

SIEMENS

SINUMERIK 840Di sl/840D sl/840D

基础软件和 HMI 高级版

开机调试手册

前言

HMI 高级版

1

补充操作界面

2

在线帮助

3

编制外语文本

4

PCU 基础软件 V8.2

5

PCU 基础软件 V7.6

6

附录

A

适用于：

控制系统

SINUMERIK 840Di sl/840DiE sl

SINUMERIK 840D sl/840DE sl

SINUMERIK 840D/840DE

软件版本

HMI 高级版 V7.5

01/2008

6FC5397-0DP10-2RA0

安全技术提示

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
带有警告三角，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。
小心
不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。
注意
表示如果不注意相应的提示，可能会出现不希望的结果或状态。

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

仅允许安装和驱动与本文件相关的附属设备或系统。设备或系统的调试和运行仅允许由**合格的专业人员**进行。本文件安全技术提示中的合格专业人员是指根据安全技术标准具有从事进行设备、系统和电路的运行，接地和标识资格的人员。

按规定使用

请注意下列说明：

 警告
设备仅允许用在目录和技术说明中规定的使用情况下，并且仅允许使用西门子股份有限公司推荐的或指定的其他制造商生产的设备和部件。设备的正常和安全运行必须依赖于恰当的运输，合适的存储、安放和安装以及小心的操作和维修。

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。标签中的其他符号可能是一些其他商标，这是出于保护所有者权利的 目地由第三方使用而特别标示的。

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 文献

SINUMERIK 文献分为 3 个类别：

- 一般文献
- 用户文献
- 制造商/维修文献

每月更新的各种可提供的语言版本的印刷品一览信息，请访问下列网址：

<http://www.siemens.com/motioncontrol>

请查阅菜单项“支持”(Support) → “技术文献”(Technical Documentation) → “印刷品概览”(Overview of Publications)。

DOConCD, DOConWEB的Internet版本,请访问下列网址:

<http://www.automation.siemens.com/doconweb>

培训课程以及 FAQ (常见问题解答) 的信息，请访问下列网址：

<http://www.siemens.com/motioncontrol> 和那里的菜单项 "支持"。

目标用户

该手册供开机调试人员使用。

设备/产品已经安装连接完毕，随时可以使用。

对于下面的操作步骤，例如：布线，接通，功能测试，开机调试手册必须包含所有必要的信息或者至少包含必要的提示。

使用

该调试手册 能够使相应的目标使用人群专业、安全地检测和运行产品/系统或者设备。

适用阶段：安装和开机调试阶段

标准功能范畴

在现有文献中描述了标准功能范畴。

机床制造商增添或者更改的功能，由机床制造商资料进行说明。

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

技术支持

技术疑难，请咨询下列热线：

	欧洲 / 非洲
电话	+49 180 5050 222
传真	+49 180 5050 223
网址	http://www.siemens.com/automation/support-request

	美洲
电话	+1 423 262 2522
传真	+1 423 262 2200
E-Mail	techsupport.sea@siemens.com

	亚洲 / 太平洋
电话	+86 1064 719 990
传真	+86 1064 747 474
E-Mail	techsupport.asia@siemens.com

说明

各个国家的技术咨询电话请访问下列网址：

<http://www.siemens.com/automation/service&support>

文献资料疑问

如果您对该文献有疑问（建议，修改），请发送传真或电子邮件到下列地址：

传真	+49 9131 98 63315
电子邮件	docu.motioncontrol@siemens.com

传真表格见本文献末页。

SINUMERIK 网址

<http://www.siemens.com/sinumerik>

欧盟一致性声明

对 EMC 方案的 EG 一致性声明参见

<http://support.automation.siemens.com>

产品/订货号为：15257461，或者在西门子股份公司 A&D MC 销售区的负责办事处获得资料。

SIEMENS

SINUMERIK 840Di sI/840D sI/840D

HMI 高级 (IM4)

开机调试手册

引言	1
配置 HMI 系统	2
机床数据参数化	3
编程 PLC 功能	4
诊断与服务	5
刀具管理	6
INI 文件表列	A
缩略语列表	B

适用于:

控制系统 SINUMERIK 840Di sI/840DiE sI
SINUMERIK 840D sI/840DE sI SINUMERIK
840D/840DE

.....

01/2008

6FC5397-0DP10-2RA0

安全技术提示

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
带有警告三角，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。
小心
不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。
注意
表示如果不注意相应的提示，可能会出现不希望的结果或状态。

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

仅允许安装和驱动与本文件相关的附属设备或系统。设备或系统的调试和运行仅允许由**合格的专业人员**进行。本文件安全技术提示中的合格专业人员是指根据安全技术标准具有从事进行设备、系统和电路的运行，接地和标识资格的人员。

按规定使用

请注意下列说明：

 警告
设备仅允许用在目录和技术说明中规定的使用情况下，并且仅允许使用西门子股份有限公司推荐的或指定的其他制造商生产的设备和部件。设备的正常和安全运行必须依赖于恰当的运输，合适的存储、安放和安装以及小心的操作和维修。

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。标签中的其他符号可能是一些其他商标，这是出于保护所有者权利的目地由第三方使用而特别标示的。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

目录

1	引言	7
1.1	系统交货状态	7
1.2	引导起动	8
1.2.1	在 HMI 上的设置	8
1.2.2	系统设置	10
1.2.3	PCU 的键盘特性	12
1.3	访问等级概念	13
1.4	许可证授权	15
1.5	数据维护极限	16
2	配置 HMI 系统	17
2.1	处理 INI 文件	17
2.1.1	结束 OEM IBN	20
2.1.2	激活屏幕保护程序	21
2.1.3	设计退出模式的查询对话框	22
2.1.4	设置报警记录存储器位置	22
2.1.5	设计 PLC 报警的确认符号	25
2.1.6	用于修改程序的存取权限	26
2.1.7	设计软键‘语言选择’	27
2.1.8	建立用户操作菜单	28
2.1.9	增加用户专用的服务显示	35
2.1.10	设置 OEM 应用的宽屏显示	39
2.1.11	建立特殊工艺的文本	41
2.1.12	创建带有工作列表的工件	43
2.1.13	刀具管理 (WZV)	45
2.1.14	无刀具管理情况下的刀具选择	46
2.1.15	硬盘处理 (m:n 配置)	48
2.1.16	激活 V.24 接口	50
2.2	设计通道菜单	51
2.2.1	通道菜单应用	51
2.2.2	通道菜单结构	53
2.2.3	设计用于直接通道选择的通道菜单 (1:1 配置)	54
2.2.4	设计一个双通道显示	57
2.2.5	在 m:n 配置时操作单元管理的通道菜单 (powerline)	59
2.2.6	在 m:n 配置时操作单元管理的通道菜单 (solution line)	60
2.3	连接网络驱动器	66
2.3.1	设置访问外部驱动器或者计算机	66
2.3.2	使用显示机床数据来连接驱动器	67
2.3.3	连接逻辑驱动器	69
2.4	优化模拟	73
2.4.1	显示模拟文件	73
2.4.2	模拟的数据匹配	75
2.4.3	加速模拟引导启动	79

2.4.4	扩展几何刀具数据	80
2.4.5	优化所需存储容量	83
2.5	建立用户报警	85
2.5.1	用户专用报警的结构	85
2.5.2	生成用户专用的报警文本	88
3	机床数据参数化	91
3.1	通过 HMI 使用机床数据	91
3.1.1	调整用于机床数据的显示过滤器	93
3.1.2	创建用户视图	95
3.2	通过机床数据对设置控制系统的参数	98
3.2.1	精确零点偏移和基本偏移	98
3.2.2	用于设定实际值, 对刀, PRESET (预设) 的机床数据	99
3.2.3	使有效的刀具补偿立即生效	100
3.2.4	确定 STAT 和 TU 的值描述	101
3.2.5	DRAM 内循环的存储器位置	102
3.2.6	重新定义用于 NC 数据的访问等级	103
3.2.7	修改软键 ‘基本零点偏移’ 上的存取权限	104
3.2.8	隐藏轴	105
3.2.9	主轴载荷的显示	107
3.2.10	主轴显示的主轴符号反向	110
3.3	创建 PLC 机床数据明码文本	111
4	编程 PLC 功能	113
4.1	激活 PLC 和 NCK 之间的数据传输	113
4.2	将 HMI 的当前任务编号传输至 PLC	116
4.3	通过 PLC 选择通道/主轴	118
4.4	设计标题中的信息显示	119
4.5	利用多个通道启动程序段搜索	122
4.6	通道交叉状态显示	124
4.7	用户状态显示 (OEM)	130
5	诊断与服务	135
5.1	在 PG/PC 上安装 HMI 软件	135
5.1.1	NCU 连接向导	136
5.1.2	在自身桌面上启动 HMI 高级	137
5.1.3	设计桌面切换应用程序	139
5.1.4	操作桌面切换应用程序	141
5.2	保存屏幕截图	142
5.3	显示版本	143
5.4	设置并升级系统	144
5.4.1	批量开机调试	144
5.4.2	PLC 升级	146
5.4.3	示例: 如下进行 PLC 升级	149
5.5	服务显示	151
5.5.1	服务轴	153
5.5.2	显示系统资源	153
5.5.3	输出配置数据	154

5.5.4	通信错误记录.....	154
5.6	运行记录仪.....	155
5.6.1	设置运行记录仪.....	155
5.6.2	记录文件的结构.....	158
5.6.3	这样保存记录文件。.....	161
5.7	HMI 分析仪.....	162
5.7.1	使用 HMI 分析仪.....	162
5.7.2	操作 HMI 分析仪.....	163
5.7.3	分析 INI 文件.....	165
5.7.4	应用实例.....	167
6	刀具管理.....	171
6.1	在 HMI 上编程刀具管理.....	171
6.1.1	创建一个真正的刀库.....	171
6.1.2	这样编程一个中间存储器.....	173
6.1.3	装载位置和装载刀位.....	176
6.1.4	选择刀位类型.....	179
6.1.5	示例： 参数化刀位类型.....	180
6.1.6	示例： 这样分配刀位类型一个级别.....	183
6.1.7	刀库配置.....	184
6.1.8	这样设置和装载一个配置.....	187
6.2	配置文件中的设置.....	189
6.2.1	配置文件 paramtm.ini.....	189
6.2.2	文件 paramtm.ini 的结构.....	191
6.2.3	设计软键显示.....	242
6.2.4	参数化刀具位图显示.....	249
6.2.5	示例： 用于机床的用户定义的设置.....	251
6.3	用户定义名称的国家语言相关性.....	253
6.3.1	刀库的国家语言相关的名称.....	253
6.3.2	中间存储器的国家语言相关的名称.....	255
6.3.3	装载刀位的国家语言相关的名称.....	256
6.3.4	刀位类型的国家语言相关的名称.....	258
6.4	刀具的任务准备.....	260
6.4.1	一般适用于所有过滤器的设置.....	260
6.4.2	适用于过滤器的设置.....	263
6.5	磨削刀具和刀具专用磨削数据.....	269
6.6	考虑设置英制/公制.....	271
6.7	调试代码载体.....	274
6.7.1	代码载体数据描述.....	274
6.7.2	说明文件的结构.....	278
6.7.3	数据的转换规定.....	284
6.7.4	示例： 说明文件.....	287
6.7.5	示例： 数据 - 字符串.....	289
A	INI 文件表列.....	291
A.1	ACTLOG.INI.....	292
A.2	AEDITOR.INI.....	293
A.3	DINO.INI.....	295
A.4	DG.INI.....	297

A.5	DGOVW.INI.....	297
A.6	DH.INI.....	299
A.7	DPDH.INI.....	299
A.8	DPSIM.INI.....	299
A.9	HEADER.INI.....	301
A.10	HMIDESK.INI.....	305
A.11	IB.INI.....	306
A.12	IF.INI.....	308
A.13	KEYS.INI.....	309
A.14	LOGDRIVE.INI.....	309
A.15	MASCHINE.INI.....	312
A.16	MBDDE.INI.....	315
A.17	MMC.INI.....	321
A.18	NETNAMES.INI.....	327
A.19	OEMFRAME.INI.....	331
A.20	PARAM.INI.....	331
A.21	PARAMTM.INI.....	332
A.22	REGIE.INI.....	333
A.23	SEEDITOR.INI.....	335
A.24	SIMTOGEO.INI.....	343
A.25	TASKCONF.INI.....	346
B	缩略语列表.....	349
B.1	缩略语.....	349
	索引.....	353

引言

1.1 系统交货状态

概览

该手册对软件 HMI 高级的开机调试进行了说明。若要对 SINUMERIK 控制器进行开机调试，还可能需要使用下列手册：

- 操作组件与联网
- PCU 基础软件开机调试手册
- 诊断手册
- 清单手册
- 基本功能手册

有关 NCK、HMI、PLC 或驱动功能的其它信息，请查阅功能手册。

软件

软件 HMI 高级在交货时可能根据预订已装载在 PCU 50.3 上。如果软件 HMI-高级未安装在 PCU 50.3 上（例如 PCU 50.3 不带系统软件），则 HMI-软件可以从 PCU 50.3 提供的 CD 上通过服务中心安装。

若要将软件 HMI 高级安装到 PCU 50.3 上，需要：

- PC 或带 CD 驱动器的 PG
- 网络连接
- USB 存储器

HMI 高级可以在 Windows XP SP2 操作系统下运行。

注意

在 PC/PG 上进行安装：

仅当非管理员用户拥有 HMI 高级的 mmc2 目录的写入权限时，该用户才可以运行 HMI 高级。HMI 高级的安装不提供该权限。

1.2 引导启动

1.2.1 在 HMI 上的设置

语言

使用该功能可以选择第一和第二语言，并通过软键“切换语言”在两者之间进行切换，可以选择的标准语言有：

- 德语
- 英语
- 法语
- 意大利语
- 西班牙语
- 简体中文

NCU 连接

通过该功能可以设置 NCU 的 IP 地址。从文件 MMC.INI 中可以显示所记录的 IP 地址。

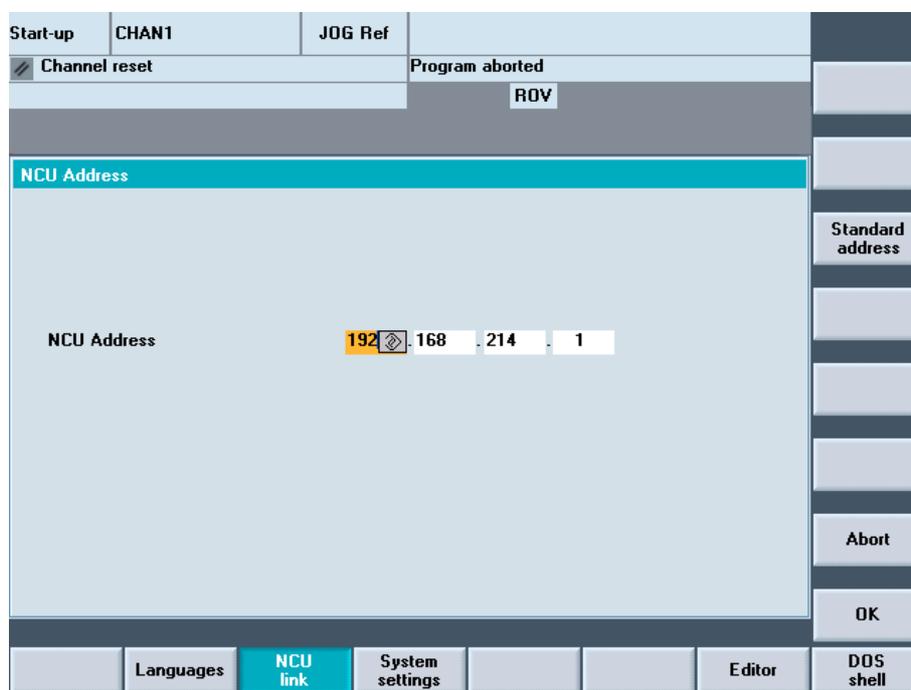


图 1-1 NCU 的 IP 地址

NCU 从缺省地址为 192.168.214.1 的车间输出。在 1:1 情况中没有其它联网时，可以保留该地址。使用软键“缺省地址”可以在 NCU 的地址栏中接收按车间标准设置的 IP 地址 192.168.214.1。

如果控制系统相连，如在一个公司网内，那么 IP 地址就会发生变化。要使更改有效，必须重新启动 HMI。带有新 IP 地址的段按照 user/mmc.ini 写入。

引导启动时的故障

引导启动时可能出现下列故障：

报警 120202：等待连接至 NC/PLC

说明：当 HMI 程序第一次启动且 NCK/PLC 启动尚未结束时或者与这些组件的通信发生故障时，出现报警。

在连接该报警时，所有与 NCK/PLC 连接的显示值无效。在控制系统启动时出现此类故障是正常的（例如复位后）。

消除方法：一旦故障状态结束，报警自动消失。如果持续出现报警，则可能有不同的故障原因（例如断线，未启动 NCK/PLC，地址/波特率配置错误 ...）。

反应 --

系统设置

参见下一章节系统设置。

打印机选择

只有当在 Windows 下至少安装了一台打印机时，才能使用该软键。软键可以打印开机调试操作区中的画面或数据。用<选择>键选择，应使用哪个已安装的打印机进行打印。

预设置：作为位图文件输出

编辑器

您可切换到 ASCII 编辑器用于在 Windows 平台上编辑文件。
通过垂直的软键您可以选择现有的驱动器。

参见

生成用户专用的报警文本 (页 88)

1.2.2 系统设置

概览

在“系统设置”下可以对询问窗口、文件树显示和加工、程序、通讯操作区屏幕上的显示进行设置。

文件显示

可以为通讯、加工和编程操作区设置文件树显示。可以选择下列栏：

- 文件类型（扩展名）
- 已装载
- 长度
- 存取保护
- 数据
- 时间
- 使能
- 显示级（目录树中分支，最大为 7）
- 名称长度（最大 25 个字符）

信息分类

会打开一个按**某一**列进行分类顺序设置的对话框。相关的设置适用于所选择列的加工、程序、通讯操作区中的相应窗口显示。

您可以选择分类标准：

- 无分类：正常情况下按“名称”升序排列。
- 各列名称中的一个：分类顺序降序或升序。用 **OK** 确认操作。

设置的分类顺序作为箭头符号显示在作为分类标准所选列的列名称旁。对于 HMI 高级版，可以另外用选件鼠标进行下列有关列分类的操作：

用箭头符号旋转方向点击列名称并进行相应的分类。点击另一个列并使其作为分类标准。如有必要，可以根据（1）所述继续点击更改方向。选择操作区中另一个分类标准将更改所有操作区的分类标准（加工、程序、通讯）。

说明

设置分类顺序时，如果操作区显示画面中没有分类标准，则按照名称以升序方式分类。对于操作区中存在对话框中的设置标准，适用于对话框中确定的分类。

询问

此处您可确定是否要在操作（例如删除）后显示一个询问窗口。征求确认：

- 删除数据/程序，
- 删除目录，
- 覆盖文件。

打开窗口“画面中操作按键的描述...”：此处可以设置是否要将 HMI 画面中的操作按键作为符号或者作为文本表示。示例：美国布局操作面板

使用工件模板

在这里确定，设置新工件时是否应将存在的模板传输到新工件（目录）中：

- 工作列表
- 零件程序
- 初始化程序

轨迹

在通信故障情况下，可以按照指令通过售后服务或者热线记录一个通信过程的轨迹记录。轨迹记录的评估只可由西门子进行。

HMI 退出模式

在开机调试阶段经常需要退出 HMI 高级以及操作系统，并接着重新启动 PCU。为避免关闭/接通 PCU 或机床，请选择“退出时自动重启控制系统”。

参见

设计退出模式的查询对话框 (页 22)

1.2.3 PCU 的键盘特性

CAPSLOCK

CAPSLOCK 功能为，使用 **外部 SINUMERIK** 键盘输入文本时的将小写字母转换为大写字母。在小写输入情况下通过功能 **SHIFT** 命令将其插入。

该功能仅在 **HMI 高级** 运行时生效，并不是 **Windows XP** 运行后就有效。

前提条件

CAPSLOCK 功能被激活时，**<SHIFT>** 键对字母键不再起作用。

CTRL/ALT 键顺序在 **Windows** 标准应用程序下仅在小写模式下起作用。对于在 **OP** 中集成的键盘无法切换到小写模式。

<CTRL> + <SHIFT> 切换顺序仅在使用外部 **PS2** 键盘和外部 **USB-PC** 键盘时起作用，而在使用面板键盘时不起作用。

说明

如果在一个外部键盘上设置有 **CAPSLOCK**，则 **OP** 键盘提供小写字母。如果在该状态下插补外部键盘，则不能够再取消 **CAPSLOCK** 模式。

<CAPSLOCK> 与 **<NUMLOCK>** 键可以通过相应的设置在参数文件 **E:\Windows\System.ini** 中加以过滤。

前提条件是，**CAPSLOCK** 和 **NUMLOCK** 未被过滤。

激活

该功能通过显示机床数据 **9009: \$MM_KEYBOARD_STATE** 激活

- 0: CAPSLOCK 关
- 2: CAPSLOCK 打开

仅在启动时评估显示机床数据。修改的机床数据在引导启动后才会生效。

显示机床数据 **9009** 设置为“**CAPSLOCK** 打开”，而您又想要输入小写字母：

1. 按下按键 **<CTRL> + <SHIFT>**，进而再次转换成为小写输入。
2. 若要将小写输入再次转换为大写输入，再次按下 **<CTRL> + <SHIFT>**。

1.3 访问等级概念

概览

访问等级概念控制功能和数据范围的访问。有 0 至 7 个访问等级，0 表示最高等级，7 表示最低等级。

访问等级 0 至 3 的联锁通过识别字进行，4 至 7 通过钥匙开关位置进行联锁。

访问等级	禁用通过	区
0	---	系统
1	口令: SUNRISE	机床制造商
2	口令: EVENING	维修
3	口令: CUSTOMER	用户
4	钥匙开关位置 3	编程员, 调试员
5	钥匙开关位置 2	合格的操作员
6	钥匙开关位置 1	受过培训的操作员
7	钥匙开关位置 0	学过相关内容的操作员

密码一直保持如此设置，直至用软键“删除密码”复位为止。访问等级 0 口令允许访问所有范围。

口令可在激活后更改。例如密码无法再被识别，则必须重新进行初始化（NCK 清零）。此时所有密码将被复位成标准状态。上电不会复位口令。

钥匙开关

访问等级 4-7 要求在机床控制面板上有一个相应的电键开关位置。因此有三个不同颜色的电键。每个电键仅可许可特定的区域。所属的接口信号位于 DB10.DBB56 中。

电键开关位置含义：

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

权限

操作员只能存取符合该指定访问等级和低于该访问等级的信息。用不同的访问等级依据一定的标准设立机床数据。

如要显示机床数据，则至少需要访问等级 4（电键开关位置 3）。进行开机调试时一般使用制造商密码“SUNRISE”。

修改口令

操作步骤：

1. 按下软键“口令”。
2. 按下软键“设置密码”。
3. 显示输入窗口“请输入密码”
4. 输入三个可能密码中的一个，并用输入键或符号对输入进行确认。确认一个允许的口令为已设置，并显示当前有效的存取等级。不接收无效的口令。在可以设置一个较低存取等级的口令作为当前激活前，必须删除该口令。
5. 按下软键“删除密码”。
6. 通过按下软键“删除口令”删除最近有效的口令并确认。设置当前有效的存取等级：钥匙开关位置为 0。
7. 如果已设置了密码，则可按以下方式对其进行更改。为此按下软键“更改密码”，会出现输入窗口“更改密码”。

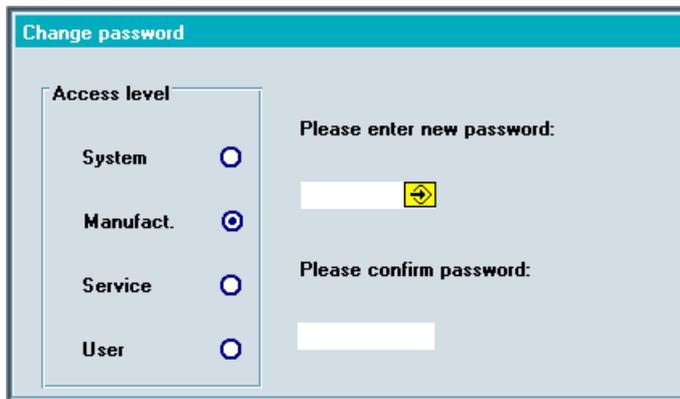


图 1-2 更改密码

8. 必须输入在两个输入栏中更改的口令，随后用软键“OK”确认。只有当两个口令一致时，才接收有效更改的口令。

1.4 许可证授权

许可证选项

HMI 高级的选项管理器在输入一个选项的许可证密钥时为您提供支持。

输入许可证密钥后软键“接收”被激活。点击软键“接收”后，许可证密钥将被写入 NCK 中并由其进行有效性检测。

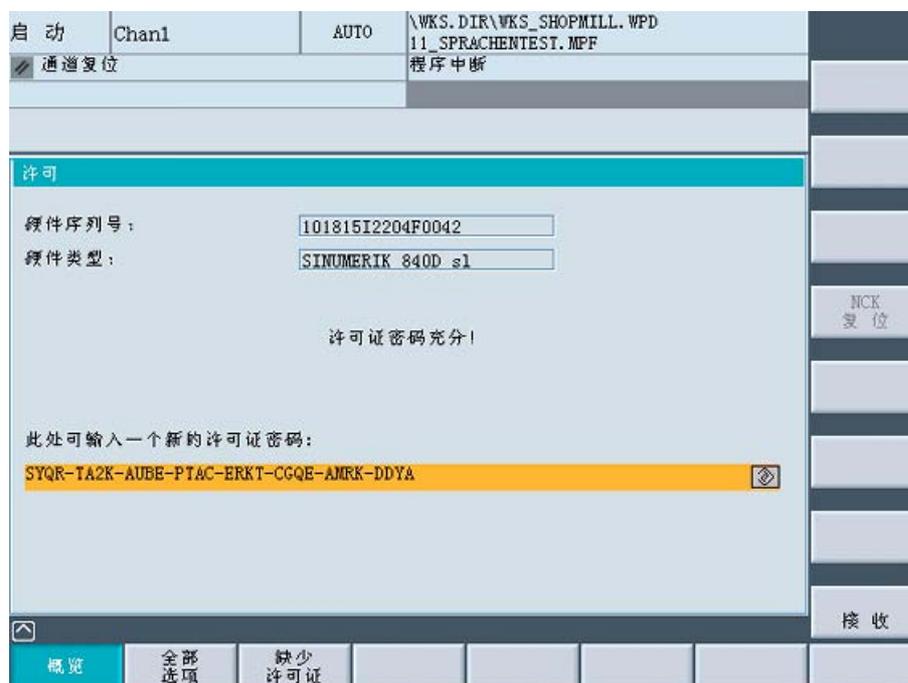


图 1-3 许可证概览

如果输入了无效的许可证密钥，会被 NCK 拒绝并通过显示信息进行说明。如果三次写入了错误的许可证密钥，就必须执行 NCK 上电复位。

说明

NCK 中开机调试时与操作步骤是相关的，是否必须执行 NCK 上电复位：

- 如果选项位已设并且接着功能起动，开机调试后就必须执行复位。
- 如果首先起动功能，触发复位并且接着在选项管理器中的选项位被设置，就必须重新执行复位，以使更改有效。

写入许可证密钥后不必执行复位，在对话框中会显示信息“许可证密钥已设！”。

参考文献： SINUMERIK 840Di sl 手册，章节“自动化许可证管理器”

全部选件

此处列出所有系统可选的选项。此外还要记录，是否已存在有效且足够的许可证密钥以及已有几个许可证存在。

缺少的许可证

这里显示已设选项，和在当前许可证密钥中仍未授权的选项。HMI 仅在引导启动中对选项进行检查，因此在每种情况下都必须进行 HMI 的重新启动。

1.5 数据维护极限

概览

数据存放在下列目录中：

- 工件
- 零件程序
- 子程序
- 用户循环
- 标准循环
- 制造商循环

在该数据维护目录中允许存放最大总和为 100 000 个文件，此时，目录下文件数目（对于工件按照工件目录 *.WPD）最大允许为 1000。其它目录在总和 100 000 中不考虑，但此时，一个目录下也最多只能有 1000 个文件，例如，在目录存档中最多有 1000 个存档。适用于网络驱动器：每个目录最多存放 1000 个文件。

实际情况取决于文件大小和可用的存储器空间。在目录显示时过多的文件会使画面建立非常缓慢。

可根据语言建立 5000 个自身的报警或提示信息。

配置 HMI 系统

2.1 处理 INI 文件

标记

[xxx]	;段落标记
命名符=值	;赋值
命名符=<empty>	;删除值
,	;存在多个指令时的分隔符号（逗号）
;	;引入注释： 一般情况下将注释命名为可以进行赋值的备用值。
REM	;引入注释

目录结构

系统目录结构是这样组织的，用户更改通过软件升级得以保留。

- 在为此而平行的目录中进行原始的用户更改。
- 在标准 HMI 软件和用户专用的补充信息之间有一个明显的分隔符。
- 目录“mmc2”和“hmi_adv”（也包括子目录）应该为只读。它包含交付的原始软件。仅在升级时覆盖这些目录。

在下图中规定了优先级顺序。这表明：目录中的条目越往右，优先级越高，右侧的条目优先级高于目录中相应的左侧条目优先级。

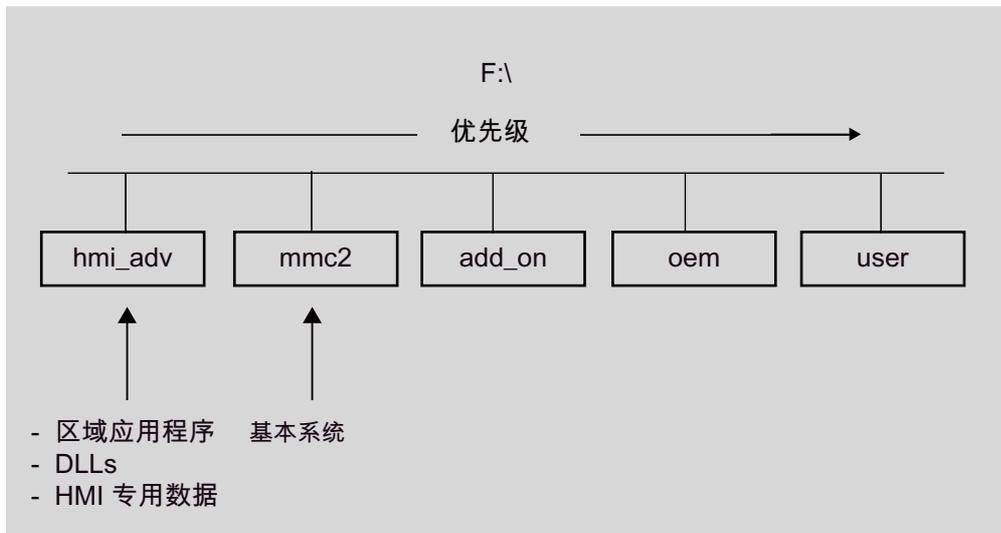


图 2-1 优先级

目录内容

原则上，mmc2 的平行目录中只存储与 mmc2 中原始数据不同的 ini 文件 **差别记录项**。

注意
不得改变目录 “mmc2”和“hmi_adv”中的所有 INI 文件。

目录包含下列内容：

- **hmi_adv:**
目录包含区域应用程序、DLL 和 HMI 特定的数据。
- **mmc2:**
HMI 软件的系统目录。
- **add_on:**
所附加西门子产品的目录（例如 TPM、MDA、DNC ...）。这里可以存放带有子目录的相同目录结构（例如：子目录\语言），与 mmc2 中一样。Add-On 产品的记录条目在这里的存储方式与在 REGIE.INI 和 re_*.ini 中一样。
- **oem:**
机床制造商和 OEM 用户的目录，在该目录中存储有他们自己的、目录结构相同的、带有子目录的 OEM 应用程序（例如：子目录\语言），与 mmc2 中一样。OEM 产品的记录条目在这里的存储方式与在 REGIE.INI 和 re_*.ini 中一样。
- **user:**
用户目录，在该目录中存放针对提供的 ini 文件的客户差异性。此处也可以保存操作界面外观的**更改信息**，可以通过在 HMI 操作界面上的设置自行进行（例如语言设置、文件选择、文件管理器视图，...）。还可以存放一些其它设置，这些设置与 OEM 或者 Add-On 产品无关，保存在目录\user 中，例如报警服务器设置。
“共同文件”中的一些修改，比如规定某个访问权限或者不显示某个软键，也必须存放在目录\user 中。

举例：

在“oem”目录中不生成完整的 REGIE.INI 文件，而仅仅储存其中经过修改的部分。

REGIE.INI 表示如下：

```
[TaskConfiguration]
Task7:=oem1appl, .....
```

错误情况下的诊断

使用 HMI 分析器对 INI 文件的设置进行分析并显示结果。

用户特定条目的操作步骤

如同在“mmc2”中一样，在目录 F:\USER 中设立相同名称的空 INI 文件，（如果需要，在 USER 中也可以设立子目录，例如 \语言）。

在 USER 目录下的 INI 文件中仅复制要改变的段和写入新的或已改变的条目。

说明

不要复制整个“MMC2”的 INI 文件！仅输入差异项即可。

示例：

文件\USER\MBDDE.INI 中用于 PLC 错误信息和报警行中翻页的记录项：

```
[Textfiles]
UserPLC = F:\DH\MB.DIR\MYPLC_
```

```
[Alarme]
; 报警行中的翻页
RotationCycle = 1000
```

相同的步骤也适用于目录 F:\ADD_ON 和 F:\OEM。 \ADD_ON 预留用于西门子产品。在目录 \OEM 中安装了所有的 OEM 应用程序。

注意

例如 REGIE.INI 中附属的记录条目应在该目录下进行，而不是在目录 \MMC2\ 下进行！
--

关闭设置

在查找顺序中如果在较低优先级的目录中存在有较高优先级目录中要废除的具体记录项，则用 <empty> 记录项替代该值。

举例：

目录 mmc2，文件 xxx.INI:

[<Sektion>]

<命名符> = <值>

目录 user，文件 xxx.INI:

[<Sektion>]

<命名符> = <empty>

最大文件长度

Windows 系统中 INI 文件的最大文件长度为 60 kB。如果超过这个限值，则忽略末尾的条目。

消除方法：例如，由于在 MMC2\PARAMT.INI 中也包含详细的注释，因此将 USER\PARAMT.INI 中的该注释删除。

2.1.1 结束 OEM IBN

功能

使用该功能把初始化文件从 USER 目录接收到 OEM 目录中。为了让用户可以使用空的 USER 目录、并且删除其中存在的初始化文件(*.ini)，而文件中包含的设置不会丢失，因为初始化文件已从 USER 目录被接收到 OEM 目录中。如果 OEM 目录还不存在，会自动创建，如在 USER 目录中有初始化文件。

汇合所有 INI 文件：

1. 如要接收 **所有** 初始化文件，按下软键“结束 IBN”。会获得下列显示信息：“汇合所有 USER 目录中的初始化文件(*.ini) 连同 OEM 目录中相应的文件。”
2. 按下软键“保存”，以开始传输。
OEM 目录中的已有记录被在 USER 目录中的相应记录覆盖。
3. 创建新文件。

保留在 USER 目录中未出现，但在 OEM 目录中存在的文件。传输中会在状态栏显示当时文件的名称。成功接收所有数据后，该文件会从 USER 目录中删除。

接收所选 INI 文件：

若只要接收确定的初始化文件，按下垂直软键“数据选择”。它包含 USER 目录下所有文件的列表。

2.1.2 激活屏幕保护程序

功能

屏幕保护程序的任务：

- HMI 屏幕保护程序：保护屏幕和背景灯
- Windows 屏幕保护程序：保护屏幕

在文件 MMC.INI 的段落[GLOBAL]中设定屏幕保护程序开始之前的延迟时间[分钟]：

[GLOBAL]

; latency for the screensaver

MMCScreenOffTimeInMinutes = 60

注意

不允许同时使用 HMI 屏幕保护程序和 Windows 屏幕保护程序。

显示服务器名称

为了显示服务器名称，需要使下列记录项生效：

[GLOBAL]

; to make the hidden servers (NCDDE, MBDDE, DHSERVER, ARSERVER) visible,
uncomment this entry

ServerVisible = 1

2.1.3 设计退出模式的查询对话框

功能（仅用于 Windows XP）

该功能可以在开机调试阶段实现 HMI 高级包括 Windows XP 的关机，然后重新启动 PCU。这样可以避免 PCU 或机床的断开/接通。

记录项既可以直接在文件 REGIE.INI 中设置，也可以通过操作界面在开机调试 → HMI → 设置 → HMI 退出模式下进行设置。如果已对菜单“退出”进行了相应设计，则根据访问级别的不同会出现“关机...”或“重新启动...”的询问。

文件 REGIE.INI 的设置：

预设置： 无询问对话框

[Miscellaneous]

EnableRebootDialog = True

如果访问级低于制造商的访问级（访问级 0-2）或已在 REGIE.INI 中将上述条目置为“假”，则可以通过以下操作进行询问：

OK	退出 HMI 并关掉 Windows XP。
中断	不退出

如果已有相应的访问级（访问级 3-7），则有下列可能：

重新启动	退出 HMI 和 Windows XP，紧接着重新启动。
结束	退出 HMI 并关掉 Windows XP。
中断	不退出

2.1.4 设置报警记录存储器位置

功能

为了在硬盘上保存报警配置，可以在不同的方式之间进行选择。必须在应用程序和硬盘负荷方面进行选择。所需要的条目在文件 MBDDE.INI 中实现。

总是在相同的硬盘区写入报警文件将导致负荷产生。通过下列控制方法可以选择不同的方式。

提供多个策略用于硬盘维护。硬盘上可以同时保留多个记录文件。在每次启动 HMI 时，在多个文件策略情况下切换到下一个文件。这不仅可以减小数据信息区中的硬盘硬件负荷，也可以

减小目录信息区中的硬盘硬件负荷。另外，在启动时策略识别数据区中的物理硬盘故障并通过新指令从存储器位置转出。

可以通过文件 `mbdde.ini` 中的下列记录项选择多文件策略：

[PROTOCOL]

DiskCare

总共可以有列值：

DiskCare=-1	MBDDE 服务器将报警记录存放于存储器中。如果在“诊断”操作区域中显示有报警记录，或者按下了<取消报警>按键，则该记录被存储在硬盘上。
DiskCare=0	立即写入记录文件。
DiskCare=n	如果 n 秒未出现变化，则报警文件的变化被写入记录文件之中。 另外适用于：如果在“诊断”操作区域中显示有报警记录，或者按下了<取消报警>按键，则该记录被存储在硬盘上。
DiskCare=-n	n>1 特指平行引入的报警文件数目。

在启动时分析评估文件“`mbdde.ini`”。将不使用的报警文件如同当前的记录文件一样存放在 `mmc2` 目录中。隐含文件的名称（属性“隐含”）由一个八位的十六进制数组成，带有扩展名“.alr”。启动时确定文件区中有写错误的文件同样由一个 8 位的十六进制数组成，然而带有扩展名“.al_”。

记录文件大小

在文件 `MBDDE.INI` 中可以确定记录文件的大小（环链缓冲器）。

[Alarme]

Records = wert ; 记录文件的大小

预设置： 150

最小值： 18

最大值： 32000

使用预设置 150 可以显示 75 个报警。一个报警总是需要 2 条记录。

根据特征选择报警

通过 `MBDDE.INI` 中的过滤器记录可以控制记录文件中的信息满溢情况。

[PROTOCOL]

过滤器=表达式

表达式表达选择特征并如下表示：

句法

[识别符号][关系][特征][运算符]

- 识别符号：

Nr	报警号
Prio	优先级
Mode	信息行/报警行或者对话框
Typ	报警类型（上电，取消，...）
From	报警发送者
Quitvar	确认变量

● 关系:

"="	相等
"<"	小于
">"	大于
"!"	非

- 特征: 数字或字符串
- 运算符

","	逗号表示逻辑或, 仅在一个过滤器中
" "	空格符/空格表示逻辑与, 并且 在单个过滤器之间。
" "	竖号表示逻辑或, 在 单个过滤器之间。

举例:

Filter=Typ<3

仅记录 POWERON（上电）和 RESET（复位）报警。

Filter=From:NCU_1

仅记录 NCU_1 的报警

Filter=From:NCU_1 Typ:1,3

仅记录 NCU_1 的 POWERON（上电）和 CANCEL（取消）报警

参见

MBDDE.INI (页 315)

2.1.5 设计 PLC 报警的确认符号

功能

您可以设立自己的确认符号用于确认 PLC 报警。可以选择在 PLC 生成报警时（通过 FC 10）要在诊断报警画面中显示哪些位图。选择的像素画面（尺寸 20 x 20 像素）应该显示用户必须用哪些按键确认由 DB2 生成的报警。

选择的按键必须与由 FC 10 规定的确认参数 QUIT 中的信号匹配，例如：

```
Call FC 10
ToUserIF:=TRUE
Quit:=DB21.DBX7.7 //Channel 1 RESET
```

选择一个段 [ALARM_PICTURE] 中要显示的画面值：

```
[ALARM_PICTURE]
;0 = PLC 按钮画面
;1 = Cancel 按钮画面
;2 = Reset 按钮画面
;3 = HMI 按钮画面
;4 = 用户定义按钮画面
按钮画面=1
```

如果选择 4（用户自定义按键），则必须通过 **ButtonNameUser**=“名称.bmp”来给出相应的画面。此时可自由选择 **名称**。

建议在 **USER** 或者 **OEM** 目录的 **DG.INI** 中设立条目。

画面必须位于一个、多个或者所有目录中：

```
;F:\User\Icons\640
;F:\User\Icons\800
;F:\User\Icons\1024
;F:\Oem\Icons\640.
;F:\Oem\Icons\800.
;F:\Oem\Icons\1024.
```

自动与当前的屏幕分辨率进行匹配。画面型号必须位于带有相应于当前分辨率的画面目录中。

2.1.6 用于修改程序的存取权限

功能

在启动阶段数据维护服务器会进行检测，看文件 **DH.INI** 中是否有用于修改的标准存取权限的记录条目。根据这些条目，数据维护服务器更新其已改变的访问权限的数据维护方案内部图。条目中所描述的存取权限现在作为标准存取权限应用于所选择的数据。忽略与现有的方案不匹配的条目。

前提条件

在安装 **HMI** 数据库时，数据总是通过标准数据方案的访问权限设立。节点的存取权限，例如 **wks.dir** 无法修改。

格式和数据类型

所有的记录条目位于段 **[ACCESSMASKS]** 中。记录条目有如下形式：

[ACCESSMASKS]

<code>\wks.dir*.wpl*.mpf =</code>	<code>75775</code>
<code>\mpf.dir*.mpf =</code>	<code>75775</code>
<code>\cus.dir*.spf =</code>	<code>33773</code>

每个数据类型和存放位置可以统一布置为一个标准访问屏幕窗口。一个条目由可以产生该数据类型的数据所在的路径和新的标准访问屏幕窗口组成。

路由数据类型命名符组成。在上述例子中，“**\wks.dir*.wpl*.mpf**”是一种数据类型，表示使用工件目录“**wks.dir**”工件“***.wpl**”和零件程序“***.mpf**”。

在所有工件中定义新的标准访问屏幕窗口 **75775**，通过该屏幕窗口设立新的零件程序。

访问屏幕窗口 **75775** 如下表示：

<code>7</code>	各个读取权限	许可 0 至 7
<code>5</code>	自防护等级 5 起的写入权限	允许 0 或 7
<code>7</code>	已设置执行权限	允许 0 或 7
<code>7</code>	各个显示权限	允许 0 或 7
<code>5</code>	自防护等级 5 起的删除权限	许可 0 至 7

参见

DH.INI (页 299)

2.1.7 设计软键 ‘语言选择’

功能

为了能够在控制系统选择两种以上的语言，可以通过文件“IB.INI”在“开机调试”基本画面中设计软键“语言选择”，用该软键可以通过菜单在各语言之间进行切换。

文件“IB.INI”中的记录条目“SkSelectLngInsteadSkChangeLng”可以在“[LANGUAGE]”段中进行控制，决定是否在开机调试基本画面中用软键“语言选择”来替代软键“切换语言”。如果该条目“SkSelectLngInsteadSkChangeLng”缺失，则仅存在语言切换。

“SkSelectLngInsteadSkChangeLng”的值有下列含义：

- False:** 软键“切换语言”可以在前景语言和背景语言之间切换，和原来一样（=预设置）。
- True:** 软键“语言选择”用于选择一个已安装的语言。
为此，可根据相应的选择在语言切换位置上在超过两种语言之间在线进行切换。

在设置“TRUE（真）”时（语言选择），要在文件 MMC.INI 中的“[LANGUAGE]”段中的条目添加以下值：

LanguageList= ... , GR

FontList= ... , Europe

FontListKO= ... , Europe2

LList= ... , 德语

说明

在进行语言选择时，不提供第一种语言用于选择。如果在第一种语言再一次列入语言列表（并因此进入相应的字体设置）之中，则语言选择菜单中也会提供第一种语言。

参数化

确定软键名称。

HSx x 1 - 8，水平软键 1 至 8

VSy y 1 - 8，垂直软键 1 至 8

不必占用所有软键。

文献：“补充操作界面”开机调试手册，
章节“设计 OP 热键和 PLC 键”。

2.1.8 建立用户操作菜单

功能

创建 **框架应用程序** 可以使用下列 **附加的方式**：

- 操作树的自由管理：设立新的操作层面、以前层面和新层面的新软键布置。
- 在操作层面上编入自身的应用程序（OEM 应用程序）。
- 从任意一个操作层面调用标准应用程序，包括已定义的子菜单。

利用功能“框架应用程序”，可以通过 **设计** 在新的软键下合并和选择现有的 HMI 高级操作区。为此，操作区/基本菜单软键对于自身的操作区来说是自由的（OEM 应用程序）。

功能“框架应用程序”可以多次使用，以从一个操作层面定义另一个下一层的操作层面。这使得操作树管理可以自由进行。

一个框架应用程序可以定义最多 **16** 个水平软键和 **8** 个垂直软键。从一个框架应用程序已选择的操作区返回，再次回到框架应用程序的上一层菜单。

由一个框架应用程序可以选择标准应用程序：

- 机床
- 参数
- 通讯
- 程序
- 模拟 → HMI 设置
- 诊断
- 开机调试
- OEM 应用程序

由框架应用程序可以有目的地在可选择的操作区中选择某个功能。（例如程序 → 模拟，下表中给定了所有的标准方式）。一个框架应用程序可以提供一个自身的基本画面作为位图文件或者显示一个标准画面。

配置原则

新的操作层面通过框架应用程序进行，框架应用程序嵌入在现有的操作层面中。可以配置框架应用程序的水平软键条和垂直软键条的布置。

配置文件

在下列文件中进行配置：

REGIE.INI 调用框架应用程序，标准应用程序
在 **REGIE.INI** 中将一个编译器任务 **mntmmc** 设定为任务，其参数
CmdLine:= "SektionName1"由段给定，在文件 **TASKCONF.INI** 的段中有其它数据用于插入的框架应用程序。

TASKCONF.INI	描述软键、以及待软键调用的操作区、背景画面，还可能包括软键标签文本。 在属性 CmdLine 规定的段中进行说明。
RE_xx.INI	用于语言的框架应用程序的软键标签，使用缩写 XX 。

举例

文件 **REGIE.INI** 中的下列记录条目说明了流程：

用于 **用户专用操作树** 的框架应用程序。

；示例条目

Task7 = name := mntmmc, cmdline := "SectionName1", Timeout := 60000

任务号（这里为 7）分配相应的软键，

；该软键调用框架应用程序。

Task0: 水平软键 1

...

Task7: 水平软键 8

mntmmc 是以标准形式存在的任务名称，用来在 **TASKCONF.INI** 中编译框架应用程序的软键并将其激活。

cmdline 的值用来为文件 **TASKCONF.INI** 中的段命名，在该段中详细描述框架应用程序的软键布置。

当在 **TASKCONF.INI** 中定义了多个段以及框架应用程序参数情况下，可以有多个条目 **任务<编号>= 名称 := mntmmc, ...**。这也可以由操作树的多级管理实现。

[SectionName1]

；在其上可自由选择的段名称必须归入到 **REGIE.INI** 中

；。

；输入举例：

；在第一次启动框架应用程序时应当将应用程序任务

；激活。

；通过启动索引的记录项来确定软键索引，

；在程序启动时激活所需的任务。

；这里将通讯的开始索引为 **5** 作为示例

StartIndex = 5

；框架应用程序基本画面

Picture= Rahmen1.bmp

；使用任务程序（任务索引 2）将第一个软键

；作为子 (**HSK1IsTask=0**) 通过参数 **<root>** 占用，

；德语软键文本通过默认值和英语 (**UK**) 规定

```
;。  
HSK1Task=2  
HSK1IsTask=0  
;应用程序专用参数： 此处跳转至  
;应用程序的基本层面， 参数： <root>  
HSK1Command=<root>  
HSK1SkText=Programm  
;没有预先为未设计的其他语言  
;设定语言数据  
HSK1SKText _UK=Program  
;英语 (_UK) 中的 SK 标签  
;使用任务服务 (任务索引 3) 将第五个软键  
; 作为子 (HSK1IsTask=0)用空命令参数占用，  
;软键文本为缺省值德语 (_GR) 和英语 (_UK)  
;规定。  
HSK5Task=3  
HSK5IsTask =0  
HSK5Command=  
HSK5SkText=DIENSTE  
HSK5SkText _GR=Dienste  
HSK5SkText _UK=service  
;软键“通讯”的访问权限  
HSK5AccessLevel= 3  
;使用“开机调试”任务 (任务索引 5) 将 ETC 软键栏的第 2 个软键  
; (任务索引 5) 作为子(HSK10IsTask=0)  
; 用空命令参数占用，  
; 规定德语 (_GR) 和英语 (UK) 软键文本  
;。  
HSK10Task=5  
HSK10IsTask=0  
HSK10Command=  
HSK10SkText _GR=开机调试  
HSK10SkText _UK=设置  
; TerminateTask=  
; HSK16=退出按钮
```

分配软键编号

对于 HSK 或者开始索引适用于下列分配:

1 - 8: 水平软键 1 - 8

9-16 ETC 软键栏的水平软键

17 - 24 垂直软键 1 - 8

25 - 32 用于 OEM 应用程序的软键栏

与语言相关的软键

根据当前设置的语言标记软键。在每次语言切换后更新软键。如果未找到当前语言的软键文本 (HSK<软键索引>SkText_<语言>=...), 则软键用缺省文本 (记录条目 HSK<软键索引>SkText=...进行分配, 或者在没有缺省文本时使用任务索引进行分配。

框架应用程序作为子任务

框架应用程序也可以由一个自身的进程开始作为子应用程序启动。然后框架应用程序通过一个回调命令返回到启动的应用程序, 并在必要时结束应用程序:

;应用程序在返回时结束

HSK<软键索引>TerminateTask=1 ; <> 0 结束

或者

;应用程序在返回时不结束。这是预设置!

HSK<软键索引>TerminateTask=0 ; 0 未结束

框架应用程序基本画面

在没有数据情况下通过 **Picture= ...** 显示一个空的背景。

可以给定一个自身的画面, 该画面在 **oem** 目录中作为位图使用, 例如 **Picture=背景框架1.bmp**。

在选择框架应用程序或者从一个框架应用程序调用的应用程序返回情况下, 显示画面。

数据 1 替代一个画面文件以显示基本画面“开机调试”, 默认。

说明

如果设计好 **开始索引** 和一个画面, 则在选择框架应用程序时会立即显示该画面, 并用于通过设计的开始索引支配应用程序。如果画面大于当前操作装置可提供的窗口, 则从画面中间开始仅显示窗口中适合的部分。

存取权限

框架应用程序的软键可以通过访问权限锁止。

0: 系统

... ..

7: 钥匙开关位置 0
 ; 举例:
 ;软键“通讯”的访问权限
HSK5AccessLevel= 3

NCK 有关的软键

通过 TASKCONF.INI 中相应的软键栏记录项下的 **HSK<软键索引>NckLink=1**，可以将一个软键同一个与 NCK 有周期相联系的存在条件相联系起来。在该连接存在的情况下，只能操作软键。默认值为 0：不检测。

在 TaskConf.INI 中缩短设计

当响应标准操作时，上述框架应用程序软键的 TASKCONF.INI 设计可以为下列跳转简化：

HSK<软键索引>任务 = x/y

x 操作区

y 命令或者状态 = i

示例：

HSK1Task=Program/<root>

表示进行的设计是相同的：

HSK1Task=2

HSK1IsTask=0

HSK1Command=<root>

HSK1SkText=Programm

表格 2-1 分配任务与软键文本

x	y	说明
Machine	<root>	通过最上层的回调命令跳回到机床的基本画面中（任务 0）中。软键文本的内容为 机床。
参数	<root>	通过最上层的回调命令跳回到参数的基本画面（任务 1）中。软键文本的内容为 参数。
Program	<root>	通过最上层的回调命令跳回到程序的基本画面（任务 2）中。软键文本的内容为 程序。
维修	<root>	通过最上层的回调命令跳回到通讯的基本画面（任务 3）中。软键文本的内容为 通讯。
诊断	<root>	通过最上层的回调命令跳回到诊断的基本画面（任务 4）中。软键文本的内容为 诊断。
诊断	状态=10	通过最上层的回调命令跳转到诊断的基本画面中用于返回和选择报警画面。 软键文本的内容为 报警。 提示： 可操作诊断的所有水平软键条。
诊断	状态=20	通过最上层的回调命令跳转到诊断的基本画面中用于返回和选择信息画面。 软键文本的内容为 信息。
诊断	状态=30	通过最上层的回调命令跳转到诊断的基本画面中用于返回和选择报警记录画面。软键文本的内容为 报警记录。
诊断	状态=40	通过最上层的回调命令跳转到诊断的服务画面中用于返回。软键文本的内容为 服务显示。
诊断	状态=50	通过最上层的回调命令跳转到诊断的 PLC 状态画面中用于返回。软键文本的内容为 PLC 状态。

x	y	说明
设置	<root>	通过最上层的回调命令跳回到 开机调试（任务 5） 的基本画面中。 软键文本的内容为 <i>开机调试</i> 。
设置	状态=10	在退出层面时通过回调命令跳转到机床数据的画面中。软键文本的内容为 <i>机床数据</i> 。
设置	状态=40	在退出层面时通过回调命令跳转到开机调试的 PLC 一览画面中。软键文本的内容为 <i>PLC</i> 。
设置	状态=50	在退出层面时通过回调命令跳转到开机调试的驱动一览画面中。软键文本的内容为 <i>驱动/伺服</i> 。
IB 设置	<root>	通过最上层的回调命令跳回到 设置（任务 34） 的基本画面中。 软键文本的内容为 <i>系统设置</i> 。
模拟	<root>	跳转到 模拟（任务 27） 。显示一个模态对话框，通过该对话框必须选择要模拟的程序。 通过取消可以再次退出模拟。软键文本的内容为 <i>模拟</i> 。

标准应用程序的其它界面

这里可响应下列标准应用程序：

- 参数
- 诊断
- 开机调试

下列命令应记录在软键命令定义的等号右侧：

HSK<软键索引>**Command**=指令 1; 指令 2; ...

或者

VSK<软键索引>**Command**=指令 1; 指令 2; ...

将分号用作多个命令之间的分隔符。

示例：

；隐藏一个标准操作的水平软键条某些软键

HSK1Command=DisableHSK (1, 3-4)

指令

DisableHSK(<软键列表>)

取消**软键列表**中的水平软键。可以通过逗号隔开各个软键或者给定软键范围 <从> - <至>，例如 **DisableHSK (1, 3, 5-7)**。

说明

ETC 软键条的软键无法取消。

DisableVSK(<软键列表>)

取消**软键列表**中的垂直软键。可以通过逗号隔开各个软键或者给定软键范围 <从> - <至>，例如 **DisableVSK (1, 3, 5-7)**。

DoVSK=<软键索引>

触发垂直软键（1-8）的动作。

Recall(<状态列表>)

在到达列表中的某个状态情况下，应用程序返回到调用的应用程序中。状态可以通过逗号隔开或者给定范围 <从> - <至>，例如 Recall（1、5-7、48）。

Highlight=<软键索引>

在调用应用程序时水平（1-8）或者垂直（9-16）的软键以蓝色高亮显示。

ZuMat=<状态矩阵>

读取分配的状态矩阵（例如 dg\dg.zus）并替换原来的状态矩阵。

RE_GR.INI

[HSoftkey-文本]

HSK7 =“特殊菜单”// 20

有关建立 HMI 高级的 OEM 应用程序的说明，参见：

文献： HMI 编程软件包

参见

将 HMI 的当前任务编号传输至 PLC (页 116)

TASKCONF.INI (页 346)

2.1.9 增加用户专用的服务显示

功能

借助文件 **DGOVW.INI** 可以在驱动诊断中定义附加的信号。必须在目录 `mmc2` 或 `addon`、`oem`、`user` 中明确建立文件。

如果已有文件，则会运用已设计的信号并在“诊断”→“服务显示”→“服务一览”中同时显示。为此，可以设计与语言相关和无关的文本。

这些与语言相关的文本以名称 **DgOvwTxt_XX.ini** 存在于目录 `mmc2\language`（或者 `addon`、`oem`、`user...`）中与语言相关的 INI 文件中。

在 `User` 目录中建立名称为 **dgovw.log** 的日志文件，输出在转换文件 `dgovw.ini` 时出现的故障信息中。

文件 DGOVW.INI 的结构

```
[GLOBAL]
NrOfSignals=

[BMP]
100="<位图名称>"
101="..."
...
[SIG1]
text=
Item=
expr=
.
.
[SIG2].
...
[SIG<n>]
...
```

条目含义

NrOfSignals:

附加信号的数目。信号位于段 [SIG<nr>] 中，此时 <nr> 从 1 至 NrOfSignal 排列（连续且无漏缺）。

Text:

与语言相关的文本形式 \$T<TextNr>

此时，<TextNr> 为与语言相关的文件 DgOvw_xx.ini 中的一个编号（带 xx 语言缩写）。

与语言相关的文本“<任意文本>”。

精确输出“<任意文本>”，其方式与该文本在 ini 文件中的输出方式一样。

对于区域或者行索引有下列设计方案：

<CH>	用于有效轴所在通道的编号。
<AX>	用于轴编号
<DRV>	用于所属轴的驱动编号
<PLC_CH>	用于一个 PLC 通道数据模块；由 20+通道编号代替，轴已在该通道中被激活。
<PLC_AX>	用于一个 PLC 通道数据模块；用 30+轴编号代替
	例如：Item = /Channel/MachineAxis/impulseEnable[u<CH>, <AX>] (Item: 参见 BTSS 说明)。

这种情况下，为当前通道的 <CH> 和 <AX> 设置当前的轴号码。包含的值可以以位方式或者作为整数评估。

expr: 这里可以有一个 IF 指令，或者分配一个位图文件。

句法:

expr=<IF_Statement> 或

expr=<Bmp_Zuweisung>

<IF_语句> ::= IF <Bool 表达式> THEN <语句> ELSE <语句>

<Bmp_分配> ::= BMP = <位图号>

<指令> ::= <项名称> 或

<指令> ::= <Bmp_分配>

<Bool 表达式> ::= <Val> 或

<Bool 表达式> ::= <Val><OP><值>

<Val> ::= # 或

<Val> ::= #.<位号>

<位号> ::= 应评估的位号码 (0..31)

<OP> ::= <, >, =, >=, <=, <>

表达式必须在一行中。

<项名称>::= 任意名称，该名称必须已在相同的段中定义，如同表达式。它也如同表达式一样处理。

<BitmapNr>::= 位图文件编号。编号 0 至 99 已预留用于西门子产品。自身的位图必须在段 [BMP] 中定义。

预定义的编号有下列含义：

0:		不显示位图
1:	OK	显示位图
2:	不正常	显示位图
3:	故障 → 发出报警	显示位图
4:	不相关	显示位图
5:	未分配驱动	显示位图
6:	通讯故障	显示位图

[BMP]

在该段中可以规定任意的自身位图。段 [BMP] 中的第一个位图必须有号码 100，第二个位图必须有号码 101，……

名称必须用双引号给出。如果未规定路径，则在目录 mmc2 或者 addon、oem、user，…中搜索位图，否则就在指定路径之中搜索。

示例：测量系统 1 信号已激活

```
[GLOBAL]
NrOfSignals=1
[BMP]
100="test.bmp"
101="c:\tmp\test2.bmp"

[SIG1]
text="测量系统 1 已激活"
Item= /Nck/MachineAxis/encChoice[u1, <AX>]
expr= if #=1 then BMP=100 else expr2
expr2= if #=0 then BMP=101 else BMP=2
```

文件 DGOVWTEXT_XX.INI 的结构

文件名中的 **xx** 代表与语言相关的文件的各个语言缩写。

仅有段 [TEXT]。

因此，列出各个文本的格式：

\$T<文本编号>=列出的 "<任意文本>"。

对于 <文本编号>，允许值为从 1000 至 32767，这允许只进行一次。低于 1000 的值已预留用于西门子产品。

必须有用于每个所需语言的文件。（语言缩写与 MMC.INI 中相同）

参见

DGOVW.INI (页 297)

2.1.10 设置 OEM 应用的宽屏显示

使用宽屏显示

HMI 程序在一个较大的屏幕上作为“宽屏显示”显示，OEM 应用程序区保留在上部区域。

上部屏幕

该区域称作上部屏幕，该区域位于 HMI 程序输出区域的上方。该区域像素为 1280 * 254。

前提条件

为了激活宽屏显示功能，有下列前提条件：

- 要使用宽屏显示功能，屏幕分辨率至少为 1280*1024 像素，并且在配置文件 REGIE.INI 的段 [UpperScreen] 中设置功能 "UpperScreen" 为 TRUE。
过程控制位于 MMC.INI 文件 [CONTROL] 段的条目 HMI_Start 中 HMI 程序的启动位置后面。
- 要使用宽屏显示的 OEM 应用程序必须用当前 OEM 包和过程控制扩展重新生成。

激活功能

配置文件 REGIE.INI 中的段：

```
[UpperScreen]
; HMI 宽屏的上部屏幕区域。
; 通过独立的本地窗口管理上部屏幕区域
; 独立于 REGIE 管理的应用程序。
; 上部屏幕区域具有独立的软键区。
; 该上部屏幕软键区只在
; 专用的 OEM 操作面板上可用。
; 在此处激活该功能
UpperScreen = TRUE
; 通知上部屏幕主窗口，
; 从 Regie. 传输软键。
; WindowName
UpperScreenWindowName = ""
; Window ClassName
UpperScreenClassName = ""
```

示例

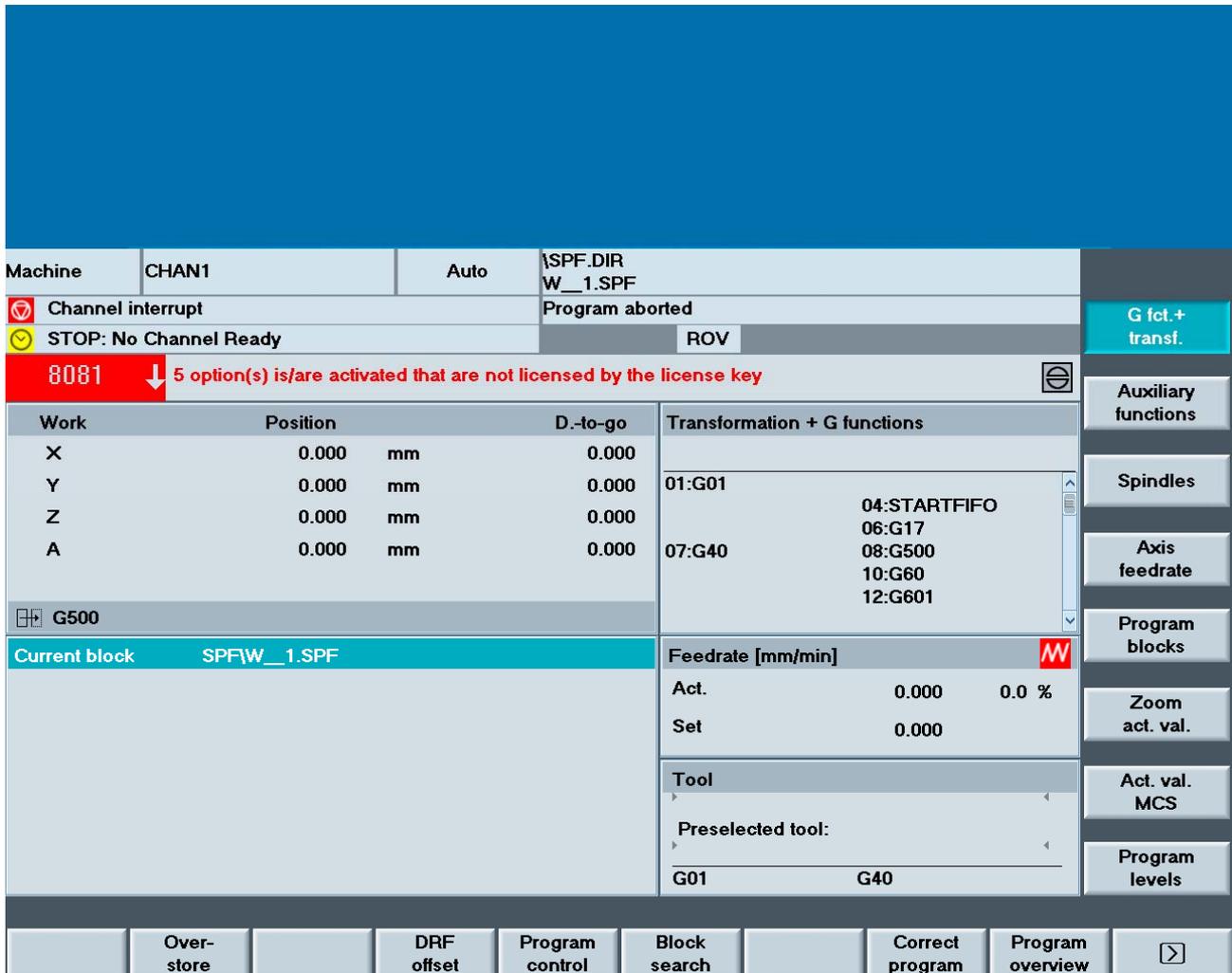


图 2-2 HMI 高级示例

操作 HMI 程序和 OEM 应用程序



在 HMI 程序和 OEM 应用程序之间切换时，使用窗口切换键 <NEXT WINDOW> 或者通过鼠标点击。

OEM 应用程序通过按下一个 OEM 键或在主窗口中鼠标点击实现操作。所有其他的按键输入都在这激活的窗口中进行，直至识别到一个 HMI 程序功能键 (F1 ... F10, ...)。然后再次激活 HMI 程序。

预留下列按键组合：

<Ctrl+1>, <Ctrl+2>, <Ctrl+3>, <Ctrl+6>, <Ctrl+7>, <Ctrl+8>

2.1.11 建立特殊工艺的文本

功能

在文件 IF.INI 中提供有下列用于参数化帮助功能的段：

- TECHNOLOGY
- TEXTFILES
- CONTEXT
- TEXTSEARCH

段 [TECHNOLOGY]

在[TECHNOLOGY]段中，通过机床工艺属性的预设值，指令文本的显示限于与机床相关的输入项。然后只显示所属工艺的输入项。

此时要考虑工艺标识：

"m" = 铣削

"t" = 车削

"g" = 磨削

"s" = 步冲

"p" = 冲裁

"a" = 显示全部工艺

段 [TEXTFILES]

在[TEXTFILES]段中设置路径参数（路径\名称），以简短帮助的文本文件为基础的帮助功能。

帮助功能访问下列文本文件：

- Siemens dh\cst.dir\lfs_gr.com
- 机床制造商 dh\cst.dir\lfm_gr.com（任意文件名）
- 最终用户 dh\cst.dir\user_gr.com（任意文件名；同样通过“设置”进行输入）

说明

西门子文本文件“Dh\Cst.dir\lfs_gr.com”的路径必须已经存在，否则会出现相应的故障信息。

使用 ISO 语言 西门子条目规定为：

SiemensI=文件

车削时文件为 ifit_xx.com 或者

铣削时文件为 ifim_xx.com。

文本路径分配时，对于文本文件的名称有不同的参数设置方法。

下表包括了参数设置方法和其帮助功能解释：

用于文本文件的参数	解释
文件名.com	读出与语言无关的文本文件
文件名_XX, com	读出与语言相关的文本文件
文件名_.com	进行参数化时，在存在的情况下首先读取与语言无关的文本文件，否则读取与语言相关的文本文件。

(XX = 语言的缩写,比如“GR”代表德语)

举例: Enduser=Cus.dir\lfm_.com

首先在路径 “C:\Dh\Cus.dir\”打开与语言无关的文本文件“lfm.com”，如其不存在再打开各个与语言相关的文本文件（比如“lfm_gr.com”）。

摘录自简短帮助 “lfm_gr.com”的文本文件：

...

a2//G00/快进线性插补 (m)

a2//G01/进给线性插补 (m)

a2//G02/ 顺时针圆弧插补 (m)

...

段 [CONTEXT]

在[CONTEXT] 段中，可以设置一个扩展的上下文相关性。用“1” 激活，用“0” 取消激活（也可以通过“设置”输入）。

如果扩展的上下文相关性激活且光标位于指令文本旁或指令文本上，则在调用帮助时显示所有带有相同起始原文的指令。

例如程序员选择了指令 “G1”，然后所有的指令以“G1”为首顺序显示（例如：“G1”、“G17”、“G18”、“G19”）。

如果输入 "0" 没有激活扩展的上下文相关性，则（如果有的话）只显示当前指令。

段 [TEXTSEARCH]

在[TEXTSEARCH] 段中，可以对文本查找进行详细说明。

对此有以下方案：

1 = 仅指令文本

2 = 仅说明文本

3 = 指令和说明文本（也可以通过“搜索”加入）。

参见

IF.INI (页 308)

2.1.12 创建带有工作列表的工件

前提条件

存在选项“多通道程序段编程”和具有 20 条通道的多通道机床是前提条件。可以进行脱机运行，但在此情况下无法检测通道。

功能

工件的模板(Templates)提供了建立程序运行过程的简单方法，程序运行过程的形式为带工作表的工件。可使模板与现存工件匹配。

在操作区程序中以“编辑工件”编辑工件。如果既没有工作表、零件程序，也没有具有工件名的 DAT 文件，则会显示故障信息：不能打开工件。如果通过文件 MMC.INI 的条目激活了工件的自动匹配，则不显示故障信息，而是尝试在打开前完备工件。

参数设置

文件 MMC.INI 中的段 [Programm]给出了下列参数:

[Program]

; 1=通过工件模板完备工件，
; 如果用 MCSP 无法打开工件（预设）
; 0=有故障信息：
; 基于模板更新工件=1

在包含 JOB/DAT 文件的 Template（模板）中只存在一个工件模板。模板内容会自动复制入选择的工件中，从而不会覆盖当前工件现有的文件。如果之后存在一个具有工件名的 JOB/DAT 文件，则打开步进链编辑器，否则显示故障信息。

如果在包含 JOB/DAT 文件的 Template（模板）中存在多个工件模板，则显示对话框“完备工件”，可以在其中选择需要的工件模板。

如果在激活“编辑工件”时选择了工件不含 INIT 指令的零件程序，则调用对话框“完备工件”，以选择 Template（模板）内存在的 JOB/DAT 文件（只要有的话）。如果选择了一个 JOB/DAT 文件，该文件会以零件程序名复制入工件并用该文件打开步进链编辑器。此时不提供工件模板，只在选择工件时提供工件模板。

关于工件中所包含模板文件的文件和信息，会输出显示在记录窗口中。

故障信息提示

在下列情况下显示故障信息：

如果选择了不包含有效 INIT 指令的零件程序：

没有查找到零件程序的 Init 指令！

在 JOB/DAT 文件中以不存在的文件为参考：

响应下列不存在的程序：

在 JOB/DAT 文件中以不存在的通道号为参考：

向下列通道分配程序，

这些程序不存在：

如果无法生成 DAT 文件中的条目：

没有查找到通道分配！

多通道程序段编程的其他设计方式：

参见

MMC.INI (页 321)

SEEDITOR.INI (页 335)

2.1.13 刀具管理 (WZV)

功能

在 HMI 高级刀具管理中已包含下列功能。

刀具管理的 NCDDE 变量根据下列标准建立:

"TMHMICurData_<MMCName>_"<DataName>

TMHMI = HMI 刀具管理

有下列当前数据 (<数据名称>):

用于 TOA 和通道的一般数据:

- curTOANo 当前 TOA 编号

- curChannelNo 当前通道号

列表视图 (其中有写入标记的刀具) 和各个刀具的刀具数据画面中的当前刀具:

- curToolTNo T 编号

- curToolIdent 刀具名称

- curToolDuplo 刀具 Duplo 号码

- curToolType 刀具类型

- curEdgeNo 刀具刀沿号码, 与刀具相关,
不针对 D 号码!

- curDLNo 刀具刀沿补偿位置编号,
相对于刀沿

- curMagNo 刀库号码: "0", 在当前刀具
不在刀位上或为此已规定。

- curPlaceNo 刀库位置编号: "0", 在当前刀具
不在刀位上或为此已规定。

刀库列表视图中的当前刀库:

- curMagLiMagNo 刀库号码,

装载、卸载、转换、定位和空位置搜索时的目标刀库和目标刀库位置:

- targetMagNo 刀库号码

- targetPlaceNo 刀库位置号码

装载、卸载、转换、定位和空位置搜索时的源刀库和源刀库位置:

- sourceMagNo 刀库号码

- sourcePlaceNo 刀库位置号码

刀具箱中的当前刀库:

- curCabToolIdent 刀具名称

- curCabToolDuplo 刀具 Duplo 号码

- curCabToolType 刀具类型
刀具目录中的当前刀库
 - curCatToolIdent 刀具名称
 - curCatToolDuplo 刀具 Duplo 号码
 - curCatToolType 刀具类型
- 有关开机调试所需信息的详细说明，请参见：
文献: 刀具管理功能手册

2.1.14 无刀具管理情况下的刀具选择

功能

应用程序“无刀具管理情况下的刀具选择”提供本地 NCDDE 变量，这些变量用于描述当前所需的刀具。为此，将刀具管理功能用于参数下的刀具选择。写入该变量的时机通过 PARAM.INI 中的设置控制：仅在切换到“补充操作界面”的画面时，或者在进行每个参数状态切换时。

用光标在无刀具管理情况下的刀具画面中选择刀具。通过光标定位到一个刀具上，为该刀具写入内部 T 号码和本地 NCDDE 变量中的刀沿号码。这些变量可以由“补充操作界面”评估。

一个变量中的所有数据

NCDDE 变量的名称由下列标准得出：

"PAHMICurData"<MMCName>，带有 MMC.INI 中的 MMC 名称。

因此，通过前缀来区分刀具管理的变量。

说明

段 [GLOBAL] 中的条目 NcddeMmcName 和 NcddeMmcName 不得保护标准值“_xxxx_”，否则，该值会被任意一个数字所替代。

在这些变量中，将当前 PAHMI 数据作为字符串执行，以下列格式：

"curToolTNo=35;curToolDNo=5;" "=" 用作为数据名称和数据值之间的分隔符，";" 用作为数据之间的分隔符

各个变量

另外，每个数据都有一个自身的 NCDDE 变量，其名称由下列部分得出：

"PAHMICurData_"<MMCName>"_"<DataName>

段中带有文件 mmc.ini 中的 <MMC 名称>：记录条目 NcddeMmcName 中的 [GLOBAL]。

说明

NCDDE 服务器不允许多变量访问 NCDDE 变量，以此在各个访问中进行写和读。

有下列用于 TOA 和通道的当前数据 (<DataName>) :

- curTOANo 当前 TOA 编号

- curChannelNo 当前通道号

列表视图 (其中有写入标记的刀具) 和各个刀具的刀具数据画面中的当前刀具:

- curToolTNo T 号码

- curToolType 刀具类型

- curEdgeNo 刀具刀沿号码, 与刀具有关, 但与 DNo 无关!

当前未识别的值在 NCDDE 变量中作为“varname=”或在该变量中取消。数据顺序未确定。只要还未设置数据值, 则其 NCDDE 变量为空或不存在。

接口控制

接口的动作可以通过 PARAM.INI 控制。

[General]

; 选项“在切换至 WIZARD 形式或 WIZARD 软键时

; 将人机界面刀具管理的当前数据

; WIZARD 形式或 WIZARD 软键时”:

; 所有选项必须通过

; 在单行中使用命名参数设置。

; 选项

; 通过值“真”“打开”且

; 通过值“假”“关闭”, 或如果命名参数

; 不在一行中。

; "EnableAllTogetherWriteToNcdde := True":

; 一个 ncdde 变量中的所有数据

; "EnableSingleWriteToNcdde := True":

; 每个数据一个 ncdde 变量

; 这两个选项都可激活。

; 如果这两个选项未激活,

; 将不会写入至 ncdde 变量。

; "WriteChangesWhenStateChanged := True":

; 按下软键时写入改变,

; 即使在未切换至 WIZARD 情况下

下列设置必须作为一个单独的行 PARAM.INI 存在:

```
HMICurDataInterface = EnableAllTogetherWriteToNcdde := True,  
EnableSingleWriteToNcdde := True,  
WriteChangesWhenStateChanged := False
```

2.1.15 硬盘处理 (m:n 配置)

前提条件

该功能仅适用于 SINUMERIK powerline。

在文件 NETNAMES.INI 在其它情况下描述如下：

- 多个 HMI 组件和多个 NCU 组件的分配配置 (m:n 配置)。

如果已经为一个 NCU 已分配了多个 HMI 组件，则必须为那些在“外部处理 (硬盘)”时提供零件程序的单个组件命名。

- 多通道设备的通道菜单配置
- 参考带有用于 PLC 信号的符号的文件

EXTCALL 功能

为了让位于 PCU 硬盘上数据管理目录中的外部程序能够由零件程序通过程序命令 EXTCALL 处理，需要使用文件 NETNAMES.INI 中段[conn HMI_x]里的下列记录条目：

```
; HMI 识别号的识别部分  
[own]  
owner= HMI_3  
; Description of possible connections  
; 连接部分  
[conn HMI_3]  
conn_1= NCU_1  
conn_2= NCU_2  
conn_3= NCU_3  
conn_4= NCU_4  
EXTCALL_conns=conn_1, conn_2, conn_3, conn_4  
; conn_i NCU 名称，由该名称可以在 PCU 上  
; (带 HMI) 进行访问 (参见提示)  
; Description of significant net-parameters  
[param network]  
bus= btss
```

```
; HMI 说明
[param HMI_3]
HMI_typ = 0x40
HMI_.....

; NCU 组件的描述部分
[param NCU_1]
type =NCU_573
nck_address= 11
plc_address= 11
name = NCU1
...

; 参考带用于 PLC 信号 NCU_1 的符号的文件
PlcSymbolFile=PlcSym
```

说明

条目 EXTCALL_Conns ... 仅为文件 NETNAMES.INI “主操作区”上的必要项目，而不是“副操作区”上的必要项目。

文献： 扩展功能功能手册： 多个 NCU 上的多个操作面板，分散式系统（B3）

参见

NETNAMES.INI (页 327)

2.1.16 激活 V.24 接口

激活 V.24 接口

在文件 DINO.INI 中可以在段 [System] 中激活串行 V.24 接口。为此可以针对接口 V.24/PG (COM1/COM2) 存储和分配文件。

预设为 0（没有接口参数化）。

[System]

；通过设置非零值激活 V24 设置

V24 设置=0

步骤

为了在 HMI 高级 7.5 的 PCU 50.3 上使用 V.24 数据传输，需要下列步骤：

1. 在 PCU 50.3 的 BIOS 中，在“Advanced → I/O Devices Configuration”下设置“Internal COM1”为“Disabled”。
2. 将接口卡 “PCI Interface Card (COM/LPT)”装入 PCU 50.3 中并安装随附的驱动程序。
3. 在 Windows 设备管理器中命名 COM1，COM2 上的 COM 接口。
4. 在 Windows 设备管理器中，对于 COM 接口，总是在 "Port Settings → Advanced" 使选项 "Enable CTS/RTS Auto Flow Control" 退出激活状态。
5. 在配置文件 DINO.INI 中激活 V.24 接口。

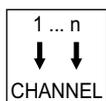
参见

DINO.INI (页 295)

2.2 设计通道菜单

2.2.1 通道菜单应用

应用



通道菜单通过用于显示的通道切换键激活。

设计在文件 NETNAMES.INI 中实现。对于 SINUMERIK powerline，操作站切换需要选项“操作单元管理”。为此可最多驱动 9 个 PCU 在 9 个用于 m:n 配置的 PCU 上，或者驱动 1 个 PCU 在最多 20 个 NCU 上（powerline）。

通道菜单可以用于下列应用：

- 不带操作单元管理 (1:1 配置) 用于切换 NCK 通道，在一个较大的 NC 通道数情况下用于直接通道选择（本地通道切换）。使用一个直接的通道选择替代一个顺序的通道切换，通过通道切换键到所需的通道。
- 不带操作单元管理 (1:1 配置) 用于切换双通道显示时的显示（这里不能显示通道菜单！）
- 带操作单元管理 (M:N 配置) 用于在另一个 NC 通道上切换操作单元 (OP 和 MCP)，可以精确的切换到另一个 NCK 上。切换整个 HMI 系统。这对于 SINUMERIK powerline 是操作单元管理模式。
- 通过操作单元管理，切换带 TCU 的操作站到另一个 HMI 上 (以此到另一个 NCK 上) 或者到另一个显示的 NCK 的通道上，在显示的 HMI 保留情况下 (T:M:1 方案)。这对于 SINUMERIK solution line 是操作单元管理模式。

用于说明对 SINUMERIK powerline 和 solution line 区别的图像描述。不描述一个本地通道切换（在保留连接的 NCK 情况下）：

对于 SINUMERIK powerline 切换过程：

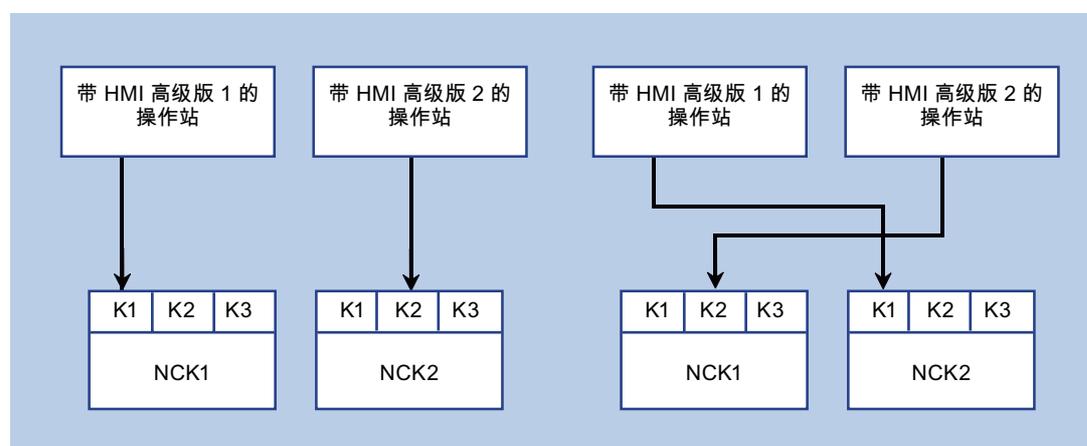


图 2-3 切换 (powerline)

初始情况:	切换后:
HMI 高级 1 在 NCK1.通道 1	HMI 高级 1 在 NCK2.通道 1
HMI 高级 2 在 NCK2.通道 2	HMI 高级 2 在 NCK1.通道 2

对于 SINUMERIK solution line 切换过程

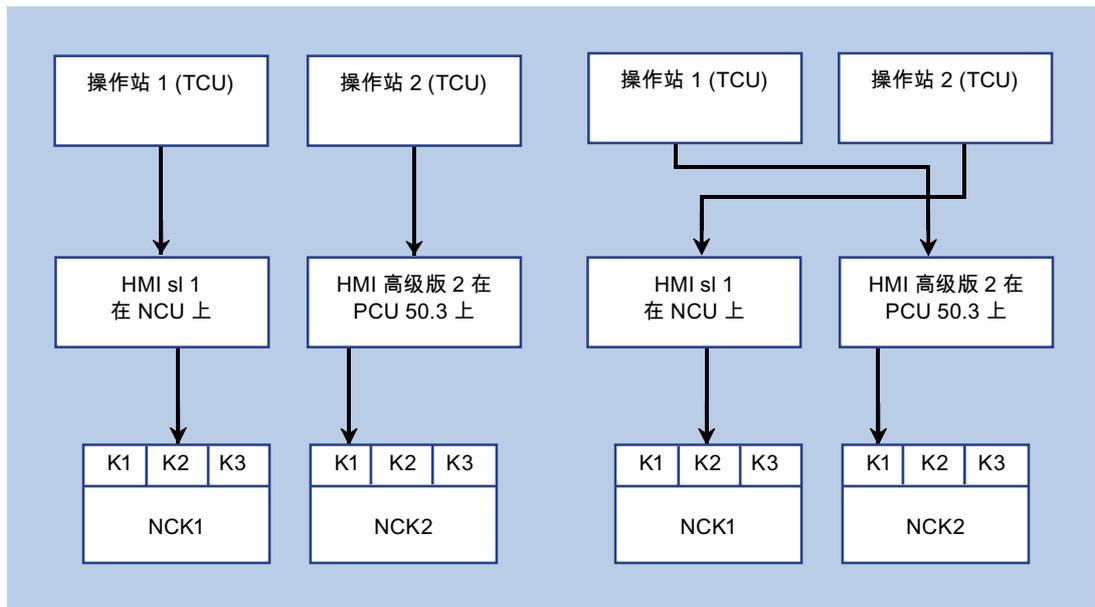


图 2-4 切换 (solution line)

初始情况:	切换后:
操作站 1 (在 HMI1 上) 在 NCK1.通道 1	操作站 1 (在 HMI2 上) 在 NCK2.通道 1
操作站 2 (在 HMI2 上) 在 NCK2.通道 2	操作站 2 (在 HMI1 上) 在 NCK1.通道 2

2.2.2 通道菜单结构

一览

在双通道显示应用情况下为了切换通道菜单操作，定义一个通道组列表。

通道组列表由一个或多个通道组组成。通道组由一个或多个 **NC** 通道组成。在通道菜单中通过水平软键可选择通道组。所选通道组的通道用垂直软键可选择。作为可触发的切换目标，总是位于一个具体 **NCK** 的 **NC** 通道。为此可以在一个通过通道菜单触发的切换到另一个通道上明确另一个 **NCK** 上切换。

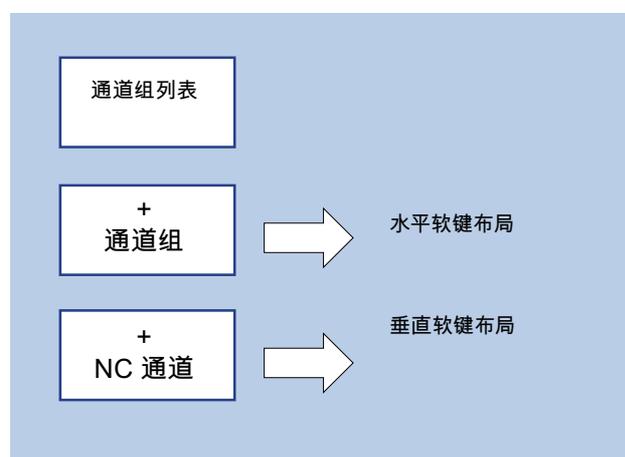


图 2-5 分配软键

说明

通道组可以在一个设备中例如符合一个加工站。

在一个通道菜单中可以最多设计 32 个通道组，各带最多 8 个通道。

2.2.3 设计用于直接通道选择的通道菜单 (1:1 配置)

设计通道菜单

在配置文件 NETNAMES.INI 中设计通道菜单，参见下列示例： 在一个 NCU 上设计 4 个通道组，带不同的通道数。

```
.*****  
,  
; ChanMenu  
.*****  
;  
[chan MMC_1]  
DEFAULT_logChanSet = Stat_1  
DEFAULT_logChan = CH_1  
ShowChanMenu = true  
logChanSetList = Stat_1, Stat_2, Stat_3, Stat_all  
  
.*****  
,  
; 定义通道组  
  
[Stat_1]  
logChanList = CH_1, CH_2, CH_3, CH_4  
  
[Stat_2]  
logChanList = CH_5, CH_7, CH_8  
  
[Stat_3]  
logChanList = CH_6  
  
[Stat_all]  
logChanList = CH_1, CH_2, CH_3, CH_4, CH_5, CH_6, CH_7, CH_8  
  
.*****  
,  
; 定义通道  
  
[CH_1]
```

logNCName = NCU_1

ChanNum = 1

[CH_2]

logNCName = NCU_1

ChanNum = 2

[CH_3]

logNCName = NCU_1

ChanNum = 3

[CH_4]

logNCName = NCU_1

ChanNum = 4

[CH_5]

logNCName = NCU_1

ChanNum = 5

[CH_6]

logNCName = NCU_1

ChanNum = 6

[CH_7]

logNCName = NCU_1

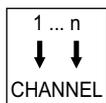
ChanNum = 7

[CH_8]

logNCName = NCU_1

ChanNum = 8

切换通道



在按下带有该设计的通道切换键时显示下列通道菜单。

用通道切换键从任意一个 HMI 操作区中选择通道菜单，如下示例：

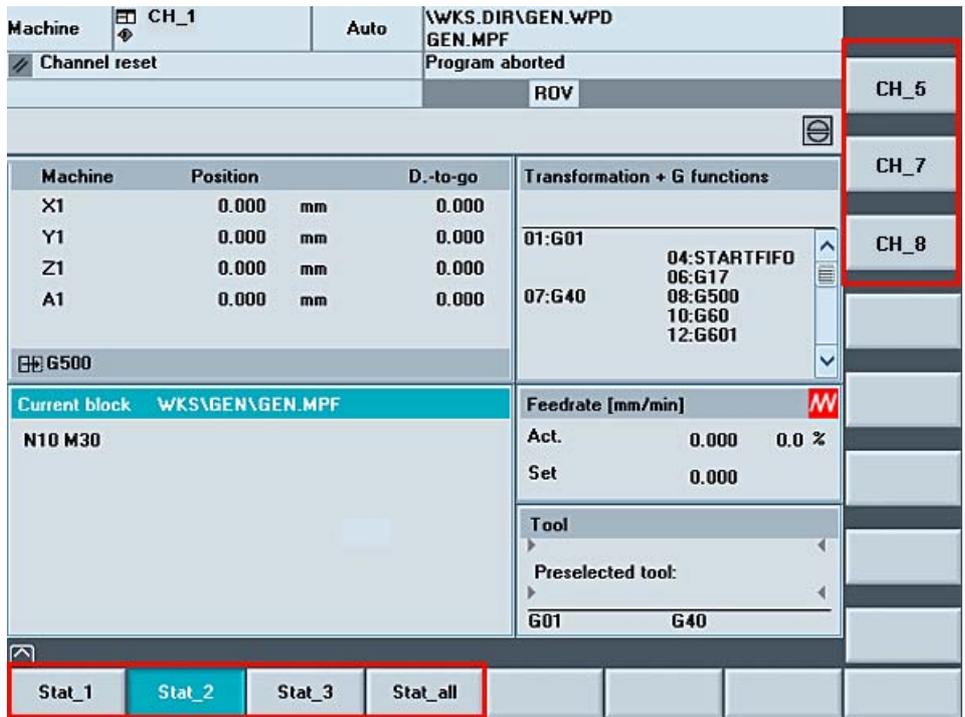


图 2-6 通道菜单示例

触发通道切换:

- 用水平软键选择通道组（站）。
- 在垂直软键上显示该组附属的通道。
切换通道首先通过垂直软键实现。

参见

NETNAMES.INI (页 327)

2.2.4 设计一个双通道显示

设计双通道显示

如果使用一个双通道显示，则不能使用操作单元管理。为了可以在操作区“加工”中使用双通道显示，在文件 NETNAMES.INI 中需要下列条目：

```
[own]
```

```
owner= HMI_1
```

```
[chan HMI_1]
```

```
ShowChanMenu=false
```

```
[MULTI_CHAN_i]
```

```
CHAN_2 = k
```

此时 “i”表示当前 NCU 中的通道号码。为此显示的第二通道特定在条目 CHAN_2 中：
CHAN_2 = k

该设计的预设值仅在从第 1 个通道显示切换到第 2 个通道显示时有效。一旦第 2 个通道显示有效且在显示内部切换通道时，其它通道不受影响。

说明

也可以使用双通道显示或者操作单元管理

对于双通道显示，设置 ShowChanMenu=false。

每个参与双通道显示分配的通道，会在 [MULTI_CHAN_i] 中显示一次并在 CHAN_2=i 中被显示为第 2 通道。

对于双通道显示，显示-MD 9034： MA_NUM_DISPLAYED_CHANNELS 设置为 2。

一个双通道显示举例

在一个带 6 通道 (1, 2, 3, 4, 5, 6) 的 NCU 上, 要一起显示通道组合 1 和 6, 2 和 5 以及 3 和 4:

```
[MULTI_CHAN_1]
```

```
CHAN_2 = 6
```

```
[MULTI_CHAN_6]
```

```
CHAN_2 = 1
```

```
[MULTI_CHAN_2]
```

```
CHAN_2 = 5
```

```
[MULTI_CHAN_5]
```

```
CHAN_2 = 2
```

```
[MULTI_CHAN_3]
```

```
CHAN_2 = 4
```

```
[MULTI_CHAN_4]
```

```
CHAN_2 = 3
```

2.2.5 在 m:n 配置时操作单元管理的通道菜单 (powerline)

m:n 配置

作为 m:n 配置，形成一个由 m HMI 高级单元和 n NCU 单元组成的组合。此时用于操作的 HMI 单元在运行模式下可以自由切换到各个 NCU。对于各个 HMI 的切换目标，针对每个 HMI 单独设计一个通道菜单在文件 NETNAMES.INI 中。各个切换目标成双由 NCU 和 NCU 上规定的加工通道组成。因为在切换一个 HMI 到另一个 NCU 上时，也必须激活该 HMI 的机床控制面板 (MCP)，必须在设计中也针对 HMI 分配机床控制面板。

设计的通道菜单在 HMI 上通过按下通道切换键调用。然后通过软键操作选择切换目标并触发切换。

注意

对于一个 NCU 同时仅允许存在 2 个激活的连接 (HMI)。这在 NETNAMES.INI 中设计时，条目 DEFAULT_logChanSet 和 DEFAULT_logChan 中，启动后确定哪些使用的连接，在所有 HMI 上都应考虑。

通过设计，如在 NETNAMES.INI 中规定附件，描述下列切换可能性：

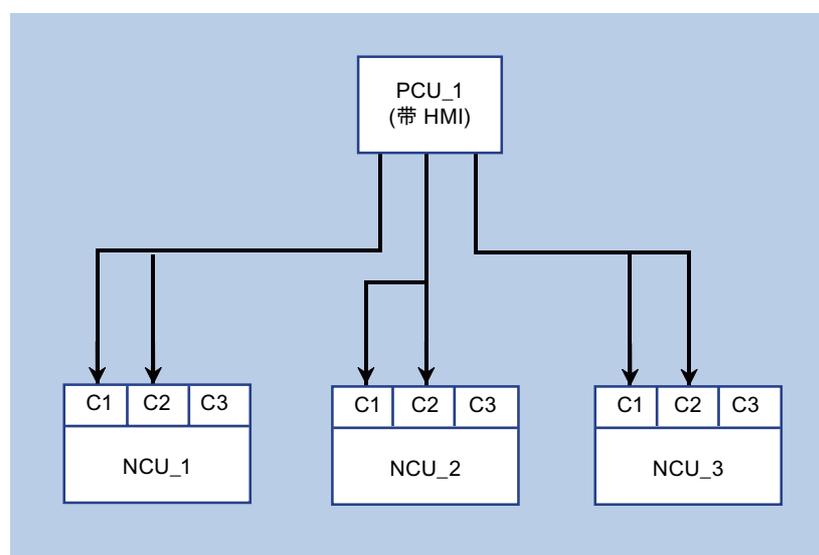


图 2-7 切换可能性

另见

- 用于该应用情况的通道菜单设计：NETNAMES.INI (页 327)
- 用于描述 HMI 高级中的通道菜单：设计用于直接通道选择的通道菜单 (1:1 配置) (页 54)

2.2.6 在 m:n 配置时操作单元管理的通道菜单 (solution line)

t:m:n 配置

作为 t:m:n 配置，由 t 个操作站 (TCU 带 MCP)，m 个 HMI 系统和 n 个 NCK 系统组成一个组合，在此操作站在运行模式下可以在不同的用于操作的 HMI 系统上自由选择切换。对于各个操作站的切换目标，针对每个操作站单独以一个二级步骤在通道菜单中设计切换目标。

HMI 系统此时固定保留与分配的 NCK 的连接。每个 HMI 是否是一个在 NCU 上运行的 embedded HMI，或者是一个在一个 PCU 上运行的所谓的外部 HMI，这并不重要。此外在各 NCK 上仅允许连接一个 HMI (静态)，即如果一个外部 HMI 高级要操作一个 NCU 的 NCK 时，则那里的 HMI sl 必须退出激活 (参见：服务命令 `sc disable HMI`)。

在运行模式中通过设计的通道菜单从 HMI 中初始化切换过程。

一个 t:m:n 模式从 HMI 外观上存在，当一个通道菜单在文件 NETNAMES.INI 中设计情况下。

可以切换目标可直接通过 HMI 地址和可选的一个 NC 通道 (该通道在切换 HMI 时加入) 规定。切换目标可以以逻辑方式，所谓的通道组合并。通道组在通道菜单中的水平软键上显示，自身的切换目标 (HMI, NC 通道) 显示在通道组的垂直软键上。

二级步骤

在第一步中在文件 NETNAMES.INI 中定义设备上要使用的所有通道组，和具体操作站无关。如下规定一个通道组：

```
[CH_GRP_1] /* 通道组名称；标记必须根据该主题进行。
/* 该组在最后位置连续编号。
/*
/*

Text = Bearbeitung1 /* 通道组软键文本用于在通道菜单中显示
/* (水平软键)

Member1.Target = NCU730PN-IP3.#1 /* 通道组切换目标 1
/* 计算机名称数据 (主机名) 或者一个计算机
/* 的固定 IP 地址，在该计算机上运行所选
/* 的 HMI (NCU 或 PCU)，带
/* "#Kanalnummer" 可以作为可选规定切换
/* 时要加入的 NC 通道。
/*
/*
```

```
Member1.Text = NCU730_K1          /* 软键文本用于通道菜单中显示（垂直软  
/* 键）文本，这些文本包含空格，在 ""  
/* 内。  
  
Member2.Target =                  /* analog  
Member2.Text =                    /* analog
```

文件 NETNAMES.INI 必须分布在所有运行 HMI 的系统 (NCU, PCU) 上，即对于 HMI sl 在 NCU 上必须将文件总是存放在目录 /oem/sinumerik/hmi/cfg 或 /user/sinumerik/hmi/cfg 下，对于 HMI 高级存放在目录 F:\OEM 或 F:\USER\ 下。

在第二步中，单独用于操作站，在各个站自身的文件 CONFIG.INI 中段 [T2M2N] 下通过用于该站相关通道组数据定义具体操作站的特性。

文件 CONFIG.INI 位于 NCU 上的目录中：

```
/user/common/tcu/<TCU-NAME>/common/tcu/config.ini
```

在 PCU 50.3 上存放在：

```
F:\user_base\common\tcu\<TCU-NAME>\common\tcu/config.ini
```

对于 <TCU-NAME>，使用操作站的 TCU 名称，如同在第一次启动 TCU 时在设备上规定的名称。

HMI sl 和 HMI 高级之间的区别

对于 HMI 高级，必须另外针对用于切换操作站的通道组，一并设计直接通道选择的通道菜单。对于 HMI sl，不必分析该设计或可以缺失该设计。

HMI 高级特性如下：

- 如果 PCU 50.3 有一个本地显示器和操作权限，在这种情况下通过通道菜单实现一个直接的通道选择。
- 如果 PCU 50.3 上带 TCU 的操作站有操作权限，该权限在文件 CONFIG.INI 中没有设计通道菜单，则通过通道菜单实现一个直接的通道选择。
- 如果 PCU 50.3 上带 TCU 的操作站有操作权限，该权限在文件 CONFIG.INI 中设计了一个通道菜单，则通过通道菜单实现一个符合设计的操作站切换。

HMI sl 特性体现如下：

- 如果一个操作站有操作权限，该权限没有设计通道菜单，则通道切换键本地有效，不带通道菜单，即继续切换到下一个 NC 通道。
- 如果操作站有操作权限，该权限在文件 CONFIG.INI 中设计了一个通道菜单，则通过通道菜单实现一个符合设计的操作站切换。

另见

用于描述 HMI 高级中的通道菜单：设计用于直接通道选择的通道菜单 (1:1 配置) (页 54)

设计通道组 (NETNAMES.INI)

```
[own]
owner= HMI_1

[chan HMI_1]          /* 直接通道选择的通道菜单
DEFAULT_logChanSet=Kanaele /* 只用于 HMI 高级
DEFAULT_logChan=K_1      /* 对于 HMI sl 无意义
ShowChanMenu=True
logChanSetList=Kanaele

[Kanaele]
logChanList=K_1, K_2, K_3, K_4

[K_1]
logNCName=NCU_1
ChanNum=1

[K_2]
logNCName=NCU_1
ChanNum=2

[K_3]
logNCName=NCU_1
ChanNum=3

[K_4]
logNCName=NCU_1
ChanNum=4

;***** /* TCU 通道名称
[chan HMI_1]          /* HMI 高级一般部件
ShowChanMenu = true /* 和 HMI sl
```

;通道组
;从通道组关联到水平软键
;-> 参见 TCU 的 config.ini

[CH_GRP_1]

Text = Bearbeitung1

Member1.Target = NCU730.#1

Member1.Text = NCU730_K1

Member2.Target = NCU730.#3

Member2.Text = NCU730_K3

Member3.Target = NCU730.#4

Member3.Text = NCU730_K4

[CH_GRP_2]

Text = Bearbeitung2

Member1.Target = NCU710-2.#1

Member1.Text = NCU710_K1

Member2.Target = NCU710-2.#2

Member2.Text = NCU710_K2

[CH_GRP_3]

Text = Transport

Member1.Target = NCU720-1.#1

Member1.Text = NCU720_K1

Member2.Target = NCU720-1.#2

Member2.Text = NCU720_K2

Member3.Target = NCU720-1.#3

Member3.Text = NCU720_K3

Member4.Target = NCU720-1.#4

Member4.Text = NCU720_K4

...

[CH_GRP_15]

Text = Kopf-Station

Member1.Target = 192.168.214.241.#1

Member1.Text = NCU720_K1

Member2.Target = 192.168.214.241.#2

Member2.Text = NCU720_K2

Member3.Target = 192.168.214.241.#3

Member3.Text = NCU720_K3

Member4.Target = 192.168.214.241.#4

Member4.Text = NCU720_K4

设计操作站

操作站通道菜单设计在文件 CONFIG.INI 的段 [T2M2N] 中实现。在那里操作站要使用的通道组如在文件 NETNAMES.INI 中定义的，为操作站分配通道菜单的水平软键。

文件 CONFIG.INI

...

[T2M2N]

SK1=CH_GRP_1

SK2=CH_GRP_3

SK8=CH_GRP_15

软键参考 CH_GRP_xx 涉及到必须在 NETNAMES.INI 中定义的通道组。标记是绑定的。

通过该示例和 NETNAMES.INI 中预设的设计确定操作站通道菜单如下配置：



图 2-8 水平软键条

2.3 连接网络驱动器

2.3.1 设置访问外部驱动器或者计算机

使用

可以将控制系统与外部网络驱动器或其它计算机相连接。例如，通过此连接，可以由其它计算机处理零件程序。

为此，有两种可选方案：

- 通过参数化显示机床数据可以最多连接 4 个驱动器。
- 通过设计文件 LOGDRIVE.INI 可以最多连接 8 个驱动器。

这两种方案不能同时使用。在配置多个 TCU 或 PCU 的情况下优先使用文件 LOGDRIVE.INI，以能够在全球范围内通过网络访问该驱动器。该文件用户专用的条目存储在目录 user 中。

参见

LOGDRIVE.INI (页 309)

2.3.2 使用显示机床数据来连接驱动器

设计驱动器/计算机

用下列显示机床数据可以设立最多 4 个网络驱动器:

MD 9676 CMM_DIRECTORY_SOFTKEY_PATH1="Pfad1"

MD 9677 CMM_DIRECTORY_SOFTKEY_PATH2="Pfad2"

MD 9678 CMM_DIRECTORY_SOFTKEY_PATH3="Pfad3"

MD 9679 CMM_DIRECTORY_SOFTKEY_PATH4="Pfad4"

必须输入完整的路径。在软键上显示路径。

连接可规定为 DOS 路径, 例如: "G:\VORBEREIMPF_DAT" 或者规定为由节点命名符、共享名称和目录组成的网络地址, 例如: "\\R111\<共享名称>MPF_DAT"。

示例:

MD9677 CMM_DIRECTORY_SOFTKEY_PATH2=F:\R4711

; 驱动器 + 目录

MD9678 CMM_DIRECTORY_SOFTKEY_PATH3=\\R4712\TEIL1\WELLE1.DIR

; 计算机名称, 带有目录数据的共享名称

MD9679 CMM_DIRECTORY_SOFTKEY_PATH4=\\R4713\Teil2\MPF_DAT .DIR ; 计算机名称, 带目录数据的共享名称

在机床数据 9676 CMM_DIRECTORY_SOFTKEY_PATH1 下待输入的网络连接预留用于 SINUMERIK ShopMill 或 ShopTurn。

设计软键文本

为了选择网络驱动器连接, 软键标签可以作为与语言相关的用户报警文本, 以编号 89901 - 89905 存放在某报告模块识别到的用户文件 (见 MBDDE.INI [TextFiles] 段) 中。此时, 文本编号 89901 与机床数据 9676 CMM_DIRECTORY_SOFTKEY_PATH1 等相关。

示例:

MBDDE.INI:

...

[Textfiles]

USER_PLC=F:\DH\MB.DIR\MYTXT_

MYTXT_GR.COM:

89901 0 0 ""

备用

89902 0 0 "F:\R4711"

软键 2 的软键文本

89903 0 0 "R4712/ MMC1"	软键 3 的软键文本
89904 0 0 "R4713/ MMC2"	软键 4 的软键文本
...	(2 个空字符表示换行)

操作区中的可用性

在下列操作区中提供网络驱动器:

- 在扩展的 SK (软键) 条上“程序一览”软键后的 MASCHINE (加工) 操作区中,
- 在 PROGRAMM (程序) 操作区作为基本菜单的扩展软键条。
- 在 DIENSTE (通讯) 操作区中的数据管理中 (如果设计 3 个网络驱动器, 则显示 3 个软键作为水平软键, 第 4 个在按下 ETC 键后显示)
- 在 DIENSTE (通讯) 操作区中用第二个垂直软键读写网络驱动器数据...

扩展软键中的软键根据机床数据 9676 (SK1) 至机床数据 (SK4) 占用 (也即如果有空位)。在软键上出现已设计的文本。如果没有为软键设计标签文本, 则显示相应的机床数据值。

分配驱动器/计算机的访问权限

在下列显示机床数据中为每个驱动器设计访问权限:

对于 程序 和 通讯 操作区:

\$MM_USER_CLASS_DIRECTORY1_P (9510)

\$MM_USER_CLASS_DIRECTORY2_P (9511)

\$MM_USER_CLASS_DIRECTORY3_P (9512)

\$MM_USER_CLASS_DIRECTORY4_P (9513)

对于 机床 操作区:

\$MM_USER_CLASS_DIRECTORY1_M (9516)

\$MM_USER_CLASS_DIRECTORY2_M (9517)

\$MM_USER_CLASS_DIRECTORY3_M (9518)

\$MM_USER_CLASS_DIRECTORY4_M (9519)

用于访问权限的显示机床数据立即生效, 即在切换至各个应用程序后, 在目录中进行更改时重新启动后才生效。

故障信息提示

在启动 HMI 高级时评估设计的网络驱动器。在评估时不检查连接是否真的存在。仅在操作时选择驱动器的情况下才进行此项检查。如果无法访问设计的路径, 则生成故障信息提示。

参见

MBDDE.INI (页 315)

2.3.3 连接逻辑驱动器

前提条件

软键“逻辑驱动器”在访问级别 2（服务）以上可供使用：

因此机床操作者不能使用该功能，而维修技术人员可以使用。为了更新软键的分配，需要重新启动 HMI 高级。

使用该软键可以设置对 PCU 和 TCU 的 USB 接口上的 USB 存储介质的存取以及在程序、通讯或诊断操作区中访问网络驱动器（选项）。

功能

可以将控制系统与外部网络驱动器或其它计算机相连接。例如，通过此连接，可以由其它计算机处理零件程序。

为此，有两种可选方案：

- 在设置画面“逻辑驱动器”下的“开机调试”——“NC”中。
- 通过设计文件 LOGDRIVE.INI。

逻辑驱动器的配置由下列数据组成：

- 带有下列结构的路径名：

“acttcu://<TCU 插头>,<分区>/<路径说明>”

对于 USB 向一个 TCU 输入名称时，由一个识别所需插头的接口数组和一个分区编号

- 和一个软键标签（有关报警文本中与语言无关或语言相关的软键标签）。

软键“本地 USB”

供货状态：使用“本地 USB”预占用连接 1。

软键“本地 USB”总是提供在加工操作区、程序操作区和通讯操作区中，且无需设计。由此，显示激活的 TCU 的前面接口或者已设计的 PCU 接口。

说明

如果已设计 OEM 软键（水平软键 7），则在程序中最多显示 7 个网络驱动器。忽略最后的输入项。

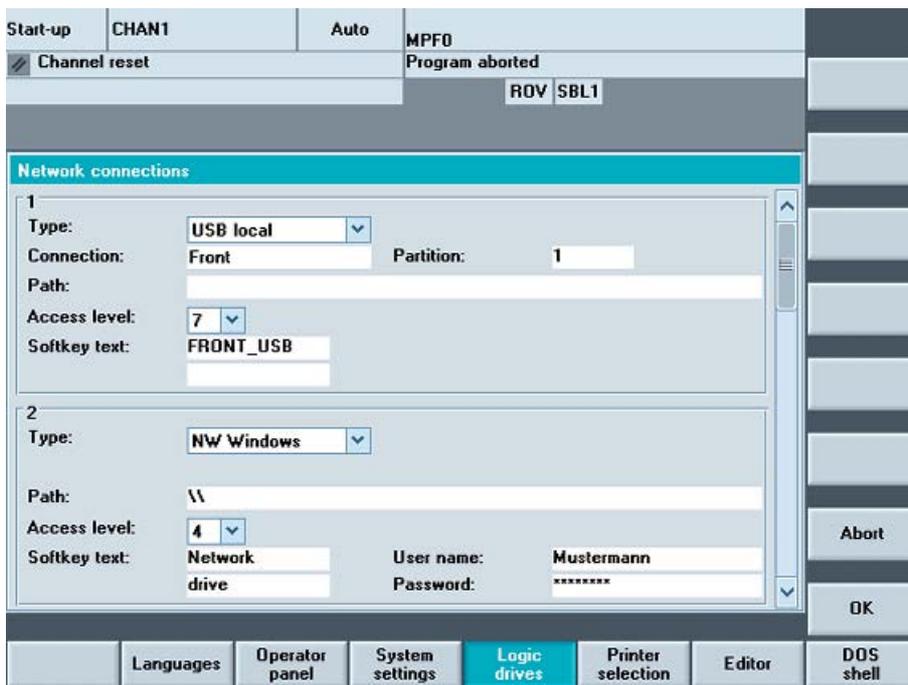


图 2-9 设置画面“逻辑驱动器”

连接类型/接口

可选择以下连接：

无驱动器：	不显示其他栏
本地 USB：	本地 USB 接口名称： - Front (0、2 或者 1:0.2 或者 FRONT) - X203 (1 或者 1:1 或者 X203) - X204 (2 或者 1:2 或者 X204)
分区： (对于本地和全局 USB)	比如 2
USB 全局：	全局 USB 接口名称： - Front (0、2 或者 1:0.2 或者 FRONT) - X203 (1 或者 1:1 或者 X203) - X204 (2 或者 1:2 或者 X204)
设备：(仅对于全局 USB)	比如 tcu
NW Windows：	网络驱动器名称(Windows 中的)。 路径以// oder \\ 开始。
用户名： (仅对于 NW Windows)	在屏幕上输入用户名，比如：Mustermann。
密码： (仅对于 NW Windows)	密码会显示在屏幕上，并加密输入到文件 logdrive.ini 中。
本地驱动器：	驱动器名称，比如：D:

访问权限

可将连接的访问权限等级设置为等级 0 到 7。如果未输入任何项，则视为预设（7 = 访问所有）

只可以为所有操作区规定一个访问级别；

- 机床
- 程序
- 通讯

软键标签

有两行字符—每行最多允许 10 个字符—可供一个软键的标签使用。\\n、%n 或者两个空格可用于分行。

报警文本 89901- 89906 可用于软键标签，从而使得在没有定义其他标签时，保存与语言相关的软键文本。

从 TCU 传输数据和传输数据至 TCU

总是访问刚刚激活的 TCU 的 USB 接口，即有操作核心的接口。如果在数据传输过程中切换激活的 TCU，则数据传输在后台中总是保持激活。

如果在切换 TCU 时有一项操作（复制、删除、编辑）在驱动器上有效，则显示保持到动作结束时。在连接时才更新显示。

说明

由于 TCU 已通过网络连接与 PCU 连接，则回答时间与每个网络相关。不建议处理 USB 存储器的零件程序 (EXTCALL)。

在写入数据至驱动器或自该驱动器读出数据时断开 USB 驱动器可能会造成数据丢失。

选件“网络驱动器”

下面的概览显示，何时需要选件网络驱动器：

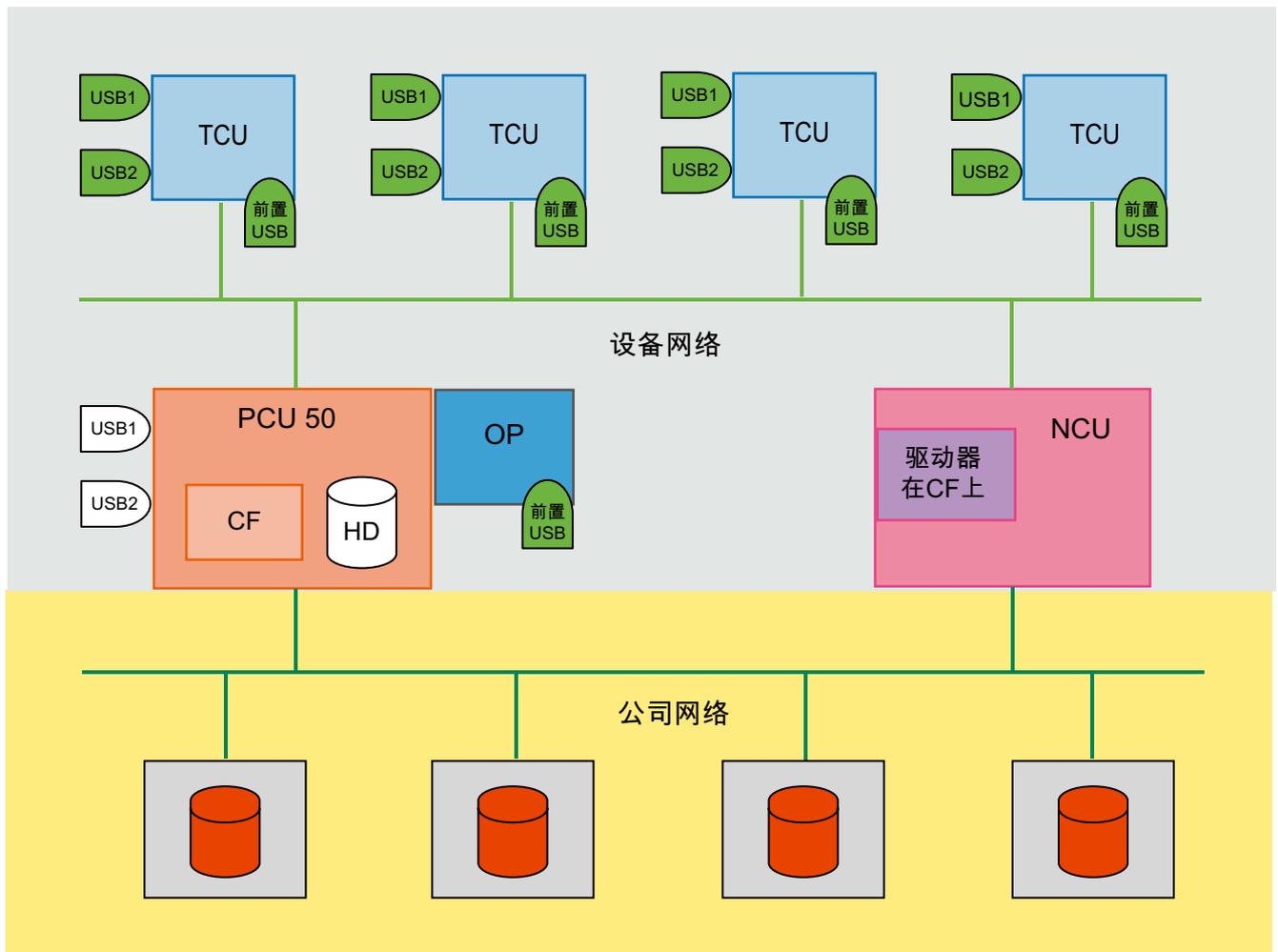


图 2-10 选件 网络驱动器

图例:

- 没有选件时可使用: TCU 上所有 USB 接口
- 没有选件时可使用: 本地可使用
- 本地没有选件时可使用, “远程” 仅在在有选件网络驱动器时可使用
- 仅在在有选件网络驱动器时可使用
- 仅在在有选件 HMI 用户存储器时可用, 本地可以使用

2.4 优化模拟

概览

通过模拟可以模拟屏幕上的处理过程。

由模拟评估的数据必须在对话编程/模拟数据（\DP.DIR\SIM.DIR）下的数据维护中作为副本存储。

模拟引导启动后，会在软键“数据匹配”下提供下列功能：

- 匹配装备数据
- 匹配刀具
- 匹配机床数据
- 匹配循环

2.4.1 显示模拟文件

用户程序

模拟中所装载的文件会在软键 **数据匹配** → **装载的数据** 中显示。显示当前复制到模拟中的所有程序。这些程序在选择另一个工件时再次从模拟中删除。

通过补偿编辑器直接在 HMI 上或 NCK 中更改程序，同时将更改后的程序自动复制到模拟中。如果在外部修改这些程序（例如通过通讯），则在下一次模拟选择时识别更改并将相应的程序复制到模拟中。

如果在当前工件下有文件（后缀为 RPA、GUD、SEA、UFR、PRO、CEC、TOA、TMA）存在，则在进行工件选择时将这些文件装载至模拟中。这些数据在工件切换时保持不变，受时间印记监控控制。

如果在外部修改这些文件（例如通过通讯/程序），则在下一次模拟选择时识别更改并将相应的文件复制到模拟的被动文件系统中。

- **基准数据**

显示装载到模拟中的初始化数据（INITIAL.INI）、刀具数据（TO_INI.INI）和定义文件（*.DEF）。文件（CH(c)_xxx.INI、NC_xxx.INI 或者 AX_SEA.IN）在“装载的数据”下显示。

对这些文件进行时间戳监控。如果在外部修改这些文件（例如通过通讯），则在下一次模拟选择时识别更改并将相应的文件自动复制到模拟中。

- **标准循环**

显示所有装载到模拟中的标准循环：

然后再需要是将循环复制到模拟中。不会自动识别到循环的更改，而是必须通过软键“匹配循环”进行。

- **用户循环**

显示所有装载到模拟中的用户循环：

如果未事先装载循环，则必须在文件 DPCUSCYC.COM 中定义循环接口。然后再需要是将循环复制到模拟中。不会自动识别到循环的更改，而是必须通过软键“匹配循环”进行。

- **制造商循环**

显示所有装载到模拟中的制造商循环：

如果未事先装载循环，则必须在文件 DPCUSCYC.COM 中定义循环接口。然后再需要是将循环复制到模拟中。不会自动识别到循环的更改，而是必须通过软键“匹配循环”进行。

2.4.2 模拟的数据匹配

概览

由模拟评估的数据必须存放在数据维护中的 \DP.DIR\SIM.DIR 下。每次启动模拟时读入这些数据。所有数据仅在重新启动模拟后激活！

另外使用 NC 对数据管理目录中的定义文件进行匹配。如果定义文件在数据管理目录中未通过 NC 匹配，则这可通过 DPSIM.INI 中的设置实现：

[MAIN]

MATCH_NCU_ENVIRONMENT=0 ;只有文件 INITIAL.INI 由 NC 进行复制。

MATCH_NCU_ENVIRONMENT=1 ; (预设置)

说明

可以通过“通讯”操作区中的编辑器手动匹配数据，在错误更改时有无法再启动模拟的危险。对于这种情况，如果数据无法由 NC 读取，应长期保存文件。

在软键“数据匹配”下提供有下列功能：

匹配装备数据

提供下列数据选择，以将数据从激活的 NC 文件系统装载到模拟中：

- R 参数 (RPA)
- 用户数据 (GUD)
- 全局用户数据
- 零点偏移 (UFR)
- 全局零点偏移
- 设定数据 (SEA)
- 全局设定数据
- 轴专用的设定数据
- 保护区 (PRO)
- 下垂/角度 (CEC)

可在任意的组合中选择数据。通过软键进行通道选择（当有多个通道用于模拟时），预设时，提供当前的模拟通道。当 NCK 上已经有所选的通道时，仅提供与通道相关的数据。通过软键“确定”将所选的数据复制到数据维护 \DP.DIR\SIM.DIR 下的相应文件中并装载到模拟中。

在不出现询问的情况下覆盖现有的文件。

如果为所有通道选择数据，则删除所有所属的与通道相关的文件（如果有），总是设立或者覆盖与通道相关的文件。

在将数据装载到模拟中之前，为模拟进行一次“通道复位”。

如果在复制 NCK 时出错，则可以重复传输过程或者中断传输（对话：中断/重复）。在装载至模拟时出错或出现模拟报警时，中断传输。

通过软键“确定”存储当前的数据选择并在下一次显示对话框时再次显示该数据选择。

匹配刀具

将用于所有通道的 NC 刀具数据复制到文件 TO_INI.INI 中，随后激活模拟。事先清除已有的刀具。如果 NC 中的刀具数据已改变，则在每次选择模拟时自动提供该功能。

匹配机床数据

将所有在 NC 中已激活的定义文件（UMAC.DEF、SMAC.DEF、MMAC.DEF、UGUD.DEF、SGUD.DEF、MGUD.DEF、GUD4.DEF – GUD9.DEF）复制到相应的 HMI 文件中。自 HMI 中删除 NCK 中未激活的文件。删除所有文件（CH(c)_xxx.INI, AX_SEA.INI, NC_XXX.INI），刀具数据例外。

由文件 INITIAL.INI 编译模拟机床数据。建议设置下列机床数据：

MD 11210: UPLOAD_MD_CHANGES_ONLY = 0

匹配循环

循环不受模拟的更改检查控制。

在需要时循环自动复制到模拟中（PRELOAD/CYCLES=0）或者在启动模拟时完全复制到模拟（PRELOAD/CYCLES=1）中。

如果循环已经更改，则必须通过软键“匹配循环”将循环复制到模拟中。仅将针对最近一次的访问而更改的循环复制到模拟中（同样还有新载入的循环）。

将模拟中现有的在其间已删除的循环自模拟删除。如果未使用自动的循环界面，则必须重新启动模拟。

镜像刀具

对于相同的机床，可以使用不同定向的刀架（例如在不同的 NC 通道中），这种情况下 TOA 长度补偿参数受到机床特定的框架转换（例如镜像），并且以此与原来的标准相反给定。

为了能够支持图形刀具定向时的位置情况，要将模拟图形模块 DPSMASCH.DLL 等在旋转版本中通过 TOA 参数的进一步全面计算来进行扩展。

Z.B. \$TC_DP3[n,m]=... 也可以根据协议使用负值，标准的正值长度尺寸一般用于“车削中心后侧”的加工，而负值长度尺寸则用于“车削中心前侧”的加工。

可能在相应的用户循环中需要另外特定的 NC 指令用于刀架选择，以确保通过修正的图形刀具定向的模拟过程，例如：

...

IF \$P_SIM

\$SC_MIRROR_TOOL_LENGTH=1 ; 刀具长度补偿

对于 SIM 共同镜像已镜像的轴

ENDIF

...

操作（铣削和车削）：

通过符号权限的评估，用于图形模块接口上的 TOA 参数不需要进行任何操作。

旋转坐标系

可以控制模拟视图的坐标系定向（在车削规格中以前不是太完整）并可以匹配例如立式车床。

为了这个目的，可修改 INI 条目 [VIS] PLANE_VIEW=..（in ..user\dpmwp.ini 用于铣削类型，或者 dptwp.ini 用于车削类型）。

为了能够使这些方式在与标准定向有偏差的机床上得以充分利用，需要打开记录项的值域（B/F: back/front、U/D: up/down、L/R: 左/右）车削时的相应功能（包括导出的视图：外部、半截面视图和全断面视图、外表面）。

操作界面的记录条目（车削时）可以部分地直接访问：

a[VIS]

...

;用于标准铣削默认平面视图：BDL(BackDownLeft)

;可能的 BDL、BUL、BUR、BDR、FDL、FUL、FUR 或 FDR PLANE_VIEW=BDL

或者

[VIS]

...

;用于标准卧式车床的默认平面视图：BDR(BackDownRight)

;对于立式车床，用 +Z 向上：BDL 或用 +Z 向下：FUL (** 自软件版本 5.3 开始 **)

;可能的 BDL、BUL、BUR、BDR、FDL、FUL、FUR 或 FDR

PLANE_VIEW=BDR

说明

目前其它诸如 BDR、BDL 或 FUL 的条目在车削时无意义。

操作（仅用于车削）：

为了支持立式车床定向，目前的表格“设置主轴”以一个附加的框架组“机床定向”通过前三个选项控制进行扩展：

“水平标准”（默认：PLANE_VIEW=BDR），

“垂直，向下进刀”（PLANE_VIEW=BDL）和

“垂直，向上进刀”（PLANE_VIEW=BDL）。

在这里，用软键“更改标准”也可以接受相应的匹配作为用于所有新工件的永久参考。

优化

- **输入监控“展开直径外表面”**

在车削规格中，根据毛坯部件输入，参数“外直径”和“展开直径外壳面积”检测稳定性，可能自动进行匹配（例如在缩小外直径时），否则指出错误输入（例如在展开直径情况下的空输入栏）。

- **刀具中心轨迹（“无刀具数据”）**

在随后的刀具数据源选择时进行临时保存（例如“使用刀具”）。

为了能够以图形方式检查刀具补偿（LK、FRK、SRK）的选择和取消选择是否正确，在取消选择刀具补偿时的刀具中点轨迹和选择刀具补偿时的刀具轨迹可以在视觉上是相互对立的，这一点将很有帮助。

为此，一般保留由模拟过程“不通过刀具数据”产生的刀具中点轨迹，在接下来第一次模拟过程“通过刀具数据”情况下（为 SIM 位置图形叠加带或不带刀具补偿的刀具轨迹）。然后，和以前一样，在每次重新启动模拟时复位队列中的刀具轨迹。

2.4.3 加速模拟引导启动

模拟启动

模拟软件包根据标准在应答操作员请求时才启动。由此，仅在需要时通过待装载的软件组件加载系统。

同时在第一次调用时必须延长模拟的启动时间。启动时间就是自选择模拟开始到第一次执行 NC 命令时的时间（“可视的”结果）。

在 HMI 启动时装载

为了加速模拟启动，可以延长 HMI 启动时的启动时间。在文件 REGIE.INI 中进行所需的设置（目录 OEM 或 USER）。

```
[StartupConfiguration]
```

```
Startup12 = name:=dpncsim, Timeout:=0
```

```
[TaskConfiguration]
```

```
Task27 = name := dpsim, Timeout := 30000
```

说明

在 [StartupConfiguration] 段中使用的条目尚不允许用于其它条目。为此，检查目录 MMC2、ADD_ON、OEM 和 USER 中的文件 REGIE.INI。

在 SimNCK 启动时装载

在启动 SimNCK 时装载用于 NC 环境的配置数据（刀具数据、循环）。为此，必须将带其子程序名称的待处理程序装载到模拟 NCK 中。这一启动特性通过文件 DPSIM.INI 的 [PRELOAD] 和 [MAIN] 控制。

可以通过下列设置缩短模拟的启动时间：

```
[PRELOAD]
```

```
CYCLES=1
```

在启动模拟 NCK DPNC SIM.EXE 时预装载所有循环（自 NCK 和硬盘）。

```
TOOLS=1
```

在启动模拟 NCK 时预装载（如果有）用于刀具图的文件（TO_INI.INI 和 TO_ADDON.INI）。

2.4.4 扩展几何刀具数据

显示刀具

之前只能显示带有刀沿、轴和刀架的刀具，这些可以从 NCK 中数据计算得出。

通过扩展几何刀具数据可以在模拟画面中仅显示刀沿（“切割刀沿”）而不显示轴和刀架。此外，可以将附加的偏差参数载入到刀具特定的模拟中。

刀具数据的定义在文件 **SIMTOGEO.INI** 中进行，该文件保存在 \USER、\OEM、\ADD_ON 其中的一个目录内或者在 \HMI_ADV 中。供货状态下该文件未设立。

其它前提条件：

- 在 DPTWP.INI 或 DPMWP.INI 中激活功能“切割刀沿”
- 机床数据：定义 OEM 参数存储器
- 创建文件 SIMTOGEO.INI，将链接记录在文件中

参数设置

在文件 Datei SimToGeo.ini 中必须在段[设置]内存在有一个记录条目，该条目用于与 NCK 刀具数据建立关系。

[Settings]

Link=TC_DPCx 和 x = 1... 10 或者

Link=TC_DPCSx 和 x = 1... 10

建议将参数 TC_DPCS1 用作新项目的链接条目，因为该参数特为该目的而预留。当参数 TC_DPCS1 已用于一个已有的项目中时，仅应该使用另一个链接设置。

举例

在例如 Link=TC_DPCS1 情况下，必须在 NCK 刀具数据中规定，对于每个刀具（针对每个刀具在 SimToGeo.ini 中已有条目），条目 \$TC_DPCS1[t, d]= <Identifier>，t = 刀具号码，d = Duplo 号码。标识符必须作为实数记录，例如 500.1，同时小数点前的数字可以表示刀具类型，小数点后的数字可以表示其它的区别。

SimToGeo.ini 中所属的值位于相应的标识符下：

[500.1]; 粗加工刀具 1

tool_type = 500

...

用于铣刀定位的可选参数

-因此，刀具在 3 个主层面 XY、YZ 或 ZX 中描述。 - 如果不能描述一个刀具，则显示一个多标记。

仅允许这些条目中的一个：

TC_DPV3= ; 1 = +X, -1 = -X

TC_DPV4= ; 1 = +Y, -1 = -Y

TC_DPV5= ; 1 = +Z, -1 = -Z

不能输入一个空的空间刀具定向。

在文件 SimToGeo.ini 中，可以在 tool_type 下定义与 NC 规定有偏差的刀具。如果可以设置相应的刀沿参数（例如 \$TC_DPCS1[t,d]），那么也可以使用 SimToGeo.ini 来连接或者计算几何数据。

一个完整的模板文件，带有各个许可刀具类型的记录条目请参见章节“SIMTOGEO.INI”

设置机床数据

MD 18080: MM_TOOL_MANAGEMENT_MASK='H4'	; 存储器预留用于 OEM 数据
MD 18204: MM_TYPE_CCS_TOA_PARAM[0]=4	; 数据类型 4 = 实数
MD 18206: MM_NUM_CCS_TOA_PARAM=1	; 每个刀沿的 OEM 数据数为 0 ..10。

开机调试时的步骤：

1. 根据车削/铣削工艺在 DPTWP.INI 或 DPMWP.INI 中激活功能“切割刀沿”：

[RUEST]

;*** 仅软件版本 6.4 或以上 ***

;仅将刀具显示为 cutting edge “切割刀沿”（开/关）

TOOL_AS_CUTTING_EDGE=ON ;default = OFF

2. 定义用于 OEM 参数的机床数据或者存储器：

\$MN_MM_TOOL_MANAGEMENT_MASK=4 ; 用于 SimToGeo.ini 位 2=1 的设置

如果已经预设了机床数据，则必须另外设置该位并定义 SIEMENS-OEM 参数 \$TC_DPCS1 的存储器和类型 \$MN_MM_NUM_CCS_TOA_PARAM=1 ; TC_DPCS1 保留。

\$MN_MM_TYPE_CCS_TOA_PARAM[0]=4 ; 用于 TC_DPCS1 的浮点数据类型

或者

定义用户 OEM 参数 \$TC_DPC1 的存储器和类型

\$MN_MM_NUM_CC_TOA_PARAM=1 ;TC_DPC1 保留

\$MN_MM_TYPE_CC_TOA_PARAM[0]=4 ; 用于 TC_DPC1 的浮点数据类型

3. 设立 SimToGeo.ini:

定义用于西门子或用户的 OEM 参数记录条目并对几何数据进行扩展。这通过已设立的 OEM 变量（\$TC_DPCS1/\$TC_DPC1）实现。该变量值确定要使用哪些几何程序段。

例如，刀具 1 刀沿 1 的条目（T1 D1）\$TC_DPCS1[1,1]=500.1 参考文件 SimToGeo.ini 中的几何数据块 [500.1]。多个刀具可以参考相同的几何数据块。

4. 添加扩展的几何刀具数据：这些会出现在程序段中，可以通过例如标识符、浮点类型加以区分。

例如 [500.1]。

[Settings]

Link=TC_DPCS1 ; 连接 TO_INI.INI, 西门子 OEM

;Link=TC_DPC1 ; 连接 TO_INI.INI, 用户 OEM

[500.1];T1 D1 粗加工刀具 Typ500 SL3 L1=100, L2=55, R1

tool_type=500 ; 则 \$TC_DP1 不需要刀具类型

insert_length=10 ; TC_DP8, 托盘长度

holder_angle=105 ;TC_DP10, 刀架角度, 不是前角

reference_direction=3 ;TC_DP11, 切削位置

clearance_angle=5 ;TC_DP24, 后角

5. 连接刀具数据/TO_INI.INI:

CHANDATA(1)

\$TC_DPCS1[1,1]=500.1 ; 西门子 OEM T1 D1

粗加工刀具 SL3 类型 500

\$TC_DPC1[1,1]=500.1 ; 用户 OEM

\$TC_DP1[1,1]=500

\$TC_DP2[1,1]=3

\$TC_DP3[1,1]=100

\$TC_DP4[1,1]=55

2.4.5 优化所需存储容量

功能

为了避免模拟在另有存储器需求情况下在 NC 机床数据中进行存储器设置，可以在文件 DPSIMNCK.INI 中的程序段 [DP_SIMNCK] 内进行这些设置。

对于模拟，在文件 DPSIMNCK.INI 中的程序段 [DP_SIMNCK] 内记录下标准值，一般情况下在 NC 中不需再进行附加的存储器设置。**这些偏移值叠加影响同名的机床数据。**

文件 DPSIMNCK.INI 中的记录项

输入时不考虑负值。

带标准值的下列条目包含在段 [DP_SIMNCK] 中：

...

[DP_SIMNCK]

DP_SIMNCK_MM_NUM_GUD_MODULES_OFFSET = 0

使用 MD 18118:MM_NUM_GUD_MODULES

DP_SIMNCK_MM_NUM_GUD_NAMES_NCK_OFFSET = 5

使用 MD 18120:MM_NUM_GUD_NAMES_NCK

DP_SIMNCK_MM_NUM_GUD_NAMES_CHAN_OFFSET = 30

使用 MD 18130:MM_NUM_GUD_NAMES_CHAN

DP_SIMNCK_MM_GUD_VALUES_MEM_OFFSET = 20

使用 MD 18150 MM_GUD_VALUES_MEM

DP_SIMNCK_MM_NUM_USER_MACROS_OFFSET = 10

MD 18160:MM_NUM_USER_MACROS

DP_SIMNCK_MM_NUM_MAX_FUNC_NAMES_OFFSET = 40

使用 MD 18170 MM_NUM_MAX_FUNC_NAMES

DP_SIMNCK_MM_NUM_MAX_FUNC_PARAM_OFFSET = 350

使用 MD 18180 MM_NUM_MAX_FUNC_PARAM

DP_SIMNCK_MM_MAX_SIZE_OF_LUD_VALUE_OFFSET = 400

使用 MD 18242: MM_MAX_SIZE_OF_LUD_VALUE

...

2.5 建立用户报警

2.5.1 用户专用报警的结构

报警的划分

为循环报警、编译循环报警和 PLC 报警提供了下列报警编号：

编号范围	名称	结果	删除
60000 - 60999	循环报警（西门子）	显示，NC 启动禁用	复位
61000 - 61999		显示、NC 启动禁用、处理预编码程序段后的运动停止	复位
62000 - 62999		显示	取消
63000 - 64999	预留		
65000 - 65999	循环报警（用户）	显示，NC 启动禁用	复位
66000 - 66999		显示、NC 启动禁用、处理预编码程序段后的运动停止	复位
67000 - 67999		显示	取消
68000 - 69000	预留		
70000 - 79999	编译循环报警		
400000 - 499999	一般 PLC 报警		
500000 - 599999	通道 PLC 报警		
600000 - 699999	轴和主轴 PLC 报警		
700000 - 799999	用户 PLC 报警		
800000 - 899999	过程链/图解 PLC 报警		

用于循环报警文本的文本文件格式

循环报警和编译循环报警的文本文件结构如下：

报警号	显示	帮助 ID	文本或报警编号
60100	1	0	“未编程 D 编号%1”
60101	1	0	60100
...
65202	0	1	“轴 %2 在通道 %1 中不存在”
//德语版循环报警文件			

表中所列的序号范围中，不是每个编号都可用。

文献： 基本功能手册：PLC 基本程序 (P3 sl)

支持的语言

表格 2-2 语言和所使用的 ANSI 表/代码页

语言	语言缩写	代码页 (DOS)	Ansi 表 (Windows)
丹麦语	DK	850	1252
德语	GR	850	1252
英语	UK	850	1252
芬兰语	FI	850	1252
法语	FR	850	1252
意大利语	IT	850	1252
荷兰语	NL	850	1252
葡萄牙语	PO	850	1252
瑞典语	SW	850	1252
西班牙语	SP	850	1252
波兰语	PL	852	1250
捷克语	CZ	852	1250
匈牙利语	HU	852	1250
俄语	RU	855	1251
土耳其语	TR	857	1254
日语	JA	932	-
中文 (简体)	CH	936	-
韩语	KO	949	-
中文 (繁体)	TW	950	-

ANSI 表和字体之间的布局

表格 2-3 Ansi 表 1250 (中欧)

字体	文件名	字体名称
Arial	Cearial.ttf	Arial CE (True Type)
Arial bold	Ceariabd.ttf	Arial CE Bold (True Type)
Arial italic	Ceariali.ttf	Arial CE Italic (True Type)
Arial bold italic	Caeriabi.ttf	Arial CE Bold Italic (True Type)

表格 2-4 Ansi 表 1251 (西里尔)

字体	文件名	字体名称
Arial	Aricyr.ttf	Arial Cyr (True Type)
Arial bold	Aricyrb.ttf	Arial Cyr Bold (True Type)
Arial italic	Aricyri.ttf	Arial Cyr Italic (True Type)
Arial bold italic	Aricyrbi.ttf	Arial Cyr Bold Italic (True Type)

表格 2-5 Ansi 表 1252 (西欧)

字体	文件名	字体名称
Arial	Windows-Standart	Arial (True Type)
Arial bold	Windows-Standart	Arial Bold (True Type)
Arial italic	Windows-Standart	Arial Italic (True Type)
Arial bold italic	Windows-Standart	Arial Bold Italic (True Type)

表格 2-6 Ansi 表 1254 (土耳其)

字体	文件名	字体名称
Arial	Trar.ttf	Turkish Arial (True Type)
Arial bold	Trarbd.ttf	Turkish Arial Bold (True Type)
Arial italic	Trari.ttf	Turkish Arial Italic (True Type)
Arial bold italic	Trarbi.ttf	Turkish Arial Bold Italic (True Type)

安装语言

必须首先安装不属于标准供货范围内的语言。对于亚洲语言，必须使用辅助工具（例如 NJWIN）。此时自动安装相应的字体。

安装每个语言包时，用工具卸载相应的字体并在 MMC.INI 和目录中进行所有所需的输入。

开机调试时，将 DLL、文本文件（机床数据文件）和语言相关的 INI 文件存放在目录“\mmc2\language”中，将报警文本存放在目录“\dh\mb.dir”中。

语言

如果要以另一种语言设计的报警文本作为控制系统中已存在的语言，可以设计一个软件“语言选择”。用户报警文本的语言分配通过文本文件名称实现。为此，为在 MBDDE.INI 中输入的用户文件名称添加相应的缩写和扩展名 .com。

关于报警的在线帮助

建立用户专用的在线帮助在调试手册 在线帮助/HE1/ 下有描述。

参见

设计软键‘语言选择’ (页 27)

2.5.2 生成用户专用的报警文本

同时提供的文件

带有 ASCII 格式报警文本的文件被保存在 PCU 硬盘上的下列路径中：

MMC	F:\dh\mb.dir\alm_XX.com
NCK	F:\dh\mb.dir\aln_XX.com
PLC	F:\dh\mb.dir\alp_XX.com
ZYK	F:\dh\mb.dir\alz_XX.com
CZYK	F:\dh\mb.dir\alc_XX.com

“X”表示相应语言的缩写符号。用户 **不得更改** 这些 **标准文件**，而为了接收自己的报警文本。

说明

为此，在软件更新时不要覆盖的文件 MBDDE.INI，必须将该文件保存在专门指定的路径中：..\user\mbdde.ini

建立用户报警

用户可以用自己的文本代替同时提供的报警文本，或在其中添加新的内容。可以在操作区“开机调试”→“HMI”→“编辑”中使用所提供的编辑器。

为此，必须通过“通讯”操作区在目录 **f:\dh\mb.dir**（MBDDE 报警文本）中建立附加文件。用户文件中的文本覆盖带相同报警编号的标准文本。补充标准文本中没有的报警编号。

编辑器

处理时可以使用任意的 **ASCII 编辑器**（例如 DOS 编辑器）。用另一个编辑器处理文本文件时必须注意，之后以 **ASCII 格式** 存储这些文件。

说明

对报警文本的修改只有在 **HMI 重新启动**后才会生效。建立文本文件时必须注意，要在 **HMI 系统**上正确设置日期和时间。否则，可能无法在屏幕上显示用户文本。

激活报警文本文件

可以通过文件 MBDDE.INI 中段 [Textfiles] 的下列记录项激活由用户创建的报警文本：

...

[Textfiles]

MMC=F:\dh\mb.dir\alm_

```
NCK=F:\dh\mb.dir\aln_  
PLC=F:\dh\mb.dir\plc_  
ZYK=F:\dh\mb.dir\alz_  
CZYK=F:\dh\mb.dir\alc_  
UserMMC=  
UserNCK=  
UserPLC=  
UserZyk=  
UserCZyk=  
...
```

例如：对于文件 MBDDE.INI 中的两个附加文件（PLC 报警文本，修改的 NCK 报警文本）：

```
...  
User MMC =  
User NCK = F:\dh\mb.dir\myncck_  
User PLC = F:\dh\mb.dir\myplc_  
User ZYK =  
User CZYK =  
...
```

报警文本的语言相关性

用户报警文本的语言分配通过文本文件名称实现。为此，为在 mbdde.ini 中输入的用户文件名称添加相应的缩写和扩展名 .com。语言缩写参见章节“支持的语言”（表格 2-2）。

如果外部建立报警和信息文本，则可通过列出的代码页，ANSI 表和字体建立 (表 2-3 至 2-6)：

报警文本和信息文本包含在 MBDDE 报警文本目录中。

1. 旧文本文件用 Windows 编辑器（WinWord, Notepad 等等）转换为 ANSI 文本文件（“存储在.....下”）
2. 将新文本文件和转换后的旧文本文件装载并汇编到 Windows 编辑器中。
3. 存储新建的文本文件，并将其添加到目录“MBDDE 报警文本”中。

参见

MBDDE.INI (页 315)

机床数据参数化

3.1 通过 HMI 使用机床数据

概览

可以根据制造商和用户的要求进行专门设置：

- 通过 INI 文件配置 HMI 操作界面
- 通过机床数据对设置控制系统的参数
- 编程 PLC 功能



危险

机床数据更改对机床会有重大影响。有错误的参数化可能危及人员生命，并导致机床损坏。

前提条件

机床数据区上的存取权限可以通过钥匙开关或者口令禁用。

机床数据的设置

机床数据分配在下列范围中：

- 一般机床数据（\$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）
- 驱动专用机床数据(\$M_)
- 控制单元机床数据:驱动参数 (solution line)
- 供电机床数据:驱动参数 (solution line)

每个范围存在一个自身的列表画面，在该画面中可以显示和更改机床数据。

从左向右显示机床数据的下列特性：

- 机床数据编号
- 机床数据名称，可能带有数组索引。
- 机床数据值
- 机床数据单位

- 有效性

机床数据的效力

在右栏中显示什么时候机床数据有效：

so = 立即生效

cf = 通过按下软键“机床数据有效设置”

re = 复位

po = POWER ON (NCK 上电复位)

机床数据的物理单位

机床数据的物理单位显示在输入区的右侧：

显示	单位	测量值
m/s**2	m/s ²	加速度
U/s**3	U/s ³	修改用于旋转轴的加速度
kg/m**2	kgm ²	惯性矩
mH	mH (毫亨利) :	电感
Nm	Nm (牛顿米) :	转矩
us	μs (微秒) :	时间
uA	μA (微安) :	电流强度
uVs	μVs (微伏特秒) :	电磁流
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。

3.1.1 调整用于机床数据的显示过滤器

功能

“通用机床数据”显示选项下提供的显示过滤器用于，有目的地降低所显示机床数据的数量。因此全部机床数据都会分成特定的显示组，例如配置数据。此时适用：

- 每个有自身组分配的区域
- 符合过滤器字中一个位的每个组（以前保留）。
- 每个区域最大有 13 组（组 14 为专家参数保留，位 15 为扩展保留）。
- 对于显示机床数据，已实现分组。

显示过滤器激活

如果显示过滤器激活，仅显示设置的显示组。用“所有其它”显示没有设置过滤器位的机床数据。软键性能如同一般机床数据的显示选项。保存设置。

举例

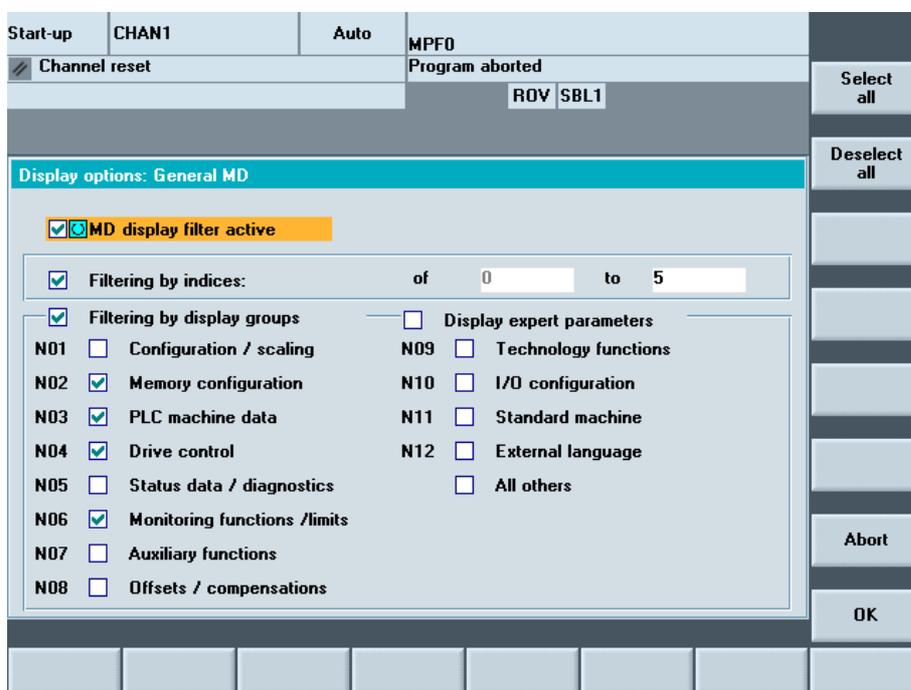


图 3-1 用于机床数据的显示过滤器

初始化

在打开一个机床数据窗口时，自动更新与范围相适合的过滤器设置。

该过滤器只用于有目的地根据使用区域作为“显示选项”显示特定的显示机床数据。此时可以根据附属性分组机床数据。

过滤器标准

下列表格指出按评估顺序显示机床数据时的标准：

标准	检查
1. 访问权限	如果访问权限不足，不显示机床数据。 否则检查标准 2。
2. 显示过滤器激活	如果过滤器未激活，总是显示机床数据。 否则检查标准 3。
3. 专家参数	如果设置专家参数位且未选择专家参数，则不显示机床数据。 否则检查标准 4。
4. 显示分组	如果不仅设置至少一个组位，而且也在显示过滤器中选择，则检查标准 6。 否则检查标准 5。
5. 所有其它	如果没有设置组位，且也在显示过滤器中选择“所有其它”，则检查标准 6。 如果没有设置组位，且在显示过滤器中未选择“所有其它”，则不显示机床数据。
6. 索引从...到...	如果已选择索引检查且在选择的范围中存在数组索引，则显示机床数据。 如果已选择索引检查且在选择的区域中不存在数组索引，则不显示机床数据。 如果未选择索引检查，显示机床数据。

3.1.2 创建用户视图

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。

机床数据概览

在用户视图中可以接收下列机床数据：

- | | |
|--------------------------------------|--|
| • 一般机床数据 | 10 000 - 19 800 |
| • 通道专用机床数据 | 20 000 - 29 000 |
| • 轴专用机床数据 | 30 100 - 38 010 |
| • 通用设定数据 | 41 010 - 41 700 |
| • 通道专用设定数据 | 42 000 - 42 990 |
| • 轴专用的设定数据 | 43 100 - 43 920 |
| • 通用显示机床数据 | 9000 - 9999 |
| • 通道专用的显示机床数据 | 9033, 9450, 9451, 9610
9639, 9650, 9672, 9673 |
| • 用于 SIMODRIVE 驱动的机床数据（powerline） | 1000 -1799 |
| • 用于 SINAMICS 驱动、控制元件、E/A 模块和供电的机床数据 | |

举例

创建用户视图 TEST01：

1. 选择“新视图”。会打开一个空白窗口，用于添加机床数据和注释行。
2. 如要添加注释，可以选择“添加文本…”并输入文本名称，例如“text zeile 1”。按下 <输入>键对文本进行确认。
3. 如要添加机床数据，可以选择“添加文本…”。选择机床数据区，例如“轴专用机床数据”。会在列表中显示出相应的编号范围 30 000 - 38 000：
4. 选定机床数据 32 000。为了让该机床数据能够被收入用户视图中，要按下软键“在行前/后添加”进行确认。
5. 可以用同样的方式添加其他机床数据，或者可以使用“查找”：在编号范围内可以使用“查找”直接选定并添加机床数据。
6. 如要结束选择，可以按下软键“<<”退回到“视图编辑”窗口。

结果:

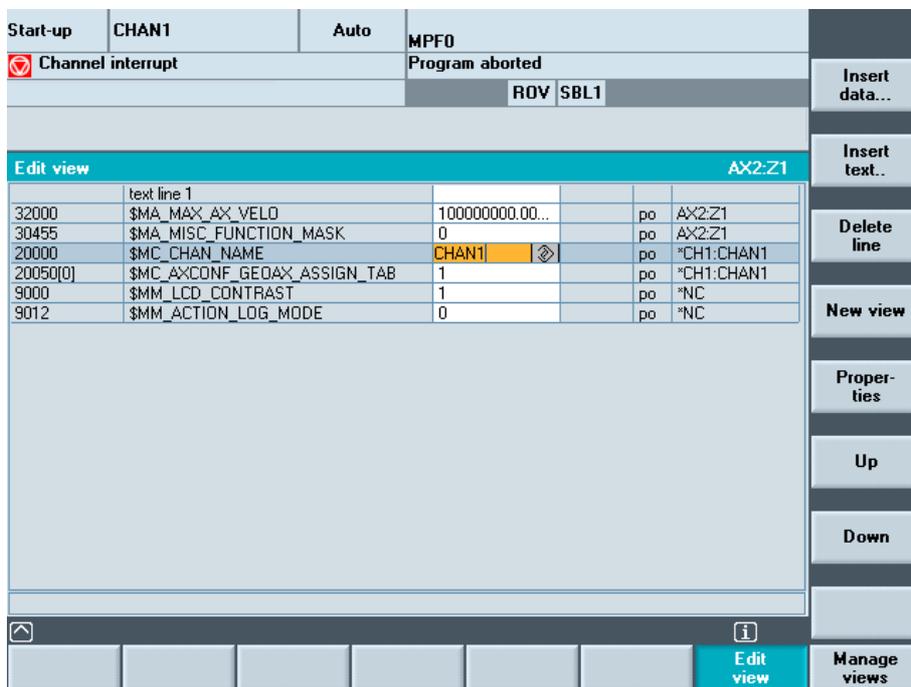


图 3-2 示例：用户视图

编辑用户视图时可以使用下列软键：

1. 使用软键“向上”或“向下”可以进行行的上下移动，相应的也可以使用“删除行”将指定行删除。
2. 选择“属性”，可以输入机床数据的说明文本或对现有的文本进行修改。说明文本会显示在下方的信息行中，例如“通道名称”。
3. 选择“视图管理”，可以存储用户视图。输入一个名称，例如 **Test01**，并按下 <输入>键进行确认。
4. 如要显示用户视图，可以选择“整理软键”并输入文本用作软键名称。



图 3-3 为直接选择分配软键

说明

可以使用 6 个软键来保存用户视图。

在“轴”栏中可以输入轴号码或者从一个列表中选择相应的轴。

在用户视图中轴号码或轴名称一直使用 “*”标记，直至对其分配作出修改。

3.2 通过机床数据对设置控制系统的参数

3.2.1 精确零点偏移和基本偏移

功能

功能与画面“可设定的零点偏移”和“基准零点偏移”中的偏移有关。除了用于输入偏移的栏外，还可插入用于微小偏移的其它栏。检查输入时的显示机床数据。

MD 9203:USER_CLASS_WRITE_FINE	保护用访问等级
MD 9451:MM_WRITE_ZOA_FINE_LIMIT	变化量

实际值显示：可设定的零点系统

MD 9424:MA_COORDINATE_SYSTEM 可以设定如何显示实际值：

- 定位工件坐标系 (= 编程的位置，按照标准设置) 或者
- 已激活刀具的接受位置与工件零点相关。

MD 9424 = 0	在工件坐标系 WCS 中显示 (预设置)
MD 9424 = 1	在可设定零点系统 ENS 中的显示 (有效刀具的安装位置)

示例：

程序	WCS 显示	ENS 显示
N110 X100	100	100
N110 X100	0	0
N120 X0	0	0
N130 \$P_PFRAME=CTRANS(X,10)	100	110
N140 X100		
N150 ...		

说明

实际值显示的设置：参见机床制造商说明

3.2.2 用于设定实际值，对刀，PRESET（预设）的机床数据

功能

对刀与设定实际值，PRESET（预设）功能的特性会受到多个机床数据的影响。所选的设置也对操作起作用。（显示软键和数值，存储输入的数值）。

通过引入系统框架可以为这些功能提供两种类型。类型通过一个通道特定的机床数据加以区分：

MD 28082: \$MC_SYSTEM_FRAME_MASK

位:	0	设定实际值，对刀
	1	外部零点偏移
	2	TCARR, PAROT
	3	TOROT, TOFRAME
	4	工件基准点
	5	循环
	6	转换

类型 1，不带系统框架：

没有机床数据或用于对刀和设定实际值的系统框架位 0 未设置。

类型 2，带系统框架：

在机床数据中已设置用于对刀和设定实际值的系统框架位 0。

显示机床数据 MD 9422: \$MM_MA_PRESET_MODE 在工作方式 JOG 下确定功能 PRESET（预设）/基准偏移。

0: 无软键

1: 在加工操作区（前提条件）中的先前 PRESET（预设）

2: 设定实际值

3: 设定实际值（参见在线帮助）

说明

若要隐藏软键 PRESET（预设）、设定实际值和对刀，也可设置 MD 9220: \$MM_USER_CLASS_PRESET。

文献： HMI 高级操作手册

设定实际值

- **用系统框架设定实际值：**
G500 有效：将值写入至系统框架中。
G500 无效：输出一个故障信息。
- **不用系统框架设定实际值：**
G500 有效（复位可设定的框架）根据条目将值写入第一个基准框架中，显示 MD 9245：
\$MM_MA_PRESET_FRAMEIDX = 基准框架索引。
G500 无效：输出一个故障信息。

对刀

选择后提供一个有效的框架。通过在栏“零点偏移”中的输入项可以选择另一个框架。

- **用系统框架对刀：**
如果是 G500，在栏“零点偏移”中将值写入至系统框架中。
如果不是 G500，在栏“零点偏移”中将值写入至那里已存在的框架中。
除了位于主轴中的刀具外，不可选择任何其它刀具。如果主轴中无刀具，则可以自由选择刀具。
- **不用系统框架对刀：**
如果是 G500，在栏“零点偏移”中将值写入至第一个系统框架中。
根据条目将值写入至设定的基准框架中，显示 MD 9245：
\$MM_MA_PRESET_FRAMEIDX = 基准框架索引。
如果不是 G500，在栏“零点偏移”中将值写入至那里已存在的框架中。刀具可以自由选择。

3.2.3 使有效的刀具补偿立即生效

功能

利用 MD 9440:ACTIVATE_SEL_USER_DATA 可以确定刀具补偿的更改立即生效，即使是在 NC 通道不处于“复位状态”或“停止状态”中时。该功能总是有效。

 小心
通过零件程序中下一个编程的轴运动在 NC 启动复位后重新进行补偿。

文献： 基本功能功能手册 轴、坐标系、框架 (K2)

3.2.4 确定 STAT 和 TU 的值描述

特殊运动和机器人操作时的特性

在工件坐标中进行笛卡儿点到点运行 (PTP) 时, 各个位置上的加工轴位置有多种含义。若要明确说明该特性, 则为此类转换 (例如 5 轴转换) 在工件坐标外部还要描述机床 STAT 位置和轴 TU 位置。

设置基数

STAT 和 TU 值在轴画面 (实际值窗口、带进给率和缩放显示的宽实际值窗口) 中显示并在示教时输入到 MDA 编辑器中。STAT 和 TU 值是否以二进制、十进制或十六进制显示, 这些要通过显示机床数据设置

MD 9242:MA_STAT_DISPLAY_BASE	用于显示支轴位置 STAT 的基数
MD 9243:MA_TU_DISPLAY_BASE	用于显示回转轴位置 TU 的基数

可能的值是:

02	显示为二进制值
10	显示为十进制值
16	显示为十六进制值

这些设置既在实际值窗口上生效, 也在编辑器窗口上生效。

举例

轴自 X120 Y20 Z-50 运动至 X200 Y200 Z-120, 在 NC 程序中以二进制方式显示时如下所示:

X=120 Y=20 Z=-50 STAT='B010' TU'B011101'

X=200 Y200 Z=-120 STAT'B110' TU='B011111'

3.2.5 DRAM 内循环的存储器位置

概述

循环可存储在 NCU 的 DRAM 中。对于用户其优点在于，有更多的 SRAM 存储器可用于零件程序和工件。

功能

操作界面 HMI 高级在 "调试"→"NC"→"NC 存储器" 下对于具有相应权限的用户可以提供确定 DRAM 的使用和通过 MD 18351: MM_DRAM_FILE_MEM_SIZE 匹配其大小。更改 DRAM 存储器会导致 NCK 存储器完全重组。

注意

在更改存储容量前必须保存所有相关的数据。更改机床数据会导致 NCK 存储器的重组。

循环存储在 DRAM 中

MD 11290: DRAM_FILESYSTEM_MASK 可以设置，将哪些对象载入 DRAM：

位 0-n	= 0:	各个目录中的数据应保存在 SRAM 中。
	= 1:	各个目录中的数据应保存在 DRAM 中。这取决于 MD 11291: DRAM_FILESYST_SAVE_MASK 在 NC 卡上的 FFS 上存有一个备份拷贝。
位 0		CST 目录（西门子循环）
位 1		CMA 目录（机床制造商循环）
位 2		CUS 目录（用户循环）

在闪存文件系统(FFS)中备份

MD 11291: DRAM_FILESYST_SAVE_MASK 可以设置，将 DRAM 中哪些对象备份到 FFS 上：

位 0-n	= 0:	没有进行备份。在关闭控制系统时，NCK 数据丢失。
	= 1:	DRAM 中现存的数据备份在 NC 卡的 FFS 中。
位 0		CST 目录（西门子循环）
位 1		CMA 目录（机床制造商循环）
位 2		CUS 目录（用户循环）

在 FFS 中备份后，无需在引导启动时重新载入对象。

注意

不允许直接更改存储在 DRAM 中并在 FFS 中备份的对象。仅允许规定在 FFS 中进行备份的循环。必须在装载入 HMI 高级前执行有可能需要的更改。

如果直接更改保存在 DRAM 中、但没有在 FFS 中备份的对象，断电后更改丢失。

3.2.6 重新定义用于 NC 数据的访问等级

功能

预设的访问等级可以在定义文件中改变。对于机床数据，仅允许比设计的存取登记低的访问等级，对于设置数据，也可以高于设计的访问等级。文件通过读入下一个 _N_INITIAL_INI 生效。读写（零件程序或 PLC）规定不同的访问等级。

举例：

```
_N_DEF_DIR/_N_SGUD_DEF           ; 用于全局变量的文件
                                   ; $PATH=/_N_DEF_DIR
REDEF $MA_CTRLLOUT_SEGMENT_NR APR 7 APW 4 MD 30100
                                   ; (APR ...读取权限)
REDEF $MA_ENC_SEGMENT_NR APR 2 APW 2 MD 30210
                                   ; (APW ...写入权限)
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 2 APW 2
M30                               SD 41050
```

MD 30100: 机床数据 10000 有保护级 2 / 7，即需要需要访问等级 2（等于口令）用于写，需要保护级 7 用于读。

若要达到机床数据范围，则需要最少为钥匙开关位置 3。

取消对访问等级的更改

若要再次取消对访问等级的更改，必须再次重新写入原始值。

示例：

```
_N_DEF_DIR/_N_SGUD_DEF           ; 用于全局变量的文件
                                   ; $PATH=/_N_DEF_DIR
REDEF $MA_CTRLLOUT_SEGMENT_NR APR 7 APW 2
                                   ; (APR ...读取权限)
REDEF $MA_ENC_SEGMENT_NR APR 0 APW 0
                                   ; (APW ...写入权限)
REDEF $SN_JOG_CONT_MODE_LEVELTRIGGRD APR 7 APW 7
M30
```

用于 REDEF 的定义文件

与 GUD 定义类似，在系统启动时处理的、自身的定义文件供 REDEF 指令编程使用。

```
/_N_DEF_DIR/_N_UACCESS_DEF // 终端用户访问等级说明定义文件
/_N_DEF_DIR/_N_MACCESS_DEF // 制造商访问等级说明定义文件
/_N_DEF_DIR/_N_SACCESS_DEF // 西门子系统应用访问等级说明定义文件 (例如标准循环, ShopMill, ...)
```

其他用系统配置处理的西门子定义文件保存在 NC 系统软件的 PCMCIA 卡或者 CF 卡上。

说明

兼容性

如果访问机床数据（这已通过指令 "REDEF <机床数据> APR <访问等级> APW <访问等级>" 许可）通过报警 15420 拒绝，则如下进行：

在激活功能或者升级系统时，REDEF 指令必须从现有的 GUD 定义文件传输至新的定义文件 _N_SACCESS_DEF、_N_MACCESS_DEF、_N_UACCESS_DEF 中。

NC 语言指令的访问等级

确定的 NC 语言指令可以与访问等级链接，以限制权限人员进行：

如果没有相应的执行权限，报警 14018 会中断零件程序的处理。当前执行权限的预设和控制系统上有效的访问权限一致，即钥匙开关位置 0 至 3 或者西门子最终用户的口令。

文献： 程序手册，章节 "NC 语言指令访问等级"。

3.2.7 修改软键‘基本零点偏移’上的存取权限

操作区“参数”

通过 MD 9247: 可以设置 \$MMP_USER_CLASS_BASE_ZERO_OFF_PA，即从哪一个访问权限等级开始，在“参数”操作区的“零点偏移”窗口中提供有软键“基本零偏”。同时在“零点偏移”窗口和“有效零点偏移 + 补偿”窗口中也显示或隐藏基准框架。

操作区域“加工”

通过 MD 9248: \$MMP_USER_CLASS_BASE_ZERO_OFF_MA 可以设定，自哪个访问等级起，在机床操作区的“对刀”功能中提供有“基准零点偏移”或在零点偏移栏中能够使用记录条目 G500。

3.2.8 隐藏轴

功能

通过通道专用机床数据 MD 20098: DISPLAY_AXIS 可以在机床操作区中将轴隐藏。默认显示所有轴。

在加工操作区中的下列窗口中可以隐藏轴:

- 小的轴窗口 (5 根轴, 机床默认设置)
- 宽的轴窗口 (5 根轴, 带进给率和倍率)
- 大的轴窗口 (8 根轴, 大字体)

此时, 为机床或工件坐标系中的视图分别设定, 是否显示或不显示轴:

位 0 至 15: 显示 (=1) /隐藏 (=0) 几何轴 (WCS)

位 16 至 31: 显示 (=1) /隐藏 (=0) 加工轴 (MCS)

回参考点运行和安全集成 (特殊情况)

机床数据 20098: DISPLAY_AXIS 在显示回参考点运行方式时, 也就是在显示轴参考点并确认安全位置时不使用。总是显示所有加工轴。

如果没有相应的访问权限, 操作员无法更改基准零点偏移的值。

将访问权限 7 作为标准值输入至机床数据 9247 和 9248 中。

机床数据参数化

3.2 通过机床数据对设置控制系统的参数

20098	DISPLAY_AXIS		
机床数据号码	在 HMI 上显示或隐藏轴		
标准预设置: 0xFFFFFFFF	最小输入界限: 0	最大输入界限: 0xFFFFFFFF	
在上电后更改生效	防护等级: 2/7	单元: -	
数据类型: DWORD	有效自软件版本: 4.4		
意义:	标识, 决定是否将轴显示为加工轴、几何轴或辅助轴。		
位 0 至 15: WCS	位 0	= 1 = 0	在实际值窗口中显示几何轴 在实际值窗口中隐藏几何轴
	位 1	= 1 = 0	在参考点窗口中显示几何轴 在参考点窗口中隐藏几何轴
	位 2	= 1 = 0	在预设/基准偏移/对刀窗口中显示几何轴 在预设/基准偏移/对刀窗口中隐藏几何轴
	位 3	= 1 = 0	在手轮选择窗口中显示几何轴 在手轮选择窗口中隐藏几何轴
位 16 至 31: MCS	位 16	= 1 = 0	在实际值窗口中显示加工轴 在实际值窗口中隐藏加工轴
	位 17		未占用
	位 18	= 1 = 0	在基准偏移窗口中显示加工轴 在基准偏移窗口中隐藏加工轴
	位 19	= 1 = 0	在手轮选择窗口中显示加工轴 在手轮选择窗口中隐藏加工轴

3.2.9 主轴载荷的显示

功能

到目前为止，加工基本画面中主轴载荷显示区域的显示为 100%。可以通过显示 MD 9429: MA_SPIND_POWER_RANGE 将显示主轴载荷栏的显示区域设置为 200%。

9428	\$MM_MA_SPIND_MAX_POWER		
机床数据号码	主轴功率显示的最大值		
标准预设置: 100	最小输入界限: 100	最大输入值: ****	
生效方式: 上电	防护等级: 3/4	单元: %	
数据类型: WORD			有效自软件版本: 6.4
意义	在该机床数据中输入一个系数，用该系数乘以所提供的主轴载荷。		

9429	\$MM_MA_SPIND_POWER_RANGE		
机床数据号码	主轴载荷显示区域		
标准预设置: 100	最小输入界限: 100	最大输入值: ****	
生效方式: 上电	防护等级:3/4	单元: %	
数据类型: WORD			有效自软件版本: 6.4
意义	<p>在该机床数据中确定用于显示主轴载荷的栏的显示区域。根据输入的值更改显示的百分数和颜色区域的大小。</p> <p>值 100: 显示百分数 0、80%和 100%。如果同时也在显示机床数据 9428 中设置值 100，则显示的颜色自 80% 起从绿变为红。</p> <p>值 > 100, 例如 200: 显示百分数 0、100%和 200%。显示的颜色自 100% 起从绿变为红。</p> <p>如果将值设为 200，则必须在显示机床数据 \$MM_MA_SPIND_MAX_POWER 中输入标准化系数。</p>		

计算标准化系数

通过用功能发生器、伺服轨迹功能和电机数据表中的数据测量主轴来计算标准化系数。

函数发生器

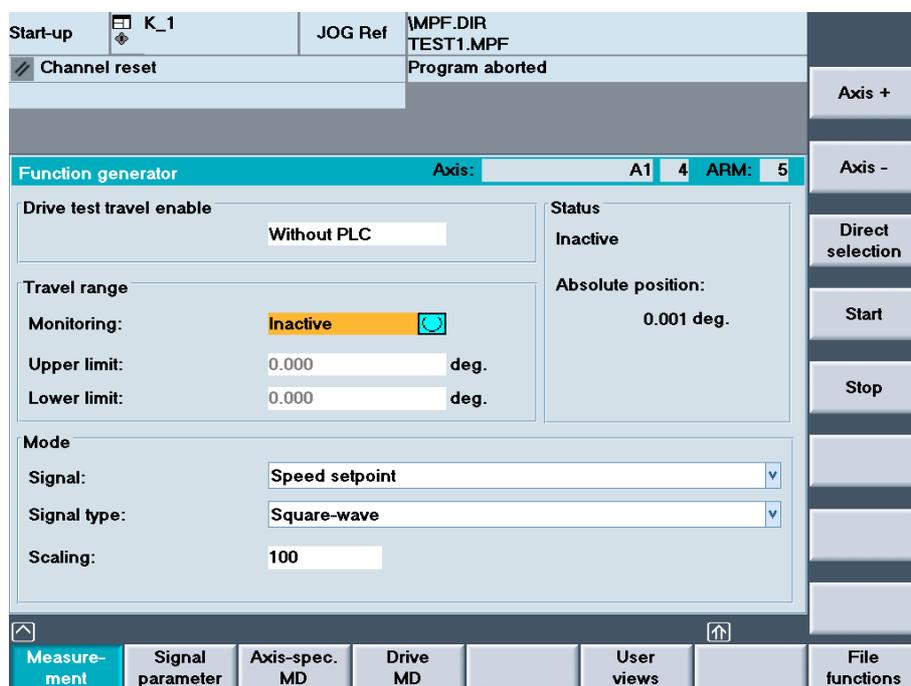


图 3-4 函数发生器

1. 在操作区“开机调试”→“优化/测试”→“功能发生器”中打开下列窗口：
2. 在隐藏的屏幕窗口中的“工作方式”下选择信号、信号类型和标准化系数。
3. 之后，按下“信号参数”软键。出现“功能发生器参数”屏幕窗口。
4. 输入用于振幅、周期时间、脉冲宽度和限制的设置。有关这些设置的数据，请参阅电机数据表。

注意
如果在 **虚拟轴** 上使用功能发生器和测量功能，则会由 NCK 引起中断。

伺服轨迹功能

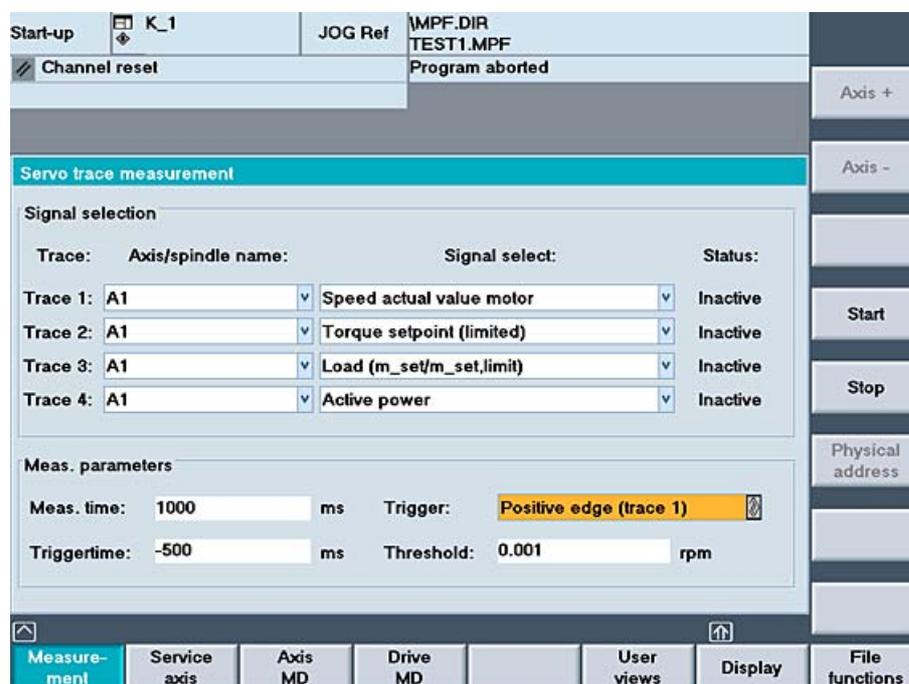


图 3-5 伺服轨迹：基本画面

1. 打开伺服轨迹功能基本画面，在该画面的“开机调试”操作区中按下软键“驱动/伺服”，随后再按下“伺服轨迹”。
2. 在“信号选择”区通过下拉列表选择待测量的尺寸。有关测量持续时间和触发时间的设置，请参阅电机数据表。进行参数化后，通过按下软键“开始”启动测量。
3. 选择恒定功率或者恒定力矩区，以用这些数据计算出标准化系数。

示例：在图中的测量功率为 10.5 kW。

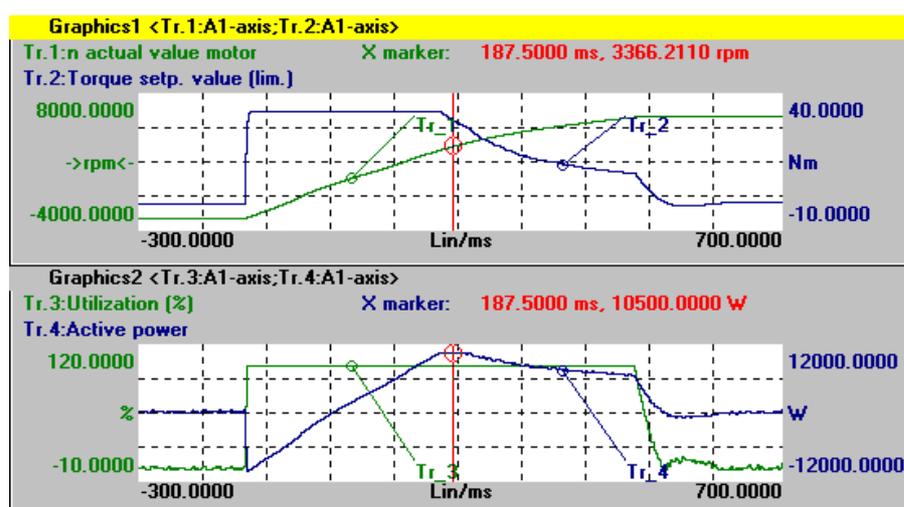


图 3-6 伺服轨迹：分析

3.2 通过机床数据对设置控制系统的参数

如果载荷显示在额定数据 100% 和经过额定点运行时显示相应较高的值，则可设置的标准化系数由电机数据或者图表和测量如下计算：

$$MA_SPIND_MAX_POWER = \text{最大_测量_功率} / \text{额定功率}_{S1} * 100$$

$$MA_SPIND_MAX_POWER = 10500W / 7000W * 100 = 150$$

在机床数据中输入系数 150。

3.2.10 主轴显示的主轴符号反向

功能

主轴符号的显示可以通过通道特定的机床数据 MD 9033:

MA_DISPL_INVERT_DIR_SPIND_M3 反向：可以正常显示或反向显示。例如，如对于副主轴的 M4，要显示与主主轴的 M3 相同的符号，应设置位 [n]=1（在接口上参照 M3/M4 反向）。

9033	MA_DISPL_INVERT_DIR_SPIND_M3	
机床数据号码	显示主轴旋转方向	
缺省值: 0x0000	最小输入值: 0x0000	最大输入值: 0x7FFFFFFF
更改生效: 立即	防护等级 (L/S): 3/4	单元: -
数据类型: LONG	有效自软件版本: 6.2	
意义	显示主轴旋转方向: Bit[n]=0 在符号中将 M3 显示为右旋 Bit[n]=1 在符号中将 M3 显示为左旋 n = 主轴号 -1 (即位 0 是主轴 S1) 注释: 对于相应设置的机床数据 9014: USE_CHANNEL_DISPLAY_DATA 可以在 NC 中管理通道特定的机床数据 9033。	

3.3 创建 PLC 机床数据明码文本

概览

对于 PLC 机床数据，可以按照特征[索引]来设计专用的与语言相关的明码文本。

可以在下列目录中建立名称为 **NCTEA_xx.TXT** (xx =语言缩写 GR、UK, ...) 的文本文件：

user\language,
add_on\language

oem\language

条目以下列格式：

<条目编号> ; 记录条目的数量

<MD number>[<Index>] <Text> ; 编号 [索引] 文本

举例

14510[i] 用户数据 (INT)	i = 0 ... 255
14514[i] 用户数据 (FLOAT)	i = 0 ... 31
14516[i] 用户数据 (HEX)	i = 0 ... 255

文件 NCTEA_GR.TXT 中的记录条目：

3

14510[0] , 我的集成文本, 用于 PLC-MD 14510.0

14514[0], 用于浮点数据 14514.0 的文本

14516[2] 用于 MD14516.2 十六进制文本

显示的结果：

14510[0] , 我的集成文本, 用于 PLC-MD 14510.0

14514[0], 用于浮点数据 14514.0 的文本

14516[2] 用于 MD14516.2 十六进制文本

说明

如同报警文本那样建立并处理 PLC 操作信息。

文本文件的语言缩写参照列表：“语言和所使用的 ANSI 表/代码页”。

参见

用户专用报警的结构 (页 85)

编程 PLC 功能

4.1 激活 PLC 和 NCK 之间的数据传输

功能

通过 PLC 可以用下列特性启动自/到 NCK 的数据传输。

- 借助**工作列表**由硬盘将数据传输至 NCK 或自 NCK 传输数据（工件目录，零件程序...）。
- 在 HMI 高级中“工作列表”功能用于处理零件程序，例如装载、卸载、选择或由硬盘处理。

通过 DB19 中 PLC 和 HMI 之间的接口，PLC 为 HMI 给出下面的任务：即触发 HMI（PCU 的硬盘）和 NCK 之间的数据传输。

规定用户控制文件： DB19.DBB16 (PLC → HMI)

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
总是 = 1	用户控制文件的 PLC 索引；值： 1 - 127						

规定工作列表： DB19.DBB17 (PLC → HMI)

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
用户控制文件中的 PLC 行偏移；值： 1 - 255							

PLC 的任务位： DB19.DBB13 (PLC → HMI)

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
选择	装载	卸载					

用于数据传输当前状态的 HMI 确认位：
DB19.DBB26 (HMI → PLC)

位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
选择	装载	卸载		激活	故障	OK	复位 PLC

注意
机床制造商 请注意机床制造商的设计说明。

文献:

- HMI 高级操作手册
- 基本功能功能手册：不同的 NC/PLC 接口信号与功能 (A2)

任务流程

PLC 任务根据下表进行:

- 如果确认位为 0，则 PLC 仅允许初始化任务字节中的任务。
- 将确认位中的 HMI 任务镜像至参数程序段上（PLC 由此识别到要处理的是其任务）。未完成的任务作为“已激活”报告至 PLC。
- 如果动作已结束（正常或出错），则 PLC 必须再次进行响应并删除任务位。
- HMI 接着将确认位设置为 0。用这种方式可以保证流程按顺序进行。

用户控制文件

待执行工作列表的名称存放在 HMI（PCU 的硬盘）上显示机床数据（/BD.DIR）目录中的控制文件 PLC_IN_OUT_xxx.TEA 中，同时 001 和 127 之间的编号为“xxx”。文件最多包含 127 个工作列表参考，控制文件由用户自行建立。

PLC 规定一个确定控制文件并以工作列表名称存储的 DB19.DBB16 索引（1-127）：

- **举例：用户控制文件 PLC_IN_OUT_003.TEA**
N1 /WKS.DIR/OTTO.WPD/OTTO.JOB
N2 /WKS.DIR/WELLE.WPD/KANAL1.JOB
N3 ...
N4 ...
行编号 N.. 可以省略。文件可以编辑。
- **举例：装载一个工作列表**
 - 现有文件 PLC_IN_OUT_003.TEA 带有下列记录条目：
N1 工作列表 A
N2 工作列表 B
N3 工作列表 C
 - 确认位 DB19.DBB26=0
 - 确定用户控制文件的 PLC 索引（DB 19.DBB16=131）
 - 确定 PLC 行偏移 2（DB 19.DBB17=2）
 - PLC 任务：装载（DB 19.DBB13.6=1）
 - 装载工作列表 B

- HMI 报告至 PLC: 工作列表已装载 (DB19.DBB26.6=1)

说明

当“机床”操作区已不包括在 HMI 的引导启动中时, 则该功能不可用。

故障识别至 PLC

下列故障识别由 HMI 通过 DB19.DBB27 输出至 PLC:

值	意义
0	无故障
1	控制文件的编号无效。(DB19.DBB16 中的值 < 127 或无效)
2	DB19.DBB15 无法读取
3	未找到控制文件 /BD.DIR/PLC_IN_OUT_xxx.TEA。(DB19.DBB16 中的值无效)
4	控制文件中的索引无效 (DB19.DBB17 中的值错误。)
5	选择的工作列表在控制文件中无法打开。
6	工作列表中有错误 (工作列表编译器报告有错)
7	工作列表编译器报告任务列表为空
9	执行工作列表时出错

解决冲突

手动执行 (通过操作) 工作列表时, 在记录中可能会出现询问, 例如当 MPF 文件既在 NC 上又在硬盘上且两者又各不相同。

如果预设置为“yes”, 即所有询问都用是回答, 如果为“no”, 则所有询问都用否回答。如果用于回答的预设置为“yes”, 即所有询问都用是回答, 如果为“no”, 则所有询问都用否回答。在文件 MASCHINE.INI 中可以设置对于询问的回答:

```
[PLC_REQUEST]
```

```
DefaultAnswer=yes
```

多个操作面板/NCUs M:N (仅用于 SINUMERIK powerline)

在一个 m:n 设备中将 HMI 高级分配到 NETNAMES.INI 中的一个 Netmaster-NCU/PLC 中。NCU 的总线地址在 [param network] 段下设计。

例如: [param network]

```
netmaster=NCU_LINKS
```

通过确定 Netmaster-NCU/PLC 仅激活带符号名称 NCU_LINKS 的 NCU/PLC 的数据传输。

4.2 将 HMI 的当前任务编号传输至 PLC

任务编号

任务标出位置，在该位置上在 HMI 区域菜单（即最上层）中挂入操作区。

软键栏

在区域菜单的软键条和任务之间有下列固定关系：

1. 软键条：

软键	1	2	3	4	5	6	7	8
任务编号	0	1	2	3	4	5	6	7

2. 软键条：

软键	1	2	3	4	5	6	7	8
任务编号	8	9	10	11	12	13	14	15

3. 软键条：

软键	1	2	3	4	5	6	7	8
任务编号	16	17	18	19	20	21	22	23

在文件 REGIE.INI 中，在段 [TaskConfiguration] 内设计哪些任务号码分配那些应用。

对于 OEM 应用程序，可以有第 4 个软键条：

软键	1	2	3	4	5	6	7	8
任务编号	96	97	98	99	100	101	102	103

任务编号 24 – 95 已预留用于标准应用程序的子任务以及指定的 OEM 任务。所属的标签文本可输入到与语言相关的文件 re_xx.ini 中。

应用

对于由制造商设计的操作界面，机床控制面板和所显示的操作区/画面上的按下按键操作的正确触发是必须的。

示例：

在一个操作屏幕窗口中选择“机组运动”功能。在 MSTT 上有“向前”和“向后”运行键，用这两个键可以实时运行机组。

若要避免，例如通过在 HMI 中进行操作区切换显示新的画面，在结束运行键运动之前，PLC 可以总是识别和检查当前任务编号。

设计

当前任务号码由 PLC 传输到数据模块范围内，数据模块范围在显示机床数据 9032: HMI_MONITOR 中可以如下作为字符串规定：

DBxx.DBByy 带有：

xx 数据模块编号

yy 传输区（字限）第 1 个字节的字节编号

传输区的长度 8 字节，在第 1 个字节里进行传输。字节 2-8 已预留。

说明

必须注意，规定的区域尚未由输入字节、输出字节或标记字节占用。模块编号和字节范围未检查。

更新

第一个字节中任务编号的更新在如下情况下进行：

- HMI 的每次任务切换
- 在 M:N 下切换与另一个 NC 的连接

对于持续时间未确定的状态（例如在 M:N 下切换至另一个 NC 时），将特殊值 255 传输为任务编号。

取值范围

任务编号可以包括由用户设计的操作区值 0-95。

由此显示所有可视的基本菜单任务（0 - 31）和不可视的子任务（32 - 95）。

4.3 通过 PLC 选择通道/主轴

通道选择

按照下列步骤激活通道选择或显示（在“通道名称”区域中显示）：

DB 19, DBB 32 = 01000001 [二进制] 通道选择

DB 19, DBB 33 = <Kanalnummer> 通道编号

用于下一通道的 FF

主轴选择

按照下列步骤激活主轴选择（在主轴窗口中显示）：

DB 19, DBB 32 = 01000010 [二进制] 主轴选择

DB 19, DBB 33 = <Spindelnummer> 要显示的主轴编号

故障标识

在 DB 19.DBB 36 中存储有下列故障标识，在 HMI 上出现 PLC 功能要求时可以给出有关故障的详细说明：

值	意义
0	无故障
1	无功能编号（DBB32 位 0-5）
2	参数无效
3	HMI 内部变量的写入异常中断
10	无通道（DBB33）

HMI ↔ PLC 之间的确认机制

通过 DB19.DBB32 位 6, 7 存在一个确保 HMI 和 PLC 在通道选择时相一致的确认机制：

PLC 将位 6 置为 1=“功能要求”。当位 6 和位 7 的值为 0 时，只允许由 PLC 进行功能要求的设置。

在标题中显示后，HMI 再次在位 6 中写入 0，许可用于下一 PLC 动作的接口。

说明

如果给一个 NCU/PLC 单元分配 2 个 HMI 单元（M:N），则 PLC 必须首先根据接口确定哪些 HMI 接口已被激活。可以由 PLC 要求通道/主轴显示仅用于激活的 HMI 单元。第二个 HMI 单元的通道/主轴显示选择用的接口位于 DB19, DBB82-86 中。描述的使用方法与 DBB32-36 相类似。

4.4 设计标题中的信息显示

功能

通过 PLC 可以进行设计（另参见章节：带符号的通道交叉状态显示），在“程序名称”栏内的工作方式 JOG 和 MDA 的标题中可以显示两行文本（例如所选的滑枕，激活的手轮）。

句法：

信息文本可以由下列各项组成：

- 一个与语言相关的部分（自报警/信息文本文件 *.COM）和
- 3 个参数（%1、%2、%3）

可以显示的最大文本长度为每行 30 个字符。

用户 PLC 数据模块

通过用户 PLC 数据模块设计显示。

带 DBBy 的 DBx 编号由用户在文件 HEADER.INI 中如下输入：

```
[UserTextInfo]
```

```
USER_TEXT_BASE=DBx.DBBy
```

使用 x 数据模块编号，y 模块中的初始字节。

规定该接口时，必须激活文本长度限制：

```
Textlength = 33
```

PLC 报警文本文件（例如“alpu_”）的名称由用户输入至“UserPLC”下 [TextFiles] 段中的文件 MBDDE.INI 中：

```
[Textfiles]
```

```
UserPLC=alpu_
```

用户数据模块与 PLC 的接口

标题中有 2 可以用于显示：

- 程序路径行
- 程序名称行

用户模块的接口提供有开始数据字节（由 HMI 进行任务识别）和后续字节，通过它们可以输出数量、轴的索引和文本（长度为 12 字符/特殊字符）。

如果在输入文本时将“00”写入一个字节，则 HMI 将视其作文本结束。该“00”之后的记录条目将不再显示。

4.4 设计标题中的信息显示

表格 4-1 用户 PLC 数据模块的接口

程序路径行	开始 DBBx	开始 DBWx +1	开始 DBBx + 3	开始 DBBx + 4	开始 DBBx +5-17
数据类型	字节	字	字节	字节	字符串
意义	位 0=1: 必须要做些什么 所有字节=1: 由 HMI 确认任务 所有字节=0: PLC 可以再次写入接口	来自 alpu_xx.com 用于显示文本的偏移, 700000 为基准地址并始终在内部进行加法运算 =0: 删除行	编号范围 0...255 (与第 1 个参数 %1 一致)	索引范围 -127...128 负编号 (-1=0xFF, -2=0xFE) 为加工轴索引, 正编号为通道轴索引。索引在显示之前转换为一个轴名称。 (与第 2 个参数 %2 一致)	字符串由最大 12 个字符组成 (与第 3 个参数 %3 一致)
程序名称行	开始 DBBx +20	开始 DBWx +21	开始 DBBx +23	开始 DBBx +24	开始 DBBx +25-37
数据类型	字节	字	字节	字节	字符串
意义	同上	同上	同上	同上	同上

说明

前面的作任务完成后所有的字节变为零并且用参数 DBBx+1 至开始 DBBx+5 为新任务开始进行补偿设置, 只有当新任务的位 0=1 时才使用开始 DBBx。

不允许在用户字符串的内部存在有功能句法内有特殊含义 (", %, \$, #) 的特殊字符 (开始 DBB+5....)。

开始 DBBx	HMI 识别一个任务并确认该任务 (所有字节 =0)。
开始 DBWx +1	开始 DBWx+1 包含用于带基本 700.000 的文本编号的偏移值 (用户 PLC 文本)。如果该参数 = 0, 则删除相应的行。
开始 DBBx +3	开始 DBBx+3 包含一个无需转换显示的数字。
开始 DBBx +4	开始 DBBx+4 包含一个轴索引。 负值 = 加工轴索引 (AXCONF_MACHAX_NAME_TAB[Index-1], MD 10000), 正值 = 通道轴索引 (AXCONF_CHANAX_NAME_TAB[Index-1], MD 20080)。 这些索引总是与刚刚在 HMI 中显示的通道相关。索引在显示之前转换为一个轴名称。
开始 DBB +5	开始 DBBx +5—17 包含一个最大长度为 12 字符的字符串。

设定 PLC 程序

设定 PLC 程序时, 需首先提供所有参数并设置第一个字节。

程序名称行 (第二行) 的数据位于开始字节后的 20 字节。

HMI ↔ PLC 之间的确认机制

有一个用于开始 DBBx 的确认机制，该机制确保 HMI 与 PLC 相一致：

PLC 将位 0 置为 1“显示标题中的用户文本”。当位 0 的值为 0 时，功能要求仅允许由 PLC 设置。

在标题中显示后，HMI 再次在位 0 中写入 0，许可用于下一 PLC 动作的接口。

M : N 时的动作（仅用于 SINUMEREIK powerline）

使用功能 M:N 时，必须在被动连接情况下删除两个文本行。与在区域切换类似，在主动连接情况下评估并显示变量内容。

任务：

在 JOG 和 MDA 运行方式下，应当会在标题“程序名称”区中文本第一行“手轮生效轴...”中输出相应的轴名称、并在文本第二行“滑枕...有效”中输出相应的滑枕编号。

步骤：

1. 将 PLC ↔ HMI 接口传输至文件 HEADER.INI 中，例如 DB60.DBB10
2. 将 PLC 用户文本文件的名称记录至 MBDDE.INI 中，例如：

```
[Textfiles]
```

```
UserPLC=alpu_
```

3. 将文本记录到 PLC 用户文本文件，例如：

```
700100 0 0 “手轮在轴 %2 上有效”
```

```
700101 0 0 “滑枕 %1 有效”
```

4. 自 PLC 出发激活文本显示并写入接口，例如对于第一个显示行：

```
DB60.DBW11 = 100
```

```
DB60.DBB14 = 2
```

```
DB60.DBB10 = 1
```

例如用于第二显示行：

```
DB60.DBW31 = 101
```

```
DB60.DBB33 = 1
```

```
DB60.DBB30 = 1
```

结果

在所采纳的轴配置中：

```
MD 20080 AXCONF_CHANAX_NAME_TAB[0] = "X"
```

```
MD 20080 AXCONF_CHANAX_NAME_TAB[1] = "Y"
```

```
MD 20080 AXCONF_CHANAX_NAME_TAB[2] = "Z"
```

需要接着进行下列显示：

“手轮在轴 Y 上有效”（第一显示行）

“滑枕 1 有效”（第二显示行）

4.5 利用多个通道启动程序段搜索

功能

在程序测试模式下进行该程序段查找过程时驱动 NCK，以让通道和同步动作之间或者多个通道之间以及在 NCK 内部可以交互作用。

在“程序测试中搜索程序段”时，NCK 会将所有的辅助功能输出至 PLC、并由 NCK 执行用于通道配置的零件程序指令（WAITE、WAITM、WAITMC、更换轴、变量说明），即 PLC 在搜索程序段时进入当前状况并且在程序搜索范围内对加工过程进行补偿处理，包括多个通道的共同工作。

激活程序段搜索

通过“轮廓程序测试”软键激活功能。NCK 接着启动程序段搜索并将当前通道中所选的程序切换到“程序测试”模式下：

- 零件程序的辅助功能由 NCK 输出到 PLC 上。
- 轴运动不执行。

如果 NC 在程序段搜索框架中查找到一个零件程序命令 WAITE、WAITC 或 WAITMC，则 NCK 等待各个规定的伙伴通道，而与该伙伴通道刚才在什么模式下无关（例如正常的程序处理、程序测试、通过程序测试进行搜索，等等）。

如果 NC 在当前的通道中到达规定的目标程序段，则 NC 停止程序段搜索并选择“程序测试”模式，目标程序段的帮助功能不再输出。

只要所有参与当前查找过程中的通道退出“程序测试”模式，就会输出提示信息要求进行确认。

该信息说明，在查找过程期间，可能已经根据零件程序在参与程序段查找的通道中建立了重定位（REPOS）偏移，该偏移在下一次 NC 启动后通过 NC 插补消除。在使用按键“NC 启动”继续在程序段搜索所到达的程序位置上进行程序处理之前，可以在 JOG 工作方式下按顺序手动消除可能出现在单个通道中的重定位偏移。

程序段搜索配置

“程序测试”模式下的程序段搜索支持下列配置：

1. 当前所选通道中的程序段搜索（当前所选通道是由 HMI 选择的通道）。
2. 在当前所选通道和所有通道中的程序段搜索，在这些通道中如同在当前通道中一样选择相同的工件。
3. 在当前所选通道和所有通道中的程序段搜索，这些通道位于相同的 BAG 中，如同当前通道。
4. 在 NCK 所有通道中的程序段搜索

配置 2 至 4： 在其它通道（与当前通道不同）中不进行程序段搜索，“程序测试”模式在其它通道中有效，直至识别到一个中止操作（例如 WAITMC）。

设置程序段配置

在文件 MASCHINE.INI 中激活不同的配置。

可以进行下列设置：

[BlockSearch]

SeruproEnabled=1

; Shows (SeruproEnabled=1) or hides (SeruproEnabled=0) the softkey used to do a block search in mode "programtest". The softkey appears in both dialogs "Blocksearch Searchposition" and "Blocksearch Searchpointer"

SeruproEnabled=0

=0	功能无效
=1	功能有效（预设置）

SeruproConfig=1

=1	当前所选通道中的程序段搜索（预设置）
=2	当前所选通道和带相同工件的所有通道中的程序段查找过程
=3	当前所选通道和带相同 BAG 的所有通道中的程序段查找过程
=4	在所有通道中的程序段搜索

在下一次搜索时修改立即生效。

4.6 通道交叉状态显示

功能

下列组件状态可以用符号在状态行（标题）中输出：

- 通道
- 主轴
- 机床状态

除了 SINUMERIK 符号外，还可以对文件 HEADER.INI 与 DB19 DBB32-34 中由制造商定义的符号进行设计。

SINUMERIK 符号

有下列带有 TrueColor 颜色设置的 SINUMERIK 符号可供使用（文件名称在括号中）：

通道状态		通道复位	(NC_RESET.BMP)
		通道激活	(NC_START.BMP)
		通道已中断	(NC_STOP.BMP)
			
进给停止		进给未释放	(FEEDSTOP.BMP)

主轴状态		主轴逆时针旋转	(SPNDLEFT.BMP)
		主轴顺时针旋转	(SPNDRGHT.BMP)
		主轴未释放	(SPNDSTOP.BMP)
		主轴停止	(SPNDM05.BMP)
			

存储 SINUMERIK 符号

SINUMERIK 符号存储在

- F:\HMI_ADV\ICONS\640 (适用于屏幕分辨率 640x480)
- F:\HMI_ADV\ICONS\800 (适用于屏幕分辨率 800x600)
- F:\HMI_ADV\ICONS\1024 (适用于屏幕分辨率 1024x768)

SINUMERIK 符号尺寸如下

- 分辨率 640x480: 16x16 像素
- 分辨率 800x600: 20x20 像素
- 分辨率 1024x768: 27x26 像素

说明

在使用“可切换主轴”或“轴容器”功能时，主轴状态在主轴转换或者分配轴/主轴容器之前输出；即在主轴状态下显示的主轴编号为“逻辑”主轴。

激活

通过机床数据 9052 SHOW_CHANNEL_SPANNING_STATE > 0 激活功能。机床数据 9052 中的更改在上电后生效。

当通过下列机床数据激活功能时，功能可用：

9052	SHOW_CHANNEL_SPANNING_STAT		
机床数据号码	更改通道综合的状态显示		
缺省值： 0	最小输入值： 0	最大输入值： 1	
在上电后更改生效	保护级 (L/S)： 4/1	单元： -	
数据类型： 字符串	有效自软件版本： 6.2		
意义	通道综合状态显示： 0 = 在标题中显示以前的程序状态 1 = 根据文件 Header.ini 中的设计在标题中的程序状态行中显示		

SINUMERIK 状态显示

有下列 4 个预制的显示 可用：

- 1 通道状态 (有效、停止、复位) 带有叠加的主轴停止和进给停止
- 2 通道状态 (有效、停止、复位) 带有叠加的进给停止
- 3 通道状态 (有效、停止、复位)
- 4 主轴状态 左旋转方向，右旋转方向，主轴停止，主轴准停 M05

1 通道状态

在显示位置上将通道状态、进给停止和主轴状态输出为符号。通道状态符号可以由符号“主轴未许可”或者“进给未许可”叠加。

下列 级别 (排列顺序为从最高优先级到最低优先级) 有效：

- 1. 通道停止 (最高优先级)
- 2. 主轴未释放
- 3. 进给未释放
- 4. 通道激活
- 5. 通道复位 (最低优先级)

句法：

ChanStatFeedStopSpindleStop = <Kanalnummer>

Spindles = <Spindelnummer1>, ..., <Spindelnummer n>

通道编号： 通道的编号

主轴编号： 待考虑的“逻辑”主轴编号

(如果在 DIN 代码中用 S1 至 Sn 编程，则在有多个相关数据情况下显示较高的编号)。

2 通道状态

在显示位置上通道状态和主轴状态显示为符号（例如可以额外输出主轴状态）。通道状态符号可以由符号“进给未许可”叠加。

有下列级别（排列顺序为从最高优先级到最低优先级）有效：

1. 通道停止（最高优先级）
2. 进给未释放
3. 通道激活
4. 通道复位（最低优先级）

句法：

`ChanStatFeedStop = <Kanalnummer>`

通道编号： 通道的编号

3 通道状态

在显示位置上通道状态输出为符号。例如，进给停止也输出在加工操作区的进给窗口中。

句法：

`ChanStat = <Kanalnummer>`

通道编号： 通道的编号

4 主轴状态

在显示位置上主轴状态输出为符号。

句法：

`SpindStat = <主轴编号> , Chan <通道编号>`

主轴编号： （逻辑）主轴编号

通道编号： 可选的通道编号

如果未规定通道号，则显示当前通道相关的主轴状态。通道切换时，相应更新显示。

HEADER.INI 中的设计

在文件 HEADER.INI 中确定通道综合状态显示的占用并可以有选择地设计 SINUMERIK 和用户符号。

说明

不必规定空位置。预定义的状态显示可以在目录 OEM 或 USER 中的文件 HEADER.INI 中通过值 <empty> 关闭，例如 `ChanStatFeedStopSpindleStop = <empty>`。

通道综合状态显示和 USER/OEM 图标的应用程序相互排斥，也就是说，要么设计“通道综合状态显示”，要么设计“USER/OEM 图标”。完整的文件见章节 HEADER.INI。

带 16 显示位的示例：

在程序状态行（标题）中确定 16 个显示位置，同时根据 2 个位置分配 SINUMERIK 符号：数字 + 符号。

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SKP1	DRY	ROV	SBL1	M01	M17	DRF	PRT								

图 4-1 16 个位置的程序状态行

为一个输出位置固定分配 SINUMERIK 符号。在设计文件 HEADER.INI 中为每个使用的位置设立一个段 (Section)。

举例 1, 带有 SINUMERIK 符号:

程序状态行视图, 带有用于 4 个通道和 2 个主轴的 SINUMERIK 符号:

3		2		3		4		S1		S2					
SKP1	DRY	ROV	SBL1	M01	M17	DRF	PRT								

图 4-2 带有符号的程序状态行 (1)

总是用预设的通道或者主轴编号进行显示。

举例 2，带有 SINUMERIK 符号：

2 个通道和 1 个主轴的状态显示应输出在第 1、第 3 和第 5 个显示位置上。

文件 HEADER.INI:

```
[Pos1]                ; 第 1 个显示位置
ChanStatFeedStopSpindleStop = 3
Spindles=1,2          ; 显示通道 3 的通道状态，并将通道 3 的进给禁止与主
                        ; 轴 1 和 2 的主轴禁止进行叠加
[Pos3]                ; 第 3 个显示位置
ChanStatFeedStop = 2  ; 显示通道 2 的通道状态，并叠加通道 2 的进给禁止
[Pos5]                ; 第 5 个显示位置
SpindStat= 3          ; 显示主轴 3 的主轴状态
```

在程序状态行中得出显示如下：

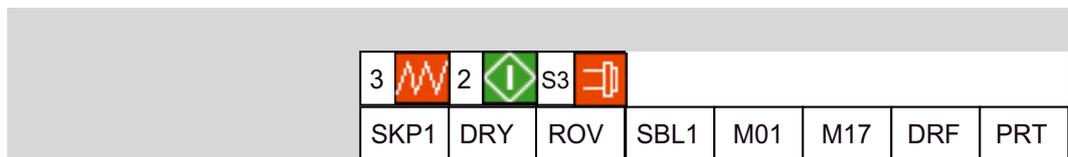


图 4-3 带有符号的程序状态行 (2)

取消激活

在根据 B 使用用户状态显示时，可以取消激活用此方式设计的符号显示。在一个用户特定的文件 HEADER.INI（不在目录 hmi_adv 中，因为在软件升级时会再次覆盖该文件）中设置空条目。

示例：

```
[Pos5]
SpindStat=<EMPTY>
```

参见

HEADER.INI (页 301)

4.7 用户状态显示 (OEM)

功能

另外，可以由 PLC 控制 (DB x, DBB y) 输出与指定机床状态相连的用户图标 (符号)。
操作步骤:

1. 图标和位置一致
2. 通过 PLC 控制图标调用

在文件 HEADER.INI 的 UserIcons 段中，在命名符 UI_0 到 UI_31 下输入用户符号名称和所属的位置。

[UserIcons]

```
UI_0= <Icon_00.bmp> , <Position>
; UI_0:名称
; Icon_00.bmp: 用户图标名称 (8 个字符的.bmp)
; 位置: 显示位置 (1 - 16)
...
UI_31= <Icon_31.bmp> , <Position>
USER_ICON_BASE = DBx.DBBy
```

DBx.DBBy: 由用户确定用于控制图标选择的双字。如果为用户图标设置了一个位，则必须有相应的 UI_x 记录条目，否则不会出现任何显示任何。如果显示的图标与上述一致，则 PLC 在双字中设置相应的位。

举例:

UI_0 显示分配的图标 → 位 0

...

UI_31 显示分配的图标 → 设置位 31。

如果位通过 PLC 复位，则在显示中删除分配的图标。

示例:

UI_3=Icon_01.bmp,12.

如果没有其它带较高标识符 (>UI_3) 和相同位置 (12) 的图标激活，可以通过设置位 3 在第 12 个位置上显示图标“Icon_01.bmp”。

应用方法

选择 **唯一** 时，可以同时为各个不同的位置在 4 个选择字节中进行最大为 16 位的设置。

- 一个位置有 32 个可选的画面，每个画面选择一个其它的位 (目前仅是一个激活的选择位)

- 所允许的 16 个位置中的每个位置有 2 个可选的画面，每个位置需要使用 32 个选择位中的 2 个（在同一时间内最多有 16 个有效的选择位）
- 将总共最多 32 个画面组合起来用于最多 16 个位置
- 多于 16 个选择位

叠加

选择 **不唯一** 时，（用于多个标识符的位，应在同一位置上显示其画面）：

带有较低命名符编号的画面由同一位置上带较高命名符编号的画面覆盖。非唯一的选择位也可以在少于 16 个激活的选择位时使用。PLC 可有意使用该叠加，以用重要的显示（高命名符编号）覆盖带较低命名符编号的次要显示。

说明

如果给一个 NC/PLC 单元分配 2 个 HMI 单元 (M:N)，则 PLC 必须首先根据接口确定已激活哪些 HMI 接口。可以由 PLC 要求图标仅用于激活的 HMI 单元。

设计 OEM 文本

机床制造商可以在程序目录和程序名称行中显示自身的文本，该文本不规定任何 % 参数，如可以在用户文本情况下。行根据 HEADER.INI 中的条目在 <OEM_NUMBER_TEXTFIELD> 区中进行划分。

Maschine	Chan1Maschine Chan1MSTT	JOG Inc 100	OEM_Text 001 OEM_Text 004	OEM_Text 002 OEM_Text 005	OEM_Text 003 OEM_Text 006
/// Kanal RESET			1 // 2 // 2 //	1 // 2 // S2 // 1 //	

图 4-4 OEM 文本举例

在用户 PLC 数据模块中提供数据。数据模块编号和精确的偏移在文件 HEADER.INI 中约定。为此在 [OEMTextInfo] 段中有下列记录条目：

[OEMTextInfo]

OEM_NUMBER_TEXTFIELD = 1; 每行最多有 3 个文本栏

OEM_TEXT_BASE = DBx.DBBy

用户数据模块中的接口有下列格式:

开始 DBB 字节 或者 开始 DBB +18 字节	位 0=1: 必须要做些什么 所有位=1: 由 HMI 确认任务 所有位=0: PLC 可以再次写入接口。
开始 DBB +1 字 或者 开始 DBB +19 字	alpu_xx.com 中待显示文本的偏移值; 700 000 是基准地址并在内部计算。 所有位=0: 删除行
开始 DBB + 3 字符串 或者 开始 DBB+20 双字	字符串由最大 12 个字符组成

第一个参数包含用于带基本 700.000 的文本编号的偏移值 (用户 PLC 文本)。如果该参数 = 0, 则删除相应的行。第二个参数包含有一个字符串, 其长度最大为 12 个字符。如果该文本不为空, 则输出该文本, 而不是输出用户 PLC 文本。

在 PLC 中必须“自后向前”写入参数, 即首先提供所有参数, 然后设置第一个字节。下一行的数据刚好位于开始字节后的第 18 个字节处。自左向右输出行; 规定每行最多 3 个输出栏 (即总共可以有 6 个栏)。

在 HMI 中为每行的第一个字节分别建立一个热链接。如果在那里出现变化, 则 HMI 进行响应。如果 HMI 已显示数据, 则 HMI 写入一个 -1 至第一个变量中。HMI 通过热链接引起该变化; 为此 HMI 写入一个 0 至该变量中。

HMI 对该变化不必自行进行响应。如果将变量置为 0, 则 PLC 可以首先再次将数据写入至接口中。通过该机制确保可以识别到 HMI 下的所有参数变化。

设计 OEM 图标

在规定用于显示程序控制 (例如: SBL、M01、...) 的栏中可以输出用于显示机床状态的 OEM 图标。如果已定义 OEM 图标, 则隐藏用于显示程序控制的元件。

图标名称与在设计文件 HEADER.INI 中的用户图标相似:

```
[OemIcons]
Ol_0= <name.bmp>, <Position>
...
Ol_31= <name.bmp>, <Position>
```

此时，位图文件名的 <name>和输出位置<Position>（1-16）在显示行中。在同一位置上可以输出多个位图。如果同一位置上有多个激活的位图，则显示带最高画面编号的位图。

通过 PLC 双字控制输出。这在文件 HEADER.INI 中在下列段中约定：

[OemIcons]

Oem_ICON_BASE = DBx.DBBy

双字中的每个位表示 OEM 图标与画面编号完全相符。如果已如此设置位 0，则显示位图 OI_0。如果要复位一个位，则删除所属的位图。

最多可显示 16 个图标，因此也存在有 16 个显示位置。不必规定空位置。

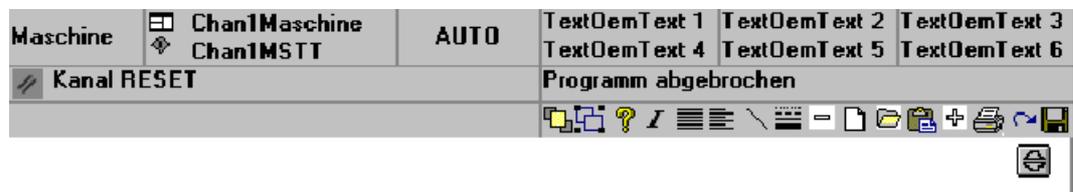


图 4-5 OEM 图标举例

参见

HEADER.INI (页 301)

诊断与服务

5.1 在 PG/PC 上安装 HMI 软件

前提条件

目标系统:	配置为 600 MHz / 256 MB RAM 的 PG 或 PC
操作系统:	空硬盘的空间为 500 MB Windows XP Professional
MPI 驱动器:	版本 6
NC 连接 (SINUMERIK powerline)	CP5511 CP5611
NC 连接 (SINUMERIK solution line) :	网络连接 (Ethernet)

在“系统控制 → 显示、设置”下必须选择“小字体”。

软件

• 名称	用于 PC/PG 的 HMI 高级
• 安装语言	德语/英语
• 分发介质	CD
• 可删除 a	是

操作步骤

1. 将 CD 放在 CD 驱动器中
2. 启动 setup.exe
3. 选择语言
4. 给出目标目录: 可自由选择
5. 对于 SINUMERIK powerline 选择:
 - 在线与 NC 通讯
 - 询问是否要安装 MPI 驱动器
 - 本地与 NC 没有通讯, 带 NC 仿真

6. 对于 SINUMERIK solution line 选择:
 - 在线与 NC 通讯
7. 系统列出所有进行的记录
8. 确认后启动复制过程
9. 结束复制后询问是否要重新启动 PC。

说明

在 PC/PG 上进行安装:

仅当非管理员用户拥有 HMI 高级目录 mmc2 的写入权限时, 该用户才可运行 HMI 高级。HMI 高级的安装不提供该权限。

5.1.1 NCU 连接向导

使用

仅在 PG/PC 上安装 HMI 高级时安装 NCU 连接向导。以便在开机调试 NCK、PLC 和驱动以及配置接口和电缆时可以支持用户。

视不同的控制系统而定, 利用连接向导既能实现 MPI 连接, 也能实现 Ethernet 连接。在 SINUMERIK solution line 上使用以太网连接, 在 SINUMERIK powerline 上使用 MPI 连接。连接方式取决于控制系统 840Di 所安装的软件。

启动“NCU 连接向导”

通过“开始菜单”→“NCU 连接向导”启动连接向导, 或者在不能建立与控制系统的连接时, 连接向导会在显示信息后自动启动。

说明

如果利用“NCU 连接向导”进行更改, 则更改在重新启动系统后才生效。

5.1.2 在自身桌面上启动 HMI 高级

概览

在标准 PC 上安装 HMI 高级会形成边界条件；在使用其他程序时，该边界条件会导致以下情况：

- 窗口不可移动，不可最小化
- 键盘的特殊功能，键盘过滤器
- 任务栏占满
- 颜色搭配变化
- 聚焦监控与 OEM 软件有关
- 所有标准 PC 的应用程序在一个桌面下运行。

在下一段落中会说明补救措施：如何在独立桌面上调用 HMI 高级。

补救措施

通过为 HMI 高级动态建立一个自身的桌面以限制上述特征。此外，最多可定义无限制的三个桌面。可以用多种方式随时切换 HMI 高级的桌面和其它桌面。桌面的创建是可以设计的。此解决方案仅适用于 PG/PC，但不适用于 PCU 50.3。

桌面应用：

- 除了总是存在的标准桌面外，动态创建一个或多个桌面。
- 将 HMI 高级装载至动态创建的独立桌面中。
- 所有现有的桌面包含一个可以随时切换至另一个桌面的操作界面。
- 可以通过文件 **HMIDesk.ini** 来设计桌面。

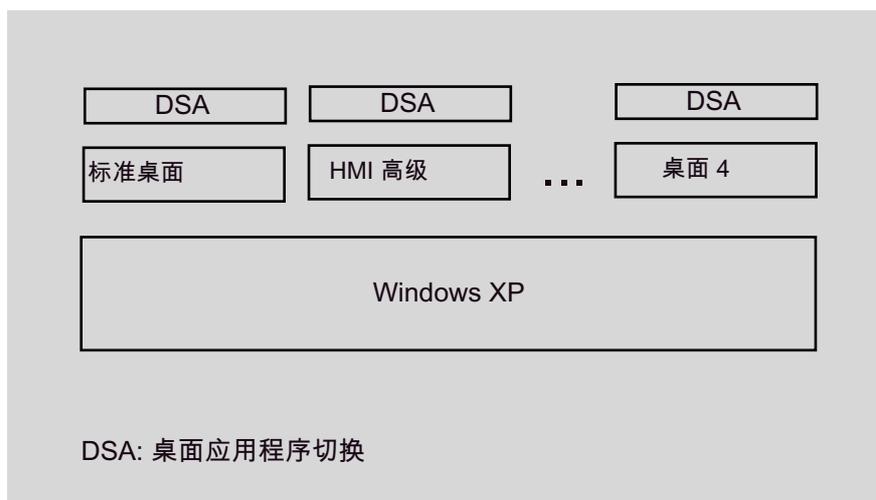


图 5-1 HMI 高级独立的桌面和其他应用程序

说明

1. 任务管理器的键组合 **CTRL+ALT+DEL** 只有在标准桌面上才有效。在动态桌面上仅可以直接通过任务栏调用任务管理器。
 2. 退出动态桌面时，用桌面结束用 `hmidesk.ini` 中的条目自动启动的应用程序。此外，其中（手动）启动的应用程序继续运行。
由此，应用程序不再能够操作，因为这些应用程序在其它桌面中不显示。如果动态桌面在之后的一个时刻里再次启动，则由此可以看见尚在运行的应用程序且能够继续操作。
 3. 所有使用 HMI 高级基本系统的 **外部应用程序** 外部应用程序（例如 **ProTool**，**TRANSLINE**），必须在 HMI 桌面中启动，如果这些应用程序需要与 HMI 高级同时执行。
背景：不能超出桌面限制使用顺序控制的总线系统，因为 **Windows** 限制窗口与桌面的通讯。
 4. **16 位应用程序** 总是由 **Windows** 在登录画面中启动，并因此不用动态 HMI 桌面上运行。因此，桌面切换器在使用 16 位应用程序时在 HMI 高级中不能使用。这种情况下，必须以常规方式启动 HMI 高级。
-

5.1.3 设计桌面切换应用程序

功能

在文件 HMIDESK.INI 的段[Global]中进行设计。

在 [Global] 下使 **NumberOfDesktops = n** 来确定所需桌面的数量。动态创建 2 到 4 的桌面。n 许可的值为 2、3、4。对于每个动态桌面，必须在 [桌面 n] 下进行其它说明。

ControlMode =	Icon	任务栏中的图标(Explorer 下面)
	Window	独立的对话框 (WindowOnTop 下面)
KeyboardControl =	TRUE	可利用<ALT+1>, <ALT+2>, ... 对话框和任务栏进行切换。
	FALSE	可以用对话框或任务栏进行切换

部分 [桌面 n]

StartupApplication = "Pfad\Programm.exe"

建立桌面后自动启动应用程序：退出桌面时，通过桌面切换器结束该应用程序。此时，桌面的退出与应用程序的结束同步。在同步期间，桌面切换器不能处理其它请求。

当应用程序在运行 15 秒后总是执行，则中断同步。这种情况下，同样退出桌面。

说明

对于桌面 (2)，自动设置以当前 HMI 高级安装为参考的路径和所属的程序名称。

Explorer =	TRUE	桌面包含浏览器和任务栏，所设计的桌面的图标出现在此任务栏中。
	FALSE	桌面不包含任务栏和浏览器仅能通过键盘<ALT+1>, <ALT+2>, ...或通过对话框选择桌面。
WindowOnTop =	TRUE	如果 ControlMode = Window，则持续显示桌面切换应用程序的对话框。
	FALSE	可屏蔽对话框
CreateDesktop =	Immediately	桌面立即启动
	Delayed	桌面在选择时启动

SetHMIColors =	TRUE	通过 SetHMIColors 桌面 2 使用自身的颜色搭配，其优点在于，该颜色搭配在其它桌面上无效。该属性仅规定用于 HMI 高级，因为使用自身颜色搭配的桌面和标准 Windows 桌面某些部分差别很大。
	FALSE	桌面，包括 HMI 高级的桌面以标准 Windows 的颜色搭配显示。

在桌面之间切换时，桌面切换器自行切换相应的颜色搭配。

桌面 1 指的是在注册后自动有操作系统校准的用户标准桌面。对于标准桌面，仅可设计特性 WindowsOnTop = TRUE。

- **激活**

在 HMI 高级的安装目录中启动 HMIDesk.exe。设立已设计的桌面。在桌面 2 中将 HMI 高级作为启动应用程序自动启动。

- **设立桌面时的动作**

通过任务栏创建有效的桌面。桌面切换器窗口不可见。通过任务栏操作桌面切换器。

5.1.4 操作桌面切换应用程序

概览

可以设计桌面切换应用程序的操作：

- 键盘热键
- 对话框
- 任务栏中的图标

热键：

- <Alt+1> 切换至桌面 1 中（用户标准桌面）
- <Alt+2> 切换至桌面 2 中（HMI 高级的桌面）
- <Alt+3> 切换至桌面 3 中（可选的其它用户桌面）
- <Alt+4> 切换至桌面 4 中（可选的其它用户桌面）

在单独的桌面中启动 HMI_高级：

1. 首先启动第二个桌面，然后启动 HMI 高级：



图 5-2 连接桌面切换应用程序

2. 点击相应的桌面编号来切换桌面

- 1 标准桌面
- 2 HMI 高级桌面
- (3, 4) 其他可选桌面



↑ 显示桌面

↑ 切换有效桌面

说明

如果在 HMIDESK.INI 中设计了 WindowOnTop = False, 就必须用<Alt + TAB>将窗口拖至前景中。

5.2 保存屏幕截图

使用

为了在故障查找和问题分析时可以提供更多信息, 将屏幕截图 (也称作“截图”) 保存在一个文件中。

截图的文件类型和文件名称

通过按钮组合 <Ctrl> + <P> 建立一个屏幕截图。整个屏幕复制到一个文件中, 带有自动生成的文件名称, 根据下列标题:

- scf00001.bmp
- scf00002.bmp
- scf00003.bmp
- 等等

可以最多保存 99 个截图。之后文件再次以 scf00001.bmp 开始覆盖以前的文件。

截图作为 BMP 文件保存在 PCU 临时目录 F:\ALTMP 下 (MMC.INI 中的条目):

```
[DIRECTORIES]
TempDir=F:\TMP
AIDir=F:\ALTMP
```

5.3 显示版本

概览

在“NCU 版本”下所显示的版本说明规定了 CF 卡的软件版本。这尤其适用于测量循环版本和循环版本。

- NCU 版本数据
- HMI 文件版本数据
- 循环版本数据
- 定义的版本数据

HMI 高级版所使用的测量循环和循环在 PCU 的硬盘上可用。其版本可以通过水平软键进行查看。

使用垂直软键“详细信息”可以查看版本的其它相关信息，例如可以在维修情况时用来联系热线人员。

诊断	CHAN1	AUTO	\MPF.DIR TEST.MPF 程序运行	详细资料
通道生效			ROV	
版本数据 NCU/NCK				
NCK				
Version		67.02.01		
InternalVersion		67.02.01		
Numeric Control		67.02.01		
NCK-SI		67.02.00		
NCKS710-6a2c		67.02.01		
License		67.02.01		
Joblist for IBN		02.04.00		
Joblist for IBN/KOMP		02.04.00		
Joblist for UPGRADE		02.04.00		
Base System		67.01.00		
nrk710.abb		01.09.07		
RtExt-process		05.03.00.00		
RtExt-module		05.03.00.00		
RtExt-module		05.03.00.00		保存版本
				<<
NCU 版本	HMI 版本	循环版本	定义	硬件版本

图 5-3 NCU 版本（举例）

5.4 设置并升级系统

5.4.1 批量开机调试

批量开机调试

“批量开机调试”功能用于对进行控制系统的数据储存。也可以将储存的数据传输给其他控制系统，使这些控制系统达到一致的状态。为此可以将 NC 数据、PLC 数据、驱动数据以及可能存在的 HMI 数据分别或者共同储存在文档文件中，并能够再次读入文件。可以选择将 NC 数据与轴的补偿数据一起保存。

驱动数据无法修改，可以选择将其储存为 acx 格式的二进制数据或以 xml 格式进行保存。

前提条件

进行批量开机调试对访问权限有所要求。在文件 DINO.INI 中规定了相应的访问权限以及用于文档的其他设置。

预设了以下访问等级：

- 从访问等级 3 开始（密码）：用来为批量开机调试建立文档。
- 从访问等级 4 开始（钥匙开关）：可以读入批量开机调试文档。



图 5-4 创建批量调试文档

创建存档

为了创建开机调试文档：

1. 在“通讯”操作区中选择扩展软键栏“创建批量调试”。
2. 选择文档内容。

如果为该文档选择了 HMI 文件：

总是同时保存目录 F:\USER。

- 使用标准值保持预设置。
- 使用软键“HMI 数据选择”来选取所需数据。在“附加产品”下可以保存目录 \ADD_ON 和 \OEM，要么完全保存（“完整”）要么仅保存 INI 文件（“配置”）。

3. 存档名称：接受系统的建议或者输入名称。
4. 使用软键“文档”或“NC 卡”建立文档。

请注意，需要有足够的可用存储空间。

结果：选择“文档”时，文档被储存在硬盘的下列路径下： F:\dh\arc.dir\ ...

或者可以将文档保存到一个分区 USB 闪存中（带有一个 Linux 和 Windows 分区）。

读取存档

前提条件：在新的控制系统中开机调试文档的存放路径，必须与创建时所在的控制系统相同。

为了创建开机调试文档：

1. 选择软键“读入开机调试文档”：显示出目录。
2. 可以用软键“开始”启动读入过程。
3. 用“是”对询问进行确认。

文档被读入并自动重新启动系统，以便使文档数据生效。

说明

在读取文档时硬盘上提供的存储器空间至少必须大于系列 IBN 文档的大小。

如果配置了一个用于 M:N 的控制系统，必须在进行完批量开机调试后重新启动 PCU，使网络用户（NC、PLC、HMI）重新达到同步。

对于可装载的编译循环（CC）则比较特殊：

可以按下列前提条件将可装载的编译循环在批量开机调试中进行存档：

- 开机调试文档中的目标目录 ../NC_CARD.DIR/CCOEM.DIR 里必须至少存在一个编译循环。
- 只能通过复制在 USB 存储器上建立单独的编译循环文档。

5.4.2 PLC 升级

前提条件

不管是 PLC 的升级还是系统数据模块（SDB）的升级，都要求达到存取权限 0 - 3。

使用

进行 PLC 升级以及/或者系统数据模块升级可以更新指定的数据，方法是在一个 INI 文件中记录下不用覆盖以及不用更新的模块：

- 升级 PLC

为了升级 PLC，请选择 PLC 并创建一个 PLC 批量开机调试文档。此外在一个单独的 INI 文件中记录下无需更新的模块。

- 升级系统数据模块

为了升级硬件，请选择“PLC”下面附加的“PLC 硬件升级（仅用于系统数据模块）”，创建并读入系统数据模块（SDB）。这里可以不使用 STEP 7，而是借助所储存的 SDB 升级文档来进行硬件配置。

只有当仅仅选择了选项“PLC”时，才能选取该选项。系统建议的文档名称为 SDB.ARC。在 RAM 中当前 PLC 的所有有效 SDB 都会被记录进该文档中。只有那些由 CPU 进行预设的 SDB 不被记录进来。

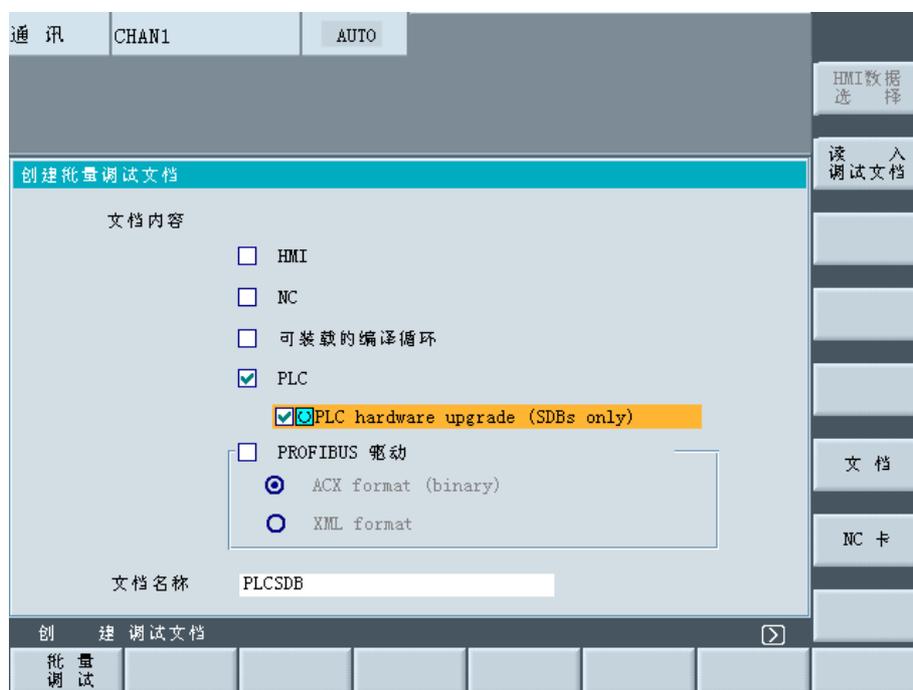


图 5-5 用于系统数据模块的选择

INI 文件的结构

在带有文件后缀名 *.INI 的说明文件中记录有相关的数据模块，它们的内容在升级时保持不变。

该文件的内容按行方式排列如下：

;message=<文本>	;	用于对话行、与语言无关的文本（可选）
DB<编号>	;	整个数据模块<编号>的内容保持不变（1）
DB<从> - <至>	;	<从> - <至>所列出的所有数据模块内容保持不变（2）
DB<编号> [<字节>]	;	数据模块<编号>的字节保持不变（3）
DB<编号> [<从> - <至>]	;	<从>-<至>数据模块<编号>的字节保持不变（4）

- 可以给定用户专用的用户注释。该文本与语言无关，在正常的说明文件中会在输出显示在对话行中，而为了能够在对话行中显示出完全的文本，规定其不得超过 50 个字符。只处理第一条文本指令。
- 可以有空行和注释 - 它们的开头部分是分号。允许使用分隔符（空格符或制表符），但它们不能放在数据模块（DB）符号、地址、范围符号（-）以及字节标识符[] 的中间。另外不区分大/小写。
- 如果已经分配了数据模块的地址范围（2），则不允许另外设定字节地址或字节范围，否则会出现错误提示。如果已经定义了一个数据模块的字节地址（3, 4），则对于该数据模块不允许使用不带字节地址的记录条目（1, 2），否则会出现错误提示。
- 指令以任意顺序排列。

INI 文件的句法检查

可以通过 USB 驱动器或网络驱动器（可选项），将带有文件后缀*.INI 的说明文件复制到文档目录中。PLC 升级时，在显示文件的同时对文件内容的句法进行检查。如果确定句法中有错误，则在显示行中显示第一处句法错误所在行的编号。打开文件时光标停留在出错行上。

升级过程

为了升级 PLC，需进行下列步骤：

- 在升级前为需升级机床创建完整的 PLC 批量开机调试文档，保证在出错情况下可以恢复到原来的 PLC 状态。
- 进行 PLC 升级需要生成完整的 PLC 批量开机调试文档以及一个说明文件，该文件可以由例如机床制造商创建。将文档和说明文件录入待升级机床的文档目录中。强制规定了目标目录。

- 选择 INI 文件，并从文档目录中选取一个 PLC 批量开机调试文档。此时开始进行句法检查。
- 使用软键“开始”可以借助说明文件将数据从文档写入到 PLC 中。为此需要首先为当前机床创建一个完整的临时原始 PLC 文档。可以借助选定的升级文档与说明文件，由该文档建立一个完整的临时目标 PLC 文档。数据模块的长度由升级文档决定，而其内容由说明文件决定。然后读入临时的目标 PLC 文档，而这两个临时文档将升级完成之后被删除。
- 如果要将多台机床升级到相同的 PLC 版本，则可以由第一台升级完的机床创建一个完整的 PLC 文档、并用该 PLC 文档对剩余的机床和原先一样进行开机调试，不再需要说明文件。

在使用这种文档进行升级之前会出现询问：

批量开机调试文档： 升级 PLC (SDB) 的硬件配置？

说明

当选定了一个文档以及一个句法说明文件后，可以按下软键“开始”。

5.4.3 示例：如下进行 PLC 升级

流程

在下例中不得覆盖数据模块 DB3, DB67 和 DB111。

1. 创建一个 PLC 数据的批量开机调试文档作为备份。
2. 创建一个说明文件，带有如下内容：

```
;Message= designed by XXX (SIEMENS AG Erlangen)
DB3[3-78]
DB67 ; DB67 complete
DB 3 [79]
DB3[2]
```

```
DB111 ; DB111 komplett
```

3. 将说明文件保存为 TEST1.INI 并将其储存在需升级控制系统的目录.\dh\arc.dir 中。
4. 选中说明文件 TEST1.INI 和批量开机调试文档来升级控制系统。进行句法检查。
5. 按下“开始”：在复制过程中会输出 INI 文件的文本信息。

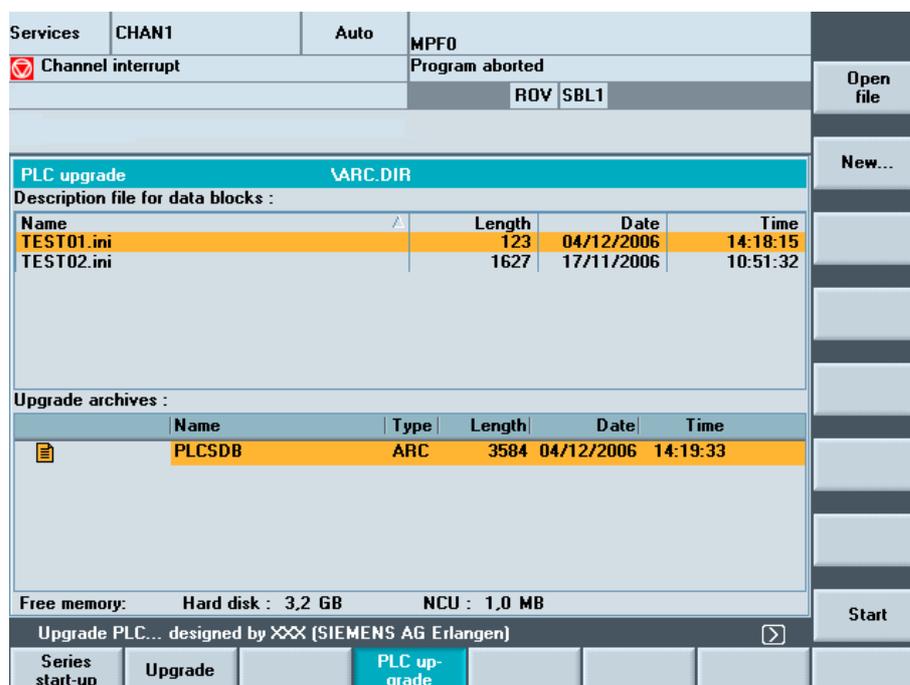


图 5-6 启动 PLC 升级

结果

如说明文件中没有记录条目，则升级文档的所有数据模块被接收至 PLC 中：PLC 升级至当前版本，而不覆盖数据模块 DB3、DB67 和 DB111。

规则

在升级中，为说明文件中所列出的数据模块定义了下列特性：

- 如果升级文档的数据模块在 PLC 中不存在，则在错误记录中会生成相应的记录条目。
- 由升级文档的长度来决定数据模块的长度。
- 如果为数据模块定义了字节地址，则 PLC 会从升级文档中接收该数据范围。如果升级文档或 PLC 的地址不存在，则会输出错误信息。
- 如未定义数据模块的字节地址，则保留 PLC 数据模块的内容。数据模块也可能被缩短至新的长度。
- 如果升级文档中的数据模块比 PLC 中的要大，则在剩余范围内填写入升级文档的内容。

5.5 服务显示

服务显示概述

在选择“服务显示”时显示窗口“服务概述”。这里针对每个机床轴显示不同的带有一个指示灯的运行状态。

另外您将得到一个进一步显示选择用于诊断故障：

- 服务轴
- 系统资源
- 配置数据
- 通信故障记录
- 运行记录仪

服务轴

画面“服务轴”中的信息用于检测下列值：

- 检测标准值分支（例如编程的位置标准值，转速标准值，主轴转速标准值）
- 检测实际值分支（例如位置实际值，测量系统 1 和 2，转速实际值）
- 优化轴的位置调节回路（例如拖行距离，调节偏差，Kv 系数）
- 检测轴的全部调节回路（例如在位置标准值和位置实际值，转速标准值和转速实际值之间比较）
- 检测硬件错误（例如检查传感器：如果机械运动轴，则必须会产生位置实际值的变化。）
- 检测和调整轴监控

文献： 功能描述 基本功能，诊断辅助方式 (D1)

系统资源显示

在窗口“NC-装载”中显示下列 NCU 的系统资源：

- 位置控制器、插补器和进刀的净运行时间与总运行时间，以毫秒为单位
- 百分比形式的 NCU 负载
- 百分比形式的缓冲区水准

使用软键“停止”来中止显示更新，使用软键“开始”重新更新显示值。

输出配置数据

机床配置数据（HMI 版本、NCU 版本、轴配置、驱动配置、总线参数、激活的总线客户端）可以写入到一个文件中，且以后可以读出/打印。

配置数据输出分 2 步进行：

1. 在诊断操作区中通过按下软键“配置数据”建立配置数据文件 CONFIGURATION_DATA。
2. 在“通讯操作区”读出文件 CONFIGURATION_DATA。为此创建配置文件 CONFIGURATION_DATA。

配置数据汇总在文件 CFGDAT.TXT 中。在信息行中显示文件路径和名称。

通信故障记录

在 HMI 和 NCK/PLC 之间通信时出现的错误输入在一个通信错误记录中。该记录通过软键“通信记录”显示。

错误记录文件优先提供给控制装置制造商（西门子）在通信错误时进行诊断。

5.5.1 服务轴

使用

画面“服务轴”中的信息用于检测下列值：

- 额定值支路：例如：位置额定值、转速额定值、编程主轴转速额定值。
- 实际值支路：例如：测量系统 1 和 2 的位置实际值，位置额定值
- 轴的位置控制回路优化：例如：滞后量、控制差值、Kv 系数
- 轴的整个控制回路：例如，通过比较位置额定值与实际值、转速额定值与实际值
- 硬件故障：例如编码器检查。如果轴进行机械运动，则必须会产生位置实际值的变化。
- 轴监控设置和检查

在对话框“服务轴/主轴”中您可通过带有附属轴名称和轴号码的机床轴得到带有单位的识别值。

使用软键“轴+”和“轴-”可以显示下一根或上一根轴的服务数值。使用软键“直接选择”可以从可用的轴中直接选择一根。

5.5.2 显示系统资源

功能

在对话框“NC 装载”中显示 NCU 的系统资源：

- 位置控制器、插补器和进刀的净运行时间与总运行时间，以毫秒为单位
- 百分比形式的 NCU 负载
- 百分比形式的缓冲区水准

使用软键“停止”来中止显示更新，使用软键“开始”重新更新显示值。

5.5.3 输出配置数据

功能

机床配置数据（HMI 版本、NCU 版本、轴配置、驱动配置、总线参数、激活的总线客户端）可以写入到一个文件中，且以后可以读出/打印。

配置数据输出分 2 步进行：

- 在诊断操作区中通过按下软键“配置数据”建立配置数据文件 CONFIGURATION_DATA。
- 在“通讯操作区”读出文件 CONFIGURATION_DATA。为此创建配置文件 CONFIGURATION_DATA。

操作步骤：

1. 选择菜单“服务显示”。
2. 切换水平软键栏。
3. 按下软键“配置数据”。

系统收集配置数据，将其写入到文件 CFGDAT.TXT 中并将其显示。在信息行中显示文件路径和名称。

5.5.4 通信错误记录

功能

在 HMI 和 NCK/PLC 之间通信时出现的错误输入在一个通信错误记录中。通过该软件可以显示该错误。

错误记录文件优先提供给控制装置制造商（西门子）在通信错误时进行诊断。

5.6 运行记录仪

5.6.1 设置运行记录仪

概览

使用“运行记录仪”功能可以记录操作过程。作为预设置，在接通运行记录仪时标记可以记录的数据。运行记录器设置通过口令（访问等级 3：最终用户）加以锁定。

通过操作区“开机调试” → “HMI” → “系统设置”在画面“运行记录仪设置”中选择需要记录控制系统的哪些动作：



图 5-7 运行记录仪设置

记录开/关

其他设置：

记录文件的路径设置固定并且不能改变。

- 记录文件大小：单位为字节
- 循环写间隔：
 - 1: 仅在例如触发器进行请求时，才将记录存储到硬盘上（预设置）。
 - > 0: 在将记录文件存储到硬盘之前的最长等待时间，以秒为单位。
- 报警时程序状态：用逗号隔开报警号的输入：此处输入的报警将直接触发运行记录仪的当前记录立即存储到硬盘上。

记录的数据

记录下列数据:

- | | |
|--|--|
| <p>报警状态更改:</p> <ul style="list-style-type: none"> • 按键 (仅用于 ShopMill): • 通道状态: • Window (到达状态): • 写 NCK 数据: • 文件存取: • PI 通讯:
(调用程序) • 程序状态 / 通道:
(自 NCK 版本 7.4 起) | <p>报警与 PLC 信息的出处与去向。可以记录所有显示出的报警和信息。</p> <p>在使用 HMI 高级时该功能无效。</p> <p>通道状态的变化和倍率开关的位置: 记录 NC/PLC 状态。通过这些状态至少部分确认 MSTT 的操作, 如果时间上允许的话。</p> <p>用软键和功能按键进行操作, 选择图形并切换操作区</p> <p>可以记录所有通过输入区写入的变量 (例如对几何数据的写入权限与刀具补偿和零点偏移一样)。通过变量视图下的参数操作区中的帮助功能调用变量解释。</p> <p>记录所有域通讯 (例如: 装载/卸载程序)。</p> <p>记录所有的 PI 通讯 (例如程序选择、删除文件、删除刀具)。通过变量视图下的参数操作区中的帮助功能调用变量解释。</p> <p>在 NC 启动、NC 停止、复位、PLC 信号 (用于文件 CRASH.COM 的触发器) 事件中会记录下列数据:</p> |
|--|--|

NC 启动、NC 停止和程序复位:

- 当前选定的主程序 (仅用于 AUTO 方式)
- 关于程序级的完整信息 (所有设定的程序级, 带有程序名称与目录)
- 所有程序级的当前 NC 程序段指针 (仅用于 AUTO 和 MDI 方式)

PLC 信号:

(用于文件 CRASH.COM 写操作的释放器)

- 工作方式
- 当前选定的主程序 (仅用于 AUTO 方式)
- 当前有效的程序段 (仅用于 AUTO 和 MDI 方式) 与其前后程序段 (三段显示)。
- 完整的程序级信息 (所有设定的程序级, 带有程序名称与目录)
- 所有程序级的当前 NC 程序段指针 (仅用于 AUTO 和 MDI 方式)

- 所有等待处理的报警和显示信息
- 工件坐标系中所有轴的当前实际值
- 机床坐标系中所有轴的当前实际值
- 所有轴的剩余行程
- 编程速度（仅用于 AUTO 和 MDA 方式）
- 编程的主轴转速（仅 AUTO 和 MDA）
- 当前主轴转速
- 轴与主轴的倍率影响
- 当前刀具与有效的刀具补偿
- 所有有效的 G 功能
- 所有有效的框架、TRAFOS...，如果没有由 G 功能得到唯一结果。
- 所有有效的 M 功能

用于运行记录仪的机床数据

机床数据按下列各值进行设置：

通用机床数据:		值	意义
11295	PROTOK_FILE_MEM[9]	0 1	记录文件的存储器类型： = SRAM = DRAM，用于 solution line
18370	MM_PROTOK_NUM_FILES[9]	3	记录文件的最大数量
18371	MM_PROTOK_NUM_ETPD_STD_LIST[9]	3	标准数据列表 ETPD 的数量

通道机床数据		值	意义
28300	MM_PROTOK_USER_ACTIVE[9]	1	为某一用户激活记录
28302	MM_PROTOK_NUM_ETP_STD_TYP[9]	3	标准事件类型 ETP 的数量

配置文件 ACTLOG.INI

所有运行记录器设置存储在文件 user\actlog.ini 中。显示 MD 9012 ACTION_LOG_MODE 没有意义。

参见

ACTLOG.INI (页 292)

5.6.2 记录文件的结构

概览

当运行记录仪有效时、即打开记录选项时，会生成文件 ACTION.COM。文件 ACTION.COM 为二进制文件，并被当作环形缓冲器一直进行覆盖写操作。最新发生的事件位于文件上方的开始部分。

存在空的二进制文件 CRASH.COM。在文件 CRASH.COM 中保存有 ACTION.COM 记录的中间情况。可以通过设置接口信号 DB19.DBX0.6 或者通过报警的写触发来释放写操作。内容保持至下一次触发。

记录文件的内容

记录文件由以下各列组成：

- 数据
- 时间
- 名称
- 事件 ID
- 事件说明

记录文件举例：

```
HMI 版本:      06.04.24.00          NCK 版本:      660000
Activated Entries:HMI_START HMI_EXIT PLC_CRASH PLC_CRASH_ALARM
KEY_PRESSED KEY_PRESSED KEY_PRESSED KEY_RELEASED KEY_RELEASED
KEY_RELEASED ALARM ALARM_QUIT OPEN_WINDOW OPEN_WINDOW
OPEN_WINDOW CLOSE_WINDOW CLOSE_WINDOW CLOSE_WINDOW
CH_STATE_CHANGED OPMODE_CHANGED TOOL_CHANGED OVERRIDE PI_CMD
DOM_CMD WRITE_VAR WRITE_VAR WRITE_VAR FINDBL_CMD OVERSTORE
FILE_ACCESS AREA_CHANGED USER ACTIVATED DEACTIVATED SUSPEND
RESUME
```

```
--- Date          Time          User          Entry-Id
Entry
=====
--- 30.03.2006 13:26:04 HMI-Adv. 0 WRITE_VAR
NCK-/PLC-Var. wrote: /plc/datablock/byte[c19,0] = 64
--- 30.03.2006 13:26:04 <default> 0 OPEN_WINDOW
```

Windowstate changed: State 9, Index 15 in Application "DG"
--- 30.03.2006 13:26:04 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: VSK 7 SHIFT+F8 (14)
--- 30.03.2006 13:26:01 <default> 0 OPEN_WINDOW
Windowstate changed: State 9, Index 13 in Application "DG"
--- 30.03.2006 13:26:01 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: VSK 5 SHIFT+F6 (12)
--- 30.03.2006 13:25:59 <default> 0 OPEN_WINDOW
Windowstate changed: State -990, Index -1 in Application "RE"
--- 30.03.2006 13:25:59 <default> 0 OPEN_WINDOW
Windowstate changed: State -992, Index -1 in Application "RE"
--- 30.03.2006 13:25:59 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: HSK 4 F5 (5)
--- 30.03.2006 13:25:04 <default> 0 OPEN_WINDOW
Windowstate changed: State 1, Index -1 in Application "DG"
--- 30.03.2006 13:25:03 <default> 0 OPEN_WINDOW
Windowstate changed: State -992, Index -1 in Application "RE"
--- 30.03.2006 13:25:03 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: HSK 4 F5 (5)
--- 30.03.2006 13:25:02 HMI-Adv. RE 0 KEY_PRESSED
Key pressed: AREA-SWITCH (a)
--- 30.03.2006 13:24:56 HMI-Adv. 0 CH_STATE_CHANGED
Channel State changed to active
--- 30.03.2006 13:24:56 HMI-Adv. 0 TOOL_CHANGED
Active tool changed to 0, Duplo 1

Mode:AUTO Program: stopped Channel: interrupted

Program-Level information:

Level Program Invoc Offset

running:

1 /_N_MPF_DIR/_N_MAINPROG_MPF 1 0

2 D:/TEST/MYSUB.SPF 1 0

stopped:

1 /_N_MPF_DIR/_N_MAINPROG_MPF 0 -1

2 D:/TEST/MYSUB.SPF 0 0

Actual Block:
T0X0Z0
Number of Machine axis:6
MCS-Name Position DistToGo
X1 46.300 0.000
Y1 108.600 0.000
Z1 4439.366 0.000
A1 319.864 0.000
B1 114.935 0.000
C1 0.000 0.000

Zero-Offset Shift Fine Mirror Rotation Factor
ACTFRAME 24.000000 22.300000 0 0.000000 1.000000
IFRAME 56.000000 3.300000 0 0.000000 1.000000
PFRAME 4398.000000 34.340000 0 0.000000 1.000000
EXTFRAME 0.000000 0.000000 0 0.000000 1.000000
TOTFRAME 0.000000 0.000000 0 0.000000 1.000000
ACTBFRAME 0.000000 0.000000 0 0.000000 1.000000
SETFRAME 12.000000 0.300000 0 0.000000 1.000000
EXTSFRAME 33.000000 2.300000 0 0.000000 1.000000
PARTFRAME 4354.000000 34.000000 0 0.000000 1.000000
TOOLFRAME 0.000000 0.000000 0 0.000000 1.000000
MEASFRAME 0.000000 0.000000 0 0.000000 1.000000
WPFAME 0.000000 0.000000 0 0.000000 1.000000

关于“状态”和“索引”的更多说明参见：
文献：HMI 编程软件包，第 1 部分 用户指南 → “顺序控制”

5.6.3 这样保存记录文件。

前提条件

在操作区“诊断”→“服务显示”→“运行记录仪”中操作（软键“MMCWIN 文件”和“Crash Log”），将当前的记录文件显示在屏幕上。

释放触发后的记录文件

在下列情况中会生成文件 CRASH.COM:

- 接口信号 DB19.DBX0.6“备份运行记录”从 0 切换到 1 时。
- 出现“报警触发写操作”区中所规定的报警时。

该位必须由 PLC 在用户接口上设置（象例如键禁用、黑屏控制...）并再次取消。HMI 高级评定信号的上升沿（即从 0 过渡到 1），并在信号出现时生成记录文件。HMI 高级仅对数据进行读取访问，即用户可在 PLC 程序内进行信号复位。

如果在启动 HMI 时已设置位，则 HMI 高级不对位的值产生响应（仅在复位和设置后再次进行）。

备份记录

为了读出记录文件，进行如下步骤:

1. 要储存记录数据可以选择“备份记录”。

可以备份下列文件:

- ACTION.COM（二进制文件）
- ACTION.LOG
- CRASH.COM（二进制文件）
- CRASH.LOG

2. 记录只能储存在 ..\诊断\记录文件下。

例如可以选择“通讯”→“数据输出”→“软盘”来读出记录文件。

5.7 HMI 分析仪

5.7.1 使用 HMI 分析仪

使用

HMI 分析仪用于诊断 PCU 50.3 上出现的问题。通过 HMI 分析仪以列表方式显示检查结果。也可以另外保存在 ASCII 文件中以便以后分析。用这种方式可以确定供货版本和经顾客更改的版本间的区别。

功能

将分析 HMI 高级上 INI 文件的所有设置并显示由此生成的设置。
可通过 DDE 消息列表、过程监控以及存储器使用和网络设置来进行其他分析。

配置 HMI 分析仪

在文件 HMIAnalyzer.ini 中配置 HMI 分析仪：

```
[HMI_DIRS]
```

```
DIR01 = MMC2
```

```
DIR02 = HMI_ADV
```

```
DIR03 = ADD_ON
```

```
DIR04 = OEM
```

```
DIR05 = USER
```

```
[HMI_FILES]
```

```
FILE01 = MMC.INI
```

```
FILE02 = REGIE.INI
```

```
FILE03 = MBDDE.INI
```

```
FILE04 = OEMFRAME.INI
```

```
FILE05 = NETNAMES.INI
```

```
FILE06 = KEYS.INI
```

```
[Additional]
```

```
FILE01 = F:\Programs\Test1.ini
```

```
FILE02 = F:\Programs\Test2.ini
```

段[HMI_DIRS]

此处输入所有用于分析 INI 文件的目录。

段[HMI_FILES]

此处输入所有用于分析的 INI 文件。同时查找和分析目录中输入到 HMI_DIRS 的文件。

段[Additional]

此处可以输入不属于 HMI 高级的 INI 概念的其他 INI 文件，例如辅助应用程序的 INI 文件。输入时需要完整的路径。只分析该路径的文件。

5.7.2 操作 HMI 分析仪

概览

在启动 HMI 分析仪时确定初始化文件 HMIAnalyzer.ini 的设置。将分析在树形图左面选择的项。分析结果以列表显示在右边。

可将分析结果保存为“Profile”。保存所有在树形图中标记的项（转换键、或者双击各自的框）。

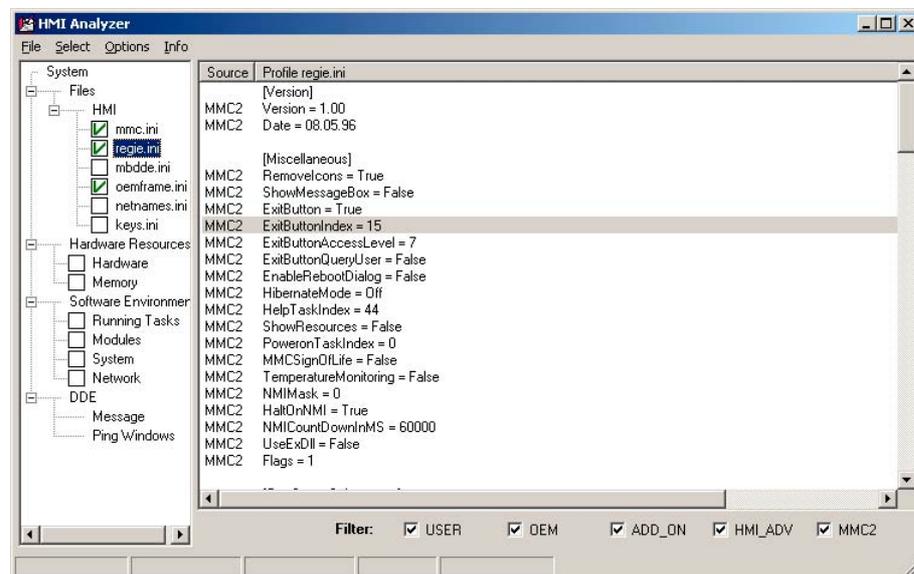


图 5-8 Ini 文件举例

保存当前数据(Profil)

通过菜单项“File”（文件）→“Save Profile”（保存 Profile）可以保存当前所选项的 Profil。只有树形图中的项标记为保存时，随后菜单项才激活。

保存时，在 HMI 分析仪所在的目录中会建立子目录。目录名称由日期和保存数据的时间组成。

举例：

如果在 2006.01.22 的 08:37:55 时保存数据，则目录名称为“20060122_083755”。在此目录中，所有于此时间点标记的 INI 文件保存为 Profile。

保存为 Profile 的 INI 文件主要是“生成”的 INI 文件。Profile 说明了当前有效的配置。

也可以单独在外部 ASCII 编辑器中查看结果数据。

删除当前数据(Profile)

通过菜单项“File”（文件）→“Delete Profile”（删除 Profile）将单个 Profile 从硬盘中删除。

比较两个保存的 Profile

通过菜单项“File”（文件）→“Compare”（比较）可以在主窗口中相互比较保存的 Profile（配置）。

目前保存的 Profile 在列表“Available Profiles”（可用 Profile）中列出。通过按钮“Add à”（添加 à）在列表中选择要比较的 Profile，并将它拖入到列表“Compared Profiles”（比较的 Profile）中。第一个所选的 Profile 作为原始 Profile，第二个所选的 Profile 作为经修改 Profile。通过按键“⌘ Remove”（删除 ⌘）可以再次把 Profile 拖回列表中。通过箭头向右可以改变列表的顺序。

在列表“Compared Profiles”（比较 Profile）中选择了两个 Profile 供比较后，便可以操作按钮“Compare”（比较）。进行比较后，结果会显示在应用程序的主窗口中并关闭选择窗口。

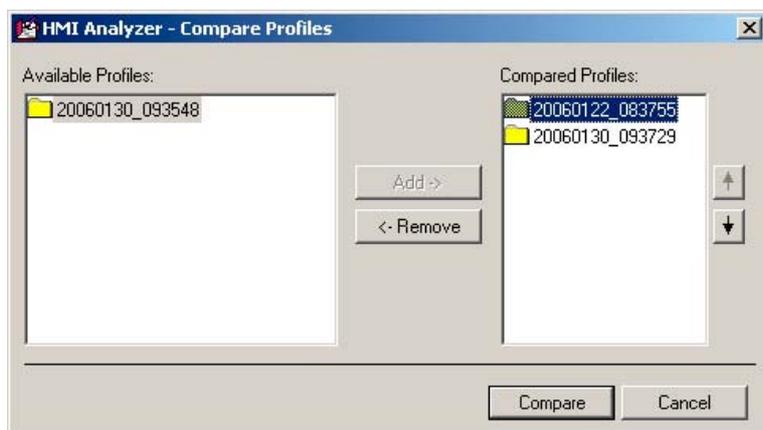


图 5-9 比较 Profile

可在应用程序的主窗口中查看比较结果。此外，也可以通过菜单项“File”（文件）→“Save Comparison”（保存比较）将比较结果保存到 ASCII 文件中。

通过菜单项“File”（文件）→“View Comparison”（显示比较）可以再次用 HMI 分析仪显示保存的比较结果。

5.7.3 分析 INI 文件

系统 → 文件

此处显示分析的 INI 文件的所有条目。

- HMI** 显示不属于 HMI 高级 INI 概念的文件中的相应 INI 条目。此外，也显示该条目所在的目录（栏“Source”）。查找所有在 HMIAnalyzer.ini 中 HMI_DIRS 下输入的目录。
- Additional Files** 显示不属于 INI 概念的其他 INI 文件，即：仅在规定的路径中查找该 INI 文件。

系统 → 硬件资源

- 硬件** 在硬件条目下，列出现有硬件的部分，例如：关于处理器的信息以及关于单个硬盘分区和驱动器的信息。
- Memory** 此条目下显示现有和可用存储器的详细信息，以 kB 为单位：

Used Memory:	已占用的存储器，%
Physikal Memory, total:	RAM 的总容量
Physikal Memory, available:	剩余可用的 RAM 存储器
Page File, total :	页面文件的总存储器
Page File, availale:	剩余可用的页面文件的存储器
Virtual Memory, total:	虚拟存储器的总容量
Virtual Memory, availble :	剩余可用的虚拟存储器

系统 → 软件环境

Running Tasks 此处显示存储器占用和所有有效程序的运行。
显示下列栏：

名称	程序名
Running	运行 'X'过程 '\ ' 过程结束
ProcessID	程序 ID
WorkingSetSize	当前存储器占用，以 kB 为单位
DiffWorkingSetSize	自启动 HMI 分析仪或者自上次复位起存储器的变化
PagePoolUsage	页库存储器，以 kB 为单位
DiffPagePoolUsage	自启动 HMI 分析仪或者自上次复位起页库存储器的变化
PageFaults	页面文件的存取损坏
DiffPageFaults	自启动 HMI 分析仪或者自上次复位起页面文件的存取损坏的变化

其他设置(如下):

Cyclic	循环更新视图。计算并显示自启动或者上次复位起的差值。可在选择“Cyclic”左边规定更新率为毫秒[ms]。
复位	复位差值。只可在循环显示时进行此动作。
Single	作为静态显示一次更新视图。重新计算所需存储容量。

Modules 选择 Module 项时，确定对于所有运行程序加载的 DLL。此外，HMI 运行时，启动所有操作区内以 ProLoad:=False 输入到 Regie.ini 中的应用程序。此时同样确定加载的 DLL。

系统 显示 PCU 或者 PC 通用数据的消息

Network 显示当前网络数据：
显示所有能通过指令 “ipconfig /all”确定的数据。对于操作系统 Windows NT 还会确定，是否安装了 NetBeui。

说明

以 PreLoad:=False 输入到 REGIE.INI 中、且作为程序无法查找到的操作区，在分析前由 HMI 分析仪启动。通过应用程序服务器进行启动。此时用“SwitchToTask”激活所有待启动的操作区。在启动所有操作区后，切换到操作区“加工”。

在通过 DCTL 进行热链接时，DCTL 会另外向数据询问发送请求。HMI 分析仪会应此请求不显示当前值（栏“Value”）。

5.7.4 应用实例

(1) 安装的 OEM 应用程序无法启动

问题:

自身的 OEM 应用程序已安装入 HMI 高级, 但在按下相应的软键后应用程序不启动或者启动其他应用程序。

可能的原因:

由于 HMI 目录处理是等级式的, 带更高优先级的另一 `regie.ini` 的条目会覆盖原先输入 `Regie.ini` 的条目, 因此原条目无效。

例如: 在 OEM 目录中已输入以下条目:

```
[TaskConfiguration]
```

```
Task6 = name := TestApp, Timeout := 10000, PreLoad := True
```

随后, 在 User 目录中会覆盖该条目:

```
[TaskConfiguration]
```

```
Task6 = <empty>
```

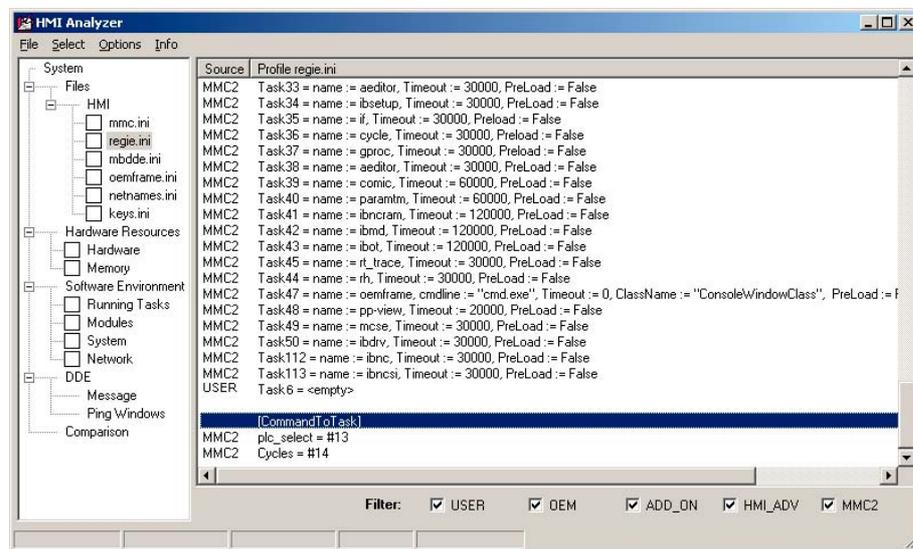


图 5-10 检测 OEM 应用程序启动

HMI 分析仪:

在文件 → HMI 中选择文件 **regie.ini**。现在显示由所有 HMI 目录生成的 **regie.ini**(mmc2、hmi_adv、add_on、oem、user)。现在可以明确，已输入的条目是否实际有效或者已由其他条目覆盖。另外，用户也可以识别出该条目所在的 HMI 目录。

(2) 与 NCU/PLC 没有连接

问题:

虽然已匹配 **mmc.in** 中的 NCU/PLC 地址，但不能建立与 NCU/PLC 的连接。

可能的原因:

由于 HMI 目录处理是等级式的，带更高优先级的条目会覆盖原先输入 **mmc.ini** 的条目，因此原条目无效。

例如：在 **OEM** 目录中已输入以下条目：

```
[NCU840D]
ADDRESS0=5,NAME=/NC,SAP=d
ADDRESS1=6,NAME=/PLC,SAP=1
```

随后，在 **User** 目录中会覆盖该条目：

```
[NCU840D]
ADDRESS0=14,NAME=/NC,SAP=d
ADDRESS1=14,NAME=/PLC,MAX_BUB_CYCLICS=0
```

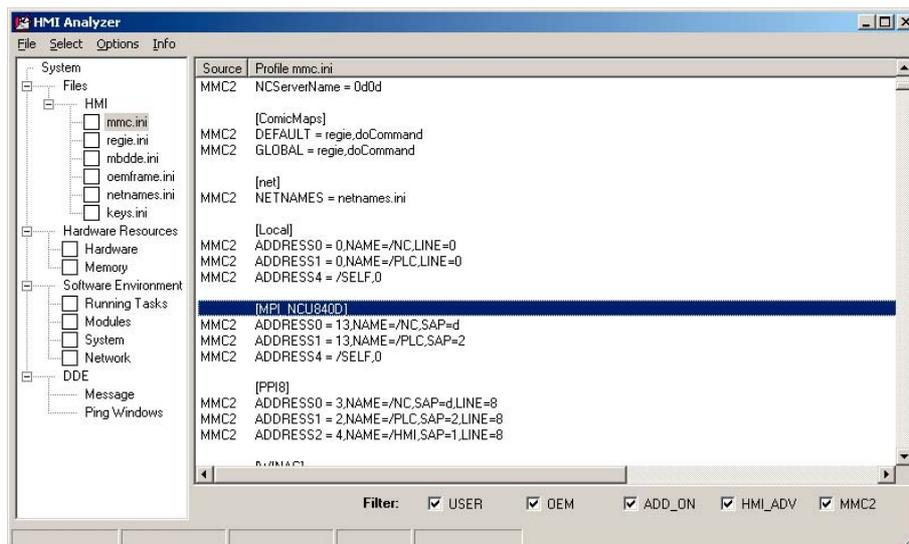


图 5-11 检测与控制系统的连接

HMI 分析仪:

在文件 → HMI 中选择文件 mmc.ini。现在显示由所有 HMI 目录生成的 mmc.ini(mmc2、hmi_adv、add_on、oem、user)。现在可以明确，已输入的条目是否实际有效或者已由其他条目覆盖。另外，用户也可以识别出该条目所在的 HMI 目录。

(3) 产生阻碍、非常迟缓的系统

问题:

系统运行非常迟缓并且完全阻碍部分区域切换。在一定时间后所有操作突然相继结束。

可能的原因:

在加载热链接时装入通过区域切换重新选择的区域应用程序。在装载 DCTL 格式时，向所有程序的主窗口发送消息 DDE 初始化。如果主窗口较晚应答该消息或者不应答，则可能引起区域切换缓慢或者受阻。

规定:

“Toplevel”窗口不能长时间阻碍系统。DDE 客户不允许使用 DDEML，必须使用 DCTL。

HMI 分析仪:

选择项 DDE/Ping Windows。现在可以通过启动循环运行，以检测是否所有的窗口应答了 DDE 广播。应答时间“Answer Time”应设置为 100 ms，而询问的循环时间“Cyclic Time”应设置为 1000 ms。勾住 Append 并取消勾住 ShowAll。出现在列表中的应用程序没有在 Answer Time 内作出应答。显示的应用程序有可能是引起系统产生障碍、运行缓慢的原因。

说明

特别是在运行时建立 DDE 连接的用户，会遇到此问题，和往常一样使用 DDEML 的用户，应该换用 DCTL 或者将连接建立限制为应用程序方式。使用高于 32 位的 DCTL 才能解决问题。相应地，必须使用 VB4.0 程序。

在存储器容量较小的系统上，此问题经常发生。必须在广播（例如：DDE 初始化）时将所有暂存在硬盘上的程序载入主存储器并执行程序。这一步骤非常耗时。有时间限制的应用程序，例如：数据库存取、大容量的计算机程序等，特别是编程背景程序时必须确保在前景中运行的应用程序不受阻碍。

(4) 较长的通讯时间

问题:

如果设置了太多热链接，会极大程度地延长通讯时间。

可能的原因:

视 PLC 负载而定，引发问题的热链接数量各有不同。热链接作为控制系统的基本负载，应尽量少地使用。

HMI 分析仪:

必须在启动 HMI 高级之前启动 HMI 分析仪。可在服务模式下进行。在 HMI 高级启动后，HMI 分析仪在 DDE → 信息会下显示实际存在的热链接。利用选项“Only Active Hotlinks”可以显示所有有效的热链接。现在可以检测，何种数据存取可能无需作为热链接实现，但同样可以作为请求实现。由此可以减轻控制系统的基本负载。

利用过滤器请求、存数或者执行可以检测，该 DDE 功能是否负载系统。短时间的循环请求会负载系统。可用栏“Time”确定循环时间。此栏以毫秒显示启动 Windows 后经过的时间。

说明

OPC DataAccess 编程器

在编程 OPC 用户应用程序时，如果各自组的变量定义为热链接，则待读取的变量自动设置为热链接。

此外，PLC 变量在 SINUMERIK OPC 服务器上自动(Flag IsActive=True)设置为热链接。

刀具管理

6.1 在 HMI 上编程刀具管理

6.1.1 创建一个真正的刀库

创建真正的刀库

在该画面中用相应数据定义刀库或显示已存在的刀库。

图 6-1 开机调试：刀库

刀库	
名称	输入（新）或选择真正刀库的名称。
显示文本	和国家语言有关的刀库名称
编号	显示连续的刀库编号
方式	选择刀库类型（链刀库，转塔，平面刀库）
刀位	输入或显示刀库位置数目。

刀库	
栏数	数据“栏数”需要用于邻近库的观察，只和平面刀库相关。

说明

最多可以设立 32 个刀库（包括中间存储器 and 装载刀库），最多 30 个真正刀库。

创建新的刀库

进行下列步骤：

1. 按下软键“新建”。由系统规定刀库号码（按照产生顺序）。
2. 输入刀库名称（= 标准名称），最多 32 个字符。如果存在，立即显示 patm_xx.ini 中的显示文本。
3. 选择刀库类型：
 - 链刀库
 - 转塔
 - 平面刀库
4. 输入刀库位置数。
5. 对于平面刀库，还应输入“栏数”。
6. 用软键“确定”接受数据。

说明

对于信息“刀库中非法值”，表示位置数据或者栏数错误。值“位置数”必须可由“栏数”除尽。

示例：20 个位置不能分为 3 栏，21 个位置可以分为 3 栏。

删除刀库：

1. 选择刀库名称。
2. 按下软键“删除”。
 - 不进行重复询问，删除刀库。

说明

如果没有分配刀库配置，只能删除一个刀库。

建立 PLC 数据：

- 如果已输入所有刀库，中间存储器 and 装载位置（用于所有通道 / TO 范围），必须建立 PLC 配置数据并传递给 PLC。这通过软键“建立 PLC 数据”进行。

说明

在下次启动 PLC 时可能出现信息“删除 PLC 中 DBxx”。在这种情况下可通过步骤 7 删除规定的 DB。或者也可以装载一个合适的 PLC 文档。

6.1.2 这样编程一个中间存储器

中间存储器

中间存储器位置时主轴，抓刀器，装载器和交付位。所有中间存储器位置都被归并在一个编号为 9998 的内部中间存储器刀库中。

在该画面中定义（新建）或显示已存在的中间存储器位置。

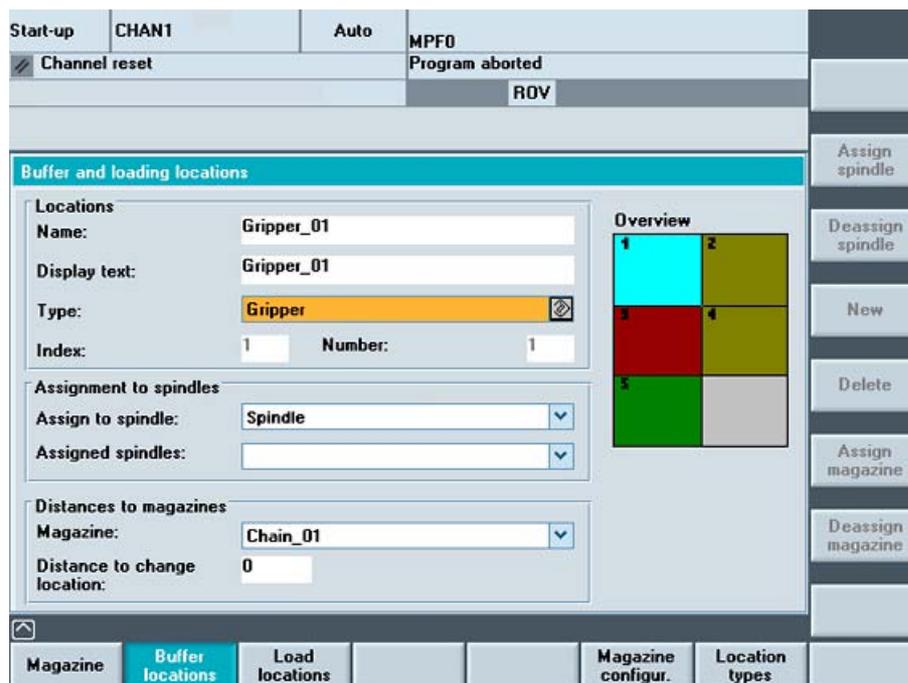


图 6-2 开机调试：中间存储器

刀位	
名称	输入或选择中间存储器名称。
显示文本	和国家语言有关的刀库名称 如果存在，立即显示 patm_xx.ini 中的显示文本。
编号	显示连续的刀库编号
方式	选择一个中间存储器类型（主轴/刀架，抓刀器，交付位置，装载器）
索引	索引计数一个类型的位置
编号	显示连续的内部编号，位置编程在这些编号下
一览	这里图像显示中间存储器位置数目。 其中以光亮标记当前所选的中间存储器位置（编号）。每种“类型”都以另一种颜色描述。

说明

在输入中间存储器时注意顺序。主轴总是作为第一个输入。对于中间存储器位置，给出一个内部编号，通过该编号对中间存储器位置编程。

中间存储器“主轴”带有索引 1 和 NV 中的主轴_1 直接关联。即带有索引 1 的中间存储器“主轴”也必须是 NC 的 1. 主轴，索引 2 = 2. 主轴。

建立一个中间存储器位置

进行下列步骤：

1. 按下软键“新建”。
2. 输入名称：例如 **抓刀器_1**。如果存在，立即显示 patm_xx.ini 中的显示文本。
3. 选择类型：交付位置，**抓刀器**，装载器，主轴
4. 用“确定”建立中间存储器位置，位置编号和索引由内部规定并升序计数。

分配/脱开主轴：

1. 通过软键“分配主轴”分配一个主轴中间存储器位置（例如抓刀器）。由此软件识别抓刀器和主轴之间的机械联系。
2. 通过软键“脱开主轴”删除一个存在的主轴分配。
3. 分配中间存储器位置到刀库/脱开
4. 在选择一个刀库后，必须输入到更换位置的距离。至少必须输入 0，否则中间存储器位置无法进行刀具运输。用软键“分配刀库”分配中间存储器位置给刀库。
5. 用软键“脱开刀库”删除某个刀库的已存在的中间存储器位置分配。

示例

例如如果由 2 刀库刀具切换到“主轴_1”，必须对两个刀库进行相应的分配。

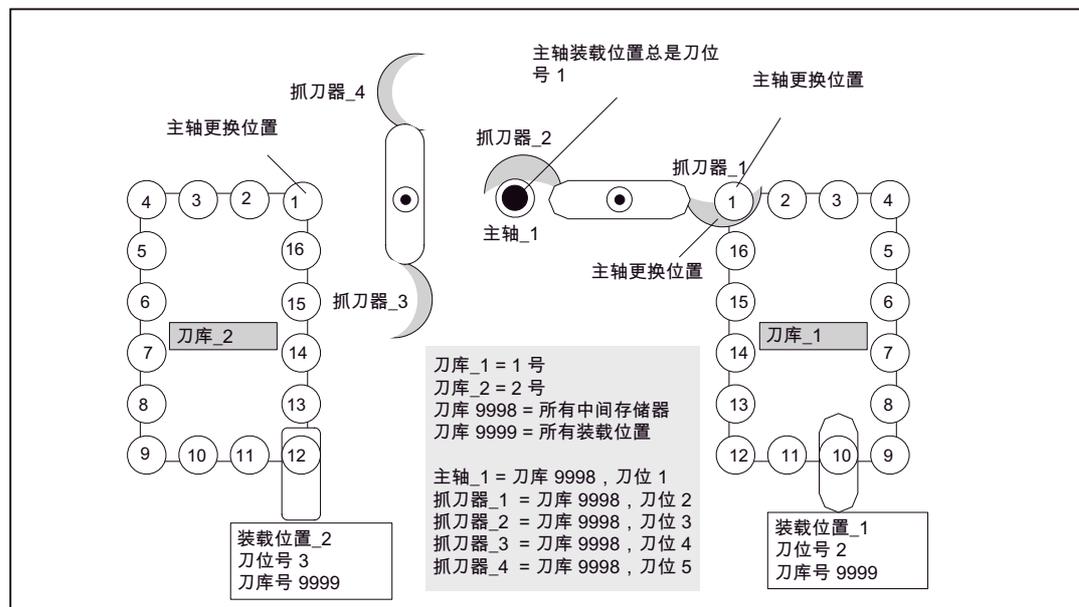


图 6-3 例如带有中间存储器和装载刀库的机床

编号	名称	至更换位置的距离
1	主轴_1	刀库_1, 距离: 0 刀库_2, 距离: 0
2	抓刀器_1	刀库_1, 距离: 0
3	抓刀器_2	刀库_1, 距离: 0
4	抓刀器_3	刀库_2, 距离: 0
5	抓刀器_4	刀库_2, 距离: 0

参见

用户定义名称的国家语言相关性 (页 253)

6.1.3 装载位置和装载刀位

装载刀位

装载刀位是刀库装载所需要的刀位。装载刀位有两种类型：

- 装载位置
- 装载站

所有装载刀位都被归并在一个编号为 9999 的内部装载刀库中。

在该画面中定义（新建）或显示已存在的装载刀位。

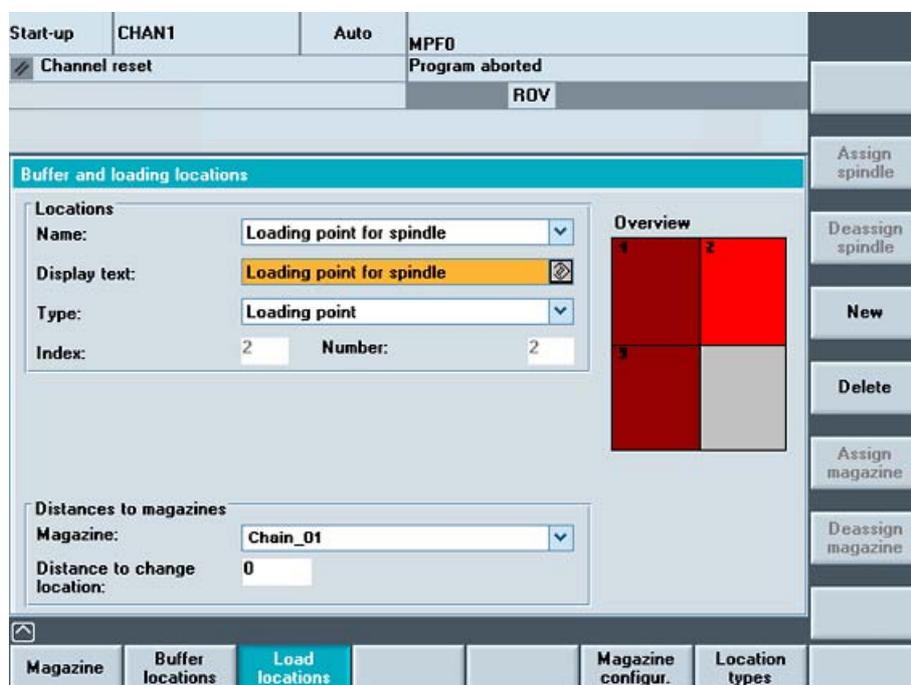


图 6-4 开机调试：装载刀位

刀位	
名称	装载刀位名称（最大 32 个字符）。
显示文本	和国家语言有关的刀库名称。如果存在，立即显示 patm_xx.ini 中的显示文本。
编号	显示连续的刀库编号
方式	可以在装载位置和装载站之间选择。
索引	索引计数一个类型的位置
编号	显示连续的内部编号，位置编程在这些编号下
一览	这里图像显示装载刀位数目。所选择的装载位置或装载站编号用颜色标记。

说明

装载位置 9999/1 在装载/卸载时总是自动设置在主轴（手动装载位置）中。

装载位置

装载位置是机床上的范围，在该机床上可以直接装载到刀库中，即刀具可以在该位置直接插到刀库中。这时将待装载的刀库位置运行到装载位置上。装载位置有例如链刀库。

装载位置在 NCK 刀库数据中有刀位类型“7” (\$TC_MPP1)。

刀位类型 7: 如果一个刀具由刀库或者刀架运动到刀位上，则刀具在确认后通过 PLC 由 NCK 自动从该刀位去除。

装载站

装载站是要插入待装载刀具的刀库之外的刀位。从这里刀具通过一个机械运输机构安放在刀库中。通常对于平面或链刀库来说是装载站。

装载站在 NCK-刀库数据中有刀位类型“6” (\$TC_MPP1)。

刀位类型 6: NCK 不做特殊处理，即刀具运动到该刀位上，保持在那里。只能通过明确的操作处理去除（卸载）。

顺序

输入装载刀位：

1. 按下软键“新建”。
2. 输入名称：例如 **装载_1**。如果存在，立即显示 patm_xx.ini 中的显示文本。
3. 选择类型：例如 **装载位置**
4. 用“确定”建立装载刀位。内部规定刀位编号和索引并升序计数。

分配/脱开装载刀位和刀库：

1. 在选择一个刀库后，必须输入到更换位置的距离（至少 0）。
2. 只能通过软键“分配主轴”建立一个分配。
3. 对于“手动装载位置”（主轴装载位置）一般使用距离为 0。

示例

两个刀库可以通过主轴_1 装载。装载位置_1 只能分配给刀库_1，装载位置_2 只能分配给刀库_2。

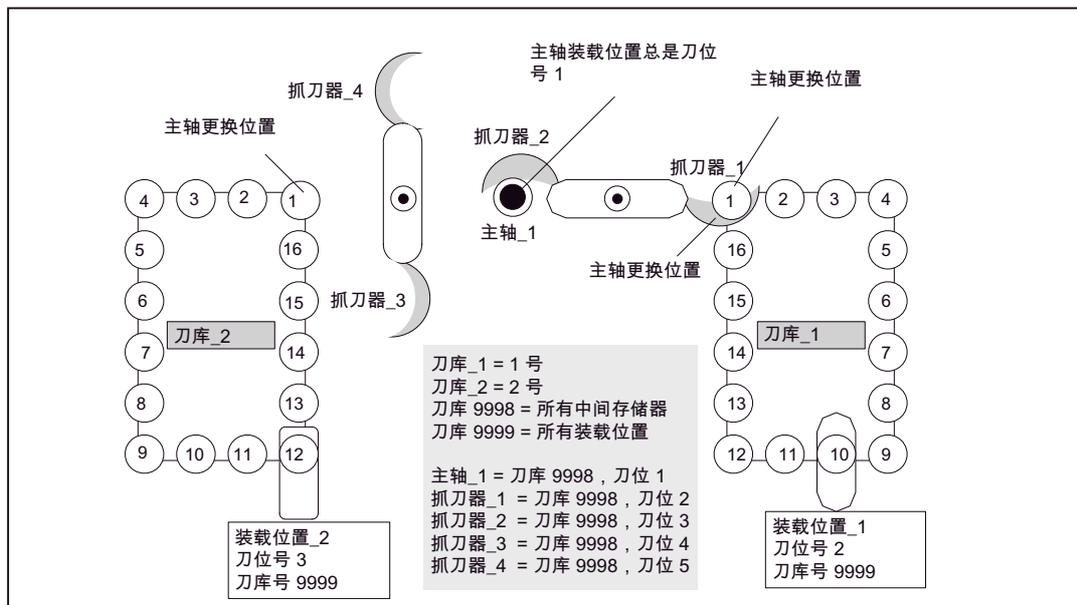


图 6-5 例如带有 2 个刀库和 3 个装载刀位的机床

编号	名称	至更换位置的距离
1	手动装载位置 (= 主轴装载位置)	刀库_1, 距离: 0 刀库_2, 距离: 0
2	装载位置_1	刀库_1, 距离: 9
3	装载位置_2	刀库_2, 距离: 11

6.1.4 选择刀位类型

刀位类型

在该画面中用所需的数据定义（新建）刀位类型或显示已存在的刀位类型。

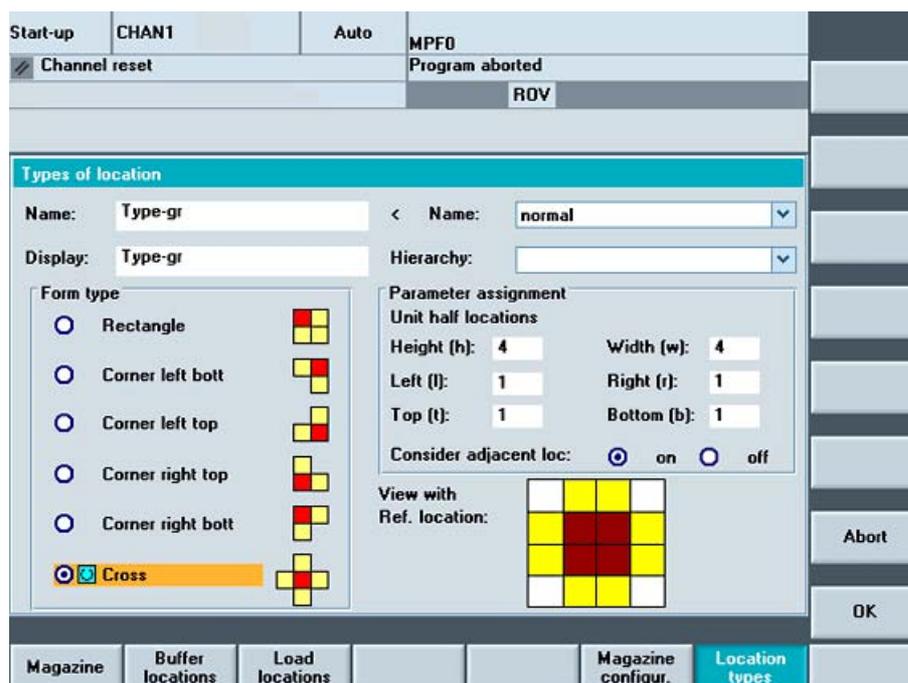


图 6-6 开机调试：刀位类型

刀位类型	
名称	刀位类型名称（最大 32 个字符）。
显示	和国家语言有关的刀库名称。如果存在，立即显示 patm_xx.ini 中的显示文本。
级别	为了取消根据刀位类型刚性划分刀库位置，刀位可以按照升序布置引入一个“级别”（参见章节“刀位类型级别”）。
形式类型	通过形式类型确定参考刀位的位置（刀具杆）。
参数化	确定高度和宽度以及未占用的半位（左，右，上，下）（参见示例）。
观察临近刀位	这些信息由刀库专用保存（刀库配置），与空位查找相关。

刀位类型的参数化

在刀位类型参数化时依据刀具在刀库中占用的半位数目设立。这符合刀具尺寸。

从参考点规定刀具尺寸的四位数，例如 2 2 2 2，半位按照顺序左、右、上、下。对于刀库类型参数化，左加右得出宽度，上加下得出高度。

此外通过数据左、右、上和下，在参数化时定义未占用的半位（不要与刀具尺寸换错！）。

参考刀位

参考刀位标示刀库中的机械刀位。用作刀具尺寸数据的参考点，计算刀库占用时需要。参考刀位的尺寸总是以刀具尺寸 1 1 1 1 描述。（参数化参见“一般刀位类型”）

6.1.5 示例：参数化刀位类型

一般刀位类型

占用一个刀库位置的刀具具有尺寸 1 1 1 1。该刀具称作“一般尺寸刀具”。

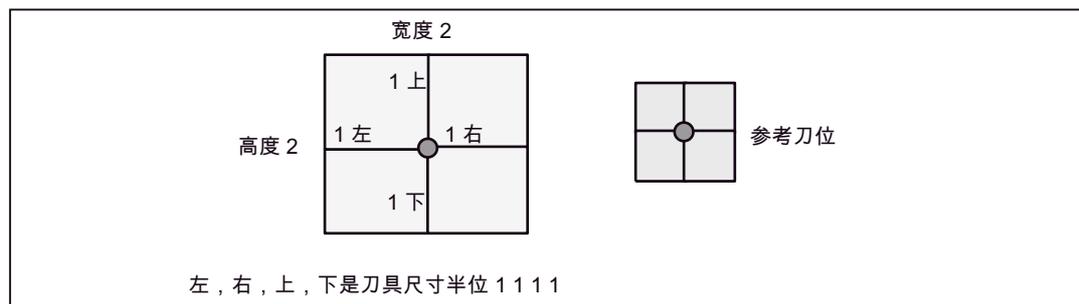


图 6-7 一般刀位类型

参数化合适的刀位类型：

- 高度 (h): 2, 宽度 (b): 2
- 左 (l): 0, 右 (r): 0
- 上 (o): 0, 下 (u): 0

链刀库过大的刀位类型

带有尺寸 2 2 1 1 的刀具在链刀库中通过从一般刀库位置起右和左占用一个附加的半位。

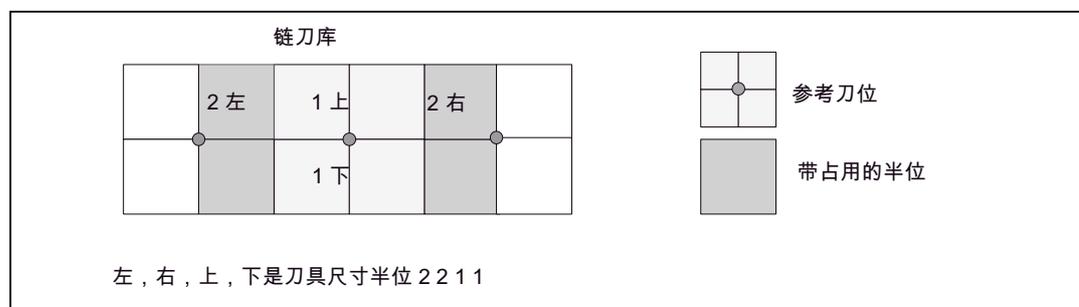


图 6-8 链刀库过大的刀位类型

参数化合适的刀位类型:

- 高度 (h): 2, 宽度 (b): 4
- 左 (l): 0, 右 (r): 0
- 上 (o): 0, 下 (u): 0

平面刀库过大的刀位类型

带有尺寸 2 2 2 2 的刀具在平面刀库中通过从一般刀库位置起在每个方向上占用一个附加的半位。

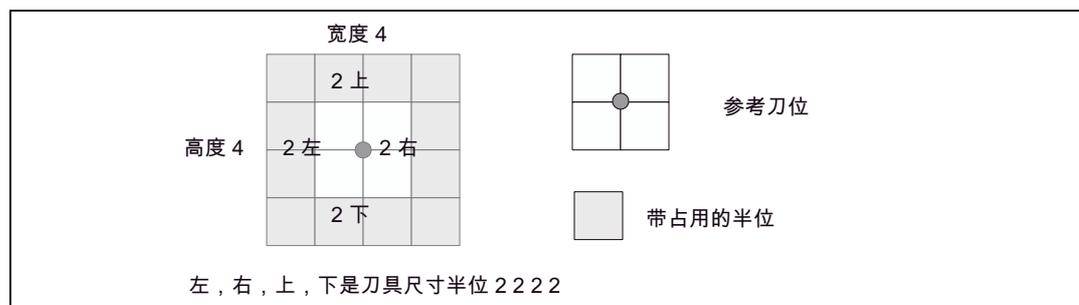


图 6-9 平面刀库过大的刀位类型

参数化合适的刀位类型:

- 高度 (h): 4, 宽度 (b): 4
- 左 (l): 0, 右 (r): 0
- 上 (o): 0, 下 (u): 0

平面刀库带有空闲半位的过大的刀位类型

带有尺寸 2 2 2 2 的刀具在平面刀库中通过从一般刀库位置起在每个方向上占用一个附加的半位。

此外不要使用角上的半位。

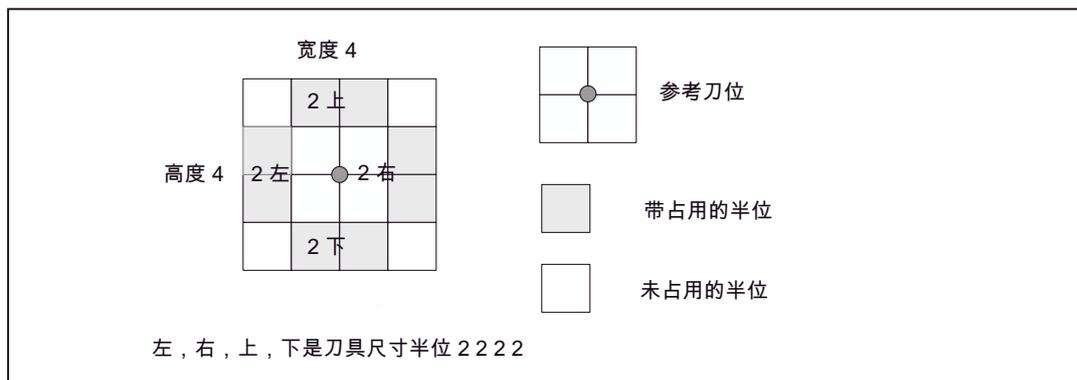


图 6-10 平面刀库带有空闲半位的过大的刀位类型

参数化合适的刀位类型:

- 高度 (h): 4, 宽度 (b): 4
- 左 (l): 1, 右 (r): 1
- 上 (o): 1, 下 (u): 1

未占用的半位通过参数左、右、上和下作为“空闲”定义。

6.1.6 示例：这样分配刀位类型一个级别

刀位类型级别

为了取消根据刀位类型刚性划分刀库位置，刀位可以按照升序布置引入一个“级别”。在一个 TO 单元内可以存在多个这样的级别。但是一个刀位类型只允许在一个这样的级别中。不允许：例如 $A < B$ 和 $A < C$ 或 $A < E$ 和 $B < E$ 。

通过一个级别可以确定，如果“较小”刀位类型不再可供，应插入在一个“小”刀位类型上的刀具也可以插入在一个“较大”刀位类型上。

如果刀具要切换入刀库中，则刀位类型确定哪些刀位可供。该刀位有一个级别，根据该级别进行刀位给定。

应用场合

带有刀位类型 A 的刀具存放在刀库中或者要查找带有类型 A 的空闲刀位。

适用下列刀位类型级别： $A < B < C$ 。

首先检查要查找的刀库是否有带有类型 A 的刀位。如果没有，则继续查找类型 B 或 C 的刀位。

示例 1:

存在的刀位类型： A, B, C

级别确定： $A < B, B < C$

得到总的级别 $A < B < C$

示例 2:

存在的刀位类型： A, B, C, D, E

级别确定： $A < B, B < D, C < E$

1. 级别： $A < B < D$

2. 级别： $C < E$

顺序

输入新的刀位类型：

1. 按下软键“新建”。
2. 输入名称（最大 32 个字符）。如果存在，立即显示 patm_xx.ini 中的显示文本。
3. 选择形式类型
4. 在半位中输入刀位类型高度和宽度。根据形式类型，如左，右，上，下（未占用的半位）。
5. 输入或者选择邻近刀位观察
6. 在视图中用光标键设置参考刀位的位置

7. 用软键“确定”保存

删除刀位类型:

- 用软键“删除”来删除选择的刀位类型。这只能在还没有分配刀库情况下。

建立级别:

1. 没有选择刀位类型（左名称）（在示例刀位类型 A）。
2. 选择较大的刀位类型（右名称）（在示例刀位类型 B）。
3. 用软键“建立级别”建立级别。图中 1-9 的级别名称与较大的刀位类型名称相符合，并显示在级别栏中（在示例 B）。

删除级别:

- 用软键“删除级别”删除“级别:”栏中所选的级别。

6.1.7 刀库配置

建立刀库配置

对于配置刀具管理，每一个 TO 单元只能有一个共同的刀库配置。一个刀库配置可以由一个或多个真正的刀库组成。因为一个 TO 单元可以分配多个通道，因此可同时为这些通道提供该刀库配置。

在该画面中为所选的刀库定义（新建）所需的数据或显示已存在的数据。

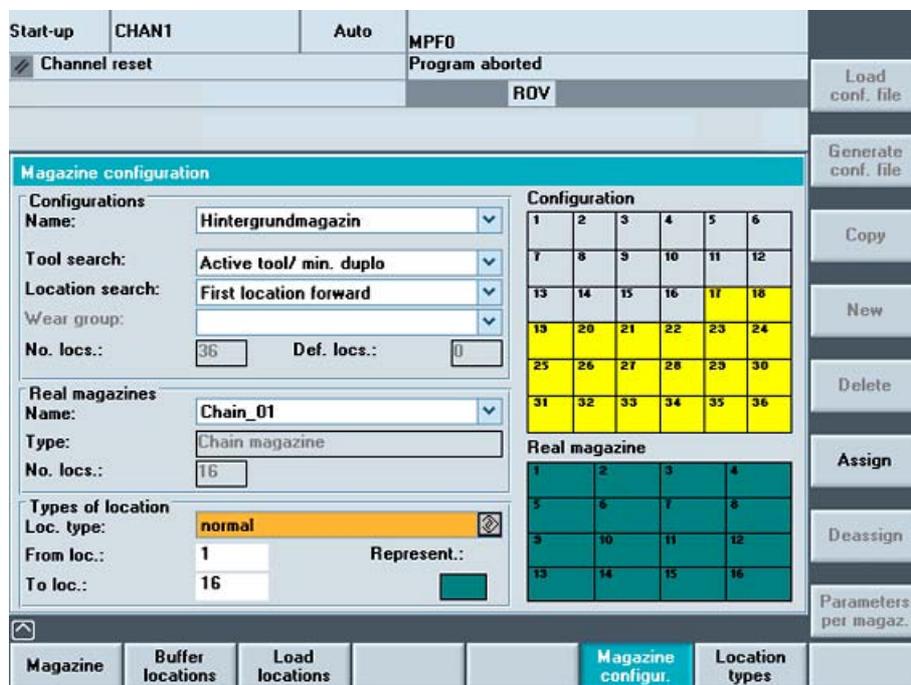


图 6-11 刀库配置

配置

名称	刀库配置名称（最大 32 个字符）。
刀具查找	从配置中接受（预设：设置用于 \$TC_MAP10=0，NCK 使用的值来自 \$TC_MAMP2）。
	激活的刀具/最小 Duplo 最短位移
	激活的刀具/最小 \$TC_TP10
	监控：最小实际值
	监控：最大实际值
	对于刀位编码可以多种（预设）选择或者固定选择。 该画面用软键“取消”或“确定”离开。
空位查找	向前第一个刀位 / 向前当前刀位 向后最后一个刀位 / 向后当前刀位 对称的当前刀位
磨损 相关（磨损相 关）	刀具状态不改变 有效改变刀具状态
刀位数：	配置中刀位总数（所有配置的刀库）
定义的刀位：	配置中刀位总数，为刀位类型分配该配置。

真正的刀库

名称：	所选的真正的刀库名称
类型：	刀库类型
数目 刀位：	所选的刀库刀位数

刀位类型

刀位类型：	所选的刀位类型名称
由刀位：	1. 要定义的刀位
至刀位：	最后要定义的刀位

软键“每个刀库参数”：

前图“刀库配置”中接受已进行的设置。

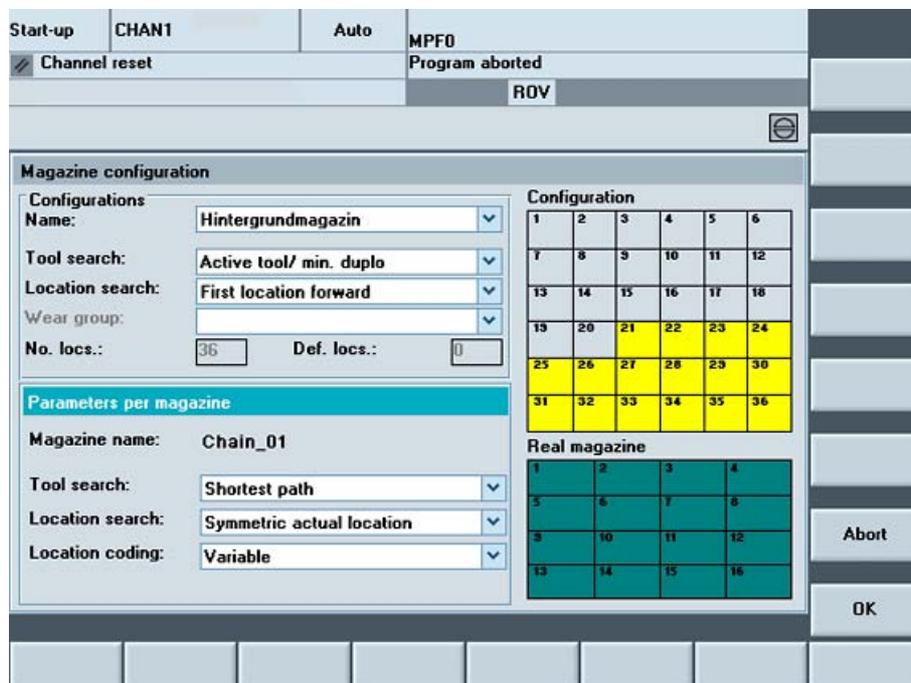


图 6-12 设置每个刀库参数

6.1.8 这样设置和装载一个配置

建立新的刀库配置

画面“配置”和“真正的刀库”显示全部配置的刀位分配或所选刀库。

1. 按下软键“新建”。
2. 输入名称，例如示例_文件（最多 32 个字符）
3. 选择用于刀具查找，空位查找和磨损相关的选择菜单并进行相应选择。
4. 按下软键“确定”（建立刀库配置）。

说明

软键“分配”和“脱开”总是与光标所在的选择栏有关：

- 刀库
 - 刀位类型
-

工作流程

分配/脱开真正的刀库：

1. 选择真正的刀库并按下软键“分配”。这个真正的刀库必须接受在刀库配置中。
2. 在每次分配后刀库位置总数在配置中相应更新。
3. 用软键“脱开”可以再次从刀库配置中触发一个刀库。

分配/脱开刀位类型：

1. 选择刀位类型。输入要分配刀位类型的刀库位置。
2. 示例：“由刀位：“1”，至刀位：“10”。
3. 按下软键“分配”。定义的刀位以这种刀位类型的颜色显示。
4. 用软键“脱开”可以删除一个相关的刀位类型分配。

创建配置文件：

- 按下软键“创建配置文件”。创建一个 INI 文件，该文件以后可以装载到 NCK 中。

复制配置

1. 按下软键“复制”。
2. 输入新的名称并用软键“确定”确认。

删除刀库配置：

1. 将光标停留在配置的选择栏上，用软键“删除”可以删除所选的配置。
2. 装载刀库配置
3. 用软键“装载配置文件”可以如图 1—16 进行“装载刀库配置”。

装载刀库配置：

- 这里通过软键“装载”将之前创建的 INI 文件装载到 NCK 中。这针对专用通道，此时每个 TO 单元仅可以一个配置。
(即 TO 单元 1 分配通道 1, 2, 3 和 6, 装载到通道 1 的配置也自动适用于通道 2, 3 和 6。)

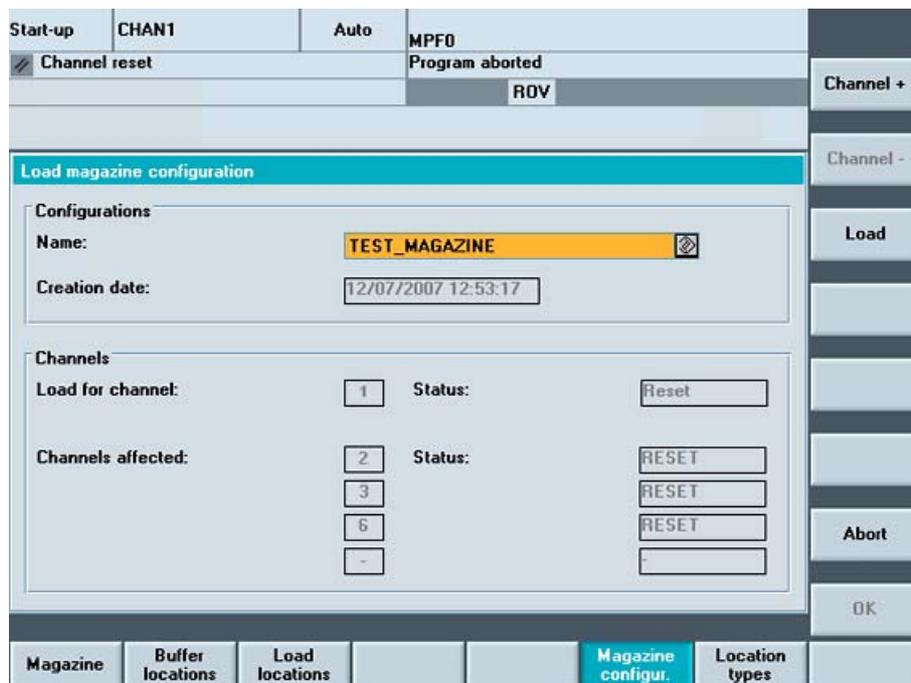


图 6-13 装载配置文件

6.2 配置文件中的设置

6.2.1 配置文件 paramtm.ini

配置

在目录“user”或“oem”中针对提供的 ini 数据将用户专用的差异保存。这里仅保存操作界面外观的变化，这可以通过 HMI 操作界面上的设置自动进行。

所有 WZV 操作界面描述的数据保存在文件 `..\user\paramtm.ini` 中。为了编辑文件，选择“调试”→“HMI”→“编辑器”。文件 `paramtm.ini` 也可以在外部计算机上建立并复制到目录 `..\user` 中。

在“`language\patm_xx.ini`”中编程国家语言相关的部分。此时国家语言标识为“xx”，例如 `gr` 代表德语，`uk` 代表英语。其他语言：参见表格 2-2。

附加功能

在刀库，刀具和工作补偿列表和刀具详细信息中另外提供下列功能：

- 参数化列表中的位图
- 列表中的刀具名称和 D 号码可更改
- 带有多行的新刀库列表
- 刀具的任务准备
- 列表图中的刀具状态位“预警极限”可更改
- 列表图中的刀具状态位“卸载标记”和“装载标记”和详细信息可更改
- 新的刀具类型：
 - 550 成形车刀
 - 700 切槽锯
 - 711 棱探头
 - 720 可定向测量探头
 - 730 挡块

- 如果在 NCK 中设置了适配器数据，则刀库列表可转换传输描述或者不转换描述（ETC 条上的软键）。设置在 paramtm.ini 中实现，段 [TMMODES] 带有条目 START_MAGLIST_TRANSFORMED
- 从 paramtm.ini 用于工具箱，代码载体，SINCOM 的状态位标记摘录：
 - ; 刀具状态：
如果从 NCK 去除一个刀具并传输到一个外部媒介上（工具箱，代码载体，SINCOM），则可以通过下列屏幕规定要保存哪些刀具状态位。
 - ;代码载体：
由于标准转换文件 wkonvert.txt 用于刀具状态 1 字节已输入，写入在代码载体上最多 92，CODECARRIER_TOOLSTATE_MASK 获得一个默认值 92。如果扩展 CODECARRIER_TOOLSTATE_MASK 值，必须相应匹配 wkonvert.txt 中对话框变量 T9 的大小。

参见

调试代码载体 (页 274)

6.2.2 文件 paramtm.ini 的结构

目录

目录 ...\user\paramtm.ini

说明

参数化刀具管理的 mmc2\paramtm.ini 包含以前各个作为注释的条目的信息。由于一个新的条目超出了文件长度临界极限（约 63 kB），必须去除几乎所有的注释。

paramtm.ini 的版本（带有注释）位于 mmc2\paramtm.txt 中。

超出或者读入参数时其他错误也和以前一样记录在文件 ...\user\paramtm.out 中。

通过一个分号 “;”可以在条目后插入一条注释。

文件结构

[ACCESSLEVEL]

SKAVTIVTM=7

; 在应用程序 PARAM 中激活的 TM

SKMGLIST=7

; 表示刀库列表

SKTLLIST=7

; 表示刀具列表

SKACLIST=7

; 显示刀具补偿列表

SKTOOLLOAD=5

; 装载刀具

SKTOOLUNLOAD=5

; 卸载刀具

SKTOOLMOVE=7

; 在刀库中移动刀具

SKSETTINGS=4

; 设置 SK

SKFILFCT=4

; SK 文件功能

SKNXTCHAN=7

; 激活 SK 下一个通道

SKMAGCONF=4
; 配置刀库

SKTOOLCAT=7
; 刀具目录

SKTOOLCAB=7
; 刀具箱

SKSINCOMLD=7
; 装载 SINCOM 刀具 (如果代码载体已安装)

SKTORSINCOM=7
; SINCOM 的代码载体功能或刀具装载
; (如果代码载体已安装)

SKMGLREPR1=7
; 描述 1_刀库列表

SKMGLREPR2=5
; 描述 2_刀库列表

SKMGLREPR3=5
; 描述 3_刀库列表

SKNCTOOLDATA=7
; 读取 DC 或文件管理的刀具数据
; (如果 ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=True)

SKNCTOOLED=7
; 刀沿数据

SKNCTOOLSUPV=7
; 超视数据

SKNCTOOLDL=7
; DL 数据

SKNCTOOLGRIND=7
; 磨削数据

SKNCDETAILS=7
; 读取 NC 的刀具数据
; (如果 ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=True)

SKNCNEWTOOLED=6
; 在 NV 中设立新的刀沿

SKNCDELTOOLED=6
; 删除 NC 中的刀沿

SKNCDELTOOL=5

; 删除 NC 中的刀具
SKTRAFO=7
; Toggle 转换 / 未转换显示边缘数据
SKCHECKACTIVATE=6
; D 检查和激活
SKMGBUFFER=7
; 显示缓冲器
SKMGFIND=7
; SK 查找和定位。
定位
SKMGLISTPOS=7
; SK 下一个刀库
SKMGNEXT=7
; 设立 NC 中的刀具
SKTLNEWTOOL=6
; 描述 1_ToolList
SKTLLREPR1=7
; 选择描述 2_ToolList
SKTLLREPR2=5
; 描述 3_ToolList
SKTLLREPR3=5
; 刀位查找, 用户定义 1
SKFINDPL1=7
; 刀位查找, 用户定义 2
SKFINDPL2=7
; 刀位查找, 用户定义 3
SKFINDPL3=7
; 刀位查找, 用户定义 4
SKFINDPL4=7
; 刀位查找
SKFINDPL=7
; 刀位查找, 刀具装载列表
SKFINDPLACE=7
; 许可 SK 当前刀位
SKACTPLACE=7
; 装载刀具状态中 SK 刀具数据
SKLDTOOLDAT=7

```

; 装载一个刀库配置
SKCONFLOAD=4
; 描述 1_ActList
SKACLREPR1=7
; 描述 2_ActList
SKACLREPR2=7
; 描述 3_ActList
SKACLREPR3=7
; SK 删除 D 号码
SKDZERO=7
; SK 根据 D 号码查找
SKDFIND=7
; SK 过滤器列表
SKBATCH=7
SKBFILTER1=7
; SK 过滤器 1
SKBFILTER2=7
; SK 过滤器 2
SKBFILTER3=7
; SK 过滤器 3
SKBFILTER4=7
; SK 过滤器 4
SKBFILTER5=7
; SK 过滤器 5
SKBFILTER6=7
; SK 过滤器 6
SKBMAGFILTER=7
; SK 刀库选择（用于过滤器）
SKBATREACT=7
; SK 批处理功能“再次激活”
SKBATTOCABIN=7
; SK 批处理功能“在箱中”
SKBATDELTOOL=7
; SK 批处理功能“删除”
SKBATUNLOAD=7
; SK 批处理功能“卸载”

```

SKBFILTERACT=7
; SK 批处理功能“更新过滤器”
SKBATLOAD=7
; SK 批处理功能“装载”
SKBATLIST=7
; 软键用于控制任务处理
ChangeToolTypeWithoutConfirmation=-1
ChangeToolSizeAndToolplace_spec=-1
READ_GUD_LUD=7
WRITE_ZOA=7
READ_SYSVAR=7
EDIT_VIEW=7

[DETAILS]

TOOLBASE_COL1=20
TOOLBASE_COL2=20
TOOLBASE_COL3=20
TOOLBASE_COL4=20

[DEFAULT SETTINGS]

; 在设立刀具时预设:
; 参见 paramtm.ini 或 paramtm.txt, 在路径 mmc2 下
; 刀具列表: 装载, 数据直接输入在列表中:
; 0=预设必须根据缺失的输入
; 通过画面“刀具数据”确认, 只要
; 需要。
; 1=在未确认情况下接受预设
; (刀具识别号除外)
; 2=在未确认情况下接受预设
; (包括刀具识别号)
DEFAULT_WITHOUT_CONFIRM=0
; 半位: 从 1 至 7
TOOLSIZE_LEFT=1
; 半位: 从 1 至 7
TOOLSIZE_RIGHT=1
; 半位: 从 1 至 7

```

TOOLSIZE_UPPER=1
;半位: 从 1 至 7
TOOLSIZE_DOWN=1
;刀具类型, 从 100 至 1000
TOOLTYPE=120
;D 号码: 从 1 至 32000
TOOLDUPLO=1
;识别号: 最大长度 27
TOOLIDENT=NEU
;添加值, (默认: 0):
;1=激活的刀具
;2=已许可
;4=已禁用
;8=已测量
;16=达到预警极限
;32=在更换
;64=固定刀位编码
;128=已使用
;256=中间存储器中的刀具
;512=已禁用, 不考虑(由于 PLC)
;1024=外面(卸载)
;2048=内部(装载)
;4096=标准刀具)一直在 NCK 中)
;8192=
;16384=
TOOLSTATE=0
;定义的刀位类型索引
TOOLPLACESPEC=1
;0=没有监控(默认)
;1=根据时间监控
;2=根据件数监控
TOOLMONITOR_MODE=0
;刀具查找, 更换刀具编号($TC_TP10)
;0 ... 32000
TOOLSEARCH_MODE=0
;在设立时预占用磨削专用的刀具数据:

```

```

; 如果机床以英寸/毫米转换
; ($MN_CONVERT_SCALING_SYSTEM=1) 处理,
; 必须规定长度单位!!!
; 下列默认值 (TOOLGRIND..., 当
; 与长度单位相关), 相关的基本
; 长度单位已规定:
; 0 = mm (默认)
; 1 = 英寸
TOOLGRIND_Default_Length_Unit=0
; 主轴号码 (如 $TC_TPG1)
TOOLGRINDspinNoDress=1
; 链接规定 (如 $TC_TPG2)
TOOLGRINDconnectPar=1050629
; 1050629 二进制: 0000 0000 0001 0000 0000 1000 0000 0101
; Bit0=1 = 类型
; Bit2=1 = Geo-L1
; Bit11=1 = Wear-L1
; Bit20 =1 = Base-L1
; 最小写半径 (如 $TC_TPG3)
TOOLGRINDminToolRadius=0
; 最小写宽度 (如 $TC_TPG4)
TOOLGRINDminToolWide=0
; 当前砂轮宽度 (如 $TC_TPG5)
TOOLGRINDactToolWide=0
; 最大砂轮转速 (如 $TC_TPG6)
TOOLGRINDmaxRotSpeed=0
; 最大砂轮圆周速度 (如 $TC_TPG7)
TOOLGRINDmaxTipSpeed=0
; 斜砂轮的倾斜角度 (如 $TC_TPG8)
TOOLGRINDinclAngle=0
; SUG 补偿参数 (如 $TC_TPG9)
TOOLGRINDparamNrCCV=3

```

[TMMODES]

```

; 0=刀具未自动删除,
; 如果已卸载 (仅刀具列表) (预设)。

```

; 1=如果已卸载，刀具自动删除
 ; (仅刀具列表)
 ; DELETE_TOOL_ON_UNLOAD=0
 ; 0=刀具类型之外的刀沿参数未
 ; 处理 (默认)
 ; 1=处理刀具类型之外的刀沿参数
 ; (如果不等于 0)
 EDGE_PARAMS_OUT_OF_TOOLTYPE=1
 ; 0=显示： 刀具尺寸 左，右，上，下 (预设)
 ; 1=显示： 左，右
 SHOW_TOOLSIZE_ONLY_LEFT_RIGHT=0
 ; 刀具尺寸显示：
 ; True=显示 (默认)
 ; False=不显示
 ; 当 SHOW_TOOLSIZE_ONLY_LEFT_RIGHT = 0 时才使用
 ; (或预设)
 SHOW_TOOLSIZE_COMPONENTS=left:=True, right:=True, top:=True,
 bottom:=True
 ; 功能“D 检查激活”涉及：
 ; -1=所有刀具，带有与主轴/刀架的距离关系
 ; (默认)
 ; 1=仅当前刀具
 DCHECK_ACTIVATE=-1
 ; 功能“D 检查激活”可以自动进行
 ; 当带有工作补偿的列表打开情况下
 ; False=功能仅通过软键激活 (默认)
 ; True=功能自动进行，如果列表带有
 ; 工作补偿已打开
 DCHECK_AUTO_ACTIVATE=False
 ; DCHECK_AUTO_ACTIVATE_MODE 才使用，
 ; 当 DCHECK_AUTO_ACTIVATE=True 情况下
 ; 0=功能“激活 D 检查”自动进行，
 ; 当工作补偿列表打开情况下，也在基本画面中
 ; (包含 SK R-参数) (默认)
 ; 1=功能“激活 D 检查”自动进行，
 ; 当工作补偿列表打开情况下，但不在

;基本画面中 (包含 SK R-参数)。 相同特性如
; 执行 DCHECK_AUTO_ACTIVATE_MODE
;(HMI-高级 06.03.19 之前版本)。
; 中间存储器刀位显示:
; DB: 原始名称由数据库中的刀库配置
; 没有语言相关的文本
; DLL: 名称 = 来自语言-DLL + 索引的文本 (默认)
; 举例: 主轴 1, 语言相关的文本
NameOfBufferPlaceFrom=DB
; 用于显示功能“设立刀沿”和“删除刀沿”
; (仅对于多行显示可以) 在主菜单中
; “刀具 / 刀库列表”中软键“刀具详细信息”可以
; 由新的软键“数据管理”替换。
; 软键“刀具详细信息”用相同的
; 功能存储在软键“数据管理”后。
; **False**=“刀具详细信息”保持激活 (默认)
; **True**=“数据管理”已激活
ACTIVATE_EDGE_MANAGEMENT_IN_LISTS=False
; 当 NCK 中的适配器数据激活时,
; 刀库列表可以作为转换的或未转换的
; 列表 (ETC 菜单中软键) 显示。 显示类型可以
; 预设, 用于启动 HMI。
; **False** = 未转换 (默认)
; **True** = 已转换
; **START_MAGLIST_TRANSFORMED=False**
; 通过软键“刀具管理”在主菜单“参数”中
; 可以是一个设置的列表下的分支。
; **0** = 标准列表取决于各个 NCK 版本 /
; 机床数据 (默认)
; **1** = 刀库列表
; **2** = 刀具列表
; **3** = 工作补偿列表
START_LIST=0
; 考虑设置英制/公制, 涉及到代码载体
; !! 注意: 这里描述的设置仅
; 相关于, 在 NC 中 MD \$MN_CONVERT_SCALING_SYSTEM=1

; 设置情况下。
; DATABASE_LENGTH_UNIT 条目才分析, 在
; 刀具数据库中没有输入单位情况下,
; 即在标准情况下!
; 涉及到一个老的 NC, 不带转换功能
; 英制/公制或 \$MN_CONVERT_SCALING_SYSTEM=0,
; 则 NC 不带英制/公制转换情况下工作。基于此
; 原因, 相关代码载体也不进行转换!!!
; -1 = 没有确定用于英制/公制 相关代码载体 (默认)。
; 设置 \$MN_CONVERT_SCALING_SYSTEM=1 在 NC 中表示,
; 要通过英制/公制转换工作。
; 为此用户必须确定要以哪些单位
; 在代码载体上存在或写入数据。
; 因为这通过设置 -1 不实现, 所有软键
; 用于代码载体已锁止。
; 0 = mm。由此相关数据的数据保存
; 要以单位 mm 保存在代码载体上
; 或者以此实现。如果 NCK 中设置“英寸”, 则
; 所有软键锁止, 启动代码载体功能。
; 1 = 英寸。由此相关数据的数据保存
; 相关以单位 mm 在代码载体上进行
; 或者以此实现。如果 NCK 中设置“mm”, 则
; 所有软键锁止, 启动代码载体功能。
DATABASE_LENGTH_UNIT=-1
; 考虑设置英制/公制, 涉及到代码载体
; !! 注意: 这里描述的设置仅
; 相关于, 在 NC 中 MD \$MN_CONVERT_SCALING_SYSTEM=1
; 设置情况下。
; 涉及到一个老的 NC, 不带转换功能
; 英制/公制或 \$MN_CONVERT_SCALING_SYSTEM=0,
; 则 NC 不带英制/公制转换情况下工作。基于此
; 原因, 相关代码载体也不进行转换!!!
; -1 = 英制/公制已忽略 (默认)。数据交换
; 在代码载体和 NCK/HMI 之间进行, 不用考虑
; 英制/公制。特性和以前一样。
; 0 = mm。将来所有相关的数据以 mm 值

; 写入在代码载体上。
; 将来代码载体和 NC 之间的数据传输
; 将相应考虑这些。
; 1 = 英制。将来所有相关的数据以英寸值
; 写入在代码载体上。
; 将来代码载体和 NC 之间的数据传输
; 将相应考虑这些。
CODECARRIER_LENGTH_UNIT=-1
; 刀具状态： 如果将一个刀具从 NCK 去除并
; 传输到一个“外部”媒介上（工具箱，代码载体，SINCOM），
; 则可以通过下列屏幕规定，哪些
; 刀具状态位要保存。
; 代码载体： 由于标准-wkonvert.txt 用于刀具状态
; 输入 1 字节，以前在代码载体上最大写 92
; 获得 CODECARRIER_TOOLSTATE_MASK
; 值 92 (预设)。
; 如果扩展 CODECARRIER_TOOLSTATE_MASK 值，
; 必须对 wkonvert.txt 中的对话框变量 T9 的尺寸
; 相应匹配。
; 1=激活的刀具
; 2=已允许
; 4=已禁用
; 8=已测量
; 16=达到报警极限
; 32=在交换
; 64=固定位置编码
; 128=已使用
; 256=缓存器中刀具
; 512=已禁用，忽略 (由于 PLC)
; 1024=外部 (卸载)
; 2048=内部 (装载)
; 4096=一般刀具 (一直在 NCK 中)
; 8192=
; 16384=
; 默认是 4828 (4+8+16+64+128+512+4096)，
; 用于代码载体 92 (4+8+16+64)

```

CABIB_TOOLSTATE_MASK=4828
SINCOM_TOOLSTATE_MASK=4828
CODECARRIER_TOOLSTATE_MASK=92
TOOLSEARCH_TC_TP10_FROM_DB=0
; 刀具查找: 处理 $TC_TP10 连同
; 刀具箱/-目录
; 0 = (预设) 如果一个刀具箱的刀具
; 装入 NCK 中, 则值“更换刀具编号 ($TC_TP10)”
; 不传输到 NCK 中。
; ($TC_TP10) 在 NCK 中设置为 0。
; 1 = 值“更换刀具编号 ($TC_TP10)”由
; 刀具箱传输到 NCK 中并显示在画面
; 刀具目录/刀具箱中。
    
```

[General]

```

; 设置“当前刀具管理数据
; 操作界面写入在 NCDDE-变量中, 如果一次更换
; 在 WIZARD 画面上进行或按下 WIZARD 软键情况下”:
; 所有设置必须在一个单独行内通过名称-
; 参数设置。
; 通过值“True”打开一个选项并
; 通过值“False”关闭一个选项,
; 使得行中缺失名称-参数。
; “EnableAllTogetherWriteToNcdde := True”: 所有数据在一个
; 单独的 NCDDE-变量中
; “EnableSingleWriteToNcdde := True”: 每个数据一个自身的
; NCDDE-变量
; 两种设置可以同时激活。
; 如果这两个选项都没有激活, 则不
; 写入 在 NCDDE-变量中。
; “WriteChangesWhenStateChanged := True”: 数据在每个
; 软键确认时写入, 不仅对于 WIZARD-软键。
HMICurDataInterface = EnableAllTogetherWriteToNcdde := True,
EnableSingleWriteToNcdde := True,
WriteChangesWhenStateChanged := False
; 应用 $MM_WRITE_TOA_FINE_LIMIT und $MM_USER_CLASS_WRITE_FINE
    
```

; 在几何值上和刀沿数据基本值

UseFineLimitForToolGeoAndAdapt=False ; (预设)

; UseFineLimitForToolGeoAndAdapt=True

; 1: 读入语言相关的 INI-文件 (language\patm_xx.ini)

; (默认)

; 0: 不读取

ReadLanguageIni=1

SearchPlaceMethod=NoInternalTool ; Don't modify!

; Tool details forms: colors for mixed adapter transformed/
untransformed display:

; Hex values, 8 characters per color(SSBBGGRR where SS=System,
BB=Blue, GG=Green, RR=Red)

; 4 值用于:

TransformedText, TransformedBackground,
NotTransformedText, NotTransformedBackground

DetailsMixedTrafoColors=WinTxt, li-brown,WinTxt ,li-blue

DetailsMixedTrafoColors =80000008,00008080,80000008,00FFFF00

; 允许显示和编辑全部 3 len 参数 L1 L2 L3 在

; 几何切削边缘, 磨损切削边缘, SC, EC 独立于刀具类型和

; 独立于 \$SS_TOOL_LENGTH_CONST 和 \$SS_TOOL_LENGTH_TYPE

; 在刀具管理列表, 详细信息形式, 箱和目录中。

AlwaysAllowL1L2L3IO=False ;默认

; AlwaysAllowL1L2L3IO=True

; 允许改变刀具状态位 8 (至少重要位是位 1)

; (TC_TP8_8 "ToolState 已使用, 刀具正在使用") 通过 GUI (列表
和详细信息形式) 用于 NCU 刀具。

; 默认: FALSE

AllowChangeOfTC_TP8_8 = False ; 默认

; AllowChangeOfTC_TP8_8 = True

[GeneralSettingsForMagAndToolList]

ToolState_Lang_12345678=<Empty> ; 使用语言-DLL

; 当 MagPlaceState_Lang_12345678 和 ToolState_Lang_12345678

; 这里未定义或同"<Empty>", 则

; 语言相关的值在刀库和刀具列表中由 pa_xx.dll

; 显示。

```

; 当这里设置值并在段"[General]"中条目
; "ReadLanguageIni" = 1, 则文件中文本
; mmc2\language\patm_gr.ini, user\language\patm_gr.ini etc. 以同样
; 段如此查找。 条目名称, 该条目在
; 语言相关的文件中使用, 则条目值
; 来自 paramtm.ini。
; 如果在语言相关的文件中找到一个条目,
; 则它作为文本使用。
; 如果未找到或者找到为"...=<Empty>",
; 则使用文件 paramtm.ini 中的值作为文本。
; 8 字符在 MagPlaceState_Lang_12345678 和
; ToolState_Lang_12345678 符合刀库位置 8 状态
; 和刀具状态, 且作为刀位和刀具状态值
; 描述在刀库列表和刀具列表中。
; 举例: ToolState_Lang_12345678=12345678_ToolState_Lang
MagPlaceState_Lang_12345678=12345678_MagPlaceState_Lang
; use patm_*.ini
; 在每个列表画面中可以显示新的刀库位置状态位
; 。 参数化在下列段中实现:
"[1_MagList]", "[2_MagList]", "[3_MagList]"
"[1_ToolList]", "[2_ToolList]", "[3_ToolList]"
"[1_ActList]", "[2_ActList]", "[3_ActList]"
; 条目根据例如:
12=TC_MPP4_9, 1, TC_MPP4_9 ;PlaceStatus Left,
; 在左半位保留
13=TC_MPP4_10, 1, TC_MPP4_10 ;PlaceStatus Right,
; 在右半位保留
14=TC_MPP4_11, 1, TC_MPP4_11 ;PlaceStatus Top,
; 在上半位保留
15=TC_MPP4_12, 1, TC_MPP4_12 ;PlaceStatus Bottom,
; 在下半位保留
16=TC_MPP4_13, 1, TC_MPP4_13 ;PlaceStatus Bit 13 (1 至 16)
17=TC_MPP4_14, 1, TC_MPP4_14 ;PlaceStatus Bit 14 of 1 至 16
18=TC_MPP4_15, 1, TC_MPP4_15 ;PlaceStatus Bit 15 (1 至 16)
19=TC_MPP4_16, 1, TC_MPP4_16 ;PlaceStatus Bit 16 (1 至 16)
; 举例: MagPlaceState_Lang_12345678=12345678_MagPlaceState_Lang
    
```

; MagPlaceState_Lang_12345678=<Empty>
; 用于字母列表栏:
; 字符宽度在“twips”中。大概的栏宽度
; 通过这里输入值的乘积得出
; 通过符号数从栏参数化
ColumnWidthTwipsPerAlphaCharacter=140
; 用于数字列表栏:
; 字符宽度在“twips”中。大概的栏宽度
; 通过这里输入值的乘积得出
; 通过符号数从栏参数化
ColumnWidthTwipsPerNumericCharacter=100
; 刀库列表中数据元素数目或者
; 刀具列表中数目通过内部数据查询。
; 范围： 1 至 27，默认 18。
; 数据查询速度从版本 P4.3.8 起
; 得以改善。
; 在内部数据查询期间没有
; 软键确认上的反应。为此这些时间
; 不长于 1 秒。
; “NumLinesPerReq”中的值用于查询完整列表的数据
; 在后台中，当选择一个列表
; 通过软键或者在调试后。
; 列表中可视行数目用于提取
; 数据，当显示的数据在数据
; 改变后或者在列表中滚动后更新，并且
; 列表中可视行的数目小于
; NumLinesPerReq. 如果数据交换在 HMI 和 NCK 之间
; 缓慢进行（对于 NCU 810 D），则改值
; 降低到 17，以在提取数据从一个
; 完整列表中在后台达到反应时间
; 约 1 s。该设置适用于所有列表，
; 其中不进行单个设置。
; 使用单个设置是在列表包含一个较高的
; 栏数时进行。
; 在这种情况下值降低到大约 10 或 5。
; 对于一个列表中多个栏，在经常使用一个

;列表时不推荐该值，因为提取
 ;全部列表的数据时间过高，并且用户必须长时间
 ;等待显示。
 ;为了对单个列表进行单个设置，在
 ;段（例如 [2_ToolList]）中应对相应列表添加行
 ;"NumLinesPerReq"。
 ;NumLinesPerReq=27 从版本 P4.3.8 起可使用 27!
 ;至 P4.3.8 值 7 作用良好。
 ;位图画面宽度，用于当前刀具和
 ;当前刀库位置，在列表画面中。
 ;单元： 字符数； 字符宽度
 ;确定通过"ColumnWidthTwipsPerAlphaCharacter"或
 ;"ColumnWidthTwipsPerNumericCharacter"。最小值： 1，
 ;较大值： 32, 预设： 5
 ;也可以使用用户定义的画面：
 ;如果不带路径规定文件名或者通过 MMC2-路径，
 ;则位图文件在目录"user"，"oem"，
 ;"add_on"和"mmc2"下查找。；使用第一个找到结果
 ;。我们建议不要建立过大用户定义的
 ;位图。宽高比应该
 ;大致符合列表中的显示，为此
 ;描述才不会走样。
 ; Use WidthOfActBitmapsInCharacters = 7, if \$SC_WEAR_TRANSFORM <> 0
 ; and G56-Reset-Value <> TOWSTD
 WidthOfActBitmapsInCharacters = 5
 WhichActChannelText=ChannelName,4
 ;默认，显示首先 4 个通道名称字符在列表中，当前
 ;刀具指示器
 ; WhichActChannelText=ChannelNumber ;显示列表中通道号码
 ;当前刀具指示器
 ;如果需要显示更多的通道名称字符，请
 ;增加 "WidthOfActBitmapsInCharacters"。
 ;为了更好可读性修改位图或使用 lpaat.bmp, lpaatd0.bmp,
 ; lpaatd10.bmp, lpapt.bmp, lpaptd0.bmp, lpaptd10.bmp, lpaap.bmp.
 ;此外可以定义一个专门栏用于位图特定
 ;显示，将不包含任何其他数据。

; 举例:

; [3_ActList]

; ShowActToolCol = 1

; 栏号码显示位图所在

; 1= NoData, 0, Activity

; 栏 1 是一个空栏, 用于显示通道活动;

; 栏带有 0 + "WidthOfActBitmapsInCharacters";

; 栏标题文本为 "活动" 或语言相关文本;

; 用于当前刀具的位图文件名称 / DNo / DL

; 此时 D <> 0 和 DL <> 0

; ActToolBitmap = paat.bmp

; 用于当前刀具的位图文件名称 / DNo / DL

; 通过 D = 0 在刀库列表和刀具列表中。

; 这些刀沿不在工作补偿列表中

; 标记。

ActToolZeroDBitmap = paatd0.bmp

; 用于当前刀具的位图文件名称 / DNo / DL

; 以当前 DL = 0。

ActToolZeroDLBitmap = paatdl0.bmp

; 如对于 ActToolBitmap 用于编程的刀具

ProgToolBitmap = papt.bmp

; 如对于 ActToolZeroDBitmap 用于编程的刀具

ProgToolZeroDBitmap = papt0.bmp

; 如对于 ActToolZeroDLBitmap 用于编程的刀具

ProgToolZeroDLBitmap = paptdl0.bmp

; 位图文件名用于当前刀库位置

ActPlaceBitmap = paap.bmp

; 显示是否启用或禁用当前刀库

; 针对装载/卸载刀具。

ShowMagFreeLocked = False

; ShowMagFreeLocked = True

; 位图文件名称用于显示当前刀库是否

; 针对装载/卸载刀具启用。

MagFreeBitmap = magfree.bmp

; 位图文件名称用于显示当前刀库是否

; 针对装载/卸载刀具禁用。

```

MagLockBitmap = maglock.bmp
; 位图文件名称用在列表中显示: G56-reset-value =
; TOWMCS = G56-current-value
G56ResetTOWMCSEqualCurrBitmap = pemcs.bmp
; 位图文件名称用在列表中显示: G56-reset-value =
; TOWMCS = G56-current-value
G56ResetTOWWCSEqualCurrBitmap = pewcs.bmp
; 位图文件名称用在列表中显示: G56-reset-value =
; TOWSTD = G56-current-value
G56ResetTOWSTDEqualCurrBitmap = pestd.bmp
; 位图文件名称用在列表中显示: G56-reset-value =
; TOWMCS <> G56-current-value
G56ResetTOWMCSUnequalCurrBitmap = pumcs.bmp
; 位图文件名称用在列表中显示: G56-reset-value =
; TOWMCS <> G56-current-value
G56ResetTOWWCSUnequalCurrBitmap = puwcs.bmp
; 位图文件名称用在列表中显示: G56-reset-value =
; TOWSTD <> G56-current-value
G56ResetTOWSTDUnequalCurrBitmap = pustd.bmp
; 为了防止过度水平滚动, 光标自动移动
; 到最左侧栏, 列表中通过下列
; a 动作: 刀库列表: 软键“装载”和软键“启动”
; 和“取消”, 在刀具装载模式下
; 刀具列表: 软键“新刀具”。
; CursorMovesLeftmostBySomeActions=True ;默认
; CursorMovesLeftmostBySomeActions=False
; 在刀具列表形式中改变软键“下一个刀库”到软键
; “选择刀库”, 激活垂直的软键, 在一个
; 附加刀具选择状态中。
; 这有助于防止用户过度使用“下一个刀库”
; 软键, 如果多个刀库可用。
; 可以定义快捷图标软键最多至 5 个自选刀库
; 通过 TOA 使用段 [ShortcutSoftKeysForMagSelect] 在
; paramtm.ini and patm_??.ini 中。
; MagListMagSelectSoftkey=NextMag ; 默认
; MagListMagSelectSoftkey=SelectMag
    
```

; 刀库列表: 显示缓存器初始
DisplayBufferInMagList = InitialVisible
; 默认为版本 6.4.1。
; 何时访问另一个刀库在刀库列表中:
; 自动显示缓存器。
; 缓存器临时隐藏, 在用户按下
; 软键“缓存器”后。
; **DisplayBufferInMagList = InitialNotVisible**
; 6.4.1 以前版本。
; 在刀库列表中: 开始不显示缓存器。
; 在访问没有缓存器的刀库后,
; 那时也总是显示没有缓存器, 用于其他刀库。
; 缓存器蓝狮显示, 在用户按下
; 软键“缓存器”后。

[SoftKeysForMagAndToolList]

; 刀库列表和刀具列表软键文本。
; 使用国家语言相关的机械机构。
; 机械机构说明参见段;
; "[GeneralSettingsForMagAndToolList]", 条目
; "MagPlaceState_Lang_12345678" 或 "second "TC_TP2""
1_MagList=M1
2_MagList=M2
3_MagList=M3
1_ToolList=T1
2_ToolList=T2
3_ToolList=T3
1_ActList=A1
2_ActList=A2
3_ActList=A3

[FormTitles]

; 未转换显示的列表画面字体
; 数据带有语言相关的和转换相关的
; 机械机构。在转换显示时替代
; 文本代码关闭; 接受段“[TrafoFormTitles]”。

;当“ReadLanguageIni=1”在“[General]”中，则文本
 ;从语言相关的文件中读取
 ;（例如 language\patm_gr.ini），参见段“[FormTitles]”。
 ;在各种情况下列表画面的文本参数要
 ;设置，不仅用于转换的显示，也用于
 ;未转换的显示。
 ;特别注意工作补偿列表：目前
 ;该列表仅用转换的数据输出。
 ;在这种情况下使用列表画面字体
 ;来自“[TrafoFormTitles]”。

1_MagList=M1
 2_MagList=M2
 3_MagList=M3
 1_ToolList=T1
 2_ToolList=T2
 3_ToolList=T3
 1_ActList=A1
 2_ActList=A2
 3_ActList=A3

[TrafoFormTitles]

;未转换数据显示的列表画面字体
 ;带有语言相关的和转换相关的机械机构。
 ;当“ReadLanguageIni=1”在“[General]”中，则文本
 ;从语言相关的文件中读取
 ;参见段“[FormTitles]”。

M1=TM1
 M2=TM2
 M3=TM3
 T1=TT1
 T2=TT2
 T3=TT3
 A1=TA1
 A2=TA2
 A3=TA3

[SearchOfMagPlaces]

; 确定要进行的空位查找。
; 第一个空位查找软键文本，语言相关
1_SoftkeyText=EL1
; 半位，左，右，上，下
1_ToolSizeLRTB=1,1,1,1
; 空位查找刀位编号
1_PlaceTypeNo=1
; 第二个空位查找软键文本，语言相关
2_SoftkeyText=EL2
; 半位，左，右，上，下
2_ToolSizeLRTB=1,2,1,1
; 空位查找刀位编号
2_PlaceTypeNo=1
; 第三个空位查找软键文本，语言相关
3_SoftkeyText=EL3
; 半位，左，右，上，下
3_ToolSizeLRTB=2,2,1,1
; 空位查找刀位编号
3_PlaceTypeNo=1
; 第三个空位查找软键文本，语言相关
4_SoftkeyText=EL4
; 半位，左，右，上，下
4_ToolSizeLRTB=1,2,1,1
; 空位查找刀位编号
4_PlaceTypeNo=2
; 刀具 OEM 数据和刀具应用数据
; 有助于 UserDataParamIO 行：
; 刀库和刀具列表中数据格式。
; 该格式只用于显示数据，在某些情况下，
; 为了输出 HMI，刀库列表和刀具列表中的数据。
; 在 NC 中 OEM 数据格式为“浮点”。
; 通常在参数行“名称参数”中。名称
; 通过 "!=" 与值隔开。参数通过 "," 隔开
; 参数左右允许空格
; 名称，"!="，值和","。

```

; OEM 数据格式中行句法示例:
; "[ToolParams]"
; "UserDataParamIO5="
; Typ:=Float, Res:=2,
; Min:=-9999, Max:=9999 ;Comment"
; "UserDataParamIO6="
; Typ:=Int, Min:=-99, Max:=99"
; "UserDataParamIO7="
; Typ:=Int, Min:=0, Max:=1"
; "5": OEM 数据编号
; "Type:=...": 确认 OEM 数据类型;
; 默认: "浮点"
; "浮点": 浮点, 如同 NC 中使用。
; 逗号后位数取决于 MD
; $MM_DISPLAY_RESOLUTION 在 mmc.ini 中从 "res:=..."
; 在该参数行内。
; "Int": 整数 (范围: -999999999 至 999999999)
; "Res:=2": 逗号后位数,
; 当 "Typ:=Float" (浮点数)。
; 范围从 0 至 6, 默认是 $MM_DISPLAY_RESOLUTION。
; 当 n "Res:=..." 大于 $MM_DISPLAY_RESOLUTION, 则
; 仅显示位 $MM_DISPLAY_RESOLUTION。
; 分辨率, 精度: 数目
; "Min:=-9999": 对于输入: 最小值
; "Max:=9999": 对于输入: 最大值
; ";comment": 注释
; Zeilen UserDataParamName: 语言相关的机械机构
; Zeilen UserDataParamSize: 语言相关的机械机构
; Zeilen UserDataParamSizex: 语言相关的机械机构

```

[ToolParams]

```

UserDataParamName1 = TC_TPC1
UserDataParamName2 = TC_TPC2
UserDataParamName3 = TC_TPC3
UserDataParamName4 = TC_TPC4
UserDataParamName5 = TC_TPC5

```

```
UserDataParamName6 = TC_TPC6
UserDataParamName7 = TC_TPC7
UserDataParamName8 = TC_TPC8
UserDataParamName9 = TC_TPC9
UserDataParamName10 = TC_TPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Type:=Float, Res:=2
; UserDataParamIO2 = Type:=Int
UserDataParamSize = TC_TPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_TPC1_UNIT
; ...
; UserDataParamSize10=TC_TPC10_UNIT
AppDataParamName1 = TC_TPCS1
AppDataParamName2 = TC_TPCS2
AppDataParamName3 = TC_TPCS3
AppDataParamName4 = TC_TPCS4
AppDataParamName5 = TC_TPCS5
AppDataParamName6 = TC_TPCS6
AppDataParamName7 = TC_TPCS7
AppDataParamName8 = TC_TPCS8
AppDataParamName9 = TC_TPCS9
AppDataParamName10 = TC_TPCS10
AppDataParamIO1 = <EndOfList>
; AppDataParamIO1 = Type:=Float, Res:=2
; AppDataParamIO2 = Type:=Int
AppDataParamSize = TC_TPCS_UNIT
AppDataParamSize1 = <EndOfList>
; AppDataParamSize1=TC_TPCS1_UNIT
; ...
; AppDataParamSize10=TC_TPCS10_UNIT
; OEM 数据格式中行句法示例:
; "[ToolEdgeParams]"
; "UserDataParamIO5="
; Typ:=Float, Res:=2, Min:=-9999, Max:=9999 "Comment"
; "UserDataParamIO6="
```

```

; Typ:=Int, Min:=-99, Max:=99"
; "UserDataParamIO7=
; Typ:=Int, Min:=0, Max:=1"
; "5": OEM 数据编号
; "Type:=...": 确认 OEM 数据类型
; 默认: "浮点"
; "浮点": 浮点, 如同 NC 中使用。 位数
; 逗号后的位数取决于 $MM_DISPLAY_RESOLUTION
; 在 mmc.ini 从 "AufL." 在该行中。
; "Int": 整数 (范围: -999999999 至 999999999)
; "Res:=2": 分辨率, 精度: 位数
; 逗号后, 当: "Typ:=Float" (浮点数)。
; 范围从 0 至 6, 默认是 $MM_DISPLAY_RESOLUTION。
; 当 n "Res:=..." 大于 $MM_DISPLAY_RESOLUTION,
; 则仅显示位 $MM_DISPLAY_RESOLUTION。
; "Min:=-9999": 对于输入: 最小值
; "Max:=-9999": 对于输入: 最大值
; ";comment": 注释;
; 行 EdgeParamName...:
; 举例: ; EdgeParamNameLLen1=TC_DP3
; 确认文本, 在该文本下刀沿参数和
; 总补偿参数在刀具详细画面中
; 显示。
; EdgeParamName...行的顺序符合
; 画面中的顺序。
; 画面中的顺序无法通过改变
; paramtm.ini 中的顺序影响。
; 对于 EdgeParamName... 值使用符号序列
; "TC_DP..."以表明文本与
; 附属的 NCK 变量关系。 但也可以使用另一个
; 该行上的符号序列, 因为与该文本
; 参考仅定义一个参数字体, 并不
; 确定从 NCK 读取哪些数据。
; EdgeParamName... 值直接写入在画面中,
; 如果 [General] ReadLanguageIni=0 且在刀具
; 详细信息画面中未转换的描述激活。

```

; 如果 [General] ReadLanguageIni=1 且在刀具
; 详细信息画面中未转换的描述激活,
; EdgeParamName... 值用作存取密码, 以
; 从国家语言相关的 INI-文件读取该文本。
; (language\patm_xx.ini, 段 [ToolEdgeParams], 存取密码
; "TC_DP3" 在该示例中)。
; 如果在刀具详细信息画面中转换的描述
; 激活, EdgeParamName... 值用作存取密码
; 对于段 [ToolEdgeParamsTrafoTextReplace] 在 paramtm.ini 中
; 。 那里找到的值用作文本或
; 存取密码 (取决于 [General] ReadLanguageIni) ,
; 以确定刀具详细信息画面的输出文本。
; 如果一个条目为在 INI-文件中找到, 则
; 从资源文件 lanuage\pa_xx.dll 中获取该文本。
; Zeilen UserDataParamName: 语言相关的机械机构
; Zeilen UserDataParamSize: 语言相关的机械机构
; Zeilen UserDataParamSizex: 语言相关的机械机构
; 行 EdgeParamName...: 未转换的或
; 转换的机械机构和语言相关的机械机构。

[ToolEdgeParams]

; 长度 1
EdgeParamNameLLen1 = TC_DP3
; 长度 2
EdgeParamNameLLen2 = TC_DP4
; 长度 3
EdgeParamNameLLen3 = TC_DP5
; 半径 I1
EdgeParamNameRLen1 = TC_DP8
; 半径 I2
EdgeParamNameRLen2 = TC_DP9
; 半径 r1
EdgeParamNameRRad1 = TC_DP6
; 半径 r2
EdgeParamNameRRad2 = TC_DP7
; 角 1
EdgeParamNameAng1 = TC_DP10

```

;角 2
EdgeParamNameAng2 = TC_DP11
UserDataParamName1 = TC_DPC1
UserDataParamName2 = TC_DPC2
UserDataParamName3 = TC_DPC3
UserDataParamName4 = TC_DPC4
UserDataParamName5 = TC_DPC5
UserDataParamName6 = TC_DPC6
UserDataParamName7 = TC_DPC7
UserDataParamName8 = TC_DPC8
UserDataParamName9 = TC_DPC9
UserDataParamName10 = TC_DPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Type:=Float, Res:=2
; UserDataParamIO2 = Type:=Int
UserDataParamSize = TC_DPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_DPC1_UNIT
; ...
; UserDataParamSize10=TC_DPC10_UNIT
AppIDataParamName1 = TC_DPCS1
AppIDataParamName2 = TC_DPCS2
AppIDataParamName3 = TC_DPCS3
AppIDataParamName4 = TC_DPCS4
AppIDataParamName5 = TC_DPCS5
AppIDataParamName6 = TC_DPCS6
AppIDataParamName7 = TC_DPCS7
AppIDataParamName8 = TC_DPCS8
AppIDataParamName9 = TC_DPCS9
AppIDataParamName10 = TC_DPCS10
AppIDataParamIO1 = <EndOfList>
; AppIDataParamIO1 = Type:=Float, Res:=2
; AppIDataParamIO2 = Type:=Int
AppIDataParamSize = TC_DPCS_UNIT
AppIDataParamSize1 = <EndOfList>
; AppIDataParamSize1=TC_DPCS1_UNIT
    
```

; ...
; ApplDataParamSize10=TC_DPCS10_UNIT

[ToolEdgeParamsTrafoTextReplace]

; 转换的长度 1
TC_DP3 = TTC_DP3
; 转换的长度 2
TC_DP4 = TTC_DP4
; 转换的长度 3
TC_DP5 = TTC_DP5
; 转换的半径 I1
TC_DP8 = TTC_DP8
; 转换的半径 I2
TC_DP9 = TTC_DP9
; 转换的半径 r1
TC_DP6 = TTC_DP6
; 转换的半径 r2
TC_DP7 = TTC_DP7
; 转换的角 1
TC_DP10 = TTC_DP10
; 转换的角 2
TC_DP11 = TTC_DP11

[CuttEdgeSupervisionOEM]

; [CuttEdgeSupervisionOEM] 刀沿监控-OEM-数据
; 和应用数据
; [MagazineOEM] 刀库-OEM-数据和应用数据
; [MagazineLocOEM] 刀库位置-OEM-数据和应用数据
; 辅助用于行“UserDataParamIO”:
; 数据格式用于刀库列表中的数据显示和
; 刀具列表。该格式仅用于显示和
; 在这种情况下也用于数据输入。
; 在参数行中使用“Namensparameter”。
; , 名称由参数值通过 “:=” 隔开。
; 参数自己通过 “,” 相互隔开;
; 参数左右两侧为空格, “:=”,

```

; 参数值和 ";" 允许。
; OEM 数据格式中行句法示例:
; "[CuttEdgeSupervisionOEM]"
; "UserDataParamIO5=Min:=-9999, Max:=9999 ;注释"
; "5": OEM 数据编号
; "Min:=-9999": 对于输入: 最小值
; "Max:=-9999": 对于输入: 最大值
; ";comment": 注释
; Zeilen UserDataParamName: 语言相关的机械机构
; Zeilen UserDataParamSize: 语言相关的机械机构
; Zeilen UserDataParamSizeX: 语言相关的机械机构
    
```

[CuttEdgeSupervisionOEM]

```

UserDataParamName1 = TC_MOPC1
UserDataParamName2 = TC_MOPC2
UserDataParamName3 = TC_MOPC3
UserDataParamName4 = TC_MOPC4
UserDataParamName5 = TC_MOPC5
UserDataParamName6 = TC_MOPC6
UserDataParamName7 = TC_MOPC7
UserDataParamName8 = TC_MOPC8
UserDataParamName9 = TC_MOPC9
UserDataParamName10 = TC_MOPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Min:=-4, Max:=6
; UserDataParamIO2 = Min:=-10, Max:=122
UserDataParamSize = TC_MOPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_MOPC1_UNIT
; ...
; UserDataParamSize10=TC_MOPC10_UNIT
ApplDataParamName1 = TC_MOPCS1
ApplDataParamName2 = TC_MOPCS2
ApplDataParamName3 = TC_MOPCS3
ApplDataParamName4 = TC_MOPCS4
ApplDataParamName5 = TC_MOPCS5
    
```

```
AppIDataParamName6 = TC_MOPCS6
AppIDataParamName7 = TC_MOPCS7
AppIDataParamName8 = TC_MOPCS8
AppIDataParamName9 = TC_MOPCS9
AppIDataParamName10 = TC_MOPCS10
AppIDataParamIO1 = <EndOfList>
; AppIDataParamIO1 = Min:=-4, Max:=6
; AppIDataParamIO2 = Min:=-10, Max:=122
AppIDataParamSize = TC_MOPCS_UNIT
AppIDataParamSize1 = <EndOfList>
; AppIDataParamSize1=TC_MOPCS1_UNIT
; ...
; AppIDataParamSize10=TC_MOPCS10_UNIT
```

[MagazineOEM]

```
UserDataParamName1 = TC_MAPC1
UserDataParamName2 = TC_MAPC2
UserDataParamName3 = TC_MAPC3
UserDataParamName4 = TC_MAPC4
UserDataParamName5 = TC_MAPC5
UserDataParamName6 = TC_MAPC6
UserDataParamName7 = TC_MAPC7
UserDataParamName8 = TC_MAPC8
UserDataParamName9 = TC_MAPC9
UserDataParamName10 = TC_MAPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Min:=-22, Max:=24
; UserDataParamIO2 = Min:=-10, Max:=162
UserDataParamSize = TC_MAPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_MAPC1_UNIT
; ...
; UserDataParamSize10=TC_MAPC10_UNIT
AppIDataParamName1 = TC_MAPCS1
AppIDataParamName2 = TC_MAPCS2
AppIDataParamName3 = TC_MAPCS3
```

```

ApplDataParamName4 = TC_MAPCS4
ApplDataParamName5 = TC_MAPCS5
ApplDataParamName6 = TC_MAPCS6
ApplDataParamName7 = TC_MAPCS7
ApplDataParamName8 = TC_MAPCS8
ApplDataParamName9 = TC_MAPCS9
ApplDataParamName10 = TC_MAPCS10
ApplDataParamIO1 = <EndOfList>
; ApplDataParamIO1 = Min:=-22, Max:=24
; ApplDataParamIO2 = Min:=-10, Max:=162
ApplDataParamSize = TC_MAPCS_UNIT
ApplDataParamSize1 = <EndOfList>
; ApplDataParamSize1=TC_MAPCS1_UNIT
; ...
; ApplDataParamSize10=TC_MAPCS10_UNIT
    
```

[MagazineLocOEM]

```

UserDataParamName1 = TC_MPPC1
UserDataParamName2 = TC_MPPC2
UserDataParamName3 = TC_MPPC3
UserDataParamName4 = TC_MPPC4
UserDataParamName5 = TC_MPPC5
UserDataParamName6 = TC_MPPC6
UserDataParamName7 = TC_MPPC7
UserDataParamName8 = TC_MPPC8
UserDataParamName9 = TC_MPPC9
UserDataParamName10 = TC_MPPC10
UserDataParamIO1 = <EndOfList>
; UserDataParamIO1 = Min:=-42, Max:=62
; UserDataParamIO2 = Min:=-210, Max:=712
UserDataParamSize = TC_MPPC_UNIT
UserDataParamSize1 = <EndOfList>
; UserDataParamSize1=TC_MPPC1_UNIT
; ...
; UserDataParamSize10=TC_MPPC10_UNIT
ApplDataParamName1 = TC_MPPCS1
    
```

```

AppIDataParamName2 = TC_MPPCS2
AppIDataParamName3 = TC_MPPCS3
AppIDataParamName4 = TC_MPPCS4
AppIDataParamName5 = TC_MPPCS5
AppIDataParamName6 = TC_MPPCS6
AppIDataParamName7 = TC_MPPCS7
AppIDataParamName8 = TC_MPPCS8
AppIDataParamName9 = TC_MPPCS9
AppIDataParamName10 = TC_MPPCS10
AppIDataParamIO1 = <EndOfList>
; AppIDataParamIO1 = Min:=-42, Max:=62
; AppIDataParamIO2 = Min:=-210, Max:=712
AppIDataParamSize = TC_MPPCS_UNIT
AppIDataParamSize1 = <EndOfList>
; AppIDataParamSize1=TC_MPPCS1_UNIT
; ...
; AppIDataParamSize10=TC_MPPCS10_UNIT

```

参数化单个刀库，刀具和工作补偿列表

```

; 通过在下段中改变可以确定，
; 显示哪些数据在各个刀库，刀具和
; 工作补偿列表中：
; [1_MagList], [2_MagList], [3_MagList],
; [1_ToolList], [2_ToolList], [3_ToolList],
; [1_ActList], [2_ActList], [3_ActList].
; 在这些段中可以不移动
; 水平导航（“滚动”）时的数目
; （也总是可见）栏（“NrOfFixedColumns=”）和
; 确定各个栏（“1=...”, “2=...”, ...）。
; 栏编号（“=”前数）允许值在
; 1 和 1000 之间选择。列表中最大栏数
; 大约为 90，此外当列表中为 90 栏时
; 显示速度会比较缓慢，
; 用户必须水平滚动，以显示所有栏，
; 这样一般情况下不要达到极限
; 。

```

```

; 在栏编号顺序中
; 允许编号之间有空缺。
; 如果要使 mmc2\paramtm.ini 中一个预定义的栏
; 退出激活, 您可以在 user\paramtm.ini 中
; 用值 "<Empty>" 添加相应的条目。
; 为了确定列表末尾, 要规定 "...=<EndOfList>"
;。 这提高了速度, 在读取
; INI-文件 (刀具管理启动后)。
; 栏定义行句法示例:
; "2=TC_TP2,11,TC_TP2 ; WzIdent "2": 条目的编号,
; 第一个 "TC_TP2": 确定在列表栏中显示哪些 NC 数据。
; 符号序列 TC_TP2 符合一个
; NCK-变量, 参见程序说明。
; 显示的符号序列在 paramtm.txt 中描述。
; 版本 P5 中新信息:
; 当 "MultiLine=SINGLE" 位于刀库或刀具列表的一个列表定义段中,
; 则刀沿号码可以
; 通过添加 "@Ee" 特定, 此时 "e" 为
; 刀沿号码 (范围从 1 至每个刀具的最大刀沿数),
; 用于所有刀沿数据。
; 这涉及到下列数据:
; 刀沿数据 TC_DPp@Ee
; 刀沿监控数据 TC_MOPp@Ee
; OEM-刀沿数据 TC_DPCp@Ee
; 自由给定的 D-号码 TC_DPCE@Ee
; 总补偿 TC_SCPz@Ee
; 校准补偿 TC_ECPz@Ee
; 如果没有在栏中规定 "@Ee", 则
; 使用刀沿 1 数据。 该方式
; 与 P5 的前期版本兼容。 为了避免误换,
; 对于该刀沿, 将刀沿
; 号码在各个栏的标题文本中规定。
; "@Ee" 不允许规定在刀库和刀具列表中
; 带 "Multitime=MULTI" 和不带工作补偿列表。
; 在这些值中当前刀沿数据自动
; 显示。

```

; "11": 字符中蓝的大致宽度, 涉及到
 ; "[GeneralSettingsForMagAndToolList]", 条目
 ; "ColumnWidthTwipsPer-AlphaCharacter" 和
 ; "ColumnWidthTwipsPerNumericCharacter"
 ; 第二个 "TC_TP2": 栏标题文本或者文本密码
 ; 。 如果在段 "[General]" 中条目
 ; "ReadLanguageIni"="1", 则栏标题文本
 ; 在文件 mmc2\language\patm_gr.ini,
 ; user\language\patm_gr.ini 等的段
 ; "[ListColumnHeaderText]", 条目 "TC_TP2" (在本示例中)
 ; 查找。 符号序列“gr”在“patm_gr.ini”中取决于
 ; 各个语言 (参见 mmc.ini, "[LANGUAGE]", 条目
 ; "Language=...")。
 ; 如果条目在 Language\patm_gr.ini 中找到,
 ; 则它作为栏标题文本使用。
 ; 如果没有找到条目或者文本 "...=<Empty>"
 ; 被找到, 则从文件 paramtm.ini 中的值
 ; 用作栏标题文本。
 ; 如果显示转换的数据, 则栏
 ; 标题文本或其存取密码“已转换”,
 ; 在该栏中相应分配段
 ; [ListColumnHeaderTrafoTextReplace] 用于替换。
 ; (这不仅用于转换的/未转换的机械机构
 ; 也用于国家语言机械机构。)
 ; "WzIdent": ";" 导入一个注释, 在
 ; 参数行末段可以导入一个注释, 也带"/"。

[1_MagList]

MultiLine=SINGLE

NrOfFixedColumns=1

1= ToolInPlace, 3, ToolInPlace

2= TC_MPP4_1, 1, TC_MPP4_1

3= TC_MPP4_2, 1, TC_MPP4_2

4= TC_MPP4_3, 1, TC_MPP4_3

5= TC_MPP4_4, 1, TC_MPP4_4

6= TC_MPP4_5, 1, TC_MPP4_5

7= TC_MPP4_6, 1, TC_MPP4_6
 8= TC_MPP4_7, 1, TC_MPP4_7
 9= TC_MPP4_8, 1, TC_MPP4_8
 10= TC_TP2, 11, TC_TP2
 11= TC_TP1, 5, TC_TP1
 12= TC_MPP6, 5, TC_MPP6
 13= TC_TP3, 1, TC_TP3
 14= TC_TP4, 1, TC_TP4
 15= TC_TP5, 1, TC_TP5
 16= TC_TP6, 1, TC_TP6
 17= TC_TP7, 4, TC_TP7
 18= TC_TP8_1, 1, TC_TP8_1
 19= TC_TP8_2, 1, TC_TP8_2
 20= TC_TP8_3, 1, TC_TP8_3
 21= TC_TP8_4, 1, TC_TP8_4
 22= TC_TP8_5, 1, TC_TP8_5
 23= TC_TP8_6, 1, TC_TP8_6
 24= TC_TP8_7, 1, TC_TP8_7
 25= TC_TP8_8, 1, TC_TP8_8
 26= NoData, 1, <automatic extend last column>
 27= <EndOfList>

[2_MagList]

MultiLine=MULTI
 NrOfFixedColumns=1
 1= ToolInPlace, 3, ToolInPlace
 2= TC_TP2, 11, TC_TP2
 3= TC_TP1, 5, TC_TP1
 4= TC_MPP6, 5, TC_MPP6
 5= TC_TP7, 4, TC_TP7
 6= CuttEdgeNo, 1, CuttEdgeNo
 7= TC_DP1, 4, TC_DP1
 8= TC_DP3, 11, TC_DP3
 9= TC_DP6, 11, TC_DP6
 10= TC_TP3, 1, TC_TP3
 11= TC_TP4, 1, TC_TP4

12= TC_TP5, 1, TC_TP5
13= TC_TP6, 1, TC_TP6
14= TC_MPP2, 4, TC_MPP2
15= TC_ADPT1, 11, TC_ADAPT1
16= TC_ADPT2, 11, TC_ADAPT2
17= TC_ADPT3, 11, TC_ADAPT3
18= TC_ADPT4, 4, TC_ADAPT4
19= <EndOfList>

[3_MagList]

NrOfFixedColumns=1
1= ToolInPlace, 3, ToolInPlace
2= TC_TP2, 11, TC_TP2
3= TC_TP1, 5, TC_MPP6
5= TC_TP9, 1, TC_TP9
6= TC_MOP1, 7, TC_MOP1
7= TC_MOP2, 7, TC_MOP2
8= TC_MOP3, 7, TC_MOP3
9= TC_MOP4, 7, TC_MOP4
10= TC_MPP3, 1, TC_MPP3
11= TC_MPP5, 2, TC_MPP5
12= NoData, 1, <automatic extend last column>
13= <EndOfList>

[1_ToolList]

NrOfFixedColumns=1
1= NO, 4, NO
2= MagNo, 4, MagNo
3= ToolInPlace, 3, ToolInPlace
4= TC_TP2, 11, TC_TP2
5= TC_TP1, 5, TC_TP1
6= TC_MPP6, 5, TC_MPP6
7= TC_TP3, 1, TC_TP3
8= TC_TP4, 1, TC_TP4
9= TC_TP5, 1, TC_TP5
10= TC_TP6, 1, TC_TP6

11= TC_TP8_1, 1, TC_TP8_1
12= TC_TP8_2, 1, TC_TP8_2
13= TC_TP8_3, 1 TC_TP8_3
14= TC_TP8_4, 1, TC_TP8_4
15= TC_TP8_5, 1, TC_TP8_5
16= TC_TP8_6, 1, TC_TP8_6
17= TC_TP8_7, 1, TC_TP8_7
18= TC_TP8_8, 1, TC_TP8_8
19= TC_TP7, 4, TC_TP7
20= NoData, 1, <automatic extend last column>
21= <EndOfList>

[2_ToolList]

MultiLine=MULTI
NrOfFixedColumns=1
1= NO, 4, NO
2= MagNo, 4, MagNo
3= ToolInPlace, 3, ToolInPlace
4= TC_TP2, 11, TC_TP2
5= TC_TP1, 5, TC_TP1
6= TC_MPP6, 5, TC_MPP6
7= CuttEdgeNo, 1, CuttEdgeNo
8= TC_DP1, 4, TC_DP1
9= TC_DP3, 11, TC_DP3
10= TC_DP4, 11, TC_DP4
11= TC_DP5, 11, TC_DP5
12= TC_DP6, 11, TC_DP6
13= TC_MPP2, 4, TC_MPP2
14= <EndOfList>

[3_ToolList]

NrOfFixedColumns=3
1= NO, 4, NO
2= MagNo, 4, MagNo
3= ToolInPlace, 3, ToolInPlace
4= TC_TP2, 11, TC_TP2

5= TC_TP1, 5, TC_TP1
6= TC_MPP6, 5, TC_MPP6
7= TC_TP3, 1, TC_TP3
8= TC_TP4, 1, TC_TP4
9= TC_TP5, 1, TC_TP5
10= TC_TP6, 1, TC_TP6
11= TC_DP3@E1, 11, e1TC_DP3
12= TC_DP3@E2, 11, e2TC_DP3
13= TC_DP3@E3, 11, e3TC_DP3
14= TC_TP9, 1, TC_TP9
15= TC_MOP1, 7, TC_MOP1
16= TC_MOP2, 7, TC_MOP2
17= TC_MOP3, 7, TC_MOP3
18= TC_MOP4, 7, TC_MOP4
19= <EndOfList>

[1_ActList]

MultiLine=SINGLE
NrOfFixedColumns=1
NumLinesPerReq = 11
1= NO, 4, NO
2= TC_TP2, 11, TC_TP2
3= TC_TP1, 5, TC_TP1
4= TC_MPP6, 5, TC_MPP6
5= CuttEdgeNo, 1, CuttEdgeNo
6= TC_DPCE, 6, TC_DPCE
7= MagNo, 4, MagNo
8= ToolInPlace, 3, ToolInPlace
9= TC_MPP2, 3, TC_MPP2
10= TC_MPP5, 4, TC_MPP5
11= TC_DP1, 11, TC_DP3
13= TC_DP4, 11, TC_DP4
14= TC_SCP13, 9, TC_SCP13
15= TC_SCP14, 9, TC_SCP14
16= TC_SCP23, 9, TC_SCP23
17= TC_SCP24, 9, TC_SCP24

18= TC_ADPT1, 11, TC_ADAPT1
19= TC_ADPT2, 11, TC_ADAPT2
20= TC_ADPT3, 11, TC_ADAPT3
21= TC_ADPT4, 4, TC_ADAPT4
22= TC_TP8_1, 1, TC_TP8_1
23= TC_TP8_2, 1, TC_TP8_2
24= TC_TP8_3, 1, TC_TP8_3
25= TC_TP8_4, 1, TC_TP8_4
26= TC_TP8_5, 1, TC_TP8_5
27= TC_TP8_8, 1, TC_TP8_8
28= <EndOfList>

[2_ActList]

MultiLine=MULTI
NrOfFixedColumns=0
NumLinesPerReq = 6
1= NO, 4, NO
2= TC_TP2, 11, TC_TP2
3= TC_TP1, 5, TC_TP1
4= TC_MPP6, 5, TC_MPP6
5= CuttEdgeNo, 1, CuttEdgeNo
6= TC_DPCE, 6, TC_DPCE
7= MagNo, 4, MagNo
8= ToolInPlace, 3, ToolInPlace
9= DLNO, 3, DLNO
10= TC_SCP3, 9, TC_SCP3
11= TC_SCP4, 9, TC_SCP4
12= <Empty>
13= <Empty>
14= TC_DP1, 4, TC_DP1
15= TC_DP2, 11, TC_DP2
16= TC_DP3, 11, TC_DP3
17= TC_DP4, 11, TC_DP4
18= <Empty>
19= <Empty>
20= TC_ADPT1, 11, TC_ADAPT1

21= TC_ADPT2, 11, TC_ADAPT2
22= TC_ADPT3, 11, TC_ADAPT3
23= TC_ADPT4, 4, TC_ADAPT4
24= <EndOfList>

[3_ActList]

NrOfFixedColumns=3
NumLinesPerReq = 11
1= NO, 4, NO
2= MagNo, 4, MagNo
3= ToolInPlace, 3, ToolInPlace
4= TC_TP2, 11, TC_TP2
5= TC_TP1, 5, TC_TP1
6= TC_DPCE, 6, TC_DPCE
7= TC_TP3, 1, TC_TP3
8= TC_TP4, 1, TC_TP4
9= TC_TP5, 1, TC_TP5
10= TC_TP6, 1, TC_TP6
11= TC_TP9, 1, TC_TP9
12= TC_MOP1, 7, TC_MOP1
13= TC_MOP2, 7, TC_MOP2
14= TC_MOP3, 7, TC_MOP3
15= TC_MOP4, 7, TC_MOP4
16= TC_MOP5, 7, TC_MOP5
17= TC_MOP6, 7, TC_MOP6
18= TC_MOP11, 7, TC_MOP11
19= TC_MOP13, 7, TC_MOP13
20= TC_MOP15, 7, TC_MOP15
21= <EndOfList>

[ListColumnHeaderTrafoTextReplace]

; 举例:

; TC_DP3 = TTC_DP3

; 确定替换文本, 对于列表的转换显示
, 用于刀沿参数和总补偿参数

; 在栏标题中。

; 在密码“TC_DP3”下查找用于转换描述的
 ; 替换密码。
 ; 如果在该段中缺失条目，则在
 ; 相应栏的栏标题中
 ; 显示“missed trafo text” (“不存在转换文本”)
 ; 。

TC_DP3 = TTC_DP3
 e1TC_DP3 = Te1TC_DP3
 e2TC_DP3 = Te2TC_DP3
 e3TC_DP3 = Te3TC_DP3
 TC_DP4 = TTC_DP4
 TC_DP5 = TTC_DP5
 TC_DP6 = TTC_DP6
 TC_DP7 = TTC_DP7
 TC_DP8 = TTC_DP8
 TC_DP9 = TTC_DP9
 TC_DP10 = TTC_DP10
 TC_DP11 = TTC_DP11
 TC_DP12 = TTC_DP12
 TC_DP13 = TTC_DP13
 TC_DP14 = TTC_DP14
 TC_DP15 = TTC_DP15
 TC_DP16 = TTC_DP16
 TC_DP17 = TTC_DP17
 TC_DP18 = TTC_DP18
 TC_DP19 = TTC_DP19
 TC_DP20 = TTC_DP20
 TC_DP21 = TTC_DP21
 TC_DP22 = TTC_DP22
 TC_DP23 = TTC_DP23
 TC_DP24 = TTC_DP24
 TC_DP25 = TTC_DP25
 TC_DPH = TTC_DPH
 TC_DPV = TTC_DPV
 TC_DPV3 = TTC_DPV3
 TC_DPV4 = TTC_DPV4

TC_DPV5 = TTC_DPV5
TC_SCP3 = TTC_SCP3
TC_SCP4 = TTC_SCP4
TC_SCP5 = TTC_SCP5
TC_SCP6 = TTC_SCP6
TC_SCP7 = TTC_SCP7
TC_SCP8 = TTC_SCP8
TC_SCP9 = TTC_SCP9
TC_SCP10 = TTC_SCP10
TC_SCP11 = TTC_SCP11
TC_SCP13 = TTC_SCP13
TC_SCP14 = TTC_SCP14
TC_SCP15 = TTC_SCP15
TC_SCP23 = TTC_SCP23
TC_SCP24 = TTC_SCP24
TC_SCP25 = TTC_SCP25
TC_SCP33 = TTC_SCP33
TC_SCP34 = TTC_SCP34
TC_SCP35 = TTC_SCP35
e1TC_SCP13 = Te1TC_SCP13
e1TC_SCP14 = Te1TC_SCP14
e1TC_SCP15 = Te1TC_SCP15
e1TC_SCP23 = Te1TC_SCP23
e1TC_SCP24 = Te1TC_SCP24
e1TC_SCP25 = Te1TC_SCP25
e1TC_SCP33 = Te1TC_SCP33
e1TC_SCP34 = Te1TC_SCP34
e1TC_SCP35 = Te1TC_SCP35
e2TC_SCP13 = Te2TC_SCP13
e2TC_SCP14 = Te2TC_SCP14
e2TC_SCP15 = Te2TC_SCP15
e2TC_SCP23 = Te2TC_SCP23
e2TC_SCP24 = Te2TC_SCP24
e2TC_SCP25 = Te2TC_SCP25
e2TC_SCP33 = Te2TC_SCP33
e2TC_SCP34 = Te2TC_SCP34

e2TC_SCP35 = Te2TC_SCP35
e3TC_SCP13 = Te3TC_SCP13
e3TC_SCP14 = Te3TC_SCP14
e3TC_SCP15 = Te3TC_SCP15
e3TC_SCP23 = Te3TC_SCP23
e3TC_SCP24 = Te3TC_SCP24
e3TC_SCP25 = Te3TC_SCP25
e3TC_SCP33 = Te3TC_SCP33
e3TC_SCP34 = Te3TC_SCP34
e3TC_SCP35 = Te3TC_SCP35
TC_ECP3 = TTC_ECP3
TC_ECP4 = TTC_ECP4
TC_ECP5 = TTC_ECP5
TC_ECP6 = TTC_ECP6
TC_ECP7 = TTC_ECP7
TC_ECP8 = TTC_ECP8
TC_ECP9 = TTC_ECP9
TC_ECP10 = TTC_ECP10
TC_ECP11 = TTC_ECP11
TC_ECP13 = TTC_ECP13
TC_ECP14 = TTC_ECP14
TC_ECP15 = TTC_ECP15
TC_ECP23 = TTC_ECP23
TC_ECP24 = TTC_ECP24
TC_ECP25 = TTC_ECP25
TC_ECP33 = TTC_ECP33
TC_ECP34 = TTC_ECP34
TC_ECP35 = TTC_ECP35
e1TC_ECP13 = Te1TC_ECP13
e1TC_ECP14 = Te1TC_ECP14
e1TC_ECP15 = Te1TC_ECP15
e1TC_ECP23 = Te1TC_ECP23
e1TC_ECP24 = Te1TC_ECP24
e1TC_ECP25 = Te1TC_ECP25
e1TC_ECP33 = Te1TC_ECP33
e1TC_ECP34 = Te1TC_ECP34

e1TC_ECP35 = Te1TC_ECP35
e2TC_ECP13 = Te2TC_ECP13
e2TC_ECP14 = Te2TC_ECP14
e2TC_ECP15 = Te2TC_ECP15
e2TC_ECP23 = Te2TC_ECP23
e2TC_ECP24 = Te2TC_ECP24
e2TC_ECP25 = Te2TC_ECP25
e2TC_ECP33 = Te2TC_ECP33
e2TC_ECP34 = Te2TC_ECP34
e2TC_ECP35 = Te2TC_ECP35
e3TC_ECP13 = Te3TC_ECP13
e3TC_ECP14 = Te3TC_ECP14
e3TC_ECP15 = Te3TC_ECP15
e3TC_ECP23 = Te3TC_ECP23
e3TC_ECP24 = Te3TC_ECP24
e3TC_ECP25 = Te3TC_ECP25
e3TC_ECP33 = Te3TC_ECP33
e3TC_ECP34 = Te3TC_ECP34
e3TC_ECP35 = Te3TC_ECP35

[BatchTools]

; 刀具的任务功能控制:
; 装载, 卸载或再次激活大量刀具
; 提示: 仅刀具过滤器有效, 当
; ToolManagementMask 位 4 (从 0 至 ...) 设置情况下。
; 可以确定最多 6 个过滤器。
; 对于每个过滤器可以如下规定:
; 软键文本, 列标题, 查找标准, 选择
; 结果列表类型和附加数据。
文件 ...user\paramini.out 包含错误信息, 针对
; 读取参数时找到的错误。
; 查找标准:
; 允许的值在“FindCondition”中:
; 允许最多 8 个条目, 用 "," 隔开。
; 用逻辑和链接。在零件条件中
; 不允许多次出现数据。每个零件条件

; 由 3 段组成:

; 1. 数据, 适用条件的数据

; 2. 条件

; 3. 比较值

; 下列数据可以是过滤器标准:

; 刀具数据:

TC-TP1 D 号码

TC-TP2 刀具名称

TC-TP3 刀具尺寸, 在左半位

TC-TP4 刀具尺寸, 在右半位

TC-TP5 刀具尺寸, 在上半位

TC-TP6 刀具尺寸, 在下半位

TC-TP7 刀具刀位类型

TC-TP8 刀具状态

TC-TP9 监控类型

TC-TP10 替换刀具查找

TC-TP11 刀具信息/姊妹刀具顺序

A_TOOLMN 刀库号码

A_TOOLMLN 刀库位置号码

P_TOOLND 刀沿数

; 刀具-OEM-数据: "TC_TPC1"至"TC_TPC10"

; 在 NC 上刀具-OEM-数据必须激活

; 并允许是编号。

; 刀具刀沿参数:

"TC_DP1" 至"TC_DP25", "TC_DPH", "TC_DPV", "TC_DPV3", "TC_DPV4",
"TC_DPV5"(NCK 设置应用替代 "25")

; 刀具刀沿-OEM-数据: "TC_DPC1" 至 "TC_DP10"

; 刀具刀沿-OEM-数据必须在 NC 上激活

; 且编号必须在 NC 允许。

; 刀具刀沿监控参数:

TC_MOP1 预警使用寿命

TC_MOP2 实际值使用寿命

TC_MOP3 预警极限件数

TC_MOP4 实际值件数

TC_MOP5 预警极限 磨损

TC_MOP6 剩余磨损

TC_MOP11 使用寿命 标准值
TC_MOP13 件数标准值
TC_MOP15 磨损标准值
; 刀具刀沿监控-OEM-数据 "TC_MOPC1" 至
; "TC_MOPC10" 刀具刀沿监控-OEM-数据必须
; 在 NC 上激活且编号必须在 NCK 上
; 允许。 如果 NCK-版本 \geq 430000: 用户数据
; 刀具用户数据:
; " TC_TPCS1" 至 "TC_TPCS10"
; 刀具用户数据必须在 NC 上激活且
; 编号必须在 NC 上允许。
; 刀具刀沿用户数据:
; "TC_DPCS1" 至 "TC_DPCS10"
; 刀具刀沿用户数据必须在 NC 上激活
; 且编号必须在 NC 上允许。
; 刀沿监控用户数据:
; "TC_MOPCS1" 至 "TC_MOPCS10"
; 刀沿监控用户数据必须在 NC 上激活
; 且编号必须在 NC 上允许。
; 条件:
"==" 等于
"<" 小于
">" 大于
"<=" 小于等于
">=" 大于等于
; && 位方式“和”，仅允许用于运算域，类型
; WORD 和 DOUBLEWORD
; "==" 是一个允许的比较运算符，用于
; 字符串运算域
; 比较值： 字符串对于 TC_TP2 (刀具数据，
; 刀具名称)，
; 最多 32 个字符，前后没有空格
; 0 ... 65535 对于所有其他的 TC_TP 数据
; 双倍对于所有其他的数据
; 每个过滤器可以最多显示一个带有附加数据的栏
; 。

```

;_FindResultAddColumnBtss:
;附加数据, BTSS-项根据 Btss-文件 (mmc2\btss_gr.hlp)。
;示例 1: "/Tool/User/data[u#TOA#,c2,#TNO#](!"!d%.#RES#lf)"
;刀具-OEM-参数 2, 浮点描述,
;标准小数点后的位数
;示例 2: "/Tool/User/data[u#TOA#,c3,#TNO#](!"!%ld)"
;刀具-OEM-参数 3, 整数描述,
;示例 3: "/Tool/MagazineDescription/userData[u#TOA#,c#MAG#,1]()"
;刀库 OEM 参数 1
;允许下列通配符: #TOA#, #TNO#, #MAG#, #RES#.
;#TOA# TOA-号 (当前通道的)
;#TNO# interne T-号 (找到的刀具)
;#MAG# 刀库号 (找到的刀具)
;#RES# 小数点后的位数标准值
;通配符由当前刀具数据或者
;由一般设置替代。
;最多允许一个 1 BTSS-项。
;结果数据前的数据分隔符 "|"必须由
;数据 "()" 建立。
;由 BTSS-项建立内部 BTSS 多变量存取
;。 BTSS-项必须写在 "" 中,
;尤其如果包含一个格式数据,
;这写在 "" 内。
;设置参数的用户完全负责
;对于正确的句法。句法不由 BTSS 检查。
;所有过滤器的一般设置:
;这些条目适用于 HMI_ADV, SW 6.3 以前版本。
;从版本 6.3 起忽略。
;结果列表颜色:
;每个颜色数据分配一个十六进制值, 由 8 个符号组成,
;十六进制值有下列结构:
;SSBBGGRR 带 SS=系统, BB=蓝色, GG=绿色, RR=红色
;对于下列列表元素必须规定颜色:
;未选择的文本
;未选择的背景
;光标选择的文本

```

```

; 光标选择的背景
; 任务-选择-选择的文本
; 任务-选择-选择的背景
; 任务-选择-和光标选择的文本
; 任务-选择-和光标选择的背景
; WinTxt, WinBa, HighLTxt, HighLiBa, TiBaTxt green,
; TiBaTxt bluegreen
; ResultColors=80000008,80000005,8000000E,8000000D,80000009,
; 000FF00,80000009,00FF8000
; experimental, LiteBlue for batch selected
; WinTxt, WinBa, HighLTxt, HighLiBa, WinTxt ,LiteBlue,
; HighLTxt, HighLiBa
; ResultColors=80000008,80000005,8000000E,8000000D,80000008,
; 00FFFF00,8000000E,8000000D
;experimental, LiteGreen for batch selected
;WinTxt, WinBa, HighLTxt,HighLiBa,WinTxt ,Litegree,HighLTxt,HighLiBa
ResultColors=80000008,80000005,8000000E,8000000D,80000008,
; 00FF000,8000000E,8000000D
; WinTxt, WinBa, HighLTxt,HighLiBa,HighLTxt,HighLiBa,HighLTxt,HighLiBa
ResultColors=80000008,80000005,8000000E,8000000D,8000000E,
; 8000000D,8000000E,8000000D
; 位图名称或位图本身可以由
; 用户自身的位图替代。对于存放
; 自身的位图，给出了目录“user”。
BatchFilterEIBUnTUnBitmap = pfbbutu.bmp
BatchFilterEIBUnTSeBitmap = pfbbutu.bmp
BatchFilterEIBSeTUnBitmap = pfbbstu.bmp
BatchFilterEIBSeTSeBitmap = pfbbstu.bmp
BatchRunEIWaitingBitmap = pbbwait.bmp
BatchRunEIInWorkBitmap = pbbwork.bmp
BatchRunEIOKBitmap = pbbok.bmp
BatchRunEIErrorBitmap = pbberr.bmp
; 典型字符的宽度
CharToGetColWidthPerCharacter= CharToGetColWidth
语言相关，参见 ...\\language\\patm_xx.ini
; 选择刀具状态位，它在结果列表中

```

```

;显示。
; ResultToolStatusColumnsEnable= 1111100100110000
; 位 1 至 16，位 1 在刀具状态中是最低级的，
; 位于标题字符序列文本的左侧
; 刀具状态栏：
; 如果 ResultToolStatusColumnsHeaderText 和
; ResultToolStatusColumnsListText 为 "<Empty>"，
; 国家语言相关的值位于批处理列表
; pa_xx.dll 中。
ResultToolStatusColumnsHeaderText=<Empty> ;国家语言相关
; ResultToolStatusColumnsHeaderText=ToolStatusColHeaderText
语言相关
; 刀具状态栏中的数据：
ResultToolStatusColumnsListText= <Empty> ;国家语言相关
; ResultToolStatusColumnsListText= ToolStatusColListText
语言相关
; 刀具名称栏宽度
ResultDisplayedNumberOfToolnameCharacters=18
TimeMSecBetweenBatchOrders=1000
; 定义单个的过滤器：
1_FindSoftkeyText = F1SK
; 国家语言相关；预警或禁用
1_FindResultHeadlineText = R1HL
; 国家语言相关；达到预警极限或禁用
1_FindCondition = TC_TP8 && 20
; 设置预警位（位 5 从位 1 至 16 (2hoch(5-1)=16) +
; 设置禁用位 (位 3 (2hoch(3-1)=4)
1_FindResultAddColumnBtss = <empty>
1_FindResultAddColumnText = <empty> ; 或 R1AddCol ; 语言相关
1_FindResultAddColumnDisplayedNumberOfCharacters=0
1_FindLimitedToCurMagazine=true
; "True", "False"(预设)；限制当前刀库，
; 如果通过刀库列表调用。
1_ResultListType =0
0 = 标准列表 (预设), 1 = 装载列表
1_ReactivatePositioningMode = 2
; 在再次激活时定位

```

; 0: 未定位,
; 1: 询问操作人员是否要定位,
; 2: 总是定位 (预设)
2_FindSoftkeyText = F2SK ; "已禁用"
2_FindResultHeadlineText = R2HL ; "刀具已禁用"
2_FindCondition = TC_TP8 && 4
; 已禁用=位 3 (2hoch(3-1)=4)
2_FindResultAddColumnBtss = <empty>
2_FindResultAddColumnText = <empty> ; 或 R2AddCol
2_FindResultAddColumnDisplayedNumberOfCharacters=0
2_FindLimitedToCurMagazine=False
; "True", "False"(预设) ; 限制当前刀库,
; 如果通过刀库列表调用。
2_ResultListType =0
; 0 = 标准列表 (预设), 1 = 装载列表
2_ReactivatePositioningMode = 0
; 0: 未定位,
; 1: 询问操作人员是否要定位,
; 2: 总是定位 (预设)
3_FindSoftkeyText = F3SK ; "全部装载"
3_FindResultHeadlineText = R3HL
; 所有装载刀具的卸载列表
3_FindCondition = A_TOOLMN > 0
; 刀具的刀库编号大于 0
3_FindResultAddColumnBtss = <empty>
3_FindResultAddColumnText = <empty> ; 或 R3AddCol
3_FindResultAddColumnDisplayedNumberOfCharacters=0
3_FindLimitedToCurMagazine=False
; "True", "False"(预设) ; 限制当前刀库,
; 如果通过刀库列表调用。
3_ResultListType =0
; 0 = 标准列表 (预设), 1 = 装载列表
3_ReactivatePositioningMode = 1
; 在再次激活时定位
; 0: 未定位, 1: 询问操作人员是否要定位
; , 2: 总是定位 (预设)

```

4_FindSoftkeyText = F4SK ; "全部卸载"
4_FindResultHeadlineText = R4HL
; 所有卸载刀具的装载列表
4_FindCondition = A_TOOLMN == 0
; 刀具的刀库编号等于 0
4_FindResultAddColumnBtss = <empty>
4_FindResultAddColumnText = <empty> ; 或 R4AddCol
4_FindResultAddColumnDisplayedNumberOfCharacters=0
4_FindLimitedToCurMagazine=False
; "True", "False"(预设); 限制当前刀库,
; 如果通过刀库列表调用。
4_ResultListType = 0
; 0 = 标准列表 (预设), 1 = 装载列表
4_ReactivatePositioningMode = 1
; 在再次激活时定位
; 0: 未定位,
; 1: 询问操作人员是否要定位,
; 2: 总是定位 (预设)
5_FindSoftkeyText = F5SK ; "装载识别符"
5_FindResultHeadlineText = R5HL
; "装载列表用于所有带有装载识别符的刀具"
5_FindCondition = TC_TP8 && 2048
; (LoadIdentifier=Bit12 (2hoch(12-1)=2048)
5_FindResultAddColumnBtss = <empty>
5_FindResultAddColumnText = <empty> ; 或 R5AddCol
5_FindResultAddColumnDisplayedNumberOfCharacters=0
5_FindLimitedToCurMagazine=False
; "True", "False"(预设) 限制当前刀库,
; 如果通过刀库列表调用。
5_ResultListType = 1
; 0 = 标准列表 (预设), 1 = 装载列表
6_FindSoftkeyText = F6SK ; "卸载识别符"
6_FindResultHeadlineText = R6HL
; "卸载列表用于所有带有卸载识别符的刀具"
6_FindCondition = TC_TP8 && 1024
; (卸载识别符=位 11 (2 高阶(11-1)=1024)

```

```

6_FindResultAddColumnBtss = <empty>
6_FindResultAddColumnText = <empty> ; 或 R6AddCol
6_FindResultAddColumnDisplayedNumberOfCharacters=0
6_FindLimitedToCurMagazine=False
; "True", "False"(预设); 限制当前刀库,
; 如果通过刀库列表调用。
6_ResultListType =0
; 0 = 标准列表 (预设), 1 = 装载列表

```

[ShortcutSoftkeysForMagSelect]

```

; 定义快捷图标软键最多至 5 个自选刀库
; 通过 TOA, 评估是否段 "[GeneralSettingsForMagAndToolList]"
, 设置条目 "MagListMagSelectSoftkey=SelectMag" 。
; 这有助于防止用户过度使用 "刀库 +"
, 和 "刀库 -" 软键, 如果大量刀库可用。
; 可以定义最多 5 快捷图标软键用于自选刀库。
, 不同刀库可以使用一个快捷图标软键,
; 如果该刀库在不同的 TOA 中。
; 句法: "magIdent = ShortKeyNummer, AutoReturn"
; 举例: revolver15=3,NoAuto<< chain50 =1,Auto<<
; 含义:
; "magIdent": 刀库识别类似 $TC_MAP2 中或
; 刀库配置在应用维修刀具
; 管理中。
; "ShortKeyNummer": 快捷图标编号值 1 至 5
; "AutoReturn": 停留在刀库段状态或自动返回
; 到刀库列表状态值 "NoAuto<<"
; 和 "Auto<<".
; 刀库 "revolver15" ($TC_MAP2) 可以通过快捷图标 3 显示,
; 必须使用 "<<" 软键明确离开刀库选择
; 状态。
; 刀库 "chain50" 可以通过快捷图标 1 显示,
; 自动返回到刀具列表状态, 在按下
; 快捷图标软键后。 指定软键文本, 使用段
; "[ShortcutSoftKeysForMagSelect]" 语言相关
; ini files patm_?.ini.

```

```

; kette10 = 1, Auto<<
; revolver20 = 2, Auto<<
; revolver10 = 3, Auto<<
; kette20 = 5, NoAuto<<
; revolver15 = 4, NoAuto<<
    
```

6.2.3 设计软键显示

用于列表的软键文本

位于刀库和刀具列表中的软键 1-3 后的画面在文件 paramtm.ini 中定义。
 在选择 WZV 时作为预设出现，这已经在 [1_MagList] 和 [1_ToolList] 时设计。

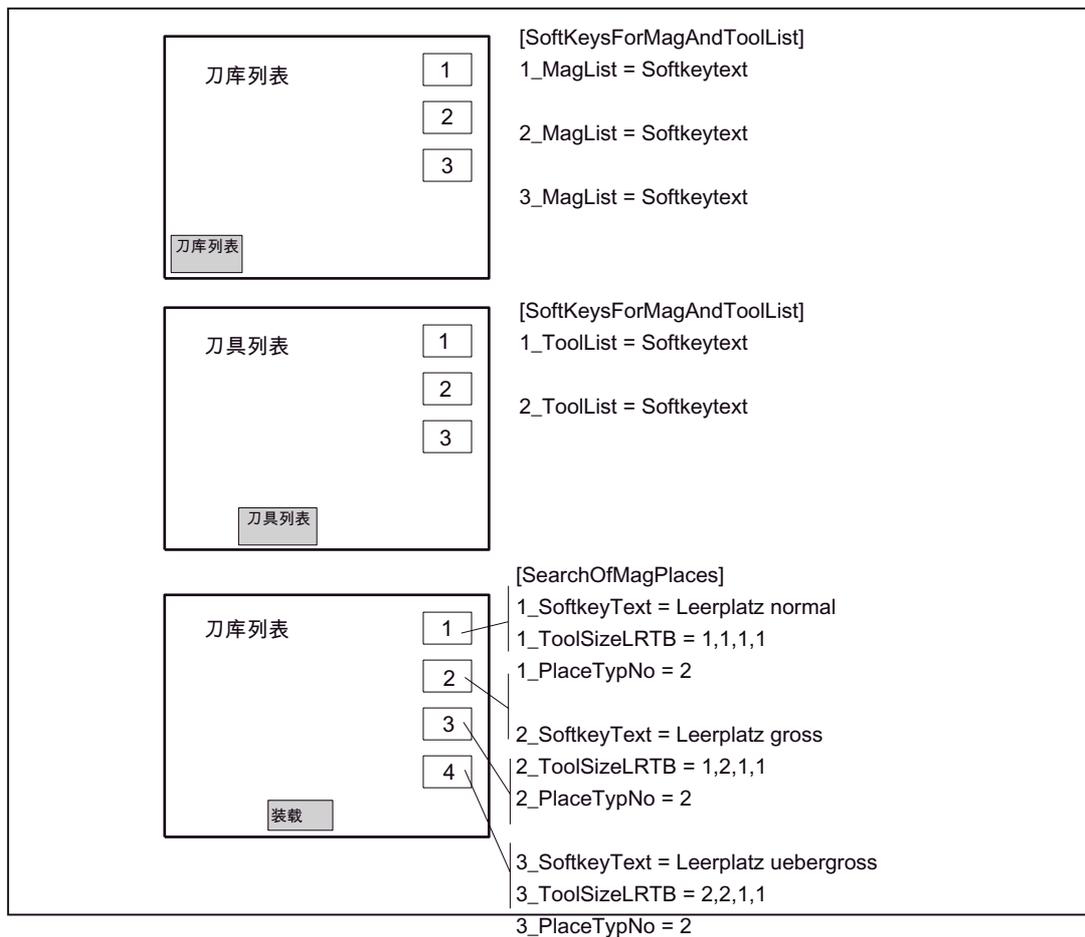


图 6-14 软键文本

画面描述

不可见的栏可通过相应一段光标键（滚动）可见。

对于刀位类型，不显示刀位名称，而现实运行编号，这些运行编号在调试刀位类型时通过输入顺序确定。在刀库列表第一垂直软键下显示的画面在文件 paramtm.ini 中根据关键字 [1_MagList] 确定。

用户数据

对于刀具和刀沿用户数据画面，可以确定参数名称和单位。如许多参数显示那样，取决于机床数据和定义的参数数目。

[ToolParams]: 刀具用户数据

[ToolEdgeParams]: 刀沿用户数据

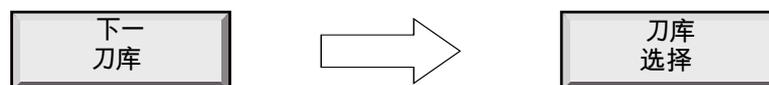
特殊字符

特殊字符如 ü, ä, ö, ß 以 ANSI 代码输入以便显示。

有目的的刀库选择

迄今为止可以在刀库列表中通过软键“下一个刀库”控制单个刀库的显示。如果存在大量的刀库，并且在带有较高的刀库编号的刀库中进行过程，则对于操作人员来说是非常不便的。

通过在 ini 文件中的条目，软键“下一个刀库”由软键“刀库选择”替代：



显示 8 个垂直软键用于较快的刀库导航：



在这种情况下显示刀库列表。

(仅使用软键 3 至 7，如果已在 ini 文件中参数化。)

通过垂直软键“刀库 +”和“刀库 -”可以在操作面板的当前通道的 TOA 中切换到带有较高刀库编号的刀库或下一个较低的刀库编号的刀库。(从最后的刀库通过“+”切换到第一个，从第一个通过“-”切换到最后一个。)

用 5 个垂直软键 3 至 7 可以在操作面板的当前通道的 TOA 中快速切换到某个刀库。针对“刀库识别”和软键文本的分配必须在 INI 文件中参数化。用垂直软键选择一个刀库，刀库列表立即切换到新的刀库。通过垂直软键 8 “<<”实现返回到带有相应软件占用的一般刀库列表状态。

对于 5 个快速选择刀库的软键，可以在 INI 文件中设置，使其自动实现跳回到一般刀库列表状态。在这种情况下建议在软键文本中刀库名称上添加符号序列“<<”。

快速选择按键可以多次占用，用在不同的 TOA 中和带有 N:M 分配的设备中（HMI 高级和 NCU 之间）。为此编写参数人员自己负责仅这些刀库占用相同的软键，这些软键位于不同的 TOA 或不同的 NCU 中。

文件 paramtm.ini 中的条目

[GeneralSettingsForMagAndToolList]

- ;在刀库列表形式中改变软键“下一个刀库”
- ;对于软键“刀库选择”，激活垂直软键
- ;在刀具选择附加状态中。
- ;这有助于防止用户过度使用“刀库 +”
- ;“下一个刀库”软键，如果多个刀库可用。
- ;可以定义快捷图标软键最多至 5 个自选
- ;刀库，通过 TOA 使用段 [ShortcutSoftKeysForMagSelect]
- ;在 paramtm.ini and patm_?.ini 中。

MagListMagSelectSoftkey=NextMag ; 默认

```

; MagListMagSelectSoftkey=SelectMag
[ShortcutSoftKeysForMagSelect]
; 定义快捷图标软键最多至 5 个自选刀库
; 通过 TOA, 评估是否段 "[GeneralSettingsForMagAndToolList]"
; 设置条目 "MagListMagSelectSoftkey=SelectMag" 。
; 这有助于防止用户过度使用 "刀库 +" 和
; 和 "刀库 -" 软键, 如果大量刀库可用。
; 可以定义最多 5 快捷图标软键用于自选刀库。
; 不同刀库可以使用一个快捷图标软键,
; 如果该刀库在不同的 TOA 中。
; 句法: "magIdent = ShortKeyNummer, AutoReturn"
; 示例: revolver15=3,NoAuto<<
; chain50 =1,Auto<<
; 解释: "magIdent": 刀库识别类似 $TC_MAP2 中或
; 刀库配置在应用维修
; 刀具管理中。 "ShortKeyNummer": 快捷图标编号,
; 值 1 至 5 "AutoReturn":
; 自动停留在刀库段状态或自动返回
; 到刀库列表状态值 "NoAuto<<" 和 "Auto<<"。
; 刀库 "revolver15" ($TC_MAP2) 可以通过快捷图标 3 显示,
; 必须使用 "<<" 软键明确离开
; 刀库选择状态。
; 刀库 "chain50" 可以通过快捷图标 1 显示,
; 有一个自动返回到刀库列表状态
; 在按下快捷图标软键后。
; 指定软键文本, 使用段
"[ShortcutSoftKeysForMagSelect]" 在语言相关 ini 文件
; patm_??.ini 。
kette10 = 1, Auto<<
; revolver20 = 2, Auto<<
> ;revolver10 = 3, Auto<<
> ;kette20 = 5, NoAuto<<
> ;revolver15 = 4, NoAuto<<
>
language\patm_*.ini:
[ShortcutSoftKeysForMagSelect]
; 刀库选择快捷图标软键的软键文本。

```

```

; 句法: magIdent=ShortcutSoftkeyText
; 解释: "magIdent": 刀库识别类似 $TC_MAP2 中或
; 刀库配置在应用
; 维修刀具管理。
; "ShortcutSoftkeyText": 软键文本, 使用双空格至
; 指示的 wordwrap。
; revolver10 = "1-Rev10 <<" // Soft key text
; revolver20 = "2-Rev20 <<" // Soft key text
; kette10 = "3-Kette10 <<" // Soft key text
; revolver15 = "4-Rev15" // Soft key text
; kette20 = "5-Kette20" // Soft key text
    
```

显示临近刀位状态

在 HMI 高级刀具管理列表画面中（刀库列表，刀具列表，工作补偿列表），现在也可以显示刀库位置状态的位 8 至 16，其下面显示 4 位“左，右，上，下，在半位下保留”。

NC 在接通“临近刀位管理”时使用这些数据。

这些数据符合 NC 变量 \$TC_MPP4。

迄今显示列表中刀库位置状态的位 1 至 8；扩展后可以类似方式显示位 8 至 16。

刀库位置状态位 8 至 16 可以借助于 HMI 高级软件显示，但是不能改变。

在用 HMI 高级软件提供的示例参数 (paramtm.ini, paramtm.txt) 中准备显示附加的刀库位置状态位，但不激活。

参数化

参数化状态位显示作为列表画面 1 至 3 的各个列表画面中的栏（刀库列表 1 至 3，刀具列表 1 至 3，工作补偿列表 1 至 3）按意义补充新的状态位。

如同以前，国家语言相关的单个字母也可以按选择方式在 INI 文件中参数化，以用于 HMI 高级 WZV 列表画面中的显示，例如当机床操作人员要使用另一种单个字母时或者还没有用于某个国家语言的改变的语言 DLL 文件。

限制

NC 使用数据“左，右，上，下半位已占用/已保留”仅针对接通的“临近刀位管理”。如果未接通 NC 中的“临近刀位管理”，则附属的刀库位置状态位显示甚至可以在 HMI 高级 WZV 列表画面中接通，但显示的值总是为“位未设置”。

在 INI 文件中设置

paramtm.ini / paramtm.txt:

```
[GeneralSettingsForMagAndToolList]
```

```
MagPlaceState_Lang_12345678=<Empty> ; 使用语言-DLL
```

MagPlaceState_Lang_12345678=12345678_MagPlaceState_Lang

; use patm_*.ini

; 在每个列表画面中可以显示新的刀库位置状态位

;。 参数化在下列段中实现:

"[1_MagList]", "[2_MagList]", "[3_MagList]"

"[1_ToolList]", "[2_ToolList]", "[3_ToolList]"

"[1_ActList]", "[2_ActList]", "[3_ActList]"

; 条目根据例如:

12=TC_MPP4_9, 1, TC_MPP4_9 ;PlaceStatus Left,

; 在左半位保留

13=TC_MPP4_10, 1, TC_MPP4_10 ;PlaceStatus Right,

; 在右半位保留

14=TC_MPP4_11, 1, TC_MPP4_11 ;PlaceStatus Top,

; 在上半位保留

15=TC_MPP4_12, 1, TC_MPP4_12 ;PlaceStatus Bottom,

; 在下半位保留

16=TC_MPP4_13, 1, TC_MPP4_13 ;PlaceStatus Bit 13 (1 至 16)

17=TC_MPP4_14, 1, TC_MPP4_14 ;PlaceStatus Bit 14 of 1 至 16

18=TC_MPP4_15, 1, TC_MPP4_15 ;PlaceStatus Bit 15 (1 至 16)

19=TC_MPP4_16, 1, TC_MPP4_16 ;PlaceStatus Bit 16 (1 至 16)

patm_gr.ini:

[GeneralSettingsForMagAndToolList]

12345678_MagPlaceState_Lang="123456789ABCDEFGH" ; // 16 exact

[ListColumnHeaderText]

TC_MPP4_9 = "P" ;刀位状态 左,

;在左半位保留 // 1

TC_MPP4_10 = "P" ;刀位状态 右,

;在右半位保留 // 1

TC_MPP4_11 = "P" ;刀位状态 上,

;在上半位保留 // 1

TC_MPP4_12 = "P" ;刀位状态 下,

;在下半位保留 // 1

TC_MPP4_13 = "P";刀位状态未定义,

;(磨损相关已禁用) // 1

TC_MPP4_14 = "P" ;刀位状态位 14 从 1 至 16 // 1

TC_MPP4_15 = "P" ;刀位状态位 15 从 1 至 16 // 1

TC_MPP4_16 = "P" ;刀位状态位 16 从 1 至 16 // 1

刀位和刀具状态编码

刀位状态		刀具状态	
G	已禁用刀位	G	已禁用刀具
F	空闲的刀位	F	释放的刀具
Z	在中间存储器保留刀具	A	激活的刀具
B	保留用于待装载的刀具	M	已测量的刀具
L	左半位已占用	V	已达到预警极限
R	右半位已占用	W	刀具处于更换中
O	上半位已占用	P	固定刀位编码的刀具
U	下半位已占用	E	刀具已使用
l	左半位已预留	R	卸载识别
r	右半位已预留	B	装载识别
o	上半位已预留	S	源刀具
u	下半位已预留		

6.2.4 参数化刀具位图显示

参数化列表中的位图

显示激活，编程的刀具和当前刀库列表中的刀位是可以自由参数化的，即在单个列表中可以出现在可参数化的栏位图中。位图可以用户专门匹配或者重新建立。显示在 paramtm.ini 中激活。

下面描述的位图在目录 "hmi_adv" 中。

预设位图

当前刀具位图总是以红色描述，以绿色描述编程的刀具。

位图		属性
		T 号 <> 0; D 号/刀沿号 <> 0 DL 号 <> 0
		T 号 <> 0; D 号/刀沿号 = 0; DL 号 = 0
		T 号 <> 0; D 号/刀沿号 <> 0; DL 号 = 0
		当前位置

在目录 "user" 中可以占用用户定义的位图。这显示在列表中标准位图位置上。

处理列表

要输入位图的列表栏可以针对每个列表视图设置。对于全部标记，位图宽度以符号设置。位图宽度此时自动提高到设置的值。

如果在相同的栏和行显示，位图相互覆盖。整个上部为当前刀具的标记，其下面为编程的刀具的标记，整个下部为当前刀位的标记。不给出隐藏的位图。

说明

对于多行的刀库和刀具列表，在当前/编程的 D 号/刀沿号 <> 0 时将标记输入到刀沿行中。对于工作补偿列表适用于 DL 号。<> 0 相应于 DL 行。由于视图中工作补偿列表仅描述刀沿，当当前/编程的 D 号/刀沿号 <> 0 时，则仅实现标记。

当前刀库位置仅标记在刀库列表视图中。标记在刀库列表中仅描述在一般刀库显示中，不在中间存储器刀库显示中。

参数化位图

根据标准位图不输入在 paramtm.ini 中，也不描述。如果要在列表中显示位图，必须更改参数文件。对于每个位图实现一个条目。

paramtm.ini 中条目：

[GeneralSettingForMagAndToolList]

; 位图显示宽度

; 单元：符号数

WidthOfActBitmapsInCharacters=5

; 用于当前刀具 / D 号 / DL 的位图名称，

; 带 D<>0 和 DL<>0

ActToolBitmap=paat.bmp

; 用于当前刀具 / D 号的位图名称，

; 当当前刀沿 D=0 情况下。

ActToolZeroDBitmap=paatd0.bmp

; 用于当前刀具 / D 号 / DL 的位图名称，

; 当当前 DL=0 情况下。

ActToolZeroDLBitmap=paatdl0.bmp

; 用于编程的刀具 / D 号 / DL 的位图名称，

; 带 D<>0 和 DL<>0

ProgToolBitmap=papt.bmp

; 用于编程的刀具 / D 号的位图名称，

; 当当前刀沿 D=0 情况下。

ProgToolZeroDBitmap=paptd0.bmp

; 用于编程的刀具 / D 号 / DL 的位图名称，

; 当当前 DL=0 情况下。

ProgToolZeroDLBitmap=paptdl0.bmp

; 位图文件名用于当前刀库位置

ActPlaceBitmap=paap.bmp

[1_MagList]

; 要使栏显示在标记（位图）中

;

ShowActToolCol=1

ShowProgToolCol=1

ShowActPlaceCol=1

6.2.5 示例：用于机床的用户定义的设置

在文件 paramtm.ini 中设计 WZV 画面

通过文件 paramtm.ini 可以匹配刀具管理操作界面：

- 更改列表结构和布置。
- 预占用确定的值。
- 通过访问权限保护或者退出激活功能

在 HMI 高级上的文件 paramtm.txt 中，显示所有功能和刀具管理可能性。

在调试时要决定某个专用机床需要哪些功能。可以预设值和部分功能，以建立一个简单的和舒适的操作。

示例

示例 1:

- 在卸载时要自动删除刀具数据（仅刀库列表）。
- 用该功能刀具列表不工作。
- 用该功能刀具目录和工具箱不工作。

参数化可以如下进行：

...

[TMMODES]

...

DELETE_TOOL_ON_UNLOAD=1 在卸载时刀具数据
自动删除。

...

[ACCESSLEVEL]

...

SKTLLIST=2 刀具列表仅通过制造商识别字
激活，也在普通模式下禁用。

SLTOOLCAB=2 刀具目录和工具箱 SKTOOLCAT=2 仅通过
制造商关键字激活，也在普通模式下禁用。

...

示例 2:

- 在卸载时不删除刀具数据，保留在刀具列表中（在 NCK 中）。在装载时可以重新恢复这些数据。
- 用该功能刀具目录和工具箱不工作

参数化可以如下进行：

...

[TMMODES]

...
DELETE_TOOL_ON_UNLOAD=0 在卸载时刀具数据
不删除。
...
[ACCESSLEVEL]
...
SKTLLIST=7 刀具列表总是可调用。
SLTOOLCAB=2 刀具目录和工具箱 SKTOOLCAT=2 仅通过
制造商关键字激活，对于用户为禁用。

示例 3:

- 在装载到刀库列表中时，自动删除刀具数据。
- 用该功能刀具目录和工具箱工作。

参数化可以如下进行：

...
[TMMODES]
...
DELETE_TOOL_ON_UNLOAD=1 在卸载时刀具数据
删除。
...
[ACCESSLEVEL]
...
SKTLLIST=2 刀具列表仅通过制造商识别字激活。
SLTOOLCAB=7 刀具目录和工具箱
SKTOOLCAT=7 可调用（未锁止）
...

说明

访问权限

对于功能规定了访问权限，当前保护等级“小于”规定的，则软键不出现在操作界面上，无法调用功能。

这适用于全部功能。例如如果功能“工具箱”不可调用，则不显示相应的软键。

6.3 用户定义名称的国家语言相关性

6.3.1 刀库的国家语言相关的名称

功能

刀库名称通过刀具管理 (IW) 调试工具由用户在刀库画面中配置。由此给定的名称位于刀具管理数据库中，在装载一个刀库配置后位于 NCK 中，但不在语言-DLL 中。这导致以前在不同语言中的不可用。

用新的功能可以使用户管理国家语言相关的刀库名称。

对此对于刀具数据库中配置的刀库，将相应的名称文本输入在国家语言相关的刀具管理 INI 文件中。

每个刀库有两个名称：

- 标准名称 (NCK 识别) 通过其功能工作。
- 一个附属的国家语言相关的名称，这显示在操作界面上。

如果用户没有给定国家语言相关的名称，则在刀具管理中标准名称由 NCK 显示，在调试工具中标准名称由刀具数据库显示。

国家语言相关的 INI 文件中的条目

这里描述的条目必须用户自己进行。它不由 HMI 高级写在 INI 文件中。

国家语言相关的 INI 文件位于 .../hmi_adv/language。

```
文件          patm_xx.ini
段落          [Magazine_VISName]
条目          Standardname="语言相关的文本"
举例:         "[Magazine_VISName]"
              kette1="Kettenmagazin1"
              rev1="Revolver1"
              ...
```

为了从 INI 文件显示一个输入的文本，必须进行一次语言切换或者重新启动 HMI 高级。

在 HMI 刀具管理中显示刀具名称

在所有刀具管理和刀具管理调试画面中，显示刀库的国家语言相关的名称。如果在相应的 INI 文件中不存在条目，则标准名称由 NCK (在刀具管理中) 或由刀具数据库 (在调试工具中) 显示。

相关的画面/ 功能:

- 刀具管理:
 - 刀库列表
 - 刀库选择
- 调试刀具管理:
 - 刀库
 - 中间存储器
 - 装载刀位
 - 刀库配置

6.3.2 中间存储器的国家语言相关的名称

功能

中间存储器刀位和其名称由用户通过刀具管理 (IW) 的调试工具在中间存储器画面中给定。由此给定的名称位于刀具管理数据库中，不在语言 DLL 中。这导致在不同语言中的不可用。

以前有两种可以用于显示刀具管理中的中间存储器刀位（不在刀具管理调试中）：

- 由刀具数据库显示名称 (paramtm.ini, [TMMODES] NameOfBufferPlaceFrom=DB)
- 显示中间存储器刀位类型由语言-DLL plus 附属的索引。例如 Spindel1, Spindel2 或 Greifer2 等等。(NameOfBufferPlaceFrom=DLL)

用新的功能可以使用户完全管理国家语言相关的中间存储器刀位名称。

对此必须选择由数据库显示名称 (NameOfBufferPlace From=DB)。

对于数据库中配置的中间存储器刀位，用户必须在语言相关的刀具管理 INI 文件中输入一个相应的名称文本。

每个中间存储器刀位有两个名称：标准名称，通过其内部（刀具数据库）工作，在操作界面上显示一个附属的语言相关的名称。

如果用户没有给定语言相关的名称，则（如同以前）由数据库显示标准名称。

刀具管理调试工具不分析条目 NameOfBufferPlaceFrom，并总是使用由刀具数据库和国家语言相关的 INI 文件给出的名称。

国家语言相关的 INI 文件中的条目

这里描述的条目必须用户自己进行。它不由 HMI 高级写在 INI 文件中。

国家语言相关的 INI 文件为 patm_xx.ini，位于 ../hmi_adv/language。用户定义的文件 patm_xx.ini 位于 ../user/language。

INI 文件 paramtm.ini 和附属的描述文件 paramtm.txt 位于 ../hmi_adv。用户定义的文件 paramtm.ini 位于 ../user。这涉及到预设，条目不需要在用户定义的 paramtm.ini 中。

文件	paramtm.ini
段落	[TMMODES]
条目	NameOfBufferPlaceFrom=DB (预设)
	和
文件	patm_xx.ini
段落	[BufferPlace_VISName]
条目	Standardname="语言相关的文本"
举例:	[BufferPlace_VISName]
	Spindel1="Hauptspindel"
	Greifer1="1. Greifer"
	...

为了从 INI 文件显示一个输入的文本，必须进行一次语言切换或者重新启动 HMI 高级。

在 HMI 刀具管理中显示中间存储器名称

在所有刀具管理和刀具管理调试涉及的画面中，显示中间存储器刀位的国家语言相关的名称。如果在相应的 INI 文件中没有条目，则显示来自刀具数据库的标准名称。

相关的画面/ 功能：

- 刀具管理：刀库列表，在显示中间存储器时
- 调试刀具管理：中间存储器

6.3.3 装载刀位的国家语言相关的名称

功能

装载刀位和其名称由用户通过刀具管理 (IW) 的调试工具在装载刀位画面中给定。由此给定的名称位于刀具管理数据库中，不在语言 DLL 中。

对此对于数据库中配置的装载刀位，必须由用户将相应的名称文本输入在国家语言相关的刀具管理 INI 文件中。

每个装载刀位有两个名称：

- 标准名称，用其内部（刀具数据库）工作。
- 一个附属的国家语言相关的名称，这显示在操作界面上。

如果用户没有给定语言相关的名称，则由刀具数据库显示标准名称。

这也适用于装载刀库中的第一个刀位，这自动给定：

通过刀具管理 (IW) 的调试工具，对于装载刀库的第一个刀位，自动将一个条目建立在刀具数据库中。这在第一次跳转到带有原始数据库的装载刀位时进行。该刀位总是必须存在，因此无法再删除。

它得到一个内部标准名称 “FirstLoadingPoint”，带下列特性：

- 在所有供货范围中获取的文件 patm_xx.ini 中有用于装载刀位“FirstLoadingPoint”的符合标准的一个条目，位于 [LoadLocation_VISName] 下。
- 在刀具管理 (IW) 的调试工具装载刀位画面中，对于第一个装载刀位，由 patm_xx.ini 的语言相关的文本也显示在选择栏“名称”中。

已存在的数据库 在已较长运行的系统中：

在以前的 HMI 版本中，对于第一个装载位置给定名称“主轴装载位置”或“手动装载位置”（在针对该时间点设置的国家语言中）。

如果在第一次启动刀具管理或附属的调试工具 (IW) 时，通过这里描述的功能识别，则由 “FirstLoadingPoint” 替代数据库中已有的名称。

国家语言相关的 INI 文件中的条目

这里描述的条目必须用户自己进行。它不由 HMI 高级写在 INI 文件中。

例外: `FirstLoadingPoint="Beladestelle Hand"`

国家语言相关的 INI 文件为 `patm_xx.ini`, 位于 `.../hmi_adv/language`。用户定义的文件 `patm_xx.ini` 位于 `.../user/language`。

文件	<code>patm_xx.ini</code>
段落	<code>[LoadLocation_VISName]</code>
条目	<code>Standardname="语言相关的文本"</code>
举例:	<code>[LoadLocation_VISName]</code> <code>FirstLoadingPoint="Beladestelle Hand"</code> <code>Beladestation1="Hauptbeladestation"</code> ...

为了从 INI 文件显示一个输入的文本, 必须进行一次语言切换或者重新启动 HMI 高级。

在 HMI 刀具管理中显示装载刀位名称

在所有刀具管理和刀具管理调试涉及的画面中, 显示装载刀位的国家语言相关的名称。如果在相应的 INI 文件中没有条目, 则显示来自刀具数据库的标准名称。

相关的画面/ 功能:

- 刀具管理:
 - 装载
 - 卸载
 - 空位查找
 - 定位
- 调试刀具管理: 装载刀位

6.3.4 刀位类型的国家语言相关的名称

功能

刀位类型和其名称由用户通过刀具管理 (IW) 的调试工具在“刀位类型”画面中给定。由此给定的名称位于刀具管理数据库中，不在语言 DLL 中。这导致在不同语言中的不可用。

通过新的功能，用户可以管理国家语言相关的刀位类型名称。

对此对于数据库中配置的刀位类型，将相应的名称文本输入在国家语言相关的刀具管理 INI 文件中。

每个刀位类型有两个名称：

- 标准名称，用其内部（刀具数据库）工作。
- 一个附属的国家语言相关的名称，这显示在操作界面上。

如果用户没有给定语言相关的名称，则由数据库显示标准名称。

在用刀具管理提供的原始数据库中得到刀位类型“普通”。对于该刀位类型有下列特殊处理：

- 在所有供货范围中获取的文件 patm_xx.ini 中有用于刀位类型“普通”的符合标准的一个条目，位于 [Placetype_VISName] 下。
- 在刀具管理 (IW) 的调试工具刀位类型画面中，对于刀位类型“普通”，由 patm_xx.ini 的语言相关的文本也显示在选择栏“名称”中。

国家语言相关的 INI 文件中的条目

这里描述的条目必须用户自己进行。它不由 HMI 高级写在 INI 文件中。

例外：normal="Platztyp normal".

国家语言相关的 INI 文件为 patm_xx.ini，位于 ../hmi_adv/language。用户定义的文件 patm_xx.ini 位于 ../user/language。

文件	patm_xx.ini
段落	[Placetype_VISName]
条目	Standardname="语言相关的文本"
示例	[Placetype_VISName]
	normal="Platztyp normal"
	SmallPlaceType="klein"
	...

为了从 INI 文件显示一个输入的文本，必须进行一次语言切换或者重新启动 HMI 高级。

在 HMI 刀具管理中显示刀位类型名称

在所有刀具管理和刀具管理调试画面中，显示刀位类型的国家语言相关的名称。如果在相应的 INI 文件中没有条目，则显示来自刀具数据库的标准名称。

相关的画面/ 功能：

- 刀具管理：
 - 刀具详细信息
 - 新刀具
 - 刀具目录
 - 刀具箱
 - 空位查找
- 调试刀具管理：
 - 刀库配置
 - 刀位类型

刀具目录/刀具箱

内部对于刀具目录/刀具箱，导入分配的刀位类型的标准名称。

在刀具目录/刀具箱画面中显示用于刀位类型的国家语言相关的名称。如果存在，不使用来自刀具数据库的标准名称。

代码载体

刀库刀位类型名称，通过对话框变量 T8 写在代码载体芯片上，总是与来自刀具数据库的刀库刀位类型标准名称相符。

或者可以通过对话框变量 T12 将刀库刀位类型编号写在代码载体上。

T8 和 T12 也接受。在读取芯片时进行一次横向检查刀库刀位类型名称/编号。

6.4 刀具的任务准备

前提条件

设置刀具任务准备位于 `paramtm.ini` 和 `language\patm_xx.ini` 的段 `[BatchTools]` 中。

过滤器仅作用在位 4 在 MD 18080: `TOOL_MANAGEMENT_MASK` 设置情况下。

最多可以定义 6 个查找过滤器。对于每个过滤器可以如下规定：软键文本，结果标题，查找标准，选择结果列表类型和附加数据。

文件 `..user\paramini.out` 包含读取参数时找到的故障信息。

6.4.1 一般适用于所有过滤器的设置

结果列表颜色

可以改变结果列表颜色。此时应避免不可读或者不可辨认的颜色组合。

对于下列列表元素必须规定颜色：

- 未选择的文本
- 未选择的背景
- 光标选择的文本
- 光标选择的背景
- 任务-选择-选择的文本
- 任务-选择-选择的背景
- 任务-选择-和光标选择的文本
- 任务-选择-和光标选择的背景

参数由 8 个颜色数据组成，用逗号隔开。每个单个颜色数据分配一个十六进制值，由 8 个符号组成，

该值有下列结构：

`SS BB GG RR` 带 `SS=系统`，`BB=蓝色`，`GG=绿色`，`RR=红色`。

颜色示例：

- `80 00 00 08`: Windows 文本
- `80 00 00 05`: Windows 背景
- `80 00 00 0E`: Windows 标记的文本
- `80 00 00 0D`: Windows 标记的背景
- `80 00 00 09`: Windows 激活的窗口，标题文本
- `80 00 00 02`: Windows 激活的窗口，标题背景
- `00 FF FF 00`: 淡蓝色
- `00 00 FF 00`: 绿色

- 00 FF 80 00: 蓝绿色

颜色设置示例，参见 [BatchTools], "General settings which apply to all filters" 在 paramtm.txt 中。

单个任务元素的状态显示位图

位图名称或位图本身可以由用户自身的位图替换。自身的位图，给出了目录“user”。

位图设置示例，参见 [BatchTools], "General settings which apply to all filters" 在 paramtm.txt 中。

举例：

BatchFilterEIBUnTUnBitmap = pfbbutu.bmp

BatchFilterEIBUnTSeBitmap = pfbbutu.bmp

BatchFilterEIBSeTUnBitmap = pfbbstu.bmp

BatchFilterEIBSeTSeBitmap = pfbbsts.bmp

BatchRunEIWaitingBitmap = pbbwait.bmp

BatchRunEIInWorkBitmap = pbbwork.bmp

BatchRunEIOKBitmap = pbbok.bmp

BatchRunEIErrorBitmap = pbberr.bmp

典型字符的宽度

根据国家语言可以规定一个符号，使用它的宽度来计算一个规定的符号数的栏宽度。参数编写人员应该使用一个宽度符号，在欧洲一般使用 "X" 或 "A"。

举例：

Paramtm.ini,[BatchTools]:

CharToGetColWidthPerCharacter=CharToGetColWidth ;国家语言相关

language\patm_gr.ini, [BatchTools]:

CharToGetColWidth="A"

刀具名称栏宽度

可以设置刀具名称栏宽度，因为通常不使用全部 32 个字符数。

举例：

Paramtm.ini,[BatchTools]:

ResultDisplayedNumberOfToolnameCharacters=18

刀具状态位

可以参数化哪些刀具状态位显示在结果列表中。此外国家语言相关的字母可以规定用于标题和列表行的位。

举例：

Paramtm.ini,[BatchTools]:

ResultToolStatusColumnsEnable=1111100100110000

1: 显示, 0: 不显示。 位 1 至 16,

位 1 是刀具状态中的最低值, 位于该符号序列的左侧。

ResultToolStatusColumnsHeaderText=<Empty>

刀具状态栏中的标题文本, 国家语言相关

ResultToolStatusColumnsListText=<Empty>

刀具状态栏中的数据文本, 国家语言相关

ToolStatusColHeaderText="123456789ABCDEFGG"

标题; 位 1 至 16, 位 1 是刀具状态中的最低值, 位于该符号序列的左侧。

ToolStatusColListText="123456789ABCDEFGG"

数据; 位 1 至 16, 位 1 是刀具状态中的最低值, 位于该符号序列的左侧。

用户权限

参与的软键的用户权限可以在 paramtm.ini, 段 [ACCESSLEVEL], 条目 "SKB..." 中设置。

举例：

[ACCESSLEVEL]

SKBATCH=7 ; SK F 过滤器列表

SKFILTER1=7 ; SK 过滤器 1

SKFILTER2=7 ; SK 过滤器 2

SKFILTER3=7 ; SK 过滤器 3

SKFILTER4=7 ; SK 过滤器 4

SKFILTER5=7 ; SK 过滤器 5

SKFILTER6=7 ; SK 过滤器 6

SKBMAGFILTER=7 ; SK 刀库选择

SKBATREACT=7 ; SK 批处理功能“再次激活”

SKBATTOCABIN=7 ; SK 批处理功能“在箱中”

SKBATDELTOOL=7 ; SK 批处理功能“删除”

SKBATUNLOAD=7 ; SK 批处理功能“卸载”

SKBFILTERACT=7 ; SK 批处理功能“更新”

SKBATLOAD=7 ; SK 批处理功能“装载”

SKBATLIST=7 ; 软键用于控制任务处理

6.4.2 适用于过滤器的设置

查找标准

查找标准在“n_FindConditio”中（以 n=1 至 6）通过过滤器设置。

允许最多 8 个条件，用 "," 隔开。用逻辑和链接。在条件中不允许多次出现日期。

每个条件由 3 部分组成：

1. 1. 数据，适用条件的数据
2. 条件
3. 比较值

举例：

1_FindCondition = TC_TP8 && 20, A_TOOLMN > 0

过滤器 1 过滤 NC 中满足下列条件的刀具：

（（设置预警位（位 5 从位 1 至 16 (2hoch(5-1)=16)）））
 或
 （设置禁用位（位 3 (2hoch(3-1)=4)））
)
 与
 ((刀库号 > 0 表示 ""装载的刀具"))

过滤器标准

下列 NC 中的数据可以是过滤器标准：

刀具数据

TC_TP1:D 号

TC_TP2:刀具名称

TC_TP3: 在左半位刀具尺寸

TC_TP4: 在右半位刀具尺寸

TC_TP5: 在上半位刀具尺寸

TC_TP6: 在下半位刀具尺寸

TC_TP7: 刀具刀位类型

TC_TP8: 刀具状态

TC_TP9: 监控类型

TC_TP10: 替换刀具查找

TC_TP11: 刀具信息 / 姊妹刀具顺序

A_TOOLMN: 刀库号码

A_TOOLMLN: 刀库编号

P_TOOLND: 刀沿数目

刀具刀沿参数：

TC_DP1 至 TC_DP24

刀具刀沿监控参数

TC_MOP1:刀具寿命预警

TC_MOP2:实际刀具寿命

TC_MOP3:工件数预警极限

TC_MOP4:实际工件数

TC_MOP5:磨损预警极限

TC_MOP6: 剩余磨损

TC_MOP11: 使用寿命标准值

TC_MOP13: 件数标准值

TC_MOP15: 磨损标准值

刀具-OEM-数据

TC_TPC1 至 TC_TPC10

刀具 OEM 数据必须在 NC 上激活，号码必须为 NC 上许可的。

刀具刀沿-OEM-数据

TC_DTPC1 至 TC_DTPC10

刀具刀沿 OEM 数据必须在 NC 上激活，号码必须为 NC 上许可的。

刀具刀沿监控-OEM-数据

TC_MOPC1 至 TC_MOPC10

刀具刀沿监控 OEM 数据必须在 NC 上激活，号码必须为 NC 上许可的。

条件

可以使用下列条件：

== 等于

< 小于

> 大于

<= 小于等于

>= 大于等于

&& 屏幕窗口和数据之间的位方式“和”：仅允许用于类型 WORD 和 DOUBLEWORD 的运算域；单个的结果位通过“或”链接。

如果在刀具状态上使用该条件，则刀具可以通过确定设置的（“和”）刀具状态位找到。如果同时查询多个设置的位，则一个设置位已可满足刀具显示在找到结果中。

比较值

对于字符串运算域（例如刀具名称）"==" 为单个允许的比较运算符。下列值范围适用于：

字符串：对于 TC_TP2，最大 32 个字符，前后没有空格

0 ... 65535: 对于其他的 TC_TP 数据

双：对于所有其他的数据

附加数据

在结果列表中每个过滤器最大可以显示一个带有附加数据的栏。

为此每个过滤器允许有三个设置：

- n_FindResultAddColumnText
栏标题文本或标题文本上参考（对于国家语言相关）
- n_FindResultAddColumnDisplayedNumberOfCharacters
栏宽度（以字符为单位）
- n_FindResultAddColumnBtss
BTSS-项根据 BTSS-文件 (mmc2\btss_gr.hlp)

示例 1:

Paramtm.ini,[BatchTools]:

1_FindResultAddColumnText=<empty>

; 没有附加栏

1_FindResultAddColumnText=R1AddCol

; 国家语言相关

1_FindResultAddColumnDisplayedNumberOfCharacters=8

1_FindResultAddColumnBtss="/Tool/User/data[u#TOA#,c3,#TNO#](!|!%ld)"

language\patm_gr.ini, [BatchTools]:

R1AddCol="Zusatz-Daten Suche1"

示例 2:

"/Tool/User/data[u#TOA#,c2,#TNO#](!|!d%.#RES#f)"

刀具-OEM-参数 2，浮点描述，标准小数点后位数

示例 3:

"/Tool/User/data[u#TOA#,c3,#TNO#](!|!%ld)"

刀具-OEM-参数 3，整数描述，

示例 4:

"/Tool/MagazineDescription/userData[u#TOA#,c#MAG,1](|)"

刀库 OEM 参数 1

通配符

允许下列通配符：

#TOA# TOA-号 (当前通道的)
 #TNO# 内部 T-号 (找到的刀具)
 #MAG# 刀库号 (找到的刀具)
 #RES# 小数点后的位数标准值

通配符由当前刀具数据或者一般设置替换。

最多允许一个 1 BTSS-项。

由 BTSS-项建立内部 BTSS 多变量存取建立。

结果数据前的数据分隔符 "|"必须由数据 "("建立。

Btss 项必须在 "" 内，尤其是当包含一个格式数据（例如 "%d"），这必须在 "" 内。

通过操作界面软件无法检测句法。参数负责正确的句法。

说明

参数化错误时不再显示结果列表或者可能出现后续错误。

过滤器标题和软键文本

对于每个过滤器，可以参数一个标题和一个软键文本，和国家语言相关。

为此每个过滤器允许有两个设置：

- n_FindResultHeadlineText
过滤器标题或参考
- n_FindSoftkeyText
过滤器软键文本或参考（文本中的一个双空格定义换行位）

举例：

Paramtm.ini,[BatchTools]:

1_FindResultHeaderText = R1HL

1_FindSoftkeyText = F1SK

language\patm_gr.ini, [BatchTools]:

R1HL ="达到预警极限或禁用"

F1SK = "预警或禁用"

结果列表类型

每个过滤器可以选择结果列表类型。这确定通过软键提供哪些任务功能。

为此每个过滤器允许有一个设置：`n_ResultListType`

结果列表类型，值范围：

0: 标准列表（预设）用于卸载，删除，箱中，再次激活

1: 装载列表用于装载，再次激活

举例：

Paramtm.ini,[BatchTools]:

1_ResultListType = 0 ;0 = 标准列表

刀库上可限制的过滤器

对于每个过滤器可以选择是否可在打一个刀库上限制。这应该已出现在过滤器标题中。

每个过滤器允许有一个设置：`n_FindLimitedToCurMagazine`

过滤器可限制为一个刀库，值范围：

真: 找到结果限制为当前刀库

假: （预设）找到结果不限制为当前刀库

举例：

Paramtm.ini,[BatchTools]:

1_FindLimitedToCurMagazine=true

; "True", "False"; 限制当前刀库，

在再次激活时定位

对于每个过滤器可以选择在任务功能“再次激活”时是否要将刀具定位到一个装载位置。
为此每个过滤器允许有一个设置：`n_ReactivatePositioningMode`

在再次激活时定位，值范围：

- 0 未定位
- 1 操作人员通过整个任务询问是否要实现定位
- 2 (预设) 总是定位

举例：

Paramtm.ini,[BatchTools]:

1_ReactivatePositioningMode=2 ;总是

PI TSEARCH 的参数

说明

不确保支持下一个版本。

对于每个过滤器，可以给定过滤使用的 PI TSEARCH 的参数。

这些设置对错误非常敏感。不允许插入空格；位数必须准确遵守；符号序列必须在 "" 内。

为此每个过滤器允许有一个设置：`n_FindPiSearchPar`

- PI SEARCH 的 8 个参数
- 作为刀库设置通配符可以使用 `#Mag#`。在通配符位上也可以给定五位数的刀库编号，用于从一至-刀库范围。
- 如果在第 8 参数中设置值等于 "2"，则过滤器标准对于刀沿专用数据，也对于多刀沿刀具都正确适用（从刀具版本 NCK.P6_43 和 NCK.P5_20.4 起）。

举例：

Paramtm.ini,[BatchTools]:

1_FindPiSearchPar="#Mag#,-0001,#Mag#,-0001,00000,00001,1,2"

6.5 磨削刀具和刀具专用磨削数据

一览

HMI 高级刀具管理这样改变，可以显示和更改磨削刀具的“刀具专用磨削数据”。

这些数据通过 BTSS-模块 TG 与 NC 互换；这完全符合 NC 变量 \$TC_TPG1 至 \$TC_TPG9。

有关刀具详细信息基本画面，刀具详细信息刀沿数据画面和刀具详细信息监控数据画面中通过软键扩展的其他信息位于：

文献：HMI 高级操作手册

参数化默认值

参数化刀具专用磨削数据默认值在设立刀具时在文件 "paramtm.ini" 的段 [DEFAULT_SETTINGS] 中实现。

说明

```
[DEFAULT_SETTINGS]
;!!! 在设立时预占用磨削专用的刀具数据：
;!!! 如果机床以英寸/毫米转换
;!!! ($MN_CONVERT_SCALING_SYSTEM=1) 工作必须给定长度
;!!! 单位!!!
;下列默认值（TOOLGRIND..., 当长度
;单位相关），相关的基本长度单位
;给定：
;0 = mm (预设)
;1 = 英寸
TOOLGRIND_Default_Length_Unit=0
;主轴号码（如 $TC_TPG1）
TOOLGRINDspinNoDress=1
;链接规定（如 $TC_TPG2）
TOOLGRINDconnectPar=1050629
;1050629 二进制： 0000 0000 0001 0000 0000 1000 0000 0101
;Bit0 =1 = Typ
;Bit2 =1 = Geo-L1
;Bit11 =1 = Wear-L1
;Bit20 =1 = Base-L1
```

;最小写半径 (如 \$TC_TPG3)
TOOLGRINDminToolRadius=0
;最小写宽度 (如 \$TC_TPG4)
TOOLGRINDminToolWide=0
;当前砂轮宽度 (如 \$TC_TPG5)
TOOLGRINDactToolWide=0
;最大砂轮转速 (如 \$TC_TPG6)
TOOLGRINDmaxRotSpeed=0
;最大砂轮圆周速度 (如 \$TC_TPG7)
TOOLGRINDmaxTipSpeed=0
;斜砂轮的倾斜角度 (如 \$TC_TPG8)
TOOLGRINDinclAngle=0
;SUG 补偿参数 (如 \$TC_TPG9)
TOOLGRINDparamNrCCV=3

说明

至今的 HMI-高级功能“更改刀具”未改变。为此对于磨削道具在更改刀具类型时将大部分刀具数据通过 HMI 设置为“0”。

磨削专用的刀具数据不设置为“0”，而是通过 NCK 处理。

6.6 考虑设置英制/公制

一览

对于刀具数据库（刀具箱和刀具目录）和代码载体，可以用英制或者公制（mm）进行工作。

在文件 `paramtm.ini` 的段 `[TMMODES]` 中有关长度单位的特性通过用于刀具数据库或代码载体的条目 `DATABASE_LENGTH_UNIT` 和 `CODECARRIER_LENGTH_UNIT` 设置。

刀具数据库

NCK 带数据转换 英制/公制：

`[TMMODES]`

；!!! 注意：

；仅相关于这里描述的设置，

；当 NCK 中机床数据 `$MN_CONVERT_SCALING_SYSTEM=1` 情况下

；设置。

；`DATABASE_LENGTH_UNIT` 条目才分析，在

；刀具数据库还未输入单位情况下。

；也在标准情况下！

；-1 表示未确定英制/公制在刀具

；数据库（预设）。

；设置 `$MN_CONVERT_SCALING_SYSTEM=1` 在 NCK 中表示，

；要通过英制/公制转换工作。

；为此用户必须确定要以哪些单位

；位于在数据库中。通过设置 -1 不实现

；所有用于数据库激活的软键

；禁用。

；0 表示公制（mm）

；对于 1. 打开一个刀具数据库通过 WZV 带

；英制/公制（mm）区别，确定相关的数据

；在刀具数据库中以公制（mm）形式存在。

；数据库扩展一个条目，这就是说，

；相关的刀具数据在数据库中以

；单位 mm 存在。

；数据库和 NCK 之间的数据传输

；将相应考虑这些。

; 1 = 英寸
 ; 对于 1. 打开一个刀具数据库通过 WZV 带
 ; 英制/公制 (mm) 区别, 确定相关的数据
 ; 在刀具数据库中以英制 (inch) 形式存在。
 ; 数据库扩展一个条目, 这就是说,
 ; 相关的刀具数据在数据库中以
 ; 英制 (inch) 存在。数据库和 NCK 之间的数据传输
 ; 将相应考虑这些。

DATABASE_LENGTH_UNIT=-1

; 如果 DATABASE_LENGTH_UNIT 包含设置 0 或 1
 ; 适用于下列: 对于 1. 启动通过新的 WZV, 带
 ; 英制/公制区别, 数据库得到一个新的
 ; 条目, 该条目将来作为数据库单位
 ; 描述。

处理时数据库中输入的单位起决定性作用。数据以确定的单位在刀具箱中写和读。

名称	类型	尺寸
InfoKey_Name	文本	255
InfoKey_Index	整数	2
Info_String	文本	255
Info_Num	双	8

表格 6-1 数据表格内容以 mm 或 inch:

InfoKey_Name	InfoKey_Index	Info_String	Info_Num
"BasicLenUnit"	0	"mm"	0
"BasicLenUnit"	0	"inch"	1

刀具管理激活

如果刀具管理在运行中和激活，有下列可能性用于 inch/mm 转换：

1. 例如通过机床中软键转换。
然后调用刀具管理。在选择后所有画面中相关的数据都以新的单位显示。
2. 通过更改机床数据 10240 转换：\$MN_SCALING_SYSTEM_IS_METRIC 和电源复位或者机床操作面板复位到调试。
然后调用刀具管理。在选择后所有画面中相关的数据都以新的单位显示。
3. 通过更改机床数据 10240 转换：\$MN_SCALING_SYSTEM_IS_METRIC 和电源复位。
在复位期间已根据刀具管理切换。
 - 刀具详细信息画面：实现跳回到一个附属的列表画面
 - 列表画面：数据被相应更新。如果需要（由于复位）进行一次画面切换。
 - 目录/刀具箱画面带有相关数据：显示的画面通过 <取消> 退出。
4. 通过更改机床数据 10240 转换：\$MN_SCALING_SYSTEM_IS_METRIC。
然后在已选择的刀具管理中调用刀具管理和电源复位。这相应于点 3。
5. 通过更改机床数据 10240 转换：\$MN_SCALING_SYSTEM_IS_METRIC。
然后在已选择的刀具管理中调用刀具管理和机床操作面板复位。
 - 刀具详细信息画面和列表画面：由于在这些画面中各个值在输入后立即写入，也一直更新，数据立即以新的单位显示。
 - 目录/刀具箱画面带有相关数据：由于输入的数据首先通过按下软键“存储器功能”（例如 <确定>，<新刀沿> 完整写入在数据库中，则新的单位在这些操作后才激活。

参见

代码载体数据描述 (页 274)

6.7 调试代码载体

6.7.1 代码载体数据描述

一览

对于代码载体系统： 参见各个刀具识别系统的功能描述。

例如一个代码载体可以通过 V.24 接口连接。 如果机床通过一个自身的代码载体系统（刀具识别系统）支配，则也可进行该系统的调试。

说明

自 SW 6.3 起代码载体系统只能通过 SinTDC 驱动，不再可以直接驱动了。

WToolIdSys = SinTDC

另见： /FBTC/ SINUMERIK 刀具数据通信 SinTDC。

为了激活刀具管理，代码载体系统必须输入在文件 ...\user\mmc.ini 中。

在文件 ...\user\paramtm.ini 中进行有关英制/公制的设置并接受刀具状态位。

说明

不得改变目录“mmc2”中的所有 INI 文件。

对于每个制造商专用服务器 ("exe-File") 有一个附属的 "ini-文件"。 在该配置文件中制造商专用的数据通过代码载体系统输入。 其中包含的参数描述位于代码载体制造商手册或者相应的用于刀具识别系统的功能描述中。

激活代码载体系统

在文件 ...\user\mmc.ini 中激活连接的代码载体系统（例如 WToolIdSys=Ballu）。

```
-----
[ToolMgmt]
-----
```

WToolIdSys=0 ; 或者 Ballu

; 识别代码载体系统

; 0 表示“没有代码载体激活”

; 制造商名称仅规定前面 5 个字符！

; Ballu 表示： 公司 Balluff 的代码载体激活

WToolIdSysKonv=wkonvert.txt

; 使用的配置文件名称用于

```

; 代码载体格式。
; 文件位于目录 ...\add_on 或 ...\user 中。
[TIS]
; 刀具识别系统
; EOT 用于代码载体数据
TIS_EOT=0x2F2F
; 这里数据最后识别符必须输入在代码载体上
;。
;

```

在文件 paramtm.ini 中设置英制/公制

之后打印的提纲位于 paramtm.ini 中。将 paramtm.ini 的注释合并到用于 paramtm.txt 的新的软件版本中。

所需的条目一般在...\user\paramtm.ini 中，为此在下一 HMI 软件更新时继续保留。

如果安装的软件版本的 paramtm.ini 或 paramtm.txt 包含下列条目中的一个 (型式 1 或型式 2)，则可以通过 ...\user\paramtm.ini 中的条目根据描述使用这些功能。否则适用于预设。

型式 1:

```

[TMMODES]
...
; 考虑设置英制/公制，涉及到代码载体
; -1 = 英制/公制已忽略 (预设)。
; 实现代码载体和 NCK/MMC 之间的数据传输
; 不考虑英制/公制。特性和以前一样。
; 0=mm
; 由此相关数据的数据保存
; 要以单位 mm 保存在代码载体上
; 或者以此实现。如果 NCK 中设置“英寸”，则
; 所有软键禁用，
; 启动代码载体。
; 1=inch
; 由此相关数据的数据保存
; 要以单位 inch 保存在代码载体上
; 或者以此实现。如果 NCK 中设置“mm”，则
; 所有软键禁用，
; 启动代码载体。
DATABASE_LENGTH_UNIT=-1

```

型式 2:

[TMMODES]

...

```

;考虑设置英制/公制，涉及到代码载体
;!! 注意： 这里描述的设置仅
;相关于，在 NC 中机床数据
;$MN_CONVERT_SCALING_SYSTEM=1 设置情况下。
;涉及到一个老的 NC，不带
;转换功能 英制/公制，或者
;$MN_CONVERT_SCALING_SYSTEM=0，则 NV 工作不带
;英制/公制转换。基于此原因
;相关代码载体也不进行转换!!!
;-1 = 英制/公制已忽略 (预设)。
;实现代码载体和 NCK/MMC 之间的数据传输
;不考虑英制/公制。特性和以前一样。
;0=mm。
; 由此相关数据的数据保存
;要以单位 mm 保存在代码载体上
;或者以此实现。如果 NCK 中设置“英寸”，则
;所有软键禁用，
;启动代码载体。
;1=inch。
; 由此相关数据的数据保存
;要以单位 inch 保存在代码载体上
;或者以此实现。如果 NCK 中设置“mm”，则
;所有软键禁用，
;启动代码载体。
CODECARRIER_LENGTH_UNIT=-1
    
```

刀具状态设置

如果安装的软件版本的 paramtm.ini 或 paramtm.txt 包含下列条目，则可以通过 ...user\paramtm.ini 中的条目根据描述使用这些功能。否则适用于预设。

```

;刀具状态： 如果将一个刀具从 NCK 去除并
;传输到一个“外部”媒介上（刀具箱，
;代码载体，SINCOM），则可以通过下列屏幕
;规定保存哪些刀具状态位
;。
    
```

```
; 代码载体: 由于标准-wkonvert.txt 用于刀具状态
; 输入 1 字节, 以前在代码载体上最大写 92
; 获得 CODECARRIER_TOOLSTATE_MASK
; 默认值 92。
; 如果扩展 CODECARRIER_TOOLSTATE_MASK 值,
; 必须对 wkonvert.txt 中的对话框变量 T9 的尺寸
; 相应匹配。

; 1=激活的刀具
; 2=已允许
; 4=已禁用
; 8=已测量
; 16=达到报警极限
; 32=在交换
; 64=固定位置编码
; 128=已使用
; 256=缓存器中刀具
; 512=已禁用, 忽略 (由于 PLC)
; 1024=外部 (卸载)
; 2048=内部 (装载)
; 4096=一般刀具 (一直在 NCK 中)
; 8192=
; 16384=
; 默认是 4828 (4+8+16+64+128+512+4096),
; 默认用于代码载体 92 (4+8+16+64)
...
CODECARRIER_TOOLSTATE_MASK=92
```

6.7.2 说明文件的结构

说明文件的结构

代码载体上所有数据都按照一定顺序存放。这在代码载体调试时定义。为此刀具管理读取这些数据流，在说明文件的型式中有一个转换规定。说明文件由刀具和刀沿对话框数据组成。仅这些对话框数据可以由刀具管理实际处理。所有代码载体上的其他数据不允许分配对话框变量，这些变量不处理。然而 OEM-应用程序也可以获取这些数据。

说明文件可以作为 ASCII-文件用一个标准编辑器建立。文件名称必须在 mmc.ini 的 **WToolldSysKonv = wkonvert.txt** 中输入。

前提条件

wkonvert.txt 最低要求：

- 名称
- 刀位类型
- 子类型
- 刀沿数（如果存在刀沿）
- 刀具尺寸（可以离开尺寸部分，该尺寸通过 paratm.ini 隐藏。）

SHOW_TOOLSIZE_ONLY_LEFT_RIGHT=0

SHOW_TOOLSIZE_COMPONENTS=left=True, right=True, top=True, bottom=True

刀具对话框数据

刀具对话框数据如下定义：

对话框变量	数据类型	名称	分配 \$TC...
T1	字符串	刀具名称, 最大 32 位	\$TC_TP2
T2	整数	D 号	\$TC_TP1
T3	整数	刀沿数目	\$P_TOOLND[tnr] tnr=刀具编号
T4	整数	左半位的刀具尺寸	\$TC_TP3
T5	整数	右半位的刀具尺寸	\$TC_TP4
T6	整数	上半位的刀具尺寸	\$TC_TP5
T7	整数	下半位的刀具尺寸	\$TC_TP6
T8	字符串	刀库位置类型	\$TC_TP7*
T9	整数	刀具状态	\$TC_TP8
T10	整数	刀具监控类型	\$TC_TP9
T11	整数	刀具查找类型	\$TC_TP10
T12	整数	刀库位置类型 至今刀库刀位类型名称作为字符串通过对话框变量 T8 存放。刀位类型编号和刀位类型名称之间的分配通过刀具管理数据库实现。	\$TC_TP7

* 这里存放的字符串时一个 HMI-内部刀位类型，该刀位类型在 \$TC_TP7 中分配值。该文本通过 WZV 调试确定并存放在数据库中。

说明

如果更换多个机床之间芯片，在使用 T12 时有下列规定。

刀位类型名称必须按照相同顺序（用相同的刀位编号）存放在所有这些机床上。

改变转换文件将使旧的代码载体不再可读！

代码载体芯片/SINTDC:

如果用户在转换文件 wkonvert.txt 中定义对话框变量 T11（用于“刀具查找类型，\$TC_TP10”），则 NCK 中的值写在芯片上，在读取时再写入 NCK 中。如果缺失 T11，则再读取芯片后在 \$TC_TP10 中值为 0。

刀沿对话框数据

刀沿对话框数据如下定义：

对话框变量	数据类型	名称	分配 \$TC...
C1	整数	子类型	\$TC_DP1
C4	整数	刀沿位置	\$TC_DP2
		刀具长度几何尺寸补偿	
C5	双	长度 1	\$TC_DP3
C6	双	长度 2	\$TC_DP4
C7	双	长度 3	\$TC_DP5
		刀具半径几何尺寸补偿	
C8	双	长度 1	\$TC_DP8
C9	双	长度 2	\$TC_DP9
C10	双	半径 1	\$TC_DP6
C11	双	半径 2	\$TC_DP7
C12	双	角 1	\$TC_DP10
C13	双	角 2	\$TC_DP11
		刀具长度磨损补偿	
C14	双	长度 1	\$TC_DP12
C15	双	长度 2	\$TC_DP13
C16	双	长度 3	\$TC_DP14
		刀具半径磨损补偿	
C17	双	长度 1	\$TC_DP17
C18	双	长度 2	\$TC_DP18
C19	双	半径 1	\$TC_DP15
C20	双	半径 2	\$TC_DP16
C21	双	角 1	\$TC_DP19
C22	双	角 2	\$TC_DP20
		基本/适配器尺寸-刀具长度补偿	
C23	双	基本长度 1	\$TC_DP21
C24	双	基本长度 2	\$TC_DP22
C25	双	基本长度 3	\$TC_DP23
C26	双	自由切削角度	\$TC_DP24
C27	整数	超头使用	\$TC_DP25
C28		刀沿号码 - 用于变量编址	-
C29*	整数	使用寿命, 单位: 分钟	\$TC_MOP2
C30*	整数	预警极限值, 刀具寿命以分计算	\$TC_MOP1
C31	整数	待完成件数	\$TC_MOP4
C32	整数	待完成件数预警极限值	\$TC_MOP3
C33	双	使用寿命, 单位: 分钟	\$TC_MOP11
C34	整数	标准工件数	\$TC_MOP13

对话框变量	数据类型	名称	分配 \$TC...
C35	双	磨损预警极限	\$TC_MOP5
C36	双	磨损	\$TC_MOP6
C37	双	标准磨损	\$TC_MOP15
C38*	双	使用寿命, 单位: 分钟	\$TC_MOP2
C39*	双	预警极限值, 刀具寿命以分计算	\$TC_MOP1
C40*	双	件数	\$TC_MOP4
C41	双	工件数预警极限	\$TC_MOP3
C42	双	标准工件数	\$TC_MOP13

对话框变量 C2 和 C3 仅内部管理。

说明

C38 和 C39 仅可用于 C29 和 C30 之一。C40, C41 和 C42 仅可用于 C31, C32 和 C34 之一。

对于代码载体, 现在也可使用刀具用户参数和新的监控参数。对于文件 `wkonvert.txt`, 提供下列新的对话框变量:

A1 - A10: 用户刀具数据 (参见 `$TC_TPCx[t]`)

U1 - U10: 用户刀沿数据 (参见 `$TC_DPCx[t,d]`)

S1 - S10: 用户监控数据 (参见 `$TC_MOPCx[t,d]`)

对于对话框变量 A, U 和 S, 数据类型定义为“双”。

数据类型

确定对话框变量数据类型:

- 整数: 值范围 -32768 至 32767
- 双: 浮点双字节
- 字符串: ASCII-字符序列

关键字

通过代码载体说明文件可以分配代码载体数据到对话框数据。作为 ASCII-文件可以用标准编辑器建立或改变。代码载体说明文件以行方式建立, 此时每行通过下列 **关键字** 之一导入:

引号

'(引号) 是一个注释开始。通读下一个符号直至行末尾。

示例: '这是一个注释。¶

说明

引号形式作为注释名称仅用在代码载体说明文件中。否则带“-”的注释开始将被导入。

Datalen

DATALEN=CONST | VARIABLE 0x<分隔符>

下列数据有恒定 (**CONST**) 或可变 (**VARIABLE**) 数据长度。带有可变长度的数据以 **0x<分隔符>** 结束。

示例:

DATALEN=VARIABLE 0x0A ' 可变数据长度, 分隔符 LF

DEFINE_KEYWORD

DEFINE_KEYWORD=<关键字> <值><关键字> := 任意用户关键字用于在代码载体上标记一个新的数据段

<值> := "<字符串>" 或 0x<十六进制值>

定义关键字 **<关键字>** 通过值 **<值>**

示例:

DEFINE_KEYWORD=DATA_OEM "OEM"

DEFINE_KEYWORD=DATA_SIN840D 0x840D

<关键字>

通过 **DEFINE_KEYWORD** 定义的关键字, 该关键字在代码载体上称为一个新的数据段。代码载体说明文件中 **<关键字>** 上的下列项必须通过 **DEFINE_KEYWORD** 获取定义的值 **<值>**。

项

(B)Item<n>=<line>

<n> := 代码载体数据运行号码, 从 1 开始依序升级

<line> := <(最大) 长度, 以字节为单位> <代码载体数据格式> <对话框变量>

<代码载体数据格式>:

<对话框变量>: 分配代码载体数据到对话框数据

如果直接在项 **<n>** 前定义一个用户关键字 **keyword**, 则 **<对话框变量>** 值为 **<关键字>**

代码载体数据转换规定 **<n>**

示例:

Item1	32	ASCII	T3	'	转换刀具名称
				'	刀具对话框数据 3

Bltem

Bltem<n>=<line>

<n> := 在块<i>内代码载体数据运行号码，从 1 开始依序升级

<line> := 类似于项<n>

一个块中代码载体数据 <n> 转换规定。如果代码载体数据分配了一个刀具对话框数据 T<n>，则对话框数据分配代码载体数据的第一个值到块中。

示例：

```
Bltem1  1    BCD C1, T2  ' 子类型转换刀沿
          ' 对话框数据 1 和刀具对话框数据 2
          ' (1. 块值相关于 T2)
```

块和 End_Block (结束_块)

块<n> <重复规定>

<n> := 块运行号码，从 1 开始依序升级

<重复规定> := * 项<n> | CONTIGUOUS Bltem1

依照一个数据块 **Bltem<n>** (至关键字 **End_Block<n>**)，相应 <重复规定> 存放在代码载体上。

说明

对于块<n> *项<n>，项<n> 必须定义在块<n> 前。

End_Block: 一个带 块<n> 定义的数据块的结束标记

示例：

```
块 1 * 项 6          ' 块 1 相当于项 6 的值
                    ' 重复
块 1 CONTIGUOUS Bltem1 ' 经常读取块 1，直至计数变量
                    ' Bltem1 不提供以 1 为单位升高值。
                    ' 经常写入块 1，值
                    ' 相当于 Bltem1 分配的对话框变量
                    ' 。
```

End_Block<n>

6.7.3 数据的转换规定

代码载体的数据格式

可支持下列代码载体数据格式：

- 数据格式：注释
- ASCII：ASCII 字符集
- INT：16 位整数（Intel 格式）
取值范围：-32768 <= INT <= +32767
- BCD：二进制编码的小数（可能带符号和小数点）
不相关的十进制以左对齐的方式预置为 0。
（比较：<代码载体数据格式> 用于 Item / BItem）

在代码载体和对话数据之间进行赋值

Item<n> 或 BItem<n> 的转换规定包括如不赋值/赋值给一个/赋值给多个对话框变量，可能带一个转化规定，在本章节中有详细说明。

Item<n> 或 BItem<n> 的一般转换规定是：

(B)Item<n>=<line>

- <n> := 代码载体数据的序列号，
无间隔地以升序排列
- <line> := <（最大）长度以字节为单位><代码载体数据格式><对话框变量>

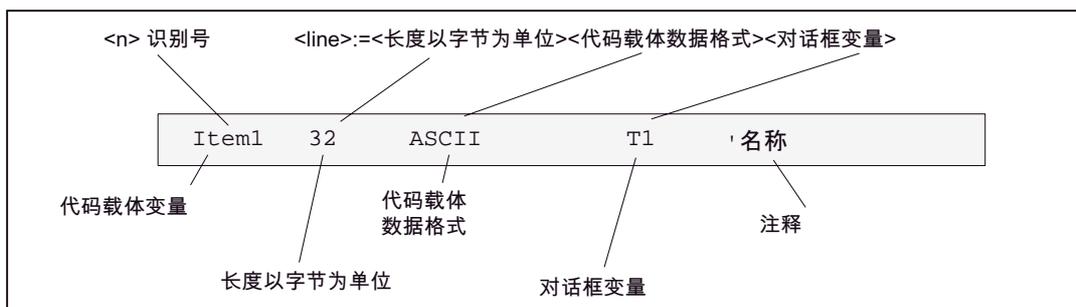


图 6-15 转换规定

对话框变量

<对话框变量>: = <dvar1>[=(<uv>)] [, <dvar2>[=(<uv>)] [, <dvar3>

[&<dvar4>[=(<uv>)] [, <dvarN>[=(<uv>)]]

<dvar> := T<index> | C<index> | -

T = 刀具数据 (tool data) ,

C = 刀沿数据 (cutting edge data) ,

index = 刀具/刀沿对话框数据内部索引

- = 没有赋值给一个对话框变量

<dvar1>&<dvar2>=<uv>: 转化规定适用于 <dvar1> 和 <dvar2>

uv := <arithm. Op1> [<arithm. Op2>] .. [<arithm. OpN>]

arithm. Op := +<const> | -<const> | *<const> | /<const>

示例:

T2=(*10), T3=(/100 + 10)

或

uv := <替换 1> [<替换 2>] .. [<替换 N>]

替换:= <const1> [, <const2>] ..

[, <constN>] ^ <constM>

或

<const1..const2> ^ <const3>

const1 = 下限值,

const2 = 上限值

说明

在转换对话框变量到代码载体变量时, 在写入时, 如果有多个运算域, 则右运算域转换到第一个左运算域。

示例:

T2=(20..29 ^ 120 40,50 ^ 130)

值为 25 的代码载体变量转换到值为 120 的对话框变量 T2 中 (读)。值为 120 的对话框变量 T2 转换到值为 20 的代码载体变量中 (写)。

或

uv := <Tetn>

Tetn := n. 四位二进制, 按字节顺序排列

Byte1, = Tet1 和 Tet2

Byte2, = Tet3 和 Tet4

划分代码载体变量 (BCD 格式) 的四位二进制到对话框变量。

示例:

T5=(Tet1), T6=(Tet2), T7=(Tet3), T8=(Tet4)

如果代码载体变量中包含, 比如值 0x1234, 则对话框变量 T5 取得的值为 1, 对话框变量 T8 取得的值为 4

或

uv := <比较>

比较 := < <const> [INVSIGN] | <= <const> | = <const> | >

<const> >= <const>

根据比较结果将代码载体变量分配到一个对话框变量中。

INVSIGN

在读取时: 对对话框变量的符号进行。

当写入时: 对代码载体变量的符号进行反向。

示例:

C1=(<0 INVSIGN), C2=(>=0)

- 读取:

代码载体变量的一个负值与对话框变量 C1 相一致, 正值与对话框变量 C2 相一直; 对话框变量 C1 被转换为一个正值。

- 写入:

对话框变量 C1 乘以 (-1)。如果值小于 0, 则代码载体变量由 C1 取得值, 否则由 C2 取得值。

说明

转换规定仅适用于分析一个数据类型为“整数”的对话框变量。

6.7.4 示例：说明文件

说明文件或转换文件举例

文件名必须输入在 ...user\mmc.ini bei WToolldSysKonv = **wkonvert.txt** 下。

文件名，例如 **wkonvert.txt**：

代码载体变量	长度 (字节)	数据格式	对话框变量	注释
Item1	32	ASCII	T1	' 名称
Item2	3	BCD	T2	' Duplo
Item3	2	BCD	T4=(Tet1),T5=(Tet2),T6=(Tet3), T7=(Tet4)	
'刀具大小： 左、右、上、下				
Item4	32	ASCII	T8	'刀位类型
Item5	1	BCD	T9	'状态
Item6	1	BCD	T3	'数量刀沿
Item7	1	BCD	T10	'刀具监控类型
Item8	1	BCD	T11	'刀具搜索类型
'用户刀具数据				
Item9	4	BCD	A1	'刀具 OEM1
Item10	4	BCD	A2	'刀具 OEM2
'刀沿数据				
Block1 * Item6				
Bitem1	2	BCD	C1	'子类型, 类型
Bitem2	1	BCD	C4	'刀沿位置
'刀具长度补偿				
Bitem3	4	BCD	C5	' 长度 1
BItem4	4	BCD	C6	' 长度 2
BItem5	4	BCD	C7	' 长度 3
'刀具半径补偿				
BItem6	4	BCD	C8	'长度 1
BItem7	4	BCD	C9	'长度 2
BItem8	4	BCD	C10	'半径 1
BItem9	4	BCD	C11	'半径 2
BItem10	4	BCD	C12	'角 1
BItem11	4	BCD	C13	'角 2
'磨损长度补偿				
BItem12	4	BCD	C14	'长度 1
BItem13	4	BCD	C15	'长度 2
BItem14	4	BCD	C16	'长度 3
'磨损半径补偿				
BItem15	4	BCD	C17	'长度 1
BItem16	4	BCD	C18	'长度 2

代码载体变量	长度 (字节)	数据格式	对话框变量	注释
Bltem17	4	BCD	C19	'半径 1
Bltem18	4	BCD	C20	'半径 2
Bltem19	4	BCD	C21	'角 1
Bltem20	4	BCD	C22	'角 2
'基本尺寸长度补偿				
Bltem21	4	BCD	C23	'基本长度 1
Bltem22	4	BCD	C24	'基本长度 2
Bltem23	4	BCD	C25	'基本长度 3
Bltem24	4	BCD	C26	'自由切削角
Bltem25	1	BCD	C27	'高架插入件
Bltem26	2	BCD	C29	'使用寿命, 单位: 分钟
Bltem27	2	BCD	C30	'使用寿命预警极限
Bltem28	2	BCD	C31	'工件数
Bltem29	2	BCD	C32	'使用寿命预警极限
'用户刀沿数据				
Bltem30	4	BCD	U1	'刀沿 OEM1
Bltem31	4	BCD	U2	'刀沿 OEM2
'用户监控数据				
Bltem32	4	BCD	S1	'监控 OEM1
Bltem10	4	BCD	S2	'监控 OEM2

6.7.5 示例：数据 — 字符串

建立一个数据字符串

转换文件：wkonvert.txt

代码载体变量	长度以字节为单位	格式	对话框变量
Item1	10	ASCII	T1 名称, \$TC_TP2
Item2	2	BCD	T2 Duplo, \$TC_TP1
Item3	2	BCD	T4 = (Tet1), T5 = (Tet2), T6 = (Tet3), T7 = (Tet4) 刀具大小: 左、右、上、下 \$TC_TP3、4、5、6
Item4	10	ASCII	T8 刀位类型, \$TC_TP7 的文本
Item5	1	BCD	T3 数目 刀沿, \$P_TOOLND[tnr], tnr = 刀具编号
Item6	4	BCD	A1 刀具 OEM1, \$TC_TPC1
Item7	4	BCD	A2 刀具 OEM2, \$TC_TPC2
Item8	2	BCD	C1 子类型, 类型, \$TC_DP1
Item9	4	BCD	C5 几何长度 1, \$TC_DP3
Item10	4	BCD	C10 几何长度 1, \$TC_DP6
Item11	4	BCD	C14 磨损长度 1, \$TC_DP12

用该转换文件可以读入下列数据字符串或者在写入时进行读入字符串—数据，生成：

626F687265725F312020000111116E6F726D616C2020202001D00010E3D000000502
05B00002E3B0000003B000E4562F2F

如果用 SINTDC 处理，则以 SINTDC 和 HMI 高级之间的接口的字符串为依据。

含义

将字符串划分为单个的值:

字符串	值	输入 wkonvert.txt	对话框变量的数据类型
626F687265725F312020	“钻头_1”	10 ASCII T1	T1 字符串名称, \$TC_TP2
0001	1	2 BCD T2	T2 整数 Duplo, \$TC_TP1
1111	1,1,1,1	2 BCD T4=(Tet1), T5=(Tet2), T6=(Tet3), T7=(Tet4))	T4 整数刀具大小: 左 T5 整数刀具大小: 右 T6 整数刀具尺寸: 上 T7 整数刀具尺寸: 下
6E6F726D616C20202020	“标准”	10 ASCII T8	T8 字符串刀位类型, \$TC_TP7 的文本
01	1	1 BCD T3	T3 整数数目 刀沿, \$P_TOOLND[tnr] tnr=刀具编号
D00010E3	-10.300	4 BCD A1	A1 双刀具 EM1, TC_TPC1
D0000005	-5	4 BCD A2	A2 双刀具 EM2, TC_TPC2
0205	205	2 BCD C1	C1 整数子类型, \$TC_DP1
B00002E3	2.3	4 BCD C5	C5 双几何长度 1, \$TC_DP
B0000003	3	4 BCD C10	C10 双几何半径 1, \$TC_DP6
B000E456	0.456	4 BCD C14	C14 双磨损长度 1, \$TC_DP12
2F2F	结束标记 (根据 TIS_EOT=0x2F2F, mmc.ini)		

A

INI 文件表列

注意:

在本章节中所有的 INI 文件按段的字母顺序进行排列，只允许机床制造商或最终用户修改 INI 文件。**不得更改所有其它的段!**

参见

AEDITOR.INI (页 293)
DINO.INI (页 295)
DG.INI (页 297)
DGOVW.INI (页 297)
DH.INI (页 299)
DPDH.INI (页 299)
DPSIM.INI (页 299)
HEADER.INI (页 301)
HMIDESK.INI (页 305)
IB.INI (页 306)
IF.INI (页 308)
KEYS.INI (页 309)
LOGDRIVE.INI (页 309)
MASCHINE.INI (页 312)
MBDDE.INI (页 315)
MMC.INI (页 321)
NETNAMES.INI (页 327)
OEMFRAME.INI (页 331)
PARAM.INI (页 331)
PARAMTM.INI (页 332)
REGIE.INI (页 333)
SEEDITOR.INI (页 335)
SIMTOGEO.INI (页 343)
TASKCONF.INI (页 346)
ACTLOG.INI (页 292)

A.2 AEDITOR.INI

概览

有关 ASCII 编辑器的信息，见文件 AEDITOR.INI。
在段[MEMORY]中记录有表示可编辑文件尺寸的数据。

[MEMORY]

MemFactor=f ; f 表示对于单位 à 8 MB 允许为 $1 \leq f \leq 31$
; 这等于 8 MB 至 248 MB

段[ACCESSLEVEL]中的设置可以影响编辑器的存取权限：

[ACCESSLEVEL]

GlobalReplace=6
SettingHiddenCheck=7
Enable_Simulation = 7
; 可以在编辑器中隐藏软键模拟。

[MMC_Wizard]

AEDITOR=AEDITOR.COM

[MEMORY]

; default factor for maximum size of file
; valid values are: from 1 to 31
; factor x ($1 \leq x \leq 31$) means: you can edit a file up to a
; maximum size of $x * 8$ MB
; for example: factor 1 = maximum size of 8 MB, 31 = 248 MB
; attention: a greater factor needs more RAM
; the application can override this default factor
MemFactor=2
; default factor for maximum size of buffer-file for editing NC-domains
; see explanation for MemFactor above
; the factor should be great enough for the whole NCU-memory
; the application can't override this default factor
MemNCFactor=2

;步（输入行！）的数目，这些可以用 ASCII 编辑器
;复位 (Undo)/恢复 (Redo)。
;默认=0： 没有 Undo/Redo 步可以激活
;> 0，最大 32767 仅在将 UndoHandling 设置为不等于 2 的情况下才有效。
NrOfUndoSteps=10
; Undo/Redo 处理模式（与应用程序共同生效）：
; UndoHandling=0 自动： 应用程序立即对 Ctrl+Z、CTRL+Y 做出响应，默认
; UndoHandling=1 通知有效的应用程序，Ctrl+Z、CTRL+Y 变为应用程序专用动作
; UndoHandling=2 无复位/恢复功能
UndoHandling=0
;也对 NC 文件有效
; EnableUndoOnNc=1 也可以用 Undo/Redo 处理 NC 文件
为了提供 NC 文件用于 HMI 中的处理操作，该模式也延长了所需时间。
; EnableUndoOnNc=0 不用 Undo/Redo 同时处理 NC 文件（默认）
EnableUndoOnNc=0

[STEPS]
; maximum number of blocks to search for a step, 0 == no limit
MaxNrOfBlocks=1500

[ACCESSLEVEL]
GlobalReplace=6
SettingHiddenCheck=7
Enable_Simulation=7 ; 编辑器中用于软键“模拟”的保护等级

A.3 DINO.INI

访问权限

在文件 DINO.INI 中规定批量调试的访问权限，并确定文档的其它设置：

[ACCESSLEVEL]

ANWENDER=3；批量调试的读入权限

；用户密码（标准）

PROGRAMMIERER=4；批量开机调试的文档创建权

；相应的钥匙开关 3（标准）

[Function]

；0: 不支持 ISO 格式的纸带

；1: 支持 ISO 格式的纸带

IsoActive=0

；1: 覆盖 CFG_RESET.INI 之前询问，0: 不询问

Ask_for_CFG_RESET.INI=0

；用于从 ISO 纸带文档存储

；ISO 格式程序的目录

；不可通过用户界面更改（在 IsoActive=1 的情况下支持）

ISO-Path=/_N_WKS_DIR/_N_SHOPMILL_WPD

；写入 ISO 纸带文档：

；1: 仅在 ISO 目录中接受 ISO 程序，

；0: 接受到所有目录中

；不可通过用户界面更改（在 IsoActive=1 的情况下支持）

AcceptOnlyISO-Path=1

；1: 先解压然后在读取时删除 ZIP 文件

；在开机调试文档上，

；0: 不解压，但在数据库中存储 ZIP 文件

；不可通过用户界面更改

UnPack=1

; 在 PLC 内存服务后等待的时间，以秒为单位

Wait for PLC=20

; 读入的内部 pc 格式的存档不是设置文档:

; 0: 解压文件的目标文件一般都通过

; 有相同名称的现有文件进行确定，如无此类文件存在则目标文件为

; is MMC

; else: destination is determined from the file header,

; i.e. from the location where the file was read from

TakeDestinationFromHeader=0

[系统]

; 通过设置非零值激活 V24 设置

V24 设置=0

A.4 DG.INI

概览

在“诊断”操作区中可以通过输入保护级分别设置单个软键的存取权限。
在文件 DG.INI 中的[ACCESSLEVEL] 下输入所需的保护级：

```
[ACCESSLEVEL]
ALARMS=7           ; 报警
SERVICE=7        ; 服务显示
COMMUNIC-LOG=3    ; 通信记录
PLC=3             ; PLC 状态
PLC-CHANGE=2      ; PLC 状态: SK "Change Value" (修改)
                  ; SK "Reset Value" (修改复位)
                  ; SK "Commit Value" (接受修改)
                  ; 通过制造商密码进行锁定。
REMOTE-DIAGNOSIS=4 ; 远程诊断
FILE=7            ; 文件功能
RUN-LOG=7         ; 使用户密码锁定软键“运行记录器”
```

A.5 DGOVW.INI

概览

文件用于诊断 → 服务显示中用户特定的附加信号的设计，画面“服务一览”。设计信号添加在本画面标准信号的下面，可能要通过移动滚动条才可显示。文件必须明确建立在 mmc2 或者 addon、oem、user 其中的一个目录之中。

```
[GLOBAL]
; 附加信号的数目
NrOfSignals=
; 待显示的自身图标的位置
```

```
[BMP]
100="<Bitmapname>"
```

```
101="..."  
.  
.  
; 每个信号一个段
```

```
[SIG1]
```

```
text=
```

```
Item=
```

```
expr=
```

```
.
```

```
.
```

```
[SIG2]
```

```
.
```

```
.
```

```
[SIG<n>]
```

```
.
```

```
.
```

DGOVWTEXT_xx.INI

文件名中的 **xx** 代表与语言相关的文件的各个语言缩写。文件包含与语言相关的文本，用于服务概览中用户特定的信号。必须在目录 `mmc2`、`user`、`addon`、`oem` 或者 `..language` 中明确建立该文件。

```
[TEXT]
```

```
;$T<Nr> =“与语言相关的文本”
```

```
;<Nr> 表示 1000 – 32767 的值，已预留 1000 以下的值。
```

```
; 举例：
```

```
$T1001=“信号 s”
```

设计的详细资料和举例可以参见：

参见

增加用户专用的服务显示 (页 35)

A.6 DH.INI

概览

在文件 DH.INI 确定了用于数据管理目录的存取权限。

```
[ACCESSMASKS]  
\wks.dir\*.wpd\*.mpf = 75775  
\mpf.dir\*.mpf = 75775  
\cus.dir\*.spf = 33773
```

A.7 DPDH.INI

概览

在文件 DPDH.INI 中设定了用于工件程序处理的存取权限。

```
[ACCESSLEVEL]  
PROGRAMMIERER=4  
BEDIENER=5  
ENABLE_DIALPROG=0  
SetStandardWpSave=2  
ENABLE_SIMULATION=7
```

A.8 DPSIM.INI

概览

在文件 DPSIM.INI 中确定了用于模拟的流程范围。

```
[ACCESSLEVEL]  
;minimum level for changing setup standard:  
SETUP=1  
USER=3  
BEDIENER=5  
  
[MAIN]
```

; <>0: simulate program extern modal (see also
; EXTERN_SIZE, EXTERN_CHECK)
EXTERN_MODAL=0
; simulate program extern modal, if program
; size >= EXTERN_SIZE
; only if EXTERN_MODAL<>0
EXTERN_SIZE=150000
; only if EXTERN_MODAL<>0
EXTERN_CHECK=1
; <>0: reset tool data, if new workpiece is selected
; =0: use current tool data, if new workpiece is selected
SIM_LOAD_TOOLS=1

[PRELOAD]

;=1: TO_INI.INI 和 TO_ADDON.INI 通过 DPNC SIM.EXE 预加载
;=0: TO_INI.INI 和 TO_ADDON.INI 通过模拟预加载
TOOLS=0
;= 1: all cycles are preloaded by DPNC SIM.EXE
; if section MAIN, entry CYCLE is set, additionally all cycles from shadow directory are
preloaded by DPNC SIM.EXE
;= 0: cycles are loaded by simulation, if cycle is called
CYCLES=1
;=2: load for simulation to \CUS.DIR, \CMA.DIR or \CST.DIR (default)
; load only cycles from NCK
; only evaluated if [MAIN]CYCLE is set and [PRELOAD]CYCLES=1
;=0: load all cycle (from NCK and MMC) for simulation to
; \CUS.DIR, \CMA.DIR or \CST.DIR
; only evaluated if [MAIN]CYCLE is set and [PRELOAD]CYCLES=1
;=1: use the cycle interface unter \DP.DIR\SIM.DIR
CYCLEINTERFACE=2

A.9 HEADER.INI

概览

在文件 HEADER.INI 设定了用户专用的图标与文本，并在全局通道交叉状态显示（标题）中输出。

[CondStopIcons]

DefaultIcon=CondWait.bmp ;0 = 没有保持状态

1=CondStop.bmp	; 没有 NC 准备
2=CondStop.bmp	; 没有 BAG 准备
3=CondStop.bmp	; 紧急关闭激活
4=CondStop.bmp	; 带有停止的报警激活
5=CondStop.bmp	; M0/M1 激活
6=CondStop.bmp	; 单程序段中的程序段已结束
7=CondStop.bmp	; NC 停止激活
8=CondStop.bmp	; 无读入许可
9=CondStop.bmp	; 无进给许可
10=CondWait.bmp	; 暂停时间激活
11=CondWait.bmp	; 无帮助功能确认
12=CondStop.bmp	; 无轴许可
13=CondWait.bmp	; 没有达到准确停止
14=CondWait.bmp	; 等待定位轴
15=CondWait.bmp	; 等待主轴
16=CondWait.bmp	; 等待其它通道
17=CondStop.bmp	; 等待进给倍率
18=CondStop.bmp	; NC 程序段有错
19=CondStop.bmp	; 等待外部 NC 程序段
20=CondWait.bmp	; 等待同步动作
21=CondWait.bmp	; 程序段运行过程激活
22=CondStop.bmp	; 无主轴许可
23=CondStop.bmp	; 轴进给倍率 0
24=CondWait.bmp	; 等待刀具更换确认
25=CondWait.bmp	; 齿轮换档
26=CondWait.bmp	; 等待位置调节
27=CondWait.bmp	; 等待螺纹切削
28=CondWait.bmp	; 备用
29=CondWait.bmp	; 等待冲孔

30=CondWait.bmp	;等待安全运行	
31=CondWait.bmp	;无通道准备	
32=CondStop.bmp	;摆动激活	
33=CondWait.bmp	;轴交换激活（阻止程序段切换，因为会对轴交换形成冲击）	
34=CondWait.bmp	;轴容器旋转	
35=CondWait.bmp	;AXCT: 随动轴激活	
36=CondWait.bmp	;AXCT: 引导轴激活	
37=CondWait.bmp	;AXCT: 跟踪激活	
38=CondWait.bmp	;AXCT: 内部状态切换	
39=CondWait.bmp	;AXCT: 进给轴/主轴禁用	
40=CondWait.bmp	;AXCT: 补偿运动激活	
	;轴容器更换: 已叠加的运动	
41=CondWait.bmp	;AXCT: 轴更换已激活	
42=CondWait.bmp	; AXCT 轴插补器激活	
43=CondWait.bmp	; 等待编译循环	
44=CondWait.bmp	;访问系统变量	
45=CondStop.bmp	;查找对象已找到（;程序段查找 过程已找到查找对象且 NCK 已停止）	
46=CondWait.bmp	;快速后退已触发	
47=CondWait.bmp	;AXCT: 等待主轴停止	
	;轴容器更换	
48=CondWait.bmp	;机床数据匹配（;新配置）	
49=CondWait.bmp	;轴更换: 轴已耦合	
50=CondWait.bmp	;轴更换: 快速提升激活	
51=CondWait.bmp	;轴更换: 新配置激活	
52=CondWait.bmp	;轴更换: AXCTSW 激活	
53=CondWait.bmp	;轴更换: Waitp 激活	
54=CondWait.bmp	;轴在另一个通道中	
55=CondWait.bmp	;轴更换: 轴是 PLC 轴	
56=CondWait.bmp	;轴更换: 轴是摆动轴	
57=CondWait.bmp	;轴更换: 轴是 Jog 轴	
58=CondWait.bmp	;轴更换: 指令轴	
59=CondWait.bmp	;轴更换: 轴是 OEM 轴	
60=CondWait.bmp	;已耦合的随动轴	
61=CondWait.bmp	;拖行的随动轴	
62=CondWait.bmp	;已耦合的从动轴	
63=CondWait.bmp	;在 M0 循环结束处停止	新 V07.10
64=CondWait.bmp	;在 M1 循环结束处停止	新 V07.10
65=CondWait.bmp	;等待: 轴在运行档块处停止	ERLde41458
66=CondWait.bmp	;“主从功能转换有效”	07.10.14.xx
67=CondWait.bmp	;“轴更换: 轴 单轴”	07.10.14.xx

68=CondWait.bmp	;"???"	07.10.14.xx
69=CondWait.bmp	;"同步旋转：同步主轴"	07.10.14.xx
70=CondWait.bmp	;"关闭位置同步主轴"	07.10.14.xx
71=CondWait.bmp	;"等待转换轴使能"	07.10.14.xx
72=CondWait.bmp	;"???"	07.10.14.xx
73=CondWait.bmp	;"???"	07.10.14.xx

[Pos1]

ChanStatFeedStopSpindleStop = 1 ; 显示通道 1 的通道状态，并
Spindles=1,2 ; 由通道 1 的进给禁用和
; 主轴 1 和 2 的主轴禁用叠加

[Pos3]

SpindStat=1 ; 主轴 1 的主轴状态
; ChanStatFeedStop = 3 ; 显示通道 3 的通道状态，并
; 由通道 3 的进给禁用叠加

[Pos5]

ChanStat = 2 ; 显示通道 2 的通道状态
; SpindStat= 3 ; 显示主轴 3 的主轴状态

[Pos9]

ChanStatFeedStopSpindleStop = 1 ; 显示通道 1 的通道状态，并
Spindles=1 ; 由通道 1 的进给禁用和
; 主轴 1 的主轴禁用叠加

[Pos11]

ChanStatFeedStop = 2 ; 显示通道 2 的通道状态，并
; 由通道 2 的进给禁用叠加

[Pos13]

SpindStat=2 ; 主轴 2 的主轴状态

[Pos15]

ChanStatFeedStopSpindleStop = 1 ; 显示通道 1 的通道状态，并
Spindles=1 ; 由通道 1 的进给禁用和
; 主轴 1 的主轴禁用叠加

[UserIcons]

;UI_0 = <name.bmp> , <Position> ; 图标的名称、位置

...

;UI_31 = <name.bmp> , <Position> ; 图标的名称、位置

;USER_ICON_BASE = DBxx.DBByy

; 数据模块的编号, 使用

; 双字, 用于用户图标

; yy 双字的初始位

[UserTextInfo]

;USER_TEXT_BASE = DBxx.DBByy

; xx 数据模块, 带用于

; 标题中用户信息的接口

; 接口的 yy 开始字节

; TextLength=33 ; 激活文本限制, 如果需要

; 激活文本限制。

[OemIcons]

;Oem_ICON_BASE=DBxx.DBByy

;OI_0= <name.bmp> , <Position> ; 图标的名称、位置

...

;OI_31 = <name.bmp> , <Position> ; 图标的名称、位置

[OEMTextInfo]

;OEM_NUMBER_TEXTFIELD=3 ; 每行最多 3 个文本栏, 即:

; 两行最多一共可以有 6 个文本栏。; 输出长度限制为

; 每行 38 个字符。

;OEM_TEXT_BASE=DBxx.DBByy

A.10 HMIDESK.INI

概览

仅在将 HMI 高级安装到 PG /标准 PC 上时才需要设置该文件。

[Global]

; 全局配置数据
; 桌面定义 (最少 2 个, 最多 4 个)
NumberOfDesktops = 2
; 用于鼠标操作的控制模式: 任务栏
ControlMode = Icon
eyboardControl = False

[Desktop1]

[Desktop1]

;带任务条的激活桌面
Explorer = True
;
;建立桌面后自动启动应用程序
; 启动应用程序
StartupApplication = "rngkrnl.exe"
;
;带自身颜色表的桌面
SetHMIColors = True
;
;确定起始时间
CreateDesktop = Immediately
;
;用户界面在前景中以窗口模式保持
WindowOnTop = False

[Desktop3]

[Desktop4].

A.11 IB.INI

概览

在“开机调试”操作区中的软键存取权限可以通过输入保护级分别设立。在段[ACCESSLEVEL]下输入所需的保护级。在段[LANGUAGE]段中确定操作语言。

[ACCESSLEVEL]

RESET =2 ;NCK-Reset NCK 复位

CHG_SCL_SYS=5 ;change scaling system 尺寸系统英制/公制

MASCHDAT =4 ; Machine data 机床数据

DRIVE_CFG=2 ;drive configuration 驱动配置

BOOTFILE=2 ;Bootfile 引导启动文件

MOT_CTRL=2 ;Motor/Valve selection 选择电机/阀门

VIEW=2 ;Userviews + Displayoptions

用户视图与
显示选项

NC_ADDR=2 ;NCK Address NCK 地址

SETUP_SWITCH=2;start-up switch 开机调试开关

PLC_STATE=3 ;plc state PLC 状态

PLC_DATE=3 ;plc date PLC 日期

PLC-CHANGE=2 ; PLC 状态:

;Sk-Change Value (修改)

;Sk-Reset Value (修改复位)

;Sk-Commit Value (接受修改)

由制造商锁止

DRIVE_SERVO=2 ;Drives/Servo 驱动伺服

SET_COL=3 ;colours 颜色

SYST_SETTINGS=4 ;system settings 系统设置

SET_LANG=2 ;Language 语言

SET_BTSS=2 ;Operator panel 操作面板

OPTION=2 ;Options 选项

EDIT=1 ;Editor 编辑器

DOS=1 ;DOS shell DOS-Shell 程序

TOOL_MNG=3 tool management 刀具管理

FILE=7 ;File functions 文件功能

PASSWD=3 ;Set/Delete password 修改和删除密码

CHG_PASSWD=7 ;SK "password ..." SK“密码...”

CHANGE_LANG=7 ;change Language 切换语言

[LANGUAGE]

;Softkey select language instead of softkey change language in the main
;menu of area setup.

;False: Softkey change language to toggle between the foreground
; and background language.

;True: Softkey select language instead of softkey change language
; to select one of the installed languages. To use this add the
; following values to the entries of the LANGUAGE section in the
; mmc.ini:

;LanguageList= ... , GR

;FontList= ... , Europe

;FontListKO= ... , Europe2

;LList= ... , 德语

SkSelectLngInsteadSkChangeLng=False

说明

如果要显示带有扩展名 *.hlp 的旧的、OEM 专用的帮助文件，在 [HELP_EXTENSION] 段中的参数 HelpFileExtension 必须设置为 0（WinHelp）。

A.12 IF.INI

概览

在文件 IF.INI 中提供有用于辅助编程以及用于输出与语言相关文本的功能:

[TECHNOLOGY]

; "m" = 铣削
; "t" = 车削
; "g" = 磨削
; "s" = 步冲
; "p" = 冲裁
; "a" = 显示所有工艺

; 段中的符号, 例如

Technology=m

[TEXTFILES]

; Siemens = Dh\Cst.dir\lfs_gr.com
; SiemensI= Dh\Cst.dir\lflT_gr.com 车削 ISO 或者
; SiemensI= Dh\Cst.dir\lflM_gr.com 铣削 ISO
; 机床制造商 = Dh\Cst.dir\lflM_gr.com
; 机床制造商 = Dh\Cst.dir\lfl_gr.com
; 最终用户 = Dh\Cst.dir\USER_gr.com

; 段中的符号, 例如

Siemens=Cst.dir\lfs_gr.com

SiemensI=Cst.dir\lflit_gr.com

Manufacturer= ...

最终用户=

[CONTEXT]

; = 0 上下文相关性未激活

; = 1 上下文相关性已激活

; 段中的符号, 例如

Extended_Sensitivity=1

[TEXTSEARCH]

; = 1 仅按指令文本进行文本查找

; = 2 仅按说明文本进行文本查找
; = 3 按照指令文本和说明文本进行查找
; 段中的符号, 例如
Searchkind=3

A.13 KEYS.INI

概览

该文件中的条目可以使用操作面板 OP 010 和 OP 010C 的硬键或者使用带有软键块和 PLC 软键的 SINUMERIK 键盘。

文献: “补充操作界面”开机调试手册,
章节“设计 OP 热键和 PLC 键”

A.14 LOGDRIVE.INI

概览

在文件 LOGDRIVE.INI 已连接了逻辑驱动器并给定了相应的访问权限。

[PCU]

; device name of PCU for USB local

UsbFront=G:

; device name of PCU

UsbX203=H:

UsbX204=I:

; - <SEARCH PATH> : USER, OEM, ADD_ON, HMI_ADV, MMC2

; - it can be set up to 8 net devices or usb devices

; - for each device access rights can set for the applications program, machine or services

; missing access rights are set to 7 (no limit) by default

; - all entries for net device are ignored if the software option (Network drive management) is not set

; - all usb devices are ignored, if no tcu configuration exist.

ACTTCU:\FRONT always can be set

; all files LOGDRIVE.INI of the <SEARCH PATH> are read and the contents is merged

; display machine data are append after the last entry of LOGDRIVE.INI,

; but the maximum is limited to 8 net devices at all

; \$MM_CMM_DIRECTORY_SOFTKEY_PATH1 (MD 9676) to

```

$MM_CMM_DIRECTORY_SOFTKEY_PATH4 (MD 9679)
; the corresponding access level can be set by
; USER_CLASS_DIRECTORY1_P (MD 9510) to USER_CLASS_DIRECTORY4_P (MD 9513)
for program and services
; USER_CLASS_DIRECTORY1_M (MD 9516) to USER_CLASS_DIRECTORY4_M (MD
9519) for machine
;
; 如果在 USER、OEM 或者 ADD_ON 中存在 LOGDRIVE.INI 文件，忽略所有显示机床数据。
; example for net device
;Connection<index>=\\server\share
; example for local device
;Connection<index>= X:
; example for usb device of current pcu/tcu, front
; one partition number may be specified for ALL tcu's. if omitted,
; the default partition is 1
: the partition number has no effect to a usb devices of the pcu
;Connection<index>= ACTTCU:\\FRONT,1
; example for usb device of tcu TCU1, port X203. for second port use X204
; a partition number may be specified. if omitted,
; the default partition is 1
: the partition number has no effect to a usb devices of the pcu
;Connection<index>=TCU:\\TCU1:X203,1
; any directory path can be append (TCU:\\TCU1:X204,2\Directory1),
; but it must exist on the device

```

[Connections]

```

; Connection1 to Connection<N> are read with the corresponding access
; levels, empty entries are ignored
; N=0 to 8, if missing ConnectionNum is set to 8
; set ConnectionNum to 0 to ignore all entries of LOGDRIVE.INI
; and display machine data
;ConnectionNum=8
; usb device of current pcu/tcu, front
Connection1=ACTTCU:\\FRONT,1
AccessProgram1=7
AccessMachine1=7
AccessServices1=7

```

示例: logdrive.ini 中条目

```
[Connections]

Connection1=ACTTCU:\\FRONT
SK_Connection1=FRONT_USB
AccessMachine1=7
AccessProgram1=7
AccessServices1=7

.
.
.

Connection5=TCU:\\station2:X203
SK_Connection5=标签 5
AccessMachine5=6
AccessProgram5=6
AccessServices5=6

Connection6=TCU:\\station3:X204
SK_Connection6=标签 6
AccessMachine6=6
AccessProgram6=6
AccessServices6=6

Connection7=smb:\\<计算机名>\\<目录>
SK_Connection7=网络驱动器
Username7=Mustermann
Password7=431S0Y2E3K530I28
AccessMachine7=6
AccessProgram7=6
AccessServices7=6

Connection8=\\<计算机名>\\tmp
SK_Connection8=驱动器%n 本地
AccessMachine8=5
AccessProgram8=5
AccessServices8=5
```

用户名: (仅对于 NW Windows)

Username<x> =

在屏幕上输入用户名，比如：Mustermann。

密码：（仅用于 NW Windows）

Password<x> =

密码会显示在屏幕上，并加密输入到文件 logdrive.ini 中。

说明

也可以通过 HMI 高级操作界面在“开机调试” → “HMI” → “逻辑驱动器”中设置逻辑驱动器：

A.15 MASCHINE.INI

概览

在文件 MASCHINE.INI 中，可以在段[ACCESSLEVEL]中设置运行方式的存取权限。

[ACCESSLEVEL]

;

; 7 = Keyposition 0 everyone in front of the machine,

; Suggestion: Permission only tfor NC-start, no selection of part-program

; 6 = Keyposition 1 operator of the machine, the man at the shop-floor

; of production

; Suggestion: Permission to select Part-Program, no editing

; 5 = Keyposition 2 machine setter (more experienced),

; the one to setup tools and test and run in new Part-Programs

; Suggestion: Permission to modify (edit) Part-Programs

; 4 = Keyposition 3 not yet used, same permissions as 5

; 1 = only manufacturer of machine, System specialist

; 0 = only manufacturer of Numeric Control (Siemens Experts)

; at the state of delevry al users got (nearly) all permissions for work

JOG_PRESET=7 ; Preset or Set actual value at mode JOG mode ==>5 Suggestion machine setter

JOG_HAND=7 ; Handweel at mode JOG

==>6 Suggestion operator

JOG_ANKRATZ=7 ; Scratch

==>6 对操作员的建议

JOG_INCREMENT=7 ; Set increment at mode JOG
==>6 Suggestion operator

JOG_SYNAKT=7 ; Synchron actions at mode JOG
==>5 Suggestion machine setter

JOG_V_ISTW=7 ; watch windows of Actual-values,
==>7 Suggestion everyone
; Spindels, G-Functions, Aux.-Functions,
; Feedrate, Tools or Axis-feedrate at mode JOG

JOG_V_ZOOM=7 ; Zoom actual values at mode JOG
==>7 Suggestion everyone

JOG_INCHMET=7 ; Switch inch/Metric at mode JOG
==>6 Suggestion operator

REF_SISITEC=4 ; Commit safety integrated Ref.Point
==>4 钥匙 3, 以保持兼容

MDI_PROGBE=7 ; Program control at MDI mode
==>6 Suggestion operator

MDI_HAND=7 ; Handweel at MDI mode
==>6 Suggestion operator

MDI_EDITOR=7 ; Edit program at MDI mode
==>5 Suggestion machine setter

MDI_ED_DAT=7 ; Editor funtions at MDI mode
==>5 Suggestion machine setter

MDI_ED_CLR=7 ; Clear buffer at MDI mode
==>5 Suggestion machine setter

MDI_V_ISTW=7 ; watch windows of Actual-values,
==>7 Suggestion everyone
; Spindels, G-Functions, Aux.-Functions,
; Feedrate, Tools or Axis-feedrate at MDI mode

MDI_V_ZOOM=7 ; Zoom actual values at MDI mode
==>7 Suggestion everyone

MDI_SYNAKT=7 ; synchron actions at MDI mode
==>5 Suggestion machine setter

MDI_DATEI=4 ; File funtions at MDI mode

MDI_INCHMET=7 ; Switch inch/Metric at MDI mode
==>6 Suggestion operator

AUTO_OVERST=7 ; Overstore at AUTO mode
==>5 Suggestion machine setter

AUTO_DRF=7 ; DRF at AUTO mode
==>6 Suggestion operator

AUTO_PROGBE=7 ; Program control at AUTO mode
 ==>6 Suggestion operator

AUTO_SASUCH=7 ; Blocksearch at AUTO mode
 ==>6 Suggestion operator

AUTO_HAND=7 ; Handweel at AUTO mode
 ==>6 Suggestion operator

AUTO_KORREK=7 ; Correct program at AUTO mode
 ==>5 Suggestion machine setter

AUTO_V_ISTW=7 ; watch windows of Actual-values,
 ==>7 Suggestion everyone
 ; Spindels, G-Functions, Aux.-Functions,
 ; Feedrate, Tools or Axis-feedrate at AUTO mode

AUTO_V_ZOOM=7 ; Zoom actual values at AUTO mode
 ==>7 Suggestion everyone

AUTO_V_NCPR=7 ; Programm-overview at AUTO mode
 ==>6 Suggestion operator

AUTO_OP_NCPR=7 ; load + select Program at AUTO mode
 ==>6 Suggestion operator

AUTO_ED_NCPR=7 ; Alter enable status of Prog at AUTO mode
 ==>5 Suggestion machine setter

AUTO_SYNAKT=7 ; synchron actions at AUTO mode
 ==>5 Suggestion machine setter

AUTO_INCHMET=7 ; Switch inch/Metric at AUTO mode
 ==>6 Suggestion operator

[Customer]

MDASingleSTEP=yes

ONLY_MKS_ToolBaseDistToGo=no

NcVersionCheck=true
 ; Due to performance issues the functionality offered with respect to
 ; block correction, block view and block search depends on the
 ; Nc-version currently used.
 ; However, to get access to the whole functionality, regardless of the ; Nc-Version in use,
 ; set the flag NcVersionCheck to False.

[BlockSearch]

; Shows (SeruproEnabled=1) or

```
; hides (SeruproEnabled=0)
; the softkey used to do a block search in mode "programtest".
; The softkey appears in both dialogs
; "Blocksearch Searchposition" and "Blocksearch Searchpointer"
SeruproEnabled=1
; When doing a block search in mode "programtest" you can do
; a multi-channel block search. There are 4 channel-configurations which
; define the NC-channels being affected by the block search:
; SeruproConfig =
; 1: selected/active channel
; 2: selected/active channel and all channels having selected the same
; workpiece as the active channel
; 3: selected/active channel and all channels belonging to the active
; chanel's mode group
; 4: all NC-channels
SeruproConfig=1

[SetZero]
; Display_Sign_Switch switch the sign of the toollength in the scratch picture
; Display_Sign_Switch=0 mean - toollength
; Display_Sign_Switch=1 mean + toollength
Display_Sign_Switch=1
```

A.16 MBDDE.INI

概览

在启动时分析评估文件 MBDDE.INI 。 该文件与报警设置和目录 \mmc2 中的当前记录文件有关。

```
[Alarme]
; determines the order in which the alarms are inserted in the alarm list
; possible values:
; FIRST: makes newer alarms stand on top of the list
; LAST: inserts the newest alarm at the bottom
ORDER=LAST
; the alarm/message line is able to scroll the occurred alarms
```

; possible values:
; 0 or negative value: no scrolling
; 500 - 32767: interval in ms
RotationCycle=0
; determines if the alarm server should connect to ALARM_S/SQ or not
; possible values:
; 0 Alarm_S/SQ not handled
; 1 Alarm_S/SQ handled (default)
Alarm_S=1
; if Alarm_S/SQ handling is enabled this key determines if Quits should
; be entered into the alarm-protocol
; possible values:
; 0 Alarm_SQ quits are not entered into the alarm protocol
;(backcompatible)
; 1 Alarm_SQ quits are entered into the alarm protocol (default)
Protocol_Alarm_SQ=1
; If Alarm_S/SQ handling is enabled this key determines if the
; workaround regarding an error in the firmware of the PLC is
; active or not. This workaround corrects the listing of
; Alarm_S/SQ in the diagnosis but causes multiple entries in
; the protocol listing of one alarm event (NEW, NULL, QUIT).
; Using a newer PLC firmware (PLC Besy Stand: S7 PLC_315 2DP
; 2AF03 12.30.03 or higher only necessary at NCU57x.1) this
; workaround can be set inactive to correct protocol writing.
;
; Attention: Setting the workaround inactive while using an PLC
; firmware elder than 12.30.03 leads to unpredictable results!
;
; possible values:
; 0 Workaround active
; 1 Workaround inactive (default)
Alarm_S_FWA_Inactive=1
; path and file names of the alarm text lists
; The filename fragments must not exceed 6 chars, because it is extended
; automatically by the message module according to the selected language.
; The alarm texts in the files specified under UserMMC, UserNCK ...

; overload the corresponding texts of the files defined in MMC, NCK ...
; Therefore the user should modify files according to the needs of his
; system exclusively in UserMMC, UserNCK ...
; Creating alarm text with any DOS-editor may cause Problems with special
; characters like ä, ö or ü. To nevertheless import DOS generated files,
; add a blank and the string "DOS" after the name of the text file.
; Example: UserPLC=F:\dh\mb.dir\myplc_ DOS
; If you add or remove the parameter DOS, you have to touch
(new date of file)
; the corresponding file (*.com) otherwise the change is not accepted.

[Textfiles]

MMC=F:\MMC_52\dh\mb.dir\alm_
NCK=F:\MMC_52\dh\mb.dir\aln_
PLC=F:\MMC_52\dh\mb.dir\alp_
ZYG=F:\MMC_52\dh\mb.dir\alz_
CZYK=F:\MMC_52\dh\mb.dir\alc_
STANDARD_CYCLES=F:\MMC_52\dh\mb.dir\alsc_
SHOPMILL_MANUALTURN_CYCLES=
MEASURE_CYCLES=F:\MMC_52\dh\mb.dir\almc_
PLC_PMC=
USER_CYCLES=
UserMMC=
UserNCK=
UserPLC=
UserZYG=
UserCZYK=
UserSTANDARD_CYCLES=
UserSHOPMILL_MANUALTURN_CYCLES=
UserMEASURE_CYCLES=
UserUSER_CYCLES=
UserPLC_PMC=
; NCU-specific texts in M:N environments
; the entry names are build by concatenating the string "PLC"
or "NCU" with
; the name of a NCU taken from netnames.ini file.

; Entry names are case sensitive!
; the entry values follow the same rules as in the Textfiles section
of this file
; the commented examples below correspond to the netnames.bsp
file delivered

[net TextFiles] (仅用于 SINUMERIK powerline)

;PLCRechteNCU = f:\dh\mb.dir\rechts
;NCURechteNCU = f:\dh\mb.dir\rechts
;PLCMittlereNCU = f:\dh\mb.dir\mitte
;NCUMittlereNCU = f:\dh\mb.dir\mitte
;PLCLinkeNCU = f:\dh\mb.dir\links
;NCULinkeNCU = f:\dh\mb.dir\links
; drive and device type specific alarm text files

[DriveTextFiles]

ALDEV_80E5=f:\dh\mb.dir\aldvs_
ALDRV_80E5=f:\dh\mb.dir\aldrs_
;ALDEV_808F=f:\dh\mb.dir\aldvu_
;ALDRV_808F=f:\dh\mb.dir\aldru_
ALDEV_x=f:\dh\mb.dir\aldvp_
ALDRV_x=f:\dh\mb.dir\aldrp_

[IndexTextFiles]

ALNX=f:\dh\mb.dir\alnx_
ALSI=f:\dh\mb.dir\alsi_

; default definition of the priorities of the different alarm types

[DEFAULTPRIO]

CANCEL=100
RESET=100
POWERON=100
NCSTART=100
PLC=100
PLCMMSG=100
MMC=100

; This section defines different characteristics of the protocol that

; specifies how particular error messages are to be registered.

[PROTOCOL]

用于控制信息满溢情况的过滤器表达式构成的详细信息，请参见章节“创建报警记录的存储区”。

; 记录用于选择报警信息的标准。符号

; is described in SINUMERIK 840D OEM package HMI User's Manual.

过滤器=表达式

; Characterizes the numbers of entries in the log file. If the actual

; number exceeds this value, the server starts to override old entries

; (principle: ring buffer)

Records=150

; Mode of alarm protocol writing

; if a number of less than -1 is specified, this number of protocol files is

; parallely held, and one of them is actually used

; if a number of -1 is specified, the protocol-file is not automatically updated

; if 0 is specified the protocol file is immediately updated

; if a number above 0 is specified, the protocol file is updated after the

; specified number of seconds elapsed with no intermediate alarm state

; changes. the protocol-file is always updated when you send the

; DDE-Command: "AlarmProt" to the MBDDE-server

; (principle: ring buffer)

DiskCare=-1 # attention: the mode changed from V4 to V5 !

也可以通过条目 DiskCare 实现多文件策略

在 Ini 文件中进行选择。

; information about the keys that may be used to clear an alarm

[KEYS]

; A function-key (e.g. ESCAPE) is assigned the cancel function.

; This makes it possible to acknowledge cancel-alarms on the

; PC development system, too.

Cancel=Escape

; This section specifies the names and paths of the help files that
; contain help texts in the WinHelp format.
; This list includes a maximum of 10 entries.
; The filename fragments must not exceed 6 chars, because it is
; extended automatically by the message module according to the
; selected language. A filename extension may be appended optionally.
; For additional information see SINUMERIK 840D OEM package
; HMI User's Manual.

[Helpcontext]

File0=help\alarm_ .pdf

对于 OEM 用户:

; reserve some sender names of HMI alarms. The names mbdde,
; NCU and PLC are reserved for mbdde internal purposes.

[MmcAlarmDisable]

DisableSenderOfMmcAlarm0=mbdde

DisableSenderOfMmcAlarm1=NCU

DisableSenderOfMmcAlarm2=PLC

DisableSenderOfMmcAlarm3=

DisableSenderOfMmcAlarm4=

DisableSenderOfMmcAlarm5=

DisableSenderOfMmcAlarm6=

DisableSenderOfMmcAlarm7=

DisableSenderOfMmcAlarm8=

A.17 MMC.INI

概览

I 在文件 MMC.INI 中设定了系统当前结构、单个系统组件通讯以及 HMI 上可用语言的相关设置。

```
[GLOBAL]
; connectiontype 0 = 预设: 连接 SINUMERIK 810D/840D
; 1 = 与 SINUMERIK 840Di 连接
connectiontype=0
NcddeServiceName=ncdde
ModeWithoutNc=False
ProductCode=MMC
; latency for the screensaver/屏幕保护程序的等待时间
MMCScreenOffTimeInMinutes=5
; ATTENTION PLEASE FOR USING a SIM1 - MACHINE !!!!!
; for working with a SIMNC arrange the following entries AND
; activate the section [NCDDE_STARTCMD] lower in this mmc.ini
; ATTENTION PLEASE FOR USING a SIM1 - MACHINE !!!!!
; for using M:N function set NcddeMachineName=MachineSwitch
; for working without NC set NcddeMachineName=local
; for working with SIMNC set NcddeMachineName=SIM1
; for connecting to a NC set NcddeMachineName=NCU840D
NcddeMachineName=NCU840D
; for using M:N function set NcddeDefaultMachineName=net:NCU_1
; for working without NC set NcddeDefaultMachineName=local
; for working with SIMNC set NcddeDefaultMachineName=SIM1
; for connecting to a NC set NcddeDefaultMachineName=NCU840D
NcddeDefaultMachineName=NCU840D
; for using M:N function set NcddeMachineNames=net,NCU840D
; for working without NC set NcddeMachineNames=
; for working with SIMNC set NcddeMachineNames=SIM1
; for connecting to a NC set NcddeMachineNames=NCU840D
NcddeMachineNames=NCU840D
```

```
; for connecting to a SIM add NcddeMachineNamesAdd1=SIM0
NcddeMachineNamesAdd1=SIM0,SIM1
; for using M:N function set NcddeStartupFile=ncdde6.nsk
; for working without NC set NcddeStartupFile=ncdde202.nsk
; for working with SIMNC set NcddeStartupFile=sim1dde5.nsk
; for connecting to a NC set NcddeStartupFile=ncdde6.nsk
NcddeStartupFile=ncdde6.nsk
; define a network uniq MMC name (8 characters) here
NcddeMmcName=__XXXX__
; 通过按下 1, 使得服务器如同 NCDDE、MBDDE、DH 服务器那样可见,
; 通过按下 ALT + TAB 可以载入至前景中, 以
; 能够在必要时将其参数化用于一个轨迹 (服务情况)。
; 为此注释该条目:
;ServerVisible = 1
```

```
[Local] 用于 OEM 用户
ADDRESS0=0,NAME=/NC,LINE=0
ADDRESS1=0,NAME=/PLC,LINE=0
ADDRESS4=/SELF,0
; 举例: 840D, 通过 BTSS 连接
```

```
[NCU840D] 仅用于 SINUMERIK powerline
ADDRESS0=13,NAME=/NC,SAP=d
ADDRESS1=13,NAME=/PLC,SAP=1
ADDRESS4=/SELF,10000103
; 举例:
; WINCOS-->CP-->TCPIP-->CP-->LINCOS-->NC/PLC/DRIVE0-5
; 1.) change TCPIP-Address below to your target TCPIP-Address
; 2.) do not forget to start your windows CP (e.g. cp_840di.exe)
; 3.) note: no disconnection detection in some cases
```

```
:[NCU840D] 仅用于 SINUMERIK solution line
;ADDRESS0=192.168.214.1,LINE=10,NAME=/NC,SAP=040d,PROFILE=CLT1__CP_L4_INT
;ADDRESS1=192.168.214.1,LINE=10,NAME=/PLC,SAP=0201,PROFILE=CLT1__CP_L4_IN
T
;ADDRESS2=192.168.214.1,LINE=10,NAME=/CP,SAP=0501,PROFILE=CLT1__CP_L4_INT
;ADDRESS10=192.168.214.1,LINE=10,NAME=/DRIVE0,SAP=0201,SUBNET=004B-
00000008:3, PROFILE=CLT1__CP_L4_INT
```

```
;ADDRESS11=192.168.214.1,LINE=10,NAME=/DRIVE1,SAP=0a00,PROFILE=CLT1__CP_
L4_INT
;ADDRESS12=192.168.214.1,LINE=10,NAME=/DRIVE2,SAP=0b00,PROFILE=CLT1__CP_
L4_INT
;ADDRESS13=192.168.214.1,LINE=10,NAME=/DRIVE3,SAP=0c00,PROFILE=CLT1__CP_L
4_INT
;ADDRESS14=192.168.214.1,LINE=10,NAME=/DRIVE4,SAP=0d00,PROFILE=CLT1__CP_
L4_INT
;ADDRESS15=192.168.214.1,LINE=10,NAME=/DRIVE5,SAP=0e00,PROFILE=CLT1__CP_
L4_INT
```

也可以通过 HMI 操作界面在操作区“开机调试” → “HMI” → “语言”中进行语言选择:

[LANGUAGE]

```
Language=UK
LanguageFont=Europe
Language2=GR
LanguageFont2=Europe
LanguageList=GR, UK, SP, FR, IT, CH
FontList=Europe, Europe, Europe, Europe, Europe, China
FontListKO=Europe2, Europe2, Europe2, Europe2, Europe2, China
LbList=deutsch, english, espanol, francais, italiano, chinese
```

[CONTROL] 仅用于与 PC/PG 相关的 HMI 高级

```
; Resolution: 0=fixed (640x480 default), 1=variable
Resolution=0
; BaseScreen specifies the resolution, for which the applications
; originally were developed (typically: 640x480)
; BaseScreen - default: 640x480
BaseScreen=640x480
```

如果控制系统已连接了一个鼠标，则鼠标指针按照标准位于指向屏幕右上角。用下列条目可以关闭该特性:

```
;MoveCursor=0: do not move cursor to upper right corner
;MoveCursor=0
```

[Dienste] 仅用于与 PC/PG 相关的 HMI 高级

```
FloppyDisk= a:
PCCard=
```

说明

如果您想要匹配不同显示类型/语言的字体大小，请拨打西门子热线。

BTSS 设置为显示机床数据，这些显示机床数据是工艺特定的，且也可以通过 HMI 操作界面在操作区“开机调试”→“机床数据”中设置。

```
[BTSS-Settings]
$MM_LCD_CONTRAST=7
$MM_DISPLAY_TYPE = 1
$MM_DISPLAY_MODE = 2
$MM_FIRST_LANGUAGE = 1
$MM_DISPLAY_RESOLUTION=3
$MM_SPIND_DISPLAY_RESOLUTION=3
...
$MM_MD_DRV_TEA_FILTER = 0
$MM_MD_DRV_TEA_IDX_LIMIT = 0
$MM_MD_TEXT_SWITCH=0
$MM_ACTION_LOG_MODE = 0xFE
$MM_PROGRAM_SETTINGS = 0
$MM_SW_OPTIONS = 0
$MM_HMI_TESTAUTOMAT_OPTION = 0
$MM_HMI_WIZARD_OPTION = 0
$MM_TRACE=0
```

也可以通过 HMI 操作界面在操作区“开机调试”→“HMI”→“语言”中设置运行记录器：
; ActionLog（运行记录器）参数

```
[ActionLog]
; CommitLatency tells how many seconds after creation an ActionLog
; record is written to disk at latest. If this value is less than 0 there is no
; time controlled automatic writing to disk at all
; Default : CommitLatency = -1
; no automatic writing
CommitLatency = -1
; DirectoryPath specifies the directory of the logfiles
DirectoryPath = D:\AC_LOG
; FileSize gives the upper limit of logfile size in bytes
FileSize = 40000
; VariableN allows monitoring of max 10 NC/PLC variables. The first
; parameter is numeric and provides the monitoring mode:
```

; if Bit0 is set and the value of the variable changes, the value changes
; are protocolled in the ActionLog files
; if Bit1 is set and the value of the variable changes, the ActionLog files
; are committed to disk
; Default : Variable0 = 1, /Channel/ProgramInfo/msg
; monitor part program messages
Variable0 = 1, /Channel/ProgramInfo/msg
; CommittedAlarms holds up to 10 comma separated alarm numbers.
; Changes of these alarms commit
; ActionLog data to disk
; Default: CommittedAlarms = 3000
; commit on emergency stop
CommittedAlarms = 3000
; for working with SIMNC uncomment the following section descriptor
; by this way a sim-process will be started to be used as a machine

:[NCDDE_STARTCMD]
CMD=dpncsim.exe /T:1 /P:0
WND_CLASS=SIM_LOCAL_NCK
WND_NAME=SIM_LOCAL_NCK
STARTUP_TIMEOUT=50
;WAIT_SYNC_MESSAGE=
CLOSE_TIMEOUT=30

[WP_SELECT]
; for displaying a message that TOA/TMA-data on NCK will be overwritten
;TOA_WARNING_MESSAGE = yes/no
TOA_WARNING_MESSAGE=yes

[PCU]
; device name of PCU for USB local
UsbFront=G: ' access rights of the PCU USB device (0-7) for the application machine,
program and services
AccessMachine=7
AccessProgram=7
AccessServices=7

```
[PROGRAM]
;0=message to user if a program of the DAT-file for MCSE was not found
; or a channel doesn't exist (default)
;1=skip all entries of DAT-file if the program or a channel was not found
IgnoreMissingProgramAndChan=0
;0=create a workpiece (optional by template) without opening by editor
;1=open partprogram, joblist or DAT-file after creating
; new workpiece (default)
OpenFileAfterNewWPD=1
```

对于多通道程序段编辑器，在 USER 目录中适用于：

```
[PROGRAM]
IgnoreMissingProgramAndChan=1
OpenFileAfterNewWPD=0
```

新建工件时，仅用工作列表而不用 MPF 的情况下，不再出现信息“无通道分配”。

参见

SEEDITOR.INI (页 335)

A.18 NETNAMES.INI

概览

在文件 NETNAMES.INI 中，通过 m:n 赋值确定，哪些操作组件 PCU（HMI）可以与哪些 NCU 在网络中连接（powerline），或者哪些 TCU 可以插入到哪些 PCU/NCU 上（solution line）。

同时确定：

- 操作组件名称
- NCU 名称
- NCU 之间的通讯连接
- 用于启动的唯一分配

文献： 扩展功能功能手册： 多个 NCU 上的多个操作面板，分散式系统（B3）

步骤

在文件 F:\mmc2\netnames.ini 中说明了所有的设计方法（powerline）。为了设计用于 m:n 通讯的 NETNAMES.INI，必须遵循下列顺序：

1. 将现有的文件 F:\mmc2\netnames.ini 复制到 F:\user\netnames.ini。
2. 文件 F:\user\netnames.ini 借助一个编辑器以下列步骤处理：
 - 在“logChanSetList”中确定组（最多 24 个）
 - 在“logChanList”中为 NCU 组和通道组确定唯一的名称。
 - 对于通道，在“logNCName”中确定 NCU 名称，在“ChanNum”中确定 NCU 上的通道号，确定软件释放是/否（通过指令 IsAGap = True/False）。
 - 对于 HMI，确定标准机床“DEFAULT_logChanGrp”和用于启动的“DEFAULT_logChan”中的标准通道。
3. 存储文件 F:\user\netnames.ini。

说明

NC 写入 HMI 的通道号码至 DB19.DBB22。

下例简单说明 HMI 的 1:1 分配：大致上对于 NCU，conn_2 不适用且所有通道都在 NCU_1 上。

文件 netnames.ini 的内容

； MMC 识别号的识别部分
； PCU50 PL3 服务器

[own]

owner= PCU_1

; Description of possible connections

; 连接部分

[conn PCU_1]

conn_1= NCU_1

conn_2= NCU_2

conn_3= NCU_3

EXTCALL_conns=conn_1, conn_2, conn_3

; Description of significant net-parameters

; NCU 组件的说明部分

[param network]

bus= btss

; MMC 说明

[param PCU_1]

mmc_typ = 0x40

mmc_bustyp = BTSS

mmc_address= 1

mstt_address=3

name = PCU50

start_mode = ONLINE

[param NCU_1]

type =NCU_573

nck_address= 11

plc_address= 11

name = NCU1

PlcSymbolFile= ...

[param NCU_2]

type =NCU_573

nck_address= 12

```
plc_address= 12  
name = NCU2  
PlcSymbolFile= ...
```

```
[param NCU_3]  
type =NCU_572  
nck_address= 14  
plc_address= 14  
name = NCU3  
PlcSymbolFile= ...
```

; 通道数据

```
.*****  
,
```

```
[chan PCU_1]  
DEFAULT_logChanSet = Stat_1  
DEFAULT_logChan = N1_K1  
ShowChanMenu = True  
logChanSetList = Stat_1, Stat_2, Stat_3
```

```
.*****  
,
```

```
[Stat_1]  
logChanList = N1_C1, N1_C2
```

```
[N1_K1]  
logNCName = NCU_1  
ChanNum = 1
```

```
[N1_K2]  
logNCName = NCU_1  
ChanNum = 2
```

```
.*****  
,
```

```
[Stat_2]
```

logChanList = N2_C1, N2_C2

[N2_K1]

logNCName = NCU_2

ChanNum = 1

[N2_K2]

logNCName = NCU_2

ChanNum = 2

.*****
,

[Stat_3]

logChanList = N3_C1, N3_C2

[N3_K1]

logNCName = NCU_3

ChanNum = 1

[N3_K2]

logNCName = NCU_3

ChanNum = 2

; 文件结束

参见

设计用于直接通道选择的通道菜单 (1:1 配置) (页 54)

设计一个双通道显示 (页 57)

A.19 OEMFRAME.INI

概览

在文件 OEMFRAME.INI 设定了 OEM 应用程序的初始特性。

```
[SysMeter]
; let OEMFrame just look for Windows which have been
; created by the OEMApp
; default is 0 (means NO)
;fSearchOnlyForTaskWindow=1
; let OEMFrame delay INIT_COMPLETE, if set to -1,
; OEMFrame will not send any WM_INITCOMPLETE at all,
; this value must be given in microseconds
;nDelayInitComplete=10000
; switching WS_SYSMENU, WS_MINIMIZEBOX and WS_MAXIMIZEBOX off
WindowStyle_Off = 720896
; other possible task entries aren't used
WindowStyle_On =
x =
y =
Width =
Height =
```

A.20 PARAM.INI

概览

在该文件中记录有用户通过 HMI 操作界面输入的刀具补偿数据。

```
[ACCESSLEVEL]
READ_SYSVAR=7
EDIT_VIEW=7
; 刀具补偿用户数据
; 刀具特定的参数

[ToolParams]
```

```
UserDataParamName1=T_Param_1  
UserDataParamName2=T_Param_2  
UserDataParamName3=T_Param_3  
UserDataParamName4=T_Param_4  
UserDataParamName5=T_Param_5  
UserDataParamName6=T_Param_6  
UserDataParamName7=T_Param_7  
UserDataParamName8=T_Param_8  
UserDataParamName9=T_Param_9  
UserDataParamName10=T_Param_10  
; 刀沿专用参数
```

```
[ToolEdgeParams]  
UserDataParamName1=D_Param_1  
UserDataParamName2=D_Param_2  
UserDataParamName3=D_Param_3  
UserDataParamName4=D_Param_4  
UserDataParamName5=D_Param_5  
UserDataParamName6=D_Param_6  
UserDataParamName7=D_Param_7  
UserDataParamName8=D_Param_8  
UserDataParamName9=D_Param_9  
UserDataParamName10=D_Param_10  
; 刀具补偿数据
```

A.21 PARAMTM.INI

概览

刀具管理操作界面的所有说明都保存在文件 ...\\user\\paramtm.ini 中。

参见

文件 paramtm.ini 的结构 (页 191)

A.22 REGIE.INI

概览

在文件 REGIE.INI 中记录有 HMI 应用程序及其附属子系统的开始和结束设置。

[Miscellaneous]

; necessary, if regie shall support an 'EXIT' button

; set 'ExitButton' to True, and 'ExitButtonIndex'

; identifies index of exit button

; The softkey-text for this button MUST be entered into

; \mmc2\language\re_xx.ini

; where xx is the language-abbreviation eg.:

; re_gr.ini (german text)

; re_uk.ini (english text)

ExitButton= True

ExitbuttonIndex = 15

; Protection level for exit button

ExitButtonAccessLevel = 7

; Dialog box to ask user

ExitButtonQueryUser = True

; 有关其它条目的提示, 请参见 OEM 文献

; 激活 PCU50 的温度监控

; TemperatureMonitoring = True

TemperatureMonitoring = False ; 关闭预设

文献: OEM 软件包 HMI

休眠 (仅用于 Windows XP)

用于休眠变量的设置存在于参数休眠模式的 [Miscellaneous] 段中

也可通过 HMI 高级的操作界面进行此设置。为此, 在操作区“开机调试” → “HMI” → “系统设置”中选择。

[Miscellaneous]

HibernateMode = ...

可以设置下列值:

HibernateMode = Off

预设置: 关闭
不进行休眠保护和休眠重新启动。

HibernateMode = Normal

HMI 高级首先结束所有应用程序且控制系统在关机后自动重启。

HibernateMode = Advanced 无法实现

; index of task, which is started by the regie after power on
; (may be differ in some circumstances from the default task index,
; which is the task index from [TaskConfiguration] with the lowest number.
PoweronTaskIndex = 0

[StartupConfiguration]

在该段中可以在 12 至 24 的范围内启动 OEM 软件。

[StartupConfiguration] 段按如下方式划分:

0 至 12 预留用于西门子产品的扩展

12 至 24 空置用于 OEM 应用程序

25 至 32 预留用于模拟

33 至 47 预留用于西门子应用程序 (在目录 ADD_ON 中)

示例:

Startup33 = name := ITS, Timeout :=15000

Startup34 = name := HMIVers, Cmdline := "/StartUp", Timeout :=10000

[TaskConfiguration]

; 第 3 栏

; Task16 = name := oemframe, cmdline := "sysmeter", Timeout := 10000,

; ClassName := "SysMeter", HeaderOnTop := False, PreLoad := False

如果自任务 16 起分配一个任务或退出按钮, 则仅显示第 3 个水平软键栏。

; 4. bar : located to task 96 - 103

;Task96 = name := oem, Timeout := 10000

;Task97 = name := oem, Timeout := 10000

;Task98 = name := oem, Timeout := 10000

;Task99 = name := oem, Timeout := 10000

;Task100 = name := oem, Timeout := 10000

;Task101 = name := oem, Timeout := 10000

;Task102 = name := oem, Timeout := 10000

;Task103 = name := oem, Timeout := 10000

如果自任务 96 起分配一个任务或退出按钮, 则仅显示第 4 个水平软键栏。任务 50 至任务 87 的条目可以由 OEM 使用。

[CommandToTask]

; only necessary for HMI invocation through NCU via command channel

plc_select = #13

Cycles = #14

在给段中可以设置，HMI 自 NCU 出发通过命令通道启动。

文献： 开机调试手册“补充操作界面”
章节“激活 NC 中的对话画面”

A.23 SEDITOR.INI

概览

通过该文件的设置或自身补充可以对 **多通道程序段编程（选项）** 中的显示进行修正。

包含符号 [*User*] 的段允许将机床制造商循环和用户识别并显示为“程序步”。以此可以设置其它的详细信息。

为了设计程序步，可以在下列段中进行设置：

[SyncMarks_User]	同步标记的扩展名
[USER_VARIABLES]	变量，用于带有同步标记的通道
[Old_or_User_Cycles_USER]	用于自身循环的扩展名；在这里可以输入任意的变量名称。
[ProgSupport_User]	自身循环的扩展名

为了设置时使用合并机械机制，与 ini 文件的标准规格相反，仅须在目录 **add_on**、**oem** 和 **user** 下存放改变和补充的内容。

句法规则

对于扩展，请注意下列提示：

```
.*****  
;  
;The following rules must be used for the sections:  
;- [SyncMarks_Siemens]  
;- [SyncMarks_User]  
;- [Old_or_User_Cycles_Siemens]  
;- [Old_or_User_Cycles_USER]  
;  
;- You can use either *- or ?-statements.  
; Do not mix these placeholders:  
;  
; L1* -> OK  
; L1?? -> OK  
; L1?1* -> NOT OK !!!
```

```

;
;- ? is replaced with numbers from 0 to 9, so the step editor
; will find the following strings:
;
; L1? -> L10, L19, L15, but NOT L1A
; LB?? -> LB23, LB99, LB00, but NOT LB1A or LBd6
;
;- You can use the ?-statement once per entry:
; L1? -> OK
; L1?? -> OK
; L23??? -> OK
; L2??3? -> NOT OK !!!
;*****
;

```

[MISC]

IconPath

相关的路径在存放图标的目录上。

路径与 ..\mmc2 相关。相同的目录也可以在 Add_on、oem 和 user 中。也在这些目录中查找图标。

预设置: IconPath = ae\skpicto

EndIcon

最后程序段的图标。这一个图标必须放置于 IconPath 下规定的目录中。

GCodeIcon

用于自由 DIN 代码的图标。这一个图标必须放置于 IconPath 下规定的目录中。

ShowBlockNumber

显示行号 (例如 N210)。总是显示程序段第一行的行号。

0: 不显示行号

1: 显示行号

TimeFormat

显示的时间可以设计为 VB 句法格式, 例如小数点位数。

####0.000: 显示带 3 个小数点后位数的时间。

CopySyncBuffers

对于同步视图, 可以在这里确定, 是否应在复制时将插入用于同步动作的空程序段也一同复制。

0: 不复制空程序段

1: 复制空程序段

[INSERT]

TemplateForNewBlock

用按键组合 **CTRL+I** 可以在程序段的编辑器模式下插入 **SEFORM** 指令。可在这里设计待插入的文本。

[Cache]

已使能

使用快取可以快速打开文件。

分开存放程序段视图的各个数据并在零件程序未更改时可以访问这些数据。

目前仅可以使用下列设置：

0: 快取未激活

1: 快取已激活

[DefaultIcons]

如果在各个定义中未规定图标，则显示该图标，例如在 **SEFORM(...)** 中不包含图标时。这一个图标必须放在 **IconPath**（图标路径）下规定的目录中。

块

主程序段图标 – **SEFORM(..., 0, ...)**

SubBlock

子程序段图标 – **SEFORM(..., 1, ...)**

Sync

同步标记图标 – 例如 **WAITM(...)**

Cycle

循环图标

GeoProz

轮廓基准图标

[SyncMarks_User]

在这里，机床制造商可以输入其用于同步标记的扩展名。

```

;Definition of synchronization marks (search texts)
;sync1 = WAITM>(*n,*c,*c,*c,*c,*c,*c,*c,*c,*c,*c)
; '#' : seperator: text before is searched in one
; channel, text behind in all others.
; It isn't possible to synchronize this kind of
; synchronization marks !!!
; *n': identifier for synchronization numbers
; WAITM(*n,1,2,3)
; *c': channel WAITM(1,*c,*c,*c), if own channel
; is missing, it will be added like WAITM-strategy
;
; Add additional / language dependent Text:
; WAITM(*n,*c,*c,*c,*c,*c,*c,*c,*c,*c,*c) "$85000 The Text"
;
;sync1_lco = clock.bmp
; belonging icon for defined sync mark
;sync1_Chans = 1,2,4
; Project the channels to synchronize here
; e.g.: you have a M-function (M4711)
; to synchronize the channels 1, 2 and 4
; Then you can write here: syncX=1,2,4
; If you synchronize the part programs the
; channel 3 will not be considered.
; If you don't project anything here, in all
; channels must stand the M4711 call
; respective if there are *C in [SyncMarks]
; only these channels will be synchronized.
;SIEMENS
[SyncMarks_Siemens]
sync1 = WAITM>(*n,*c,*c,*c,*c,*c,*c,*c,*c,*c,*c)
sync1_lco = clock.bmp
sync1_Chans =
;USER
[SyncMarks_User]

```

sync1

定义搜索文本。这关系到下列句法：

* 占位符； 在 '*'中包含的文本必须在所有通道中都相同。

n 通过编号识别，例如 WAITM(*n,1,2,3)

通过该编号可以对等待标记进行归类。

*c 通道编号，不必保持同一顺序，
不是一定要指定自身的通道
(例如： WAITM(1,*c,*c,*c)。

通道分隔符： 分隔符之前的文本在一个通道中搜索
并在其他通道中搜索其后的文本。这些同步标记
不能互相同步。

sync1_lco

所属的图标必须放在于 IconPath 下规定的目录中。

sync1_Chans

在这里可以规定，是否仅在指定的通道中搜索所属的同步标记。如果条目为空，则在所有通道中搜索，或者在已设计 *c 的情况下，在规定的通道中搜索。

示例：

M4711 为仅在通道 1、2 和 6 中有效的同步。条目必须如下所示：

sync2 = M4711

sync1_lco = clock.bmp

sync2_Chans = 1,2,6

[Old_or_User_Cycles_User]

在这里，机床制造商可以输入其用于自身循环的扩展名。

举例：

Cycle1 = Cycle*

Cycle1_Type = 1

Cycle1_lco = cycle.bmp

Cycle2 = Pocket*

Cycle2_Type = 1

Cycle1_lco = cycle.bmp

...

CycleX

定义循环，不支持图形。在这里同样可以输入子程序调用。此外还可以规定一个与语言相关的文本。X 在这里表示循环号码，该循环号码必须等同于通过三个共同所属的条目。

句法：

CycleX = L100(*

CycleX = L100(* "\$85001"

CycleX = L100(* "\$85001/fester
Text"

显示，例如：

"L100(2,3,4)"

存储在 85001 中与语言相关的文本。

存储在 85001 中与语言相关的文本为 + "/固定的文本"

CycleX_Type

确定是否应将循环显示为子程序段或显示为主程序段。

1: 子程序段

0: 主程序段

CycleX_Ico

所属的图标必须放在于 IconPath 下规定的目录中。

[ProgSupport_User]

在这里，机床制造商可以输入其用于自身循环的扩展名。

举例：

;Programming Support - Siemens Standard

[ProgSupport_Siemens]

NCG_1=CYC82

NCG_2=CYC86

NCG_3=CYC83

NCG_4=CYC84

NCG_5=CYC84_1

NCG_6=CYC840_1

NCG_7=HOL1

NCG_8=HOL2

NCG_9=POS1

NCG_10=WPOS

...

;drilling

[CYC82]

;drilling

NCG_1_Label=NC1

NCG_1_Text=\$80854

NCG_1_Ico=cm_poly_dr_drill.bmp

NCG_1_Level=0

NCG_2_Label=NC2

NCG_2_Text=\$80855

NCG_2_Ico=cm_poly_dr_drill.bmp

NCG_2_Level=0

;reaming

NCG_3_Label=NC3

NCG_3_Text=\$80853

NCG_3_Ico=cm_poly_dr_reaming.bmp

NCG_3_Level=0

```
NCG_4_Label=NC4  
NCG_4_Text=$80852  
NCG_4_Ico=cm_poly_dr_reaming.bmp  
NCG_4_Level=0  
...
```

```
[CYC86]  
;boring  
NCG_1_Label=NC1  
NCG_1_Text=$80666  
NCG_1_Ico=cm_poly_dr_bore.bmp  
NCG_1_Level=0  
NCG_2_Label=NC2  
NCG_2_Text=MCALL CYCLE86  
NCG_2_Ico=cm_poly_dr_bore.bmp  
NCG_2_Level=0
```

```
[CYC83]  
;deep hole drilling  
NCG_1_Label=NC1  
NCG_1_Text=$80794  
NCG_1_Ico=cm_poly_dr_deepdrill.bmp  
NCG_1_Level=0  
NCG_2_Label=NC2  
NCG_2_Text=MCALL CYCLE83  
NCG_2_Ico=cm_poly_dr_deepdrill.bmp  
NCG_2_Level=0
```

说明:

这里将两个图形支持的循环作为示例说明。

在 NC 程序在循环位置的下列行中出现:

```
;NCG#CYC82#ACST.DIR\bohren.com#NC1#2#*NCG;*RO*;*HD*  
;#1#1#1#1#1####"M3"#####1##1#*NCG;*RO*;*HD*  
CYCLE82(1,2,3,4,,0)  
;#END#*NCG;*RO*;*HD*  
[ProgSupport_User]  
Nr
```

在这里必须规定下列条目的数量。

NCG_1

各个搜索文本的条目或者循环名称。

一般情况下，NC 程序中的文本位于“;NCG#”之后。

在上述例子中为“CYC82”。

由于为了这样一个循环可以给出多个型号（例如 NC1, NC2, ...），对于该条目必须在 INI 文件中设立一个附加的段 [CYC82]。

用于循环的段（例如：[CYC82]）

Nr

循环版本数。

NCG_1_Label

版本“名称”。一般情况下，NC 程序中的文本位于 COM 文件的路径后（NC1, NC2, ...）。

NCG_1_Text

显示文本。可以规定一个固定的文本、一个与语言相关的文本（\$85111）或这两个文本的组合（\$85111/）。

NCG_1_Ico

所属的图标必须放在于 IconPath 下规定的目录中。

NCG_1_Level

确定是否应将循环显示为子程序段或显示为主程序段。

1: 子程序段

0: 主程序段

文献： 开机调试手册“补充操作界面”

A.24 SIMTOGEO.INI

概览

带所需数据用于模拟的各个刀具类型的示例文件。

说明

如果段[Settings]中 Link= 0，则其余的 SIMTOGEO.INI 条目无效。

```
[SETTINGS]
; this entry is depending on settings like
;
; MD 18206: $MN_MM_NUM_CCS_TOA_PARAM=1
; MD 18207: $MN_MM_TYPE_CCS_TOA_PARAM[1]=4
; MD 18080: $MN_MM_TOOL_MANAGEMENT_MASK='H4'
;
; if wanted and / or other oem parameters are used the index can be changed
; ( e.g. TC_DPCS4 if $MN_MM_NUM_CCS_TOA_PARAM=4 )
; or any other oem variable as TC_DPCS1 can be used :
;
;LINK=TC_DPC7
;
; MD 18096: $MN_MM_NUM_CC_TOA_PARAM=7
; MD 18097: $MN_MM_TYPE_CC_TOA_PARAM[7]=4
; MD 18080: $MN_MM_TOOL_MANAGEMENT_MASK='H4'
;
LINK=0 ; example LINK=TC_DPCS1
; example data sets

[500.1]; 粗加工刀具 1 的唯一标识符 (t,d)
tool_type=500; 刀具类型, 如未设置, 则使用
; $TC_DP1
insert_length= ; TC_DP8, 托盘长度
holder_angle= ; TC_DP10, 刀架角度
reference_direction= ; TC_DP11, 切削方向
```

clearance_angle= ; TC_DP24, 后角

[500.2]; 精加工刀具 1 的唯一标识符 (t,d)

tool_type=510 ; 刀具类型, 如未设置, 则使用\$TC_DP1

insert_length= ; TC_DP8, 托盘长度

holder_angle= ; TC_DP10, 刀架角度

reference_direction= ; TC_DP11, 切削方向

clearance_angle= ; TC_DP24, 后角

[520.1]; 唯一的切槽刀 1 标识符 (t,d)

tool_type=520 ; 刀具类型, 如未设置, 则使用\$TC_DP1

insert_length= ; TC_DP8, 托盘长度

reference_direction= ; TC_DP11, 切削方向

[530.1]; 唯一的切槽刀具 1 的标识符 (t,d)

tool_type=530 ; 刀具类型, 如未设置, 则使用\$TC_DP1

insert_length= ; TC_DP8, 托盘长度

reference_direction= ; TC_DP11, 切削方向

[540.1]; 唯一的螺纹车刀 1 的标识符 (t,d)

tool_type=540 ; 刀具类型, 如未设置, 则使用\$TC_DP1

reference_direction= ; TC_DP11, 切削方向

[160.1]; 钻孔螺纹铣刀的唯一标识符 (t,d)

tool_type=160 ; 刀具类型, 如未设置, 则使用\$TC_DP1

tip_angle= ; TC_DP10, 顶角

;TC_DPV3= ; 1=+X; -1=-X

;TC_DPV4= ; 1=+Y; -1=-Y

;TC_DPV5= ; 1=+Z; -1=-Z

[151.1]; 锯的唯一标识符 (t,d)

tool_type=151 ; 刀具类型, 如未设置, 则使用\$TC_DP1

width= ; TC_DP9, 宽度

;TC_DPV3= ; 1=+X; -1=-X

;TC_DPV4= ; 1=+Y; -1=-Y

;TC_DPV5= ; 1=+Z; -1=-Z

; 用于 tool_types 200-299 的数据组

[200.001]; 唯一标识符 (t,d)

tool_type= ; 刀具类型, 如未设置, 则使用\$TC_DP1

tip_angle= ; TC_DP24, 顶角

;TC_DPV3= ; 1=+X; -1=-X

;TC_DPV4= ; 1=+Y; -1=-Y

;TC_DPV5= ; 1=+Z; -1=-Z

; default tool data sets reserved by SIEMENS AG for later use

; default data sets are recognized by non digit values

; examples

:[500]; 编号与刀具类型相对应, 粗加工刀具

;insert_length= ; TC_DP8, 托盘长度

;holder_angle= ; TC_DP10, 刀架角度

;reference_direction= ; TC_DP11, 切削方向

;clearance_angle= ; TC_DP24, 后角

:[510]; 编号与刀具类型相对应, 精加工刀具

;insert_length= ; TC_DP8, 托盘长度

;holder_angle= ; TC_DP10, 刀架角度

;reference_direction= ; TC_DP11, 切削方向

;clearance_angle= ; TC_DP24, 后角

A.25 TASKCONF.INI

概览

在文件 TASKCONF.INI 中设定用户专用操作区的软键与软键栏。

[CONTROL]

MDIList=MntMMC\MntMMC.mdi

ControlFile=MntMMC\MntMMC.zus

ScreenTwips=1

[DEBUG]

;MDISize 0 = Debug, 1 = Full screen

MDISize=0

;按钮: 0 = 不可视按钮, 1 = 可视按钮

Stop_Button=1

; 在这里设计一个框架应用程序。使用 regie.ini 中设定的段名称, 而不是

; APPLICATION (应用程序)。

; 对于 regie.ini 中, 每一个使用

; 任务 i = name:= mntmmc cmdLine:= "SektionName", ... 进行设定

; 此处的段名称必须是一个独立的段

; 设计

[APPLICATION]

; HSK1.. - HSK8.. (水平软键)

; HSK9.. - HSK16.. (ETC 软键)

; VSK1.. - VSK8.. (垂直软键)

; 背景画面 (默认: 无)

; 1: 机床配置

; <文件名>: 从 OEM 目录装载位图

Picture=

; 索引 [1,..] 初始应用程序 (默认: 无)

StartIndex=

; 0: 永久任务 (默认)

; <> 0: 取消时终止任务

TerminateTask=

```
=====;
; 对于每个软键 (HSK1 - HSK16 / VSK1 - VSK8) ;
=====;
; taskid (见 regie.ini) 或逻辑任务名称
; (program/<root> 或 ExitButton)
; !!! ExitButton, 用于 HMI 高级关机的显示按钮 !!!
HSK1Task=
; 0: 取消调用过程 (默认)
; <> 0: 取消时终止应用程序 (添加
; 终止 HSK1Command 的任务)
; 必须由应用程序支持!
HSK1TerminateTask=
; 0: SwitchToChild (默认)
; 1: SwitchToTask -> 仅切换至任务
; 2: SwitchToTask2 -> 仅切换至指定任务的子任务
; (如果存在)
HSK1IsTask=
; 通过应用程序评估命令字符串 (默认: 空)
HSK1Command=
; 软键访问级 (0 - 7, 默认: 7)
HSK1AccessLevel=
; 没有与 NCK 的连接
; <> 0: 链接软键到 NCK
; (在没有与 NCK 的连接时锁住软键)
HSK1NckLink=
; 默认软键文本, 忽略用于逻辑任务名称
HSK1SkText=
; 与语言相关的软键文本, 忽略用于逻辑任务名称
HSK1SkText_GR=
HSK1SkText_UK=
```


缩略语列表

B.1 缩略语

缩写	意义
A	PLC 中的输出
ASCII	American Standard Code for Information Interchange: 美国信息互换标准码
BAG	工作方式组
BTSS	操作面板接口
CAD	计算机辅助设计
CNC	Computerized Numerical Control (计算机数字控制)
CR	回车
DAU	数字模拟转换器
DB	PLC 中数据块
DBB	PLC 中数据块字节
DBW	PLC 中数据块字
DBX	PLC 中数据块位
DDE	Dynamic Data Exchange: 动态数据交换
DIN	德国工业标准
DIR	Directory: 目录
DPM	双端口内存
DOS	磁盘操作系统
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
DW	PLC 中的数据字
E	PLC 中的输入
EG	扩展设备
ESR	扩展的停止和退回
FRAME	数据段(框架)
FIFO	先进-先出: 数据如何保存在存储器以及如何重新调用的过程。
GP	主程序
GUD	Global User Data (全局用户数据)
HD	Hard Disk: 硬盘
HMI	Human Machine Interface: 控制系统操作区

缩略语列表

B.1 缩略语

缩写	意义
HSA	主轴驱动
HW	硬件
开机调试	Startup
IKA	Interpolative Compensation: 可插补补偿
INC	Increment: 增量尺寸
INI	Initializing Data: 初始化数据
IPO	插补器
ISO	国际标准组织
JOG	Jogging: 手动工作方式
K1 .. K4	通道 1 到通道 4
LED	Light Emitting Diode: 发光二极管
LF	线路馈电
K _v	回路放大系数
LUD	Local User Data: 局部用户数据
MB	兆字节
MCP	Machine Control Panel 机床控制面板 (→ MSTT)
MD	机床数据
MDA	Maual Data Automatic: 手动数据输入
MCS	机床坐标系
MLFB	机器可识别产品符
MPF	Main Program File: NC 零件程序 (主程序)
MPI	Multi Port Interface: 多端口接口
MSTT	(机床控制面板)
NC	数字控制: 数字控制装置
NCK	Numerical Control Kernel: 带有程序段处理, 运行范围等等的数字内核
NCU	Numerical Control Unit: NCK 硬件单元
NV	零点偏移
OEM	原装设备制造商
OP	Operation Panel: 操作面板
PCU	可编程控制单元
PCMCIA	Card International Association: 存储卡标准
PG	编程器
PLC	Programmable Logic Control:
REF	返回参考点功能
REPOS	再定位功能
ROV	Rapid Override: 输入端校正
RPA	R-Parameter Active: NCK 中的存储范围用于 R 参数编号的 R- NCK
SBL	Single Block: 单程序段
SD	设定数据
SDB	系统数据块
SEA	Setting Data Active: 设定数据符号 (文件类型)
SK	软键

缩写	意义
SKP	Skip: 跳过程序段
SPF	Sub Program File: 子程序
SRAM	静态存储器 (缓存)
SUG	砂轮圆周速度
SW	软件
SYF	System Files: 系统文件
TEA	Testing Data Aktive: 机床数据标识
TO	Tool Offset: 刀具补偿
TOA	Tool Offset Active: 刀具补偿符号 (文件类型)
UFR	User Frame (用户框架)
VSA	进给驱动
WCS	工件坐标系
WZK	刀具补偿
WZW	换刀
ZOA	Zero Offset Active: 零点偏移数据符号 (文件类型)

索引

A

AEDITOR.INI, 293
ANSI 表/字体, 86

C

CAPSLOCK, 12

D

DG.INI, 297
DGOVW.INI, 297
DGOVWTXT.INI, 298
DH.INI, 299
DINO.INI, 295
DPDH.INI, 299
DPSIM.INI, 79, 299

E

EXTCALL, 48

F

Flash File System (FFS), 102

H

HEADER.INI, 301
HMI
 功能, 91
 配置, 21
HMI 分析仪
 分析数据, 165
 示例, 167
 配置, 162
 操作, 163
HMIAnalyzer.INI, 162
HMIDESK.INI, 139, 305

I

IB.INI, 306
IF.INI, 308

K

KEYS.INI, 309

L

LOGDRIVE.INI, 66, 309

M

m:n 配置, 59
MASCHINE.INI, 312
MBDDE.INI, 315
MD 10240
 SCALING_SYSTEM_IS_METRIC, 273
MD 11290
 DRAM_FILESYSTEM_MASK, 102
MD 11291
 DRAM_FILESYSTEM_SAVE_MASK, 102
MD 18118
 MM_NUM_GUD_MODULES, 83
MD 18120
 MM_NUM_GUD_NAMES_NCK, 83
MD 18130
 MM_NUM_GUD_NAMES_CHAN, 83
MD 18150
 MM_GUD_VALUES_MEM, 83
MD 18160
 MM_NUM_USER_MACROS, 84
MD 18170
 MM_NUM_MAX_FUNC_NAMES, 84
MD 18180
 MM_NUM_MAX_FUNC_PARAM, 84
MD 182420
 MM_MAX_SIZE_OF_LUD_VALUE, 84
MD 18351
 MM_DRAM_FILE_MEM_SIZE, 102

MD 20098
 DISPLAY_AXIS, 105
MD 28082
 SYSTEM_FRAME_MASK, 99
MD 30100
 CTRLOUT_SEGMENT_NR, 103
MD 9009
 KEYBOARD_STATE, 12
MD 9032
 HMI_MONITOR, 117
MD 9033
 MA_DISPL_INVERT_DIR_SPIND_M3, 110
MD 9034
 MA_NUM_DISPLAYED_CHANNELS, 57
MD 9052
 SHOW_CHANNEL_SPANNING_STAT, 126
MD 9203
 USER_CLASS_WRITE_FINE, 98
MD 9220
 USER_CLASS_PRESET, 99
MD 9245
 MA_PRESET_FRAMEIDX, 100
MD 9247
 USER_CLASS_BASE_ZERO_OFF_PA, 104
MD 9248
 USER_CLASS_BASE_ZERO_OFF_MA, 104
MD 9422
 MA_PRESET_MODE, 99
MD 9424
 MA_COORDINATE_SYSTEM, 98
MD 9428
 MA_SPIND_MAX_POWER, 107
MD 9429
 MA_SPIND_POWER_RANGE, 107
MD 9440
 ACTIVATE_SEL_USER_DATA, 100
MD 9451
 MM_WRITE_ZOA_FINE_LIMIT, 98
MD 9676
 CMM_DIRECTORY_SOFTKEY_PATH1, 67
MD 9677
 CMM_DIRECTORY_SOFTKEY_PATH2, 67
MD 9678
 CMM_DIRECTORY_SOFTKEY_PATH3, 67
MD 9679
 CMM_DIRECTORY_SOFTKEY_PATH4, 67
MMC.INI, 321

N

NC 语言指令, 104
NCU 连接, 8

NCU 连接向导, 136
NETNAMES.INI, 48, 54, 327

O

OEM 开机调试, 20
OEMFRAME.INI, 331

P

PARAM.INI, 331
PARAMTM.INI, 191, 332

PLC

 主轴选择, 118
 机床数据, 111
 应答报警, 25
 通道选择, 118
 编程, 113
PRESET (预设), 99

R

REGIE.INI, 333
 HMI 引导启动, 79
 要求重新启动, 22
 睡眠模式, 333

S

SEEDITOR.INI, 335
SimNCK, 79
SIMTOGEO.INI, 80, 343
STAT (机床位置), 101

T

t:m:n 配置, 60
TASKCONFIG.INI, 346
TU (回转轴位置), 101

V

V.24 接口, 50

二划

- 刀位类型, 179, 258
 - 级别, 183
- 刀库, 253
 - 分配/脱开, 187
 - 选择, 243
 - 装载配置, 188
- 刀具中心轨迹, 78
- 刀具对话框数据, 279
- 刀具目录, 271
- 刀具库, 171
 - 参数, 186
 - 配置, 184, 187
- 刀具补偿, 100
- 刀具管理(WZV), 45
- 刀具箱, 271
- 刀沿对话框数据, 280

三划

- 上部屏幕, 39

四划

- 中间存储器, 173, 255
- 切换通道, 55
- 切割刀沿, 80
- 匹配刀具, 76
- 匹配装备数据, 75
- 双通道显示, 57
- 无刀具管理情况下的刀具选择, 46

五划

- 主轴符号反向, 110
- 代码载体
 - MMC.INI, 274
 - 开机调试, 274
 - 文件, 274
 - 对话框数据, 284
 - 数据类型, 281
 - 数据格式, 284
- 处理 INI 文件, 17
- 对刀, 99
- 生成报警文本, 88
- 用于 OEM 的 INI 文件, 20
- 用于机床数据的明码文本, 111
- 用户视图, 95

- 用户控制文件, 114
- 记录文件
 - 内容, 158
 - 读出, 161

六划

- 交货状态, 7
- 任务准备
 - TSEARCH, 268
 - 刀具, 260
 - 用户权限, 262
 - 再次激活, 268
 - 状态显示, 261
 - 定位, 268
 - 软键文本, 266
 - 通配符, 266
- 优化, 78
- 回转轴位置, 101
- 多文件策略, 23
- 多通道程序段编程, 326, 335
- 存取权限
 - 驱动器, 68
 - 修改预设置, 26
 - 基准偏移, 104
- 存储器空间 DRAM, 102
- 许可证密钥, 15
- 设计 OEM 文本, 131, 304
- 设计 OEM 图标, 132
- 设计软键文本, 242
- 设计屏幕保护程序, 21
- 设置工艺, 41
- 设置实际值, 99
- 访问等级
 - NC 语言指令, 104
 - NC 数据, 103
 - 方案, 13
- 过滤器
 - 用于报警, 23
 - 在一个刀库上限制, 267
 - 条件, 264
 - 附加数据, 265
 - 查找标准, 263
 - 标准, 263
 - 标题, 266

七划

- 位图, 249
- 批量开机调试, 144
- 报警

- 设置报警记录, 22
 - 编号, 85
- 系统设置, 10
- 系统极限, 16
- 系统资源, 151, 153
- 补充服务一览, 35
- 运行记录仪, 155
- 运行信息 (PLC), 111
- 驱动器
 - 设置画面, 69
 - 通过机床数据进行连接, 67

八划

- 参考刀位, 180
- 参数化
 - 位图, 249
 - 清单, 221
- 定义文件, 104
- 建立操作菜单, 28
- 服务显示, 151
- 服务轴, 151, 153
- 英制/公制, 271
- 询问对话框, 22
- 软键, 69
- 软键 ‘语言选择’, 27
- 软键标签, 67

九划

- 临近刀位, 246
- 屏幕截图, 142
- 带有工作列表的工件, 43
- 带有符号的通道显示, 124, 130
- 故障识别, 118
- 故障信息报告至 PLC, 115
- 显示
 - 标题中的信息, 119
 - 模拟数据, 73
- 显示主轴载荷, 107
- 显示过滤器, 93
- 显示服务器名称, 21
- 显示版本, 143
- 结果列表
 - 类型, 267
 - 颜色, 260
- 语言
 - 安装, 87
 - 选择, 87
- 说明文件, 278, 287

十划

- 宽屏显示, 39
- 桌面切换, 137
- 通信故障记录, 152
- 通道菜单, 54
- 配置, 189
- 配置数据, 152

十一划

- 旋转坐标系, 77
- 隐藏轴, 105

十二划

- 循环
 - 匹配, 76
 - 储存在 DRAM 中, 102
- 确认机制, 118
- 程序段查找, 122
- 程序测试, 122
- 编程 WZV 画面, 251
- 装载刀位, 176, 256

十三划

- 数据传输
 - TCU, 71
 - 由 PLC, 113
 - 激活, 114
- 数据维护 (DH), 16
- 输入监控, 78
- 键盘特性, 12

十四划

截图, (参见截屏)

模拟

 匹配刀具, 76

 匹配机床数据, 76

 匹配循环, 76

 匹配装备数据, 75

 引导起动, 79

 坐标系, 77

 数据, 73

 镜像刀具, 76

模拟的数据匹配, 75

模板 (Templates), 43

精确零点偏移, 98

十六划

操作单元管理 (选项), 51

操作核心 TCU, 71

磨削刀具, 269

磨削数据, 刀具专用, 269

SIEMENS

SINUMERIK 840Di sl/840D sl
SINUMERIK 810D/840D

补充操作界面

调试手册

功能范畴	1
编程	2
编程支持	3
设计热键和 PLC 键	4
操作区“Custom (定制)”	5
设计环境	6
附录	A
缩略语列表	B

安全技术提示

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

带有警告三角，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。

小心

不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。

注意

表示如果不注意相应的提示，可能会出现不希望的结果或状态。

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

仅允许安装和驱动与本文件相关的附属设备或系统。设备或系统的调试和运行仅允许由**合格的专业人员**进行。本文件安全技术提示中的合格专业人员是指根据安全技术标准具有从事进行设备、系统和电路的运行，接地和标识资格的人员。

按规定使用

请注意下列说明：



警告

设备仅允许用在目录和技术说明中规定的使用情况下，并且仅允许使用西门子股份有限公司推荐的或指定的其他制造商生产的设备和部件。设备的正常和安全运行必须依赖于恰当的运输，合适的存储、安放和安装以及小心的操作和维修。

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。标签中的其他符号可能是一些其他商标，这是出于保护所有者权利的目地由第三方使用而特别标示的。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

目录

1	功能范畴	7
2	编程	9
2.1	第一步	9
2.1.1	设计基础	9
2.1.2	设计文件和登入文件	10
2.1.3	设计文件的结构	11
2.1.4	错误处理 (日志)	12
2.2	对话框的结构和组成单元	14
2.2.1	定义对话框	14
2.2.2	举例：调用对话框	16
2.2.3	对话框属性	18
2.2.4	对话框单元	21
2.2.5	多列对话框	23
2.2.6	统一对话框显示图	23
2.2.7	使用图/图形	25
2.3	变量	27
2.3.1	变量属性	27
2.3.2	编程变量	29
2.3.3	变量类型 [1] 的详细说明	31
2.3.4	转换栏 [2] 的详细说明	35
2.3.5	预设值 [3] 的详细说明	36
2.3.6	短文本 [8]、输入/输出栏 [9] 位置的详细说明	37
2.3.7	帮助 [11] 的详细说明 (仅适用于 HMI 高级)	37
2.3.8	Anwendungsbeispiele (应用举例)	38
2.3.9	举例 1：分配变量类型、文本、帮助、颜色	40
2.3.10	举例 2：定义变量类型，极限值，属性，短文本位置	41
2.3.11	举例 3：定义变量类型、预设、系统或者用户变量、输入/输出栏位置	42
2.3.12	转换栏、帮助调用和画面显示的举例	43
2.3.13	使用字符串	45
2.3.14	变量 CURPOS	46
2.3.15	变量 CURVER	47
2.3.16	变量 ENTRY	48
2.3.17	变量 ERR	49
2.3.18	变量 FILE_ERR	50
2.3.19	变量 FOC	51
2.3.20	变量 S_CHAN	51
2.4	综合对话框单元	52
2.4.1	数组	52
2.4.2	存取数组单元的值	53
2.4.3	举例：存取数组单元	54
2.4.4	查询数组单元的状态	56
2.4.5	表格栅格 (Grid)	57
2.4.6	定义表格栅格	58
2.4.7	定义列	59
2.4.8	表格栅格中的聚焦控制	61

2.4.9	举例：定义列.....	62
2.4.10	举例：载入不同的表格栅格.....	64
2.5	软键栏.....	66
2.5.1	软键描述.....	66
2.5.2	定义软键栏.....	67
2.5.3	运行时改变软键属性.....	69
2.5.4	定义登入软键.....	71
2.5.5	登入软键的功能.....	71
2.6	方法.....	73
2.6.1	CHANGE.....	73
2.6.2	FOCUS.....	75
2.6.3	LOAD GRID.....	76
2.6.4	LOAD.....	76
2.6.5	UNLOAD.....	77
2.6.6	OUTPUT.....	77
2.6.7	PRESS.....	78
2.6.8	举例：管理带 OUTPUT 块的版本.....	79
2.7	功能.....	81
2.7.1	主动程序 (AP).....	81
2.7.2	定义块(/B).....	83
2.7.3	子程序调用(CALL).....	84
2.7.4	检查变量 (CVAR).....	84
2.7.5	复制程序 (CP).....	86
2.7.6	对话框行(DLGL).....	86
2.7.7	删除程序 (DP).....	87
2.7.8	评估(EVAL).....	88
2.7.9	执行(EXE).....	89
2.7.10	存在程序 (EP).....	90
2.7.11	退出对话框(EXIT).....	91
2.7.12	退出装载软键(EXITLS).....	93
2.7.13	生成代码(GC).....	93
2.7.14	装载数组(LA).....	96
2.7.15	装载块 (LB).....	97
2.7.16	装载屏幕窗口 (LM).....	98
2.7.17	装载软键(LS).....	99
2.7.18	被动程序 (PP).....	100
2.7.19	读取 NC PLC (RNP), 写入 NC PLC (WNP).....	101
2.7.20	多次读取 NC PLC (MRNP).....	102
2.7.21	刷新.....	104
2.7.22	寄存器(REG).....	105
2.7.23	RETURN.....	107
2.7.24	反编译.....	107
2.7.25	向前/后查找(SF, SB).....	109
2.7.26	选择程序 (SP).....	110
2.7.27	字符串功能.....	111
2.7.28	PI 服务.....	115
2.7.29	外部功能 (仅 HMI 高级).....	116
2.7.30	编程举例.....	118
2.8	运算符.....	121
2.8.1	数学运算符.....	121
2.8.2	位运算符.....	123

3	编程支持	125
3.1	编程支持提供什么?	125
3.2	循环辅助	126
3.3	从 NC 程序激活对话框	128
3.3.1	指令 "MMC" 的结构	129
3.3.2	MMC 指令举例	130
3.3.3	举例 1: 无确认情况下的 MMC 指令	132
3.3.4	举例 2: 停留时间和可选的文本变量	133
3.3.5	举例 3: MMC 指令, 在同步确认模式下	135
3.3.6	举例 4: 输入/输出栏定位	136
3.3.7	举例 5: 在对话框画面中显示图形	138
3.3.8	举例 6: 显示 BTSS 变量	139
3.3.9	举例 7: 用软键进行的异步确认模式	140
4	设计热键和 PLC 键	143
4.1	引言	143
4.1.1	OP 的热键	144
4.1.2	供货状态下键的功能	145
4.2	设计	146
4.2.1	设计概述	146
4.2.2	在文件 "IF.INI" 中进行设计	148
4.2.3	编程热键事件	151
4.2.4	扩展及特殊情况	153
4.2.5	PLC 按键的扩展	154
4.3	PLC 接口	155
4.3.1	接口结构	155
4.3.2	PLC 画面选择说明	156
4.3.3	设计对话框选择	159
4.4	选择对话框/软键栏	160
4.4.1	分配 INI 文件到操作区域	160
4.4.2	设计功能“补充操作界面”	161
4.5	可选状态的列表	162
4.5.1	HMI 高级上的可选状态	162
4.5.2	HMI 内置 si 上可选的状态	164
4.5.3	NCU 上 ShopMill 的可选状态	165
4.5.4	NCU 上 ShopTurn 的可选状态	168
5	操作区“Custom (定制)”	173
5.1	供货状态和应用	173
5.2	激活操作区	174
5.3	定义开始对话框	175
6	设计环境	177
6.1	供货范围	177
6.2	创建设计文件	178
6.2.1	使用文件 COMMON.COM	178
6.2.2	文件 COMMON.COM 的结构	179
6.2.3	设计登入软键	181
6.2.4	和语言相关的文本	183

6.3	设计文件的存档结构	184
6.3.1	HMI 内置 sl	184
6.3.2	HMI 高级	185
6.4	HMI 系统使用共同的硬件平台时的查找方案	186
6.4.1	查找方案的原理	186
6.4.2	COMMON.COM 的查找方案	189
6.4.3	图形的查找方案	189
A	附录	191
A.1	登入软键表	191
A.2	颜色表	194
A.3	可用的系统变量列表	196
A.4	PI 服务列表	206
B	缩略语列表	209
B.1	缩略语	209
	索引	索引-219

功能范畴

概述

“补充操作界面”通过编译器和包含操作界面说明的设计文件实现。通过 ASCII 文件配置“补充操作界面”：该设计文件包含了关于操作界面的说明。创建文件所需的句法参见下列章节。

通过“补充操作界面”方法可以设计显示机床制造商专用/最终用户专用的扩展功能的操作界面，或者仅设计自定义的对话框。也可以改善或者更换由西门子或者机床制造商设计的操作界面。

编译器在 HMI 内置 sl、ShopMill/ShopTurn 以及 HMI 高级上的使用方法一致。

例如，可通过新建的操作界面编辑零件程序。或直接在控制系统上创建对话框。

前提条件

创建图形/图还需要使用图形程序。在 HMI 内置 sl 上需要使用应用程序磁盘和 Paint Shop Pro (www.jasc.com)。

附随提供的工具箱中包含了设计新建对话框的实例，可以在此实例的基础上创建自定义的对话框。

使用

使用“补充操作界面”可以实现以下功能：

1. 显示对话框并提供：
 - 软键
 - 变量，表格
 - 文本和帮助文本
 - 图形和帮助画面
2. 通过以下方法调用对话框：
 - 按下（登入）软键
 - 选择 PLC
3. 动态重组对话框
 - 修改、删除软键
 - 定义并设计变量栏
 - 显示、更换、删除显示文本（和语言相关或无关）
 - 显示、更换、删除图形

4. 在进行以下操作时触发动作：
 - 显示对话框
 - 输入数值 (变量)
 - 按下软键
 - 关闭对话框
5. 对话框间的数据交换
6. 变量
 - 读取 (NC 变量、PLC 变量、用户变量)
 - 写入 (NC 变量、PLC 变量、用户变量)
 - 和数学、比较或者逻辑运算符相连
7. 执行下列功能：
 - 子程序
 - 文件功能
 - PI 服务
 - 外部功能 (HMI 高级)
8. 根据用户组考虑保护等级

边界条件

必须符合以下边界条件：

- 只允许在 HMI 操作区内交换对话框。
- 在 HMI 高级上，请求后才初始化用户、设定数据和机床数据。
- 用户变量不允许与系统变量或者 PLC 变量有相同的名称。
- 由 PLC 激活的对话框在 HMI 高级上构成自定义的操作区 (和测量循环图类似)。

注意

通过“补充操作界面”可以建立编程支持 (参见同名章节) 以及建立西门子循环的操作界面。因此机床制造商或者最终用户可以在必要时并在上述说明的范围内修改编程支持和操作界面。

另见

章节“设计环境”中说明的设计文件。

编程

2.1 第一步

2.1.1 设计基础

设计文件

新操作界面的说明存储在设计文件中。这些文件自动编译并显示屏幕上的结果。在供货时并不提供设计文件，因此必须首先创建此文件。

创建设计文件时使用一个 ASCII 编辑器（例如记事本或者 HMI 编辑器）。

操作树的原理

多个相连的对话框构成了一个操作树。

如果能从一个对话框切换入另一个对话框，则表示这两个对话框间存在联系。通过此对话框内重新定义的水平或者垂直软键可以返回上级对话框或者进入任意一个对话框。

在每个登入软键下都可以生成一个操作树：

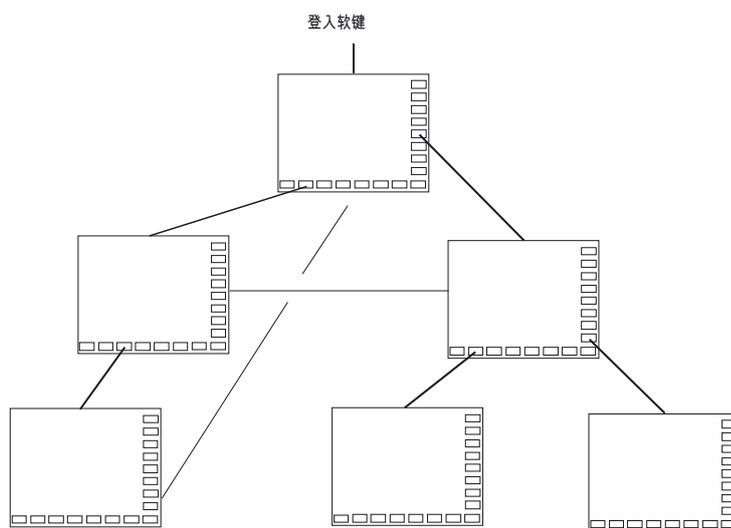


图 2-1 操作树

登入软键

在一个规定的设计文件中定义作为自身操作过程出发点的一个或者多个软键（登入软键）。

软键的定义决定执行下一步动作的自定义对话框或者其他软键栏。

按下登入软键则载入所属的对话框。同时，属于对话框的相应软键激活。

如果没有设计特定位置，则在标准位置上给出变量。

返回到标准应用程序

可以退出新定义的操作界面并返回到标准应用程序。

通过<RECALL>（回调）键可以退出新定义的操作界面，如果这个按键还未被设计用于其它用途。

PLC 调用自定义对话框

除了软键，也可以由 PLC 选择对话框。PLC 和 HMI 间存在用于信号交换的接口（在 DB19 中）。

参见

接口结构 (页 155)

设计登入软键 (页 181)

2.1.2 设计文件和登入文件

概述

每个应用程序都拥有固定（NCU 上的 HMI 内置 sl、ShopMill 和 ShopTurn）或者预设的登入软键，在该软键下新建的对话框形成树形图。在 HMI 高级上可以设计其他登入软键。

其他文件：

调用设计文件中的“载入屏幕窗口”（LM）或者“载入软键栏”（LS）可以重新命名已调用对象所在的文件。

采用这种方式可以划分设计，例如：一个自定义的设计文件同一操作级的所有功能。

创建设计文件为 ASCII 文件

对话框可以包含下列单元：

- 带下列文本的输入/输出栏 (变量)
 - 短文本
 - 图形文本
 - 单位文本
- 图形
- 表
- 软键栏

设计文本的查找顺序

- **NCU 上的 HMI 内置 sl**

在 NCU 上的 HMI 内置 sl 以及 ShopMill/ShopTurn 上，存取时会在 CF 卡相应的目录中查找设计文件。

在 HMI 内置 sl 中的标准循环目录/用户循环目录下的文件 COMMON.COM 中可以设定，是否在每次存取时都重新查找设计文件（仅当在控制系统上直接创建对话框时比较重要），或者是否再次使用已经找到的或者保存在中间存储器中的设计文件（和标准操作情况一致）。

- **HMI 高级**

在 HMI 高级中，存取时首先在用户循环目录，然后在制造商循环目录，最后在标准循环目录中查找设计文件。

参见

查找方案的原理 (页 186)

2.1.3 设计文件的结构

概述

设计文件由以下单元组成：

- 登入软键说明
- 对话框定义
- 变量定义
- 块说明
- 软键栏定义

举例：

```
//S (START) ; 登入软键定义 ( 可选 )
....
//END
//M (.....) ; 对话框定义
DEF ..... ; 变量定义
LOAD ; 块说明
...
END_LOAD
UNLOAD
...
END_UNLOAD
ACTIVATE
...
END_ACTIVATE
...
//END
//S (....) ; 软键栏定义
//END
```

2.1.4 错误处理 (日志)

概述

日志是记录解释句法时出现的错误消息的文件(Error.com)。文件必须由操作员自己在注释目录中设立 (HMI 高级)。

举例：

```
DEF VAR1 = (R)
DEF VAR2 = (R)
LOAD
VAR1 = VAR2 + 1 ; 日志中的错误消息，因为 VAR2 没有数值
```

句法

当定义登入软键并且设计了带开始和结束标识的对话框和定义行后，才开始解释句法。

```
//S(Start)
HS6=("第 1 屏幕窗口")
PRESS(HS6)
    LM("屏幕窗口 1")
END_PRESS
//END

//M(屏幕窗口 1)
    DEF Var1=(R)
//END
```

ERROR.COM 的内容

如果“补充操作界面”解释定义文件时出现错误，则该错误保存在 ASCII 文件 ERROR.COM 中。

文件包含以下信息：

- 在执行何种动作时出现错误
- 第一个错误字符的行号和列号
- 设计文件中所有的错误行

如果借助 PC 测试环境创建对话框，则该文件位于环境变量 RAMDISK 参考的目录中 (HMI 内置 sl)。

仅当解释设计文件时确实出现错误，才创建文件 ERROR.COM。

文件 ERROR.COM 的保存路径：

- HMI 内置 sl：CF 卡上的目录 /tmp/hmiemb
- HMI 高级：目录 \DH\COM.DIR\ 中

每次重新启动 HMI 内置 sl 和 HMI 高级时删除该文件。

显示文件 ERROR.COM

HMI 高级：

- 在操作区“通讯”或者“开机调试”中调用编辑器

HMI 内置 sl：

- 进入操作区“开机调试”→“HMI”→“编辑器”→“临时驱动器”（垂直软键栏的第 4 个软键）。虽然使用制造商密码不能显示该软键，但软键仍作出反应（使用系统密码可显示软键）。
- 选中 ERROR.COM。
- 按下 <INPUT> 键。
- 按下软键“文件功能”可以通过设置的 Windows 网络驱动器将文件复制到 PC 上。

2.2 对话框的结构和组成单元

2.2.1 定义对话框

定义

对话框是操作界面上的一个组成部分，操作界面包含标题行、对话框单元和/或图形、显示消息的输出行以及 8 个水平软键和 8 个垂直软键。

对话框单元包含：

- 变量
 - 极限值
 - 变量预设值
- 帮助画面
- 文本
- 属性
- 系统或者用户变量
- 短文本的位置
- 输入/输出栏的位置
- 颜色
- 帮助(仅 HMI 高级)

对话框的属性：

- 标题
- 图形
- 尺寸
- 系统或者用户变量
- 图形的位置
- 属性

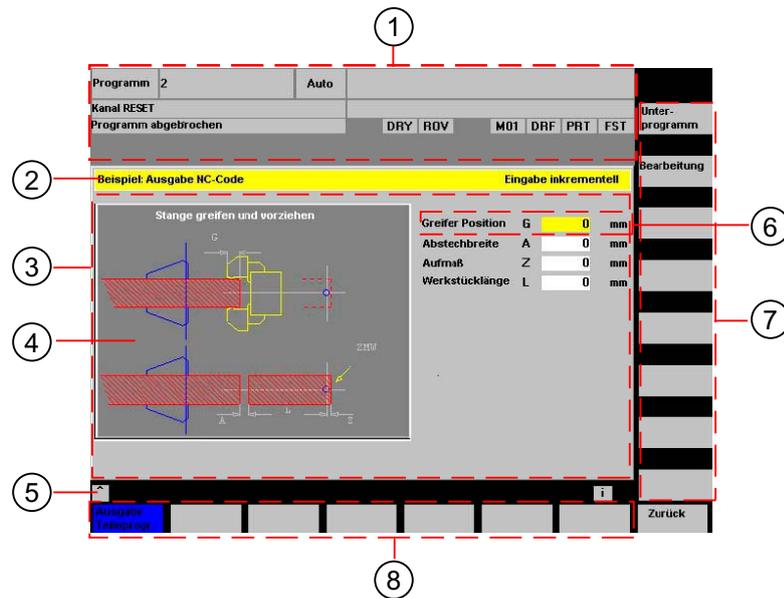


图 2-2 对话框结构

- 1 机床状态显示 (“标题”)
- 2 对话框的标题栏，包含标题和长文本
- 3 对话框
- 4 图形
- 5 显示消息的输出栏
- 6 对话框单元
- 7 8 个垂直软键
- 8 8 个水平软键

概述

基本上对话框说明（说明块）的结构如下：

说明块	注释	参考章节
//M...	; 对话框的开始标识	
DEF Var1=... ...	; 变量	参见章节“变量”
HS1=(...) ...	; 软键	参见章节“软键栏”
PRESS(HS1) LM... END_PRESS	; 方法的开始标识 ; 动作 ; 方法的结束标识	参见章节“方法”
//END	; 对话框的结束标识	

在对话框的说明块中，首先定义在对话框的对话框单元中显示的不同变量，然后定义水平和垂直软键。然后在方法中设计不同的动作。

2.2.2 举例：调用对话框

编程

进入操作区“参数”，按下登入软键“举例”可以调用一个新的对话框。

```
//S(Start)
HS7=("举例", ac7, sel)

PRESS(HS7)
  LM("屏幕窗口 1")
END_PRESS

//END
//M(屏幕窗口 1/"循环")
HS1=""
HS2=""
HS3=""
HS4=""
HS5=""
HS6=""
HS7=""
HS8=""
VS1=""
VS2=""
VS3=""
VS4=""
VS5=""
VS6=""
VS7=""
VS8=""
... ; 方法
//END
```

结果

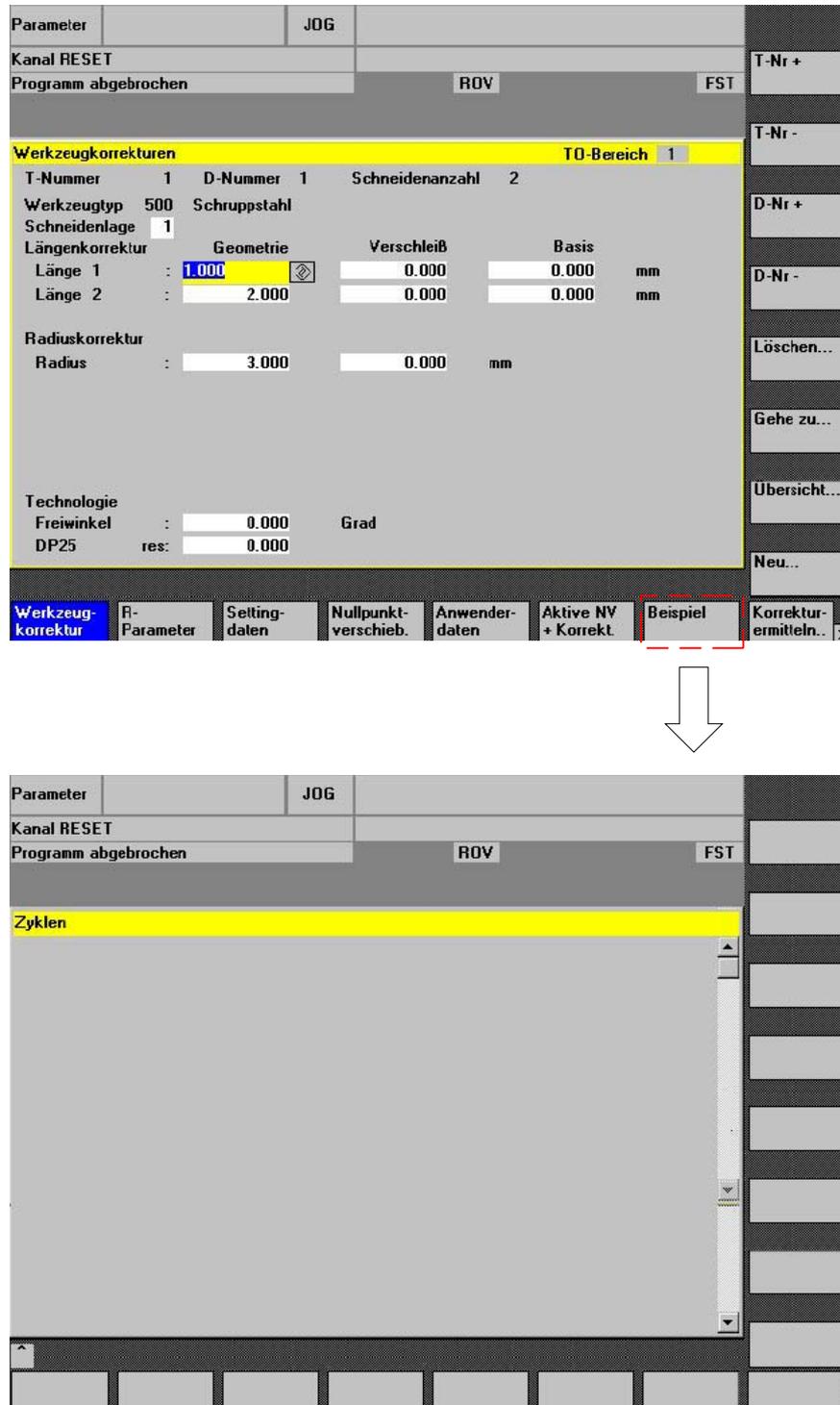


图 2-3 举例：按下登入软键“举例”调用对话框“循环”

2.2.3 对话框属性

说明

使用对话框的开始标识可以同时定义对话框的属性。

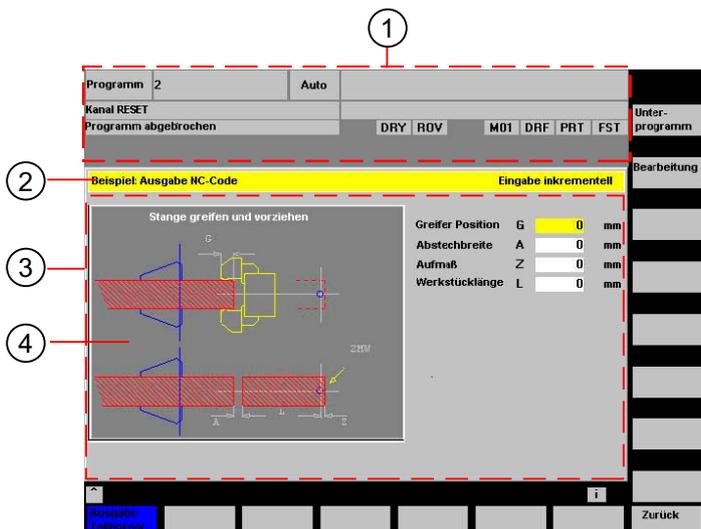


图 2-4 对话框属性

- 1 机床状态显示 (“标题”)
- 2 对话框的标题栏，包含标题和长文本
- 3 对话框
- 4 图形

编程

句法：**//M(名称/[标题]/[图形]/[尺寸]/[系统或用户变量]/[图形位置]/[属性])**

说明：定义对话框

参数：

名称	对话框的名称
标题	对话框的文本标题或者从和语言相关的文本文件中调用文本 (例如：\$85011)
图形	图形文件，路径在双引号内
尺寸	对话框的位置和大小，单位像素 (和左/右边缘的间距、宽度、高度)，以屏幕的左上角为基准。数据值用逗号相隔。
系统或者用户变量	指定当前光标位置的系统和用户变量。可以通过系统或者用户变量将光标位置传送给 NC 或 PLC。第一个变量索引为 1。变量的顺序和变量设计顺序一致。

图形的位置	图形的位置，单位像素（和左/上边缘的间距），以对话框的左上角为基准。和上边缘的最低间距为 18 像素。数据值用逗号相隔。
属性	给定的属性用逗号相隔。 允许的属性有：
CMx	Column Mode: 列对齐 CM0 预设置：每行单独分列。 CM1 以包含最多列的行为标准分列。
CB	CHANGE 块: 打开对话框时的属性： 定义变量时规定的 cb 属性，优先于定义对话框时的总定义 CB0 预设置：在打开对话框时处理所有 CHANGE 部分。 CB1 只有在附属的值改变后才处理 CHANGE 部分。
系统	在运行期间可读取“系统”的属性： 0: HMI_内置 1: HMI_高级

存取对话框属性

在方法（例如：PRESS 块）的范围内可以读取和写入对话框的以下属性：

- Hd = 标题
- Hlp = 帮助画面
- Var = 系统或者用户变量

举例

```
//S(Start)
HS7=("举例", se1, ac7)

PRESS(HS7)
    LM("屏幕窗口 1")
END_PRESS

//END
//M(屏幕窗口 1/"举例 2 : 显示图形"MCP.BMP")
HS1=("新%n标题")
HS2=(" ")
HS3=(" ")
HS4=(" ")
HS5=(" ")
HS6=(" ")
HS7=(" ")
HS8=(" ")
VS1=(" ")
VS2=(" ")
VS3=(" ")
VS4=(" ")
VS5=(" ")
VS6=(" ")
VS7=(" ")
VS8=(" ")

PRESS(HS1)
    Hd= "新标题"
END_PRESS
...
//END
```

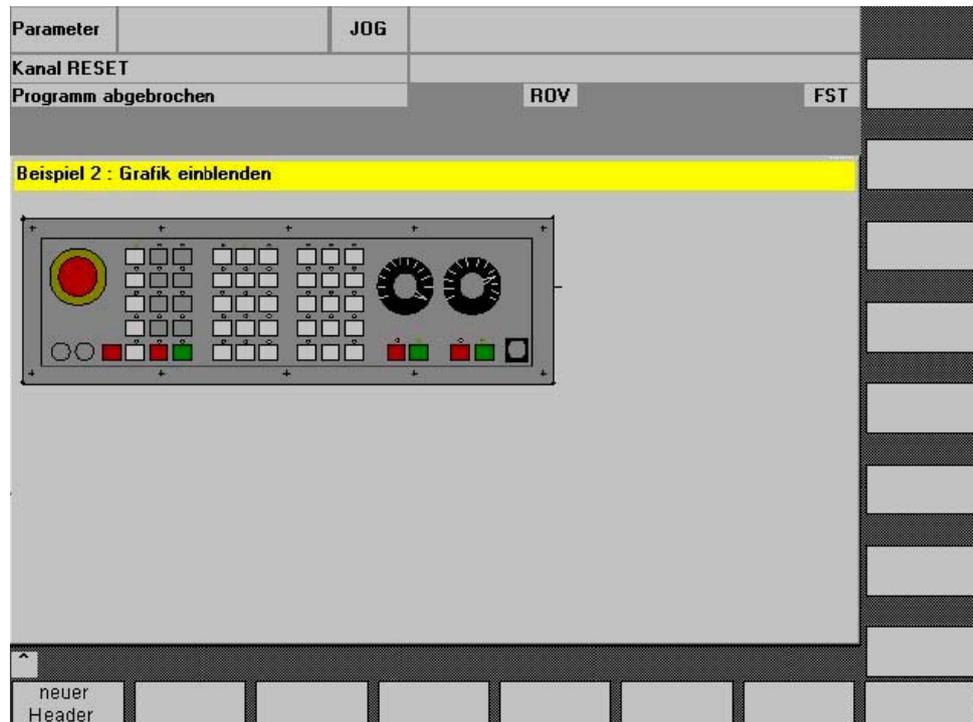


图 2-5 “举例 2：显示图形”

参见

使用图/图形 (页 25)

和语言相关的文本 (页 183)

2.2.4 对话框单元

对话框单元

对话框单元是变量的可见部分，即短文本、图形文本、输入/输出栏和单位文本。对话框单元位于对话框主体的行中。每行可以定义一个或者多个对话框单元。

变量属性

所有变量仅在激活的对话框中有效。通过定义变量指定其属性。在方法（例如：PRESS 块）的范围内可以存取对话框属性值。

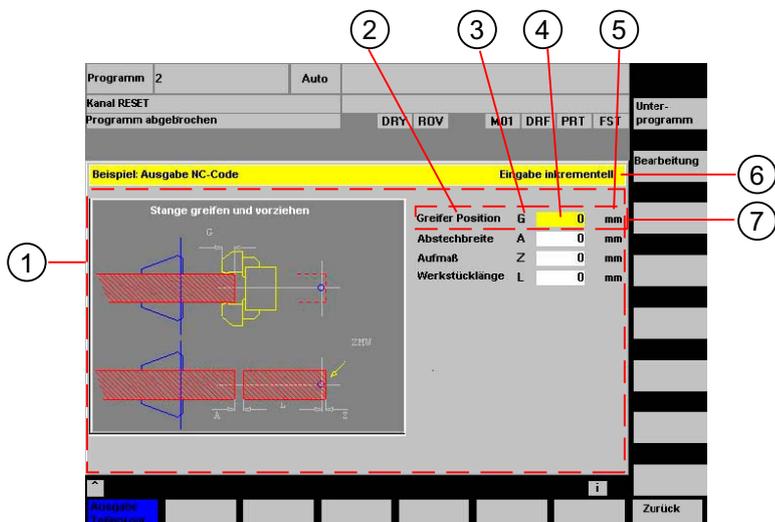


图 2-6 对话框单元

- 1 对话框主体
- 2 短文本
- 3 图形文本
- 4 输入/输出栏
- 5 单位文本
- 6 对话框的标题栏，包含标题和长文本
- 7 对话框单元

编程一览

在圆括号中通过逗号隔开各个参数。

DEF <i>Bezeichner</i> =	Bezeichner = 变量名称		
	变量类型	→	1
	/[极限值或者转换栏或者表格名称]	→	2
	/[预设值]	→	3
	/[文本 (长文本, 短文本 图, 图形文本, 单位文本)]	→	4
	/[属性]	→	5
	/[帮助画面]	→	6
	/[系统或者用户变量]	→	7
	/[短文本位置]	→	8
	/[输入/输出栏位置 (左、上、宽度、高度)]	→	9
	/[颜色]	→	10
	/[帮助] (仅 HMI 高级)	→	11

参见

多列对话框 (页 23)

变量属性 (页 27)

2.2.5 多列对话框

概述

在对话框中，一行可以显示多个变量。

在这种情况下，设计文件中的所有变量都定义在一个定义行内。

```
DEF VAR11 = (S///"Var11"), VAR12 = (I///"Var12")
```

为了可以更清除地显示设计文件中的各个变量，在每个变量定义和其紧随的逗号后可以换行。

关键字“DEF”总是表示新的一行的开始：

```
DEF Tnr1=(I//1/"", "T ", ""/wr1///, ,10/20, ,50),
  TOP1=(I///, "Typ="/WR2//"$TC_DP1[1,1]"/80, ,30/120, ,50),
  TOP2=(R3///, "L1="/WR2//"$TC_DP3[1,1]"/170, ,30/210, ,70),
  TOP3=(R3///, "L2="/WR2//"$TC_DP4[1,1]"/280, ,30/320, ,70),
  TOP4=(R3///, "L3="/WR2//"$TC_DP5[1,1]"/390, ,30/420, ,70)
DEF Tnr2=(I//2/"", "T ", ""/wr1///, ,10/20, ,50),
  TOP21=(I///, "Typ="/WR2//"$TC_DP1[2,1]"/80, ,30/120, ,50),
  TOP22=(R3///, "L1="/WR2//"$TC_DP3[2,1]"/170, ,30/210, ,70),
  TOP23=(R3///, "L2="/WR2//"$TC_DP4[2,1]"/280, ,30/320, ,70),
  TOP24=(R3///, "L3="/WR2//"$TC_DP5[2,1]"/390, ,30/420, ,70)
...
```

在设计多列对话框时须注意所使用的硬件，例如：HMI 内置 sl 最多允许 10 列和 60 个 DEF 指令。

2.2.6 统一对话框显示图

边界条件

如果在操作面板同时安装了 HMI 高级和 ShopMill 或者 ShopTurn，则两个系统有不同字体类型。

HMI 高级为 Proportional Font (均衡字体)，而 JobShop 产品以及 HMI 内置 sl 使用 Fixed Font (固定字体)。

如果在 HMI 内置 sl 和 HMI

高级上使用“补充操作界面”，则定义相同的对话框有不同的显示画面。

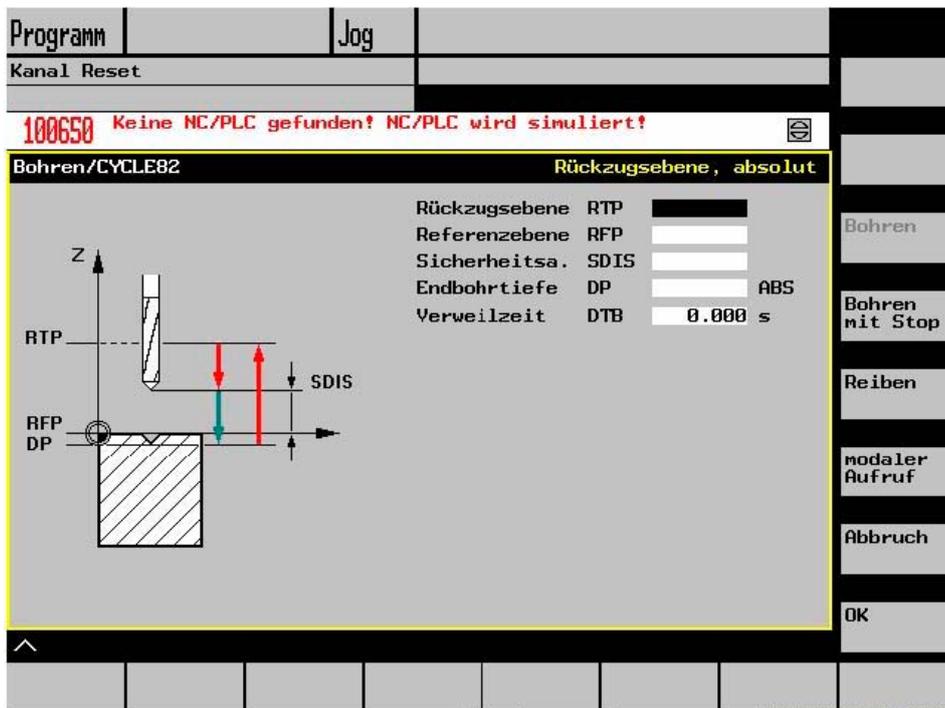


图 2-7 HMI 内置 si 的显示画面

下图为图形内容相同的 HMI 高级下的显示画面。

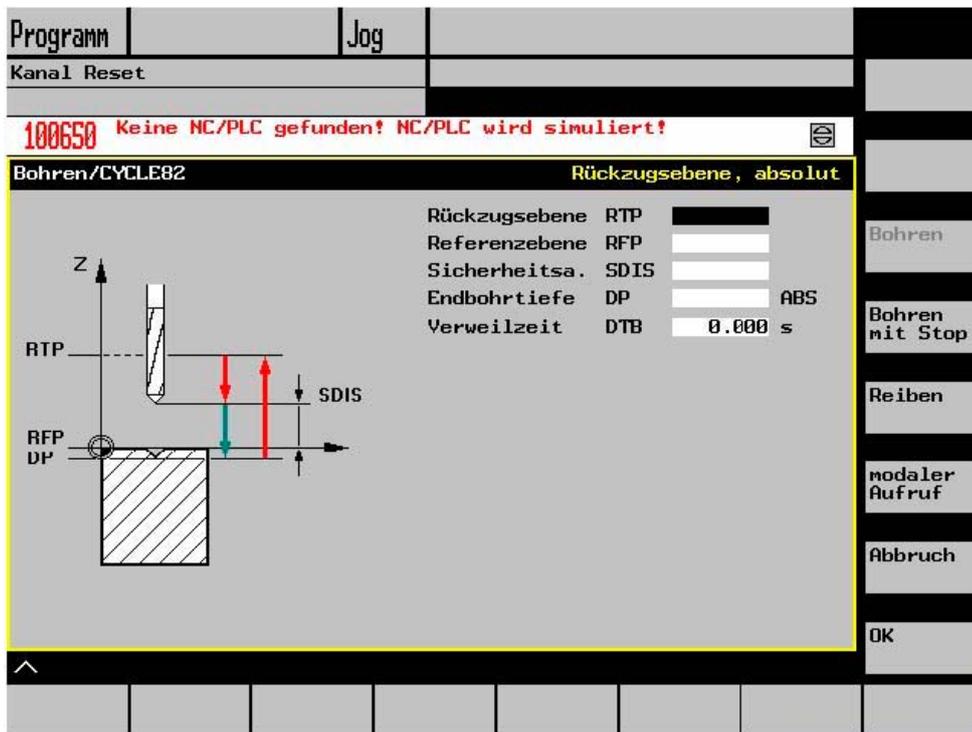


图 2-8 HMI 高级的显示画面

设置相同的显示画面

为了统一 HMI 高级和 HMI 内置 sl 的显示画面，可以在配置文件 WIZARD.INI 中将字体改为固定字体。

```
[FONT]
FixedFont=1
```

当前语言的文本为固定字体。仅软键标签文本除外。与 HMI 内置 sl 相反，HMI 高级的输入栏具有表示输入模式的象形图。因此，该栏比 HMI 内置 sl 中的栏略小。

如上图所示，全部单元，如短文本、图形文本、输入栏和单位文本在相同的信息量情况下长度也大致相当。

然而如果必须定位某个栏，则可以在一定条件下通过待显示文本的长度得出不同 HMI 类型下不同的文本栏位置。

为了使两个系统使用同一个设计文件，则应设置对话框属性“属性”。在对话框属性“系统”中查询运行环境。从而，两个目标系统可以使用一个设计文件。

2.2.7 使用图/图形

使用图形

分为：

- 使用图/图形
- 帮助画面，例如：图示说明各个变量并突出显示在图形区中。
- 可以设计其它帮助画面替代可任意定位的短文本或者输入/输出栏。

最大尺寸	系统
560* 326 像素	HMI 高级/ HMI 内置 sl
688* 376 像素	PCU 50, 带 OP012

HMI 内置 sl 的图形创建

创建图形，例如：用 MS Paint。

HMI 内置 sl 可以处理转换图形和 BMP 文件。通常能够使用转换程序 BMP2BIN 生成的图形。对于 BMP 文件，颜色在线转换。为此，包含颜色表的文件(syscol.col, sysbw.col, ...), 必须添加段 [BMP]。该文件总是分配一个 BMP 像素的颜色给 HMI 内置 sl 颜色。

存放位置

HMI 内置 sl 自动查找所连显示器的分辨率，并首先在所属的分辨率目录中查找所需文件。如果在那里没有找到，就在下一个更小的分辨率目录中查找，直至目录 ico640 - 如果之前没有找到。

步骤

插入位图的步骤（用户图形）：

1. HMI 内置 sl 上的插入位图
2. 使用 Microsoft Paint 版本 4.0 或者更高版本创建 BMP 文件
3. 通过随附的 arj.exe（版本 2.41）将 BMP 文件压缩在文件夹 CUS.ARJ 中或者将每个单独的 BMP 文件压缩为一个独立的文件夹，使用 BMP 文件的文件名和扩展名为“.BM_”

例如：

- 每个文件夹包含多个文件：

```
arj a cus.arj my_file1.bmp my_file2.bmp my_file3.bmp my_file4.bmp
```

- 每个文件夹包含一个文件：

```
arj a my_file1.bm_ my_file1.bmp
```

限制

压缩是可选的。然而，必须注意 CF 卡上的容量至少为 10 MB。

可以分配给位图任意一个软键，软键文本以 2 个反斜线符号开始。反斜线符号后的文本则为包含位图的文件名称。

例如在 ALUC.TXT 85000 0 0 “\\mybitmap.bmp”

注意

图形颜色在 HMI 内置 sl 上可能与画笔程序中的不同。

在 HMI 高级上创建图形

只要能够以此建立所述格式的一种，就可自由选择图像程序。图像 / 图形和帮助画面可以有如下格式：

- 位图 (BMP)
- Windows Meta 文件 (WMF)
- 图标文件 (ICO)

“图中图”

在背景（帮助画面）中可以显示其它的图形，但也取决于变量值。
例如对于软键，可以分配一个显示区给一个图形文件。

举例：

- 显示带有图形的短文本栏：

`DEF VAR1=(S///,"\\图 1.bmp" ////160,40,50,50) ;类型 S 无意义`

- 显示带有图像的转换栏，此时可以通过一个 PLC 标记字节选择图像。

`DEF VAR1=(IDB/*1="\\图 1.bmp",2="\\图 2.bmp"//,$85000/wr1//"MB[0]"//160,40,50,50)`

作为第四个参数在此位置给定显示图形的高度（输入/输出栏）。
也可以在该栏（短文本，输入/输出栏）中给定位图。

参见

图形的查找方案 (页 189)

颜色表 (页 194)

2.3 变量

2.3.1 变量属性

变量值

变量的重要属性即变量值。

可以通过如下方式分配值：

- 定义变量时预设
- 分配系统或者用户变量
- 采取方法

编程

句法：	名称 val = 变量值 名称 = 变量值
说明：	变量值 val (value)
参数：	名称： 变量名称 变量值 变量值
举例：	<code>VAR3 = VAR4 + SIN(VAR5)</code> <code>VAR3.VAL = VAR4 + SIN(VAR5)</code>

变量状态

通过变量状态的属性可以在运行时查询，变量是否包含有一个有效值。该属性可通过值 FALSE = 0 读写。

编程

```

句法：          名称vld
说明：          变量状态 vld (validation)
参数：          名称：          变量名称
                  查询结果可能是：
                  FALSE = 无效值
                  TRUE = 有效值
举例：          IF VAR1.VLD == FALSE
                  VAR1 = 84
                  ENDIF

```

变量：更改属性

在符号属性的名称 = 值 中重新分配变量值。分析等号右边的表达式并分配变量或者变量属性。

举例：

名称ac = 存取级	(ac: access level)
名称al = 文本对齐	(al: alignment)
名称bc = 背景颜色	(bc: back color)
名称fc = 前景颜色	(fc: front color)
名称al = 字体大小	(fs: font size)
名称gt = 图形文本	(gt: graphic text)
名称hlp = 帮助画面	(hlp: help)
名称htx = 帮助文本	(htx: help text)
名称li = 极限值	(li: limit)
名称lt = 长文本	(lt: long text)
名称max = 最大极限值	(max: maximum)
名称min = 最小极限值	(min: minimum)
名称st = 短文本	(st: short text)
名称typ = 变量类型	(typ: type)
名称ut = 单位文本	(ut: unit text)
名称val = 变量值	(val: value)
名称var = 系统或者用户变量	(var: variable)
名称vld = 变量状态	(vld: validation)
名称wr = 输入模式	(wr: write)

2.3.2 编程变量

编程

在以下概述中简要说明了变量参数。详细的说明请参见后续章节。

参数	说明	
1 变量类型	必须规定变量类型。	
	R[x]:	REAL (小数点位数为正数 +)
	I:	INTEGER
	S[x]:	字符串 (字符串长度为正数 +)
	C:	字符 (单字符)
	B :	BOOL
	V:	VARIANT
2 极限值	最小极限值，最大极限值 预设置：空 极限值通过逗号隔开。类型 I、C 和 R 的极限值可以使用十进制的格式或者字符 "A"、"F" 定义。	
转换栏	输入/输出栏中带有预设输入项的列表：列表通过 * 开始，各输入项用逗号隔开。可以赋值输入项。 极限值的输入项视为转换栏的列表。如果只输入一个 *，则建立一个可变的转换栏。 预设置：无	
表格名称	NCK/PLC 相同类型值的表格名称，它可以通过通道模块编译地址。 表格名称通过一个前置的极限值或者转换栏的 % 符号进行区分。 表格名称可以通过逗号分隔，还可以跟随一个文件名，该文件名指定定义表格描述的文件。	
3 预设值	如果没有定义任何预设值并且没有分配系统或者用户变量给变量，则分配转换栏的第一个单元。如果没有定义转换栏，则不进行预设，即：变量处于状态“未计算”。 预设置：未预设	
4 文本	顺序已预先规定。也可以显示一个图形代替短文本。 预设置：空	
	长文本：	显示行的文本
	短文本：	对话框单元的名称
	图形文本：	文本参考图形名称。
	单位文本：	对话框单元的单位
5 属性	属性影响下列特性： <ul style="list-style-type: none"> • 输入模式 • 存取等级 • 短文本的文本对齐 • 字体大小 • 极限值 • 打开对话框时的属性和 CHANGE 块有关 属性通过逗号隔开，顺序任意 属性不适用于转换栏。每个组件都可以进行定义。	

参数	说明	
	输入模式	wr0: 输入/输出栏不可见，短文本可见， wr1: 读取（没有输入中心） wr2: 读取和写入（行以白色显示） wr3: wr1 带输入中心 wr4: 所有变量单元不可见，没有输入中心 wr5: 按下任何键立即保存输入的值（和 wr2 相反 - 该模式下，在退出栏或者按下返回键后才开始保存值）。 预设置：wr2
	存取等级	空：总是可以写入 ac0...ac7: 保护等级 如果存取等级未达到，行显示为灰色，标准设置：ac7
	短文本的文本对齐	al0: 左对齐 al1: 右对齐 al2: 中间对齐 预设置：al0
	字体大小	fs1: 标准字体大小（8 Pt） fs2: 双倍字体大小 预设置：fs1 确定行间距。标准字体大小为每对话框 16 行。 图形和单位文本只能为标准字体大小。
	极限值	通过极限值可以检查变量值是否在规定的最小极限值和最大极限值之内。 预设置：取决于规定的极限值 li0: 没有检查 li1: 检查最小极限值 li2: 检查最大极限值 li3: 检查最大极限值和最小极限值
	打开时的属性	定义变量时规定的 cb 属性，优先于定义对话框时的总定义 多个属性通过逗号隔开。
	cb0:	在打开对话框时处理定义变量的 CHANGE 块（预设置）。 多个属性通过逗号隔开。
	cb1:	只有在变量值改变时，才处理定义变量的 CHANGE 块。
6 帮助画面	帮助画面文件：	pdf 文件名称。 预设置：空
	帮助画面文件的名称在双引号中。 如果光标移至该变量，则自动显示画面（替代目前的图形）。	
7 系统或者用户变量	可向该变量分配 NC/PLC 上的系统或者用户变量。系统或者用户变量用双引号括起。 参考文献：列表手册，“系统变量列表” /PGA1/	
8 短文本位置	短文本位置（和左边缘/上边缘的间距、宽度） 位置数据的单位为像素，并且以对话框主体的左上角为基准。数据总是用逗号隔开。	
9 输入/输出栏位置	输入/输出栏位置（和左边缘/上边缘的间距、宽度、高度） 位置数据的单位为像素，并且以对话框主体的左上角为基准。数据总是用逗号隔开。 如果改变此位置，则短文本、图形文本和单位文本的位置也同时改变。	

参数	说明											
10 颜色	<p>前景颜色、背景颜色：颜色通过逗号隔开。 颜色的设置仅适用于输入/输出栏；对于其他的文本，无需定义颜色。 值范围：1...10 预设置： 前景颜色：黑色，背景颜色：白色 输入/输出栏的标准颜色取决于写入模式：</p>											
	<table border="0"> <tr> <td style="padding-right: 10px;">wr0:</td> <td>前景和背景颜色：窗口背景颜色</td> </tr> <tr> <td>wr1:</td> <td>文本颜色：黑色，窗口背景颜色</td> </tr> <tr> <td>wr2:</td> <td>文本颜色：黑色，背景颜色：白色</td> </tr> <tr> <td>wr3:</td> <td>同 wr0</td> </tr> <tr> <td>wr4:</td> <td>同 wr1</td> </tr> <tr> <td>wr5:</td> <td>同 wr2</td> </tr> </table>	wr0:	前景和背景颜色：窗口背景颜色	wr1:	文本颜色：黑色，窗口背景颜色	wr2:	文本颜色：黑色，背景颜色：白色	wr3:	同 wr0	wr4:	同 wr1	wr5:
wr0:	前景和背景颜色：窗口背景颜色											
wr1:	文本颜色：黑色，窗口背景颜色											
wr2:	文本颜色：黑色，背景颜色：白色											
wr3:	同 wr0											
wr4:	同 wr1											
wr5:	同 wr2											
11 帮助 (仅 HMI 高级)	<table border="0"> <tr> <td style="padding-right: 10px;">帮助文件：</td> <td>pdf 文件路径</td> </tr> <tr> <td>索引：</td> <td>包含帮助文本的文件的索引</td> </tr> <tr> <td>帮助文本：</td> <td>帮助文本文件中显示的帮助文本</td> </tr> </table>	帮助文件：	pdf 文件路径	索引：	包含帮助文本的文件的索引	帮助文本：	帮助文本文件中显示的帮助文本					
	帮助文件：	pdf 文件路径										
索引：	包含帮助文本的文件的索引											
帮助文本：	帮助文本文件中显示的帮助文本											
	<p>数据总是通过逗号隔开，顺序固定用双引号括起帮助文件和帮助文本。 PDF 文件必须位于目录 CUS.DIR\hlp.dir 或者 CST.DIR\hlp.dir 下。其它相应的 PDF 和文本文件的名称必须一致。在文本文件中 PDF 文件名必须大写。 每个对话框单元都可以设计多个链式帮助参考（帮助链），即依次调用帮助并在最后的帮助后重新显示第一个帮助。 如果调用第二个或其他帮助时仍是同一个文件/索引/帮助文本，则表示缺少数据。 当光标位于相应栏中并且按下信息键时，帮助显示。</p>											

2.3.3 变量类型 [1] 的详细说明

INTEGER 变量类型

“INTEGER”型变量可以使用下列扩展符号显示在输入/输出栏中并保存在存储器中。

- 扩展数据类型中的第 2 个字符

描述格式	
B	二进制
D	带有正负号的十进制
H	十六进制
没有数据	带有正负号的十进制

- 扩展数据类型中的第 3 个和第 4 个字符

存储器保存	
B	字节
W	字
D	双字
BU	不带正负号的字节
WU	不带正负号的字
DU	不带正负号的双字

INTEGER 型数据的字符顺序

1. “I” 作为 INTEGER 的基本标识符
2. 描述格式
3. 存储器保存
4. “U”，没有正负号

确定有效的 INTEGER 型：	
IB	整数变量 32 位二进制描述
IBD	整数变量 32 位二进制描述
IBW	整数变量 16 位二进制描述
IBB	整数变量 8 位二进制描述
I	整数变量 32 位十进制描述，带正负号
IDD	整数变量 32 位十进制描述，带正负号
IDW	整数变量 16 位十进制描述，带正负号
IDB	整数变量 8 位十进制描述，带正负号
IDDU	整数变量 32 位十进制描述，不带正负号
IDWU	整数变量 16 位十进制描述，不带正负号
IDBU	整数变量 8 位十进制描述，不带正负号
IH	整数变量 32 位十六进制描述
IHDU	整数变量 32 位十六进制描述
IHWU	整数变量 16 位十六进制描述
IHB	整数变量 8 位十六进制描述

VARIANT 型变量

变量类型 VARIANT 通过最后赋值的数据类型确定。它可以通过功能 ISNUM 或者 ISSTR 查询。VARIANT 类型主要适用于向 NC 码写入变量名或者数值。

编程

可以检查变量的数据类型：

句法：**ISNUM** (*VAR*)

参数：**VAR** 待检查其数据类型的变量名称。

查询结果可能是：

FALSE = 没有数值 (数据类型 = STRING)

TRUE = 数值 (数据类型 = REAL)

句法：**ISSTR** (*VAR*)

参数：**VAR** 待检查其数据类型的变量名称。

查询结果可能是：

FALSE = 数值 (数据类型 = REAL)

TRUE = 没有数值 (数据类型 = STRING)

举例：

```
IF ISNUM(VAR1) == TRUE
IF ISSTR(REG[4]+2) == TRUE
```

可以改变变量的显示模式：

- 对于 INTEGER 型变量可以改变显示方式。

B 二进制

D 带有正负号的十进制

H 十六进制

不带正负号

U 总是用于未标记

- 对于 REAL 型变量，只能更改小数点的位数。

不允许更改基本类型，若更改，则导致文件 ERROR.COM 出现错误信息。

举例：

```
Var1.typ = "IBW"
```

```
Var2.typ = "R3"
```

数值显示

计算可以以二进制、十进制、十六进制或者指数方式描述。
二进制、十六进制和指数描述方式的计算值都用单引号括起。

二进制	'B01110110'
十进制	123.45
十六进制	'HF1A9'
指数	'-1.23EX-3'

举例：

```
VAR1 = 'HF1A9'  
REG[0]= 'B01110110'  
DEF VAR7 = (R//'-1.23EX-3')
```

注意

在通过功能 GC
生成代码时计算值仅考虑十进制或者指数描述方式，而不使用二进制和十六进制描述方式。

2.3.4 转换栏 [2] 的详细说明

说明

通过转换栏扩展可以显示与 NC/PLC 变量有关的文本（在转换栏中的输入项）。
只能读取使用转换栏扩展的变量。

编程

句法：
DEF 名称 =(变量类型 /+ \$文本号码 | * 值="\图",[值="\图 2.bmp"][, ...]
 /[预设值]
 /[文本（长文本、短文本、图形文本、单位文本）]
 /[属性]
 /[帮助画面]
 /[系统或者用户变量]
 /[短文本位置]
 /[输入/输出栏位置（左、上、宽度、高度）]
 /[颜色]
 /[帮助])

说明：
 显示对话框时，在输入/输出栏中显示文本号码 \$85015 的内容。
 在系统变量 DB90.DBB5 输入预设值 15。如果系统变量 DB90.DBB5 中的值改变，在每次改变时显示的文本号码重新生成 \$ (85000 + <DB90.DBB5>)。

参数：	变量类型	系统或者用户变量中指定的变量类型
	文本号码	与语言相关的文本号码（基本），该号码作为基本号码使用
	系统或者用户变量	通过最终的文本号码（基本 + 补偿）形成的系统或者用户变量（补偿）。

举例：
 DEF VAR1=(IB/+ \$85000/15//// "DB90.DBB5")

可变转换栏

可以将可变转换栏分配给对话框单元，即：按下转换键则可以向变量分配 CHANGE 方法中定义的值。

为了标记一个可变的转换栏，在定义变量时在特性极限值或者转换栏中输入一个单独的星号 *。

举例： DEF VAR1=(S/*)

图形与转换栏有关

转换栏用替换的图像覆盖。如果标记字节值为 1，则显示“图 1.bmp”；如果标记字节值为 2，则显示“图 2.bmp”

```
DEF VAR1=(IDB/*1="\图 1.bmp",
           2="\图 2.bmp"//,$85000/wr1/"MB[0]"//160,40,50,50)
```

图形的大小和位置在“输入/输出栏位置 (左侧、上部、宽度、高度)”中规定。

2.3.5 预设值 [3] 的详细说明

概述

根据变量栏 (输入/输出栏或者转换栏) 是否分配了一个预设值，一个系统或者用户变量或者两者都分配，得到不同的变量状态。只有当变量分配了一个有效值时，转换才可行。

预设值生效

当...			则...
栏类型	预设值	系统或者用户变量	栏类型反应
输入/输出栏	是	是	在系统或者用户变量中写预设值
	否	是	使用系统或者用户变量作为预设值
	错误	是	未计算，系统或者用户变量未描述/未使用
	是	否	预设值
	否	否	未计算
	错误	否	未计算
	是	错误	未计算
	否	错误	未计算
	错误	错误	未计算
Toggle	是	是	在系统或者用户变量中写预设值
	否	是	使用系统或者用户变量作为预设值
	错误	是	未计算，系统或者用户变量未描述/未使用
	是	否	预设值
	否	否	预设值 =转换栏的第一个单元
	错误	否	未计算
	是	错误	未计算
	否	错误	未计算
	错误	错误	未计算

2.3.6 短文本 [8]、输入/输出栏 [9] 位置的详细说明

概述

短文本和图形文本以及输入/输出栏和单位文本总是形成一个单元。即短文本的位置数据也对图形文本和输入/输出栏数据以及单位文本上的数据有效。

编程

设计的位置数据覆盖标准值，即也仅能改变一个单独的值。
如果下列对话框单元没有设计位置数据，则采用上一对话框单元的数据。

如果对话框单元没有规定位置，则使用预设置。
短文本和输入/输出栏的栏宽度在标准情况下各行由栏数和最大栏宽度确定，即：栏宽度=最大行宽/列数。

图像和单位文本宽度是固定的，根据编程支持的请求优化。
如果已设计图像和单位文本宽度，则短文本或者输入/输出栏的宽度相应缩短。

短文本和输入/输出栏的顺序可以通过位置数据互换。

2.3.7 帮助 [11] 的详细说明 (仅适用于 HMI 高级)

说明

可以在运行时扩展或者删除对话框单元的帮助环链。
通过多次调用功能可以任意扩展帮助环链。

编程

句法：**ADDHTX** (名称、帮助文件、索引、帮助文本)
说明：扩展帮助环链
参数：
 名称 待扩展的帮助环链的变量名。
 帮助文件： 文件的路径数据 (PDF 格式)
 索引： 包含帮助文本的文件的索引
 帮助文本： 帮助文本文件中显示的帮助文本
举例：
`ADDHTX(VAR1, "C:\OEM\HLP\MYHLP.PDF", 15, "机床数据")`

句法：**CLRHTX** (名称)
说明：删除帮助环链
参数：
 名称 待删除的帮助环链的变量名称。

帮助文件： 文件的路径数据 (PDF 格式)
 索引： 包含帮助文本的文件的索引
 帮助文本： 帮助文本文件中显示的帮助文本

举例： CLRHTX (VAR1)

2.3.8 Anwendungsbeispiele (应用举例)

辅助变量

辅助变量是内部计算变量。
 计算变量如同变量一样定义，但是另外除了变量值和状态之外没有属性，即辅助变量在对话框中不可见。辅助变量是 VARIANT 类型。

编程

句法： DEF 名称
 说明： VARIANT 类型的内部计算变量
 参数： 名称： 辅助变量名称

举例： DEF OTTO ; 定义一个辅助变量

句法： 名称 val = 辅助变量值
 名称 = 辅助变量值
 说明： 辅助变量值在方式中指定。
 参数： 名称： 辅助变量名称
 辅助变量值： 辅助变量内容

举例：

```
LOAD
    OTTO = "Test" ; 分配值 "Test" 给辅助变量 Otto
END_LOAD
LOAD
    OTTO = REG[9].VAL ; 分配寄存器的值给辅助变量
END_LOAD
```

通过变量计算

在每次退出输入/输出栏 (通过 ENTER 或者转换键) 后计算变量。计算在 CHANGE 方式中设计并在每次更改值时运算。

通过变量状态可以查询，变量是否包含有效值，例如：

```
Var1 = Var5 + SIN(Var2)
```

```
Otto = PI * Var4
```

系统变量间接编译地址

系统变量也可以间接编译地址，即和另一个变量相关。

```
PRESS(HS1)
```

```
轴= 轴 +1
```

```
WEG.VAR="$AA_DTBW[ "<<ACHSE<<" ]" ;编程变量轴地址
```

```
END_PRESS
```

更改软键标签

举例：

```
HS3.st = "新文本" ;改变软键标签
```

2.3.9 举例 1：分配变量类型、文本、帮助、颜色

举例 1

分配变量类型、文本、帮助、颜色的属性

DEF Var1 = (R///,"实际值",,"mm"//"/"Var1.bmp"////8,2)

变量类型：	REAL
极限值或者转换栏输入项：	无
预设值：	无
文本	
长文本：	无
短文本：	实际值
图形文本：	无
单位文本：	mm
属性：	无
帮助画面：	Var1.bmp
系统或者用户变量：	无
短文本位置：	没有规定，则为标准设置
输入/输出栏位置：	没有规定，则为标准设置
颜色：	
前景颜色：	8
背景颜色：	2
帮助：	无

2.3.10 举例 2：定义变量类型，极限值，属性，短文本位置

举例 2

定义变量类型、极限值，属性，短文本位置的属性

DEF Var2 = (I/O,10//wr1,al1/// , ,300)

变量类型：	INTEGER
极限值或者转换栏输入项：	MIN：0
	MAX: 10
预设值：	无
文本	无
属性：	
输入模式	只读
短文本的文本对齐	右对齐
帮助画面：	无
系统或者用户变量：	无
短文本位置：	
到左边缘的间距	无
到上边缘的间距	无，则为到左边缘、上边缘的标准间距
宽度：	300
输入/输出栏位置：	没有规定，则为标准设置
颜色：	没有规定，则为预设置
帮助：	无

2.3.11 举例 3：定义变量类型、预设、系统或者用户变量、输入/输出栏位置

举例 3

定义变量类型属性、预设、系统或者用户变量、输入/输出栏位置

```
DEF Var3 =(S//10//"$R[1]"//300,10,200//"Help.pdf",1,"帮助
1")
```

变量类型：	字符串
极限值或者转换栏输入项：	无
预设值：	10
文本	无
属性：	无
帮助画面：	无
系统或者用户变量：	\$R[1] (R 参数 1)
短文本位置：	相对于输入/输出栏的标准位置
输入/输出栏位置：	
到左边缘的间距	300
到上边缘的间距	10
宽度：	200
颜色：	没有规定，则为预设置
帮助：	如果按下 <i> 键，则用帮助文本“帮助 1”从文件 Help.pdf 中调用包含索引 1 的文件页。

2.3.12 转换栏、帮助调用和画面显示的举例

举例 4

在转换栏中不同的输入项：

极限值或者转换栏输入项：

DEF Var1 = (I/* 0,1,2,3)

DEF Var2 = (S/* "Ein", "Aus")

DEF Var3 = (B/* 1="Ein",
0="Aus")

DEF Var4 = (R/* ARR1)

；1 和 0

是数值，显示“接通 (EIN) ”和“关闭 (AUS) ”

；ARR1 是数组名称。

举例 5 (仅 HMI 高级)

每个对话框包含多个帮助调用：

DEF Var5 = (R/////////"Help1.pdf",1,"帮助 1",,2,"帮助 2","Help3.pdf",3,)

变量类型：	REAL
极限值或者转换栏输入项：	无
预设值：	无
文本	无
属性：	无
帮助画面：	无
系统或者用户变量：	无
短文本位置：	无
输入/输出栏位置：	无
颜色：	没有规定，则为预设置
帮助：	1. 帮助项
帮助文件：	HELP1.PDF
索引：	1
帮助文本：	帮助 1
2. 帮助项	
帮助文件：	HELP2.PDF
索引：	2
帮助文本：	帮助 2
3. 帮助项	
帮助文件：	HELP3.PDF
索引：	3
帮助文本：	帮助 3

举例 6

显示画面，而不是短文本：
图的大小和位置在“输入/输出栏位置（左侧、上部、宽度、高度）”中规定。

DEF VAR6= (V///,"\\图 1.bmp" ///160,40,50,50)

变量类型：	VARIANT
极限值或者转换栏输入项：	无
预设值：	无
文本	无
属性：	无
帮助画面：	无
系统或者用户变量：	无
短文本位置：	图 1.bmp
输入/输出栏的位置	
左侧距离：	160
上部距离：	40
宽度：	50
高度：	50
颜色：	没有规定，则为预设置
帮助：	无

2.3.13 使用字符串

字符串链

设计时也可以使用字符串，以动态配置文本显示或者合并代码生成的不同文本。

规则

使用字符串变量时注意以下规定：

- 链接由左向右处理。
- 层叠的表达式由内向外运算。
- 忽略大小写。

字符串可以通过一个简单的空字符串指令删除。

字符串在等号右边以运算符 "<<" 开始。字符串中双引号 (") 通过两个连续的双引号标记。字符串可以在 IF 指令中检查相等性。

举例

下列举例预设：

```
VAR1.VAL = "这是一个"
```

```
VAR8.VAL = 4
```

```
VAR14.VAL = 15
```

```
VAR2.VAL = "错误"
```

```
$85001 = "这是一个"
```

```
$85002 = "报警文本"
```

编辑字符串：

- 合并字符串：

```
VAR12.VAL = VAR1 << "错误" ;结果： "这是一个错误"
```

- 删除一个变量：

```
VAR10.VAL = "" ;结果： 空字符串
```

- 设定一个带有文本变量的变量：

```
VAR11.VAL = VAR1.VAL ;结果： "这是一个"
```

- 数据类型匹配：

```
VAR13.VAL = "这是 " << (VAR14 - VAR8) << ". 错误"  
;结果： "这是第 11 个错误"
```

- 处理数字值：

```
VAR13.VAL = "错误" << VAR14.VAL << ": " << $T80001 << $T80002
           ;结果： "错误 15： "这是一个报警文本"
IF VAR15 == "错误" ; IF 指令中的字符串
  VAR16 = 18.1234
           ;结果： VAR16 等于 18.1234 ,
           ;当 VAR15 等于"错误"时
ENDIF
```

- 字符串中的双引号：

```
VAR2="Hallo dies ist ein " Test"
           ;结果： 你好，这是一个"测试
```

- 和变量内容有关的系统或者用户变量的字符串：

```
VAR2.Var = "$R[" << VAR8 << "]" ;结果： $R[4]
```

2.3.14 变量 CURPOS

说明

通过变量 CURPOS 可以在当前对话框的激活的输入栏中调出或者操纵光标位置。变量显示光标前有多少个字符。如果光标位于输入栏开始处，则 CURPOS 接受值为 0。如果更改 CURPOS 值，则光标停留在输入栏中相应的位置上。

为了可以在变量值更改情况下反应，可以借助于一个 CHANGE (改变) 块监控改变情况。如果 CURPOS 值改变，则跳转 CHANGE (改变) 块并执行包含的指令。

2.3.15 变量 CURVER

说明

属性 CURVER (当前版本) 允许匹配编程用于处理不同的版本。变量 CURVER 仅可读。

注意

在代码生成时自动以最新的版本生成，即使之前已用老的版本反编译。命令 "GC" 总是生成最新的版本。在生成代码中，在使用注释当版本 > 0 时插入一个生成的代码的附加标记。

规则

总是显示带有所有变量的最新的对话框。

- 以前的变量不允许改变。
- 新的变量以任意顺序插入在以前 (循环) 的编程中。
- 不允许从对话框由一个版本到下一个版本去除变量。
- 对话框必须包含所有版本的变量。

举例

```
( IF CURVER==1 ... ) ; CURVER 在反编译时自动通过反编译代码的版本设置。
```

2.3.16 变量 ENTRY

说明

通过变量 ENTRY 可以检查如何调用对话框。

编程

句法 : **ENTRY**
说明 : 变量 ENTRY 仅可读。
返回值 : 查询结果可能是 :
 0 = 没有编程支持
 1 = 编程支持 (由编程支持调用对话框。)
 2 = 编程支持 + 上一对话框的预设置 (子对话框)
 3 = 编程支持 + 反编译
 4 = 编程支持 + 反编译, 带有生成的注释, 带有 # 符号
 5 = 编程支持 + 反编译, 带有生成的注释, 不带 # 符号

举例

```
IF ENTRY == 0
  DLGL ( "在编程下不调用对话框" )
ELSE
  DLGL ( "在编程下调用对话框" )
ENDIF
```

2.3.17 变量 ERR

说明

通过变量 ERR 可以检查是否已正确执行先前行。

编程

句法 : **ERR**
说明 : 变量 ERR 仅可读。
返回值 : 查询结果可能是 :
 FALSE = 已正确执行先前行
 TRUE = 未正确执行先前行

举例

```
VAR4 = 螺纹[VAR1,"KDM",3] ; 作为数组给出值
IF ERR == TRUE ; 查询是否在数组中找到值
VAR5 = "数组存取出错" ; 如果未在数组中找到值,则分配值"数组存取出错"给变量。
ELSE
VAR5 = "一切正常" ; 如果在数组中找到值,则分配值"一切正常"给变量。
ENDIF
```

2.3.18 变量 FILE_ERR

说明

用变量 FILE_ERR 可以检查前面的 GC 或者 CP 指令是否正确执行。

编程

句法：**FILE_ERR**

说明：变量 FILE_ERR 仅可读。

返回值：可能的结果是：

- 0 = 排列中的操作
- 1 = 驱动器/路径不存在
- 2 = 路径/文件存取故障
- 3 = 驱动器未就绪
- 4 = 错误的文件名
- 5 = 文件已经打开
- 6 = 存取失败
- 7 = 目标路径不存在或不允许
- 8 = 复制源符合目标
- 10 = 内部错误：FILE_ERR = 10
时，与一个未分类到其他类别故障有关。

举例

```

CP("D:\source.mpf","E:\target.mpf")
; 从 source.mpf 向 E:\target.mpf 复制
IF FILE_ERR > 0
; 询问是否出现故障
IF FILE_ERR == 1
; 查询特定的错误号并输出所属的故障文本
VAR5 = "驱动器/路径不存在"
ELSE
IF FILE_ERR == 2
VAR5 = "路径 - /文件存取故障"
ELSE
IF FILE_ERR == 3
VAR5 = "错误文件名"
ENDIF
ENDIF
ENDIF
ELSE
VAR5 = "一切正常"
; 如果 CP (或者 GC)中没有出现错误，则输出“一切正常”
ENDIF

```

2.3.19 变量 FOC

说明

使用变量 FOC 可以控制对话框内的输入中心 (当前有效的输入/输出栏) 。
已预先固定定义光标左右键、前后键以及 PGUP、PGDN 的反应。

注意

FOC 不允许通过一个导航事件触发。光标位置只允许在软键 PRESS 块, CHANGE 块, ... 中改变。

带有输入模式 wr = 0 和 wr = 4 的变量以及辅助变量无法实现聚焦。

编程

句法 :	FOC				
说明 :	可以读取和写入该变量。				
返回值 :	<table border="0" style="margin-left: 20px;"> <tr> <td>读取</td> <td>聚焦变量的名称作为结果输出。</td> </tr> <tr> <td>写入</td> <td>可以写入一个字符串或者数值。 字符串视为变量名称, 数值视为变量索引。</td> </tr> </table>	读取	聚焦变量的名称作为结果输出。	写入	可以写入一个字符串或者数值。 字符串视为变量名称, 数值视为变量索引。
读取	聚焦变量的名称作为结果输出。				
写入	可以写入一个字符串或者数值。 字符串视为变量名称, 数值视为变量索引。				

举例

```

IF FOC == "Var1"                ; 读取输入中心
    REG[1] = Var1
ELSE
    REG[1] = Var2
ENDIF

FOC = "Var1"                    ; 分配变量 1 给输入中心。
FOC = 3                        ; 分配第 3 对话框单元, WR ≥ 2 给输入中心。
    
```

2.3.20 变量 S_CHAN

说明

通过变量 S_CHAN 可以确定用于显示或者某个评估的当前通道号码。

2.4 综合对话框单元

2.4.1 数组

定义

通过数组可以排列或者保存同一数据类型的数据，从而可以通过索引存取数据。

说明

数组可以是一维或者二维。一个一维数组可视为带有一行或者一列的一个二维数组。数组通过标记 //A 定义并通过 //END 结束。行和列数目任意。一个数组有下列结构：

编程

```
句法：      //A(名称)
              (a/b...)
              (c/d...)
              ...
              //END
说明：      定义数组
参数：      名称          数组名称
              a, b, c, d    数组值
                          STRING 类型的值必须用双引号括起。
```

举例

```
//A( 螺纹)          ;   高度/螺距/底直径
(0.3 / 0.075 / 0.202)
(0.4 / 0.1   / 0.270)
(0.5 / 0.125 / 0.338)
(0.6 / 0.15  / 0.406)
(0.8 / 0.2   / 0.540)
(1.0 / 0.25  / 0.676)
(1.2 / 0.25  / 0.676)
(1.4 / 0.3   / 1.010)
(1.7 / 0.35  / 1.246)
//END
```

2.4.2 存取数组单元的值

说明

通过属性值 (名称.val) 可以继续传送一个数组存取值。

行索引 (数组的行编号) 和列索引 (数组的列编号) 各自从 0 开始。如果显示数组外的行索引或者列索引, 则输出值 0 或者空字符串并且变量 ERR 的值为 TRUE。当未找到查找关键字时, 变量 ERR 同样为 TRUE。

编程

句法 :	名称 [Z,[M],[C]].val 或者 名称 [Z,[M],[C]]
说明 :	存取仅带有一列的一维数组 :
句法 :	名称 [S,[M],[C]].val 或者 名称 [S,[M],[C]] 或者
说明 :	存取仅带有一行的单维数组
句法 :	名称 [Z,S,[M],[C]].val 或者 名称 [Z,S,[M],[C]]
说明 :	存取二维数组
参数 :	名称 : 数组名称 Z: 行值 (行索引或者查找关键字) S: 列值 (列索引或者查找关键字) M: 存取模式 0 直接 1 按行查找, 列直接 2 按列查找, 行直接 3 查找 4 查找行索引 5 查找列索引 C: 比较模式 0 查找关键字必须位于行或者列的值范围内。 1 查找关键字必须准确找到
举例 :	<pre>VAR1 = MET_G[REG[3],1,0].VAL ; 分配 Var1 数组中的一个值 MET_G</pre>

存取模式

- 存取模式“直接”

在存取模式“直接” (M = 0) 情况下, 数组上的存取通过行索引 (以 Z 表示) 和列索引 (以 S 表示) 实现。不评估比较模式 C。

- 存取模式“查找”

在存取模式 M = 1、2 或者 3 情况下，查找总是在行 0 或者列 0 中实现。

模式 M	行值 Z	列值 S	输出值
0	行索引	栏索引	行 Z 和列 S 中的值
1	查找关键字： 在列 0 中查找	列索引，从该列中读取值	查找的行和列 S 中的值
2	行索引，从该行中读取返回值	查找关键字： 在行 0 中查找	行 Z 和查找的列中的值
3	查找关键字： 在列 0 中查找	查找关键字： 在行 0 中查找	查找的行和查找的列中的值
4	查找关键字： 在列 S 中查找	行索引，在该行中查找	行索引
5	行索引，在该行中查找	查找关键字： 在行 Z 中查找	栏索引

比较模式

在使用比较模式 C = 0 时，查找行或者查找列的内容以升序分类。
如果查找关键字小于第一个元素或者大于最后一个元素，则给出值 0
或者一个空字符并且错误变量 ERR 为真。

在使用比较模式 C = 1 时，查找关键字必须可在查找行或者查找列中找到。
如果没有找到查找关键字，则给出值 0 或者一个空字符并且错误变量 ERR 为真。

2.4.3 举例：存取数组单元

前提条件

下列定义两个数组是下面例子的前提条件。

```
//A(螺纹
      (0.3 / 0.075 / 0.202)
      (0.4 / 0.1   / 0.270)
      (0.5 / 0.125 / 0.338)
      (0.6 / 0.15  / 0.406)
      (0.8 / 0.2   / 0.540)
      (1.0 / 0.25  / 0.676)
      (1.2 / 0.25  / 0.676)
      (1.4 / 0.3   / 1.010)
      (1.7 / 0.35  / 1.246)

//END
```

```
//A(Array2)
      ("BEZ" /      "STG" /      "KDM" )
      (0.3 /        0.075 /      0.202 )
      (0.4 /        0.1 /        0.270 )
      (0.5 /        0.125 /      0.338 )
      (0.6 /        0.15 /       0.406 )
      (0.8 /        0.2 /        0.540 )
      (1.0 /        0.25 /       0.676 )
      (1.2 /        0.25 /       0.676 )
      (1.4 /        0.3 /        1.010 )
      (1.7 /        0.35 /       1.246 )

//END
```

举例

- **存取模式 1 举例：**

在 Z 中有查找关键字。该关键字总是在列 0 中查找。通过查找的关键字的行索引给出列 S 中的值：

```
VAR1 = 螺纹[0.5,1,1] ;VAR1 值为 0.125
```

说明：

数组“螺纹”的列 0 中查找值 0.5 并给出列 1 中查找的相同行的值。

- **存取模式 2 举例：**

在 S 中有查找关键字。该关键字总是在行 0 中查找。通过查找的关键字的列索引给出行 Z 中的值：

```
VVAR1 = ARRAY2[3,"STG",2] ;VAR1 值为 0.125
```

说明：

在数组“Array2”的行 0 中查找带有内容“STG”的列。得到查找的列中的值和带有索引 3 的行。

- **存取模式 3 举例：**

在 Z 和 S 中总是有一个查找关键字。通过 Z 中的关键字在列 0 中找到行索引和通过 S 中的关键字在行 0 中找到列索引。通过查找的行索引和列索引输出数组中的值：

```
VAR1 = ARRAY2[0.6,"STG",3] ;VAR1 值为 0.15
```

说明：

在数组“Array2”的列 0 中查找带有内容 0.6 的行，在数组“Array2”的行 0 中查找带有内容“STG”的列。根据 VAR1 给出查找的行和列中的值。

• 存取模式 4 举例：

在 Z 中有查找关键字。在 S 中有要查找的列索引。给出查找的关键字的行索引：

VAR1 = 螺纹[0.125,1,4] VAR1 值为 2

说明：

在数组“螺纹”的列 1 中查找值 0.125 并根据 VAR1 给出查找的值的行索引。

• 存取模式 5 举例：

在 Z 中有要查找行的行索引。查找关键字在 S 中。给出查找的关键字的列索引：

VAR1 = 螺纹[4,0.2,5,1] VAR1 值为 1

说明：

在数组“螺纹”的行 4 中查找值 0.2 并根据 VAR1 给出查找的值的列索引。已选择比较模式 1，因为行 4 的值不是按升序分类的。

2.4.4 查询数组单元的状态

说明

通过属性状态可以查询数组存取是否提供一个有效值。

编程

句法：*名称 [Z, S, [M, C]]*.vld

说明：该属性仅可读。

参数：名称 数组名称

返回值：FALSE = 无效值
TRUE = 有效值

举例

```

DEF MPIT = (R///"MPIT",,"MPIT",""/wr3)
DEF PIT = (R///"PIT",,"PIT",""/wr3)
PRESS(VS1)
  MPIT = 0.6
  IF MET_G[MPIT,0,4,1].VLD == TRUE
    PIT = MET_G[MPIT,1,0].VAL
    REG[4] = PIT
    REG[1] = "OK"
  ELSE
    REG[1] = "ERROR"
  ENDF
END_PRESS

```

2.4.5 表格栅格 (Grid)

定义

和数组相反，表格栅格 (Grid) 的值持续更新。
它与系统变量值的表格式描述有关，系统变量可以通过一个通道中的一个模块编译地址。

分配

通过表格名称将变量定义分配给表格单元定义：

- 变量定义确定显示的值，表格单元定义确定屏幕的外观和布置。
表格栅格从变量定义行接收输入 / 输出栏的属性。
- 栅格的可见区域通过输入 / 输出栏宽度和高度确定。
如果比可见区域位置存在更多的行和列，则可以通过水平和垂直滚动条进行查看。

说明

表格描述参考内容收录在变量描述中：

DEF <i>Bezeichner</i> =	Bezeichner = 变量名称		
	变量类型	→	1
	/[极限值或者转换栏或者表格名称]	→	2
	/[预设值]	→	3
	/[文本 (长文本, 短文本 图, 图形文本, 单位文本)]	→	4
	/[属性]	→	5
	/[帮助画面]	→	6
	/[系统或者用户变量]	→	7
	/[短文本位置]	→	8
	/[输入/输出栏位置 (左、上、宽度、高度)]	→	9
	/[颜色]	→	10
	/[帮助] (仅 HMI 高级)	→	11

表格名称 [2]

NCK/PLC 相同类型值的表格名称，它可以通过通道模块编译地址。
表格名称通过一个前置的极限值或者转换栏的 % 符号进行区分。
表格命名符可以通过逗号分隔，还可以跟随一个文件名，该文件名指定定义表格描述的文件。

系统或者用户变量 [7]

参数留空用于表格栅格，因为详细信息中要显示的变量规定在列定义行中。
表格描述可以动态提供。

2.4.6 定义表格栅格

说明

表格块由以下部分组成：

- 标题描述
- 1 至 n 列描述

编程

句法：*IIG (表格名称/表格类型/行数 / [固定行属性], [固定列属性])*

说明：定义表格栅格

参数：

表格名称	这里表格名称不使用前缀 % 符号。 表格名称只能在一个对话框中使用一次。
表格类型	0 (预设置) PLC 或者用户数据 (NCK 和通道专用的数据) 表 1 备用
行数	行数包括标题行 固定行或者固定列无法滚动显示。 列数由设计的列的数目给定。
固定行属性	1: 激活 0: 未激活
固定列属性	1: 激活 0: 未激活

2.4.7 定义列

说明

表格栅格中可以通过索引使用变量。索引号码对于带有一个或者多个索引的 PLC 或者 NC 变量比较重要。

表格栅格中显示的值可以由最终用户在由属性确定的权限范围和可能定义的极限值范围中直接编辑。

编程

句法：	<i>(类型/极限值/空/长文本, 列标题/属性/帮助画面/系统或者用户变量/列宽/偏移 1、 偏移 2、 偏移 3)</i>	
说明：	定义列	
参数：	和变量类似	
	类型	数据类型
	极限值	最小极限值，最大极限值
	长文本，列标题	
	属性	
	帮助画面	
	系统或者用户变量	作为变量在双引号内给出 PLC 或者 NC 变量。
	列宽	参数，单位像素
	偏移	在分配的偏移参数内规定步宽（在该步宽中各个索引应指数运算），以填写该列。
		<ul style="list-style-type: none"> • 偏移 1：第 1 个索引的步宽 • 偏移 2：第 2 个索引的步宽 • 偏移 3：第 3 个索引的步宽

STRING 类型变量

如果变量类型是 STRING，必须规定该类型的变量长度，例如：`DEF CHAN STRING [16]
TEXT[41]`

变量 CHAN 的列定义开始，例如：`(S16/...)`

文本文件的列标题

列标题可以规定为文本或者文本号码（`$8xxxx`），并且同样无法滚动显示。

改变列属性

可动态改变的（可写）的列属性称为：

- 极限值（最大、最小）、
- 列标题(st)、
- 属性(wr, ac 和 li)
- 帮助画面(hlp)和
- BTSS 变量(var)。

通过定义行中的变量命名符和列索引（以 1 开始）改变列属性。

举例：`VAR1[1].st="列 1"`

无法在 LOAD 块中读取列属性。

对于列定义，可以规定属性 wr、ac 和 li。

参见

对话框单元 (页 21)

可用的系统变量列表 (页 196)

2.4.8 表格栅格中的聚焦控制

说明

通过列和行属性可以在表格中设置和确定聚焦：

- 名称Row
- 名称Col

编程

表格的每行都具有属性 Val 和 VId。

对于行属性写入和读出，除了定义行中的变量命名符之外，还规定一个行索引和列索引。

句法：	名称[行索引，列索引].VId 或者 名称[行索引，列索引]
说明：	Val 属性
句法：	名称[行索引，列索引].VId
说明：	VId 属性

举例

```
Var1[2,3].val=1.203
```

如果没有规定行索引和列索引，则适用于聚焦行的索引，即：

```
Var1.Row =2  
Var1.Col=3  
Var1.val=1.203
```

2.4.9 举例：定义列

概述

下面三个举例指出了标准表格 (表格类型= 0) 中单格和 PLC 变量之间的分配。

举例 1：

在第一行中显示列标题：

```

//G(MB_TAB/0/4/,1)
(I///,"MB 1 至 MB 3"///"MB1"/100/1)
(I///,"MB 4 至 MB 6"///"MB4"/100/1)
    
```

结果：

MB 1 至 MB 3	MB 4 至 MB 6
值(MB1)	值(MB4)
值(MB2)	值(MB5)
值(MB3)	值(MB6)

举例 2：

如果列定义的偏移 > 1，则产生以下行和列的分配：

```

//G(MB_TAB/0/4/,1)
(I///," MB1, MB3, MB5"///"MB1"/100/2)
(I///," MB2, MB4, MB6"/// "MB2"/100/2)
    
```

结果：每行中都已提高了变量索引偏移 (=2)。

MB1、MB3、MB5	MB2、MB4、MB6
值(MB1)	值(MB2)
值(MB3)	值(MB4)
值(MB5)	值(MB6)

举例 3 :

列定义中的偏移和索引号码 :

- 在第一列中每行变量的第一个索引提高 1。偏移 1 = 1
- 在第二列中每行变量的第二个索引提高 1。偏移 2 = 1

```
//G(MB_TAB/0/4/,1)
(IB///,"M1.1, M2.1, M3.1"/// "M1.1"/100/1)
(IB///,"M1.1, M1.2, M1.3"/// "M1.1"/100/,1)
```

结果 :

M1.1, M2.1, M3.1	M1.1、M1.2、M1.3
值 (M1.1)	值 (M1.1)
值 (M2.1)	值 (M1.2)
值 (M3.1)	值 (M1.3)

其它可能 :

- 第一列可以用递升的数字填写 :
举例: (I///,"行"///"0"/60/1)
- 第一列可以用语言文件中的连续文本填写 :
举例: (S///,"行"///"\$80000"/60/1)

2.4.10 举例：载入不同的表格栅格

说明

在本举例中变量"VAR1"首先指定表格"dummygrid"。LOAD 块中，根据 R 参数 R[0] 的内容，选择载入表格 "grid1" 或者 "grid2"。
在同一文件中和定义变量"VAR1"一样定义表格：

```
//M(MASKE1/"GRID")
DEF VAR1=(R/% dummygrid/////////200,75,300,85)
HS1=""
HS2=""
HS3=""
HS4=""
HS5=""
HS6=""
HS7=""
HS8=""
VS1=""
VS2=""
VS3=""
VS4=""
VS5=""
VS6=""
VS7=("EXIT",ac7,sel)
VS8=""

LOAD
  REG[0] = RNP ("R[0]")
  IF (REG[0] == 0)
LG ("grid1", "var1")
ELSE
LG ("grid2", "var1")
ENDIF
END_LOAD

PRESS(VS7)
EXIT
END_PRESS

//END

//G(grid1/0/5/1,1) ; (名称/类型/行...)
(R///"长文本 1", "R1 至 R4" /wr2//"$R[1]" /80/1) ; 第 1 列，标题"R1 至 R4"，自 R1 起带偏移 1
```

```

(R///"长文本 2", "R5 至 R8"/wr2/"$R[5]"/80/1) ; 第 2 列自 R5 起
(R///"长文本 3", "R9 至 R15"/wr2/"$R[9]"/80/2) ; 第 3 列自 R9 带偏移 2, R9 11 13 15
//END

//G(grid2/0/5/1,1) ; (名称/类型/行...)
(R///"长文本 1", "R1 至 R4"/wr2/"$R[1]"/60/1) ; 第 1 列, 标题"R1 至 R4", 自 R1 起带偏移 1
(R///"长文本 2", "R5 至 R8"/wr2/"$R[5]"/60/1) ; 第 2 列自 R5 起
(R///"长文本 3", "R9 至 R15"/wr2/"$R[9]"/60/2) ; 第 3 列自 R9 带偏移 2, R9 11 13 15
(R///"长文本 4", "R9 至 R15"/wr2/"$R[9]"/60/2) ; 第 3 列自 R9 带偏移 2, R9 11 13 15
//END

//G(dummygrid/0/5/1,1) ; (名称/类型/行...)
(R///"长文本 1", "R1 至 R4"/wr2/"$R[1]"/80/1) ; 第 1 列, 标题"R1 至 R4", 自 R1 起带偏移 1
(R///"长文本 2", "R5 至 R8"/wr2/"$R[5]"/80/1) ; 第 2 列自 R5 起
//END

```

2.5 软键栏

2.5.1 软键描述

概述

确定软键名称。不必占用所有软键。

HSx x 1 - 8, , 水平软键 1 至 8

VSy y 1 - 8, 垂直软键 1 至 8

所有水平软键和所有垂直软键分别布置在一起称作软键栏。

另外针对已有的软键栏，还可以定义其它软键栏，可以部分或者完全覆盖已有的软键栏。

原则上软键栏描述（描述块）如下构建：

说明块	注释	参考章节
//S...	; 软键栏开始标记	
HSx=...	; 定义软键	
PRESS(HSx) LM... END_PRESS	; 方法的开始标识 ; 动作 ; 方法的结束标识	参见章节“方法”
//END	; 软键栏结束标记	

2.5.2 定义软键栏

说明

定义软键栏也同时分配软键属性。

编程

句法 :	<i>//S(名称)</i>	;软键栏开始标记
	...	
	<i>//END</i>	;软键栏结束标记
说明 :	定义软键栏	
参数 :	名称	软键栏名称
句法 :	SK = (文本[, 存取等级][, 状态])	
说明 :	定义软键	
参数 :	SK	软键, 例如: HS1 到 HS8, VS1 到 VS8
	文本	给定文本
	图文件名称	"\\my_pic.bmp" 或者通过单独的文本文件 \$85199, 例如: 和语言相关的文本文件的下列文 本: 85100 0 0 "\\c:\pic\my_pic.bmp".
		显示在软键上的图形大小: 最大 80 x 34 像素
	存取等级	ac0 到 ac7 (ac7: 预设置)
	状态	se1: 可见 (预设置) se2: 不可操作 (灰色标签) se3: 高亮显示 (最后操作的软键)

注意

对于软键标签, 用 %n 进行换行。

在 HMI 高级上, 软键最多 2 行, 每行最多 10 个字符; 在 HMI 内置 sl 上, 最多 2 行, 每行最多 9 个字符。

分配保护等级

操作员只能访问符合其保护等级和各个低于其保护等级的信息。

不同的保护等级有下列不同的含义：ac0 是最高的保护等级，ac7 是最低的保护等级。

保护等级	禁用密码	区域
ac0	密码	西门子
ac1	密码	机床制造商
ac2	密码	维修
ac3	密码	用户
ac4	钥匙开关位置 3	编程员，调试员
ac5	钥匙开关位置 2	合格的操作员
ac6	钥匙开关位置 1	受过培训的操作员
ac7	钥匙开关位置 0	学过相关内容的操作员

举例

```

//S(软键栏 1) ; 软键栏开始标记
HS1=("新建", ac6, se2) ; 定义软键 HS1, 分配标签 "OK", 保护等级 6
                        ; 和状态"不可操作"。
HS3(("\\图 1.bmp") ; 分配图形给软键
HS5("Exit")
VS2("子屏幕窗口")
VS3($85011, ac7, se2)
VS7("取消", ac1, se3) ; 定义软键 VS7, 分配标签"取消", 保护等级 1 和状态"高亮"。
VS8("OK", ac6, se1) ; 定义软键 VS8, 分配标签 "OK", 保护等级 6 和状态"可见"。

PRESS(HS1) ; 方法开始标记
  HS1.st="计算" ; 分配标签文本给软键
  ...
END_PRESS ; 方法结束标记

PRESS(RECALL) ; 方法开始标记
  LM("屏幕窗口 21") ; 载入对话框
END_PRESS ; 方法结束标记
//END ; 软键栏结束标记

```

按键<RECALL> (回调)



另外还有回调 <RECALL> 按键可用于操作。

该按键与软键相反，不必定义。在运行期间内，可以向按键分配属性“状态”和“存取等级”。

如果该按键没有指定动作（功能、变量计算、属性改变），则您可以通过回调 <RECALL> 按键退出新配置的操作界面并返回标准应用程序。

举例

```

PRESS(RECALL)
    RECALL.ac = 1
    LM("屏幕窗口 5")
END_PRESS
    
```

2.5.3 运行时改变软键属性

说明

文本、存取等级和状态的属性可以在运行期间在方法中改变：

编程

句法：	SK.st = "文本" ; 软键标签 SK.ac = 存取等级 ; 软键存取等级 SK.se = 状态 ;软键状态
说明：	分配属性
参数：	文本 引号中的标签文本 存取等级 值范围：0 ... 7 状态 1: 可见并可操作 2: 不可操作 (灰色标签) 3: 高亮显示 (最后操作的软键)

举例

```

//S(Start)
HS7=("举例", ac7, se1)

PRESS(HS7)
    LM("屏幕窗口 3")
END_PRESS

//END

//M(屏幕窗口 3/"举例 3: 图形和软键"/"MST.BMP")
HS1=("")
    
```

```
HS2= (" ")  
HS3= (" ")  
HS4= (" ")  
HS5= (" ")  
HS6= (" ")  
HS7= (" ")  
HS8= (" ")  
VS1= (" ")  
VS2= (" ")  
VS3= (" ")  
VS4= (" ")  
VS5= (" ")  
VS6= (" ")  
VS7= (" ")  
VS8= ("OK", AC7, SE1)  
PRESS(VS8)  
EXIT  
END_PRESS  
//END
```

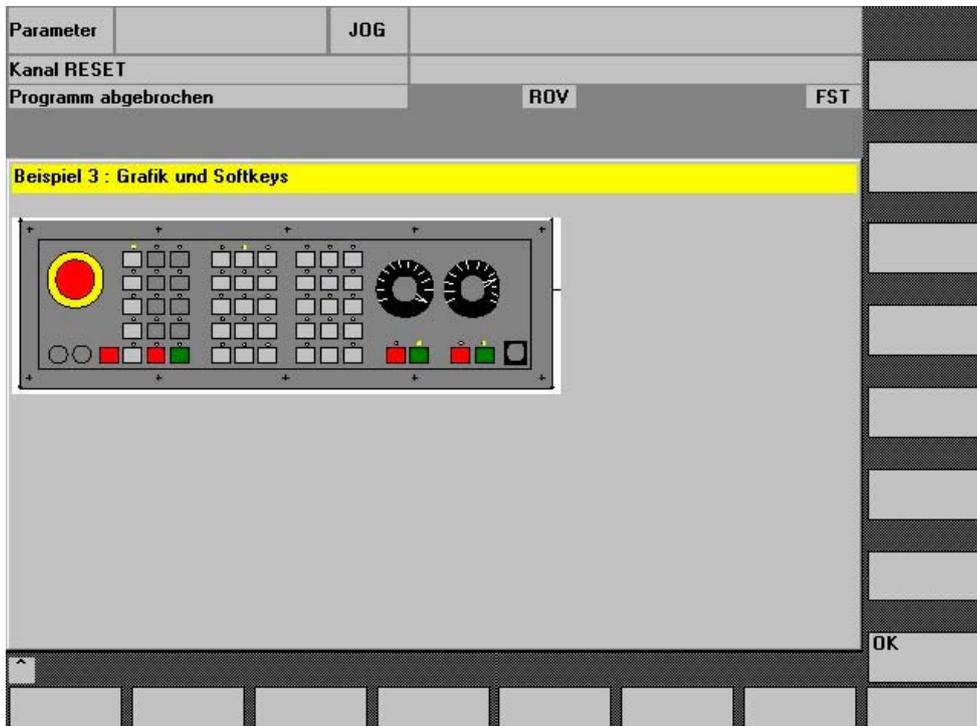


图 2-9 举例 3 : 图形和软键

2.5.4 定义登入软键

和对话框无关的软键

登入软键是和对话框无关的软键，它不由对话框调用，而是在第一个新对话框之前设计。为了可以到达登入对话框或者一个登入软键栏，必须对登入软键进行定义。

编程

登入软键的描述块如下构建：

```
//S(Start)           ; 登入软键开始标记
HS1=(...)           ; 定义登入软键：水平软键 SK 1
PRESS(HS1)          ; 方法
LM...               ; 功能 LM 或者 LS
END_PRESS           ; 方式结束
//END                ; 登入软键结束标记
```

2.5.5 登入软键的功能

和对话框无关的软键的功能

通过登入软键只可以触发特定的功能。

允许下列功能：

- 通过功能 **LM** 可以装载另一个对话框。 **LM("名称", "文件")**
- 通过功能 **LS** 可以显示另一个软键栏。 **LS("名称", "文件", [合并])**
- 通过功能 **"EXIT"** 可以离开新配置的操作界面并返回标准应用程序。
- 通过功能 **"EXITLS"** 可以离开当前的操作界面并装载一个定义的软键栏。
- 通过功能 **"EXE"** 可以在 HMI 高级上调用一个程序，该程序通过 HMI 高级的 OEM 包建立作为应用程序，或者调用自由轮廓编程。在 HMI 内置 sl 上只能通过 **"EXE"** 启动自由轮廓编程。

方法 PRESS

在描述块内定义软键并在方法 PRESS 内分配功能 "LM" 或者 "LS"。

如果登入软键定义标记为注释（在行开始处用分号；）或者已删除设计文件，则登入软键无效。

```

//S(Start) ; 开始标记
HS6=("第 1 屏幕窗口") ; 水平 SK 6, 标签为"第 1 屏幕窗口"
PRESS(HS6) ; PRESS 方法用于水平软键 SK 6
    LM("屏幕窗口 1") ; 装载功能屏幕窗口
                        ; 1, 此时必须在同一文件内定义屏幕窗口 1。
END_PRESS ; PRESS 方法结束
HS7=("第 2 屏幕窗口") ; 水平 SK 7, 标签为"第 2 屏幕窗口"
PRESS(HS7) ; PRESS 方法用于水平软键 SK 7
    LM("屏幕窗口 2") ; 装载功能屏幕窗口
                        ; 2, 此时必须在同一文件内定义屏幕窗口 2。
END_PRESS ; PRESS 方法结束
//END ; 登入块的结束标记

```

举例

```

HS1 = ("新的软键栏")
HS2 = ("没有功能")
PRESS(HS1)
    LS("软键栏 1") ; 载入新的软键栏
END_PRESS
PRESS (HS2) ; 空的 PRESS 方法
END_PRESS

```

参见

装载软键(LS) (页 99)

装载屏幕窗口 (LM) (页 98)

2.6 方法

概述

在对话框和与对话框相关的软键栏中（软键栏由新设计的对话框调用），可以通过不同的事件（退出输入栏，按下软键）触发某些特定的动作。这些动作设计在方法中。

方法的基本编程按如下方式进行：

说明块	注释	参考章节
PRESS(HS1)	；方法的开始标识	
LM... LS...	；功能	参见章节“功能”
Var1.st = ...	；改变属性	参见章节“软键栏” 和章节“对话框单元”
Var2 = Var3 + Var4 ... EXIT	；通过变量计算	参见章节“定义变量”
END_PRESS	；方法的结束标识	

2.6.1 CHANGE

说明

当变量值已改变时运行 CHANGE（改变）方法。即在 CHANGE（改变）方法中设计变量改变时立即运行的变量计算。

单元特定的 CHANGE 方法和全局的 CHANGE 方法有所不同：

- 当特定变量值已改变时运行**单元特定的 CHANGE 方法**。
如果系统或者用户变量已分配一个变量，则可以在 CHANGE 方法中循环更新变量值。
- 当改变任意一个变量值且没有设计单元特定的 CHANGE 方法时，运行**全局 CHANGE 方法**。

编程“单元特定”

```

句法：          CHANGE(名称)
                ...
                END_CHANGE

说明：          修改指定变量的值

参数：          名称          变量名称
    
```

举例

```
DEF VAR1=(S////////"DB20.DBB1")           ; Var1 分配一个系统变量
CHANGE(VAR1)
IF VAR1.Val <> 1
    VAR1.st="工具正确!"                   ; 如果系统变量的值 ≠ 1，则变量的短文本为：
                                           工具正确！
        otto=1
ELSE
    VAR1.st="注意 错误！"                 ; 如果系统变量的值 = 1，则变量的短文本为：
                                           注意 错误！
        otto=2
ENDIF
VAR2.Var=2
END_CHANGE
```

编程“全局特定”

句法：	CHANGE() ... END_CHANGE
说明：	改变任意变量值
参数：	- 无 -

举例

```
CHANGE()
EXIT                                       ; 如果任何一个变量值改变，则退出对话框。
END_CHANGE
```

2.6.2 FOCUS

说明

当对话框中聚焦 (光标) 定位在另一个栏上时，运行 FOCUS 方法。

方法 FOCUS 不允许通过一个导航事件触发。光标位置只允许在软键 PRESS 块，CHANGE 块，...中改变。光标移动的反应预先固定定义。

注意

在 FOCUS 块中不允许定位在另一个变量上并且也不允许装载新的对话框。

编程

句法：	FOCUS
	...
	END_FOCUS
说明：	光标：定位
参数：	- 无 -

举例

```
FOCUS
DLGL( "聚焦已设定在变量" << FOC << "上。" )
END_FOCUS
```

o o

2.6.3 LOAD GRID

说明

表格描述可以通过方法 LG 在 LOAD 块内动态提供。

为此可以通过方法 LG

指定一个表格，必须已定义变量作为栅格变量并参考有效且已存在的表格。

编程

句法：	LG (栅格名称, 变量名称 [,文件名称])	
说明：	载入表格	
参数：	栅格名称	表格 (栅格) 名称，用双引号括起
	变量名	应指定表格的变量名称，用双引号括起
	文件名	文件名称，在该文件中定义表格 (栅格)，用双引号括起
		如果表格未在文件中定义，在该文件中也定义变量，则必须指定。

2.6.4 LOAD

说明

在已编译变量定义和软键定义 (DEF Var1= ..., HS1= ...) 后运行 LOAD 方法。
此时，对话框还未显示。

编程

句法：	LOAD
	...
	END_LOAD
说明：	装载
参数：	- 无 -

举例

```
LOAD ; 开始标记
  屏幕窗口 1.Hd = $85111 ; 分配语言文件中的对话框标题
  VAR1.Min = 0 ; 分配变量的最小极限值
  VAR1.Max = 1000 ; 分配变量的最大极限值
END_LOAD ; 结束标记
```

2.6.5 UNLOAD

说明

在卸载对话框之前，运行 UNLOAD 方法。

编程

句法：	UNLOAD
	...
	END_UNLOAD
说明：	卸载
参数：	- 无 -

举例

```
UNLOAD
  REG[1] = VAR1           ; 保存寄存器的变量
END_UNLOAD
```

2.6.6 OUTPUT

说明

当调用功能 "GC" 时，运行 OUTPUT 方法。在 OUTPUT 方法中变量和辅助变量作为 NC 代码设计。代码行各个单元的连接用一个空格符实现。

注意

NC 代码可以用文件功能在一个额外的文件中生成并移向 NC。

编程

句法：	OUTPUT(名称)
	...
	END_OUTPUT
说明：	输出 NC 程序中的变量
参数：	名称 OUTPUT 方法的名称

程序段号码和隐藏标记

当要继续保留编程支持激活时在零件程序中直接设置的行号和反编译时的隐藏标记时，OUTPUT 块不允许包含行号和隐藏标记。

在零件程序中通过编辑器更改影响下列行为：

条件	行为
程序段数目保持不变。	程序段号码继续保留。
程序段数目变小。	删除最大的程序段号码。
程序段数目变大。	新的程序段没有程序段号码。

举例

```

OUTPUT(CODE1)
  "CYCLE82(" Var1.val "," Var2.val "," Var3.val ","Var4.val "," Var5.val
  "," Var6.val ") "
END_OUTPUT

```

2.6.7 PRESS

说明

当已按下相应的软键时，运行 PRESS 方法。

编程

句法：	PRESS(软键)
	...
	END_PRESS
名称：	按下软键
参数：	软键 软键名称：HS1 - HS8 和 VS1 - VS8
	回调 按键<RECALL> (回调)
	PU Page Up 向前翻页
	PD Page Down 向后翻页
	SL Scroll Left 光标向左
	SR Scroll Right 光标向右
	SU Scroll Up 光标向上
	SD Scroll Down 光标向下

举例

```

HS1 = ("其他软键栏")
HS2 = ("没有功能")
PRESS(HS1)
    LS("软键栏 1")                ; 载入其他软键栏
    Var2 = Var3 + Var1
END_PRESS
PRESS (HS2)
END_PRESS
PRESS(PU)
    INDEX = INDEX -7
CALL("UP1")
END_PRESS

```

2.6.8 举例：管理带 OUTPUT 块的版本

概述

已有的对话框可以通过扩展 补充附加的变量。

附加的变量在定义中变量名后的圆括号内有一个版本识别号：（0 = 原始，未写入），1 = 版本 1，2 = 版本 2，...

举例：

```

DEF var100=(R//1)                ; 原始，相当于版本 0
DEF var101(1)=(S//"Hallo")      ; 补充，从版本 1 起

```

在写入 OUTPUT 块时可以参考某个版本状态，与定义的总体性有关。

举例：

```

OUTPUT(NC1)                      ; ;在 OUTPUT 块中仅提供原始变量
OUTPUT(NC1,1)                    ; 在 OUTPUT 块中提供原始变量和带有版本标识符 1
                                ; 的补充变量

```

原始的 OUTPUT 块不需要版本标识符，然而也可以记为 0。OUTPUT(NC1) 相当于 OUTPUT(NC1,0)。OUTPUT 块中的版本标识符 n 包括所有变量，从原始 0、1、2、... 直至 n。

编程版本标识

```

//M(XXX)                                ; 版本 0 (默认)
DEF var100=(R//1)
DEF var101=(S//"Hallo")
DEF TMP
VS8=("GC")
PRESS(VS8)
GC("NC1")
END_PRESS

OUTPUT(NC1)
var100",,"var101
END_OUTPUT

; ***** 版本 1, 补充的定义 *****
//M(XXX)
DEF var100=(R//1)
DEF var101=(S//"Hallo")
DEF var102(1)=(V//"HUGO")
DEF TMP
VS8=("GC")
PRESS(VS8)
GC("NC1")
END_PRESS
...

OUTPUT(NC1)                                ; ;原始和其它新的版本
var100", "var101
END_OUTPUT
...

OUTPUT(NC1,1)                              ; 版本 1
var100", "var101", " var102
END_OUTPUT

```

2.7 功能

概述

在对话框和与对话框相关的软键栏中提供不同的功能，这些功能通过事件（例如：退出输入栏，按下软键）触发并在方法中设计。

子程序

重复的或者其它的设计指令，这些指令总结为一个特定的过程，可以在子程序中设计。子程序可以随时装载到主程序或者其它子程序中并随时进行编辑，即指令不必多次重复设计。作为主程序适用于对话框描述块或者软键栏。

PI 服务

通过功能 PI_SERVICE 可以在由 PLC 在 NC 区中启动 PI 服务（程序实例服务）。

外部功能（仅 HMI 高级）

借助于外部功能可以引入其它一些用户特定的功能。外部功能存放在一个 DLL 文件中并通过设计文件定义行中的条目识别。

参见

PI 服务列表 (页 206)

外部功能（仅 HMI 高级）(页 116)

2.7.1 主动程序（AP）

说明

功能 AP（主动程序）将一个文件从被动 HMI 文件系统传输到 NC 的主动文件系统中。将文件装载到 NC 中并许可，然后在 HMI 文件系统中删除。对于 HMI 内置 sl，该功能作用如同设置许可。

编程

句法：	AP("文件")
说明：	将被动 HMI 文件系统的文件传输至 NC 的主动文件系统
参数：	文件 要传输的 HMI 文件的完整路径数据

举例

```
//M(测试GC/"代码生成:")  
DEF VAR1 = (R//1)  
DEF VAR2 = (R//2)  
DEF D_NAME  
LOAD  
VAR1 = 123  
VAR2 = -6  
END_LOAD  
OUTPUT(CODE1)  
  "Cycle123(" VAR1 "," VAR2 ")"  
  "M30"  
END_OUTPUT  
PRESS(VS1)  
  D_NAME = "\MPF.DIR\MESSEN.MPF"  
  GC("CODE1",D_NAME) ; 将 OUTPUT 方法的代码写入文件  
                        \MPF.DIR\MESSEN.MPF 中  
END_PRESS  
PRESS(HS8)  
  D_NAME = "\MPF.DIR\MESSEN.MPF"  
  AP(D_NAME) ; 文件装载到 NC 中  
END_PRESS
```

2.7.2 定义块(//B)

说明

子程序在程序文件中用块标记 //B 标记并通过 //END 结束。
每个块标记可以定义多个子程序。

注意

必须在调用子程序的对话框中定义子程序所使用的变量。

编程

一个块有下列结构：

句法：	//B(块名称)	
	SUB(名称)	
	END_SUB	
	[SUB(名称)	
	...	
	END_SUB]	
	...	
	//END	
说明：	定义子程序	
参数：	块名称	块标记名称
	名称	子程序名称

举例

```
//B(PROG1) ; 块开始
SUB(UP1) ; 子程序开始
...
REG[0] = 5 ; 寄存器 0 赋值 5
...
END_SUB ; 子程序结束
SUB(UP2) ; 子程序开始
    IF VAR1.val=="Otto"
        VAR1.val="Hans"
    RETURN
ENDIF
    VAR1.val="Otto"
END_SUB ; 子程序结束
//END ; 块结束
```

2.7.3 子程序调用(CALL)

说明

通过 CALL 功能可以从方法的任意一个位置调用一个装载的子程序。
允许叠加，即由一个子程序调用另一个子程序。

编程

句法：	CALL("名称")
说明：	调用子程序
参数：	名称 子程序名称

举例

```
//M(屏幕窗口 1)
VAR1 = ...
VAR2 = ...
LOAD
...
LB("PROG1") ; 装载块
...
END_LOAD
CHANGE( )
...
CALL("UP1") ; 调用并编辑子程序
...
END_CHANGE
...
//END
```

2.7.4 检查变量 (CVAR)

说明

借助于功能
CVAR (检查变量) 可以查询对话框的所有变量或者仅特定变量或者辅助变量是否正确。
在用功能 GC 产生一个 NC 代码之前，查询变量是否包含一个有效值是必要的。
如果变量状态命名符.vld = 1，则变量正确，

2.7.5 复制程序 (CP)

说明

功能 CP (复制程序) 在 HMI 文件系统或者 NC 文件系统中复制文件。

注意

对于 NCU 上的 HMI 内置，只能在 NC 文件系统中复制。

编程

句法：	CP("源文件", "目标文件")	
说明：	文件：复制	
参数：	源文件	源文件完整的路径数据
	目标文件	目标文件完整的路径数据

举例

```
CP( "\MPF.DIR\CFI.MPF ", "\spf.dir\cfi.nc" )
```

2.7.6 对话框行(DLGL)

说明

在对话框的对话框行中可以根据确定的情况给出短文本 (信息或者输入帮助) 。

标准字体大小时允许的字符数量：

- HMI 内置 sl：约 50 个字符
- HMI 高级：约 100 个字符

编程

句法：	DLGL("字符串")	
说明：	显示对话框行的文本	
参数：	字符串	显示在对话框中的文本

举例

```
IF Var1 > Var2  
    DLGL("值太大!") ; 如果变量 1>变量 2，则对话框行中显示文本“值过大！”。  
ENDIF
```

2.7.7 删除程序 (DP)

说明

功能 DP (删除程序) 删除一个被动的 HMI 文件系统或者主动的 NC 文件系统的文件。

编程

句法 : DP("文件")
说明 : 文件 : 删除
参数 : 文件 要删除文件的完整路径数据

举例

```
DP( "\MPF.DIR\CFI.MPF" )
```

2.7.8 评估(EVAL)

说明

功能 EVAL 评估作出的输出结果，然后执行。为此可以首先在运行期间建立表达式。例如用于变量上的显示存取。

编程

句法： **EVAL**(*exp*)
 说明： 评估表达式
 参数： *exp* 逻辑表达式

举例

```

VAR1=(S)
VAR2=(S)
VAR3=(S)
VAR4=(S)
CHANGE(
  REG[7] = EVAL("VAR"<<REG[5])      ; 如果 REG[5] 的值为3，则括号中的表达式为
                                      VAR3。然后，分配 REG[7] VAR3 的值。
  IF REG[5] == 1
    REG[7] = VAR1
  ELSE
    IF REG[5] == 2
      REG[7] = VAR2
    ELSE
      IF REG[5] == 3
        REG[7] = VAR3
      ELSE
        IF REG[5] == 4
          REG[7] = VAR4
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF
END_CHANGE

```

2.7.9 执行(EXE)

说明

通过功能 EXE 可以在 HMI 高级上调用一个程序，该程序通过 HMI 高级的 OEM 包建立作为应用程序，或者调用自由轮廓编程。

在 HMI 内置 sl 上只能通过 EXE 启动自由轮廓编程。

注意

功能 EXE 仅提供用在零件程序编辑器中。为了启动程序，需要在应用程序 INI 文件中的 [CHILDS] 下面输入程序的任务索引，如存放在 REGIE.INI 中。

编程

句法： **EXE**(程序名称) ;HMI 高级
 EXE(GPROC) ;HMI 内置 sl
说明： 执行程序
参数： 程序名称 要执行的程序名称

举例

```
PRESS (VS3)
  EXE (GPROC)                            ;    启动 GPROC.EXE (自由轮廓编程)
END_PRESS
```


2.7.11 退出对话框(EXIT)

说明

通过功能 EXIT 可以退出对话框并返回主对话框。
如果不存在主对话框，则退出新配置的操作界面并返回标准应用程序。

编程 (没有参数)

句法 : **EXIT**
说明 : 退出对话框
参数 : - 无 -

举例

```
PRESS(HS1)  
EXIT  
END_PRESS
```

说明

如果调用带传输变量的当前对话框，则变量值改变并且返回初始对话框。
变量值分别分配给通过功能"LM"从初始对话框传输到后续对话框的变量。可以最多传输 20 个变量值，各自之间通过逗号隔开。

注意

变量/变量值的顺序必须根据 LM 功能的传递变量顺序进行，以此达到分配明确。
如果一些变量值没有规定，则这些变量不改变。改变的传输变量在功能 LM 后立即在初始对话框中生效。

使用传输变量编程

句法 : **EXIT**[(VARx)]
说明 : 退出对话框，传输一个或者多个变量
参数 : VARx 计算变量

举例

```
//M(屏幕窗口 1)
...
PRESS(HS1)
  LM("屏幕窗口 2","CFI.COM",1, POSX, POSY, 直径)
                                     ; 中断屏幕窗口 1 并显示屏幕窗口 2。传输变量
                                     POSX、POSY 和直径。
DLGL("退出屏幕窗口 2")             ; 从屏幕窗口 2 返回后在屏幕窗口 1
                                     的对话框行中显示文本：退出屏幕窗口 2。
END_PRESS
...
//END

//M(屏幕窗口 2)
...
PRESS(HS1)
  EXIT(5, , 计算_直径)
                                     ; 退出屏幕窗口 2 并返回到屏幕窗口 1 的 LM 后的行中。
                                     此时，分配值 5 给变量 POSX
                                     并分配变量“计算_直径”的值给变量“直径”。变量 POSY
                                     包含当前值。
END_PRESS
...
//END
```

2.7.12 退出装载软键(EXITLS)

说明

通过功能 EXITLS 可以离开当前的操作界面并装载一个定义的软键栏。

编程

句法： **EXITLS**("软键栏",["路径名"])
 说明： 退出时装载软键栏
 参数： 软键栏 待装载的软键栏名称
 路径名 要装载的软键栏目录路径

举例

```

PRESS(HS1)
    EXITLS( "软键栏 1", "AEDITOR.COM" )
END_PRESS
    
```

2.7.13 生成代码(GC)

说明

功能 GC (生成代码) 由 OUTPUT (输出) 方法生成 NC 代码。

编程

句法： **GC**("名称",["目标文件"][, *Opt*],[*Append*])
 说明： 生成 NC 代码
 参数： 名称 OUTPUT (输出) 块名称用作代码生成的基础
 目标文件 HMI 或者 NC 文件系统的目标文件路径数据。
 如果未规定目标文件 (只能在编程支持中) ，则在该位置
 上写入代码，在该位置处光标停留在当前打开的文件
 中。
 可选 可选注释生成
 0: (预设) 建立用于反编译的带有注释的预置代码。
 1: 在生成的代码中不生成注释。
 说明： 该代码无法反编译。

- Append 只有当规定目标文件情况下，该参数才有意义。
- 0: (预设) 当文件已存在情况下，删除旧的内容。
 - 1: 当文件已存在情况下，在文件开始处写入新的代码。
 - 2: 在文件已存在的情况下，在结束处加入新的代码。

举例

```
//M(测试GC/"代码生成:")
DEF VAR1 = (R//1)
DEF VAR2 = (R//2)
DEF D_NAME
LOAD
VAR1 = 123
VAR2 = -6
END_LOAD
OUTPUT(CODE1)
    "Cycle123(" VAR1 ", " VAR2 ") "
    "M30"
END_OUTPUT

PRESS(VS1)
    D_NAME = "MPF.DIR\MESSEN.MPF"
    GC("CODE1",D_NAME)                ; 将 OUTPUT 方法的 NC 代码写入文件
                                        C:\MPF.DIR\MESSEN.MPF 中
                                        Cycle123(123, -6)
                                        M30
END_PRESS
```

反编译

- **目标文件没有数据：**
功能 GC 只能在编程支持中使用并待当前编辑器中打开的文件内写入 NC 代码。NC 代码可以反编译。如果功能 GC 在没有目标文件数据情况下在“补充操作界面”下设计，则此时发出故障信息。
- **目标文件数据：**
从 OUTPUT (输出) 块生成的代码输入在目标文件中。 如果不存在目标文件，则在 NC 文件系统中设立。 如果目标文件在 HMI 文件系统中，则该文件存放在硬盘上 (仅 HMI 高级)。 不设立使用注释行 (反编译需要的信息)，即无法进行反编译。

目标文件的数据特点

基本上规定一个目标文件有三个不同的类型：

- **NC 格式**：/_N_MPF_DIR/_N_MY_FILE_MPF
只适用于 HMI 内置 sl。
该文件存放在 NC 上的目录 MPF 下。
- **DH 格式**：/MPF.DIR\MY_FILE.MPF
适用于 HMI 高级 和 HMI 内置 sl。
对于 HMI 内置 sl，目标文件的数据以 NC 格式转换并且文件存放在 NCU 上。
对于 HMI 高级或者对于 HMI 内置 WIN32，文件存放在数据管理路径下。
- **DOS 格式**：d:\abc\my_file.txt 或者 \\RemoteRechner\files\my_file.txt
适用于 HMI 高级 和 HMI 内置 sl。
文件写入在硬盘上规定的目录中或者指定的计算机上，此时硬盘上的目录或者远程计算机必须已存在。
对于 HMI 内置 sl，可以以该格式仅写入在 RAM 驱动器上或者网络中的一台计算机上，前提是已配置好网络连接。

注意

无效的变量在生成的 NC 代码中产生一个空字符串，如果读取该代码，在日志中出现一条出错信息。

反编译时的特殊情况

在子对话框中无法调用 GC 功能，因为在子对话框中可以使用源自主对话框的变量，然而在直接调用时并不存在。

在通过编辑器手动介入生成的代码时不允许改变由代码生成产生的值的符号数目。这些改变会妨碍到反编译。

解决方法：

1. 反编译
2. 通过设计的对话框输入修改（例如：99 → 101）
3. GC

参见

反编译 (页 107)

2.7.14 装载数组(LA)

说明

通过功能 LA (装载数组) 可以从另一个文件装载一个数组。

编程

句法 :	LA (名称 [, 文件])
说明 :	从文件装载数组
参数 :	名称 待装载数组的名称
	文件 在该文件中定义数组

注意

如果在当前的设计文件中，一个数组要由另一个设计文件中的数组替代，则数组名称必须相同。

举例

```

; 文件 maske.com 部分摘录
DEF VAR2 = (S/*ARR5/"关闭", "转换栏")
PRESS(HS5)
  LA("ARR5", "arrayext.com") ; 从文件 arrayext.com 中装载数组 ARR5
  VAR2 = ARR5[0] ; 代替 "Aus"/"Ein" ("关"/"开") 显示在 VAR2 的转换栏
; "上"/"下"/"右"/"左"
END_PRESS
//A(ARR5)
("关"/"开")
//END
; 文件 arrayext.com 部分摘录
//A(ARR5)
("上"/"下"/"右"/"左")
//END

```

注意

请注意，在通过 LA 功能分配给变量转换栏另一个数组后，必须赋予一个有效值给变量。

2.7.15 装载块 (LB)

说明

通过功能 LB (装载块) 可以在运行期间内装载带有子程序的块。首先 LB 要在 LOAD (装载) 方法中设计，由此可以随时调用装载的子程序。

注意

子程序也可以直接在对话框中定义，如果那样则不必装载。

编程

句法：	LB ("块名称", "文件")
说明：	运行时装载子程序
参数：	块名称 块标记名称
	文件 设计文件的路径数据
	预设 = 当前的设计文件

举例

```
LOAD
LB( "PROG1" ) ; 在当前的设计文件中查找块"PROG1"，接着装载块。
  LB( "PROG2", "XY.COM" ) ; 在设计文件 XY.COM 中查找块"PROG2"，接着装载块。
END_LOAD
```

2.7.16 装载屏幕窗口 (LM)

说明

通过功能 LM 可以载入一个新的对话框。

主对话框/子对话框

能够调用其他对话框并且不能自行结束的对话框称为主对话框。
由主对话框调用的对话框是子对话框。

编程

句法 :	LM ("名称"[, "文件"][, MSx [, VARx]])												
说明 :	载入对话框												
参数 :	<table border="0"> <tr> <td>名称</td> <td>待装载对话框的名称</td> </tr> <tr> <td>文件</td> <td>设计文件的路径数据 (HMI 文件系统或者 NC 文件系统) ; 标准设置 : 当前的设计文件</td> </tr> <tr> <td>MSx</td> <td>对话框切换模式</td> </tr> <tr> <td></td> <td>0: (预设) 退出当前对话框 , 装载并显示新的对话框。在 EXIT 时 (参见“退出”功能) 返回到标准应用程序。 通过参数 MSx 可以确定在对话框切换时是否退出当前对话框。 保留当前的对话框 , 可以将变量传递到新的对话框中。 使用参数 MSx 的优点在于 , 在切换时无需总是重新初始化对话框 , 而是保留当前对话框的数据和设计 , 简化数据传输</td> </tr> <tr> <td></td> <td>1: 自功能 LM 起中断当前对话框 , 装载并显示新的对话框。 退出时关闭子对话框并且返回到主对话框的中断位置。 中断时 , 不处理主对话框中的 UNLOAD (卸载) 块。</td> </tr> <tr> <td>VARx</td> <td>前提条件 : MS1 列出可以从主对话框传送到子对话框的变量。 可以最多传输 20 个变量 , 各自之间通过逗号隔开。</td> </tr> </table>	名称	待装载对话框的名称	文件	设计文件的路径数据 (HMI 文件系统或者 NC 文件系统) ; 标准设置 : 当前的设计文件	MSx	对话框切换模式		0: (预设) 退出当前对话框 , 装载并显示新的对话框。在 EXIT 时 (参见“退出”功能) 返回到标准应用程序。 通过参数 MSx 可以确定在对话框切换时是否退出当前对话框。 保留当前的对话框 , 可以将变量传递到新的对话框中。 使用参数 MSx 的优点在于 , 在切换时无需总是重新初始化对话框 , 而是保留当前对话框的数据和设计 , 简化数据传输		1: 自功能 LM 起中断当前对话框 , 装载并显示新的对话框。 退出时关闭子对话框并且返回到主对话框的中断位置。 中断时 , 不处理主对话框中的 UNLOAD (卸载) 块。	VARx	前提条件 : MS1 列出可以从主对话框传送到子对话框的变量。 可以最多传输 20 个变量 , 各自之间通过逗号隔开。
名称	待装载对话框的名称												
文件	设计文件的路径数据 (HMI 文件系统或者 NC 文件系统) ; 标准设置 : 当前的设计文件												
MSx	对话框切换模式												
	0: (预设) 退出当前对话框 , 装载并显示新的对话框。在 EXIT 时 (参见“退出”功能) 返回到标准应用程序。 通过参数 MSx 可以确定在对话框切换时是否退出当前对话框。 保留当前的对话框 , 可以将变量传递到新的对话框中。 使用参数 MSx 的优点在于 , 在切换时无需总是重新初始化对话框 , 而是保留当前对话框的数据和设计 , 简化数据传输												
	1: 自功能 LM 起中断当前对话框 , 装载并显示新的对话框。 退出时关闭子对话框并且返回到主对话框的中断位置。 中断时 , 不处理主对话框中的 UNLOAD (卸载) 块。												
VARx	前提条件 : MS1 列出可以从主对话框传送到子对话框的变量。 可以最多传输 20 个变量 , 各自之间通过逗号隔开。												

注意

参数 VARx 总是只传递变量值 , 也就是说变量可以在子对话框中读写 , 但是不可见。
由子对话框向主对话框返回传递变量可以通过功能 EXIT (退出) 进行。

举例

```

PRESS(HS1)
  LM("屏幕窗口 2","CFI.COM",1, POSX, POSY, 直径)
                                ; 中断屏幕窗口 1 并显示屏幕窗口 2：传输变量
                                POSX、POSY 和直径。
DLGL("退出屏幕窗口 2")      ; 从屏幕窗口 2 返回后在屏幕窗口 1 的对话框行中显示文本：
                                退出屏幕窗口 2。
END_PRESS

```

2.7.17 装载软键(LS)

说明

通过功能 LS 可以显示另一个软键栏。

编程

句法：**LS("名称"[, "文件"][, 合并])**

说明：显示软键栏

参数：

名称	软键栏名称
文件	设计文件的路径数据 (HMI 文件系统或者 NC 文件系统)
	预设置：当前的设计文件

Merge

0: 删除所有存在的软键，输入重新设计的软键。

1: (预设) 仅重新设计的软键覆盖存在的软键，而其他软键 (= HMI 或者 ShopMill/ShopTurn 的标准软键) 的功能和文本保持不变。

举例

```

PRESS(HS4)
  LS("软键栏 2",,0)          ; 软键栏 2 覆写存在的软键栏，删除所有存在的 软键
END_PRESS

```

注意事项

只要编译器还没有显示出对话框，即还没有处理 LM 功能，则在登入软键描述块和软键栏描述块的 PRESS (按压) 方法中总是仅设计一个 LS 命令或者 LM 命令，并且没有其它动作。

功能 LS 和 LM 仅允许在软键 PRESS (按压) 块中调用，然而不作为导航键 (PU、PD、SL、SR、SU、SD) 上的反应。

参见

登入软键的功能 (页 71)

2.7.18 被动程序 (PP)

说明

功能 PP (被动程序) 将一个文件从 NC 的主动文件系统传输到 HMI 高级的被动文件系统中。文件在执行功能 PP 后不再存在于 NC 的主动文件系统中。对于 HMI 内置 sl，该功能作用起效，如删除使能。

编程

句法： PP("文件")
说明： 将文件从 NC 的主动文件系统中传送到 HMI 高级的被动文件系统中。
参数： 文件 要传输的 NC 文件的完整路径数据

举例

```
PP( "\MPF.DIR\MESSEN.MPF" )
```

2.7.19 读取 NC PLC (RNP), 写入 NC PLC (WNP)

说明

通过指令 RNP (读取 NC PLC) 可以读取 NC 或者 PLC 变量或者机床数据。

编程

句法：**RNP** ("系统或者用户变量",值)
 说明：读取 PLC 或者 NC 变量或者机床数据
 参数：系统或者用户变量 NC 或者 PLC 变量名称
 值 要写入系统或者用户变量中的值。
 如果值类型为字符串，则必须以双引号括起。

举例

```
VAR2=RNP( "$AA_IN[ 2 ] " ) ; 读取 NC 变量
```

说明

通过指令 WNP (写入 NC PLC) 可以写入 NC 或者 PLC 变量或者机床数据。

在每次处理功能 WNP 都重新存取 NC 变量或 PLC 变量，即：总是在 CHANGE 方法下存取 NC 变量或 PLC 变量。这在系统或者用户变量经常改变值情况下有意义。如果只进行一次 NC/PLC 存取，则必须在 LOAD (装载) 或者 UNLOAD (卸载) 方法下设计。

编程

句法：**WNP** ("系统或者用户变量", 值)
 说明：写入 PLC 或者 NC 变量或者机床数据
 参数：系统或者用户变量 NC 或者 PLC 变量名称
 值 要写入系统或者用户变量中的值。
 如果值类型为字符串，则必须以双引号括起。

举例

```
WNP( "DB20.DBB1 " , 1 ) ; 写入 PLC 变量
```

2.7.20 多次读取 NC PLC (MRNP)

说明

用命令 MRNP 可以通过在寄存器中一次存取输入多个系统或者 BTSS 变量。这种存取比通过单个存取速度明显加快。系统或者 BTSS 变量必须在由自身区域的 MRNP 命令中。

系统或者 BTSS 变量区域分为如下几种：

- 一般 NC 数据 (\$MN..., \$SN..., /nck/...)
- 通道专用的 NC 数据 (\$MC..., \$SC..., /channel/...)
- PLC 数据 (DB..., MB..., /plc/...)
- 相同轴专用的 NC 数据 (\$MA..., \$SA...)

编程

句法：**MRNP (变量名称 1*变量名称 2[* ...], 寄存器编号)**

说明：读取多个变量

参数：变量名称用“*”作为分隔符。按照命令中变量名称的顺序，采用寄存器 REG[寄存器编号] 中的值和下列值。

关系到：

第一个变量的值在 REG[寄存器编号] 中。

第二个变量的值在 REG[寄存器编号 + 1]或

注意事项

必须注意，变量列表最多为 500 行，并且寄存器的数量受限。

举例

MRNP (" \$R[0]*\$R[1]*\$R[2]*\$R[3]", 1)	;	REG[1] 至 REG[4] 以变量值 \$R[0] 至 \$R[3] 描述。
--	---	--

读取显示机床数据：

显示机床数据通过 RNP (\$MM...) 在 LOAD (装载) 块中读取。

显示机床数据上一般的读写访问不通过“补充操作界面”进行。

注意

用户变量不允许与系统变量或者 PLC 变量有相同的名称。

NC 变量

提供所有的机床和设置数据以及 R 参数，但是仅提供某些系统变量（参见附录列表）。在 HMI 高级中，在操作范围参数/系统变量/编辑视图/插入变量中找到可获取的系统变量。

可访问所有全局的和通道专用的用户变量（GUD）。本地和程序全局的用户变量无法编辑。

机床数据	
全局机床数据	\$MN_...
轴专用的机床数据	\$MA_...
通道专用的机床数据	\$MC_...

设定数据	
全局设定数据	\$SN_...
轴专用的设定数据	\$SA_...
通道专用的设定数据	\$SC_...

系统变量	
R 参数 1	\$R[1]

PLC 变量

提供所有 PLC 数据。

PLC 数据	
数据模块 x 的字节 y 位 z	DBx.DBXy.z
数据模块 x 的字节 y	DBx.DBBy
数据模块 x 的字 y	DBx.DBWy
数据模块 x 的双字 y	DBx.DBDy
数据模块 x 的实数 y	DBx.DBRy
标志器字节 x 位 y	Mx.y
标志器字节 x	MBx
标志器字 x	MWx
标志器双字 x	MDx
输入字节 x 位 y	Ix.y 或者 Ex.y
输入字节 x	IBx 或者 EBx
输入字 x	IWx 或者 EWx

PLC 数据	
输入双字 x	IDx 或者 EDx
输出字节 x 位 y	Qx.y 或者 Ax.y
输出字节 x	QBx 或者 ABx
输出字 x	QWx 或者 AWx
输出双字 x	QDx 或者 ADx
数据模块 x 中带有长度 z 的字符串 y	DBx.DBsy.z

2.7.21 刷新

说明

在所有块内可以调用功能“刷新”。该功能没有参数。

生效方式：

- 所有显示区中激活的变量内容（输入/输出栏）给定新的背景和前景。
- 所有显示区中可见的激活的短文本，图像文本和单位文本重新给定，之前可不删除该文本的背景。

编程

句法：**刷新**
 说明：刷新输入//输出栏和输入文本
 参数：- 无 -

2.7.22 寄存器(REG)

寄存器说明

寄存器用于在两个不同的对话框之间切换数据。
寄存器分配给每个对话框并且在载入第一个对话框时生成，以 0 或空字符串预占。

注意

寄存器不允许直接用在 OUTPUT (输出) 块中用于 NC 代码生成。

编程

句法 : **REG**[*x*]
说明 : 定义寄存器
参数 : *x* 寄存器编号，以 $x = 0 \dots 19$
 类型：REAL 或者 STRING = VARIANT
 $x \geq 20$ 的寄存器由西门子使用。

寄存器值说明

寄存器值的分配在方法中设计。

注意

如果由一个对话框通过功能 LM 生成另一个对话框，则寄存器的内容自动接受入新的对话框中并在第二个对话框中用于其它计算。

编程

句法 : *名称.val = 寄存器值*
 或者
 名称 = 寄存器值
说明 :
参数 : 名称 寄存器名称
 寄存器值 寄存器的值

举例

```
UNLOAD
    REG[0] = VAR1                ; 寄存器 0 赋值为变量 1 的值
END_UNLOAD

UNLOAD
    REG[9].VAL = 84              ; 寄存器 9 赋值为变量 84 的值
END_UNLOAD
                                ; 在下列对话框中该寄存器可以在方法中重新分配本地变
                                ; 量。

LOAD
    VAR2 = REG[0]
END_LOAD
```

寄存器状态说明

通过状态特性可以在设计中查询寄存器是否包含一个有效值。

此外，当一个对话框作为主对话框使用时，可以使用寄存器的状态查询向一个寄存器中仅写入一个值。

编程

句法： 名称.vld
说明： 该属性仅可读。
参数： 名称 寄存器名称
返回值： 查询结果可能是：
 FALSE = 无效值
 TRUE = 有效值

举例

```
IF REG[15].VLD == FALSE        ; 查询寄存器值的有效性
    REG[15] = 84
ENDIF
VAR1 = REG[9].VLD              ; 向 Var1 分配 REG[9] 状态查询的值。
```

2.7.23 RETURN

说明

通过 RETURN 功能可以提前取消当前的子程序处理并返回到最后一次 CALL 命令的跳转位置。

如果子程序中没有设计 RETURN，则执行子程序，直至结束然后返回到跳转位置。

编程

句法： **RETURN**
说明： 返回跳转位置
参数： - 无 -

举例

```
//B(PROG1)                               ; 块开始
SUB(UP2)                                 ; 子程序开始
    IF VAR1.val=="Otto"
        VAR1.val="Hans"
RETURN                                   ; 如果变量值 = Otto，则分配值 "Hans" 给变量，
                                         子程序在此位置结束。
ENDIF
    VAR1.val="Otto"                       ; 如果变量值 ≠ Otto，则分配值 "Otto" 给变量。
END_SUB                                 ; 子程序结束
//END                                    ; 块结束
```

2.7.24 反编译

说明

在编程支持中可以反编译功能 GC 生成的 NC 代码并在所属输入对话框中的输入/输出栏中再次显示变量值。

编程

来自 NC 代码的变量接受至对话框中。此时对来自 NC 代码的变量值与来自设计文件的计算的变量值加以比较。如果存在不一致，则在日志文件中给出错误信息，因为在生成的 NC 代码中数值已改变。

如果一个变量在 NC 代码中多次存在，则在反编译时总是分析该变量最后一次出现的值。
另外在日志文件中给出警告。

代码生成时，不在 NC 代码中使用的变量作为使用注释存储。
通过使用注释标记所有反编译时需要的信息。使用注释不得更改。

注意

NC 代码块和使用注释如果在一行的起始处开始，则只能反编译。

举例：

在程序中有下列 NC 代码：

```
DEF VAR1 = (I//101)
OUTPUT(CODE1)
  "X" VAR1 " Y200"
  "X" VAR1 " Y200"
END_OUTPUT
```

在零件程序中存放下列代码：

```
;NCG#TestGC#\cus.dir\aeditor.com#CODE1#1#3#
X101 Y200
X101 Y0
;#END#
```

在反编译时读取编辑器：

```
X101 Y200
X222 Y0 ; 在零件程序中更改 X 的值 (X101 → X222)
```

在输入对话框中给出下列 VAR1 值： VAR1 = 222

参见

生成代码(GC) (页 93)

2.7.25 向前/后查找(SF, SB)

说明

通过功能 **向前/后查找 (SF, SB)** 可以在编辑器当前的 NC 程序中从当前光标位置开始查找某个字符串并给出其值。

编程

句法 :	SF ("字符串")
名称 :	Search Forward : 从当前光标位置向前查找
句法 :	SB ("字符串")
名称 :	Search Backward : 从当前光标位置向后查找
参数 :	字符串 要查找的文本

查找规则 :

- 要查找的字符串和数值的单元在 NC 程序中前后必须有空格。
- 查找对象无法在注释中和字符串内查找。
- 给出的值必须是一个数字表达式，表达式形式“X1=4+5”无法识别。
- 可识别十六进制常数的形式 X1='HFFFF'、二进制常数的形式 X1='B10010' 和指数常数的形式 X1='-5EX-4'。
- 在字符串和数值之间有下列符号，可以给出字符串值：
 - 无
 - 空格键
 - 等号

举例

允许以下的写入方式：

X100 Y200	;	变量 Abc 包含数值 200
Abc = SB("Y")		
X100 Y 200	;	变量 Abc 包含数值 200
Abc = SB("Y")		
X100 Y=200	;	变量 Abc 包含数值 200
Abc = SB("Y")		

2.7.26 选择程序 (SP)

说明

功能 SP (选择程序) 选择一个主动的 NC 文件系统文件，以对其进行处理。即该文件之前必须就已装载在 NC 中。

编程

句法： **SP("文件")**
 名称： **Select Program**
 参数： “文件” NC 文件的完整路径数据

举例

```
//M(测试GC/"代码生成:")
DEF VAR1 = (R//1)
DEF VAR2 = (R//2)
DEF D_NAME
LOAD
  VAR1 = 123
  VAR2 = -6
END_LOAD
OUTPUT(CODE1)
  "Cycle123(" VAR1 "," VAR2 ") "
  "M30"
END_OUTPUT
PRESS(VS1)
  D_NAME = "\MPF.DIR\MESSEN.MPF"
  GC("CODE1",D_NAME) ; 将 OUTPUT 方法的代码写入文件
                       \MPF.DIR\MESSEN.MPF 中
END_PRESS
PRESS(HS8)
  AP("\MPF.DIR\MESSEN.MPF") ; 文件装载到 NC 中
  SP("\MPF.DIR\MESSEN.MPF") ; 文件选择
END_PRESS
```

2.7.27 字符串功能

概述

以下的功能允许进行字符串处理：

- 确定字符串长度
- 查找字符串中的一个字符
- 由左提取部分字符串
- 由右提取部分字符串
- 由字符串中间提取部分字符串
- 替换部分字符串

功能 LEN：字符串长度

句法：	LEN(字符串 / 变量名称)	
说明：	确定一个字符串的字符数目。	
参数：	字符串	每个有效的字符串表达式。 对于一个空字符串，返回为零。
	变量名称	每个有效的和表示的变量名称 仅允许两个可行参数中的一个。

举例

```
DEF VAR01  
DEF VAR02  
  
LOAD  
VAR01="HALLO"  
VAR02=LEN(VAR01) ; 结果 = 5  
END_LOAD
```

功能 INSTR : 查找字符串中的字符

句法 :	INSTR (开始, 字符串 1, 字符串 2 [,方向])	
说明 :	查找字符	
参数 :	启动	从字符串 1 向字符串 2 查找的开始位置。 如果从字符串 2 的开头开始查找, 则指定为 0。
	字符串 1	要查找的字符。
	字符串 2	在该字符串链中查找
	方向 (可选)	查找的方向
		0: 从左向右 (预设)
		1: 从右到左
	如果字符串 2 中没有包含 字符串 1 , 则返回数值 0。	

举例

```

DEF VAR01
DEF VAR02

LOAD
VAR01="HALLO/WELT"
VAR02=INST(1,"/",VAR01)           ;   结果 = 6
END_LOAD

```

功能 LEFT : 左边字符串

句法 :	LEFT (字符串, 长度)	
说明 :	LEFT 返回一个字符串, 该字符串从字符串左侧开始包含指定的字符数。	
参数 :	字符串	字符串或者带有要处理的字符串的变量
	长度	要读取的字符数目

举例

```

DEF VAR01
DEF VAR02

LOAD
VAR01="HALLO/WELT"
VAR02=LEFT(VAR01,5)             ;   结果 = "HALLO"
END_LOAD

```

功能 RIGHT : 右边字符串

句法 :	RIGHT (字符串, 长度)	
说明 :	RIGHT 返回一个字符串链, 该字符串链从字符串右侧开始包含指定的字符数。	
参数 :	字符串	字符串或者带有要处理的字符串的变量
	长度	要读取的字符数目

举例

```

DEF VAR01
DEF VAR02
LOAD
VAR01="HALLO/WELT"
    VAR02=LEFT(VAR01,4)                ; 结果 = "WELT"
END_LOAD

```

功能 MIDS : 中间字符串

句法 :	MIDS (字符串, 开始 [, 长度])	
说明 :	MIDS 返回一个字符串, 该字符串从字符串的指定位置开始包含指定的字符数。	
参数 :	字符串	字符串或者带有要处理的字符串的变量
	开始	开始, 从字符串链中该处开始读取
	长度	要读取的字符数目

举例

```

DEF VAR01
DEF VAR02
LOAD
VAR01="HALLO/WELT"
    VAR02=LEFT(VAR01,4,4)            ; 结果 = "LO/W"
END_LOAD

```

功能 REPLACE : 替换字符

句法 : **REPLACE** (字符串, 查找字符串, 替换字符串 [, 开始 [, 计数]])

说明 : 功能 REPLACE
用另一个字符/字符链替代字符串中的一个字符/字符链。

参数 :

字符串	待通过替换字符串替换查找字符串的字符串。
查找字符串	需被替代的字符串
替代字符串	替代字符串 (位于查找字符串位置)
开始	查找和替换的开始位置
计数器	从开始位置起要开始查找的查找字符串的字符数量。

返回值 :

字符串 = 空字符串	复制字符串
查找字符串 = 空字符串	复制字符串
替换字符串 = 空字符串	复制字符串, 在该字符串中删除所有出现的查找字符串
开始 > Len(长度)	空字符串
计数 = 0	复制字符串

2.7.28 PI 服务

说明

通过功能 PI_SERVICE 可以在由 PLC 在 NC 区中启动程序实例服务 (PI 服务)。

一般编程

句法：	PI_SERVICE (<i>服务</i> , <i>n 参数</i>)	
说明：	执行 PI 服务	
参数：	服务	PI 服务的标识
	<i>n 参数</i>	<i>n</i> 个 PI 服务参数组成的参数列表。 参数用逗号隔开。

举例

```

PRESS (HS2)
  PI_SERVICE( "_N_CREATO" , 55 )
END_PRESS
PRESS(VS4)
  PI_SERVICE( "_N_CRCEDN" , 17 , 3 )
END_PRESS

```

启动 OEM 服务

命令 PI_START 根据 OEM 文献执行一个 PI 服务。

编程

句法：	PI_START ("传输字符串")	
说明：	执行 PI 服务	
参数：	"传输字符串"	传输的字符串与 OEM 文献相反，应用双引号括起。

举例

```

PI_START( "/NC,001,_N_LOGOUT" )

```

注意

通道相关的 PI 服务总是与当前的通道有关。

刀具功能 (TO 区) 的 PI 服务总是以分配到当前通道的 TO 区为参考。

参见

PI 服务列表 (页 206)

2.7.29 外部功能 (仅 HMI 高级)

说明

通过调用可以使用更多用户特定的功能。外部功能存放在一个 DLL 文件中并通过设计文件定义行中的条目识别。

注意

一个外部功能必须至少有一个返回参数。

编程

句法 : **FCT 功能名称 =**
(“文件”/返回类型/固定调用参数的类型/可变调用参数的类型)

说明 : 定义其他外部功能

参数 :	功能名称	外部功能的名称
	文件	DLL 文件的完整路径数据
	返回类型	功能返回值数据类型
	R, I, S, C, B.	固定 调用参数的数据类型和返回值。 数据类型用逗号隔开。
	变量或者寄存器	可变 调用参数的数据类型

句法 : **FCT 功能名称 (调用参数)**
调用参数 所有调用参数列表。参数用逗号隔开。

举例

```
//M(屏幕窗口 1)
DEF VAR1 = (R)
DEF VAR2 = (I)
DEF RET = (I)
FCT InitConnection = ("c:\user\mydll.dll"/I/R,I,S/I,S)
                        ; 定义外部功能"InitConnection"。定义的外部功能的数据类型。
                        ; 返回值是整数，固定调用参数是实数、整数和字符串，可变调用参
                        ; 数是整数和字符串。
LOAD
  RET = InitConnection(VAR1+SIN(VAR3),13,"Servus",VAR2,
  REG[2])
                        ; 将带调用参数 VAR1+SIN(VAR3), 13, Servus, VAR2 和 REG[2]
                        ; 的外部功能 "InitConnection" 的值分配给 RET。
END_LOAD
```

DLL 文件中的摘录

```
void __export WINAPI InitConnection(ExtFctStructPtr FctRet, ExtFctStructPtr
FctPar, char cNrFctPar)
FctRet->value.i  功能的返回值
FctPar[0]->value.r  第 1 个参数 (VAR1+SIN(VAR3))
FctPar[1]->value.i  第 2 个参数(13)
FctPar[2]->value.s  第 3 个参数 ("Servus")
FctPar[4]->value.i  第 4 个参数 (参考 VAR2)
FctPar[5]->value.s  第 5 个参数 (参考 REG[2])
cNrFctPar  参数数量 (5)
```

2.7.30 编程举例

编程

```

//S(Start)
HS7=("举例", se1, ac7)
PRESS(HS7)
    LM("屏幕窗口 4")
END_PRESS
//END
//M(屏幕窗口 4/"举例 4: 机床操作面板"/"MST.BMP")
DEF byte=(I/0/0/"输入字节, 标准
=0", "字节号.:", ""/wr1,li1//380,40,100/480,40,50)
DEF Feed=(IBB//0/"", "进给倍率", ""/wr1//"EB3"/20,180,100/130,180,100),
Axistop=(B//0/"", "进给停止", ""/wr1//"E2.2"/280,180,100/380,180,50/0,11)
DEF Spin=(IBB//0/"", "主轴倍率", ""/wr1//"EB0"/20,210,100/130,210,100),
spinstop=(B//0/"", "主轴停止", ""/wr1//"E2.4"/280,210,100/380,210,50/0,11
)
DEF custom1=(IBB//0/"", "用户定义键
1", ""/wr1//"EB6"/20,240,100/130,240,100)
DEF custom2=(IBB//0/"", "用户定义键
2", ""/wr1//"EB7"/20,270,100/130,270,100)
DEF By1
DEF By2
DEF By3
DEF By6
DEF By7

HS1=("输入字节", SE1, AC4)
HS2=("")
HS3=("")
HS4=("")
HS5=("")
HS6=("")
HS7=("")
HS8=("")
VS1=("")
VS2=("")
VS3=("")
VS4=("")
VS5=("")
VS6=("")
VS7=("")
VS8=("OK", SE1, AC7)

```

```

LOAD
    By1=1
    By2=2
    By3=3
    By6=6
    By7=7
END_LOAD

PRESS(HS1)
    Byte.wr=2
END_PRESS

CHANGE(Byte)
    By1=byte+1
    By2=byte+2
    By3=byte+3
    By6=byte+6
    By7=byte+7
    Feed.VAR="EB"<<By3
    Spin.VAR="EB"<<Byte
    Custom1.VAR="EB"<<By6
    Custom2.VAR="EB"<<By7
    Axisstop.VAR="E"<<By2<<".2"
    Spinstop.VAR="E"<<By2<<".4"
    Byte.wr=1
END_CHANGE

CHANGE(Axis stop)
    IF Axistop==0
        Axistop.BC=9
    ELSE
        Axistop.BC=11
    ENDIF
END_CHANGE

CHANGE(Spin stop)
    IF Spinstop==0
        Spinstop.BC=9
    ELSE
        Spinstop.BC=11
    ENDIF
END_CHANGE

```

```
PRESS(VS8)  
EXIT  
END_PRESS
```

结果

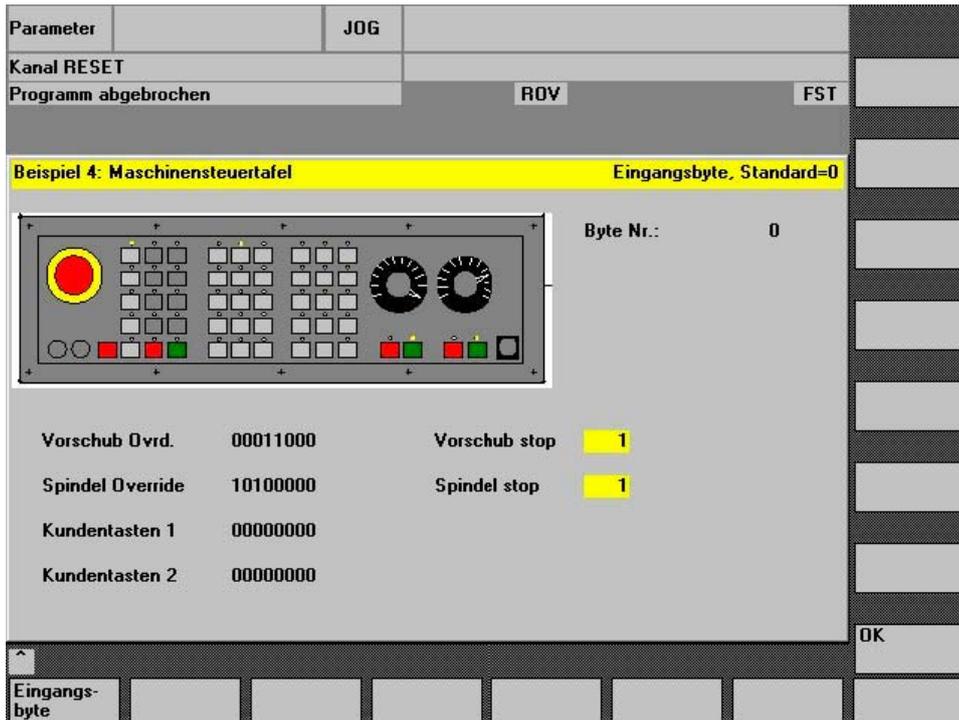


图 2-10 (机床控制面板)

注意

其它举例可在随附的工具箱中找到。

2.8 运算符

概述

在编程时可以使用以下几种运算符：

- 数学运算符
- 比较运算符
- 逻辑（布尔）运算符
- 位运算符
- 三角函数功能

2.8.1 数学运算符

概述

数学运算符	名称
+	加法
-	减法
*	乘法
/	除法
MOD	模数运算
()	括号
AND	和运算符
OR	或运算符
NOT	非运算符
ROUND	带有小数位的数四舍五入

举例： `VAR1.VAL = 45 * (4 + 3)`

ROUND

在执行对话框设计时，用 ROUND 指令把数值四舍五入直至小数点后面的 12 位。小数点后位置在变量栏不能显示。

使用

ROUND 通过用户的两个参数来控制：

`VAR1 = 5,2328543`

`VAR2 = ROUND(VAR1, 4)`

结果： `VAR2 = 5,2339`

VAR1 包含在四舍五入的数中。参数"4"给出了结果中的小数点后的位数，保存在 VAR2 中。

三角函数功能

三角函数功能	名称
SIN(x)	正弦 x
COS(x)	余弦 x
TAN(x)	正切 x
ATAN(x, y)	反正切 x/y
SQRT(x)	平方根 x
ABS(x)	绝对值 x
SDEG(x)	换算为度数
SRAD(x)	换算为弧度

注意

这些函数处理弧度。为了换算，可以使用函数 SDEG() 和 SRAD()。

举例: `VAR1.VAL = SQRT(2)`

常量

常量	
PI	3.14159265358979323846
FALSE	0
TRUE	1

举例: `VAR1.VAL = PI`

比较运算符

比较运算符	
==	相等
<>	不等
>	大于
<	小于
>=	大于等于
<=	小于等于

举例：

```
IF VAR1.VAL == 1
    VAR2.VAL = TRUE
ENDIF
```

条件

层叠深度没有限制。

带有一个命令的条件：	IF
	...
	ENDIF
带有两个命令的条件：	IF
	...
	ELSE
	...
	ENDIF

2.8.2 位运算符

概述

位运算符	名称
BOR	位方式 OR
BXOR	位方式 XOR
BAND	位方式 AND
BNOT	位方式 NOT
SHL	位向左移动
SHR	位向右移动

运算符 SHL

通过运算符 SHL(SHIFT LEFT)向左移动位。
 此时要移动的值或者移动步数可以直接规定或者作为变量规定。
 如果达到数据格式极限，则已超出位，没有出错信息。

使用

句法：	变量 = 值 SHL 步数
说明：	向左移动
参数：	值 要移动的值
	步数 移动步数

举例

```
PRESS(VS1)
VAR01 = 16 SHL 2           ; 结果 = 64
VAR02 = VAR02 SHL VAR04   ; VAR02 内容转换为 32 位格式，无正负之分;并且位向左移动
                           ; VAR04 内容。接着 32 位值重新转换回变量 VAR02
                           ; 的格式。
END_PRESS
```

运算符 SHR

通过运算符 SHR (SHIFT RIGHT)向右移动位。
此时要移动的值或者移动步数可以直接规定或者作为变量规定。
如果达到数据格式极限，则已超出位，没有出错信息。

使用

- 句法： 变量 = 值 SHR 步数
- 说明： 向右移动
- 参数： 值 要移动的值
- 步数 移动步数

举例

```
PRESS(VS1)
VAR01 = 16 SHR 2           ; 结果 = 4
VAR02 = VAR02 SHR VAR04   ; VAR02 内容转换为 32
                           ; 位格式，无正负之分并且位向右移动 VAR04 内容。接着
                           ; 32 位值重新转换回变量 VAR02 的格式。
END_PRESS
```

编程支持

3.1 编程支持提供什么？

概述

在用 ASCII 编辑器建立 NC 程序时编程支持可以辅助编程人员开展工作。在编程支持下也可以设计自身的操作界面。创建操作界面通过 ASCII 编辑器和方法“补充操作界面”工具进行。

为此提供下列标准化工具：

- 循环辅助
- 自由轮廓编程
- 轮廓编程
- 反编译
- 模拟

注意

循环支持 (//C...) 出于兼容性的原因另外通过以前的描述语言支持，并且不用“补充操作界面”语句。

建立新对话框

原则上，要借助于“补充操作界面”工具设计新的操作界面。在编程支持下现有的另外一些区别在本章节中有相关描述。

设计文件

编程支持的新对话框的描述在设计文件 AEDITOR.COM 中定义。

- 新建立的对话框可以在编辑器基本画面中通过 5 个登入软键 (水平软键 2、3、4、5、6) 显示。
- 水平软键 2 至 5 预设有标准化的“轮廓”、“钻削”、“铣削”和“车削”。
- 水平软键 14 和 15 (扩展栏的软键 6 和 7) 预设为“车削测量”和“铣削测量”。

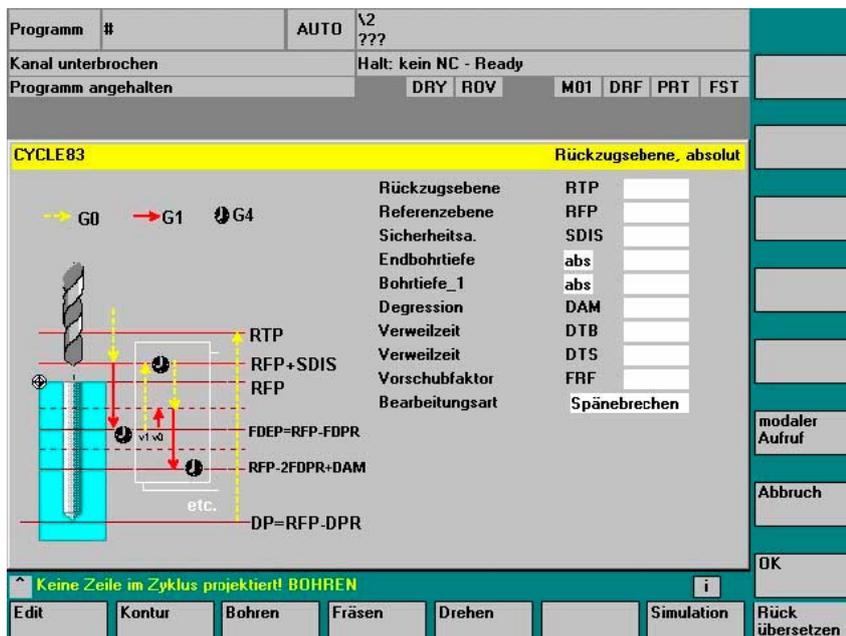


图 3-1 预设置下的视图

3.2 循环辅助

使用

在使用循环支持时在零件程序中循环调用的前后另外建立注释行，这些注释行用于反编译。该行以;# ...作为开始。

对于直接用编辑器记录的循环调用（也称作“旧”循环），这些信息缺失。在下列中描述，如;#-行包含的循环信息可以通过设计 INI 文件制备。

对于西门子循环，在供货范围中包含所需的设计文件。

为此可以在反编译后通过对话框支持用循环调用处理以下情况：

- 直接记录的循环调用
- 西门子循环调用
- 用户循环调用

前提条件

对于该循环，必须存在或/建立一个使用“补充操作界面”工具而创建的设计。

对于用于用户循环支持的、通过“补充操作界面”建立的对话框可以在用于此过程的对话框说明文件中设置这些信息。

确定是否要为了循环在零件程序中用 ;#-行处理或者通过设计文件反编译，可以根据循环在设计时进行。

用这种方式可以在零件程序中以 ;#-行选择性地建立用于直接记录的“旧”循环，用于对话框支持所需的扩展：

设计

文件 WIZARD.INI 可以保存在以下目录中。按照此顺序查找目录：

```
..\user
..\loem
..\hmi_adv
..\mmc2
```

以下的条目参考循环软件包，例如：

```
INI_1=bohren.ini
INI_2=drehen.ini
...
INI_n=paket.ini
```

现在合并章节[MMC_CycleWizard]的信息。

在每个 paket.ini 文件中有以下形式的循环特定的条目：

```
[cycleName]           ; 循环名称作为自身段
Mname=                ; 必须设置   对话框名称
Dir=                  ; 必须设置   包含对话框信息的文件的目录
Dname=                ; 必须设置   包含对话框信息的文件
Output=               ; 必须设置   OUTPUT 块的名称
Anzp=                 ; 必须设置   参数 ( 定义的变量 ) 数目
Version=              ; 可选       循环版本, 不带数据 0
Code_typ=             ; 可选       输出带有 = 0 或者不带 = 1 ; #-行
```

举例：

```
[CYCLE83]
Mname=CYC83
Dir=cst.dir
Output=bohren.ini
Anzp=17
Version=3
Code_typ=0
```

不同的循环版本

对不同版本相同名称的循环工具版本号进行区分。
程序调用前，主程序段/辅助程序段和程序段号码保留不变。

零件程序中循环调用行：

```
/1234 :44 CYCLE94( , , )
```

扩展后以字符串和 ; #-行：

```
;NCG#CYC94#\CST.DIR\DREHEN.COM#NC1#1#*NCG;*RO*;*HD*
#####*NCG;*RO*;*HD*
/1234 :44 CYCLE94( , , " , )
;#END#*NCG;*RO*;*HD*
```

参见

反编译 (页 107)

向前/后查找(SF, SB) (页 109)

3.3 从 NC 程序激活对话框

引言

通过 HMI 高级和 HMI 内置 sl 可以显示由用户定义的对话框：
通过设计确定对话框的外观（在循环目录中修改 COM 文件）。

通过零件程序中的功能调用来调用并再次退出对话框。HMI 软件（操作界面）此时保持不变。不能同时在不同的通道中调用由用户定义的对话框。

命令通道

功能“从 NC 程序激活对话框”也指“命令通道”。

激活命令通道

由用户定义的对话框可以，例如用于下列场合，在进行零件程序过程之前用定义的值占用用户变量（GUD）。

- **最多 2 个通道：**

按照标准为通道 1 和 2 激活“命令通道”。

- **多于 2 个通道：**

对于 HMI 高级，必须激活“命令通道”（在尚未安装西门子测量循环时）。

为此，在开机调试操作区的文件 F:\MMC2\COMIC.NSK 中进行更改：

在操作区“开机调试”中按下软键“HMI”→“编辑器”选择文件 F:\MMC2\COMIC.NSK 并插入以下文本（通道 1 和通道 2 后）：

```
REM CHANNEL  
TOPIC(machineswitch) COMIC_START(COMIC001MachineSwitch"...) [对于通道 1 和 2 对照文本]
```

在随后的控制系统重新启动时（关闭/打开），为相应的通道激活命令通道。

HMI 高级时激活

文件 COMIC.NSK 的内容：

```
REM ----- TYPICAL COMIC START
REM CHANNEL 1
TOPIC(machineswitch) COMIC_START("COMIC001MachineSwitch
",/Channel/Configuration/mmcCmd[u1],
/Channel/Configuration/mmcCmdQuit[u1])
REM CHANNEL 2
TOPIC(machineswitch) COMIC_START("COMIC002MachineSwitch
",/Channel/Configuration/mmcCmd[u2],
/Channel/Configuration/mmcCmdQuit[u2])
```

3.3.1 指令 "MMC" 的结构

编程

句法	MMC (“操作区、指令、COM 文件、对话画面名称、用户数据定义文件、图形文件、显示时间或确认变量、文本变量...”，“确认模式”)	
参数：	操作区	软键名称，通过该软键可以调用设计的用户对话框。 预设置：CYCLES，在软键 14 上显示为“循环”且可以通过按键 < ETC> 实现。
	指令	PICTURE_ON 画面选择 PICTURE_OFF 取消画面选择
	COM 文件	对话画面文件名（最多 8 个字符，在用户循环、制造商循环或标准循环目录中）。在此确定对话画面的外观。在对话画面中可以显示用户变量和/或注释文本。
	对话框名称	单个的对话框通过对话框名称选择。
	GUD 文件	在读/写变量时存取的用户数据定义文件。
	图形文件	待显示的 BMP 图形的文件名
	(仅 HMI 高级)	
	确认变量(仅 HMI 高级)	确认变量 确认模式“A”下
	或显示时间	显示时间 确认模式“N”下
	文本变量	COM 文件文本变量中的画面标题或者注释文本
	确认模式	"S" 表示 同步 通过软键“确定”确认 "A" 表示 异步 通过设计的软键确认 "N" 表示 不退出 没有确认，而是显示时间

图形的保存结构

*.bmp 文件根据分辨率分别存放在各个子目录中，它们是：

- 对于标准循环：

\CST.DIR\HLP.DIR\640.DIR	对于分辨率 640 dpi
\CST.DIR\HLP.DIR\800.DIR	对于分辨率 800 dpi
\CST.DIR\HLP.DIR\1024.DIR	对于分辨率 1024 dpi

- 对于用户循环：

\CUS.DIR\HLP.DIR\640.DIR	对于分辨率 640 dpi
\CUS.DIR\HLP.DIR\800.DIR	对于分辨率 800 dpi
\CUS.DIR\HLP.DIR\1024.DIR	对于分辨率 1024 dpi

- 对于制造商循环

\CMA.DIR\HLP.DIR\640.DIR	对于分辨率 640 dpi
\CMA.DIR\HLP.DIR\800.DIR	对于分辨率 800 dpi
\CMA.DIR\HLP.DIR\1024.DIR	对于分辨率 1024 dpi

3.3.2 MMC 指令举例

零件程序中的 MMC 指令

```
MMC ( "CYCLES, PICTURE_ON, T_SK.COM, 画面  
1, MGUD.DEF, BILD3.BMP, TEST_1, A1", "S" )
```

CYCLES	操作区
PICTURE_ON	选择对话框
T_SK.COM	循环目录中的文件名
画面 1	对话框的名称
MGUD.DEF	用户数据定义文件
画面 3.BMP	图形文件的名称 (仅 HMI 高级)
TEST_1	确认变量(仅 HMI 高级) 或者模式"N"下的显示时间
A1	COM 文件文本变量中的画面标题或者注释
S	应答方式：同步

定义目录中的用户变量

```

%_N_UGUD_DEF
; $PATH=/_N_DEF_DIR
DEF CHAN REAL TEST_1

CHAN          通道专用的适用范围
REAL         数据类型
TEST_1       用户变量的名称

```

循环目录中的对话画面文件 (*.COM)

```

//C3(画面 2)
R/ 15 75 / 5 /COMMENT, %1 %2 %3/ W,RJ / TEST_1 / ...

R          变量类型：实型、整型或字符串
15 75     允许范围：15 至 75
5         用户变量的预设置
COMMENT, %1 %2 %3  注释文本，带可选的文本变量
W,RJ      存取方式  W = 读取和写入
           :         R = 仅读取
           W,RJ = 读取和写入，带注释
           J = 向右对齐输入/输入栏
           <无> = 向左对齐输入/输入栏

TEST_1    用户变量

```

文本变量

[文本变量]

A1 = 举例 2：无确认情况下的 MMC 指令

A1

举例 2：无确认情况下的 MMC 指令

MMC 指令的参考参数

画面标题或注释文本

注意

用大写方式写入变量名称、文本变量和循环名称。

设计用于对话框调用的软键

MMC 指令的软键占用，带异步确认模式。

```

[画面 3]
SK1 = 结束
SK2 = 画面 2
可以设计软键 SK0 至 SK15

```

3.3.3 举例 1：无确认情况下的 MMC 指令

零件程序

```

N10 MMC("CYCLES,PICTURE_ON,T_SK.COM,画面 1,GUD4.DEF,,,A1","N")
N20 TEST_1 = 1
N25 G4 F10
N30 MMC("CYCLES,PICTURE_OFF","N")
M30

```

参数：

```

对话框文件 (*.COM) //C1(画面 1)
(R///ANW.VAR TEST_1/W/TEST_1///)

文本变量
[文本变量]
A1 =.....举例 1：无确认情况下的 MMC 指令

```

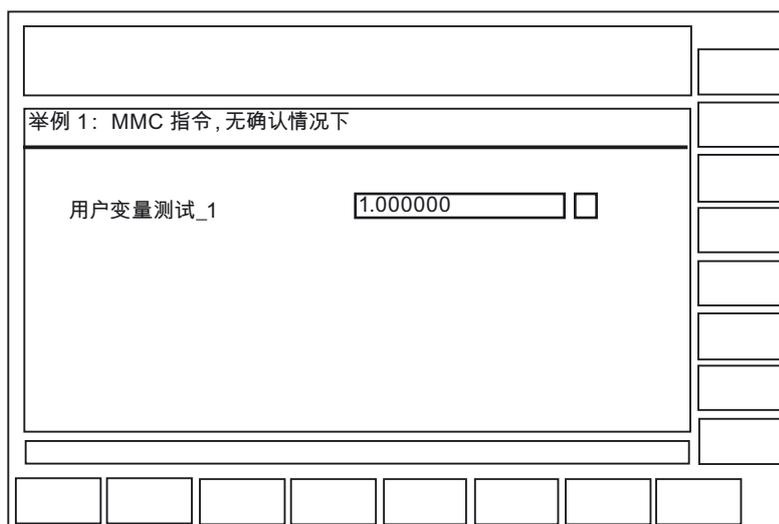


图 3-2 没有确认的举例 1

操作步骤

通过标题 A1 短时显示 GUD4.DEF 中的用户变量 TEST_1。由零件程序段 N25 得出停留时间。

3.3.4 举例 2：停留时间和可选的文本变量

零件程序

```
N10 MMC("CYCLES,PICTURE_ON,T_SK.COM,画面 6,GUD4.DEF,,10,T1,G1","N")
N15 G4 F15
N30 MMC("CYCLES,PICTURE_OFF","N")
M30
```

参数：

对话框文件 (*.COM) //C6(画面 6)
R///ANW.VAR TEST_1, %1/W/TEST_1///)

文本变量

[文本变量]

T1 = 举例 2：停留时间和可选的文本变量 ...

G1 = 可选的文本变量

在这里将第 7 个参数视为用于无确认情况下模式的显示时间。随后删除表格内容。对话框一直保持至 PICTURE_OFF。第 8 个参数 (T1) 为标题的文本变量。如果没有条目存在，则显示操作区名称“循环”。参数 9 至 23

为可选的文本变量 (“G1=可选的文本变量”)。必须在 COM 文件段[文本变量]下预设可选的文本变量。

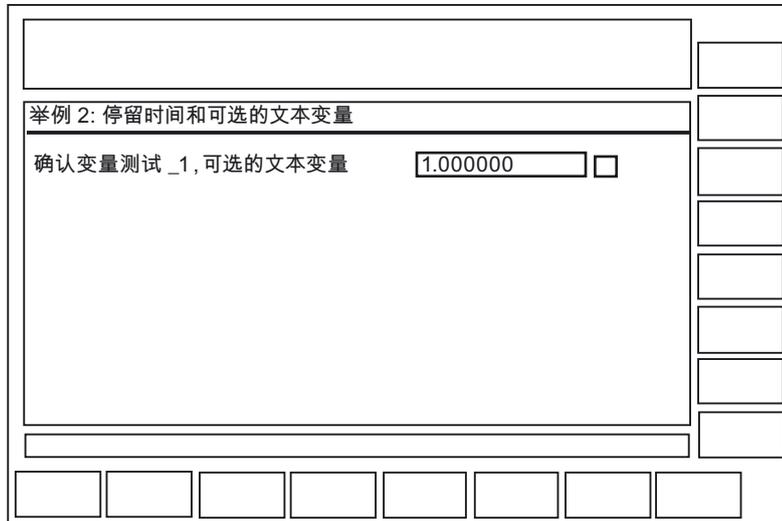


图 3-3 举例 2 停留时间

操作步骤

在该例中，COM 文件 (ANW.VAR TEST_1) 中的注释文本在第一个占位符 (%1) 的位置上，以扩展文本变量“G1=可选的文本变量”的内容。通过调用 MMC 指令中的文本变量 (第 9 至第 23 个参数) 可以用这种方式“组合”信息或名称。

3.3.5 举例 3：MMC 指令，在同步确认模式下

零件程序

```

N15 MMC( "CYCLES,PICTURE_ON,T_SK.COM,画面 1,GUD4.DEF,,,F1","S")
N18 STOPRE
N20 TEST_1 = 5
N25 MMC( "CYCLES,PICTURE_OFF","N")
M30

```

参数：

对话框文件 (*.COM) //C1(画面 1)
(R///ANW.VAR TEST_1/W/TEST_1///)

文本变量

F1 = ...举例 3：MMC 指令，在同步确认模式下...

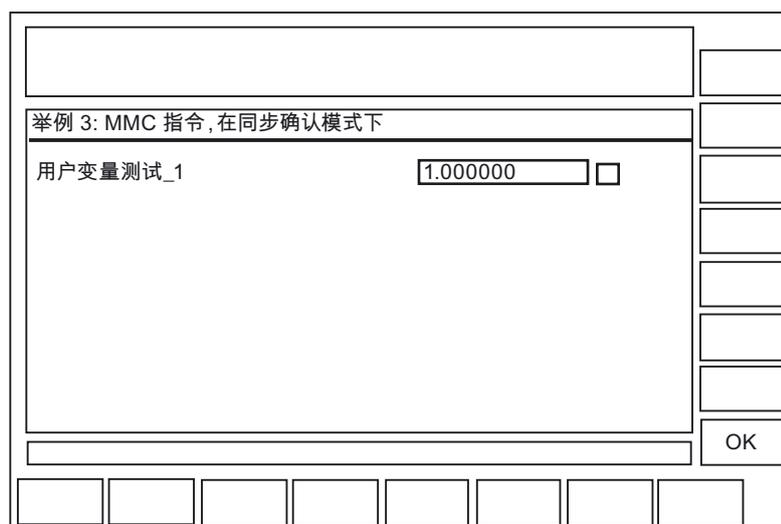


图 3-4 举例 3 同步确认模式

操作步骤

一直显示用户变量 Test_1，直至按下软键“确定”。此时，在程序中另外用数值 5 覆盖用户变量。

在无 STOPRE 的情况下，在键盘输入之前进行分配（在有 STOPRE 的情况下则在键盘输入后进行分配！）。

3.3.6 举例 4：输入/输出栏定位

说明

通过在 COM 文件中规定位置参数，可以在显示区域内部的任意位置上显示注释栏或者输入/输出栏。

零件程序

```

N15 MMC("CYCLES,PICTURE_ON, T_SK.COM; 2,GUD4.DEF,,C1","S")
N20 TEST_3 = 5
N30 MMC("CYCLES,PICTURE_OFF","N")
N40 M30

```

参数：

这两个参数分别由三个数值构成，它们规定了位置和栏长度。以 Twips 为单位规定数值，一个像素等于 15 个 Twips。栏高确定为 250 Twips。

```

对话框文件 (*.COM //C2(画面 12)
) (R///Var.Name/R/TEST_3/6000,2800,8000/
200,3000,7500)
/6000,2800,8000 注释栏位置
/200,3000,7500 输入/输出栏的位置
第一个值 = 0 → 按照预设置自动定位
无规定 → 如同 PCU 20 定位 (预设置)

```

数值含义：

(./6000,2800,8000/....)

6000	到画面左边缘的距离
2800	到画面上边缘的距离
8000	栏长度

操作步骤

可以为 16 个注释栏和 16 个输入或输出栏进行图形设计。如果多于 16 个栏，可以通过滚动条控制这些栏。

图 3-5 举例：4a 定位参数

为此，若要光标控制功能完全正常，必须重叠已设计的栏：

图 3-6 举例：4b 定位参数

3.3.7 举例 5：在对话框画面中显示图形

说明

图形，例如用画笔建立的图形可以通过图形文件数据在对话框中显示。
图形的注释文本可以通过 COM 文件预设。此时，注释文本也可以通过位置参数定位。

注意

图形仅可以通过在图形程序中移动自行定位。

零件程序

```
N10 MMC ("CYCLES,PICTURE_ON,T_SK.COM,画面 8,GUD4.DEF,GRA.BMP,,M1","S")
N20 MMC ("CYCLES,PICTURE_OFF","N")
N30 M30
```

参数：

对话框文件 (*.COM) //C8(画面 8)
(I/// 画面显示编号 2///4000,3000,7500)
(I/// 用画笔建立 ///4000,3250,7500)
文本变量 M1 =举例 5：画面显示.....
例如，通过程序“画笔”建立画面。画面尺寸：300X500
像素，画面尺寸仅可以用图形程序更改。

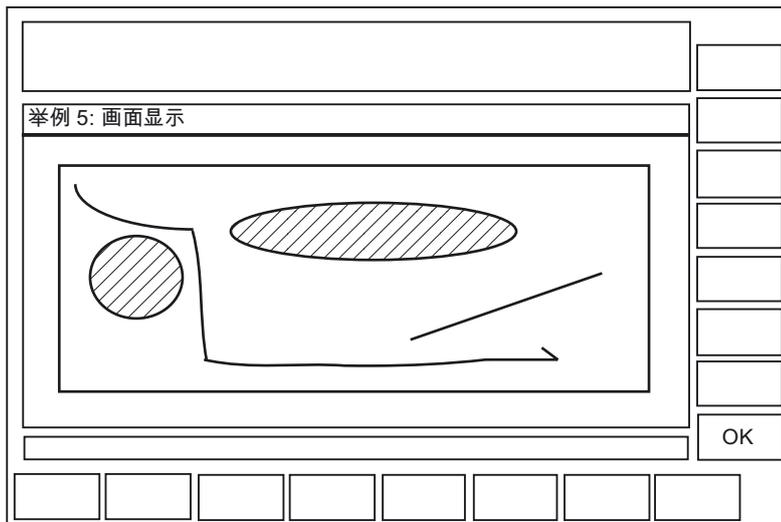


图 3-7 举例 5: 带图形

3.3.8 举例 6：显示 BTSS 变量

说明

可以显示以下 BTSS 变量。

零件程序

```
MMC("CYCLES,PICTURE_ON,T_SK.COM,画面 7,GUD4.DEF,,TEST_1,J1","S")
```

参数：

```
对话框文件 (*.COM) //C7(画面 7)
(R///Test_1/R/Test_1)
(I///); (视为空行)
(R///轴实际值1/R/$实际值)
(R//1/R 参数 12/W/$R[12])
```

文本变量

```
J1 = ...举例 7：BTSS 变量
```

```
[BTSSVar]
$实际值=/Channel/machineaxis/actToolbasePos[u1,1]
$R[12]=/Channel/Parameter/rpa[u1,12]
```

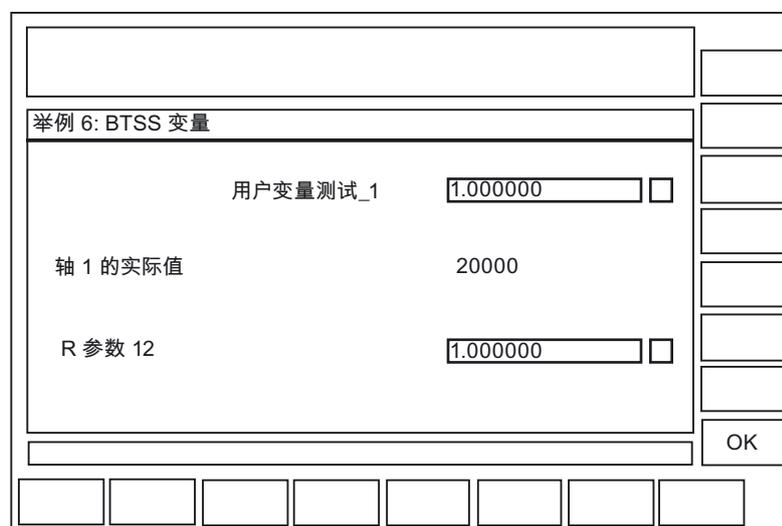


图 3-8 举例：BTSS 变量

操作步骤

在变量 TEST_1 下生成空行。

轴实际值为只读。

R12 预设置为 1。

3.3.9 举例 7：用软键进行的异步确认模式**说明**

软键可以用异步模式通过在 COM 文件中预设显示，与应答变量链接并在零件程序中评估。

零件程序

```

N10 QUIT_1 = "START"
N20 MMC ("CYCLES,PICTURE_ON,T_SK.COM,画面 3,GUD4.DEF,"QUIT_1,K1","A")
N30 LABEL0:
N40 STOPRE
N50 IF MATCH (QUIT_1,"SK1") >= 0 GOTOF LABEL1
N60 IF MATCH (QUIT_1,"SK2") >= 0 GOTOF LABEL2
N70 GOTOB LABEL0
N80 LABEL2:
N90 MMC ("CYCLES,PICTURE_ON,T_SK.COM,画面 1,GUD4.DEF,"N1","N")
N100 G4F10
N110 LABEL1:
N120 MMC ("CYCLES,PICTURE_OFF","N")
N130 M30

```

参数：

对话框文件 (*.COM) //C3(画面 3)
(S/// ANW.VAR QUIT_1/W/QUIT_1//)

文本变量

[文本变量]

K1 = .. 举例 8：MMC 指令，异步确认模式下

N1 = .. 举例 8：画面 2

软键

[画面 3]

SK1 = 结束

SK2 = 画面 2

程序结构

将应答变量定义为字符串。

字符串长度：>= 20

(值< 20 仅在内部评估，在按下软键 SK0 ...SK15 时输入到位置 17...20 上)。

在零件程序中为字符串赋一个值并由此删除可能的旧软键信息。

在可以选用零件程序之前(与确认变量有关)，程序段过程通过命令 STOPRE 停止。

```
IF Match (Quit_1," SK1") >= 0 GotoF Label1
```

；在字符串范围内查找一个字符串

如果没有按下软键，则重新循环一次。



图 3-9 举例 7：异步确认模式

操作步骤

一直显示通过异步 MMC 指令调用的画面，直至按下以下两个设计的软键：

- 用软键“结束”立即结束用户对话。
- 接着，按下软键“画面 2”持续 10 秒显示一个其它的对话框。

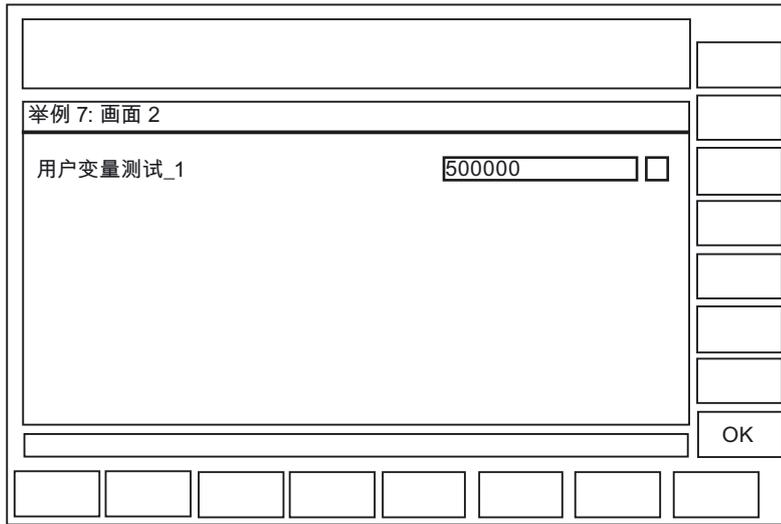


图 3-10 举例 7：画面 2

设计热键和 PLC 键

4.1 引言

概述

在本章节中说明了以下操作单元的设计方法：

- OP 010 和 OP 010C 以及带热键区的 SINUMERIK 键盘的 6 个热键以及按键 <Machine> 和 <MENU SELECT>，其布局可以选择性修改
- 由 PLC 分析的按键，例如：机床操作面板上的按键
- 由 PLC 分析为 PLC 按键或者“虚拟按键”并且能够触发 HMI 程序内设计的操作流程的事件。

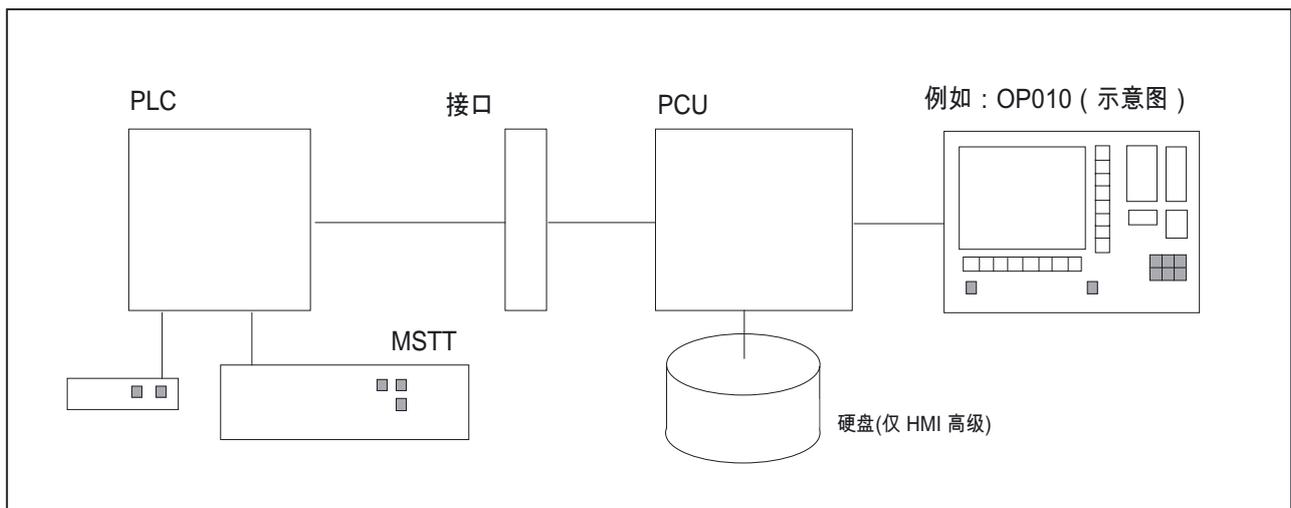


图 4-1 配置 OP010

应用

热键和按键可以用于以下任务，例如：

- 选择操作界面（例如机床，参数等等）
- 选择所选的子菜单（例如在“诊断”操作区选择报警画面）
- 触发动作（例如在“参数”操作区选择刀具列表并按下软键 HS3）
- 有针对性地选择用“补充操作界面”建立的菜单
- 选择其它画面，取决于用“补充操作界面”建立的窗口中当前的操作情况。

设计

设计通过功能“补充操作界面”进行。

- OP 的 6 个热键允许直接显示 HMI 的每个操作区域。以此缩短通过基本菜单执行的一般选择过程。可以由此来改变 6 个热键的标准占用。
- PLC 和 HMI 的接口允许扩展 PLC 键的功能。为此可以设计 HMI 上触发的操作处理过程。对于 PLC 的使用，提供按键号码 50 至 254。
- 按键 <Machine> 和 <MENU SELECT> 可以用与 OP 的 6 个热键相同的方式设计（可选）。这些热键用作为 HK7 和 HK8。

4.1.1 OP 的热键

排列（预设置）

OP 的 6 个热键分 2 行排列，每行 3 个键

行 1：标签（不带符号）		设计作为
OP 相关：		
OP10	Machine	HK1
OP10C	Machine	HK1
OP10S	Position	HK1
Program		HK2
Offset		HK3

行 2：标签（不带符号）		设计作为
Program Manager		HK4
Alarm		HK5
Custom		HK6

可选 HK7 和 HK8

按键 <Machine> 和 <MENU SELECT> 可以如同 HK1 至 HK6 一样进行设计。由此可以使这些键的预设失效并激活新的自身功能。

标签（不带符号）	设计作为
Machine	HK7
MENU SELECT	HK8

关于 HK7 和 HK8 的详细信息，参见章节“<M> 键和 <MENU-SELECT> 键设计为 HK7, HK8”。

注意

热键 1 和 7 (<M>) 键对于 OP10S (标签"Position") 无法从硬件上进行区分。
在按下按键的任意一个时总是触发热键 7。 如果已设计 HK1, 则这些事件只能通过外部 MF2 键盘触发。

MF2 键盘上的热键排列

热键	OP 上的标签	MF2 的键
HK1	Position	<SHIFT+F11>
HK2	Program	<END> (NB)*
HK3	Offset	<Page Down> (NB)*
HK4	程序管理器	<Home> (NB)*
HK5	Alarm	<Page Up> (NB)*
HK6	Custom	<SHIFT+F12> 或者光标下 (NB)*
HK7	M Machine	<SHIFT+F10>
HK8	Menu Select	<F10>

*) 键位于编号程序块: <NumLock> 必须关闭。

4.1.2 供货状态下键的功能**供货状态**

对于提供的系统, 热键的功能布置输入在文件 KEYS.INI 中。

对于 HMI 高级文件可以存放于不同的目录下:

- user
- oem
- add_on *)
- mmc0w32 *)
- mmc2 *)
- hmi_adv *)

*) 所标记的目录保留用于西门子。

文件按照此处规定的顺序处理。最高级目录的条目使得较低一级目录的条目失效。

供货时 HK1 至 HK6 的分配如下存放在标准系统的目录mmc2的文件 KEYS.INI 中:

按键		功能
HK1	Position	"加工"操作区, 最后的画面
HK2	Program	"程序"操作区, 最后的画面
HK3	Offset	"参数"操作区, 最后的画面

4.2 设计

按键		功能
HK4	Program Manager	编程基本画面
HK5	Alarm	“诊断”操作区，报警画面
HK6	Custom	“默认定制”操作区，最后的画面 (用户设计的操作界面)

对于 ShopMill/ShopTurn，按键功能如下：

按键		功能
HK1	Position	“加工”操作区，最后的画面
HK2	Program	程序编辑器，最后的状态
HK3	Offset	“参数”操作区，最后的状态
HK4	Program Manager	程序目录，最后的状态
HK5	Alarm	“诊断”操作区，报警画面
HK6	Custom	“默认定制”操作区，最后的画面 (用户设计的操作界面)

4.2 设计

4.2.1 设计概述

概述

下图以图表方式描述了 OP 设计的热键和 PLC 按键之间的关系和根据“补充操作界面”进行设计的步骤。

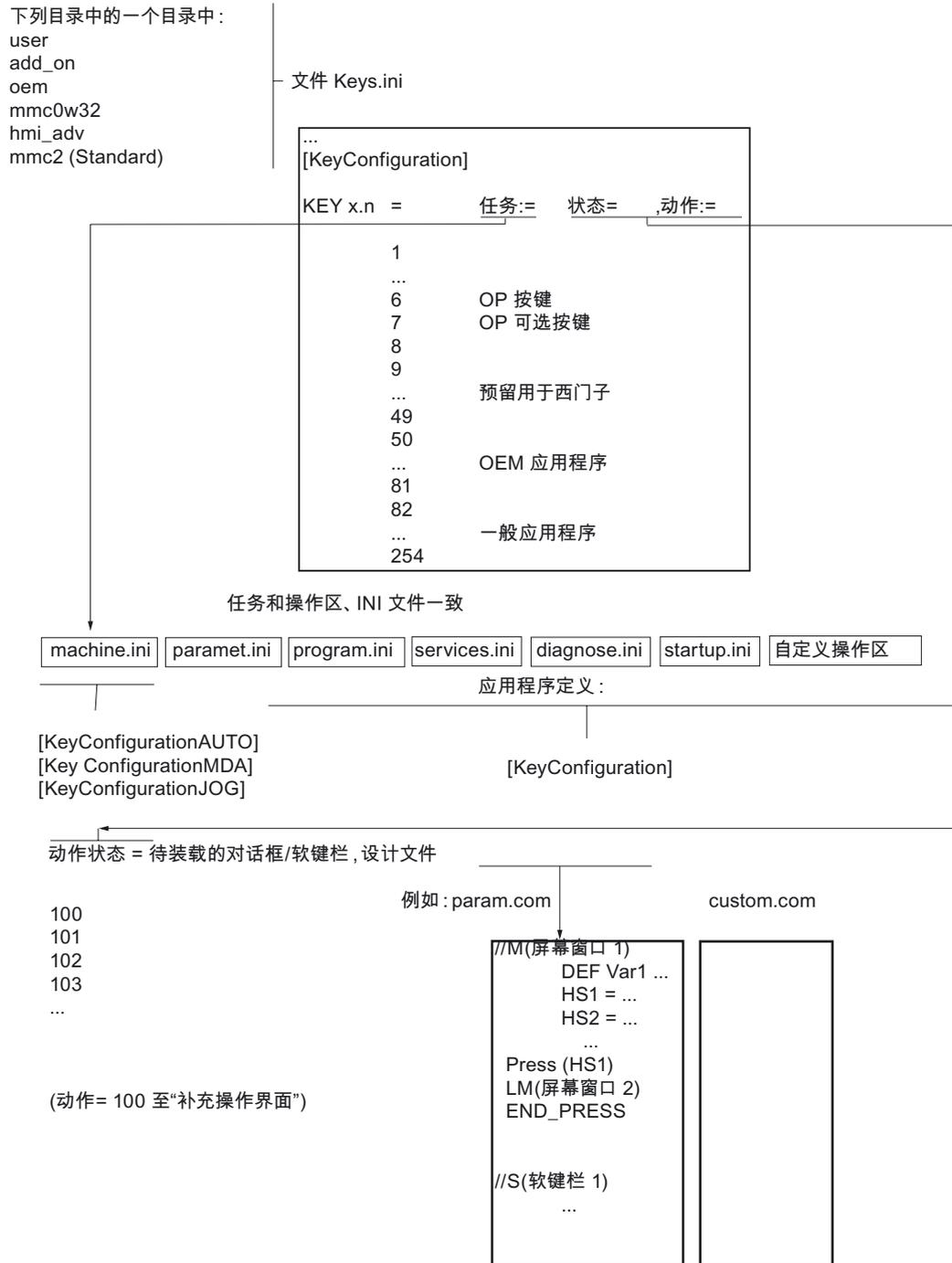


图 4-2 设计概述

4.2.2 在文件"IF.INI"中进行设计

概述

每次在上述的目录中标示配置文件 KEYS.INI 都将引起程序段 [键配置] 中的热键反应。
每个条目 (行) 定义某个热键的某个操作 (多次操作) 上的系统反应, 下面称作热键事件。

配置文件 KEYS.INI

配置文件 KEYS.INI 获得一个自身的程序段, 该程序段用于分配 PLC 按键的 ini 文件。
没有该条目就无法识别 ini 文件。

在标准情况下存在下列条目:

```
[HMI_INI_FILES]
任务 0 = machine.ini
任务 1 = paramet.ini
任务 2 = program.ini
任务 3 = services.ini
任务 4 = diagnose.ini
任务 5 = startup.ini
任务 6 = shopmill.ini
( 或者: 任务 6 = shopturn.ini je, 视系统而定, 必须设置! )
任务 11=custom.ini
```

注意

为了使 KEYS.INI 中的更改有效, 必须重新启动。

条目形式

条目的形式在“热键事件”下进行定义。
为了可以重复操作那里所使用的相同热键, 首先描述多重操作。

多重操作

功能扩展确定热键多重操作时的过程：在操作热键时由于文件 KEYS.INI 中增加的重复次数“n”，可以分配一个自主的组合：任务/状态/动作。为此每次按下按键都可以进行状态切换和画面选择以及软键选择。

在多重操作时切换到另一个操作区将删除当前的热键。为此也适用于未操作的热键。此时例如实现切换到另一个操作区（操作区菜单等等）与此无关。此外每次按下功能键（例如软键，操作区切换键，通道切换键等等）时同样可能导致删除当前状态。

在当前的对话框中进行栏输入不会中断重复计划。
在当前的对话框中按下软键，通过用来调用当前对话框的按键中断重复计划。

注意

多重操作可以用于 HK1 – HK8 以及备用的西门子按键 HK9 – HK49。

软键多重操作举例

在第一次操作热键时激活相关的操作区并可能触发该操作区中的一个状态/动作（条目 1）。在再次操作该热键时，接着执行该热键接下来的条目，操作区切换不再进行。如果热键的所有设计的条目已运行，则重复该循环。条目总是以正向（从 0 至 9）运行。不进行负向的运行。在设计时不允许存在“事件缺空”（缺少的条目）。一个缺空将被视作链的结束并在下次按下按键时立即再次在按键 x.0 处开始处理。

设计：

```
[KeyConfiguration]
```

```
按键 1.0 = 任务:=0, 状态:=10,      ; 热键 1, 第 1 次操作
动作:=2
按键 1.1 = 状态:=10, 动作:=3      ; 热键 1, 第 2 次操作
按键 1.2 = 动作:=4                ; 热键 1, 第 3 次操作
```

KEYS.INI 中用来选择操作区“加工”的条目：

```
按键 1.0 = 任务:=0                ; 第 1 次按下按键的最后画面
按键 1.1 = 任务:=0, 动作:=0      ; 第 2 次按下按键的开始画面
按键 1.2 = 状态:=0, 动作:=3      ; 在第 3 次按下按键和选择第 3
                                   个软键时的开始画面
按键 1.3 = 状态:=0, 动作:=100    ; 在第 4 次按下按键并选择通过
                                   machine.ini 中 <= 100
                                   设计的“补充操作界面”时的开始画面
```

针对动作值的软键分配

水平软键 (SKHi)	动作
SKH1	0
SKH2	1
SKH3	2
SKH4	3
SKH5	4
SKH6	5
SKH7	6
SKH8	7

垂直软键(SKVi)	动作
SKV1	8
SKV2	9
SKV3	10
SKV4	11
SKV5	12
SKV6	13
SKV7	14
SKV8	15

按键	动作
<RECALL>	16
<ETC> (仅 HMI 内置 sl)	17

4.2.3 编程热键事件

说明

热键事件由最多四个属性组成。关键属性 KEY (按键) 识别事件，并且必须总是位于第一位。所有其它的属性可选，但是必须至少再规定一个属性。该属性的顺序可以在行中的任意位置。

编程热键事件

句法： 按键 x.n = Task:=任务, State:=状态, Action:=动作

参数： **Key (热键)** 值 x.n
包含热键号x和一个直接的事件n，该事件应该为第 n 次操作 (对于多次操作)。
热键号的取值范围为 1 - 254。
事件 n 的取值范围是 0 - 9 (10 次按键操作)。
热键 1 - 8 分配 OP 的按键。热键 9 - 49 预留用于西门子。另外还有 PLC 专用的按键 50 - 254。这些按键不是真正的热键，而是用于通过 PLC 选择画面 (虚拟键)。按键 50 - 254 也可以设计为不带任务。
在这种情况下事件总是指定当前的任务。

Task(操作区) 各操作区 (任务) 热键的分配通过一个任务号实现。任务号同时定义水平软键，在该水平软键上操作区挂在操作区切换菜单中。它也形成一个任务号和附属软键之间的关联。在 HMI 系统中备有三个 (对于 HMI 内置 sl 为两个) 软键条用于操作区切换。为此规定为下列分配方式：
在按下操作区切换键后选择软键栏 1。
SK1 → 任务 0 ...
SK8 → 任务 7
接下来按下键<ETC>切换到软键栏 2
SK1 → 任务 8 ...
SK8 → 任务 15
然后按下键<ETC>切换到软键栏 3 (只适用于 HMI 高级)：
SK1 → 任务 16 ...
SK8 → 任务 23

按下键<ETC>后再次切换到软键栏 1。HMI 高级的任务号值在 0 至 23 之间；HMI 内置 sl 的任务号值在 0 至 15 之间。如果没有规定任务，则事件对于当前的任务有效（当前的操作区）。

可以获得任务号：

- 对于 HMI 高级，从文件 REGIE.INI 获得。
- 对于 HMI 内置sl，通过工具SCK（软件配置套件）。

可以通过设计在软键和操作区（任务）之间进行用途特定的分配：

- HMI 高级：
文件 REGIE.INI 的段落[TaskConfiguration]
- HMI 内置 sl：SCK “修改配置”

状态

通过属性“状态”可以在任务中选择某个画面。值范围取决于相应的应用程序，最多可以在 0 至 65534 之间（65535 由系统占用）。

动作

通过属性“动作”可以在任务内选择某个软键。只有当处于一个定义的状态中，例如基本菜单，该属性才有意义。

取值范围与对应的应用程序有关，最多在 0 到 17 之间。属性动作这里是比较特殊的，以≥ 100 的值通过“补充操作界面”触发画面选择。

值范围 0 - 99 中的动作在 ShopMill/ShopTurn 上不执行。

另见

章节“可选状态的列表”

4.2.4 扩展及特殊情况

设计 M 键和“Menu Select”键为 HK7 , HK8

可以如同热键一样有选择地设计 <M> 键 (Machine) 和 <MENU-SELECT> 键。由此, 这些键失去了其最初的意义和功能。通过配置文件 KEYS.INI 确定新的功能。如果在按键的配置文件中有定义, 但是存放没有反应, 则按键无效。

热键的按键布置:

		MF2 键盘	热键
	MACHINE	按键 <Machine>	SHIFT+F10 HK7
	MENU SELECT	按键 <MENU-SELECT>	F10 HK8

如果对于热键 7 或者热键 8, 在配置文件 keys.ini 中不存在条目, 则该按键没有映射到热键上, 而只包含其原来的功能 (兼容模式)。一个没有反应的按键定义通过属性 <empty> 规定。

举例:

```
[KeyConfiguration]
按键 7.0 = 任务:= 3, 状态 := 10      ; 激活 M 键到热键 7
                                       ; 的映射 并确定新的按键反应
                                       ; 激活 <MENU SELECT> 键 ( F10 ) 到热键
                                       ; 8 的映射 , 按键无效
Key8.0 = <empty>                      ; 未指定反应
```

HMI 高级的扩展

机床制造商可以用自身的设置覆盖目录 mmc2 的 KEYS.INI 中的条目。为此, 机床制造商可在以前执行时查找过的目录 \user 和 \oem 的其中一个里面输入其协议。只需要输入其偏差之处, 而无需整个定义语句。

HMI 高级特殊情况下的动作:



<ETC> 键, 动作 17 无效。

HMI 内置 sl 上的特殊情况

特殊情况下的任务、状态和动作：

- 用于任务的值范围从 0 至 15。
- HMI 内置 sl 选择最后一次任务切换前已激活的画面组合（状态未设计）。
- 状态的值范围限制为 0。只能选择操作区的基本菜单。
- 动作的值范围为从 0 至 17：
 - 水平软键 1 到 8
 - 垂直软键 1 到 8
 - 回调
 - ETC

4.2.5 PLC 按键的扩展

概述

对于 PLC 按键，即可以设计一个任务切换，也可以设计一个状态切换。

动作的值域：

标准应用程序	0 – 17
“补充操作界面”应用程序	≥100

PLC 专用键的编号为 50 - 254;其中 50 - 81 号为 OEM 应用程序预留。

举例：

- 这里指示的配置不进行任务切换和状态切换，然而保留在当前的任务中和当前的画面中，并传输一个动作 100。

```
[KeyConfiguration]
```

```
HK50.0 = 动作:=100
```

- 同样这里也可以设计一个任务切换和状态切换。

```
HK50.0 = 任务:=1, 状态:=10, 动作:=100
```

4.3 PLC 接口

4.3.1 接口结构

概述

在 PLC 接口中规定范围 DB19.Byte10 用于选择按键。这里 PLC 可以直接规定 50 至 254 之间的一个按键。

(键 1 至 49 预留用于西门子，键 50 到 254 专门用于 PLC 按键。)

注意

对于 M:N 运行方式，第 2 个 HMI 接口的范围是 DB19.Byte60。

确认

通过 HMI 系统确认分两步进行：

- 在第一步中由 HMI 软件将控制信息 255 传输到 DB19.Byte10 中。
- 接着在第二步中删除 DB19.Byte10 中向 PLC 的自身确认信息。

这是必要的，为此尽管 HMI 和 PLC 不同步，可以立即接连两次由 HMI 识别相同的按键代码。通过该定义的虚拟按键按下 PLC 可以清楚识别每个按键顺序。控制信息对于 PLC 程序没有意义 (透明) 并且不允许改变。

下一次按键规定

如果在传输字节中为 0，则 PLC 程序可以给出一个新的按键。为此同时在 HMI 系统中处理当前按键的请求。

请求切换到相应的任务或者在当前的任务中触发一条状态/动作命令。

如果无法进行任务切换，则在操作界面上给出一条相应的信息。

对 PLC 程序的请求

只有当 HMI 系统确认了以前的请求 (在接口中为 0)，才允许处理一条新的请求。当 PLC 程序从一个机床控制面板的按键或者另一个信息源导出按键，接着必须有足够的按键中间缓存，以此在快速操作时才不会丢失按键按下信息。

从 PLC 选择对话框

在 PLC 和 HMI 内置 sl 之间存在一个接口用于选择对话框。在该 PLC 对话框中可以提供与由软键选择的对话框中相同的描述方式和功能。

4.3.2 PLC 画面选择说明

接口说明

接口包含画面号、PLC 传到 HMI 的控制位和 HMI 传到 PLC 的控制位。该接口在 DB19 中需要总共 8 个字节，HMI 内置 sl 各占用 4 个字节。

由于在“多个 NCU 上的多个操作面板”范围内每个 NCU 最多可以同时响应两个 HMI 内置 sl，因此也存在两个这样的接口。

文献：/FB2/，“多个 NCU 上的多个操作面板”(B3)

接口结构

HMI 内置 sl 和 PLC 之间的接口使用下列数据：

HMI 1:	DB19.DBW28:	画面号
	DB19.DBB30:	控制位 PLC → HMI, PLC 字节
	DB19.DBB31:	控制位 HMI → PLC, HMI 字节
HMI 2:	DB19.DBW78:	画面号
	DB19.DBB80:	控制位 PLC → HMI, PLC 字节
	DB19.DBB81:	控制位 HMI → PLC, HMI 字节

PLC 字节	位 0	画面选择
	位 1	画面选择
HMI 字节	位 0	接收画面选择或者接收取消画面选择
	位 1	选择画面或者取消选择
	位 2	已选择画面
	位 3	已取消选择画面
	位 4	错误，画面选择无法进行
	位 7	无效位

对于从 PLC 要传输的**画面号**使用两个字节，每一个字节（PLC 字节）和 HMI（HMI 字节）用于画面选择的**协调**。

运行接口

接口由制造商的 PLC 程序运行，包含以下功能（DB 19 摘录，第一个 HMI 接口）：

	PLC → HMI	选择	撤销选择	HMI → PLC	选择	撤销选择
DBW 28	画面号	(1)				
位	DBB 30			DBB 31		
0	画面选择	1 (2) 0 (4)		已接收选择 / 取消选择	1 (3) 0 (6)	1 (2) 0 (3)
1	取消画面选择		1 (1) 0 (4)	已选择画面 取消选择画面	0 (3) 1 (5)	0 (3)
2				已选择画面	0 (3) 1 (7)	0 (3)
3				已取消选择画面	0 (3)	0 (2) 1 (3)
4				错误，画面选择无法进行	0	0 (2)
5				-		
6				-		
7				无效	0	

画面选择

在上述表格中各个步骤的对象通过括号中的数字（步骤号）指定。

- PLC 将画面号输入在号码字中。
- 当 HMI 字节中位 0 和位 7 为零时，PLC 在 PLC 字节中设定位 0 用于画面选择。
- HMI 内置 sI 通过设置 HMI 字节中的位 0 确认 PLC 接收画面选择。在 HMI 字节中同时将位 3 和位 4 置零。
- PLC 复位 PLC 字节中的位 0。
- HMI 内置 sI 通过 HMI 字节中的位 1 设定确认 PLC 启动编译器。
- HMI 复位 HMI 内置 sI 字节中的位 0。
- 如果显示相应的画面，在 HMI 字节中设定位 2。

取消画面选择

在上述表格中各个步骤的对象通过括号中的数字（步骤号）指定。

- 如果在 HMI 字节中设定位 1 和位 2，并复位位 7，则 PLC 设定 PLC 字节的位 1 用于取消画面选择。
- HMI 内置 sl 通过设置 HMI 字节中的位 0 确认 PLC 接收取消画面选择。在 HMI 字节中同时将位 3 和位 4 置零。
- 当编译器结束时，HMI 内置 sl 通过设置 HMI 字节的位 3 和复位位 0、位 1 和位 2 来确认 PLC 取消画面选择。
- PLC 复位 PLC 字节中的位 1。
- HMI 内置 sl 通过 PLC 或者通过退出命令在最后选择的 HMI 画面中切换到画面撤销选择。

画面选择时出错

如果编译器在 20 秒后没有报告或者所需的画面无法显示，则设置 HMI 字节的位 4。

通过 HMI 字节中的位 7 显示 HMI 的 PLC 状态，在该状态中无法进行 PLC 画面选择，例如在 HMI 内置 sl 中标准操作界面和 ShopMill/ShopTurn 操作界面之间转换时。

操作区域

对于 HMI 高级，画面显示在一个自身的操作区中，类似于测量循环画面。该操作区可以手动通过第六个软键（PLC 画面）在第二个操作区条上选择。如果选择操作区时没有 PLC 画面激活，则在标题栏中出现文本“目前没有 PLC 画面激活”。操作区的手动选择或撤销选择由 HMI 字节中的位 2 通知 PLC。

通过自身的操作区也可以在选择画面后通过 PLC 切换到另一个操作区。

对于 HMI 内置 sl，由 PLC 选择的画面无法通过一个操作区手动选择或者撤销选择。即使在切换操作范围之后，PLC 画面也始终在前景。

4.3.3 设计对话框选择

说明

在 HMI 启动时通过文件 COMMON.COM 中的段 [PLC_SELECT] 激活接口。在文件 COMMON.COM 中在段 [PLC_SELECT] 下给设计的画面分配画面号码。

设计

句法 : **PC***i* = *画面名称*, *文件*, *注释*
说明 : 将画面号分配给设计的画面
参数 : *i* 接口中的画面号
 画面名称 对话框名称
 文件 设计对话框的文件
 注释 对话框的注释

举例

```
[PLC_SELECT]  
PC1= CYC82, bohren.com            ;    分配行  
PC2= CYKLE90, gewfraes.com  
PC3= ...
```

参见

COMMON.COM 的查找方案 (页 189)

4.4 选择对话框/软键栏

4.4.1 分配 INI 文件到操作区域

概述

HMI 软件根据动作值 ≥ 100 在“补充操作界面”过程中分配。

如果动作 ≥ 100 ，则在 INI 文件中设计在哪些状态应显示哪些软键栏或者哪些对话框。

HMI 高级/ HMI 内置 si

在下列 INI 文件和段中可以在“补充操作界面”中进行跳转设计：

操作区	文件	段
Maschine	machine.ini	[KeyConfigurationAuto] [KeyConfigurationMDA] [KeyConfigurationJOG]
Parameter	paramet.ini	[KeyConfiguration]
Program	program.ini	[KeyConfiguration]
Service	services.ini	[KeyConfiguration]
Diagnose	diagnose.ini	[KeyConfiguration]
Startup	startup.ini	[KeyConfiguration]
Custom	custom.ini	[KeyConfiguration]

NCU 中的 ShopMill / NCU 中的 ShopTurn

对于 NCU 上的 ShopMill 和 ShopTurn，INI 文件相应地称为 SHOPMILL.INI 和 SHOPTURN.INI。文件中包含了下列可以设计功能“补充操作界面”的章节。

```
[MachineManual]
...
[MachineAutomatic]
...
[Programmanager]
...
[Program]
...
[MessagesAlarms]
...
[ToolsZeroOffset]
...
[MachineMDI]
```

4.4.2 设计功能“补充操作界面”

说明

对于每个动作 (≥ 100)，都可以设计一个或者多个状态 (根据状态列表) 和设计应触发哪个“补充操作界面”功能。如果没有在 INI 文件中设计，则不会显示对话框或者软键栏。

编程任务.状态

句法：	任务.状态 = 要装载的对话框/软键栏，设计文件
参数：	动作 动作是在 KEYS.INI 中设计的动作 ≥ 100 。
	状态 应用程序当前处于的状态
	设计文件 用于存放设计的文件。
	LS/LM 装载软键/对话框的“补充操作界面”命令

举例

```
[KeyConfiguration]
100.10=LS("软键 1","param.com")
100.30=LM("屏幕窗口 1","param.com")
101.10=LS("软键 2","param.com")
101.30=LM("屏幕窗口 2","param.com")
102=LM("屏幕窗口 2","param1.com")
```

说明：

- 例如第一行表示：
如果触发一个动作 100，当前在状态 (画面) 10 中，则显示文件 param.com 中设计的名称为软键 1 的软键栏。
- 例如最后一行表示：
如果触发动作 102，则在当前的对话框中调用名称为屏幕窗口 2 的对话框，该对话框在文件 param1.com 中设计。

4.5 可选状态的列表

4.5.1 HMI 高级上的可选状态

概述

一般情况下，适用于 HMI 高级的所有任务：

没有设计： 当前的状态保留
 设计为 0： 操作区的基本状态跳转。

操作区域“加工”

在“加工”区，状态总是取决于加工状态（AUTO、MDA、JOG、REF）。通过 PLC 按键可以直接选择：

- 每个运行方式的基本画面

状态	BAG	机床功能	显示
0	JOG	REF	“JOG/REF”基本画面
0	JOG	没有	“JOG”基本画面
0	JOG	REPOS	“REPOS”基本画面
0	MDA	没有	“MDA”基本画面
0	MDA	TEACH	“MDA/示教”基本画面
0	MDA	REF	“MDA/REF”基本画面
0	AUTO	没有	“自动运行”基本画面

- 对应的实际值大画面（垂直软键 6）

状态	BAG	机床功能	显示
10	JOG	没有	缩放 实际值 JOG
20	MDA	没有	缩放 实际值 MDA
30	AUTO	没有	缩放 实际值 Auto

- 还可能在 JOG / MDA

可以通过 PLC 在 WCS（工件坐标系）和 MCS（机床坐标系）之间切换。

状态	BAG	机床功能	显示
60	JOG	没有	手轮 选择 JOG
70	JOG	没有	增量 选择 JOG
80	MDA	没有	手轮 选择 MDA

操作区“参数”

这里只能在当前的画面中运行。

操作区“程序”

状态	功能	
10	数据选择	
20	程序管理	
70	记录	

操作区“通讯”

状态	功能	
10	读入数据	
20	读出数据	
40	管理数据	
60	数据选择	
80	外部驱动器	
90	批量开机调试	
100	升级	

操作区“诊断”

诊断的基本画面为报警一览。

在该状态下可以通过水平软键到达其它状态：

状态	功能	
10	报警	
20	显示信息	
30	报警记录	
40	服务显示	
50	PLC 状态	

信息、报警记录和服务显示总是可以到达的。

操作区“开机调试”

开机调试中的基本画面为现有的 NC 轴和驱动一览。

状态	功能	
0	NC 轴和驱动	
10	机床数据	
40	PLC 状态	
50	优化/测试	(高于 V7.1)
60	HMI	

操作区“定制”

参见章节“操作区‘定制’”

4.5.2 HMI 内置 sl 上可选的状态

概述

在 HMI 内置 sl 中只有下列状态的设计方法：

一种设计：	当前的状态保留
设计为 0：	操作区的基本状态跳转。

4.5.3 NCU 上 ShopMill 的可选状态

手动加工

图例说明：

* 如果有选项（显示机床数据已设置）

用户屏幕窗口 标记的功能可以通过“补充操作界面”设计。

如果存在这样的设计，则将其激活，否则出现相应的 ShopMill 标准画面。

状态	功能
19	基本画面
2	T, S, M, ...
30	工件零点
5	工件零点-调整边沿
7	工件零点-/用户屏幕窗口
31	工件零点-校正边沿/用户屏幕窗口
32	工件零点-2 边间距/用户屏幕窗口
33	工件零点-直角
8	工件零点-任意角/用户屏幕窗口
34	工件零点-矩形腔
9	工件零点-钻孔 1/用户屏幕窗口
35	工件零点-钻孔 2
36	工件零点-钻孔 3
37	工件零点-钻孔 4
38	工件零点-矩形轴颈
10	工件零点-圆形轴颈 1/用户屏幕窗口
39	工件零点-圆形轴颈 2
40	工件零点-圆形轴颈 3
41	工件零点-圆形轴颈 4
42	工件零点-调整平面*
11	工件零点-校正按键长度*/用户屏幕窗口
12	工件零点-校正按键半径*
50	测量刀具
16	测量刀具-手动长度/用户屏幕窗口
17	测量刀具-直径/用户屏幕窗口
13	测量刀具-自动长度*/用户屏幕窗口
14	测量刀具-自动直径*/用户屏幕窗口
51	测量刀具-/用户屏幕窗口
15	测量刀具-校正测量头*/用户屏幕窗口
52	测量刀具-校正固定点*/用户屏幕窗口
60	摆动*
4	定位

4.5 可选状态的列表

状态	功能
18	平面铣削
1	ShopMill 设置
90	-/用户屏幕窗口

MDA

状态	功能
20	MDA

自动加工

状态	功能
200	基本画面
210	程序控制
220	程序段查找
230	-/用户屏幕窗口
242	绘制- 顶视图*
243	绘制- 3 面视图*
244	绘制- 立体模型*
250	设置

Program Manager

状态	功能
300	NC 目录
310	零件程序*
320	子程序*
330	用户目录 1*
340	用户目录 2*
350	用户目录 3*
360	用户目录 4*
380	标准循环*
381	制造商循环*
382	用户循环*
383	用户目录 5*
384	用户目录 6*
385	用户目录 7*
386	用户目录 8*

Program

状态	功能
400	工作计划 / G 代码编辑器
412	模拟- 顶视图*
413	模拟- 3 面视图*
414	模拟- 立体模型*

报警信息

状态	功能
500	显示信息
510	-/用户屏幕窗口
520	-/用户屏幕窗口

刀具零点偏移

状态	功能
600	刀具表
610	刀具磨损
620	用户刀具列表*
630	刀具库
640	零点偏移
650	R 参数
660	-/用户屏幕窗口
680	用户数据
690	机床数据

4.5.4 NCU 上 ShopTurn 的可选状态

手动加工 (无选项 “手动加工”)

图例说明：

- * 如果有选项 (显示机床数据已设置)
- 用户屏幕窗口 标记的功能可以通过“补充操作界面”设计。
如果存在这样的设计,则将其激活,否则出现相应的 ShopMill 标准画面。

状态	功能
19	基本画面
2	T, S, M, ...
30	工件零点
31	工件零点-/用户屏幕窗口
34	工件零点-/用户屏幕窗口
35	工件零点-/用户屏幕窗口
36	工件零点-/用户屏幕窗口
37	工件零点-/用户屏幕窗口
38	工件零点-/用户屏幕窗口
40	工件零点-/用户屏幕窗口
5	工件零点-测量边沿 Z
50	测量刀具
51	测量刀具- 手动长度 X /用户屏幕窗口
52	测量刀具- 手动长度 Z /用户屏幕窗口
53	测量刀具-放大镜*/用户屏幕窗口
54	测量刀具-/用户屏幕窗口
55	测量刀具-/用户屏幕窗口
56	测量刀具- 校正测量头*/用户屏幕窗口
57	测量刀具-/用户屏幕窗口
58	测量刀具-自动 Z*
59	测量刀具-自动 X*
4	Position
18	平面铣削*
80	切削*
90	-/用户屏幕窗口 (尾座)
1	ShopTurn 设置

手动加工 (带选项“手动机床”)

状态	功能
19	基本画面
50	测量刀具
51	测量刀具- 手动长度 X /用户屏幕窗口
52	测量刀具- 手动长度 Z /用户屏幕窗口
53	测量刀具-放大镜*/用户屏幕窗口
54	测量刀具-/用户屏幕窗口
55	测量刀具-/用户屏幕窗口
56	测量刀具- 校正测量头*/用户屏幕窗口
57	测量刀具-/用户屏幕窗口
58	测量刀具-自动 Z*
59	测量刀具-自动 X*
1300	直线
1400	钻削
1410	钻孔-中心
1420	钻孔-螺纹中心
1433	钻削-定中心*
1434	钻削-钻削*
1435	钻削-铰孔*
1440	钻削-深钻孔*
1453	钻削-螺纹钻削*
1454	钻削-螺纹铰削*
1500	车削
1513	车削 - 切削 1
1514	车削 - 切削 2
1515	车削 - 切削 3
1523	车削 - 切槽 1
1524	车削 - 切槽 2
1525	车削 - 切槽 3
1533	车削- E 型退刀槽
1534	车削- F 型退刀槽
1535	车削- 螺纹 DIN 退刀槽
1536	车削- 螺纹退刀槽
1543	车削- 螺纹长度
1544	车削- 螺纹锥
1545	车削- 螺纹端面
1550	车削- 断削
1600	铣削*
1613	铣削-矩形腔*
1614	铣削-圆形腔*
1623	铣削-矩形轴颈*

4.5 可选状态的列表

状态	功能
1624	铣削-圆形轴颈*
1633	铣削-长形腔*
1634	铣削-圆形槽*
1640	铣削-多边形*
1670	铣削-雕刻*
1730	模拟- 3 窗口视图*
1740	模拟- 侧视图*
1750	模拟- 端视图*
90	-/用户屏幕窗口 (尾座)
1	ShopTurn 设置

MDA

状态	功能
20	MDA

自动加工

状态	功能
200	当前的程序段显示
210	程序控制
220	程序段查找
230	-/用户屏幕窗口
242	绘制- 3 窗口视图*
243	绘制- 侧视图*
244	绘制- 端视图*
250	设置

Program Manager

状态	功能
300	NC 目录
310	零件程序*
320	子程序*
330	用户目录 1*
340	用户目录 2*
350	用户目录 3*
360	用户目录 4*
380	标准循环*
381	制造商循环*
382	用户循环*

状态	功能
383	用户目录 5*
384	用户目录 6*
385	用户目录 7*
386	用户目录 8*

Program

状态	功能
400	工作计划 / G 代码编辑器
412	模拟- 3 窗口视图*
413	模拟- 侧视图*
414	模拟- 端视图*

报警信息

状态	功能
500	显示信息
510	-/用户屏幕窗口
520	-/用户屏幕窗口

刀具零点偏移

状态	功能
600	刀具表
610	刀具磨损
620	OEM 刀具列表*
630	刀具库
640	零点偏移
650	R 参数
660	-/用户屏幕窗口
670	主轴
680	用户数据
690	机床数据

操作区“Custom (定制)”

5.1 供货状态和应用

概述

前面描述的操作区可以通过“补充操作界面”方式补充和修改。
补充只能放置在还未使用的软键上。

使用下面说明的方法都可以在 HMI 内置 sl 和 HMI 高级上设计一个独立的操作区，该操作区所有的 8 个水平和 8 个垂直软键都供用户专用的操作界面使用。

操作区的默认名为“定制”。

对于带热键区的 OP (例如 : OP 010, OP 010C) 可以通过以下按键直接选择操作区“定制”：

- 热键“定制”
- 第 1 个水平扩展栏的软键 4 (预设)

供货状态下的属性

供货状态下，“定制”操作区为覆盖整个本地菜单区域的空窗口，其中，标题可以设计。此外，“定制”操作区还允许在全局菜单的操作区显示栏内输入可配置的文本。

全部软键都是空的，用户可以通过“补充操作界面”任意定义软键。

在离开并再次返回“定制”操作区后，离开操作区“定制”前的有效画面生效。

- **HMI 高级**

“定制”操作区属于供货的标准范畴，可以通过文件 REGIE.INI 中的条目释放或者定制在其他任意一个水平键上。

- **HMI 内置 sl**

“定制”操作区属于标准配置。

它与应用程序磁盘一起提供，并且可以由客户通过软键配置套件 (SCK.exe) 定制在任意一个水平软键上。

5.2 激活操作区

HMI 高级

操作区在文件 REGIE.INI 中释放并定制在软键上。

举例：

通过第 1 个扩展栏的水平软键 4 激活（预设）。

- 条目在段落 [TaskConfiguration] 中：

```
[TaskConfiguration]
Task11 = name := custom, Timeout := 12000
```

- 在文件 REGIE.INI 中也可以选择该操作区用于标准启动。条目在段落 [Miscellaneous] 中：

```
[Miscellaneous]
PoweronTaskIndex = 11
```

当不选择“定制”操作区或者在启动时不激活情况下，相应的行可以通过行开始处的注释标记“;”退出激活。通过更改任务号可以将操作区定制在另一个软键上。

HMI 内置 sl

您可以通过显示机床数据 MD 9016：MM_SWITCH_TO_AREA 确定，应该在哪个操作区中运行 HMI 内置 sl，此时也可以规定操作区“定制”。

机床数据中的值规定定制所需的操作区的软键号码。

预设置：	12
水平软键 1 – 8：	1 – 8
扩展栏软键	9 – 16

用于应用程序 CUSTOM (定制) 附属的 HMI 软件分析文件 CUSTOM.INI 并决定是否要显示操作区。对于 HMI 内置 sl，文件包含段落 [Activate]：

```
[Activate]
Activate=True
```

操作区可以通过软件配置套件 (SCK.EXE)，菜单项“修改配置”由客户在任意一个水平软键上定制。

为了使用标准启动的定制操作区，软键号必须输入在显示机床数据 MD 9016：MM_SWITCH_TO_AREA 中。

如果显示机床数据有值 -1，则 HMI 内置 sl 在该操作区中启动，该操作区已通过软件配置套件 (SCK) 指定作为启动操作区。在供货状态下，基本栏的软键 1 在操作区“加工”中。

通过键激活

- **热键**
通过操作面板 OP 010，OP 010C 上的“定制”热键总是可以激活该操作区（预设）。可以进行其它和附加的设计。
- **水平软键**
通过第 1 个扩展栏的水平软键 4 激活（预设）。在 HMI 高级上可以更改软键的布局，而在 HMI 内置 sl 上则需要借助软键配置包(SCK)。

操作区域切换时的特性

如果由“定制”操作区切换到另一个操作区并再次返回，则在返回后显示离开“定制”操作区前已激活的画面。

5.3 定义开始对话框

概述

文件 CUSTOM.INI 中包含设计开始对话框的条目；对于 HMI 高级，文件 RE_xx.INI 可用于定义操作区的名称。

- **对话框标题**
在段落 [Header] 中可以输入一个带有对话框标题的文本。
标题可以是文本或者是报警文本号，从而可以根据不同语言设计标题：

```
[Header]
Text="XY 特殊功能"
Text=$80XXX
预设： Text="Custom"
```

- **开始对话框中的图形**
在段落 [Picture] 中可以输入在启动应用程序时显示的图形的路径。

```
[Picture]
Picture=\directory\bild.bmp
```

- **操作区名称**

HMI 高级	HMI 内置 sl
给定的名称出现在所显示的开始对话框的左上方。 。 [HSoftkeyTexts] HSK11 = "Custom"	给定的名称出现在所显示的开始对话框的左上方。 [Taskname] 标记： Text = \$80xxx

- 软键标签

HMI 高级	HMI 内置 sl
文件 RE_xx.IN 中的操作区名称显示在配置的软键上。 "xx" 为语言缩写。	规定的文本显示在分配的软键上。 如果没有规定，则标准设置为“Custom”。 [Softkey] Text = \$80xxx

操作区“定制”的其他所有单元，例如：软键栏或者输入/输出栏以及附属功能必须通过“补充操作界面”设计。

在应用程序“定制”中**所有**软键作为“补充操作界面”的软键使用。

根据上述说明，在文件 CUSTOM.COM 中 (类似与供货范围内的 AEDITOR.COM) 设计软键。

对于 HMI 内置 sl，文件 COMMON.COM 必须参照文件 COMMON.COM (用户登入软键)。而供货状态下的 COMMON.COM，操作区“定制”的所有软键都参照 CUSTOM.COM。

设计环境

6.1 供货范围

概述

用于编译操作界面设计文件和编制或者激活上述功能的软件都是 HMI 软件供货范围的一部分；同样用于建立设计文件内容的 ASCII 编辑器（程序编辑器）也是 HMI 软件供货范围的一部分。

产品

NCU 上的产品 ShopMill 和 ShopTurn 是以 HMI 内置 sl 为基础。HMI 内置 WIN32 和 HMI 高级组合可以在一个 HW 平台上运行。为此该产品的操作系统可以作为 HMI 高级的选择之一在 HW 平台上运行。

系统区别

由于不同的硬件，设计文件的保存也各不相同：

- PCU 50 上的 HMI 高级使用硬盘。
- HMI 内置 sl 只使用 CF 卡上的工作存储器 and 用户存储器。
- PCU 50 或者装有 Windows 的计算机上的 HMI 内置 WIN32 分析硬盘上的设计文件，除此之外仍然和 HMI 内置 sl 一样运行。

图形描述

如果要在设计的对话框中使用图形，则还需要一个合适的图形程序（例如：微软画笔）。

参见

查找方案的原理 (页 186)

6.2 创建设计文件

6.2.1 使用文件 COMMON.COM

概述

在使用 HMI 高级时不需要 COMMON.COM 中的条目。

HMI 内置 sl 上该中央控制文件包含下列信息：

- 分配登入软键给设计文件
- 分配画面号给 PLC 接口 DB19 中的设计文件
- 控制条目 (LOG 文件的大小、设计文件临时目录中可用的存储位置)。

HMI 高级/ HMI 内置 sl 的预设：

基本画面	水平软键	设计文件
加工 JOG	1	MA_JOG.COM
加工 MDA	1	MA_MDA.COM
加工 自动方式	2	MA_AUTO.COM
Parameter	7	PARAM.COM
Program	8	PROG.COM
Service	7	SERVICE.COM
Diagnose	7	DIAG.COM
Startup	7	STARTUP.COM
扩展软键栏	6, 7	
编辑器 (已占用)	2, 3, 4, 5	AEDITOR.COM
编辑器	6	AEDITOR.COM
扩展软键栏	6, 7	

命名惯例和文件大小

- HMI 内置 sl
 - 所有文件名必须符合 DOS 规定(xxxxxxx.com)。
 - 最多允许 10 个设计文件。
 - 帮助图形的颜色格式：bmp 格式的 256 色位图。
 - 由所用 CF 卡的大小决定图形和设计文件的存储位置需求。可以不规定文件最大数量。
- HMI 高级
 - 根据上述给定的顺序在目录中查找带有登入软键名称的文件。
 - 如果在不同的目录中存放相同文件名的文件，则找到的文件为按照查找顺序首先找到的文件。设计文件的大小没有特别的限制。然而应该注意，文件越大，文件处理速度就越慢。

6.2.2 文件 COMMON.COM 的结构

概述

文件 COMMON.COM 连同循环一起提供。它包含特定硬件设置的不同段落。对于 HMI 内置 sl，段落[MMC_DOS] 关系到“补充操作界面”。

设计 COMMON.COM

句法	参数 = 值	;	在参数、'=' 和 值之间可以有多个任意空格。
说明	[MMC_DOS]	;	DOS 段开始
		;	分号“;”后行内的文本是注释语句，不作分析。
参数	所有参数都是可选的。		
	SCxxx= 文件	软键连接：连接软键和设计文件	
		"xxx" 表示登入软键的内部软键属性。	
		软键属性必须紧跟在 SC... 后面。	
		仅显示已定义软键连接的软键。	
	HCyyy= 文件	热键连接：连接实际按键和设计文件。	
		"yyy" 表示登入热键的热键属性，热键属性必须紧跟在 HC... 后面。只有已定义热键连接的热键才有效。	
	文件	包含软键和对话框定义的配置文件。文件名最多允许 8 个字符长。文件扩展名用点隔开。	
		举例：SC101= my_file.com ; (my_file.com 在 HMI, NC 上)	

注意

COMMON.COM 中的更改只有在重新启动后才生效。

控制项

- 句法 : `CHK_FILE_EXIST=ram`
- 名称 : 控制标志：规定每次的设计文件是否要由 NC 复制，或者是否要进行检查，是否该文件已位于 HMI 的 RAM 驱动器上了。
- 参数 : `ram` 可能的值：
- 0: 不检查文件是否已存放在 临时目录上。该模式仅在 NC 上在线建立设计文件阶段设置。为此 NC 设计文件的更改立即在 HMI 内置 `sl` 上生效，这导致图形建立的速度较慢。
 - 1: 预设（如果在 `CHK_FILE_EXIST` 未规定情况下）：设计文件只向临时目录中读入一次并于日后从那里进行处理。这表明：运行时间得到改善，然而 NC 上设计文件中的改变不会生效。
- 句法 : `LOGSIZE=kB`
- 名称 : 设立 HMI 的临时目录上的 LOG 文件名称为 `ERROR.COM`，它的大小通过该参数确定。
- 参数 : `kB` LOG 文件的大小，以 KB 为单位（最大可以是 64 KB）。
- 句法 : `RAMDISK_SIZE = kBrd`
- 名称 : `RAMDISK_SIZE` 可以在任意位置上，在段内或者段外。如果该单元多次出现，则取**第一次出现**的单元。
如果在复制过程后超过设置的大小，则在下一次复制过程前删除临时目录中的所有 COM 文件。（操作区切换时后台的对话框的文件保持不变。）
- 参数 : `kBrd` 可支配的 `RAMDISK` 大小，以 kB 为单位。
预设置：300 kByte
(如果没有规定单元 `RAMDISK_SIZE`)

COMMON.COM 举例

```
[ MMC_DOS]
sc101=verzahn.com           ; MASCHINE
sc111=mda.com               ; 文件位于 HMI 的闪存存储器上。
sc122=auto.com
sc207=param.com             ; PARAMETER
sc314=aeditor.com           ; 程序编辑器
sc315=aeditor.com
sc316=aeditor.com
sc407=dienste.com           ; Service
sc507=diagnose.com          ; DIAGNOSE
sc607=inbetrn.com           ; Startup
sc826=cmm.com               ; ShopMill, 加工, AUTO
sc857=cmm.com               ; 报警/信息
sc858=cmm.com
sc867=cmm.com               ; 刀具, 零偏
CHK_COMMON.COM=1            ; 更快处理 HMI
LOGSIZE=30                  ; 故障纪录 ( LOG 文件 ) 的大小 : 30 kB

[PLC_SELECT]                ; 定义可由 PLC 调用的画面
PC1= CYC82 bohren.com        ; 画面 1
PC2= CYCLE90, gewfraes.com   ; 画面 2
```

6.2.3 设计登入软键

概述

借助于规定的登入软键可以激活分配的设计文件。固定定义可以对话框的登入软键。其它登入软键是不可以的。登入软键在各个操作区各有不同。

编程

句法	SCxxx= 文件
名称	软键连接：连接软键和设计文件
	"xxx" 表示登入软键的内部软键属性
参数	文件 设计文件名称

操作区中的登入点

操作区	SCxxx	初始对话框	
MASCHINE	SC101	加工 JOG	水平软键 1
	SC111	加工 MDA	水平软键 1
	SC122	加工 AUTO	水平软键 2
PARAMETER	SC207	参数基本画面	水平软键 7
PROGRAM	SC308	程序基本画面	水平软键 8
	SC312	1. 编辑器软键行	水平软键 2
	SC313	编辑器的第 1 软键栏	水平软键 3
	SC314	编辑器的第 1 软键栏	水平软键 4
	SC315	编辑器的第 1 软键栏	水平软键 5
	SC316	编辑器的第 1 软键栏	水平软键 6
	SC326	编辑器的第 2 软键栏	水平软键 6
	SC327	编辑器的第 2 软键栏	水平软键 7
Service	SC407	通讯基本画面	水平软键 7
DIAGNOSE	SC507	诊断基本画面	水平软键 7
开机调试	SC607	开机调试基本画面	水平软键 7
	SC616	编辑器的第 2 软键栏	水平软键 6
	SC617	编辑器的第 2 软键栏	水平软键 7

章节“登入软键表”中给出的名称是预设。然而附属的文件仍然必须在 NC 或者 HMI 上由用户设立。

参见

登入软键表 (页 191)

6.2.4 和语言相关的文本

概述

对话框中和语言相关的文本保存在 ASCII 文本文件中；句法和报警文本文件一致。

可以使用和语言相关的文本：

- 软键标记
- 标题
- 辅助文本
- 其它任意文本

允许的文件名

按如下方式确定文件名：

Alsc.txt	用于西门子标准循环、和语言相关的文本
Almc.txt	用于制造商循环、和语言相关的文本
Aluc.txt	和语言相关的用户文本

文本输入形式

句法	8xxxx 0 0 "文本"
说明	文件中文本号和文本的排列
参数	xxxx 5000 至 9899 预留于用户的文本识别号范围。 号码必须是唯一的。
	"文本" 显示在对话框中的文本

两个通过空格分开的参数 2 和 3 是用于报警文本输出的控制符号。
鉴于带有报警文本的文本格式的一致性，在任何情况下控制符号必须置零。

在文本中可以有下列控制符号：

%n	换行
%@x	x. 轴的轴名称 (x 为轴编号)，仅适用于 HMI 内置 sl 显示轴名称 (HMI 内置 sl 和 HMI 高级)： NC 存取包含各轴名称的相应机床数据，文本组合通过包含的字符串功能。

举例：

85000 0 0	"退回平面"
85001 0 0	"钻削深度"
85002 0 0	"螺距"
85003 0 0	"凹槽半径"

6.3 设计文件的存档结构

6.3.1 HMI 内置 sl

存档

在 Linux 环境下，将用户设计文件复制到 CF 卡的目录 `/user/sinumerik/hmi/proj` 下（适用于所有“标准”的用户对话框，即：除用于循环支持外的所有对话框）。将供用户循环支持使用的用户设计文件复制到 CF 卡的目录 `/user/sinumerik/cycles/proj` 下。一般来说，所有文件未压缩地直接复制到各目录下。

与此类似，将制造商的设计文件复制到目录 `/oem/sinumerik/...` 下。

路径	内容
<code>/card/user/sinumerik/hmi/proj</code>	用户设计文件（“标准”用户对话框的 com 文件，即：除用于循环支持外的所有对话框）：
<code>/card/user/sinumerik/cycles/proj</code>	用于用户循环支持的的用户设计文件
<code>/card/oem/sinumerik/cycles/proj</code> <code>/card/oem/sinumerik/hmi/proj</code>	制造商设计文件
<code>/card/user/sinumerik/cycles/ico/icoxxx</code> <code>/card/oem/sinumerik/cycles/ico/icoxxx</code> <code>/card/user/sinumerik/hmi/ico/icoxxx</code> <code>/card/oem/sinumerik/hmi/ico/icoxxx</code>	位图
<code>/card/user/sinumerik/hmi/cfg</code> <code>/card/oem/sinumerik/hmi/cfg</code>	Ini 文件
<code>/card/user/sinumerik/hmi/ico/icoxxx</code> <code>/card/oem/sinumerik/hmi/ico/icoxxx</code>	标题图标
<code>/card/user/sinumerik/cycles/lng/xxx</code> <code>/card/oem/sinumerik/cycles/lng/xxx</code> <code>/card/user/sinumerik/hmi/lng/xxx</code>	附属的文本(aluc.txt oder aluctx.s0x)，其中 xxx 表示语言

供货时，“程序”操作区设置了 3 个 USB 驱动器并能访问 CF 卡。

USB 存储器上设置了下列目录结构：

```

\cycles
  \cycles\proj      (com 文件)
  \cycles\prog     (用户循环 (.spf))
  \cycles\lng      (语言目录- 仅包含子目录)
  \cycles\lng\xxx  (语言目录，例如：deu, eng... – 每种语言一个目录)。
                  (这里保存每个语言的aluc.txt文件。)
  \cycles\ico      (图形目录- 仅包含各分辨率的子目录)
  \cycles\ico\ico640 对于分辨率 640*480 的图形目录为.bmp 或 .bin
  \cycles\ico\ico800
  \cycles\ico\ico1024

```

分配给其他操作区的对话框：

\hmi\proj

\hmi\lng\...

\hmi\ico\...

在操作区“程序”中，把整个目录\cycles 或 \hmi 从 USB 存储器复制到 CF 卡的目录 /user/sinumerik 中。

6.3.2 HMI 高级

概述

在使用 HMI 高级时不需要控制文件条目。

在目录中按给出的顺序查找设计文件。

如果在不同的目录中存放相同文件名的文件，则找到的文件为按照查找顺序首先找到的文件。

在 PG/标准 PC 上测试

在 PG/标准 PC 上测试设计完成的对话框时，需满足以下边界条件：

- 已经在 PC/PG 上安装 HMI 高级软件的 PC 版本。
- 目录结构与 HMI 高级上的结构一致。
- 故障记录建立在：\DH\COM.DIR\ERROR.COM

存储报警文本文件

报警文本文件保存在下列目录中：\DH\MB.DIR\

文件名：ALUC_XX.COM

文献：HMI 高级版开机调试手册

6.4 HMI 系统使用共同的硬件平台时的查找方案

6.4.1 查找方案的原理

概述

HMI 内置 WIN32 如同 HMI 高级一样以相同的路径查找用于“补充操作界面”的配置文件。开始处是数据维护路径。

当 HMI 高级和 HMI 内置 sl 在同一硬件平台上共同运行时，下列描述中的查找顺序对于 NCU 上的 ShopMill / ShopTurn 比较重要。在这种情况下，HMI 内置 sl 上运行的 NCU 上的 ShopMill / ShopTurn 下的“补充操作界面”使用相同的配置文件，如同 HMI 高级下的操作。

框架条件

HMI 内置 WIN32 分析配置文件 DH.INI 的段 [DHSTART] 的特性 mmchome，该配置文件包含数据维护路径的根目录。DH.INI 必须存放在 BIN 目录中，在该目录中启动 MMC0.EXE。数据维护路径的数据最大长度为 100 个字符。

引导启动

在启动时 HMI 内置 WIN32 根据注册表中的条目首先确定 HMI 高级安装在哪里。在注册表规定的目录内在子目录 `..user`, `..oem`, `..add_on`, `..mmc2` 中以所述的顺序查找文件 DH.INI。如果那里没有找到 DH.INI，则在 HMI 内置 WIN32 的当前设置的目录中查找文件。

举例：

如果 HMI 高级可在 `f:\HMI\HMI Advanced` 上找到；然后按照下列顺序查找 DH.INI：

- `F:\HMI\HMI-Advanced\user`
- `F:\HMI\HMI-Advanced\oem`
- `F:\HMI\HMI-Advanced\add_on`
- `F:\HMI\HMI-Advanced\mmc2`

查找顺序控制大小

HMI 高级的查找顺序

- CUS 目录，在 `dh.ini` 内特定的数据维护路径中
- CMA 目录，在 `dh.ini` 内特定的数据维护路径中
- CST 目录，在 `dh.ini` 内特定的数据维护路径中
- COM 目录，在 `dh.ini` 内特定的数据维护路径中

RAMDISK 上的目录：

Filename.bin

Filename.bmp .

不带路径的位图名称

对于 HMI 高级，这种文件有文件后缀 .bin

在没有规定文件夹和文件名不包含路径的情况下，查找顺序为：

Filename.bin ，在数据维护路径的**CUS**目录中

Filename.bmp ，在数据维护路径的**CUS**目录中

Filename.bin ，在文件夹*Filename.bi_* 中，数据维护路径的**CUS**目录下

Filename.bmp ，在文件夹*Filename.bm_* 中，数据维护路径的**CUS**目录下

Filename.bin ，在数据维护路径的**CUS\分辨率**目录中

Filename.bmp ，在数据维护路径的**CUS\分辨率**目录中

Filename.bin ，在文件夹*Filename.bi_* 中，数据维护路径下的 **CUS\分辨率**目录下。

Filename.bmp ，在文件夹*Filename.bm_* 中，数据维护路径下的 **CUS\分辨率**目录下。

Filename.bin ，在数据维护路径的**CMA**目录中

Filename.bmp ，在数据维护路径的**CMA**目录中

Filename.bin ，在文件夹*Filename.bi_* 中，数据维护路径的**CMA**目录下

Filename.bmp ，在文件夹*Filename.bm_* 中，数据维护路径的**CMA**目录下

Filename.bin ，在数据维护路径的**CMA\分辨率**目录中

Filename.bmp ，在数据维护路径的**CMA\分辨率**目录中

Filename.bin ，在文件夹*Filename.bi_* 中，数据维护路径下的 **CMA\分辨率**目录下。

Filename.bmp ，在文件夹*Filename.bm_* 中，数据维护路径下的 **CMA\分辨率**目录下。

Filename.bin ，在数据维护路径的**CST**目录中

Filename.bmp ，在数据维护路径的**CST**目录中

Filename.bin ，在文件夹*Filename.bi_* 中，数据维护路径的**CST**目录下

Filename.bmp ，在文件夹*Filename.bm_* 中，数据维护路径的**CST**目录下

Filename.bin ，在数据维护路径的**CST\分辨率**目录中

Filename.bmp ，在数据维护路径的**CST\分辨率**目录中

Filename.bin ，在文件夹*Filename.bi_* 中，数据维护路径下的 **CST\分辨率**目录下。

Filename.bmp ，在文件夹*Filename.bm_* 中，数据维护路径下的 **CST\分辨率**目录下。

Filename.bin 在当前目录中 (Bin 目录)

Filename.bmp 在当前目录中 (Bin 目录)

Filename.bin ，在文件夹*Filename.bi_* 中，当前目录下 (Bin 目录)

Filename.bmp ，在文件夹*Filename.bm_* 中，当前目录下 (Bin 目录)

Filename.bin ，在文件夹**CUS.ARJ** 中，数据维护路径的**CUS**目录下

Filename.bmp ，在文件夹**CUS.ARJ** 中，数据维护路径的**CUS**目录下

Filename.bin ，在文件夹**CUS.ARJ** 中，数据维护路径的**CUS\分辨率**目录下

Filename.bmp ，在文件夹**CUS.ARJ** 中，数据维护路径的**CUS\分辨率**目录下

Filename.bin ，在文件夹**CUS.ARJ** 中，数据维护路径的**CMA**目录下

Filename.bmp , 在文件夹CUS.ARJ 中, 数据维护路径的CMA目录下
Filename.bin , 在文件夹CUS.ARJ 中, 数据维护路径的CMA\分辨率目录下
Filename.bmp , 在文件夹CUS.ARJ 中, 数据维护路径的CMA\分辨率目录下
Filename.bin , 在文件夹CUS.ARJ 中, 数据维护路径的CST目录下
Filename.bmp , 在文件夹CUS.ARJ 中, 数据维护路径的CST目录下
Filename.bin , 在文件夹CUS.ARJ 中, 数据维护路径的CST\分辨率目录下
Filename.bmp , 在文件夹CUS.ARJ 中, 数据维护路径的CST\分辨率目录下
对于 CMA.ARJ, 步骤 29 至 40 现在重复进行
Filename.bin 在文件夹 CMA.ARJ 中, 数据维护路径的 CUS 目录下
....
Filename.bmp在文件夹 CMA.ARJ中, 数据维护路径的 CST\分辨率目录下
对于 CST.ARJ, 步骤 41 至 52 现在重复进行
Filename.bin 在文件夹 CST.ARJ 中, 数据维护路径的 CUS 目录下
....
Filename.bmp, 在文件夹 CST.ARJ 中, 数据维护路径的 CST\分辨率目录下
Filename.bin, 在文件夹 CUS.ARJ 中, 当前目录下 (Bin 目录)
Filename.bmp, 在文件夹 CUS.ARJ 中, 当前目录下 (Bin 目录)
Filename.bin, 在文件夹 CMA.ARJ 中, 当前目录下 (Bin 目录)
Filename.bmp, 在文件夹 CMA.ARJ 中, 当前目录下 (Bin 目录)
Filename.bin, 在文件夹 CST.ARJ 中, 当前目录下 (Bin 目录)
Filename.bmp, 在文件夹 CST.ARJ 中, 当前目录下 (Bin 目录)
Filename.bxx

部分二进制文件, 出于与实际模式和保护模式开始的兼容性原因还必须处理。
这些文件仅在当前设置的目录中查找。

6.4.2 COMMON.COM 的查找方案

COMMON.COM 的存档位置

中央控制文件 COMMON.COM 必须位于 NC 的下列目录中的一个：

- CUS: 用户循环目录
- CMA: 制造商循环目录
- CST: 标准循环目录
- COM: 注释目录

目录按规定的顺序查找 COMMON.COM。分析首先找到的符合该名称的文件。

此外，对于 HMI 高级，控制文件 COMMON.COM 在数据维护的路径下：

..\dh\cus.dir

..\dh\cma.dir

..\dh\cst.dir

..\dh\com.dir

注意

必须根据 HMI 高级(ShopMill / ShopTurn)重新启动 HMI 内置 WIN32，文件中的修改才能生效。

6.4.3 图形的查找方案

查找顺序

HMI 内置 sl 的图形（位图）的扩展的查找方案也适用于“补充操作界面”的图形。

对于 HMI 内置 WIN32（ShopMill/ShopTurn），在一个硬件平台上与 HMI 高级一起，可以规定一个相关的数据维护路径。

如果已定义数据维护路径，则在当前目录前要首先处理，为此 HMI 内置 WIN32 和 HMI 高级使用 **相同的图形**。

新的查找结构也包括与补充操作界面相关的数据维护路径目录，以及图形可行的文件夹（cus.arj, cma.arj, cst.arj）。

原则上总是首先查找单个文件，然后才在可行的文件夹中查找。

为此得到下列图形的查找顺序：

- 文件夹前的单个图形（首先查找 .bin，然后查找 .bmp。
在单个文件后在仅包含一个文件的文件夹中查找（.bi_，然后 .bm_）。
- 先是带有路径的位图名称，然后是不带路径的位图名称

参见

查找方案的原理 (页 186)

附录

A.1 登入软键表

ShopMill 和 ShopTurn 的登入软键

ShopMill	SCxxx	初始对话框	
	SC818	手动加工操作区 (大对话框)	水平软键 8
	SC8181	手动加工操作区 (中等对话框)	水平软键 8
	SC8182	手动加工操作区 (小对话框)	水平软键 8
	SC8131	手动加工操作区-工件零点	垂直软键 1
	SC8132	手动加工操作区-工件零点	垂直软键 2
	SC8133	手动加工操作区-工件零点	垂直软键 3
	SC8134	手动加工操作区-工件零点	垂直软键 4
	SC8135	手动加工操作区-工件零点	垂直软键 5
	SC8136	手动加工操作区-工件零点	垂直软键 6
	SC8137	手动加工操作区-工件零点	垂直软键 7
	SC8141	手动加工操作区-刀具测量	垂直软键 1
	SC8142	手动加工操作区-刀具测量	垂直软键 2
	SC8143	手动加工操作区-刀具测量	垂直软键 3
	SC8144	手动加工操作区-刀具测量	垂直软键 4
	SC8145	手动加工操作区-刀具测量	垂直软键 5
	SC8146	手动加工操作区-刀具测量	垂直软键 6
	SC8147	手动加工操作区-刀具测量	垂直软键 7
	SC826	自动加工操作区 (大对话框)	水平软键 6
	SC8261	自动加工操作区 (中等对话框)	水平软键 6
	SC8262	自动加工操作区 (小对话框)	水平软键 6
	SC8426	程序操作区-钻削	垂直软键 6
	SC8436	程序操作区-铣削	垂直软键 6
	SC8454	程序操作区-不同的	垂直软键 4
	SC8951	程序操作区-不同的-工件零点	垂直软键 1
	SC8952	程序操作区-不同的-工件零点	垂直软键 2
	SC8953	程序操作区-不同的-工件零点	垂直软键 3
	SC8954	程序操作区-不同的-工件零点	垂直软键 4
	SC8955	程序操作区-不同的-工件零点	垂直软键 5
	SC8956	程序操作区-不同的-工件零点	垂直软键 6

ShopMill	SCxxx	初始对话框	
	SC8957	程序操作区-不同的-工件零点	垂直软键 7
	SC8961	程序操作区-不同的-刀具测量	垂直软键 1
	SC8962	程序操作区-不同的-刀具测量	垂直软键 2
	SC8963	程序操作区-不同的-刀具测量	垂直软键 3
	SC8964	程序操作区-不同的-刀具测量	垂直软键 4
	SC8965	程序操作区-不同的-刀具测量	垂直软键 5
	SC8966	程序操作区-不同的-刀具测量	垂直软键 6
	SC8967	程序操作区-不同的-刀具测量	垂直软键 7
	SC857	信息 / 报警操作区	水平软键 7
	SC858	信息 / 报警操作区	水平软键 8
	SC867	刀具 / 零点偏移操作区	水平软键 7
	SC8492	程序操作区-G 代码编辑器	水平软键 2 (循环支持轮廓) *
	SC8493	程序操作区-G 代码编辑器	水平软键 3 (循环支持 钻削) *
	SC8494	程序操作区-G 代码编辑器	水平软键 4 (循环支持 铣削) *
	SC8495	程序操作区-G 代码编辑器	水平软键 5 (循环支持 车削) *
	SC8496	程序操作区-G 代码编辑器	水平软键 6*
	SC8406	程序操作区-G 代码编辑器 (扩展范围)	水平软键 6 (测量循环支持) *
	SC8407	程序操作区-G 代码编辑器 (扩展范围)	水平软键 7 (测量循环支持) *

* 即西门子对话框。

ShopTurn	SCxxx	初始对话框	
	SC818	手动加工操作区 (大对话框)	水平软键 8
	SC8181	手动加工操作区 (中等对话框)	水平软键 8
	SC8182	手动加工操作区 (小对话框)	水平软键 8
	SC8131	手动加工操作区-工件零点	垂直软键 1
	SC8132	手动加工操作区-工件零点	垂直软键 2
	SC8133	手动加工操作区-工件零点	垂直软键 3
	SC8134	手动加工操作区-工件零点	垂直软键 4
	SC8135	手动加工操作区-工件零点	垂直软键 5
	SC8136	手动加工操作区-工件零点	垂直软键 6
	SC8137	手动加工操作区-工件零点	垂直软键 7
	SC8141	手动加工操作区-刀具测量	垂直软键 1
	SC8142	手动加工操作区-刀具测量	垂直软键 2
	SC8143	手动加工操作区-刀具测量	垂直软键 3
	SC8144	手动加工操作区-刀具测量	垂直软键 4
	SC8145	手动加工操作区-刀具测量	垂直软键 5
	SC8146	手动加工操作区-刀具测量	垂直软键 6

ShopTurn	SCxxx	初始对话框	
	SC8147	手动加工操作区-刀具测量	垂直软键 7
	SC826	自动加工操作区 (大对话框)	水平软键 6
	SC8261	自动加工操作区 (中等对话框)	水平软键 6
	SC8262	自动加工操作区 (小对话框)	水平软键 6
	SC8246	程序操作区-钻削	垂直软键 6
	SC9436	程序操作区-车削	垂直软键 6
	SC9456	程序操作区-铣削	垂直软键 6
	SC8454	程序操作区-不同的	垂直软键 4
	SC857	信息 / 报警操作区	水平软键 7
	SC858	信息 / 报警操作区	水平软键 8
	SC867	刀具 / 零点偏移操作区	水平软键 7
	SC8492	程序操作区-G 代码编辑器	水平软键 2 (循环支持轮廓) *
	SC8493	程序操作区-G 代码编辑器	水平软键 3 (循环支持 钻削) *
	SC8494	程序操作区-G 代码编辑器	水平软键 4 (循环支持 铣削) *
	SC8495	程序操作区-G 代码编辑器	水平软键 5 (循环支持 车削) *
	SC8496	程序操作区-G 代码编辑器	水平软键 6*
	SC8406	程序操作区-G 代码编辑器 (扩展范围)	水平软键 6 (测量循环支持 - 车削) *
	SC8407	程序操作区-G 代码编辑器 (扩展范围)	水平软键 7 (测量循环支持 - 铣削) *

* 即西门子对话框。

A.2 颜色表

可以使用的颜色

对于 HMI 高级 和 HMI 内置
sl，提供统一的颜色表用于对话框设计（各个标准颜色的部分数量）。

序号	颜色
1	黑色
2	红褐色
3	深绿色
4	浅灰
5	深蓝色
6	蓝色
7	红色
8	棕色
9	黄色
10	白色

每个 HMI 程序中，颜色可能略有不同。

HMI 高级

对于 HMI 高级中的位图，在字符程序中必须使用软件提供的当前颜色表。

HMI 内置 sl

对于 HMI 内置 sl 中的位图，在字符程序中必须使用软件提供的当前颜色表。
颜色表取决于选项“新式”。

您可在工具箱 8x0d\examples_tools\wizard.bsp\hmi_emb\... 下找到颜色表。

文件名给出了使用各个表格的提示：

- **HMI_EMB_NEW_FASHION.PAL:**
该调色板用于“新式”的 HMI 内置 sl
可提供颜色索引 160 至 231。
- **HMI_EMB_OLD_AND_NEW_FASHION.PAL:**
该调色板用于 HMI 内置
sl，和新式老式无关。用该调色板建立的位图对于新老式外观上完全相同。
可提供颜色索引
160、163、184、187、196、199、204、205、207、217、219、220、221、223、226
和 228。

以前的颜色表 HMI_EMB.PAL 已由上述表格替换。仅允许使用颜色 160 至 231。
只有这样才能确保 HMI 内置 sl 和 HMI 高级 上的图形外观上完全相同。

在 Paint Shop Pro 下激活调色板：

- 文件 → 打开 → ...*.bmp
- 颜色 → 打开图形板 → ...*.PAL
- 通过选项“颜色索引：打开”使用调色板

系统颜色

对于单元（文本、输入栏、背景等等）可以从 10 种颜色中选择一种。
系统颜色上得以扩展，老式和新式之间颜色有所不同（例如：标题颜色）。

为了可以区别单一颜色和**设计相关**的颜色，确定单一颜色位于颜色索引 0 至 128 之间。
扩展框架内新引入的与设计相关的颜色从颜色索引 128 起定义。
以此即使对于单一颜色的扩展（最大至 128），也可以避免两种颜色类型的混淆。

新定义的颜色

索引	颜色描述	颜色	
		老式	新式
128	聚焦系统颜色	黄色	桔黄色
129	背景颜色	灰色	浅灰
130	标题颜色（激活）	黄色	蓝色
131	标题字体颜色（激活）	黑色	白色

A.3 可用的系统变量列表

名称	索引	说明
\$A_DBB[x]	x=字节号	数据位 由 / 位于 PLC 上
\$A_DBD[x]	x=偏移	数据双字 (32 位) 由 / 位于 PLC 上
\$A_DBR[x]	x=偏移	实数数据 (32 位) 由 / 位于 PLC 上
\$A_DBW[x]	x=偏移	数据字 (16 位) 由 / 位于 PLC 上
\$A_DLB[索引]	索引=偏移	左边区域中的数据位
\$A_DLD[索引]	索引=偏移	在左边区域中访问双字节数据
\$A_DLR[索引]	索引=偏移	左边区域中的实数数据
\$A_DLW[索引]	索引=偏移	左边区域中的数据字
\$A_IN[x]	x=数字输入号	HW 数字输入端的值
\$A_INA[x]	x=模拟输入号	HW 模拟输入端的值
\$A_INCO[x]	x=输入号	NC 比较仪输入端
\$A_INSE		安全可编程逻辑： NCK 外设外部输入端
\$A_INSED		安全可编程逻辑： 外部 NCK 输入端等价
\$A_INSEP		安全可编程逻辑： PLC 外设外部输入端
\$A_INSEPD		安全可编程逻辑： 等价外部 PLC 输入端
\$A_INSI		安全可编程逻辑： 内部 NCK 安全输入端
\$A_INSID		安全可编程逻辑： 等价内部 NCK 安全输入端
\$A_INSIP		安全可编程逻辑： 内部 PLC 安全输入端
\$A_LINK_TRANS_RATE		连接传输率
\$A_MARKERSI		安全可编程逻辑：NCK 标记
\$A_MARKERSIP		安全可编程逻辑： 等价 PLC 标记
\$A_OUT[x]	x=数字输出号	HW 数字输出端值
\$A_OUTA[x]	x=模拟输出号	HW 模拟输出端的值
\$A_OUTSE		安全可编程逻辑： NCK 外设外部输出端
\$A_OUTSED		安全可编程逻辑： 等价外部 NCK 输出端
\$A_OUTSEP		安全可编程逻辑： 外部 PLC 外设输出端
\$A_OUTSEPD		安全可编程逻辑： 等价外部 PLC 输出端

名称	索引	说明
\$A_OUTSI		安全可编程逻辑： 内部 NCK 安全输出端
\$A_OUTSID		安全可编程逻辑： 等价内部 NCK 安全输出端
\$A_OUTSIP		安全可编程逻辑： 内部安全 PLC 输出端 611D
\$A_OUTSIPD		安全可编程逻辑： 等价内部安全 PLC 输出端 611D
\$A_TIMERSI		安全可编程逻辑：NCK 计时器
\$A_PBB_IN[索引]	索引=偏移	IN 数据位
\$A_PBB_OUT[索引]	索引=偏移	OUT 数据位
\$A_PBD_IN[索引]	索引=偏移	IN 数据双字
\$A_PBD_OUT[索引]	索引=偏移	OUT 数据双字
\$A_PBR_IN[索引]	索引=偏移	IN 实数数据
\$A_PBR_OUT[索引]	索引=偏移	OUT 实数数据
\$A_PBW_IN[索引]	索引=偏移	IN 数据字
\$A_PBW_OUT[索引]	索引=偏移	OUT 数据字
\$A_TC_FCT		命令号
\$A_TC_LFN		源位置号
\$A_TC_LFO		源位置号
\$A_TC_LTN		目标位置号
\$A_TC_LTO		目标位置号
\$A_TC_MFN		源库
\$A_TC_MFO		源库号
\$A_TC_MTN		目标库号
\$A_TC_MTO		目标库号
\$A_TC_STATUS		命令状态
\$A_TC_THNO		刀架号
\$A_TC_TNO		T 号码
\$A_TOOLMLN[x]	x=刀具号 T	当前位置
\$A_TOOLMN[x]	x=刀具号 T	当前库
\$AA_COUP_ACT[x]	x=跟随主轴	跟随主轴当前的耦合状态
\$AA_COUP_OFFS[x]	x=轴	与导向轴 / 导向主轴的偏移，额定值
\$AA_COUP_OFFS[x]	x=主轴	同步主轴位置偏移，额定值一方
\$AA_CURR[x]	x=轴	轴或主轴的电流实际值
\$AA_DELT[x]	x=轴	工件坐标系中的驱动专用余程
\$AA_DTBB[x]	x=轴	基本坐标系中从程序段开始的驱动专用路程
\$AA_DTBW[x]	x=轴	工件坐标系中从程序段开始的驱动专用路程
\$AA_DTEB[x]	x=轴	基本坐标系中程序段结束的驱动专用路程
\$AA_DTEPB[x]	x=轴	基本坐标系中摆动进给的驱动专用余程
\$AA_DTEPW[x]	x=轴	工件坐标系中摆动进给的驱动专用余程
\$AA_DTEW[x]	x=轴	工件坐标系中程序段结束的驱动专用路程

A.3 可用的系统变量列表

名称	索引	说明
\$AA_EG_ACTIVE [a,b]	a=随动轴 b=导向轴	EG 耦合激活
\$AA_EG_AX[n,a]	n=索引导向轴 a=随动轴	EG 导向轴号码
\$AA_EG_DENOM [a,b]	a=随动轴 b=导向轴	EG 命名器耦合系数
\$AA_EG_NUM_LA[a]	a=随动轴	EG 导向轴数量
\$AA_EG_NUMERA [a,b]	a=随动轴 b=导向轴	EG 计数器耦合系数
\$AA_EG_SYN[a,b]	a=随动轴 b=导向轴	EG 同步位置 导向轴
\$AA_EG_SYNCDIFF[a]	a=轴命名符	EG 同步运行偏差
\$AA_EG_SYNFA[a]	a=随动轴	EG 同步位置 随动轴
\$AA_EG_TYPE[a,b]	a=随动轴 b=导向轴	EG 耦合类型
\$AA_ESR_ENABLE[a]	a=轴	ESR 轴使能
\$AA_ESR_ENABLE[a]	a=轴	ESR 许可
\$AA_ESR_STAT[a]	a=轴	ESR 状态
\$AA_ETRANS[x]	x=框架号	外部框架偏移
\$AA_FXS[x]	x=轴	运行到固定挡块后状态
\$AA_IBN[x]	x=轴	刀具托架实际值
\$AA_IEN[x]	x=轴	接收参照工件零点的有效刀具
\$AA_IM[x]	x=轴	刀具托架
\$AA_IW[x]	x=轴	刀具托架位置, 额定值
\$AA_LEAD_P[x]	x=轴	实际的导向值 - 位置
\$AA_LEAD_SP[x]	x=轴	模拟的导向值 - 位置
\$AA_LEAD_SV[x]	x=轴	模拟的导向值 - 速度
\$AA_LEAD_TYP[x]	x=轴	引导值源
\$AA_LEAD_V[x]	x=轴	实际的导向值 - 速度
\$AA_LOAD[x]	x=轴	驱动载荷 (以百分比为单位) (仅对于 611D)
\$AA_MM[x]	x=轴	机床坐标系中的测量值
\$AA_MM1[x]	x=轴	访问机床坐标系中的测量结果
\$AA_MM2[x]	x=轴	访问机床坐标系中的测量结果
\$AA_MM3[x]	x=轴	访问机床坐标系中的测量结果
\$AA_MM4[x]	x=轴	访问机床坐标系中的测量结果
\$AA_MW[x]	x=轴	工件坐标系中的测量值
\$AA_OFF[x]	x=轴	编程轴的叠加运动
\$AA_OFF_LIMIT[x]	x=轴	达到驱动专用极限值, 用于补偿 \$AA_OFF
\$AA_OSCILL_REVERSE_POS1[x]	x=轴	同步指令往复运动中的当前返回位置 1
\$AA_OSCILL_REVERSE_POS2[x]	x=轴	同步指令往复运动中的当前返回位置 2
\$AA_OVR[x]	x=轴	运动同步指令的驱动专用倍率
\$AA_POWER[x]	x=轴	驱动有效功率[Watt]
\$AA_S[x]	x=主轴号 (主轴号)	主轴转速, 标准值
\$AA_SOFTENDN[x]	x=轴	软件限位, 负方向

名称	索引	说明
\$AA_SOFTENDP[x]	x=轴	软件限位，正方向
\$AA_STAT[x]	x=轴	轴状态
\$AA_SYNA_MEM		运动同步指令的空余存储器
\$AA_SYNC[x]	x=轴	导向值耦合时随动轴耦合
\$AA_TORQUE[x]	x=轴	驱动力矩额定值单位[Nm]
\$AA_TYP[x]	x=轴	轴类型
\$AA_VACTB[x]	x=轴	驱动专用进给，实际值
\$AA_VACTM[x]	x=轴	驱动专用进给，额定值
\$AA_VACTW[x]	x=轴	驱动专用进给，实际值
\$AA_VC[x]	x=轴	驱动专用进给 附加路径进给补偿
\$AC_ALARM_STAT		ESR 报警状态
\$AC_AXCTSWA[CTn]	CTn=轴容器号	轴容器状态
\$AC_DELT		工件坐标系余程轨迹
\$AC_DRF[x]	x=轴	DRF 值
\$AC_DTBB		基本坐标系中与程序段开始的距离
\$AC_DTBW		工件坐标系与程序段开始的距离
\$AC_DTEB		基本坐标系中与程序段结束的距离
\$AC_DTEPB		基本坐标系中往复运动进给的余程
\$AC_DTEPW		工件坐标系中往复运动进给的余程
\$AC_DTEW		工件坐标系与程序段结束的距离
\$AC_FCT0[x]	x=多项式号	a0-系数 n. 同步措施多项式
\$AC_FCT1[x]	x=多项式号	a1-系数 n. 同步措施多项式
\$AC_FCT2[x]	x=多项式号	a2-系数 n. 同步措施多项式
\$AC_FCT3[x]	x=多项式号	a3-系数 n. 同步措施多项式
\$AC_FCTLL[x]	x=多项式号	下限值 n. 同步措施多项式
\$AC_FCTUL[x]	x=多项式号	上限值 n. 同步措施多项式
\$AC_FIFOx[y]	x=FIFONo (1-10) y=参数号	FIFO 同步措施变量
\$AC_MARKER[x]	x=标记号	用于运动同步指令的计数器
\$AC_MEA		已接通测量探头
\$AC_OVR		同步指令的轨迹倍率
\$AC_PARAM[x]	x=参数号	运动同步指令动态参数
\$AC_PATHN		标准轨迹参数
\$AC_PLTBB		基本坐标系中程序段开始的轨迹路程
\$AC_PLTEB		基本坐标系中程序段结束轨迹行程
\$AC_PRESET[x]	x=轴	PRESET 值
\$AC_PROG		程序状态
\$AC_RETPOINT[x]	x=轴	再次返回的轮廓复位点
\$AC_SDIR[x]	x=主轴号 (主轴号)	旋转状态
\$AC_SMODE[x]	x=主轴号 (主轴号)	主轴工作方式
\$AC_STAT		通道状态

A.3 可用的系统变量列表

名称	索引	说明
\$AC_TIME		程序段开始时间 (秒) (包括内部生成的中间程序段时间)
\$AC_TIMES		程序段开始时间 (以秒为单位) (不包括内部生成的中间程序段时间)
\$AC_TIMEC		程序段开始时间 (以插补周期为单位) (包括内部生成的中间程序段周期)
\$AC_TIMESC		程序段开始时间 (以插补周期为单位) (不包括内部生成的中间程序段周期)
\$AC_TIMER[x]	x=定时器号	时间单元 (以秒为单位)
\$AC_VACTB		插补进给, 额定值
\$AC_VACTW		工件坐标系中的轨迹速度
\$AC_VC		同步指令的增加轨迹进给补偿
\$AN_AXCTAS[n]		轴容器实际地址 (旋转位置)
\$AN_AXCTSWA[CTn]	CTn=轴容器号	轴容器旋转激活
\$AN_ESR_TRIGGER		ESR 启动信号
\$AN_MARKER[x]	x=标记号	运动同步措施的标记变量
\$MC_DISPLAY_AXIS	位 16-31	识别机床轴
\$MC_DISPLAY_AXIS	位 0-15	几何轴 / 辅助轴标识
\$MC_MM_NUM_BASE_FRAMES		通道中基本框架数目
\$MN_EXTERN_LANGUAGE \$MN_EXTERN_CNC_SYSTEM	和	CNC 系统语言
\$MN_MAX_CUTTING_EDGE_NO		D 号码最大值
\$MN_MAX_CUTTING_EDGE_ PER_TOOL		每个刀具的最大刀沿数
\$MN_MAX_SUMCORR_ PERCUTTING_EDGE		每个刀沿的最大总和修正数
\$MN_MM_KIND_OF_SUMCORR		NCK 中总和修正特性
\$MN_MM_NUM_CC_MAGAZINE_ PARAM		刀具库的参数数目
\$MN_MM_NUM_CC_MAGLOC_ PARAM		刀具库位置的参数数目
\$MN_MM_NUM_CC_MON_ PARAM		刀沿监控用户数据的参数数目
\$MN_MM_NUM_GLOBAL_BASE_ FRAMES		与通道相关的基本框架数
\$MN_MM_NUM_SUMCORR		NCK 中总和修正的总数目
\$MN_MM_NUM_TOOL_ADAPTER		运行的刀具适配器的最大数组数
\$MN_MM_TOOL_MANAGEMENT_ MASK		设置 NCK 刀具管理
\$P_UBFR[x ,MI]	x=框架号	镜像可设置框架
\$P_UBFR[x,RT]	x=框架号	旋转可设置框架
\$P_UBFR[x,SC]	x=框架号	可设置框架校准系数
\$P_UBFR[x,SI]	x=框架号	框架的微小偏移
\$P_UBFR[x,TR]	x=框架号	转换可设置框架
\$P_CHBFRMASK		有效的通道专用基本框架
\$P_EG_BC[a]		EG 程序段切换标准

名称	索引	说明
\$P_NCBFRMASK		有效的与通道相关的基本框架
\$P_OFFN		偏移标准
\$P_PFRAME[x,TR] \$P_ACTFRAME 比如在编程 G54 之后, \$P_IFRAME包含由 G54 定义的平移、旋转、比例和镜像。	或者 或者	转换激活的框架
\$P_TOOL		激活的刀具刀沿号码
\$P_TOOLL[1]		激活的刀具长度 1
\$P_TOOLL[2]		激活的刀具长度 2
\$P_TOOLL[3]		激活的刀具长度 3
\$P_TOOLND[x]	x=刀具号	刀沿数目
\$P_TOOLNO		激活的刀具号码
\$P_TOOLR		激活的刀具半径
\$P_UIFR[x,y,MI]	x=框架号, y=轴	镜像可设置框架
\$P_UIFR[x,y,RT]	x=框架号, y=轴	旋转可设置框架
\$P_UIFR[x,y,SC]	x=框架号, y=轴	可设置框架校准系数
\$P_UIFR[x,y,SI]	x=框架号, y=轴	框架的微小偏移
\$P_UIFR[x,y,TR]	x=框架号, y=轴	转换可设置框架
\$P_UIFRNUM		激活的设置的框架索引
\$R[x]	x=参数号	R 参数
\$SC_PA_ACTIV_IMMED[x]	x=保护区号	返回参考点后立即激活
\$SC_PA_CENT_ABS[x,0]	x=保护区号	第 1 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ABS[x,1]	x=保护区号	第 2 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ABS[x,2]	x=保护区号	第 3 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ABS[x,3]	x=保护区号	第 4 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ABS[x,4]	x=保护区号	第 5 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ABS[x,5]	x=保护区号	第 6 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ABS[x,6]	x=号码保护区	第 7 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ABS[x,7]	x=保护区号	第 8 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ABS[x,8]	x=保护区号	第 9 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ABS[x,9]	x=保护区号	第 10 个轮廓元素的圆心横坐标
\$SC_PA_CENT_ORD[x,0]	x=保护区号	第 1 个轮廓元素的圆心纵坐标
\$SC_PA_CENT_ORD[x,1]	x=保护区号	第 2 个轮廓元素的圆心纵坐标
\$SC_PA_CENT_ORD[x,2]	x=保护区号	第 3 个轮廓元素的圆心纵坐标
\$SC_PA_CENT_ORD[x,3]	x=保护区号	第 4 个轮廓元素的圆心纵坐标
\$SC_PA_CENT_ORD[x,4]	x=保护区号	第 5 个轮廓元素的圆心纵坐标
\$SC_PA_CENT_ORD[x,5]	x=保护区号	第 6 个轮廓元素的圆心纵坐标
\$SC_PA_CENT_ORD[x,6]	x=保护区号	第 7 个轮廓元素的圆心纵坐标
\$SC_PA_CENT_ORD[x,7]	x=保护区号	第 8 个轮廓元素的圆心纵坐标
\$SC_PA_CENT_ORD[x,8]	x=保护区号	第 9 个轮廓元素的圆心纵坐标
\$SC_PA_CENT_ORD[x,9]	x=保护区号	第 10 个轮廓元素的圆心纵坐标
\$SC_PA_CONT_ABS[x,0]	x=保护区号	第 1 个轮廓元素的终点横坐标

A.3 可用的系统变量列表

名称	索引	说明
\$SC_PA_CONT_ABS[x,1]	x=保护区号	第 2 个轮廓元素的终点横坐标
\$SC_PA_CONT_ABS[x,2]	x=保护区号	第 3 个轮廓元素的终点横坐标
\$SC_PA_CONT_ABS[x,3]	x=保护区号	第 4 个轮廓元素的终点横坐标
\$SC_PA_CONT_ABS[x,4]	x=保护区号	第 5 个轮廓元素的终点横坐标
\$SC_PA_CONT_ABS[x,5]	x=保护区号	第 6 个轮廓元素的终点横坐标
\$SC_PA_CONT_ABS[x,6]	x=保护区号	第 7 个轮廓元素的终点横坐标
\$SC_PA_CONT_ABS[x,7]	x=保护区号	第 8 个轮廓元素的终点横坐标
\$SC_PA_CONT_ABS[x,8]	x=保护区号	第 9 个轮廓元素的终点横坐标
\$SC_PA_CONT_ABS[x,9]	x=保护区号	第 10 个轮廓元素的终点横坐标
\$SC_PA_CONT_NUM[x]	x=保护区号	有效的轮廓元素数目
\$SC_PA_CONT_ORD[x,0]	x=保护区号	第 1 个轮廓元素的终点纵坐标
\$SC_PA_CONT_ORD[x,1]	x=保护区号	第 2 个轮廓元素的终点纵坐标
\$SC_PA_CONT_ORD[x,2]	x=保护区号	第 3 个轮廓元素的终点纵坐标
\$SC_PA_CONT_ORD[x,3]	x=保护区号	第 4 个轮廓元素的终点纵坐标
\$SC_PA_CONT_ORD[x,4]	x=保护区号	第 5 个轮廓元素的终点纵坐标
\$SC_PA_CONT_ORD[x,5]	x=保护区号	第 6 个轮廓元素的终点纵坐标
\$SC_PA_CONT_ORD[x,6]	x=保护区号	第 7 个轮廓元素的终点纵坐标
\$SC_PA_CONT_ORD[x,7]	x=保护区号	第 8 个轮廓元素的终点纵坐标
\$SC_PA_CONT_ORD[x,8]	x=保护区号	第 9 个轮廓元素的终点纵坐标
\$SC_PA_CONT_ORD[x,9]	x=保护区号	第 10 个轮廓元素的终点纵坐标
\$SC_PA_CONT_TYP[x,0]	x=保护区号	第 1 个轮廓元素的轮廓类型
\$SC_PA_CONT_TYP[x,1]	x=保护区号	第 2 个轮廓元素的轮廓类型
\$SC_PA_CONT_TYP[x,2]	x=保护区号	第 3 个轮廓元素的轮廓类型
\$SC_PA_CONT_TYP[x,3]	x=保护区号	第 4 个轮廓元素的轮廓类型
\$SC_PA_CONT_TYP[x,4]	x=保护区号	第 5 个轮廓元素的轮廓类型
\$SC_PA_CONT_TYP[x,5]	x=保护区号	第 6 个轮廓元素的轮廓类型
\$SC_PA_CONT_TYP[x,6]	x=保护区号	第 7 个轮廓元素的轮廓类型
\$SC_PA_CONT_TYP[x,7]	x=保护区号	第 8 个轮廓元素的轮廓类型
\$SC_PA_CONT_TYP[x,8]	x=保护区号	第 9 个轮廓元素的轮廓类型
\$SC_PA_CONT_TYP[x,9]	x=保护区号	第 10 个轮廓元素的轮廓类型
\$SC_PA_LIM_3DIM[x]	x=保护区号	保护区极限 垂直坐标
\$SC_PA_MINUS_LIM[x]	x=保护区号	保护区下限, 垂直坐标
\$SC_PA_ORI[x]	x=保护区号	保护区层面分配
\$SC_PA_PLUS_LIM[x]	x=保护区号	保护区上限, 垂直坐标
\$SC_PA_T_W[x]	x=保护区号	工件或者刀具相关的保护区
\$SN_PA_ACTIV_IMMED[x]	x=保护区号	返回参考点后立即激活
\$SN_PA_CENT_ABS[x,0]	x=保护区号	第 1 个轮廓元素的圆心横坐标
\$SN_PA_CENT_ABS[x,1]	x=保护区号	第 2 个轮廓元素的圆心横坐标
\$SN_PA_CENT_ABS[x,2]	x=保护区号	第 3 个轮廓元素的圆心横坐标
\$SN_PA_CENT_ABS[x,3]	x=保护区号	第 4 个轮廓元素的圆心横坐标
\$SN_PA_CENT_ABS[x,4]	x=保护区号	第 5 个轮廓元素的圆心横坐标
\$SN_PA_CENT_ABS[x,5]	x=保护区号	第 6 个轮廓元素的圆心横坐标

名称	索引	说明
\$SN_PA_CENT_ABS[x,6]	x=保护区号	第 7 个轮廓元素的圆心横坐标
\$SN_PA_CENT_ABS[x,7]	x=保护区号	第 8 个轮廓元素的圆心横坐标
\$SN_PA_CENT_ABS[x,8]	x=保护区号	第 9 个轮廓元素的圆心横坐标
\$SN_PA_CENT_ABS[x,9]	x=保护区号	第 10 个轮廓元素的圆心横坐标
\$SN_PA_CENT_ORD[x,0]	x=保护区号	第 1 个轮廓元素的圆心纵坐标
\$SN_PA_CENT_ORD[x,1]	x=保护区号	第 2 个轮廓元素的圆心纵坐标
\$SN_PA_CENT_ORD [x,2]	x=保护区号	第 3 个轮廓元素的圆心纵坐标
\$SN_PA_CENT_ORD[x,3]	x=保护区号	第 4 个轮廓元素的圆心纵坐标
\$SN_PA_CENT_ORD[x,4]	x=保护区号	第 5 个轮廓元素的圆心纵坐标
\$SN_PA_CENT_ORD[x,5]	x=保护区号	第 6 个轮廓元素的圆心纵坐标
\$SN_PA_CENT_ORD[x,6]	x=保护区号	第 7 个轮廓元素的圆心纵坐标
\$SN_PA_CENT_ORD[x,7]	x=保护区号	第 8 个轮廓元素的圆心纵坐标
\$SN_PA_CENT_ORD[x,8]	x=保护区号	第 9 个轮廓元素的圆心纵坐标
\$SN_PA_CENT_ORD[x,9]	x=保护区号	第 10 个轮廓元素的圆心纵坐标
\$SN_PA_CONT_ABS[x,0]	x=保护区号	第 1 个轮廓元素的终点横坐标
\$SN_PA_CONT_ABS[x,1]	x=保护区号	第 2 个轮廓元素的终点横坐标
\$SN_PA_CONT_ABS[x,2]	x=保护区号	第 3 个轮廓元素的终点横坐标
\$SN_PA_CONT_ABS [x,3]	x=保护区号	第 4 个轮廓元素的终点横坐标
\$SN_PA_CONT_ABS[x,4]	x=保护区号	第 5 个轮廓元素的终点横坐标
\$SN_PA_CONT_ABS[x,5]	x=保护区号	第 6 个轮廓元素的终点横坐标
\$SN_PA_CONT_ABS[x,6]	x=保护区号	第 7 个轮廓元素的终点横坐标
\$SN_PA_CONT_ABS[x,7]	x=保护区号	第 8 个轮廓元素的终点横坐标
\$SN_PA_CONT_ABS[x,8]	x=保护区号	第 9 个轮廓元素的终点横坐标
\$SN_PA_CONT_ABS[x,9]	x=保护区号	第 10 个轮廓元素的终点横坐标
\$SN_PA_CONT_NUM[x]	x=保护区号	有效的轮廓元素数目
\$SN_PA_CONT_ORD[x,0]	x=保护区号	第 1 个轮廓元素的终点纵坐标
\$SN_PA_CONT_ORD[x,1]	x=保护区号	第 2 个轮廓元素的终点纵坐标
\$SN_PA_CONT_ORD[x,2]	x=保护区号	第 3 个轮廓元素的终点纵坐标
\$SN_PA_CONT_ORD[x,3]	x=保护区号	第 4 个轮廓元素的终点纵坐标
\$SN_PA_CONT_ORD[x,4]	x=保护区号	第 5 个轮廓元素的终点纵坐标
\$SN_PA_CONT_ORD[x,5]	x=保护区号	第 6 个轮廓元素的终点纵坐标
\$SN_PA_CONT_ORD[x,6]	x=保护区号	第 7 个轮廓元素的终点纵坐标
\$SN_PA_CONT_ORD[x,7]	x=保护区号	第 8 个轮廓元素的终点纵坐标
\$SN_PA_CONT_ORD[x,8]	x=保护区号	第 9 个轮廓元素的终点纵坐标
\$SN_PA_CONT_ORD[x,9]	x=保护区号	第 10 个轮廓元素的终点纵坐标
\$SN_PA_CONT_TYP[x,0]	x=保护区号	第 1 个轮廓元素的轮廓类型
\$SN_PA_CONT_TYP[x,1]	x=保护区号	第 2 个轮廓元素的轮廓类型
\$SN_PA_CONT_TYP[x,2]	x=保护区号	第 3 个轮廓元素的轮廓类型
\$SN_PA_CONT_TYP[x,3]	x=保护区号	第 4 个轮廓元素的轮廓类型
\$SN_PA_CONT_TYP[x,4]	x=保护区号	第 5 个轮廓元素的轮廓类型
\$SN_PA_CONT_TYP[x,5]	x=保护区号	第 6 个轮廓元素的轮廓类型
\$SN_PA_CONT_TYP[x,6]	x=保护区号	第 7 个轮廓元素的轮廓类型

A.3 可用的系统变量列表

名称	索引	说明
\$SN_PA_CONT_TYP[x,7]	x=保护区号	第 8 个轮廓元素的轮廓类型
\$SN_PA_CONT_TYP[x,8]	x=保护区号	第 9 个轮廓元素的轮廓类型
\$SN_PA_CONT_TYP[x,9]	x=保护区号	第 10 个轮廓元素的轮廓类型
\$SN_PA_LIM_3DIM[x]	x=保护区号	保护区极限 垂直坐标
\$SN_PA_MINUS_LIM[x]	x=保护区号	保护区下限, 垂直坐标
\$SN_PA_ORI[x]	x=保护区号	保护区层面分配
\$SN_PA_PLUS_LIM[x]	x=保护区号	保护区上限, 垂直坐标
\$SN_PA_T_W[x]	x=保护区号	工件或者刀具相关的保护区
\$TC_ADPT ...		适配器数据
\$TC_ADPTx \$TC_ADPTT	x=1 ... 3	每个适配器的参数数目
\$TC_DPCE		转换的刀沿修正值
\$TC_DPCx[y,z]	x=参数号 y=刀具号, z=刀沿号	用户定义的刀具刀沿参数
\$TC_DPx[y,z]	x=参数号 y=刀具号, z=刀沿号	刀沿修正值
\$TC_DPx[y,z]	x=参数号 y=刀具号, z=刀沿号	转换的刀沿修正值
\$TC_ECP ...		转换的位置相关的设立修正
\$TC_MAMP3		磨损相关策略
\$TC_MAP1		刀库类型
\$TC_MAP2		刀库命名符
\$TC_MAP3		刀库状态
\$TC_MAP4		到后一个刀库的刀库链 1
\$TC_MAP5		到前一个刀库的刀库链 2
\$TC_MAP6		刀库尺寸
\$TC_MAP9		激活的磨损组号码
\$TC_MAPCx[y]	x=参数号 y=刀库号	每个刀库的用户数据
\$TC_MOP1(x,y) ?\$TC_MOP15(x,y)	x=刀具号 y=刀沿号	每个刀沿的监控数据
\$TC_MOPCx[y,z]	x=参数号 y=T 号 z=刀沿	刀沿的监控用户数据
\$TC_MPPCx[y,z]	x= 参数号 y= 刀库号 z=刀库位置号	一个刀库的刀库位置用户数据
\$TC_MPPx	x=1,...7	每个刀库位置的参数数目
\$TC_SCP...		转换的位置相关的磨损修正
\$TC_SCP...		与位置相关的磨损修正
\$TC_SCPx	x=13,...21,...71	每个总补偿程序段的总补偿参数数目
\$TC_TP1		DUPLO 号码
\$TC_TP10		替换刀具的刀具查找类型
\$TC_TP11		HMI 的刀具信息

名称	索引	说明
\$TC_TP2		刀具命名符
\$TC_TP3		在半位置向左的大小
\$TC_TP4		在半位置向右的大小
\$TC_TP5		在半位置向上的大小
\$TC_TP6		在半位置向下的大小
\$TC_TP7		刀库位置类型
\$TC_TP8		刀具状态
\$TC_TP9		刀具监控类型
\$TC_TPCx[y]	x=参数号 y=刀具号	用户定义的刀具参数
\$TC_TPG1		主轴号码
\$TC_TPG2		级联规则
\$TC_TPG3		最小砂轮直径
\$TC_TPG4		最小砂轮宽度
\$TC_TPG5		当前砂轮宽度
\$TC_TPG6		砂轮最大速度
\$TC_TPG7		砂轮最大圆周速度 (SUG)
\$TC_TPG8		斜砂轮的倾斜角度
\$TC_TPG9		SUG 的修正参数
\$VA_COUP_OFFS[x]	x=轴	对导向轴 / 导向主轴的偏移, 实际值
\$VA_IS[x]	x=轴	轴的安全实际位置
\$VA_VACTM[x]	x=轴	机床坐标系 (MCS) 中实际值侧负载侧轴速度

A.4 PI 服务列表

编程

句法 **PI_SERVICE(服务, n 参数)**

服务 PI 服务的标识

n 参数 PI 服务的参数列表。
参数用逗号隔开。

服务	说明
参数	举例
_N_ASUP_	一种零件程序，位于 NCK 中（通过路径名称和程序名称表示），在规定的通道中分配一个中断号。该 PI 服务与程序指令‘SETINT’相同。
	<i>Par1</i> 中断号（0-8） <i>Par2</i> 优先级（0-8） <i>Par3</i> 快速提升（0, 1） <i>Par4</i> 块同步（0, 1） <i>Par5</i> 最大为 32 位的路径数据
	中断指令 5 用于当前通道中的程序 MPF_DIR/TEST_MPF。中断有优先级 3 并且在不快速提升的情况下在轮廓上执行 <i>PI_SERVICE("_N_ASUP_", 5, 3, 0, 0, "_N_MPF_DIR/_N_TEST_MPF")</i>
_N_CANCEL	分类为“取消报警”的所有报警可以用该命令确认。无法进行某些报警的个别确认。

	删除分类为“取消报警”的所有报警。 <i>PI_SERVICE("_N_CANCEL")</i>
_N_CRCEDN	依据刀沿号码规定设立一个刀具刀沿。如果在 PI 服务中的参数 T 号码下已规定了一个存在的刀具 T 号码，则设立该刀具的刀沿（在这种情况下参数 D 号码 - 设立的刀沿号码 - 的值范围为 1-9）。如果规定一个正的 T 号码作为参数并且不存在该规定 T 号码的刀具，则 PI 服务失败。 如果对于该 T 号码规定值为 0（绝对 D 号码的模型），则 D 号码的值范围扩展为 1-31999。新的刀沿通过规定的 D 号码产生。新的刀沿通过规定的 D 号码产生。 如果已存在规定的刀沿，则在两种情况下 PI 服务失败。
	<i>Par1</i> T 号码 <i>Par2</i> D 号码 T 号 == 0 ==> 1 - 31999 T 号 > 0 ==> 1 - 9
	对于在当前的 TO 范围中带有号码 17 的刀具，设立带有号码 3 的刀沿。 <i>PI_SERVICE("_N_CRCEDN", 17, 3)</i>

服务	说明
参数	举例
_N_CREACE	<p>产生一个用于特定刀具的刀具刀沿。自动产生下一个较高的空闲 D 号码。通过该 PI 服务影响下列激活的文件系统模块：</p> <p>刀具修正 TO：设立各个刀沿（带有内容零）</p> <p>监控数据 TS：（如果存在）设立各个刀沿（带有内容零）</p> <p>用户刀沿数据 TUE：（如果存在）设立各个刀沿（带有内容零）</p> <p>(SW 版本 NCK < 10.x)</p>
	<i>Par1</i> 刀具号码 1 至 31999
	<p>在以 T 范围 1 中的号码 55 设立刀具后，还设立用于该刀具的另 2 个刀沿。该刀具总共具有 3 个刀沿。</p> <p><i>PI_SERVICE("_N_CREATO",55)</i></p> <p><i>PI_SERVICE("_N_CREACE",55)</i></p> <p><i>PI_SERVICE("_N_CREACE",55)</i></p>
_N_CREATO	<p>以某个 T 号码设立刀具。</p> <p>通过该 PI 服务影响下列激活的文件系统模块：</p> <p>刀具目录 TV：刀具作为存在输入</p> <p>刀具修正 TO：设立第一个刀沿 D1（带有内容零）。</p> <p>用户刀沿数据 TUE：（如果存在）设立第一个刀沿 D1（带有内容零）</p> <p>用户刀具数据 TU：（如果存在）准备一个空的数组用于该刀具</p>
	<i>Par1</i> 刀具号码 1 至 31999
	<p>在以 T 范围 1 中的号码 55 设立刀具后，还设立用于该刀具的另 2 个刀沿。该刀具总共具有 3 个刀沿。</p> <p><i>PI_SERVICE("_N_CREATO",55)</i></p> <p><i>PI_SERVICE("_N_CREACE",55)</i></p> <p><i>PI_SERVICE("_N_CREACE",55)</i></p>
_N_DELECE	<p>删除一个刀具刀沿。</p> <p>如果在 PI 服务中的参数 T 号码下已规定了一个存在的刀具 T 号码，则删除针对该刀具的刀沿（在这种情况下参数 D 号码 - 要删除的刀沿号码 - 的值范围为 1 - 9）。如果规定一个正的 T 号码作为参数并且不存在该规定 T 号码的刀具，则 PI 服务失败。</p> <p>如果对于该 T 号码规定值为 0（绝对 D 号码的模型），则 D 号码的值范围扩展为 1 - 31999。新的刀沿通过规定的 D 号码产生。</p> <p>如果不存在规定的刀沿，则在两种情况下 PI 服务失败。</p>
	<p><i>Par1</i> 刀具 T 号码，该刀具应删除刀具刀沿。</p> <p>值表示为 0，针对该刀具不应该存在参考（绝对 D 号码）。</p> <p><i>Par2</i> 应删除的刀具刀沿号码。</p> <p>值范围：</p> <p>T 号 == 0 ⇒ 1 - 31999</p> <p>T 号 > 0 ⇒ 1 - 9</p>
	<p>对于在当前的 TO 范围中带有号码 17 的刀具，删除带有号码 3 的刀沿。</p> <p><i>PI_SERVICE("_N_DELECE",17,3)</i></p>
_N_DELETO	<p>删除数据模块中存储的刀具及其全部刀沿。</p> <p>在下列数据模块中（如果存在），同样删除刀具：TO, TU, TUE, TV, TG (类型 400), TD, TS.</p>
	<i>Par1</i> 刀具号码 1 至 31999

服务	说明
参数	举例
	在当前的 T 范围中删除带有 T 号码 21 的刀具。 <i>PI_SERVICE("_N_DELETE",21)</i>
	<i>Par1</i> 查找模式标识 1: 无计算搜索程序段 2: 带计算的查找 3: 在考虑主程序段情况下查找
	在当前的通道中通过计算启动查找。 要有效启动 PI 服务，必须之前满足数据段查找的数据结构（模块 SPRAF；通过变量服务在 HMI 内置上编译地址， /Channel/Search/.. ）。 <i>PI_SERVICE("_N_FINDBL",2)</i>
<i>_N_LOGIN_</i>	发送一个口令到 NCK，通过该口令设置当前的存取等级。
	<i>Par1</i> 口令（正好 8 个字符，少于 8 个字符必须用空格代替）
	传送一个口令到 NCK 并以此设置一个新的存取等级。 <i>PI_SERVICE("_N_LOGIN_","TESTWORD")</i>
<i>_N_LOGOUT</i>	复位当前的存取等级。

	复位当前的存取等级。 <i>PI_SERVICE("_N_LOGOUT")</i>
<i>_N_SETUFR</i>	通过通道特定的数据模块 FU 中系统或者用户变量'inShift', 'mirrorImgActive', 'rotation'和'scaleFact'可以为每个通道定义最多 8 个零点偏移。 为了可以激活用户定义的零点偏移，必须调用 PI 服务 <i>_N_SETUFR</i> 。

	激活一个用户框架。 <i>PI_SERVICE("_N_SETUFR")</i>

缩略语列表

B.1 缩略语

A	输出端
ASCII	American Standard Code for Information Interchange: 美国信息互换标准码
BAG	工作方式组
BTSS	操作面板接口
CAD	计算机辅助设计
CNC	Computerized Numerical Control (计算机数字控制)
CR	回车
DAU	数字模拟转换器
DB	PLC中数据块
DBB	PLC中数据块字节
DBW	PLC中数据块字
DBX	PLC中数据块位
DDE	Dynamic Data Exchange: 动态数据交换
DIN	德国工业标准
DIR	Directory: 目录
DPM	双端口内存
DOS	磁盘操作系统
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
DW	数据字
E	输入端
EG	扩展设备
ESR	扩展的停止和退回
FRAME	数据段(框架)
FIFO	先进-先出 : 数据如何保存在存储器以及如何重新调用的过程。
GP	主程序
GUD	Global User Data (全局用户数据)
HD	Hard Disk: 硬盘
HMI	Human Machine Interface: 控制系统操作区
HSA	主轴驱动
HW	硬件

缩略语列表

B.1 缩略语

开机调试	Startup
IKA	Interpolative Compensation: 可插补补偿
INC	Increment: 增量尺寸
INI	Initializing Data: 初始化数据
IPO	插补器
ISO	国际标准组织
JOG	Jogging: 手动工作方式
K1 .. K4	通道1到通道4
LED	Light Emitting Diode : 发光二极管
LF	线路馈电
K _v	回路放大系数
LUD	Local User Data: 局部用户数据
MB	兆字节
MCP	Machine Control Panel 机床控制面板 (→ MSTT)
MD	机床数据
MDA	Maual Data Automatic: 手动数据输入
MCS	机床坐标系
MLFB	机器可识别产品符
MPF	Main Program File: NC 零件程序 (主程序)
MPI	Multi Port Interface: 多端口接口
MSTT	(机床控制面板)
NC	数字控制: 数字控制装置
NCK	Numerical Control Kernel: 带有程序段处理, 运行范围等等的数字内核
NCU	Numerical Control Unit : NCK 硬件单元
NV	零点偏移
OEM	原装设备制造商
OP	Operation Panel : 操作面板
PCU	可编程控制单元
PCMCIA	Card International Association: 存储卡标准
PG	编程器
PLC	Programmable Logic Control:
REF	返回参考点功能
REPOS	再定位功能
ROV	Rapid Override: 输入端校正
RPA	R-Parameter Active: NCK 中的存储范围用于 R 参数编号的 R- NCK
SBL	Single Block: 单程序段
SD	设定数据
SDB	系统数据块
SEA	Setting Data Active: 设定数据符号 (文件类型)
SK	软键
SKP	Skip: 跳过程序段
SPF	Sub Program File: 子程序
SRAM	静态存储器 (缓存)

SUG	砂轮圆周速度
SW	软件
SYF	System Files: 系统文件
TEA	Testing Data Aktive: 机床数据标识
TO	Tool Offset: 刀具补偿
TOA	Tool Offset Active: 刀具补偿符号 (文件类型)
UFR	User Frame (用户框架)
VSA	进给驱动
WCS	工件坐标系
WZK	刀具补偿
WZW	换刀
ZOA	Zero Offset Active: 零点偏移数据符号 (文件类型)

词汇表

动作

所有在 →方式中设计的：→功能，→计算变量，→更改特性，...

用户变量

由用户在 →零件程序或者数据块中定义的变量。

数组

通过数组可以归类、保存同一数据类型的数据，从而可以通过索引存取数据。

属性

属于某个对象（→对话框或者→变量）的特定→特性。

操作树

多个相互连接的→对话框

块

用于→设计文件的装载单元

对话框

→操作界面的显示

- **和对话框相关的软键栏**

由一个新设计的对话框调用的软键栏。

- **和对话框无关的软键**

不由对话框调用的软键，即由第一个新对话框设计的登入软键和软键栏。

定义行

定义→变量和→软键的程序部分。

编辑器

ASCII 编辑器可以通过符号在文件中输入和编辑。

属性

对象的特征 (例如 : → 变量)

输入/输出栏

即 I/O 栏 : 用来输入或者输出变量值。

登入软键

→ 启动第一次新建立的 → 对话框的软键。

事件

所有触发处理→方法的事件 : 输入符号 , 按下 → 软键 , ...

焦点

屏幕上的突出显示处 , 表示当前 → 单元 , 例如 : 光标所在处。

功能

根据 → 参数在 → 方法中编程的流程。

辅助变量

没有 → 特性并因此不显示在 → 对话框中内部计算变量。

热键

OP 010、OP 010C 和带有热键块的 SINUMERIK 键盘上的 6 个按键 , 按下此键直接选择一个操作区。也可以选择 2 个其它的按键作为热键操作。

HSx

水平 → 软键 x

编译器

编译器自动将 → 设计文件中定义的代码转换成 → 对话框并控制其使用。

机床数据

由西门子 / 机床制造商 / 最终用户存放在系统中的 SINUMERIK 系统特性的设置。它分为：

\$MN_... 一般 NC 机床数据

\$MC_... 通道专用机床数据

\$MA_... 轴专用机床数据

\$MM_... 操作面板机床数据

此外存在的 → 设定数据和驱动机床数据。

方法

如果出现相应的 → 事件，则执行编程的步骤。

HMI 高级/ HMI 内置 si

控制系统的操作界面

NC

数字控制: 根据 → 零件程序控制轴运动过程的 SINUMERIK 系统组件。

NC 代码

建立 SINUMERIK → 零件程序允许的语言单元

NC 功能

→ PI 服务

实用注释

NC 代码生成时自动产生的注释。

参数

参数是编程语句的可改变部分，在 → 设计文件中可用另一个字 / 符号替换。

PI 服务

在 → NC 上执行固定操作的 → 功能。

PI 服务可以由 → PLC 和 → HMI 高级/HMI 内置 si 调用。

PLC

Programmable Logic Control: 执行大部分 SINUMERIK 系统中逻辑链接的存储器可编程的控制器。

PLC 按键

PLC 按键通过 HMI 软件的 PLC 接口提供，如同热键。在 HMI 中由它触发的功能都可以设计。

该按键可以是机床操作面板上的按键或者是 PLC 的信号连接端。因此，它们也被称作“虚拟按键”。

编程支持

提供对话框以支持 → 建立零件程序，组件版本较高

设计文件

包含定义和指令，并确定 → 对话框的外观和 → 功能的文件。

寄存器

用于 → 对话框间数据交换的存储器。

反编译

由 → 编程支持 → 对话框的输入栏可以在 → 零件程序中产生 NC 代码段。反编译描述相反的过程。产生一个所选择的 NC 代码段的输入栏由 NC 代码重新恢复并显示在原来的对话框中。

设定数据

按照 NC 控制系统软件定义的方法表明机床特性的数据。与 → 机床数据相反，可由操作人员立即更改。

ShopMill

优化的 SINUMERIK 使用和操作界面，用于 2½D 铣削。

ShopTurn

优化的 SINUMERIK 使用和操作界面，用于车削。

模拟

模仿 → 零件程序过程，但没有实际的机床轴运动。

软键

触发所属屏幕上显示的功能的操作面板按键。

软键标签

→ 软键在屏幕上的文本/图形。

软键栏

所有水平或者所有垂直的 → 软键

栏索引

数组栏索引

标准应用程序

存在的 → 标准操作界面。

系统变量

提供零件程序存取和 → HMI 存取时的 NC 状态的 NC 变量。

零件程序

规定轴的运动过程以及各种特殊动作的、以 NC 语言编制的程序。

转换栏

→ 输入 / 输出栏中的值列表；通过转换栏检查：某栏中的输入必须与列表值中的一个相符合。

变量

通过 → 特性分配显示在 → 对话框中并可以记录到输入数据和计算结果中的存储器空间。

VSx

垂直 → 软键 x

行索引

数组行号码

存取等级

权限分级，即根据用户的不同权限而使用 → 操作界面上的功能。

索引

C

Custom

 捆绑, 155

 热键, 156

 特点, 155

CUSTOM.COM, 158

D

DLL 文件, 104

G

Grid → 表格栅格, 52

H

HMI 字节, 141

K

KeyConfiguration, 145

Keys.ini, 134

M

MD 9016

 MM_SWITCH_TO_AREA, 156

N

NC 变量

 写入, 91

 读取, 90

P

PI 服务, 73

PLC 变量

 写入, 91

 读取, 90

PLC 字节, 141

PLC 接口, 139

PLC 程序, 140

PLC 软键, 139

PLC接口, 140

T

Task (任务), 137

三

三角函数功能, 109

中

中间缓冲, 140

主

主对话框, 87

二

二进制显示, 33

位

位置

 短文本, 30, 35

 输入/输出栏, 30, 35

保

保护等级, 60

写

写入模式, 30

- 前**
前景颜色, 30
- 功**
功能
AP (主动程序), 73
CALL (子程序调用), 75
CP (复制程序), 77
CVAR (检查变量), 76
DLGL (对话框行), 78
DLL 文件, 73
DP (删除程序), 78
EP (存在程序), 80
EVAL (评估), 79
EXE (执行), 80
EXIT, 81
EXITLS (退出装载软键), 83
FCT, 104
GC (生成代码), 83
INSTR (字符串), 100
LA (装载数组), 85
LB (装载块), 87
LEFT (字符串), 101
LEN (字符串), 100
LM (装载屏幕窗口), 87
LS (装载软键), 89
MIDS (字符串), 102
MRNP (多次读取 NC PLC), 91
PI 服务, 103
PI_START, 103
PP (被动程序), 90
REFRESH (刷新), 93
REPLACE (字符串), 102
RETURN (返回), 96
RIGHT (字符串), 101
RNP (读取 NC PLC 变量), 90
WNP (写入 NC PLC 变量), 91
反编译 NC 代码, 97
向前查找(SF), 98
向后查找(SB), 98
概述, 72
选择程序 (SP), 99
- 动**
动作, 137
- 十**
十六进制显示, 33
- 单**
单位文本, 29
- 取**
取值范围, 136
- 变**
变量
CURPOS, 42
CURVER, 43
ENTRY, 43
ERR, 44
FILE_ERR, 45
FOC, 46
S_CHAN, 47
传输, 82
更改属性, 27
检查, 76
计算, 37
变量值, 26
变量状态, 27
变量类型, 28
INTEGER, 31
VARIANT, 32
- 命**
命令通道 → 对话框, 116
- 图**
图形文本, 29
- 多**
多重操作, 134
- 子**
子对话框, 87
子程序, 73
中断, 96

变量, 74
块标记, 74
调用, 75

字

字符串链, 41

定

定义软键栏, 59

寄

寄存器
 交换数据, 94
 值, 94
 状态, 95

对

对话框
 多列, 22
 定义, 13
 属性, 18
 序号, 140
 标题, 157
 激活, 116
 说明块, 15
对话框切换模式, 88
对话框单元, 21

属

属性, 29

帮

帮助, 30
帮助(仅 HMI 高级), 35
帮助画面, 30

常

常量, 110

指

指数显示, 33

按

按键, 129
 ETC, 137, 138
 M (Machine), 130, 138
 MENU SELECT, 130, 138
 回调, 61
按键区, 129

操

操作区
 Custom, 155
 切换, 157
 图形, 142
操作树, 9

数

数组
 元素, 48
 存取模式, 48
 定义, 47
 栏索引, 48
 比较模式, 48
 状态, 51
 行索引, 48

文

文件
 传输, 73, 90
 删除, 78
 复制, 77
文本, 29

方

方法
 CHANGE, 65
 LOAD, 68
 LOAD GRID, 67
 OUTPUT, 69
 PRESS, 70
 UNLOAD, 68
 概述, 65

条

条件, 110

极

极限值, 28

比

比较运算符, 110

热

热键事件, 134, 136

状

状态, 137

生

生成 NC 代码, 83

用

用户变量, 30

画

画面

撤销选择, 142

显示为短文本, 40

选择, 141

登

登入软键, 63

短

短文本, 29

系

系统变量, 30, 37

聚

聚焦控制, 55

背

背景颜色, 30

虚

虚拟键, 136

表

表格栅格

定义, 52

定义列, 54

编程, 53

设

设计文件, 9, 10

转

转换栏, 33

软

软键

分配属性, 59

属性, 61

辅

辅助变量, 36

输

输入模式, 29

运

运算符

位, 111

数学, 109

返

返回, 10

长

长文本, 29

预

预设值, 28

颜

颜色, 30, 173

SINUMERIK

SINUMERIK 840D sl/840Di sl/810D 在线帮助 (HE1)

调试手册

适用于：控制系统

SINUMERIK 840D sl/840DE sl
SINUMERIK 840Di sl/840DiE sl
SINUMERIK 840D/840DE
SINUMERIK 840Di/840DiE
SINUMERIK 810D/810DE

软件
HMI 高级

软件版本
7.3

11/2006
6FC5397-0DP10-0RA0

安全技术提示

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

带有警告三角，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。

小心

不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。

注意

表示如果不注意相应的提示，可能会出现不希望的结果或状态。

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

仅允许安装和驱动与本文件相关的附属设备或系统。设备或系统的调试和运行仅允许由**合格的专业人员**进行。本文件安全技术提示中的合格专业人员是指根据安全技术标准具有从事进行设备、系统和电路的运行，接地和标识资格的人员。

按规定使用

请注意下列说明：



警告

设备仅允许用在目录和技术说明中规定的使用情况下，并且仅允许使用西门子股份有限公司推荐的或指定的其他制造商生产的设备和部件。设备的正常和安全运行必须依赖于恰当的运输，合适的存储、安放和安装以及小心的操作和维修。

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。标签中的其他符号可能是一些其他商标，这是出于保护所有者权利的目地由第三方使用而特别标示的。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

目录

1	编程帮助	5
1.1	概述	5
1.2	简短帮助 "ASCII"	8
1.2.1	帮助功能的扩展/替代	8
1.2.2	文件 "IF.INI" 中的输入项	8
1.2.3	文本文件扩展	11
1.2.4	标题的扩展	13
1.2.5	指令的扩展	14
1.2.6	详细帮助 "PDF" 的参数设置	15
1.2.7	带报警文本变量的扩展	18
1.3	详细帮助 "PDF"	18
1.3.1	功能	18
1.3.2	扩展编程帮助	19
1.4	ISO 模式帮助	21
1.4.1	功能	21
1.4.2	扩展 ISO 模式帮助	22
2	报警/机床数据帮助	25
2.1	概述	25
2.2	报警帮助	28
2.2.1	功能	28
2.2.2	扩展报警帮助	29
2.2.3	用于帮助的用户注释	30
2.2.4	把报警帮助 "Win 帮助" 转换为 "PDF 帮助"	30
2.3	机床数据帮助	31
2.3.1	功能	31
2.3.2	扩展机床数据帮助	32
	索引	35

表格

表格 1-1	标题 ID	13
表格 1-2	预先规定的标题 ID	13

编程帮助

1.1 概述

编程帮助

在 PCU50.3 控制系统中可以使用以下扩展的编程帮助：

- 简短帮助 "ASCII"
- 详细帮助 "PDF"
 - 出自简短帮助 "ASCII" 的附加帮助 (PDF 文件)。
 - 循环参数掩码输入栏的附加帮助
 - ISO 模式帮助

下列章节中描述了帮助的功能和扩展性。

软件

- HMI 高级
- Adobe Reader 4.x (可通过 Internet / 专业贸易免费获取)
通常软件 "Adobe Acrobat Reader 4.x" 在控制系统 SINUMERIK 840D sl 中已安装。
- Adobe Acrobat 4.x
需要软件 Adobe Acrobat Distiller 4.x 来把文件转换成 PDF 格式并编辑 PDF 文档。

辅助文件存放

编程帮助：F:\DH\CST.DIR\HLP.DIR (可能包含 HLP, PDF 和 TXT 文件)

报警帮助/机床数据帮助：F:\MMC2\HLP (可能包含 HLP, PDF 和 TXT 文件)

BTSS 帮助：F:\HMI_ADV\HLP (可能包含 HLP, PDF 和 TXT 文件)

PDF 帮助原理

PDF 帮助需要帮助文件存储中的下列部分：

- 调用文件，例如：PG_GR.TXT，带数据说明：
 <PDF 文件名> <ID> <查找概念>每次调用
- 所属的 PDF 文件

按照下列步骤进行帮助调用：

- 启动 Adobe Acrobat Reader（包含在控制系统中）
- 跳转到页数（通过索引在文件中确定）
- 查找概念（Adobe 软件的查找功能）

创建时的步骤

- 在帮助文件中确定（例如：通过 DTP 系统的标准索引功能）并生成跳转
- 保存跳跃文件作为文本文件
- 为编程指令分配 ID 编号
- 生成 PDF 文件

结果：

- 带有数据说明的调用文件（文本文件）：
 <PDF 文件名> <ID> <查找概念>每次调用
- 一个或者多个所属的 PDF 文件

注意

对一个主题有多个跳跃时，必须在另一行中写出页码。

举例：

```
840D_PG.PDF 44 155
840D_PG.PDF 45 180
840D_PG.PDF 46 199
```

提示：

对于一个更大的索引容量，可以通过微软的 Excel 自动创建一个调用文件：

- 在 Excel 中插入“索引文件” (*.TXT)
- 插入列
- 在第一行中输入 PDF 文件的名称
- 执行菜单“编辑” - “填充” - “下一步”
- 复制数据并
- 保存为 "TXT" 文件

此外还有以下句法：

<PDF 文件名> <ID> <查找概念>

840D_PG.PDF 44 155

在文件中测试跳跃

为了测试是否能找到所有的查找词，按下列步骤进行：

- 启动 Adobe Acrobat
- 选择菜单“编辑”->“查找” (CTRL + F)
- 输入查找词

注意

如果没有找到查找词，可能有以下原因：

- 输入错误
- 生成 PDF 时，文本转换不正确，所以搜索器无法找到。

*分析：*用 Adobe Acrobat 选中要查找的词，复制并插入 Word 文档。

解决方法：

- 重新生成 PDF 或者
 - 用 Adobe 打印机驱动器 Adobe PS 5.1.1 生成后脚本文件，然后用 Adobe Distiller 进行转换。
-

查找顺序

对于帮助遵循下列查找顺序：

- F:\USER\HLP
- F:\OEM\HLP
- F:\ADD_ON\HLP
- F:\MMC2\HLP
- F:\HMI_ADV\HLP
- F:\DH\CUS.DIR\HLP.DIR (用户目录)
- F:\DH\CMA.DIR\HLP.DIR (制造商目录)
- F:\DH\CST.DIR\HLP.DIR (西门子目录)
- F:\DH\COM.DIR\HLP.DIR

1.2 简短帮助 "ASCII"

为了在零件程序编辑时提供编程支持，可以在编辑器中通过“信息键”使用以下帮助功能：

- 简短帮助 "ASCII"
 - 指令的说明文本，上下文相关文本参照零件程序中的光标位置（例如：G9“准停 - 降低速度”）
 - 类别概述（例如：“位移条件”，“位移指令”，“轨迹特性”等等），对应不同的指令
 - 带有说明文本的指令概览
 - 在帮助窗口中通过标题或文本查找
 - 把所选的指令接收到编辑器
- 详细帮助 "PDF"
 - 从简短帮助 "ASCII" 中调用 附加帮助（PDF 文件），例如：跳转至编程手册基础部分。
 - 调用循环窗口输入栏的附加帮助（PDF 文件）

1.2.1 帮助功能的扩展/替代

机床制造商/最终用户可以把附加的 M 功能、H 功能、循环或者宏集成到帮助系统中。

可以扩展帮助功能访问的相应文本文件（参见章节“文本文件的扩展”）。

生成的 ASCII 文本文件的路径必须在初始化文件 "If.ini" 中或在“设置”（用户文件名）中进行参数设定或补充。

1.2.2 文件 "IF.INI" 中的输入项

概述

在文件 "IF.INI" 中，下列部分用于帮助功能的参数设置：

- TECHNOLOGY
- TEXTFILES
- CONTEXT
- TEXTSEARCH

TECHNOLOGY (工艺) 部分

在 [TECHNOLOGY] 部分，通过机床工艺属性的预设值，指令文本的显示限于与机床相关的输入项。然后只显示所属工艺的输入项。

此时要考虑工艺标识：

"m" = 铣削

"t" = 车削

"g" = 磨削

"s" = 步冲

"p" = 冲裁

"a" = 显示全部工艺

TEXTFILES (文本文件) 部分

在 [TEXTFILES] 部分设置路径参数 (路径\名称)，以简短帮助的文本文件为基础的帮助功能。

帮助功能访问下列文本文件：

- 西门子 "DH\CST.DIR\IFS_GR.COM"
- 机床制造商 "DH\CMA.DIR\IFM_GR.COM" (任意文件名)
- 最终用户 "DH\CUS.DIR\USER_GR.COM" (任意文件名)
(通过 “设置”输入)

注意

必须存在用于西门子文本文件 "DH\CST.DIR\IFS_GR.COM" 的路径，否则会出现相应的故障信息。

文本路径分配时，对于文本文件的名称有不同的参数设置方法。

下表包括了参数设置方法和其帮助功能解释。

用于文本文件的参数	解释
文件名.com	读出与语言无关的文本文件
文件名_XX.com (XX=语种缩写，例如：“GR”代表德语，语种缩写参见章节“报警帮助”中的“其他语言”)	读出与语言相关的文本文件

举例： ENDUSER=CUS.DIR\IFM_.COM

首先在路径 "C:\DH\CUS.DIR\" 下打开和语言不相关的文本文件 "IFM.COM"，如果该文件不存在，则打开各个和语言相关的文本文件 (例如：“IFM_GR.COM”)。

TXT 文件 "IFM_GR.COM" 的摘录:

```
...  
a2//G00/快进线性插补 (m) 简短帮助  
a2//G01/进给线性插补 (m)  
...  
a/20/"PGA.PDF",2,"="/= /赋值运算符 带详细帮助  
...  
a/3/"PG.PDF",17,"ANG"/ANG/轮廓牵引角 (s) 带详细帮助循环  
...  
m/14/"PGZ.PDF",302,"CYCLE77"/CYCLE77/铣削循环 圆弧轴颈铣削带详细帮助循环  
...
```

CONTEXT (上下文) 部分

在 [CONTEXT] 部分，可以设置一个扩展的上下文相关性。用 "1" 激活，用 "0" 取消激活 (通过"设置"输入)。

如果激活了扩展的上下文相关性，而且光标靠左 (在指令文本旁或上面)，调用帮助时，所有指令以相同的起始字句显示。

如果程序员选择了指令，例如："G1"，则显示所有以 "G1" 开头的指令 (例如："G1", "G17", "G18", "G19")。

如果输入 "0" 没有激活扩展的上下文相关性，则 (如果有的话) 只显示当前指令。

TEXTSEARCH (文本搜索) 部分

在 [TEXTSEARCH] 部分，可以对文本查找进行详细说明。对此有以下方案：

1 = 仅指令文本

2 = 仅说明文本

3 = 指令文本和说明文本

(通过"查找"输入)。

文件 IF.INI

```
[TECHNOLOGY]
"m" = 铣削
"t" = 车削
"g" = 磨削
"s" = 步冲
"p" = 冲裁
"a" = 显示全部工艺

[TEXTFILES]
西门子 = DH\CST.DIR\IFS_GR.COM
机床制造商 = DH\CMA.DIR\IFM_GR.COM
机床制造商 = DH\CMA.DIR\IFI_GR.COM
最终用户 = DH\CUS.DIR\USER_GR.COM

[CONTEXT]
= 0: 上下文相关性未激活
= 1: 上下文相关性激活

[TEXTSEARCH]
= 1: 仅按指令文本查找
= 2: 仅按说明文本查找
= 3: 按照指令文本和说明文本查找
```

1.2.3 文本文件扩展

类别

在文本文件中规定了下面 2 个类别：

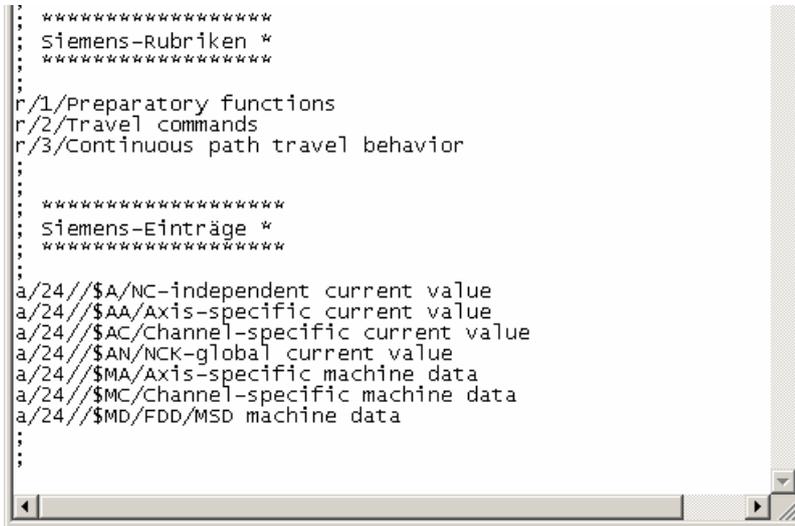
- 标题定义
- 指令定义

按照预先确定的样本，在帮助的文本文件 *.com 中添加下列功能：

- 标题作为一个**指令组**的标识
- 指令带**说明文本**
- 指令带扩展帮助调用
 - 编辑器帮助 "PDF"或者
 - 对 (提供指令参数的) **循环掩码输入栏**的循环帮助
- 带有**报警文本变量**的指令作为说明文本

这里在每个文本文件中有一行可供使用。

下图显示了一个例子：



```
*****
: Siemens-Rubriken *
*****
:
:
r/1/Preparatory functions
r/2/Travel commands
r/3/Continuous path travel behavior
:
:
*****
: Siemens-Einträge *
*****
:
a/24//$A/NC-independent current value
a/24//$AA/Axis-specific current value
a/24//$AC/Channel-specific current value
a/24//$AN/NCK-global current value
a/24//$MA/Axis-specific machine data
a/24//$MC/Channel-specific machine data
a/24//$MD/FDD/MSD machine data
:
:
:
```

图 1-1 西门子文本文件

标题定义包含下列参数：

标题定义

工艺名称 "r"/标题 ID/标题名称

指令定义包含下列参数：

指令定义

工艺/标题 ID/扩展帮助/指令文本/
说明文本

注意

关于单独的文本文件，首先显示西门子标题，然后机床制造商标题，最后是最终用户标题。

只有在文本文件中**首先声明标题**随后指令时，才出现上述顺序。

为了正确分类标准指令、机床制造商指令和最终用户指令，文本文件中的指令必须按**字母顺序**保存。

1.2.4 标题的扩展

标题的扩展

在标题下隐藏了一个指令组，通过它来显示指令。

标题定义行包括下面 3 个参数：

标题定义

工艺名称 "r"/标题 ID/标题名称

例如：在文本文件中可能有下列行：

r/2/位移指令

r：在标题声明处预先规定。

2：通过标题 ID 分配指令

位移指令：标题名称。

符号 "/" 用来分隔参数，而且只允许在最后一个参数中用作文本符号。

标题的长度受到显示的限制，例如：标准文本长度不允许超过 8pt 40 个字符。

下面的标题 ID 是预留的：

表格 1-1 标题 ID

标题 ID	用户
0..49	西门子
50..99	机床制造商
大于 100	最终用户

正常情况下，在西门子文本文件中有下列标题 ID：

表格 1-2 预先规定的标题 ID

标题 ID	标题名称	标题 ID	标题名称
1	位移条件	13	程序操作
2	位移指令	14	标准循环
3	轨迹运行特性	15	测量循环
4	进给率	16	测量
5	主轴	17	同步动作
6	轴操作	18	中断操作
7	M 功能	19	字符串/数组运算
8	H 功能	20	算术函数/运算
9	偏移	21	定义
10	转换	22	规范语言段
11	刀具管理/ 补偿	23	参数
12	工艺功能	24	系统变量

1.2.5 指令的扩展

指令的扩展

指令包含说明文本，例如："G0/快进线性插补"。通过标题 ID 把指令分配给标题。

指令定义参数

指令定义包含下列参数：

工艺/标题 ID/扩展帮助/指令文本/说明文本

编辑器帮助举例：

a/20/"PGA.PDF",2,"="/=/*赋值运算符*

a: *工艺铣削*

20: *标题 ID 标准循环*

"PGA.PDF",2,"=": *扩展帮助用于指令 "="*

=: *指令文本*

赋值运算符：说明文本

在某个用户文本文件（例如：IFM_GR.COM）中输入上述行之后，对于指令 "=" 调用了帮助文件 "pga.pdf"，并显示相应的帮助段落。

循环帮助举例：

m/14/"PGZ.PDF",302,"CYCLE83"/CYCLE83/*铣削循环 深钻孔*

m: *工艺铣削*

14: *标题 ID 标准循环*

"PGZ.PDF",302,"RTP": *扩展帮助用于循环参数设置掩码的参数 "RTP"*

CYCLE83: *指令文本 CYCLE83*

铣削循环 深钻孔：说明文本

在某个用户文本文件中输入上述行后，在帮助系统中支持用户循环 „CYCLExx“。当在初始化文件 "IF.INI" 中

- 输入了与工艺有关的标识 "a"
- 或者铣床专用标识 "m" 时，显示用户循环。

通过标题 ID "14"，把该用户循环归于标准循环，它也可以在标题查找中通过输入相应的项被找到。

该说明文本位于最后一个参数中，显示在帮助系统相应的列中，而且会给用户关于该循环功能的首个提示。

符号 "/" 用来分隔参数，而且只允许在最后一个参数中用作文本符号。

工艺

对于工艺属性规定了下列符号：

"m" = 铣削

"t" = 车削

"g" = 磨削

"s" = 步冲

"p" = 冲裁

"a" = 显示全部工艺

注意

为了能够把输入项归于多个工艺以及不同的标题，必须通过“逗号”进行分隔。

标题 ID

用标题 ID 可以把相应的标题指令归于标题定义，并显示在标题下。

捆绑附加指令

如果要在帮助中捆绑带有简短说明的附加指令，必须按照目录 Dh\CMA.dir\ 中的上述句法在文件 lfM_gr.com (任意文件名) 中捆绑。如果要对该指令调用一个附加帮助 (详细帮助 "pdf")，必须按相应段落对参数 *扩展帮助* 进行补充。

1.2.6 详细帮助 "PDF" 的参数设置

调用扩展帮助

通过指令定义“扩展帮助”参数中的特定标识，可以在简短帮助外附加调用一个详细帮助 "PDF" 作为在线帮助，它可能包括例如：关于所选指令的详细信息。为此有下面可能的调用：

- **编辑器帮助：**

从程序编辑器中一个概念 (指令) 的简短帮助 "ASCII" 直接跳到帮助 (PDF 文件)

- **循环帮助：**

从循环窗口的输入栏跳到帮助 (PDF 文件)

指令定义的参数

工艺/标题 ID/扩展帮助/指令文本/说明文本

扩展的帮助

在参数“扩展帮助”中可能有下列项：

- 编辑器帮助

/"PDF 文件名" , 帮助 ID , "查找概念" /

- 循环帮助

/"PDF 文件名" , 帮助 ID , "查找概念" (文件名 , 帮助画面) /

在参数“扩展帮助”中设置了两个逗号，即使没有输入项。

举例：

声明行需要下列参数，例如：

m,t/14/"PDF 文件名" , 帮助 ID , "查找概念" (文件名 , 帮助画面) /CYCLE82/钻孔循环, 钻孔, 镗平面

参数“扩展帮助”的输入项有下列意义：

PDF 文件名

= 从该文件中显示说明文本

帮助 ID

= 对于待显示帮助的概念分类是明确的

查找概念

= 在帮助文本中显示的 PDF 文件中的查找概念。

(文件名 , 帮助画面)

= 帮助画面所在的文件的名称，以及要显示的帮助画面名称 (-> 仅与循环帮助相关)

CYCLE82

= 指令文本

钻孔循环, 钻孔, 镗平面

= 说明文本

举例调用：

m,t/14/"PGZ.PDF" , 1 , "RTP" (钻孔 , CYC82) /CYCLE82/钻孔循环, 钻削, 镗平面

注意

如果仅存取 PDF 文件，为此该参数只位于“扩展帮助”部分。

例如： m,t/14/"PGZ.PDF",1,"RTP"/CYCLE82/钻孔循环, 钻削, 镗平面

如果仅显示帮助画面，那么为此该参数只位于“扩展帮助”部分。

例如： m,t/14/,,(钻孔,CYC82)/CYCLE82/钻孔循环, 钻削, 镗平面

指令文本

指令文本可能是，例如："G17"。该文本在“接收至编辑器”时被录入刚刚编辑过的零件程序中。

指令文本的长度可能约 12 个字符长。

说明文本

说明文本是帮助文本，对指令进行更详细的描述。

说明文本的长度受到显示的限制，例如：标准文本长度不允许超过 8pt 40 个字符。

多次跳转

对于每个帮助元素，几个帮助参考可以作为循环保存（即：在最后一帮助后又显示第一条）。

例如：m,t/14/"Hlp.pdf",1,"帮助 1",,2,"帮助 2","帮助 3.pdf",3,,/
CYCLE82/钻孔循环, 钻削, 镗平面

即：

- 帮助文本循环中的输入项
文件：HLP.PDF
索引：1
文本：帮助 1
- 帮助文本循环中的输入项
文件：HLP.PDF
索引：2
文本：帮助 2
- 帮助文本循环中的输入项
文件：HELP3.PDF
索引：3
文本：帮助 2

为了避免重复，如同举例时看到的，一个程序段的组成部分被前面的程序段记住。只接收整个程序段。

注意

即使没有输入项，在参数“扩展帮助”中也设置了两个逗号。

1.2.7 带报警文本变量的扩展

功能

关于说明文本或标题名称部分的参数设置的特殊性在于，替代说明文本（“报警变量”），保存在该部分中。

这些变量以 "\$" 开始。

例如：`m/14//CYCLExx/$888888`

替代说明文本直接输入帮助文本文件中，例如：保存在报警文本变量 "\$888888" 中。

说明文本自身的声明以这种方式保存到用户专用的报警文本文件中。

因此得到一个与语言无关的文本文件。

1.3 详细帮助 "PDF"

1.3.1 功能

为了在零件程序编辑时提供编程支持，可以在编辑器中通过“信息键”使用以下帮助功能：

- 简短帮助 "ASCII"
 - 指令的说明文本，上下文相关文本参照零件程序中的光标位置（例如：G9“准停 - 降低速度”）
 - 类别概述（例如：“位移条件”，“位移指令”，“轨迹特性”等等），对应不同的指令
 - 带有说明文本的指令概览
 - 在帮助窗口中通过标题或文本查找
 - 把所选的指令接收到编辑器
- 详细帮助 "PDF"
 - 从简短帮助 "ASCII" 中调用附加帮助（PDF 文件），例如：跳转至编程手册基础部分。
 - 调用循环窗口输入栏的附加帮助（PDF 文件）

1.3.2 扩展编程帮助

通过指令定义“扩展帮助”参数中的特定标识，可以在简短帮助外附加捆绑一个详细帮助（例如：关于所选指令的详细信息）。

为此有下面可能的调用：

- **编辑器帮助：**

从程序编辑器中一个概念（指令）的简短帮助 "ASCII" 直接跳到帮助（PDF 文件）

前提条件：

在简短指令中显示的附加指令的列表，已经按照章节“文本文件的扩展”创建（例如：IFM_GR.COM）并保存到了控制系统。因此，通过信息键在简短帮助之后也可以调用详细帮助。请注意，简短帮助的指令列表与详细帮助的指令列表一致（主要语句索引）。

- **循环帮助：**

从循环窗口的输入栏跳到帮助（PDF 文件）

指令定义的参数

工艺/标题 ID/扩展帮助/指令文本/
说明文本

“扩展的帮助”：

在参数“扩展帮助”中可能有下列项：

- **编辑器帮助**

/“PDF 文件名”，帮助 ID，“查找概念”/

举例：a/20/"PGA.PDF",2,"="/= 赋值运算符

- **循环帮助**

/“PDF 文件名”，帮助 ID，“查找概念”（文件名，帮助画面）/

举例：a/3/"PG.PDF",17,"ANG"/ANG/轮廓牵引角 (s)

必要的文件

- IFM_GR.COM:带有程序指令和简短说明的文本文件
- *.PDF: 帮助文件，例如：840D_PG_GR.PDF
- *.txt: 跳转文件，例如：840D_PG_GR.TXT

1) 确定文档中的跳转标识

在控制系统中显示为 PDF 的文件（例如：WinWord）中，必须用一个索引项标出跳转标识，并且生成单列索引目录（页码右对齐）。

对于一个需要多个跳转的指令，也要相应分配多个索引标识。

2) 生成文本文件

复制上面生成的单列索引 (页码右对齐) 并把它作为 *.TXT 文本文件保存。

举例：

MEAW 155

<查找词> <页>

文本文件用于 PDF 调用时需要下列句法：

<PDF 文件名> <ID> <页 >

在每行前插入 PDF 文件的名称 (无语言标识) 。

举例：

840D_PG.PDF MEAW 155

<PDF 名称> <ID> <页 >

3) 确定 ID 编号

对于查找词，每个文件必须确定一个 ID 编号 (例如：1...n)。ID 编号符合简短帮助 "ASCII" 指令的列表，因为对于简短说明应显示详细的说明：

举例：

840D_PG.PDF 44 MEAW 155

<PDF 文件名> <ID> <查找词> <页 >

现在删除列“查找字”，剩下下列句法：

结果：

<PDF 文件名> <ID> <页 >

840D_PG.PDF 44 155

4) 生成 PDF 文件

把文件作为后脚本文件打印出来，并用 Adobe Acrobat Distiller 把它转换为 "PDF" 格式。

5) 把文件读入控制系统

生成的文件 (*.PDF 和 *.TXT) 必须读入控制系统，例如：读入目录 F:\DH\CUS.DIR\HLP 中。

注意

对于一个更大的索引容量，可以通过微软的 Excel 自动创建一个调用文件：

- 在 Excel 中插入文本文件 (*.TXT)
- 插入列
- 在第一行中输入 PDF 文件的名称
- 执行“编辑 - 填充”
- 现在删除列“查找字”，剩下下列句法：

<PDF 文件名> <ID> <页>

840D_PG.PDF 44 155

- 复制数据并
- 保存为 "TXT" 文件

1.4 ISO 模式帮助

1.4.1 功能

功能

正常情况下，显示 840D / 840D sl 专用的帮助：

- 编辑器中的帮助
- 报警帮助
- 机床数据帮助

如果要显示 ISO 语言 0 的帮助，则必须区分要显示哪种数据，因为例如：相同的指令具有不同的含义。

为此，文件 "IF.INI" 在 [TEXTFILES] 部分扩展了项 Siemens=CST.DIR\IFIT_COM (车削工艺) 或者 SiemensI=CST.DIR\IFIM_COM (铣削工艺)，从中给出了用于 ISO 模式简短帮助 "ASCII" 的路径 (CST.DIR) 和名称 (IFIT_COM 或 IFIM_COM)。

由简短帮助 "ASCII" 可以相应地调用详细帮助 "PDF"。

IF.INI [TEXTFILES]

[TEXTFILES]

；文本文件中包含帮助系统中显示的信息

Siemens=cst.dir\ifs_.com

SiemensI=cst.dir\ifit_.com 或 ifim_.com

制造商=

最终用户=

必要的文件

- 带有所属简短说明的“ISO 语言 0”指令列表
- TXT 文件用于跳转到 PDF 文件
- 应在帮助中显示的 PDF 文件

程序运行过程

引导启动时，该帮助检查下列机床数据：

- MD18800 MM_EXTERN_LANGUAGE
如果激活了 MD，则控制系统切换到模式 ISO 语言 0。
如果 MD 18800 = 1，则使用 MD 20150。
- MD20150 GCODE_RESET_VALUES[n(当前通道),47]
当 MD 20150 [47] = 2 时，使用在 INI 文件中"SiemensI" (ISO 模式)下输入的文件
(否则使用 "Siemens" 下输入的文件)。
- MD10880 MM_EXTERN_CNC_LANGUAGE
用 MD 10880 确定，是否该控制系统设计用于 ISO 模式的车削或铣削。

1.4.2 扩展 ISO 模式帮助

如果要在 ISO 模式帮助中显示附加的说明，必须按以下格式插入：

- 作为目录中带有简短说明的 ISO 模式指令列表
"SiemensI=CST.DIR\" 作为文件 "IFIT_COM" 或文件 "IFIM_COM"
(参见章节“文本文件的扩展”)，
- 作为帮助说明 (PDF 文件) 和相同内容的 TXT 文件在目录 "F:\USER\HLP" 中插入。

指令定义的参数

工艺/标题 ID/扩展帮助/指令文本/
说明文本

“扩展的帮助”：

在参数“扩展帮助”中可能有下列项：

- 编辑器帮助
/"PDF 文件名", 帮助 ID, "页数"/
- 循环帮助
/"PDF 文件名", 帮助 ID, "页数" (文件名, 帮助画面)/

必要的文件

- IFIT_.COM 或 IFIM_.COM : 带有 ISO 模式指令和简短说明的文本文件
- *.PDF: 例如 : 840D_PGIT_GR.PDF
- *.TXT: 例如 : 840D_PITG_GR.TXT

1) 确定文档中的跳转标识

在控制系统中显示为 PDF 的文件 (例如 : WinWord) 中 , 必须用一个索引项标出跳转标识 , 并且生成单列索引目录 (页码右对齐) 。

2) 生成文本文件

复制上面生成的单列索引 (页码右对齐) 并把它作为 TXT 文件保存。

举例 :

G65 155

<查找词> <页>

文本文件用于 PDF 调用时需要下列句法 :

<PDF 文件名> <查找词> <页 >

在每行前插入 PDF 文件的名称 (**无语言标识**) 。

举例 :

840D_PGIT.PDF G65 155

<PDF 名称> <查找词> <页 >

3) 确定 ID 编号

对于查找词 , 每个文件必须确定一个 ID 编号 (例如 : 1...n) 。 ID 编号符合简短帮助 "ASCII" 指令的列表 , 因为对于简短说明应显示详细的说明 :

举例 :

840D_PGIT.PDF 44 MEAW 155

<PDF 文件名> <ID> <查找词> <页 >

现在删除列“查找字” , 剩下下列句法 :

结果 :

<PDF 文件名> <ID> <页 >

840D_PGIT.PDF 44 155

4) 生成 PDF 文件

把文件作为后脚本文件打印出来 , 并用 Adobe Acrobat Distiller 把它转换为 "PDF" 格式。

5) 把文件读入控制系统

生成的文件 (*.PDF 和 *.TXT) 必须读入控制系统的目录 F:\USER\HLP 中。

注意

对于一个更大的索引容量，可以通过微软的 Excel 自动创建一个调用文件：

- 在 Excel 中插入文本文件 (*.TXT)
- 插入列
- 在第一行中输入 PDF 文件的名称
- 执行“编辑 - 填充”
- 现在删除列“查找字”，剩下下列句法：

<PDF 文件名> <ID> <页 >

840D_PGIT.PDF 44 155

- 复制数据并
 - 保存为 "TXT" 文件
-

报警/机床数据帮助

2.1 概述

下列章节中描述了报警和机床数据帮助的功能和扩展性。

软件

- HMI 高级
- Adobe Reader 4.x (可通过 Internet / 专业贸易免费获取)
通常软件 "Adobe Acrobat Reader 4.x" 在控制系统 SINUMERIK 840D/840D sl 中已安装
- Adobe Acrobat 4.x
需要软件 "Adobe Acrobat Distiller 4.x" 来把文件转换成 PDF 格式并编辑 PDF 文档。

辅助文件存放

编程帮助：F:\DH\CST.DIR\HLP.DIR (可能包含 HLP, PDF 和 TXT 文件)
报警帮助/机床数据帮助：F:\MMC2\HLP (可能包含 HLP, PDF 和 TXT 文件)
BTSS 帮助：F:\HMI_ADV\HLP (可能包含 HLP, PDF 和 TXT 文件)

PDF 帮助原理

PDF 帮助需要帮助文件存储中的下列部分：

- 调用文件，例如：PG_GR.TXT，带数据说明
<PDF 文件名> <ID> <查找概念> 每次调用
- 所属的 PDF 文件

按照下列步骤进行帮助调用：

- 启动 "Adobe Acrobat Reader" (包含在控制系统中)
- 跳转到页数 (通过索引在文件中确定)
- 查找概念 (Adobe 软件的查找功能)

创建时的步骤

- 在帮助文件中确定 (例如 : 通过 DTP 系统的标准索引功能) 并生成跳转
- 保存跳跃文件作为文本文件
- 生成 PDF 文件

结果 :

- 带有数据说明的调用文件 (文本文件)
<PDF 文件名> <ID> <查找概念 >每次调用
- 一个或者多个所属的 PDF 文件

注意

对一个主题有多个跳跃时 , 必须在另一行中写出页码。

举例 :

```
840D_PG.PDF 44 155  
840D_PG.PDF 45 180  
840D_PG.PDF 46 199
```

提示 :

对于一个更大的索引容量 , 可以通过微软的 Excel 自动创建一个调用文件 :

- 在 Excel 中插入“索引文件” (*.TXT)
- 插入列
- 在第一行中输入 PDF 文件的名称
- 执行菜单“编辑” - “填充” - “下一步”
- 复制数据并
- 保存为 "TXT" 文件

此外还有以下句法 :

```
<PDF 文件名> <ID> <查找概念 >  
840D_PG.PDF 44 155
```

在文件中测试跳跃

为了测试是否能找到所有的查找词，按下列步骤进行：

- 启动 Adobe Acrobat
- 选择菜单“编辑”->“查找”(CTRL + F)
- 输入查找词

注意

如果没有找到查找词，可能有以下原因：

- 输入错误
- 生成 PDF 时，文本转换不正确，所以搜索器无法找到。

*分析：*用 Adobe Acrobat 选中要查找的词，复制并插入 Word 文档。

解决方法：

- 重新生成 PDF 或者
 - 用 Adobe 打印机驱动器 Adobe PS 5.1.1 生成后脚本文件，然后用 Adobe Distiller 进行转换。
-

查找顺序

对于帮助遵循下列查找顺序：

- F:\USER\HLP
- F:\OEM\HLP
- F:\ADD_ON\HLP
- F:\MMC2\HLP
- F:\HMI_ADV\HLP
- F:\DH\CUS.DIR\HLP.DIR (用户目录)
- F:\DH\CMA.DIR\HLP.DIR (制造商目录)
- F:\DH\CST.DIR\HLP.DIR (西门子目录)
- F:\DH\COM.DIR\HLP.DIR

2.2 报警帮助

2.2.1 功能

调用帮助时打开一个 PDF 文件，并在控制系统中用 Adobe Reader 显示相应的报警。

带有报警的文件位于下列目录：

- 西门子报警：F:\HMI_ADV\HLP
- 制造商报警：F:\DH\CMA.DIR\HLP
- 用户报警：F:\DH\USER\HLP

文件名

即使对于附加的用户报警，也有义务遵循下列文件名（不允许更改）：

- ALARM_GR.PDF [德语报警]
- ALARM_UK.PDF [英语报警]
- ALARM_FR.PDF [法语报警]
- ALARM_IT.PDF [意大利语报警]
- ALARM_SP.PDF [西班牙语报警]

所属的 TXT 文件（包括跳转列表）的名称必须相对应：

- ALARM_GR.TXT [德语 TXT 文件]
- ALARM_UK.TXT [英语 TXT 文件]
- ALARM_FR.TXT [法语 TXT 文件]
- ALARM_IT.TXT [意大利语 TXT 文件]
- ALARM_SP.TXT [西班牙语 TXT 文件]

其它语言

报警文本有下列语种：

CH: 简体中文

TW: 繁体中文

KO: 韩语

JA: 日语

SW: 瑞典语

HU: 匈牙利语

PO: 葡萄牙语

CZ: 捷克语

TR: 土耳其语

RU: 俄语

PL:波兰语

NL:荷兰语

必须按照上述名称约定使用这些缩写，用于用户报警的文件名（不允许更改）。

2.2.2 扩展报警帮助

如果要在报警帮助中显示附加的用户报警，必须把它以 PDF 文件格式和相同内容的 TXT 文件格式添加到目录 C:\USER\HLP 中。

必要的文件

- *.PDF: 例如：ALARM_GR.PDF
- *.TXT: 例如：ALARM_GR.TXT

1) 确定文档中的跳转标识

在控制系统中显示为 PDF 的文件（例如：WinWord）中，必须用一个索引项标出跳转标识，并且生成单列索引目录（页码右对齐）。

2) 生成文本文件

复制上面生成的单列索引（页码右对齐）并把它作为 TXT 文件保存。

举例：

10702 90

<ID> <页>

查找词（报警编号）在此和 ID 编号一致。

文本文件用于 PDF 调用时需要下列句法：

<PDF 文件名> <ID> <页>

在每行前插入 PDF 文件的名称。

举例：

ALARM.PDF 10702 90

ALARM.PDF 10703 90

ALARM.PDF 10704 91

<PDF 名称> <ID> <页>

3) 生成 PDF 文件

把文件作为后脚本文件打印出来，并用 Adobe Acrobat Distiller 把它转换为 "PDF" 格式。

4) 把文件读入控制系统

生成的文件 (*.PDF 和 *.TXT) 必须读入控制系统，例如：读入目录 F:\DH\USER\HLP 中。

2.2.3 用于帮助的用户注释

功能

用于报警的帮助 (诊断中的 [j] 键) 提供功能“记录段”，用该功能可以输入单个报警的用户专用文本。这些文本存放在下列目录中：

F:\USER\RH

注意

如果用键 [j] 把 HMI 帮助从报警概览或者报警记录调用到相应的报警，报警专用文本可以被保存。

无法从目录中选择报警。

2.2.4 把报警帮助 "Win 帮助" 转换为 "PDF 帮助"

如果要把帮助功能从“Win 帮助”转换为“PDF 帮助”，该过程符合章节：“报警帮助扩展”中的说明：

1. 为每个报警号确定文档中的跳转标识：

在控制系统中显示为 PDF 的文件 (例如：WinWord) 中，必须用一个索引项标出跳转标识，并且生成单列索引目录 (页码右对齐)。

2. 生成文本文件：

复制上面生成的单列索引 (页码右对齐) 并把它作为 TXT 文件保存。

文本文件用于 PDF 调用时需要下列句法：

<PDF 文件名> <ID> <页>

查找词 (报警编号) 在此和 ID 编号一致。

3. 在每行前插入 PDF 文件的名称。

举例：

ALARM.PDF 10702 90

ALARM.PDF 10703 90

ALARM.PDF 10704 91

<PDF 名称> <ID> <页>

4. 带有报警文本的文件，作为后脚本文件打印 (可从所有常用平台进行)。用 WinWord 生成文件后缀例如 *.prn，用 Interleaf 生成 *.ps。

5. 用软件 Adobe Acrobat，确切说是用 Distiller，把后脚本文件转换成 "PDF"。

6. 将文件 (*.PDF 和 *.TXT) 读入控制系统的目录 F:\DH\USER\HLP.DIR 中。

注意

如果要在报警帮助中显示附加的用户报警，必须把它以 PDF 文件格式和相同内容的 TXT 文件格式添加到目录 F:\USER\HLP 中。

请注意在章节“文件名”中提到的名称约定。

2.3 机床数据帮助

2.3.1 功能

机床数据帮助以“PDF 帮助”为基础，也就是说，调用帮助时打开一个 PDF 文件，并在控制系统中用 Adobe Reader 显示相应的报警。

带有机床数据的文件位于下列目录：

- 西门子机床数据：F:\MMC2\HLP
- 用户机床数据：F:\USER\HLP

文件名

即使对于附加的用户机床数据，也有义务遵循下列文件名（不允许更改）：

- N_NCK_GR.PDF [德语], N_NCK_UK.PDF [英语]
- C_NCK_GR.PDF, C_NCK_UK.PDF
- A_NCK_GR.PDF, A_NCK_UK.PDF
- S_NCK_GR.PDF, S_NCK_UK.PDF
- DISPL_GR.PDF, DISPL_UK.PDF
- DROT_GR.PDF, DROT_UK.PDF
- DLIN_GR.PDF, DLIN_UK.PDF
- HYDRM_GR.PDF, HYDRM_UK.PDF

所属的 TXT 文件（包括跳转列表）的名称必须相对应：

- N_NCK_GR.TXT [德语], N_NCK_UK.TXT [英语]
- C_NCK_GR.TXT, C_NCK_UK.TXT
- A_NCK_GR.TXT, A_NCK_UK.TXT
- S_NCK_GR.TXT, S_NCK_UK.TXT
- DISPL_GR.TXT, DISPL_UK.TXT
- DROT_GR.TXT, DROT_UK.TXT
- DLIN_GR.TXT, DLIN_UK.TXT
- HYDRM_GR.TXT, HYDRM_UK.TXT

文本文件结构

无语言标识的 PDF 文件	机床数据编号	页
DROT.PDF	1000	1
DROT.PDF	1001	1
DROT.PDF	1002	1
...

名称的意义

名称	含义
N_NCK_GR.PDF	通用机床数据
C_NCK_GR.PDF	通道专用机床数据
A_NCK_GR.PDF	轴专用机床数据
S_NCK_GR.PDF	设定数据
DISPL_GR.PDF	显示机床数据
DROT_GR.PDF	驱动机床数据，旋转
DLIN_GR.PDF	驱动机床数据，直线
HYDRM_GR.PDF	驱动机床数据，液压模式

2.3.2 扩展机床数据帮助

如果要在 MD 帮助中显示附加的用户机床数据，必须把它以 PDF 文件格式和相同内容的 TXT 文件格式添加到目录 C:\USER\HLP 中。

必要的文件

- *.PDF: 例如：N_NCK_GR.PDF
- *.TXT: 例如：N_NCK_GR.TXT

1) 确定文档中的跳转标识

在控制系统中显示为 PDF 的文件（例如：WinWord）中，必须用一个索引项标出跳转标识，并且生成单列索引目录（页码右对齐）。

2) 生成文本文件

复制上面生成的单列索引 (页码右对齐) 并把它作为 TXT 文件保存。

举例：

10000 170

<ID> <页>

查找词 (MD 编号) 在此和 ID 编号一致。

文本文件用于 PDF 调用时需要下列句法：

<PDF 文件名> <查找词> <页 >

在每行前插入 PDF 文件的名称。

举例：

N_NCK.PDF 10000 170

N_NCK.PDF 10002 170

N_NCK.PDF 10010 171

<PDF 名称> <ID> <页 >

3) 生成 PDF 文件

把文件作为后脚本文件打印出来，并用 Adobe Acrobat Distiller 把它转换为 "PDF" 格式。

4) 把文件读入控制系统

生成的文件 (*.PDF 和 *.TXT) 必须读入控制系统的目录 F:\USER\HLP 中。

索引

C

CONTEXT, 9

I

IF.INI, 10

IFM_GR.COM, 9

ISO 模式帮助, 20

 扩展, 21

T

TECHNOLOGY, 8

TEXTFILES, 8

TEXTSEARCH, 10

三划

工艺, 14

四划

文本文件

 扩展, 10

六划

扩展报警文本变量, 17

机床数据帮助, 29

 文件名, 29

 扩展, 30

七划

初始化文件, 文件 IF.INI, 8

报警文本

 外语缩写, 26

报警帮助, 25

 文件名, 25

 扩展, 26

 注释, 27

八划

详细帮助 PDF

 参数化, 14

九划

帮助文件

 存档, 5

帮助功能, 7, 17

 扩展, 8

帮助参考

 几个, 16

指令

 文本属性, 16

 扩展, 13

 定义, 部分, 13

查找顺序

 帮助, 7, 25

标题

 扩展, 12

 定义, 11

标题 ID, 12

说明

 文本属性, 16

十二划

编程辅助

 扩展, 17

十三划

简短帮助 ASCII, 7, 17

跳转文件

 创建报警帮助/机床数据帮助, 23

 创建编程帮助, 6

 测试, 6, 24

SIEMENS

SINUMERIK

SINUMERIK

840D sl/840Di sl/840D/840Di/810D 创建外语文本 (TX2)

调试手册

引言

1

使用 Word 编辑文件

2

文本输入

3

问题和消除方法

4

用于生成文本的文献

5

适用于：

控制系统

SINUMERIK 840D sl/840DE sl

SINUMERIK 840D powerline/840DE powerline

SINUMERIK 840Di powerline/840DiE powerline

SINUMERIK 810D powerline/810DE powerline

软件

HMI 高级

HMI 内置

版本

V7.3

V7.2

11/2006

6FC5397-0DP10-0RA0

安全技术提示

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

带有警告三角，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。

小心

不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。

注意

表示如果不注意相应的提示，可能会出现不希望的结果或状态。

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

仅允许安装和驱动与本文件相关的附属设备或系统。设备或系统的调试和运行仅允许由**合格的专业人员**进行。本文件安全技术提示中的合格专业人员是指根据安全技术标准具有从事进行设备、系统和电路的运行，接地和标识资格的人员。

按规定使用

请注意下列说明：



警告

设备仅允许用在目录和技术说明中规定的使用情况下，并且仅允许使用西门子股份有限公司推荐的或指定的其他制造商生产的设备和部件。设备的正常和安全运行必须依赖于恰当的运输，合适的存储、安放和安装以及小心的操作和维修。

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。标签中的其他符号可能是一些其他商标，这是出于保护所有者权利的目地由第三方使用而特别标示的。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

目录

1	引言	5
1.1	概述	5
1.2	前提条件	5
1.3	语族	6
1.4	使用工具	7
1.5	术语	8
2	使用 Word 编辑文件	11
2.1	概述	11
2.2	创建新文本文件	11
2.3	打开已有文件	12
2.4	保存文本文件	14
2.5	SINUMERIK HMI 编码	16
3	文本输入	17
3.1	不同语族的特殊字符	17
3.2	通过字符映射表输入	18
3.3	复合输入	20
3.4	使用其它键盘布局	21
3.4.1	键盘语言属性方案	22
3.4.2	在 Windows XP 下安装键盘布局	22
3.4.3	使用	24
3.5	数字文本输入	26
3.6	Word 文档中的语言属性	26
3.7	多语言 Word 文档	29
3.8	希腊语和西里尔语文本的特点	31
3.9	东亚语言文本的特点	33
4	问题和消除方法	37
5	用于生成文本的文献	39
	索引	41

表格

表格 1-1	语族	6
表格 2-1	编码	16
表格 3-1	文本文件编码	19

引言

1.1 概述

90 年代以来，微软公司一直致力于满足世界范围信息处理和软件开发的要求。

除了尽可能支持多语言外，还追求这样一个目标，尽力开发一个统一的可以支持所有语言的操作系统版本，与早期使用的分别针对各个语言的操作系统版本不同。

经过努力已经逐步将许多系统组件和应用程序转换至跨语言平台 Unicode。

这些转换在很多情况下对于用户来说是隐蔽的，在此期间，会有一个基本不同的操作步骤用于在本指南中描述的主题，取决于使用哪种操作系统版本。

推荐用于 Windows XP 的操作步骤相比于 Windows 95 / NT4 要简便得多。因此，建议改装新的系统。旧系统的指南不再进行编辑更新。

1.2 前提条件

前提条件

该文档描述了“创建文本”的操作步骤，说明如下：

- 文本应当在微软 Windows 操作系统下创建（Windows XP）。
- 应以和操作系统语言（= 标准语言）不同的语言来创建文本。
- 是以德语作为标准语言出发来举例描述的。但是所描述的操作步骤在使用其他 Windows 标准语言时也是可以使用的。
- 文本应创建为文本文件，即无字体，字体大小，下划线等设置。除了文本字符外，文件只包含 ASCII 控制字符，如 Tab、CR 和 LF（制表、退格和回车）。
- 除了标准语言，文本语言要求一个另外的 Windows 代码页，也就是说，目标语言属于标准语言外的另一个语族。
- 文本文件应用该代码页来编码，而不是作为 Unicode 文件。
- 创建文本应尽可能简化操作，也就是说，特别是尽可能不安装附加软件。
- 该指南假设您已经安装了微软 Word 2003 或者更高版本的产品，并且是相信该软件的。

HMI 内置和 HMI 高级的文本类型

可以在单个系统中更改下列文本类型：

文本类型	系统
报警文本 补充用于操作界面的文本 所有操作区域的应用文本，包括软键文本	HMI 内置
报警文本 补充用于操作界面的文本 编辑器中的帮助文本 机床数据 所有操作区域的软键文本	HMI 高级

1.3 语族

语族是由几种语言构成的组，对于该语言组微软操作系统使用相同的代码页。微软使用概念“语言组”。

表格 1-1 语族

语族	Windows 代码页	语言
中欧	1250	阿尔巴尼亚语、克罗地亚语、波兰语、罗马尼亚语、塞尔维亚语（拉丁语）、斯洛伐克语、斯洛文尼亚语、捷克语、匈牙利语
西里尔文	1251	保加利亚语、马其顿语、俄语、塞尔维亚语（西里尔语）、乌克兰语、白俄罗斯语
西欧	1252	德语、英语、芬兰语、法语、印度尼西亚语、冰岛语、意大利语、荷兰语、挪威语、葡萄牙语、瑞典语、西班牙语
希腊语	1253	希腊语
土耳其语	1254	土耳其语
波罗的海文	1257	爱沙尼亚语、拉脱维亚语、立陶宛语
日语	932	日语
中文	936	简体中文
韩语	949	韩语
中文	950	标准中文（繁体中文）

操作系统和目标语言来自相同语族

如果操作系统的语言（例如：德语）和目标语言（例如：葡萄牙语）属于相同的语族，则在输入单个字符时可能会有问题，而不是在纯文本编辑器中显示字符时有问题。

消除方法：

- 使用 Windows 辅助工具“字符映射表”
（见章节“字符映射表输入”）或者
- 切换键盘
（见章节“使用其它键盘布局”）或者
- 使用 ALT+ 数字键组合输入单个字符
（见章节：“数字文本输入”）

其他问题在该情况下应该不会出现。
请使用纯文本编辑器，例如：记事本，并按常规保存文本。
无需考虑章节“Word 2003 文件”。

操作系统和目标语言来自不同语族

如果操作系统的标准语言（例如：德语）和目标语言（例如：匈牙利语）属于不同的语族，则输入字符和在纯文本编辑器（记事本）中显示字符时都可能出问题。

1.4 使用工具

标准功能范畴

在本文档中描述了微软 Word 2003 和其他帮助工具的使用，您的 PC 不一定已经安装了这些帮助工具。

应该已经安装了下列组件：

- 微软 Word 2003（必须）
- Windows XP: 用于东亚语言的支持软件（对于这些语言必须）
- Windows XP: 字符映射表（推荐）

检查东亚语言支持软件的安装

东亚语言支持软件的安装见章节“东亚语言特点”。

1.5 术语

代码页

是一个字符列表，由一种或者几种语言的字符组构成。每个列表中的字符都分配有一个编码，该编码在代码页内是唯一的。

所有在 Windows 下使用的代码页有共同的 ASCII 范围（编码 0 到 127）。

区别单字节代码页和多字节代码页。

单字节代码页（所有欧洲语言）的每个字符用一个唯一的字节来表示。因此单字节代码页包含 256 个字符。

多字节代码页（亚洲语言）不仅包含一个字节字符，而且包含两个（或者更多）字节的字符。包含了共同的 ASCII 范围作为单字节字符的范围。

在 Windows 代码页和语族之间是 1:1 的分配关系。

语族详见章节“语族”。

变音符

（变音 = 特殊）

变音符通常是一个相对较小的附加符号，加在一个字母上来改变字母重音，发音或者含义。

变音字符表见章节“不同语族的特殊字符”。

如果基础字母和变音符的组合属于一个语言专用的字母表，则作为新的独立的字符包含在相应的 Windows 字符集（代码页，Unicode）中。

变音符的应用

变音符用于

- 命名组合字符（例如：Ğ = G 软音符，Ö = O 发音符号），
- 复合输入，如果不存在带有组合字符的按键时
（见章节“使用其它键盘布局”），
- 越南语和泰语文本文件中的正常编码，
- 仅在印刷格式设置的文本中用于输出：
变音符和基础字母会分开输出，这样可以进行任意的组合，
例如：在使用旧语言和音标时。

在所有其他情况下，组合的字母会被视作不可分的单元。

当然有时会错误地将组合的字母作为变音符。

输入方法编辑器 (IME)

用于输入 CJK 字符的辅助工具（CJK：中文、日文和韩文的缩写）。

语族

语族是一个由使用相同代码页的语言构成的组。这里只是假设使用了相同的基础字母表（例如：拉丁语或者西里尔语），但并不表示该组中所有语言的字母表是相同的。

在 Windows 代码页和语族之间是 1:1 的分配关系。

语族详见章节“语族”。

字体文件 (TrueType 字体)

在 Windows XP 中不使用每个代码页都分开存储的字体文件，而是使用“Big Fonts”，这是一种包含多个代码页字符组的字体。用这种方式可以节省共同字符的存储空间。

从共同字符组选择字符，在使用 Word 2003 时不是通过代码页编码，而是通过 Unicode 编码。因此可以避免混淆属于不同代码页但是却具有相同编码的字符。

标准语言

在本指南中标准语言是指操作系统的语言，如：德语的 Windows 时为德语，英语的 Windows 时为英语。

标准键盘设置

在“控制面板” → “区域和语言选项”下，标签“语言”中，点击“文字服务和输入语言”标题下的按钮“详细信息...”，来打开“文字服务和输入语言”对话框。

此处在此标签“设置”中，“默认输入语言”标题下是标准键盘设置。

标准键盘设置由语言属性（左）和键盘布局（右）组成。

标准键盘设置的语言属性不一定要和标准语言一致。

文本文件

文本文件可以理解为一个不包含任何格式（字体、字号、下划线、粗体、斜体）和嵌入对象（图片、表格、图表、脚注）的文件。

除了文本字符，该文件仅包含 ASCII 控制字符，如 Tab、CR 和 LF（制表、退格和回车）。

文本字符取自一个特定的代码页。

有时文本文件也被错误地称为 ASCII 文件。只有当文本文件仅包含 ASCII 字符并且因此代码页不变时，这种表达才是正确的。

Unicode

Unicode 是一个 16 位字符集，使用该字符集时不再需要代码页。

所有字符编码在没有代码页说明的情况下是单一的。Unicode 程序，如 Word 2003，可以不依赖于代码页来解释字符，并且也不通过文字或者语言属性的分配来显示其他字符。

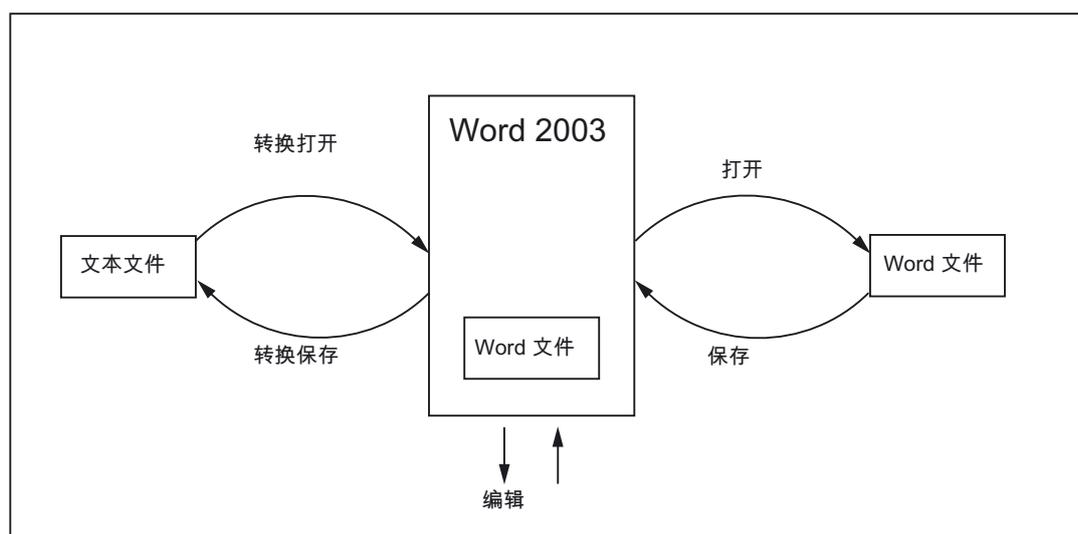
使用 Word 编辑文件

2.1 概述

本章节描述如何用微软 Word 2003 来编辑文本文件。

Word 2003 不直接编辑文本文件，而是转换为 Word 文件来编辑。
保存文件时，Word 可以将编辑的文本再次转换为一个文本文件。

建议为文本文件额外保存一个 Word 版本，作为对文档进行其它处理的基础。
文本文件的“转换打开”取消。



2.2 创建新文本文件

1. 按以下方法创建新文本文件：

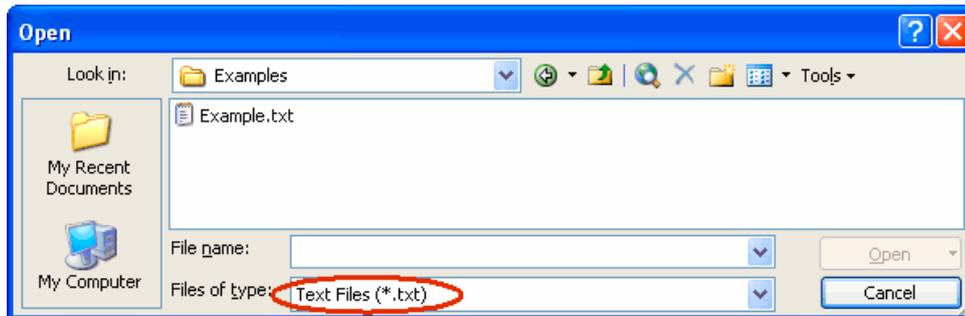
- 在 Windows Explorer 中选择菜单“文件”→“新建”→“文本文档”。
- 启动记事本，不打开文件。打开一个新的空文件。选择菜单“文件”→“另存为”→“编码：ANSI”
- 复制一个现有文本文件，作为新文本文件的模板。

2. 现在用 Word 2003 打开该文本文件，操作方法见下一章节的描述。

2.3 打开已有文件

1. 在 Word 2003 中从“文件”菜单选择指令“打开...”。
2. 在“打开”对话框中“文件类型”选项下选择“文本文件(*.txt)”。

如果文件的后缀名不为“txt”，则必须重命名文件。



3. 选择您想要打开的文件，并点击“打开”。

步骤 1 到 3 也可以用下列简化步骤来替代：

- 在 Word 中使用最近几个打开过的文件列表。
- 从浏览器中将文本文件拖到桌面的 Word 图标上。
- 如果 Word 已经启动，则从浏览器中将文本文件拖到 Word 窗口的标题栏中。

4. 请注意现在出现的情况：

如果“转换文件”对话框（参见步骤 5）打开，则在那里继续。

如果“文件转换”对话框（参见步骤 6）打开，则在那里继续。

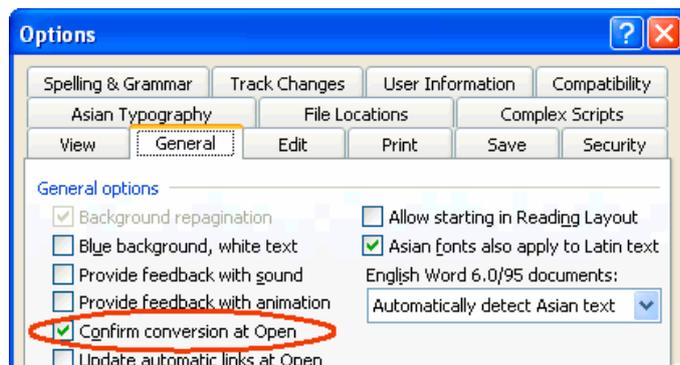
然而，如果现在立即打开了该文件，而此时“转换文件”对话框和“文件转换”对话框都没有打开，则 Word 可能开错了文件，假设是这种情况，会以标准语言编辑文本。

请注意特殊字符。当文本文件的代码页和标准语言的代码页不一致时，特殊字符的显示会出错。

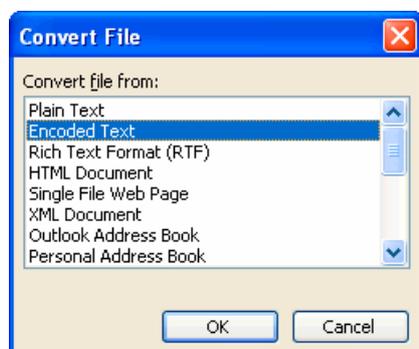
该情况下通过“工具”→“选项...”打开 Word 选项对话框，在“常规”标签页上激活选项“打开时确认转换”。

如果该选项已激活，则附加对话框“转换文件”（参见步骤 5）每次都会出现，如果选项未激活，则会跳过该对话框。

接着关闭该文件并且重新从步骤 1 开始。

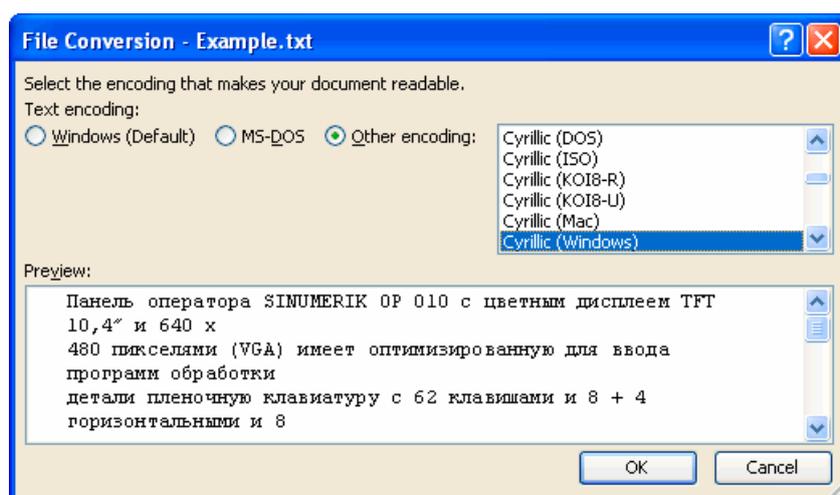


5. 现在出现对话框“文件转换”。
- 选择“编码文本”并点击“确认”。



6. 现在出现对话框“文件转换 - Example.txt”：

- 如果标记为“Windows (标准)”，则激活“其它编码”。选择正确的编码 (Word 会给出一个建议，但不一定是正确的)。



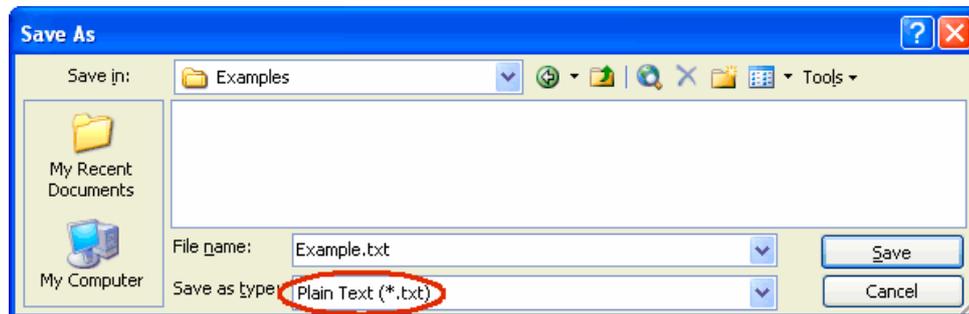
- 在章节“SINUMERIK HMI 编码”中，表中列出了哪些编码是合适的。对此请注意预览区。

7. 点击“确定”。
打开文件并转换到一个 Word 文件。
现有的文本以一个标准字体进行格式化。

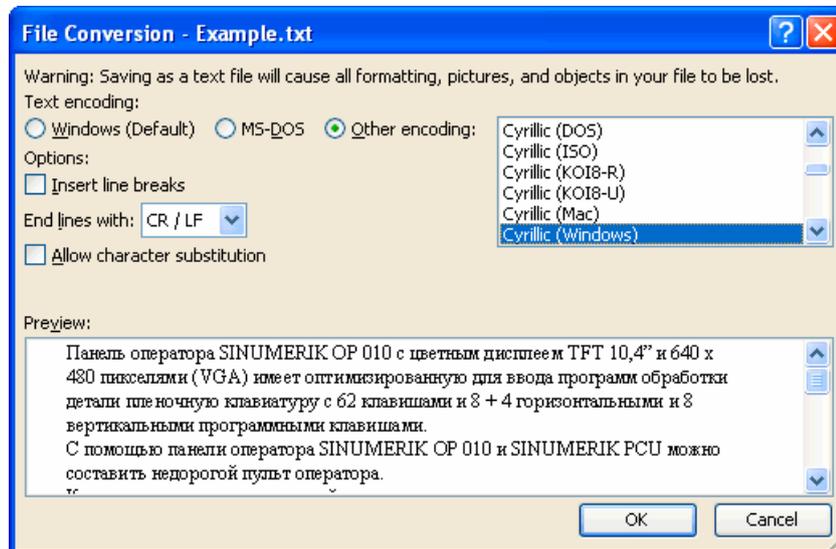
2.4 保存文本文件

将已输入的文本保存为“文本文件”：

1. 在 Word 中从“文件”菜单选择指令“另存为...”

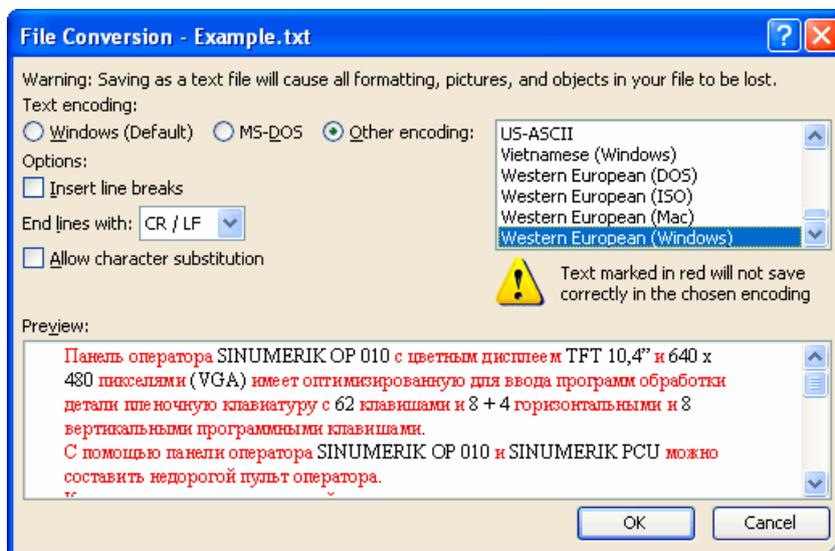


2. 在“文件类型”选项下选择“仅为文本文件(*.txt)”。
3. 在“文件名”一栏中输入名称，并点击“保存”。
 - 现在出现对话框“文件转换”。



- 选择适合您目标语言的编码，并点击“确定”。在章节“SINUMERIK HMI 编码”中，表中列出了哪些编码是合适的。
- 您的文件现在保存为文本文件，并且重新编码为所选择的代码页。

- 如果您选择了一个完全不合适的编码，则对话框中会一直出现警告提示。虽然您选择了正确的编码，但如果文件中包含不允许的字符，例如：在西里尔语文本中有拉丁语变元音，则也会出现该警告提示。



- 如果忽略该警告提示，则所有不能编码的字符会用问号“?”来代替。

2.5 SINUMERIK HMI 编码

如果打开或保存一个文件，不使用代码页编号，而是用代码页名称，参见章节“打开已有文件”或“保存文本文件”。

分别根据系统的安装也可能列出许多其他的编码，其中包括用于 Unix (ISO), Apple (Mac), OEM (DOS) 和 IBM (EBCDIC) 平台的编码。

对于 SINUMERIK HMI 请使用 Windows (ANSI) 编码。

请使用下列编码来打开和保存文件（按字母表顺序排列）：

表格 2-1 编码

名称	代码页
繁体中文 (Big5)	950
简体中文 (GB2312)	936
日文 (Shift-JIS)	932
韩语	949
西里尔文 (Windows)	1251
中欧 (Windows)	1250
土耳其文 (Windows)	1254
西欧 (Windows)	1252

文本输入

3.1 不同语族的特殊字符

拉丁语族

使用带拉丁语基础字母的语族（波罗的海文、中欧、土耳其文、西欧）时，大多数字符，即拉丁语的基础字母表，可以直接用（例如：西欧）键盘布局来输入。

大多数特殊字符，例如 Aáâãçđéèíñňöóúū，由拉丁字母 A-Z a-z 和变音符号组成，如：尖音符 '，短音符 ˘，软音符 ˆ，抑扬符 (Caret) ^，圆点符 ˙，双圆点符 (Trema) ¨，双尖音符 ˚，沉音符 ˘，扬抑符 (Caron) ˇ，长音符 ˉ，鼻音符 ̃，小圆圈 ˚，斜线 /，横线 – 或发音符号 ˘。

其他字母表字符是特殊字母 (þ, ð, ß) 或者连字 (Ææ, Œœ, IJij)。

此外还有各语言专用的省略号和标点符号，例如：西班牙语中用于序数末尾的 (ª 和 º) 以及倒立的引号和问号 (¡ 和 ¿)。

以下描述的输入方法仅对于使用拉丁语基础字母的语族完全适用。

希腊/西里尔语族

输入希腊语或者西里尔语字符的文本还会有一些特殊性。

对此请参见章节“希腊和西里尔语文本的特点”。

东亚语言

创建日语、中文和韩语的文本还会有一些特殊性。

对此请参见章节“东亚语言文本的特点”。

Word 中的干扰选项

取消激活 Word 中的下列选项，以便文本中不会出现不需要的字符：

在对话框“工具”→“自动更正...”，标签“键入时自动套用格式”，“键入时自动应用”中：

- 直引号替换为弯引号
- 序号 (1st) 替换为上标
- 分数 (1/2) 替换为分数字符 (½)
- 连字符(--)替换为长划线(—)

“键入时自动应用”：

- 自动项目符号列表
- 自动编号列表

如果激活单个选项，则可以在进行自动替换后用按键组合 Ctrl+Z 来撤销替换。通常情况下，获得的结果和未激活选项时是一样的。

3.2 通过字符映射表输入

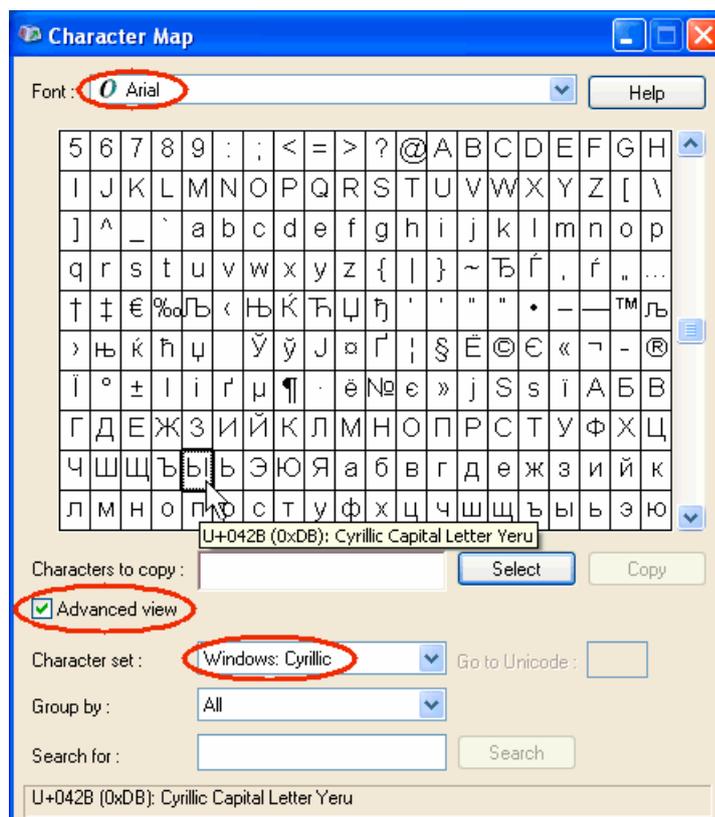
注意

Word 中集成的字符映射表 (菜单“插入”→“符号...”) 不适合用于生成代码页编码的文本文件，因为无法限制各代码页。

请按下列描述来使用系统程序“字符映射表”。

选择字符映射表

1. 在开始菜单中选择
“程序”→“附件”(“Accessories”)→“系统工具”→“字符映射表”，
来启动辅助工具“字符映射表”。
2. 在“字体”下选择与文本文件所选字体相同的字体。



3. 激活复选框“高级查看”。
4. 在“字符集”下选择您文本文件的代码页，以便将所提供字符的数量限制在该代码页。
代码页如下：

表格 3-1 文本文件编码

名称	代码页
Windows : 中文 (台湾地区)	950
Windows : 中文 (中国)	936
Windows : 日语	932
Windows : 韩语	949
Windows : 西里尔文	1251
Windows : 东欧	1250
Windows : 土耳其语	1254
Windows : 西欧	1252

输入字符

1. 选择一个您想要输入的字符。
在下面的信息栏中显示 Unicode 编码 (U+..., 十六进制)、代码页编码 (在括号中) , 以及字符的明码文本描述。
如果适当地切换了键盘 (参见章节 3.4 : 使用其他键盘布局) , 则在右下方的状态栏中会显示用于数字输入 (参见章节“数字的文本输入”) 的按键组合。



2. 点击“选择”。
该字符会拷贝到“复制字符”栏中。
3. 重复步骤 1 和 2 可以选择更多的字符。
4. 如果选择好了全部字符，则点击“复制”。
这些字符会从“复制字符”栏复制到剪贴板上。
5. 切换回 Word 并按下 Ctrl-V。
字符从剪贴板复制到文本中。

3.3 复合输入

使用许多复合的特殊字符时可以首先输入变音符，然后再输入字母。变音符先不显示出来，随后和字母一起组合成一个特殊字符。

举例

例如：先后按下 (使用德语键盘布局时) 按键 ´ (尖音符) 和 “a” 来输入字符 á (a-尖音符)。

以这种方式进行特殊操作的带变音符的按键在键盘布局描述中用颜色特别标出。

键盘布局

在下面的概述中可以看到用于一些键盘布局各自所支持的变音符：

键盘布局	尖音符	沉音符	抑扬符	扬抑符	短音符	鼻音符	圆点号	双圆点符	软音符	双尖音符	小圆圈	发音符
	´	˘	ˆ	ˇ	˘	ˆ	˙	¨	¸	ˆˆ	◌◌	˘
德语	X	X	X									
法语		X	X					X				X
波兰语	X		X	X	X	X	X	X	X	X	X	
西班牙语	X	X	X					X				X
希腊语	X		X	X	X	X	X	X	X	X	X	
匈牙利语	X		X	X	X	X		X	X	X	X	
美国英语-国际	X	X	X					X				X

设法获得键盘布局的图像显示，以便了解变音符的按键组合。

图像形式的键盘布局

键盘布局的图像您可以在微软公司的网站上获得，网址如下：

<http://www.microsoft.com/globaldev/reference/keyboards.mspix>

如果在按下变音符之后按下空格键，则仅输入该变音符。

注意

切换到“美国英语-国际”键盘布局可能会有帮助，该布局基本上符合美式键盘，并且包含 5 个变音符，有很多特殊字符可供使用。

3.4 使用其它键盘布局

更改键盘布局的必要性

何时需要更改键盘布局？

- 当您必须输入很多文本时。
- 翻译需要使用您的 PC，且习惯使用一种特定语言专用的键盘。

但是如果作为德国人只是想在文件中做一些微小的改动，可能的话，使用德语键盘布局也可以完成，当然只是在少数特殊字符不能用德语键盘来显示的情况下。这些字符可以通过字符映射表来输入。

举例

例如在匈牙利语中必须按照下列方法输入这些字符：

- öŐűŰ (O-双尖音符，U-双尖音符) 通过字符映射表输入
- öÖüÜ 直接输入
- áÁéÉíÍóÓúÚ 复合输入

3.4.1 键盘语言属性方案

Windows 管理一对概念，即“语言（‘输入法区域’）”和“键盘布局”用于键盘。设置的键盘语言决定了赋予输入字符什么样的“语言属性”。对此设置的“键盘布局”决定了每个按键会输入哪个字符。

可以建立几个这样的设置，在几种设置之间可以进行切换。将其中一个设置定义为默认。

Windows 使用概念“输入法区域”，因为除了语言属性，还可以切换货币、日期格式等的设置。

3.4.2 在 Windows XP下安装键盘布局

请按照下列步骤来设置新的键盘布局：

1. 选择开始菜单“设置”→“控制面板”并点击“区域和语言选项”。
2. 选择标签“语言”并在标题“文本服务和输入语言”下点击按钮“详细信息...”。

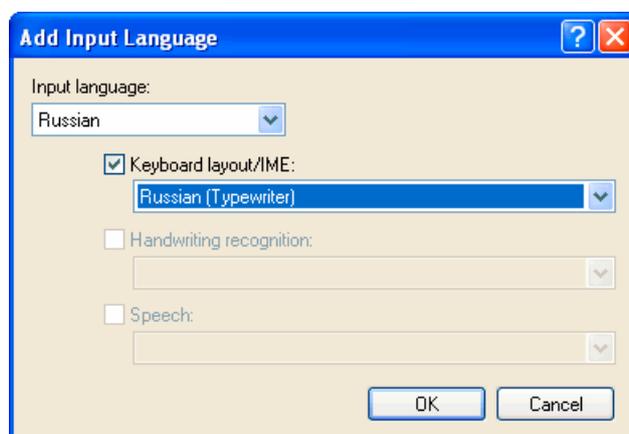
如果在任务栏上显示符号如DE，则也可以单击右键选择指令“设置...”，这样可以省略步骤1和2。

3. 打开“文字服务和输入语言”对话框。选择标签“设置”。



4. 点击“添加”。

打开“添加输入法区域”对话框。



5. 在“添加输入法区域”对话框的“输入法区域”标题下选择一种语言，在“键盘布局/IME:”标题下选择与之匹配的键盘布局。

语言（“输入法区域”）可以和默认语言一致，或者和另一种已经设置为键盘布局的语言一致。但在这种情况下，不能很好的区分两种键盘布局，因为 Windows 通常仅显示语言（“输入法区域”）。

6. 点击“确定”。关闭“添加输入法区域”对话框。返回到“文字服务和输入语言”对话框。

7. 在“键设置...”下选择用何种方式来切换键盘布局。如果选择“无”，则只能用鼠标来切换。
8. 在“语言栏”可以设置当前键盘布局是否显示并且如何显示在任务栏或者某个单独的按键栏中。
9. 点击“确定”，关闭对话框。

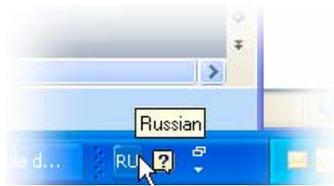
3.4.3 使用

现在可以使用适合于目标语言的“键盘布局”来生成文本了（用于输入外语文本）或者使用到目前为止的标准键盘布局来工作（用于其余操作）。在两种布局之间可以根据需要进行切换。

在任务栏的右边现在应该显示一个蓝色的小图标，用两个字母的缩写表示所设置的语言：

将鼠标移动到该图标上（不用点击），会显示其设置：

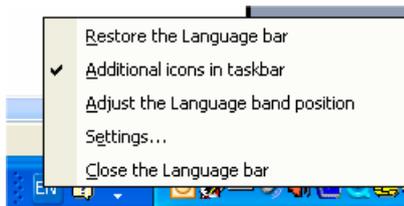
如果所选键盘布局是所选语言的标准布局（如“土耳其语”- "Turkish Q"），则只显示语言（“土耳其语”），否则会显示全部设置、语言和键盘布局，（“土耳其语-Türkisch F”）。



用鼠标左键单击该图标，会显示所设置语言的小菜单。
当前位置通过一个小钩突出显示。



用鼠标右键单击该图标，可以通过菜单项“设置”直接打开“键盘设置”对话框。



在 Windows XP 下提供了其它设置，另外可将该显示作为语言栏独立于任务栏放置，如放置在屏幕上方边缘。

每个输入窗口的键盘布局

Windows 操作系统会记忆每个输入窗口的当前键盘布局。

- 当您转换到另一个窗口时，键盘布局会自动切换。
新窗口总是以标准设置启动。可能的话，必须在启动 Word 后重新选择键盘布局。
- 在选择该窗口的键盘设置前，先将您要输入文本的窗口激活。

菜单操作

同样，以切换过的键盘布局来使用当前有效程序的所有菜单。有可能无法找到按键组合“Alt-字母键”和“Ctrl-字母键”或者甚至不小心触发错误指令。
从此时开始使用鼠标或者在菜单中使用方向键。

注意

如果要经常切换到其它键盘布局，可以在开始菜单“设置”→“控制面板”下的“区域和语言选项”→标签“语言”在对话框“文字服务和输入语言”中设定用于目标语言的特殊键盘布局的新按键组合为默认设置。

请在“默认输入法区域”下选择新的设置。

注意现在所有新窗口、开始菜单和桌面都使用新的键盘布局。

图像形式的键盘布局

设法获取图像形式的键盘布局。

关于互联网地址，参见章节“文本创建文献”

检查键盘布局

比较键盘布局，并记住一个按键在两种不同布局中的使用。如果您不确定，则可以在文本输入时试一下这个键，看看当前有效的布局是哪一种。

举例：按键“1”（字母区）在德语键盘布局时会输入“1”，在捷克语键盘布局时会输入“+”。

注意

检查在 Word 中是否激活了选项“自动键盘切换”。通过 Word 引起键盘布局无意的自动转换会造成混乱。

在章节“多语言 Word 文件”，说明了在哪些情况下可以使用自动键盘切换以及如何激活或取消激活。

3.5 数字文本输入

只在安装了适合的键盘布局时 (参见章节“使用其它键盘布局”) , 才可以使用 Word 进行数字输入。

按住 ALT 键, 在数字键盘上输入零, 然后输入所需特殊字符的三位十进制编码。再次松开 ALT 键。

示例 (土耳其语) :

ALT+0222 会得到 Ş (S 带软音符) 。

仅需要相关代码页的字符映射表 (参见章节“语族”) 。

在章节“通过字符映射表输入”中, 还说明了辅助工具“字符映射表”中的按键组合在哪里显示。

代码页的概述表

可以找到代码页的概述表, 例如在微软的网址 :

<http://www.microsoft.com/globaldev/reference/WinCP.msp>

使用映射表中行标记和列标记的组合 (十六进制 00 到 FF) 并将其换算成十进制数 (0 到 255) 。

单个表格区中包含的 4 位十六进制数是相对应的 Unicode 编码, 可以用来识别字符, 但不能用来进行数字的输入。

3.6 Word 文档中的语言属性

语言属性

语言是一种属性(Attribut), 在 Word 中像其它文本属性 (粗体、下划线, ...) 一样对每个字符进行管理。

- 在语言属性中, Word 会记忆字符、字符串或者单词及完整的句子属于哪种语言。
- 语言属性和文字无关。所以一个西里尔语字符完全可以有德语的语言属性。
- 语言属性是作为主语言和子语言的组合, 例如: “法语 (加拿大)”, 来进行内部管理的。

微软有时将区域名称用于语言属性, 这个概念也包含如货币、日期格式、十进制分隔符等的属性。

语言属性的作用

该属性用于拼写检查和自动切换键盘 (参见章节“多语言 Word 文件”) 。

如果即不要使用正字法检查也不要使用自动键盘切换, 则通常没有必要注意语言属性。但是在这种情况下不要忘记关闭自动键盘切换。

保存语言属性

微软 Word 程序在 Word 文件 (*.doc) 中保存该属性。但保存在文本文件 (*.txt) 时，该属性会和其他文本属性一样丢失。

如果要使用语言属性，则建议使用 Word 文件来编辑文本，并且保存该文本用于以后的更改。Word 文件的编辑完成后，就可按章节“保存文本文件”中所述，将其转换为文本文件。

显示语言属性

在 Word 中，当前语言属性显示在下部窗口边缘的状态栏中。



设置语言属性

按照下列步骤来设置一个文本的语言属性：

1. 选中想要设置语言属性的那部分文本，或者按下 Ctrl+A 选中全部文本。
2. 选择指令“工具”→“语言”→“设置语言...”
3. 在“语言”对话框中选择正确的语言，并点击“确定”。

快速设置语言属性

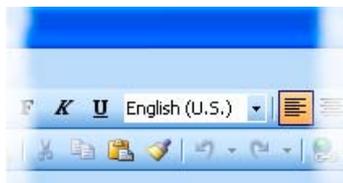
在微软 Word 中，可以在符号栏中插入一个组合的显示区和选择区，在该区总是显示当前语言属性，并且也可以用来设置语言。

按照下列步骤来将该区域插入符号栏：

1. 在菜单“工具”中选择指令“自定义...”



2. 在“自定义”对话框中选择标签“命令”。
3. 在类别列表（左）中选择“工具”。
4. 在命令列表（右）中选择“语言”（参见插图）。
5. 按住鼠标左键将该条目拖到符号栏中。



6. 关闭该对话框。
现在可以使用该选择区来更改文本各个部分的设置语言。

语言属性说明

Word 在下列情况下需要使用语言属性：

- 打开“编码文本文件”类型的文件时（见章节“打开已有文件”）。
- 在用键盘进行输入时，新输入的文本会获得设置的键盘语言。
- 在选项“自动测定语言”激活时，Word 会尝试给输入完毕的句子分配一个语言，并自动设置语言属性。

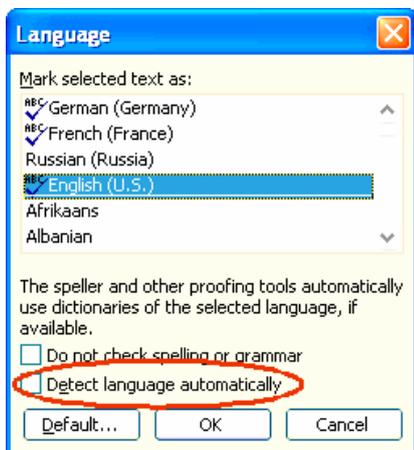
注意

选项“自动测定语言”（特别是对于技术文本）可能会造成混乱的结果。
该选项通常作为预设置激活。

推荐：取消激活选项“自动测定语言”。

按照下列步骤来取消激活选项“自动测定语言”。

1. 确认没有选中任何文本，然后选择指令“工具”→“语言”→“设置语言...”
2. 在“语言”对话框中取消激活选项“自动测定语言”，并点击“确定”。



3.7 多语言 Word 文档

在特定情况下，在一个 Word 文件内刻意地为文本的特定部分设置不同的语言属性是有意义的。

多语言 Word 文件的前提条件

想要用德语键盘布局来输入文本的特定部分，且今后还要用德语键盘布局来编辑该部分文本，而在此期间文本的其他部分要用目标语言的键盘布局来创建和编辑。

带两种语言属性的表格

要创建多列的文本，例如：第一列具有德语语言属性，第二列具有俄语语言属性，可以按照如下方式操作：

- 切换到德语键盘，输入第一行的第一列。然后切换到俄语键盘，输入第二列：
[文本 103][模式 28] "йцукенгшщзхъфывапролджэячсмить"
- 此处 Word 表格不可以有边框和域。也可以使用简单的行，不同的部分之间用空格来分隔。
- 选中整行，用 Ctrl-C 复制到剪贴板上，然后多次按下 Ctrl-V 插入该行。现在您会得到一个两列的表格，左边是德语的，右边是俄语的。

[文本 103][模式 28]	"йцукенгшщзхъфывапролджэячсмить"

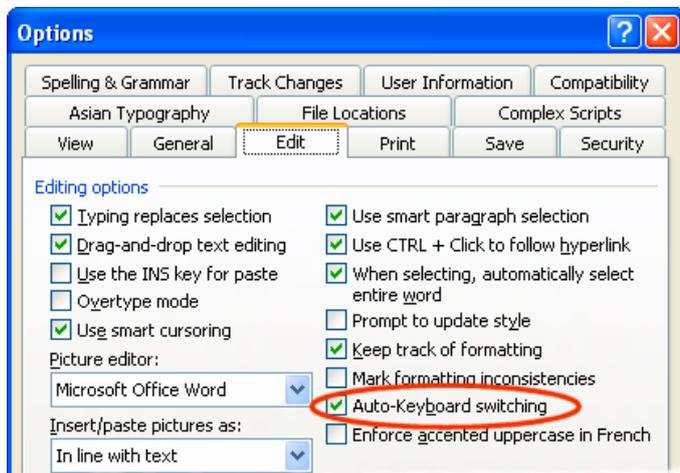
示例

这在下列情况下是必要的：

- 在输入希腊语和西里尔语文本时，相关的键盘布局缺少拉丁语字母和其他 ASCII 字符。
- 使用一些键盘布局时，例如：捷克语键盘，数字不能在正常层面上输入。
- 拉丁语字母的布局和您所习惯使用的布局差别很大（例如：土耳其语-F 键盘布局）。
- 您作为德国人要准备文本的第一部分，而第二部分要让您的翻译来创建。

自动键盘切换

为了在编辑不同文本部分时可以自动获得正确的键盘布局，请在 Word 的“工具” → “选项” 标签“编辑”中激活选项“自动键盘切换”。



注意

无意的自动键盘切换可能会造成混乱。只有当上面提到的边界条件满足时才可以使用该选项！

当前有效的键盘布局始终可见

为了总是能形象的看见哪种键盘布局是当前有效的，可以将任务栏设置为总是可见。

1. 用鼠标右键点击任务栏并选择“属性”。
2. 激活选项“总在最前”
3. 取消激活选项“自动测定语言”

可以将语言栏拖动到屏幕的任意位置。在任务栏中点击键盘切换符号（如 DE）并选择“还原语言栏”。

或者也可以在 Word 中激活键盘切换：

1. 在菜单“工具”中选择指令“自定义”。



2. 在“自定义”对话框中选择标签“命令”。
3. 在类别列表（左）中选择“格式”。
4. 在命令列表（右）中选择“DE 键盘语言”（参见插图）。
5. 按住鼠标左键将该条目拖到符号栏中。

3.8 希腊语和西里尔语文本的特点

注意

如果是第一次用希腊语或者西里尔语文字创建文本，则无论如何要阅读本章节！

希腊语和西里尔语的字母表从语言学的角度被视为独立的字母表，不是拉丁语字母表的扩展。这意味着，希腊语和西里尔语的字母虽然看起来和拉丁语字母一样，但并不是相同的字符，所以也不允许有相同的编码。

有时也意味着，看起来相同的大写字母其对应的各个小写字母看起来不同。

文字	字符	含义
拉丁语	P, p	拉丁语字母 P
希腊语	?	希腊语字母 RHO
西里尔文	?	西里尔语字母 ER
拉丁语	H, h	拉丁语字母 H
希腊语	?	希腊语字母 ETA
西里尔文	?	西里尔语字母 EN

举例

错误的编码可能会导致下列问题：

假设要输入 "NEIN" (德语：否)，您写入了俄语的拉丁语字母 "HET"，而不是西里尔语字母 "HET"。开始看不出什么差别。但是查找小写字母 "het" 时，如果使用查找选项“忽略大小写”，则找不到编码错误的书写方式。

希腊/西里尔语文字的键盘布局

出于这个原因，希腊语和西里尔语文字的键盘布局上根本无法输入拉丁语字母，用 AltGr 按键组合也不可以。更确切地说该键盘布局通常只有两级（正常和上档）。

俄语文字的键盘布局

在俄语的键盘布局中也缺少一些 ASCII 特殊字符，如：# \$ [] { } | ~ &。

所以对于这种语言，尽量使用双语的 Word 文件，如章节“多语言 Word 文档”中所述。

注意

- 只有当的确必要时，才将键盘切换到拉丁语。不要为了单个字母切换键盘，而是用于整句或者整段。

无论如何要避免输入混合编码的句子！

- 请详细了解在您的目标语言中，确切的德语、英语或者国际的缩写（例如：DIN/ISO）在拉丁语文字中是否允许写入，是怎样写入的，或者是否有一个更通行的写法（ДИИ/МСО）。

组合键（如 Ctrl-C、Ctrl-V、Ctrl-Z、Alt-A、等）的键盘布局通常情况下按照美式键盘布局。

3.9 东亚语言文本的特点。

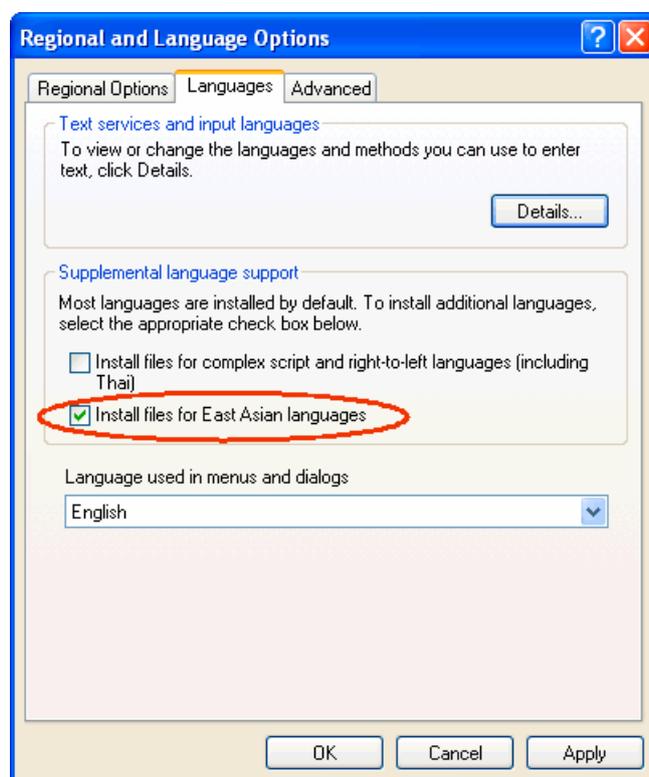
前提条件

如果想以日语、中文或者韩语来编辑或者修正文本，则必须安装相应的操作系统扩展包。

检查系统安装

在开始菜单 → 设置 → 控制面板中打开“区域和语言选项”对话框。

检查在“语言”标签中是否选择了选项“为东亚语言安装文件”。



在 Windows XP 上补充安装东亚语言文件通常不需要安装 CD，因为文件已经以压缩形式保存在硬盘上了。

安装要求有管理员权限。

输入

使用输入方法编辑器 (IME) 来进行输入，这是一种用于在欧式键盘上组合图形字符的辅助工具，如同激活了一个键盘布局。

通常需要一个以目标语言为母语的翻译来操作 IME。

不需要目标语言的 Windows 操作系统。

全角欧式字符

在东亚语言的代码页中包含许多全角的欧式字符（拉丁语字母、数字、以及符号 # \$ % & 等）：

- 单字节（半角）设计。
这里的 ASCII 字符和欧式代码页兼容。
- 双字节（全角）设计。
这种类型的字符是双字节宽度，和 ASCII 字符不兼容。

使用 ASCII 类型字符对于带有功能含义的字符很重要。

正确的示例（SINUMERIK 报警文本，日语）：

014195 0 0 “DコードとG49が同時に指令されています (Ch%1 %2)”

带有功能含义的特殊字符如下所示：

014195 0 0	识别符和属性
空格符	句法上的分隔符
引号	文本限制符
%1, %2	参数占位符

这些字符仅允许使用 ASCII 类型。

注意

在其余的文本中，为了便于阅读完全允许使用全角类型字符（如此处 G49 的示例）。

注意事项

甚至于有空格的全角类型字符（表意的空格）。只有当该字符被标记出来时，才能根据宽度识别该字符。

错误的示例：

014195 0 0 “DコードとG49が同時に指令されています (Ch%1 %2)”

带有功能含义的特殊字符牵涉到不允许使用的全角类型字符。

注意

在给翻译分配任务时指出，对于带有功能含义的字符仅允许使用半角，ASCII 可兼容字符。

简体中文/繁体中文

请仔细区分简体中文和繁体（或者标准）中文。前者在中华人民共和国使用，后者在台湾地区使用。这两种类型的编码不同，因此必须区别对待。注意在分配翻译任务时也要给出明确的说明。

文本创建

在任何情况下，首先创建或让别人创建一个 Word 文档对于日语、中文和韩语文本的创建都是有意义的。

如果使用 Word 2003 作为外部翻译人员的交换格式应该不会有问题。

接下来将这个 Word 文档转换为文本文件，如章节“保存文本文件”中所述。

问题和消除方法

无意的键盘布局切换

问题：

在输入文本时，有时键盘在没有刻意操作的情况下会转换成另一种布局。

消除方法：

检查选项“自动键盘切换”，如章节“多语言 Word 文档”中所述。取消激活该选项。

无意的语言属性更改

问题：

在使用特定的键盘布局（例如：波兰语）创建文本后，确定该文本中的一些部分具有德语或者英语的语言属性。

消除方法：

取消激活选项“自动测定语言”，如章节“Word 文件中的语言属性”中所述。

热键无效

问题：

特定指令（Ctrl+C，Alt+T，等）不再起作用。

消除方法：

由于设置了另一个键盘布局，用于特定指令的键盘组合也改变了。在使用其他键盘布局工作期间，一直使用鼠标。

个别文本无效

问题：

创建完毕的文本文件在 SINUMERIK 运行时间系统中部分无效。
个别文本或者文本行看来没有正确读取。

消除方法：

检查文本的形式标准：

所有具有功能含义的字符编码都正确吗？

- 如果规定了引号，则要检查 Word 是否将输入的直引号（ASCII 字符）转换成了弯引号。在章节“不同语族的特殊字符”中列出了一些 Word 中的干扰选项。
- 如果关系到东亚语言文本，则要检查非亚洲字符的编码，参见章节“东亚语言文本的特点”。如果使用了全角字符作为功能字符，则要用相应的 ASCII 类型字符来替换。

文本中的日元字符 (日语) 或韩元字符 (韩语)

问题：

翻译人员经常会碰到这样的问题，即在使用反斜杠 "\" 的地方用了一个日元字符 ¥ 或韩元字符 ₩。

消除方法：

这可能不是错误。这里有历史的原因。许多年以来在日本一直使用日元字符，在韩国一直使用韩元字符来代替反斜杠字符。

那里的 PC 使用者习惯在文件路径中使用这种显示之后（否则文件路径应该使用反斜杠），该显示就沿用了下来。但是内部认为这是个反斜杠字符。

在字体 "MS Gothic" (日语) 和 "Batang" (韩语) 中反斜杠字符显示为货币字符。

但仍然可以输入日元字符 (Unicode U+00A5) 或韩元字符 (Unicode U+20A9) 来代替反斜杠字符。为了进行检查，执行如下步骤：

1. 在 Word 中将文本格式化为另一种字体，例如："Courier New"。如果看见反斜杠字符按常规正常显示，则没有任何问题。
2. 但如果仍然看见货币字符，或者不是有效的字符，则应修正该字符并用反斜杠来替换。
3. 然后转换回普通字体 (MS Gothic 或 Batang)。

用于生成文本的文献

关于文本创建的更多信息参见下列参考文献：

- 书：*Developing International Software, Second Edition*
微软出版社，2002年10月，ISBN 0-7356-1583-7
http://www.microsoft.com/globaldev/getwr/dis_v2/default.aspx
- 微软的代码页概述表
<http://www.microsoft.com/globaldev/reference/WinCP.aspx>
- 微软键盘布局的图片可在下列网址获取：
<http://www.microsoft.com/globaldev/reference/keyboards.aspx>
- Windows XP 常见问题解答：
<http://www.microsoft.com/globaldev/DrIntl/faqs/winxp.aspx>
- 微软全球软件开发
国际软件各个方面的详细阐述
<http://www.microsoft.com/globaldev>
- Unicode
表格、定义、标准和工具
<http://www.unicode.org>

索引

C

CJK 字符, 8

I

IME (Input Method Editor), 8

T

TrueType 字体, 8

U

Unicode, 9

W

Windows 代码页, 6

Word 文件

多语言, 28

四劃

不同语族的特殊字符, 17

文本文件

打开, 12

保存, 14

新建, 11

转换, 13

文本输入

数字, 25

文件转换, 13

五劃

代码页, 7

名称, 15

序号, 15

六劃

多语言 Word 文档, 28

字符映射表

代码页, 19

输入, 18

输入字符, 19

选择, 18

西里尔语族, 30

七劃

希腊/西里尔语族, 30

系统安装

Windows XP, 32

十七劃

繁体中文, 34

符號

东亚语言文本, 32

变音符

概念, 8

概述, 17

应用, 8

检查安装, 7

简体中文, 34

语言属性

快速设置, 26

带两种属性的表格, 28

设置, 26

语言组, 6

语族, 8

概述, 6

输入方法编辑器 (IME), 8

键盘切换

自动, 29

键盘布局

在 Windows XP 下安装, 22

更改, 21

每个输入窗口, 24
始终可见, 30
图像形式, 21, 25

键盘设置
标准, 9

SIEMENS

SINUMERIK 840Di sI/840D sI/840D

PCU 基础软件(IM8)

开机调试手册

系统开机调试 1

配置系统 2

安装软件和升级软件 3

备份和恢复数据 4

诊断和维修 5

缩略语列表 A

适用于:

PCU 基础软件 版本 8.2

01/2008

6FC5397-0DP10-2RA0

安全技术提示

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
带有警告三角，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。
小心
不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。
注意
表示如果不注意相应的提示，可能会出现不希望的结果或状态。

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

仅允许安装和驱动与本文件相关的附属设备或系统。设备或系统的调试和运行仅允许由**合格的专业人员**进行。本文件安全技术提示中的合格专业人员是指根据安全技术标准具有从事进行设备、系统和电路的运行，接地和标识资格的人员。

按规定使用

请注意下列说明：

 警告
设备仅允许用在目录和技术说明中规定的使用情况下，并且仅允许使用西门子股份有限公司推荐的或指定的其他制造商生产的设备和部件。设备的正常和安全运行必须依赖于恰当的运输，合适的存储、安放和安装以及小心的操作和维护。

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。标签中的其他符号可能是一些其他商标，这是出于保护所有者权利的 目地由第三方使用而特别标示的。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

目录

1	系统开机调试	5
1.1	系统交货状态	5
1.1.1	硬盘分区	7
1.1.2	系统属性	9
1.2	管理用户	11
1.2.1	设定了哪些用户?	11
1.2.2	可以这样创建一个新的用户	12
1.2.3	用户全局设置	14
1.2.4	举例: “operator”用户的设置	17
1.3	PCU 启动属性	18
1.3.1	启动中的维修对话框	18
1.3.2	PCU 启动: 未安装 HMI 程序	20
1.3.3	PCU 启动: HMI 程序已安装	22
1.3.4	设置屏幕的分辨率	23
1.4	BIOS 设置	26
2	配置系统	31
2.1	系统设置	31
2.1.1	可以这样更改 PCU 的名称	31
2.1.2	可以这样设置 PCU 50.3 的 IP 地址	32
2.1.3	可以这样将 PCU 加入一个域中	33
2.1.4	可以这样连接外部显示器	34
2.2	配置用户专用操作界面	35
2.2.1	可以这样选择 Windows 系统的语言	35
2.2.2	HMI 启动画面的保存位置	37
2.2.3	显示自定义启动画面	38
2.2.4	更改维修桌面背景	39
2.3	启动时用户专用设置	40
2.3.1	设计按键过滤	40
2.3.2	保存维修桌面的设置	41
2.3.3	引导启动时启动程序	42
2.3.4	启动 OEM 程序	43
2.3.5	在维修运行中启动应用程序	43
2.4	PCU 50.3 带有 SITOP USV 模块	44
2.4.1	启动并配置 SITOP 显示器	45
2.4.2	SITOP USV 模块配置	47
2.4.3	用于关闭 HMI 的配置	49
3	安装软件和升级软件	51
3.1	安装 SINUMERIK 产品	51
3.2	通过维修桌面安装	51
3.3	可以这样进行 SIMATIC STEP 7 的安装与授权	54

3.4	可以这样在 Windows XP 下安装其他语言 (DVD)	55
4	备份和恢复数据	57
4.1	备份和恢复数据	57
4.1.1	可以这样创建 PCU 的维修系统	57
4.1.2	启动 ServiceCenter Backup-Restore	59
4.1.3	选择维修任务	61
4.1.4	可以这样保存和修复本地分区	62
4.1.5	可以这样保存和修复硬盘	63
4.1.6	从“紧急图像”恢复系统文件	64
4.2	备份 HMI 高级版的运行环境	65
4.3	PG/PC 与设备网络中的 PCU 连接	66
4.3.1	前提条件	66
4.3.2	在设备网络中可以这样连接带有 PCU 的 PC/PG	68
4.4	调试备件硬盘	72
5	诊断和维修	73
5.1	PCU 硬件诊断	73
5.2	运用 7 段显示	74
5.3	引导启动时激活/取消激活故障记录	76
5.4	可以这样在设备网络中查找用户	77
5.5	用于 WinSCP 和 Putty 的 OpenSSH	81
A	缩略语列表	83
A.1	缩写	83
	索引	85

系统开机调试

1.1 系统交货状态

概览

功能强大的 SINUMERIK PCU 50.3 具有用于 Ethernet、MPI 和 PROFIBUS DP 通讯的接口。集成的空闲插口用于其它任务。PCU 50.3 使用 Windows XP ProEmbSys 操作系统，并用软件 Symantec Ghost 来进行数据备份。

接口：

- 四个 USB 接口 (USB 2.0)可以连接键盘、鼠标和其他外设。
- 插 CF 卡的位置被盖起来了。
- 使用 SINUMERIK 840D/840D sl 时：
内部的两个 PCI 接口用于专门的扩展设备。
- 使用 SINUMERIK 840Di sl 时：
一个 PCI 接口已经被 MCI2 板占用，另一个可能被选件 MCI 板扩展件占用。

用于开机调试：

- 集成了两个 7 段显示和两个 LED 用于诊断。它们显示当前的运行状态并在引导启动时输出 BIOS 故障代码。
- 如 PCU 在无操作板工作面时运行，在下列情况下需要另外安装显示器与键盘：
 - 在 PCU 启动时进行诊断
 - 安装备件硬盘
(也可以准备外置硬盘。)

文献： 操作组件与联网手册

交付时 PCU 上所带的软件

PCU 交付时已经安装好的软件包括下列组件等：

MS Windows XP Professional SP2	
Internet Explorer	V 6.0
MPI 驱动器	V 6.03
Symantec Ghost (预设置)	V 8.2 (包括 Ghost Explorer)
TCU 支持 (已经安装并在硬盘路径 D:\Updates 下，如果要重新安装。)	V 8.2

用于全部 Ghost 工具的 PCU 硬盘上的文件将安装到路径 E:\TOOLS 下。

说明

PCU 基础软件中包含的系统组件版本摘自文件 C:\BaseVers.txt。

1.1.1 硬盘分区

硬盘的划分

硬盘有 40G 的存储空间，有一个基本的 C 分区和一个扩展的分区分为 D、E 和 F 三个逻辑驱动器，分别用于 NTFS 文件存取。

HMI 系统软件可能按照订货要求在交付时已经装好了。由客户在以后的首次引导启动时安装。

为了保证数据安全，把 HMI 系统软件和 Windows XP 系统软件分到了不同的硬盘区。

下列映射显示了 PCU 硬盘的使用：

EMERGENCY (C:)

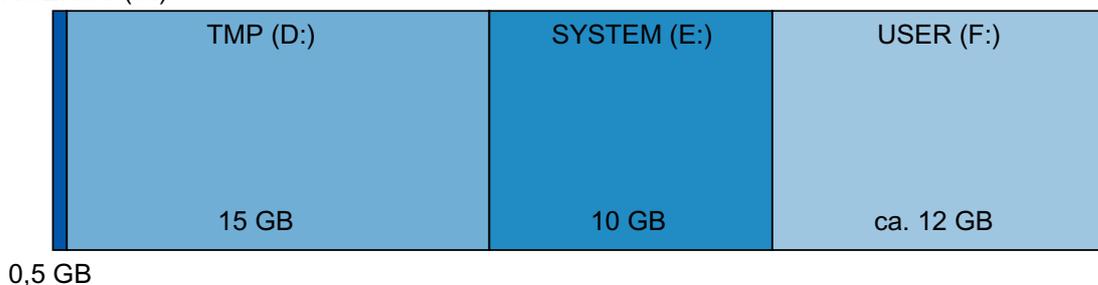


图 1-1 硬盘的划分

分区的内容

各分区规定用于下列文件或者已经被这些文件占用：

EMERGENCY (C:)	预留给 WinPE 2005 下的维修任务。
TMP (D:)	用于保存 GHOST 图像（如交货状态）以及本地备份图像。 驱动器 D 也包含安装目录，待安装的软件先从远程 PG/PC 上复制到该安装目录，再启动真正的安装过程。
SYSTEM (E:)	预留给软件 Windows XP。 Windows XP 软件在恢复盘 CD 上，通过网络驱动器可用于安装驱动程序或者随后安装升级软件。
USER (F:)	用于用户程序的安装。 必须在这里安装应用程序，如：HMI 系统软件（包括数据维护和临时数据）、STEP7、用于 HMI 的 OEM 应用程序或用户应用程序。

注意
所有应用程序必须安装在 USER (F:) 下；即使该应用程序已经预留了另一个驱动盘上的安装路径。
分区名称 EMERGENCY, TMP, SYSTEM, USER 不允许更改；否则“ServiceCenter”不再有效。

参见

安装 SINUMERIK 产品 (页 51)

启动 ServiceCenter Backup-Restore (页 59)

1.1.2 系统属性

操作系统的配置

出于安全的考虑，对 Windows XP 进行了如下预设置：

- 取消自动运行功能。
- 取消 Windows 自动升级。
- 取消防病毒软件的监控和报警以及自动升级。
- 取消从服务桌面或者从开始菜单调用 Internet Explorer 的快捷图标。
- 对于未证实的调用可以进行远程程序调用（RPC）。
- 防火墙设置在网卡 Eth 1 上激活，在 Eth 2 上取消。

在 Windows 通讯（维修）上的更改

其它的预设置：

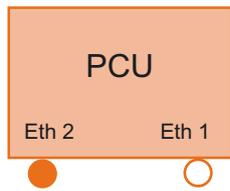
Windows 通讯/维修:	调试类型:	
电脑浏览器	手动	(未启动)
维修故障报告	禁止使能	
便携式媒体序列号	手动	(未启动)
SSDP 发现维修	禁止使能	
通用插拔主机	禁止使能	
网络客户	手动	(未启动)
无线零点配置	手动	(未启动)

PCU 名称

系统交付时，生成一个唯一的计算机名称并可在以下路径读到：“开始”→“设置”→“控制面板”→“系统”，标签“计算机名称”。

PCU 的预设置

PCU 50.3 有两个 Ethernet 接口，并按设备网络与 SINUMERIK solution line 的连接方式进行了合适的预设置：



Eth 1 作为标准 DHCP 客户机，被预设为连接到公司网络上。

Eth 2 作为 SINUMERIK-DHCP 服务器，被预设为连接到设备网络上。在 Eth 2 上预设的固定 IP 地址为 192.168.214.241。

文献： 操作组件与联网手册，章节联网

参见

可以这样设置 PCU 50.3 的 IP 地址 (页 32)

1.2 管理用户

1.2.1 设定了哪些用户？

启动 ServiceCenter Users



使用服务桌面上的快捷图标启动 ServiceCenter Users。

在 ServiceCenter Users 中设置了所有用户通用的或单个用户独立的 PCU 启动特性。

用户管理在 ServiceCenter Users 中执行，使得调试人员/维修技术人员不用在注册表中进行相应设置。

预设用户

每个用户都属于某一种用户类型和用户组。

用户名	密码	用户类型	用户群
operator	operator	HMI	操作员
auduser	SUNRISE	HMI+Service	系统管理员
siemens	*****	---	系统管理员

对于 HMI 用户和 HMI+Service 用户可以单独进行关于启动、HMI 程序和维修的设置。

出厂交付系统时，定义了下列用户：

- 用户 “operator”

“operator” 用户为 HMI 用户类型并属于操作员用户组。在 Windows 中操作员具有限制性用户权限。

- 用户 “auduser”

“auduser” 用户为 HMI+Service 用户类型，并属于系统管理员用户组。在 Windows 中系统管理员具有本地管理员的用户权限。

用户类型

一用户可以为以下的用户类型：

用户类型	任务区域
HMI	<ul style="list-style-type: none"> • PCU 启动 • HMI 程序操作
HMI+Service	<ul style="list-style-type: none"> • 维修任务 • PCU 启动 • HMI 程序操作

在 Windows 下用户类型划分为带有不同用户权限的用户组。

用户组

用户类型划分为以下用户组：

用户群	用户类型	Windows 用户组	用户权限
操作员	HMI	operator.group	限制性
系统管理员	HMI+Service	管理员	本地管理员

服务桌面

维修桌面向 HMI+Service 用户提供了一个 Windows 桌面，其扩展了工具和维修任务功能，例如用户管理，软件安装，数据备份/恢复，检验系统一致性等。

1.2.2 可以这样创建一个新的用户

概览

在 ServiceCenter Users 中作为 HMI+Service 用户可以进行下列任务：

- 创建新用户。
- 删除用户。
- 更改用户名。
- 从域中添加用户。

创建新用户

为了创建一个新的用户：

1. 按下“新用户”按钮。
2. 输入用户名并分配用户类型：例如“HMI + Service”。
3. 然后会要求设定密码。
4. 按下“确定”按钮后，新用户建立完毕并且会显示在“用户”列表中（参见下图）。

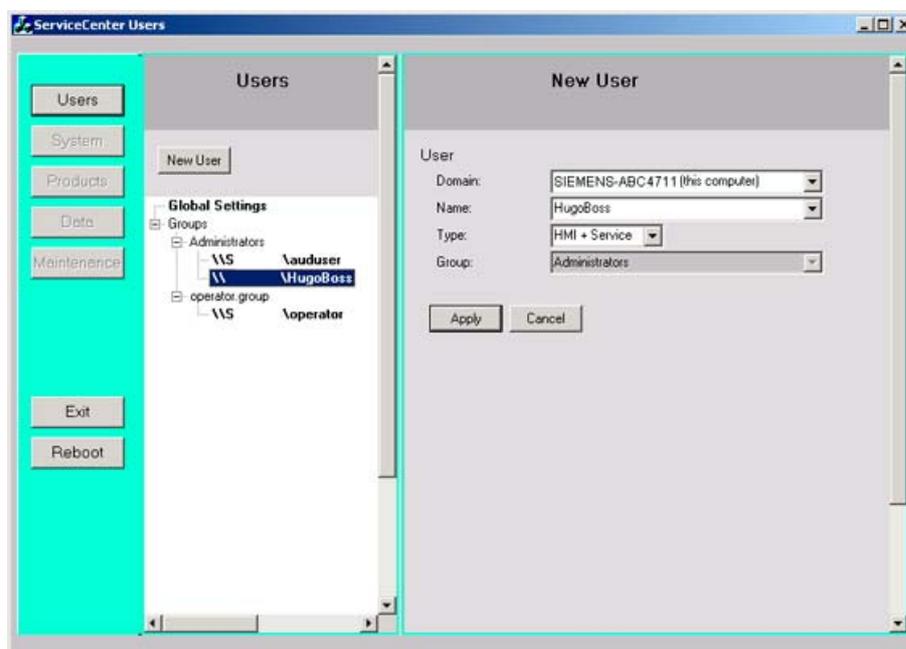


图 1-2 ServiceCenter Users: 新用户

删除用户

如要删除一个用户，则要执行下列步骤：

1. 使用鼠标左键或右键在列表选定需要删除的用户。
2. 在鼠标右键的关联菜单中选择“删除”选项将该用户删除。

结果： 删除用户不会出现安全询问。

更改用户名

1. 使用鼠标左键或右键在列表选定需要更名的用户。
2. 在鼠标右键的关联菜单中选择“更改名称”选项来更改用户名。
3. 输入一个新的名称并用“确定”按钮确认该名称。

从域中添加用户

如果 PCU 成员在一个域中，则可以将该域中已有的用户记录为 HMI 用户或 HMI+Service 用户：

1. 按下“新用户”按钮。
2. 在“域”列表中选择相应的选项并在“名称”列表选取用户，为其在 PCU 上分配用户类型。随后会为其自动分配相应的 Windows 用户组。

参见

可以这样将 PCU 加入一个域中 (页 33)

1.2.3 用户全局设置

全局设置

在“ServiceCenter Users”中的“全局设置”下对以下属性进行设置：

- PCU 启动属性
- HMI 程序开始
- 维修运行中的属性

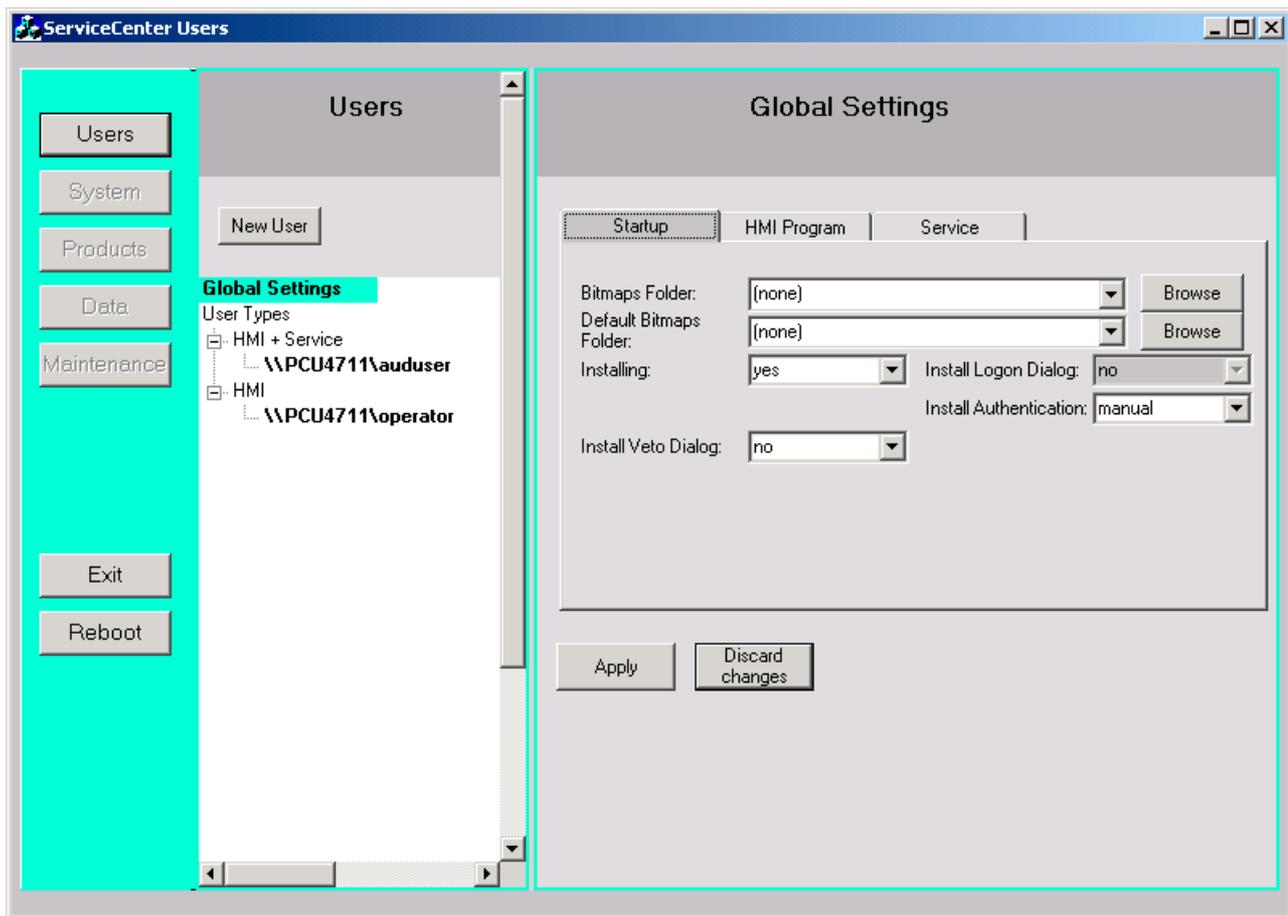


图 1-3 用户的全局设置（安装 HMI 高级版）

预设置以“**粗体**”标记。

表格 1-1 启动

	选择	影响	
位图文件夹:	F:\hmi_adv\ib\DATA\	带有启动画面的目录	
默认位图文件夹:	F:\hmi_adv\ib\DATA\default	带有默认启动画面的目录	
安装:	“是”	启动时对用户软件的安装设有权限。	
	“否”	无安装权限:	
	安装登录对话框:	“是”	显示安装登录对话框。
		“否”	不显示安装登录对话框。
	安装认证:	手动	在安装登录对话框中对 HMI+Service 用户的认证以手动方式进行。
		EKS	在安装登录对话框中对 HMI+Service 用户的认证通过 EKS 进行。一旦背景中通过 EKS 进行的认证失败,就会出现安装登录对话框。
		手动+EKS	在安装登录对话框中对 HMI+Service 用户的认证以手动方式或通过 EKS 进行。
禁止安装对话:	“否”	启动过程中进行安装时: 不询问, 安装过程立即开始。	
	“是”	启动过程中进行安装时: 询问, 是否要进行安装。	

表格 1-2 HMI 程序

	选择	影响
开始:	“是”	将开始 HMI 程序。
	“否”	将不会开始 HMI 程序。
文件:	F:\hmi_adv\rngkml.exe	HMI 程序目录
自动隐藏任务栏:	“否”	HMI 程序:隐藏开始菜单
	“是”	HMI 程序:显示开始菜单
任务栏置顶:	“否”	HMI 程序:开始菜单位于后台
	“是”	HMI 程序: 启动栏始终可见

表格 1-3 维修

	选择	影响
开始对话框:	“是”	显示维修启动对话框
	“否”	不显示维修启动对话框
登录对话框:	“是”	显示维修登录对话框
	“否”	不显示维修登录对话框
认证:	手动	使用用户名和密码进行认证（手动）
	EKS	使用 EKS 进行认证
	手动+EKS	认证时要么使用用户名和密码要么使用 EKS
默认密码映射:	“是”	密码可以用大写或小写字母输入。
	“否”	密码必须严格按设定输入。
位图文件夹:	(无)	维修桌面背景画面目录
默认位图文件夹:	(无)	维修桌面默认背景画面目录
自动隐藏任务栏:	“否”	维修桌面: 隐藏开始菜单
	“是”	维修桌面: 显示开始菜单
任务栏置顶:	“否”	维修桌面: 开始菜单位于后台
	“是”	维修桌面: 启动栏始终可见

继承原则

在“全局设置”下涉及到用户的设置，单个用户会加以继承。每位用户可以按自己的需要对被继承的设置进行专门的匹配设置。

1.2.4 举例：“operator”用户的设置

用户“operator”

“全局设置”下的各项设置会对所有用户生效：

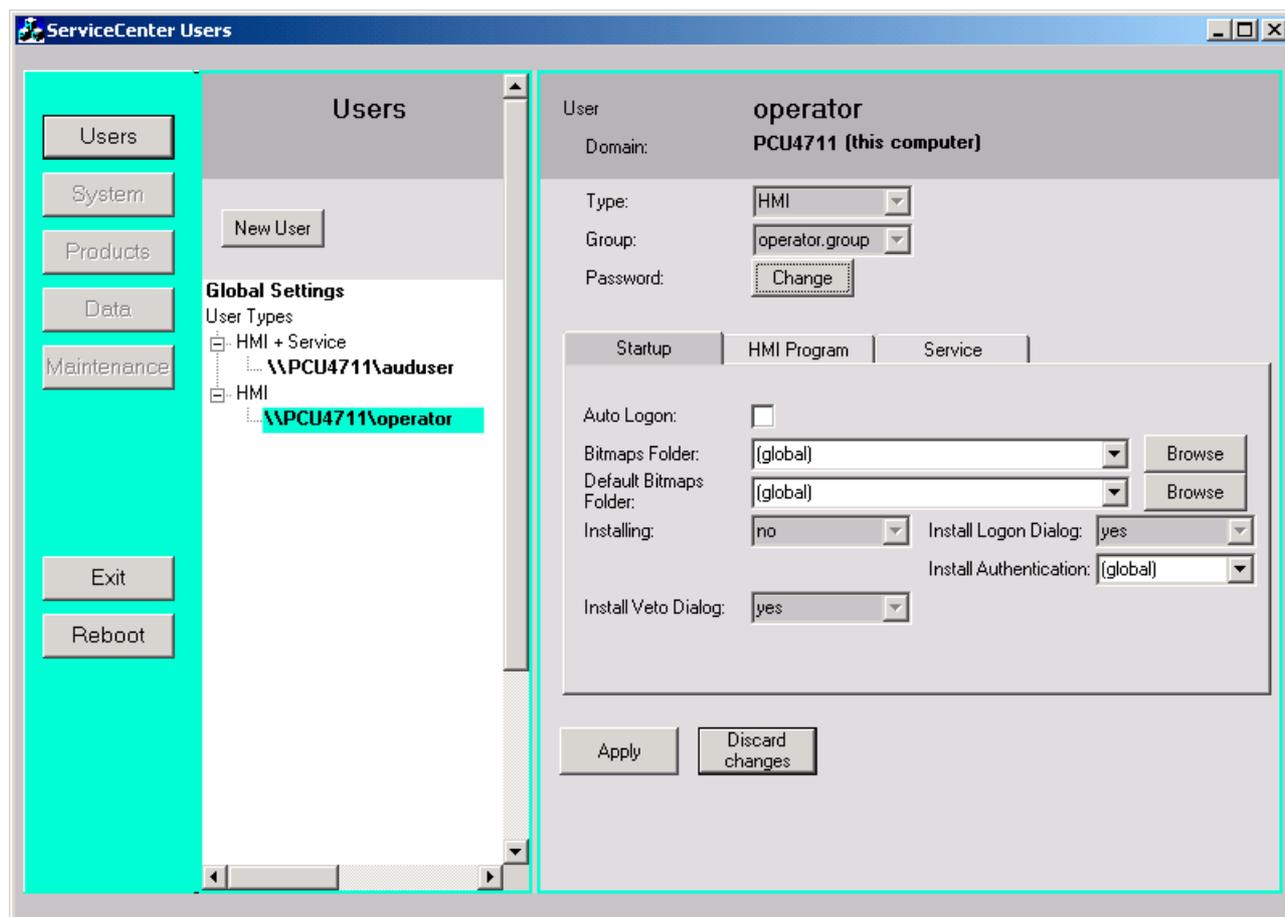


图 1-4 “operator”用户的设置

使用鼠标左键在“用户”列表选定用户。输入密码后，可以根据需要对设置进行匹配。不可修改的设置以灰色背景显示。

说明

当按下按钮“重新启动”时，会立即执行重新启动（无事先询问）。使用“退出”键退出 ServiceCenter Users。

1.3 PCU 启动属性

1.3.1 启动中的维修对话框

启动时的属性

启动时会显示以下对话框：

- 如果未安装 HMI 程序或者在启动时取消启动 HMI 程序，则会显示 **维修启动对话框**。
通过改变全局或用户设置可以抑制该对话框的显示。这时可以选择预设置“维修桌面”。



图 1-5 不带 HMI 程序的维修启动对话框

- 如果安装了 HMI 程序，则在**维修启动对话框**中会另外出现按钮“启动 HMI”。
- 如果设置为“手动”方式的认证，则按下按钮“维修桌面”或“用户维修中心”后在维修启动对话框中会显示 **维修登录对话框**：



图 1-6 维修登录对话框（带有认证：“手动”）

此外，当认证系统（EKS）没有提供有效的用户数据时，也会显示该维修登录对话框。

注意

通过 EKS 设置存取等级：

- 如果 EKS 单元有效，即钥匙已插入并可以使用，则 EKS 单元的钥匙信息独自决定有效的存取等级。EKS 有效时禁止在 HMI 程序中更改存取等级。
- 如将钥匙从 EKS 单元中拔出，即 EKS 单元无效时，则系统将使用钥匙开关的当前存取等级。
- 在操作站中更换操作权限时，将以新有效操作站的 EKS 单元为标准。如果那里没有分配 EKS 单元，则与 EKS 单元失效时的结果相同。
- 由 HMI 对钥匙信息进行计算。在由钥匙信息计算出存取等级之后，HMI 会接受该存取等级，并且还会将该存取等级用在 NCK 中：这将作为系统最后使用的存取等级。

1.3.2 PCU 启动：未安装 HMI 程序

前提条件

启动 PCU 时未安装 HMI 程序。

启动阶段

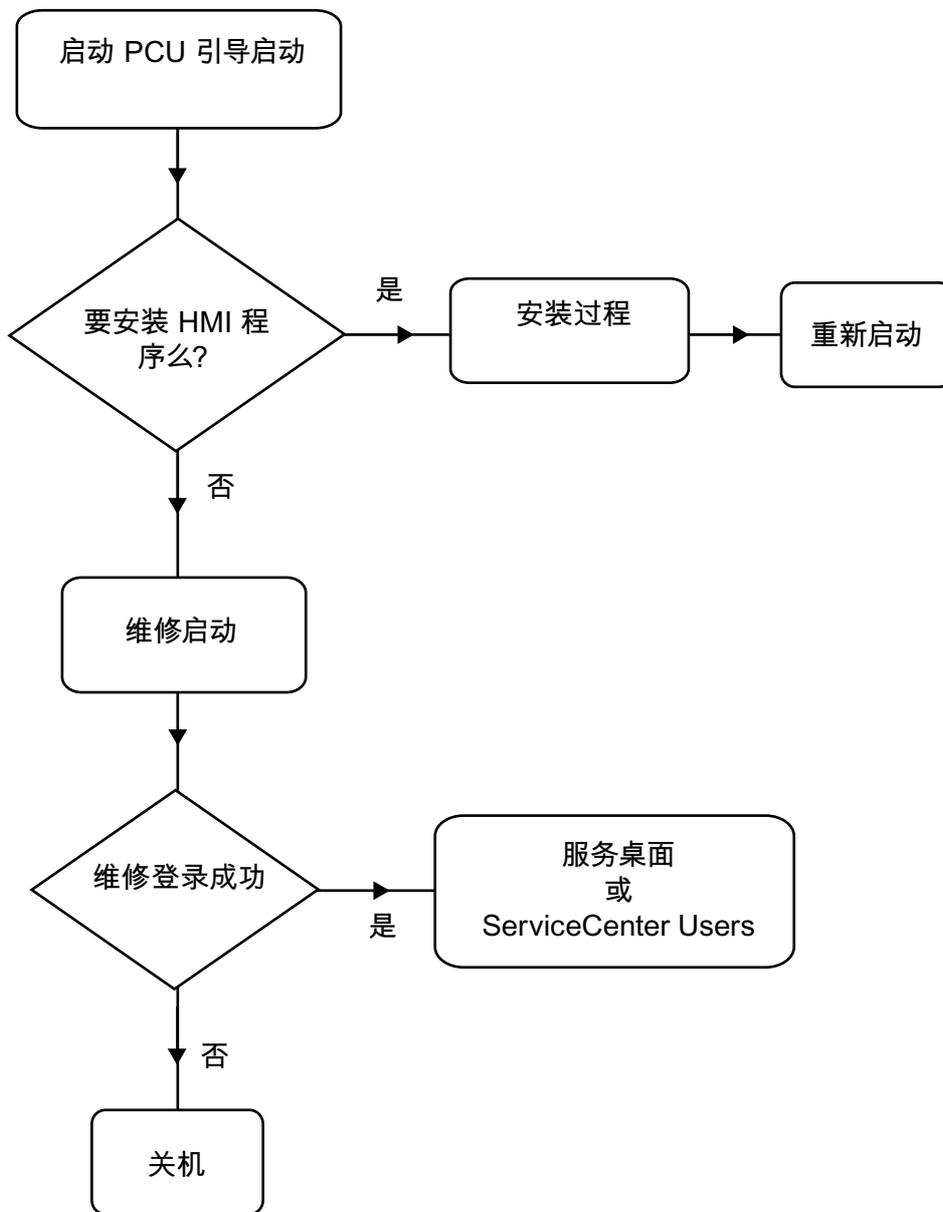


图 1-7 PCU 启动流程图（无 HMI 程序）

说明:

- 如果在目录 D:\Install 下存在待安装的程序，则会在首次启动时询问，是否要启动安装过程。成功安装后必须重新启动。
安装也可以跳过，稍后再执行。
- 在维修启动对话框中可以选择“维修桌面”、“用户维修中心”或“关机”。
- 选择“维修桌面”或“用户维修中心”时打开维修登录对话框。

说明

在首次启动 PCU 50.3 时用户只能以‘auduser’登录。

1.3.3 PCU 启动：HMI 程序已安装

前提条件

已经安装了 HMI 程序。

启动阶段

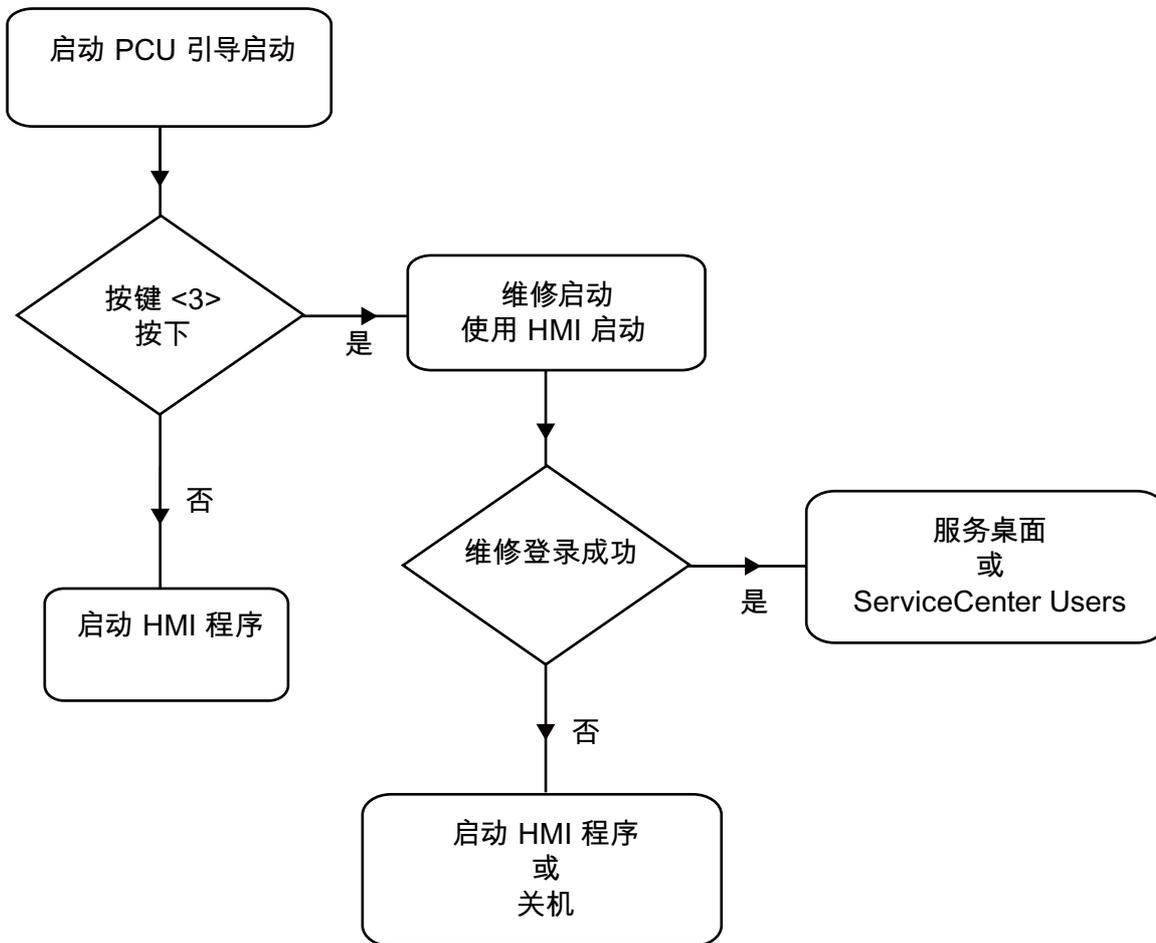


图 1-8 PCU 启动流程图（已安装 HMI 程序）

说明：

- 如果已经安装了 HMI 程序，PCU 启动并且 HMI 程序开始运行（预设置）。
- 启动期间有一个时间间隔，用于当背景画面中出现版本信息时按下右下方的按键 <3>：然后会打开维修启动对话框。
- 要进行维修任务就需要作为维修用户登录。为此提供有下列选项：

- 使用用户名和密码在域中手动登录。
- 使用 EKS 登录：为此必须提供用于认证的密钥和有效的用户数据。如果不能通过 EKS 获得有效的用户数据，则会显示带有用户名和密码的维修登录对话框。
- 两种方式都可以选择。

1.3.4 设置屏幕的分辨率

概览

引导启动时与分辨率相关的系统属性可以在文件 `tcu.ini` 中进行设置。

`tcu.ini` 的供货状态位于 `E:\siemens\system\etc` 中。修改后的 `tcu.ini` 被保存在 `F:\addon_base\...`，`F:\oem_base\...`，`F:\user_base\...` 当中。

设置 PCU 启动时的分辨率

在段 [RESOLUTION] 中的 `tcu.ini` 文件里有列方式可供选择：

- 0 = SYSTEM
- 1 = AUTO_OP_1 （预设置）
- 2 = AUTO_OP_2
- 3 = AUTO_MON_1
- 4 = AUTO_MON_2
- 5 = 640X480
- 6 = 800X600
- 7 = 1024X768
- 8 = 1280X1024

设置的含义如下：

设置	含义
SYSTEM	如在启动时未对分辨率进行专门设置，即系统中进行上一次会议时所使用的分辨率有效，例如手动在系统控制器中进行设置。
AUTO_OP_1	预设置： 在启动时会自动（使用“PCU 面板”优先权）按照下列情况对分辨率进行设置：
情况 1:	存在 PCU 面板（与可能存在的 PCU 显示器和 TCU 面板无关） [在 TCU 运行被激活时： 关系到下列附加条件：PCU 有焦点（在动态分辨率转换打开时）]： 分辨率被设置成 PCU 面板的最大分辨率（最大 1280x1024）。

设置	含义
情况 2:	<p>不存在 PCU 面板，但存在 PCU 显示器（与存在的 TCU 面板无关）： [在 TCU 运行被激活时： 关系到下列附加条件：PCU 有焦点（在动态分辨率转换打开时）]： 如在未对分辨率进行专门设置，即系统中最后一次所使用的分辨率有效，例如手动在系统控制器中进行设置。 （与 AUTO_OP_2 的区别!）</p>
AUTO_OP_2	<p>和 AUTO_OP_1 相同，除了： 情况 2：不存在 PCU 面板，但存在 PCU 显示器（与存在的 TCU 面板无关）： [在 TCU 运行被激活时： 关系到下列附加条件：PCU 有焦点（在动态分辨率转换打开时）]： 分辨率被设置成 PCU 显示器的最大分辨率，减小为下一个较小的 SINUMERIK 分辨率。SINUMERIK 分辨率指分辨率 640x480，800x600，1024x768 和 1280x1024。 举例： 在最大分辨率为 1440x900 的 PCU 显示器上 SINUMERIK 分辨率设置为 1280x1024。</p>
AUTO_MON_1	<p>在启动时会自动（使用“PCU 显示器”优先权）按照下列情况对分辨率进行设置：</p>
情况 1:	<p>存在 PCU 显示器（与可能存在的 PCU 面板和 TCU 面板无关） [在 TCU 运行被激活时： 关系到下列附加条件：PCU 有焦点（在动态分辨率转换打开时）]： 分辨率被设置成 PCU 显示器的最大分辨率，减小为下一个较小的 SINUMERIK 分辨率。SINUMERIK 分辨率指分辨率 640x480，800x600，1024x768 和 1280x1024。 举例： 在最大分辨率为 1440x900 的 PCU 显示器上 SINUMERIK 分辨率设置为 1280x1024。 当存在 PCU 面板时，如果 PCU 面板的最大分辨率小于 PCU 显示器的最大分辨率，将会通过随动摄影进行显示。</p>
情况 2:	<p>不存在 PCU 显示器，但存在 PCU 面板（与存在的 TCU 面板无关）： [在 TCU 运行被激活时： 关系到下列附加条件：PCU 有焦点（在动态分辨率转换打开时）]： 如在未对分辨率进行专门设置，即系统中最后一次所使用的分辨率有效，例如手动在系统控制器中进行设置。 （与 AUTO_MON_2 的区别!）</p>

设置	含义
情况 3:	<p>不存在 PCU 显示器和 PCU 面板 (= 无头运行):</p> <p>A) 在 TCU 运行被激活时:</p> <p>a) 动态分辨率转换已打开 (TCU.ini 中的分辨率适配记录) 并且最少已经登录了一个 TCU: 分辨率被设置成当前聚焦 TCU 的分辨率。</p> <p>b) 动态分辨率转换已打开 (TCU.ini 中的分辨率适配记录) 但还没有登录 TCU, 或者动态分辨率转换关闭: 分辨率被设置成当前 TCU 面板和最后一次会议中所登录 TCU 面板的最大分辨率。</p> <p>预设置: 按照注册表对 TCU 分辨率的进行预设置。</p> <p>注意: (随后) 将要登陆的第一个 TCU 面板获得焦点。焦点处理器自动将分辨率设置成 TCU 面板的分辨率 (在动态分辨率转换时)。</p> <p>B) 在 TCU 运行未被激活时: 如在未对分辨率进行专门设置, 即系统中最后一次会议时所使用的分辨率有效, 例如手动在系统控制器中进行设置。</p>
AUTO_MON_2	<p>和 AUTO_MON_1 相同, 除了:</p> <p>情况 2: 不存在 PCU 显示器, 但存在 PCU 面板 (与存在的 TCU 面板无关):</p> <p>[在 TCU 运行被激活时: 关系到下列附加条件: PCU 有焦点 (在动态分辨率转换打开时)]: 分辨率被设置成 PCU 面板的最大分辨率 (最大 1280x1024)。</p>
640X480	启动时 SINUMERIK 分辨率设置为 640x480。
800X600	启动时 SINUMERIK 分辨率设置为 800x600。
1024X768	启动时 SINUMERIK 分辨率设置为 1024x768。
1280X1024	启动时 SINUMERIK 分辨率设置为 1280x1024。

1.4 BIOS 设置

概述

PCU 50.3 的 BIOS 预设置不必更改。日期和时间可以在 Windows 或者 HMI 操作面板下设置。

注意

设备配置用于随附软件的使用。仅当在设备上已经进行了技术更改或者接通时出现故障时，才更改设置的值。
--

启动 BIOS-Setup

1. 按如下方式启动 BIOS-Setup:
 - 复位设备（热启动或冷启动）。
 - 引导启动测试后出现如下信息：
按 < F2 > 进入 SETUP 或<ESC> 显示引导启动菜单
2. 出现 BIOS 信息，按下按键 F2。
 - 打开 BIOS 主菜单：

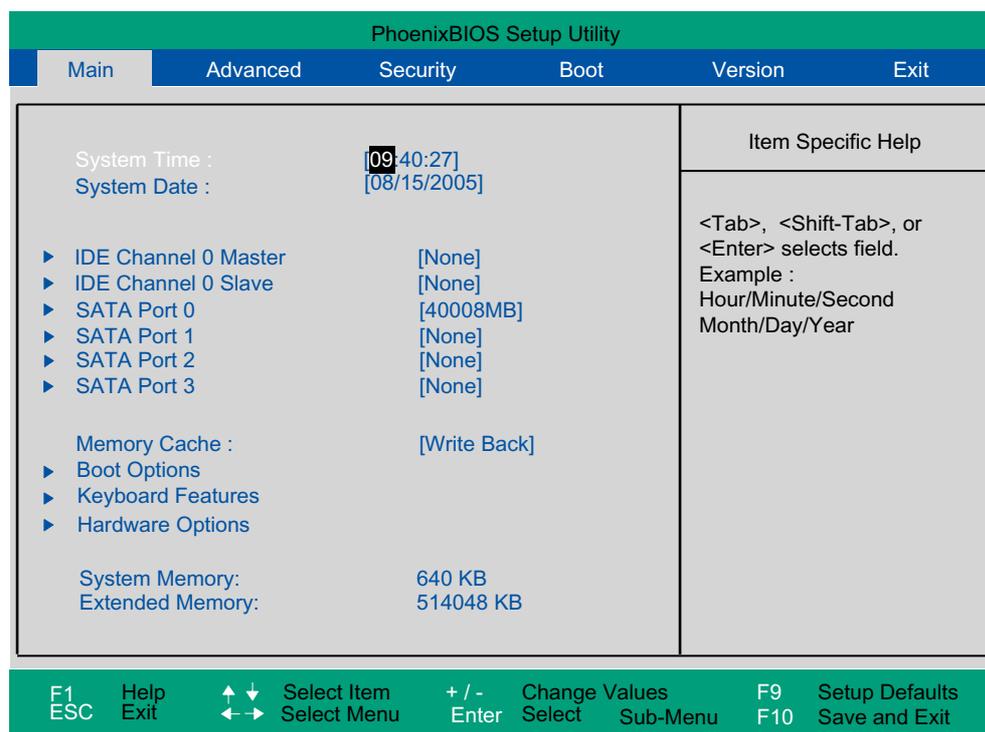


图 1-9 BIOS 主菜单 (举例)

BIOS-Setup: 标准设置

交付时须保存了下列系统参数:

菜单: 主机

系统参数	标准设置	专用登入
系统时间	小时:分:秒	
系统日期	月/日/年	
IDE 通道 0 主机	无	
IDE 通道 0 从机	无	
SATA 端口 0	40008 MB	
SATA 端口 1	无	
SATA 端口 2	无	
SATA 端口 3	无	
高速缓冲存储器	回写	

引导启动选项		
快速引导启动模式	已使能	
SETUP 快速	已使能	
POST 错误	全部, 但非键盘	

引导启动选项		
汇总屏幕	已使能	
诊断屏幕	已使能	
POST 代码/状态	LPC Bus 总线	

键盘功能		
NumLock	开	
键点击	禁止使能	
键盘自动重复率	30/秒钟	
键盘自动重复延迟	½ sec	

硬件选项		
PCI - MPI / DP	已使能	
机载 Ethernet 1	已使能	
机载 Ethernet 1 地址	08 00 06 90 xx xx	
机载 Ethernet 1 远程引导	已使能	
机载 Ethernet 2	已使能	
机载 Ethernet 2 地址	08 00 06 90 xx xx	
机载 Ethernet 2 远程引导	禁止使能	
安全卡功能	已使能	
扇区控制	已使能	
CRT/LCD 选择	同时 自动	

菜单: 高级配置

系统参数	标准设置	专用登入
安装的 O/S	其它	
复位配置数据	否	
Legacy USB 支持	禁止使能	
USB 控制器重启	已使能	
I/O 装置配置		
内部 COM 1	已使能	
基本 I/O 地址	3F8	
中断	IRQ 4	

PCI 配置		
PCI 设备槽 1		
选项 ROM 扫描	已使能	
使能主盘	已使能	
等待定时器	默认	
PCI 设备槽 2		
选项 ROM 扫描	已使能	
使能主盘	已使能	
等待定时器	默认	

SATA/PATA 配置		
PATA 控制器	已使能	
SATA 控制器模式	增强版	
AHCI 配置	禁止使能	
RAID 支持	禁止使能	

菜单: 安全

系统参数	标准设置	专用登入
主管口令为	禁止使能	
用户口令为	禁止使能	
设置用户口令	回车	
设置主管口令	回车	
口令引导中	禁止使能	
固定盘引导区	标准	

菜单: 引导

系统参数	标准设置	专用登入
启动优先顺序:		
1: SATA0:	Fujitsu MHT2040BHTBD	
2: PCI BEV:	VIA BootAgent	
3:		
4:		
5:		
6:		
7:		
8:		
从引导启动中排除或者:		

菜单: 版本

系统参数	标准设置	
SIMATIC PC	SINUMERIK PCU50.3	
BIOS 版本	V05.01.06	
BIOS 编号	A5E00370214-ES005	
MPI/DP 固件	V01	
CPU 类型	Celeron ® M processor 1.50GHz	
CPU ID	06D8	
代码修订	0020	

菜单: 退出

保存更改&退出	保存所有更改并随后用新参数执行系统重启。
---------	----------------------

更改 BIOS 设置

在装入或者添加额外的组件后，必要的话，必须将这一情况告知在 BIOS 启动中的系统：

1. 启动设备。
2. 按照激活 BIOS 启动的要求按下 <F2> 键（适合操作面板上的水平软键 2）。
3. 出现 BIOS-Setup 菜单。用光标键在菜单中导航至所需的选择区域。
4. 使用 <+>（同时按下 <SHIFT> 和 <X>）或使用位于数字区的按键 ↔ 来更改设置。
5. 可以用向右/向左光标键达到其它设置菜单。
6. 按下 <ESC>（<取消报警> 键），以到达菜单“退出”（也可以通过继续按向右光标键来实现）。
7. 按下 <输入> 键，离开设置菜单。

随后启动系统。

说明

更改 BIOS 设置时 - 引导顺序除外 - 需要签订 OEM 合同。

配置系统

2.1 系统设置

2.1.1 可以这样更改 PCU 的名称

预设置

PCU 会被赋予一个自动生成的计算机名称。

步骤

为了更改 PCU 的名称，需要：

1. 选择“开始”→“控制面板”→“系统”。
2. 选择选项卡“计算机名称”并点击“更改”。
会打开下列对话框：

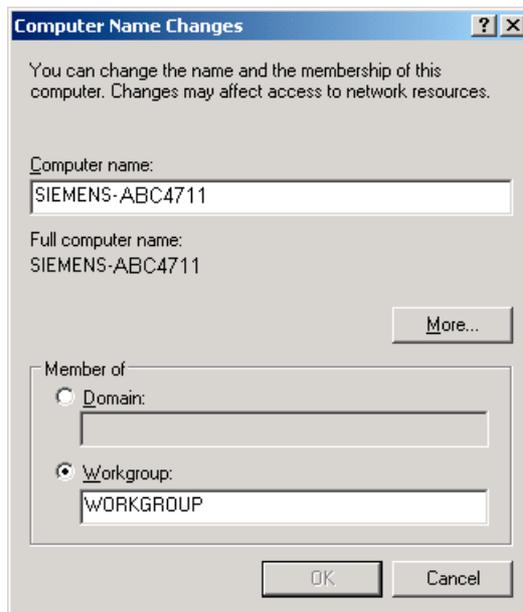


图 2-1 更改 PCU 名称

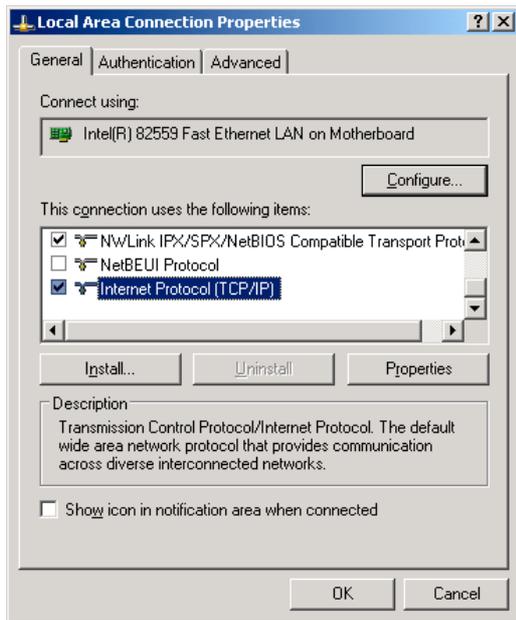
2.1.2 可以这样设置 PCU 50.3 的 IP 地址

预设置

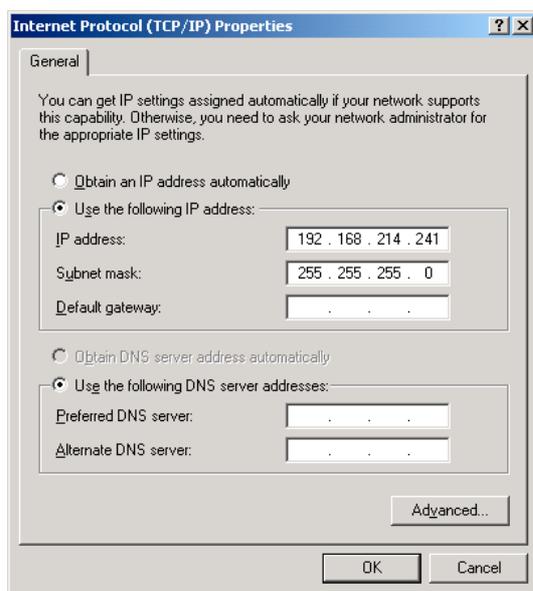
注意
设备网中的每个 PCU 50.3，其 IP 地址在出厂时已经被预设为 192.168.214.241。
只有当您想要改变此预设值时，才需执行下述步骤！

步骤

1. 在 PCU 的维修桌面上进行如下选择：“开始”→“设置”→“网络连接”。
打开“网络连接”的窗口。
2. 请您双击将要设置参数的接口 **Ethernet 2**，将使用它连接 TCU 或者与设备网相连。
打开“Ethernet 2（系统网络）属性”窗口。
3. 请您在“常规”选项卡上选择“Internet 协议 (TCP/IP)”，并点击“属性”按钮。
打开“Internet 协议 (TCP/IP)”窗口：



4. 请您在“常规”选项卡上选择选项“使用下面的 IP 地址”，并输入 IP 地址和子网掩码。
用于第一个 PCU 的推荐设置：



5. 请您输入想要的新的 IP 地址并点击“确定”对设置进行确认。

2.1.3 可以这样将 PCU 加入一个域中

前提条件

只有具有相应权限的用户，比如域管理员，才能将一个 PCU 加入到一个已有的域中。

域控制器 (DC)

域控制器 (DC) 是在网络中对计算机和用户进行中央认证与授权的服务器。在带有域控制器的网络中，多台计算机被归入一个域。

步骤

为了将 PCU 加入一个域中：

1. 选择：“开始” → “设置” → “控制面板” → “系统”，标签“计算机名称”。
预设置：PCU 属于“工作组”并且未分配域。
2. 按下“更改...”按钮
会打开下列对话框：

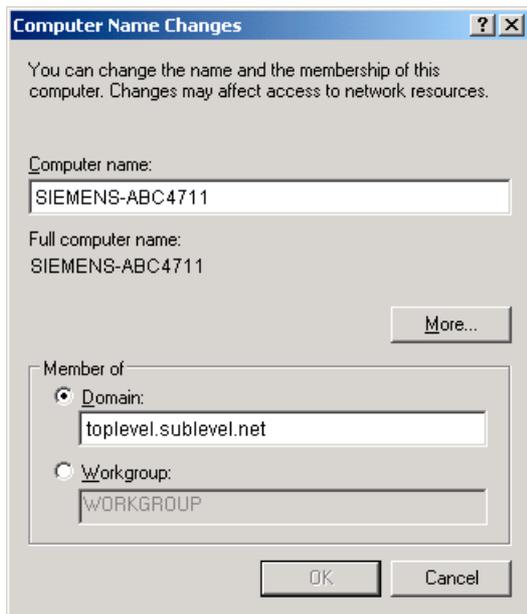


图 2-2 将 PCU 加入域中

3. 给定 PCU 要加入域的域名。
4. 然后会被要求，作为具有相应权限的用户登录来结束设置过程。

2.1.4 可以这样连接外部显示器

前提条件

要连接外部显示器，需要满足下列前提条件：

- 外部显示连接在 PCU 的 DVI 接口上（有时使用适配器）。
- 正在运行时不允许连接显示器。

步骤

要连接外部显示器：

1. 在 PCU 的维修桌面上用鼠标右键在关联菜单中选择菜单项“属性”。
2. 选择选项卡“设置”并点击按钮“高级”。
3. 选择选项卡“疑难解答”并设置“硬件加速”不等于零。建议使用设置“完全”。
4. 点击“确定”进行确认并关闭所有对话框。
5. 重复步骤 2 和 3：另外有一个选项卡“Intel(R) ... 图形控制器”
6. 按下“图形属性”按钮。在选项卡“驱动程序”上的外部显示器与“显示器”的选择一致；PCU 上 OP/TP 与“笔记本”的选择一致。
7. 选择“主要驱动程序”与“次要驱动程序”。
8. 点击“确定”进行确认并关闭所有对话框：现在就可以使用外部显示器了。

2.2 配置用户专用操作界面

2.2.1 可以这样选择 Windows 系统的语言

预设置

交货时，在 PCU 上安装的操作系统 Windows XP 只有英语并使用美国键盘布局。

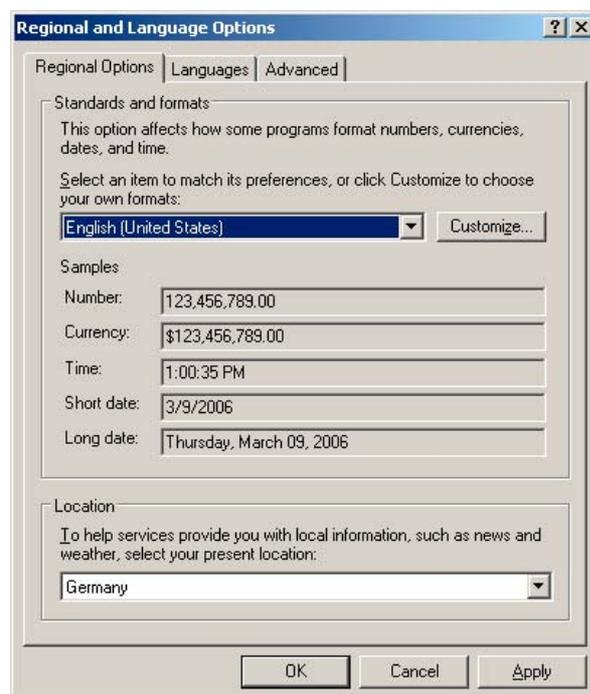
前提条件

为切换语言，必须从 DVD “SINUMERIK 维修包恢复盘 Win XP ProEmbSys SP2” 上安装所需的语言。用“多语言用户接口”（MUI）可以把 Windows 系统的菜单、对话框和键盘布局切换为另一种语言。

选择语言

从相应的 CD 安装语言后，操作步骤如下：

1. 选择“开始”→“控制面板”→“语言和区域设置”，打开以下对话框：



2. 选择标签“语言”，来切换用于 Windows XP 操作界面的语言。在“菜单和对话框中使用的语言”中选择新的语言并按“确定”来确认。



3. 在标签“高级”上选择不支持 Unicode 程序的语言。

结果

为了使语言切换生效，需要重启 PCU。显示可选的语言在每种语言的字符组中进行。

注意 不允许更改标签“区域设置”上设置的键盘布局以及日期、时间和数字表示的格式。这些设置会根据 HMI 高级版所选的操作界面语言自动匹配。

参见

可以这样在 Windows XP 下安装其他语言 (DVD) (页 55)

2.2.2 HMI 启动画面的保存位置

预设置

西门子启动画面在各自的 PCU 上，如：对于控制系统 SINUMERIK 840D，保存在如下路径：

F:\hmi_adv\ib\DATA\10700\0\

可以为制造商专用的启动画面创建目录树，制造商画面可以按照下列框架保存在其中：

制造商专用启动画面目录树：

F:\oem\ib\DATA\

<Nck 类型>	适用于：
0	840 D
10700	840D sl
15000	840Di sl

<分辨率>: 640、800、1024 或者 1280 dpi

如果总是要显示一个固定画面（与 NCU 无关的画面），可以在缺省目录下按要求的分辨率保存该画面。如果对于不同的 NCU 要显示不同的画面，在子目录<Nck 类型>\<分辨率>下按相应的分辨率保存这些画面。

画面名称和分辨率

<名称>.bmp:可以自由选择名称；每个目录下只允许有一个文件。必须使用图形工具以子目录名称给出的分辨率创建画面，并保存在相应的目录中。HMI 软件根据 NCK 类型和现有操作设备分辨率选出画面。

2.2.3 显示自定义启动画面

目录

建立一个带有多个启动画面（对于不同的面板分辨率）的目录。为此，该目录划分为子目录 640、800、1024 和 1280，在其中包含了相应分辨率的各引导启动画面。

此外，还可以建立带有缺省启动画面的目录，同样有子目录 640、800、1024 和 1280。如果在上述引导启动画面目录中没有找到启动画面（较低分辨率的也未找到），则显示子目录中保存的特定分辨率的画面。

如果没有发现启动画面（较低分辨率的也未找到），则显示通用的启动画面，它是 PCU 基础软件的组成部分。

用户专用设置

目录可以在 ServiceCenter Users 中设置为用户专用的：

- 启动：位图文件夹
- 启动：默认位图文件夹

2.2.4 更改维修桌面背景

概述

维修桌面的背景样式或者背景画面不是通过“控制面板”（系统控制）设置，而是在注册表中设置：

- 背景样式：

密码：	HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\<version>\HMIDesktop
值：	图案 (STRING)
日期：	<位代码>" (例如 "0 80 114 32 0 5 39 2", 参见 HKCU\Control Panel\Patterns) “(无)” (= 无背景样式)
初始日期：	迄今为止通过系统控制设置的背景样式
缺省日期：	“(无)”(当不存在注册信息或者不可读时)

- 背景画面：

维修桌面的背景画面不是通过“控制面板”（系统控制）设置，而是在 **ServiceCenter Users** 中通过“维修: 位图文件夹”或“维修: 默认位图文件夹”进行用户自定义设置。

说明

尝试通过系统控制来设置维修桌面背景 - 如同在标准 Windows 中 - 仅影响启动画面的显示，然而不影响维修桌面背景的显示。

2.3 启动时用户专用设置

概览

引导启动后立即启动 HMI 程序，如 HMI 高级，在设置和功能上与引导启动后立即启动维修桌面不同。

2.3.1 设计按键过滤

过滤键

引导启动 HMI 程序时，过滤键顺序和同时按下的功能键。在文件 E:\Windows\System.ini 中可以设计待过滤的键顺序和同时按下的功能键。

过滤键顺序：

段：	MMC103Keyb
密码：	SeqAct
值：	<位掩码> (= 待过滤的键顺序，根据 E:\Windows\System.ini 中的注释)
初值：	262143

过滤同时按下的功能键：

段：	MMC103Keyb
密码：	ConcurrentKeyMask
值：	<位掩码> (= 待过滤的功能键，根据 E:\Windows\System.ini 中的注释)
初值：	255

2.3.2 保存维修桌面的设置

启动维修桌面

通过维修桌面可以自由进入 Windows 平台。启动维修桌面时将运行所有程序，在登录时已经自动以默认运行方式启动 Windows。

- 执行 HMI 程序

也可以从维修桌面启动 HMI 程序。

- 结束 HMI 程序

结束一个从维修桌面启动的 HMI 程序时，将重新回到维修桌面。

保存维修桌面（预设置）

注销时，不保存维修桌面上的设置（例如：维修桌面上的链接命令）。维修人员要在维修桌面上始终找到同一个输出状态，而不是前一次会议的设置。

保存维修桌面设置

可以通过注册表中的某一注册项来更改这种属性。通过注册表中的该项保存下列设置：

- 打开窗口的位置
- 任务栏的大小和位置
- 移动和删除链接

密码:	HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\ <version>\HMIManager
值:	SINDesktopSettings (DWORD)
日期:	1 (保存设置) 或者 0 (不保存设置)
初始日期:	- 值不由基础软件创建 -
缺省日期:	0 (如果注册信息不存在或者不可读)

该密码对所有维修用户和所有其它用户都有效。

与注册信息无关，始终保存维修桌面上的链接。

说明

在退出维修桌面前，在“保存设置”的设置下必须关闭注销前的仍然打开的应用程序窗口。否则重新启动时，在 HMI 程序开始前，上述应用程序窗口会直接短时出现并再次关闭。

2.3.3 引导启动时启动程序

启动附加程序

这些程序可以与 HMI 程序并行启动，维修桌面打开时 Windows 自动运行。可以设定与 HMI 程序并行启动。

如果待运行的程序在 Windows 目录 E:\Documents and Settings 下，则要设置以下注册表信息：

密码：	HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\ <version>\HMIManager
值：	StartSINHMIStartupDirsPrograms (DWORD)
日期：	1（程序被启动）或者 0（程序不被启动）
初始日期：	0
缺省日期	0（如果注册信息不存在或者不可读）

若待启动的程序设置了注册信息 'HKCU\Software\Microsoft\Windows\CurrentVersion\Run' 和 'HKLM\Software\Microsoft\Windows\CurrentVersion\Run'，则要设置以下注册信息：

密码：	HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\ <version>\HMIManager
值：	StartSINHMIRunPrograms (DWORD)
日期：	1（程序被启动）或者 0（程序不被启动）
初始日期：	0
缺省日期：	0（如果注册信息不存在或者不可读）

执行 HMI 程序

在 HMI 程序执行过程中，运行一个不可见的 Windows Explorer，通过其设置（禁用的浏览器功能、任务栏、开始菜单等等）阻止在 Windows 平台上不希望的访问。

任务栏属性可以在 ServiceCenter Users 中通过菜单“HMI 程序”用户自定义设置“自动隐藏任务栏”和“任务栏置顶”。

结束 HMI 程序

结束 HMI 程序时，Windows XP 完全关闭。

2.3.4 启动 OEM 程序

概述

直接在 HMI 系统软件启动前，可以启动 OEM 程序。为此，这些程序或其链接必须保存在目录 C:\RunOEM 的子目录中。

开始时的顺序

以列出的顺序来执行子目录。一个子目录中的程序按照它在子目录中的创建时间顺序启动。

- 在子目录中 C:\RunOEM\SeqOnce 程序**一次性并按顺序**启动，即：当前一启动程序结束后才启动后一程序。
- 在子目录中 C:\RunOEM\Seq 程序**在每次引导启动时，按顺序**启动，即：当前一启动程序结束后才启动后一程序。
- 在子目录 C:\RunOEM\ParOnce 中程序**一次性同时**启动。其与 HMI 系统软件并行运行。
- 在子目录 C:\RunOEM\Par 中程序**在每次引导启动时同时**启动。其与 HMI 系统软件并行运行。

除程序文件外，也可以在子目录中保存其它类型的文件，按相应的文件类型打开。

例如：".txt" 型文件用记事本打开，".htm"类型文件用 Internet Explorer 打开。

2.3.5 在维修运行中启动应用程序

启动其他应用程序

如果要在维修运行时启动其他应用程序，则要在文件 WINBOM.INI 的段落 [OEMRun] 中输入带有完整路径的程序。

举例：启动程序“记事本”

```
[OEMRunOnce]
```

```
"启动 WinVnc", "x:\I386\system32\StartWinVnc.exe"
```

```
"验证密码", "x:\I386\system32\CheckPEPwd.exe"
```

```
[OEMRun]
```

```
"启动 Backup/Restore", "x:\I386\system32\GhostOrder.exe"
```

```
"记事本", "e:\windows\notepad.exe"
```

不得更改所有其它的输入。

2.4 PCU 50.3 带有 SITOP USV 模块

前提条件

- PCU 基础软件 Windows XP，版本高于 V08.00.00 用于 USB 接口
- HMI 高级，版本高于 V07.01.00
- 安装了版本高于 2.5.2.4 的 SITOP-SW:
SITOP-SW 可以用于下载：www.ad.siemens.de/sitop
- 安装了 SITOP 监视器/配置程序：

为此必须把 SITOP 软件复制到 PCU 上准备好的目录 E:\SITOP。在该文件夹下已经有 PCU 的工具，用于 SITOP USV 的关闭。如果在 PCU 基础软件的旧版本中没有该目录，则必须创建一个，以便以后升级 PCU 基础软件时可以一致升级。

- 为 Windows XP 安装了 USV-USB 驱动器：
在所附的 SITOP 文献中对安装作了描述。文献位于 SITOP 软件的下载包中。
- 连接了 SITOP-USV 的硬件。

应用

当 PCU 上的电源出现扰动时，SITOP USV 模块可以通过使用缓冲蓄电池在有限时间内维持运行，而且当电池耗尽时使 PCU 有序停止。

名称	订货号 (MLFB)
SITOP POWER DC-USV 模块 15 (USB 接口)	6EP1931-2EC41

说明

机床制造商测试环境的信息

在 HMI 高级版的标准配置中测试“SITOP USV”功能。安装附加软件组件和 OEM 软件组件时，必须由用户对整个系统的关闭进行测试。

2.4.1 启动并配置 SITOP 显示器

Windows 启动

在启动 Windows 时必须自动启动 SITOP 显示器，因此对于在 Windows 注册数据库中的 SITOP 显示器必须在下列密码下输入一个新值：

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

值名称：SITOP

值类型：REG_SZ

值数据：E:\SITOP\SITOP_DC_USV.exe

在文件夹 E:\SITOP 中有一个脚本文件 sitop.reg。通过执行该文件，将在注册数据库中自动输入所需的密码。

重新启动 PCU 后，自动启动 SITOP 显示器。安装的下一步是配置显示器。

注意

必须避免通过 Windows 自动启动目录来启动 SITOP 显示器。

一般设置

在 SITOP 显示器的配置对话框中必须执行下列设置：

- 设置接口参数：对于带有 USB 接口的模块 USB
- 设置更改行动的参数：
必须取消监控窗口的显示，因为该功能会导致 HMI 操作面板偶尔出现故障。

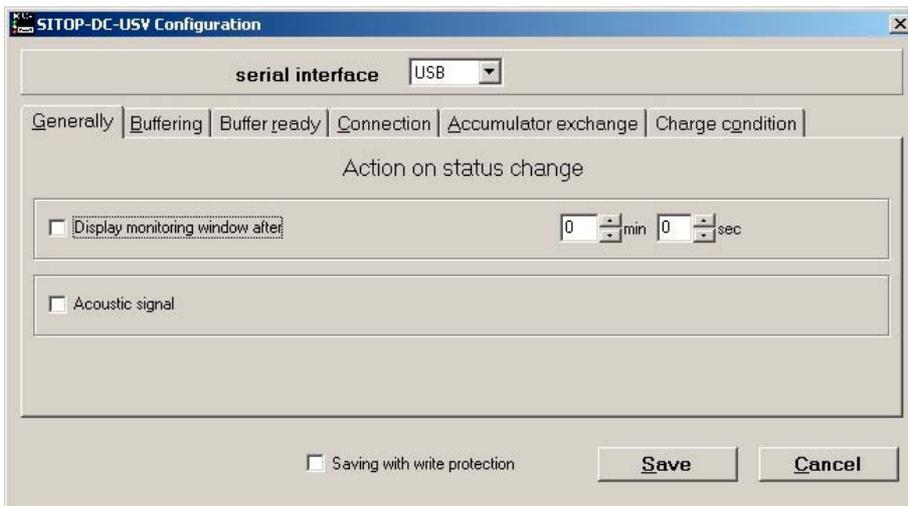


图 2-3 配置 SITOP 显示器的参数：一般设置

配置 SITOP 显示器的参数：

在参数区域内，为缓冲存储器输入用于断电时 HMI 高级版有序停止和 PCU 关闭的程序路径：
E:\SITOP\shutdown.bat

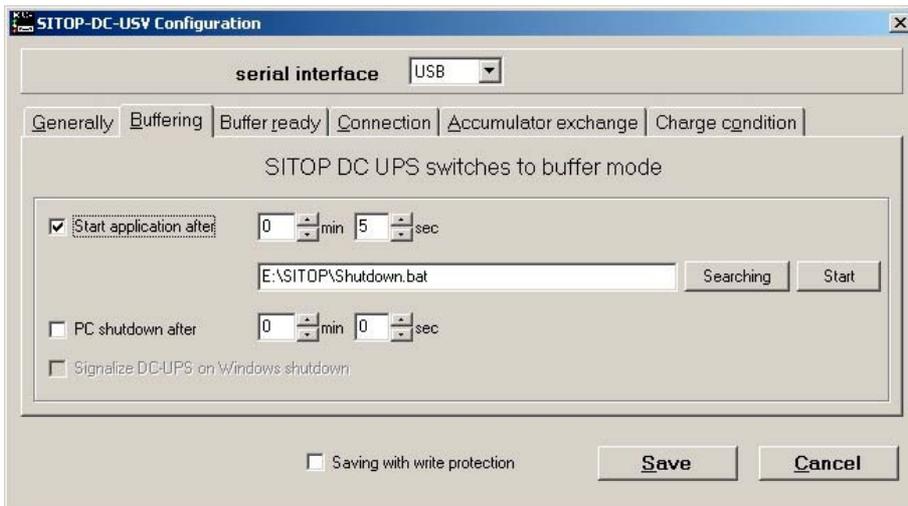


图 2-4 配置 SITOP 显示器的参数：设置缓冲状态下的运行

也可以使用以下设置： E :Windows\system32\hmiexit.exe

2.4.2 SITOP USV 模块配置

为缓冲状态下的运行设置参数

基本上可以在 USV 模块上选择，是否在规定的时间内或者在达到蓄电池深度放电域值 (=最大缓冲时间) 时，结束缓冲运行。由此得出两个用于缓冲运行的参数设置。

运行方式“最大缓冲时间”(高于 PCU 基础软件 XP 08.00.00)

在该运行方式下，可以适时对系统进行暂时关闭。对此，USV 模块与操作系统的关闭同步。一直保持缓冲运行，直至操作系统关闭。操作系统的关闭最多允许 5 分钟（包括所有应用程序）。另外，USV 模块以最大缓冲时间（取决于蓄电池状态）缓冲。

在 USV 模块上的必要设置 (USB 接口)

	开 - 关			
1		●	+2V	接通阈值 (cut-in threshold) +22V 固定设置的 (fixed)
2		●	+1V	
3	●		+0.5V	
4		●	+1V	结束充电电压(end-of-charge voltage) + 26.3V 固定设置的 (fixed)
5		●	+1V	
6	●		+0.5V	
7	●		+0.2V	
8		●	+0.2V	
9		●	+0.1V	
10		●	0.35A / 0.7A	充电电流(charging current)

	开 - 关			
1		●		设定时间/最大时间(set time / max.)
2		●	+320s	缓冲时间 +5s 固定设置 (fixed)
3		●	+160s	
4		●	+80s	
5		●	+40s	
6		●	+20s	
7		●	+10s	
8	●			
9		●		运行状态 蓄电池 开/关(operating state battery)

图例说明:



供货状态设置

在 PCU 50 上运行所需的设置

运行方式 “固定缓冲时间”

在该运行方式下，USV 模块总是以预选的固定时段缓冲 USV 模块。USV 模块与操作系统关闭无法同步。

在 USV 模块上的必要设置

	开 - 关			
1		●	+2V	接通阈值 (cut-in threshold) +22V 固定设置的 (fixed)
2		●	+1V	
3	●		+0.5V	
4		●	+1V	° ° 结束充电电压(end-of-charge voltage) + 26.3V 固定设置的 (fixed)
5		●	+1V	
6	●		+0.5V	
7	●		+0.2V	
8		●	+0.2V	
9		●	+0.1V	
10		●	0.35A / 0.7A	

	开 - 关			
1	●			设定时间/最大时间(set time / max.)
2		●	+320s	° ° 缓冲时间 +5s 固定设置 (fixed)
3	●		+160s	
4		●	+80s	
5		●	+40s	
6		●	+20s	
7		●	+10s	
8	●			
9		●		运行状态 蓄电池 开/关(operating state battery)

图例说明:



供货状态设置

在 PCU 50 上运行所需的设置

2.4.3 用于关闭 HMI 的配置

监控 HMI

通过专用应用程序 `hmiexit.exe` 对 HMI 高级版的结束进行监控。该应用程序通过内含的批处理文件 `shutdown.bat` 启动。发生故障时，该应用程序强制关闭操作系统。如果 HMI 不在设定的等待时间内结束，就出现故障。

可能在下列文件中为 HMI 退出随机设置下列参数：`E:\SITOP\hmiexit.ini`。

[Actions]

#等待时间以秒表示，用于结束 HMI 高级版应用程序

Wait = 120

#等待时间超时的措施

ForceShutdown = True

仅当在一个 OEM 安装程序中的 HMI 应用程序结束时间超过 120 秒，才要求更改预设置。一般不对该设置进行更改。

退出键的设计

应当在生产操作中切断通过操作范围菜单 EXIT 键的 HMI 的关闭功能，因为该功能与 USV 模块不同步。

在文件 `Regie.ini` 中用注册信息 `EixtButton=FALSE` 关闭退出键。

睡眠模式

运行 USV 时，操作系统的睡眠模式禁用，因为 USV 模块的 USB 接口必须总是激活。

说明

进一步提示参见相应订货号的产品说明。

安装软件和升级软件

3.1 安装 SINUMERIK 产品

概述

该章节说明了在预先安装了 PCU 基础软件的基础上安装补充软件或如何进行升级。

下面从硬件组件和软件组件的供货状态开始进行说明。

维修桌面可用于下列任务，例如：

- 安装 HMI 系统软件
- 设置 HMI 系统软件运行环境
- 检查硬盘或版本
- SIMATIC STEP 7 授权

3.2 通过维修桌面安装

安装附加软件

维修桌面也提供安装系统软件或升级软件功能。这主要涉及通过 Windows 网络安装的的安裝/升级包。

可以两种方式进行安装：

- 安装/升级包位于目录 D:\INSTALL 下。PCU 下次引导启动时，自动启动安装/升级过程。只有在安装或升级过程结束后，才继续进行正常的系统引导启动，必要时启动 HMI 软件。
- 可以从维修桌面直接通过执行安装/升级包来安装或升级。

使用安装目录

自动执行用户安装时，可以设置多个安装目录。安装目录是子目录 D:\Install 和在参数文件 E:\Windows\System32\HMIServe.ini 的 [SetupDirs] 分区中列出的目录。

那里预设置了安装目录 D:\Install

一个分区内的密码名在参数文件 E:\Windows\System32\HMIServe.ini 中必须是明确的。

使用安装目录时按照参数文件中所描述的顺序进行。如果没有参数文件或者文件没有 [SetupDirs] 分区，则考虑用预设置的安装目录 'D:\Install' 来代替。

如果分区 [SetupDirs] 包含有安装目录，而下面不是 'D:\Install'，则对于设置的安装目录不考虑 'D:\Install'。

用 OpFile 执行安装时必须要注意，是否安装结束后需要重启并通过 OpFile.txt 中的相应注册信息来显示重启。如果有相应的注册信息，则释放重启。对于一系列先后执行的安装，要等最后一个安装结束后进行重启。

用 HMI-Explorer 来安装

在维修桌面上提供程序“HMI-Explorer”。调用该程序时，可以得到关于 HMI 系统软件安装包和 Windows XP 的详细版本信息。从 HMI-Explorer 可以单独启动或卸载应用程序。

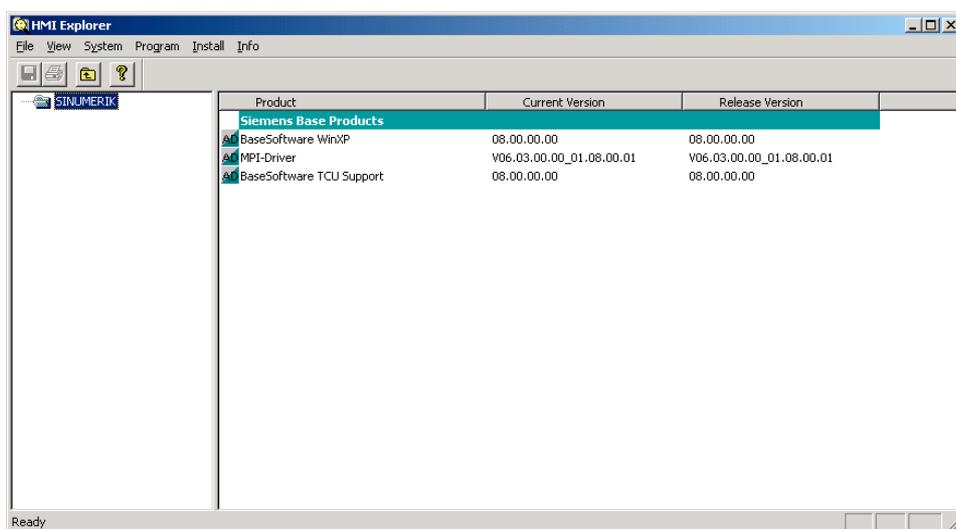


图 3-1 HMI-Explorer (举例)

HMI-Explorer 的说明

属性对话框给出了安装软件产品的详细信息：

- **SINUMERIK 产品信息：**

“信息”对话框告知所选 SINUMERIK 产品的信息：

- 当前版本：给出安装的 SINUMERIK 产品是哪个版本。版本以完整格式输出。
- 内部版本：显示该产品的当前内部版本号。
- 安装日期/时间：显示当前版本的安装日期和时间。
- 安装路径：表示用于 SINUMERIK 产品的主目录路径。
- 启动应用程序：给出 *.exe 文件的路径信息，用于启动 SINUMERIK 产品。

- **产品语言**

对话框“语言”列出每个 SINUMERIK 产品的安装语言，并给出关于安装语言的名称信息。如果该语言未知，则输出名称简略记号。HMI-Explorer 的已知语言是德语、英语、西班牙语、法语和意大利语。此外还输出安装语言的版本。另外还给出了安装时间和日期。

- **产品历史**

在“历史”对话框中，给出了 SINUMERIK 产品的历史。在该对话框中，给出了关于释放版本、可能的维修包和 HotFix 补丁信息。总是有“释放”项。按照是否安装，依情况来显示两个项“维修包”和“Hotfix”。对于该列表中的每一项每次显示信息“版本”、“内部版本”和“安装日期/时间”。

- 组件信息

“组件”对话框显示产品随附的组件信息：

组件：	组件名称
版本：	组件内部版本
路径：	组件路径
文件：	*.exe 文件
使能：	显示组件是否激活。
说明：	组件说明
类型：	组件类型

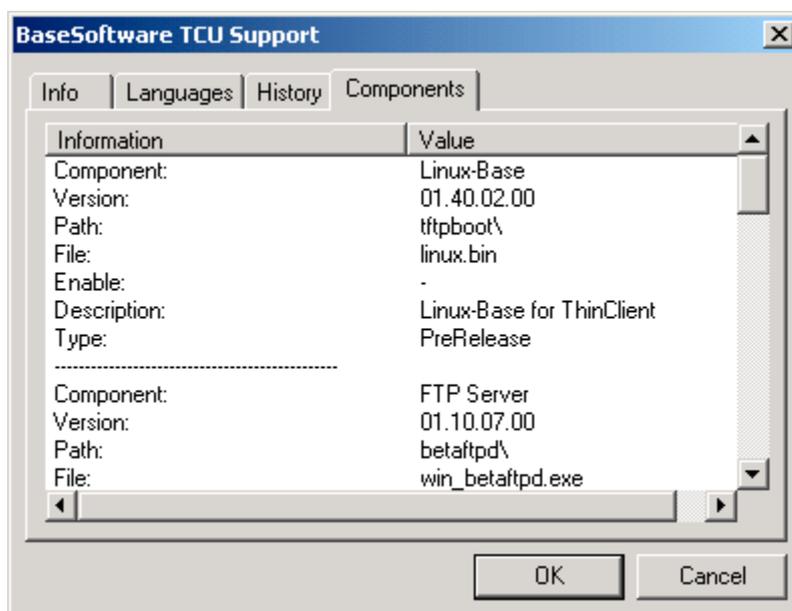


图 3-2 “组件”对话框 (举例)

3.3 可以这样进行 SIMATIC STEP 7 的安装与授权

概述

SIMATIC STEP 7 V5.4 SP1 同样可以安装在 PCU 上。

交付物品:	SIMATIC STEP 7 V5.4 SP1
组成部分:	SIMATIC STEP 7 V5.4 SP1 和 AddOn 用于 SINUMERIK 840Di sl/840D sl/840D
交付形式:	1 张安装 DVD
系统前提条件:	高于 V8.0 的 PCU 基础软件 要求网络连接或者连接外部 DVD 驱动器。 推荐: 鼠标连接

说明

PCU 上现有的“MPI 驱动器”包属于 HMI 软件，不允许卸载！

在 Windows XP 下安装

必须遵守这里所述的安装顺序！

1. 在维修桌面上启动 PCU。
2. 建立从 DVD 的存取（通过网络或者外部 DVD 驱动器）并从那里在启动目录 SETUP.EXE 中调用。
3. 由操作者执行安装。STEP 7 安装目录应当在 F:\... 更改，因此 F: 上的目录可任意选择。
4. 询问“传输许可证密码”时，必须回答“否，以后传输许可证密码”。安装 SINUMERIK-AddOn 后随即进行许可证验证。安装结束后需要重新启动 PCU。执行启动时重新选择维修桌面。
5. 切换到 DVD 的目录 Sinumerik_Add_On 并在那里调用 SETUP.EXE。由操作者执行安装。安装结束后需要重新启动 PCU。执行启动时重新选择维修桌面。
6. 在维修桌面上启动“STEP7 授权”的快捷图标。由此授权 STEP7，现在可以从 HMI 高级的操作界面来启动（STEP 7 显示为操作菜单扩展栏上的专用操作区域，存取等级 3）。

在 F:\Add_on\loemframe.ini 中自动输入：

[s7tgtopx]

；使用 HMI 高级版:取消最小化/最大化按钮

；属于 Step7 窗口

WindowState_Off=196608

；使用 HMI 高级版:当 Step7 结束时切换到前一任务

nSwitchToTaskAfterTermination= -2

该输入项在 OEM 配置中必须修改。

3.4 可以这样在 Windows XP 下安装其他语言 (DVD)

使用

SINUMERIK 维修包恢复盘 WIN XP ProEmbSys SP2 应用于:

- 事后安装 Windows 组件
- 恢复无 HMI 高级版的 PCU 交付状态
- 安装其他 Windows XP 语言

DVD 内容

DVD 上有以下目录:

目录	目录
1_WIN_Components	Windows XP ProEmbSys SP2 操作系统 Windows XP ProEmbSys 包括 SP2 用来补充安装 PCU 中所没有的软件组件。
2_XP_Base	Symantec Ghost 图像, 用于 PCU 50.3 和 EBOOT <ul style="list-style-type: none"> • PCU 基础软件 (用于 PCU 50.3 的 Windows XP 不带 HMI 高级版和其他应用软件) 的供货状态的 Ghost 图像。 • “紧急启动系统” (与 PCU 上的目录 D:\EBOOT 一致) 设置的 Ghost 图像。
3_MUI_1	中文 (简体) 中文 (默认) 日语 韩语 罗马尼亚语 斯洛伐克语
4_MUI_2	丹麦语 德语 法语 荷兰语 意大利语 西班牙语 瑞典语
5_MUI_3	巴西-葡萄牙语 芬兰语 波兰语 俄语 捷克语 土耳其语 匈牙利语
EULA TERMS 可靠性标准	包含 不包含

安装语言

为了安装其语言, 进行如下步骤:

3.4 可以这样在 Windows XP 下安装其他语言 (DVD)

1. 通过“Explorer”→“工具”→“映射网络驱动器”可以使用已释放的 DVD 驱动器的网络来访问 DVD，如果 DVD 驱动器没有直接连接在 USB 接口上。应当选择 G: 作为驱动器字母。
2. 选择相应语言的目录，接着“MUISETUP.EXE”程序启动。接受许可后，用“继续”启动安装过程。显示所有已经安装的语言和 DVD 上可用语言的列表。
3. 现在可以通过设置和清除语言前的叉号来安装/卸载所需的语言。
4. 其它的预设置：
 - 选择用于标准用户/新用户的语言版本
 - 对于不支持 Unicode 的程序，语言必须设置为“英语（USA）”。
 - 字符组（字体）同样必须设置为“英语（USA）”。
5. 按“确定”确认后开始安装。有些语言（例如：中文）还需要相应 DVD 上的系统文件。如果安装驱动器不是建议的“G:”可以要求“Windows XP 专业版维修包 2 CD”或者“Windows XP 专业版 CD”。路径数据必须作相应的修改。
6. 成功安装后，根据各语言需要来要求重启。

说明

- 只能从之前 PCU 上安装的语言中进行选择。
 - 当用户在切换后重新登录时或者 PCU 重新关/开时，Windows XP 的新语言才生效。
 - HMI 高级版的操作界面语言与之无关。其语言在“开机调试”→“HMI”→“更改语言”下独立设置。
-

备份和恢复数据

4.1 备份和恢复数据

概述

借助软件 **Symantec Ghost** 可以把整个硬盘的内容作为磁盘图像保存。保存这些不同存储载体上的磁盘图像用于以后硬盘上的数据恢复。

出厂时，**Symantec Ghost** 在 PCU 的备用硬盘和整个 PCU 硬盘上发送。**Symantec Ghost** 在下列章节中说明的数据备份和数据恢复时使用。

其他信息请访问互联网网站：<http://www.ghost.com/>

4.1.1 可以这样创建 PCU 的维修系统

使用

对于这种维修情况，基于 WinPE 在 USB 存储器上生成一个便携式的维修系统作为“紧急启动系统” (EBS)。

推荐：

最好使用容量为 512 MB 的 SIMATIC PC USB 闪存驱动。

创建维修系统

为了在 USB 存储器上创建用于 PCU 50.3 的维修系统，在硬盘 D:\Eboot 目录下有 Ghost 图像可供使用。

进行下列步骤：

1. 启动 PCU 50.3，在服务模式下。
2. 在 PCU 50.3 的 4 个 USB 接口之一上连接一个不小于 256MB 的 USB 存储器。
3. 在目录 E:\Tools 中启动 Ghost32.exe
4. 选择 Ghost: 本地 → 硬盘 → 来自图像。选择数据源 D:\Eboot\ebboot.gho 并作为 USB 存储器的目标盘（可识别存储容量）。

结果：

把 ebboot.gho 成功传输到 USB 存储器后，PCU 的维修系统可以使用。

与该步骤类似创建 PG/PC 上的维修系统。为此，必须在 PG/PC 上安装程序 Symantec Ghost。

从维修系统上引导启动

1. 把 EBS 连接在 PCU 50.3 的后端 USB 接口上。
2. 当 PCU 进行 BIOS 引导启动时，按下操作面板上的键 <ALARM CANCEL> 或者外部键盘上的 <ESC>，以显示“启动菜单”。
3. 选择输入项“USB-HDD:XXX”（从可用媒介的列表中）。

结果：

从维修系统开始 PCU 引导启动，并启动 ServiceCenter。

4. 首先选择“启动”来启动 ServiceCenter，然后选择“备份/恢复硬盘图像”，来用硬盘图像恢复硬盘。

注意
启动时： <ul style="list-style-type: none">• 通过直接连接的 OP 的前端 USB 接口（USB V1.1； 后端 USB V2.0）可以从 EBS 同样引导启动 PCU，但是速度明显较慢。• 通过 TCU 的 USB 接口无法从 EBS 引导启动 PCU。• 无法在 EBS 上保存网络设置。• 如果在 PCU 50.3 的 DVI 接口上直接连接 DVI 监视器，则 EBS 不起作用。只能通过 DVI → VGA 适配器来进行 VGA 运行。

参见

可以这样保存和修复硬盘 (页 63)

4.1.2 启动 ServiceCenter Backup-Restore

启动 ServiceCenter Backup-Restore

启动 ServiceCenter Backup-Restore 用于以下任务：

- 备份/恢复数据
 - 通过连接维修桌面启动
 - 启动时通过 SINUMERIK 下的“不可见”输入进行
- 在维修系统的维修情况中
- 安装备件硬盘时

为确保用户有权限执行维修任务，要在 PCU 启动时通过密码保护直接输入。该密码可以修改。

注意

当维修系统通过维修桌面启动 ServiceCenter Backup-Restore，或者在安装备件硬盘时启动，则不需要密码。

双击维修桌面上的“ServiceCenter Backup-Restore（维修中心备份恢复）”后会打开下列对话框：

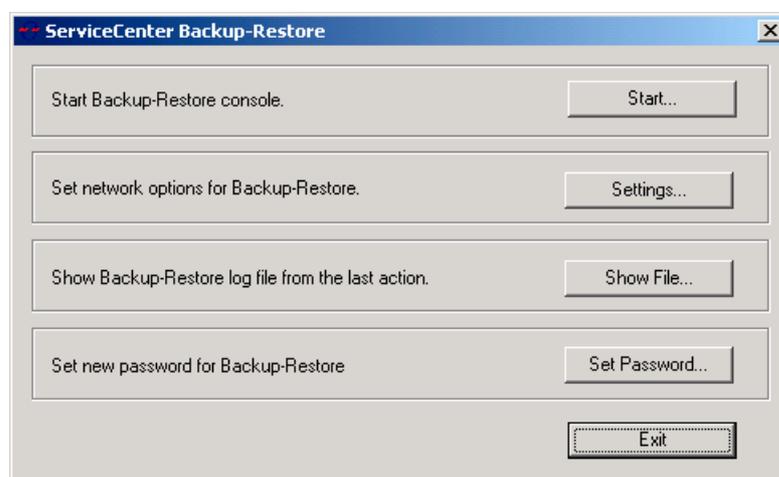


图 4-1 启动维修

- | | |
|----------|---|
| 开始 ... | 用“开始”关闭系统并启动 ServiceCenter。 |
| 设置 ... | 用“设置”打开用于网络设置的对话框。 |
| 显示文件 ... | 在“显示文件”下可以观察上一个数据备份的记录。 |
| 设置密码 ... | 为 ServiceCenter Backup-Restore 设置一个新密码。
(与维修用户的预设置相同为“auduser”。) |

退出 中断并返回维修桌面。

网络设置

1. 连接 PCU 和编程机或者 PC 时选择“设置”，用来检测设置的 IP 地址或者可能重新进行设置。
2. 选择 “使用 Windows 设置”，用于保留出厂设置。(这里是预设置。)

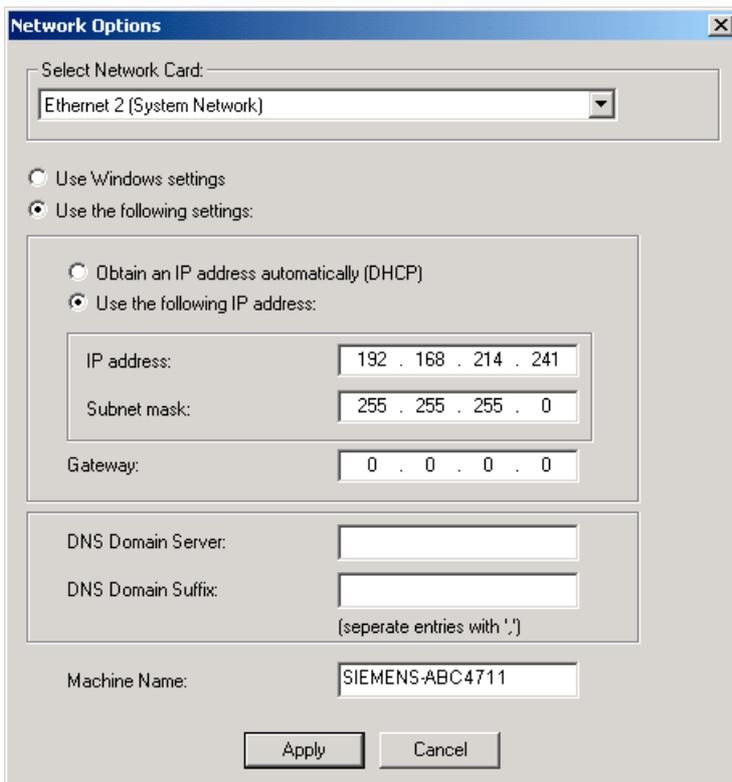


图 4-2 网络设置

3. 选择 “使用下列设置”，用于设置新配置：
 - 用 “自动获取 IP 地址 (DHCP)”来得到一个 DHCP 维修器自动分配的 IP 地址。
 - 用“使用下列 IP 地址”在 192.168.214.250 – 254 带有子网屏幕 255.255.255.0 的范围内输入 IP 地址。
4. 为了激活 DNS 名称服务，需要在“DNS 域服务器”中给定服务器的 IP 地址并在“DNS 域后缀”中给定扩展名，例如“network.com”。
如果没有输入，则预设置为 “.local”。

说明

更改规定的网络设置，仅在重新启动 PCU 后才有效。
而如果从维修桌面开始更改网络设置，则更改会被立刻接受。

参见

系统属性 (页 9)

4.1.3 选择维修任务

选择维修任务

启动 ServiceCenter 后, 显示下列对话框:

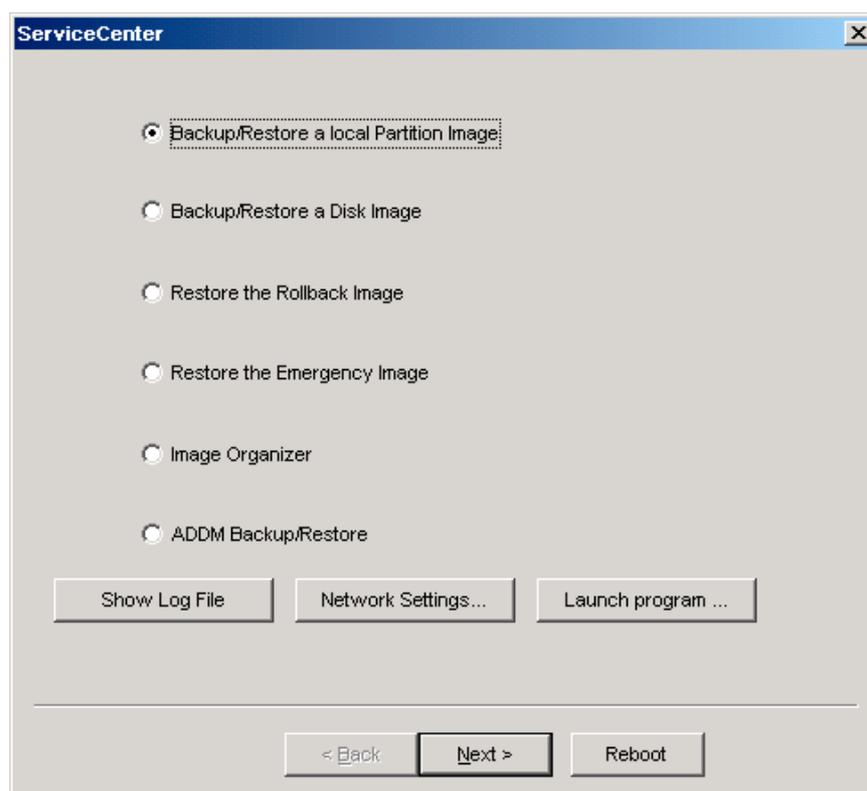


图 4-3 选择 ServiceCenter

选择下列维修任务:

- 备份/恢复一个本地分区图像 (“Backup/Restore a local Partition Image”)
- 备份/恢复一个硬盘图像 (“Backup/Restore a Disk Image”)
- 恢复当前图像 (“Restore the Rollback Image”)
- 恢复 Windows 系统分区图像 (“Restore the Emergency Image”)
- 管理图像 (“Image Organizer”)

显示记录文件 (“Show Log File ...”)

这里打开 bacres.txt 文件, 对所有备份作了记录。

网络设置 (“Network Settings”)

用“网络设置”打开用于网络设置的对话框。

启动程序 (“Launch Program ...”)

为了启动维修系统中的程序，在这里输入程序名称，例如：“cmd”用于启动 DOS Shell。

4.1.4 可以这样保存和修复本地分区

备份分区

1. 在“备份/恢复本地分区图像”下选择指令“备份”，在硬盘分区 D:\Images 上保存一个或多个分区 C, E 和 F 的图像：

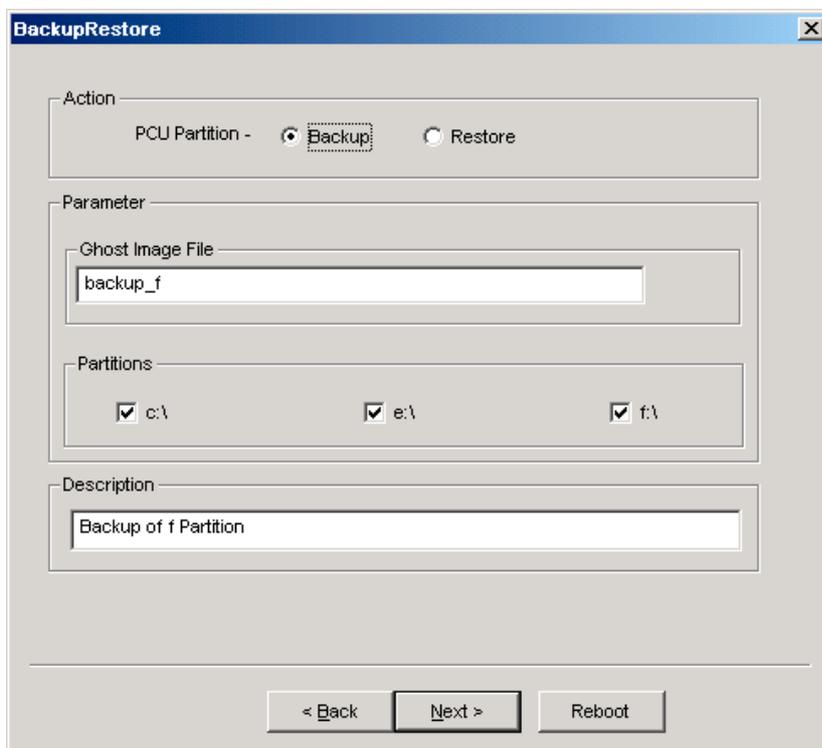


图 4-4 本地分区备份

2. 选择想要创建图像的分区。
3. 在启动备份前，在每个分区的下一个对话框中输出大小。

推荐：

如果保存了备份并希望以后重新修复，推荐创建 C、E 和 F 分区的完整图形。

恢复分区

在“备份/恢复本地分区图像”下选择指令“恢复”，在硬盘分区 D:\Images 上恢复保存一个或多个分区 C, E 和 F 的图像。

恢复当前图像(“Restore the Rollback Image”)

为了恢复上次保存过的即：新图像(“Rollback Image”)，选择“恢复反转图像”。
“反转图像”是上次创建的分区备份。

4.1.5 可以这样保存和修复硬盘

备份硬盘

选择“备份/恢复硬盘图像”，来通过网络连接保存硬盘图像：

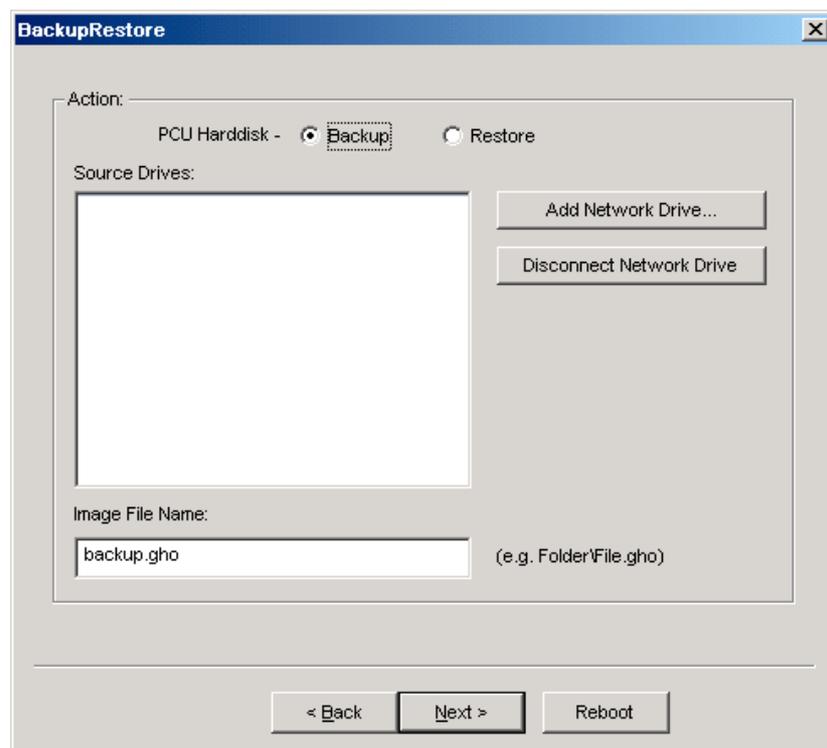


图 4-5 通过网络驱动器保存硬盘

1. 为了与访问使能驱动器建立网络连接，选择“添加网络驱动器”并对于“图像文件名称”注明文件名称。
2. 在“共享”下，输入计算机名称和使能的目录。
3. 为了得到存取授权，输入一个用户名和密码。

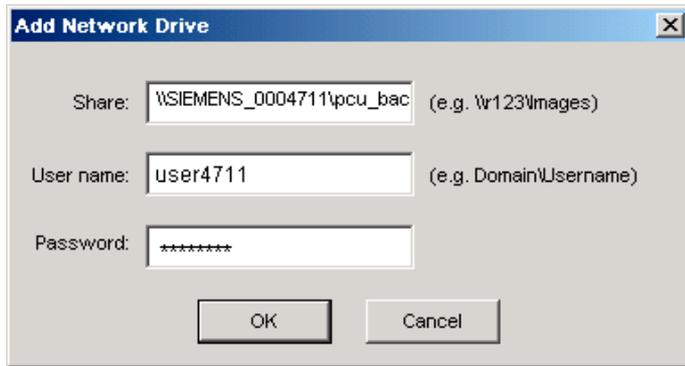


图 4-6 驱动器连接

- 4. 在“选项”下选择要生成的硬盘图像是否在特定大小的多个文件中列出，使这些文件每次匹配一张 CD。

恢复硬盘

在“备份/恢复硬盘图像”下选择“恢复”指令，来恢复图像。用“下一步>”来按步骤执行。

参见

前提条件 (页 66)

4.1.6 从“紧急图像”恢复系统文件

恢复系统数据

选择“恢复紧急图像”功能来恢复紧急图像。该图像必需包括 E 分区备份并能够包括分区 C、D 或者 F。规定仅为了分区 E: 上的系统有故障的情况。分区 F: 上的用户数据 在当前状态下仍可以获得。

 小心
读入分区 E: 的“紧急图像”仅当生成备份后没有安装或配置额外的软件，或者所有安装在分区 F: 上的应用程序注册项 都必须包含在该图像中。 使用功能“图像管理器” 仅为了识别满足这些条件的作为紧急图像的图像。

为了用“紧急图像”恢复系统数据必须引导启动维修系统（EBS）的 PCU:

管理图像管理器

选择“图像管理器”，从显示的列表中标识一个图像作为紧急图像或者删除现有的图像。

参见

可以这样创建 PCU 的维修系统 (页 57)

4.2 备份 HMI 高级版的运行环境

设置初始 SINUMERIK HMI 运行环境

在维修桌面上，提供脚本文件形式的功能“初始 SINUMERIK HMI 环境”。执行该功能时，设置初始交付状态，即：保存下列目录的内容：

- C:\RUNOEM
- F:\ADD_ON
- F:\OEM
- F:\USER

随即清空这些目录。

设置当前 SINUMERIK HMI 环境

在维修桌面上，提供脚本文件形式的功能“当前 SINUMERIK HMI 环境”。执行这些功能时，将再次取消初始交付状态设置，即：退回目前保存的目录内容。

4.3 PG/PC 与设备网络中的 PCU 连接

应用情况

对于下列应用情况，例如：需要在设备网络中在 PCU 和 PG/PC 之间进行连接：

- 用于在 PG/PC 上保存 PCU 50 的硬盘备份图像。
- 用于通过 PG/PC 的 CD-ROM 驱动器来恢复 PCU 50 硬盘。
- 引导启动替代件硬盘。

4.3.1 前提条件

概述

在下列图像中显示设备网络中的典型连接方式：

- PCU 在“Eth 2”上与直接用交叉的 Ethernet 电缆连接维修 PG/PC
- PCU 在“Eth 2”上通过带不交叉 Ethernet 电缆的开关连接维修 PG/PC

如果要通过公司网络（Eth 1）连接维修 PG/PC，请咨询您的网络咨询员。

接口和连接的含义：

- Eth 1 作为 DHCP 客户机
- Eth 2 作为 DHCP 维修器
- Eth 2 带固定 IP 地址
- 绿色连线：未交叉的以太网电缆
- 灰色连线：交叉的 Ethernet 电缆（crossover）

直接在 PCU 上配置 PG/PC

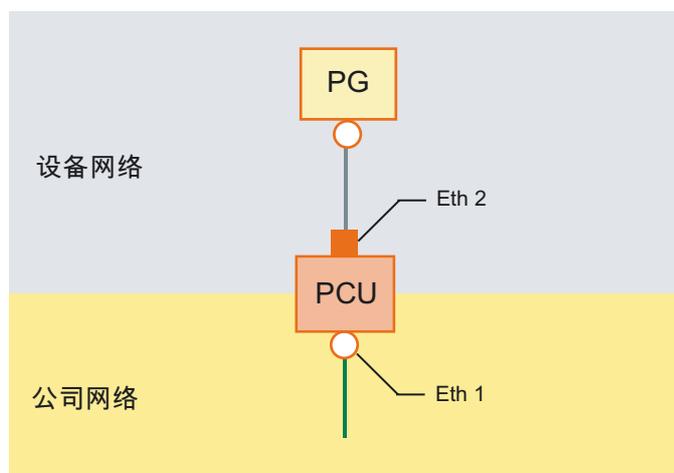


图 4-7 PG 直接连接在 PCU 上

配置 PCU 上的 PG/PC 和开关

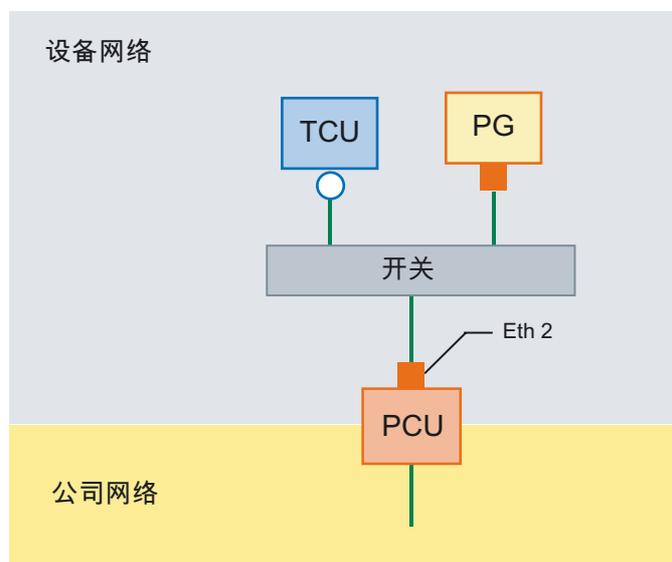


图 4-8 PG 通过开关连接在 PCU 上

说明

如果关闭或再次打开没有专用 OP/TP 包括 TCU 在内 PCU，并随后从维修系统（EBS）开始引导启动 PCU，则需要外部的 VGA 监视器和键盘来操作 PCU。

当 TCU 运行时，以及 PCU 使用 EBS 进行启动未关闭时，不需要外部 VGA 显示器和键盘。

运行原理

在带有 Windows XP 的 PG/PC 上：

- 连接 PCU 50.3 上的 PG/PC，按照上述的图像配置之一进行。
- TCP/IP 协议用作网络协议。
在 PCU 基础软件中已经预配置了 TCP/IP。
- 在相同的子网络中设置 IP 地址。
- 在 PG/PC 上使能一个用于网络存取的目录。

在 PCU 上，位于 WinPE：

- 在 PCU 50.3 上启动 WinPE 下的维修中心。
- 建立与 PG/PC 使能目录的网络连接。
- 用“备份”功能在 PG/PC 的使能目录中保存用于该维修情况的 PCU 硬盘的 Ghost 图像。
- 用“恢复”功能在 PG/PC 的使能目录中从 Ghost 图像恢复的 PCU 50.3 硬盘。

参见

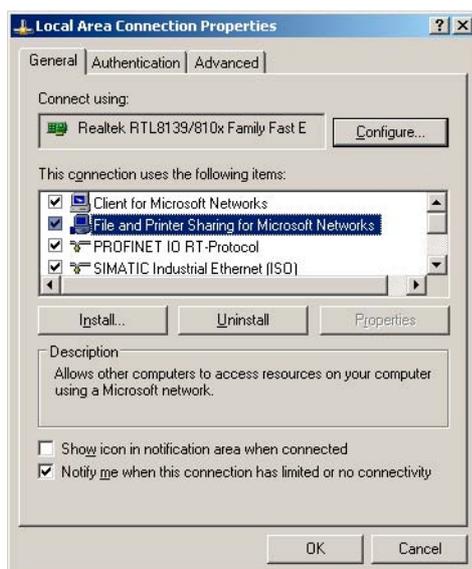
可以这样保存和修复硬盘 (页 63)

4.3.2 在设备网络中可以这样连接带有 PCU 的 PC/PG

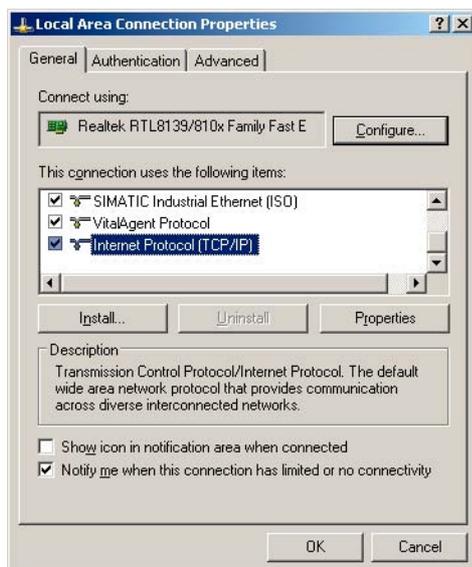
带有 Windows XP 的 PG/PC 上的设置

在 PG/PC 上执行下列设置：

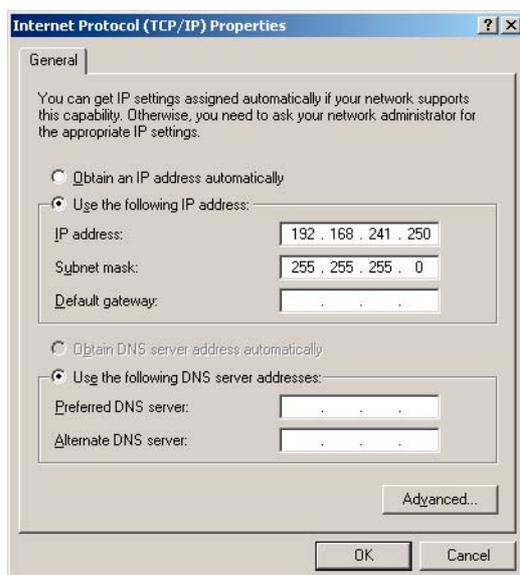
1. 选择“控制面板” → “网络连接” → “本地区域连接属性”，然后得到下列对话框：



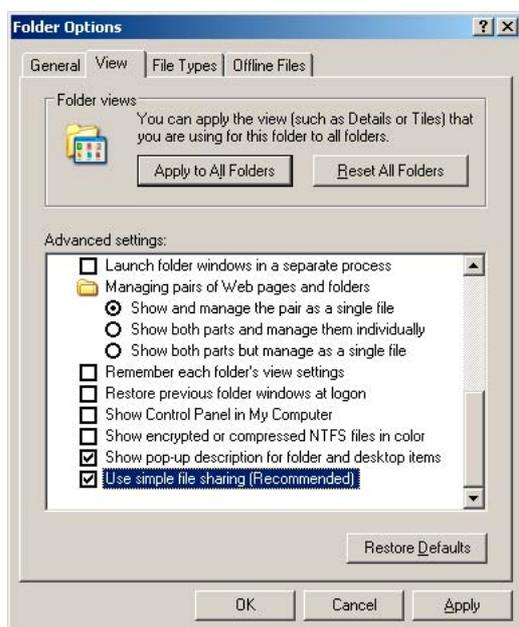
2. 检查是否选择了“文件和打印机共享 ...”，以此能够使能目录并选择“网络协议 (TCP/IP)”。



3. 打开“属性”对话框并选择“使用下列 IP 地址”，来输入 IP 地址，例如 192.168.214.250 和子网掩码 255.255.255.0。

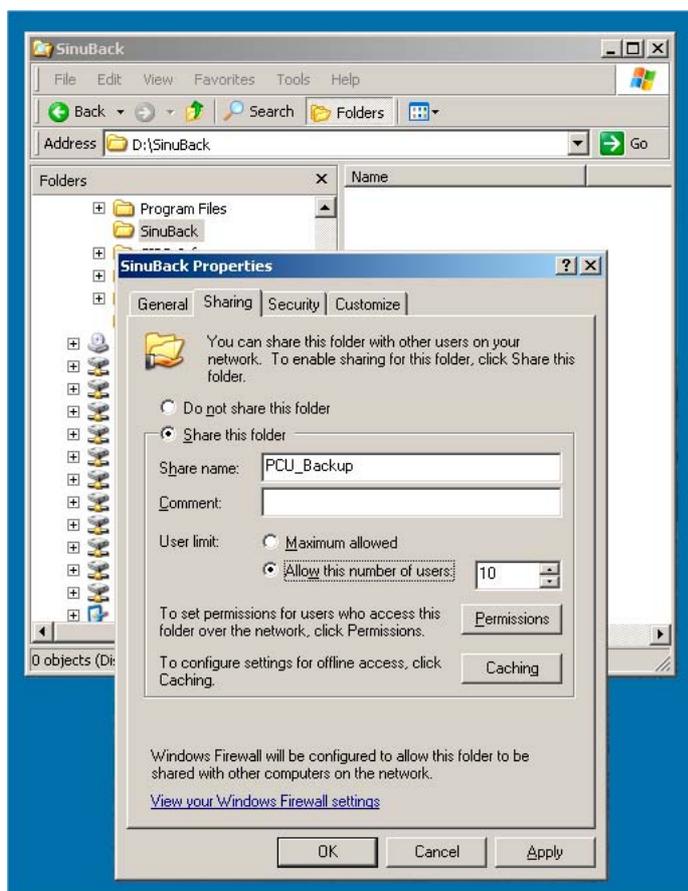


4. 选择“控制面板”→“系统”→标签“计算机名称”，来获悉 PCU 的计算机名称：例如 SIEMENS-ABC4711
5. 选择“控制面板”→“文件夹选项”→“查看”并激活“使用简单文件夹共享 (推荐)”，来避免目录释放时出现问题。



使能用于网络存取的目录 (Windows XP)

1. 在本地驱动器上建立目录；例如 D:\PCU_Backup
2. 用鼠标右键打开目录的“属性”对话框和标签“共享”。



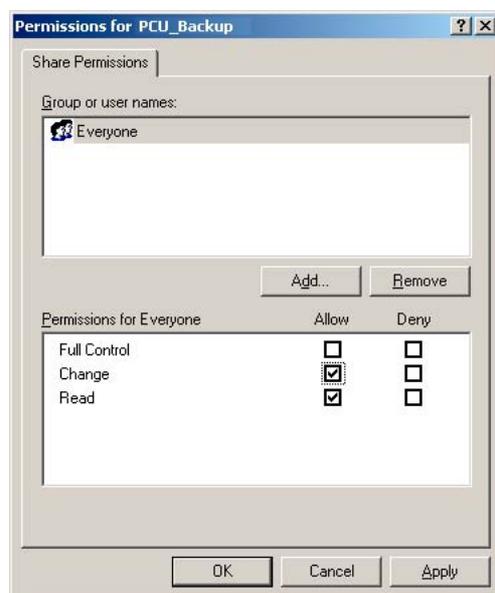
3. 选择“共享该文件夹”。接受目录名称作为使能名称（共享名称），例如：**PCU_Backup**。

更改目录名称，必须在连接驱动器时注明新的名称！

说明

注意保留足够的空余存储空间，以便生成备份时能够在 **PG/PC** 上保存 **Ghost** 图像。

4. 为了允许在该目录中保存（例如：**Ghost** 图像），选择“许可”，并在“允许”栏中为所有用户激活复选框更改”。



PCU 50.3 上的过程

在 PCU 上执行下列步骤：

1. 用“启动备份/恢复控制台”启动 ServiceCenter。
2. 在 PCU 上保持“网络选项”的预设置：

PCU 的 IP 地址： 192.168.214.241 使用子网络掩码 255.255.255.0
PG/PC 的 IP 地址： 192.168.214.250 使用子网络掩码 255.255.255.0

3. 在 ServiceCenter 中选择维修任务“备份/恢复硬盘图像”。
4. 建立与使能目录的网络连接，例如 \\SIEMENS-ABC4711\PCU_Backup。
5. 恢复带有 Ghost 图像的 PCU 硬盘。

说明

当“恢复”过程中传输中断，在硬盘上没有一致的系统，即：不能重复“恢复”过程，因为 PCU 不再启动。

这种情况下使用 USB 存储器上的“紧急启动系统”。

参见

可以这样保存和修复硬盘 (页 63)

可以这样创建 PCU 的维修系统 (页 57)

4.4 调试备件硬盘

概述

关于 PCU 50.3 硬盘的机械/电子部件更换所要求的措施说明参见：

文献： 操作组件与联网手册

说明

备件硬盘中没有 Windows 操作系统和 HMI 系统软件。

出厂时，ServiceCenter 包括程序 Symantec Ghost 随每个 PCU 和备件硬盘发送。

生成硬盘备份（磁盘图像）

借助软件 Symantec Ghost 可以把 PCU 的全部硬盘内容保存为“硬盘图像”。该硬盘文件可以保存在不同的存储媒介上用于以后的硬盘恢复，例如：在 CD-ROM 或者网络驱动器上。

调试备件硬盘

装入备件硬盘后，PCU 的 Ethernet 接口预设置如下：

- Ethernet 1（公司网络）作为标准 DHCP 用户
- Ethernet 2（系统网络）作为 SINUMERIK-DHCP 服务器，使用固定的 IP 地址 192.168.214.241 和子网掩码 255.255.255.0

因此要将 PCU 以及安装之前的备件硬盘与设备网络分离。

硬盘调试时如下进行：

1. 按照推荐配置连接 PG/PC。
 2. 启动 ServiceCenter 并选择“恢复硬盘图像”。
-

说明

当“恢复”过程中传输中断，在硬盘上没有一致的系统，即：不能重复“恢复”过程，因为 PCU 不再启动。

这种情况下使用“紧急启动系统”。

参见

前提条件 (页 66)

可以这样保存和修复硬盘 (页 63)

可以这样创建 PCU 的维修系统 (页 57)

诊断和维修

5.1 PCU 硬件诊断

使用目的

通过集成的“安全卡”，PCU 硬件支持重要的系统组件诊断，该安全卡被称作主板上的安全卡（SOM）。

该诊断功能仅由带有 HMI 高级版的系统来分析。硬件故障状态通过 HMI 高级的操作面板作为报警发送。由此不仅在 HMI 高级版中数据可视而且可以进行外部分析。

监控参数

监控下列 PCU 50.3 硬件的物理参数：

- CPU 温度
- 机箱温度
- I/O 芯片温度
- 两个机箱风扇的转速
- S.M.A.R.T –硬盘状态

记录无 HMI 高级版的故障

在 Windows 的事件记录中，PCU 硬件监视器将记录所有硬件故障，即使没有安装 HMI 软件的 PCU 上也可以输出故障。

在记录“控制面板” → “管理工具” → “事件浏览器”中输出报警。

说明

HMI 高级版的 PLC 接口位于 PLC 程序的 DB 10 中，并在故障情况下由 PCU 硬件提供。

报警说明：SINUMERIK 诊断手册

操作组件与联网手册：章节 PCU 50.3, 备件

参见

调试备件硬盘 (页 72)

5.2 运用 7 段显示

使用目的

7 段显示用于 PCU 50.3 在无本地 OP 运行时（所谓的：“无头运行”）的首次诊断。如果确定了启动故障，则需要连接一台本地显示器用于分析其他故障。

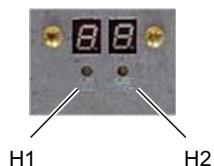


图 5-1 带 LED 的 7 段显示

这两种显示方式都具有下列功能：

- 段显示 H1
段显示 H1 和相应的 LED 被分配给 PCU 基础软件。在系统引导启动、正常运行和关机时会输出状态代码。需要考虑与 NCU 相关的兼容性。
- 段显示 H2
段显示 H2 和相应的 LED 被分配给应用软件。

状态代码的含义

- 当系统引导启动时：

LED H1	LED H2	7 段显示	状态	
桔黄色	桔黄色	BIOS 状态代码的输出	在系统接通之后	
桔黄色	关闭		在 BIOS 运行之后	
桔黄色	关闭		在 Windows 启动之后	
		1	启动 Windows	装载 Windows 启动所需要的设备驱动器
		2	PCU 硬件维修	PCU 硬件维修已启动。
		5	网络	等到网络接口运行准备完毕。
		8	TCU 支持 测试步骤 1	等到 FTP 服务器启动完毕
		9	TCU 支持 测试步骤 2	等到 TCU 网络启动的引导启动服务器以及 TCU 硬件维修启动完毕
		A	TCU 支持 测试步骤 3	等到 VNC 服务器启动完毕
		B	HMI 管理器：	等到 HMI 管理器启动完毕

- 在运行中:

LED H1	LED H2	7 段显示	状态	
绿光闪烁	--	0	OK	
红光闪烁	--		故障:	
		1	温度报警	外壳或 CPU 温度; 温度超过极 限值 (SOM 故障)
		2	风扇报警	外壳或 CPU: 风扇转速过低或 风扇故障 (SOM 故障)
		3	硬盘报警	S.M.A.R.T 硬盘故障
		6		VNC 服务器故障或 VNC 服务 器服务停止

- 关机时

LED H1	LED H2	7 段显示	状态	
红/绿光闪烁	--	5	进行关机	

5.3 引导启动时激活/取消激活故障记录

应用

每次引导启动都会在文件 D:\\$\$Base.log 中描述一个信息程序段，包含有日期、时间和管理干预指令。

关于 HMI 管理器控制的引导启动信息可能会显示在屏幕上，并写入日志文件 D:\\$\$Base.log。

日志文件中输出

通过注册表/值控制输出：

- 信息方式

‘HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\<version>\HMIManager\InfoLevel’ 设置要输出何种信息方式：

信息等级(DWORD)

- = 1: 输出必需的信息（预设置）
- = 2: 输出必需的信息和补充信息
- = 3: 输出必需的信息、补充和跟踪信息
(信息等级 <= 0 处理方式同信息等级 == 1;
(信息等级 > 3 处理方式同信息等级 == 3)

- 输出屏幕信息

‘HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\<version>\HMIManager\ShowInfo’ 设置，是否信息 - 除了输出日志文件中外 - 也要在屏幕上显示：

显示信息 (DWORD)

- = 0: 显示必需的信息，
没有显示补充信息和跟踪信息
- = 1: 显示必需信息、补充和跟踪信息（预设置）

5.4 可以这样在设备网络中查找用户

使用目的

可以在设备网络中使用 VNC 扫描器。扫描器的主要用于识别那些包含了有效 VNC 服务器的设备，例如 NCU 或 PCU。

VNC 扫描器可以直接与 VNC 服务器连接，并服务于例如 HMI 应用程序。这时需要使用集成的 VNC 观察器。

此外还可以列出全部的网络设备，并以此建立设备网络概览。

说明

VNC 扫描器可以安装在 PG/PC 上 PCU 50.3 的维修桌面以及 PCU 基本软件的 DVD 上。

操作该程序需要使用鼠标。

操作“VNC 扫描器”



1. 使用 PCU 50.3 维修桌面上的连接来启动程序“VNC 扫描器”。

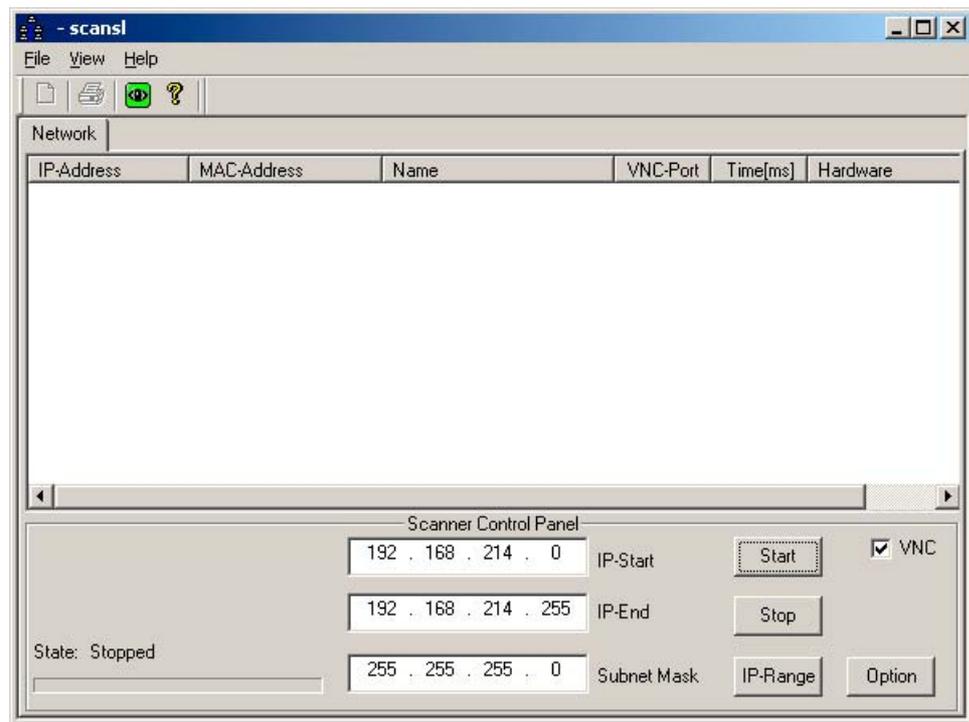


图 5-2 VNC 扫描器

2. “IP 启动 / IP 结束”：设置需要查找的 IP 地址范围。
可以保存 8 个地址范围。使用按钮“IP 范围”切换至下一范围。
3. “VNC”选项：

- 打开：仅查找 VNC 服务器。
 - 关闭：查找网络中的所有用户。
4. 使用“启动”按钮开始进行查找。
会列出用户的下列参数：
- IP 地址
 - MAC 地址
 - 网络中的名称
 - VNC 端口
 - 以毫秒为单位的用户应答时间
 - 设备类型

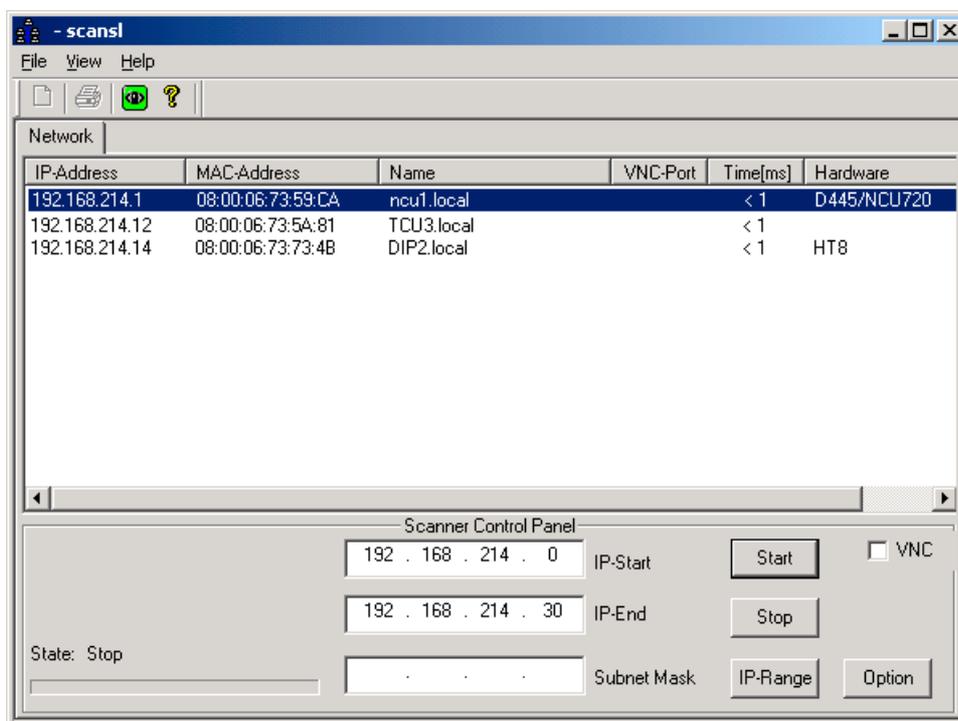


图 5-3 VNC 扫描器：用户列表

5. 在列表中选择一位网络用户，可以通过关联菜单（鼠标右键）执行下列动作：
- “VNC”：直接使用 VNC 服务器进行连接并在给定的情况下对 HMI 进行操作。
 - “VNC → 自动重新连接”：按照选项“如服务器断开连接，自动重新连接”可以恢复与选定网络用户的连接。
 - 声脉冲信号：打开指令解释程序，并且执行到选定网络用户的“ping”指令。

说明

通过 VNC 观察器进行操作：

只可以通过外部的 VNC 观察器从其他站对系统进行观察（预设置）。

要从其他站对系统进行操作，必须经该系统同意。可以在“系统网络中心”中进行相关设置或者直接在文件 `tcu.ini` 的段落 `[VNCViewer]`中设置。

如果要通过带有 VNC 扫描器或 VNC 观察器的公司网络在 PCU 50.3 上进行存取，则必须在 PCU 50.3 上的“控制面板” → “Windows 防火墙” → “例外”中添加端口 5900

文献： 操作组件手册和网络连接

其他选项



1. 使用该按钮启动 VNC 观察器。
2. 使用按钮“选项”可以对查找参数进行设置。

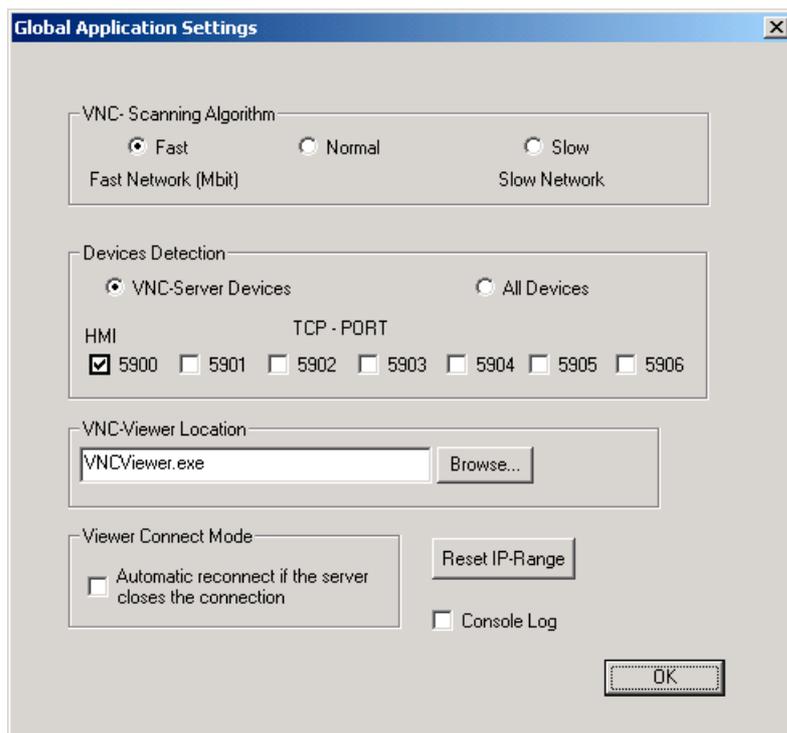


图 5-4 VNC 扫描器： 设置查找参数

- VNC 扫描法则：
 - 预设置为“快速”，即设备的反馈大约在 100 毫秒以内
- 侦测设备：（与选项“VNC”一致，参见第 3 点）。
 - “VNC 服务器设备”： 仅查找 VNC 服务器。
 - “所有设备”： 查找网络中的所有用户。

- VNC 观察器位置:

VNC 观察器的快捷连接（打开源程序；已经预先安装在 PCU 上；在维修 PC 上必须给定该连接）。

- 观察器的缺省连接模式:

“如服务器断开连接，自动重新连接”（预设置： 关闭）

– 关闭： 会尝试建立一次连接。

– 打开（循环模式）： 例如为了达到维修目的需要通过互联网进行存取，则会等待较长时间并会在数毫秒后重新进行尝试与该用户进行连接。

- “复位 IP 范围”： 所有的 IP 参数都将复位。

- 控制台日志： 打开控制台，输出例如一份故障分析的记录（预设置： 关闭）。

VNC 观察器的选项

小心
预设置未改变!
为了保证 VNC 观察器可以正确工作，不允许改变下列选项。

VNC 观察器启动后，会打开下列对话框：

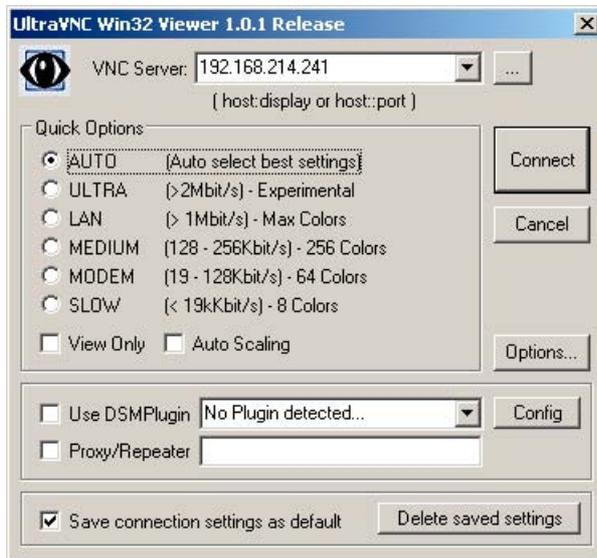


图 5-5 预设置： UltraVNC

按下“选项...”按钮后，会打开下列对话框：

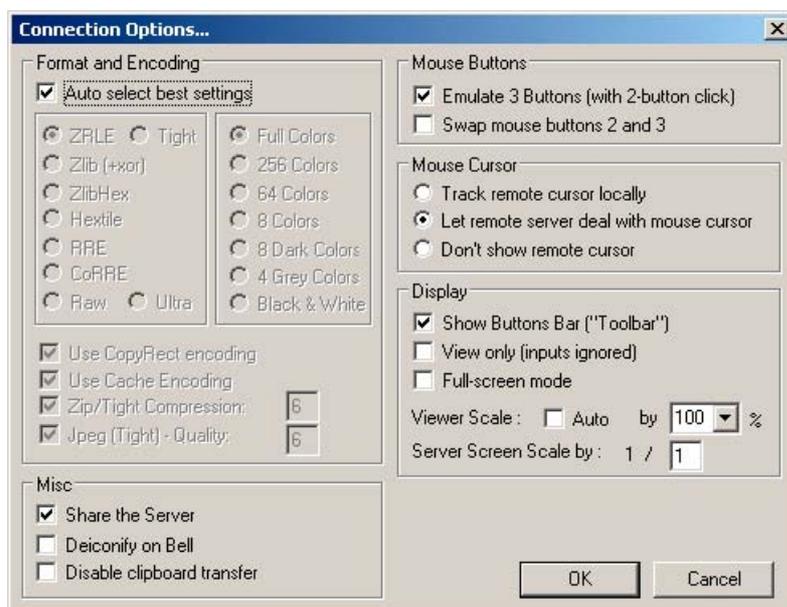


图 5-6 预设置：连接选项 ...

5.5 用于 WinSCP 和 Putty 的 OpenSSH

使用目的

使用维修工具 WinSCP 和 Putty 可以在 NCU 7x0 上进行存取。现在可以使用它们以同样的方式在 PCU50 上进行存取。

预设置

用户“auduser”通过登录名称+密码的方式释放 SSH 入口。享有优先权的 SSH 客户机为 Putty 或 WinSCP。

功能

可以由维修桌面通过“启动”→“程序”→“维修工具”→“COPSSH”打开 COPSSH 工具“激活用户”和“撤销用户”包括文献。

通过 SSH 进行登录，会在开始进入登录时将用户的本地目录设置为当前目录。通过更改目录至 `.././cygdrive` 也可以扩展到下级的驱动器 `c:d: e: f:` 。

另见

“NCU 操作系统”(IM7)开机调试手册：
关于 WinSCP 和 Putty 的说明和许可。

缩略语列表

A.1 缩写

CF	Compact Flash: 存储卡
DCK	Direct Control Keys: 直接控制按键
DHCP	Dynamic Host Configuration Protocol: 动态主机配置协议, 向网络内的计算机动态分配 IP 地址和其他配置参数
DNS	Domain Name System: 域名系统, 将域名转换为 IP 地址
EBS	紧急启动系统
EKS	电子密码系统: 用于检验并识别用户的系统 (授权系统)
HMI	Human Machine Interface: 操作界面
IRT	Isochronous Real Time (Ethernet)
MAC	媒介访问控制 (MAC): MAC 地址是一个 48 位的 Ethernet-ID。
MCP	Machine Control Panel 机床控制面板
MPI	Multi Point Interface: 多点接口
MUI	Multilingual User Interface
NCK	Numerical Control Kernel: 带有程序段处理、运行范围等等的数字内核
NCU	Numerical Control Unit: NCK 硬件单元
NRT	Non Real Time (Ethernet)
NTFS	New Technology File System
NTP	Network Time Protocol: 网络时间协议, 同步整个网络时间的标准
NTPD	NTP Daemon: 在后台运行, 无需由用户启动的服务程序。
PCU	PC Unit: 计算机单元
PG	编程器
PLC	Programmable Logic Control: 存储器可编程控制系统
PROFIBUS	Process Field Bus 自动化技术中的场总线通讯标准
RAM	Random Access Memory: 随机存储器, 可读取、描述的程序存储器
RDY	Ready: 就绪
TCU	Thin Client Unit
TFTP	微小文件传输协议: 非常简单的数据传输协议
UDP	User Datagram Protocol: 用户数据包协议, 一般通过 UDP 处理 NTP
USB	Universal Serial Bus
USV	不间断电源
UTC	Universal Time, Coordinated: 协调世界时
VNC	Virtual Network Computing

索引

7

7 段显示, 70

B

BIOS

- 启动 Setup, 22
- 系统参数, 23
- 修改预设置, 26

N

NTFS 文件系统, 6

P

PCU

- 交货状态, 5
- 名称 (预设置), 8
- 检查硬盘, 69
- 操作系统, 7

PCU 维修系统, 51

PG 在设备网络中, 60

S

SITOP 显示器

- 参数化, 40
- 配置, 39

U

USB 存储器, 51

USV 模块 (SITOP), 39

V

VNC 扫描器, 72

W

Windows 语言 (MUI), 31

四划

分辨率

启动时, 20

文件系统 NTFS, 6

五划

用户

- 删除, 11
- 更改名称, 11
- 继承, 14
- 预设置, 9

用户组, 10

用户类型, 9

六划

关机, 71

多语言用户接口 (MUI), 48

存取等级, 17

安装 SIMATIC STEP 7, 47

网络设置, 53

网络运行, 45

七划

启动 OEM 程序, 37

启动 ServiceCenter, 52

启动画面

切换, 33

存放位置, 32

更改计算机名称, 27

系统引导启动, 70

八划

- 备件硬盘, 67
- 备份 HMI 的运行环境, 59
- 备份/恢复
 - 分区图像, 56
 - 反转图像, 57
 - 硬盘图像, 57
- 备份数据, 51
- 软件
 - 安装, 45
 - 组件, 5
- 软件组件版本, 5

九划

- 恢复系统数据, 58, 62
- 恢复盘, 48
- 恢复数据, 51
- 故障记录, 71
- 选择语言 (MUI), 31

十划

- 监控 HMI, 43
- 紧急启动系统, 51
- 紧急图像, 58, 62
- 配置操作界面, 31

十一划

- 域, 29
- 域名系统 (DNS), 54
- 维修任务, 55
- 维修桌面
 - 安装, 45
 - 启动, 52
 - 背景, 33

十二划

- 硬盘分区, 6
- 缓冲时间, 41

十四划

- 端口 80 显示, (参见 7 段显示)

十六划

- 操作系统 PCU, 7
- 激活记录, 71
- 激活防火墙, 7

SIEMENS

SINUMERIK 810D / 840Di / 840D

PCU 基础软件

调试手册

系统开机调试

1

配置系统

2

软件安装和网络运行

3

备份和恢复数据

4

适用于：

软件 PCU 基础软件版本 7.6

10/2006

6FC5397-0DP10-0RA0

安全技术提示

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

带有警告三角，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。

小心

不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。

注意

表示如果不注意相应的提示，可能会出现不希望的结果或状态。

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

仅允许安装和驱动与本文件相关的附属设备或系统。设备或系统的调试和运行仅允许由**合格的专业人员**进行。本文件安全技术提示中的合格专业人员是指根据安全技术标准具有从事进行设备、系统和电路的运行，接地和标识资格的人员。

按规定使用

请注意下列说明：



警告

设备仅允许用在目录和技术说明中规定的使用情况下，并且仅允许使用西门子股份有限公司推荐的或指定的其他制造商生产的设备和部件。设备的正常和安全运行必须依赖于恰当的运输，合适的存储、安放和安装以及小心的操作和维修。

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。标签中的其他符号可能是一些其他商标，这是出于保护所有者权利的目地由第三方使用而特别标示的。

责任免除

我们已对印刷品中所述内容与硬件和软件的一致性作过检查。然而不排除存在偏差的可能性，因此我们不保证印刷品中所述内容与硬件和软件完全一致。印刷品中的数据都按规定经过检测，必要的修正值包含在下一版本中。

目录

1	系统开机调试	5
1.1	系统交货状态	5
1.1.1	硬盘分区	6
1.1.2	系统设置	7
1.2	管理用户	9
1.2.1	设定了哪些用户？	9
1.2.2	修改维修用户	10
1.2.3	用户全局设置	11
1.2.4	用户专用设置	14
1.3	PCU 启动属性	15
1.3.1	PCU 启动：未安装 HMI 程序	15
1.3.2	PCU 启动：HMI 程序已安装	16
1.4	BIOS 设置	17
2	配置系统	23
2.1	服务菜单的功能（DOS-功能）	23
2.1.1	服务菜单（DOS）参数配置	25
2.1.2	基本菜单操作	26
2.1.3	设置初始 SINUMERIK HMI 环境	27
2.1.4	启动 DOS Shell 并检查软件版本	29
2.1.5	启动服务桌面	30
2.1.6	检查硬盘	31
2.2	配置操作界面	31
2.2.1	HMI 启动画面的保存位置	31
2.2.2	显示自定义启动画面	32
2.2.3	更改维修桌面背景	33
2.3	启动时用户专用设置	34
2.3.1	启动 HMI 程序	34
2.3.2	启动时运行维修桌面	35
2.3.3	启动 OEM 程序	37
2.3.4	引导启动时激活/取消激活故障记录	38
2.4	PCU 带有 SITOP USV 模块	39
2.4.1	启动并配置 SITOP 显示器	40
2.4.2	SITOP USV 模块配置	42
2.4.3	用于关闭 HMI 的配置	45
3	软件安装和网络运行	47
3.1	软件的安装和升级	47
3.1.1	安装升级或服务包	47
3.1.2	安装“安静模式”升级或服务包	48
3.1.3	对于 FAT16 文件结构的安装	51
3.1.4	对于 FAT32 文件结构的安装	52
3.1.5	通过网络连接来安装	54

3.2	网络驱动器的设置	58
3.3	用服务桌面运行网络	64
3.3.1	SIMATIC STEP 7 安装与授权	66
3.3.2	通过服务菜单授权 SIMATIC STEP 7	67
4	备份和恢复数据	69
4.1	通过 Ethernet 连接进行数据备份/修复	69
4.1.1	向 PG/PC 硬盘备份 PCU 硬盘	69
4.1.2	从 PG/PC 硬盘修复 PCU 硬盘	73
4.1.3	使用备用硬盘	79
4.2	本地备份/修复 PCU 分区	81
4.2.1	备份分区	83
4.2.2	恢复分区	85
4.2.3	E 区 : (Windows) 在 PCU 上修复	87
4.2.4	删除备份图像	89
4.2.5	确定紧急备份图像	90
4.2.6	分区 E: 上的紧急备份 : 录入	92
4.3	一个 TCU 配置时的数据备份/修复	93
	索引	索引-97

系统开机调试

1.1 系统交货状态

概述

功能强大的 SINUMERIK PCU 50 在机身上就有用于通讯的接口 (Ethernet 和 PROFIBUS-DP/MPI) - 集成的未占用的插接位置可用于其他任务。反面的 USB 接口可用于 PC 标准键盘和鼠标的“热插拔”。SINUMERIK PCU 50 和 Windows XP ProEmbSys 操作系统以及数据保存软件 Ghost 6 和 7 一起交付。

可另行订购操作界面软件 HMI 高级。

文献：“操作组件”手册

交付时 PCU 上所带的软件

PCU 交付时已经安装好的软件包括下列组件等：

MS Windows XP Professional SP2	
Internet Explorer	V 6.0
MPI 驱动器	V 6.03
Symantec Ghost (包括 Ghost Explorer)	V 7.0 (预设置)
NFS 客户	“用于 UNIX 的 Windows 服务”中的 NFS 客户安装在 E:\SFUNFSCLIENT 下。

注意

PCU 基础软件中包含的系统组件版本摘自文件 C:\BaseVers.txt。

不包括软件包“PCU-基本软件薄型客户”，必须另行订购并安装。

用于全部 Ghost 工具的 PCU 硬盘上的文件将安装到路径 E:\TOOLS 下。

1.1.1 硬盘分区

硬盘的划分

硬盘划分为基本 C 盘和其他三个逻辑驱动盘 D、E 和 F。C 盘和 D 盘用于 FAT16 文件存取，E 和 F 用于 NTFS 文件存取。

根据在 Windows XP 系统下在 PCU 上安装 HMI 系统软件的需要来对硬盘进行分区。在 PCU 50 V2 (566 MHz 或者 1.2 GHz，硬盘 10 GB 或者 40 GB) 上可以安装 Windows XP。为了保证数据安全，把 HMI 系统软件和 Windows XP 系统软件分到了不同的硬盘区。

HMI 系统软件可能按照订货要求在交付时已经装好了。
由客户在以后的首次引导启动时安装。

下列映射显示了 PCU 硬盘的使用：

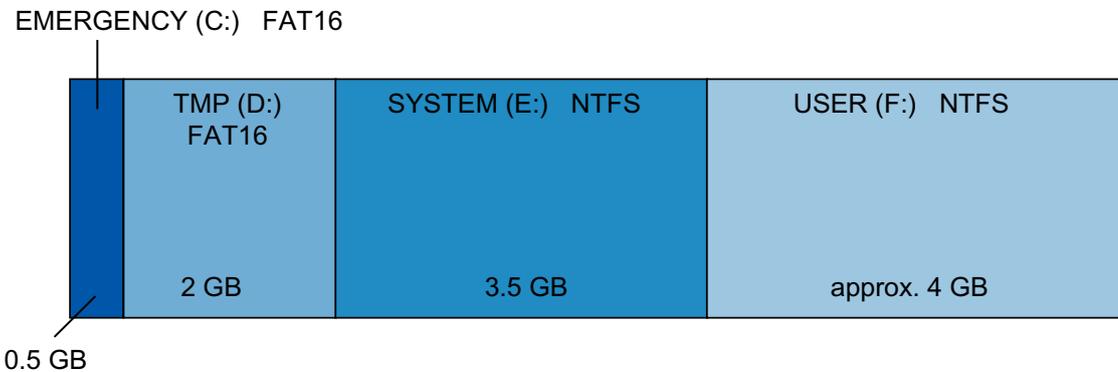


图 1-1 硬盘的划分 (以 10 GB 为例)

分区的内容

各分区规定用于下列文件或者已经被这些文件占用：

- EMERGENCY (C:) C 区包含 DOS 6.2 和实现服务菜单的工具和脚本程序 (如 : GHOST)。
- TMP (D:) 用于保存 GHOST 图像 (如交货状态) 以及本地备份图像。驱动器 D 也包含安装目录，待安装的软件先从远程 PG/PC 上复制到该安装目录，再启动真正的安装过程。
- SYSTEM (E:) 预留给软件 Windows XP。Windows XP 软件在恢复盘 CD 上，通过网络驱动器可用于安装驱动程序或者随后安装升级软件。
- USER (F:) 用于用户程序的安装。必须在这里安装应用程序，如：HMI 系统软件 (包括数据维护和临时数据)、STEP7、用于 HMI 的 OEM 应用程序或用户应用程序。

注意事项

所有应用程序必须安装在 USER (F:) 下；即使该应用程序已经预留了另一个驱动盘上的安装路径。

分区名称 EMERGENCY, TMP, SYSTEM, USER 不允许更改；否则“ServiceCenter”不再有效。

1.1.2 系统设置

PCU 预设置

一个 PCU 50 V2 装有一个集成的 Ethernet 接口。它在这个接口上被预设为标准 DHCP 客户机，用来从公司网络中获得地址。将该 PCU 与一个 NCU 或一个 TCU 相连时需要另外装入第二块带有 Ethernet 接口的 Ethernet 网卡。

注意

如果对于机载 Ethernet 没有设置固定的网络地址，而且网络中没有 DHCP 服务器，在 Windows XP 下分配一个所谓的 APIPA 地址。这至少把系统引导启动时间延长了 1 分钟；这跟系统有关而不是 HMI 管理器的故障。

软件包“PCU 基本软件薄型客户”用来驱动 SINUMERIK solution line 组中的 PCU 50 V2，必须安装在 PCU 上。在 PCU 50 V2 安装软件包“PCU 基础软件薄型客户”时，需要分别根据应用情况对附加内置 Ethernet 网卡的 Ethernet 接口进行设置（参见章节“配置网络运行（SINUMERIK powerline）”）。

文献：薄型客户配置开机调试手册

PCU 名称

系统交付时，生成一个唯一的计算机名称并可在以下路径读到：“开始”→“设置”→“控制面板”→“系统”，标签“计算机名称”。

操作系统的配置

出于安全的考虑，对 Windows XP 进行了如下预设置：

- 取消自动运行功能。
- 取消 Windows 自动升级。
- 取消防病毒软件的监控和报警以及自动升级。
- 取消从服务桌面或者从开始菜单调用 Internet Explorer 的快捷图标。
- 对于未证实的调用可以进行远程程序调用 (RPC)。
- 在机载 Ethernet 网卡上激活防火墙设置。当插入额外的 Ethernet 网卡时，同样激活防火墙设置。只有通过安装 PCU 基础软件薄型客户，才能对第二张网卡取消防火墙设置。

在 Windows 通讯 (维修) 上的更改

其它的预设置：

Windows 通讯/维修:	调试类型:	
电脑浏览器	手动	(未启动)
错误报告维修	禁止使能	
便携式媒体序列号	手动	(未启动)
SSDP 发现维修	禁止使能	
通用插拔主机	禁止使能	
网络客户	手动	(未启动)
无线零点配置	手动	(未启动)

预定义用户

出厂交付系统时，定义了下列用户：

用户名	密码	用户类型	Windows 用户组
operator	operator	HMI 用户	操作员
auduser	SUNRISE	维修用户	系统管理员
siemens	*****	---	系统管理员

启动 ServiceCenter



使用服务桌面上的快捷图标启动 ServiceCenter Users。

在 ServiceCenter Users 中设置了所有用户通用的或单个用户独立的 PCU 启动特性。

1.2 管理用户

1.2.1 设定了哪些用户？

概述

用户管理在“ServiceCenter Users”中执行，以便调试人员/维修用户不须直接在注册表中进行相应设置。

已设定的用户有：

- 用户“auduser”

“auduser”

用户为维修用户类型并属于系统管理员用户组。系统管理员具有本地管理员的用户权限。

- 用户“operator”

“operator”用户为 HMI 用户类型并属于操作员用户组。操作员具有限制性用户权限。

对于 HMI 用户和维修用户可以单独进行启动，HMI 程序和维修设置。

注意事项

PCU 基础软件版本 7.6 只支持唯一的维修用户（预定义为“auduser”）和固定名称为“operator”的唯一 HMI 用户。

用户

每个用户都属于某一种用户类型和用户组。

用户类型

—用户可以为以下的用户类型：

用户类型	任务区域
HMI 用户	<ul style="list-style-type: none"> • PCU 启动 • HMI 程序操作
维修用户	<ul style="list-style-type: none"> • 维修任务 • PCU 启动 • HMI 程序操作

在 Windows 下用户类型划分为带有不同用户权限的用户组。

用户组

用户类型划分为以下 Windows 用户组：

用户组	组名 (Windows)	用户权限
操作员	operator.group	限制性
系统管理员	管理员	本地管理员

服务桌面

服务桌面向维修用户提供了一个 Windows 桌面，其扩展了工具和维修任务功能，例如用户管理，软件安装，数据备份/恢复，检验系统一致性等。

1.2.2 修改维修用户

概述

维修用户也可以 - 由 'auduser' 替换 - 成为在 Windows 下设置的任意其它本地管理员或者拥有本地管理员权限的域用户。

修改维修用户

维修用户在注册表中通过两项输入加以识别：用户名和用户类型。用户名预定义为“auduser”，维修用户的用户类型为“2”。

为了修改维修用户，必须在 Windows 下以本地管理员用户登录，就成为新的维修用户。只有该用户有足够的用户权限，能在注册表用户专用的分支下进行输入。

通过以下注册表信息，维修用户注册为：

关于			默认
密码		HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\<version>\HMIManager	
HMIUserName	注册	<user name> (STRING)	auduser
HMIUserDomainName	注册	<domain name> (STRING)	(local)

关于			默认
密码		HKCU\SOFTWARE\Siemens\SINUMERIK\Basesoftware\HMIManager	
UserType	注册	2 (DWORD)	

在“HMIUserDomainName”下输入下列名称：

- 对于一个本地管理员：(local)
- 对于一个域用户：<domain name>

1.2.3 用户全局设置

继承原则

在“ServiceCenter Users”中在“用户”→“全局设置”下设置以下属性：

- PCU 启动属性
- HMI 程序开始
- 维修运行中的属性

在“全局设置”下涉及到用户的设置，单个用户会加以继承。
所继承的设置仍可针对各个用户进行特定的匹配。因此用户必须以密码进行登录。

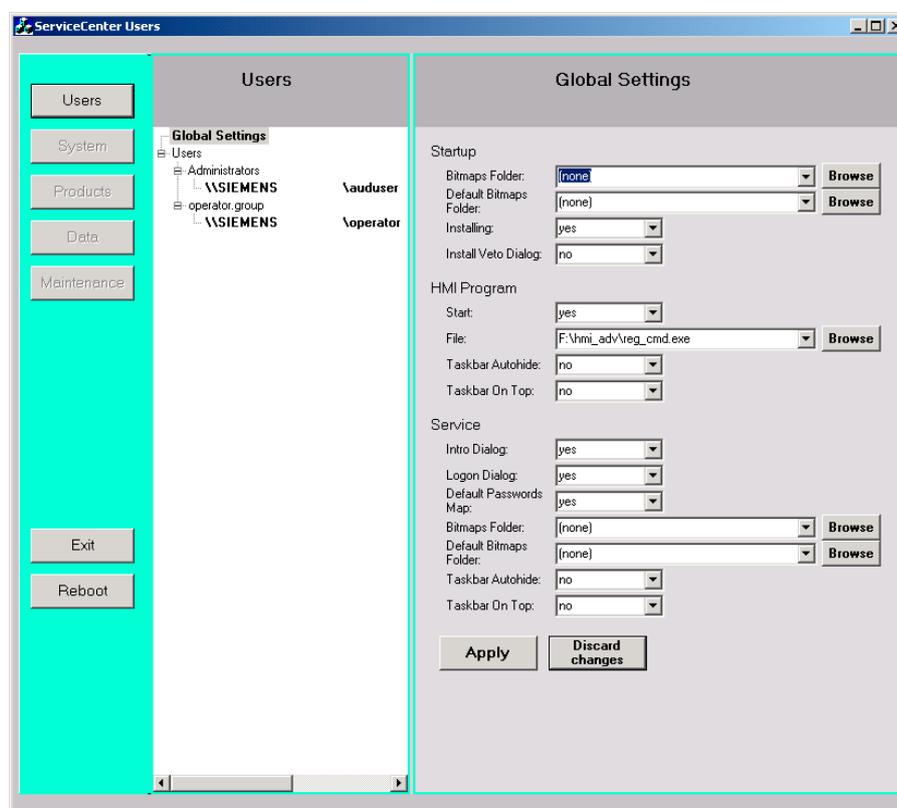


图 1-2 全局用户设置

启动：

	选择	结果
位图文件夹：	- 无 -	带有启动画面的目录
默认位图文件夹：	F:\hmi_adv\ib\DATA\default	带有默认启动画面的目录：
安装：	“是”（预设置）	启动时存在安装权限
	“否”	无安装权限
禁止安装对话：	“否”（预设置）	启动过程中进行安装时： 不询问，安装过程立即开始。

	选择	结果
	“是”	启动过程中进行安装时： 询问，是否要进行安装。

HMI 程序:

	选择	结果
开始：	“是” (预设置)	将开始 HMI 程序。
	“否”	将不会开始 HMI 程序。
文件：	F:\hmi_adv\reg_cmd.exe (预设置)	HMI 程序目录
自动隐藏任务栏：	“否” (预设置)	HMI 程序: 隐藏开始菜单
	“是”	HMI 程序: 显示开始菜单
任务栏置顶：	“否” (预设置)	HMI 程序: 开始菜单位于后台
	“是”	HMI 程序: 开始菜单在最前台

维修：

	选择	结果
介绍对话框：	“是” (预设置)	显示维修介绍对话框
	“否”	不显示维修介绍对话框
登录对话框：	“是” (预设置)	显示维修登录对话框
	“否”	不显示维修登录对话框
默认密码映射：	“是” (预设置)	密码可以用大写或小写字母输入。
	“否”	密码必须严格按设定输入。
位图文件夹：	- 无 -	维修桌面背景画面目录
默认位图文件夹：	- 无 -	维修桌面默认背景画面目录
自动隐藏任务栏：	“否” (预设置)	维修桌面：隐藏开始菜单
	“是”	维修桌面：显示开始菜单
任务栏置顶：	“否” (预设置)	维修桌面：开始菜单在最前台
	“是”	维修桌面：开始菜单位于后台

注意

当按下按钮“重新启动”时，会立即执行重新启动（无事先询问）。使用“退出”键退出 ServiceCenter Users。

启动中的维修对话框

启动时会显示以下对话框：

- 如果未安装 HMI 程序或者取消启动 HMI 程序，则显示**维修介绍对话框**。

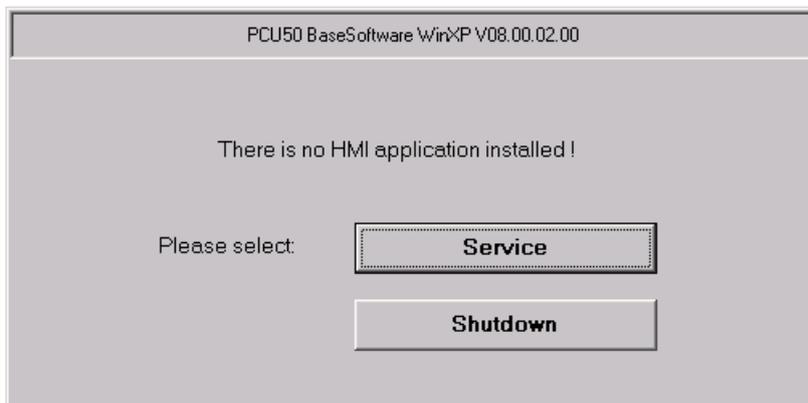


图 1-3 维修介绍对话框

- 当维修用户登录时，则显示**维修登录对话框**：
 - 在启动中按下按键 <3> 后。
 - 在维修介绍对话框中按下按钮“维修”后。



图 1-4 维修登录对话框

1.2.4 用户专用设置

继承设置

在“全局设置”下涉及到用户的设置，各个用户都会加以继承。

举例“operator”

输入密码后，可以对设置进行用户专用匹配。不可修改的设置以灰色背景显示。

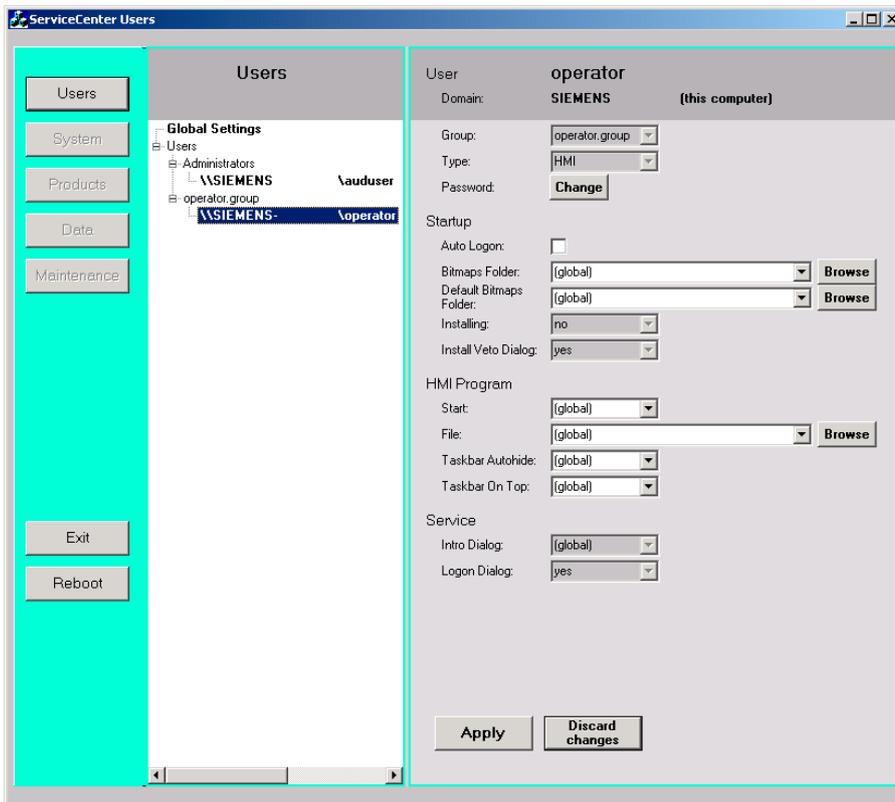


图 1-5 “operator”用户的设置

1.3 PCU 启动属性

1.3.1 PCU 启动：未安装 HMI 程序

前提条件

启动 PCU 时未安装 HMI 程序。

启动阶段

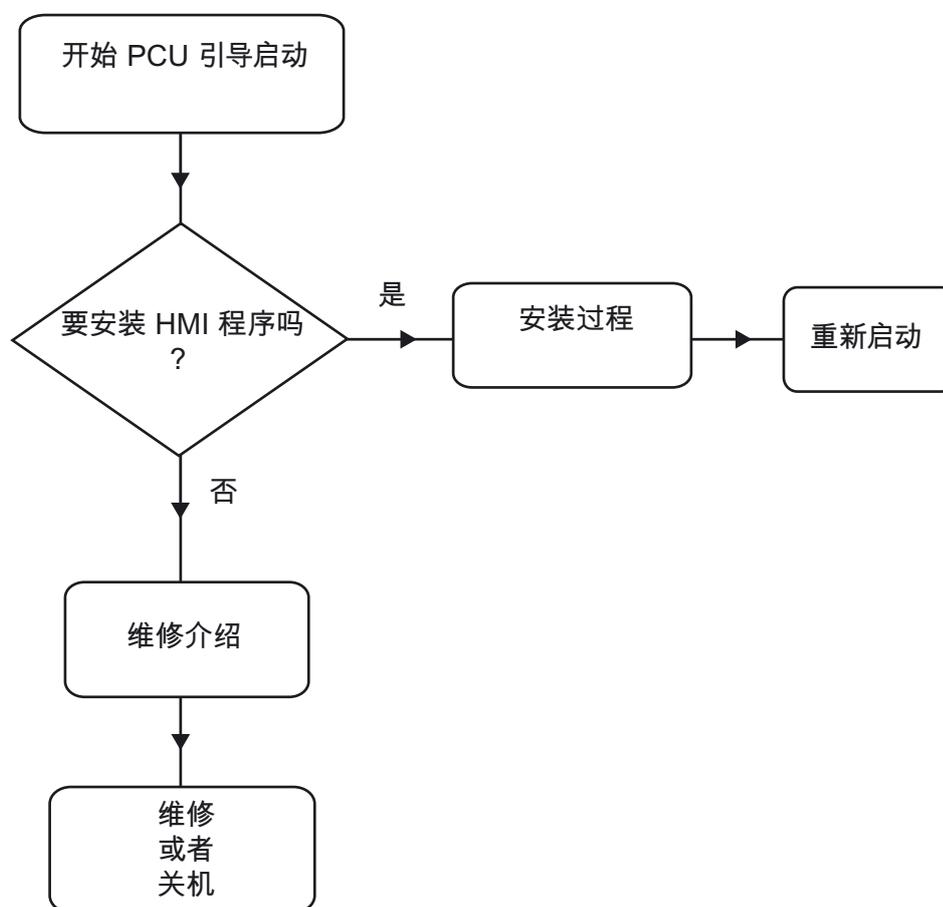


图 1-6 PCU 启动流程图 (无 HMI 程序)

说明：

- 如果在目录 D:\Install 下存在待安装的程序，则会在首次启动时询问，是否要启动安装过程。成功安装后必须重新启动。
安装也可以跳过，稍后再执行。

- 在维修介绍对话框中可以选择“维修”或“关机”。
- 选择“维修”打开维修登录对话框。

注意

PCU 50.3 首次启动时，用户只能以 'auduser' 登录。

1.3.2 PCU 启动：HMI 程序已安装

前提条件

已经安装了 HMI 程序，如 HMI 高级版。

启动阶段

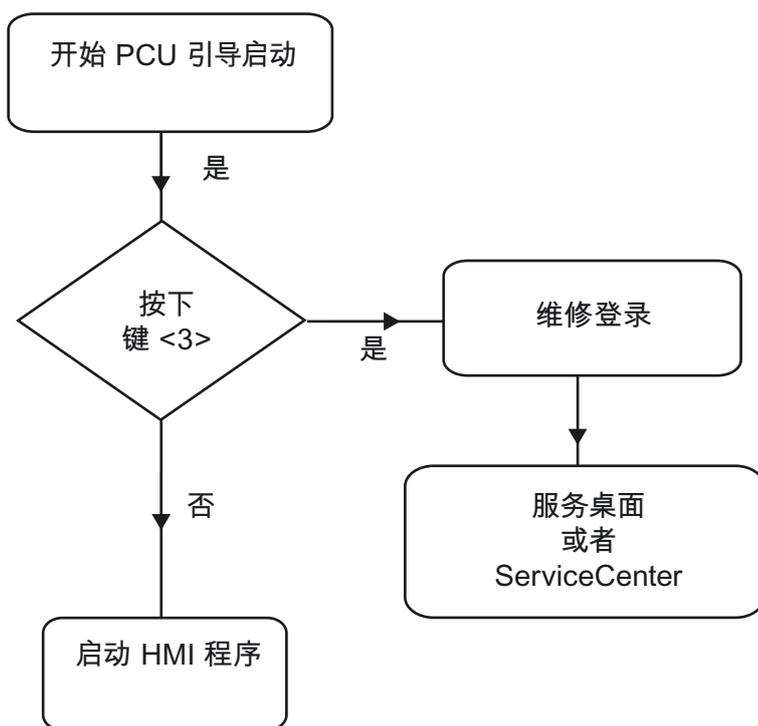


图 1-7 PCU 启动流程图 (已安装 HMI 程序)

说明：

- 如果已经安装了 HMI 程序，PCU 启动并且 HMI 程序开始运行（预设置）。
- 启动期间有一个时间间隔，用于当背景画面中出现版本信息时按下按键<3>：打开维修登录对话框。
- 以维修用户登录后，可以启动“维修桌面”，以执行其它维修任务，或者启动“ServiceCenter Users”，以设置各用户的启动属性。

1.4 BIOS 设置

概述

PCU 的 BIOS 预设置不必更改。日期和时间可以在 Windows 或者 HMI 操作面板下设置。

注意事项

设备配置用于随附软件的使用。

仅当在设备上已经进行了技术更改或者接通时出现故障时，才更改设置的值。

启动 BIOS-Setup

1. 按如下方式启动 BIOS-Setup：
复位设备（热启动或冷启动）。
引导启动测试后出现如下信息：
按 < F2 > 进入SETUP或<ESC> 显示引导启动菜单
2. 出现 BIOS 信息，按下按键 F2。

打开 BIOS 主菜单：

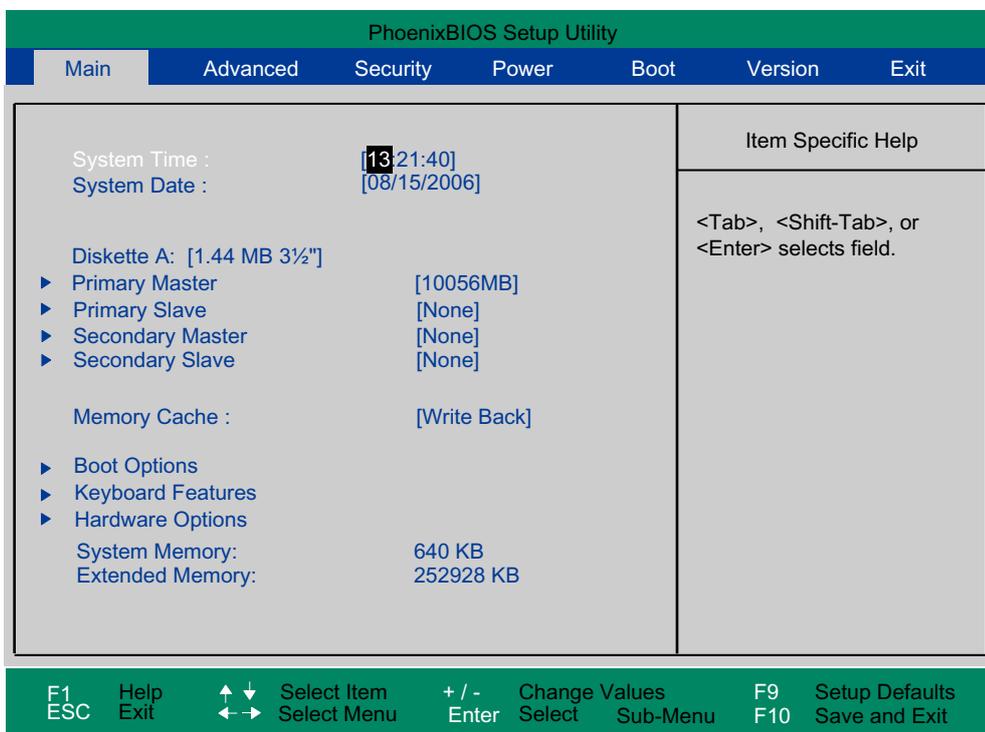


图 1-8 BIOS 主菜单 (举例)

BIOS-Setup：标准设置

交付 10 GB 硬盘的 PCU 时，保存有以下系统参数：

菜单:主机

系统参数	标准设置	专用登入
系统时间	小时:分:秒	
系统日期	月/日/年	
软盘 A	1.44 MB 3 1/2"	
初级主盘	10056 MB	
初级副盘	无	
次级主盘	无	
次级副盘	无	
高速缓冲存储器	回写	

引导启动选项		
快速引导模式	已使能	
SETUP 快速	已使能	
POST 错误	已使能	

引导启动选项		
软盘检查	禁止使能	
诊断屏幕	已使能	

键盘功能		
NumLock	关	
键点击	禁止使能	
键盘自动重复率	30/秒钟	
键盘自动重复延迟	½ sec	

硬件选项		
PCI - MPI / DP	已使能	
机载 Ethernet	已使能	
Ethernet 地址	08 00 06 xx xx xx	
LAN 远程引导启动	已使能	
CARDBUS 总线/ PCMCIA 槽	已使能	
扇区控制	已使能	
安全卡功能	已使能	
Legacy USB 支持	已使能	
CRT/LCD 选择	SIMULTAN (同时)	
LCD 屏幕尺寸	扩展的	
DSTN 对比度	154	
PS/2 鼠标	自动检测	

菜单:高级配置

系统参数	标准设置	专用登入
COM/LPT1 配置		
内部 COM 1	已使能	
基本 I/O 地址	3F8	
中断	IRQ 4	
内部 COM 2	已使能	
基本 I/O 地址	2F8	
中断	IRQ 3	
内部 LPT 1	已使能	
Mode	EPP / 双向的	

PCI 配置		
PCI 设备槽 1		
选项 ROM 扫描	已使能	
使能主盘	已使能	
等待定时器	0040h	

PCI 配置		
PCI 设备槽 2		
选项 ROM 扫描	已使能	
使能主盘	已使能	
等待定时器	0040h	
PCI 设备槽 3		
选项 ROM 扫描	已使能	
使能主盘	已使能	
等待定时器	0040h	
PCI 设备槽 4		
选项 ROM 扫描	已使能	
使能主盘	已使能	
等待定时器	0040h	
Ethernet		
选项 ROM 扫描	禁止使能	
使能主盘	禁止使能	
等待定时器	0040h	

系统参数	标准设置	专用登入
PCI/PnP ISA IRQ 资源专用		
IRQ 3	可用	
IRQ 4	可用	
IRQ 5	可用	
IRQ 7	可用	
IRQ 9	可用	
IRQ 10	可用	
IRQ 11	已保存	
PCI IRQ 线 1	5	
PCI IRQ 线 2	自动选择	
PCI IRQ 线 3	自动选择	
PCI IRQ 线 4	自动选择	
安装的 OS	其它	
复位配置数据	否	
软盘控制器	已使能	
本地 Bus IDE 适配器	两者都	
大磁盘存取模式	DOS	
存储间隔 15 MB	禁止使能	
视频适配器设置		
框架缓存器尺寸	8 Mb	
AGP 率	4X	
缺省初级视频适配器	AGP	

菜单:安全

系统参数	标准设置	专用登入
主管口令为	禁止使能	
用户口令为	禁止使能	
设置用户口令	回车	
设置主管口令	回车	
口令引导中	禁止使能	
固定盘引导区	标准	
软盘存取	主管	

菜单:电源

系统参数	标准设置	专用登入
APM	已使能	
节电	禁止使能	
待机超时	关	
硬盘超时	禁止使能	

菜单:引导

系统参数	标准设置	专用登入
<input type="checkbox"/> 可移动设备		
<input type="checkbox"/> Legacy 软盘驱动器		
<input type="checkbox"/> 硬盘驱动器		
<input type="checkbox"/> 备选设备 TOSHIBA MK1016 GAP-(PM)		
CD ROM 驱动 IBA 4.0.22 槽 0048		
MBA UNDI (Bus 总线 0 槽 19)		

菜单:版本

系统参数	标准设置	
SINUMERIK	SINUMERIK PCU 50/70	
BIOS 版本	V02.03.06	
BIOS 编号	A5E00124434-ES03	
主板修改	V02	
MPI/DP 固件	V01	
Ethernet 配置版本	V01	
CPU 类型	赛扬 (TM)	
CPU 速度	1200 MHz	
CPU ID	06B1	
代码修订	001C	

菜单:退出

Exit Saving Changes	退出时保存更改
Exit Discarding Changes	退出时不保存更改
Load Setup Defaults	载入缺省设置
Discard Changes	不保存更改
Save Changes	保存更改

更改 BIOS 设置

在装入或者添加额外的组件后，必要的话，必须将这一情况告知在 BIOS 启动中的系统：

1. 启动设备。
2. 按照激活 BIOS 启动的要求按下 <F2> 键（适合操作面板上的水平软键 2）。
3. 出现 BIOS-Setup 菜单。用光标键在菜单中导航至所需的选择区域。
4. 使用 <+>（同时按下 <SHIFT> 和 <X>）或使用位于数字区的按键 ↔ 来更改设置。
5. 可以用向右/向左光标键达到其它设置菜单。
6. 按下 <ESC>（<取消报警> 键），以到达菜单“退出”（也可以通过继续按向右光标键来实现）。
7. 按下 <输入> 键，离开设置菜单。

随后启动系统。

注意

更改 BIOS 设置时 - 引导顺序除外 - 需要签订 OEM 合同。

配置系统

2.1 服务菜单的功能（DOS-功能）

功能概览

服务菜单提供：

- HMI 系统软件的安装和升级功能
- 保存和重建数据功能

服务菜单可以提供以下功能：

基本菜单	功能
1	安装/升级 SINUMERIK 系统：SINUMERIK 系统的安装和补充或者更新
1	从软盘安装
2	通过串口/并口安装：串口/并口连接
3	从网络驱动器安装网络连接
1	管理网络驱动器
1	连接到网络驱动器，输入连接数据
2	显示连接的网络驱动器，显示连接
3	断开所有网络驱动器，断开所有连接
4	改变网络设置 更改连接设置
1	更改机床名称（仅用于 DOS 网络）
2	更改用户名称
3	触发记录
4	触发登录到域（是或否）
[5]	更改域名
[6]	更改 TCP/IP 设置
1	触发“获取 IP 地址”（自动 / 手动）
2	更改 IP 地址
3	更改子网掩码
4	更改网关
5	域名服务器
6	更改 DNS 扩展
9	返回上一级菜单
9	返回上一级菜单
9	返回上一级菜单

配置系统

2.1 服务菜单的功能 (DOS-功能)

基本菜单	功能
	4 更改安装目录
	[5] 从 (网络驱动器) 安装, 如果已经设置, 由网络安装
	9 返回上一级菜单
5	重新引导启动: 重启系统
9	返回主菜单: 返回基本菜单
2	SINUMERIK 工具和选项: 加载附加工具并释放选项。
1	激活的 STEP7 用于 PCU
9	返回主菜单: 返回基本菜单
3	DOS Shell: 调用 DOS 的指令解释程序, 显示驱动器 C:\ ; 用 EXIT 退出 DOS Shell。
4	启动 Windows: 这里可以使用的 Windows 功能
1	标准 Windows (不启动 SINUMERIK HMI)
4	下次引导启动时, 启动初始 SINUMERIK HMI 环境。
5	下次引导启动时, 启动当前 SINUMERIK HMI 环境。
9	返回主菜单: 返回基本菜单
5	检查 SINUMERIK 系统: 测试文件系统的一致性, 必要的话, 用 SCANDISK 修复一致性。
7	备份/修复:
1	用 GHOST 备份/修复硬盘
	1 配置 GHOST 参数: 设置用于 Norton Ghost 的参数 (接口、路径、网络连接)
	1 设置连接方式并口 (LPT:)
	2 设置连接方式本地/网络
	3 更改备份图像文件名称
	4 更改修复图像文件名称
	5 更改机床名称 (用于 Windows 和 DOS 网)
	1 手动输入机床名称
	2 随机生成机床名称
	9 返回上一级菜单
	6 管理网络驱动器
	7 更改拆分方式
	1 不拆分
	2 拆分中
	9 返回上一级菜单
	2 硬盘备份: 备份硬盘
	3 恢复硬盘: 恢复硬盘数据存储
	1 Windows XP
	1 标准分区 (缺省)
	2 用户定义的分区
	3 图形定义的分区
	2 Windows NT/Windows 2000 (子菜单如 Windows XP)
	3 Windows 95 (子菜单如 Windows XP)
	4 WfW3.11 (子菜单如 Windows XP)
	5 DOS (备件) (子菜单如 Windows XP)
	9 返回上一级菜单

基本菜单	功能
	4 切换到其它 GHOST 版本
	9 返回上一级菜单：返回到前一级菜单
4	用 GHOST 备份/修复区：C:, E:, F: 区的内容：向 D:保存或者从 D:重新返回到 C:, E:, F:设置。
	1 配置 GHOST 参数
	1 更改最大备份图像
	2 定义紧急备份图像
	9 返回上一级菜单
	2 分区备份，本地模式
	3 分区修复，本地模式
	4 E 区 (Windows) 修复，本地模式
	8 删除图像
	9 返回上一级菜单
	9 返回主菜单：返回基本菜单
8	启动 PC 链接: 通过 CD-ROM 安装软件 (内部链接/内部服务)
9	重新引导启动: 重启系统

2.1.1 服务菜单 (DOS) 参数配置

配置服务菜单参数

服务菜单中提供的点可以如下设置参数：

- [a] 提供/抑制
- [b] 考虑专用产品名称 (替代 SINUMERIK)
- [c] 更改文本

目录 C:\ 中，必须创建文件 "envparam.bat" 而且设置参数时必须用下列指令：

注册编号	DOS 指令	文本/功能，如果 =TRUE	文本/功能，如果 =FALSE	预置值
1	[a] set_INSTALL	1 安装/升级 SINUMERIK 系统	没有显示	TRUE
2	[a] set_TOOLS	2 SINUMERIK 工具和选项	没有显示	TRUE
4	[a] set_SERVICE	4 启动 Windows (服务模式)	没有显示	TRUE
6	[a] set_REBOOT	6 重启系统 (热启动)	没有显示	FALSE
	[a] set_HMI	用 HMI 功能显示子菜单	没有显示	TRUE

2.1 服务菜单的功能 (DOS-功能)

注册编号	DOS 指令	文本/功能, 如果 =TRUE	文本/功能, 如果 =FALSE	预置值
字符串注册信息:				
9	[c]	set_ENDTXT	=字符串	重新启动 (热启动)
	[b]	set_PRODUCTLINE	=字符串	SINUMERIK

调用恢复控制台

在引导启动管理器中启动时, 选择菜单项“SINUMERIK”下的 第三个不可见的菜单项。

出现: 启动 Windows 恢复控制台.

在原始 MS 文献中可以找到相关详细信息。

2.1.2 基本菜单操作

基本菜单操作

为了调用服务菜单:

1. 在引导启动管理器中启动时, 选择菜单项“SINUMERIK”下的 第二个不可见的菜单项。
显示基本菜单。

请选择

- 1 安装/升级 SINUMERIK 系统
- 2 SINUMERIK 工具和选项
- 3 DOS Shell
- 4 启动 Windows (服务模式)
- 5 检查 SINUMERIK 系统
- 7 备份/修复
- 8 启动 PC 链接
- 9 重启 (热启动)

您选择 [1, 2, 3, 4, 5, 7, 8, 9]?

2. 选择一个菜单项, 例如: 键 <7> 用于备份/修复
系统要求输入口令。
3. 输入 0 到 2 级的口令。
 - 系统
 - 厂商
 - 维修

2.1.3 设置初始 SINUMERIK HMI 环境

建立初始 HMI 环境

将设置 HMI 环境的初始交付状态。为此要保存下列目录内容：

- ADD_ON
- 用户
- OEM

随即清空这些目录。

步骤

PCU 引导启动期间（在关闭和打开控制系统之后）：

1. 在引导启动管理器中选择服务菜单 [不可见的菜单项]。

显示基本菜单。

2. 按下 <4> 键。启动 Windows (服务模式)

显示下列菜单：

请选择

- 1 标准 Windows (不启动 SINUMERIK HMI)
- X 4 初始 SINUMERIK HMI 环境
- 5 当前 SINUMERIK HMI 环境

9 返回主菜单

您选择 [1, 4, 5, 9]?

3. 按下 <4> 键。

系统要求输入口令。

4. 输入 0 到 2 级的口令。

结果：

接着进行自动重启：Windows 进行下列引导启动时，该功能的执行要在启动 HMI 系统软件之前。

注意

也可以在服务桌面上通过启动功能初始 SINUMERIK HMI 环境来设置 HMI 环境的初始交付状态。

建立当前 HMI 环境

将再次取消 HMI 环境的初始交付状态设置，即：退回目前保存的目录内容。

步骤

PCU 引导启动期间（在关闭和打开控制系统之后）：

1. 在引导启动管理器中选择服务菜单 [不可见的菜单项]。

显示基本菜单：

2. 按下 <4> 键。启动 Windows (服务模式)

显示下列菜单：

```
          请选择
          1 标准 Windows (不启动 SINUMERIK HMI )
          4 初始 SINUMERIK HMI 环境
X        5 当前 SINUMERIK HMI 环境

          9 返回主菜单
```

您选择 [1, 4, 5, 9]?

3. 按下 <5> 键。

系统要求输入口令用：

4. 输入 0 到 2 级的口令。

结果：

接着进行自动重启：Windows 进行下列引导启动时，该功能的执行要在启动 HMI 系统软件之前。

注意

也可以在服务桌面上通过启动功能当前 SINUMERIK HMI 环境来再次取消 HMI 环境初始交付状态的设置。

2.1.4 启动 DOS Shell 并检查软件版本

启动 DOS Shell

PCU 引导启动期间 (在关闭和打开控制系统之后) :

1. 在引导启动管理器中选择服务菜单 [不可见的菜单项]。
2. 按下键 <3> : DOS Shell
系统要求输入口令。
3. 输入 0 到 2 级的口令。

结果 : 它只位于 DOS Shell。

退出 DOS Shell

通过输入 "EXIT (退出)" 退出 DOS Shell。

检查软件版本

文件 C:\BaseVers.txt 中列出了 PCU 基础软件中所包含的服务菜单的以及 Windows XP 的系统组件的版本。

步骤

1. 启动 DOS Shell。
2. 输入: 类型 BaseVers.txt
结果 :
列出了版本信息。
3. 通过输入 "EXIT (退出)" 退出 DOS Shell。

结果 :

引导启动画面中 , PCU 基础软件的版本显示在右下方。

也可以选择在服务桌面上激活 HMI-Explorer。

2.1.5 启动服务桌面

目的

下次引导启动时，服务桌面 (Windows) 代替 HMI 系统软件启动。

步骤

PCU 引导启动期间 (在关闭和打开控制系统之后)：

1. 在引导启动管理器中选择服务菜单 [不可见的菜单项]。

显示基本菜单。

2. 按下键 <4>：启动 Windows (服务模式)

显示下个菜单：

请选择

- X 1 标准 Windows (不启动 SINUMERIK HMI)
- 4 初始 SINUMERIK HMI 环境
- 5 当前 SINUMERIK HMI 环境

9 返回主菜单

您选择 [1, 4, 5, 9]?

3. 按下键 <1>：

结果：

接着进行自动重启：替代 HMI 系统软件启动服务桌面 (Windows)。

注意

当启动中在背景图像中出现 PCU 基础软件的版本说明时按下键 <3>，也可以启动服务桌面。(在3秒的时间窗口内)

2.1.6 检查硬盘

目的

检查硬盘上的所有分区 (C:, D:, E:, F)。

步骤

PCU 引导启动期间 (在关闭和打开控制系统之后)

1. 在引导启动管理器中选择服务菜单 [不可见的菜单项]
显示基本菜单。
2. 按下键 <5> : 检查 SINUMERIK 系统

结果 :

接着进行自动重启 : Windows 进行下列引导启动时 , 该功能的执行要在启动 HMI 系统软件之前。

注意

硬盘检查也可以在服务桌面上通过启动功能检查 SINUMERIK 系统来进行。

2.2 配置操作界面

2.2.1 HMI 启动画面的保存位置

预设置

西门子引导启动画面在各自的 PCU 上 , 如 : 对于控制系统 SINUMERIK 840D , 保存在如下路径 :

F:\hmi_adv\ib\DATA\0\0\<>Auflösung>\0_0.bmp

可以为制造商专用的启动画面创建目录树 , 制造商画面可以按照下列框架保存在其中 :

制造商专用启动画面目录树 :

F:\oem\ib\DATA\<>NckType>\>Auflösung>\>Name>.bmp

<Nck 类型>	适用于 :
0	840D
2000	810D
5000	840Di

<分辨率>:640 或 800 或 1024 dpi

如果总是要显示一个固定画面 (与 NCU 无关的画面) , 可以在缺省目录下按要求的分辨率保存该画面。如果对于不同的 NCU 要显示不同的画面 , 在子目录<Nck 类型>\<分辨率>下按相应的分辨率保存这些画面。

画面名称和分辨率

<名称>.bmp: 可以自由选择名称 ; 每个目录下只允许有一个文件。必须使用图形工具以子目录名称给出的分辨率创建画面 , 并保存在相应的目录中。HMI 软件根据 NCK 类型和现有操作设备分辨率选出画面。

2.2.2 显示自定义启动画面

前提条件

建立一个带有多个启动画面 (对于不同的面板分辨率) 的目录。为此 , 该目录划分为子目录 640、800 和 1024 , 在其中包含了相应分辨率的各引导启动画面。

此外 , 还可以建立带有缺省启动画面的目录 , 同样有子目录 640、800 和 1024。如果在上述引导启动画面目录中没有找到启动画面 (较低分辨率的也未找到) , 则显示子目录中保存的特定分辨率的画面。

如果没有发现启动画面 (较低分辨率的也未找到) , 则显示通用的启动画面 , 它是 PCU 基础软件的组成部分。

目录

目录可以在 ServiceCenter Users 中设置为用户专用的 :

- 启动 : 位图文件夹
- 启动 : 默认位图文件夹

2.2.3 更改维修桌面背景

概述

维修桌面的背景样式或者背景画面不是通过“控制面板”（系统控制）设置，而是在注册表中设置：

- 背景样式：

密码：	HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\ <version>\HMIDesktop
值：	图案 (STRING)
日期：	<位代码>" (例如 "0 80 114 32 0 5 39 2", 参见HKCU\Control Panel\Patterns) “(无)” (= 无背景样式)
初始日期：	迄今为止通过系统控制设置的背景样式
缺省日期：	“(无)”(当不存在注册信息或者不可读时)

- 背景画面：

维修桌面的背景画面不是通过“控制面板”（系统控制）设置，而是在 ServiceCenter Users 中通过“维修: 位图文件夹”或“维修: 默认位图文件夹”进行用户自定义设置。

注意

尝试通过系统控制来设置维修桌面背景 - 如同在标准 Windows 中 - 仅影响启动画面的显示，然而不影响维修桌面背景的显示。

2.3 启动时用户专用设置

概述

引导启动后立即启动 HMI 程序，如 HMI 高级版，在设置和功能上与引导启动后立即启动维修桌面不同。

2.3.1 启动 HMI 程序

过滤键

引导启动 HMI 程序时，过滤键顺序和同时按下的功能键。在文件 E:\Windows\System.ini 中可以设计待过滤的键顺序和同时按下的功能键。

过滤键顺序：

段：	MMC103Keyb
密码：	SeqAct
值：	<位掩码> (= 待过滤的键顺序，根据 E:\Windows\System.ini 中的注释)
初值：	262143

过滤同时按下的功能键：

段：	MMC103Keyb
密码：	ConcurrentKeyMask
值：	<位掩码> (= 待过滤的功能键，根据 E:\Windows\System.ini 中的注释)
初值：	255

启动附加程序

这些程序可以与 HMI 程序并行启动，维修桌面打开时 Windows 自动运行。可以设定与 HMI 程序并行启动。

如果待运行的程序在 Windows 目录 E:\Documents and Settings 下，则要设置以下注册表信息：

密码：	HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\ <version>\HMIManager
值：	StartSINHMIStartupDirsPrograms (DWORD)
日期：	1 (程序被启动) 或者 0 (程序不被启动)
初始日期：	0
缺省日期	0 (如果注册信息不存在或者不可读)

若待启动的程序设置了注册信息 'HKCU\Software\Microsoft\Windows\CurrentVersion\Run' 和 'HKLM\Software\Microsoft\Windows\CurrentVersion\Run'，则要设置以下注册信息：

密码：	HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\ <version>\HMIManager
值：	StartSINHMIrunPrograms (DWORD)
日期：	1 (程序被启动) 或者 0 (程序不被启动)
初始日期：	0
缺省日期：	0 (如果注册信息不存在或者不可读)

执行 HMI 程序

在 HMI 程序执行过程中，运行一个不可见的 Windows Explorer，通过其设置（禁用的浏览器功能、任务栏、开始菜单等等）阻止在 Windows 平台上不希望的访问。

任务栏属性可以在 ServiceCenter Users 中通过菜单“HMI 程序”用户自定义设置“自动隐藏任务栏”和“任务栏置顶”。

结束 HMI 程序

结束 HMI 程序时，Windows XP 完全关闭。

2.3.2 启动时运行维修桌面

过滤键

对键顺序和同时按下的功能键的过滤取决于文件中的设计。E:\Windows\System.ini

过滤键顺序：

段：	MMC103Keyb
密码：	KeySequencesEnable
值：	1 (= 过滤在密码 SeqAct 中特有的键顺序) 或者 0 (=无过滤器)
初值：	0
缺省值：	0 (如果注册信息不存在或者不可读)：

密码：	SeqAct
值：	<位掩码> (= 待过滤的键顺序，根据 E:\Windows\System.ini 中的注释)
初值：	262143

过滤同时按下的功能键：

段：	MMC103Keyb
密码：	ConcurrentKeyEnable
值：	1 (= 过滤在密码 ConcurrentKeyMask 中特有的功能键) 或者 0 (=无过滤器)
初值：	1
缺省值：	0 (如果注册信息不存在或者不可读)

密码：	ConcurrentKeyMask
值：	<位掩码> (= 待过滤的功能键，根据 E:\Windows\System.ini 中的注释)
初值：	255

启动维修桌面

通过维修桌面可以自由进入 Windows 平台。
启动维修桌面时将运行所有程序，在登录时已经自动以默认运行方式启动 Windows。

执行 HMI 程序

也可以从维修桌面启动 HMI 程序。

结束 HMI 程序

结束一个从维修桌面启动的 HMI 程序时，将重新回到维修桌面。

保存维修桌面 (预设置)

注销时，不保存维修桌面上的设置 (例如：维修桌面上的链接命令)。
维修人员要在维修桌面上始终找到同一个输出状态，而不是前一次会议的设置。

保存维修桌面设置

可以通过注册表中的某一注册项来更改这种属性。通过注册表中的该项保存下列设置：

- 打开窗口的位置
- 任务栏的大小和位置
- 移动和删除链接

密码：	HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\ <version>HMIManager
值：	SINDesktopSettings (DWORD)
日期：	1 (保存设置) 或者 0 (不保存设置)
初始日期：	- 值不由基础软件创建 -
缺省日期：	0 (如果注册信息不存在或者不可读)

该密码对所有维修用户和所有其它用户都有效。

与注册信息无关，始终保存维修桌面上的链接。

注意

在退出维修桌面前，在“保存设置”的设置下必须关闭注销前的仍然打开的应用程序窗口。否则重新启动时，在 HMI 程序开始前，上述应用程序窗口会直接短时出现并再次关闭。

2.3.3 启动 OEM 程序

概述

直接在 HMI 系统软件启动前，可以启动 OEM 程序。为此，这些程序或其链接必须保存在目录 C:\RunOEM 的子目录中。

开始时的顺序

以列出的顺序来执行子目录。一个子目录中的程序按照它在子目录中的创建时间顺序启动。

- 在子目录中 C:\RunOEM\SeqOnce
程序**一次性并按顺序**启动，即：当前一启动程序结束后才启动后一程序。
- 在子目录中 C:\RunOEM\Seq
程序在**每次引导启动时，按顺序启动**，即：当前一启动程序结束后才启动后一程序。
- 在子目录 C:\RunOEM\ParOnce 中程序**一次性同时启动**。其与 HMI 系统软件并行运行。
- 在子目录 C:\RunOEM\Par 中程序在**每次引导启动时同时启动**。其与 HMI 系统软件并行运行。

除程序文件外，也可以在子目录中保存其它类型的文件，按相应的文件类型打开。

例如：".txt" 型文件用记事本打开，".htm"类型文件用 Internet Explorer 打开。

2.3.4 引导启动时激活/取消激活故障记录

应用

每次引导启动都会在文件 D:\\$\$Base.log 中描述一个信息程序段，包含有日期、时间和管理干预指令。

关于 HMI 管理器控制的引导启动信息可能会显示在屏幕上，并写入日志文件 D:\\$\$Base.log。

日志文件中输出

通过注册表/值控制输出：

- 信息方式

'HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\<<version>\HMIManager\InfoLevel' 设置要输出何种信息方式：

信息等级(DWORD)

- = 1: 输出必需的信息 (预设置)
- = 2: 输出必需的信息和补充信息
- = 3: 输出必需的信息、补充和跟踪信息
(信息等级 <= 0 处理方式同信息等级 == 1;
(信息等级 > 3 处理方式同信息等级 == 3)

- 输出屏幕信息

'HKLM\SOFTWARE\Siemens\SINUMERIK\Basesoftware\<<version>\HMIManager\ShowInfo' 设置，是否信息 - 除了输出日志文件中外 - 也要在屏幕上显示：

显示信息 (DWORD)

- = 0: 显示必需的信息，
没有显示补充信息和跟踪信息
- = 1: 显示必需信息、补充和跟踪信息 (预设置)

2.4 PCU 带有 SITOP USV 模块

应用

当 PCU50/70 上的电源出现扰动时，下列的 SITOP USV 模块可以通过使用缓冲蓄电池在有限时间内维持运行，而且当电池耗尽时使 PCU 有序停止：

名称	接口	订货号
6A 模块	USB	6EP1931-2DC41
15A 模块	USB	6EP1931-2EC41
6A 模块	串行	6EP1931-2DC31
15A 模块	串行	6EP1931-2EC31

注意

带有 USB 接口的 USV 模块只能在带有 PCU 基础软件 Windows XP 的 PCU 50 上使用。

前提条件

- 高于 07.03.02 版本的 PCU 基础软件 Windows XP 用于串行接口或者高于 07.04.00 版本的 PCU 基础软件 Windows XP 用于 USB 接口
- 高于 06.03.14 版本的 HMI 高级
- 安装了高于 2.5.2.4 版本的 SITOP-SW:
SITOP-SW 可以用于下载：www.ad.siemens.de/sitop
- 安装了 SITOP 监视器/配置程序：
为此必须把 SITOP 软件复制到 PCU 50 上准备好的目录 E:\SITOP 下。在该文件夹下已经有 PCU 50 的工具，用于 SITOP USV 的关闭。如果在 PCU 基础软件的旧版本中没有该目录，则必须创建一个，以便以后升级 PCU 基础软件时可以一致升级。
- 为 Windows XP 安装了 USV-USB 驱动器：
在所附的 SITOP 文献中对安装作了描述。文献位于 SITOP 软件的下载包中。
- 连接了 SITOP-USV 的硬件。

测试环境

在 HMI 高级的标准配置中测试“SITOP USV”功能。安装附加软件组件和 OEM 软件组件时，必须由用户对整个系统的关闭进行测试。

2.4.1 启动并配置 SITOP 显示器

Windows 启动

在启动 Windows 时必须自动启动 SITOP 显示器，因此对于在 Windows 注册数据库中的 SITOP 显示器必须在下列密码下输入一个新值：

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run  
    值名称： SITOP  
    值类型： REG_SZ  
    值数据： E:\SITOP\SITOP_DC_USV.exe
```

在文件夹 E:\SITOP 中有一个脚本文件 sitop.reg。
通过执行该文件，将在注册数据库中自动输入所需的密码。

重新启动 PCU 后，自动启动 SITOP 显示器。安装的下一步是配置显示器。

注意事项

必须避免通过 Windows 自动启动目录来启动 SITOP 显示器。

一般设置

在 SITOP 显示器的配置对话框中必须执行下列设置：

- 设置接口参数：对于带有 USB 接口的模块 USB
- 设置更改行动的参数：
必须取消监控窗口的显示，因为该功能会导致 HMI 操作面板偶尔出现故障。

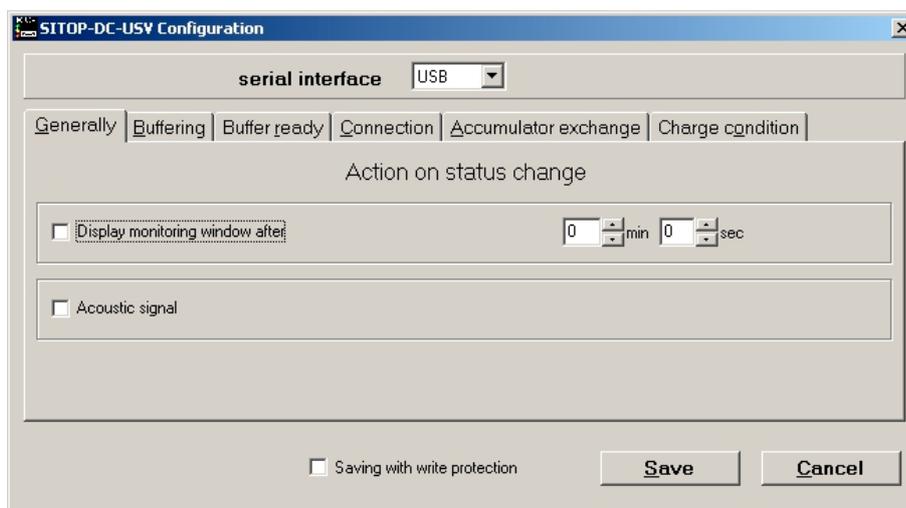


图 2-1 配置 SITOP 显示器的参数：一般设置

配置 SITOP 显示器的参数：

在参数区域内，为缓冲存储器输入用于断电时 HMI 高级版有序停止和 PCU 关闭的程序路径：
E:\SITOP\shutdown.bat

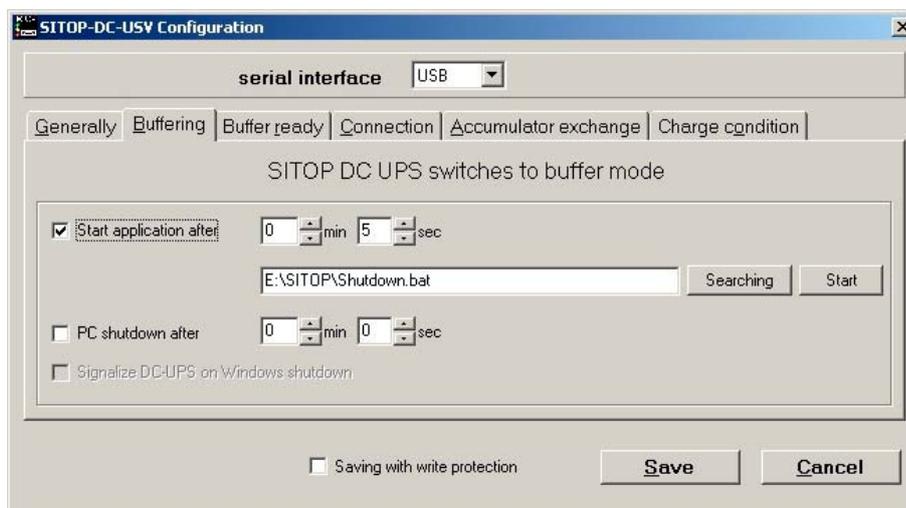


图 2-2 配置 SITOP 显示器的参数：设置缓冲状态下的运行

也可以使用以下设置：E:\Windows\system32\hmiexit.exe

2.4.2 SITOP USV 模块配置

为缓冲状态下的运行设置参数

基本上可以在 USV 模块上选择，是否在规定的时间内或者在达到蓄电池深度放电域值 (=最大缓冲时间) 时，结束缓冲运行。由此得出两个用于缓冲运行的参数设置。

运行方式“最大缓冲时间”(高于 V07.04.00 的 PCU 基础软件 Windows XP)

在该运行方式下，可以适时对系统进行暂时关闭。对此，USV 模块与操作系统的关闭同步。一直保持缓冲运行，直至操作系统关闭。操作系统的关闭最多允许 5 分钟 (包括所有应用程序)。另外，USV 模块以最大缓冲时间 (取决于蓄电池状态) 缓冲。

在 USV 模块上的必要设置 (USB 接口)

		开 - 关		
1			+2V	接通阈值 (cut-in threshold) +22V 固定设置的 (fixed)
2			+1V	
3			+0.5V	
4			+1V	结束充电电压(end-of-charge voltage) + 26.3V 固定设置的 (fixed)
5			+1V	
6			+0.5V	
7			+0.2V	
8			+0.2V	
9			+0.1V	
10			0.35A / 0.7A	充电电流(charging current)

		开 - 关		
1				设定时间/最大时间(set time / max.)
2			+320s	缓冲时间 +5s 固定设置的 (fixed)
3			+160s	
4			+80s	
5			+40s	
6			+20s	
7			+10s	
8				中断输出 (disconnection)
9				运行状态 蓄电池 开/关(operating state battery)

图例：
 供货状态设置
 在 PCU 50 上运行所需的设置

在 USV 模块上的必要设置 (串行接口)

		开 - 关			
1			●	+2V	接通阈值 (cut-in threshold) +22V 固定设置的 (fixed)
2			●	+1V	
3	●			+0.5V	
4			●	+1V	° ° 结束充电电压(end-of-charge voltage) + 26.3V 固定设置的 (fixed)
5			●	+1V	
6	●			+0.5V	
7	●			+0.2V	
8			●	+0.2V	
9			●	+0.1V	
10			●	0.35A / 0.7A	

		开 - 关			
1	●				设定时间/最大时间(set time / max.)
2			●	+320s	° ° 缓冲时间
3			●	+160s	
4			●	+80s	
5			●	+40s	
6	●			+20s	
7	●			+10s	+5s 固定设置的 (fixed)
8	●				中断输出 (disconnection)
9			●		运行状态 蓄电池 开/关(operating state battery)

图例：

● 供货状态设置

● 在 PCU 50 上运行所需的设置

运行方式“固定缓冲时间”

在该运行方式下，USV 模块总是以预选的固定时段缓冲 USV 模块。USV 模块与操作系统关闭无法同步。

在 USV 模块上的必要设置

		开 - 关		
1			+2V	接通阈值 (cut-in threshold) +22V 固定设置的 (fixed)
2			+1V	
3			+5V	
4			+1V	° ° 结束充电电压(end-of-charge voltage) + 26.3V 固定设置的 (fixed)
5			+1V	
6			+0.5V	
7			+0.2V	
8			+0.2V	
9			+0.1V	
10			0.35A / 0.7A	

		开 - 关		
1				设定时间/最大时间(set time / max.)
2			+320s	° ° 缓冲时间 +5s 固定设置的 (fixed)
3			+160s	
4			+80s	
5			+40s	
6			+20s	
7			+10s	
8				
9				运行状态 蓄电池 开/关(operating state battery)

图例：
 供货状态设置
 在 PCU 50 上运行所需的设置

2.4.3 用于关闭 HMI 的配置

监控 HMI

通过专用应用程序 hmiexit.exe 对 HMI 高级版的结束进行监控。
该应用程序通过内含的批处理文件 shutdown.bat 启动。
发生故障时，该应用程序强制关闭操作系统。如果 HMI
不在设定的等待时间内结束，就出现故障。

可能在下列文件中为 HMI 退出随机设置下列参数：E:\SITOP\hmiexit.ini.

```
[Actions]
```

```
#等待时间以秒表示，用于结束 HMI 高级版应用程序
```

```
Wait = 120
```

```
#等待时间超时的措施
```

```
ForceShutdown = True
```

仅当在一个 OEM 安装程序中的 HMI 应用程序结束时间超过 120 秒，才要求更改预设置。
一般不对该设置进行更改。

退出键的设计

应当在生产操作中切断通过操作范围菜单 EXIT 键的 HMI 的关闭功能，因为该功能与 USV 模块不同步。

在文件 Regie.ini 中用注册信息 EixtButton=FALSE 关闭退出键。

睡眠模式

运行 USV 时，操作系统的睡眠模式禁用，因为 USV 模块的 USB 接口必须总是激活。

注意

进一步提示参见相应订货号的产品说明。

软件安装和网络运行

概述

本章描述了在预先安装了 PCU 基础软件 Windows XP 的基础上附加软件的安装与升级。

下面从硬件组件和软件组件的供货状态开始进行说明。 服务菜单和服务桌面是重要的工具。

3.1 软件的安装和升级

3.1.1 安装升级或服务包

PCU 基础软件 WinNT4.0 的升级和服务包提供两种方案：

- 安装目录：

所有安装所需的文件，以及其它 Setup.exe、cab 文件、Setup.ini 等都位于一个目录中。通过启动 Setup.exe 开始安装。

- 自行打开安装：

启动时，Setup.exe 首先自动在临时安装目录中打开，并通过启动那里的 Setup.exe 开始安装。

与提供的方案无关，升级和服务包可以两种方式运行：

- 独立安装：

安装程序可自行控制整个安装运行过程。

尤其，安装程序要自己负责在安装期间和/或安装后执行可能必要的重启。

通过控制文件 Setup.ini 来控制安装程序的运行过程。它一般包含在（临时）安装目录中。对于交互式的安装预设置了控制参数。

仅当在（临时）安装目录中找到控制文件时，安装程序才使用控制文件。

- 框架安装程序的子安装：

从一个框架安装程序开始启动安装，并且在状态更改、结果（如：成功/不成功）、结束、中断或在安装期间和/或安装后可能必要的重启的范围内向框架安装程序通报，不完全控制安装过程。框架安装程序负责执行相应的指令。

通过控制文件 OpFile.txt

来控制安装程序的运行过程。它一般包含在（临时）安装目录中。

对于“安静方式”的安装预设置了控制参数。

仅当调用安装程序时给出了调用参数 -OF，例如：Setup.exe -OF:.\OpFile.txt 时，才使用控制文件。控制文件可以位于任意目录。

注意

在 -OF: 和 .\OpFile.txt 之间不能有空格。

基于调用参数 -OF，安装程序识别到作为框架安装程序的子安装来运行。这种情况下，在其结束前把关于其它框架安装程序执行指令的信息保存在控制文件的结果参数中，例如：

[RESULTS]

安装成功 = 1

请求重启 = 1

框架安装程序负责执行重启。

注意

当从命令行开始用调用参数 -OF 调用安装程序时，操作者必须在结束后自己执行必要的指令，如：重启。

3.1.2 安装“安静模式”升级或服务包

适用于：PCU 50 带有 PCU 基础软件 WinNT4.0 升级版 V06.02.01

安装是独立安装或者自行打开安装

在安装程序 Setup.exe 的控制文件 Setup.ini 中，已经缺省设置了用于安静模式安装的相关控制参数，即：

[安装调试]

显示用户对话框 = 否：

将不显示用户对话框，如：欢迎或者许可证对话框。

显示重启对话框 = 否：

不显示最终对话框（重启 是/否）。

不过如果控制参数这样设置，在安装过程中还是会出现一个对话框，询问发现相同的 MPI 版本时是否安装新的 MPI。也可以借助 NagsAway.exe 工具在该对话框上探测，来阻止该对话框，使它出现时自动被确认。

必须在调用 Setup.exe 前启动 NagsAway.exe。需要配置文件 NagsAway.ini，它和 NagsAway.exe 位于同一目录并具有下列内容：

[通用]

时间间隔 = 100

[1]

Window 标题 = ... 检查安装的 MPI 版本

控制标题 = &是

WinSubCheck=0

CtlSubCheck=0

适用于：

PCU 50 带有 PCU 基础软件 WinNT4.0 升级版 V06.03.06 和

PCU 50 带有 PCU 基础软件 WinNT4.0 升级版 V07.03.05

安装是独立安装

在安装程序 Setup.exe 的控制文件 Setup.ini 中，设置了用于交互安装的控制参数。所以，对于安静模式安装必须设置：

[安装调试]

显示用户对话框 = 否：

将不显示用户对话框，如：欢迎或者许可证对话框。

显示重启对话框 = 否：

不显示最终对话框（重启 是/否）。

自行打开安装

也可以借助 NagsAway.exe 工具在该对话框上探测，来阻止最终显示的对话框（重启 是/否），使它出现时自动被确认。

必须在调用 Setup.exe 前启动 NagsAway.exe。需要配置文件 NagsAway.ini，它和 NagsAway.exe 位于同一目录并具有下列内容：

[通用]

时间间隔 = 100

[1]

Window 标题 = 安装完成

控制标题 = 是，立即重启计算机。

WinSubCheck=0

CtlSubCheck=0

[2]

Window 标题 = 安装完成

控制标题 = 完成

WinSubCheck=0

CtlSubCheck=0

安装是框架安装程序的子安装

在带有 Setup.exe 的安装目录中也有控制文件 OpFile.txt。

标准方式下已经在相关安装参数安装目录的控制文件 OpFile.txt 中设置了安静模式安装。

[SETUPREQUESTS]

安装模式 = 批处理

假设 Setup.exe 位于带有路径 <安装路径> 的目录中，则启动安装程序必须通过：

<安装路径>\Setup.exe -OF:<安装路径>\OpFile.txt

注意

在 -OF: 和 \OpFile.txt 之间不能有空格。

自行打开安装

在 Setup.exe 打开的安装文件下，也有控制文件 OpFile.txt。

标准方式下已经在相关安装参数的控制文件 OpFile.txt 中设置了安静模式安装，即：

[SETUPREQUESTS]

安装模式 = 批处理

假设 Setup.exe 位于带有路径 <安装路径> 的目录中，则启动自打开安装程序必须通过：

<安装路径>\Setup.exe -s -a -OF:.\OpFile.txt

3.1.3 对于 FAT16 文件结构的安装

前提条件

需要一台 PC 或者带有 CD-ROM 驱动器的 PG。

使用外部 PG/PC 上的 INTERSVR 来传输到 PCU。如果在 PG/PC 上有带有 FAT16 文件结构 (非 FAT32!) 的硬盘分区, 这就仅在 DOS / Win3.11 / Win95 / Win98 环境下可以。

传输

该功能将待安装的 HMI 软件传输到 D:\INSTALL。如果软件有一个执行文件如: "setup.exe", 在下次引导启动时进行自行安装。

传输时, 必须用并行接口安装 HMI 软件。

注意事项

必须注意, 在安装开始前在硬盘 D 区上有足够的空间可用。

操作

1. 关闭 PCU。
2. 用并行传输电缆 (PC 连接电缆、盖板连接电缆) 连接 PG/PC 的并行端口 LPT 和 PCU 的 LPT :
 - 传输方式: 并行
 - PC/PG - LPT1
 - PCU-LPT (25 芯)
3. 把目录 HMI 高级型从 CD 复制到 PG/PC 硬盘的驱动器 C:\ 上, 并更名为 "MMC2"。假如驱动器 C:\ 上没有足够的存储空间, 或者没有 FAT16 文件结构, 也可以使用另一个本地硬盘驱动器 (例如: D:\)。不能从 CD 上直接安装。
4. HMI 软件的安装通过 Windows 开始菜单执行 INTERSVR C: 的调用 (或者目录 MMC2 位于的驱动器) 来启动。

注意事项

如果您的 PC/PG 处于该状态, 则 PCU 必须处于关闭状态。(如果要接通 PCU, 则现在关闭它。)

5. 接通 PCU。

PCU 引导启动期间（在关闭和打开控制系统之后）：

6. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。

显示基本菜单：

7. 按下键 <1>：安装/升级 SINUMERIK 系统

系统要求输入口令。

8. 输入 0 到 2 级的一个口令。

显示下列菜单：

请选择媒介：

- 1 从软盘安装
- 2 通过串口/并口安装
- 3 从网络驱动器安装

9 返回主菜单

您选择 [1, 2, 3, 5, 9]？

9. 按下键 <2>：通过串口/并口安装。

数据窗口运行中。

重新启动 PCU 后，安装将自动启动。

10. 按照加载的包的安装方案，执行每个包专用的询问，直至安装结束。

安装结束后，重新执行重启。

11. 用 <ALT> + F4 来结束 PC/PG 服务模式。

3.1.4 对于 FAT32 文件结构的安装

原理

该功能将待安装的 HMI 软件传输到 D:\INSTALL。如果软件有一个执行文件如：“setup.exe”，由 Windows 在下次引导启动时进行自行安装。

注意事项

必须注意，在安装开始前在硬盘 D 区上有足够的空间可用。

前提条件

需要一台 PC 或者带有 CD-ROM 驱动器的 PG。此外要求 PCU 上有一个全键盘。对于这种安装，需要与系统软件一起发送的 CD-ROM。
如果在所用的 PC/PG 上只能用带有 FAT32 文件结构的分区，必须使用该过程。

传输

传输时，必须用并行接口安装 HMI 软件。

操作

1. 从 CD 的 HMI 目录向 PC/PG 的 WINDOWS\COMMAND 复制 INTERLNK.EXE。在 CONFIG.SYS 中输入下行：
`device=c:\windows\command\interlnk.exe /AUTO`
2. 关闭 PCU。
3. 用并行传输电缆 (PC 连接电缆、盖板连接电缆) 连接 PC 的并行端口 LPT 和 PCU 的 LPT。
4. 接通 PCU。
5. 在引导启动管理器中选择“服务菜单”。[不可见的菜单项]。
显示基本菜单：
6. 按下键 <8>：启动 PC 链接
系统要求输入口令。
7. 输入 0 到 2 级的一个口令。
8. 现在把 CD 上的目录 HMI 高级的整个内容复制到 PCU 的目录 D:\INSTALL 中。
9. 传输结束后，用 <ALT>+F4 来结束 PCU 上的 INTERSVR。之后执行重启。
现在自动从 PCU 开始用 InstallShield 进行安装。可以用 TAB 和光标键来操作 InstallShield 的询问框 (TAB 符合 OP0xx 上的 END 键)。
10. PCU 重启后，进行系统引导启动。

3.1.5 通过网络连接来安装

概述

该功能将待安装的 HMI 软件传输到 PCU 的 D:\INSTALL。
。如果软件有一个执行文件如："setup.exe"，由 Windows 在下次引导启动时进行自行安装。

注意

必须注意，在安装开始前在硬盘 D 区上有足够的空间可用。

对于另一种语言版本的 Windows 系统，下列设置过程含义不同。然而，总是保持给出的步骤顺序。

操作

PCU 引导启动期间（在关闭和打开控制系统之后）

1. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。

显示基本菜单：

2. 按下键 <1>：安装/升级 SINUMERIK 系统

系统要求输入口令。

3. 输入 0 到 2 级的一个口令。

显示下列菜单：

请选择媒介：

- 1 从软盘安装
- 2 通过串口/并口安装
- 3 从网络驱动器安装

9 返回主菜单

您选择 [1, 2, 3, 9]？

4. 按下键 <3>。

可能会出现用于对网络进行必要设置的输入屏幕：

请选择

- 1 管理网络驱动器
- 4 更改安装目录

- 9 返回上一级菜单

您选择 [1, 4, 9]?

5. 按下键 1，首先对网络连接进行设置。出现一个带有网络详细功能的屏幕：

请选择

- 1 连接到网络驱动器
- 2 显示连接的网络驱动器
- 3 从所有网络驱动器断开
- 4 更改网络设置

- 9 返回上一级菜单

您选择 [1, 2, 3, 4, 9]?

6. 按下键 <1>：

输入口令：

输入口令，用该口令可以从联网的 PG/PC 撤销设置的用户，从释放的目录撤销软件：
PG/PC 的本地用户口令

网络驱动器字母：

7. 给出连接相应的 PCU 驱动器字母 (例如 : "G")

待安装的目录 (例如 \\r4711\dir66)：

8. 输入 PG/PC 的计算机名称和存取目录名称 (使能名称)

输出到 PG/PC 的连接数据 :

机床名称: PCU_1_名称
用户名: auduser
传输协议: NETBEUI
登录到域: 否
连接网络驱动器 (最近) : G : (\\r3344\HMINEW)

请选择

- 1 连接到网络驱动器
- 2 显示连接的网络驱动器
- 3 从所有网络驱动器断开
- 4 更改网络设置

9 返回上一级菜单

您选择 [1, 2, 3, 4, 9] ?

9. 按两次键 <9> 。

显示下列菜单 :

请选择

1 管理网络驱动器

- 4 更改安装目录
- 5 从 G 安装 :

9 返回上一级菜单

您选择 [1, 4, 5, 9] ?

结果 : 新近激活的网络连接在 <5> 下可用于安装。

更改所连驱动器的安装目录

如果待安装的软件位于使能目录下的一个子目录中，可对安装目录进行下列更改。

1. 按下键 <4>：更改安装目录

出现一个输入屏幕，其中给出了安装目录（例如：HMI_INS）：

旧安装目录： G:\

新安装目录： G:\HMI_INST

现在可以启动传输：

请选择

- 1 管理网络驱动器

- 4 更改安装目录
- 5 从 G:\HMI_INST 安装：

- 9 返回上一级菜单

您选择 [1, 4, 5, 9]？

2. 按下键 <5> 来启动安装：

- 系统启动从 PG/PC 到 PCU 的传输。
- 传输后自动重启 PCU（重启）。
- 在引导启动管理器中选择“SINUMERIK”。
传输的软件在这种情况下通过执行菜单来安装。

3. 系统检查在目录 D:\INSTALL 中是否已经有文件。

启动安装程序。为此通过网络连接结束安装。

注意事项

如果软件不能复制到 PCU 的目录 D:\INSTALL，系统不能启动安装。

4. 退出菜单时询问，是否应该保存设置的网络连接参数。

注意

自动执行用户安装时，可以设置多个安装目录。安装目录是子目录安装，在参数文件 E:\Windows\System32\HMI\Serve.ini 的 SetupDirs 分区中列出。一个分区内的密码名在参数文件 E:\Windows\System32\HMI\Serve.ini 中必须是明确的。

预设了安装目录 "D:\Install"。

3.2 网络驱动器的设置

概述

服务菜单的下面几个功能把网络驱动器用作软件安装或系统备份或修复时的来源或目标。

用于网络运行的设置

为了进行所需要的设置，在必要的操作菜单位置提供菜单管理网络驱动器。

然后得到下列网络管理功能：

-
- 1 连接到网络驱动器
 - 2 显示连接的网络驱动器
 - 3 从所有网络驱动器断开
 - 4 更改网络设置

 - 9 返回上一级菜单
-

对于单个的菜单项得到下列子菜单：

1 连接到网络驱动器

1. 输入口令，设置好的用户可以通过该口令从使能目录通过网络连接的 PG/PC 调用需要的数据（例如：软件）或者写入使能目录（例如：PCU 图形）。

带有 Windows XP 的 PG/PC：

PG/PC 的本地用户口令

输入口令：

2. 给出连接相应的 PCU 驱动器字母（例如："G"）
-

网络驱动器字母：

-
3. 输入 PG/PC 的计算机名称和存取目录名称 (使能名称) 出现屏幕 , 可以看到与 PG/PC 的连接数据 :

待安装的目录 (例如 \\r3344\HMINEW) :

机床名称: PCU_1_名称

用户名: auduser

传输协议: NETBEUI

登录到域: 否

连接网络驱动器 (最近) : G : (\\r3344\HMINEW)

请选择

- 1 连接到网络驱动器
- 2 显示连接的网络驱动器
- 3 从所有网络驱动器断开
- 4 更改网络设置

9 返回上一级菜单

您选择 [1, 2, 3, 4, 9] ?

2 显示连接的网络驱动器

显示连接驱动器的当前状态 :

状态: 确定关闭对话框

本地名称: G:

远程名称: \\r3344\HMINEW

状态: 确定关闭对话框

本地名称: H:

远程名称: \\r3344\HMI_P1

按任意键继续...}

3 从所有网络驱动器断开

按任意键继续...}
断开所有网络连接。

4 更改网络设置

显示下列子菜单：

-
- 1 更改机床名称 (仅用于 DOS 网络)
 - 2 更改用户名
 - 3 选择协议 (NETBEUI 或者 TCP/IP)
 - 4 选择登录到域 (是或否)
 - [5 更改域名]
 - [6 更改 TCP/IP 设置]

 - 9 返回上一级菜单

您选择 [1, 2, 3, 4, [5,] [6,] 9] ?

更改网络设置的菜单项：

1 更改机床名称 (仅用于 DOS 网络)

将输出带有提示的概述：

当前网络设置：

机床名称：
用户名： USER
传输协议： NETBEUI
登录到域： 否

请选择：

- 1 更改机床名称 (仅用于 DOS 网络)
- 2 更改用户名
- 3 选择协议 (NETBEUI 或者 TCP/IP)
- 4 选择登录到域 (是或否)

9 返回上一级菜单

您选择 [1, 2, 3, 4,9]?

提示：本更改暂时只用于 DOS 网络！

在下一个 Windows 上启动，该机床名称被重置到

Windows 机床名称。如果要永久的设置机床名称，

请直接使用上层菜单中“更改机床名称（用于 Windows 和 DOS 网络）”

在上层菜单等级！

旧机床名称：...

新机床名称：...

注意

如果没有输入新的名称，保持当前名称。

2 更改用户名

旧用户名：...

新用户名：...

注意

如果没有输入新的名称，保持当前名称。

3 选择协议 (NETBEUI 或者 TCP/IP)

每次按下键 <3> , 都会从当前协议切换到另一个协议。在连接的标题栏中显示设置的协议。

注意

如果已经有了一个带 NETBEUI 的连接, 只能通过重启切换到 TCP/IP。如果尚无连接激活, 可立即设置另一个协议。

如果已经有了一个带 TCP/IP 的连接, 只能通过重启切换到 NETBEUI。

当切换到了 TCP/IP 时, 扩展屏幕更改网络设置指向附加项 6 更改 TCP/IP 设置, 如果 TCP/IP 设置成了当前协议。

切换协议的屏幕 TCPIP→NETBEUI , 已经存在连接 :

当前网络设置 :

机床名称: PCUXXXXXXXX

用户名: auduser

传输协议: TCPIP, 自动通过 DHCP 得到 IP 地址

登录到域: 否

请选择 :

- 1 更改机床名称 (仅用于 DOS 网络)
- 2 更改用户名
- 3 选择协议 (NETBEUI 或者 TCP/IP)
- 4 选择登录到域 (是或否)
- 6 更改 TCPIP 设置

9 返回上一级菜单

您选择 [1, 2, 3, 4, 6, 9]?

警告 :

您已经有与 TCPIP 的连接, 现在要切换协议。

可以执行 (并保存更改) , 但是您不能重新引导启动就再次连接 !

真的要切换协议 [Y,N] ?

4 选择登录到域 (是或否)

如果用 Y
应答选择登录到域，而且域口令保存在口令列表中，将立即连接到服务器，如果定义了激活所属的登录脚本文件。

如果按
N，则出现一个域数据和口令数据的对话框，当域口令不包含在口令列表中时，情况相同。
仅当设置了 Y 时，激活域登录。

5 更改域名。

可以按照 1 和 2 下的名称更改来进行相应的操作。

如果用 N 来对切换登录到域进行应答，则在菜单更改网络设置中略去附加项 **5 更改域名**。

6 更改 TCP/IP 设置

提供下列用于 TCP/IP 的子项：

-
- 1 选择 "获取 IP 地址" (自动或手动)
 - 2 更改 IP 地址
 - 3 更改子网掩码
 - 4 更改网关
 - 5 域名服务器
 - 6 更改 DNS 扩展名

 - 9 返回上一级菜单

您选择 [1, 2, 3, 4, 5, 6, 9]?

子项条件：

1 选择 "获取 IP 地址"

可以在自动和手动之间切换。如果切换到自动方式，通过 DHCP 获取 IP 地址。接下来必须操作合适的域名服务器(DNS)。

手动方式时，在 2 更改 IP 地址下必须直接给出 IP 地址。

注意

在该屏幕界面时，IP 地址/子网掩码的所有输入用空格而不是通用的点作为分隔符。

举例：192 168 3 2 而不是 192.168.3.2

2 更改 IP 地址

如果获取 IP 地址设置到手动方式，进行 PG/PC 的 IP 地址输入。不允许选择 DHCP。

3 更改子网掩码

如果获取 IP 地址设置到手动方式，进行 PG/PC 子网掩码的输入。不允许选择 DHCP。书写方式象 IP 地址一样用空格作为分隔符！

旧子网掩码： ...

新子网掩码： <子网掩码>

4 更改网关

如果获取 IP 地址设置到手动方式，进行网关的输入。输入的网关作为 IP 地址。

旧网关： ...

新网关： < IP 地址 >

5 域名服务器

如果获取 IP 地址通过 DHCP 设置到自动方式，进行所属域名服务器 (DNS) 的 IP 地址输入。DHCP 通常能自动识别到一个合适的域名服务器。如果 DNS 有合适的计算机和 IP 地址分配，则不必输入。

6 更改 DNS 扩展名

这里给出了域名服务器的扩展名，如：SIEMENS.COM

退出菜单

退出菜单时询问，是否应该保存设置的网络连接参数。

3.3 用服务桌面运行网络

概述

服务桌面也用作服务模式，可在该模式下可以利用 Windows 功能用于控制系统的开机调试。另外，SINUMERIK 桌面可用作网络运行的平台（例如：用于安装 HMI 软件），用于设置 HMI 系统软件运行环境或者系统伴有的措施如：SIMATIC STEP7 的授权、硬盘检查、版本检查等。

配置网络运行

PCU 具有标准的网络连接：

1. 插上用于 TCP/IP 通讯的 Ethernet 电缆。
预设置的是 DHCP。（除了协议 TCP/IP 外，也可以预安装协议 NETBEUI。）
2. 通过开始→设置→网络连接→本区域连接→属性，来匹配设置。
3. 最晚到此时，PCU 的计算机在网络范围内必须是明确的：
在开始→设置→控制面板→系统注册计算机名称，更改按钮中，可以输入所需的名称。

通过 TCP/IP 连接

对于 TCP/IP，可以配置一个 WINS 服务器。取消 DNS 服务器和 DNS 扩展名。对 DHCP，也可以配置 WINS 输入项和网关输入项并显示。

IP 地址现在也可以输入"."作为分隔符。

注意

WINS 不允许用"0 0 0 0"占用，否则该 DHCP 连接对于子网络无效。

补充安装 Windows 组件

前提条件：需要带有维修包 2 的 Windows XP 恢复 CD，用于补充安装单个组件。此外，在 D:\Updates\Base 下保留了软件，它可能在安装外部应用程序时，用于补充安装。

1. \MDAC21.SP2:数据访问组件 V2.1 SP2 (2000 年生效)
2. \SP6a:Windows NT 4.0 维修包 6a
3. \DRIVERS\SYSTEM.INI：键盘表（需要时可复制到 E:\Windows）。
4. \DDEFIX

对"安装新固定"时要输入的安装路径是：D:\Updates\Base\DDEFIX\hotfix.inf

5. \VBFIX：VB 补丁程序

3.3.1 SIMATIC STEP 7 安装与授权

概述

SIMATIC STEP 7 V5.3 SP2 同样可以安装在 PCU 上。

交付物品：	SIMATIC STEP 7 V5.3 SP2
组成部分：	SIMATIC STEP 7 V5.3 SP2 和 添加用于 SINUMERIK 810D/840D/840D sl/840Di sl
交付形式：	2 张安装 CD
系统前提条件：	高于 V8.0 的 PCU 基础软件 要求网络连接或者连接外部 CD-ROM 驱动器。 推荐： 鼠标连接

注意

PCU 上现有的“MPI 驱动器”包属于 HMI 软件，不允许卸载！

在 Windows XP 下安装

必须遵守这里所述的安装顺序！

1. 在维修桌面上启动 PCU。
2. 建立从 CD 1 的存取（通过网络或者外部 CD-ROM 驱动器）并在那里从启动目录 SETUP.EXE 中调用。
3. 由操作者执行安装。STEP 7 安装目录应当在 F:\... 更改，因此 F: 上的目录可任意选择。
4. 询问“传输许可证密码”时，必须回答“否，以后传输许可证密码”。安装 SINUMERIK-AddOn 后随即进行许可证验证。安装结束后需要重新启动 PCU。
执行启动时重新选择维修桌面。
5. 切换到 CD2 的目录 SINUMERIK_Add_on 并在那里调用 SETUP.EXE。
由操作者执行安装。安装结束后需要重新启动 PCU。执行启动时重新选择维修桌面。
6. 在维修桌面上启动“STEP7 授权”的快捷图标。由此授权 STEP7，现在可以从 HMI 高级版的操作界面来启动（STEP 7 显示为操作菜单扩展栏上的专用操作区域，保护等级 3）。

在 F:\Add_on\oemframe.ini 中自动输入：

```
[s7tgtopx]
; 使用 HMI 高级版：取消最小化/最大化按钮
; 属于 Step7 窗口
WindowStyle_Off=196608
; 使用 HMI 高级版：当 Step7 结束时切换到前一任务
nSwitchToTaskAfterTermination= -2
```

该输入项在 OEM 配置中必须修改。

3.3.2 通过服务菜单授权 SIMATIC STEP 7

前提条件

用于 SINUMERIK 硬件的 SIMATIC STEP7 V5.3 SP2 已经安装好了。

操作

PCU 引导启动期间（在关闭和打开控制系统之后）：

1. 在引导启动管理器中选择服务菜单 [不可见的菜单项]
显示基本菜单。
2. 按下键 <2>：SINUMERIK 工具和选项
系统要求输入口令。
3. 输入 0 到 2 级的口令。
显示下列菜单：

```
          请选择
X         1 为 PCU 激活 Step7

          9 返回主菜单
```

您选择 [1, 9]？

4. 为了对 STEP 7 进行授权，按下键 <1>。

接着进行自动重启。

Windows 进行下列引导启动时，该功能的执行要在启动 HMI 系统软件之前。

备份和恢复数据

应用

借助软件 Norton Ghost

可以把整个硬盘的内容作为磁盘图像（文件图形）保存。保存这些不同数据载体上的磁盘图像用于以后硬盘上的数据恢复。

出厂时，Norton Ghost 在 PCU 的备用硬盘和整个 PCU 硬盘上发送。Norton Ghost 可在下面描述的几个备份和修复过程中使用。

进一步提示见网址 www.ghost.com

4.1 通过 Ethernet 连接进行数据备份/修复

4.1.1 向 PG/PC 硬盘备份 PCU 硬盘

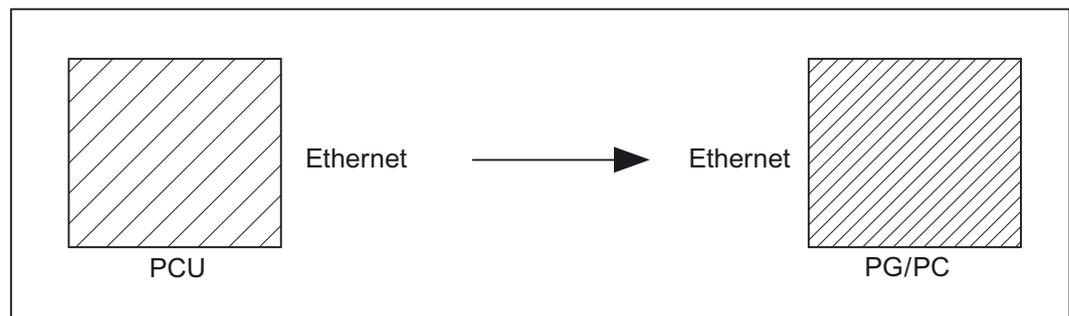


图 4-1 备份硬盘

前提条件

- 在 PCU 和 PG/PC 上安装的程序 Ghost 版本 $\geq 7.x$ 。
- PCU 和 PG/PC 上的 Ghost 版本必须相同。
- PG/PC 上有保存备份图像的目录。
- PG/PC 上有足够的空余存储空间。
- 在 PG/PC 上可以安装任意的 Windows 版本。
- 用 Ethernet 电缆连接 PCU 和 PG/PC。

操作

PCU 引导启动期间 (在关闭和打开控制系统之后)

1. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。

显示基本菜单：

2. 按下键 <7>：备份/恢复

系统要求输入口令。

3. 输入 0 到 2 级的一个口令。

显示下列菜单：

请选择

- 1 用 GHOST 备份/修复硬盘
- 4 用 GHOST (本地) 备份/修复区
- 5 ADDM 备份/修复

9 返回主菜单

您选择 [1, 4, 5, 9]?

4. 按下键 <1>：

显示下列菜单：

请选择

- 1 配置 GHOST 参数
- 2 硬盘备份到 <路径>, 模式 ...
- 3 硬盘修复从 <路径>, 模式 ...
- 4 切换到其它 GHOST 版本

9 返回上一级菜单

您选择 [1, 2, 3, 4, 9]?

注意

路径数据是缺省设置与驱动器有关。您可以更改路径数据。

开始"可能"

只有缺省设置不合适时，才要求下列操作。

这里可以设置接口方式、网络连接、备份图像路径以及备份图像分割。

1. 设置程序 Norton Ghost 的参数：

按下键 <1>：配置 GHOST 参数

显示下列菜单：

请选择

- 1 设置连接方式并联 (LPT:)
- 2 设置连接方式本地/网络
- 3 更改备份图像文件名
- 4 更改修复图像文件名
- 5 更改机床名称 (用于 windows 和 DOS 网)
- 6 管理网络驱动器
- 7 更改分割方式

- 9 返回上一级菜单

您选择 [1, 2, 3, 4, 5, 6, 7, 9]?

注意事项

对于这里所述的网络连接，接口方式必须始终设置到本地/网络。

1. 更改接口：

按下键 <2>：选择并确认设置连接方式本地/网络。

2. 建立/更改网络连接：

按下键 <6>：管理网络驱动器

3. 更改备份图像路径：

按下键 <3>：更改备份图像文件名：在 PG/PC
上设置用于备份文件的完整的路径名，例如：C:\SINUBACK\PCU\HMI.gho

4. 分割备份图像：

按下键 <7>：更改拆分方式

选择是否把备份图像分割到区，如果是，给出分割大小（最大分割大小：2048 MB；
分割大小预设置：640 MB）。分隔文件的名称分配如下：

<名称>.gho

<名称>.001

<名称>.002

<名称>....

5. 返回上一级菜单：

按下键 <9>：返回上一级菜单

询问：保存 GHOST 参数？回答是。

结束"可能"

下面是一般要求：

备份硬盘

现在可以启动传输：

1. 按下键 <2>：硬盘备份到<路径名>，LOCAL/NETWORK 模式

如果没有本地分区的数据备份，则显示下列菜单：

请选择

- 1 备份无本地图像
- 2 备份有本地图像

您选择 [1, 2]？

如果不想对本地分区数据备份（显著减少传输时间和存储空间），按：

1. 按下键 <1>：备份无本地图像或
2. 按下键 <2>：备份有本地图像。

– 随即出现一个信息窗口：

则要求检查，PCU 和 PG/PC

之间是否建立了连接。显示用于目录的目标路径，该目录中保存了 PG/PC 上的备份图像。要求检查是否在 PG/PC 上有该目录存在。

– 启动备份，通过在消息窗口中按 Y 来应答。出现软件 Norton Ghost 的消息窗口：显示传输步骤、所用的路径和待传输的数据量。

– 备份中断后询问：

是否重新尝试备份 [Y,N]？

用 N 应答，出现主菜单。用 Y 应答，重新开始备份。

3. 按下键 <9>：返回上一级菜单

4. 断开连接的驱动器：可以断开与 PG/PC 的所有连接。

4.1.2 从 PG/PC 硬盘修复 PCU 硬盘



图 4-2 修复硬盘内容

前提条件

- 在 PCU 和 PG/PC 上安装的程序 Ghost 版本 $\geq 7.x$ 。
- 在 PG/PC 上有保存修复图像的目录。
- 在 PG/PC 上安装任意 Windows。
- 用 Ethernet 电缆连接 PCU 和 PG/PC。

操作

1. 接通 PG/PC，在驱动器中放入 CD。
PCU 引导启动期间（在关闭和打开控制系统之后）
2. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。
显示基本菜单：
3. 按下键 <7>：备份/恢复
系统要求输入口令。
4. 输入 0 到 2 级的一个口令。
显示下列菜单：

请选择：

- 1 用 GHOST 备份/修复硬盘
- 4 用 GHOST（本地）备份/修复区
- 5 ADDM 备份/修复

9 返回主菜单

您选择 [1, 4, 5, 9]?

1. 按下键 1。

显示下列菜单：

请选择：

- 1 配置 GHOST 参数
- 2 硬盘备份到 <路径>, 模式 ...
- 3 硬盘修复从 <路径>, 模式 ...
- 4 切换到其它 GHOST 版本

9 返回上一级菜单

您选择 [1, 2, 3, 4, 9]?

注意

路径数据是缺省设置与驱动器有关。您可以更改路径数据。

开始"可能"

只有缺省设置不合适时，才要求下列操作。

这里可以设置接口方式、网络连接、备份图像路径以及备份图像分割。

1. 设置程序 Norton Ghost 的参数：

按下键 <1>：配置 GHOST 参数

显示下列菜单：

请选择：

- 1 设置连接方式并联 (LPT:)
- 2 设置连接方式本地/网络
- 3 更改备份图像文件名
- 4 更改修复图像文件名
- 5 更改机床名称 (用于 Windows 和 DOS 网)
- 6 管理网络驱动器
- 7 更改分割方式

9 返回上一级菜单

您选择 [1, 2, 3, 4, 5, 6, 7, 9]?

注意事项

对于这里所述的网络连接，接口方式必须始终设置到本地/网络。

1. 更改接口：

按下键 <2>：选择并确认设置连接方式本地/网络。

2. 建立/更改网络连接：

按下键 <6>：管理网络驱动器

3. 设置图像修复：

按下键 <4>：更改修复图像文件名

在 PG/PC 上设置用于修复文件 HMI.gho 的完整的路径名，例如：
C:\SINUBACK\PCU\HMI.gho

4. 返回上一级菜单：

按下键 <9>：返回上一级菜单

询问：保存 GHOST 参数？回答是。

结束"可能"

下面是一般要求：

录入硬盘内容

1. 按下键 <3> : 硬盘修复从<路径名> , 本地方式
通过选择该菜单显示下列菜单 :

磁盘分区时, 知道
图像配置的系统基础非常重要。
请选择创建图像的系统基础 !

- 请选择
- 1 Windows XP
 - 2 Windows NT
 - 3 Win95
 - 4 WfW3.11
 - 5 DOS (空余部分)

 - 9 返回上一级菜单

您选择 [1, 2, 3, 4, 5, 9] ?

2. 这里选出要录入图像位于的操作系统。硬盘分区的进行方式与此有关。
选择相应的操作系统后出现其它菜单 :

要进行何种磁盘分区 ?

- 1 标准分区 (缺省)
- 2 用户定义的分區
- 3 图像分区

您选择 [1, 2, 3] ?

3. 说明是否以标准方式 (与操作系统有关) 即 : 用固定的分区数和分区大小来划分硬盘或者通过操作者自定义分区。在第一种情况时, 软件 Norton Ghost 自动得到分区参数, 而在第二中情况时, 操作者要在 Norton Ghost 中自行输入分区参数。第三种情况时, 待加载的图像决定分区。
4. 按下键 <9> : 返回上一级菜单
5. 仅当首次开机调试或者修复时 (在 Ghost 运行或修复时自动) : 在菜单 1 配置 GHOST 参数中选择

设置 PCU 网络名称

如果您的机床是网络的一部分，则该机床需要有一个在整个网络中唯一的名称。

修复的图像带来一个机床名称。

可立即更改该机床名称。

注意：

不能更改机床名称，当：

1. 修复图像是机床的备份图像且
2. 由于正在生成备份图像，机床名称尚未改变。

要求微软域安全性的网络注意事项：

无论何时您更改机床名称时（即使您输入的名称与恢复图像中包含的机床名称一致），您的机床名称和它的用户将不再属于微软网络域（参见微软文献）。

按任意键继续... .

确实要输入新的机床名称吗？

如果您要手动输入新的机床名称，

1. 请勿使用 "`\/\[\]:;|<>+=,?*`" 和空格而且
2. 只要旧的机床名称仍在恢复图像中使用，新的机床名就必须准确。

如果您不想输入新的机床名，恢复图像中的机床名仍然有效。

提示： 如果要输入一个不同长度的机床名，就必须在 Windows 中进行操作！

请选择：

- 1 手动输入机床名称
- 2 随机输入机床名称

9 无新机床名称

您选择 [1, 2, 9]?

您可以自己为 PCU 输入一个计算机名称，或者让系统自动生成一个名称：

- 自己输入名称：

<1 手动输入机床名称>

出现一个可以输入 10 位名称的输入屏幕。按输入键确认后，系统中的名称被更改。

标准名称：PCUxxxxxxx [10 位]。

- 自动输入名称：

<2 随机输入机床名称>

自动生成一个名称并在系统中更改。如果要求，可在 Windows 中通过控制系统检查：

“开始” → “设置” → “系统控制” → “系统”，标签“计算机名称”

- 在 PCU 上出现软件 Norton Ghost 的消息窗口：

- 显示传输的步骤
- 显示所用的路径
- 显示剩余的传输数据量。

如果要从 GHOST 图像进行批量调试，必须设置当前的 PCU 名称来替代 GHOST 图像中包含的名称。

通过工具 Ghostwalker 对新计算机名称（和带有 2 个随机生成名称的输入机床名称）以及 PCU 上的新系统 ID (SID) 进行赋值。

Ghostwalker 运行结束后，操作者可以重复录入一次。

这对于录入执行不正确的情况（如：连接问题、被用户中断，...）是必要的。5 秒之内用户存取无效，PCU 自动重启。需要时，在 Windows 下按要求更改计算机名称。

看起来，Ghost 修复成功。

但是如果没有连接或者取消了修复，

Ghost 报告无故障！就必须更改为：

请选择：

- 1 重启系统（5 秒后缺省）
- 2 再次尝试硬盘修复

9 返回上一级菜单

您选择 [1, 2, 9]?

如果在以后的时间要再次更改计算机名称，可以通过子菜单备份/修复->用 GHOST 备份/修复硬盘->配置 GHOST 参数中的第 5 项更改机床名称（用于 Windows 和 DOS 网络）来进行。同样通过 Ghostwalker 来更改。

引导启动：成功修复后必须引导启动。

注意事项

中断修复过程时，硬盘处于不一致的状态。在这种情况下，需要用带有 Norton-Ghost 软件和网络软件的引导盘。引导盘通过 eSupport 可用。

4.1.3 使用备用硬盘

关于 HMI 硬盘的机械和电子部件更换所要求的措施参见：

文献： 操作组件手册

下面对整个硬盘数据备份的再次录入进行了说明，以便维修时用户数据和系统数据一致可用。

Norton Ghost

借助软件 Norton Ghost 可以把 PCU 的全部硬盘内容保存为图像文件。这些在不同数据载体上的磁盘图像文件，可用于以后的硬盘修复。

出厂时，程序 Norton Ghost 随每个 PCU 和备用硬盘发送。

推荐：

在 CD 上存档硬盘备份（硬盘图像）包括程序 Norton Ghost。

录入数据备份

前提条件：

- 程序 Ghost 安装在了 PG 上。
- 有新的备份硬盘
- PCU 用并联电缆与 PC/PG 连接
- PG 上有 CD 驱动器。

注意

备份硬盘中没有 Windows 操作系统和 HMI 系统软件。

操作：

1. 在 PCU 中装入新的备用硬盘 (参见硬盘操作说明) :
 - 把硬盘装入插接处。
 - 插上硬盘 PCU 连接电缆。
 - 用 4 个滚花高头螺钉固定住硬盘。
 - 松开运输保险栓：从“操作中”旋转直至衔接处。
2. 接通 PG，在驱动器中放入 CD。
PCU 引导启动期间 (在关闭和打开控制系统之后)
3. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。
显示基本菜单：
4. 按下键 <7>：备份/恢复
系统要求输入口令。
5. 输入 0 到 2 级的一个口令。
显示下列菜单：

请选择

- 1 用 GHOST 备份/修复硬盘
- 4 用 GHOST (本地) 备份/修复区
- 5 ADDM 备份/修复

- 9 返回主菜单

您选择 [1, 4, 5, 9]?

6. 按下键 <1>
设置程序 Norton Ghost 的参数：
7. 按下键 <1>：配置 GHOST 参数
8. 按下键 <3>：硬盘修复从<路径名>，并联方式
选择该菜单时出现消息窗口：
则要求检查，HMI 和 PG/PC 之间是否建立了连接。选择一个 HMI 图像文件，作为要录入的修复图像。

- PG/PC:
在 DOS 窗口中或者在 DOS 平面中输入指令"ghost -lps" 用于启动程序 Norton Ghost。
 - PCU:通过应答 (是) 信息窗口来启动修复。
9. 在 PCU 上出现软件 Norton Ghost 的消息窗口 :
- 显示传输的步骤
 - 显示所用的路径
 - 显示剩余的传输数据量

注意

如果在修复过程中传输中断，在硬盘上没有一致的系统。
因此需要启动引导磁盘，它包含 MS-DOS _ 6.X 引导启动和 Norton Ghost 软件。

- 10.按下键 <9> : 返回主菜单
- 11.成功修复后，将自动重启 PCU。
时间长度：大约 15-20 分钟，用于通过 LPT 对于一个 540 MB 的硬盘生成压缩的磁盘图像 = 130 MB。

4.2 本地备份/修复 PCU 分区

备份分区

除了在外部驱动器上备份全部硬盘内容外，也可以把 PCU 的 C:、E: 和 F:本地备份到 PCU 的D: 区上 (分区数据备份)。

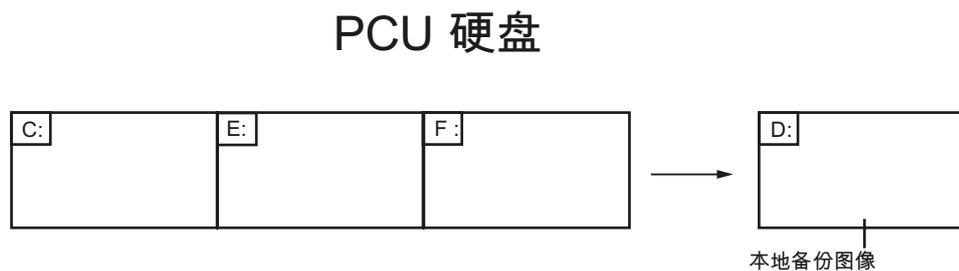


图 4-3 向驱动器 TEMP (D:)进行分区数据备份

修复分区

分区数据备份时生成的本地备份图像可用于，需要时重新修复 C:, E:和 F: (分区数据备份)。

PCU 硬盘

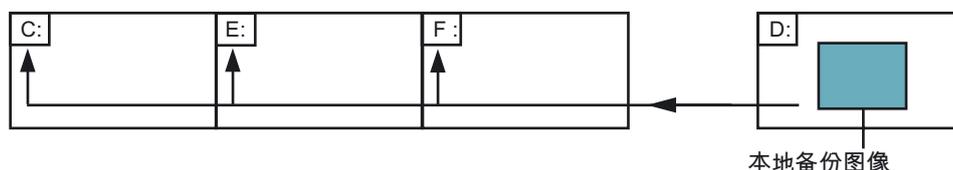


图 4-4 从驱动器 TEMP (D:) 进行分区数据备份

修复 E 区 (Windows)

此外需要时，也可以用本地备份图像单独修复 E: 区(Windows)。例如：当系统异常终止或者 PCU Windows 的失控关闭后，无法再引导启动时，(分区 E 的数据修复)是必要的。

PCU 硬盘

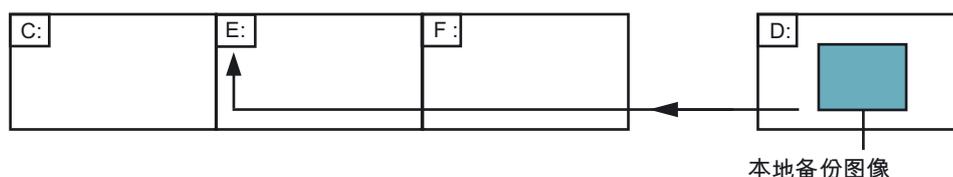


图 4-5 从驱动器 TEMP (D:)进行分区修复

对于分区数据修复或者分区 E 数据修复，理论上每个在 D: 区上的本地备份图像都可以修复。

当然，对分区 E 的数据修复时，要经考虑选择要用的本地备份图像：所有在 F: 区上安装的应用程序的注册信息将会在要修复的分区 E: 上。

因此有必要始终确定一个特定的本地备份图像 (紧急备份图像)，该图像可在紧急情况下 (Windows 不能再引导启动时) 通过分区 E 数据修复可以录入，而不损害安装的应用程序的执行。

在服务菜单中特别提供支持，来确定一个用于可能出现的紧急情况的本地备份图像，并在紧急情况时方便的触发 E 分区数据修复。

同时可以在 D: 区上设置本地备份图像的最大数 (≤ 7)。

4.2.1 备份分区

PCU 硬盘

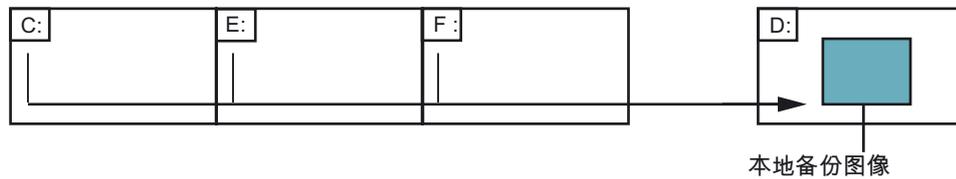


图 4-6 向驱动器 TEMP (D:)进行分区数据备份

操作

PCU 引导启动期间 (在关闭和打开控制系统之后)

1. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。

显示基本菜单：

2. 按下键 <7>：备份/恢复

系统要求输入口令。

3. 输入 0 到 2 级的一个口令。

显示下列菜单：

请选择

- 1 用 GHOST 备份/修复硬盘
- 4 用 GHOST (本地) 备份/修复区
- 5 ADDM 备份/修复

- 9 返回主菜单

您选择 [1, 4, 5, 9]?

4. 按下键 <4>。

显示下列菜单：

请选择

- 1 配置 GHOST 参数
- 2 分区备份，本地模式
- 3 分区修复，本地模式
- 4 E 区 (仅 Windows) 修复，本地模式
- 8 删除图像

9 返回上一级菜单

您选择 [1, 2, 3, 4, 5, 8, 9] ?

开始"可能"

只有缺省设置不合适时，才要求下列操作。

1. 为了设置本地备份图像的最大数目(最大 7)，按键 < 1 >:配置 GHOST 参数

显示下列菜单：

GHOST 连接方式：本地

最多备份图像：1

当前备份图像：0

请选择

- 1 配置 GHOST 参数
- 2 分区备份，本地模式

9 返回上一级菜单

您选择 [1, 2, 9] ?

2. 如果要更改本地备份图像最大可能数值的预设值，选择菜单 1：输入一个新的数字 (最大 7)。

询问：保存 GHOST 参数？回答是。

3. 按下键 <9>，返回上级菜单：返回上一级菜单

结束"可能"

下述过程普遍适用：

1. 按下键 <2>：分区备份，本地模式

询问要保存新本地备份图像的名称，例如：SICHER1。名称长度：最多 7 个字符。

询问描述，是否要保存对本地备份图像的附加描述，例如：通过 HUGO 数据备份 1。

当再次录入可用备份图像列表中时，会提供这些附加的描述作为可选文本（如果缺少这些描述，在列表中会显示备份图像名称）。

询问，是否这个新的本地备份图像将来是否也用作紧急备份图像（而且换掉目前使用的紧急备份图像）。随即启动向 D: 的分区数据备份。

在 D: 下在目录 IMAGES 中对于每个分区创建一个文件，例如：SICHER1.gh1, SICHER1.gh3, SICHER1.gh4。再次录入时，这自然就被识别为 1 备份图像的附属部分。

分区数据备份结束后，出现信息备份结束。

2. 按下键 <9>，返回上级菜单：返回上一级菜单

4.2.2 恢复分区

PCU 硬盘

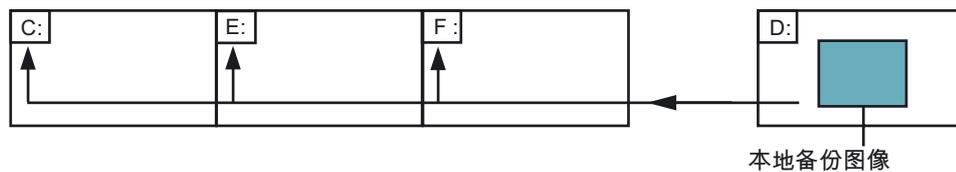


图 4-7 从驱动器 TEMP (D:) 进行分区数据备份

操作

PCU 引导启动期间（在关闭和打开控制系统之后）

1. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。

显示基本菜单：

2. 按下键 <7>：备份/恢复

系统要求输入口令。

3. 输入 0 到 2 级的一个口令。

显示下列菜单：

请选择

- 1 用 GHOST 备份/修复硬盘
- 4 用 GHOST (本地) 备份/修复区
- 5 ADDM 备份/修复

- 9 返回主菜单

您选择 [1, 4, 5, 9]?

4. 按下键 <4>。

显示下列菜单：

请选择

- 1 配置 GHOST 参数
- 2 分区备份，本地模式
- 3 分区修复，本地模式
- 4 E 区 (仅 Windows) 修复，本地模式
- 8 删除图像

- 9 返回上一级菜单

您选择 [1, 4, 5, 9]?

5. 按下键 <3>：分区修复，本地模式

询问要重新录入的本地备份图像，例如：通过 HUGO 数据备份 1。

请选择紧急图像：

- 1 PCU 基础软件 V06.00.02 Win NT 4.0
[02-01-2000: 05:32:32pm]
- EMERG 2 数据备份 1 通过 HUGO
[02-03-2000; 04:53:24pm]
- ...

- 9 返回上一级菜单

您选择 [1, 2, 3, ..., 9] ?

6. 选择所需的备份图像。

确定后出现小心警告：所有数据将被覆盖，而且将触发所选的备份图像向分区 C:、E:和 F:的录入。分区数据修复结束后，出现信息修复结束。

结果：

每次重新录入后，分区 E:和 F: 在下次系统引导启动时将通过检查程序 CHKDSK (磁盘检查) 来检查。在检查 E:和 F: 之间进行重新启动。

4.2.3 E 区：(Windows) 在 PCU 上修复

PCU 硬盘

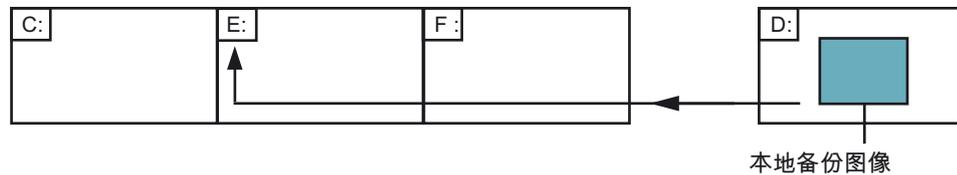


图 4-8 分区 E 数据修复

操作

PCU 引导启动期间 (在关闭和打开控制系统之后)

1. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。

显示基本菜单：

2. 按下键 <7>：备份/恢复

系统要求输入口令。

3. 输入 0 到 2 级的一个口令。

显示下列菜单：

请选择

- 1 用 GHOST 备份/修复硬盘
- 4 用 GHOST (本地) 备份/修复区
- 5 ADDM 备份/修复

9 返回主菜单

您选择 [1, 4, 5, 9]?

4. 按下键 <4> 。

显示下列菜单：

请选择

- 1 配置 GHOST 参数
- 2 分区备份，本地模式
- 3 分区修复，本地模式
- 4 E 区 (仅 Windows) 修复，本地模式
- 8 删除图像

- 9 返回上一级菜单

您选择 [1, 4, 5, 9]?

5. 按下键 <4> : E 区 (仅 Windows) 修复，本地模式

询问本地备份图像，其分区 E: (= Windows 分区)需要再次录入，例如：通过 HUGO 数据备份 1。

请选择紧急图像：

- 1 PCU 基础软件 V06.00.02 Win NT 4.0
[02-01-2000: 05:32:32pm]
- EMERG 2 数据备份 1 通过 HUGO
[02-03-2000; 04:53:24pm]
- ...

- 9 返回上一级菜单

您选择 [1, 2, 3, ...,9] ?

6. 选择所需的备份图像：

确定后出现小心警告：Windows 分区将被覆盖而且触发分区 E:的录入。

分区 E 数据修复结束后，出现信息修复结束。

结果：

下次 Windows 进行引导启动时，通过检查程序 CHKDSK (检查磁盘)来检查分区 E:。

4.2.4 删除备份图像

操作

PCU 引导启动期间 (在关闭和打开控制系统之后)

1. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。

显示基本菜单：

2. 按下键 <7>：备份/恢复

系统要求输入口令。

3. 输入 0 到 2 级的一个口令。

显示下列菜单：

请选择

- 1 用 GHOST 备份/修复硬盘
- 4 用 GHOST (本地) 备份/修复区
- 5 ADDM 备份/修复

9 返回主菜单

您选择 [1, 4, 5, 9] ?

4. 按下键 <4>。

显示下列菜单：

请选择

- 1 配置 GHOST 参数
- 2 分区备份，本地模式
- 3 分区修复，本地模式
- 4 E 区 (仅 Windows) 修复，本地模式
- 8 删除图像

9 返回上一级菜单

您选择 [1, 2, 3, 4, 5, 8, 9] ?

5. 按下键 <8>：删除图像

从提供的现有备份图像中选出要删除的图像。

进行选择后，仍将询问是否真要删除。

按“ Y ”删除备份图像。

6. 按下键 <9>：返回上一级菜单

返回上一级菜单。

4.2.5 确定紧急备份图像

操作

PCU 引导启动期间 (在关闭和打开控制系统之后)

1. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。

显示基本菜单：

2. 按下键 <7>：备份/恢复

系统要求输入口令。

3. 输入 0 到 2 级的一个口令。

显示下列菜单：

请选择

- 1 用 GHOST 备份/修复硬盘
- 4 用 GHOST (本地) 备份/修复区
- 5 ADDM 备份/修复

9 返回主菜单

您选择 [1, 4, 5, 9] ?

4. 按下键 <4>。

显示下列菜单：

请选择

- 1 配置 GHOST 参数
- 2 分区备份，本地模式
- 3 分区修复，本地模式
- 4 E 区 (仅 Windows) 修复，本地模式
- 8 删除图像

9 返回上一级菜单

您选择 [1, 2, 3, 4, 5, 8, 9] ?

5. 按下键 <1> : 配置 GHOST 参数 :

显示下列菜单 :

请选择

- 1 更改最多备份图像
- 2 定义紧急备份图像

9 返回上一级菜单

您选择 [1, 2, 9]?

6. 按下键 <2>。

询问将来 (直至撤销) 用作紧急备份图像的本地备份图像 :

请选择紧急图像 :

- 1 PCU 基础软件 V06.00.02 Win NT 4.0
[02-01-2000: 05:32:32pm]
- EMERG 2 数据备份 1 通过 HUGO
[02-03-2000; 04:53:24pm]
- 3 数据备份 2
[02-05-2000; 07:07:14pm]
- N 否 无紧急图像

9 返回上一级菜单

您选择 [1, 2, 3, N,9] ?

7. 选择所需的备份图像 , 例如 : 数据备份 2。

用于当前有效紧急备份图像的标识 EMERG , 将设置到所选的备份图像 :

请选择紧急图像 :

- 1 PCU 基础软件 V06.00.02 Win NT 4.0
[02-01-2000: 05:32:32pm]
- EMERG 2 数据备份 1 通过 HUGO
[02-03-2000; 04:53:24pm]
- 3 数据备份 2
[02-05-2000; 07:07:14pm]
- N 否 无紧急图像

9 返回上一级菜单

您选择 [1, 2, 3, N,9] ?

8. 按下键 <9> : 返回上一级菜单
返回上一级菜单。

4.2.6 分区 E: 上的紧急备份 : 录入

概述

如果在系统异常终止或者失控关闭后，PCU Windows 无法再引导启动，分区 E: 通过确定的备份图像来修复。

- 基于本地备份图像中的一个
必要：0 到 2 级的口令。
- 基于紧急备份图像（只要确定了一个本地备份图像作为紧急备份图像。）
必要：0 到 3 级的口令
通过紧急备份图像可以直接在主菜单中触发分区 E 的数据修复。

注意

在 D:\Eboot 中准备了用于生成紧急引导盘的软件。

操作

PCU 引导启动期间（在关闭和打开控制系统之后）

1. 在引导启动管理器中选择“服务菜单” [不可见的菜单项]。
显示基本菜单：
2. 按下键 <E> : 修复 Windows 分区 E (紧急)
系统要求输入口令。
3. 输入 0 到 3 级的一个口令。
 - 系统
 - 厂商
 - 维修
 - 用户

结果

出现小心警告：“Windows 分区 E: 被覆盖”而且再次询问后分区 E: 的重新录入将被触发。
分区 E 数据修复结束后，出现信息修复结束。

下次 Windows 进行引导启动时，通过检查程序 CHKDSK (检查磁盘)来检查分区 E:。

4.3 一个 TCU 配置时的数据备份/修复

应用

用一个或者多个 TCU 配置时，不能用服务程序在 DOS 界面上执行数据备份/数据修复。

为此可以使用服务桌面上与程序 "ServiceCenter" 的链接。其中实现的备份/修复功能和在 DOS 下调用时一样，例如：创建图形、重新录入，标记为紧急图像等。

概述

通过双击服务中心，打开下列选择：

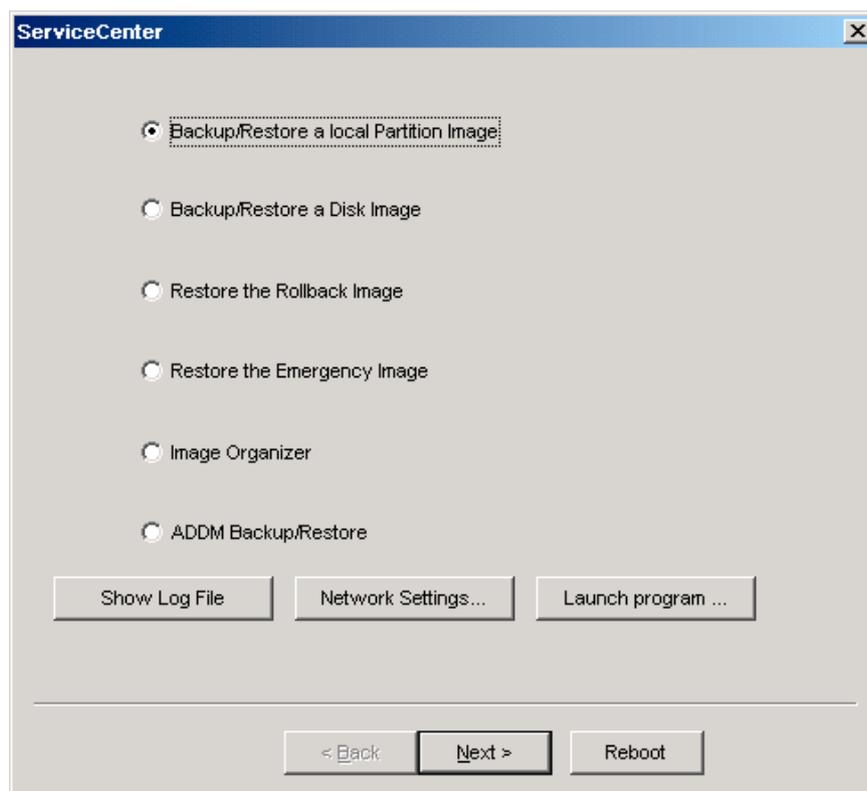


图 4-9 选择 ServiceCenter

备份/修复一个本地分区图像

选择该项，在本地分区 D 上创建或录入一个或多个分区的图形。

用“下一个 >”打开下一个选择窗口：这里可以单独选择分区 C、E 或者 F:如果保存了备份并希望以后重新修复，推荐创建完整的图形（C 和 E 和 F）。

启动备份前，输出每个分区的大小。例如：为了得到合适的数据包用于 CD ROM 存档，还可以在选项下确定所需的大小。

备份/修复磁盘图像

选择该项，在网络驱动器上创建或录入一个或多个分区的图形。

用“下一个

>”打开下一个选择窗口：如果已经设置与一个使能的驱动器存取建立网络连接，选择网络共享，并输入驱动器名称和文件名。

- 在更改网络设置下，可以重新设置通讯参数。
- 在“选项”下，从备份文件或者多个数据包间进行选择并确定大小。

使用 Ghost 软件的帮助 Ghost Cast Client 建立一个连接（例如：对于一个 1：1 的连接）；为此必须把带许可证的 Ghost 软件安装到目标驱动器。

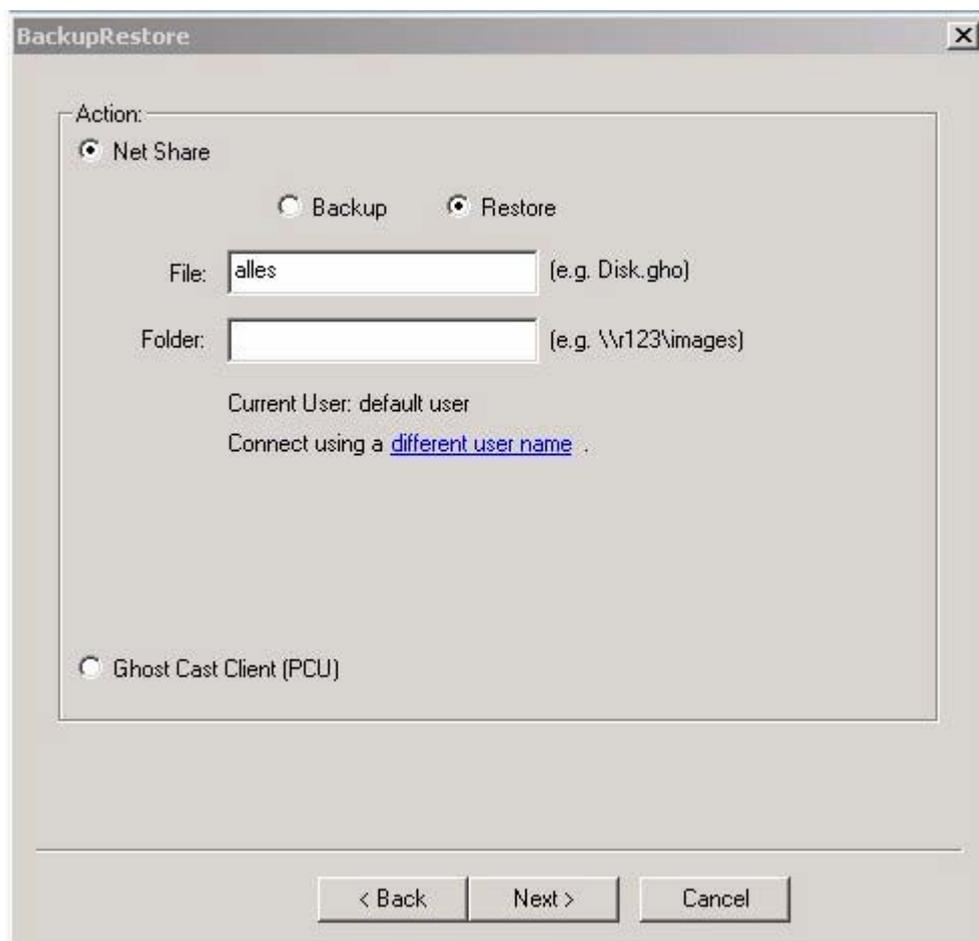


图 4-10 通过网络驱动器保存硬盘

修复压缩图形

选择该项，录入压缩图像（C、E 和 F 上最后一次保存/最新的图像）。

修复紧急图像

选择该项，录入一个或多个分区的紧急图像。

该图像适用的情况，如：当系统异常终止或者失控关闭后无法再引导启动时，而且包括分区 C 和 E。

该图像不能损害已安装的应用程序的运行，即：所有在 F: 盘上安装的应用程序的注册信息都必须包含在该图像中。都必须包含在该图像中。

为了标识紧急图像，使用“功能图像组织器”。

图像组织器

选择该项，从显示的列表中标识一个图像作为紧急图像或者删除现有的图像。

ADDM 备份/修复

选择该项，在 D 区或者网络驱动器上创建一个软件 ADDM (A&D 数据管理)的图像，或者重新录入该图像。

显示备份/修复记录文件

这里打开 bacres.txt 文件，对所有备份作了记录。

输出记录文件 (功能服务菜单“指令记录”)

可以通过引导启动目录读出记录文件。将提供一个引导启动菜单，当控制系统出现一个 HMI 系统软件的异常中断事件时 (PLC: DB19, 字节 0, 位 6)。该位必须由 PLC 在用户接口上设置 (象例如键禁用、黑屏控制...) 并且再次撤消。

参考文献： HMI 高级开机调试手册

索引

A

ADDM (A&D 数据管理), 96

B

BIOS

- 修改预设置, 22
- 启动 Setup, 17
- 系统参数, 18

D

DHCP 服务器, 7

DNS

- 域名服务器, 23
- 服务器, 65

DOS Shell, 29

E

Ethernet

- 连接, 7, 69

H

HMI 环境

- 原尺寸, 27
- 当前, 27

HMI-Explorer, 29

P

PCU

- IP 地址, 7
- 交货状态, 5
- 名称 (预设置), 7

PCU 网络名称, 77

S

ServiceCenter (维修中心), 93

SIMATIC STEP 7

- 授权, 67

SITOP 显示器

- 参数化, 41
- 配置, 40

T

TCP/IP

- 设置, 63
- 通讯, 7

U

USB 接口, 39

USV 模块, 39

W

WINS 服务器, 65

从

从所有网络驱动器断开, 23

从软盘安装, 23

分

分区图像, 93

反

反转图像, 93

启

- 启动 OEM 程序, 37
- 启动画面
 - 切换, 32
 - 存放位置, 31

回

- 回复控制台, 26

备

- 备份/恢复, 93
- 备份/恢复数据, 69
- 备份分区, 81, 83
- 备份图像, 89

安

- 安装 HMI 程序, 52
- 安装 SIMATIC STEP 7, 66

恢

- 恢复 CD, 65

故

- 故障记录, 38

文

- 文件系统
 - FAT16, 6
 - NTFS, 6

显

- 显示网络驱动器, 23

更

- 更改 DNS 扩展, 23
- 更改 IP 地址, 23
- 更改 TCP/IP 设置, 23
- 更改域名, 23
- 更改子网掩码, 23
- 更改机床名称, 23

- 更改用户名称, 23
- 更改网关, 23
- 更改网络设置, 23

服

- 服务桌面
 - 启动, 30
 - 网络运行, 64
- 服务菜单
 - 功能, 23
 - 参数化, 25

激

- 激活记录, 38
- 激活防火墙, 8

版

- 版本
 - 检查软件, 29
 - 软件组件, 5

用

- 用户 (预设置), 8
- 用户类型, 9
- 用户组, 10

监

- 监控 HMI, 45

硬

- 硬盘
 - 分区, 6
 - 备用件, 79
 - 手动更换, 79

磁

- 磁盘图像, 93

管

- 管理网络驱动器, 23

系

系统设置, 8

紧

紧急图像, 93

紧急备份, 92

紧急备份图像, 90

维

维修桌面

背景, 33

缓

缓冲时间, 42

缓冲蓄电池, 39

网

网络运行, 65

补

补充安装 Windows 组件, 65

触

触发记录, 23

软

软件组件, 5

运

运行记录仪记录文件, 96

连

连接

Ethernet, 69

并行, 69

连接到网络驱动器, 23

A

附录

A.1 资料反馈

本资料将在质量以及易于用户理解等方面不断改善。
您的建议和改善意见也为我们提供支持，请发送建议至：

电子邮件：<mailto:docu.motioncontrol@siemens.com>

传真：[+49 \(0\) 9131 / 98 - 63315](tel:+49(0)913179863315)
请使用手册末页的传真样表。

寄 SIEMENS AG A&D MC MS1 邮政信箱 3180 D-91050 Erlangen (爱尔兰根) 传真 : +49 (0) 9131 / 98 - 63315 (文献资料)	寄信人
	名称 :
	公司/单位地址
	街道 :
	邮编 : 地区 :
	电话 : /
传真 : /	
建议和/或更正	