

SIEMENS

SINUMERIK

SINUMERIK 828D 车削和铣削

开机调试手册

前言

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

调试外设

4

机床数据参数化

5

调试驱动

6

优化驱动

7

维护管理器

8

Easy Extend

9

刀具管理

10

Easy Archive

11

附录

A

适用于:

数控软件 版本 4.5 SP2

03/2013

6FC5397-3DP40-3RA1

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 (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” (www.siemens.com/automation/service&support)，
然后点击右侧的“支持”。

SINUMERIK

SINUMERIK 的信息点击 (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献中未描述的某些功能。但是这并不意味着在提供系统时必须带有这些功能，或者为其提供有关的维修服务。

同样，因为只是概要，所以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 (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 (www.siemens.com/automation/service&support)。

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

目录

前言	3
1 供货范围和前提条件.....	11
1.1 系统概述.....	11
1.2 调试工具和服务工具	13
1.3 调试过程.....	15
1.3.1 手册编排结构.....	15
1.4 启动控制系统.....	18
1.5 建立和控制系统的通讯	21
1.5.1 通过 X130 建立和控制系统的通讯.....	21
1.5.2 通过编程工具建立和控制系统的通讯的步骤	23
1.5.3 这样通过Access MyMachine与控制系统进行通讯.....	26
1.5.4 示例： 通过 NCU Connection Wizard 建立和控制系统的通讯的步骤.....	29
2 数据级的导入与应用.....	33
2.1 NCK 中的数据级.....	34
2.2 PLC 中的数据级	37
2.3 操作软件中的数据级	38
3 操作软件的设置.....	43
3.1 存取级别.....	43
3.2 设置和修改口令的步骤	45
3.3 设置日期和时间的步骤	46
3.4 设置操作软件的语言	47
3.4.1 提供的标准系统语言和扩展语言	47
3.4.2 使用输入法编辑器输入亚洲字符。.....	47
3.4.3 输入汉字的步骤。.....	49
3.4.4 输入韩语字符的步骤.....	50
3.5 检查和输入许可证.....	52
3.5.1 输入许可证密码的步骤	52
3.5.2 检查缺少的许可证/选件的步骤.....	54
3.5.3 许可证管理的相关定义	55
3.6 定义PLC用户报警.....	57
3.6.1 用户 PLC 报警的结构	58
3.6.2 创建用户 PLC 报警的步骤.....	59

3.6.3	配置报警记录.....	61
3.6.4	配置记录选项的步骤.....	62
3.7	创建 OEM 专用在线帮助	65
3.7.1	配置文件的结构和句法	66
3.7.2	帮助手册的结构和句法	68
3.7.3	在线帮助采用的句法的描述	70
3.7.4	示例：创建 OEM 专用在线帮助手册的步骤.....	74
3.7.5	示例：创建针对用户 PLC 报警的在线帮助的步骤	77
3.7.6	示例：创建针对 NC /PLC 变量的在线帮助的步骤	80
3.7.7	示例：创建编程在线帮助的步骤	82
4	调试外设.....	85
4.1	外设模块的连接	85
4.1.1	激活外设模块.....	85
4.1.2	设置外设模块的地址.....	87
5	机床数据参数化	89
5.1	机床数据的分类	89
5.2	执行外部 CNC 系统的零件程序	92
5.3	用 Advanced Surface 加工自由形状.....	93
5.4	Advanced Surface 的机床数据	95
5.5	用于模拟量主轴的机床数据	99
5.6	管理数据.....	102
5.6.1	在控制系统内传输数据的步骤.....	103
5.6.2	保存和装载数据的步骤	104
5.6.3	比较数据的步骤	105
6	调试驱动.....	107
6.1	配置驱动.....	107
6.1.1	示例： SINAMICS S120 Combi 的配置.....	107
6.1.2	示例： 配置带 SMC 编码器的主轴的步骤	112
6.1.3	示例： 配置带 SMI 编码器的轴	118
6.1.4	示例： SINAMICS S120 书本型的配置	123
6.1.5	示例： 配置驱动系统的步骤	126
6.1.6	示例： 配置电源的步骤	131
6.1.7	示例： 配置驱动对象的步骤	136
6.1.8	示例： 配置外部编码器的步骤.....	143
6.1.9	在基本调试后更换驱动组件	146
6.1.10	示例：包含 TM120 的并联	147
6.1.11	交换过程数据.....	150
6.1.12	用于进给轴/主轴测试运行的参数	151

6.2	轴指定	153
6.2.1	示例：分配轴的步骤	153
6.2.2	示例：设置进给轴/主轴的机床数据	159
6.3	配置数据组	161
6.3.1	数据组一览	161
6.3.2	添加数据组	162
6.3.3	删除数据组	168
6.3.4	修改数据组	170
6.4	DRIVE-CliQ 拓扑规则	172
6.4.1	S120 Combi 的拓扑规则	172
6.4.2	S120 书本型的拓扑规则	174
6.4.3	SMC40 的拓扑结构规则	178
6.5	端子布局	179
6.5.1	X122 和 X132 的引脚布局	179
6.5.2	X242 和 X252 的端子布局	180
6.5.3	示例：CU 和电源接触器的连接	182
6.5.4	连接测量头	184
7	优化驱动	187
7.1	引言	187
7.2	自动伺服优化	189
7.2.1	目标	189
7.2.2	设置优化方案	192
7.2.3	启动自动伺服优化的步骤	196
7.2.4	插补轴组的优化方式	199
7.3	优化轴急动度	202
7.3.1	检查轴急动度	202
7.3.2	轴急动度的测试程序	203
7.3.3	选择跟踪信号	204
7.3.4	优化急动度设置	207
7.4	转矩使用率	210
7.5	圆度测试	214
7.5.1	圆度测试：功能	214
7.5.2	圆度测试：执行测量	215
7.5.3	圆度测试：示例	217
7.5.4	圆度测试：保存数据	220
7.6	优化主轴	223
7.6.1	设置主轴的机床数据	223
7.6.2	主轴：检查转速控制器的方式	225
7.6.3	主轴：检查位置控制器的方式	231

8	维护管理器	235
8.1	PLC 用户程序中的接口	237
8.2	操作软件接口.....	244
8.3	导入和导出维护任务的步骤	247
8.4	应答维护任务.....	250
9	Easy Extend	251
9.1	功能概述.....	251
9.2	PLC 用户程序中的配置	253
9.3	机床制造商和经销商的选件位.....	255
9.4	操作界面上的显示画面	258
9.5	建立与语言有关的文本	259
9.6	示例	261
9.6.1	控件示例.....	261
9.6.2	调试支持参数示例	262
9.6.3	功率部件应用示例	264
9.7	脚本语言的说明	267
9.7.1	特殊字符和运算符	268
9.7.2	XML 脚本的结构.....	269
9.7.3	CONTROL_RESET	270
9.7.4	DATA	271
9.7.5	DATA_ACCESS	271
9.7.6	DATA_LIST	272
9.7.7	DRIVE_VERSION	273
9.7.8	FILE	273
9.7.9	FUNCTION	275
9.7.10	FUNCTION_BODY.....	276
9.7.11	INCLUDE	277
9.7.12	LET	278
9.7.13	MSGBOX	279
9.7.14	OP	280
9.7.15	OPTION_MD	281
9.7.16	PASSWORD	282
9.7.17	PLC_INTERFACE	282
9.7.18	POWER_OFF.....	283
9.7.19	PRINT	283
9.7.20	WAITING.....	284
9.7.21	?up	285
9.7.22	用于窗口的 XML 标签.....	285
9.7.23	BOX	288

9.7.24	CONTROL.....	288
9.7.25	IMG	291
9.7.26	PROPERTY	292
9.7.27	REQUEST.....	293
9.7.28	SOFTKEY_OK, SOFTKEY_CANCEL.....	294
9.7.29	TEXT	295
9.7.30	TYPE_CAST	295
9.7.31	UPDATE_CONTROLS	297
9.7.32	参数的编址	297
9.7.33	驱动对象的编址	300
9.7.34	用于指令的 XML 标签.....	302
9.8	字符串函数	305
9.8.1	string.cmp.....	306
9.8.2	string.icmp.....	307
9.8.3	string.left.....	308
9.8.4	string.right	309
9.8.5	string.middle	310
9.8.6	string.length.....	311
9.8.7	string.replace.....	312
9.8.8	string.remove	313
9.8.9	string.delete.....	314
9.8.10	string.insert.....	315
9.8.11	string.find.....	316
9.8.12	string.reversefind.....	317
9.8.13	string.trimleft.....	318
9.8.14	string.trimright	319
9.9	三角函数.....	320
10	刀具管理.....	323
10.1	基础部分.....	323
10.1.1	刀具管理的结构.....	324
10.1.2	刀具管理的组件.....	325
10.1.3	手动装卸刀具.....	328
10.2	NC/PLC 用户接口.....	329
10.2.1	换刀、卸刀、装刀、刀库定位.....	330
10.2.2	换刀.....	337
10.2.3	传输步骤表和应答步骤表.....	345
10.3	刀具管理的机床数据.....	348
10.3.1	刀具管理设置.....	348
10.3.2	MD20360 与SD54215 的关联.....	353
10.3.3	JOG模式下刀具测量.....	356
10.4	PLC 程序块	358

10.4.1	应答过程.....	358
10.4.2	应答类型.....	359
10.4.3	应答状态.....	360
10.4.4	配置步骤表.....	365
10.4.5	配置应答步骤.....	368
10.4.6	调整 PLC 用户程序.....	369
10.4.7	刀库刀位信息.....	371
10.4.8	PI 服务: TMMVTL.....	374
10.5	示例: 装刀/卸刀.....	376
10.6	示例: 更换手动刀具.....	378
10.7	车床应用示例.....	382
10.7.1	示例: 车床, 带转塔刀库 (MAG_CONF_MPF)	382
10.7.2	示例: 应答步骤 (车床)	388
10.7.3	示例: 车床的换刀循环	389
10.7.4	示例: 车床, 带副主轴	390
10.7.5	示例: 空缓冲区测试.....	391
10.7.6	示例: 将刀具从缓冲区传输到刀库	392
10.7.7	示例: 重复“准备换刀”任务	393
10.8	铣床应用示例.....	394
10.8.1	示例: 铣床, 带链式刀库和双夹爪(MAG_CONF_MPF).....	394
10.8.2	流程图: 换刀.....	400
10.8.3	示例: 应答步骤 (铣床)	410
10.8.4	示例: 铣床的换刀循环	412
11	Easy Archive	413
11.1	备份和存档数据.....	413
11.2	通过存档来备份数据.....	414
11.3	创建调试存档的步骤.....	417
11.4	导入调试存档的步骤.....	419
11.5	仅保存变更的机床数据	420
11.6	示例: 数据文档“Easy Archive” (Use case)	421
11.7	设置串行接口参数	423
A	附录.....	425
A.1	文件名中的语种缩写表	425
A.2	缩略语列表	426
A.3	SINUMERIK 828D 文档一览	429
	词汇表	431
	索引.....	439

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:

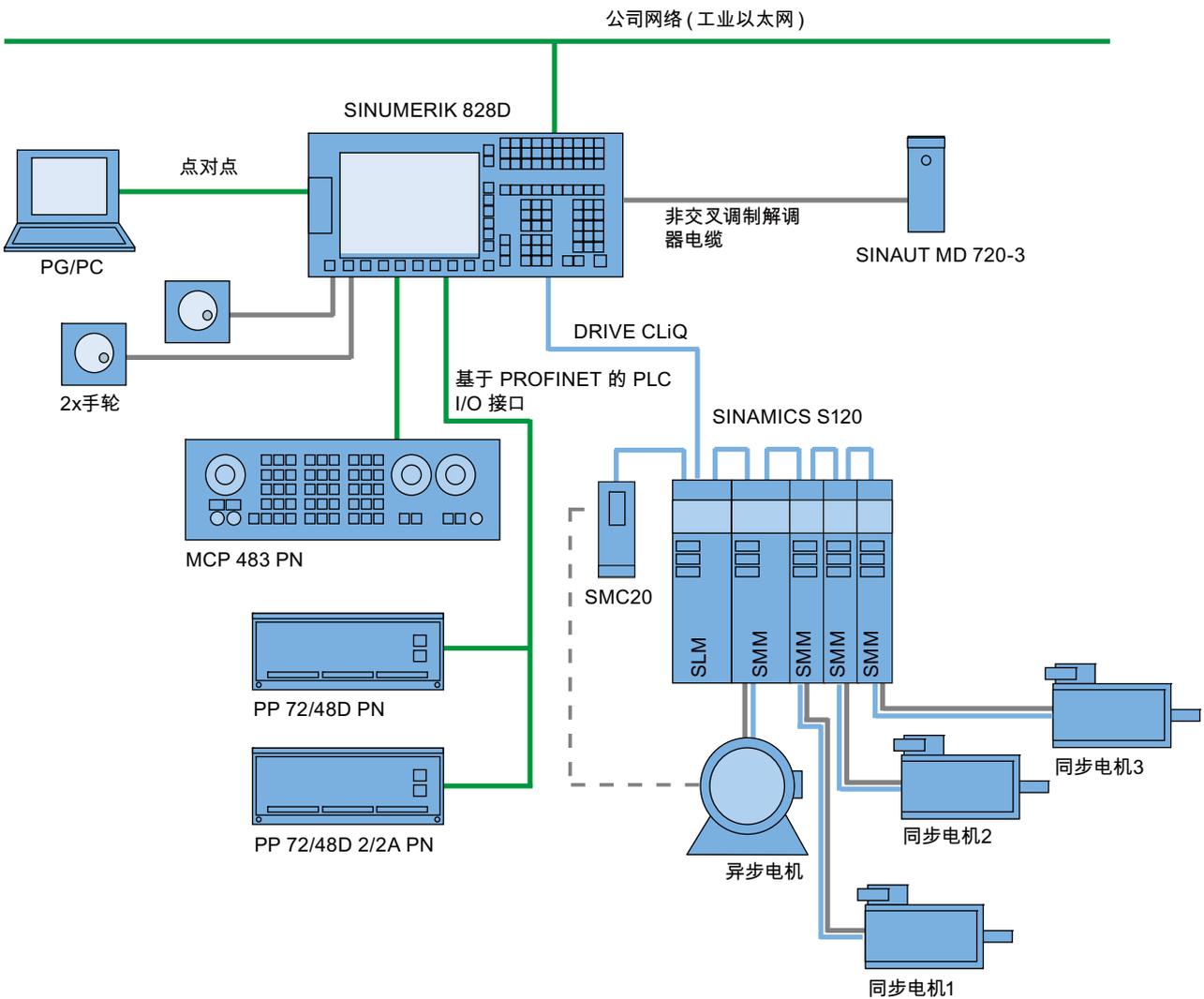


图 1-1 配置示例 1: 4 轴的基本扩展

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：

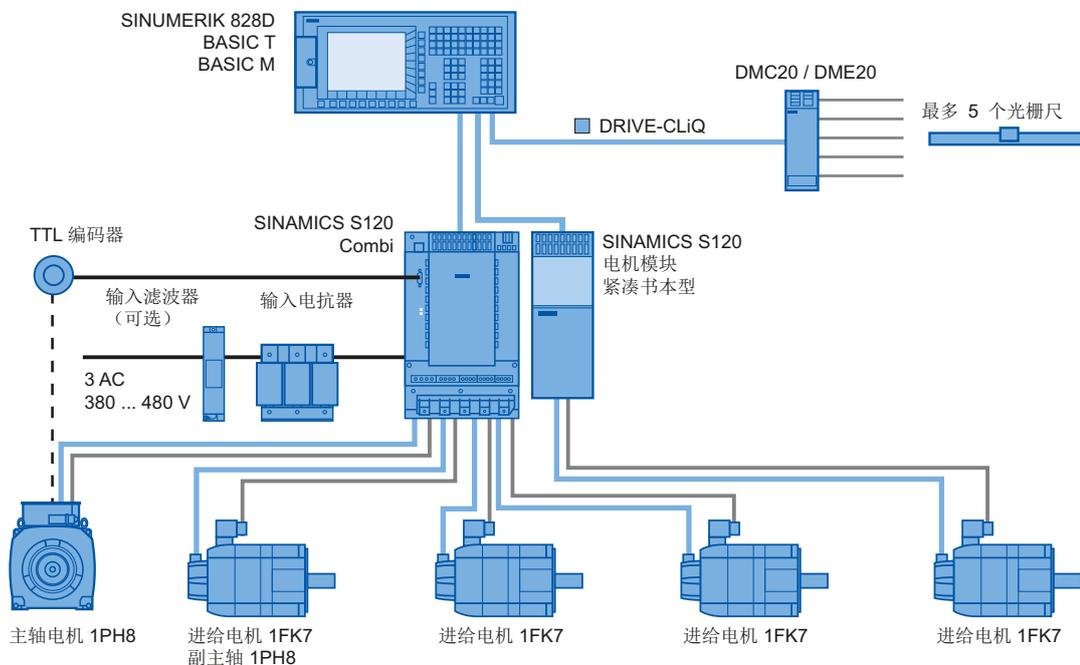


图 1-2 配置示例 2： SINUMERIK 828D BASIC T/M 配备 S120 Combi

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：

- 用于集成 PLC 的编程工具“PLC Programming Tool”
- 用于 SINAMICS S120 的调试软件
- PLC 程序库（示例）

用于集成 PLC 的编程工具“PLC Programming Tool”

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”。

用于 SINAMICS S120 的调试软件

在通过操作界面完全显示 SINAMICS S120 调试功能前，您可以采用 SINAMICS S120 调试软件开展配置并优化配置。PC 通过 SINUMERIK 828D 正面的以太网接口接入。

说明

订货数据

以下工具的订货数据请参见产品样本 NC 82 和 NC 62。

SINUMERIK Integrate Access MyMachine /P2P

Access MyMachine /P2P (以前的 RCS Commander)是一种调试工具，调试人员利用它可以非常方便地通过拖放在 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”。

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，例如：驱动的优化工作，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

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。

一览：

<B060> 系统概述	调试手册 “车床版和铣床版” (页 11)
<B002> 调试工具和服务工具	调试手册 “车床版和铣床版” (页 13)
基础知识：	
<B061> 菜单结构	测量循环编程手册
<B021> 访问等级	调试手册 “车床版和铣床版” (页 43)
<B021> 日期和时间	调试手册 “车床版和铣床版” (页 46)
<B041> 设置机床数据	调试手册 “车床版和铣床版” (页 89)
<B032> SINUMERIK Integrate Access MyMachine	在线帮助
<B044> 检查和输入授权	调试手册 “车床版和铣床版” (页 52)
<B045> 网络配置 → 设置驱动器	操作手册车床版 操作手册铣床版
<B007> 电子日志	调试手册 “SINUMERIK Operate”
<B017> 数据管理	调试手册 “车床版和铣床版” <ul style="list-style-type: none"> • 数据级的导入与应用 (页 33) • Easy Archive (页 413)
<B006> 电气柜设计	设计手册 “电磁安装指南”

1.3 调试过程

建立连接和诊断:

<B010> PPU 连接的诊断

设备手册“PPU”

- 系统概述
- 数字量输入/输出

<B030> 数字量输入/输出 → 端子布局
手轮和探头

调试手册
“车床版和铣床版” (页 179)

<B051> 电源模块、电机模块

SINAMICS S120 设备手册

<B053> 诊断

调试手册 “SINUMERIK Operate”

- 驱动系统诊断
- 诊断与服务 → 轴

<B008> PROFINET 连接的诊断:

设备手册“PPU”

- MCP 310C PN / MCP 483C PN
- PP 72/48D PN/PP 72 / 48D 2/2A PN

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 用户报警

功能手册“基本功能”(P4)

<B058> PLC 板载诊断

操作手册车床版

操作手册铣床版

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”

调试过程:

<B068> 调试过程 → 调试驱动

调试手册
“车床版和铣床版” (页 107)

<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
“车床版和铣床版” (页 187)

<B102> 刀具管理	调试手册 “车床版和铣床版” (页 323)
<B009> Easy Extend	调试手册 “车床版和铣床版” (页 251)
<B011> SINUMERIK Integrate Run MyScreens	编程手册 ”SINUMERIK Integrate Run MyScreens” (BE2)
<B020> 维护管理器	调试手册 “车床版和铣床版” (页 235)
<B018> Easy Message	操作手册车床版 操作手册铣床版
<B016> 安装附加轴	---
<B029> 模拟量主轴	调试手册 “车床版和铣床版” (页 99)

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 SINUMERIK Operate 中各个循环的调试细节请参考操作软件和基本软件的调试手册, SINUMERIK Operate (IM9), 章节“配置循环”。

1.4 启动控制系统

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:

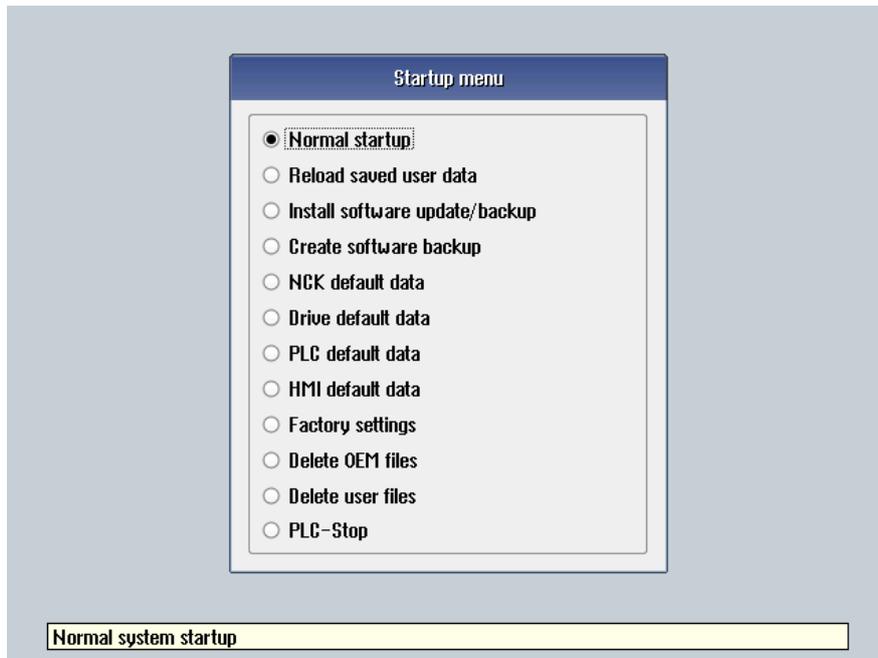


2. 在 3 秒钟内, 按下<SELECT>键。

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. “Setup 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)。

选择	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。
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	系统删除目录“/user”下的用户数据。
Factory settings	<p>有两个方案： No [方案 1]/ Yes [方案 2]</p> <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。
Delete OEM files	系统删除“/oem”和“/addon”中的所有数据： 制造商存档、OEM 报警文本、Easy Screen 应用程序。
Delete user files	
PLC stop	系统停止 PLC。

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。

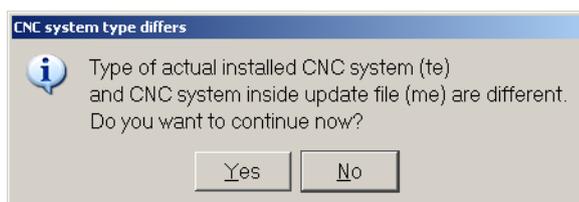
此时，如果在此之前已使用软键“数据存储”手动备份了数据，系统会载入备份数据。如果没有找到备份数据，系统会自动以“NCK default data”的模式启动。

1.4 启动控制系统

硬件检测

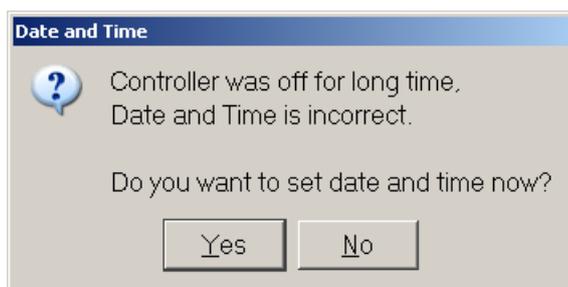
PPU 24x.2 硬件分为车削型 BASIC T 和铣削型 BASIC M :

系统在启动时会根据硬件信息块中找到的 MLFB 检测 PPU 硬件型号，并将结果和当前的 NCK 性能等级比较。如果系统发现有和该硬件不配套的软件，会发出以下报告：

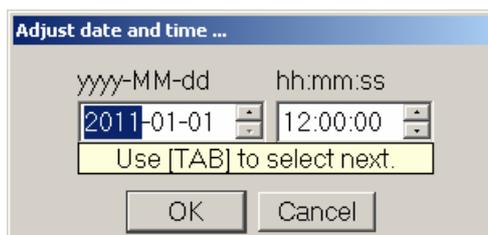


空的 RTC 电容

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。

1.5 建立和控制系统的通讯

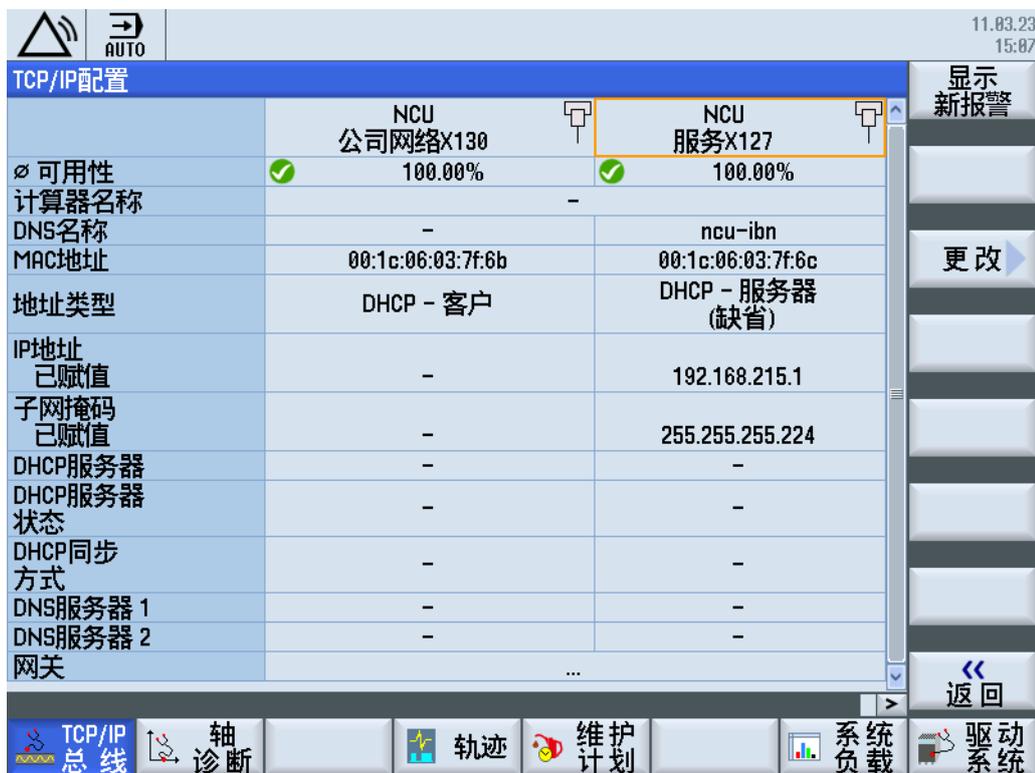


图 1-3 网络设置

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 编程工具中设置通讯接口

在 PLC 编程工具中建立网络连接的步骤:

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。

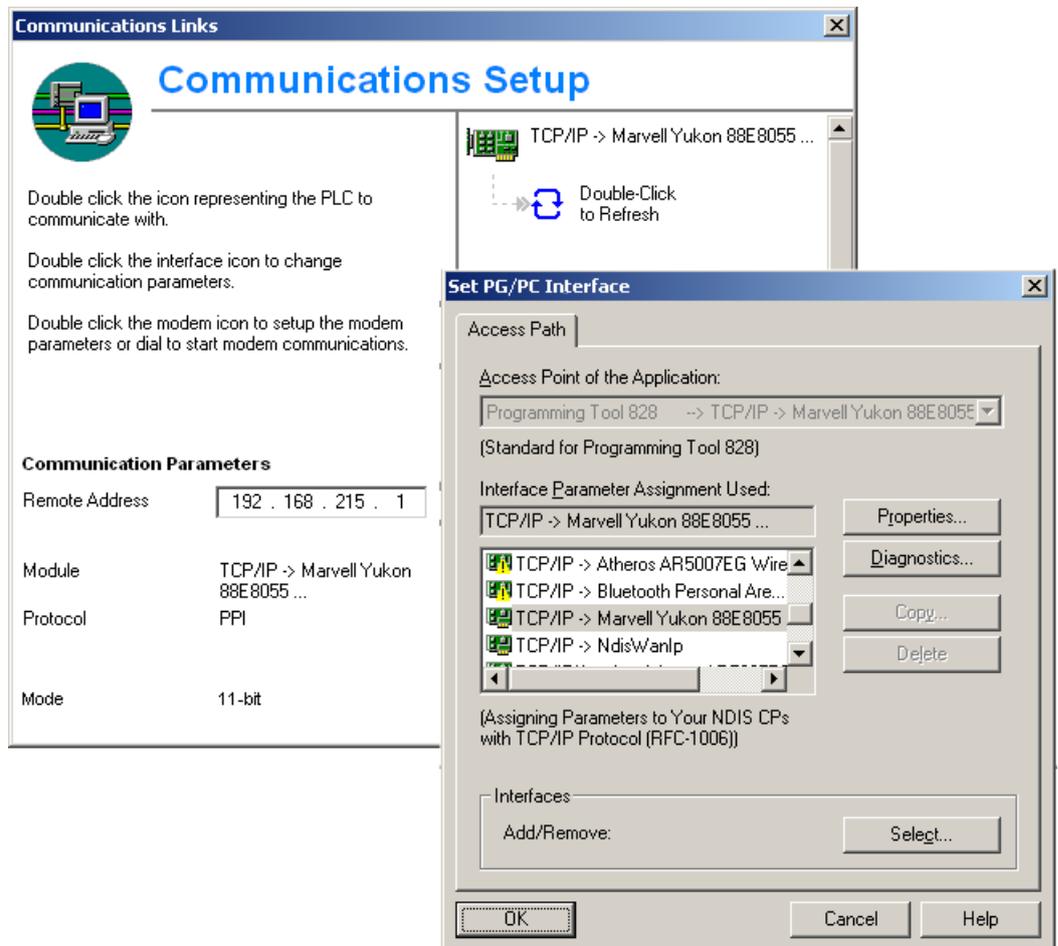


图 1-4 TCP/IP 通讯设置

1.5 建立和控制系统的通讯

- 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：

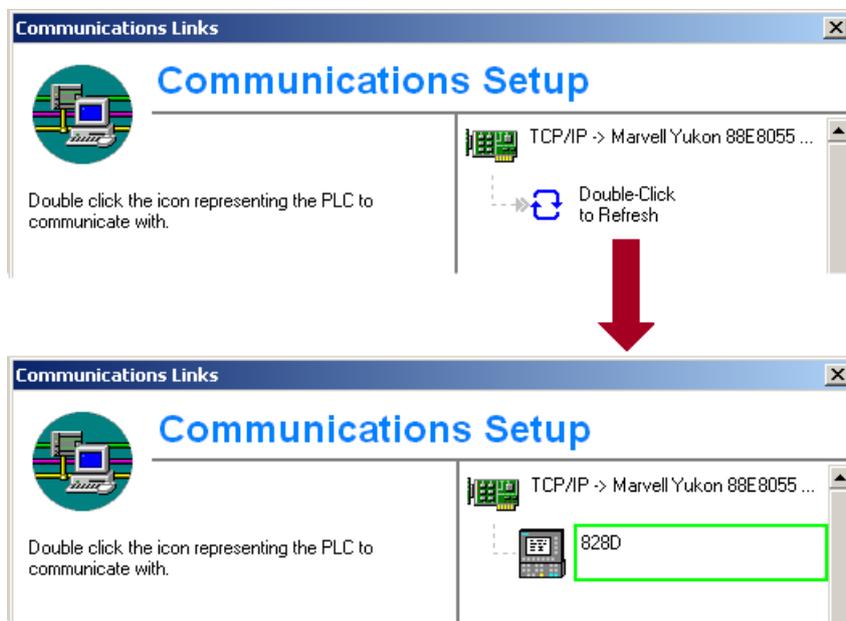


图 1-5 在线连接

8. 如果没有成功建立连接，可能需要取消以下设置：

选择“设置” → “网络连接” → “本地连接” → “属性” → “高级” → “Windows 防火墙” → “设置” → “高级”：取消勾选选项“本地连接”。

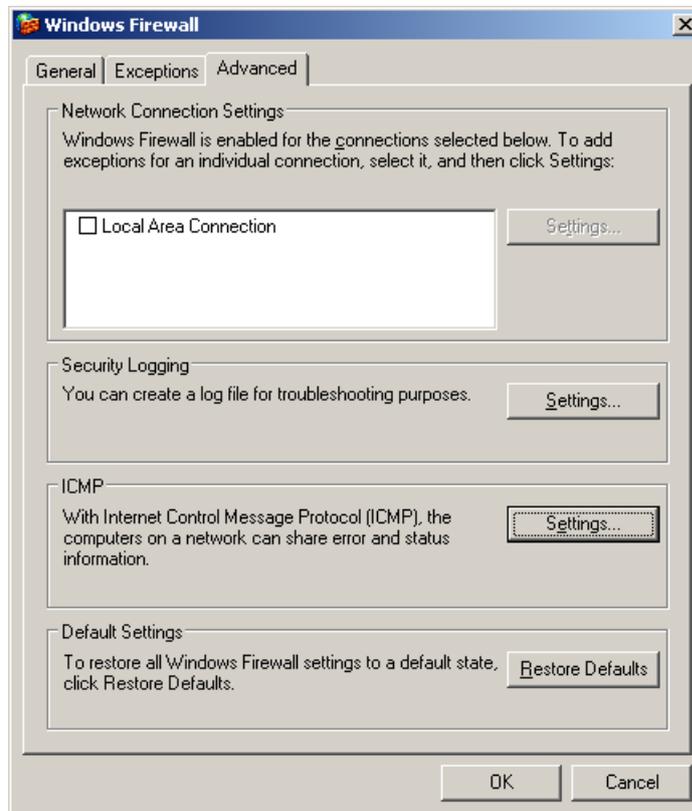


图 1-6 取消勾选选项

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：

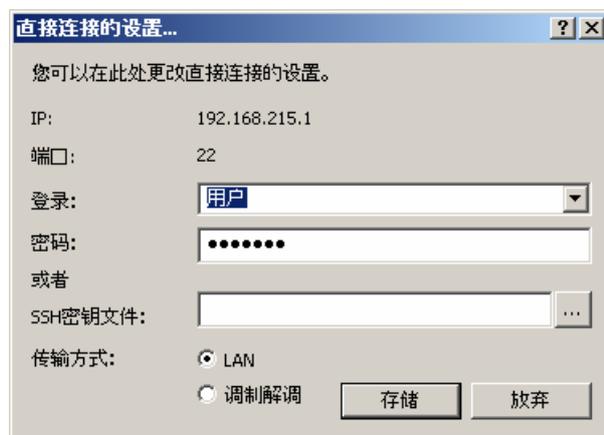


图 1-7 对话框：输入直接连接的登录数据

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 1-8 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”。

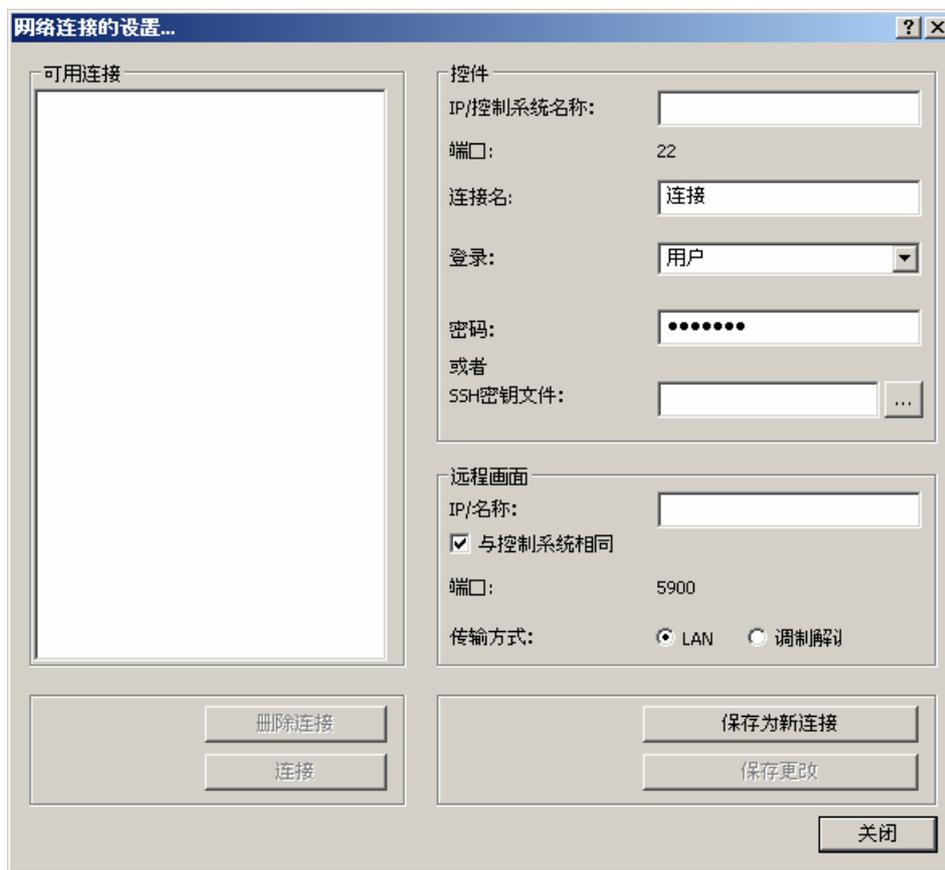


图 1-9 对话框：网络连接

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”。

1.5.4 示例：通过 NCU Connection Wizard 建立和控制系统的通讯的步骤

前提条件

SINAMICS S120 调试软件已经安装在 PG/PC 上。“NCU Connection Wizard”是该软件的一部分。

已经通过 PLC 编程工具建立和控制系统的通讯。

建立和控制系统的连接

PG/PC 上的操作步骤：

1. 通过该快捷方式或“开始”菜单启动“NCU Connection Wizard”：



2. 在“Select Control Model”对话框中选择 NCU 类型“840D solution line”，用于连接 SINUMERIK 828D。

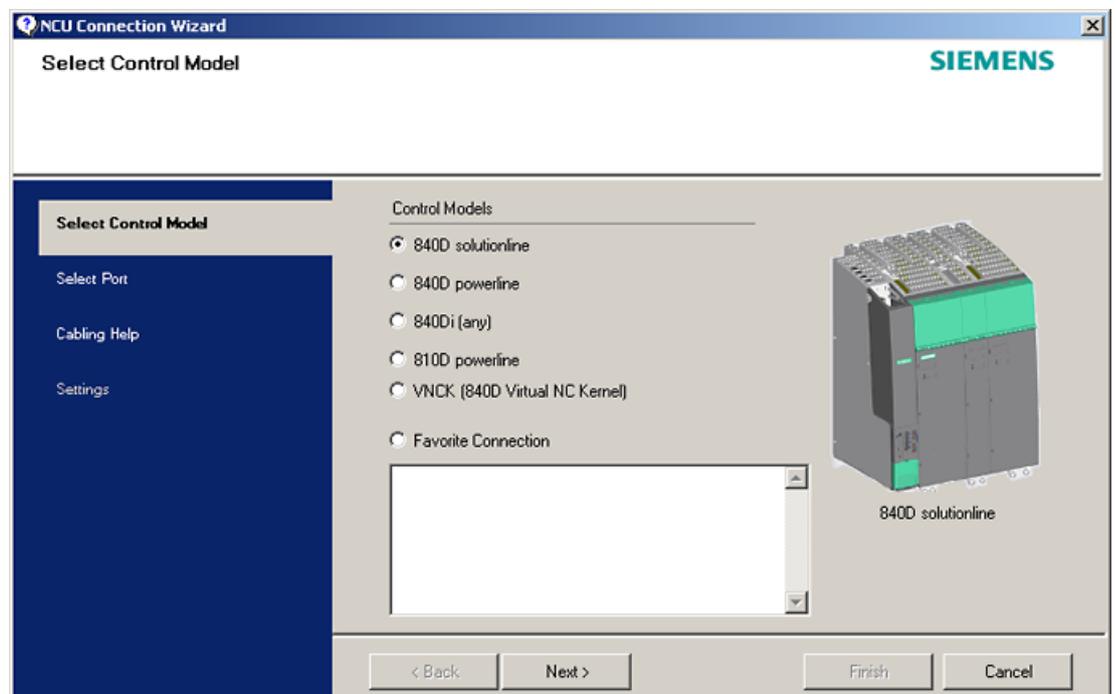


图 1-10 选择 NCU 类型

3. 在“Select Port”对话框中选择通过 Ethernet 连接的控制系统接口。

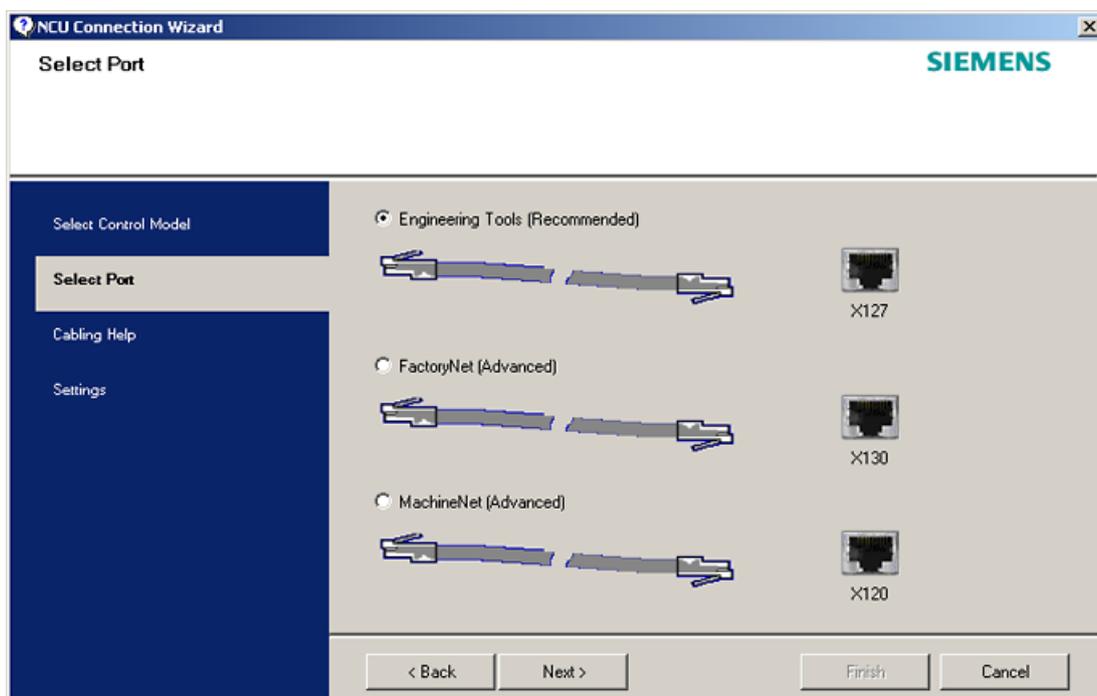


图 1-11 选择接口

4. 在“Cabling Help”对话框中，确认两个设备的电缆连接。

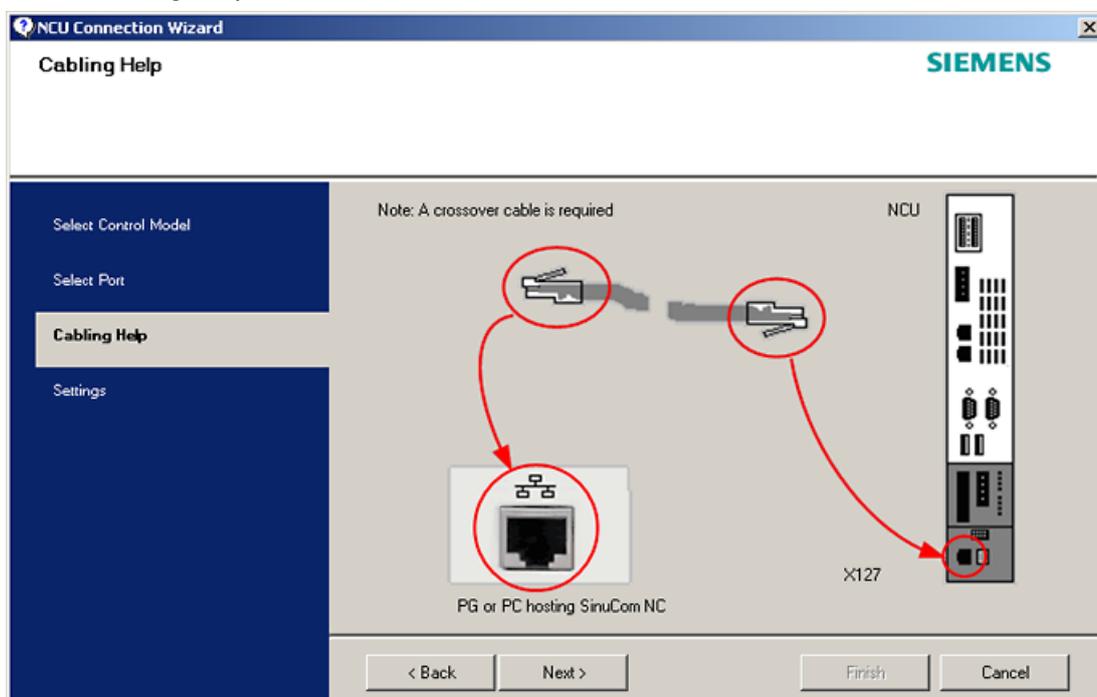


图 1-12 电缆连接

5. 在对话框“Settings”中检查 IP 地址，并输入该设置的名称。

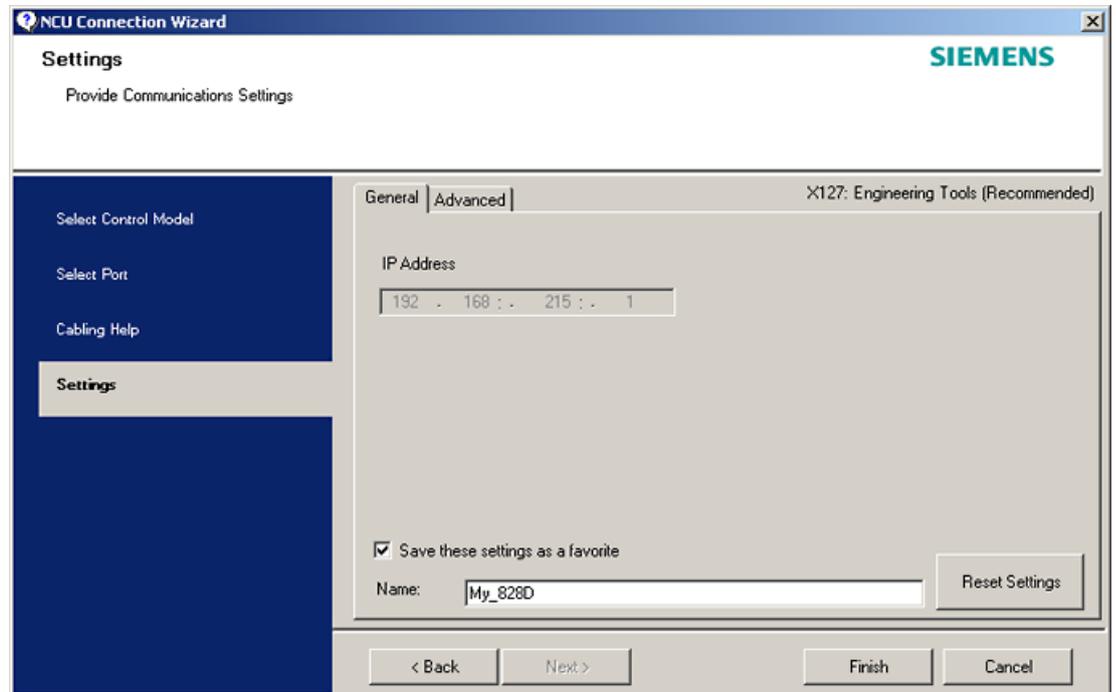


图 1-13 网络设置

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）
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。

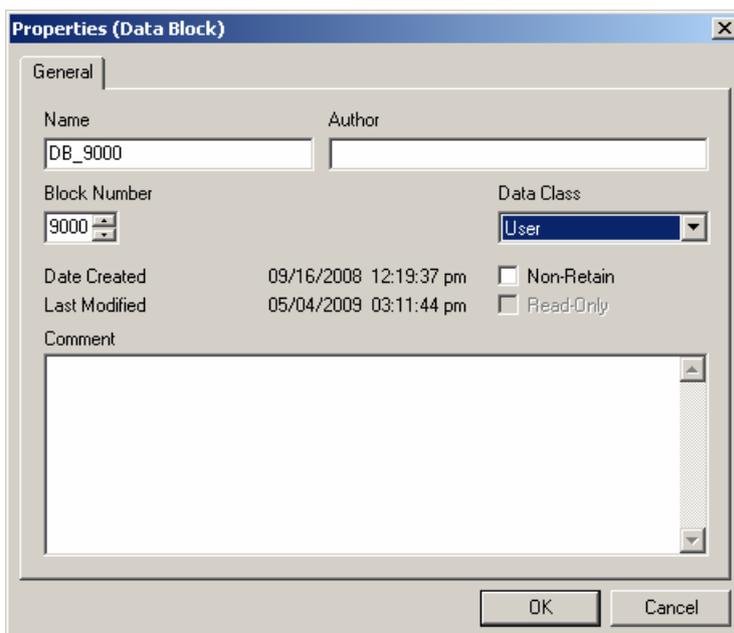


图 2-1 选择数据级

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 siemens	SYSTEM
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	
	/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	
	/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	
	/lngs	打包文本文件和版本文件
	/osal	
	/ace	ACE/TAO
	/qt	Qt
	/proj	Easy Screen 配置
	/template	
	/cfg	配置文件的模板
	/ing	文本文件的模板
	/tmpp	

2.3 操作软件中的数据级

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
	/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	图标文件

oem 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

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	
	/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
0	---	(预留)	---
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。

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。

另见

口令也可以由 PLC 删除（PI 服务: PI 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：

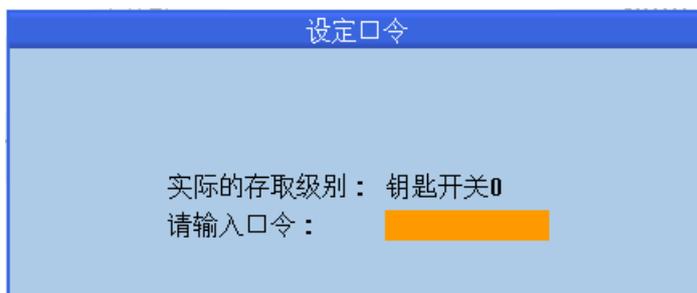


图 3-1 设定口令

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：

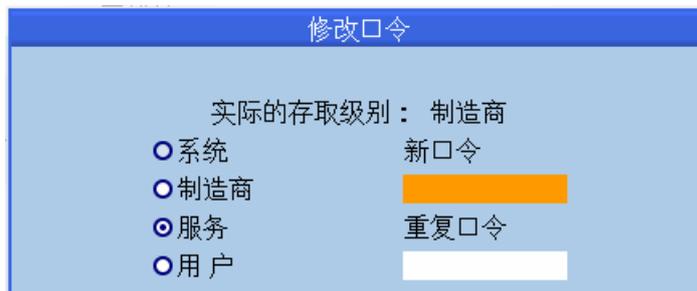


图 3-2 修改口令

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

在标准供货范围内，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。

更多系统语言

其他系统语言以 DVD 光盘的形式提供。DVD 光盘上也有一份安装说明。

3.4.2 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。

3.4 设置操作软件的语言

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：

<u>ai</u>	哎	哀	唉	埃	挨
<u>caise</u>	彩色				
<u>hongse</u>	红色				
<u>huise</u>	灰色				
<u>heli</u>	河裹				
<u>zuihaowan</u>	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：通过存档来备份数据(页 414)）

3.4.3 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。

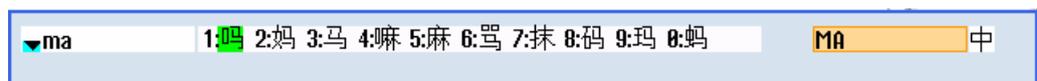


图 3-3 示例：简体中文

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。

3.4 设置操作软件的语言

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。

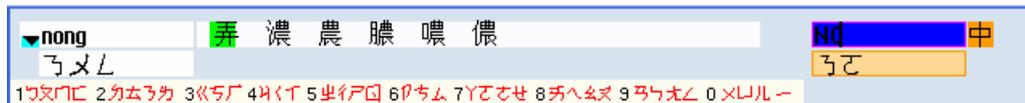


图 3-4 示例：注音方法

3.4.4 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。

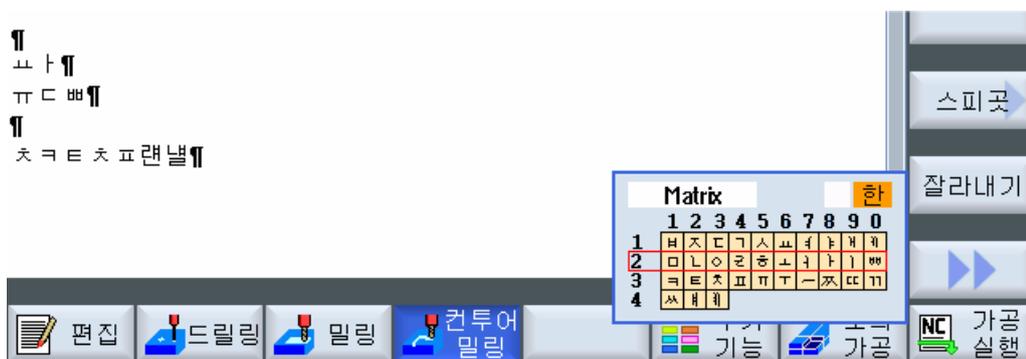


图 3-5 示例：程序编辑器

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	W	E	R	T	Y	U	I	O	P		Enter ↵
Caps Lock	₩	A	S	D	F	G	H	J	K	L			
↑		Z	X	C	V	B	N	M				↑	
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

产品使用

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（许可证密码）。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码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码输入到控制系统上。

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码，或没有有效的许可证密码，控制系统便发出报警 8081，并且不能输出 NC-START。

另见

许可证管理的相关定义 (页 55)

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码，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码

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，输入新的许可证密码。

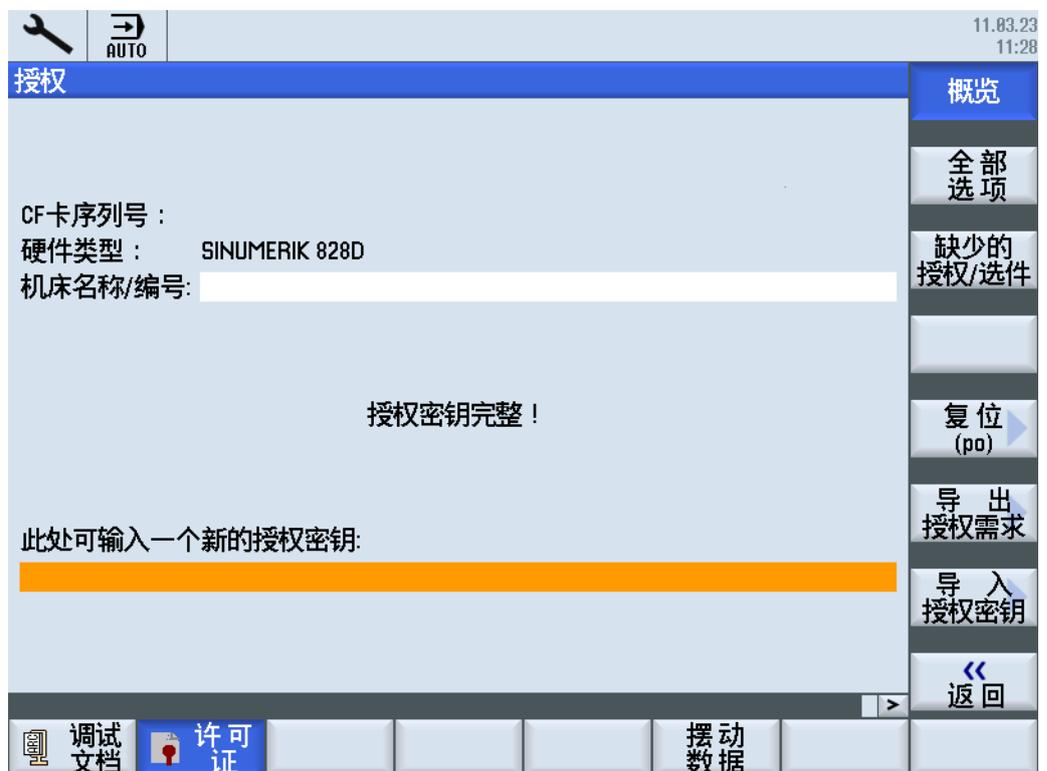


图 3-6 授权

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 54)
- 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. 点击软键“缺少的许可证/选件”，显示所有已授权的选件。在“已设置”栏中可以取消不需要的选件。

选项	已设置	获得授权
附加的1个轴/主轴 6FC5800-0AC20-0YB0	2	2
额外的1个定位轴/辅助主轴 6FC5800-0AC30-0YB0	0	0
drive based SI-Achse/Spindel zus. 1 Achse/Spindel 6FC5800-0AC50-0YB0	0	0
运行到固定挡块 (使用强制控制) 6FC5800-0AM01-0YB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
轮廓手轮 6FC5800-0AM08-0YB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
TRANSMIT和圆周表面转换 6FC5800-0AM27-0YB0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
双向补偿 6FC5800-0AM54-0YB0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
悬垂度补偿, 多维 6FC5800-0AM55-0YB0	<input type="checkbox"/>	<input checked="" type="checkbox"/>
生成的耦合'CP-STATIC' 6FC5800-0AM75-0YB0	<input type="checkbox"/>	<input type="checkbox"/>
用于刀具管理的替换刀具 6FC5800-0AM78-0YB0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
管理网络驱动器	<input type="checkbox"/>	<input type="checkbox"/>

图 3-7 许可 (示例)

4. 按下软键“根据许可证设置选件”，激活许可证密钥所涵盖的所有选件。用“确认”键确认接下来的安全询问。
5. 按下软键“复位(po)”使刚刚激活的选件生效。出现一条安全询问。
6. 按下软键“确认”，开始热启动。
 - 或者-
7. 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
- 订货号
- → 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 → 许可密钥

3.5 检查和输入许可证

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，具有唯一性。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

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。

许可密钥

许可密钥是所有 → 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 用户报警

创建用户 PLC 报警

700 000 - 700 247 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”，在操作界面上输入用户 PLC 报警。

随后可以选择：

报警文本类型	xml 文件的名称
用户循环报警	oem_alarms_cycles
用户 PLC 报警	oem_alarms_plc
用户子程序信息文本	oem_partprogram_messages

载入用户 PLC 报警

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。

资料

包含系统响应和清除条件的报警的详细说明请见：SINUMERIK 828D 诊断手册

3.6.1 用户 PLC 报警的结构

用户 PLC 报警的结构

用户 PLC 报警具有以下结构：

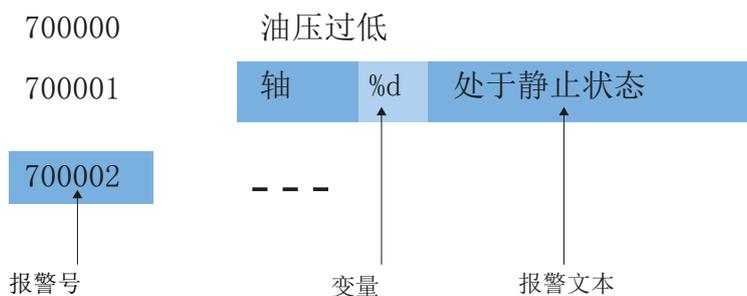


图 3-8 报警结构

下表展示了 PLC 报警的生效方式。

1. PLC 信号使具有对应编号的报警触发并输出。
2. 如果为该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。
3. 报警触发时 NCK 的响应在 MD14516[x] 的各个索引中定义，见下表。
4. 报警文本可自由选择，最长不超过 255 个字符。

报警号码	PLC 信号	PLC 变量	报警响应 (MD)	报警文本
700000	DB1600.DBX0.0	DB1600.DBD1000	14516[0]	报警 1
700001	DB1600.DBX0.1	DB1600.DBD1004	14516[1]	报警 2
700002	DB1600.DBX0.2	DB1600.DBD1008	14516[2]	报警 3
700003	DB1600.DBX0.3	DB1600.DBD1012	14516[3]	报警 4
700004	DB1600.DBX0.4	DB1600.DBD1016	14516[4]	报警 5
700005	DB1600.DBX0.5	DB1600.DBD1020	14516[5]	报警 6
700006	DB1600.DBX0.6	DB1600.DBD1024	14516[6]	报警 7

续:				
700247	DB1600.DBX30.7	DB1600.DBD1988	14516[247]	报警 248

定义 NCK 的响应

NCK 可能出现以下响应：

MD14516[x]	含义
位 0	禁止 NC 启动
位 1	读入禁止
位 2	所有轴禁止进给
位 3	急停
位 4	PLC 停止
位 5	备用
位 6	报警或信息的定义 位 6=1: → 报警, 位 6=0: → 信息
位 7	上电

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

3.6.2 创建用户 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”，创建用户 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 425)

示例：创建针对用户 PLC 报警的在线帮助的步骤 (页 77)

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

为持久保存报警记录，可在 CF 卡上执行备份。

说明

备份报警记录

为持久保存报警记录，可在 CF 卡上执行备份，但备份次数受限。

- 确保只有在必要时才执行备份！
- 确保一旦不再需要备份报警记录时，“对每一事件”设置会被取消。

缺省设置：不备份报警记录。

筛选报警

为限制报警记录中的事件，请设置筛选。可按以下标准进行筛选：

- 时间间隔
- 报警号范围

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”。
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_slaesvccconf.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“slaesvccconf.xml”
4. 在编辑器中打开用户专用文件“slaesvccconf.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“slaesvccnf.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还无法识别。

3.6 定义 PLC 用户报警

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 82)

- 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 80)

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)
Ill-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)

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"/>
```

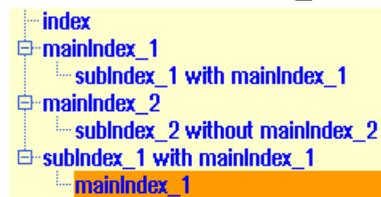


图 3-9 示例： 二级索引

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 77)

- 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	

标签	说明	注释
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	
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	

3.7 创建 OEM 专用在线帮助

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

3.7 创建 OEM 专用在线帮助

属性	参数值	说明
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。

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/him/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 425)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"。

3.7 创建 OEM 专用在线帮助

3. 将帮助手册如“hmi_myhelp.xml”移到文件夹“eng”中。

/oem/sinumerik/him/hlp/eng/hmi_myhelp.xml

4. 将帮助文件复制到目录:

/oem/sinumerik/him/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/him/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。

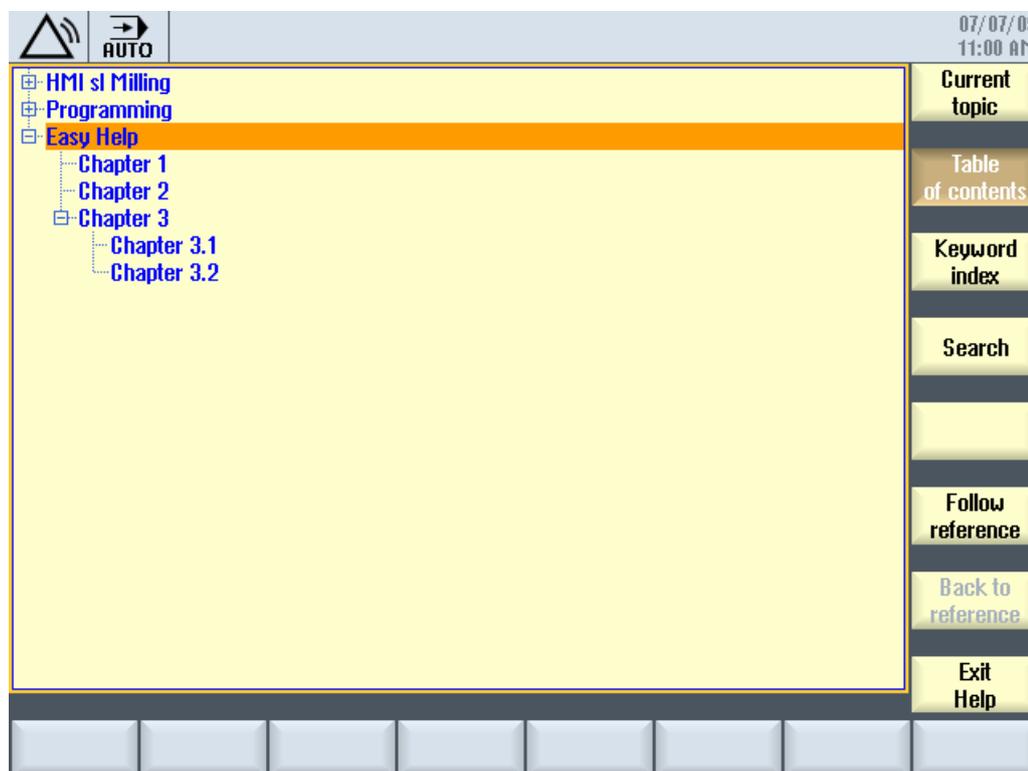


图 3-10 示例：OEM 在线帮助

关键字索引中的条目：

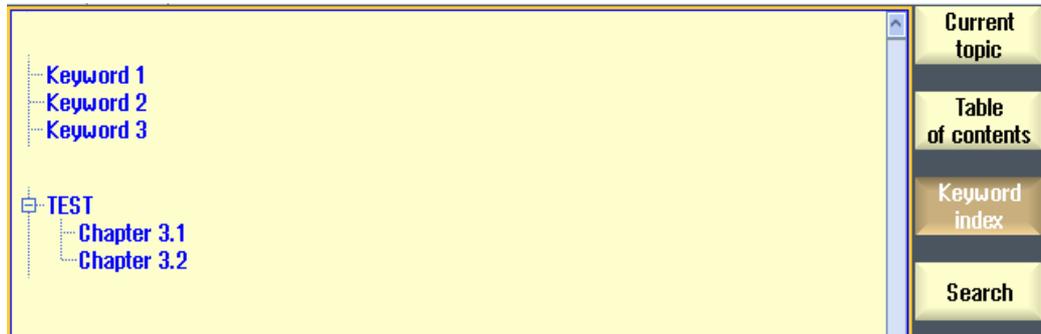


图 3-11 示例：索引

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”

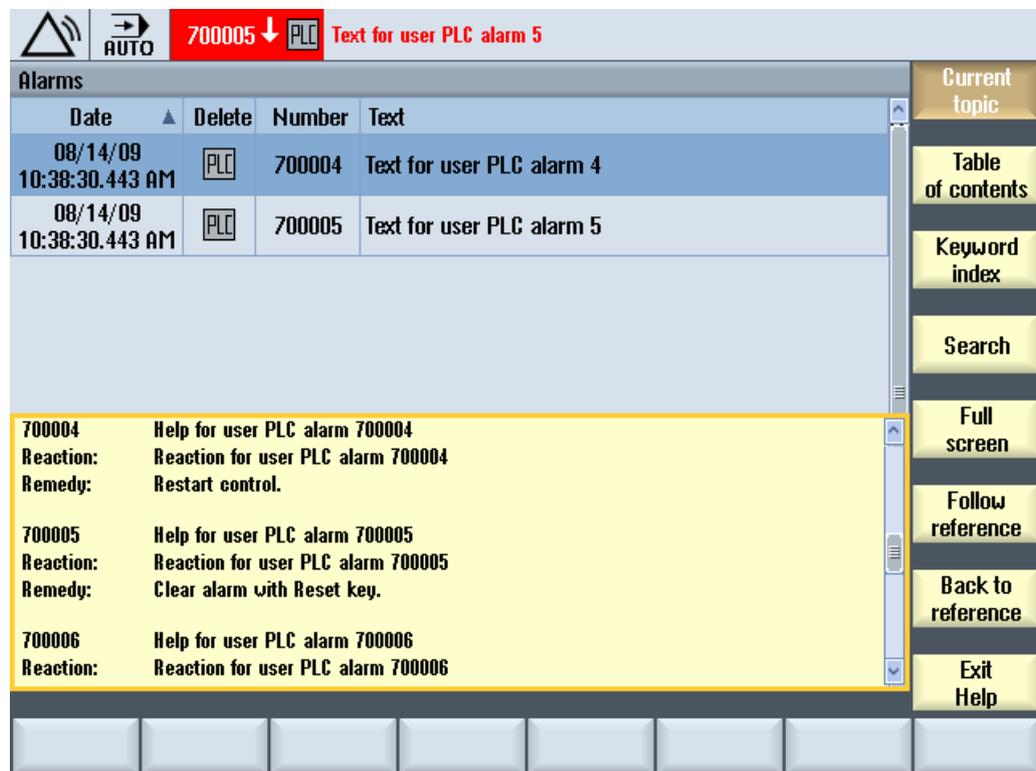


图 3-12 示例：针对用户 PLC 报警的在线帮助

3.7 创建 OEM 专用在线帮助

帮助文件的结构

帮助文件中可以使用的条目有：

条目	含义
<code>AlarmNr</code>	报警编号超级链接
<code></code>	相应报警的帮助文本
<code><td width="85%">.....</td></code>	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>

```

```
sinumerik_alarm_oem_plc_pmc.html
```

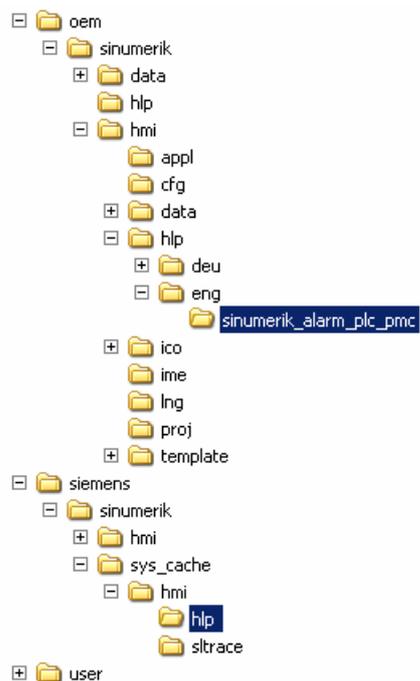
```
</tr>  
<br>  
...  
</table>  
<p></p>  
</body>  
</html>
```

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。

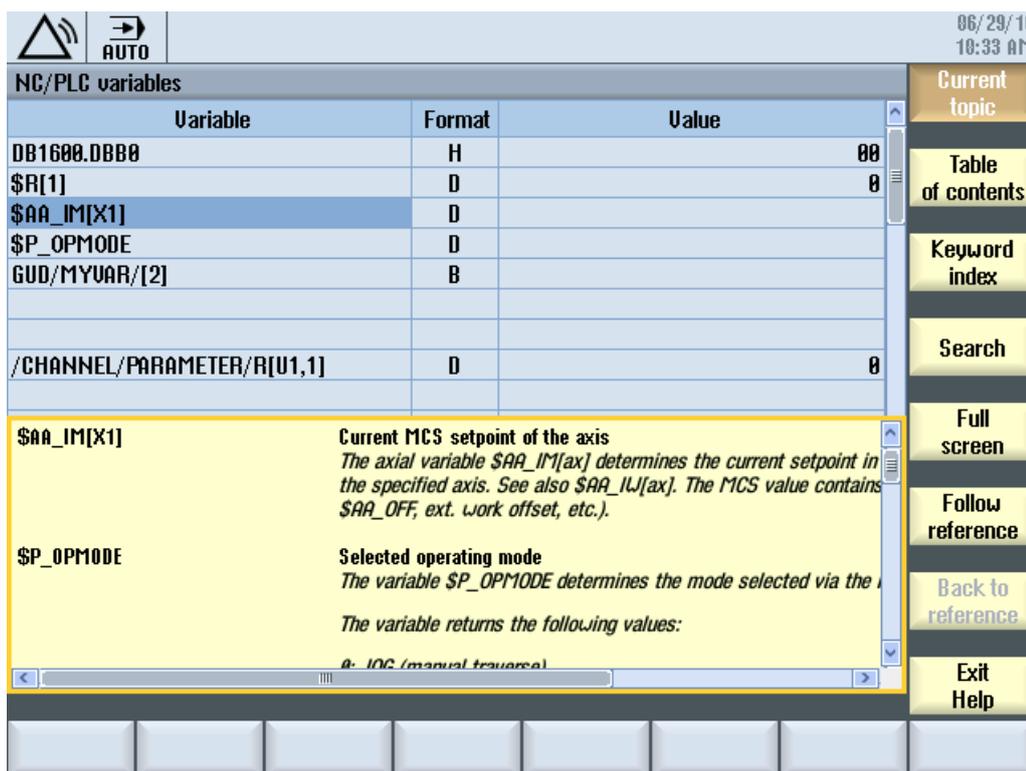


图 3-13 示例：用户变量的在线帮助

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]
/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：

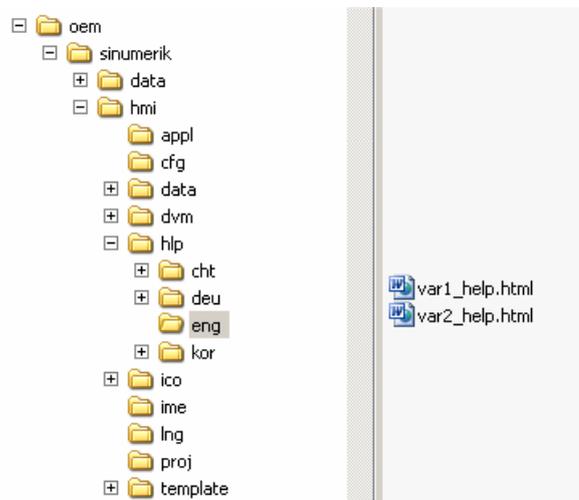
```
/oem/sinumerik/hmi/cfg/sldgvarviewhelp.ini
```

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。



3.7 创建 OEM 专用在线帮助

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">
  <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>
```

3.7 创建 OEM 专用在线帮助

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 74)

调试外设

4.1 外设模块的连接

4.1.1 激活外设模块

PLC 外设的机床数据

以下组件具有用于 PLC 的输入/输出映像区的固定地址：外设模块、机床控制面板、SENTRON PAC3200、SENTRON PAC4200 和 PN 总线耦合器。

机床数据包含两个数组，用于取消 PLC 输入/输出映像区的更新：

机床数据		取值范围	
12986[i]	\$MN_PLC_DEACT_IMAGE_LADDR_IN	$0 \leq i \leq 7$	输入地址
12987[i]	\$MN_PLC_DEACT_IMAGE_LADDR_OUT	$0 \leq i \leq 7$	输出地址

SINUMERIK 828D 采用外设模块固定的最大配置运行。在出厂时，禁止外设模块传输数据到 PLC 输入/输出映像区。

包含逻辑输入地址的数组：

MD	逻辑输入地址	禁止传输数据到 PLC
12986[0]	0	1. PP 模块 失效
12986[1]	9	2. PP 模块 失效
12986[2]	18	3. PP 模块 失效
12986[3]	27	4. PP 模块 失效
12986[4]	36	5. PP 模块 失效
12986[5]	96	PN 总线耦合器失效
12986[6]	112	机床控制面板失效
12986[7]	132	SENTRON PAC4200 失效
12986[8]	144	SENTRON PAC3200 失效

输出地址数组为空（缺省设置）：MD12987[i] = -1

4.1 外设模块的连接

需要激活外设模块时，既不要在 MD12986[i] 中，也不要再 MD12987[i] 中输入模块的地址。而是输入值 -1 (“空”)。

示例

激活两个外设模块、机床控制面板和 SENTRON PAC4200:

MD	逻辑输入地址	禁止传输数据到 PLC
12986[0]	-1	第 1 个外设模块生效
12986[1]	-1	第 2 个外设模块生效
12986[2]	18	第 3 个外设模块失效
12986[3]	27	第 4 个外设模块失效
12986[4]	36	第 5 个外设模块失效
12986[5]	96	PN 总线耦合器失效
12986[6]	-1	机床控制面板生效
12986[7]	-1	SETRON PAC4200 生效
12986[8]	144	SETRON PAC3200

说明

在 PLC 用户程序中使用一个失效的模块的输入/输出地址时，不会触发任何报警。PLC 用户程序始终通过映像存储器运行。模块和物理输入/输出是否连接，由 MD12986[i] 和 MD12987[i] 设定。

随后会周期性监控生效的模块是否出现故障。

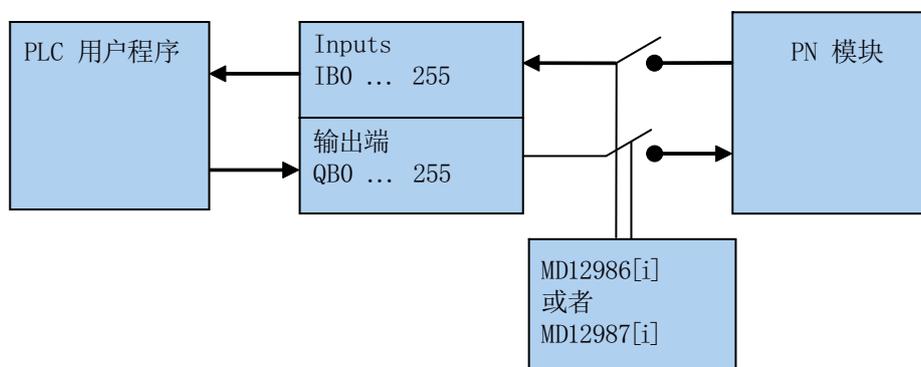


图 4-1 外设开关

4.1.2 设置外设模块的地址

外设模块的 IP 地址

下表列出了 DIP 开关设置 S1 和对应的各个外设模块的 IP 地址。其中考虑了 PROFINET 中 PLC IO 接口支持的最大配置：由外设模块、总线耦合器和机床控制面板组成。

外设模块	总线	设备名称	IP 地址	输入地址	输出地址
			192.168.214.	(设置 MD12986[x] = -1 激活)	
				索引 n:	
第 1 个数字量 PP 模块	PN	pp72x48pn9	9	0 ... 8	0 ... 5
第 2 个数字量 PP 模块	PN	pp72x48pn8	8	9 ... 17	6 ... 11
第 3 个数字量 PP 模块	PN	pp72x48pn7	7	18 ... 26	12 ... 17
第 4 个数字量 PP 模块	PN	pp72x48pn6	6	27... 35	18 ... 23
第 5 个数字量 PP 模块	PN	pp72x48pn5	5	36 ... 44	24 ... 29
未指定				45	30 ... 55
				索引 d:	
第 1 个诊断 PP 模块	PN	pp72x48pn9	9	46 ... 47	--
第 2 个诊断 PP 模块	PN	pp72x48pn8	8	48 ... 49	--
第 3 个诊断 PP 模块	PN	pp72x48pn7	7	50 ... 51	--
第 4 个诊断 PP 模块	PN	pp72x48pn6	6	52 ... 53	--
第 5 个诊断 PP 模块	PN	pp72x48pn5	5	54 ... 55	--
				索引 m:	
第 1 个模拟量 PP 模块	PN	pp72x48pn9	9	56 ... 63	56 ... 63
第 2 个模拟量 PP 模块	PN	pp72x48pn8	8	64 ... 71	64 ... 71
第 3 个模拟量 PP 模块	PN	pp72x48pn7	7	72 ... 79	72 ... 79
第 4 个模拟量 PP 模块	PN	pp72x48pn6	6	80 ... 87	80 ... 87
第 5 个模拟量 PP 模块	PN	pp72x48pn5	5	88 ... 95	88 ... 95
PN/PN 耦合器 *	PN	pn-pn-coupler20	20	96 ... 111	96 ... 111
外部机床控制面板	PN	mcp-pn64	64	112 ... 125	112 ... 121

4.1 外设模块的连接

外设模块	总线	设备名称	IP 地址	输入地址	输出地址
保留		--	--	126 ... 131	122 ... 123
Sentron PAC 4200 *	PN	pac4200-pn21	21	132 ... 143	132 ... 143
Sentron PAC 3200 *	PN	pac3200-pn22	22	144 ... 155	144 ... 155

索引 n、m、d 一直为地址范围的开始地址。

*) 组件的 IP 地址不是通过开关位置设置，而是通过配置设置。

机床数据参数化

5.1 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。

5.1 机床数据的分类

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：

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 5-1 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
rev/s**3	rev/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。

5.2 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

5.3 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。

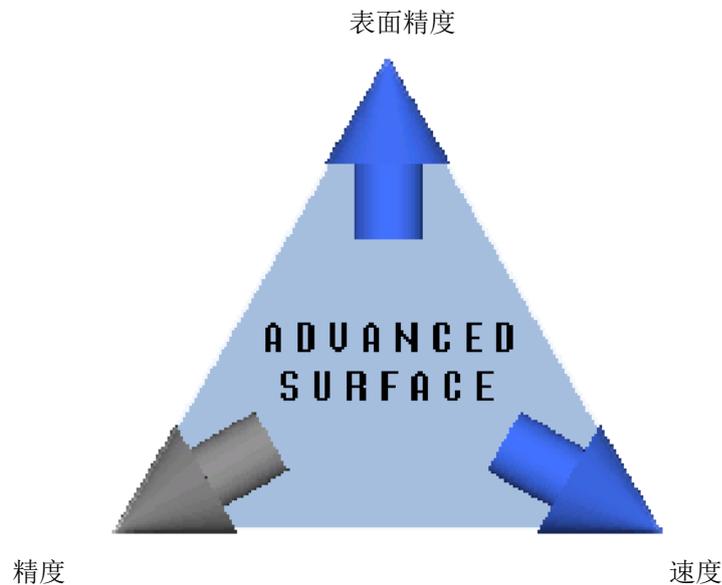


图 5-2 工件表面加工

检查 *Advanced Surface* 的设置

在 Toolbox 828D 安装成功后，目录\Examples\AS_checkprograms 下会生成以下测试程序：

- MDC3AXV3B_SW2_6.SPF
- MDC3AXV3A_SW2_7_SW4_4.SPF
- MDC3AXV3B_SW4_5.SPF

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_FFWD=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
```

5.4 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_ADIS	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_ACCEL	0	0	M	--
20602[1]	\$MC_CURV_EFFECT_ON_PATH_ACCEL	0	0	M	--
20602[2]	\$MC_CURV_EFFECT_ON_PATH_ACCEL	0,6	0,65	M	--
20602[3]	\$MC_CURV_EFFECT_ON_PATH_ACCEL	0,6	0,6	M	--
20602[4]	\$MC_CURV_EFFECT_ON_PATH_ACCEL	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_FACTOR	0	1	M	--
20605[1]	\$MC_PREPDYN_SMOOTHING_FACTOR	0	1	M	--
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

机床数据编号	名称	缺省设置	建议	数据级	参数
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_PROTOC_NUM_ETP_STD_TYP	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	--
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

5.4 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
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	--

5.5 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
	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_CTRL_OUT_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

- 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

三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 (10V 的 80%)	额定输出电压
MD32260 RATED_VELO	3000 (8V)	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

5.6 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

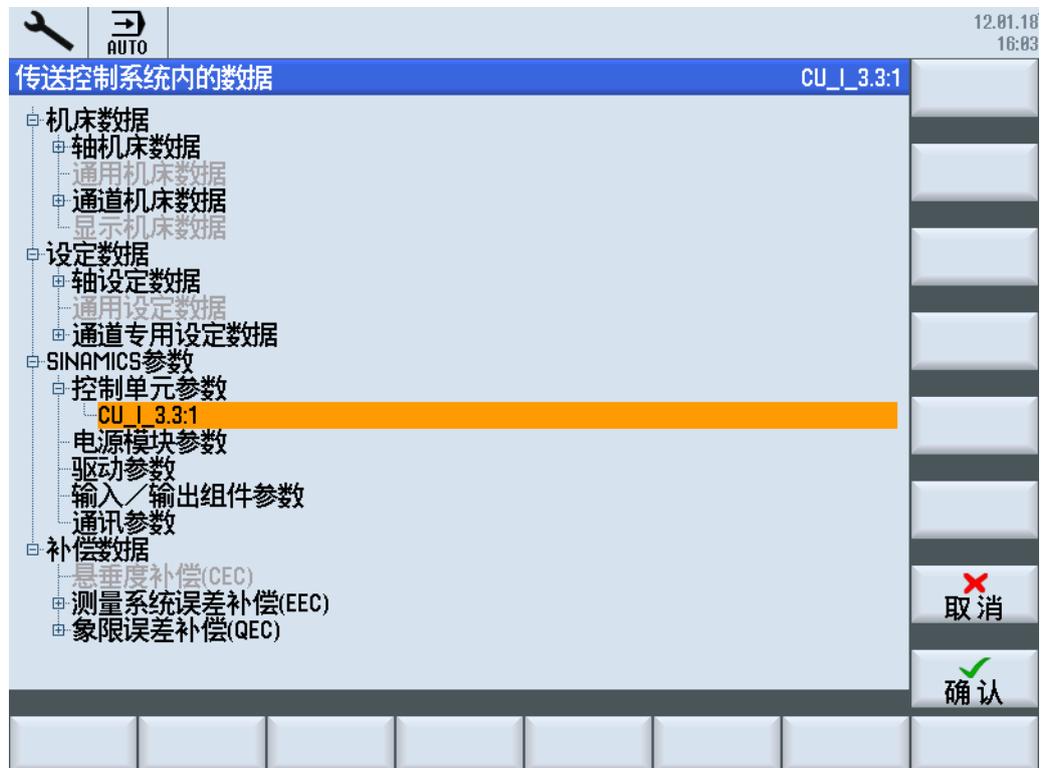


图 5-3 管理数据

5.6.1 在控制系统内传输数据的步骤

在控制系统内传输数据

说明

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. 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。

5.6.2 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

说明

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。

5.6.3 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。

调试驱动

6.1 配置驱动

6.1.1 示例： SINAMICS S120 Combi 的配置

概述

Toolbox CD 光盘中含有免费的 SINAMICS S120 调试软件。在系统的操作界面上可以完全使用 SINAMICS S120 调试功能之前，驱动的调试目前分为两个步骤：

- 通过操作界面上的功能自动检测拓扑结构。
- 通过 SINAMICS S120 调试软件完成编码器的配置以及参数设置。PG/PC 连接到 SINUMERIK 828D 正面的以太网接口上。

配置驱动

驱动配置可以参照章节“系统概述 (页 11)”中关于 SINAMICS S120 Combi 的示例 2。DRIVE-CLiQ 连接必须符合“S120 Combi 的拓扑规则 (页 172)”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	名称
---	1	控制单元
---	2	电源模块
MSP1	3	伺服 1
MX1	4	伺服 2
MY1	5	伺服 3
MZ1	6	伺服 4
MA1	7	伺服 5
---	11	集线器模块
---	9	TM54F 主站

配置过程

配置主要分为以下几个步骤：

- 步骤 1：配置主轴。
- 步骤 2：配置进给轴。
- 步骤 3：分配轴并备份数据：

操作步骤参见章节“示例：分配轴的步骤 (页 153)”。

下面详细说明各个步骤的具体操作。

初始状态

在开始前：

- 将 PG/PC 和控制系统连在一起：参见章节“建立和控制系统的通讯 (页 21)”
- 在“Siemens default data”模式下启动控制系统。
- 控制系统的操作区域“调试”中显示如下画面：



图 6-1 “Siemens default data”启动后，控制系统运行

操作步骤

配置驱动的步骤:

1. 按下“驱动系统”软键启动调试。
2. 使用“确认”软键确认以下询问，以启动设备配置。

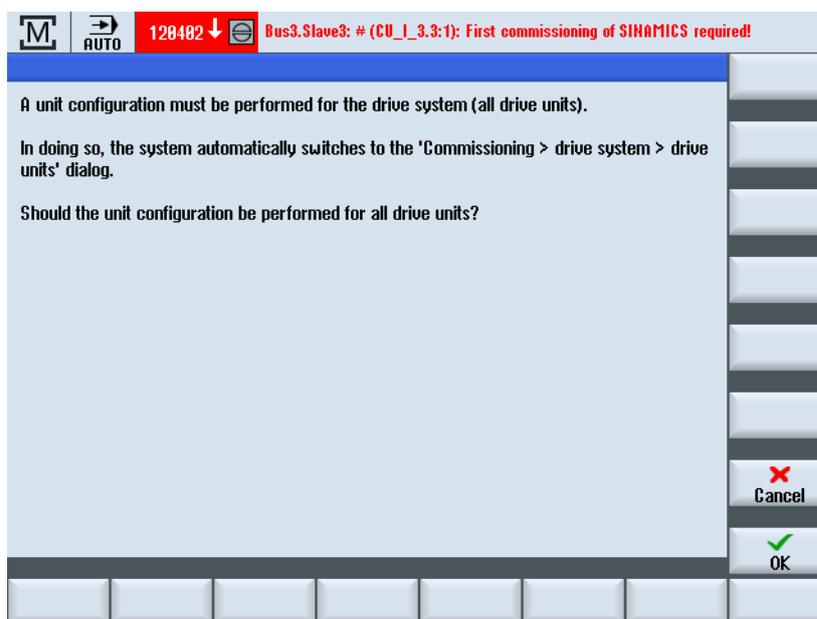


图 6-2 用 SINUMERIK Operate 启动调试

之后，系统将自动读取拓扑结构。

3. 按下“确认”软键确认以下询问，以执行设备配置。

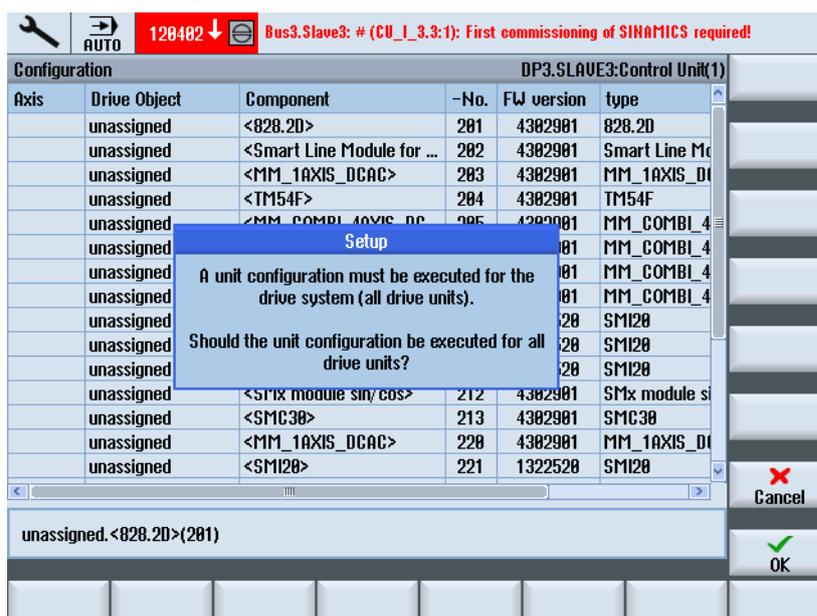


图 6-3 设备配置

- 按下“确认”软键确认以下询问，以执行 NCK 上电复位。该过程可能要持续几分钟。

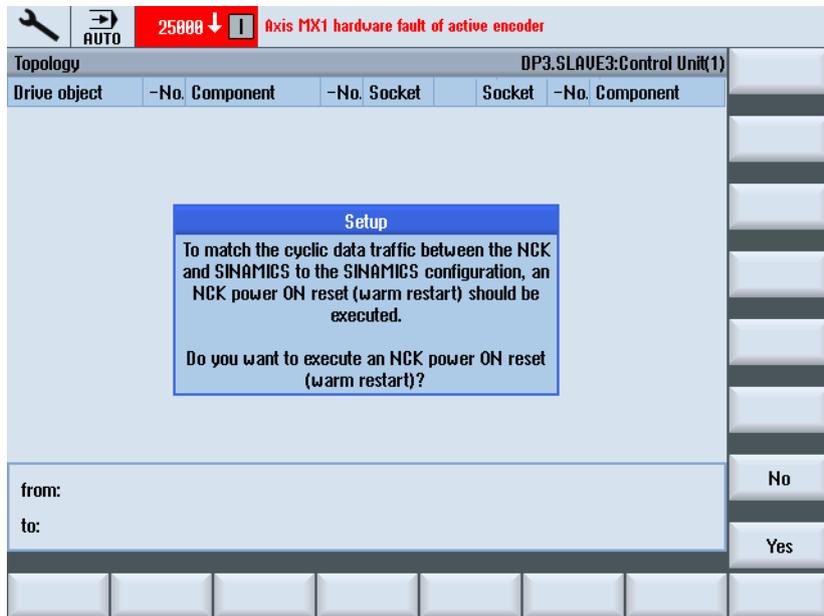


图 6-4 确认上电复位

- 启动后，您会看到如下提示：
- 按下“确认”。

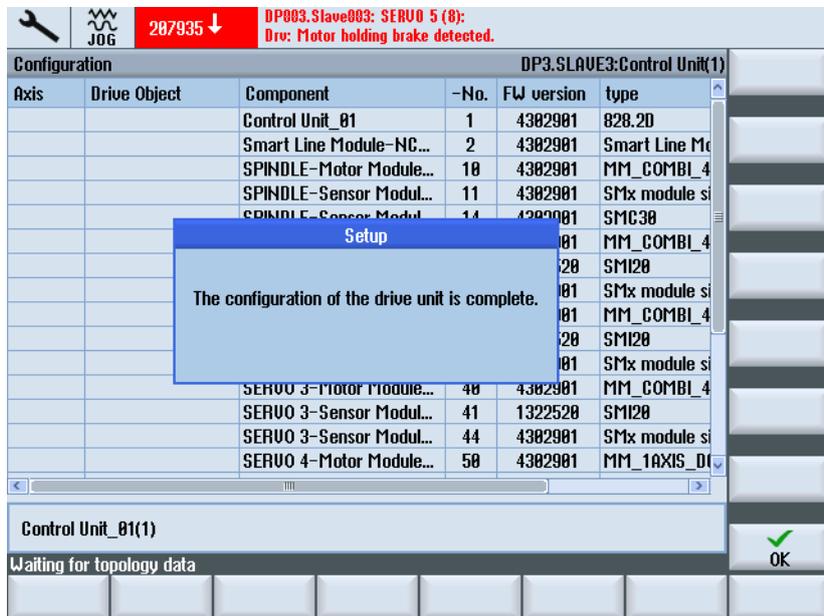


图 6-5 结束自动配置

- 在得出拓扑数据后，配置结束。

后续调试需要使用 PG/PC 上安装的SINAMICS S120 调试软件 (页 13)。

8. 选择操作区域“调试”。
9. 使用存取级别“制造商”的口令。
10. 按下“确认”软键确认以下询问，以继续进行配置。

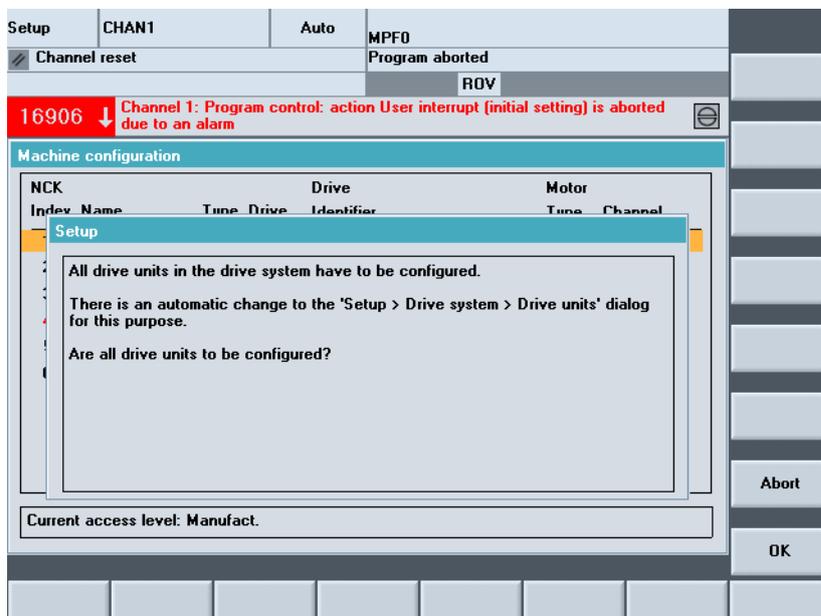


图 6-6 开始调试

6.1.2 示例：配置带 SMC 编码器的主轴的步骤

带 SMC20 的主轴配置

操作步骤：

1. 首先用软键“驱动+”或“驱动-”选定主轴。
2. 选择“更改”，配置主轴。

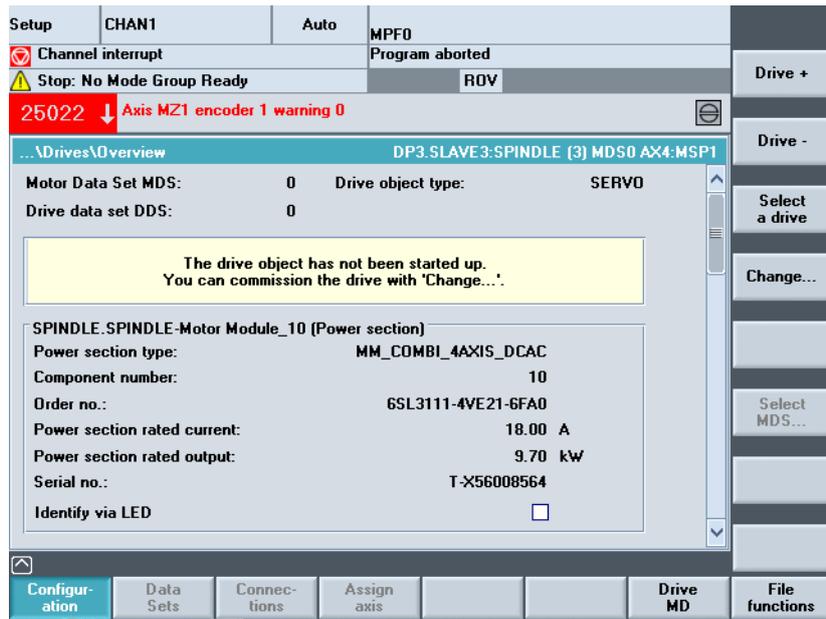


图 6-7 驱动一览

显示下列主轴数据：

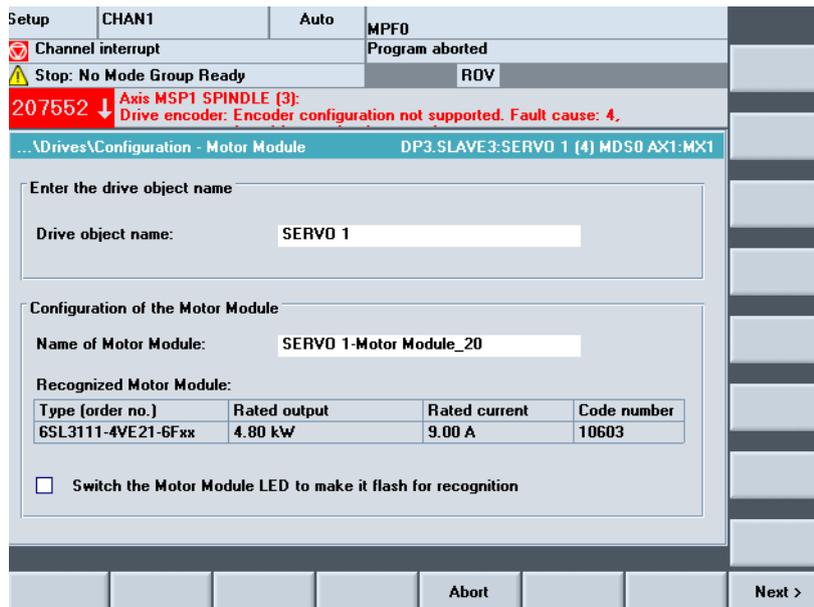


图 6-8 驱动配置：电机模块（主轴）

3. 选择“继续 >”，选定相应的电机： 可以通过“电机类型”或“搜索”电机代码，选择电机。

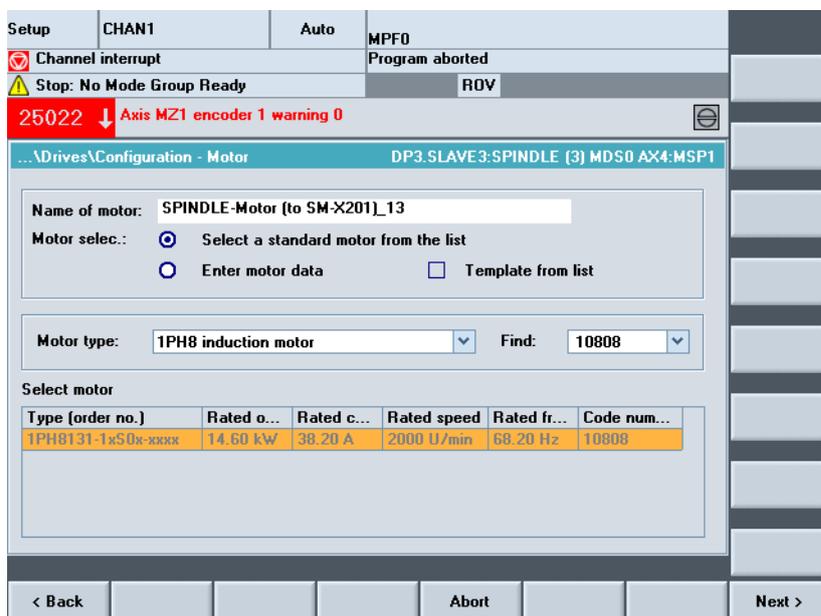


图 6-9 驱动\配置：电机

4. 选择“继续 >”，配置电机抱闸：

请注意关于轴承型号的说明。

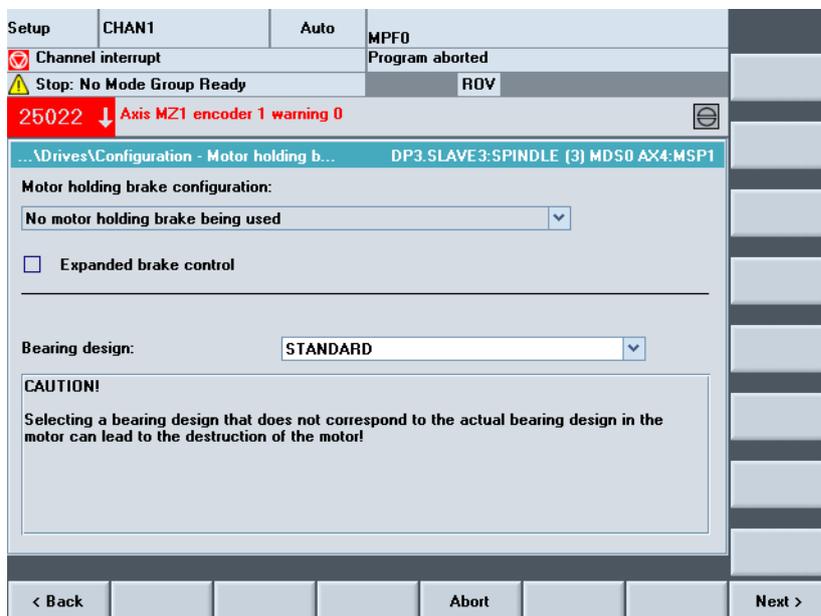


图 6-10 驱动\配置：电机抱闸

- 选择“继续 >”，选定编码器：在自动检测拓扑结构时系统已经正确指定了两个编码器（电机上的测量系统和机床上的直接测量系统）。

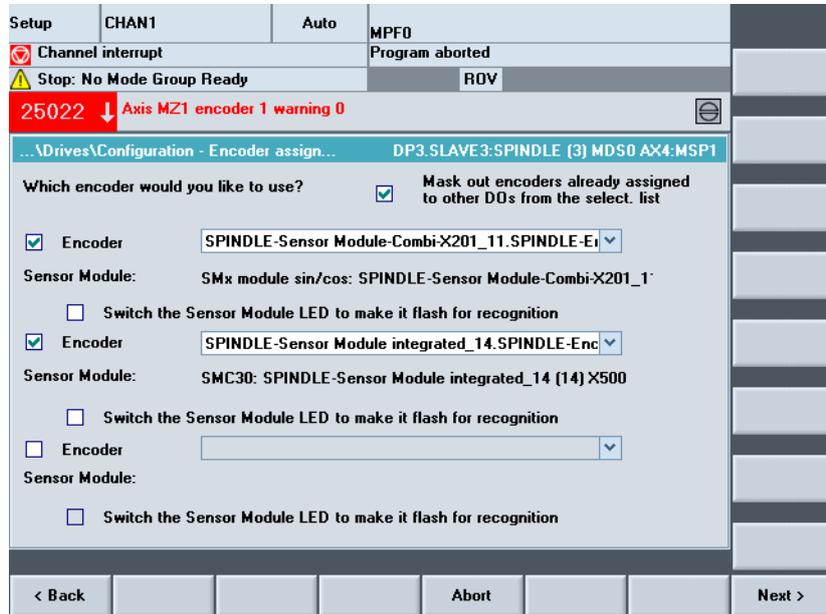


图 6-11 驱动配置：指定编码器

- 选择“继续 >”，配置电机编码器：

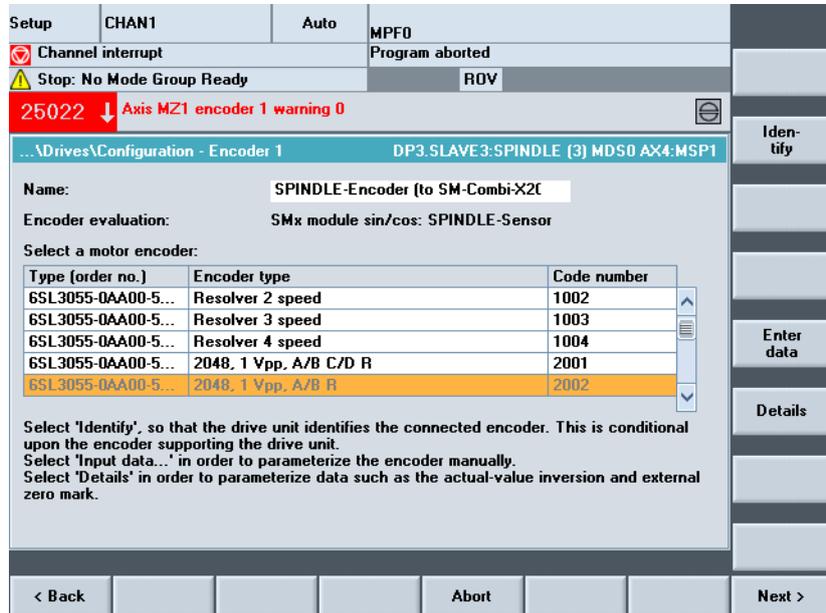


图 6-12 驱动配置 - 编码器 1

- 选择“输入数据”，使编码器的属性和 4096 转相配套。

8. 如果点击“确认”来确认设置，编码器将显示为“用户定义”：

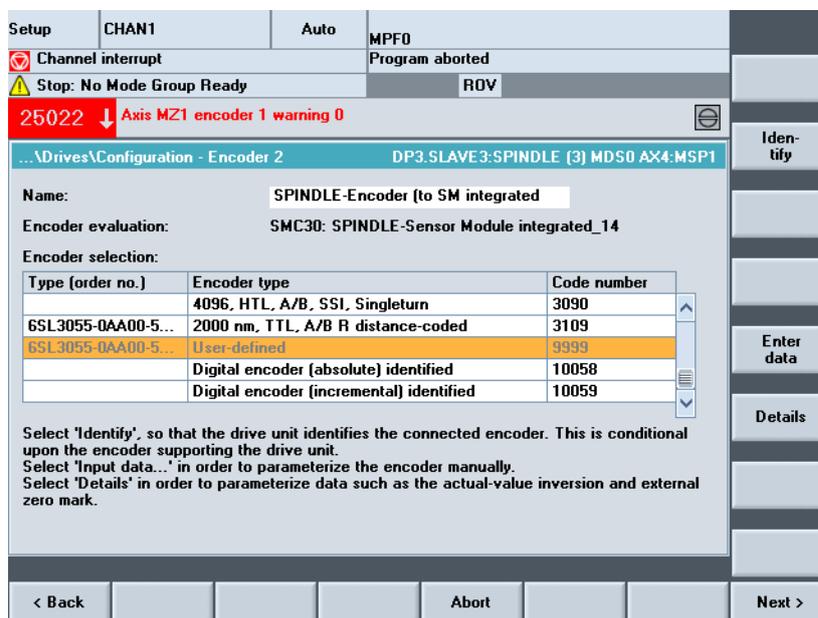


图 6-13 驱动配置 - 编码器 2: 用户定义

9. 选择“继续 >”，采用报文 116（转速前馈）的缺省设置。现在系统会传送用于内部驱动参数的附加过程数据(PZD)，并将这些数据保存在系统变量中。

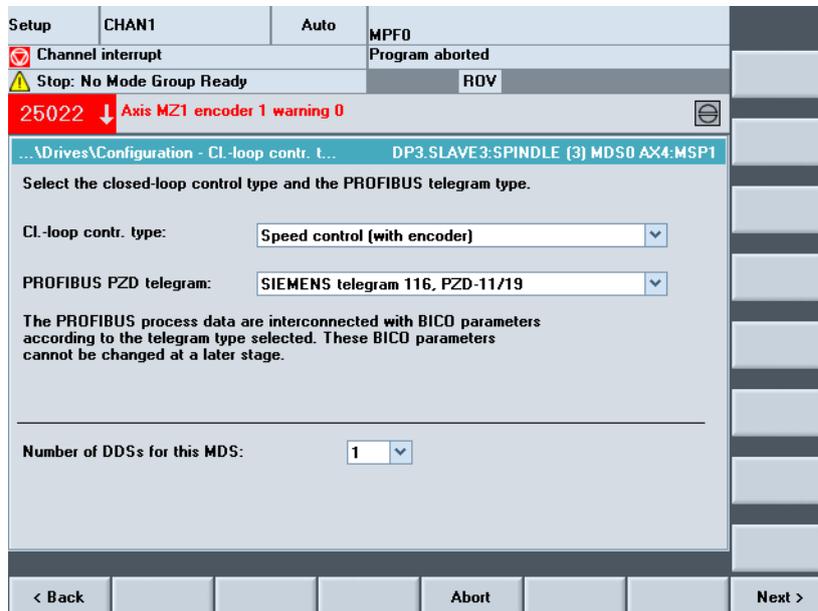


图 6-14 驱动配置 - 控制方式

-或者-

10. 选择报文 136（转矩前馈），使系统传送用于内部驱动参数的附加过程数据(PZD)，并将这些数据保存在系统变量中。

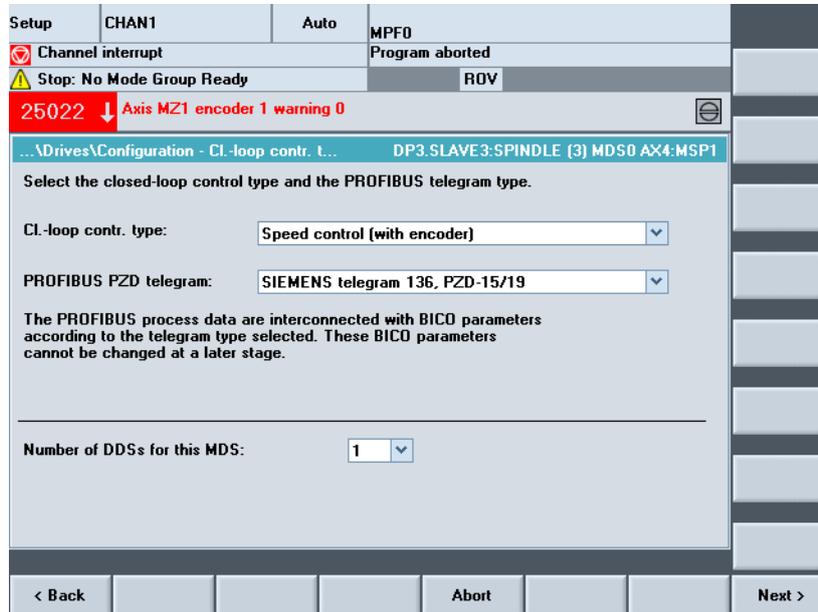


图 6-15 驱动配置 - 控制方式

11. 选择“继续>”，保存所作选择。该过程可能要持续几分钟。
12. 选择“继续 >”，进行 BICO 配置。

随后系统会显示以下主轴配置一览，方明您再次检查主轴的所有数据：

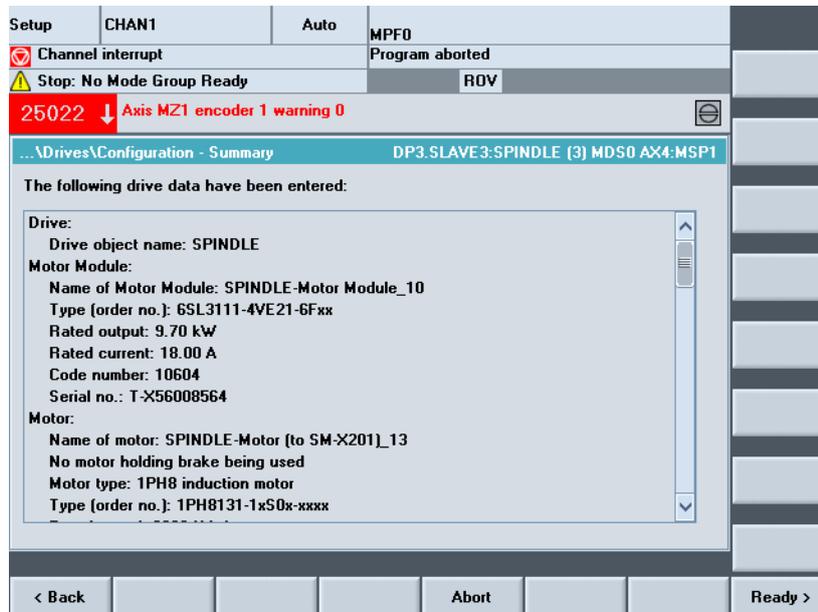


图 6-16 驱动配置 - 总结

13. 选择“完成 >”，结束主轴配置

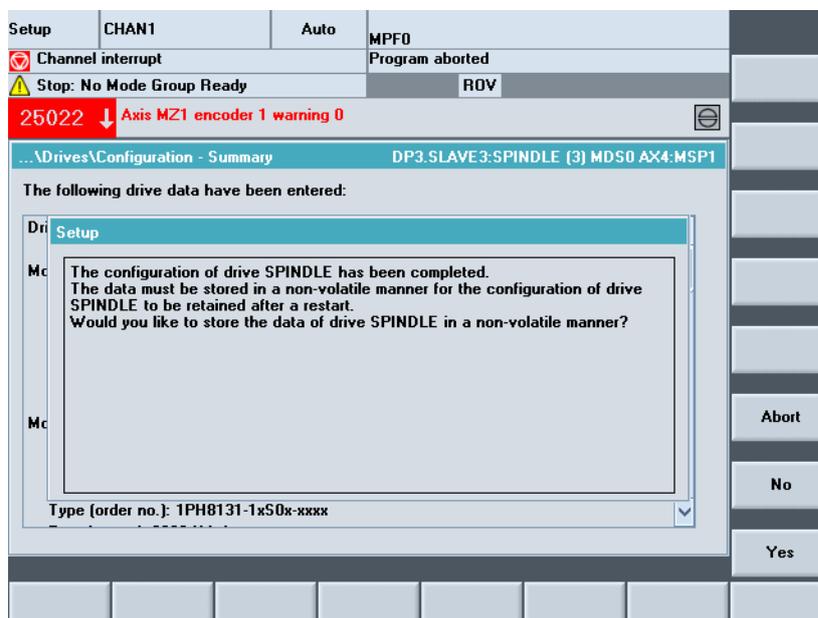


图 6-17 调试 - 询问

14. 选择“是”保存配置。

结果

主轴的配置现在成功结束。

6.1.3 示例：配置带 SMI 编码器的轴

带 SMI 编码器的轴配置

操作步骤：

1. 首先用软键“驱动+”或“驱动-”选定轴。
2. 选择“更改”，配置轴。

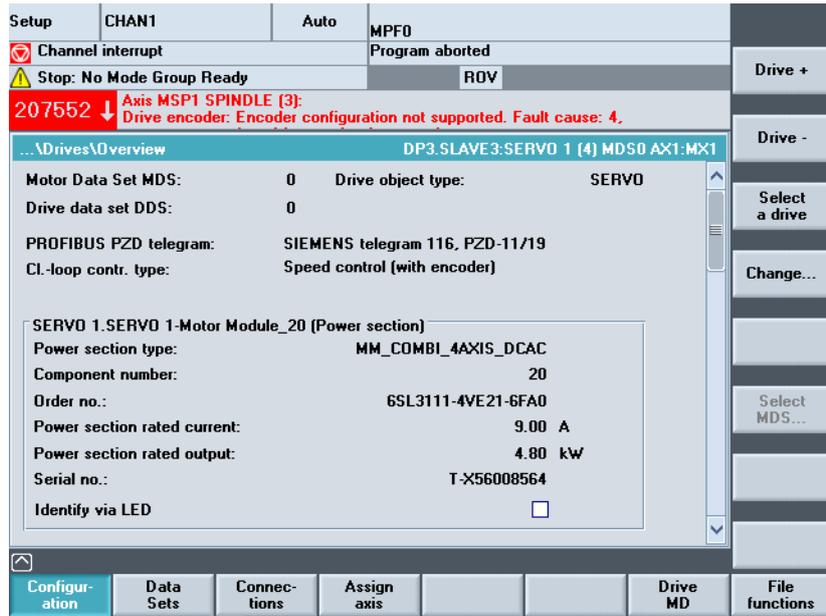


图 6-18 驱动一览

显示下列轴数据：

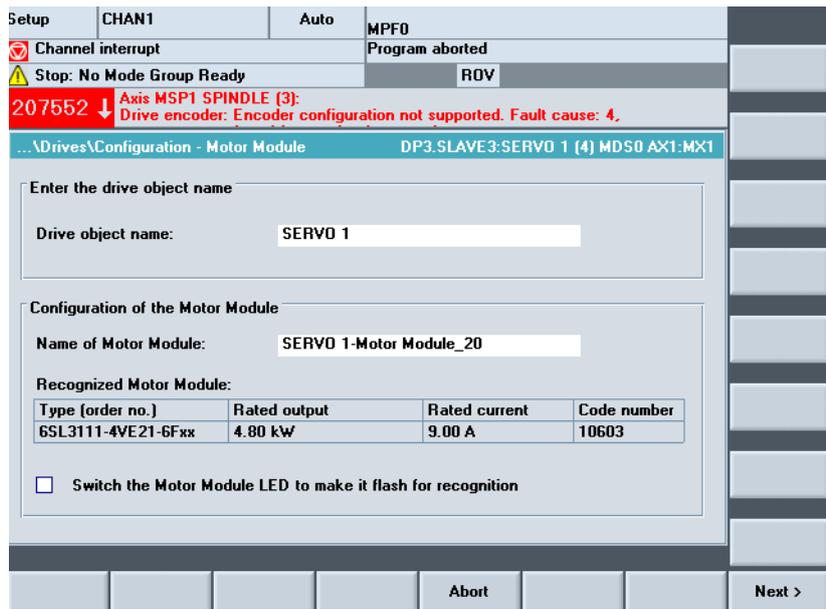


图 6-19 驱动配置 - 电机模块

3. 选择“继续 >”，配置相应的电机：在系统自动检测拓扑结构时已经指定了正确的电机型号。

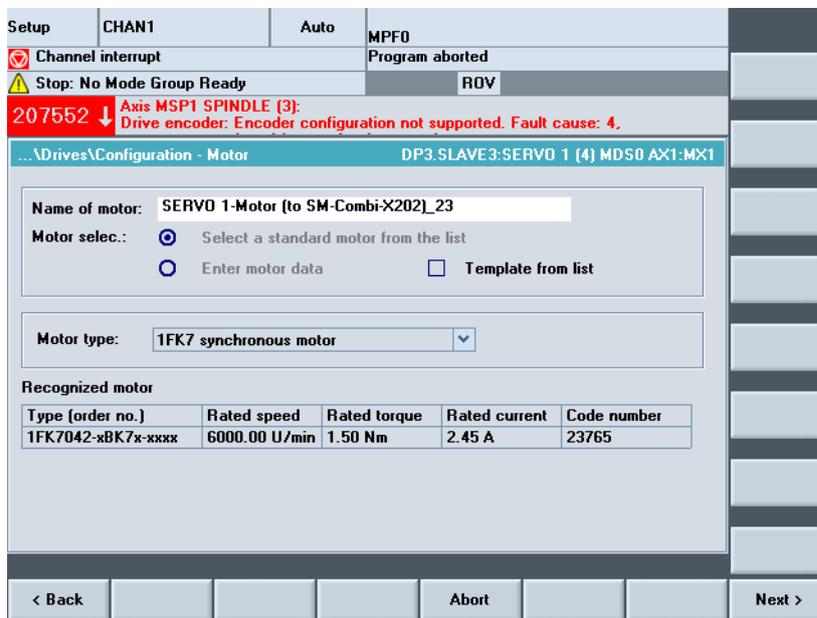


图 6-20 驱动\配置 - 电机

4. 选择“继续>”，配置电机抱闸：

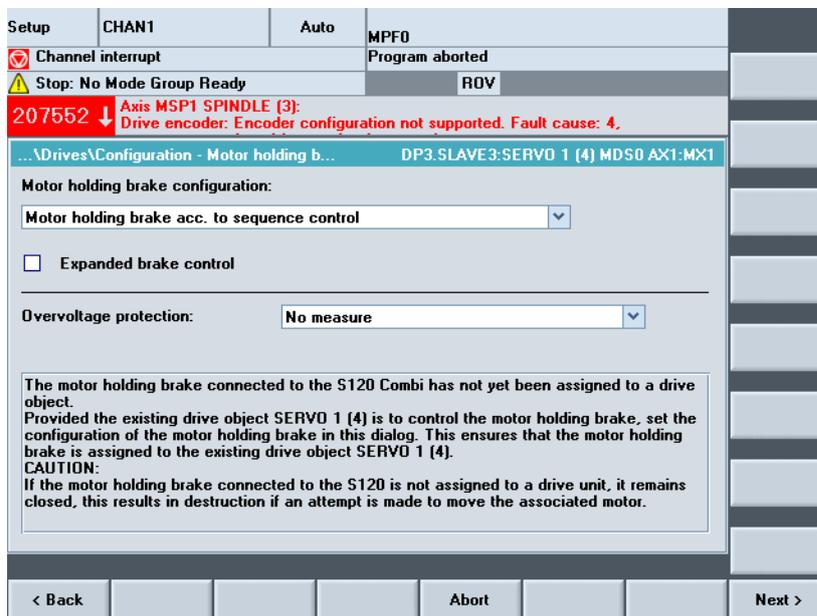


图 6-21 驱动\配置 - 电机抱闸

请注意关于过压防护的说明。过压防护选项有：

- 无措施
- VPM（电压保护模式）

- 选择“继续>”，选定编码器：在自动检测拓扑结构时系统已经正确指定了两个编码器（电机上的测量系统和机床上的直接测量系统）。

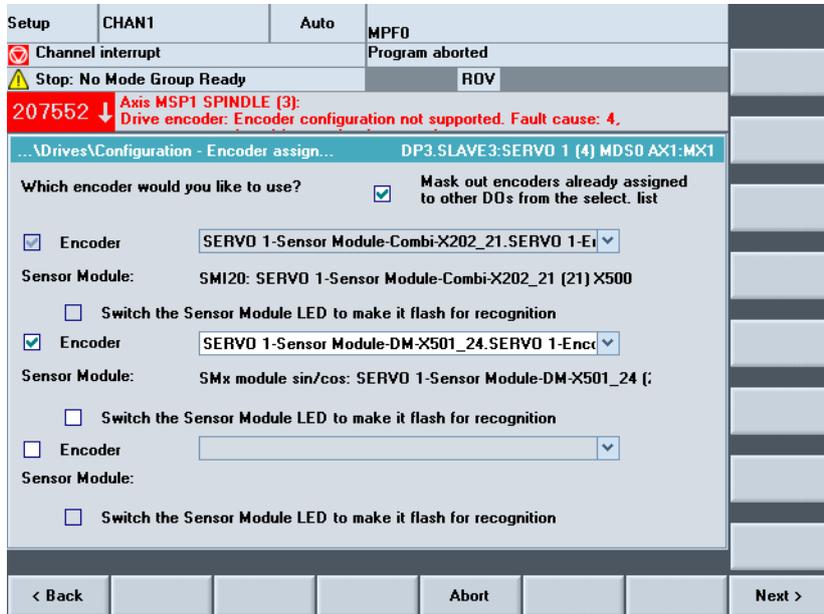


图 6-22 驱动配置 - 编码器指定

- 选择“继续 >”，配置电机编码器：

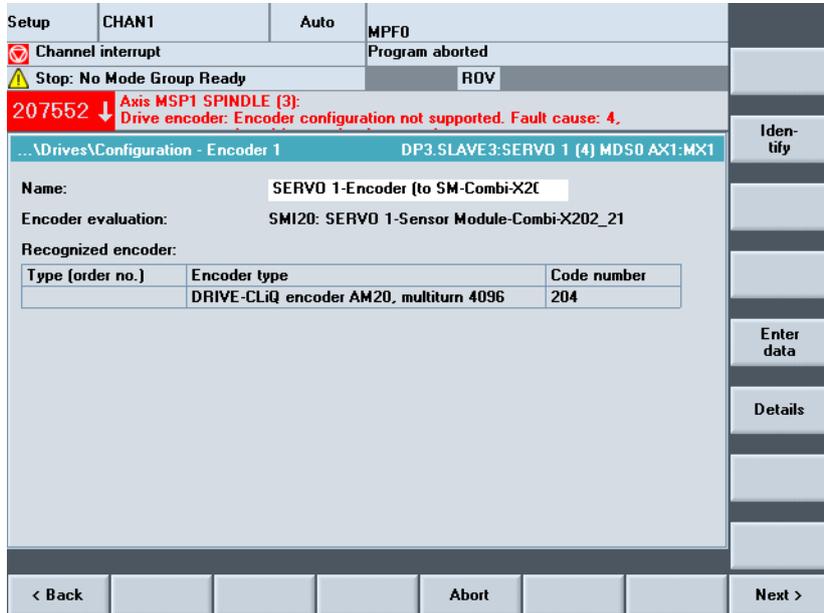


图 6-23 驱动配置 - 编码器 1

- 选择“输入数据”，使编码器属性和“取反位置实际值”相配套。

8. 如果点击“确认”来确认设置，编码器将显示为“用户定义”：

Type (order no.)	Encoder type	Code number
	4096, HTL, A/B, SSI, Singleturn	3090
6SL3055-0AA00-5...	2000 nm, TTL, A/B R distance-coded	3109
6SL3055-0AA00-5...	User-defined	9999
	Digital encoder (absolute) identified	10058
	Digital encoder (incremental) identified	10059

图 6-24 驱动\配置 - 编码器 2

9. 选择“继续 >”，采用报文 116（转速前馈）的缺省设置。现在系统会传送用于内部驱动参数的附加过程数据(PZD)，并将这些数据保存在系统变量中。

图 6-25 驱动\配置 - 控制方式

-或者-

10. 选择报文 136（转矩前馈），使系统传送用于内部驱动参数的附加过程数据(PZD)，并将这些数据保存在系统变量中。

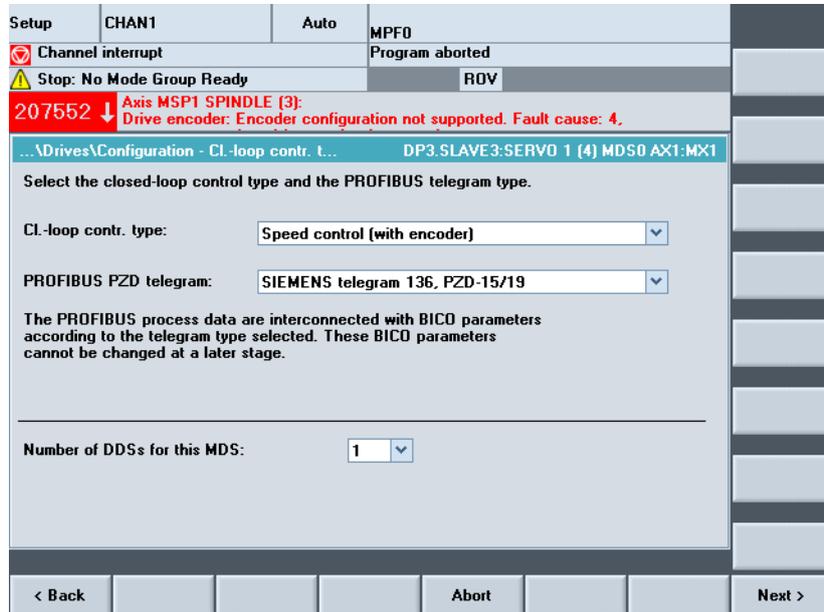


图 6-26 驱动配置 - 控制方式

11. 选择“继续 >”，进行 BICO 配置。

随后系统会显示以下配置一览，方明您再次检查轴的所有数据：

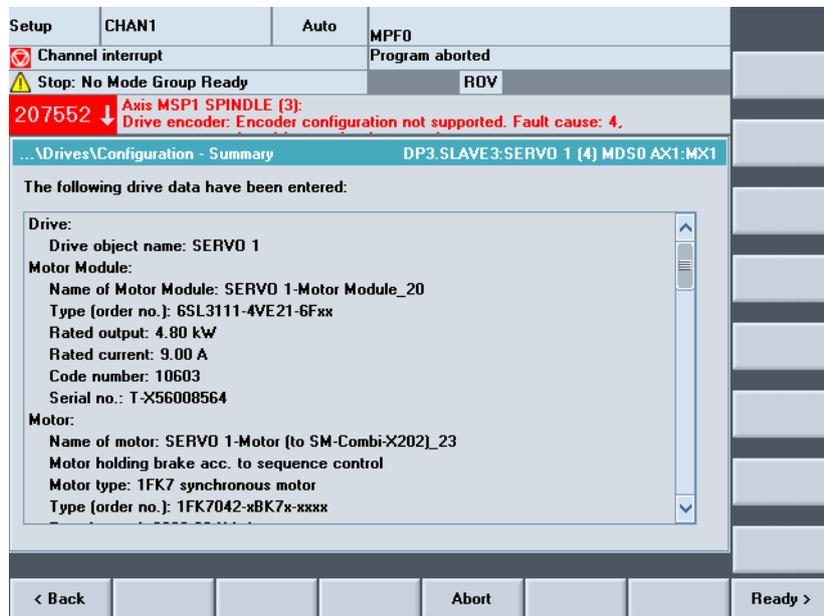


图 6-27 驱动配置 - 总结

12. 选择“完成 >”结束轴配置。

13. 选择“是”保存配置。该过程可能要持续几分钟。

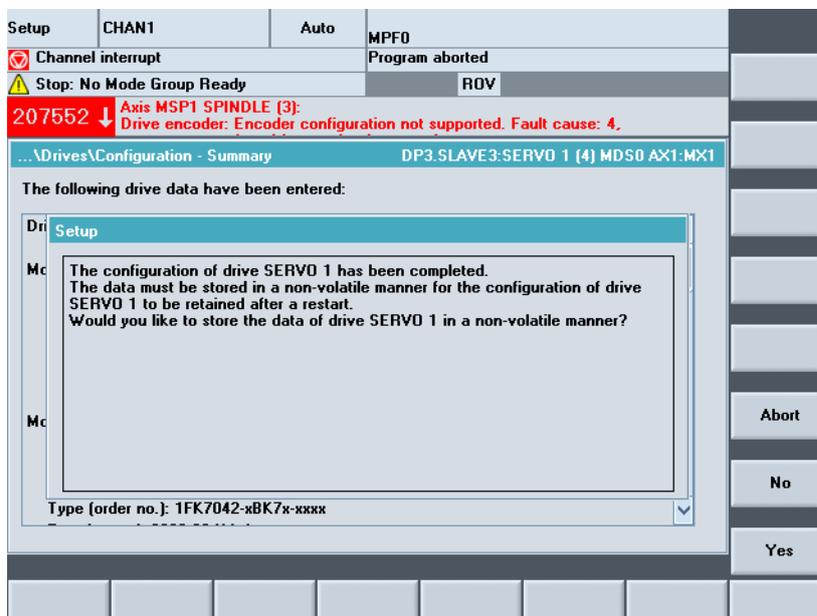


图 6-28 调试 - 询问

结果

轴的配置现在成功结束。

6.1.4 示例： SINAMICS S120 书本型的配置

概述

Toolbox CD 光盘中含有免费的 SINAMICS S120 调试软件。

在系统的操作界面上可以完全使用 SINAMICS S120 调试功能之前，驱动的调试还需要借助 SINAMICS S120 调试软件。PG/PC 通过 SINUMERIK 828D 前面的以太网接口接入。

配置驱动

驱动配置可以参照“系统概述 (页 11)”中关于 4 轴以及 SINAMICS S120 书本型的示例。下图展示了 DRIVE-CLiQ 的连接方式：

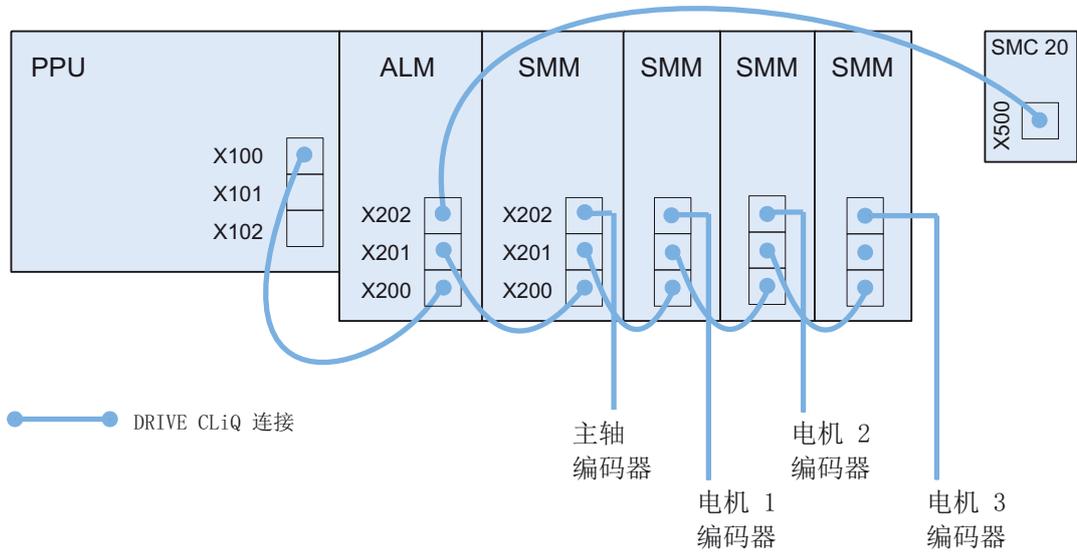


图 6-29 DRIVE-CLiQ 连接

在 4 轴配置中：

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。只有当该设置不适用于驱动组中的顺序时，才需要调整设置。

索引	轴	SINAMICS 驱动对象	
		编号	名称
		1	CU_I_3.3:1
		2	SLM_3.3:2
4	MSP1	3	SERVO_3.3:3
1	MX1	4	SERVO_3.3:4
2	MY1	5	SERVO_3.3:5
3	MZ1	6	SERVO_3.3:6

配置过程

配置主要分为以下几个步骤：

- 建立和控制系统的连接。
- 步骤 1：配置驱动系统。
- 步骤 2：配置电源。
- 步骤 3：配置驱动对象。
- 步骤 4：指定编码器。
- 步骤 5：指定轴。
- 最后：保存数据。

下面详细说明各个步骤的具体操作。

另见

S120 书本型的拓扑规则 (页 174)

6.1.5 示例：配置驱动系统的步骤

初始状态

在开始前：

- 将 PG/PC 和控制系统连在一起：参见章节示例：通过 NCU Connection Wizard 建立和控制系统的通讯的步骤 (页 29)
- 在“Siemens default data”模式下启动控制系统。
- 控制系统的操作区域“调试”中显示如下画面：

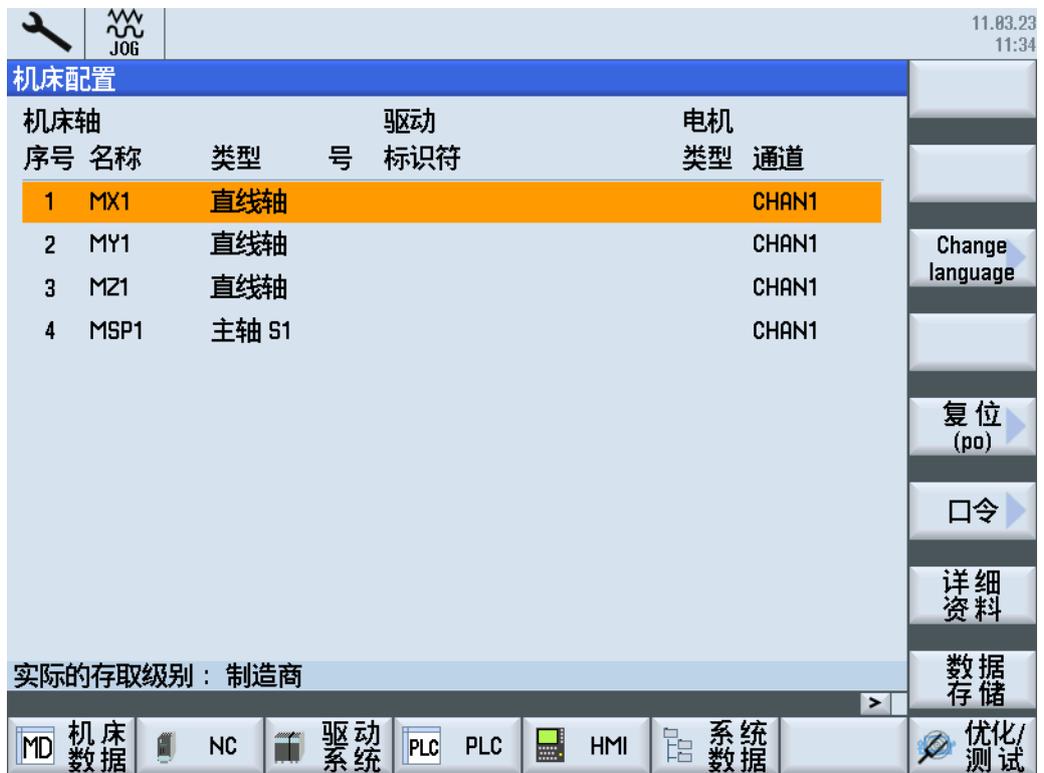


图 6-30 启动后的机床配置

操作步骤

配置驱动的步骤：

1. 在 PG/PC 上启动 SINAMICS S120 调试软件：



2. 选择操作区域“调试”。
3. 使用存取级别“制造商”的口令。然后会出现以下画面：



图 6-31 机床配置：开始调试

说明

更新固件

如果连接了驱动，固件会自动更新。

- 在载入固件期间，会出现一个状态条，显示载入进度。
- 同时，正在载入固件的模块上的“RDY”LED 灯会以红色-绿色交替闪烁。结束载入后，灯会变为绿色。“DC LINK”LED 灯则一直为“橙色”。

4. 成功载入固件后，按下“确认”应答此提示信息。

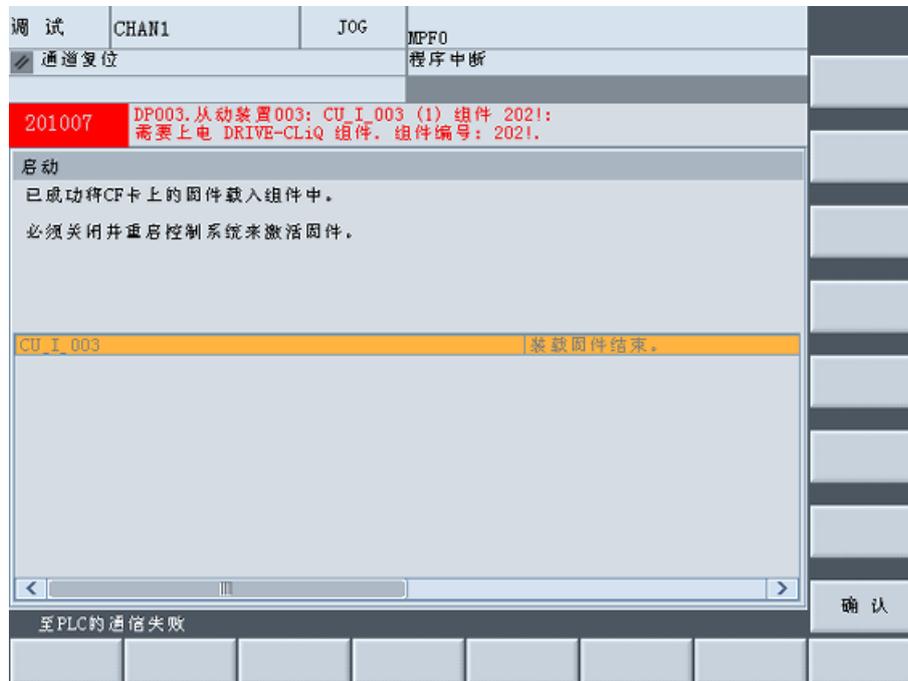


图 6-32 调试：固件装载

5. 按照提示信息的要求，重新给驱动上电：完全关闭驱动，并重新启动。

下面的选项取决于目标系统的配置：

- 没有 NX 扩展模块：软键“驱动设备”
- 有 NX 扩展模块：软键“驱动系统”

本示例展示的是没有 NX 扩展模块的拓扑结构。

6. 选择垂直软键“驱动设备”，点击“确认”继续配置。

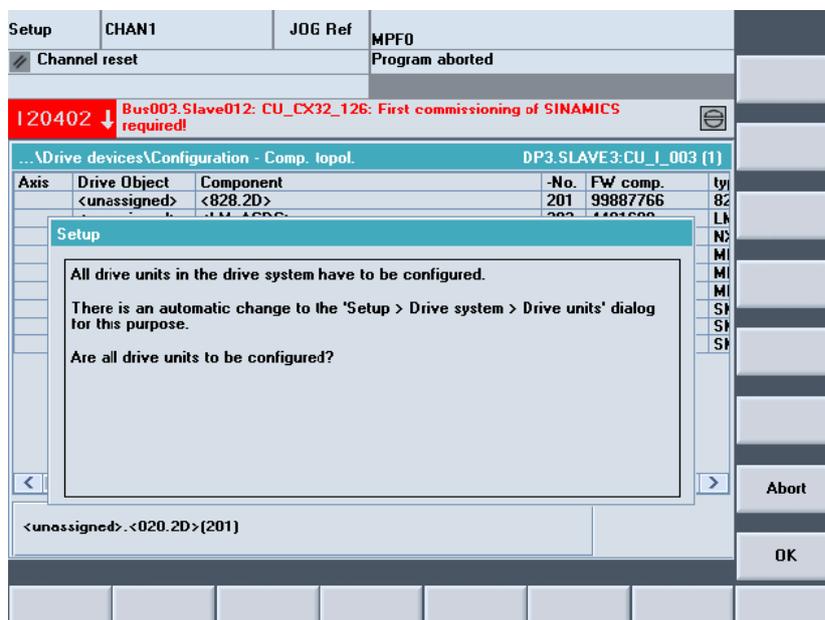


图 6-33 启动设备\配置 - 拓扑结构比较

7. 随后会跳出一条信息，提示驱动的配置过程可能需要持续好几分钟。

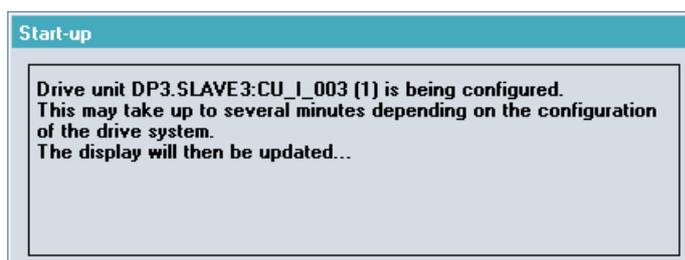


图 6-34 提示配置的信息

8. 然后会出现以下画面：

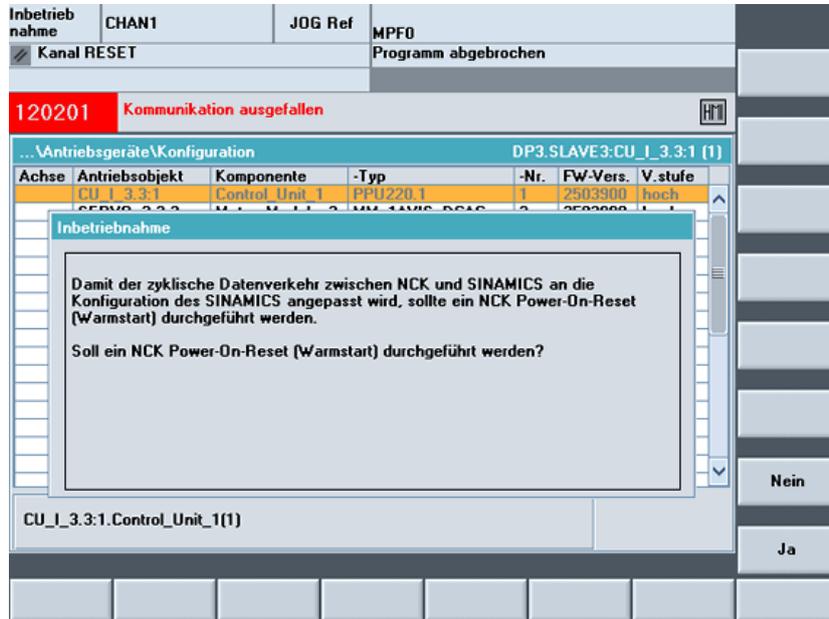


图 6-35 驱动设备\配置： NCK 上电复位

9. 点击“是”确认。系统热启动。该过程可能要持续几分钟。热启动结束后可以选择：

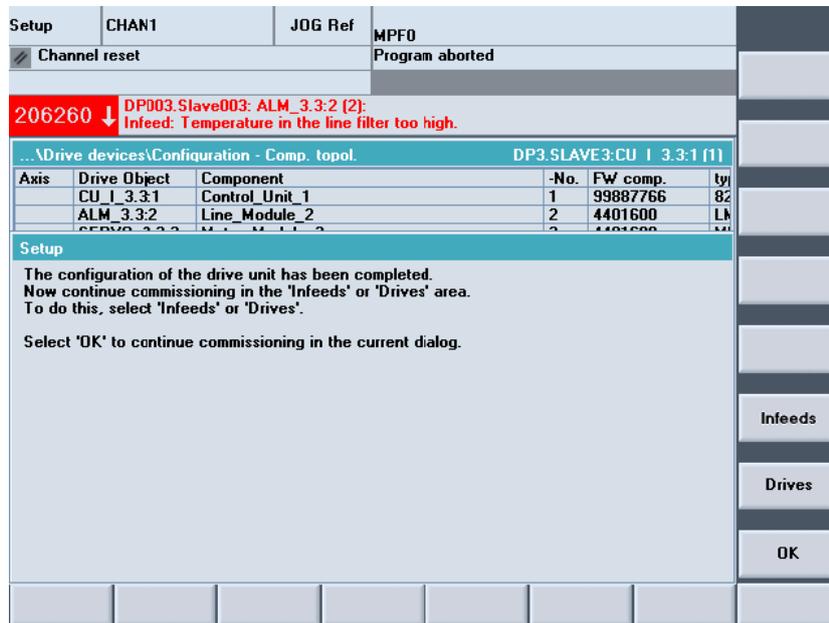


图 6-36 驱动设备\配置： 继续调试

10. 要继续调试，点击软键“驱动”。

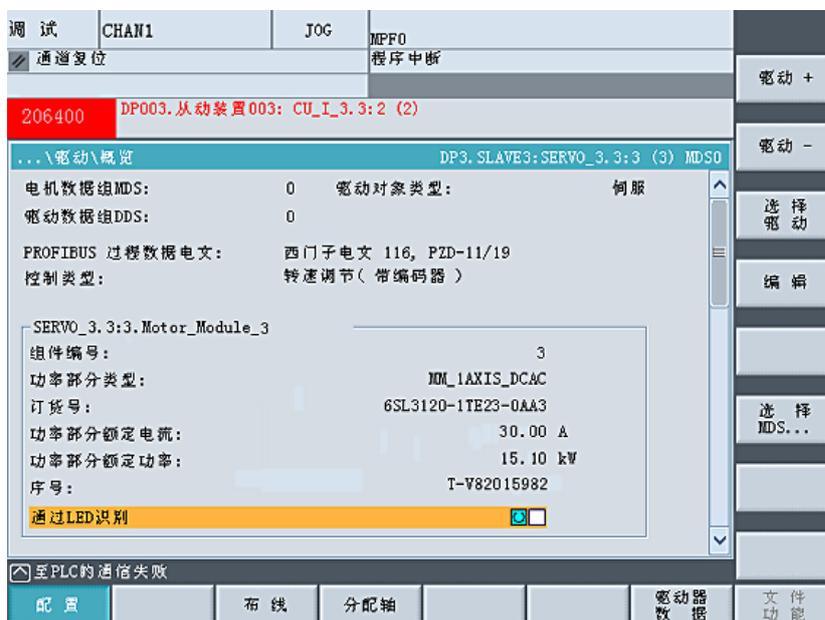


图 6-37 驱动一览

11. 选项：需要检测单个电机模块时，选择“通过 LED 识别”：“RDY”LED 灯以红色-橙色交替闪烁。

按下软键“驱动 +”和“驱动 -”选择下一个模块。

结果

第 1 步“驱动调试”结束。

6.1.6 示例：配置电源的步骤

配置供电

说明

如果在接通系统后出现需要应答的报警，必须首先应答该报警。然后再继续调试。

操作步骤：

1. 按下水平软键“供电”，显示以下一览：

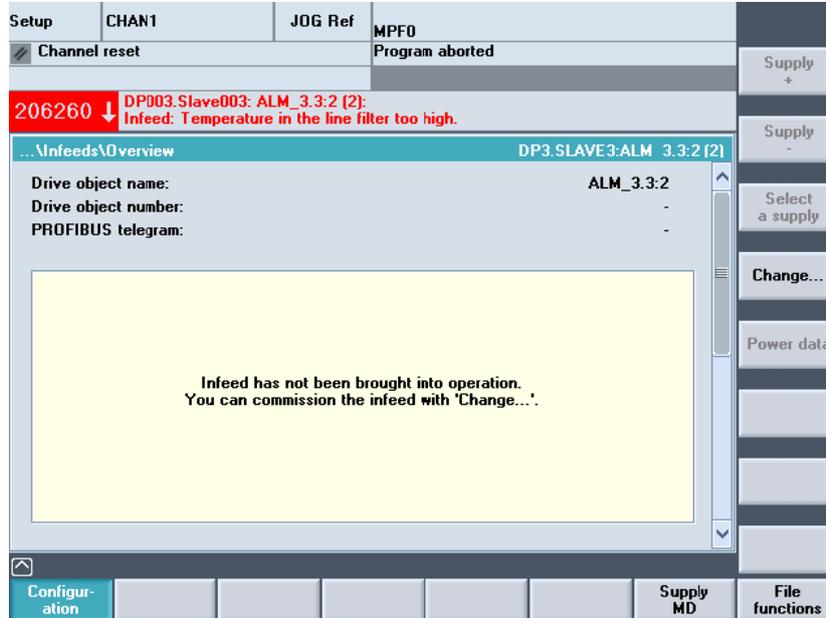


图 6-38 供电\一览

2. 按下垂直软键“更改”，配置供电。在下面的对话框中为供电命名或者直接采用缺省名称：

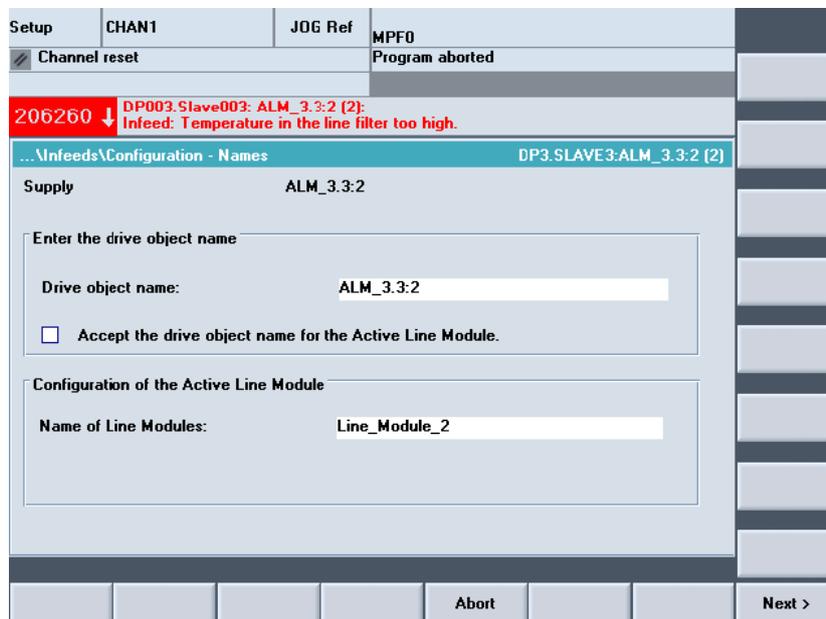


图 6-39 供电\配置 - 名称

3. 在下面的对话框“配置 - 电源模块”中会显示自动检测出的电源模块和相连的输入滤波器。按下“继续>”以继续调试；如果有制动模块，也可选择选项“外部制动模块”

Type [order no.]	Rated output	Rated current	Code number
6SL3130-7TE21-6Axx	16 kW	27 A	10015

图 6-40 供电\配置 - 电源模块

4. 在下面的对话框“配置 - 更多数据”中，系统会自动识别电源/直流母线，并保存识别出的数据。按下“继续>”以继续调试；也可选择设备输入电压和报文类型。

图 6-41 供电\配置 - 更多数据

5. 在下面的对话框“配置 - 端子接线”中选择 SINAMICS 内部电源接触器控制，或者按下“继续>”以继续调试。

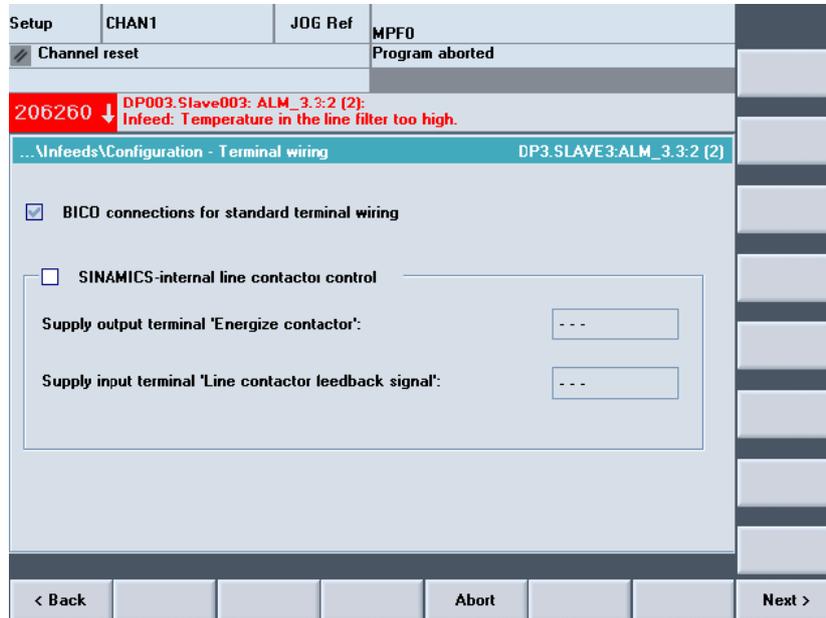


图 6-42 供电配置 - 端子接线

6. 在下面的对话框“配置 - 摘要”中显示电源数据一览。按下“完成”结束配置，或按下“返回”修改设置。

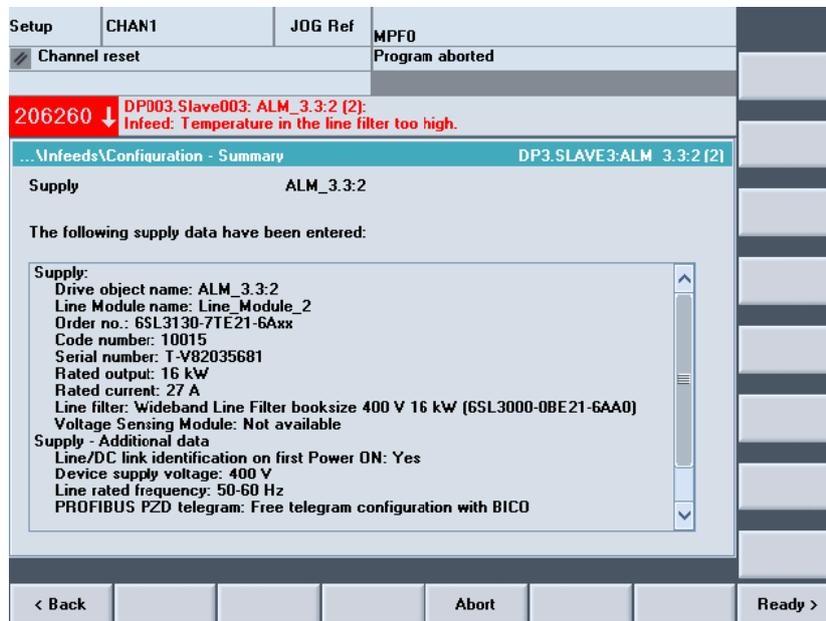


图 6-43 供电配置 - 摘要

7. 按下“是”，保存数据。该过程可能要持续几分钟。

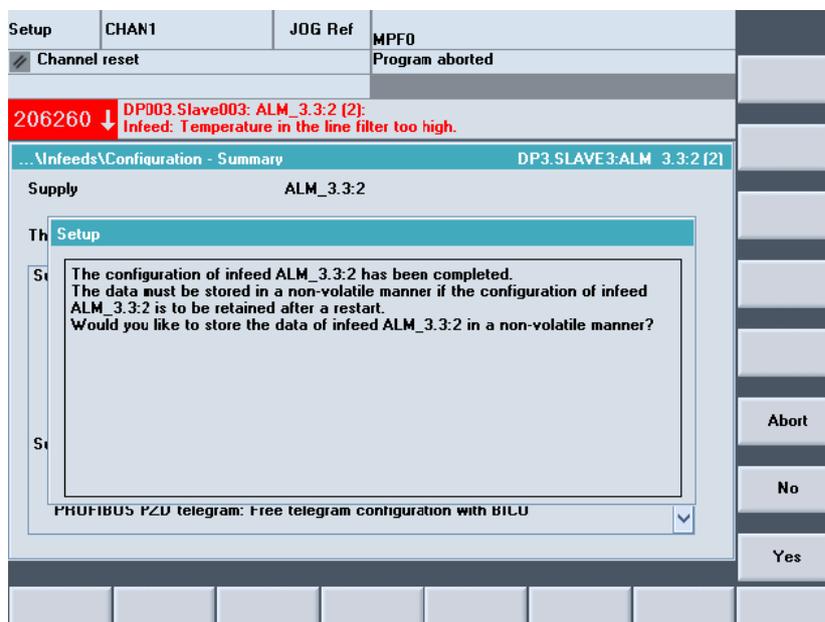


图 6-44 供电\配置 - 摘要： 询问

8. 在下面的对话框“一览”中简要显示了驱动对象的配置数据：

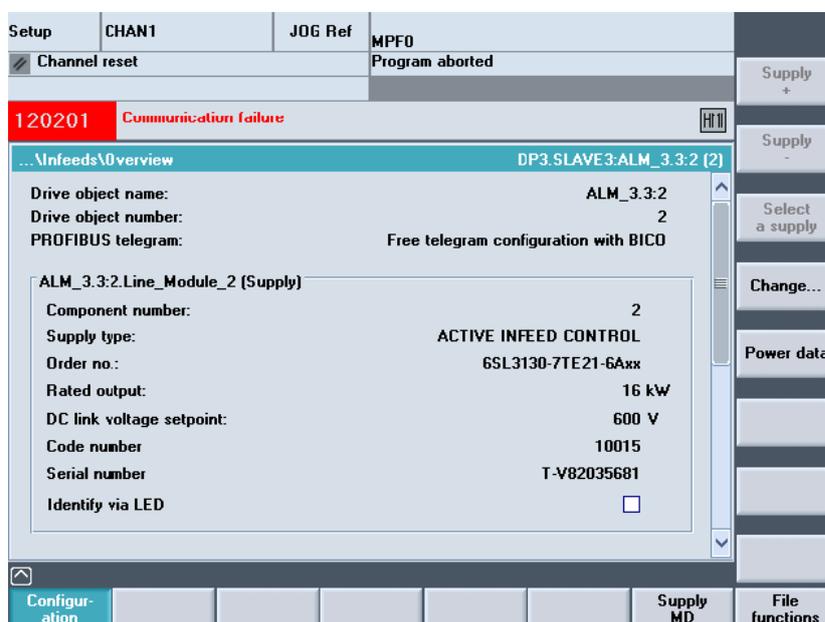


图 6-45 供电\一览

结果

- 第 2 步“供电调试”结束。

6.1.7 示例：配置驱动对象的步骤

初始状态

在“驱动调试”一览中会显示以下驱动对象(DO):

The screenshot shows a software interface for drive commissioning. At the top, there are status indicators: 'Setup' (CHAN1), 'JOG Ref', and 'MPFO'. Below this, a 'Channel reset' button is visible, along with a 'Program aborted' message and a 'ROV' indicator. The main section is titled 'Drive commissioning' and contains a table of drive objects. The table has columns for 'Axis', 'Bus', 'Address', 'Drive Object -Name', and '-Number'. The first row is highlighted in orange, showing '3' for Axis, '3' for Bus, '3' for Address, 'CU_I_3.3:1' for the name, and '1' for the number. Other rows include 'ALM_3.3:2' (2), 'SERVO_3.3:3' (3), 'SERVO_3.3:4' (4), 'SERVO_3.3:5' (5), and a group for Bus 12: 'CU_NX_3.12:1' (1), 'SERVO_3.12:2' (2), and 'SERVO_3.12:3' (3). To the right of the table are several buttons: 'Factory setting...', 'Reset (p)', 'Load firm-ware...', 'Assign axis', and 'Identify via LED'. At the bottom, there are tabs for 'Drive devices', 'Infeeds', and 'Drives'.

Axis	Bus	Address	Drive Object -Name	-Number
3	3	3	CU_I_3.3:1	1
			ALM_3.3:2	2
			SERVO_3.3:3	3
			SERVO_3.3:4	4
			SERVO_3.3:5	5
	3	12	CU_NX_3.12:1	1
			SERVO_3.12:2	2
			SERVO_3.12:3	3

驱动调试

操作步骤

配置驱动对象的步骤:

1. 用光标键选择一个驱动对象，按下软键“驱动”。如果该驱动对象还没有经过调试，请选择“更改”。

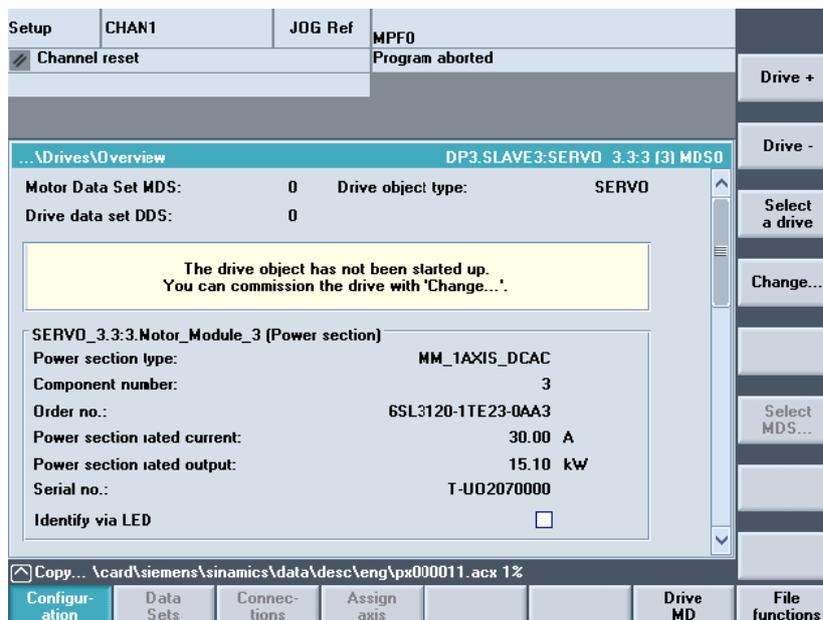


图 6-46 驱动\一览：开始调试

2. 在下面的对话框“配置 - 电机模块”中为电机模块命名或者直接采用缺省名称:

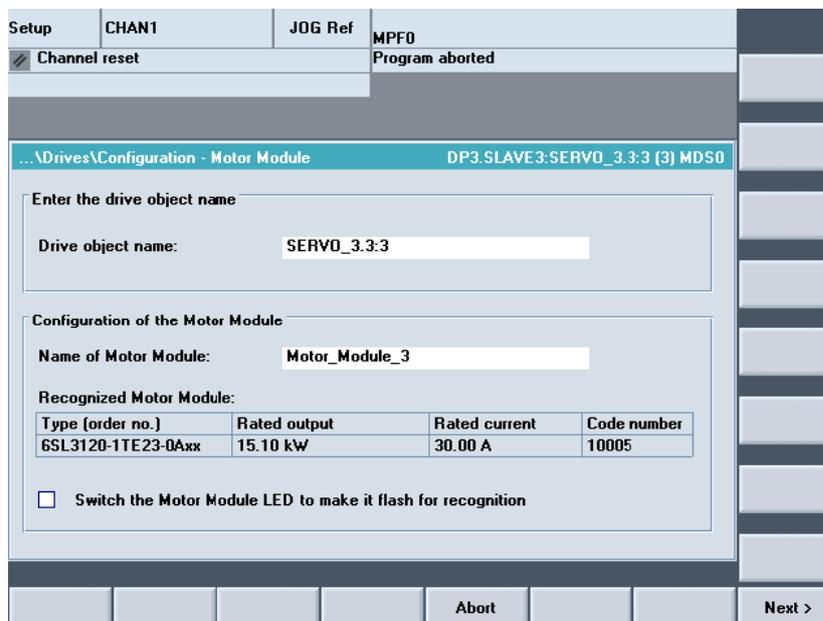


图 6-47 驱动\配置 - 电机模块

3. 在下面的对话框“配置 - 电机”中从电机列表选择一个型号。按下软键“继续>”继续调试。

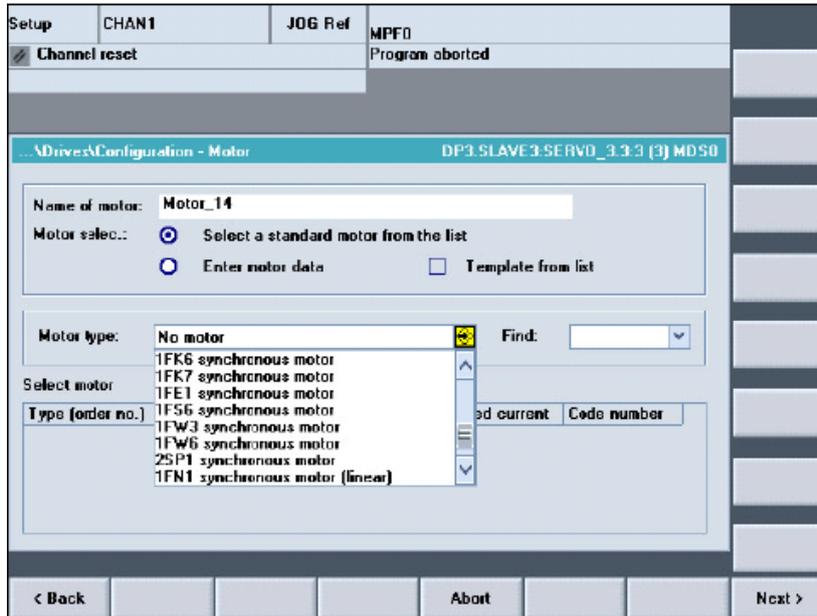


图 6-48 驱动配置 - 电机

4. 在下面的对话框“配置 - 电机抱闸”中选择一个电机抱闸（如果有），并选择扩展的抱闸控制。按下软键“继续>”继续调试。

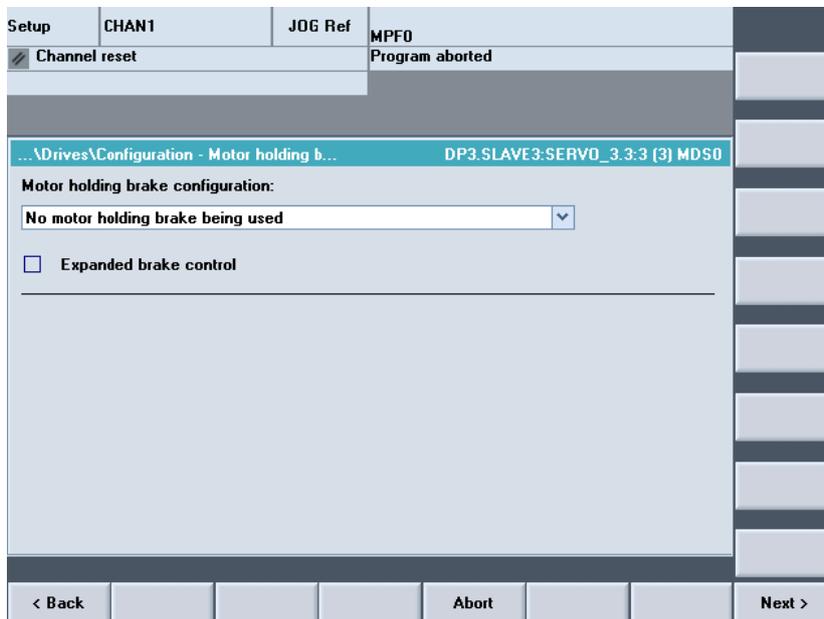


图 6-49 驱动配置 - 电机抱闸

5. 在下面的对话框“配置 - 编码器指定”中，从驱动系统现有的编码器中选择一个电机上连接的编码器。按下软键“继续>”继续调试。

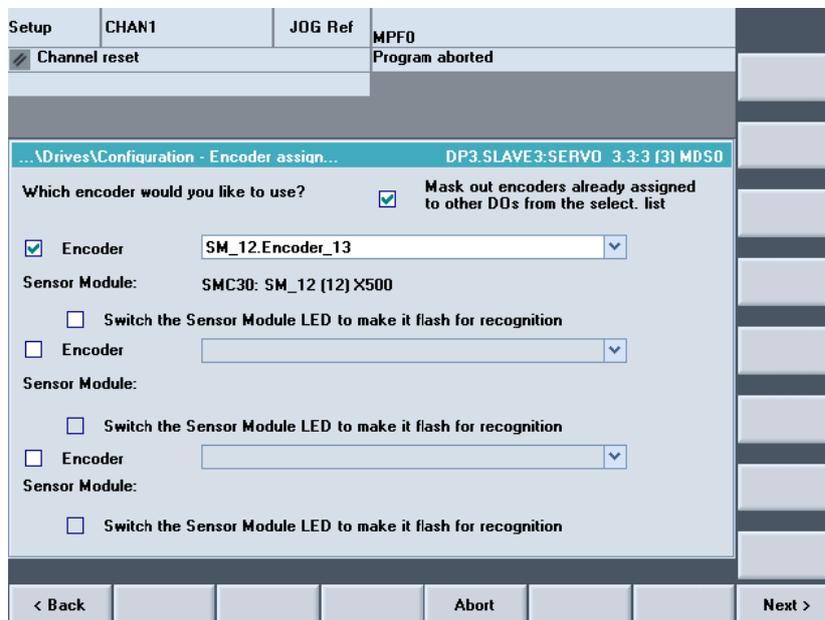


图 6-50 驱动\配置 - 编码器指定

6.1 配置驱动

6. 在下面的对话框“配置 - 编码器 1”中，在“选择电机编码器”下选择编码器类型。
 - 如果下拉菜单中没有提供所需编码器，请选择“输入数据”。
 - 选择“详细信息”输入编码器的配置数据。

按下软键“继续>”继续调试。

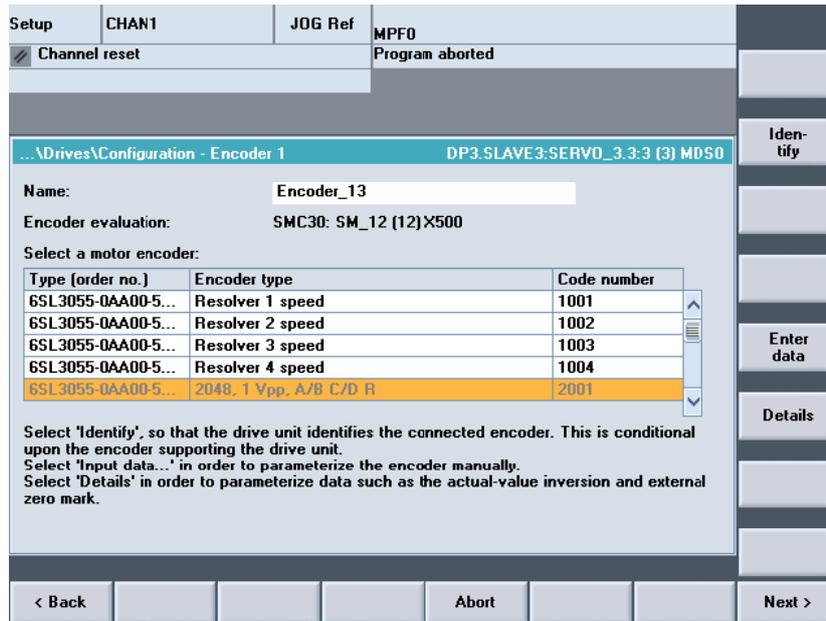


图 6-51 驱动配置 - 编码器 1

7. 在下面的对话框“配置 - 控制方式”中选择带或者不带编码器的控制方式以及 PROFIBUS 报文。控制方式和 PROFIBUS 报文通常已由系统正确指定。

另外，您还可以在此选择该电机数据组(MDS)所属驱动数据组(DDS)的数量。缺省设置是 1 个 DDS；每个 MDS 最多允许 8 个 DDS。按下软键“继续>”继续调试。

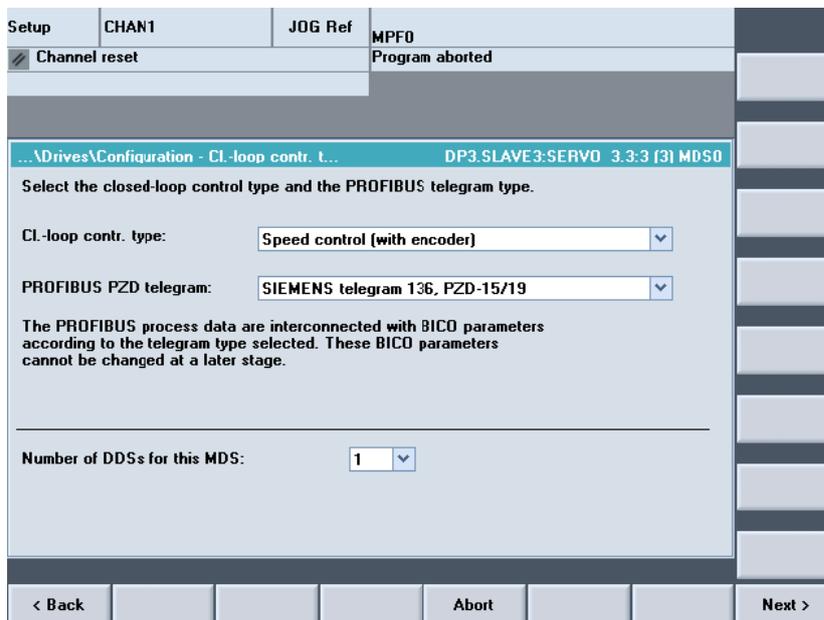


图 6-52 驱动\配置 - 控制方式

8. 在下面的对话框“配置 - BICO 互联”中选择 OFF2 的 BICO 互联。系统会根据所选报文预设该互联。

按下软键“继续>”继续调试。

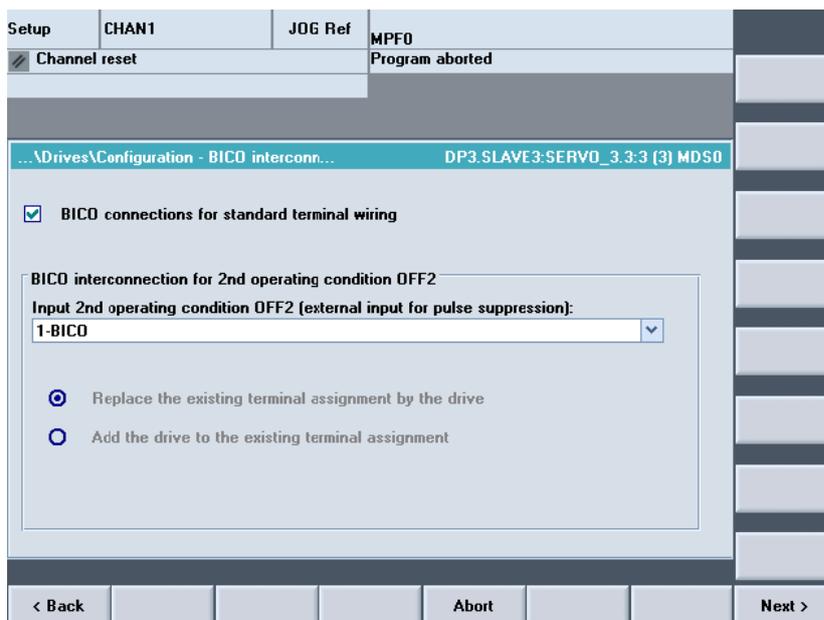


图 6-53 驱动\配置 - BICO 互联

9. 在下面的对话框“配置 - 总结”中显示了为电机模块输入的数据一览。按下“完成>”结束调试。

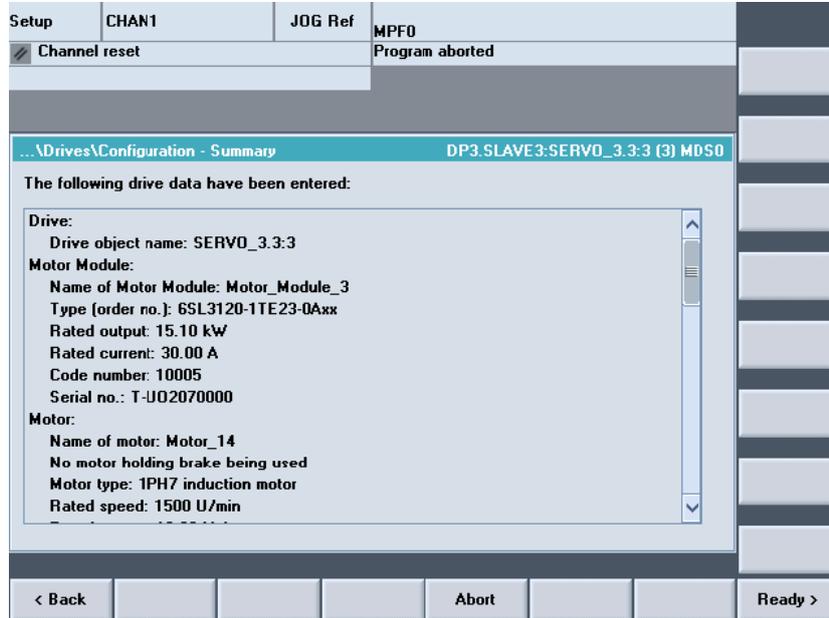


图 6-54 驱动配置 - 总结

10. 在下面的询问框中选择是否保存驱动数据：按下“是>”来保存驱动配置。

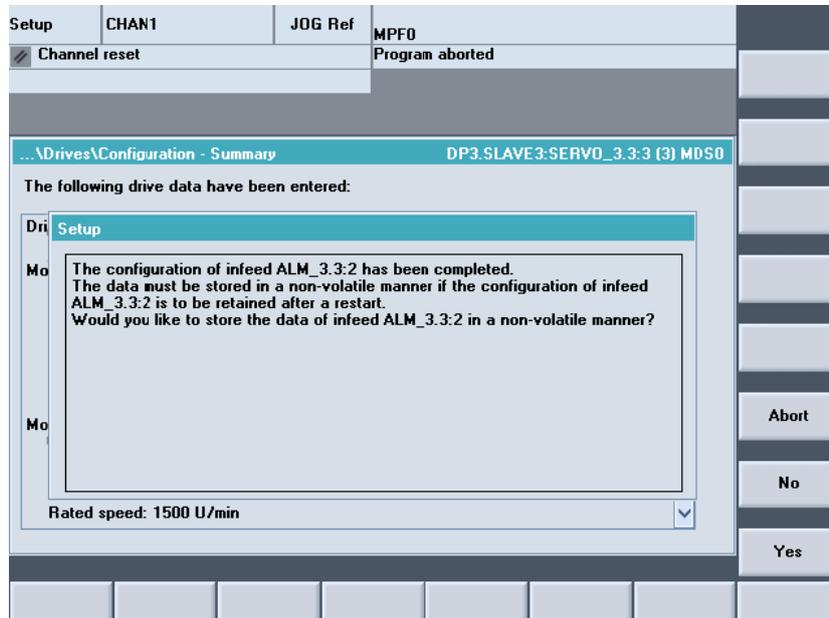


图 6-55 调试 - 询问

结果

步骤 3 “调试驱动对象”结束。为所有其他驱动对象重复上述步骤。

6.1.8 示例：配置外部编码器的步骤

直接测量系统的连接

在章节示例：SINAMICS S120 书本型的配置 (页 123) 的示例中，还另外连接了一个直接测量系统用于主轴。下面介绍如何配置该编码器。

操作步骤

更改驱动配置：

1. 按下垂直软键“更改”。

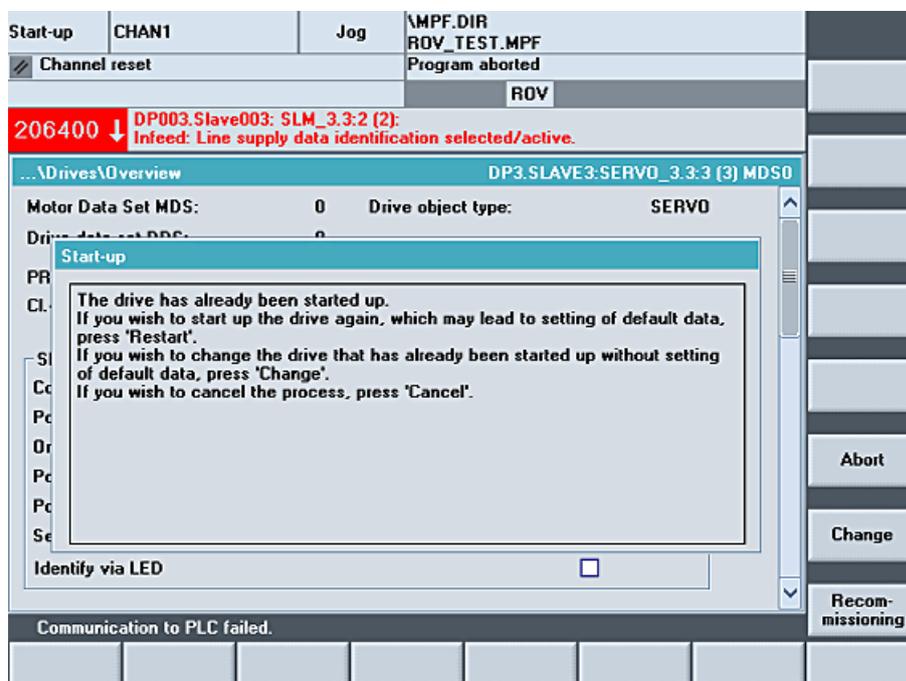


图 6-56 更改驱动

2. 按下软键“更改”确认询问。
窗口“配置 - 电机”打开，选择驱动对象 3（即主轴）。按下“继续 >”。
3. 在随后出现的窗口中会显示，哪些电机已经指定给了哪些驱动。
按下“继续 >”。

6.1 配置驱动

4. 随后出现的窗口显示电机指定情况。按下“继续 >”。

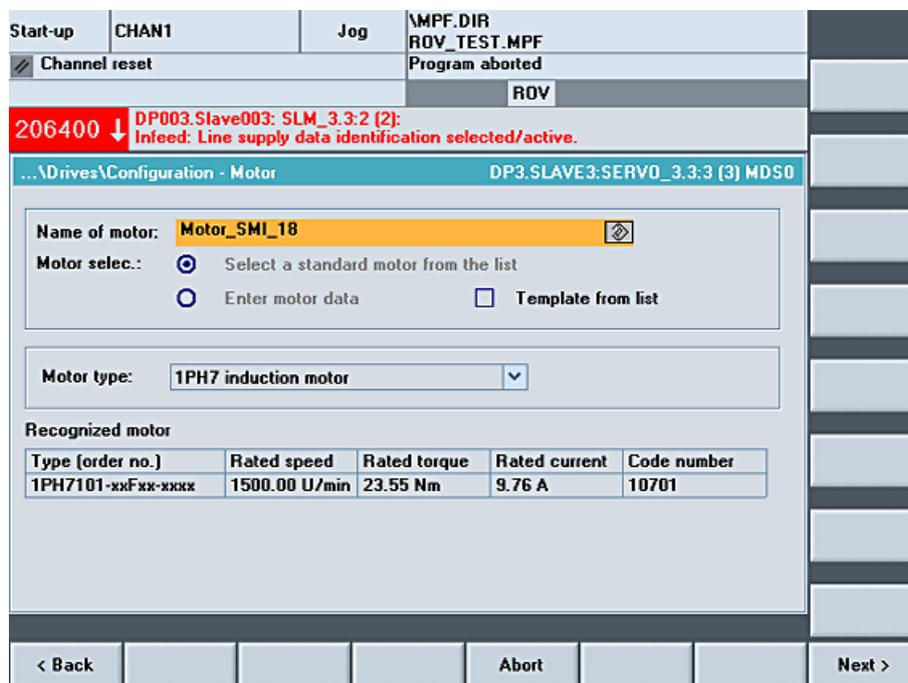


图 6-57 配置电机

5. 该窗口显示系统检测出的电机的详细数据。

按下“继续 >”。

6. 随后出现的窗口显示电机抱闸的配置。

按下“继续 >”。

7. 在随后出现的窗口中会显示，哪些编码器已经指定给了该驱动对象（即主轴）：

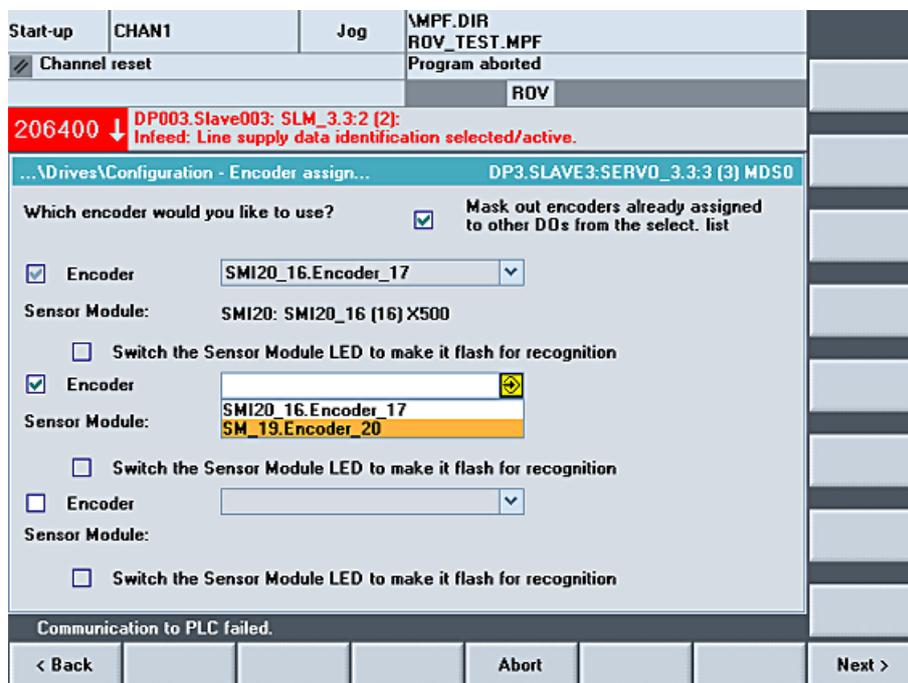


图 6-58 指定编码器

8. 选中选项“编码器 2”。
9. 在下拉表中选择编码器“SM_19.Encoder_20”。
10. 按下 <INPUT> 键。
11. 按下“继续 >”。
12. 按下“确认”应答询问。

数据保存可能要持续好几分钟。

13. 在保存数据时，会跳出以下状态条：

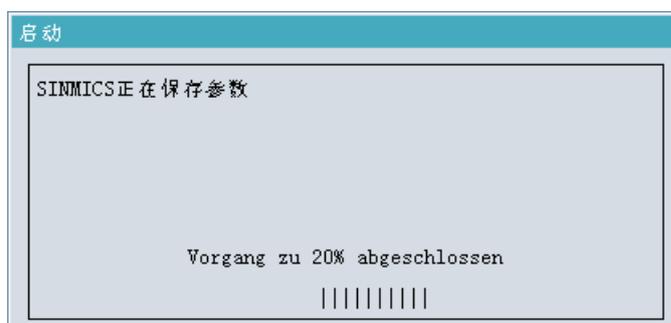


图 6-59 编码器 2：保存数据

14. 随后出现的窗口显示电机抱闸的配置。
15. 按下“继续 >”。

6.1 配置驱动

16. 随后出现的窗口显示 BICO 互联。
17. 按下“继续 >”。
18. 最后会显示一个包含所有数据的总结。
19. 按下“完成 >”后，跳出以下信息：

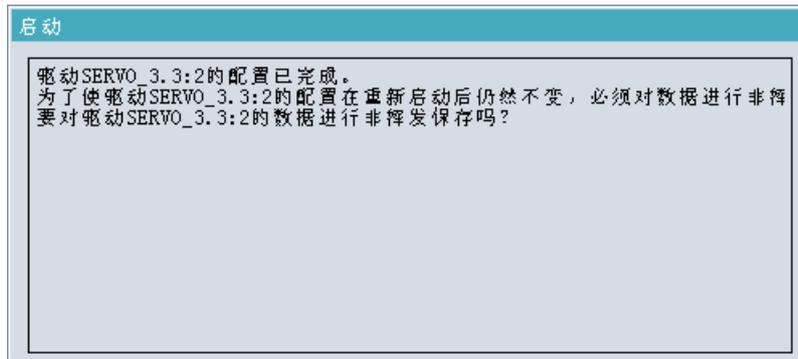


图 6-60 确认保存

20. 按下“是”保存配置数据。
保存可能要持续几分钟。

结果

步骤 4 “调试直接测量系统”结束。

6.1.9 在基本调试后更换驱动组件

组件更换

基本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订货号的组件，例如替换为功率更大的模块，请注意以下操作过程：

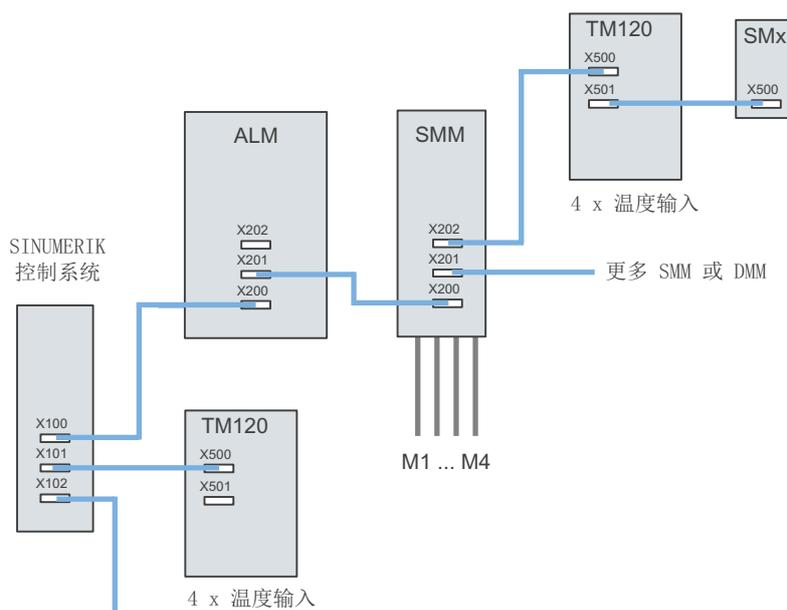
操作步骤:

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

6.1.10 示例：包含 TM120 的并联

应用：4 电机并联

示例拓扑:



- 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。

TM120 上的措施：

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。

6.1 配置驱动

6.1.11 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：

- 驱动系统的输入/输出地址
- 控制单元的输入/输出地址
- 报文

将 PLC 项目中设置的输入/输出地址输入到以下的机床数据中：

机床数据		无 NX 模块
13120[0]	\$MN_CONTROL_UNIT_LOGIC_ADRESS	= 6500
13120[1]	\$MN_CONTROL_UNIT_LOGIC_ADRESS	= 0

机床数据		有 NX 模块
13120[0]	\$MN_CONTROL_UNIT_LOGIC_ADRESS	= 6500
13120[1]	\$MN_CONTROL_UNIT_LOGIC_ADRESS	= 6516

PROFIBUS 报文类型

在预设置的 SDB 中将驱动报文预设为 PROFIBUS 报文类型 136（西门子报文类型如 116 以及转矩前馈）。调试时可设置组件 NCK (MD13060)和 SINAMICS S120 (p0922)的该报文类型。

所设置的系统数据块可通过 MD11240[0..3]选择：

MD11240	\$MN_PROFIBUS_SDB_NUMBER 所用的系统数据块(SDB)号，用于配置外设。
MD11240[0] = 0...n	PROFIBUS
MD11240[1] = -1	必须保留缺省设置。
MD11240[2] = 0...n	SINAMICS S120 集成的 PROFIBUS
MD11240[3] = 0...n	PROFINET

通过设定 MD11240[0..3] = n 可设置以下属性:

属性		PPU24x.2	PPU26x.2	PPU28x.2
总线周期 (系统周期)		1.5 ms	1.5 ms	1.5 ms
SERVO 周期		3 ms	3 ms	TE: 1.5 ms ME: 3 ms
n=0.1	CU-I 轴	6x 报文 136	6x 报文 136	6x 报文 136
	NX10 轴	3x 报文 136	3x 报文 136	3x 报文 136
n=3	CU-I 轴	6x 报文 116	6x 报文 116	6x 报文 116
	NX10 轴	3x 报文 116	3x 报文 116	3x 报文 116

文档

报文的说明请见以下文档:

- SINAMICS S120 功能手册/FH1/, 章节“通讯”
- SINAMICS S120 参数手册/LH1/ “功能图”

6.1.12 用于进给轴/主轴测试运行的参数

相关参数和端子

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

6.1 配置驱动

参数/端子	含义
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)

如下输入驱动参数： 操作区域“调试” > ”软键“驱动设备” > “输入/输出”

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

6.2 轴指定

6.2.1 示例：分配轴的步骤

分配轴

在保存了编码器 2 的配置数据后，会显示以下概览：

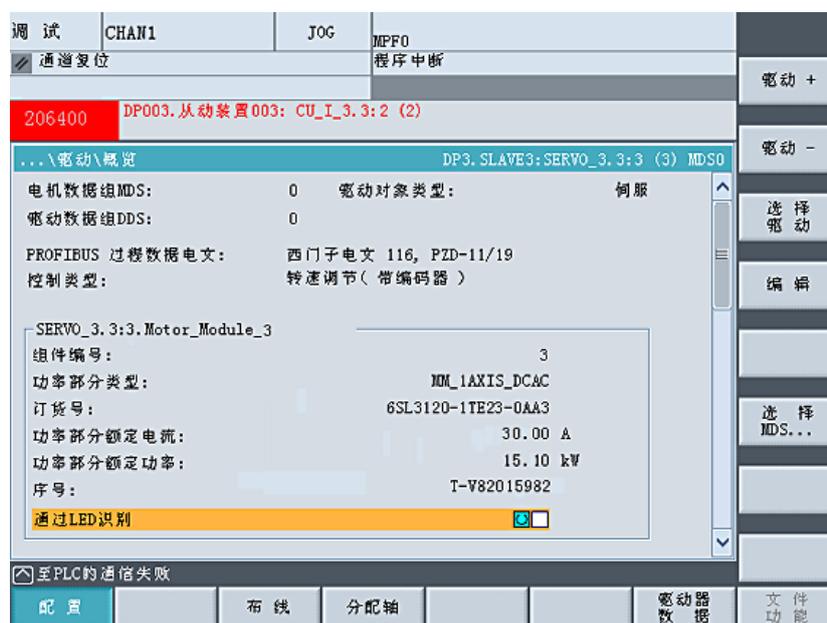


图 6-61 驱动一览

1. 按下水平软键“分配轴”，为每个逻辑驱动指定一个真实的轴。“分配轴”对话框打开：



图 6-62 轴分配

说明

下面的两项操作都必须各执行**两次**，也就是说：软键“更改”和“接收”各自按下两次。

2. 按下软键“更改”，为 DO 2 指定一根轴。
3. 从下拉表中选择“MSP1”，按下软键“接收”。
4. 必须对 NCK 执行上电复位，系统中的轴分配才能生效。

5. 按下软键“取消”，继续指定其他轴。



图 6-63 轴分配：主轴

6. 按下软键“驱动 +”和“驱动 -”选择下一个模块。

7. 依次指定所有轴：

轴	驱动
MSP1	SERVO_3.3:3
MX1	SERVO_3.3:4
MY1	SERVO_3.3:5
MZ1	SERVO_3.3:6

接收设置

最后对 NCK 执行上电复位，再次检查以下设置：

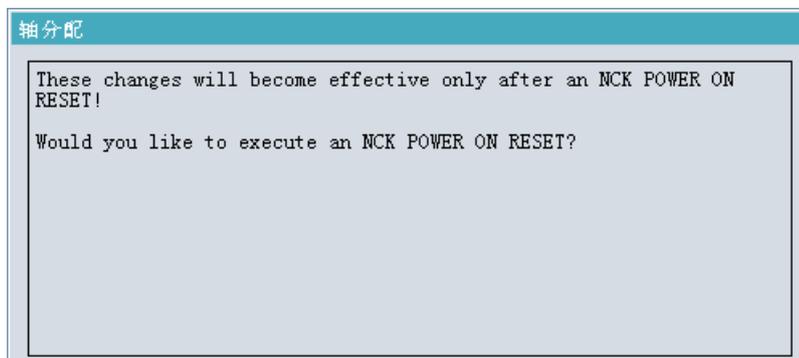


图 6-64 轴分配：开机重置

按下“确认”，重新启动驱动和控制系统。在 PG/PC 上显示以下轴分配：



图 6-65 机床配置

在重新启动后，控制系统上会显示以下机床配置：



图 6-66 机床配置：所有轴经过调试

结果

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 6-67 提示保存数据的信息

6.2.2 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
30200	\$MA_NUM_ENCS	1	1	1	1	1
30230	\$MA_ENC_INPUT_NR	1	1	1	2	1
30240	\$MA_ENC_TYPE	1	1	1	1	4
31020	\$MA_ENC_RESOL	2048	2048	2048	1024	512
34200	\$MA_ENC_REFP_MODE	1	1	1	1	0

主轴机床数据

请重新设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
32000	\$MA_MAX_AX_VELO	10000	3000
32010	\$MA_JOG_VELO_RAPID	10000	100
32020	\$MA_JOG_VELO	2000	50
35100	\$MA_SPIND_VELO_LIMIT	10000	3000
35110[0]	\$MA_GEAR_STEP_MAX_VELO[0]	500	3000
35110[1]	\$MA_GEAR_STEP_MAX_VELO[1]	500	3000
35130[0]	\$MA_GEAR_STEP_MAX_VELO_LIMIT[0]	500	3150
35130[1]	\$MA_GEAR_STEP_MAX_VELO_LIMIT[1]	500	3150
36200[0]	\$MA_AX_VELO_LIMIT[0]	11500	3300
36200[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

不允许同时设置 STIFFNESS_CONTROL_ENABLE=1 和 ENC_FEEDBACK_POL=-（编码器取反）。

编码器取反必须由 SINAMICS 驱动参数 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。

前提条件：报文 ≥ 116

SINAMICS 驱动参数：

p1192[0] 编码器选择

p1193[0] 编码器调整系数

6.3 配置数据组

6.3.1 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。

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)”

6.3.2 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）。

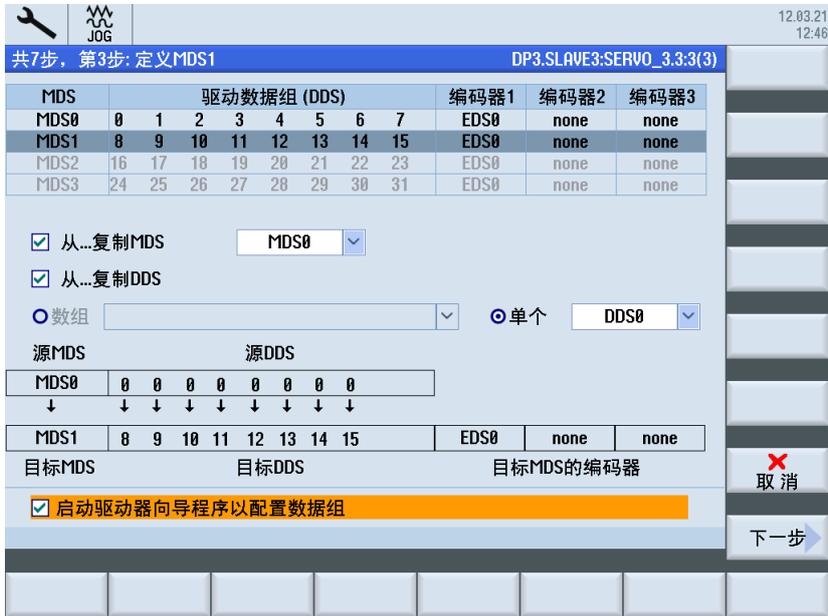


2. 按下“下一步 >”。

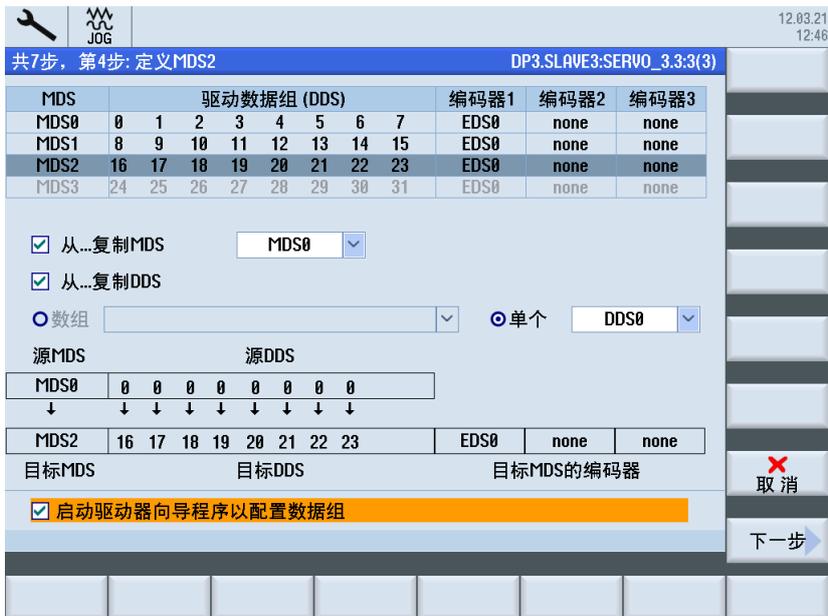
第一步在 MDS0 中创建驱动数据组 DDS1 至 DDS7。



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。

共7步, 第5步: 定义MDS3 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	1	2	3	4	5	6	7	EDS0	none	none
MDS1	8	9	10	11	12	13	14	15	EDS0	none	none
MDS2	16	17	18	19	20	21	22	23	EDS0	none	none
MDS3	24	25	26	27	28	29	30	31	EDS0	none	none

从...复制MDS MDS0
 从...复制DDS
 数组 单个 DDS0

源MDS 源DDS

MDS0	0	0	0	0	0	0	0	0
	↓	↓	↓	↓	↓	↓	↓	↓

MDS3 24 25 26 27 28 29 30 31 EDS0 none none

目标MDS 目标DDS 目标MDS的编码器

启动驱动器向导程序以配置数据组

取消 下一步 >

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。

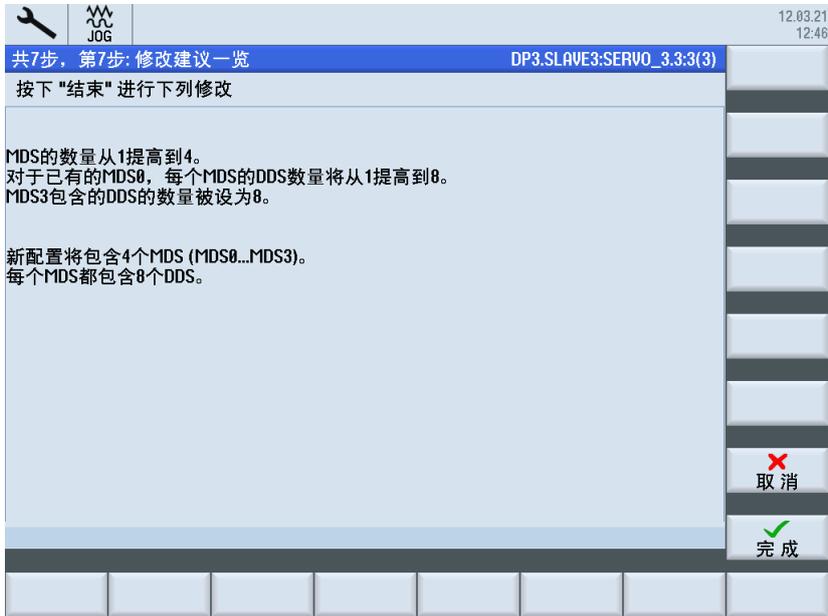
共7步, 第6步: 调整DDS MDS3 (可选) DP3.SLAVE3:SERVO_3.3:3(3)

MDS	驱动数据组 (DDS)							编码器1	编码器2	编码器3	
MDS0	0	1	2	3	4	5	6	7	EDS0	none	none
MDS1	8	9	10	11	12	13	14	15	EDS0	none	none
MDS2	16	17	18	19	20	21	22	23	EDS0	none	none
MDS3	24	25	26	27	28	29	30	31	EDS0	none	none

DDS的数量 MDS3 8

取消 下一步 >

7. 按下“下一步 >”，在摘要中检查需要执行的修改。



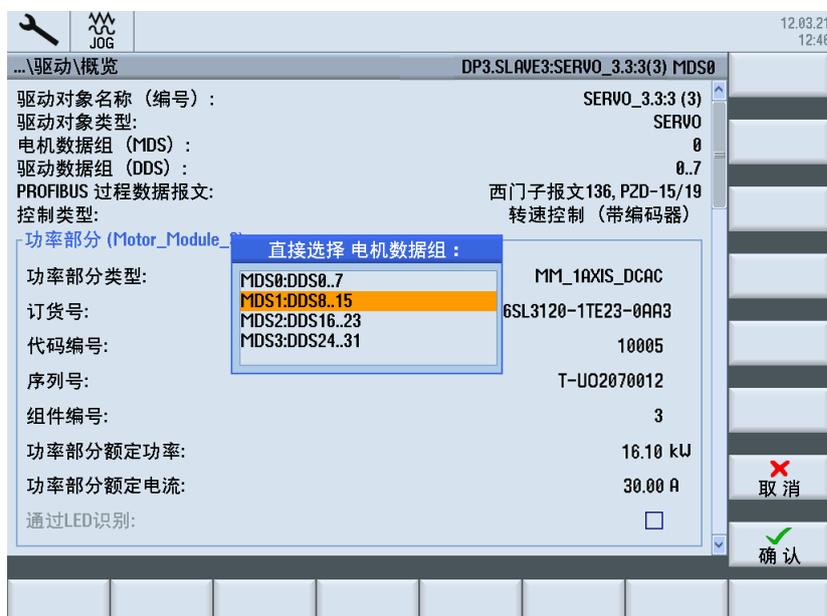
8. 按下“完成 >”接收修改。



9. 点击“确定”，对数据进行非易失性存储。

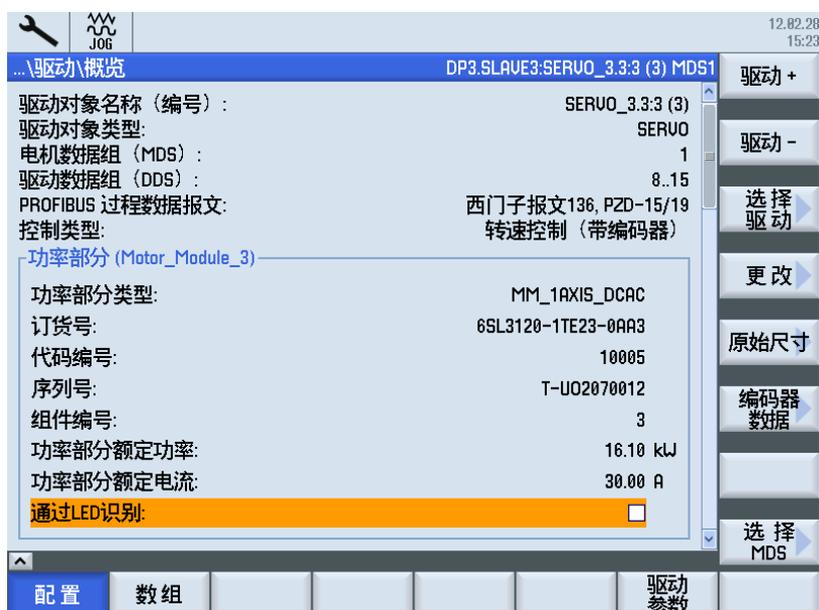
参数的保存/写入可能需要持续几分钟。

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。

6.3.3 删除数据组

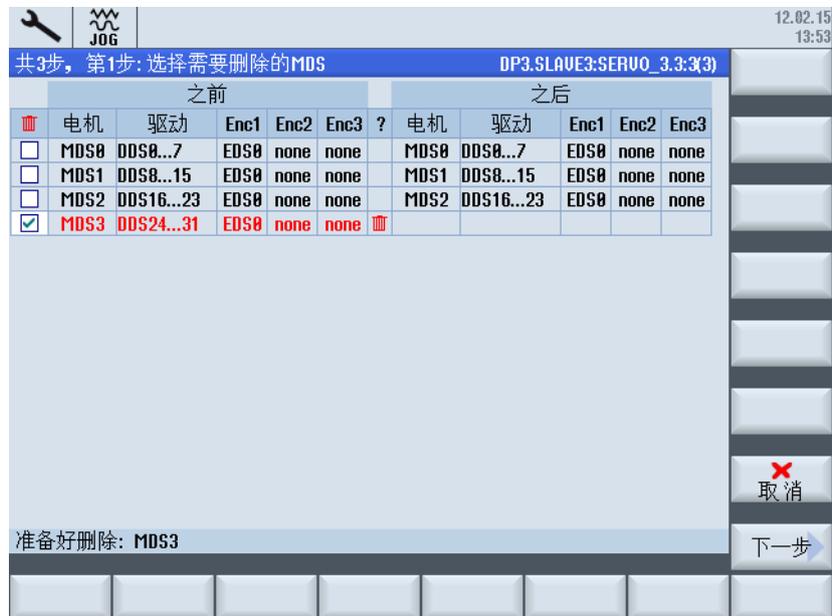
前提条件

执行删除数据组操作须满足以下前提条件：

- MDS 中 DDS 数量 > 1 ⇒ 可删除 DDS。
- MDS 数量 > 1 ⇒ 可删除 MDS。

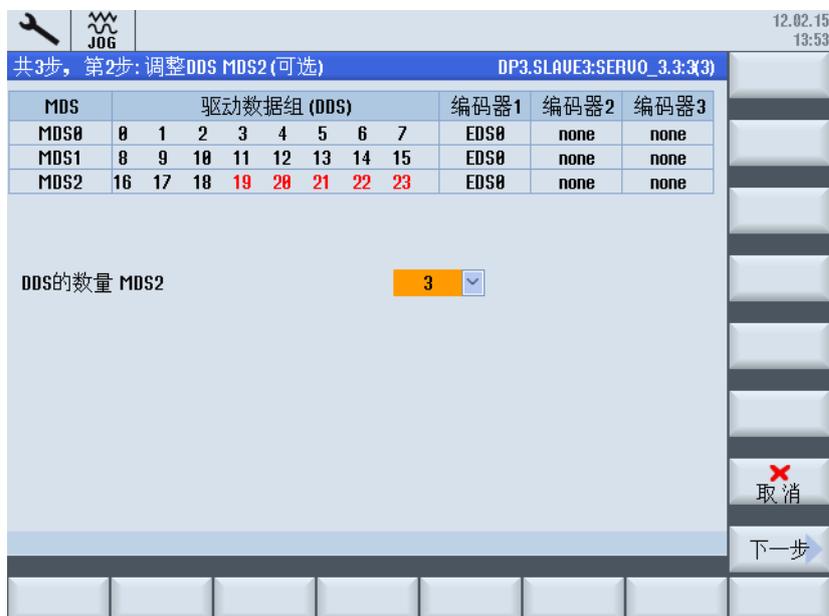
删除数据组的操作步骤

1. 选择“删除数据组”。



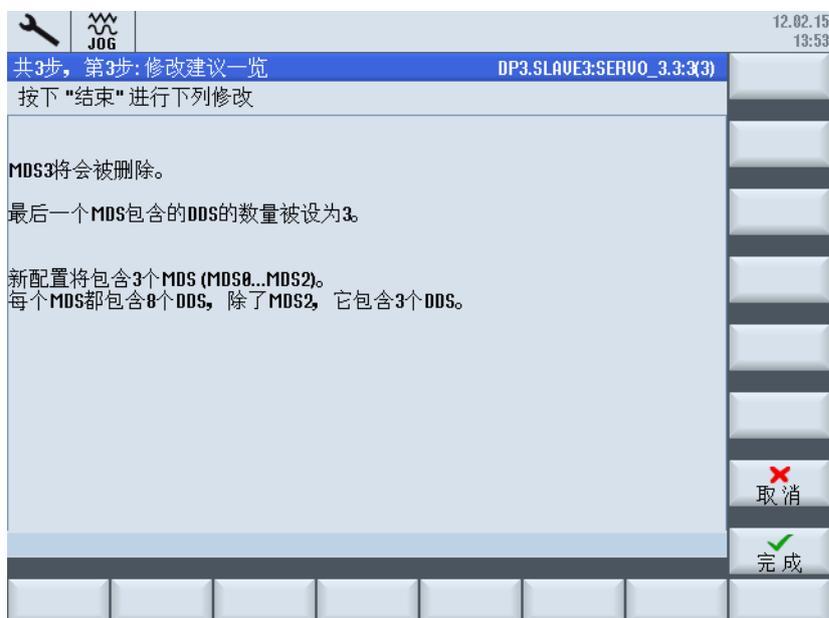
在第一列中通过钩号标记相应行，从而选中需要删除的 MDS。可选择多个数据组。

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。

6.3 配置数据组

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。

6.3.4 修改数据组

前提条件

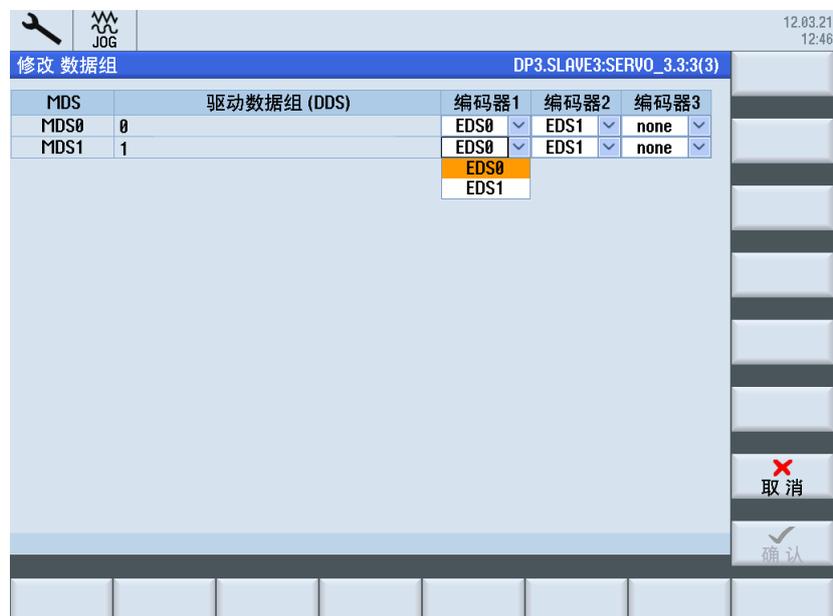
若对驱动对象（SERVO）分配了超过一个编码器，则创建附加的编码器数据组（EDS0...2）。

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：



6.4 DRIVE-CLiQ 拓扑规则

6.4.1 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 6-1 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 6-2 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 6-3 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 6-4 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

6.4 DRIVE-CLiQ 拓扑规则

表格 6-5 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。

6.4.2 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。

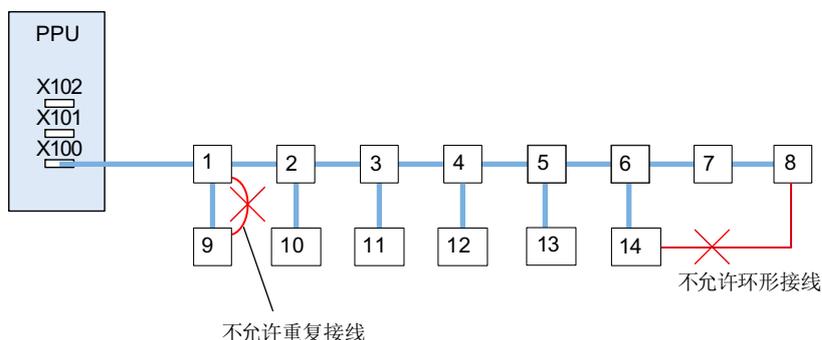


图 6-68 示例：端子 X100 上的 DRIVE-CLiQ 支路（无 Safety Integrated）

- 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。

- 在“PPU + NX10.3”的组合上可以最多连接 16 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。

- 每个 PPU 和 NX10.3 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。

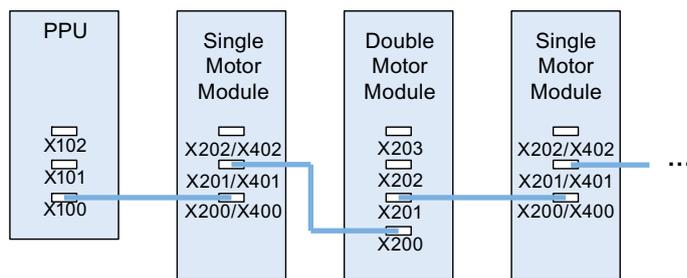


图 6-69 示例：DRIVE-CLiQ 线路

- 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。

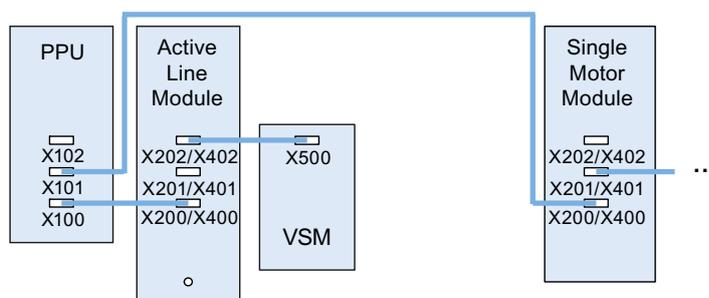


图 6-70 示例：书本型和装机装柜型组件带有 VSM 的拓扑

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。

6.4.3 SMC40 的拓扑结构规则

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

只有在 DRIVE-CLiQ 接口 X500/x 以及对应的编码器接口 X520/x 被占用后，SMC40 才会传送到实际拓扑结构中。

如果没有连接编码器，SMC40 也不会传送到拓扑结构中。

6.5 端子布局

6.5.1 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

6.5 端子布局

引脚	信号名称	状态	含义	
6	DI21	DI21	r722.21	数字量输入端 21
7	M2	MEXT2	---	引脚 1...6 的接地
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

6.5.2 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]	数字量 NCK 输入端 1
4	IN2	DIN2	\$A_IN[2]	数字量 NCK 输入端 2
5	IN3	DIN3	\$A_IN[3]	数字量 NCK 输入端 3
6	IN4	DIN4	\$A_IN[4]	数字量 NCK 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]	数字量 NCK 输出端 1
10	O2	DOUT2	\$A_OUT[2]	数字量 NCK 输出端 2
11	M3	MEXT3		引脚 9、10、12、13 的接地
12	O3	DOUT3	\$A_OUT[3]	数字量 NCK 输出端 3
13	O4	DOUT4	\$A_OUT[4]	数字量 NCK 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]	数字量 NCK 输入端 9
4	IN10	DIN10	\$A_IN[10]	数字量 NCK 输入端 10
5	IN11	DIN11	\$A_IN[11]	数字量 NCK 输入端 11
6	IN12	DIN12	\$A_IN[12]	数字量 NCK 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]	数字量 NCK 输出端 9
10	O10	DOUT10	\$A_OUT[10]	数字量 NCK 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： 数字量 NCK 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： 数字量 NCK 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 99)

6.5.3 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。

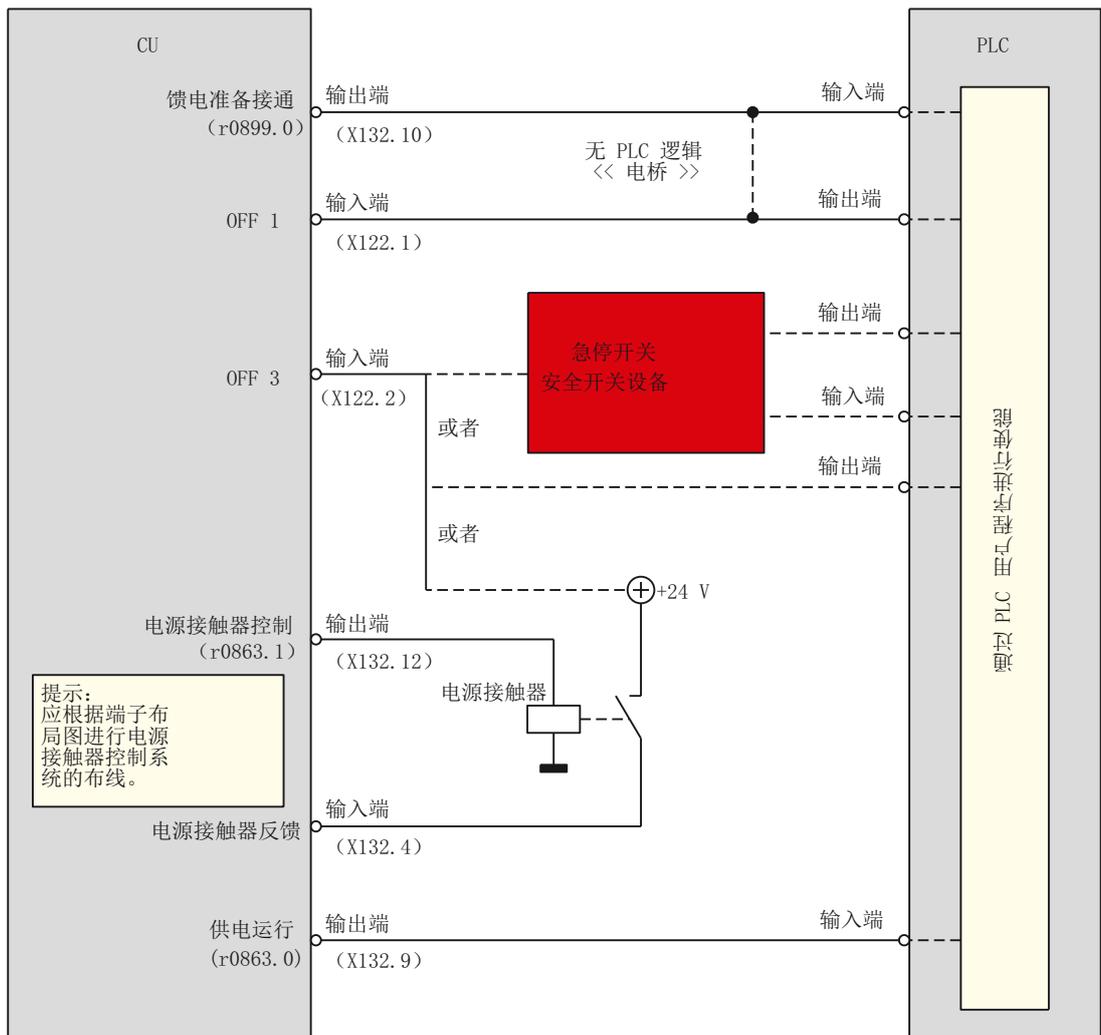
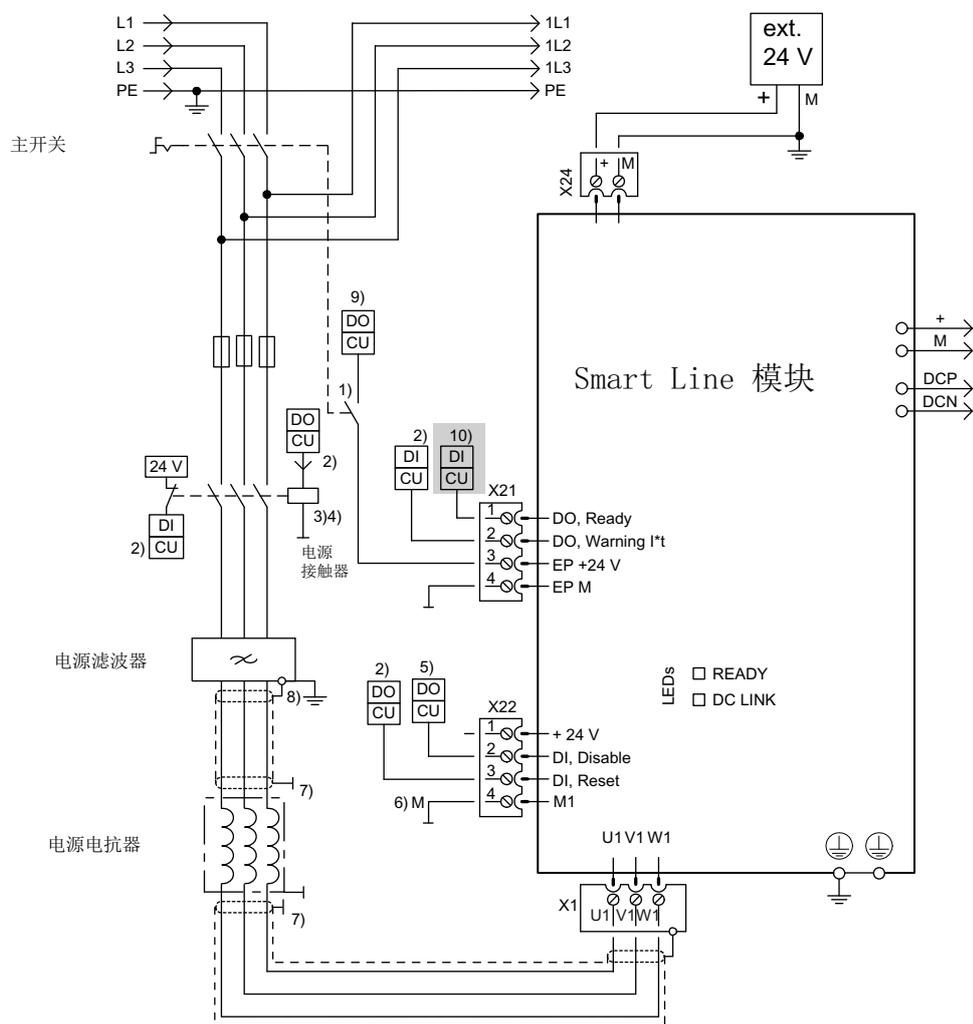


图 6-71 控制单元和电源接触器的连接

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 6-72 示例: 连接 SLM

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

6.5 端子布局

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

6.5.4 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：

测头 1 连接到 CU 端子 X122 的引脚 13 上或连接到 NX 端子 X122 的引脚 13 上

测头 2 连接到 CU 端子 X132 的引脚 13 上或连接到 NX 端子 X122 的引脚 12 上

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：SYSTEM）

- **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。

测头状态

DB2700	来自 NCK 的一般信号[r] NCK → PLC 的接口							
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB0	--	--	--	--	--	--	急停有效	---
DBB1	系统英制单位	--	--	--	--	--	测头已动作	
							测头 2	测头 1

优化驱动

7.1 引言

激活机床数据修改

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 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”。

控制框图

控制器和驱动包含了三个相连的控制闭环：

- 电流环
- 转速环
- 位置环

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。

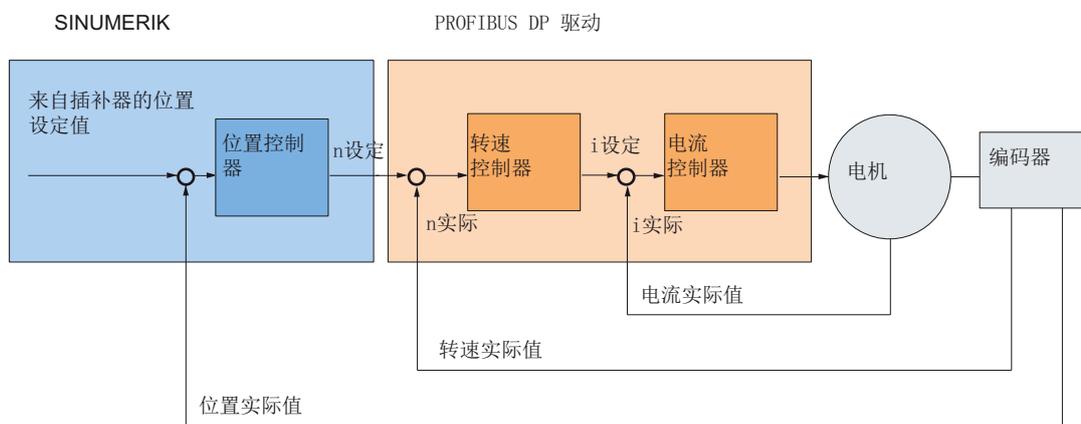


图 7-1 控制结构

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 189)
- 轴急动度 (页 202)
- 转矩使用率 (页 210)
- 圆度测试 (页 214)
- 主轴优化 (页 223)

文档

更多信息请参见以下手册：

- SINUMERIK Operate 调试手册 (IM9)，章节“跟踪”
- 功能手册之基本功能分册；章节“轴监控，保护区 (A3)”
- 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)”

7.2 自动伺服优化

7.2.1 目标

引言

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
- 前馈控制模式：等转速控制时间
- 插补轴组

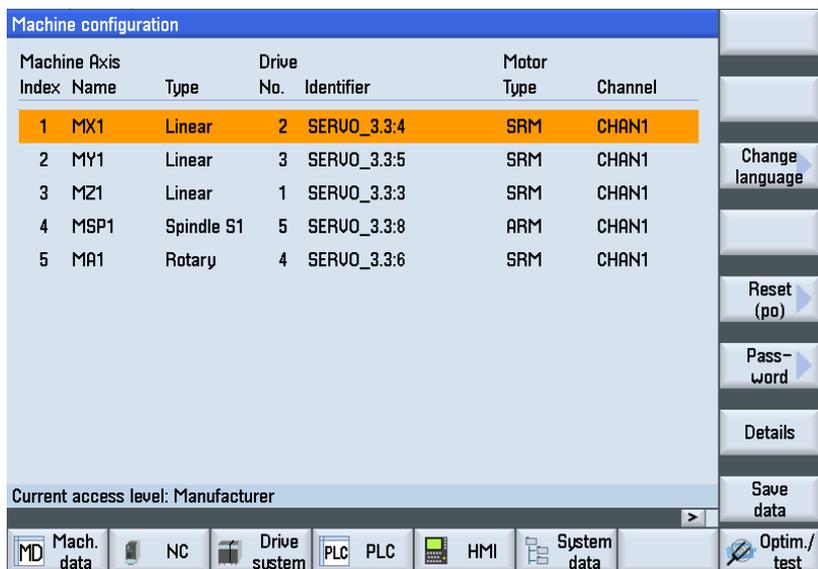
在开始自动优化前，要设置 MD32640[0] \$MA_STIFFNESS_CONTROL_ENABLE = 1 激活动态刚性控制。位置环因此可以更短的周期时间工作。在使用动态刚性控制时还要设置 MD32110 \$MA_ENC_FEEDBACK_POL = 1。需要反转极性时，必须在驱动中通过参数 p0410 设置编码器实际值取反：

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

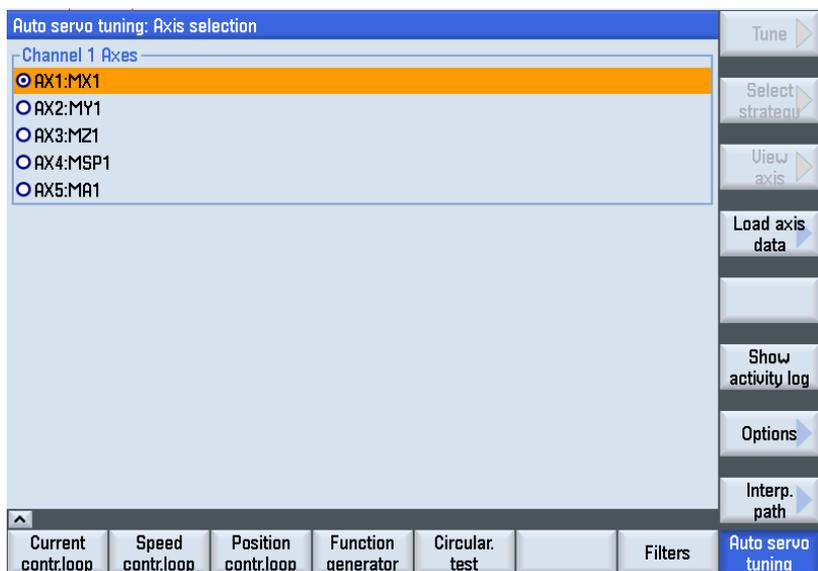
在主轴上使用“自动伺服优化”功能时，可能会缺少伺服使能 PLC 信号。机床厂商应向用户提供使能该信号的方法，以使用户能进行优化。比如：可以通过一个特殊的按键或者设置一个 PLC 状态标记来给出使能。

自动伺服优化

“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。



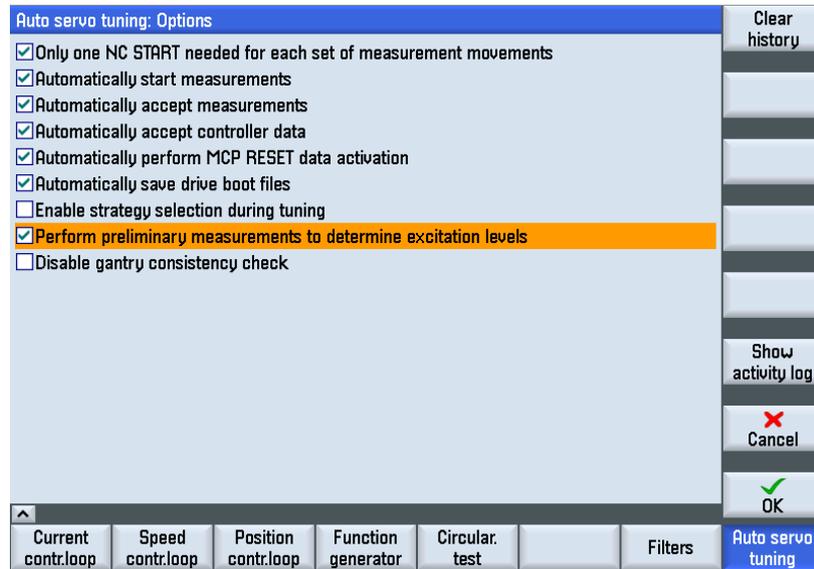
用方向键浏览，然后按下<SELECT> 键选中轴，此处选择 X1 轴：



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。

检查选项

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。

7.2.2 设置优化方案

优化方案

我们推荐保留轴方案、转速控制器方案和位置控制器方案的缺省设置（方案 102、303 和 203，参见下图）：

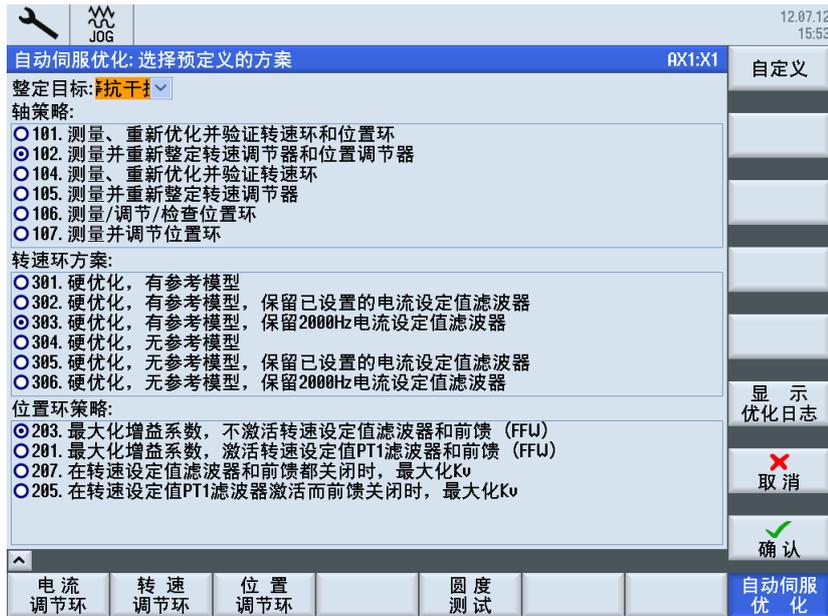


图 7-2 预定义方案选择

“中等抗干扰性”是优化目标的缺省设置。该选择取决于机床/轴的机械特性。选项“最佳抗干扰性”不适用于性能比较弱的轴。

优化目标

优化目标的说明：

- **选项：“最佳抗干扰性”**

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。

- **选项：“适当抗干扰性”**

转速控制器和位置控制器以 80% 的最大增益和良好的稳定性进行优化。

- 应用：有很多应用场合。
- 前提条件：对机床结构没有很高的要求，推荐用于一般应用。

- **选项：“阻尼最佳”**

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性能。

- 应用：运动质量较大的大型机床。
- 前提条件：由于较大的质量/惯性导致机床结构刚性较低。机床的大尺寸导致动态响应较慢。推荐用于操作台，操作仓随之升降。如果“阻尼最佳”得出了非常低的位置控制器增益，可使用该选项。

用户自定义设置

通过软键“用户自定义”可以重新定义优化过程。

设置转速控制器:



重要设置：优化程度和最小积分时间 Tn。

- 优化程度:

该参数基于稳定性限值确定了 Kp 和 Tn 的设置。

- 缺省值 = 0.6
- 最小值 = 0 [最小稳定性]
- 最大值 = 1 [最大稳定性]

- 最小积分时间 Tn:

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

- 缺省值 = 5.0 ms
- 最小值 = 0.5 ms
- 最大值 = 100 ms

设置位置控制器:



重要设置: Kv (位置控制器) 降低系数

- Kv 降低系数:

该参数功能与位置控制器的“优化程度”一样。

该参数会降低由自动伺服优化计算出的最大 Kv。

最大 Kv 是进给失效时导致零点位置超调的最大 Kv。

- 缺省值 = 0.8
- 最小值 = 0.1
- 最小值 = 1 [无降低]

7.2.3 启动自动伺服优化的步骤

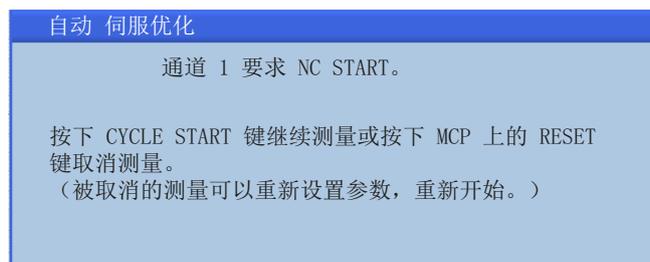
启动自动伺服优化

操作步骤：

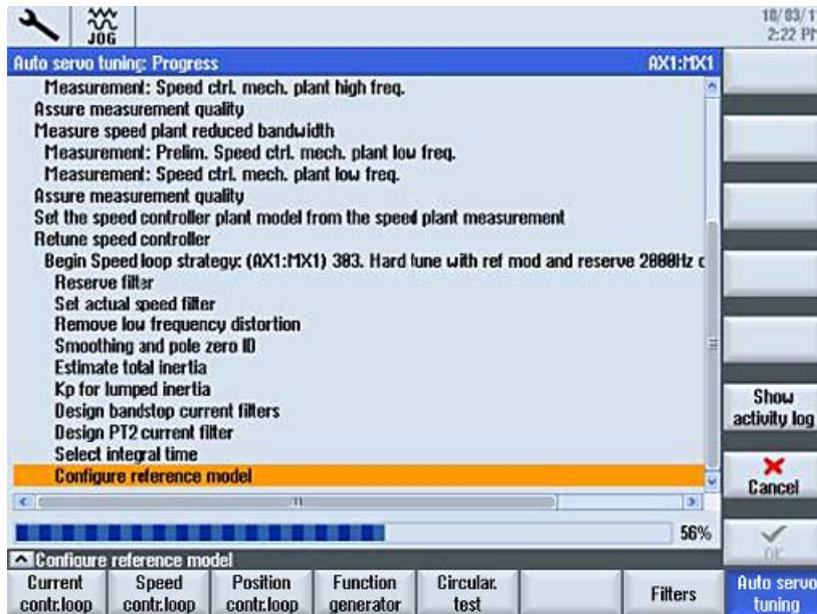
1. 确认需要优化的轴处于安全位置。如果不处于安全位置，在 JOG 模式中将轴移动到其行程中心：



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：

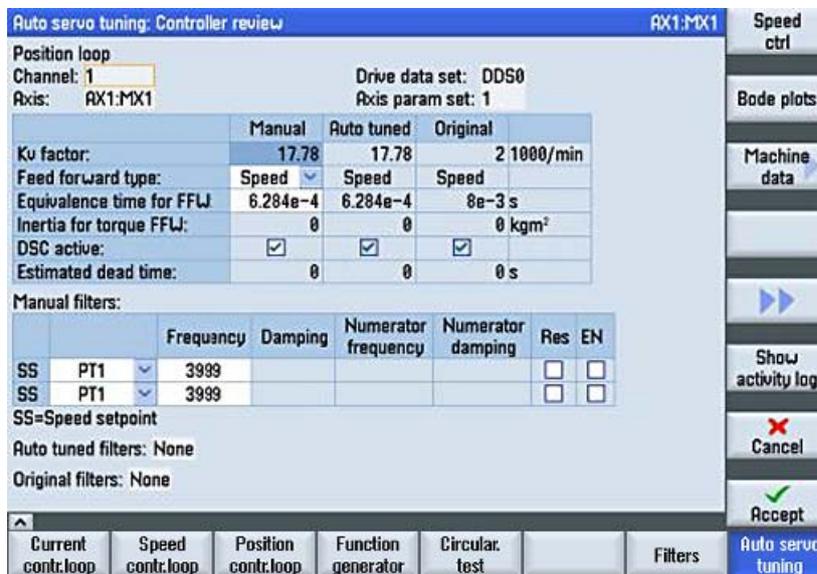


3. 在自动伺服优化的后续过程中还会弹出此类提示。需要多次按下“CYCLE START”键以开始下一个测量。

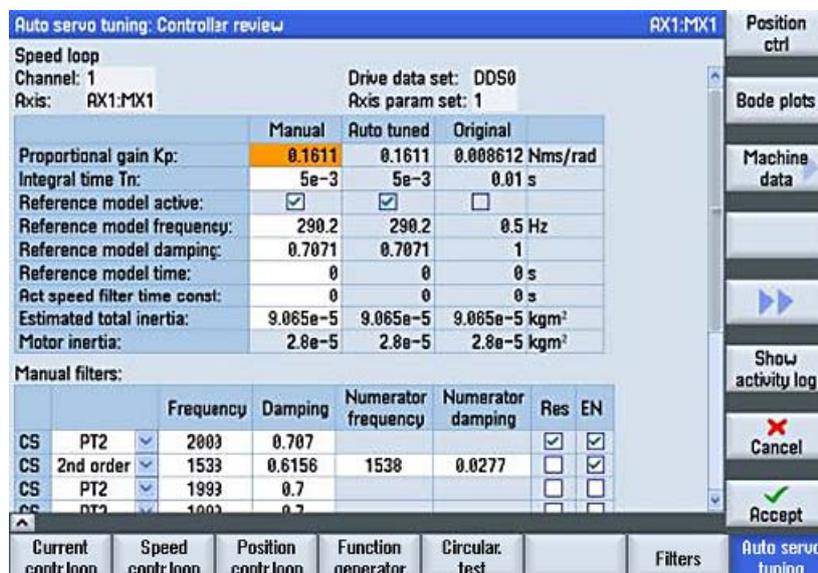


4. 在优化结束后会显示以下画面，在其中可以检查优化后的新值和优化前的原始值。

示例：位置控制器

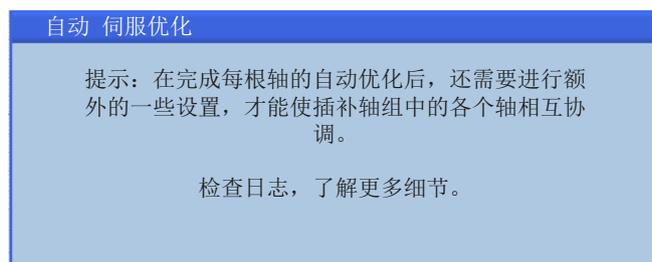


示例：转速控制器



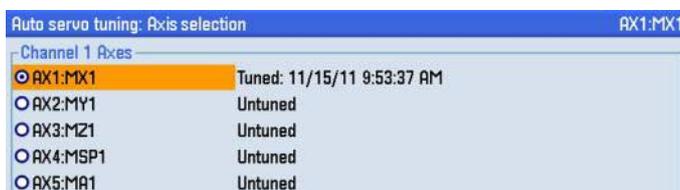
按下软键“接收”后，采用新值。

- 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：

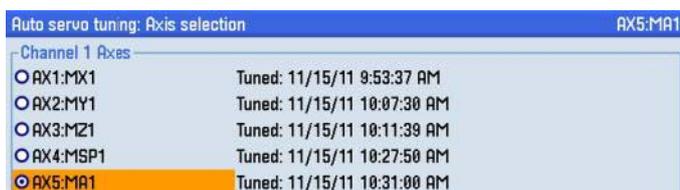


优化结果

接着显示优化的时间和日期：



其他轴可按同样方式优化：



7.2.4 插补轴组的优化方式

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 系数和前馈模式。

插补轴组的优化方案选择

操作步骤：

1. 确定插补轴组的优化方案。此处使用缺省设置：

- 1103 “通过部分调整削弱单轴最佳优化效果”

Auto servo tuning: Interpolation path strategy selection	
<input type="radio"/> 1101. Reduce standalone optimal regulators with maximal matching	
<input type="radio"/> 1102. Reduce standalone optimal regulators with moderate matching	
<input checked="" type="radio"/> 1103. Reduce standalone optimal regulators with partial matching	
<input type="radio"/> 1104. Reduce standalone optimal regulators with custom matching	
Match effective K_p :	<input type="checkbox"/>
Minimum K_p reduction factor:	0.4
Spindle can limit other axes:	<input type="checkbox"/>
Other axes can limit spindle:	<input type="checkbox"/>

各个方案的含义：

- **最大程度的调整：** 调整生效的比例增益 K_p 。
比例增益 K_p 可最多下调到原始值的 10%。
某些条件下进给轴 K_p 不下调，以便使主主轴可使用下调后的值。主轴轴 K_p 不下调，以便使进给轴可使用下调后的值。
- **中等程度的调整：** 调整生效的比例增益 K_p 。
比例增益 K_p 可最多下调到原始值的 50%。
某些条件下进给轴 K_p 不下调，以便使主主轴可使用下调后的值。
主轴轴 K_p 不下调，以便使进给轴可使用下调后的值。
- **部分调整：**
不调整生效的比例增益 K_p 。
- **自定义调整：**
按照用户自定义设置选择方案。复选框可激活。

一般设置

“轴组最佳”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 K_v 系数。

- 前馈模式

– 如果所有轴的优化都加了转矩前馈, 则在插补轴组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_FFW_MODE = 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_FFW_MODE = 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。

插补轴组

下一步是选择需要进行优化的插补轴组包含的轴。

操作步骤:

1. 按下软键“插补轴组”。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：

5. 按下“接收”和“确认”，采用新设置。

自动优化到此结束。

7.3 优化轴急动度

7.3.1 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/s³”，即加速度的时间变化率：加速度（距离单位/s²）除以时间（s）。

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

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。

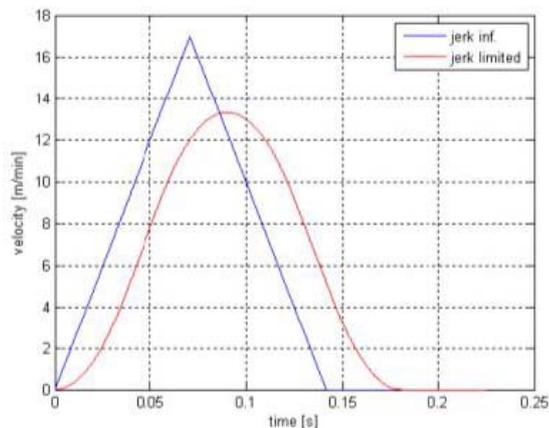
- 在程序中写入 **SOFT** 后，急动度限制激活。
- 在程序中写入 **BRISK** 后，急动度限制关闭。

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：



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到滑块的响应特性。
- 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。

7.3.2 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。

7.3.3 选择跟踪信号

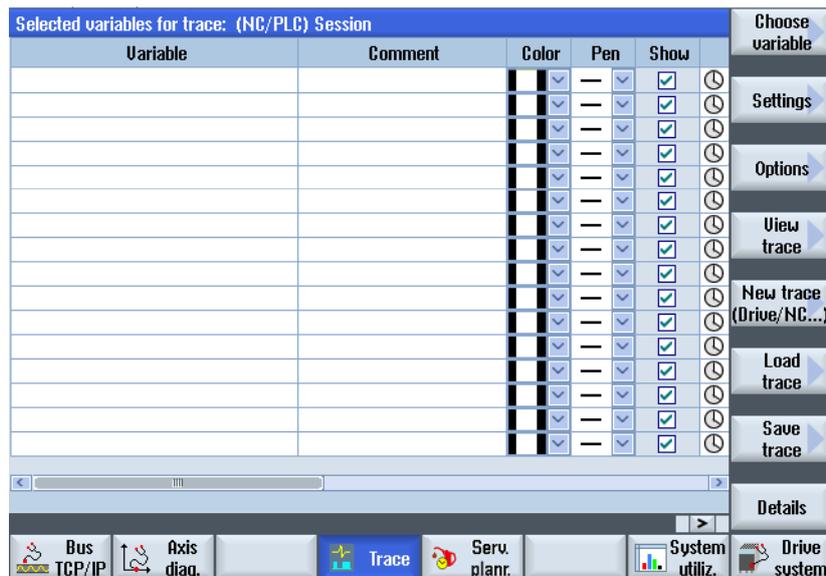
选择跟踪信号

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。

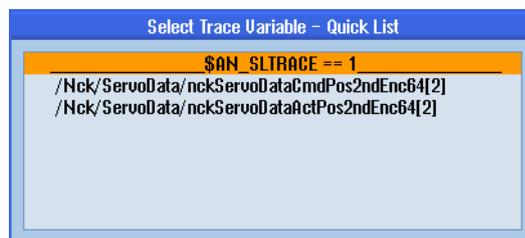
3. 按下软键“筛选/搜索”，以筛选变量。有以下筛选条件：
 - 系统变量
 - NC
 - 轴
 - 通道
 - 操作方式组（BAG）
 - 伺服
 - 全局用户数据(GUD)
 - PLC
 - PLC 报警
 - 全部（无筛选）
4. 设置筛选条件“伺服”，从中选择两个变量，比如： Y 轴位置设定值和位置实际值：

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



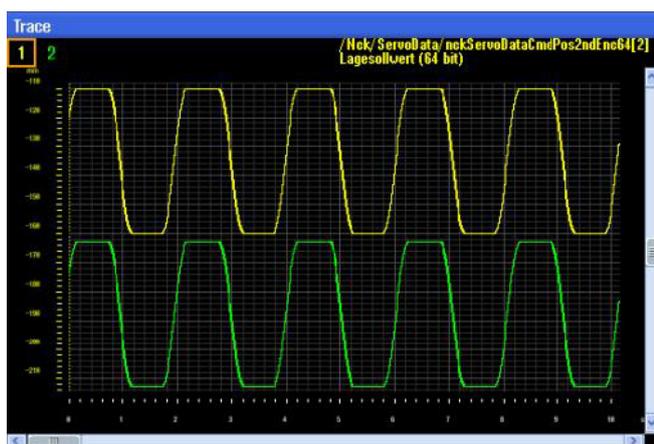
7.3 优化轴急动度

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。

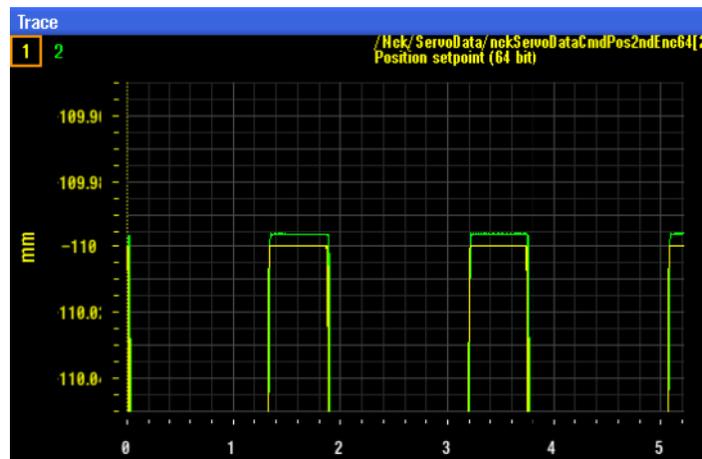
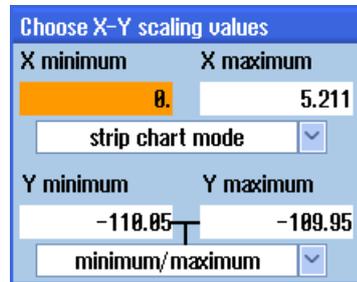


设置跟踪的显示比例

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。

7.3.4 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。

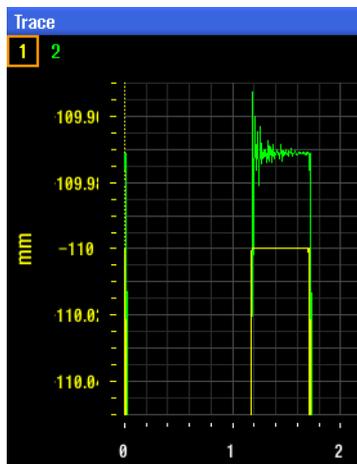
示例

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。

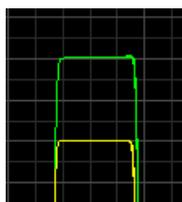
1. 重新启动跟踪，然后执行测试程序。
2. 按下软键“启动跟踪”，然后按下<RESET>和<CYCLE START>。
3. 调高 MAX_AX_JERK，排除超调。

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。



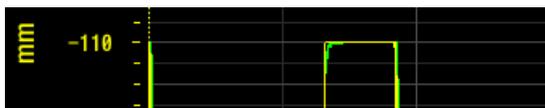
示例：加入了前馈和位置控制器的定位

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：



7.4 转矩使用率

检查最大轴加速度

在优化了 K_v 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：

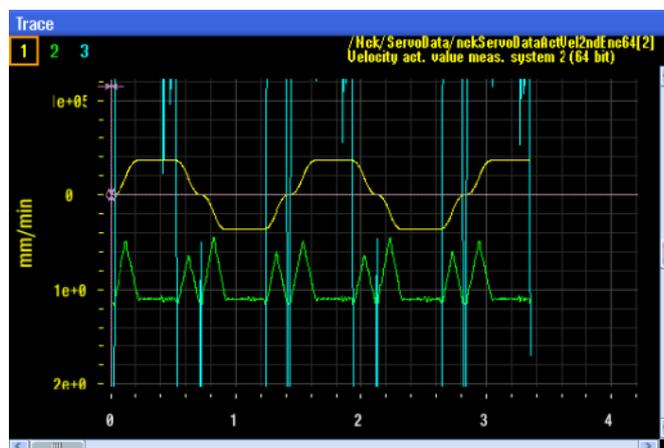
```
SOFT
$AN_SLTRACE=0; Reset Start Servo-Trace-Trigger
LAB:
G0 X10
$AN_SLTRACE=1; Start Servo-Trace-Trigger
X310
GOTOB LAB
M30
```

选择以下信号进行跟踪：

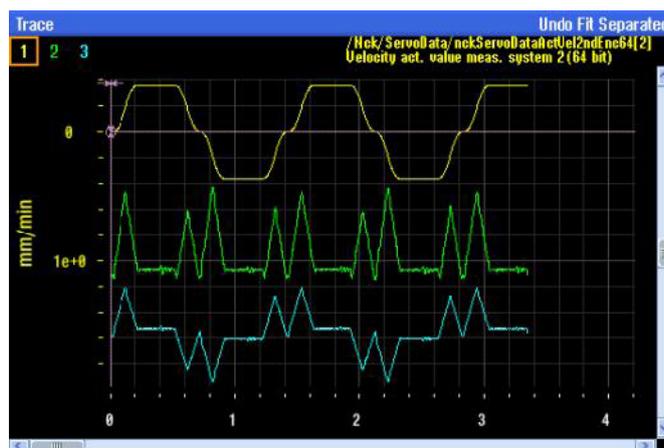
- 测量系统 2 提供的速度实际值
- 转矩使用率
- 转矩电流实际值 $i(q)$

Selected variables for trace: (NC/PLC) Session					
Variable	Comment	Color	Pen	Show	
/nckServoDataActVel2ndEnc64[2]	Velocity act. value meas. system	Yellow	—	<input checked="" type="checkbox"/>	<input type="checkbox"/>
/nckServoDataDruLoad64[2]	Load (m_set/m_set,limit) (64 bit)	Green	—	<input checked="" type="checkbox"/>	<input type="checkbox"/>
a/nckServoDataActCurr64[2]	Torque-prod. Current act. Val. $i(q)$	Cyan	—	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		Black	—	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		Black	—	<input checked="" type="checkbox"/>	<input 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：

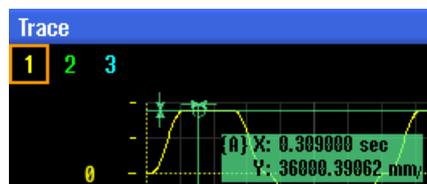


按下软键“指针”，将指针移动到需要标出最大值的曲线图上：软键“指针” → “指针 A”。

按下软键“指针”，将指针移动到需要查看最大值的曲线图上：

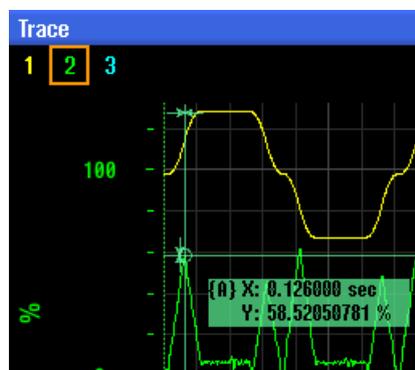
- 速度

跟踪曲线图上的指针表明，达到的最大速度为 36 m/min：



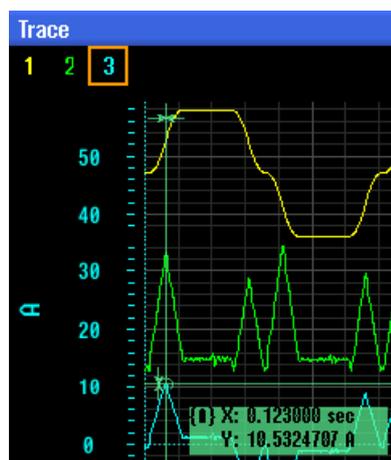
- 使用率

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。

7.5 圆度测试

7.5.1 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

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。

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

示例

用于 X-Y 轴测量的 NC 测试程序：

```
FFWON  
SOFT  
G90 G01 F3000 X400 Y200 Z500  
LAB:  
G91 G64 G02 X0 Z0 I10  
GOTOB LAB  
M30
```

位置、进给率和有效平面应根据机床进行匹配！

定义与结果

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。

7.5.2 圆度测试：执行测量

设置参数

Circularity test measurement				Axis+							
Measurement				Axis-							
Axis:	Meas. system:	Absolute position:	Status:	Start							
MX1	1 Active	0.00000 mm	Inactive	Stop							
MZ1	3 Active	0.00000 mm	Inactive	Graphic							
Parameter		Representation:		Optimization							
Radius:	10.00000 mm	Resolution	0.01000 mm/grad.								
Feedrate:	3000.00000 mm/min	Represent.:	Mean radius								
Multiplier:	1.00000										
Meas. time:	1257 ms										
<table border="1"> <tr> <td>Current contr.loop</td> <td>Speed contr.loop</td> <td>Position contr.loop</td> <td>Function generator</td> <td>Circular test</td> <td>Filters</td> <td>Auto servo tuning</td> </tr> </table>				Current contr.loop	Speed contr.loop	Position contr.loop	Function generator	Circular test	Filters	Auto servo tuning	
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。
- “**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

操作步骤：

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），则记录时会进行相对更粗略的采样（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”。

7.5.3 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

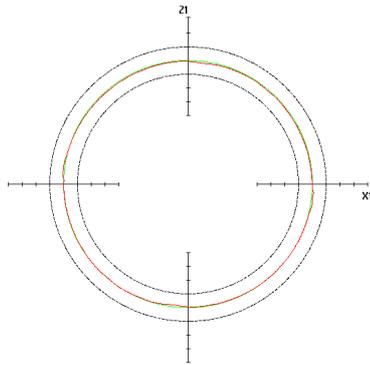
一般在使用优化的进给前馈时，半径的实际值都会过大。这可使用时间常数 MD32410 \$MC_AX_JERK_TIME 来修正。需要时，所有轴可使用统一的一个时间常数。

7.5 圆度测试

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

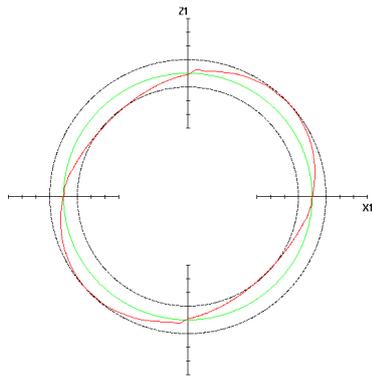
优化示例 2

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 μ m

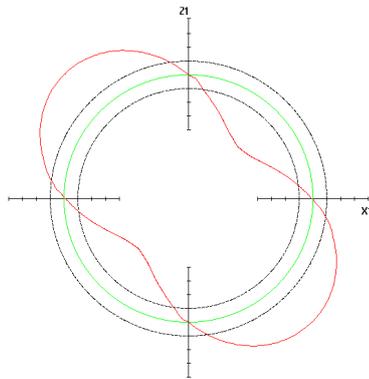
优化示例 3

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 μm

7.5.4 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]

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]

7.6 优化主轴

7.6.1 设置主轴的机床数据

检查机床数据设置

调整下表中列出的和主轴/进给轴插补相关的各机床数据以及参数。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	回转轴和主轴的模数转换
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	带快速移动倍率的 JOG 模式下的旋转进给率
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	前馈控制类型
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

7.6 优化主轴

序号	标识	值	含义
MD34020	\$MA_REFP_VELO_SEARCH_CAM	30	
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
MD36030	\$MA_STANDSTILL_POS_TOL	5	静态公差

序号	标识	值	含义
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

7.6.2 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)。

现在查看电机技术数据表，比如：异步电机 1PH8089-1VM02-0MG1:

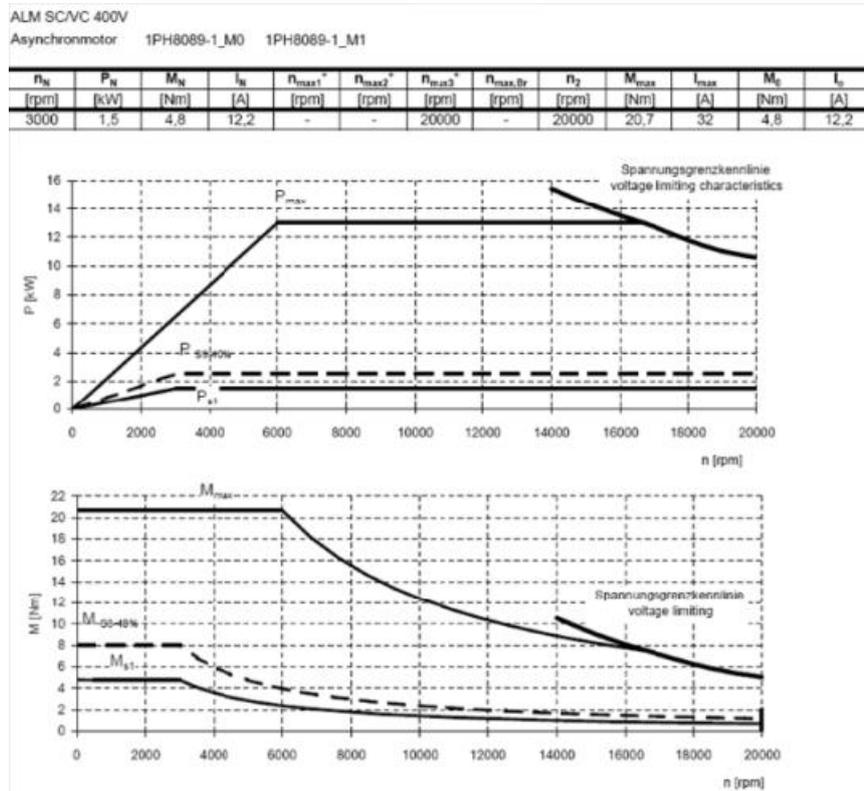


图 7-3 1PH8 异步电机

电机的初始加速和减速方式符合 S1 曲线。

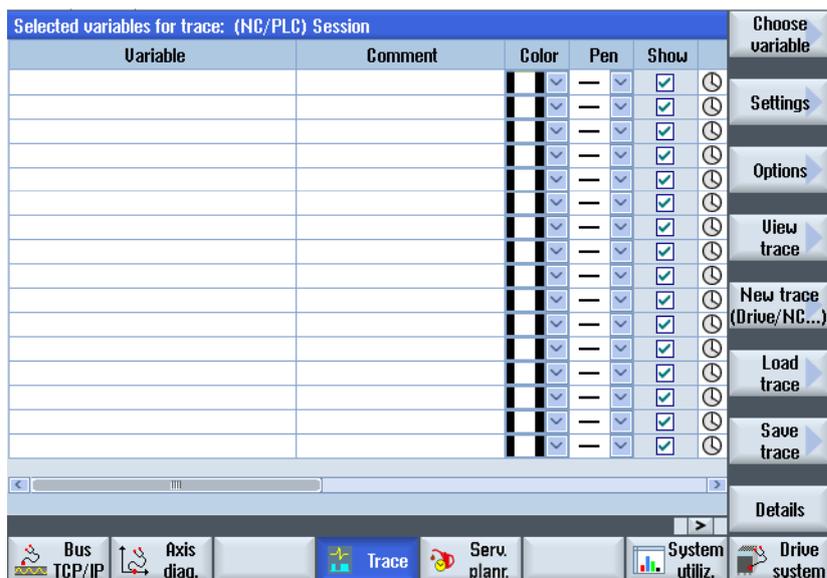
序号	名称	值
p0640	电流限值 (p0305 电机额定电流 x 1.5)	17.7 A
p1520	CO: 电动式转矩上限	4.8 Nm
p1521	CO: 再生式转矩下限	-4.8 Nm
p1530	电动式功率限值	1.5 kW
p1531	再生式功率限值	-1.5 kW

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）

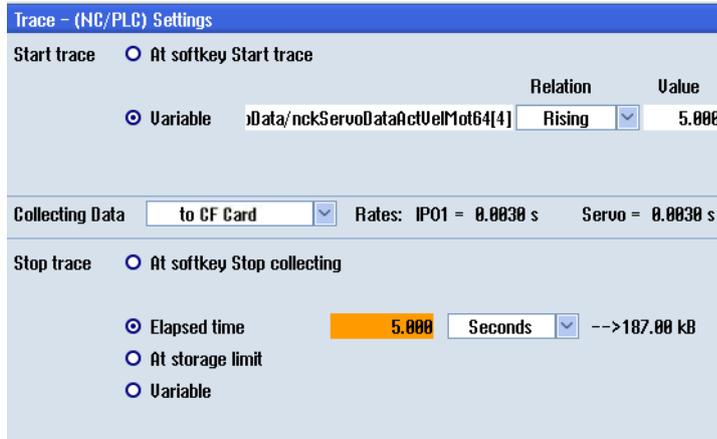
3. 选择所需变量： 软键“选择变量”

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：

Selected variables for trace: (NC/PLC) Session					
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"/>	
q/nckServoDataActCurr64[4]	Torque-prod. Current act. Val. $i(q)$	Cyan	—	<input checked="" type="checkbox"/>	
nkServoDataCmdTorque64[4]	Torque / force setpoint (limited) (t	Magenta	—	<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：

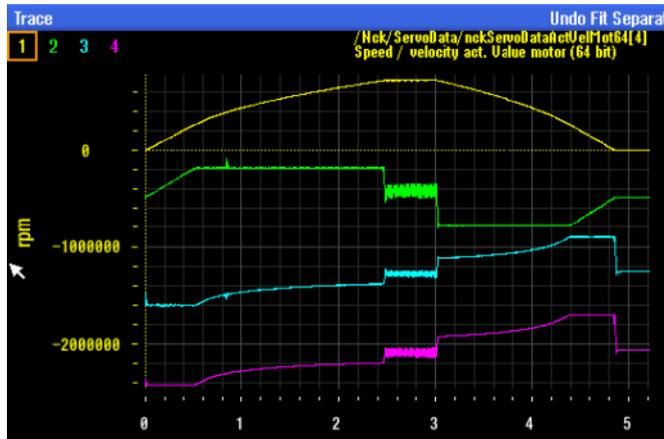
```
S8000 M03
G04 F5
M30
```

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

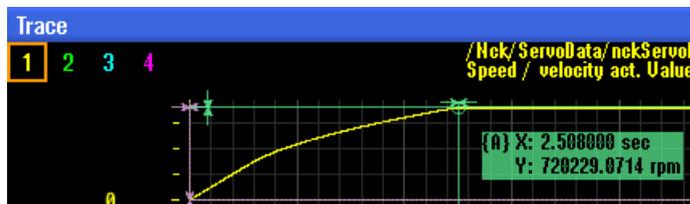
● 在跟踪结束后，按下软键“平铺显示”：



用方向键调整加速时间。

选择跟踪： 实际转速（跟踪 1）

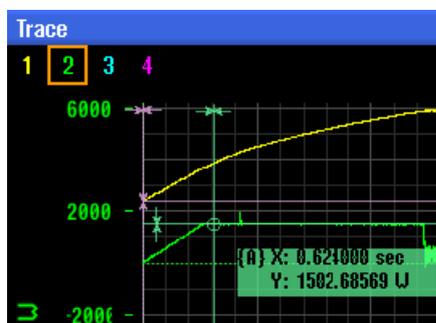
加速到设定转速需要 2.5 秒。



指针会移动到曲线图开头，并定位在达到设定转速的点上。

- 电机加速度所需的功率在“跟踪 2”中查看：

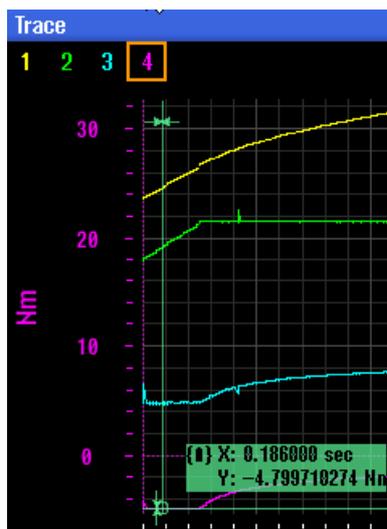
选择跟踪： 实际转速



最大可用功率 = 1.5 kW = S1 曲线

- 转速设定值可在“跟踪 4”中查看：

选择跟踪： 转矩设定值



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。

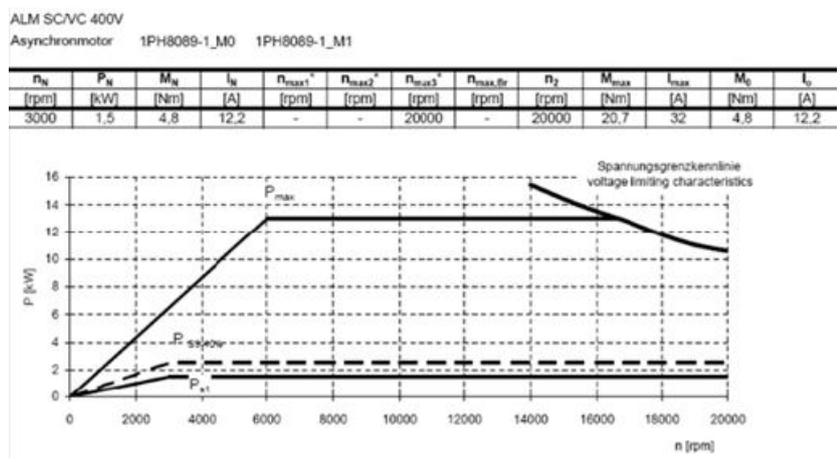


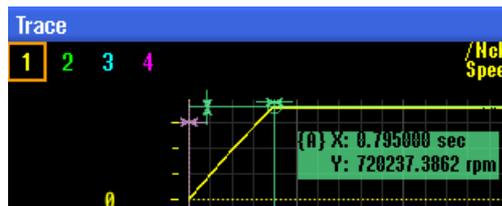
图 7-4 1PH8 异步电机

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
```

7.6.3 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. 检查机床数据 35550 [0] & [1] DRILL_VELO_LIMIT（最大攻丝速度）。
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	
servoDataCmdPos2ndEnc64[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"/>	
			---	<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 - (NC/PLC) Settings

Start trace At softkey Start trace

	Relation	Value
<input checked="" type="radio"/> Variable /nckServoDataActVelMot64[4]	>	5.000

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

Stop trace At softkey Stop collecting

Elapsed time 10.000 Seconds -->100.00 kB

At storage limit

Variable

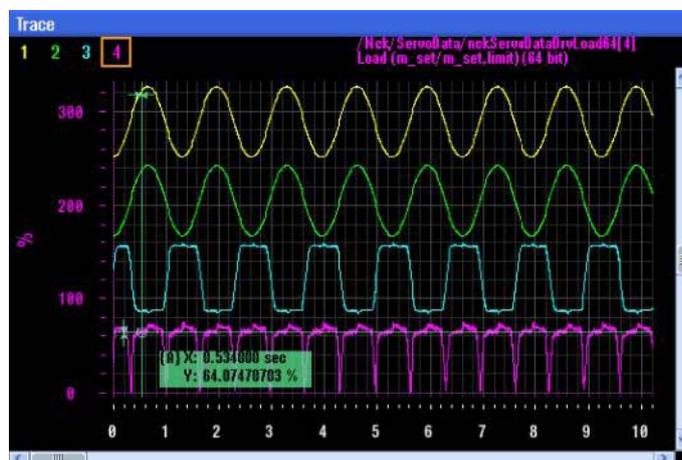
5. 启动跟踪，按下机床面板上的“CYCLE START”。

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。

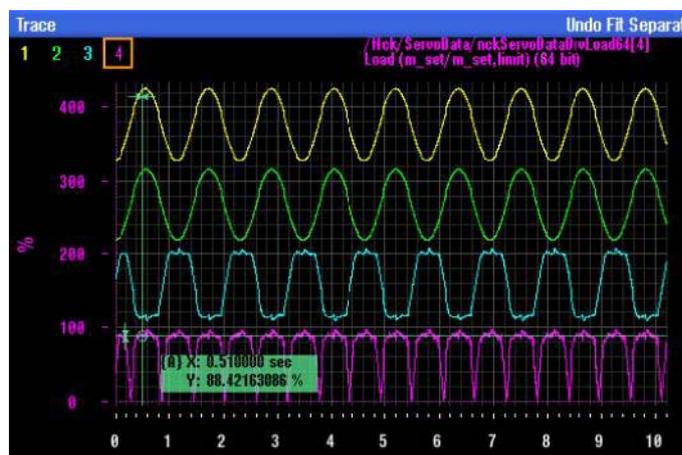
1. 调整机床数据 MD35210 \$MA_GEAR_STEP_POSCTRL_ACCEL 和 MD32431 \$MA_MAX_AX_JERK，以获得最佳结果。
2. 另外检查主轴的可调范围 p1464, p1465。

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%



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。

系统中的配置

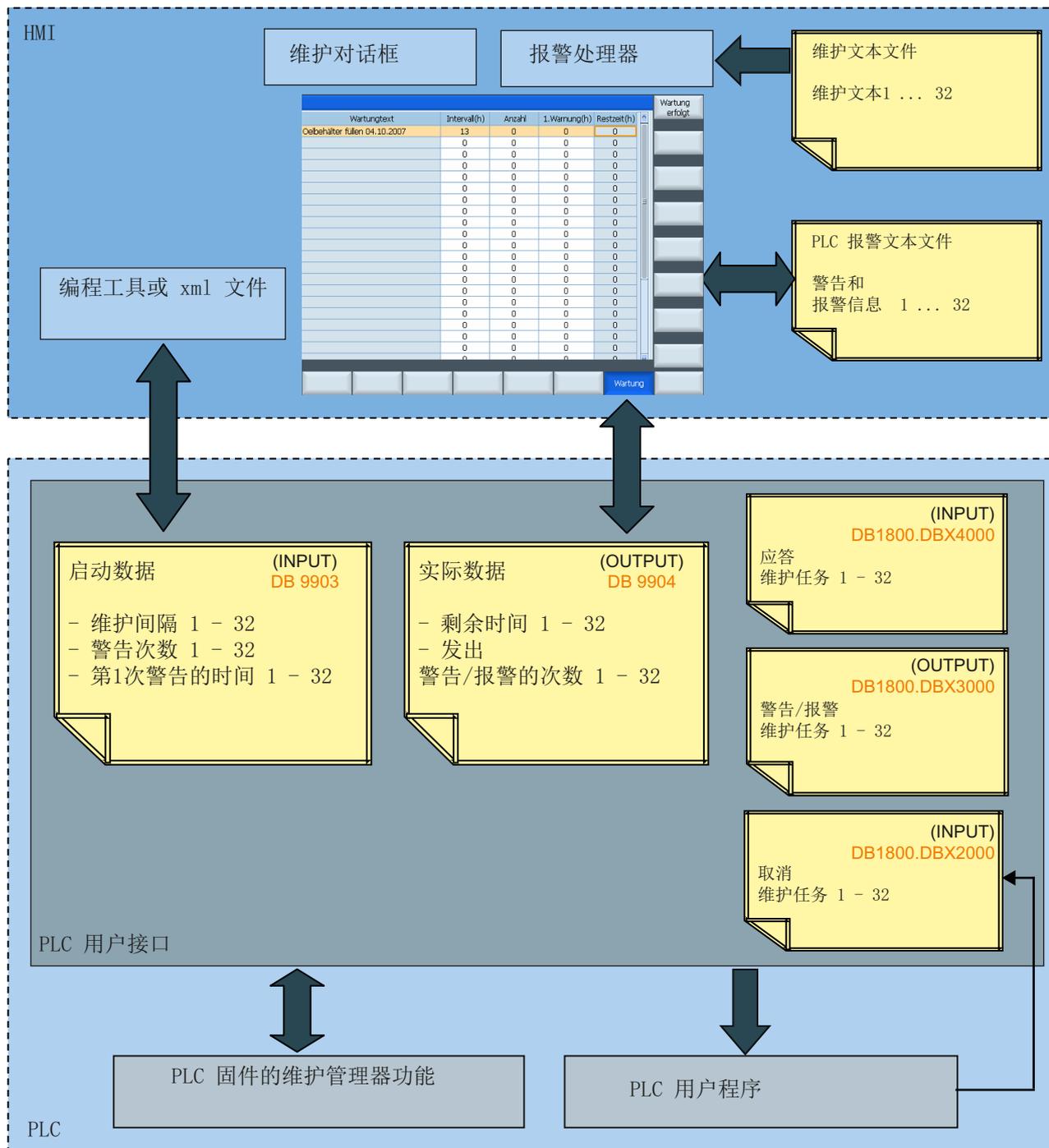


图 8-1 维护管理器：配置

8.1 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 报警/警告信息。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]							
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							

名称	含义
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）。
保留	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							
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							

8.1 PLC 用户程序中的接口

名称	含义
剩余时间	任务启动离期满还剩下的时间（小时）。 剩余时间 ≠ 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
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

名称	含义
应答 n	<p>指定给任务 n 的应答位：</p> <p>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：</p> <ul style="list-style-type: none"> • 剩余时间:= 间隔， • 待输出的警告的数量 := 0 <p>在 PLC 扫描周期结束后，该位会自动复位。</p>

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

名称	含义
报警 n	<p>指定给任务 n 的报警位。</p> <p>在每个 PLC 扫描周期中，该位都置位一次：</p> <p>作为警告（剩余时间 $\neq 0$）和作为报警（剩余时间 = 0）。</p>

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
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

名称	含义
取消 n	<p>指定给任务 n 的取消位。</p> <p>如果该位由 HMI 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。</p> <p>TRUE: 取消任务</p> <p>FALSE: 激活任务</p> <p>这样便可以根据模块的实际运行时间修改维护间隔。</p>

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

名称	含义
应答禁止 n	<p>指定给任务 n 的应答禁止位。</p> <p>如果该位由 HMI 或 PLC 用户程序置位，即使应答位置位，也不应答任务。</p> <p>TRUE: 禁止应答任务</p> <p>FALSE: 允许应答任务</p> <p>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。</p>

8.2 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：

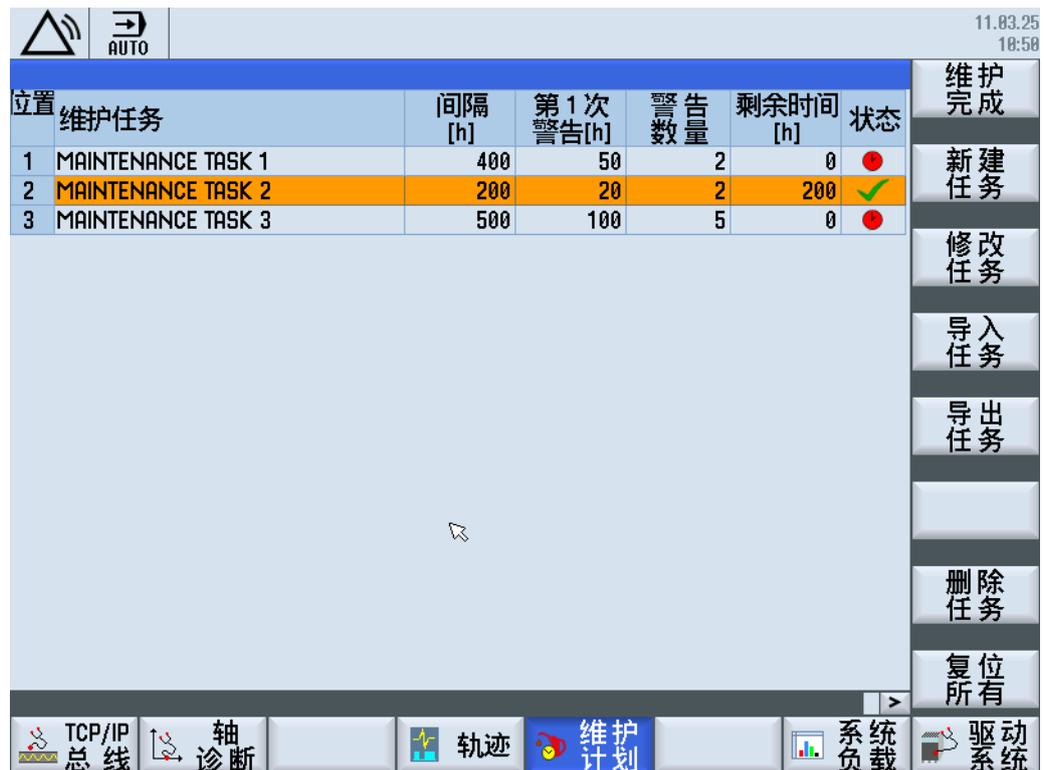
栏目名称	含义
维护任务	维护任务的名称
间隔 [小时]	指离下一次维护相距的最长时间，单位为小时；只要不为 0，PLC 就会将它作为有效的维护任务接收。
第 1 次警告[h]	指出现第一个警告的时间，单位为小时，该值必须小于上面的间隔。
警告数量	指由 PLC 输出、PLC 在间隔期满（剩余时间为 0）后最后一次置位报警位前的报警数量。
剩余时间 [小时]	到间隔期满的时间，单位为小时
状态	<ul style="list-style-type: none"> • 绿色对号表示离下次维护还有一段时间。 • 红色时钟表示目前有维护任务需要完成。

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。



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	●

维护管理器界面右侧操作按钮：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。

底部菜单栏：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”

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	✓

11.03.24
16:51

维护完成

TCP/IP 总线 | 轴 诊断 | 轨迹 | 维护 计划 | 系统 负载

8.3 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

文件名随意选择，例如 `task.xml`。

8.3 导入和导出维护任务的步骤

```
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>
    <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。

8.4 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

9.1 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

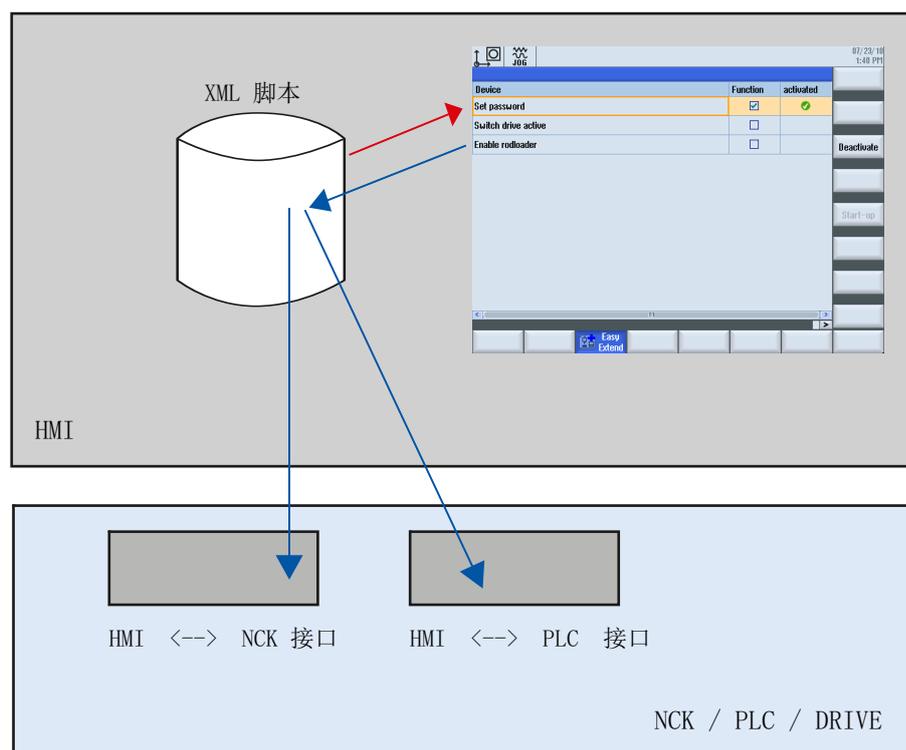


图 9-1 “Easy Extend”的功能

机床制造商首先定义以下功能，才可以使用“Easy Extend”：

- **PLC ↔ 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

和“Easy Extend”相关的文件保存在系统 CF 卡的目录“oem” (MANUFACTURER) 和“oem_i” (INDIVIDUAL)中。

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

9.2 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 417)

9.3 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" index="<index>"/>
index - 分区索引标识符：
0: 设备 0 到 16 (默认设置)
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

9.3 机床制造商和经销商的选件位

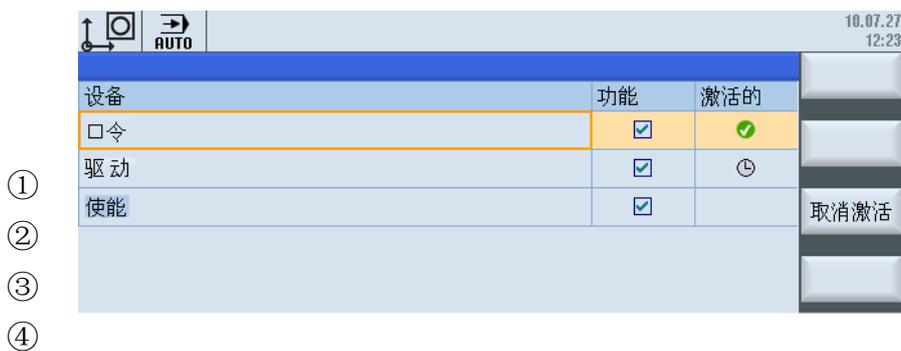
索引	MD14510	数据块	设备名称
....
47	\$MN_USER_DATA_INT[2]位 15	DB9905.DBB188	设备 48
48	\$MN_USER_DATA_INT[3]位 0	DB9905.DBB192	设备 49
49	\$MN_USER_DATA_INT[3]位 1	DB9905.DBB196	设备 50
....
62	\$MN_USER_DATA_INT[3]位 14	DB9905.DBB248	设备 63
63	\$MN_USER_DATA_INT[3]位 15	DB9905.DBB252	设备 64

固定匹配的示例:

```

<device option_bit="bit number">
...
</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。

另见

属性 `option_bit`: XML 脚本的结构 (页 269)

XML 标签: `OPTION_MD` (页 281)

9.4 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。

9.5 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：显示输入的表达方式。 -->
  <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>
    <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>
```

9.6 示例

9.6.1 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>
```

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。

9.6.2 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[D05]"
  />
    <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[D05]"
  />
    <control name = "编辑 1" xpos = "400" ypos = "94" refvar = "drive/dc/r2[D05]" />
  </init>

  <paint>
    <text xpos = "40" ypos = "34">dc[D05]/p105</text>
    <text xpos = "40" ypos = "54">$MC_AXCONF_MACHAX_USED[4]</text>
    <text xpos = "40" ypos = "74">dc[D05]/p971</text>
    <text xpos = "40" ypos = "94">dc[D05]/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>
```

9.6.3 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 = "$MC_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 = "$MC_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

通过用户程序激活或禁止设备。

编程:

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

9.7 脚本语言的说明

脚本文件：扩展 XML

XML(Extended Markup Language)可以用作脚本语言，它增加了数据处理元素和标准语言元素。

因此，和标准 XML 相比，它还能够：

- 保存 NC/PLC 数据、调试数据
- 读写 NC/PLC 数据和驱动数据
- 在一个 XML 块内有条件地跳转
- 执行程序循环
- 执行算术运算指令
- 创建局部变量
- 导入/建立调试存档
- 显示信息

另外，还可以采用“up”执行指令处理 SinuCom Installer 的脚本元素。

脚本的程序段落

脚本由以下段落组成：

- 用于 Easy Extend 的标签
- 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
- 用于设备测试的标签
- 机床数据和标准语言元素的标签
- 参数窗口的标签

下面详细说明各个标签。

9.7.1 特殊字符和运算符

特殊字符的显示

如果要在通用 XML 解释器中正确解释 XML 句法中有特殊含义的字符，必须加以转义。

其中涉及到以下字符：

字符	XML 中的标记符	含义
<	>	大于
>	<	小于
&	&	--
"	"	双引号
'	'	单引号

允许的运算符

运算指令可以处理下列运算：

运算符	XML 中的标记符	含义
=	=	赋值
==	==	等于
!	!	非
!=	!=	不等于
>	>, >	大于
<	<, <	小于
>=	>=, >=	大于等于
<=	<=, <=	小于等于
		位 OR 连接
		逻辑 OR 连接
&	&	位/逻辑 AND 连接
&&	&&	逻辑 AND 连接
+	+	加法
-	-	减法
*	*	乘法
/	/	除法

9.7.2 XML 脚本的结构

概述

XML 中有以下标签，用于描述设备：

- 用于 Easy Extend 的标签
- 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
- 用于设备测试的标签

说明

标签 <tag>	含义
AGM	用于 Easy Extend 的标签
DEVICE 属性: option_bit	用于描述设备的标签。 设备指定有一个固定的位编号，用于管理选件。
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。
SET_INACTIVE	该标签包含取消设备所需流程的说明。
TEST	该标签包含设备测试所需的指令。
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>                                用于 Easy Extend 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>

```

9.7.3 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: <CONTROL_RESET resetnc="TRUE" />
- 属性: resetnc="true" 重新启动 NC 组件。
- resetdrive="true" 重新启动驱动组件。

9.7.4 DATA

说明

用于访问 NCK、PLC 和驱动数据的标签。

详细说明参见章节：参数的编址 (页 297)

编程

标签: **DATA**

句法: <data name = "<地址>" > 值</data>

属性: name 用于变量名称的标签

9.7.5 DATA_ACCESS

说明

该标签用于调整保存用户输入的数据时窗口的属性。该属性应在标签 INIT 中定义。如果不使用该标签，则总是暂时保存输入数据。

例外： 设置了以下属性：hotlink = true

编程

标签: **DATA_ACCESS**

句法: --

属性: type=true 不暂时保存输入值。窗口直接将输入值复制到参考变量中。

 type=false 输入值只由标签 UPDATA_DATA type = "FALSE" 复制到参考变量中。

9.7.6 DATA_LIST

说明

该标签可以暂时备份或恢复驱动数据/机床数据。 可以为每个设备最多创建 20 个临时数据表。

说明

退出 Easy Extend 功能后，系统会全部删除这些数据表。

编程

标签:	DATA_LIST						
句法:	<code><DATA_LIST action = "<read/write>" id = "<列表名称>" ></DATA_LIST></code>						
属性:	<table><tr><td>action</td><td>用于变量值的标签:</td></tr><tr><td><ul style="list-style-type: none">• action="read"• action="append"• action="write"</td><td><ul style="list-style-type: none">• 将列出变量的值存储到一个临时存储器中。• 将列出变量的值添加到已经存在的列表中。• 将保存的值复制到相应的机床数据中。</td></tr><tr><td>id</td><td>用于区分临时存储器的标签</td></tr></table>	action	用于变量值的标签:	<ul style="list-style-type: none">• action="read"• action="append"• action="write"	<ul style="list-style-type: none">• 将列出变量的值存储到一个临时存储器中。• 将列出变量的值添加到已经存在的列表中。• 将保存的值复制到相应的机床数据中。	id	用于区分临时存储器的标签
action	用于变量值的标签:						
<ul style="list-style-type: none">• action="read"• action="append"• action="write"	<ul style="list-style-type: none">• 将列出变量的值存储到一个临时存储器中。• 将列出变量的值添加到已经存在的列表中。• 将保存的值复制到相应的机床数据中。						
id	用于区分临时存储器的标签						

示例

```
<DATA_LIST action = "read" id = "<名称>">
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 = "<名称>" />
```

9.7.7 DRIVE_VERSION

说明

用于驱动版本的标签。版本号会复制到标签“DEVICE”中有效的变量`$driveversion` 中。
详细说明参见章节：参数的编址 (页 297)

编程

标签: **DRIVE_VERSION**

句法: --

属性: --

9.7.8 FILE

说明

该标签可以读入或创建标准文档或数据级文档。

- 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。

如果另外还定义了属性 `class`, 则创建一个数据级文档。文档内容由属性 `class` 和属性 `group` 确定。

编程

标签: **FILE**

句法: `<file name = "<文档名称>" />`

`<file name = "<文档名称>" create="true" class="<数据级>" group="<数据组>" />`

属性:	name	用于文件名称的标签
	class	指定文档包含的数据级。如果需要备份多个数据级，请用空格隔开。 允许指定的数据级有： <ul style="list-style-type: none">• user• manufacturer• individual
	create	以指定的名称在目录 .../dvm/archives/ 下创建一个调试文档。 如果没有定义属性“class”，则该文档是一个包含 NC/PLC/HMI 数据和驱动数据的标准文档。
	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"
class="user manufacturer individual"
group="nc plc hmi" />

<!--将文档读入控制系统 à
<file name="user.arc" />
; 或
<file name="user.ard" />
```

9.7.9 FUNCTION

说明

函数调用： 该标签用于调用“name”属性下指定的函数。

编程

标签： **FUNCTION**

句法： `<FUNCTION name = "函数名称"/>`

属性： **name** 函数名称

return 用于储存函数结果的变量名称

调用参数： 调用参数会作为 XML 的指令值传递。

列出的变量相互之间应以逗号分隔。最多可以传递 10 个参数。

此外，还可以将常数或文本表达式指定为调用参数。

为区分文本表达式，应在文本前加上标志“_T”。

示例

待调用的函数不等待返回值。

```
<FUNCTION name = "函数名称" return="变量名称" />
;参数传递
<FUNCTION name = "函数名称"> 变量, 变量 2, 变量 3</FUNCTION>
<FUNCTION name = "函数名称"> _T"文本", 1.0, 1 </FUNCTION>
```

9.7.10 FUNCTION_BODY

说明

函数体： 该标签包含了子函数的函数体。

编程

标签： **FUNCTION_BODY**

句法：

- 无参数的函数体 `<FUNCTION_BODY name = "函数名称" >`
`...`
`</ FUNCTION_BODY>`
- 带参数的函数体 `<FUNCTION_BODY name="函数名称" parameter="p1, p2, p3" >`
`...`
`<let name="tmp"></let>`
`<op> tmp = p1 </op>`
`...`
`</ FUNCTION_BODY>`
- 带返回值的函数体 `<FUNCTION_BODY name="函数名称" parameter="p1, p2, p3"`
`return="true" >`
`...`
`<let name="tmp"></let>`
`<op> tmp = p1 </op>`
`...`
`<op> $return= tmp </op>`
`</ FUNCTION_BODY>`

属性：	name 子函数函数体的名称 return 如果属性为真，系统会创建局部变量 \$return 。 函数的返回值会复制到变量中，在退出函数体时 该值传递给待调用的函数。 (可选) parameter 该属性列出了传递参数。参数相互之间应以逗号 分隔。 调用函数体时，在函数体中指定参数的值会复制 到列出的传递函数中。
-----	---

示例

```
<function_body name="test" parameter="c1,c2,c3" return="true">
  <let name="tmp">0</let>
  <op> tmp = c1+c2+c3 </op>
  <op> $return = tmp </op>
</function_body>
...
  <let name="my_var"> 4 </let>
<function name="test" return=" my_var " > 2, 3, 4</function>
  <print text="result=%d"> my_var </print>
...
  <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>
...
```

9.7.11 INCLUDE

说明

该指令中包含了 XML 说明。

编程

标签:	INCLUDE
句法:	<?include src="文件名称" ?>
属性:	src 用于文件名称的标签

9.7.12 LET

说明

该标签用于指定名称的局部变量。

变量范围一直延伸到上级 XML 块的结束处。

需要全局使用的变量应直接在标签 AGM 后创建。

说明

字符串变量的缺省设置

将格式化的文本作为数值传递，便可以将多行文本指定给字符串变量。如果一行要使用 Line Feed<LF> 结尾，应在行的末尾添加字符 ,\n'。

编程

标签: LET

句法: <let name = "<Name>" > 缺省设置</let>
<let name = "<Name>" type = "<变量类型>" > 缺省设置</let>

属性: name 用于变量名称的标签
type 允许用于变量类型的标签:

- 整数(INT)
- 双整数(DOUBLE)
- 浮点数(FLOAT)
- 字符串(STRING)

缺省设置:

如果未定义类型，系统会创建一个整数型变量。

```
<LET name = "VAR1" type = "INT" />
```

一个变量可以初始化一个数值。

```
<LET name = "VAR1" type = "INT" > 10 </LET>
```

如果将 NCK 或 PLC 变量的值保存在一个局部变量中，赋值指令会自动根据读入变量的格式调整格式。

permanent 如果属性为 true，则永久保存变量值。

该属性只针对全局变量！

句法:

```
<let name = "<Name>" type = "<变量类型>"  
permanent = "TRUE" > 缺省设置</let>
```

示例

```
<LET name="text" type = "字符串"> F4000 G94\n
  G1 X20\n
  Z50\n
  M2\n
</LET>
```

9.7.13 MSGBOX

说明

该标签用于打开信息窗口，窗口的返回值会引导不同分支。如果使用参考文本，信息窗口中会显示标签下定义的文本。

编程

标签:	MSGBOX	
句法:	<code><MSGBOX text="<文本>" caption="<标题>" retvalue="<变量>" type="<应答>" /></code>	
	<code><MSGBOX text="<文本>" caption="<\$\$文本参考>" retvalue="<变量>" type="<应答>" /></code>	
属性:	caption	用于信息窗口标题的标签
	retvalue	用于变量名称的标签，返回值会复制到该变量中。
	retval=0	0: OK
	retval=1	1: Cancel
	type	用于应答信息的标签
	type="btn_ok"	
	type="btn_cancel"	
	type="btn_okcancel"	

9.7.14 OP

说明

用于执行指令的标签：所有允许的运算符都可以执行。需要访问 NCK/PLC/驱动数据时，请在引号中键入完整的变量名称。

编程

标签: OP
句法: <op> 运算符</op>
属性: --

示例

```
<OP> tmpVar = "PLC/MB170" </OP>  
<OP> tmpVar = "PLC/MB170" + 5 </OP>
```

字符串处理

运算指令可以对字符串进行处理，并将结果字符串赋值给等式中指定的字符串变量。

为了区分文本表达式，应在文本前加上标志 `_T`。此外还能格式化变量值。格式化规定以标志 `_F` 开始，后面跟有格式指令。

接着给定变量地址。

示例

```
<LET name="缓冲" type="字符串"></LET>  
...  
...  
<op> 缓冲= _T"未格式化的值 R0= " + "nck/Channel/Parameter/R[0]" + _T" and " +  
_T"$$85051" + _T" 格式化的值 R1 " + _F%9.3f"nck/Channel/Parameter/R[1]" </op>
```

9.7.15 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[j] 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
		<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:	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

9.7 脚本语言的说明

9.7.16 PASSWORD

说明

该标签指定给一个设备后，没有勾选的选项旁会出现一个软键，它要求输入该设备的口令。字符串由 PLC 处理，结果通过选项数据告知 HMI。

编程

标签: **PASSWORD**
句法: <password refvar = "变量名称" />
属性: refvar 参考变量的名称

示例:

<password refvar="plc/db9900.dbd0" />

9.7.17 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

编程

标签: **PLC_INTERFACE**
句法: <plc_interface name = "数据的地址标识" />
属性: name 地址标识, 例如: plc/mb170

示例: plc/mb170

9.7.18 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。显示文本在系统中固定定义。

编程

标签: **POWER_OFF**

句法: `<power_off />`

属性: --

9.7.19 PRINT

说明

该标签用于在信息栏中输出文本，或将该文本复制到指定的变量中。如果文本包含有格式化标志，则会将变量值粘贴到相应的位置上。

- 字符串“%n”表示分行。
- 字符,%'表示格式化输入的变量值：
%[旗标] [宽度] [.小数位] 类型

参数	应用
旗标	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 (- 左对齐) • 前面加零 (0) • 使用空格符填充 (blank)
宽度	确定一个非负数最小输出宽度的依据。如果待输出值占用的位置少于依据所确定的位置，则使用空格符填充空缺的位置。 小数位： 使用浮点数时该可选参数用来确定小数点后面的位数。

参数	应用
类型	类型字符用来确定 PRINT 指令要输出哪些数据格式。该字符必须给定。 支持以下数据格式： <ul style="list-style-type: none"> • d: 整数值 • f: 浮点数 • s :字符串
参数值	其值应当插入到文本中的变量数量。变量类型必须与格式规定的相应类型标识符一致。

编程

标签: **PRINT**
 句法: <print name = "变量名称" text="文本%格式化"> 变量, ...</print>
 属性: name 保存文本的变量的名称。
 text 文本

9.7.20 **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。

9.7.21 ?up

说明

SinuCom Installer:

本段包含了 SinuCom Installer 的脚本语言。如果需要纳入 Update Installer 文件的代码，应使用指令INCLUDE (页 277)。

编程

标签:	?up
句法:	<?up <?include src="文件名称" ?> ?>
属性:	--

9.7.22 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。

窗口的组成部分有：

- 输入对话框
- 窗口标题
- 组合输入输出栏
- 文本显示
- 图片显示

说明

标签 <tag>	含义
CAPTION	用于窗口标题的标签： 句法： <caption> 标题</caption>
CLOSE	窗口信息： 在关闭窗口前执行该标签。
FORM	用于用户窗口的标签。 属性 color: 背景色的代码
INIT	窗口信息： 用于初始化窗口的标签。 在创建窗口后立即执行该标签。其中应建立窗口所有的输入单元和热链接。
PAINT	窗口信息： 用于显示窗口文本和图片的标签。 该标签在显示窗口时执行。
TIMER	窗口信息： 该标签循环调用。

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 /CONTROL>          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>
```

9.7 脚本语言的说明

9.7.23 BOX

说明

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。

编程

标签:	BOX										
句法:	<code><box xpos="X 位置" ypos = "Y 位置" width="X 宽度" height="Y 高度" color="颜色代码" /></code>										
属性:	<table><tr><td>xpos</td><td>X 方向的位置（零点位于左上角）</td></tr><tr><td>ypos</td><td>Y 方向的位置（零点位于左上角）</td></tr><tr><td>width</td><td>X 方向上的宽度（以像素为单位）</td></tr><tr><td>height</td><td>Y 方向上的高度（以像素为单位）</td></tr><tr><td>color</td><td>颜色</td></tr></table>	xpos	X 方向的位置（零点位于左上角）	ypos	Y 方向的位置（零点位于左上角）	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
xpos	X 方向的位置（零点位于左上角）										
ypos	Y 方向的位置（零点位于左上角）										
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										

9.7.24 CONTROL

说明

该标签用于创建窗口控制组件。

缺省设置: `fieldtype="edit"`数据可写。

编程

标签:	CONTROL		
句法:	<code><control name = "编辑 1" xpos = "X-定位" ypos = "Y-定位" refvar="NC-变量" hotlink="true" format="格式" display_format="FLOAT" /></code>		
属性:	<table><tr><td>name</td><td>字段名称: 每个字段都有一个相同名称的本地变量。</td></tr></table>	name	字段名称: 每个字段都有一个相同名称的本地变量。
name	字段名称: 每个字段都有一个相同名称的本地变量。		

xpos	X 方向的位置（零点位于左上角）
ypos	Y 方向的位置（零点位于左上角）
fieldtype	字段类型： <ul style="list-style-type: none"> • fieldtype="edit" • 数据可写。 • fieldtype="readonly" • 数据可读。 • fieldtype="combobox" • 显示标签，而不是数据。
format	其属性定义了说明变量的显示格式，例如左对齐或小数点后的位数。
display_format	其属性定义了说明变量的处理格式。在 PLC-Float 上读取数据时，其属性必须使用变量，因为要通过双字符完成读取。 允许使用下列数据格式： <ul style="list-style-type: none"> • FLOAT • INT • DOUBLE • STRING
hotlink	用于一旦发生改动便立即更新(=TRUE)的字段。
refvar	用于参考变量（NC/PLC/驱动变量）的标签

说明

周期性更新

属性“hotlink”会引起周期性更新对应的控制组件。

这表明：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。

示例

如果选择了字段类型“combobox (= 下拉框)”，则需要另外定义显示在窗口中的表达式。此时应使用标签<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。

句法： <item>表达式</item>

```
<control name = "按钮 1" 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>

; 如果要为表达式指定任意一个整数值，则要为标签 item 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>
```

9.7.25 IMG

说明

该标签用于显示目录中的像素图片： ../oem/sinumerik/hmi/dvm

- 位图应以 **BMP** 或 **PNG** 的格式保存。
- 所有文件名称应采用小写。
- 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。

编程

标签: **IMG**

句法: `<img name = "<名称>" xpos = " X 位置" ypos = " Y 位置" height="=Y 比例" width="X 比例" />`

属性:	name	Y 方向上的高度（以像素为单位）
	xpos	X 方向的位置（零点位于左上角）
	ypos	Y 方向的位置（零点位于左上角）
	width	X 方向的缩放比例（可选）
	height	Y 方向的缩放比例（可选）

9.7 脚本语言的说明

9.7.26 PROPERTY

说明

该标签用来确定操作单元的附加属性。

编程

标签:	PROPERTY
句法:	<code><property Attribut="<数值>" /></code>
属性:	max 最大输入值
	min 最小输入值
	default 缺省设置

示例

```
<control name = "编辑" xpos = "10" ypos = "10" width = "100" hotlink="true"
refvar="nck/Channel/GeometricAxis/actProgPos[1]" >
    <property min="0" />
    <property max="1000" />
</control>

<control name = "编辑 1" xpos = "10" ypos = "10" >
    <property min = "20" />
    <property max = "40" />
    <property default="25" />
</control>
```

9.7.27 REQUEST

说明

该标签仅在一种形式的 INIT 指令中有效。借助该标签可以在周期性读取服务（hotlink 热链接）中传送一个变量。

编程

标签:	REQUEST
句法:	<code><REQUEST name = "NC 变量" /></code>
属性:	name 地址标识

9.7.28 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>
```

9.7.29 TEXT

说明

该标签用于显示文本。

编程

标签: **TEXT**
句法: `<text xpos="X 位置" ypos = "Y 位置" color="颜色代码"> Text </文本>`
属性: `xpos` X 方向的位置 (零点位于左上角)
`ypos` Y 方向的位置 (零点位于左上角)
`color` 颜色

9.7.30 TYPE_CAST

描述

用于转换本地变量的数据类型的标签。

编程

名称: **TYPE_CAST**
句法: `<type_cast name="变量名称" type="新类型" />`
属性: `name` 变量名称
`type` 新数据类型
`convert` 转换位组合 (间接赋值)

示例

```
<let name="v1" type="string" >1234.566</let>  
<let name="result" type="double" />  
...  
<type_cast name="v1" type="double" />  
<op> result = v1 *23.5 </op>  
...
```

特殊情况:

如果将 PLC 双字作为浮点数处理，则必须对本地变量进行赋值。如需正确保存二进制值，则必须将本地变量的类型设为整型。

接着通过指令 TYPE_CAST 将其转换为浮点数。为了将整型值转换为浮点数，请使用属性 convert 替代 type。可将整型值的位组合转换为浮点值并将变量类型更改为浮点型。

```
<let name="plc_float" />  
<let name="result" type="float" />  
...  
...  
<timer>  
  
    <type_cast name="plc_float" type="int" />  
    <op> plc_float = "PLC/MD100"</op>  
    <type_cast name="plc_float" convert="float" />  
  
    <op> result = plc_float *1.2 </op>  
</timer>
```

9.7.31 UPDATE_CONTROLS

说明

该标签用于比较操作单元和参考变量。

编程

标签: **UPDATE_CONTROLS**

句法: `<update_controls type="<方向>"/>`

属性: **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。

9.7.32 参数的编址

参数编址

为了给 NC 变量、PLC 功能块或驱动数据编址，需要构成所需数据的地址标识。地址由路径段（组件名称）和变量地址构成。请使用斜线作为分隔符。

PLC 数据块的编址：

数据格式 f: **B:** 字节

W: 字

D: 双字

地址 x: 有效的 PLC 地址标识

位地址: **b =** 位号

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
M (f) x	位内存
V (f) x	变量

在位编址中省略了数据格式标识:

DBx.DBXx.b	数据块
Ix.b	输入端
Qx.b	输出端
Mx.b	位内存
Vx.b	变量

示例:

```
<data name = "plc/mb170">1</data>  
<data name = "plc/db9905.dbb0"> 0 </data>  
<data name = "plc/i0.1"> 1 </data>  
<op> plc/m19.2 = 1 </op>
```

NC 变量的编址

编址从路径段 **nck** 开始。随后是数据地址，其结构可以查阅 **OEM** 产品包的参数手册 2。

示例:

```
<let name = "tempStatus"></let>  
<op> tempStatus = "nck/channel/state/chanstatus" </op>
```

机床数据和设定数据的编址

可以使用 \$ 符号标出机床数据和设定数据，后面跟有数据名称。

- 机床数据:

\$Mx_<名称[索引, AX<轴编号>]>

- HMI 机床数据:

\$MxS_<名称[索引, AX<轴编号>]>

- 选件数据

\$Ox_<名称[索引, AX<轴编号>]>

- 设定数据:

\$Sx_<名称[索引, AX<轴编号>]>

\$SxS_<名称[索引, AX<轴编号>]>

结构		含义
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:		为字段编址时，该参数指定数据的索引。
AX<轴编号>:		数据为轴专用的数据时，该参数用于指定所需轴 (<轴>)。 可以选择利用“替代符号” \$<变量名称> 从一个局部变量 (如 AX\$lokaleVariable) 中读取轴索引:

示例:

```
<data name = "$MN_AXCONF_MACHAX_NAME_TAB[0] ">X1</data>
```

- 轴的直接编址:

```
<data name = "$MA_CTRLOUT_MODULE_NR[0, AX1] ">1</data>
```

- 轴的间接编址

```
<let name = "axisIndex"> 1 </let>
```

```
<data name = "$MA_CTRLOUT_MODULE_NR[0, AX$axisIndex] ">1</data>
```

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>
```

9.7.33 驱动对象的编址

驱动对象的句法 (DO)

编址从路径段 **drive** 开始。后面跟有驱动设备的指定：**CU** 或 **DC**。然后是待设置的参数。需要给单个对象编址时，请在参数后的尖括号中指定所需对象。

参数号 [do<DO 索引>]

示例：p0092 [do1]

说明

编号

驱动对象编号会与驱动窗口中使用的编号不同，因为 **CU**、**ALM** 以及所有连接的集线器等都一同参与了连续编号。

可以通过以下方式确定 **DO** 编号：

所有相连的驱动对象都显示在对应 **CU** 的字段 **p0978** 中。其中，字段的数值即驱动对象的插接位号。应首先确定所需插接位置的字段索引，然后加上一。得出的值便是编址所需的 **DO** 索引。

如果驱动对象上连接了一个 **NX**，应首先确定上一个 **CU** 驱动对象的索引，然后再加上 **NX** 驱动对象的索引。

或者利用“替代符号” **\$<变量名称>** 从局部变量（如 **DO\$lokaleVariable**）中读取驱动索引

示例：

```
<data name = "drive/cu/p0092">1</data>
<data name = "drive/dc/p0092[do1] " >1</data>
```

间接编址：

```
<let name = "driveIndex" > 0 </let>
<op> driveIndex = $ctrlout_module_nr[0, AX1] </op>
<data name = "drive/dc[do$driveIndex]/p0092">1</data>
```

NX 编址

一个 **NX** 模块相当于第二个 **CU**，该模块的编址也通过 **CU** 指定：在参数后的尖括号中指定所需 **NX** 索引（**CU** 索引加一）。

参数编号 [CU<CU 索引>

示例

```
<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 NX1 上 p107[0] 的值 -->
<op> p107_content = "drive/cu/p107[0,CU2]" </op>
  <print text="%d"> p107_content </print>
```

9.7.34 用于指令的 XML 标签

程序指令

以下标签可用于指令：

标签 <tag>	含义
IF	<p>附条件的指令(IF, THEN, ELSE)</p> <p>标签 THEN 和 ELSE 合并到标签 IF 中。</p> <p>标签 IF 后跟有条件，而条件在标签 CONDITION 中执行。运算结果会决定是否继续执行指令。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，Parser 执行 ELSE 分支。</p> <p>示例：</p> <pre> <IF> <CONDITION> plc/mb170 != 7 </CONDITION> <THEN> <OP> plc/mb170 = 7 </OP> ... </TEHN> <ELSE> </ELSE> </IF> </pre>
THEN	满足条件时的指令(IF, THEN, ELSE)。
ELSE	不满足条件时的指令(IF, THEN, ELSE)。

标签 <tag>	含义
FOR	<p>按下列方式执行 FOR 循环：</p> <ol style="list-style-type: none"> 1. 计算表达式初始化 (INIT)。 1. 作为布尔表达式计算表达式测试(CONDITION)。如果值为假 (FALSE)，则结束 FOR 循环。 2. 执行后续指令。 3. 计算表达式继续 (INCREMENT)。 4. 再次从步骤 2 开始。 <p>FOR (初始化、测试、继续) 指令</p> <p>句法：</p> <pre><FOR> <INIT>...</INIT> <CONDITION>...</CONDITION> <INCREMENT>...</INCREMENT> 指令 ... </FOR></pre>
BREAK	有条件中断一个循环
WHILE	<p>WHILE 循环用于多次执行指令串，只要满足条件。在执行指令串前对条件进行检查。</p> <p>while (测试) 指令</p> <p>句法：</p> <pre><WHILE> <CONDITION>...</CONDITION> 指令 ... </WHILE></pre>

标签 <tag>	含义
DO_WHILE	<p>DO ... WHILE 循环由一个指令段和一个条件构成。首先执行指令段中的代码，然后计算条件。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>do 指令 while (测试)</p> <p>句法: <DO_WHILE> 指令 ... <CONDITION>...</CONDITION> </DO_WHILE></p>
SWITCH	<p>指令 SWITCH 表示多项选择。表达式一次计算，并和常数相比。如果表达式与常数一致，则执行 CASE 指令中的相关指令。如果没有常数与表达式一致，则执行 DEFAULT 指令。</p> <p>句法: <SWITCH> <condition> 表达式</condition> <CASE value="<常数 1>" > 指令 ... </CASE> <CASE value="<常数 2>" > 指令 ... </CASE> <DEFAULT> 指令 ... </DEFAULT> </SWITCH></p>

9.8 字符串函数

函数概述

脚本语言提供了多个字符串函数。函数名称为备用名称，不能被覆写。

名称	功能
string.cmp	比较字符串 (string.cmp (页 306))
string.icmp	不区分大/小写的字符串比较(string.icmp (页 307))
string.left	选择左边 n 个字符(string.left (页 308))
string.right	选择右边 n 个字符(string.right (页 309))
string.middle	选择中间 n 个字符(string.middle (页 310))
string.length	确定字符串的长度(string.length (页 311))
string.replace	替换字符串 (string.replace (页 312))
string.remove	删除字符串 (string.remove (页 313))
string.insert	在指定索引处插入字符串(string.insert (页 315))
string.delete	删除字符串中的 n 个字符 (string.delete (页 314))
string.find	向前搜索字符串的子集(string.find (页 316))
string.reversefind	向后搜索字符串的子集(string.reversefind (页 317))
string.trimleft	删除左侧的空格(string.trimleft (页 318))
string.trimright	删除右侧的空格(string.trimright (页 319))

9.8.1 string.cmp

说明

该函数用于比较两个字符串。

当字符串相同时该函数提供返回值零；当第一个字符串小于第二个时，返回值小于零；当第二个字符串小于第一个时，返回值大于零。

编程

名称:	string.cmp
句法:	<code><function name="string.cmp" retvar ="<int var>" > str1, str2 </function></code>
参数:	str1 字符串
	str2 比较字符串
	rval 结果

示例

```
<let name="rval">0</let>
<let name="str1" type="string">A brown bear hunts a brown dog.</let>
<let name="str2" type="string">A brown bear hunts a brown dog.</let>
<function name="string.cmp" return="rval"> str1, str2 </function>

; 结果:  rval=0
```

9.8.2 string.icmp

说明

该函数用于在不区分大/小写的情况下比较两个字符串。

当字符串相同时该函数提供返回值零；当第一个字符串小于第二个时，返回值小于零；当第二个字符串小于第一个时，返回值大于零。

编程

名称:	string.icmp						
句法:	<code><function name="string.icmp" retvar ="<int var>" > str1, str2 </function></code>						
参数:	<table><tr><td><code>str1</code></td><td>字符串</td></tr><tr><td><code>str2</code></td><td>比较字符串</td></tr><tr><td><code>rval</code></td><td>结果</td></tr></table>	<code>str1</code>	字符串	<code>str2</code>	比较字符串	<code>rval</code>	结果
<code>str1</code>	字符串						
<code>str2</code>	比较字符串						
<code>rval</code>	结果						

示例

```
<let name="rval">0</let>
<let name="str1" type="string">A brown bear hunts a brown dog.</let>
<let name="str2" type="string">A brown Bear hunts a brown Dog.</let>
<function name="string.icmp" return="rval"> str1, str2 </function>

; 结果:  rval=0
```

9.8.3 string.left

说明

该函数选取字符串 1 最初的几个字符，并将它们复制到 Return 变量中。

编程

名称: **string.left**

句法: `<function name="string.left" return="< result string">"> str1, nCount
</function>`

参数: **str1** 字符串
 nCount 字符数量

示例

```
<let name="str1" type="string">A brown bear hunts a brown dog.</let>  
<let name="str2" type="string"></let>  
<function name="string.left" return="str2"> str1, 12 </function>  
  
; 结果: str2="A brown bear"
```

9.8.4 string.right

说明

该函数选取字符串 1 最后的几个字符，并将它们复制到 Return 变量中。

编程

名称: **string.right**

句法: `<function name="string.right" return="< result string">"> str1, nCount
</function>`

参数: **str1** 字符串
 nCount 字符数量

示例

```
<let name="str1" type="string">A brown bear hunts a brown dog.</let>  
<let name="str2" type="string"></let>  
<function name="string.right" return="str2"> str1, 10 </function>  
  
; 结果: str2="brown dog"
```

9.8.5 string.middle

说明

该函数从索引 `iFirst` 起提取字符串 `1` 中指定数量的字符，并将其复制到返回变量中。

编程

名称: **string.middle**

句法: `<function name="string.middle" return="< result string">> str1, iFirst, nCount </function>`

参数:

<code>str1</code>	字符串
<code>iFirst</code>	起始索引
<code>nCount</code>	字符数量

示例

```
<let name="str1" type="string">A brown bear hunts a brown dog.</let>
<let name="str2" type="string"></let>
<function name="string.middle " return="str2"> str1, 2, 5 </function>

; 结果: str2="brown"
```

9.8.6 string.length

说明

该函数提供字符串的字符数量。

编程

名称:	string.length				
句法:	<code><function name="string.length" return="< int var"> str1 </function></code>				
参数:	<table><tr><td><code>str1</code></td><td>字符串</td></tr><tr><td><code>length</code></td><td>结果</td></tr></table>	<code>str1</code>	字符串	<code>length</code>	结果
<code>str1</code>	字符串				
<code>length</code>	结果				

示例

```
<let name="length">0</let>
<let name="str1" type="string">A brown bear hunts a brown dog.</let>
<function name="string.length" return="length"> str1 </function>

; 结果: length=31
```

9.8.7 string.replace

说明

该函数使用新的字符串替换所有搜索到的子字符串。

编程

名称:	string.replace	
句法:	<code><function name="string.replace"> string, find string, new string </function></code>	
参数:	string	字符串
	find string	需替换的字符串
	new string	新字符串

示例

```
<let name="str1" type="string">A brown bear hunts a brown dog. </let>  
<function name="string.replace" > str1, _T"a brown dog" ,  
_T"a big salmon"</function>  
  
; 结果: str1="A brown bear hunts a big salmon."
```

9.8.8 string.remove

说明

该函数用于删除所有搜索到的子字符串。

编程

名称: **string.remove**
句法: `<function name="string.remove" > string, remove string </function>`
参数: **string** 字符串
 remove string 需删除的字符串

示例

```
<let name="index">0</let>
<let name="str1" type="string">A brown bear hunts a brown dog. </let>
<function name="string.remove" > str1, _T"a brown dog" </function>

; 结果: str1="A brown bear hunts ."
```

9.8.9 string.delete

说明

该函数用于从指定的起始位置开始删除定义数量的字符。

编程

名称:	string.delete						
句法:	<code>function name="string.delete"> string, start index , nCount </function></code>						
参数:	<table><tr><td>string</td><td>字符串</td></tr><tr><td>start index</td><td>起始索引</td></tr><tr><td>nCount</td><td>字符数量</td></tr></table>	string	字符串	start index	起始索引	nCount	字符数量
string	字符串						
start index	起始索引						
nCount	字符数量						

示例

```
<let name="str1" type="string">A brown bear hunts. </let>
<function name="string.delete" > str1, 2, 5 </function>

; 结果: str1="A bear hunts."
```

9.8.10 string.insert

说明

该函数用于在指定的索引位置插入一个字符串。

编程

名称:	string.insert
句法:	<code><function name="string.insert"> string, index, insert string </function></code>
参数:	string 字符串变量
	索引 需插入的字符的数量
	insert string 需插入的字符串

示例

```
<let name="str1" type="string">A brown bear hunts. </let>
<let name="str2" type="string">a brown dog </let>
<function name="string.insert"> str1, 19, str2 </function>

; 结果: str1="A brown bear hunts a brown dog."
```

9.8.11 string.find

说明

该函数用于在传输的字符串中搜索与子字符串一致的第一处地方。如果找到了子字符串，则该函数会提供到第一个字符的索引（从零开始），否则为 -1。

编程

名称: **string.find**

句法: `<function name="string.find" return="<int val>"> str1, find string
</function>`

参数: **string** 字符串变量
 find string 搜索的字符串。

示例

```
<let name="index">0</let>
<let name="str1" type="string">A brown bear hunts a brown dog. </let>
<function name="string.find" return="index"> str1, _T"brown" </function>

; 结果: index=2
```

9.8.12 string.reversefind

说明

该函数用于在传输的字符串中搜索与子字符串一致的最后一处地方。如果找到了子字符串，则该函数会提供到第一个字符的索引（从零开始），否则为 -1。

编程

名称:	string.reversefind
句法:	<code><function name="string.reversefind" return="<int val>"> str1, find string </function></code>
参数:	string 字符串变量 find string 搜索的字符串。

示例

```
<let name="index">0</let>  
<let name="str1" type="string">A brown bear hunts a brown dog. </let>  
<function name="string.reversefind" return="index"> str1, _T"brown" </function>  
  
; 结果: index=21
```

9.8.13 string.trimleft

说明

该函数用于删除字符串中打头的空格。

编程

名称: **string.trimleft**
句法: `<function name="string.trimleft" > str1 </function>`
参数: **str1** 字符串变量

示例

```
<let name="str1" type="string">            test trim left</let>
<function name="string.trimleft"> str1 </function>

; 结果: str1="test trim left"
```

9.8.14 string.trimright

说明

该函数用于删除字符串中后续的空格。

编程

名称: **string.trimright**
句法: `<function name="string.trimright" > str1 </function>`
参数: **str1** 字符串变量

示例

```
<let name="str1" type="string"> test trim right </let>
<function name="string.trimright" > str1 </function>

; 结果: str1=" test trim right"
```

9.9 三角函数

函数概述

脚本语言可提供多种三角函数功能。函数名称为备用名称，不能被覆写。

三角函数和反函数：

名称	功能
sin	正弦
cos	余弦
tan	正切
arcsin	反正弦
arccos	反余弦
arctan	反正切

正弦、余弦、正切的说明

该函数用于计算给定的值的正弦、余弦或正切值。

编程

名称: **sin**
 句法: `<function name="sin" return="<double val>"> double </function >`

名称: **cos**
 句法: `<function name="cos" return="<double val>"> double </function >`

名称: **tan**
 句法: `<function name="tan" return="<double val>"> double </function >`

参数: double 角度 (0° 到 360°)

示例

```
<let name= "sin_val" type="double"></let>
<function name="sin" return="sin_val"> 20.0 </function>
```

反正弦、反余弦、反正切的说明

该函数用于计算给定的值的反正弦、反余弦或反正切值。

arcsin、arccos 编程

名称:	arcsin	
句法:	<code><function name="arcsin" return="<double val>"> double </function ></code>	
名称:	arccos	
句法:	<code><function name="arccos" return="<double val>"> double </function ></code>	
参数:	double	x 在 -1 至 +1 范围内取值
取值范围:	arcsin	y 在 $-\pi/2$ 至 $+\pi/2$ 范围内取值
	arccos	y 在 0 至 π 范围内取值

arctan 编程

名称:	arctan	
句法:	<code><function name="arctan" return="<double val>"> double </function ></code>	
参数:	double	x 任意值
取值范围:		y 在 $-\pi/2$ 至 $+\pi/2$ 范围内取值

示例

```
<let name= "arccos_val" type="double"></let>
<function name="arccos" return="arctan_val"> 0.47 </function>
```


刀具管理

10.1 基础部分

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、夹爪) 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 NCK 提供当前刀具位置的信息, 并且记录通过 PLC 或 HMI 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 (M/T 版本) 的供货范围中。

功能“刀具管理的备用刀具” (姊妹刀具) 为**选件**。

另见

其他文献:

- SINUMERIK 828D 参数手册
- 刀具、刀库参数以及内部数据结构的详细说明请参见:

→ 功能说明, SINUMERIK 840D sl 刀具管理

本手册的 NCK 相关部分若包含在功能范围中, 则也适用于 SINUMERIK 828D。

本资料中 PLC 功能的说明和 NC 和 PLC 之间的通讯不适用于 SINUMERIK 828D。

- 刀具管理操作界面的设置请参见:
→ 调试手册, 基本软件和操作软件(IM9)

10.1.1 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：

- **HMI:**

- 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

- **NCK:**

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（姊妹刀具，如果存在的话）继续执行加工。

- **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
- 安全连锁
- 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

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

10.1.2 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）。

说明

刀具库的缺省设置

NCK 可管理的刀库的数量由系统设定为固定值。

- PPU24x.2: 库数量 = 3
- PPU26x.2: 库数量 = 3
- PPU28x.2: 库数量 = 4

由于总是需要有一个缓冲区和一个装载位，因此使用 PPU24x.2/PPU26x.2 时可管理一个真实刀库，使用 PPU28x.2 时可管理两个真实刀库。

在操作系统启动时，会根据各个工艺程序对以下刀具库进行缺省设置以及调整：

- 铣削：有 20 个刀位，带双夹爪和一个刀架的链式刀库
 - 车削：有 8 个刀位和一个刀架的转塔
-

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 and 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

MD22550 中的设置：\$MC_TOOL_CHANGE_MODE 中的赋值通常为 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，会分析刀具库刀位参数 \$TC_MPP4 中的位 4 ... 11（半刀位占用/预留）。

另见

更多信息请参见章节 刀具管理的机床数据 (页 348)。

10.1.3 手动装卸刀具

手动刀具

MD22562: \$MC_TOOL_CHANGE_ERROR_MODE 位 1 决定在换刀时是否能选择未分配刀具的附加刀具。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。

操作人员的职责

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。这种手动添加用于加工的刀具被称为“手动刀具”。

说明

用户必须通过 PLC 程序确保换刀遵循安全规程。

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）作为提示。在 PLC 用户程序发出换刀应答后，该报警删除。

手动刀具包含以下类型的刀具：

- 过尺寸刀具
- 不可在刀具库中保存的刀具
- 不允许通过设备的夹爪系统运行的刀具

10.2 NC/PLC 用户接口

概述

刀具管理从零件程序或 HMI 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。

程序组件和接口：

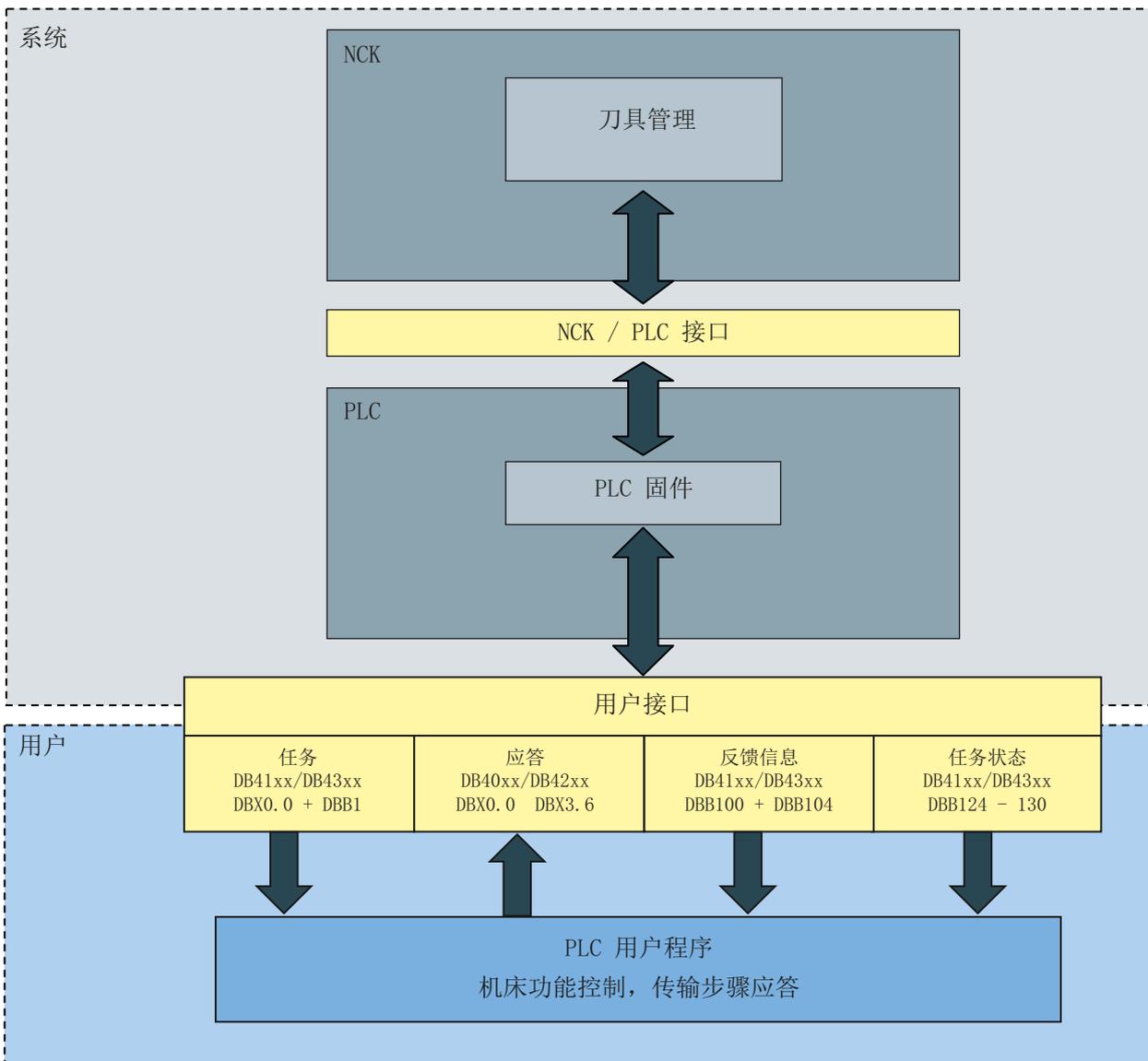


图 10-1 刀具管理的接口

用户接口一方面为卸刀、装刀、换位和刀库定位，另一方面为换刀提供独立的数据块。

10.2.1 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。

规则

任务根据以下规则分配至接口：

如果任务包含装载位（9999/x），则使用其接口。

否则会使用第 1 装载位（9999/1）的接口。

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
xx: 装载位	
DB40xx.DBX0.0 – DBX 3.6	PLC 用户程序： 对装刀、卸刀、刀具换位或刀库定位的应答
DB40xx.DBX9.0	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.DBX0.0	刀具管理： 装刀、卸刀、刀具换位或刀库定位任务
DB41xx.DBB1	刀具管理： 任务定义
DB41xx.DBW 6 – DBW34	任务描述
DB41xx.DBX100:0 DB41xx.DBX100.1	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
DB41xx.DBB104	刀具管理： 反馈信息故障状态
DB41xx.DBX108.0 - DB41xx.DBX111.6	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.DBW124 – DBW130	任务状态

任务

DB4100...41xx	从刀具管理发出的信号 [r]							
xx: 装载位								
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB0								任务
DBB1				NC 程序的 任务	定位	换位	卸刀	装刀
DBB2	备用							
DBB3	备用							
DBB4	备用							
DBB5	备用							
DBW6	源刀库号 (INT)							
DBW8	源刀位号 (INT)							
DBW10	目标刀库号 (INT)							
DBW12	目标刀位号 (INT)							
DBW14: HMI → PLC								装刀/卸刀, 不运行刀库

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。
- 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。

- 通过 NC 程序定位刀库：
定位任务由零件程序发出。
- 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。
- 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。

应答

DB4000...40xx		发送至刀具管理的信号 [r/w]						
xx: 装载位								
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB0	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
DBB1	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
DBB2	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
DBB3	备用	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
DBB4	备用							
DBB5	备用							
DBB6	备用							
DBB7	备用							
DBB8	备用							
DBB9								复位应答故障

信号描述:

- 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。
- 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

DB4100...41xx	从刀具管理发出的信号 [r]							
xx:装载位								
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB100							应答故障	应答正常
DBB101	备用							
DBB102	备用							
DBB103	备用							
DBW104	故障状态 (WORD)							
DBB106	备用							
DBB107	备用							
DBB108	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
DBB109	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
DBB110	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
DBB111	备用	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

信号描述:

- **应答正常 (DB41xx.DBX100.0) :**

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.DBX100.1) :**

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。

故障状态:

发生故障时, 故障状态 (DB41xx.DBB104) 为不等于零的诊断号。

状态	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、NCK 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 (DB41xx.DBB108 至 DBB111)

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]								
xx: 装载位								
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBW124	刀具的当前刀库号 (INT)							
DBW126	刀具的当前刀位号 (INT)							
DBW128	刀具的目标刀库号 (INT)							
DBW130	刀具的目标刀位号 (INT)							

10.2.2 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 NCK 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设置 (页 348) 中描述。

通过对“执行换刀”任务的结束应答可继续进行 NCK 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
xx: 主轴/刀套索引	
DB42xx.DBX0.0 – DBX 3.6	PLC 用户程序： “换刀准备”和“换刀执行”应答
DB42xx.DBX9.0	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。
DB43xx.DBX0.0	刀具管理：“准备换刀”和“执行换刀”任务
DB43xx.DBB1	刀具管理任务说明
DB43xx.DBW 6 – DBW34	任务描述
DB43xx.DBX100.0 DB43xx.DBX100.1	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.DBX100:0 DB43xx.DBX100:1	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.DBX100:0	刀具管理反馈信息：应答状态
DB43xx.DBB104	刀具管理反馈信息：故障状态
DB43xx.DBX108.0 - DB43xx.DBX111.6	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.DBW124 – DBW138	任务状态

任务

DB4300...43xx 从刀具管理发出的信号 [r]								
xx: 刀套								
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB0	--	--	--	--	--	--	--	任务
DBB1	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
DBB2	备用							
DBB3	备用							
DBB4	备用							
DBB5	备用							
DBW6	新刀具的源刀库号 (INT)							
DBW8	新刀具的源位置号 (INT)							
DBW10	备用							
DBW12	备用							
DBW14	备用							
DBW16	备用							
DBW18	旧刀具的目标刀库号 (INT)							
DBW20	旧刀具的目标位置号 (INT)							
DBW22	位置类型 (INT)							
DBW24	尺寸, 左侧 (INT)							
DBW26	尺寸, 右侧 (INT)							
DBW28	备用							
DBW30	备用							
DBB32	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	刀具已使能	激活的刀具

DB4300...43xx	从刀具管理发出的信号 [r]							
DBB33	新刀具的状态:							
	手动刀具	1:1 更换	--	原刀具	待装载	待卸载	禁用	周转区刀具的标识
DBW34	新刀具: NCK 的内部 T 号 (INT)							
DBW36	备用							
DBW38	备用							
DBW40	备用							
DBW42	备用							
DBW44	自由参数 1 (DWORD)							
DBW48	自由参数 2 (DWORD)							
DBW52	自由参数 3 (DWORD)							

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。
- 固定位置编码: 新刀具为固定位置编码。
- 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 HMI 上显示需要装载的刀具。
- 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
 - 位置类型：新刀具的位置类型
 - 大小：新刀具的尺寸（右侧、左侧、上侧、下侧）
 - 自由参数：3 个从子程序传输至 PLC 的自由参数。

应答

DB4200 ... 42xx	发送至刀具管理的信号 [r/w]							
xx: 刀套								
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB0	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
DBB1	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
DBB2	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
DBB3	备用	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
DBB4	备用							
DBB5	备用							
DBB6	备用							
DBB7	备用							
DBB8	备用							
DBB9								复位应答故障

信号描述:

- 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。

- 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。

反馈信息

DB4300...43xx	从刀具管理发出的信号 [r]							
xx: 装载位								
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBB100							应答故障	应答正常
DBB101	备用							
DBB102	备用							
DBB103	备用							
DBW104	故障状态 (WORD)							
DBB106	备用							
DBB107	备用							
DBB108	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
DBB109	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

DB4300...43xx	从刀具管理发出的信号 [r]							
DBB110	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
DBB111	备用	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

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) : 反馈“应答状态不正常”。

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。

故障状态:

发生故障时，故障状态（DB43xx.DBB104）为不等于零的诊断号。

状态	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	状态不允许更改位置（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、NCK 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]							
xx: 刀套								
字节	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
DBW124	新刀具的实际刀库号（INT）							
DBW126	新刀具的实际位置号（INT）							
DBW128	新刀具的目标刀库号（INT）							
DBW130	新刀具的目标位置号（INT）							
DBW132	新刀具的实际刀库号（INT）							
DBW134	新刀具的实际位置号（INT）							
DBW136	旧刀具的目标刀库号（INT）							
DBW138	旧刀具的目标位置号（INT）							

10.2.3 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的数据，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)。DB 9901 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)							
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)							

DB9900	固定传输步骤表 [r]
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 358)

10.3 刀具管理的机床数据

10.3.1 刀具管理设置

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。

机床数据号	名称	值	
10715[0]	M_NO_FCT_CYLE	6	(M 版本)
10716[0]	M_NO_FCT_CYLCLE_NAME	L6	(M 版本)
10717	T_NO_FCT_CYLCLE_NAME	TCHANGE	(T 版本)
17500	MAXNUM_REPLACEMENT_TOOLS	0	
20124	TOOL_MANAGEMENT_TOOLHOLDER	1	
20270	CUTTING_EDGE_DEFAULT	1	
20310	TOOL_MANAGEMENT_MASK	181400F	(T 版本)
		180400F	(M 版本)
22550	TOOL_CHANGE_MODE	0	(T 版本)
		1	(M 版本)
22560	TOOL_CHANGE_MCODE	206	
22562	TOOL_CHANGE_ERROR_MODE	0	(用于手动刀具)

MD20270: \$MC_CUTTING_EDGE_DEFAULT

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	
未编程时刀具刀沿的基本设置 (DWORD)	
> 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:

前提条件: 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。

零件程序中的顺序:

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

设置:

MD20310: \$MC_TOOL_MANAGEMENT_MASK		
激活刀具管理功能		
版本 T/M:		
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。
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, 则勾选了选件“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。

通道 MD52270: \$MCS_TM_FUNCTION_MASK

设置:

SD52270: \$MCS_TM_FUNCTION_MASK	
	刀具管理功能屏幕
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。
位 11:	在所有监控类型中重新激活刀具。 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。
位 12:	隐藏“刀具重新激活”。 “刀具重新激活”功能在操作界面中隐藏。

车削工艺的缺省设置

对于车削工艺，由两个机床数据定义功能：

- **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 389)

- **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

配置刀库

可通过调试工具或配置程序创建刀库配置。配置程序作为普通的零件程序选择和启动。

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

示例:

- 带双爪的链式刀库的配置 (页 394)
- 转塔刀库的配置 (页 382)

程序包含在工具盒光盘中。

文献

详细信息:

- SINUMERIK 828D 参数手册: 机床数据与接口信号
- SINUMERIK 802D sl/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。

10.3.2 MD20360 与 SD54215 的关联

刀具管理设置

刀具管理的设置在以下设定数据中进行:

SD54215: \$SNS_TM_FUNCTION_MASK_SET	
刀具管理功能屏幕	
位 0:	正在旋转的刀具用直径值显示: 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: 刀具 100 到 299、580、710、711、712、713、714 和 725。 在以下刀具的类型上, 刀沿参数 7 (外半径) 不用半径值, 而用直径值显示: 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, 旋转方向预设为 M4。
位 2:	在创建刀具时不提供建议名称。
位 3:	禁止输入已装载刀具的名称和类型。刀具装载后, 不能再更改其名称和类型。

SD54215: \$SNS_TM_FUNCTION_MASK_SET	
位 4:	通道不处于复位状态时，禁止输入已装载刀具的数据。
位 5:	累计计算刀具磨损值：输入的磨损数据累加在现有磨损值上。
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。可通过刀库号 9998 访问缓冲区。
位 11:	禁止在抓刀器刀位上创建新刀具。
位 12:	在执行功能“全部卸载”时不卸载测量刀具。

MD20360 与 SD54215 的关联

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		描述	建议
SD54215	↔ MD20360		
位 9	↔ 位 0	MD20360 位 0 置位后, SD54215 位 9 无效, 因为 NC 已经将横轴上的长度磨损量作为直径值提供。	我们不建议使用 MD20360 位 0 和位 1。 建议您使用 SD54215 位 8 和位 9。
位 8	↔ 位 1	MD20360 位 1 置位后, SD54215 位 8 无效, 因为 NC 已经将横轴上的长度尺寸作为直径值提供。	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。
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。
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。	MD20360 位 11 和位 12 可用于所有类型的刀具, 其中也包括车刀。无需在车床/铣床上进行该设置, 因为车刀的刀沿半径从不以直径值给定。 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。

10.3.3 JOG 模式下刀具测量

JOG 模式下，在刀具测量时，调出刀具

在特定应用（例如 JOG 模式下的测量循环）中，需要将特定刀具（例如 Fraeser_15, Duplo-Nummer 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。同时，应在新的动作发出之前进行测量。

刀具调用指令的替代指令，不考虑刀具状态，选中姊妹刀具号和刀套号，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。

TCA（“刀具名称”、姊妹刀具号和刀套号）

TCA 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：

- TCA 不能被替代（T 替代循环）。
- Der in MD10717：在使用 TCA 指令时，\$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）。

- 必须把机床功能 TCHANGE.SPF 转载到 TCA.SPF 循环中。TCA.SPF 循环在工具箱光盘上。

示例：

以下示例对流程进行了描述：

```

PROC TCA (STRING[64] _TOOL_NAME, INT _DUPLO, INT _TH_NO)
;VERSION: 01.00.07. 2009 年 10 月 22 日
; CHANGE: 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
    
```

说明

按照 MD20270 进行补偿选择：\$MC_CUTTING_EDGE_DEFAULT 与 T 指令操作相同。TCA 和 D 不可以在同一个组中进行编程。

在 JOG 模式下不用电子测头进行测量

操作方式 JOG 中的手动测量功能要求主轴运转，同时：

- 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 DB380x.DBB5006 中的位移动或定位。
- 在固定转速时使用工艺功能 DB380x.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。

10.4 PLC 程序块

10.4.1 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。

10.4.2 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。

10.4.3 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。
	6	“刀具移动”的结束应答,即刀具从真实刀库移动到缓冲区(夹爪、主轴),并在刀库中为该刀具保留原先的刀位。其他含义同状态 1。
	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 348)。
	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	6	7	99	105	201	204
新刀具	传输步骤	x	-	x	-	-	x	x	x
	• 从刀库	-	-	-	-	-	-	xx	xx
	• 从刀位	-	-	-	-	-	-	xx	xx
	• 向刀库	xx	-	xx	-	-	xx	xx	zz
	• 向刀位	xx	-	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	-

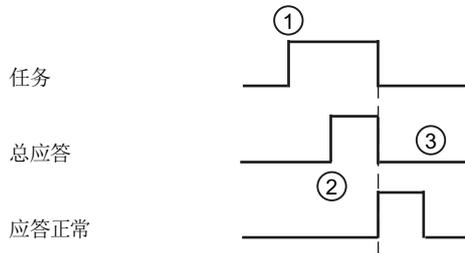
图例:

- 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。

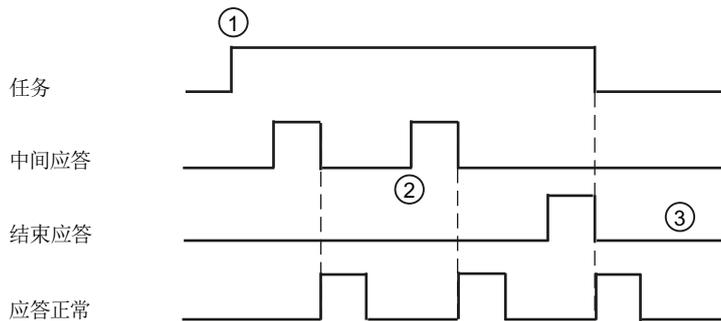
任意任务处理的典型时序图，采用总应答



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



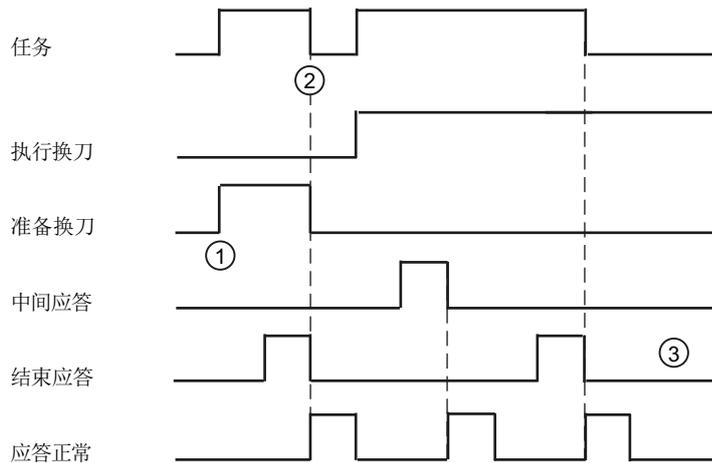
信号描述:

- 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:

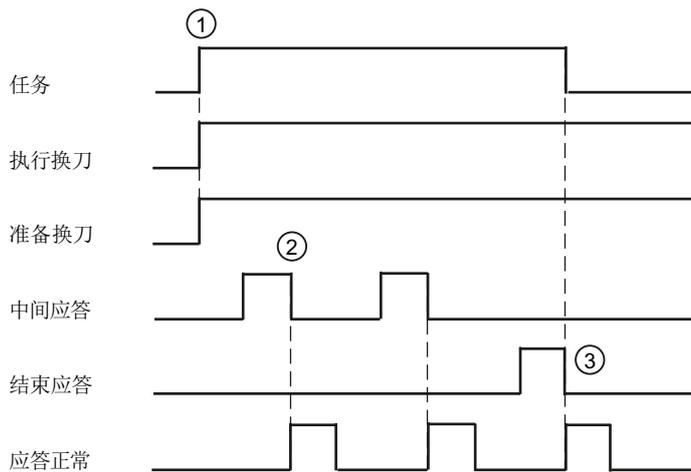
B43xx.DBB1 的接口特性以及 NC 程序段执行，会受到 MD20270 : \$MC_CUTTING_EDGE_DEFAULT 以及 MD20310: \$MC_TOOL_MANAGEMENT_MASK 位 5、6、7 和 8 中设置的影响。
这里描述的时序是机床数据的缺省设置。

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。

10.4.4 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夹爪 1
3	9998	1	9998	3	换刀：将刀具从主轴移动至夹爪 2
4	9998	2	9998	1	换刀：将刀具从夹爪 1 移动至主轴
5	9998	3	0	2	换刀：将刀具从夹爪 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
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。

10.4.5 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。

10.4.6 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 394)
- 车床应用示例 (页 382)

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。

10.4.7 刀库刀位信息

概述

通过现有的 NC 通讯接口（DB1200），可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 NCK 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	NCK 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 NCK 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	NCK 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 NCK 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	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	NCK 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

10.4.8 PI 服务：TMMVTL

功能

通过 PI 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 NCK 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。

应用

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。

10.5 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)

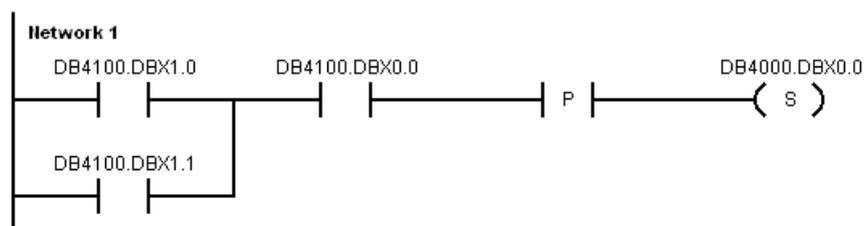


图 10-2 PLC 用户程序的编程

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。

下面描述了各种装刀方法：

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。

此方法总是可以执行，而与是否允许使用手动刀具无关。如果允许手动刀具，在通过主轴装刀时必须始终采用该方法。

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。刀具管理会选择可放入刀具的空刀位。

仅在未选中“手动刀具”功能时才能采用此方法，MD22562:

\$MC_TOOL_CHANGE_ERROR_MODE 位 1 = 0（缺省设置）。

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。

10.6 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

10.6 示例：更换手动刀具

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。

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)

10.7 车床应用示例

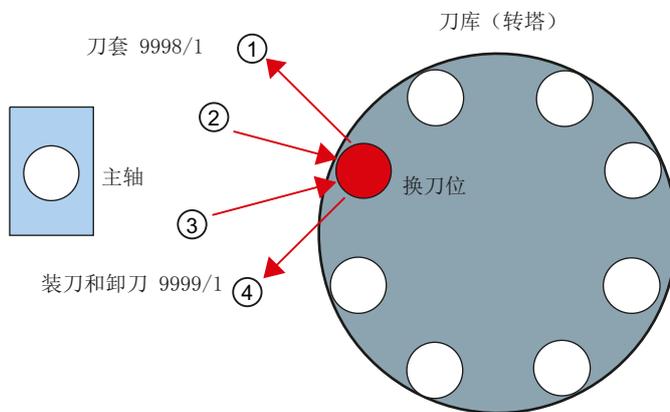
10.7.1 示例：车床，带转塔刀库（MAG_CONF_MPF）

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 ... 4 传输步骤

图 10-3 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
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 10-4 车床刀具列表

10.7.2 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)

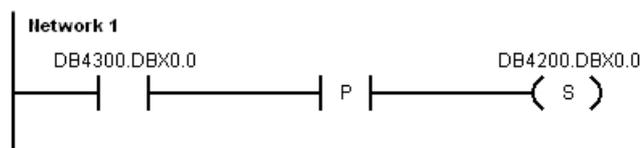


图 10-5 PLC 用户程序的编程

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)。

10.7.3 示例：车床的换刀循环

传输变量

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。

程序示例

前提是每个刀库移动都已通知给刀具管理。示例是以具有车削工艺缺省设置的机床数据为基础的。选件“用于刀具管理的备用刀具”无效。

T 替代循环的传输变量：

变量	说明
\$SC_T	刀具 T 编号（数字）
\$SC_T_Prog	布尔变量，显示 \$C_T 中的 T 字是否可用。
\$C_TS	刀具标识符（字符串）
\$C_TS_Prog	布尔变量，显示 \$C_TS 中的标识符是否可用。
\$C_TE	T 字的地址扩展名
\$C_D	设置的 D 号
\$C_D_Prog	布尔变量，显示 \$C_D 中的补偿号是否可用。
\$C_DL	设置的总/调整补偿
\$C_DL_Prog	布尔变量，显示 \$C_DL 中的补偿号是否可用。

在下面的示例中会使用 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
    
```

另见

该功能的使用与刀具管理无关，详见：基础功能手册，“工作方式组，通道，程序运行，复位属性(K1)”一章。

10.7.4 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。

10.7.5 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

10.7.6 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。

10.7.7 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

10.8 铣床应用示例

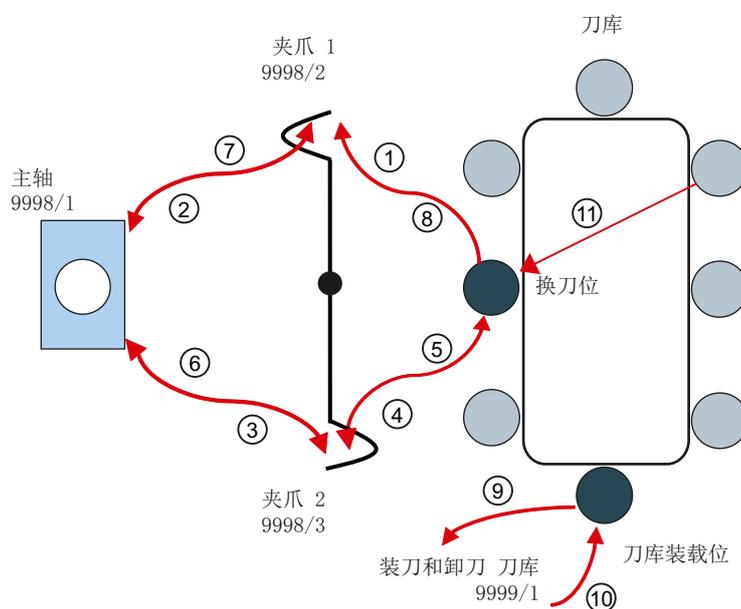
10.8.1 示例：铣床，带链式刀库和双夹爪(MAG_CONF_MPF)

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 10-6 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
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 382）
- 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
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 10-7 铣床刀具列表

10.8.2 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 376)
- 示例：更换手动刀具 (页 378)

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

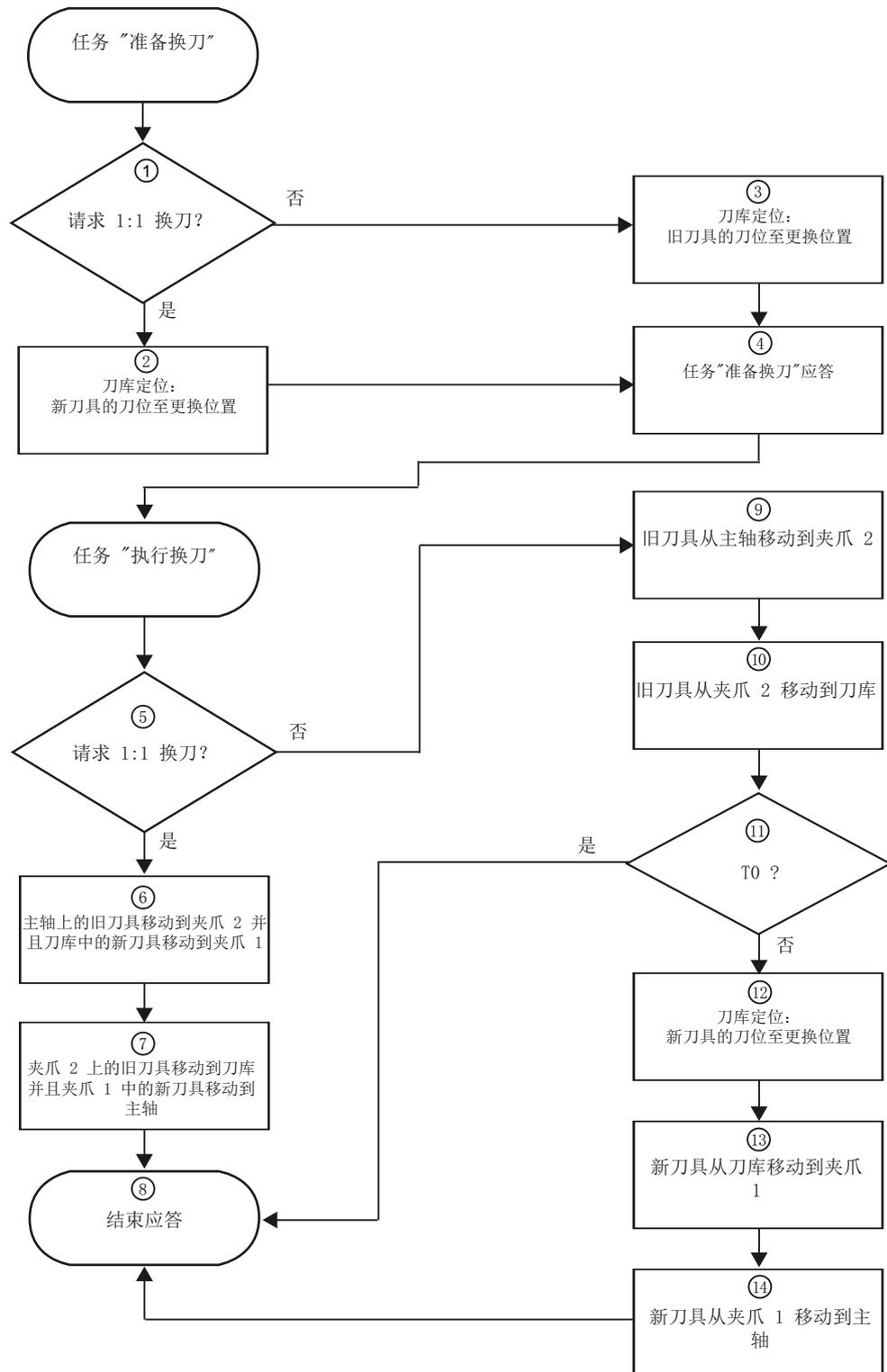
在 NC 程序中进行 T0 编程时会触发无新刀具的换刀。只将主轴的刀具传输到刀库中（主轴空运行）。

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 410)

“换刀”流程图



1: 1 换刀: 旧刀具会存放到新刀具的刀位上。

图 10-8 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（即使仍在主轴上）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。

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
4	DB4200.DBX0.4	101	0	204

发送给刀具管理的应答：

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。

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

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。

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： 应答“准备换刀”任务

← 上一步： 第 2 步或第 3 步

在很多系统上这时换刀准备已完成。

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 5

- 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

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： 步骤 5 PLC 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。

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

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	1	9998	3	旧刀具从主轴移动到夹爪 2 步骤 6 或 9

→ 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 中的新刀具移动到主轴

← 上一步: 步骤 6 PLC 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。

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
7	DB4200.DBX0.7	3	4	105

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
3 DB9900.DBW16	9998	2	9998	1	新刀具从夹爪 1 到主轴 步骤 7 或 14
4 DB9900.DBW24	9998	3	0	2	旧刀具从夹爪 2 移动到刀库 步骤 7 或 10

→ 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。

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
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： 步骤 5 PLC 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。

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
8	DB4200.DBX1.0	0	2	105

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
2 DB9900.DBW8	9998	1	9998	3	旧刀具从主轴移动到夹爪 2 步骤 6 或 9

→ 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: 步骤 9 PLC 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。

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
9	DB4200.DBX1.1	0	4	105

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
4 DB9900.DBW24	9998	3	0	2	旧刀具从夹爪 2 移动到刀库 步骤 7 或 10

→ 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

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
4	DB4200.DBX0.4	101	0	204

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。

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 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。

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
10	DB4200.DBX1.2	1	0	105

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

→ 继续执行步骤 14

步骤 14: 新刀具从夹爪 1 移动到主轴

← 上一步: 第 13 步

PLC 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。

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
11	DB4200.DBX1.3	3	0	105

传输步骤	从		向		注释
	刀库	刀位	刀库	刀位	
3 DB9900.DBW16	9998	2	9998	1	新刀具从夹爪 1 到主轴 步骤 7 或 14

→ 继续执行步骤 8

10.8.3 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 ≠ 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 ⑧
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
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 400)中的流程图

10.8.4 示例： 铣床的换刀循环

程序示例

```

PROC L6 SAVE DISPLOF
;-----
; 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
    G0                       ; 回到换刀点
    G75 Z=0
    WAITS(1)
  ENDIF
ELSE
ENDIF
;
; 换刀： 刀具管理和 PLC
M206
M17
;-----
; ENDE
;-----

```

Easy Archive

11.1 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。

西门子提供的系统存档可以是例如 NCK 的新版本或 HMI 集成版本或者也可以包括循环 Hotfix。

<p>注意</p> <p>系统数据的保护</p> <p>HMI, NCK, 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。</p> <p>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。</p> <p>使用 USB 设备存储用户数据时，数据大小不可超过 4 MB!</p>

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： 828D TE 或 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。

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。

11.2 通过存档来备份数据

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“调试存档”
 - 创建调试存档
 - 导入调试存档

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

11.2 通过存档来备份数据

组件	数据
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： 机床厂商的应用程序 • 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。

说明

调试存档会根据数据类（数据类型 ARD）进行保存。驱动数据作为二进制数据保存，不能进行编辑。

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

注意
<p>USB 设备 USB 设备不适宜作为永久保存媒介。</p>

11.3 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：

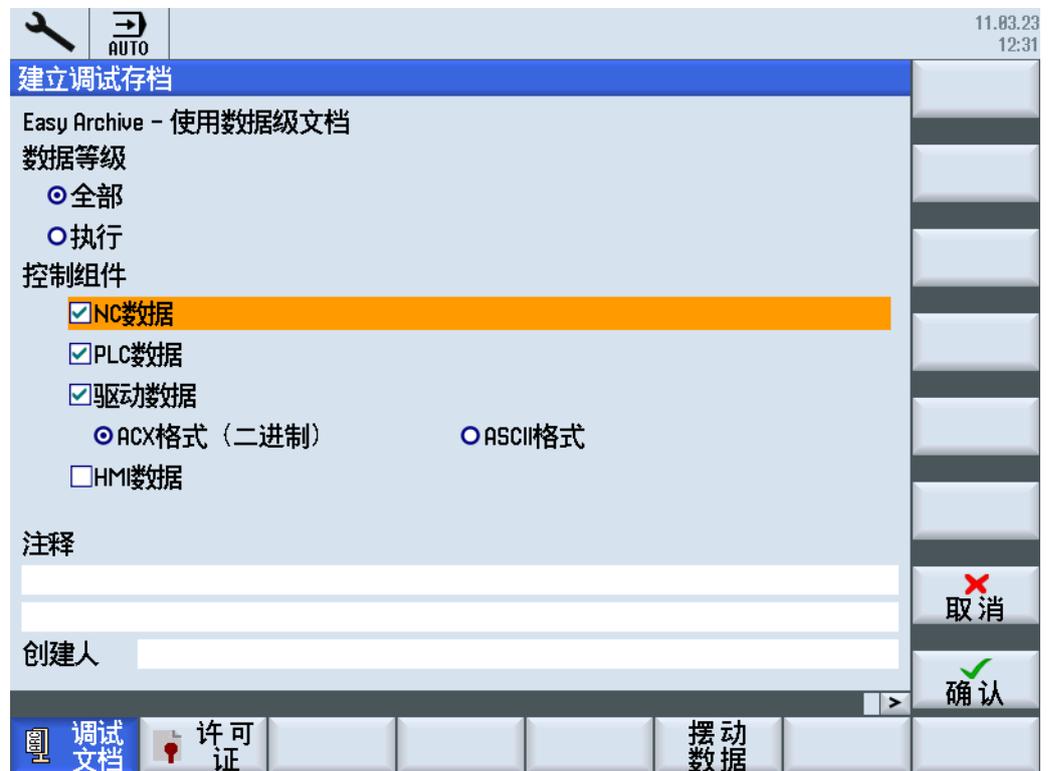


图 11-1 创建调试存档

11.3 创建调试存档的步骤

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。

11.4 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。

11.5 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。

另见： MD11210 \$MN_UPLOAD_MD_CHANGES_ONLY

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

MD11210 的作用

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。

11.6 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 自定义数据（机床数据，制造商循环）和操作人员数据（零件程序，刀具补偿）。自定义数据还划分为一类在所有类型的机床上都相同的数据，和另一类机床各自定义的数据。

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。自定义数据会在创建在原型机床上。该原型机床的自定义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。

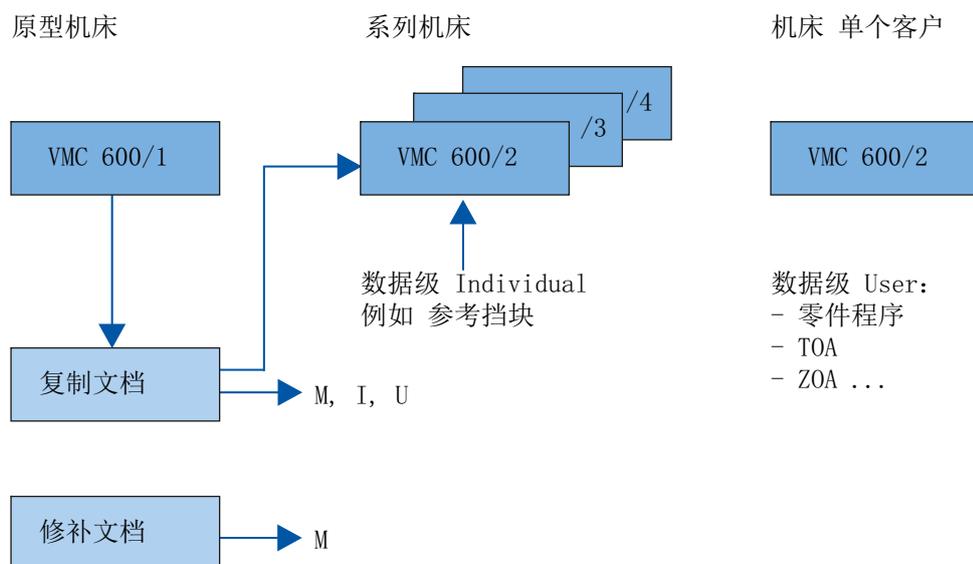


图 11-2 用户示例

例如，测量参考挡块以及滚珠丝杠并作为各自的自定义数据输入。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，则该机床独立的自定义数据都会被原型机床的独立自定义数据所覆盖。

对于 SINUMERIK 828D，机床制造商的自定义数据（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，则可以保证自定义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。

11.6 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自定义数据和用户数据，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。

11.7 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。

参数描述



图 11-3 设置参数

参数	允许的值
协议:	RTS/CTS (缺省设置) Xon/Xoff
传输:	标准 断点传输 (缺省设置)
波特率:	19200 (缺省设置) 300 ... 最大 19200
存档格式:	穿孔带 二进制格式 (PC 格式) → 只使用 RTS/CTS 协议
接口:	COM1
奇偶性:	无 (缺省设置) 偶 奇
停止位:	1 (缺省设置) 2
数据位:	5 位 6 位 7 位 8 位 (缺省设置)
Xon (仅在设置 Xon/Xoff 时)	11
Xoff (仅在设置 Xon/Xoff 时)	13
传输结束 (hex)	1a
时间监控 (秒)	4

禁用 V24

设置以下机床数据，禁用数据交换的接口：

- MD51233 \$MNS_ENABLE_GSM_MODEM = 0 (Voreinstellung)

允许 V24 接口进行数据交换。

- MD51233 \$MNS_ENABLE_GSM_MODEM = 1

此后和数据交换相关的软键不在显示。

该设置激活了 GSM 调制解调器。

A.1 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
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
日语	jpn
荷兰语	nld
波兰语	plk
罗马尼亚语	rom
俄语	rus
斯洛伐克语	sky
斯洛文尼亚语	slv
瑞典语	sve
土耳其语	trk

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	
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
BTSS	操作面板接口	
CEC	交叉误差补偿	
CNC	Computerized Numerical Control	计算机数字控制
DB	PLC 中的数据块	
DBB	PLC 中的数据块“字节”	
dbSI	drive based Safety Integrated	驱动内部集成的安全功能
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
HSC	高速切削	
HW	硬件	
IGBT	Insulated Gate Bipolar Transistor	
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
IPO	插补器	
IRT	Isochronous Real Time	等时同步通讯
ISO	International Standardization Organization	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
MAIN	Main program	主程序 (OB1, PLC)
MB	兆字节	
MCP	Machine Control Panel	机床控制面板
MCS	机床坐标系	
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
MLFB	产品订货号	
MPF	Main Program File	主程序 (NC 零件程序)
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	
PG	编程器	

A.2 缩略语列表

PI	程序等级	
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
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 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
本铣削版



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



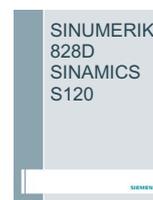
功能手册
- Safety Integrated



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

词汇表

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 → 许可密钥

DRIVE CLiQ

Drive Component Link with IQ 的缩写。

它是连接 SINAMICS 驱动系统各个组件的通讯系统，可连接的组件有：→ 控制单元、→ 电源模块、→ 电机模块、→ 电机和转速/位置编码器。

DRIVE CLiQ 基于工业以太网，使用双扭线电缆。除了接收和发送信号，也可以通过 DRIVE CLiQ 电缆提供 +24 V 电源。

编码器

用于采集并提供位置值用于后续电子处理。根据机械规格，编码器可以安装在 → 电机中（→ 电机编码器）或者安装在外部机械上（→ 外部编码器）。根据运行方式分为旋转式编码器（有时候也叫“旋转编码器”）和位移编码器（例如 → 直线量尺）。根据测量值的提供方式分为 → 绝对值编码器（代码编码器）和 → 增量编码器。参见 → 增量编码器 TTL/HTL → 增量编码器 sin/cos 1 Vpp → 旋转变压器

编码器模块

即 **Sensor Module**：硬件模块，用于计算转速/位置编码器信号，并在 → DRIVE CLiQ 插口上输出计算出的、数字格式的实际值。一共有 3 种机械规格的编码器模块：

- **SMCxx = Sensor Module Cabinet-Mounted** = 卡装在控制柜中的编码器模块。
- **SME = Sensor Module Externally Mounted** = 安装在控制柜外、防护等级更高的编码器模块。

参数

驱动系统内可变的参数，用户可以读取这些参数，部分也可写入。在 → **SINAMICS** 中，参数满足所有在 → **PROFIdrive** 协议中针对驱动参数的规定。参见 → 显示参数 → 设置参数

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
- 订货号
- → 许可证号

电机

从运动方向的角度划分，可由 → **SINAMICS** 控制的电机很大程度可分为旋转电机和直线电机；从电磁工作原理的角度划分，可分为同步电机和异步电机。在 **SINAMICS** 上，电机连接到一个 → 电机模块上。参见 → 同步电机 → 异步电机 → 电机编码器 → 外部编码器

电机编码器

指内装在电机中或安装在电机上的 → 编码器，例如：旋转编码器、→ 增量编码器 TTL/HTL 或 → 增量编码器 $\sin/\cos 1 V_{pp}$ 。该编码器用于测定电机转速。该编码器应用在同步电机上时，还可以测定转子位置角（即电机电流的换向角）。在不带附加 → 直接位置测量系统的驱动上，它也可以用作位置控制中的 → 位置编码器。除了电机编码器外，还有 → 外部编码器用于 → 直接测定位置。

电机模块

即 **Motor Module**，是为相连的电机提供电能的功率部件（DC-AC 逆变器）。电能经过 → 驱动装置的 → 直流母线提供给电机。电机模块必须通过 → **DRIVE CLiQ** 与控制单元连接在一起，后者中存储有用于电机模块的控制和调节功能。电机模块分为单电机模块和 → 双电机模块。

电源

指变频器设备的输入部件，用于生成直流母线电压，提供给一个或多个 → 电机模块以及所有必要组件如：→ 电源模块、熔断器、电抗器、电源滤波器和固件，有时还需要给 → 控制单元中的相关计算功率部件提供电压。

电源模块

即 **Line Module**，是一个功率部件，它将三相电源电压转换为直流母线电压，提供给一个或多个 → 电机模块使用。在 **SINAMICS** 中，一共有三种类型的电源模块：→ 基本型电源模块、→ 调节型电源模块和 → 非调节型电源模块。

电源模块和所需组件如 → 电源电抗器，→ 控制单元中的相关计算功率部件、开关装置等构成的功能整体被称为 → 基本型电源、→ 调节型电源和 → 非调节型电源。

调节型电源模块（ALM）

可调节、自控的供电/回馈单元（通过供电/回馈模块中的 **IGBT**），它为 ->电机模块提供直流母线电压。

非调节型电源模块（SLM）

即 **Smart Line Module**，不可调节的供电/回馈单元，供电通过二极管整流桥进行，而稳定的、电网控制的回馈则通过 **IGBT** 进行。非调节型电源模块为 → 电机模块提供直流母线电压。

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。

基本型电源

→ 基本型电源模块以及所需的附加组件（滤波器，开关装置等）构成的供电功能总称。

基本型电源模块

即 **Basic Line Module**：不可调节的供电单元（二极管或者晶闸管，没有回馈能力），用于电源电压的整流，形成 → 直流母线。

控制单元

中央控制和调节模块。有下列几种控制单元：

- **SIMOTION** 控制单元，如 **D425** 和 **D435**
- **SINAMICS** 控制单元，如 **CU320**
- **SINUMERIK solution line** 控制单元，如 **NCU**, **PPU**

驱动

驱动是由电机（电气或者液压的）、执行机构（变频器，阀门）、控制系统、测量系统和供电装置（电源，蓄压器）组成的总单元。电气驱动系统可分为变频器系统和逆变器系统。在用户看来，变频器系统（例如：→ MICROMASTER 4）包含了供电、执行机构和控制系统；而逆变器系统（例如：→ SINAMICS S）上则由 → 电源模块提供电源，它生成直流母线电压，提供给 → 逆变器(→ 电机模块)。控制系统（→ 控制单元）安装在一个独立的装置内并通过 → DRIVE CLiQ 与其它组件连接。

驱动参数

驱动轴的参数，包含例如附属的控制器参数、电机/编码器数据。上一级的工艺功能（定位、斜坡函数发生器）参数与此相反称作“应用参数”。

驱动对象(Drive Object DO)

驱动对象是一个独立、封闭的软件功能，它包含独立的 → 参数，有时还包含独立的 → 故障与 → 警告。驱动对象可以是标准配置，例如：On Board I/O；可单路设立，例如：端子板 30, TB30，或者也可以多路设立，例如 → 伺服控制”。一般情况下每个驱动对象都有自己的参数设置和诊断窗口。

驱动系统

驱动系统是一个产品系列中（如 SINAMICS）属于驱动的组件的总称。例如：一个驱动系统包含 → 电源模块、→ 电机模块、→ 编码器、→ 电机、→ 端子模块和 → 编码器模块以及扩展组件，如电抗器、滤波器、电缆等等。

驱动装置

所有通过 → DRIVE CLiQ 连接的组件的总单元，执行驱动任务时需要这些组件：→ 电机模块 → 控制单元 → 电源模块以及所需的 → 固件和 → 电机，但不包含扩展组件，如滤波器和电抗器。在一个驱动装置中可以具有多个 → 驱动。参见 → 驱动系统

驱动组

一个驱动组由一个 → 控制单元和通过 → DRIVE CLiQ 连接的 → 电机模块和 → 电源模块组成。

驱动组件

指通过 → DRIVE CLiQ 连接到 → 控制单元或其他模块的硬件组件。驱动组件有：→ 电机模块、→ 电源模块、→ 电机、→ 编码器模块和 → 端子模块。控制单元和相连的驱动组件构成的整体被称为 → 驱动装置。

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。

双电机模块(DMM)

在一个双电机模块（Double Motor Module）上能够连接并运行两个电机。参见 → 电机模块。

伺服控制

这种控制类型可以实现高精度、高动态的 → 电机运行（带 → 电机编码器）。除了转速控制外，其中也包含位置控制。

伺服驱动

电气伺服驱动由电机、→ 电机模块和 → 伺服控制系统以及转速/位置编码器组成。通常这种驱动的工作精度非常高，动态响应也很快。它可以用于最高达 100 ms 的时钟周期。它通常具有很强的短时过载能力，因此能够实现极快的加速过程。伺服驱动有旋转驱动和直线驱动两种。

外部编码器

位置编码器，不是内置在或外装在 → 电机上，而是直接或通过一个机械元件安装在加工机械上。外部编码器（外装编码器）可以直接确定位置。

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。

许可密钥

许可密钥是所有 → 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。

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

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。

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。

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，具有唯一性。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

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。

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

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。

索引

符号

\$MC_CUTTING_EDGE_DEFAULT (MD20270), 349
\$MC_TOOL_CHANGE_ERROR_MODE
(MD22562), 328, 348, 377, 378
\$MC_TOOL_CHANGE_MCODE (MD22560), 348
\$MC_TOOL_CHANGE_MODE (MD22550), 327, 348
\$MC_TOOL_MANAGEMENT_MASK (MD20310), 350
\$MC_TOOL_MANAGEMENT_TOOLHOLDER
(MD20124), 348
\$MCS_TM_FUNCTION_MASK (MD52270), 351
\$MN_M_NO_FCT_CYCLE_NAME (MD10716), 348
\$MN_MAXNUM_REPLACEMENT_TOOLS
(MD17500), 348
\$MN_PLC_DEACT_IMAGE_LADDR_IN (MD12986), 85
\$MN_PLC_DEACT_IMAGE_LADDR_OUT
(MD12987), 85
\$MN_T_NO_FCT_CYCLE_NAME (MD10717), 352
\$MN_UPLOAD_MD_CHANGES_ONLY
(MD11210), 420
\$MN_USER_DATA_INT (MD14510), 255, 281
\$MN_USER_DATA_PLC_ALARM (MD14516), 59
\$MNS_ACCESS_RESET_SERV_PLANNER
(MD51235), 250
\$SNS_TM_FUNCTION_MASK_SET (SD54215), 353

1

1:1 换刀, 400

A

Access MyMachine /P2P, 13
Advanced Surface, 95

C

CF 卡, 55
CYCLE832, 93

D

DB1800, 240, 241
DB9903, 238
DB9904, 239
DRIVE-CLiQ, 124
拓扑规则, 174

E

Easy Archive, 421
Easy Extend, 251
选件位, 255
EE_IFC (DB9905), 253
Ethernet 接口, 21

G

GUD, 36

I

IP 地址, 87
ISO 语言, 92

- J**
- JOG 模式下测量
 刀具调用, 356
 使用旋转主轴, 357
- M**
- MD10715[0]
 M_NO_FCT_CYCLE, 348
- MD10716[0]
 M_NO_FCT_CYCLE_NAME, 348
- MD10717
 \$MN_T_NO_FCT_CYCLE_NAME, 352
- MD11210
 \$MN_UPLOAD_MD_CHANGES_ONLY, 420
- MD12986
 PLC_DEACT_IMAGE_LADDR_IN, 85
- MD12987
 PLC_DEACT_IMAGE_LADDR_OUT, 85
- MD14510
 \$MN_USER_DATA_INT[i], 255, 281
- MD14516
 USER_DATA_PLC_ALARM, 59
- MD17500
 MAXNUM_REPLACEMENT_TOOLS, 348
- MD20124
 TOOL_MANAGEMENT_TOOLHOLDER, 348
- MD20270
 CUTTING_EDGE_DEFAULT, 349
- MD20310
 TOOL_MANAGEMENT_MASK, 350
- MD22550
 TOOL_CHANGE_MODE, 327, 348
- MD22560
 TOOL_CHANGE_MCODE, 348
- MD22562
 TOOL_CHANGE_ERROR_MODE, 328, 377, 378
- MD51235
 ACCESS_RESET_SERV_PLANNER, 250
- MD52270
 TM_FUNCTION_MASK, 351
- N**
- NC 指令 TCA, 356
- NCK 变量, 371
- P**
- PI 服务, 374
- PLC 用户报警, 57
- PLC 用户程序, 324
 调整, 369
- PLC 固件, 325
- R**
- RTC 电容, 20
- S**
- SD54215
 TM_FUNCTION_MASK_SET, 353
- T**
- TCP/IP, 21
- TMMVTL (PI 服务), 374
- Toolbox, 13
- U**
- USB 设备, 416

V

V24

- 参数, 423
- 取消, 424
- 接口, 423

X

XML

- 运算符, 268
- 指令, 302
- 标签, 269
- 特殊字符, 268

XML 标签

- ?up, 285
- AGM, 269
- BOX, 288
- CAPTION, 286
- CLOSE, 286
- CONTROL, 288
- CONTROL_RESET, 270
- DATA, 271
- DATA_ACCESS, 271
- DATA_LIST, 272
- DEVICE, 269
- DRIVE_VERSION, 273
- FILE, 273
- FORM, 286
- FUNCTION, 275
- FUNCTION_BODY, 276
- IMG, 291
- INCLUDE, 277
- INIT, 286
- LET, 278
- MSGBOX, 279
- NAME, 269

- OP, 280
- OPTION_MD, 281
- PAINT, 286
- PASSWORD, 282
- PLC_INTERFACE, 282
- POWER_OFF, 283
- PRINT, 284
- PROPERTY, 292
- REQUEST, 293
- SET_ACTIVE, 269
- SET_INACTIVE, 269
- SOFTKEY_CANCEL, 294
- SOFTKEY_OK, 294
- START_UP, 269
- TEST, 269
- TEXT, 295
- TYPE_CAST, 295
- UID, 269
- UPDATE_CONTROLS, 297
- VERSION, 269
- WAITING, 284

二划

- 刀库, 326, 365
 - 配置, 390
- 刀库列表, 326
- 刀具列表, 326
- 刀具管理, 323
 - MD20360 / SD54215 关联, 353

三划

- 三角函数, 320
- 口令
 - 设置, 45
 - 修改, 45

四划

中间应答, 359, 360
公司网络, 21
反馈信息, 334
开机调试档案文件, 414
 创建, 417
 读入, 419
引脚布局
 数字量输入/输出端, 180
手动刀具, 328

五划

外设模块的 DIP 开关, 87
用户示例, 421
用户视图, 91
用户接口, 329
电机数据组直接选择, 167
示例
 车床, 382
 更换手动刀具 (1), 378
 更换手动刀具 (2), 379
 铣床, 394
 零件程序 车削, 383
 零件程序 铣削, 395

六划

任务状态, 344
优化方案, 192
优化目标
 阻尼最佳, 193
 适当抗干扰性, 193
 最佳抗干扰性, 193
传输步骤, 365
传输步骤表
 可变, 346
 固定, 345

字典

 导入, 48
 编辑, 48
字符串函数, 305
存取级别, 43
存档
 存储区域, 416
机床数据, 89
 生效条件, 91
 单位, 90
网络连接, 26
许可证, 52, 56
许可证书 (CoL), 55
许可证号, 56
许可证密码, 52
许可密钥, 56
设定数据, 89
设置日期/时间, 46
过程数据, 150

七划

串行接口, 423
初始数据表, 238, 239
宏, 36
应答
 WZV, 333
 同步, 359
 存取级别, 250
 过程, 358
 状态, 360
 规则, 370
 维护任务, 240, 250
应答步骤, 368
应答步骤表, 347
应答禁止, 242
报文类型, 150

报警

- 记录, 61
- 定义变量, 59
- 结构, 58
- 语种缩写, 425
- 筛选事件, 61

系统语言, 47

邻接刀位, 328

驱动

- 互联, 182
- 参数, 151
- 配置, 124

驱动软件/调试软件 STARTER, 14

八划

供电, 131

定义文件, 36

拓扑规则 DRIVE-CLiQ, 174

直接连接, 26

转塔刀库, 327

九划

信息, 异步, 359

八划

变量

- T 编号, 373
- 刀位类型, 371
- 位置状态, 372

九划

总应答, 360

测头, 184

结束应答, 359, 360

语种缩写, 425

选件, 52, 56

选择方案

插补, 199

十划

圆度测试

优化示例 1, 218

优化示例 2, 219

优化示例 3, 220

执行测量, 216

设置参数, 216

位置设定值滤波器的示例: , 217

保存参数, 221

保存图形, 221

换刀, 337, 400

结束应答, 337

配置测量系统, 143

高速切削 (HSC), 93

十一划

维护周期, 250

维护管理器

标准模式, 246

配置模式, 245

十二划

硬件序列号, 56

缓冲区, 326

编址

DO, 300

GUD, 300

MD, 299

NC 变量, 298

NX, 301

车削和铣削

开机调试手册, 03/2013, 6FC5397-3DP40-3RA1

设定数据, 299

参数, 297

编程工具 (Programming Tool) , 13

装载刀库, 326

链式刀库, 327

十三划

数字量输入/输出端 X122, 179

数字量输入/输出端 X132, 179

数据区, 415

维护管理器, 237

数据级, 34

数据备份, 414

数据组

电机 (MDS) , 161

删除, 168

驱动 (DDS) , 161

修改, 171

添加, 162

编码器 (EDS) , 161

输入法编辑器 (IME), 47

十六划

激活 GSM 调制解调器, 424