要说明(ShortText)并在旁边显示其参数值。配置列表参数 (页 275).

如需在窗口“其它数据”中显示其他描述文本，可将该文本在行/列中输入。该文本在窗口中接受相同的宽度，与简称和参数值相同。

可以设置任意多行和列。窗口超出一定大小时，显示滚动条。

条目	含义
TOOLTYPE_XXX	刀具类型编号。 参见示例： 刀具类型 111= 球面铣刀（圆锥形锻模铣刀）。
ROWX_COLY	各行 X_列 Y。 如果行 X 和列 Y 后面没有输入内容，将该栏空置。 参见示例：“其它数据”窗口中的第一行为空。
Item	允许下列条目： <ul style="list-style-type: none"> ● 参数标记 ● 文本 参见示例： 窗口的第二行中出现标题“拐角半径”。 窗口的第三行上出现“半径”字样以及数值输入栏。 第四行为空。

示例

```
<MOREDATACONFIGURATION>
  <TOOLTYPE_111>
    <ROW1_COL1>
  </ROW1_COL1>
```

9.2 配置操作界面

```

<ROW2_COL1>
    <Item value="TM_DGL_CORNER_RADIUS" type="QString" />
</ROW2_COL1>
<ROW3_COL1>
    <Item value="GeoCornerRadius" type="QString" />
</ROW3_COL1>
<ROW4_COL2>
</ROW4_COL2>
</TOOLTYPE_111>
</MOREDATACONFIGURATION>
    
```

9.2.7 配置窗口“新刀具 - 收藏”

标签<NEWTOLFAVORITECONFIGURATION>

在标签<NEWTOLFAVORITECONFIGURATION> 中定义最经常使用的刀具类型。所确定的刀具类型在窗口“新建刀具 - 收藏”中显示。

条目	含义
StaticTooltypes	刀具类型编号。输入各个编号，相互之间用空格符隔开。 刀具类型的分类和编号参见章节：刀具类型列表 (页 280)

示例

```

<NEWTOLFAVORITECONFIGURATION>
    <StaticTooltypes value="120 140 200 220 710 711" type="QString"/>
</NEWTOLFAVORITECONFIGURATION>
    
```

9.2.8 配置窗口“新建刀具”

标签<NEWTOLCONFIGURATION>

利用对话框“新建刀具”，您可以在创建刀具或者将刀具装入某个刀库位置之前便在该对话框中显示某些刀具数据。这些数据通常是刀具尺寸或者刀具位置类型，因为一旦装入刀具，这些参数便无法再被修改。我们建议除了这些参数外，最好还在窗口中显示刀具类型和刀具名称。

在标签<NEWTOOLCONFIGURATION> 中确定需要在对话框“新建刀具”中显示的参数。

说明

参数的数量由窗口的大小决定。

条目	含义
ItemX	刀具参数标记。X 指定参数在窗口中显示的顺序。 刀具参数的标记参见章节： 刀具参数标识 (页 264)

示例

```
<NEWTOOLCONFIGURATION>
  <!-- 缺省栏“刀具过大”被清除 -->
  <!--由“刀具刀位类型”，“刀具左侧大小”和“刀具右侧大小”-->
  <!-- 代替 -->
  <Item2 value="ToolPlaceSpec" type="QString"/>
  <Item3 value="ToolSizeLeft" type="QString"/>
  <Item4 value="ToolSizeRight" type="QString"/>
</NEWTOOLCONFIGURATION>
```

9.2.9 配置新刀具的默认值

标签<NEWTOOLDEFAULTVALUECONFIGURATION>

在标签<NEWTOOLDEFAULTVALUECONFIGURATION>中确定在屏幕上创建的刀具的默认值。默认值的配置取决于刀具类型，某些情况下也取决于刀沿位置。

条目	含义
TOOLTYPE_ALL	参数设置作用于所有刀具类型。
TOOLTYPE_XXX	参数设置只作用于特定刀具类型。XXX 代表刀具类型号。
CUTTEDGEPOS_XX X	参数设置只作用于特定刀沿位置。XXX 代表刀沿位置号。
Default	参数的默认值。

示例

```

<NEWTTOOLDEFAULTVALUECONFIGURATION>
  <!-- 所有刀具类型的几何长度 1 的初始值为 70.0 -->
  <TOOLTYPE_ALL>
    <GeoLengthGeoAx1>
      <Default value="70.0" type="double" />
    </GeoLengthGeoAx1>
  </TOOLTYPE_ALL>
  <!-- 麻花钻的尖角的初始值为 120.0 -->
  <TOOLTYPE_200>
    <NoseAngle>
      <Default value="120.0" type="double" />
    </NoseAngle>
  </TOOLTYPE_200>
  <TOOLTYPE_500>
    <!-- 粗加工刀具的刀片角的初始值为 82.0 -->
    <PlateAngle>
      <Default value="82.0" type="double" />
    </PlateAngle>
    <!-- 粗加工刀具的夹角的初始值根据刀沿位置 -->
    <HolderAngle>
      <CUTTEDGEPOS_1>
        <Default value="93.0" type="double" />
      </CUTTEDGEPOS_1>
      <CUTTEDGEPOS_2>
        <Default value="93.0" type="double" />
      </CUTTEDGEPOS_2>
      <CUTTEDGEPOS_3>
        <Default value="93.0" type="double" />
      </CUTTEDGEPOS_3>
      <CUTTEDGEPOS_4>
        <Default value="93.0" type="double" />
      </CUTTEDGEPOS_4>
      <CUTTEDGEPOS_5>
        <Default value="49.0" type="double" />
      </CUTTEDGEPOS_5>
      <CUTTEDGEPOS_6>
        <Default value="49.0" type="double" />
      </CUTTEDGEPOS_6>
      <CUTTEDGEPOS_7>

```



```

        <Default value="49.0" type="double" />
    </CUTTEDGEPOS_7>
    <CUTTEDGEPOS_8>
        <Default value="49.0" type="double" />
    </CUTTEDGEPOS_8>
    </HolderAngle>
</TOOLTYPE_500>
<!-- 切入刀具的冷却液 2 的初始状态为 “接通”-->
<TOOLTYPE_520>
    <Coolant2>
        <Default value="true" type="bool" />
    </Coolant2>
</TOOLTYPE_520>
</NEWTOLDEFAULTVALUECONFIGURATION>

```

创建刀具时的默认值

系统变量		缺省值	
刀具尺寸			
左侧	\$TC_TP3	1	
右侧	\$TC_TP4	1	
上部	\$TC_TP5	1	
下部	\$TC_TP6	1	
刀具的刀库位置类型			
在刀库外面创建	\$TC_TP7	1	
在某个刀库位置上创建	\$TC_TP7		刀库位置的类型
刀具状态			
已使能	\$TC_TP8	2	

9.2 配置操作界面

不同的刀具类型产生以下各种刀沿位置和切削方向的组合。在创建刀具时操作人员从窗口“新建刀具 - 收藏”内选择一种组合。

刀具类型 \$TC_DP1	缺省值	
	刀沿位置 \$TC_DP2	切削方向 \$TC_DP11
500 粗加工刀具 510 精加工刀具 560 旋转钻头	1	4
	2	3
	3	3
	4	4
	5	1
	6	3
	7	1
	8	3
520 切入刀具 530 切断刀 磨削刀具类型	1	2
	1	4
	4	4
	3	1
	4	1
	2	3
	3	3
	2	2
540 螺纹车刀	5	2
	5	1
	8	3
	8	4
	6	3
	6	4
	7	2
	7	1

设置了选件 ShopMill/ShopTurn 时，主轴旋转方向的默认值为：

刀具类型 \$TC_DP1	默认值 TC_DP25	主轴旋转方向
测量刀具		
580/585/710/712/713/725	位 8 = 0 位 9 = 0	主轴停止
车刀类型取决于 SD54215 TM_FUNCTION_MASK_SET 位 1 的设置		
SD54215 位 1 = 0	位 8 = 1 位 9 = 0	主轴正转
SD54215 位 1 = 1	位 8 = 0 位 9 = 1	主轴反转
特种刀具		
730/731/732	位 8 = 0 位 9 = 0	主轴停止
所有其它刀具类型		
-	位 8 = 1 位 9 = 0	主轴正转

刀具类型 \$TC_DP1	默认值 \$TC_DP25，表示切换方向
712 单向测头	位 17 = 1 位 20 = 1
713 L 形测头	位 17 = 1 位 21 = 1
714 星形测头	位 16 = 1 位 17 = 1 位 18 = 1 位 19 = 1

刀具类型 \$TC_DP1	默认值 \$TC_DP24，表示尖角
200 麻花钻	118
220 中心钻	90

9.2 配置操作界面

在操作界面上不显示后角，而是显示刀片角度。

刀片角度 = 180 - 夹角 - 后角

刀具类型 \$TC_DP1	缺省值		
	刀沿位置 \$TC_DP2	夹角 \$TC_DP10	后角 \$TC_DP24
500 粗加工刀具	1 至 4	95	5
	5 至 8	50	50
510 精加工刀具	1 至 4	93	32
	5 至 8	62.5	62.5
560 旋转钻头	-	90	2

刀具类型 \$TC_DP1	尺寸单位	默认值 \$TC_DP8, 表示刀片长度
500 粗加工刀具	公制	11
510 精加工刀具	英制	0.4

对于所有磨削刀具而言，\$TC_TPG9 = 3（等于长度 1），以计算半径。

9.2.10 配置“详细信息”窗口

标签<DETAILSCONFIGURATION>

在标签<DETAILSCONFIGURATION>中可以设置“详细信息”窗口：

条目	含义
ShowToolNumber	刀具号显示在“详细信息”窗口的右上方。
UseAxisNameInLength	指定刀具长度用轴标识符来显示，例如用“X 轴长度”，而不是用“长度 1”来显示。 没有指定时，采用刀具列表中的设置。

条目	含义
ShowBaseLength	指定基本长度显示在“详细信息”窗口(AdaptLength1, AdaptLength2 和 AdaptLength3)中。 没有指定时，采用刀具列表中的设置。
ShowYAxis	指定 Y 轴显示在“详细信息”窗口中。 没有指定时，采用刀具列表中的设置。

示例

```
<DETAILSCONFIGURATION>
  <ShowToolNumber value="true" type="bool" />
</DETAILSCONFIGURATION>
```

9.2.11 TO 单元和刀套至通道的分配

TO 单元和刀套至通道的分配

通过机床数据 MD28085 进行 TO 单元至通道的分配。通过机床数据 MD20124 进行刀套至通道的分配（仅针对 TE822）。

为每个通道分配一个 TO 单元。

- 通道 1

MD28085 \$MC_MM_LINK_TOA_UNIT	TO 单元至通道的分配 (SRAM)
= 1	为通道分配 TO 单元 1。

- 通道 2

MD28085 \$MC_MM_LINK_TOA_UNIT	TO 单元至通道的分配 (SRAM)
= 2	为通道分配 TO 单元 2。

9.2 配置操作界面

为每个通道分配一个刀套。

● 通道 1

MD20124 \$MC_TOOL_MANAGEMENT_TOOLHOLDER	刀套号
= 1	为通道分配刀套 1。

● 通道 2

MD20124 \$MC_TOOL_MANAGEMENT_TOOLHOLDER	刀套号
= 1	为通道分配刀套 1。

两个通道中都有一个带 12 个刀位的转塔刀库且各有一个装载位和一个刀套。通过分配给一个单独的 TO 单元可以明确地分配刀具管理任务。

通道 1, TOA 1	通道 2, TOA 2
刀库 1 12 个刀位: 1/1 至 1/12	刀库 1 12 个刀位: 1/1 至 1/12
缓冲区刀库 9998 1 个刀位 9998/1	缓冲区刀库 9998 1 个刀位 9998/1
装载刀库 9999 1 个刀位 9999/1	装载刀库 9999 1 个刀位 9999/1

可选配置，带用于通道 1 和通道 2 的 TOA 1

两个通道也可共用一个 TO 单元。

带两个刀库的示例：

- 为每个刀库分配一个刀套。
- 在通道 1 中，刀套 1 被确定为主刀套。
- 在通道 2 中，刀套 2 被确定为主刀套。
- 可在无地址扩展的情况下在 NC 程序中进行编程。
- 两个刀库通过同一个装载位装载。
- 两个通道中的 HMI 偏移区域中的刀具表和刀库表相同。
- 两个刀库可从两个通道中装载和定位。

- 为两个通道分配 TO 单元 1。

通道 1:

MD28085 \$MC_MM_LINK_TOA_UNIT	TO 单元至通道的分配 (SRAM)
= 1	为通道分配 TO 单元 1。

通道 2:

MD28085 \$MC_MM_LINK_TOA_UNIT	TO 单元至通道的分配 (SRAM)
= 2	为通道分配 TO 单元 1。

- 为每个通道分配一个刀套。

通道 1:

MD20124 \$MC_TOOL_MANAGEMENT_TOOLHOLDER	刀套号
= 1	为通道分配刀套 1。

通道 2:

MD20124 \$MC_TOOL_MANAGEMENT_TOOLHOLDER	刀套号
= 2	为通道分配刀套 2。

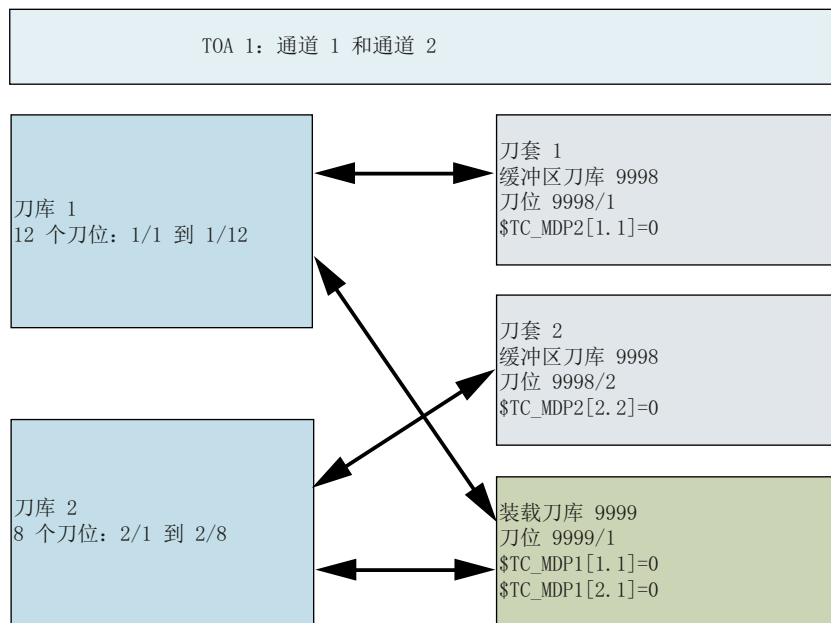


图 9-1 刀库配置

通过分配给刀库定义缓冲区：

```

$TC_MAP1[9998]=7
$TC_MAP2[9998]="缓冲器 1"
$TC_MAP3[9998]=17
$TC_MAP6[9998]=1
$TC_MAP7[9998]=2
$TC_MAP8[9998]=0

$TC_MPP1[9998.1]=2
$TC_MPP2[9998.1]=0
$TC_MPP3[9998.1]=0
$TC_MPP4[9998.1]=2
$TC_MPP5[9998.2]=1

$TC_MPP1[9998.2]=2
$TC_MPP2[9998.2]=0
$TC_MPP3[9998.2]=0
$TC_MPP4[9998.2]=2
$TC_MPP5[9998.2]=2

$TC_MAP1[9999]=9
$TC_MAP2[9999]="装载刀库 1"
$TC_MAP3[9999]=17
$TC_MAP7[9999]=1
$TC_MAP8[9999]=0

$TC_MPP1[9999.1]=7
$TC_MPP2[9999.1]=0
$TC_MPP3[9999.1]=0
$TC_MPP4[9999.1]=2

$TC_MDP1[1.1]=0
$TC_MDP1[2.1]=0

$TC_MDP2[1.1]=0
$TC_MDP2[2.2]=0
    
```

9.2.12 命名刀库和刀库位置

标签<MAGAZINEPLACENAMECONFIGURATION>

在标签 <MAGAZINEPLACENAMECONFIGURATION> 中提供了命名刀库的条目。可通过文本显示刀库位置。随后会在刀具列表中输出定义的文本，比如输出“第一个刀库，位置五”，而不是“1/5”。

条目	含义
MAGAZINE_XXX	XXX 表示刀库的编号。
PLACE_XXX	XXX 表示刀位的编号。
Name	刀库位置的文本标记。

示例

```

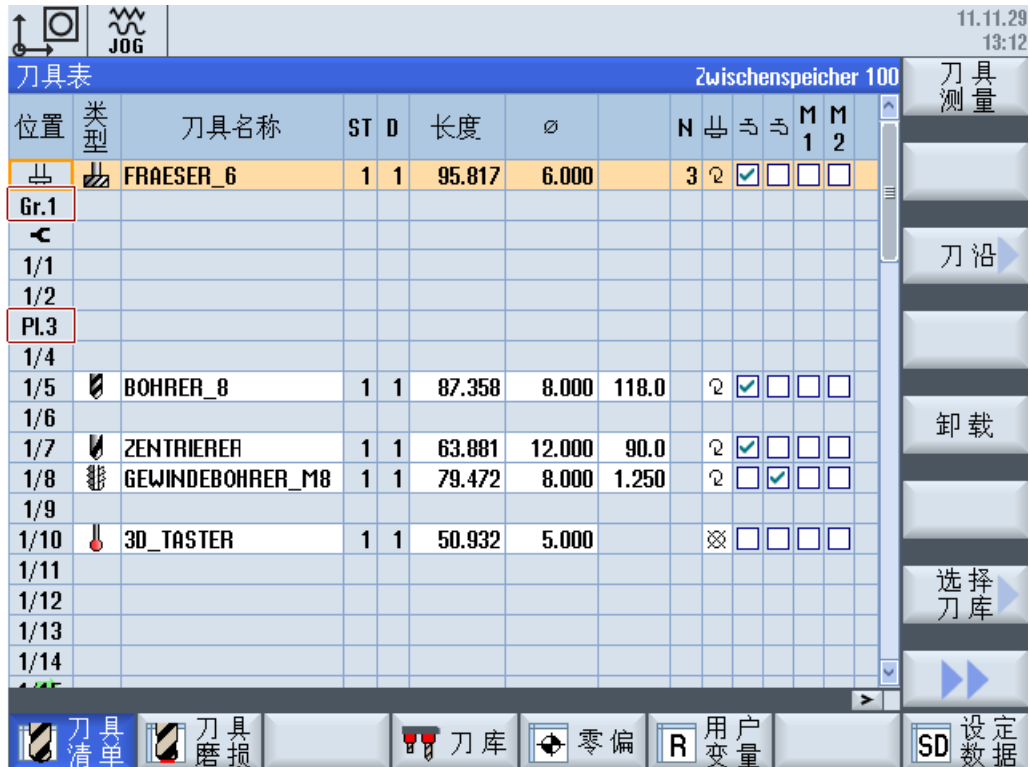
<MAGAZINEPLACENAMECONFIGURATION>
  <MAGAZINE_9998>
    <PLACE_2>
      <Name value="TM_SPECIAL_MAG_PLACE_1" type="QString" />
    </PLACE_2>
  </MAGAZINE_9998>
</MAGAZINEPLACENAMECONFIGURATION>
    
```



```

</MAGAZINE_9998>
<MAGAZINE_1>
  <PLACE_3>
    <Name value="TM_SPECIAL_MAG_PLACE_2" type="QString" />
  </PLACE_3>
</MAGAZINE_1>
</MAGAZINEPLACENAMECONFIGURATION>

```



刀库名称

刀库也可以用文本命名。该文本名称或者文本标记在到 NC 刀库配置文件（ini 文件）中输入。用于刀库命名的系统变量是“\$TC_MAP2[magNo]”。刀库名称会显示到刀具表的右上方。

示例

刀库 1 应命名为“主刀库”。

```
$TC_MAP2[1]="主刀库"
```

如果希望以某种语言显示刀库名“主刀库”，则必须创建“主刀库”对应的该语言的文本。

另见

如何创建和语言相关的文本请参见以下章节：

创建 OEM 文本 (页 312)

9.2.13 设定刀库刀位类型的名称

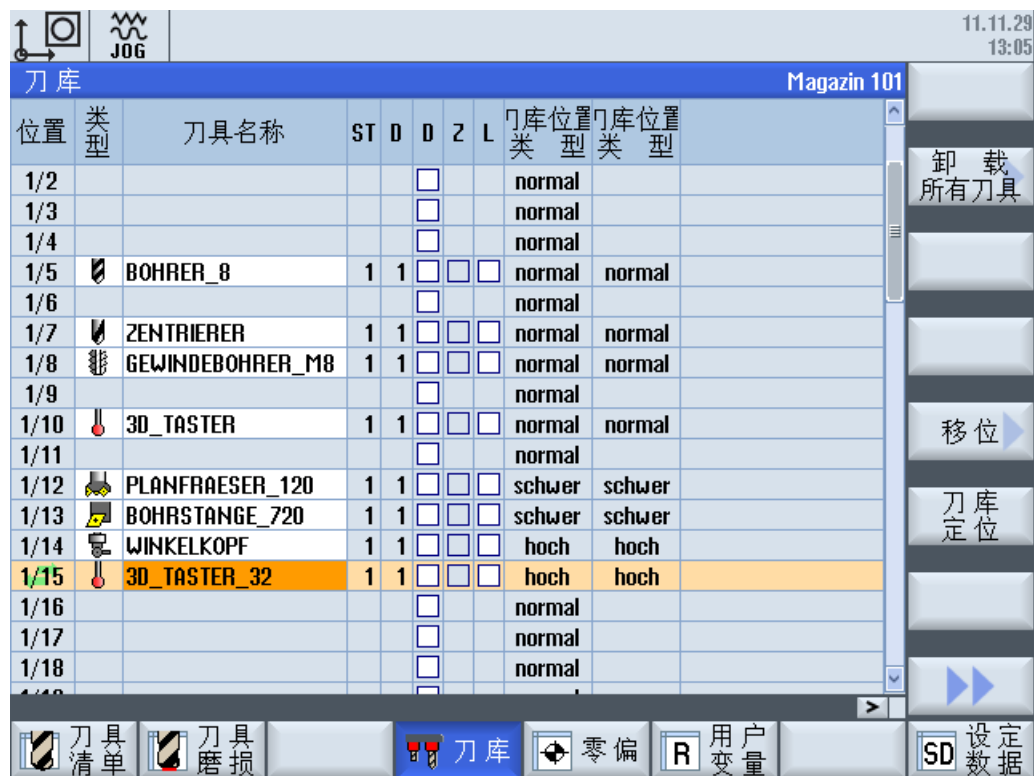
标签<PLACETYPECONFIGURATION>

在标签<PLACETYPECONFIGURATION>中提供了指定刀位类型符的条目。刀库和刀位类型可以以文本格式显示在刀具列表中，这样在刀具列表中便不会显示 ID（例如 2 代表刀位类型 2），而是输出配置的文本。

条目	含义
PLACETYPEXXX	XXX 表示刀位类型的编号
文本	刀位类型的文本标记
Tooltip	刀位类型的气泡悬浮框文本标记

示例

```
<PLACETYPECONFIGURATION>
  <PLACETYPE1>
    <Text value="TM_PLACETYPE_1" type="QString" />
    <Tooltip value="TM_TT_PLACETYPE_1" type="QString" />
  </PLACETYPE1>
  <PLACETYPE2>
    <Text value="TM_PLACETYPE_2" type="QString" />
    <Tooltip value="TM_TT_PLACETYPE_2" type="QString" />
  </PLACETYPE2>
  <PLACETYPE3>
    <Text value="TM_PLACETYPE_3" type="QString" />
    <Tooltip value="TM_TT_PLACETYPE_3" type="QString" />
  </PLACETYPE3>
</PLACETYPECONFIGURATION>
```



另见

如何创建和语言相关的文本请参见以下章节：

创建 OEM 文本 (页 312)

9.2.14 确定刀库和通道的对应关系

前提条件

已设置了车削工艺： 对应配置文件的名称为 `sltmturninglistconfig.xml`。

指定刀库和通道之间的关系

通过该设置可以指定刀库和通道之间的对应关系。仅在对机床进行了相应配置，为一个刀具区域（TOA 区域）指定了多个通道时，才推荐进行该设置。如果每个刀库只占用一个通道，则可以通过该设置固定两者的分配关系。


```
</CHANNELNO_4>
</CHANNELMAGAZINEASSIGNMENT>
```

9.2.15 冷却剂和刀具专用功能

指定冷却液功能 (ShopMill / ShopTurn)

在 ShopMill / ShopTurn 的刀具管理中，可以为每把刀具指定冷却液功能和刀具专用功能，这些功能在换入刀具时激活。可通过如下机床数据将冷却液功能指定为某个机床功能 (M 功能)：

MD52230 \$MCS_M_CODE_ALL_COOLANTS_OFF	用于全部冷却液关闭的 M 代码
MD52231 \$MCS_M_CODE_COOLANT_1_ON	用于冷却液 1 打开的 M 代码
MD52232 \$MCS_M_CODE_COOLANT_2_ON	用于冷却液 2 打开的 M 代码
MD52233 \$MCS_M_CODE_COOLANT_1_AND_2_ON	用于两个冷却液打开的 M 代码

刀具专用功能 (ShopMill)

刀具专用功能 1...4 为其它 M 功能所设计，这些功能可以针对某个刀具激活。例如：第三种冷却液、转速监控、刀具折断监控等。

可以编程最多 8 个 M 功能用于打开或关闭刀具专用功能。通过下列机床数据进行定义：

MD52281 \$MCS_TOOL_MCODE_FUNC_ON[0..3]	刀具专用功能打开的 M 代码
= - 1	

MD52282 \$MCS_TOOL_MCODE_FUNC_OFF[0...3]	用于关闭刀具专用功能的 M 代码
= - 1	

用于刀具专用功能的 M 功能由 ShopMill 循环在换刀时执行。如果勾选了该项目，则相应的 M 代码由 MD52281 生成。如果未勾选该项目，则相应的 M 代码由 MD52282 生成。

9.2 配置操作界面

如果您不希望在换刀时自动任何附加的 M 代码，可以将对应的机床数据设为 -1。之后刀具列表中的对应栏会隐藏。

机床数据的缺省值为 -1，不为相应功能输出 M 值。

工作流程

刀具专用功能的 M 功能由 ShopMill 循环在程序段中生成（最多可有 4 个 M 指令）。

M 指令的数量和顺序取决于机床数据 MD52281 至 MD52282 中的设置和编程。

若 M 功能的使用小于 100，则 M 指令的数量和顺序无任何意义。可在用户 PLC 的已解码区域（DB250x.DBB1000 至 DB250x.DBB1012）中使用，与编程中的数量和顺序无关。

但是，若 M 功能的使用超过 100，则须在用户 PLC 中自行解码。因此，应对数量和顺序予以考虑（M 功能 1 至 M 功能 4）。

运行方式 JOG

若在 JOG 运行方式中，不会向接口输出冷却液和刀具专用功能。在 JOG 运行方式下换刀时，操作者可通过机床控制面板上的按钮触发这些功能（通过 PLC 用户程序完成）。

示例：机床数据设置

MD52281 \$MCS_TOOL_MCODE_FUNC_ON[0]	= 90
MD52281 \$MCS_TOOL_MCODE_FUNC_ON[1]	= 92
MD52281 \$MCS_TOOL_MCODE_FUNC_ON[2]	= 94
MD52281 \$MCS_TOOL_MCODE_FUNC_ON[3]	= 96

MD52282 \$MCS_TOOL_MCODE_FUNC_OFF[0]	= 91
MD52282 \$MCS_TOOL_MCODE_FUNC_OFF[1]	= - 1
MD52282 \$MCS_TOOL_MCODE_FUNC_OFF[2]	= 95
MD52282 \$MCS_TOOL_MCODE_FUNC_OFF[3]	= 97

现在，刀具列表栏 1 到 4 中显示刀具专用功能。

以下栏处于激活状态：

刀具专用功能 1:	打开
刀具专用功能 2:	关闭
刀具专用功能 3:	关闭
刀具专用功能 4:	关闭

在换刀时 ShopMill 循环生成 M 功能 M90、M95 和 M97。

“冷却液 1/2 开/关”和“刀具专用功能 1、...”之类文字可以加以更改，见章节 标准文本的标识 (页 313)。为了使刀具列表内的文字和程序区内机床功能的文字相互匹配，请仔细阅读下文。

为“机床功能”窗口创建用户专用文本

若可激活或取消所需的冷却液和刀具专用功能，则可在操作区域“程序”中通过输入窗口“机床功能”进行编程。可创建在此窗口中使用的刀具专用功能的文本和气泡悬浮框。

在文件“slstepforms_xxx.ts”中为 M 功能定义文本。

如果需要其他语言的文本，则须为每种语言创建一个单独的文件。这些文件的名称只能通过语言标识“xxx”加以区分。请只使用规定的语言标记。

文件“slstepforms_xxx.ts”

标签	含义
source	<p>标签的文本标记: SIStepLabels</p> <p>可定义最多 4 个 M 功能:</p> <p>T_LAB_USER_MACHINE_FUNC_1</p> <p>T_LAB_USER_MACHINE_FUNC_2</p> <p>T_LAB_USER_MACHINE_FUNC_3</p> <p>T_LAB_USER_MACHINE_FUNC_4</p> <p>气泡悬浮框的文本标记: SIStepToolTip</p> <p>对于每个机床功能 x (1 - 4) 有 3 个气泡悬浮框:</p> <p>T_TT_USER_MACHINE_FUNC_x: 未选择机床功能时的标准气泡悬浮框。</p> <p>T_TT_USER_MACHINE_FUNC_x_ON: 激活了机床功能时的气泡悬浮框</p> <p>T_TT_USER_MACHINE_FUNC_x_OFF: 禁用机床功能的气泡悬浮框</p>
translation	<p>操作界面上显示的文本。</p> <p>标签文本: 可只输入单行文本。省略条目“lines”。</p> <p>气泡悬浮框: 可输入多行文本。通过字符“%n”进行换行。</p>
chars	<p>文本长度</p> <p>最大文本长度为 30 个字符。</p>
lines	<p>行数</p> <p>最大行数为 3。</p>
remark	<p>独立注释 - 在操作界面上不显示此条目。</p>

步骤

1. 可从以下目录中复制模板文件“oem_slstepforms_deu.ts” : /siemens/sinumerik/hmi/template/lng.
2. 在目录 /oem/sinumerik/hmi/lng 或 /user/sinumerik/hmi/lng 中保存或创建文件。
3. 将文件名更改为“slstepforms_deu.ts” 。
如果需要其它语言的文本, 必须为每种语言创建一个单独的文件。在文件名称中使用相应的语言标记并保存此文件。
4. 打开此文件, 并在区域 <message> 和 </message> 中定义文本。
5. 重新启动系统。
为了在程序运行时能显示这些文本, 必须先将文件转换为二进制格式。此转换仅在启动时进行。

示例：文件“slstepforms_deu.ts”

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS><TS>
<context>
  <name>SlStepLabels</name>
  <message>
    <source>T_LAB_USER_MACHINE_FUNC_1</source>
    <translation>WZ spez. Funktion 1</translation>
    <chars>20</chars>
  </message>
</context>
<context>
  <name>SlStepToolTip</name>
  <!-- 用户机床数据 1 的刀具栏 -->
  <message>
    <source>T_TT_USER_MACHINE_FUNC_1</source>
    <translation>Anwender%nMaschinenfunktionen 1</translation>
    <chars>25</chars>
    <lines>3</lines>
  </message>
</context>
</TS>
```

9.2.16 重新激活时的换刀原因

前提条件



软件选件

该功能需要使用软件“MC 信息系统 TDI 统计”。

功能

在重新激活刀具时可收集换刀原因，例如将这些原因用于统计。按下软键“重新激活”，打开选择窗口“换刀原因”。在复选框中选择换刀原因。换刀记录在待配置的刀具参数中或 SICap Service 局部变量中。

换刀原因

可任意调整或扩展换刀原因。操作软件提供以下预定义文本用于配置：

文本标记	文本
TM_DGL_DESIRED_PIECES_ELAPSED	设定工件数耗尽
TM_DGL_TOOL_BREAKAGE	刀具受损
TM_DGL_CUTTING_EDGE_DAMAGE	刀沿受损
TM_DGL_PREMATURE_WEAR	早期磨损
TM_DGL_TYPE_REEQUIPPING	类型改换
TM_DGL_TOOL_TRIAL	刀具试验
TM_DGL_PREVENTIVE_CHANGE	预防性更换

标签<TOOLCHANGEREASON>

在<TOOLCHANGEREASON>标签中配置功能：

参数	名称
Enabled	true - 在重新配置时显示选择窗口“换刀原因”。
	false - 缺省值，不显示选择窗口。
OnlyOneReasonAllowed	只可以选择一个换刀原因。
UseCapLocalParameter	换刀原因记录在 SICap Service 局部变量中。
REASONXXX	XXX 为内部编号。换刀原因的编号必须唯一。
Text	换刀原因的文本标记。
Parameter	参数标记，其中记录了是否选择了换刀原因。
BitMask	在 SICap Service 局部变量中记录换刀原因的位标记。位标记作为整数值输入。位 0 -> 1、位 1 -> 2、位 2 -> 4

SICap Service 局部变量

局部变量标识为“/Hmi/ToolEdgeReplacementReason”，类型为字符串型。

下表列出了参数及其含义：

参数	含义
datetime	事件的日期和时间
containerNo	管理范围编号
toaNo	To 区编号
toolNo	刀具号
edgeNo	刀沿号
toolIdent	刀具名称
toolDuplo	姐妹刀具编号
ncuName	NCU 名称
reactivationReason	刀具更换原因
monitoringMode	监控类型
remainingBefore	重新激活前的剩余工件数目/寿命

示例 1

下面的示例中使用操作软件中已预定义的换刀原因文本。在此示例中换刀原因保存在尚需配置的参数中，例如 OEM 刀具数据 1，位 1 及之后的位。

```
<TOOLCHANGEREASON>
  <Enabled value="true" type="bool" />
  <REASON1>
    <Text value="TM_DGL_DESIRED_PIECES_ELAPSED" type="QString" />
    <Parameter value="ToolUser_1_Bit1" type="QString" />
  </REASON1>
  <REASON2>
    <Text value="TM_DGL_TOOL_BREAKAGE" type="QString" />
    <Parameter value="ToolUser_1_Bit2" type="QString" />
  </REASON2>
  <REASON3>
    <Text value="TM_DGL_CUTTING_EDGE_DAMAGE" type="QString" />
    <Parameter value="ToolUser_1_Bit3" type="QString" />
  </REASON3>
  <REASON4>
    <Text value="TM_DGL_PREMATURE_WEAR" type="QString" />
    <Parameter value="ToolUser_1_Bit4" type="QString" />
  </REASON4>
```

```

<REASON5>
  <Text value="TM_DGL_TYPE_REEQUIPPING" type="QString" />
  <Parameter value="ToolUser_1_Bit5" type="QString" />
</REASON5>
<REASON6>
  <Text value="TM_DGL_TOOL_TRIAL" type="QString" />
  <Parameter value="ToolUser_1_Bit6" type="QString" />
</REASON6>
<REASON7>
  <Text value="TM_DGL_PREVENTIVE_CHANGE" type="QString" />
  <Parameter value="ToolUser_1_Bit7" type="QString" />
</REASON7>
</TOOLCHANGEREASON>

```

示例 2

本示例中也使用了预定义的换刀原因文本。只允许选择一个换刀原因作为有效输入。换刀原因记录在 **SICap Service** 局部变量中。

```

<TOOLCHANGEREASON>
  <Enabled value="true" type="bool" />
  <OnlyOneReasonAllowed value="true" type="bool" />
  <UseCapLocalParameter value="true" type="bool" />
  <REASON1>
    <Text value="TM_DGL_DESIRED_PIECES_ELAPSED" type="QString" />
    <BitMask value="1" type="int" />
  </REASON1>
  <REASON2>
    <Text value="TM_DGL_TOOL_BREAKAGE" type="QString" />
    <BitMask value="2" type="int" />
  </REASON2>
  <REASON3>
    <Text value="TM_DGL_CUTTING_EDGE_DAMAGE" type="QString" />
    <BitMask value="4" type="int" />
  </REASON3>
  <REASON4>

```

```

    <Text value="TM_DGL_PREMATURE_WEAR" type="QString" />
    <BitMask value="8" type="int" />
</REASON4>
<REASON5>
    <Text value="TM_DGL_TYPE_REEQUIPPING" type="QString" />
    <BitMask value="16" type="int" />
</REASON5>
<REASON6>
    <Text value="TM_DGL_TOOL_TRIAL" type="QString" />
    <BitMask value="32" type="int" />
</REASON6>
<REASON7>
    <Text value="TM_DGL_PREVENTIVE_CHANGE" type="QString" />
    <BitMask value="64" type="int" />
</REASON7>
</TOOLCHANGEREASON>

```

另见

新参数的配置参见配置列表参数 (页 275)。

9.2.17 配置载码器连接

前提条件



软件选件

使用此功能需要以下选件“Access MyTool ID”（Tool Ident Connection）。

标签<IDENTCONNECTIONCONFIGURATION>

<IDENTCONNECTIONCONFIGURATION> 标签中包括操作界面刀具列表中载码器连接配置的条目。通过 TDI Ident Connection 进行载码器连接。

可使用以下刀具列表中的功能：

1. 从载码器中创建刀具：



图 9-2 从载码器中新建刀具

刀具数据从载码器中读取并在“新建刀具”窗口中显示。按下“确认”，刀具便在 NC 中创建，并以载码器数据初始化。

2. 卸载刀具，数据写入载码器：
刀具被卸载，刀具数据接着被写入到载码器中。也可通过设置（如下）删除刀具。
3. 删除刀具，数据写入载码器：
刀具被卸载，刀具数据接着被写入到载码器中。接着该刀具被从 NC 中删除。通过设置（如下）也可以在删除刀具时隐藏“到载码器”软键。

可以进行如下设置：

条目	含义
Enabled	true - 提供载码器连接。 false - 不提供载码器连接。
ShowInternalMessage	true - 显示附加内部信息，用于故障分析。 false - 缺省设置：不显示内部故障信息。

条目	含义
UnloadWithDelete	<p>true - “卸载刀具，数据写入载码器”时删除刀具。“删除刀具”时不提供“到载码器”功能。</p> <p>false - 缺省设置：“卸载刀具，数据写入载码器”时，刀具仅被卸载但仍保留在 NC 存储器中。</p>
OnlyInLoadStation	<p>true - 缺省设置：一方面只能将已装载的刀具保存“到载码器”上；另一方面也可直接在刀库中“从载码器”创建新刀具。</p> <p>false - 刀具可“从载码器”创建或删除后保存“到载码器”（与刀库位置无关）。</p>
CreateNewToolWithDialog	<p>true - 缺省设置：在从载码器创建刀具之前在“新刀具”窗口中显示刀具数据。</p> <p>false - 直接在刀具列表中创建刀具。</p>

示例

```

<IDENTCONNECTIONCONFIGURATION>
  <Enabled value="true" type="bool" />
  <ShowInternalMessage value="false" type="bool" />
  <UnloadWithDelete value="false" type="bool" />
  <OnlyInLoadStation value="true" type="bool" />
  <CreateNewToolWithDialog value="true" type="bool" />
</IDENTCONNECTIONCONFIGURATION>

```

参考文档

关于使用刀具管理载码器的更多信息参见下列参考文献：

功能手册 SINUMERIK Integrate Access MyTool Ident Connection
(MCIS TDI Ident Connection)

9.3 创建 OEM 文本

9.3.1 创建 OEM 文本

简介

本章节描述了使用各语言创建 OEM 文本的步骤。

OEM 文本包含在文件“sltmlistdialog_XXX.ts”中。

每种语言均需创建一个单独的文件。可使用带不同语言标识的文件名称来区分这些文件。“xxx”为语言标记。

创建文本

所有上面章节中的 OME 文本（例如：气泡悬浮框的 OEM 文本）都可以变为特定语言的文本，方法是，在标签“source”中输入 OEM 文本作为源语言，然后在标签“translation”输入到目标语言的翻译。

条目	含义
source	文本标记 参见示例 1：带有新文本标记“MY_NEW_TEXT”的一行文本。 参见示例 2：带有新文本标记“MY_NEW_TEXT_2_LINES”的两行文本。
translation	操作界面上显示的文本。 可输入单行和多行文本。通过“%n”进行换行。 参见示例 1：显示单行文本“我的新文本”。 参见示例 2：显示两行文本“我的新 - 文本”。
chars	文本长度 参见示例 1：文本长度确定为 30 个字符。 参见示例 2：文本长度确定为每行 10 个字符。
lines	行数 参见示例 2：行数为 2。
remark	不显示自己标注的注释。

步骤

1. 可从如下目录中复制模板文件“oem_sltmlistdialog_deu.ts”：`/siemens/sinumerik/hmi/template/lng`.
2. 将复制件保存到目录 `/oem/sinumerik/hmi/lng` 或 `/user/sinumerik/hmi/lng` 中。
3. 更改文件名“sltmlistdialog_deu.ts”。
若需创建其它语种的文本，必须为此语种创建一个自身文件。在文件名称中使用相应的语言标记并保存此文件。
4. 打开此文件，并在区域 `<message>` 和 `</message>` 中确定文本。
5. 重新启动系统。
为了在程序运行时能显示这些文本，必须先将文件转换为二进制格式。此转换仅在启动时进行。

9.3.2 标准文本的标识

标准文本的标记

借助该标记您可以修改 OEM 参数、冷却液和刀具专用功能的标准文本。其中包含了刀具表的列标题、气泡悬浮框，有可能也包含了细节图中的名称。

下面列出了这些标记：

OEM 刀具参数标记

"ToolUser_1" 至 "ToolUser_10"

"ToolAppl_1" 至 "ToolAppl_10"

列标题	Tooltip
TM_HL_TOOL_USER_1 至...10	TM_TT_TOOL_USER_1 至...10
TM_HL_TOOL_APPL_1 至...10	TM_TT_TOOL_APPL_1 至...10

刀具详细信息	刀具详细信息 - 所有参数
TM_HL_TOOL_USER_1_DETAILS 至... 10_DETAILS	TM_HL_TOOL_USER_1_LONG 至...10_LONG
TM_HL_TOOL_APPL_1_DETAILS 至... 10_DETAILS	TM_HL_TOOL_APPL_1_LONG 至...10_LONG

9.3 创建 OEM 文本

OEM 刀沿参数标记

"EdgeUser_1" 至 "EdgeUser_10"

"EdgeAppl_1" 至 "EdgeAppl_10"

列标题	Tooltip
TM_HL_EDGE_USER_1 至...10	TM_TT_EDGE_USER_1 至...10
TM_HL_EDGE_APPL_1 至...10	TM_TT_EDGE_APPL_1 至...10

刀具详细信息	刀具详细信息 - 所有参数
TM_HL_EDGE_USER_1_DETAILS 至... 10_DETAILS	TM_HL_EDGE_USER_1_LONG 至...10_LONG
TM_HL_EDGE_APPL_1_DETAILS 至... 10_DETAILS	TM_HL_EDGE_APPL_1_LONG 至...10_LONG

OEM 监控参数标记

"EdgeSupUser_1" 至 "EdgeSupUser_10"

"EdgeSupAppl_1"至"EdgeSupAppl_10"

列标题	Tooltip
TM_HL_EDGE_SUPUSER_1 至...10	TM_TT_EDGE_SUPUSER_1 至...10
TM_HL_EDGE_SUPAPPL_1 至...10	TM_TT_EDGE_SUPAPPL_1 至...10

刀具详细信息	刀具详细信息 - 所有参数
TM_HL_EDGE_SUPUSER_1_DETAILS 至... 10_DETAILS	TM_HL_EDGE_SUPUSER_1_LONG 至... 10_LONG
TM_HL_EDGE_SUPAPPL_1_DETAILS 至... 10_DETAILS	TM_HL_EDGE_SUPAPPL_1_LONG 至... 10_LONG

OEM 多刀参数标记

"MultiToolUser_1" 至 "MultiToolUser_10"

"MultiToolAppl_1" 至 "MultiToolAppl_10"

列标题	Tooltip
TM_HL_MULTITOOOL_USER_1 至...10	TM_TT_MULTITOOOL_USER_1 至...10
TM_HL_MULTITOOOL_APPL_1 至...10	TM_TT_MULTITOOOL_APPL_1 至...10

刀具详细信息	刀具详细信息 - 所有参数
TM_HL_MULTITOOOL_USER_1_DETAILS 至... 10_DETAILS	TM_HL_MULTITOOOL_USER_1_LONG 至... 10_LONG
TM_HL_MULTITOOOL_APPL_1_DETAILS 至... 10_DETAILS	TM_HL_MULTITOOOL_APPL_1_LONG 至... 10_LONG

OEM 刀库位置参数标记

"MagPlaceUser_1" 至 "MagPlaceUser_10"

"MagPlaceAppl_1" 至 "MagPlaceAppl_10"

列标题	Tooltip
TM_HL_MAGPLACE_USER_1 至...10	TM_TT_MAGPLACE_USER_1 至...10
TM_HL_MAGPLACE_APPL_1 至...10	TM_TT_MAGPLACE_APPL_1 至...10

OEM 多刀位置参数标记

"MultiToolPlaceUser_1" 到 "MultiToolPlaceUser_10"

"MultiToolPlaceAppl_1" 到 "MultiToolPlaceAppl_10"

列标题	Tooltip
TM_HL_MTPLACE_USER_1 至...10	TM_TT_MTPLACE_USER_1 至...10
TM_HL_MTPLACE_APPL_1 至...10	TM_TT_MTPLACE_APPL_1 至...10

冷却液和刀具专用功能标记

"Coolant1" 和 "Coolant2"

9.3 创建 OEM 文本

"MFunction1" 至 "MFunction4"

列标题	Tooltip
无此列	TM_TT_STATE_COOL_1 和 ...2
TM_HL_MFCT1 至 ...4	TM_TT_MFCT1 至 ...4

9.3.3 OEM 文本举例

单行和两行文本示例 1

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TS><TS>
<!-- ***** -->
<!-- OEM 文本定义 -->
<!-- ***** -->
<context>
  <name>SlTmListForm</name>
  <!-- ***** -->
  <!-- 在此注释后输入文本 -->
  <!-- ***** -->
  <!-- ***** -->
  <!-- 第 1 个单行文本示例 -->
  <!-- ***** -->
  <!-- <message>
    <source>MY_NEW_TEXT</source>
    <translation>我的新文本</translation>
    <chars>30</chars>
  </message>-->
  <!-- ***** -->
  <!-- 第 2 个双行文本示例 -->
  <!-- ***** -->
  <!-- <message>
    <source>MY_NEW_TEXT_2_LINES</source>
    <translation>我的新%n 文本</translation>
    <remark>我对文本的注视</remark>
    <chars>10</chars>
    <lines>2</lines>
  </message>

```

```

        </message>-->
<!-- ***** -->
<!-- 在此注释前输入文本 -->
<!-- ***** -->
</context>
</TS>

```

示例 2

为章节配置列表参数 (页 275) 中的示例配置一个新的参数:

```

<PARAMETERCONFIGURATION>
  <EdgeUser_1_Bit0>
    <Base value="EdgeUser_1" type="QString" />
    <ItemType value="CheckBox" type="QString" />
    <DisplayMode value="AnyMode" type="QString" />
    <BitMask value="1" type="int" />
    <Width value="17" type="int" />
    <HeadLine value="TM_HL_EDGE_USER_1_Bit0" type="QString" />
    <ToolTip value="TM_TT_EDGE_USER_1_Bit0" type="QString" />
  </EdgeUser_1_Bit0>
</PARAMETERCONFIGURATION>

```

HeadLine 和 ToolTip 中定义的文本需要以特定语言显示, 为此, 必须将这两个文本输入到文件“sltmlistdialog_xxx.ts”中。

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TS><TS>
<!-- ***** -->
<!-- OEM 文本定义 -->
<!-- ***** -->
<context>
  <name>SlTmListForm</name>
<!-- ***** -->
<!-- 在此注释后输入文本 -->
<!-- ***** -->
<!-- ***** -->
<!-- 翻译标题-->
<!-- ***** -->

```

9.4 示例

```

<message>
  <source>TM_HL_EDGE_USER_1_Bit0</source>
  <translation>重型%n 刀具</translation>
  <chars>10</chars>
  <lines>2</lines>
</message>
<!-- ***** -->
<!-- 翻译气泡悬浮框-->
<!-- ***** -->

<message>
  <source>TM_TT_EDGE_USER_1_Bit0</source>
  <translation>重型刀具</translation>
  <chars>20</chars>
</message>

<!-- ***** -->
<!-- 在此注释前输入文本 -->
<!-- ***** -->

</context>
</TS>

```

9.4 示例

9.4.1 示例：配置 OEM 刀具列表

9.4.1.1 修改配置文件

下文摘选了配置文件的部分内容：

- 铣削工艺：sltmlistconfig.xml
- 车削工艺：sltmturninglistconfig.xml

本例中的文件位于目录“user/sinumerik/hmi/cfg”下。

本例说明了如何配置 OEM 刀具列表。

- 列 6 显示了 OEM 刀具参数 1。
- 列 7 插入了一个自定义的新参数“冷却剂 3”，显示 OEM 刀具参数 2 的位 0。
- 列 8 显示了 OEM 刀沿参数 1。

示例

文件“sltmconfig.xml”或“sltmturningconfig.xml”

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CONFIGURATION>
  <!-- **** 列表配置的标记 *** -->
  <LISTCONFIGURATION>
    <!-- **** OEM 刀具列表 *** -->
    <SlTmTooloemForm>
      <!-- **** 激活 OEM 刀具列表 *** -->
      <Enabled value="true" type="bool" />
      <!-- **** 列 1 到 5 已填入默认值。 *** -->
      <!-- **** 列 6 *** -->
      <COLUMN6>
        <!-- **** 用于所有类型的刀具 *** -->
        <TOOLTYPE_ALL>
          <!-- **** 显示 OEM 刀具参数 1 *** -->
          <Item value="ToolUser_1" type="QString" />
        </TOOLTYPE_ALL>
      </COLUMN6>
      <!-- **** 列 7 *** -->
      <COLUMN7>
        <!-- **** 用于所有类型的刀具 *** -->
        <TOOLTYPE_ALL>
          <!-- **** 优选刀具，自定义参数 *** -->
          <!-- **** 显示 OEM 刀具参数 2 位 0 *** -->
          <Item value="PriorTool" type="QString" />
        </TOOLTYPE_ALL>
      </COLUMN7>
      <!-- **** 列 8 *** -->
      <COLUMN8>
```

```

<!-- **** 用于所有类型的刀具 *** -->
<TOOLTYPE_ALL>
  <!-- **** 显示 OEM 刀沿参数 1 *** -->
  <Item value="EdgeUser_1" type="QString" />
</TOOLTYPE_ALL>
</COLUMN8>
</SlTmTooloemForm>
</LISTCONFIGURATION>
<!-- **** 配置自定义参数的标记 *** -->
<PARAMETERCONFIGURATION>
  <!-- **** 修改 OEM 刀具参数 1 *** -->
  <ToolUser_1>
    <!-- **** 列宽 *** -->
    <Width value="36" type="int" />
    <!-- **** 数据类型修改为无符号的整数 *** -->
    <DisplayMode value="UnsignedIntegerMode" type="QString" />
  </ToolUser_1>
  <!-- **** 自定义参数的标记 *** -->
  <!-- **** 优选刀具 *** -->
  <PriorTool>
    <!-- **** 新参数以 OEM 刀具参数 2 为基础 *** -->
    <Base value="ToolUser_2" type="QString" />
    <!-- **** 计算位 0 *** -->
    <BitMask value="1" type="int" />
    <!-- **** 标题文字的标记 *** -->
    <HeadLine value="TMO_HL_PRIOR_TOOL" type="QString" />
    <!-- **** 气泡悬浮框文字的标记 *** -->
    <ToolTip value="TMO_TT_PRIOR_TOOL" type="QString" />
    <!-- **** 列宽 *** -->
    <!-- **** 推荐宽度 640x480 - 19 像素 *** -->
    <!-- **** 800x600 - 21, 1024x768 - 17 *** -->
    <Width value="19" type="int" />
    <!-- **** 参数的显示项目 *** -->
    <ItemType value="CheckBox" type="QString" />
  </PriorTool>
</PARAMETERCONFIGURATION>
</CONFIGURATION>

```


9.4.1.2 修改用户文本文件

下文摘选了用户文本文件“sltmlistdialog_deu.ts”的部分内容。

本例中的文件位于目录“user/sinumerik/hmi/lng”下。

示例

文件“sltmlistdialog_deu.ts”

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TS><TS>

<!-- ***** -->
<!-- OEM 文本定义 -->
<!-- ***** -->
<context>
  <name>SlTmListForm</name>

<!-- ***** -->
<!-- 在注释后输入自己的文本 -->
<!-- ***** -->

  <!-- **** OEM 刀具参数 1 标题 **** -->
  <!-- **** 重量 **** -->
  <message>
    <!-- **** 文本标记 **** -->
    <source>TM_HL_TOOL_USER_1</source>
    <!-- **** 被隐藏的文本 **** -->
    <translation>重%n 量</translation>
    <!-- **** 仅供翻译人员参考的注释 **** -->
    <remark>优选刀具</remark>
    <!-- **** 一行的最大字符数 **** -->
    <!-- **** 仅供翻译人员参考 **** -->
    <chars>7</chars>
    <!-- **** 允许的行数 **** -->
    <!-- **** 仅供翻译人员参考 **** -->
    <lines>2</lines>
  </message>
```

```

<!-- **** OEM 刀具参数 1 的气泡悬浮框 **** -->
<!-- **** 重量 **** -->
<message>
  <source>TM_TT_TOOL_USER_1</source>
  <translation>刀具重量</translation>
  <chars>35</chars>
  <lines>2</lines>
</message>
<!-- **** 自定义参数的标题 **** -->
<!-- **** 优选刀具 **** -->
<message>
  <source>TMO_HL_PRIOR_TOOL</source>
  <translation>V</translation>
  <remark>优选刀具</remark>
  <chars>1</chars>
  <lines>2</lines>
</message>
<!-- **** 自定义参数的气泡悬浮框 **** -->
<!-- **** 优选刀具 **** -->
<message>
  <source>TMO_TT_PRIOR_TOOL</source>
  <translation>优选刀具,%n 使用优选刀具</translation>
  <chars>35</chars>
  <lines>3</lines>
</message>
<!-- **** OEM 刀沿参数 1 标题 **** -->
<!-- **** 刀具最大宽度 **** -->
<message>
  <source>TM_HL_EDGE_USER_1</source>
  <translation>最大%n 宽度</translation>
  <chars>8</chars>
  <lines>2</lines>
</message>
<!-- **** OEM 刀沿参数 1 的气泡悬浮框 **** -->
<!-- **** 刀具最大宽度 **** -->
<message>
  <source>TM_TT_EDGE_USER_1</source>
  <translation>刀具最大%n 宽度%n</translation>
  <chars>35</chars>

```

```

        <lines>2</lines>
    </message>

<!-- ***** -->
<!-- 在注释前输入自己的文本 -->
<!-- ***** -->

</context>
</TS>

```

9.4.2 示例：配置刀库位置类型及名称

9.4.2.1 修改配置文件

下文摘选了配置文件的部分内容：

- 铣削工艺：sltmconfig.xml
- 车削工艺：sltmturningconfig.xml

本例中的文件位于目录“user/sinumerik/hmi/cfg”下。

本例说明了如何配置刀具列表中的位置类型。

在本例中，下拉菜单中的刀库位置类型和刀具位置类型要用简明文字显示。

在“LISTCONFIGURATION”段落中为各列指定参数。列 10 显示刀库位置类型。列 11 显示刀具位置类型。

“PLACETYPECONFIGURATION”段落中指定位置类型的文字。位置类型的编号是“PLACETYPE”末尾的数字。

您可以为每一种位置类型定义下拉菜单选项的文字和气泡悬浮框的文字。

示例

文件“sltmconfig.xml”或“sltmturningconfig.xml”

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CONFIGURATION>
    <!-- **** 列表配置 **** -->
    <LISTCONFIGURATION>
        <!-- **** 刀库列表 **** -->

```

```

<SlTmToolmagazinForm>
  <!-- **** 列 10 **** -->
  <COLUMN10>
    <!-- **** 用于所有类型的刀具 **** -->
    <TOOLTYPE_ALL>
      <!-- **** 显示刀库位置类型 **** -->
      <Item value="MagPlaceTypeIdent" type="QString" />
    </TOOLTYPE_ALL>
  </COLUMN10>
  <!-- **** 列 11 **** -->
  <COLUMN11>
    <!-- **** 用于所有类型的刀具 **** -->
    <TOOLTYPE_ALL>
      <!-- **** 显示刀具位置类型 **** -->
      <Item value="ToolPlaceSpecIdent" type="QString" />
    </TOOLTYPE_ALL>
    <!-- **** 用于所有多刀 **** -->
    <TOOLTYPE_9997>
      <!-- **** 显示所有多刀位置类型 **** -->
      <Item value="MultiToolPlaceSpecIdent" type="QString" />
    </TOOLTYPE_9997>
  </COLUMN11>
</SlTmToolmagazinForm>
</LISTCONFIGURATION>
<!-- **** 刀库位置类型配置 **** -->
<PLACETYPECONFIGURATION>
  <!-- **** 刀库位置类型 1 **** -->
  <PLACETYPE1>
    <!-- **** 下拉菜单选项文字的标记 **** -->
    <Text value="TMMP_ITEM_NORMAL_TOOL" type="QString" />
    <!-- **** 气泡悬浮框文字的标记 **** -->
    <Tooltip value="TMMP_TT_NORMAL_TOOL" type="QString" />
  </PLACETYPE1>
  <!-- **** 刀库位置类型 2 **** -->
  <PLACETYPE2>
    <Text value="TMMP_ITEM_HEAVY_TOOL" type="QString" />
    <Tooltip value="TMMP_TT_HEAVY_TOOL" type="QString" />
  </PLACETYPE2>

```

```

<!-- **** 刀库位置类型 3 **** -->
<PLACETYPE3>
  <Text value="TMMP_ITEM_BIG_TOOL" type="QString" />
  <Tooltip value="TMMP_TT_BIG_TOOL" type="QString" />
</PLACETYPE3>
<!-- **** 刀库位置类型 4 **** -->
<PLACETYPE4>
  <Text value="TMMP_ITEM_SENSITIVE_TOOL" type="QString" />
  <Tooltip value="TMMP_TT_SENSITIVE_TOOL" type="QString" />
</PLACETYPE4>
</PLACETYPECONFIGURATION>
</CONFIGURATION>

```

9.4.2.2 修改用户文本文件

下文摘选了用户文本文件“sltmlistdialog_deu.ts”的部分内容。
本例中的文件位于目录“user/sinumerik/hmi/lng”下。

示例

文件“sltmlistdialog_deu.ts”

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TS><TS>

<!-- ***** -->
<!-- OEM 文本定义 -->
<!-- ***** -->
<context>
  <name>SlTmListForm</name>

<!-- ***** -->
<!-- 在注释后输入自己的文本 -->
<!-- ***** -->
  <!-- **** 刀库位置类型 1 项目 **** -->
  <!-- **** 普通刀具 **** -->
  <message>
    <!-- **** 文本标记 **** -->
    <source>TMMP_ITEM_NORMAL_TOOL</source>

```

```

<!-- **** 被隐藏的文本 *** -->
<translation>普通</translation>
<!-- **** 一行的最大字符数 *** -->
<!-- **** 仅供翻译人员参考 *** -->
<chars>9</chars>
</message>
<!-- **** 刀库位置类型 1 的气泡悬浮框 *** -->
<!-- **** 普通刀具 *** -->
<message>
  <source>TMMP_TT_NORMAL_TOOL</source>
  <translation>普通刀具,%n 没有限制条件</translation>
  <chars>35</chars>
  <lines>2</lines>
</message>
<!-- **** 刀库位置类型 2 项目 *** -->
<!-- **** 重型刀具 *** -->
<message>
  <source>TMMP_ITEM_HEAVY_TOOL</source>
  <translation>重型</translation>
  <chars>9</chars>
</message>
<!-- **** 刀库位置类型 2 的气泡悬浮框 *** -->
<!-- **** 重型刀具 *** -->
<message>
  <source>TMMP_TT_HEAVY_TOOL</source>
  <translation>重型刀具</translation>
  <chars>35</chars>
  <lines>2</lines>
</message>
<!-- **** 刀库位置类型 3 项目 *** -->
<!-- **** 大型刀具 *** -->
<message>
  <source>TMMP_TT_BIG_TOOL</source>
  <translation>大型</translation>
  <chars>9</chars>
</message>
<!-- **** 刀库位置类型 3 的气泡悬浮框 *** -->
<!-- **** 大型刀具 *** -->
<message>

```

```

    <source>TMMP_TT_BIG_TOOL</source>
    <translation>大型刀具</translation>
    <chars>35</chars>
    <lines>2</lines>
</message>
<!-- **** 刀库位置类型 4 项目 **** -->
<!-- **** 易损刀具 **** -->
<message>
    <source>TMMP_ITEM_SENSITIVE_TOOL</source>
    <translation>易损</translation>
    <chars>9</chars>
</message>
<!-- **** 刀库位置类型 4 的气泡悬浮框 **** -->
<!-- **** 易损刀具 **** -->
<message>
    <source>TMMP_TT_SENSITIVE_TOOL</source>
    <translation>注意: 刀具很容易%被损坏</translation>
    <chars>35</chars>
    <lines>2</lines>
</message>

<!-- ***** -->
<!-- 在注释前输入自己的文本 -->
<!-- ***** -->

</context>
</TS>

```

9.5 替换 T、M 或 D 功能

通过以下机床数据和系统变量实现用循环替换 T、M 或 D 功能。

有关换刀循环的信息请参见章节“用于换刀的制造商循环 CUST_T 和 CUST_M6 (页 88)”。

参考文档

- SINUMERIK 828D 调试手册之 CNC 调试；刀具管理
 章节“车床应用示例”
 章节“铣床应用示例”
- SINUMERIK 840D sl 功能手册之刀具管理

机床数据

MD10715 \$MN_M_NO_FCT_CYCLE	需要替换的 M 功能的条目。
= 6	缺省值

MD10716 \$MN_M_NO_FCT_CYCLE_NAME	需要执行的、用于替换 M 功能的循环的名称。
= L6	缺省值

MD10717 \$MN_T_NO_FCT_CYCLE_NAME	需要执行的、用于替换 T 功能的循环的名称。
= TCHANGE	缺省值

MD11717 \$MN_D_NO_FCT_CYCLE_NAME	需要执行的、用于替换 D 功能的循环的名称。
= 补偿	示例值

MD15710 \$MN_TCA_CYCLE_NAME	需要执行的、用于替换 TCA 指令的循环的名称。
= TCHANGE	示例值

替换循环的系统变量

系统变量	含义
\$C_T	刀具 T 编号（数字）。
\$C_T_PROG	布尔变量 该变量显示 \$C_T 中的 T 字是否可用。
\$C_TS	该变量提供刀具名称的字符串。
\$C_TS_PROG	布尔变量 该变量显示 \$C_TS 中的名称是否可用。
\$C_TE	该变量提供 T 字的地址扩展。
\$C_TCA	布尔变量 该变量显示 TCA 替换是否生效。
\$C_DUPLO_PROG	布尔变量 该变量显示在进行 TCA 替换时是否编写了副编号。
\$C_DUPLO	\$C_DUPLO_PROG == TRUE 时，该变量提供编写的副编号的值。
\$C_THNO_PROG	布尔变量 该变量显示在进行 TCA 替换时是否编写了刀架/主轴编号。
\$C_THNO	\$C_THNO_PROG == TRUE 时，该变量提供编写的刀架/主轴编号的值。
\$C_D	该变量提供编写的 D 编号。
\$C_D_PROG	布尔变量 该变量显示 \$C_T 中的补偿编号是否可用。
\$C_DL	该变量提供编写的总和/设置补偿。
\$C_DL_PROG	布尔变量 该变量显示 \$C_DL 中的补偿编号是否可用。
\$C_ME	该变量提供 M 字的地址扩展。
\$C_M	该变量提供替换的地址 M 的值（整数）。
\$C_M_PROG	如果已经编程了换刀所需的 M 功能，该变量提供值 TRUE。

刀具管理接口的系统变量

系统变量	含义
\$AC_TC_FCT	功能号 (NC 的指令号)
\$AC_TC_STATUS	待通过 \$AC_TC_FCT 读取的指令的状态
\$AC_TC_THNO	需要换入新刀具的刀套的编号 (主轴号)。
\$AC_TC_TNO	待更换或准备更换的刀具的内部 T 号。
\$AC_TC_MFN	新刀具的源刀库号。
\$AC_TC_LFN	新刀具的源刀位号。
\$AC_TC_MTN	新刀具的目标刀库号。
\$AC_TC_LTN	新刀具的目标刀位号。
\$AC_TC_MFO	旧 (待更换) 刀具的源刀库号。
\$AC_TC_LFO	旧 (待更换) 刀具的源刀位号。
\$AC_TC_MTO	旧 (待更换) 刀具的目标刀库号。
\$A_TC_LTO	旧 (待更换) 刀具的目标刀位号。
\$AC_TC_CMDT	用于输出 NC 指令的触发变量 (设置用于 IPO)
\$AC_TC_ACKT	用于应答 PLC 的触发变量 (设置用于 IPO)
\$AC_TC_CMDC	指令输出计数器
\$AC_TC_ACKC	应答计数器
\$AC_TC_MMYN	新 (待换入) 刀具的固有刀库号。
\$AC_TC_LMYN	新 (待换入) 刀具的固有刀位号。

补充的多刀系统变量

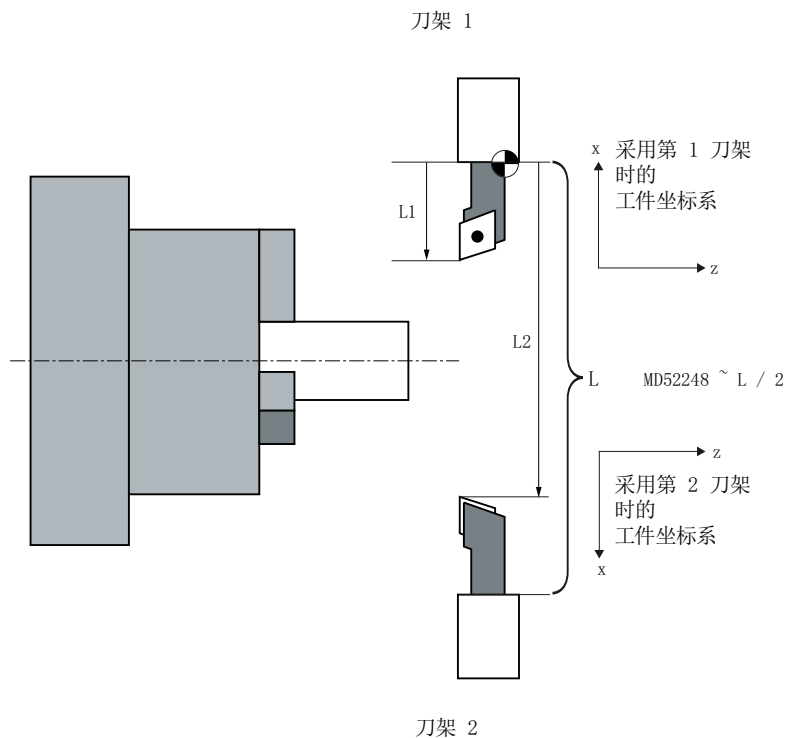
系统变量	含义
\$AC_TC_TOOLIS	所运输刀具的类型 0 = 单个刀具 1 = 多刀, 带作为间距编码的刀位号 2 = 多刀, 带作为间距编码的长度间距 3 = 多刀, 带作为间距编码的角度位移
\$AC_TC_MTDIST	多刀中的新刀具与参考刀位的间距。如果作为间距编码的多刀有长度间距且包含变量, 根据所设的单位制, 长度单位为毫米或英寸。

系统变量	含义
\$AC_TC_MTNLOC	位于换刀或刀具运输中的新多刀的刀位数。
\$AC_TC_MTTN	使用新刀具的多刀的编号。
\$AC_TC_MTLTN	带新刀具的多刀的刀位号。

9.6 双刀架加工

一台车床上可以用两个刀架进行加工，这两个刀架都安装在一个 X 轴上。刀架可为转塔刀架、多点固定刀架或二者的组合。安装在同一轴上的两个刀架只能使用一个刀具进行工作。工件位于两个刀架之间。

所有刀具的参考点都相同，即都在刀架 1 上，不管刀具在哪个刀架上。因此，第二个刀架上的刀具总是比第一个刀架上的刀具长。主要加工在 X 轴的负方向上进行。



刀架定义

通过 MD52248 REV_2_BORDER_TOOL_LENGTH 定义刀具长度极限值 X，从而使刀具能自动分配到两个刀架：

刀具长度 $X < MD52248$: 刀具分配到第 1 个刀架

刀具长度 $X \geq MD52248$: 刀具分配到第 2 个刀架

为了在激活第二个刀架的刀具时激活 X 轴镜像, 必须设置以下 T-/TCA-替代:

MD10717 \$MN_T_NO_FCT_CYCLE_NAME="F_T_REV2"

MD15710 \$MN_TCA_CYCLE_NAME="F_T_REV2"

说明

如果已经有很多现成的、没有写入 X 轴镜像的 G 代码程序, 您要考虑是否需要为 G 代码程序设置镜像。

如果希望在写入 TRANSMIT 指令时一同自动激活所需的镜像, 您必须用一个子程序 (比如: 制造商目录中的一个子程序) 来替换 TRANSMIT 指令:

MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[0]="TRANSMIT"

MD10712 \$MN_NC_USER_CODE_CONF_NAME_TAB[1]="TRANSM"

子程序 TRANSMIT.SPF:

PROC TRANSMIT (INT_NR)

F_T_REV2

TRANSM (_NR)

F_T_REV2

RET

说明

现在在状态显示区中只显示 TRANSM, 而不显示 TRANSMIT。

其他设置

对于所用的 X 轴和 Y 轴镜像和 C 轴位移必须配置以下机床数据:

MD24006 \$MC_CHSFRAME_RESET_MASK	复位后有效的系统框架。
位 4 = 1	系统框架工件参考在复位后仍有效

MD24007 \$MC_CHSFRAME_RESET_CLEAR_MASK	在复位后删除系统框架
位 4 = 0	系统框架工件参考在复位后不删除

MD28082 \$MC_MM_SYSTEM_FRAME_MASK	设置系统框架（SRAM）。
位 4 = 1	工件参考（\$P_WPFRAME）

MD28083 \$MC_MM_SYSTEM_DATAFRAME_MASK	设置系统框架（SRAM）数据管理
位 4 = 1	工件参考（\$P_WPFRAME）

因为坐标系镜像，必须设置以下机床或设置数据：

MD21202 \$MC_LIFTFAST_WITH_MIRROR	镜像下的快速退刀
= 1	在确定回退方向时，激活的轮廓镜像也能反射回退方向。回退方向的镜像只基于垂直于刀具方向的方向分量。

SD42910 \$SC_MIRROR_TOOL_LENGTH	镜像中的刀具磨损符号切换
= 1	相关轴发生镜像的刀具长度分量，通过符号反对其磨损量进行镜像。

功能的基本条件

- 锥旋转在“用两个刀架加工”时未开启。
- B 轴功能（运动学）在“用两个刀架加工”时也未开启。

9.7 使用多刀

9.7.1 激活多刀显示

显示刀具列表中的多刀

在控制系统的目录“User”或“OEM”下的文件“sltmturninglistconfig.xml”（车削）中对刀具列表进行设置。XML 文件位于目录“System-CF-Card/oem/sinumerik/hmi/cfg”下。

有关步骤和设置方法的详细说明参见“刀具管理，配置操作界面 (页 251)”一章。设置模板请参考目录“/siemens/sinumerik/hmi/template/cfg”中的“oem_sltmlistconfig.xml”。

为获取多刀显示，必须将配置文件中的条目“DisableMultiTools”的值设为“false”。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CONFIGURATION>
  <SETTINGS>
    <DisableMultiTools value="false" type="bool" />
  </SETTINGS>
</CONFIGURATION>
```

示例

以下是一个带主主轴和副主轴的典型车床示例。在多刀中，固定有两个刀位的间距编码被设为“刀位”。刀位 1 包含提示框“用于主主轴的刀套”，刀位 2 包含提示框“用于副主轴的刀套”。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CONFIGURATION>
  <SETTINGS>
    <DisableMultiTools value="false" type="bool" />
    <FixNumberOfMultiToolPlaces value="2" type="int" />
    <FixKindOfDistInMultiTools value="1" type="int" />
  </SETTINGS>
  <MULTITOOLPLACETOOLTIPCONFIGURATION>
    <PLACE_1>
      <ToolTip value="TM HOLDER_OF_MAIN_SPINDLE" type="QString" />
    </PLACE_1>
  </MULTITOOLPLACETOOLTIPCONFIGURATION>
</CONFIGURATION>
```

```
</PLACE_1>
<PLACE_2>
    <ToolTip value="TM HOLDER_OF_COUNTER_SPINDLE" type="QString" />
</PLACE_2>
</MULTITOOPLACETOOLTIPCONFIGURATION>
</CONFIGURATION>
```

9.7.2 装卸多刀

在刀库中或从刀库装载和卸载多刀的方式与装载和卸载单个刀具的方式相同。

由 PLC 应答及多刀位置信息

应答位置值“0”(DB40xx.DBWy004)。允许值“0”，其含义是多刀未定位。如果将多刀直接装载在主轴或转塔刀架上，在选择刀位或通过复位初始化后激活位于机床数据 MD20274 \$MC_MULTITOOLLOC_DEFAULT 的刀位上的刀具。

示例

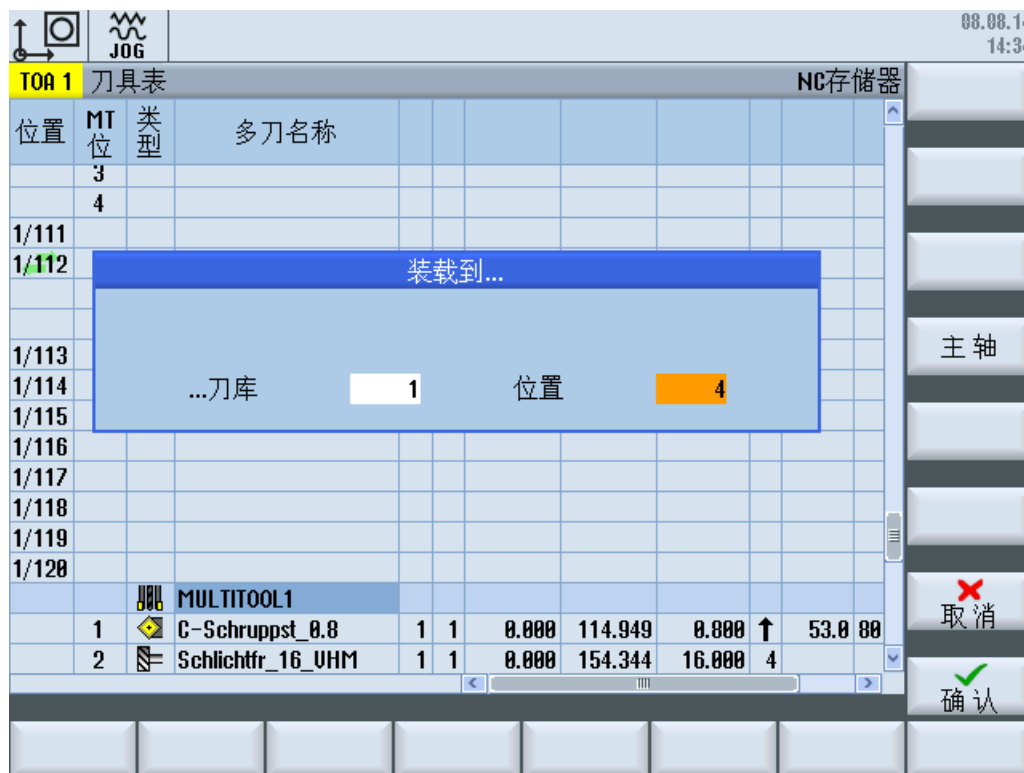


图 9-3 多刀装载对话框

下表中列出了任务状态及应答准备，未给定装载位置。

数据块	含义	值
DB4100.DBB0	任务	10000001
DB4100.DBB1	任务描述	00000001
DB4100.DBW6	源刀库号	9999
DB4100.DBW8	源刀位号	1
DB4100.DBW10	目标刀库号	1
DB4100.DBW12	目标刀位号	4
DB4100.DBW56	多刀类型	1
DB4100.DBW58	多刀刀位数量	0
DB4100.DBD60	多刀间距	0.0000
DB4100.DBW64	多刀编号	0
DB4100.DBW66	多刀刀位号	0

数据块	含义	值
DB4100.DBW68	刀套号	0
DB4000.DBB0	应答	00000001
DB4000.DBW4	多刀刀位号	0

除“标准”任务数据外，还设置了其他任务数据：

- DB4100.DBX0.7: 扩展区域中有数据。 刀具为多刀。
- DB4100.DBW56: 该数据包含间距编码方式 (= 1, = 刀位)。

扩展区域中的所有剩余数据都为“0”。 没有更多关于 NC 的信息。 应答方式为总应答。

参考文档

有关多刀定位的详细说明及示例请参考以下文档：

SINUMERIK 840D sl 功能手册之刀具管理

9.7.3 多刀中的换刀

通过 T=<刀具名称> 调用刀具

换刀指令的编程方式保持不变。 NC 程序中的编程：

```
T="ROUGHING_TOOL_2"
```

该刀具调用选中一个合适的刀具并将其换入（车削工艺）。

如果刀具位于多刀中，该刀具不影响补偿激活。 只有将用于换刀指令的数据组合在一起时，多刀才开始发挥作用。 为 PLC 指令添加附加数据（多刀刀位号、多刀刀位间距）。 PLC 程序要能使机床作出响应，确保多刀内部生效的刀具能够进行加工。

如果编程了多刀名称，系统会生成一个报警。

示例

下表中列出了任务状态及应答准备，未给定换刀位置。

数据块	含义	值
DB4300.DBB0	任务	10000001
DB4300.DBB1	任务描述	00010110

9.7 使用多刀

数据块	含义	值
DB4300.DBW6	源刀库号	1
DB4300.DBW8	源刀位号	4
DB4300.DBW10	目标刀库号	9998
DB4300.DBW12	目标刀位号	1
DB4300.DBW56	多刀类型	1
DB4300.DBW58	多刀刀位数量	2
DB4300.DBD60	多刀间距	2.0000
DB4300.DBW64	源多刀号	2
DB4300.DBW66	源多刀刀位号	2
DB4300.DBW68	目标多刀号	0
DB4300.DBW70	目标多刀刀位号	0
DB4200.DBB0	应答	00000001
DB4200.DBW4	多刀刀位号	0

除“标准”任务数据外，还设置了其他任务数据：

数据块	含义
DB4100.DBX0.7	扩展区域中有数据。刀具为多刀。
DB4100.DBW56	间距编码方式 (= 1, = 刀位)
DB4300.DBW58	多刀刀位数量
DB4300.DBD60	角度编码时的多刀间距
DB4300.DBW64	新刀具的多刀号
DB4300.DBW66	新刀具的多刀刀位号
DB4300:DBW68	旧刀具的数据（如果有）
DB4300:DBW70	

应答方式为总应答。

通过 T=<刀库号> 调用刀具

在对刀库中的多刀刀位号进行编程时，必须采用“MTL = p”（p = 刀位号）的格式。编程的 MTL 值必须符合所选多刀中现有的刀位号。如果不是这种情况，系统会生成一个报警。

NC 程序中的编程：

T4 MTL=1

示例

下表中列出了任务状态及应答准备，给定了换刀位置。

数据块	含义	值
DB4300.DBB0	任务	10000001
DB4300.DBB1	任务描述	00010110
DB4300.DBW6	源刀库号	1
DB4300.DBW8	源刀位号	4
DB4300.DBW10	目标刀库号	9998
DB4300.DBW12	目标刀位号	1
DB4300.DBW56	多刀类型	1
DB4300.DBW58	多刀刀位数量	2
DB4300.DBD60	多刀间距	1.0000
DB4300.DBW64	源多刀号	2
DB4300.DBW66	源多刀刀位号	1
DB4300.DBW68	目标多刀号	0
DB4300.DBW70	目标多刀刀位号	0
DB4200.DBB0	应答	00000001
DB4200.DBW4	多刀刀位号	0

在 T 换刀循环中，可通过 `$C_MTL_PROG` 读取是否编程了 MTL。可通过 `$C_MTL` 读取编程的 MTL 值。

应答方式为总应答。

DB4200.DBW4: 应答时，可在该数据中将多刀的实际位置传送给 PLC。如果位置与 NC 任务中要求的位置不一致，系统会生成一个报警。如果位置没有传送给 PLC (`DB4200.DBW4 = 0`)，系统则不会检查多刀位置是否正确。只有在总应答时才允许值“0”。

9.8 刀具和刀库管理编辑器

9.8.1 概述

刀具管理提供了通过操作软件定义所有真实刀库和内部刀库特性并根据通道的不同连接相应刀具单元的可能性。刀库、抓刀器和主轴之间的连接可通过一个树状结构进行配置。

常规流程



按如下步骤创建机床模型：

- 创建刀具单元
- 定义缓存位置
- 定义装载位置
- 设置刀库配置
- 链接分配

说明

如果创建一个新的刀具单元，则一个缓存位置默认设置为一个主轴位置，一个装载位置默认设置为一个装载位置，一个真实的刀库默认设置为 5 个刀位。

标记的含义：

符号	含义
	有效使用刀具单元的文件夹
	模板访问文件夹
	刀具主轴/夹具
	抓刀器 1
	抓刀器 2
	显示文件夹中还未激活的有效数据的变更

树形图的含义

树形图中有以下文件夹可用：

- 生效的数据 配置和激活刀库管理。
- 示例： 复制模板到文件夹有效数据。
- 网络/USB 复制新的刀库管理到文件夹有效数据或将现有的用于其他系统。

9.8.2 调用刀具管理

按照如下方式打开刀具管理窗口：

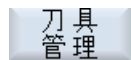
步骤



1. 选择操作区域“调试”。



2. 按下软键“NC”和“机床模型”。



“特殊字符”窗口打开。

9.8.3 编辑刀具管理

通过刀具管理树形图可设置刀库数据并将其相互连接。

说明

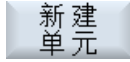
当光标定位在某一元素时，该元素所有可能的功能可被激活。所有其他功能禁用，通过灰色的软键显示。

步骤



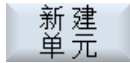
“机床日志”窗口已打开。

新建刀具单元



1. 将光标定位在文件夹元素“有效数据”。
2. 按下软键“新建单元”。
插入标准名称为<刀具单元>的文件夹元素，输入窗口“TO 单元”打开。
3. 在文件夹元素的输入窗口可进行更改。
4. 按下软键“确认”，插入文件夹元素和设置。

新增刀库



1. 将光标定位在文件夹元素“有效数据”。
2. 通过<右光标>和<下光标>键浏览刀具单元的文件夹元素。
3. 按下软键“新建单元”。
插入标准名称为<刀库>的文件夹元素，输入窗口“刀库”打开。
4. 在文件夹元素的输入窗口可进行更改。
5. 按下软键“确认”，插入文件夹元素和设置。

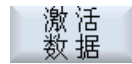
编辑元素



1. 将光标定位在文件夹元素“有效数据”。
2. 通过<右光标>和<下光标>键浏览需要编辑的元素。
3. 按下<光标向右>或<INPUT>键。
下一轮廓元素的输入窗口打开。
4. 输入所需的修改并按下软键“确认”，结束输入。

说明

缓存和刀库位置的数量在上一级文件夹元素<刀库>的输入窗口通过参数“位置”定义。

数据：激活

1. 按下软键“备份数据”。
所有更改的参数都在 NC 中接受并激活。
系统会再次显示所有的故障信息，您可以通过修改数据来排除这些故障。

说明

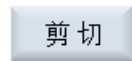
一旦刀具管理作了更改，数据会通过“*”号标记。该数据还未在 NC 中，因此无效。

如果想创建一个和现有刀具管理类似的刀具管理，则可通过复制现有元素并在其他位置插入来节省时间。这样就可以将如无效的子树复制到有效的树结构中。

复制/剪切/粘贴单元

1. 请将光标定位到所需的单元上。
2. 按下软键“复制”。

-或者-



按下软键“剪切”。



3. 将光标定位至所需的位置，并按下软键“粘贴”。
如果该元素在所需的位置已经存在，则可以得到更多的插入选项。



4. 如果要将该单元作为下一个单元附在运动链上，则按下软键“确认”。
单元被选中。

说明：

您可以在对应的输入窗口中修改单元名称。

删除单元



1. 将光标定位到所需的单元上并按下软键“删除单元”。



2. 按下软键“确认”，确认删除。
选中的单元已删除。

9.8.4 设置刀具管理参数

9.8.4.1 创建刀具单元

一个刀具单元包含了单元内的刀具和刀库。一个刀具单元可分配多个通道。刀具单元内的刀具和刀库必须是唯一的。在窗口“TO 单元”确定刀具单元的搜索参数和通道。

目录值

您位于目录“有效数据”-><刀具单元>下的操作区域“调试”->“NC”->“刀具管理”。

通过刀具单元的目录可以在窗口“TO 单元”设置以下参数：

参数	含义
名称	显示在树结构中的名称。
通道号	分配到刀具单元的通道。
刀具查找	搜索方法： <ul style="list-style-type: none"> ● 无搜索方法 ● 激活的刀具 ● 最短行程
搜索空刀位	搜索方法： <ul style="list-style-type: none"> ● 无搜索方法 ● 第一位置向前 ● 当前位置向前 ● 当前位置向后 ● 上一位置向后 ● 对称的当前刀位 ● 1:1 更换

9.8.4.2 定义缓存位置

中间存储器位置时主轴，抓刀器，装载器和交付位。所有中间存储器位置都被归并在一个编号为 9998 的内部中间存储器刀库中。通过窗口“刀库”定义缓存位置数量。通过窗口“位置”定义缓存位置类型。

目录值

您位于目录“有效数据”→<刀具单元>→<缓存>下的操作区域“调试”→“NC”→“刀具管理”。

通过缓存的目录可以在窗口“刀库”定义以下参数：

参数	含义
名称	缓存刀库名称 显示在树结构中的名称。
刀库编号	是（不可变）
位置	缓存位置数量

单元值

缓存位置类型可在窗口“位置”中通过参数定义：

参数	含义
(选择菜单)	缓存类型选择： <ul style="list-style-type: none"> ● 刀具主轴/夹具 ● 装载 ● 夹爪 ● 传输位置
索引	索引计数一个类型的位置

9.8.4.3 定义装载位置

装载位置指的是装载位或装载站。所有装载刀位都被归并在一个编号为 9999 的内部装载刀库中。通过窗口“刀库”定义装载位置数量。通过窗口“位置”定义装载位置类型。

目录值

您位于目录“有效数据”-><刀具单元>-><装载刀库>下的操作区域“调试”->“NC”->“刀具管理”。

通过装载刀库的目录可以在窗口“刀库”定义以下参数：

参数	含义
名称	装载刀库名称 显示在树结构中的名称。
刀库编号	是（不可变）
位置	装载位置数量

单元值

装载位置类型可在窗口“位置”中通过参数定义：

参数	含义
(选择菜单)	装载类型选择： <ul style="list-style-type: none"> ● 装载位 装载位指的是机床上可以直接装载刀库的区域。刀具可以在该位置直接插入刀库。这时将待装载的刀库位置运行到装载位置上。装载位置有例如链刀库。 ● 装载站 装载站是要插入待装载刀具的刀库之外的刀位。从这里刀具通过一个机械运输机构安放在刀库中。通常对于平面或链刀库来说是装载站。
索引	索引计数一个类型的位置

说明

装载位 1 通常自动设置为主轴 1（手动装载位）的装载/卸载位。

9.8.4.4 设置刀库配置

在一个刀具单元可定义多个刀库。通过窗口“刀库”设置刀库配置。通过窗口“位置”定义刀库位置类型。

目录值

您位于目录“有效数据”→<刀具单元>→<刀库>下的操作区域“调试”→“NC”→“刀具管理”。

通过刀库的目录可以在窗口“刀库”定义以下参数：

参数	含义
刀库类型	选择菜单： <ul style="list-style-type: none"> ● 链式刀库 ● 平面刀库 ● 刀塔
名称	刀库名称。 显示在树结构中的名称。
刀库编号	； 刀库号
位置	刀库位置数
刀具查找	搜索方法： <ul style="list-style-type: none"> ● 无搜索方法 ● 从 TOA 接收 ● 激活的刀具 ● 最短行程
搜索空刀位	搜索方法： <ul style="list-style-type: none"> ● 无搜索方法 ● 从 TOA 接收 ● 第一位置向前 ● 当前位置向前 ● 当前位置向后 ● 上一位置向后 ● 对称的当前刀位 ● 1:1 更换
位置编码	<ul style="list-style-type: none"> ● 可变 ● 固定位置编码
位置类型	刀库位置类型标识 (集中更改刀库所有位置的值)
邻接刀位观察	<ul style="list-style-type: none"> ● 是 ● 否 (集中更改刀库所有位置的值)

说明

如果单独更改一个参数位置类型或邻接刀位观察，而且所有位置的值不同，则显示定义“可变”。

单元值

刀库位置类型可在窗口“位置”中通过参数定义：

参数	含义
位置类型	刀库位置类型标识
邻接刀位观察	<ul style="list-style-type: none"> ● 是 ● 否

9.8.4.5 链接分配

通过窗口“分配”可以建立刀具单元单个组件之间的机械连接。

单元值

您位于目录“有效数据”→<刀具单元>下的操作区域“调试”→“NC”→“刀具管理”。

在元素分配中通过选择列表可将以下参数相互链接：

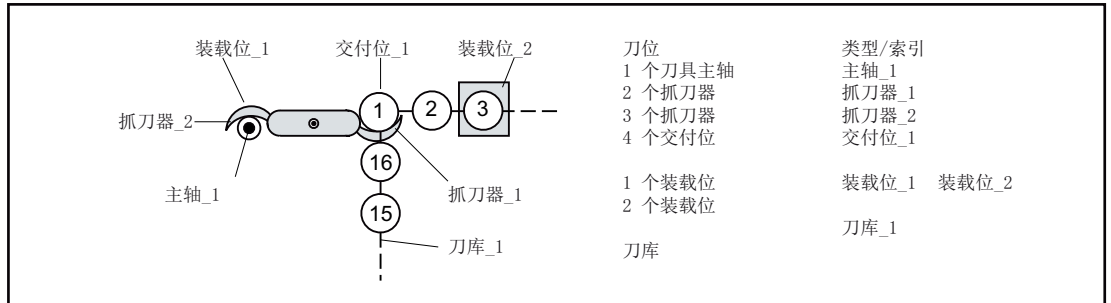
- 主轴/抓刀器/装载器至刀库
- 抓刀器/装载器至主轴
- 装载位至刀库

说明

对于每个刀库分配必须确定相关刀库的刀库零点位置到内部刀库位置之间的距离。

示例

如果刀具从刀库切换到主轴，则可以进行以下分配：



参数	分配
主轴/抓刀器/装载器至刀库	<ul style="list-style-type: none"> • 刀具主轴位置 1 至刀库 (间距 0) • 抓刀器位置 2 至刀库 (间距 0) • 抓刀器位置 3 至刀库 (间距 0) • 交付位 4 至刀库 (间距 0)
抓刀器/装载器至主轴	<ul style="list-style-type: none"> • 刀具主轴位置 1 至抓刀器 2 • 刀具主轴位置 1 至抓刀器 3
装载位至刀库	<ul style="list-style-type: none"> • 装载位位置 1 至刀库 (间距 0) • 装载位位置 2 至刀库 (间距 3)

诊断与维修

10.1 NC/PLC 变量

10.1.1 显示和编辑 PLC 变量和 NC 变量

通过“NC/PLC 变量”窗口，您可以查看、更改 NC 系统变量和 PLC 变量。

在该窗口中会显示一张列表，您可以在其中输入希望查看其当前值的 NC / PLC 变量，列表分为以下几列：

- 变量
NC/PLC 变量的地址
错误的变量背景色为红色，且在“数值”列中显示 #。
- 注释
可随意添加变量注释。
该列可以显示和隐藏。
- 格式
变量显示的格式。
该格式可以固定设置，例如：浮点值。
- 值
NC/PLC 变量的当前数值显示

PLC 变量	
输入端	输入位 (Ex)，输入字节 (EBx)，输入字(EWx)，输入双字(EDx)
输出端	输出位 (Ax)，输出字节 (ABx)，输出字(AWx)，输出双字(Adx)
标志	标志位 (Mx)，标志字节 (MBx)，标志字 (MWx)，标志双字 (MDx)
时间	时间 (Tx)
计数器	计数器 (Zx)
数据	数据模块 (DBx)：数据字 (DBXx)，数据字节 (DBBx)，数据字 (DBWx)，数据双字 (DBDx)

格式	
B	二进制
H	十六进制
D	十进制，无符号
+/-D	十进制，带符号
F	浮点（对于双字）
A	ASCII 字符

变量的输入方式

- PLC 变量
 - EB2
 - A1.2
 - DB2.DBW2
- NC 变量
 - NC 系统变量 - 输入方式
\$AA_IM[1]
 - 用户变量/GUD - 输入方式
GUD/MyVariable[1.3]
 - BTSS - 输入方式
/CHANNEL/PARAMETER/R[u1.2]


说明

NC 系统变量和 PLC 变量

- 系统变量受通道影响，切换通道时，会显示相应通道的数值。
- 用户变量(GUD)无须指定为全局 GUD 或通道专用 GUD，和系统变量句法中的 NC 变量，GUD 数组的索引从 0 开始，即第一个元素从索引 0 开始。
- 窗口中弹出的提示条会说明 NC 变量的 BTSS 输入方式（GUD 除外）。

修改 PLC 变量

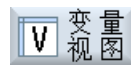
只有具有口令，才能修改 PLC 变量。

 警告
<p>错误的参数设置</p> <p>NC/PLC 变量的状态更改对机床有重大影响，错误的参数设置可能危及人员生命，并导致机床损坏。</p>

修改和删除变量的数值



1. 选择操作区域“诊断”。



2. 按下软键“NC/PLC 变量”。

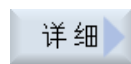
“NC/PLC 变量”窗口打开。

3. 将光标移到“变量”列中，并输入所需的变量。



4. 按下 <INPUT> 键。

现在，操作数及其数值一同显示。



5. 按下软键“详细信息”。

“NC/PLC 变量：详细信息”窗口打开。“变量”、“注释”和“值”的数据完整显示在窗口中。



6. 将光标移到“格式”栏，按下 <SELECT> 键选择所需格式。

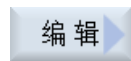


7. 按下软键“显示注释”。

“注释”列随即显示。现在，您可添加注释或者编辑已有注释。



再次按下软键“显示注释”，隐藏注释列。



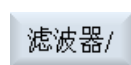
8. 如果要编辑数值，按下软键“更改”。

“值”列切换到可编辑状态。



9. 若需从所有已有变量中选择一个变量并插入该变量，按下软键“插入变量”。

“选择变量”窗口打开。



10. 按下软键“过滤/搜索”，以通过下拉菜单“过滤”筛选变量的显示（比如，按照运行方式组变量筛选），或通过输入栏“搜索”选择所需的变量。



如果要删除操作数条目，按下软键“全部删除”。



11. 按下软键“确认”，确认完成的修改或删除。

-或者-



按下软键“取消”，放弃修改。

说明

插入变量时的“过滤/搜索”操作

按照不同条件进行“过滤/搜索”时，初始值有较大差别。

例如，按下“过滤/搜索”，查找变量 \$R[0]:

- 如果按照“系统变量”筛选，则初始值为 0。
- 如果按照“全部（无过滤）”筛选，则初始值为 1, 此时会用 BTSS 输入方式显示所有信号。

修改操作数

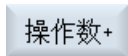
按下软键“操作数+”和“操作数-”，地址或地址索引（由操作数的类型决定）会加或减 1。

说明

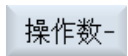
轴名称作为索引

将轴名称作为索引时，软键“操作数+”和“操作数-”无效，比如在 \$AA_IM[X1] 中。

示例



DB97.DBX2.5
 结果: DB97.DBX2.6
 \$AA_IM[1]
 结果: \$AA_IM[2]



MB201
 结果: MB200
 /Channel/Parameter/R[u1.3]
 结果: /Channel/Parameter/R[u1.2]

10.1.2 保存和加载屏幕

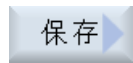
可将“NC/PLC 变量”窗口中进行的变量配置保存在屏幕中，并根据需要加载。

编辑屏幕

更改已加载的屏幕时，屏幕后会有 * 进行标识。

显示关闭后屏幕的名称保存不变。

步骤



1. 已在“NC/PLC 变量”窗口中为所需的变量输入了数值。
2. 按下软键“>>”。
3. 按下软键“保存屏幕”。
“保存屏幕：选择存档位置”窗口打开。
4. 将光标放置在您希望保存当前窗口的模板文件夹上，然后按下“确认”。
“保存屏幕：名称”窗口打开。
5. 输入文件名称并按下软键“确定”。
状态行中会出现消息提示：屏幕已保存至指定的文件夹。
若与已有文件同名，会发出询问。
6. 按下软键“载入屏幕”。
窗口“载入屏幕”打开并显示变量屏幕的模板文件夹。
7. 选中所需文件并按下软键“确认”。
重新返回变量视图。显示已确定的所有 NC 变量和 PLC 变量列表。

10.2 显示维修概览




在窗口“服务概览”中显示机床轴和驱动的运行状态。可以通过软键“扩展选择>”显示或直接通过选择列表根据以下标准配置：

- 所有 NC 轴和驱动 (=预设)
- NC 轴
- 未指定 NC 轴的驱动

显示使能

- 开/关 1
- 关 2
- 关 3
- 来自驱动：使能运行
- 来自电源：使能运行
- 脉冲使能
- 转速调节器使能 NC
- 释放脉冲
- 驱动就绪
- 散热器温度
- 功率部件受 i_{rt} 限制
- 电机温度
- **测量系统 1 生效/测量系统 2 生效**

测量系统 1/2 的状态有以下含义：

符号	含义
	位置测量系统已生效。
	位置测量系统已驻停。 - 或者： - 未配置位置测量系统。
	位置测量系统未生效。

步骤



1. 选择操作区域“诊断”。



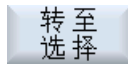
2. 按下菜单扩展键。
显示新水平软键。



3. 按下软键“轴诊断”。
窗口“服务概览”打开。

4. 在下拉菜单“选择”中选择概览。

-或者-



按下软键“转至选择”，选择概览。

10.2.1 选择轴和驱动

要显示某个使能和机床轴状态，可以从窗口“扩展选择”中按任意顺序编排的所有可用轴和驱动中进行选择。

说明

以下选择配置是预先定义的，不可更改或删除：

- 所有 NC 轴和驱动
- NC 轴
- 未指定 NC 轴的驱动

选择概览

您可在选择概览中查看以下信息：

列	说明
总线编号	未指定 NC 轴的驱动的总线编号。 无 NC 轴选项。
从站地址	未指定 NC 轴的驱动的从站地址。 无 NC 轴选项。
设备编号	未指定 NC 轴的驱动的设备编号。 无 NC 轴选项。
驱动对象编号	未指定 NC 轴的驱动的驱动对象编号。 无 NC 轴选项。
名称	对于 NC 轴，其中包含了标准机床轴名称和用户自定义的机床轴名称。 对于未指定 NC 轴的驱动，其为驱动对象名称。
分配	内部对象分配时显示 NC。 外部对象分配时显示 PLC。

说明

在选择概览中，PROFIBUS 和 PROFINET 配置之间有关从站地址和设备编号的信息数据会有所不同。

步骤



1. 选择操作区域“诊断”。



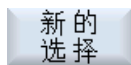
2. 按下菜单扩展键。
显示新水平软键。



3. 按下软键“轴诊断”。
窗口“服务概览”打开。



4. 按下软键“扩展选择”。
窗口“扩展选择：轴和驱动”打开。



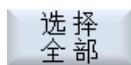
5. 按下软键“新选择”。
对话框“新选择”打开。



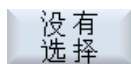
6. 在输入栏中输入一个显示名和一个文件名。
7. 按下软键“确认”结束输入。
窗口“编辑选择”打开，用于编辑新的选择。

8. 在复选框中勾选您要显示的轴和驱动。
-或者-

按下以下某个软键，以进行某项选择：



选择所有轴/驱动。



未选择任何轴/驱动。



只选择内部驱动。





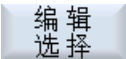

只选择外部驱动。



选择指定了真实驱动的轴。



选择所有至少在一个通道中定义的轴。

- | | |
|---|--|
|  | 9. 按下软键“确认”，确认选择。 |
|  | 10. 按下软键“删除选择”，删除当前选择。 |
|  | 11. 按下软键“编辑选择”，对当前选择进行编辑。
窗口“编辑选择”打开。 |
|  | 12. 按下软键“返回”，进入“服务概览”初始画面。 |




10.2.2 轴诊断

使用



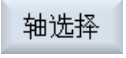

在“轴/主轴信息”窗口会显示以下信息：

- 检测给定值部分（例如：编程的位置给定值、转速给定值、写入的主轴转速给定值）
- 检测实际值部分（例如：位置实际值，测量系统 $\frac{1}{2}$ ，转速实际值）
- 优化轴的位置控制回路（例如：跟随误差、控制偏差、Kv 系数）
- 检测轴的全部控制回路（例如：通过比较位置给定值和位置实际值，转速给定值和转速实际值）
- 检测硬件错误（例如：编码器的控制：如果轴开始机械运动，必然会产生一个位置实际值变化）
- 轴监控设置和检查

步骤

- | | |
|---|--------------------------------|
|  | 1. 选择操作区域“诊断”。 |
|  | 2. 按下菜单扩展键。
显示新水平软键。 |
|  | 3. 按下软键“轴诊断”。
窗口“服务概览”打开。 |
|  | 4. 按下软键“轴信息”。
“轴/主轴信息”窗口打开。 |

10.2 显示维修概览

-  5. 按下软键“轴 +”或“轴 -”，向前或向后选择轴。
-  -或者-
-  按下软键“轴选择”。
“轴直接选择”窗口打开。
通过选择列表直接从提供的轴中选择所需的轴。
-  6. 按下软键“确认”，确认选择。
显示轴的数值。

显示数据

显示数据	含义
跟随误差	位置给定值和有效测量系统 1 或 2 的位置实际值之间的差值 单位：毫米，英寸或者度
控制偏差	位置控制器输入端的位置给定值和有效测量系统 1 或 2 位置实际值之间的差值 单位：毫米，英寸或者度
轮廓偏差（轴向）	该值显示当前的轮廓偏差（通过调速器的调节过程基于负载变化导致跟随误差的波动）。 轮廓偏差可以通过从位置给定值预先计算出来的位置实际值和有效测量系统 1 或 2 的位置实际值之间的差值得出。 单位：毫米，英寸或者度
伺服增益系数（计算出的）	所显示的伺服增益系数将由 NC 根据以下公式计算： <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> $K_v - \text{系数} = \frac{\text{速度给定值}}{\text{跟随误差}} ;$ <p>单位（标准设置）： $\frac{[m/min]}{[毫米]} ;$</p> </div> 速度给定值 = 当前输出至轴/主轴的给定值 文档： 功能手册之基本功能分册；章节“速度，设定值/实际值系统，闭环控制（G2）”

显示数据	含义
已激活的测量系统	<p>0:没有测量系统生效。</p> <p>1:测量系统 1 生效。</p> <p>2:测量系统 2 生效。</p>
测量系统 1 的状态 测量系统 2 的状态	<p>输出测量系统 1/2 的状态： 生效 - 驻停 - 未生效</p>
测量系统 1 位置实际值 测量系统 2 位置实际值	<p>通过测量系统 1/2 测量轴的实际位置 在机床坐标系中显示位置（不考虑零点偏移和刀具补偿）。 单位：毫米，英寸或者度</p>
位置设定值	<p>从插补器到位置控制输出的设定位置 单位：毫米，英寸或者度</p>
绝对补偿值 测量系统 1 绝对补偿值 测量系统 2	<p>显示测量系统 1/2 的绝对补偿值 补偿值是当前轴位置的间隙补偿和丝杠螺距误差补偿的和。 单位：毫米，英寸或者度</p>
补偿 垂度 + 温度	<p>显示由当前轴位置的垂度补偿和温度补偿总和得出的补偿值 单位：毫米，英寸或者度</p>
有效编码器的转速实际值	<p>NC 会分析并显示编码器输出的脉冲。 单位： % 100% 指的是最大转速。</p>
驱动转速给定值	<p>传输至驱动的转速给定值（= 位置控制器和预控制的转速给定值） 单位： % 100% 指的是最大转速给定值。</p>
主轴转速给定值已编程	<p>由用户编程的转速给定值 单位： 转/分钟 例如： 输入： S1000； 显示： 1000 rpm 显示仅适用主轴。</p>
当前主轴转速给定值	<p>符号正确、当前有效的转速给定值，带计算的补偿值和有效的转速限制（通过设定数据或机床数据来规定） 单位： 转/分钟 显示仅适用主轴。</p>

显示数据	含义
主动轴/主轴实际值的位置偏置	<p>如果在同步主轴功能内编程一个位置偏置（跟随主轴和引导主轴之间的角度偏差），则在这里显示基于实际值的当前有效值。</p> <p>单位：毫米、英寸、度</p> <p>文档： 功能手册之扩展功能分册；同步主轴（S3）</p>
主动轴/主轴给定值的位置偏置	<p>如果在同步主轴功能内编程一个位置偏置（跟随主轴和引导主轴之间的角度偏差），则在这里显示基于给定值的当前有效值。</p> <p>单位：毫米、英寸、度</p> <p>文档： 功能手册之扩展功能分册；同步主轴（S3）</p>
倍率	<p>显示进给率倍率开关和主轴倍率开关的有效倍率。</p> <p>单位：%</p>
当前传动级	<p>显示当前实际传动级</p> <p>当主轴分配到轴时才会显示该轴的数据。显示和 NC/PLC 接口信号相对应：</p> <p>DB31, ... DBX16.0-2（实际传动级）</p> <p>文档： 功能手册之基本功能分册；主轴（S1）</p>
参数组（轴）	<p>显示位置控制器的 6 个参数组中哪些是有效的。</p> <p>文档： 功能手册之基本功能分册；章节“速度，设定值/实际值系统，闭环控制（G2）”</p>
控制器模式	<p>显示当前控制器状态：</p> <ul style="list-style-type: none"> ● 位置控制 ● 转速控制 ● 停止 ● 驻留 ● 跟踪 ● 制动 <p>文档： 功能手册之基本功能；不同的 NC/PLC 接口信号与功能(A2)</p>

显示数据	含义
预控制模式	<p>显示轴的动态预控制模式是否有效，如果有效哪个模式有效：</p> <ul style="list-style-type: none"> ● 当前无效 ● 速度 和速度相关的转速预控制有效。 ● 转矩 和加速度相关的转矩预控制有效（结合转速预控制） <p>文档： 功能手册之扩展功能分册；章节“补偿（K3）”</p>
状态“回参考点”	<p>回参考点的状态显示（轴）：</p> <ul style="list-style-type: none"> ● 有效测量系统无需回参考点 （有效测量系统无需回参考点） ● 有效测量系统回参考点 ● 有效测量系统需要回参考点 （有效测量系统需要回参考点） <p>该显示取决于机床数据中的设置：</p> <ul style="list-style-type: none"> ● MD34110 \$MA_REFP_CYCLE_NR ● MD20700 \$MC_REFP_NC_START_LOCK <p>显示和 NC/PLC 接口信号相对应： DB31, ... DBX60.4 和 60.5（已回参考点/已同步 1 或 2）</p> <p>文档： 功能手册之基本功能分册；章节“回参考点(R1)”</p>

显示数据	含义
QEC 状态	<p>显示轴的象限误差补偿 (QEC) 方法是否有效，如果有效哪些方法有效：</p> <ul style="list-style-type: none"> ● 当前无效 ● 学习系统 QEC 有效 ● 常规 QEC 有效 ● 常规 带补偿值调整的 QEC 有效 (带补偿值调整的常规 QEC 有效) ● 系统 QEC 有效 ● 系统 带测量时间调整的 QEC 有效 (带测量时间调整的系统 QEC 有效) ● 系统 带补偿值衰变时间调整的 QEC 有效 (带补偿值衰变时间调整的系统 QEC 有效) ● 系统 带补偿值测量时间+衰变时间调整的 QEC 有效 (带补偿值测量时间和衰变时间调整的系统 QEC 有效) <p>文档： 功能手册之扩展功能分册；章节“补偿 (K3)”</p>
状态“运行至固定挡块”	<p>显示轴在激活功能“运行至固定挡块”时是否满足“到达固定挡块”的条件 (NST DB31, ... DBX62.5)：</p> <ul style="list-style-type: none"> ● 正常控制 (功能“运行至固定挡块”未激活) ● 到达固定挡块 ● 失败 <p>文档： 功能手册之基本功能分册；运行至固定挡块 (F1)</p>
转矩限制值	<p>给出“运行至固定挡块”时通过 FXST[x] 或 SD43510 \$SA_FIXED_STOP_TORQUE 编程的值以及通过 MD37010 \$MA_FIXED_STOP_TORQUE_DEF 定义的夹紧转矩值。</p> <p>单位：最大转矩的 %</p> <p>文档： 功能手册之基本功能分册；运行至固定挡块 (F1)</p>

10.2.3 驱动信息

显示驱动数据和电机数据

在窗口“驱动信息”中显示电机和驱动模块状态的重要信息，例如：电机温度和直流母线电压。

步骤



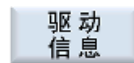
1. 选择操作区域“诊断”。



2. 按下菜单扩展键。
将切换到一个新的软键栏。



3. 按下软键“轴诊断”。
窗口“服务概览”打开。



4. 按下软键“驱动信息”。
窗口“驱动信息”打开。



5. 按下软键“驱动+”或“驱动-”，向前或向后选择驱动。



-或者-



按下软键“驱动选择”。
“驱动直接选择”窗口打开。
从下拉菜单中直接选择所需驱动。



6. 按下软键“确认”，确认选择。
显示驱动数据。

概述

下文将介绍窗口“驱动信息”内显示的各个状态、报警和消息等内容。

PLC 脉冲使能

窗口中显示的 PLC 是否给出了驱动所需的脉冲使能取决于接口信号：
DB380x, ... DBX4001.7 “脉冲使能” 的状态。

状态	含义	显示
1	PLC 给出了该驱动需要的脉冲使能。	是
0	PLC 封锁了该驱动需要的脉冲使能。	否

NC 转速控制器使能

窗口中显示的 NC 是否给出了驱动所需的转速控制器使能取决于接口信号：
DB390x ... DBX1.6 “转速控制器激活” 的状态。

状态	含义	显示
1	NC 给出了转速控制器使能。	是
0	NC 未给出转速控制器使能。	否

快速停机斜坡函数发生器

窗口中显示的快速停机斜坡函数发生器的状态取决于接口信号：
DB3900 ... DBX4000.1 “斜坡函数发生器封锁激活” 的状态。

状态	含义	显示
1	快速停机斜坡函数发生器激活。 驱动因此不利用下降斜坡制动，而是利用转速设定值 0 制动，并且不封锁脉冲。	是
0	快速停机斜坡函数发生器未激活。	否

脉冲已使能

窗口中显示的关于驱动是否获得了脉冲使能的消息取决于接口信号：
DB390x, ... DBX4001.7 “脉冲已使能” 的状态。

状态	含义	显示
1	驱动模块所需的脉冲已经使能。 进给轴/主轴可以移动。	是
0	驱动模块所需的脉冲已被封锁。 进给轴/主轴无法移动。	否

驱动就绪

窗口中显示的所选驱动力的当前状态取决于接口信号：
DB390x, ... DBX4001.5“驱动就绪”的状态。

状态	含义	显示
1	驱动已经准备好运行。	是
0	驱动没有准备好运行。	否

启动阶段

窗口中显示的所选驱动力的当前启动状态取决于驱动参数：
r0002“驱动运行显示”。

出错生命符号的数量

显示利用硬件检测出的 NC 和驱动之间的通讯故障。

说明

如果窗口中的显示不为 0，请联系当地的西门子办事处！

直流母线 1 驱动报警消息

显示是否有 1 类状态的消息。1 类状态的消息具有以下特性的报警：

- 报警会引发内部响应，比如：再生式制动、立即封锁脉冲等。
- 报警是无需应答的。

此类报警是总报警。关于实际可能出现的驱动报警请查看窗口“驱动系统诊断”，见驱动信息 (页 365)的说明。

经过滤波的直流母线电压

窗口中显示的所选驱动经过滤波的直流母线电压取决于驱动参数：
r0026“经过滤波的直流母线电压”。

单位： V

10.2 显示维修概览

经过滤波的转速设定值

窗口中显示的经过滤波的转速设定值（在针对转速控制器比例元件的设定值限幅后测出）取决于驱动参数：
r1438 “转速控制器转速设定值”。

单位：rpm

转速实际值

窗口中显示的经过滤波的实际电机转速取决于驱动参数：
r0021 “经过滤波的转速实际值”。

单位：rpm

经过滤波的电流实际值

窗口中显示的经过滤波的电流实际值取决于驱动参数：
r0078[1]“转矩电流实际值”。

单位：A

电机温度

窗口中显示的实际电机温度取决于驱动参数：
r0035 “电机温度”。

单位：°C

积分元件封锁

窗口中显示的关于转速控制器的积分元件是否激活的状态取决于接口信号：
DB390x, ... DBX4001.6“转速控制器积分元件已封锁”的状态。

状态	含义	显示
1	驱动中转速控制器的积分元件已被关闭，因此转速控制器从比例积分控制器转换成比例控制器。	是
0	转速控制器中的积分元件已使能，位置控制器作为比例积分控制器工作。	否

驻留轴

显示是否有被驻留的进给轴或主轴。

说明

进给轴/主轴一旦被驻留后，所有和编码器相关的监控和计算功能都将关闭。因此可以卸下编码器，而不会触发任何报警。

设定驱动数据组

窗口中显示的 PLC 应激活 8 个驱动数据组中的哪个数据组取决于接口信号：
DB380x, ... DBX4001.0 到 4001.2“驱动数据组选择 ABC”，缺省值是 DDS0。

实际驱动数据组

窗口中显示的 8 个驱动数据组中当前激活的驱动数据组取决于接口信号：
DB390x, ... DBX4001.0 到 4001.2“激活的驱动数据组 ABC”，缺省值是 DDS0。

设定电机数据组

窗口中显示的 PLC 应激活哪个电机数据组取决于接口信号：
DB380x, ... DBX4001.3 到 4001.4“电机选择 A, B”，显示值范围为：MDS0...3。

具体可以分为：

电机数据组	编码	
MDS0	0	0
MDS1	0	1
MDS2	1	0
MDS3	1	1

实际电机数据组

窗口中显示的当前激活的电机数据组取决于接口信号：
DB390x, ... DBX4001.3 到 4001.4 “激活的电机数据组 AB”，显示值范围为：MDS0...3。

运行方式

窗口中显示的驱动的控制方式取决于驱动参数：
p1300[0...n] “开环控制/闭环控制方式”。

不同的下标 n 中显示值有所不同：

n	显示
20	无编码器的转速控制
21	带编码器的转速控制
23	带编码器的转矩控制

测量系统 1/2 提供的位置实际值

测量系统 1/2 测出的轴的实际位置。该位置在机床坐标系中显示，没有考虑零点偏移和刀具补偿。

单位：毫米，英寸或者度

散热器温度

窗口中显示的关于散热器温度是否正常的消息取决于接口信号：
DB3390x, ... DBX4002.1 “散热器温度预警” 的状态。

状态	含义	显示
1	驱动向 PLC 发出散热器温度预警。	过热
0	驱动模块的散热器温度监控没有发出响应。	OK

电机温度

窗口中显示的关于电机温度是否正常的消息取决于接口信号：
DB390x, ... DBX4002.0 “电机温度预警” 的状态。

状态	含义	显示
1	电机温度超出了驱动中定义的报警阈值。	过热
0	电机温度低于报警阈值。	OK

加速过程结束

窗口中显示的驱动是否结束加速取决于接口信号：
DB390x, ... DBX4002.2“加速过程结束” 的状态。

状态	含义	显示
1	在给出新的转速设定值后，转速实际值达到了 p2164 定义的转速公差带，并且在 p2166 时间期满后仍在公差带内。	是
0	修改转速设定值后，驱动仍进行加速。	否

低于转矩阈值

窗口中显示的驱动是否低于转矩阈值取决于接口信号：
DB390x, ... DBX4002.3 " $|M_d| < M_{dx}$ " 的状态。

状态	含义	显示
1	当前转矩利用率低于 p2194 中设置的转矩利用率阈值。驱动向 PLC 报告，转矩设定值 $ M_d $ 没有超过转矩阈值 M_{dx} 。	是
0	转矩设定值 $ M_d $ 超过了转矩阈值 M_{dx} 。借助该信号可判断电机是否过载。	否

低于最小转速

窗口中显示的驱动是否低于最小转速取决于接口信号：
DB390x, ... DBX4002.4 " $|n_{实际}| < n_{最小}$ " 的状态。

状态	含义	显示
1	转速实际值 $ n_{实际} $ 低于 p2161 中设置的最小转速 $n_{最小}$ 。	是
0	转速实际值超过了最小转速。	否

低于转速阈值

窗口中显示的驱动是否低于转速阈值取决于接口信号：
DB390x, ... DBX4002.5 " $|n_{实际}| < n_x$ " 的状态。

状态	含义	显示
1	转速实际值 $ n_{实际} $ 低于 p2155 中设置的最小转速 n_x 。	是
0	转速实际值超过了设置的转速阈值。	否

转速实际值 = 转速设定值

窗口中显示的驱动转速实际值是否等于转速设定值取决于接口信号：
DB390x, ... DBX4002.6 " $|n_{\text{实际}}| < n_{\text{设定}}$ " 的状态。

状态	含义	显示
1	转速实际值与转速设定值之差在 p2163 设置的公差内。	是
0	转速实际值与转速设定值之差在设置的公差外。	否

出现报警时开展诊断

在出现报警时这些信息也可以为诊断提供便利：

- 驱动故障信息：
 - ⇒ 直流母线 1 驱动报警已设置。
 - ⇒ 检查设定驱动数据组、设定电机数据组和直流母母线电压。
- 报警 25040“静态监控”
报警 25050“轮廓监控”
报警 25060“转速设定值限制”
报警 25080“定位监控”
 - ⇒有可能是因为缺少驱动使能（不存在 PLC 脉冲使能）；从而导致屏幕上显示“脉冲已使能 = 否”。
- 电机温度过热
 - ⇒ 检查实际电机温度。

有关报警和排除故障原因的信息请参见在线帮助。

参考文档

有关接口信号的进一步信息请参见：

NC 变量和接口信号手册

关于驱动参数的更多信息参见：

系统变量手册

有关驱动和电机数据组的进一步信息请参见：

基本功能的功能手册，章节“A2: 不同的 NC/PLC 接口信号与功能” > “电机/驱动数据组的切换”>“接口概述”

10.3 PROFINET 诊断

10.3.1 显示 PROFINET 配置

在配置时或出现故障时，系统会显示 PROFINET 的状态，以便进行诊断。该诊断窗口只用于提供信息，而不能作任何修改。

PROFINET 接口

- PN1
- PN2

PPU 类型 PPU 24x.3 BASIC 只有一个 PROFINET 接口 (PN)。

说明

PROFIBUS DP 接口

SINUMERIK 828D 只有一个集成的 PROFIBUS DP。NC 通过该集成的 PROFIBUS DP 与驱动器进行通讯。PPU 上不能从外部进行访问，所以无法继续写入。

显示	含义/信息
状态	
总线状态	POWER_ON (0): 控制系统开机后的状态。 OFFLINE (1): 已进行初始化。 STOP (2): 根据硬件配置(SDB)启动。 CLEAR (3): PROFINET 设备已根据硬件设计(SDB)设定参数和配置，并以零输出数据参与到循环数据交换中。 OPERATE (4): 与 PROFINET 设备进行循环数据交换。 ERROR (9): 检测到严重故障 (例如: 无效或错误的 SDB)。 (<n>) 与总线状态的 BTSS 变量值相同。
总线配置	
S7 子网 ID	PROFINET 子网的 S7 子网 ID
波特率	传输速度, 单位 MBd
周期时间	配置的总线循环时间, 单位 ms; 它同时定义了位置控制器周期
同步分量 (TDX)	配置的一个 PROFINET 循环内周期性数据交换的时间段, 单位 ms。

显示	含义/信息
PROFIBUS 诊断/设备	
设备名称	PROFINET 设备的名称
设备编号	PROFINET 设备的编号
分配	PROFINET 设备是否分配给 NC 或 PLC 的信息。 NC 例如：一个或多个由 NC 控制的驱动 PLC：例如：I/O 外设或由 PLC 控制的轴 NC/PLC
通讯状态	通讯状态信息，是否能识别总线上的 PROFINET 设备。 ✔绿色：在 PROFINET 上已检测到 PROFINET 设备，并且和相应组件（NC 和/或 PLC）的数据交换正常。 ✘红色：故障/没有通讯
和 NC 的同步	信息：总线上的 PROFINET 设备是否和 NC 同步运行。 ✔绿色：PROFINET 上 PROFINET 设备和 NC 同步运行，即：等时的数据交换。 ✘红色：故障/没有通讯 ○灰色：PROFINET 设备没有分配给 NC，而是分配给 PLC。
子插槽数量	PROFINET 设备内设计的子插槽数量。

参考文档

关于 PROFINET 网络接口的更多信息参见下列参考文档：

- 设备手册 SINUMERIK 828D PPU
- 调试手册 SINUMERIK 828D CNC 调试

步骤



1. 选择操作区域“诊断”。



2. 按下菜单扩展按键和软键“TCP/IP 总线”。
“PROFINET/PROFIBUS 诊断...” 窗口打开。



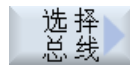


3. 按下软键“总线 -”或“总线 +”，选择所需总线。

...



-或者-



4. 按下软键“选择总线”，从列表中选择所需总线。



5. 按下软键“确认”。

10.3.2 显示 PROFINET 设备的详细信息

此处可以显示所选 PROFINET 设备子插槽的详细信息。

说明

PROFIBUS DP 接口

SINUMERIK 828D 只有一个集成的 PROFIBUS DP。NC 通过该集成的 PROFIBUS DP 与驱动器进行通讯。PPU 上不能从外部进行访问，所以无法继续写入。

显示	含义/信息
Device	
名称	PROFINET 设备的名称
序号	PROFINET 设备的编号
IP 地址	PROFINET 设备的 IP 地址
分配	PROFINET 设备是否分配给 NC 或 PLC 的信息。
通讯状态	PROFINET 连接至设备的当前通讯状态。 ✓绿色： 通讯激活。 ✗红色： 通讯未激活。
槽	
编号	子插槽编号
子插槽编号	设备内的子插槽编号。
I/O 地址	PLC I/O 地址范围内、分配给该子插槽的 I/O 地址。 对于 NC 轴，设定值和实际值必须始终定义相同的 I/O 地址。

显示	含义/信息
类型	指出子插槽是输入、输出还是诊断插口。 如果子插槽分配给了一个 NC 轴，则输出始终视为设定值，输入始终视为实际值。
长度（字节）	STEP7 I/O 地址范围内预留给孩子插槽的 I/O 长度。
通讯状态	子插槽的当前通讯状态 如不存在 NC 轴的值，则不进行显示。 ✔绿色： 通讯激活。 ✘红色： 通讯未激活。 ○灰色： 无 NC 轴
机床轴	机床数据中定义的该槽的名称 如槽未分配给 NC 轴，则不进行显示。
报文类型(NC)	显示分配的报文类型 如未在 NC 机床数据中分配报文类型，则不进行显示。

步骤



1. 选择操作区域“诊断”。



2. 按下菜单扩展按键和软键“TCP/IP 总线”。
“PROFINET/PROFIBUS 诊断...”窗口打开。



3. 按下软键“总线-”或“总线+”，选择所需总线。

...



4. 选择您需要了解详细信息的 PROFINET 设备。



5. 按下软键“详细”。
“PROFINET 诊断...详细信息”窗口打开。

参见

可访问节点 (页 47)

10.4 跟踪

“跟踪”（Trace）为您提供示波器功能，能够用形象的图形显示以下数据（信号）：

- NC 的通用数据
- PLC 数据

跟踪会话

该功能用于故障搜索和消除、机床效率分析以及过程效率分析。会在会话中记录事件发生前或发生后的数据（信号）。

图形显示

记录下来的数据以曲线图的形式显示。

您可以自定义记录类型。

保存和载入数据

可将记录下来的数据保存在一个 XML 文件中。保存下来的数据可以再次载入，以便之后开展分析。

记录的基础：信号（数据）

信号（数据）决定该记录的内容：

- NC 变量（轴数据、驱动数据和通道数据等）
- PLC 数据

周期性事件



插补周期 (IPO 周期)

PLC 周期 OB1



位置控制周期（伺服周期）

非周期性事件



启动会话

- 通过触发事件记录
- 手动启动和停止记录

分析结果

对记录结果进行分析时可以使用如下功能:

- 设置曲线图的比例
- 缩放曲线图
- 通过标识定义一段曲线

10.4.1 一般步骤

步骤

- 按以下步骤建立跟踪会话文件：
 - 选择变量，然后确定它们的显示方式
 - 调节图形窗口，以及保存模态
 - 设置触发事件的属性，确定记录开始或结束的时间
- 保存本次记录，以及设置的属性
- 分析结果，若需要进一步的细节，可以设置显示画面，例如：缩放，滚动，剪切等。

参见

用于跟踪的变量 (页 382)

跟踪设置 (页 388)

跟踪的选项 (页 390)

保存跟踪会话文件 (页 379)

分析跟踪 (页 391)

10.4.2 跟踪会话

10.4.2.1 创建跟踪会话文件

步骤



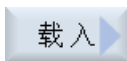
1. 选择操作区域“诊断”。



2. 按下菜单扩展键。



3. 按下软键“跟踪”。



4. 按下软键“载入跟踪”和“新建”。



窗口“新建会话”打开，其名称默认为“会话”。文件格式自动设为“.xml”。



5. 添加文件名或输入一个新文件名。

6. 选择一个变量类型并按下软键“确认”。

此时跳回窗口“选择用于跟踪的变量：...”。

在标题行会显示新建会话的名称。

参见


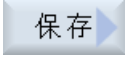



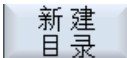
筛选/搜索变量 (页 382)

10.4.2.2 保存跟踪会话文件

可使用两种方法保存跟踪会话：

- 变量、设置：只保存变量和设置。
- 变量、设置和已记录的数值：会保存变量、设置以及所记录的数值。

步骤

- | | |
|---|---|
|  | 1. 已编辑了一个有效的跟踪会话。 |
|  | 2. 按下软键“保存跟踪”。
窗口“保存跟踪：会话...”打开，提供保存选项。 |
|  | 3. 激活相应的复选框后按下软键“确认”。
窗口“保存跟踪”打开，在“跟踪文件名称”栏中输入文件名。文件格式自动设为“.xml”，无法更改。 |
|  | 4. 按下软键“确认”，保存会话文件。
会话文件已经存在时，会跳出一条安全询问。 |
|  | 5. 希望覆盖已有会话文件时，再次按下软键“确认”。
-或者-
不希望覆盖已有会话文件时，按下软键“另存为”。
您可以采用建议的名称或自己命名。 |
|  | -或者-
若需要将会话文件保存至标准目录“跟踪”之外的其它目录，按下软键“更改目录”。
文件树形图打开，您可以在已配置的驱动器之间选择。 |
|  | 若需创建新的目录，按下软键“新建目录”。
窗口“新建目录”打开，你可以为新建目录命名。 |
|  | 按下软键“确认”。
窗口“保存跟踪”打开，显示目标目录的名称和会话文件。
可以更改会话文件的名称。 |
|  | 6. 再次按下软键“确认”，保存会话文件。
-或者-
具有该名称的会话文件已经存在时，会跳出一条安全询问。 |

10.4.2.3 加载跟踪会话文件

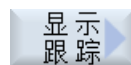
您可以加载已保存的会话文件，以重新启动此文件或者显示跟踪记录。

设置	含义
变量、设置	可以重新启动一个已创建的跟踪会话。
变量、设置和已记录的数值	可以查看跟踪会话的记录、修改变量以及在必要时重新启动。

前提条件

跟踪记录的会话文件已经存在。

步骤



1. 已选择了功能“跟踪”。
2. 按下软键“载入跟踪”。
窗口“载入会话文件：选择文件”打开。
3. 从相应目录中选择所需的会话文件（比如“跟踪”）后按下软键“显示跟踪”。
窗口“载入跟踪：会话...”打开。
4. 若需要同时启动跟踪会话，则选定复选框“变量、设置”。
5. 按下软键“确认”和“启动跟踪”，以重新启动记录。

-或者-

若需要显示跟踪会话，则选定复选框“变量、设置和已记录的数值”。
按下软键“确认”。
所选择会话的数据将载入并显示在窗口“跟踪”中。

10.4.3 用于跟踪的变量

10.4.3.1 筛选/搜索变量

选择变量

为了限制变量的选择，可以设置筛选和/或输入一个查找初始值。有以下筛选选项：

过滤器	
驱动参数	显示驱动参数
系统变量	显示系统变量
NC	显示 NC 信号、轴信号、通道信号以及运行方式组的信号 (BTSS 输入方式)
轴	
通道	
运行方式组	
伺服	显示伺服跟踪信号
PLC	显示 PLC 信号
PLC 报警	显示制造商专用 PLC 报警信号
所有 (无筛选)	显示所有信号，不筛选：NC 信号、轴信号、通道信号以及运行方式组信号 (BTSS 输入方式)。

说明

插入变量时的“过滤/搜索”操作

按照不同条件进行“过滤/搜索”时，初始值有较大差别。

例如，按下“过滤/搜索”，查找变量 \$R[0]：


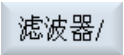
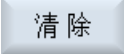



- 如果按照“系统变量”筛选，则初始值为 0。
- 如果按照“全部 (无过滤)”筛选，则初始值为 1，此时会用 BTSS 输入方式显示所有信号。

步骤

在开展一个跟踪会话前，首先在窗口“选择用于跟踪的变量：会话...”中选择需要记录其信号的变量。



1. 可在窗口“选择用于跟踪的变量：会话...”中显示跟踪会话。
2. 将光标移动到“变量”栏的一个空行中。

- 或者-
- 将光标移动到“变量”栏中的一个需要被替代的变量上。
-  3. 按下软键“插入变量”。
- “选择变量”窗口打开。
- 提示:**
- 需要选择一个驱动参数时，首先选择一个驱动对象（DO），然后再从列表中选择所需参数。
-  4. 按下软键“过滤/搜索”，开始查找特定对象。
- 在下拉菜单“过滤”中选择一个变量范围：
- 或者-
- 直接在输入框“搜索”中输入所需的变量。
- 光标会跳转到找到的目标上。
-  按下软键“删除搜索”，结束查找。
-  5. 按下软键“插入”，将所需变量插入到跟踪会话中。
- 窗口“变量属性”打开，可从下拉框中选择所需数值，例如：通道号、驱动总线号等。
- 或者-
-  按下软键“替换”，替换所选变量。
-  6. 按下软键“确认”，插入或替换变量。
- 每个变量都会自动获得一个参考号，它显示在窗口上方。这个编号自动分配。一个变量被删除时，它后面的变量自动重新编号。
7. 现在选择显示所需的设置，如颜色、线形、曲线图和显示等。

编辑跟踪会话中的变量

功能	含义
Ctrl + X 或者右击鼠标，弹出右键菜单“剪切”	按下快捷键或打开右键菜单，剪切变量。
Ctrl + C 或者右击鼠标，弹出右键菜单“复制”	按下快捷键或打开右键菜单，复制变量。
Ctrl + V 或者右击鼠标，弹出右键菜单“粘贴”	按下快捷键或打开右键菜单，粘贴变量。

10.4.3.2 选择变量的属性

属性

需要开展跟踪会话时，选择一个变量，并指定它的属性：

列	含义
变量	变量的地址

属性	含义
注释	<ul style="list-style-type: none"> ● 所选变量的名称 ● 指定变量的一般性说明。
颜色	曲线图的颜色选择 在创建变量时会自动从调色板中选择下一个颜色。

属性	含义
线形	曲线图的线形选择 <ul style="list-style-type: none"> ● 无线 ● 实线 ● 虚线 ● 点划线 ● 双点划线 ● 点线 ● 阶梯线 XY ● 阶梯线 XY ● 星线 ● + 线 ● X 线
显示	确定是否在窗口“跟踪”中显示变量的曲线图。不选中这个复选框时，不显示曲线图。
事件	选择触发信号记录的事件：
	<ul style="list-style-type: none"> ● 插补周期 (IPO 周期) ● 插补周期(IPO2) ● 位置控制周期（伺服周期） ● PLC 周期 OB1 ● 驱动跟踪速率 ● 几何轴启动/方向转换 ● 几何轴停止 ● 机床轴启动/方向转换 ● 机床轴停止 ● NC 启动（程序运行中） ● NC 停止（程序已运行） ● 数据记录启动

属性	含义
	<ul style="list-style-type: none"> ● 1 类程序段开始、程序段结束（所有程序级，无中间程序段） ● 2 类程序段开始、程序段结束（所有程序级，有中间程序段） ● 3 类程序段开始、程序段结束（所有主程序，无中间程序段） ● 程序段开始、程序段结束、程序段搜索（所有程序级，无中间程序段） ● 几何轴启动/切换方向（第 2 事件） ● 几何轴停止（第 2 事件） ● 2 类程序段开始（第 2 事件：所有程序级，无中间程序段） ● 2 类程序段结束（第 2 事件：所有程序级，有中间程序段） ● 1 类程序段开始，程序段预处理（第 2 事件） ● 1 类 OEM 测试事件（非周期式、程序段预处理） ● 2 类 OEM 测试事件（非周期式、程序段主处理）
	<ul style="list-style-type: none"> ● 激活/取消激活同步动作 ● 同步动作已被触发（满足条件） ● 报告的报警 ● 软键“取消”被按下 ● 程序级切换 ● 程序段结束（编译器） ● 跟踪结束（最后一个事件！） ● 跟踪启动
	<ul style="list-style-type: none"> ● WRTPR 零件程序指令 ● WRTPR 零件程序指令（程序段搜索） ● 换刀 ● 刀沿切换 ● 换刀（程序段搜索） ● 刀沿（程序段搜索） ● 启动触发事件已发生 ● 停止触发事件已发生 ● 程序段结束（程序段预处理） ● 程序段结束（第 2 事件：程序段预处理） ● WRTPR 零件程序指令（程序段预处理） ● 换刀（程序段预处理） ● 刀沿切换（程序段预处理）
事件通道	<p>此处可以确定 SINUMERIK 上特定事件所在的通道。对于通道通用的事件，此栏为空。</p>

属性	含义
位标记	如果是整数变量，可在此栏中指定位标记。位标记会首先和对应的信号值逻辑“与”运算，然后显示。设置标记后，所选择位将向右移动，所以看起来该位或者这些位似乎都是从零位开始的。 即：若除位 7 外，所有的位都设置了标记，则产生的整数数值只能为 0 或者 1，而不可能为 0 或者 128。 若除了位 7 和位 0 外所有的位都设置了标记，则所得整数结果为 0、1、2 或 3，而不会是 0、1、128 或 129。
小数位	指定轴名称上显示的小数位位数。
坐标轴	坐标轴显示在图形窗口的左边或右边或者根本不显示。
Y 轴显示	输入的数值或 0
比例系数	确定比例尺
单位	显示尺寸单位，例如：mm/min。尺寸由系统给定，无法更改。

10.4.3.3 显示变量的详细信息

可在详细信息窗口中直接看到关于图形的重要信息和设置，即：无须使用滚动条便可以移到“事件”列。

显示的信息

- 变量地址
- 含有变量描述的注释
- 事件
- 通道、轴、访问级别、运行方式组等
- 缩小的图形窗口，含有触发曲线图记录的事件及曲线图设置，如颜色、线形等

步骤



1. 您位于窗口“选择用于跟踪的变量：...”。
2. 将光标移动到所需变量。

10.4 跟踪

详细

3. 按下软键“详细信息”。
在窗口下方显示了局部画面，其中包含了最重要的信息以及图形。
再次按下软键“详细信息”，退出局部画面。

详细信息

10.4.4 跟踪设置

10.4.4.1 跟踪设置（PLC、NC、伺服）

在窗口“跟踪-设置（PLC、NC、伺服）”中确定跟踪会话的启停条件。

启动跟踪的条件	
通过软键启动跟踪	按下软键“启动跟踪”后，立即开始记录。
有变量时	会话由变量激活。通过软键“插入变量”或“快捷列表”选择变量。
<ul style="list-style-type: none"> 条件和值 	确定激活会话后启动跟踪记录的值。 提供以下值： =, ≠, >, >=, <, <=, 升序、降序、更改。
数据采集 [s]	
确定所记录数据的存储位置： <ul style="list-style-type: none"> 采用“NCK”时，这些数据会一直保存在 NC 工作存储器中，直到记录停止。然后会话文件才保存在 CF 卡上。 采用“CF 卡”或“硬盘”时，记录期间数据仍会保存到 CF 卡或硬盘（如 PCU）上。这样便可以处理更多的信号。 	
停止跟踪的条件	
通过软键停止跟踪	按下软键“停止跟踪”后，立即开始记录。
时间届满后	确定跟踪记录运行至结束的时间周期。
存储器存满时	达到规定的数量 [kByte] 后，跟踪会话结束。
有变量时	记录由变量停止。通过软键“插入变量”或“快捷列表”选择变量。
<ul style="list-style-type: none"> 条件和值 	确定会话激活后结束跟踪记录的数值。 提供以下值： =, ≠, >, >=, <, <=, 升序、降序、更改。
<ul style="list-style-type: none"> 后续运行时间 [s] 	确定会话结束后记录仍运行的时间长度。
<ul style="list-style-type: none"> 超出存储极限 [kByte] 	确定覆盖数据的最低存储极限（环形缓冲器）。

10.4.4.2 跟踪设置（驱动）

在窗口“跟踪-设置（驱动）”中确定跟踪会话的启停条件。

启动跟踪的条件		
通过软键启动跟踪	按下软键“启动跟踪”后，立即开始记录。	
有变量时	如果所选变量符合条件和给定的值，按下软键“启动跟踪”后，记录激活。 提供以下条件： <ul style="list-style-type: none"> • 上升沿 • 下降沿 • 位掩码 选择“位掩码”条件时，系统会显示软键“位掩码”，用于选择所需的位。可以输入一个用于启动跟踪记录的值（十进制），与通过位掩码可以设置的最大值（十六进制）无关。 示例： 设置了以下位时，跟踪记录启动：位 0 = 0；位 1 = 1；位 2 = 0；位 3 = 1	
	位掩码 = fH	条件
	变量：CU/ r722/3.3.1	位掩码 <input checked="" type="checkbox"/> 位 0 <input checked="" type="checkbox"/> 位 1 <input checked="" type="checkbox"/> 位 2 <input checked="" type="checkbox"/> 位 3
	值（十进制）	10
预触发时间 [s]：	输入预触发时间在功能上等同于参数 p4722“触发延时跟踪”。 <ul style="list-style-type: none"> • 如果预触发时间为正值，则在触发事件前启动记录。 • 如果预触发时间为负值，则在触发事件后延长记录。 输入预触发时间时须考虑整个记录时间： <ul style="list-style-type: none"> • 如果输入的预触发时间大于整个记录时间，系统会取消该输入。 • 如果输入的预触发时间过大，系统可能不会记录触发事件。 	
将数据保存到控制单元上		




驱动跟踪速率 [μs]:	通过软键“采样速率”输入一个系数，设置有效的采样速率。采样速率越高，记录时间也越长。
停止跟踪的条件	
时间届满后	确定记录运行至结束的时间周期。
存储器存满时	达到规定的数据量 [kByte] 后，记录结束。

10.4.4.3 跟踪的选项

在窗口“跟踪 - 选项”中您可以确定一个跟踪会话的属性。

- 调整图形窗口
 - 显示水平栅格线：确定是否显示水平栅格以及栅格的颜色。
 - 显示垂直栅格线：确定是否显示垂直栅格以及栅格的颜色。
 - 背景：选择图形窗口的背景色。
- 确定保存选项
 - 若需在覆盖跟踪会话前总是弹出安全询问，则激活复选框“确认文件覆盖”。若取消此复选框，则覆盖跟踪会话文件时不会有安全询问。
 - 激活复选框“导出数据到 csv 文件中”，除 xml 会话文件外，再将数据保存为 csv 文件。该文件仅用于导出数据，无法重新读入。

步骤

1.  您位于窗口“选择用于跟踪的变量：...”中，并选择了一个跟踪会话。
2.  按下软键“选项”。
“跟踪 - 选项”窗口打开。
3.  选择所需的设置并按下软键“确认”，采用该设置。

10.4.4.4 启动跟踪

在窗口“跟踪”中展示了一张曲线图，此图含有为跟踪会话选择的变量数据。




该图由两个直角坐标轴构成：


- 时间轴
即水平 X 轴，提供时间值。
- 数值轴
即垂直 Y 轴，提供精度（振幅）。

记录跟踪会话

- 手动启动和停止记录
- 使用触发事件启动和停止记录

步骤

- | | |
|---|---|
|  | 1. 已经将跟踪会话所需的变量整合在一起。 |
|  | 2. 按下软键“显示跟踪”。
“跟踪”窗口打开。记录被载入，载入进度会显示在一个进度窗口中。 |
|  | 3. 需要立即开始记录过程时，按下软键“启动跟踪”。

-或者-
通过特定触发条件启动记录。 |
|  | 4. 需要手动结束记录时，按下软键“停止跟踪”。

-或者-
通过特定触发条件停止记录。 |

10.4.5 分析跟踪

10.4.5.1 设置跟踪的视图

为了更加精确地分析记录下来的信号，您可以编辑曲线图的视图：

- 选择/图形
 - 显示图形：除了图形外，变量数据的显示还附加了注释、编号和颜色。
 - 选择：确定在图形窗口中显示哪些变量。
- 比例
 - 确定最大、最小 X/Y 值的比例系数

10.4 跟踪

- 缩放
 - 缩放视图
- 光标：光标视图能够精确测量一条或两条曲线。
 - 光标 A
 - 光标 B
 - 两个光标
 - 沿曲线：光标一直沿着曲线移动。
 - 点模式：光标直接跳转到不同值。
 - 峰值模式：光标从一个峰值跳转到下一个峰值。
 - 谷值模式：光标跳转到最低值。

前提条件

图形窗口“跟踪”已打开且跟踪记录已完成。

编辑曲线



可使用按键 <TAB> 切换至下一个变量。



使用 <SHIFT> + <TAB> 返回。



通过光标键可以设置标记，移动到某个值，并定义缩放区域。

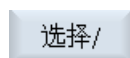


10.4.5.2 选择变量

您可以将通过复选框选中的变量显示为图形。

图形
变量的编号
轴名称
变量的注释

显示图形和选择变量



1. 在图形窗口“跟踪”中，一个记录打开。
2. 按下软键“图形/选择”。
图形窗口缩小一半。在窗口左侧中显示了变量（注释）和复选框。
3. 通过这些复选框勾选希望作为曲线显示的变量，或取消曲线显示。
4. 再次按下软键，退出曲线区域。

10.4.5.3 设置曲线图的比例

通过“比例”，您可以更加精确地测量 X 时间轴和 Y 数值轴上选中的变量。您可以任意缩放曲线图，并保存修改。

设置	含义	
X 轴最小值或 X 轴最大值	X 时间轴的最大值和最小值 在重新开始“跟踪”后，会使用这些值进行记录。	
	X 轴固定比例	保留 X 时间轴的值。
	穿孔带记录仪模式	在记录期间采用 X 时间轴的值。
Y 轴最小值或 Y 轴最大值	Y 数值轴（幅度）的最大值和最小值。	
偏移或 分辨率	确定幅度的有效值（平均值）。 比例按照系数 10 变化。	

调整时间轴和数值轴的比例

- | | |
|--|---|
| 









 | <ol style="list-style-type: none"> 1. 在图形窗口“跟踪”中，一个记录打开。 2. 按下软键“比例”，调节曲线图的比例。

输入窗口“选择 X-Y 比例系数”打开，显示 X 时间轴的值。 3. 需要隐藏该值时，按下软键“X 时间轴”。

-或者-
按下软键“曲线 Y 轴”显示 Y 数值轴的值。

需要隐藏该值时，再次按下软键。

-或者-
您选择了两根轴。 4. 按软键“比例 +”或者“比例 -”，直至达到所需的比例为止。

-或者-
直接在输入窗口中输入比例值，选择所需属性。 5. 按下软键“比例”，保存设置。


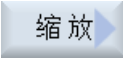

-或者-
按下软键“多图全窗口”，全屏显示所选的多个曲线图。

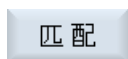
按下软键“单图全窗口”，全屏显示所选的曲线图。 |
|--|---|

10.4.5.4 缩放曲线图

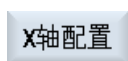
通过“缩放”，您可以查看所选曲线段的细节。确定了曲线段后，您便可以开始缩放。

步骤

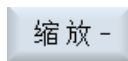
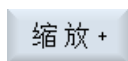
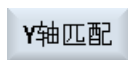
- | | |
|---|---|
| 

 | <ol style="list-style-type: none"> 1. 在图形窗口“跟踪”中，一个记录打开。 2. 若要放大或者缩小曲线图，按下软键“缩放”。 3. 选择软键“X 时间轴”或者“Y 数值轴”。 |
|---|---|



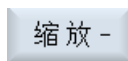
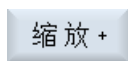
或



或



缩放区域



4. 若需要每个曲线图在图形窗口中依次单独显示，可按下软键“纵向排列”或“平铺显示”。

-或者-

若需调整图形窗口的所有曲线图，可使用软键“层叠显示”。由于现在所有的曲线占满了整个窗口，因此曲线以层叠的方式显示。

-或者-

若只需要使 X 时间轴占满图形窗口，可按软键“X 轴最大”。

-或者-

若只需要使 Y 数值轴占满图形窗口，可按软键“Y 轴等分”或“平铺显示”。

-或者-

若需使所有曲线图的数值轴占满图形窗口，可使用软键“Y 轴最大”。所有曲线都会占满层叠窗口，以层叠方式显示。

5. 按软键“缩放 +”或“缩放 -”，直至达到所需的显示大小为止。

按下软键“缩放区域”，确定特定区域。
显示一个边框。

按下软键“缩放 +”或者“缩放 -”，逐步调整大小。

按下软键“确认”，显示缩放区域。

10.4.5.5 定位光标

您可以借助光标获得定义的曲线段的精确数值。此时，光标直接定位在曲线上。

步骤

- | | |
|---|--|
| 












 | <ol style="list-style-type: none"> 1. 在图形窗口“跟踪”中，一个记录打开。 2. 按下软键“光标 A”。
在该软键上显示变量的参考号和颜色，以及坐标轴的颜色。 3. 将“(A)”移动到您需要开始测量的曲线位置上。
在提示条和状态条中会提示坐标位置。 4. 按下软键“光标 B”。
在该软键上也显示变量的参考号和颜色，以及坐标轴的颜色。
将“(B)”移动到您需要结束测量的曲线位置上。 5. 按下软键“两个光标”，显示 A 和 B 之间的范围。
状态栏中会显示 A 和 B 的坐标，以及两者的差值“Δ”。
再次按下已经激活的软键“光标 A”和“光标 B”，取消选中的位置。
再次选中时，位置再次复位。 <p>按下软键“截屏”，将“光标 A”和“光标 B”选中的区域占满整个图形窗口。</p> <p>按下软键“曲线模式”，使光标沿曲线逐步移动。</p> |
|---|--|

10.4.5.6 获取测量值

您可以直接在曲线上获取以下值：

- 峰峰值
- 峰谷值

步骤



在图形窗口“跟踪”中，一个记录打开。

按下软键“点模式”，使光标一直沿曲线移动。

按下软键“峰值模式”，使光标跳转到最高值。

按下软键“谷值模式”，使光标跳转到最低值。

按下软键“峰峰值模式”，只检查峰峰值。

按下软键“峰谷值模式”，只检查峰谷值。

10.5 驱动系统

10.5.1 显示驱动状态

状态显示

可在窗口“驱动系统诊断”中获得驱动设备及其所属驱动对象的状态信息。状态符号的含义如下：

显示图标		含义
	绿色	驱动对象处于循环运行中，没有识别出故障。
	黄色	驱动对象识别出略微严重的问题，例如：出现警告或缺少使能。
	红色	驱动对象识别出严重问题，例如：出现报警。
	灰色	不能确定此驱动对象的驱动状态。
#	特殊字符	读取数据时出错。

步骤



1. 选择操作区域“诊断”。

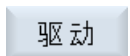


2. 按下菜单扩展键和软键“驱动系统”。

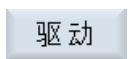


3. “驱动系统诊断”窗口打开。

在窗口标题中给出了所选驱动设备的名称。

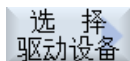


按下软键“驱动对象 +”或者“驱动对象 -”。



将选定下一 (+) 或者前一 (-) 驱动对象..

-或者-



按下软键“选择驱动设备”。

窗口“选择驱动设备”打开。

4. 通过选择栏选择需要的驱动设备，并按下软键“确认”，确认选择。

参见

显示驱动对象的详细信息 (页 398)

10.5.2 显示驱动对象的详细信息

驱动系统诊断 - 详细信息

显示驱动对象的下列信息：

- 运行显示(r0002)
 - 显示驱动对象的运行显示（参数 r0002）。
- 缺少使能 (r0046)
 - 除了没有使能信号的驱动对象，例如：控制单元，所有使能都被诊断。缺少的使能将显示在画面的子窗口中。
- 调试参数过滤器
 - (p0009) – 控制单元
显示控制单元参数“设备调试参数过滤器”(p0009)的数值。
 - (p0010) – 其他驱动对象
显示参数“驱动对象调试参数过滤器”(p0010)的数值。

- 当前故障





显示驱动对象的当前故障编号；如果驱动对象上不存在故障，则显示“当前无故障”。可在另一个窗口中概览待处理的故障，其中包含如下信息：

 - 报警号
 - 时间点：日期和时间
 - 报警文本
- 当前警告

显示驱动对象的当前警告号；如果驱动对象上不存在警告，则显示“当前无警告”。可在另一个窗口中概览待处理的报警，其中包含如下信息：

 - 报警号
 - 时间点：日期和时间
 - 报警文本

标记的含义

显示图标		含义
	绿色	驱动对象的相关值表明没有识别出故障。
	黄色	驱动对象的相关值表明识别出略微严重的问题，例如：出现警告或缺少使能。
	红色	驱动对象的相关值表明出现严重的问题，例如：出现报警。
	灰色	不能确定此驱动对象的驱动状态。
#	特殊字符	读取数据时出错。

步骤



1. 选择操作区域“诊断”。



2. 按下菜单扩展键和软键“驱动系统”。



“驱动系统诊断”窗口打开。

3. 选择您需要了解详细信息的驱动对象。



4. 按下软键“详细信息”。
“驱动系统诊断-详细信息”窗口打开。
在窗口标题中显示驱动对象的名称。



按下软键“故障”。



窗口“驱动设备故障”打开，显示一张已输出故障的清单。
按下软键“警告”。



窗口“驱动设备警告”打开，显示一张已输出警告的清单。
若您的控制系统中连接了 **Safety Integrated**，则可显示待处理的 **SI** 消息。

10.6 操作日志

10.6.1 设置操作日志

在窗口“操作日志设置”中可以确定，是否记录整个操作过程。
在此您可以选择记录哪些控制系统的动作，以便将来分析操作过程。
激活了操作日志后，您可以通过下面的文件查看系统数据日志：

- “actual_actionlog.com”，当前操作日志
- “actual_crashlog.com”，系统崩溃日志备份

可记录的数据

记录开	打开或关闭记录。
报警状态改变	记录报警的来、去信息。
按键操作	记录所有操作面板和外部键盘上的所有操作。
通道状态更改	NC/PLC 状态通过通道状态信息进行记录。在时间充裕时，通过这些状态至少可以部分验证机床操作面板上的操作。
窗口切换	打开和关闭窗口时记录格式名称和对话框名称（由编程人员命名的名称）。
写入 NC/PLC 数据	记录 NC 和 PLC 变量的写入过程。
文件存取	记录向 NC 进行的复制过程。

NC 中的功能调用（PI 服务记录特定的程序调用，例如 ASUP。服务）

当前的程序状态 为特定事件激活辅助信息。在出现需要使用 NC 停止、NC 启动或 NC 复位的重要报警时，会记录例如实际值和当前程序段。可以确定辅助信息要使用哪个通道或哪根主轴。

文件写间隔 文件备份的预设：

- “自动”：操作日志会在内部缓冲器中备份信息。缓冲器写满后将记录储存到 CF 卡上。控制系统关闭时记录可能会丢失。
- “在每个事件后”：直接储存所有的记录项，断电或其他情况时也不会丢失。
注意：因为 CF 卡只允许使用限制数量的写存取，因此不建议正常运行时使用该设置。
- “时间控制的”：按照特点的时间储存新的记录项。另外在输入栏中可以以秒为单位设定时间。

出现报警时将日志保存为 给出生成“Crashlog”的报警号。输入报警时用逗号隔开。
文件

创建系统崩溃日志文件

出现紧急事件时，例如急停报警，“actual_crashlog.com”相当于当前日志的一个备份文件。

因此文件不再处于操作日志的环形缓冲器内，并且也不会被新的记录项覆盖。

只有当出现其他紧急事件时，系统崩溃日志文件中的记录项才会被覆盖。

文件一开始为空文件，只有出现以下事件后才被写入：

- “出现报警时将日志保存为文件”栏中输入的报警出现。

步骤



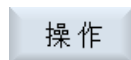
1. 选择操作区域“调试”。



2. 按下软键“HMI”、“诊断”和“操作日志”。



窗口“操作日志设置”打开。



10.6 操作日志

3. 选中复选框“记录开”，激活操作日志。
4. 如果希望操作日志中包含特定的事件，请选中对应的复选框。

参见

显示记录文件 (页 402)

10.6.2 显示记录文件

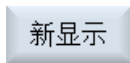
在窗口“操作日志 日志文件”中会显示所有在日志功能激活时可记录的数据。

日志文件

可以通过以下文件显示当前数据：

- "actual_actionlog.com"
- actual_crashlog.com

步骤



1. 选择操作区域“调试”。
2. 按下软键“系统数据”。
显示树形图。
在文件夹“HMI 数据”的路径：/日志/操作日志下可以查看保存的文件。
3. 选择所需的 com 文件，并按下软键“打开”或者双击此文件。
4. 按下软键“重新显示”，可以刷新日志（actual_actionlog.com）中的事件显示。
会显示上一次调用日志文件后的记录项。

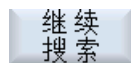
10.6.3 在记录文件中搜索

您可以在日志中根据需要查找事件。

前提条件

打开所需的日志文件。

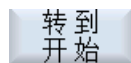
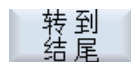
步骤



1. 按下软键“搜索”。
显示新的垂直软键栏。同时打开“搜索”窗口。
2. 在“文本”栏中输入搜索词。
3. 将光标定位至“方向”栏中并通过 <SELECT> 键选择搜索方向（向前、向后）。
4. 按下软键“确认”，开始进行搜索。
如果找到要搜索的文本，则标记出相应的行。
如果需要继续查找，按下“继续搜索”软键。

-或者-
如果需要中断搜索，则按下软键“取消”。

其他搜索方法



1. 按下软键“转到结尾”，可以比较长的日志中一直翻页至结尾。跳转到日志中最旧的条目。
2. 按下软键“转到开始”，跳转到日志中最新的条目。

10.6.4 保存记录

所显示的记录可保存在“hmi 数据/记录/报警记录”目录中。它可以作为二进制文件或 ASCII 文件加以保存。可以使用任意编辑器阅读 ASCII 文件。

可以生成以下日志：

- action.com（二进制文件）
- action.log（ASCII 文件）
- crash.com（二进制文件）
- crash.log（ASCII 文件）

前提条件

打开所需的日志文件。

步骤



1. 按下软键“保存日志”。
记录保存在“hmi 数据/记录/报警记录”目录中。

说明

软键“保存日志”在文件尚未保存时才能使用。

10.6.5 日志文件的结构

日志文件的内容

对于记录的每个事件，都会记录以下数据：

HMI 版本	NC 版本
系统	记录关键字 记录文本 详细信息
	日期/时间

条目	说明
HMI 和 NCK 版本	
系统:	
HMI	按键操作，窗口切换
NC	写入变量，PI 通讯
MSG	报警记录
USR	由机床制造商创建的用户记录
ERR	操作日志错误，例如未找到日志文件
记录关键字:	
HMI_START	“HMI 启动”记录

条目	说明	
HMI_EXIT	“HMI 关闭” 记录	
PLC_CRASH	“PLC 崩溃” 记录	
KEY_PRESSED	“键按下” 记录	
KEY_HOLD	“键按住” 记录	
KEY_RELEASED	“键松开” 记录	
ALARM_ON	“报警显示” 事件的记录	见下 “报警显示”
ALARM_OFF	“报警消失” 事件的记录	
ALARM_ACK	“报警应答” 事件的记录	
OPEN_WINDOW	“窗口打开” 的记录	
CHN_STATE_CHANGE D	“通道状态变更” 的记录	
OPMODE_CHANGED	“运行方式切换” 的记录	
TOOL_CHANGED	“换刀” 的记录	
OVERRIDE_CHANGED	“倍率变更” 的记录	
DOM_CMD	“下载至 NC” 的记录	
PI_CMD	“PI 通讯” 的记录	
WRITE_VAR	“写入 NCK/PLC 变量” 的记录	
AREA_CHANGED	“区域切换” 的记录	
NC_CONNECTION	“NC 连接建立/断开” 的记录	
USER	OEM 接口的用户记录	
ACTIVATED	操作日志已激活	
DEACTIVATED	操作日志已取消激活	
INTERNAL	内部操作日志记录	
日期/时间	事件的日期和时间 在 HMI_Start、PLC_Crash 或日期变更中从最后一条记录起会在关键字和日期之间标记一个“-”。	
记录文本	事件描述为明码文本	
详细信息	记录的更多详细信息，例如记录的 IPO 跟踪。	

示例

说明

接口信号 DB19.DBW24

示例中所用的接口信号 DB19.DBW24 仅适用 SINUMERIK 840D sl。适用 SINUMERIK 828D 的是 DB1900.DBW4。

```

NCK   WRITE_VAR                               18.02.2009  09:40:12
      ncul.local wrote Variable: DB19.DBW24 = 0
HMI   KEY_PRESSED                             18.02.2009  09:40:12
      Key up: Key up: "Return" (1000004/ 0)
HMI   HMI_START -----                      18.02.2009  09:34:15
      HMI started.
    
```

报警显示

```

MSG      ALARM                                日期/时间
      编号      删除标准      文本
      详细信息
    
```

条目	说明
MSG	
ALARM	报警类型: ALARM_ON, ALARM_OFF, ALARM_ACK
日期/时间	事件的日期和时间
编号	报警号
删除标准:	报警被应答时的设定

条目	说明
AUTOMATIC	自动应答
POWER-ON	通过 PowerOn 应答
RESET	通过 NCK 复位应答
NC-START	通过 NC 启动应答
NC-RESET	通过 NC 复位应答
ALARM-CANCEL	通过 Alarm-Cancel 键应答
RECALL	通过回调键应答
HMI	通过 HMI 应答
PLC	通过 PLC 应答
文本	显示英语报警文本
详细信息	记录的更多详细信息。

示例

```
MSG      ALARM_ON                20.02.2009   14:25:37
8020    POWER-ON   : Option 'activation of more than 1 channels' not
set
DETAILS:
Mode : JOG Program: canceled Channel: interrupted
Program-Level information:
Level Program running:                Invoc      偏移
1      /_N_MPF0                        0          0
```

10.6.6 扩展设置

操作日志的日志文件的大小在配置文件“actlog.ini”中定义。

步骤

1. 可从以下目录中复制模板配置文件“actlog.ini”： /siemens/sinumerik/hmi/template/cfg。
2. 在目录 /oem/sinumerik/hmi/cfg 或 /user/sinumerik/hmi/cfg 中创建和保存文件。
3. 打开文件，然后在“; ActionLogSize=”后输入文件的大小（单位：字节）。

配置文件“actlog.ini”

操作日志的模板配置文件

```

;
; 要激活设置时，删除
; 行首的注释符';'

[ActionLogSettings]
;使用以下设置修改内部操作日志的路径
;ActionLogPath=/user/sinumerik/hmi/ac_log/action.com
;使用以下设置修改内部操作日志的大小
;ActionLogSize=5000000
;使用以下设置修改内部崩溃日志的路径
;CrashLogPath=/user/sinumerik/hmi/ac_log/crash.com

```

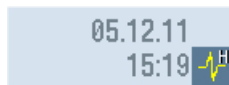
10.7 HMI 跟踪

创建 HMI 跟踪

“HMI 跟踪”功能用于进一步诊断操作软件。前提条件是您至少拥有访问级别 3（用户）。

创建 HMI 跟踪文件的步骤：

1. 从服务与支持中心的专家处获取文件 `sltrc_keybitset.ini`。将文件 `sltrc_keybitset.ini` 复制到 USB 存储器上，以便将文件载入控制系统或操作面板。
2. 选择操作区“调试”→“HMI”→“诊断”→“HMI 跟踪”。
3. 按下软键“载入配置”，将 `ini` 文件从 USB 存储器中载入系统。
4. 选中对应的复选框来启动跟踪功能，或者从操作界面的任意一个位置起利用以下快捷键启动跟踪功能：
 在操作面板上：<ALT> + <T>
 在外部键盘上：<ALT> + <Shift> + <T>
 屏幕标题栏的右上方会显示以下图标，表示正在进行跟踪。



5. 在操作面板上重复需要进行诊断的操作。

6. 撤销对应的复选框来停止跟踪功能，或者从操作界面的任意一个位置起利用以下快捷键停止跟踪功能：
在操作面板上：<CTRL> + <T>
在外部键盘上：<CTRL> + <Shift> + <T>
7. 现在生成一份输出文件 `sltrc.out`，按下软键“导出数据”将文件导出到 USB 存储器上。
8. 将文件 `sltrc.out` 发回给服务与支持中心进行后续分析。

说明

在结束分析后我们建议删除 `./user/sinumerik/hmi/log/sltrc` 下的所有文件。

说明

软键“高级”

该对话框中有更多选项，有助于专业的服务人员分析系统和评估 HMI 跟踪结果。

A.1 关于使用的第三方软件的说明

Copyright 1995 Sun Microsystems, Inc.

Printed in the United States of America.

All Rights Reserved.

This software product (LICENSED PRODUCT), implementing the Object Management Group's "Internet Inter-ORB Protocol", is protected by copyright and is distributed under the following license restricting its use. Portions of LICENSED PRODUCT may be protected by one or more U.S. or foreign patents, or pending applications.

LICENSED PRODUCT is made available for your use provided that you include this license and copyright notice on all media and documentation and the software program in which this product is incorporated in whole or part. You may copy, modify, distribute, or sublicense the LICENSED PRODUCT without charge as part of a product or software program developed by you, so long as you preserve the functionality of interoperating with the Object Management Group's "Internet Inter-ORB Protocol" version one. However, any uses other than the foregoing uses shall require the express written consent of Sun Microsystems, Inc.

The names of Sun Microsystems, Inc. and any of its subsidiaries or affiliates may not be used in advertising or publicity pertaining to distribution of the LICENSED PRODUCT as permitted herein. This license is effective until terminated by Sun for failure to comply with this license. Upon termination, you shall destroy or return all code and documentation for the LICENSED PRODUCT.

LICENSED PRODUCT IS PROVIDED AS IS WITH NO WARRANTIES OF ANY KIND INCLUDING THE WARRANTIES OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE.

LICENSED PRODUCT IS PROVIDED WITH NO SUPPORT AND WITHOUT ANY OBLIGATION ON THE PART OF SUN OR ANY OF ITS SUBSIDIARIES OR AFFILIATES TO ASSIST IN ITS USE, CORRECTION, MODIFICATION OR ENHANCEMENT. SUN OR ANY OF ITS SUBSIDIARIES OR AFFILIATES SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY LICENSED PRODUCT OR ANY PART THEREOF.

A.1 关于使用的第三方软件的说明

IN NO EVENT WILL SUN OR ANY OF ITS SUBSIDIARIES OR AFFILIATES BE LIABLE FOR ANY LOST REVENUE OR PROFITS OR OTHER SPECIAL, INDIRECT AND CONSEQUENTIAL DAMAGES, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

SunOS, SunSoft, Sun, Solaris, Sun Microsystems and the Sun logo are trademarks or registered trademarks of Sun Microsystems, Inc.

SunSoft, Inc.
2550 Garcia Avenue
Mountain View, California 94043

Copyright (c) 1991 by AT&T.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR AT&T MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

This product includes software developed by the University of California, Berkeley and its contributors.

QLocale's data is based on Common Locale Data Repository v1.6.1.

说明

关于使用的第三方软件的详细说明可以参见产品 DVD 光盘中的自述文件 `Readme_OSS`。

A.2 缩略语

缩写	含义
ASCII	American Standard Code for Information Interchange: 美国信息互换标准码
BAG	操作方式组
BTSS	操作面板接口
CNC	Computerized Numerical Control: 计算机数控系统
CEC	Cross Error Compensation: 垂度补偿
CP	Communication Processor
DCP	Discovery and basic Configuration Protocol: ProfiNet 协议
DHCP	Dynamic Host Configuration Protocol
DIN	Deutsche Industrie Norm
DNS	Domain Name System
DIR	Directory: 目录
DRAM	Dynamic Random Access Memory
EES	Execution from External Storage
ERTEC	Enhanced Real Time Ethernet Controller
ESR	扩展的停止和退回
ETC	ETC 键“>”；相同菜单中的扩展软键栏
FRAME	数据段（框架）
FIFO	First In - First Out: 数据进出存储器的方式。
GDIR	Global Directory: 全局零件程序存储器
GUD	Global User Data: 全局用户数据
HSA	主主轴驱动
HSC	High Speed Cutting: 高速加工
INC	Increment: 增量尺寸
INI	Initializing Data: 初始化数据
IP	Internet Protocol
IPO	插补器
ISO	International Standard Organization
JOG	Jogging: 点动模式

A.2 缩略语

缩写	含义
LED	Light Emitting Diode: 发光二极管
LLDP	Link Layer Discovery Protocol
LUD	Local User Data: 局部用户数据
MAC	Media Access Control
MB	Megabyte
MD	机床数据
MDA	Manual Data Automatic: 手动输入, 自动运行
MCS	机床坐标系
MLFB	Maschinenlesbare Fabrikatbezeichnung
MPF	Main Program File: NC 零件程序 (主程序)
MCP	Machine Control Panel: 机床控制面板
NC	Numerical Control: 数控系统
NCK	Numerical Control Kernel: 带有程序段处理、运行范围等等的数控内核
NCU	Numerical Control Unit: NCK 硬件单元
NFS	Network File System
NTP	Network Time Protocol
NV	零点偏移
OEM	Original Equipment Manufacturer
OID	Object Identifier: 信息标签
OP	Operator Panel
PCU	Programmable Control Unit
PI 服务	“程序-实例” 服务
PG	编程器
PLC	Programmable Logic Control
RFC	Request for Comments
REF	返回参考点功能
REPOS	再定位功能
ROV	Rapid Override: 快进补偿
RPA	R-Parameter Active: NCK 中用于 R 参数的存储器区域
SBL	Single Block: 单程序段
SD	设定数据

缩写	含义
SDB	系统数据块
SEA	Setting Data Active: 设定数据符号（文件类型）
SK	Softkey
SMB	Server Message Block
SNMP	Simple Network Management Protocol
SPF	Sub Program File: 子程序
SRAM	Static Random Access Memory: 静态存储器（缓存）
SSH	Secure Shell: 网络协议
SW	软件
SYF	System Files: 系统文件
TCU	Thin Client Unit
TEA	Testing Data Active: 机床数据标识
TMA	Tool Magazine Active: 刀库数据
TO	Tool Offset: 刀具补偿
TOA	Tool Offset Active: 刀具补偿符号（文件类型）
UDP	User Datagram Protocol: 网络协议
UFR	User Frame
VNC	Virtual Network Computing
VSA	进给驱动
WCS	工件坐标系
WZV	刀具管理

索引

3

3D 模拟, 67, 72

A

action.com, 403
action.log, 403
actlog.ini, 407
actual_actionlog.com, 400
actual_crashlog.com, 400

C

crash.com, 403
crash.log, 403
CUST_800.SPF, 86
CUST_832.SPF, 86, 186
CUST_CLAMP, 86
CUST_M6, 86, 88
CUST_MEACYC
 结构程序, 191
CUST_MEACYC.SPF, 86
CUST_MULTICHAN, 86, 93
CUST_T, 86, 88
CUST_TECHCYC.SPF, 86
 调整, 90
CYCLE495, 147
CYCLE63, 102
CYCLE79, 103
CYCLE800
 结构程序, 177
CYCLE832, 184
CYCLE84, 85
CYCLE840, 85
CYCLE930, 126
CYCLE950, 126
CYCLE951, 126
CYCLE952, 126
CYCLE99, 126
CYCLE996, 182

D

DHCP 服务器 ID, 46

E

Ethernet / PROFINET
 诊断, 46
EXTCALL
 关闭/打开, 33

G

G 代码程序, 332
G 代码组
 设置显示, 52
G 功能组 59, 184

H

HEAD_1, 165
HEAD_2, 166
HMI 跟踪, 408
HOLES2, 103

M

Manual Machine
 激活功能, 62
 设置, 62
MIXED_45, 171

N

NC/PLC 变量
 显示, 351
 修改, 353

P

PROFINET
 连接, 373
 设备, 375
PROG_EVENT.SPF, 87, 373

S**ShopMill**

- 刀具专用功能, 301
- 用于铣削的设置, 104

ShopTurn

- 设置副主轴, 121
- 用于车削的设置, 115

slmagcodeconfig.ini, 52

SLOT2, 103

T

TABLE_45, 168

TABLE_5, 173

TO 单元

- 分配, 293

TRAANG, 142

- 在车削中, 139

TRACON, 142

TRACYL, 143

- 在车削中, 132

TRANSMIT

- 带 Y 轴, 137
- 在车削中, 136

TRANSMIT 指令, 332

编

编程 G 代码程序, 66

编辑刀具和刀库管理, 340

变

变量屏幕, 355

标

标记

- OEM 刀具参数, 313
- OEM 刀沿参数, 314
- OEM 多刀参数, 314
- OEM 监控参数, 314
- 刀具参数, 264
- 刀具专用功能, 315
- 刀库位置参数, 271
- 刀库位置适配器参数, 272
- 刀沿参数, 267
- 多刀参数, 273

多刀刀位参数, 275

监控参数, 269

冷却液, 315

磨削参数, 270

标志

OEM 刀库位置参数, 315

操

操作区域“加工”, 51

操作日志

actlog.ini, 407

保存日志文件, 404

打开日志文件, 402

日志文件, 404

设置, 400

设置日志大小, 407

搜索事件, 402

测

测量工件

在车削中, 218

在铣削中, 195, 209

测头

测量刀具, 186

测量工件, 186

检查程序示例, 188

检查功能, 187

查

查找方式

配置, 60

车

车削

ShopTurn 中的设置, 115

X 轴的零点偏移, 143

测量工件, 218

刀具测量, 202

端面加工, 136

工艺循环, 126

示例, 设置车床, 131

显示软键, 79

斜置 Y 轴, 139

柱面转换, 132

车削轮廓, 126

成

成型磨削, 147

程

程序测试

 激活, 58

 取消, 58

程序列表

 创建, 36

 设计软键, 36

创

创建系统崩溃日志文件, 401

刀

刀架, 331

刀具表

 列表视图, 259

刀具参数

 OEM 标记, 313

 标记, 264

刀具测量

 在车削中, 202

 在铣削中, 197

刀具管理

 MD 设置, 243

 PLC, TRANSLINE 2000, 252

 编辑管理, 341

 车削工艺, 252

 打开功能, 341

 带刀库管理, 244

 刀具单元, 344

 刀具磨损, 250

 刀库配置, 347

 分配, 348

 缓存位置, 345

 磨削工艺, 252

 使能功能, 247

 手动刀具, 250

 图形化的刀具显示和刀库显示, 249

 无刀库管理, 243

 铣削工艺, 252

 直径/刀沿半径的机床数据设置, 247

 装载刀位, 346

刀具类型

 车刀, 282

 磨削刀具, 281

 特种刀具, 282

 铣刀, 280

 钻头, 281

刀具专用功能

 标记, 315

刀库位置参数

 OEM 标记, 315

 标记, 271

刀库位置适配器参数

 标记, 272

刀套

 分配, 293

刀沿参数

 OEM 标记, 314

 标记, 267

调

调整 JOG 运行方式下的测量, 192

调整刀具测量

 车削, 219

 铣削时, 211

 铣削示例, 213

调整工件测量, 206

端

端面加工

 在车削中, 136

多

多边形, 103

多刀

 换刀, 337

 装载和卸载, 335

多刀参数

 OEM 标记, 314

 标记, 273

多刀刀位参数

 标记, 275

多个不同工件的加工（选件），112

访

访问级别, 11

访问权限

- 单独, 13
- 全局, 14
- 生效, 14
- 远程访问, 14

服

- 服务概览, 355
- 轴和驱动, 358

高

- 高速设定, 184

跟

跟踪

- 保存会话, 380
- 保存选项, 390
- 编辑比例, 394
- 编辑会话, 384
- 编辑图形视图, 391
- 变量属性, 384
- 步骤, 378
- 插入变量, 383
- 创建会话, 379
- 定位光标 A/B, 396
- 定义缩放区域, 395
- 过滤/搜索, 382
- 会话, 382
- 将数据保存到控制单元上, 389
- 启动会话, 381
- 启动记录, 391
- 设置 (PLC、NC、伺服), 388
- 设置 (驱动), 389
- 设置栅格线, 390
- 数据采集, 388
- 缩放曲线图, 394
- 替换变量, 383
- 停止记录, 391
- 显示变量的详细信息, 387
- 显示峰峰值, 396
- 显示峰谷值, 396
- 选择变量, 382, 393
- 一览, 377
- 载入会话, 381

更

- 更快速的程序段搜索, 61

工

工艺

- 车削, 126
- 回转, 147
- 平面磨削, 146
- 外圆磨削, 146
- 铣削, 102
- 钻削, 84

功

功能

- 替换, 327

攻

- 攻丝, 85

拐

- 拐角车削, 126

回

回转

- 回退, 164
- 回转轴, 160
- 激活, 148
- 配置输入窗口, 148
- 偏移矢量, 158
- 运动链调试, 156

机

- 机床功能, 定义文本, 303
- 机床数据
 - 配置明码文本, 29

基

- 基于子站的网络诊断, 42

夹

- 夹具
 - 功能设定, 74

监

- 监控参数
 - OEM 标记, 314
 - 标记, 269

键

- 键盘
 - 设置, 22

截

- 截屏
 - 创建, 23
 - 复制, 24

口

- 口令
 - 删除, 12
 - 设置, 12
 - 修改, 12

冷

- 冷却液
 - 标记, 315
 - 分配, 301

列

- 列表参数
 - 新创建, 278
 - 修改, 276

零

- 零点偏移
 - 在车削中, 143

轮

- 轮廓槽式车削, 126
- 轮廓铣削, 102

螺

- 螺纹切削, 126

模

- 模拟
 - 车削工艺, 69
 - 功能的基本条件, 68
 - 磨削工艺, 70
 - 取消, 71
 - 铣削工艺, 69
- 模拟主轴, 229

磨

- 磨削参数
 - 标记, 270

配

- 配置换刀原因, 305

平

- 平面磨削, 146
 - 显示软键, 79

切

- 切端面齿, 150

驱

- 驱动器
 - 从外部存储器执行, 34
 - 逻辑驱动器, 33
 - 全局零件程序存储器, 35
 - 设置, 33
 - 设置 EXTCALL, 34

驱动系统

- 诊断, 397
- 诊断 - 详细信息, 398

驱动信息

- 参数描述, 365
- 显示, 365

日

日期

- 格式选择, 19
- 设置, 21

日志文件

- 操作日志, 404

软

软键

- 访问等级, 18
- 恢复设置, 18

设

设置带铣刀的车床, 130

设置副主轴

- ShopTurn, 121

时

时间

- 格式选择, 20
- 设置, 21

实

实际值窗口

- 设置字体大小, 51

示

示教, 59

替

替换商标, 51

添

添加引导启动画面, 23

通

通道运行信息, 54

- 标准设计, 54

同

同步记录, 72

- 激活处理时间, 73
- 选件, 72

拓

拓扑结构, 47

外

外圆磨削, 146

- 显示软键, 79

网

网络

- 配置, 39

网络接口

- 配置, 43

网络诊断, 42

尾

尾座

- 选件, 72

铣

铣削

ShopMill 中的设置, 104

测量工件, 195, 209

刀具测量, 197

示例, 铣床设置, 108

显示软键, 79

柱面转换, 106

斜

斜置 Y 轴
在车削中, 139

旋

旋转方向
副主轴, 122
主轴, 116

循

循环版本
显示, 221
指定, 221
循环报警, 154

用

用户视图
编辑, 27
创建, 26
导入, 25
删除, 29

语

语言
确定选择模式, 21

圆

圆弧槽, 103
圆位置模式, 103

运

运动
测量, 182
带切端面齿, 150
调试示例, 165
检查列表, 155
运行状态, 355

诊

诊断
Ethernet / PROFINET, 46
HMI 跟踪, 408

轴

轴诊断, 359

主

主轴控制, 227
主轴诊断, 231
电机温度传感器, 235
附加温度传感器, 235
夹紧时间统计, 241
夹紧状态, 237
温度, 233
温度柱形图, 236
物流数据, 241
诊断统计, 240
主轴负载率, 236
转速限值, 239

柱

柱面转换
没有槽壁补偿, 133
铣削, 示例, 108
有槽壁补偿, 134
在车削中, 132
在铣削中, 106

转

转换
用于铣削的设置, 109

状

状态显示
插入商标, 51

钻

钻削, 84

坐

坐标系

MD52000, 80

SINUMERIK

SINUMERIK 840D sl/828D Easy XML

编程手册




适用于:

数控软件 V4.7 SP1

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。

目录

1	创建用户对话框.....	5
1.1	功能范围.....	5
1.2	配置基础.....	7
1.3	设计文件.....	10
1.4	设计文件的结构.....	14
1.5	语言相关性.....	16
1.6	XML 标签.....	16
1.6.1	通用结构.....	16
1.6.2	指令/标签说明.....	17
1.6.3	颜色代码.....	47
1.6.4	专用 XML 句法.....	47
1.6.5	运算符.....	48
1.6.6	创建软键菜单和对话框窗口.....	49
1.7	创建用户菜单.....	81
1.7.1	创建加工循环窗口.....	81
1.7.2	替代符号.....	85
1.8	元素定址.....	85
1.8.1	PLC 定址.....	85
1.8.2	NC 变量定址.....	87
1.8.3	在运行时生成 NC/PLC 地址.....	87
1.8.4	驱动组件的定址.....	87
1.8.5	机床数据和设定数据的定址.....	90
1.8.6	用户数据的定址.....	91
1.9	预定义功能.....	92
2	创建调试对话框.....	135
2.1	功能概述.....	135
2.2	PLC 用户程序中的配置.....	138
2.3	操作界面上的显示画面.....	140
2.4	创建和语言相关的文本.....	141
2.5	功率部件应用示例.....	142
2.6	脚本语言.....	143
2.6.1	CONTROL_RESET.....	146
2.6.2	FILE.....	146

2.6.3	OPTION_MD.....	147
2.6.4	PLC_INTERFACE.....	148
2.6.5	POWER_OFF.....	149
2.6.6	WAITING.....	149
2.6.7	用于窗口的 XML 标签.....	149
2.6.8	SOFTKEY_OK, SOFTKEY_CANCEL.....	150
	索引.....	153

创建用户对话框

1.1 功能范围

一览

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。

控制系统提供基于 XML 的脚本语言用于创建用户对话框。

该脚本语言可以在 HMI 上的操作区 <CUSTOM> 中显示机床专用菜单和对话框窗口。

所有对话框窗口的构成都与语言无关。这种情况下系统会从随附的语言数据库中读取要显示的文本。

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（登入）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
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

1.1 功能范围

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 16）。

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。

1.2 配置基础

配置文件

新操作界面的说明存储在配置文件中。这些文件自动编译并显示屏幕上的结果。在供货时并不提供配置文件，因此必须首先创建或下载此文件。

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

重要:

所有文件名称必须小写

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：

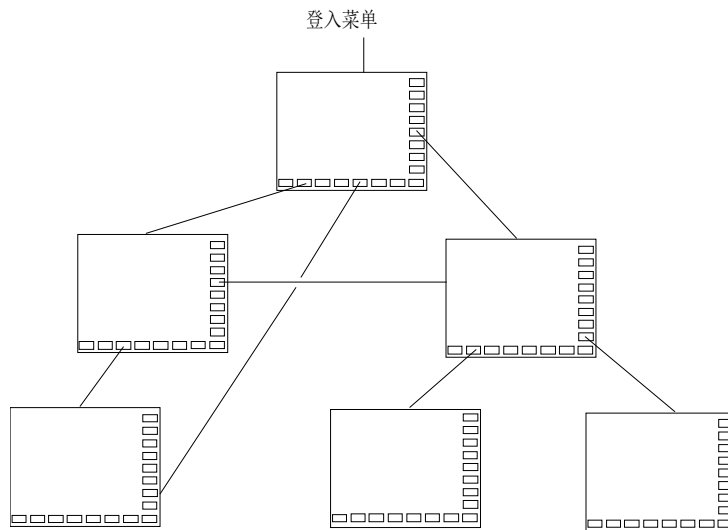


图 1-1 用户对话框菜单树

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。

1.2 配置基础

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，而使用它们又可以执行其他动作。

下图显示了控制系统上的制造商驱动器。

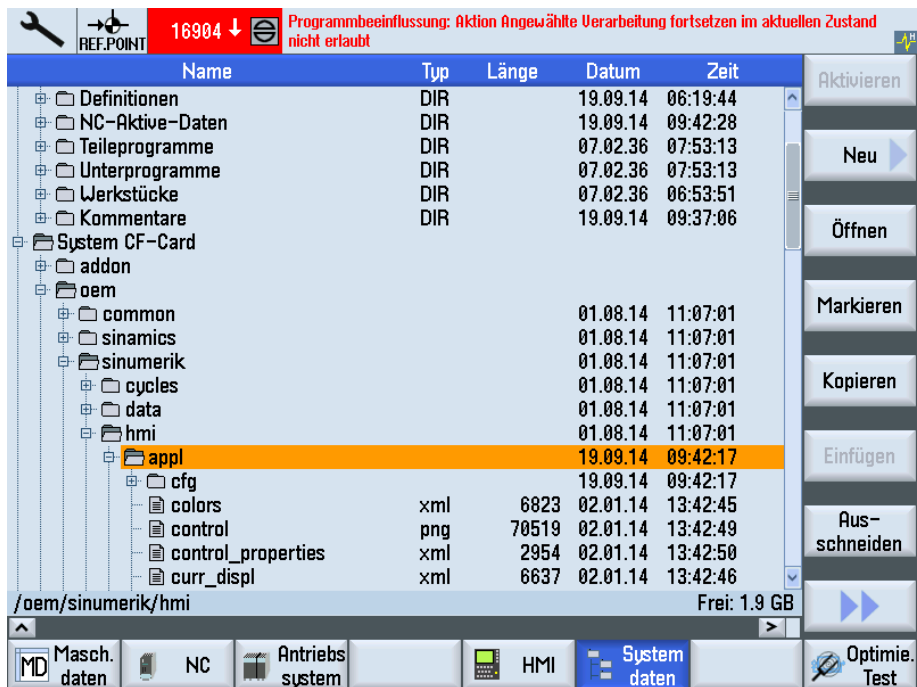


图 1-2 制造商驱动器

为了进行用户对话框的配置，需要使用控制系统中“制造商驱动器”目录下的以下文件：

表格 1-1 用于配置的文件

文件类型	文件名称	含义	HMI 操作区中的存储位置 操作区调试
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 <CUSTOMER> 中的对话框屏幕。	“Systemcard” > “oem/sinumerik/hmi” > 在用于应用的子目录“appl”中
文本文件	“oem_xml_screens_xxx.ts”	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。	“调试文件” > “制造商驱动器” > 在用于所需语言的子目录“lng”中

文件类型	文件名称	含义	HMI 操作区中的存储位置 操作区 调试
位图		控制系统支持 BMP 格式和 PNG 格式。	“调试文件” > “制造商驱动器” > 在子目录“ico”中
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。	“调试文件” > “制造商驱动器” > 在用于应用的子目录“appl”中

1.3 设计文件

前言

下图显示了控制系统上的制造商驱动器。

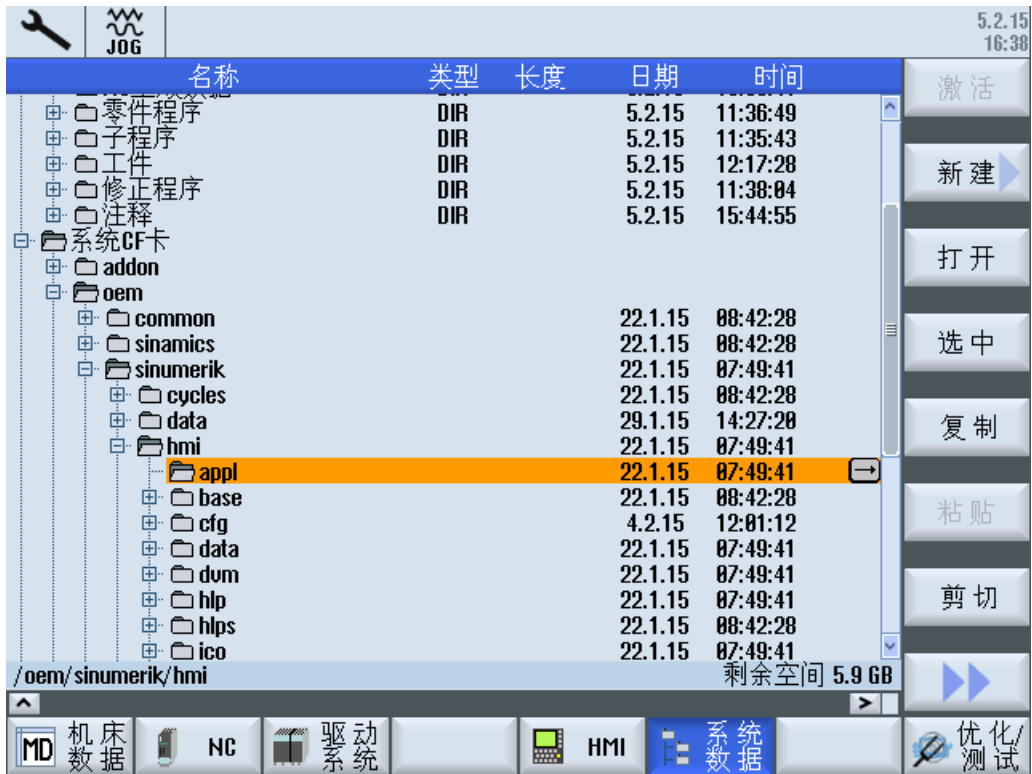


图 1-3 制造商驱动器

为了进行用户对话框的配置，需要使用控制系统中“制造商驱动器”目录下的以下文件：

表格 1-2 用于配置的文件

文件类型	文件名称	含义	HMI 操作区 <SYSTEM> (系统) 或 <PROGRAM MANAGER> (程序 管理器) 中的存储位 置
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 <COSTUMER> 中的对话框屏幕。	“调试文件” > “制造商驱动器” > 在用于应用的子目录“appl”中
文本文件	“oem_xml_screens_xxx.ts”	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。	“调试文件” > “制造商驱动器” > 在用于语言的子目录“lng”中
位图		控制系统支持 BMP 格式和 PNG 格式。	“制造商驱动器” > 在子目录“ico”中 位图保存在控制系统所属的屏幕分辨率的子目录中。 Bem.: 如给定了位图文件的路径，则可以将文件直接保存至该目录中。
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。	“制造商驱动器” > 在用于应用的子目录“appl”中

文件与用户对话框配置的相关性

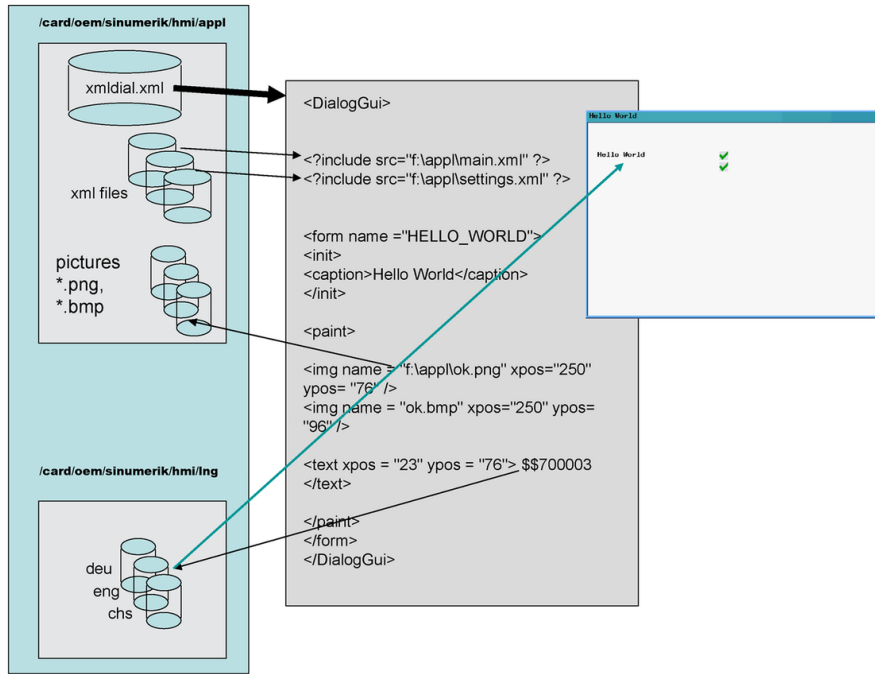


图 1-4 相关性

加载配置

与前面表格“用于配置的文件”的“HMI 中的存储位置”一系列的说明相同，所创建的文件被复制到菜单 > “制造商驱动器” 中相应的子目录下。

说明

一旦制造商驱动器子目录中有了用于应用的脚本文件“xmldial.xml”，用户就可以在操作区 `<CUSTOM>` 中启动用户对话框。

第一次复制后必须通过“正常上电启动”对控制系统进行复位。

HMI 上用户对话框示例

在调用操作区 <CUSTOM> 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。



图 1-5 操作区 <CUSTOM> 中用户对话框示例

参见

预定义功能 (页 92)

1.4 设计文件的结构

概览

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

下列图片说明了文件“xmldial.xml”的 XML 脚本以及相应的截屏。

脚本包含有用于显示实际值和剩余行程的对话框，以及一张 R 参数表。

```

<DialogGui>
<!--
main menu
It is called by the system software. It starts the application.
The menu tag manages the soft key reactions. One input form can be assigned to a menu tag.
-->

<menu name = "MAIN">
<OPEN_FORM name = "CURRENT_DISPLAY" />

<softkey POSITION="1">
<caption>R-Parameter</caption>
<navigation>MENU_R_PARAMETER</navigation> <!-- opens the menu R parameter -->
</softkey>

</menu>

<form name = "CURRENT_DISPLAY">

<init>

<caption>Istwerte</caption>

<control name = "label1" xpos = "36" ypos = "56" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[0]" />
<control name = "label2" xpos = "36" ypos = "70" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[1]" />
<control name = "label3" xpos = "36" ypos = "84" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[2]" />
<control name = "label4" xpos = "36" ypos = "98" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[3]" />
<control name = "label5" xpos = "36" ypos = "112" width = "32" fieldtype="readonly" refvar="nck/Channel/GeometricAxis/name[4]" />

<control name = "edit1" xpos = "80" ypos = "56" refvar="nck/Channel/GeometricAxis/actProgPos[0]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
<control name = "edit2" xpos = "80" ypos = "70" refvar="nck/Channel/GeometricAxis/actProgPos[1]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
<control name = "edit3" xpos = "80" ypos = "84" refvar="nck/Channel/GeometricAxis/actProgPos[2]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
<control name = "edit4" xpos = "80" ypos = "98" refvar="nck/Channel/GeometricAxis/actProgPos[3]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
<control name = "edit5" xpos = "80" ypos = "112" refvar="nck/Channel/GeometricAxis/actProgPos[4]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>

<control name = "edit11" xpos = "210" ypos = "56" refvar="nck/Channel/GeometricAxis/progDistToGo[0]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
<control name = "edit12" xpos = "210" ypos = "70" refvar="nck/Channel/GeometricAxis/progDistToGo[1]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
<control name = "edit13" xpos = "210" ypos = "84" refvar="nck/Channel/GeometricAxis/progDistToGo[2]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
<control name = "edit14" xpos = "210" ypos = "98" refvar="nck/Channel/GeometricAxis/progDistToGo[3]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>
<control name = "edit15" xpos = "210" ypos = "112" refvar="nck/Channel/GeometricAxis/progDistToGo[4]" hotlink="true" fieldtype="readonly" format="%9.3f" time="super fast"/>

</init>

<paint>
<text xpos = "36" ypos="30">Name</text>

<text xpos = "80" ypos="30">Current pos</text>
<text xpos = "210" ypos="30">Prog. pos</text>

</paint>

</form>

<!--
Menu R- Parameter
-->

<menu name = "MENU_R_PARAMETER">
<OPEN_FORM name = "R-Parameter" />

<softkey POSITION="16">
<caption>Back</caption>
<navigation>MAIN</navigation>
</softkey>

</menu>

<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 = "48" refvar="nck/Channel/Parameter/R[2]" />
<control name = "edit3" xpos = "322" ypos = "62" refvar="nck/Channel/Parameter/R[3]" />
<control name = "edit4" xpos = "322" ypos = "76" refvar="nck/Channel/Parameter/R[4]" />
<control name = "edit5" xpos = "322" ypos = "90" refvar="nck/Channel/Parameter/R[5]" />
<control name = "edit6" xpos = "322" ypos = "104" refvar="nck/Channel/Parameter/R[6]" />
<control name = "edit7" xpos = "322" ypos = "118" refvar="nck/Channel/Parameter/R[7]" />
<control name = "edit8" xpos = "322" ypos = "132" refvar="nck/Channel/Parameter/R[8]" />
<control name = "edit9" xpos = "322" ypos = "146" refvar="nck/Channel/Parameter/R[9]" />
<control name = "edit10" xpos = "322" ypos = "160" refvar="nck/Channel/Parameter/R[10]" />
<control name = "edit11" xpos = "322" ypos = "174" refvar="plc/mb170" />

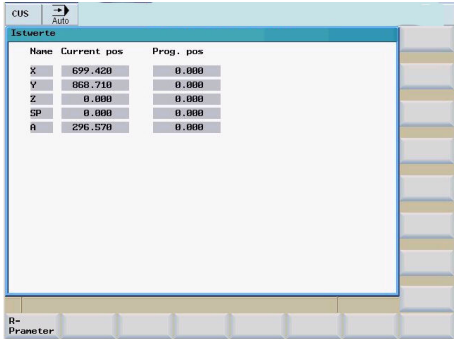
</init>

<paint>
<text xpos = "23" ypos = "34">R - Parameter 1</text>
<text xpos = "23" ypos = "48">R - Parameter 2</text>
<text xpos = "23" ypos = "62">R - Parameter 3</text>
<text xpos = "23" ypos = "76">R - Parameter 4</text>
<text xpos = "23" ypos = "90">R - Parameter 5</text>
<text xpos = "23" ypos = "104">R - Parameter 6</text>
<text xpos = "23" ypos = "118">R - Parameter 7</text>
<text xpos = "23" ypos = "132">R - Parameter 8</text>
<text xpos = "23" ypos = "146">R - Parameter 9</text>

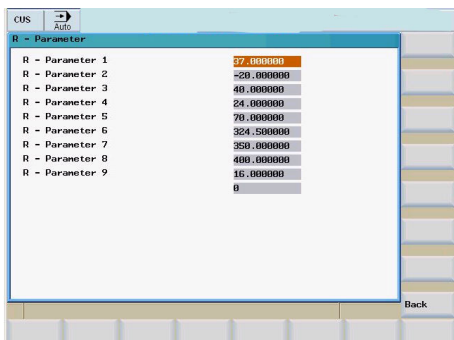
</paint>

</form>

</DialogGui>
    
```



实际值



R 参数

图 1-6 XML 脚本和带有参数的对话框窗口

1.5 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中。

说明

使用该文本文件时需要进行下列操作：

- 提供需要使用的语言。
 - 传输到控制系统相应的语言目录中。
-

1.6 XML 标签

1.6.1 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：

- 用于循环
- 循环时
- 使用循环进行
- 有条件加工
- 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 指令/标签说明 (页 17)

1.6.2 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

通过符号“<...>”标记的是属性值，会被实际使用的文字表达所替代。

示例：

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

会进行如下的参数设置：

```
<DATA_LIST action="read/write/append" id="my datalist">
```

1.6 XML 标签

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系统组件。</p> <p>句法: <CONTROL_RESET resetnc="TRUE" /></p> <p>属性:</p> <ul style="list-style-type: none">• RESETNC = "TRUE" 重新启动 NC 组件• RESETDRIVE = "TRUE" 重新启动驱动组件。

标签名称	含义
<p>CREATE_CYCLE_EVENT</p>	<p>如果分析程序开始处理标签 CREATE_CYCLE，则会首先将信息 <CREATE_CYCLE_EVENT> 发送到有效的 FORM 标签中。该信息会在分析程序根据参数表和创建规则生成 NC 指令之前，被用于循环参数的创建准备。</p> <div data-bbox="512 421 1286 704" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;"><CREATE CYCLE 7></p> </div> <p>句法:</p> <pre><CREATE_CYCLE_EVENT> </CREATE_CYCLE_EVENT></pre> <p>示例:</p> <pre><SOFTKEY_OK> ... <CREATE_CYCLE /> <SOFTKEY_OK> <FORM> <NC_INSTRUCTION>MY_CYCLE(\$P1, \$P2)</ NC_INSTRUCTION > <CREATE_CYCLE_EVENT> <type_cast name="P1" type="int"/> <OP> P1 = P1 * 150 </OP> </CREATE_CYCLE_EVENT> </FORM></pre>

1.6 XML 标签

标签名称	含义
DATA	<p>该标签可以直接写入 NC、PLC、GUD 和驱动数据中。</p> <p>地址构成可以查阅章节“组件地址分配”(页 85)。</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:</p> <pre><DATA name="<variable name>"> value </DATA></pre> <p>示例: <pre><DATA name = "plc/mb170"> 1 </DATA> ... <LET name = "tempVar"> 7 </LET> <!-- 局部变量"tempVar"的内容会写入到存储器字节 170 中 --> <DATA name = "plc/mb170">\$tempVar</DATA></pre> </p>

标签名称	含义
DATA_LIST	<p>该标签可以保存或者恢复所列出的驱动数据和机床数据。</p> <p>按行方式列出地址。地址构成可以查阅章节“组件地址分配”(页 85)。</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 <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有列表中 <i>write</i> - 将保存的值复制到相应的机床数据中 • id 标签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 地址汇编 </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>

1.6 XML 标签

标签名称	含义
DYNAMIC_INCLUDE	<p>标签包含 XML 脚本文件。</p> <p>与标签 INCLUDE 相反，在执行相应指令时才进行读取。</p> <p>在大型项目中该标签的使用能缩短用户使用或循环支持的载入时间。此外，对系统资源的平均需求也会下降，因为在会话过程中不需要一直调用所有的对话框。</p> <p>句法：</p> <pre><DYNAMIC_INCLUDE src="path name"/></pre> <p>示例：<pre><SOFTKEY POSITION="3"> <CAPTION>MY_MENU</CAPTION> <DYNAMIC_INCLUDE src/> <NAVIGATION>MY_MENU</NAVIGATION> </SOFTKEY></pre></p>
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>该标签包含对用户对话框的说明。其他相关的标签见章节“创建菜单与对话框窗口”中的说明。</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 47)） – 了解缺省值 • name 格式的名称 • type 允许的值是 <i>cycle</i>，表示一个用户循环窗口 • xpos 对话框左上角的 X 坐标（可选） • ypos 左上角的 Y 坐标（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>该标签会触发 HMI 重新启动。</p> <p>编译在该指令后会被中断。</p>

1.6 XML 标签

标签名称	含义
IF	<p>有条件的指令 (IF, THEN, ELSE)</p> <p>标签 THEN 和 ELSE 包含在标签 IF 中。</p> <p>标签 IF 后跟有条件，而条件在标签 CONDITION 中执行。需要通过其他的指令处理来确定运算结果。如果函数结果为真，则执行 THEN 分支而跳过 ELSE 分支。如果函数结果为假，则分析程序执行 ELSE 分支。</p> <p>句法:</p> <pre><IF> <CONDITION> 条件 != 7 </CONDITION> <THEN> 情况说明: 条件满足 </THEN> <ELSE> 情况说明: 条件不满足 </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>(另见该表格中的 DYNAMIC_INCLUDE)</p> <p>属性:</p> <ul style="list-style-type: none"> • src 包含路径名。 <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>数组：</p> <p>通过属性 dim（维数）可创建一维或二维的数组。各个数组元素的定址通过数组下标进行。</p> <p>对于二维数组，先指定行下标，再指定列下标。</p> <ul style="list-style-type: none"> 一维数组： 下标 0 至 4 <table border="1" data-bbox="549 612 580 768"> <tr><td>0</td></tr> <tr><td>1</td></tr> <tr><td>2</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> </table> <ul style="list-style-type: none"> 二维数组： 行下标 0 至 3，列下标 0 至 5 <table border="1" data-bbox="549 868 778 989"> <tr><td>0,0</td><td>0,1</td><td>0,2</td><td>0,3</td><td>0,4</td><td>0,5</td></tr> <tr><td>1,0</td><td>1,1</td><td>1,2</td><td>1,3</td><td>1,4</td><td>1,5</td></tr> <tr><td>2,0</td><td>2,1</td><td>2,2</td><td>2,3</td><td>2,4</td><td>2,5</td></tr> <tr><td>3,0</td><td>3,1</td><td>3,2</td><td>3,3</td><td>3,4</td><td>3,5</td></tr> </table> <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。 <pre data-bbox="512 1293 1002 1321"><LET name = "VAR1" type = "INT" /></pre> <ul style="list-style-type: none"> 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 用来指定数组元素的数量。对于二维数组，用逗号分隔第一维和第二维。对数组元素的存取通过数组下标进行，写在变量名称后的方括号内。 name[index] 或 name[row,column] <ul style="list-style-type: none"> 一维数组：dim="<元素数量>" 二维数组：dim="<行数>,<列数>" 未赋初值的数组元素用“0”占用。 	0	1	2	3	4	0,0	0,1	0,2	0,3	0,4	0,5	1,0	1,1	1,2	1,3	1,4	1,5	2,0	2,1	2,2	2,3	2,4	2,5	3,0	3,1	3,2	3,3	3,4	3,5
0																														
1																														
2																														
3																														
4																														
0,0	0,1	0,2	0,3	0,4	0,5																									
1,0	1,1	1,2	1,3	1,4	1,5																									
2,0	2,1	2,2	2,3	2,4	2,5																									
3,0	3,1	3,2	3,3	3,4	3,5																									

1.6 XML 标签

标签名称	含义
LET 续	<p>示例:</p> <p>一维数组: <code><let name="array" dim="10"></let></code></p> <p>二维数组: <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。如果一行要用 line feed <LF>（换行）结尾，应在行尾添加字符 <code>"\n"</code>。

标签名称	含义
	<pre data-bbox="512 285 1209 438"><LET name = "text" type = "string"> F4000 G94\\n G1 X20\\n Z50\\n M2\\n </LET>></pre> <p data-bbox="512 476 703 506">数组 (Arrays):</p> <pre data-bbox="512 512 916 634"><let name="list" dim="10,3"> {1,2,3}, {1,20} </let></pre> <pre data-bbox="512 672 1222 793"><let name="list_string" dim="10,3" type="string"> {"text 10","text 11"}, {"text 20","text 21"} </let></pre> <p data-bbox="512 895 576 925">赋值:</p> <p data-bbox="512 944 1337 974">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。</p> <p data-bbox="512 995 1158 1025">变量的有效性会一直延伸到最上一级的 XML 数据块。</p> <p data-bbox="512 1046 1174 1076">需要全局使用的变量应直接在标签 DialogGUI 后创建。</p> <p data-bbox="512 1098 858 1127">对于对话框须注意以下事项:</p> <ul data-bbox="512 1149 1038 1270" style="list-style-type: none"> ● 处理信息会打开相应的标签。 ● 在信息执行后会关闭标签。 ● 标签中的所有变量会随着关闭而被清零。 <p data-bbox="512 1336 703 1366">变量类型 struct</p> <p data-bbox="512 1387 1469 1459">该变量类型包含在使用结构名称时所能响应的变量汇总。一个结构能包含所有变量类型以及结构。</p> <p data-bbox="512 1481 1461 1553">结构中的变量通过标签“element”进行说明。标签的属性和初始化与 let 说明的属性和初始化相符。</p> <pre data-bbox="512 1719 1203 1896"><let name="name" type="struct"> <element name="<名称>" type="<变量类型>" /></pre>

1.6 XML 标签

标签名称	含义
	<pre>...</pre> <p data-bbox="472 385 576 412"></let></p> <p data-bbox="472 566 1166 593">通过结构和变量名称访问结构变量。两个名称由点隔开。</p> <pre><op> 结构_名称.变量_名称 = 值; </op></pre> <p data-bbox="472 1044 539 1072">示例:</p> <pre><let name="info" type="struct"> <element name="id" type="int" /> <element name="name" type="string" /> <element name="phone" type="string" /> </let></pre> <pre><op> info.id = 1; info.name = _T"my name"; info.phone = _T"0034 45634"; </op></pre>

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： <code><MSG text ="my message" /></code></p>
MSGBOX	<p>指令会打开一个信息盒，其给分支的返回值可供使用。</p> <p>句法： <code><MSGBOX text="<Message>" caption="<caption>" retvalue="<variable name>" type="<button type>" /></code></p> <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： 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，则信息盒会显示为该编号所保存的文本。</p> <p>示例： <code><MSGBOX text="测试消息" caption="信息" retvalue="result" type="BTN_OK" /></code></p>

标签名称	含义
OP	<p>该标签会执行给定的操作。</p> <p>可以执行章节“运算符 (页 48)”中所列出的各种运算。</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。地址构成可以查阅章节“组件地址分配” (页 85)。</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>一个运算标签内可以使用多个等式。用分号标记指令结束。</p> <p>示例:</p> <pre><op> x= x+1; y = y+1; </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>

标签名称	含义
XML_PARSER	<p>标签“XML_PARSER”可用于分析 XML 文件。</p> <p>分析程序编译 XML 文件并调用定义的回调功能。每个回调功能都属于一个预定义事件。编程人员可在该功能内部处理 XML 文件。</p> <p>预定义事件：</p> <ul style="list-style-type: none">● start document 分析程序打开文档并开始分析。● end document 分析程序关闭文档。● start element 分析程序找到了一个元素并创建了一个含所有属性和属性值的列表。

标签名称	含义
	<p>这些列表被转送给回调功能。</p> <ul style="list-style-type: none"> • end element 找到元素末尾。 • characters 分析程序转送元素的所有字符。 • error 分析程序发现了一个句法错误。 <p>如果出现事件，分析程序会调用回调功能并检查功能的返回值。如果功能“true”回馈，分析程序会继续进程。</p> <p>接口</p> <p>属性值 name 包含 XML 文件的路径。</p> <p>为给回调功能分配事件，须确定以下属性：</p> <p>标准</p> <ul style="list-style-type: none"> startElementHandler endElementHandler charactersHandler <p>可选</p> <ul style="list-style-type: none"> errorHandler documentHandler <p>属性值用于定义回调功能的名称。</p> <p>示例：</p> <pre><XML_PARSER name="f:\appl\xml_test.xml"> <!-- standard handler --> <property startElementHandler="startElementHandler" /> <property endElementHandler="endElementHandler" /> <property charactersHandler="charactersHandler" /> <!-- optional handler --> <property errorHandler="errorHandler" /> <property documentHandler="documentHandler" /> </XML_PARSER></pre>

标签名称	含义
XML_PARSER 续	<p>此外，分析程序还会提供可使回调功能访问事件数据的变量。</p> <p>startElementHandler:</p> <p>功能参数</p> <p>tag_name - 标签名称</p> <p>num - 找到的属性数</p> <p>系统变量</p> <p>\$xmlAttribute 属性名称范围为 0-num 的字符串数组。</p> <p>\$xmlValue 属性值范围为 0-num 的字符串数组。</p> <p>示例:</p> <pre><function_body name="startElementHandler" return="true" parameter="tag_name, num"> <print text="attribute name %s"> \$xmlAttribute[0]</print> <print text="attribute value %s"> \$xmlValue [0]</print> </function_body></pre> <p>endElementHandler:</p> <p>功能参数</p> <p>tag_name - 标签名称</p> <p>示例:</p> <pre><function_body name="endElementHandler" parameter="tag_name"> <print text="name %s"> tag_name </print> </function_body></pre>

1.6 XML 标签

标签名称	含义
XML_PARSER 续	<p>charactersHandler:</p> <p>系统变量</p> <p>\$xmlCharacters 含数据的字符串</p> <p>\$xmlCharactersStart 始终为 0</p> <p>\$xmlCharactersLength 字节数</p> <p>示例:</p> <pre><function_body name="charactersHandler" return="true" > <print text="chars %s"> \$xmlCharacters </print> </function_body></pre> <p>documentHandler:</p> <p>功能参数</p> <p>state 1 start document, 2 end document</p> <p>errorHandler:</p> <p>系统变量</p> <p>\$xmlErrorString 包含无效行（系统变量）</p> <p>示例:</p> <pre><function_body name="errorHandler" > <print text="error %s">\$xmlErrorString</print> </function_body></pre> <p>回调结果:</p> <p>\$return 如果为 1 (true), 分析程序会继续分析文件</p>

标签名称	含义
CONTROL	<p>创建后修改控制项</p> <p>可使用控制项标签在创建后修改已有控制项的控制项属性。必须同时给定待修改控制项的名称及新属性。该操作可在“Form”标签内部进行。</p> <p>可修改以下属性：</p> <ul style="list-style-type: none"> • xpos • ypos • width • height • color_bk • color_fg • access level • fieldtype • itemdata • min • max • default <p>无法修改参考变量。</p> <p>如果要通过软键“事件”触发来修改属性，则须使用标签“send message”，以上下文格式传送该要求。使用标签“message”采集信息。</p> <p>示例</p> <pre> <menu name = "main"> <open_form name = "attrib_form" /> <softkey POSITION="3"> <caption>Set%nr</caption> <send_message>1, 0</send_message> </softkey> <softkey POSITION="4"> <caption>Set%nrw</caption> <send_message>2, 0</send_message> </softkey> </menu> <form name="attrib_form"> <init> <control name="c_p0" xpos="60" ypos="70" width="272" /> </init> <message> <switch> <condition>\$message_par1</condition> <case value="1"> <control name="c_p0" fieldtype="readonly" /> </pre>

1.6 XML 标签

标签名称	含义
	<pre> </case> <case value="2"> <control name="c_p0" fieldtype="edity" /> </case> </switch> </message> </form> </pre>
OPERATION	<p>Operation (运算)</p> <p>一个等式内可以使用一个移位运算。</p> <p>向左移动运算符 “<<”</p> <p>通过功能 << 向左移动位。可直接或通过一个变量确定待移动值和移动增量数。如果达到数据格式极限，则在极限外移动位，此时，系统不显示故障信息。</p> <p>运算法则</p> <p>从左往右算且 ‘+’ 和 ‘-’ 位于 ‘<<’ 或 ‘>>’ 前</p> <p>示例：</p> <pre> <op> idx = idx << 2 </op> <op> idx = 3 + idx << 2 </op> </pre> <p>向右移动运算符 “>>”</p> <p>通过功能 >> 向右移动位。可直接或通过一个变量确定待移动值和移动增量数。如果达到数据格式极限，则在极限外移动位，此时，系统不显示故障信息。</p> <p>运算法则</p> <p>从左往右算且 ‘+’ 和 ‘-’ 位于 ‘<<’ 或 ‘>>’ 前</p> <p>示例：</p> <pre> <op> idx = idx >> 2 </op> <op> idx = 3+idx >> 2</op> </pre>

标签名称	含义
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 保存文本的变量的名称（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一致并用逗号加以间隔。</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>
PROGRESS_BAR	<p>标签打开或关闭一根进度条。进度条显示在应用窗口的下方。</p> <p>句法:</p> <pre><PROGRESS_BAR type="<true/false>"> value </ PROGRESS_BAR></pre> <p>属性:</p> <ul style="list-style-type: none"> • type = "TRUE" - 打开进度条 • type = "FALSE" - 关闭进度条 • min (可选) - 最小值 • max (可选) - 最大值 <p>值:</p> <ul style="list-style-type: none"> • Value 进度条的百分比位置 <p>示例:</p> <pre><PROGRESS_BAR type="true" min="0" max="101" >20 </ PROGRESS_BAR>.....<PROGRESS_BAR >50 </ PROGRESS_BAR>.....<PROGRESS_BAR type="false" >100 </PROGRESS_BAR></pre>

1.6 XML 标签

标签名称	含义
SEND_MESSAGE	<p>该标签会向有效的 FORM 标签发送含两个参数的信息，该信息在标签“Message”中进行处理。</p> <p>句法: <code><SEND_MESSAGE>p1, p2</SEND_MESSAGE></code></p> <p>示例: <code><SOFTKEY POSITION="3"> <caption>Set%nParameter</caption> <send_message>1, 0</send_message> </SOFTKEY></code></p> <code><FORM> <MESSAGE> <SWITCH> <CONDITION>\$message_par1</CONDITION> <CASE value="1"> </CASE> </SWITCH> </MESSAGE> </FORM></code>
SLEEP	<p>该标签会在指定的时间区间内中断脚本的执行。中断时间为返回值乘以 50ms 的基准时间。</p> <p>句法: <code><SLEEP value="中断时间" /></code></p> <p>示例: 等待时间 1.5 s。 <code><SLEEP value="30" /></code></p>
STOP	解释在该位置中断。

标签名称	含义
SWITCH	<p>指令 SWITCH 表示多项选择。表达式运用一次并与常数进行比较。如果表达式与常数一致，则执行 CASE 指令中的指令。</p> <p>如果常数与表达式不一致，则执行 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>
SHOW_CONTROL	<p>借助该标签可以指定是否显示控制项。</p> <p>句法:</p> <pre><SHOW_CONTROL name="<name>" type="<type>" /></pre> <p>属性:</p> <ul style="list-style-type: none"> • name 控制项的名称 • type = "TRUE" - 控制项可见 • type = "FALSE" - 控制项不可见（隐藏） <p>示例:</p> <pre><SHOW_CONTROL name="myEditfield" type="false" /> <SHOW_CONTROL name="myEditfield" type="true" /></pre>

1.6 XML 标签

标签名称	含义
TYPE_CAST	<p>该标签用于控制局部变量的数据类型转换。</p> <p>句法:</p> <pre><type_cast name="variable name" type=" new type" /></pre> <p>属性:</p> <ul style="list-style-type: none">• name 变量名称• type 新数据类型
THEN	当条件满足时应执行的指令 (IF, THEN, ELSE)

标签名称	含义
FOR	<p>FOR 循环 for （初始化；测试；继续）指令</p> <p>句法： <code><FOR></code> <code><INIT>...</INIT></code> <code><CONDITION>...</CONDITION></code> <code><INCREMENT>...</INCREMENT></code> 指令 ... <code></FOR></code></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 文本（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 继续（INCREMENT）.. 5. 继续使用 2。 <p>所有在 INIT、CONDITION 和 INCREMENT 分支中使用的变量都要在 FOR 循环的外部创建。</p> <p>示例： <code><LET name = "count">0</LET></code> <code><FOR></code> <code><INIT></code> <code><OP> count = 0</OP></code> <code></INIT></code> <code><CONDITION> count <= 7 </CONDITION></code> <code><INCREMENT></code> <code><OP> count = count + 1 </OP></code> <code></INCREMENT></code> <code><OP> "plc/qb10" = 1+ count </OP></code> <code></FOR></code></p>

1.6 XML 标签

标签名称	含义
WAITING	<p>标签在 NC 或者驱动复位后等待组件重新启动。</p> <p>属性:</p> <ul style="list-style-type: none"> • WAITINGFORNC = "TRUE" - 等待 NC 重启 • WAITINGFORDRIVE = "TRUE" - 等待驱动重启 <p>句法:</p> <pre><WAITING WAITINGFORNC = "TRUE"/></pre> <p>示例:</p> <pre>... <CONTROL_RESET resetnc = "true" resetdrive = "true"/> <WAITING waitingfornc = "true" waitingfordrive = "true" /> ...</pre>
WHILE	<p>While 循环</p> <p>WHILE (Test) 指令</p> <p>句法:</p> <pre><WHILE> <CONDITION>...</CONDITION> 指令 ... </WHILE></pre> <p>While 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>

标签名称	含义
DO_WHILE	<p>Do-While 循环</p> <p>DO 指令 WHILE (测试)</p> <p>句法:</p> <pre><DO_WHILE> 指令 ... <CONDITION>...</CONDITION> </DO_WHILE></pre> <p>Do-While 循环由一个指令程序段和一个条件构成。首先执行指令程序段中的代码，然后运用条件。如果条件为真 (true)，则重新执行代码部分。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>

1.6 XML 标签

标签名称	含义
TYPEDEF	<p>通过标签“typedef”可为数据类型确定一个新的名称。这对于结构定义的优势在于只要定义一次数据类型，然后便能在 LET 指令中作为数据类型使用。名称和类型预期作为属性值。分析程序仅仅支持结构定义的确立。</p> <p>类型定义中的变量通过标签“element”进行说明。标签的属性与 let 指令的属性相符。</p> <pre data-bbox="472 634 1198 878"> <typedef name="<名称>" type="struct"> <element name="<名称>" type="<变量类型>" /> </typedef> </pre> <p>定以后，名称可作为数据类型，用于 LET 指令。</p> <pre data-bbox="472 1017 1161 1049"> <let name="<变量名称>" type="<名称>"></let> </pre> <p>示例:</p> <pre data-bbox="472 1151 1150 1349"> <typedef name="my_struct" type="struct"> <element name="id" type="int" /> <element name="name" type="string" /> <element name="phone" type="string" /> </typedef> </pre> <pre data-bbox="472 1449 1150 1776"> <let name="info" type="my_struct"></let> <op> info.id = 1; info.name = _T"my name"; info.phone= _T"0034 45634"; </op> </pre>

1.6.3 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"
```

1.6.4 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，则必须进行写法转换。

关系到下列字符：

字符	XML 中的表示法
<	<
>	>
&	&
"	"
'	'

1.6.5 运算符

运算指令可以处理下列运算：

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

运算由左向右进行。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。

1.6.6 创建软键菜单和对话框窗口

欲链接用户菜单，在 XML 说明中应含有名称为“main”的菜单标签。在激活操作区 <CUSTOM> 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。

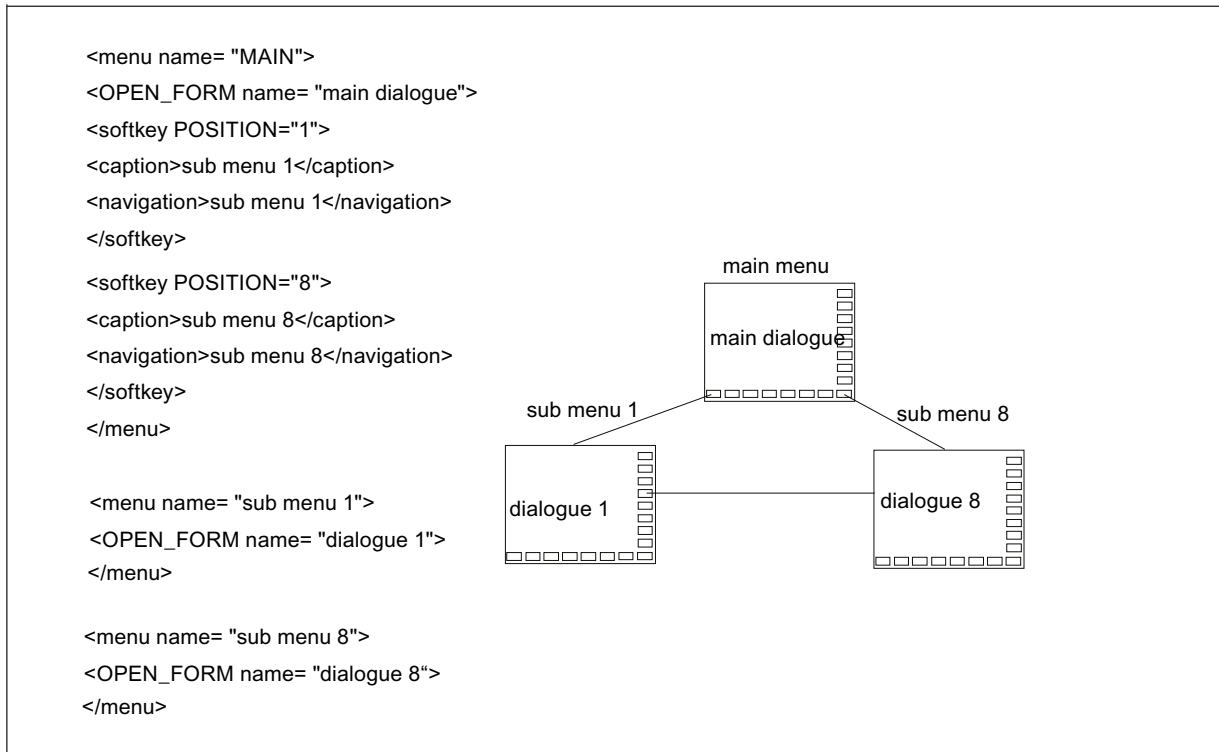



图 1-7 菜单结构

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 对话框的背景颜色（颜色代码参见章节“颜色代码”） • name 格式的名称 • type <i>cycle</i> -属性定义一个循环窗口 • xpos 对话框左上角的 X 坐标（可选） • ypos 左上角的 Y 坐标（可选） • width X 方向上的长度（以像素为单位）（可选） • height Y 方向上的长度（以像素为单位）（可选）
FORM 续	<p>对话框信息：</p> <ul style="list-style-type: none"> • INIT • PAINT • TIMER • CLOSE • FOCUS_IN • INDEX_CHANGED • EDIT_CHANGED • KEY_EVENT • MESSAGE

标签名称	含义
FORM 续	 <pre data-bbox="1018 321 1300 1202"> <form name="test form"> <init> </init> <paint> </paint> <timer> </timer> <focus_in> </focus_in> <index_changed> </index_changed> <edit_changed> </edit_changed> <message> </ message > <key_event> </ key_event > <close> </close> </form> </pre>

1.6 XML 标签

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>

标签名称	含义
KEY_EVENT	<p>对话框信息</p> <p>标签 KEY_EVENT 可将键盘事件的评估链接到标签 FORM 中。如果在 FORM 中存在该标签，则系统会将 MF2 键盘代码发送给有效的 FORM。如果变量 \$actionresult 未置零，系统会接着处理键盘事件。</p> <p>键盘代码会作为整型值在变量 \$keycode 中提供。</p> <p>示例：</p> <p>在变量 exclude_key 中输入的信号应从输入电流中剔除。</p> <pre> <LET name="stream" type="string"/> <LET name="exclude_key" type="string"/> <FORM name = "keytest_form"> <INIT> <CONTROL name = "p1" xpos = "120" ypos = "84" width ="200" refvar="stream" hotlink="true" /> <CONTROL name = "p2" xpos = "160" ypos = "104" width ="8" refvar="exclude_key" hotlink="true" /> </INIT> <PAINT> <text xpos = "8" ypos = "84">data stream</text> <text xpos = "8" ypos = "104">exclude key</text> </PAINT> <KEY_EVENT> <LET name="excl_keycode" type="string"/> <OP>excl_keycode = exclude_key</OP> <type_cast name="excl_keycode" type="int" /> <PRINT text="%d %d">\$keycode, excl_keycode</PRINT> <IF> <CONDITION>\$keycode == excl_keycode</CONDITION> <THEN> <op> \$actionresult = 0</op> </THEN> </IF> </KEY_IVENT> </FORM> </pre>

标签名称	含义
MESSAGE	<p>对话框信息</p> <p>如果在脚本中执行指令 <code>Send_message</code>，分析程序会执行标签 <code>MESSAGE</code>。值 <code>p1</code> 和 <code>p2</code> 会在变量 <code>\$message_par1</code> 和 <code>\$message_par2</code> 中提供（参见标签“<code>SEND_MESSAGE</code>”）。</p> <p>句法:</p> <pre><MESSAGE> </MESSAGE></pre> <p>示例:</p> <pre><LET name="user_selection" /> <SOFTKEY POSITION="3"> <CAPTION>Set%nParameter</CAPTION> <SEND_MESSAGE>1, 10</SEND_MESSAGE> </SOFTKEY> <FORM> <MESSAGE> <SWITCH> <CONDITION>\$message_par1</CONDITION> <CASE value="1"> <OP> user_selection = \$message_par2 </OP> </CASE> </SWITCH> </MESSAGE> </FORM></pre>

标签名称	含义
SEND_MESSAGE	<p>该标签会向有效的 FORM 标签发送含两个参数的信息，该信息在标签“Message”中进行处理（另见 MESSAGE）。</p> <p>句法: <code><SEND_MESSAGE>p1, p2</SEND_MESSAGE></code></p> <p>示例: <code><LET name="user_selection" /></code> <code><SOFTKEY POSITION="3"></code> <code> <CAPTION>Set%nParameter</CAPTION></code> <code> <SEND_MESSAGE>1, 10</SEND_MESSAGE></code> <code></SOFTKEY></code> <code>...</code> <code>...</code> <code><FORM></code> <code>...</code> <code>...</code> <code><MESSAGE></code> <code><SWITCH></code> <code><CONDITION>\$message_par1</CONDITION></code> <code><CASE value="1"></code> <code><OP> user_selection = \$message_par2 </OP></code> <code>...</code> <code></CASE></code> <code></SWITCH></code> <code></MESSAGE></code> <code>...</code> <code>...</code> <code></FORM></code></p>
FOCUS_IN	<p>对话框信息</p> <p>当系统将焦点设置到控制项时，调用该标签。为了识别控制项，系统将控制项的名称复制到变量 \$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: <code><focus_in></code> <code><PRINT text="focus on filed:%s, %d">\$focus_name,</code> <code>\$focus_item_data </PRINT></code> <code></focus_in></code></p>

标签名称	含义
PAINT	<p>对话框信息</p> <p>该标签在显示对话框时执行。在此要给定需要在对话框中显示的所有文本与图像。</p> <p>当系统确定对话框的一些地方需要重新显示时，会继续执行该标签。这可以通过，例如，关闭上一级窗口来实现。</p>
TIMER	<p>对话框信息</p> <p>循环处理该标签。</p> <p>每个 FORM 都分配了一个 TIMER，大约每 100 ms 就会触发一次 TIMER 标签的执行。</p>
CAPTION	<p>该标签包含有对话框的标题。</p> <p>在 INIT（初始化）标签内使用该标签。</p> <p>标题行可划分为多列。</p> <p>输出文本前必须对 caption 标签和 define_section 属性进行编程，以便划分标题行。define_section 属性确定行的数量</p> <p>通过 property 属性定义每一列的长度、起始位置和文本方向。位置和长度说明是基于表格宽度的百分数值。</p> <p>属性值 value 表示列编号（从零开始）</p> <pre><caption define_sections="3"> <property value="0" length="20" position="2" alignment="left" /> <property value="1" length="20" position="79" alignment="right" /> </caption></pre> <p>然后可以借助列索引指定属性 index 来指定列文本。</p> <pre><caption index="0">Spalte1</caption> <caption index="1">Spalte2</caption></pre> <p>句法:</p> <pre><CAPTION>Titel</CAPTION></pre> <p>示例:</p> <pre><CAPTION>my first dialogue</CAPTION></pre>
CLOSE	<p>对话框信息</p> <p>在关闭对话框之前处理该标签。</p>

标签名称	含义
CLOSE_FORM	<p>该标签会关闭激活的对话框。</p> <p>只有通过 MMC 指令打开对话框且为操作者提供了关闭对话框的软键功能时，该指令才是必须的。一般而言，会自动管理对话框，而无需另外加以关闭。</p> <p>句法:</p> <pre><CLOSE_FORM/></pre> <p>示例:<pre><softkey_ok> <caption>OK</caption> <CLOSE_FORM /> </navigation>main_menu</navigation> </softkey_ok></pre></p>

标签名称	含义
CONTROL	<p>该标签用来创建控制项。</p> <p>句法:</p> <pre><CONTROL name = "<control name>" xpos = "<X 坐标>" ypos = "<Y 坐标>" refvar = "<NC 变量>" hotlink = "true" format = "<格式>" /></pre> <p>属性:</p> <ul style="list-style-type: none"> • name 数组名称 该名称会同时显示一个局部变量，并且在 FORM 中不允许重复使用。 • xpos 左上角的 X 坐标 • ypos 左上角的 Y 坐标 • fieldtype 数组类型 如未给定类型，则数组被设置为编辑数组。 <ul style="list-style-type: none"> - edit 数据可以修改。 - readonly 数据无法修改。 <p>combobox 数组显示相应的名称来代替数值。 如果选择了数组类型“combobox”，则要为数组另外分配显示表达式。 为此要使用标签 <ITEM>。 Combo Box 将当前选定文本的下标保存在控制项所属的变量中（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 – listbox 该数组类型会生成一个空的列表框控制项。 通过该标签 <ITEM> 会向列表框中添加一个列表框元素。 ITEM-属性 value 会向该元素赋一个唯一的值。 这可用于，例如，识别该元素。 参数 width 和 height 用来指定列表框的宽度和高度。 在创建了控制项后，其他的列表框元素可通过功能 AddItem、InsertItem 或 LoadItem 来添加。 在创建了控制项后，其他的列表框元素可通过功能 添加项 或 插入项 来添加。参数 插入数据 在该控制项时不进行处理。 – itemlist 该数组类型会生成一个静态控制项，取代数值而显示相应的名称。 通过该标签 <ITEM> 会向数组中分配一个名称。 • item_data 可以使用用户专用的整数值为该属性赋值。该值被提供给 FOCUS_IN 信息，用来识别焦点数组。 • refvar 可以与数组互连的参考变量的名称（可选）。 • hotlink = "TRUE" 如果参考变量的值发生了变化，该数组会自动进行更新（可选）。 • format 该属性定义指定变量的显示格式。 格式定义参见“print-Tag （可选）”。

标签名称	含义
CONTROL 续	<p>属性:</p> <ul style="list-style-type: none"> • time 定义数据的刷新率（可选）。 默认值 200 ms 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 • color_bk 该属性设置控制项的背景颜色。 • color_fg 该属性设置控制项的前景颜色。 (“颜色代码”参见章节“颜色代码 (页 47)”)) • display_format 该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、图形框或复合框分配表达内容（例如要显示的文本或图形）：</p> <p>句法:</p> <p><ITEM>表达</ITEM></p> <p><ITEM value = "<值>">表达</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: 可使用功能 additem 或 loaditem 添加元素。 ● 清除内容: 可使用功能 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
CONTROL 续	<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 XmlUserDlg_ 并转发到帮助系统中。这样就可以从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 (数据存取)	<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 <code>INIT</code> 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：在 <code>hotlink</code> 设置为 <code>true</code> 的控制器中总是直接写入和读取。</p> <p>属性：</p> <ul style="list-style-type: none"> • <code>type = "TRUE"</code> – 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。 • <code>type = "FALSE"</code> – 值只通过标签 <code>UPDATA_DATA type = "FALSE"</code> 复制到参考变量中。 <p>示例：</p> <pre><DATA_ACCESS type = "true" /></pre>
EDIT_CHANGED	<p>对话框信息</p> <p>当 <code>EDIT</code> 控制项的内容发生了变化时，调用该标签。</p> <p>为了识别控制项，系统将控制项的名称复制到变量 <code>\$focus_name</code> 中并将 <code>item_data</code> 的属性值复制到变量 <code>\$focus_item_data</code> 中。系统自动创建变量。</p> <p>示例：</p> <pre><EDIT_CHANGED> <print text="index changed filed:%s, %d"> \$focus_name, \$focus_item_data </print> </EDIT_CHANGED></pre>

标签名称	含义
INDEX_CHANGED	<p>对话框信息</p> <p>当操作员修改了复合框的选择时，调用该标签。</p> <p>为了识别控制项，系统将控制项的名称复制到变量 \$focus_name 中，将属性 item_data 的值复制到变量 \$focus_item_data 中。系统自动创建变量。</p> <p>提示：</p> <p>分配给控制项的参考变量此时还未与控制项变量进行匹配，所含的仍是上一次复合框选择的下标。</p> <p>示例：</p> <pre><INDEX_CHANGED> <print text="index changed filed:%s, %d"> \$focus_name, \$focus_item_data </print> </INDEX_CHANGED></pre>
MENU	<p>该标签定义一个菜单，其中包含有软键说明和要打开的对话框。</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>

标签名称	含义
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: <NAVIGATION>menu name</NAVIGATION></p> <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 对话框窗口的名称 <p>句法:</p> <pre><OPEN_FORM name = "<form name>" /></pre> <p>示例:<pre><menu name = "main"> <open_form name = "main_form" /> <softkey POSITION="1"> <caption>main form</caption> <navigation>main</navigation> </softkey> </menu></pre><pre><form name="main_form"> <init> </init></pre><pre> <paint> </paint></pre><pre></form></pre></p>

标签名称	含义
PROPERTY	<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 = “<背景颜色代码>” • color_fg = “<字体颜色代码>” • 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 • transparent = “位图的透明色” 颜色代码（颜色代码参见章节“颜色代码”） • tooltip = 当光标移动到控制项上时，显示提示文本。 句法为: <code><property tooltip="提示文本" /></code> <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>

标签名称	含义
SOFTKEY	<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>以下属性生效于:</p> <ul style="list-style-type: none"> • type 定义软键的属性。 user_controlled - 软键的显示由脚本确定 toggle_softkey - 软键在交替按下时显示或不按时显示 • refvar 只在连接时使用类型 toggle_softkey 。 参考变量，将当前的软键属性复制到其中。 定义了类型为“String”的变量，含有属性“按下”、“不按”或“禁用”（参见标签 state）。 • picture 借助该属性，位图会按左对齐方式输出在软键上。 必须给出完整的路径。 可显示的文本字符的数量受位图宽度的限制。 <p>可以在软键程序块中确定下列其他操作:</p> <ul style="list-style-type: none"> • picturealignment 通过该属性指定图片的校准。 图片默认位于软键的左侧。 可以设置下列值，用于校准: <ul style="list-style-type: none"> - top 上边沿 - bottom 下边沿 - left 左边沿 - right 右边沿 - center 居中 • caption 软键文本 • state 只在连接时使用类型 user_controlled 。 该标签向系统分配所需的软键显示。 <p>句法:</p>

标签名称	含义
	<p data-bbox="520 287 823 319"><code><state type="<state>" /></code></p> <p data-bbox="520 321 783 353">可以指定以下字符串：</p> <ul data-bbox="520 370 852 734" style="list-style-type: none"><li data-bbox="520 370 798 434">- notpressed 软键在不按时显示。<li data-bbox="520 453 798 517">- pressed 软键在按下时显示。<li data-bbox="520 536 852 600">- disabled 软键禁用并显示为灰色。<li data-bbox="485 619 651 651">• navigation<li data-bbox="485 661 719 693">• update_controls<li data-bbox="485 704 619 736">• function


1.6 XML 标签

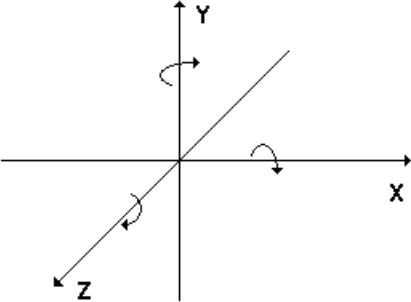
标签名称	含义
SOFTKEY 续	<p>句法:</p> <p>标准软键:</p> <pre><state type="<softkey state>" /> <softkey position = "<1>"> </softkey></pre> <p>或</p> <p>受脚本控制的软键:</p> <pre><softkey position = "<1>" type="<user_defined>" > <state type="<softkey state>" /> </softkey></pre> <p>或</p> <p>切换软键:</p> <pre><softkey position = "<1>" type="<toggle_softkey>" refvar="<variable name>" > </softkey></pre> <p>示例:</p> <pre><let name="define_sk_type" type="string">PRESSED</let> <let name="sk_type">1</let></pre> <pre><softkey POSITION="1" type="user_controlled" > <caption>Toggle%nSK</caption> <if> <condition>sk_type == 0 </condition> <then> <op> sk_type = 1 </op> <op> define_sk_type = _T"PRESSED" </op> </then> <else> <op> define_sk_type = _T"NOTPRESSED" </op> <op> sk_type = 0 </op> </else> </if> <state type="\$\$\$define_sk_type" /> </softkey></pre>

标签名称	含义
SOFTKEY 续	<p>示例:</p> <p>或</p> <pre><let name="curr_softkey_state" type="string">PRESSED</let> </softkey> <softkey POSITION="3" type="toggle_softkey" refvar="curr_softkey_state"> <caption>Toggle%nSK</caption> ... </softkey></pre> 
SOFTKEY_OK	<p>该标签定义软键“确认”的反应。</p>  <p>可以在软键程序块中确定下列其他操作:</p> <ul style="list-style-type: none"> • navigation • update_controls • function <p>句法:</p> <pre><SOFTKEY_OK> </SOFTKEY_OK></pre>

1.6 XML 标签

标签名称	含义
SOFTKEY_CANCEL	<p>该标签定义软键“取消”的反应。</p>  <p>可以在软键程序块中确定下列其他操作：</p> <ul style="list-style-type: none"> • navigation • update_controls • function <p>句法：</p> <pre><SOFTKEY_CANCEL> </SOFTKEY_CANCEL></pre>
SOFTKEY_BACK	<p>该标签定义软键“返回”的反应。</p>  <p>可以在软键程序块中确定下列其他操作：</p> <ul style="list-style-type: none"> • navigation • update_controls • function <p>句法：</p> <pre><SOFTKEY_BACK> </SOFTKEY_BACK></pre>

标签名称	含义
SOFTKEY_ACCEPT	<p>该标签定义软键“接收”的反应。</p>  <p>可以在软键程序块中确定下列其他操作：</p> <ul style="list-style-type: none"> • navigation • update_controls • function <p>句法：</p> <pre><SOFTKEY_ACCEPT> </SOFTKEY_ACCEPT></pre>
文本	<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：</p> <pre><TEXT xpos = "<X 坐标>" ypos = "<Y 坐标>"> Text </TEXT></pre> <p>属性：</p> <ul style="list-style-type: none"> • xpos 左上角的 X 坐标 • ypos 左上角的 Y 坐标 • color 文本颜色（颜色代码参见章节 颜色代码 (页 47)） <p>值：</p> <p>待显示的文本</p>

标签名称	含义
IMG (图像)	<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:</p> <pre><IMG xpos = "<X 坐标>" ypos = "<Y 坐标>" name = "<名称>" /> </pre> <p>属性:</p> <ul style="list-style-type: none"> • xpos 左上角的 X 坐标 • ypos 左上角的 Y 坐标 • name 完整路径名称 • transparent 位图的透明色 (参见章节“颜色代码”) : • xrot, yrot, zrot (仅用于 802Dsl / 808D) X 轴、Y 轴和 Z 轴内的图片围绕旋转点旋转。 • xref, yref, zref (仅用于 802Dsl / 808D) 定义旋转点 没有指定旋转点时, 图片绕其中心旋转。此时 Z 平面为 0。 

标签名称	含义
IMG 续	<p data-bbox="480 287 549 321">示例:</p> <p data-bbox="480 336 746 370">图片绕 Z 轴旋转 34°。</p> <pre data-bbox="480 393 1414 519"> </pre>  <p data-bbox="480 861 549 895">可选:</p> <p data-bbox="480 910 1374 944">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 data-bbox="480 963 719 1115" style="list-style-type: none">• width 宽度，单位像素• height 高度，单位像素 <p data-bbox="480 1181 549 1215">示例:</p> <pre data-bbox="480 1223 1458 1315"> </pre>

1.6 XML 标签

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:</p> <pre><BOX xpos = "<X 坐标>" ypos = "<Y 坐标>" width = "<X 长度>" height = "<Y 长度>" color = "<颜色代码>" /></pre> <p>属性:</p> <ul style="list-style-type: none">• xpos 左上角的 X 坐标• ypos 左上角的 Y 坐标• width X 方向上的宽度（以像素为单位）• height Y 方向上的高度（以像素为单位）• color 颜色代码（颜色代码参见章节“颜色代码”）

标签名称	含义
FUNCTION (功能)	<p>功能调用</p> <p>该标签可以执行属性 “name” 中定义的功能体。</p> <p>属性:</p> <ul style="list-style-type: none">• name = “功能体的名称”• return = “用于储存功能结果的变量名称” <p>值:</p> <p>需要将其传输到功能体的变量列表。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:</p> <pre><FUNCTION name = "<function name>" /></pre> <p>待调用的功能等待一个返回值</p> <pre><FUNCTION name = "<function name>" return = "<Variablennam>" /></pre> <p>参数传递</p> <pre><FUNCTION name = "<function name>"> var1, var2, var3 </FUNCTION> <FUNCTION name = "<function name>"> _T"Text", 1.0, 1 </FUNCTION></pre> <p>示例:</p> <p>参见“FUNCTION_BODY”</p>

标签名称	含义
FUNCTION_BODY	<p>功能体</p> <p>该标签包含有一个子功能的功能体。要在 <code>DialogGui</code> 标签中对功能体进行编程。</p> <p>属性:</p> <ul style="list-style-type: none"> • name= “功能体的名称” • parameter = “参数列表”（可选） 属性列出了需要的传输参数。参数相互之间以逗号分隔。 调用功能体时将功能调用中给定参数的值复制到所列出的传输参数中。 • return = “true”（可选） 如果属性设为 <code>true</code>，则创建局部变量 <code>\$return</code>。将功能的返回值复制到其中，在放弃该功能时可以将其继续传输给待调用的功能。 <p>句法:</p> <p>无参数的功能体</p> <pre><FUNCTION_BODY name = "<function name>"> </ FUNCTION_BODY></pre> <p>带参数的功能体</p> <pre><FUNCTION_BODY name = "<function_name>" parameter = "<p1, p2, p3>"> ... <LET name = "tmp"></LET> <OP> tmp = p1 </OP> ... </FUNCTION_BODY></pre> <p>带返回值的功能体</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>

1.6 XML 标签

标签名称	含义
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功能，则应将功能的名称作为其它属性加以定义。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 地址名称 • function 功能名称 <p>句法：</p> <pre><REQUEST name = "<NC-Variable>" /></pre> <p>或</p> <pre><REQUEST name = "<NC-Variable>" function="<function name>" /></pre> <p>示例：</p> <pre><request name ="plc/mb10" /></pre> <p>或</p> <pre><function_body name="my_function" > <print text="value changed" /> </function_body> <request name ="plc/mb10" function="my_function"/></pre>

标签名称	含义
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>

1.7 创建用户菜单

1.7.1 创建加工循环窗口

循环支持功能可以通过格式对话框自动创建和反编译循环调用。

提供以下标签实施该功能:

- **NC_INSTRUCTION**
- **CREATE_CYCLE**

为了标记循环窗口,应在标签 **FORM** 中将属性 **type** 的值定义为 **cycle**。该标记可实现指令 **NC_INSTRUCTION** 的执行。

示例

```
<FORM name = "cycle100_form" type= "CYCLE">
...
...
</FORM>
```

标签 **NC_INSTRUCTION** 含有要执行的循环调用。所有循环参数都通过占位符进行了预留。

1.7 创建用户菜单

示例

```
<FORM name = "cycle100_form" type= "CYCLE">
<NC_INSTRUCTION refvar= "cyc_string" >Cycle100 ($p1, $p2, $p3)</
NC_INSTRUCTION>
...
...
...
</FORM
```

标签 **CREATE_CYCLE** 提供保存在占位符变量中的值并生成 NC 指令。

然后复制到指定的变量中。

标签名称	含义
NC_INSTRUCTION	<p>使用该标签可定义要创建的 NC 指令。</p> <p>列出的所有循环参数都会自动作为 FORM 字符串变量被创建并可用于 FORM。</p> <p>前提条件： FORM 的属性 type 设置为 CYCLE。</p> <p>属性：</p> <ul style="list-style-type: none"> • refvar 如果向标签分配了一个参考变量，则所有参数都会预设为参考变量中保存的 NC 程序段中的值。 <p>句法： <NC_INSTRUCTION> 带占位符的 NC 指令 </ NC_INSTRUCTION></p> <p>示例： <let name="cyc_string" type="string"> Cycle100(0, 1000, 5)</let> <FORM name = "cycle100_form" type= "CYCLE"> <NC_INSTRUCTION refvar= "cyc_string" >Cycle100(\$p1, \$p2, \$p3)</ NC_INSTRUCTION> </FORM></p>

标签名称	含义
CREATE_CYCLE	<p>该标签会生成一个 NC 程序段，其句法由标签 NC_INSTRUCTION 确定。</p> <p>在生成 NC 指令前，分析程序会调用标签 FORM 的 CYCLE_CREATE_EVENT。该标签可用于计算循环参数。</p> <p>句法： <code><CREATE_CYCLE/></code></p> <p>属性：</p> <ul style="list-style-type: none"> • refvar 如果向该标签分配了一个参考变量，则 NC 指令会复制到该变量中。 <p>示例： <code><LET name="cyc_string" type="string"> Cycle100(0, 1000, 5)</LET></code></p> <pre> <SOFTKEY_OK> <caption>OK</caption> <CREATE_CYCLE /> <close_form /> <navigation>main_menu</navigation> </SOFTKEY_OK> </pre> <p>或</p> <pre> <SOFTKEY_OK> <caption>OK</caption> <CREATE_CYCLE refvar= "cyc_string" /> <close_form /> <navigation>main_menu</navigation> </SOFTKEY_OK> </pre>

1.7.2 替代符号

系统可以确定有关运行时间的控制特性（属性值）。为了能够使用该功能，需要在一个局部变量中提供所需的特性并使用前置的 **\$ 符号** 将变量名称作为属性值传输给标签。

如果标签要以字符串作为属性值或值，则应在变量名称前加上符号 **\$\$\$**。

示例：

```
<let name="my_ypos">100</let>
<let name="field_name" type="string"></let>

<control name = "edit1" xpos = "322" ypos = "$my_ypos"
refvar="nck/Channel/Parameter/R[1]" />

<op>my_ypos = my_ypos +20 </op>

<control name = "edit2" xpos = "322" ypos = "$my_ypos"
refvar="nck/Channel/Parameter/R[2]" />

<print name =" field_name" text="edit%d">3</print>
<op>my_ypos = my_ypos +20 </op>

<control name = "$field_name" xpos = "322" ypos = "$my_ypos"
refvar="nck/Channel/Parameter/R3]" />

<caption>$$$field_name</caption>
```

1.8 元素定址

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。

1.8.1 PLC 定址

PLC 地址分配从路径 **plc** 开始。

表格 1-3 允许使用下列地址：

DBx.DB(f)	数据块
I(f)x	输入端
Q(f)x	输出端

1.8 元素定址

M(f)x	标志
V(f)x	变量

DBx.DBXx.b	数据块
Ix.b	输入端
Qx.b	输出端
Mx.b	标志
Vx.b	变量

表格 1-4 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>
```

1.8.2 NC 变量定址

NC 变量的定址从路径 **nck** 开始。

在该部分之后跟有数据地址，其结构可以查阅 **SINUMERIK 802D sl** 的参数手册。

示例：

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

1.8.3 在运行时生成 NC/PLC 地址

支持在运行时生成地址名称。

此时可将字符串变量的内容作为地址用在操作指令以及功能 **nc.cap.read** 和 **nc.cap.write** 中。

对于该定址模式应注意以下事项：

- 变量名称写在引号内。
- 在变量名称前标记三个 **,\$** 符号。

句法：

```
"$$$variable name"
```

示例：

```
<LET name="var_adr" type="string"></LET>
<PRINT name="var_adr" text="VB9000%d"> 2000</PRINT>
或
<PRINT name="var_adr" text="DB9000.DBW%d"> 2000</PRINT>
<OP> "$$$var_adr" = 1 </OP>
```

1.8.4 驱动组件的定址

驱动组件的定址从路径 **drive** 开始。

后面跟有驱动设备的详细规格：

CU

DC

该部分要添加上待设置参数。

示例：

1.8 元素定址

```
<LET name="r0002_content"></LET>
<LET name="p107_content"></LET>

<!-- 读取 CU 上 r0002 的值 -->
<OP> r0002_content = "drive/cu/r0002" </OP>
<OP> r0002_content = "drive/cu/r0002[CU1]" </OP>

<!-- 读取 NX1 上 r0002 的值 -->
<OP> r0002_content = "drive/cu/r0002[CU2]" </OP>

<!-- 读取 CU 上 p107[0] 的值 -->
<OP> p107_content = "drive/cu/p107[0]" </OP>
<PRINT text="%d"> p107_content </PRINT>

<!-- 读取 CU 上 p107[0] 的值 -->
<OP> p107_content = "drive/cu/p107[0, CU1]" </OP>
<PRINT text="%d"> p107_content </PRINT>

<!-- 读取 NX1 上 p107[0] 的值 -->
<OP> p107_content = "drive/cu/p107[0, CU2]" </OP>
<PRINT text="%d"> p107_content </PRINT>
```

驱动对象的地址分配:

为了给单个对象分配地址，要在参数之后的方括号中给出所需的对象。

说明

驱动对象编号会与驱动对话框中所使用的编号有所偏差，因为组件 CU、ALM 以及所有已连接的集线器都是包含在一起进行连续的编号。

可以通过以下方式获得 DO 编号:

- 所有已连接的驱动对象都根据其接口编号列在相应 CU 的数组 p978 中。
 - 获取所需接口的数组下标并增加一。得出的值便是定址所需的 DO 下标。
-

Control Unit Parameter		DP3.SLAUE3:CU_1_083(1)	
r975[7]	Antriebsobjekt Identifikation:Antriebsobjektn...	1	M
r975[8]	Antriebsobjekt Identifikation:Reserviert	0	M
r975[9]	Antriebsobjekt Identifikation:Reserviert	0	M
r975[10]	Antriebsobjekt Identifikation:Firmware patch...	6400	M
p976	Alle Parameter zurücksetzen und laden	[0] Inaktiv	M
p977	Alle Parameter speichern	[0] Inaktiv	M
p978[0]	Liste der Antriebsobjekte	1	M
p978[1]	Liste der Antriebsobjekte	0	M
p978[2]	Liste der Antriebsobjekte	0	M
p978[3]	Liste der Antriebsobjekte	0	M
p978[4]	Liste der Antriebsobjekte	0	M
p978[5]	Liste der Antriebsobjekte	0	M
p978[6]	Liste der Antriebsobjekte	0	M
p978[7]	Liste der Antriebsobjekte	0	M
p978[8]	Liste der Antriebsobjekte	0	M
p978[9]	Liste der Antriebsobjekte	0	M
p978[10]	Liste der Antriebsobjekte	0	M
p978[11]	Liste der Antriebsobjekte	0	M

Liste der Antriebsobjekte

图 1-8 控制系统上的驱动参数 p0978 [...]

参数号 [do<DO-index>]

示例:

p0092 [do1]

可以选择利用“替代符号”**\$<变量名称>** 从一个局部变量中读取驱动下标。
z.B. DO\$局部变量

示例:

```
<DATA name = "drive/cu/p0092">1</DATA>
<DATA name = "drive/dc/p0092 [do1] " >1</DATA>
```

间接地址分配:

```
<LET name = "driveIndex" 0 </LET>
<OP> driveIndex = $ctrlout_module_nr[0, AX1] </OP>
<DATA name = "drive/dc [do$driveIndex] /p0092">1</DATA>
```

1.8.5 机床数据和设定数据的定址

可以使用 \$ 符号标记驱动和设定数据，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：

参数在该数组中给出数据索引。

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>
```

1.8.6 用户数据的定址

用户数据的定址以路径 **gud** 开始，后面跟有 **GUD** 名称。

在数组名称后面的方括号中指定所需的数组下标。

示例：

```
<DATA name ="gud/syg_rm[0]"
<OP>"gud/syg_rm[0]" 0 10 </op>
```

全局用户数据的编址

编址从路径段 **gud** 开始，后面跟有范围指定 **CHANNEL**。该地址部分之后跟有 **GUD** 范围指定：

GUD 范围	分配
sgud	西门子的 GUD
mgud	机床制造商的 GUD
ugud	用户的 GUD

然后指定 **GUD** 名称。为一个字段编址时，名称之后是尖括号的字段索引。

示例：

```
<data name ="gud/channel/mgud/syg_rm[0]">1</data>
<op>"gud/channel/mgud/syg_rm[0]" = 5*2 </op>
```

1.9 预定义功能

脚本语言提供有不同的字符串处理和数学标准功能。

后面列出的功能名称是预留的并且不能过载。

功能名称	含义
Ncfunc cap read	<p>该功能会将指定地址中的值复制到局部变量中。如果读取成功，返回变量中的值为零。</p> <p>与运算指令相反，该功能在故障时不会取消执行脚本指令。</p> <p>句法:</p> <pre><function name="ncfunc.cap.read" return="error"> lokale variable, "address"</function></pre> <p>示例:</p> <pre><let name="error"></let> <function name="ncfunc.cap.read" return="error"> 3, "drive/cu/p0009"</function> <if> <condition>error != 0</condition> <then> <break /> </then> </if></pre>
Ncfunc cap write	<p>该功能会向指定变量写入一个值。如果写入成功，返回变量中的值为零。</p> <p>与运算指令相反，该功能在故障时不会取消执行脚本指令。</p> <p>句法:</p> <pre><function name="ncfunc.cap.read" return="error"> local variable or constant, "address"</function></pre> <p>示例:</p> <pre><let name="error"></let> <function name="ncfunc.cap.write" return="error"> 0, "drive/cu/p0009"</function> <if> <condition>error != 0</condition> <then> <break /> </then> </if></pre>

功能名称	含义
Ncfunc PI-Service	<p>通过程序实例（PI）服务可将任务传输给 NCK。 如果服务执行成功，该功能会将值 1 返回到返回变量中。</p> <p>刀具表的操作</p> <p>_N_CREATO - 创建刀具 _N_DELETEO - 删除刀具 _N_CREACE - 创建刀沿 _N_DELECE - 删除刀沿</p> <p>零点偏移的激活</p> <p>_N_SETUFR - 将当前的用户框架激活 _N_SETUDT - 将当前的用户数据激活</p> <p>程序段查找</p> <p>_N_FINDBL - 激活查找 _N_FINDAB - 取消程序段查找</p> <p>句法:</p> <pre><function name="ncfunc.pi_service" return="return var"> pi name, var1, var2, var3, var4, var5 </ function></pre> <p>属性:</p> <ul style="list-style-type: none"> • name - 功能名称 • return - 保存执行结果的变量名称 <ul style="list-style-type: none"> - 值 == 1 - 成功执行任务 - 值 == 0 - 任务失败 <p>标签值:</p> <ul style="list-style-type: none"> • pi name - PI 服务的名称（字符串） • var1 到 var5 - PI 专用自变量

功能名称	含义
Ncfunc PI-Service 续	<p>自变量:</p> <ul style="list-style-type: none"> • _N_CREATO var1 - 刀具号 • _N_DELETEO var1 - 刀具号 • _N_CREACE var1 - 刀具号 var2 - 刀沿号 • _N_DELECE var1 - 刀具号 var2 - 刀沿号 • _N_SETUFR 无自变量 • _N_SETUDT var1 - 待激活的用户数据区域 <ul style="list-style-type: none"> - 1 - 刀具补偿数据 - 2 - 有效基本框架 - 3 - 有效可设置框架 • _N_FINDBL var1 - 查找模式 <ul style="list-style-type: none"> - 2 - 带轮廓计算的查找 - 4 - 查找到程序段末尾 - 1 - 不带计算的查找 • _N_FINDAB 无自变量 <p>示例:</p> <ul style="list-style-type: none"> • 创建刀具 - 刀具号 3 <pre data-bbox="730 1542 1444 1602"><function name="ncfunc.pi_service">_T"_N_CREATO", 3</function></pre> <ul style="list-style-type: none"> • 删除刀具 5 的刀沿 1 <pre data-bbox="730 1768 1444 1844"><function name="ncfunc.pi_service">_T"_N_DELECE", 5, 1</function></pre>

1.9 预定义功能

功能名称	含义
Ncfunc display resolution	<p>该功能提供由控制系统确定的浮点数转换规则。提供了字符串变量。</p> <p>另见显示机床数据 MD203 DISPLAY_RESOLUTION 和 MD204 DISPLAY_RESOLUTION_INCH</p> <p>句法:</p> <pre><function name="ncfunc.displayresolution" return="display_res" /></pre> <p>示例:</p> <pre><let name="display_res" type="string"></let> ... <function name="ncfunc.displayresolution" return="display_res" /></pre> <pre><control name = "cdistToGo" xpos = "210" ypos = "156" refvar="nck/Channel/GeometricAxis/ progDistToGo[2]" hotlink="true" height="34" fieldtype="readonly" format="\$\$\$display_res" time="superfast" color_bk="#ffffff"/></pre>
Ncfunc bico to int	<p>该功能将 BICO 格式的字符串转换为整型值。（参见 SINAMICS）。</p> <p>句法:</p> <pre><function name="ncfunc.bicotoint" return="integer variable">bico-string</function></pre> <p>示例:</p> <pre><let name="s_np0480_0" type="string"></let> <let name="i_p0480_0">0</let></pre> <pre><function name="ncfunc.bicotoint" return="i_p0480_0">s_np0480_0</function></pre>

功能名称	含义
Ncfunc int to bico	<p>该功能将整型值转换为 BICO 格式的字符串。（参见 SINAMICS）。</p> <p>句法:</p> <pre><function name="ncfunc.inttobico" return="string variable">integer variable</function></pre> <p>示例:</p> <pre><function name="ncfunc.inttobico" return="s_p0480_0">"drive/dc/p0480[0, D02]"</ function></pre>
Ncfunc is bico str valid	<p>当结果是以 BICO 格式给定的字符串时，该功能会提供零值。（参见 SINAMICS）</p> <p>句法:</p> <pre><function name="ncfunc.isbicostrvalid" return="integer variable">string variable</ function></pre> <p>示例:</p> <pre>... <let name="s_np0480_0" type="string"></let> ... <control name = "cp0480_0" xpos = "402" ypos = "76" hotlink="true" refvar="s_np0480_0" > <property item_data="4001" /> </control> ... <function name="ncfunc.isbicostrvalid" return="valid">cp0480_0</function></pre>

功能名称	含义
Ncfunc password	<p>该功能用来设置或取消口令等级。</p> <ul style="list-style-type: none"> ● 设置口令： 所需口令等级的口令将作为参数给定。 ● 删除口令： 空字符串会取消口令等级。 <p>句法： <code><function name="ncfunc.password">口令 </function></code></p> <p>示例： <code><let name="口令" type="string"></let></code> <code><function name="ncfunc.password" > 口令 </function></code> <code><function name="ncfunc.password" > _T"CUSTOMER" </function></code></p> <p>删除口令： <code><function name="ncfunc.password" > _T"" </function></code></p>
Control form color	<p>该功能提供对话框的文本颜色或背景颜色。</p> <p>区域：</p> <ul style="list-style-type: none"> ● BACKGROUND – 请求背景颜色值 ● TEXT – 请求文本（前景）颜色值 <p>句法： <code><function name="control.formcolor" return="变量">_T"区域"</function></code></p> <p>示例： <code><let name="bk_color"></let></code> <code><function name="control.formcolor" return="bk_color">_T"BACKGROUND"</function></code></p>

功能名称	含义
Control local time	<p>该功能将本地时间复制到带有 7 个数组元素的数组中。</p> <p>变量名称将会用于调用参数。</p> <p>数组元素中保存以下内容：</p> <ul style="list-style-type: none">● 下标 0 - 年份● 下标 1 - 月份● 下标 2 - 星期● 下标 3 - 日● 下标 4 - 小时● 下标 5 - 分钟● 下标 6 - 秒 <p>句法：</p> <pre><function name ="control.localtime">_T"time_array"</function></pre> <p>示例：</p> <pre><!-- index 0 = Year 1 = Month 2 = Day of week 3 = Day 4 = Hour 5 = Minute 6 = Second --> <let name="time_array" dim="7" /> <function name ="control.localtime">_T"time_array"</function></pre>

1.9 预定义功能

功能名称	含义
字符串比较	<p>对两个字符串从书写方式上进行比较。</p> <p>当字符串相同时该功能提供返回值零，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.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>

功能名称	含义
不区分大/小写的字符串比较	<p data-bbox="730 287 1474 363">在不区分大/小写的情况下对两个字符串从书写方式上进行比较。</p> <p data-bbox="730 378 1474 502">当字符串相同时该功能提供返回值零，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 data-bbox="730 570 799 602">参数:</p> <p data-bbox="730 623 887 655">str1 - 字符串</p> <p data-bbox="730 666 943 697">str2 - 比较字符串</p> <p data-bbox="730 763 799 795">句法:</p> <pre data-bbox="730 810 1458 868"><function name="string.icmp" return ="<int var>" > str1, str2 </function></pre> <p data-bbox="730 917 799 949">示例:</p> <pre data-bbox="730 959 1078 991"><let name="rval">0</let></pre> <pre data-bbox="730 1023 1458 1208"><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 data-bbox="730 1225 799 1257">结果:</p> <p data-bbox="730 1278 820 1310">rval= 0</p>

1.9 预定义功能

功能名称	含义
字符串左侧	<p>该功能提取字符串 1 中第一个的 nCount 字符并将其复制到返回变量中。</p> <p>参数: str1 - 字符串 nCount - 字符数量</p> <p>句法: <code><function name="string.left" return="<result string">"> str1, nCount </function></code></p> <p>示例: <code><let name="str1" type="string">A brown bear hunts a brown dog.</let> <let name="str2" type="string"></let></code> <code><function name="string. left" return="str2"> str1, 12 </function></code></p> <p>结果: <code>str2="A brown bear"</code></p>

功能名称	含义
字符串右侧	<p>该功能提取字符串 1 中最后一个 nCount 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:<pre><let name="str1" type="string">A brown bear hunts a brown dog.</let> <let name="str2" type="string"></let></pre><pre><function name="string. right " return="str2"> str1, 10 </function></pre><p>结果:</p><pre>str2="brown dog."</pre></p>

功能名称	含义
字符串中间	<p>该功能从下标 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: str1 - 字符串 iFirst - 起始下标 nCount - 字符数量</p> <p>句法: <code><function name="string.middle" return="<result string">"> str1, iFirst, nCount </function></code></p> <p>示例: <code><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></code></p> <p>结果: str2="brown"</p>
字符串长度	<p>该功能可以提供字符串的字符数量。</p> <p>参数: str1 - 字符串</p> <p>句法: <code><function name="string.length" return="<int var">"> str1 </function></code></p> <p>示例: <code><let name="length">0</let> <let name="str1" type="string">A brown bear hunts a brown dog.</let> <function name="string.length" return="length"> str1 </function></code></p> <p>结果: length = 31</p>

功能名称	含义
替代字符串	<p>该功能可以使用新的字符串替代所有查找到的字符串部分。</p> <p>参数:</p> <p>string - 字符串变量 find string - 要替代的字符串 new string - 新字符串</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> <pre>str1 = "A brown bear hunts a big salmon!"</pre>
删除字符串	<p>该功能可以删除所有查找到的字符串部分。</p> <p>参数:</p> <p>string - 字符串变量 remove string - 要删除的字符串部分</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>

功能名称	含义
插入字符串	<p>该功能可以在指定的下标中插入一个字符串。</p> <p>参数:</p> <ul style="list-style-type: none">string - 字符串变量index - 下标（从零开始）insert string - 要插入的字符串 <p>句法:</p> <pre><function name="string.insert"> string, index, insert string </function></pre> <p>示例:<pre><let name="str1" type="string">A brown bear hunts. </let> <let name="str2" type="string">a brown dog</let> <function name="string.insert" > str1, 19, str2</ function></pre><p>结果:</p><pre>str1 = "A brown bear hunts a brown dog"</pre></p>

功能名称	含义
删除字符串字符	<p>该功能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 start index - 起始下标（从零开始） nCount - 要删除的字符数量</p> <p>句法:</p> <pre data-bbox="730 672 1385 729"><function name="string.delete"> string, start index , nCount </function></pre> <p>示例:</p> <pre data-bbox="730 825 1455 883"><let name="str1" type="string">A brown bear hunts. </let></pre> <pre data-bbox="730 921 1385 978"><function name="string.delete" > str1, 2, 5</function></pre> <p>结果:</p> <pre data-bbox="730 1049 991 1076">str1 = "A bear hunts"</pre>

1.9 预定义功能

功能名称	含义
查找字符串	<p>该功能可以在所传输的字符串中查找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，则该功能会提供到第一个字符的下标（从零开始），否则为 -1。</p> <p>参数:</p> <p>string - 字符串变量</p> <p>find string - 要查找的字符串</p> <p>startindex = 起始下标（可选）</p> <p>句法:</p> <pre data-bbox="689 817 1390 874"><function name="string.find" return="<int val>"> str1, find string </function></pre> <p>示例:</p> <pre data-bbox="689 966 1422 1123"><let name="index">0</let> <let name="str1" type="string">A brown bear hunts a brown dog. </let> <function name="string.find" return="index"> str1, _T"brown" </function></pre> <p>结果:</p> <p>Index = 2</p> <p>或</p> <pre data-bbox="689 1285 1417 1342"><function name="string.find" return="index"> str1, _T"brown", 1 </function></pre>

功能名称	含义
反向查找字符串	<p>该功能可以在所传输的字符串中查找与部分字符串一致的最后一处地方。</p> <p>如果找到了部分字符串，则该功能会提供到第一个字符的下标（从零开始），否则为 -1。</p> <p>参数:</p> <p>string - 字符串变量 find string - 要查找的字符串 startindex = 起始下标（可选）</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> <p>或</p> <pre><function name="string.reversefind" return="index"> str1, _T"brown", 10 </function></pre> <p>结果:</p> <p>Index = 2</p>

功能名称	含义
字符串左侧删除	<p>该功能可以从字符串中删除打头的空格。</p> <p>参数: str1 - 字符串变量</p> <p>句法: <code><function name="string.trimleft" > str1 </function></code></p> <p>示例: <code><let name="str1" type="string"> test trim left</let></code> <code><function name="string.trimleft" > str1 </function></code></p> <p>结果: <code>str1 = "test trim left"</code></p>
字符串右侧删除	<p>该功能可以从字符串中删除后续的空格。</p> <p>参数: str1 - 字符串变量</p> <p>句法: <code><function name="string.trimright" > str1 </function></code></p> <p>示例: <code><let name="str1" type="string"> test trim right</let></code> <code><function name="string.trimright" > str1 </function></code></p> <p>结果: <code>str1 = "test trim right"</code></p>

功能名称	含义
正弦	<p>该功能可以计算所传输值（单位度）的正弦值。</p> <p>参数: double - 角度</p> <p>句法: <code><function name="sin" return="<double val>"> double </function></code></p> <p>示例: <code><let name= "sin_val" type="double"></let> <function name="sin" return="sin_val"> 20.0 </ function></code></p>
余弦	<p>该功能可以计算所传输值（单位度）的余弦值。</p> <p>参数: double - 角度</p> <p>句法: <code><function name="cos" return="<double val>"> double </function></code></p> <p>示例: <code><let name= "cos_val" type="double"></let> <function name="cos" return="cos_val"> 20.0 </ function></code></p>

1.9 预定义功能

功能名称	含义
正切	<p>该功能可以计算所传输值（单位度）的正切值。</p> <p>参数: double - 角度</p> <p>句法: <function name="tan" return="<double val>"> double </function></p> <p>示例: <let name= "tan_val" type="double"></let> <function name="tan" return="tan_val"> 20.0 </function></p>
反正弦	<p>该功能可以计算所传输值（单位度）的反正弦值。</p> <p>参数: double - x 取值范围 $-\pi/2$ 至 $+\pi/2$</p> <p>句法: <function name="arcsin" return="<double val>"> double </function></p> <p>示例: <let name= "arcsin_val" type="double"></let> <function name="arcsin" return=" arcsin_val"> 20.0 </function></p>

功能名称	含义
反余弦	<p>该功能可以计算所传输值（单位度）的反余弦值。</p> <p>参数: double - x 取值范围 $-\pi/2$ 至 $+\pi/2$</p> <p>句法: <code><function name="arccos" return="<double val>"> double </function></code></p> <p>示例: <code><let name= "arccos_val" type="double"></let> <function name="arccos" return=" arccos_val"> 20.0 </function></code></p>
反正切	<p>该功能可以计算所传输值（单位度）的反正切值。</p> <p>参数: double - 反正切 y/x</p> <p>句法: <code><function name="arctan" return="<double val>"> double </function></code></p> <p>示例: <code><let name= "arctan_val" type="double"></let> <function name="arctan" return="arctan_val"> 20.0 </function></code></p>

1.9 预定义功能

功能名称	含义
装载 DLL	<p>该功能可以在存储器中装载其他的用户 DLL。</p> <p>参数:</p> <p>dll_name - DLL 名称 class_name - 功能级的名称</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 功能	<p>该功能可以调用用户 DLL 中的功能。将参数 ID 后面列出的所有参数传输给所调用的功能。</p> <p>参数:</p> <p>class_name - 功能级的名称 id - 功能 parameter - 最多 7 个功能参数（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>
文件处理	

功能名称	含义
装载文件	<p>该功能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</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>
写文件	<p>该功能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</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>

1.9 预定义功能

功能名称	含义
删除文件	<p>该功能可以从目录中删除指定的文件。</p> <p>参数: programe - 文件名</p> <p>句法: <code><function name="doc.remove" > programe </function></code></p> <p>示例: <code><function name="doc.remove">_T"\mpf\test.mpf" </function></code></p>
存在	<p>如文件存在，则功能提供值 1。</p> <p>参数: programe - 文件名</p> <p>句法: <code><function name="doc.exist" return="<int_var>" > programe </function></code></p> <p>示例: <code><let name ="exist">0</let></code></p> <p><code><function name="doc.exist" return="exist">_T"\mpf\test.mpf" </function></code></p>
NC 程序选择	<p>该功能可以选择将要处理的程序。程序必须保存在 NC 文件系统中。</p> <p>参数: programe - 文件名</p> <p>句法: <code><function name="ncfunc.select"> programe </function></code></p> <p>示例: <code><function name="ncfunc.select"> _T"\mpf\test.mpf" </function></code></p>

功能名称	含义
设置位	<p>该功能用于设置给定变量的各个位。</p> <p>可以置位或复位。</p> <p>句法:</p> <pre><function name="ncfunc.bitset" refvar="address" value="set/reset" > bit0, bit1, ... bit9 </function></pre> <p>属性:</p> <p>refvar - 指定要写入的位组合所在变量的名称</p> <p>value – 位的取值为 0 和 1</p> <p>值:</p> <p>作为功能值，位号从 0 开始给定。</p> <p>每次调用最多可以修改 10 个位。</p> <p>示例:</p> <pre><function name="ncfunc.bitset" refvar="nck/Channel/Parameter/R[1]" value="1" > 0, 2, 3, 7 </function></pre> <pre><function name="ncfunc.bitset" refvar="nck/Channel/Parameter/R[1]" value="0" > 1, 4 </function></pre>

功能名称	含义
删除控制项	<p>该功能可删除指定的控制项。</p> <p>句法:</p> <pre><function name="control.delete"> control name </function></pre> <p>或</p> <pre><function name="ncfunc.bitset" refvar="\$MN_USER_DATA_HEX" value="1" > 0, 2, 3, 7 </ function></pre> <p>属性:</p> <p>name – 功能名称</p> <p>值:</p> <p>control name – 控制项的名称</p> <p>示例:</p> <pre><function name="<control.delete>"> _T"my_editfield" </function></pre>

功能名称	含义
添加项	<p>该功能可在列表末尾添加一个新元素。</p> <p>提示： 该功能只能用于控制项类型“listbox”和“graphicbox”。</p> <p>句法： <function name="control.additem"> control name, item </function></p> <p>属性： name – 功能名称</p> <p>值： control name – 控制项名称 item - 要添加的表达式 itemdata - 整数值；由用户确定</p> <p>示例： <let name ="itemdata">1</let> <op> item_string = _T"text1" </op> <function name="control.additem">_T"listbox1", item_string, itemdata </function></p>

1.9 预定义功能

功能名称	含义
插入项	<p>该功能可在指定位置添加一个新元素。</p> <p>提示： 该功能只能用于控制项类型“listbox”和“graphicbox”。</p> <p>句法： <function name="control.insertitem"> control name, index, item, itemdata </function></p> <p>属性： name – 功能名称</p> <p>值： control name – 控制项名称 index – 位置从零开始 item - 要添加的表达式 itemdata - 整数值；由用户确定</p> <p>示例： <let name ="itemdata">1</let> <op> item_string = _T"text2" </op> <function name="control.insertitem">_T"listbox1", 1, item_string, itemdata </function></p>

功能名称	含义
删除项	<p>该功能可删除指定位置上的元素。</p> <p>提示： 该功能只能用于控制项类型“listbox”和“graphicbox”。</p> <p>句法： <function name="control.deleteitem"> control name, index </function></p> <p>属性： name – 功能名称</p> <p>值： control name – 控制项名称 index- 下标从 0 开始</p> <p>示例： <function name="control.deleteitem">_T"listbox1", 1</function></p>

功能名称	含义
载入项	<p>该功能可向控制项中添加一个表达式列表。</p> <p>该功能只能用于控制项类型“listbox”和“graphicbox”。</p> <p>句法:</p> <pre><function name="control.loaditem"> control name, list </function></pre> <p>属性:</p> <p>name – 功能名称</p> <p>值:</p> <p>control name – 控制项名称</p> <p>list - 字符串变量</p> <p>列表结构:</p> <p>列表包含一定数量的表达式，它们之间用 <code>\n</code> 分隔。</p> <p>示例:</p> <pre><let name="item_string" type="string"></let> <let name="plotlist" type="string"></let> <print name="item_string" text="p; %f; %f; %f; %f \n">s_z, s_x</print> <op>plotlist = plotlist + item_string</op> <print name="item_string" text="l; %f; %f; %f; %f\n">s_z, s_x, e_z, e_x </print> <op>plotlist = plotlist + item_string</op> <op> s_x = e_x </op> <op> s_z = e_z</op> <op> e_x = s_x + 10 </op> <op> e_z = s_z - 100 </op> <print name="item_string" text="l; %f; %f; %f; %f\n">s_z, s_x, e_z, e_x </print> <op>plotlist = plotlist + item_string</op> <function name="control.loaditem">_T"gbox", plotlist</function></pre>

功能名称	含义
清空	<p>该功能可清除指定列表框或图形框的内容。</p> <p>句法: <code><function name="control.empty"> control name, </function></code></p> <p>属性: name – 功能名称</p> <p>值: control name – 控制项名称</p> <p>示例: <code><function name=" control.empty">_T"listbox1"</function></code></p>
获取活动项	<p>该功能可提供当前正处在输入状态的控制项的名称。</p> <p>句法: <code><function name="control.getfocus" return="focus_name" /></code></p> <p>属性: name – 功能名称 return – 给定字符串变量，将控制项的名称复制到其中。</p> <p>示例: <code><let name="focus_field" type="string"></let> <function name="control.getfocus" return="focus_field"/></code></p>

1.9 预定义功能

功能名称	含义
设置活动项	<p>该功能可将指定的控制项设置到输入状态。</p> <p>控制项的名称将以功能的文本表达方式传输。</p> <p>句法:</p> <pre><function name="control.setfocus"> control name </function></pre> <p>属性:</p> <p>name – 功能名称</p> <p>值:</p> <p>control name – 控制项的名称</p> <p>示例:</p> <pre><function name="control.setfocus" ">_T"listbox1"</function></pre>
获取光标选择	<p>该功能可提供列表框中光标所在的下标。</p> <p>控制项的名称将以功能的文本表达方式传输。</p> <p>句法:</p> <pre><function name="control.getcurssel" retvar="var"> control name </function></pre> <p>示例:</p> <pre><let name>="index"></let> <function name="control.getcurssel" ">_T"listbox1"</function></pre>

功能名称	含义
设置光标选择	<p>该功能可在列表框中将光标放置在相应的行中。 控制项的名称将以功能的文本表达方式传输。</p> <p>句法: <code><function name="control.setcurssel" > control name, index</function></code></p> <p>示例: <code><let name="index">2</let> <function name="control.setcurssel" >_T"listbox1",index</function></code></p>
获取项	<p>该功能可将列表框中所选择行的内容复制到指定的变量中。 提供了字符串变量作为参考变量。 控制项的名称将以功能的文本表达方式传输。</p> <p>句法: <code><function name="control.getitem" return="var"> control name, index </function></code></p> <p>示例: <code><let name="index">2</let> <let name="item" type="string"></let> <function name="control.getitem" return="item" >_T"listbox1",index</function></code></p>

1.9 预定义功能

功能名称	含义
获取项数据	<p>该功能可将列表框中用户自定义的元素值复制到指定的变量中。</p> <p>在 EDIT 控制项中，该功能可将用户自定义的值 (<code>item_data</code>) 复制到指定的变量中。</p> <p>提供了整型变量作为参考变量。</p> <p>控制项的名称将以功能的文本表达方式传输。</p> <p>句法:</p> <pre><function name="control.getitemdata" return="var"> control name, index </function></pre> <p>示例:</p> <pre><let name>="index">2</let> <let name>="itemdata"></let></pre> <pre><function name="control.getitemdata" return="itemdata" ">_T"listbox1",index</function></pre>
Abs	<p>该功能提供给定数值的绝对值。</p> <p>句法:</p> <pre><function name="abs" return="var"> value </ function></pre>
SDEG	<p>该功能可将给定值换算为度。</p> <p>句法:</p> <pre><function name="sdeg" return="var"> value </ function></pre>
SRAD	<p>该功能可将给定值换算为弧度。</p> <p>句法:</p> <pre><function name="srad" return="var"> value </ function></pre>
SQRT	<p>该功能可以计算给定值的根。</p> <p>句法:</p> <pre><function name="sqrt" return="var"> value </ function></pre>

功能名称	含义
ROUND	<p>该功能可以将所传输的数值的小数位保留到指定的位数。如果未指定要保留的小数点后的位数，则该功能会保留到小数点后第一位。</p> <p>句法: <code><function name="round" return="var"> value, nDecimalPlaces </function></code></p>
FLOOR	<p>该功能提供小于等于所传输值的最大的整数值。</p> <p>句法: <code><function name="floor" return="var"> value </function></code></p>
CEIL	<p>该功能提供大于等于所传输值的最小的整数值。</p> <p>句法: <code><function name="ceil" return="var"> value </function></code></p>
LOG	<p>该功能可以计算给定值的对数。</p> <p>句法: <code><function name="log" return="var"> value </function></code></p>
LOG10	<p>该功能可以计算给定值的十进制对数。</p> <p>句法: <code><function name="log10" return="var"> value </function></code></p>
POW	<p>该功能计算“a^b”的值。</p> <p>句法: <code><function name="pow" return="var"> a, b </function></code></p>
MIN	<p>该功能会比较所传输的值并将较小的值返回。</p> <p>句法: <code><function name="min" return="var"> value1, value2 </function></code></p>

1.9 预定义功能

功能名称	含义
MAX	<p>该功能会比较所传输的值并将较大的值返回。</p> <p>句法: <function name="max" return="var"> value1, value2 </function></p>
RANDOM	<p>该功能提供伪随机数。</p> <p>句法: <function name="random" return="var" </function></p>

功能名称	含义
MMC	<p>功能</p> <p>通过 MMC 指令，可从 HMI 上的零件程序中显示用户定义的对话框窗口（对话框屏幕）。对话框窗口的式样由纯文本配置确定（目录 F:\appl（制造商驱动器）下的 XML 文件），HMI 系统软件保持不变。</p> <p>句法</p> <p>MMC（“操作区、指令、XML 脚本文件、菜单名称、预留、预留、显示时间或者应答变量、预留”，“应答模式”）</p> <p>含义</p> <ul style="list-style-type: none"> ● MMC 在 HMI 上从零件程序中调用交互式对话框。 ● EasyXml 应从零件程序中开始的用户对话框的标识。 ● XML_ON bzw. XML_OFF 指令：图形选择或图形撤销 ● XML 文件 XML 脚本文件的名称 ● 菜单 管理待显示对话框的菜单标签的名称 ● 预留 预留于 HMI 操作 ● 预留 预留于 HMI 操作 ● 时间 应答模式“N”中的对话框显示时间 ● 预留 预留于 HMI 操作 ● 应答模式 “S”同步，通过软键“OK”应答 “N”异步，在规定的时间内关闭对话框

功能名称	含义
MMC 续	<p>同步调用示例:</p> <p>NC 指令 MMC("EASYXML oder CYCLES", XML_ON, mmc_cmd.xml, cmd1,,,,,"S") Datei: mmc_cmd.xml</p> <pre> <menu name = "cmd1"> <open_form name = "cmd1_form" /> <softkey_ok> <close_form /> </softkey_ok> </menu> <form name = "cmd1_form" xpos ="12" ypos="100" width="500" height="240"> <init> </init> <paint> </paint> </form> </pre> <p>异步调用示例 (无应答):</p> <p>NC 指令 MMC("EASYXML oder CYCLES", PICTURE_ON, mmc_cmd.xml, cmd1,,10,,,"N") Datei: mmc_cmd.xml <menu name = "cmd1"></p> <pre> <open_form name = "cmd1_form" /> <softkey_ok> <close_form /> </softkey_ok> </menu> <form name = "cmd1_form" xpos ="12" ypos="100" width="500" height="240"> <init> </init> <paint> </paint> </form> </pre>

功能名称	含义
查找字符串	<p>该功能可以在所传输的字符串中查找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，则该功能会提供到第一个字符的下标（从零开始），否则为 -1。</p> <p>参数:</p> <p>string - 字符串变量</p> <p>findstring - 要查找的字符串</p> <p>startindex - 起始下标（可选）</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> <let name="str1" type="string">A brown bear hunts a brown dog. </let> <function name="string.find" return="index"> str1, _T"brown" </function></pre> <p>结果:</p> <p>下标 = 2</p> <p>或</p> <pre><function name="string.find" return="index"> str1, _T"brown", 1 </function></pre>

功能名称	含义
反向查找字符串	<p>该功能可以在所传输的字符串中查找与部分字符串一致的最后一处地方。</p> <p>如果找到了部分字符串，则该功能会提供到第一个字符的下标（从零开始），否则为 -1。</p> <p>参数:</p> <p>string - 字符串变量</p> <p>find string - 要查找的字符串</p> <p>startindex - 起始下标（可选）</p> <p>句法:</p> <pre><function name="string.reversefind" return="<interval>"> 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>索引 = 21</p> <p>或</p> <pre><function name="string.reversefind" return="index"> str1, _T"brown" 10 </function></pre> <p>结果:</p> <p>索引 = 2</p>
对话框	

功能名称	含义
对话框程序选择	<p>该功能可以打开一个对话框用于选择零件程序。</p> <p>选定的程序名称被复制到指定的参考变量中。</p> <p>只能在软键标签内调用该对话框。</p>  <p>句法:</p> <pre><function name= "doc.fileselect" return="string var" /></pre> <p>示例:</p> <pre><let name = "program_content" type="string" > </let> <let name = "curr_program" type="string" > </let> <function name= "doc.fileselect" return="curr_program" /> <function name="doc.readfromfile " return=" program_content" > curr_program </function></pre>

功能名称	含义
对话框程序模拟	<p>该功能可以将程序模拟作为弹出式对话框打开。</p> <p>如果给定了一个程序名称，会自动执行程序选择。否则，模拟将显示当前所选择程序的处理过程。</p> <p>只能在软键标签内调用该对话框。</p> <p>使用软键功能“返回”，可以回到前一对话框。</p>  <p>句法:</p> <pre><function name= "ncfunc.progsimulation"> prog name </function></pre> <p>示例: <pre><let name = "curr_program" type="string" > </let> <function name= "doc.fileselect" return="curr_program" /> <function name= "ncfunc.progsimulation"> curr_program </function></pre> </p>

创建调试对话框

2.1 功能概述

使用目的

使用“调试向导”能够非常简单地调试、激活、取消激活或测试附加设备。控制系统中有一张显示所有可用设备和设备状态的清单。系统可以最多管理 64 个设备。

设备的激活或取消由软键操作实现。

“调试向导”功能在操作区域<系统>中使用。



图 2-1 “调试向导”软键激活时，系统操作区域的初始画面

2.1 功能概述

配置

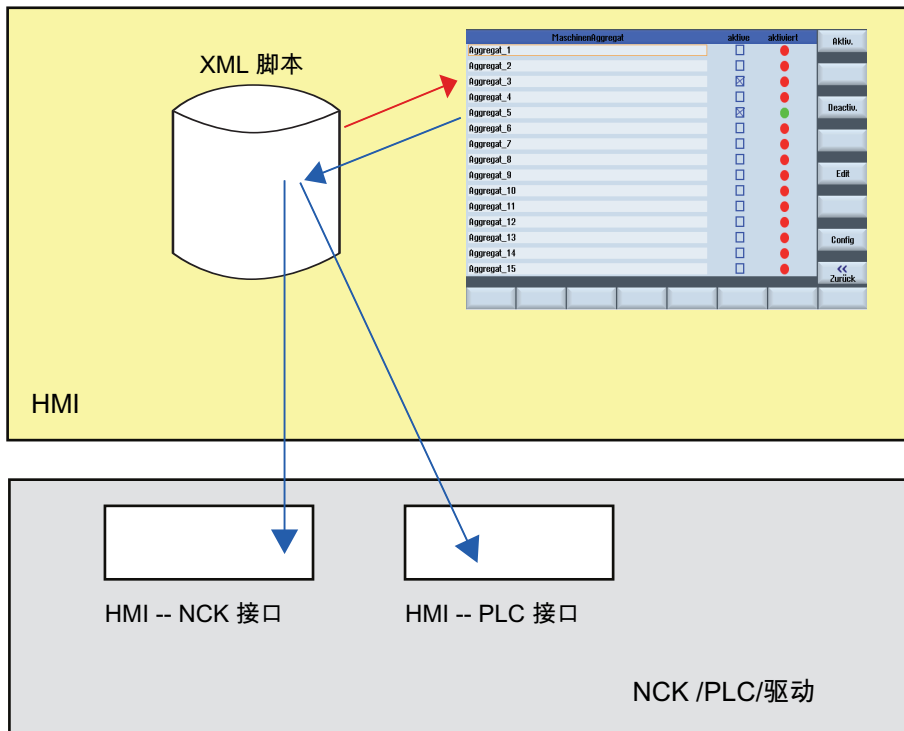


图 2-2 “调试向导” 的工作原理

机床制造商首先定义以下功能，才可以使用“调试向导”：

- **PLC ↔ HMI 的接口**
可选设备是通过操作界面和 PLC 之间的接口来管理的。
- **脚本编辑**
机床制造商应在一个指令脚本中定义如何测安装、激活、取消和测试设备的流程。
- **参数窗口（可选）**
参数窗口用于显示在脚本文件中定义的设备信息。

文件保存

“调试向导”的相关文件保存在系统 CF 卡的目录“dvm”（机床制造商）下。

文件	名称	目标目录
文本文件	oem_aggregate_xxx .ts	card\oem\sinumerik\hmi\lmg
脚本文件	agm.xml	card\oem\sinumerik\hmi\dvm

文件	名称	目标目录
存档文件	任意	card\oem\sinumerik\hmi\dvm \archives
PLC 用户程序	任意	PLC

2.2 PLC 用户程序中的配置

载入配置

首先将制造商创建的配置和脚本文件、文本文件一起传送到控制系统的制造商目录中。另外，还需要载入对应的 PLC 用户程序。

设备的编程

操作组件和 PLC 之间是通过 PLC 用户程序中的数据块 VB99050000 通讯的，该数据块中有 128 个字预留用于设备的管理。

首先从设备 1 开始，指定 PLC 数据字：

数据块	设备名称
DB9905.DBB0	设备 1
DB9905.DBB4	设备 2
...	...
DB9905.DBB192	设备 49
DB9905.DBB196	设备 50

每个设备可以占用四个字节，含义如下：

字节	位	说明
0	0	== 1 设备已经启动（HMI 反馈信息）
	1	== 1 请激活设备（HMI 请求）
	2	== 1 请取消激活设备（HMI 请求）
	3-7	预留
1	0-7	预留
2	0	== 1 设备已激活（PLC 反馈）
	1	== 1 设备出错
	2-7	预留
3	0-7	设备的唯一标识

文件保存

文件保存在系统 CF 的目录“oem”（机床制造商）和“oem_i”（个人）下。

文件	名称	目标目录
文本文件	oem_aggregate_xxx .ts	/oem/sinumerik/hmi/lng/ /oem_i/sinumerik/hmi/lng/
脚本文件	agm.xml	/oem/sinumerik/hmi/dvm /oem_i/sinumerik/hmi/dvm
存档文件	任意	/oem/sinumerik/hmi/dvm/archives /oem_i/sinumerik/hmi/dvm/archives
PLC 用户程序	任意	PLC

一般操作顺序

机床制造商首先执行以下操作，提供所需数据：

1. 创建 PLC 用户程序，该程序中包含了由 PLC 控制的设备。
2. 调试“标准机床”，接着将数据备份到批量调试文档中。
3. 安装、调试设备，将数据作为差别批量调试文档读出。

说明

修改机床配置

需要修改驱动机床数据时，首先在控制系统中执行修改。然后在所有设置和装置上再次修改。

添加轴

在需要为机床增加机床数据时，请按照固定顺序添加驱动对象(DO)，因为批量调试文档包含了机床制造商参考机床的结构，一旦顺序被改动，便无法使用。

我们推荐为控制系统组件选择以下设置：

- NC 数据
- PLC 数据
- 驱动数据
 - ACX 格式（二进制）

2.3 操作界面上的显示画面

操作界面上的窗口

以下窗口用于“调试向导”功能：

- 控制系统提供了一个**自定义窗口**，其中可显示可用设备。
- 如果还没有开展首次调试，系统会打开**调试窗口**。

如果已经为设备编写了一个调试流程（XML 指令：“START_UP”），而设备还没有经过调试，系统会启动调试流程。

此时，在读取脚本文件中保存的批量调试文档前，会完整备份数据。

出错时调试人员可以决定，是否取消调试，或手动排除可能出现在机床配置中的错误。

- 按下“取消”，可以中断调试。系统随后会将取消调试数据的备份。

调试成功结束后需要断开机床时，可以编写 XML 指令“POWER_OFF”，控制系统上便会输出对应的信息。

2.4 创建和语言相关的文本

文本文件结构

包含和语言相关的文本的 xml 文件应以 UTF8 格式创建:

示例 oem_aggregate_eng.ts

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
  <context>
    <name>EASY_EXTEND</name>
    <message>
      <source>DEVICE_ONE</source>
      <translation>Device one</translation>
    </message>
    <message>
      <source>DEVICE_TWO</source>
      <translation>Device two</translation>
    </message>
  </context>
</TS>
```

示例 oem_aggregate_chs.ts

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE TS>
<TS>
  <context>
    <name>EASY_EXTEND</name>
    <message>
      <source>DEVICE_ONE</source>
      <translation>第一个设备</translation>
    </message>
    <message>
      <source>DEVICE_TWO</source>
      <translation>第二个设备</translation>
    </message>
  </context>
</TS>
```

2.5 功率部件应用示例

激活驱动对象

需要激活的驱动对象已经过调试，并由机床制造商重新取消激活，用于将轴作为选件投入市场。

必须执行以下步骤激活轴：

- 通过 p0105 激活驱动对象。
- 2. 在通道机床数据中激活轴。
- 通过 p0971 备份驱动机床数据。
- 等待数据写入。
- 重启 NCK 和驱动。

编程：

```
<DEVICE>
  <list_id>1</list_id>
  <name> "激活驱动" </name>

  <SET_ACTIVE>
    <data name = "drive/dc/p105[D05]">1</data>
    <data name = "$SMC_AXCONF_MACHAX_USED[4]">5</data>
    <data name = "drive/dc/p971[D05]">1</data>
    <while>
      <condition> "drive/dc/p971[D05]" !=0 </condition>
    </while>
    <control_reset resetnc ="true" resetdrive = "true"/>
  </SET_ACTIVE>

  <SET_INACTIVE>
    <data name = "drive/dc/p105[D05]">0</data>
    <data name = "$SMC_AXCONF_MACHAX_USED[4]">0</data>
    <data name = "drive/dc/p971[D05]">1</data>
    <while>
      <condition> "drive/dc/p971[D05]" !=0 </condition>
    </while>
    <control_reset resetnc ="true" resetdrive = "true"/>
  </SET_INACTIVE>
```

```
</DEVICE>
```

激活由 PLC 控制的设备

设备通过输出字节 10 应答，并且通过输出字节 9 向 PLC 发送运行就绪反馈信息。

输出字节被设置为固定的编码用于激活设备。之后 WHILE 循环等待设备运行就绪。

编程：

```
<SET_ACTIVE>
  <DATA name = "plc/qb10"> 8 </DATA>
  <while>
    <condition> "plc/ib9" !=1 </condition>
  </while>
</SET_ACTIVE>
```

2.6 脚本语言

说明

在功能 创建用户对话框 (页 5) 中描述的所有脚本元素形成了“调试向导”功能的基础。为管理附加装置，还定义了其它的脚本元素。

脚本的程序段落

脚本由以下段落组成：

- “调试向导” - 框架
- 用于确定设备可执行操作的框架
- 用于设备的标签
- 用于设备调试的标签
- 用于设备激活的标签
- 用于设备取消的标签

2.6 脚本语言

- 用于设备测试的标签
- 参数窗口的标签

下面的章节详细说明各个标签。

说明

标签 <tag>	含义
AGM	“调试向导”功能的标签
DEVICE	用于描述设备的标签。 属性: <ul style="list-style-type: none"> • option_bit 设备指定有一个固定的位编号，用于管理选项。
NAME	该标签定义在窗口中显示的设备名称。 如果使用参考文本，窗口中会显示该标签下定义的文本。
START_UP	该标签包含调试设备所需流程的说明。
SET_ACTIVE	该标签包含激活设备所需流程的说明。 属性: <ul style="list-style-type: none"> • timeout 该属性指定超时时间，单位秒。如果脚本在该时间后仍未结束，系统会中断处理。
SET_INACTIVE	该标签包含取消设备所需流程的说明。 属性: <ul style="list-style-type: none"> • timeout 该属性指定超时时间，单位秒。如果脚本在该时间后仍未结束，系统会中断处理。
TEST	该标签包含设备测试所需的指令。 属性: <ul style="list-style-type: none"> • timeout 该属性指定超时时间，单位秒。如果脚本在该时间后仍未结束，系统会中断处理。
UID	用于区分 PLC ↔ HMI 接口中设备的唯一数字标识码。
VERSION	用于版本的标签

应答功能执行 (negative)

通过自动提供的变量“\$actionresult”系统可以向 XML-Parser 传递一个“negative”的执行结果。该值为零时，Parser 会中断执行功能。

示例

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE AGM>
<AGM>                                “调试向导”的标签
<DEVICE>
  <NAME> 设备 1 </NAME>                用于设备的标签
  <START_UP>                            用于设备调试的标签
  ...
  </START_UP>
  <SET_ACTIVE>                          用于设备激活的标签
  ...
  </SET_ACTIVE>
  <SET_INACTIVE>                        用于设备取消的标签
  ...
  </SET_INACTIVE>
  <TEST>                                用于设备测试的标签
  ...
  </TEST>
</DEVICE>
...
</AGM>
```

2.6.1 CONTROL_RESET

说明

该标签可以重新启动一个或多个控制系统组件。只有当控制系统恢复周期运行时，才可以继续执行脚本。

编程

标签:	CONTROL_RESET
句法:	<code><CONTROL_RESET resetnc="TRUE" /></code>
属性:	<code>resetnc="true"</code> 重新启动 NC 组件。 <code>resetdrive="true"</code> 重新启动驱动组件。

2.6.2 FILE

说明

该标签可以读入或创建标准文档。

- 读入文档：
请输入文档名称。
- 创建文档：
定义了属性 `create="true"` 后，该功能会以指定名称创建一个标准文档(*.arc)，并保存在目录 `.../dvm/archives` 中。

编程

标签:	FILE
句法:	<code><file name ="<文档名称>" /></code> <code><file name ="<文档名称>" create="true" class="<数据类别>" group="<范围>" /></code>
属性:	<code>name</code> 用于文件名称的标签

create	以指定的名称在目录 .../dvm/archives/ 下创建一个调试文档。
group	指定文档包含的数据组。如果需要备份多个数据组，请用空格隔开。 文档可以包含以下数据组： <ul style="list-style-type: none"> • NC • PLC • HMI • DRIVES

示例

```
<!-- 创建数据级文档 -->
<file name="user.arc" create="true"
group="nc plc hmi" />

<!--将文档读入控制系统-->
<file name="user.arc" />
```

2.6.3 OPTION_MD

说明

该标签允许重新定义选件机床数据。在出厂时，系统使用的是 MD14510 \$MN_USER_DATA_INT[0] ~ \$MN_USER_DATA_INT[3]。

如果由 PLC 用户程序管理选件，请在一个数据块或 GUD 中提供对应的数据字。

数据按位管理。从位 0 开始，各个位按照设备清单上的顺序固定指定给各个设备，即：位 0 指定给设备 1；位 1 指定给设备 2，如此类推。如果需要管理多于 16 个设备，请通过区域索引将地址标识指定给设备组 1-3。

说明

换算取值范围

MD14510 \$MN_USER_DATA_INT[i] 的取值范围是 -32768 ~ +32767。为了在机床数据窗口中按位使能各个设备，需要将位组换算成十进制格式。

编程

标签:	OPTION_MD				
句法:	区域 0: <code><option_md name = "数据的地址标识" /></code> 或者是 <code><option_md name = "数据的地址标识" index= "0"/></code> 区域 1~3: <code><option_md name = "数据的地址标识" index= "区域索引"/></code>				
属性:	<table> <tr> <td>name</td> <td>地址标识, 例如: \$MN_USER_DATA_INT[0]</td> </tr> <tr> <td>index</td> <td>用于区域索引的标签: 0 (缺省设置): 设备 1 ~ 16 1: 设备 17 ~ 32 2: 设备 33 ~ 48 3: 设备 49 ~ 64</td> </tr> </table>	name	地址标识, 例如: \$MN_USER_DATA_INT[0]	index	用于区域索引的标签: 0 (缺省设置): 设备 1 ~ 16 1: 设备 17 ~ 32 2: 设备 33 ~ 48 3: 设备 49 ~ 64
name	地址标识, 例如: \$MN_USER_DATA_INT[0]				
index	用于区域索引的标签: 0 (缺省设置): 设备 1 ~ 16 1: 设备 17 ~ 32 2: 设备 33 ~ 48 3: 设备 49 ~ 64				

2.6.4 PLC_INTERFACE

说明

该标签允许重新定义 PLC ↔ HMI 接口。系统提供 128 个可编址的数值。

缺省设置: DB9905 (VB99050000)

编程

标签:	PLC_INTERFACE
句法:	<code><plc_interface name = "数据的地址标识" /></code>
属性:	name 地址标识, 例如“plc/mb170”

示例: plc/mb170

2.6.5 POWER_OFF

说明

该标签用于设定一条要求操作员关闭机床的信息。显示文本固定保存在系统中。

编程

标签: **POWER_OFF**
句法: <power_off />
属性: --

2.6.6 WAITING

说明

该标签用于指定：在 NC 或驱动复位后，等待各个组件重新启动。

编程

标签: **WAITING**
句法: <WAITING WAITINGFORNC ="TRUE" />
属性: waitingfornc="true" 等待 NC 重新启动。
 waitingfordrive="true" 等待驱动重新启动。

2.6.7 用于窗口的 XML 标签

用于设置参数的窗口

可以为每个设备设计一个窗口，用于设置或输出运行时间所需的附加参数。按下软键“附加参数”后，会显示该窗口。

在创建对话框时可以使用章节 创建用户对话框 (页 5) 中描述的所有脚本元素。

示例

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE AGM>
<AGM>
<DEVICE>
  <NAME> 设备 1 </NAME>
  <START_UP>
  ...
</START_UP>
  <SET_ACTIVE>
  ...
</SET_ACTIVE>
...
  <FORM>                                用于用户窗口的标签
    <INIT>
      <CONTROL name = "编辑 1" .../>      用于输入栏的标签
    </INIT>
    <PAINT>                                用于文本显示或图片显示的标签
      <TEXT>大家好! </TEXT>
    </PAINT>
  </FORM>
</DEVICE>
...
</AGM>
```

2.6.8 SOFTKEY_OK, SOFTKEY_CANCEL

说明

标签 SOFTKEY_OK 的用途在于，在按下软键“OK”关闭窗口时覆盖标准属性。标签 SOFTKEY_CANCEL 的用途在于，在按下软键“CANCEL”关闭窗口时覆盖标准属性。

在该标签内，可以执行以下功能：

- 数据复制
- 有条件处理
- 循环处理

编程

标签:	SOFTKEY_OK
句法:	<SOFTKEY_OK> ... </SOFTKEY_OK>
标签:	SOFTKEY_CANCEL
句法:	<SOFTKEY_CANCEL> ... </SOFTKEY_CANCEL>

索引

X

XML

句法, 47

运算符, 48

XML 标签

AGM, 144

BOX, 76

BREAK, 18

CAPTION, 56

CLOSE, 56

CLOSE_FORM, 57

CONTROL, 58

CONTROL_RESET, 18, 146

CREATE_CYCLE, 84

CREATE_CYCLE_EVENT, 19

DATA, 20

DATA_ACCESS (数据存取), 63

DATA_LIST, 21

DEVICE, 144

DO_WHILE (做...什么时候), 45

DYNAMIC_INCLUDE, 22

EDIT_CHANGED, 63

ELSE (否则), 22

FILE, 146

FOCUS_IN, 55

FOR, 43

FORM, 23, 50

FUNCTION (功能), 77

FUNCTION_BODY, 78

HEPL_CONTEXT (上下文帮助), 62

HMI_RESET, 23

IF, 24

IMG (图像), 74

INCLUDE (包括), 24

INDEX_CHANGED, 64

INIT, 52

KEY_EVENT, 53

LET (让), 25

MENU, 64

MESSAGE, 54

MSG, 29

MSGBOX (信息盒), 29

NAME, 144

NAVIGATION, 65

NC_INSTRUCTION, 83

OP, 30

OPEN_FORM, 66

OPTION_MD, 148

PAINT, 56

PASSWORD (口令), 37

PLC_INTERFACE, 148

POWER_OFF, 149

POWER_OFF (电源关闭), 37

PRINT (打印), 38

PROGRESS_BAR, 39

PROPERTY, 67

REQUEST, 80

SEND_MESSAGE, 40, 55

SET_ACTIVE, 144

SET_INACTIVE, 144

SHOW_CONTROL, 41

SLEEP, 40

SOFTKEY, 68

SOFTKEY_ACCEPT, 73

SOFTKEY_BACK, 72

SOFTKEY_CANCEL, 72, 151

SOFTKEY_OK, 71, 151

START_UP, 144

STOP, 40

SWITCH (转换), 41

TEST, 144

THEN, 42

TIMER, 56

TYPE_CAST, 42

UID, 144

UPDATE_CONTROLS, 81

VERSION, 144

WAITING, 149

WAITING (等待), 44

WHILE, 44

XML_PARSER, 31

文本, 73

XML 功能

Abs, 126

CEIL, 127

FLOOR, 127

LOG, 127

LOG10, 127

MAX, 128

MIN, 127

MMC, 129

NC 程序选择, 116

PI 服务, 94, 95

POW, 127

RANDOM, 128

ROUND, 127

SDEG, 126
SQRT, 126
SRAD, 126
不区分大/小写的字符串比较, 101
插入项, 120
插入字符串, 106
查找字符串, 131
查找字符串部分, 108
查找字符串部分 (反向), 109
从 **bico** 转换为 **int** 的数控功能, 96
存在, 116
对话框程序模拟, 134
对话框程序选择, 133
反向查找字符串, 132
反余弦, 113
反正切, 113
反正弦, 112
复制和读取值, 93
复制和写入值, 93
获取光标选择, 124
获取活动项, 123
获取项, 125
获取项数据, 126
将 **int** 转换为 **bico** 的数控功能, 97
控制本地时间, 99
控制格式颜色, 98
口令功能, 98
清空, 123
删除控制项, 118
删除文件, 116
删除项, 121
删除字符串, 105
删除字符串一定数量的字符, 107
设置光标选择, 125
设置活动项, 124
设置位, 117
替代字符串, 105
添加项, 119
显示器分辨率, 96
写文件, 115
有效的 **bico** 字符串, 97
余弦, 111
载入项, 122
正切, 112
正弦, 111
执行用户功能, 114
装载文件, 115
装载用户 DLL, 114
字符串比较, 100, 101
字符串长度, 104
字符串右侧, 103

字符串中间, 104
字符串左侧, 102

操

操作树, 7

创

创建调试对话框, 135

登

登入软键, 7

调

调试向导, 135

配

配置文件, 7

SIEMENS

SINUMERIK

SINUMERIK 840D sl / 828D SINUMERIK Integrate for Engineering Access MyMachine / OPC UA

开机调试手册

简介

1

设置 OPC UA 组件

2

测试连接

3

OPC UA 数据访问

4

适用于:

数控软件 版本 4.7

06/2014

6FC5397-3DP40-5RA1

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。

目录

1	简介.....	5
2	设置 OPC UA 组件.....	7
2.1	前提条件.....	7
2.2	设置.....	8
3	测试连接.....	13
4	OPC UA 数据访问.....	19
4.1	功能范畴.....	20
4.1.1	浏览.....	20
4.1.2	变量路径.....	22
4.1.3	支持的服务.....	27
4.1.4	OPC UA 规范的特点.....	28
	索引.....	29

简介

概述

OPC 统一体系架构 (OPC UA) 是一个针对工业环境的标准化通讯协议。



软件选件

使用该功能需要下列选件：SINUMERIK Integrate for Engineering Access MyMachine / OPC UA。

重要特征是：

- 平台独立性
- 安全性
- 超时服务调用
- 心跳
- 缓冲能力

网络之间的数据传输目前有两种协议：

1. 二进制协议

URL: `opc.tcp://ServerTCP-Port 4840`

针对以下内容进行了优化

- 低成本
- 占用资源少
- 可互操作性

2. 网络服务 (SOAP)

URL: `http://Server`

端口 80 (http) 和 443 (https)

针对以下内容进行了优化

- 工具支持 (“Java” 或 “.NET” 应用)
- 防火墙兼容性

所谓的“地址空间”中提供了相关信息，该部分内容由“节点”和“引用”共同组成。详细信息请访问 OPC Foundation (<https://opcfoundation.org/>) 网站。

SINUMERIK 环境下支持二进制协议，当前提供 OPC UA 数据访问服务。

设置 OPC UA 组件

2.1 前提条件

OPC UA 组件可安装在目标系统 PCU、NCU 和 PPU 上。

为此，需执行以下步骤：

1. 扩展系统配置
2. 创建 OPC UA 配置文件
3. 调整 OPC UA 配置文件
4. 安装授权
5. 激活通讯端口
6. 检查 HMI 时间
7. 执行重启

2.2 设置

扩展系统配置

为确保 OPC UA 组件能在 SINUMERIK Operate 启动时启动，需对系统配置进行相应的扩展。

将模板文件

“`.../siemens/sinumerik/hmi/template/cfg/MiniWeb_<Zielsystem>_systemconfiguration.ini`”

复制到

“`.../oem/sinumerik/hmi/cfg/systemconfiguration.ini`” 中。

说明

- 文件名会变为“`systemconfiguration.ini`”。复制后必须删除前缀“`MiniWeb_<Zielsystem>_`”。
 - 如果已经有“`.../oem/sinumerik/hmi/cfg/systemconfiguration.ini`”文件，则只须复制模板文件的内容。
 - 模板文件关联有一个链接。请确保复制的是整个文件，而不只是链接。
-

创建 OPC UA 配置文件

将模板文件

“`.../siemens/sinumerik/hmi/template/cfg/OPC_UAApplication.xml`” 复制到

“`.../oem/sinumerik/hmi/miniweb/WebCfg/OPC_UAApplication.xml`” 中。

说明

模板文件关联有一个链接。请确保复制的是整个文件，而不只是链接。

调整 OPC UA 配置文件

用文本编辑器打开复制的文件

“`.../oem/sinumerik/hmi/miniweb/WebCfg/OPC_UAApplication.xml`”。

将文件中的所有“`localhost`”条目替换为目标系统的 IPv4 地址。

- NCU 和 PPU 上：-X130
- PCU 50 上：本地连接 2

示例

```

<?xml version="1.0" standalone="yes"?>
<OPCUAAPPLICATION>
  <CERTDATA
    KeyLength="1024"
    Hash="SHA1"
    txtC="DE"
    txtP="Bavaria"
    txtL="Erlangen"
    txtO="Siemens AG"
    txtOU="Industry"
    txtCN=""
    txtE="opcua@siemens.com"
    txtCA="selfsigned" />
  <KEYS
    PublicKey="MiniWeb OPCUA_certificate.crt"
    PrivateKey="keys/MiniWeb OPCUA_key.crt"
    CAKey="" />
  <SESSION
    MaxSessionCount="100"/>
  <BROWSE
    MaxNodesPerBrowse="50"/>
  <SUBSCRIPTION
    MinSamplingRate="100"
    MaxKeepAlive="20"
    MinPublishRate="1000"/>
  <BUILDINFO
    ProductName="Sinumerik OPC UA"
    ProductUri="http://automation.siemens.com"
    ManufacturerName="Siemens AG I DT MC"
    SoftwareVersion="4.4.0"
    BuildNumber="2"/>
  <APPLICATIONDESCRIPTION
    ApplicationUri="urn:192.168.10.222:miniweb"
    ApplicationNameLocale="en_us"
    ApplicationNameText="Sinumerik OPC UA OEM"
    DNSNAME="192.168.10.222"/>
  <ENDPOINTDESCRIPTION
    URL="opc.tcp://192.168.10.222:4840"/>
  <NODEMANAGEMENT
    TargetProviderName="NodeManagementProvider"/>
</OPCUAAPPLICATION>

```

2.2 设置

设置授权

为通过 OPC UA 客户端进行数据访问，应通过操作区域“调试 > 授权”设置“Access MyMachine / OPC UA”的授权。

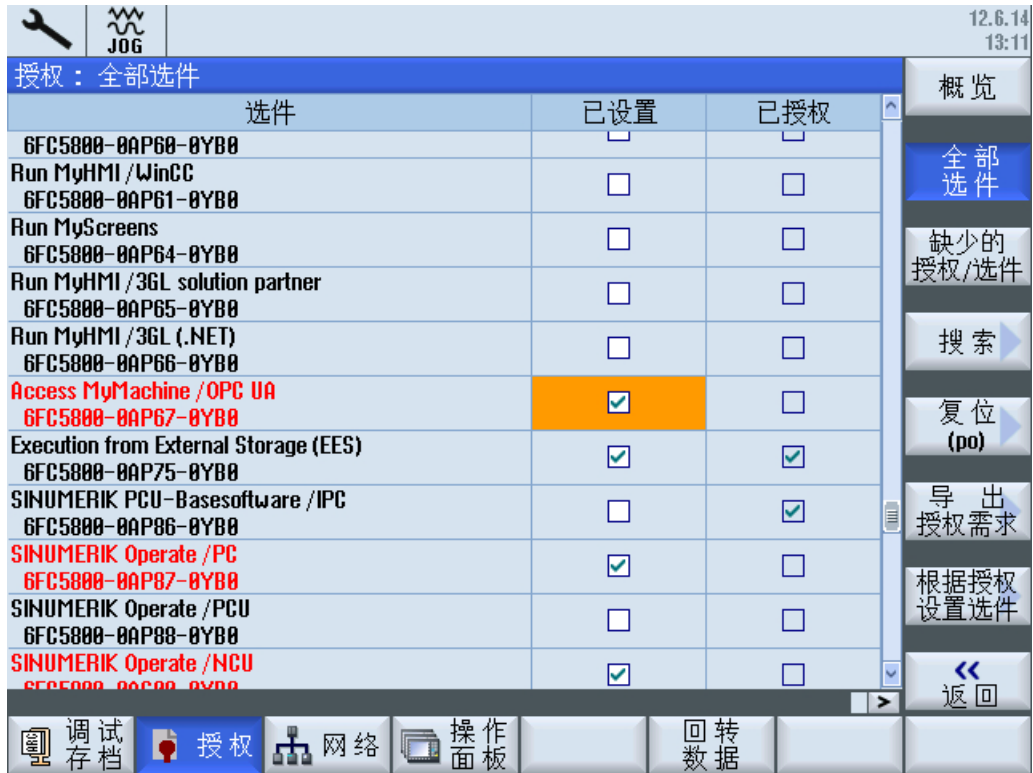


图 2-1 设置授权

激活通讯端口

说明

只允许在目标系统 NCU 和 PPU 上执行该步骤。在 PCU 50 上请直接进入下一步。

通过操作区域“调试 > 网络 > 公司网络”激活 OPC UA 通讯端口（TCP 端口 4840）。
按下按钮“更改”。

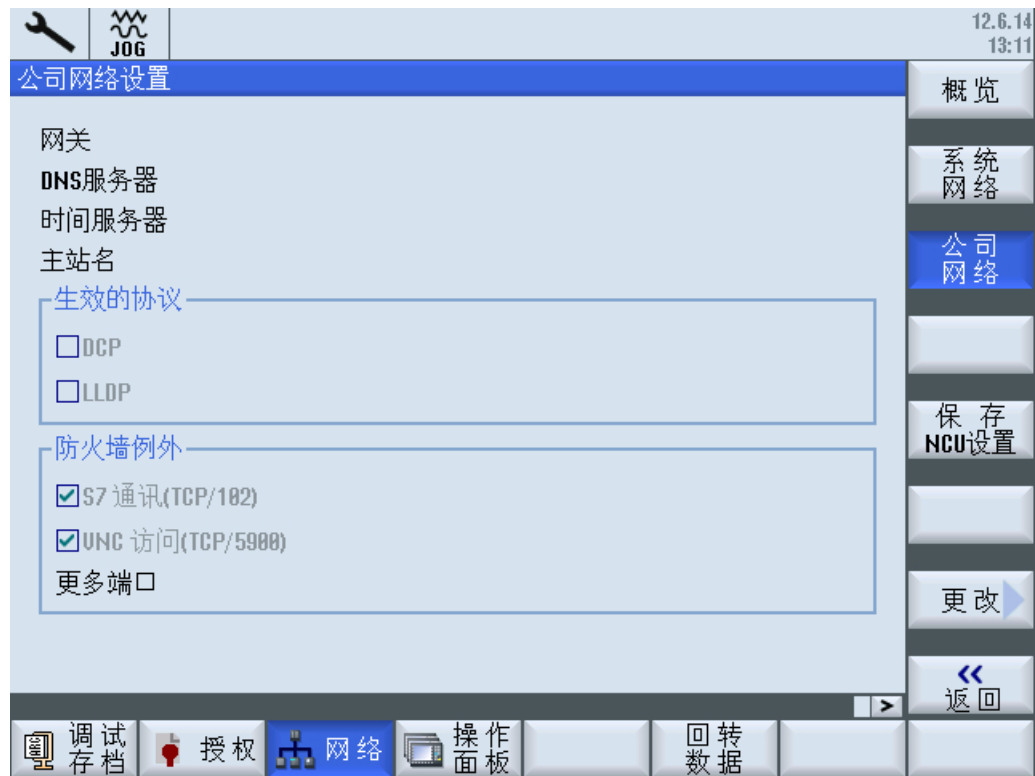


图 2-2 激活通讯端口

添加 TCP 端口 4840 并点击“确认”确认更改。

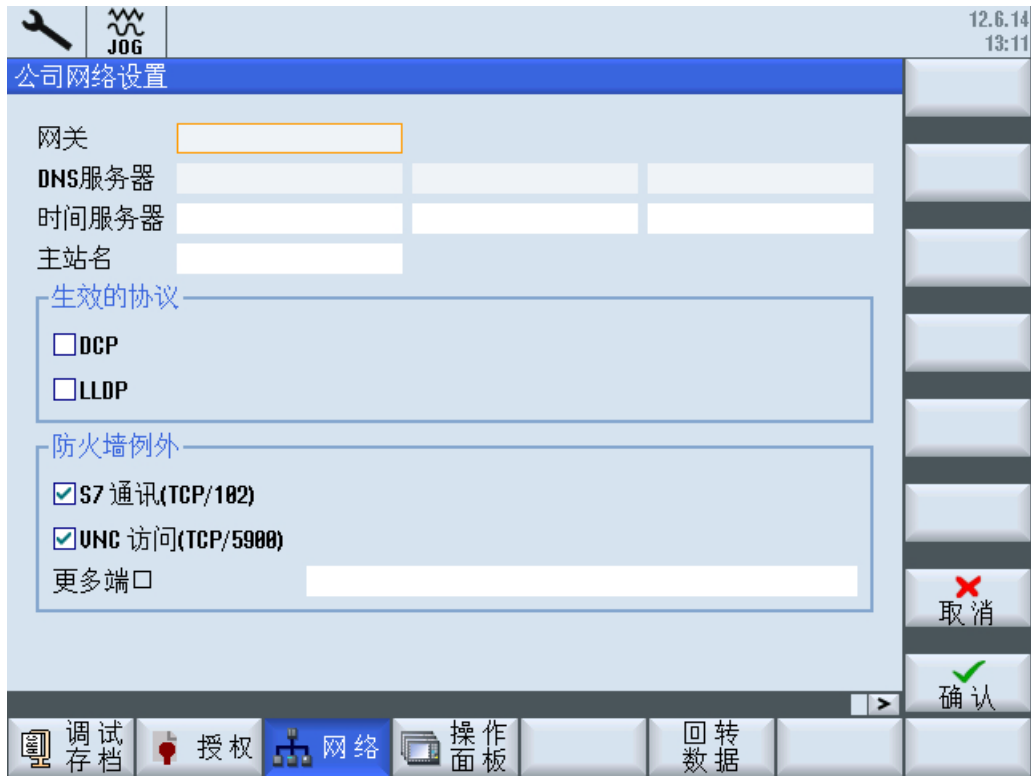


图 2-3 添加 TCP 端口

检查 HMI 时间

为确保 OPC UA 通讯能够正常运行，必须正确设置 HMI 时间。

说明

第一次启动时系统会自动生成 OPC UA 通讯所需的证书 (../siemens/sinumerik/hmi/miniweb/System/SSL/MiniWeb OPCUA_certificate.crt)。起始时间从当前时间算起，有效期为 30 年。

如果之后调整的时间超出了有效期范围，OPC UA 通讯则无法正常运行。

执行重启

重启 SINUMERIK Operate 并执行复位，确保更改和授权生效。

测试连接

前提条件

可使用 OPC Foundation (<http://www.opcf.org/>) 下 “Downloads / Sample Code and Apps / OPC UA 1.02 Sample Applications” 下的“Sample Client” 测试连接。

说明

有两种建立连接的方法：

- 无安全的连接
- 带安全策略 “Basic128Rsa15” 和安全模式 “SignAndEncrypt” 的连接

步骤

1. 启动 OPC UA “Sample Client” 。

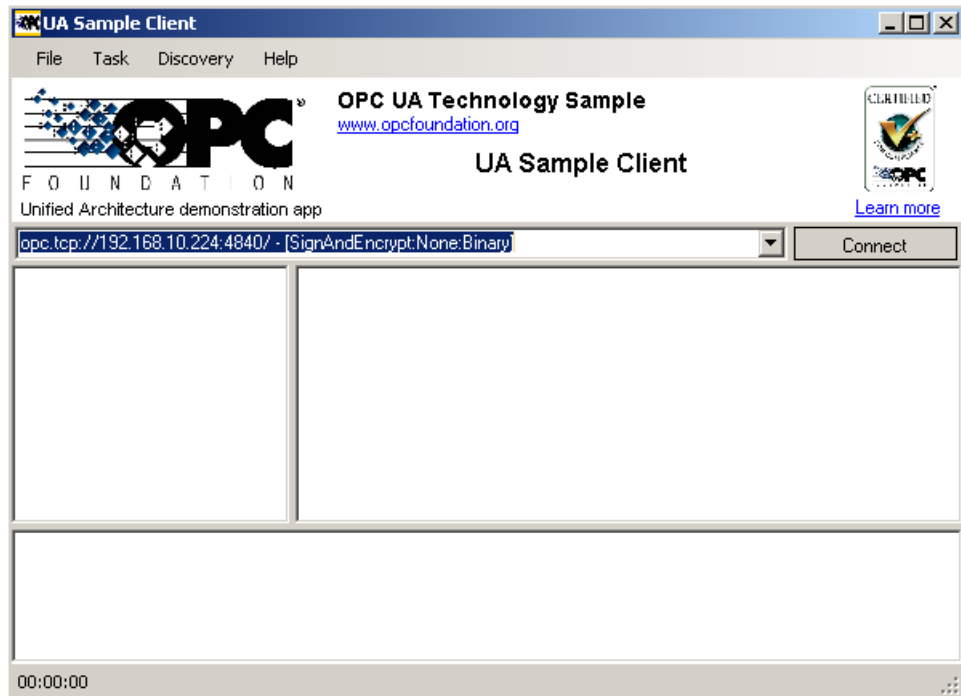


图 3-1 “Sample Client” 的主窗口

2. 在选择列表中选中条目 “New” 。
弹出窗口 “Discover Servers” 。

3. 输入目标系统的 IPv4 地址并按下按钮“Discover”。

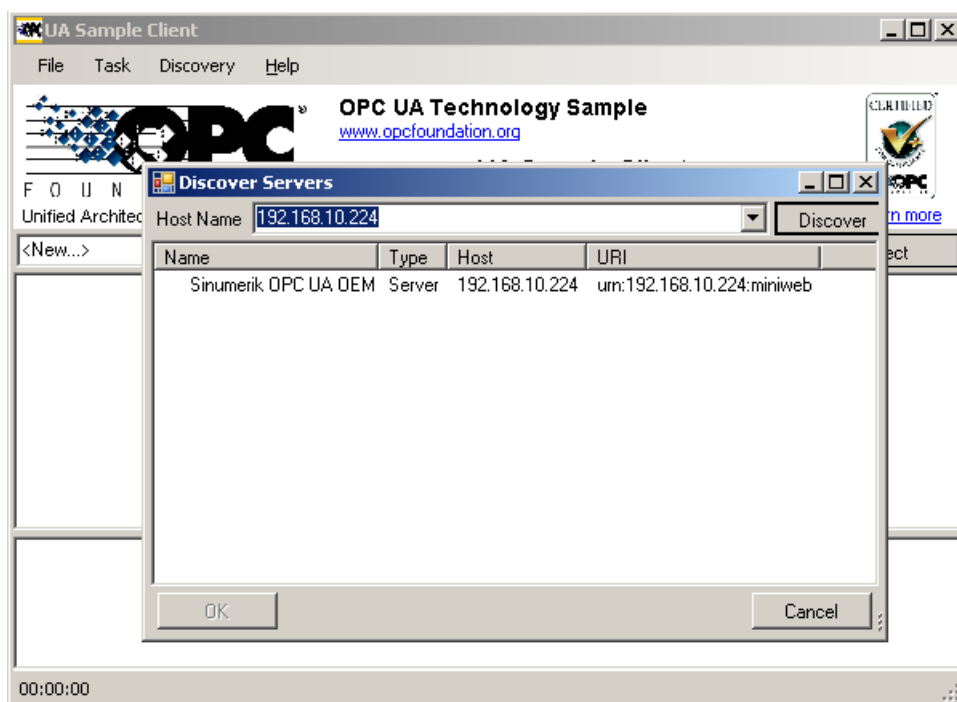


图 3-2 查看服务器

4. 列表中显示 SINUMERIK OPC UA 服务器。选中服务器并按下“OK”确认。
5. 返回主窗口并按下按钮“Connect”。

6. 为建立一个无安全的简单连接，请进行以下设置并按下 "OK" 加以确认。

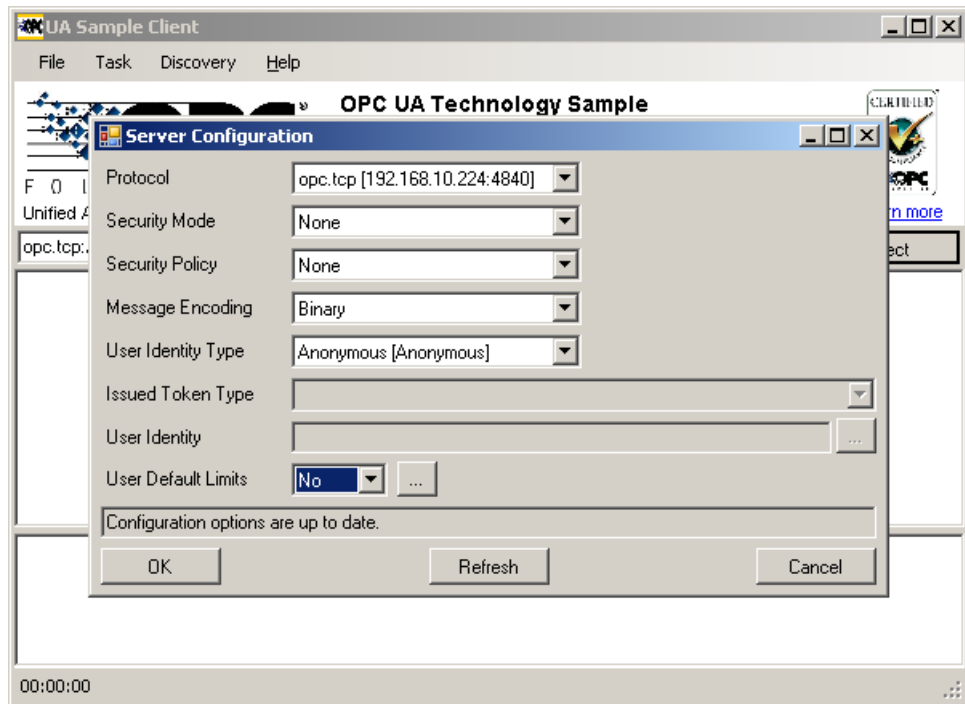


图 3-3 服务器配置

7. 按下“OK”确认以下对话框“Open Session”。

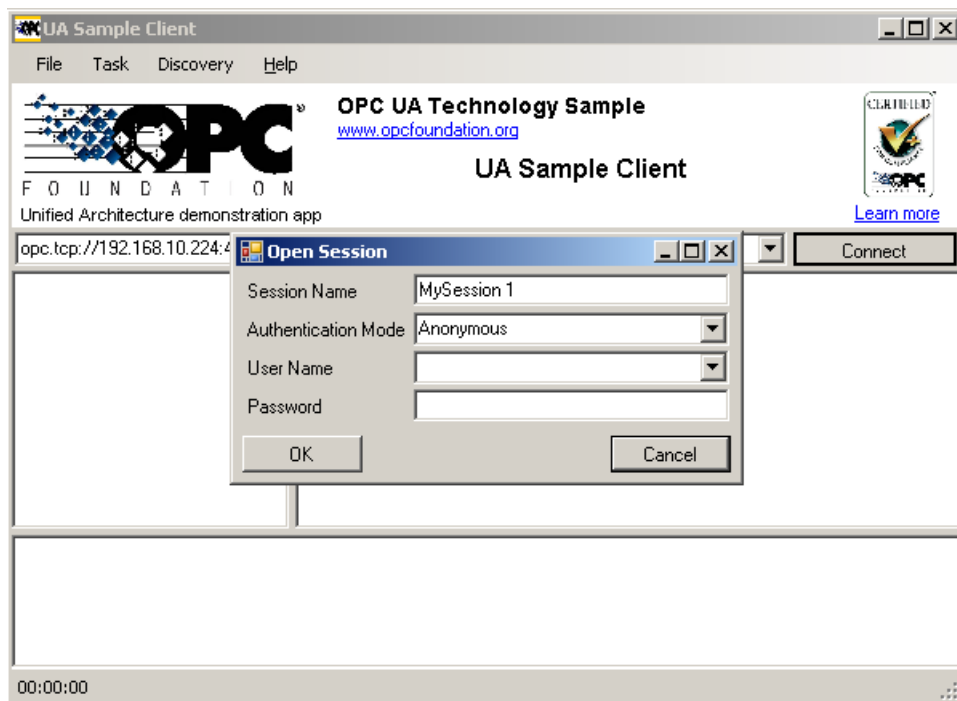


图 3-4 开放信息

已建立与 SINUMERIK OPC UA 服务器的连接，能看到可用的地址空间。

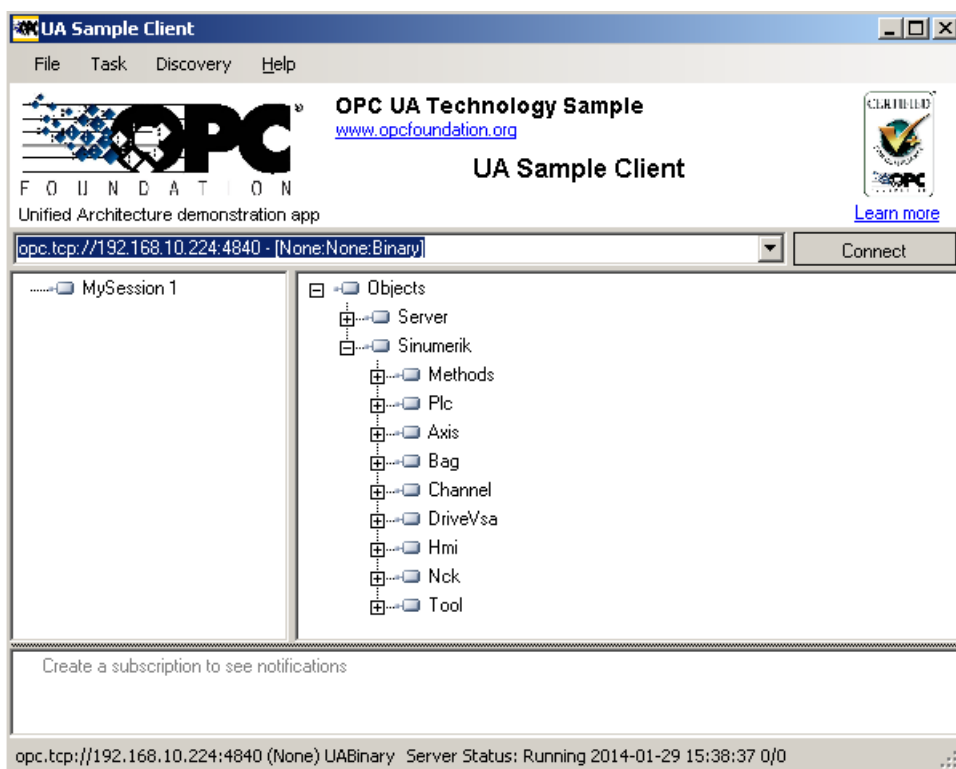


图 3-5 SINUMERIK OPC UA 服务器的地址空间

8. 将光标移至一个节点标识（例如：“Sinumerik > Channel > Parameter > R” 下的 R 参数）上并右击相应的条目。现在可以测试不同的功能：
 - 此外，还可以设置读取、写入和监控

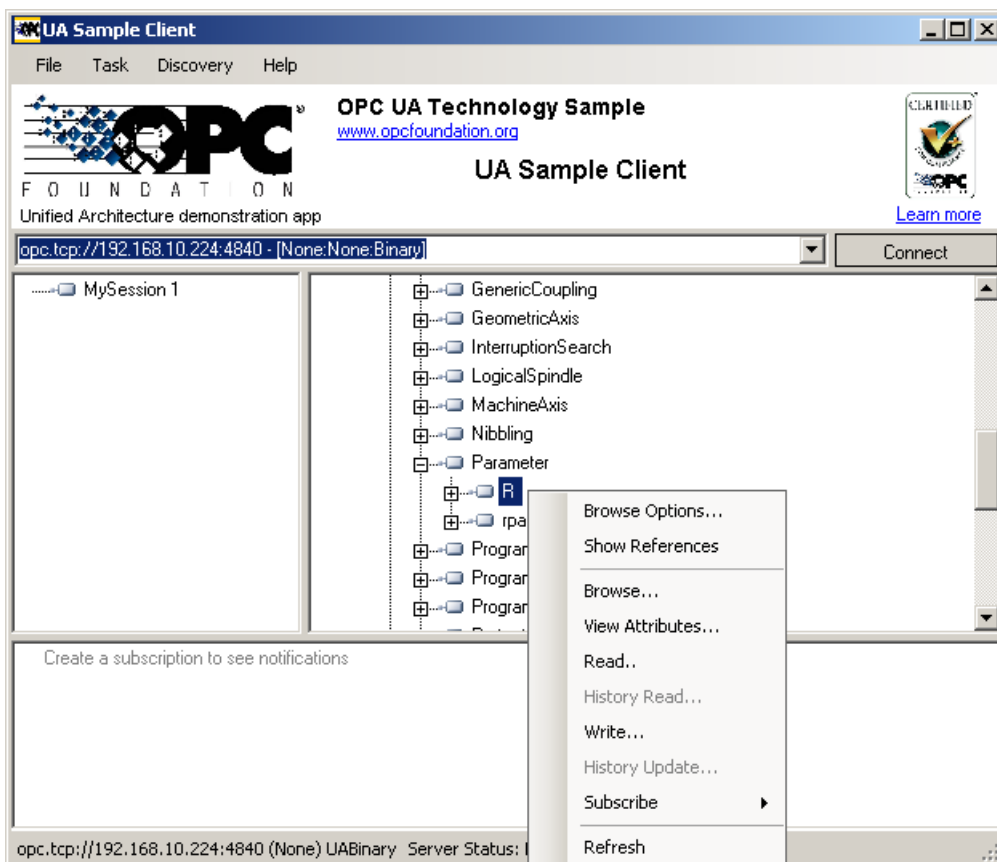


图 3-6 节点标识 “Sinumerik > Channel > Paramter > R”

- 通过条目“View Attributes”可以查询节点标识的属性。其中一个属性是“Value”，提供相应的 R1 值。

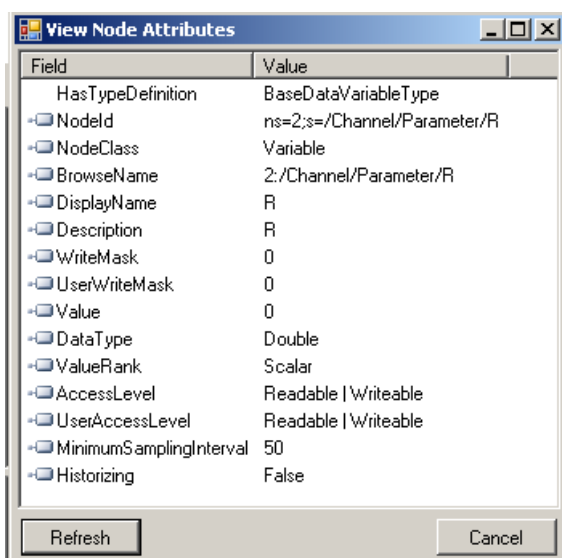


图 3-7 查看节点属性

OPC UA 数据访问

本章主要内容

本章节描述的是 OPC UA 数据访问接口的功能范畴及在 Sinumerik 上的特点。

说明

通讯没有时间保障。因此，OPC UA 数据访问接口不能解决实时任务。

有关 OPC UA 规范的详细信息请访问 OPC Foundation (<https://opcfoundation.org/>) 网站。

4.1 功能范畴

通过 OPC UA 数据访问接口可对 NC 和 PLC 进行数据访问。尤其是可以读/写变量及获取值变化的信息。

可访问单个变量（例如：/Channel/Parameter/R[u1,1]）和数组变量（例如：/Channel/Parameter/R[u1,1,10]）。

此外，还可使用以下两种方法读/写变量：

- Sinumerik / 方法 / 读变量
- Sinumerik / 方法 / 写变量

说明

同时监控的变量数受限

对象“Sinumerik”中同时待监控的变量数目有一定限制：

- SINUMERIK 828D 最多 20 个变量
- SINUMERIK 840D sl 最多 200 个变量

如果尝试监控更多的变量，系统会反馈故障“BadWaitingForInitialData”。

4.1.1 浏览

如果在 OPC UA 数据访问接口上进行浏览，可用的地址空间会在“Sinumerik”下反映。

全局用户数据（简称 GUD）位于节点“/Sinumerik/GUD”下。

PLC 功能块（输入端、输出端、标志、数据块）位于节点“/Sinumerik/Plc”下。

浏览时请注意以下几点：

- 在 NC 的地址空间中，所显示的变量始终只表示第一个操作面板单元的第三个参数。
示例：R 参数位于“Sinumerik > Channel > Parameter > R”下，对应的标识为：“/Channel/Parameter/R”，最终映射为“/Channel/Parameter/R[u1, 1]”。如果要访问其他参数，必须对标识进行相应的扩展且不可以直接采用浏览过程中获得的标识，例如：“/Channel/Parameter/R[u2, 56]”。
- 在 PLC 的地址空间中，所显示的变量表示访问格式，必须对其进行相应的扩展。
示例：变量“/Plc/MB”位于地址空间中。必须为该变量扩展相应的字节号（比如扩展为“/Plc/MB6”），之后系统才会提供一个值。

- NC 的地址空间中也包含相应机器配置中没有的变量。这些变量提供一个“BadAttributeIdInvalid”作为反馈值。
- 通过浏览无法访问机床数据和设定数据的地址空间。如何进行访问以及标识是如何组成的请参考 变量路径 (页 22) 一章。

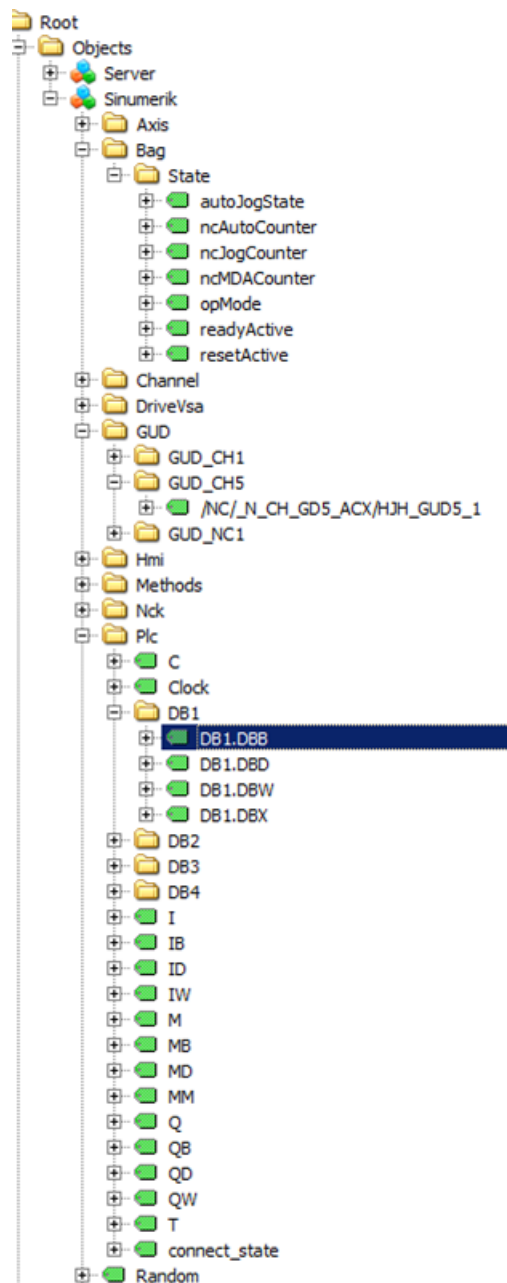


图 4-1 在 OPC UA 数据访问接口上进行浏览的节选

4.1.2 变量路径

用于访问 NC 的变量路径

用于访问 NC 的变量路径保存在 SINUMERIK Operate OPC UA 服务器的地址空间里。此时，请注意“节点标识”正确的大小写。“节点标识”指出了正确的表示方式。

更多信息可参考 840D sl 和 828D sl 参数手册中的“NC 变量和接口信号”。

Attribute	Value
[-] NodeId	NodeId
NamespaceIndex	2
IdentifierType	String
Identifier	/Channel/Parameter/R
NodeClass	Variable
BrowseName	2, "/Channel/Parameter/R"
DisplayName	"en_us", "R"
Description	"en_us", "R"
WriteMask	0
UserWriteMask	0
[-] Value	
SourceTimestamp	1/30/2014 3:17:25.822 PM
ServerTimestamp	1/30/2014 3:17:25.822 PM
SourcePicoseconds	0
ServerPicoseconds	0
Value	66
[-] DataType	Double
NamespaceIndex	0
IdentifierType	Numeric
Identifier	11
ValueRank	-1
ArrayDimensions	BadAttributeIdInvalid
AccessLevel	Readable, Writeable
UserAccessLevel	Readable, Writeable
MinimumSamplingInterval	50
Historizing	false

图 4-2 R 参数标识

所显示的 NC 变量始终只表示相应 NC 数据区的第一个参数（通道、刀具补偿范围、工作方式组）。

示例

R 参数位于标识“/Channel/Parameter/R”下，最终映射为“/Channel/Parameter/R[u1, 1]”。如果要访问其他参数，必须对标识进行相应的扩展，例如：“/Channel/Parameter/R[u2, 56]”。

表格 4-1 变量路径（NC 访问）示例

变量路径	说明
/Channel/Parameter/R[u1,10]	通道 1 中的 R 参数 10
/Channel/Parameter/R[u1,1,5]	R 参数数组
/Channel/Parameter/R[u1,1,#5]	通道 1 中的 R 参数 1 到 5

变量路径	说明
/Channel/GeometricAxis/name[u2,3]	通道 2 中第 3 轴的名称
/Channel/GeometricAxis/ actToolBasePos[u1,3]	通道 1 中第 3 轴的位置

用于访问 GUD 的变量路径

用于访问 GUD 的变量路径保存在 SINUMERIK Operate OPC UA 服务器的地址空间里，位于节点“/Sinumerik/GUD”下。请注意“节点标识”正确的大小写。“节点标识”指出了正确的表示方式。

所显示的 GUD 变量始终只表示第一个 NC 通道（取决于通道的 GUD 变量）的第一个参数（GUD 数组）。如果要访问 GUD 数组的另一个参数或另一个通道，必须依据 NC 访问对标识进行扩展。

GUD 数组在访问 1 中有索引且访问始终是一维的。也就是说，在多维数组中必须要计算索引。

示例 1：一维数组，NC 全局 GUD 数组

文件“UGUD.DEF”

```
DEF NCK INT ARRAY[2]
M17
```

访问方式如下：

```
ARRAY[0] → /NC/_N_NC_GD3_ACX/ARRAY[1]
ARRAY[1] → /NC/_N_NC_GD3_ACX/ARRAY[2]
```

示例 2，二维数组，取决于通道的 GUD 数组

文件“UGUD.DEF”

```
DEF CHAN INT ABC[3,3]
M17
```

访问方式如下：

```
ABC[0.0] → /NC/_N_CH_GD3_ACX/ABC[u1, 1]
ABC[0.1] → /NC/_N_CH_GD3_ACX/ABC[u1, 2]
ABC[0.2] → /NC/_N_CH_GD3_ACX/ABC[u1, 3]
ABC[1.0] → /NC/_N_CH_GD3_ACX/ABC[u1, 4]
ABC[1.1] → /NC/_N_CH_GD3_ACX/ABC[u1, 5]
ABC[1.2] → /NC/_N_CH_GD3_ACX/ABC[u1, 6]
ABC[2.0] → /NC/_N_CH_GD3_ACX/ABC[u1, 7]
```

4.1 功能范畴

ABC[2.1] → /NC/_N_CH_GD3_ACX/ABC[u1, 8]
 ABC[2.2] → /NC/_N_CH_GD3_ACX/ABC[u1, 9]

用于访问 PLC 的变量路径

用于访问 PLC 的变量路径保存在 SINUMERIK Operate OPC UA 服务器的地址空间里，位于节点“/Sinumerik/PLC”下。请注意“节点标识”正确的大小写。“节点标识”指出了正确的表示方式。

在 PLC 的地址空间中，所显示的变量表示访问格式，必须对其进行相应的扩展。

示例

变量“/Plc/MB”位于地址空间中。必须为该变量扩展相应的字节号（比如扩展为“/Plc/MB6”），之后系统才会提供一个值。

不同的访问格式可参考下表。必须为这些访问格式扩展前缀“/Plc”。

说明

通过 OPC UA 数据访问接口进行访问时，数据类型会发生转换。数据类型的转换情况可参考下表。

表格 4-2 PLC 句法

范围	地址 (IEC)	允许的数据类型	OPC UA 数据类型
输出端字节分配表	Qx.y	BOOL	Boolean
输出端字节分配表	QBx	BYTE, CHAR, STRING	UInt32 String
输出端字节分配表	QWx	WORD, CHAR, INT,	UInt32 Int32
输出端字节分配表	QDx	DWORD, DINT, REAL	UInt32 Int32 Double
数据块	DBz.DBXx.y	BOOL	Boolean
数据块	DBz.DBBx	BYTE, CHAR, STRING	UInt32 String
数据块	DBz.DBWx	WORD, CHAR, INT	UInt32 Int32

范围	地址 (IEC)	允许的数据类型	OPC UA 数据类型
数据块	DBz.DBDx	DWORD , DINT, REAL	UInt32 Int32 Double
输入映像	Ix.y	BOOL	Boolean
输入映像	IBx	BYTE , CHAR, STRING	UInt32 String
输入映像	IWx	WORD , CHAR, INT	UInt32 Int32
输入映像	IDx	DWORD , DINT, REAL	UInt32 Int32 Double
标志	Mx.y	BOOL	Boolean
标志	MBx	BYTE , CHAR, STRING	UInt32 String
标志	MWx	WORD , CHAR, INT	UInt32 Int32
标志	MDx	DWORD , DINT, REAL	UInt32 Int32 Double
分子	Cx	WORD	UInt32
计时器	Tx	-	Int32

表格注释:

- 表中的“x”指数据区中的字节偏移，“y”指字节中的位编号，“z”指数据块。
- **粗体格式的数据类型**是各自的默认数据类型，在寻址时无需说明。因此例如这两种写法 DB2.DBB5:BYTE 和 DB2.DBB5 是相同的。
- 在访问数组时会使用方括号，例如：“/Plc/DB5.DBW2:[10]”（长度为 10 的字数组）。

表格 4-3 变量路径（PLC 访问）示例

变量路径	说明
/Plc/M5.0	从字节偏移 5 开始的标记位 0
/Plc/DB5.DBW2	数据块 5 中从字节偏移 2 开始的字（16 位）
/Plc/DB8.DBB2:STRING	数据块 8 中从字节偏移 2 开始的 UTF8 字符串
/Plc/DB8.DBW2:[10]	数据块 8 中从字节偏移 2 开始的长度为 10 个字的数组

4.1 功能范畴

变量路径	说明
/Plc/DB100.DBB1	数据块 100 中从字节偏移 1 开始的字节
/Plc/ DB2.DBD0:REAL[10]	数据块 2 中从字节偏移 0 开始的长度为 10 个双字（32 位）的数组，格式化为浮点值

请注意如下提示：

说明

- 计时器只能读取。当值不为 0 时，计时器生效。
- 当使用数据类型 CHAR 或 STRING 时，如果是字节访问，就读取 UTF8 字符；如果是字访问，就读取 UTF16 字符。
- STRING 型变量的首个字节中为最大长度，第二个字节中为实际长度。写入字符串时，会对实际长度进行相应的调整，但不会改变最大长度。
- 关系到字节访问（例如：“/Plc/DB99.DBB0:STRING”）的 STRING 数据类型上最大字符串长度为 255 个字符。UTF8 格式中单个字符（例如：字符“μ”）需要两个字节，这样最大字符串长度会减少。
- 只支持一维数组。STRING 型数组的所有元素必须具有相同的最大长度。

机床数据和设定数据的变量路径

通过浏览无法访问机床数据和设定数据的变量路径。

用于读、写和监控机床数据和设定数据的变量路径由一个指定的前缀及机床数据或设定数据组成。

可采用以下前缀：

表格 4-4 机床数据和设定数据的前缀

范围	前缀
通用机床数据	/Random/NC/_N_NC_TEA_ACX
通道专用机床数据	/Random/NC/_N_CH_TEA_ACX
轴专用机床数据	/Random/NC/_N_AX_TEA_ACX
通用设定数据	/Random/NC/_N_NC_SEA_ACX
通道专用设定数据	/Random/NC/_N_CH_SEA_ACX
轴专用设定数据	/Random/NC/_N_AX_SEA_ACX

机床数据数组在访问 1 中有索引。

表格 4-5 变量路径（机床数据和设定数据访问）示例

变量路径	说明
/Random/NC/_N_NC_TEA_ACX/\$MN_IPO_CYCLE_TIME	插补周期 (通用机床数据)
/Random/NC/_N_CH_TEA_ACX/\$MC_DISPLAY_AXIS[u2, 1]	显示 HMI 上的轴 (通道专用机床数据)
/Random/NC/_N_AX_SEA_ACX/\$SA_LEAD_TYPE[u2]	主值类型 (轴专用设定数据)

4.1.3 支持的服务

支持以下 OPC UA 数据访问规范的服务：

- ActivateSessionService
- AddNodesService
- AddReferencesService
- BrowseNextService
- BrowseService
- CallService
- CloseSessionService
- CreateMonitoredItemsService
- CreateSessionService
- CreateSubscriptionService
- DeleteMonitoredItemsService
- DeleteNodesService
- DeleteReferencesService
- DeleteSubscriptionService
- FindServersService
- GetEndpointsService
- ModifyMonitoredItemsService
- PublishService

4.1 功能范畴

- ReadService
- RegisterNodesService
- RepublishService
- SetMonitoringModeService
- SetPublishingModeService
- TransferSubscriptionsService
- TranslateBrowsePathsToNodeIdsService
- UnregisterNodesService
- WriteService

4.1.4 OPC UA 规范的特点

TranslateBrowsePathsToNodeIdsService

如果在 TranslateBrowsePathsToNodeIds 中指定一个不存在的节点标识，得到的反馈则会 是“BadNoMatch”，而不是“BadNodeIdUnknown”。

索引

B

BadWaitingForInitialData, 20

O

OPC_UAApplication.xml, 8

读

读变量, 20

二

二进制协议, 5

方

方法

 读变量, 20

 写变量, 20

架

架构, 5

授

授权, 10

数

数据类型, 24

通

通讯端口, 10

写

写变量, 20

SIEMENS

SINUMERIK

SINUMERIK 840D sl/828D SINUMERIK Integrate Run MyScreens

编程手册

前言	1
基本操作	2
基本知识	3
对话框	4
变量	5
编程指令	6
图形单元和逻辑单元	7
操作区“Custom（定制）”	8
对话框选择	9
Referenz - Listen	A
提示和技巧	B
动画元素	C
词汇表	10




适用于:

数控软件 V4.7 SP1

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。

目录

1	前言	9
2	基本操作	13
2.1	简介	13
2.2	示例	13
2.2.1	任务说明	13
2.2.2	创建配置文件	17
2.2.3	将配置文件保存到 OEM 目录下	20
2.2.4	创建在线帮助	22
2.2.5	将在线帮助保存到 OEM 目录下	24
2.2.6	将 easyscreen.ini 复制到 OEM 目录中	25
2.2.7	申明 easyscreen.ini 中的 COM 文件	25
2.2.8	测试项目	25
3	基本知识	27
3.1	设计文件的结构	27
3.2	操作树的结构	29
3.3	登入软键的定义及功能	30
3.3.1	定义登入软键	30
3.3.2	登入软键的功能	31
3.4	错误处理（日志）	33
3.5	easyscreen.ini 的说明	34
3.6	针对改用“Run MyScreens”人员的说明	36
3.7	扩展配置句法	38
4	对话框	41
4.1	对话框的结构和组成单元	41
4.1.1	定义对话框	41
4.1.2	定义对话框属性	43
4.1.3	定义对话框单元	48
4.1.4	定义多列对话框	50
4.1.5	口令对话框	52
4.1.6	使用图/图形	54
4.2	定义软键栏	54
4.2.1	运行时改变软键属性	57
4.2.2	和语言相关的文本	60

4.3	设计在线帮助.....	62
5	变量.....	65
5.1	定义变量.....	65
5.2	应用举例.....	66
5.3	示例 1: 分配变量类型、文本、帮助画面、颜色、提示框.....	68
5.4	举例 2: 定义变量类型, 极限值, 属性, 短文本位置.....	70
5.5	举例 3: 定义变量类型、预设、系统或者用户变量、输入/输出栏位置.....	71
5.6	示例 4: 转换栏和列表栏.....	71
5.7	示例 5: 图片显示.....	73
5.8	示例 6: 进度条.....	73
5.9	示例 7: 口令输入模式 (星号)	76
5.10	变量参数.....	77
5.11	各个变量类型的详细说明.....	84
5.12	转换栏的详细说明.....	88
5.13	预设值的详细说明.....	89
5.14	短文本位置、输入输出栏位置的详细说明.....	90
5.15	使用字符串.....	91
5.16	变量 CURPOS.....	92
5.17	变量 CURVER.....	93
5.18	变量 ENTRY.....	93
5.19	变量 ERR.....	94
5.20	变量 FILE_ERR.....	95
5.21	变量 FOC.....	97
5.22	变量 S_ALEVEL.....	98
5.23	变量 S_CHAN.....	98
5.24	变量 S_CONTROL.....	99
5.25	变量 S_LANG.....	100
5.26	变量 S_NCCODEREADONLY.....	100
5.27	变量 S_RESX 和 S_RESY.....	101
6	编程指令.....	103
6.1	运算符.....	103

6.1.1	数学运算符.....	103
6.1.2	位运算符.....	107
6.2	方法.....	108
6.2.1	ACCESSLEVEL.....	109
6.2.2	CHANGE.....	109
6.2.3	CHANNEL.....	111
6.2.4	CONTROL.....	111
6.2.5	FOCUS.....	112
6.2.6	LANGUAGE.....	113
6.2.7	LOAD.....	113
6.2.8	UNLOAD.....	114
6.2.9	OUTPUT.....	115
6.2.10	PRESS.....	116
6.2.11	PRESS(ENTER).....	117
6.2.12	PRESS(TOGGLE).....	118
6.2.13	RESOLUTION.....	118
6.2.14	RESUME.....	119
6.2.15	SUSPEND.....	119
6.2.16	示例：带 OUTPUT 方法的版本管理.....	120
6.3	功能.....	122
6.3.1	读写驱动参数：RDOP, WDOP, MRDOP.....	122
6.3.2	子程序调用(CALL).....	125
6.3.3	定义块(/B).....	126
6.3.4	检查变量 (CVAR)	127
6.3.5	CLEAR_BACKGROUND.....	128
6.3.6	文件功能“Copy Program (CP, 复制程序)”.....	128
6.3.7	文件功能“Delete Program (DP, 删除程序)”.....	129
6.3.8	文件功能“Exist Program (EP, 退出程序)”.....	130
6.3.9	文件功能“Move Program (MP, 移动程序)”.....	131
6.3.10	文件功能“Select Program (SP, 选择程序)”.....	132
6.3.11	文件存取：RDFILE、WRFILE、RDLINEFILE、WRLINEFILE.....	133
6.3.12	对话框行(DLGL).....	136
6.3.13	DEBUG.....	137
6.3.14	退出对话框(EXIT).....	137
6.3.15	转换栏或列表框栏的动态列表操作.....	139
6.3.16	评估(EVAL).....	142
6.3.17	退出装载软键(EXITLS).....	143
6.3.18	功能 (FCT).....	143
6.3.19	生成代码(GC).....	145
6.3.20	口令功能.....	148
6.3.21	装载数组(LA).....	149
6.3.22	装载块 (LB)	150
6.3.23	装载屏幕窗口 (LM).....	151
6.3.24	装载软键(LS).....	153

6.3.25	Load Grid (LG).....	154
6.3.26	多次选择 SWITCH.....	155
6.3.27	多次读取 NC PLC (MRNP).....	156
6.3.28	PI 服务.....	158
6.3.29	读取 (RNP) 和写入 (WNP) 系统或用户变量.....	159
6.3.30	RESIZE_VAR_IO 和 RESIZE_VAR_TXT.....	160
6.3.31	寄存器(REG).....	161
6.3.32	RETURN.....	163
6.3.33	反编译.....	163
6.3.34	无注释反编译.....	165
6.3.35	向前/后查找(SF, SB).....	168
6.3.36	字符串功能.....	169
6.3.37	WHILE/UNTIL 循环.....	176
6.3.38	循环执行脚本: START_TIMER, STOP_TIMER.....	178
7	图形单元和逻辑单元.....	181
7.1	线、分割线、矩形、圆形和椭圆形.....	181
7.2	定义数组.....	184
7.2.1	存取数组单元的值.....	185
7.2.2	举例: 存取数组单元.....	187
7.2.3	查询数组单元的状态.....	189
7.3	表格描述(栅格).....	190
7.3.1	定义表格(栅格).....	191
7.3.2	定义列.....	192
7.3.3	表格中的焦点控制 (栅格).....	194
7.4	自定义小部件.....	195
7.4.1	自定义小部件.....	195
7.4.2	自定义小部件库的结构.....	196
7.4.3	自定义小部件接口的结构.....	196
7.4.4	自定义小部件与对话框的互动 - 自动数据交换.....	198
7.4.5	自定义小部件与对话框的互动 - 手动数据交换.....	199
7.4.5.1	读取和写入属性.....	200
7.4.5.2	执行自定义小部件的方法.....	201
7.4.5.3	响应一个自定义小部件信号.....	203
8	操作区“Custom (定制)”.....	207
8.1	这样激活操作区“自定义”.....	207
8.2	这样设计“自定义”软键.....	207
8.3	这样设计操作区“自定义”.....	208
8.4	操作区“自定义”的编程示例.....	209
9	对话框选择.....	213
9.1	通过 PLC 软键选择对话框.....	213

9.2	通过 PLC 硬键选择对话框.....	214
9.3	通过 NC 选择对话框.....	217
A	Referenz - Listen.....	219
A.1	登入软键表.....	219
A.1.1	车床登入软键表.....	219
A.1.2	铣床登入软键表.....	221
A.2	访问等级列表.....	222
A.3	颜色表.....	222
A.4	文件名中的语种缩写表.....	224
A.5	可用的系统变量列表.....	225
A.6	对话框打开方式（属性 CB）.....	225
B	提示和技巧.....	227
B.1	一般提示.....	227
B.2	DEBUG 的提示.....	228
B.3	CHANGE 方法的提示.....	229
B.4	DO-LOOP 循环的提示.....	230
C	动画元素.....	231
C.1	简介.....	231
C.2	建模.....	232
C.2.1	前提条件.....	232
C.2.2	建模规则.....	232
C.2.3	导入图形（模型）.....	235
C.2.4	建模模板.....	237
C.3	XML 指令.....	239
C.3.1	一览.....	239
C.3.2	场景说明文件的结构.....	239
C.3.3	镜像和旋转.....	242
C.3.4	视图类型.....	243
C.4	转换为 hmi 文件.....	244
C.5	在 Create MyHMI /3GL 中显示.....	244
C.5.1	X3D 浏览器.....	244
C.5.2	类 SIX3dViewerWidget.....	244
C.5.3	公共方法.....	245
C.5.4	公共槽.....	245
C.5.5	库.....	246
C.5.6	执行示例.....	246

C.5.7	机床数据.....	246
C.5.8	使用说明.....	247
C.6	在 Run MyScreens 中显示.....	247
10	词汇表.....	249
	索引.....	253

前言

概述

通过“Run MyScreens” 机床制造商或最终用户可以自行设计扩展功能的操作界面，或者设计操作界面的布局。

还可以在其他循环调用下生成自定义的操作界面。或直接在控制系统上创建对话框。

“Run MyScreens” 通过编译器和包含操作界面说明的配置文件实现。

通过 ASCII 文件配置“Run MyScreens”：该配置文件包含了关于操作界面的说明。创建文件所需的句法参见下列章节。

基本功能

通过“Run MyScreens” 机床制造商可自定义对话框。基本功能支持在操作菜单树中设计 5 个对话框，或者用于设计用户自定义的循环对话框。



软件选件

扩展对话框数量，需要下列软件选件：

- SINUMERIK 828D/840D sl, SINUMERIK Integrate Run MyScreens (6FC5800-0AP64-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyScreens + Run MyHMI (6FC5800-0AP65-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyHMI / 3GL (6FC5800-0AP60-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyHMI / WinCC (6FC5800-0AP61-0YB0)

边界条件

必须符合以下边界条件：

- 只允许在一个操作区内切换对话框。
- 用户变量不允许与系统变量或者 PLC 变量有相同的名称（另请参见参数手册系统变量/PGAsI/）
- 由 PLC 激活的对话框构成一个单独的操作区（和测量循环屏幕类似）。

工具

- UTF8 编辑器（例如：操作界面的集成编辑器或记事本）
- 创建图形/屏幕所需的图形程序。

文件名和编码

说明

使用 NCU 中的 HMI Operate 时要注意 CF 卡上的所有文件名都是以小写字母保存的（如：com、png、txt）。这是 Linux 操作系统的要求。

在 PCU 上您可以任意以大小写字母保存文件名。为便于之后传送到 Linux，此处仍建议只使用小写字母。

说明

保存配置文件和语言文件时请注意，应将您所使用的编辑器中的编码设为 UTF 8。

保存路径

保存配置文件、帮助文件等时应遵循以下规定：

<i>[系统西门子目录]</i>	
Linux:	/card/siemens/sinumerik/hmi
Windows 7:	C:\ProgramData\Siemens\MotionControl \siemens
<i>[系统 oem 目录]</i>	
Linux:	/card/oem/sinumerik/hmi
Windows 7:	C:\ProgramData\Siemens\MotionControl \oem
<i>[系统用户目录]</i>	
Linux:	/card/user/sinumerik/hmi
Windows 7:	C:\ProgramData\Siemens\MotionControl \user
<i>[系统插件目录]</i>	
Linux:	/card/addon/sinumerik/hmi

Windows 7:	C:\ProgramData\Siemens\MotionControl\addon
保存位置	
“Run MyScreens” 配置:	[系统 oem 目录]\proj
配置文件:	[系统 oem 目录]\cfg
语言文件:	[系统 oem 目录]\lng
图形文件:	[系统 oem 目录]\col\ico[分辨率] 示例: [系统 oem 目录]\col\ico640
在线帮助:	[系统 oem 目录]\hlp/[语言] 示例: [系统 oem 目录]\hlp/eng

使用

可以实现以下功能:

显示对话框并提供:	<ul style="list-style-type: none"> ● 软键 ● 变量 ● 文本和帮助文本 ● 图形和帮助图形
通过以下方法调用对话框:	<ul style="list-style-type: none"> ● 按下（登入）软键 ● 选择 PLC/NC
动态重组对话框	<ul style="list-style-type: none"> ● 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
在进行以下操作时触发动作:	<ul style="list-style-type: none"> ● 显示对话框 ● 输入数值（变量） ● 按下软键 ● 关闭对话框 ● 系统或用户变量的值变化
对话框间的数据交换	

变量	<ul style="list-style-type: none">● 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
执行下列功能:	<ul style="list-style-type: none">● 子程序● 文件功能● PI 服务
根据用户组考虑保护等级	

基本操作

2.1 简介

根据以下示例可了解使用 Run MyScreen 在 SINUMERIK Operate 操作界面中添加自定义对话框的必要操作步骤。此外还可了解如何创建自定义对话框、如何添加上下文有关的帮助图形和帮助调用框、如何定义软键以及如何如何在对话框之间进行导航。

2.2 示例

2.2.1 任务说明

本例中将创建以下对话框。

对话框 1

第一个对话框中显示可写 R 参数（0 和 1）和几何轴名称（输入栏）。两个 R 参数均链接有相应的帮助图形。几何轴名称未链接上下文有关的帮助信息。此外对话框中还包含诸如分割线（水平和垂直）、转换栏、集成了单位选择框的输入/输出栏和进度条（变色或不变色）等元素。

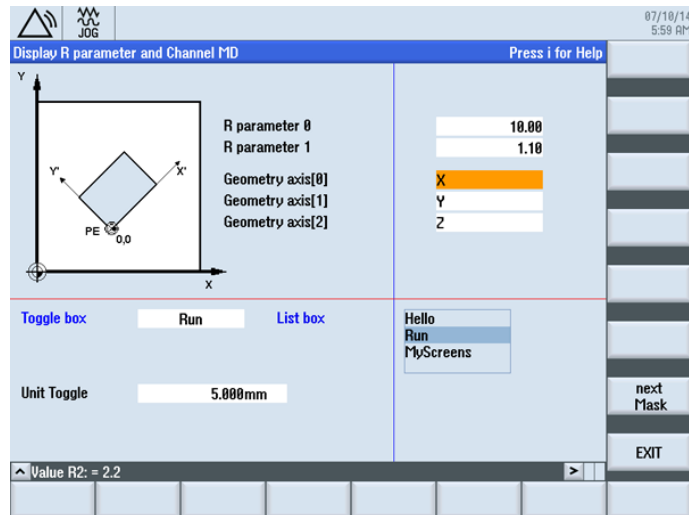


图 2-1 带辅助图的 R 参数

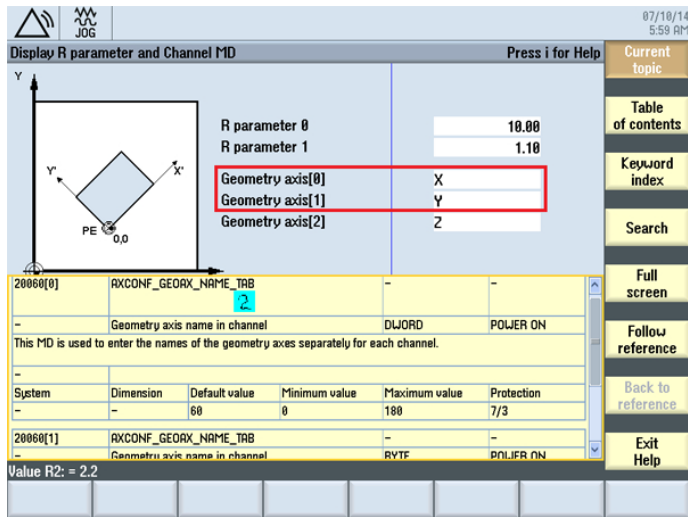


图 2-2 带上下文相关帮助信息的几何轴

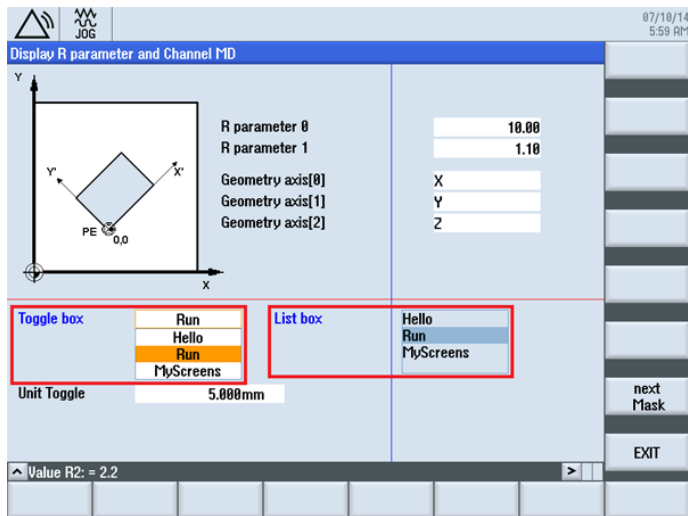


图 2-3 转换栏：列表和选项框

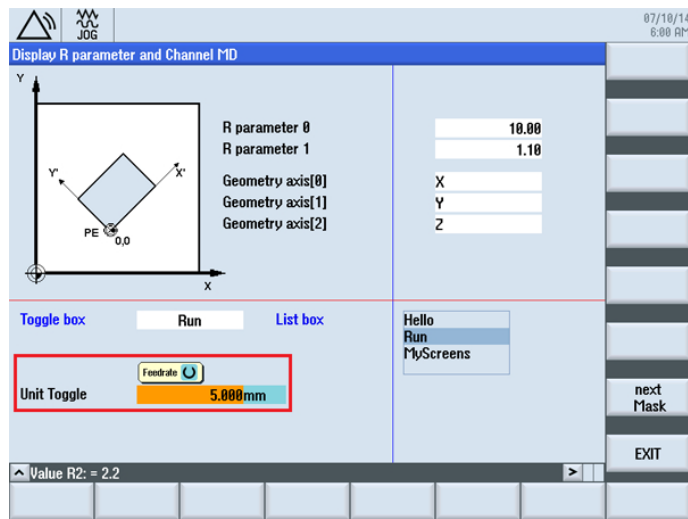


图 2-4 集成了单位选择框的输入/输出栏

对话框 2

第二个对话框中显示 MCS 值和 WCS 值。此外对话框中还含有变色或不变色的进度显示。

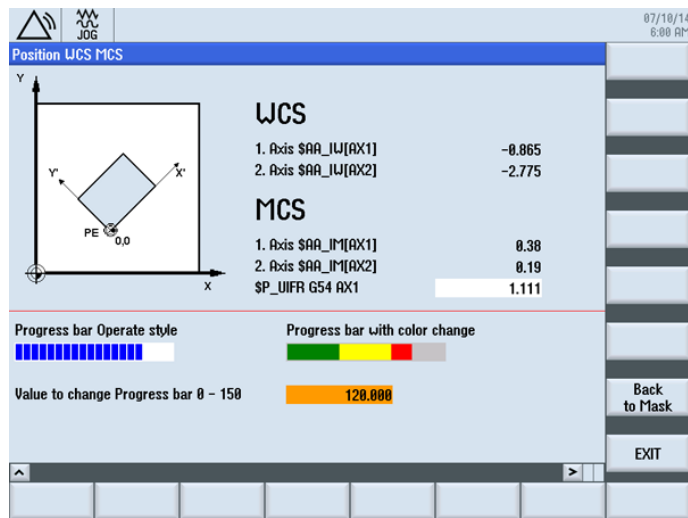


图 2-5 变色（右）和不变色（左）的进度条

浏览

通过“诊断”操作区中的“START”登入软键调用第一个对话框。使用水平软键 SK7。

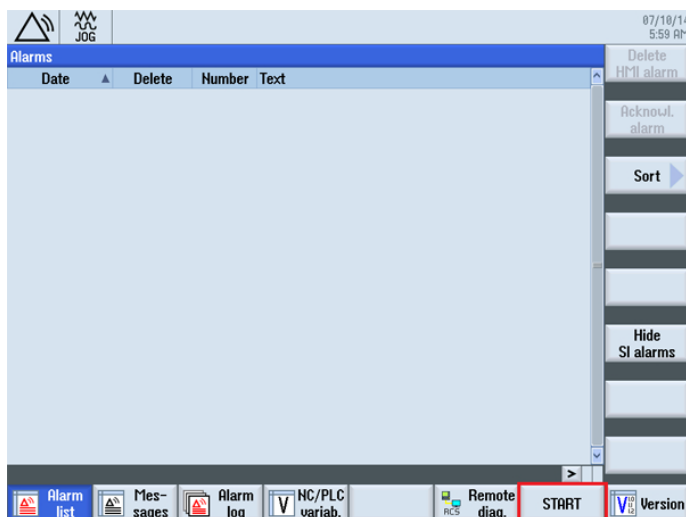


图 2-6 AUTO 运行模式下“加工”操作区的登入软键,,START“

在第一个对话框中使用软键“next Mask”可调用第二个对话框，使用软键“EXIT”返回操作区的基本画面（见上图）。

在第二个对话框中使用软键“EXIT”也同样能返回操作区的基本画面（见上图）。使用软键“Back to Mask”可以返回第一个对话框。

步骤

以下章节中说明了必要的操作步骤：

1. 创建配置文件（COM 文件）
2. 将配置文件保存到 OEM 目录下
3. 创建在线帮助
4. 将在线帮助保存到 OEM 目录下
5. 将 easyscreen.ini 复制到 OEM 目录中
6. 申明 easyscreen.ini 中的 COM 文件
7. 测试项目

2.2.2 创建配置文件

配置文件的内容

在 UTF8 编辑器中为两个对话框创建配置文件 `diag.com`。

```

; 登入软键开始标记
//S (START)
; 纯文本登入软键
HS7=("START")
; 登入软键, 带和语言相关的文本和 png
;HS7=([$80792,"\\sk_ok.png"])

; PRESS 方法
PRESS (HS7)
; 功能 LM 或者 LS
LM("MASK1")
; 带 com 文件说明的 LM
LM("MASK1","TEST.COM")
END_PRESS
; 登入软键结束标记
//END

; 带标题和图像的对话框 1 的定义
//M(MASK1/"Display R parameter and Channel MD"/"mz961_01.png")
; 变量定义
DEF VAR1 = (R2///,"R parameter 0"///"$R[0]"/200,50,150/400,50,100,)
; 带帮助图形
DEF VAR2 = (R2///,"R parameter 1"///"mz961_02.png"/"$R[1]"/200,70,150/400,70,100)
; 带在线帮助
DEF ACHS_NAM1 = (S///"Press i for Help","Geometry axis[0]"////
200,100,150/400,100,100//"$sinumerik_md_1.html","20060[0]")
; 带在线帮助
DEF ACHS_NAM2 = (S///"Press i for Help","Geometry axis[1]"////
200,120,150/400,120,100//"$sinumerik_md_1.html","20060[1]")
DEF ACHS_NAM3 = (S///,"Geometry axis[2]"////200,140,150/400,140,100)
; 转换栏和单位转换栏的定义
DEF VAR_TGL = (S/* "Hello", "Run", "MyScreens"/"Run"/,"Toggle box"///10,230,100/120,230,100/,,6)

DEF VAR_TGB = (S/* "Hello", "Run", "MyScreens"/"Run"/,"List box"/
dt4///250,230,100/370,230,100,60/,, "#0602ee")
; BC, FC, BC_ST, FC_ST, BC_GT, FC_GT, BC_UT, FC_UT, SC1, SC2
DEF VarEdit = (R///,"Unit Toggle",,, "Feedrate"///"$R[11]"/5,300,100/120,300,100///"VarTgl"),
VarTgl = (S/*0="mm",1="inch"/0//WR2///220,300,40)

```

2.2 示例

```
; 对话框中的软键定义
HS1= ("")
HS2= ("")
HS3= ("")
HS4= ("")
HS5= ("")
HS6= ("")
HS7= ("")
HS8= ("")
VS1= ("")
VS2= ("")
VS3= ("")
VS4= ("")
VS5= ("")
VS6= ("")
VS7= ("next Mask")
VS8= ("EXIT")

; LOAD 块定义
LOAD
; 通过 RNP 读取值
ACHS_NAM1 = RNP (" $MC_AXCONF_GEOAX_NAME_TAB [0] ")
ACHS_NAM2 = RNP (" $MC_AXCONF_GEOAX_NAME_TAB [1] ")
ACHS_NAM3 = RNP (" $MC_AXCONF_GEOAX_NAME_TAB [2] ")
; 对话框行输出
DLGL ("Value R2: = " << RNP (" $R [2] "))
; 通过 WNP 写入值
WNP (" $R [3] ", VAR0)

; 垂直分割线
V_SEPARATOR (360, 1, 6, 1)
; 水平分割线
H_SEPARATOR (220, 1, 7, 1)
END_LOAD

; PRESS 方法
PRESS (VS7)
; 载入其他对话框
LM ("MASK2")
; PRESS 方法的结束标记
END_PRESS

; PRESS 方法
PRESS (VS8)
```

```

; 退出对话框
EXIT
; PRESS 方法的结束标记
END_PRESS
; 对话框 1 的结束标记
//END

; 带标题和图像的对话框 2 的定义
//M(MASK2/"Position WCS MCS"/"mz961_01.png")
; 变量定义
DEF TEXT1 = (I///,"WCS"/WR0,fs2///230,30,120/, ,1)
DEF VAR1 = (R3///,"1. Axis $AA_IW[AX1]"/WR1/"$AA_IW[AX1]"/230,70,150/400,70,100)
DEF VAR2 = (R3///,"2. Axis $AA_IW[AX2]"/WR1/"$AA_IW[AX2]"/230,90,150/400,90,100)

DEF TEXT2 = (I///,"MCS"/WR0,fs2///230,120,120/, ,1)
DEF VAR3 = (R2///,"1. Axis $AA_IM[AX1]"/WR1/"$AA_IM[AX1]"/230,160,150/400,160,100)
DEF VAR4 = (R2///,"2. Axis $AA_IM[AX2]"/WR1/"$AA_IM[AX2]"/230,180,150/400,180,100)
DEF VAR5 = (R3///,"$P_UIFR G54 AX1"///"$P_UIFR[1,AX1,TR]"/230,200,150/400,200,100)

; 进度条的定义
DEF PROGGY0 = (R/0,150//,"Progress bar Operate style"/DT2,DO0/"$R[10]"/5,240,190/5,260,150/6,10)
DEF PROGGY4 = (R/0,150,50,100/120//,"Progress bar with color change"/DT1,DO0/"$R[10]"/
260,240,190/260,260,150/3,4, , ,9,7)

; 变量定义
DEF PROGVAL = (R/0,150//,"Value to change Progress bar 0 - 150"///"$R[10]"/5,300,230/260,300,100)

; 对话框中的软键定义
HS1=("")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("")
VS5=("")
VS6=("")
VS7=("Back to Mask")
VS8=("EXIT")

```

2.2 示例

```
; Load 方法
LOAD
  H_SEPARATOR(230,1,7,1)
END_LOAD

; 更改方法
CHANGE (VAR5)
  ; PI_START 功能
  PI_START ("/NC,201,_N_SETUFR")
; 更改方法的结束标记
END_CHANGE

; PRESS 方法
PRESS (RECALL)
  ; 返回主调对话框
  LM ("MASK1")
; PRESS 方法的结束标记
END_PRESS

; PRESS 方法
PRESS (VS7)
  ; 返回主调对话框
  LM ("MASK1")
; PRESS 方法的结束标记
END_PRESS

; PRESS 方法
PRESS (VS8)
  ; 退出对话框, 返回到标准应用程序
  EXIT
; PRESS 方法的结束标记
END_PRESS
; 对话框的结束标记
//END
```

2.2.3 将配置文件保存到 OEM 目录下

保存路径

将配置文件 `diag.com` 保存到以下路径中:

[系统 oem 目录]/proj

2.2.4 创建在线帮助

在线帮助的内容

创建 HTML 文件 `sinumerik_md_1.html`。下段中通过 `name="20060[0]"` 和 `name="20060[1]"` 显示上下文有关的帮助调用框。帮助文件没有加注释，因为如何创建在线帮助不属于本手册内容。如何加入在线帮助的说明参见章节设计在线帮助 (页 62)。

```

<html><head><meta http-equiv="Content-Type" content="text/html; charset="UTF-8"/><title></
title></head>
  <body>
    <table border="1" cellspacing="0" cellpadding="2" width="100%">
      <tr>
        <td><a name="20060[0]"><b>20060[0]</b></a></td>
        <td colspan="3">AXCONF_GEOAX_NAME_TAB</td>
        <center>
          
        </center>
        <td>-</td>
        <td>-</td>
      </tr>
      <tr>
        <td>-</td>
        <td colspan="3">Geometry axis name in channel</td>
        <td>DWORD</td>
        <td>POWER ON</td>
      </tr>
      <tr>
        <td colspan="6">This MD is used to enter the names of the geometry axes
separately for each channel.<br />
      <tr>
        <td>-</td>
        <td colspan="5">&nbsp;</td>
      </tr>
      <tr>
        <td width="16*">System</td><td width="16*">Dimension</td><td width="16*">Default
value</td>
        <td width="16*">Minimum value</td><td width="16*">Maximum value</td>
        <td width="16*">Protection</td>
      </tr>
      <tr>
        <td>-</td>
        <td>-</td>
        <td>60</td>
        <td>0</td>
        <td>180</td>
        <td>7/3</td>
      </tr>
    </table>
    <p></p>
    <table border="1" cellspacing="0" cellpadding="2" width="100%">
      <tr>
        <td><a name="20060[1]"><b>20060[1]</b></a></td>
        <td colspan="3">AXCONF_GEOAX_NAME_TAB</td>
        <td>-</td>
        <td>-</td>
      </tr>
      <tr>

```

2.2 示例

```

        <td>--</td>
        <td colspan="3">Geometry axis name in channel</td><td>BYTE</td>
        <td>POWER ON</td>
    </tr>
    <tr>
        <td colspan="6">This MD is used to enter the names of the geometry axes
separately for each channel.<br />
    <tr>
        <td>--</td>
        <td colspan="5">&nbsp;</td>
    </tr>
    <tr>
        <td width="16*">System</td>
        <td width="16*">Dimension</td>
        <td width="16*">Default value</td>
        <td width="16*">Minimum value</td>
        <td width="16*">Maximum value</td>
        <td width="16*">Protection</td>
    </tr>
    <tr>
        <td>--</td>
        <td>--</td>
        <td>0</td>
        <td>0</td>
        <td>2</td>
        <td>7/3</td>
    </tr>
</table>
<p></p>
</body>
</html>

```

2.2.5 将在线帮助保存到 OEM 目录下

保存路径

将 HTML 文件 `sinumerik_md_1.html` 的德语帮助信息保存到以下路径下：

`[系统 oem 目录]/hlp/deu`

必要时，需要保存其他语言文件夹（例如：`chs`、`eng`、`esp`、`fra`、`ita` ...）。

语种缩写列表参见附录。

2.2.6 将 easyscreen.ini 复制到 OEM 目录中

保存路径

从以下目录中复制文件 easyscreen.ini:

[系统西门子目录]/cfg

复制到

[系统 oem 目录]/cfg.

2.2.7 申明 easyscreen.ini 中的 COM 文件

easyscreen.ini 中的调整

在 OEM 目录下的 easyscreen.ini 中进行如下修改，以便申明配置文件 diag.com。

```
#####  
;# AREA Diagnosis      #  
#####  
;<=====>  
;< OEM Softkey on first horizontal Main Menu      >  
;< SOFTKEY position="7"                          >  
;<=====>  
StartFile28 = area := AreaDiagnosis, dialog:= SLDgDialog, menu := DgGlobalHu, startfile := diag.com
```

2.2.8 测试项目

测试对话框调用

转换至“诊断”操作区。按下水平软键“START”。

如果“Run MyScreens”编译配置文件时发现错误，则该错误会保存在文本文件 easyscreen_log.txt 中（参见章节 错误处理（日志）（页 33））。

测试上下文相关的帮助图形和在线帮助

测试上下文相关的帮助图形和在线帮助，如果出现错误请检查保存路径。

基本知识

3.1 设计文件的结构

简介

新操作界面的描述保存在配置文件中。这些文件自动编译并显示屏幕上的结果。在供货时并不提供配置文件，因此必须首先创建此文件。

创建配置文件和语言文件时使用 **UTF 8** 编辑器（例如：操作界面的集成编辑器或记事本）。描述可以通过注释来进一步说明。注释字符前会添加“; ”。

说明

保存配置文件和语言文件时请注意，应将您所使用的编辑器中的编码设为 **UTF 8**。

然而，关键字只允许由 **ASCII** 字符集中的字符组成（**UTF 8** 编译的 **COM** 文件中也一样）。否则就无法保证编译和屏幕/对话框的显示。

配置

每个 **HMI** 操作区都包含固定的登入软键，以进入新创建的对话框。

调用配置文件中的“载入屏幕”(**LM**) 或者“载入软键栏”(**LS**)可以重新命名被调对象所在的文件。采用这种方式可以对配置文件进行分类，例如：同一操作级的所有功能在一个单独的配置文件中。

配置文件的结构

配置文件由以下单元组成：

1. 登入软键说明
2. 对话框定义
3. 变量定义
4. 方法说明
5. 软键栏定义

3.1 设计文件的结构

说明

顺序

必须遵循上文列出的各单元在配置文件中的顺序。

示例：

```
//S (START)                ; 登入软键定义 (可选)
....
//END
//M (.....)              ; 对话框定义
DEF .....                 ; 变量定义
LOAD                      ; 块描述
...
END_LOAD
UNLOAD
...
END_UNLOAD
...
//END
//S (...)                  ; 软键条定义
//END
```

配置文件的保存

配置文件保存在目录 *[系统用户目录]/proj* 下，也相应的保存在目录 *[系统插件目录]* 和 *[系统 oem 目录]* 下。

转换其他 HMI 应用程序中的文本

从 Codepage 编码的文本文件向 UTF-8 编码文本转换的步骤：

1. 在 PG/PC 上的文本编辑器中打开文本文件。
2. 保存时选择 UTF 8 编码。

使用 Codepage 编码的读取机制还继续被支持。

3.2 操作树的结构

操作树的原理

多个相互关联的对话框构成了一个操作树。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关联。点击在这种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。

在每个登入软键下都可以生成一个操作树：

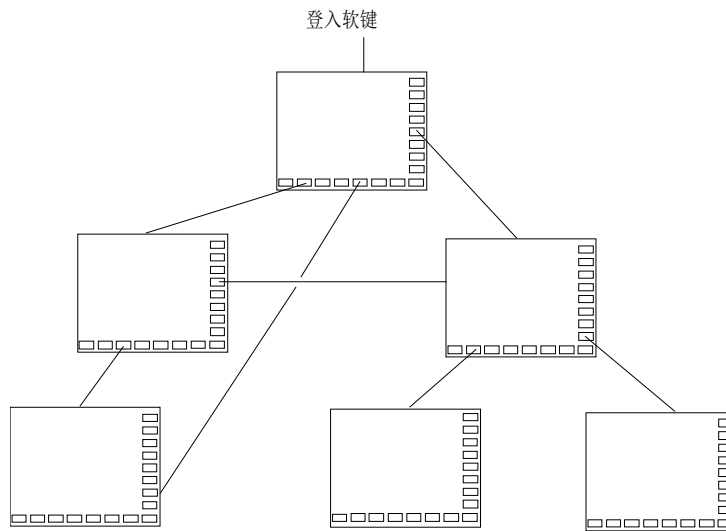


图 3-1 操作树

登入软键

在一个 `easyscreen.ini` 中指定的配置文件中，定义了一个或者多个软键（登入软键），这些软键可以作为操作的出发点。

软键的定义决定执行下一步动作的自定义对话框或者其他软键栏。

按下登入软键则载入所属的对话框。同时，属于对话框的相应软键激活。如果没有设计特定位置，则在标准位置上给出变量。

返回到标准应用程序

可以退出新定义的对话框并返回到标准操作区。

3.3 登入软键的定义及功能

通过<RECALL>（回调）键可以退出新定义的操作界面，如果这个按键还未被设计用于其它用途。

说明

在 PLC 用户程序中调用对话框

除了软键，也可以由 PLC 选择对话框。DB19.DBB10 中的接口信号用于 PLC → HMI 的信号交换。

3.3 登入软键的定义及功能

3.3.1 定义登入软键

和对话框无关的软键

登入软键是和对话框无关的软键，它不由对话框调用，而是在第一个新对话框之前设计。为访问一个登入对话框或者一个登入软键栏，必须对登入软键进行定义。

编程

登入软键的描述块如下构建：

```
//S (Start)           ; 登入软键开始标记
HS1=(...)            ; 定义登入软键： 水平软键 SK 1
PRESS (HS1)         ; 方法
LM...               ; 功能 LM 或者 LS
END_PRESS          ; 方法结束
//END               ; 登入软键结束标记
```

登入软键的允许位置

在操作区中“Run MyScreens” 登入软键的允许位置有：

操作区	位置
加工	HSK6
参数	HSK7

操作区	位置
程序	HSK6 和 HSK15 测量循环: HSK13 和 HSK14
程序管理器	HSK2-8 和 HSK12-16, 如未被驱动器占用。
诊断	HSK7
调试	HSK7

登入软键在专门的文件中设计。文件的名称在文件 `easyscreen.ini` 中声明。通常为一个操作区的名称（例如：调试区为 `startup.com`）。加工区有所例外，针对不同运行方式有不同文件 (`ma_jog.com`, `ma_auto.com`)。

带有登入软键的软键栏叫“**Start**”。现有的登入软键设计仍可使用。登入软键与登入软键菜单中各个 HMI 应用（操作区）中软键的汇总功能（Merge）还不被支持。

直到首次对话框调用时，即从全部功能（例如，执行 **PRESS** 块）都可用的时刻起，才能完全替换菜单或软键栏。

登入软键的配置模板

有关登入软键的全部允许位置的详细描述及其设计见以下目录中的文件 `easyscreen.ini`:

[系统西门子目录]/cfg

该文件用作登入软键的自定义配置模板。

另见

登入软键表 (页 219)

3.3.2 登入软键的功能

和对话框无关的软键的功能

通过登入软键只可以触发特定的功能。

允许下列功能：

- 通过功能 **LM** 可以装载另一个对话框。 **LM("名称"/,"文件")**
- 通过功能 **LS** 可以显示另一个软键栏。 **LS("名称"/,"文件"/[, 合并])**

3.3 登入软键的定义及功能

- 通过功能 "EXIT"可以离开新配置的操作界面并返回标准应用程序。
- 通过功能 "EXITLS"可以离开当前的操作界面并装载一个定义的软键栏。

PRESS 方法

在描述块内定义软键并在方法 PRESS 内分配功能 "LM" 或者 "LS"。

如果登入软键定义标记为注释（在行开始处用分号 ;）或者已删除配置文件，则登入软键无效。

```

//S(Start)                ; 开始标记
HS6=("第 1 屏幕")         ; 水平软键 6，文字为“第 1 屏幕”
PRESS(HS6)                ; 水平软键 6 的 PRESS 方法
    LM("屏幕 1")         ; 装载功能屏幕 1，此时必须在同一文件内定义了屏幕 1。
END_PRESS                 ; PRESS 方法结束标记
HS7=("第 2 屏幕")         ; 水平软键 7，文字为“第 2 屏幕”
PRESS(HS7)                ; 水平软键 7 的 PRESS 方法
LM("屏幕 2")             ; 装载功能屏幕 2，此时必须在同一文件内定义屏幕 2。
END_PRESS                 ; PRESS 方法结束标记
//END                     ; 登入块的结束标记
    
```

示例

```

HS1 = ("新的软键栏")
HS2 = ("没有功能")
PRESS(HS1)
    LS("软键栏 1")       ; 载入新的软键栏
END_PRESS
PRESS(HS2)                ; 空的 PRESS 方法
END_PRESS
    
```

各种登入软键设计

各个登入软键的设计会汇总在一起。其中，首先从 easyscreen.ini 中读取需要编译的文件的名称。然后在以下目录中查找后缀为 .com 的文件：

- [系统用户目录]/proj
- [系统 oem 目录]/proj

- [系统插件目录]/proj
- [系统西门子目录]/proj

此时所包含的登入软键设计会被汇总到一个设计中，即会对单个软键进行比较。如果对于一个软键存在两个或多个设计，则总是将较高的值接收到汇总版本。

可能包含的软键栏或对话框被忽略。如果一个软键包含一个无文件设定的指令，例如 LM("test")，因为所需的软键栏或者对话框包含在相同的文件中，则在内部汇总版本中会添加相应的文件名，这样就不需再进行调整。接下来显示得到的汇总设计。

3.4 错误处理（日志）

概述

如果“Run MyScreens”编译配置文件时出现错误，则该错误会保存在文本文件 easyscreen_log.txt 中。系统只记录当前选中的对话框的错误。之前选中的对话框中的错误记录则被删除。

文件包含以下信息：

- 在执行何种动作时出现错误
- 第一个错误字符的行号和列号
- 设计文件中所有的错误行
- DEBUG 功能生成的记录

说明

在每个记录前会附加一个带有方括号的当前时间戳。这可为（例如）分析对时间要求苛刻的配置提供帮助。

easyscreen-log.txt 的保存

文件 easyscreen_log.txt 保存在以下目录中：

[系统用户目录]/log

3.5 easyscreen.ini 的说明

句法

只有在定义了登入软键并且设计了带开始和结束标记的对话框和定义行后，才开始编译句法。

```

//S(Start)
HS6=("第 1 屏幕")
PRESS(HS6)
    LM("Maske1")
END_PRESS
//END

//M(Maske1)
DEF Var1=(R)
DEF VAR2 = (R)
LOAD
VAR1 = VAR2 + 1           ; 日志中的错误报告，因为 VAR2 没有数值
...
//END

; 正确，比如：
//M(Maske1)
DEF Var1=(R)
DEF VAR2 = (R)

LOAD
    VAR2 = 7
    VAR1 = VAR2 + 1 ;
...
    
```

3.5 easyscreen.ini 的说明

自 SINUMERIK Operate V4.7 起，easyscreen.ini 还扩展了本章节中描述的记录。easyscreen.ini 位于目录 [系统西门子目录]/cfg 下。

帮助画面起始位置

easyscreen.ini 中的记录:

```

[GENERAL]
HlpPicFixPos=true
    
```

说明:

帮助画面的起始位置会定位在所设置的像素位置，与分辨率无关（默认=true）。

拉伸特性，默认行高和行间距**easyscreen.ini 中的记录:**

```
[GENERAL]
SymmetricalAspectRatio=false
DefaultLineHeight=18
DefaultLineSpacing=3
```

说明:

- “SymmetricalAspectRatio” 来确定，根据 X 和 Y 方向上确定的屏幕分辨率来调整配置时是否要使用相同的拉伸系数。
 - “false”（默认）：使用宽屏分辨率时，各栏和图片会在 Y 方向上被压缩（基于 640x480 的非对称拉伸）。例如，一个以 640x480 设置的正方形在宽屏面板上会被垂直压缩，而成为一个长方形。
 - “true”：在 X 和 Y 方向上以相同的拉伸系数进行拉伸，此时各栏和图片会以 640x480 为基础并保持其原先设置的比例。例如，一个以 640x480 设置的正方形在宽屏面板上仍然会显示为正方形。
- “DefaultLineHeight” 和 “DefaultLineSpacing” 可在 640x480 的基础上确定默认行高（默认值：18 像素）和行间距（默认值：3 像素）。只有在短文本或输入/输出栏的位置设置中未给定 Y 位置或高度，此项设置才会一直有效。

与分辨率相关的屏幕位置：窗体面板**easyscreen.ini 中的记录:**

```
[640x480]
MyPanel = x:=0, y:=220, width:=340, height:=174
[800x480]
MyPanel = x:=0, y:=220, width:=420, height:=174
...
```

3.6 针对改用“Run MyScreens”人员的说明

说明：

屏幕位置可通过以下方式定义：

- 屏幕位置可以“基于 640x480 的像素”给定。

示例：

```
//M(MyMask/"MyCaption"/"\myhelp.png"/0,219,335,174)
```

- 屏幕位置可从 Operate 默认屏幕布局（Operate Standard-Screen-Layout）中，例如来自“机床”区域的屏幕布局“slmastandardscreenlayout.SIMaStandardScreenLayout”中的“FormPanel4”（“辅助功能”）

```
//M(MyMask/"MyCaption"/"\myhelp.png"/"slmastandardscreenlayout.SIMaStandardScreenLayout.FormPanel4")
```

连接到与分辨率相关的位置。

这样可更为方便地使 Operate 的默认窗体交叉淡入淡出，或者可将 Easyscreen 屏幕精确地放置到位。

示例：

```
//  
M(Mask/"Mask"/"slstandardscreenlayout.SIStandardScreenLayout.LowerForm")  
但也可以使用其他任意的屏幕布局。
```

- 屏幕位置可以连接到 easyscreen.ini 文件中自行定义的、与分辨率相关的窗体面板的定义。

示例：

```
//M(MyMask/"MyCaption"/"\myhelp.png"/"MyPanel")
```

3.6 针对改用“Run MyScreens”人员的说明

说明

使用 NCU 中的 HMI Operate 时要注意 CF 卡上的所有文件名都是以小写字母保存的（如：com、png、txt）。

说明

保存配置文件和语言文件时请注意，应将您所使用的编辑器中的编码设为 UTF 8。

图形文件

图形文件必须保存为 PNG 格式（xxx.png）。

数据（例如：用于 OEM 自定义的数据）必须保存在

[系统 oem 目录]/ico/[分辨率]

目录中。

更多信息另见 使用图/图形 (页 54) 一章。

调整配置文件

根据以下步骤检查配置文件：

- 比较登入软键和当前允许的软键（参见附录中的登入软键列表），必要时进行调整。
- 根据上述“图形文件”重命名相连的图形文件。

数据（例如：用于 OEM 自定义的数据）保存在以下目录中：

[系统 oem 目录]/proj

[系统用户目录]/proj

[系统插件目录]/proj

调整帮助文件

所有帮助文件必须以 UTF 8 格式保存。检查已有文件并使用合适的编辑器重新保存。

HTML 文件保存在以下目录中（例如：德语）：

[系统 oem 目录]/hlp/deu

[系统用户目录]/hlp/deu

[系统插件目录]/hlp/deu

其他语言的目录必须根据相应的语种缩写（参见附录）保存。

检查“Run MyScreens”授权

检查所添加的对话框数目是否超出了最大 5 个对话框的基本范围。

扩展对话框数量，需要下列软件选件：

- SINUMERIK 828D/840D sl, SINUMERIK Integrate Run MyScreens (6FC5800-0AP64-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyScreens + Run MyHMI (6FC5800-0AP65-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyHMI / 3GL (6FC5800-0AP60-0YB0)
- SINUMERIK 840D sl, SINUMERIK Integrate Run MyHMI / WinCC (6FC5800-0AP61-0YB0)

3.7 扩展配置句法

自 SINUMERIK Operate V4.7 起，提供了用于屏幕定义、变量定义、软键定义以及表格列定义的简化句法。该句法改善了可读性并缩短了等待时间。特性和属性可以任意顺序给定，空白记录被取消。相比于以前的句法，特性和属性列表不再使用圆括号“(”和“)”，而是使用圆弧括号“{”和“}”。

特性和属性按以下方式给定：

```
{<名称> = <值>, <名称> = <值>, ...}
```

以前的句法仍然兼容。

屏幕定义的扩展句法

```
//M {<屏幕名称> [,HD=<标题>] [,HLP=<图片>] [,X=<X 位置>] [,Y=<Y 位置>] [,W=<宽度>]
[,H=<高度>] [,VAR=<>系统变量或用户变量] [,HLP_X=<辅助画面 X 位置>] [,HLP_Y=<辅助画面 Y 位置>] [,CM=<列对齐>] [,CB=<对话框打开方式>] [,XG=<将辅助画面编译为 X3d 图片>] [,PANEL=<已链接 FormPanels 的名称>][,MC=<屏幕背景色>][,AL=<屏幕标题对齐>]}
```

示例：

```
//M{VariantTest, HD="My Mask"}
```

变量定义的扩展句法

DEF <变量名称> = {[TYP=<类型>] [,MIN=<最小值>] [,MAX=<最大值>] [,TGL=<转换值>] [,VAL=<预设值>] [,LT=<长文本>] [,ST=<短文本>] [,GT=<图片文本>] [,UT=<单位文本>] [,TT=<提示框文本>] [,TG=<转换选项>] [,WR=<输入模式>] [,AC=<访问级>] [,AL=<文本对齐>] [,FS=<字体大小>] [,LI=<极限值处理>] [,UR=<刷新率>] [,CB=<对话框打开方式>] [,HLP=<辅助画面>] [,VAR=<系统变量和用户变量>] >] [,TXT_X=<短文本 X 位置>] [,TXT_Y=<短文本 Y 位置>] [,TXT_W=<短文本宽度>] [,TXT_H=<短文本高度>] [,X=<输入/输出栏 X 位置>] [,Y=<输入/输出栏 Y 位置>] [,W=<输入/输出栏宽度>] [,H=<输入/输出栏高度>] [,BC=<输入/输出栏背景色>] [,FC=<输入/输出栏前景色>] [,BC_ST=<短文本背景色>] [,FC_ST=<短文本前景色>] [,BC_GT=<图片文本背景色>] [,FC_GT=<图片文本前景色>] [,BC_UT=<单位文本背景色>] [,FC_UT=<单位文本前景色>] [,SC1=<进度条信号色 1>] [,SC2=<进度条信号色 2>] [,SVAL1=<进度条阈值 1>] [,SVAL2=<进度条阈值 2>] [,DT=<显示类型>] [,DO=<显示对齐>] [,OHLP=<在线帮助>] [,LINK_TGL=<已链接转换变量的名称>]}

示例:

```
DEF MyVar5={TYP="R2", ST="MyVar5", VAL=123.4567, OHLP="myhelp.html", MIN=100.1,
MAX=200.9}
DEF MyVar2={TYP="I", TGL="*1,2,3", VAL=1}
DEF MyVar3={TYP="R2", TGL="*0=""Aus"", 1=$80000", VAL=1}
DEF MyVar4={TYP="R2", TGL="*MyArray",VAL=1}
DEF MyVar1={TYP="R2", TGL="%grid99", X = 0, W=300, H=200}
DEF MyVar6={TYP="R2", TGL="+$80000", VAR="$R[10]", ST="Textoffset"}
```

软键定义的扩展句法

SK = {[ST=<名称>] [,AC=<访问级>] [,SE=<状态>]}

示例:

```
HS1={ST=""MySk"", AC=6, SE=1}
HS3={ST="SOFTKEY_CANCEL"}
HS5={ST="[$81251,""\sk_ok.png""]}
HS8={ST="[""Test"",""\sk_ok.png""]}
```

表格列定义的扩展句法

{[TYP=<类型>][,MIN=<最小值>][,MAX=<最大值>][,LT=<长文本>][,ST=<短文本>][,WR=<输入模式>][,AC=<访问级>][,AL=<文本对齐>][,FS=<字体大小>][,LI=<极限值处理>][,UR=<刷新率>][,HLP=<辅助画面>][,VAR=<系统变量或用户变量>]>][,W=<列宽度>][,OF1=<偏移 1>][,OF2=<偏移 2>][,OF3=<偏移 3>]}

示例：

```
DEF MyGridVar={TYP="R", TGL="%MyGrid1", X=10, W=550, H=100}
//G (MyGrid1/0/5)
  {TYP="I", ST="Index", WR=1, VAR="1", W=80, OF1=1}
  {TYP="S", LT="LongText2", ST="Text", WR=1, VAR="$80000", AL=2, W=330, OF1=1}
  {TYP="R3", LT="LongText1", ST="R9,R11,R13,R15", WR=2, VAR="$R[1]", W=110, OF1=2}
//END
```


对话框

4.1 对话框的结构和组成单元

4.1.1 定义对话框

定义

对话框是操作界面上的一个组成部分，操作界面包含标题行、对话框单元和/或图形、显示消息的输出行以及 8 个水平软键和 8 个垂直软键。

对话框单元包含：

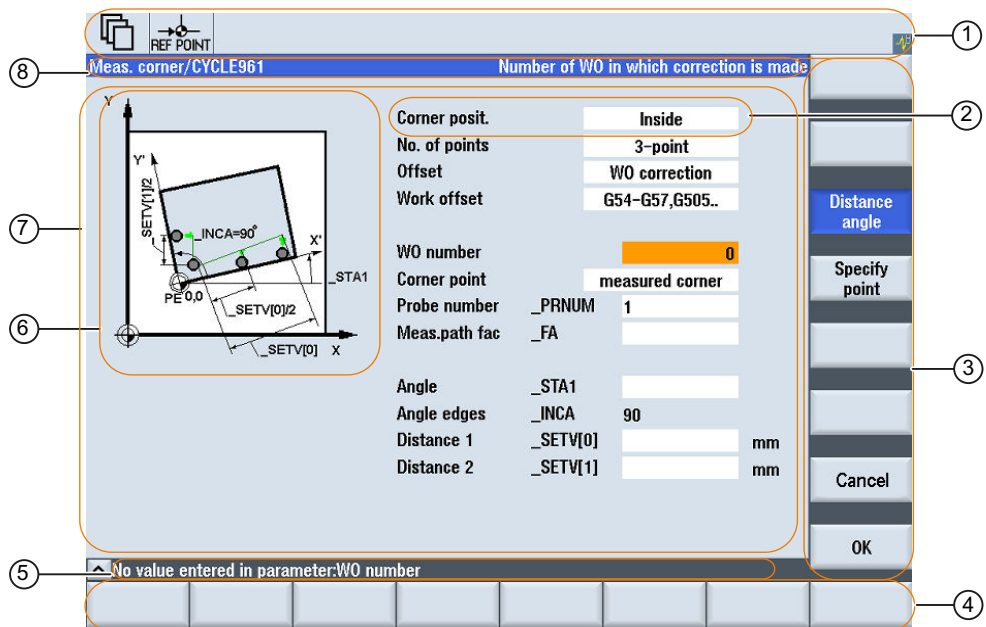
- 变量
 - 极限值/转换栏
 - 变量预设值
- 帮助图形
- 文本
- 属性
- 系统或者用户变量
- 短文本的位置
- 输入/输出栏的位置
- 颜色

对话框的属性：

- 标题
- 图形
- 尺寸
- 系统或者用户变量

4.1 对话框的结构和组成单元

- 图形的位置
- 属性



- ① 机床状态显示（“标题”）
- ② 对话框单元
- ③ 8个垂直软键
- ④ 8个水平软键
- ⑤ 诊断信息的输出
- ⑥ 图形
- ⑦ 对话框
- ⑧ 对话框的标题栏，包含标题和长文本

图 4-1 对话框结构

概述

基本上对话框说明（说明块）的结构如下：

说明块	注释	参考章节
//M...	；对话框的开始标记	
DEF Var1=...	；变量	参见章节“变量”
...		

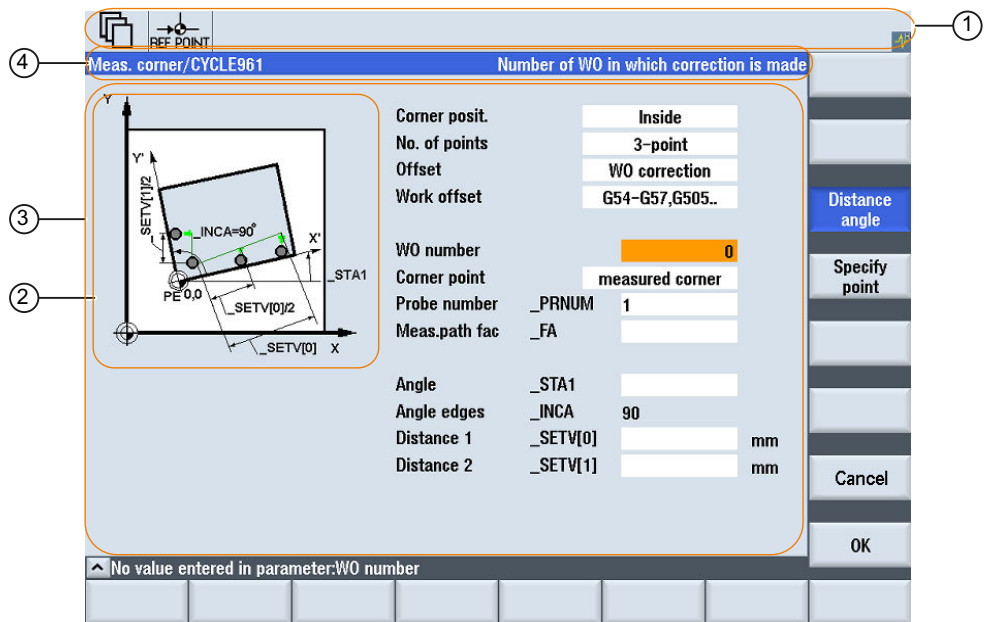
说明块	注释	参考章节
HS1=(...) ...	; 软键	参见章节“软键栏”
PRESS (HS1) LM... END_PRESS	; 方法的开始标记 ; 动作 ; 方法的结束标记	参见章节“方法”
//END	; 对话框的结束标记	

在对话框的说明块中，首先定义在对话框的对话框单元中显示的不同变量，然后定义水平和垂直软键。然后在方法中设计不同的动作。

4.1.2 定义对话框属性

说明

使用对话框的开始标记可以同时定义对话框的属性。



- ① 机床状态显示（“标题”）
- ② 图形
- ③ 对话框
- ④ 对话框的标题栏，包含标题和长文本

图 4-2 对话框属性

4.1 对话框的结构和组成单元

编程

句法:	//M(名称/[标题]/[图形]/[尺寸]/[系统或用户变量]/[图形位置]/[属性]) 另见章节 扩展配置句法 (页 38)。
说明:	定义对话框

参数:	名称		对话框的名称	
	标题		对话框的文本标题或者从和语言相关的文本文件中调用文本（例如： \$85011 ）	
	图形		图形文件，路径在双引号内	
	尺寸		对话框的位置和大小，单位像素（和左/右边缘的间距、宽度、高度），以屏幕的左上角为基准。数据值用逗号相隔。	
	系统或者用户变量		指定当前光标位置的系统和用户变量。可以通过系统或者用户变量将光标位置传送给 NC 或 PLC 。第一个变量索引为 1 。变量的顺序和变量设计顺序一致。	
	图形的位置		图形的位置，单位像素（和左/上边缘的间距），以对话框的左上角为基准。数据值用逗号相隔。	
	属性		给定的属性用逗号相隔。 允许的属性有：	
		CM		Column Mode: 列对齐
			CM0	预设置：每行单独分列。
			CM1	以包含最多列的行为标准分列。
		CB		CHANGE 块: 打开对话框时的属性：定义变量时规定的 cb 属性，优先于定义对话框时的总定义 另见 对话框打开方式（属性 CB ）（页 225）。
			CB0	预设置：在打开对话框时处理所有 CHANGE 部分。
			CB1	只有在附属的值改变后才处理 CHANGE 部分。
		XG		连接 X3D 动画作为辅助画面（只用在工步链编程中）。 示例： <pre>//M(Meas/ \$85605/"myx3dhelpfile.hmi,,Z_Animation,,G17"/// 30,10/XG1)</pre>
		XG0	缺省值 = 0	
		XG1	屏幕属性 XG 必须已在屏幕定义中设为 1 ，运行期间无法进行更改。 说明： 屏幕属性 HLP 中的数据也应进行相应的设置。	
	AL		使用屏幕属性 AL 可以控制屏幕标题的对齐。 示例： 窗口标题“Set password”居中	

4.1 对话框的结构和组成单元

		<code>//M(MY_PWD_SET/"Set password"//" EasyPwdModalLayout"//AL2)</code>
	AL0	左对齐, 缺省设置
	AL1	右对齐
	AL2	居中
	MC	<p>屏幕背景色(Mask Color)</p> <p>示例:</p> <p>设置屏幕背景色为蓝色(= 6)</p> <pre>//M(MyMask/"MyMask"/////6)</pre> <p>或者</p> <pre>PRESS (HS1) MC=6 END_PRESS</pre>

存取对话框属性

在方法（例如：PRESS 块）的范围内可以读取和写对话框的以下属性：

- HD = 标题 (Header)
- HLP = 辅助画面
- VAR = 系统或用户变量
- MC = 屏幕背景色
- CM = 列对齐（只读）
- CB = 打开方式（只读）
- XG = 连接 X3d（只读）
- AL = 屏幕标题对齐（只读）

示例

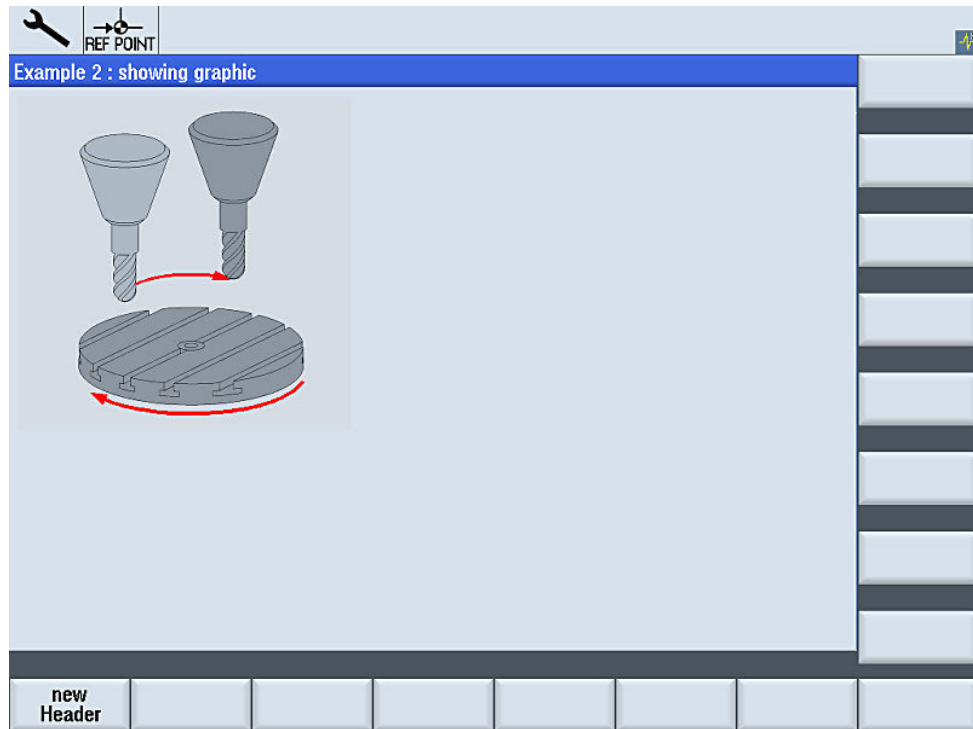


图 4-3 "Example 2: showing graphic"

```

//S(Start)
HS7=("Example", sel, ac7)

PRESS(HS7)
  LM("Mask2")
END_PRESS

//END
//M(Mask2/"Example 2 : showing graphic"/"example.png")
HS1=("new%nHeader")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")

```

4.1 对话框的结构和组成单元

```
VS2= ("")  
VS3= ("")  
VS4= ("")  
VS5= ("")  
VS6= ("")  
VS7= ("")  
VS8= ("")  
  
PRESS (HS1)  
    Hd= "new Header"  
END_PRESS  
...  
//END
```

参见

操作区“自定义”的编程示例 (页 209)

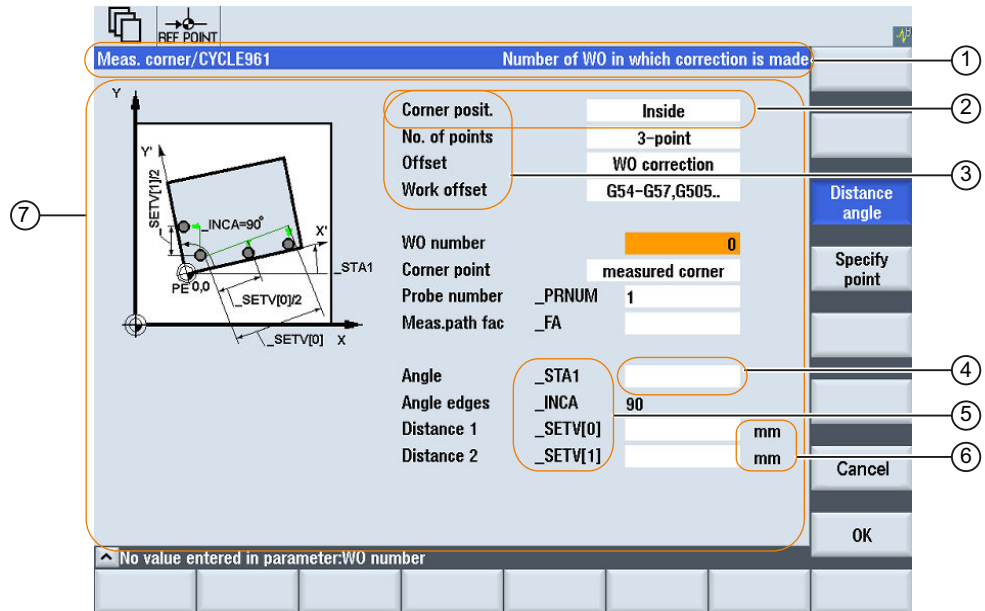
4.1.3 定义对话框单元

对话框单元

对话框单元是变量的可见部分，即短文本、图形文本、输入/输出栏、单位文本和提示框。对话框单元位于对话框主体的行中。每行可以定义一个或者多个对话框单元。

变量属性

所有变量仅在激活的对话框中有效。通过定义变量指定其属性。在方法（例如：PRESS 方法）的范围内可以存取对话框属性值。



- ① 对话框的标题栏，包含标题和长文本
- ② 对话框单元
- ③ 短文本
- ④ 输入/输出栏
- ⑤ 图形文本
- ⑥ 单位文本
- ⑦ 对话框主体

图 4-4 对话框单元

编程一览

在圆括号中通过逗号隔开各个参数。

DEF <i>Bezeichner</i> =	Bezeichner = 变量名称
	变量类型
	[/极限值或转换栏]
	[/预设值]
	[/文本（长文本，短文本 图，图形文本，单位文本）]

4.1 对话框的结构和组成单元

	/[属性]
	/[帮助画面]
	/[系统变量或者用户变量]
	/[短文本位置]
	/[输入栏/输出栏位置 (左、上、宽度、高度)]
	/[颜色]
	/[在线帮助] (页 62)

另见

变量参数 (页 77)

4.1.4 定义多列对话框

概述

在对话框中，一行可以显示多个变量。在这种情况下，设计文件中的所有变量都定义在一个定义行内。

```
DEF VAR11 = (S///"Var11"), VAR12 = (I///"Var12")
```

为了可以更清晰地显示设计文件中的各个变量，在每个变量定义和其紧随的逗号后可以换行。

关键字“DEF”总是表示新的一行的开始：

```
DEF Tnr1=(I//1/"", "T ", ""/wr1///, 10/20,, 50),
  TOP1=(I///, "Typ="/WR2//"$TC_DP1[1,1]"/80,, 30/120,, 50),
  TOP2=(R3///, "L1="/WR2//"$TC_DP3[1,1]"/170,, 30/210,, 70),
  TOP3=(R3///, "L2="/WR2//"$TC_DP4[1,1]"/280,, 30/320,, 70),
  TOP4=(R3///, "L3="/WR2//"$TC_DP5[1,1]"/390,, 30/420,, 70)
DEF Tnr2=(I//2/"", "T ", ""/wr1///, 10/20,, 50),
  TOP21=(I///, "Typ="/WR2//"$TC_DP1[2,1]"/80,, 30/120,, 50),
```

```
TOP22=(R3///,"L1="/WR2//"$TC_DP3[2,1]"/170,,30/210,,70),  
TOP23=(R3///,"L2="/WR2//"$TC_DP4[2,1]"/280,,30/320,,70),  
TOP24=(R3///,"L3="/WR2//"$TC_DP5[2,1]"/390,,30/420,,70)
```

说明

在设计多列对话框时注意，列过多可能会导致系统变慢！

4.1.5 口令对话框

使用标准口令对话框

您可在屏幕配置中连接以下预定义的口令对话框：

- 设置口令

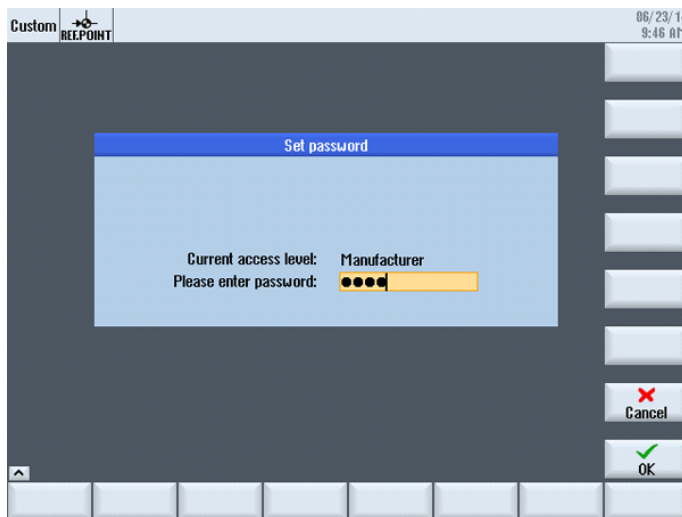


图 4-5 设置口令对话框

- 更改口令

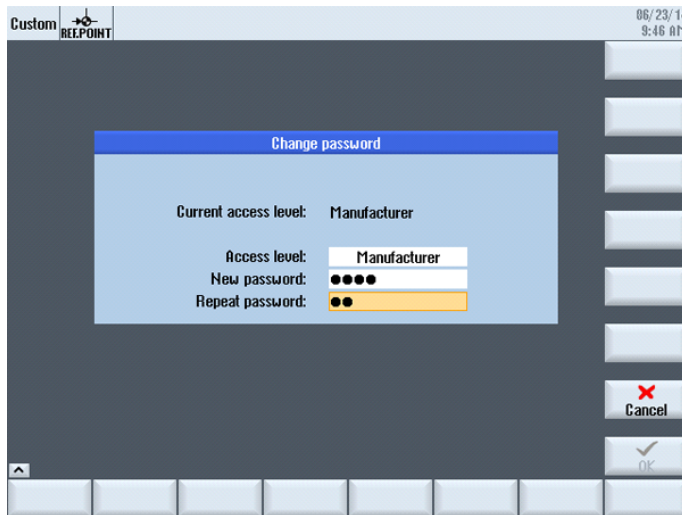


图 4-6 更改口令对话框

- 删除口令

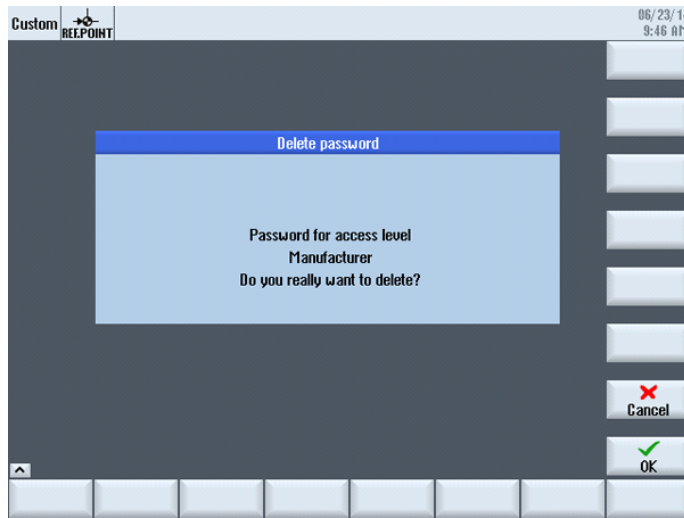


图 4-7 删除口令对话框

说明

使用这些对话框可为您提供有关口令的各项功能。这些对话框与 SINUMERIK Operate 的对话框并不一样。

调用对话框既可以通过功能 Load Softkey (LS)进行，也可以通过 Load Mask (LM)进行：

1. 通过功能 Load Softkey (LS)调用：

```
LS ("Passwd", "slespasswd.com", 1)
```

垂直软键扩展：

- 软键 1： 设置口令
- 软键 2： 更改口令
- 软键 3： 删除口令

2. 或者也可以直接通过调用功能 Load Mask (LM)跳转到这三个屏幕：

- 设置口令： LM("PWD_SET", "slespasswd.com", 1)
- 更改口令： LM("PWD_CHG", "slespasswd.com", 1)
- 删除口令： LM("PWD_CLEAR", "slespasswd.com", 1)

4.2 定义软键栏

4.1.6 使用图/图形

使用图形

分为:

- 使用图/图形
- 帮助图形，例如：图示说明各个变量并突出显示在图形区中。
- 可以设计其它帮助图形替代可任意定位的短文本或者输入/输出栏。

保存位置

与相连监视器的分辨率相匹配的图片，请先从相应的分辨率目录下按顺序查找：

[系统用户目录]/ico/ico<分辨率>

[系统 oem 目录]/ico/ico<分辨率>

[系统插件目录]/ico/ico<分辨率>

如果未显示或未找到图片，将其复制到分辨率为 640 x 480 像素的目录中：

[系统用户目录]/ico/ico640

[系统 oem 目录]/ico/ico640

[系统插件目录]/ico/ico640

说明

在面板分辨率不同时，图片会按比例定位。

4.2 定义软键栏

定义

所有水平软键和所有垂直软键分别布置在一起称作软键栏。另外针对已有的软键栏，还可以定义其它软键栏，可以部分或者完全覆盖已有的软键栏。

确定软键名称。不必占用所有软键。

HSx x 1 - 8, 水平软键 1 至 8

VSy y 1 - 8, 垂直软键 1 至 8

原则上软键栏描述（描述块）如下构建：

说明块	注释	参考章节
//S...	; 软键栏开始标记	
HSx=...	; 定义软键	
PRESS (HSx) LM...	; 方法的开始标记 ; 动作	参见章节“方法”
END_PRESS	; 方法的结束标记	
//END	; 软键栏结束标记	

说明

定义软键栏也同时分配软键属性。

编程

句法:	//S(名称) ... //END	;软键栏开始标记 ;软键栏结束标记
	另见章节 扩展配置句法 (页 38)。	
说明:	定义软键栏	
参数:	名称	软键栏名称
	文本或图形文件名称	
句法:	SK = (文本[, 访问等级[, 状态[, 软键图对齐[, 以软键图为基准的文本对齐])])	
说明:	定义软键	
参数:	SK	软键, 例如: HS1 到 HS8, VS1 到 VS8

4.2 定义软键栏

文本	给定文本										
图文件名称	<p>"\\my_pic.png"</p> <p>或者通过单独的文本文件 \$85199, 例如: (和语言相关的) 文本文件中的下列文本: 85100 0 0 "\\my_pic.png"。</p> <p>所能显示的软键图片的大小取决于所使用的 OP:</p> <table border="1"> <tr> <td>OP 08:</td> <td>640 x 480 mm → 25 x 25 像素</td> </tr> <tr> <td>OP 010:</td> <td>640 x 480 mm → 25 x 25 像素</td> </tr> <tr> <td>OP 012:</td> <td>800 x 600 mm → 30 x 30 像素</td> </tr> <tr> <td>OP 015:</td> <td>1024 x 768 mm → 40 x 40 像素</td> </tr> <tr> <td>OP 019:</td> <td>1280 x 1024 mm → 72 x 72 像素</td> </tr> </table>	OP 08:	640 x 480 mm → 25 x 25 像素	OP 010:	640 x 480 mm → 25 x 25 像素	OP 012:	800 x 600 mm → 30 x 30 像素	OP 015:	1024 x 768 mm → 40 x 40 像素	OP 019:	1280 x 1024 mm → 72 x 72 像素
OP 08:	640 x 480 mm → 25 x 25 像素										
OP 010:	640 x 480 mm → 25 x 25 像素										
OP 012:	800 x 600 mm → 30 x 30 像素										
OP 015:	1024 x 768 mm → 40 x 40 像素										
OP 019:	1280 x 1024 mm → 72 x 72 像素										
访问等级	ac0 到 ac7 (ac7: 缺省设置)										
状态	<p>se1: 可见 (预设置)</p> <p>se2: 不可操作 (灰色标签)</p> <p>se3: 高亮显示 (最后操作的软键)</p>										
软键图对齐	<p>PA (PictureAlignment)</p> <p>有效值:</p> <p>0: 左</p> <p>1: 右</p> <p>2: 居中</p> <p>3: 上 (默认)</p> <p>4: 下</p>										
以软键图为基准的文本对齐	<p>TP (TextAlignedToPicture)</p> <p>有效值:</p> <p>0: 文本和图形未对齐</p> <p>1: 文本和图形已对齐 (默认)</p>										

说明

对于软键标签, 用 %n 进行换行。

最多 2 行, 每行各 9 个字符。

分配访问等级

操作员只能访问符合其访问等级和各个低于其访问等级的信息（另见章节 访问等级列表 (页 222)）。

示例

//S (软键栏 1)	; 软键栏开始标记
HS1=("新建", ac6, se2)	; 定义软键 HS1: 软键标签为“新建”，保护等级为 6，状态为“不可操作”。
HS2=("\\bild1.png")	; 指定软键图片
HS3=("退出")	
HS4=(["Confirm", "\\sk_ok.png"], PA0, TP1)	; 带文本和图形的软键，文本="Confirm"，图形="sk_ok.png"，软键图对齐：左，文本 和图形对齐
VS1=("子窗口")	
VS2=(\$85011, ac7, se2)	; 定义软键 VS2: 软键标签来自语言文件，保护等级为 1，状态为“不可操作”
VS3=("取消", ac1, se3)	; 定义软键 VS3: 软键标签为“取消”，保护等级为 1，状态为“高亮”
VS4=("确认", ac6, se1)	; 定义软键 VS4: 软键标签为“确认”，保护等级为 6，状态为“可见”
VS5=(SOFTKEY_CANCEL,,se1)	; 定义标准软键 VS5: 软键标签为“取消”，状态为“可见”
VS6=(SOFTKEY_OK,,se1)	; 定义标准软键 VS6: 软键标签为“确认”，状态为“可见”
VS7=("\\bild1.png", "OEM 文本",,,se1)	; 定义软键 VS7: 指定图片，软键标签为“OEM 文本”，状态为“可见”
VS8=("\\bild1.png", \$83533,,se1)	; 定义软键 VS8: 指定图片，软键标签来自语言文件，状态为“可见”
PRESS (HS1)	; 方法开始标记
HS1.st="计算"	; 指定软键标签
...	
END_PRESS	; 方式结束标记
PRESS (RECALL)	; 方法开始标记
LM("屏幕 21")	; 载入对话框
END_PRESS	; 方式结束标记
//END	; 软键栏结束标记

4.2.1 运行时改变软键属性

说明

文本、访问等级和状态的属性可以在运行期间在方法中改变：

4.2 定义软键栏

编程

句法:	SK.st = "文本" SK.ac = 访问等级 SK.se = 状态 SK.pa = "软键图对齐" SK.tp = "以软键图为准的文本对齐"	; 软键标签 ; 软键访问等级 ;软键状态 ; 带图形的软键 ; 带图形和标签的软键
说明:	分配属性	
参数:	文本	引号中的标签文本
	访问等级	值范围: 0 ... 7
	状态	1: 可见并可操作 2: 不可操作 (灰色标签) 3: 高亮显示 (最后操作的软键)
	软键图对齐	0: 左 1: 右 2: 居中 3: 上 (默认) 4: 下
	以软键图为准的文本对齐	0: 文本和图形未对齐 1: 文本和图形已对齐 (默认)

示例

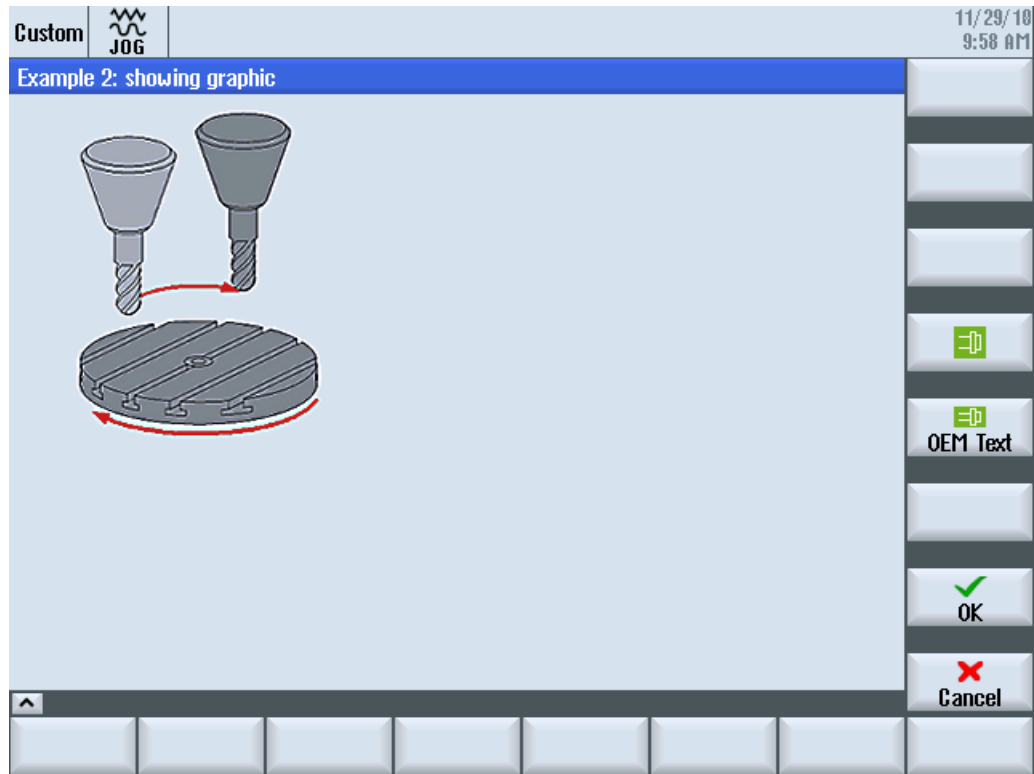


图 4-8 示例 3: 图形和软键

```

//S(Start)
HS7=("Example", ac7, se1)

PRESS(HS7)
  LM("Maske3")
END_PRESS

//END

//M(Maske3/"Example 2: showing graphic"/"example.png")
HS1=("")
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")

```

4.2 定义软键栏

```
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("\\sp_ok.png",,SE1)
VS5=(["\\sp_ok_small.png","OEM Text"],,SE1)
VS6=("")
VS7=(SOFTKEY_OK,,SE1)
VS8=(SOFTKEY_CANCEL,,SE1)
PRESS(VS4)
    EXIT
END_PRESS
PRESS(VS5)
    EXIT
END_PRESS
PRESS(VS7)
    EXIT
END_PRESS
PRESS(VS8)
    EXIT
END_PRESS
//END
```

4.2.2 和语言相关的文本

概述

使用和语言相关的文本：

- 软键标签
- 标题
- 辅助文本
- 其它任意文本

对话框中与语言相关的文本保存在文本文件中。

文本文件位于以下目录中：

- [系统用户目录]/lng
- [系统 oem 目录]/lng
- [系统插件目录]/lng

说明

文本文件必须如同项目文件一样进行保存。

例如：

[系统用户目录]/lng/[文本文件]

[系统用户目录]/proj/[配置文件]

alsc.txt 用于西门子标准循环、和语言相关的文本

almc.txt 用于西门子测量循环、和语言相关的文本

程序运行中所使用的文本文件在 **easyscreen.ini** 文件中给定：

```
[LANGUAGEFILES]
LngFile03 = user.txt            ;->user<_xxx>.txt (例如: user_eng.txt)
```

文件 **user.txt** 在这里是作为文本文件的示例。原则上可自由选择名称。根据文件中文本的不同语种，还必须按照下面的句法附加上各语言的缩写。在名称后加上一个下划线以及相应的语种缩写，例如 **user_eng.txt**。

说明

由于 **LngFile01** 和 **LngFile02** 已经指定给标准 **easyscreen.ini**，因而不可用于用户文本。

另见

文件名中的语种缩写表 (页 224)

文本文件的格式

文本文件必须保存为 **UTF-8** 编码格式。

说明

保存配置文件和语言文件时请注意，应将您所使用的编辑器中的编码设为 **UTF 8**。

4.3 设计在线帮助

文本输入形式

句法:	8xxxx 0 0 “文本”		
说明:	文件中文本号和文本的排列		
参数:	xxxx	85000 到 89999	预留于用户的文本识别号范围。号码必须是唯一的。
	“文本”		显示在对话框中的文本
	%n		文本中用于分行的控制符

两个通过空格分开的参数 2 和 3 是用于报警文本输出的控制符。鉴于与报警文本的格式一致性，在任何情况下控制符都必须置零。

和语言相关的文本示例:

85000 0 0 “退回平面”

85001 0 0 “钻削深度”

85002 0 0 “螺距”

85003 0 0 “凹槽半径”

4.3 设计在线帮助

在线帮助

可以 HTML 格式创建对配置的对话框和单元的在线帮助。

创建在线帮助的句法和步骤与 SINUMERIK Operate 相似。

如需为输入栏设计在线帮助，则会使用在线帮助的标准屏幕，以显示针对用户的在线帮助:

```
DEF RFP=(R//1/, "RFP", "RFP"/////////"sinumerik_md_1.html", "9006")
```

说明

LINUX 系统要求 HTML 文件名称必须以小写字母保存!

HTML 文件保存在以下目录中 (例如: 德语)。

[系统用户目录]/hlp/deu

[系统 oem 目录]/hlp/deu

[系统插件目录]/hlp/deu

其他语言的目录必须根据相应的语种缩写（参见附录）保存。

文档

调试手册“基本软件和操作软件”(IM9)，章节“针对 OEM 的在线帮助”

4.3 设计在线帮助

变量

5.1 定义变量

变量值

变量的重要属性即变量值。

可以通过如下方式赋值：

- 定义变量时预设
- 分配系统或者用户变量
- 采取方法

编程

句法:	名称 val = 变量值 名称 = 变量值	
说明:	变量值 val (value)	
参数:	名称:	变量名称
	变量值	变量值
示例:	VAR3 = VAR4 + SIN(VAR5) VAR3.VAL = VAR4 + SIN(VAR5)	

变量状态

通过变量状态的属性可以在运行时查询，变量是否包含有一个有效值。该属性可通过值 **FALSE = 0** 读写。

编程

句法:	名称. vld
说明:	变量状态 vld (validation)

5.2 应用举例

参数:	名称:	变量名称
	FALSE = TRUE =	查询结果可能是: 无效值 有效值
示例:	<pre>IF VAR1.VLD == FALSE VAR1 = 84 ENDIF</pre>	

5.2 应用举例

辅助变量

辅助变量是内部计算变量。计算变量如同变量一样定义，但是另外除了变量值和状态之外没有属性，即辅助变量在对话框中不可见。辅助变量是 **VARIANT** 类型。

编程

句法:	DEF 名称	
说明:	VARIANT 类型的内部计算变量	
参数:	名称:	辅助变量名称
示例:	DEF OTTO; 定义一个辅助变量	

句法:	名称 val = 辅助变量值	
	名称 = 辅助变量值	
说明:	辅助变量值在方式中指定。	
参数:	名称:	辅助变量名称
	辅助变量值:	辅助变量内容

示例:

```
LOAD
    OTTO = "Test"           ; 为辅助变量"Test" 赋值 Otto
END_LOAD
LOAD
    OTTO = REG[9].VAL      ; 为辅助变量 Otto 赋寄存器的值
END_LOAD
```

通过变量计算

在每次退出输入/输出栏（通过 **ENTER** 或者转换键）后计算变量。计算在 **CHANGE** 方式中设计并在每次更改值时运算。

通过变量状态可以查询，变量是否包含有效值，例如：

```
IF VAR1.VLD == FALSE
    VAR1 = 84
ENDIF
```

系统变量间接编译地址

系统变量也可以间接编译地址，即和另一个变量相关。

```
PRESS (HS1)
    ACHSE=ACHSE+1
    WEG.VAR="$AA_DTBW["<<ACHSE<<"]"      ; 通过变量编译轴地址
END_PRESS
```

更改软键标签**示例:**

```
HS3.st = "新文本"           ; 更改软键标签
```

5.3 示例 1: 分配变量类型、文本、帮助画面、颜色、提示框

示例 1a

下面的示例中会定义一个变量，并对它的变量类型、文本、辅助画面和颜色属性进行设置。

DEF Var1 = (R///,"实际值",,"mm"//"/"Var1.png"////8,2)		
	变量类型:	REAL
	文本:	
	短文本:	实际值
	单位文本:	mm
	帮助图形:	Var1.png
	颜色:	
	前景色:	8 (棕色)
	背景色:	2 (橙色)

5.3 示例 1: 分配变量类型、文本、帮助画面、颜色、提示框

示例 1b

下面的示例中会定义一个变量，并对它的变量类型、预设值、文本、提示框、输入模式和短文本位置属性进行设置。

DEF Var2 = (I/5/"", "值", "", "", "提示框文本"/wr2///20,250,50)		
	变量类型:	INTEGER
	预设值:	5
	文本:	
	短文本:	值 (可能的语言文本 ID)
	提示框:	提示框文本
	属性:	
	输入模式	读和写
	短文本位置:	
	到左边缘的间距	20
	到上边缘的间距	250
	宽度:	50

参见

变量参数 (页 77)

5.4 举例 2: 定义变量类型, 极限值, 属性, 短文本位置

示例 2

下面的示例中会定义一个变量, 并对它的变量类型、极限值、输入模式、对齐和位置属性进行设置。

DEF Var2 = (I/O,10//wr1,al1/// , ,300)		
	变量类型:	INTEGER
	极限值或者转换栏输入项:	MIN: 0 MAX: 10
	属性:	
	输入模式	只读
	短文本的文本对齐	右对齐
	短文本位置:	
	宽度:	300

参见

变量参数 (页 77)

5.5 举例 3: 定义变量类型、预设、系统或者用户变量、输入/输出栏位置

示例 3

下面的示例中会定义一个变量，并对它的变量类型、预设值、系统变量或用户变量和位置属性进行设置。

DEF Var3 = (R//10///"\$R[1]"//300,10,200//)		
变量类型:		REAL
预设值:		10
系统变量或者用户变量:		\$R[1] (R 参数 1)
短文本位置:		相对于输入/输出栏的标准位置
输入/输出栏位置:		
	到左边缘的间距	300
	到上边缘的间距	10
	宽度:	200

参见

变量参数 (页 77)

5.6 示例 4: 转换栏和列表栏

示例 4a

转换栏中不同的条目:

DEF Var1 = (I/* 0,1,2,3)	: 用于数值转换的简单转换栏
DEF Var2 = (S/* "On", "Off")	: 用于字符串转换的简单转换栏
DEF Var3 = (B/* 1="On", 0="Off")	: 用于数值转换的扩展转换栏, 为每个数值分配了一个显示文本
DEF Var4 = (R/* ARR1)	: 转换栏中要转换的值基于一个数组

5.6 示例 4: 转换栏和列表栏

示例 4b

列表栏与转换栏的配置一样，但要额外设置列表栏的显示类型（变量属性 DT = 4）。

DEF VAR_LISTBOX_Text = (S/*\$80000,\$80001,\$80002,\$80003,\$80004/\$80001//DT4////200,,340,60)		
	变量类型:	字符串
	极限值/转换栏:	* \$80000,\$80001,\$80002,\$80003,\$80004 4 (要显示的与语言相关的文本列表)
	预设值:	\$80001
属性:		
	显示类型:	4 (列表栏)
输入/输出栏位置:		
	左边距:	200
	宽度:	340
	高度:	60
颜色:		
	前景色:	6 (蓝色)
	背景色:	10 (白色)

5.7 示例 5: 图片显示

示例 5

显示图片，而不是短文本：图片的大小和位置在“输入/输出栏位置（左侧、上部、宽度、高度）”中规定。

DEF VAR6 = (V///,"\\bild1.png" ///160,40,50,50)		
	变量类型:	VARIANT
	文本:	
	短文本:	bild1.png
	输入输出栏的位置:	
	左边距:	160
	上边距:	40
	宽度:	50
	高度:	50

5.8 示例 6: 进度条

进度条是一种特殊的输入/输出栏显示类型并且只用于显示而不能输入。

原则上有两种进度条类型:

1. 最多可进行两次变色的进度条，例如温度或负载显示（见示例 6a）。
2. Operate 风格的进度显示（不变色）条（见示例 6a）

示例 6a

两次变色的进度条:

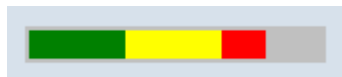


图 5-1 两次变色的进度条

5.8 示例 6: 进度条

DEF PROGGY0 = (R/0,150,50,100//DT1,DO0/"\$R[10]"//,,150/3,4,,,,,9,7)		
变量类型:		REAL
极限值/转换栏:		
	MIN:	0
	MAX:	150
	信号值 SVAL1:	50
	信号值 SVAL2:	100
属性:		
	显示模式 DT:	1 (进度条)
	显示选项 DO:	0 (从左往右 (默认))
系统变量或者用户变量:		\$R[10]
输入/输出栏位置:		
	宽度:	150
颜色:		
	前景色:	3 (深绿)
	背景色:	4 (浅灰)
	信号色 SC1:	9 (黄色)
	信号色 SC2:	7 (红色)

如要使用可变色的进度条应将显示模式 DT (DisplayType) 设为 1。

进度显示的方向通过属性 “显示选项 DO (DisplayOption)” 来确定:

0: 从左往右 (默认)

1: 从右往左

2: 从下往上

3: 从上往下

进行进度条的显示时, 应给定最小值和最大值 (示例中: MIN: 0, MAX: 150)。

在此设置下, 取决于变量 PROGGY0 的当前值, 进度条将显示前景色 3 (= 深绿) 和背景色 4 (= 浅灰)。

另外还可以选择定义一或两个信号值 SVAL1 和 SVAL2 (参数限值) (示例中: SVAL1: 50 以及 SVAL2: 100)。达到这两个信号值时, 进度显示的前景色会变化。相应的信号色通过参数 SC1 和 SC2 确定 (上例中 SC1: 9 (= 黄色) 以及 SC2: 7 (= 红色))。

进度显示的限值给定适用如下规则:

MIN < SVAL1 < SVAL2 < MAX。

示例 6b

不变色进度条:



图 5-2 进度条

DEF PROGGY0 = (R/0,150//DT2,DO0/!\$"R[10]"//,,150/6,10)		
变量类型:		REAL
极限值/转换栏:		
	MIN:	0
	MAX:	150
属性:		
	显示类型 DT:	2 (进度条)
	显示选项 DO:	0 (从左往右 (默认))
系统变量或者用户变量:		\$R[10]
输入/输出栏位置:		
	宽度:	150
颜色:		
	前景色:	6 (蓝色)
	背景色:	10 (白色)

如要使用不变色的进度条应将显示模式 DT (DisplayType) 设为 2。

进度显示的方向通过属性“显示选项 DO (DisplayOption)”来确定 (参见示例 6a 的说明)：

进行进度条的显示时，应给定最小值和最大值 (示例中：MIN: 0, MAX: 150)。

在此设置下，取决于变量 PROGGY0 的当前值，进度条将显示前景色 6 (= 蓝色) 和背景色 10 (= 白色)。

5.9 示例 7: 口令输入模式 (星号)

5.9 示例 7: 口令输入模式 (星号)

示例 7

如要对输入栏中的输入内容实现隐藏，例如在输入口令时，必须将显示模式 DT 设为 5。所输入的字符会显示为星号。



图 5-3 口令输入模式 (星号)

DEF VAR_SET_PWD=(S//"/DT5)		
	变量类型:	STRING
	预设值:	空字符串
	属性:	
	显示模式 DT:	5 (口令输入模式)

5.10 变量参数

参数 - 一览

在以下概述中简要说明了变量参数。详细的说明请参见后续章节。

参数	说明
变量类型 (页 84)	必须规定变量类型。
	<p>R[x]: REAL (小数点位数为正数 +)</p> <p>I: INTEGER</p> <p>S[x]: 字符串 (字符串长度为正数 +)</p> <p>C: 字符 (单字符)</p> <p>B: BOOL</p> <p>V: VARIANT</p>
极限值 (页 70)	<p>最小极限值, 最大极限值</p> <p>预设置: 空</p> <p>极限值通过逗号隔开。类型 I、C 和 R 的极限值可以使用十进制的格式或者字符 "A"、"F" 定义。</p> <p>对于可变色的进度条显示 (变量属性 DT = 1), 可选择配置两个信号色 SC1 和 SC2 (Signal Color), 在超出信号值 SVAL1 或 SVAL2 (Signal Value) 时会使用相应的进度条前景色。信号值为整型值。</p> <p>参见进度条 (页 73) 应用示例。</p>
预设值 (页 89)	<p>如果没有定义任何预设值并且没有分配系统或者用户变量给变量, 则分配转换栏的第一个单元。如果没有定义转换栏, 则不进行预设, 即: 变量处于状态 "未计算"。</p> <p>预设置: 未预设</p>
转换栏 (页 88)	<p>输入/输出栏中带有预设输入项的列表: 列表通过 * 开始, 各输入项用逗号隔开。可以赋值输入项。</p> <p>极限值的输入项视为转换栏的列表。如果只输入一个 *, 则建立一个可变的转换栏。</p> <p>预设置: 无</p>

5.10 变量参数

参数	说明	
文本 (页 68)	顺序已预先规定。也可以显示一个图形代替短文本。 预设置：空	
	长文本： 短文本： 图形文本： 单位文本： 提示框	显示行的文本 对话框单元的名称 文本参考图形名称。 对话框单元的单位 在设计窗口时，用于为各个显示栏和切换栏提供简短的说明信息。该信息通过纯文本和语言文本 ID 设计。

参数	说明
属性 (页 70)	<p>属性影响下列特性:</p> <ul style="list-style-type: none"> ● 显示模式 ● 显示选项 ● 刷新率 ● 转换符号 ● 提示框 ● 输入模式 ● 访问等级 ● 短文本的文本对齐 ● 字体大小 ● 极限值 <p>属性通过逗号隔开, 顺序任意 每个组件都可以进行定义。</p>
	<p>显示模式</p> <p>dt0: 标准 (输入/输出栏或转换栏) (默认)</p> <p>dt1: 可变色的进度条:</p> <p>dt2: Operate 风格的不变色进度条</p> <p>dt4: 列表栏</p> <p>dt5: 口令输入模式 (星号)</p> <p>dt6: 多行显示的输入/输出栏(multiline)</p>
	<p>显示选项</p> <p>特别是对于显示模式 dt1 和 dt2 (进度条), 还需要同时设置显示选项。</p> <p>do0: 从左往右 (默认)</p> <p>do1: 从右往左</p> <p>do2: 从下往上</p> <p>do3: 从上往下</p>
	<p>刷新速度</p> <p>使用属性 UR (update rate)可以控制显示的刷新以及相应配置的变量或 Grid 列的 CHANGE 块的处理。根据配置的不同, CPU 负载可能被大幅降低并且可达到更短的用户接口响应时间。</p> <p>ur0: SLCap::standardUpdateRate() (当前 = 200 ms, 默认值)</p> <p>ur1: 50 ms</p> <p>ur2: 100 ms</p> <p>ur3: 200 ms</p> <p>ur4: 500 ms</p>

5.10 变量参数

参数	说明
	<p>ur5: 1000 ms ur6: 2000 ms ur7: 5000 ms ur8: 10000 ms</p>
转换符号	<p>tg0: 转换符号关闭 (默认) tg1: 转换符号打开 如果将属性 TG 设为 1, 则在输入栏提示框中还会额外显示转换符号。 示例: DEF OFFS = (R//123.456/,,,,,"My ToolTip"/TG1)</p>
提示框	<p>提示框文本在运行期间可通过变量属性 "TT" 来更改。 示例: PRESS (VS1) MyVar.TT = "My new ToolTip" END_PRESS 或者 DEF MyVar=(R///,,,,,"My ToolTip-text")</p>
输入模式	<p>wr0: 输入/输出栏不可见, 短文本可见 wr1: 读取 (没有输入中心) wr2: 读取和写入 (行以白色显示) wr3: wr1 带输入中心 wr4: 所有变量单元不可见, 没有输入中心 wr5: 按下任何键立即保存输入的值 (和 wr2 相反 - 该模式下, 在退出栏或者按下返回键后才开始保存值)。 预设置: wr2</p>
访问等级 (页 222)	<p>空: 总是可以写入 ac0...ac7: 保护等级 如果访问等级未达到, 行显示为灰色, 标准设置: ac7</p>
短文本的文本 对齐	<p>al0: 左对齐 al1: 右对齐 al2: 居中 预设置: al0</p>

参数	说明	
	字体大小	fs1: 标准字体大小 (8 Pt) fs2: 双倍字体大小 预设置: fs1 确定行间距。标准字体大小为每对话框 16 行。图形和单位文本只能为标准字体大小。
	极限值	通过极限值可以检查变量值是否在规定的最小极限值和最大极限值之内。 预设置: 取决于规定的极限值 li0: 没有检查 li1: 检查最小极限值 li2: 检查最大极限值 li3: 检查最大极限值和最小极限值
	打开时的属性	定义变量时规定的 cb 属性, 优先于定义对话框时的总定义。多个属性通过逗号隔开记录。(另见 对话框打开方式 (属性 CB) (页 225))
	CB0: CB1:	如果变量当前有一个有效值 (例如: 通过预设值或 NC/PLC 变量), 则通过显示屏幕触发 CHANGE 方法。 通过显示屏幕无法显式触发 CHANGE 方法。如果变量有一个配置的 NC/PLC 变量, 则一定会调用 CHANGE 方法。
帮助画面 (页 68)	帮助画面文件:	png 文件的名称 预设置: 空
	帮助画面文件的名称在双引号中。如果光标移至该变量, 则自动显示画面 (替代目前的图形)。	
系统或者用户变量 (页 71)	可以将一个来自 NC/PLC 的系统数据或用户数据指定给变量。系统或者用户变量用双引号括起。 参考文献: 参数手册, 系统变量, /PGAsI/	
短文本的位置 (页 90)	短文本位置 (和左边缘/上边缘的间距、宽度) 位置数据的单位为像素, 并且以对话框主体的左上角为基准。数据总是用逗号隔开。	
输入/输出栏的位置 (页 90)	输入/输出栏位置 (和左边缘/上边缘的间距、宽度、高度) 位置数据的单位为像素, 并且以对话框主体的左上角为基准。数据总是用逗号隔开。如果改变此位置, 则短文本、图形文本和单位文本的位置也同时改变。	

参数	说明
颜色 (页 68)	<p>输入/输出栏、短文本、图形文本、单位文本的前景色和背景色及进度条的信号色。</p> <p>颜色通过逗号隔开。</p> <p>颜色数据参见章节 颜色表 (页 222)。</p> <p>输入/输出栏的缺省设置：前景色：黑色，背景色：白色</p> <p>输入/输出栏的标准颜色取决于写入模式“wr”：</p> <p>短文本、图形文本、单位文本的缺省设置：前景色：黑色；背景色：透明的。</p> <p>对于可变色的进度条显示（变量属性 DT = 1），可选择配置两个信号色 SC1 和 SC2 (Signal Color)，在超出信号值 SVAL1 或 SVAL2 (Signal Value)时会使用相应的进度条前景色。</p> <p>参见进度条 (页 73)应用示例。</p> <p>变量定义中要求的颜色顺序如下：</p> <ol style="list-style-type: none"> 1. 输入/输出栏的前景色：FC 2. 输入/输出栏的背景色：BC 3. 短文本的前景色：FC_ST 4. 短文本的背景色：BC_ST 5. 图形文本的前景色：FC_GT 6. 图形文本的背景色：BC_GT 7. 单位文本的前景色：FC_UT 8. 单位文本的背景色：BC_UT 9. 信号色 1 10 信号色 2 <p>.</p>

参数	说明
在线帮助文件 (页 62)	在线帮助文件的名称在双引号中。
集成了单位选择框的输入/输出栏	<p>使用集成了单位选择的输入/输出栏可在不同的单位之间进行切换。如果将光标置于输入栏中，集成的单位选择框会被高亮显示（焦点无需显式设置在其上）。此外会显示一个带有转换符号的提示框，其中为该功能的相应说明信息。</p> <p>示例：</p> <pre>DEF VarEdit=(R/////////200,,100// "VarTgl") VarTgl=(S/*0="mm",1="inch"/0//WR2///302,,40)</pre> <p>或者</p> <pre>DEF VarEdit_2={TYP="R", VAL=1.234, X=200, W=100, LINK_TGL="vartgl_2"}, VARTgl_2={TYP="S", TGL="* 0=""mm"", 1=""inch""",WR=2, X=302, W=40}</pre>

变量：更改属性

以格式 **名称.属性 = 值** 重新赋值。系统会分析等号右边的表达式并为变量赋值或者分配变量属性。

名称. ac = 存取级	(ac: access level)
名称. al = 文本对齐	(al: alignment)
名称. bc = 输入/输出栏的背景色	(bc: back color)
名称 bc_gt = 图形文本的背景色	(bc: back color) (gt:graphic text)
名称. bc_st = 短文本的背景色	(bc:back color) (st:short text)
名称. bc_ut = 单位文本的背景色	(bc:back color) (ut:unit text)
名称 do = 显示选项	(do: display option)
名称. dt = 显示模式	(dt: display type)
名称 .fc = 输入/输出栏的前景色	(fc: front color)
名称. fc_gt = 图形文本的前景色	(fc:front color) (gt:graphic text)

名称. fc_st = 短文本的前景色	(fc:front color) (st:short text)
名称. fc_ut = 单位文本的前景色	(fc:front color) (ut:unit text)
名称. al = 字体大小	(fs: font size)
名称. gt = 图形文本	(gt: graphic text)
名称. hlp = 帮助图形	(hlp: help)
名称. li = 极限值	(li: limit)
名称. lt = 长文本	(lt: long text)
名称. max = 最大极限值	(max: maximum)
名称. min = 最小极限值	(min: minimum)
名称. sc = 信号色	(sc: signal color)
名称. st = 简要说明	(st: short text)
名称. tg = 转换符号	(tg: toggle)
名称. tt = 提示框	(tt: 提示框)
名称. typ = 变量类型	(typ: type)
名称. ur = 刷新速度	(ur: update rate)
名称. ut = 单位文本	(ut: unit text)
名称 val = 变量值	(val: value)
名称. var = 系统或者用户变量	(var: variable)
名称. vld = 变量状态	(vld: validation)
名称. wr = 输入模式	(wr: write)

另见

扩展配置句法 (页 38)

5.11 各个变量类型的详细说明

INTEGER 型变量

“INTEGER” 整型变量可以使用下列扩展符号显示在输入/输出栏中并保存在存储器中。

扩展数据类型中的**第 2 个字符**

表示格式	
B	二进制
D	十进制，有符号
H	十六进制
没有数据	十进制，有符号

扩展数据类型中的**第 3 个和第 4 个字符**

存储器保存	
B	字节
W	字
D	双字
BU	字节，无符号
WU	字，无符号
DU	双字，无符号

INTEGER 型数据的字符顺序

1. “I” 主标识符，表示 INTEGER（整型）
2. 表示格式
3. 存储器保存
4. “U”，无符号

有效的整型定义：	
IB	整数变量 32 位二进制表示
IBD	整数变量 32 位二进制表示
IBW	整数变量 16 位二进制表示
IBB	整数变量 8 位二进制表示
I	整数变量 32 位十进制，有符号
IDD	整数变量 32 位十进制，有符号
IDW	整数变量 16 位十进制，有符号
IDB	整数变量 8 位十进制，有符号

有效的整型定义:	
IDDU	整数变量 32 位十进制, 无符号
IDWU	整数变量 16 位十进制, 无符号
IDBU	整数变量 8 位十进制, 无符号
IH	整数变量 32 位十六进制
IHDU	整数变量 32 位十六进制
IHWU	整数变量 16 位十六进制
IHBU	整数变量 8 位十六进制

VARIANT 型变量

VARIANT 型变量通过最后赋值的数据类型确定。如果分配或输入的值是以 '-'、'+'、'!' 或数字 ('0'-'9') 开头的, 该值则被编译为数值, 其它情况下为字符串。

它可以通过功能 ISNUM 或者 ISSTR 查询。VARIANT 型主要用于向 NC 码写入变量名或者数值。

编程

可以检查变量的数据类型:

句法:	ISNUM (VAR)	
参数:	VAR	待检查其数据类型的变量名称。
	FALSE = TRUE =	查询结果可能是: 没有数值 (数据类型 = STRING) 数值 (数据类型 = REAL)

句法:	ISSTR (VAR)	
参数:	VAR	待检查其数据类型的变量名称。
	FALSE = TRUE =	查询结果可能是: 数值 (数据类型 = REAL) 没有数值 (数据类型 = STRING)
示例:	<pre>IF ISNUM(VAR1) == TRUE IF ISSTR(REG[4]+2) == TRUE</pre>	

可以改变变量的显示模式：

- 对于 INTEGER 型变量可以改变显示方式。

B	二进制
D	十进制，有符号
H	十六进制
不带正负号	
另加一个 U，表示 unsigned（无符号）	

- 对于 REAL 型变量，只能更改小数点的位数。
不允许更改基本类型，若更改，则导致文件 easyscreen_log.txt 出现错误报告。

示例：

```
Var1.typ = "IBW"
```

```
Var2.typ = "R3"
```

数字格式

数字可以以二进制、十进制、十六进制或者指数方式描述。

二进制	B01110110
十进制	123.45
十六进制	HF1A9
指数	-1.23EX-3
示例：	
	VAR1 = HF1A9
	REG[0]= B01110110
	DEF VAR7 = (R// -1.23EX-3)

说明

在通过功能 GC 生成代码时计算值仅考虑十进制或者指数描述方式，而不使用二进制和十六进制描述方式。

参见

变量参数 (页 77)

5.12 转换栏的详细说明

说明

通过转换栏扩展可以显示与 NC/PLC 变量有关的文本（在转换栏中的输入项）。只能读取使用转换栏扩展的变量。按下 INSERT 按键，打开转换栏列表。

编程

句法:	DEF VAR1=(IB/+ \$85000/15////"DB90.DBB5") 或者 DEF VAR_TGL = (S/* "Hello", "Run", "MyScreens"/"Run")	
说明:	显示对话框时，在输入/输出栏中显示文本号码 \$85015 的内容。在系统变量 DB90.DBB5 输入预设值 15。如果系统变量 DB90.DBB5 中的值改变，在每次改变时显示的文本号码重新生成 \$ (85000 + <DB90.DBB5>)。	
参数:	变量类型	系统或者用户变量中指定的变量类型
	文本号码	与语言相关的文本号码（基本），该号码作为基本号码使用
	系统或者用户变量	通过最终的文本号码（基本+补偿）形成的系统或者用户变量（补偿）。

图形与转换栏有关

转换栏用替换的图像覆盖。如果标记字节值为 1，则显示“bild1.png”；如果标记字节值为 2，则显示“bild2.png”。

```
DEF VAR1=(IDB/*1="\bild1.png", 2="\bild2.png"//,$85000/wr1//"MB[130]"/"/160,40,50,50)
```

图形的大小和位置在“输入/输出栏位置（左侧、上部、宽度、高度）”中规定。

虚拟转换键

未配置列表的转换栏，例如 DEF NoTglList=(R/*)，不带有列表。下个元素在按下转换键或运行完相应变量的相应 CHANGE() 方法后才会生成。

此时，当操作触摸屏时，一个虚拟的小键盘会显示在可变转换栏的右侧，该转换栏中只有转换键。

通过配置文件 `slguiconfig.ini` 中的以下条目也可以强制取消虚拟转换键的显示与触摸屏的关联。

```
[VirtualKeyboard]
; Forces easyscreen virtual toggle keyboard function
ForceEasyscreenVirtualToggleKey = true
```

参见

变量参数 (页 77)

5.13 预设值的详细说明

概述

根据变量栏（输入/输出栏或者转换栏）是否分配了一个预设值，一个系统或者用户变量或者两者都分配，得到不同的变量状态。只有当变量分配了一个有效值时，转换才可行。

预设值生效

条件			栏类型反应
栏类型	预设值	系统或者用户变量	
输入/输出栏	是	是	在系统或者用户变量中写预设值
	否	是	使用系统或者用户变量作为预设值
	错误	是	未计算，系统或者用户变量未描述/未使用
	是	否	预设值
	否	否	未计算
	错误	否	未计算
	是	错误	未计算
	否	错误	未计算
	错误	错误	未计算

5.14 短文本位置、输入输出栏位置的详细说明

条件			栏类型反应
栏类型	预设值	系统或者用户变量	
转换	是	是	在系统或者用户变量中写预设值
	否	是	使用系统或者用户变量作为预设值
	错误	是	未计算， 系统或者用户变量未描述/未使用
	是	否	预设值
	否	否	预设值 =转换栏的第一个单元
	错误	否	未计算
	是	错误	未计算
	否	错误	未计算
	错误	错误	未计算

参见

变量参数 (页 77)

5.14 短文本位置、输入输出栏位置的详细说明

概述

短文本和图形文本以及输入/输出栏和单位文本总是形成一个单元。即短文本的位置数据也对图形文本和输入/输出栏数据以及单位文本上的数据有效。

编程

设计的位置数据覆盖标准值，即也仅能改变一个单独的值。如果下列对话框单元没有设计位置数据，则采用上一对话框单元的数据。

如果对话框单元没有规定位置，则使用预设置。短文本和输入/输出栏的栏宽度在标准情况下各行由栏数和最大栏宽度确定，即：栏宽度=最大行宽/列数。

图像和单位文本宽度是固定的，根据编程支持的请求优化。如果已设计图像和单位文本宽度，则短文本或者输入/输出栏的宽度相应缩短。

短文本和输入/输出栏的顺序可以通过位置数据互换。

参见

变量参数 (页 77)

5.15 使用字符串

字符串链

设计时也可以使用字符串，以动态配置文本显示或者合并代码生成的不同文本。

规则

使用字符串变量时注意以下规定：

- 链接由左向右处理。
- 层叠的表达式由内向外运算。
- 忽略大小写。
- 字符串变量通常左对齐显示。

字符串可以通过一个简单的空字符串指令删除。

字符串在等号右边以运算符 "<<" 开始。字符串中双引号 (") 通过两个连续的双引号标记。字符串可以在 IF 指令中检查相等性。

示例

下列示例预设：

```
VAR1.VAL = "这是一个"
```

```
VAR8.VAL = 4
```

```
VAR14.VAL = 15
```

```
VAR2.VAL = "错误"
```

```
$85001 = "这是一个"
```

```
$85002 = "报警文本"
```

编辑字符串:

- 合并字符串:
VAR12.VAL = VAR1 << "错误" ;结果: "这是一个错误"
- 删除一个变量:
VAR10.VAL = "" ;结果: 空字符串
- 设定一个带有文本变量的变量:
VAR11.VAL = VAR1.VAL ;结果: "这是一个"
- 数据类型匹配:
VAR13.VAL = "这是 " << (VAR14 - VAR8) << ". 错误"
;结果: "这是第 11 个错误"
- 处理数字值:
VAR13.VAL = "错误" << VAR14.VAL << ": " << \$85001 << \$85002
;结果: "错误 15: "这是一个报警文本"
IF VAR15 == "错误" ; IF 指令中的字符串
VAR16 = 18.1234
;结果: VAR16 等于 18.1234,
; 当 VAR15 等于 "错误" 时
ENDIF
- 字符串中的双引号:
VAR2="你好, 这是一个 "测试" "
;结果: 你好, 这是一个 "测试"
- 和变量内容有关的系统或者用户变量的字符串:
VAR2.Var = "\$R[" << VAR8 << "]" ; 结果: \$R[4]

另见

字符串函数 (页 169)

5.16 变量 CURPOS

说明

通过变量 CURPOS 可以在当前对话框的激活的输入栏中调出或者操纵光标位置。变量显示光标前有多少个字符。如果光标位于输入栏开始处, 则 CURPOS 接受值为 0。如果更改 CURPOS 值, 则光标停留在输入栏中相应的位置上。

为了可以在变量值更改情况下反应，可以借助于一个 **CHANGE** 方法监控改变情况。如果 **CURPOS** 值改变，则跳转 **CHANGE** 方法并执行包含的指令。

5.17 变量 CURVER

说明

属性 **CURVER**（当前版本）允许匹配编程用于处理不同的版本。变量 **CURVER** 仅可读。

说明

在代码生成时自动以最新的版本生成，即使之前已用老的版本反编译。命令 **"GC"** 总是生成最新的版本。在生成代码中，在使用注释当版本 **> 0** 时插入一个生成的代码的附加标记。

规则

总是显示带有所有变量的最新的对话框。

- 以前的变量不允许改变。
- 新的变量以任意顺序插入在以前（循环）的编程中。
- 不允许从对话框由一个版本到下一个版本去除变量。
- 对话框必须包含所有版本的变量。

举例

```
(IF CURVER==1 ...)
```

 ; CURVER 在反编译时自动通过反编译代码的版本设置。

5.18 变量 ENTRY

说明

通过变量 **ENTRY** 可以检查如何调用对话框。

5.19 变量 ERR

编程

句法:	ENTRY	
说明:	变量 ENTRY 仅可读。	
返回值:		查询结果可能是:
	0 =	没有编程支持
	1 =	通过软键启动屏幕; 还未生成任何代码 (预设值, 同配置)
	2 =	通过软键启动屏幕; 已生成代码 (预设值来自最后由该屏幕生成的代码)
	3 =	重新编译, 带使用注释 (#行)
	4 =	Code_typ = 0: 带使用注释的 NC 代码 (#行)
	5 =	Code_typ = 1: 无使用注释的 NC 代码 (#行)

示例

```

IF ENTRY == 0
    DLGL ( "在编程下不调用对话框" )
ELSE
    DLGL ( "在编程下调用对话框" )
ENDIF
    
```

5.19 变量 ERR

说明

通过变量 ERR 可以检查是否已正确执行先前行。

编程

句法:	ERR
说明:	变量 ERR 仅可读。

返回值:		查询结果可能是:
	FALSE =	已正确执行先前行
	TRUE =	未正确执行先前行

示例

```

VAR4 = 螺纹[VAR1,"KDM",3]           ; 作为数组给出值
IF ERR == TRUE                       ; 查询是否在数组中找到值
VAR5 = "数组存取出错"                ; 如果未在数组中找到值, 则赋值“数组存取出错”给变量。

ELSE
VAR5 = "一切正常"                    ; 如果在数组中找到值, 则赋值“一切正常”给变量。
ENDIF

```

5.20 变量 FILE_ERR

说明

用变量 FILE_ERR 可以检查前面的 GC 或者 CP 指令是否正确执行。

编程

句法:	FILE_ERR
说明:	变量 FILE_ERR 仅可读。

5.20 变量 FILE_ERR

返回值:		可能的结果是:
	0 =	排列中的操作
	1 =	驱动器/路径不存在
	2 =	路径/文件存取故障
	3 =	驱动器未就绪
	4 =	错误的文件名
	5 =	文件已经打开
	6 =	存取失败
	7 =	目标路径不存在或不允许
	8 =	复制源符合目标
	10 =	内部错误: FILE_ERR = 10 时, 与一个未分类到其他类别故障有关。

示例

```

CP("D:\source.mpf", "E:\target.mpf")
; 从 source.mpf 向 E:\target.mpf 复制

IF FILE_ERR > 0
; 询问是否出现故障
IF FILE_ERR == 1
; 询问特定的错误号并输出所属的故障文本
VAR5 = "驱动器/路径不存在"
ELSE
IF FILE_ERR == 2
VAR5 = "路径-/文件存取故障"
ELSE
IF FILE_ERR == 3
VAR5 = "错误文件名"
ENDIF
ENDIF
ENDIF
ELSE
VAR5 = "一切正常"
; 如果 CP (或者 GC) 中没有出现错误
; 则输出 "一切正常"
ENDIF

```


5.21 变量 FOC

说明

使用变量 **FOC** 可以控制对话框内的输入中心（当前有效的输入/输出栏）。已预先固定定义光标左右键、前后键以及 **PGUP**、**PGDN** 的反应。

说明

FOC 不允许通过一个导航事件触发。光标位置只允许在软键 **PRESS** 方法、**CHANGE** 方法、...中改变。

带有输入模式 **wr = 0** 和 **wr = 4** 的变量以及辅助变量无法实现聚焦。

编程

句法:	FOC	
说明:	可以读取和写入该变量。	
返回值:	读取	聚焦变量的名称作为结果输出。
	写入	可以写入一个字符串或者数值。字符串视为变量名称，数值视为变量索引。

示例

```

IF FOC == "Var1"                ; 读取输入中心
    REG[1] = Var1
ELSE
    REG[1] = Var2
ENDIF

FOC = "Var1"                    ; 分配变量 1 给输入中心。
FOC = 3                         ; 分配第 3 对话框单元， WR ≥ 2 给输入中心。

```

参见

FOCUS (页 112)

5.22 变量 S_ALEVEL

说明

在配置中会通过屏幕属性 S_ALEVEL 询问当前访问级。

编程

句法:	S_ALEVEL
说明:	查询当前访问级
返回值:	0: 系统 1: 制造商 2: 维修 3: 用户 4: 钥匙开关位置 3 5: 钥匙开关位置 2 6: 钥匙开关位置 1 7: 钥匙开关位置 0

示例

```
REG[0] = S_ALEVEL
```

参见

ACCESSLEVEL (页 109)

5.23 变量 S_CHAN

说明

通过变量 S_CHAN 可以确定用于显示或者某个评估的当前通道号码。

编程

句法: **S_CHAN**
说明: 查询当前通道号
返回值: 通道号

示例

```
REG[0] = S_CHAN
```

参见

CHANNEL (页 111)

5.24 变量 S_CONTROL

说明

在配置中会通过屏幕属性 S_CONTROL 询问当前控制系统名称。

编程

句法: **S_CONTROL**
说明: 查询当前控制系统名称
返回值: 控制系统名称为 mmc.ini 中的段落名。

示例

```
REG[0] = S_CONTROL
```

参见

CONTROL (页 111)

5.25 变量 S_LANG

说明

在配置中会通过屏幕属性 S_LANG 询问当前语言。

编程

句法:	S_LANG
说明:	查询当前语言
返回值:	resources.xml 中的语言缩写, 例如: “deu”、“eng”等

示例

```
REG[0] = S_LANG
```

参见

LANGUAGE (页 113)

5.26 变量 S_NCCODEREADONLY

说明

只有在编辑器中通过对话框“Run MyScreens”反编译循环时, 屏幕属性 S_NCCODEREADONLY 才有意义。通过 S_NCCODEREADONLY 确定是否可以修改编辑器中的反编译 NC 代码。

返回值适用条件:

- **TRUE:** 可以修改编辑器中的反编译 NC 代码
- **FALSE:** 由于编辑器中的反编译 NC 代码已经处于预处理状态 (= TRUE), 因此, 无法进行修改 (只读)。

5.27 变量 S_RESX 和 S_RESY

说明

在配置中会通过屏幕属性 S_RESX 和 S_RESY 询问当前分辨率或其 X 或 Y 分量。

示例

```
REG[0] = S_RESY
```

参见

RESOLUTION (页 118)

编程指令

6.1 运算符

概述

在编程时可以使用以下几种运算符：

- 数学运算符
- 比较运算符
- 逻辑（布尔）运算符
- 位运算符
- 三角函数功能

6.1.1 数学运算符

概述

数学运算符	名称
+	加法
-	减法
*	乘法
/	除法
MOD	模数运算
()	括号
AND	和运算符
OR	或运算符
NOT	非运算符
ROUND	带有小数位的数四舍五入

举例: $\text{VAR1}.\text{VAL} = 45 * (4 + 3)$

ROUND

在执行对话框设计时，用 **ROUND** 指令把数值四舍五入直至小数点后面的 12 位。小数点后位置在变量栏不能显示。

使用

ROUND 通过用户的两个参数来控制：

```
VAR1 = 5,2328543
```

```
VAR2 = ROUND( VAR1, 4 )
```

结果： VAR2 = 5,2339

VAR1 包含在四舍五入的数中。参数"4"给出了结果中的小数点后的位数，保存在 VAR2 中。

三角函数功能

三角函数功能	名称
SIN(x)	正弦 x
COS(x)	余弦 x
TAN(x)	正切 x
ATAN(x, y)	反正切 x/y
SQRT(x)	平方根 x
ABS(x)	绝对值 x
SDEG(x)	换算为度数
SRAD(x)	换算为弧度
CALC_ASIN(x)	反正弦 x
CALC_ACOS(x)	反余弦 x

说明

这些函数处理弧度。为了换算，可以使用函数 **SDEG()** 和 **SRAD()**。

举例： VAR1.VAL = SQRT(2)

数学函数

数学函数	名称
CALC_CEIL(x)	计算下一个比 x 大的整数（向上取整）
CALC_FLOOR(x)	计算下一个比 x 小的整数（向下取整）
CALC_LOG(x)	计算以 e 为底的 x 的（自然）对数
CALC_LOG10(x)	计算以 10 为底的 x 的对数
CALC_POW(x, y)	计算 x 的 y 次方（ x 的 y 次乘方）
CALC_MIN(x, y)	计算 x 与 y 哪个小
CALC_MAX(x, y)	计算 x 与 y 哪个大

随机功能：随机数

句法:	RANDOM (下限值, 上限值)	
说明:	RANDOM 功能为提供预先给定范围内的伪随机数。	
参数:	下限值	下限值 ≥ -32767 , 下限值 $<$ 上限值
	上限值	上限值 ≤ 32767

示例

```
REG[0] = RANDOM(-10,10) ; 可能结果 = -3
```

常量

常量	
<i>PI</i>	3.14159265358979323846
<i>FALSE</i>	0
<i>TRUE</i>	1

举例: VAR1.VAL = PI

6.1 运算符

比较运算符

比较运算符	
==	相等
<>	不等
>	大于
<	小于
>=	大于等于
<=	小于等于

示例:

```
IF VAR1.VAL == 1  
    VAR2.VAL = TRUE  
ENDIF
```

条件

层叠深度没有限制。

带有一个命令的条件:

```
IF  
...  
ENDIF
```

带有两个命令的条件:

```
IF  
...  
ELSE  
...  
ENDIF
```

6.1.2 位运算符

概述

位运算符	名称
BOR	位方式 OR
BXOR	位方式 XOR
BAND	位方式 AND
BNOT	位方式 NOT
SHL	位向左移动
SHR	位向右移动

运算符 SHL

通过运算符 SHL(SHIFT LEFT)向左移动位。此时要移动的值或者移动步数可以直接规定或者作为变量规定。如果达到数据格式极限，则已超出位，没有出错信息。

使用

句法:	变量 = 值 SHL 步数	
说明:	向左移动	
参数:	值	要移动的值
	步数	移动步数

示例

```

PRESS (VS1)
VAR01 = 16 SHL 2           ; 结果 = 64
VAR02 = VAR02 SHL VAR04   ; VAR02 内容转换为 32 位格式，无符号
                           ; 并且位向左移动 VAR04 内容。接着
                           ; 32 位值重新转换回
                           ; 变量 VAR02 的格式。
END_PRESS

```

6.2 方法

运算符 SHR

通过运算符 SHR (SHIFT RIGHT)向右移动位。此时要移动的值或者移动步数可以直接规定或者作为变量规定。如果达到数据格式极限，则已超出位，没有出错信息。

使用

句法:	变量 = 值 SHR 步数	
说明:	向右移动	
参数:	值	要移动的值
	步数	移动步数

示例

```

PRESS (VS1)
VAR01 = 16 SHR 2           ; 结果 = 4
VAR02 = VAR02 SHR VAR04   ; VAR02 内容转换为 32 位格式，无符号
                           ; 并且位向右移动 VAR04 内容。接着
                           ; 32 位值重新转换回
                           ; 变量 VAR02 的格式。
END_PRESS
    
```

6.2 方法

概述

在对话框和与对话框相关的软键栏中（软键栏由新设计的对话框调用），可以通过不同的事件（退出输入栏，按下软键）触发某些特定的动作。这些动作设计在方法中。

方法的基本编程按如下方式进行：

说明块	注释	参考章节
PRESS (HS1)	； 方法的开始标识	
LM...	； 功能	参见章节“功能”
LS...		

说明块	注释	参考章节
Var1.st = ...	; 改变属性	参见章节“软键栏” 和章节“对话框单元”
Var2 = Var3 + Var4 ... EXIT	; 通过变量计算	参见章节“定义变量”
END_PRESS	; 方法的结束标识	

6.2.1 ACCESSLEVEL

说明

当在屏幕打开时更改当前访问等级时，就会运行 ACCESSLEVEL 方法。

编程

句法:	ACCESSLEVEL <指令> END_ACCESSLEVEL
说明:	访问等级
参数:	- 无 -

参见

变量 S_ALEVEL (页 98)

6.2.2 CHANGE

说明

当变量值已改变时运行 CHANGE（改变）方法。即在 CHANGE（改变）方法中设计变量改变时立即运行的变量计算。

6.2 方法

单元特定的 **CHANGE** 方法和全局的 **CHANGE** 方法有所不同:

- 当特定变量值已改变时运行**单元特定的 CHANGE 方法**。如果系统或者用户变量已分配一个变量, 则可以在 **CHANGE** 方法中循环更新变量值。
- 当改变任意一个变量值且没有设计单元特定的 **CHANGE** 方法时, 运行**全局 CHANGE 方法**。

编程 “单元特定”

句法:	CHANGE(<i>名称</i>) ... END_CHANGE	
说明:	修改指定变量的值	
参数:	名称	变量名称

示例

```

DEF VAR1=(I////////"DB20.DBB1")           ; Var1 分配一个系统变量
CHANGE (VAR1)
IF VAR1.Val <> 1
    VAR1.st="工具正确!"                   ; 如果系统变量的值 ≠ 1, 则变量的简要说明为: 工具正确!
    otto=1
ELSE
    VAR1.st="注意 错误!"                   ; 如果系统变量的值 = 1, 则变量的简要说明为: 注意 错误!
    otto=2
ENDIF
VAR2.Var=2
END_CHANGE
    
```

编程 “全局特定”

句法:	CHANGE() ... END_CHANGE
说明:	改变任意变量值
参数:	- 无 -

示例

```
CHANGE ()
EXIT           ; 如果任何一个变量值改变，则退出对话框。
END_CHANGE
```

参见

对话框打开方式（属性 CB）（页 225）

6.2.3 CHANNEL

说明

当在屏幕打开时更改当前通道时，即通道切换时，就会运行 CHANNEL 方法。

编程

句法:	CHANNEL <指令> END_CHANNEL
说明:	通道切换
参数:	- 无 -

参见

变量 S_CHAN (页 98)

6.2.4 CONTROL

说明

当在屏幕打开时更改当前控制系统时，即通常在 1:n 切换时，就会运行 CONTROL 方法。

6.2 方法

编程

句法:	CONTROL <指令> END_CONTROL
说明:	系统切换
参数:	- 无 -

参见

变量 S_CONTROL (页 99)

6.2.5 FOCUS

说明

当对话框中聚焦（光标）定位在另一个栏上时，运行 **FOCUS** 方法。

方法 **FOCUS** 不允许通过一个导航事件触发。光标位置只允许在软键 **PRESS** 方法、**CHANGE** 方法、...中改变。光标移动的反应预先固定定义。

说明

在 **FOCUS** 方法中不允许定位在另一个变量上并且也不允许装载新的对话框。

编程

句法:	FOCUS ... END_FOCUS
说明:	光标: 定位
参数:	- 无 -

示例

```
FOCUS
DLGL ("聚焦已设定在变量"<< FOC <<"上。")
END_FOCUS
```

参见

变量 FOC (页 97)

6.2.6 LANGUAGE

说明

当在屏幕打开时更改当前语言时，就会运行 LANGUAGE 方法。

编程

句法:	LANGUAGE <指令> END_LANGUAGE
说明:	语言
参数:	- 无 -

参见

变量 S_LANG (页 100)

6.2.7 LOAD

说明

在已编译变量定义和软键定义 (DEF Var1= ..., HS1= ...) 后运行 LOAD 方法。此时，对话框还未显示。

6.2 方法

编程

句法:	LOAD ... END_LOAD
说明:	装载
参数:	- 无 -

示例

```

LOAD                ; 开始标记
  屏幕窗口 1.Hd = $85111 ; 对话框标题来自文本文件
  VAR1.Min = 0        ; 指定变量的最小极限值
  VAR1.Max = 1000    ; 指定变量的最大极限值
END_LOAD           ; 结束标记
    
```

参见

线、分割线、矩形、圆形和椭圆形 (页 181)

6.2.8 UNLOAD

说明

在卸载对话框之前，运行 UNLOAD 方法。

编程

句法:	UNLOAD ... END_UNLOAD
说明:	卸载
参数:	- 无 -

示例

```

UNLOAD
  REG[1] = VAR1          ; 保存寄存器的变量
END_UNLOAD

```

6.2.9 OUTPUT

说明

当调用功能 "GC" 时，运行 OUTPUT 方法。在 OUTPUT 方法中变量和辅助变量作为 NC 代码设计。代码行各个单元的链接用一个空格符实现。

说明

NC 代码可以用文件功能在一个额外的文件中生成并移向 NC。

编程

句法:	OUTPUT(名称) ... END_OUTPUT	
说明:	输出 NC 程序中的变量	
参数:	名称	OUTPUT 方法的名称

程序段号码和隐藏标记

当要继续保留编程支持激活时在零件程序中直接设置的行号和反编译时的隐藏标记时，OUTPUT 方法不允许包含行号和隐藏标记。

在零件程序中通过编辑器更改影响下列行为：

条件	行为
程序段数目保持不变。	程序段号码继续保留。
程序段数目变小。	删除最大的程序段号码。
程序段数目变大。	新的程序段没有程序段号码。

6.2 方法

示例

```

OUTPUT (CODE1)
  "CYCLE82(" Var1.val "," Var2.val "," Var3.val ","Var4.val "," Var5.val ","
Var6.val ") "
END_OUTPUT
    
```

6.2.10 PRESS

说明

当已按下相应的软键时，运行 PRESS 方法。

编程

句法:	PRESS(软键) ... END_PRESS		
名称:	按下软键		
参数:	软键	软键名称: HS1 - HS8 和 VS1 - VS8	
	回调	按键<RECALL> (回调)	
	ENTER	按键 <ENTER>, 参见 PRESS(ENTER) (页 117)	
	转换	按键 <TOGGLE>, 参见 PRESS(TOGGLE) (页 118)	
	PU	Page Up	向前翻页
	PD	Page Down	向后翻页
	SL	Scroll Left	光标向左
	SR	Scroll Right	光标向右
	SU	Scroll Up	光标向上
	SD	Scroll Down	光标向下

示例

```

HS1 = ("其他软键栏")
    
```

```

HS2 = ("没有功能")
PRESS (HS1)
  LS ("软键栏 1")           ; 载入其他软键栏
  Var2 = Var3 + Var1
END_PRESS
PRESS (HS2)
END_PRESS
PRESS (PU)
  INDEX = INDEX -7
CALL ("UP1")
END_PRESS

```

6.2.11 PRESS(ENTER)

说明

对于带有输入/输出栏且输入模式为 WR3 或 WR5 的变量，当按下回车(Enter)键时，就会调用 PRESS(ENTER)方法：

- WR3: 定位到栏的位置并按下回车键
- WR5: 在输入模式下，使用回车键接收值

编程

句法:	PRESS(ENTER) <指令> END_PRESS
说明:	回车键已按下
参数:	- 无 -

6.2.12 PRESS(TOGGLE)

说明

与当前处于焦点下的变量无关，只要按下转换键就会调用 PRESS(TOGGLE)方法。

需要时，可借助屏幕属性 FOC 来检测哪个变量当前处于焦点下。

编程

句法:	PRESS(TOGGLE) <指令> END_PRESS
说明:	转换键已按下
参数:	- 无 -

示例

```
PRESS(TOGGLE)
  DLGL("Toggle key pressed at variable " << FOC)      ; 借助屏幕属性 FOC 检测哪个变量当前处于焦点下
END_PRESS
```

6.2.13 RESOLUTION

说明

当在屏幕打开时更改当前分辨率时，即通常在 TCU 切换时，就会运行 RESOLUTION 方法。

编程

句法:	RESOLUTION <指令> END_RESOLUTION
说明:	分辨率
参数:	- 无 -

参见

变量 S_RESX 和 S_RESY (页 101)

6.2.14 RESUME

说明

当屏幕（例如）被区域切换中断并要重新显示时，就会调用 **RESUME** 方法。此时会重新处理数值变化，必要时启动计时器并执行 **RESUME** 块。

编程

句法:	RESUME <指令> END_RESUME
说明:	屏幕重新激活
参数:	- 无 -

6.2.15 SUSPEND

说明

当中断屏幕且不卸载时，就会调用 **SUSPEND** 方法。例如，当因为简单的区域切换而离开了屏幕，但不显式卸载该屏幕时。屏幕仍保留在后台，但此时不处理数值变化和可能存在的计时器。屏幕暂时中止。

编程

句法:	SUSPEND <指令> END_SUSPEND
说明:	屏幕中断
参数:	- 无 -

示例

	SUSPEND
--	---------

6.2 方法

```
MyVar1 = MyVar1 + 1
END_SUSPEND
```

6.2.16 示例：带 OUTPUT 方法的版本管理

概述

已有的对话框可以通过扩展补充附加的变量。附加的变量在定义中变量名后的圆括号内有一个版本识别号：（0 = 原始，未写入），1 = 版本 1，2 = 版本 2，...

举例：

```
DEF var100=(R//1) ; 原始，相当于版本 0
DEF var101(1)=(S//"Hallo") ; 补充，从版本 1 起
```

在写入 OUTPUT 方法时可以参考某个版本状态，与定义的总体性有关。

举例：

```
OUTPUT (NC1) ; 在 OUTPUT 方法中仅提供原始变量。
OUTPUT (NC1,1) ; 在 OUTPUT 方法中提供原始变量和带有版本标识符 1 的补充变量。
```

原始版的 OUTPUT 方法不需要版本标识符，然而也可以记为 0。OUTPUT(NC1) 相当于 OUTPUT(NC1,0)。OUTPUT 方法中的版本标识符 n 包括所有变量，从原始 0、1、2、... 直至 n。

编程版本标识

```
//M(XXX) ; 版本 0 (默认)
DEF var100=(R//1)
DEF var101=(S//"Hallo")
DEF TMP
VS8=("GC")
PRESS (VS8)
GC ("NC1")
END_PRESS

OUTPUT (NC1)
```



```

var100",,"var101
END_OUTPUT

; ***** 版本 1, 补充的定义 *****
//M(XXX)
DEF var100=(R//1)
DEF var101=(S//"Hallo")
DEF var102(1)=(V//"HUGO")
DEF TMP
VS8=("GC")
PRESS(VS8)
GC("NC1")
END_PRESS
...

OUTPUT(NC1) ; 原始和其它新的版本
var100",,"var101
END_OUTPUT
...

OUTPUT(NC1,1) 版本 1
var100",,"var101", " var102
END_OUTPUT

```

6.3 功能

概述

在对话框和与对话框相关的软键栏中提供不同的功能，这些功能通过事件（例如：退出输入栏，按下软键）触发并在方法中设计。

子程序

重复的或者其它的设计指令，这些指令总结为一个特定的过程，可以在子程序中设计。子程序可以随时装载到主程序或其它子程序中并随时进行编辑，即指令不必多次重复设计。作为主程序适用于对话框描述块或者软键栏。

6.3 功能

外部功能

通过外部功能可以引入其它一些针对用户的功能。外部功能存放在一个 DLL 文件中并通过设计文件定义行中的条目识别。

PI 服务

通过功能 PI_START 可以在由 PLC 在 NC 区中启动 PI 服务（程序实例服务）。

另见

功能 (FCT) (页 143)

PI 服务 (页 158)

6.3.1 读写驱动参数：RDOP, WDOP, MRDOP

说明

使用功能 RDOP、WDOP 和 MRDOP 可以读写驱动参数。

说明

读取驱动参数时不要短于 1 秒的周期，时间长些更佳。

原因：否则与驱动的通讯可能会受到很大干扰，甚至导致故障。

说明

如果在读写驱动参数时出错，则会对屏幕属性 ERR 进行相应的设置。

编程

句法:	RDOP (“驱动对象的名称” , “参数号”)
说明:	读驱动参数(Drive Object Parameter)

参数:	驱动对象的名称	驱动对象的名称可参考“诊断”操作区 (> ETC > HSK 8: 驱动系统) “驱动系统诊断”中的“DO Name” 一列 (见下图)。
	参数号	

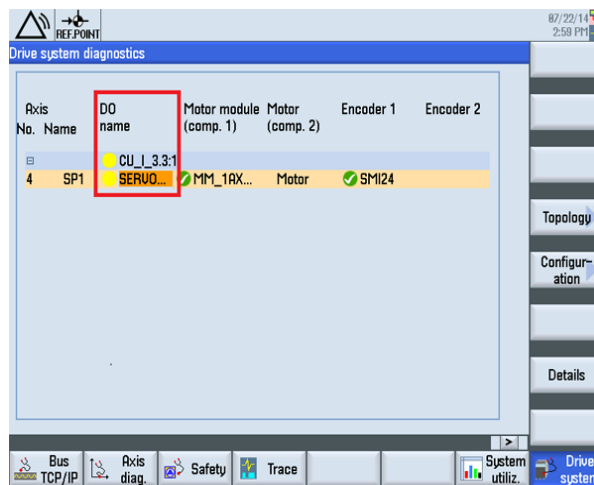


图 6-1 驱动对象的名称

句法:	MRDOP (“驱动对象的名称” , “参数号 1”**“参数号” [*...] , 寄存器索引)
说明:	多次读取驱动参数。 使用命令 MRDOP 可以通过一次存取向寄存器中传输一个驱动对象的多个驱动参数。这种存取比通过单个存取读取速度明显加快。

6.3 功能

参数:	驱动对象的名称	驱动对象的名称可从（例如）“调试”操作区域的基本画面中获取
	参数号 1 ... n	变量名称用“*”作为分隔符。按照命令中变量名称的顺序将值接收到寄存器 REG[寄存器索引]中。
	寄存器索引	第一个变量的值在 REG[寄存器索引] 中。 第二个变量的值在 REG[寄存器索引 + 1] 中。

句法:	WDOP (“驱动对象的名称”, “参数号”, “值”)	
说明:	写驱动参数(Drive Object Parameter)	
参数:	驱动对象的名称	驱动对象的名称可从（例如）“调试”操作区域的基本画面中获取
	参数号	参数号
	值	待写入的值

示例

读取驱动对象“SERVO_3.3:2”的电机温度 r0035:

```
MyVar=RDOP ("SERVO_3.3:2", "35") ;
```

读取驱动对象“SERVO_3.3:2”的电机温度 r0035 和转矩实际值 r0080 并从寄存器索引 10 开始保存各个结果:

```
MRDOP ("SERVO_3.3:2", "35*80", 10)
```

6.3.2 子程序调用(CALL)

说明

通过 **CALL** 功能可以从方法的任意一个位置调用一个装载的子程序。允许叠加，即由一个子程序调用另一个子程序。

编程

句法:	CALL("名称")	
说明:	调用子程序	
参数:	名称	子程序名称

示例

```
//M(屏幕窗口 1)
DEF VAR1 = ...
DEF VAR2 = ...
CHANGE (VAR1)
...
CALL ("MY_UP1")           ; 调用并编辑子程序
...
END_CHANGE

CHANGE (VAR2)
...
CALL ("MY_UP1")           ; 调用并编辑子程序
...
END_CHANGE

SUB (MY_UP1)
                           ;do something

END_SUB
//END
```

6.3 功能

6.3.3 定义块(//B)

说明

子程序在程序文件中用块标记 **//B** 标记并通过 **//END** 结束。每个块标记可以定义多个子程序。

说明

必须在调用子程序的对话框中定义子程序所使用的变量。

编程

一个块有下列结构：

句法：	//B(块名称) SUB(名称) END_SUB [SUB(名称) ... END_SUB] ... //END	
说明：	定义子程序	
参数：	块名称	块标记名称
	名称	子程序名称

示例

```

//B(PROG1)                ; 块开始
SUB(UP1)                  ; 子程序开始
...
REG[0] = 5                ; 寄存器 0 赋值 5
...
END_SUB                   ; 子程序结束
SUB(UP2)                  ; 子程序开始
    IF VAR1.val=="Otto"
    
```

```

    VAR1.val="Hans"
RETURN
ENDIF
    VAR1.val="Otto"
END_SUB           ; 子程序结束
//END            ; 块结束

```

6.3.4 检查变量 (CVAR)

说明

借助于功能 **CVAR** (检查变量) 可以查询对话框的所有变量或者仅特定变量或者辅助变量是否正确。

在用功能 **GC** 产生一个 **NC** 代码之前, 查询变量是否包含一个有效值是必要的。

如果变量状态命名符 **.vld = 1**, 则变量正确,

编程

句法:	CVAR(VarM)	
说明:	检查变量的有效内容	
参数:	VarN	要检查的变量列表。 可以最多有 29 个变量, 各自之间通过逗号隔开。最大字符长度为 500 。 查询结果可能是:
	1 =	TRUE (真) (所有变量都有有效内容)
	0 =	FALSE (假) (至少一个变量没有有效内容)

示例

```

IF CVAR == TRUE           ; 检查所有变量
    VS8.SE = 1           ; 如果所有变量正确, 则软键 VS8 可见
ELSE
    VS8.SE = 2           ; 如果变量包含错误值, 则软键 VS8 不可操作
ENDIF

```

6.3 功能

```

IF CVAR("VAR1", "VAR2") == TRUE
    ; 检查变量 VAR1 和 VAR2
    DLGL ("VAR1 和 VAR2 正确")
    ; 如果 VAR1 和 VAR2 无错误, 则显示对话框行"VAR1
    和 VAR2 正确"
ELSE
    DLGL ("VAR1 和 VAR2 不正确")
    ; 如果 VAR1 和 VAR2 错误, 则显示对话框行"VAR1
    和 VAR2 不正确"
ENDIF
    
```

6.3.5 CLEAR_BACKGROUND

说明

使用功能 CLEAR_BACKGROUND 可以删除图形元素 LINE、RECT、ELLIPSE、V_SEPARATOR 和 H_SEPARATOR。

参见

线、分割线、矩形、圆形和椭圆形 (页 181)

6.3.6 文件功能“Copy Program (CP, 复制程序)”

说明

功能 CP (复制程序) 用于在 HMI 文件系统或者 NC 文件系统中复制文件。

编程

句法:	CP("源文件", "目标文件")	
说明:	文件: 复制	
参数:	源文件	源文件完整的路径数据
	目标文件	目标文件完整的路径数据

可通过返回值 (VAR1 被定义为辅助变量) 查询功能是否成功执行:

```
CP ("//NC/MPF.DIR/HOHO.MPF", "//NC/MPF.DIR/ASLAN.MPF", VAR1)
```


示例

带返回值的应用情况:

```
CP("//NC/MPF.DIR/HOHO.MPF", "//NC/MPF.DIR/ASLAN.MPF", VAR1)
CP("CF_CARD:/wks.dir/MESS_BILD.WPD/MESS_BILD.MPF", "//NC/WKS.DIR/AAA.WPD/HOHO2.MPF", VAR1)
CP("//NC/MPF.DIR/HOHO.MPF", "CF_CARD:/wks.dir/WST1.WPD/MESS.MPF", VAR1) ; WPD 必须存在
```

无返回值的应用情况:

```
CP("//NC/MPF.DIR/HOHO.MPF", "//NC/MPF.DIR/ASLAN.MPF")
CP("CF_CARD:/mpf.dir/MYPROG.MPF", "//NC/MPF.DIR/HOHO.MPF")
CP("//NC/MPF.DIR/HOHO.MPF", "CF_CARD:/XYZ/MYPROG.MPF") ; XYZ 必须存在
```

另见

FILE_ERR 的支持: 变量 FILE_ERR (页 95)

6.3.7 文件功能“Delete Program (DP, 删除程序)”

说明

功能 DP (删除程序) 删除一个被动的 HMI 文件系统或者主动的 NC 文件系统的文件。

编程

句法:	DP("文件")	
说明:	文件: 删除	
参数:	文件	要删除文件的完整路径数据

示例

此功能使用以下句法进行数据管理:

- 带返回值

```
DP("//NC/MPF.DIR/MYPROG.MPF", VAR1)
DP("//NC/WKS.DIR/TEST.WPD/MYPROG.MPF", VAR1)
```

6.3 功能

DP ("//NC/CMA.DIR/MYPROG.SPF", VAR1)

VAR1 = 0 文件已删除。

VAR1 = 1 文件尚未删除。

- 无返回值:

DP ("//NC/MPF.DIR/MYPROG.MPF")

DP ("//NC/WKS.DIR/TEST.WPD/MYPROG.MPF")

DP ("//NC/CMA.DIR/MYPROG.SPF")

6.3.8 文件功能“Exist Program (EP, 退出程序)”

说明

功能 EP（存在程序）检查 NC 文件系统的特定 NC 程序或者 HMI 文件系统中在规定的路径下面是否存在某个文件。

编程

句法:	EP("文件")	
说明:	检查 NC 程序的存在	
参数:	文件	NC 文件系统或者 HMI 文件系统中文件的完整路径数据
返回值:	待分配查询结果的变量的名称。	

功能 EP 使用新的句法和旧的逻辑（调整句法）。

使用符合要求的名称直接响应文件:

//NC/MPF.DIR/XYZ.MPF

或

CF_CARD: /MPF.DIR/XYZ.MPF (指示 /user/sinumerik/data/prog)

或

LOC: (即 CF_CARD)

新句法示例:

```

EP ("//NC/WKS.DIR/TEST.WPD/XYZ.MPF", VAR1)
EP ("CF_CARD:/mpf.dir/XYZ.MPF", VAR1)
EP ("LOC:/mpf.dir/XYZ.MPF", VAR1)
; 带返回值:
; VAR1 = 0 文件存在。
; VAR1 = 1 文件不存在。

```

旧句法示例:

```

EP ("/MPF.DIR/CFI.MPF", VAR1)
; 带返回值:
; VAR1 = M 文件位于 HMI 文件系统中。
; VAR1 = N 文件位于 NC 文件系统中。
; VAR1 = B 文件位于 HMI 和 NC 文件系统中。

```

示例

```

EP ("/MPF.DIR/GROB.MPF", VAR1) ; 旧句法 - 路径现在采用 /, 而不是 \

IF VAR1 == "M"
  DLGL ("文件位于 HMI 文件系统中")
ELSE
  IF VAR1 == "N"
    DLGL ("文件位于 NC 文件目录中")
  ELSE
    DLGL ("文件既不在 HMI 文件系统中, 也不在 NC 文件系统中")
  ENDIF
ENDIF
ENDIF

```

6.3.9 文件功能“Move Program (MP, 移动程序)”**说明**

功能 MP (移动程序) 用于在 HMI 文件系统或者 NC 文件系统中复制文件。

6.3 功能

编程

句法:	MP("源", "目标")	
	MP ("CF_CARD:/MPF.DIR/MYPROG.MPF", "//NC/MPF.DIR")	
说明:	移动文件	
参数:	源文件	完整路径
	目标文件	完整路径
返回值	查询结果	

示例

带返回值:

```

MP ("//NC/MPF.DIR/123.MPF", "//NC/MPF.DIR/ASLAN.MPF", VAR1)           ; 位于 NC 内部
MP ("//NC/MPF.DIR/123.MPF", VAR0, VAR1)                               ; 通过变量设定目标
MP (VAR4, VAR0, VAR1)                                                 ; 通过变量设定源和目标
MP ("CF_CARD:/mpf.dir/myprog.mpf", "//NC/MPF.DIR/123.MPF", VAR1)     ; 从 CF 卡至 NC
MP ("//NC/MPF.DIR/HOHO.MPF", "CF_CARD:/XYZ/123.mpf", VAR1)         ; 从 NC 至 CF 卡

; 带返回值:
; 执行 VAR1 = 0
; 未执行 VAR1 = 1
    
```

6.3.10 文件功能“Select Program (SP, 选择程序)”

说明

功能 SP（选择程序）选择一个主动的 NC 文件系统文件，以对其进行处理。即该文件之前必须就已装载在 NC 中。

编程

句法:	SP("文件")	
名称:	选择程序	
参数:	“文件”	NC 文件的完整路径数据

示例

此功能使用以下句法进行数据管理：

- 带返回值

```
SP ("//NC/MPF.DIR/MYPROG.MPF", VAR1)
```

VAR1 = 0 文件已装载。

VAR1 = 1 文件未装载。

- 无返回值：

```
SP ("//NC/MPF.DIR/MYPROG.MPF")
```

6.3.11 文件存取：RDFILE、WRFILE、RDLINEFILE、WRLINEFILE

说明

采用 INI 句法对文件进行读写访问时可使用功能 RDFILE 和 WRFILE。

对文件中的各行进行读写访问时可使用功能 RDLINEFILE 和 WRLINEFILE。

编程

句法：	RDFILE(“文件名 + 路径”，“段”，“关键字”)	
说明：	从文件中读	
参数：	文件名 + 路径	路径和文件名
	段	INI 文件中的段落
	关键字	INI 文件中的关键字

句法：	WRFILE(数值，“文件名 + 路径”，“段”，“关键字”)	
说明：	写入文件	
参数：	Value	待写入的值
	文件名 + 路径	路径和文件名
	段	INI 文件中的段落
	关键字	INI 文件中的关键字

句法:	RDFILE (“文件名 + 路径”, 行编号)	
说明:	从文件中读取一行	
参数:	文件名 + 路径	路径和文件名
	行号	行号 第一行从 0 开始编号。

句法:	WRLINEFILE (数值, “文件名 + 路径”)	
说明:	在文件末尾写一行	
参数:	值	待写入的值
	文件名 + 路径	路径和文件名

说明

- 文件不得位于 NC 文件系统（数据维护）中。
- 如果文件不存在或者到达了文件末尾又或者是出现了其他错误，都会相应地设置变量 **FILE_ERR** 和 **ERR**。您可事先使用文件功能 **Exist Program (EP)** 检查一个文件是否存在。
- 处理过的文件会以“UTF-8” 编码（无 BOM（Byte Order Mask））。读取的文件要求以“UTF-8” 编码。
- 通过文件功能 **Delete Program (DP)** 可在需要时将文件显式删除。

示例

从 INI 文件中读:

前提条件/假设:

文件 C:/tmp/myfile.ini 的内容:

```
<...>
[MyData]

MyName=Daniel
<...>
```

```
MyVar = RDFILE("C:/tmp/myfile.ini", "MyData", "MyName")
```

结果:

MyVar 现在包含值 "Daniel" 。

写入 INI 文件:

前提条件/假设:

VAR5=12

```
WRFILE (VAR5, "C:/tmp/myfile.ini", "MySession", "NrOfSessions")
```

结果:

文件 **C:/tmp/myfile.ini** 的内容:

<...>

```
[MySession]
```

```
NrOfSessions=12
```

<...>

读取文件的第 4 行:

前提条件/假设:

c:/tmp/myfile.mpf:

R[0]=0 的内容

```
F3500 G1
```

```
MYLOOPIDX1: X0 y-50 Z0
```

```
    X150 Y50 Z10
```

```
    R[0]=R[0]+1
```

```
GOTOB MYLOOPIDX1
```

```
M30
```

```
MyVar = RDLINFILE("C:/tmp/myfile.mpf", 4)
```

结果:

MyVar 现在包含值 "R[0]=R[0]+1"

6.3 功能

在文件末尾写入:

前提条件/假设:

VARX=123

VARY=456

```
WRLINEFILE("F100 X" << VARX << " Y" << VARY, "C:/tmp/mypp.mpf")
```

结果:

c:/tmp/mypp.mpf 的内容:

<...>

F100 X123 Y456

6.3.12 对话框行(DLGL)

说明

在对话框的对话框行中可以根据确定的情况给出短文本（信息或者输入帮助）。

标准字体大小时允许的字符数量： 约 50 个字符

编程

句法:	DLGL("字符串")	
说明:	显示对话框行的文本	
参数:	字符串	显示在对话框中的文本

示例

```
IF Var1 > Var2
    DLGL("值太大!")
ENDIF
```

：如果变量 1>变量 2，则对话框行中显示文本“值过大！”。

6.3.13 DEBUG

说明

通过功能 **DEBUG** 可在 **Run MyScreen** 用户屏幕的设计阶段为您提供分析帮助。采用功能 **DEBUG** 可对括号中返回的运行时间的表述进行检测。结果会在日志文件“**easyscreen_log.txt**”中作为独立条目加以记录。

在每条记录前会附加一个带有方括号的当前时间戳（参见下例）。

建议在对时间要求苛刻的部件上根据情况将 **DEBUG** 输出文件删除。建议在设计阶段结束后为 **DEBUG** 输出文件加上注释。对不断增多的日志文件“**easyscreen_log.txt**”的写访问可能会导致速度变慢。

编程

句法:	DEBUG(表述)
说明:	在日志文件“ easyscreen_log.txt ”中进行记录
参数:	待检测的表述，由此在日志文件中生成一条记录

示例

```
IF Var1 > Var2
  DEBUG("Value of ""Var1"": " << Var1)
; "easyscreen_log_txt" 中的记录:
[10:22:40.445] DEBUG: Value of "Var1":
123.456
ENDIF
```

6.3.14 退出对话框(EXIT)

说明

通过功能 **EXIT** 可以退出对话框并返回主对话框。如果不存在主对话框，则退出新配置的操作界面并返回标准应用程序。

6.3 功能

编程（没有参数）

句法:	EXIT
说明:	退出对话框
参数:	- 无 -

示例

```

PRESS (HS1)
EXIT
END_PRESS
    
```

说明

如果调用带传输变量的当前对话框，则变量值改变并且返回初始对话框。
 变量值分别分配给通过功能"LM"从初始对话框传输到后续对话框的变量。可以最多传输 20 个变量值，各自之间通过逗号隔开。

说明

变量/变量值的顺序必须根据 LM 功能的传递变量顺序进行，以此达到分配明确。如果一些变量值没有规定，则这些变量不改变。改变的传输变量在功能 LM 后立即在初始对话框中生效。

使用传输变量编程

句法:	EXIT [(VARx)]
说明:	退出对话框，传输一个或者多个变量
参数:	VARx 计算变量

示例

```

//M(屏幕窗口 1)
...
PRESS (HS1)
    
```

```

LM("屏幕窗口 2","CFI.COM",1, POSX, POSY,
直径)
; 中断屏幕 1 并显示屏幕 2。 传输变量 POSX、 POSY
和直径。
DLGL("退出屏幕 2")
; 从屏幕 2 返回后在屏幕 1 的对话框中显示文本：
退出屏幕 2。

END_PRESS
...
//END

//M(屏幕 2)
...
PRESS(HS1)
EXIT(5, , 计算_直径)
; 退出屏幕 2 并返回到屏幕 1 的 LM 后的行中。 此
时，赋值 5 给变量 POSX 并分配变量“计算_直径”的
值给变量“直径”。 变量 POSY 包含当前值。

END_PRESS
...
//END

```

6.3.15 转换栏或列表框栏的动态列表操作

说明

功能 LISTADDITEM、LISTINSERTITEM、LISTDELETEITEM 和 LISTCLEAR 用于对转换栏或列表框栏进行动态列表操作。

这些功能只对本身具有列表的变量有效，例如

- “基本”列表

```
DEF VAR_AC1 = (I/* 0,1,2,3,4,5,6,7,8) 或
```

- “扩展”列表

```
DEF VAR_AC2 = (I/* 0="AC0", 1="AC1", 2="AC2", 3="AC3", 4="AC4",
5="AC5", 6="AC6", 7="AC7", 8="AC8").
```

当变量指向一个数组时，例如 DEF VAR_AC3 = (I/* MYARRAY)，该功能无法使用，否则全局数组会发生变化。

一个变量应至少在 DEF 行中定义了一个值。这样可确定为“基本”或“扩展”型列表。

此后仍允许将列表完全删除以及全部重建。但类型“基本”或“扩展”必须保留或无法动态更改。

编程

句法:	LISTINSERTITEM (变量名, 位置, <i>ItemValue</i> [, <i>ItemDispValue</i>])	
说明:	在某个特定位置添加元素	
参数:	变量名	
	位置	列表中需要添加元素的位置
	<i>ItemValue</i>	列表条目的值
	<i>ItemDispValue</i>	应在列表中显示的值

句法:	LISTADDITEM (变量名, <i>ItemValue</i> [, <i>ItemDispValue</i>])	
说明:	在列表末尾附加一个元素	
参数:	变量名	
	<i>ItemValue</i>	列表条目的值
	<i>ItemDispValue</i>	应在列表中显示的值

句法:	LISTDELETEITEM (变量名, 位置)	
说明:	删除某个元素	
参数:	变量名	
	位置	列表中要删除元素的位置

句法:	LISTCOUNT (变量名)	
说明:	提供当前列表元素数	
参数:	变量名	

句法:	LISTCLEAR (变量名)	
说明:	删除整个列表	
参数:	变量名	

示例

说明

以下几个示例彼此互为基础，它们的顺序对各个结果的理解起着重要作用。

前提条件/假设:

```
DEF VAR_AC = (I/* 0="Off",1="On"/1/,"Switch"/WR2)
```

在列表末尾附加一个元素 -1="Undefined":

```
LISTADDITEM("VAR_AC", -1, ""Undefined"")
```

结果: 0="Off", 1="On", -1="Undefined"

在位置 2 上添加一个元素 99="Maybe":

```
LISTINSERTITEM("VAR_AC", 2, 99, ""Maybe"")
```

结果: 0="Off", 1="On", 99="Maybe", -1="Undefined"

确定当前列表元素数:

```
REG[10]=LISTCOUNT("VAR_AC")
```

结果: REG[10] = 4

删除位置 1 上的元素:

```
LISTDELETEITEM("VAR_AC", 1)
```

结果: 0="Off", 99="Maybe", -1="Undefined"

删除整个列表:

```
LISTCLEAR("VAR_AC")
```

结果: 列表为空

6.3 功能

6.3.16 评估(EVAL)

说明

功能 **EVAL** 评估作出的输出结果，然后执行。为此可以首先在运行期间建立表达式。例如用于变量上的显示存取。

编程

句法:	EVAL (<i>exp</i>)	
说明:	评估表达式	
参数:	exp	逻辑表达式

示例

```

VAR1=(S)
VAR2=(S)
VAR3=(S)
VAR4=(S)
CHANGE ()
    REG[7] = EVAL("VAR"<<REG[5]) ; 如果 REG[5] 的值为 3， 则括号中的表达式为 VAR3。 然后，分
                                配 REG[7] VAR3 的值。
    IF REG[5] == 1
        REG[7] = VAR1
    ELSE
        IF REG[5] == 2
            REG[7] = VAR2
        ELSE
            IF REG[5] == 3
                REG[7] = VAR3
            ELSE
                IF REG[5] == 4
                    REG[7] = VAR4
                ENDIF
            ENDIF
        ENDIF
    ENDIF
END_CHANGE
    
```

6.3.17 退出装载软键(EXITLS)

说明

通过功能 EXITLS 可以离开当前的操作界面并装载一个定义的软键栏。

编程

句法:	EXITLS ("软键栏", "路径名")	
说明:	退出时装载软键栏	
参数:	软键栏	待装载的软键栏名称
	路径名	要装载的软键栏目录路径

示例

```

PRESS (HS1)
    EXITLS ( "软键栏 1", "AEDITOR.COM" )
END_PRESS

```

6.3.18 功能 (FCT)

说明

外部功能存放在一个 DLL 文件中，并通过配置文件定义行中的条目识别。

说明

一个外部功能必须至少有一个返回参数。

编程

句法:	FCT 功能名称 = ("文件"/返回类型/固定参数的类型/可变参数的类型)	
	FCT InitConnection = ("c:\tmp\xyz.dll"/I/R,I,S/I,S)	
说明:	例如可从 LOAD 方法中或在 PRESS 方法中执行外部功能的调用。	

6.3 功能

参数:	功能名称	外部功能的名称
	文件	DLL 文件的完整路径数据
	返回类型	返回值的数据类型
	固定参数类型	值参数
	可变参数类型	参考参数
	数据类型用逗号隔开。	

例如可从 **LOAD** 方法中或在 **PRESS** 方法中执行外部功能的调用。

示例:

```
press (vs4)
RET = InitConnection (VAR1, 13, "Servus", VAR2, VAR17)
end_press
```

外部功能的结构

外部功能必须遵循特定的符号:

句法:	extern "C" dlllexport void InitConnection (ExtFctStructPtr FctRet, ExtFctStructPtr FctPar, char cNrFctPar)	
说明:	仅在 Windows 环境下执行 DLL 导出 区分符和传输参数为固定设定。通过传输的结构传送原先的调用参数。	
参数:	cNrFctPar	调用参数的数量 = FctPar 中结构单元的数量
	FctPar	指示某结构单元栏, 其包含相应的调用参数 (含数据类型)。
	FctRet	指示某结构用于返回功能值 (含数据类型)。

传输结构定义

```
union CFI_VARIANT
(
char b;
short int i;
double r;
```



```

        char*                s;
    )
typedef struct ExtFctStructTag
    (
        char                cTyp;
        union CFI_VARIANT    value;
    )ExtFctStruct;
typedef struct ExtFct* ExtFctStructPtr;

```

如需独立于操作平台（Windows、Linux）开发外部功能，则不可使用关键字 `__declspec(dllexport)`。只在 Windows 下需要此关键字。例如在 Qt 下可使用以下宏：

```

#ifdef Q_WS_WIN
    #define MY_EXPORT __declspec(dllexport)
#else
    #define MY_EXPORT
#endif

```

函数声明如下：

```

extern "C" MY_EXPORT void InitConnection
    (ExtFctStructPtr FctRet, ExtFctStructPtr FctPar, char cNrFctPar)

```

如需将通过 Run MyScreens 设计的画面用于 NCU 和 PCU/PC，则必须略去二进制文件的扩展名：

```

FCT InitConnection = ("xyz"/I/R,I,S/I,S)

```

在省略了绝对路径信息时，Run MyScreens 会首先在项目索引中搜索二进制文件。

6.3.19 生成代码(GC)

说明

功能 GC（Generate Code）从 OUTPUT 方法中生成 NC 代码。

编程

句法:	GC ("名称"[,"目标文件"][,Opt],[Append])
说明:	生成 NC 代码

6.3 功能

参数:	名称	OUTPUT 方法名称，作为代码生成的基础
	目标文件	HMI 或者 NC 文件系统的目标文件路径数据。 如果未规定目标文件（只能在编程支持中），则在该位置上写入代码，在该位置处光标停留在当前打开的文件中。
	可选	可选注释生成
	0:	（缺省设置）生成带有反编译注释的代码（另见 反编译 (页 163)）。
	1:	在生成的代码中不生成注释。 说明： 该代码不能反编译（另见 Auto-Hotspot）。
	Append	只有当规定目标文件情况下，该参数才有意义。
	0:	（预设）当文件已存在情况下，删除旧的内容。
	1:	当文件已存在情况下，在文件开始处写入新的代码。
	2:	在文件已存在的情况下，在结束处加入新的代码。

示例

```

//M(测试 GC/"代码生成:")
DEF VAR1 = (R//1)
DEF VAR2 = (R//2)
DEF D_NAME
LOAD
    VAR1 = 123
    VAR2 = -6
END_LOAD
OUTPUT(CODE1)
    "Cycle123(" VAR1 "," VAR2 ")"
    "M30"
END_OUTPUT

PRESS(VS1)
    D_NAME = "\\MPF.DIR\MESSEN.MPF"
    GC("CODE1",D_NAME)           ; 将 OUTPUT 方法的 NC 代码写入文件 \MPF.DIR
                                   \MESSEN.MPF 中:
                                   Cycle123(123, -6)
                                   M30
END_PRESS
    
```

反编译

- **目标文件没有数据:**

功能 GC 只能在编程支持中使用并待当前编辑器中打开的文件内写入 NC 代码。NC 代码可以反编译。如果功能 GC 在没有目标文件数据的情况下在“Run MyScreens”下设计，则此时发出故障信息。

- **目标文件数据:**

从 OUTPUT 方法中生成的代码输入到目标文件中。如果不存在目标文件，则在 NC 文件系统中设立。如果目标文件在 HMI 文件系统中，则该文件存放在硬盘上。不设立使用注释行（反编译需要的信息），即无法进行反编译。

指定目标文件的特殊性

基本上，指定目标文件有两种方式：

- **NC 格式:** /_N_MPF_DIR/_N_MY_FILE_MPF

该文件存放在 NC 上的目录 MPF 下。

- **DOS 格式:** d:\abc\my_file.txt 或者 \\RemoteRechner\files

\my_file.txt

文件写入在硬盘上规定的目录中或者指定的计算机上，此时硬盘上的目录或者远程计算机必须已存在。

说明

无效的变量在生成的 NC 代码中产生一个空字符串，如果读取该代码，在日志中出现一条出错信息。

反编译时的特殊情况

在子对话框中无法调用 GC 功能，因为在子对话框中可以使用源自主对话框的变量，然而在直接调用时并不存在。

在通过编辑器手动介入生成的代码时不允许改变由代码生成产生的值的符号数目。这些改变会妨碍到反编译。

解决方法：

1. 反编译
2. 通过设计的对话框输入修改（例如：99 → 101）
3. GC

6.3 功能

6.3.20 口令功能

概述

提供以下可用的口令功能：

- 设置口令
- 删除口令
- 更改口令

HMI_LOGIN 功能：设置新口令

句法:	HMI_LOGIN (<i>Passwd</i>)	
说明:	通过功能 HMI_LOGIN() 可发送一个口令到 NCK，通过该口令设置当前的存取等级。	
参数:	Passwd	口令

示例

```
REG[0] = HMI_LOGIN("CUSTOMER") ; 口令成功设置时，结果 = TRUE，否则为 FALSE
```

HMI_LOGOFF 功能：删除口令

句法:	HMI_LOGOFF	
说明:	通过功能 HMI_LOGOFF 可复位当前的存取等级。	
参数:	- 无 -	

示例

```
REG[0] = HMI_LOGOFF ; 口令成功删除时，结果 = TRUE，否则为 FALSE
```

HMI_SETPASSWD 功能：更改口令

句法:	HMI_SETPASSWD (<i>AC, Passwd</i>)	
说明:	通过功能 HMI_SETPASSWD 可对当前口令级或更低口令级的口令进行更改。	
参数:	AC	访问等级
	Passwd	口令

示例

```
REG[0] = HMI_SETPASSWD(4, "MYPWD") ; 口令成功更改时, 结果 = TRUE, 否则为 FALSE
```

6.3.21 装载数组(LA)**说明**

通过功能 LA（装载数组）可以从另一个文件装载一个数组。

编程

句法:	LA (名称 [, 文件])	
说明:	从文件装载数组	
参数:	名称	待装载数组的名称
	文件	在该文件中定义数组

说明

如果在当前的配置文件中，一个数组要由另一个配置文件中的数组替代，则数组名称必须相同。

示例

```

; 文件 maske.com 部分摘录
DEF VAR2 = (S/*ARR5/"关闭", "转换栏")
PRESS (HS5)
  LA("ARR5","arrayext.com")
; 从文件 arrayext.com 中
; 装载数组 ARR5
  VAR2 = ARR5[0]
; 代替 "Aus"/"Ein" ("关"/"开") 显示在 VAR2
; 的转换栏
; "上"/"下"/"右"/"左"
END_PRESS
//A (ARR5)
("关"/"开")
//END

; 文件 arrayext.com 部分摘录
//A (ARR5)
("上"/"下"/"右"/"左")
//END

```

说明

请注意，在通过 LA 功能分配给变量转换栏另一个数组后，必须赋予一个有效值给变量。

6.3.22 装载块 (LB)**说明**

通过功能 LB (装载块) 可以在运行期间内装载带有子程序的块。首先 LB 要在 LOAD (装载) 方法中设计，由此可以随时调用装载的子程序。

说明

子程序也可以直接在对话框中定义，如果那样则不必装载。

编程

句法:	LB("块名称"[, "文件"])	
说明:	运行时装载子程序	
参数:	块名称	块标记名称
	文件	配置文件的 路径数据 预设 = 当前的配置文件

示例

```

LOAD
LB ("PROG1")           ; 在当前的配置文件中查找块"PROG1"，接着装载块。
  LB ("PROG2", "XY.COM") ; 在配置文件 XY.COM 中查找块"PROG2"，接着装载块。
END_LOAD

```

6.3.23 装载屏幕窗口 (LM)

说明

通过功能 LM 可以载入一个新的对话框。取决于对话框切换的模式（见下表），还可采用该功能实现信息对话框。

主对话框/子对话框

能够调用其他对话框并且不能自行结束的对话框称为主对话框。由主对话框调用的对话框是子对话框。

编程

句法:	LM("名称"[, "文件"][, MSx [, VARx]])
说明:	载入对话框

6.3 功能

参数:	名称	待载入对话框的名称
	文件	配置文件的路径数据 (HMI 文件系统或者 NC 文件系统); 缺省设置: 当前的配置文件
	MSx	对话框切换模式
	0:	(缺省设置) 退出当前对话框, 装载并显示新的对话框。在 EXIT 时 (参见“退出”功能) 返回到标准应用程序。通过参数 MSx 可以确定在对话框切换时是否退出当前对话框。保留当前的对话框, 可以将变量传递到新的对话框中。 使用参数 MSx 的优点在于, 在切换时无需总是重新初始化对话框, 而是保留当前对话框的数据和设计, 简化数据传输
	1:	自功能 LM 起中断当前对话框, 装载并显示新的对话框 (例如用于实现信息对话框)。退出时关闭子对话框并且返回到主对话框的中断位置。 中断时, 不处理主对话框中的 UNLOAD (卸载) 块。
VARx	前提条件: MS1 列出可以从主对话框传送到子对话框的变量。可以最多传输 20 个变量, 各自之间通过逗号隔开。	

说明

参数 VARx 总是只传递变量值, 也就是说变量可以在子对话框中读写, 但是不可见。由于子对话框向主对话框返回传递变量可以通过功能 EXIT (退出) 进行。

示例

```

屏幕 1 (主对话框): 跳至子对话框屏幕 2 (带变量传输)
PRESS (HS1)

    LM ("屏幕 2", "CFI.COM", 1, POSX, POSY, 直径)
                                ; 中断屏幕 1 并显示屏幕 2: 传输变量 POSX、POSY 和直径。
DLGL ("退出屏幕 2")          ; 从屏幕 2 返回后在屏幕 1 的对话框行中显示文本: 退出屏幕 2。
END_PRESS
    
```


屏幕 2 (子对话框)：返回主对话框屏幕 1 (带变量传输)

PRESS (VS8)

EXIT (POSX, POSY, DURCHMESSER)

；隐藏屏幕 2 并继续显示屏幕 1，传输修改的变量 POSX、POSY 和直径。

END_PRESS

6.3.24 装载软键(LS)

说明

通过功能 LS 可以显示另一个软键栏。

编程

句法:	LS("名称"[, "文件"[, 合并])	
说明:	显示软键栏	
参数:	名称	软键栏名称
	文件	配置文件的路径数据 (HMI 文件系统或者 NC 文件系统) 预设置: 当前的配置文件
	Merge	
	0:	删除所有存在的软键, 输入重新设计的软键。
	1:	预设置 仅重新设计的软键覆盖现有的软键, 其他软键 (= HMI 应用程序的软键) 的功能和文本保持不变。

示例

PRESS (HS4)

LS ("软键栏 2", , 0) ; 软键栏 2 覆盖现有的软键栏, 删除显示的软键。

END_PRESS

说明

只要编译器还没有显示出对话框，即还没有处理 LM 功能，则在登入软键描述块和软键栏描述块的 PRESS（按压）方法中总是仅设计一个 LS 命令或者 LM 命令，并且没有其它动作。

功能 LS 和 LM 仅允许在软键 PRESS 方法中调用，然而不作为导航键（PU、PD、SL、SR、SU、SD）上的反应。

6.3.25 Load Grid (LG)

说明

表格描述(Grid)可以通过 LG 方法在 LOAD 方法内动态提供。

为此可以通过 LG 方法指定一个表格，必须已将变量定义为 Grid 变量并引用已存在的有效表格。

编程

句法:	LG (栅格名称, 变量名称 [, 文件名称])	
说明:	载入表格	
参数:	栅格名称	表格（栅格）名称，用双引号括起
	变量名	应指定表格的变量名称，用双引号括起
	文件名	文件名称，在该文件中定义表格（栅格），用双引号括起 如果表格未在文件中定义，在该文件中也定义变量，则必须指定。

示例

```

//M(MyGridSample/"My Grid Sample")
DEF MyGridVar=(R/% MyGrid1///WR2////100,,351,100)
LOAD
    LG("MyGrid1","MyGridVar","mygrids.com")
END_LOAD
    
```

mygrids.com 的内容:

```
//G(MyGrid1/0/5)
(I///,"MyGrid1"/wr1//"/1"/80/1)
(R3///"LongText1","R1-R4"/wr2//"$R[1]"/80/1)
(IBM///"LongText2","M2.2-M2.5"/wr2//"/M2.2"/80/,1)
(R3///"LongText3","R9,R11,R13,R15"/wr2//"$R[9]"/110/2)
//END
```

6.3.26 多次选择 SWITCH

说明

使用 SWITCH 命令可以对变量的不同数值进行检验。

在 SWITCH 命令中设置的表达式会与在接下来的 CASE 命令中设置的所有数值进行比较。

如不相同，则会继续和之后的 CASE 命令进行比较。如果之后有 DEFAULT 命令并且之前一直没有与在 SWITCH 命令中设置的表达式相同的 CASE 命令，则会执行 DEFAULT 命令之后的指令。

如果相同，则会执行接下来的指令，直到出现 CASE、DEFAULT 或 END_SWITCH 命令为止。

示例

```
FOCUS
  SWITCH (FOC)
    CASE "VarF"
      DLGL("Variable ""VarF"" has the input focus.")
    CASE "VarZ"
      DLGL("Variable ""VarZ"" has the input focus.")
    DEFAULT
      DLGL("Any other variable has the input focus.")
  END_SWITCH
END_FOCUS
```

6.3.27 多次读取 NC PLC (MRNP)

说明

用命令 MRNP 可以通过在寄存器中的一次存取输入多个 NC/PLC 变量。这种存取比通过单个存取读取速度明显加快。NC/PLC 变量必须在相同区域的 MRNP 命令中。

NC/PLC 变量区域分为如下几种：

- 一般 NC 数据 (\$MN..., \$SN..., /nck/...)
- 通道专用的 NC 数据 (\$MC..., \$SC..., /channel/...)
- PLC 数据 (DB..., MB..., /plc/...)
- 相同轴专用的 NC 数据 (\$MA..., \$SA...)

编程

句法:	MRNP (变量名称 1*变量名称 2[* ...], 寄存器索引)
说明:	读取多个变量
参数:	变量名称用“*”作为分隔符。按照命令中变量名称的顺序，采用寄存器 REG[寄存器索引] 中的值和下列值。 关系到： 第一个变量的值在 REG[寄存器索引] 中。 第二个变量的值在 REG[寄存器索引 + 1] 中，以次类推

说明

请注意，变量列表最多为 500 个字符，寄存器的数量也有一定限制。

示例

```
MRNP("$R[0]*$R[1]*$R[2]*$R[3]",1) ; REG[1] 至 REG[4] 以变量值 $R[0] 至 $R[3] 描述。
```

NC 变量

系统提供所有的机床数据、设定数据和 R 参数，但是只提供一部分的系统变量（参见 可用的系统变量列表 (页 225)）。

可访问所有全局的和通道专用的用户变量（GUD）。本地和程序全局的用户变量无法编辑。

机床数据	
全局机床数据	\$MN_...
轴专用的机床数据	\$MA_...
通道专用的机床数据	\$MC_...

设定数据	
全局设定数据	\$SN_...
轴专用的设定数据	\$SA_...
通道专用的设定数据	\$SC_...

系统变量	
R 参数 1	\$R[1]

PLC 变量

所有 PLC 数据可用。

PLC 数据	
数据模块 x 的字节 y 位 z	DBx.DBXy.z
数据模块 x 的字节 y	DBx.DBBy
数据模块 x 的字 y	DBx.DBWy
数据模块 x 的双字 y	DBx.DBDb
数据模块 x 的实数 y	DBx.DBRy
标记字节 x 位 y	Mx.y
标记字节 x	MBx
标记字 x	MWx
标记双字 x	MDx
输入字节 x 位 y	Ix.y 或者 Ex.y
输入字节 x	IBx 或者 EBx
输入字 x	IWx 或者 EWx

6.3 功能

PLC 数据	
输入双字 x	IDx 或者 EDx
输出字节 x 位 y	Qx.y 或者 Ax.y
输出字节 x	QBx 或者 ABx
输出字 x	QWx 或者 AWx
输出双字 x	QDx 或者 ADx
数据模块 x 中长度 z 的字符串 y	DBx.DBsy.z

6.3.28 PI 服务

说明

通过功能 PI_START 可以在由 PLC 在 NC 区中启动 PI 服务（程序实例服务）。

说明

可用的 PI 服务列表请见功能手册之基本功能。

编程

句法:	PI_START (“传输字符串”)	
说明:	执行 PI 服务	
参数:	“传输字符串”	传输的字符串与 OEM 文献相反，应用双引号括起。

示例

```
PI_START("/NC,001,_N_LOGOUT")
```

说明

通道相关的 PI 服务总是与当前的通道有关。

刀具功能（TO 区）的 PI 服务总是以分配到当前通道的 TO 区为参考。

6.3.29 读取 (RNP) 和写入 (WNP) 系统或用户变量

说明

通过指令 RNP (读取 NC PLC) 可以读取 NC 或者 PLC 变量或者机床数据。

编程

句法:	RNP ("系统或者用户变量",值)	
说明:	读取 PLC 或者 NC 变量或者机床数据	
参数:	系统或者用户变量	NC 或者 PLC 变量名称
	值	要写入系统或者用户变量中的值。 如果值类型为字符串, 则必须以双引号括起。

示例

VAR2=RNP ("SAA_IN[2] ")	; 读取 NC 变量

说明

通过指令 WNP (写入 NC PLC) 可以写入 NC 或者 PLC 变量或者机床数据。

在每次处理功能 WNP 都重新存取 NC 变量或 PLC 变量, 即: 总是在 CHANGE 方法下存取 NC 变量或 PLC 变量。这在系统或者用户变量经常改变值情况下有意义。如果只进行一次 NC/PLC 存取, 则必须在 LOAD (装载) 或者 UNLOAD (卸载) 方法下设计。

编程

句法:	WNP ("系统或者用户变量", 值)	
说明:	写入 PLC 或者 NC 变量或者机床数据	
参数:	系统或者用户变量	NC 或者 PLC 变量名称
	值	要写入系统或者用户变量中的值。 如果值类型为字符串, 则必须以双引号括起。

6.3 功能

示例

```
WNP ("DB20.DBB1",1) ; 写入 PLC 变量
```

6.3.30 RESIZE_VAR_IO 和 RESIZE_VAR_TXT

说明

使用命令 RESIZE_VAR_IO() 和 RESIZE_VAR_TXT() 可以更改变量的输入/输出分量或文本分量的几何尺寸。

设置了新的几何尺寸后，屏幕的所有栏都会调整位置，就像屏幕从一开始就是按这样的位置配置的。从而根据配置，各栏会相互重新对齐。

编程

句法:	RESIZE_VAR_IO (变量名, [X], [Y], [宽度], [高度]) RESIZE_VAR_TXT (变量名, [X], [Y], [宽度], [高度])	
说明:	更改变量的输入/输出分量或文本分量的几何尺寸。	
参数:	变量名	要更改输入/输出分量或文本分量的几何尺寸的变量的名称。
	X	左上 X 坐标
	Y	左上 Y 坐标
	宽度	宽度
	高度	高度

说明

如未给定 X、Y、宽度或高度的值，则保留当前值。如将其中一个值给定为 -1，则会为由“Run MyScreens”预设的标准位置分量。

示例

```
RESIZE_VAR_IO("MyVar1", 200, , 100) ; 变量“MyVar1”的 IO 分量偏移至 x 位置 200 像素，宽度设为 100 像素。之前的 y 位置和高度值被保留。
```


6.3.31 寄存器(REG)

寄存器说明

寄存器用于在两个不同的对话框之间切换数据。寄存器分配给每个对话框并且在载入第一个对话框时生成，以 0 或空字符串预占。

说明

寄存器不允许直接在 OUTPUT 方法中被用于 NC 代码的生成。

编程

句法:	REG[x]	
说明:	定义寄存器	
参数:	x	寄存器编号，以 x = 0...19 类型： REAL 或者 STRING = VARIANT x ≥ 20 的寄存器由西门子使用。

寄存器值说明

寄存器值的分配在方法中设计。

说明

如果由一个对话框通过功能 LM 生成另一个对话框，则寄存器的内容自动接受入新的对话框中并在第二个对话框中用于其它计算。

编程

句法:	<i>名称.val = 寄存器值</i> 或者 <i>名称 = 寄存器值</i>	
说明:		
参数:	名称	寄存器名称
	寄存器值	寄存器的值

6.3 功能

示例

```

UNLOAD
  REG[0] = VAR1           ; 寄存器 0 赋值为变量 1 的值
END_UNLOAD

UNLOAD
  REG[9].VAL = 84        ; 寄存器 9 赋值为变量 84 的值
END_UNLOAD

                                ; 在下列对话框中该寄存器可以在方法中重新分配本地变量。

LOAD
  VAR2 = REG[0]
END_LOAD
    
```

寄存器状态说明

通过状态特性可以在设计中查询寄存器是否包含一个有效值。

此外，当一个对话框作为主对话框使用时，可以使用寄存器的状态查询向一个寄存器中仅写入一个值。

编程

句法:	名称.vld	
说明:	该属性仅可读。	
参数:	名称	寄存器名称
返回值:		查询结果可能是:
	FALSE =	无效值
	TRUE =	有效值

示例

```

IF REG[15].VLD == FALSE    ; 查询寄存器值的有效性
  REG[15] = 84
ENDIF
VAR1 = REG[9].VLD          ; 向 Var1 分配 REG[9] 状态查询的值。
    
```

6.3.32 RETURN

说明

通过 RETURN 功能可以提前取消当前的子程序处理并返回到最后一次 CALL 命令的跳转位置。

如果子程序中没有设计 RETURN，则执行子程序，直至结束然后返回到跳转位置。

编程

句法:	RETURN	
说明:	返回跳转位置	
参数:	- 无 -	

示例

//B (PROG1)	; 块开始
SUB (UP2)	; 子程序开始
IF VAR1.val=="Otto"	
VAR1.val="Hans"	
RETURN	; 如果变量值 = Otto, 则赋值 "Hans" 给变量, 子程序在此位置结束。
ENDIF	
VAR1.val="Otto"	; 如果变量值 ≠ Otto, 则赋值 "Otto" 给变量。
END_SUB	; 子程序结束
//END	; 块结束

6.3.33 反编译

说明

在编程支持中可以反编译功能 GC 生成的 NC 代码并在所属输入对话框中的输入/输出栏中再次显示变量值。

编程

来自 NC 代码的变量接受至对话框中。此时对来自 NC 代码的变量值与来自配置文件的计算的变量值加以比较。如果存在不一致，则在日志文件中给出错误信息，因为在生成的 NC 代码中数值已改变。

如果一个变量在 NC 代码中多次存在，则在反编译时总是分析该变量最后一次出现的值。另外在日志文件中给出警告。

代码生成时，不在 NC 代码中使用的变量作为使用注释存储。通过使用注释标记所有反编译时需要的信息。使用注释不得更改。

说明

NC 代码块和使用注释如果在一行的起始处开始，则只能反编译。

示例：

在程序中有下列 NC 代码：

```
DEF VAR1 = (I//101)
OUTPUT(CODE1)
  "X" VAR1 " Y200"
  "X" VAR1 " Y0"
END_OUTPUT
```

在零件程序中存放下列代码：

```
;NCG#TestGC#\cus.dir\aeditor.com#CODE1#1#3#
X101 Y200
X101 Y0
;#END#
```

在反编译时读取编辑器：

```
X101 Y200
X222 Y0           ; 在零件程序中更改 X 的值 (X101 → X222)
```

在输入对话框中给出下列 VAR1 值： VAR1 = 222

参见

生成代码(GC) (页 145)

6.3.34 无注释反编译

说明

在编程支持中，可以对通过 GC 功能生成的 NC 代码执行**无注释反编译**，并在对应输入对话框的输入输出栏中显示变量值。

编程

可以按照以下方式执行 GC 指令，忽略在常规的代码生成中产生的注释行：

```
GC ("CODE1", D_NAME, 1)
```

产生的代码通常是无法反编译的，但是执行以下操作，便可以反编译生成的循环调用代码：

- **补充文件“easyscreen.ini”**

在文件“easyscreen.ini”中加入段落 [RECOMPILE_INFO_FILES]。在该段落中列出所有“ini”文件，这些文件说明了哪些循环可以进行无注释反编译：

```
[RECOMPILE_INFO_FILES]
IniFile01 = cycles1.ini
IniFile02 = cycles2.ini
```

可以指定多个“INI”文件，文件名称可自由选择。

- **新建包含循环说明的“INI”文件**

将含有循环说明的 ini 文件保存到以下路径中：

```
[系统用户目录]/cfg
[系统 oem 目录]/cfg
[系统插件目录]/cfg
```

在该文件中，每个循环需要占用一个单独的说明段落。段落的名称就是循环的名称：

```
[Cycle123]
Mname = TestGC
Dname = testgc.com
OUTPUT = Code1
Anzp = 3
Version = 0
Code_typ = 1
Icon = cycle123.png
Desc_Text = This is describing text
```

6.3 功能

Mname	窗口名称
Dname	对窗口进行定义的文件的名称
OUTPUT	所涉及 OUTPUT 方法的名称
Anzp	需要反编译的窗口的参数数量（指所有用 DEF 创建的变量，也包含辅助变量）
Version	循环的版本，可选
Icon	<p>在工步链程序中显示的图符，格式为“png”</p> <p>不同图片大小对应的图符分辨率：</p> <p>640 x 480 mm → 16 x 16 像素</p> <p>800 x 600 mm → 20 x 20 像素</p> <p>1024 x 768 mm → 26 x 26 像素</p> <p>1280 x 1024 mm → 26 x 26 像素</p> <p>1280 x 768 mm → 26 x 26 像素</p> <p>保存目录： [系统用户目录]/ico/ico<分辨率></p> <p>说明： 分辨率为“1280 x xx mm” 的图片保存在用于“1024 x 768 mm” 的文件夹下（仅针对工步链程序）。</p>
Desc_Text	工步链程序中显示画面上的说明性文本，可选，字符串最多 17 个字符（仅针对工步链程序）。

示例

```
//M(测试 GC/"代码生成:")
DEF VAR1 = (R//1)
DEF VAR2 = (R//2)
DEF D_NAME
LOAD
  VAR1 = 123
  VAR2 = -6
END_LOAD
OUTPUT(CODE1)
  "Cycle123(" VAR1 "," VAR2 ")"
  "M30"
END_OUTPUT

PRESS(VS1)
  D_NAME = "\\MPF.DIR\MESSEN.MPF"
```

GC("CODE1",D_NAME)	; 将 OUTPUT 方法的 NC 代码写入文件 \MPF.DIR \MESSEN.MPF 中: Cycle123(123, -6) M30
END_PRESS	

边界条件

- “无注释反编译”功能不具备“带注释反编译”的全部功能。
“无注释反编译”支持典型的循环调用，如 MYCYCLE(PAR1, PAR2, PAR3, ...)。但在函数调用行中不允许存在任何注释。但是，不在函数调用时传递的以及字符串 S 型的可选参数至少应带有空的引号，如 ""。否则，“Run MyScreens”会尝试使用逗号替代该参数，接着将其反编译为“被替代的循环调用”。
- 字符串型参数不允许待传递的字符串中含有逗号和分号。
- “无注释反编译”时，OUTPUT 方法中包含的所有变量都应位于括号内，这样功能“使用逗号替代缺少循环参数”才能生效。

示例:

允许:

```
OUTPUT
    "MYCYCLE (" MYPAR1 ", " MYPAR2 ", " MYPAR3 ") "
END_OUTPUT
```

不允许（变量 MYCOMMENT 位于右括号后面）:

```
OUTPUT
    "MYCYCLE (" MYPAR1 ", " MYPAR2 ", " MYPAR3 ") " MYCOMMENT
END_OUTPUT
```

6.3.35 向前/后查找(SF, SB)

说明

通过功能 **SF**（向前查找）、**SB**（向后查找）可以在编辑器当前的 NC 程序中从当前光标位置开始查找某个字符串并输出其值。

编程

句法:	SF ("字符串")	
名称:	Search Forward : 从当前光标位置向前查找	
句法:	SB ("字符串")	
名称:	Search Backward : 从当前光标位置向后查找	
参数:	字符串	要查找的文本

查找规则:

- 要查找的字符串和数值的单元在 NC 程序中前后必须有空格。
- 查找对象无法在注释中和字符串内查找。
- 给出的值必须是一个数字表达式，表达式形式“X1=4+5”无法识别。
- 可识别十六进制常数的形式 X1='HFFFF'、二进制常数的形式 X1='B10010' 和指数常数的形式 X1='-.5EX-4'。
- 在字符串和数值之间有下列符号，可以给出字符串值：
 - 无
 - 空格键
 - 等号

示例

允许以下的写入方式:

```
X100 Y200 ; 变量 Abc 包含数值 200
Abc = SB("Y")
X100 Y 200 ; 变量 Abc 包含数值 200
Abc = SB("Y")
```



```
X100 Y=200 ; 变量 Abc 包含数值 200
Abc = SB("Y")
```

6.3.36 字符串功能

概述

以下的功能允许进行字符串处理：

- 确定字符串长度
- 查找字符串中的一个字符
- 由左提取部分字符串
- 由右提取部分字符串
- 由字符串中间提取部分字符串
- 替换部分字符串
- 字符串的比较
- 将字符串插入到另一个字符串中
- 从字符串中删除一个字符串
- 删除空格符（从左或从右）
- 插入带格式化标识的值或字符串

LEN 功能：字符串长度

句法：	LEN(字符串 / 变量名称)	
说明：	确定一个字符串的字符数目。	
参数：	字符串	每个有效的字符串表达式。对于一个空字符串，会返回零。
	变量名称	每个有效的和表示的变量名称
	仅允许两个可行参数中的一个。	

6.3 功能

示例

```

DEF VAR01
DEF VAR02

LOAD
  VAR01="HALLO"
  VAR02=LEN(VAR01)           ; 结果 = 5
END_LOAD
    
```

NSTR 功能：查找字符串中的字符

句法:	INSTR (开始, 字符串 1, 字符串 2 [,方向])	
说明:	查找字符	
参数:	开始	从字符串 1 向字符串 2 查找的开始位置。如果从字符串 2 的开头开始查找，则指定为 0。
	字符串 1	要查找的字符。
	字符串 2	在该字符链中查找
	方向 (可选)	查找的方向 0: 从左向右 (预设) 1: 从右向左
	如果字符串 2 中没有包含字符串 1，则返回数值 0。	

示例

```

DEF VAR01
DEF VAR02

LOAD
  VAR01="HALLO/WELT"
  VAR02=INST(1,"/",VAR01)     ; 结果 = 6
END_LOAD
    
```

LEFT 功能：左边字符串

句法:	LEFT(字符串, 长度)	
说明:	LEFT 返回一个字符串, 该字符串从字符串左侧开始包含指定的字符数。	
参数:	字符串	字符串或者带有要处理的字符串的变量
	长度	要读取的字符数目

示例

```

DEF VAR01
DEF VAR02
LOAD
VAR01="HALLO/WELT"
  VAR02=LEFT(VAR01,5)           ; 结果 = "HALLO"
END_LOAD

```

RIGHT 功能：右边字符串

句法:	RIGHT(字符串, 长度)	
说明:	RIGHT 返回一个字符链, 该字符链从字符串右侧开始包含指定的字符数。	
参数:	字符串	字符串或者带有要处理的字符串的变量
	长度	要读取的字符数目

示例

```

DEF VAR01
DEF VAR02
LOAD
VAR01="HALLO/WELT"
  VAR02=RIGHT(VAR01,4)        ; 结果 = "WELT"
END_LOAD

```

6.3 功能

MIDS 功能：中间字符串

句法:	MIDS (字符串, 开始 [, 长度])	
说明:	MIDS 返回一个字符串, 该字符串从字符串的指定位置开始包含指定的字符数。	
参数:	字符串	字符串或者带有要处理的字符串的变量
	开始	开始, 从字符链中该处开始读取
	长度	要读取的字符数目

示例

```

DEF VAR01
DEF VAR02
LOAD
VAR01="HALLO/WELT"
    VAR02=LEFT (VAR01,4,4)           : 结果 = "LO/W"
END_LOAD
    
```

REPLACE 功能：替换字符

句法:	REPLACE (字符串, 查找字符串, 替换字符串 [, 开始 [, 计数]])	
说明:	功能 REPLACE 用另一个字符/字符链替代字符串中的一个字符/字符链。	
参数:	字符串	待通过替换字符串替换查找字符串的字符串。
	查找字符串	需被替代的字符串
	替代字符串	替代字符串 (位于查找字符串位置)
	开始	查找和替换的开始位置
	计数器	从开始位置起要开始查找的查找字符串的字符数量

返回值:		
	字符串 = 空字符串	复制字符串
	查找字符串 = 空字符串	复制字符串
	替换字符串 = 空字符串	复制字符串, 在该字符串中删除所有出现的查找字符串
	开始 > Len(长度)	空字符串
	计数 = 0	复制字符串

STRCMP 功能: 比较字符串

句法:	STRCMP (字符串 1, 字符串 2 [, <i>CaseInsensitive</i>])	
说明:	STRCMP 将一个字符链与另一个字符链进行比较。	
参数:	字符串 1	要与第二个字符链进行比较的字符链
	字符串 2	要与第一个字符链进行比较的字符链
	<i>CaseInsensitive</i>	比较时区分/不区分大/小写: 未说明或 FALSE = 区分大/小写 TRUE = 不区分大/小写

示例

REG[0]=STRCMP("Hugo", "HUGO")	; 结果 = 32 ; (<> 0, 字符串不相同)
REG[0]=STRCMP("Hugo", "HUGO", TRUE)	; 结果 = 0 ; (== 0, 字符串相同)

STRINSERT 功能: 插入字符串

句法:	STRINSERT (字符串 1, 字符串 2, 插入位置)	
说明:	STRINSERT 可在一个字符链的某个特定位置上插入另一个字符链。	
参数:	字符串 1	要向其中插入一个字符链的字符链
	字符串 2	要插入的字符链
	插入位置	插入字符链的位置

6.3 功能

示例

```
REG[0]=STRINSERT("Hello!", " world", 5) ; 结果 = "Hello world!"
```

STRREMOVE 功能：删除字符串

句法:	STRREMOVE (字符串 1, 删除位置, 计数)	
说明:	STRREMOVE 可从字符链的某一位置上删除一定数量的字符。	
参数:	字符串 1	要从其中删除字符的字符链
	删除位置	从字符链的该位置起删除字符
	计数	要删除的字符数量

示例

```
REG[0]=STRREMOVE("Hello world!", 5, 6) ; 结果 = "Hello!"
```

TRIMLEFT 功能：从字符串左侧删除空格符

句法:	TRIMLEFT (字符串 1)	
说明:	TRIMLEFT 可从字符链的左侧删除空格符。	
参数:	字符串 1	要从其左侧开始删除空格符的字符链

示例

```
REG[0]=TRIMLEFT(" Hello!") ; 结果 = "Hello!"
```

TRIMRIGHT 功能：从字符串右侧删除空格符

句法:	TRIMRIGHT (字符串 1)	
说明:	TRIMRIGHT 可从字符链的右侧删除空格符。	
参数:	字符串 1	要从其右侧开始删除空格符的字符链

示例

```
REG[0]=TRIMRIGHT("Hello!   ") ; 结果 = "Hello!"
```

FORMAT 功能：插入带格式化标识的值或字符串

句法:	FORMAT (带格式化标识的文本[,值 1]... [,值 28])	
说明:	FORMAT 功能可通过使用格式化标识在预设文本的某个位置插入最多 28 个值或字符串。	
格式化标识:	句法	%[标记][宽度][小数点后面的位置]类型
	标记	用于确定输出格式的可选字符： <ul style="list-style-type: none"> ● 右对齐或左对齐（“-”用于左对齐） ● 前面加零（“0”） ● 缺省值：当要输出的值的数位少于[宽度]定义时，使用空格符补齐
	宽度	确定一个非负数最小输出宽度的依据。如果值的数位少于依据所确定的，则使用空格符补齐。
	小数点后的数位	使用浮点数时，该可选参数用来确定小数点后面的位数。
	类型	类型说明符用来确定 PRINT 指令会输出哪些数据格式。 该说明符必须进行给定。 <ul style="list-style-type: none"> ● d:整型值 ● f: 浮点数 ● s:字符串 ● o: 八进制 ● x:十六进制 ● b:二进制

示例

```
DEF VAR1
DEF VAR2
LOAD
  VAR1 = 123
```

6.3 功能

```
VAR2 = FORMAT("Hello %08b %.2f %s!", VAR1 + 1, 987.654321, "world")
; 结果 = "Hello 01111100 987.65 world!"
END_LOAD
```

参见

使用字符串 (页 91)

6.3.37 WHILE/UNTIL 循环

说明

使用 **DO-LOOP** 命令可实现一个循环。取决于配置，这个循环会一直执行，直到满足一个条件(**WHILE**)或一个条件适用(**UNTIL**)。

取决于配置，由于循环会影响系统的性能，因此请您慎重使用这些循环并且放弃使用循环中费时的动作。

例如，建议使用寄存器(**REG[]**)作为运行变量，因为一般显示变量（尤其是那些使用系统或用户变量连接的）会极其频繁地刷新或写入，这同样会影响系统性能。

借助功能 **DEBUG**（参见章节 **DEBUG** (页 137)）可对“Run MyScreens”方法的运行时间进行检测。这样还可对循环可能导致的问题（高 CPU 负载、响应性能降低）进行识别。

说明

由于 **FOR** 循环可被 **WHILE** 循环替代，因此在 **EasyScreen** 中不支持 **FOR** 循环的表达句法。

编程

```
DO
    <指令>
LOOP_WHILE <循环继续执行的条件>

DO
    <指令>
LOOP_UNTIL <循环结束的条件>
```



```
DO_WHILE <循环继续执行的条件>
  <指令>
LOOP

DO_UNTIL <循环结束的条件>
  <指令>
LOOP
```

示例

```
REG[0] = 5
DO
  DEBUG("OUTER: " << REG[0])
  REG[0] = REG[0] + 1
  REG[1] = -5

  DO
    DEBUG("INNER: " << REG[1])
    REG[1] = REG[1] + 1
    LOOP_WHILE REG[1] < 0

  LOOP_WHILE REG[0] < 10
```

```
REG[0] = 5
DO
  DEBUG("OUTER: " << REG[0])
  REG[0] = REG[0] + 1
  REG[1] = -5

  DO
    DEBUG("INNER: " << REG[1])
    REG[1] = REG[1] + 1
    LOOP_UNTIL 0 <= REG[1]

  LOOP_WHILE 10 > REG[0]
```

```
REG[0] = 5
```

6.3 功能

```
DO_WHILE 10 > REG[0]
  DEBUG("OUTER: " << REG[0])
  REG[0] = REG[0] + 1
  REG[1] = -5

  DO_UNTIL 0 <= REG[1]
    DEBUG("INNER: " << REG[1])
    REG[1] = REG[1] + 1
  LOOP
LOOP
```

```
REG[0] = 5
DO_WHILE 10 > REG[0]
  DEBUG("OUTER: " << REG[0])
  REG[0] = REG[0] + 1
  REG[1] = -5

  DO
    DEBUG("INNER: " << REG[1])
    REG[1] = REG[1] + 1
  LOOP_UNTIL 0 <= REG[1]
LOOP
```

6.3.38 循环执行脚本： START_TIMER, STOP_TIMER

说明

借助计时器可循环调用 SUB 方法。为此提供了功能 START_TIMER()和 STOP_TIMER()。

说明

只可为每个 SUB 方法配置一个计时器。

编程

句法:	START_TIMER ("SUB 名称", 间隔)	
说明:	开始 SUB 方法的循环执行	
参数:	SUB 名称:	要循环调用的 SUB 方法的名称
	间隔:	以 ms 为单位的时间间隔

句法:	STOP_TIMER ("SUB 名称")	
说明:	停止 SUB 方法的循环执行	
参数:	SUB 名称:	要停止其计时器的 SUB 方法的名称

示例

```
//M(TimerSample/"My timer")
DEF MyVariable=(I//0/,"Number of cyclic calls:"/WR1)
VS1=("Start%ntimer")
VS2=("Stop%ntimer")

SUB(MyTimerSub)
    MyVariable = MyVariable + 1
END_SUB

PRESS(VS1)
    ; 每 1000 ms 调用一次 SUB "MyTimerSub"
    START_TIMER("MyTimerSub", 1000)
END_PRESS

PRESS(VS2)
    STOP_TIMER("MyTimerSub")
END_PRESS
```

如果为已经分配给一个 SUB 方法的计时器重新调用 **START_TIMER**，则在有差异时会接收新的间隔。否则这两次调用都会被忽略。

由于系统的限制，最短间隔为 100 ms。

如果为一个 SUB 方法调用 **STOP_TIMER**，而当时没有正在运行的计时器，则会忽略此次调用。

图形单元和逻辑单元

7.1 线、分割线、矩形、圆形和椭圆形

说明

线、分割线、矩形和椭圆形/圆形在 `LOAD` 方法中配置：

- 可将填充色设置为系统背景色，以得到透明的矩形。
- 只要高度和宽度设为相同，元素 `ELLIPSE` 上就可生成圆形。
- 水平和垂直分割线始终与窗口宽度或窗口高度完全一致。分割线不会产生滚动条。
如果此前一直将元素 `RECT` 用于显示分割线，例如 `RECT(305,0,1,370,0,0,1)`，这会被编程支持识别到并自动转换为 `V_SEPARATOR(305,1,"#87a5cd", 1)`。这同时适用于垂直和水平分割线。

LINE 元素

编程：

句法：	<code>LINE (x1,y1,x2,y2,f,s)</code>	
说明：	定义线	
参数：	<code>x1</code>	起点的 x 坐标
	<code>y1</code>	起点的 y 坐标
	<code>x2</code>	终点的 x 坐标
	<code>y2</code>	终点的 y 坐标
	<code>f</code>	线的颜色
	<code>s</code>	线的样式： 1 = 实线 2 = 虚线 3 = 点线 4 = 点虚线

RECT 元素

编程:

句法:	RECT (x,y,w,h,f1,f2,s)	
说明:	定义矩形	
参数:	x	左上 x 坐标
	y	左上 y 坐标
	w	宽度
	h	高度
	f1	边框颜色
	f2	填充色
	s	边框样式: 1 = 实线 2 = 虚线 3 = 点线 4 = 点虚线

ELLIPSE 元素

编程:

句法:	ELLIPSE(x,y,w,h,f1,f2,s)	
说明:	定义椭圆或圆形	
参数:	x	左上 x 坐标
	y	左上 y 坐标
	w	宽度
	h	高度
	f1	边框颜色
	f2	填充色
	s	边框样式: 1 = 实线 2 = 虚线 3 = 点线 4 = 点虚线

高度与宽度值相同时为圆形。

V_SEPARATOR 元素

编程:

句法:	V_SEPARATOR (x,w,color,pen)	
说明:	定义垂直分割线	
参数:	x	x 位置
	w	线条粗细
	color	颜色
	pen	线的样式: 1 = 实线 2 = 虚线 3 = 点线 4 = 点虚线

H_SEPARATOR 元素

编程:

句法:	H_SEPARATOR (y,h,color,pen)	
说明:	定义水平分割线	
参数:	y	y 位置
	h	线条粗细
	color	颜色
	pen	线的样式: 1 = 实线 2 = 虚线 3 = 点线 4 = 点虚线

另见

LOAD (页 113)

7.2 定义数组

参见

CLEAR_BACKGROUND (页 128)

7.2 定义数组

定义

通过数组可以排列或者保存同一数据类型的数据，从而可以通过索引存取数据。

说明

数组可以是一维或者二维。一个一维数组可视为带有一行或者一列的一个二维数组。

数组通过标记 `//A` 定义并通过 `//END` 结束。行和列数目任意。一个数组有下列结构：

编程

句法:	<code>//A(名称)</code> <code>(a/b...)</code> <code>(c/d...)</code> ... <code>//END</code>	
说明:	定义数组	
参数:	名称	数组名称
	<code>a, b, c, d</code>	数组值 STRING 类型的值必须用双引号括起。

示例

<code>//A(螺纹)</code>	<code>; 高度/螺距/底直径</code>
<code>(0.3 / 0.075 / 0.202)</code>	
<code>(0.4 / 0.1 / 0.270)</code>	
<code>(0.5 / 0.125 / 0.338)</code>	
<code>(0.6 / 0.15 / 0.406)</code>	
<code>(0.8 / 0.2 / 0.540)</code>	


```
(1.0 / 0.25 / 0.676)
(1.2 / 0.25 / 0.676)
(1.4 / 0.3 / 1.010)
(1.7 / 0.35 / 1.246)
//END
```

7.2.1 存取数组单元的值

说明

通过属性值（名称.val）可以继续传送一个数组存取值。

行索引（数组的行编号）和列索引（数组的列编号）各自从 0 开始。如果显示数组外的行索引或者列索引，则输出值 0 或者空字符串并且变量 ERR 的值为 TRUE。当未找到查找关键字时，变量 ERR 同样为 TRUE。

编程

句法:	名称 [Z,[M,[C]]].val 或者 名称 [Z,[M,[C]]]
说明:	存取仅带有一列的一维数组:
句法:	名称 [S,[M,[C]]].val 或者 名称 [S,[M,[C]]] 或者
说明:	存取仅带有一行的单维数组
句法:	名称 [Z,S,[M,[C]]].val 或者 名称 [Z,S,[M,[C]]]
说明:	存取二维数组

7.2 定义数组

参数:	名称:	数组名称	
	Z:	行值 (行索引或者查找关键字)	
	S:	列值 (列索引或者查找关键字)	
	M:	存取模式	
		0	直接
		1	按行查找, 列直接
		2	按列查找, 行直接
		3	查找
		4	查找行索引
	C:	比较模式	
0		查找关键字必须位于行或者列的值范围内。	
1		查找关键字必须准确找到	
示例:	VAR1 = MET_G[REG[3], 1,0].VAL	; 分配 Var1 数组中的一个值 MET_G	

存取模式

- **存取模式“直接”**
在存取模式“直接”(M = 0)情况下, 数组上的存取通过行索引(以 Z 表示)和列索引(以 S 表示)实现。不评估比较模式 C。
- **存取模式“查找”**
在存取模式 M = 1、2 或者 3 情况下, 查找总是在行 0 或者列 0 中实现。

模式 M	行值 Z	列值 S	输出值
0	行索引	栏索引	行 Z 和列 S 中的值
1	查找关键字: 在列 0 中查找	列索引, 从该列中读取值	查找的行和列 S 中的值
2	行索引, 从该行中读取 返回值	查找关键字: 在行 0 中查找	行 Z 和查找的列中的值

模式 M	行值 Z	列值 S	输出值
3	查找关键字： 在列 0 中查找	查找关键字： 在行 0 中查找	查找的行和查找的列中的 值
4	查找关键字： 在列 S 中查找	行索引，在该行中查找	行索引
5	行索引，在该行中查找	查找关键字： 在行 Z 中查找	栏索引

比较模式

在使用比较模式 $C = 0$ 时，查找行或者查找列的内容以升序分类。如果查找关键字小于第一个元素或者大于最后一个元素，则给出值 0 或者一个空字符并且错误变量 ERR 为真。

在使用比较模式 $C = 1$ 时，查找关键字必须可在查找行或者查找列中找到。如果没有找到查找关键字，则给出值 0 或者一个空字符并且错误变量 ERR 为真。

7.2.2 举例：存取数组单元

前提条件

下列定义两个数组是下面例子的前提条件。

```
//A(螺纹)
(0.3 / 0.075 / 0.202)
(0.4 / 0.1 / 0.270)
(0.5 / 0.125 / 0.338)
(0.6 / 0.15 / 0.406)
(0.8 / 0.2 / 0.540)
(1.0 / 0.25 / 0.676)
(1.2 / 0.25 / 0.676)
(1.4 / 0.3 / 1.010)
(1.7 / 0.35 / 1.246)
//END
```

7.2 定义数组

```
//A(Array2)
      ("BEZ" /      "STG" /      "KDM" )
      (0.3 /        0.075 /      0.202 )
      (0.4 /        0.1 /        0.270 )
      (0.5 /        0.125 /      0.338 )
      (0.6 /        0.15 /       0.406 )
      (0.8 /        0.2 /        0.540 )
      (1.0 /        0.25 /       0.676 )
      (1.2 /        0.25 /       0.676 )
      (1.4 /        0.3 /        1.010 )
      (1.7 /        0.35 /       1.246 )

//END
```

示例

- **存取模式 1 示例:**

在 Z 中有查找关键字。该关键字总是在列 0 中查找。通过查找的关键字的行索引给出列 S 中的值:

VAR1 = 螺纹[0.5,1,1] ; VAR1 值为 0.125

说明:

数组“螺纹”的列 0 中查找值 0.5 并给出列 1 中查找的相同行的值。

- **存取模式 2 示例:**

在 S 中有查找关键字。该关键字总是在行 0 中查找。通过查找的关键字的列索引给出行 Z 中的值:

VVAR1 = ARRAY2[3,"STG",2] ; VAR1 值为 0.125

说明:

在数组“Array2”的行 0 中查找带有内容“STG”的列。得到查找的列中的值和带有索引 3 的行。

- **存取模式 3 示例:**

在 Z 和 S 中总是有一个查找关键字。通过 Z 中的关键字在列 0 中找到行索引和通过 S 中的关键字在行 0 中找到列索引。通过查找的行索引和列索引输出数组中的值:

VAR1 = ARRAY2[0.6,"STG",3] ;VAR1 值为 0.15

说明:

在数组“Array2”的列 0 中查找带有内容 0.6 的行，在数组“Array2”的行 0 中查找带有内容“STG”的列。根据 VAR1 给出查找的行和列中的值。

- **存取模式 4 示例:**

在 Z 中有查找关键字。在 S 中有要查找的列索引。给出查找的关键字的行索引:

VAR1 = 螺纹[0.125,1,4] VAR1 值为 2

说明:

在数组“螺纹”的列 1 中查找值 0.125 并根据 VAR1 给出查找的值的行索引。

- **存取模式 5 示例:**

在 Z 中有要查找行的行索引。查找关键字在 S 中。给出查找的关键字的列索引:

VAR1 = 螺纹[4,0.2,5,1] VAR1 值为 1

说明:

在数组“螺纹”的行 4 中查找值 0.2 并根据 VAR1 给出查找的值的列索引。已选择比较模式 1，因为行 4 的值不是按升序分类的。

7.2.3 查询数组单元的状态

说明

通过属性状态可以查询数组存取是否提供一个有效值。

编程

句法:	名称 [Z, S, [M[,C]]].vld
说明:	该属性仅可读。
参数:	名称 数组名称
返回值:	FALSE = 无效值 TRUE = 有效值

示例

```
DEF MPIT = (R///"MPIT",,"MPIT",""/wr3)
DEF PIT  = (R///"PIT",,"PIT",""/wr3)
PRESS(VS1)
  MPIT = 0.6
  IF MET_G[MPIT,0,4,1].VLD == TRUE
    PIT  = MET_G[MPIT,1,0].VAL
    REG[4] = PIT
    REG[1] = "OK"
```

7.3 表格描述(栅格)

```
ELSE  
    REG[1] = "ERROR"  
ENDIF  
END_PRESS
```

7.3 表格描述(栅格)

定义

与数组相反，表格(栅格)的值会持续更新。这会涉及（例如）来自 NC 或 PLC 的数值的表格式显示。表格是按列组织排列的，因此一列中始终为同一类型的变量。

分配

表格描述参考内容收录在变量描述中：

- 变量定义确定显示的值，表格单元定义确定屏幕的外观和布置。来自变量定义行的输入/输出栏属性将会接收到表格中。
- 表格的可见区域通过输入/输出栏的宽度和高度确定。如果比可见区域位置存在更多的行和列，则可以通过水平和垂直滚动条进行查看。

表格名称

NCK/PLC 相同类型值的表格名称，它可以通过通道模块编译地址。表格名称通过一个前置的极限值或者转换栏的 % 符号进行区分。表格命名符可以通过逗号分隔，还可以跟随一个文件名，该文件名指定定义表格描述的文件。

同类型值的表格命名符，例如来自 NCK 或 PLC 的值。表格命名符在配置极限值或转换栏的地方给定。表格命名符前会带有 % 符号。之后可用逗号隔开并加上一个文件名。这用于给定在哪个文件中定义表格描述。表格默认会在配置文件中查找屏幕本身的配置位置。

表格定义也可以通过功能 Load Grid (LG)进行动态加载，例如在 LOAD 方法中。

系统或者用户变量

这些参数保留为空以用于表格，因为详细信息中要显示的变量会在列定义行中进行给定。表格描述可以动态提供。

示例

变量 `MyGridVar` 的定义，该变量在您的输入/输出栏（左边距：100，上边距：标准，宽度：350，高度：100）中显示为一个栅格“`MyGrid1`”。

```
DEF MyGridVar=(V/% MyGrid1/////////100,,350,100)
```

另见

变量参数 (页 77)

Load Grid (LG) (页 154)

7.3.1 定义表格(栅格)

说明

表格块由以下部分组成：

- 标题描述
- 1 至 n 列描述

编程

句法：	//G (表格名称/表格类型/行数 / [固定行属性], [固定列属性])
说明：	定义表格(栅格)

7.3 表格描述(栅格)

参数:	表格名称	这里表格名称不使用前缀 % 符号。表格名称只能在一个对话框中使用一次。	
	表格类型	0 (预设置)	PLC 或者用户数据 (NCK 和通道专用的数据) 表
		1	备用
	行数	行数包括标题行	
		固定行或者固定列无法滚动显示。列数由设计的列的数目给定。	
	列标题行的显示	-1:	不显示列标题行
	0:	显示列标题行 (默认)	
固定列的数量	0:	无固定列	
	1 ... n:	列数应在各列都可见的范围内, 即不需水平滚动	

示例

<code>//G(grid1/0/5/-1,2)</code>	列标题行隐藏并有 2 个固定列
<code>//G(grid1/0/5/,1)</code>	列标题行显示并有 1 个固定列
<code>//G(grid1/0/5)</code>	列标题行显示并且无固定列

7.3.2 定义列

说明

为表格(栅格)使用带下标的变量非常重要。索引号码对于带有一个或者多个索引的 PLC 或者 NC 变量比较重要。

您可以对表格中显示的值在由属性确定的权限范围和可能已设定的极限范围内直接加以编辑。

编程

句法:	(类型/极限值/空/长文本,列标题/属性/帮助图形/系统或用户变量/列宽/偏移 1、偏移 2、偏移 3) 另见 扩展配置句法 (页 38)。	
说明:	定义列	
参数:	和变量类似	
	类型	数据类型
	极限值	最小极限值, 最大极限值
	长文本, 列标题	
	属性	
	帮助图形	
	系统或者用户变量	作为变量在双引号内给出 PLC 或者 NC 变量。
	列宽	参数, 单位像素
偏移	在分配的偏移参数内规定步宽 (在该步宽中各个索引应指数运算), 以填写该列。 <ul style="list-style-type: none"> ● 偏移 1: 第 1 个索引的步宽 ● 偏移 2: 第 2 个索引的步宽 ● 偏移 3: 第 3 个索引的步宽 	

文本文件的列标题

列标题可以规定为文本或者文本号码 (\$8xxxx), 并且同样无法滚动显示。

改变列属性

可动态改变的 (可写) 的列属性称为:

- 极限值 (最大、最小)、
- 列标题(st)、
- 属性(wr, ac 和 li)
- 帮助图形(hlp)和
- BTSS 变量(var)。

通过定义行中的变量命名符和列索引 (以 1 开始) 改变列属性。

示例: VAR1 [1] .st="列 1"

7.3 表格描述(栅格)

无法在 **LOAD** 方法中读取列属性。

对于列定义，可以规定属性 **wr**、**ac** 和 **li**。

另见

Load Grid (LG) (页 154)

7.3.3 表格中的焦点控制（栅格）

说明

通过列和行属性可以在表格中设置和确定聚焦：

- 名称 **Row**
- 名称 **Col**

编程

表格的每行都具有属性 **Val** 和 **VId**。

对于行属性写入和读出，除了定义行中的变量命名符之外，还规定一个行索引和列索引。

句法:	名称[行索引, 列索引].VId 或者 名称[行索引, 列索引]
说明:	Val 属性
句法:	名称[行索引, 列索引].VId
说明:	VId 属性

示例

```
Var1[2,3].val=1.203
```

如果没有规定行索引和列索引，则适用于聚焦行的索引，即：

```
Var1.Row =2
```

```
Var1.Col=3
```

```
Var1.val=1.203
```

7.4 自定义小部件

7.4.1 自定义小部件

说明

通过自定义小部件，可以在对话框内设计用户专用的显示部件。



软件选项

使用自定义小部件功能额外需要下列软件选项：

“SINUMERIK Integrate Run MyHMI /3GL” (6FC5800-0AP60-0YB0)

编程

定义:	DEF (名称)	
句法:	(W//"/", (库名称) . (类别名称) "////a,b,c,d);	
说明:	W	自定义小部件
参数:	名称	自定义小部件的名称，可自由选择
	库名称	可自由选择，dll (Windows)的名称或 so (Linux)库文件的名称
	类别名称	可自由选择，是前面指出的库的类别功能的名称
	a, b, c, d	小部件的位置和大小

示例

在对话框配置文件中，可以按照以下方式自定义小部件：

```
DEF Cus =
(W//"/", "slestestcustomwidget.SlEsTestCustomWidget"////
20,20,250,100);
```

7.4.2 自定义小部件库的结构

说明

自定义小部件库基本上都包含了一个定义类别，在对话框配置文件中，应在指定库名称后指定该类别的名称。Run MyScreens 根据库名称访问同名 dll 文件，例如：

```
slestestcustomwidget.dll
```

编程

dll 文件的类别定义应为：

```
#define SLESTESTCUSTOMWIDGET_EXPORT Q_DECL_EXPORT  
  
class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget : public QWidget  
{  
    Q_OBJECT  
    ....  
    ....  
}
```

7.4.3 自定义小部件接口的结构

说明

自定义小部件的显示需要在库中增加一个接口。该接口包含了 Run MyScreens 初始化自定义小部件的宏定义。接口在 cpp 文件中定义。文件名称可以自由定义，例如：
sleswidgetfactory.cpp

编程

可以按照下面的方式定义接口：

```
#include "slestestcustomwidget.h" ; 将相关自定义小部件的标题文件添加到文件开头  
....  
//Makros ; 保持宏定义  
....
```

```
WIDGET_CLASS_EXPORT(SlEsTestCustomWidget); 在文件末尾申明相关的自定义小部件
dget)
```

示例

自定义小部件文件“sleswidgetfactory.cpp”的内容，类别名称为“SlEsTestCustomWidget”：

```
#include <Qt/qglobal.h>
#include "slestestcustomwidget.h"

////////////////////////////////////
// MAKROS FOR PLUGIN DLL-EXPORT - DO NOT CHANGE
////////////////////////////////////

#ifndef Q_EXTERN_C
#ifdef __cplusplus
#define Q_EXTERN_C extern "C"
#else
#define Q_EXTERN_C extern
#endif
#endif

#define SL_ES_FCT_NAME(PLUGIN) sl_es_create_ ##PLUGIN
#define SL_ES_CUSTOM_WIDGET_PLUGIN_INSTANTIATE( IMPLEMENTATION , PARAM ) \
{ \
    IMPLEMENTATION *i = new PARAM; \
    return i; \
}

#ifndef Q_WS_WIN
#   ifdef Q_CC_BOR
#       define EXPORT_SL_ES_CUSTOM_WIDGET_PLUGIN(PLUGIN,PARAM) \
Q_EXTERN_C __declspec(dllexport) void* \
__stdcall SL_ES_FCT_NAME(PLUGIN) (QWidget* pParent) \
SL_ES_CUSTOM_WIDGET_PLUGIN_INSTANTIATE( PLUGIN,PARAM )
#   else
#       define EXPORT_SL_ES_CUSTOM_WIDGET_PLUGIN(PLUGIN,PARAM) \
Q_EXTERN_C __declspec(dllexport) void* SL_ES_FCT_NAME(PLUGIN) \
(QWidget* pParent) \
SL_ES_CUSTOM_WIDGET_PLUGIN_INSTANTIATE( PLUGIN,PARAM )
#   endif
#else
#endif
```

7.4 自定义小部件

```
# define EXPORT_SL_ES_CUSTOM_WIDGET_PLUGIN(PLUGIN, PARAM) \
    Q_EXTERN_C void* SL_ES_FCT_NAME(PLUGIN) (QWidget* pParent) \
    SL_ES_CUSTOM_WIDGET_PLUGIN_INSTANTIATE( PLUGIN, PARAM )
#endif

#define WIDGET_CLASS_EXPORT(CLASSNAME) \
    EXPORT_SL_ES_CUSTOM_WIDGET_PLUGIN(CLASSNAME, CLASSNAME(pParent))

////////////////////////////////////
// FOR OEM USER - please declare here your widget classes for export
////////////////////////////////////

WIDGET_CLASS_EXPORT(SlEsTestCustomWidget)
```

7.4.4 自定义小部件与对话框的互动 - 自动数据交换

自定义小部件与对话框相互影响并能显示或处理值。

条件

自动数据交换会在以下条件下进行：

条件	方向
打开或反编译对话框时	对话框 → 自定义小部件
执行用于生成循环调用的 GC 指令时	自定义小部件 → 对话框

编程

针对互动，以下定义都是必须的：

对话框设计扩展

定义:	DEF (变量)	
句法:	((类型)/I5/","(变量)","/wr2/)	
变量类型:	类型	带任意数据类型（无 W）的标准输入区（无栅格或转换）
参数:	变量	用于数据交换的变量的任意名称
输入模式:	wr2	读和写

示例

```
DEF CUSVAR1 = (R//5/"" , "CUSVAR1", ""/wr2/)
```

类定义扩展

在自定义小部件的类定义中必须创建 **QProperty**，其名称应与在对话框设计中选择的变量名称一致，例如：

```
Q_PROPERTY(double CUSVAR1 READ cusVar1 WRITE setCusVar1);
```

示例

dll 文件的类定义应为：

```
#define SLESTESTCUSTOMWIDGET_EXPORT Q_DECL_EXPORT

class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget : public QWidget
{
    Q_OBJECT
    Q_PROPERTY(double CUSVAR1 READ cusVar1 WRITE setCusVar1);
    ....
    ....
}
```

7.4.5 自定义小部件与对话框的互动 - 手动数据交换

除自动数据交换外，也可进行手动数据交换。该交换是动态进行的，即在对话框运行期间进行。可执行以下操作：

- 可读取和写入自定义小部件的属性。
- 可从 **Run MyScreens** 配置中调用自定义小部件的方法。
- 可以响应一个特定的自定义小部件信号并借此在 **Run MyScreens** 配置中调用子程序 (SUB)。

7.4 自定义小部件

7.4.5.1 读取和写入属性

说明

Run MyScreens 配置中提供了 `ReadCWProperties` 和 `WriteCWProperties` 函数用于读取和写入自定义小部件的属性。

编程

句法:	ReadCWProperty (“变量名称”, “属性名称”)	
说明:	读取自定义小部件的属性	
参数:	变量名称	分配到一个自定义小部件的对话框变量的名称
	属性名称	待读取的自定义小部件属性的名称
返回值:	自定义小部件属性的当前值	

句法:	WriteCWProperty (“变量名称”, “属性名称”, “值”)	
说明:	写入 CustomWidget 的属性	
参数:	变量名称	分配到一个自定义小部件的对话框变量的名称
	属性名称	待写入的自定义小部件属性的名称
	值	应在 CustomWidget 属性中写入的值

示例

示例 1:

读取包含对话框变量 “MyCWVar1” 的自定义小部件属性 “MyStringVar” 并在寄存器 7 中分配值。

CustomWidget 类声明:

```
class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget : public QWidget
{
    Q_OBJECT
    Q_PROPERTY(QString MyStringVar
                READ myStringVar
                WRITE setMyStringVar);
    ...
}
```

对话框配置:

```
DEF MyCWVar1 = (W///,"slestestcustomwidget.SlEsTestCustomWidget")
PRESS (VS1)
    REG[7]=ReadCWProperty("MyCWVar1", "MyStringVar")
END_PRESS
```

示例 2:

在包含对话框变量“MyCWVar1”的 CustomWidget 属性“MyRealVar”中写入计算结果“3 + sin(123.456)”。

CustomWidget 类声明:

```
class SLESTESTCUSTOMWIDGET_EXPORT SlEsTestCustomWidget : public QWidget
{
    Q_OBJECT
    Q_PROPERTY(double MyRealVar
                READ myRealVar
                WRITE setMyRealVar);
    ...
}
```

对话框配置:

```
DEF MyCWVar1 = (W///,"slestestcustomwidget.SlEsTestCustomWidget")
PRESS (VS1)
    WriteCWProperty("MyCWVar1", "MyRealVar", 3 + sin(123.456))
END_PRESS
```

7.4.5.2 执行自定义小部件的方法**说明**

Run MyScreens 配置中提供了 CallCWMethod 函数用于执行自定义小部件的方法。

7.4 自定义小部件

待调用的自定义小部件方法最多可以有 10 个传输参数。

支持以下传输参数数据格式：

- bool
- uint
- int
- double
- QString
- QByteArray

编程

句法:	CallCWidgetMethod (“变量名称”, “方法名称[, 自变量 0][, 自变量 1 ... [, 自变量 9]”)	
说明:	调用 CustomWidget 方法	
参数:	变量名称	分配到一个自定义小部件的对话框变量的名称
	方法名称	待调用的自定义小部件方法的名称
	自变量 0 - 9	CustomWidget 方法的传输参数 支持的数据格式: 见上 提示: 传输参数将始终传输 “ByVal”, 即始终只传输值, 不传输变量。
返回值:	自定义小部件方法的返回值 支持以下传输参数数据格式: <ul style="list-style-type: none"> • void • bool • uint • int • double • QString • QByteArray 提示: 即使自定义小部件方法的返回值的数据格式为 “void”, 也必须将该值分配给一个变量。	

示例

CustomWidget 类声明:

```
class SLESTESTCUSTOMWIDGET_EXPORT SLEsTestCustomWidget : public QWidget
{
    Q_OBJECT
    public slots:
        void myFunc1(int nValue, const QString& szString, double dValue);
    ...
}
```

对话框配置:

```
DEF MyCWVar1 = (W///,"slestestcustomwidget.SLEsTestCustomWidget")
DEF MyStringVar1 = (S)
DEF MyRealVar = (R)

PRESS (VS3)
    REG[9] = CallCWMethod("MyCWVar1", "myFunc1", 1+7, MyStringVar1, sin(MyRealVar) -
8)
END_PRESS
```

7.4.5.3 响应一个自定义小部件信号

说明

在 Run MyScreens 中，可以响应一个特定的自定义小部件信号 (invokeSub()) 并借此调用一个子程序 (SUB)。

传输值（自定义小部件信号 -> SUB）时，有 10 个在配置中可与寄存器 (REG) 进行比较的全局变量（即所谓的 SIGARG）可用。系统会保存通过自定义小部件信号传输的值。

支持以下传输参数数据格式:

- bool
- uint
- int
- double
- QString
- QByteArray

编程

调用子程序:

句法:	<code>void invokeSub(const QString& rszSignalName, const QVariantList& rvntList);</code>	
说明:	自定义小部件信号, 通过其调用一个 Run MyScreens 子程序。	
参数:	<code>rszSignalName</code>	待调用的 Run MyScreens 子程序名称
	<code>rvntList</code>	QVariantList Array, 用于传输保存在全局参数 SIGARG 中的参数, 可在配置中使用。 最大数量: 10 个元素 支持的数据格式: 见上 提示: 传输参数将始终传输 “ByVal”, 即始终只传输值, 不传输变量。

待调用的子程序:

句法:	<code>SUB (on_ <变量名称>_ <信号名称>)</code> ... <code>END_SUB</code>	
说明:	响应一个自定义小部件信号	
参数:	变量名称	分配到一个自定义小部件的对话框变量的名称。
	信号名称	自定义小部件信号的名称
	<code>SIGARG 0 - 9</code>	自定义小部件方法的传输参数。 支持的数据格式: 见上 提示: 传输参数将始终传输 “ByVal”, 即始终只传输值, 不传输变量。

示例

CustomWidget 类声明:

```
class SLESTESTCUSTOMWIDGET_EXPORT SLEstTestCustomWidget : public QWidget
{
    Q_OBJECT signals: void invokeSub(const QString& szSubName, const QVariantList&
vntList);
    ...
}
```

CustomWidget 类:

```
QVariantList vntList;  
vntList << 123.456;  
emit invokeSub("MySub", vntList);
```

对话框配置:

```
DEF MyCWVar1 = (W///, "slestestcustomwidget.SlEsTestCustomWidget")  
SUB(on_MyCWVar1_MySub)  
    DEBUG("SUB(on_MyCWVar1_MySub) was called with parameter: "" << SIGARG[0] << """)  
END_SUB
```

结果 easyscreen_log.txt:

```
[10:22:40.445] DEBUG: SUB(on_MyCWVar1_MySub) was called with parameter: "123.456"
```

7.4 自定义小部件

操作区“Custom（定制）”

8.1 这样激活操作区“自定义”

激活操作区“Custom”

“自定义”操作区在交付时未激活。

1. 首先将文件 `slamconfig.ini` 从 `[系统西门子目录]/cfg` 目录下复制到 `[系统 oem 目录]/cfg` 下或者复制到相应的 `[系统插件目录]/cfg` 或 `[系统用户目录]/cfg` 下。
2. 为激活操作区“Custom”，必须进行以下输入：

```
[Custom]
Visible=True
```

结果

激活后，操作区“Custom”的软键位于 HSK4 (= 预设置) 扩展菜单栏的主菜单(F10)中。

操作区“Custom”显示为带有可设置标题的空窗口。所有的水平和垂直软键都是可设置的。

8.2 这样设计“自定义”软键

设计用于操作区“Custom”的软键

在文件 `slamconfig.ini` 中设计操作区“Custom”软键的标签和位置。

登入软键的设计如下：

1. 为了使用**语言相关的文本**替代软键上的标签，则在`[Custom]`段中必须有以下输入项：

```
TextId=MY_TEXT_ID
TextFile=mytextfile
TextContext=mycontext
```

 在该示例中软键显示语言相关的文本，文本 ID 为“MY_TEXT_ID”保存在文本文件 `mytextfile_xxx.qm` 中的“`MyContext`”下（xxx 代表语种缩写）。
2. 为了使用**与语言无关的文本**替代软键上的标签，则在`[Custom]`段中必须有以下输入项：

```
TextId=HELLO
TextFile=<empty>
TextContext=<empty>
```

 在该示例中操作区“Custom”软键在每种语言中都显示为“HELLO”。

8.3 这样设计操作区“自定义”

3. 除了文本外，软键上还可以显示**图标**。
为此在[Custom]段中必须有以下输入项：
Picture=mypicture.png
这样软键就会显示 mypicture.png 文件中的图标。图形和位图保存在以下路径下：[系统 oem 目录]/icol/ico<分辨率>。根据显示器的分辨率来使用相应目录下的图片。
4. 此外还可以设置软键的**位置**。为此在[Custom]段中必须有以下输入项：
SoftkeyPosition=12
位置 12 是预设置。这是操作区菜单扩展栏上的 HSK4。位置 1-8 是菜单栏上的 HSK1 到 HSK8，位置 9-16 是菜单扩展栏上的 HSK1 到 HSK8。

8.3 这样设计操作区“自定义”

设计用于操作区“Custom”的软键

进行操作区设计需要文件 easyscreen.ini 和 custom.ini。两个文件的模板位于目录 [系统西门子目录]/templates/cfg 下。

1. 首先将这些文件复制到目录 [系统 oem 目录]/cfg 下，再在这里进行修改。
2. 在文件 easyscreen.ini 中已经包含了操作区“自定义”的定义行：
;StartFile02 = area := Custom, dialog := SlEsCustomDialog,
startfile := custom.com
一行开头的“;”代表是注释内容。这一行即是注释并因此无效。要变其状态必须删除“;”。
通过该行中的属性“startfile”来定义，在选择操作区“自定义”时输入项参考设计文件 custom.com。
3. 将 Projektdatei custom.com 保存到目录 [系统 oem 目录]/proj 下。其中包括模拟操作区“程序”的 aeditor.com 文件的设计。所设计的登入软键于是显示在操作区“自定义”中。
4. 在文件 custom.ini 中设计用于对话框标题行的**与语言无关的文本**。
为此在模板中必须有以下输入项：
[Header]
Text=Custom
这些文本可以替换成用户自定义的文本。

5. 为了设计操作区“自定义”的**开始画面**，模板中必须有以下输入项：

```
[Picture]
```

```
Picture=logo.png
```

Logo.png 是在操作区“自定义”的开始对话框中显示的开始画面的名称。此处可以显示公司徽标或其他图片。文件保存在相应分辨率的文件夹下：*[系统 oem 目录]icol ...*

6. 如要在首次显示“自定义”操作区时直接显示某个“Run MyScreens”屏幕，则在模板中应存在以下记录：

```
; Mask shown with startup of area "custom"
```

```
[Startup]
```

```
;Startup = Mask:=MyCustomStartupMask, File:=mycustommasks.com
```

典型的应用场合，例如在启动 SINUMERIK Operate 时直接以某个“Run MyScreens”屏幕启动。

为此，在配置文件 **systemconfiguration.ini** 中应按以下方式设置段“[miscellaneous]”关键字“**startuparea**”：

```
[miscellaneous]
```

```
;name of the area to be shown at system startup
```

```
startuparea = Custom
```

8.4 操作区“自定义”的编程示例

示例

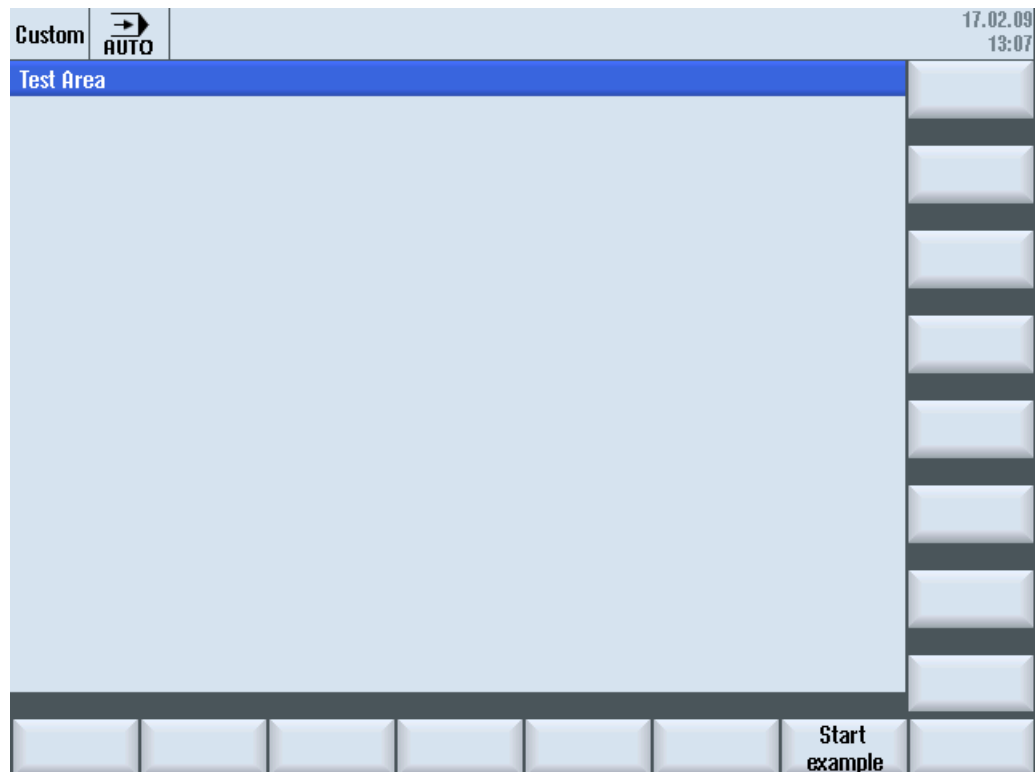


图 8-1 软键“Start example”示例

8.4 操作区“自定义”的编程示例

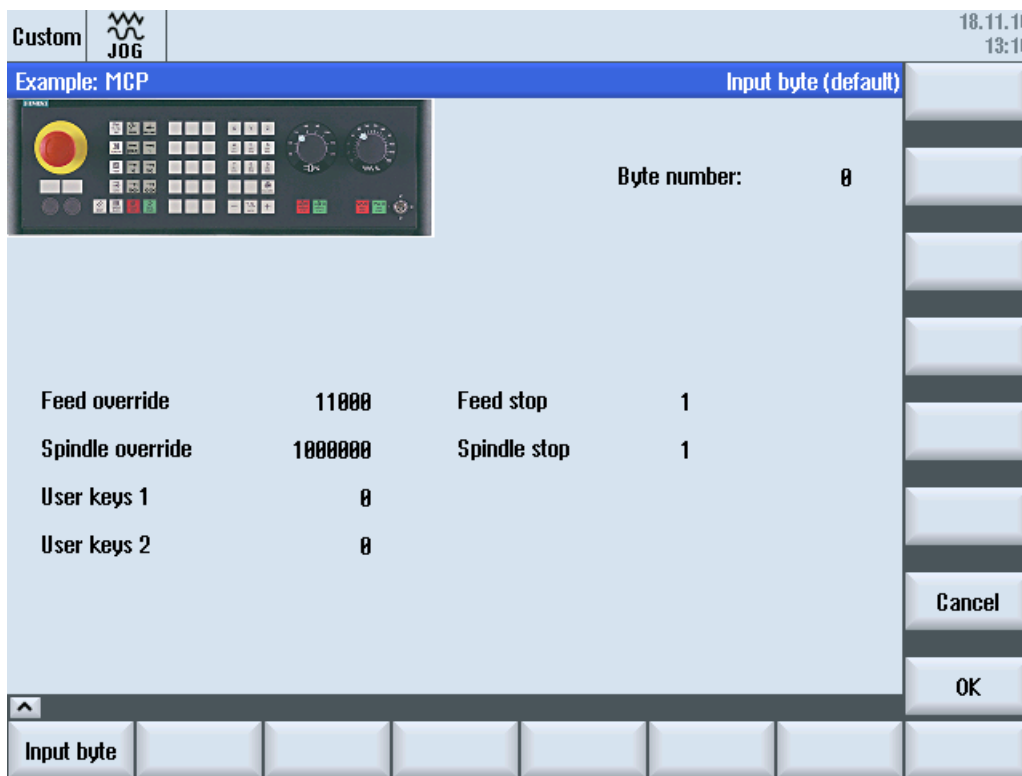


图 8-2 位图和文本栏示例

文件概览

需要以下文件：

- custom.com
- easyscreen.ini

编程

文件 custom.com 的内容：

说明

本例中链接的图形文件 mcp.png 仅用于示范。 如果希望改编本示例，可以使用自己的图形文件。

```
//S(Start)  
HS7=("Start example", sel, ac7)
```

```

PRESS (HS7)
  LM ("Maske4")
  END_PRESS
//END
//M(Maske4/"Example: MCP"/"mcp.png")
  DEF byte=(I/0/0/"Input byte=0 (default)", "Byte number:", ""/wr1,li1//380,40,100/480,40,50)
  DEF Feed=(IBB//0/""/"Feed override", ""/wr1//EB3"/20,180,100/130,180,100), Axistop=(B//0/""/"Feed
stop", ""/wr1//E2.2"/280,180,100/380,180,50/100)
  DEF Spin=(IBB//0/""/"Spindle override", ""/wr1//EB0"/20,210,100/130,210,100), spinstop=(B//
0/""/"Spindle stop", ""/wr1//E2.4"/280,210,100/380,210,50/100)
  DEF custom1=(IBB//0/""/" User keys 1", ""/wr1//EB7.7"/20,240,100/130,240,100)
  DEF custom2=(IBB//0/""/"User keys 2", ""/wr1//EB7.5"/20,270,100/130,270,100)
  DEF By1
  DEF By2
  DEF By3
  DEF By6
  DEF By7

HS1=("Input byte", SE1, AC4)
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("")
VS5=("")
VS6=("")
VS7=("Cancel", SE1, AC7)
VS8=("OK", SE1, AC7)

PRESS (VS7)
  EXIT
  END_PRESS

PRESS (VS8)
  EXIT
  END_PRESS

LOAD
  By1=1

```

8.4 操作区“自定义”的编程示例

```
By2=2
By3=3
By6=6
By7=7
END_LOAD

PRESS(HS1)
  Byte.wr=2
END_PRESS

CHANGE(Byte)
  By1=byte+1
  By2=byte+2
  By3=byte+3
  By6=byte+6
  By7=byte+7
  Feed.VAR="EB"<<By3
  Spin.VAR="EB"<<Byte
  Custom1.VAR="EB"<<By6
  Custom2.VAR="EB"<<By7
  Axisstop.VAR="E"<<By2<<".2"
  Spinstop.VAR="E"<<By2<<".4"
  Byte.wr=1
END_CHANGE

CHANGE(Axis stop)
  IF Axistop==0
    Axistop.BC=9
  ELSE
    Axistop.BC=11
  ENDIF
END_CHANGE

CHANGE(Spin stop)
  IF Spinstop==0
    Spinstop.BC=9
  ELSE
    Spinstop.BC=11
  ENDIF
END_CHANGE
//END
```

对话框选择

9.1 通过 PLC 软键选择对话框

设计

步骤说明：

- 在 `systemconfiguration.ini` 中有 `[keyconfiguration]` 一段。此记录定义了指定 PLC 软键的动作。
- 设定动作编号，如其大于等于 100 则为 Run MyScreens 调用。
- 在文件 `easyscreen.ini` 中必须创建一个段用于定义需要执行的动作，其名称由操作区域名称和对话框名称组成（参见 `[keyconfiguration]` 下的记录 → `Area:=...`, `Dialog:=...`） → `[<Area>_<Dialog>]` → 例如 `[AreaParameter_SIPaDialog]`
- 在此段中定义动作编号（已在 `systemconfiguration.ini` 中设定 → 参见 `Action:=...`）。此处涉及两个指令：
 1. `LS("Sofkeyleiste1","param.com") ... 装载软键栏`
 2. `LM("Maske1","param.com") ... 装载屏幕`

通过 PLC 软键选择软键栏

在 Run MyScreens 中可通过 PLC 软键选择 Run MyScreens 软键栏和 Run MyScreens 对话框。为此在设计相关 PLC 软键时候，必须为属性“action”设定大于等于 100 的赋值。

PLC 软键的设计在文件 `systemconfiguration.ini` 中的 `[keyconfiguration]` 段中进行：

```
[keyconfiguration]

KEY75.1 = Area:=area, Dialog:=dialog, Screen:=screen, Action:=
100,
```

9.2 通过 PLC 硬键选择对话框

对于相应 PLC 软键，需要执行的 LM 和 LS 指令在文件 `easyscreen.ini` 中的段中进行，段的名称按照以下规定创建。

[areaname_dialogname]	名称的第一部分“areaname”指示操作区域，第二部分“dialogname”指示对话框，段中设计的指令适用于此对话框。
<pre>[AreaParameter_SlPaDialog] 100.screen1 = LS("Softkey1","param.com") 101.screen3 = LM("Maskel","param.com")</pre>	<p>必须使用在文件 <code>systemconfiguration.ini</code> 中为操作区域和对话框设定的名称。对话框的设定是可选项。特别是对于通过单个对话框执行的操作区域，可省略此设定，见旁边的例子。</p> <p>如果在通过对话框 <code>SlPaDialog</code> 执行的操作区域 <code>AreaParameter</code> 中显示“screen1”，则在出现值为 100 的“action”时执行指令 <code>"LS("Softkey1","param.com")"</code>。</p>
action.screen=指令	<p>通过两个属性“action”和“screen”将执行指定指令的时间设定为唯一值。</p> <p>“screen”的设定是可选项。</p> <p>允许的指令为：</p> <pre>LM (LoadMask) LS (LoadSoftkeys)</pre>

9.2 通过 PLC 硬键选择对话框

使用范围

PLC 可以启动操作软件中的以下功能：

- 选择操作区域，
- 选择操作区内的某个屏幕
- 执行软键上标注的功能

硬键

所有按键，包括 PLC 按键在下文都称为“硬键”。可以最多定义 254 个硬键，具体可以分为：

按键号	使用
按键 1 - 9	操作面板上的按键
按键 10 - 49	保留
按键 50 - 254	PLC 按键：
按键 50 - 81	预留给 OEM 使用
按键 255	由控制信息预定义。

硬键 1 - 9 的预定义如下：

按键标识	动作/作用
HK1 位置	选择操作区“加工”，回到上一个对话框
HK2 程序	选择操作区“程序”，回到上一个对话框或上一个程序
HK3 Offset	选择操作区“参数”，回到上一个对话框
HK4 程序管理器	没有功能
HK5 Alarm	选择操作区“诊断”，选择对话框“报警列表”

配置

按键的定义在配置文件“systemconfiguration.ini”的章节[keyconfiguration]中进行。每一行定义了一个“硬键事件”。“硬键事件”指第几次按下某个硬键。比方说第二次按下某个硬键和第三次按下该硬键产生的动作就不一样。

配置文件“systemconfiguration.ini”中的条目可以载入用户自定义的设置，可以使用目录 [系统用户目录]/cfg 和 [系统 oem 目录]/cfg。

配置文件中用于定义硬键事件的行的结构为：

```
KEYx.n = Area:=area, Dialog:=dialog, Screen:=screen,
Forms:=form, Menus:=menu,
Action:=menu.action, Cmdline:=cmdline
KEYx.n = Area:=area, Dialog:=dialog, Cmdline:=cmdline, Action:=
action
```

x: 硬键的编号，值范围：1 - 254

n: 事件号 - 相当于第几次按下硬键, 值范围: 0 - 9

前提条件

PLC 用户程序必须首先符合以下前提条件:

每次总是只执行一个硬键功能。因此只有当操作软件应答了前面的一个请求后, 才可以处理新的请求。当 PLC 用户程序将 MCP 按键转成硬键时, 必须有足够大的中间缓存, 这样即使在快速操作中也不会丢失某个按键事件。

PLC 接口

在 PLC 接口上对用于选择硬键的范围进行了规定。该范围定义在 DB19.DBB10 中。在该范围中, PLC 可以直接指定 50 到 254 之间的按键值。

操作软件分两步应答硬键事件。这样操作软件便可以将两个连续的不同按键代码看成两个单独的事件进行处理。在第一步中, 控制信息 255 被写入到字节 DB19.DBB10 中。借助该虚拟的按键操作可以明确地看出 PLC 的每个按键序列。控制信息对于 PLC 程序没有作用, 不允许被修改。在第二步中, 操作软件删除 DB19.DBB10, 真正地向 PLC 发出应答。从此刻起, PLC 用户程序便成功指定了一个新硬键。与此同时, 操作软件开始处理新的硬键请求。

示例

配置文件:

```
; PLC 硬键 (KEY50-KEY254)
```

```
[keyconfiguration]
```

```
KEY50.0 = name := AreaMachine, dialog := SlMachine
```

```
KEY51.0 = name := AreaParameter, dialog := SlParameter
```

```
KEY52.0 = name := AreaProgramEdit, dialog := SlProgramEdit
```

```
KEY53.0 = name := AreaProgramManager, dialog := SlPmDialog
```

```
KEY54.0 = name := AreaDiagnosis, dialog := SlDgDialog
```

```
KEY55.0 = name := AreaStartup, dialog := SlSuDialog
```

```
KEY56.0 = name := Custom, dialog := SlEsCustomDialog
```

区域和对话框名称请从 *[系统西门子目录]*cfg 目录下的 systemconfiguration.ini 文件中获取。

9.3 通过 NC 选择对话框

HMI Operate 中的 MMC 指令

可按下述方法使用 MMC 指令：

1. MMC 指令的定义

标准文件 `systemconfiguration.ini` 中含有以下内容。组合：

地址:=MCYCLES --> 指令:=LM

地址:=CYCLES --> 指令:=PICTURE_ON

该区别用于区分测量循环和其他循环。即：

- LM 只用于测量循环，PICTURE_ON 只用于非测量循环
- 新定义的 MMC 指令不允许称为 PICTURE_ON 和 LM

2. “Run MyScreens” 授权

所有“Run MyScreen”打开的对话框均需获得“Run MyScreens”授权，若无授权，只能使用有限数量的对话框。

使用 `test.com` (Run MyScreens) 的调用示例：

```
g0 f50
```

```
MMC ("CYCLES, PICTURE_ON, test.com, maske1", "A")
```

```
m0
```

```
MMC ("CYCLES, PICTURE_OFF", "N")
```

```
M30
```


Referenz - Listen

A.1 登入软键表

A.1.1 车床登入软键表

车床的程序操作区

HSK1	HSK2	HSK3	HSK4	HSK5	HSK6	HSK7	HSK8
编辑	钻削	车削	车削轮廓	铣削	其它	模拟	NC 选择
HSK9	HSK10	HSK11	HSK12	HSK13	HSK14	HSK15	HSK16
--	直线 圆弧 (仅限 ShopTurn 程序)	--	--	测量工件	测量刀具	OEM	--

车削

在下表中列出了车削工艺中可能的登入软键。各登入软键的分配根据不同的系统可能会有所不同。给定的 OEM 软键允许用于 Run MyScreens。

programGUIDE (G 代码) 登入软键:

	钻削	车削	车削轮廓		铣削		其它	
	HSK2	HSK3	HSK4		HSK5		HSK6	
VSK1	钻中心孔	轮廓车削	轮廓	--	端面铣削	轮廓	设置	高速设定
VSK2	钻削铰孔	凹槽	轮廓车削	--	型腔	轨迹	平面回转	平行轴
VSK3	深孔钻削	退刀槽	余料车削	--	多边形凸台	预钻削	刀具回转	--
VSK4	镗孔	螺纹	槽式车削	--	槽	型腔	--	--

A.1 登入软键表

VSK5	螺纹	切断	余料槽式车削	--	螺纹铣削	型腔余料	--	--
VSK6	OEM	--	往复车削	--	雕刻	凸台	子程序	--
VSK7	位置	OEM	余料往复车削	OEM	OEM	凸台余料	--	OEM
VSK8	重复位置	--	>>	<<	轮廓铣削	<<	>>	<<

ShopTurn 登入软键:

	钻削	车削	车削轮廓		铣削		其它		
	HSK2	HSK3	HSK4		HSK5		HSK6		HSK10
VSK1	中心钻孔	轮廓车削	新建轮廓	--	端面铣削	新建轮廓	设置	高速设定	刀具
VSK2	钻中心孔	凹槽	轮廓车削	--	型腔	轨迹	平面回转	平行轴	直线
VSK3	钻削铰孔	退刀槽	余料车削	--	多边形凸台	预钻削	刀具回转	重复程序	圆心
VSK4	深孔钻削	螺纹	槽式车削	--	槽	型腔	副主轴	--	圆半径
VSK5	螺纹	切断	余料槽式车削	--	螺纹铣削	型腔余料	转换	--	极坐标
VSK6	OEM	--	往复车削	--	雕刻	凸台	子程序	--	逼近/返回
VSK7	位置	OEM	余料往复车削	OEM	OEM	凸台余料	--	OEM	--
VSK8	重复位置	--	>>	<<	轮廓铣削	<<	>>	<<	--

参见

定义登入软键 (页 30)

A.1.2 铣床登入软键表

铣床的程序操作区

HSK1	HSK2	HSK3	HSK4	HSK5	HSK6	HSK7	HSK8
编辑	钻削	铣削	轮廓铣削	车削 (仅限 G 代码程序)	其它	模拟	NC 选择
HSK9	HSK10	HSK11	HSK12	HSK13	HSK14	HSK15	HSK16
--	直线 圆弧 (仅限 ShopMill 程序)	--	--	测量工件	测量刀具	OEM	--

铣削

在下表中列出了铣削工艺中可能的登入软键。各登入软键的分配根据不同的系统可能会有所不同。给定的 OEM 软键允许用于 Run MyScreens。

programGUIDE (G 代码) 登入软键:

	钻削	铣削	轮廓铣削	车削	其它
	HSK2	HSK3	HSK4	HSK5	HSK6
VSK1	钻中心孔	端面铣削	轮廓	--	轮廓车削 轮廓 设置 --
VSK2	钻削铰孔	型腔	轨迹	--	凹槽 轮廓车削 平面回转 平行轴
VSK3	深孔钻削	多边形凸台	预钻削	--	退刀槽 余料车削 刀具回转 --
VSK4	镗孔	槽	型腔	--	螺纹 槽式车削 高速设定 --
VSK5	螺纹	螺纹铣削	型腔余料	--	切断 余料槽式车削 -- --
VSK6	OEM	雕刻	凸台	--	-- 往复车削 子程序 --
VSK7	位置	OEM	凸台余料	OEM	OEM 余料往复车削 -- OEM
VSK8	重复位置	--	>>	<<	车削轮廓 << >> <<

A.3 颜色表

ShopMill 登入软键:

	钻削	铣削	轮廓铣削		其它		直线圆弧
	HSK2	HSK3	HSK4		HSK6		HSK10
VSK1	钻中心孔	端面铣削	新建轮廓	--	设置	--	刀具
VSK2	钻削铰孔	型腔	轨迹	--	平面回转	平行轴	直线
VSK3	深孔钻削	多边形凸台	预钻削	--	刀具回转	重复程序	圆心
VSK4	镗孔	槽	型腔	--	高速设定	--	圆半径
VSK5	螺纹	螺纹铣削	型腔余料	--	转换	--	螺旋线
VSK6	OEM	雕刻	凸台	--	子程序	--	极坐标
VSK7	位置	OEM	凸台余料	OEM	--	OEM	--
VSK8	重复位置	--	>>	<<	>>	<<	--

A.2 访问等级列表

不同的访问等级有以下不同的含义:



















保护等级	禁用密码	区域
ac0	预留用于西门子	
ac1	密码	机床制造商
ac2	密码	维修
ac3	密码	用户
ac4	钥匙开关位置 3	程序员, 调试员
ac5	钥匙开关位置 2	合格的操作员
ac6	钥匙开关位置 1	受过培训的操作员
ac7	钥匙开关位置 0	学过相关内容的操作员

A.3 颜色表

系统颜色

提供统一的颜色表用于对话框设计（各个标准颜色的色块）。对于单元（文本、输入栏、背景等等）可以从 0 到 133 号的颜色中选择一种。

除了预定义的颜色，您也可以使用 RGB 值 (“#RRGGBB”)来给定颜色。

索引	图标	颜色	颜色描述
1		黑色	
2		桔黄色	
3		深绿色	
4		浅灰	
5		深蓝色	
6		蓝色	
7		红色	
8		棕色	
9		黄色	
10		白色	
126		黑色	当前处于焦点下的输入/输出栏的文本颜色
127		浅橙色	当前处于焦点下的输入/输出栏的背景色
128		桔黄色	聚焦系统颜色
129		浅灰	背景颜色
130		蓝色	标题颜色 (激活)
131		黑色	标题字体颜色 (激活)
132		蓝绿色	转换栏的背景色
133		淡蓝色	列表框的背景色

A.4 文件名中的语种缩写表

支持的语言

标准语言：

语言	文件名中的缩写
简体中文	chs
德语	deu
英语	eng
法语	fra
意大利语	ita
西班牙语	esp

其他语言：

语言	文件名中的缩写
繁体中文	cht
丹麦语	dan
芬兰语	fin
印度尼西亚语	ind
日语	jpn
韩语	kor
马来西亚语	msl
荷兰语	nld
波兰语	plk
葡萄牙语（巴西）	ptb
罗马尼亚语	rom
俄语	rus
瑞典语	sve
斯洛伐克语	sky
斯洛文尼亚语	slv
泰语	tha
捷克语	csy

语言	文件名中的缩写
土耳其语	trk
匈牙利语	hun
越南语	vit

A.5 可用的系统变量列表

文档

参数手册 系统变量 /PGAsI/

A.6 对话框打开方式 (属性 CB)

下表列出了在哪些条件下调用 CHANGE 方法。

属性 CB 的适用条件:

CB0

如果变量当前有一个有效值 (例如: 通过预设值或 NC/PLC 变量), 则通过显示屏幕触发 CHANGE 方法。

A.6 对话框打开方式 (属性 CB)

CB1 (默认)

通过显示屏幕无法显式触发 CHANGE 方法。如果变量有一个配置的 NC/PLC 变量，则一定会调用 CHANGE 方法。

条件				响应	
类型	系统或者用户变量	预设值	编辑 CHANGE 方法		
输入/输出 栏	是	是	是	有了配置的 NC/PLC 变量，系统至少会通过当前 NC/PLC 变量值自动调用一次 CHANGE 方法。此时，CB 的预设值没有用。	
			否		
		否	是		
			否		
	否	是	是		通过预设值调用 CHANGE 方法。
			否		不调用 CHANGE 方法。
		否	是		由于当前没有可以用来调用 CHANGE 方法的有效值，因此不调用。
			否		
转换	是	是	是	有了配置的 NC/PLC 变量，系统至少会通过当前 NC/PLC 变量值自动调用一次 CHANGE 方法。此时，CB 的预设值没有用。	
			否		
		否	是		
			否		
	否	是	是		通过预设值调用 CHANGE 方法。
			否		不调用 CHANGE 方法。
		否 (转换列表中的第一个值默认为预设值)	是		通过转换列表中的第一个值 (预设值) 来调用 CHANGE 方法。
			否		不调用 CHANGE 方法。

参见

CHANGE (页 109)

提示和技巧

B.1 一般提示

- 可视情况将 BTSS 变量用作系统变量(\$变量)。
原因：
这样可避免可能较为费时的名称分析。
示例：
最好使用“/通道/参数/R[u1,10]”替代“\$R[10]”。
- 避免循环（同类）设置，例如屏幕和变量属性 HLP。比较之前要设置的值与当前值，不相同再真正设置该值。
原因：
避免在查找与分辨率相关的辅助画面上费时。
示例：

```
DEF MyVar=(R3///,"X1",,"mm"/WR1//"$AA_IM[0]")
CHANGE(MyVar)
  IF MyVar.VAL < 100
    HLP="mypic1.png"
  ELSE
    HLP="mypic2.png"
  ENDIF
END_CHANGE
```

建议：

```
CHANGE(MyVar)
  IF MyVar.VAL < 100
    IF HLP <> "mypic1.png"
      HLP="mypic1.png"
    ENDIF
  ELSE
    IF HLP <> "mypic2.png"
      HLP="mypic2.png"
    ENDIF
  ENDIF
```

```
ENDIF  
ENDIF  
END_CHANGE
```

- 使用一个 `MRNP()` 函数替代多个连续的 `RNP()` 函数。
使用一个 `MRDOP()` 函数替代多个连续的 `RDOP()` 函数。
原因：
降低通讯负载并提升性能。
- 读取驱动参数时不要短于 `•1•` 秒的周期，时间长些更佳。
原因： 否则与驱动的通讯可能会受到很大干扰，甚至导致故障。
- 避免随之发生的与系统或用户变量的计算。为此请使用，例如寄存器(`REG[x]`)或(不可见)辅助变量。
原因： 每次赋值都会导致向所连接的系统或用户变量中写入。
- 出于对清晰、可维护和性能（显示屏幕时）的考虑，不同块中所使用的同类代码应放在一个 `SUB` 块中。这可在相应位置上使用函数 `CALL()` 进行调用。
- 通过对对话框（在 `slguiconfig.xml` 中设置）中 `CPU` 负载的查看可以确定屏幕上的哪些更改会在多大程度上影响性能。

B.2 DEBUG 的提示

- 函数 `DEBUG()` 用于进行诊断。调用函数 `DEBUG()` 时返回的字符串会写入 `easyscreen_log.txt` 文件中。函数 `DLGL()` 对向对话框输出信息也会有帮助。
屏幕开发阶段结束后，出于对性能的考虑应将该函数删除或取消。
- 日志文件 `easyscreen_log.txt` 在屏幕开发完成后始终为空。

B.3 CHANGE 方法的提示

- **CHANGE** 方法总是只持续很短的时间，特别是那些与系统或用户变量相连并且频繁变化的变量。
原因：
提升屏幕性能。
- 视情况请不要在 **CHANGE** 方法中设置 **RNP()**函数。而是最好与要读取的系统或用户变量一起创建一个不可见的变量并使用。
原因：
每次调用都会强制取消 **RNP()**函数。否则会直接存取已存在的当前值。

示例：

此处会在轴运行每次变化时通过 **RNP()**进行名称分析，以读取通道专用的机床数据：

```
DEF AXIS_POSITION_X = (R///, ""///"$AA_IM[X]")

CHANGE (AXIS_POSITION_X)
    DLGL("Axis "" << RNP("$MC_AXCONF_GEOAX_NAME_TAB[0]") <<
    "" has moved: "
    << AXIS_POSITION_X)
END_CHANGE
```

借助不可见变量保持通道专用机床数据的最新状态，每个值变化都会复制到临时变量中，例如寄存器。

该临时变量可以在轴位置变化的 **CHANGE** 方法中使用，无需每次都进行机床数据的名称分析以及相应的读访问：

```
DEF AXIS_POSITION_X = (R///, ""///"$AA_IM[X]")
DEF AXIS_NAME_X = (S///, ""/WR0/"$MC_AXCONF_GEOAX_NAME_TAB[0]")

CHANGE (AXIS_NAME_X)
    REG[0] = AXIS_NAME_X
END_CHANGE

CHANGE (AXIS_POSITION_X1)
```

B.4 DO-LOOP 循环的提示

```
DLGL("Axis "" << REG[0] << "" has moved " <<  
AXIS_POSITION_X)  
END_CHANGE
```

- 刷新率以及与频繁变化的系统或用户变量相连接的变量的 **CHANGE** 方法的执行速度会由于变量属性 **UR** 的使用而降低，例如与轴实际值耦合的变量。
原因：
由此，值变化时相应的 **CHANGE** 方法将在固定设置的光栅中执行。

B.4 DO-LOOP 循环的提示

取决于配置，由于循环会影响 **SINUMERIK Operate** 的性能，因此请您慎重使用这些循环并且视情况放弃使用循环中费时的动作。

同时应使用寄存器(**REG[]**)作为运行变量，因为一般显示变量（特别是那些使用 **BTSS** 连接的）会极其频繁地刷新或写入，这同样会影响系统性能

请注意，程序块中执行的脚本行的数量目前限制为 **10000** 行。如果达到该数量，文件 **easyscreen_log.txt** 中的相应记录会中断脚本执行。

原因：

- 防止无限循环
- “Run MyScreens” 必须保持可操作

动画元素

C.1 简介

简介

本手册介绍的是通过 X3D 浏览器进行作业，借助该浏览器可将动态和静态图形场景（动画元素） - 以下称作辅助画面 - 集成到 SINUMERIK Operate（V4.7 SP1 起）的图形界面中。

您可在上下文关联的辅助画面中显示目标运动过程和参数。这样可以提升应用的可操作性并设计相应的用户界面。

从实施理念到完成辅助画面共包含以下几步：

- 创建用于最后可显示在 X3D 浏览器中的辅助画面的图形元素和 3D 模型（参见章节 建模 (页 232)）。
- 创建场景说明文件，在该文件中图形文件的模型数据被分配给待显示的场景和动画（参见章节 场景说明文件的结构 (页 239)）。
- 将 X3D 和 XML 文件转换为 HMI 文件（参见章节 转换为 hmi 文件 (页 244)）。
- 将 X3D 浏览器以 C++ 语言格式集成到自身应用中（参见章节 执行示例 (页 246)）。
- 在 Run MyScreens 中使用（参见章节 在 Run MyScreens 中显示 (页 247)）。

文件格式一览

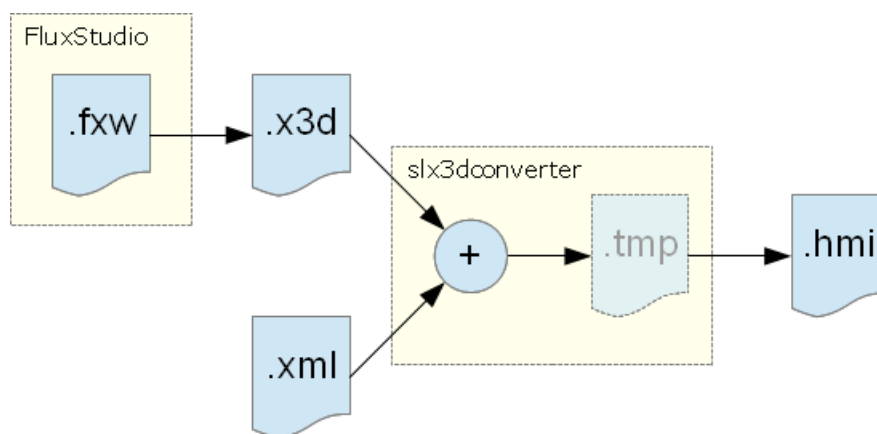


图 C-1 文件格式

文件格式	说明
.fxw	Flux Studio 的图形文件 (3D)
.x3d	交换格式的模型文件
.xml	动画和场景的定义文件
.hmi	X3D 浏览器的输出文件

C.2 建模

C.2.1 前提条件

建模的目的在于创建一个含生成的 3D 模型（文件格式为 .x3d）和一个 3D 内容官方标准的文件。

建模在 **Flux Studio Web3D Authoring Tool** 中进行。Flux Studio Web3D Authoring Tool 是一个可以提供多种导出和导入方式的免费 3D 建模工具。

前提条件

- 您已安装了 Flux Studio Web3D Authoring Tool V 2.1。
- 知道如何进行建模和使用 Flux Studio。
- 本说明中没有对文件格式 .x3d 作进一步说明，需提前作相应的了解。

说明

您可通过以下链接下载 Flux Studio Web3D Authoring Tool。

<http://mediamachines.wordpress.com/flux-player-and-flux-studio> (<http://mediamachines.wordpress.com/flux-player-and-flux-studio>)

C.2.2 建模规则

建模时请注意以下规则。之后处理辅助画面文件时需遵循这些规则。

建模规则

1. TimeSensor 的名称必须是“动画”。
2. 所有配套元素都必须被分配到一个组中。
3. 所有组都必须设为以下位置：
 $x = 0, y = 0, z = 0$
4. 如需隐藏组，则须按如下方式设置翻译值：
 $x = 1000000, y = 1000000, z = 1000000$
5. 下列元素在 Flux Studio 图形模型和 XML 场景定义（对比 一览 (页 239) 一章）中的名称必须相同：
 - 刀具
 - 摄像机
6. 如需创建一个 .x3d 文件，则须在对话框“导出选项”中进行以下设置：

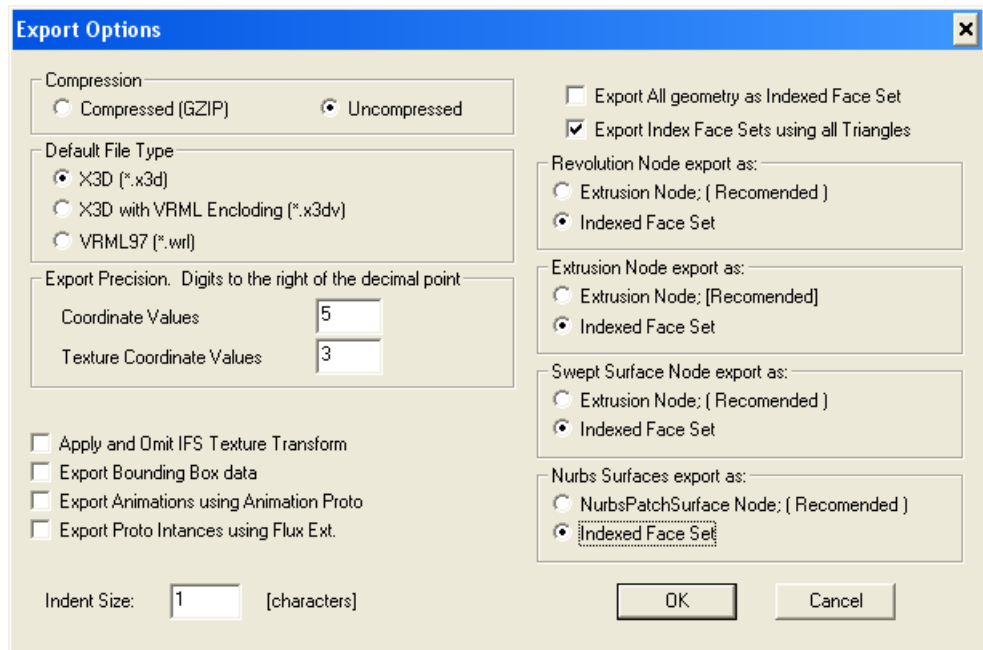


图 C-2 对话框“导出选项”中的设置

- 7. 对于文本建议采用以下设置（另见下列对话框）：
 - 文本必须始终中心对齐。
 - 文本大小应设为 **0.2**。便于文本进行精准的定位。文本输出是通过 **X3D** 浏览器进行的，与操作界面中的字体大小无关。

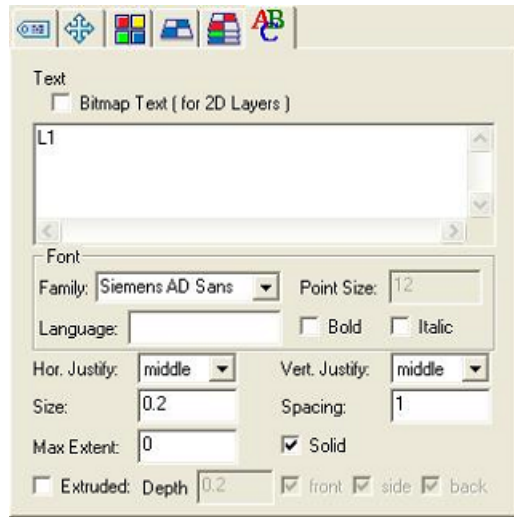


图 C-3 文本设置

8. 如需创建一个阴影线，应将文件 `schnitt.png` 用作结构。阴影线的走线方式始终是左下到右上，成 45° 度角。两个尺寸中的缩放比例都应设为 3。旋转取决于结构形式且必须手动调整。

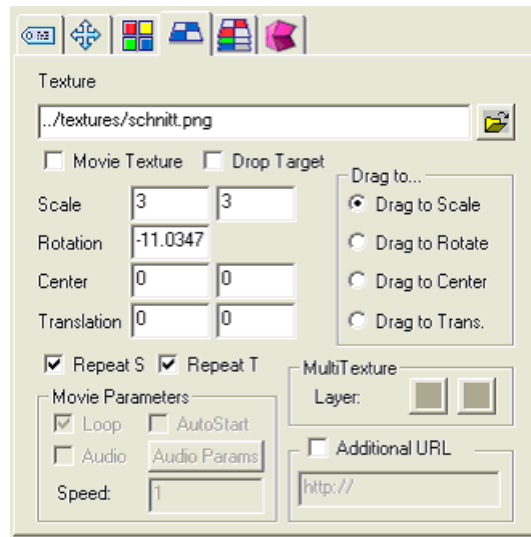


图 C-4 阴影线设置

9. 毛坯的大小/尺寸建议按如下方式设置：
- 用于车削加工的圆柱体：
长度 = 5.5；半径 = 1.4（对比模板 `turning_blank.fxw`）
 - 用于铣削加工的长方六面体：
 $x = 4.75$ ； $y = 3$ ； $z = 3$ （对比模板 `milling_blank.fxw`）

参见

XML 指令 (页 239)

C.2.3 导入图形（模型）

在 **Flux Studio** 中可以导入外部图形（模型）。常用模型（如刀具）可作为 `.x3d` 或 `.hmi` 文件集中保存并作为完成的元素在其他辅助画面中重复使用。随附的建模模板列表请见 建模模板 (页 237) 一章。

复杂的对象也可通过其他建模工具（如 **Cinema4D**）创建和导入。

下文介绍了如何重复使用这些模型。

作为“Inline”导入

Flux Studio 提供了对象“Inline”用于导入 .x3d 文件。这样可以关联外部元素，而不增加这些模型的 3D 数据。

通过菜单项 **Create -> Create Inline** 插入对象。之后在对象属性中输入文件名。

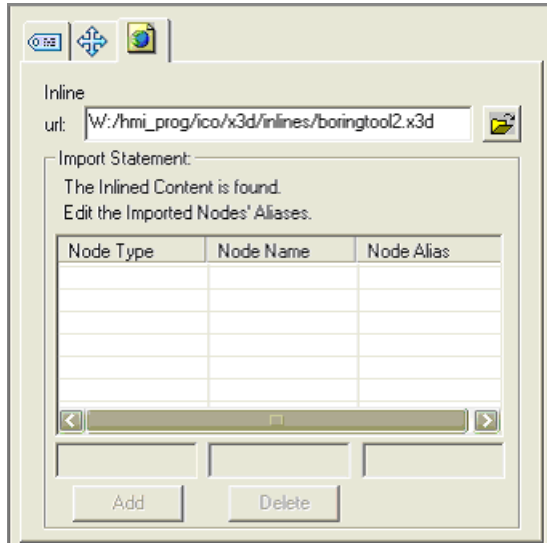


图 C-5 通过对象“Inline”导入

导入 3D 模型数据

按如下步骤导入一个文件中的模型数据。

1. 通过菜单项 **File -> Import Other Format** 打开对话框 “Flux Studio Accutrans3D Translation Utility”。
2. 通过下拉菜单 “File Types” 选择相应的格式。例如：如需导入一个 Cinema4D 文件，则须选择文件类型 “3D Studio”。
3. 然后通过 **Select Files** 选择所需的文件并开始导入。

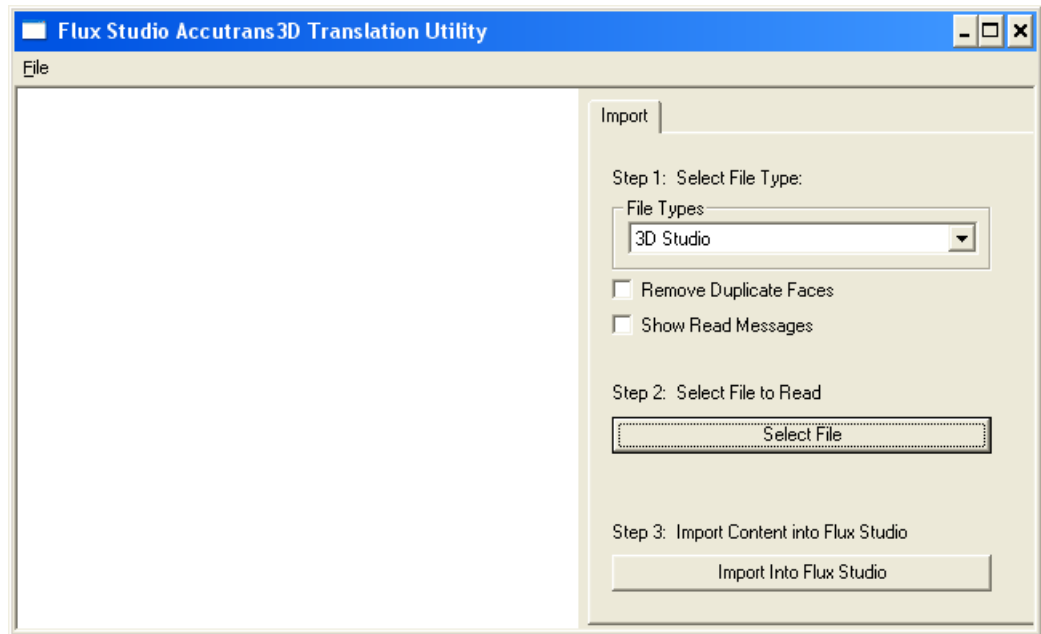


图 C-6 导入 3D 模型数据

将 Cinema4D 文件中的模型数据作为整个组插入。删除一并导入的所有摄像机和其他信息。只需要 Cinema4D 文件的图形元素。

C.2.4 建模模板

本章节中列举了一个作为外形、颜色和尺寸基础的图形元素列表。为确保操作界面中的“外观”一致，建议使用模板格式或模板创建自有图形。

概述

模板	说明
intersection_texture.png	截面结构
rapid_traverse_line_hori.fwx	用于阐明刀具运行路径的水平线（快速模式）
rapid_traverse_line_vert.fwx	用于阐明刀具运行路径的垂直线（快速模式）
feed_traverse_line_hori.fwx	用于阐明刀具路径的水平线（进给模式）
feed_traverse_line_vert.fwx	用于阐明刀具路径的垂直线（进给模式）
dimensioning_lines_hori.fwx	水平尺寸辅助线
dimensioning_lines_vert.fwx	垂直尺寸辅助线
z1_inc.fwx	水平尺寸线 Z1
x1_inc.fwx	垂直尺寸线 X1
3d_coordinate_origin.fwx	3D 坐标原点
3d_zero_point.fwx	3D 零点

车削

模板	说明
turning_blank.fwx	用于车削工艺的毛坯：一个简易圆柱体
turning_centerline.fwx	中心线
turning_centerpoint.fwx	用于显示 WCS 中零点的坐标原点
turning_refpoint.fwx	加工参考点
turning_machining_area.fwx	加工面的边界线

铣削

模板	说明
milling_blank.fwx	用于车削工艺的毛坯：一个简易长方六面体
milling_centerline.fwx	中心线
milling_refpoint.fwx	加工参考点

C.3 XML 指令

C.3.1 一览

为了使 X3D 浏览器能够访问图形，需要场景说明文件（结尾 .xml）。在场景说明文件中分配从配置中调用的场景名称及图形所在的 .x3d 文件 (TimeSensor) 中的时间。

C.3.2 场景说明文件的结构

下文显示了场景说明文件的结构并对各个 XML 指令进行了说明：

结构和说明

<!-- 在车削加工时处理平面相关的文本 -->

```
<TextPlane plane="G18_ZXY" />
```

说明

```
<TextPlane plane="G18_ZXY" />
```

在车削加工时处理平面相关的文本（轴名称）。如果使用一个标记为“Z”的文本（G18 第一根几何轴），相应的控制系统几何轴名称则会显示在辅助画面中（例如：“Z”）。

<!--文本对齐 -->

```
<TextPosition center='true' />
```

说明

```
<TextPosition center='true' />
```

文本对中的标记。这有助于在旋转画面中显示文本。推荐使用此项设置。

<!--调整刀具速度 -->

```
<ToolSpeedFactors planeSpeedFactor="0.4" rapidSpeedFactor="0.7"
reducedSpeedFactor="1.0"/>
```

说明

<ToolSpeedFactors	如需调整刀具速度，可添加该项。
planeSpeedFactor="0.4"	进给速度系数 (0.4 = 40%)
rapidSpeedFactor="0.7"	快速速度系数 (0.7 = 70%)
reducedSpeedFactor="0.1" />	第三方速度系数 (0.1 = 10%)

<!--动画定义-->

```
<SceneKey name='BoringAnimation' masterRotationSpeed="-64.0"
maxRotAngle="45.0" beginTime='0.110' endTime='0.118' view='camiso'
speedMaster='boringtool' Type="VIEW_3D_TURN_CYL">
```

说明

<SceneKey name='BoringAnimation'	动画名称
masterRotationSpeed="-64.0"	刀具的旋转方向和旋转速度。 该设置为可选项。
maxRotAngle="45.0"	避免混淆效应（刀具似乎按错误的方向旋转）。该值通常是四分之一的刀具对称。 该设置为可选项。
beginTime='0.110'	Timesensor 上的动画起始时间点
endTime='0.118'	Timesensor 上的动画结束时间点
view='camiso'	用于动画的摄像机
speedMaster='boringtool'	动画刀具名称
type="VIEW_3D_TURN_CYL">	视图类型确定视图（参见章节 视图类型 (页 243)）

<!-- 元素 -->

```

<Element name='1' time='0.110' feed='rapid' dwell='2' />
<Element name='2' time='0.112' feed='rapid' />
<Element name='3' time='0.114' feed='rapid' />
<Element name='4' time='0.116' feed='plane' />
<Element name='5' time='0.118' feed='plane' />

```

说明**<Element name='1'**

表示针对的是第一个动画元素。

time='0.110'**Timesensor** 上第一个元素的时间点**feed='rapid'**

元素的运行速度

plane = 加工进给**rapid** = 快速**reduced** = 减速，速度低于“plane”**dwell='2' />**

动画运行暂停 旋转的刀具保持旋转状态。

<!-- 动画定义结束 -->

```

</SceneKey>

```

说明**</SceneKey>**

动画定义结束

<!-- 现有动画作为新动画的来源 -->

```

<SceneKey source='BoringAnimation' name='BoringAnimationRear' mirror='
MirrorScreenX ' />

```

说明**<SceneKey source="BoringAnimation'**

用作另一个动画来源的动画的名称。

name="BoringAnimationRear'

基于来源的新动画的名称

mirror=' MirrorScreenX ' />

按 X 轴方向镜像，在新动画中输出。

<!-- 静态场景（画面）定义（4 个示例） -->

```

<SceneKey name='Default' time='0.026' view='camiso'
type="VIEW_3D_DRILL_CUT"/>
<SceneKey name='Cut' time='0.046' view='camside' type="VIEW_SIDE"/>
<SceneKey name='Zlink' time='0.056' view='camside' type="VIEW_SIDE"
  highLightedGroup='dad_Zlink'/>
<SceneKey name='Zlabs' time='0.066' view='camside' type="VIEW_SIDE"
  highLightedGroup='dad_Zlabs'/>

```

说明

<code><SceneKeyname='Z1abs'</code>	画面名称
<code>time='0.066'</code>	Timesensor 上的画面时间点
<code>view='camside'</code>	用于画面的摄像机
<code>type="VIEW_SIDE"</code>	视图类型确定视图（参见章节 视图类型 (页 243)）
<code>highLightedGroup='dad_Z1abs'/></code>	需要加亮的组的名称。组在 FluxStudio 中已 被命名为“Z1abs”。前缀“dad_”由 Flux 自 动添加。

```
<!-- end of xml file -->
```

C.3.3 镜像和旋转

通过指令 **mirror** 确定画面/模型的镜像或旋转。

说明

<code>mirror="RotateScreenX"</code>	画面围绕 X 轴旋转
<code>mirror="RotateScreenY"</code>	画面围绕 Y 轴旋转
<code>mirror="RotateScreenZ"</code>	画面围绕 Z 轴旋转
<code>mirror="MirrorScreenX"</code>	画面按 X 轴方向镜像
<code>mirror="MirrorScreenY"</code>	画面按 Y 轴方向镜像
<code>mirror="MirrorScreenZ"</code>	画面按 Z 轴方向镜像

<code>mirror="RotatePieceX"</code>	模型围绕 X 轴旋转
<code>mirror="RotatePieceY"</code>	模型围绕 Y 轴旋转
<code>mirror="RotatePieceZ"</code>	模型围绕 Z 轴旋转
<code>mirror="MirrorPieceX"</code>	模型按 X 轴方向镜像
<code>mirror="MirrorPieceY"</code>	模型按 Y 轴方向镜像
<code>mirror="MirrorPieceZ"</code>	模型按 Z 轴方向镜像

所有旋转和镜像方式都可组合使用。旋转需要确定旋转角度。

示例

```
mirror="RotatePieceZ=180"
mirror="RotatePieceZ=-90 MirrorScreenX"
mirror="RotateScreenY=90"
mirror="MirrorPieceX MirrorPieceY"
mirror="MirrorPieceZ RotatePieceX=-90"
```

C.3.4 视图类型

通过视图类型确定视图。视图以适用于显示对象的经验值和规定为基础。借此确保对象的显示与操作界面坐标系（MD 52000 \$MCS_DISP_COORDINATE_SYSTEM 或 MD 52001 \$MCS_DISP_COORDINATE_SYSTEM_2）的设置相匹配。

说明

<code>"VIEW_STATIC"</code>	无需转换的直接视图
<code>"VIEW_3D_TURN_CYL"</code>	车削加工的 3D 视图（圆柱体）
<code>"VIEW_3D_MILL_CUBE"</code>	铣削加工的 3D 视图（长方六面体）
<code>"VIEW_3D_DRILL_CUT"</code>	钻削加工的 3D 视图（剖视图）
<code>"VIEW_SIDE"</code>	钻削加工的 2D 视图（剖视图）
<code>"VIEW_TOP_GEO_AX_1"</code>	第 1 根几何轴方向上的 2D 视图
<code>"VIEW_TOP_GEO_AX_2"</code>	第 2 根几何轴方向上的 2D 视图
<code>"VIEW_TOP_GEO_AX_3"</code>	第 3 根几何轴方向上的 2D 视图

C.4 转换为 hmi 文件

说明

在 HMI 启动时将含有配套 XML 文件的 X3D 文件转换为 HMI 文件。

必须为每个 X3D 文件创建一个相应（同名）的 XML 文件。因此，必须将 X3D 文件和 XML 文件保存到目录 Suchpfad des HMI\ico\x3d\turning 或 milling 中。步骤与 .ts 文件类似。

C.5 在 Create MyHMI /3GL 中显示

C.5.1 X3D 浏览器

如需在自有 OA 应用中显示目标辅助画面，则须将 X3D 浏览器小部件集成到自有 OA 应用中。

X3D 浏览器小部件提供了可在 HMI 内部显示 X3D 内容的接口。

类 SIX3dViewerWidget 用于显示图形场景。该类的定义是在全局 GUI Include 目录 \hmi_prog\gui\include 下的标题文件 slx3dviewerwidget.h 中进行的。

C.5.2 类 SIX3dViewerWidget

该类提供了一个可灵活使用的小部件，该部件可以独立显示指定模型文件中的内容，必要时还可运行动画。

类接口由构造函数、析构函数及两种图形输出控制方法组成。

提供了功能全面的接口用于直接推导 Qt 类 QWidget（例如：show(), hide()和 resize(...)，但此处不进行进一步说明，详细信息请参考 Qt 文档）。

C.5.3 公共方法

SIX3dViewerWidget (QWidget* pParent = 0)

X3D 浏览器小部件的构造函数。

参数	含义
pParent	将参数传送给 QWidget 的构造函数。

~SIX3dViewerWidget ()

X3D 浏览器小部件的析构函数。

参数	含义
-	-

C.5.4 公共槽

void viewSceneSlot (const QString& rsFileName, const QString& rsScene, const QString& rsAnimationScene, int nChannel, int nPlane, SStepTechnology nTechnology)

借助 viewSceneSlot 方法，X3D 浏览器可指定载入和显示 rsFileName 文件中的静态场景 rsScene 和动画场景 rsAnimationScene。

正确的显示方式是先显示静态场景，再显示动画场景（按固定时间循环）。

如果未指定静态场景，则立即显示动画；相应的也可能未指定动画场景。

使用通道号、平面和工艺使场景转动到正确的位置上（取决于设置的机床坐标系）。

参数	含义
rsFileName	含待显示场景的文件的名称。
rsScene	静态场景名称。
rsAnimationScene	动画场景名称。
nChannel	通道号

参数	含义
nPlane	平面
nTechnology	工艺 针对计数类型 SIStepTechnology 定义了以下常量： <ul style="list-style-type: none"> • SL_STEP_NO_TECHNOLOGY • SL_STEP_MILLING • SL_STEP_TURNING • SL_STEP_SURFACE_GRINDING • SL_STEP_CIRCULAR_GRINDING

void viewSceneSlot (const QString& rsFileName, const QString& rsScene, const QString& rsAnimationScene)

一种简化 viewSceneSlot 方法，X3D 浏览器可借助其指定载入和显示 rsFileName 文件中的静态场景 rsScene 和动画场景 rsAnimationScene。

参数	含义
rsFileName	含待显示场景的文件的名称。
rsScene	静态场景名称。
rsAnimationScene	动画场景名称。

C.5.5 库

如需在自有项目中使用 X3D 浏览器，则须为库相关性列表扩展项 'slx3dviewer.lib'。

C.5.6 执行示例

执行示例请参考 Create MyHMI/3GL 程序包中的 \examples\GUIFrameWork\SIExGuiX3D。

C.5.7 机床数据

显示 MD 9104 \$MM_ANIMATION_TIME_DELAY

至使用辅助画面中的动画的延时，单位：秒。该设置对只进行了动画处理的辅助画面无效。

该设置对 SINUMERIK Operate 的所有动画都有效。

C.5.8 使用说明

- 如果 X3D 浏览器小部件消失，动画则会中断。此时无需选中无动画场景选择。
- 考虑到性能和存储容量，应避免经常或多次实例化 X3D 浏览器小部件。此类应用场景下推荐使用（执行）X3D 浏览器小部件。
- X3D 浏览器也可在信号槽方案中使用。
- 如果出错（例如：文件未找到或场景名称不可识别），X3D 浏览器小部件则会显示一个信息区。一旦操控另一个辅助画面文件，该信息区则会再次自动消失。

C.6 在 Run MyScreens 中显示

本章节面向有经验的“Run MyScreens”开发人员。可在对应的文档中查阅相关背景知识。

除了使用 .bmp 或 .png 格式的图片外，也可通过 X3D 浏览器显示动画辅助画面。

通常使用用于辅助画面的 Run MyScreens 接口进行集成。

如需输出 .bmp 或 .png 格式的图片，则应在定义对话框时在属性中输入数据 XG0 或清空参数。如需将 X3D 辅助画面集成到对话框中，则须在属性中输入数据 XG1。

```
//M(MASK_F_DR_O1_MAIN/$85407////52,80/XG1)
```

操控单个辅助画面需要以下参数，其含义参见章节 5.3 中的公开槽：

1. 文件名
2. 静态场景（可选）
3. 动画场景（可选）
4. 工艺（可选）
5. 平面（可选）

参数按此顺序以字符串的形式排列汇总，用逗号隔开。

```
Hlp = "Dateiname,StaticScene,AnimationScene,Technologie,Ebene"
```

示例

- 默认辅助画面可通过指定变量 `Hlp` 设置。
下面的示例是从文件“`MyDlgHelp.hmi`”中输出动画“`MyAnimation`”。未指定静态场景，即不会输出静态场景。也没有指定工艺和平面数据，即使用默认值。
`Hlp = "MyDlgHelp.hmi,,MyAnimation"`
- 对于输入窗口的单个参数，可将对应变量上的属性 `hlp` 设为一个特定的辅助画面。
下面的示例是从文件“`MyDlgHelp.hmi`”中输出平面 `G17` 中的静态场景“`MyParam`”和动画“`MyAnimation`”。未指定工艺数据，即使用默认值。
`VarMyParam.hlp = "MyDlgHelp.hmi,MyParam,MyAnimation,,G17"`

词汇表

动作

所有在 → 方法中设计的： → 功能， → 计算变量， → 更改属性， ...

用户变量

由用户在 → 零件程序或者数据块中定义的变量。

数组

通过数组可以归类、保存同一数据类型的数据，从而可以通过索引存取数据。

属性

属于某个对象（→ 对话框或者 → 变量）的 → 属性。

操作树

多个相互连接的 → 对话框

块

用于 → 配置文件的装载单元

对话框

→ 操作界面的显示

- **和对话框相关的软键栏**
由一个新设计的对话框调用的软键栏。
- **和对话框无关的软键**
不由对话框调用的软键，即由第一个新对话框设计的登入软键和软键栏。

定义行

定义 → 变量和 → 软键的程序段。

编辑器

ASCII 编辑器可以通过符号在文件中输入和编辑。

属性

对象的特征（例如：→ 变量）

输入/输出栏

即 I/O 栏： 用来输入或者输出变量值。

登入软键

启动第一次新建的 → 对话框的软键。

事件

所有触发处理→方法的事件： 输入字符，按下 → 软键， ...

辅助变量

没有 → 属性并因此不显示在 → 对话框中内部计算变量。

热键

OP 010、OP 010C 和带有热键块的 SINUMERIK 键盘上的 6 个按键，按下此键直接选择一个操作区。 也可以选择 2 个其它的按键作为热键操作。

编译器

编译器自动将 → 配置文件中定义的代码转换成 → 对话框并控制其使用。

方法

如果出现相应的 → 事件，则执行编程的步骤。

参数

参数是编程语句的可改变部分，在 → 配置文件中可用另一个字 / 符号替换。

PI 服务

在 NC 上执行固定操作的功能。PI 服务可以由 PLC 和 HMI 系统调用。

PLC 硬键

PLC 硬键通过 HMI 软件的 PLC 接口提供，如同热键。由操作界面触发的功能都是可设计的。

该按键可以是机床操作面板上的按键或者是 PLC 用户程序中的 PLC 信号连接端。因此，它们也被称作“虚拟按键”。

编程支持

提供对话框以支持 → 建立零件程序，组件版本较高

配置文件

包含定义和指令，并确定 → 对话框的外观和 → 功能的文件。

反编译

由 → 编程支持 → 对话框的输入栏可以在 → 零件程序中生成数控程序代码段。反编译指的是相反的过程。它会将数控程序代码段再次转为输入栏，然后显示在初始对话框中。

模拟

模仿 → 零件程序过程，但没有实际的机床轴运动。

软键标签

软键在屏幕上的文本/图形。

软键栏

所有水平软键或者所有垂直软键

栏索引

数组栏索引

系统变量

提供零件程序存取和 HMI 存取时的 NC 状态的 NC 变量。

零件程序

规定轴的运动过程以及各种特殊动作的、以数控语言编制的程序。

转换栏

→ 在输入 / 输出栏中列明多个数值；通过转换栏检查：某栏中的输入必须与列明值中的一个相符合。

变量

内存位的标记，可以通过指定 → 属性显示在 → 对话框中，可在其中输入数据和计算结果。

行索引

数组行号码

访问等级

权限分级，即根据用户的不同权限来使用操作界面上的功能。

索引

D

DLL 文件, 143

E

ELLIPSE (定义椭圆形), 182

ELLIPSE (定义圆形), 182

F

Flux Studio Web3D Authoring Tool, 232

H

H_SEPARATOR (定义水平分割线), 183

L

LINE (定义线), 181

N

NC 变量

 读取, 159

 写入, 159

P

PI 服务, 122

PLC 变量

 读取, 159

 写入, 159

R

RECT (定义矩形), 182

S

SIX3dViewerWidget, 244

V

V_SEPARATOR (定义垂直分割线), 183

帮

帮助画面, 81

帮助文件, 83

保

保护等级, 222

报

报警

 语种缩写, 224

背

背景色, 82

比

比较运算符, 106

变

变量

 CURPOS, 92

 CURVER, 93

 ENTRY, 93

 ERR, 94

 FILE_ERR, 95

 FOC, 97

 S_ALEVEL, 98

 S_CHAN, 98

 S_CONTROL, 99

 S_LANG, 100

 S_NCCODEREADONLY, 100

 S_RESX, 101

 S_RESY, 101

 参数, 77

 传输, 138

 更改属性, 83

计算, 67
检查, 127
变量类型, 77
 INTEGER, 84
 VARIANT, 86
变量值, 65
变量状态, 65

表

表格
 编程, 191
 定义, 190
 定义列, 193

不

不变色进度条, 75

操

操作树, 29

长

长文本, 78

常

常量, 105

单

单位文本, 78

导

导入
 图形（模型）, 235

登

登入软键, 29, 30

定

定义软键栏, 55

短

短文本, 78

对

对话框
 定义, 41
 多列, 50
 说明块, 42
 属性, 43
对话框单元, 48
对话框切换模式, 152

方

方法
 ACCESSLEVEL, 109
 CHANGE, 109
 CHANNEL, 111
 CONTROL, 111
 LANGUAGE, 113
 LOAD, 113
 LOAD GRID, 154
 OUTPUT, 115
 PRESS, 116
 PRESS(ENTER), 117
 PRESS(TOGGLE), 117
 RESOLUTION, 118
 RESUME, 119
 SUSPEND, 119
 UNLOAD, 114
 概述, 108

访

访问等级, 57

辅

辅助变量, 66

工

工步链支持, 165

公

公共槽, 245

功

功能

CALL (子程序调用), 125
 CLEAR_BACKGROUND), 128
 CP (复制程序), 128
 CVAR (检查变量), 127
 DEBUG, 137
 DLGL (对话框行), 136
 DO-LOOP), 176
 DP (删除程序), 129
 EP (存在程序), 130
 EVAL (评估), 142
 EXIT, 137
 EXITLS (退出装载软键), 143
 FCT, 143
 FORMAT(字符串), 175
 GC (生成代码), 145
 HMI_LOGIN, 148
 HMI_LOGOFF, 148
 HMI_SETPASSWD, 149
 INSTR (字符串), 170
 LA (装载数组), 149
 LB (装载块), 150
 LEFT(字符串), 171
 LEN (字符串), 169
 LISTADDITEM), 140
 LISTCLEAR), 140
 LISTCOUNT), 140
 LISTDELETEITEM, 140
 LISTINSERTITEM, 140
 LM (装载屏幕), 151
 LS (装载软键), 153
 MIDS(字符串), 172
 MP (移动程序), 131
 MRDOP), 122
 MRNP (多次读取 NC PLC), 156
 PI_START, 158
 RDFILE), 133
 RDLINEFILE), 133
 RDOP), 122
 REPLACE(字符串), 172
 RESIZE_VAR_IO, 160
 RESIZE_VAR_TXT, 160
 RETURN (返回), 163
 RIGHT(字符串), 171

RNP (读取 NC PLC 变量), 159
 SP (选择程序), 132
 START_TIMER), 179
 STOP_TIMER, 179
 STRCMP(字符串), 173
 STRINSERT(字符串), 173
 STRREMOVE(字符串), 174
 SWITCH), 155
 TRIMLEFT(字符串), 174
 TRIMRIGHT(字符串), 174
 UNTIL, 176
 WDOP, 122
 WHILE, 176
 WNP (写入 NC PLC 变量), 159
 WRFILE, 133
 WRLINEFILE), 133
 读写驱动参数, 122
 反编译 NC 代码, 164
 概述, 121
 忽略注释的反编译, 165
 文件访问, 133
 向后查找(SB), 168
 向前查找(SF), 168
 循环执行脚本, 179

机

机床数据, 247

极

极限值, 77

集

集成了单位选择框的输入/输出栏, 83

寄

寄存器

交换数据, 161
 值, 161
 状态, 162

聚

聚焦控制, 194

口

口令对话框, 52
口令输入模式 (星号), 76

两

两次变色的进度条, 73

列

列表栏, 72

配

配置句法 “;” 变量, 39
配置句法 “;” 表格列, 40
配置句法 “;” 扩展句法, 38
配置句法 “;” 屏幕, 38
配置句法 “;” 软键, 39
配置文件, 27

前

前景色, 82

软

软键
 分配属性, 55
 属性, 57

三

三角函数功能, 104

设

设计 PLC 软键, 213

生

生成 NC 代码, 145

输

输入模式, 80

数

数学函数, 105
数字格式, 87
数组
 比较模式, 186
 存取模式, 186
 单元, 185
 定义, 184
 栏索引, 186
 行索引, 186
 状态, 189

刷

刷新速度, 79

提

提示框, 78, 80

条

条件, 106

图

图片取代短文本, 73
图形文本, 78

位

位置
 短文本, 81, 90
 输入/输出栏, 81, 90

文

文本, 78
文件
 复制, 128
 删除, 129
 移动, 131

文件格式

fxw, 231
hmi, 231
x3d, 231
xml, 231

系

系统变量, 67, 81
系统颜色, 222

显

显示模式, 79
显示选项, 79

写

写入模式, 82

信

信号色 “; ” 进度条, 82
信号值 “; ” 进度条, 77

颜

颜色, 82

用

用户变量, 81

语

语种缩写, 224

预

预设值, 77

运

运算符
数学, 103
位, 107

在

在线帮助, 62

栅

栅格 → 表格, 190

主

主对话框, 151

属

属性, 79

转

转换符号, 80
转换栏, 71, 77, 88

子

子程序, 121
变量, 126
调用, 125
块标记, 126
中断, 163
子对话框, 151

字

字符串链, 91

自

自定义小部件
定义, 195
读取和写入属性, 200
接口, 196
库, 196
手动数据交换, 199
响应自定义小部件信号, 203
执行方法, 201
自动数据交换, 198

附件

A.1 SINUMERIK 828D 文档一览

一般文档



产品样本NC 82



广告印刷品
SINUMERIK 828D



广告印刷品
SINUMERIK 828D BASIC T 基
SINUMERIK 828D BASIC M 基
本印刷版



广告印刷品
SINUMERIK 828D

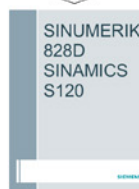
用户文档



操作手册
工艺特定



编程手册
- 基本知识
- 工作准备
- 测量循环
- ISO 车削
- ISO 铣削



诊断手册

制造商/服务文档



设备手册
调试手册
服务手册



功能手册
- Safety Integrated



功能手册
- 基本功能
- 扩展功能
- 特殊功能
- 同步动作
- ISO 术语



参数手册
- 机床数据
- 接口信号
- 变量



参数手册
参数

制造商/服务文档



系统手册
- Ctrl-Energy



配置手册
- EMC安装准则

调试

开机调试手册, 01/2015, 6FC5397-3DP40-5RA2