

STEP 7 V10.5 SP2

自述	1
安装	2
新手上路	3
TIA 门户简介	4
编辑项目	5
组态设备和网络	6
对 PLC 进行编程	7
可视化过程	8
使用在线和诊断功能	9
源文件	10

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。

本文档使用“我的文档管理器”系统创建。

创建者 Sebastian Sander 创建日期 30.06.2010.

可以通过以下方式访问“我的文档管理器”系统 <http://www.automation.siemens.com/MDM>

请注意以下法律说明：

本文档摘录自西门子所编写的产品文献，由用户使用“我的文档管理器”免费系统自行整理。

附录中有已用文档目录，这些文档可通过以下链接在西门子主页上查找：<http://support.automation.siemens.com>

用户有责任通过 <http://support.automation.siemens.com> 定期检查文档，以确保其内容的时效性。

用户为自行整理的文档承担责任和风险，对此西门子不承担任何法律责任。

若“我的文档管理器”的补充使用条件未明确允许可以传播给第三方，本文档仅供用户个人和内部使用。

本文档的使用须遵循“我的文档管理器”的补充使用条件，点击以下链接查看其内容：<https://www.automation.siemens.com/mdm/ExtendedAGB>

目录

1	自述	13
1.1	常规说明	13
1.1.1	常规说明	13
1.1.2	安装说明	15
1.1.3	使用实例项目	16
1.1.4	显示通信接口	17
1.2	STEP 7 自述文件	18
1.2.1	使用注意事项	18
1.2.2	组态设备和网络	18
1.2.3	对 PLC 进行编程	20
1.3	WinCC 自述文件	47
1.3.1	安装	47
1.3.2	使用注意事项	49
1.3.3	移植	49
1.3.4	工程系统	50
1.3.5	运行系统	55
1.3.6	HMI 设备	55
2	安装	57
2.1	安装的系统要求	57
2.1.1	关于系统要求的说明	57
2.1.2	STEP 7 的系统要求	57
2.2	授权	58
2.3	开始安装	58
2.4	开始卸载	60
3	新手上路	61
3.1	Totally Integrated Automation Portal 的基本功能	61
3.1.1	总览	61
3.1.2	创建项目	69
3.1.3	配置硬件和网络	70
3.1.4	对 PLC 编程	78
3.1.5	组态工艺功能	84
3.1.6	组态可视化	88
3.1.7	将项目数据装载到设备	93
3.1.8	使用在线功能	95

4	TIA 门户简介	99
4.1	用户界面和操作	99
4.1.1	特定于操作系统的特性	99
4.1.2	启动、设置和退出 TIA Portal	101
4.1.3	用户界面的布局	103
4.1.4	调整用户界面	117
4.1.5	键盘快捷键	119
4.2	信息系统的帮助	123
4.2.1	信息系统的常规备注	123
4.2.2	使用帮助	126
4.2.3	禁用工具提示层叠项的显示	129
4.2.4	安全指南	129
5	编辑项目	133
5.1	项目的基本知识	133
5.2	创建新项目	133
5.3	打开项目	134
5.4	保存项目	134
5.5	关闭项目	135
5.6	删除项目	135
5.7	比较项目数据	136
5.7.1	比较编辑器概述	136
5.7.2	在线/离线比较	138
5.7.3	使用比较结果过滤器	138
5.7.4	运行详细比较	139
5.7.5	更新比较结果	139
5.7.6	指定动作	140
5.7.7	同步比较动作	140
5.8	编译项目数据	141
5.8.1	编译项目数据的常规信息	141
5.8.2	编译项目数据	142
5.9	在设备上装载项目数据	142
5.9.1	下载的常规信息	142
5.9.2	将项目数据从项目树下载到设备	143
5.9.3	将项目数据装载到可访问设备	144
5.10	打印项目内容	145
5.10.1	文档设置	145
5.10.2	创建打印预览	145
5.10.3	打印项目数据	146
5.11	移植项目	147
5.11.1	移植项目	147
5.11.2	显示项目历史	147
5.11.3	显示移植的日志文件	148

5.12	在项目中查找和替换	148
5.12.1	有关搜索功能的信息	148
5.12.2	搜索和替换	149
5.13	使用存储卡	150
5.13.1	关于存储卡的基本知识	150
5.13.2	添加用户自定义读卡器	151
5.13.3	访问存储卡	152
5.13.4	选择存储卡的卡类型	152
5.13.5	显示存储卡的属性	153
5.14	使用文本列表	154
5.14.1	文本列表	154
5.14.2	创建用户自定义文本列表	155
5.14.3	编辑用户自定义文本列表	155
5.14.4	编辑系统定义的文本列表	156
5.15	使用多语言项目	157
5.15.1	项目文本基本知识	157
5.15.2	选择项目语言	158
5.15.3	设置编辑语言	159
5.15.4	直接翻译文本	159
5.15.5	使用参考文本翻译文本	160
5.16	使用库	161
5.16.1	库基本知识	161
5.16.2	“库”任务卡	162
5.16.3	使用元素和部件视图	164
5.16.4	使用项目库	165
5.16.5	使用全局库	169
5.17	保护项目数据	176
5.17.1	项目数据的保护原理	176
5.18	使用交叉引用	176
5.18.1	使用交叉引用	176
6	组态设备和网络	177
6.1	硬件和网络编辑器	177
6.1.1	硬件和网络编辑器概述	177
6.1.2	网络视图	178
6.1.3	设备视图	180
6.1.4	打印硬件组态和网络组态	182
6.1.5	更改打印选项	183
6.1.6	巡视窗口	183
6.1.7	硬件目录	184
6.1.8	有关硬件组件的信息	186
6.1.9	启用产品支持	187
6.1.10	硬件和网络编辑器中的键盘操作	187
6.2	组态设备	188
6.2.1	基本知识	188

6.2.2	组态各个设备	195
6.3	联网设备	204
6.3.1	通信和网络	204
6.3.2	在网络视图中联网设备	205
6.3.3	表格式网络概览	207
6.3.4	在设备视图中联网设备	208
6.3.5	检查或更改网络参数和接口参数	208
6.3.6	更改联网设置	209
6.3.7	复制、剪切或删除子网	210
6.3.8	组态工业以太网	211
6.3.9	开放式用户通信	213
6.3.10	HMI 连接	221
6.4	组态操作	229
6.4.1	选择 CPU	229
6.4.2	更改模块的属性	231
6.4.3	CPU 属性	232
6.4.4	寻址模块	237
6.4.5	日时钟功能	240
6.4.6	高速计数器	241
6.4.7	点对点通信	245
6.4.8	使用时钟存储器	253
6.4.9	保护等级设置选项	254
6.4.10	装载组态	255
6.5	诊断硬件	256
6.5.1	硬件诊断概述	256
6.5.2	显示可组态模块属性的不可编辑值和当前值	262
6.5.3	显示动态模块属性的当前值	264
6.5.4	检查模块故障	267
6.5.5	更改模块或编程设备/PC 的属性	270
6.5.6	在 STOP 模式下诊断	273
6.6	参考	275
6.6.1	创建未指定的 CPU	275
6.6.2	开放式用户通信	276
7	对 PLC 进行编程	287
7.1	S7-1200 CPU 的功能说明	287
7.1.1	工作模式	287
7.1.2	存储区	293
7.1.3	I/O 数据区	297
7.1.4	程序执行的基础知识	298
7.2	创建用户程序	309
7.2.1	编程基本知识	309
7.2.2	声明 PLC 变量	363
7.2.3	创建和管理块	374
7.2.4	对块进行编程	388
7.2.5	比较块	482

7.2.6	编译块	486
7.2.7	下载块	488
7.2.8	保护块	496
7.3	显示程序信息	498
7.3.1	可用程序信息概述	498
7.3.2	显示分配列表	498
7.3.3	显示调用结构	505
7.3.4	显示从属性结构	511
7.3.5	显示 CPU 资源	516
7.4	显示交叉引用	519
7.4.1	关于交叉引用的常规信息	519
7.4.2	交叉引用列表的结构	520
7.4.3	显示交叉引用列表	521
7.4.4	在巡视窗口中显示交叉引用	522
7.5	测试用户程序	523
7.5.1	测试用户程序的基本知识	523
7.5.2	使用程序状态进行测试	523
7.5.3	使用监视表格测试	527
7.6	编程实例	558
7.6.1	LAD 编程实例	558
7.6.2	FBD 编程实例	567
7.7	使用工艺对象	576
7.7.1	使用运动控制	576
7.7.2	使用 PID Compact	600
7.8	参考	617
7.8.1	指令	617
7.8.2	扩展指令	870
8	可视化过程	945
8.1	移植	945
8.1.1	移植	945
8.1.2	移植基础知识	945
8.1.3	移植期间的对象支持	946
8.1.4	移植 WinCC flexible 2008 项目	949
8.1.5	数据类型的移植	951
8.2	使用画面	953
8.2.1	基础知识	953
8.2.2	使用对象	969
8.2.3	使用文本列表和图形列表	1005
8.2.4	画面动态化	1019
8.2.5	使用功能键	1028
8.2.6	使用层	1038
8.2.7	使用库	1044
8.2.8	显示操作对象	1055
8.2.9	为精简面板组态画面浏览	1078

8.3	使用变量	1080
8.3.1	基础知识	1080
8.3.2	使用变量	1084
8.3.3	使用数组	1099
8.3.4	使用周期	1103
8.3.5	显示变量	1104
8.4	使用报警	1106
8.4.1	基础知识	1106
8.4.2	使用报警	1115
8.4.3	操作运行系统中的报警	1138
8.4.4	参考	1144
8.5	使用配方	1173
8.5.1	基础知识	1173
8.5.2	元素和基本设置	1178
8.5.3	在运行系统中显示和编辑配方	1181
8.5.4	组态配方	1184
8.5.5	在运行系统中使用配方	1192
8.6	使用用户管理	1198
8.6.1	用户管理的应用领域	1198
8.6.2	用户管理的形式	1199
8.6.3	元素和基本设置	1200
8.6.4	设置用户管理	1205
8.6.5	参考	1222
8.6.6	实例	1222
8.7	使用系统函数	1231
8.7.1	基础知识	1231
8.7.2	使用函数列表	1233
8.7.3	实例	1238
8.7.4	参考	1241
8.8	使用调度程序	1294
8.8.1	调度程序的应用领域	1294
8.8.2	使用任务和触发源	1295
8.8.3	元素和基本设置	1295
8.8.4	实例	1299
8.9	使用连接	1300
8.9.1	基础知识	1300
8.9.2	使用连接	1302
8.9.3	用户数据区	1309
8.9.4	调试组件	1328
8.10	导入和导出项目数据	1329
8.10.1	导入和导出项目数据	1329
8.10.2	导入和导出配方	1330
8.11	编译和装载	1334
8.11.1	编译和装载项目	1334
8.11.2	仿真项目	1341

8.11.3	维护 HMI 设备	1347
8.11.4	参考	1350
8.12	运行系统中的操作	1351
8.12.1	基础知识	1351
8.12.2	调试项目	1354
8.12.3	运行系统中的语言	1359
8.12.4	操作项目	1362
8.13	性能特点	1413
8.13.1	工程系统	1413
8.13.2	精简面板	1414
8.14	显示交叉引用	1416
8.14.1	关于交叉引用的常规信息	1416
8.14.2	显示交叉引用列表	1417
8.14.3	交叉引用列表的结构	1417
8.14.4	在巡视窗口中显示交叉引用	1418
9	使用在线和诊断功能	1421
9.1	有关在线模式的常规信息	1421
9.2	在线访问	1422
9.3	显示可访问设备	1422
9.4	打开接口的属性	1423
9.5	设置以太网接口的参数	1423
9.5.1	设置工业以太网接口的参数	1423
9.5.2	显示操作系统参数	1424
9.5.3	将 PG/PC 接口连接到子网	1425
9.5.4	设置以太网接口的参数	1425
9.5.5	分配临时 IP 地址	1426
9.5.6	管理临时 IP 地址	1427
9.6	建立和取消在线连接	1427
10	源文件	1429
	索引	1431

Tabellen

表格 7-1	按符号名称排序	304
表格 7-2	按数字名称排序	306
表格 7-3	位操作数	536
表格 7-4	字节操作数	536
表格 7-5	字操作数	536
表格 7-6	双字操作数	537
表格 7-7	位操作数	537
表格 7-8	字节操作数	538
表格 7-9	字操作数	538

表格 7-10	双字操作数	538
表格 8-1	下表列出了用符号来显示各功能键的分配：	1029
表格 8-2	10000 - 打印机报警	1146
表格 8-3	20000 - 全局脚本报警	1146
表格 8-4	30000 - IFwSetValue 的报警： SetValue()	1147
表格 8-5	40000 - 线性转换报警	1147
表格 8-6	50000 - 数据服务器报警	1147
表格 8-7	60000 - Win32 函数报警	1147
表格 8-8	70000 - Win32 函数报警	1148
表格 8-9	80000 - 归档报警	1150
表格 8-10	90000 - FDA 报警	1152
表格 8-11	110000 - 离线函数报警	1153
表格 8-12	120000 - 趋势报警	1153
表格 8-13	130000 - 系统信息报警	1154
表格 8-14	140000 - 连接报警： chns7： 连接类型 + 设备	1154
表格 8-15	150000 - 连接报警： chnAS511： 连接	1156
表格 8-16	160000 - 连接报警： IVar (WinLC)/OPC： 连接	1156
表格 8-17	170000 - S7 诊断报警	1157
表格 8-18	180000 - 其它/通用报警	1157
表格 8-19	190000 - 变量报警	1158
表格 8-20	190100 - 区域指针报警	1159
表格 8-21	200000 - PLC 协调报警	1159
表格 8-22	200100 - PLC 用户版本报警	1160
表格 8-23	210000 - PLC 作业报警	1160
表格 8-24	220000 - WinCC 通道适配器报警	1160
表格 8-25	230000 - 视图报警	1161
表格 8-26	240000 - 授权报警	1163
表格 8-27	250000 - S7 强制报警	1163
表格 8-28	260000 - 密码报警	1164
表格 8-29	270000 - 系统事件	1165
表格 8-30	280000 - DPHMI 报警连接	1166
表格 8-31	290000 - 配方系统事件	1167
表格 8-32	300000 - Alarm_S 报警	1170
表格 8-33	310000 - 报表系统事件	1170
表格 8-34	320000 - 报警	1171
表格 8-35	330000 - GUI 报警	1171
表格 8-36	350000 - GUI 报警	1172

自述

1.1 常规说明

1.1.1 常规说明

本自述文档中的信息优先于其它文档中的说明。

请仔细阅读以下说明，因为它们包括了安装与使用方面的重要信息。请在安装前阅读这些说明。

安全设置

要使软件包能够在 TIA Portal 中运行，需要在安装期间修改系统的安全设置：

- 在 Windows 防火墙中把 TCP 的端口 4410 排除之外。
- 安装文件夹中的以下子文件夹将对所有用户共享：Portal V10.5\Data

安装新的 .Net 版本或 .Net 服务包

- 在编程设备/PC 上安装新的 .Net 版本或 .Net 服务包之前，需要先关闭 TIA Portal。
- 请在安装成功新的 .Net 版本或新的 .Net 服务包后再重新启动 TIA Portal。

修复已有的 STEP 7 Basic V10.5 安装时出错

如果已安装 WinCC flexible 2008 SP1 和 STEP 7 Basic V10.5，使用 DVD 上的安装程序启动 STEP 7 修复安装操作时，会出现错误。

要修复安装，请将 STEP 7 Basic V10.5 DVD 的内容复制到硬盘中，并在这里使用安装程序重新启动修复。

有关处理的说明

- 如果上次使用的项目列表中的项目位于未连接的网络驱动器上，则打开“项目”(Project) 菜单时可能会有延迟现象。
- 插入 CPU 时，如果同时打开了项目编辑器，则可能需要等待一段时间。如果是在新创建的项目中插入第一个 CPU，则此过程通常需要更长时间。为了能更快地继续工作，应在插入 CPU 之前关闭项目编辑器。

- 如果运行某项功能需要很长时间（例如装载 CPU），在 VISTA 中可能会显示消息“应用程序未响应”(Application is not responding)。若出现这种情况，请等候直到正确完成该功能。
- 如果您安装了使用 IntelliPoint 的 Microsoft 鼠标，则可能会发现它将组件叠加在标题栏的按钮上。在这种情况下，请卸载 Microsoft 的 IntelliPoint 软件。
- 使用 NVIDIA 显卡时启用“虚拟桌面”(Virtual Desktop) 选项可能会出现问题。在这种情况下，请禁用 NVIDIA 图形驱动程序的“nView 虚拟桌面管理器”(nView virtual desktop manager)。

通过远程桌面使用 TIA Portal

原则上，可以通过远程桌面连接来使用 TIA Portal。但是，组态期间应避免与桌面客户机断开连接。在极少数情况下，这可能会导致软件用户界面没有反应。

如果遇到此情况，请在桌面客户机上按以下步骤操作。

1. 打开 Windows 任务管理器，关闭“rdpclip.exe”进程。
2. 在命令提示中键入“rdpclip.exe”以重新启动该进程。

请注意，剪贴板的当前内容会丢失。然后可正常继续组态。为安全起见，有机会时尽早重新启动 TIA Portal。

多次打开 TIA Portal

如果正在运行 TIA Portal 的几个应用程序，且这几个应用程序依次持续变为激活状态，则您可直接切换到另一个应用程序或者使用组合键 <ALT+Tab> 解决问题。

有关 SD 卡的说明

西门子事先已将 SD 卡格式化和初始化，使其可用于 S7-1200 模块。不得覆盖此格式，否则 S7-1200 模块将不再接受这张卡。因此不允许使用 Windows 工具进行格式化。

存在未关闭的强制作业时的行为

注意，即使您已经将新项目装载到 SD 卡中，仍然会保留激活的强制作业。因此，在您从 CPU 上取下 SD 卡之前或者使用新项目重写 PC 上的 SD 卡之前，应当先删除所有的强制作业。

关闭 Windows XP 或激活屏幕保护程序时的问题

Windows XP 使用 ACPI（高级配置和电源管理接口）关闭计算机或进入待机模式。系统在处理新安装的工具期间，ACPI 无法激活屏幕保护程序，或者退出工具后无法正常关闭 Windows XP。

如果 TIA Portal 正在运行，计算机的待机功能会被禁用。要使计算机进入待机模式，必须先退出 TIA Portal。

下文说明了在“电源选项属性”(Power Options Properties) 中可用来通过“休眠”(Hibernate) 功能设置计算机待机模式的几个可选设置：

1. 在 Windows XP 中，通过选择“开始 > 设置 > 控制面板 > 电源选项”(Start > Settings > Control Panel > Power Options) 打开“电源选项属性”(Power Options Properties)，然后选择“休眠”(Hibernate) 选项卡。选中“启用休眠”(Enable hibernation) 复选框。
2. 切换到“高级”(Advanced) 选项卡。在对话框域“电源按钮”(Power buttons) 中单击“在合上便携式计算机时:”(When I close the lid of my portable computer:) 下的下拉列表框 并选择“休眠”(Hibernate) 选项。

3. 然后单击“在按下计算机电源按钮时:”(When I press the power button on my computer:) 下的下拉列表框 并选择“关机”(Shut down) 选项。
4. 单击“应用”(Apply) 并单击“确定”(OK) 进行确认。
5. 之后, 重新启动 PC。

如果关闭计算机时遇到问题, 则确保 TIA Portal 已完全关闭。

1. 从任务栏上的快捷菜单中, 选择“任务管理器”(Task Manager)。
2. 如果在“进程”(Processes) 选项卡中看到进程“Siemens.Automation.ObjectFrame.FileStorage.Server.exe”, 则等到该进程关闭。
3. 然后可以关闭计算机。

关于 TIA Portal 的常见问题解答

有关 TIA Portal 的常见问题解答, 可访问 <http://support.automation.siemens.com>。

1.1.2 安装说明

内容

在线帮助中未包含的信息。

STEP 7 Basic V10.5 SP2 的安装要求

除了安装说明中列出的要求外, 还必须满足以下条件:

- 安装网上下载的 SP2 版本要求有 STEP 7 Basic V10.5。
- SP2 试用版与 STEP 7 Basic V10.5 不兼容。您不能在已经安装了一个 STEP 7 Basic V10.5 版本的计算机上安装该版本。

使用“start.exe /unattendedmode”进行安装

如果要在几台计算机上进行相同的安装, 可使用安装程序将所有的设置保存在一个 INI 文件中。

1. 使用“开始 > 运行”(Start > Run) 打开 Windows 命令提示符。
2. 如果要创建 INI 文件, 则使用“start.exe/recordmode”启动安装程序。在对话框中选择要用于安装的设置。在出现授权对话框后, 安装程序将关闭。安装程序关闭时未执行任何安装。全部设置均存储在“SIA_Auto.ini”文件中, 该文件保存在“我的文档”(My Documents) 文件夹内。
3. 如果要基于一个 INI 文件进行安装, 则使用“start.exe/unattendedmode”启动安装程序。程序将会在“我的文档”(My Documents) 或者“InstData\Resources”文件夹中寻找“SIA_Auto.ini”文件。

找到 INI 文件后, 将采用其中包含的设置执行安装。

如果未找到 INI 文件, 则会出现提示消息。

在批处理中集成使用“start.exe /unattendedmode”安装

要通过批处理启动使用“start.exe/unattendedmode”安装, 可以根据需要更改“SIA_Auto.ini”文件的参数。

- SuppressReboot

安装过程结束时禁止重新启动，无论是否有必要。

- SuppressLicenseDialog
禁止显示许可证请求对话框。
- SuppressErrorDialog
禁止显示错误消息。
- SuppressDoneDialog
显示安装结束时禁止显示完成对话框。

在 Windows XP 下使用土耳其语区域和语言选项安装 STEP 7 Basic V10.5

在 Windows XP 下，如果已经将区域和语言选项设置成“土耳其语”，将会取消 STEP 7 Basic V10.5 的安装。在这种情况下，请将区域和语言选项从“土耳其语”更改为“英语”或“德语”。

1. 使用以下一个命令在 Windows 下打开控制面板：
 - “开始 > 控制面板”(Start > Control Panel) (Windows XP 的“开始”菜单)
 - “开始 > 设置 > 控制面板”(Start > Settings > Control Panel) (经典开始菜单)
2. 打开“区域和语言选项”(Regional and Language Options)。
3. 选择“区域选项”(Regional Options) 选项卡。
4. 在“标准和格式”(Standards and formats) 中，从下拉菜单中选择“德语”(German) 或“英语”(English)。
5. 单击“应用”(Apply) 并单击“确定”(OK) 进行确认。
6. 重启 PC 以使设置生效。此时即可继续安装 STEP 7 Basic V10.5。
7. 完成安装后，可将区域和语言设置改回 (如步骤 1 到 4 所述)“土耳其语”。

删除

即使安装了 SQL Server 2005 的完整版本，在极少数情况下删除程序时也可能导致计算机冻结。在这种情况下，请断开计算机同网络的连接，以继续删除过程。

1.1.3 使用实例项目

内容

在线帮助中未包含的信息。

简介

在安装数据介质中有一个实例项目，可帮助用户迅速了解如何使用 TIA Portal 中的项目。用户可编辑该实例项目以满足自己的需要。

步骤

要使用实例项目，请按以下步骤操作：

1. 将安装介质插入相应的驱动器。

2. 浏览到文件夹“<Drive>\Documents\Examples\DEMO Project S7-1200”。
3. 将“DEMO Project S7-1200”文件夹复制到本地驱动器。
4. 打开 TIA Portal。
5. 在“项目”(Project) 菜单中，选择“打开”(Open) 命令。
将打开“打开项目”(Open project) 对话框，其中包括最近所用项目的列表。
6. 单击“浏览”(Browse) 按钮并浏览到本地驱动器上的“DEMO Project S7-1200”文件夹。
7. 选择“DEMO Project S7-1200.ap10”文件。
8. 使用“打开”(Open) 确认选择。
实例项目随即打开并可进行编辑。
可以随时从安装数据介质中重新复制该实例项目。

1.1.4 显示通信接口

内容

在线帮助中未包含的信息。

简介

只有在安装 TIA Portal 时计算机上已存在的通信接口才会显示在 TIA Portal 中。如果已在计算机上安装 TIA Portal，然后安装新的 CP (通信处理器)，则该 CP 会被操作系统检测到并显示在 Windows 设备管理器中，但不会显示在 TIA Portal 的项目树下的“在线访问”(Online access) 中。

步骤

要在 TIA Portal 中显示后来安装的通信处理器，请按以下步骤操作：

1. 如果插入设备后 Windows“硬件更新向导”(Hardware Update Wizard) 打开，则安装/更新相关驱动程序。
2. 关闭 TIA Portal。
3. 选择“开始 > 设置 > 控制面板 > 设置 PG/PC 接口”(Start > Settings > Control Panel > Set PG/PC Interface)，然后单击“确定”(OK) 关闭应用程序。
4. 重新启动 TIA Portal。

结果

此时，硬件已存在并可使用，且通信接口显示在“在线访问”(Online access) 下。

1.2 STEP 7 自述文件

1.2.1 使用注意事项

内容

在线帮助中未包含的信息。

在线运行

不允许同时在线运行 STEP 7 和 STEP 7 Basic。

时间错误中断 OB 的编号

在线帮助的一些主题中提到了时间错误中断 OB280。该时间错误中断 OB 的正确编号是 80。

组态和分配模块参数

可以在 <http://support.automation.siemens.com> 下找到可以使用 STEP 7 Basic V10.5 组态和分配的模块的总览。

1.2.2 组态设备和网络

1.2.2.1 为 CM 1241

内容

在线帮助中未包含的信息。

XON 和 XOFF 的值

如果为 CM 1241 (RS-232) 通信模块启用了“XON/XOFF”流控制，则可以输入相同的 XON 和 XOFF 字符值。然而，从技术角度讲是不切实际的。因此应该使用不同的 XON 和 XOFF 值。

1.2.2.2 什么是开放式用户通信？

开放式用户通信的唯一连接 ID

如果您知道连接伙伴，您将需要在开放式用户通信的连接设置中为连接 ID 输入一个唯一值。连接设置不会检查连接 ID 是否唯一，在您创建新连接时也不会为连接 ID 输入默认设置。

1.2.2.3 关于在线和诊断的说明

内容

在线帮助中未包含的信息。

设置诊断文本的语言

诊断文本的显示语言与创建项目时设置的用户界面语言相同。如果要以其它语言显示诊断文本，请转到相应项目的项目树中的“语言和资源 > 项目语言”(Language & Resources > Project languages)。选中其它语言的复选框。然后编译与诊断相关的设备。诊断文本随即会以新设置的用户界面语言显示。

在线访问期间更改编辑语言后显示事件文本

如果单击“在线 > 可访问设备 > 更新”(Online > Accessible devices > Update)，然后在项目树的“语言和资源 > 项目语言”(Language & Resources > Project languages) 中设置不同的编辑语言，或者在“选项 > 设置”(Options > Settings) 中更改用户界面语言，则在“在线 > 在线和诊断 > 诊断缓冲区”(Online > Online & Diagnostics > Diagnostics buffer) 将不再为 CPU 显示任何事件文本。再次单击“在线 > 可访问设备 > 更新”(Online > Accessible devices > Update)。随即将再次显示这些文本。

来自高速计数器和脉冲发生器的诊断数据

对于尚未激活的高速计数器和脉冲发生器，包含设备概览的表格将显示具有以下含义的诊断图标：“由于当前在线组态数据与离线组态数据不同，因而无法获得诊断数据。”

打开输入/输出的在线和诊断视图

在硬件和网络编辑器中，可使用组合键 <Ctrl+D> 调用与所选设备上下文相关的“在线和诊断”功能。可在设备概览表格中选择整个 CPU 或者选择各输入/输出模块，然后使用组合键 <Ctrl+D> 打开相应的在线和诊断视图。如果已经在 CPU 的表格式设备视图中选择了某个集成输入/输出的相应行，则也可以打开集成输入/输出的在线和诊断视图。

在线和诊断视图中的语言

如果通过可用节点列表中的某个设备启动在线和诊断视图，在极少数情况下，在线和诊断视图中的事件不以用户界面语言显示。要以正确的语言显示事件，必须在相应项目中将编辑语言设置为与用户界面语言相同。重新启动 TIA Portal。

硬件检测后进行在线连接

如果针对尚未指定的 CPU 执行“在线 > 硬件检测”(Online > Hardware detection) 命令，将不会装载该 CPU 的在线组态。如果不将硬件检测发现的组态装载到 CPU 中，则设备视图和网络视图将总是显示离线组态和在线组态之间有差异。尽管实际 CPU 和离线 CPU 的 MLFB 完全相同，在线和诊断视图中也显示不同的组态。

分配 IP 地址

如果通过在线和诊断功能使用“功能 > 分配 IP 地址”(Functions > Assigning an IP address) 直接将一个 IP 地址分配给 PLC，则该 IP 地址将被永久设置，即使重新启动或电源故障，也保持不变。

1.2.2.4 循环时间说明

违反循环监视时间

在循环时间首次超过循环监视时间时，CPU 将尝试启动时间错误中断 OB (OB 80)。如果 CPU 中没有时间错误中断 OB，CPU 将切换到“RUN”模式。如果在同一个循环中，循环时间第二次超出循环监视时间，CPU 将切换到“STOP”模式。

1.2.2.5 编译脉冲发生器的硬件

内容

在线帮助中未包含的信息。

脉冲发生器被禁用时进行编译

如果脉冲发生器未激活，但在编译硬件期间仍显示错误消息“无法选中 PTO 脉冲发生器。相关高速计数器组态错误”，则请按以下步骤操作：

1. 禁用高速计数器。
2. 激活脉冲发生器并将操作模式设置为“PTO”。
3. 禁用脉冲发生器。
4. 重新编译硬件。

1.2.3 对 PLC 进行编程

1.2.3.1 关于 PLC 编程的一般说明

内容

在线帮助中未包含的信息。

删除在线块或者下载到设备后丢失保持性数据

如果删除在线块或将某个项目元素（例如程序块、数据块或硬件配置）下载到 CPU，则 CPU 在下次切换到 RUN 模式时将执行冷启动。除了会删除输入、初始化输出和删除非保持性存储器之外，还将删除保持性存储区。

其后的每次 STOP 到 RUN 切换均执行暖启动（在此过程中不会删除保持性存储器）。

更新可用节点列表中的块文件夹

请注意，可用节点列表中块文件夹的内容只会在关闭并再次打开相应块文件夹后进行更新。为确保在更改在线程序后显示最新内容，请关闭块文件夹，然后再次将其打开。

以多重背景的方式调用块

只能以多重背景方式调用随 STEP 7 V10.5 提供的库中包含的功能块。用户创建的任何功能块均不能以多重背景方式调用。

IEC 检查

- 默认情况下会禁用“IEC 检查”(IEC check) 选项。
- 无论“IEC 检查”(IEC check) 的设置是什么，都无法在同一条指令中连接 REAL 数据类型的操作数和 DWORD 数据类型的操作数。您必须使用“CONVERT”指令执行显式转换。

全局库

在产品 DVD 的“<驱动器>\Documents\AdditionalDocuments”目录下可找到有关全局库的信息。

MODBUS 库

STEP 7 V10.5 SP2 中更新了指令“MB_SLAVE”。

如果已经在项目中使用了“MB_SLAVE”V1.0，在安装 SP2 后，您必须使用最新版本“MB_SLAVE”V1.1 手动替换该版本。

为此，请按以下步骤操作：

1. 删除该项目所有块中的“MB_SLAVE”V1.0。
2. 从项目库中删除“MB_SLAVE”V1.0。
3. 在全部需要的位置插入“MB_SLAVE”V1.1。
4. 编译项目。

LAD 和 FBD 功能框的程序状态

如果 LAD/FBD 功能框的 ENO 未连接，则在某些情况下，将无法显示功能框的状态，例如，使用以下指令时即是如此

- SCALE
- NORMALIZE
- MOVE

PTO/PWM 输出的过程映像

请勿在过程映像区使用 PTO/PWM 输出（例如，用于用户程序中的访问、在线功能或在 HMI 中）过程映像区的更新速率比信号变化的速率低很多。过程映像区中的显示不能反映信号流的实际情况。

移动信号板后符号常量丢失

将信号板移到另一个设备后，不会创建任何符号常量。由于所需常量不存在，这将影响块的编程。编译期间，将生成与缺少常量有关的报警。必须先在信号板上禁用硬件中断然后重新激活，以便重新创建符号常量。

更改助记符设置

为避免在更改助记符设置后编译块时出现错误消息，请先保存项目，关闭项目，然后再重新打开项目。

1.2.3.2 LREAL 数据类型

1.2.3.2 使用 LREAL 数据类型

内容

在线帮助中未包含的信息。

简介

在一些指令中除可以使用 REAL (32 位) 数据类型之外，还可以使用 LREAL (64 位) 数据类型表示浮点数。只有在已设置为纯符号寻址的块中才能使用 LREAL 数据类型。

在指令中使用

下表列出了可使用 LREAL 数据类型的指令：

运算	助记符	描述
比较器	CMP ==	查询第一个比较值是否等于第二个比较值。
	CMP <>	查询第一个比较值是否不等于第二个比较值。
	CMP >=	查询第一个比较值是否大于或等于第二个比较值。
	CMP <=	查询第一个比较值是否小于或等于第二个比较值。
	CMP >	查询第一个比较值是否大于第二个比较值。
	CMP <	查询第一个比较值是否小于第二个比较值。
	- OK -	查询某个变量的值是否为有效的浮点数。
	- NOT_OK -	查询某个变量的值是否为无效的浮点数。

运算	助记符	描述
移动	MOVE	当使能输入 EN 的信号状态为“1”时，将输入 IN 的内容复制到输出 OUT。
	MOVE_BLK	将输入 IN 存储区（源区域）的内容复制到输出 OUT 的存储区（目标区域）。使用参数 COUNT 指定要复制到目标区域的元素数量。
	UMOVE_BLK	将输入 IN 存储区（源区域）的内容连续复制到输出 OUT 的存储区（目标区域）。使用参数 COUNT 指定要复制到目标区域的元素数量。
	FILL_BLK	用输入 IN 的值填充输出 OUT 的存储区（目标区域）。从输出 OUT 指定的地址开始填充目标区域。重复的复制操作次数使用参数 COUNT 指定。
	UFILL_BLK	用输入 IN 的值连续填充输出 OUT 的存储区（目标区域）。从输出 OUT 指定的地址开始填充目标区域。重复的复制操作次数使用参数 COUNT 指定。

运算	助记符	描述
数学函数	ADD	将输入 IN1 的值与输入 IN2 的值相加，并通过输出 OUT 输出总和 ($OUT = IN1 + IN2$)。
	SUB	从输入 IN1 的值中减去输入 IN2 的值，并通过输出 OUT 输出差值 ($OUT = IN1 - IN2$)。
	MUL	将输入 IN1 的值与输入 IN2 的值相乘，并通过输出 OUT 输出乘积 ($OUT = IN1 * IN2$)。
	DIV	将输入 IN1 的值除以输入 IN2 的值，并通过输出 OUT 输出商 ($OUT = IN1 / IN2$)。
	NEG	更改输入 IN 的值符号，并通过输出 OUT 输出结果。
	ABS	计算某个数的绝对值。
	SQR	计算浮点数的平方。
	SQRT	计算浮点数的平方根。
	LN	计算浮点数的自然对数。
	EXP	计算底数为 e 的浮点数指数值。
	SIN	计算浮点数的正弦值。这里的浮点数代表一个以弧度为单位的角。
	COS	计算浮点数的余弦值。这里的浮点数代表一个以弧度为单位的角。
	TAN	计算浮点数的正切值。这里的浮点数代表一个以弧度为单位的角。
	ASIN	计算定义范围为 $-1 \leq \text{输入值} \leq 1$ 的浮点数的反正弦值。在这种情况下，结果是一个以弧度为单位的角。
	ACOS	计算定义范围为 $-1 \leq \text{输入值} \leq 1$ 的浮点数的反余弦值。在这种情况下，结果是一个以弧度为单位的角。
	ATAN	计算浮点数的反正切值。在这种情况下，结果是一个以弧度为单位的角。
FRAC	确定输入 IN 的值的的小数位。	
EXPT	以输入 IN2 的值为指数，对输入 IN1 的值取幂。	

运算	助记符	描述
转换器	CONVERT	读取参数 IN 的数据，然后根据指定的数据类型对其进行转换。
	ROUND	将输入 IN 的值舍入为最接近的整数，并通过输出 OUT 输出结果。
	CEIL	将输入 IN 的值向上取整为相邻整数，并通过输出 OUT 输出结果。
	FLOOR	将输入 IN 的值向下取整为相邻整数，并通过输出 OUT 输出结果。
	TRUNC	选择输入 IN 的浮点数的整数部分，并将其输出到输出 OUT 中（不带小数位）。

1.2.3.2 LREAL (浮点数)

内容

在线帮助中未包含的信息。

描述

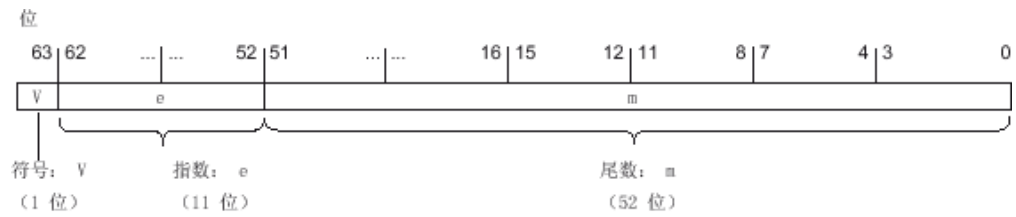
LREAL 数据类型的变量的长度为 64 位，用于显示浮点数。LREAL 数据类型的变量由以下三部分组成：

- 符号：符号由位 63 的信号状态确定。位 63 可以是“0”(正数)或“1”(负数)。
- 以 2 为底的 11 位指数：该指数加一个常数 (基数 +1023)，因此其范围为 2047。
- 52 位尾数：仅显示尾数的小数部分。不存储尾数的整数部分，因为其在有效值范围内始终等于“1”。

下表列出了 LREAL 变量的属性：

长度 (位)	格式	取值范围	输入值实例
64	符合 IEEE 754 标准的浮点数	-1.7976931348623158e+308 到 -2.2250738585072014e-308	1.0e-5
	浮点数	±0 +2.2250738585072014e-308 到 +1.7976931348623158e+308	1.0

下图所示为 LREAL 变量的结构：



参见

使用 LREAL 数据类型 (页 22)

1.2.3.3 重置 IEC 定时器

1.2.3.3 ---

内容

在线帮助中未包含的信息。

简介

除上述指令外，LAD 中还有一条用于重置 IEC 定时器的指令。

符号

<操作数>

--- (RT) ---

参数	数据类型	存储区	描述
<操作数>	DB	D	在 RLO =“1”时重置的 IEC 定时器的 DB。

描述

使用“重置 IEC 定时器”运算将 IEC 定时器重置为“0”。

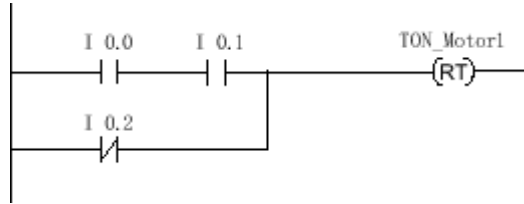
只有线圈输入端的逻辑运算结果 (RLO) 为“1”时，才会执行该操作。如果电流通过线圈 (RLO 为“1”)，则 IEC 定时器 DB 的参数被设置为“0”。如果线圈输入端的逻辑运算结果为“0” (没有信号流过线圈)，则参数保持不变。

该运算不会影响 RLO。线圈输入端的 RLO 将立即发送到线圈输出端。

放置

“重置 IEC 定时器”运算可以放置在程序段中的任何位置。

实例



如果满足以下条件之一，将重置 IEC 定时器“TON_Motor1”：

- 输入 I 0.0 与 I 0.1 为“1”。
- 输入 I 0.2 的信号状态为“0”。

1.2.3.3 ---

内容

在线帮助中未包含的信息。

简介

除上述指令外，FBD 中还有一条用于重置 IEC 定时器的指令。

符号



参数	数据类型	存储区	描述
<操作数>	DB	D	在 RLO =“1”时重置的 IEC 定时器的 DB。

描述

使用“重置 IEC 定时器”运算将 IEC 定时器重置为“0”。

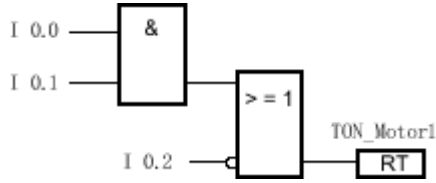
只有当功能框输入端的逻辑运算结果 (RLO) 为“1”时，才会执行该操作。如果功能框输入端的信号状态为“1”，则 IEC 定时器 DB 的参数被设置为“0”。如果功能框输入端的逻辑运算结果为“0”，则参数保持不变。

该运算不会影响 RLO。功能框输入端的 RLO 将直接传送到功能框输出上。

放置

“重置 IEC 定时器”运算可以放置在逻辑串中的任何位置。

实例



如果满足以下条件之一，将重置 IEC 定时器“TON_Motor1”：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 I 0.2 的信号状态为“0”。

1.2.3.4 读取域

1.2.3.4 读取域

简介

除上述指令外，LAD 中还有一条用于读取域中各个元素的指令。

注意

插入指令“读取域”

插入“读取域”指令的方法是，从“收藏夹”(Favorites) 窗格中拖出一个空功能框，并从空功能框的下拉列表中选择该指令。

符号

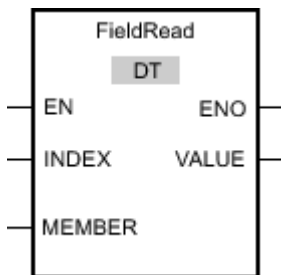


图 1-1

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出

参数	数据类型	存储区	说明
INDEX	DINT	I、Q、M、D、L 或常数	要读取的域元素的下标
MEMBER	ARRAY 变量的元素的全部基本数据类型	I、Q、M、D、L	要读取的第一个域元素
VALUE	所有基本数据类型	I、Q、M、D、L	要将域元素写入其中的变量

可以从“DT”下拉列表中选择用于该指令的数据类型。

说明

“读取域”指令可用于从参数 MEMBER 所显示的域中读取指定元素，并将其内容传送到参数 VALUE 的变量中。在参数 INDEX 中指定要读取的域元素的下标。在参数 MEMBER 中输入要读取的第一个域元素。

参数 MEMBER 中域元素的数据类型和参数 VALUE 中变量的数据类型必须与“读取域”指令的数据类型相匹配。

只有使能输入 EN 的信号状态为“1”时，才能启动“读取域”指令。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 在参数 MEMBER 输入的域中未定义参数 INDEX 指示的域元素。
- 处理过程中发生溢出之类的错误。

实例

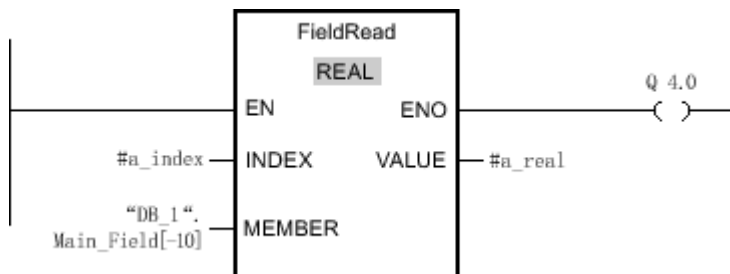


图 1-1

参数	变量	值
INDEX	a_index	4
MEMBER	"DB_1".Main_Field[-10]	数据块“DB_1”中域“Main_Field[-10..10] of REAL”的第一个元素
VALUE	a_real	域“Main_Field[-10..10] of REAL”中下标为 4 的元素

从域“Main_Field[-10...10] of REAL”中读取下标为 4 的域元素，并将其写入到变量“a_real”中。通过参数 INDEX 的值指定要读取的域元素。

1.2.3.4 读取域

简介

除上述指令外，LAD 中还有一条用于读取域中各个元素的指令。

注意

插入指令“读取域”

插入“读取域”指令的方法是，从“收藏夹”(Favorites) 窗格中拖出一个空功能框，并从空功能框的下拉列表中选择该指令。

符号

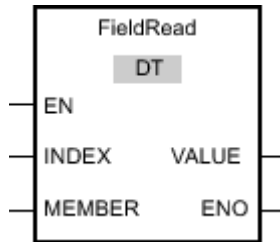


图 1-1

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
INDEX	DINT	I、Q、M、D、L 或常数	要读取的域元素的下标
MEMBER	ARRAY 变量的元素的全部基本数据类型	I、Q、M、D、L	要读取的第一个域元素
VALUE	所有基本数据类型	I、Q、M、D、L	要将域元素写入其中的变量

可以从“DT”下拉列表中选择用于该指令的数据类型。

说明

“读取域”指令可用于从参数 MEMBER 所显示的域中读取指定元素，并将其内容传送到参数 VALUE 的变量中。在参数 INDEX 中指定要读取的域元素的下标。在参数 MEMBER 中输入要读取的第一个域元素。

参数 MEMBER 中域元素的数据类型和参数 VALUE 中变量的数据类型必须与“读取域”指令的数据类型相匹配。

只有使能输入 EN 的信号状态为“1”时，才能启动“读取域”指令。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 在参数 MEMBER 输入的域中未定义参数 INDEX 指示的域元素。
- 处理过程中发生溢出之类的错误。

实例

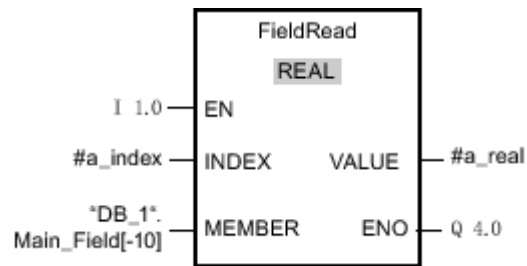


图 1-1

参数	变量	值
INDEX	a_index	4
MEMBER	"DB_1".Main_Field[-10]	数据块“DB_1”中域“Main_Field[-10..10] of REAL”的第一个元素
VALUE	a_real	域“Main_Field[-10..10] of REAL”中下标为 4 的元素

从域“Main_Field[-10..10] of REAL”中读取下标为 4 的域元素，并将其写入到变量“a_real”中。通过参数 INDEX 的值指定要读取的域元素。

1.2.3.5 写入域

1.2.3.5 写入域

简介

除上述指令外，LAD 中还有一条用于写入域中各个元素的指令。

注意**插入指令“写入域”**

插入“写入域”指令的方法是，从“收藏夹”(Favorites) 窗格中拖出一个空功能框，并从空功能框的下拉列表中选择该指令。

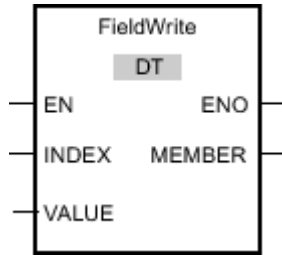
符号

图 1-1

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
INDEX	DINT	I、Q、M、D、L 或常数	要写入的域元素的下标
VALUE	所有基本数据类型	I、Q、M、D、L 或常数	将复制其内容的变量
MEMBER	ARRAY 变量的元素的全部基本数据类型	I、Q、M、D、L	要写入的第一个域元素

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

“写入域”指令用于将参数 VALUE 的变量内容传送到在参数 MEMBER 的域的特定元素中。通过参数 INDEX 的值指定上述域元素的下标。在参数 MEMBER 输入要写入的第一个域元素。在参数 MEMBER 指定的域元素的数据类型和参数 VALUE 中变量的数据类型必须与“写入域”指令的数据类型相匹配。

只有使能输入 EN 的信号状态为“1”时，才能启动“写入域”指令。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 在参数 MEMBER 输入的域中未定义参数 INDEX 指示的域元素。
- 处理过程中发生溢出之类的错误。

实例

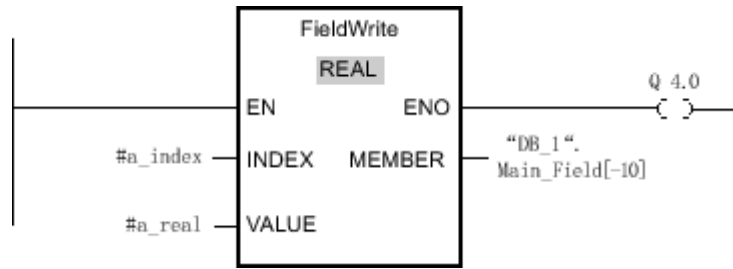


图 1-1

参数	变量	值
INDEX	a_index	4
VALUE	a_real	10.54
MEMBER	"DB_1".Main_Field[-10]	数据块“DB_1”中域 “Main_Field[-10..10] of REAL”的第一个元素

变量“a_real”的值“10.54”将写入到域“Main_Field[-10..10] of REAL”下标为 4 的域元素中。通过参数 INDEX 的值指定要接收变量“a_real”内容的域元素的下标。

1.2.3.5 写入域

简介

除上述指令外，LAD 中还有一条用于写入域中各个元素的指令。

注意**插入指令“写入域”**

插入“写入域”指令的方法是，从“收藏夹”(Favorites) 窗格中拖出一个空功能框，并从空功能框的下拉列表中选择该指令。

符号

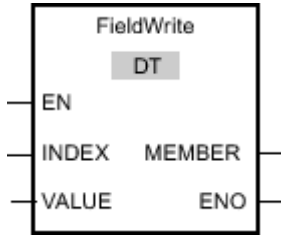


图 1-1

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
INDEX	DINT	I、Q、M、D、L 或常数	要写入的域元素的下标
VALUE	所有基本数据类型	I、Q、M、D、L 或常数	将复制其内容的变量
MEMBER	ARRAY 变量的元素的全部基本数据类型	I、Q、M、D、L	要写入的第一个域元素

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

“写入域”指令用于将参数 VALUE 的变量内容传送到在参数 MEMBER 的域的特定元素中。通过参数 INDEX 的值指定上述域元素的下标。在参数 MEMBER 输入要写入的第一个域元素。

在参数 MEMBER 指定的域元素的数据类型和参数 VALUE 中变量的数据类型必须与“写入域”指令的数据类型相匹配。

只有使能输入 EN 的信号状态为“1”时，才能启动“写入域”指令。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 在参数 MEMBER 输入的域中未定义参数 INDEX 指示的域元素。
- 处理过程中发生溢出之类的错误。

实例

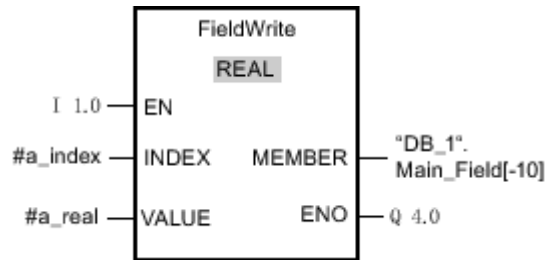


图 1-1

参数	变量	值
INDEX	a_index	4
VALUE	a_real	10.54
MEMBER	"DB_1".Main_Field[-10]	数据块“DB_1”中域“Main_Field[-10..10] of REAL”的第一个元素

变量“a_real”的值“10.54”将写入到域“Main_Field[-10..10] of REAL”下标为 4 的域元素中。通过参数 INDEX 的值指定要接收变量“a_real”内容的域元素的下标。

1.2.3.6 比较块

1.2.3.6 专有技术受保护的块

内容

在线帮助中未包含的信息。

比较专有技术受保护的块时的特别注意事项

如果将专有技术受保护的块下载到设备，则在安装该块时不会安装任何恢复信息。这意味着将无法再打开或读取已装载到设备上的专有技术受保护块的用户程序。即使使用正确的密码，也无法解除该限制。

由于无法再读取用户程序，因此无法详细比较专有技术受保护的块。

注意事项

既不能读取，也不能编辑从设备装载到项目中的那些专有技术受保护的块。建议在将专有技术受保护的块从设备装载到项目之前，先创建相关块离线版本的备份副本。

1.2.3.7 测试用户程序

1.2.3.7 使用监视表格测试

内容

在线帮助中未包含的信息。

对同一个 CPU 的多重访问

只有当 TIA Portal 已打开时才允许从 PG/PC 访问 CPU。不允许对同一个 CPU 的多重访问，这会导致错误。

使用触发器修改

使用触发器修改时（例如，永久性修改变量），如果当前正在重置 CPU (MRES)，则会中止现有控制作业。即使对监视表格对话框中询问是否停止使用触发器修改的提示回答“否”，控制作业也同样会被终止。

对浮点数取整

在监视表格中，浮点数以 IEEE 格式的二进制数形式存储。由于一些能在用户界面中显示的浮点数（实数、长实数）无法精确地转换成 IEEE 格式，因此可能会将浮点数取整。

如果浮点数已经因此被取整，然后被复制到监视表格中的另一个输入单元格中，则该取整操作可能会导致细微的偏差。

在存在激活的控制作业期间装载数据块

注意事项

在存在激活的控制作业期间装载经过更改的数据块，可能导致无法预料的运行状态。尽管数据块中的地址分配可能已发生变化，但是控制作业会继续控制之前指定的地址。因此，请在装载数据块之前完成激活的控制作业。

1.2.3.8 使用工艺对象

1.2.3.8 使用 PID Compact

内容

在线帮助中未包含的信息。

重启 CPU 后重新启动“PID_Compact”工艺对象

通电并重启 CPU 后，如果“PID_Compact”工艺对象上次的模式为自动模式，则工艺对象将切换到该模式。

可以禁用该自动模式的自动切换。为此，可将背景 DB 中“sb_RunModeByStartup”变量的值设置为 FALSE。

使用“PID_Compact”工艺对象的模拟受控变量输出

如果使用受控变量输出“Output”或“Output_PER”，则每次“PWM 限制”发生变化时都必须被更正为值 0.0。在“高级设置 > PWM 限制”(Advanced settings > PWM limits) 组态窗口中设置“PWM 限制”。

“PID_Compact”指令的参数不完整

如果“PID_Compact”指令的参数不完整 (三个红色问号)，在编译期间不会以错误形式指出。

在编译和下载到设备之前，请确保“PID_Compact”指令的参数是完整的且具有正确的值。

1.2.3.8 使用运动控制

1.2.3.8 关于运动控制的一般说明

内容

在线帮助中未包含的信息。

控制面板的响应时间

控制面板的响应时间取决于 CPU 的通信负载。请关闭“Portal”的所有其它在线窗口以保持较低的响应时间。

通过“MC_Reset”确认错误后启动运动命令

如果出现了必须使用运动控制指令“MC_Reset”确认的轴错误，则请执行以下操作：

1. 解决问题的原因。
2. 使用运动控制指令“MC_Reset”确认错误。
3. 在启动新运动命令之前检查以下信号状态：
 - 输出参数“Done”= TRUE
 - 用于工艺对象的变量 "Axis".StatusBits.Error = FALSE

使用到达硬件限位开关后自动反向功能主动回原点

如果满足以下组态条件，则到达硬件限位开关后将自动反向运动：“逼近速度”>“减小的速度”。

1.2.3.8 脉冲发生器

内容

在线帮助中未包含的信息。

脉冲发生器 (PTO) 的限制

将 CPU 1211C、CPU 1212C 和 CPU 1214C 的脉冲发生器与“轴”工艺对象结合使用时，会受到以下限制：

- 最小频率 2 Hz
- 最大频率 100 kHz (使用 20 kHz 的信号板时)
- 最小频率变化 (加速/减速) 0.28 Hz/s
- 最大频率变化 (加速/减速) 9500 MHz/s

1.2.3.8 硬件限位开关和参考点开关

内容

在线帮助中未包含的信息。

延时硬件限位开关和参考点开关

数字量输入的默认滤波时间设置为 6.4 ms。

如果将数字量输入用作硬件限位开关，则可能出现意外的延迟。

如果将数字量输入用作参考点开关，可能出现意外的延迟和偏差。根据逼近速度和参考点开关的电平设置而定，有可能检测不到参考点。

如果发生这种情况，请在数字量输入的设备组态中减少相应数字量输入的滤波时间。

1.2.3.8 运行时的轴组态

内容

在线帮助中未包含的信息。

在用户程序运行期间更改轴组态

在运行期间，可通过用户程序更改轴组态数据的选择。在用户程序中通过工艺对象的变量能够访问到轴组态数据中的以下内容：

- 硬件/软件限位开关的组态
 - "Axis".Config.PositionLimits_HW.Active
硬件限位开关的启用状态 (TRUE 表示启用)
 - "Axis".Config.PositionLimits_SW.Active

- 软件限位开关的启用状态 (TRUE 表示启用)
 - "Axis".Config.PositionLimits_SW.MinPosition
软件限位开关下限位
 - "Axis".Config.PositionLimits_SW.MaxPosition
软件限位开关上限位
- 更改在轴停止后启动新运动作业时生效。
- 动态值的组态
 - "Axis".Config.DynamicDefaults.Acceleration
轴的加速度
 - "Axis".Config.DynamicDefaults.Deceleration
轴的延迟
 - "Axis".Config.DynamicDefaults.EmergencyDeceleration
轴的紧急停止延迟
- 更改在启动新运动作业时生效。

1.2.3.8 调用运动控制指令

内容

在线帮助中未包含的信息。

调用运动控制指令

运动控制指令可能会被同一条使用相同背景块但优先级更高的运动控制指令所中断。

请采取以下措施避免此类中断：

- 在用户程序中避免同时执行使用相同背景块的运动控制指令 (例如 , 通过使用有条件的块调用) 。
- 对不同的优先级使用不同的背景数据块。

1.2.3.8 错误 ID 和错误信息列表

内容

在线帮助中未包含的信息。

错误 ID/错误信息列表

下表列出了可以显示在运动控制语句中的错误 ID 和错误信息。该列表将取代“错误 ID 和错误信息列表”小节。

运行故障时停止轴

错误 ID	错误信息	描述	解决方法
16#8000		驱动器错误,“驱动器准备就绪”故障	
	16#0001	-	使用“MC_Reset”语句确认错误;提供驱动器信号
16#8001		下限软件限位开关已被触发	
	16#000E	轴已到达该限位开关	通过“MC_Reset”语句确认错误;设置运动命令使轴沿正方向移动,脱离软件限位开关的范围
	16#000F	轴已到达该限位开关(紧急停止)	
	16#0010	轴已越过该限位开关(紧急停止)	
16#8002		上限软件限位开关已被触发	
	16#000E	轴已到达该限位开关	通过“MC_Reset”语句确认错误;设置运动命令使轴沿负方向移动,脱离软件限位开关的范围
	16#000F	轴已到达该限位开关(紧急停止)	
	16#0010	轴已越过该限位开关(紧急停止)	
16#8003		下限硬件限位开关已被触发	
	16#000E	轴已到达该限位开关	通过“MC_Reset”语句确认错误;设置运动命令使轴沿正方向移动,脱离硬件限位开关的范围
16#8004		上限硬件限位开关已被触发	
	16#000E	轴已到达该限位开关	通过“MC_Reset”语句确认错误;设置运动命令使轴沿负方向移动,脱离硬件限位开关的范围
16#8005		PTO 和 HSC 已被其它轴占用	
	16#0001	-	更正 PTO 和 HSC 的组态并将其下载到控制器

运行故障时不停止轴

错误 ID	错误信息	描述	解决方法
16#8200		轴未启用	
	16#0001	-	启用轴;重新启动命令
16#8201		轴已被另一个“MC_Power”语句启用	
	16#0001	-	仅通过一个“MC_Power”语句启用轴
16#8202		超出并行激活运动控制命令数的最大值(所有运动控制工艺对象最多 256 条命令)	
	16#0001	-	减少并行激活命令数;重新启动命令
16#8203		轴当前在“手动控制”模式(控制面板)下运行	
	16#0001	-	退出“手动控制”;重新启动命令
16#8204		轴未回原点	
	16#0001	-	通过“MC_Home”语句使轴回原点;重新启动命令

错误 ID	错误信息	描述	解决方法
16#8205		轴当前由用户程序控制 (该错误仅显示在控制面板中)	
	16#0001	-	使用“MC_Power”指令将轴锁定并再次在控制面板中选择“手动控制”
16#8206		尚未启用工艺对象	
	16#0001	-	使用“MC_Power”语句或在轴控制面板中启用轴。
16#8207		命令被拒绝	
	16#0016	正在执行主动回原点；无法启动被动回原点。	等待主动回原点完成执行，或者使用诸如“MC_Halt”的运动命令取消主动回原点。然后可启动被动回原点。

块参数错误

错误 ID	错误信息	描述	解决方法
16#8400		运动控制语句的“位置”(Position) 参数值不正确	
	16#0002	值使用了无效的数字格式	更正“位置”值；重新启动命令
	16#0005	该值超出值范围 (大于 $1e^{12}$)	
	16#0006	该值超出值范围 (小于 $-1e^{12}$)	
16#8401		运动控制语句的“Distance”参数值不正确	
	16#0002	值使用了无效的数字格式	更正“Distance”值；重新启动命令
	16#0005	该值超出取值范围 (大于 $1e^{12}$)	
	16#0006	该值超出取值范围 (小于 $-1e^{12}$)	
16#8402		运动控制语句的“Velocity”参数值不正确	
	16#0002	值使用了无效的数字格式	更正“Velocity”值；重新启动命令
	16#0008	超出最大速度	
	16#0009	该速度小于启动/停止速度	
16#8403		运动控制语句的“Direction”参数值不正确	
	16#0011	选取值无效	更正选取值；重新启动命令
16#8404		运动控制语句“MC_Home”的“Mode”参数值不正确	
	16#0011	选取值无效	更正选取值；重新启动命令
	16#0015	未组态主动/被动回原点	更正组态并将其下载到控制器；重新启动命令
	16#0017	虽然已禁用硬件限位开关，但还是激活了在该硬件限位开关处轴反向	将变量“Axis”.Config.PositionLimits_HW.Active 设为 TRUE 启用硬件限位开关，或者更正组态并将其下载到控制器；重新启动命令
16#8405		运动控制语句“MC_Power”的“StopMode”参数值不正确	
	16#0011	选取值无效	更正选取值；重新启动命令

错误 ID	错误信息	描述	解决方法
16#8406		不允许同时向前和向后点动	
	16#0001	-	防止参数“JogForward”和“JogBackward”的信号状态同时为 TRUE；重新启动命令。
16#8407		仅允许使用“MC_Power”语句更改锁定的轴。	
	16#0001	-	锁定当前轴；当 Status = FALSE 与 Busy = FALSE 时可更改并再次启用该轴。

组态错误

错误 ID	错误信息	描述	解决方法
16#8600		脉冲发生器 (PTO) 的组态无效	
	16#000B	地址无效	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
16#8601		高速计数器 (HSC, High-Speed Counter) 的组态无效	
	16#000B	无效地址	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
16#8602		“驱动器启用”(Drive enable) 输出的组态无效	
	16#000D	无效地址	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
16#8603		“驱动器准备就绪”(Drive ready) 输入的组态无效	
	16#000D	无效地址	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
16#8604		“电机每转的脉冲数”(Pulses per motor revolution) 值无效	
	16#000A	该值小于或等于 0	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
16#8605		“电机每转的距离”(Distance per motor revolution) 值无效	
	16#0002	值使用了无效的数字格式	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#000A	该值小于或等于 0	
16#8606		“启动/停止速度”(Start / stop velocity) 值无效	
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	
	16#0007	启动/停止速度大于最大速度	
16#8607		“最大速度”(maximum velocity) 值无效	

错误 ID	错误信息	描述	解决方法
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	
16#8608		“加速度”(Acceleration) 值无效	
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	
16#8609		“减速度”(Deceleration) 值无效	
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	
16#860A		“急停减速度”(Emergency stop deceleration) 值无效	
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0003	值超出硬件限值	
	16#0004	值小于硬件限值	
16#860B		下限软件限位开关的位置值无效	
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0005	该值超出取值范围 (大于 $1e^{12}$)	
	16#0006	该值超出取值范围 (小于 $-1e^{12}$)	
	16#0007	下限软件限位开关的位置值大于上限软件限位开关的位置值	
16#860C		上限软件限位开关的位置值无效	
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0005	该值超出取值范围 (大于 $1e^{12}$)	
	16#0006	该值超出取值范围 (小于 $-1e^{12}$)	
16#860D		下限硬件限位开关的地址无效	
	16#000C	下降沿的地址无效	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#000D	上升沿的地址无效	
16#860E		上限硬件限位开关的地址无效	
	16#000C	下降沿的地址无效	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#000D	上升沿的地址无效	
16#860F		“参考点位移”(reference point shift) 值无效	

错误 ID	错误信息	描述	解决方法
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0005	该值超出取值范围 (大于 $1e^{12}$)	
	16#0006	该值超出取值范围 (小于 $-1e^{12}$)	
16#8610		“启动速度”(startup velocity) 值无效	
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0008	超出最大速度	
	16#0009	该速度小于启动/停止速度	
16#8611		“逼近速度”(approach velocity) 值无效	
	16#0002	值使用了无效的数字格式	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#0008	超出最大速度	
	16#0009	该速度小于启动/停止速度	
16#8612		参考点开关的地址无效	
	16#000C	下降沿的地址无效	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	16#000D	上升沿的地址无效	
16#8613		已启用到达硬件限位开关时主动回原点反向运动，但未组态相应的硬件限位开关	
	16#0001	-	更正该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴

内部错误

错误 ID	错误信息	描述	解决方法
16#8FFF		内部错误	
	16#F0**	-	通过断电和上电的方式重新启动 CPU

1.2.3.9 通信指令

1.2.3.9 关于通信指令的说明

内容

在线帮助中未包含的信息。

通过 ISO-on-TCP 建立通信连接

使用“S7-1200”控制器组态通过 ISO-on-TCP 进行的连接时，请在被动通信伙伴的 TSAP 扩展中仅使用 ASCII 字符。

发送和接收数据

- 如果在指令“TRCV_C”、“TSEND_C”、“TRCV”或“TSEND”的参数 DATA 中使用符号值，则参数 LEN 的值必须为“0”。
- 使用 TSEND_C 指令发送数据（参数 REQ 出现上升沿）时，为建立或保持连接，参数 CONT 的值必须为 TRUE。
- 使用 TRCV_C 指令接收数据（参数 EN_R 出现下降沿）时，为建立或保持连接，参数 CONT 的值必须为 TRUE。

TSEND_C

在线帮助中描述的参数 STATUS 的值有以下含义：

STATUS	描述
8086	参数 ID 超出允许范围。
8088	参数 LEN 的值与参数 DATA 中设置的接收区不匹配。
80A1	<ul style="list-style-type: none"> • 连接或端口已被用户使用。 • 通信错误： <ul style="list-style-type: none"> – 尚未建立指定的连接。 – 正在终止指定的连接。无法通过此连接进行传送。 – 正在重新初始化接口。
80A3	<ul style="list-style-type: none"> • 正尝试重新建立现有连接。 • 正在尝试终止不存在的连接。
80B3	不一致的参数分配：错误代码 W#16#80A0 到 W#16#80A2、W#16#80A4、W#16#80B4 到 W#16#80B9 对应的组错误
80C3	<ul style="list-style-type: none"> • 所有连接资源都在使用。 • 具有该 ID 的块正在一个具有不同优先级的组中处理。 • 内部缺乏资源。
80C4	临时通信错误： <ul style="list-style-type: none"> • 此时无法建立连接。 • 接口正在接收新参数或正在建立连接。 • 正通过 TDISCON 指令删除已组态连接。

除在线帮助中描述的值外，状态参数还可以是以下值：

STATUS	描述
80A0	错误代码 W#16#80A1 和 W#16#80A2 对应的组错误

STATUS	描述
80A2	本地端口或分布式端口已被系统使用。
80A4	分布式连接端点的 IP 地址无效，即它与本地伙伴的 IP 地址重复。
80B5	参数“active_est”错误。
80B6	连接描述数据块的“connection_type”参数存在参数分配错误
80B7	连接描述数据块的下列某个参数出错：block_length、local_tsap_id_len、rem_subnet_id_len、rem_staddr_len、rem_tsap_id_len、next_staddr_len

TRCV_C

在线帮助中描述的参数 STATUS 的值有以下含义：

STATUS	描述
8085	<ul style="list-style-type: none"> • 参数 LEN 大于最大允许值。 • 参数 LEN 或 DATA 的值在第一次调用后发生改变。
8086	参数 ID 超出允许范围。
8088	参数 LEN 的值与参数 DATA 中设置的接收区不匹配。
80A1	<ul style="list-style-type: none"> • 连接或端口已被用户使用。 • 通信错误： <ul style="list-style-type: none"> – 尚未建立指定的连接。 – 正在终止指定的连接。无法通过此连接进行传送。 – 正在重新初始化接口。

除在线帮助中描述的值外，状态参数还可以是以下值：

STATUS	描述
80A0	错误代码 W#16#80A1 和 W#16#80A2 对应的组错误
80A2	本地端口或分布式端口已被系统使用。
80A3	<ul style="list-style-type: none"> • 正尝试重新建立现有连接。 • 正在尝试终止不存在的连接。
80A4	分布式连接端点的 IP 地址无效，即它与本地伙伴的 IP 地址重复。
80A7	通信错误：在 TCON 完成前执行了 TDISCON。
80B3	不一致的参数分配：错误代码 W#16#80A0 到 W#16#80A2、W#16#80A4、W#16#80B4 到 W#16#80B9 对应的组错误
80B4	分布式连接端点的 IP 地址无效，即它与本地伙伴的 IP 地址重复。
80B5	参数“active_est”错误。
80B6	连接描述数据块的“connection_type”参数存在参数分配错误

STATUS	描述
80B7	连接描述数据块的下列某个参数出错：block_length、local_tsap_id_len、rem_subnet_id_len、rem_staddr_len、rem_tsap_id_len、next_staddr_len
80C3	<ul style="list-style-type: none"> 所有连接资源都在使用。 具有该 ID 的块正在一个具有不同优先级的组中处理。 内部缺乏资源。
80C4	临时通信错误： <ul style="list-style-type: none"> 此时无法建立连接。 接口正在接收新参数或正在建立连接。 正通过 TDISCON 指令删除已组态连接。
8722	参数“CONNECT”错误：无效源区域（数据块中未声明的区域）。
873A	参数“CONNECT”错误：无法访问连接描述（不能访问数据块）。
877F	参数“CONNECT”错误：内部错误

“CONDITIONS”数据类型

以下“CONDITIONS”数据类型的参数的允许取值范围为：

参数	取值范围	描述
MAXLEN	0 到 1024	消息中的最多字符数。如果超过消息的最大长度，将识别到消息结束。
N	1 到 1024	消息中长度域的偏移量。该值指定长度域起始字符的位置。

这些取值范围同样适用于指定消息结束时使用的相应硬件设置。

1.3 WinCC 自述文件

1.3.1 安装

内容

在线帮助中未包含的信息。

同时安装 STEP 7 V10.5 (包括 WinCC V10.5) 与其它 SIMATIC HMI 产品

STEP 7 V10.5 (包括 WinCC V10.5) 已经可以与下列产品同时安装

- WinCC flexible 2008 和 WinCC flexible 2008 SP1

- WinCC V6.2 SP3 或更高版本 (如果要在安装 STEP 7 V10.5 后再安装 WinCC V6.2 SP3 , 请阅读 [FAQ 30576253](#) 中的信息)。
- WinCC V7

STEP 7 V10.5 (包括 WinCC V10.5) 不可以与下列产品同时安装

- WinCC flexible 2004 和 WinCC flexible 2004 SP1
- WinCC flexible 2005 和 WinCC flexible 2005 SP1
- WinCC flexible 2007
- 早于 WinCC V6.2 SP2 的所有 WinCC 版本

为安装组态的系统时间

要成功执行 Microsoft SQL Express 的安装 , 安装计算机的系统时间需设置为当前时间。

安装或卸载 WinCC flexible 后丢失字体

安装或卸载 WinCC flexible 后 , 某些字体可能无法再使用。 在下列情况中会发生该问题 :

- 如果在安装了 STEP 7 V10.5 之后还安装了 WinCC flexible 2008。
- 如果在卸载了 STEP 7 V10.5 之后还卸载了 WinCC flexible 2008 或者 WinCC flexible 2008 SP1。

然而这些字体仍在计算机上 , 只需让 Windows 系统重新识别即可。 请执行以下步骤 :

1. 打开 Windows 资源管理器。
2. 在地址栏中输入“%WINDIR%\Fonts”。
将打开存放字体的文件夹。
3. 选择“文件 > 安装新字体...”(File > Install New Font...)
4. 在“文件夹”(Folders) 框中 , 选择 Windows 存储字体的文件夹。 在 Windows 资源管理器的地址栏中可看到该文件夹。
5. 在“字体列表”(List of fonts) 框中选择以下字体。
Siemens Sans Global
Siemens Sans Global Bold
Siemens Sans Global Bold Italic
Siemens Sans Global Italic
6. 单击“确定”(OK) 确认选择。

注意

在 Windows Vista 中 , 需要有管理员权限才能安装字体。

1.3.2 使用注意事项

内容

在线帮助中未包含的信息。

编译和装载

在 WinCC V10.5 中组态期间，项目不会持续在后台编译。

如果编译期间产生了内部错误或警告，则使用 HMI 设备快捷菜单中的“编译”>“软件 (全部重建)”命令来编译整个项目。

启动项目的生产操作之前，使用 HMI 设备快捷菜单中的“编译”>“软件 (全部重建)”命令对项目进行完全编译。

如果在项目中使用连接到控制变量的 HMI 变量，则使用快捷菜单中的“编译”>“软件”命令对所有修改的块进行编译，然后再编译 HMI 设备。

复制具有 HMI 连接的 HMI 设备

如果复制与 PLC 存在 HMI 连接关系的 HMI 设备，则新 HMI 设备中的 HMI 连接不会自动连接到同名的现有 PLC。本说明适用于项目内和项目间的复制操作。

要在新 HMI 设备中通过 HMI 变量访问 PLC 变量，必须在复制步骤后立即完成 HMI 组态。步骤如下：

1. 打开“设备和网络”(Devices & Networks) 编辑器。
2. 将新 HMI 设备与所需网络连接。
3. 打开连接表。
4. 选择新 HMI 设备的 HMI 连接。
5. 在“伙伴”(Partner) 下选择所需 PLC。

如果在复制 HMI 设备和完成连接之间编译新 HMI 设备或者连接其它 PLC 变量，则在某些情况下可能会创建到同一个 PLC 的其它 HMI 连接。将 HMI 变量与 DB 数组元素连接时更是如此。

1.3.3 移植

内容

在 WinCC V10.5 中可继续使用 WinCC flexible 2008 的项目。可以移植“精简面板”型的 HMI 设备。

不能将项目直接从 WinCC flexible 2008 SP1 移植到 WinCC V10.5。如果要在 WinCC V10.5 中继续使用此类项目，则必须首先将其移植到 WinCC flexible 2008。

更改报警类别的名称

与 WinCC flexible 不同的是，预定义报警类别的名称不依赖于当前使用的用户界面语言。移植期间，报警类别的名称按如下方式分配：

WinCC flexible	WinCC
错误	Alarms
系统	System
警告	Events

报警类别的名称可在移植后根据需要更改。

WinCC V10.5 中的项目语言

WinCC V10.5 并不支持在 WinCC flexible 中可使用的所有项目语言，例如阿拉伯语。如果移植后收到空项目，最好检查一下设置的编辑语言。不要在源项目中设置不支持用作编辑语言的项目语言。步骤如下：

1. 在 WinCC flexible 中，打开项目。
2. 例如，将编辑语言改为英语。
3. 保存项目。
4. 重新开始移植。

参见

移植期间的对象支持 (页 946)

→ *移植期间的对象支持 (13279660683/14023607307.htm)*

1.3.4 工程系统

1.3.4.1 画面和画面对象

内容

在线帮助中未包含的信息。

报警文本中输出域的文本格式

不能对变量和文本列表条目加下划线。

在两个项目或两台设备之间复制显示对象

例如，在 Project_1 的“全局画面”内组态报警窗口。复制报警窗口并将其粘贴到 Project_2 中的“全局画面”内。

粘贴后，报警窗口中一部分已启用报警类别会不再处于启用状态。

这种行为适用于以下显示对象：

- 报警窗口
- 报警指示灯

- 报警视图

在巡视窗口中显示交叉引用

巡视窗口在“关于 > 交叉引用”(About > Cross-reference) 选项卡中显示画面对象所使用的对象。

画面打开并且已选中一个对象。该对象中的某个 HMI 变量正在充当过程变量。

对象和链接的 HMI 变量显示在交叉引用中。使用该对象和 HMI 变量的所有位置均会列出。

如果 HMI 变量与 PLC 变量或 DB 变量互连，则使用互连的 PLC 变量或 DB 变量的位置都将显示出来。

发生报警时巡视窗口的“信息”(Info) 选项卡中的事件名称

对于巡视窗口中的某些报警，“信息”(Info) 选项卡中的事件名称与“属性”(Properties) 选项卡中的名称略有不同。

巡视窗口的“属性”(Properties) 选项卡中的名称	巡视窗口的“信息”(Info) 选项卡中的名称
已清除	ClearScreen
已装载	GenerateScreen
启用	Activate
更改	Change
在对话框打开时	ONMODALBEGIN
在对话框关闭时	ONMODALEND
用户更改	PASSWORD
画面切换	SCREEN
禁用	Deactivate
按下	Press
离开	Going
进入	Coming
违反“上限误差”限制	AboveUpperLimit
违反“下限误差”限制	BelowLowerLimit
单击	Click
报警回路	LoopInAlarm
释放	Release
报警缓冲区溢出	OVERFLOW
确认	Acknowledgement
运行系统停止	Shutdown
按下按键	KeyDown
释放按键	KeyUp

巡视窗口的“属性”(Properties) 选项卡中的名称	巡视窗口的“信息”(Info) 选项卡中的名称
接通	SwitchOn
关闭	SwitchOff
值改变	Change value

功能键的全局和局部分配

在“全局画面”(Global screen) 编辑器中，功能键 F1 分配有一个功能和一个图形。

在模板“Temp_1”中，对于功能键 F1 禁用选项“使用全局分配”(Use global assignment)。将不分配该功能键。

画面“Screen_1”基于模板“Temp_1”。选项“使用局部模板”(Use local template) 已启用。

在画面“Screen_1”中功能键 F1 的功能列表为空，但全局画面的图形可见。

要更改该行为，请按以下步骤操作：

1. 在画面“Screen_1”中启用功能键 F1。
2. 在巡视窗口中的“属性 > 常规”(Properties > General) 下禁用选项“使用局部模板”(Use local template)。
3. 如果图形仍然可见，则在列表中的“图形”(Graphic) 下单击。
4. 选择“<无>”(None)。

1.3.4.2 变量

内容

在线帮助中未包含的信息。

变量名

HMI 变量名不能以 @ 字符开头。

在 HMI 变量的使用位置显示已删除的数组元素

HMI 变量（例如 IO 域的过程值）的使用位置通常由变量名称指示。如果使用数组变量的元素，则该变量的名称将以方括号括起来的数组元素下标来扩展。

对于项目中不再包含的使用过的变量，变量的名称仍将显示在使用位置。该域将以红色背景显示以表示缺少变量。如果使用过的数组元素或数组变量已不存在，将仅在方括号中显示该数组元素的下标。不再显示变量名称。该域以红色高亮显示。在这种情况下，无法再根据使用位置确定关联的数组变量的名称。

如果不知道哪个数组变量曾链接到该使用位置，则可能需要再次链接数组元素。

如果变量或数组变量是基于引用创建的，则所选引用将自动关闭。

如果 HMI 变量与 PLC 变量的数组元素相连接，但该 PLC 变量已不在该项目中，则在“HMI 变量”(HMI tags) 编辑器中将发生相同的行为。

作为指针变量列表条目的数组变量

可以像使用字符串数据类型的变量一样使用字符数据类型的数组变量。

在“HMI 变量”(HMI tags) 编辑器中，不支持将字符数据类型的数组变量用作指针变量的列表条目。

1.3.4.3 报警系统和报警显示

内容

在线帮助中未包含的信息。

显示报警文本中的特殊字符

组态报警文本时，在工程系统中使用固定的字符集。该字符集允许用户在报警文本中使用大量的特殊字符。

在运行系统中使用与语言相关的字体来显示文本，例如，MS P Gothic、宋体。运行系统中使用的字体并不支持所有特殊字符。因此，有些特殊字符无法在运行系统中显示。

在具有报警文本的输出框中使用指针变量

在工程系统中，还可以在具有报警文本的输出框中使用指针变量。运行期间，这将导致报警显示错误，因为精简面板不支持使用指针变量。

1.3.4.4 系统函数

内容

在线帮助中未包含的信息。

1.3.4.5 配方

内容

在线帮助中未包含的信息。

1.3.4.6 用户管理

内容

在线帮助中未包含的信息。

导出和导入用户数据

无法导入或导出用户数据。

1.3.4.7 连接

内容

在线帮助中未包含的信息。

将“DTL”数据类型用于区域指针

将“DTL”数据类型用于区域指针“日期/时间”和“日期/时间 PLC”的组态。“DTL”数据类型支持纳秒级的时间戳信息。由于精简面板对时间戳信息的支持只能到毫秒级，所以使用区域指针时将遇到以下限制：

- 区域指针“日期/时间”
用于将时间信息从精简面板传送到 PLC，最小时间单位为 1 毫秒。“DTL”数据类型中微秒到纳秒的取值范围将用零填充。
- 区域指针“日期/时间 PLC”
用于将时间信息从 PLC 传送到精简面板，微秒到纳秒的区域将被忽略。时间信息将在面板中降至毫秒进行处理。

可用 HMI 连接数的限值

设备编译期间将显示一条错误消息，指示“设备和网络”(Devices & Networks) 编辑器中 HMI 连接的组态无效。原因可能是超出了 HMI 设备或 PLC 可用的最大连接数。

请检查可用的最大连接数。请查阅正在使用的设备的设备手册。

1.3.4.8 编译和装载

内容

在线帮助中未包含的信息。

ProSave 的安装不正确

如果在安装 ProSave 期间将数据装载到目标设备时或维护 HMI 设备期间，收到错误消息，则不能使用安装程序的修复功能纠正此错误。通过控制面板删除 ProSave。然后重新启动安装程序并安装“ProSave”组件。

检查地址参数

通过快捷菜单中的命令“编译”>“软件”编译项目树中的 HMI 设备期间，将不会检查 HMI 设备的地址参数，例如 IP 地址。如果要确保同样也检查地址参数，您需要在工具栏的“设备和网络”编辑器中编译 HMI 设备。

将数据下载到 PLC 时的错误消息

面板和 PLC 已连接且正在相互通信。

如果将数据从面板下载到 PLC 时访问变量，则在面板上将显示一条错误消息。

1.3.5 运行系统

1.3.5.1 在运行系统中操作的注意事项

内容

在线帮助中未包含的信息。



小心 以太网通信

在基于以太网的通信中，是由最终用户负责其数据网络的安全。不能保证设备在所有情况下都能正常运行；例如，遭受蓄意攻击就会导致设备过载。

用户视图中的特殊字符

在用户视图中输入名称或密码时，不允许使用特殊字符，如 / " § \$ % & ' ?。

语言行为 - 屏幕键盘的布局

屏幕键盘的布局不会在运行系统语言更改时切换。

变量值超过最大长度

通过 I/O 域将字符串输入到字符串变量中。如果字符串超过了已组态变量的数目，则会将此字符串缩短到组态的长度。

1.3.6 HMI 设备

1.3.6.1 常规说明

内容

在线帮助中未包含的信息。

精简面板的仿真

在报警文本中使用输出域来输出外部变量。输出域的内容在仿真期间始终显示为“0”。

使用真实的 PLC 连接进行仿真

仿真使用的访问点与 ES 系统的设置无关，只能在控制面板中使用“设置 PG/PC 接口”工具进行更改。如果一启动仿真 PLC 连接就终止而且出现了消息 140001，就应该通过“设置 PG/PC 接口”检查仿真使用的访问点。

1. 在 Windows “控制面板”中双击“设置 PC/PG 接口”。随即打开一个对话框。
2. 在“应用程序的访问点”区域选择“S7ONLINE”作为 HMI 的标准访问点。
3. 在“已使用的接口参数分配”区域选择该接口。
4. 单击“确定”退出对话框“设置 PG/PC 接口”。

装载无配方数据记录的项目

正在项目中使用配方。将无配方数据记录的项目传送到精简面板。

如果在工程系统中更改了配方结构并且设备已存有配方数据记录，则可能会发生不一致。

在这种情况下，请检查数据记录的一致性。设备并不会对所有结构性变化发出说明。

安装

2.1 安装的系统要求

2.1.1 关于系统要求的说明

各款产品的系统要求

系统要求可能因您所安装的产品而异。因此，您应当检查具体产品的系统要求。

如果要安装多个产品，请确保您的系统满足其中的最高要求。

显示 PDF 文件

为能够阅读所提供的 PDF 文件，需要与 PDF 1.7 (ISO32000-1:2008 PDF) 兼容的 PDF 阅读器。

2.1.2 STEP 7 的系统要求

2.1.2.1 STEP 7 的软件和硬件要求

安装的系统要求

下表列出了安装“SIMATIC STEP 7 Basic”软件包对软件和硬件的最低要求：

硬件/软件	要求
处理器类型	Pentium 4 1.7 GHz 或类似
RAM	Windows XP : 1 GB Windows Vista : 2 GB
可用硬盘空间	2 GB

硬件/软件	要求
操作系统 *	<ul style="list-style-type: none"> Windows XP (Home SP3、Professional SP3) Windows Vista (Home Premium SP1、Business SP1、Ultimate SP1)
显卡	32 MB RAM 32 位颜色深度
屏幕分辨率	1024 x 768
网络	10 M 以太网或更快
光驱	DVD-ROM

* 有关操作系统的详细信息，请参见 Microsoft Windows 或 Microsoft 主页上的帮助。

2.2 授权

授权的可用性

安装数据介质中提供了 TIA Portal 产品的授权，在安装过程中由 Automation License Manager (ALM) 自动安装授权。这意味着您不会收到单独的含授权的安装介质，并且无法将授权传送到其它计算机。如果卸载 TIA Portal，相应的授权也将自动卸载。

Automation License Manager (ALM)

Automation License Manager 用于管理授权。该管理器在安装数据介质中提供，并且在安装过程中会自动安装。如果卸载 TIA Portal，Automation License Manager 仍会安装在系统中。

2.3 开始安装

简介

软件包通过安装程序自动安装。将安装介质插入驱动器后，安装程序便会立即启动。

注意

首次在 Windows Vista 中安装 TIA Portal 时，将创建一个名为“TIA 工程师”(TIA-Engineer) 的用户组。当前安装 TIA Portal 的登录用户将自动被添加到该新用户组中。

要求

- 编程设备或 PC 的硬件和软件满足系统要求。

- 尚未在计算机上安装软件包。
- 所有打开的程序均已关闭。

步骤

要安装软件包，请按以下步骤操作：

1. 将安装介质插入相应的驱动器。
安装程序将自动启动，除非您在编程设备或 PC 上禁用了自动启动功能。
2. 如果安装程序没有自动启动，则双击“Start.exe”文件手动启动它。
将打开选择安装语言的对话框。
3. 根据需要选择安装程序对话框的语言，然后单击“下一步”(Next)。
将打开选择产品语言的对话框。
4. 选择产品用户界面的语言。
5. 如果要阅读产品说明，请单击“是，我要阅读产品信息”(Yes, I would like to read the Product Information) 按钮。
将打开包含产品说明的帮助文件。
6. 阅读完产品说明之后，请再次关闭帮助文件。
7. 单击“下一步”(Next) 按钮。
将打开选择产品组态的对话框。
8. 选择要用来安装所选产品的目录。 请注意，安装路径的长度不能超过 89 个字符。
9. 单击“下一步”(Next) 按钮。
下一个对话框将显示安装设置及授权协议的概述。
 - 1 检查所选的安装设置。 如果要进行任何更改，请单击“上一步”(Back) 按钮，直到到达想要在其中进行更改的对话框位置。
 - 1 阅读并接受所有授权条件。
 1. 单击“安装”(Install) 按钮。
 2. 安装随即启动。

注意

如果安装成功，屏幕上会显示一条有关安装成功的消息。 如果安装期间出现错误，则会显示一条错误消息，通知您错误类型。

- 1 可能必须重新启动计算机。 在这种情况下，请选择“是，立即重启计算机。”(Yes, restart my computer now.) 选项按钮。
 3. 单击“完成”(Finish) 或“重新启动”(Restart) 按钮。
 - 4.

结果

在您的计算机上安装了 TIA Portal 和您订购的产品和授权以及 Automation License Manager。

参见

安装的系统要求 (页 57)

开始卸载 (页 60)

2.4 开始卸载

简介

软件包可通过安装程序自动卸载。安装程序包含在软件安装包中，并且无需单独安装。安装程序启动后，便会引导您逐步完成整个卸载步骤。

步骤

要卸载软件包，请按以下步骤操作：

1. 使用“开始 > 设置 > 控制面板”(Start > Settings > Control Panel) 打开“控制面板”(Control Panel)。
2. 在控制面板中双击“添加或删除程序”(Add or Remove Programs)。将打开“添加或删除程序”(Add or Remove Programs) 对话框。
3. 在“添加或删除程序”(Add or Remove Programs) 对话框中，选择要删除的软件包，然后单击“删除”(Remove)。将打开选择安装语言的对话框。
4. 根据需要选择安装程序对话框的语言，然后单击“下一步”(Next)。将打开显示要卸载的产品的对话框。
5. 选择要卸载的产品。
6. 单击“下一步”(Next) 按钮。
7. 检查包含要卸载的产品的列表。如果要进行任何更改，请单击“上一步”(Back) 按钮。
8. 单击“卸载”(Uninstall) 按钮。卸载随即启动。
9. 可能必须重新启动计算机。在这种情况下，请选择“是，立即重启计算机。”(Yes, restart my computer now.) 选项按钮。
10. 单击“完成”(Finish) 或“重新启动”(Restart) 按钮。

注意

卸载软件包时，不会自动卸载 Automation License Manager，因为它要用于管理 Siemens AG 所提供产品的多种授权密钥。

新手上路

3.1 Totally Integrated Automation Portal 的基本功能

3.1.1 总览

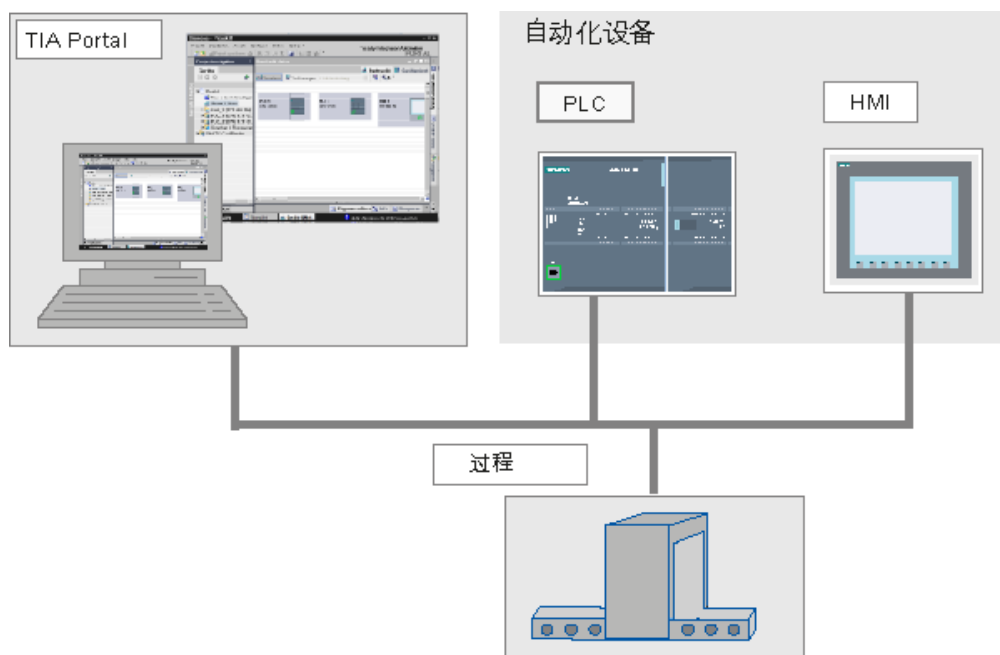
3.1.1.1 Totally Integrated Automation Portal 的基本功能

简介

Totally Integrated Automation Portal (TIA Portal) 在一个软件应用程序中集成了各种 SIMATIC 全集成自动化 (TIA, Totally Integrated Automation) 产品，利用这一软件您可以提高生产力和效率。所有 TIA 产品在 TIA Portal 中的协作顺畅且一目了然，能够支持您创建各种领域的自动化解决方案。

典型的自动化解方案通常包含以下内容：

- PLC 使用用户程序来控制机器。
- 用户使用 HMI 设备操作和监视过程。



任务

TIA Portal 可用来帮助您组态自动化解决方案。关键的组态步骤为：

- 创建项目
- 配置硬件
- 设备联网
- 对 PLC 编程
- 组态可视化
- 装载组态数据
- 使用在线和诊断功能

新的工程概念

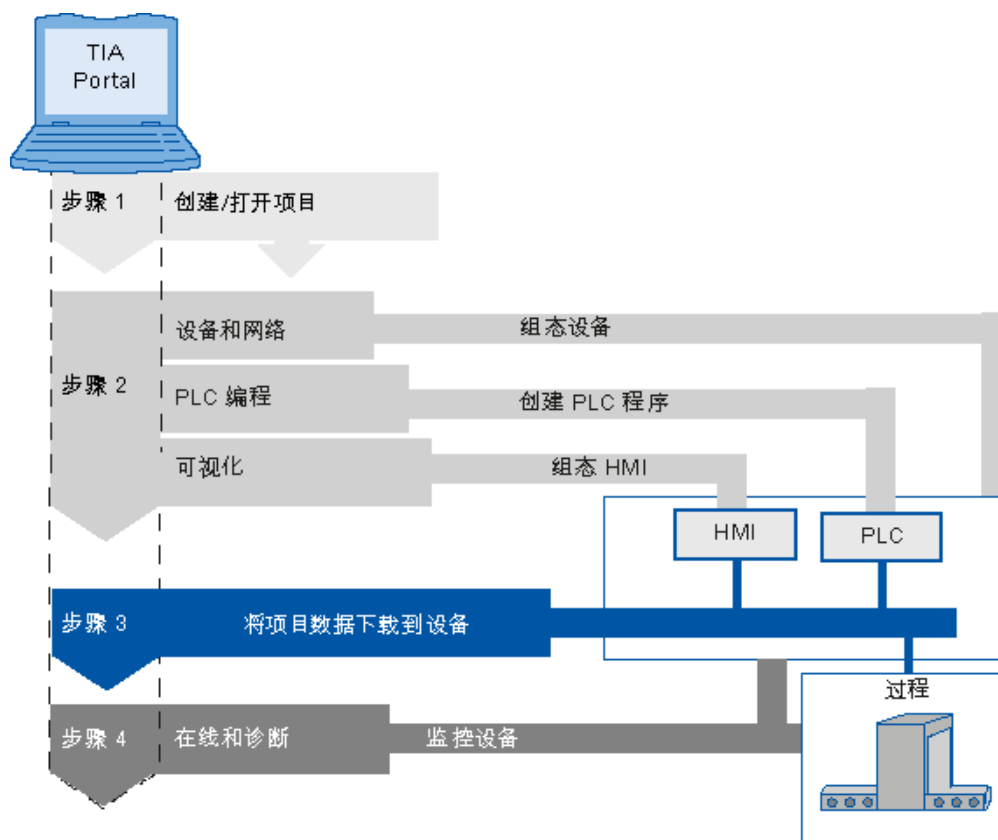
可以使用 TIA 在同一个工程中组态 PLC 和可视化。以上列出的任务可以使用一个公共的用户接口，还能通过最佳方式整合在一起。TIA Portal 有以下优点：

- 公共数据
- 程序、组态数据和可视化数据处理简单
- 可使用拖放操作轻松编辑
- 易于将数据装载到设备
- 易于操作
- 支持图形组态和诊断

3.1.1.2 基本步骤

过程顺序

完成自动化任务的方法有多种。下图显示了在您执行这些工作步骤期间 TIA Portal 是如何提供支持的。

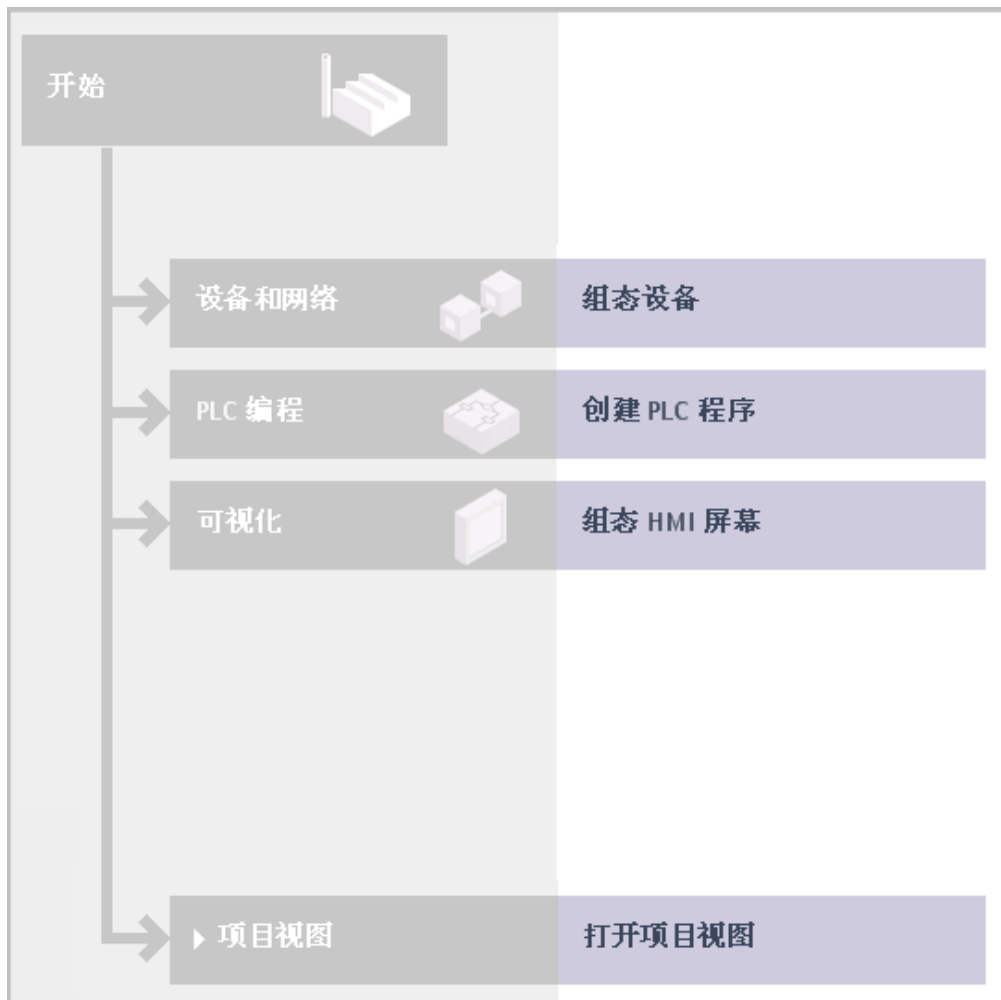


基本步骤

1. 打开/创建项目
首先必须创建新项目或者打开现有项目。接下来的步骤顺序不是固定的。您可以从任意任务继续。
2. 或者：
 - 组态设备
 - 创建 PLC 程序
 - 组态 HMI 画面
3. 将项目数据装载到设备
4. 在线监视设备

Portal 视图中的典型步骤

创建新项目后，如果在 Portal 视图中选择“入门向导”(First Steps)，则将显示 TIA Portal 中的典型步骤。



3.1.1.3 使用各种视图

Totally Integrated Automation Portal 提供了两种不同的界面以帮助用户提高效率：Portal 视图和项目视图。

Portal 视图引导您完成关键任务，而项目视图则提供了包含项目中各元素的面向对象的视图。用户只需通过单击即可在 Portal 视图和项目视图间切换。

Portal 视图中面向功能的显示

Portal 视图用于提供一种简单的方式来浏览项目任务和数据。这意味着通过各个登陆选项可访问处理关键任务所需的应用程序功能。下图是一个组态设备及设备联网的登陆选项实例：



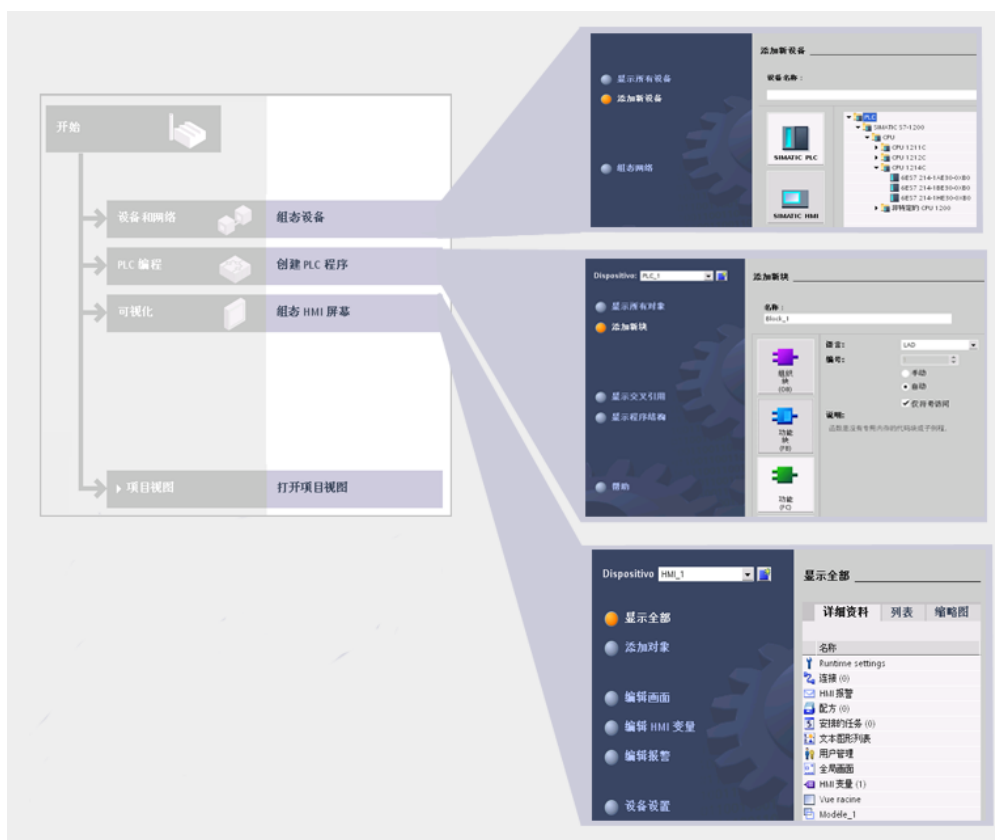
- ① 登陆选项选择
- ② 操作
- ③ 项目数据

登陆选项说明

登陆选项	描述
开始 (Start)	使用“开始”(Start) 登陆选项为您的应用建立项目。可以在此处创建新项目、打开现有项目以及浏览到项目中的关键任务。
设备和网络 (Devices & networks)	在“设备和网络”(Devices & Networks) 登陆选项中定义和组态项目中的设备及其通信关系。例如，组态 PLC 和 HMI 设备并定义设备间的网络连接。还可以创建逻辑链接，通过这种方法，设备可以使用共同的变量。
PLC 编程 (PLC programming)	使用“PLC 编程”(PLC Programming) 登陆选项为项目中的各个设备创建控制程序。
可视化	在“可视化”(Visualization) 登陆选项中为项目中的 HMI 设备创建画面。
在线和诊断	在“在线和诊断”(Online & Diagnostics) 登陆选项中可以显示可访问的设备及其在线状态。

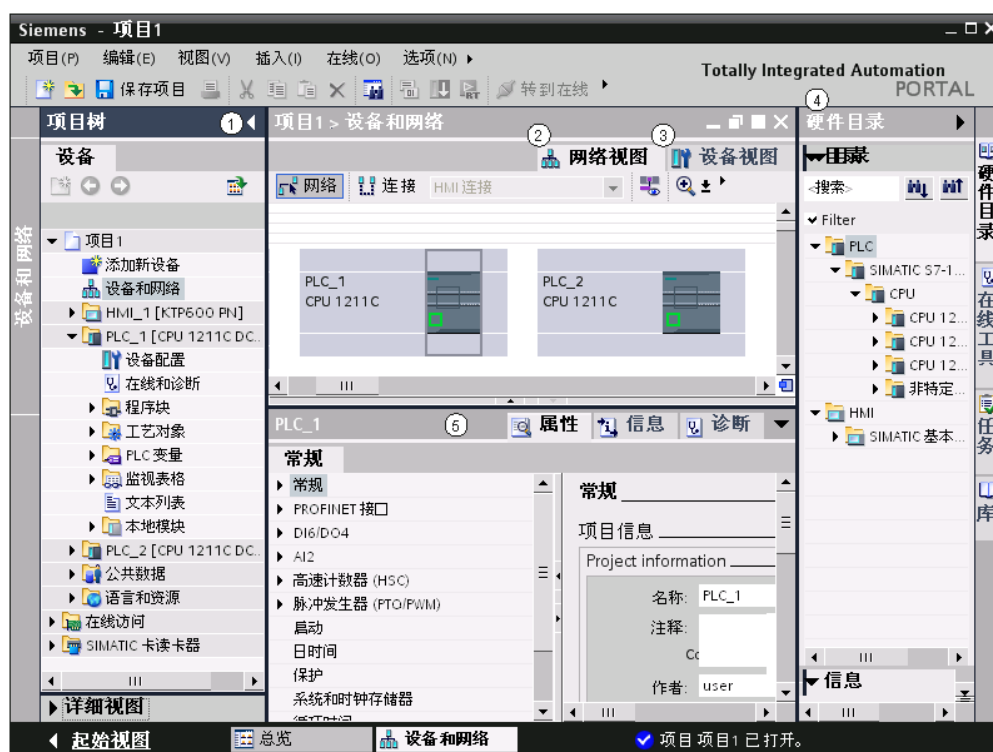
快速访问入门向导

通过开始登陆选项中显示的“入门向导”(First steps) 总览，可以快速选择处理具体任务的登陆选项。单击其中的特定链接可转到所需的登陆选项。



在项目视图访问项目的所有元素

项目视图是基于项目数据的视图，且相应项目的元素以结构化形式显示。通过该视图可以快速访问所有项目数据。



- ① 项目树
- ② 工作区
- ③ 任务卡
- ④ 巡视窗口

项目树

通过项目树可以访问所有设备和项目数据。例如，可在项目树中执行以下任务：

- 添加新设备
- 编辑现有设备
- 请求属性
- 打开编辑器以处理项目数据

工作区

在工作区中打开可用来处理项目数据的编辑器。

任务卡

根据所编辑对象或所选对象，编辑器中提供了用于执行附加操作的任务卡。这些操作包括：

- 从库中或者从硬件目录中选择对象
- 在项目中搜索和替换对象

- 将预定义的对象拖入工作区

巡视窗口

有关所选对象或所执行操作的附加信息均显示在巡视窗口中。

参见

- 项目树 (页 107)
- 工作区 (页 110)
- 巡视窗口 (页 111)
- 任务卡 (页 113)

3.1.1.4 访问帮助

帮助系统的组成元素

全面的帮助系统可用来帮助您完成任务。帮助系统介绍了基本概念、说明和功能。在使用程序时，还提供以下支持：

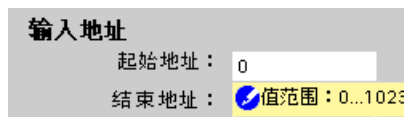
- 提示在对话框中进行正确输入的弹出项
- 包含有关用户界面元素（例如，文本框、按钮和图标）信息的工具提示。有些工具提示还补充有包含更多信息的层叠项。
- 当前上下文的帮助，例如，按 <F1> 或 <Shift+F1> 键时可获得菜单命令的帮助。

弹出项

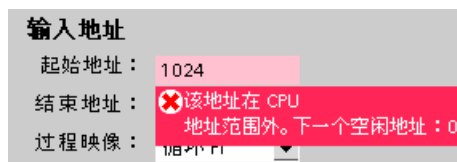
某些文本框提供的弹出信息，可帮助您输入有效的参数和值。弹出项提示您这些文本框中合理的取值范围和数据类型。

下图显示了几个弹出项实例：

弹出项实例：Valid value



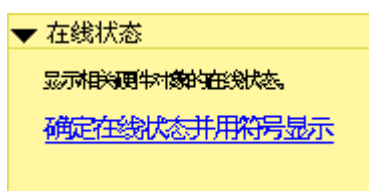
弹出项实例：Error message if value is invalid



工具提示

界面元素提供了工具提示，从而使标识更容易。

左侧有箭头符号的工具提示，其工具提示层叠项中包含了附加信息。如果帮助中包含更多信息，则层叠项中会出现指向相应帮助主题的连接。下图显示了层叠项已打开的工具提示：



帮助

帮助介绍了概念、说明和功能。而且还包含参考信息和实例。

“帮助”在屏幕右侧单独打开一个窗口。





可以在帮助中显示导航区域。在此可访问下列功能：

- 目录
- 搜索索引
- 整个帮助的全文搜索
- 收藏夹

有关使用帮助的更多信息，请参见“[使用帮助系统 \(页 126\)](#)”一章。

帮助中各主题的标识

根据帮助主题所包含的信息类型，用不同的符号来标识各种帮助主题。

符号	信息类型	说明
	操作说明	描述执行特定任务需遵守的步骤。
	实例	包含解释任务的具体实例。
	实际信息	包含执行任务时需要了解的背景信息。
	参考	包含详细的参考信息，例如，程序块说明。

3.1.2 创建项目

简介

首先，创建一个新项目。在创建自动化任务期间出现的数据和程序按有序的方式存储在项目中。本例是在 Portal 视图中打开 Totally Integrated Automation Portal。开始登陆选项包含用于创建新项目 and 用于打开现有项目的命令。

步骤

要创建项目，请按以下步骤操作：

1. 选择“创建新项目”(Create new project) 命令。



2. 输入项目名称 (例如，“Project1”或“First steps”)。
3. 单击“创建”(Create)

结果

创建了项目。

Portal 视图中用于选择各种面向任务的登陆选项的按钮已启用。

提示

创建或打开项目之后，可以使用“入门向导”(First steps) 命令来执行创建自动化应用的任务。

后续步骤

创建项目完成之后，请执行以下任务：

- 通过在线 PLC 识别或手动配置来配置硬件
- 创建用于控制应用的程序

您可以定义这些步骤的顺序。如果首先配置硬件，则可随后将组态的 PLC 分配给程序。如果首先创建程序块，则这些块将被分配给一个自动创建的、未指定的 PLC。

3.1.3 配置硬件和网络

3.1.3.1 硬件配置简介

要建立一个自动化系统，需要对各硬件组件进行组态、参数化和互连。PLC 和 HMI 设备都能在相同的环境以相同的方式插入项目中。

Portal 视图和项目视图中的硬件配置

可以在 Portal 视图或项目视图中添加新设备。

在“设备和网络”(Devices & Networks) 登陆选项中添加 PLC 设备 (CPU) 或 HMI 设备。

如果想插入其它模块或将设备联网，则需要使用项目视图。

3.1.3.2 在 Portal 视图中添加设备

在 Portal 视图中添加 SIMATIC PLC

要添加 SIMATIC PLC，请按以下步骤操作：

1. 在 Portal 视图中选择“设备和网络”(Devices & Networks)。
2. 选择“添加新设备”(Add new device) 命令
3. 单击“SIMATIC PLC”按钮。将打开 SIMATIC PLC 的列表。
4. 选择一个 SIMATIC S7-1200 并单击“添加”(Add)。



在 Portal 视图添加 HMI 设备

要添加 HMI 设备，请按以下步骤操作：

1. 选择“添加新设备”(Add new device) 命令
2. 单击“SIMATIC HMI”按钮。将打开 SIMATIC HMI 面板的列表。
3. 选择一个 HMI 面板并单击“添加”(Add)。

下一步做什么？

组态 PLC 和 HMI 设备。

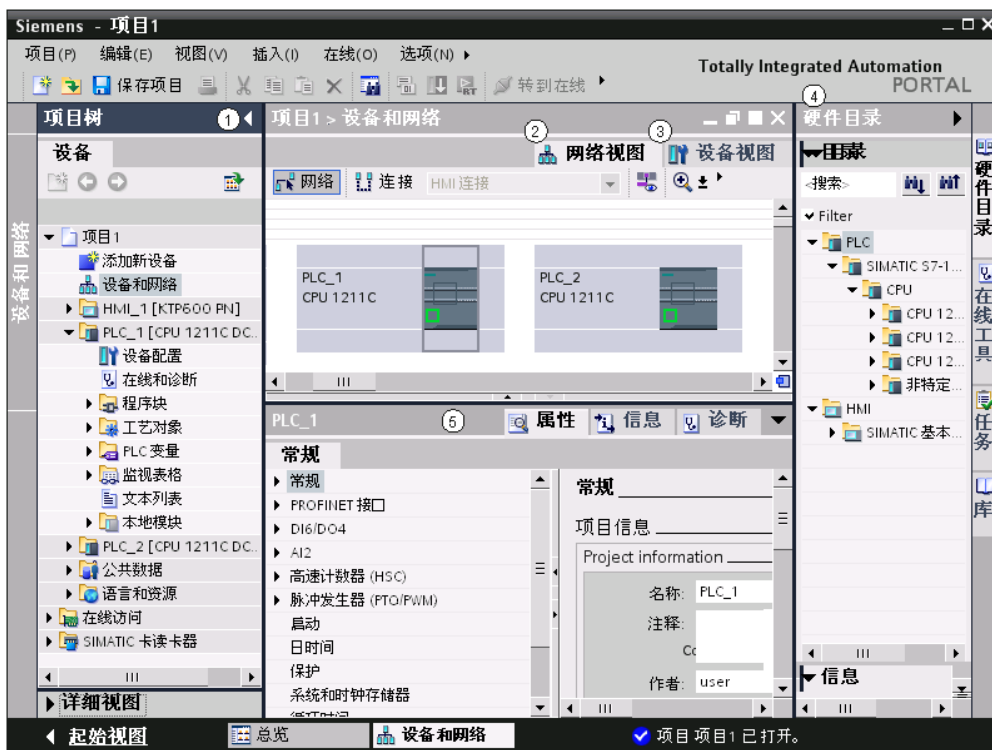
双击项目中显示的设备之一，即可浏览到项目视图的硬件和网络编辑器。

3.1.3.3 项目视图中的设备和网络

项目视图中的硬件配置

在项目视图中提供了硬件和网络编辑器。该编辑器是一个集成开发环境，供用户执行组态、参数化以及联网设备和模块，为用户执行自动化任务提供了最有力的支持。

在项目树中双击“设备和网络”(Devices and Networks) 可打开硬件和网络编辑器。



- ① 项目树
- ② 网络视图
- ③ 设备视图
- ④ 硬件目录
- ⑤ 巡视窗口

网络视图和设备视图

硬件和网络编辑器包含设备视图和网络视图。可以根据是要生成和编辑单个设备还是整个网络，随时在这两个视图间切换。

硬件目录

将自动化系统所需的设备和模块从硬件目录拖到网络视图或设备视图中。

巡视窗口

巡视窗口的“属性”(Properties) 下面包含当前所标记对象的相关信息。在此处可更改所标记对象的设置。

3.1.3.4 在项目视图中添加设备

在项目视图中插入设备

在项目视图中有多种方法可将 CPU 或 HMI 设备添加到硬件配置中，例如：

- 项目树中的命令“添加新设备”(Add new device)
- 在硬件产品目录中双击设备
- 从硬件产品目录中拖放到网络视图中
- 通过网络视图菜单栏中的命令“添加 > 设备”(Add > Device)

新建 CPU 时将同时创建一个配套的机架。所选设备插在机架上第一个可用的插槽中。无论选择哪种方法，添加的设备在项目树以及硬件和网络编辑器的网络视图中都可见。

下一步做什么？

通过将其它模块插入机架来配置 PLC。

定义 HMI 面板的设备设置。

3.1.3.5 将其它模块插入机架

简介

将 CPU 添加到组态中之后，再为设备配置其它模块。可使用多种方法来添加这些模块，例如：

- 在硬件目录中双击模块
- 将模块从硬件产品目录拖到空闲插槽中

设备视图包含机架中所选设备的图示。软件中机架的图示与实际结构一样，这意味着可以看到的插槽数与实际结构中存在的相同。

提示

要从网络视图进入设备视图，可在网络视图中双击某个设备或站。

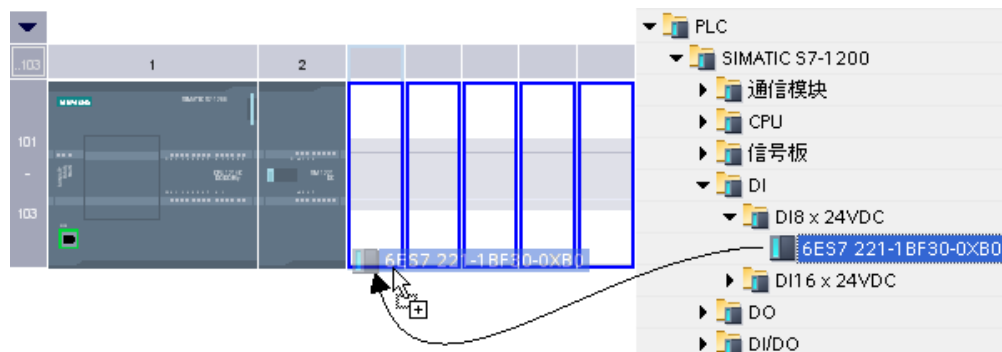
要求

- 您处于设备视图中。
- 硬件目录已打开。

从硬件目录中添加模块

下面以信号模块为例说明如何将模块从硬件目录插入到机架。为此，请按下列步骤操作：

1. 转到硬件目录中所需的模块板卡。
2. 选择所需的模块。
3. 将该信号模块拖到机架上空闲的插槽上。



这样就将数字量信号模块插入到了机架的插槽中。对其它模块重复执行这些步骤。

3.1.3.6 编辑属性和参数

在机架上分配硬件组件后，即可编辑其默认属性，例如，在网络或设备视图中编辑参数或地址。

要求

您处于设备视图中。

注意

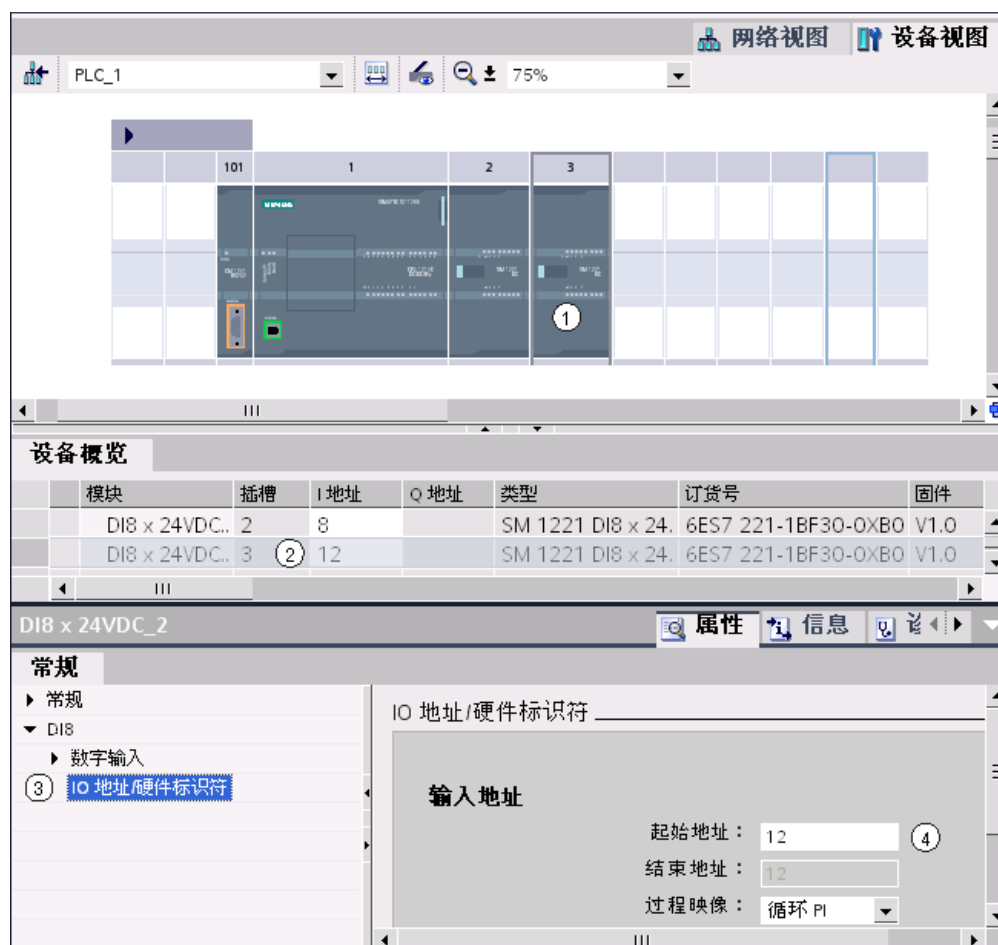
也可在网络视图中编辑属性和参数。在网络视图中，只能访问网络相关的硬件组件和站。

步骤

要更改硬件组件的属性和参数，请按以下步骤操作：

1. 在图形视图中，选择要编辑的模块、机架或接口。
2. 编辑所选对象的设置：
 - 例如，使用表格视图编辑地址和名称。
 - 在巡视窗口中，可以在“属性”(Properties) 中进行其它可能的设置。

更改设置实例



- ① 接口的选择
- ② 设备概览中的地址编辑选项
- ③ 巡视窗口中的选择选项
- ④ 巡视窗口中的地址编辑选项

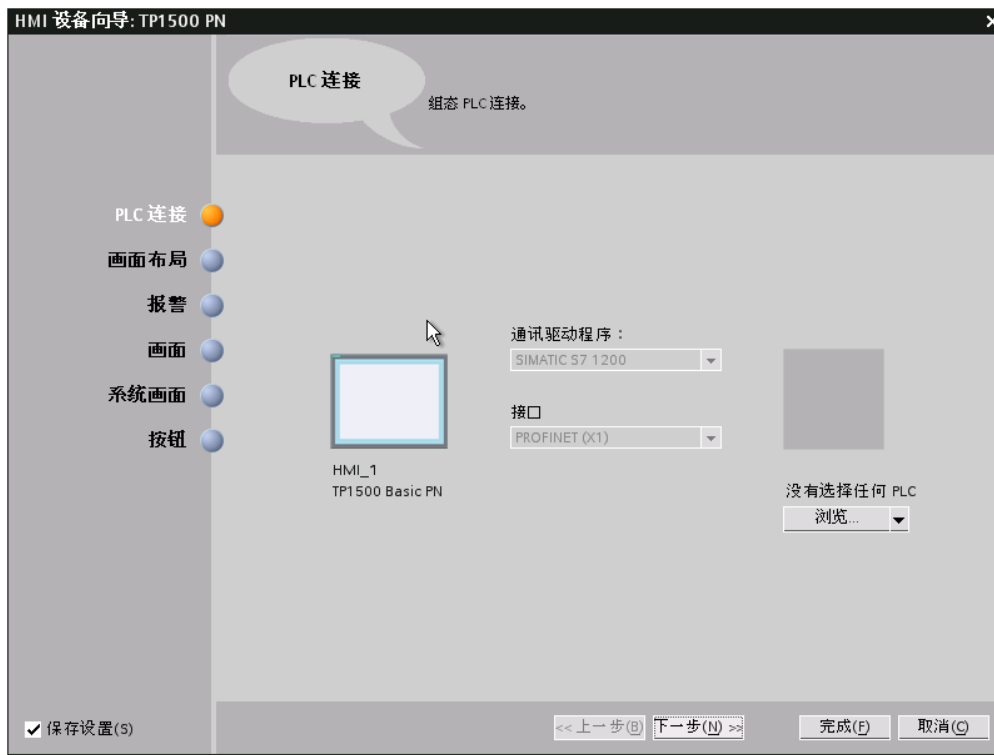
3.1.3.7 修改 HMI 设备的设置

HMI 设备的设置

插入 HMI 设备后，将打开 HMI 设备向导。您可在其中立即定义以下属性，例如：

- 图形
- 报警
- 用户管理
- 语言和字体

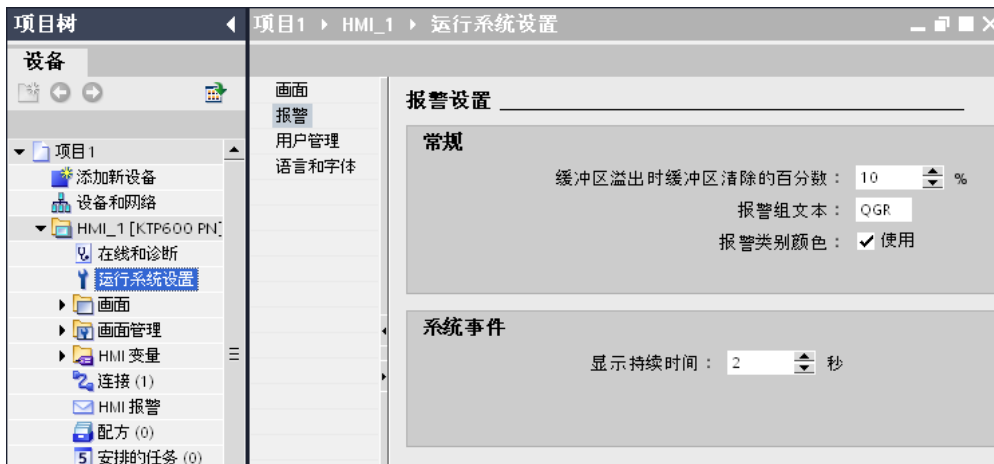
下图显示了 HMI 设备向导的实例：



更改运行系统的设备设置

要在项目视图中查询和更改设备设置，请按以下步骤操作：

1. 打开项目树并转到 HMI 设备。
2. 单击“设备设置”(Device settings)。



1. 在巡视窗口或相应的编辑器中更改其它属性，例如画面的和 HMI 变量的属性（请参见组态画面和画面对象(页 89)一章）。

3.1.3.8 联网设备

选项

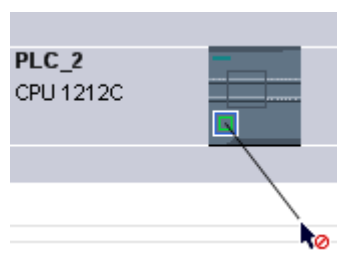
在图形网络视图的工作区中，可以方便地将具备通信功能的组件的接口联网。网络视图表格提供了一个包含所有相关信息的有序视图。

将两个目标设备连接到新子网

要通过尚不存在的子网连接接口与其它设备，请按以下步骤操作：

1. 将鼠标指针置于有通信功能且需要联网的组件的接口上。
2. 单击鼠标左键并按住该按钮。
3. 移动鼠标指针。

指针此时将使用联网符号来指示“联网”模式。同时，鼠标指针显示锁定符号，该符号只有在指针移动到有效目标上时才会消失。



4. 现在，将指针移动到目标设备的接口上。可以一直按住鼠标按钮，也可释放鼠标按钮。
5. 此时根据前面的操作释放鼠标左键或再次按下。

创建了新的子网。相关接口现在通过新子网相连接。同时会自动为接口设置一致的地址参数。

下图显示了这两个联网的设备：



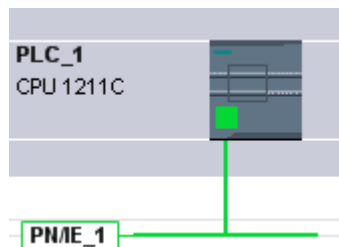
创建子网

要创建子网并将其连接到一个接口，请按以下步骤操作：

1. 选择 CPU 的接口。
2. 在接口的快捷菜单中选择“创建子网”(Create subnet) 命令。

所选的接口将连接到具有相应子网类型的新子网。同时会自动为接口设置一致的地址参数。

下图显示了引出线路连接到子网的接口：



3.1.4 对 PLC 编程

3.1.4.1 TIA Portal 的编程环境。

本部分概要介绍了 TIA Portal 的编程环境：

- 编程语言
- 程序编辑器
- 符号编程
- 库

编程语言

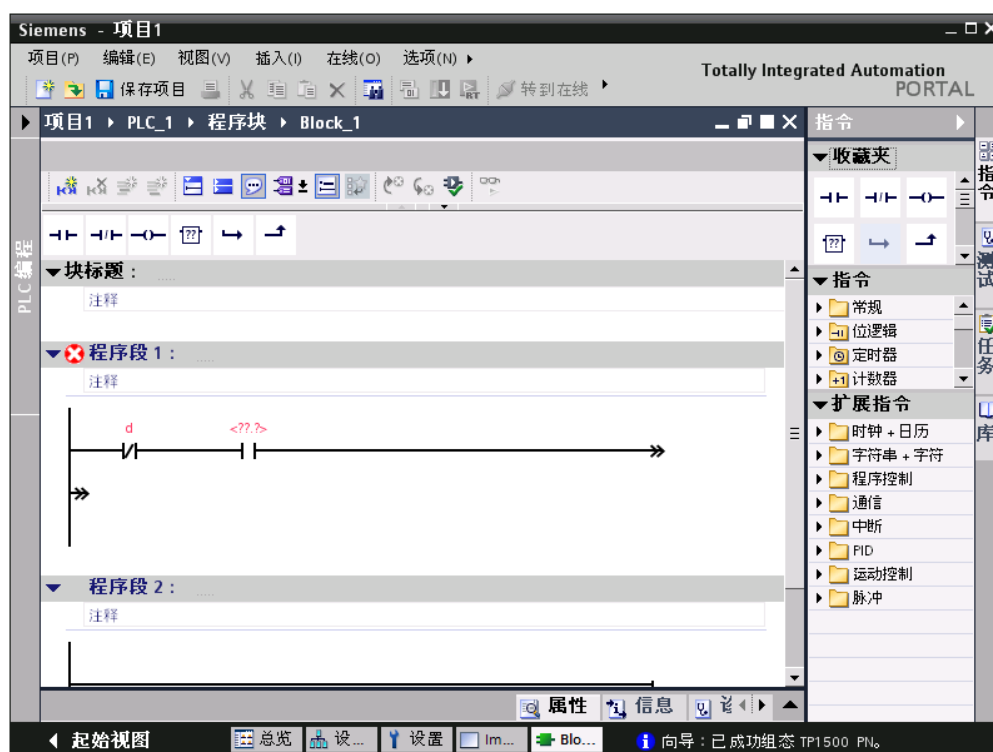
TIA Portal 支持用以下编程语言进行代码块编程：

- 梯形图 (LAD)
- 功能块图 (FBD)

程序编辑器

程序编辑器是用于设计功能、功能块和组织块的集成开发环境。它为编程和故障排除提供了全面的支持。

下图以 LAD 为例显示了程序编辑器的各个组成部分：



符号编程

为使程序布局更加清晰和简化故障排除，建议在编程时使用操作数和变量的符号名。

指定符号名：

- 在 PLC 变量表中为全局变量定义符号名。
- 在块接口中，为逻辑块或全局数据块中的局部变量指定符号名。

编程期间的支持

编程时可先使用符号操作数，以后再分配绝对地址。即使分配没有完成，也仍可保存程序。

尚未分配绝对地址的操作数会以红色波浪下划线突出显示。

库

您可以将想要多次使用的对象存储在库中。可以在项目范围内或跨项目重复使用已存储的对象。这意味着您可以创建块模板以便在不同项目中使用并根据自动化任务的特定要求修改这些模板。

库可以容纳大量对象。例如，这包括：

- 功能 (FC)
- 功能块 (FB)
- 数据块 (DB)
- 设备
- PLC 数据类型

- 监视表格
- 过程画面
- 面板

3.1.4.2 添加块

要创建自动化解决方案的程序，必须在项目中添加块并进行编程。可以在 Portal 视图或项目视图中添加新设备。项目视图中的程序编辑器和任务卡可提供编程方面的支持。

在 Portal 视图中添加块

要在 Portal 视图中添加块，请按以下步骤操作：

1. 选择“PLC 编程 > 添加新块”(PLC programming > Add new block)。



2. 单击其中一个按钮创建所需的块类型。
3. 选择代码块的编程语言。
4. 必要时，可更改块属性。
5. 单击“创建”(Create)

结果

在程序编辑器中创建并打开块。

在 Portal 视图添加块

要在 Portal 视图添加块，请按以下步骤操作：

1. 转到项目树中的所需设备。
2. 在“程序块”(Program blocks) 下双击“添加新块”(Add new block)
将打开与 Portal 视图中相同的选择对话框。
3. 如在 Portal 视图中一样执行步骤 2 到 5。

其它步骤

对代码块编程。

3.1.4.3 编写代码块

本部分说明如何在程序编辑器中使用 LAD 编程语言编写功能块。

要求

在程序编辑器中创建并打开功能块。

步骤

要编写功能块，请按以下步骤操作，顺序不限：

- 编程指令
- 声明块接口
- 声明 PLC 变量并在程序中使用这些变量

以下基于实例对这些操作进行了说明。

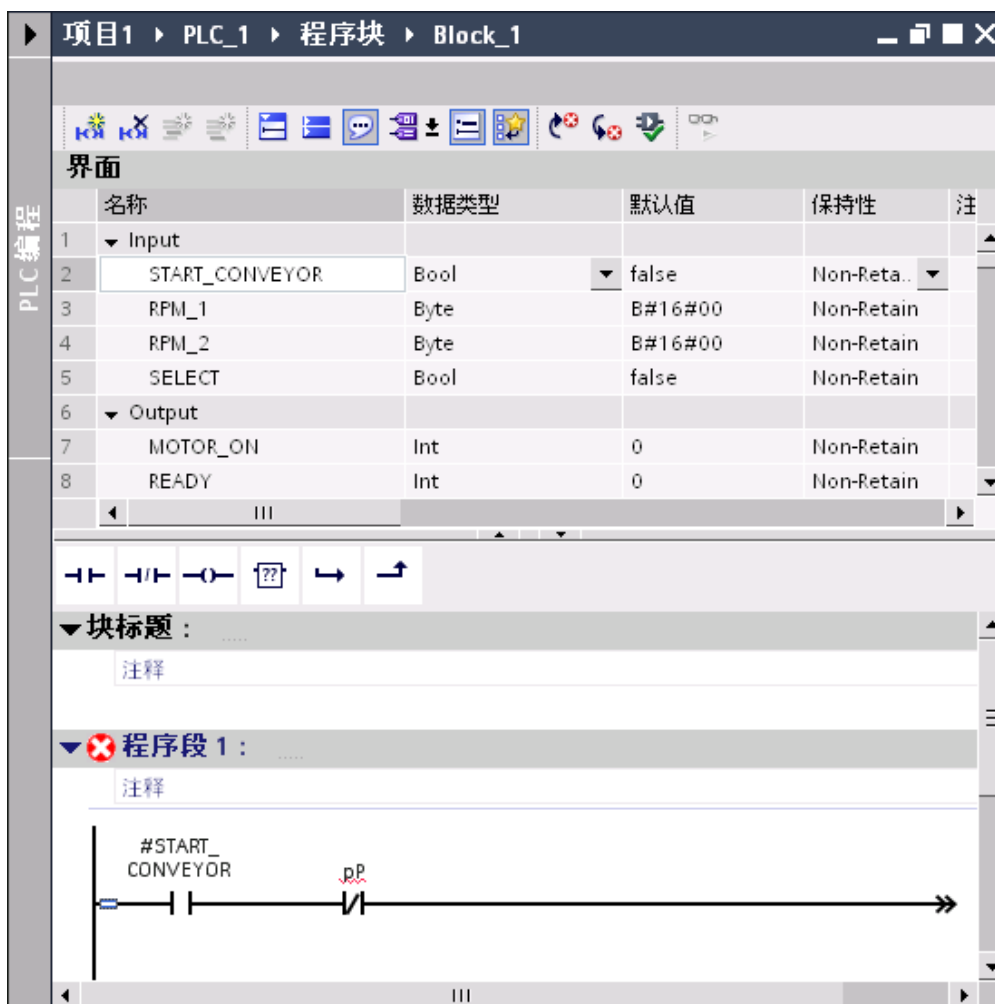
编程指令

可使用以下方法在块中插入 LAD 元素：

- 从“指令”(Instructions) 任务卡中通过拖放操作
- 通过空功能框，然后向其分配 LAD 指令。可以在“指令”(Instructions) 任务卡的“常规”(General) 下找到空功能框。要分配指令，请单击右上角的黄色三角形并从下拉列表框中选择指令。

声明块接口

块接口以表格形式显示在程序编辑器的上方区域。



要在块接口中声明变量，请按以下步骤操作：

1. 选择相应部分（输入、输出...）。
2. 在“名称”(Name) 列中输入变量名称。
3. 在“数据类型”(Data type) 列中，从列表里选择数据类型。
4. 可根据需要输入默认值。

结果

创建该变量。每次输入后都会执行语法检查，并且找到的任何错误都将以红色显示。

3.1.4.4 声明 PLC 变量

在程序中使用 PLC 变量和局部变量

可以为程序定义适用范围不同的变量：

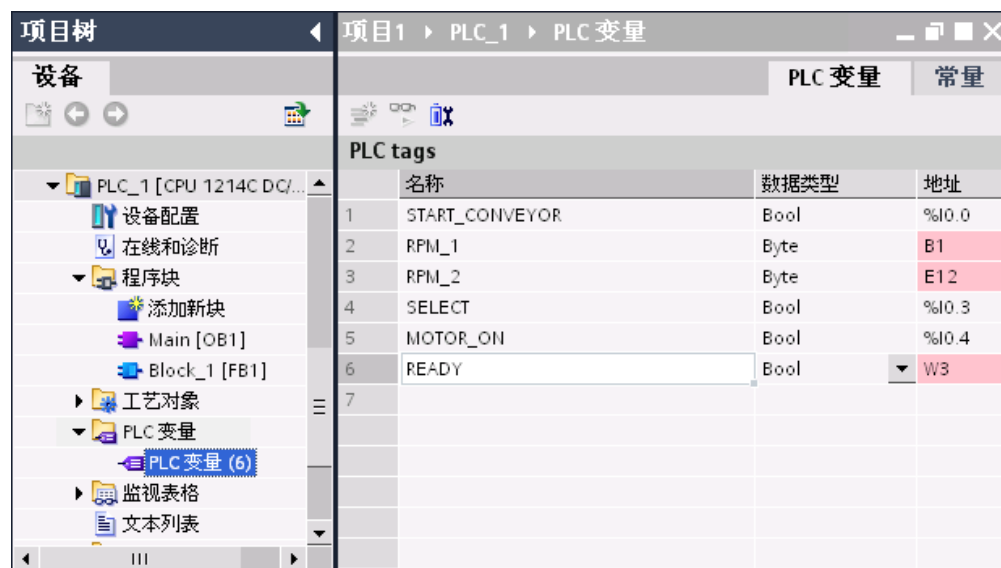
- PLC 变量对于整个 CPU 有效并且可以用于 CPU 的所有块。在 PLC 变量表中定义 PLC 变量。

- 局部变量仅适用于定义这些变量的块。

打开 PLC 变量表

要在 CPU 中打开 PLC 变量表，请按以下步骤操作：

1. 在项目树中打开 CPU 下面的“PLC 变量”(PLC tags) 文件夹。
2. 双击该文件夹中的 PLC 变量表。
3. 在右上角选择“PLC 变量”(PLC tags) 选项卡或“常量”(Constants) 选项卡。



PLC 变量表的结构

下图显示了一个 PLC 变量表结构的实例。



可以单击第一列中的符号，将变量拖到程序段中并将其用作操作数。

声明 PLC 变量

未被声明为 PLC 变量的操作数在程序编辑器中以红色显示。可通过以下方法声明 PLC 变量：

- 在程序中直接声明

选择操作数并在快捷菜单中单击“定义变量”(Define tag)。将打开“定义变量”(Define tag) 对话框。此对话框显示一个声明表，其中已输入相应操作数的名称。

在“部分”(Section) 列选择所需的类型，并在其它列中输入地址、数据类型和注释。



- 在 PLC 变量表中

打开想要使用的 PLC 变量表。为每个 PLC 变量输入名称并选择数据类型。

3.1.5 组态工艺功能

3.1.5.1 工艺功能简介

集成的设备工艺功能（例如，工艺对象“轴”或 PID 控制器）可使用自带的编辑器来进行组态。

工艺对象

可以直接在项目树中创建和组态工艺对象。

以下部分将介绍如何编辑运动控制功能。

3.1.5.2 处理运动任务的基本方法

要求

已经创建包含 CPU 12xx 的项目。

步骤

要使用工艺对象“轴”，请按以下步骤操作：

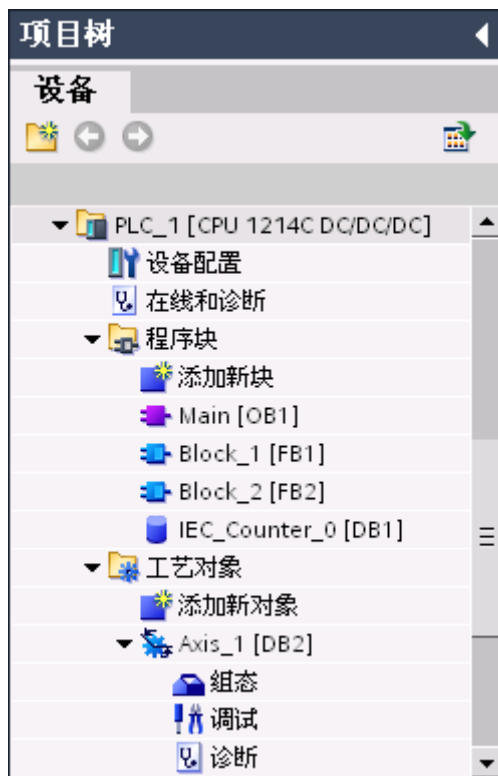
- 添加工艺对象“轴”

在项目树中单击 <PLC>“工艺对象 > 添加新对象”(Technological objects > Add new object)。



2. 选择对象类型和所需属性，然后单击“确定”(OK)。

将创建新的工艺对象并将其保存在项目树的“工艺对象”(Technology objects) 文件夹中。



3. 组态轴参数

在项目树中打开所需工艺对象的组。

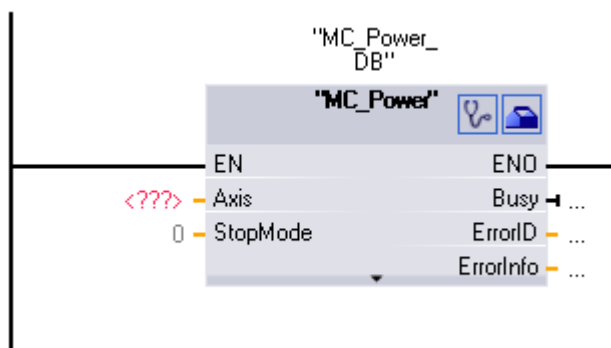
双击对象“组态”(Configuration)。



4. 创建用户程序

可用的块将在任务卡“扩展指令，运动控制”(Extended instructions, Motion Control) 中列出，可将这些块插入用户程序的块 OB1 中。

实例：运动控制指令“MC_Power”



5. 在 CPU 中装载项目

6. 使用轴控制面板进行轴的功能测试

7. 使用“诊断”(Diagnostics) 测试轴控制

3.1.6 组态可视化

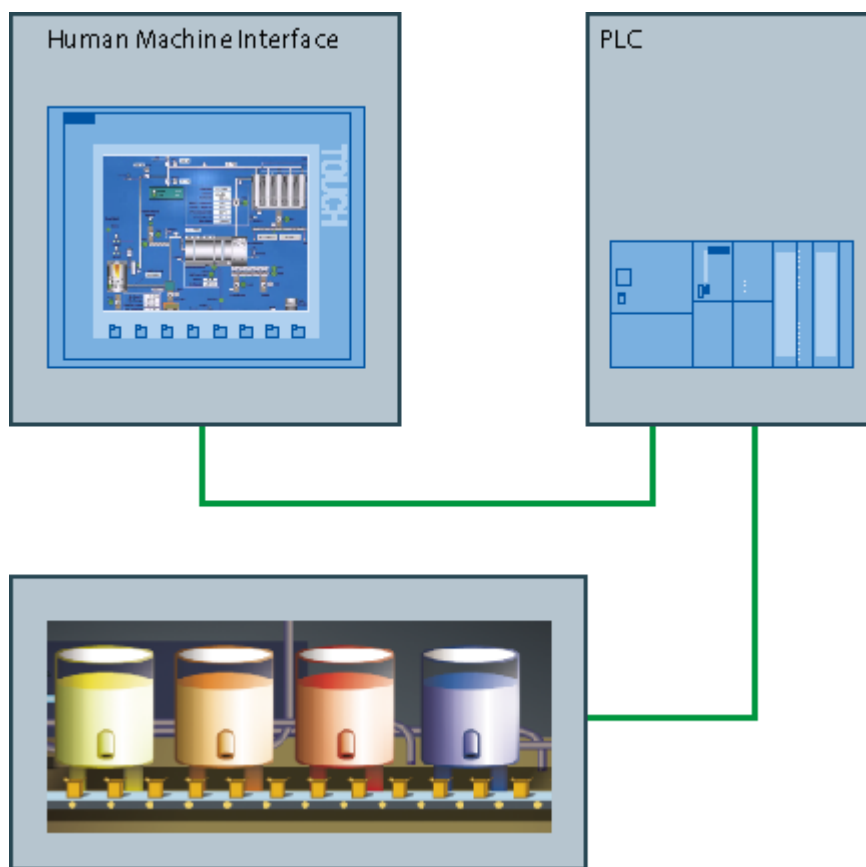
3.1.6.1 TIA Portal 中的可视化

本节概述了 TIA Portal 的可视化选项。

HMI - 人机界面

HMI 系统代表操作员和过程之间的接口。PLC 对过程进行实际控制。因此在操作员和 HMI 面板以及 HMI 面板和 PLC 之间均具有接口。

人机界面实现了操作和监视过程的透明。



TIA Portal 提供以下选项用于操作和监视机器与工厂：

- 显示过程
- 操作过程
- 输出报警
- 归档过程值和报警
- 记录过程值和报警
- 管理过程参数和机器参数

编辑器

在可视化的核心区域内，每个组态任务均有专用的编辑器。

- 图形
- HMI 变量
- HMI 报警
- 配方
- 调度程序
- 文本和图形列表
- 用户管理

3.1.6.2 组态画面和画面对象

可以使用 TIA Portal 创建用于操作和监视机器与工厂的画面。预定义对象可用于协助您创建这些画面；可以使用这些对象模拟工厂、显示过程和定义过程值。

HMI 画面

画面可以包含静态和动态元素。

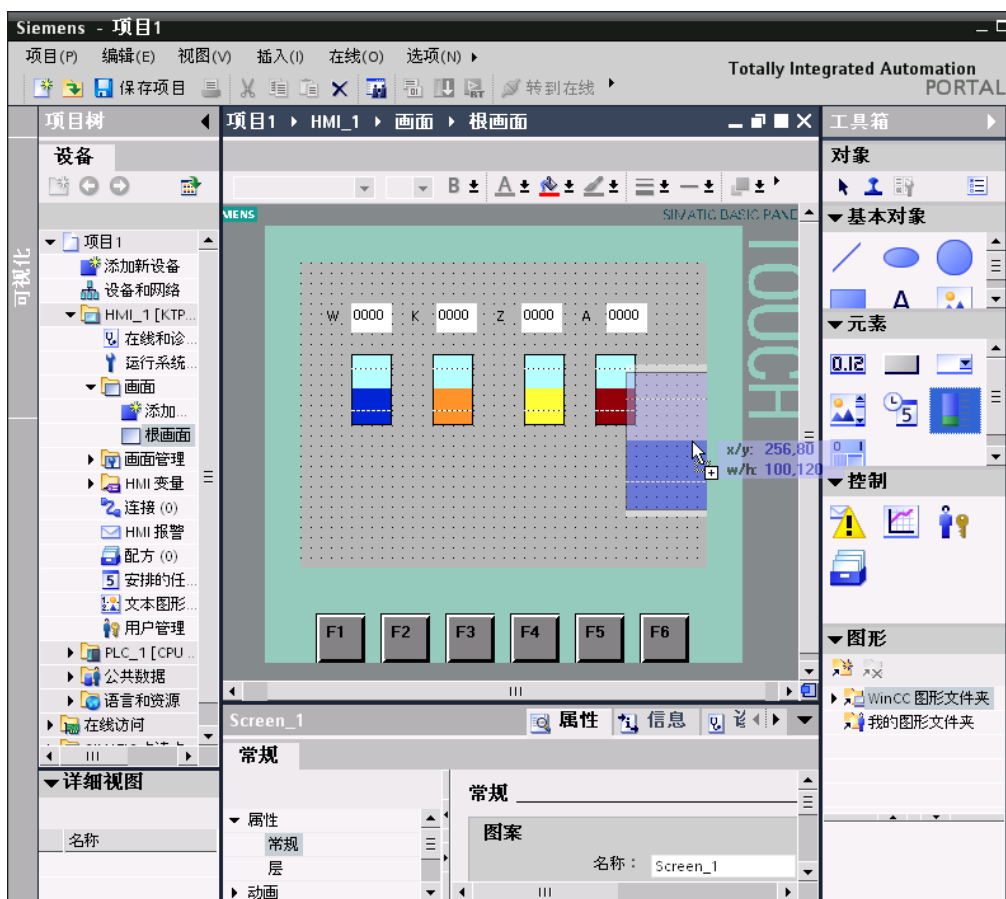
- 静态元素（例如文本或图形对象）在运行时不改变它们的状态。这个搅拌装置实例中的罐标签就是这样的静态元素。
- 动态元素根据过程改变它们的状态。用户可可视化下列当前过程值：
 - PLC 的存储器中的过程值
 - 以字母数字、趋势图和棒图的形式显示 HMI 设备存储器中的过程值
 - HMI 设备上的输入域也认作是动态对象。混合设备实例中的罐填充量值是动态对象。

通过变量可以在控制器和 HMI 设备之间切换过程值和操作员输入值。

在 TIA Portal 中组态画面对象

HMI 设备的功能决定着 WinCC 中的项目可视化和编辑器的功能范围。

只需将画面对象从“工具”(Tools) 任务卡拖到画面中。在巡视窗口的“属性”(Properties) 中对选定的画面对象进行具体设置。



3.1.6.3 组态 HMI 变量

使用“HMI 变量”编辑器组态内部变量和外部变量。

外部变量

外部变量使得自动化过程的组件之间（例如 HMI 设备和 PLC 之间）能够进行通信（数据交换）。

外部变量是 PLC 中所定义的存储位置的映像。从 HMI 设备和 PLC 都可以读写该存储单元。

内部变量

内部变量与 PLC 之间不存在连接。内部变量存贮在 HMI 设备的内存中。因此，只有这台 HMI 设备能够对内部变量进行读写访问。例如，可以创建内部变量来执行本地计算。

“HMI 变量”编辑器

用于可视化项目的所有 HMI 变量都在“HMI 变量”编辑器中创建。根据类别，用户在属性视图中更改变量的属性。也可以选择表格视图中直接更改属性。您可根据需要调整表格视图。



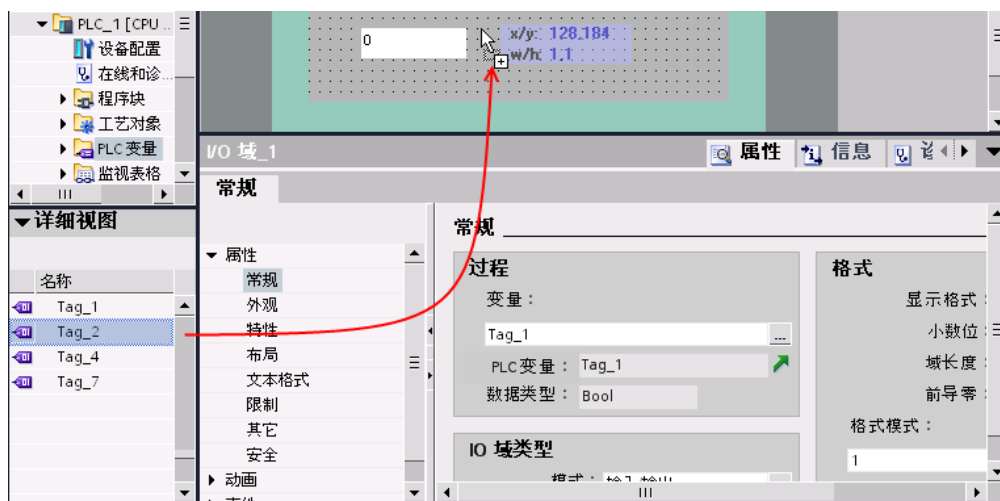
3.1.6.4 TIA：在 HMI 画面中组态 PLC 变量

组态 PLC 变量

如果要在 HMI 画面中组态已经组态过的 PLC 变量，请按以下步骤操作：

1. 打开 HMI 画面。
2. 在项目树中，选择“PLC 变量”编辑器。
所有的外部变量便会显示在“详细视图”下
3. 通过拖放操作将外部变量拖动到 HMI 画面中。

会在 HMI 画面中创建带有链接了 PLC 变量的 IO 域。



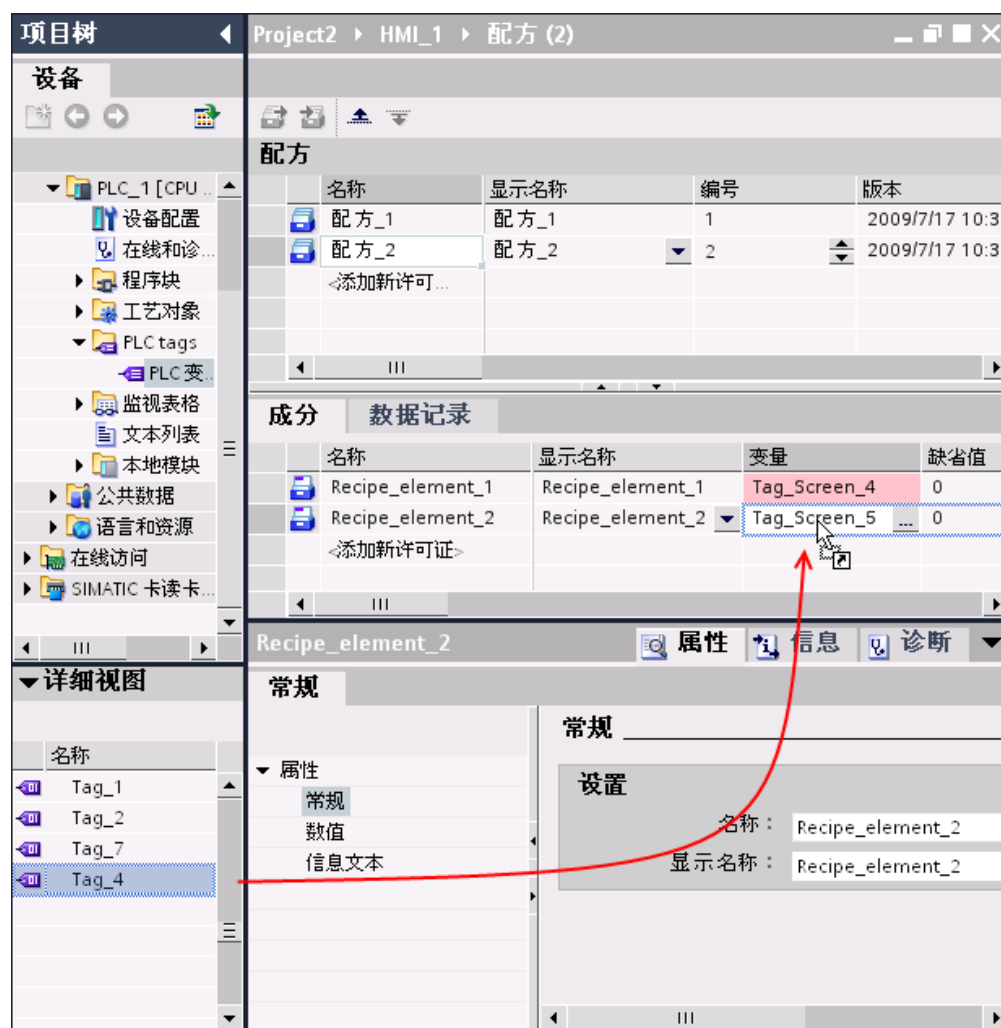
3.1.6.5 组态配方

配方是同一类生产数据的集合，如混合比。可在一个工作步骤中将混合比从面板传送到 PLC，以进行诸如更改生产类型等操作。

配方包含多个元素：为每个元素分配一个变量。配方的数据记录是各个配方元素的不同混合比。

将外部变量分配给配方元素。

1. 打开“配方”编辑器。
2. 在项目树中，选择“PLC 变量”编辑器。
 所有的外部变量便会显示在“详细视图”下
3. 通过拖放操作将外部变量拖动到区域“...”
 现在，配方元素便链接到外部变量。



3.1.7 将项目数据装载到设备

装载项目数据的范围

设备上装载的项目数据分为硬件项目数据和软件项目数据：

- 硬件项目数据是通过组态硬件、网络和连接而生成的。
- 软件项目数据包括用户程序的块。

根据不同设备，将有以下选项：

- 硬件配置
- 软件
- 软件（所有块）
- 全部 - 硬件和软件项目数据的公共装载过程。

装载选项

根据安装的适用范围，可装载单个对象、文件夹或完整设备。可通过以下方法进行装载：

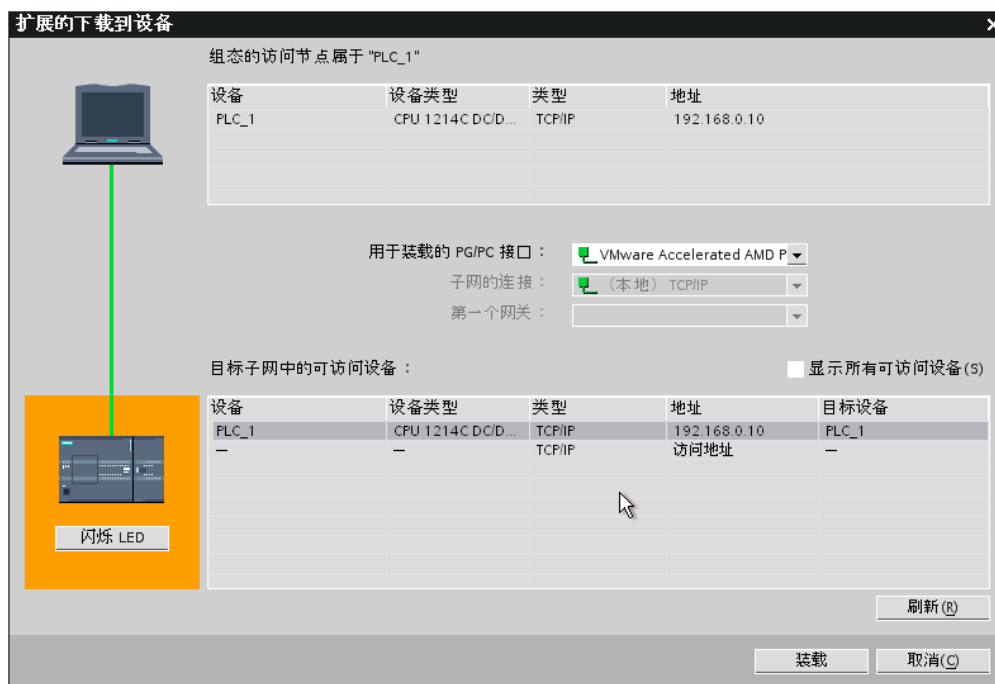
- 在项目树中装载项目数据
- 可以拖放项目数据以将其装载到可访问设备。
- 可以拖放项目数据以将其装载到存储卡。

要求

项目数据一致。

已经为所选设备或对象完成编译项目数据。如果没有，则会在装载项目数据之前将自动进行编译。

实例： 可以拖放项目数据以将其装载到可访问设备。



结果

项目数据装载到所选的设备。用户随即可在设备上执行测试和诊断功能。

3.1.8 使用在线功能

3.1.8.1 在线功能概述

在线模式

在线模式下，编程设备/PC 和一个或多个设备之间会建立在线连接。将程序和组态数据加载到设备中以及执行下列操作时，编程设备/PC 与设备之间必须存在在线连接：

- 测试用户程序
- 显示和切换 CPU 的工作模式
- 显示和设置 CPU 的日期和日时钟
- 显示模块信息
- 比较在线块和离线块
- 诊断硬件

显示可访问设备

可访问设备是指所有连接到编程设备/PC 接口上并且已经开启的设备。

要显示可访问设备，请按以下步骤操作：

1. 在 Portal 视图中选择“在线和诊断”(Online & Diagnostics)。
2. 单击“可访问设备”(Accessible devices)。



“可访问设备”(Accessible devices) 对话框将在表格中列出所有的设备及其地址。可以选择用于在线连接的 PG/PC 接口并使所选设备上的 LED 闪烁。

转到在线

在 Portal 视图和项目视图中都可以建立与设备间的在线连接。下图显示了如何从“在线和诊断”(Online & Diagnostics) Portal 视图启动在线模式。为此，请单击“在线状态”(Online Status)。



显示诊断信息

然后即可使用“在线和诊断”视图或“在线工具”(Online tools) 任务卡访问设备上的数据。设备的当前在线状态在各种视图以符号形式显示。有关各种状态符号的含义，请参见相关的工具提示。

参见

用于硬件和网络的诊断功能 (页 97)

3.1.8.2 测试用户程序

您可以测试用户程序在设备中的运行情况。然后，可以监视信号状态和变量值，并且可以在程序运行期间给变量赋值以便仿真特定情况。

要求

- 已建立与设备的在线连接。
- 包含可执行程序的项目数据已装载到设备中。

测试选项

可使用下列测试选项：

- 使用程序状态测试

通过程序状态可以监视程序的运行情况。可以显示操作数的值和程序段逻辑运算的结果，从而可以识别和修复程序中的错误。

- 使用监视表格测试

通过监视表格可以监视、修改或强制用户程序或 CPU 中的各变量的当前值。可以给各变量赋值来进行测试和在各种不同的情况下运行程序。也可以在 STOP 模式下给 CPU 的输出 I/O 分配固定值，例如用于检查接线情况。

实例：程序状态的显示

程序状态的显示是周期性更新的。它从所选程序段开始。下图所示为 LAD 程序状态显示实例：

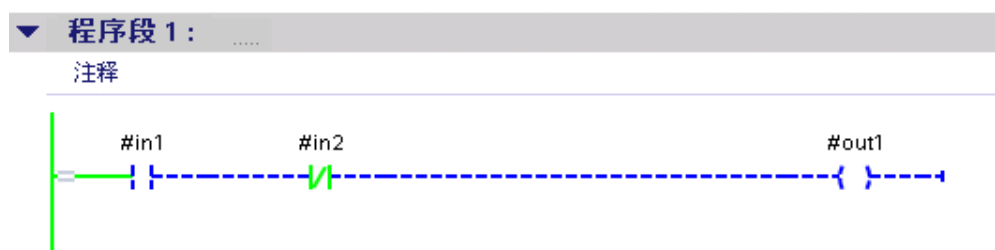


图 3-1 根据线和符号的颜色和类型，可以快速识别程序段中各运算和线的状态。

实例：使用监视表格测试

监视表格包含用户为整个 CPU 定义的变量。系统会为项目中创建的每个 CPU 自动生成一个“监视表格”(Watch tables) 文件夹。通过选择“添加新监视表格”(Add new watch table) 命令可以在该文件夹中创建新的监视表格。

下图显示了基本模式下的监视表格布局，其中包括“控制”(Control) 和“强制”(Force) 列：

	名称	地址	显示格式	监视值	值		F	注释
1	"BM_Count"	%MW4	Hex	0001	0001	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	"DB_Count"	%MW2	Hex	0000		<input type="checkbox"/>	<input type="checkbox"/>	
3	"DM_reach"	%I1.6	Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>	<input type="checkbox"/>	
4	"DM_turn_off"	%I1.5	Bool	<input type="checkbox"/> FALSE		<input type="checkbox"/>	<input type="checkbox"/>	
5	"DM_Rotation"	%I1.4	Bool	<input type="checkbox"/> FALSE	TRUE	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6						<input type="checkbox"/>	<input type="checkbox"/>	

可以使用“立即监视”(Monitor immediately) 符号立即启动监视，然后对变量进行控制和强制。

3.1.8.3 用于硬件和网络的诊断功能

用于硬件和网络的诊断功能

下列视图和工作区可用于硬件和网络的诊断：

- “在线和诊断”视图提供了关于设备状态的综合信息（同时也会以图形方式提供信息）并能向设备传送命令，例如设置日时钟。
- 对于本身具有工作模式的模块，通过“在线工具”(Online Tools) 任务卡可以查询模块的当前诊断信息以及向模块传送命令，例如更改工作模式命令。
- 在巡视窗口的“诊断 > 设备信息”(Diagnostics > Device Info) 区域中，用户可以从总体上了解在线连接的故障设备。

状态和诊断信息的符号显示

与设备建立在线连接时，还会同时确定和显示相关模块的状态和工作模式（如有必要）。

在下列视图中进行显示：

- 在设备视图中，将在大多数硬件组件中显示诊断符号。对于本身具有工作模式的模块，同时还会显示工作状态符号。
- 在网络视图中显示各个设备的诊断符号，以此显示所有相关硬件组件的总体状态。
- 每个硬件组件的诊断图标都会在网络总览和设备总览中显示。
- 在项目树中，诊断符号显示在集中式 IO 设备的硬件组件的后面。对于本身具有工作模式的模块，还会在模块图标的右上角显示工作状态图标。
- 在“在线和诊断”视图中，在线状态显示在“在线访问”(Online access) 组的“状态”(Status) 区域。

实例

下图显示了网络视图中处于在线模式的设备。



TIA 门户简介

4.1 用户界面和操作

4.1.1 特定于操作系统的特性

4.1.1.1 用户权限的影响

用户权限不够时的限制

软件提供的若干功能需要直接访问编程设备/PC 的硬件，因而也需要直接访问所安装的操作系统。要充分利用这些功能，软件必须与操作系统紧密合作。为确保无故障交互，因此您需要以足够的用户权限登录操作系统。理想情况下，应该以管理员（拥有全部权限）身份登录。

特别是，如果用户权限不够，可能无法使用那些需要在线连接或更改接口卡设置的功能。

识别受限功能

可以通过下列方法识别需要特殊权限的功能：

- 功能旁边显示有盾形图标。
该功能可以使用，但受用户帐户控制的管制。
- 框灰显且无法访问。
需要管理员权限才能访问该框。在某些操作系统环境中，可以通过输入管理员密码来获得管理员权限。

注意

灰显的框不一定表示权限不够。还应该检查层叠工具提示中的附加信息，找出能够编辑该框的条件。

4.1.1.2 扩展用户权限

消除因用户权限引起的限制

如果没有以足够的权限登录操作系统，某些功能可能无法使用。可以通过下列方法消除这些限制：

- 使用 Windows 用户帐户控制启用扩展权限
- 获得管理员权限
 - 以管理员权限登录操作系统
 - 临时获得管理员权限

使用 Windows 用户帐户控制获得扩展权限

为了能使用由 Windows 用户帐户控制的盾形图标所指示的功能，请按以下步骤操作：

1. 单击带有盾形图标的框或按钮。

将打开 Windows 用户帐户控制的安全提示。
2. 按 Windows 用户帐户控制的指示操作，如有可能，在提示时输入管理员密码。

一旦解除限制，该功能就可以立即使用。

以管理员权限登录操作系统

为了能使用因用户权限不足而禁用的功能，请按以下步骤操作：

1. 关闭软件。
2. 从操作系统注销。
3. 以管理员权限登录操作系统。
4. 重新启动软件。

临时获得管理员权限

要临时获得管理员权限，请按以下步骤操作：

1. 单击“更改设置”(Change settings) 按钮。您可以在允许临时分配管理员权限的对话框中找到该按钮。

将打开用于输入管理员密码的操作系统对话框。
2. 输入管理员密码。

可以临时更改该设置。再次调用对话框时，必须重复该步骤。

注意

并非所有操作系统均支持该功能。如果没有出现“更改设置”(Change settings) 按钮或该按钮灰显，则需要以管理员权限登录操作系统。

4.1.2 启动、设置和退出 TIA Portal

4.1.2.1 启动和退出 TIA Portal

启动 TIA Portal

请按照下面的步骤启动 TIA Portal：

1. 在 Windows 中，选择“开始”>“程序”>“西门子自动化”(Siemens Automation) > Totally Integrated Automation Portal V10。

退出 TIA Portal

请按照下面的步骤退出 TIA Portal：

1. 在“项目”(Project) 菜单中，选择“退出”(Exit) 命令。
如果该项目包含任何尚未保存的更改，系统将询问您是否要保存这些更改。
 - 选择“是”(Yes) 将更改保存在当前项目中，然后关闭 TIA Portal。
 - 选择“否”(No)，则仅关闭 TIA Portal 而不在项目中保存最近的更改。
 - 选择“取消”(Cancel) 则取消关闭过程。如果选择此选项，TIA Portal 将保持打开状态。

参见

[设置起始视图 \(页 101\)](#)
[启动设置 \(页 102\)](#)
[存储位置 \(页 103\)](#)

4.1.2.2 设置起始视图

您可以确定打开项目时将要显示的视图。可以选择下列选项之一：

- 最近的视图
该程序将在上次关闭程序时所显示的视图中打开。
- [Portal 视图 \(页 104\)](#)
始终在 Portal 视图中打开该程序。
- [项目视图 \(页 105\)](#)
始终在项目视图中打开该程序。

要求

已打开项目视图。

步骤

要确定起始视图，请按照下面的步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。

4.1 用户界面和操作

在工作区中将出现“设置”(Settings) 窗口。

2. 选择“常规”(General) 组。
3. 在“起始视图”(Start view) 区域中，选择所需的起始视图。

此更改将在下次启动该程序时生效。

参见

启动和退出 TIA Portal (页 101)

启动设置 (页 102)

存储位置 (页 103)

4.1.2.3 启动设置

可通过启动设置选择以下选项：

- 启动最近的项目
激活此选项后，每当程序启动时就会装载最近打开的项目。
- 显示最近所用项目的列表
使用此选项可确定在登陆界面的项目列表中显示的现有项目的数目。
- 装载最近的窗口设置
激活此选项后，每当程序启动时就会装载最近保存的窗口设置。

要求

已打开项目视图。

步骤

要确定启动设置，请按以下步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
在工作区中将出现“设置”(Settings) 窗口。
2. 选择“常规”(General) 组。
3. 在“启动设置”(Start settings) 区域中选择所需选项。

此更改将在下次启动该程序时生效。

参见

启动和退出 TIA Portal (页 101)

设置起始视图 (页 101)

存储位置 (页 103)

4.1.2.4 存储位置

您可以确定程序中用于保存新项目和新库的位置。这样就不必在创建项目和库时手动选择保存位置。

要求

已打开项目视图。

步骤

要确定保存设置，请按以下步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
在工作区中将出现“设置”(Settings) 窗口。
2. 选择“常规”(General) 组。
3. 在“存储位置”(Storage locations) 区域中，选择“最近使用的存储位置”(Recently used storage location) 或“默认存储位置”(Default storage location)。
4. 如果选择了“默认存储位置”(Default storage location)，请输入新项目和库的保存位置。
如果指定的位置不可用，则会使用标准保存位置设置。
5. 要应用一个尚不可用的保存位置，可单击“浏览”(Browse)。
6. 将打开“查找文件夹”(Find folder) 对话框。
7. 浏览到所需的目录，然后单击“创建新文件夹”(Create new folder)。
8. 输入新文件夹的名称。
此更改将在下次创建项目或库时生效。

参见

启动和退出 TIA Portal (页 101)
设置起始视图 (页 101)
启动设置 (页 102)

4.1.3 用户界面的布局

4.1.3.1 视图

视图

在自动化项目中可以使用两种不同的视图：

- [Portal 视图 \(页 104\)](#) 是面向任务的项目任务视图。
- [项目视图 \(页 105\)](#) 是项目各组件以及相关工作区和编辑器的视图。

可以使用链接在两种视图间进行切换。

4.1.3.2 Portal 视图

Portal 视图的用途

Portal 视图为您提供面向任务的工具视图。在此处，您可以快速确定要执行什么操作并为当前任务调用工具。如有必要，该界面会针对所选任务自动切换为项目视图(页 105)。

Portal 视图的布局

下图显示了 Portal 视图各组件的示例：



- ① 不同任务的登陆选项
- ② 所选登陆选项对应的操作
- ③ 所选操作的选择面板
- ④ 切换到项目视图
- ⑤ 当前打开的项目的显示区域

登陆选项

登陆选项为各个任务区提供了基本功能。在 Portal 视图中提供的登陆选项取决于所安装的产品。

所选登陆选项对应的操作

此处提供了在所选登陆选项中可使用的操作。可在每个登陆选项中调用上下文相关的帮助功能。

所选操作的选择面板

所有登陆选项中都提供了选择面板。该面板的内容取决于您当前的选择。

切换到项目视图

可以使用“项目视图”(Project view) 链接切换到项目视图。

当前打开的项目的显示区域

在此处可了解当前打开的是哪个项目。

参见

项目树 (页 107)
工作区 (页 110)
巡视窗口 (页 111)
任务卡 (页 113)
详细视图 (页 114)

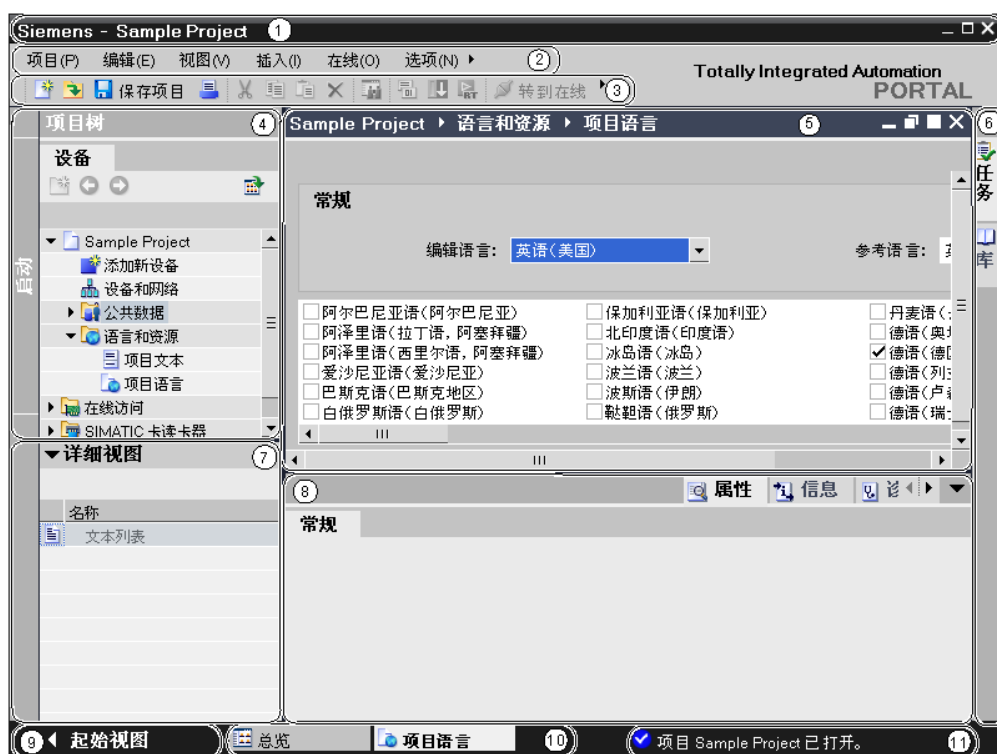
4.1.3.3 项目视图

项目视图的用途

项目视图是项目所有组件的结构化视图。

项目视图的布局

下图显示了项目视图各组件的示例：



- ① 标题栏
- ② 菜单栏
- ③ 工具栏
- ④ 项目树 (页 107)
- ⑤ 工作区 (页 110)
- ⑥ 任务卡 (页 113)
- ⑦ 详细视图 (页 114)
- ⑧ 巡视窗口 (页 111)
- ⑨ 切换到 Portal 视图 (页 104)
- ⑩ 编辑器栏
- ⑪ 状态栏

标题栏

项目名称显示在标题栏中。

菜单栏

菜单栏包含您工作所需的全部命令。

工具栏

工具栏提供了常用命令的按钮。这使您可以更快地访问这些命令。

切换到 Portal 视图

可以使用“Portal 视图”(Portal view) 链接切换到 Portal 视图。

编辑器栏

编辑器栏显示打开的编辑器。如果已打开多个编辑器，它们将组合在一起显示。可以使用编辑器栏在打开的元素之间进行快速切换。

状态栏

在状态栏上可以找到最近生成的报警。

4.1.3.4 项目树

项目树的功能

使用项目树功能可以访问所有组件和项目数据。可在项目树中执行以下任务：

- 添加新组件
- 编辑现有组件
- 扫描和修改现有组件的属性

项目树的布局

下图显示了项目树组件的示例：



- ① 标题栏
- ② 工具栏
- ③ 项目
- ④ 设备
- ⑤ 公共数据
- ⑥ 语言和资源
- ⑦ 在线访问
- ⑧ SIMATIC 卡读卡器

标题栏

项目树的标题栏有一个用于折叠项目树的按钮。项目树折叠后，该按钮将显示在左侧空白区域。它此时会从指向左侧的箭头变为指向右侧的箭头，并可用于重新打开项目树。

工具栏

可以在项目树的工具栏中执行以下任务：

- 创建新的用户文件夹；例如，为了组合“程序块”(Program blocks) 文件夹中的块。
- 向前浏览到链接的源，然后往回浏览到链接本身。

项目树中有两个用于链接的按钮。可使用这两个按钮从链接浏览到源，然后再往回浏览。

- 在工作区中显示所选对象的总览。

显示总览时，将隐藏项目树中元素的更低级别的对象和操作。要再次显示这些对象和操作，可将总览最小化。

项目

在“项目”文件夹中，您将找到与项目相关的所有对象和操作，例如：

- 设备
- 语言和资源
- 在线访问

设备

项目中的每个设备都有一个单独的文件夹，该文件夹具有内部的项目名称。属于该设备的对象和操作都排列在此文件夹中。

公共数据

此文件夹包含可跨多个设备使用的数据，例如公用消息类、脚本和文本列表。

语言和资源

可在此文件夹中确定项目语言和文本。

在线访问

该文件夹包含了编程设备/PC 的所有接口，即使未用于与模块通信的接口也包括在其中。

SIMATIC 卡读卡器

该文件夹用于管理所有连接到编程设备/PC 的读卡器。

参见

Portal 视图 (页 104)
项目视图 (页 105)
工作区 (页 110)
巡视窗口 (页 111)
任务卡 (页 113)

详细视图 (页 114)

4.1.3.5 工作区

工作区的功能

为进行编辑而打开的对象将显示在工作区内。例如，这些对象包括：

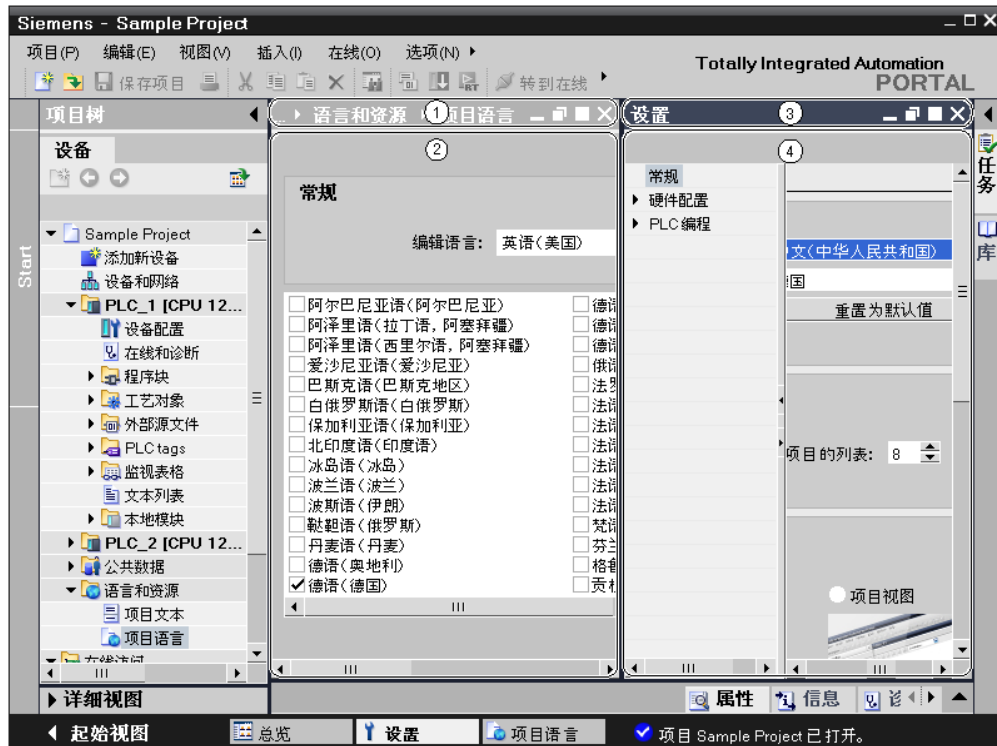
- 编辑器和视图
- 表格

可以打开若干个对象。但通常每次在工作区中只能看到其中一个对象。在编辑器栏中，所有其它对象均显示为选项卡。如果为了执行某些任务需要同时查看两个对象，可将工作区进行水平拆分或垂直拆分。如果没有打开任何对象，则工作区是空的。

另请参见：[调整工作区 \(页 117\)](#)

工作区的布局

下图显示了一个垂直拆分的工作区的示例：



- ① 左侧编辑器的标题栏
- ② 左侧编辑器的工作区
- ③ 右侧编辑器的标题栏

- ④ 右侧编辑器的工作区

参见

项目树 (页 107)
Portal 视图 (页 104)
项目视图 (页 105)
巡视窗口 (页 111)
任务卡 (页 113)
详细视图 (页 114)

4.1.3.6 巡视窗口

巡视窗口的功能

有关所选对象或所执行操作的附加信息均显示在巡视窗口中。

巡视窗口的布局

下图显示了巡视窗口的组成部分：





- ① “属性”(Properties) 选项卡
- ② “信息”(Info) 选项卡
- ③ “诊断”选项卡
- ④ 选项卡内的导航：
 - “属性”(Properties) 选项卡内的区域导航
 - “信息”(Info) 和“诊断”(Diagnostics) 选项卡中的更低级别选项卡

“属性”选项卡

此选项卡显示所选对象的属性。 可以在此处更改可编辑的属性。

“信息”选项卡

此选项卡显示有关所选对象的附加信息以及执行操作（例如编译）时发出的报警。

“诊断”选项卡

此选项卡提供有关系统诊断事件和已组态报警事件的信息。

选项卡内的导航

可以使用区域导航和更低级别的选项卡来显示在选项卡内所需的信息。

参见

项目树 (页 107)
 工作区 (页 110)
 Portal 视图 (页 104)
 项目视图 (页 105)
 任务卡 (页 113)
 详细视图 (页 114)

4.1.3.7 任务卡

任务卡的功能

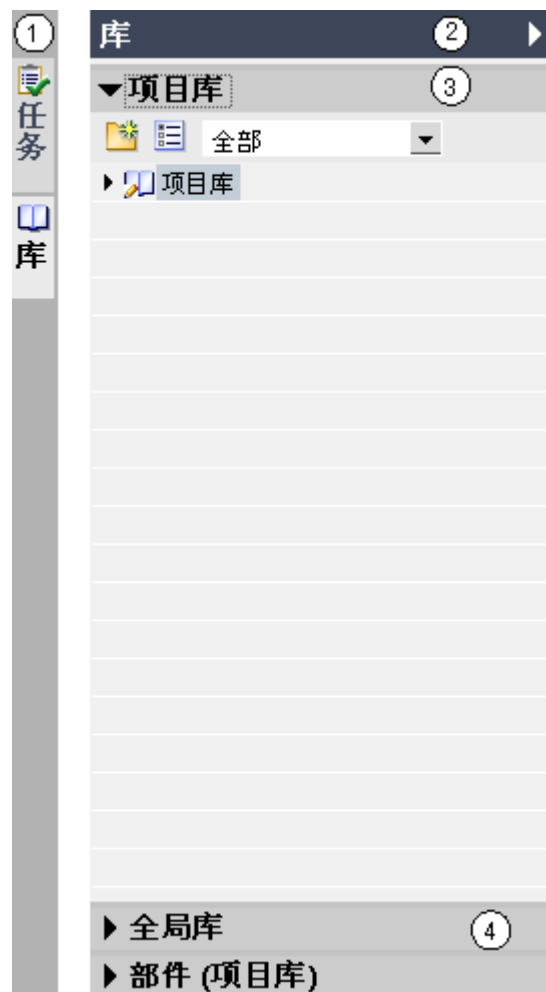
根据所编辑对象或所选对象，提供了用于执行附加操作的任务卡。这些操作包括：

- 从库中或者从硬件目录中选择对象
- 在项目中搜索和替换对象
- 将预定义的对象拖入工作区

在屏幕右侧的条形栏中可以找到可用的任务卡。可以随时折叠和重新打开这些任务卡。哪些任务卡可用取决于所安装的产品。比较复杂的任务卡会划分为多个窗格，这些窗格也可以折叠和重新打开。

任务卡的布局

下图显示了一个含有任务卡的条形区域实例：



① 关闭的任务卡

- ② 打开的“库”(Library) 任务卡
- ③ 打开的“项目库”(Project library) 窗格
- ④ 关闭的窗格

参见

项目树 (页 107)
 工作区 (页 110)
 巡视窗口 (页 111)
 Portal 视图 (页 104)
 项目视图 (页 105)
 详细视图 (页 114)

4.1.3.8 详细视图

详细视图的用途

在详细视图中显示所选对象的某些内容。其中可以包含文本列表或变量。但不显示文件夹的内容。要显示文件夹的内容，可使用项目树或总览窗口。

详细视图的布局

下图显示了详细视图的实例：



- ① 标题栏
- ② 所选对象的内容

标题栏

用于关闭详细视图的箭头位于详细视图的标题栏中。该窗口关闭后，箭头的方向会从指向左方变为指向右方。此时，可使用该箭头重新打开详细视图。

对象

显示的内容因所选对象而异。可通过拖放操作将对象的内容从详细视图移至所需位置。

参见

项目树 (页 107)
工作区 (页 110)
巡视窗口 (页 111)
任务卡 (页 113)
Portal 视图 (页 104)
项目视图 (页 105)
总览窗口 (页 115)

4.1.3.9 总览窗口

总览窗口的功能

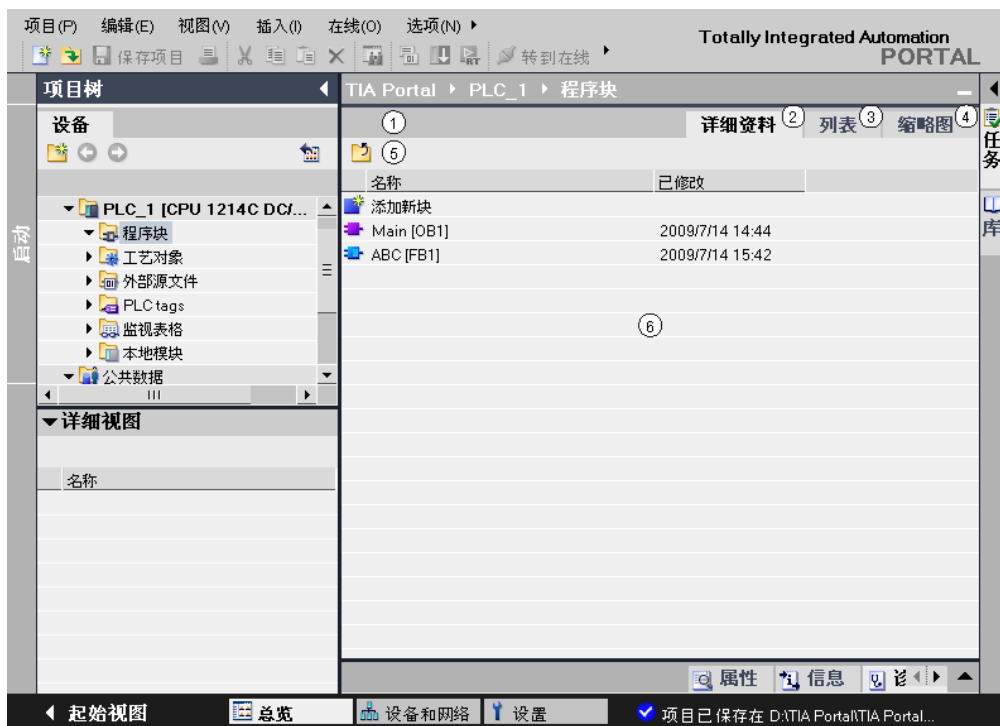
总览窗口是对项目树的补充。总览窗口显示项目树中当前所选文件夹的内容。

此外，可以在总览窗口中执行以下操作：

- 打开对象
- 在巡视窗口中显示和编辑对象的属性
- 重命名对象
- 从快捷菜单中调用对象特定的操作

总览窗口的布局

下图显示了总览窗口的组成部分：



- ① 总览窗口
- ② 切换到详细视图
- ③ 切换到列表视图
- ④ 切换到图标视图
- ⑤ 转到更高级别
- ⑥ 文件夹内容的显示区域

总览窗口的显示形式

可按以下形式显示总览窗口的内容：

- 详细视图
对象显示在一个含有附加信息（例如上次更改日期）的列表中。
- 列表视图
对象显示在一个简单列表中。
- 图标视图
以图标的形式显示对象。

参见

项目树 (页 107)

4.1.4 调整用户界面

4.1.4.1 选择语言

您可以更改用户界面的语言。

步骤

要更改用户界面语言，请按以下步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
2. 在区域导航中选择“常规”(General) 组。
3. 在“常规设置”(General settings) 中，从“用户界面语言”(User interface language) 下拉列表中选择所需语言。

用户界面语言设置即被更改。以后，程序将以该用户界面语言打开。

参见

调整工作区 (页 117)

4.1.4.2 调整工作区

您可以根据需要来调整工作区。为此，可使用以下功能：

- 将工作区最大化
可以通过单击关闭任务卡、项目树和巡视窗口。这样就增加了工作区的大小。
- 将工作区中的元素最小化
可以将工作区中打开的元素（例如编辑器或表格）最小化。但元素即使被最小化也仍然保持打开状态，可以通过编辑器栏快速地再次将其最大化。
- 垂直拆分或水平拆分工作区
如果想要在工作区中同时显示两个元素，可以垂直拆分或水平拆分工作区。
- 从工作区分离元素
也可将元素从工作区中完全分离出来，然后在新窗口中打开这些元素。
- 恢复工作区
如果已经将工作区最大化或已将元素从工作区分离出来，可以通过单击撤消这些操作。

将工作区最大化

要将工作区最大化，请按以下步骤操作：

1. 打开一个元素，如编辑器或表格。
该元素将出现在工作区中。
2. 单击元素标题栏上的“最大化”(Maximize) 按钮。
任务卡、项目树和巡视窗口随即折叠，工作区将以其最大尺寸显示。

将工作区最小化

要将工作区再次最小化，请按以下步骤操作：

1. 在显示的元素标题栏上，单击“向下还原”(Restore down) 按钮。
任务卡、项目树和巡视窗口将重新打开。

将工作区中的元素最小化

要将工作区中的元素最小化，请按以下步骤操作：

1. 单击元素标题栏上的“最小化”(Minimize) 按钮。
该元素随即最小化，但仍然可以通过编辑器栏访问该元素。

要将所有元素同时最小化，请按以下步骤操作：

1. 在“窗口”(Window) 菜单中，选择“全部最小化”(Minimize all) 命令。

将工作区中的元素最大化

要将工作区中的元素再次最大化，请按以下步骤操作：

1. 在编辑器栏上单击所需元素。
该元素随即最大化并出现在工作区中。

垂直拆分或水平拆分工作区

要垂直拆分或水平拆分工作区，请按以下步骤操作：

1. 在“窗口”(Window) 菜单中，选择“垂直拆分编辑器空间”(Split editor space vertically) 或“水平拆分编辑器空间”(Split editor space horizontally) 命令。
编辑器栏上您已单击的元素和下一个元素将并列显示或堆叠显示。

注意

如果工作区中的元素少于两个，则“垂直拆分编辑器空间”(Split editor space vertically) 和“水平拆分编辑器空间”(Split editor space horizontally) 功能均不可用。

从工作区分离元素

要从工作区中分离元素，请按以下步骤操作：

1. 单击元素标题栏上的“分离”(Detach) 按钮。
该元素将从工作区释放，并显示在单独的窗口中。现在，可将该窗口放置到任何想要的位置。如果已经将窗口最小化，可以通过编辑器栏将其恢复。

在工作区中停放元素

要将元素再次停放在工作区中，请按以下步骤操作：

1. 单击元素标题栏上的“嵌入”(Embed) 按钮。
该元素将再次出现在工作区中。

在工作区中的各元素之间进行切换

可随时在工作区中的各元素之间进行切换。要切换到上一个或下一个编辑器，请按以下步骤操作：

1. 在“窗口”(Window) 菜单中，选择“下一个编辑器”(Next editor) 或“上一个编辑器”(Previous editor) 命令。
将显示下一个或上一个编辑器。

参见

工作区 (页 110)
选择语言 (页 117)

4.1.4.3 调整表格

为了能阅读行中的所有文本，可以优化表格列的宽度。

优化列宽

要根据内容来调整列宽，请按以下步骤操作：

1. 在表头中，将鼠标指针置于要调整宽度的列的右侧分隔线上。
鼠标指针将变成一段垂线，两侧各有一个箭头，分别指向分隔线的左边和右边。
2. 双击分隔线。

注意

请注意以下事项：

- 如果列宽过窄，将鼠标指针短暂悬停在相应区域时，各单元格的全部内容将显示为工具提示。
 - 关闭项目时，列宽和可见列将与项目特定的设置一同保存。这意味着，即使在另一台计算机上打开该项目，所有设置在该计算机上也保持有效。
-

4.1.5 键盘快捷键

4.1.5.1 用于项目编辑的键盘快捷键

编辑项目

功能	按键组合	菜单命令
创建新项目	<Ctrl+N>	项目 (Project) > 新建 (New)
打开项目	<Ctrl+O>	项目 (Project) > 打开 (Open)
关闭项目	<Ctrl+W>	项目 (Project) > 关闭 (Close)

4.1 用户界面和操作

功能	按键组合	菜单命令
保存项目	<Ctrl+S>	项目 (Project) > 保存 (Save)
以另一个名称保存项目	<Ctrl+Shift+S>	项目 (Project) > 另存为 (Save as)
打印项目	<Ctrl+P>	项目 (Project) > 打印 (Print)
编译项目	<Ctrl+B>	编辑 (Edit) > 编译 (Compile)

在项目内编辑对象

功能	按键组合	菜单命令
重命名项目	<F2>	编辑 (Edit) > 重命名 (Rename)
在某区域中高亮显示所有对象	<Ctrl+A>	编辑 (Edit) > 全选 (Select all)
复制对象	<Ctrl+C>	编辑 (Edit) > 复制 (Copy)
剪切对象	<Ctrl+X>	编辑 (Edit) > 剪切 (Cut)
粘贴对象	<Ctrl+V>	编辑 (Edit) > 粘贴 (Paste)
删除对象		编辑 (Edit) > 删除 (Delete)
查找或替换对象	<Ctrl+F>	编辑 (Edit) > 查找和替换 (Find and replace)

调用帮助功能

功能	按键组合	菜单命令
调用帮助功能	<F1> 或 <Shift+F1>	帮助 (Help) > 显示帮助 (Show help)

4.1.5.2 用于窗口的键盘快捷键

打开和关闭窗口

功能	按键组合	菜单命令
打开/关闭项目树	<Ctrl+1>	视图 (View) > 项目树 (Project tree)
打开/关闭详细视图	<Ctrl+4>	视图 (View) > 详细视图 (Details view)
打开/关闭总览	<Ctrl+2>	视图 (View) > 总览 (Overview)
打开/关闭任务卡	<Ctrl+3>	视图 (View) > 任务卡 (Task card)
打开/关闭巡视窗口	<Ctrl+5>	视图 (View) > 巡视窗口 (Inspector window)

功能	按键组合	菜单命令
关闭所有编辑器	<Ctrl+Shift+F4>	窗口 (Window) > 全部关闭 (Close all)
打开快捷菜单	<Shift+F10>	-

4.1.5.3 项目树中的键盘快捷键

项目树中的键盘快捷键

功能	按键组合
跳转到项目树的起始处	<Home> 或 <Page Up>
跳转到项目树的结束处	<End> 或 <Page Down>
打开文件夹	<右箭头>
关闭文件夹	<左箭头>

4.1.5.4 表格中的键盘快捷键

表格中的常规键盘快捷键

功能	按键组合
将单元格置于编辑模式	<F2> 或 <回车键>
在单元格中打开下拉列表框	<回车键>

在表格中导航

功能	按键组合
转到下一个单元格	<箭头键>
转到右侧的下一个可编辑单元格	<Tab>
转到左侧的下一个可编辑单元格	<Shift+Tab>
屏幕上移	<PgUp>
屏幕下移	<PgDn>
转到行中第一个单元格	<Home>

4.1 用户界面和操作

功能	按键组合
转到行中最后一个单元格	<End>
转到表格中第一个单元格	<Ctrl+Home>
转到表格中最后一个单元格	<Ctrl+End>
转到列中顶部单元格	<Ctrl+上箭头>
转到列中底部单元格	<Ctrl+下箭头>

高亮显示表格中的区域

功能	按键组合
高亮显示列	<Ctrl+空格键>
高亮显示行	<Shift+空格键>
高亮显示所有单元格	<Ctrl+A>
高亮显示以扩展单元格	<Shift+箭头键>
将高亮显示扩展到第一个可见单元格	<Shift+PgUp>
将高亮显示扩展到最后一个可见单元格	<Shift+PgDn>
将高亮显示扩展到第一行	<Ctrl+Shift+上箭头>
将高亮显示扩展到最后一行	<Ctrl+Shift+下箭头>
将高亮显示扩展到行中第一个单元格	<Ctrl+Shift+左箭头>
将高亮显示扩展到行中最后一个单元格	<Ctrl+Shift+右箭头>

4.1.5.5 用于文本编辑的键盘快捷键

编辑文本

功能	按键组合
切换到插入模式或覆盖模式	<Insert>
退出编辑模式	<Esc>
删除	
删除字符	<Backspace>

4.1.5.6 屏幕键盘

简介

使用 TIA Portal 时，也可以使用 Microsoft 屏幕键盘。

显示屏幕键盘

要显示屏幕键盘，请按以下步骤操作：

1. 在“视图”(View) 菜单中，选择“屏幕键盘”(Screen keyboard) 命令。

退出屏幕键盘

要退出屏幕键盘，请按以下步骤操作：

1. 在屏幕键盘的“文件”(File) 菜单中，选择“退出”(Exit) 命令。

4.2 信息系统的帮助

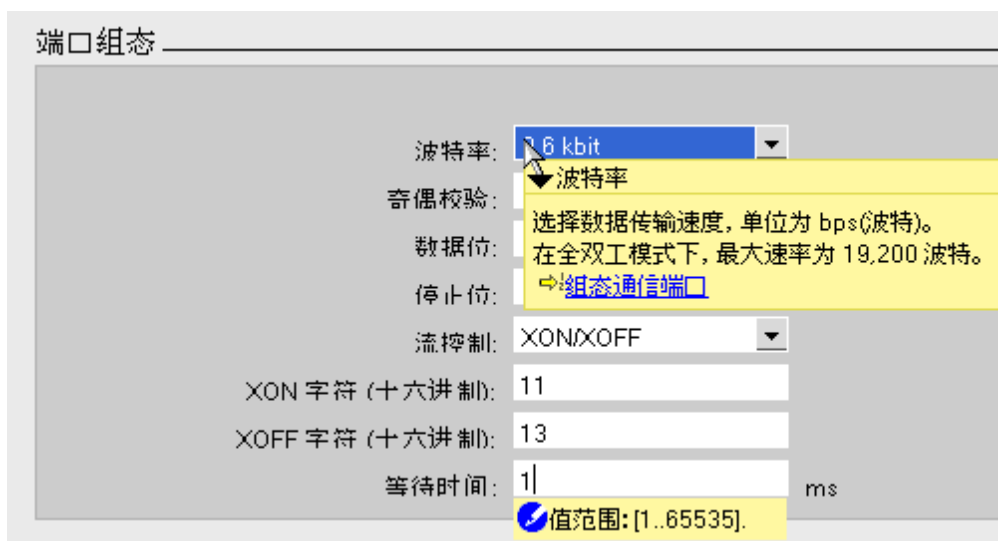
4.2.1 信息系统的常规备注

快速解答问题

全面的帮助系统可用来帮助您完成任务。帮助系统介绍了基本概念、说明和功能。使用程序时，您还会得到以下支持：

- 提示在对话框中进行正确输入的弹出项
- 提供有关用户界面元素（例如，文本框、按钮和图标）信息的工具提示。有些工具提示还补充有包含更精确信息的层叠项。
- 当前上下文的帮助，例如，按 <F1> 或 <Shift+F1> 键时可获得菜单命令的帮助。

下图显示了一个层叠工具提示（上方）和弹出项（下方）的实例：



帮助

帮助系统介绍了概念、说明和功能。而且还包含参考信息和实例。

帮助系统在屏幕右侧单独打开。





可以在帮助系统中显示导航区域。在此可访问下列功能：

- 目录
- 搜索索引
- 整个帮助的全文搜索
- 收藏夹

有关使用帮助的更多信息，请参见“[使用帮助系统 \(页 126\)](#)”一章。

帮助系统中主题的标识

根据帮助主题所包含的信息类型，用不同的符号来标识各种帮助主题。

符号	信息类型	说明
	操作说明	描述执行特定任务需遵守的步骤。
	实例	包含解释任务的具体实例。
	实际信息	包含执行任务时需要了解的背景信息。
	参考	包含详细的参考信息，可供后面参考。

弹出项

某些文本框提供的弹出信息，可帮助您输入有效的参数和值。弹出项提示您这些文本框中合理的取值范围和数据类型。

下图显示了弹出项（黄色）和指示值无效的弹出错误消息（红色）：



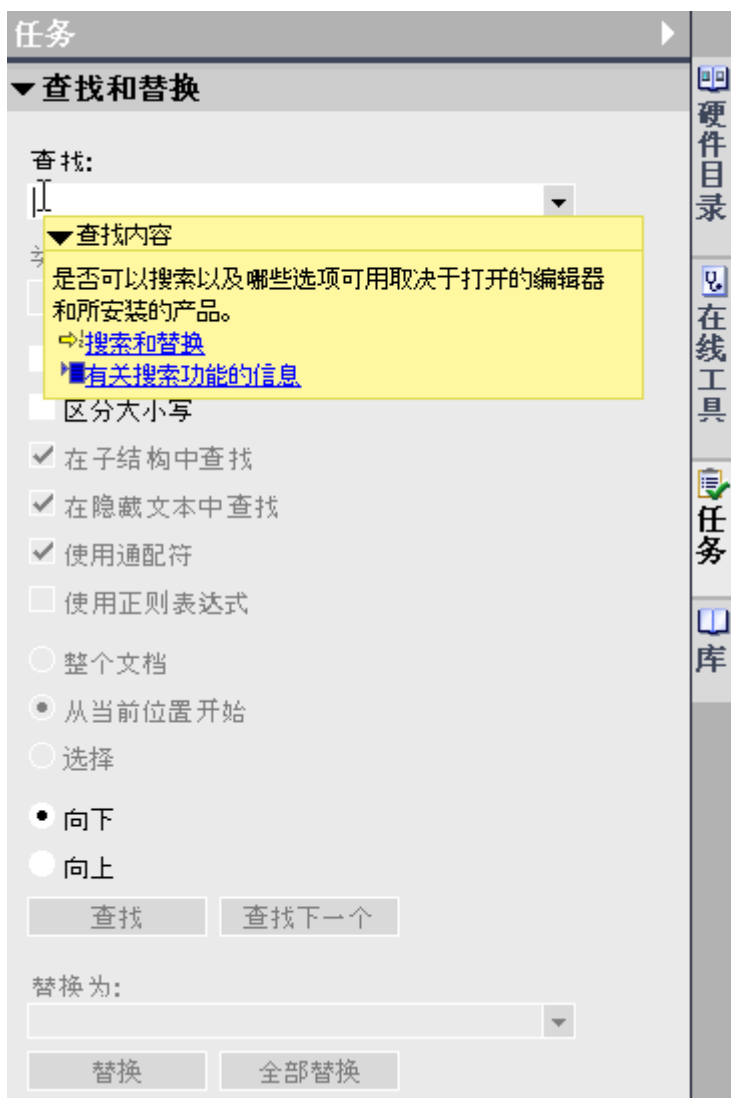
工具提示

界面元素提供了工具提示，从而使标识更容易。

左侧具有箭头图标的工具提示，其工具提示层叠项中包含了更多信息。如果将鼠标指针短暂停留在工具提示上或单击箭头图标，该信息就会显示。可禁用工具提示层叠项的自动显示。

如果帮助系统中包含附加信息，则层叠项中会出现指向相应帮助主题的连接。如果单击该链接，相应主题便会在帮助中打开。

下图显示了层叠项已打开的工具提示：



参见

[使用帮助 \(页 126\)](#)

[禁用工具提示层叠项的显示 \(页 129\)](#)

[安全指南 \(页 129\)](#)

4.2.2 使用帮助

打开帮助系统

可以通过以下方式打开帮助系统：

1. 在“帮助”(Help) 菜单中，选择命令“显示帮助”(Show help) 或按 <F1> 键，可以显示相应上下文的相关帮助。

或

1. 单击工具提示层叠项中的链接，可以直接转到帮助系统中的其它位置。

通过工具栏导航

单击工具栏左侧的“显示/隐藏目录”(Show/hide table of contents) 按钮，可显示/隐藏具有目录、搜索功能、索引和收藏夹的导航区域。

使用“前进”(Forward) 和“后退”(Back) 按钮可再次显示前面的帮助主题。

单击“显示启动页”(Display start page) 按钮可返回到帮助系统的起始页面。

通过菜单导航

帮助系统有其自己的菜单，其中包含菜单项“实例”(Examples)、 “另请参见”(See also)、 “历史”(History) 和“其它”(Extra)。

- 实例
如果有实例，则会在当前打开的主题中提供相应的实例。
- 另请参见
如果存在内容与当前主题相关的其它帮助主题，则在此处也可以找到这些主题。从列表中选择 一个主题，以获得更多信息。
- 历史
选择一个最近查看过的主题可返回到该主题。
- 其它
选择“词汇表”(Glossary) 命令可调用词汇表。词汇表按字母顺序对重要词进行了解释。

根据索引查找帮助主题

1. 单击帮助工具栏中的“显示/隐藏目录”(Show/hide table of contents) 按钮。
目录随即显示且“索引”(Index)、“搜索”(Search) 和“收藏夹”(Favorites) 选项卡可见。
2. 打开“索引”(Index) 选项卡。
3. 在输入框中输入搜索词或从关键字列表中选择搜索词。
4. 单击“显示”(Display)。

全文搜索

1. 单击帮助工具栏中的“显示/隐藏目录”(Show/hide table of contents) 按钮。
目录随即显示且“索引”(Index)、“搜索”(Search) 和“收藏夹”(Favorites) 选项卡可见。
2. 打开“搜索”(Search) 选项卡。
3. 在文本框中键入搜索词。
4. 必要时，使用附加条件进一步搜索：
 - 选择“在搜索结果中搜索”(Search previous results) 可启动仅对上次搜索到的内容再次搜索的操作。
 - 选择“搜索相似词”(Search for similar words) 可查找与搜索词仅有略微差别的词。
 - 选择“仅搜索标题”(Search titles only) 可搜索到仅标题中包含搜索词的结果。搜索期间不考虑帮助主题的内容。

5. 单击搜索域右侧的箭头图标可使用逻辑运算。可以使用下列逻辑运算：
 - 使用“AND”运算符组合两个或多个搜索词，可以仅查找包含文本中的所有搜索词的帮助主题。
 - 使用“OR”运算符组合两个或多个搜索词，可以仅查找包含文本中的一个或多个搜索词的帮助主题。
 - 使用“NEAR”运算符组合两个或多个搜索词，可以仅查找包含直接相关词（8 个词）的帮助主题。
 - 在词前加“NOT”运算符，可以搜索不包含该词的帮助主题。
6. 单击“列出主题”(List topics) 启动搜索。

结果随即被列出，其中包含标题、位置和等级。“位置”(Position) 列包含已找到的帮助主题所在的章节。根据所找到的帮助主题在目录中的位置，以及帮助主题中的搜索结果，按等级进行排序。

使用收藏夹

可将单个帮助主题保存在收藏夹中。这样可节省再次搜索这些帮助主题的时间。

要将帮助主题保存在收藏夹中，请按以下步骤操作：

1. 打开要收藏的帮助主题或章节。
 2. 单击帮助工具栏中的“显示/隐藏目录”(Show/hide table of contents) 按钮。

目录随即显示且“索引”(Index)、“搜索”(Search) 和“收藏夹”(Favorites) 选项卡可见。
 3. 打开“收藏夹”(Favorites) 选项卡。
 4. 单击“添加”(Add)。
- 帮助主题或章节随即保存在收藏夹中，并可在下次打开帮助系统时使用。

要打开已保存在收藏夹中的帮助主题，请按以下步骤操作：

1. 单击帮助工具栏中的“显示/隐藏目录”(Show/hide table of contents) 按钮。

目录随即显示且“索引”(Index)、“搜索”(Search) 和“收藏夹”(Favorites) 选项卡可见。
2. 打开“收藏夹”(Favorites) 选项卡。
3. 从列表中选择要打开的主题。
4. 单击“显示”(Display) 按钮。

要从收藏列表中删除帮助主题，请按以下步骤操作：

1. 单击帮助工具栏中的“显示/隐藏目录”(Show/hide table of contents) 按钮。

目录随即显示且“索引”(Index)、“搜索”(Search) 和“收藏夹”(Favorites) 选项卡可见。
2. 打开“收藏夹”(Favorites) 选项卡。
3. 从列表中选择要删除的主题。
4. 单击“删除”(Remove) 按钮。

打印信息

可以打印帮助系统的所有内容，也可以仅打印单个主题。要标记想要打印的主题，请按以下步骤操作：

1. 单击“打印”(Print) 按钮。目录随即在单独的窗口中打开。
2. 在“打印帮助主题”(Print help topics) 中，标记想要打印的文件夹和帮助主题。

3. 单击“打印”(Print) 按钮打印所选信息。
将打开“打印”(Print) 对话框。
4. 选择要用于打印帮助主题的打印机。
5. 如果要进行其它打印机设置，请单击“设置”(Settings)。
6. 单击“打印”(Print) 确认输入。
将在所选的打印机上打印帮助主题。

参见

安全指南 (页 129)
禁用工具提示层叠项的显示 (页 129)

4.2.3 禁用工具提示层叠项的显示

可禁止工具提示层叠项的自动显示。但是还可以手动显示。

步骤

要禁用工具提示层叠项的自动显示，请按以下步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
2. 在区域导航中选择“常规”(General) 组。
3. 取消选中“常规设置”(General settings) 中的“自动显示层叠工具提示”(Show cascading tooltips automatically) 复选框。

如果要手动显示工具提示层叠项，请单击工具提示中的箭头图标。

参见

信息系统的常规备注 (页 123)
使用帮助 (页 126)

4.2.4 安全指南

安全指南

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

表示如果不采取适当的预防措施，将导致死亡或严重的人身伤害。



警告

表示如果不采取适当的预防措施，可能会导致死亡或严重的人身伤害。



小心

带有警告三角，表示如果不采取适当的预防措施，可能会造成轻微的人身伤害。

小心

不带警告三角，表示如果不采取适当的预防措施，可能会造成财产损失。

注意事项

表示如果不考虑相应的提示，可能会出现意外的结果或情况。

如果存在多种危险等级，则采用最高等级的警告提示。带有警告三角的人身安全警告可能也包含财产损失的警告。

合格的专业人员

必须结合本文档来安装和使用该设备/系统。只有合格的专业人员才能调试和操作设备/系统。在本文档的安全注意事项中，合格的专业人员是指经授权按照既定安全惯例和标准，对设备、系统和线路进行调试、接地和标记的人员。

规定用法

请注意以下事项：



警告

本设备只能用于目录或技术说明中所述的应用场合，如果要使用其他公司的产品和组件，必须得到西门子的推荐和允许。只有正确地运输、储存、定位和装配产品，并且小心地对其进行操作和维护，产品才能正常、可靠地运行。

商标

所有带有标记符号®的都是西门子股份有限公司的注册商标。本出版物中的其它商标可能是注册商标，任何第三方擅自使用此商标可能会侵犯注册商标所有人的权利。

免责声明

我们已对本出版物中的内容进行了审核，以确保所述内容与所介绍的硬件和软件相一致。由于差错在所难免，因此我们无法保证完全一致。但是，我们会定期审查本手册中的信息，并在后续的版本中作必要的修正。

编辑项目

5.1 项目的基本知识

简介

项目用于有条理地存储因创建自动化解决方案而产生的数据和程序。构成项目的数据包括：

- 有关硬件结构的组态数据和模块的参数分配数据
- 用于网络通信的项目工程数据
- 用于设备的项目工程数据

项目层级

数据以对象的形式存储在项目中。在项目中，对象按树形结构安排（项目层级）。

项目层级根据设备和工作站以及属于它们的组态数据和程序而构成。

例如，项目和在线访问的公共数据也显示在项目树中。

参见

创建新项目 (页 133)

打开项目 (页 134)

保存项目 (页 134)

删除项目 (页 135)

5.2 创建新项目

步骤

要创建新项目，请按以下步骤操作：

1. 在“项目”(Project) 菜单中，选择“新建”(New) 命令。
“创建新项目”(Create a new project) 对话框随即打开。
2. 输入项目名称和路径，或接受建议的设置。
3. 单击“创建”(Create) 按钮。

结果

新项目即会创建并显示在项目树中。

参见

项目的基本知识 (页 133)
打开项目 (页 134)
保存项目 (页 134)
删除项目 (页 135)

5.3 打开项目

步骤

要打开现有项目，请按以下步骤操作：

1. 在“项目”(Project) 菜单中，选择“打开”(Open) 命令。
将打开“打开项目”(Open project) 对话框，其中包括最近所用项目的列表。
2. 从列表中选择个项目，然后单击“打开”(Open)。
3. 如果所需的项目没有包括在列表中，则单击“浏览”(Browse) 按钮。浏览到所需的项目文件夹，然后打开项目文件。文件扩展名为“.ap10”。

结果

项目在项目视图中打开。

参见

项目的基本知识 (页 133)
创建新项目 (页 133)
保存项目 (页 134)
删除项目 (页 135)

5.4 保存项目

可以随时以相同或不同的名称保存项目。甚至在项目仍然包含错误元素时也可以保存项目。

保存项目

要保存项目，请按以下步骤操作：

1. 在“项目”(Project) 菜单中，选择“保存”(Save) 命令。
对项目的所有更改都以当前项目名称保存。

项目另存为

要以其它名称保存项目，请按以下步骤操作：

1. 在“项目”(Project) 菜单中，选择“另存为”(Save as) 命令。
将打开“将当前项目另存为”(Save current project as) 对话框。
2. 在“保存在”(Save in) 框中选择项目文件夹。
3. 在“文件名”(File name) 框中输入新项目名称。
4. 单击“保存”(Save) 确认输入。
项目即以新名称保存并打开。

参见

项目的基本知识 (页 133)
创建新项目 (页 133)
打开项目 (页 134)
删除项目 (页 135)

5.5 关闭项目

步骤

要关闭项目，请按以下步骤操作：

1. 在“项目”(Project) 菜单中，选择“关闭”(Close) 命令。
如果保存项目之后又进行过更改，则会显示一条消息。
2. 决定是否要保存更改。

5.6 删除项目

注意事项

删除项目时，全部项目数据都将从存储介质中删除。

要求

要删除的项目未打开。

步骤

请按下面的步骤操作来删除现有项目：

1. 在“项目”(Project) 菜单中，选择“删除项目”(Delete project) 命令。
将打开“删除项目”(Delete project) 对话框，其中包括最近所用项目的列表。
2. 从列表中选择个项目，然后单击“删除”(Delete) 按钮。

如果所需的项目没有包括在列表中，则单击“浏览”(Browse) 按钮。浏览到所需的项目文件夹，然后打开项目文件。文件扩展名为“.ap10”。

结果

从文件系统中删除全部项目文件夹。

参见

- 项目的基本知识 (页 133)
- 创建新项目 (页 133)
- 打开项目 (页 134)
- 保存项目 (页 134)

5.7 比较项目数据

5.7.1 比较编辑器概述

功能

使用比较编辑器可对在线与离线项目数据进行比较。这样用户能够查看差异并对差异做出反应。可进行比较的项目数据取决于所安装的产品。

比较编辑器的组成

比较编辑器由以下部分组成：



- ① 比较编辑器工具栏
- ② 表格式比较总览

比较编辑器工具栏

通过该工具栏可访问以下比较编辑器功能：

- 比较结果过滤器
使用过滤器，可以显示或隐藏在线与离线版本相同的对象。
- 开始详细比较
可以对在线与离线版本不同的对象启动详细比较。但是，此功能并非对每个对象都可用。
- 更新比较结果
修改对象后，可以使用此功能更新比较编辑器的视图。
- 同步比较操作
可通过特定、自定义的比较操作同步在线与离线版本不同的对象。所选的比较操作对所有对象均有效。同步后，这些对象的在线与离线版本相同。

表格式比较总览

下表给出了比较总览中各列的含义：

列	含义
对象名称	正在进行在线/离线比较的对象或文件夹的名称。
状态	在线/离线比较的状态，通过符号表示
动作	对象同步的动作
说明	所选动作的说明

比较总览的符号

比较的状态通过符号进行表示。下表给出了相关符号及其含义：

符号	列	含义
	状态	文件夹包含在线和离线版本不同的对象
	状态	对象的在线和离线版本相同
	状态	对象的在线和离线版本不同
	状态	对象仅离线存在
	状态	对象仅在线存在
	动作	无动作
	动作	在设备上装载对象
	动作	将对象从设备上传到编程设备/PC

状态信息也会显示在项目树中。

参见

- [在线/离线比较 \(页 138\)](#)
- [使用比较结果过滤器 \(页 138\)](#)
- [更新比较结果 \(页 139\)](#)
- [指定动作 \(页 140\)](#)
- [同步比较动作 \(页 140\)](#)

5.7.2 在线/离线比较

要求

项目树已打开。

步骤

要获取项目数据的比较状态总览，请按以下步骤操作：

1. 在项目树中选择设备。
2. 在快捷菜单中选择“比较离线/在线”(Compare offline/online) 命令。

如果尚未建立在线连接，则会打开“在线连接”(Connect online) 对话框。如果已定义在线连接，则会打开比较编辑器。

结果

将显示所有在线存在和离线存在的对象。可以根据图标查看这些对象的状态。现在可以定义对象所需的动作。

参见

- [比较编辑器概述 \(页 136\)](#)
- [使用比较结果过滤器 \(页 138\)](#)
- [更新比较结果 \(页 139\)](#)
- [指定动作 \(页 140\)](#)
- [同步比较动作 \(页 140\)](#)

5.7.3 使用比较结果过滤器

要使比较结果更易于查看，可以显示或隐藏那些在线和离线版本相同的对象。

要求

比较编辑器已打开。

显示相同对象

要再次显示相同对象，请按以下步骤操作：

1. 单击工具栏中的“比较结果过滤器”(Filter for comparison results) 按钮。

将再次显示所有元素。

隐藏相同对象

要隐藏相同对象，请按以下步骤操作：

1. 单击工具栏中的“比较结果过滤器”(Filter for comparison results) 按钮。
将仅显示在线和离线版本不同的元素。

参见

在线/离线比较 (页 138)
比较编辑器概述 (页 136)
更新比较结果 (页 139)
指定动作 (页 140)
同步比较动作 (页 140)

5.7.4 运行详细比较

可以显示在线和离线版本不同的对象的详细比较。

注意

此功能并非对每个对象都可用。

要求

比较编辑器已打开。

步骤

要显示对象的详细比较，请按以下步骤操作：

1. 在比较编辑器中选择对象。
2. 单击工具栏中的“开始详细比较”(Start detailed comparison) 按钮。
如果可对所选对象进行详细比较，则将打开该对象的在线和离线版本并高亮显示差异。

5.7.5 更新比较结果

只要更改了对象，比较结果就不再有效，因此必须进行更新。

要求

比较编辑器已打开。

步骤

要更新比较结果，请按以下步骤操作：

1. 单击工具栏中的“更新比较结果”(Update comparison results) 按钮。
比较结果即会更新。

参见

比较编辑器概述 (页 136)
在线/离线比较 (页 138)
使用比较结果过滤器 (页 138)
指定动作 (页 140)
同步比较动作 (页 140)

5.7.6 指定动作

如果执行了比较，则可以在比较编辑器中为不相同的对象指定要执行的动作。无法为相同的对象选择任何动作。

要求

比较编辑器已打开。

步骤

要为不相同的对象选择动作，请按以下步骤操作：

1. 在要为其选择动作的对象的行中，右键单击“动作”(Action) 列中的图标。
2. 在对象的快捷菜单中选择所需动作。

参见

比较编辑器概述 (页 136)
在线/离线比较 (页 138)
使用比较结果过滤器 (页 138)
更新比较结果 (页 139)
同步比较动作 (页 140)

5.7.7 同步比较动作

同步时，将执行为不相同对象所指定的动作。

另请参见：[指定动作 \(页 140\)](#)

要求

- 比较编辑器已打开。
- 已选择所需动作。

步骤

要同步比较动作，请按以下步骤操作：

1. 单击工具栏中的“同步在线和离线对象”(Synchronize online and offline) 按钮。

结果

将执行为对象指定的动作。

参见

比较编辑器概述 (页 136)

在线/离线比较 (页 138)

使用比较结果过滤器 (页 138)

更新比较结果 (页 139)

5.8 编译项目数据

5.8.1 编译项目数据的常规信息

编译项目数据

编译期间会转换项目数据，便于设备读取。硬件配置数据和程序数据可分别编译或一起编译。可以同时编译一个或多个目标系统的项目数据。

以下项目数据必须在装载前编译：

- 硬件项目数据，例如设备或网络和连接的组态数据
- 软件项目数据，例如程序块或过程画面

编译的范围

编译项目数据时，根据所涉及的设备可选择以下选项：

- 全部
- 硬件配置
- 软件
- 软件 (重建所有块)

参见

编译项目数据 (页 142)

5.8.2 编译项目数据

以下部分介绍了在项目树中编译项目数据的一般步骤。有关如何编译特定对象以及所有要特别注意的事项的详细信息，可参考有关编程和组态特定硬件的帮助。

步骤

要编译项目数据，请按以下步骤操作：

1. 在项目树中，右键单击要编译其项目数据的设备。
2. 在“编译”(Compile) 快捷菜单中选择所需选项。

注意

请注意，哪些选项可用取决于所选设备。

结果

编译项目数据。可以在巡视窗口中通过“信息 > 编译”(Info > Compile) 检查编译是否成功。

参见

编译项目数据的常规信息 (页 141)

5.9 在设备上装载项目数据

5.9.1 下载的常规信息

装载的项目数据

设备上装载的项目数据分为硬件项目数据和软件项目数据：

- 硬件项目数据是通过组态硬件、网络 and 连接而生成的。首次装载时会装载全部硬件项目数据。以后再装载时，将只装载组态的更改内容。
- 软件项目数据包括用户程序的块。首次装载时会装载全部软件项目数据。以后再次装载时，用户可以决定是装载整个软件还是只装载软件更改内容。

根据安装的适用范围，可装载单个对象、文件夹或完整设备。可通过以下方法进行装载：

- 在项目树中装载项目数据
- 将项目数据装载到可访问设备
可以拖放项目数据以将其装载到可访问设备。
- 将项目数据装载到存储卡
可以拖放项目数据以将其装载到存储卡。

装载项目数据的范围

将项目数据装载到项目的设备时，根据设备可选择以下选项：

- 全部
- 硬件配置
- 软件
- 软件 (所有块)

哪些选项可用取决于要装载的对象。

参见

将项目数据从项目树下载到设备 (页 143)

5.9.2 将项目数据从项目树下载到设备

以下部分介绍了在项目树中将项目数据下载到设备的一般步骤。有关如何装载特定对象以及所有要特别注意的事项的详细信息，可参考有关编程和组态特定硬件的帮助。

要求

- 项目数据一致。
- 每个设备均可通过在线访问进行访问。

步骤

要将项目数据装载到所选设备，请按以下步骤操作：

1. 在项目树中选择一个或多个设备系统。
2. 右键单击所选元素。
将打开快捷菜单。
3. 在“下载到设备”(Download to device) 菜单的快捷菜单中选择所需选项。

注意

请注意，哪些选项可用取决于所选设备。

如果之前尚未建立在线连接，则会打开“扩展的下载到设备”(Extended download to device) 对话框。如果已定义在线连接，则会打开“装载预览”(Load preview) 对话框。然后，继续执行步骤 6。

4. 从“扩展的下载到设备”(Extended download to device) 对话框中的“用于装载的 PG/PC 接口”(PG/PC interface for loading) 下拉列表框中选择编程设备/PC 的接口。如果有子网，则还应从“子网的连接”(Connection to subnet) 下拉列表中选择子网。
5. 在“目标子网中的可访问设备”(Accessible devices in target subnet) 表中选择设备并使用“装载”(Load) 确认选择。

必要时会编译项目数据。“装载预览”(Load preview) 对话框将打开。此对话框会显示消息并建议装载必需的操作。

6. 查看这些消息，并在必要时启用“操作”(Action) 列中的操作。

注意事项

如果存在功能混乱或程序错误，则在设备运行期间执行所建议的操作可能会导致严重的财产损失或人员受伤！

在启动该操作之前，请确保不会出现任何危险情况！

只要可以进行装载了，“装载”(Load) 按钮就可使用。

7. 单击“装载”(Load) 按钮。
8. 即会执行装载。“装载结果”(Load results) 对话框随即打开。在此对话框中，可检查装载是否成功并进行其它可能必要的操作。
9. 单击“完成”(Finish) 按钮。

结果

所选的项目数据已装载到设备。

参见

下载的常规信息 (页 142)

5.9.3 将项目数据装载到可访问设备

以下部分介绍了在项目树中将项目数据下载到可访问设备的一般步骤。有关如何装载特定对象以及所有要特别注意的事项的详细信息，可参考有关编程和组态特定硬件的帮助。

要求

可访问设备已显示。

另请参见：[显示可访问设备 \(页 1422\)](#)

步骤

要将项目数据装载到可访问设备，请按以下步骤操作：

1. 在项目树中，将包含设备的文件夹拖到可访问设备。
“装载预览”(Load preview) 对话框将打开。此对话框会显示消息并建议装载必需的操作。
2. 查看这些消息，并在必要时启用“操作”(Action) 列中的操作。

注意事项

如果存在功能混乱或程序错误，则在设备运行期间执行所建议的操作可能会导致严重的财产损失或人员受伤！

在启动该操作之前，请确保不会出现任何危险情况！

3. 只要可以进行装载了，“装载”(Load) 按钮就可使用。
4. 单击“装载”(Load) 按钮。
即会执行装载。“装载结果”(Load results) 对话框随即打开。在此对话框中，可检查装载是否成功并进行其它可能必要的所有操作。
5. 单击“完成”(Finish) 按钮。

5.10 打印项目内容

5.10.1 文档设置

简介

创建项目后，即可清楚地打印输出全部项目数据。结构良好的打印输出有助于编辑项目或执行服务工作。打印输出内容也可用于向客户展示。

文档设置

文档设置用于创建和管理书面设备文档。可以标准化电路图的形式准备项目并以统一的布局进行打印。用户可以打印整个项目或打印项目的精简总览。打印输出始终是基于当前项目工程创建的。

您也可以根据自己的要求设计打印页面的外观，例如，带有公司标识或使用公司布局。打印输出的封面更是提供了多个编辑选项。

软件还提供了可用于设计后续打印输出的模板。其中包括符合 ISO 标准的技术文档模板。

参见

打印项目数据 (页 146)

5.10.2 创建打印预览

创建打印预览

可以创建打印输出的预览。就像实际打印时一样，您可以选择打印模板和封面的模板。这些设置会保留下来，用于以后的打印。

步骤

要创建打印预览和设置以后的打印输出的范围，请按以下步骤操作：

1. 在“项目”(Project) 菜单中，选择“打印预览”(Print preview) 命令。
将打开“打印预览”(Print preview) 对话框。
2. 选择要用于打印输出的框架布局。
 - 在“框架”(Frame) 下拉列表中选择打印模板。
 - 取消选中“不包括封面”(Exclude cover page) 复选框以指定打印输出的封面。从“封面”(Cover page) 下拉列表中选择封面的模板。
3. 选中“精简”(Compact) 复选框以打印简化版本的项目数据。如果未选择此选项，将打印全部项目数据。
4. 单击“预览”(Preview) 确认
在工作视图中将创建打印预览。

参见

打印项目数据 (页 146)

5.10.3 打印项目数据

要求

至少组态了一台打印机。

打印项目数据

要打印当前项目的数据，请按以下步骤操作：

1. 在“项目”(Project) 菜单中，选择“打印”(Print) 命令。
将打开“打印”(Print) 对话框。
2. 在“名称”(Name) 框中选择打印机。
3. 如果要指定其它打印机设置，请单击“高级”(Advanced)。
4. 选择要用于打印输出的框架布局。
 - 在“框架”(Frame) 下拉列表中选择打印模板。
 - 取消选中“不包括封面”(Exclude cover page) 复选框以指定打印输出的封面。从“封面”(Cover page) 下拉列表中选择封面的模板。
5. 选中“精简”(Compact) 复选框以打印简化版本的项目数据。如果未选择此选项，将打印全部项目数据。
6. 单击“打印”(Print) 确认输入。
将在所选的打印机上打印输出项目数据。

参见

文档设置 (页 145)

5.11 移植项目

5.11.1 移植项目

现有项目的移植

可以将对象或整个项目从之前的自动化解决方案移植到 TIA Portal。每次移植时都将为移植的数据创建一个新项目，之后即可使用该项目。

组成项目的所有移植项都将清楚地显示在一个表格中。在该表格中，可以访问系统为每个移植项自动创建的日志文件。

可移植的项目部分和必须满足的要求取决于所使用的初始产品与当前安装的产品。有关产品移植选项的更多信息，请参见“服务和支持”网页以及您的软件产品文档。

参见

显示项目历史 (页 147)

显示移植的日志文件 (页 148)

5.11.2 显示项目历史

可以显示所有以前移植的总览表。

该表显示以下内容：

- 产品和创建项目时所使用的产品版本
- 产品和已集成移植的产品版本
- 每个移植的移植日志
- 每个移植的时间

步骤

要在总览表中显示以前的移植，请按以下步骤操作：

1. 在项目树中选择打开的项目。
2. 在该项目的快捷菜单中选择“属性”(Properties)。将打开该项目的“属性”(Properties) 对话框。
3. 在区域导航中选择“项目历史”(Project history)。将显示总览表。

参见

显示移植的日志文件 (页 148)

5.11.3 显示移植的日志文件

系统会为每个移植都创建一个 XML 格式的日志文件。可以在 Microsoft Internet Explorer 中显示该日志文件。

该日志文件包含以下信息：

- 移植的对象
- 因移植引起的对象修改内容
- 移植期间发生的错误

步骤

要显示移植的日志文件，请按以下步骤操作：

1. 在项目树中选择打开的项目。
2. 在该项目的快捷菜单中选择“属性”(Properties)。将打开该项目的“属性”(Properties) 对话框。
3. 在区域导航中选择“项目历史”(Project history)。将显示总览表。
4. 在“日志文件”(Log file) 列中单击日志文件的链接。将在 Microsoft Internet Explorer 中显示该日志文件。

5.12 在项目中查找和替换

5.12.1 有关搜索功能的信息

查找和替换

可以在编辑器中搜索文本。搜索功能可在当前打开的编辑器中查找包含搜索关键字的所有文本。在打开的编辑器中会按顺序选中结果。

用户还有以下选择：

- 使用附加选项缩小搜索范围
- 替换找到的文本

附加选项和可以搜索的文本类型取决于所安装的产品和当前打开的编辑器。

参见

搜索和替换 (页 149)

5.12.2 搜索和替换

使用查找功能

使用“查找和替换”(Find and replace) 功能可在编辑器中搜索或替换文本。

附加搜索选项

可通过选择以下附加选项之一来缩小搜索范围：

- 全字匹配
仅查找整个单词。忽略部分包含搜索关键字的单词。
- 区分大小写
搜索时考虑大写和小写字母。
- 在子结构中查找
搜索还包括包含在父对象中的文本。
- 在隐藏文本中查找
搜索还包括分配给其它文本但当前隐藏的文本。
- 使用通配符
输入星号作为代表任意数量字符的通配符。实例：如果要搜索所有以“Device”开头的词，则在搜索关键字框中键入“Device*”。
但是如果只想省去单个字符，则输入问号作为通配符。
- 使用正则表达式（仅用于在脚本中搜索）
正则表达式是用于描述数值集和过滤操作的字符串。通过它可创建复杂的搜索模式。

可用的附加选项取决于所安装的产品和打开的编辑器。

启动搜索

请按以下步骤启动“查找和替换”功能：

1. 在“编辑”(Edit) 菜单中选择“查找和替换”(Find and replace) 命令或在“任务”(Tasks) 任务卡中选择“查找和替换”(Find and replace) 窗格。
将打开“查找和替换”(Find and replace) 窗格。
2. 在“查找”(Find) 下拉列表中输入搜索词。
也可以从下拉列表中选择最近的搜索关键字。
3. 选择所需的搜索选项。
4. 使用选项按钮，选择搜索的起点和搜索方向。
 - 如果要搜索整个编辑器而不考虑当前选择内容，则选择“整个文档”(Whole document)。
 - 如果要从当前选择位置开始搜索，则选择“从当前位置开始”(From current position)。
 - 如果只想在当前选择内容中搜索，则选择“选择内容”(Selection)。
 - 选择“向下”(Down) 从上到下或从左到右搜索编辑器。
 - 选择“向上”(Up) 从下到上或从右到左搜索编辑器。

5. 单击“查找”(Find)。
将在编辑器中标出第一个搜索结果。
6. 单击“查找下一个”(Find next) 显示下一个搜索结果。
将在编辑器中标出下一个搜索结果。 重复此过程直至到达最后一个搜索结果。

替换搜索关键字

您可以选择分别替换搜索结果或自动替换所有找到的文本 (如果相应的编辑器支持此功能)。请按以下步骤替换词：

1. 在“查找”(Find) 下拉列表中输入搜索词。
也可以从下拉列表中选择最近的搜索关键字。
2. 选择所需的搜索选项。
3. 单击“查找”(Find) 按钮开始搜索特定的搜索关键字。
将在编辑器中显示第一个搜索结果。
4. 在“替换”(Replace) 下拉列表中，输入要用来替换搜索关键字的文本。
也可以从下拉列表中选择最近指定的文本。
5. 单击“替换”(Replace) 按钮将所选搜索结果替换为指定文本。
将替换找到的文本并在编辑器中标出下一个搜索结果。
重复此过程，直至如愿替换完所有搜索结果。要跳到下一个搜索结果而不替换标出的单词，请单击“查找”(Find) 按钮而不是“替换”(Replace)。
6. 单击“全部替换”(Replace all) 可立即自动替换所有搜索结果。

参见

有关搜索功能的信息 (页 148)

5.13 使用存储卡

5.13.1 关于存储卡的基本知识

简介

存储卡是有多种用途的插入式卡。有多种不同类型的存储卡。对于 S7-1200 系列的设备，只能使用 Siemens 的 SD 卡。可使用连接到编程设备的 SD 卡读卡器或直接通过设备访问 SD 卡。

在 CPU 中插入或从中取出存储卡之前，请确保已关闭 CPU。再次上电时 CPU 将自动重启。

使用存储卡

使用存储卡之前，需要先指定卡类型。根据所选的卡类型，可将卡用于不同的任务。可以从以下卡类型中选择：

- 程序
如果用作程序卡，则可以在存储卡上装载用户程序。在这种情况下，设备的内部装载存储器将被存储卡代替，并且会擦除内部装载存储器的数据。然后完全可以从存储卡执行用户程序。如果移除含有用户程序的存储卡，则不再有程序可用。
- 传送
如果用作传送卡，则可将用户程序从存储卡中传送到 CPU 的内部装载存储器。然后可再次移除存储卡。

参见

添加用户自定义读卡器 (页 151)

访问存储卡 (页 152)

选择存储卡的卡类型 (页 152)

显示存储卡的属性 (页 153)

5.13.2 添加用户自定义读卡器

简介

如果没有自动检测到读卡器，则可以手动添加。

要求

已打开项目视图。

步骤

要添加读卡器，请按以下步骤操作：

1. 打开项目树。
2. 在“项目”(Project) 菜单中选择“SIMATIC 卡读卡器 > 添加用户自定义读卡器”(SIMATIC Card Reader > Add user-defined Card Reader)。

将打开“添加用户自定义读卡器”(Add user-defined Card Reader) 对话框。

3. 在下拉列表框中，选择读卡器的路径。
4. 单击“确定”(OK) 确认输入。

参见

关于存储卡的基本知识 (页 150)

访问存储卡 (页 152)

选择存储卡的卡类型 (页 152)

显示存储卡的属性 (页 153)

5.13.3 访问存储卡

要求

- 读卡器中已插入存储卡。
- 已打开项目视图。

步骤

要访问存储卡，请按以下步骤操作：

1. 打开项目树。
2. 在“项目”(Project) 菜单中选择“SIMATIC 卡读卡器 > 显示 SIMATIC 卡读卡器”(SIMATIC Card Reader > Show SIMATIC Card Reader)。将项目树中打开“SIMATIC 卡读卡器”(SIMATIC Card Reader) 文件夹。
3. 打开“SIMATIC 卡读卡器”(SIMATIC Card Reader) 文件夹。现在即可访问存储卡。

参见

关于存储卡的基本知识 (页 150)
添加用户自定义读卡器 (页 151)
选择存储卡的卡类型 (页 152)
显示存储卡的属性 (页 153)

5.13.4 选择存储卡的卡类型

要求

- 读卡器中已插入存储卡。
- 已打开项目视图。

步骤

要选择存储卡的卡类型，请按以下步骤操作：

1. 右键单击要选择其卡类型的存储卡。
2. 在快捷菜单中选择“属性”(Properties) 命令。将打开“存储卡 <存储卡名称>”(Memory Card <name of the memory card>) 对话框。
3. 在“卡类型”(Card type) 下拉列表中，选择所需的卡类型。

参见

关于存储卡的基本知识 (页 150)
添加用户自定义读卡器 (页 151)
访问存储卡 (页 152)
显示存储卡的属性 (页 153)

5.13.5 显示存储卡的属性

属性概述

下表给出了您可以显示的当前存储卡的属性：

属性	说明
名称	存储卡的名称
写保护	显示存储卡是否为只读。
文件系统	存储卡的文件系统
储存容量	存储卡的可用存储空间
已用空间 (字节)	已使用的卡存储空间 (字节)
空闲空间 (字节)	存储卡的空闲存储空间 (字节)
序列号	如果存储卡支持此功能则显示存储卡的序列号。
卡类型	卡类型显示和选项

要求

- 读卡器中已插入存储卡。
- 已打开项目视图。

步骤

要显示存储卡的属性，请按以下步骤操作：

1. 右键单击要显示其属性的存储卡。
2. 在快捷菜单中选择“属性”(Properties) 命令。

将打开“存储卡 <存储卡名称>”(Memory Card <name of the memory card>) 对话框。属性即显示在此对话框中。

参见

关于存储卡的基本知识 (页 150)

添加用户自定义读卡器 (页 151)

访问存储卡 (页 152)

选择存储卡的卡类型 (页 152)

5.14 使用文本列表

5.14.1 文本列表

简介

可以集中管理要在报警中引用的文本。所有文本都存储在文本列表中。每个文本列表都具有唯一的名称，通过该名称可以调用其内容。文本列表中的每个文本都分配有一个取值范围。如果出现取值范围内的值，则调用相应文本。

可以将所有文本翻译成任意项目语言。此时用户有两种选择：

- 可以在列表中输入文本的译文。
- 可以将所有文本导出到 Excel 或 CSV 文件，并在电子表格程序中输入译文。然后再导入译文。

项目中的每个设备均有其自己的文本列表。因此，在项目树中，文本列表排列在设备下。

用户自定义和系统定义的文本列表

有两种类型的文本列表：

- 用户自定义文本列表

您可以自行创建用户自定义文本列表并填充文本；即，可以自行指定取值范围和相应的文本。用户自定义文本列表的名称以“USER”或用户选择的名称开头。

- 系统定义的文本列表

系统定义的文本列表由系统创建。这些列表始终包含与模块有关的文本。只要在项目中插入模块，这些列表就会自动创建。对于系统报警，文本列表的名称以“SYSTEM”开头。其包含的文本列表名称和取值范围无法修改。只能编辑分配给单个取值范围的文本。

用户自定义文本列表	系统定义的文本列表
用户自定义文本列表只能分配给一个设备。	系统定义的文本列表可以分配给设备以及整个项目。
用户可以创建新的文本列表和删除现有文本列表。	用户无法创建新的文本列表或删除文本列表。
用户可以在文本列表中添加和删除取值范围。	用户无法在文本列表中添加或删除取值范围。
用户可以指定取值范围以及相关的文本。	用户只能编辑与某个取值范围相关的文本。

设备特定的和跨设备的文本列表

设备特定的文本列表仅与项目中的某个设备有关，所以仅对该设备有效。因此，在项目树中，这些文本列表排列在设备下。设备特定的文本列表可以由用户自定义或由系统创建。

如果系统定义的文本列表通常对多个设备有效或不仅仅适用于一个设备，则会在项目树的“公共数据”(Common data) 下将它们归组。这些文本列表可用于所有设备。跨设备的文本列表总是由系统创建且仅用于系统诊断报警。因此，无法在“公共数据”(Common data) 下存储任何用户自定义的文本列表。

参见

项目文本基本知识 (页 157)
创建用户自定义文本列表 (页 155)
编辑用户自定义文本列表 (页 155)
编辑系统定义的文本列表 (页 156)

5.14.2 创建用户自定义文本列表

创建文本列表

可以为各个设备创建用户自定义文本列表。

要求

- 处于项目视图中
- 已打开一个项目
- 该项目至少包含一个设备

步骤

要创建用户自定义文本列表，请按以下步骤操作：

1. 在项目树中单击设备左侧的箭头。
将显示排列在该设备下的元素。
2. 双击“文本列表”(Text lists)。
将在表中列出的工作区内显示分配给该设备的所有文本列表。
3. 双击表中第一个空行。
将创建新的用户自定义文本列表。
4. 在“名称”(Name) 列中输入新文本列表的名称。
5. 从“选择”(Selection) 列的下拉列表中选择要指定十进制、二进制还是位形式的取值范围。
6. 在“注释”(Comment) 列中输入注释。
新的用户自定义文本列表即完成创建，并且现在可以输入取值范围和文本。

参见

编辑用户自定义文本列表 (页 155)

5.14.3 编辑用户自定义文本列表

编辑用户自定义文本列表

可以在用户自定义文本列表中输入取值范围和相应的文本。在项目树中，用户自定义文本列表始终位于设备下。

要求

- 处于项目视图中
- 已打开一个项目
- 该项目至少包含一个设备

向用户自定义文本列表添加取值范围和文本

要向用户自定义文本列表添加取值范围和文本，请按以下步骤操作：

1. 单击项目树中设备左侧的箭头或单击“公共数据”(Common data) 元素。
将显示排列在下方的元素。
2. 双击“文本列表”(Text lists)。
将在表中列出的工作区内显示分配给该设备的所有文本列表。
3. 在表中选择一个文本列表。
将在该文本列表下的工作区中打开另一个表，在其中可以输入取值范围和文本。
4. 在“范围始于”(Range from) 和“范围止于”(Range to) 列中输入所需的取值范围。必须按照为该文本列表选择的数字格式进行输入。
5. 在“条目”(Entry) 列中输入各个取值范围的文本。

参见

编辑系统定义的文本列表 (页 156)

直接翻译文本 (页 159)

5.14.4 编辑系统定义的文本列表

编辑系统定义的文本列表

在系统定义的文本列表中，只能修改分配给取值范围的各文本。

在项目树中，系统定义的文本列表位于设备或“公共数据”(Common data) 下。

要求

- 处于项目视图中
- 已打开一个项目
- 该项目至少包含一个设备

修改系统定义的文本列表中的文本

要添加或编辑系统定义的文本列表中分配给取值范围的文本，请按以下步骤操作：

1. 单击项目树中设备左侧的箭头或单击“公共数据”(Common data) 元素。
将显示排列在下方的元素。
2. 双击“文本列表”(Text lists)。

将在表中列出的工作区内显示分配给该设备或公用的所有文本列表。

3. 在表中选择一个文本列表。

将在文本列表下的工作区中打开另一个表。您可以在其中添加或编辑分配给取值范围的文本。

4. 在“条目”(Entry) 列中输入各个取值范围的文本。

参见

编辑用户自定义文本列表 (页 155)

直接翻译文本 (页 159)

5.15 使用多语言项目

5.15.1 项目文本基本知识

项目中不同语言的文本

处理期间在显示设备上输出的文本通常是以设计自动化解决方案时所用的语言输入的。对象的注释和名称也是以该语言输入的。

如果操作员不懂这种语言，则需要将所有操作员相关文本翻译成他们了解的语言。因此，您可以将所有文本翻译成任何语言。这样，可以确保以后遇到这些文本的任何人都能以其选择的语言来查看文本。

编辑语言和项目语言

每个项目都有编辑语言。输入文本时，文本始终以编辑语言创建。

项目语言是以后使用项目时所采用的所有语言。基于编辑语言，所有文本都可翻译成各种项目语言。

可以用多种语言管理的文本类型

例如，可以用多种语言管理下列类型的文本：

- 标题和注释
 - 块标题和块注释
 - 程序段标题和注释
 - STL 程序中的语句注释
 - 表中的注释
- 显示文本
 - 报警文本
 - 操作员相关的文本
 - 文本列表

翻译文本

有两种翻译文本的方法。

- 直接翻译文本
可在“项目文本”(Project texts) 表中直接输入各种项目语言的译文。
- 使用参考文本翻译文本
对于少量的文本，可以更改编辑器语言。所有文本单元格再次以默认值填充并可用当前语言填充。

注意

使用亚洲项目语言

只有操作系统中已安装的语言才能用作运行系统语言。如果要创建含有亚洲字体的项目，则需要在操作系统中启用对相应语言的支持。有关该步骤的详细信息，请参见操作系统文档。

参见

文本列表 (页 154)
选择项目语言 (页 158)
直接翻译文本 (页 159)
选择语言 (页 117)
设置编辑语言。(页 159)
使用参考文本翻译文本 (页 160)

5.15.2 选择项目语言

选择项目语言

可以使用同一种语言（为软件用户界面选择的语言）显示所有文本。这意味着必须存在相应语言的所有文本。您可以自己选择可用的语言。

要求

- 处于项目视图中
- 已打开一个项目

步骤

要选择项目语言，请按以下步骤操作：

1. 在项目树中单击“语言和资源”(Languages & resources) 左侧的箭头。
将显示其下面的元素。
2. 双击“项目语言”(Project languages)。
在工作区中，将看到可选择的语言列表。
3. 选择所需的语言。

所有文本即可以使用启用的语言显示。

5.15.3 设置编辑语言。

设置编辑语言。

项目中的所有文本都是以编辑语言创建的。如果更改编辑语言，以后输入的所有文本都将以新的编辑语言进行存储。

要求

- 处于项目视图中
- 已打开一个项目

步骤

要更改编辑语言，请按以下步骤操作：

1. 在项目树中单击“语言和资源”(Languages & resources) 左侧的箭头。
将显示下级元素。
2. 双击“项目语言”(Project languages)。
将在工作区中显示可能的项目语言设置。
3. 在“常规 > 编辑语言”(General > Editing language) 中选择编辑语言。

5.15.4 直接翻译文本

翻译文本

如果在项目中使用多种语言，则可以用所选项目语言直接输入各文本的译文。只要更改软件用户界面的语言，就可以所选的语言输入翻译文本。

要求

- 处于项目视图中
- 已打开一个项目
- 已选择至少一种其它项目语言。

步骤

要翻译各文本，请按以下步骤操作：

1. 在项目树中单击“语言和资源”(Languages & resources) 左侧的箭头。
将显示其下面的元素。
2. 双击“项目文本”(Project texts)。
将在工作区中显示项目中文本的列表。每种项目语言都对应单独的一列。

- 要将相同的文本归组并立即将其全部翻译，请单击工具栏中的“启用/禁用分组”(Switch on/off grouping) 按钮。
 - 要隐藏没有译文的文本，请单击工具栏中的“启用/禁用空文本过滤器”(Filter for empty texts on/off) 按钮。
3. 在相关列中输入译文。

注意

如果文本没有特定语言的译文，则显示英语文本。

5.15.5 使用参考文本翻译文本

简介

更改编辑语言后，所有文本都将以新的编辑语言显示在输入框中。如果还没有该语种的译文，则输入框为空或以默认值填充。

如果再次在输入域中输入文本，则该文本将以当前编辑语言进行保存。之后，该输入域的文本以两种项目语言形式存在，即先前的编辑语言和当前的编辑语言。这样便可用多种项目语言创建文本。

可以用其它项目语言显示输入框中的现有译文。这些语言充当当前编辑语言的输入文本的参照，称为参考语言。

注意

“显示参考文本”(Show reference texts) 功能取决于所安装的产品和打开的编辑器。

要求

输入域至少有一种不同项目语言的译文。

步骤

要以参考语言显示输入单元格的译文，请按以下步骤操作：

1. 在“任务”(Tasks) 任务卡中，选择“语言和资源”(Languages & resources) 窗格。
2. 从“参考语言”(Reference language) 下拉列表中选择参考语言。

结果

参考语言已预置。如果在文本块中单击，将在“任务 > 参考文本”(Tasks > Reference text) 任务卡中显示已经以其它项目语言形式存在的译文。

5.16 使用库

5.16.1 库基本知识

简介

您可以将想要多次使用的对象存储在库中。可以在项目范围内或跨项目重复使用已存储的对象。这意味着您可以创建块模板以便在不同项目中使用并根据自动化任务的特定要求修改这些模板。

库类型

根据任务，用户可使用以下库类型之一：

- 项目库
每个项目都有其自己的库。您可在其中存储想要在项目中多次使用的对象。该项目库总是随当前项目一起打开、保存和关闭。
- 全局库
除项目库外，还可以创建其它库。您可在其中存储想要在多个项目中使用的对象。可以创建、更改、保存和传送这些独立于项目的全局库。
在全局库区域中，还能找到随软件一起提供的库。这些库包括可以在项目中使用的现成功能和功能块。您无法修改所提供的库。

库对象

库可以容纳大量对象。例如，这包括：

- 功能 (FC)
- 功能块 (FB)
- 数据块 (DB)
- 设备
- PLC 数据类型
- 监视表格
- 过程画面
- 面板

使用类型

可将库对象用作副本模板或实例。副本模板用于创建实际副本。如果以后更改模板，将不会对使用中的副本进行这些更改。但如果创建实例，更改将应用到每个使用点。

注意

可是，并不是每个对象都能用作实例。

参见

项目库基本知识 (页 165)

全局库基本知识 (页 169)

5.16.2 “库”任务卡

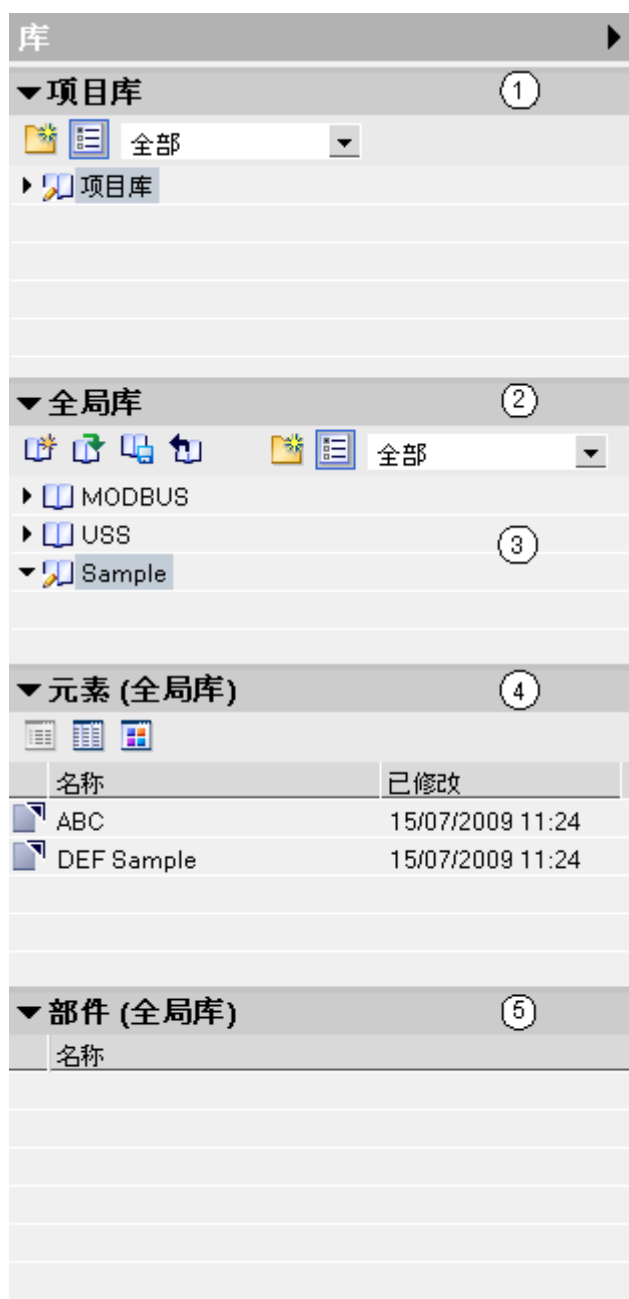
“库”任务卡的功能

通过“库”(Libraries) 任务卡可以高效使用项目库和全局库。

可根据需要显示或隐藏该任务卡。

“库”任务卡的结构

“库”(Libraries) 任务卡由以下部分组成：



- ① “项目库”窗格
- ② “全局库”窗格
- ③ 所提供的全局库
- ④ “元素”窗格
- ⑤ “部件”窗格

“项目库”窗格

在该窗格中，可以存储想要在项目中多次使用的对象。

“全局库”窗格

在该窗格中，可以存储想要在不同项目中多次使用的对象。

所提供的全局库

“全局库”(Global library) 窗格还列出了随系统一起提供的库。这些库提供了现成的功能和功能块。可以使用这些提供的全局库但无法修改它们。

“元素”窗格

在该窗格中，可以显示库的元素。

“部件”窗格

在该窗格中，可以显示库元素的内容。

5.16.3 使用元素和部件视图

简介

首次打开“库”(Libraries) 任务卡时，将打开“项目库”(Project library) 和“全局库”(Global libraries) 窗格，而“部件”(Parts) 窗格为关闭状态。您需要显式打开“元素”(Elements) 窗格。

元素视图显示所选库的元素。可以选择下列视图之一：

- 详细资料
- 列表
- 总览

部件视图显示所选库元素的内容。

要求

已显示“库”(Libraries) 任务卡。

步骤

要使用元素和部件视图，请按以下步骤操作：

1. 在“项目库”(Project library) 或“全局库”(Global libraries) 窗格中，单击“打开或关闭元素视图”(Open or close element view)。

5.16.4 使用项目库

5.16.4.1 项目库基本知识

功能

在项目库中，可以存储想要在项目中多次使用的元素。项目库随项目自动生成和保存。

参见

库基本知识 (页 161)

5.16.4.2 在项目库中创建文件夹

可以使用文件夹和子文件夹分别组织项目库的元素以满足需要。

要求

已显示“库”(Libraries) 任务卡。

步骤

要在项目库中创建新文件夹，请按以下步骤操作：

1. 右键单击项目库或现有文件夹。
2. 从快捷菜单中选择“添加文件夹”(Add folder)。
将创建一个新文件夹。
3. 输入新文件夹的名称。

5.16.4.3 将元素添加到项目库

先决条件

已显示“库”(Libraries) 任务卡。

步骤

要将新元素添加到项目库，请按以下步骤操作：

1. 选择要添加到项目库的元素。
2. 将所选元素拖到“库”(Libraries) 任务卡“项目库”(Project library) 选项板中项目库的书形图标上。在鼠标指针下方出现小加号时，释放鼠标左键。

5.16.4.4 使用项目库的元素

要求

已显示“库”(Libraries) 任务卡。

步骤

要在项目中使用项目库的元素，请按以下步骤操作：

1. 打开项目库，以便可以看到该库的元素。
2. 将元素从“项目库”(Project library) 窗格拖到想要使用该元素的位置。如果该位置不允许插入，则鼠标指针将变为带有斜线的圆。

如果该位置已有同名的元素，则会打开“粘贴”(Paste) 对话框。在该对话框中，可决定是替换现有元素还是以不同的名称插入项目库的元素。

注意

并非对所有元素都会显示“粘贴”(Paste) 对话框。对哪些元素显示该对话框取决于所安装的产品。

或者：

1. 打开元素视图。
2. 将元素从“元素”(Elements) 窗格拖到想要使用该元素的位置。如果该位置不允许插入，则鼠标指针将变为带有斜线的圆。

如果该位置已有同名的元素，则会打开“粘贴”(Paste) 对话框。在该对话框中，可决定是替换现有元素还是以不同的名称插入项目库的元素。

注意

并非对所有元素都会显示“粘贴”(Paste) 对话框。对哪些元素显示该对话框取决于所安装的产品。

5.16.4.5 编辑项目库的元素

可以对库元素使用以下编辑命令：

- 复制
- 剪切
- 粘贴
- 在库内移动
- 重命名

- 删除

始终可以通过键盘 (页 119)、菜单和快捷菜单来执行其中的每个命令。

要求

已显示“库”(Libraries) 任务卡。

复制元素

要复制库元素，请按以下步骤操作：

1. 右键单击要复制的库元素。
2. 在快捷菜单中选择“复制”(Copy) 命令。

剪切元素

要剪切库元素，请按以下步骤操作：

1. 右键单击要剪切的库元素。
2. 在快捷菜单中选择“剪切”(Cut) 命令。

粘贴元素

要粘贴库元素，请按以下步骤操作：

1. 复制或剪切一个库元素。
2. 右键单击要在其中粘贴该元素的库。
3. 在快捷菜单中选择“粘贴”(Paste) 命令。

移动元素

要在库内移动库元素，请按以下步骤操作：

1. 选择要移动的库元素。
2. 将库元素拖到要插入该元素的库。

注意

将元素从一个库移动到另一个库时，将复制该元素而不是移动。

重命名元素

要重命名库元素，请按以下步骤操作：

1. 右键单击要重命名的元素。
2. 在快捷菜单中选择“重命名”(Rename) 命令。
3. 输入新名称。

删除元素

要删除库元素，请按以下步骤操作：

1. 右键单击要删除的库元素。
2. 在快捷菜单中选择“删除”(Delete) 命令。

5.16.4.6 从项目库中删除元素

先决条件

已显示“库”(Libraries) 任务卡。

步骤

要从项目库中删除元素，请按以下步骤操作：

1. 在“项目库”(Project library) 选项板中将项目库最大化，以便可以查看该库的元素。
2. 右键单击要删除的元素。
3. 在上下文菜单中选择“删除”(Delete) 命令。

或者：

1. 打开元素视图。
2. 在“元素”(Elements) 选项板中右键单击要删除的元素。
3. 在上下文菜单中选择“删除”(Delete) 命令。

5.16.4.7 过滤视图

要使大量的库看起来更简单直观，可以使用过滤器选项限制显示。

要求

已显示“库”(Libraries) 任务卡。

步骤

要过滤视图，请按以下步骤操作：

1. 打开“项目库”(Project library) 窗格或“全局库”(Global libraries) 窗格。
2. 在下拉列表中，选择要显示其库元素的对象类型。

结果

仅显示可用于该对象类型的库元素。可以随时将过滤器设置为“全部”(All) 恢复未过滤的视图。

5.16.5 使用全局库

5.16.5.1 全局库基本知识

功能

如果想要在其它项目中使用某些元素，可以将这些元素存储在全局库中。必须显式创建全局库。

共享使用全局库

可以与其他用户一起使用全局库。但是要求所有要访问该全局库的用户以只读方式打开该库。

参见

库基本知识 (页 161)

5.16.5.2 创建新全局库

要求

已显示“库”(Libraries) 任务卡。

步骤

要创建新全局库，请按以下步骤操作：

1. 单击“全局库”(Global libraries) 窗格工具栏中的“创建新全局库”(Create new global library)，或选择菜单命令“选项 > 全局库 > 创建新库”(Options > Global libraries > Create new library)。

将打开“创建新全局库”(Create new global library) 对话框。

2. 指定新全局库的名称和存储位置。
3. 单击“创建”(Create) 确认输入。

结果

新全局库即生成并插入“全局库”(Global libraries) 窗格。在文件系统中该全局库的存储位置处创建了具有该全局库名称的文件夹。实际库文件具有文件扩展名“.al10”。

5.16.5.3 打开全局库

要求

已显示“库”(Libraries) 任务卡。

步骤

要打开全局库，请按以下步骤操作：

1. 单击“全局库”(Global libraries) 窗格工具栏中的“打开新全局库”(Open new global library)，或选择菜单命令“选项 > 全局库 > 打开库”(Options > Global libraries > Open library)。

将打开“打开全局库”(Open global library) 对话框。

2. 选择要打开的全局库。可通过文件扩展名“.al10”识别库文件。
3. 还可以写保护方式打开库。为此，请启用“打开全局库”(Open global library) 对话框中的“以只读方式打开”(Open as read-only) 选项。

注意

请注意以下事项：

- 如果以只读方式打开库，则无法在全局库中输入任何其它元素。
- 如果多个用户要访问库，则所有用户都必须以只读方式打开该库。这是共享访问库的要求。

4. 单击“打开”(Open)。

所选的全局库随即打开并插入“全局库”(Global libraries) 窗格。

5.16.5.4 保存全局库

可以随时保存对西门子提供的全局库所做的更改。可以使用“另存为”(Save as) 命令以另一个名称保存全局库。

要求

- 已显示“库”(Libraries) 任务卡。
- 全局库没有写保护。

保存更改

要保存全局库，请按以下步骤操作：

1. 右键单击要保存的全局库。
2. 在快捷菜单中选择“保存库”(Save library) 命令。

以其它名称保存全局库

要以其它名称保存全局库，请按以下步骤操作：

1. 右键单击要重命名的全局库。
2. 在快捷菜单中选择“将库另存为”(Save library as) 命令。
将打开“将全局库另存为”(Save global library as) 对话框。
3. 选择存储位置并输入文件名。
4. 单击“保存”(Save) 确认输入。

5.16.5.5 关闭全局库

全局库与项目无关。这意味着全局库不会随项目一起关闭。因此，必须显式关闭全局库。

要求

已显示“库”(Libraries) 任务卡。

步骤

要关闭全局库，请按以下步骤操作：

1. 右键单击要关闭的全局库。
2. 在快捷菜单中选择“关闭库”(Close library) 命令。
3. 如果对全局库进行过更改，将打开一个对话框，在其中可以选择是否要保存对全局库所做的更改。根据是否要保存更改，单击“是”(Yes) 或“否”(No)。
全局库即会关闭。

5.16.5.6 删除全局库

在 Windows 资源管理器中删除全局库。

要求

所有用户均已关闭全局库。

步骤

要删除全局库，请按以下步骤操作：

1. 打开 Windows 资源管理器。
2. 右键单击库目录。
3. 在快捷菜单中选择“删除”(Delete) 命令。
4. 单击“是”(Yes) 确认安全提示。

5.16.5.7 在全局库中创建文件夹

可以使用文件夹和子文件夹组织自定义全局库的元素以满足需要。

要求

- 已显示“库”(Libraries) 任务卡。
- 全局库没有写保护。

步骤

要在全局库中创建新文件夹，请按以下步骤操作：

1. 右键单击全局库或现有文件夹。

2. 从快捷菜单中选择“添加文件夹”(Add folder)。
将创建一个新文件夹。
3. 输入新文件夹的名称。

5.16.5.8 将元素添加到全局库

先决条件

- 已显示“库”(Libraries) 任务卡。
- 全局库没有写保护。

步骤

要将新元素添加到全局库，请按以下步骤操作：

1. 选择要添加到库的元素。
2. 将在“库”(Libraries) 任务卡“全局库”(Global libraries) 选项板中所选的元素拖到要添加该元素的库的书形图标上。在鼠标指针下方出现小加号时，释放鼠标左键。

5.16.5.9 使用全局库的元素

要求

已显示“库”(Libraries) 任务卡。

步骤

要在项目中使用全局库的元素，请按以下步骤操作：

1. 最大化全局库，以便可以看到该库的元素。
2. 将元素从“全局库”(Global libraries) 窗格拖到想要使用该元素的位置，例如，拖到项目树中。如果该位置不允许插入，则鼠标指针将变为带有斜线的圆。

如果该位置已有同名的元素，则会打开“粘贴”(Paste) 对话框。在该对话框中，可决定是替换现有元素还是以重复的标识 (名称_编号) 插入项目库的元素。

注意

并非对所有元素都会显示“粘贴”(Paste) 对话框。对哪些元素显示该对话框取决于所安装的产品。

或者：

1. 打开元素视图。
2. 将元素从“元素”(Elements) 窗格或“部件”(Parts) 窗格拖到想要使用该元素的位置，例如，拖到项目树中。如果该位置不允许插入，则鼠标指针将变为带有斜线的圆。

如果该位置已有同名的元素，则会打开“粘贴”(Paste)对话框。在该对话框中，可决定是替换现有元素还是以重复的标识(名称_编号)插入项目库的元素。

注意

并非对所有元素都会显示“粘贴”(Paste)对话框。对哪些元素显示该对话框取决于所安装的产品。

5.16.5.1 编辑全局库的元素

0

可以对库元素使用以下编辑命令：

- 复制
- 剪切
- 粘贴
- 在库内移动
- 重命名
- 删除

始终可以通过键盘 (页 119)、菜单和快捷菜单来执行其中的每个命令。

要求

- 已显示“库”(Libraries)任务卡。
- 全局库没有写保护。

复制元素

要复制库元素，请按以下步骤操作：

1. 右键单击要复制的库元素。
2. 在快捷菜单中选择“复制”(Copy)命令。

剪切元素

要剪切库元素，请按以下步骤操作：

1. 右键单击要剪切的库元素。
2. 在快捷菜单中选择“剪切”(Cut)命令。

粘贴元素

要粘贴库元素，请按以下步骤操作：

1. 复制或剪切一个库元素。
2. 右键单击要在其中粘贴该元素的库。

3. 在快捷菜单中选择“粘贴”(Paste) 命令。

移动元素

要在库内移动库元素，请按以下步骤操作：

1. 选择要移动的库元素。
2. 将库元素拖到要插入该元素的库。

注意

将元素从一个库移动到另一个库时，将复制该元素而不是移动。

重命名元素

要重命名库元素，请按以下步骤操作：

1. 右键单击要重命名的元素。
2. 在快捷菜单中选择“重命名”(Rename) 命令。
3. 输入新名称。

删除元素

要删除库元素，请按以下步骤操作：

1. 右键单击要删除的库元素。
2. 在快捷菜单中选择“删除”(Delete) 命令。

5.16.5.1 从全局库中删除元素

1

先决条件

- 已显示“库”(Libraries) 任务卡。
- 全局库没有写保护。

步骤

要从全局库中删除元素，请按以下步骤操作：

1. 在“全局库”(Global library) 选项板中将全局库最大化，以便可以查看该库的元素。
2. 右键单击要删除的元素。
3. 在上下文菜单中选择“删除”(Delete) 命令。

或者：

1. 打开元素视图。
2. 在“元素”(Elements) 选项板中右键单击要删除的元素。
3. 在上下文菜单中选择“删除”(Delete) 命令。

5.16.5.1 使用所提供的全局库

2

随系统提供的全局库取决于所安装的产品。

要求

已显示“库”(Libraries) 任务卡。

步骤

要在项目中使用所提供的全局库中的元素，请按以下步骤操作：

1. 打开相关库，以便可以看到该库的元素。
2. 将元素从“全局库”(Global libraries) 窗格拖到想要使用该元素的位置，例如，拖到项目树中。如果该位置不允许插入，则鼠标指针将变为带有斜线的圆。

或者：

1. 打开元素视图。
2. 将元素从“部件”(Parts) 窗格拖到想要使用该元素的位置，例如，拖到项目树中。如果该位置不允许插入，则鼠标指针将变为带有斜线的圆。

5.16.5.1 过滤视图

3

要使大量的库看起来更简单直观，可以使用过滤器选项限制显示。

要求

已显示“库”(Libraries) 任务卡。

步骤

要过滤视图，请按以下步骤操作：

1. 打开“项目库”(Project library) 窗格或“全局库”(Global libraries) 窗格。
2. 在下拉列表中，选择要显示其库元素的对象类型。

结果

仅显示可用于该对象类型的库元素。可以随时将过滤器设置为“全部”(All) 恢复未过滤的视图。

5.17 保护项目数据

5.17.1 项目数据的保护原理

简介

可以防止未经授权用户访问项目数据。 例如，这包括：

- 访问设备时的限制
- 块的复制和显示保护

5.18 使用交叉引用

5.18.1 使用交叉引用

交叉引用简介

交叉引用列表提供项目中对象的使用概况。 可以看到哪些对象相互依赖以及各对象所在的位置。 因此，交叉引用是项目文档的一部分。

您还可以直接跳到对象的使用位置。

可以在交叉引用列表中显示和定位的对象取决于所安装的产品。

组态设备和网络

6.1 硬件和网络编辑器

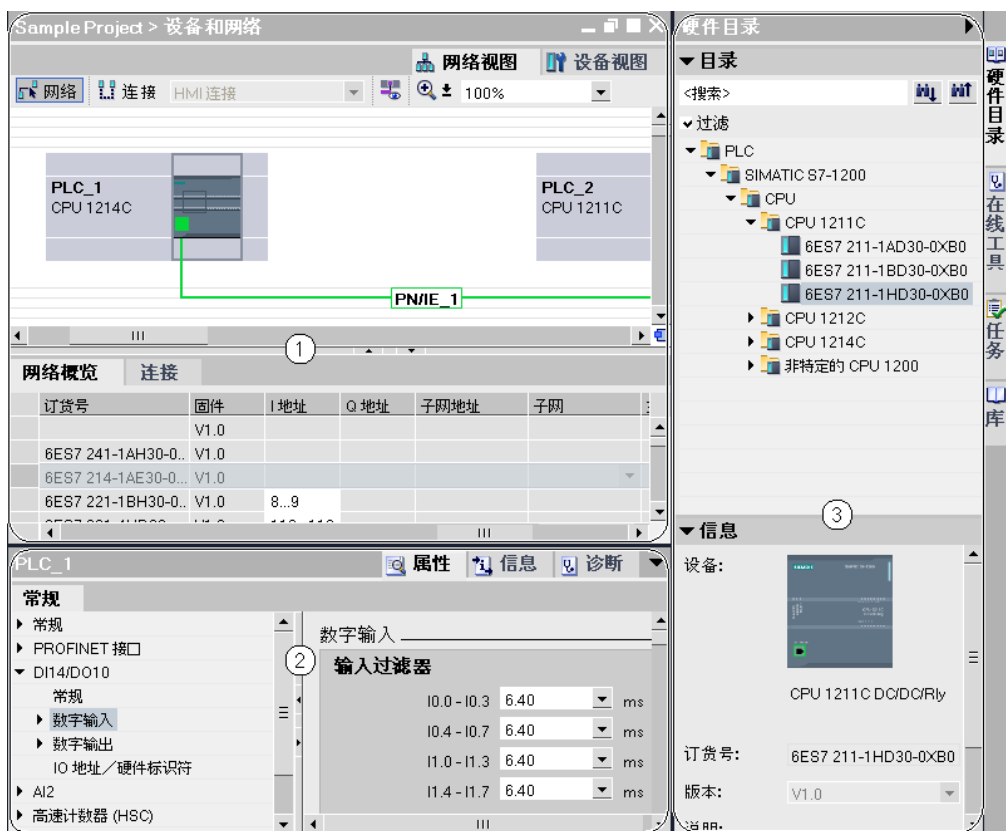
6.1.1 硬件和网络编辑器概述

硬件和网络编辑器的功能

硬件和网络编辑器是一个集成开发环境，用于对设备和模块进行组态、联网和参数分配。该编辑器为自动化项目的实现提供最多的帮助。

硬件和网络编辑器的结构

硬件和网络编辑器由以下部分组成：



- ① 设备视图 (页 180) 和/或网络视图 (页 178)
- ② 巡视窗口 (页 183)
- ③ 硬件目录 (页 184)

硬件和网络编辑器包含设备视图和网络视图。可以根据是要生成和编辑单个设备和模块还是整个网络和设备组态，随时在这两个视图间切换。

巡视窗口包含当前所标记对象的相关信息。在此处可更改所标记对象的设置。

将自动化系统所需的设备和模块从硬件目录拖到网络视图或设备视图中。

6.1.2 网络视图

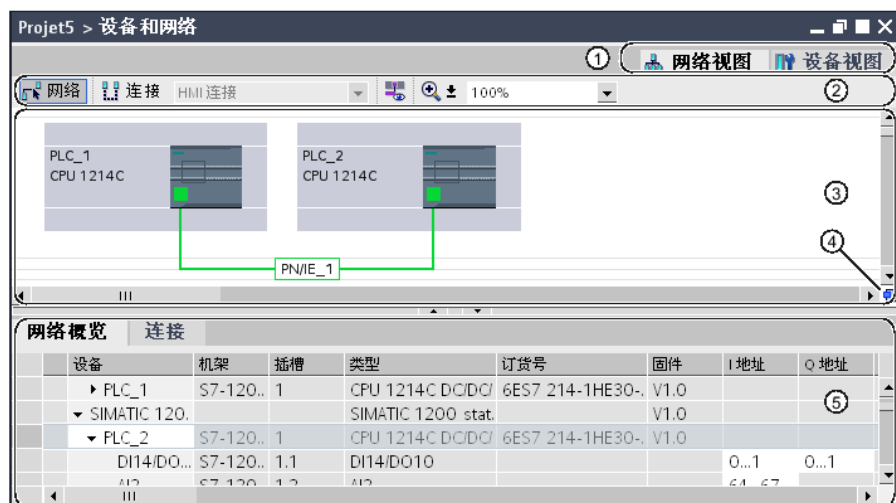
简介

网络视图是网络编辑器的工作区域。在此处可执行以下任务：

- 组态和分配设备参数
- 设备间组网

结构

下图显示了网络视图的组成部分：



- ① 选择网络视图/设备视图
- ② 网络视图的工具栏
- ③ 设备视图的图形区域
- ④ 总览导航
- ⑤ 设备视图的表格区域

可以使用鼠标更改设备视图图形区域与表格区域之间的间距。要执行此操作，请单击表格视图的顶边，通过按住鼠标按钮并移动鼠标来扩大或缩小该区域。可以使用两个小箭头键执行最小化、最大化操作或通过一次单击来选择表格视图的最新表格结构。

工具栏

使用工具栏可从设备的联网或连接 (页 204) 中进行以下选择：

- 网络
- 连接和连接类型

使用“显示地址”符号在网络视图的图形区域显示接口地址。

符号	含义
	显示地址

使用缩放功能可以更改图形区域中的显示。

图形区域

网络视图的图形区域显示所有与网络相关的设备、网络、连接和关系。在该区域中，可以添加硬件目录中的设备 (页 195)，通过其接口使其彼此相连以及组态通信设置。

总览导航

单击总览导航可在图形区域总览所创建的对象。按住鼠标按钮，可以快速导航到所需的对象并在图形区域中显示它们。

表格区域

网络视图的表格区域包括与当前设备、连接和通信设置对应的各种表格：

- 网络概览
- 连接

参见

用户界面的布局 (页 103)

调整用户界面 (页 117)

确定在线状态并用符号显示 (页 258)

6.1.3 设备视图

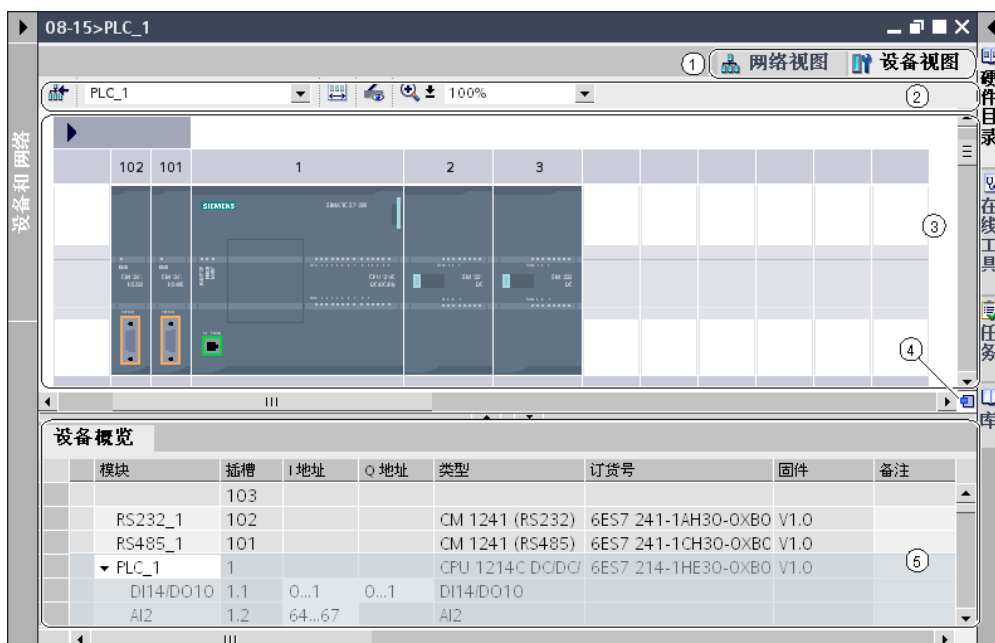
简介

设备视图是硬件编辑器的工作区域。在此处可执行以下任务：

- 组态和分配设备参数
- 组态和分配模块参数

结构

下图显示了设备视图的组成部分：



- ① 选择网络视图/设备视图
- ② 设备视图的工具栏
- ③ 网络视图的图形区域
- ④ 总览导航
- ⑤ 网络视图的表格区域

可以使用鼠标更改网络视图图形区域与表格区域之间的间距。要执行此操作，请单击表格视图的顶边，通过按住鼠标按钮并移动鼠标来扩大或缩小该区域。可以使用两个小箭头键执行最小化、最大化操作或通过一次单击来选择表格视图的最新表格结构。

工具栏

在设备视图中，通过从下列列表中选择相应的设备在不同设备间进行切换。

还有其它用于显示各种信息的符号：

符号	含义
	转到网络视图 (页 178)
	未分配模块的剪贴板 (页 193)
	显示模块标题 (页 191)

使用缩放功能可以更改图形区域中的显示。

图形区域

设备视图的图形区域显示设备与相关模块，它们彼此间通过一个或多个机架来分配给对方。在这里，可以将硬件目录中的其它硬件对象安装到机架上的插槽中 (页 196)。

总览导航

单击总览导航可在图形区域总览所创建的对象。按住鼠标按钮，可以快速导航到所需的对象并在图形区域中显示它们。

表格区域

通过设备视图的表格区域可总览所用的模块以及最重要的技术数据和组织数据 (页 192)。

参见

- 用户界面的布局 (页 103)*
- 调整用户界面 (页 117)*
- 确定在线状态并用符号显示 (页 258)*

6.1.4 打印硬件组态和网络组态

硬件组态和网络组态的打印输出

用户可将以下硬件视图和网络视图元素作为项目文档的一部分进行打印输出：

- 图形网络视图
- 网络总览表格
- 图形设备视图
- 设备总览表格
- 编辑器中当前选定对象的参数

打印输出的范围

启动打印时，总是打印当前打开的编辑器。默认情况下，总是打印与该编辑器相关的图形表示以及表格。用户可以调整打印输出的范围。相关操作说明，请参见“更改打印选项 (页 183)”。

以文本形式同时打印输出所选对象的参数及其当前值设置。如果选择了某个模块，则还会打印相关模块的参数。例如，如果选择了一个 CPU，则还会打印所插入信号板 (如果存在) 的参数。

打印输出的形式

图形总是以当前所选缩放设置打印。如果图形或表格不适合所选纸张格式，将分多页进行打印输出。

参见

- 更改打印选项 (页 183)*
- 文档设置 (页 145)*

创建打印预览 (页 145)

打印项目数据 (页 146)

6.1.5 更改打印选项

更改打印输出的范围

从设备或网络视图打印时，可以指定是同时打印图形和表格还是仅打印其中一项。默认情况下，两项都打印。

步骤

要更改打印输出的范围，请按以下步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
2. 在区域导航中打开“硬件配置”(Hardware configuration) 组。
3. 根据是否也要打印网络和设备视图的图形，选中或取消选中“激活的图形视图”(Active graphic view) 复选框。
4. 根据是否也要打印设备和网络总览表格，选中或取消选中“激活的表格”(Active table) 复选框。

参见

打印硬件组态和网络组态 (页 182)

6.1.6 巡视窗口

在巡视窗口中可以编辑所选对象的下述属性和参数。

结构

巡视窗口由下列部分组成：



- ① 在各种信息和工作区域间切换
- ② 在各种信息和参数间浏览
- ③ 显示所选信息和参数的显示区域

功能

巡视窗口中的信息和参数分为不同类型的信息：

- 属性
- 信息
- 诊断

要显示相应的信息和参数，请在需要的区域中单击。“属性”(Properties) 区域是组态自动化系统的最重要的区域。默认情况下显示该区域。

巡视窗口的左窗格用于区域导航。信息和参数分组安排在此处。单击组名称左侧的箭头符号可以展开该组，前提是它有子组。如果选择一个组或子组，则相应的信息和参数将在巡视窗口的右窗格中显示，并且也可在此对其进行编辑。

参见

- 编辑属性和参数 (页 202)*
- 硬件和网络编辑器概述 (页 177)*

6.1.7 硬件目录

使用“硬件目录”(Hardware catalog) 任务卡可以很容易地访问各式各样的硬件组件。

结构

“硬件目录”(Hardware catalog) 任务卡由以下窗格组成：



- ① “目录”(Catalog) 窗格、搜索和过滤功能
- ② “目录”(Catalog) 选项板，组件选项
- ③ “信息”(Information) 窗格

功能

- 使用“目录”(Catalog) 窗格、搜索和过滤功能可方便您搜索具体的硬件组件。如果激活过滤功能，则仅会显示适合当前环境的对象。可以在当前环境中使用的对象包括网络视图中可互连的对象或仅与设备视图中的设备相兼容的模块等。
- “目录”(Catalog) 窗格在树形结构中提供各种硬件组件。可以将所需的设备或模块从目录中移动到设备视图或网络视图的图形工作区域。
- “信息”(Information) 窗格包含目录中所选对象的详细信息：
 - 简图表示
 - 名称

6.1 硬件和网络编辑器

- 版本号
- 订货号
- 简要描述

参见

浏览硬件目录 (页 190)
硬件和网络编辑器概述 (页 177)
有关硬件组件的信息 (页 186)

6.1.8 有关硬件组件的信息

在硬件目录中，可以在“信息”(Information) 窗格中显示有关所选硬件组件的信息。还可以使用快捷菜单显示有关所选硬件组件的详细信息。

访问详细信息

如果在硬件目录中选择一个硬件对象且打开了快捷菜单，则不但可以使用“复制”(Copy) 功能，还可以使用用于访问服务和支持相关信息的三个选项：

- 产品支持
- 常见问题解答
- 手册

所需的信息显示在硬件和网络编辑器的工作区中。

注意

只有在连接到 Internet 且启用该功能后，才能访问服务和支持。默认情况下会禁用该功能。要启用该功能，请参见“启用产品支持 (页 187)”中的说明。

有关产品支持的信息

可以在此处访问硬件和软件组件的常规信息。所选硬件对象的订货号自动输入到搜索屏蔽中。但是，您还可以搜索其它硬件和软件组件。

常见问题解答

可以在此处访问常见问题解答 (FAQ, Frequently Asked Question)。您可以查看各种有关硬件和软件的问题。使用详细的搜索屏蔽，可以过滤所需的主题。

手册

可以在此处访问各种硬件组件的手册。如果在您规划组态、寻址或参数分配时需要所用硬件的更详细知识，该选项会特别有用。

参见

硬件目录 (页 184)
启用产品支持 (页 187)

6.1.9 启用产品支持**启用服务和支持功能**

对于硬件目录中的每个设备，用户可以选择显示在 Siemens 网站的服务和支持中存储的附加信息。默认情况下会禁用该功能。以下介绍了如何启用该功能。

要求

软件必须能够访问 Internet。

步骤

要启用服务和支持功能，请按以下步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
2. 在区域导航中打开“硬件配置”(Hardware configuration) 组。
3. 选中“通过 Internet”(Via Internet) 复选框。

结果

现在可以从硬件目录通过相应模块的快捷菜单来访问产品支持、FAQ 和手册。

参见

有关硬件组件的信息 (页 186)

6.1.10 硬件和网络编辑器中的键盘操作

可以在硬件和网络编辑器中通过组合使用键盘和鼠标来直接执行网络和设备视图的某些功能。下表中的键盘操作 (页 119) 对应于标准行为。在此处可以找到网络和设备视图的图形工作区域的键盘操作。

常规键盘操作

功能	键组合	注释
放大框架中的视图	<Ctrl+空格> + 按下鼠标按钮	不会选择对象
移动视图	<空格> + 按下鼠标按钮	鼠标不可定位在对象上
取消当前操作	<Esc>	-
分开连接	<Esc> 或双击	进行连接时

所选对象

功能	键组合	注释
选择对象	单击鼠标	-
移动对象	<Ctrl+X> , 然后是 <Ctrl+V>	复制所选的要移动的对象, 然后将其粘贴到新位置
复制对象	<Ctrl+C>	将所选对象复制到剪贴板
粘贴对象	<Ctrl+V>	将对象从剪贴板粘贴到选定位置
删除所选对象		-
选择多个对象 1	<Shift> + 单击	单独地选择多个对象
选择多个对象 2	<Ctrl> + 单击	也可取消选择所选对象
移动选择内容	按下鼠标按钮	移动到允许的插槽
复制选择内容	<Ctrl> + 按下鼠标按钮	移动到允许的插槽

6.2 组态设备

6.2.1 基本知识

6.2.1.1 组态设备简介

设置自动化系统需要对各硬件组件进行组态、分配参数和互连。在设备和网络视图中执行这些操作。

组态

“组态”意指在设备或网络视图对设备和模块进行布置、设置和联网。机架用符号表示。和“实际的”机架一样，可以在其上插入规定数量的模块。

将给各模块自动分配一个地址。如果 CPU 允许自由分配地址，则可更改模块地址。

自动化系统启动时，CPU 会比较软件所生成的设定组态与系统的实际组态。如此将检测可能的错误并直接进行报告。

分配参数

“分配参数”意指设置所用组件的属性。将为硬件组件和用于交换数据的设置分配：

- 具有可选参数的模块的属性

- 组件间数据交换的设置

这些参数将加载到 CPU 并在 CPU 启动期间传送给相应的模块。因为设置的参数在启动期间自动加载到新的模块，所以可轻松地更换模块。

需要配置硬件

如果想要设置、扩展或更改自动化项目，则需要配置硬件。为此，需要向结构中添加其它硬件组件、将它们与现有的组件相连并根据任务要求修改硬件属性。

自动化系统和模块的属性是预设的，从而在很多情况下，都无须再次对其进行参数化。但在下列情况下需要进行参数分配：

- 您想要更改模块的默认参数设置。
- 您想要使用特殊功能。
- 您想要组态通信连接。

参见

更改模块的属性 (页 231)

6.2.1.2 常规插槽规则

简介

对每个自动化系统和模块将应用特定的插槽规则。

如果在设备视图中从硬件目录选择模块，将标记机架中所有可用于该模块的插槽。只能将模块拖动到标记的插槽中。

插入、移动或交换模块时，该插槽规则仍然适用。

一致性

某些插槽规则取决于环境的组态情况。即有时可以将模块插入机架，即使这样做会导致目前不一致。如果更改了组态，例如，选择了其它模块或模块参数设置，您仍然可以再次使组态一致。

如果插入模块导致的不一致可更正，则允许执行该插入操作。在传送组态时将执行一致性检查。不一致情况将以报警形式显示在巡视窗口的“信息”(Info) 下。可以根据一致性检查结果修改组态，从而使其一致。

布置模块的常规规则

以下是一些适用于机架中模块的经验法则：

- 只能插到某些插槽中
- 需根据其它模块、CPU 或设置进行插入
- 在机架中的使用次数有限制

6.2.1.3 浏览硬件目录

简介

选择要用于硬件配置的硬件组件时，请使用“硬件目录”(Hardware catalog) 任务卡。硬件目录用于在网络视图中选择可互连的硬件组件以及在设备视图中选择所需的模块。

环境过滤器

硬件目录中有过滤功能。如果禁用过滤功能，则硬件目录中将始终显示目录提供的所有对象。

如果要只显示可在当前环境中使用的对象，则请激活“过滤器”(Filter) 复选框。

如果已激活该过滤器，将仅显示以下对象：

- 在网络视图中只显示可联网的对象。
- 设备视图中显示作为当前设备环境一部分的所有模块。

如果在网络和设备视图间切换，过滤对象的视图会根据当前的环境进行调整。

搜索选项



可以使用搜索功能在硬件目录中搜索特定条目。输入搜索术语时请注意下列规则：

- 不区分大小写文本。
- 根据硬件目录中的显示形式输入空格和/或制表符或连字符。
- 搜索功能会考虑搜索术语的各个部分。

注意

请注意订货号中空格与连字符的区别。

从硬件目录中高亮显示的对象开始搜索，并且可向上或向下进行搜索。

符号	含义
	向下搜索
	向上搜索

浏览硬件目录

如果想浏览硬件目录，请按以下步骤操作：

1. 在搜索功能的输入域中单击。
2. 输入搜索术语。搜索会包括以下元素：
 - 设备或模块的名称
 - 订货号
 - 信息文本

- 单击“向下搜索”(Downwards search) 或“向上搜索”(Upwards search) 按钮。

注意

为了确保搜索方向正确，请注意已在硬件目录中标记了哪个位置。若要浏览整个目录，请单击硬件目录最顶层的对象并在输入搜索术语后单击“向下搜索”(Downwards search) 开始搜索。

所找到的第一个与搜索术语匹配的条目作为结果显示。要获得更多搜索结果，可再次单击“向下搜索”(Downwards search) 或“向上搜索”(Upwards search) 按钮。

参见

[硬件目录 \(页 184\)](#)

[有关硬件组件的信息 \(页 186\)](#)

6.2.1.4 使用机架

简介

要将模块分配给设备，需要使用机架，即安装导轨。将模块固定到机架上并通过背板总线将其与 CPU、电源或其它模块连接。


创建机架

如果在网络视图中插入设备，则会自动创建适合所选设备的一个站和机架。机架和可用的插槽显示在设备视图中。可用的插槽数也取决于所使用设备的类型。

机架结构

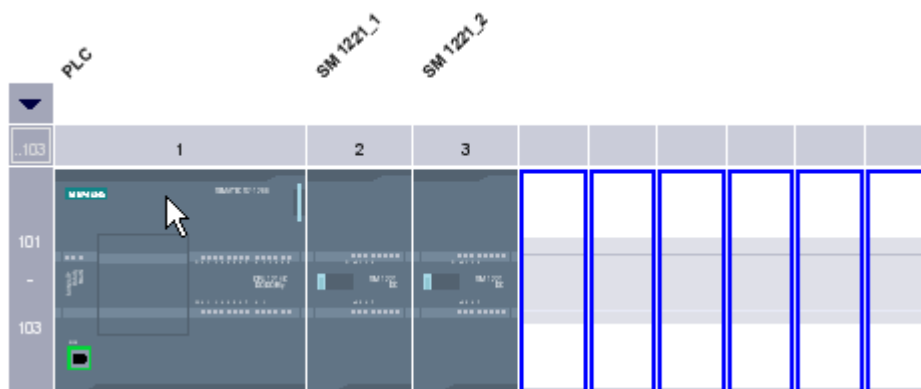
机架始终包含在网络视图中插入的设备。该设备将被永久分配一个插槽，具体插槽取决于相应设备的类型。设备的右侧还会有其它插槽，必要时在设备的左侧也有；插槽编号位于插有设备的插槽的上方。

相应的简短描述显示在所插入设备和模块的上方。使用“视图”(View) 下菜单栏命令“显示模块标题”(Display module titles) 或设备视图工具栏 (页 180) 中的相应符号可显示或隐藏该简短描述。

符号	含义
	显示模块标题

在硬件目录中选择模块时，将标记所有允许插入这些模块的插槽。这样，用户便可立即看到可插入所选模块的插槽。

在下面的截屏中，已在硬件目录中为一个部分装配的机架选择了一个信号模块：



因为插槽 101-103 保留给通信模块，所以仅其它空闲插槽显示为可用插槽。

可以使用可扩展插槽上方的箭头符号展开和折叠前侧插槽组。插槽组折叠后，将显示该组中第一个和最后一个插槽号。

下图显示了展开后的插槽组：



无法折叠已插入模块的插槽组。

选择多个模块和插槽

有多种选择多个模块或插槽的方法：

- 按 <Shift> 或 <Ctrl>，可以同时选择多个模块或插槽。
- 在机架外单击，然后按住鼠标按钮并拖出一个框来框住想要选择的模块或插槽。

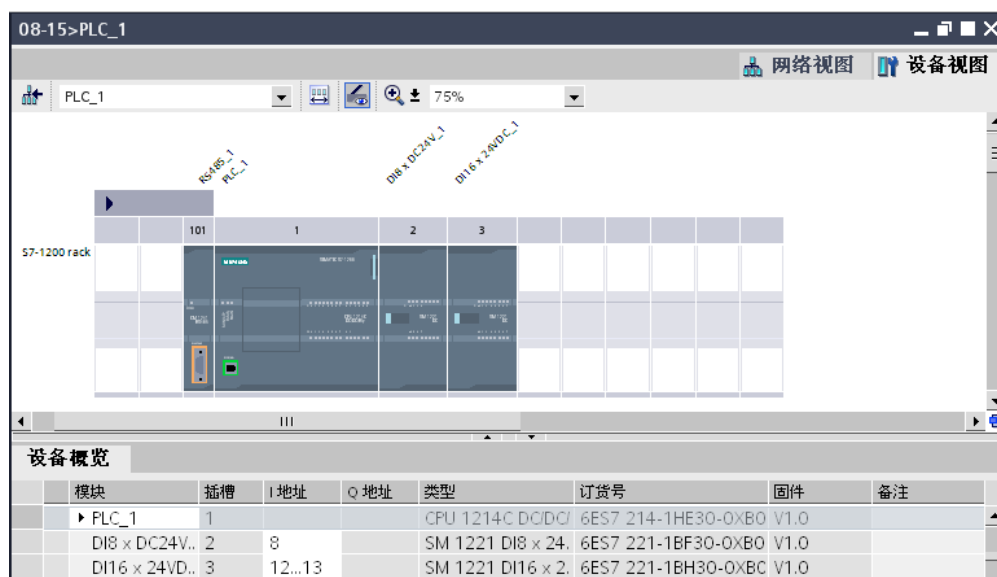
6.2.1.5 设备概览中的对象

设备概览位于设备视图的下半部分。设备概览是一个表格，其中显示机架中所插入模块的重要信息。

设备概览的结构和内容

设备概览中以表格形式显示机架的分配情况。表格中的各行包含分配插槽的信息。

下面的截屏显示了具有 SIMATIC S7-1200 CPU 组态的设备视图。



在上半部分中，可看到图形视图，该视图显示了各种模块在机架的插槽 1 到 3 和 101 中的分配情况。在下半部分中，可看到设备概览中该机架的表格表示法。

设备概览中的每一行代表一个插槽。每个插槽的重要信息在各列中进行显示：

列	含义
模块	模块名称，可随意编辑
插槽	插槽号
I 地址	输入地址区，可随意编辑
Q 地址	输出地址区，可随意编辑
类型	模块的目录名称
订货号	模块订货号
固件	模块固件版本
注释	可选的注释

参见

设备视图 (页 180)

6.2.1.6 未插入模块区域

在某些情况下，短期内未给用于硬件组态的模块分配插槽。这些未插入的模块将移动到“未插入模块区域”(设备视图中的特殊区域)。

将模块添加到存储区域

例如，对于要通过复制操作分配给设备的模块，如果相应的机架没有可用于该模块的空闲可用兼容插槽，则该模块会被自动移动到未插入模块区域中。

在以下情况下，模块将被添加到未插入模块区域：

- 在网络视图中，将模块移动到设备但机架没有兼容的空闲插槽。
- 在设备视图中，将模块从机架、硬件目录或项目树直接移动到该存储区域。

使用未插入模块区域

使用相应的按钮可打开未插入模块区域。

可在设备视图找到“未插入模块区域”。



使用设备视图工具栏 (页 180) 中的相应符号打开未插入模块区域。

符号	含义
	打开未插入模块区域

注意

要释放插槽，请将模块从组态中移动到存储区域，然后将所需的模块从存储区域插入到释放的插槽中。

可以使用该方法将已经参数化的模块临时从组态中移出，而不删除它们。

存储区域中的模块处理

下列规则适用于存储区域的模块：

- 模块出现在项目树中相应设备下的“本地模块”(Local modules) 文件夹中。
- 这些模块将保留先前提供的所有设置和参数。
- 下载到目标系统时不会考虑这些模块，因此对于未插入模块区域中的模块不会执行一致性检查。

6.2.2 组态各个设备

6.2.2.1 将设备添加到硬件配置中

简介

在网络视图中有多种方法可将 CPU 添加到硬件配置中：

- 项目树中的命令“添加新设备”(Add new device)
- 在硬件目录中双击设备
- 在网络视图中从硬件目录中拖放
- 通过网络视图菜单栏中的命令“添加”(Add) >“设备”(Device)
- 使用硬件目录中相应设备的快捷菜单进行复制和粘贴

合适的机架将随新设备一起创建。所选设备插在机架上第一个允许的插槽中。

无论选择哪种方法，添加的设备在项目树以及硬件和网络编辑器的网络视图中都可见。

通过项目树添加设备

要通过项目树将设备添加到硬件配置中，请按以下步骤操作：

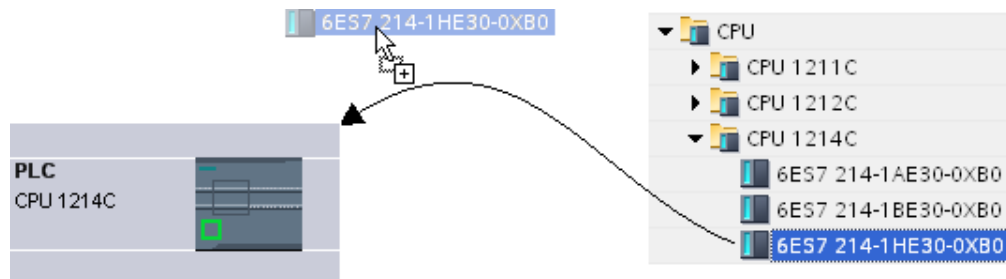
1. 在项目树中单击命令“添加新设备”(Add new device)。将打开“添加新设备”(Add new device) 对话框。
2. 在树结构中显示所需设备：
 - 在树结构中转到所需设备。
 - 在输入域中输入设备名称。
3. 从树中选择所需设备。有关当前所选设备的更多信息显示在该对话框的右侧。
4. 必要时，通过对话框中的下拉列表设置固件版本。
5. 如果要在添加设备后立即切换到设备视图，则选中“打开设备视图”(Open device view) 复选框。在这里可以立即继续进行设备组态和装配机架。
6. 单击“确定”(OK) 添加所选设备。该对话框关闭。

从硬件目录中添加设备

要通过硬件目录将设备添加到硬件配置中，请按以下步骤操作：

1. 打开网络视图。
2. 打开硬件目录。
3. 转到硬件目录中所需的 CPU。
4. 单击所选的 CPU，将其选中。

5. 必要时，通过硬件目录中的下拉列表设置固件版本。
6. 将 CPU 拖到网络视图中。



现在已将该 CPU 置于网络视图中。同时还创建了一个“PLC”站。双击设备或站框架打开设备视图并查看新机架和插入的 CPU。接下来可以组态设备视图中的设备并给机架装配模块。

参见

网络视图 (页 178)

创建未指定的 CPU (页 275)

有关硬件组件的信息 (页 186)

6.2.2.2 将模块插入机架

简介

在网络视图将设备从硬件目录添加到组态后，即可向设备中添加模块。

在设备视图有多种将模块添加到机架的方法：

- 如果有可用的有效插槽，则在硬件目录中双击模块。
- 通过拖放操作将该模块从硬件目录移动到图形或表格区域内可用的有效插槽中。
- 选择硬件目录中相应模块的快捷菜单中的“复制”(Copy)，然后在图形或表格区域中可用的有效插槽上选择相应快捷菜单中的“粘贴”(Paste)。

要从网络视图访问设备视图，可在网络视图中双击某个设备或站，或者选择“设备视图”(Device view) 选项卡。设备视图包含机架中所选设备的图示。软件中机架的图示与实际结构一样，即，可以看到的插槽数与实际结构中存在的相同。

注意

也可在网络视图将模块移动到机架。本例中必须禁用硬件目录的过滤功能。模块将自动插入允许的空闲插槽中。如果没有可用的插槽，模块将移动到未插入模块区域 (页 193)。

装配机架

依据适用的插槽规则在机架上布置模块。

模块插入机架后，将自动检查地址区，以防止地址被重复分配两次。因此，每个模块在插入后只具有一个有效的地址范围。

要求

- 您处于设备视图中。
- 硬件目录已打开。

从硬件目录中添加模块

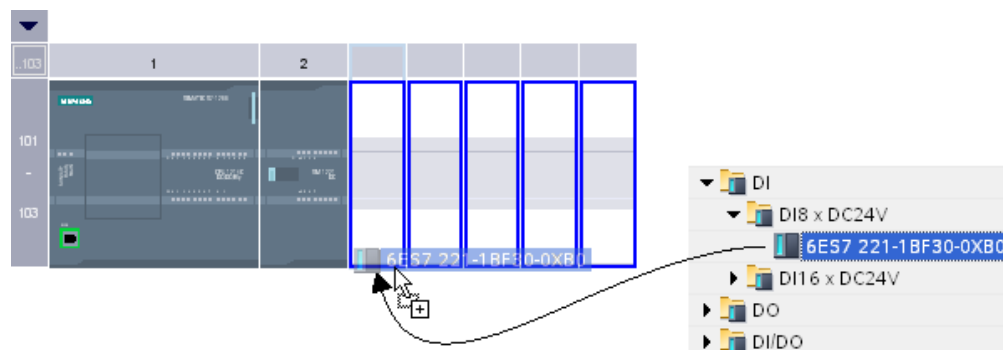
下面以信号模块为例说明如何将模块从硬件目录插入到机架。为此，请按下列步骤操作：

1. 转到硬件目录中所需的模块板卡。

注意

如果激活了硬件目录的过滤功能，将仅显示那些与所选设备类型相匹配的模块。

2. 选择所选的模块。
3. 必要时，通过硬件目录中的下拉列表设置固件版本。
4. 将该信号模块拖到机架上空闲的插槽上。

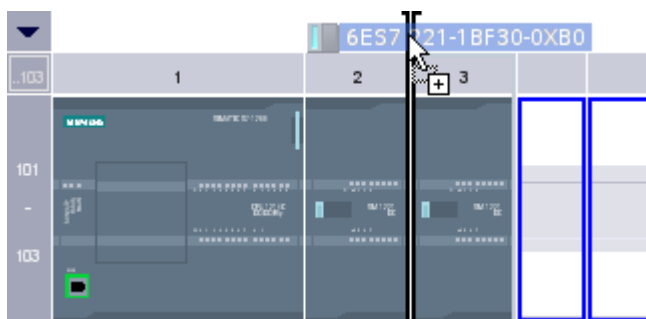


现在已完成将数字量信号模块插入到机架上的一个插槽中。对其它模块重复执行这些步骤。

模块名称显示在插入模块的上方。在菜单栏中使用“视图 > 显示模块标签”(View > Show module labels) 可以激活或禁用模块标签。

插入模块

还可以拖动该模块并将其插在已插好的两个模块之间。为此，可按住鼠标按钮将模块拖到停在两个现有模块之间。



将出现一个鼠标指针。释放鼠标按钮时，插在该指针右侧的所有模块将向右移动一个插槽位置。所有多余的模块将移动到未插入模块区域。新模块插在释放的插槽位置。

参见

- 设备视图 (页 180)
- 未插入模块区域 (页 193)
- 有关硬件组件的信息 (页 186)

6.2.2.3 删除硬件组件

可以在设备或网络视图中删除硬件组件。已删除的硬件组件将从系统中移除，并且为其分配的地址将再次可用。

规则

- 可以删除插在机架上和放置在未插入模块区域中的模块。
- 无法单独删除多个 CPU 和插有 CPU 的机架，只能在网络视图或项目树中与所有插入的硬件组件一起将其删除 (删除整个站)。

步骤

要删除硬件组件，请按以下步骤操作：

1. 选择要删除的硬件组件。
 - 设备视图：选择机架或者机架上或未插入模块区域中的模块。
 - 网络视图：从网络视图中选择站或相关硬件组件。
 2. 从快捷菜单中选择“删除”(Delete) 或按 。
 - 如果“删除”(Delete) 菜单项不可用，则说明所选内容中至少包含一个不能删除的组件。
- 这将删除所选的硬件组件。

注意

删除硬件组件可能会导致项目中出现不一致，即，违反插槽规则。一致性检查期间会报告不一致性。采取适当的措施纠正不一致，例如，确保遵守插槽规则。

参见

硬件和网络编辑器中的键盘操作 (页 187)

6.2.2.4 复制硬件组件

可在设备或网络视图中复制硬件组件。复制的硬件组件存储在剪贴板中，并且可以从该剪贴板将其粘贴到任意其它位置。复制的站将作为网络视图中的新站粘贴。复制的设备和模块可以粘贴到网络视图和设备视图的现有机架中。

规则

- 可以复制插在机架上和未插入模块区域中的模块。
- 只能在遵守插槽规则的情况下将设备和模块复制到空闲的有效插槽中。
- 无法单独复制 CPU 和插有 CPU 的机架，只能以完整单元的形式与所有插入的硬件组件一起复制。

步骤

要复制硬件组件，请按以下步骤操作：

1. 选择要复制的硬件组件。
 - 设备视图：选择机架上或未插入模块区域中的模块。
 - 网络视图：从网络视图中选择站或相关硬件组件。
 - 项目树：选择站或模块。
2. 从快捷菜单中选择“复制”(Copy) 或按 <Ctrl+C>。

如果“复制”(Copy) 菜单项不可用，说明所选内容中至少包含一个不能复制的组件。
3. 选择要粘贴剪贴板内容的位置。
 - 设备视图：选择机架上的一个空闲插槽或未插入模块的区域。
 - 网络视图：在网络视图中选择要插入设备或模块的站或者将鼠标指针移动到网络视图中要粘贴复制站或相关硬件组件的空闲位置。
4. 从快捷菜单中选择“粘贴”(Paste) 或按 <Ctrl+V>。

如果“粘贴”(Paste) 菜单项不可用，说明剪贴板为空或至少包含一个无法在该位置粘贴的组件。

所选对象将粘贴到选定的位置。

在网络视图中选择要插入模块的站后，该模块即会插入到第一个空闲的有效插槽中。如果没有空闲的有效插槽，对象将插入到未插入模块区域。

注意

也可将模块从一个设备复制到另一个设备：

为此，可在硬件和网络编辑器中复制模块，在网络视图中或设备视图的下拉列表中选择另一个设备，然后插入模块。

可以将复制的对象直接插入插槽中或将其置于设备视图中的未插入模块区域。如果在站的网络视图中添加复制的对象，则会将其插入到第一个可用的插槽中。

如果没有可用于该对象的插槽，则会自动将其放置到未插入模块区域 (页 193)。

注意

可以使用 <Ctrl> 和拖放操作直接复制选定的硬件组件。

参见

硬件和网络编辑器中的键盘操作 (页 187)

6.2.2.5 移动硬件组件

可以在设备或网络视图中移动硬件组件。

规则

- 可以在遵守插槽规则的情况下移动机架上和未插入模块区域中的设备和模块。
- 无法移动 CPU。

步骤

要移动硬件组件，请按以下步骤操作：

1. 选择要移动的硬件组件。
 - 设备视图：选择机架上的模块或将其放到未插入模块区域中。
 - 网络视图：选择与网络视图有关的硬件组件。
2. 从快捷菜单中选择“剪切”(Cut) 或按 <Ctrl+X>。

如果“剪切”(Cut) 菜单项不可用，说明所选内容中至少包含一个不能剪切的组件。
3. 选择被剪切对象将移动到的位置。
 - 设备视图：选择机架上的一个空闲插槽或未插入模块的区域。
 - 网络视图：选择要插入设备或模块的站。
4. 从快捷菜单中选择“粘贴”(Paste) 或按 <Ctrl+V>。

如果“粘贴”(Paste) 菜单项不可用，说明剪贴板为空或至少包含一个无法在该位置粘贴的组件。

所选的硬件组件将移动到目标位置。如果要移动的硬件组件是已联网对象，则会将其从网络中断开。

注意

可以通过拖放操作直接移动选定的硬件组件。

参见

硬件和网络编辑器中的键盘操作 (页 187)

6.2.2.6 替换硬件组件

用户可以替换硬件组件。例如，用户可以使用硬件目录中提供的 CPU 替换未指定的 CPU (页 275)。

规则

只有硬件组件支持模块替换且两个组件兼容时，才能替换硬件组件。

步骤

要用一个模块替换另一个模块，请按以下步骤操作：

1. 选择要替换的模块。
2. 打开快捷菜单：
 - 如果启用了“更改设备类型”(Change device type) 条目，则可以替换该模块。
 - 如果禁用了“更改设备类型”(Change device type) 条目，则无法替换该模块。
3. 在快捷菜单中，单击“更改设备类型”(Change device type)。将打开“更改设备类型”(Change device type) 对话框。
4. 在树结构中的“新设备”(New device) 下，选择要用来替换当前模块的模块。
5. 单击“确定”(OK)。

现有模块即会被新模块替换。

也可以将模块从硬件目录拖动到要替换的模块来替换模块。如果所选的模块可以替换该模块，则将通过鼠标指针符号加以指示。

6.2.2.7 将信号板插入到 CPU 中**简介**

使用信号板可以增加 CPU 自身的输入和输出数目。与所有其它硬件组件一样，也可以在硬件目录中找到信号板。但不是象其它模块那样将信号板插到机架上，而是直接插在 CPU 自身的插槽中。

使用信号板时请注意以下几点：

- 一个 CPU 只能插一个信号板。
- 只有 CPU 上的插槽空闲时，才可插入信号板。

有多种将信号板插入 CPU 中的方法：

- CPU 上有空闲插槽时，在硬件目录中双击信号板
- 将信号板从硬件目录拖到 CPU 上的空闲插槽
- 使用硬件目录中的相应信号板的快捷菜单进行复制和粘贴

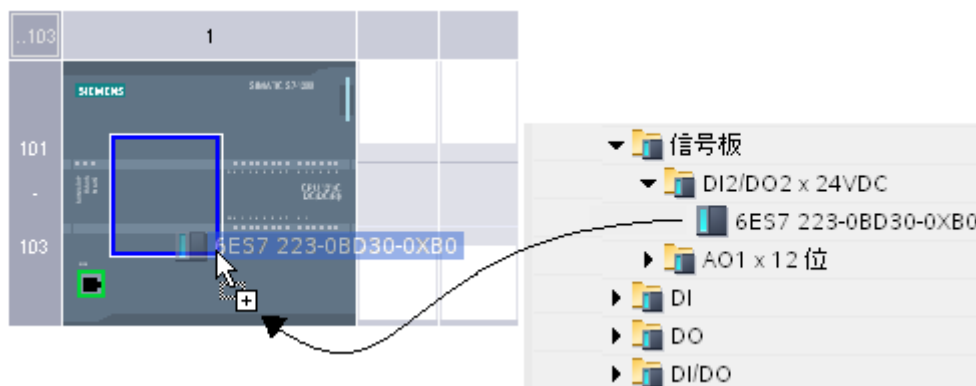
要求

- 硬件目录已打开。
- CPU 有用于信号板的空闲插槽。

将信号板插入到 CPU 中

要将信号板插入 CPU 中，请按以下步骤操作：

1. 转到硬件目录中所需的信号板。
2. 选择要使用的信号板。
3. 将该信号板拖到 CPU 的空闲插槽上。



现在已完成将信号板插入 CPU 上的插槽中。

如果您处于网络视图中，也可将信号板拖到设备上。如果 CPU 有用于信号板的空闲插槽，则信号板将自动插到该插槽中。

6.2.2.8 编辑属性和参数

在机架上插入硬件组件后，即可编辑其默认属性，例如，在网络或设备视图中编辑参数或地址。

要求

您处于设备视图中。

注意

也可在网络视图中编辑属性和参数。在图形网络视图中，可以访问网络相关的硬件组件和站。通过表格网络视图可以访问图形网络视图中未显示的模块和硬件组件。

步骤

要更改硬件组件的属性和参数，请按以下步骤操作：

1. 在图形视图中，选择要编辑的 CPU、模块、机架或接口。
2. 编辑所选对象的设置：
 - 例如，使用表格视图编辑地址和名称。
 - 在巡视窗口中，可以在“属性”(Properties) 中进行其它可能的设置。

更改设置实例

模块	插槽	I 地址	Q 地址	类型	订货号
DI8 x DC24V / 2	2	8		SM 1221 DI8 x 24	6ES7 221-1BF30-0XBC
AI4 x 13bits / 3	3	112...11	112...11	SM 1234 AI4/AO2	6ES7 234-4HE30-0XBI

DI8 x DC24V_1

常规

- General
- DI8
 - Digital inputs
 - IO 地址/硬件标识符

IO 地址/硬件标识符

输入地址

起始地址: 8

结束地址: 8

过程映像: 循环 PI

硬件标识符

硬件 ID: 12

- ① 模块选择
- ② 设备概览中的地址编辑选项
- ③ 巡视窗口中的选择选项
- ④ 巡视窗口中的地址编辑选项

参见

巡视窗口 (页 183)

6.3 联网设备

6.3.1 通信和网络

设备间的通信

各种通信的基础始终是一个预先组态好的网络。网络组态构成了通信的基础，也就是说：

- 网络中的所有设备都具有唯一的地址；
- 设备负责处理传输属性一致的通信。

网络组态

组态网络时必须执行以下步骤：

- 将设备连接到子网
- 为每个子网指定属性/参数
- 为每个联网模块指定设备属性
- 将组态数据下载到设备以给接口提供网络组态所生成的设置
- 保存网络组态

对于开放式用户通信，可通过连接参数分配来创建和组态子网。

网络组态和项目之间的关系

子网及其属性在项目中管理。属性主要通过可调整的网络参数以及所连接设备的数量和通信属性派生而来。

要联网的设备必须在同一个项目中。

子网名称和子网 ID

在项目中，通过子网名称和 ID 明确地标识子网。子网 ID 与可互连的接口一起保存在所有组件中。这样，即使上传到项目中后，也可明确地将组件分配给子网。

联网选项

在项目中，可以创建设备并将其与具有通信功能的组件联网。可使用以下基本选项来执行设备联网：

- 连接彼此间能够通信的组件的接口。将创建适合该接口类型的新子网。
- 连接能够与新子网或现有子网通信的设备的接口。
- 创建开放式用户通信连接。将参数分配给用于开放式用户通信的连接时，将自动在通信伙伴间创建一个子网。

由于设备的任务不同或工厂规模不同，可能需要使用多个子网。这些子网在项目中进行管理。

6.3.2 在网络视图中联网设备

选项

在图形网络视图中可总览项目中整个系统的子网。可使用表格网络概览以获得更多支持。

根据起始条件，可以采用多种方式执行组态以将具有通信功能的组件的接口联网。可能的起始条件有：

- 还没有合适的子网。
- 要连接组件的子网已经存在。

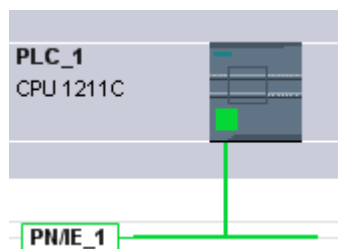
步骤 - 创建单个子网

要创建子网并将其连接到一个接口，请按以下步骤操作：

1. 选择 CPU 的接口。
2. 在接口的快捷菜单中选择“创建子网”(Create subnet) 命令。

所选接口将连接到新子网。自动为该接口设置了一致的地址参数。

下图显示了引出线路连接到子网的接口：



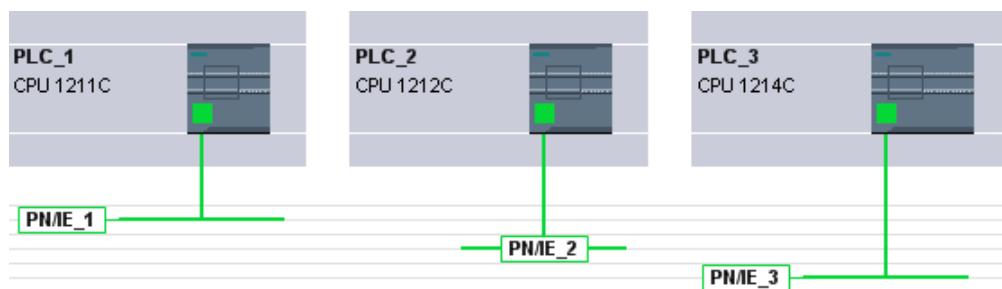
步骤 - 一次创建多个子网

要一次创建多个子网，请按以下步骤操作：

1. 通过按住 <Ctrl> 按钮的同时单击接口来选择若干接口。
2. 在接口的快捷菜单中选择“创建子网”(Create subnet) 命令。

所选的各接口将连接到新子网。同时会自动为接口设置一致的地址参数。

下图显示了通过选择多个接口创建的多个子网：

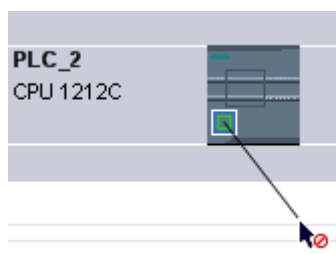


步骤 - 将两个目标设备连接到新子网

要通过尚不存在的子网连接接口与其它设备，请按以下步骤操作：

1. 将鼠标指针置于有通信功能且需要联网的组件的接口上。
2. 单击鼠标左键并按住该按钮。
3. 移动鼠标指针。

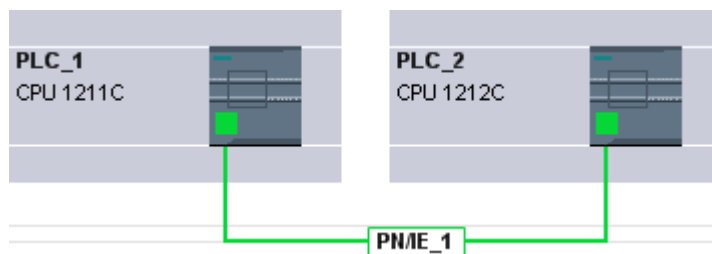
指针此时将使用联网符号来指示“联网”模式。同时，鼠标指针显示锁定符号，该符号只有在指针移动到有效目标上时才会消失。



4. 现在，将指针移动到目标设备的接口上。可以一直按住鼠标按钮，也可释放鼠标按钮。
5. 此时根据前面的操作释放鼠标左键或再次按下。

创建了新的子网。相关接口现在通过新子网相连接。同时会自动为接口设置一致的地址参数。

下图显示了两个联网的设备：



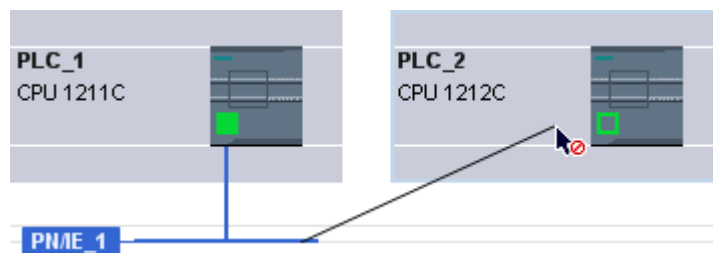
步骤 - 将两个设备连接到现有子网

要将接口连接到现有子网，请按以下步骤操作：

1. 将鼠标指针置于有通信功能且需要联网的组件上或现有子网上。
2. 单击鼠标左键并按住该按钮。
3. 移动鼠标指针。

指针此时将使用联网符号来指示“联网”模式。同时，鼠标指针显示锁定符号，该符号只有在指针移动到有效目标上时才会消失。

- 现在，将鼠标指针移动到现有子网或要联网的接口上。可以一直按住鼠标按钮，也可释放鼠标按钮。



- 此时根据前面的操作释放鼠标左键或再次按下。

结果：

接口和所选子网现在会相互连接。同时会自动为接口设置一致的地址参数。

步骤 - 从列表中选择现有子网

要连接接口与已创建的子网，请按以下步骤操作：

- 选择 CPU 的接口。
- 在接口的快捷菜单中选择“分配到新子网”(Assign to new subnet) 命令。

将出现一个列表框，其中包含可用子网。

- 从列表中选择子网。

接口和所选子网现在会相互连接。同时会自动为接口设置一致的地址参数。

6.3.3 表格式网络概览

含义

表格式网络概览会将以下功能添加到了图形网络视图中：

- 用户可以获得有关设备结构和参数设置的详细信息。
- 通过“子网”(Subnet) 列，可以连接具有通信功能的组件与已创建的子网。

用于表格的基本功能

网络概览支持以下用于编辑表格的基本功能：

- 显示和隐藏表格列
注意：无法隐藏与组态有关的列。
- 优化列宽
- 排序表格
- 通过工具提示显示列、行或域的含义。

6.3.4 在设备视图中联网设备

在设备视图中联网

在设备视图中，可以详细地检查和设置所有组件和接口的所有参数。在此处还可以将接口分配给项目中创建的子网。

要求

- 已经创建要连接到接口的子网。
- 如果尚未创建子网，则切换到网络视图并进行联网所需的设置。

步骤 - 连接到现有子网

要将接口连接到现有子网，请按以下步骤操作：

1. 选择要联网的整个有通信功能的组件或接口。
所选接口或组件的属性将显示在巡视窗口中。
2. 在巡视窗口中选择所选接口的“以太网地址”(Ethernet Addresses) 参数组。
3. 从“接口的连接目标”(Interface connected with) 下拉列表中选择要连接的子网。

接口和所选子网现在会相互连接。同时会自动为接口设置一致的地址参数。

步骤 - 创建新子网

要创建子网并将其连接到指定接口，请在设备视图中按以下步骤操作：

1. 选择要联网的整个有通信功能的组件或接口。
所选接口或组件的属性将显示在巡视窗口中。
2. 在巡视窗口中选择所选接口的“以太网地址”(Ethernet Addresses) 参数组。
3. 在“接口的连接目标”(Interface connected with) 中，单击“添加新子网”(Add new subnet) 按钮。

该接口将连接到具有相应子网类型的新子网。同时会自动为接口设置一致的地址参数。

6.3.5 检查或更改网络参数和接口参数

简介

联网设备间的通信要求组态以下参数：

- 网络参数
网络参数标识系统组态内的网络，例如，使用名称标识。

- 接口参数
接口参数定义具备通信功能的组件的特定属性。地址和传输特性会自动设置并且与网络参数一致。

注意

网络参数和接口参数通常在联网期间设置，以便在不必更改参数的情况下，众多应用设备间可以进行通信。

步骤 - 检查或更改网络参数

要检查或更改网络参数，请按以下步骤操作：

1. 进入网络视图。
2. 从网络视图中选择子网。
可以在巡视窗口的“属性”(Properties) 选项卡中查看网络参数。
3. 必要时，检查或修改相关参数组中的网络参数。

步骤 - 检查或更改接口参数

可以在网络和设备视图中检查和修改接口参数。

要检查或更改接口参数，请按以下步骤操作：

1. 进入网络视图或设备视图。
2. 选择接口。
可以在巡视窗口的“属性”(Properties) 选项卡中查看接口参数。
3. 必要时，检查或修改相关参数组中的接口参数。

6.3.6 更改联网设置

简介

可以取消接口的网络连接或将其分配给其它子网类型相同的子网。

结果

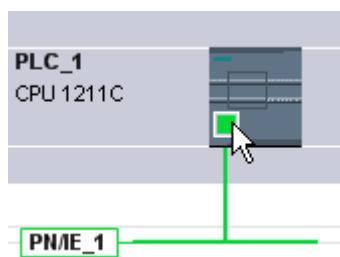
根据版本，下述更改之间会有区别：

- 取消接口的网络连接
组态的接口参数保持不变。
- 将网络连接分配给另一个子网
如果地址在分配的子网中不唯一，即地址已经存在，则会自动更改地址以使其唯一。

步骤 - 从网络中断开

要取消接口的网络连接，请按以下步骤操作：

1. 选择联网的接口。



2. 在接口的快捷菜单中选择“从子网断开”(Disconnect from subnet) 命令。

将删除网络连接，但接口地址仍保持不变。

组态的连接会得到保留；但这些连接会因为缺少联网而在连接表中以红色标记。指定的连接仍处于被指定状态。

参见

在网络视图中联网设备 (页 205)

6.3.7 复制、剪切或删除子网

简介

可以以单个对象的形式复制子网或将其与联网的设备或其它网络一起复制。

例如，您可以创建复杂的组态以在项目中不同的地方多次使用而无需再执行更多的工作。

对所复制子网的影响

复制已复制的对象时，将相应地重新分配必须在项目中明确分配的属性。

对于子网，这意味着：子网 ID 和名称将重新分配给所复制的子网。

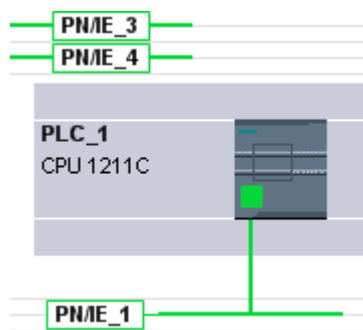
在所复制的子网中采用组态的属性。

步骤 - 复制子网

要复制一个或多个子网，请按以下步骤操作：

1. 选择一个或多个子网。
2. 在快捷菜单中，选择“复制”(Copy) 命令。
3. 在快捷菜单中选择“粘贴”(Paste) 命令。

复制的子网在网络视图的上半部分显示成“孤立”的子网。



步骤 - 复制已连接设备的子网

要复制一个或多个具有联网设备的子网，请按以下步骤操作：

1. 选择一个或多个已连接设备的子网，例如，通过框选。
2. 在快捷菜单中，选择“复制”(Copy) 命令。
3. 在快捷菜单中选择“粘贴”(Paste) 命令。

将创建子网和所连接设备的完整副本。

组态的连接被采用并保留在复制的设备中。指向尚未复制的设备的连接被中断，并变为未被指定状态。

6.3.8 组态工业以太网

网络组态规则

以太网接口具有一个默认 IP 地址，用户可以更改该地址。

IP 地址

如果具有通信功能的模块支持 TCP/CP 协议，则 IP 参数可见。通常对于所有以太网模块都是这样。

IP 地址由 4 个 0 到 255 之间的十进制数字组成。各十进制数字相互之间用点隔开。

实例：140.80.0.2

IP 地址包括

- (子)网的地址
- 节点的地址 (通常也称为主机或网络节点)

子网掩码

子网掩码将这两个地址拆分。它确定 IP 地址的哪一部分用于网络定址，哪一部分用于节点定址。

子网掩码的设置位确定 IP 地址的网络部分。

实例：

子网掩码：255.255.0.0 = 11111111.11111111.00000000.00000000

在针对上述 IP 地址给出的实例中，此处显示的子网掩码具有以下含义：

IP 地址的前 2 个字节标识子网 - 即 140.80。最后的两个字节标识节点，例如 0.2。

通常的实际情况是：

- 用 AND 连接 IP 地址和子网掩码就产生网络地址。
- 用 AND NOT 连接 IP 地址和子网掩码就产生节点地址。

IP 地址和默认子网掩码之间的关系

在 IP 地址范围的分配和所谓的“默认子网掩码”方面，存在一个共识。IP 地址中的第一个十进制数字（从左边起）决定默认子网掩码的结构。如下所示，它决定数值“1”（二进制）的个数：

IP 地址（十进制）	IP 地址（二进制）	地址类别	默认子网掩码
0 到 126	0xxxxxxxx.xxxxxxxxx....	A	255.0.0.0
128 到 191	10xxxxxxxx.xxxxxxxxx....	B	255.255.0.0
192 到 223	110xxxxx.xxxxxxxxx...	C	255.255.255.0

注意

第一个十进制小数点的值范围

IP 地址的第一个十进制数字也可以是 224 和 255 之间的值（地址类别 D 等）。但由于对这些值不进行地址检查，因此不建议使用该方法。

屏蔽其它子网

可使用子网掩码添加更多结构并为被指定了地址类别 A、B 或 C 之一的子网形成“专用”子网。这通过将子网掩码的其它低位部分设置为“1”实现。每将一个位设置为“1”，“专用”网络的数目就会加倍，而它们包含的节点数将减半。在外部，该网络像以前那样，以单个网络的方式运行。

实例：

您有一个地址类别为 B 的子网（例如 IP 地址 129.80.xxx.xxx）并按以下所示更改默认子网掩码：

掩码	十进制	二进制
默认子网掩码	255.255.0.0	11111111.11111111.00000000.00000000
子网掩码	255.255.128.0	11111111.11111111.10000000.00000000

结果：

地址在 129.80.001.xxx 和 129.80.127.xxx 之间的所有节点都位于一个子网上，地址在 129.80.128.xxx 和 129.80.255.xxx 之间的所有节点都位于另一个子网上。

路由器

路由器的工作是连接子网。如果要 IP 数据报发送到另一个网络，则它必须先传送到路由器。为实现此目的，这种情况下，必须为子网中的每个节点输入路由器的地址。

子网中节点的 IP 地址和路由器的地址可能只是在子网掩码中有“0”的部分有所差异。

6.3.9 开放式用户通信

6.3.9.1 开放式用户通信的基本知识

简介

开放式用户通信是一种程序控制的通信技术的名称，它用于通过 CPU 的集成 PN/IE 接口进行通信。许多连接类型都适用于此通信技术。

开放式用户通信的主要特点是在所传送的数据结构方面具有高度的灵活性。这就允许 CPU 与任何通信设备进行开放式数据交换，前提是这些设备支持该集成接口可用的连接类型。由于此通信仅由用户程序中的指令进行控制，因此可建立和终止事件驱动型连接。在运行期间，也可以通过用户程序修改连接。

对于具有集成 PN/IE 接口的 S7-1200 CPU，可使用 TCP 和 ISO-on-TCP 连接类型进行开放式用户通信。通信伙伴可以是两个 SIMATIC PLC，也可以是 SIMATIC PLC 与适合的第三方设备。

开放式用户通信的指令

要创建连接，在打开程序编辑器后，可使用“指令 > 扩展指令 > 通信”(Instructions > Extended instructions > Communication) 任务卡中提供的各种指令：

- 用于发送和接收数据并集成了连接建立/终止功能的简化指令：
 - [TSEND_C \(页 891\)](#) (连接建立/终止，发送)
 - [TRCV_C \(页 894\)](#) (连接建立/终止，接收)
- 单独用于发送或接收数据或者用于建立或终止连接的指令：
 - [TCON \(页 898\)](#) (连接建立)
 - [TDISCON \(页 900\)](#) (连接终止)
 - [TSEND \(页 902\)](#) (发送)
 - [TRCV \(页 904\)](#) (接收)

连接建立

对于开放式用户通信，两个通信伙伴都必须具有用来建立和终止连接的指令。其中一个通信伙伴通过 TSEND 或 TSEND_C 发送数据，而另一个通信伙伴通过 TRCV 或 TRCV_C 接收数据。

其中一个通信伙伴作为主动方启动连接建立过程。另一个通信伙伴通过作为被动方启动连接建立过程来进行响应。如果两个通信伙伴都触发了连接建立过程，操作系统便完全能够建立通信连接。

连接参数分配

可按如下方式使用具有 TCON_Param 结构的连接描述 DB 来分配参数以建立连接：

- 手动创建、分配参数并直接写入指令。
- 连接参数分配支持的方式。

由于可通过连接参数分配建立连接，因此，该方法的优先级高于其它方法。

用户可在连接参数分配中指定以下内容：

- 连接伙伴
- 连接类型
- 连接 ID
- 连接描述 DB
- 与所选连接类型相应的地址详细信息

此外，可在此指定哪个通信伙伴发起连接建立过程，哪个伙伴为响应其通信伙伴的请求而被动建立连接。

参见

面向连接协议的工作原理 (页 276)

6.3.9.2 连接参数分配概述

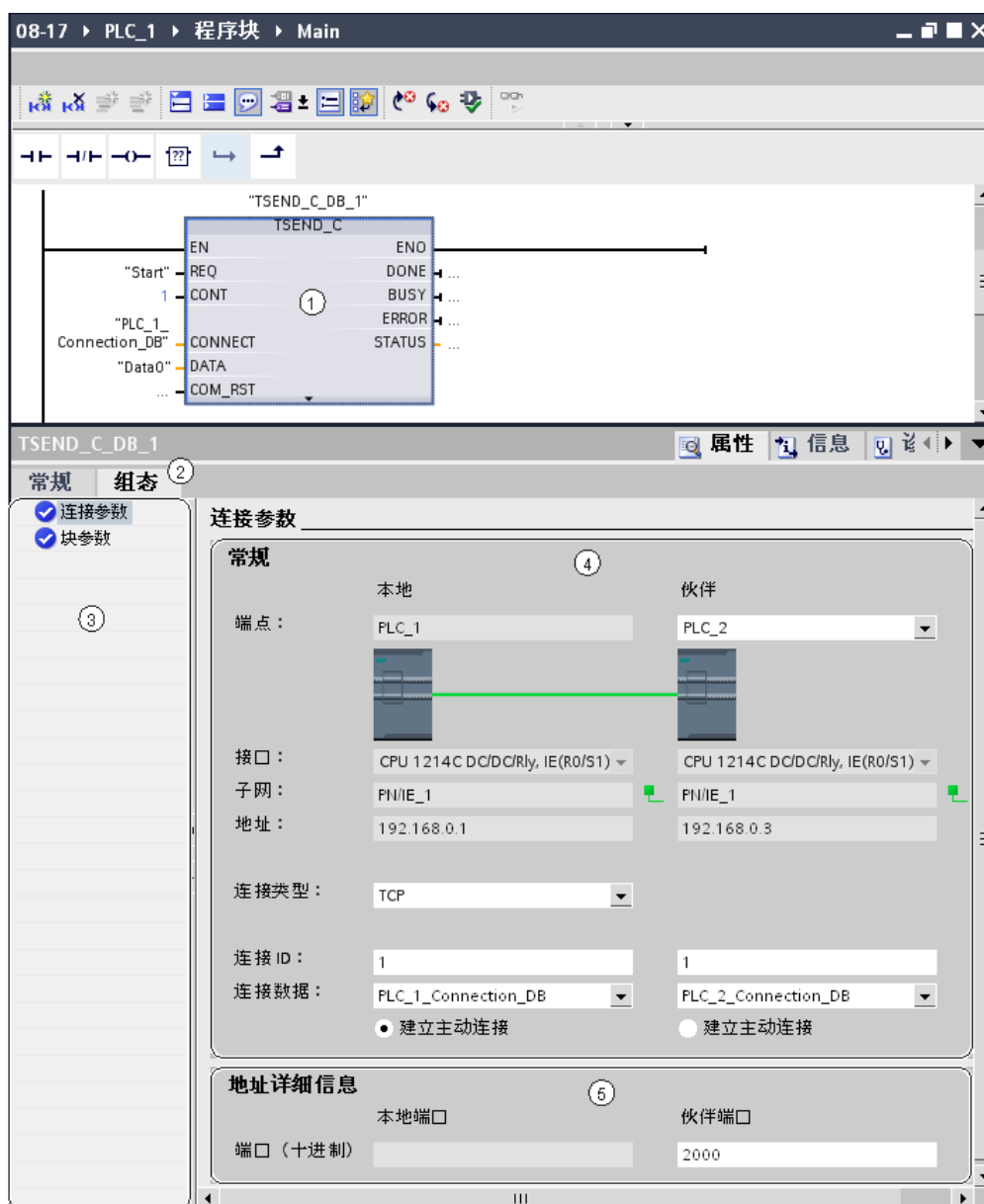
简介

如果要通过 TSEND_C、TRCV_C 或 TCON 扩展指令来编写开放式用户通信程序，可在程序编辑器的巡视窗口中找到连接参数分配区域。

连接参数分配支持灵活的通信编程功能：在连接参数分配中输入的参数将存储在使用类型 TCON_Param 的固定结构自动创建的背景数据块中。可在此连接描述 DB 中修改连接参数。

连接参数分配的结构

连接参数分配由以下部分组成：



- ① 用于开放式用户通信的扩展指令 (TCON、TSEND_C 或 TRCV_C)
- ② “属性”(Properties) 选项卡中的“组态”(Configuration) 选项卡
- ③ “组态”(Configuration) 选项卡的区域导航
- ④ 连接参数的常规属性
- ⑤ 连接参数中的地址详细信息

“组态”选项卡

在“组态”(Configuration) 选项卡中输入所需的连接参数。“组态”(Configuration) 选项卡的区域导航包含“连接参数”(Connection parameters) 组。该组包含连接参数分配。可以在此处借助系统

支持，输入连接参数以及地址详细信息。也可以在此处初始化所选通信指令的 CONNECT (TCON、TSEND_C、TRCV_C) 和 ID (TCON、TSEND、TRCV) 参数。

分配了所有需要的参数后，区域导航中的“连接参数”(Connection parameters) 组前面将设置一个复选标记。

注意事项

连接参数分配不会检查连接 ID 和端口号 (TCP) 或 TSAP (ISO-on-TCP) 是否唯一。因此，组态开放式用户通信时，应确保参数设置在设备内是唯一的。

参见

通信连接的参数 (页 277)

6.3.9.3 连接参数描述

概述

下表列出了常规的连接参数：

参数	描述
端点	<p>显示本地端点和伙伴端点的名称。</p> <p>本地端点就是为其设置 TCON、TSEND_C 或 TRCV_C 的 CPU。因此，本地端点始终是已知的。</p> <p>伙伴端点则需要从下拉列表中选择。下拉列表将显示所有可用的连接伙伴，包括那些项目中还未知其数据的设备对应的未指定的连接伙伴。</p> <p>只要未设置任何连接伙伴，就会禁用其它所有被屏蔽的参数。</p>
接口	<p>显示本地端点的接口。只有指定伙伴端点后，才会显示伙伴接口。</p>
子网	<p>显示本地端点的子网。只有选择伙伴端点后，才会显示伙伴子网。</p> <p>如果所选伙伴端点未通过子网连接到本地端点，则会自动将两个连接伙伴联网。为此，必须指定伙伴端点。</p> <p>不同子网中的伙伴之间只能通过 IP 路由建立连接。可在相关的接口属性中编辑路由设置。</p>
地址	<p>显示本地端点的 IP 地址。只有选择伙伴端点后，才会显示伙伴的 IP 地址。</p> <p>如果选择了未指定的连接伙伴，输入框将为空并且背景为红色。在这种情况下，需要指定有效的 IP 地址。</p>
连接类型	<p>从“连接类型”(Connection type) 下拉列表选择要使用的连接类型：</p> <ul style="list-style-type: none"> • TCP • ISO-on-TCP <p>所需连接数据的参数会因所选连接类型的不同而变化。</p>

参数	描述
连接 ID	在输入框中输入连接 ID。创建新连接时，会分配默认值 1。可以在输入框中更改连接 ID，也可以在 TCON 中直接输入连接 ID。 请确保所分配的连接 ID 在设备内是唯一的。
连接数据	下拉列表中将显示其连接描述结构符合 TCON_Param 的连接描述 DB 的名称。 创建连接时，将为指定的每个连接伙伴生成一个数据块，并会用连接参数分配的值自动填充该数据块。对于本地连接伙伴，所选数据块的名称将自动输入所选 TSEND_C、TRCV_C 或 TCON 指令的块参数 CONNECT 中。 对于另一个连接伙伴，也可以在 TSEND_C、TRCV_C 或 TCON 指令的 CONNECT 输入中直接使用第一个连接伙伴所生成的连接描述 DB。对于本步骤，可在选择第一个连接伙伴后使用现有的连接描述 DB，或创建新的连接描述 DB。 也可以从下拉列表中引用另一个有效数据块。如果使用 TSEND_C、TRCV_C 或 TCON 扩展指令的 CONNECT 输入参数引用数据块，而该数据块的结构与 TCON_Param 的结构不符，则下拉列表将不显示任何内容，且背景为红色。
主动连接建立	启用“主动连接建立”(Active connection establishment) 复选框，可指定开放式用户通信的主动方。
端口 (仅限 TCP)	TCP 连接的地址部分。创建新的 TCP 连接之后的默认值为 2000。 可以更改端口号。 端口号必须在设备中唯一！
TSAP (仅限 ISO-on-TCP)	ISO-on-TCP 连接的地址部分。创建新的 ISO-on-TCP 连接之后的默认值为 E0.01.49.53.4F.6F.6E.54.43.50.2D.31。 可输入具有扩展的 TSAP-ID 或者输入 ASCII TSAP。 TSAP 必须在设备中唯一！

参见

[通信连接的参数 \(页 277\)](#)
[端口号的分配 \(页 280\)](#)
[TSAP 结构 \(页 281\)](#)
[TSAP 分配的实例 \(页 283\)](#)
[回读连接描述参数功能 \(页 280\)](#)
[创建和分配连接参数 \(页 218\)](#)

6.3.9.4 启动连接参数分配

只要在程序块中选择了用于通信的 TCON、TSEND_C 或 TRCV_C 指令，便会启用开放式用户通信的连接参数分配。

要求

- 用户项目中必须至少包含一个 S7-CPU。
- 程序编辑器已打开。
- 有一个可用的程序段。

步骤

要插入开放式用户通信的扩展指令，请按以下步骤操作：

1. 打开任务卡、窗格和文件夹“指令 > 扩展指令 > 通信”(Instructions > Extended instructions > Communication)。
2. 将以下指令之一拖到程序段中：
 - TSEND_C
 - TRCV_C
 - TCON (在“其它”(Others)子文件夹中)

将打开“调用选项”(Call options)对话框。

3. 在“调用属性”(Call properties)对话框中，编辑背景数据块的属性。用户有以下选择：
 - 更改默认名称。
 - 选中“手动”(Manual)复选框分配用户自己的编号。
4. 单击“确定”(OK)完成输入。

结果

创建了一个根据 TCON_Param 构造的连接描述 DB，且它是所插入指令的背景数据块。

选中 TSEND_C、TRCV_C 或 TCON 时，可在巡视窗口的“属性”(Properties)下看到“组态”(Configuration)选项卡。现在，用户可以在区域导航的“连接参数”(Connection parameters)组中进行连接参数分配。

参见

使用“指令”任务卡插入 FBD 元素 (页 447)

使用“指令”任务卡插入 LAD 元素 (页 417)

创建和分配连接参数 (页 218)

6.3.9.5 创建和分配连接参数

在开放式用户通信的连接参数分配中，可创建 TCP 或 ISO-on-TCP 类型的连接并设置参数。

要求

CPU 中存在 TCON、TSEND_C 或 TRCV_C 扩展指令。

步骤

要创建开放式用户通信的连接，请按以下步骤操作：

1. 在程序编辑器中，选择开放式用户通信的 TCON、TSEND_C 或 TRCV_C 块。
2. 在巡视窗口中，打开“属性 > 组态”(Properties > Configuration)选项卡。
3. 选择“连接参数”(Connection parameters)组。在选择连接伙伴之前，只启用了伙伴端点的空下拉列表。其它所有输入选项均被禁用。

同时显示一些已知的连接参数：

- 本地端点的名称
 - 本地端点的接口
 - 本地端点的 IP 地址
 - 默认值为 1 的连接 ID
 - 包含连接数据的数据块的唯一名称
 - 作为主动连接伙伴的本地端点
4. 在伙伴端点的下拉列表框中选择连接伙伴。可以选择项目中未指定的设备或 CPU 作为通信伙伴。随后会自动输入一些特定的连接参数。

现有伙伴将自动与本地端点联网，同时会为伙伴 CPU 创建一个数据块，该数据块是根据 TCON_Param 为连接数据构造的。用户需要设置以下参数：

- 伙伴端点的接口
- 本地子网和伙伴子网的名称
- 伙伴端点的 IP 地址
- ISO-on-TCP 连接类型
- 默认值为 1 的连接 ID
- 包含连接数据的数据块的唯一名称
- TSAP-ID E0.01.54.43.50.2D.31

若是未指定的伙伴，请设置以下参数：

- TCP 连接类型
- 端口号 2000

5. 从相关下拉列表中选择所需的连接类型。可在以下类型中选择：

- TCP
- ISO-on-TCP

地址详细信息将根据连接类型在端口号 (TCP) 和 TSAP (ISO-on-TCP) 之间进行切换。

6. 在连接伙伴的相应输入框中，输入连接 ID。不能为未指定的伙伴分配任何连接 ID。
7. 可在相应的“连接数据”(Connection data) 下拉列表中选择其它连接描述 DB，也可以更改连接描述 DB 的名称以创建新的数据块：
- 用户也可以在所选 TCON、TSEND_C 或 TRCV_C 指令的 CONNECT 输入参数互连中查看所选的数据块。
 - 如果已使用 TCON、TSEND_C 或 TRCV_C 指令的 CONNECT 参数为连接伙伴指定了连接描述 DB，则可使用此 DB 或创建一个新 DB。
 - 如果编辑下拉列表中所显示的数据块的名称，则会生成一个新数据块用于该连接，新数据块使用更改的名称，但结构和内容不变。
 - 更改的数据块名称必须在通信伙伴环境中唯一。
 - 连接描述 DB 必须具有 TCON_Param 结构。
 - 无法为未指定的伙伴选择数据块。

8. 使用“主动连接建立”(Active connection establishment) 复选框设置连接建立行为。用户可以决定由哪个通信伙伴主动建立连接。
9. 可以编辑地址详细信息中的输入框。根据所选的协议，可以编辑端口 (TCP) 或 TSAP (ISO-on-TCP)。连接参数分配将立即检查更改后的值是否存在输入错误，然后将值输入连接描述数据块中。

注意

只有在将伙伴端点的程序段下载到硬件后，两个通信伙伴之间的开放式用户通信才能进行工作。要实现功能完整的通信，应确保在设备上不仅装载了本地 CPU 的连接描述，而且还装载了伙伴 CPU 的连接描述。

参见

- 连接参数描述 (页 216)*
- 启动连接参数分配 (页 217)*
- TSAP 结构 (页 281)*
- 端口号的分配 (页 280)*
- 通信连接的参数 (页 277)*

6.3.9.6 删除连接

简介

为开放式用户通信所创建的连接数据存储在连接描述 DB 中。通过删除包含连接描述的数据块，便可删除连接。

要求

已创建开放式用户通信连接。

步骤

要删除连接，请按以下步骤操作：

1. 在项目树中，选择开放式用户通信的通信伙伴。
2. 打开所选通信伙伴下方的“程序块”(Program blocks) 文件夹。
3. 从包含连接参数分配的数据块的快捷菜单中选择“删除”(Delete) 命令。

注意

如果不确定要删除哪个块，请打开扩展指令 TCON、TSEND_C 或 TRCV_C。您将找到 CONNECT 输入参数使用的数据块名称，或者连接参数分配中“连接数据”(Connection data) 参数对应的数据块名称。

如果仅删除扩展指令 TCON、TSEND_C 或 TRCV_C 的背景数据块，并不能一同删除所分配的连接。

注意

如果扩展指令的其它块也使用了该连接数据块，则也会从块文件夹中删除相应的调用、背景数据块以及块组合 TSEND_C 和 TRCV_C，前提是未在其它地方使用它们。

此操作可防止程序出现不一致。

结果

删除了连接。

注意

可以再次插入扩展指令 TCON、TSEND_C 或 TRCV_C，以通过“连接数据”(Connection data) 参数引用具有 TCON_Param 结构的现有连接描述。

6.3.10 HMI 连接

6.3.10.1 有关组态连接的简介

定义

连接定义的是为执行通信服务而对两个通信伙伴进行的逻辑分配。连接定义了以下方面：

- 涉及的通信伙伴
- 连接类型 (例如 HMI 连接)
- 特殊属性 (例如，连接是否是永久建立，或是否是在用户程序中动态建立和终止，以及是否要发送状态消息)
- 连接路径

连接组态须知

连接组态期间，需要为 HMI 连接分配一个本地连接名称作为唯一的本地标识。

在网络视图中，除显示“网络概览”(Network overview) 选项卡之外，还会显示“连接”(Connections) 选项卡。该选项卡包含连接表。此连接表中的每一行都表示一个组态的连接（例如 HMI 设备与 PLC 之间的连接）及其属性。

参见

组态连接 (页 1303)

自动创建连接 (页 1306)

6.3.10.2 连接资源使用须知

简介

每个连接都需要连接资源，用于所涉及设备上的端点和/或转换点。连接资源的数目取决于设备。

如果通信伙伴的所有连接资源均已分配，则不能建立新的连接。当连接表中新创建的连接具有红色背景时，显然就是这种情况。组态随后便会不一致且无法进行编译。

HMI 连接

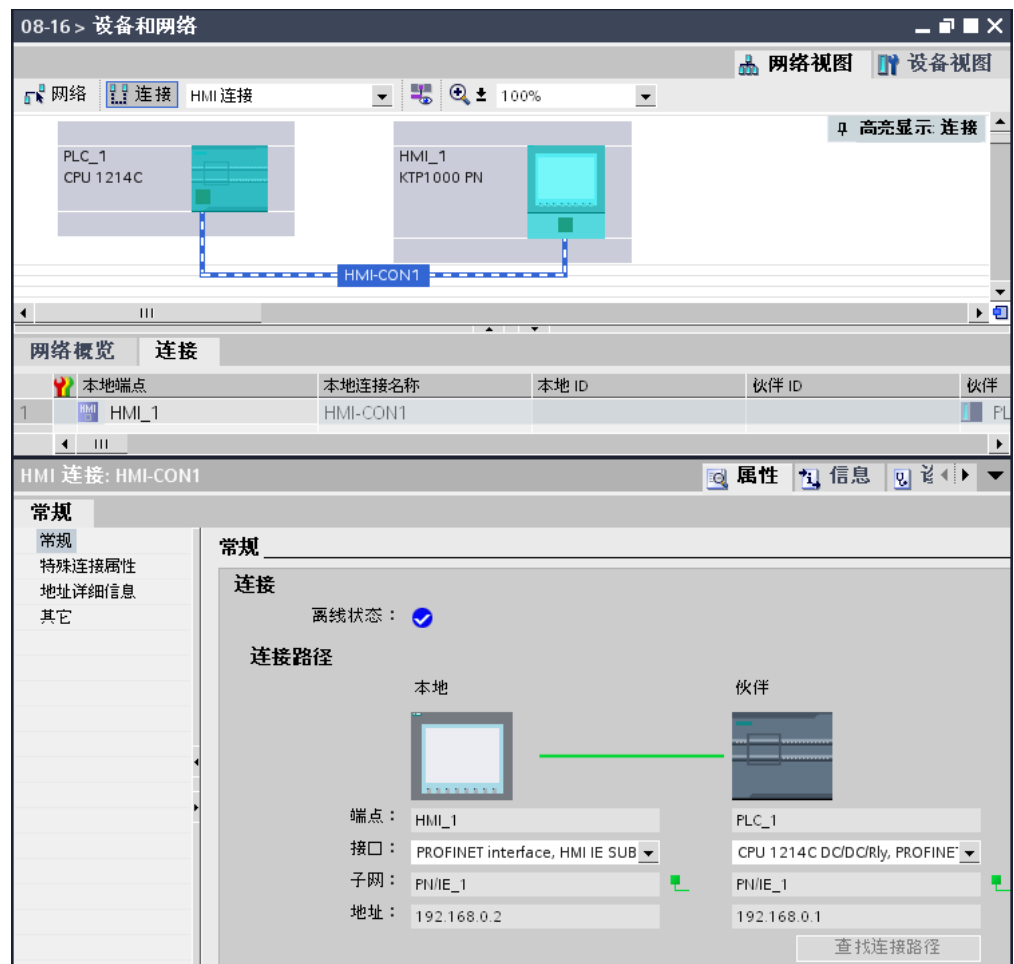
对于通过**集成** PN 接口建立的 HMI 连接，HMI 设备每个 HMI 连接的端点都会占用一个连接资源。

连接伙伴 (PLC) 也需要一个连接资源。

6.3.10.3 包含已组态连接信息的视图

通过以下所述的视图可以对有关组态和检查通信连接的所有信息和功能进行全面访问。

- 网络视图中的连接显示
- 连接表
- 巡视窗口中的连接“属性”(Properties) 选项卡



优点

这些视图中显示的信息始终按照当前的用户操作保持最新。这表示：

- 连接表会显示所创建的所有连接。
- 如果已在连接表中选择一个连接：
 - 将在网络视图中以图形方式显示连接路径。
 - 巡视窗口中的“属性”(Properties) 选项卡将显示此连接的参数。

连接表

连接表提供以下功能：

- 列出项目中的所有连接
- 选择一个连接，然后会在网络视图中显示与其关联的连接
- 更改连接伙伴
- 显示状态信息

巡视窗口中的连接“属性”(Properties) 选项卡

属性对话框具有以下含义：

- 显示连接参数
- 显示连接路径
- 随后可使用“查找连接路径”(Find connection path) 按钮指定连接

6.3.10.4 创建新连接

创建连接 - 备选项

有以下选项可用于在网络视图中创建连接：

- 图形连接组态
- 交互连接组态

可在以下章节中找到此操作的各个步骤。

要求和结果

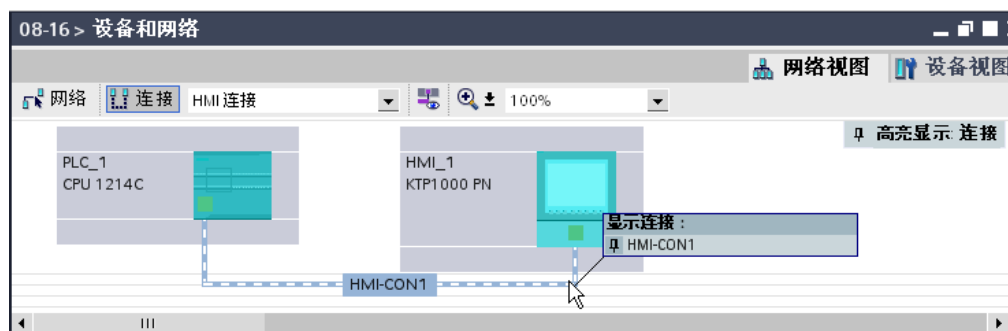
已经创建具有 CPU 的设备和 HMI 设备，用户要在网络视图中组态这些设备之间的连接。

指定连接

如果所选连接类型的两个伙伴都在同一网络上联网，则使用两个通信伙伴的图形或交互选择来创建全指定连接。

该连接将自动输入到 HMI 设备的连接表中。同时会为该连接分配一个本地连接名称。

下面的示意图显示了具有已联网设备的已组态连接：



6.3.10.5 以图形方式创建新连接

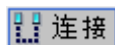
以图形方式组态连接

使用图形连接组态时，系统会根据需要要求用户定义连接路径。在当前组态中选择要连接的设备。

自动确定连接路径

要以图形方式创建连接，请按以下步骤操作：

1. 单击“连接”(Connections) 按钮。

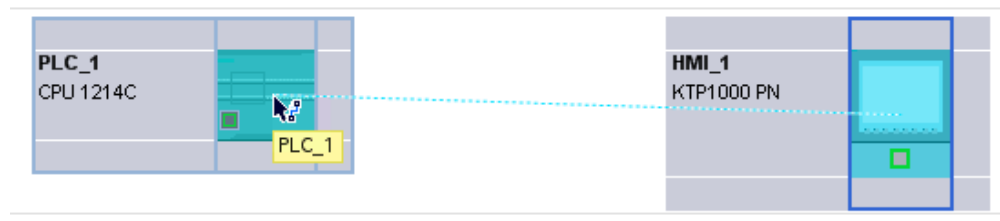


您所选连接类型的连接模式随即激活。

您将从以下变化中看到这点：

项目中可用于所选连接类型的设备在网络视图以彩色高亮显示。

2. 按住鼠标按钮，并将鼠标指针从连接发出的设备拖到连接终止的设备。



3. 在目标设备上释放鼠标按钮以在这两个设备之间创建连接。

结果

- 创建了指定的连接。
- 连接路径高亮显示。
- 连接将输入到连接表中。

6.3.10.6 交互创建新连接

交互组态连接

定义本地设备及其连接伙伴。

步骤

要交互创建连接，请按照以下步骤操作：

1. 在要创建为其连接的连接伙伴的快捷菜单中，选择“创建新连接”(Create new connection) 命令。

将打开“创建新连接”(Create new connection) 对话框。

2. 选择伙伴端点。

在对话框的右窗格中，将显示适合所选端点的可能连接路径（如果适用）。不完整路径（例如，未指定的 CPU 的不完整路径）将以红色背景上的感叹号进行标记。

3. 要关闭对话框，可单击“确定”(OK)。

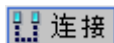
要接受已组态的连接以及要另外组态到其它端点的连接，可单击“应用”(Apply)。

6.3.10.7 在网络视图中工作

高亮显示网络视图中的连接路径和伙伴

要在网络视图中显示所有或某些连接类型的连接伙伴，请执行如下操作：

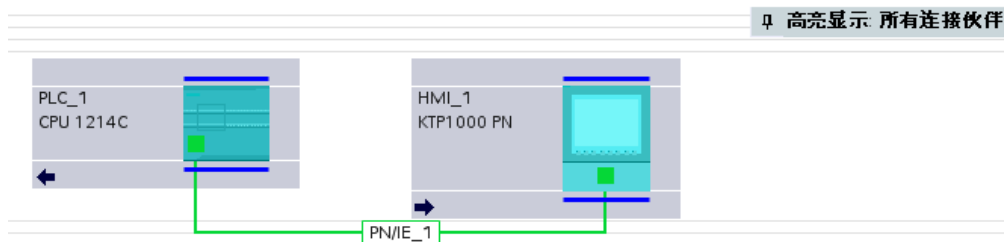
1. 单击“连接”(Connections) 按钮。



2. 对于要在网络视图中显示其连接伙伴的 HMI 设备，在其快捷菜单中选择“高亮显示连接伙伴”(Highlight connection partners) 命令。
3. 在随后出现的菜单中，选择“所有连接伙伴”(All connection partners)。

本地设备以及目标设备的 CPU 将被选中。本地连接伙伴将显示指向右侧的箭头，而远程连接伙伴将显示指向左侧的箭头。

4. 要打开包含目标设备信息的列表，可单击本地设备的相应箭头。此附加功能在有些设备不可见的复杂网络组态中很有用。



注意

可以显示在网络视图的当前显示范围中看不到的其中一个连接伙伴。在显示的列表中单击通信伙伴。结果：显示内容将会移动，以使该连接伙伴变为可见状态。

参见

以图形方式创建新连接 (页 224)

6.3.10.8 使用连接表

用于表格的基本功能

连接表支持下列用于编辑表格的基本功能：

- 更改列宽
- 通过工具提示显示列、行或域的含义。

更改列宽

要根据内容调整列宽，以显示行中的所有文本，请按以下步骤操作：

1. 在连接表的标题中，将光标放在要优化的列的右侧，直到光标形状变成两条平行线（如同您要使用光标拖动列来更改列宽那样）。
2. 双击该点。

对于过窄的列，当将光标暂停在相应区域上时，会显示该区域的全部内容。

使用光标键在连接表内移动

可使用向上和向下光标键从连接表中选择连接；选定的连接会被标记并在网络视图中高亮显示。

更改连接属性

在某些情况下，可直接编辑连接表中显示的参数。要更改连接的名称，无需浏览到巡视窗口。

更改连接伙伴

可以更改连接的连接伙伴，操作如下：

1. 选择连接。
2. 在“伙伴”(Partner) 列中，从打开的下拉列表中选择新连接伙伴。

6.3.10.9 删除连接

可通过网络视图或连接表删除已组态连接。

在网络视图中，可删除高亮显示的连接。在连接表中，可删除一个或多个连接。

在连接表中的操作步骤

要删除高亮显示的连接，可按如下操作进行：

1. 在连接表中，选择要删除的连接。
2. 通过单击鼠标右键，打开快捷菜单。
3. 选择“删除”(Delete) 命令。

将彻底删除选定的连接。

4. 装载相关设备。

6.3.10.1 复制连接

0

简介

连接不会被单独复制，而是始终与项目或设备一起复制。

您可以复制：

6.3 联网设备

- 整个项目
- 在项目内或项目间复制一个或多个设备

复制项目

复制项目时，同时还会复制所有已组态连接。由于连接保持一致，因而不要求对已复制的连接进行任何设置。

复制设备

如果复制已组态了连接的设备（HMI 设备），相应的连接也将被复制。要完成连接路径，仍必须先完成联网。

S7-1200 只是 HMI 连接的服务器，本身没有连接组态。因此，复制 S7-1200 CPU 时，不会复制任何连接。

6.3.10.1 不一致连接 - 没有分配的连接

1

对于不一致连接，连接数据的结构被破坏，或连接在项目环境中不起作用。

不一致连接无法进行编译和装载 - 不能操作这类连接。

在连接表中，不一致连接以红色标记。

引起不一致连接的可能原因

- 删除或更改硬件配置。
- 项目中缺少连接所需的接口网络链接。
- 连接资源过多
- 由于内存不足，备份数据时出错
- 在没有伙伴地址信息的情况下，连接到未指定连接伙伴。

编译后，可在“编译”(Compile) 选项卡（编辑 > 编译）中找到有关不一致原因的详细信息。

解决方法

如果通过打开连接属性、在组态中更改或撤消属性也无法修复连接，则可能必须删除该连接然后重新创建。

6.3.10.1 HMI 连接的常规设置

2

常规连接参数

常规连接参数在该连接属性下的“常规”(General) 参数组中显示；这些连接参数标识本地连接端点。

也可以在此处分配连接路径以及指定连接伙伴的各种属性。

特殊连接属性

显示的连接属性 (无法更改):

- 主动连接建立
始终从 HMI 设备开始建立连接。 如果指定了伙伴的地址，则默认情况下会选择此选项。
- 单向
单向表示连接伙伴在此类连接中作为服务器运行，不能主动发送或接收数据。
- 发送工作模式消息
与 HMI 设备无关。

地址详细信息

显示 HMI 连接的地址详细信息。 对于未指定的伙伴，可以更改机架和插槽的值。 其它所有值都从当前组态获得，且无法更改。

其它

显示 HMI 设备与连接伙伴之间的在线连接的访问点。

6.4 组态操作

6.4.1 选择 CPU

简介

从硬件目录中选择一个 CPU，并且将其与机架一起置于网络视图中。 从硬件目录中将所需的模块拖到该设备上；这些模块会自动排列在机架上。

在硬件目录中选择组件

在硬件目录中，每个硬件组件显示为一个文件夹。 打开此文件夹时，会显示所选硬件组件的不同版本及其订货号。

以下是如何在网络视图中设置 CPU 与机架的示例。

要求

- 硬件目录已打开。
- 您必须处于网络视图中。

步骤

要从硬件目录中选择 CPU，请按以下步骤操作：

1. 在硬件目录中导航到包含所需 CPU 的文件夹。
2. 打开这个具有所需 CPU 型号的文件夹；将显示所选 CPU 型号的所有订货号。
3. 单击某个 CPU 订货号，以便在“信息”(Information) 窗格下获得有关所选 CPU 的信息。



4. 设置 CPU 和机架。您有以下选择：
 - 通过拖放操作将该 CPU 从硬件目录拖入网络视图中。
 - 使用复制和粘贴操作将 CPU 复制到网络视图。
 - 在硬件目录中双击该 CPU 条目。

参见

- [浏览硬件目录 \(页 190\)](#)
- [将设备添加到硬件配置中 \(页 195\)](#)
- [将模块插入机架 \(页 196\)](#)
- [使用机架 \(页 191\)](#)
- [创建未指定的 CPU \(页 275\)](#)
- [有关硬件组件的信息 \(页 186\)](#)

6.4.2 更改模块的属性

默认设置

所有带参数的硬件组件在出厂时，都具有适合标准应用的默认设置。这些默认值使用户可立即使用硬件组件，而无需进行任何其它设置。

但是，用户可以根据应用的要求和环境，修改硬件组件的行为和属性。参数可设置的硬件组件包括通信模块以及若干模拟模块和数字量模块。

设置和装载参数

在设备视图或网络视图中选择了硬件组件后，即可在巡视窗口中设置其属性。保存设备组态及其参数后，将生成需要装载到 CPU 的数据。该数据将在启动期间传送到相关的模块。

CPU 的属性

CPU 的属性对系统行为有特殊意义。例如，对于 CPU 可设置：

- 接口
- 输入和输出
- 高速计数器
- 脉冲发生器
- 启动特性
- 日时钟
- 保护等级
- 系统位存储器和时钟位存储器
- 循环时间
- 通信负载

可输入的条目用于指定哪些设置可以调整以及处于什么值范围。在属性窗口中会禁用或不显示不能编辑的域。

要求

已经在机架上放置了要更改其属性的硬件组件。

步骤

要更改硬件组件的属性和参数，请按以下步骤操作：

1. 在设备视图或网络视图中，选择要编辑的硬件组件或接口。
2. 编辑所选对象的设置：
 - 例如，在设备视图中可以编辑地址和名称。
 - 在巡视窗口中，可以进行其它可能的设置。

无需确认您的输入，即会立即应用所更改的值。

参见

[编辑属性和参数 \(页 202\)](#)

[装载组态简介 \(页 255\)](#)

6.4.3 CPU 属性

6.4.3.1 CPU 属性概述

概述

下表概要说明了 CPU 属性：

组	属性	描述
常规	项目信息	描述所插入的 CPU 的常规信息。可更改除插槽号外的所有其它设置
	目录信息	此 CPU 硬件目录中的只读信息。
PROFINET 接口	常规	此 PROFINET 接口的名称和注释。名称被限制为 128 个字符。
	以太网地址	选择以太网接口是否联网。如果已在项目中创建了子网，则可在下拉列表中进行选择。如果未创建子网，则可使用“添加新子网”(Add new subnet) 按钮创建新子网。 IP 协议中提供了有关子网中 IP 地址、子网掩码和 IP 路由器的使用信息。如果使用 IP 路由器，则需要有关 IP 路由器的 IP 地址信息。
	扩展	以太网接口的名称和端口注释
	日时钟同步	NTP 时间格式的日时钟同步设置。 NTP (Network Time Protocol , 网络时间协议) 是用于同步局域网和全域网中系统时钟的一种通用机制。 在 NTP 模式下, CPU 的接口按固定时间间隔将时间查询 (以客户机模式) 发送到子网 (LAN) 的 NTP 服务器, 同时, 必须在此处的参数中设置地址。将根据服务器的响应计算并同步最可靠、最准确的时间。这种模式的优点是它能够实现跨子网的时间同步。精确度取决于所使用的 NTP 服务器的质量。

组	属性	描述
DI#/DO#	常规	CPU 的名称和集成数字量输入注释。
	数字量输入	可为数字量输入设置输入延迟。可分组设置输入延迟（每组 4 个输入）。 用户可为每个数字量输入启用上升沿和下降沿检测。可为该事件分配名称和硬件中断。 根据 CPU 的不同，可激活各个输入的脉冲捕捉。激活脉冲捕捉后，即使脉冲沿比程序循环时间短，也能将其检测出来。
	数字量输出	可为所有数字量输出设置 RUN 到 STOP 模式切换的响应： 可以将状态冻结（相当于保留上一个值），也可以设置替换值（“0”或“1”）
	I/O/诊断地址	象过程映像中那样指定输入和输出地址的地址空间。显示设备的硬件 ID。
AI#	常规	CPU 的名称和集成模拟量输入注释。
	模拟量输入	在降低噪声期间，指定的积分时间会抑制指定频率（单位为 Hz）大小的干扰频率。 必须在“通道号”(Channel #) 组中指定通道地址、测量类型、电压范围、平滑化和溢出诊断。测量类型和电压范围被永久设置为电压和 0 到 10 V。 模拟值平滑化可以提供稳定的模拟信号，以供进一步处理。模拟值平滑化对于测量值变化缓慢的情况很有用，例如温度测量。使用数字过滤对测量值进行平滑处理。模块通过指定数量的已转换（数字化）模拟值计算平均值来实现平滑化。所选等级（弱、中、强）决定用于创建平均值的模拟信号数量。 如果启用溢出诊断，则发生溢出时会生成诊断事件。
	I/O/诊断地址	象过程映像中那样指定输入地址的地址空间。显示设备的硬件 ID。
高速计数器 (HSC)	高速计数器 (HSC)#	高速计数器通常用于驱动计数机制。 相关的描述和参数分配信息，请参见： 组态高速计数器 (页 243)
脉冲发生器 (PTO/ PWM)	PTO#/PWM#	脉冲发生器被激活且可使用项目信息进行初始化。 对激活的脉冲发生器进行参数分配时，需指定将其用作 PWM（Pulse Width Modulation，脉冲宽度调制）还是 PTO（Pulse Train Output，脉冲串输出）。 对于 PWM，指定输出源、时间基数、脉宽宽度格式、循环时间和初始脉冲宽度。将脉冲输出指定为硬件输出。PWM 输出由 CTRL_PWM 指令控制，请参见 CTRL_PWM (页 942) 。 对于 PTO，指定输出源。将脉冲输出和方向输出指定为硬件输出。PTO 与“运动轴”计数模式下的高速计数器一起运行，并由“运动控制”工艺对象进行控制，请参见 组态 - 常规 (页 580) 。 硬件 ID 显示在“I/O/诊断”地址中，如果选择 PWM 功能，则可选择输出地址的地址空间以及过程映像。
启动	启动类型	设置循环上电后的启动行为。 请参见： STARTUP 模式的原理 (页 288)
日时钟	当地时间和夏令时	设置 CPU 的运行时区以及设置夏令时/标准时间切换。

组	属性	描述
保护	读/写访问保护和密码	保护等级设置选项 (页 254)
系统和时钟存储器位	系统存储器位和时钟存储器位	可使用系统存储器位进行以下扫描： <ul style="list-style-type: none"> • 当前循环是否为循环上电后的第一个循环？ • 与上一个循环相比，诊断状态是否有变化？ • 扫描“1”(高) • 扫描“0”(低) 时钟存储器位按指定的周期性间隔更改自身的值。 请参见：使用时钟存储器 (页 253)
循环时间	最大循环时间和最小循环时间。	请参见：循环时间和最大循环时间 (页 234)
通信负载	周期中分配给通信的最大时间 (以百分比表示)	请参见：通信引起的周期负载 (页 235)
I/O 地址概览	-	CPU 为集成输入/输出和插入模块使用的全部地址的表格表示。 未被任何模块使用的地址以间隙表示。 该视图可按照以下内容进行过滤： <ul style="list-style-type: none"> • 输入地址 • 输出地址 • 地址间隙

参见

[指定输入和输出的地址 \(页 238\)](#)
[组态高速计数器 \(页 243\)](#)
[保护等级设置选项 \(页 254\)](#)
[循环时间和最大循环时间 \(页 234\)](#)
[通信引起的周期负载 \(页 235\)](#)
[给硬件中断 OB 分配参数 \(页 384\)](#)
[STARTUP 模式的原理 \(页 288\)](#)
[访问 I/O 地址 \(页 297\)](#)
[使用时钟存储器 \(页 253\)](#)
[寻址模块 \(页 237\)](#)
[启动期间的特性 \(页 291\)](#)

6.4.3.2 循环时间和最大循环时间

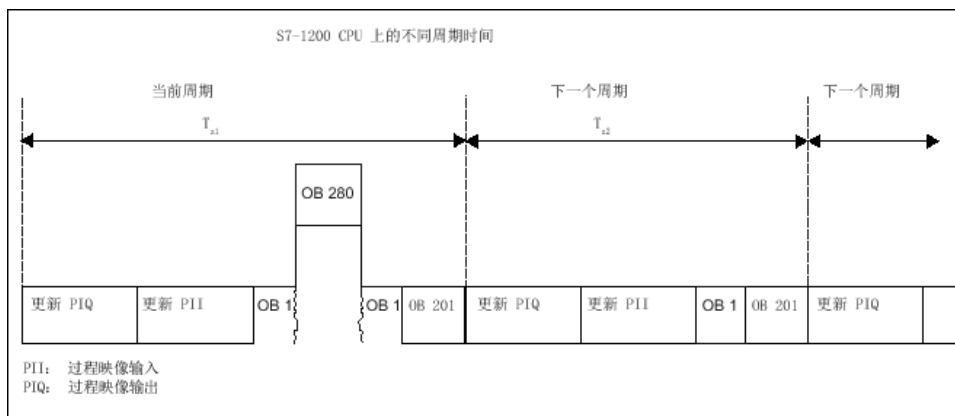
功能

循环时间是指操作系统执行循环程序以及中断此循环的所有程序段所需的时间。程序执行会被以下事件中断：

- 时间错误和 2 次最大循环时间错误
- 系统活动，如过程映像更新

因此，每个循环的循环时间 (T_{cyc}) 均不相同。

下图显示了 S7-1200 CPU 不同循环时间 ($TZ1 \neq TZ2$) 的实例：



在当前循环中，所使用的循环 OB（如 OB 1）被时间错误（如 OB 280）中断。在该循环 OB 之后，处理下一个循环 OB 201。

最大循环时间

操作系统监视循环程序的执行时间是否超过一个可组态的上限时间（称为最大循环时间）。通过调用 RE_TRIGR 指令，可在程序的任何位置重新启动该时间监视功能。

如果循环程序超过最大循环时间，则操作系统会尝试启动时间错误 OB（例如，OB 280）。如果该时间错误 OB 不存在，CPU 将切换到“STOP”模式。

除了监视执行时间是否超过最大循环时间外，还要保证满足最小循环时间。为此，操作系统会延迟新循环的启动，直到达到了最小循环时间。在此等待时间内，将处理新的事件和操作系统服务。

如果在某些情况下（例如处理时间错误 OB 时）再次超过最大循环时间（2 次最大循环时间错误），CPU 将切换为 STOP 模式。

6.4.3.3 通信引起的周期负载

功能

CPU 的循环时间会因通信过程而延长。这些通信过程包括：

- 将数据传送到另一个 CPU
- 由编程设备启动的块装载

可使用 CPU 参数“由通信引起的周期负载”(Cycle load due to communication) 在一定程度上控制这些通信过程的持续时间。

除通信过程外，测试功能也可延长循环时间。“由通信引起的周期负载”(Cycle load due to communication) 参数对这些功能的持续时间几乎没有什么影响。

该参数的工作原理

使用“由通信引起的周期负载”(Cycle load due to communication) 参数，可输入 CPU 总处理能力中可用于通信过程的百分比。这部分 CPU 处理能力随即始终可用于通信。当通信不需要这部分处理能力时，便可供其它处理过程使用。

对实际循环时间的影响

“由通信引起的周期负载”(Cycle load due to communication) 参数可用于按某个因子延长循环组织块 (如 OB 1) 的循环时间，该因子的计算公式如下：

$$\frac{100}{100 - \text{“由通信引起的周期负载”}}$$

以上公式不考虑异步事件 (如硬件中断或循环中断) 对循环时间的影响。

如果循环时间因通过程而被延长，则在循环组织块的循环时间内可能会发生更多异步事件。这会进一步地延长周期。延长量取决于所发生的事件数量以及处理这些事件所需的时间。

实例 1 – 无附加异步事件：

如果将“由通信引起的周期负载”(Cycle load due to communication) 参数设置为 50%，则会导致循环组织块的循环时间按因子 2 增加。

实例 2 - 附加异步事件：

纯循环时间为 500 ms 时，假定 CPU 始终有足够的通信作业要处理，则 50% 的通信负载可使实际循环时间高达 1000 ms。如果与此同时，每隔 100 ms 执行一次处理时间为 20 ms 的循环中断，则在无通信负载时，该循环中断将使周期延长共 $5 * 20 \text{ ms} = 100 \text{ ms}$ 。即，实际循环时间将为 600 ms。由于循环中断也会中断通信，因此在有 50% 的通信负载时，它会使循环时间增加 $10 * 20 \text{ ms}$ 。即，此时的实际循环时间将达到 1200 ms，而不是 1000 ms。

注意

请注意以下说明：

- 请检查在系统运行期间更改“由通信引起的周期负载”(Cycle load due to communication) 参数的值所产生的影响。
 - 当设置最小循环时间时，必须始终考虑通信负载；否则，将发生时间错误。
-

建议

- 请仅在 CPU 主要用于通信且用户程序对时间要求不是很严格时，才增加该数值。
- 在所有其它情况下，只应当减小该值。

6.4.4 寻址模块

6.4.4.1 寻址模块

简介

在设备视图中，模块的地址或地址范围在“I 地址”(I-address) 和“Q 地址”(Q-address) 列中显示。此外，还有在下面介绍的其它地址。

I/O 地址

在用户程序中读取输入和/或设置输出时，需要 I/O 地址。

在将模块插入机架中时会自动分配输入和输出地址。第一个通道的地址是模块的起始地址。其它通道的地址皆源自此起始地址。结束地址源于模块特定的地址长度。

设备地址 (例如，以太网地址)

设备地址是指可编程模块的地址 (工业以太网地址)。它们用于对子网中的不同站点进行寻址，例如，为了将用户程序装载到某个 CPU 中。

用于标识模块或模块功能单元的硬件 ID

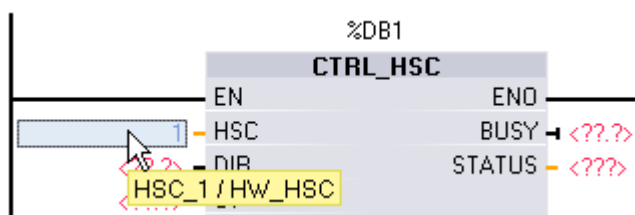
除 I 地址和 Q 地址外，还会自动分配硬件标识符用于标识模块。模块的功能单元 (例如，集成计数器) 也会接收到硬件标识符。

硬件标识符由整数组成，并与诊断报警一起由系统输出，以便能够定位故障模块或功能单元。

也可以将硬件标识符用于许多指令以标识将使用相应指令的相关模块。

硬件 ID 无法更改。

实例：标识高速计数器



在设备视图或网络视图以及 PLC 变量的常量表中插入组件时，将自动分配硬件 ID。同时还会自动为硬件 ID 分配名称。PLC 变量的常量表中的这些条目也无法更改。

参见

指定输入和输出的地址 (页 238)

将地址分配到程序中的某个位置 (页 238)

装载组态简介 (页 255)

6.4.4.2 指定输入和输出的地址

系统会自动设置默认的输入和输出的地址。但用户以后可以更改该地址分配。模块的所有地址都位于过程映像区中。过程映像将自动进行周期性更新。

要求

您处于设备视图中。

步骤

要更改预设的地址范围，请按以下步骤操作：

1. 在设备视图中，单击要为其设置起始地址的模块。
2. 转到巡视窗口中“属性”(Properties)的“I/O 诊断地址”(I/O diagnostics addresses)。
3. 在“起始地址”(Start address)下，输入所需的起始地址。
4. 按 <回车键> 或单击任一对象以接受修改的值。

如果输入的地址无效，将会显示一条消息，指示下一个可用地址。

注意

也可以在设备概览中直接更改地址。

参见

编辑属性和参数 (页 202)

6.4.4.3 将地址分配到程序中的某个位置

可将模块 I/O 通道的地址或符号直接分配到程序中的各个位置。

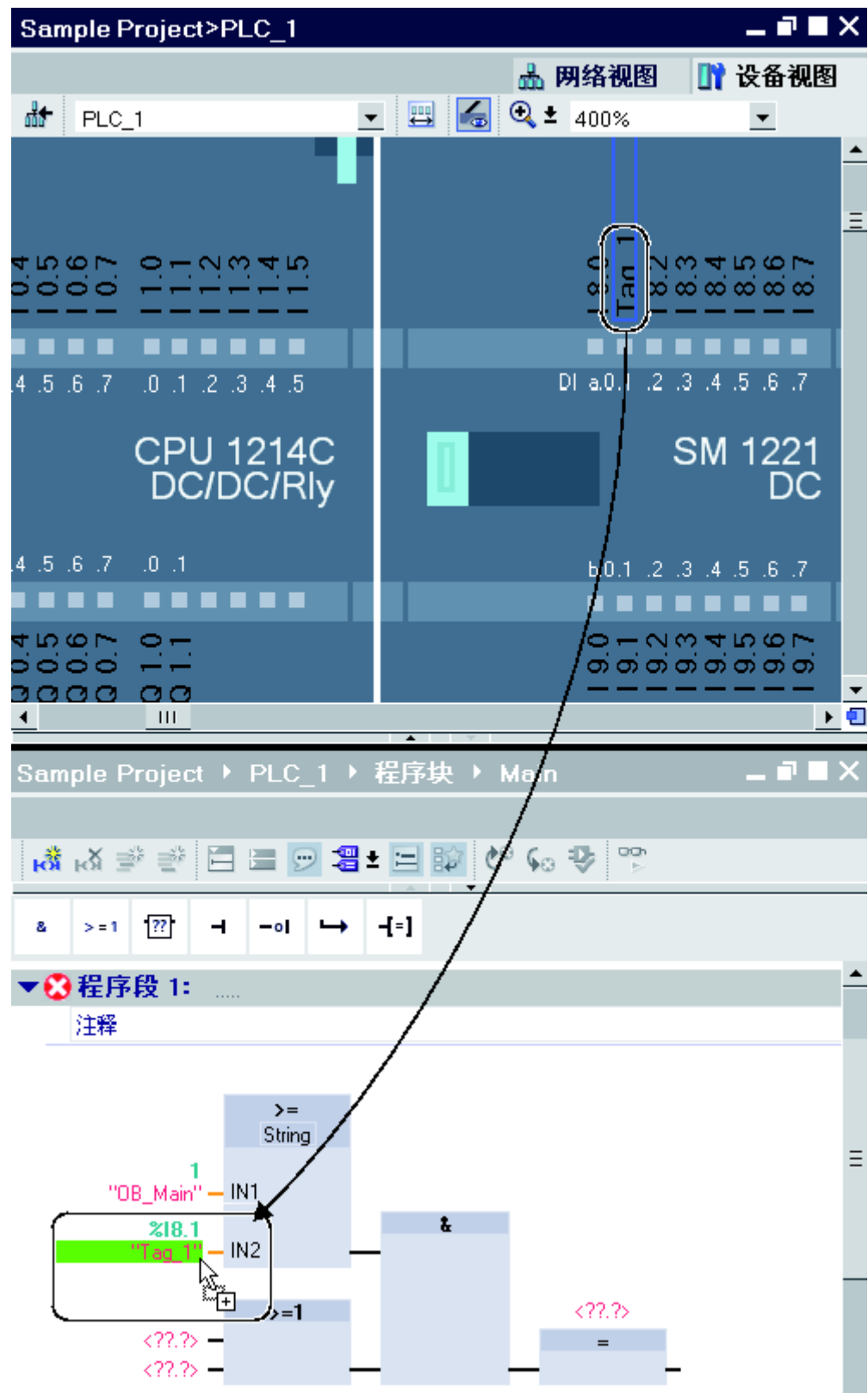
要求

必须打开硬件和网络编辑器的设备视图以及程序编辑器的指令窗口，并且依次上下进行排列。

步骤

要在程序中分配模块的地址或符号以及位置，请按以下步骤操作：

1. 在设备视图中，导航到具有所需 I/O 通道的模块。
2. 使用缩放功能以指定至少 200% 的放大倍数：在 200% 和更高的放大级别上，会显示各个地址通道的标签并且可进行编辑。
3. 在程序中，导航到具有匹配位置的块。
4. 按住鼠标键并将所需地址或图标拖动到该块的适当位置或模块的 I/O 通道。



即可将模块的地址或符号分配到程序中的相应位置，或者将程序中的地址或符号分配到模块的 I/O 通道。

6.4.5 日时钟功能

6.4.5.1 日时钟功能的基本原理

所有 S7-1200 CPU 都配有内部时钟。如果电源中断，备用电源可支持长达 10 小时的正确时间显示。

日时钟格式

时钟始终显示精度为 1 毫秒的日时钟以及包括星期在内的日期。同时还会考虑夏令时的时间调整。

6.4.5.2 设置和读取日时钟

通过系统功能设置日时钟

可使用以下系统功能来设置和启动 CPU 时钟的日时钟和日期：

- “SFC 0 SET_CLK”
- “SFC 100 SET_CLKS”，如果 CPU 中存在该系统功能

注意

要避免 HMI 系统中的日时钟显示差异，请将 CPU 设置为标准时间或使用中央日时钟同步。

通过系统功能读取日时钟

可使用系统功能“SFC 1 READ_CLK”或在编程设备中通过菜单命令读取 CPU 的当前日期和日时钟。

手动设置

也可以在在线和诊断视图的“功能 > 设置日时钟”(Functions > Set time of day) 下读取以及手动设置日时钟。

6.4.5.3 参数化时钟

时钟参数

通过时钟参数，可以进行以下设置：

- 启用使用 NTP 服务器的日时钟同步
如果要使用 NTP 同步模式同步内部时钟，则选中此复选框。
- 网络时间服务器
需要组态最多四个 NTP 服务器的 IP 地址。
- 更新间隔
更新间隔定义两次时间查询之间的时间间隔。

6.4.6 高速计数器

6.4.6.1 有关高速计数器的常规信息

简介

高速计数器通常用于驱动计数机制，在该机制中，恒速转动的轴配有一个增量式步编码器。该增量式步编码器可确保每转有一定数量的计数值以及每转一个复位脉冲。该增量式步编码器的时钟存储器位和复位脉冲为高速计数器提供输入。

不同的 S7-1200 CPU 具有不同数量的高速计数器：

S7-1200 CPU	HSC 数量	HSC 标识
CPU 1211C	3 (数字量信号板 4) *	HSC1...3 (以及 HSC5) *
CPU 1211C	4 (数字量信号板 5) *	HSC1...4 (以及 HSC5) *
CPU 1214C	6	HSC1...6

* 带 DI2/DO2 信号板

工作原理

多个默认值中的第一个将装载到高速计数器上。在计数器的当前值低于该默认值这段时间内，所需输出处于启用状态。计数器被设置成在计数器的当前值等于默认值或者计数器复位时发生中断。

如果当前值等于默认值并因此出现中断事件，则会装载新的默认值并为输出设置下一个信号状态。如果由于计数器复位而发生中断事件，则设置输出的第一个默认值和第一个信号状态，并重复该循环。

由于出现中断的次数远比高速计数器的计数次数少，因此只要稍微影响自动化系统的总周期，便可实现对快速操作的精确控制。用户也可为中断分配特定的中断程序，因此可以将每个新默认值装载到单独的中断程序中，从而实现简单的状态控制。

注意

也可以在一个中断程序中处理所有中断事件。

各种计数器的描述

所有计数器在相同计数器模式下的工作方式完全相同，但是，一些高速计数器并不支持所有计数模式。有四种基本计数器模式：

- 具有内部方向控制的单相计数器
- 具有外部方向控制的单相计数器
- 具有 2 个时钟输入的双相计数器
- A/B 计数器

每个高速计数器都可以使用或不使用复位输入。如果激活了复位输入，则该输入重置当前值。在禁用复位输入之前，当前值会一直保持重置状态。

参见

组态高速计数器 (页 243)
计数器模式和计数器输入的相互依赖性 (页 242)

6.4.6.2 计数器模式和计数器输入的相互依赖性

有关计数器模式和计数器输入的常规信息

用户不仅可以为高速计数器分配计数器模式和计数器输入，还可以为其分配一些功能，如时钟脉冲发生器、方向控制和复位等功能。以下规则适用：

- 一个输入不能用于两个不同的功能。
- 如果所定义的高速计数器的当前计数器模式不需要某个输入，则可将该输入用于其它用途。

例如，如果将 HSC1 设置为计数器模式 1，其中需要输入 I0.0 和 I0.3，则可将 I0.1 用于沿中断或用于 HSC2。

例如，如果设置 HSC1 和 HSC5，则计数和频率计数器模式总是会用到输入 I0.0 (HSC1) 和 I1.0 (HSC5)。因此，运行计数器时，以上两个输入不能用于任何其它功能。

如果使用数字信号板，则可使用一些附加输入。

计数器模式和计数器输入的相互依赖性概览

计数器模式	描述	输入		
	HSC1	I0.0 (CPU) I4.0 (信号板)	I0.1 (CPU) I4.1 (信号板)	I0.3 (CPU) I4.3 (信号板)
	HSC2	I0.2 (CPU) I4.2 (信号板)	I0.3 (CPU) I4.3 (信号板)	I0.1 (CPU) I4.1 (信号板)
	HSC3*	I0.4 (CPU)	I0.5 (CPU)	I0.7 (CPU)
	HSC4 (仅限 CPU 1212/14C)	I0.6 (CPU)	I0.7 (CPU)	I0.5 (CPU)
	HSC5 (仅限 CPU 1214C) **	I1.0 (CPU) I4.0 (信号板)	I1.1 (CPU) I4.1 (信号板)	I1.2 (CPU) I4.3 (信号板)
	HSC6 (仅限 CPU 1214C) **	I1.3 (CPU)	I1.4 (CPU)	I1.5 (CPU)
计数/频率	具有内部方向控制的单相计数器	时钟脉冲发生器		
计数		时钟脉冲发生器		复位
计数/频率	具有外部方向控制的单相计数器	时钟脉冲发生器	方向	
计数		时钟脉冲发生器	方向	复位
计数/频率	具有 2 个时钟输入的双相计数器	正向时钟脉冲发生器	反向时钟脉冲发生器	
计数		正向时钟脉冲发生器	反向时钟脉冲发生器	复位
计数/频率	A/B 计数器	时钟脉冲发生器 A	时钟脉冲发生器 B	
计数		时钟脉冲发生器 A	时钟脉冲发生器 B	复位
运动轴	脉冲发生器 PWM/PTO	在使用 PTO1 和 PTO2 脉冲发生器时，HSC1 和 HSC2 支持运动轴计数模式： <ul style="list-style-type: none"> • 对于 PTO1，HSC1 评估 Q0.0 输出来确定脉冲数。 • 对于 PTO2，HSC2 评估 Q0.2 输出来确定脉冲数。 Q0.1 用作运动方向输出。		

* HSC3 只能用于 CPU 1211，且没有复位输入

** 如果使用 DI2/DO2 信号板，则 HSC5 也可用于 CPU 1211/12

参见

有关高速计数器的常规信息 (页 241)

组态高速计数器 (页 243)

6.4.6.3 组态高速计数器

要求

已在硬件配置中插入了 S7-1200 CPU。

步骤

要组态高速计数器，请按以下步骤操作：

1. 在设备视图或网络视图中选择 S7-1200 CPU。
2. 在巡视窗口的“属性 > 高速计数器”(Properties > High-speed counter) 下，单击所需的高速计数器：
 - CPU 1211C：HSC1 到 HSC3 (以及带 DI2/DO2 信号板的 HSC5)
 - CPU 1212C：HSC1 到 HSC4 (以及带 DI2/DO2 信号板的 HSC5)
 - CPU 1214C：HSC1 到 HSC6
3. 在“常规”(General) 参数组中，选中相应复选框以启用高速计数器。

注意

如果使用带 DI2/DO2 信号板的 CPU 1211C 或 CPU 1212C，则还可以启用高速计数器 HSC5。

注意

如果激活脉冲发生器并以 PTO1 或 PTO2 方式运行，则它们会使用相关的高速计数器 HSC1 或 HSC2 与“运动轴”计数模式来评估硬件输出。如果组态高速计数器 HSC1 或 HSC2 用于其它计数任务，则脉冲发生器 PTO1 或 PTO2 将无法分别使用这两个计数器。

用户可根据需要在此输入高速计数器的名称和注释。

4. 在“功能”(Function) 参数组中，定义高速计数器的功能：
 - 计数模式：从下拉列表中选择要计数的对象。
 - 运行阶段：从下拉列表中选择计数算法。
 - 输入源：从下拉列表中选择板载 CPU 输入或可选数字信号板输入作为计数脉冲的输入源。
 - 指定计数方向的方式：如果选择了单相运行阶段，请打开下拉列表并选择是通过用户程序的 SFB 参数在内部设置计数方向，还是通过数字量输入在外部设置计数方向。
 - 初始计数方向：如果通过用户程序设置将计数方向设置为内部方向控制，则可从下拉列表中选择开始计数时的计数方向。
 - 频率计周期：如果计数模式设置为频率，则可从下拉列表中选择频率计周期的持续时间。
5. 在“重置为初始值”(Reset to initial values) 参数组中，指定高速计数器的初始值和复位条件：
 - 初始计数器值：输入高速计数器的起始值。
 - 初始参考值：输入高速计数器的最大值。

也可以在此指定高速计数器是否使用复位输入，并通过下拉列表为复位输入设置相应的信号电平。
6. 在“事件组态”(Event configuration) 参数组中，组态高速计数器对特定事件的响应。下列事件可以触发中断：
 - 计数器值与参考值一致。

- 生成了外部复位事件。
- 触发了方向变化。

使用该复选框启用中断响应，输入一个名称并从下拉列表中选择该中断对应的硬件中断。

7. 在“I/O/诊断地址”(I/O/Diagnostic addresses) 参数组中，为高速计数器分配起始地址。

注意

在“硬件输入”(Hardware inputs) 参数组中，用户只能看到用于时钟、方向确定、复位脉冲和最大计数速度的硬件输入和值。

结果

现在，已按照项目要求调整了高速计数器的参数。

参见

CTRL_HSC：在 FBD 中控制高速计数器 (页 775)
有关高速计数器的常规信息 (页 241)
计数器模式和计数器输入的相互依赖性 (页 242)

6.4.7 点对点通信

6.4.7.1 点对点通信概述

PtP 通信是一种通过串行接口使用标准化 UART (Universal Asynchronous Receiver/Transmitter，通用异步接收器/发送器) 数据传输的通信。S7-1200 使用带 RS-232 或 RS-485 接口的通信模块来建立 PtP 通信。

点对点通信的功能

点对点通信 (PtP) 的应用极为广泛：

- 将信息直接传输到外部设备，例如打印机或条码阅读器
- 接收来自外部设备 (如条码阅读器、RFID 阅读器、照相机、第三方光学系统以及许多其它设备) 的信息。
- 与第三方设备 (如 GPS 设备、无线调制解调器和许多其它设备) 进行信息交换

自由口协议

S7-1200 支持使用自由口协议进行基于字符的串行通信。该数据传输协议使用自由口通信，完全可通过用户程序进行组态。

Siemens 提供了具有自由口通信功能的库，可供用户在用户程序中使用：

- USS 驱动协议
- Modbus RTU 主站协议

- Modbus RTU 从站协议

参见

组态通信端口 (页 246)

6.4.7.2 使用 RS-232 和 RS-485 通信模块

带 RS-232 和 RS-485 接口的通信模块

在 S7-1200 CPU 中，可使用以下两种不同的通信模块：

- RS-232 通信模块
- RS-485 通信模块

以上通信模块可通过左侧的 I/O 通道连接到 S7-1200 CPU。最多可插入三个不同的模块。

通信模块的属性

通信模块具有以下特征：

- 支持自由口协议
- 通过用户程序借助扩展指令和库功能进行组态

6.4.7.3 组态通信端口

组态通信端口

插入具有 RS-232 或 RS-485 接口的通信模块之后，便可以设置接口参数。可在接口属性中设置接口参数，或者利用 PORT_CFG 指令在用户程序中控制接口参数。以下内容是关于图形组态的描述。

注意

如果通过用户程序更改端口设置，则图形组态的设置将被覆盖。

同时还应记住，如果发生断电，则通过用户程序进行的设置不会被保留。

要求

- 已插入通信模块。
- 您处于设备视图中。

步骤

要组态通信端口，请按以下步骤操作：

1. 在设备视图的图形表示中选择接口。

接口属性随即显示在巡视窗口中。

2. 在巡视窗口的区域导航中，选择“端口组态”(Port configuration) 组。
将显示端口的设置。
3. 在“波特率”(Baud rate) 下拉列表中，选择电缆的传输速度。对于用户自定义的通信，请注意传输速度对转换时间的影响。
4. 从“奇偶校验”(Parity) 下拉列表中，选择无效信息字的检测类型。
5. 通过“数据位”(Data bits) 下拉列表，确定一个字符是七位还是八位。
6. 在“停止位”(Stop bit) 下拉列表中，选择用来识别传输字结束的位数。
7. 在“流控制”(Flow control) 下拉列表中，选择用来确保发送方和接收方之间的数据流不出现错误的方法。只能为 RS-232 接口设置该参数。
 - 在“XON 字符”(XON character) 框中输入一个 HEX 值，当检测到该值时，将继续传输消息。只能为受软件控制的数据流控制设置该参数。
 - 在“XOFF 字符”(XOFF character) 框中输入一个 HEX 值，使消息传输暂停所设置的等待时间。只能为受软件控制的数据流控制设置该参数。
8. 在“等待时间”(Wait time) 框中，输入以 ms 为单位的等待时间，即在消息结束后下一次传输开始前必须暂停的时间。

注意

注意

也可在网络视图中组态接口。为此，必须先表格网络视图中选择通信模块，然后在巡视窗口中选择接口。随后便可按上述步骤继续操作。

参见

设置数据流控制 (页 247)

6.4.7.4 设置数据流控制

数据流控制

数据流控制是一种确保发送和接收行为保持平衡的方法。在理想情况下，智能控制可确保不会丢失数据。它确保设备发送的信息不会多于接收伙伴所能处理的信息。

有两种数据流控制方法：

- 硬件控制的数据流控制
- 软件控制的数据流控制

对于这两种方法，在传输开始时都必须激活通信伙伴的 DSR 信号。如果未激活 DSR 信号，则传输不会开始。

RS-232 通信模块可以处理这两种方法。RS-485 通信模块不支持数据流控制。

硬件控制的数据流控制

硬件控制的数据流控制采用请求发送 (RTS, Request To Send) 信号和允许发送 (CTS, Clear To Send) 信号。对于 RS-232 通信模块，RTS 信号通过输出引脚 7 进行传输，而 CTS 信号通过引脚 8 进行接收。

如果启用了硬件控制的数据流控制，则在发送数据时 RTS 信号总是设置为激活状态。同时，对 CTS 信号进行监视，以检查接收设备是否能接受数据。如果激活了 CTS 信号，则模块可以一直传输数据，直到 CTS 信号变为非激活状态。如果未激活 CTS 信号，则数据传输必须暂停所设置的等待时间。如果 CTS 信号在经过了所设置的等待时间后仍未激活，则数据传输将被中止，并向用户程序发送错误信号。

使用硬件握手的数据流控制

如果数据流控制由硬件握手进行控制，则默认情况下，发送设备将 RTS 信号设置为激活状态。这样，诸如调制解调器等设备便可随时传输数据。它不会等待接收方的 CTS 信号。发送设备通过只发送有限数量的帧（字符）来监视自身的传输，例如，防止接收缓冲区溢出。如果仍然出现溢出，则传送设备必须阻止消息并向用户程序发回错误信号。

软件控制的数据流控制

软件控制的数据流控制采用消息中的特定字符并通过这些字符来控制传输。这些字符是为 XON 和 XOFF 选择的 ASCII 字符。

XOFF 指示何时必须暂停传输。XON 指示何时可以继续传输。

如果发送设备接收到 XOFF 字符，它必须暂停发送所选的等待时间长度。如果在所选的等待时间之后发送了 XON 字符，则将继续传输。如果在等待时间之后未接收到 XON 字符，则将向用户程序发回错误信号。

因为接收伙伴需要在传输期间发送 XON 字符，所以软件数据流控制需要全双工通信。

参见

组态通信端口 (页 246)

6.4.7.5 组态消息传送

用户自定义的通信

可以使用您自己的机制来控制通信模块与通过串行接口连接的外部设备间的数据通信量。为此，您需要自行定义通信协议。在用户自定义的通信中，支持使用 ASCII 协议和二进制协议进行消息传送。

在通信协议中，需要指定用来在数据流中识别所传送消息开始和结束的条件。

只能在 RUN 模式下激活用户自定义的通信。如果切换到 STOP 模式，将停止用户自定义的通信。

指定通信协议

可按以下方式指定通信协议：

- 使用用户程序
 - 发送数据时的行为由 SEND_CFG 指令控制。
 - 接收数据时的行为由 RCV_CFG 指令控制。
- 使用巡视窗口中以图形方式设置的参数设置

注意

如果通过用户程序更改通信协议，则图形组态的设置将被覆盖。

请记住，如果发生断电，通过用户程序进行的设置不会被保留。

参见

有关 RS-232 设备的用户自定义通信 (页 249)

进行发送设置 (页 250)

指定消息开始 (页 251)

指定消息结束 (页 252)

6.4.7.6 有关 RS-232 设备的用户自定义通信**有关 RS-232 设备的 RS-232/PPI 多主站编程电缆和用户自定义通信**

使用 RS-232/PPI 多主站编程电缆和用户自定义通信，可以将各种 RS-232 兼容设备连接到 S7-1200 的通信模块。但是，必须将该电缆设置为“PPI/用户自定义通信”模式。

电缆设置

必须如下设置电缆上的开关：

- 开关 5 必须设为 0
- 开关 6 设置为本地模式 (DCE) 或远程模式 (DTE)：
 - 开关设置为 0 时，代表本地模式
 - 开关设为 1 时，代表远程模式

在发送模式和接收模式间切换

将数据从 RS-232 接口发送到 RS-485 接口时，RS-232/PPI 多主站编程电缆处于发送模式。当电缆空闲或数据从 RS-485 接口发送到 RS-232 接口时，电缆处于接收模式。当电缆在 RS-232 发送线路上检测到字符时，它会立即从接收模式切换到发送模式。

支持的波特率

RS-232/PPI 多主站编程电缆支持 1200 波特到 115.2 千波特之间的波特率。使用 PC/PPI 电缆上的 DIP 开关，可将 RS-232/PPI 多主站编程电缆设置为所需的波特率。

下表给出了各种波特率所对应的开关设置：

波特率	切换时间	设置 (1 为向上)
115200	0.15 ms	110
57600	0.3 ms	111
38400	0.5 ms	000
19200	1.0 ms	001
9600	2.0 ms	010
4800	4.0 ms	011
2400	7.0 ms	100
1200	14.0 ms	101

当 RS-232 发送线路空闲的时间达到所定义的电缆切换时间后，电缆将重新切换回接收模式。所设置的波特率对切换时间的影响如上表所示。

切换时间的影响

在采用用户自定义通信的系统中使用 RS-232/PPI 多主站编程电缆时，程序必须考虑切换时间，原因如下：

- 通信模块会对 RS-232 设备发送的消息做出响应。
通信模块一旦接收到来自 RS-232 设备的请求，它就必须将响应消息延迟一段时间，该时间段等于或大于电缆的切换时间。
- RS-232 设备会对通信模块发送的消息做出响应。
通信模块一旦接收到来自 RS-232 设备的响应消息，它就必须将下一个请求消息延迟一段时间，该时间段等于或大于电缆的切换时间。

在以上两种情况下，RS-232/PPI 多主站编程电缆都应当有足够的时间从发送模式切换到接收模式，以便数据能够从 RS-485 接口发送到 RS-232 接口。

参见

组态消息传送 (页 248)
进行发送设置 (页 250)
指定消息开始 (页 251)
指定消息结束 (页 252)

6.4.7.7 进行发送设置

发送消息

可以在各个消息之间设置中断。

下表列出了可以设置的中断：

参数	定义
RTS 接通延时	可以设置从发送请求 RTS (Request To Send , 请求发送) 后到实际数据传输开始前必须经过的时间。
RTS 关断延时	可以设置从传输完成后到 RTS 信号被禁用前必须经过的时间。
在消息开始时发送中断	可以指定当经过了 RTS 接通延时后, 在每次消息传输开始时发送中断。以位时间为单位指定该中断。
发送中断后线路空闲信号	您可以设置在经过消息开始的所选中断过后, 输出“线路空闲”信号, 以指示当前未使用线路。要启用该参数, 必须设置“在消息开始时发送中断”(Send pause at message start)。 “线路空闲”信号的持续时间以位时间为单位。

参见

指定消息开始 (页 251)

指定消息结束 (页 252)

有关 RS-232 设备的用户自定义通信 (页 249)

6.4.7.8 指定消息开始

识别消息开始

要在消息传送完成时以及下一次消息传输开始时向接收方发送信号, 必须在传输协议中指定用来识别消息开始和结束的条件。

如果满足了用来指示消息开始的条件, 则接收方将开始在数据流中搜索表示消息结束的条件。

有以下两种不同的方法可用来识别消息开始:

- 以任意字符开始:
任何字符都可以定义消息开始。这是默认方法。
- 以特殊条件开始:
基于所选条件确定消息开始。

用于检测消息开始的条件

下表列出了用于定义消息开始的各种选项:

参数	定义
通过线路中断识别消息开始	如果接收的数据流中断超过一个字符时间, 接收方将识别线路中断。如果是这种情况, 说明是通过线路中断识别消息开始。
通过线路空闲识别消息开始	如果发送传输线路在空闲一段时间 (该时间以位时间为单位) 后发生诸如接收字符之类的事件, 将识别到消息开始。
通过单个字符识别消息开始	在出现某个特定字符时识别到消息开始。可以 HEX 值的形式输入该字符。

参数	定义
通过字符串识别消息开始	当指定字符序列之一到达数据流中时，将检测到消息开始。可以指定最多四个字符序列，每个序列最多包含五个字符。

各个条件可以按任意方式进行逻辑组合。

参见

进行发送设置 (页 250)

有关 RS-232 设备的用户自定义通信 (页 249)

6.4.7.9 指定消息结束

识别消息结束

要在消息传送完成时以及下一次消息传输开始时向接收方发送信号，必须在传输协议中指定用来识别消息开始和结束的条件。

一共有六种不同的方法可用来识别消息结束，这些方法可以按任意方式进行逻辑组合。下表列出了各种可能的设置选项：

参数	定义
通过消息超时识别消息结束	如果消息超出所选的最长持续时间，将自动识别到消息结束。可设置的值为 0 到 65535 ms。
通过响应超时识别消息结束	如果在传送数据后的设定时间内没有响应，将识别到消息结束。可设置的值为 0 到 65535 ms。
通过字符间超时识别消息结束	如果超过传送两个字符之间的时间（单位为位时间），将检测到消息结束。可设置的值为 0 到 2500 位时间。 即使所设置的值导致持续时间超过八秒，S7-1200 CPU 也只接受最长八秒的时间。
通过最大长度识别消息结束	如果超过消息的最大长度，将识别到消息结束。可设置的值为 1 到 4132 个字符。

参数	定义
读取消息中的消息长度	<p>消息本身包含有关消息长度的信息。如果达到了消息中的长度值，则表明该消息结束。通过以下参数可以指定用来计算消息长度的字符：</p> <ul style="list-style-type: none"> • 消息中长度域的偏移量 该值决定用来指示消息长度的字符在消息中所处的位置。 可设置的值为 0 到 4131 个字符。 • 长度域大小 该值指定从第一个评估位置开始，使用多少个字符来指示消息长度。 可设置的值为 0、1、2 和 4 个字符。 • 长度域后面的数据 (不属于消息长度) 该值指定在计算消息长度时必须忽略的长度域后的字节数。 可设置的值为 0 到 255 个字符。
通过字符串识别消息长度	当指定的字符序列到达数据流中时，将检测到消息结束。可以定义字符串中最多有五个字符。

参见

进行发送设置 (页 250)

有关 RS-232 设备的用户自定义通信 (页 249)

6.4.8 使用时钟存储器

时钟存储器

时钟存储器是一个按 1:1 的占空比周期性改变其二进制状态的位存储器。

在分配时钟存储器参数时，需要指定要用作时钟存储器字节的 CPU 存储器字节。

优点

例如，可以使用时钟存储器来激活闪烁指示灯或启动周期性的重复操作（如记录实际值）。

可用频率

时钟位存储器字节的每一位均分配有一个频率。下表列出了分配情况：

时钟存储器字节的位	7	6	5	4	3	2	1	0
周期 (s)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
频率 (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

注意

时钟存储器的运行与 CPU 周期不同步，即，时钟存储器的状态在一个长周期内可以改变多次。选定的存储器字节不能用于存储中间数据。

6.4.9 保护等级设置选项

保护等级

在下列各部分中，您将学习如何使用 CPU 的各种保护等级。

保护等级设置的作用

可以选择以下保护等级：

- 无保护：对应于默认行为。用户无法输入密码。总是允许进行读写访问。
- 写保护：只能进行读访问。无法更改 CPU 上的任何数据，也无法装载任何块或组态。已标记为与 HMI 相关的诊断数据和 PLC 变量不属于写保护范围。
- 读/写保护：对于“可访问设备”区域或项目中已切换到在线状态的设备，无法进行写访问或读访问。只有 CPU 类型和标识数据可以显示在项目树中“可访问设备”(Accessible devices) 下。除非输入密码，否则无法显示“可访问设备”(Accessible devices) 或项目中已切换到在线状态的设备的在线信息或块。

读/写保护不包括以下对象：

- 标记为 HMI 相关的 PLC 变量
- HMI 对象的可监视属性
- 诊断数据 (仅读保护)

操作期间受密码保护的 CPU 的行为

在执行在线功能之前，将检查所需的权限，必要时将提示用户输入密码。

实例：模块已分配了保护等级，用户想要执行“修改变量”功能。这需要写访问权限；因此，必须输入所分配的密码，才能执行该功能。

受密码保护的功能在任何时刻都只能由一个编程设备/PC 执行。其它编程设备/PC 无法使用密码进行登录。

注意

用户可以不受限制地使用过程控制、监视以及通信功能。因此，诸如“设置日时钟/日期”这样的功能不应该用密码锁定。

某些功能由于要用作在线数据，因而仍需受到保护。因此，“在线工具”(Online Tools) 任务卡中的 RUN/STOP 或诊断和在线编辑器中的“设置日时钟”(Set time of day) 功能受到写保护。

6.4.10 装载组态

6.4.10.1 装载组态简介

在设备上装载组态

在项目中插入新设备并对其进行组态之后，或者更改了现有硬件配置后，下一步就是在设备上装载当前组态。这样可确保编程设备/PC 的组态总是与物理模块上的组态保持一致。

首次装载时会装载全部硬件项目数据。以后再装载时，将只装载组态的更改内容。

装载硬件配置时，用户有下列选择：

- 在设备或网络视图中装载
- 在项目树中装载
- 在可访问设备上装载

参见

启动期间的特性 (页 291)

6.4.10.2 在设备上装载硬件配置

可以在项目树中将设备的硬件配置装载到相关设备上。



警告 只能在 STOP 模式下装载

如果参数设置不正确，则在装载之后，可能会出现意外的机器行为或过程行为。为了避免出现可能的设备损坏或人员受伤，在执行装载操作时必须将 CPU 设置为 STOP 模式。

在项目树中将多个块装载到设备上

要在相关设备上装载硬件配置，请按以下步骤操作：

1. 在设备的快捷菜单中，选择“装载到设备 > 硬件配置”(Load to device > Hardware configuration) 命令。
 - 如果之前尚未建立在线连接，则会打开“扩展装载”(Extended loading) 对话框。
 - 如果已定义在线连接，则会打开“装载预览”(Load preview) 对话框。然后，继续执行步骤 4。
2. 从“扩展装载”(Extended loading) 对话框的“用于装载过程的编程设备/PC 接口”(Programming device/PC interface for loading process) 下拉列表框中，选择 PG/PC 接口。
3. 在“目标子网中的可访问设备”(Accessible devices in the target subnet) 表中选择设备并使用“确定”(OK) 确认选择。
“装载预览”(Load preview) 对话框将打开。
4. 单击“装载”(Load)。

将装载硬件配置，并且将打开“装载结果”(Load results) 对话框。此对话框会显示装载后的状态和操作。

5. 单击“确定”(OK) 关闭“装载结果”(Load results) 对话框。

结果

现在，编程设备/PC 上组态的硬件配置与设备上的硬件配置完全相同。

巡视窗口中“信息 > 常规”(Info > General) 下的消息将报告装载过程是否成功。

参见

启动期间的特性 (页 291)

6.5 诊断硬件

6.5.1 硬件诊断概述

6.5.1.1 硬件诊断的主要方法

硬件诊断的主要方法

可按如下方式执行硬件诊断：

- 使用在线和诊断视图
- 使用“在线工具”(Online Tools) 任务卡
- 使用巡视窗口的“诊断 > 设备信息”(Diagnostics > Device Info) 区域
- 使用诊断符号，例如，在设备视图和项目树中

在线和诊断视图的结构

在线和诊断视图由两个并排的窗口组成：

- 左侧窗口显示一个包含文件夹和组（打开文件夹时显示）的树结构。
- 右侧窗口包含有关所选文件夹或所选组的详细信息。

其中包括“在线访问”(Online access) 组以及“诊断”(Diagnostics) 和“功能”(Functions) 文件夹：

- “在线访问”(Online access) 组：显示当前是否存在与相关目标的在线连接。另外，可以建立或断开在线连接。
- “诊断”(Diagnostics) 文件夹：包含所选模块的多个诊断组
- “功能”(Functions) 文件夹：包含多个组，可以在其中对所选模块进行设置或向该模块发送命令。

“在线工具”任务卡的功能和结构

对于本身具有工作模式的模块（例如 CPU），通过“在线工具”(Online Tools) 任务卡可以读取模块的当前诊断信息及命令。

如果选择了本身不具有工作模式的模块，或者在激活“在线工具”(Online Tools) 任务卡之前选择了多个模块，则该任务卡仅与相应的 CPU 有关。

“在线工具”(Online Tools) 任务卡由以下窗格组成：

- CPU 控制面板
- 循环时间
- 存储器

注意

只有当模块控制着相关功能并且存在在线连接时，窗格才会填充有内容。

如果没有到相应模块的在线连接，则“无在线连接”(No online connection) 将以蓝色显示。

巡视窗口的结构或“诊断”选项卡

巡视窗口的“诊断”(Diagnostics) 选项卡本身包含一个选项卡：

- 设备信息

该选项卡与项目中建立了在线连接的所有 CPU 相关。如果一个或多个 CPU 有故障或未处于 RUN 模式，将在此处报告报警。

参见

任务卡 (页 113)

巡视窗口 (页 111)

6.5.1.2 确定哪些在线连接的设备有故障

故障设备概述

在巡视窗口的“诊断 > 设备信息”(Diagnostics > Device Info) 区域中，用户可以从总体上了解在线连接的故障设备。

巡视窗口的“诊断 > 设备信息”(Diagnostics > Device Info) 区域由以下元素组成：

- 包含故障设备数量的标题行
- 包含各故障设备详细信息的表格

如果与无法访问的设备、报告存在一个或多个故障的设备或未处于 RUN 模式的设备建立了在线连接，这些连接设备也将被视为故障设备。

包含各故障设备详细信息的表格的结构

该表由以下各列组成：

- 在线状态：包含诊断符号形式和文字形式的在线状态
- 工作模式：包含符号形式和文字形式的工作模式
- 设备/模块：受影响设备或模块的名称
- 消息：解释前面的列
- 详细资料：该链接可打开设备的在线和诊断视图，并将其置于最顶层。
- 帮助：该链接提供有关所发生故障的更多信息。

参见

巡视窗口 (页 111)

确定在线状态并用符号显示 (页 258)

6.5.1.3 确定在线状态并用符号显示

确定在线状态并用符号显示

与设备建立在线连接时，还会同时标识和显示相关模块的状态和工作模式（如有必要）。

在下列视图中进行显示：

- 设备视图
除了 CPU 上的信号板外，每个硬件组件的诊断图标都会在此显示。对于本身具有工作模式的模块，同时还会显示工作状态图标。
- 设备概览
每个硬件组件的诊断图标都会在此显示。
- 网络视图
在此显示各个设备的诊断符号，以此显示所有相关硬件组件的总体状态。
- 网络概览
每个硬件组件的诊断图标都会在此显示。
- 项目树
其中，诊断符号显示在硬件组件后面。对于本身具有工作模式的模块，还会在模块图标的右上角显示工作状态图标。
- 在线和诊断视图
在线状态显示在“在线访问”(Online access) 组的“状态”(Status) 区域。也可以在此处建立和终止在线连接。









注意

只有 CPU 具有“在线访问”(Online access) 组。然而，如果使用“显示/更新可访问设备”功能调用了在线和诊断视图，该视图将不会显示。

- 项目树中的详细视图
该步骤要求您在项目树中选择了“本地模块”(Local modules)。





模块和设备的诊断符号

下表列出了可用的符号及其各自的含义：

符号	含义
	当前正在建立到 CPU 的连接。
	无法通过所设置的地址访问 CPU。
	组态的 CPU 和实际 CPU 型号不兼容。
	无故障
	要求维护
	维护请求
	错误
	无法从 CPU 访问模块或设备（指 CPU 下面的模块和设备）。
	由于当前在线组态数据与离线组态数据不同，因而无法获得诊断数据。
	组态的模块或设备与实际的模块或设备不兼容（指 CPU 下面的模块或设备）。
	建立了连接，但尚未确定模块的状态。
	组态的模块不支持诊断状态显示



比较状态的图标

诊断图标可与其它用来指示在线/离线比较结果的小图标组合使用。下表列出了可用的符号及其含义。

符号	含义
	文件夹包含在线和离线版本不同的对象（仅在项目树中）
	对象的在线和离线版本不同
	对象仅在线存在
	对象仅离线存在

组合的诊断与比较图标

下表给出了组合图标及其含义的实例。

符号	含义
	无故障。文件夹包含在线和离线版本不同的对象 (仅在项目树中)
	无法从 CPU 访问模块或设备 (指 CPU 下面的模块和设备)。对象仅离线存在

CPU 和 CP 的工作模式符号

下表列出了可用的符号及其各自的工作状态。

符号	工作模式
	RUN
	STOP
	STARTUP
	HOLD
	DEFECTIVE
	未知工作模式
	组态的模块不支持工作模式显示

6.5.1.4 启动在线和诊断视图

要求

包含要诊断模块的项目已打开。

注意

如果您在标识了可访问设备后从项目树调用在线和诊断视图，则该要求不适用。

步骤

要启动模块的在线和诊断视图，请按以下步骤操作：

1. 在项目树中，打开相应的设备文件夹。
2. 双击“在线和诊断”(Online & Diagnostics) 功能。

注意

如果项目树中的模块拥有子条目（例如在 PLC 中），则“在线和诊断”(Online & Diagnostics) 功能始终可用。

或者：

1. 在项目树中，选择相应的设备文件夹。
2. 在快捷菜单或“在线”(Online) 主菜单中，选择“在线和诊断”(Online & Diagnostics) 命令。

或者：

1. 在设备组态中，打开设备视图。
2. 选择要诊断的模块。
3. 在快捷菜单或“在线”(Online) 主菜单中，选择“在线和诊断”(Online & Diagnostics) 命令。

或者：

1. 在设备组态中，打开网络视图。
2. 选择要诊断的模块所在的站。
3. 在快捷菜单或“在线”(Online) 主菜单中，选择“在线和诊断”(Online & Diagnostics) 命令。

或者：

1. 在项目树中，打开“在线访问”(Online access) 文件夹。
2. 打开要用来建立在线连接的接口所在的文件夹。
3. 双击“显示/更新可访问设备”(Show/Update accessible devices)。
4. 选择要诊断的模块。
5. 在快捷菜单或“在线”(Online) 主菜单中，选择“在线和诊断”(Online & Diagnostics) 命令。

或者：

1. 在项目树中，打开“本地模块”(Local modules) 文件夹。
2. 选择要诊断的相应设备或模块。
3. 在快捷菜单或主菜单中，选择“在线和诊断”(Online & Diagnostics) 命令。

结果

将启动要诊断模块的在线和诊断视图。如果先前已创建了到相关 CPU 的在线连接，则在线和诊断视图标题栏此时将具有橙色背景。

注意

如果在启动在线和诊断视图时不存在任何在线连接，将不显示任何在线信息，并且显示域将空白。

6.5.1.5 激活“在线工具”任务卡

激活“在线工具”任务卡

可按以下方式激活该任务卡：

1. 启动在线和诊断视图。
2. 单击“在线工具”(Online Tools) 任务卡。

或者：

1. 启动设备视图。
2. 单击“在线工具”(Online Tools) 任务卡。

或者：

1. 启动网络视图。
2. 单击“在线工具”(Online Tools) 任务卡。

6.5.2 显示可组态模块属性的不可编辑值和当前值

6.5.2.1 显示模块的常规属性和系统相关信息

在何处查找所需的信息？

在要诊断模块的在线和诊断视图中“诊断”(Diagnostics) 文件夹的“常规”(General) 组中，可以找到模块的常规属性和系统相关信息。

“常规”组的结构

“常规”(General) 组由以下各区域组成：

- 模块 (Module)
- 模块信息 (Module information)
- 制造商信息 (Manufacturer information)

“模块”区域

该区域显示模块的以下数据：

- 简短标识，例如 CPU 1214C DC/DC/DC
- 订货号
- 硬件
- 固件
- 机架
- 插槽

“模块信息”区域

该区域显示用户在硬件配置期间指定的以下模块规范：

- 模块名称

“制造商信息”区域

该区域显示模块的以下数据：

- 制造商

注意

除 Siemens 公司之外,制造商代码均以十进制数显示。相关的制造商名称可以在 PROFIBUS International 的制造商 ID 表格中找到 (请访问“www.profibus.com”)。

- 序列号
- 配置文件：配置文件 ID 为十六进制数

注意

可以在 PROFIBUS International 的配置文件 ID 表格中找到相应的配置文件名称 (请访问“www.profibus.com”)。

- 配置文件详细信息：配置文件特定的类型为十六进制数

注意

可以在 PROFIBUS International 的配置文件特定类型表格中找到相应的配置文件特定类型名称 (请访问“www.profibus.com”)。

6.5.2.2 显示组态的循环时间

在何处查找所需的信息？

可以在以下位置找到所需信息：

- 在要诊断模块的在线和诊断视图中“诊断”(Diagnostics) 文件夹的“循环时间”(Cycle time) 组中 (此处显示所有已分配的循环时间)。
- 在“在线工具”(Online Tools) 任务卡的“循环时间”(Cycle time) 窗格中 (此处只显示已组态循环时间的最大循环时间)。

在线和诊断视图中“诊断”文件夹下“循环时间”组的结构

“循环时间”(Cycle time) 组由以下各区域组成：

- 循环时间图 (测量循环时间的图形显示)

- 组态的循环时间 (以绝对值形式显示组态的循环时间)
- 测量的循环时间 (以绝对值形式显示测量的循环时间)

参数化的循环时间

显示以下组态的时间：

- 最小循环时间
- 最大循环时间

“在线工具”任务卡中“循环时间”窗格的结构

“循环时间”(Cycle time) 窗格提供有关组态的最大循环时间以及测量循环时间的信息。

最大循环时间

时间图结尾的值代表最大循环时间。

6.5.3 显示动态模块属性的当前值

6.5.3.1 显示测量循环时间

在何处查找所需的信息？

可在以下各位置找到测量循环时间：

- 在要诊断模块的在线和诊断视图中“诊断”(Diagnostics) 文件夹下的“循环时间”(Cycle time) 组中
- 在“在线工具”(Online Tools) 任务卡的“循环时间”(Cycle time) 窗格中

在线和诊断视图中“诊断”文件夹下“循环时间”组的结构

“循环时间”(Cycle time) 组由以下各区域组成：

- 循环时间图 (测量循环时间的图形显示)
- 组态的循环时间 (以绝对值形式显示组态的循环时间)
- 测量的循环时间 (以绝对值形式显示测量的循环时间)

测量循环时间

显示以下时间：

- 最短循环时间：自上次从 STOP 切换到 RUN 以来的最短循环的持续时间。
该时间在图形显示中以黑色箭头显示。
- 当前/上次循环时间：上个循环的持续时间
该时间在图形显示中以绿色箭头显示。

如果上次循环的持续时间接近最大循环时间，这表示有可能会超过最大循环时间并且模块可能会切换到 STOP 工作模式。例如，如果正在监视程序中的变量，这会使循环时间延长。

- 最长循环时间：自上次从 STOP 切换到 RUN 以来的最长循环的持续时间。

该时间在图形显示中以蓝色箭头显示。

注意

图形显示

时间轴由参数化的循环时间限值来限定。

红色箭头和红色条表示超出参数化限值的时间。

“在线工具”任务卡中“循环时间”窗格的结构

“循环时间”(Cycle time) 窗格提供有关组态的最大循环时间以及测量循环时间的信息。

测量的时间

显示以下测量的时间：

- 自上次激活“在线工具”(Online Tools) 任务卡以来的最短循环时间，显示为灰色。
- 当前/上次循环时间，显示为绿色。
- 自上次激活“在线工具”(Online Tools) 任务卡以来的最长循环时间，显示为蓝色。

6.5.3.2 显示 CPU 上 LED 的当前状态

在何处查找所需的信息？

CPU 上 LED 的当前状态可在“在线工具”(Online tools) 任务卡中“CPU 控制面板”(CPU control panel) 窗格的显示区域找到。

“在线工具”任务卡中“CPU 控制面板”窗格的显示区域

该区域包含以下显示内容：

- 站名称和 CPU 类型 (简短标识)
- RUN/STOP (对应于 CPU 的“RUN/STOP”LED)
- ERROR (对应于 CPU 的“ERROR”LED)
- MAINT (对应于 CPU 的“MAINT”LED)

6.5.3.3 显示 CPU 上所有存储器的使用率

在何处查找所需的信息？

可在以下两个页面中找到 CPU 上所有存储器的使用率：

- 在要诊断模块的在线和诊断视图中“诊断”(Diagnostics) 文件夹下“存储器”(Memory) 组的显示区域。
- 在“在线工具”(Online Tools) 任务卡的“存储器”(Memory) 窗格显示区域。

在线和诊断视图中“诊断”文件夹下“存储器”组的显示区域

该区域包含了相应模块的当前存储器使用情况以及各存储区的详细信息。

存储器使用情况以棒图和数字值 (百分数) 两种形式显示。

显示以下存储器使用情况：

- 装载存储器
如果未插入任何存储卡，则显示内部装载存储器。
如果插入了存储卡，则操作系统将仅使用所插入的装载存储器作为装载存储器。并在此处显示插入的装载存储器。
- 工作存储器
- 保持性存储器

“在线工具”任务卡中“存储器”窗格的显示区域

该区域包含模块的当前存储器使用情况。可用的存储空间以棒图和数字值 (百分数) 两种形式显示。数字值将舍入为整数值。

注意

如果存储区的利用率小于 1%，则该存储区的可用部分将显示为“99%”。

显示以下存储器使用情况：

- 装载存储器
如果未插入任何存储卡，则显示内部装载存储器。
如果插入了存储卡，则操作系统将仅使用所插入的装载存储器作为装载存储器。并在此处显示插入的装载存储器。
- 工作存储器
- 保持性存储器

参见

装载存储器 (页 293)
工作存储器 (页 293)
保持性存储区 (页 295)

6.5.4 检查模块故障

6.5.4.1 确定模块的诊断状态

何处显示了模块的诊断状态？

模块的诊断状态显示在要诊断模块的在线和诊断视图中“诊断”(Diagnostics) 文件夹的“诊断状态”(Diagnostic status) 组中。

“诊断状态”组

在“诊断状态”(Diagnostic status) 组中会显示以下状态信息：

- 可供 CPU 查看的模块状态，例如：
 - 模块可用且正常。
 - 模块有故障。
如果模块出现故障并且用户在组态期间启用了诊断中断，将显示“模块有故障”状态。
 - 模块已组态，但不可用。
- 检测到组态的模块与插入的模块之间有差异。如果能够确定，将显示设置型号与实际型号的订货号。

显示信息的范围

显示信息的范围取决于所选的模块。

6.5.4.2 读出 CPU 的诊断缓冲区

在何处读出 CPU 的诊断缓冲区？

可从在线和诊断视图中“诊断”(Diagnostics) 文件夹的“诊断缓冲区”(Diagnostics buffer) 组中读出 CPU 的诊断缓冲区。

“诊断缓冲区”组的结构

“诊断缓冲区”(Diagnostics buffer) 组由“事件”(Events) 选项卡组成。

诊断缓冲区

诊断缓冲区用作记录诊断事件的日志文件，这些诊断事件是在分配给它的 CPU 和模块上发生的。这些事件按其发生的顺序输入，并且最新的事件显示在顶部。

“事件”选项卡

“事件”(Events) 选项卡由以下元素组成：

- “包括 CPU/当地时差的时间”复选框
- 事件列表

- 事件的详细信息
- “冻结视图/取消冻结”(Freeze view/Cancel freeze)、“关于事件的帮助”(Help on event)、“打开块”(Open block)、“另存为...”(Save as...) 按钮。

“包括 CPU/当地时差的时间”复选框

如果尚未选中该复选框，将使用模块时间显示诊断缓冲区条目。

如果选中了该复选框，将使用按以下公式计算的时间显示诊断缓冲区条目：

显示的时间 = 模块时间 + 编程设备/PC 上的时区偏移量

这要求模块时间与 UTC 时间一致。

如果要查看以编程设备/PC 的本地时间表示的模块诊断缓冲区条目时间，则应使用该设置。

选中或取消选中该复选框会立即更改诊断缓冲区条目的显示时间。

注意

如果在程序中使用“WR_SYS_T”指令，或者使用 HMI 设备而非 UTC 来设置 CPU 的实时时钟，我们建议您取消选中“实时时钟包括 CPU/本地时差”(Real-time incl. CPU/local time difference) 复选框。在这种情况下，模块时间是唯一有意义的时间。

事件列表

对于每个诊断事件，都会列出以下信息：

- 条目的顺序号
第一个条目包含最新的事件。
- 诊断事件的日期和时间
如果未显示日期和时间，则表明模块没有集成时钟。
- 事件的短名称以及 CPU 的响应（如果适用）

注意

如果无法确定文本的个别参数，则相应参数位置将显示字符串“###”。

如果还没有可用于新模块或新事件的显示文本，事件编号和各个参数将以十六进制值表示。

事件的详细信息

如果在列表中选择一行，即可获得相应事件的详细说明：

- 事件 ID 和事件名称
- 与事件相关的其它信息，如引起事件的命令的地址以及由诊断事件引起的工作模式切换
- 进入事件或离开事件

“冻结视图”或“取消冻结”

只有存在到 CPU 的在线连接时，才会启用“冻结视图”(Freeze view) 或“取消冻结”(Cancel freeze) 按钮。

默认设置为“冻结视图”(Freeze view)。

如果单击“冻结视图”(Freeze view) 按钮，将出现以下情况：

- 诊断缓冲区条目的当前显示将被冻结。
- 按钮标签变为“取消冻结”(Cancel freeze)。

如果系统中出现错误，可能会非常快速并连续发生诊断事件。如此会使显示的更新速率加快。通过冻结显示，用户便可冷静而详细地检查所出现的情况。

如果冻结了显示然后单击“取消冻结”(Cancel freeze) 按钮，将发生以下情况：

- 诊断缓冲区条目的显示会再次更新。
- 按钮标签变为“冻结视图”(Freeze view)。

注意

如果冻结了诊断缓冲区显示，CPU 会继续在诊断缓冲区中输入事件。

“关于事件的帮助”按钮

如果单击此按钮，将更详细地说明所选事件并给出所有解决方法。

注意

如果所选事件不是 CPU 事件，则“关于事件的帮助”(Help on event) 按钮将不可用。

“打开块”按钮

仅当诊断事件中存在指向块相对地址的指针时，才能选择“打开块”(Open block) 按钮。这是引起事件的命令的地址。

“打开块”功能用于在离线视图中导致错误的程序指令位置打开被引用块。这样，用户就可以检查并根据需要更改该指定位置的块的源代码，然后将其再次下载到 CPU。

“另存为...”按钮

如果单击该按钮，会将诊断缓冲区的内容保存到文本文件中。系统建议的文件名是“Diagnose.txt”。但可以更改此名称。

参见

有关诊断缓冲区的基本信息 (页 273)

6.5.4.3 对非 CPU 模块执行模块特定的诊断

在何处显示了非 CPU 模块的模块特定的诊断信息？

非 CPU 模块的模块特定的诊断信息显示在在线和诊断视图的“标准诊断”(Standard diagnostics) 组中。

“标准诊断”组

“标准诊断”(Standard diagnostics) 组中显示了非 CPU 模块的以下诊断信息：

- 与整个模块相关的内外部故障
- 相关诊断事件

这类诊断信息的实例有：

- 整个备用电源出现故障
- 模块有故障

注意 诊断中断

仅当模块具有诊断中断功能并且启用了诊断中断时，才能向 CPU 报告诊断中断。

诊断中断以快照形式显示。模块中偶尔发生的故障可在相应 CPU 的诊断缓冲区中识别。

6.5.5 更改模块或编程设备/PC 的属性

6.5.5.1 更改 CPU 的模式

要求

存在与要更改模式的 CPU 的在线连接。

步骤

要更改 CPU 的模式，请按以下步骤操作：

1. 启用 CPU 的“在线工具”(Online tools) 任务卡。
2. 如果要将在 CPU 切换到 RUN 模式，则在“CPU 控制面板”(CPU control panel) 窗格中单击“RUN”按钮；如果要将在 CPU 切换到 STOP 模式，则单击“STOP”按钮。

注意

那个激活的按钮是在 CPU 当前工作模式下可以选择的唯一按钮。

3. 单击“确定”(OK) 响应确认提示。

结果

CPU 将切换到要求的模式。

6.5.5.2 执行存储器复位

要求

- 存在与 CPU 的在线连接，将通过该连接执行存储器复位。
- CPU 处于 STOP 模式。

注意

如果 CPU 仍处于 RUN 模式，而用户要启动存储器复位，则可在响应确认提示后将其切换到 STOP 模式。

步骤

要对 CPU 执行存储器复位，请按以下步骤操作：

1. 启用 CPU 的“在线工具”(Online Tools) 任务卡。
2. 在“CPU 控制面板”(CPU control panel) 窗格中，单击“MRES”按钮。
3. 单击“确定”(OK) 响应确认提示。

结果

CPU 根据需要切换到 STOP 模式并且对 CPU 执行存储器复位。

参见

存储器复位的基础知识 (页 292)

6.5.5.3 确定和设置 CPU 的日时钟

在何处查找所需的功能？

可以在在线和诊断视图中“功能”(Functions) 文件夹的“设置日时钟”(Set time of day) 组中确定和更改 CPU 的日时钟。这需要在线连接。

“设置日时钟”组的结构

“设置日时钟”(Set time of day) 组由以下各区域组成：

- 编程设备/PC 时间
此处显示编程设备/PC 的时区设置、当前日期和当期时间设置。
- 模块时间
当前从模块（例如 CPU）读取的日期和时间值将转换为本地日期和时间并在此处显示。

如果选中“从 PG/PC 获取”(Take from PG/PC) 复选框，则单击“应用”(Apply) 按钮后，PG/PC 时间和日期会转换为 UTC，然后传送给模块。

如果未选中“从 PG/PC 获取”(Take from PG/PC) 复选框，则可为模块的集成时钟分配日期和时间。单击“应用”(Apply) 按钮后，将重新计算时间和日期所对应的 UTC 时间，然后传送给模块。

6.5.5.4 组态以太网设备

概述

可在首次组态时为模块分配 IP 地址和 IP 参数。

然后便可使用所设置的 IP 地址来访问模块以执行某些操作，例如装载组态数据或执行诊断。

要求

- 必须建立了以太网 LAN 连接。
- 编程设备或 PC 的以太网接口必须与要组态的模块位于相同的 IP 范围内。
- 模块与编程设备或 PC 必须属于同一个子网。

步骤

要将模块组态为以太网设备，请按以下步骤操作：

1. 打开模块的在线和诊断视图。
2. 从“功能”(Functions) 文件夹中选择“分配 IP 地址”(Assign IP address) 组。
3. 输入 MAC 地址 (如果知道)。
4. 输入 IP 地址。
5. 输入子网掩码。
6. 如果要使用路由器，请选中“使用路由器”(Use router) 复选框并输入其 IP 地址。
7. 单击“分配 IP 地址”(Assign IP address)。

结果

为模块分配了 IP 组态。

6.5.5.5 将 CPU 重置为出厂设置

要求

- CPU 中未插入存储卡。
- 存在与 CPU 的在线连接，将通过该连接重置为出厂设置。
- CPU 处于 STOP 模式。

注意

如果 CPU 仍处于 RUN 模式，而用户要启动重置操作，则可在响应确认提示后将其切换到 STOP 模式。

步骤

要将 CPU 重置为出厂设置，请按以下步骤操作：

1. 打开 CPU 的在线和诊断视图。
2. 从“功能”(Functions) 文件夹中选择“重置为出厂设置”(Reset to factory settings) 组。
3. 如果要保留 IP 地址，请选中“保留 IP 地址”(Retain IP address) 复选框；如果要删除 IP 地址，则选中“删除 IP 地址”(Delete IP address)。
4. 单击“重置”(Reset) 按钮。
5. 单击“确定”(OK) 响应确认提示。

结果

模块根据需要切换到 STOP 模式并且随后重置为出厂设置。这表示：

- 工作存储器、内部装载存储器以及所有操作数区域都会被删除。
- 所有参数都重置为其默认值。
- 诊断缓冲区被清空。
- 日时钟被重置。
- 将根据用户设置保留或删除 IP 地址。

6.5.6 在 STOP 模式下诊断

6.5.6.1 有关诊断缓冲区的基本信息

功能

CPU 的操作系统将 CPU 和具有诊断功能的模块所检测到的错误按其发生顺序依次输入诊断缓冲区。其中包括以下事件：

- CPU 的每次模式切换（上电、切换到 STOP 模式、切换到 RUN 模式）
- 所有硬件和诊断错误中断

最上面的条目包含最新发生的事件。诊断缓冲区中的条目会永久保存。这些条目即使在电源出现故障时也会被保留，并且只能通过将 CPU 重置为出厂设置进行删除。

诊断缓冲区条目由以下部分组成：

- 时间戳
- 错误 ID
- 错误 ID 特定的附加信息

诊断缓冲区的优点

诊断缓冲区有以下优点：

- 在 CPU 切换到 STOP 模式后，可以评估在切换到 STOP 模式之前发生的最后几个事件，从而可以查找并确定导致进入 STOP 模式的原因。
- 用户可以更快地检测并排除出现错误的原因，从而提高系统的可用性。
- 可以评估和优化动态系统响应。

诊断缓冲区的结构

诊断缓冲区是环形缓冲区。S7-1200 CPU 可保存最多 50 个条目。当诊断缓冲区已满而又需要创建新条目时，所有现有条目都将向下移动一个位置（这意味着删除最旧的条目），并在当前空闲的顶部位置创建新条目（FIFO 原理：先进先出）。先进先出）

诊断缓冲区的评估

可通过以下方式访问诊断缓冲区的内容：

- 使用在线和诊断视图

通过对错误事件（例如，切换到 STOP 模式）发生之前的事件进行评估，使用户能够大致了解可能的原因，或者是能够更紧密地追踪或更详细地确定可能的原因（取决于错误类型）。

请仔细阅读有关事件的详细信息，并使用“关于事件的帮助”(Help on event) 按钮来获得各条目的更多信息以及可能的原因。

注意

为了充分利用时间要求非常严格的系统中诊断缓冲区条目的时间戳信息，建议用户不定期地检查和更正 CPU 的日期和日时钟。

也可以利用 NTP 时间服务器执行日时钟同步。

参见

- 将 CPU 重置为出厂设置 (页 272)*
- 确定 CPU 进入 STOP 模式的原因 (页 274)*
- 确定和设置 CPU 的日时钟 (页 271)*
- 参数化时钟 (页 241)*

6.5.6.2 确定 CPU 进入 STOP 模式的原因

要求

要分析的 CPU 处于 STOP 模式。

步骤

要查找 CPU 切换到 STOP 模式的原因，请按以下步骤操作：

1. 打开 CPU 的在线和诊断视图。
2. 从“诊断”(Diagnostics) 文件夹中选择“诊断缓冲区”(Diagnostics buffer) 组。
3. 评估在切换到 STOP 模式之前所发生的事件。通过评估结果可大致了解可能的原因，或者可追踪或更详细地确定可能的原因（取决于错误类型）。

请仔细阅读有关事件的详细信息，并使用“关于事件的帮助”(Help on event) 按钮来获得各条目的更多信息以及可能的原因。

结果

能够追踪或更详细地确定 CPU 进入 STOP 模式的原因。

注意

如果通过分析仍旧无法解决问题，请联系客户支持。在这种情况下，请使用“另存为”(Save as) 按钮将诊断数据的内容备份为文本文件，然后将其发送给客户支持。

参见

读出 CPU 的诊断缓冲区 (页 267)

6.6 参考

6.6.1 创建未指定的 CPU

简介

如果还没有选择 CPU 但已开始编程或想要使用现有程序，则可以选择使用未指定的 CPU。

创建未指定的 CPU

要创建未指定的 CPU，请按以下步骤操作：

1. 转到 Portal 视图。
2. 单击“开始 > 入门指南”(Start > Getting started)
3. 单击“创建 PLC 程序”(Create a PLC program)

随即会创建一个未指定的 CPU。

或者：

1. 转到 Portal 视图。
2. 现在单击下列其中一个选项：
 - “设备和网络 > 添加新设备”(Devices & networks > Add new device)
 - “PLC 编程”(PLC programming) > “设备”(Device) 按钮

3. 从“添加新设备”(Add new device) 对话框的树结构中选择未指定的 CPU。
4. 单击“确定”(OK)。

随即会创建一个未指定的 CPU。

用于创建未指定 CPU 的其它选择

在项目视图中，可以通过下列方式创建未指定的 CPU 以及指定的 CPU：

- 使用项目浏览器中的“创建新设备”(Create new device) 按钮
- 使用“硬件目录”(Hardware catalog) 任务卡

还可以使用这些方法创建多个未指定的 CPU。

指定未指定的 CPU

可选择以下两种方式来指定未指定的 CPU：

- 通过拖放操作以模块替换 (页 201) 的方式将硬件目录中的现有 CPU 分配给未指定的 CPU。
- 选择未指定的 CPU，然后在菜单栏中单击“在线 > 硬件检测”(Online > Hardware detection)，并在线分配所标识的 CPU。为此，使用“添加 PG/PC 地址”(Add address for PG/PC) 按钮分配 IP 地址。

参见

选择 CPU (页 229)

将设备添加到硬件配置中 (页 195)

6.6.2 开放式用户通信

6.6.2.1 面向连接协议的工作原理

简介

面向连接的协议会在数据传输开始之前建立到通信伙伴的逻辑连接。数据传输完成后，这些协议会在必要时终止连接。面向连接的协议尤其适用于注重可靠性的数据传输。一条物理线路上可以存在多个逻辑连接。

开放式用户通信支持以下连接类型：

- TCP
- ISO-on-TCP

对于不支持 ISO-on-TCP 连接的通信伙伴，应使用 TCP 连接。对于诸如第三方设备或 PC 这些类型的通信伙伴，请在分配连接参数时，为伙伴端点输入“未指定”(unspecified)。

TCP 的特征

在通过 TCP 连接进行数据传输期间，不会传送有关消息长度或消息开始及消息结束的信息。这在发送期间并不会引起问题，因为发送方知道要传送的数据量。但是，接收方无法识别数据流中的某条消息在何处结束以及下一条消息在何处开始。因此，建议为要接收的字节数（参

数 LEN，指令 TRCV/TRCV_C) 和要发送的字节数 (参数 LEN，指令 TSEND/TSEND_C) 分配相同的值。

如果所发送数据的长度和所要求数据的长度不一致，则将出现以下情况：

- 要接收的数据的长度 (参数 LEN，指令 TRCV/TRCV_C) 大于要发送的数据的长度 (参数 LEN，指令 TSEND/TSEND_C)：

仅当达到所分配的长度后，TRCV/TRCV_C 才会将接收到的数据复制到指定的接收区 (参数 DATA)。达到所分配的长度时，已经接收了下一个作业的数据。因此，接收区包含的数据来自两个不同的发送作业。如果不知道第一条消息的确切长度，将无法识别第一条消息的结束以及第二条消息的开始。

- 要接收的数据的长度 (参数 LEN，指令 TRCV/TRCV_C) 小于要发送的数据的长度 (参数 LEN，指令 TSEND/TSEND_C)：

TRCV/TRCV_C 将 LEN 参数中指定字节的数据复制到接收数据区 (参数 DATA)。然后，将 NDR 状态参数设置为 TRUE (作业成功完成) 并将 LEN 的值分配给 RCVD_LEN (实际接收的数据量)。对于每次后续调用，都会接收已发送数据的另一个块。

ISO-on-TCP 的特征

在通过 ISO-on-TCP 连接进行数据传输期间，同时还会提供有关消息长度和消息结束的信息。

如果所发送数据的长度和所要求数据的长度不一致，则将出现以下情况：

- 要接收的数据的长度 (参数 LEN，指令 TRCV/TRCV_C) 大于要发送的数据的长度 (参数 LEN，指令 TSEND/TSEND_C)：

TRCV/TRCV_C 将所有发送数据复制到接收数据区 (参数 DATA)。然后，将 NDR 状态参数设置为 TRUE (作业成功完成) 并将所发送数据的长度分配给 RCVD_LEN (实际接收的数据量)。

- 要接收的数据的长度 (参数 LEN，指令 TRCV/TRCV_C) 小于要发送的数据的长度 (参数 LEN，指令 TSEND/TSEND_C)：

TRCV/TRCV_C 并不会将任何数据复制到接收数据区 (参数 DATA)，而是提供以下错误信息：
ERROR=1，STATUS=W#16#8088 (目标缓冲区太小)。

参见

开放式用户通信的基本知识 (页 213)

TSEND_C (页 891)

TRCV_C (页 894)

TSEND (页 902)

TRCV (页 904)

6.6.2.2 通信连接的参数

连接描述数据块

结构符合 TCON_Param 的连接描述数据块用于分配 TCP 和 ISO-on-TCP 的通信连接参数。TCON_Param 的固定数据结构包含了建立连接所需的全部参数。使用 TSEND_C、TRCV_C 或 TCON 指令时，将根据开放式用户通信的连接参数分配情况自动为新连接创建连接描述数据块。

TSEND_C、TRCV_C 或 TCON 相应背景数据块中的 CONNECT 连接参数包含对所用数据块的引用。

符合 TCON_Param 的连接描述结构

字节	参数	数据类型	初始值	描述
0 ... 1	block_length	UINT	64	长度：64 个字节（固定）
2 ... 3	id	CONN_OUC	1	对该连接的引用（取值范围：1 - 4095）。 对于 TSEND_C、TRCV_C 或 TCON 指令，在 ID 中指定该参数的值。
4	connection_type	USINT	17	连接类型： <ul style="list-style-type: none"> • 17: TCP • 18: ISO-on-TCP
5	active_est	BOOL	TRUE	建立连接的方式所对应的 ID： <ul style="list-style-type: none"> • FALSE：被动连接建立 • TRUE：主动连接建立
6	local_device_id	USINT	2	本地 PN/IE 接口的 ID。
7	local_tsap_id_len	USINT	0	所用参数 local_tsap_id 的长度（字节）；可能值： <ul style="list-style-type: none"> • 0 或 2（连接类型 = 17 (TCP) 时） 主动端只允许使用值 0。 • 2 到 16（连接类型 = 18 (ISO-on-TCP) 时）
8	rem_subnet_id_len	USINT	0	该参数未使用。
9	rem_staddr_len	USINT	4	伙伴端点的地址长度（字节）： <ul style="list-style-type: none"> • 0: 未指定，即参数 rem_staddr 不相关。 • 4: 参数 rem_staddr 中有效的 IP 地址
10	rem_tsap_id_len	USINT	2	所用参数 rem_tsap_id 的长度（字节）；可能值： <ul style="list-style-type: none"> • 0 或 2（连接类型 = 17 (TCP) 时） 被动端只允许使用值 0。 • 2 到 16（连接类型 = 18 (ISO-on-TCP) 时）
11	next_staddr_len	USINT	0	该参数未使用。

字节	参数	数据类型	初始值	描述
12 ... 27	local_tsap_id	BYTE 类型的 ARRAY [1..16]	-	<p>连接的本地地址部分：</p> <ul style="list-style-type: none"> 17 (TCP)：本地端口号 (可能值：1 到 49151；建议值：2000...5000)； local_tsap_id[1] = 端口号的高位字节 (十六进制表示)； local_tsap_id[2] = 端口号的低位字节 (十六进制表示)； local_tsap_id[3-16] = 不相关 18 (ISO-on-TCP)：本地 TSAP ID： local_tsap_id[1] = B#16#E0； local_tsap_id[2] = 本地端点的机架和插槽 (位 0 到 4：插槽号；位 5 到 7：机架号)； local_tsap_id[3-16] = TSAP 扩展 (可选) <p>注意：请确保 local_tsap_id 的每个值在 CPU 中都是唯一的。</p>
28 ... 33	rem_subnet_id	USINT 类型的 ARRAY [1..6]	-	该参数未使用。
34 ... 39	rem_staddr	USINT 类型的 ARRAY [1..6]	-	<p>伙伴端点的 IP 地址，例如，192.168.002.003：</p> <p>rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = 不相关</p>
40 ... 55	rem_tsap_id	BYTE 类型的 ARRAY [1..16]	-	<p>连接的伙伴地址部分</p> <ul style="list-style-type: none"> 17 (TCP)：伙伴端口号 (可能值：1 到 49151；建议值：2000...5000)； rem_tsap_id[1] = 端口号的高位字节 (十六进制表示)； rem_tsap_id[2] = 端口号的低位字节 (十六进制表示)； rem_tsap_id[3-16] = 不相关 18 (ISO-on-TCP)：伙伴 TSAP-ID： rem_tsap_id[1] = B#16#E0； rem_tsap_id[2] = 伙伴端点的机架和插槽 (位 0 到 4：插槽号；位 5 到 7：机架号)； rem_tsap_id[3-16] = TSAP 扩展 (可选)
56 ... 61	next_staddr	BYTE 类型的 ARRAY [1..6]	-	该参数未使用。
62 ... 63	spare	WORD	W#16#0000	保留。

参见

面向连接协议的工作原理 (页 276)

连接参数描述 (页 216)

回读连接描述参数功能 (页 280)

数据块编程的基本原理 (页 472)

连接参数分配概述 (页 214)

6.6.2.3 端口号的分配

简介

创建开放式用户通信时，系统会自动分配值 2000 作为端口号。

端口号的允许值为 1 到 49151。可以分配该范围内的任何端口号。但是，由于某些端口已被使用（取决于系统），因而建议使用 2000 到 5000 范围内的端口号。

端口号总览

下表汇总了系统对各个端口号的响应。

端口号	描述	系统响应
2000 ... 5000	建议范围	未出现警告或错误消息 允许使用并且接受端口号
1 ... 1999, 5001 ... 49151	可以使用，但不在建议范围内	出现了警告消息 允许使用并且接受端口号
20, 21, 25, 80, 102, 135, 161, 34962 ... 34964	在一定条件下可以使用*	
53, 80, 102, 135, 161, 162, 443, 520, 9001, 34962 ... 34964	在一定条件下可以使用**	

* 采用 TCP 连接类型时，这些端口由 TSEND_C 和 TRCV_C 使用。

** 这些端口是否封锁取决于所用 S7-1200 CPU 的功能范围。相应 CPU 的文档中提供了这些端口的分配信息。

参见

连接参数描述 (页 216)

创建和分配连接参数 (页 218)

6.6.2.4 回读连接描述参数功能

更改连接描述中的参数值

与开放式用户通信的连接——对应的连接描述通过连接参数分配输入到连接描述数据块中。

可以不通过连接参数分配在用户程序中更改连接描述数据块的参数值。不能更改连接描述的结构。

随后可通过连接参数分配回读包含更改值的连接描述数据块。在“属性 > 组态 > 连接参数”(Properties > Configuration > Connection parameters) 下，巡视窗口仅显示连接描述数据块中所存储的连接参数。

连接参数分配不支持 (只能通过偏移引用来进行查找的) 数据块类型 (例如，全局数据块) 的连接描述的嵌套输入。

回读各个连接参数的功能

对于通信伙伴的“地址”(Address) 参数，所显示的 IP 地址来自连接描述的“rem_staddr”参数。

以下值也可以从连接描述重新装载：

- 连接类型
- 本地连接 ID
- 连接建立 (主动/被动)
- 本地 TSAP (仅限 ISO-on-TCP)
- 伙伴 TSAP (仅限 ISO-on-TCP)
- 本地端口 (仅限 TCP)
- 伙伴端口 (仅限 TCP)

通信伙伴的连接 ID 参数值、连接数据以及连接建立，都不包含在本地连接描述数据块的连接描述中。因此，在重新打开连接参数分配时无法显示这些参数。伙伴的连接建立是由本地连接建立而产生的，所以也会显示出来。

可随时在“伙伴”(Partners) 下拉列表框中选择新的通信伙伴。

将项目中识别的 CPU 选为指定的通信伙伴时，连接 ID 和连接数据相应的输入选项将再次显示。

参见

通信连接的参数 (页 277)

连接参数描述 (页 216)

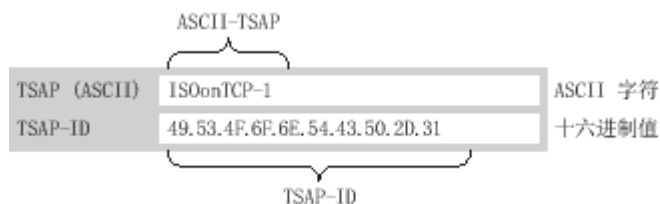
6.6.2.5 TSAP 结构

简介

对于 ISO-on-TCP 连接，必须同时为两个通信伙伴分配传输服务访问点 (TSAP, Transport Service Access Point)。创建了 ISO-on-TCP 连接后，系统会自动分配 TSAP ID。要确保 TSAP ID 在设备中唯一，可以在连接参数分配中更改预分配的 TSAP。

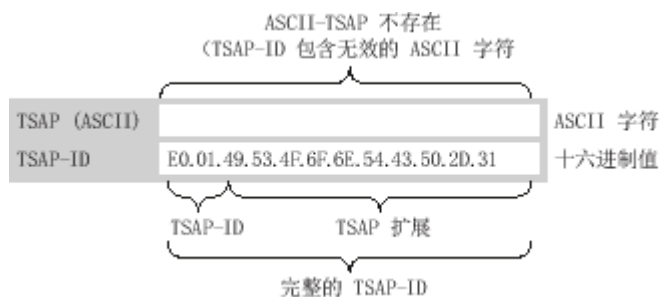
TSAP 的结构

分配 TSAP 时，必须遵守某些规则。TSAP 必须包含一定数量的字节，这些字节可以十六进制值 (TSAP-ID) 或 ASCII 字符 (ASCII-TSAP) 的形式显示和输入：



在相应输入域中输入或更改 TSAP-ID 或 ASCII-TSAP 后，它们也总是能够以另一种显示格式生效。

如果 TSAP 包含无效的 ASCII 字符，则 TSAP 将只显示为 TSAP-ID，而不显示为 ASCII-TSAP。创建连接后就是如此。前两个十六进制字符作为 TSAP-ID，用来标识通信类型和机架/插槽。因为这些字符对于 CPU 是无效 ASCII 字符，所以在这种情况下不显示 ASCII-TSAP。



除了遵守 TSAP 在长度和结构方面的规则外，还必须确保 TSAP-ID 是唯一的。所分配的 TSAP 不会自动具有唯一性。

TSAP 的长度和内容

TSAP 的结构如下：

- 具有 TSAP 扩展的 TSAP-ID
 - 长度 = 2 到 16 个字节
 - x_tsap_id[0] = 0xE0 (开放式用户通信)
 - x_tsap_id[1] (位 0 到 4) = CPU 的插槽号
 - x_tsap_id[1] (位 5 到 7) = CPU 的机架号
 - x_tsap_id[2...15] = 任何字符 (TSAP 扩展, 可选)
 - (x = loc (本地) 或 x = rem (伙伴))
- ASCII-TSAP 形式的 TSAP-ID
 - 长度 = 3 到 16 个字节
 - x_tsap_id[0 到 2] = 3 个 ASCII 字符 (0x20 到 0x7E)
 - x_tsap_id[3...15] = 任何字符 (可选)
 - (x = loc (本地) 或 x = rem (伙伴))

下表给出了 TSAP-ID 的示意性结构：

TSAP-ID	tsap_id_len	tsap_id[0]	tsap_id[1]	tsap_id[2..15]	tsap_id[3..15]
...具有扩展	2 到 16 个字节	0xE0	0x01 (0x00)*	扩展 (可选)	扩展 (可选)
...ASCII-TSAP 形式	3 到 16 个字节	0x20...0x7E	0x20...0x7E	0x20...0x7	任意 (可选)

* 已识别的 CPU 通常插在机架 0 的插槽 1 中。因此，十六进制值 01 对具有扩展 TSAP-ID 的第二个位置有效。如果连接伙伴是未指定的 CPU (例如第三方设备)，则还允许对插槽地址使用十六进制值 00。

注意

对于未指定的通信伙伴，本地 TSAP-ID 和伙伴 TSAP-ID 的长度可以为 0 到 16 个字节，其中允许使用 00 到 FF 之间的所有十六进制值。

用于输入 ASCII TSAP 的 ASCII 代码表

在连接参数分配中输入 ASCII-TSAP 时，只允许使用 20 到 7E 之间的十六进制值：

代码	..0	..1	..2	..3	..4	..5	..6	..7	..8	..9	..A	..B	..C	..D	..E	..F
2..		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3..	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4..	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5..	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6..	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7..	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

参见

TSAP 分配的实例 (页 283)
连接参数描述 (页 216)
创建和分配连接参数 (页 218)

6.6.2.6 TSAP 分配的实例

以下实例说明了编辑 TSAP 所涉及的各个方面：

- 实例 1：为 PLC-PLC 通信创建新连接
- 实例 2：输入本地 ASCII-TSAP
- 实例 3：在 TSAP-ID 中输入 TSAP 扩展

- 实例 4：TSAP-ID 的错误编辑
- 实例 5：通过“TSAP-ID”输入域输入 ASCII-TSAP

实例 1：为 PLC-PLC 通信创建新连接

使用两个用于开放式用户通信的 PLC 创建新连接后，系统将立即自动分配 TSAP 扩展“ISOonTCP-1”。

该 TSAP 扩展会产生 TSAP-ID E0.01.49.53.4F.6F.6E.54.43.50.2D.31，这会被自动输入到连接描述数据块以及本地和伙伴 TSAP 的输入域中。ASCII-TSAP 的输入域为空白：

	本地 TSAP	伙伴 TSAP
TSAP (ASCII)		
TSAP-ID	E0.01.49.53.4F.6F.6E.54.43.50.2D.31	E0.01.49.53.4F.6F.6E.54.43.50.2D.31

您可以随时更改 TSAP-ID 和 ASCII-TSAP 输入域中的这些值。

TSAP-ID 的输入域显示了连接描述数据块中所存储的完整 TSAP。具有 TSAP 扩展的 TSAP-ID (限制为 16 个字符) 不会显示在“TSAP (ASCII)”输入域中，因为字符 E0 不是 ASCII-TSAP 的有效字符。

如果显示的 TSAP-ID 是有效 ASCII-TSAP，它将显示在“TSAP (ASCII)”输入域中。

更改 TSAP-ID 和 ASCII-TSAP 的输入域会影响其它域。

实例 2：输入本地 ASCII-TSAP

如果已创建新连接并在“TSAP (ASCII)”输入域中为本地 TSAP 分配了 ASCII 值 (例如，“ISOonTCP-1”)，则将自动创建最终的 TSAP-ID。

退出“TSAP (ASCII)”输入域后，系统将自动检查 ASCII 字符数是否符合限值 (3 到 16 个字符) 并将所生成的 TSAP-ID 输入到相应的输入域中：

	本地 TSAP	伙伴 TSAP
TSAP (ASCII)	ISOonTCP-1	
TSAP-ID	49.53.4F.6F.6E.54.43.50.2D.31	E0.01.49.53.4F.6F.6E.54.43.50.2D.31

实例 3：在 TSAP-ID 中输入 TSAP 扩展

如果在创建连接并在本地 TSAP-ID 的输入域中输入 ASCII-TSAP (请参见实例 1 和 2) 后，用户将前缀“E0.01”添加到 TSAP 值中，则在退出该输入域后将不再显示 ASCII-TSAP。

	本地 TSAP	伙伴 TSAP
TSAP (ASCII)		
TSAP-ID	E0.01.49.53.4F.6F.6E.54.43.50.2D.31	E0.01.49.53.4F.6F.6E.54.43.50.2D.31

一旦退出 TSAP-ID 的输入域，系统就会自动执行检查以确定 TSAP-ID 的第一个字符是否是有效的 ASCII 字符。由于 TSAP-ID 中此时所显示的字符“E0”对于 ASCII-TSAP 不是有效字符，因此“TSAP (ASCII)”输入域不再显示 ASCII-TSAP。

如果使用的是有效的 ASCII 字符，接下来就会检查是否符合 2 到 16 个字符的长度规范。

实例 4：TSAP-ID 的错误编辑

如果从以“E0.01”开头的“TSAP-ID”中删除十六进制值“E0”，TSAP-ID 此时将以“01”开头，因此不再符合这些规则且无效：

	本地 TSAP	伙伴 TSAP
TSAP (ASCII)		
TSAP-ID	01.49.53.4F.6F.6E.54.43.50.2D.31	E0.01.49.53.4F.6F.6E.54.43.50.2D.31

退出输入域后，将会输出一条消息，因为该 TSAP-ID 既不是有效的 ASCII-TSAP（第一个值必须是 20 到 7E 范围内的十六进制值），也不是有效的 TSAP-ID（第一个值必须是标识符“E0”）。

实例 5：通过“TSAP-ID”输入域输入 ASCII-TSAP

在实例 4 中，如果删除值“E0”后又从错误的 TSAP-ID 中删除值“01”，则 TSAP-ID 将以十六进制值 49 开头。该值在 ASCII-TSAP 的允许值范围内：

	本地 TSAP	伙伴 TSAP
TSAP (ASCII)		
TSAP-ID	49.53.4F.6F.6E.54.43.50.2D.31	E0.01.49.53.4F.6F.6E.54.43.50.2D.31

退出输入域后，TSAP-ID 将被识别为有效的 ASCII-TSAP，而所生成的 ASCII-TSAP“ISOonTCP-1”将被写入到“TSAP (ASCII)”输入域中。

参见

TSAP 结构 (页 281)

连接参数描述 (页 216)

对 PLC 进行编程

7.1 S7-1200 CPU 的功能说明

7.1.1 工作模式

7.1.1.1 S7-CPU 工作模式的原理

简介

通过工作模式描述 CPU 的行为。有下列工作模式：

- STARTUP
- RUN
- STOP

例如，在这些工作模式下，CPU 可通过 PN/IE 接口进行通信。

其它工作模式

如果 CPU 没有准备好运行，则它处于以下两种工作模式之一：

- 断电，即电源电压关闭。
- 故障，表示发生了内部错误。

如果此“故障”状态由固件错误导致，则将通过 CPU 的状态 LED 指示该状态（请参见 CPU 的说明）。要找出原因，请按以下步骤操作：

- 断开电源开关，然后再次接通。
- 读出 CPU 启动时的诊断缓冲区内容并将需要分析的数据发送给客户支持。

如果 CPU 没有启动，则更换该 CPU。

参见

“STARTUP”工作模式 (页 288)

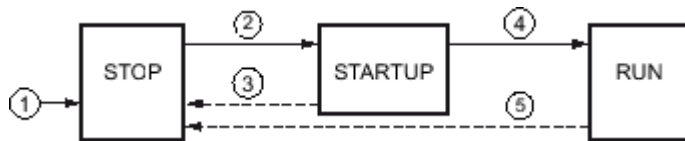
STOP 模式 (页 292)

RUN 模式 (页 292)

7.1.1.2 工作模式转换

概述

下图显示了 S7-1200 CPU 的工作模式和工作模式转换：



下表给出了工作模式发生改变的条件：

编号	工作模式转换	条件
①	STOP	接通电源后，CPU 处于“STOP”模式。然后确定所需的启动类型并切换到下一工作模式。
②	STOP → STARTUP	如果硬件配置和程序块一致，则在以下情况下 CPU 将切换到“STARTUP”模式： <ul style="list-style-type: none"> • 从编程设备将 CPU 设置为“RUN”模式。 • “POWER-ON”自动触发 STARTUP 工作模式后。
③	STARTUP → STOP	在以下情况下 CPU 将返回到“STOP”模式： <ul style="list-style-type: none"> • 启动期间检测到错误。 • 从编程设备将 CPU 设置为“STOP”模式。 • 在 STARTUP OB 中执行了 STOP 命令。
④	STARTUP → RUN	如果 STARTUP 成功，CPU 将切换到“RUN”模式。
⑤	RUN → STOP	在以下情况下 CPU 将返回到“STOP”模式： <ul style="list-style-type: none"> • 检测到阻止继续处理的错误。 • 从编程设备将 CPU 设置为“STOP”模式。 • 在用户程序中执行了 STOP 命令。

7.1.1.3 “STARTUP”工作模式

7.1.1.3 STARTUP 模式的原理

功能

接通 CPU 后，它在开始执行循环用户程序之前首先执行启动程序。

通过适当编写启动 OB，可以在启动程序中为循环程序指定一些初始化变量。对启动 OB 的数量没有要求。即，可以在用户程序中创建一个或多个启动 OB，或者一个也不创建。

启动特性的参数设置

可以指定 CPU 是否保持在 STOP 模式或者是否在暖启动。此外，还可以在 CPU 属性的“启动”(Startup) 组中设置启动期间的响应 (RUN 或前一模式)。

特殊特性

请注意以下几点有关“STARTUP”模式的内容：

- 执行启动 OB。不管选择哪种启动模式，已编写的所有启动 OB 都会执行。
- 不能执行基于时间或基于中断的程序。
- 禁用模块上的输出。
- 不更新过程映像；可以对输入进行直接 I/O 访问。

参见

编辑属性和参数 (页 202)

S7-CPU 工作模式的原理 (页 287)

启动组织块 (页 313)

暖启动 (页 289)

7.1.1.3 暖启动

功能

在暖启动期间，所有非保持性位存储器内容都将删除并且非保持性数据块内容将复位为来自装载存储器的初始值。保持性位存储器和数据块内容将保留。

程序执行在调用第一个启动 OB 时开始。

触发暖启动

可以在以下情况下，在编程设备上使用相应的菜单命令触发“暖启动”：

- CPU 必须处于“STOP”模式。
- 存储器复位后
- 在 CPU 处于“STOP”模式时装载一致的程序和一致的硬件配置后。

如果为启动响应设置了以下参数，则“POWER ON”将触发“暖启动”：

- 启动类型“暖启动 - RUN”(不管 POWER OFF 前 CPU 所处的工作模式是什么)。
- “暖启动 - POWER OFF 前的模式”(取决于 POWER OFF 前的 CPU 工作模式。CPU 在此前必须已处于 RUN 模式。)

参见

保持性存储区 (页 295)

7.1.1.3 启动操作

概述

下表给出了 STARTUP 时 CPU 执行的操作：

执行顺序中的各操作	暖启动时
清除非保持性位存储器的内容	执行
清除所有位存储器内容	不执行
清除过程映像输出内容	执行
处理启动 OB	执行
更新过程映像输入内容	执行
切换到“RUN”模式后启用输出	执行

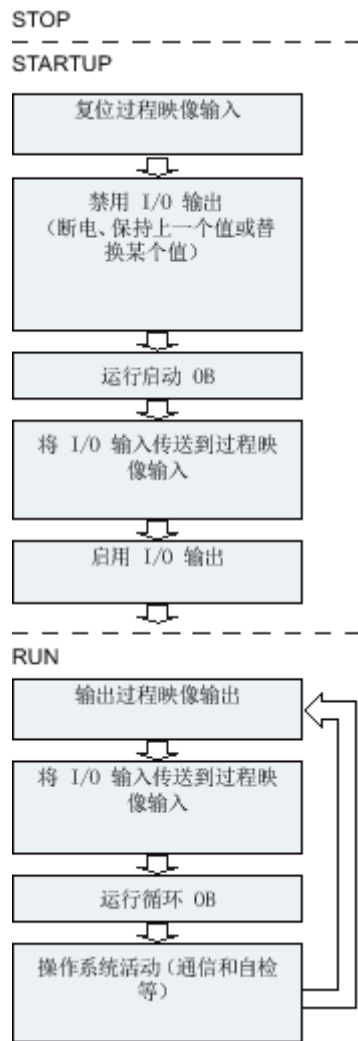
顺序

下图显示了在“STOP”、“STARTUP”和“RUN”模式下 CPU 的操作。

可以使用以下方法在用户程序的第一个周期中指定 I/O 输出的状态：

- 使用可分配的输出模块可以输出替换值或保持上一个值。
- 在启动 OB 中为输出设置默认值。

启动期间，所有中断事件都输入到队列中，以便随后可以在 RUN 模式期间处理它们。在 RUN 模式下，可随时处理硬件中断。



7.1.1.3 启动期间的特性

期望组态和实际组态不匹配时的响应

期望组态是指装载到 CPU 中的工程组态。实际组态是指自动化系统的实际组态。如果期望组态和实际组态不同，CPU 最初仍将切换到 RUN 模式。

取消 STARTUP

如果启动期间出错，则取消启动且 CPU 保持在“STOP”模式。

在以下情况下，将不会执行启动或者会取消启动：

- 如果没有插入有效的 SIMATIC SD 卡。
- 如果尚未装载硬件配置。

参见

CPU 属性概述 (页 232)

7.1.1.4 RUN 模式

功能

在“RUN”模式下，执行循环、时间驱动和中断驱动的程序段：

- 读出过程映像输出。
- 执行用户程序。
- 读取过程映像输入表。

只有在“RUN”模式下，才能通过开放式用户通信在 S7-1200 CPU 间进行主动数据交换。

参见

S7-CPU 工作模式的原理 (页 287)

开放式用户通信 (页 213)

7.1.1.5 STOP 模式

功能

在“STOP”模式下，不执行用户程序。根据参数设置，禁用或响应所有输出：根据参数中的设置提供替换值或保持上一个值输出并将受控过程置于安全状态。

CPU 检查以下几点：

- 硬件，例如是否所有模块均可用
- CPU 的默认设置是否适用，或者参数集是否存在
- 是否满足所编写的启动行为的一般条件

参见

S7-CPU 工作模式的原理 (页 287)

7.1.1.6 存储器复位的基础知识

功能

只有在 STOP 模式下，才能执行 CPU 的存储器复位。

复位存储器时，CPU 切换到“初始状态”。这表示：

- 工作存储器内容被清除（保持性和非保持性数据）。
- 装载存储器内容（代码和数据块）随后复制到工作存储器。结果，数据块不再具有当前值而是具有其初始值。

- 终止编程设备/PC 和 CPU 间的现有在线连接。
- 保留诊断缓冲区、时间、IP 地址、硬件配置和激活的强制作业。

注意

如果在 CPU 断电时更换了存储卡，则 CPU 将在再次上电时运行存储器复位。

7.1.2 存储区

7.1.2.1 装载存储器

功能

每个 CPU 都具有内部装载存储器。该内部装载存储器的大小取决于所使用的 CPU。

该内部装载存储器可以用外部存储卡来替代。如果未插入存储卡，CPU 将使用内部装载存储器；如果插入了存储卡，CPU 将使用该存储卡作为装载存储器。

但是，可使用的外部装载存储器的大小不能超过内部装载存储器的大小，即使插入的 SD 卡有更多空闲空间。

参见

使用存储卡 (页 150)

7.1.2.2 工作存储器

功能

工作存储器是非保持性存储器区域，用于存储与程序执行有关的用户程序元素。用户程序只能在工作存储器和系统存储器中执行。

7.1.2.3 系统存储器

7.1.2.3 系统存储器区域

功能

系统存储器包含各 CPU 为用户程序提供的存储器单元，例如，过程映像和位存储器。

通过在用户程序中使用合适的操作，可以在相关操作数区域中直接对数据寻址。

下表给出了系统存储器的操作数区域：

操作数区域	描述	以如下大小的单位进行访问：	S7 符号
过程映像输出	CPU 在循环开始时将过程映像输出表中的值写入到输出模块。	输出 (位)	Q
		输出字节	QB
		输出字	QW
		输出双字	QD
过程映像输入	CPU 在循环开始时从输入模块读取输入值并将这些值保存到过程映像输入表中。	输入 (位)	I
		输入字节	IB
		输入字	IW
		输入双字	ID
位存储区	此区域用于存储程序中计算出的中间结果。	位存储器 (位)	M
		存储器字节	MB
		存储器字	MW
		存储器双字	MD
数据块	数据块存储程序信息。可以对它们进行定义以便所有代码块都可以访问它们 (全局数据块)，也可将其分配给特定的 FB 或 SFB (背景数据块)。 要求：未启用块属性“仅符号访问”(Symbolic access only)。	数据位	DBX
		数据字节	DBB
		数据字	DBW
		数据双字	DBD
局部数据	此区域包含块处理过程中块的临时数据。 要求：未启用块属性“仅符号访问”(Symbolic access only)。 建议：仅以符号方式访问局部数据 (临时数据)。	局部数据位	L
		局部数据字节	LB
		局部数据字	LW
		局部数据双字	LD
I/O 输入区域	I/O 输入和输出区域允许直接访问集中式和分布式输入和输出模块。	I/O 输入位	<变量>:P
I/O 输出区域		I/O 输入字节	
		I/O 输入字	
		I/O 输入双字	
		I/O 输出位	
		I/O 输出字节	
		I/O 输出字	
		I/O 输出双字	

参见

诊断缓冲区 (页 297)
过程映像的基本原理 (页 295)
数据块编程的基本原理 (页 472)
在块接口中声明局部变量 (页 401)
块接口的布局 (页 398)
访问 I/O 地址 (页 297)

7.1.2.3 保持性存储区

保持性存储区

通过将某些数据标记为具有保持性可以避免电源故障后数据丢失。此类数据存储在保持性存储区中。保持性存储区是指在暖启动后（换言之，CPU 从 STOP 切换到 RUN 时的循环上电后）其内容依然保留的区域。

保持性数据的值在冷启动期间将会删除。

可以为以下数据赋予保持性：

- 位存储器：可以在 PLC 变量表或分配列表中为位存储器定义精确的存储器宽度。
- 功能块 (FB) 的变量：在 FB 的接口中，可以在启用符号变量寻址的情况下将各变量定义为具有保持性。仅当没有为 FB 激活符号寻址时，才可以在分配的背景数据块中定义保持性设置。
- 全局数据块的变量：根据符号寻址的设置，可以为全局数据块中的个别或所有变量定义为具有保持性。
启用数据块的“仅符号访问”(Symbolic access only) 属性：可以针对每个具体变量设置其保持性。
禁用数据块的“仅符号访问”(Symbolic access only) 属性：保持性设置应用于数据块的所有变量；即所有变量都具有保持性或所有变量都不具有保持性。

参见

设置 PLC 变量的保持性 (页 369)
设置局部变量的保持性 (页 406)
设置全局数据块中的保持性 (页 478)
暖启动 (页 289)

7.1.2.3 过程映像输入/输出

7.1.2.3 过程映像的基本原理

功能

用户程序对输入 (I) 和输出 (O) 操作数区域寻址时，不会查询或更改数字量信号模块端的信号状态。而是访问 CPU 系统存储器中的存储区。该存储区称为过程映像。

过程映像的优点

与直接访问输入和输出模块相比，访问过程映像的主要优点在于在一个程序周期期间，CPU 具有一致性的过程信号映像。如果程序执行期间输入模块端的信号状态发生变化，过程映像中的信号状态仍保持不变，直到下一个周期再次更新过程映像。在用户程序中周期性地扫描输入信号的过程，确保了总有一致的输入信息。

访问过程映像还比直接访问信号模块更节省时间，因为过程映像位于 CPU 的内部存储器中。

7.1.2.3 更新过程映像

顺序

除非另行定义，否则操作系统将以组态中的循环时间间隔更新过程映像。过程映像输入/输出按以下顺序更新：

1. 执行操作系统的内部任务。
2. 将过程映像输出 (PIQ) 表写入到模块的输出。
3. 将输入状态读取到过程映像输入 (PII) 表中。

除非强制，否则没有相应硬件输入的输入位将设置为 0。

4. 执行用户程序以及它内部所调用的所有块。

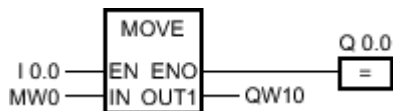
操作系统自动控制过程映像输出到模块输出的写入操作和过程映像输入的读取操作。

特殊特性

可以选择通过直接 I/O 访问来直接访问输入或输出。

- 如果指令直接访问输出且输出地址位于过程映像输出中，将更新相关输出的过程映像。
- 如果指令直接访问输出但输出地址不在过程映像输出中，则**不会**更新相关输出的过程映像。

通过过程映像进行正常 I/O 访问的实例



使用来自 MW0 的值更新 I/O 输出区中的 QW10。

过程映像更新期间的 I/O 访问错误

如果更新过程映像期间出错 (I/O 访问错误)，则 CPU 将以默认的系统响应“STOP”进行响应。

参见

故障排除选项 (页 469)

模块的起始地址 (页 297)

访问 I/O 地址 (页 297)

启动操作 (页 290)

7.1.2.3 诊断缓冲区

功能

诊断缓冲区是 CPU 系统存储器的一部分。诊断缓冲区包含由 CPU 或具有诊断功能的模块所检测到的错误。其中包括以下事件：

- CPU 的每次模式切换（例如，POWER UP、切换到 STOP 模式、切换到 RUN 模式）
- 每次诊断中断

S7-1200-CPU 的诊断缓冲区可容纳 50 个条目，其中最后（最近的）10 个条目在循环上电后将保留下来。

这些条目只能通过将 CPU 恢复到工厂默认设置进行清除。

可借助“在线和诊断”视图来读取诊断缓冲区的内容。

参见

有关诊断缓冲区的基本信息 (页 273)

7.1.3 I/O 数据区

7.1.3.1 模块的起始地址

定义

起始地址就是模块的最低字节地址。它表示模块中用户数据区的起始地址，且在很多情况下用于表示整个模块。

组态模块起始地址

组态模块时，调整用户程序中所使用的地址与模块起始地址。

在模块属性（“I/O 地址”(I/O addresses) 组）中，可更改插入模块后自动分配的起始地址。

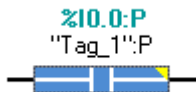
还可进行设置，以确定地址是否位于过程映像中。

7.1.3.2 访问 I/O 地址

I/O 地址

如果将模块插入设备视图中，其用户数据将位于 S7-1200-CPU 的过程映像中（默认设置）。在过程映像更新期间，CPU 会自动处理模块和过程映像区之间的数据交换。

如果希望程序直接访问用户数据（而不是使用过程映像），则为 I/O 地址附加后缀“:P”。



在某些情况下这可能很有必要，例如，在执行时效性程序期间且该程序必须控制同一周期内的输出时。

7.1.4 程序执行的基础知识

7.1.4.1 事件和 OB

事件和 OB

事件是 S7-1200-CPU 操作系统的基础。有两种类型的事件：

- 能够启动 OB 的事件
- 无法启动 OB 的事件

能够启动 OB 的事件会触发以下响应：

- 它会调用可能已分配给该事件的事件。如果当前无法调用此 OB，则按照事件的优先级将其输入队列。
- 如果没有为该事件分配 OB，则会触发默认系统响应。

无法启动 OB 的事件会触发相关事件类别的默认系统响应。

因此，用户程序循环取决于事件和给这些事件分配的 OB，以及包含在 OB 中的代码或在 OB 中调用的代码。

下表简要介绍了能够启动 OB 的事件，其中包括相关的事件类别和 OB。此表根据 OB 优先级进行排序。优先等级 1 为最低优先级。

事件类别	OB 号	OB 数目	启动事件	OB 优先级	优先级组
循环程序	1, >= 200	>= 1	启动或结束上一个循环 OB	1	1
启动	100, >= 200	>=0	STOP 到 RUN 的转换	1	

事件类别	OB 号	OB 数目	启动事件	OB 优先级	优先级组
延时中断	>= 200	最多 4 个	延迟时间结束	3	2
循环中断	>= 200		等长总线循环时间结束	4	
硬件中断	>= 200	最多 50 个 (通过 DETACH 和 ATTACH 指 令可使用更 多)	<ul style="list-style-type: none"> • 上升沿 (最多 16 个) • 下降沿 (最多 16 个) 	5	
			<ul style="list-style-type: none"> • HSC : 计数值 = 参考值 (最多 6 次) • HSC : 计数方向变化 (最多 6 次) • HSC : 外部复位 (最多 6 次) 	6	
诊断错误中断	82		0 或 1	模块检测到错误	9
时间错误	80	0 或 1	超出最大循环时间	26	3
			<ul style="list-style-type: none"> • 仍在执行所调用的 OB • 队列溢出 • 因中断负载过高而导致中断丢失 		

下表列出了不触发 OB 启动的事件，其中包括操作系统的相应响应。此表根据事件优先级进行排序。

事件类别	事件	事件优先级	系统响应
插入/卸下	插入/卸下模块	21	STOP
访问错误	过程映像更新期间的 I/O 访问错误	22	忽略
编程错误	块中的编程错误 (向其使用操作系统提供的系统响应) (请注意: 如果激活了本地错误处理, 则会执行块程序中的错误处理程序)。	23	STOP
I/O 访问错误	块中的 I/O 访问错误 (向其使用操作系统提供的系统响应) (请注意: 如果激活了本地错误处理, 则会执行块程序中的错误处理程序)。	24	STOP
超出最大循环时间两倍	超出最大循环时间两倍	27	STOP

OB 和事件之间的分配

除循环程序和启动程序外，事件只能分配给一个 OB。但是，在某些事件类别（如硬件中断）中，同一个 OB 可以分配给多个事件。

OB 和事件之间的分配在硬件配置中定义。已定义的分配可在运行时通过 ATTACH 和 DETACH 指令进行更改。

OB 优先级和运行时特性

S7-1200 CPU 支持优先等级 1 (最低) 到 27 (最高), 这些优先等级划分为 3 个优先级组。为 OB 分配其启动事件的优先级。

某一优先级组中的事件优先于任何较低优先级组中的 OB。这表示：

- 优先级组 2 中的任何事件都会中断循环程序执行。
- 优先级组 2 中的 OB 无法由优先级组 2 中的任何其它事件中断。此规则也适用于优先级高于当前激活 OB 的事件。

正在执行优先级组 2 中的 OB 时，优先级组 2 中生成的任何其它事件都会按照其优先级顺序添加到队列中。这些队列稍后会根据事件的优先级顺序进行处理（从最高优先级开始）。按时间顺序处理队列中的启动事件。

- 时间错误会中断任何其它 OB。

OB 启动信息

只有某些 OB 具有启动信息。在相关 OB 的描述中对此进行了更加详细的说明。

参见

执行基于事件的程序 (页 300)

局部错误处理的原理 (页 470)

ATTACH (页 922)

DETACH (页 923)

组织块的基本知识 (页 312)

故障排除选项 (页 469)

7.1.4.2 执行基于事件的程序

OB 优先级和运行时特性

S7-1200-CPU 支持优先等级 1（最低）到 27（最高）。为 OB 分配其启动事件的优先级。

中断 OB 只能由时间错误中断进行中断。此规则也适用于优先级高于当前激活 OB 的事件。即，除时间错误中断 OB 外，只能激活一个中断 OB。

正在执行中断 OB 时，生成的任何其它事件都会按照其优先级添加到队列中。稍后按照其发生的时间顺序来处理队列中的启动事件。

CPU 上的程序执行

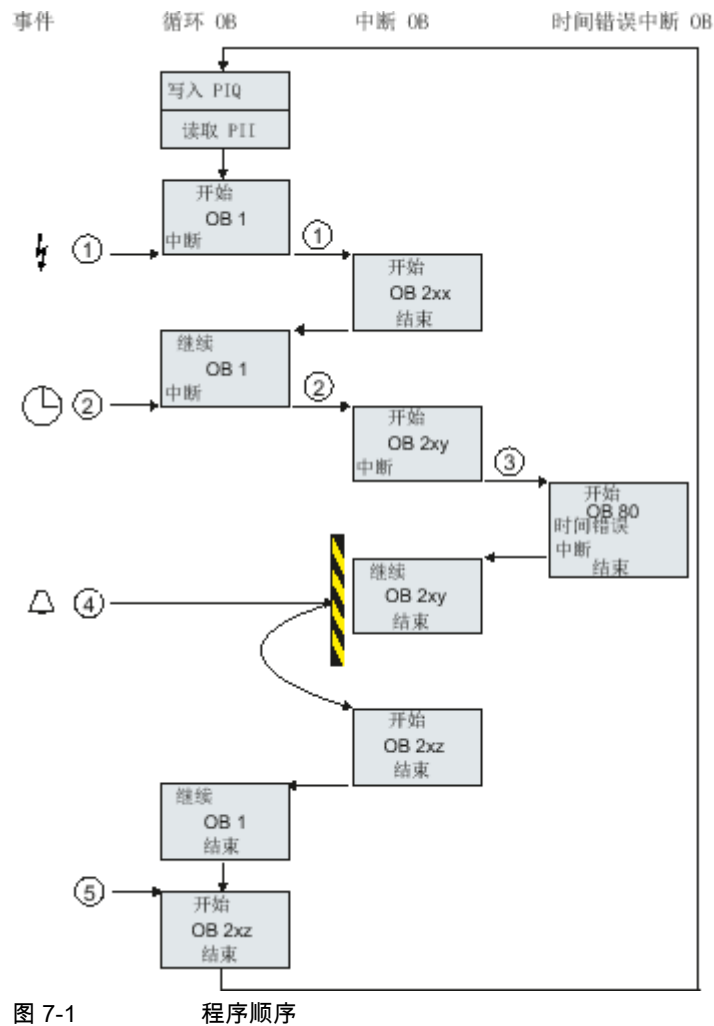
以三种不同的等级处理 CPU 上的程序执行：

- 循环组织块等级（例如 OB 1）
- 由于出现硬件中断之类的事件而调用的组织块（中断 OB）等级
- 时间错误中断组织块（时间错误中断 OB）等级

循环 OB 由中断 OB 进行中断。

中断 OB 只能由时间错误中断 OB 进行中断。

下图显示了基本顺序：



程序执行的描述

- ① and ② 某个事件（如硬件中断）调用其相关联的 OB。
不中断地执行已调用的 OB 及其所有嵌套块。如果队列不包含任何可触发 OB 启动的事件，则在完成中断处理时会继续执行循环 OB。
- ③ 中断 OB 只能由时间错误中断 OB (OB 80) 进行中断。
- ④ 中断处理期间会发生新的报警触发事件。这个新事件会添加到队列中。排队的事件只能在完成执行当前中断 OB 后按照以下规则连续调用其相应的 OB：
 - 事件按照其优先级顺序进行处理（从最高优先级开始）
 - 优先级相同的事件按照时间顺序进行处理
- ⑤ 循环 OB 相继进行处理。

关于队列的注意事项

- 每个优先等级（要调用的优先级相同的 OB）均分配单独的队列。已默认设置了这些队列的大小。

- 导致队列溢出的任何新事件都将丢弃，因此将丢失。同时会生成“时间错误中断事件”。标识导致此错误的 OB 的信息包含在时间错误中断 OB (OB 80) 的启动信息中。可在时间错误中断 OB 中编写相应的响应（如报警触发）。

参见

组织块的基本知识 (页 312)

7.1.4.3 硬件中断事件的实例

根据硬件中断触发模块的实例来描述 S7-1200-CPU 中面向事件的程序执行的功能原理。

过程事件及其优先级

过程事件由 I/O（例如通过数字量输入）触发并在 S7-1200 CPU 中启动对已分配 OB 的调用。分配给硬件中断事件 OB 就是所谓的硬件中断 OB。

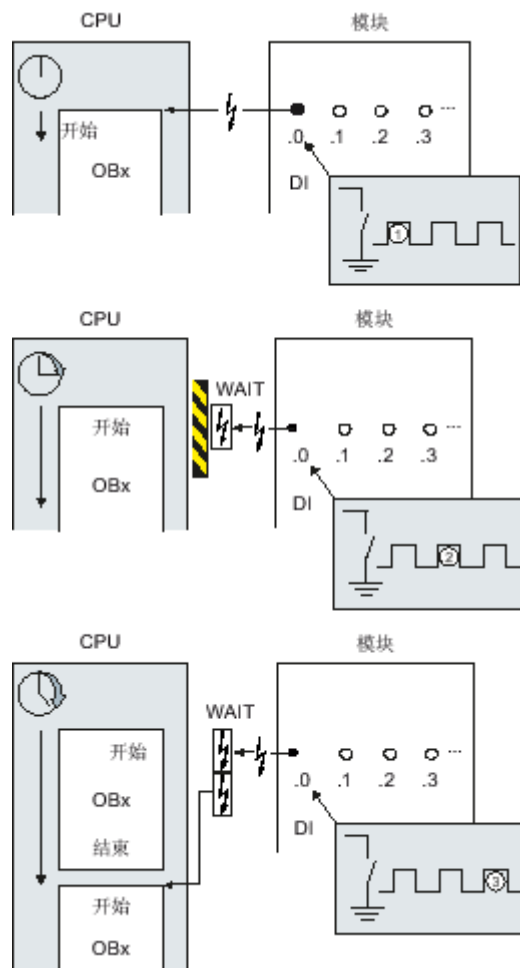
过程事件及其优先级的实例：

- 中断触发模块中的过程事件“上升沿”或“下降沿”：由此事件启动的硬件中断 OB 总是分配为优先级 5。
- 来自高速计数器的过程事件
 - 计数值对应于参考值
 - 更改计数方向
 - 高速计数器的外部复位

由此事件启动的硬件中断 OB 总是分配为优先级 6。

OB 按照其优先级升序执行。

下图显示了硬件中断执行的顺序。



硬件中断执行

- ① 硬件中断触发事件（如输入端的上升沿）调用它所属的 OB。
- ② 如果执行 OB 时发生触发硬件中断的新事件，则将该事件输入到队列中。
- ③ 触发硬件中断的新事件会启动分配给此事件的硬件中断 OB。

分配中断触发事件

在设备视图的输入属性中，将中断触发事件分配给 OB。

- 只能将中断触发事件分配给单个 OB。
- 但是，可将 OB 分配给多个中断触发事件。
即，可将上升沿事件和下降沿事件分配给同一个中断 OB，以便对任何输入信号的切换触发相同的响应。
- 每条指令中，已启动的 OB 都可以中断循环 OB。一致的数据访问由最多双字大小来保证。
- 可参数化模块特定的中断触发事件，如输入端的上升沿和下降沿。

- 在中断触发模块的组态中分配中断触发事件和要启动的 OB。但是，在已启动的硬件中断 OB 中，可通过 DETACH 指令覆盖此分配，或者通过 ATTACH 指令将相同的事件分配给不同的 OB。使用该功能可对外部过程信号进行灵活的响应。

参见

硬件中断组织块 (页 317)
 给硬件中断 OB 分配参数 (页 384)
 ATTACH (页 922)
 DETACH (页 923)

7.1.4.4 指令的符号名称和数字名称

符号名称和数字名称

高级指令和全局库中的指令可调用内部以数字标识的功能 (FC)、功能块 (FB)、系统功能 (SFC) 和系统功能块 (SFB)。

此规则同样适用于高级指令中所使用的默认数据结构 (UDT)。

下表列出了对 AS1200-CPU 的高级指令和全局库中的指令的分配，按符号名称或数字名称排序。

表格 7-1 按符号名称排序

符号名称	数字名称
ATH	SFC198
ATTACH	SFC192
CAN_DINT	SFC33
CONDITIONS	UDT1001
CTRL_PWM	SFB122
DETACH	SFC193
DIS_AIRT	SFC41
EN_AIRT	SFC42
HTA	SFC199
MB_COMM_LOAD	FC1080
MB_MASTER	FB1081
MB_PORT	FBT1083
MB_SLAVE	FB1082
◦MC_Halt	FB1100
MC_Home	FB1101
MC_MoveAbsolute	FB1102

符号名称	数字名称
MC_MoveJog	FB1103
MC_MoveRelative	FB1104
MC_MoveVelocity	FB1105
MC_Power	FB1107
MC_Reset	FB1108
PID_Compact	FB1130
PORT_CFG	SFB110
RCV_CFG	SFB112
RCV_PTP	SFB114
RCV_RST	SFB117
RD_LOC_T	SFC154
RD_SYS_T	SFC151
RE_TRIGR	SFC43
S_CONV (DI_STRG)	SFC216
S_CONV (I_STRG)	SFC215
S_CONV (R_STRG)	SFC218
S_CONV (SI_STRG)	SFC214
S_CONV (STRG_DI)	SFC206
S_CONV (STRG_I)	SFC205
S_CONV (STRG_R)	SFC208
S_CONV (STRG_SI)	SFC204
S_CONV (STRG_UDI)	SFC202
S_CONV (STRG_UI)	SFC201
S_CONV (STRG_USI)	SFC200
S_CONV (UDI_STRG)	SFC212
S_CONV (UI_STRG)	SFC211
S_CONV (USI_STRG)	SFC210
SEND_CFG	SFB111
SEND_PTP	SFB113
SGN_GET	SFB115
SGN_SET	SFB116
SRT_DINT	SFC32
STP	SFC46

符号名称	数字名称
T_ADD	SFC159
T_DIFF	SFC158
T_SUB	SFC157
TCON	SFB102
TDISCON	SFB103
TO_AXIS_1	FBT1
TO_AXIS_PTO	FBT2
TRCV	SFB101
TRCV_C	FB1031
TSEND	SFB100
TSEND_C	FB1030
USS_COMM_LOAD	FC1070
USS_DRIVE	FB1071
USS_RPM	FB1072
USS_WPM	FB1073
WR_SYS_T	SFC156

表格 7-2 按数字名称排序

数字名称	符号名称
FB1030	TSEND_C
FB1031	TRCV_C
FB1071	USS_DRIVE
FB1072	USS_RPM
FB1073	USS_WPM
FB1081	MB_MASTER
FB1082	MB_SLAVE
FB1100	○MC_Halt
FB1101	MC_Home
FB1102	MC_MoveAbsolute
FB1103	MC_MoveJog
FB1104	MC_MoveRelative
FB1105	MC_MoveVelocity
FB1107	MC_Power

数字名称	符号名称
FB1108	MC_Reset
FB1130	PID_Compact
FBT1	TO_AXIS_1
FBT2	TO_AXIS_PTO
FBT1083	MB_PORT
FC1070	USS_COMM_LOAD
FC1080	MB_COMM_LOAD
SFB100	TSEND
SFB101	TRCV
SFB102	TCON
SFB103	TDISCON
SFB110	PORT_CFG
SFB111	SEND_CFG
SFB112	RCV_CFG
SFB113	SEND_PTP
SFB114	RCV_PTP
SFB115	SGN_GET
SFB116	SGN_SET
SFB117	RCV_RST
SFB122	CTRL_PWM
SFC32	SRT_DINT
SFC33	CAN_DINT
SFC41	DIS_AIRT
SFC42	EN_AIRT
SFC43	RE_TRIGR
SFC46	STP
SFC151	RD_SYS_T
SFC154	RD_LOC_T
SFC156	WR_SYS_T
SFC157	T_SUB
SFC158	T_DIFF
SFC159	T_ADD
SFC192	ATTACH

数字名称	符号名称
SFC193	DETACH
SFC198	ATH
SFC199	HTA
SFC200	S_CONV (STRG_USI)
SFC201	S_CONV (STRG_UI)
SFC202	S_CONV (STRG_UDI)
SFC204	S_CONV (STRG_SI)
SFC205	S_CONV (STRG_I)
SFC206	S_CONV (STRG_DI)
SFC208	S_CONV (STRG_R)
SFC210	S_CONV (USI_STRG)
SFC211	S_CONV (UI_STRG)
SFC212	S_CONV (UDI_STRG)
SFC214	S_CONV (SI_STRG)
SFC215	S_CONV (I_STRG)
SFC216	S_CONV (DI_STRG)
SFC218	S_CONV (R_STRG)
UDT1001	CONDITIONS

注意

该列表列出了可根据传送的数据类型 (指令 S_CONV) 在指令中调用的任何子级块 (以括号括起来)。请注意, 该列表仅包含以块的方式实现的指令。包含宏或机器命令内部调用的指令不具有块编号。

7.2 创建用户程序

7.2.1 编程基本知识

7.2.1.1 操作系统和用户程序

7.2.1.1 操作系统

功能

每个 CPU 都带有集成的操作系统，用于组织与特定控制任务无关的所有 CPU 功能和顺序。

例如，操作系统任务包括下列各项：

- 处理暖启动 (页 289)
- 更新输入和输出过程映像
- 调用用户程序
- 检测中断和调用中断 OB
- 检测和处理错误
- 管理存储区

操作系统是 CPU 的组件，交付时已安装在其中。

参见

用户程序 (页 309)

7.2.1.1 用户程序

功能

用户程序包含处理特定自动化任务所需的全部功能。

用户程序任务包括：

- 使用启动 OB 检查 (暖) 启动的要求，例如，限位开关是否处在正确位置或安全继电器是否激活。
- 处理过程数据，例如，链接二进制信号，读入并评估模拟值，定义输出的二进制信号以及输出模拟值。
- 响应中断，例如，模拟扩展模块的限值过冲时的诊断错误中断。
- 正常程序执行中的错误处理

用户编写用户程序，并将其装载到 CPU 中。

参见

操作系统 (页 309)

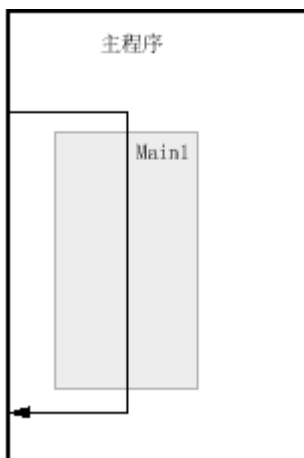
7.2.1.2 用户程序中的块

7.2.1.2 线性编程与结构化编程

线性编程

可以在程序循环 OB 中线性设计处理小型自动化任务解决方案的程序。但我们建议仅对简单程序进行线性编程。

下图所示为一个线性程序示意图：在本例中，“Main1”程序循环 OB 包含整个用户程序。



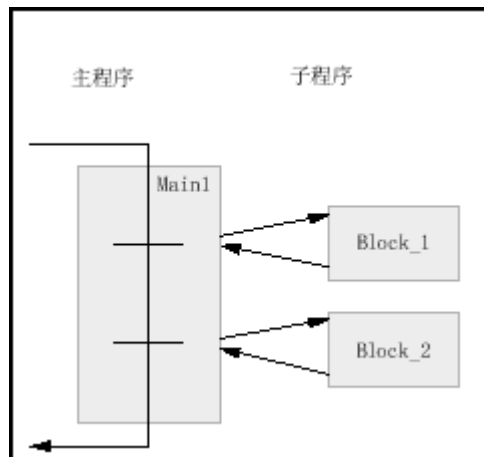
结构化编程

将复杂自动化任务分割成与过程工艺功能相对应或可重复使用的更小的子任务，将更易于对这些复杂任务进行处理和管理。这些子任务在用户程序中以相应的程序段来表示，这些程序段称为块。因此，每个块是用户程序的独立部分。

结构化程序有以下优点：

- 更易于理解大程序。
- 可以标准化各程序段。
- 程序结构更简单。
- 更改程序变得更容易。
- 可分别测试程序段，因而可简化程序排错过程。
- 简化了调试。

下图所示为一个结构化程序示意图：“Main1”程序循环 OB 依次调用一些子程序，它们执行所定义的子任务。



7.2.1.2 块类型

7.2.1.2 块类型概述

块类型

我们提供了不同的块类型来执行自动化系统中的任务。下表给出了可用的块类型：

块类型	简要描述
组织块 (页 312) (OB)	组织块定义用户程序的结构。
功能 (页 320) (FC)	功能包含用于处理重复任务的程序例程。功能没有“存储器”。
功能块 (页 320) (FB)	功能块是一种代码块，它将值永久地存储在背景数据块中，从而即使在块执行完后，这些值仍然可用。
背景数据块 (页 321)	调用背景数据块来存储程序数据时，该背景数据块将分配给功能块。
全局数据块 (页 321)	全局数据块是用于存储数据的数据区，任何块都可以使用这些数据。

7.2.1.2 组织块

7.2.1.2 组织块的基本知识

定义

组织块 (OB) 构成了操作系统和用户程序之间的接口。组织块由操作系统调用，并控制下列操作：

- 自动化系统的启动特性
- 循环程序执行
- 中断驱动的程序执行
- 错误处理

可以对组织块进行编程并同时确定 CPU 的特性。

组织块的使用

在程序中使用组织块时，存在多种选择：

- 启动 OB、程序循环 OB、时间错误中断 OB 和诊断 OB：

在项目中插入这些组织块和对它们进行编程很容易。无须分配参数或调用这些组织块。

- 硬件中断 OB 和循环中断 OB：

将这些组织块插入程序中后，需要为其设置参数。硬件中断 OB 还可以在运行时使用 ATTACH 指令连接到事件或使用 DETACH 再次断开连接。

- 延时中断 OB：

可以在项目中插入延时中断 OB 和对其进行编程。必须使用 SRT_DINT 指令调用它。无须进行参数分配。

组织块的启动信息

多个组织块启动时，操作系统将输出信息，该信息可在用户程序中进行评估。组织块的说明包含有关是否输出信息以及输出什么信息的信息。

参见

程序执行的基础知识 (页 298)

启动组织块 (页 313)

循环程序执行的组织块 (页 313)

用于中断驱动程序执行的组织块 (页 314)

处理时间错误的组织块 (页 319)

7.2.1.2 启动组织块

描述

可以确定 CPU 启动特性的边界条件，例如，“RUN”对应的初始值。为此，请编写启动程序。启动程序由一个或多个启动 OB（OB 编号为 100 或大于等于 200）组成。

启动程序在从“STOP”模式切换到“RUN”模式期间执行一次。输入过程映像中的当前值对于启动程序不可用，也不能设置这些值。

启动 OB 执行完毕后，将读入输入过程映像并启动循环程序。

启动程序的执行没有时间限制。因此，未激活最大循环时间。不能使用时间驱动或中断驱动的组织块。

启动信息

启动 OB 具有以下启动信息：

变量	数据类型	描述
LostRetentive	BOOL	= 1，如果保持性数据存储区已丢失
LostRTC	BOOL	= 1，如果实时时钟已丢失

参见

“STARTUP”工作模式 (页 288)
组织块的基本知识 (页 312)
事件和 OB (页 298)

7.2.1.2 循环程序执行的组织块

简介

要启动程序执行，项目中至少要有一个程序循环 OB。操作系统每个周期调用该程序循环 OB 一次，从而启动用户程序的执行。可以使用多个 OB（OB 编号大于等于 200）。

程序循环 OB 的优先等级为 1。这对应于所有 OB 的最低优先级。任何其它事件类别的事件都可以中断循环程序。

对循环程序执行进行编程

可通过在循环 OB 以及所调用的块中编写用户程序对循环程序执行进行编程。

一旦成功执行启动程序后，就会开始第一次循环程序执行。每次循环程序执行结束后，循环重新开始。

循环程序执行的顺序

一个程序执行周期包括以下步骤：

1. 操作系统启动最大循环时间。
2. 操作系统将输出过程映像中的值写到输出模块。
3. 操作系统读取输入模块的输入状态，并更新输入过程映像。
4. 操作系统处理用户程序并执行程序中包含的运算。
5. 在循环结束时，操作系统执行所有未决的任务，例如加载和删除块，或调用其它循环 OB。
6. 最后，CPU 返回到循环起点，并重新启动监视狗。

另请参见：[过程映像输入/输出 \(页 295\)](#)

中断选项

循环程序执行可被以下事件中断：

- 中断
- 以下情况触发的 STOP 命令
 - 编程设备的运行
 - “STP”指令
- 电源故障
- 发生设备故障或程序出错

启动信息

循环 OB 没有启动信息。

参见

[组织块的基本知识 \(页 312\)](#)
[事件和 OB \(页 298\)](#)

7.2.1.2 用于中断驱动程序执行的组织块

7.2.1.2 有关中断驱动程序执行的信息

功能

循环组织块的处理可以被其它组织块的启动事件中断。这些中断组织块可以是中断 OB 或时间错误中断 OB 等。

在以下两种情况下将调用中断 OB：

- 用户程序部分应该定期执行。
- 用户程序应响应来自过程的外部信号。

如果发生中断，操作系统将确保调用相应的中断 OB。在中断 OB 中对自动化系统如何响应中断进行编程。

中断 OB 只能被优先级更高的 OB 中断，不能被优先级相同或更低的 OB 中断。

中断类型和应用

下表给出了中断类型和中断 OB 的使用实例：

中断类型	中断 OB	OB 数目	应用实例
延时中断 (页 315)	> = 200	< = 4	控制风扇，风扇应在电机关闭后继续运转 20 秒钟关闭。
循环中断 (页 316)	> = 200		扫描闭环控制系统的信号电平。
硬件中断 (页 317)	> = 200	< = 50	报告已达到罐的最大液位。

中断 OB 没有启动信息。

7.2.1.2 延时中断组织块

描述

延时中断 OB 在经过操作系统中一段可组态的延迟时间后启动。在调用 SRT_DINT 指令后开始计延迟时间。

在用户程序中最多可使用 4 个延时中断 OB 或循环 OB (OB 编号大于等于 200)。例如，如果已使用 2 个循环中断 OB，则在用户程序中最多可以再插入 2 个延时中断 OB。

可以使用 CAN_DINT 指令阻止执行尚未启动的延时中断。

延时中断 OB 的功能

在完成将 OB 编号和标识符传送给 SRT_DINT 指令后，操作系统即会在延迟时间过后启动相应的 OB。

要使用延时中断 OB，必须执行以下任务：

- 必须调用指令 SRT_DINT。
- 必须将延时中断 OB 作为用户程序的一部分下载到 CPU。

延迟时间的测量精度为 1 ms。延迟时间到达后可立即再次开始计时。

只有在 CPU 处于“RUN”模式时才会执行延时中断 OB。暖启动将清除延时中断 OB 的所有启动事件。

如果发生以下事件之一，操作系统将调用延时中断 OB：

- 如果操作系统试图启动一个尚未装载的 OB，并且用户在调用 SRT_DINT 指令时指定了其编号。
- 如果在完全执行延时 OB 之前发生下一个延时中断启动事件。

可以使用 DIS_AIRT 和 EN_AIRT 指令来禁用和重新启用延时中断。

注意

如果执行 SRT_DINT 之后使用 DIS_AIRT 禁用中断，则该中断只有在使用 EN_AIRT 启用后才会执行。延迟时间将相应地延长。

启动信息

延时中断 OB 没有启动信息。

参见

SRT_DINT (页 924)

CAN_DINT (页 925)

组织块的基本知识 (页 312)

事件和 OB (页 298)

7.2.1.2 循环中断组织块

描述

循环中断 OB 以周期性时间间隔启动程序，而与循环程序执行无关。循环中断 OB 的启动时间通过时间基数和相位偏移量来指定。

时间基数定义循环中断 OB 启动的时间间隔，并且它是基本时钟周期 1 ms 的整数倍。相位偏移量是与基本时钟周期相比启动时间所偏移的时间。如果使用多个循环中断 OB，当这些循环中断 OB 的时间基数有公倍数时，可以使用该偏移量防止同时启动。

可以指定 1 ms 和 60000 ms 间的时间段。

在用户程序中最多可使用 4 个循环中断 OB 或延时 OB (OB 编号大于等于 200)。例如，如果已使用 2 个延时中断 OB，则在用户程序中最多可以再插入 2 个循环中断 OB。

注意

各循环中断 OB 的执行时间必须明显小于其时间基数。如果尚未执行完循环中断 OB，但由于周期时钟已到而导致执行再次暂停，则将启动时间错误 OB。稍后将执行导致错误的循环中断或将其放弃。

使用相位偏移的实例

已在用户程序中插入 2 个循环中断 OB：

- 循环中断 OB1
- 循环中断 OB2

对于循环中断 OB1，已设置时间基数为 20 ms；对于循环中断 OB2，已设置时间基数为 100 ms。时间基数 100 ms 到期后，循环中断 OB1 第五次到达启动时间，而循环中断 OB2 是第一次到达启动时间。然而要执行循环中断 OB 偏移，则请为其中一个循环中断 OB 输入相位偏移量。

启动信息

循环中断 OB 没有启动信息。

参见

组织块的基本知识 (页 312)

给循环中断 OB 分配参数 (页 385)

事件和 OB (页 298)

7.2.1.2 硬件中断组织块

描述

可以使用硬件中断 OB 来响应特定事件。只能将触发报警的事件分配给一个硬件中断 OB。而一个硬件中断 OB 可以分配给多个事件。

高速计数器和输入通道可以触发硬件中断。对于将触发硬件中断的各高速计数器和输入通道，需要组态以下属性：

- 将触发硬件中断的过程事件，例如，高速计数器的计数方向改变
- 分配给该过程事件的硬件中断 OB 的编号

最多可使用 50 个硬件中断 OB (OB 编号大于等于 200)，它们在用户程序中彼此独立。

硬件中断 OB 的功能

触发硬件中断后，操作系统将识别输入通道或高速计数器并确定所分配的硬件中断 OB。

如果没有其它中断 OB 激活，则调用所确定的硬件中断 OB。如果已经在执行其它中断 OB，硬件中断将被置于与其同优先等级的队列中。所分配的硬件中断 OB 完成执行后，即确认了该硬件中断。

如果在对硬件中断进行标识和确认的这段时间内，在同一模块中发生了触发硬件中断的另一事件，则应用以下规则：

- 如果该事件发生在先前触发硬件中断的通道中，则不会触发另一个硬件中断。只有确认当前硬件中断后，才能触发其它硬件中断。
- 如果该事件发生在另一个通道中，将触发硬件中断。

只有在 CPU 处于“RUN”模式时才会调用硬件中断 OB。

启动信息

硬件中断 OB 没有启动信息。

参见

组织块的基本知识 (页 312)
 给硬件中断 OB 分配参数 (页 384)
 事件和 OB (页 298)

7.2.1.2 诊断错误中断组织块**描述**

可以为具有诊断功能的模块启用诊断错误中断功能，使模块能检测到 I/O 状态变化。因此模块会在发生以下情况时触发诊断错误中断：

- 出现故障（进入事件）
- 故障不再存在（离开事件）

如果没有其它中断 OB 激活，则调用诊断错误中断 OB。如果已经在执行其它中断 OB，诊断错误中断将置于同优先级的队列中。

在用户程序中只能使用一个诊断错误中断 OB。

启动信息

诊断错误中断 OB 具有以下启动信息：

变量	数据类型	描述
IO_state	WORD	包含具有诊断功能的模块的 I/O 状态。
laddr	HW_ANY	HW-ID
Channel	UINT	通道编号
multi_error	BOOL	= 1，如果有多个错误

IO_state 变量

下表列出了 IO_state 变量所能包含的可能 I/O 状态：

IO_state	描述
位 0	组态正确： = 1，如果组态正确 = 0，如果组态不再正确
位 4	错误： = 1，如果存在错误，例如断路 = 0，如果错误不再存在

IO_state	描述
位 5	组态不正确： = 1，如果组态不正确 = 0，如果组态再次正确
位 6	无法访问 I/O： = 1，如果发生了 I/O 访问错误 在这种情况下，laddr 包含存在访问错误的 I/O 的硬件标识符。 = 0，如果可以再次访问该 I/O

参见

组织块的基本知识 (页 312)
事件和 OB (页 298)

7.2.1.2 处理时间错误的组织块

描述

如果发生以下事件之一，操作系统将调用时间错误中断 OB：

- 循环程序超出最大循环时间。
- 被调用 OB 当前正在执行（对于延时中断 OB 和循环中断 OB 有这种可能）。
- 中断 OB 队列发生溢出。
- 由于中断负载过大而导致中断丢失。

如果没有规划时间错误中断 OB，将执行相应的系统响应。

另请参见：[事件和 OB \(页 298\)](#)

两次重复超出最大循环时间属于严重错误，将不会调用 OB，而是 CPU 转到 STOP。使用 RE_TRIGR 指令重新启动 CPU 的循环监视可以避免第二次超出。

在用户程序中只能使用一个时间错误中断 OB。

启动信息

时间错误中断 OB 具有以下启动信息：

变量	数据类型	描述
fault_id	BYTE	0x01：超出最大循环时间 0x02：仍在执行被调用 OB 0x07：队列溢出 0x09：中断负载过大导致中断丢失
csg_OBnr	OB_ANY	出错时要执行的 OB 的编号

变量	数据类型	描述
csg_prio	UINT	出错时要执行的 OB 的优先级

参见

组织块的基本知识 (页 312)

7.2.1.2 功能

定义

功能 (FC) 是无存储器的代码块。因此，在功能执行完后，临时变量中的数据将丢失。功能可以使用全局数据块永久性存储数据。

应用

功能包含其它代码块调用该功能时执行的程序。例如，可以将功能用于下列目的：

- 将功能值返回给调用块，例如，数学函数
- 执行工艺功能，例如，通过位逻辑运算进行单个的控制

可以在程序中的不同位置多次调用同一个功能。因此，功能块简化了对重复发生的复杂功能的编程。

参见

创建功能和功能块 (页 375)

7.2.1.2 功能块

定义

功能块是一种代码块，它将值永久地存储在背景数据块中，从而即使在块执行完后，这些值仍然可用。功能块将其输入、输出和输入/输出参数永久保存在背景数据块中。因此，在块执行后这些参数仍然可用。所以功能块也称为“有存储器”的块。

应用

功能块包含总是在其它代码块调用该功能块时执行的子例程。可以在程序中的不同位置多次调用同一个功能块。因此，功能块简化了对重复发生的复杂功能的编程。

功能块的实例

功能块的调用称为实例。

功能块的各个实例都分配有背景数据块，其中包含功能块使用的数据。

注意

为避免在使用功能块时出错，请参见部分“[块调用时的参数传递 \(页 326\)](#)”

参见

[一个功能调用另一个功能 \(页 328\)](#)
[创建功能和功能块 \(页 375\)](#)
[多重背景 \(页 324\)](#)
[背景数据块 \(页 321\)](#)

7.2.1.2 背景数据块**定义**

功能块的调用称为实例。将背景数据块分配给传送参数的每个功能块调用。该背景数据块用作数据存储。实参和功能块的静态数据存储在其中。

背景数据块的最大大小因 CPU 的不同而各异。功能块中声明的变量决定背景数据块的结构。

另请参见：[以单个背景或多重背景的方式调用功能块 \(页 324\)](#)

参见

[创建数据块 \(页 376\)](#)
[在 LAD 中插入块调用 \(页 423\)](#)
[在 FBD 中插入块调用 \(页 421\)](#)

7.2.1.2 全局数据块**定义**

数据块与代码块相反，它不含指令，只用于存储用户数据。因此，数据块包含用户程序将使用的变量数据。全局数据块存储所有其它块都可使用的数据。

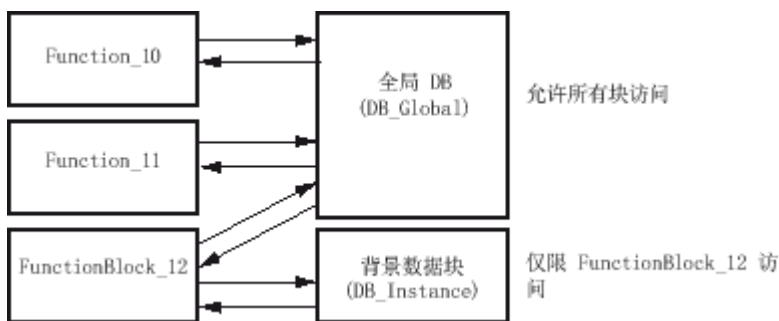
数据块的最大大小因 CPU 的不同而各异。可以以自己喜欢的方式定义全局数据块的结构。

用户程序中的全局数据块

如果调用代码块，则代码块可以数据块的形式打开存储区。当数据块已关闭或相应代码块的执行结束时，数据块中包含的数据不会被删除。

每个功能块、功能或组织块都可以从全局数据块中读取数据或向其中写入数据。即使在退出数据块后，这些数据仍然会保存在其中。可以同时打开一个全局数据块和一个背景数据块。

下图所示为不同的数据块访问：



参见

创建数据块 (页 376)

7.2.1.3 块调用

7.2.1.3 块调用的原理

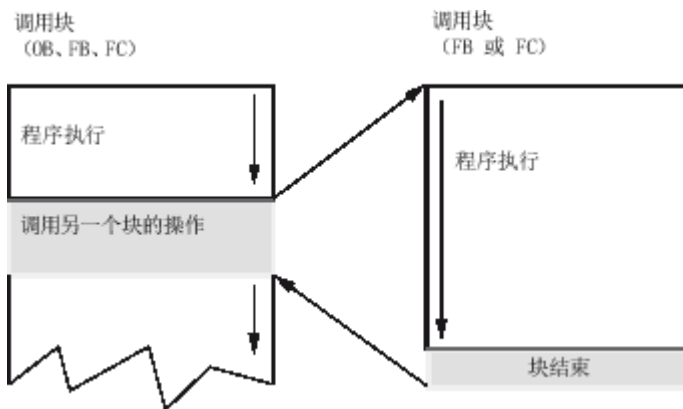
块调用的功能

要执行用户程序中的块，必须通过其它块对它们进行调用。

当一个块调用另一个块时，将执行被调用块的操作。只有完成被调用块的执行后，才会继续执行调用块。并且继续执行块调用后的操作。

调用块时，必须给块接口中的参数赋值。通过提供输入参数，用户可以指定用于执行块的数据。通过提供输出参数，用户可以指定执行结果的保存位置。

下图所示为用户程序中块调用的顺序：



参见

调用层级 (页 323)

单个背景和多重背景的基本知识 (页 323)

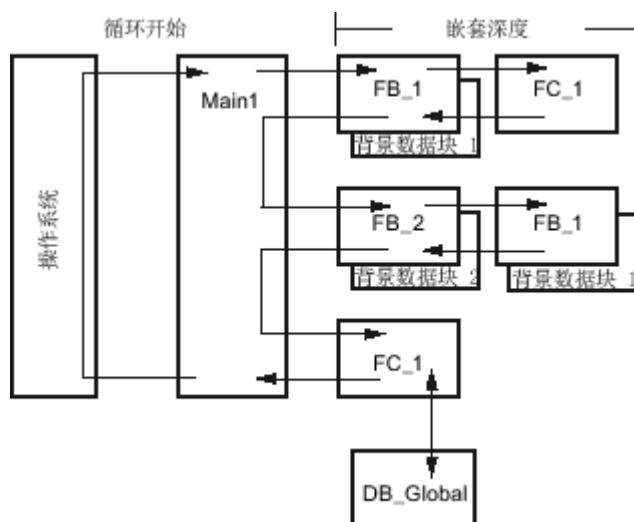
块调用时的参数传递 (页 326)

7.2.1.3 调用层级

定义

块调用的顺序和嵌套称为调用层级。可用的嵌套深度取决于 CPU。

下图所示为在一个执行周期内的块调用顺序和嵌套深度实例：



参见

单个背景和多重背景的基本知识 (页 323)
块调用的原理 (页 322)

7.2.1.3 以单个背景或多重背景的方式调用功能块

7.2.1.3 单个背景和多重背景的基本知识

单个背景和多重背景的使用

功能块 (FB) 将其数据存储存储在背景数据块中。对功能块的调用进行编程时，需要将背景数据块分配给该功能块。块参数值和静态局部数据存储在所分配的背景数据块中。

可如下分配背景数据块：

- 以单个背景的方式调用：
 - 一个背景数据块分配给一个功能块的一个实例
- 以多重背景的方式调用：
 - 一个背景数据块分配给一个功能块的多个实例
 - 一个背景数据块分配给不同功能块的多个实例

参见

块调用的原理 (页 322)
访问多重背景的数据 (页 430)
多重背景 (页 324)
单个背景 (页 324)
调用层级 (页 323)

7.2.1.3 单个背景

定义

分配有自己的背景数据块的功能块调用称为单个背景数据块。

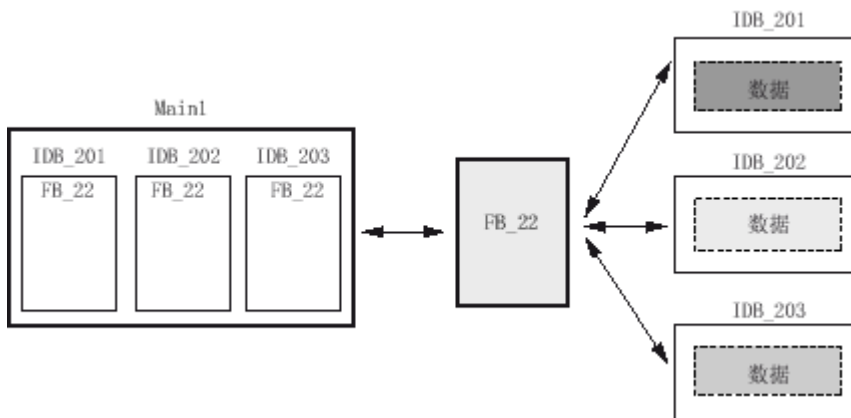
通过分配背景数据块，可以指定要处理哪些数据。可以为每个调用分配不同的背景数据块，以此提高块的可复用性。

单个背景的实例

可以使用一个功能块控制多台电机。为实现此目的，需要为执行电机控制的每个功能块调用分配一个不同的背景数据块。

不同电机的不同数据（例如，速度、加速时间、总运行时间）保存在不同的背景数据块中。不同的电机将根据所分配的背景数据块进行控制。

下图显示了怎样使用一个功能块和三个不同的数据块来控制三台电机。



参见

单个背景和多重背景的基本知识 (页 323)
多重背景 (页 324)
访问多重背景的数据 (页 430)

7.2.1.3 多重背景

如果 S7-CPU 的性能数据限制可用的背景数据块数目，则可以使用多重背景。

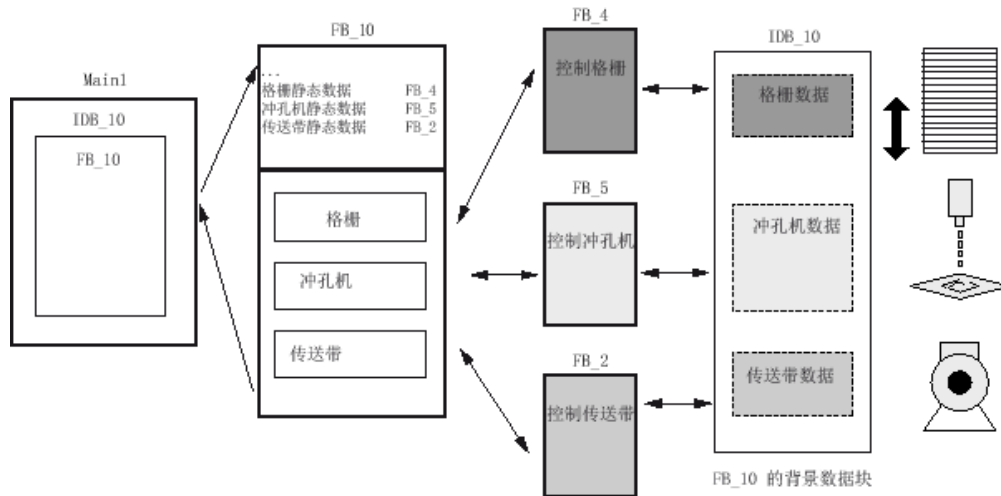
定义

多重背景允许被调用功能块将其数据存储在被调用功能块的背景数据块中。

这样便可将实例数据集中放在一个背景数据块中，从而更有效地使用可用的背景数据块。

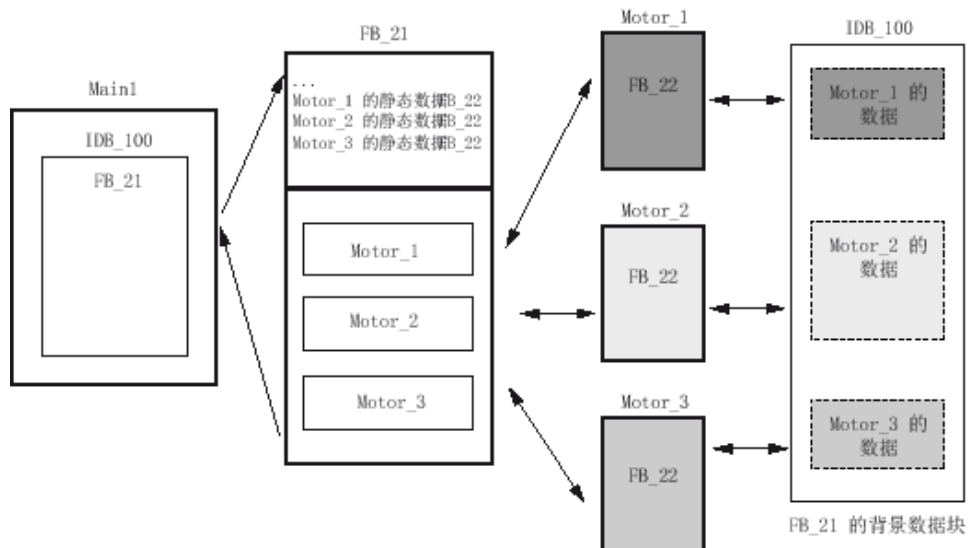
一个背景数据块用于不同功能块的实例

下图显示了多个不同的功能块如何将数据存储在一个调用块中。FB 10 连续调用 FB 4、FB 5 和 FB 2。这些被调用块都将数据存储在被调用块的数据块 DB 10 中。



一个背景数据块用于一个功能块的多个实例

下图显示了在多个实例中调用的一个功能块如何在一个背景数据块中存储所有实例的数据。



功能块 FB 21 调用 FB 22 的三个实例。这些实例是“Motor_1”、“Motor_2”和“Motor_3”。每个调用使用不同的实例数据。但所有实例数据都位于同一个背景数据块 (DB 100) 中。

注意事项

如果在 CPU 运行时希望进行在线修改，则切勿使用多重背景数据块。只有使用背景数据块，才能确保无扰动的重新装载。

参见

背景数据块 (页 321)
单个背景和多重背景的基本知识 (页 323)
单个背景 (页 324)
访问多重背景的数据 (页 430)

7.2.1.3 块调用时的参数传递

7.2.1.3 参数传递

参数传递的原理

调用功能或功能块时，在调用块和被调用块间将发生数据交换。被调用块的接口包含用于执行该块的参数。这些参数称为形参。它们只是在块被调用时传递给该块的参数的占位符。调用块时传递给块的参数称为实参。



警告

对被调用块进行编程时，请确保还写入了声明为输出参数的参数。否则，输出值为随机值！

7.2.1.3 功能的参数分配

功能参数的分配

功能 (FC) 没有可以保存块参数值的数据存储器。因此，调用功能时，必须给所有形参分配实参。

未分配的输出参数

如果没有给功能的输出参数赋值，则返回给被调用块的值可以是随机值。

未分配的输入/输出参数

对于功能的输入/输出参数，输出参数不会是随机值，因为如果未对该参数执行写操作，将保留其中的旧输出值或输入值。但仍应注意上述信息，以确保不会意外地处理这些旧值。

建议

分配参数时请注意下列规则：

- 请确保始终写入输出参数，而不管块中的程序路径为何。此处要特别注意调用语句。
- 请注意，置位和复位命令取决于逻辑运算的结果。如果输出参数值通过这些命令和 RLO = 0 来确定，则不会生成值。

7.2.1.3 功能块的参数分配

功能块参数的分配

功能块 (FB) 的参数值保存在背景数据块中。

未分配的输入、输出和输入/输出参数

如果没有给功能块的输入、输出或输入/输出参数赋值，将使用背景数据块中存储的值。

与功能 (FC) 相反，不会使用随机值。但存在意外继续处理旧值的危险。因此，建议分配所有块参数。

某些情况下，必须分配参数。

下表显示了必须分配实参的功能块参数：

参数	基本数据类型	复杂数据类型	参数类型
输入	可选	可选	必需
输出	可选	可选	必需
InOut	可选	必需	必需

给形参赋初值

可以给功能块接口中的形参赋初值。这些值将传送到相关的背景数据块。

如果在调用指令中没有将实参分配给形参，将使用当前存储在背景数据块中的值。

下表显示了可以分配初始值的变量：

变量	基本数据类型	复杂数据类型	参数类型
输入	可选	可选	不可能
输出	可选	可选	不可能
InOut	可选	不可能	不可能
静态	可选	可选	不可能
临时	不可能	不可能	不可能

7.2.1.3 参数传递时可用的数据类型

7.2.1.3 规则

块间参数传递的规则

可将变量或常量指定为实参。但会有一些限制。例如，不能给输出和输入/输出参数分配常数值，因为输出和输入/输出参数的用途是接收变量值（例如，计算结果）。

块参数赋值的规则

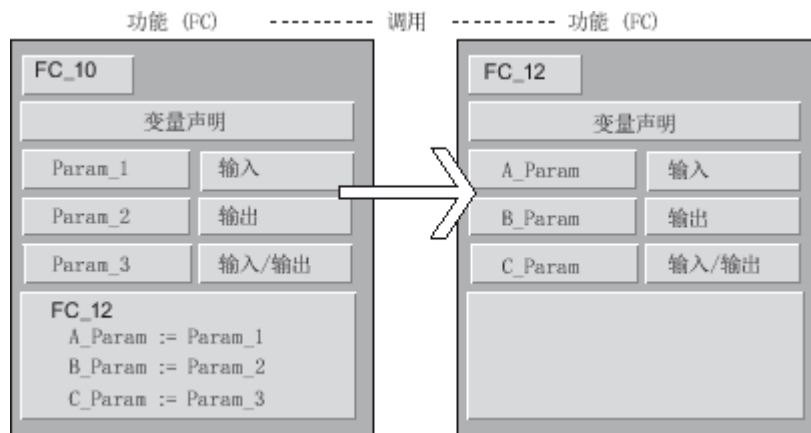
下表显示了可以分配给块参数的数据类型：

声明类型	PLC 变量	块接口中的变量	常数
输入	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型	基本数据类型
输出	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型	-
InOut	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型	-

7.2.1.3 一个功能调用另一个功能

一个功能调用另一个功能时可用的数据类型

可以将调用功能的形参分配给被调用功能的形参。下图显示了将功能 FC 10 的形参作为实参分配给功能 FC 12：



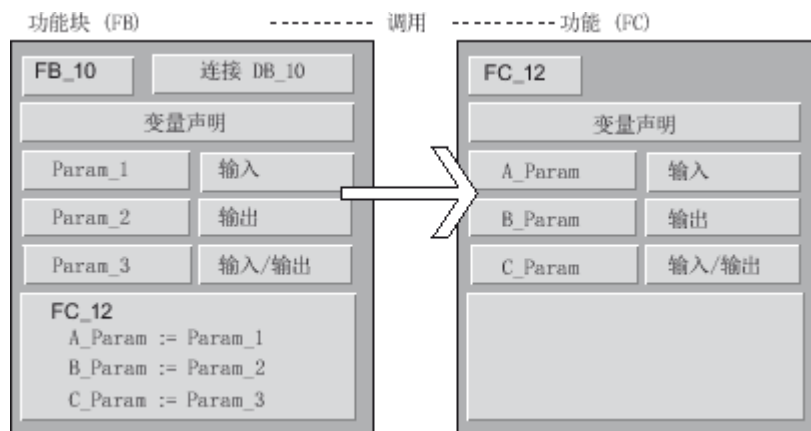
将一个功能的形参分配给另一个功能的形参时会受到限制。下表显示了一个功能调用另一个功能时可用的数据类型：

FC → FC	输入	InOut	输出
输入	基本数据类型 复杂数据类型	-	-
InOut	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型
输出	-	-	基本数据类型 复杂数据类型

7.2.1.3 功能块对功能的调用

功能块调用功能时可用的数据类型

可以将调用功能块的形参分配给被调用功能的形参。下图显示了将功能块 FB 10 的形参作为实参分配给功能 FC 12：



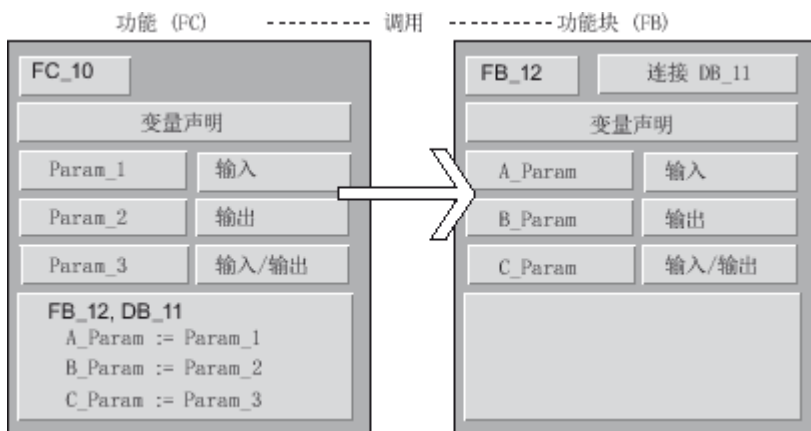
将功能块的形参分配给功能的形参时会受到限制。例如，某些参数类型的参数不能作为实参分配。下表显示了功能块调用功能时可用的数据类型：

FB → FC	输入	InOut	输出
输入	基本数据类型 复杂数据类型	-	-
InOut	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型
输出	-	-	基本数据类型 复杂数据类型

7.2.1.3 通过功能调用功能块

功能调用功能块时可用的数据类型

可以将调用功能的形参分配给被调用功能块的形参。下图显示了将功能 FC 10 的形参作为实参分配给功能块 FB 12：



将功能的形参分配给功能块的形参时会受到限制。例如，复杂数据类型的参数不能作为实参分配。

下表显示了功能调用功能块时可用的数据类型：

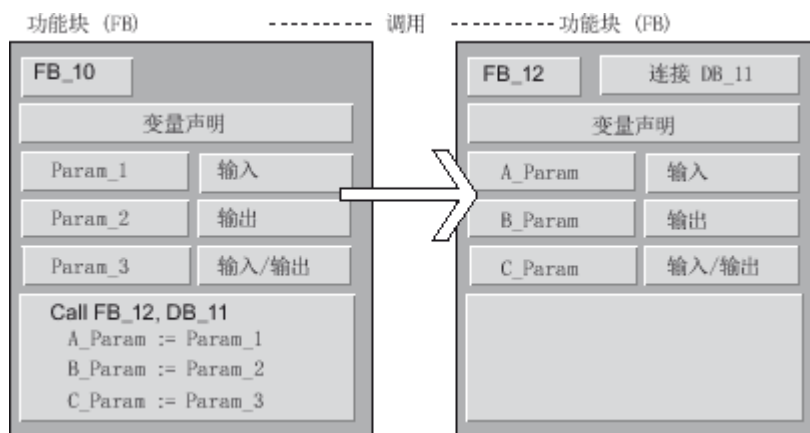
FC → FB	输入	InOut	输出
输入	基本数据类型 复杂数据类型	-	-
InOut	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型

输出	-	-	基本数据类型 复杂数据类型
----	---	---	------------------

7.2.1.3 通过一个功能块调用另一个功能块

一个功能块调用另一个功能块时可用的数据类型

可以将调用功能块的形参分配给被调用功能块的形参。下图显示了将功能块 FB 10 的形参作为实参分配给功能块 FB 12：



将一个功能块的形参分配给另一个功能块的形参时会受到限制。例如，不能将复杂数据类型的输入或输出参数作为实参分配给被调用功能块的输入或输出参数。

下表显示了一个功能块调用另一个功能块时可用的数据类型：

FB → FB	输入	InOut	输出
输入	基本数据类型 复杂数据类型	-	-
InOut	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型	基本数据类型 复杂数据类型
输出	-	-	基本数据类型 复杂数据类型

7.2.1.4 EN/ENO 机制

7.2.1.4 EN/ENO 机制的基本知识

描述

有些运算会引起运行错误，这些错误可能会要求程序中中止。为了防止此类程序中中止，受影响的运算需要有一个使能输入 (EN) 和一个使能输出 (ENO)。借助该 EN/ENO 机制，可以查询和响应块或操作中的运行错误。CPU 内部状态字用于传送执行结果。如果运算完成且没有错误，则置位使能输出 ENO。如果发生错误，EN 和 ENO 的值将不同。

- ENO <> EN (功能框错误)
发生运行错误。
- ENO = EN (功能框错误 = 0)
未发生运行错误。

使用

EN/ENO 机制用于以下运算：

- 数学运算
- 传送和转换运算
- 移位和循环移位运算
- 字逻辑指令

EN/ENO 机制不用于以下运算：

- 位逻辑运算
- 比较器
- 计数器
- 定时器
- 程序控制

这些运算不会引起要求程序中中止的运行错误。因此，在这种情况下不需要 EN/ENO 机制。

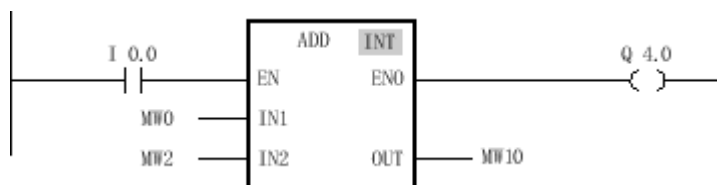
参见

使用 EN/ENO 机制的实例 (页 332)

7.2.1.4 使用 EN/ENO 机制的实例

描述

下图所示为具有 EN/ENO 保护电路的 ADD 指令：



在常开触点后，RLO 包含此前的逻辑运算结果：

- 若 RLO 为“0”，则不执行加法运算。EN 和 ENO 的信号状态都会是“0”。
- 若 RLO 为“1”，则 EN 也为“1”，并且会执行加法运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

ADD : 加 (页 663)

7.2.1.5 符号编程

7.2.1.5 符号编程

符号编程

编程期间将使用操作数和变量，例如，输入、输出和位存储器。CPU 根据数字绝对地址来识别这些操作数。该绝对地址由操作数 ID 和地址构成（例如，Q 4.0、I 1.1、M 2.0）。

但如果要使程序布局更加清晰和简化故障排除，在编程时应使用代表操作数和变量的符号名。

指定符号名：

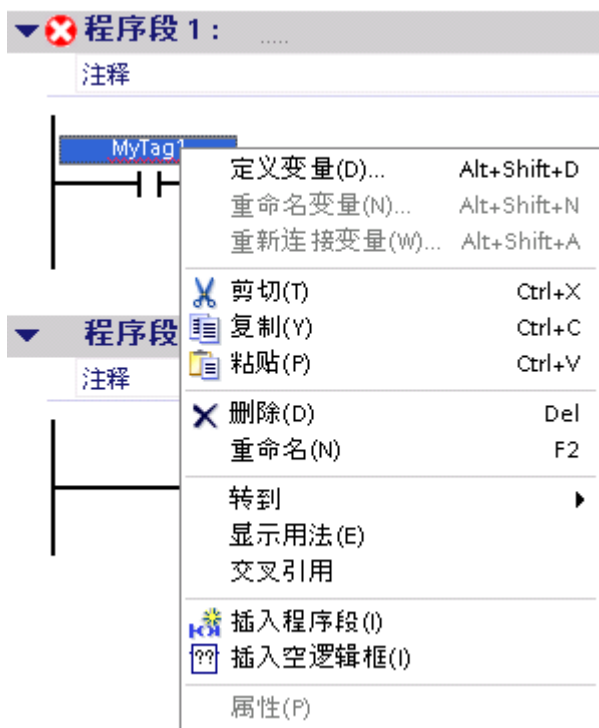
- 在 PLC 变量表中为全局变量定义符号名。
- 在块接口中，为代码块或全局数据块中的局部变量指定符号名。

例如，可以将符号名“Motor_on”分配给输入信号 I 1.0。

编程期间的支持

编程时可先使用符号操作数，以后再分配绝对地址。即使分配没有完成，也仍可保存程序。

尚未分配绝对地址的操作数会以红色波浪下划线突出显示。要分配绝对地址，请选择操作数并单击快捷菜单中的“定义变量”(Define tag)。



参见

- PLC 变量和局部变量 (页 337)*
- 显示符号地址和绝对地址 (页 334)*
- 仅块的符号寻址 (页 335)*
- 仅激活新块的符号寻址 (页 336)*

7.2.1.5 显示符号地址和绝对地址

有以下选项可用于在程序编辑器中显示操作数：

- 符号表示法
在程序中显示符号操作数。如果将鼠标指针放在操作数上，将在工具提示中显示相应的绝对地址。
- 绝对表示法
在程序中显示绝对地址。在工具提示中显示相应的符号操作数。
- 符号和绝对表示法
在程序中显示符号操作数和绝对地址。

要求

程序编辑器已打开。

步骤

要改变操作数的表示法，请按下列步骤操作：

1. 在程序编辑器工具栏中单击“启用/禁用绝对操作数”(Absolute operands on/off) 按钮。
每次单击该按钮，表示法和按钮上的符号将随之改变。

或者：

1. 在程序编辑器工具栏中单击“启用/禁用绝对操作数”(Absolute operands on/off) 按钮旁的小箭头。
将显示一个下拉列表。
2. 从该下拉列表中选择所需的表示法。
按钮上的符号将随之改变。

参见

符号编程 (页 333)
PLC 变量和局部变量 (页 337)

7.2.1.5 仅块的符号寻址

7.2.1.5 仅块的符号寻址

变量的符号声明和寻址

可以两种不同的方式声明块中的变量：

- 符号声明

该声明仅包含一个符号名，且块中没有固定寻址。变量的绝对地址在编译期间动态传递并且不会在块接口中显示。变量以最佳利用存储区容量的方式存储在可用存储区中。只能以符号形式对所声明的变量进行符号寻址。例如，可按如下方式访问“Data”数据块中的“Fill Level”变量：

"Data".Fill Level

- 符号和绝对声明

在块中，声明仅包含一个符号名和固定的绝对地址。例如，该地址显示在块接口的“Offset”列中。

可以以符号或绝对方式对这些变量进行寻址。例如，可按如下方式访问“Data”数据块中的“Fill Level”变量：

"Data".Fill Level 或 *%DB1.DBX0.0*

纯符号声明的优点

对变量的纯符号声明具有以下优点：

- 数据以最适合所用 CPU 的方式结构化和排列。这样可提高 CPU 的性能。
- 您可以将特定的各变量定义为有保持性。对于绝对声明，保持性设置适用于所有块变量。

一个程序中包含符号寻址的块和非符号寻址的块

在一个程序中，用户可以任意组合这两类块。

参见

符号编程 (页 333)
保持性存储区 (页 295)

7.2.1.5 仅激活新块的符号寻址

简介

可以为程序中新建的所有代码块和全局数据块集中设置变量的符号访问。所有新建的块之后将使用该默认设置来创建。该设置不会影响背景数据块，因为其会接管来自更高级代码块的访问。该设置也不会影响现有的块或移植的块。

通过设置符号访问，可以纯符号方式声明块变量，而无需绝对地址信息。此外，符号访问还允许用户设置块中各变量的保持特性。

步骤

要为程序中的所有新块设置符号访问，请按以下步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
工作区中将显示“设置”(Settings) 窗口。
2. 在区域导航中，选择“PLC 编程”(PLC Programming) 组。
3. 在“新块的默认设置”(Default settings for new blocks) 组中，选中“仅符号访问”(Symbolic access only) 复选框。

结果

为程序中的所有新块激活了符号访问。可以专门为块中的各变量设置保持特性。

参见

仅块的符号寻址 (页 335)
符号编程 (页 333)
保持性存储区 (页 295)

7.2.1.6 变量的使用

7.2.1.6 在程序中使用变量

定义

变量的定义是在程序中内容可变的数据值。变量由操作数（例如，M 3.1）和数据类型（例如，BOOL）组成，并用符号表示（例如，BELT_ON）。

程序中变量的使用

使用变量使程序变得更灵活。例如，在块接口中声明变量后，对于每个块调用，即可给变量分配不同的值。从而可以重复使用已编程的块，用于实现多种用途。

参见

PLC 变量和局部变量 (页 337)

保留关键字 (页 338)

7.2.1.6 PLC 变量和局部变量

PLC 变量和局部变量

可以为程序定义适用范围不同的变量：

- PLC 变量在 CPU 的所有区域都适用
- 局部变量仅适用于定义这些变量的块。

下表显示了 PLC 变量和局部变量间的区别：

	PLC 变量	局部变量
范围	<ul style="list-style-type: none"> • 在整个 CPU 中有效。 • CPU 中的所有块均可使用。 • 在所有块中具有相同含义。 • 该名称在 CPU 中唯一。 	<ul style="list-style-type: none"> • 仅在定义它们的块中有效。 • 可以在其它块中使用相同的变量名以实现不同用途。
可用的字符	<ul style="list-style-type: none"> • 字母、数字、特殊字符 • 不可使用引号。 	<ul style="list-style-type: none"> • 字母、数字、特殊字符 • 如果使用特殊字符，必须用引号将变量括起来。
使用	可以为以下对象定义 PLC 变量： <ul style="list-style-type: none"> • I/O 信号 (I、IB、IW、ID、Q、QB、QW、QD) • 位存储器 (M、MB、MW、MD) 	可以为以下对象定义局部变量： <ul style="list-style-type: none"> • 块参数 (输入、输出以及输入/输出参数)， • 块的静态数据 • 块的临时局部数据
定义位置	PLC 变量表	块接口
表示法	PLC 变量显示在引号中。 实例：“Anna”。	块接口中的局部变量使用前缀 # 表示。 实例：#Berta。

参见

符号编程 (页 333)

在程序中使用变量 (页 336)

保留关键字 (页 338)

显示符号地址和绝对地址 (页 334)

7.2.1.6 保留关键字

SIMATIC 识别一系列关键字，这些关键字有固定的定义且在程序中有特定含义。不应将这些关键字用作变量名。

如果必须将这些关键字用作变量名，则首先必须在 PLC 变量表或块接口中声明它。不能在程序中直接输入。

保留关键字表格

下表列出了所有保留关键字。

关键字 德语助记符	关键字 英语助记符	描述
A	Q	输出，位
AB	QB	输出，字节
AD	QD	输出，双字
ANY	ANY	ANY 数据类型的标志
AR1	AR1	地址寄存器 1
AR2	AR2	地址寄存器 2
ARRAY	ARRAY	引入数组规范，其后是“[”和“]”括起来的下标列表
AUTHOR	AUTHOR	作者名、公司名、部门名或其它名称 (最多 8 个字符，不含空格)
AW	QW	输出，字
AX	QX	输出，位
B	B	字节
BEGIN	BEGIN	引入代码块的指令部分或数据块的初始化部分
BIE	BR	二进制结果
BOOL	BOOL	二进制数据的基本数据类型
BY	BY	FOR 循环的增量
BYTE	BYTE	基本数据类型
CALL	CALL	调用
CASE	CASE	引入 Case 指令
CHAR	CHAR	基本数据类型
CODE_VERSION1	CODE_VERSION1	标签，可确定 FB 是否具有多重背景功能。如果希望声明多重背景，FB 不得具有该特性。
CONST	CONST	开始常量声明

关键字 德语助记符	关键字 英语助记符	描述
CONTINUE	CONTINUE	在 SCL 中退出循环的指令
DATA_BLOCK	DATA_BLOCK	引入数据块
DATE	DATE	用于日期的基本数据类型
DATE_AND_TIME	DATE_AND_TIME	用于日期和时间的复杂数据类型
DB	DB	数据块, 位
DBB	DBB	数据字节
DBD	DBD	数据双字
DBLG	DBLG	数据块长度
DBNO	DBNO	数据块编号
DBW	DBW	数据字
DBX	DBX	数据位
DI	DI	数据块, 用“DB OPEN” 打开
DIB	DIB	数据字节
DID	DID	数据双字
DILG	DILG	背景数据块长度
DINO	DINO	背景数据块编号
DINT	DINT	用于双精度整数的基本数据类型
DIW	DIW	数据字
DIX	DIX	数据位
DO	DO	在 FOR 和 WHILE 指令中引入指令部分
DT	DT	用于日期和时间的的基本数据类型
DWORD	DWORD	用于双字的基本数据类型
E	I	输入,(通过过程映像), 位
EB	IB	输入,(通过过程映像), 字节
ED	ID	输入,(通过过程映像), 双字
ELSE	ELSE	IF 和 CASE 指令中的选择分支
ELSIF	ELSIF	IF 指令的选择条件
EN	EN	EN/ENO 机制的系统操作数
ENO	ENO	EN/ENO 机制的系统操作数
END_CASE	END_CASE	结束 CASE 指令
END_DATA_BLOCK	END_DATA_BLOCK	结束数据块

关键字 德语助记符	关键字 英语助记符	描述
END_FOR	END_FOR	结束 FOR 指令
END_FUNCTION	END_FUNCTION	结束功能
END_FUNCTION_BLOCK	END_FUNCTION_BLOCK	结束功能块
END_IF	END_IF	结束 IF 指令
END_ORGANIZATION_BLOCK	END_ORGANIZATION_BLOCK	结束组织块
END_REPEAT	END_REPEAT	结束 REPEAT 指令
END_STRUCT	END_STRUCT	结束结构说明
END_SYSTEM_ FUNCTION	END_SYSTEM_ FUNCTION	结束系统功能
END_SYSTEM_ FUNCTION_BLOCK	END_SYSTEM_ FUNCTION_BLOCK	结束系统功能块
END_TYPE	END_TYPE	结束 UDT
END_VAR	END_VAR	结束声明块
END_WHILE	END_WHILE	结束 WHILE 指令
EW	IW	输入,(通过过程映像), 字
EXIT	EXIT	在 SCL 中退出循环的指令
FALSE	FALSE	预定义的布尔型常数: 逻辑条件为假, 值等于 0
FAMILY	FAMILY	块系列名称: 例如, 闭环控制器 (最多 8 个不含空格的字符)
FB	FB	功能块
FC	FC	功能
FOR	FOR	引入 FOR 指令
FUNCTION	FUNCTION	引入功能
FUNCTION_BLOCK	FUNCTION_BLOCK	引入功能块
GOTO	GOTO	引入 GOTO 指令
IF	IF	引入 IF 指令
INT	INT	用于单精度整数的基本数据类型
KNOW_HOW_PROTECT	KNOW_HOW_PROTECT	块保护; 用此选项编译的块不允许查看指令部分。
L	L	局部数据位
LB	LB	局部数据字节
LD	LD	局部数据双字
LW	LW	局部数据字

关键字 德语助记符	关键字 英语助记符	描述
M	M	位存储器，位
MB	MB	位存储器，字节
MD	MD	位存储器，双字
MOD	MOD	求模运算符
MW	MW	位存储器，字
NAME	NAME	块名称（最多 8 个字符）
NETWORK	NETWORK	程序段
NOT	NOT	逻辑取反
OB	OB	组织块
OF	OF	引入数据类型规范/引入 CASE 指令的指令部分
OR	OR	“或”逻辑运算逻辑表达式
ORGANIZATION_ BLOCK	ORGANIZATION_ BLOCK	引入组织块
OS	OS	保存溢出
OV	OV	溢出
PA	PQ	输出（直接输出），位
PAB	PQB	输出（直接输出），字节
PAD	PQD	输出（直接输出），双字
PAW	PQW	输出（直接输出），字
PE	PI	输入（直接输入），位
PEB	PIB	输入（直接输入），字节
PED	PID	输入（直接输入），双字
PEW	PIW	输入（直接输入），字
POINTER	POINTER	指针数据类型，只允许在参数块的参数声明中使用
READ_ONLY	READ_ONLY	用于数据块的写保护；其数据只能读取，不能修改。
REAL	REAL	基本数据类型
REPEAT	REPEAT	引入 REPEAT 指令
RET_VAL	RET_VAL	返回值
RETURN	RETURN	SCL 中的 RETURN 指令
S5T	S5T	S5 数据类型的语法

关键字 德语助记符	关键字 英语助记符	描述
S5TIME	S5TIME	提供时间信息的基本数据类型，特殊的 S5 格式
S7_	S7_	用于系统属性的关键字（基本软件包）
SDB	SDB	系统数据块
SFB	SFB	系统功能块
SFC	SFC	系统功能
STANDARD	STANDARD	
STRING	STRING	用于字符串的数据类型
STRUCT	STRUCT	引入结构规范，其后是元素列表
STW	STW	状态字
SYSTEM_FUNCTION	SYSTEM_FUNCTION	系统功能
SYSTEM_FUNCTION_BLOCK	SYSTEM_FUNCTION_BLOCK	系统功能块
T	T	时间元件（定时器）
THEN	THEN	引入 IF 指令的指令部分
TIME	TIME	提供时间信息的基本数据类型
TIME_OF_DAY	TIME_OF_DAY	用于日时钟的基本数据类型
TITLE	TITLE	可选块标题或程序段标题
TO	TO	定义 FOR 指令的整个范围值
TOD	TOD	用于日时钟的基本数据类型
TRUE	TRUE	预定义的布尔型常数：逻辑条件为真，值不等于 0
TYPE	TYPE	引入 UDT
UDT	UDT	全局或 PLC 数据类型
UNLINKED	UNLINKED	标记“与运行无关”
UNTIL	UNTIL	结束 REPEAT 指令的指令部分
UO	AO	(Q1=1) AND (Q0=1) 后的查询
VAR	VAR	引入声明块
VAR_IN_OUT	VAR_IN_OUT	引入声明块
VAR_INPUT	VAR_INPUT	引入声明块
VAR_OUTPUT	VAR_OUTPUT	引入声明块
VAR_TEMP	VAR_TEMP	引入声明块
VERSION	VERSION	块的版本号

关键字 德语助记符	关键字 英语助记符	描述
VOID	VOID	功能调用无返回值
WHILE	WHILE	引入 WHILE 指令
WORD	WORD	用于字的基本数据类型
XOR	XOR	逻辑运算
C	C	计数器
\$_<任意字符>	\$_	对齐符号

参见

在程序中使用变量 (页 336)
PLC 变量和局部变量 (页 337)

7.2.1.7 常量的使用

7.2.1.7 常量的基本知识

常量简介

常量是分配了固定值的变量，使用户可以在程序中用一个名称来代替静态值。常量在程序执行期间可由各种程序元素读取，但不能被覆盖。在程序执行期间更改常量值会导致语法或运行错误。

常量声明

常量在 PLC 变量表的“常量”(Constants) 选项卡中声明。要声明常量，必须输入符号名、数据类型和固定值。常量的输入格式和取值范围取决于常量的数据类型。

参见

数据类型 (页 344)
PLC 变量表的结构 (页 363)

7.2.1.8 数据类型

7.2.1.8 数据类型和参数类型简介

数据类型

用户程序中使用的所有数据都必须通过具体数据类型进行标识。可使用下列数据类型：

- 基本数据类型
- 复杂数据类型，通过链接基本数据类型构成。
- 参数类型，使用该类型可以定义要传送到功能或功能块的参数。
- 由系统提供的系统数据类型，其结构是预定义的并且不可编辑。
- 由 CPU 提供的硬件数据类型。

数据对象的大小

操作将使用特定大小的数据对象。例如，位指令使用位，传送操作使用字节、字和双字。

位是二进制数“0”或“1”。字节由 8 个位组成，字由 16 个位组成，双字由 32 个位组成。

数学运算使用具有字节、字或双字宽度的数据类型。不同格式的数字（例如，整数或浮点数）可使用不同的位位置进行编码。

7.2.1.8 基本数据类型

7.2.1.8 基本数据类型概述

基本数据类型概述

使用数据类型为变量或常量值指定长度和容许范围以及表示类型。

下表显示了基本数据类型的基本属性：

数据类型	长度 (位)	标准格式	取值范围	输入值实例
BOOL (页 345)	1	布尔型	TRUE/FALSE	TRUE
BYTE (页 346)	8	十六进制的数字	16#0 到 16#FF	16#F0
WORD (页 346)	16	十六进制的数字	16#0 到 16#FFFF	16#F0F0
DWORD (页 346)	32	十六进制的数字	16#0000_0000 到 16#FFFF_FFFF	16#F0F0_F0F0
SINT (页 347)	8	有符号整数	-128 到 127	(+)120

数据类型	长度 (位)	标准格式	取值范围	输入值实例
USINT (页 348)	8	无符号整数	0 到 255	50
INT (页 348)	16	有符号整数	-32768 到 32767	(+)1
UINT (页 349)	16	无符号整数	0 到 65535	300
DINT (页 349)	32	有符号整数	-2147483648 到 +2147483647	(+)2131754992
UDINT (页 350)	32	无符号整数	0 到 4294967295	4042322160
REAL (页 350)	32	浮点数	-3.402823e+38 到 -1.175495e-38 ±0 +1.175495e-38 到 +3.402823e+38	1.234567e+13
TIME (页 351)	32	带符号的时间段：	T# -24d20h31m23s648ms 到 T#+24d20h31m23s647ms	T#10d20h30m20s630ms
CHAR (页 351)	8	ASCII 字符	ASCII 字符集	'I'

7.2.1.8 BOOL (位)

说明

BOOL 数据类型的变量可以包含以下位值之一：

- TRUE
- FALSE

下表给出了 BOOL 变量的属性：

长度 (位)	格式	取值范围	输入值实例
1	布尔型	FALSE 或 TRUE	TRUE
	二进制的数字	0 或 1	1
	八进制的数字	8#0 或 8#1	8#1
	十六进制的数字	16#0 或 16#1	16#1

7.2.1.8 BYTE (字节)

描述

BYTE 数据类型的变量是一个 8 位的位序列。

下表给出了 BYTE 变量的属性：

长度 (位)	格式	取值范围	输入值实例
8	无符号整数	0 到 255	15
	二进制的数字	2#0 到 2#11111111	2#00001111
	八进制的数字	8#0 到 8#377	8#17
	十六进制的数字	B#16#0 到 B#16#FF	B#16#F、16#F

7.2.1.8 WORD

描述

WORD 数据类型的变量是一个 16 位的位序列。

下表给出了 WORD 变量的属性：

长度 (位)	格式	取值范围	输入值实例
16	无符号整数	0 到 65535	61680
	二进制的数字	2#0 到 2#1111111111111111	2#1111000011110000
	八进制的数字	8#0 到 8#177777	8#170360
	十六进制的数字	W#16#0 到 W#16#FFFF 16#0 到 16#FFFF	W#16#F0F0、16#F0F0
	2x8 位有符号十进制数	B#(0,0) 到 B#(255,255)	B#(240,240)

7.2.1.8 DWORD

描述

DWORD 数据类型的变量是一个 32 位的位序列。

下表给出了 DWORD 变量的属性：

长度 (位)	格式	取值范围	输入值实例
32	无符号整数	0 到 4294967295	15793935
	二进制的数字	2#0 到 2#11111111111111111111111111111111 1111111111	2#1111000011111111100001111
	八进制的数字	8#0 到 8#3777777777	8#74177417
	十六进制的数字	DW#16#0000_0000 到 DW#16#FFFF_FFFF , 16#0000_0000 到 16#FFFF_FFFF	DW#16#F0FF0F、16#F0FF0F
	4x8 位无符号十进制数	B#(0,0,0,0) 到 B# (255,255,255,255)	B#(240,255,240,255)

7.2.1.8 SINT (8 位整数)

描述

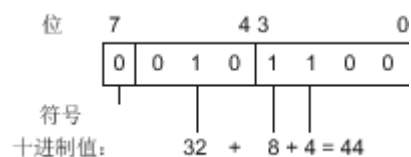
SINT (短 INT) 数据类型的变量的长度为 8 位，可由一个符号和一个数字值两部分组成。位 0 到 6 的信号状态代表数字的值。位 7 的信号状态表示符号。符号可以是“0”(代表正数) 或“1”(代表负数)。

SINT 数据类型的变量在内存中占用一个字节。

下表给出了 SINT 变量的属性：

长度 (位)	格式	取值范围	输入值实例
8	有符号整数	-128 到 127	+50
	二进制的数字	2#0 到 01111111	2#01010000
	八进制的数字	8#0 到 8#177	8#120
	十六进制的数字	16#0 到 16#7F	16#50

下图将整数 +44 显示为二进制数：



7.2.1.8 USINT (8 位整数)

描述

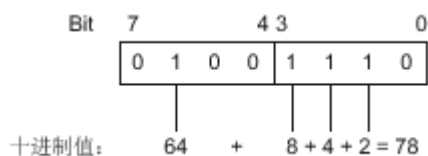
USINT 数据类型 (无符号短 INT) 的变量的长度为 8 位 , 可包含无符号数字值。

USINT 数据类型的变量在内存中占用一个字节。

下表给出了 USINT 变量的属性 :

长度 (位)	格式	取值范围	输入值实例
8	无符号整数	0 到 255	78
	二进制的数字	2#0 到 2#11111111	2#01001110
	八进制的数字	8#0 到 8#377	8#116
	十六进制的数字	16#0 到 16#FF	16#4E

下图将整数 78 显示为二进制数 :



7.2.1.8 INT (16 位整数)

描述

INT 数据类型的变量的长度为 16 位并由两部分组成 : 一个符号和一个数字值。位 0 到 14 的信号状态代表数字的值。位 15 的信号状态表示符号。符号可以是“0”(正数) 或“1”(负数)。

INT 数据类型的变量在内存中占用两个字节。

下表给出了 INT 变量的属性 :

长度 (位)	格式	取值范围	输入值实例
16	有符号整数	-32768 到 32767	+44
	二进制数 (仅正数)	2#0 到 2#0111111111111111	2#0000000000101100
	八进制数 (仅正数)	8#0 到 8#7777	8#54
	十六进制数 (仅正数)	16#0 到 16#7FFF	16#2C

下图将整数 +44 显示为二进制数 :



7.2.1.8 UINT (16 位整数)

描述

UINT 数据类型 (无符号 INT) 的变量的长度为 16 位，可包含无符号数字值。

UINT 数据类型的变量在内存中占用两个字节。

下表给出了 UINT 变量的属性：

长度 (位)	格式	取值范围	输入值实例
16	无符号整数	0 到 65535	65295
	二进制的数字	2#0 到 2#11111111100001111	2#11111111100001111
	八进制的数字	8#0 到 8#177777	8#177417
	十六进制的数字	16#0 到 16#FFFF	16# FF0F

下图将整数 44 显示为二进制数：



7.2.1.8 DINT (32 位整数)

描述

DINT 数据类型的变量的长度为 32 位并由两部分组成：一个符号和一个数字值。位 0 到 30 的信号状态代表数字的值。位 31 的信号状态表示符号。符号可以是“0”(正数) 或“1”(负数)。

DINT 数据类型的变量在内存中占用四个字节。

下表给出了 DINT 变量的属性：

长度 (位)	格式	取值范围	输入值实例
32	有符号整数	-2147483648 到 +2147483647	2131754992
	二进制的数字	2#0 到 2#01111111111111111111111111111111 1111111111	2#01111111110000111111111111110000
	八进制的数字	8#0 到 8#1777777777	8#17703777760
	十六进制的数字	16#0000_0000 到 16#7FFFF_FFFF	16#7F0FFFF0

7.2.1.8 UDINT (32 位整数)

描述

UDINT 数据类型 (无符号双精度 INT) 的变量的长度为 32 位 , 可包含无符号数字值。

UDINT 数据类型的变量在内存中占用四个字节。

下表给出了 UDINT 变量的属性 :

长度 (位)	格式	取值范围	输入值实例
32	无符号整数	0 到 4294967295	4042322160
	二进制的数字	2#0 到 2#11111111111111111111111111111111 1111111111	2#1111000011110000111100001111000011110000
	八进制的数字	8#0 到 8#3777777777	8#36074170360
	十六进制的数字	16#0000_0000 到 16# FFFF_FFFF	16#F0F0F0F0

7.2.1.8 REAL (浮点数)

描述

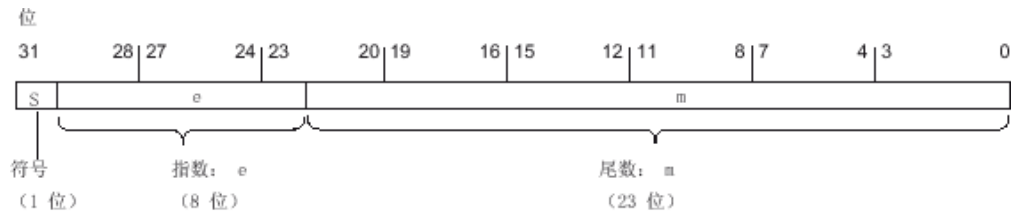
REAL 数据类型的变量的长度为 32 位 , 用于显示浮点数。 REAL 数据类型的变量由以下三部分组成 :

- 符号 : 符号由位 31 的信号状态确定。位 31 可以是“0”(正数) 或“1”(负数)。
- 以 2 为底的 8 位指数 : 该指数加一个常数 (基数 +127), 因此其范围为 255。
- 23 位尾数 : 仅显示尾数的隔断部分。不存储尾数的整数部分 , 因为其在有效值范围内始终等于“1”。

下表给出了 REAL 变量的属性 :

长度 (位)	格式	取值范围	输入值实例
32	符合 IEEE 754 标准的浮点数	-3.402823e+38 到 -1.175495e-38 ±0	1.0e-5
	浮点数	+1.175495e-38 到 +3.402823e+38	1.0

下图所示为 REAL 变量的结构：



7.2.1.8 TIME (IEC 时间)

描述

TIME 数据类型的变量在内存中占用 32 位。持续时间包括天 (d)、小时 (h)、分钟 (m)、秒 (s) 和毫秒 (ms) 信息。

下表给出了 TIME 变量的属性：

长度 (位)	格式	取值范围	输入值实例
32	带符号的时间段：	T# -24d20h31m23s648ms 到 T#+24d20h31m23s647ms	T#10d20h30m20s630ms , TIME#10d20h30m20s630 ms , 10d20h30m20s630ms ;

没必要指定所有时间单位。所以，T#5h10s 是有效时间。如果仅指定了一个单位，则天、小时和分钟的绝对值不能超过上限或下限。当指定了多个时间单位时，各单位值不能超过 23 小时 59 分 59 秒 999 毫秒。

7.2.1.8 CHAR (字符)

描述

CHAR 数据类型的变量的长度为 8 位，在内存中占用一个字节。

CHAR 数据类型存储 ASCII 格式的单个字符。

下表给出了 CHAR 数据类型的取值范围：

长度 (位)	格式	取值范围	输入值实例
8	ASCII 字符	ASCII 字符集	'Q'

7.2.1.8 复杂数据类型

7.2.1.8 复杂数据类型概述

简介

复杂数据类型的定义是由其它数据类型组成的数据组。

不能将任何常量用作复杂数据类型的实参。也不能将任何绝对地址作为实参传送给复杂数据类型。

下表概要介绍了复杂数据类型：

数据类型	描述
DTL (页 352)	DTL 数据类型表示由日期和时间定义的时间点。
STRING (页 353)	STRING 数据类型表示最多包含 254 个字符的字符串。
ARRAY	ARRAY 数据类型表示由固定数目的同一数据类型的元素组成的域。
STRUCT (页 355)	STRUCT 数据类型表示由固定数目的元素组成的结构。不同的结构元素可具有不同的数据类型。

7.2.1.8 DTL

描述

DTL 数据类型的变量的长度为 12 个字节，在预定义结构中保存日期和时间信息。

下表给出了 DTL 变量的属性：

长度 (字节)	格式	取值范围	输入值实例
12	时钟和日历 (年-月变量:小时:分钟:秒.纳秒)	最小值： DTL#1970-01-01-00:00:00.0 最大值： DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

DTL 变量的结构由若干元素构成，各元素可以有不同的数据类型和取值范围。指定值的数据类型必须与相应元素的数据类型相匹配。

下表给出了 DTL 变量的结构元素及其属性：

字节	元素	数据类型	取值范围
0	年	UINT	1970 到 2554
1			
2	月	USINT	0 到 12
3	日	USINT	1 到 31
4	星期	USINT	1 (星期日) 到 7 (星期六) 值输入中不考虑工作日。
5	小时	USINT	0 到 23
6	分钟	USINT	0 到 59
7	秒钟	USINT	0 到 59
8	纳秒	UDINT	0 到 999999999
9			
10			
11			

7.2.1.8 STRING

描述

STRING 数据类型的变量将多个字符保存在一个字符串中，该字符串最多由 254 个字符组成。每个变量的字符串最大长度可由方括号中的关键字 STRING 指定（例如 STRING[4]）。如果省略了最大长度信息，则为相应的变量设置 254 个字符的标准长度。

在内存中，STRING 数据类型的变量比指定最大长度多占用两个字节。

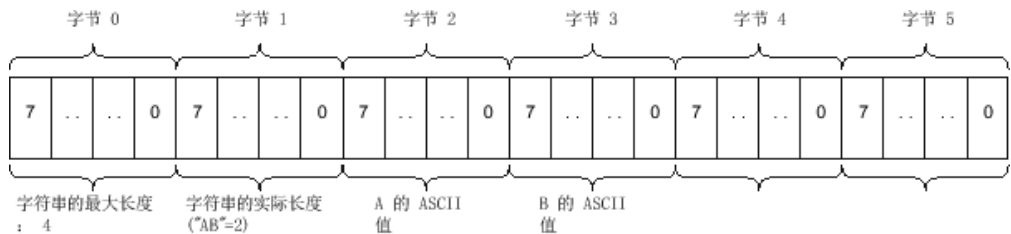
可为 STRING 数据类型的变量分配字符。字符在单引号中指定。如果指定字符串的实际长度小于声明的最大长度，则剩余的字符空间留空。在值处理过程中仅考虑已占用的字符空间。

下表给出了 STRING 变量的属性：

长度 (字节)	格式	取值范围	输入值实例
n + 2	ASCII 字符串	0 到 254 个字符	'Name'

实例

下面的实例显示对输出值 'AB' 指定了 STRING[4] 数据类型时的字节序列：



7.2.1.8 ARRAY

7.2.1.8 ARRAY 的格式

描述

ARRAY 数据类型表示由固定数目的同一数据类型的元素组成的域。所有基本数据类型的元素都可以组合在 ARRAY 变量中。

ARRAY 元素的范围信息显示在关键字 ARRAY 后面的方括号中。范围的下限值必须小于或等于上限值。ARRAY 可包含一维。

下表给出了 ARRAY 变量的属性：

长度	格式	取值范围
元素的数目 * 数据类型的长度	<数据类型> 的 ARRAY [下限值... 上限值]	[- 32 768.. <数据类型> 的 [-32 768.. +32 767]

实例

下面的实例说明可如何声明 ARRAY 数据类型的变量：

名称	数据类型	注释
测量值	REAL 类型的 ARRAY[1..20]	具有 20 个元素的一维 ARRAY 变量
日时钟	INT 类型的 ARRAY[-5..5]	具有 11 个元素的一维 ARRAY 变量

参见

声明 ARRAY 数据类型的变量 (页 403)
一维 ARRAY 的实例 (页 355)

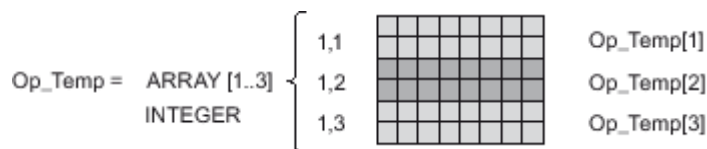
7.2.1.8 一维 ARRAY 的实例

一维 ARRAY 变量的声明

下表说明一维 ARRAY 变量的声明：

名称	数据类型	注释
Op_Temp	INT 类型的 ARRAY[1..3]	具有 3 个元素的一维 ARRAY 变量。

下图所示为已声明的 ARRAY 变量的结构：



访问 ARRAY 元素

通过下标访问各数组元素。第一个 ARRAY 元素的下标为 [1]，第二个元素的下标为 [2]，第三个元素的下标为 [3]。在本例中要访问第二个 ARRAY 元素的值，需要在程序中指定“Op_Temp [2]”。

变量“Op_Temp”也可声明为 INT 的 ARRAY，则第一个 ARRAY 元素的下标为 [-1]，第二个元素的下标为 [0]，第三个元素的下标为 [1]

参见

ARRAY 的格式 (页 354)
声明 ARRAY 数据类型的变量 (页 403)

7.2.1.8 STRUCT

描述

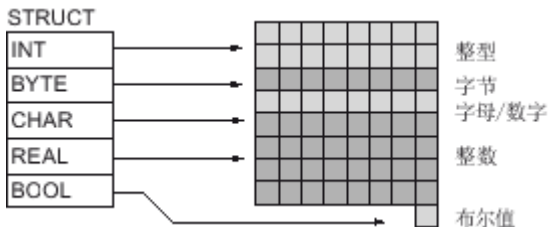
STRUCT 数据类型的变量将值保存在一个由固定数目的元素组成的结构中。不同的结构元素可具有不同的数据类型。不能在 STRUCT 变量中嵌套结构。

STRUCT 变量始终以具有偶地址的一个字节开始，并占用直到下一个字限制的内存。

下表给出了 STRUCT 变量的属性：

长度	格式	取值范围	输入值实例
STRUCT 变量以具有偶地址的一个字节开始，并占用直到下一个字限制的内存。	STRUCT	应用所用数据类型的值范围。	应用所用数据类型的值输入规则。

下图所示为一个 STRUCT 变量结构的实例：



参见

声明 STRUCT 数据类型的变量 (页 404)

7.2.1.8 参数类型

7.2.1.8 参数类型概述

简介

除了基本数据类型和复杂数据类型，还可以为块之间传送的形参定义参数类型。

下表概要说明了这些参数类型：

参数	大小 (字节)	描述
<u>VARIANT</u> (页 357)	0	VARIANT 类型的参数是一个可以指向各种数据类型或参数类型变量的指针。VARIANT 参数类型可识别结构并指向这些结构。
<u>VOID</u>	-	不保存任何值。如果某个功能不需要任何返回值，则使用 VOID 数据类型。

7.2.1.8 VARIANT

描述

VARIANT 类型的参数是一个可以指向各种数据类型或参数类型变量的指针。VARIANT 参数类型可识别结构并指向这些结构。使用参数类型 VARIANT 还可以指向 STRUCT 变量的各元素。VARIANT 参数类型变量在内存中不占用任何空间。

下表给出了 VARIANT 参数类型的属性：

表示法	格式	长度 (字节)	输入值实例
符号	操作数	0	MyTag
	数据块名称.操作数名称.元素		MyDB.StructTag.FirstComponent
绝对	操作数		%MW10
	数据块编号.操作数 类型长度		P#DB10.DBX10.0 INT 12

7.2.1.8 VOID

描述

VOID 数据类型不保存任何值。如果某个功能不需要任何返回值，则使用此数据类型。

7.2.1.8 系统数据类型

7.2.1.8 系统数据类型

描述

系统数据类型 (SDT) 由系统提供并具有预定义的结构。系统数据类型的结构由固定数目的可具有各种数据类型的元素构成。不能更改系统数据类型的结构。

系统数据类型只能用于特定指令。下表给出了可用的系统数据类型及其用途：

系统数据类型	以字节为单位的结构长度	描述
IEC_TIMER	16	时钟的结构 例如，此数据类型用于“TP”、“TOF”、“TON”和“TONR”指令。

系统数据类型	以字节为单位的结构长度	描述
IEC_SCOUNTER	3	计数器的结构，其计数为 SINT 数据类型 例如，此数据类型用于“CTU”、“CTD”和“CTUD”指令。
IEC_USCOUNTER	3	计数器的结构，其计数为 USINT 数据类型 例如，此数据类型用于“CTU”、“CTD”和“CTUD”指令。
IEC_COUNTER	6	计数器的结构，其计数为 INT 数据类型 例如，此数据类型用于“CTU”、“CTD”和“CTUD”指令。
IEC_UCOUNTER	6	计数器的结构，其计数为 UINT 数据类型 例如，此数据类型用于“CTU”、“CTD”和“CTUD”指令。
IEC_DCOUNTER	12	计数器的结构，其计数为 DINT 数据类型 例如，此数据类型用于“CTU”、“CTD”和“CTUD”指令。
IEC_UDCOUNTER	12	计数器的结构，其计数为 UDINT 数据类型 例如，此数据类型用于“CTU”、“CTD”和“CTUD”指令。
ERROR_STRUCT	28	编程或 I/O 访问错误的错误信息结构 例如，此数据类型用于“GET_ERROR”指令。
CONDITIONS	52	定义的数据结构，定义了数据接收开始和结束的条件。 例如，此数据类型用于“RCV_GFG”指令。
TCON_Param	64	指定数据块结构，用于存储通过工业以太网 (PROFINET) 进行的开放式通信的连接说明。
VOID	-	VOID 数据类型不保存任何值。如果输出不需要任何返回值，则使用此数据类型。例如，如果不需要错误信息，则可以在输出 STATUS 上指定 VOID 数据类型。

7.2.1.8 硬件数据类型

7.2.1.8 硬件数据类型

描述

硬件数据类型由 CPU 提供。可用硬件数据类型的数目取决于 CPU。

根据硬件配置中设置的模块存储特定硬件数据类型的常量。在用户程序中插入用于控制或激活已组态模块的指令时，可将这些可用常量用作参数。

下表给出了可用的硬件数据类型及其用途：

数据类型	基本数据类型	描述
HW_ANY	WORD	任何硬件组件（如模块）的标识
HW_IO	HW_ANY	I/O 组件的标识
HW_SUBMODULE	HW_IO	中央硬件组件的标识
HW_INTERFACE	HW_SUBMODULE	接口组件的标识
HW_HSC	HW_SUBMODULE	快速计数器的标识 例如，此数据类型用于“CTRL_HSC”指令。
HW_PWM	HW_SUBMODULE	脉冲宽度调制的标识 例如，此数据类型用于“CTRL_PWM”指令。
HW_PTO	HW_SUBMODULE	脉冲传感器的标识 此数据类型用于运动控制
AOM_IDENT	DWORD	AS 运行系统中对象的标识。
EVENT_ANY	AOM_IDENT	用于标识任意事件
EVENT_ATT	EVENT_ANY	用于标识可动态分配给 OB 的事件。 例如，此数据类型用于“ATTACH”和“DETACH”指令。
EVENT_HWINT	EVENT_ATT	用于标识硬件中断事件。
OB_ANY	INT	用于标识任意 OB。
OB_DELAY	OB_ANY	用于标识发生延时中断时调用的 OB。例如，此数据类型用于“SRT_DINT”和“CAN_DINT”指令。
OB_CYCLIC	OB_ANY	用于标识发生循环中断时调用的 OB。

数据类型	基本数据类型	描述
OB_ATT	OB_ANY	用于标识可动态分配给事件的 OB。 例如，此数据类型用于“ATTACH”和“DETACH”指令。
OB_PCYCLE	OB_ANY	用于标识可分配给“循环程序”事件类别事件的 OB。
OB_HWINT	OB_ATT	用于标识发生硬件中断时调用的 OB。
OB_DIAG	OB_ANY	用于标识发生诊断错误中断时调用的 OB。
OB_TIMEERROR	OB_ANY	用于标识发生时间错误时调用的 OB。
OB_STARTUP	OB_ANY	用于标识发生启动事件时调用的 OB。
PORT	UINT	用于标识通信端口 此数据类型用于点对点通信。
CONN_ANY	WORD	用于标识任意连接
CONN_OUC	CONN_ANY	用于标识通过工业以太网 (PROFINET) 进行开放式通信的连接

7.2.1.8 数据类型转换

7.2.1.8 数据类型转换概述

简介

如果在一个指令中链接多个操作数，必须确保这些数据类型是兼容的。这一点也适用于分配或提供块参数。如果操作数不是同一数据类型，则必须执行转换。

可选择两种转换方式：

- 隐式转换
执行指令时自动进行转换。
- 显式转换
在执行实际指令之前使用显式转换指令。

隐式转换

如果操作数的数据类型是兼容的，则自动执行隐式转换。可根据较严格或较宽松的条件执行此兼容性测试：

- 有 IEC 检查 (默认设置):
应用严格的兼容性规则: 指令中链接的操作数必须属于同一种数据类型。
- 无 IEC 检查
根据较宽松的规则执行此兼容性测试。指令中链接的操作数不必属于同一种数据类型。但是, 这些数据类型必须具有相同的数据宽度。
例如, 如果禁用 IEC 检查, 可以将 INT 数据类型的操作数和 WORD 数据类型的操作数相链接。

注意

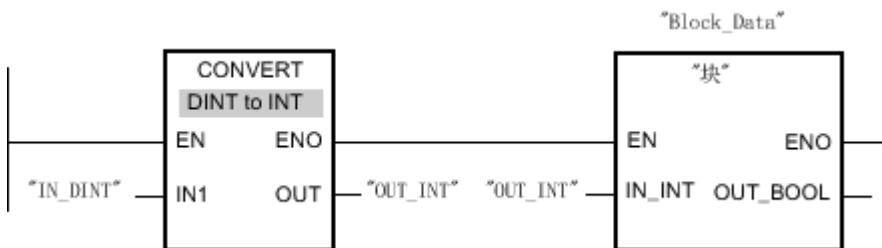
从 REAL 到 TIME 或从 TIME 到 REAL 的转换例外。不能隐式进行此转换。

显式转换

如果因操作数不兼容而不能进行隐式转换, 则可以使用显式转换指令。可以在“指令”(Instructions) 任务卡的“数学”、“字符串 + 字符”和“转换”部分中找到转换指令。

显式转换的优点是可通过输出 ENO 检测到所有超出范围的情况。

下图所示为一个必须执行显式数据类型转换的实例:



“Block”功能块的输入参数“IN_INT”需要 INT 数据类型的变量。由此变量“IN_DINT”的值必须从 DINT 变为 INT, 方可以 INT 数据类型传输到“模块”。

在从 DINT 到 INT 的转换期间, 会使用 DINT 的低值位并将其解释为 INT。但是当前出现的符号可能会在进程中丢失。

参见

设置 IEC 测试 (页 361)

7.2.1.8 设置 IEC 测试

执行指令时, 会检查所用操作数的数据类型是否兼容。可根据较严格或较宽松的条件执行此兼容性测试。如果激活“IEC 检查”, 则应用较严格条件。

可以为项目的所有新块或为特定块集中设置 IEC 检查。

为新块设置 IEC 检查

要为项目中的所有新块设置 IEC 检查, 请按以下步骤操作:

1. 在“选项”(Options) 菜单中, 选择“设置”(Settings) 命令。

工作区中将显示“设置”(Settings) 窗口。

2. 在区域导航中选择“PLC 编程 > 常规”(PLC programming > General) 组。
3. 在“新块的默认设置”(Default settings for new blocks) 组中选中或清除“IEC 检查”(IEC Check) 复选框。
将为程序中的所有新块启用或禁用 IEC 检查。

为某个块设置 IEC 检查

要为某个块设置 IEC 检查，请按以下步骤操作：

1. 打开块。
2. 在巡视窗口中打开“属性”(Properties) 选项卡。
3. 在区域导航中单击“属性”(Attributes) 组。
4. 选中或清除“IEC 检查”(IEC Check) 复选框。

将为此块启用或禁用 IEC 检查。但是，直到保存项目时才会保存此设置。

参见

[数据类型转换概述 \(页 360\)](#)

7.2.1.9 块编程的步骤

7.2.1.9 基本步骤

概述

在创建了自动化解决方案的概念后，下一步是执行此概念。建议按下列步骤操作：

声明 PLC 变量
↓
创建块 (页 374)
↓
声明局部变量 (页 398)
↓
创建程序代码 (页 409)
↓
保存 (页 134)
↓
编译 (页 487)
↓

装载 (页 488)
↓
测试 (页 523)

7.2.2 声明 PLC 变量

7.2.2.1 PLC 变量表的结构

定义


PLC 变量表包含在整个 CPU 范围有效的变量和常量的定义。系统会为项目中使用的每个 CPU 自动创建一个 PLC 变量表。PLC 变量表包含一个变量选项卡和一个常量选项卡。

“PLC 变量”选项卡的结构

在“PLC 变量”(PLC tags) 选项卡中声明程序中所需的全局变量。下图所示为该选项卡结构：

PLC tags						
	名称	数据类型	地址	保持性	监视值	注释
1	 Tag_1	Bool	%I1.2	<input type="checkbox"/>	<input type="checkbox"/> FALSE	
2	 Tag_2	Bool	%I0.1	<input type="checkbox"/>	<input type="checkbox"/> FALSE	
3	 Tag_3	Bool	%I0.0	<input type="checkbox"/>	<input type="checkbox"/> FALSE	

下表说明了各列的含义：


列	描述
	可以单击该符号，以便通过拖放操作将变量移动到程序段中以用作操作数。
名称	为变量指定的在整个 CPU 范围唯一的名称。
数据类型	为变量指定的数据类型
地址	变量地址。
保持性	将变量标记为具有保持性。 保持性变量的值将保留，即使在电源关闭后也是如此。
监视值	CPU 中的当前数据值。 只有当在线连接可用并选择“监视”(Monitor) 按钮时，此列才会出现。
注释	用于描述变量的注释。

“常量”选项卡的结构

系统需要的常量显示在“常量”(Constants) 选项卡中。下图所示为该选项卡结构：

PLC tags				
	名称 ▲	数据类型	值	注释
1	AI2[AI]	Hw_SubModule	7	
2	DI14/DO10[DI/DO]	Hw_SubModule	8	
3	HSC_1[HSC]	Hw_Hsc	1	
4	HSC_2[HSC]	Hw_Hsc	2	
5	HSC_3[HSC]	Hw_Hsc	3	

下表说明了各列的含义：

列	描述
	常量的符号
名称	常量在 CPU 范围内的唯一名称。
数据类型	常量的数据类型
值	常量的值
注释	用于描述变量的注释。

参见

- 数据类型 (页 344)*
- 硬件数据类型 (页 359)*
- 打开 PLC 变量表 (页 364)*
- 在 PLC 变量表中声明变量 (页 366)*
- PLC 变量和局部变量 (页 337)*
- 常量的基本知识 (页 343)*
- 符号编程 (页 333)*
- 表格中的键盘快捷键 (页 121)*

7.2.2.2 打开 PLC 变量表

步骤

要在 CPU 中打开 PLC 变量表，请按以下步骤操作：

1. 在项目树中打开 CPU 下面的“PLC 变量”(PLC tags) 文件夹。
2. 双击该文件夹中的 PLC 变量表。
3. 在右上角选择“PLC 变量”(PLC tags) 选项卡或“常量”(Constants) 选项卡。

结果

与 CPU 关联的 PLC 变量表打开。可以声明所需的变量和常量。

7.2.2.3 声明 PLC 变量

7.2.2.3 PLC 变量的规则

7.2.2.3 可用的 PLC 变量地址和数据类型

使用的助记符

在 PLC 变量表中输入的地址会自动调整，以适合设置助记符。

可用的变量地址和数据类型

下表给出了可用的地址和数据类型：

英语助记符	德语助记符	说明：	数据类型：	地址区域：
I	E	输入位	BOOL	0.0..1023.7
IB	EB	输入字节	BYTE、CHAR、INT、USINT	0..1023
IW	EW	输入字	WORD、INT、UINT	0..1022
ID	ED	输入双字	DWORD、DINT、UDINT、REAL、TIME	0..1020
Q	A	输出位	BOOL	0.0..1023.7
QB	AB	输出字节	BYTE、CHAR、INT、USINT	0..1023
QW	AW	输出字	WORD、INT、UINT	0..1022
QD	AD	输出双字	DWORD、DINT、UDINT、REAL、TIME	0..1020
M	M	存储器位	BOOL	0.0..8191.7
MB	MB	存储器字节	BYTE、CHAR、INT、USINT	0..8191
MW	MW	存储器字	WORD、INT、UINT	0..8190
MD	MD	存储器双字	DWORD、DINT、UDINT、REAL、TIME	0..8188

参见

设置助记符 (页 409)

7.2.2.3 PLC 变量表中不完整和不明确的条目

不完整的定义

您甚至可以保存未完成定义的变量或常量。例如，可以首先仅指定名称，以后再添加数据类型信息。因此您可以随时中断在 PLC 变量表上的工作，并保存中间信息。

但是，只有这些变量或常量的定义完整时才能编译包含这些变量或常量的块。

不明确的定义

PLC 变量的名称在整个 CPU 范围内必须唯一。不能对新的 PLC 变量使用已经分配给 CPU 中某个对象（例如，一个块或另一个 PLC 变量）的名称。如果输入了一个已经在其它时间分配的名称，则会在输入的第二个名称上自动添加序号。例如，如果第二次输入名称“Motor”，则第二个条目将更改为“Motor_1”。

地址在整个 CPU 范围内也必须唯一。如果输入一个已经分配给其它变量的地址，则该地址将在两个位置都以黄色高亮显示，并发出错误消息。

7.2.2.3 声明 PLC 变量

7.2.2.3 在 PLC 变量表中声明变量

要求

PLC 变量表的“PLC 变量”(PLC tags) 选项卡已打开。

步骤

要定义 PLC 变量，请按以下步骤操作：

1. 在“名称”(Name) 列中输入变量名称。
2. 单击“数据类型”(Data type) 列中的箭头键，然后选择所需的数据类型。
将自动添加对应该数据类型的地址。
3. 如有必要，在“地址”(Address) 列中修改变量的地址。
4. 可根据需要在“注释”(Comment) 列中输入注释。
5. 对所需的所有变量重复步骤 1 到 4。

另请参见：[可用的 PLC 变量地址和数据类型 \(页 365\)](#)

语法检查

每次输入后都会执行语法检查，并且找到的任何错误都将以红色显示。无需立即更正这些错误，您可以继续编辑并在以后进行所有更正。但是，如果变量声明包含语法错误，将无法编译程序。

参见

PLC 变量表中不完整和不明确的条目 (页 366)
PLC 变量表的结构 (页 363)
编辑 PLC 变量表 (页 373)
插入表格行 (页 373)
PLC 变量表中的行排序 (页 373)
表格中的键盘快捷键 (页 121)

7.2.2.3 在程序编辑器中定义 PLC 变量

要求

- 程序编辑器已打开。
- 程序中使用一个或多个操作数。

步骤

要将操作数声明为全局 PLC 变量，请按以下步骤操作：

1. 选择一个或多个操作数。
2. 在快捷菜单中选择“定义变量”(Define tag) 命令。
将打开“定义变量”(Define tag) 对话框。此对话框显示一个声明表，其中已输入相应操作数的名称。
3. 在“部分”(Section) 列中单击箭头键，并选择以下条目之一：
 - 全局存储器
 - 全局输入
 - 全局输出
4. 在其它列中，输入地址、数据类型和注释。
另请参见：[可用的 PLC 变量地址和数据类型 \(页 365\)](#)
5. 单击“定义”(Define) 按钮完成输入。

结果

变量声明写入 PLC 变量表，且可适用于 CPU 中的所有块。

参见

PLC 变量表中不完整和不明确的条目 (页 366)
表格中的键盘快捷键 (页 121)
自动填充 PLC 变量表中的单元格 (页 373)

7.2.2.3 更改 PLC 变量的地址

可以在 PLC 变量表中或直接在程序编辑器中更改 PLC 变量的地址。更改将立即在整个程序内生效。

步骤

要更改 PLC 变量的地址，请按以下步骤操作：

1. 在项目树中双击 PLC 变量表。
将打开 PLC 变量表。
2. 打开“变量”(Tag) 选项卡。
3. 更改“地址”(Address) 列中的条目。

或

1. 在程序中的变量使用点选择一个或多个变量。
2. 在快捷菜单中选择“重新连接变量”(Rewire tag) 命令。
将打开“重新连接变量”(Rewire tag) 对话框。
3. 更改“地址”(Address) 列中的条目。
4. 单击“更改”(Change) 按钮完成输入。

结果

自动在程序中的所有使用点更改了变量地址。

参见

可用的 PLC 变量地址和数据类型 (页 365)

7.2.2.3 更改 PLC 变量的名称

可以在 PLC 变量表中或直接在程序编辑器中更改 PLC 变量的名称。更改将立即在整个程序内生效。

步骤

要更改 PLC 变量的名称，请按下列步骤操作：

1. 在项目树中双击 PLC 变量表。
将打开 PLC 变量表。
2. 打开“变量”(Tag) 选项卡。
3. 更改“名称”(Name) 列中的条目。

或

1. 在程序中的变量使用点选择一个或多个变量。
2. 在快捷菜单中选择“重命名变量”(Rename tag) 命令。
将打开“重命名变量”(Rename tag) 对话框。
3. 更改“名称”(Name) 列中的条目。
4. 单击“更改”(Change) 按钮完成输入。

结果

自动在程序中的所有使用点更改了变量名称。

参见

可用的 PLC 变量地址和数据类型 (页 365)

7.2.2.3 设置 PLC 变量的保持性

7.2.2.3 PLC 变量的保持特性

位存储器的保持性存储区

为防止在发生电源故障时数据丢失，可以为位存储器定义保持性存储区。保持性存储区是指在电源关闭后以及从“STOP”切换到“RUN”上电后，其内容会被保留的区域。可以在 PLC 变量表中指定保持性存储区的具体宽度。

保持性存储区中的数值保存在 CPU 的备份存储器中。这些值在热启动时仍将保留。

参见

设置 PLC 变量的保持特性 (页 369)

7.2.2.3 设置 PLC 变量的保持特性

简介

在 PLC 变量表中，可以为位存储器指定保持性存储区的宽度。编址在该存储区中的所有变量随即被标识为有保持性。通过 PLC 变量表的“保持性”(Retain) 列中设置的复选标记可以识别变量的保持性设置。

要求

PLC 变量表的“PLC 变量”(PLC tags) 选项卡已打开。

步骤

要为位存储器定义保持性存储区的宽度，请按以下步骤操作：

1. 在工具栏上，单击“保持性”(Retain) 按钮。
将打开“保持性存储器”(Retain memory) 对话框。
2. 通过在文本框中输入区域的最后一个字节（从 0 开始计数）来指定保持性存储区的可用空间。指定宽度时，请注意不要声明为位存储器的现有变量的地址。
3. 单击“确定”(OK) 按钮。

结果

定义了保持性存储器的宽度。在变量表的“保持性”(Retain) 列中，自动为位于保持性存储区内的所有变量设置复选标记。

参见

PLC 变量的保持特性 (页 369)

7.2.2.3 编辑 PLC 变量的属性**7.2.2.3 PLC 变量的属性****概述**

下表概述了 PLC 变量的属性：

组	属性	描述
常规	变量名称	表内唯一的名称。
	地址	变量地址。
	注释	变量注释。
	数据类型	变量的数据类型。
时间戳	创建日期	创建变量的时间 (无法更改)。
	更改日期	上次修改变量的时间 (无法更改)。
用法	HMI 中可见	HMI 中操作数选择的默认设置。
	对于 HMI 只读	在 HMI 中写保护
	用户定义的排序标准	对外部表格中变量排序的标识符。
值	初始值	数据块中未存储当前值时的变量值。 此值必须与声明的数据类型兼容。
	下限	变量不可小于此值。
	上限	变量不可超出此值。

参见

编辑 PLC 变量的属性 (页 371)

7.2.2.3 编辑 PLC 变量的属性

在 PLC 变量表中编辑常规属性

要编辑一个或多个变量的常规属性，请按以下步骤操作：

1. 在项目树中双击 CPU 的 PLC 变量表。
将打开 PLC 变量表。
2. 更改“名称”(Name)、“地址”(Address) 或“注释”(Comment) 列中的条目。

在属性窗口中编辑具体属性

要编辑各变量的具体属性，请按以下步骤操作：

1. 在程序段中选择一个变量。
2. 在快捷菜单中选择“属性”(Properties) 命令。
将打开属性对话框。此对话框显示变量的具体属性。
3. 更改该属性窗口中的条目。

参见

PLC 变量的属性 (页 370)

7.2.2.4 PLC 变量的监视

7.2.2.4 PLC 变量的监视

可以在 PLC 变量表中直接监视 CPU 中变量的当前数据值。

要求

在线连接可用。

步骤

要监视数据值，请按以下步骤操作：

1. 单击“全部监视”(Monitor all) 按钮启动监视。
此表中将显示附加的“监视值”(Monitor value) 列。该列显示当前数据值。
2. 再次单击“全部监视”(Monitor all) 按钮结束监视。

参见

PLC 变量表的结构 (页 363)

7.2.2.5 编辑 PLC 变量表

7.2.2.5 在 PLC 变量表中复制条目

可以在表格中复制 PLC 变量。

步骤

要复制变量，请按以下步骤操作：

1. 选择要复制的变量。

也可以在按住 <Ctrl> 键时依次单击变量或按住 <Shift> 键单击第一个和最后一个变量来选择多个变量。

2. 在快捷菜单中选择“复制”(Copy)。
3. 将插入指针放置到想要插入这些变量的位置。
4. 在快捷菜单中选择“粘贴”(Paste)。

或

1. 选择变量。
2. 按住鼠标左键。
3. 同时按 <Ctrl> 键。
4. 将该变量拖动到目标位置。

结果

- 该变量即被复制到目标位置。
- 如果存在名称冲突，则将在该变量名后自动附加一个编号。例如，“Tag”可能会变为“Tag_1”。
- 此变量的所有其它属性保持不变。

参见

表格中的键盘快捷键 (页 121)

调整表格 (页 119)

7.2.2.5 在 PLC 变量表中删除条目

步骤

要删除变量，请按以下步骤操作：

1. 选择含有要删除的变量的行。也可以在按住 <Ctrl> 键时依次单击行或按住 <Shift> 键单击第一行和最后一行来选择多行。
2. 在快捷菜单中选择“删除”(Delete) 命令。

参见

调整表格 (页 119)

表格中的键盘快捷键 (页 121)

7.2.2.5 自动填充 PLC 变量表中的单元格

可以将一个或多个表格单元格的内容装载到下面的单元格，从而自动填充最多 100 个连续单元格。

如果自动填充“名称”(Name) 列中的单元格，将在每个名称后面附加一个连续的编号。例如，“Motor”将变为“Motor_1”。

如果自动填充“地址”(address) 列中的单元格，将根据所指定的数据类型增加地址。

步骤

要自动填充连续单元格，请按以下步骤操作：

1. 选择要装载的单元格。
2. 单击该单元格右下角的“填充”(Fill) 符号。
鼠标指针将变为十字形。
3. 按住鼠标按键并向下拖动鼠标指针扫过要自动填充的单元格。
4. 释放鼠标按键。

参见

调整表格 (页 119)
表格中的键盘快捷键 (页 121)

7.2.2.5 插入表格行

步骤

要在鼠标指针位置上方插入行，请按以下步骤操作：

1. 将鼠标指针放置在要在其上方插入一个新行的行中。
2. 在表格的工具栏上，单击“插入行”(Insert row) 按钮。
将在所选行上方插入一个新行。

参见

表格中的键盘快捷键 (页 121)
调整表格 (页 119)

7.2.2.5 PLC 变量表中的行排序

可以依据名称、数据类型或地址按字母数字顺序对表中的行排序。

步骤

要对表行排序，请按以下步骤操作：

1. 选择要排序的列。
2. 单击列标题。
 该列将按值增大的顺序排序。
 上箭头指示排序顺序。
3. 要更改排序顺序，请单击该箭头。
 该列将按值减小的顺序排序。
 下箭头指示排序顺序。
4. 要恢复原始顺序，请第三次单击列标题。

参见

表格中的键盘快捷键 (页 121)
调整表格 (页 119)

7.2.3 创建和管理块

7.2.3.1 创建块

7.2.3.1 块文件夹

功能

可以在项目树中找到“程序块”(Program blocks) 文件夹，在其中可以创建和管理以下块：

- [组织块 \(OB\) \(页 374\)](#)
- [功能块 \(FB\) \(页 375\)](#)
- [功能 \(FC\) \(页 375\)](#)
- [数据块 \(DB\) \(页 376\)](#)

系统会自动为每个设备生成一个程序循环 OB 并将其插入到“程序块”(Program blocks) 文件夹中。

7.2.3.1 创建组织块

要求

“程序块”(Program blocks) 文件夹在项目树中已打开。

步骤

要创建组织块，请按以下步骤操作：

1. 双击“添加新块”(Add new block) 命令。
将打开“添加新块”(Add new block) 对话框。
2. 单击“组织块 (OB)”(Organization block (OB)) 按钮。
3. 选择组织块。
4. 如果想要分配组织块的编号，则选择“手动”(Manual) 选项按钮。
5. 在“语言”(Language) 下拉列表中，为新组织块选择编程语言。
6. 如果插入循环中断 OB，则可在“扫描时间”(Scan time) 文本框中输入循环时间。
7. 要为新组织块添加其它属性，请单击该对话框下半部分中的“更多信息”(Further information) 旁的箭头。
将显示一个具有更多输入域的区域。
8. 输入所需的所有属性。
9. 单击“确定”(OK) 确认输入。

结果

创建了新组织块。可以在项目树的“程序块”(Program blocks) 文件夹中找到该组织块。

注意

您可选中该对话框底部的“添加新对象并打开”(Add new and open) 复选框。这样，创建组织块后会立即将其打开。

参见

组织块 (OB) (页 312)
给硬件中断 OB 分配参数 (页 384)
给循环中断 OB 分配参数 (页 385)

7.2.3.1 创建功能和功能块

要求

“程序块”(Program blocks) 文件夹在项目树中已打开。

步骤

要创建功能 (FC) 或功能块 (FB)，请按以下步骤操作：

1. 双击“添加新块”(Add new block) 命令。
将打开“添加新块”(Add new block) 对话框。
2. 单击“功能块 (FB)”(function block (FB)) 或“功能 (FC)”(function (FC)) 按钮。

3. 输入块名称。
4. 在“语言”(Language) 下拉列表中，为新块选择编程语言。
5. 如果想要分配块编号，则选择“手动”(Manual) 选项按钮。
6. 如果选择使用手动分配，请在输入域中输入块编号。
7. 要为新块添加其它属性，请单击该对话框下半部分中的“更多信息”(Further information) 旁的箭头。
将显示一个具有更多输入域的区域。
8. 输入所需的所有属性。
9. 单击“确定”(OK) 确认输入。

结果

创建了新块。可以在项目树的“程序块”(Program blocks) 文件夹中找到该块。

注意

您可选中该对话框底部的“添加新对象并打开”(Add new and open) 复选框。这样，创建块后会立即将其打开。

参见

功能块 (FB) (页 320)

功能 (FC) (页 320)

7.2.3.1 创建数据块

先决条件

“程序块”(Program blocks) 文件夹在项目树中已打开。

步骤

要创建数据块，请按以下步骤操作：

1. 双击“添加新块”(Add new block) 命令。
将打开“添加新块”(Add new block) 对话框。
2. 单击“数据块 (DB)”(Data block (DB)) 按钮。
3. 输入数据块名称。
4. 选择数据块类型。用户有以下选择：
 - 要创建全局数据块，请选择列表条目“全局 DB”(Global DB)。
 - 要创建背景数据块，请从列表中选择要为其分配背景数据块的目标功能块。
5. 如果想要分配块编号，则选择“手动”(Manual) 选项按钮。
6. 如果选择使用手动分配，请在输入域中输入块编号。

7. 要为新数据块添加其它属性，请单击该对话框下半部分中的“更多信息”(Further information) 旁的箭头。
将显示一个具有更多输入域的区域。
8. 输入所需的所有属性。
9. 单击“确定”(OK) 确认输入。

结果

创建了新数据块。可以在项目树的“程序块”(Program blocks) 文件夹中找到该数据块。

注意

您可选中该对话框底部的“添加新对象并打开”(Add new and open) 复选框。这样，创建块后会立即将其打开。

参见

背景数据块 (页 321)
数据块编程 (页 472)
全局数据块 (DB) (页 321)

7.2.3.1 使用库中的块

可以将块保存在项目库或全局库中，以便可在用户程序中多次使用它们。

要求

- 已显示“库”(Libraries) 任务卡。
- 未对全局库设置写保护。

将块添加到项目库或全局库

要将块添加到项目库或全局库，请按以下步骤操作：

1. 选择要添加到库的块。
2. 通过拖放操作将选定的块移动到“库”(Libraries) 任务卡的库窗格中库的书形符号上。在鼠标指针下方出现小加号之后，释放鼠标左键。

使用项目库或全局库中的块

要在项目中使用项目库或全局库中的块，请按以下步骤操作：

1. 最大化项目库或全局库，以便可以看到该库中的元素。
2. 通过拖放操作将块移动到 CPU 的块文件夹。如果所选的插入点不允许该操作，则鼠标指针将显示为带有斜线的圆。

参见

使用库 (页 164)

打开全局库 (页 169)

7.2.3.1 复制和粘贴块

7.2.3.1 复制和插入块的原则

功能

还可以通过复制现有的块和粘贴副本来创建新块。使用该方法时，请遵守以下原则：

- 可以无限制地复制组织块 (OB)、功能 (FC)、功能块 (FB) 和全局数据块 (DB)。
- 因为背景数据块分配给功能块之后便无法更改，因此只能复制属于具体功能块的背景数据块。但如果将背景数据块复制到其他 CPU，则将取消这种分配。如果那里存在编号相同的功能块，则会将背景数据块分配给该功能块。如果将背景数据块和功能块一起复制到其他 CPU，则背景数据块将分配给该功能块的副本。

复制数据

通过粘贴操作可将所有块数据复制并转发到副本。这些数据包括：

- 块接口变量
- 所有程序段
- 所有现有编译中的注释
- 块中定义的中断
- 被复制块的全部程序代码，包括块中所含的调用指令。

但是，不会复制被调用块及其相关的背景数据块。

避免粘贴期间发生名称冲突

将复制的同名块粘贴到已有块中时，将采用以下机制来避免名称冲突：

- 将复制的块粘贴到同一个 CPU 中：
块的副本将获得带编号的名称。例如，如果复制了块“A”，则副本的名称可能是“A_1”。不使用连续编号，而是使用最小的空闲编号。如果没有更小的编号可用，则块“A”的副本还可能获得名称“A_25”。
- 将复制的块粘贴到其他 CPU 中：
将打开一个对话框，用户可以选择是替换同名的块还是以副本名称（名称_编号）粘贴复制的块。

注意

如果所粘贴的块与现有块的块编号相同，则可能发生编号冲突。该块编号在粘贴期间不会自动更改。这种双重编号可能会影响块调用。因此，在复制块时，应仔细检查块编号并手动或使用块属性纠正重复的块编号。

参见

复制块 (页 379)
插入块 (页 379)

7.2.3.1 复制块

要求

“程序块”(Program blocks) 文件夹在项目树中已打开。

步骤

要复制块，请按下列步骤操作：

1. 右键单击要复制的块。
2. 在快捷菜单中选择“复制”(Copy)。

结果

该块的副本此时在剪贴板中，并且可以粘贴到同一个 CPU 或其它 CPU 中。

参见

复制和插入块的原则 (页 378)

7.2.3.1 插入块

要求

已复制块。

步骤

要将所复制的块及其数据粘贴到 CPU 中，请按以下步骤操作：

1. 在项目树中，打开要粘贴复制块的目标 CPU 的文件夹结构。
2. 右键单击“程序块”(Program blocks) 文件夹。
3. 在快捷菜单中选择“粘贴”(Paste)。
 - 如果将该块粘贴到源块所在的 CPU，则会在副本名称后面附加“_<连续的编号>”。
 - 如果将该块粘贴到其中已有同名块的其它 CPU 中，则将打开“粘贴”(Paste) 对话框。选择所需的选项并单击“确定”(OK) 进行确认。

参见

复制和插入块的原则 (页 378)

7.2.3.1 输入块标题

块标题是块的标题。它与块名称不同，块名称是在创建块时分配的。块标题的长度被限制为一行。可以为打开的块和已关闭的块输入块标题。

要求

代码块可用。

为打开的块输入块标题

要在打开的块中插入块标题，请按以下步骤操作：

1. 在程序编辑器的块标题栏中单击“.....”。
- “.....”文本行将被选中。
2. 输入块标题。

为已关闭的块输入块标题

要在已关闭的块中插入块标题，请按以下步骤操作：

1. 在项目树中右键单击该块。
2. 在快捷菜单中选择“属性”(Properties) 命令。
将打开“属性”(Properties) 对话框。
3. 在区域导航中选择“信息”(Information)。
4. 在“标题”(Title) 输入域中输入块标题。
5. 单击“确定”(OK) 确认输入。

7.2.3.1 输入块注释

可以使用块注释对全部代码块加以说明。例如，可以指出块的用途或提醒用户注意某些特性。可以为打开的块和已关闭的块输入块注释。

要求

代码块可用。

为打开的块输入块注释

要在打开的块中插入块注释，请按以下步骤操作：

1. 单击块标题前的小箭头。
右箭头将变为下箭头，并显示注释区域。
2. 在注释区域中单击“注释”(Comment)。
“注释”(Comment) 文本行将被选中。
3. 输入块注释。

为已关闭的块输入块注释

要在已关闭的块中插入块注释，请按以下步骤操作：

1. 在项目树中右键单击该块。
2. 在快捷菜单中选择“属性”(Properties) 命令。
将打开“属性”(Properties) 对话框。
3. 在区域导航中选择“信息”(Information)。
4. 在“注释”(Comment) 输入域中输入块注释。
5. 单击“确定”(OK) 确认输入。

7.2.3.2 指定块属性和参数

7.2.3.2 块属性和参数的基本知识

块属性

每个块都有一些属性，用户可以显示和编辑这些属性。这些属性用于：

- 标识块
- 显示块的存储器要求和编译状态
- 显示时间戳
- 显示参考信息
- 指定访问保护

块参数

组织块有一些块参数，可用于为特定类型的行为分配参数，例如，将事件分配给组织块。

参见

显示和编辑块属性 (页 383)

设置助记符 (页 409)

7.2.3.2 块属性

7.2.3.2 块属性概述

概述

下表显示了块属性：

组	属性	描述
常规	名称	站内唯一的块名称
	常量名称	PLC 变量表中为 OB 粘贴的常量的名称
	类型	块类型 (无法更改)
	编号	块编号
	事件类别	OB 的事件类别 (无法更改)
	语言	块的编程语言
信息	标题	块标题
	注释	块注释
	版本	块的版本号
	作者	作者名、公司名、部门名或其它名称
	系列	块系列名称
	用户自定义 ID	用户创建的 ID
时间戳	块/数据类型	块的创建时间和更改时间 (无法更改)
	接口	块接口的创建时间 (无法更改)
	代码/数据	代码/数据的更改时间 (无法更改)
编译	状态	上次编译运行的详细信息 (无法更改)
	长度	块长度的详细信息 (无法更改)
保护	保护	块的专有技术保护 (无法更改)
属性	仅符号访问	指定该块接口中的变量是否以纯符号方式声明，而不声明绝对地址。您指定该设置作为在项目中创建新块的默认设置。无法更改现有块中的这个设置。 另请参见： PLC 编程的常规设置 (页 398)
	IEC 检查	根据 IEC 61131 标准测试比较运算和算术运算中的操作数的兼容性。必须明确转换不兼容的操作数。 另请参见： 数据类型转换
	在块内处理错误	使用 GetError 或 GetErrorID 指令执行块内的错误处理 (无法更改)。 另请参见： 处理程序执行错误 (页 469)
	数据块在设备中受写保护	指示数据块在目标系统中是否为只读，且在程序运行期间是否无法将其覆盖 (仅限数据块)
	仅存储在装载存储器中	激活时，数据块仅存储在装载存储器中，不占用工作存储器的空间，且不链接到程序。“移动”(Move) 部分中的“指令”(Instructions) 任务卡提供了用于将数据块传送到工作存储器的选项。(仅限数据块)
触发器	触发器	将组织块分配给事件，通过该事件可以启动组织块。(仅硬件中断 OB) 另请参见： 给硬件中断 OB 分配参数 (页 384)

组	属性	描述
循环中断	循环中断	循环中断 OB 的设置 另请参见： 给循环中断 OB 分配参数 (页 385)

参见

[显示和编辑块属性 \(页 383\)](#)

[仅块的符号寻址 \(页 335\)](#)

[保护块 \(页 496\)](#)

7.2.3.2 显示和编辑块属性

可以显示和编辑的属性根据所选的块而变化。只能显示的属性受写保护。

显示和编辑已关闭的块的属性

要显示和编辑已关闭块的属性，请按以下步骤操作：

1. 在项目树中打开“程序块”(Program blocks) 文件夹。
2. 右键单击要显示或编辑其属性的块。
3. 在快捷菜单中选择“属性”(Properties) 命令。
将打开该块的属性对话框。
4. 在区域导航中，单击要显示或编辑其属性的块。
5. 更改相关属性。
6. 单击“确定”(OK) 确认输入。

显示和编辑打开块的属性

要显示或编辑打开块的属性，请按以下步骤操作：

1. 在“视图”(View) 菜单中，选中“巡视窗口”(Inspector window) 复选框。
巡视窗口随即打开。
2. 单击“属性”(Properties) 选项卡。
3. 单击块标题。
将在巡视窗口的“属性”(Properties) 选项卡中显示块的属性。
4. 在区域导航中，单击要显示或编辑其属性的块。
5. 更改相关属性。

结果

将更改具体的块属性。在保存项目前不会保存所作更改。

参见

[块属性概述 \(页 381\)](#)

7.2.3.2 组织块的块参数

7.2.3.2 块参数的基本知识

简介

许多组织块 (OB) 都有用户可用来控制其行为或将其分配给特定事件的属性。通过分配参数可编辑这些属性。

概述

可以给以下组织块的属性分配参数：

- 硬件中断 OB
- 循环中断 OB

参见

给硬件中断 OB 分配参数 (页 384)

给循环中断 OB 分配参数 (页 385)

7.2.3.2 给硬件中断 OB 分配参数

简介

必须选择相应的事件并为将触发硬件中断的各输入通道和高速计数器分配以下参数：

- 事件名称
- 分配给该过程事件的硬件中断 OB 的编号

在相应设备的属性中分配硬件中断的参数。最多可以为 50 个硬件中断 OB 分配参数。

可以在事件激活前或激活期间创建要分配参数的硬件中断 OB。

步骤

要为硬件中断 OB 分配参数，请按以下步骤操作：

1. 在项目树中双击“设备和网络”(Devices & Networks) 命令。
将在网络视图中打开硬件和网络编辑器。
2. 切换到设备视图。
3. 如果在设备视图中巡视窗口已关闭，则在“视图”(View) 菜单中选中“巡视窗口”(Inspector window) 复选框。
巡视窗口随即打开。
4. 单击“属性”(Properties) 选项卡。
5. 在设备视图中，单击要为其分配硬件中断的模块。
6. 选择相应的事件。

7. 输入事件名称。
8. 从“硬件中断”(Hardware interrupt) 下拉列表中选择现有硬件中断 OB。

注意

如果先前未创建任何硬件中断 OB，则可以单击该下拉列表中的“添加新块”(Add new block)。
另请参见：[创建组织块 \(页 374\)](#)

9. 如果要分配其它硬件中断，则重复步骤 5 到 8。

参见

[块参数的基本知识 \(页 384\)](#)
[硬件中断组织块 \(页 317\)](#)
[事件和 OB \(页 298\)](#)
[CTRL_HSC：在 FBD 中控制高速计数器 \(页 775\)](#)

7.2.3.2 给循环中断 OB 分配参数

简介

可以使用循环中断 OB 以固定时间间隔启动程序。为此，必须为所使用的每个循环中断 OB 输入扫描时间和相位偏移。

在用户程序中最多可使用 4 个循环中断 OB 或延时 OB (OB 编号大于等于 200)。例如，如果已使用 2 个延时中断 OB，则在用户程序中最多可以再插入 2 个循环中断 OB。

注意

如果分配多个循环 OB，请确保给每个循环中断 OB 分配不同的循环时间或相位偏移，以避免它们同时执行或必须排队。创建循环中断 OB 时，将输入循环时间 100 和相位偏移 0 作为起始值。

步骤

要为循环中断 OB 输入扫描时间和相位偏移，请按以下步骤操作：

1. 在项目树中打开“程序块”(Program blocks) 文件夹。
2. 右键单击现有的循环中断 OB。
3. 在快捷菜单中选择“属性”(Properties) 命令。
将打开“<循环中断 OB 的名称>”对话框。
4. 在区域导航中单击“循环中断”(Cyclic interrupt) 组。
将显示扫描时间和相位偏移文本框。

5. 输入扫描时间和相位偏移。
6. 单击“确定”(OK) 确认输入。

参见

创建组织块 (页 374)
块参数的基本知识 (页 384)
循环中断组织块 (页 316)

7.2.3.3 管理块

7.2.3.3 保存块

块总是随项目一起保存。 同时也可保存错误块。 这样，便可以在方便时解决错误。

步骤

另请参见：[保存项目 \(页 134\)](#)

7.2.3.3 关闭块

步骤

要关闭块，请按下列步骤操作：

1. 在程序编辑器的块标题栏中单击“关闭”(Close) 按钮。

注意事项

请注意，关闭块时不会保存块。

7.2.3.3 重命名块

要求

“程序块”(Program blocks) 文件夹在项目树中已打开。

步骤

要更改块名称，请按下列步骤操作：

1. 右键单击要重命名的块。
2. 在快捷菜单中选择“重命名”(Rename) 命令。
项目树中的块名称将变为输入域。
3. 输入新的块名称。
4. 按 Enter 键确认输入。

结果

这样在程序中的所有使用点更改了该块名称。

7.2.3.3 离线删除块

要求

“程序块”(Program blocks) 文件夹在项目树中已打开。

步骤

要离线删除现有块，请按以下步骤操作：

1. 在项目树的“程序块”(Program blocks) 文件夹中，右键单击要删除的块。
2. 在快捷菜单中选择“删除”(Delete) 命令。
3. 单击“是”(Yes) 确认安全提示。
将从项目中离线删除该块。

注意

如果要删除组织块，请注意可能给这些块分配了事件。如果删除这类组织块，程序将无法响应参数化的事件。

参见

下载块 (页 488)

7.2.3.3 删除在线块

要求

可访问节点的“程序块”(Program blocks) 文件夹已打开。

步骤

要删除现有在线块，请按以下步骤操作：

1. 在可访问节点的“程序块”(Program blocks) 文件夹中，右键单击要删除的块。
2. 在快捷菜单中选择“删除”(Delete) 命令。
3. 单击“是”(Yes) 确认安全提示。
 将从设备中删除该块。

7.2.4 对块进行编程

7.2.4.1 程序编辑器

7.2.4.1 程序编辑器概述

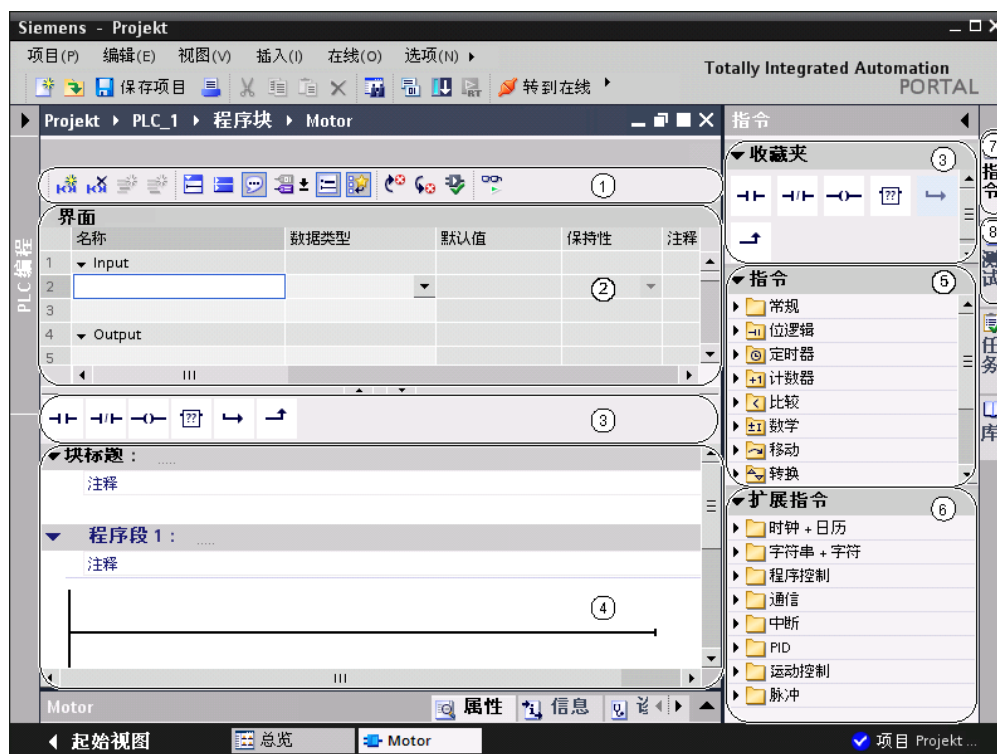
程序编辑器的功能

程序编辑器是用于设计功能、功能块和组织块的集成开发环境。它为编程和故障排除提供了广泛的支持。

程序编辑器的外观和功能随所用的编程语言不同而变化。

程序编辑器的结构

下图以 LAD 为例显示了程序编辑器的各个组成部分：



- ① 工具栏 (页 389)
- ② 块接口 (页 390)
- ③ “指令”任务卡中的“收藏夹”窗格 (页 390)
- ④ 指令窗口 (页 394)
- ⑤ “指令”任务卡中的“指令”窗格 (页 391)
- ⑥ “指令”任务卡中的“扩展指令”窗格 (页 391)
- ⑦ “指令”任务卡 (页 391)
- ⑧ “测试”任务卡 (页 392)

7.2.4.1 功能栏

功能

使用工具栏可以访问程序编辑器的主要功能，例如：

- 插入、删除、打开和关闭程序段
- 显示和隐藏绝对操作数
- 显示和隐藏程序段注释
- 显示和隐藏收藏夹
- 跳到语法错误
- 更新块调用

- 显示和隐藏程序状态

7.2.4.1 块接口

功能

通过块接口可以创建和管理局部变量。

另请参见：[声明块接口 \(页 398\)](#)。

表示法

块接口以表格形式显示。列数取决于块类型。下表说明了各列的含义：

列	描述
名称	变量名称。 可以在块接口的各部分声明变量。 另请参见： 块接口的布局 (页 398)
数据类型	变量的数据类型。
默认值	预分配给代码块接口中的变量的值。 可选择是否指定默认值。如果未指定任何值，将使用指定数据类型的预定义值。例如，为 BOOL 预定义了值“false”。 在相应的背景数据块中将默认值作为初始值。可以在背景数据块中使用实例特定的初始值替换这些值。 仅功能块的接口中提供了该列。
保持性	将变量标记为具有保持性。 保持性变量的值将保留，即使在电源关闭后也是如此。 只有将块标记为“仅符号访问”(Symbolic access only)，该列才可见。 另请参见： 仅块的符号寻址 (页 335) 仅功能块的接口中提供了该列。
注释	用于描述变量的注释。

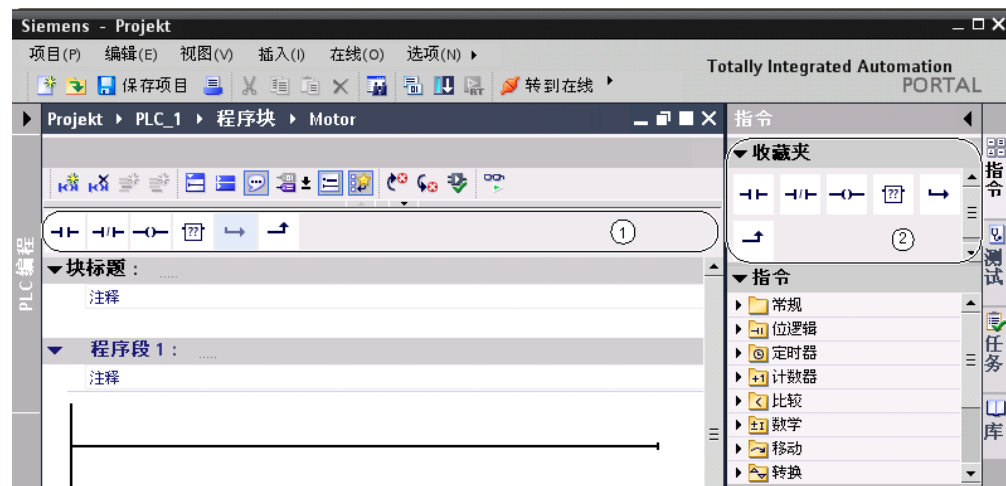
7.2.4.1 收藏夹

功能

通过收藏夹可以快速访问常用的指令。

程序编辑器将在“指令”(Instructions) 任务卡显示用户创建的收藏夹。这样，即使用户将其它任务卡置于前景或已关闭所有任务卡，用户仍可以访问这些收藏夹。可以使用程序编辑器工具栏显示和隐藏这些收藏夹。

下图所示为“指令”(Instructions) 任务卡的“收藏夹”(Favorites) 窗格中以及程序编辑器中的收藏夹：



- ① 程序编辑器中的收藏夹
- ② “指令”(Instructions) 任务卡的“收藏夹”(Favorites) 窗格中的收藏夹

参见

将元素添加到收藏夹 (页 420)

7.2.4.1 “指令”任务卡

功能

通过“指令”(Instructions) 任务卡可以很容易地访问创建程序时可使用的所有操作。

“指令”任务卡的布局

“指令”(Instructions) 任务卡由以下部分组成：



- ① “收藏夹”(Favorites) 窗格
- ② “指令”(Instructions) 窗格
- ③ “扩展指令”(Extended instructions) 窗格

参见

使用“指令”任务卡插入 LAD 元素 (页 417)
使用“指令”任务卡插入 FBD 元素 (页 447)

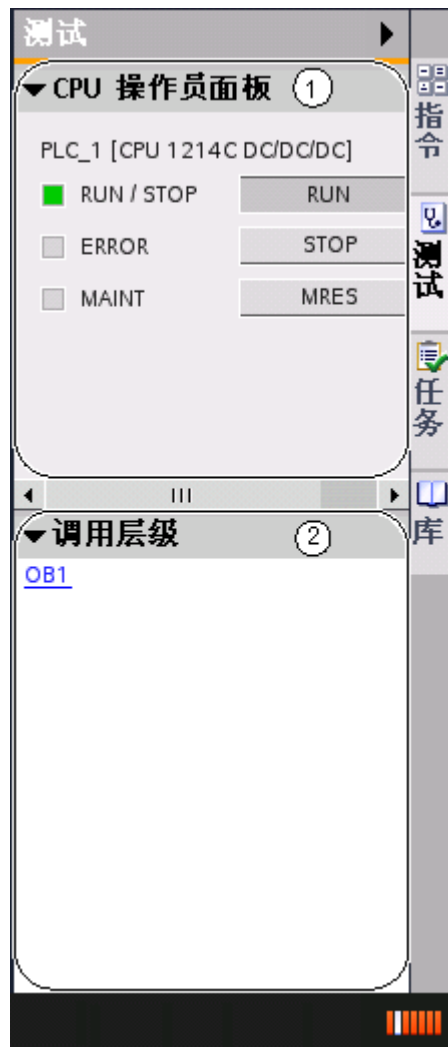
7.2.4.1 “测试”任务卡

“测试”任务卡的功能

在“测试”任务卡中，可以通过程序状态对故障排除进行设置。“测试”(Test) 任务卡的功能仅在在线模式下可用。

“测试”任务卡的布局

“测试”任务卡由以下部分组成：



- ① CPU 控制面板
- ② 调用层级

CPU 控制面板

在 CPU 控制面板中，可以切换 CPU 工作模式。

另请参见：[更改 CPU 的模式 \(页 270\)](#)

调用层级

在该面板中，可以跟踪块的调用层级。只能在监视块期间查看调用层级。

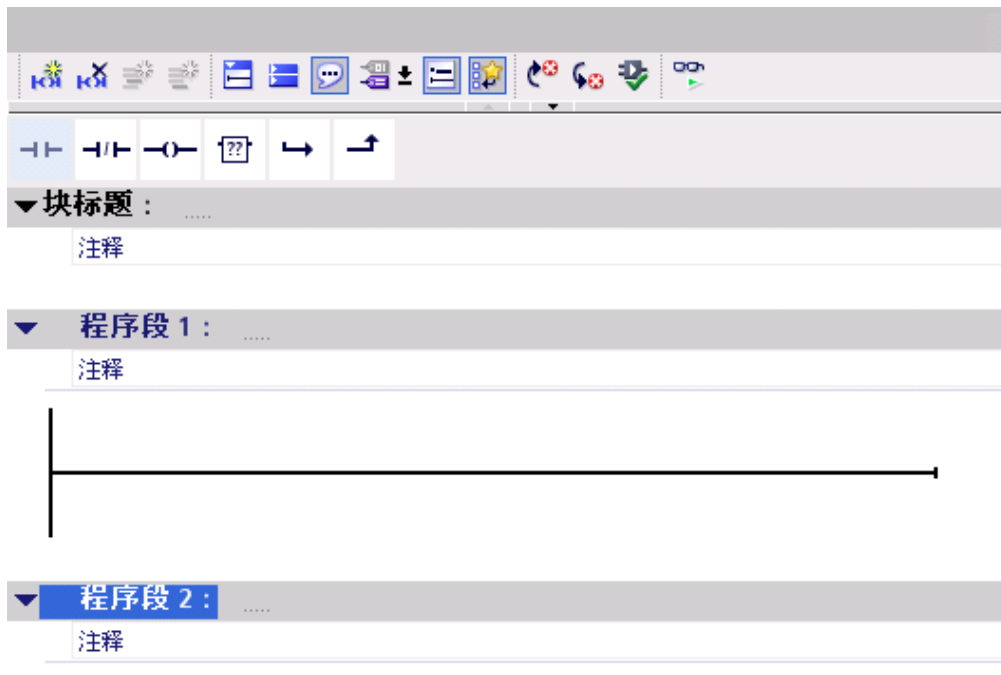
7.2.4.1 指令窗口

功能

指令窗口是程序编辑器的工作区。在其中可执行以下任务：

- 创建和管理程序段
- 输入块和程序段的标题与注释
- 输入程序代码

下图所示为程序编辑器的指令窗口：



参见

创建程序代码 (页 409)

7.2.4.1 扩大指令窗口的工作区

简介

当应用程序的所有组成部分都显示时，指令窗口区域相对比较小。如果程序代码很多，则您可能会发现必须经常重新排列该工作区。为避免此问题，您可以隐藏或最小化应用程序和程序编辑器的以下组成部分的显示：

- 项目树
- 任务卡
- 块接口
- 收藏夹

- 注释
- 程序段

隐藏和显示项目树

通过项目树可以访问项目的所有区域。因此，项目打开后会显示项目树。可以在创建程序时隐藏项目树，以获得更大的指令窗口区域。

要显示和隐藏项目树，请按以下步骤操作：

1. 在“视图”(View) 菜单中选择“项目树”(Project tree) 或在项目树标题栏中单击“折叠”(Collapse)。

打开和关闭任务卡

任务卡位于应用程序窗口的右侧边上。

要打开或关闭任务卡，请按以下步骤操作：

1. 在“视图”(View) 菜单中选择“任务卡”(Task card) 复选框或在任务卡标题栏中单击“折叠”(Collapse) 或“展开”(Expand)。

隐藏和显示块接口

块打开后，块接口显示在程序编辑器的上方区域。编程期间，可以根据需要显示和隐藏块接口。

要显示和隐藏块接口，请按以下步骤操作：

1. 在接口下部的窗口拆分条中，单击上箭头或下箭头。

显示和隐藏收藏夹

要显示或隐藏程序编辑器中的收藏夹，请按以下步骤操作：

1. 在程序编辑器工具栏中单击“也在编辑器中显示收藏夹”(Show favorites also in the editor) 按钮。

显示和隐藏注释

可以在块中为块或各程序段输入注释。这两类注释的显示和隐藏各不相同。

要显示或隐藏块注释，请按以下步骤操作：

1. 单击块标题行起始位置处的三角形。

要显示或隐藏程序段注释，请按以下步骤操作：

1. 在程序编辑器工具栏中单击“启用/禁用程序段注释”(Network comments on/off)。

打开和关闭程序段

可以分别或一起打开或关闭程序中的多个程序段。

要打开或关闭程序段，请按下列步骤操作：

1. 如果要打开程序段，请单击程序段标题前的右箭头。如果要关闭程序段，请单击程序段标题前的下箭头。

要打开和关闭程序段，请按下列步骤操作：

1. 在程序编辑器工具栏中单击“打开所有程序段”(Open all networks) 或“关闭所有程序段”(Close all networks)。

7.2.4.1 在程序编辑器中使用键盘

在程序段中浏览

功能	所选对象	键盘快捷键
浏览程序段中的对象。	程序段中的对象	箭头键

编辑指令 (LAD/FBD)

功能	所选对象	键盘快捷键
删除操作。	操作	

输入操作数 (LAD/FBD)

功能	所选对象	键盘快捷键
激活操作的第一个操作数的输入域。	操作	<Enter> 或 <任意字母> <Return> 打开空输入域；按任何其它字母，该字母将输入到输入域中。
激活操作数的输入域。	操作数	<F2>
删除操作数的值。		
定义变量		<Alt+Shift+D>
输入操作数	操作数的输入域	<任意字母/数字>
确认操作数的输入。		<Enter>
打开操作数选择项。		<Ctrl+I>
放弃当前更改。		<Esc> 禁用输入域并恢复先前的内容。

参见

用于项目编辑的键盘快捷键 (页 119)
项目树中的键盘快捷键 (页 121)

用于窗口的键盘快捷键 (页 120)
 表格中的键盘快捷键 (页 121)
 用于文本编辑的键盘快捷键 (页 122)

7.2.4.1 PLC 编程的常规设置

7.2.4.1 常规设置概述

概述

下表给出了用户可进行的常规设置：

组	设置	描述
视图	具有注释	显示注释
打印	具有接口	同时打印输出块接口
	具有注释	同时打印输出注释
	缩放因子	缩放要打印输出的程序段
编译	接口更新时删除实参	在被调用块中删除相关的形参后，在运行“更新块调用”(Update block call) 功能时将删除实参。
新块的默认设置	IEC 检查	将检查比较运算和算术运算中的多个操作数是否一致。需要明确转换不一致的操作数。
	仅符号访问	该块接口中的变量以纯符号方式声明，而不声明绝对地址，因此只能对其进行符号寻址。
附加设置	显示操作数选择项	显示操作数选择项
	助记符	指令的德语和国际表示法

参见

更改设置 (页 398)
 将操作数插入 LAD 指令 (页 430)
 将操作数插入 FBD 指令 (页 457)
 可用的 PLC 变量地址和数据类型 (页 365)

7.2.4.1 更改设置

步骤

为改变设置，可进行如下操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
工作区中将显示“设置”(Settings) 窗口。
2. 在区域导航中，选择“PLC 编程”(PLC programming) 组。
3. 更改设置。

结果

更改将直接装载，无需显式保存。

7.2.4.2 代码块编程

7.2.4.2 声明块接口

7.2.4.2 块接口的布局

定义

另请参见：[块接口 \(页 390\)](#)

接口包含只能在块中使用的局部变量的声明。变量划分为两组：

- 在程序中调用时构成块接口的块参数。
- 用于存储中间结果的局部数据。

块接口以表格形式显示在程序编辑器的上方区域。

块参数

下表显示了块参数的类型：

类型	部分	功能	可用于
输入参数	输入	其值由块读取的参数。	功能、功能块和某些类型的组织块
输出参数	输出	其值由块写入的参数。	功能和功能块
参数 InOut	InOut	调用时由块读取其值，执行后又由块写入其值的参数。	功能和功能块

局部数据

下表显示了局部数据的类型：

类型	部分	功能	可用于
临时局部数据	临时	用于存储临时中间结果的变量。只保留一个周期的临时局部数据。	功能、功能块和组织块
静态局部数据	静态	用于在背景数据块中存储静态中间结果的变量。静态数据会一直保留到被覆盖，这可能在几个周期之后。	功能块

7.2.4.2 局部变量声明的用途

变量声明的用途

使用变量声明定义要在块中使用的变量的名称和数据类型。

在功能块中，可以为块参数和静态局部数据分配默认值。

变量声明的作用

变量声明具有以下作用：

- 功能块的变量声明可在背景数据块中保留存储空间。
- 代码块的变量声明可确定程序中功能块的调用接口。
- 功能块的变量声明可确定分配给 FB 的每个背景数据块的数据结构。

参见

PLC 变量和局部变量 (页 337)
块接口 (页 390)

7.2.4.2 局部变量的规则

7.2.4.2 代码块接口中有效的数据类型

对组织块接口有效的数据类型

以下规则适用于组织块的接口：

- “临时”(TEMP) 和“输入”(INPUT) 部分可包含基本和复杂数据类型以及 VARIANT 数据类型。不允许使用其它参数类型。
- 组织块没有输入、输出或输入/输出参数，因为它不会被其它块所调用。

- 组织块没有静态变量，因为它没有背景数据块。

下表显示了对组织块接口有效的数据类型：

部分	基本数据类型	复杂数据类型	VARIANT 参数类型
临时	◆	◆	◆
输入	◆	◆	◆

◆: 有效分配

-: 无效分配

对功能块接口有效的数据类型

以下规则适用于功能块的接口：

- 声明输入参数时，允许使用基本和复杂数据类型以及 VARIANT 参数类型。
- 声明输出参数时不允许使用任何参数类型。
- 声明输入/输出参数时，只允许使用 VARIANT 参数类型。
- 临时变量的声明只允许使用 VARIANT 参数类型。其它所有参数类型均无效。

下表显示了对功能块接口有效的数据类型：

部分	基本数据类型	复杂数据类型	VARIANT 参数类型
输入	◆	◆	◆
输出	◆	◆	-
InOut	◆	◆ ⁽³⁾	◆
静态	◆	◆	-
临时	◆	◆	◆

◆: 有效分配

-: 无效分配

⁽³⁾ 只能将 STRING 定义为默认 254 个字符的长度。

对功能接口有效的数据类型

以下规则适用于功能的接口：

- 功能没有静态变量，因为它没有背景数据块。
- 在输入、输出和 InOut 部分中只允许使用 VARIANT 参数类型。禁止使用其它所有参数类型。
- “临时”部分可包含基本和复杂数据类型以及 VARIANT 数据类型。不允许使用其它参数类型。

下表显示了对功能接口有效的数据类型：

部分	基本数据类型	复杂数据类型	VARIANT 参数类型
输入	◆	◆(2)	◆
输出	◆	◆(2)	◆
InOut	◆	◆(2)	◆
临时	◆	◆	◆

◆: 有效分配

-: 无效分配

(2) 只能将 STRING 定义为默认 254 个字符的长度。

7.2.4.2 声明局部变量

7.2.4.2 在块接口中声明局部变量

要求

块已打开。

步骤

要声明基本数据类型的变量，请按以下步骤操作：

1. 选择接口中的相应部分：

- 输入
- 输出
- InOut
- 静态
- 临时

另请参见：[块接口的布局 \(页 398\)](#)

在“名称”(Name) 列中输入变量名称。

2. 单击“数据类型”(Data type) 列中的箭头键，然后选择所需的数据类型。

在“默认值”(Default value) 列中将会输入该数据类型的默认初始值。

3. 必要时，可更改默认值。

结果

创建该变量。

语法检查

每次输入后都会执行语法检查，并且找到的任何错误都将以红色显示。无需立即更正这些错误，您可以继续编辑并在以后进行所有更正。但是，如果变量声明包含语法错误，将无法编译程序。

注意

如果更改块的接口，则还必须更改程序中调用该块的所有位置。

如果在项目树中选择“程序块”(Program blocks) 文件夹并进行编译，则会自动更新调用位置。

参见

在接口中插入表行 (页 373)

在末尾添加表行 (页 408)

表格中的键盘快捷键 (页 121)

编辑块接口 (页 372)

有关初始值的基本信息 (页 477)

7.2.4.2 在程序编辑器中直接定义局部变量

要求

程序中使用一个或多个操作数。

步骤

要将操作数声明为局部变量，请按以下步骤操作：

1. 选择一个或多个操作数。
2. 在快捷菜单中选择“定义变量”(Define tag) 命令。
将打开“定义变量”(Define tag) 对话框。此对话框会显示一个声明表，其中已输入所选操作数的名称。
3. 将光标放置在“部分”(Section) 列中。
将显示一个箭头按钮。
4. 单击该箭头按钮。
5. 要声明局部变量，请选择以下部分之一：
 - 局部输入
 - 局部输出
 - 局部 InOut
 - 局部静态
 - 局部临时

6. 在其它列中，输入数据类型和注释。
7. 单击“定义”(Define) 按钮完成输入。

结果

声明将直接写入块接口并在整个块内有效。

注意

如果更改块的接口，则还必须更改程序中调用该块的所有位置。

如果在项目树中选择“程序块”(Program blocks) 文件夹并进行编译，则会自动更新调用位置。

7.2.4.2 声明 ARRAY 数据类型的变量

要求

块已打开。

步骤

要声明 ARRAY 数据类型的变量，请按以下步骤操作：

1. 在接口的“名称”(Name) 列中输入变量名称。
2. 在“数据类型”(Data type) 列中，单击箭头键并选择数据类型“类型的 Array [o .. hi]”。
3. 然后在同一列中输入维数，用于显示上限、下限和数据类型（例如，BOOL 类型的 ARRAY [1..10]）。在右方括号和关键字“of”之间以及在“of”和 ARRAY 元素的数据类型信息之间必须插入一个空格。

注意

用户无法为 ARRAY 元素定义特定的默认值。但是，用户可以在背景数据块中为它们分配一个起始值。

参见

ARRAY 的格式 (页 354)
一维 ARRAY 的实例 (页 355)

7.2.4.2 声明 STRUCT 数据类型的变量

要求

块接口已打开。

步骤

要声明 STRUCT 数据类型的变量，请按以下步骤操作：

1. 在“名称”(Name) 列中输入变量名称。
2. 在“数据类型”(Data type) 列中，单击箭头键并选择“Struct”条目。
在新变量后将插入两个空行。
3. 在第一个空行中插入第一个结构元素。
在该元素后会再插入一个空行。
4. 选择结构元素的数据类型。
5. 也可以输入结构元素的默认值。
6. 对所有其它结构元素重复步骤 3 到 5。
不必显式结束结构。结构会以输入的最后一个元素结束。
7. 要在结构后插入新变量，请在结构结尾后留一个空行，然后在第二个空行中启动新变量。

参见

STRUCT (页 355)

7.2.4.2 更改局部变量的名称

可以在以下位置更改局部变量的名称：

- 在块接口中
- 就在程序编辑器中

步骤

要更改局部变量的名称，请按以下步骤操作：

1. 在块接口中选择变量。
 2. 更改“名称”(Name) 列中的条目。
- 或
1. 在程序中变量的使用点选择变量。
 2. 在快捷菜单中选择“重命名变量”(Rename tag) 命令。
将打开“重命名变量”(Rename tag) 对话框。
 3. 更改“名称”(Name) 列中的条目。

结果

自动在程序中的所有使用点更改了变量名称。

7.2.4.2 更改局部变量的数据类型

可以在块接口中更改局部变量的数据类型。

步骤

要更改变量的数据类型，请按以下步骤操作：

1. 在块接口中选择变量。
2. 单击“数据类型”(Data type) 列中的箭头按钮。
将显示允许选用的数据类型。选择项取决于块类型和部分。
另请参见：[代码块接口中有效的数据类型 \(页 399\)](#)
3. 在所显示的数据类型中选择一个。
4. 可根据需要更改变量的默认值。

结果

将更改数据类型。

7.2.4.2 定义局部变量的默认值

可以在功能块的接口中定义每个变量的默认值。在创建背景数据块的过程中会将这些默认值用作初始值。然后可以在背景数据块中使用实例特定的初始值替换这些值。

在块接口中不能为 ARRAY 数据类型的变量分配任何默认值。

要求

在接口的某行中输入了变量名称。

步骤

要更改变量的默认值，请按以下步骤操作：

1. 单击块接口中的“默认值”(Default value) 列。
2. 输入适用于所示数据类型的默认值。

注意

如果没有输入默认值，将使用所示数据类型的标准值。例如，BOOL 数据类型的默认值为“False”。

参见

有关初始值的基本信息 (页 477)

7.2.4.2 设置局部变量的保持性

7.2.4.2 代码块的保持特性和符号寻址

块的符号访问

您可以将块定义为“仅符号访问”。如果该设置为“仅符号访问”，则可以纯符号方式声明块变量，无需指定绝对地址。

局部变量的保持特性

设置保持性的选项取决于所设置的块的可寻址方式。

- 在只能以符号方式访问的块中，可以将各个变量定义为具有保持性。
- 在可通过其它方式访问的块中，不能对块接口中的各个变量进行保持性设置。只能将分配的背景数据块定义为具有保持性。该块包含的所有变量随后会被视为具有保持性。

下表概要说明了可能设置：

块类型	启用了块属性“仅符号访问”	禁用了块属性“仅符号访问”
FB	可为各个变量设置保持性。	不能设置保持性。
背景数据块	通过更高级别的代码块进行保持性设置。	可为整个背景数据块设置保持性。

参见

仅块的符号寻址 (页 335)

7.2.4.2 设置功能块中局部变量的保持特性

简介

如果启用了符号访问，则可以设置功能块中各个变量的保持特性。块接口的“保持性”(Retentivity)列提供了一个下拉列表，其中包含以下选择项：

- **保持性**
变量定义为具有保持性。该设置会被所分配的背景数据块接受，且无法在此块中更改。
- **非保持**
变量定义为不具有保持性。该设置会被所分配的背景数据块接受，且无法在此块中更改。

如果未进行任何设置，则变量被定义为不具有保持性。

要求

- 在打开的功能块中启用了变量的符号访问。
- 在块接口中声明变量

步骤

要设置局部变量的保持特性，请按以下步骤操作：

1. 在“保持性”(Retain) 列中，选择要为其设置保持特性的变量的单元格。
2. 单击该单元格右边框上的箭头。

将打开含有“保持性”(Retain) 和“非保持”(Non-retain) 条目的下拉列表。

3. 从该下拉列表中选择所需的设置。

7.2.4.2 编辑局部变量的属性

7.2.4.2 编辑局部变量的属性

可在以下页面中找到局部变量属性的说明：

[块接口的布局 \(页 398\)](#)

在块接口中编辑属性

要编辑一个或多个变量的属性，请按以下步骤操作：

1. 打开块接口。
2. 更改“名称”(Name)、“注释”(Comments) 和其它列中的条目。

在程序编辑器中直接重命名变量

要重命名一个或多个变量，请按以下步骤操作：

1. 在程序中选择一个或多个变量。
2. 在快捷菜单中选择“重命名变量”(Rename tag) 命令。

将打开“重命名变量”(Rename tag) 对话框。此对话框会显示包含所选变量的声明表。

3. 更改“名称”(Name) 列中的条目。
4. 单击“更改”(Change) 按钮确认输入。

7.2.4.2 编辑块接口

7.2.4.2 在 PLC 变量表中删除条目

步骤

要删除变量，请按以下步骤操作：

1. 选择含有要删除的变量的行。也可以在按住 <Ctrl> 键时依次单击行或按住 <Shift> 键单击第一行和最后一行来选择多行。
2. 在快捷菜单中选择“删除”(Delete) 命令。

参见

调整表格 (页 119)

表格中的键盘快捷键 (页 121)

7.2.4.2 插入表格行

步骤

要在鼠标指针位置上方插入行，请按以下步骤操作：

1. 将鼠标指针放置在要在其上方插入一个新行的行中。
2. 在表格的工具栏上，单击“插入行”(Insert row) 按钮。
将在所选行上方插入一个新行。

参见

表格中的键盘快捷键 (页 121)

调整表格 (页 119)

7.2.4.2 在末尾添加表行

步骤

要在表的结尾或复杂数据类型的结尾添加新行，请按以下步骤操作：

1. 单击表工具栏上的“添加行”(Add row) 按钮。
在表的末尾或复杂数据类型的结尾将添加一个新的空行。

参见

表格中的键盘快捷键 (页 121)

调整表格 (页 119)

7.2.4.2 创建程序代码

7.2.4.2 设置助记符

可以使用德语或国际助记符对块进行编程。如果首次打开 TIA Portal，国际助记符被设置为默认值。可以随时更改助记符。

步骤

要设置助记符，请按以下步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
工作区中将显示“设置”(Settings) 窗口。
2. 在区域导航中选择“常规”(General) 组。
3. 在“常规设置”(General settings) 组中，选择要使用的助记符。
将更改所有块中的助记符。

7.2.4.2 创建 LAD 程序

7.2.4.2 有关 LAD 的基本信息

7.2.4.2 LAD 编程语言

梯形逻辑 (LAD) 编程语言概述

LAD 是一种图形编程语言。它采用基于电路图的表示法。

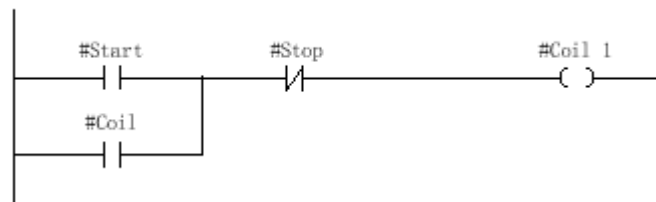
程序以一个或多个程序段表示。在程序段的左边和右边都有一根引出梯级的电源线。二进制信号以触点的形式排列在梯级上。元素在分支上的顺序排列构成串联，并行分支上的排列构成并联。复杂功能用功能块表示。

梯级以写入逻辑运算结果的线圈结束。

LAD 程序段实例

下图显示了一个具有常开触点、常闭触点和一个线圈的 LAD 程序段：

程序段 1



7.2.4.2 LAD 元素概述

LAD 元素

LAD 程序由可在程序段电源线上串联或并联的各个元素组成。必须给多数程序元素提供变量。对于定时器和计数器运算，还必须创建数据块以存储形参。

至少有一个梯级从电源线引出。程序段的编程从梯级的左边开始。可以使用多个梯级和分支延长电源线。

例如，下图所示为一个 LAD 程序段的元素：

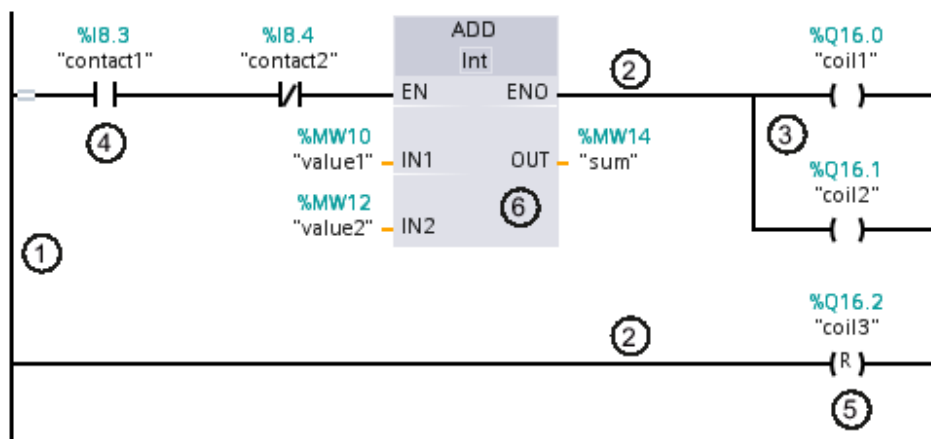


图 7-2

- 1) 电源线
- 2) 梯级
- 3) 分支
- 4) 触点
- 5) 线圈
- 6) 功能框

电源线

每个 LAD 程序段都包含至少有一个梯级的电源线。通过添加其它梯级可扩展程序段。可以使用分支在特定梯级中创建并联结构。

触点

可以使用触点创建或中断两个元素之间的载流连接。在这种情况下，元素可以是 LAD 程序元素或电源线的边。电流从左向右传递。可以使用触点查询操作数的信号状态或值，并根据电流的结果对其进行控制。

在 LAD 程序中可以使用以下类型的触点：

- 常开触点：
如果指定二进制操作数的信号状态为“1”，则常开触点传送电流。

- 常闭触点：
如果指定二进制操作数的信号状态为“0”，则常开触点传送电流。
- 带附加功能的触点：
如果满足特定条件，则带附加功能的触点传送电流。还可以使用这些触点执行附加功能，例如 RLO 沿检测和比较。

线圈

可以使用线圈控制二进制操作数。线圈可根据逻辑运算结果的信号状态置位或复位二进制操作数。

在 LAD 程序中可以使用以下类型的线圈：

- 标准线圈：
如果电流流经线圈，则标准线圈置位二进制操作数。“继电器线圈，输出”运算是标准线圈的一个实例。
- 带附加功能的线圈：
这些线圈除了判断逻辑运算的结果，还具有附加功能。用于 RLO 沿检测和程序控制的线圈是带附加功能的线圈的实例。对于用于程序控制的线圈，需要指定跳转目标，而不是操作数。

功能框

功能框是具有复杂功能的 LAD 元素。但空功能框除外。可以使用空功能框作为占位符，在其中可以选择所需的运算。

在 LAD 程序中可以使用以下类型的功能框：

- 无 EN/ENO 机制的功能框：
功能框根据其输入的信号状态执行。无法查询处理过程中的错误状态。
- 具有 EN/ENO 机制的功能框：
只有使能输入“EN”的信号状态为“1”时才执行功能框。如果正确处理了该功能框，则“ENO”使能输出的信号状态为“1”。如果处理期间出错，则复位“ENO”输出。

代码块的调用在程序段中也显示为具有 EN/ENO 机制的功能框。

参见

LAD 元素的使用规则 (页 416)

7.2.4.2 LAD 的设置

7.2.4.2 LAD 的设置概述

概述

下表给出了用户可以进行的设置：

组	设置	描述
字体	大小	程序编辑器中的字体大小

组	设置	描述
视图	布局	“紧凑”或“宽幅” 更改操作数与其它对象（例如操作数和触点）之间的垂直间距。重新打开块时即可看到所做的更改。
	具有绝对信息	附加显示绝对地址
操作数域	最大宽度	在操作数域中可水平输入的最多字符数
	最大高度	在操作数域中可垂直输入的最多字符数

7.2.4.2 更改设置

步骤

为改变设置，可进行如下操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
工作区中将显示“设置”(Settings) 窗口。
2. 在区域导航中，选择“PLC 编程”(PLC programming) 组。
3. 更改设置。

结果

更改将直接装载，无需显式保存。

7.2.4.2 使用程序段

7.2.4.2 使用程序段

功能

用户程序在块的程序段中创建。对于要编程的代码块，它必须包含至少一个程序段。为获得更佳的用户程序总览效果，还可以将您的程序划分为多个程序段。

参见

插入程序段 (页 413)

选择程序段 (页 413)

复制和插入程序段 (页 414)

删除程序段 (页 414)

展开和折叠程序段 (页 415)

7.2.4.2 插入程序段

要求

块已经打开。

步骤

要插入新程序段，请按以下步骤操作：

1. 选择要在其后插入新程序段的程序段。
2. 在快捷菜单中选择“插入程序段”(Insert network) 命令。

注意

如果在 LAD 或 FBD 程序中将元素插入块的最后一个（且仍为空的）程序段，则新的空程序段会自动在其下方插入。

结果

在块中插入了新的空程序段。

参见

使用程序段 (页 412)

7.2.4.2 选择程序段

要求

有一个可用的程序段。

选择一个程序段

要选择一个程序段，请按以下步骤操作：

1. 单击要选择的程序段的标题栏。

选择多个程序段

要选择多个程序段，请按以下步骤操作：

1. 按住 <Ctrl> 键。
2. 单击要选择的所有程序段。

要选择多个连续的程序段，请按以下步骤操作：

1. 按住 <Shift> 键。
2. 单击要选择的第一个程序段。

3. 单击要选择的最后一个程序段。
将选中第一个和最后一个程序段以及它们之间所有的程序段。

参见

使用程序段 (页 412)

7.2.4.2 复制和插入程序段

复制的程序段可粘贴在同一个块或其它块中。如果将程序段粘贴到用不同编程语言编写的块中，该程序段的编程语言将保持不变。

先决条件

有一个可用的程序段。

步骤

要复制和粘贴程序段，请按以下步骤操作：

1. 选择要复制的一个或多个程序段。
2. 在快捷菜单中选择“复制”(Copy)。
3. 选择要在其后粘贴所复制程序段的程序段。
4. 在快捷菜单中选择“粘贴”(Paste)。

注意

还可以将程序段从一个块复制到另一个块。

参见

使用程序段 (页 412)

7.2.4.2 删除程序段

要求

有一个可用的程序段。

步骤

要删除程序段，请按以下步骤操作：

1. 选择要删除的程序段。
2. 在快捷菜单中选择“删除”(Delete) 命令。

参见

使用程序段 (页 412)

7.2.4.2 展开和折叠程序段

要求

有一个可用的程序段。

打开和关闭程序段

1. 选择要打开或关闭的程序段。
2. 在快捷菜单中选择“展开”(Expand) 或“折叠”(Collapse)。

打开和关闭所有程序段

1. 在工具栏中单击“打开所有程序段”(Open all networks) 或“关闭所有程序段”(Close all networks)。

参见

使用程序段 (页 412)

7.2.4.2 输入程序段标题

程序段标题是程序段的标头。程序段标题的长度无限制，但只能占用一行。

要求

有一个可用的程序段。

步骤

要输入程序段标题，请按以下步骤操作：

1. 双击程序段标题栏中的“.....”。
“.....”文本行将被选中。
2. 输入程序段标题。

7.2.4.2 在 LAD 程序段中输入程序段注释

可以使用程序段注释描述各个程序段。例如，可以指出程序段的功能或提醒用户注意某些特性。

要求

有一个可用的程序段。

步骤

要输入程序段注释，请按以下步骤操作：

1. 单击程序段标题前的小箭头。
右箭头将变为下箭头。
2. 如果注释区域不可见，请单击工具栏中的“启用/禁用程序段注释”(Network comments on/off)。
将显示注释区域。
3. 在注释区域中单击“注释”(Comment)。
“注释”(Comment) 文本行将被选中。
4. 输入程序段注释。

7.2.4.2 插入 LAD 元素

7.2.4.2 LAD 元素的使用规则

规则

插入 LAD 元素时请注意下列规则：

- 每个 LAD 程序段都必须使用线圈或程序框来终止。但是，不可使用下列 LAD 元素来终止程序段：
 - 比较框
 - 用于 RLO 上升/下降沿检测的线圈
- 用于功能框连接的分支起点必须始终为左电源线。逻辑运算或其它功能框可出现在功能框前面的分支中。
- 线圈会自动放置在程序段的右边，用于终止分支。但用于 RLO 上升/下降沿检测的线圈例外。这些线圈不能放置在分支的最左边和最右边。也不允许放在其它分支中。
- 只能将触点插入具有前导逻辑运算的并行分支中。但用于将逻辑运算 (-|NOT|-) 结果取反的触点例外。用于将逻辑运算结果取反的触点以及线圈和功能框可用在并行分支中，前提是它们要直接来源于电源线。
- 用于 RLO 上升/下降沿检测的运算不能直接安排在电源线的左边。
- 跳转运算只能放在末尾，RET 还可放在开头。
- 使能输入“EN”与使能输出“ENO”可连接到功能框，但这不是必需的。
- 不能将常数（例如 TRUE 或 FALSE）分配给常开触点或常闭触点。相反，请使用 BOOL 数据类型的操作数。

有关在电源线的什么位置可插入 LAD 元素的信息，请参见 LAD 参考帮助。

参见

EN/ENO 机制 (页 332)

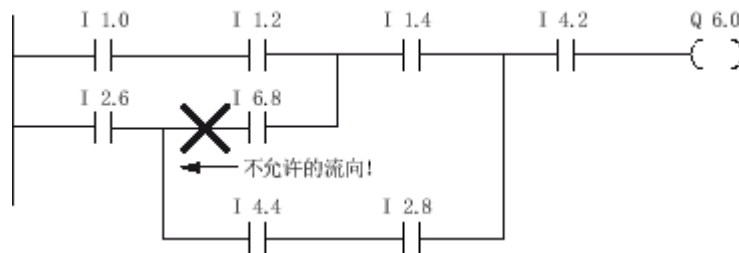
LAD 中禁止的互连 (页 417)

LAD 元素概述 (页 410)

7.2.4.2 LAD 中禁止的互连

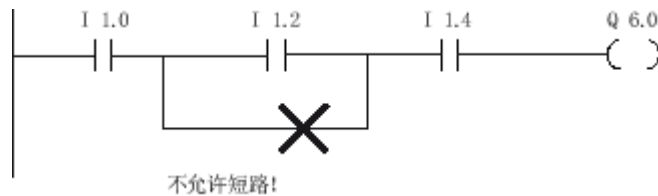
能流从右到左

不能创建可能导致反向能流的分支。



短路

不能创建可能导致短路的分支。



逻辑运算

下列规则适用于逻辑运算：

- 前导逻辑运算中只能使用布尔输入。
- 只有第一个布尔输出还可与一个逻辑运算组合。
- 每个程序段仅允许有一条完整的逻辑路径。 可链接未连接的路径。

参见

LAD 元素的使用规则 (页 416)

7.2.4.2 使用“指令”任务卡插入 LAD 元素

要求

有一个可用的程序段。

步骤

要使用“指令”(Instructions) 任务卡将 LAD 元素插入程序段，请按以下步骤操作：

1. 打开“指令”(Instructions) 任务卡。
2. 浏览到要插入的 LAD 元素。
3. 通过拖放操作将元素移动到程序段中的所需位置。

如果该元素是系统内的功能块 (FB)，则该功能块的数据必须存储在数据块中。因此将打开“调用选项”(Call options) 对话框，可以在其中创建相应的数据块。

或

1. 在程序段中选择要插入元素的位置。
2. 打开“指令”(Instructions) 任务卡。
3. 双击要插入的元素。

如果该元素是系统内的功能块 (FB)，则该功能块的数据必须存储在数据块中。因此将打开“调用选项”(Call options) 对话框，可以在其中创建相应的数据块。

结果

所选的 LAD 元素与参数的占位符一同被插入。

参见

使用空功能框插入 LAD 元素 (页 418)

使用收藏夹插入 LAD 元素 (页 420)

7.2.4.2 使用空功能框插入 LAD 元素

要求

有一个可用的程序段。

步骤

要使用空功能框将 LAD 元素插入程序段，请按以下步骤操作：

1. 打开“指令”(Instructions) 任务卡。
2. 浏览到“常规 > 空功能框”(General > Empty box)。
3. 通过拖放操作将“空功能框”元素移动到程序段中的所需位置。
4. 将光标悬停在该空功能框右上角的黄色三角形上。
将显示一个下拉列表。
5. 从该下拉列表中选择所需的 LAD 元素。

结果

该空功能框变为相应的 LAD 元素。插入了参数的占位符。

参见

使用“指令”任务卡插入 LAD 元素 (页 417)

使用收藏夹插入 LAD 元素 (页 420)

7.2.4.2 选择 LAD 元素的数据类型

7.2.4.2 选择数据类型

简介

有些运算可使用几种不同的数据类型。如果在程序中使用一个这样的运算，则必须为程序中特定位置的运算指定有效的数据类型。必须为某些运算显式选择输入和输出的数据类型。

注意

使能输入 EN 和使能输出 ENO 上变量的有效数据类型 (BOOL) 由系统预定义，无法更改。

对某个运算有效的数据类型列在该运算的下拉列表中。可从下拉列表中选择相应条目来指定运算的数据类型。如果指定变量的数据类型与运算的数据类型不符，则变量名称以红色显示。

7.2.4.2 指定运算的数据类型

简介

有些运算可使用几种不同的数据类型。如果一定要在程序中插入此类运算，则必须在程序中的特定位置为这些运算指定数据类型。

通过下拉列表指定数据类型

要通过下拉列表指定运算的数据类型，请按以下步骤操作：

1. 通过拖放操作在程序中要求的位置插入运算。
将在程序中要求的位置插入运算。在下拉列表中会显示条目“???”(未定义)。
2. 单击下拉列表上方的黄色三角形。
下拉列表将打开，并显示对于该运算有效的数据类型。
3. 从下拉列表中选择一种数据类型。
将显示所选的数据类型。
4. 如果运算有两个下拉列表，则在左下拉列表中选择运算输入的数据类型，在右下拉列表中选择运算输出的数据类型。

通过分配变量指定数据类型

要通过分配变量来指定运算的数据类型，请按以下步骤操作：

1. 通过拖放操作在程序中要求的位置插入运算。
将在程序中要求的位置插入运算。在下拉列表中会显示条目“???”(未定义)。
2. 在输入或输出上，指定有效变量，以便将其数据类型作为运算的数据类型。

所选的数据类型将显示在下拉列表中。

3. 如果必须为运算的输入和输出指定数据类型，则在输入和输出上各输入一个有效变量。在输入上指定的变量确定运算输入的数据类型，在输出上指定的变量确定运算输出的数据类型。

7.2.4.2 在 LAD 中使用收藏夹

7.2.4.2 将元素添加到收藏夹

要求

块已经打开。

步骤

要将元素添加到收藏夹，请按以下步骤操作：

1. 打开“指令”(Instructions) 任务卡。
2. 将“指令”(Instructions) 窗格最大化。
3. 在“指令”(Instructions) 窗格中浏览到要添加到收藏夹的元素。
4. 将该元素拖动到程序编辑器中的“收藏夹”(Favorites) 窗格或“收藏夹”(Favorites) 区域。

参见

使用收藏夹插入 FBD 元素 (页 420)

7.2.4.2 使用收藏夹插入 LAD 元素

要求

- 有一个可用的程序段。
- 有可用的收藏夹。

步骤

要使用收藏夹将元素插入程序段，请按以下步骤操作：

1. 如果在程序编辑器中未显示收藏夹，则单击程序编辑器工具栏中的“也在编辑器中显示收藏夹”(Show favorites also in the editor) 按钮。

收藏夹将显示在块接口下。

2. 将所需的元素从收藏夹拖动到程序段。

或

1. 在程序段中选择要插入元素的位置。
2. 在收藏夹中，单击要插入的元素。

参见

将元素添加到收藏夹 (页 420)
使用“指令”任务卡插入 FBD 元素 (页 447)
使用空功能框插入 FBD 元素 (页 448)

7.2.4.2 从收藏夹中删除元素

先决条件

代码块已打开。

步骤

要从收藏夹中删除元素，请按以下步骤操作：

1. 在“指令”任务卡中打开“收藏夹”(Favorites) 窗格或在程序编辑器的工具栏中，单击“也在编辑器中显示收藏夹”(Show favorites also in the editor) 按钮来显示收藏夹。
2. 右键单击要删除的元素。
3. 在快捷菜单中选择“删除指令”(Remove instruction) 命令。

7.2.4.2 在 LAD 中插入块调用

7.2.4.2 使用拖放操作插入块调用

可以从项目树中通过拖放操作插入现有功能 (FC) 和功能块 (FB) 的调用。如果从其它功能块调用功能块，则可以单个背景或多重背景的方式调用这些功能块。

要求

- 有一个可用的程序段。
- 要被调用的块可用。

插入功能 (FC) 的调用

要使用拖放操作在程序段中插入功能 (FC) 的调用，请按以下步骤操作：

1. 将功能从项目树拖到程序段中。

插入单个背景方式的功能块 (FB) 调用

可以单个背景的方式调用功能块。在这种情况下，被调用的功能块将其数据存储在本身的块中。

另请参见：[单个背景 \(页 324\)](#)

要使用拖放操作在程序段中插入单个背景方式的功能块 (FB) 调用，请按以下步骤操作：

1. 将功能块从项目树拖到程序段中。

将打开“调用选项”(Call options) 对话框。

2. 单击“单个背景”(Single instance) 按钮。
3. 输入要分配给该功能块的数据块的名称。
4. 单击“确定”(OK) 确认输入。

插入多重背景方式的功能块 (FB) 调用

可以多重背景的方式调用功能块。在这种情况下，被调用的功能块将其数据存储于调用功能块的背景数据块中。

另请参见：[多重背景 \(页 324\)](#)

要使用拖放操作在程序段中插入多重背景方式的功能块 (FB) 调用，请按以下步骤操作：

1. 将功能块从项目树拖到程序段中。

将打开“调用选项”(Call options) 对话框。

2. 单击“多重背景”(Multi instance) 按钮。
3. 在“接口中的名称”(Name in the interface) 文本块中，输入变量的名称，通过该名称将被调用功能块作为调用块接口中的静态变量输入。
4. 单击“确定”(OK) 确认输入。

结果

功能或功能块及其参数一同被插入。然后可以分配这些参数。

另请参见：[块调用时的参数传递 \(页 326\)](#)

注意

如果调用功能块时指定了不存在的背景数据块，则将创建该背景数据块。如果以多重背景方式调用了功能块，则会将该功能块作为接口中的静态变量输入。

7.2.4.2 调用所提供的全局库中的功能和功能块

在用户程序中，可调用所提供的全局库中包含的预编程的功能和功能块。

要求

已显示“库”(Libraries) 任务卡。

调用所提供的全局库中的功能

要调用所提供的全局库中的功能 (FC)，请按以下步骤执行：

1. 将“全局库”(Global libraries) 窗格最大化。
2. 将所需元素所在的库最大化。

3. 将功能拖到程序段中。

或者：

1. 将“全局库”(Global libraries) 窗格最大化。
2. 将所需元素所在的库最大化。
3. 打开元素视图。
4. 将功能从“元素”(Elements) 窗格拖到程序段中。

调用所提供的全局库中的功能块

要调用所提供的全局库中的功能块 (FB)，请按以下步骤执行：

1. 将“全局库”(Global libraries) 窗格最大化。
2. 将所需元素所在的库最大化。
3. 使用拖放操作将功能块移动到程序段中。
将打开“调用选项”(Call options) 对话框。
4. 选择单个或多重背景并输入数据块的名称。
5. 单击“确定”(OK) 确认选择。

或者：

1. 将“全局库”(Global libraries) 窗格最大化。
2. 将所需元素所在的库最大化。
3. 打开元素视图。
4. 将功能块从“元素”(Elements) 窗格拖到程序段中。
将打开“调用选项”(Call options) 对话框。
5. 选择单个或多重背景并输入数据块的名称。
6. 单击“确定”(OK) 确认选择。

参见

块调用时的参数传递 (页 326)

7.2.4.2 更新块调用

如果更改被调用块的接口参数，则无法再正确执行块调用。可通过更新块调用来避免此类不一致的块调用。

用户可选择两种方式来更新块调用：

- 在程序编辑器中显式更新。
将更新打开块中的块调用。
- 在编译期间隐式更新。
将更新程序中的所有块调用以及所用的 PLC 数据类型。

在程序编辑器中更新块

要更新块中的块调用，请按以下步骤操作：

1. 在程序编辑器中打开块
2. 单击工具栏中的“更新不一致的块调用”(Update inconsistent block calls)。

或者：

1. 在程序编辑器中打开块。
2. 右键单击包含块调用的指令。
3. 在快捷菜单中选择“更新块调用”(Update block call) 命令。

将打开“接口更新”(Interface update) 对话框。此对话框显示使用中的块接口与被调用块的已更改接口之间的差别。

4. 如果要更新块调用，请单击“确定”(OK)。要取消更新，请单击“取消”(Cancel)。

在编译期间更新块调用

按如下操作在编译期间隐式更新所有块调用和 PLC 数据类型的使用：

1. 打开项目树。
2. 选择“程序块”(Program blocks) 文件夹。
3. 在快捷菜单中选择“编译”(Compile) > “软件 (重建所有块)”(Software (rebuild all blocks)) 命令。

7.2.4.2 更正调用类型

调用类型

有两种调用功能块的方法：

- 以单个背景的方式
- 以多重背景的方式

另请参见：[以单个背景或多重背景的方式调用功能块 \(页 324\)](#)

可以随时修改定义的调用类型。

要求

程序段包含块调用。

步骤

要更改功能块的调用类型，请按以下步骤操作：

1. 选择块调用。
2. 在快捷菜单中选择“更改调用类型”(Change call type) 命令。

将打开“调用选项”(Call options) 对话框。

3. 单击“单个背景”(Single instance) 或“多重背景”(Multi instance) 按钮。

- 如果选择“单个背景”(Single instance) 调用类型，则输入将分配给功能块的数据块的名称。
 - 如果选择“多重背景”(Multi instance) 调用类型，则在“接口中的名称”(Name in the interface) 文本框中输入变量的名称，通过该名称将被调用功能块作为调用块接口中的静态变量输入。
4. 单击“确定”(OK) 确认输入。

7.2.4.2 更改 LAD 元素

7.2.4.2 替换 LAD 元素

可以很容易地将 LAD 元素与同类型的其它 LAD 元素互换。其优点是可以保留元素的参数，无需重新输入。例如，可交换常开触点和常闭触点，RS 触发器和 SR 触发器或定时器。

要求

存在至少含有一个 LAD 元素的程序段。

步骤

要将一个 LAD 元素替换为另一个 LAD 元素，请按以下步骤操作：

1. 选择要替换的 LAD 元素。
2. 将光标悬停在该 LAD 元素右上角的黄色三角形上。
将显示一个下拉列表。
3. 从该下拉列表中，选择要用来替换现有 LAD 元素的 LAD 元素。

7.2.4.2 编辑 LAD 元素

可以使用以下编辑命令编辑 LAD 元素：

- 复制
- 剪切
- 粘贴
- 删除

通常可以通过键盘、菜单和快捷菜单来执行所有命令。

要求

有一个可用的 LAD 元素。

复制

要复制 LAD 元素，请按以下步骤操作：

1. 右键单击要复制的 LAD 元素。

2. 在快捷菜单中选择“复制”(Copy)。

剪切

要剪切 LAD 元素，请按以下步骤操作：

1. 右键单击要剪切的 LAD 元素。
2. 在快捷菜单中选择“剪切”(Cut)。

插入

要粘贴 LAD 元素，请按以下步骤操作：

1. 复制一个 LAD 元素或剪切一个 LAD 元素。
2. 右键单击程序段中想要粘贴该元素的位置。
3. 在快捷菜单中选择“粘贴”(Paste)。

删除

要删除 LAD 元素，请按以下步骤操作：

1. 右键单击要删除的 LAD 元素。
2. 在快捷菜单中选择“删除”(Delete) 命令。

7.2.4.2 插入附加输入和输出

简介

可以使用执行交换性算术运算的附加输入扩展 LAD 元素。例如，这些元素有“加法”(ADD) 和“乘法”(MUL) 指令。插入的输入数没有限制。

只能为 LAD 元素“传送值”(MOVE) 插入附加输出。

要求

存在允许插入附加输入和输出的 LAD 元素。

插入附加输入

要将附加输入添加到 LAD 元素的功能框中，请按以下步骤操作：

1. 右键单击 LAD 元素的现有输入。
2. 在快捷菜单中选择“插入输入”(Insert input)。

附加输入将添加到 LAD 元素的功能框中。

插入附加输出

要将附加输出添加到 LAD 元素的功能框中，请按以下步骤操作：

1. 右键单击 LAD 元素的现有输出。
2. 从快捷菜单中选择“插入输出”(Insert output)。

附加输出将添加到 LAD 元素的功能框中。

参见

插入 LAD 元素 (页 417)

7.2.4.2 删除输入和输出

简介

对于已经使用附加输入扩展的 LAD 元素，随后可以删除这些输入。如果已将输出添加到 LAD 元素“复制值”(MOVE) 中，则也可删除这些输出。

要求

存在已添加附加输入和输出的 LAD 元素。

删除输入

要删除输入，请按下列步骤操作：

1. 选择要删除的输入。
2. 在快捷菜单中选择“删除”(Delete) 命令。

将删除 LAD 元素的该输入。

删除输出

要删除 LAD 元素“复制值”的输出，请按以下步骤操作：

1. 选择要删除的输出。
2. 在快捷菜单中选择“删除”(Delete) 命令。

将删除 LAD 元素“复制值”的该输出。

7.2.4.2 将操作数插入 LAD 指令

7.2.4.2 插入操作数

插入 LAD 元素时，将插入一串问号“???”和“...”作为参数的占位符。以红色显示的“???”字符串表示必须连接的参数。以黑色显示的字符串“...”表示可以连接的参数。

注意

将光标悬停在占位符上可显示可使用的数据类型。

要求

有一个可用的 LAD 元素。

步骤

要连接 LAD 元素的参数，请按以下步骤操作：

1. 双击该参数的占位符。
将打开一个输入域，并且已选中该占位符。
2. 输入适当的参数。

注意

如果输入了已经定义的参数的绝对地址，则只要确认输入，该绝对地址就会变成相应参数的符号名称。如果尚未定义该参数，则将在 PLC 变量表中输入具有该绝对地址和默认名称“Tag_1”的新变量。确认输入时，该绝对地址将替换为符号名称“Tag_1”。

3. 按 Enter 键确认该参数。
4. 如果尚未定义该参数，则可以在程序编辑器中使用快捷菜单直接定义。
另请参见：[在程序编辑器中定义 PLC 变量 \(页 367\)](#)

或者使用操作数选择项：

1. 双击该参数的占位符。
2. 要打开操作数选择项，请单击文本块旁边的符号。
3. 输入参数的首字母。
将显示以该字母开头的所有已定义参数的列表。
4. 从列表中选择相关参数，并按 Enter 键确认选择。

另请参见：[使用操作数选择项 \(页 429\)](#)

或从 PLC 变量表中拖出：

1. 在项目树中，选择“PLC 变量”(PLC tags) 文件夹或打开 PLC 变量表。
2. 如果已打开 PLC 变量表，则将符号从所需变量的第一列拖到程序中的适当位置。如果尚未打开 PLC 变量表，则现在打开详细视图。将所需的变量从详细视图拖到程序中的相应位置。

或从块接口中拖出：

1. 打开块接口。
2. 将所需的操作数从块接口拖到指令窗口。

结果

- 如果语法没有错误，则显示的参数为黑色。编辑器随后会跳转到下一个占位符。
- 如果有语法错误，则光标将停留在输入域中并且在状态行中显示相应的错误消息。再次按 Enter 键后，该输入域将关闭，且该错误输入将以红色斜体显示。

参见

[符号编程 \(页 333\)](#)

7.2.4.2 使用操作数选择项

操作数选择项是一个下拉列表，显示所选指令的所有可用操作数。它包含适合的数据类型的所有全局变量和局部变量。

要求

启用操作数选择项。

另请参见：[PLC 编程的常规设置 \(页 398\)](#)

步骤

要从操作数选择项中选择操作数，请按以下步骤操作：

1. 选择指令。
2. 按 <Return> 键。
操作数的输入域将打开。操作数选择项图标出现在该输入域旁。
3. 单击该操作数选择项图标。
操作数选择项打开。其中包含全局和局部变量、数据块和多重背景。
4. 如有必要，可过滤该列表：
 - 输入一个或几个字母可以仅显示以这些字母开头的操作数。
 - 在第一个位置处输入 # 可以仅显示块接口中的局部操作数。
 - 在第一个位置处输入 " 可以显示 PLC 变量表和现有数据块中的全局操作数。
5. 选择一个操作数。
如果所选的操作数是结构变量、数据块或多重背景，则会在较高级的元素后自动插入句点。第一个操作数的较低级别元素随后将显示在该列表中。
6. 选择该较低级别的元素。

注意

即使在常规设置中禁用了操作数选择项，也可以在必要时显示操作数选择项列表。可使用 <Ctrl+I> 键盘快捷键执行此操作。

参见

[插入和编辑操作数 \(页 457\)](#)

[访问 ARRAY \(页 430\)](#)

[访问 STRUCT 的数据 \(页 430\)](#)

[访问多重背景的数据 \(页 430\)](#)

7.2.4.2 访问 ARRAY

访问 ARRAY 中的数据

可以通过 ARRAY 中特定元素的下标访问 ARRAY 中的数据。下标由方括号括起来的任意整数值 (-32768 至 32767) 组成，例如，Motor.Waerme_1x3[2]。

将 ARRAY 用作块参数

可以将 ARRAY 作为参数传送。如果块具有 ARRAY 类型的输入参数，则必须传送具有相同结构的 ARRAY 作为实参。也可以传送 ARRAY 的个别元素作为实参，但这些元素必须与形参的数据类型一致。

7.2.4.2 访问 STRUCT 的数据

访问结构中的数据

可以通过“结构名称.元素名称”的形式访问结构中的各个元素。

访问结构的实例

可按如下方式访问“Stack_1”结构中的“Temperature”元素：

“Stack_1.Temperature”

将结构用作块参数

可以将结构作为参数传送。如果块具有 STRUCT 类型的输入参数，则必须传送具有相同结构的 STRUCT 作为实参。也可以传送 STRUCT 的个别元素作为实参，但这些元素必须与形参的数据类型一致。

7.2.4.2 访问多重背景的数据

访问多重背景数据

可以在多重背景块内或从调用块中访问多重背景块的数据。

- 在多重背景块内可以寻址块接口中的数据，与寻址所有其它数据的方式相同。
例如，使用“#On”访问多重背景块的输入参数“On”。
- 在调用块中使用“多重背景名称.变量名”访问多重背景块的数据。
例如，使用“Multi.On”访问多重背景块“Multi”的输入参数“On”。

7.2.4.2 通过 LAD 访问 I/O

基本知识

CPU 的过程映像是每个周期更新一次。但是，在对时间要求苛刻的应用中，读取或传送数字输入或输出当前状态的频率可能必须高于每个周期一次。为此，可以在操作数中使用 I/O 访问 ID 的后缀直接访问 I/O。

立即读

如果要直接从 I/O 读取输入，请使用外围设备输入存储区 (PI)，而不是使用输入 (I) 的过程映像。可以位、字节、字或双字形式读取 I/O 存储区。

要直接从输入上读取信号，可以给操作数添加 I/O 访问后缀“:P”，例如“Switch:P”。

立即写

要对输出执行立即写操作，请使用外围设备输出 (PQ) 存储区，而不是使用输出 (Q) 的过程映像。可以位、字节、字或双字形式对外围输出存储区执行写操作。

要将信号直接传送到外围设备输出，可以给操作数添加 I/O 访问后缀“:P”，例如“Motor1:P”。



警告

请注意，对 I/O 执行立即写操作可能会导致出现危险状态，例如，在一个程序循环中对一个输出执行多次写操作。

参见

I/O 数据区 (页 297)
过程映像输入/输出 (页 295)

7.2.4.2 连接隐藏参数

简介

根据所用的 CPU，可以在程序中使用通过 TIA Portal 分派的复杂指令。这些指令可以包含已声明为隐藏的参数。

如果某个指令包含隐藏参数，则指令框的下边缘上会有一个小箭头。可以通过白色字体来识别隐藏参数。

您可以随时显示和连接隐藏参数。

显示或隐藏隐藏参数

要显示或隐藏隐藏参数，请按以下步骤操作：

1. 要显示隐藏参数，请单击指令框下边缘上的向下小箭头。
2. 要隐藏隐藏参数，请单击指令框下边缘上的向上小箭头。

连接隐藏参数

要连接参数，请按以下步骤操作：

1. 如同连接可见参数那样连接隐藏参数。
隐藏参数随即会变成可见参数。

参见

将操作数插入 FBD 指令 (页 457)

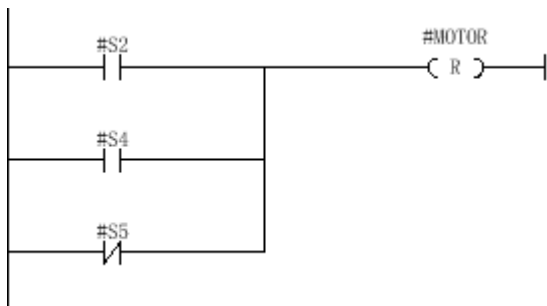
7.2.4.2 LAD 中的分支

7.2.4.2 有关 LAD 中分支的基本信息

定义

使用梯形逻辑 (LAD) 编程语言时，可以使用分支来设计并联电路。分支插在主梯级中。可以将多个触点插入分支中，从而实现串联的并联电路。这样便能设计复杂的梯形逻辑。

下图显示了一个使用分支的实例：



满足下列条件之一时，MOTOR 上为信号 1：

- S2 或 S4 上为信号 1
- S5 上为信号 0。

参见

LAD 中分支的规则 (页 432)

7.2.4.2 LAD 中分支的规则

规则

下列规则适用于并行分支：

- 并行分支向下打开或直接连接到电源线。并行分支向上终止。

- 并行分支在所选 LAD 元素之后打开。
- 并行分支在所选 LAD 元素之后终止。
- 要删除并行分支，必须删除该分支中的所有 LAD 元素。从分支中删除最后一个 LAD 元素时，还会删除该分支的剩余部分。

参见

有关 LAD 中分支的基本信息 (页 432)

在 LAD 程序段中插入分支 (页 433)

在 LAD 程序段中删除分支 (页 434)

在 LAD 程序段中关闭分支 (页 433)

7.2.4.2 在 LAD 程序段中插入分支

可以在程序段中创建多个分支。

要求

- 有一个可用的程序段。
- 程序段包含元素。

步骤

要在程序段中插入新分支，请按以下步骤操作：

1. 打开“指令”(Instructions) 任务卡。
2. 在“指令”(Instructions) 窗格中浏览到“常规 > 打开分支”(General > Open branches)。
3. 通过拖放操作将元素移动到程序段中的所需位置。

如果要在新分支直接连接到电源线，则将该元素拖动到电源线。

参见

在 LAD 程序段中关闭分支 (页 433)

LAD 中分支的规则 (页 432)

7.2.4.2 在 LAD 程序段中关闭分支

分支必须在合适的位置再次关闭。关闭分支后，将添加一些必需的空元素。必要时，可对分支进行整理，以便这些分支不会相互交叉。

要求

有一个可用的分支。

步骤

要关闭打开的分支，请按以下步骤操作：

1. 选择打开的分支。

2. 按住鼠标左键。
移动光标时即会显示一条虚线。
3. 将虚线拖至程序段中的合适位置。可用的连接会通过绿线进行指示。
4. 释放鼠标左键。

参见

在 LAD 程序段中插入分支 (页 433)
在 LAD 程序段中删除分支 (页 434)
LAD 中分支的规则 (页 432)

7.2.4.2 在 LAD 程序段中删除分支

要求

有一个可用的已闭合分支。

步骤

要断开闭合分支，请按以下步骤操作：

1. 选择将分支链接到主分支的连线。
2. 在快捷菜单中选择“删除”(Delete) 命令。

参见

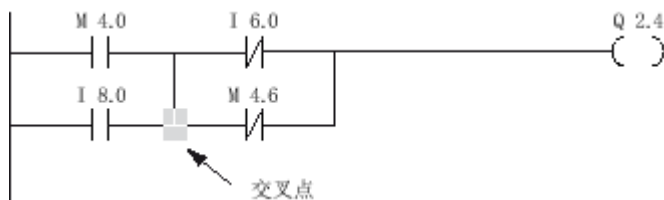
在 LAD 程序段中关闭分支 (页 433)
LAD 中分支的规则 (页 432)

7.2.4.2 LAD 中的交叉点

7.2.4.2 简介

定义

交叉点是指 LAD 程序段中一个分支被关闭而同时另一个分支被打开的位置。



满足以下两个条件时，Q 2.4 为信号 1：

- M4.0 或 I8.0 上为信号 1

- I6.0 或 M4.6 上为信号 0。

参见

插入交叉点 (页 435)
重新安排交叉点 (页 435)
删除交叉点 (页 436)

7.2.4.2 插入交叉点

可通过在主分支和附加分支之间或两个不同分支之间创建连接，在 LAD 程序段中插入交叉点。

要求

有一个可用的分支。

步骤

1. 打开“指令”(Instructions) 任务卡。
2. 在“指令”(Instructions) 窗格中浏览到“常规 > 打开分支”(General > Open branch)。
3. 使用拖放操作将元素移动到新分支的后面。
4. 在打开的分支中插入任意元素。
5. 单击所插入元素后的打开分支的行。
6. 按住鼠标左键并将虚连接线拖动到主分支。
7. 释放鼠标左键。

参见

简介 (页 434)
在 LAD 程序段中插入分支 (页 433)

7.2.4.2 重新安排交叉点

要求

有一个可用的交叉点。

重新安排交叉点

要重新安排连接，请按以下步骤操作：

1. 选择相应分支中定义交叉点的连线。
2. 在快捷菜单中选择“删除”(Delete) 命令。
3. 打开“指令”(Instructions) 任务卡。
4. 在“指令”(Instructions) 窗格中浏览到“常规 > 打开分支”(General > Open branch)。
5. 使用拖放操作将元素移动到程序段中要插入新交叉点的位置。

6. 在打开的分支中插入任意元素。
7. 单击所插入元素后的打开分支的行。
8. 按住鼠标左键并将虚连接线拖动到主分支。
9. 释放鼠标左键。

参见

简介 (页 434)

7.2.4.2 删除交叉点

要求

有一个可用的交叉点。

步骤

要删除交叉点，请按以下步骤操作：

1. 选择相应分支中定义交叉点的连线。
2. 在快捷菜单中选择“删除”(Delete) 命令。

参见

简介 (页 434)

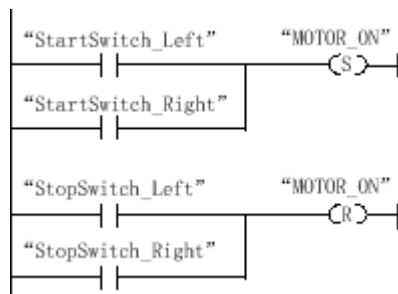
7.2.4.2 LAD 中的梯级

7.2.4.2 简介

使用梯级

程序以一个或多个程序段表示。在程序段的左边和右边都有一根引出一个或多个梯级的电源线。二进制信号以触点的形式排列在梯级上。元素在分支上的顺序排列构成串联，并行分支上的排列构成并联。梯级以写入逻辑运算结果的线圈结束。

下图显示了在一个程序段中使用多个梯级的实例：



规则

使用多个梯级时，请记住下列规则：

- 梯级间不允许存在连接。
- 每个程序段只允许有一个跳转指令。跳转指令的定位规则仍然有效。

运行梯级

梯级和程序段从上到下、从左到右执行。这意味着首先处理第一个程序段第一个梯级中的第一条指令。然后处理该梯级的所有其它指令。接着处理第一个程序段的所有其它梯级。只有先运行完所有梯级后，才处理下一个程序段。

分支和梯级之间的区别

分支和梯级之间的区别是：梯级是独立分支，还可以出现在其它程序段中。另一方面，通过分支可设计并联结构。

参见

插入梯级 (页 437)

删除梯级 (页 438)

7.2.4.2 插入梯级

要求

- 块已经打开。
- 有一个可用的程序段。

步骤

要在程序段中插入新梯级，请按以下步骤操作：

1. 在左侧电源线上插入任意线圈。
将插入新的梯级，且线圈放置在该梯级的末尾。
2. 在新梯级中插入其它指令。

参见

简介 (页 436)

删除梯级 (页 438)

7.2.4.2 删除梯级

要求

有一个可用的梯级。

步骤

要删除梯级，请按以下步骤操作：

1. 按住鼠标左键，框选该梯级。同时确保选中所有指令。还可以按住 <Shift> 键并选择该梯级的第一条和最后一条指令。
2. 右键单击该梯级中的一条指令。
3. 在快捷菜单中选择“删除”(Delete) 命令。

参见

简介 (页 436)

插入梯级 (页 437)

7.2.4.2 使用自由格式的注释

7.2.4.2 有关在 LAD 中使用自由格式的注释的基本信息

简介

使用自由格式的注释可在图形编程语言的源代码中添加注释，其与文本语言的行注释类似。

自由格式的注释可用于以下元素：

- 功能框
- 线圈

7.2.4.2 插入自由格式的注释

步骤

要插入自由格式的注释，请按以下步骤操作：

1. 右键单击要为其插入自由格式的注释的对象。
2. 在快捷菜单中选择“插入注释”(Insert comment) 命令。

将打开一个含有标准注释的注释框。该注释框通过箭头连接到相应的对象。

7.2.4.2 编辑自由格式的注释

简介

可以如下所述编辑自由格式的注释：

- 更改注释文本
- 更改注释框的位置或大小
- 将注释附加到其它元素
- 显示和隐藏自由格式的注释

更改注释文本

要更改自由格式注释的文本，请按以下步骤操作：

1. 在注释框中双击。
2. 输入所需的文本。

更改注释框的位置

要更改注释框的位置，请按下列步骤操作：

1. 左键单击注释框，并按住鼠标按键。
2. 将注释框拖到所需位置。

更改注释框的大小

要更改注释框的大小，请按下列步骤操作：

1. 单击注释框的边。
2. 拖动注释框右下角的移动滑块直至注释框达到所需的大小。

将注释附加到其它元素

要将自由格式的注释附加到其它元素，请按以下步骤操作：

1. 左键单击链接注释框和指令的箭头的尖端，并按住鼠标按键。
2. 将箭头拖动到要附加该注释的元素。可能的插入点以绿色正方形标记。
3. 释放鼠标按键。

显示和隐藏自由格式的注释

要显示或隐藏自由格式的注释，请按以下步骤操作：

1. 在工具栏中单击“显示/隐藏自由格式的注释”(Show / hide free comments)。

7.2.4.2 删除自由格式的注释

步骤

要删除自由格式的注释，请按以下步骤操作：

1. 右键单击要删除的自由格式的注释。
2. 在快捷菜单中选择“删除”(Delete) 命令。

7.2.4.2 创建 FBD 程序

7.2.4.2 有关 FBD 的基本信息

7.2.4.2 FBD 编程语言

功能块图 (FBD) 编程语言概述

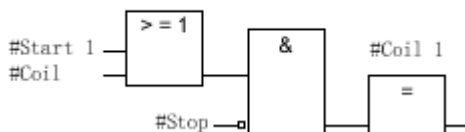
FBD 是一种图形编程语言。它采用基于电路系统的表示法。

程序以一个或多个程序段表示。程序段包含一个或多个逻辑运算路径。二进制信号通过功能框来链接。逻辑表示法以布尔代数中使用的图形逻辑符号为基础。

FBD 程序段实例

下图所示为一个含有 AND 和 OR 功能框以及赋值操作的 FBD 程序段：

程序段 1



参见

使用程序段 (页 412)

7.2.4.2 FBD 元素概述

FBD 元素

FBD 程序由通过二进制信号流链接的各个元素组成。必须给多数程序元素提供变量。对于定时器和计数器运算，还必须创建数据块以存储形参。

从左到右对 FBD 程序段进行编程。

例如，下图所示为一个 FBD 程序段的元素：

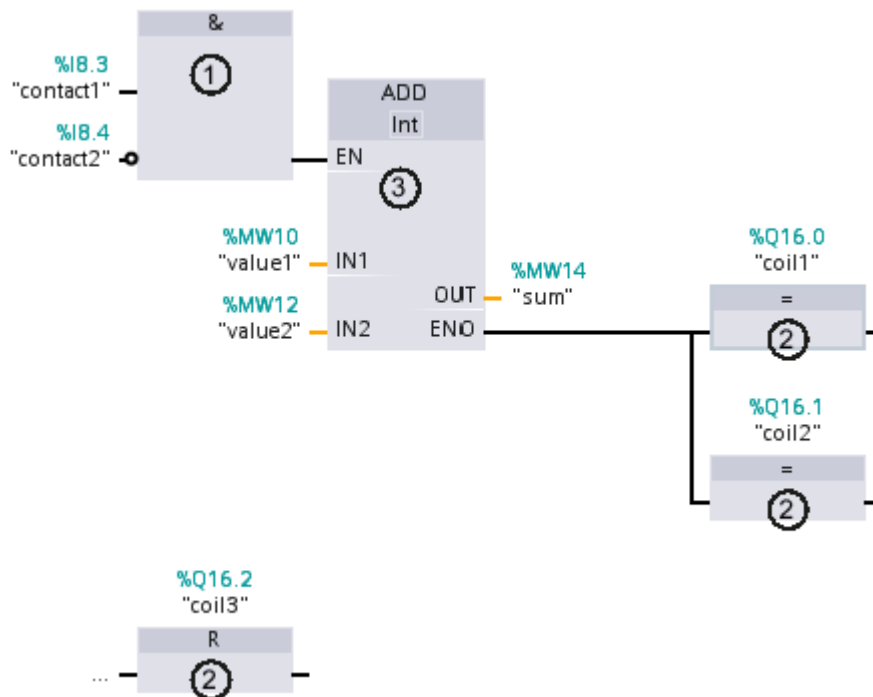


图 7-2

- 1) 二进制功能
- 2) 标准功能框
- 3) 复杂功能框

二进制功能

可以使用二进制功能查询二进制操作数和组合这些操作数的信号状态。下列运算是二进制功能的实例：“AND 运算”、“OR 运算”和“XOR 运算”。

标准功能框：

可以在程序中使用标准功能框控制二进制操作数，执行 RLO 沿检测或执行跳转功能。标准功能框通常只有一个输入。

复杂功能框

复杂功能框代表具有复杂功能的程序元素。但空功能框除外。可以使用空功能框作为占位符，在其中可以选择所需的运算。

在 FBD 程序中可以使用以下类型的功能框：

- 无 EN/ENO 机制的复杂功能框：
功能框的执行与功能框输入的信号状态无关。无法查询处理过程中的错误状态。
- 具有 EN/ENO 机制的功能框：
只有使能输入“EN”的信号状态为“1”时才执行功能框。如果正确处理了该功能框，则“ENO”使能输出的信号状态为“1”。如果处理期间出错，则复位“ENO”输出。

如果未互连 EN 使能输入，则始终执行功能框。

代码块的调用在程序段中也显示为具有 EN/ENO 机制的复杂功能框。

7.2.4.2 FBD 的设置

7.2.4.2 LAD 的设置概述

概述

下表给出了用户可以进行的设置：

组	设置	描述
字体	大小	程序编辑器中的字体大小
视图	布局	“紧凑”或“宽幅” 更改操作数与其它对象（例如操作数和触点）之间的垂直间距。重新打开块时即可看到所做的更改。
	具有绝对信息	附加显示绝对地址
操作数域	最大宽度	在操作数域中可水平输入的最多字符数
	最大高度	在操作数域中可垂直输入的最多字符数

7.2.4.2 更改设置

步骤

为改变设置，可进行如下操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
工作区中将显示“设置”(Settings) 窗口。
2. 在区域导航中，选择“PLC 编程”(PLC programming) 组。
3. 更改设置。

结果

更改将直接装载，无需显式保存。

7.2.4.2 使用程序段

7.2.4.2 使用程序段

功能

用户程序在块的程序段中创建。对于要编程的代码块，它必须包含至少一个程序段。为获得更佳的用户程序总览效果，还可以将您的程序划分为多个程序段。

参见

插入程序段 (页 413)

选择程序段 (页 413)

复制和插入程序段 (页 414)

删除程序段 (页 414)

展开和折叠程序段 (页 415)

7.2.4.2 插入程序段

要求

块已经打开。

步骤

要插入新程序段，请按以下步骤操作：

1. 选择要在其后插入新程序段的程序段。
2. 在快捷菜单中选择“插入程序段”(Insert network) 命令。

注意

如果在 LAD 或 FBD 程序中将元素插入块的最后一个（且仍为空的）程序段，则新的空程序段会自动在其下方插入。

结果

在块中插入了新的空程序段。

参见

使用程序段 (页 412)

7.2.4.2 选择程序段

要求

有一个可用的程序段。

选择一个程序段

要选择一个程序段，请按以下步骤操作：

1. 单击要选择的程序段的标题栏。

选择多个程序段

要选择多个程序段，请按以下步骤操作：

1. 按住 <Ctrl> 键。
2. 单击要选择的所有程序段。

要选择多个连续的程序段，请按以下步骤操作：

1. 按住 <Shift> 键。
2. 单击要选择的第一个程序段。
3. 单击要选择的最后一个程序段。

将选中第一个和最后一个程序段以及它们之间所有的程序段。

参见

使用程序段 (页 412)

7.2.4.2 复制和插入程序段

复制的程序段可粘贴在同一个块或其它块中。如果将程序段粘贴到用不同编程语言编写的块中，该程序段的编程语言将保持不变。

先决条件

有一个可用的程序段。

步骤

要复制和粘贴程序段，请按以下步骤操作：

1. 选择要复制的一个或多个程序段。
2. 在快捷菜单中选择“复制”(Copy)。
3. 选择要在其后粘贴所复制程序段的程序段。
4. 在快捷菜单中选择“粘贴”(Paste)。

注意

还可以将程序段从一个块复制到另一个块。

参见

使用程序段 (页 412)

7.2.4.2 删除程序段

要求

有一个可用的程序段。

步骤

要删除程序段，请按以下步骤操作：

1. 选择要删除的程序段。
2. 在快捷菜单中选择“删除”(Delete) 命令。

参见

使用程序段 (页 412)

7.2.4.2 展开和折叠程序段

要求

有一个可用的程序段。

打开和关闭程序段

1. 选择要打开或关闭的程序段。
2. 在快捷菜单中选择“展开”(Expand) 或“折叠”(Collapse)。

打开和关闭所有程序段

1. 在工具栏中单击“打开所有程序段”(Open all networks) 或“关闭所有程序段”(Close all networks)。

参见

使用程序段 (页 412)

7.2.4.2 输入程序段标题

程序段标题是程序段的标头。程序段标题的长度无限制，但只能占用一行。

要求

有一个可用的程序段。

步骤

要输入程序段标题，请按以下步骤操作：

1. 双击程序段标题栏中的“.....”。
“.....”文本行将被选中。
2. 输入程序段标题。

7.2.4.2 在 LAD 程序段中输入程序段注释

可以使用程序段注释描述各个程序段。例如，可以指出程序段的功能或提醒用户注意某些特性。

要求

有一个可用的程序段。

步骤

要输入程序段注释，请按以下步骤操作：

1. 单击程序段标题前的小箭头。
右箭头将变为下箭头。
2. 如果注释区域不可见，请单击工具栏中的“启用/禁用程序段注释”(Network comments on/off)。
将显示注释区域。
3. 在注释区域中单击“注释”(Comment)。
“注释”(Comment) 文本行将被选中。
4. 输入程序段注释。

7.2.4.2 插入 FBD 元素

7.2.4.2 FBD 元素的使用规则

规则

插入 FBD 元素时请注意下列规则：

- 一个 FBD 程序段可由多个元素组成。逻辑路径的所有元素必须按 IEC 61131-3 标准相互链接。

- 可以将标准功能框 (触发器、计数器、定时器、数学运算等) 作为输出添加到具有二进制逻辑运算 (例如 AND、OR) 的功能框。该规则不适用于比较功能框。
 - 前导逻辑运算中只能使用布尔输入。
 - 只有底部的布尔输出还可用于另一个逻辑运算。
 - 跳转运算只能放在末尾，RET 还可放在开头。
 - 使能输入“EN”与使能输出“ENO”可连接到功能框，但这不是必需的。
 - 不能给二进制逻辑运算分配常量 (例如，TRUE 或 FALSE)。而是使用 BOOL 数据类型的变量。
- 有关在程序段中的什么位置可以粘贴 FBD 元素的信息，请参见 FBD 参考帮助。

7.2.4.2 使用“指令”任务卡插入 FBD 元素

要求

有一个可用的程序段。

步骤

要使用“指令”任务卡将 FBD 元素插入程序段，请按以下步骤操作：

1. 打开“指令”(Instructions) 任务卡。
2. 浏览到要插入的 FBD 元素。
3. 通过拖放操作将元素移动到程序段中的所需位置。

如果该元素是系统内的功能块 (FB)，则该功能块的数据必须存储在背景数据块中。在这种情况下，将打开“调用选项”(Call options) 对话框，可以在其中创建相应的数据块。

或

1. 在程序段中选择要插入元素的位置。
2. 打开“指令”(Instructions) 任务卡。
3. 双击要插入的元素。

如果该元素是系统内的功能块 (FB)，则该功能块的数据必须存储在背景数据块中。在这种情况下，将打开“调用选项”(Call options) 对话框，可以在其中创建相应的背景数据块。

结果

所选的 FBD 元素与参数占位条目一同被插入。

参见

使用空功能框插入 FBD 元素 (页 448)

使用收藏夹插入 FBD 元素 (页 420)

7.2.4.2 使用空功能框插入 FBD 元素

要求

有一个可用的程序段。

步骤

要使用空功能框将 FBD 元素插入程序段，请按以下步骤操作：

1. 打开“指令”(Instructions) 任务卡。
2. 浏览到“常规 > 空功能框”(General > Empty box)。
3. 通过拖放操作将“空功能框”元素移动到程序段中的所需位置。
4. 将光标悬停在该空功能框右上角的黄色三角形上。
将显示一个下拉列表。
5. 从该下拉列表中选择所需的 FBD 元素。

结果

该空功能框变为相应的 FBD 元素。插入了参数的占位符。

参见

使用“指令”任务卡插入 FBD 元素 (页 447)

使用收藏夹插入 FBD 元素 (页 420)

7.2.4.2 选择 FBD 元素的数据类型

7.2.4.2 选择数据类型

简介

有些运算可使用几种不同的数据类型。如果在程序中使用一个这样的运算，则必须为程序中特定位置的运算指定有效的数据类型。必须为某些运算显式选择输入和输出的数据类型。

注意

使能输入 EN 和使能输出 ENO 上变量的有效数据类型 (BOOL) 由系统预定义，无法更改。

对某个运算有效的数据类型列在该运算的下拉列表中。可从下拉列表中选择相应条目来指定运算的数据类型。如果指定变量的数据类型与运算的数据类型不符，则变量名称以红色显示。

7.2.4.2 指定运算的数据类型

简介

有些运算可使用几种不同的数据类型。如果一定要在程序中插入此类运算，则必须在程序中的特定位置为这些运算指定数据类型。

通过下拉列表指定数据类型

要通过下拉列表指定运算的数据类型，请按以下步骤操作：

1. 通过拖放操作在程序中要求的位置插入运算。
将在程序中要求的位置插入运算。在下拉列表中会显示条目“???”(未定义)。
2. 单击下拉列表上方的黄色三角形。
下拉列表将打开，并显示对于该运算有效的数据类型。
3. 从下拉列表中选择一种数据类型。
将显示所选的数据类型。
4. 如果运算有两个下拉列表，则在左下拉列表中选择运算输入的数据类型，在右下拉列表中选择运算输出的数据类型。

通过分配变量指定数据类型

要通过分配变量来指定运算的数据类型，请按以下步骤操作：

1. 通过拖放操作在程序中要求的位置插入运算。
将在程序中要求的位置插入运算。在下拉列表中会显示条目“???”(未定义)。
2. 在输入或输出上，指定有效变量，以便将其数据类型作为运算的数据类型。
所选的数据类型将显示在下拉列表中。
3. 如果必须为运算的输入和输出指定数据类型，则在输入和输出上各输入一个有效变量。在输入上指定的变量确定运算输入的数据类型，在输出上指定的变量确定运算输出的数据类型。

7.2.4.2 使用收藏夹

7.2.4.2 将元素添加到收藏夹

要求

块已经打开。

步骤

要将元素添加到收藏夹，请按以下步骤操作：

1. 打开“指令”(Instructions) 任务卡。
2. 将“指令”(Instructions) 窗格最大化。

3. 在“指令”(Instructions) 窗格中浏览到要添加到收藏夹的元素。
4. 将该元素拖动到程序编辑器中的“收藏夹”(Favorites) 窗格或“收藏夹”(Favorites) 区域。

参见

使用收藏夹插入 FBD 元素 (页 420)

7.2.4.2 使用收藏夹插入 LAD 元素

要求

- 有一个可用的程序段。
- 有可用的收藏夹。

步骤

要使用收藏夹将元素插入程序段，请按以下步骤操作：

1. 如果在程序编辑器中未显示收藏夹，则单击程序编辑器工具栏中的“也在编辑器中显示收藏夹”(Show favorites also in the editor) 按钮。

收藏夹将显示在块接口下。

2. 将所需的元素从收藏夹拖动到程序段。

或

1. 在程序段中选择要插入元素的位置。
2. 在收藏夹中，单击要插入的元素。

参见

将元素添加到收藏夹 (页 420)

使用“指令”任务卡插入 FBD 元素 (页 447)

使用空功能框插入 FBD 元素 (页 448)

7.2.4.2 从收藏夹中删除元素

先决条件

代码块已打开。

步骤

要从收藏夹中删除元素，请按以下步骤操作：

1. 在“指令”任务卡中打开“收藏夹”(Favorites) 窗格或在程序编辑器的工具栏中，单击“也在编辑器中显示收藏夹”(Show favorites also in the editor) 按钮来显示收藏夹。
2. 右键单击要删除的元素。
3. 在快捷菜单中选择“删除指令”(Remove instruction) 命令。

7.2.4.2 在 FBD 中插入块调用

7.2.4.2 使用拖放操作插入块调用

可以从项目树中通过拖放操作插入现有功能 (FC) 和功能块 (FB) 的调用。如果从其它功能块调用功能块，则可以单个背景或多重背景的方式调用这些功能块。

要求

- 有一个可用的程序段。
- 要被调用的块可用。

插入功能 (FC) 的调用

要使用拖放操作在程序段中插入功能 (FC) 的调用，请按以下步骤操作：

1. 将功能从项目树拖到程序段中。

插入单个背景方式的功能块 (FB) 调用

可以单个背景的方式调用功能块。在这种情况下，被调用的功能块将其数据存储在本身的块中。

另请参见：[单个背景 \(页 324\)](#)

要使用拖放操作在程序段中插入单个背景方式的功能块 (FB) 调用，请按以下步骤操作：

1. 将功能块从项目树拖到程序段中。
将打开“调用选项”(Call options) 对话框。
2. 单击“单个背景”(Single instance) 按钮。
3. 输入要分配给该功能块的数据块的名称。
4. 单击“确定”(OK) 确认输入。

插入多重背景方式的功能块 (FB) 调用

可以多重背景的方式调用功能块。在这种情况下，被调用的功能块将其数据存储在调用功能块的背景数据块中。

另请参见：[多重背景 \(页 324\)](#)

要使用拖放操作在程序段中插入多重背景方式的功能块 (FB) 调用，请按以下步骤操作：

1. 将功能块从项目树拖到程序段中。
将打开“调用选项”(Call options) 对话框。
2. 单击“多重背景”(Multi instance) 按钮。
3. 在“接口中的名称”(Name in the interface) 文本块中，输入变量的名称，通过该名称将被调用功能块作为调用块接口中的静态变量输入。
4. 单击“确定”(OK) 确认输入。

结果

功能或功能块及其参数一同被插入。然后可以分配这些参数。

另请参见：[块调用时的参数传递 \(页 326\)](#)

注意

如果调用功能块时指定了不存在的背景数据块，则将创建该背景数据块。如果以多重背景方式调用了功能块，则会将该功能块作为接口中的静态变量输入。

7.2.4.2 调用所提供的全局库中的功能和功能块

在用户程序中，可调用所提供的全局库中包含的预编程的功能和功能块。

要求

已显示“库”(Libraries) 任务卡。

调用所提供的全局库中的功能

要调用所提供的全局库中的功能 (FC)，请按以下步骤执行：

1. 将“全局库”(Global libraries) 窗格最大化。
2. 将所需元素所在的库最大化。
3. 将功能拖到程序段中。

或者：

1. 将“全局库”(Global libraries) 窗格最大化。
2. 将所需元素所在的库最大化。
3. 打开元素视图。
4. 将功能从“元素”(Elements) 窗格拖到程序段中。

调用所提供的全局库中的功能块

要调用所提供的全局库中的功能块 (FB)，请按以下步骤执行：

1. 将“全局库”(Global libraries) 窗格最大化。
2. 将所需元素所在的库最大化。
3. 使用拖放操作将功能块移动到程序段中。

将打开“调用选项”(Call options) 对话框。

4. 选择单个或多重背景并输入数据块的名称。
5. 单击“确定”(OK) 确认选择。

或者：

1. 将“全局库”(Global libraries) 窗格最大化。
2. 将所需元素所在的库最大化。
3. 打开元素视图。
4. 将功能块从“元素”(Elements) 窗格拖到程序段中。
将打开“调用选项”(Call options) 对话框。
5. 选择单个或多重背景并输入数据块的名称。
6. 单击“确定”(OK) 确认选择。

参见

块调用时的参数传递 (页 326)

7.2.4.2 更新块调用

如果更改被调用块的接口参数，则无法再正确执行块调用。可通过更新块调用来避免此类不一致的块调用。

用户可选择两种方式来更新块调用：

- 在程序编辑器中显式更新。
将更新打开块中的块调用。
- 在编译期间隐式更新。
将更新程序中的所有块调用以及所用的 PLC 数据类型。

在程序编辑器中更新块

要更新块中的块调用，请按以下步骤操作：

1. 在程序编辑器中打开块
2. 单击工具栏中的“更新不一致的块调用”(Update inconsistent block calls)。

或者：

1. 在程序编辑器中打开块。
2. 右键单击包含块调用的指令。
3. 在快捷菜单中选择“更新块调用”(Update block call) 命令。

将打开“接口更新”(Interface update) 对话框。此对话框显示使用中的块接口与被调用块的已更改接口之间的差别。

4. 如果要更新块调用，请单击“确定”(OK)。要取消更新，请单击“取消”(Cancel)。

在编译期间更新块调用

按如下操作在编译期间隐式更新所有块调用和 PLC 数据类型的使用：

1. 打开项目树。

2. 选择“程序块”(Program blocks) 文件夹。
3. 在快捷菜单中选择“编译”(Compile) >“软件 (重建所有块)”(Software (rebuild all blocks)) 命令。

7.2.4.2 更正调用类型

调用类型

有两种调用功能块的方法：

- 以单个背景的方式
- 以多重背景的方式

另请参见：[以单个背景或多重背景的方式调用功能块 \(页 324\)](#)

可以随时修改定义的调用类型。

要求

程序段包含块调用。

步骤

要更改功能块的调用类型，请按以下步骤操作：

1. 选择块调用。
2. 在快捷菜单中选择“更改调用类型”(Change call type) 命令。
将打开“调用选项”(Call options) 对话框。
3. 单击“单个背景”(Single instance) 或“多重背景”(Multi instance) 按钮。
 - 如果选择“单个背景”(Single instance) 调用类型，则输入将分配给功能块的数据块的名称。
 - 如果选择“多重背景”(Multi instance) 调用类型，则在“接口中的名称”(Name in the interface) 文本框中输入变量的名称，通过该名称将被调用功能块作为调用块接口中的静态变量输入。
4. 单击“确定”(OK) 确认输入。

7.2.4.2 更改 FBD 元素

7.2.4.2 插入 FBD 元素

可以很容易地将 FBD 元素与同类型的其它 FBD 元素互换。其优点是可以保留元素的参数，无需重新输入。例如，可以交换 OR 和 AND、RS 触发器和 SR 触发器、比较功能或跳转指令。

要求

存在至少含有一个 FBD 元素的程序段。

步骤

要将一个 FBD 元素替换为另一个 FBD 元素，请按以下步骤操作：

1. 选择要替换的 FBD 元素。
如果存在与所选 FBD 元素兼容的元素，则该元素的右上角将显示一个黄色三角形。
2. 将光标移动到该 FBD 元素的黄色三角形上。
将显示一个下拉列表。
3. 从该下拉列表中，选择要用来替换现有 FBD 元素的 FBD 元素。

7.2.4.2 编辑 FBD 元素

可以使用以下编辑命令编辑 FBD 元素：

- 复制
- 剪切
- 粘贴
- 删除

通常可以通过键盘、菜单和快捷菜单来执行所有命令。

复制

要复制 FBD 元素，请按以下步骤操作：

1. 右键单击要复制的 FBD 元素。
2. 在快捷菜单中选择“复制”(Copy)。

剪切

要剪切 FBD 元素，请按以下步骤操作：

1. 右键单击要剪切的 FBD 元素。
2. 在快捷菜单中选择“剪切”(Cut)。

插入

要粘贴 FBD 元素，请按以下步骤操作：

1. 复制一个 FBD 元素或剪切一个 FBD 元素。
2. 右键单击程序段中想要粘贴该元素的位置。
3. 在快捷菜单中选择“粘贴”(Paste)。

删除

要删除 FBD 元素，请按以下步骤操作：

1. 右键单击要删除的 FBD 元素。
2. 在快捷菜单中选择“删除”(Delete) 命令。

7.2.4.2 粘贴附加输入和输出

简介

可以使用执行算术或二进制运算的附加输入扩展 FBD 元素。例如，这些元素有“加法”(ADD)、“乘法”(MUL)、AND 或 OR 指令。

只能为 FBD 元素“传送值”(MOVE) 插入附加输出。

要求

存在允许插入附加输入和输出的 FBD 元素。

插入附加输入

要将附加输入添加到 FBD 元素的功能框中，请按以下步骤操作：

1. 右键单击 FBD 元素的现有输入。
2. 在快捷菜单中选择“插入输入”(Insert input)。

附加输入将添加到 FBD 元素的功能框中。

插入附加输出

要将附加输出添加到 FBD 元素的功能框中，请按以下步骤操作：

1. 右键单击 FBD 元素的现有输出。
2. 从快捷菜单中选择“插入输出”(Insert output)。

附加输出将添加到 FBD 元素的功能框中。

参见

插入 FBD 元素 (页 447)

7.2.4.2 删除输入和输出

简介

对于已经使用附加输入扩展的 FBD 元素，随后可以删除这些输入。如果已将输出添加到 FBD 元素“传送值”(MOVE) 中，则也可删除这些输出。

要求

存在已使用附加输入或输出扩展的 FBD 元素。

删除输入

要删除输入，请按下列步骤操作：

1. 选择要删除的输入。
2. 在快捷菜单中选择“删除”(Delete) 命令。

将删除 FBD 元素的该输入。

删除输出

要删除 FBD 元素“传送值”的输出，请按以下步骤操作：

1. 选择要删除的输出。
2. 在快捷菜单中选择“删除”(Delete) 命令。

将删除 FBD 元素“传送值”的该输出。

7.2.4.2 将操作数插入 FBD 指令

7.2.4.2 插入和编辑操作数

插入 FBD 元素时，将插入字符串“???”、“??.”和“...”作为参数的占位符。以红色显示的字符串“???”和“??.”表示必须连接的参数。以黑色显示的字符串“...”表示可以连接的参数。“??.”代表布尔占位符。

注意

要在工具提示中显示可用的数据类型，请将光标移动到占位符上。

要求

存在 FBD 元素。

步骤

要连接 FBD 元素的参数，请按以下步骤操作：

1. 单击参数的占位符。
将打开一个输入域。
2. 输入相应的参数，例如 PLC 变量、局部变量或常量。

注意

如果输入了已经定义的参数的绝对地址，则只要确认输入，该绝对地址就会变成相应参数的符号名称。如果尚未定义该参数，则将在 PLC 变量表中输入具有该绝对地址和默认名称“Tag_1”的新变量。确认输入时，该绝对地址将替换为符号名称“Tag_1”。

3. 按 Enter 键确认该参数。
4. 如果尚未定义该参数，则可以在程序编辑器中使用快捷菜单直接定义。

另请参见：“[在程序编辑器中定义 PLC 变量 \(页 367\)](#)”。

或者使用操作数选择项：

1. 单击参数的占位符。
2. 按 <Ctrl + I> 打开操作数选择项。
3. 输入参数的首字母。

将显示一个列表，其中包括所有以该字母开头并具有兼容数据类型的已定义参数。可以输入更多字母以进一步限制选择项。

4. 从列表中选择相关参数，并按 Enter 键确认选择。

另请参见：[使用操作数选择项 \(页 429\)](#)

或从 PLC 变量表中拖出：

1. 在项目树中，选择“PLC 变量”(PLC tags) 文件夹或打开 PLC 变量表。
2. 如果已打开 PLC 变量表，则将所需的变量拖到程序中的相应位置。如果尚未打开 PLC 变量表，则现在打开详细视图。将所需的变量从详细视图拖到程序中的相应位置。

或从块接口中拖出：

1. 打开块接口。
2. 将所需的操作数从块接口拖到程序中的相应位置。

结果

- 如果语法没有错误，则显示的参数为黑色。
- 如果有语法错误，则光标将停留在输入域中，并且在巡视窗口的“信息 > 语法”(Info > Syntax) 寄存器中会显示相应的错误消息。

参见

[符号编程 \(页 333\)](#)

7.2.4.2 使用操作数选择项

操作数选择项是一个下拉列表，显示所选指令的所有可用操作数。它包含适合的数据类型的所有全局变量和局部变量。

要求

启用操作数选择项。

另请参见：[PLC 编程的常规设置 \(页 398\)](#)

步骤

要从操作数选择项中选择操作数，请按以下步骤操作：

1. 选择指令。
2. 按 <Return> 键。

操作数的输入域将打开。操作数选择项图标出现在该输入域旁。

3. 单击该操作数选择项图标。

操作数选择项打开。其中包含全局和局部变量、数据块和多重背景。

4. 如有必要，可过滤该列表：

- 输入一个或几个字母可以仅显示以这些字母开头的操作数。
- 在第一个位置处输入 # 可以仅显示块接口中的局部操作数。
- 在第一个位置处输入 " 可以显示 PLC 变量表和现有数据块中的全局操作数。

5. 选择一个操作数。

如果所选的操作数是结构变量、数据块或多重背景，则会在较高级的元素后自动插入句点。第一个操作数的较低级别元素随后将显示在该列表中。

6. 选择该较低级别的元素。

注意

即使在常规设置中禁用了操作数选择项，也可以在必要时显示操作数选择项列表。可使用 <Ctrl+I> 键盘快捷键执行此操作。

参见

- 插入和编辑操作数 (页 457)*
- 访问 ARRAY (页 430)*
- 访问 STRUCT 的数据 (页 430)*
- 访问多重背景的数据 (页 430)*

7.2.4.2 访问 ARRAY

访问 ARRAY 中的数据

可以通过 ARRAY 中特定元素的下标访问 ARRAY 中的数据。下标由方括号括起来的任意整数值 (-32768 至 32767) 组成，例如，Motor.Waerme_1x3[2]。

将 ARRAY 用作块参数

可以将 ARRAY 作为参数传送。如果块具有 ARRAY 类型的输入参数，则必须传送具有相同结构的 ARRAY 作为实参。也可以传送 ARRAY 的个别元素作为实参，但这些元素必须与形参的数据类型一致。

7.2.4.2 访问 STRUCT 的数据

访问结构中的数据

可以通过“结构名称.元素名称”的形式访问结构中的各个元素。

访问结构的实例

可按如下方式访问“Stack_1”结构中的“Temperature”元素：

“Stack_1.Temperature”

将结构用作块参数

可以将结构作为参数传送。如果块具有 STRUCT 类型的输入参数，则必须传送具有相同结构的 STRUCT 作为实参。也可以传送 STRUCT 的个别元素作为实参，但这些元素必须与形参的数据类型一致。

7.2.4.2 访问多重背景的数据

访问多重背景数据

可以在多重背景块内或从调用块中访问多重背景块的数据。

- 在多重背景块内可以寻址块接口中的数据，与寻址所有其它数据的方式相同。

例如，使用“#On”访问多重背景块的输入参数“On”。

- 在调用块中使用“多重背景名称.变量名”访问多重背景块的数据。

例如，使用“Multi.On”访问多重背景块“Multi”的输入参数“On”。

7.2.4.2 通过 LAD 访问 I/O

基本知识

CPU 的过程映像是每个周期更新一次。但是，在对时间要求苛刻的应用中，读取或传送数字输入或输出当前状态的频率可能必须高于每个周期一次。为此，可以在操作数中使用 I/O 访问 ID 的后缀直接访问 I/O。

立即读

如果要直接从 I/O 读取输入，请使用外围设备输入存储区 (PI)，而不是使用输入 (I) 的过程映像。可以位、字节、字或双字形式读取 I/O 存储区。

要直接从输入上读取信号，可以给操作数添加 I/O 访问后缀“:P”，例如“Switch:P”。

立即写

要对输出执行立即写操作，请使用外围设备输出 (PQ) 存储区，而不是使用输出 (Q) 的过程映像。可以位、字节、字或双字形式对外围输出存储区执行写操作。

要将信号直接传送到外围设备输出，可以给操作数添加 I/O 访问后缀“:P”，例如“Motor1:P”。

**警告**

请注意，对 I/O 执行立即写操作可能会导致出现危险状态，例如，在一个程序循环中对一个输出执行多次写操作。

参见

I/O 数据区 (页 297)

过程映像输入/输出 (页 295)

7.2.4.2 连接隐藏参数**简介**

根据所用的 CPU，可以在程序中使用通过 TIA Portal 分派的复杂指令。这些指令可以包含已声明为隐藏的参数。

如果某个指令包含隐藏参数，则指令框的下边缘上会有一个小箭头。可以通过白色字体来识别隐藏参数。

您可以随时显示和连接隐藏参数。

显示或隐藏隐藏参数

要显示或隐藏隐藏参数，请按以下步骤操作：

1. 要显示隐藏参数，请单击指令框下边缘上的向下小箭头。
2. 要隐藏隐藏参数，请单击指令框下边缘上的向上小箭头。

连接隐藏参数

要连接参数，请按以下步骤操作：

1. 如同连接可见参数那样连接隐藏参数。
隐藏参数随即会变成可见参数。

参见

将操作数插入 FBD 指令 (页 457)

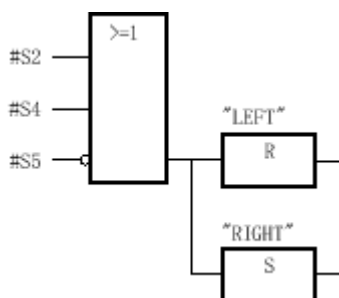
7.2.4.2 FBD 中的分支

7.2.4.2 有关 FBD 中分支的基本信息

定义

可以使用功能块图 (FBD) 编程语言创建并行分支。通过在功能框之间插入的分支可以完成该操作。可以在分支内插入附加的功能框，以此可以建立复杂的功能块图。

下图显示了一个使用分支的实例：



参见

- FBD 中分支的规则 (页 462)*
- 在 FBD 程序段中插入分支 (页 463)*
- 删除 FBD 程序段中的分支 (页 463)*

7.2.4.2 FBD 中分支的规则

规则

以下规则适用于 FBD 中分支的使用：

- 分支向下打开。
- 只能在 FBD 元素之间插入分支。
- 要删除分支，必须删除所有 FBD 元素，包括分支本身。
- 如果删除两个分支之间的连接，则这些分支的 FBD 元素将移动到单独的逻辑路径中。

参见

- 有关 FBD 中分支的基本信息 (页 462)*
- 在 FBD 程序段中插入分支 (页 463)*
- 删除 FBD 程序段中的分支 (页 463)*

7.2.4.2 在 FBD 程序段中插入分支

要求

有一个可用的程序段。

步骤

要在程序段中插入新分支，请按以下步骤操作：

1. 打开“指令”(Instructions) 任务卡。
2. 在“指令”(Instructions) 窗格中浏览到“常规 > 分支”(General > Branch)。
3. 通过拖放操作将元素移动到程序段中的所需位置。

参见

FBD 中分支的规则 (页 462)
有关 FBD 中分支的基本信息 (页 462)
删除 FBD 程序段中的分支 (页 463)

7.2.4.2 删除 FBD 程序段中的分支

要求

有一个可用的分支。

步骤

要删除分支，请按以下步骤操作：

1. 选择将分支链接到主分支的连线。
2. 在快捷菜单中选择“删除”(Delete) 命令。

结果

这样便删除了分支。可在程序段内自由放置连接到已删除分支的功能框。

参见

FBD 中分支的规则 (页 462)
有关 FBD 中分支的基本信息 (页 462)
在 FBD 程序段中插入分支 (页 463)

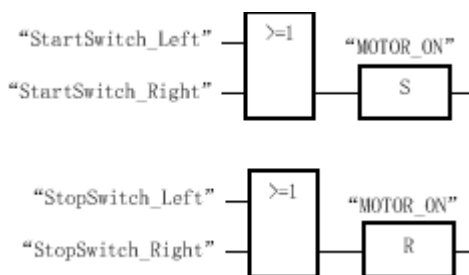
7.2.4.2 FBD 中的逻辑路径

7.2.4.2 简介

逻辑路径的使用

程序以一个或多个程序段表示。程序段可以包含一条或多条逻辑路径，路径上的二进制信号以功能框形式来安排。

下图所示为在一个程序段中使用多条逻辑路径的实例：



规则

使用逻辑路径时，请记住下列规则：

- 逻辑路径之间不允许存在连接。
- 每个程序段只允许有一个跳转指令。跳转指令的定位规则仍然有效。

执行逻辑路径

逻辑路径从上到下、从左到右执行。这意味着首先执行第一个程序段的第一条逻辑路径中的第一条指令。然后执行该逻辑路径的所有其它指令。接着执行第一个程序段的所有其它逻辑路径。只有先执行完所有逻辑路径后，才执行下一个程序段。

使用跳转时，将绕开常规的逻辑路径执行顺序，而是执行跳转目标的指令。

分支和逻辑路径之间的区别

分支和逻辑路径之间的区别是：逻辑路径是独立分支，还可以出现在其它程序段中。另一方面，通过分支可设计并联结构。

参见

插入逻辑路径 (页 465)

删除逻辑运算路径 (页 465)

7.2.4.2 插入逻辑路径

要求

- 块已经打开。
- 有一个可用的程序段。

步骤

要在程序段中插入新逻辑路径，请按以下步骤操作：

1. 在程序段中插入任意指令，使它不与现有指令连接。
这样便插入了一个新逻辑路径。
2. 在新逻辑路径的末端插入赋值指令。
3. 在新逻辑路径中插入其它指令。

参见

简介 (页 464)
删除逻辑运算路径 (页 465)

7.2.4.2 删除逻辑运算路径

要求

有一条可用的逻辑路径。

步骤

要删除逻辑路径，请按以下步骤操作：

1. 按住鼠标左键，并拖出一个框包围该逻辑路径。同时确保选中该逻辑路径的所有指令。还可以按住 <Shift> 键并选择该逻辑路径的第一条和最后一条指令。
2. 右键单击该逻辑路径中的一条指令。
3. 在快捷菜单中选择“删除”(Delete) 命令。

参见

简介 (页 464)
插入逻辑路径 (页 465)

7.2.4.2 使用自由格式的注释

7.2.4.2 有关在 FBD 中使用自由格式的注释的基本信息

简介

使用自由格式的注释可在图形编程语言的源代码中添加注释，其与文本语言的行注释类似。
自由格式的注释可用于所有非二进制功能框。

7.2.4.2 插入自由格式的注释

步骤

要插入自由格式的注释，请按以下步骤操作：

1. 右键单击要为其插入自由格式的注释的对象。
2. 在快捷菜单中选择“插入注释”(Insert comment) 命令。

将打开一个含有标准注释的注释框。该注释框通过箭头连接到相应的对象。

7.2.4.2 编辑自由格式的注释

简介

可以如下所述编辑自由格式的注释：

- 更改注释文本
- 更改注释框的位置或大小
- 将注释附加到其它元素
- 显示和隐藏自由格式的注释

更改注释文本

要更改自由格式注释的文本，请按以下步骤操作：

1. 在注释框中双击。
2. 输入所需的文本。

更改注释框的位置

要更改注释框的位置，请按下列步骤操作：

1. 左键单击注释框，并按住鼠标按键。
2. 将注释框拖到所需位置。

更改注释框的大小

要更改注释框的大小，请按下列步骤操作：

1. 单击注释框的边。
2. 拖动注释框右下角的移动滑块直至注释框达到所需的大小。

将注释附加到其它元素

要将自由格式的注释附加到其它元素，请按以下步骤操作：

1. 左键单击链接注释框和指令的箭头的尖端，并按住鼠标按键。
2. 将箭头拖动到要附加该注释的元素。可能的插入点以绿色正方形标记。
3. 释放鼠标按键。

显示和隐藏自由格式的注释

要显示或隐藏自由格式的注释，请按以下步骤操作：

1. 在工具栏中单击“显示/隐藏自由格式的注释”(Show / hide free comments)。

7.2.4.2 删除自由格式的注释

步骤

要删除自由格式的注释，请按以下步骤操作：

1. 右键单击要删除的自由格式的注释。
2. 在快捷菜单中选择“删除”(Delete) 命令。

7.2.4.2 查找程序中的语法错误

用户程序中的语法错误会通过红色下划线高亮显示。这样有助于用户一眼就看到错误条目。也可以在块内从一个错误跳到另一个错误。此外，还会在巡视窗口的“信息”(Info) 选项卡中列出语法错误和错误消息。

步骤

要查找程序中的语法错误，请按以下步骤操作：

1. 选择块标题。
2. 单击工具栏上的“转到下一个错误”(Go to next error)。

将标记块中的第一个错误。

可以使用工具栏上的“转到下一个错误”(Go to next error) 和“转到上一个错误”(Go to previous error) 查找和更正块中的所有错误。

或者：

1. 在巡视窗口中使用“信息”(Info) >“语法”(Syntax) 打开错误列表。

所有语法错误都在该表格中列出，并带有错误的简短描述。

2. 如果存在错误，则单击错误文本旁的蓝色问号可获得有关解决该问题的信息。
3. 双击要更正的错误。
相应的错误将高亮显示。

7.2.4.2 更改编程语言

7.2.4.2 更改编程语言的规则

规则

如果要更改块的编程语言，请遵守以下规则：

- 只能在图形编程语言 LAD 和 FBD 间切换。
- 如果无法更改块的个别程序段的语言，这些程序段将以其原来的语言显示。
- 只能更改整个块的编程语言。不能更改单个程序段的编程语言。
- 但是可以使用其它编程语言在块内创建程序段，然后将其复制到所需的块。

参见

更改编程语言 (页 468)

7.2.4.2 更改编程语言

步骤

要更改编程语言，请按以下步骤操作：

1. 在项目树中右键单击该块。
2. 在快捷菜单中选择“属性”(Properties) 命令。
将打开“属性”(Properties) 对话框。
3. 在区域导航中选择“常规”(General) 条目。
4. 在“语言”(Language) 下拉列表中选择新的编程语言。
5. 单击“确定”(OK) 确认选择。

参见

更改编程语言的规则 (页 468)

7.2.4.2 处理程序执行错误

7.2.4.2 故障排除选项

简介

可以使用错误处理设置，指定在发生任何编程错误或 IO 区域访问错误时，系统应如何响应。可使用下列选项：

- 使用操作系统已提供的系统响应。
- 使用单独的局部错误处理。

使用系统响应

操作系统将以下列系统响应进行错误处理：

- 停机
- 忽略

使用局部错误处理

局部错误处理是指块内的错误处理。局部错误处理有下列优点：

- 错误信息存储在系统存储器中，可对其进行查询和评估。
- 该错误信息可用于为块中发生的错误设计响应。
- 所设计的错误评估和错误响应不会中断程序循环。
- 如非必要，局部错误处理不会对系统性能增加负荷。如果没有错误发生，则不执行所设计的错误分析和响应功能。

局部错误处理仅适用于已明确进行过这种设置的块。如果为某个块设置了局部错误处理，则执行该块期间将忽略系统响应。

参见

程序执行的基础知识 (页 298)

事件和 OB (页 298)

局部错误处理 (页 470)

GetError (页 887)

GetErrorID (页 890)

7.2.4.2 系统响应

使用

执行用户程序期间，可能会发生必须要对其进行响应的编程错误和 IO 区域访问错误。操作系统为此提供两种系统响应：“忽略”和“停止”。

如果没有使用局部错误处理，则始终使用系统响应处理编程错误。如果为块设置了局部错误处理，则它优先于系统响应。

7.2.4.2 局部错误处理

7.2.4.2 局部错误处理的原理

简介

通过局部错误处理可查询块内发生的错误和评估相关的错误信息。可以为组织块 (OB)、功能块 (FB) 和功能 (FC) 设置局部错误处理。如果启用了局部错误处理，则会忽略系统响应。

局部错误处理仅适用于已明确进行过这种设置的块。局部错误处理设置不会被调用块采用，也不会传送到被调用块。对于更高级别的块和更低级别的块，如果没有对这些块的错误处理进行编程，则系统设置仍然适用。

局部错误处理的一般步骤

如果根据局部错误处理设置执行块期间出错，将根据以下错误类型启动预定义的响应：

- 写错误：忽略这些错误，程序继续执行。
- 读错误：程序将使用替换值“0”继续执行。
- 执行错误：中止指令的执行。程序从下一条指令继续执行。

有关所发生的第一个错误的信息存储在系统存储器中。可以使用指令 (GET_ERROR 或 GET_ERRORID) 查询和输出该信息。还可对输出的错误信息进行进一步处理。其它指令可用于分析错误信息并针对该错误编写响应功能。

查询第一个错误的相关信息时，将启用系统存储器中的错误存储空间。如果随后发生其它错误，则会输出下一个错误的相关信息。

用于局部错误处理的指令

可使用下列指令进行局部错误处理：

- [GET_ERROR \(页 887\)](#)
- [GET_ERRORID \(页 890\)](#)

指令 GET_ERROR 和 GET_ERRORID 在各自输出的错误信息量方面有所不同。

7.2.4.2 错误输出优先级

优先级概述

在局部错误处理期间，将显示发生的第一个错误的相关信息。如果指令执行期间同时发生多个错误，将根据优先级显示这些错误。下表给出了不同类型的错误的优先级。

优先级	错误类型
1	程序代码错误
2	缺少引用

优先级	错误类型
3	无效范围
4	DB 不存在
5	操作数不兼容
6	指定区域的宽度不够
7	定时器或计数器不存在
8	禁止对 DB 进行写访问
9	I/O 错误
10	指令不存在
11	块不存在
12	无效嵌套深度

最高优先级为 1，最低优先级为 12。

参见

GetError (页 887)
GetErrorID (页 890)

7.2.4.2 启用块的局部错误处理

简介

如果在程序段中插入以下指令之一，即可启用块的局部错误处理。

- [GET_ERROR](#) (页 887)
- [GET_ERRORID](#) (页 890)

如果启用了块的局部错误处理，将忽略该块的系统响应。

先决条件

- 块已打开。
- “指令”(Instructions) 任务卡已打开。

步骤

要启用块的局部错误处理，请按以下步骤操作：

1. 浏览到“指令”(Instructions) 任务卡的“扩展指令”(Extended Instructions) 窗格。
2. 打开“程序控制”(Program Control) 文件夹。
3. 将 GET_ERROR 或 GET_ERRORID 指令拖到所需的程序段中。

结果

为打开的块启用了局部错误处理。在巡视窗口的“属性 > 特性”(Properties > Attributes) 下,“在块内处理错误”(Handle errors within block) 复选框会被选中。无法在巡视窗口中编辑该设置。通过删除所插入的局部错误处理指令,可以禁用局部错误处理。

7.2.4.3 数据块编程

7.2.4.3 数据块编程的基本原理

数据块 (DB) 用于保存程序执行期间写入的值。

与代码块相比,数据块仅包含变量声明。不包含任何程序段或指令。变量声明定义数据块的结构。

数据块类型

有两种类型的数据块:

- 全局数据块

全局数据块不能分配给代码块。可以从任何代码块访问全局数据块的值。全局数据块仅包含静态变量。

在声明表中输入全局数据块要包含的变量。

- 背景数据块

背景数据块是直接分配给功能块 (FB) 的块。背景数据块的结构取决于功能块的接口声明。它准确地包含接口中声明的块参数和静态变量。

但可以在背景数据块中定义实例特定的值,例如,所声明的变量的初始值。

数据值的保持性

要防止在发生电源故障时数据丢失,可以将数据值存储在保持性存储区中。

在线监视数据值

如果存在在线连接,则可显示数据块中的当前数据值。

参见

创建数据块 (页 376)

7.2.4.3 数据块声明表的结构

数据块声明表的结构

声明表的结构取决于各种设置,例如,块的符号访问。

下图以在线视图中的符号访问数据块为例显示声明表的结构。

Block_1_DB							
	名称	数据类型	默认值	初始值	监视值	保持性	注释
1	▼ Input					<input type="checkbox"/>	
2	MyTag1	Bool	false	false	<input checked="" type="checkbox"/> FALSE	<input type="checkbox"/>	
3	MyTag2	Bool	false	false	<input checked="" type="checkbox"/> FALSE	<input type="checkbox"/>	
4	▼ Output					<input type="checkbox"/>	
5	MyTag3	Bool	false	false	<input checked="" type="checkbox"/> FALSE	<input type="checkbox"/>	
6	▼ InOut					<input type="checkbox"/>	
7	▼ Static	▼				<input type="checkbox"/>	

各列的含义

下表说明了各列的含义：

列	描述
名称	变量名称。
数据类型	变量的数据类型。
偏移量	变量的相对地址。 该列仅在无法以符号形式寻址的数据块中可见。
默认值	预分配给更高级别代码块接口中的变量的值。 默认情况下会隐藏“默认值”(Default value) 列。要显示该列，请单击“扩展模式”(Expanded mode)。只能在更高级别的代码块中更改该列中包含的值。这些值仅显示在数据块中。
初始值	在启动时变量采用的值。 代码块中定义的默认值被用作数据块创建期间的初始值。然后可以在数据块中使用实例特定的初始值替换这些值。 可选择是否指定初始值。如果未指定初始值，则在启动时变量将采用默认值。如果也没有定义默认值，将使用相应数据类型的有效默认值。例如，将值“FALSE”指定为 BOOL 的标准值。
监视值	CPU 中的当前数据值。 只有当在线连接可用并选择“监视”(Monitor) 按钮时，此列才会出现。
保持性	将变量标记为具有保持性。保持性变量的值将保留，即使在电源关闭后也是如此。
注释	用于描述变量的注释。

参见

创建数据块 (页 376)
创建全局数据块的数据结构 (页 475)
定义初始值 (页 477)
设置数据值的保持性

有关初始值的基本信息 (页 477)

7.2.4.3 创建数据块

先决条件

“程序块”(Program blocks) 文件夹在项目树中已打开。

步骤

要创建数据块，请按以下步骤操作：

1. 双击“添加新块”(Add new block) 命令。
将打开“添加新块”(Add new block) 对话框。
2. 单击“数据块 (DB)”(Data block (DB)) 按钮。
3. 输入数据块名称。
4. 选择数据块类型。 用户有以下选择：
 - 要创建全局数据块，请选择列表条目“全局 DB”(Global DB)。
 - 要创建背景数据块，请从列表中选择要为其分配背景数据块的目标功能块。
5. 如果想要分配块编号，则选择“手动”(Manual) 选项按钮。
6. 如果选择使用手动分配，请在输入域中输入块编号。
7. 要为新数据块添加其它属性，请单击该对话框下半部分中的“更多信息”(Further information) 旁的箭头。
将显示一个具有更多输入域的区域。
8. 输入所需的所有属性。
9. 单击“确定”(OK) 确认输入。

结果

创建了新数据块。 可以在项目树的“程序块”(Program blocks) 文件夹中找到该数据块。

注意

您可选中该对话框底部的“添加新对象并打开”(Add new and open) 复选框。 这样，创建块后会立即将其打开。

参见

背景数据块 (页 321)
数据块编程 (页 472)
全局数据块 (DB) (页 321)

7.2.4.3 创建全局数据块的数据结构

7.2.4.3 声明基本数据类型的变量

要求

全局数据块已打开。

步骤

要声明基本数据类型的变量，请按以下步骤操作：

1. 在“名称”(Name) 列中输入变量名称。
2. 单击“数据类型”(Data type) 列中的箭头键，然后选择所需的数据类型。
另请参见：[声明 STRUCT 数据类型的变量 \(页 476\)](#)
另请参见：[声明 ARRAY 数据类型的变量 \(页 475\)](#)
3. 对要声明的所有变量重复步骤 1 和 2。

参见

[显示和编辑块属性 \(页 383\)](#)
[表格中的键盘快捷键 \(页 121\)](#)
[编辑数据块的声明表 \(页 408\)](#)
[定义初始值 \(页 477\)](#)
[设置数据值的保持性](#)
[在线监视数据值 \(页 480\)](#)

7.2.4.3 声明 ARRAY 数据类型的变量

要求

数据块已打开。

步骤

要声明 ARRAY 数据类型的变量，请按以下步骤操作：

1. 在“名称”(Name) 列中输入变量名称。
2. 在“数据类型”(Data type) 列中，单击箭头键并选择数据类型“类型的 Array [o .. hi]”。
3. 然后在同一列中输入维数，用于显示上限、下限和数据类型（例如 BOOL 类型的 ARRAY [1..10] ）。
在右方括号和关键字“of”之间以及在“of”和 ARRAY 元素的数据类型信息之间必须插入一个空格。

输入 ARRAY 元素的初始值

要初始化 ARRAY 的各个元素，请按以下步骤操作：

1. 单击“扩展模式”(Expanded mode) 按钮。

该 ARRAY 随即打开且各个 ARRAY 元素在单独的行中显示。

2. 在“初始值”(Initial value) 列中输入所需值。
3. 要再次关闭 ARRAY，请单击“扩展模式”(Expanded mode) 按钮。

参见

定义初始值 (页 477)
设置数据值的保持性
在线监视数据值 (页 480)
编辑数据块的声明表 (页 408)

7.2.4.3 声明 STRUCT 数据类型的变量

要求

数据块已打开。

步骤

要声明 STRUCT 数据类型的变量，请按以下步骤操作：

1. 在“名称”(Name) 列中输入变量名称。
2. 在“数据类型”(Data type) 列中，单击箭头键并选择“Struct”条目。
在新变量后将插入两个空行。
3. 在第一个空行中插入第一个结构元素。
在该元素后会再插入一个空行。
4. 对所有其它结构元素重复步骤 3。
不必显式结束结构。结构会以输入的最后一个元素结束。
5. 要在结构后插入新变量，请在结构结尾后留一个空行，然后在第二个空行中启动新变量。

输入 STRUCT 元素的初始值

要初始化结构的各个元素，请按以下步骤操作：

1. 单击“扩展模式”(Expanded mode) 按钮。
结构随即打开且各个 STRUCT 元素在单独的行中显示。
2. 在“初始值”(Initial value) 列中输入所需值。
3. 要再次关闭 ARRAY，请单击“扩展模式”(Expanded mode) 按钮。

参见

定义初始值 (页 477)
设置数据值的保持性
在线监视数据值 (页 480)
编辑数据块的声明表 (页 408)
STRUCT (页 355)

7.2.4.3 定义初始值

7.2.4.3 有关初始值的基本信息

定义

变量的初始值是在启动后变量将采用的值，由用户来定义。

保持性变量具有特殊状态。“暖启动”之后，这些变量会保留自身的值，不会复位为初始值。

默认值和实例特定的初始值

- 数据块的结构可派生自更高级别的元素。例如，背景数据块以更高级别代码块的接口为基础。

在这种情况下，您可以定义更高级别的元素中每个变量的默认值。代码块中定义的默认值被用作数据块创建期间的初始值。然后可以在数据块中使用实例特定的初始值替换这些值。

可选择是否指定初始值。如果未指定初始值，则在启动时变量将采用默认值。如果也没有定义默认值，将使用相应数据类型的有效默认值。例如，将值“FALSE”指定为 BOOL 的标准值。

参见

定义初始值 (页 477)

数据块声明表的结构 (页 472)

在块接口中声明局部变量 (页 401)

7.2.4.3 定义初始值

要求

数据块已打开并包含已声明的变量。

步骤

要确定变量的初始值，请按以下步骤操作：

1. 单击“扩展模式”(Expanded mode) 按钮以显示结构化数据类型的所有元素。
2. 在“初始值”(Initial value) 列中输入所需的初始值。该值必须与变量的数据类型相匹配且不可超出数据类型可用的范围。

注意

“默认值”(Default value) 列包含了为更高级别代码块的接口中的变量所定义的默认值。

结果

这样就定义了实例特定的初始值。如果未将变量声明为具有保持性，则启动时该变量会采用定义的值。

参见

有关初始值的基本信息 (页 477)

7.2.4.3 设置数据值的保持性

7.2.4.3 设置全局数据块中的保持性

7.2.4.3 全局数据块中的执行保持特性和符号寻址

符号访问

通过设置符号访问，可指定全局数据块的变量声明方式，即仅符号方式或者混用符号方式和绝对方式。如果启用符号访问，则只能通过输入符号名来声明变量。这种情况下会自动寻址变量，从而以最佳方式利用存储容量。如果未启用符号访问，变量也会获得一个固定的绝对地址。存储区的分配取决于所声明变量的地址。

保持特性

如果启用了符号访问，则可指定全局数据块中各个变量的保持特性。如果将变量定义为具有保持性，则该变量会自动存储在全局数据块的保持性存储区中。

如果在全局数据块中禁用符号访问，则无法指定各个变量的保持特性。在这种情况下，保持性设置对全局数据块的所有变量都有效。

参见

仅块的符号寻址 (页 335)

保持性存储区 (页 295)

7.2.4.3 设置全局数据块中的保持特性

简介

根据符号访问的设置，可设置全局数据块中单个变量或所有变量的保持特性。要指定保持特性，必须考虑全局数据块的以下设置：

- 启用变量的符号访问：
可为各个变量指定保持特性。对于具有结构化数据类型的变量，将为所有变量元素传送保持性设置。
- 禁用变量的符号访问：
保持性设置对全局数据块的所有变量都有效。

要求

全局数据块已打开。

步骤

要设置全局数据块中变量的保持特性，请按以下步骤操作：

1. 在“保持性”(Retain) 列中，选择要为其设置保持特性的变量的单元格。
2. 选中“保持性”(Retain) 列中的复选框。

结果

- 如果启用了符号访问，则所选变量被定义为具有保持性。
- 如果禁用了符号访问，则全局数据块的所有变量都被定义为具有保持性。

参见

全局数据块中的执行保持特性和符号寻址 (页 478)
仅块的符号寻址 (页 335)
保持性存储区 (页 295)

7.2.4.3 设置背景数据块中的保持性

7.2.4.3 背景数据块中的执行保持特性

背景数据块中局部变量的保持特性

根据在较高级别的功能块中符号访问的设置，可以设置背景数据块中所有变量或部分变量的保持性。要指定背景数据块中的保持特性，必须考虑功能数据块的以下设置：

- **在较高级别的功能块中启用符号访问：**
背景数据块会采用较高级别功能块中所有变量的保持性设置。此属性会显示在背景数据块的结构中且无法编辑。
- **在较高级别的功能块中禁用符号访问：**
保持性设置可编辑，且对背景数据块的所有变量有效。

在背景数据块属性视图的“属性”(Attribute) 组中，较高级别功能块的符号访问设置显示为写保护，且无法更改。在背景数据块中，可通过“保持性”(Retain) 列中设置的复选标记来识别被定义为具有保持性的变量。

参见

设置背景数据块中的保持特性 (页 480)
代码块的保持特性和符号寻址 (页 406)
设置功能块中局部变量的保持特性 (页 406)
保持性存储区 (页 295)

7.2.4.3 设置背景数据块中的保持特性

简介

在背景数据块中，保持特性是否可编辑取决于较高级别功能块中的符号访问设置。在这种情况下，无法单独为单个的变量设置保持特性。保持性设置始终会影响背景数据块中包含的所有变量。

另请参见：[背景数据块中的执行保持特性 \(页 479\)](#)

要求

- 背景数据块已打开。

步骤

要设置背景块中变量的保持特性，请按以下步骤操作：

1. 选择一个变量。
2. 选中变量的“保持性”(Retain) 列中的复选框。

结果

所有变量都被定义为具有保持性。

7.2.4.3 在线监视数据值

7.2.4.3 在线监视数据块中的数据值

可以在声明表中直接监视 CPU 中变量的当前数据值。

要求

- 在线连接可用。
- 数据块已装载到 CPU 中。

步骤

要监视数据值，请按以下步骤操作：

1. 单击“全部监视”(Monitor all) 按钮启动监视。
此表中将显示附加的“监视值”(Monitor value) 列。该列显示当前数据值。
另请参见：[数据块声明表的结构 \(页 472\)](#)
2. 再次单击“全部监视”(Monitor all) 按钮结束监视。

7.2.4.3 编辑数据块的声明表

7.2.4.3 删除数据块中的变量

要求

全局数据块已打开。

步骤

要删除变量，请按以下步骤操作：

1. 选择包含要删除的变量的行。
2. 在快捷菜单中选择“删除”(Delete) 命令。

注意

只有全局数据块才允许删除变量。不能直接更改背景数据块的结构，因为这些块的结构是由功能块定义的。

在块属性中输入数据块的类型。

另请参见：[显示和编辑块属性 \(页 383\)](#)

参见

[表格中的键盘快捷键 \(页 121\)](#)
[调整表格 \(页 119\)](#)

7.2.4.3 插入表格行

步骤

要在鼠标指针位置上方插入行，请按以下步骤操作：

1. 将鼠标指针放置在要在其上方插入一个新行的行中。
2. 在表格的工具栏上，单击“插入行”(Insert row) 按钮。

将在所选行上方插入一个新行。

参见

[表格中的键盘快捷键 \(页 121\)](#)
[调整表格 \(页 119\)](#)

7.2.4.3 在末尾添加表行

步骤

要在表的结尾或复杂数据类型的结尾添加新行，请按以下步骤操作：

1. 单击表工具栏上的“添加行”(Add row) 按钮。
在表的末尾或复杂数据类型的结尾将添加一个新的空行。

参见

[表格中的键盘快捷键 \(页 121\)](#)

[调整表格 \(页 119\)](#)

7.2.5 比较块

7.2.5.1 块比较的基本知识

功能

块比较功能可用于将编程设备中的块与 CPU 中的块进行比较。有两种显示比较差异的方法：

- [比较编辑器 \(页 136\)](#)
在比较编辑器中仅比较块的时间戳。此比较给出所有块间差异的概况。可通过符号读取相应的状态。
- [详细比较 \(页 483\)](#)
并排打开块的离线版本和在线版本并高亮显示差异。

可对以下块执行块比较：

- 代码块
对于代码块，可使用比较编辑器运行比较并进行详细比较。
- 数据块
只能在比较编辑器中使用时间戳来比较数据块。

要比较的对象

比较代码块时，会比较块接口以及各个程序段。此外，还会确定所有不同的变量名。所有注释和其它块属性则不包括在比较范围内。

详细比较代码块的步骤

比较程序段时，会先检测插入或删除的程序段。然后比较其它程序段。如果运算符和操作数相同，则指令是相同的。在任何情况下，总是会显示每次操作的第一个差异。但是，可显示每个程序段的多个差异。

如果块接口发生变化，则在线和离线版本代码块的块接口的相应时间戳也会有变化。该变化意味着程序代码的时间戳发生变化。因此，比较块接口时，会首先比较在线和离线版本代码块的程序代码的时间戳。如果这些时间戳相同，则认为接口是相同的。如果这些时间戳不同，则接

下来会逐区段地比较接口的数据类型。在比较中包括了多重背景和 PLC 数据类型的比较。如果区段中的数据类型相同，则比较变量的初始值。将显示所有差异。

7.2.5.2 详细比较

7.2.5.2 开始详细比较

要求

- 比较编辑器已打开。
- 存在在线和离线版本不同的块。

另请参见：[比较编辑器概述 \(页 136\)](#)

步骤

要对块执行详细比较，请按以下步骤操作：

1. 在比较编辑器中，右键单击要对其执行详细比较的块。
2. 在快捷菜单中选择“开始详细比较”(Start detailed comparison) 命令。

结果

为块的在线版本打开程序编辑器的一个实例，为块的离线版本打开另一个实例，并且并排排列这两个实例。这两个不同版本中的差异以彩色高亮显示。

另请参见：[详细比较的表示法 \(页 483\)](#)

7.2.5.2 详细比较的表示法

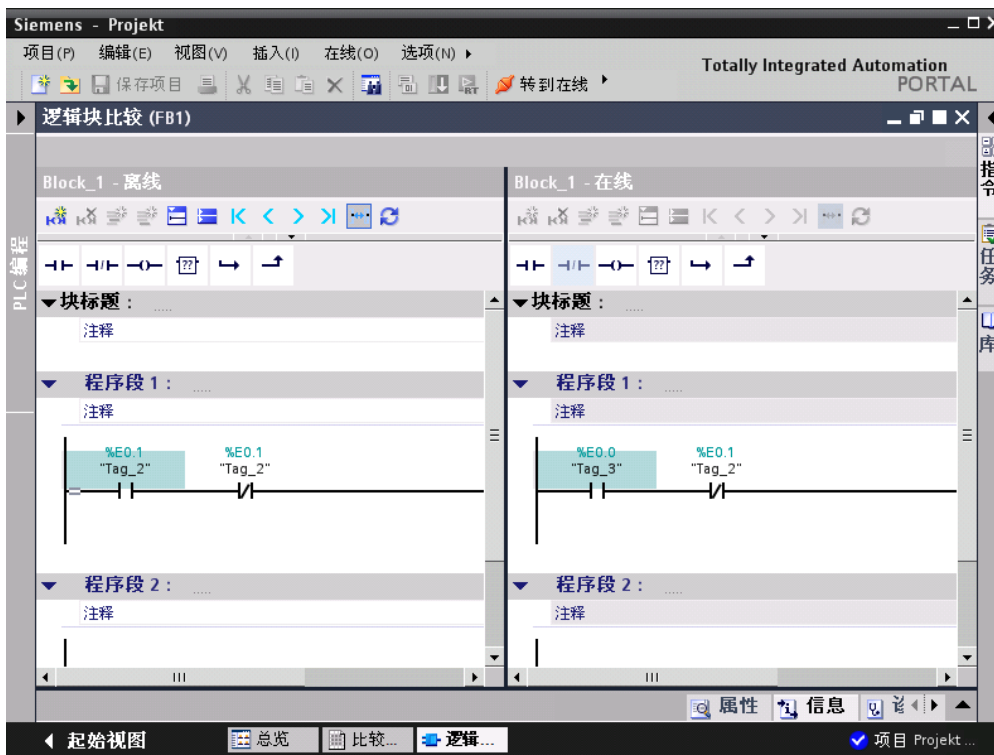
差异的标识

通过详细比较可以对块的在线版本不同于离线版本的确切位置进行标识。用户通过以下颜色编码可以尽快找到这些位置：

- 差异所在的行以灰色高亮显示。
- 不同的操作数和操作以绿色高亮显示。
- 如果程序段数不同，则会添加伪程序段，使相同程序段的显示同步。这些伪程序段会以灰色高亮显示并在程序段标题栏中包含文本“未找到任何相应的程序段”(No corresponding network found)。伪程序段不可编辑。
- 如果程序段的顺序不正确，则会在相应位置插入伪程序段。这些伪程序段以灰色高亮显示并在程序段的标题栏中包含文本“程序段未同步”(The networks are not synchronized)。伪程序段还特别包含一个“转到程序段 <编号>”链接，可用于浏览到相应的程序段。

实例

下图所示为 LAD 编程语言的详细比较实例：



减少所显示的差异数

为清晰起见，并不高亮显示每种情况下的所有差异，而是仅高亮显示操作的第一个差异。例如，在块的离线版本和在线版本中，如果某个有多个输入的功能框的所有输入均不同，则仅将第一个输入高亮显示为差异。可解决此差异并更新比较列表。然后下一个输入将高亮显示为差异。

因此，程序段中高亮显示的差异数目取决于现有操作的数量。

7.2.5.2 浏览详细比较

要求

已运行详细比较。

浏览到差异

要浏览到两个块之间的差异，请按以下步骤操作：

1. 在巡视窗口的“信息”(Info) > “比较结果”(Comparison result) 下打开详细比较的结果列表。
2. 双击一个差异。

在两个编辑器中均将选中该差异。

或者：

1. 单击工具栏上的以下浏览按钮之一：
 - 第一个差异的位置
浏览到块中第一个差异，并在两个编辑器中均显示该差异。
 - 上一个差异的位置
从当前位置开始，浏览到上一个差异，并在两个编辑器中均显示该差异。
 - 下一个差异的位置
从当前位置开始，浏览到下一个差异，并在两个编辑器中均显示该差异。
 - 最后一个差异的位置
浏览到块中最后一个差异，并在两个编辑器中均显示该差异。

关闭/启用编辑器之间的垂直滚动同步

两个编辑器同步滚动可确保垂直滚动期间相应的程序段并排显示。可关闭或启用此模式。为此，请按以下步骤操作：

1. 要关闭同步滚动，请单击工具栏中的“同步编辑器之间的滚动”(Synchronize scrolling between editors) 按钮。
2. 要重新启用同步滚动，可再次单击工具栏中的“同步编辑器之间的滚动”(Synchronize scrolling between editors) 按钮。

7.2.5.2 详细比较期间更改程序段

更改离线块

可以随时更改离线块。

更改在线块

无法更改在线块。

7.2.5.2 更新比较结果

只要更改了对象，比较结果就不再有效，因此必须进行更新。

要求

已运行详细比较。

步骤

要更新比较结果，请按以下步骤操作：

1. 单击工具栏中的“更新比较结果”(Update the comparison result)。

7.2.6 编译块

7.2.6.1 有关编译块的基本信息

简介

用户程序必须先经过编译才能在 CPU 中执行。每次进行更改后都需要重新编译程序。

编译期间会执行以下步骤：

- 检查用户程序的语法错误。
- 检查被编译块中的所有块调用。 如果对被调用块的接口进行过更改，则会更新块调用。
- 最后，将用户程序编译成 CPU 可读取的代码。

编译方法

可在以下窗口或编辑器中启动编译：

- 在项目树中编译块
用于编译单个块或同时编译“程序块”(Program blocks) 文件夹中的所有块。
- 在程序编辑器中编译块
该选项用于编译单个打开的块。
- 在调用结构或从属性结构中编译块
用于编译个别的块。
另请参见：[调用结构 \(页 505\)](#)，[从属性结构 \(页 511\)](#)

编译选项

如果要在项目树中编译块，可选择更多选项：

- 软件
仅编译有更改的块。
- 软件 (重建所有块)
编译所有块。建议在第一次编译时以及在进行了重大修订后执行此操作。

参见

[编译项目数据 \(页 142\)](#)

7.2.6.2 在项目导航中编译块

可在项目树中编译一个块、多个块或所有块。

要求

项目树已打开。

在项目树中编译多个块

要在项目树中编译多个块，请按以下步骤操作：

1. 在项目树中打开“程序块”(Program blocks) 文件夹。
2. 选择要编译的块。
3. 可选择以下两个不同选项之一进行编译：
 - 如果要仅编译自上次编译后进行的更改，请在快捷菜单中选择“编译 > 软件”(Compile > Software) 命令。
 - 如果要全部编译所有块，请在快捷菜单中选择“编译 > 软件 (重建所有块)”(Compile > Software (rebuild all blocks)) 命令。

在项目树中编译所有块

要在项目树中编译“程序块”(Program blocks) 文件夹中的所有块，请按以下步骤操作：

1. 在项目树中选择“程序块”(Program blocks) 文件夹。
2. 可选择以下两个不同选项之一进行编译：
 - 如果要仅编译自上次编译后进行的更改，请在快捷菜单中选择“编译 > 软件”(Compile > Software) 命令。
 - 如果要全部编译所有块，请在快捷菜单中选择“编译 > 软件 (重建所有块)”(Compile > Software (rebuild all blocks)) 命令。

结果

生成块的代码。

如果要编译的块调用了接口已更改的块，则会更新块调用。

- 如果已选择一个或多个块进行编译，则会更新所选块中的块调用。
- 如果选择了“程序块”(Program blocks) 文件夹进行编译，则程序中的所有块调用以及所使用的 PLC 数据类型都将更新。

巡视窗口的“信息”(Info) > “编译”(Compilation) 下的消息将报告编译是否成功。

另请参见：[更正编译错误 \(页 488\)](#)

参见

[查找程序中的语法错误 \(页 467\)](#)

7.2.6.3 在程序编辑器中编译块

要求

要编译的块已打开。

步骤

要在程序编辑器中编译块，请按以下步骤操作：

1. 在程序编辑器的指令窗口中，右键单击程序段下方的白色区域。
2. 在快捷菜单中选择“编译”(Compile) 命令。

结果

生成块的代码。如果要编译的块具有接口已更改的块调用，则会更新这些块调用。

巡视窗口的“信息”(Info) >“编译”(Compilation) 下的消息将报告编译是否成功。

另请参见：[更正编译错误 \(页 488\)](#)

7.2.6.4 更正编译错误

在巡视窗口的“信息”(Info) >“编译”(Compile) 中，可查看编译是否成功或在程序中是否检测到错误。如果发生错误，则需要更正，然后重新启动编译。

步骤

要更正通过编译发现的错误，请按以下步骤操作：

1. 在巡视窗口中使用“信息”(Info) >“编译”(Compile) 打开错误列表。
2. 单击错误文本旁的蓝色问号以获得有关解决该问题的信息。
3. 双击要更正的错误。
相应的错误将高亮显示。
4. 更正错误。
5. 重新启动编译。

7.2.7 下载块

7.2.7.1 块装载简介

将块下载到设备

为使 CPU 可以执行用户程序，必须先编译程序，然后将其下载到设备中。可使用以下选项进行下载：

- 将块下载到程序编辑器
该选项用于下载单个打开的块。
- 将块下载到项目树
该选项用于同时下载块文件夹中的多个或全部块。
- 将块下载到可访问设备
可通过拖动块将其下载到可访问设备。

注意

请遵守以下规则：

- 如果将专有技术受保护的块下载到设备，则在安装该块时不会安装任何恢复信息。这意味着从设备上上传专有技术受保护的块后，将无法重新打开。
 - 要避免调用块与被调用块之间的不一致性，每次进行全局更改（例如块接口中的更改）时都会编译并下载所有受影响的块。
-

从设备上传块

可将块从可访问设备上传到项目中。例如，如果要编辑仅存在于设备中的块，此步骤是必需的。所有现有块（组织块、功能块、功能、数据块）和全局变量也将被上传到项目中。

从存储卡上传块或将块下载到存储卡

存储卡是插入卡，例如，用于扩展设备的存储器。只能将西门子 SD 卡用于 S7-1200 系列的设备。

要使用存储卡代替装载存储器，必须将用户程序或各个块下载到存储卡。同样可以将块从存储卡上传回项目中。

注意

上传到存储卡或从存储卡下载时，请注意以下说明：

- 如果 CPU 之前不包含任何程序且在 CPU 中插入空存储卡，则会将程序从 PG/PC 装载到存储卡，而不是装载到 CPU。
 - 如果在 CPU 启动之前插入空存储卡，则 CPU 上的程序会自动传送到存储卡。CPU 上的程序即被删除。
 - 如果在 CPU 启动之前在 CPU 中插入包含程序的存储卡，且 CPU 已包含程序，则会执行存储卡上的程序，而不会执行 CPU 上的程序。CPU 上的程序将被删除。
-

参见

从存储卡上传块或将块下载到存储卡 (页 494)

从可访问设备装载块 (页 494)

7.2.7.2 将块下载到设备

7.2.7.2 在程序编辑器中将块装载到设备

要求

要下载的块已打开。

步骤

要将块从程序编辑器下载到设备，请按以下步骤操作：

1. 在程序编辑器的指令窗口中，右键单击程序段下方的白色区域。
2. 在快捷菜单中选择“下载到设备”(Download to device) 命令。

如果尚未建立在线连接，则会打开“扩展的下载到设备”(Extended download to device) 对话框。

如果已指定在线连接，则会打开“下载预览”(Download preview) 对话框。然后，继续执行步骤 5。

注意

也可以使用“在线”(Online) 菜单打开“扩展的下载到设备”(Extended download to device) 对话框。

3. 从“扩展的下载到设备”(Extended download to device) 对话框中的“用于下载的 PG/PC 接口”(PG/PC interface for downloading) 下拉列表中选择 PG/PC 的接口。如果子网可用，也可以从“子网的连接”(Connection to subnet) 下拉列表中选择子网。
4. 在“目标子网中的可访问设备”(Accessible devices in target subnet) 表中选择设备并单击“装载”(Load) 确认选择。
必要时会编译项目数据。“下载预览”(Download preview) 对话框将打开。此对话框会显示消息并建议上传必需的操作。
5. 查看这些消息，并在必要时启用“操作”(Action) 列中的操作。

注意事项

如果存在功能混乱或程序错误，则在设备运行期间执行所建议的操作可能会导致严重的财产损失或人员受伤！

在启动这些操作之前，请确保不会出现任何危险情况。

只要可进行下载，“下载”(Download) 按钮就会被启用。

6. 单击“下载”(Download)。
7. 将下载块并打开“下载结果”(Download results) 对话框。此对话框会显示下载后的状态和操作。
8. 如果要在下载后直接重新启动模块，则选中“全部启动”(Start all) 复选框。
9. 要关闭“下载结果”(Download results) 对话框，请单击“完成”(Finish)。

结果

块的代码下载到设备中。如果更改操作影响了其它块，则将编译这些块并将它们下载到设备中。会删除设备中仅在线存在的块。通过下载所有受影响的块并删除设备中不需要的块，可避免用户程序中块与块之间的不一致。

巡视窗口的“信息”(Info) > “常规”(General) 下的消息将报告下载过程是否成功。

7.2.7.2 在项目树中将块装载到设备中

在项目树中，可将一个块、多个块或所有块下载到设备中。

在项目树中将多个块下载到设备

要从项目树中将一个或多个块下载到设备，请按以下步骤操作：

1. 在项目树中打开“程序块”(Program blocks) 文件夹。
2. 选择要下载的块。
3. 在快捷菜单中选择“下载到设备”(Download to device) 命令。
4. 如果只想下载自上次下载之后进行的更改，请选择“软件”(Software) 选项。如果要完整地下载所有块，请选择“软件 (全部)”(Software (all)) 选项。
 - 如果尚未建立在线连接，则会打开“扩展的下载到设备”(Extended download to device) 对话框。
 - 如果已指定在线连接，则会打开“下载预览”(Download preview) 对话框。然后，继续执行步骤 7。

注意

也可以使用“在线”(Online) 菜单打开“扩展的下载到设备”(Extended download to device) 对话框。

5. 从“扩展的下载到设备”(Extended download to device) 对话框中的“用于下载的 PG/PC 接口”(PG/PC interface for downloading) 下拉列表中选择 PG/PC 的接口。如果子网可用，也可以从“子网的连接”(Connection to subnet) 下拉列表中选择子网。
6. 在“目标子网中的可访问设备”(Accessible devices in target subnet) 表中选择设备并单击“装载”(Load) 确认选择。

必要时会编译项目数据。“下载预览”(Download preview) 对话框将打开。此对话框会显示消息并建议下载必需的操作。
7. 查看这些消息，并在必要时启用“操作”(Action) 列中的操作。

注意事项

如果存在故障或程序错误，则在设备运行期间执行所建议的操作可能会导致严重的财产损失或人员受伤！

在启动这些操作之前，请确保不会出现任何危险情况。

只要可进行下载，“下载”(Download) 按钮就会被启用。

8. 单击“下载”。
将下载块并打开“下载结果”(Download results) 对话框。此对话框会显示下载后的状态和操作。
9. 如果要在下载后直接重新启动模块，则选中“全部启动”(Start all) 复选框。
1. 要关闭“下载结果”(Download results) 对话框，请单击“完成”(Finish)。
- 0.

在项目树中将所有块下载到设备

要将“程序块”(Program blocks) 文件夹中的所有块从项目树下载到设备，请按以下步骤操作：

1. 在项目树中选择“程序块”(Program blocks) 文件夹。
2. 在快捷菜单中选择“下载到设备”(Download to device) 子菜单。
3. 如果只想下载自上次下载之后进行的更改，请选择“软件”(Software) 选项。如果要完整地下载所有块，请选择“软件 (全部)”(Software (all)) 选项。
 - 如果尚未建立在线连接，则会打开“扩展的下载到设备”(Extended download to device) 对话框。
 - 如果已指定在线连接，则会打开“下载预览”(Download preview) 对话框。然后，继续执行步骤 6。

注意

也可以使用“在线”(Online) 菜单打开“扩展的下载到设备”(Extended download to device) 对话框。

4. 从“扩展的下载到设备”(Extended download to device) 对话框中的“用于下载的 PG/PC 接口”(PG/PC interface for downloading) 下拉列表中选择 PG/PC 的接口。如果子网可用，也可以从“子网的连接”(Connection to subnet) 下拉列表中选择子网。
5. 在“目标子网中的可访问设备”(Accessible devices in target subnet) 表中选择设备并使用“装载”(Load) 确认选择。
必要时会编译项目数据。“下载预览”(Download preview) 对话框将打开。此对话框会显示消息并建议下载必需的操作。
6. 查看这些消息，并在必要时启用“操作”(Action) 列中的操作。

注意事项

如果存在故障或程序错误，则在设备运行期间执行所建议的操作可能会导致严重的财产损失或人员受伤！

在启动该操作之前，请确保不会出现任何危险情况！

只要可进行下载，“下载”(Download) 按钮就会被启用。

7. 单击“下载”。
将下载块并打开“下载结果”(Download results) 对话框。此对话框会显示下载后的状态和操作。
8. 如果要在下载后直接重新启动模块，则选中“全部启动”(Start all) 复选框。

9. 要关闭“下载结果”(Download results) 对话框，请单击“完成”(Finish)。

结果

块的代码下载到设备中。如果更改操作影响了其它块，则将编译这些块并将它们下载到设备中。会删除设备中仅在线存在的块。通过下载所有受影响的块并删除设备中不需要的块，可避免用户程序中块与块之间的不一致。

巡视窗口的“信息”(Info) > “常规”(General) 下的消息将报告下载过程是否成功。

7.2.7.2 将块装载到可访问设备

要求

可访问设备已显示。

另请参见：[显示可访问设备 \(页 1422\)](#)

步骤

要将块下载到可访问设备，请按以下步骤操作：

1. 在项目树中打开 PLC 的“程序块”(Program blocks) 文件夹。
2. 选择要下载到可访问设备的块。
3. 在项目树中，将这些块拖到可访问设备的“程序块”(Program blocks) 文件夹。
“下载预览”(Download preview) 对话框将打开。此对话框会显示消息并建议下载必需的操作。
4. 查看这些消息，并在必要时启用“操作”(Action) 列中的操作。

注意事项

如果存在功能混乱或程序错误，则在设备运行期间执行所建议的操作可能会导致严重的财产损失或人员受伤！

在启动该操作之前，请确保不会出现任何危险情况！

5. 只要可进行下载，“下载”(Download) 按钮就会被启用。
6. 单击“下载”(Download) 按钮。
将执行下载。“下载结果”(Download results) 对话框随即打开。在此对话框中，可检查下载是否成功并进行任何其它可能必要的操作。
7. 如果要在下载后直接重新启动模块，则选中“全部启动”(Start all) 复选框。
8. 单击“完成”(Finish)。

结果

块下载到可访问节点。如果更改操作影响了其它块，则也会将这些块下载到可访问节点。会删除设备中仅在线存在的块。通过下载所有受影响的块并删除设备中不需要的块，可避免用户程序中块与块之间的不一致。

巡视窗口的“信息”(Info) >“常规”(General) 下的消息将报告下载过程是否成功。

7.2.7.3 从设备上传块

7.2.7.3 从可访问设备装载块

要求

- 可访问设备已显示。
另请参见：[显示可访问设备 \(页 1422\)](#)
- 项目中的 PLC 文件夹不包含任何数据。

步骤

要将块从可访问设备上传到项目，请按以下步骤操作：

1. 在项目树中，将可访问设备的“程序块”(Program blocks) 文件夹拖到项目中 PLC 的“程序块”(Program blocks) 文件夹。
将打开“上传预览”(Upload preview) 对话框。此对话框会显示消息并建议上传必需的操作。如果项目中的 PLC 文件夹已包含数据，将显示一条消息通知您这些数据将被替换。
2. 查看这些消息，并在必要时启用“操作”(Action) 列中的操作。
3. 只要可进行上传，“从设备上传”(Upload from device) 按钮就会被启用。
4. 单击“从设备上传”(Upload from device) 按钮。

7.2.7.4 从存储卡上传块或将块下载到存储卡

7.2.7.4 将块装载到存储卡

要求

- 存储卡未被标记为程序卡。
- 存储卡的“程序块”(Program blocks) 文件夹已打开。

另请参见：[访问存储卡 \(页 152\)](#)

步骤

要将块下载到存储卡，请按以下步骤操作：

1. 在项目树中打开 PLC 的“程序块”(Program blocks) 文件夹。
2. 选择要下载到存储卡的块。

3. 将项目树中的块拖到存储卡的“程序块”(Program blocks) 文件夹中。
“下载预览”(Download preview) 对话框将打开。此对话框会显示消息并建议上传必需的操作。
4. 查看这些消息，并在必要时启用“操作”(Action) 列中的操作。
5. 只要可进行下载，“下载”(Download) 按钮就会被启用。
6. 单击“下载”(Download) 按钮。
将执行下载。“下载结果”(Download results) 对话框随即打开。在此对话框中，可检查下载是否成功并进行任何其它可能必要的操作。
7. 单击“完成”(Finish)。

结果

块下载到存储卡。如果更改操作影响了其它块，则也会将这些块下载到存储卡。会删除仅存储卡上存在的块。通过下载所有受影响的块并删除存储卡中不需要的块，可避免用户程序中块与块之间的不一致。

巡视窗口的“信息”(Info) >“常规”(General) 下的消息将报告下载过程是否成功。

7.2.7.4 从存储卡装载块

要求

存储卡的“程序块”(Program blocks) 文件夹已打开。

另请参见：[访问存储卡 \(页 152\)](#)

步骤

要将块从存储卡上传到项目，请按以下步骤操作：

1. 在项目树中存储卡的“程序块”(Program blocks) 文件夹中选择要上传的块。
2. 将块拖到 PLC 的“程序块”(Program blocks) 文件夹。
将打开“上传预览”(Upload preview) 对话框。此对话框会显示消息并建议上传必需的操作。
3. 查看这些消息，并在必要时启用“操作”(Action) 列中的操作。
4. 只要可进行上传，“从设备上传”(Upload from device) 按钮就会被启用。
5. 单击“从设备上传”(Upload from device) 按钮。

7.2.8 保护块

7.2.8.1 保护块

简介

通过密码，可防止未经授权用户访问 OB、FB、FC 和 DB 类型的一个或多个块。

如果块专有技术受保护，则只能读取以下数据：

- 接口参数 Input、Output、InOut、Return
- 块标题
- 块注释
- 块属性
- 程序结构
- 交叉引用中不带使用点信息的全局变量

对于专有技术受保护的块，仍然可以执行以下操作：

- 复制和删除
- 在程序中调用
- 编译时进行检查并在必要时进行更新

另一方面，块的代码也受到保护，可防止未授权的读取和修改。

注意

请遵守以下规则：

- 如果将专有技术受保护的块下载到设备，则在安装该块时不会安装任何恢复信息。这意味着从设备上上传专有技术受保护的块后，即使有正确的密码也无法重新打开该块。如果有正确的密码，则还可以比较块的离线版本和在线版本。
 - 如果忘了密码，则无法再访问该块。
-

可能的操作

根据块的保护状态可进行以下操作：

- 如果不存在任何专有技术保护，则可设置专有技术保护。
- 如果已存在专有技术保护，则可以取消专有技术保护或更改密码。

参见

设置专有技术保护 (页 497)

7.2.8.2 设置专有技术保护

设置专有技术保护

通过密码对块应用专有技术保护功能，防止未经授权的访问。

设置专有技术保护

要设置专有技术保护，请按以下步骤操作：

1. 选择要保护的块。
2. 在“编辑”(Edit) 菜单中选择命令“专有技术保护 > 启用专有技术保护”(Know-how protection > Enable know-how protection)。将打开“启用专有技术保护”(Enable know-how protection) 对话框。
3. 在“输入密码”(Enter password) 框中输入密码。
4. 在“确认密码”(Confirm password) 框中输入相同的密码。
5. 单击“确定”(OK) 确认输入。

更改密码

要更改密码，请按以下步骤操作：

1. 选择要更改其密码的专有技术受保护的块。
2. 在“编辑”(Edit) 菜单中选择“专有技术保护 > 更改密码”(Know-how protection > Change password) 命令。“更改密码”(Change password) 对话框随即打开。
3. 在“输入旧密码”(Enter old password) 框中输入旧密码。
4. 在“输入新密码”(Enter new password) 框中输入新密码。
5. 在“确认新密码”(Confirm new password) 框中输入相同的新密码。
6. 单击“确定”(OK) 确认输入。

禁用专有技术保护

要禁用专有技术保护，请按以下步骤操作：

1. 选择要禁用其专有技术保护的块。
2. 在“编辑”(Edit) 菜单中选择命令“专有技术保护 > 禁用专有技术保护”(Know-how protection > Disable know-how protection)。“禁用专有技术保护”(Disable know-how protection) 对话框随即打开。
3. 输入密码。
4. 单击“确定”(OK) 确认输入。

7.3 显示程序信息

7.3.1 可用程序信息概述

程序信息

用户程序的程序信息包含下表指定的视图。

视图	应用
分配列表 (页 498)	概要说明用户程序中已分配的 I、Q 和 M 存储区的地址位。 还指示是否通过访问从 S7 程序中分配了地址或是否已将地址分配给 SIMATIC S7 模块。
调用结构 (页 505)	显示用户程序内块的调用结构并概要说明所用的块及块间的关系。
从属性结构 (页 511)	显示用户程序中使用的块的列表。块显示在第一级，调用或使用此块的块缩进排列在其下方。与调用结构不同，实例块单独列出。
资源 (页 516)	显示用于对象 (OB、FC、FB、DB、PLC 变量和用户自定义数据类型)、CPU 存储区及现有 I/O 模块的 CPU 硬件资源。

同时显示多个视图

可以为一个或多个用户程序生成和显示多个视图以便于测试和更改用户程序。

例如，通过显示多个视图，能够：

- 并排显示用户程序的所有程序信息
- 比较不同的用户程序

7.3.2 显示分配列表

7.3.2.1 分配列表简介

分配列表中的程序信息

分配列表显示是否通过访问从 S7 程序中分配了地址或是否已将地址分配给 SIMATIC S7 模块。因此，它是在用户程序中查找错误或进行更改的重要基础。

分配列表概要说明了下列存储区的字节中的位：

- 输入 (I)

- 输出 (O)
- 位存储器 (M)
- I/O (P)

分配列表的显示

输入、输出和位存储器的分配列表显示在若干独立工作窗口中。

过滤器

可以过滤分配列表中的显示。用户可以使用预定义的过滤器或创建自己的过滤器。

显示交叉引用信息

可选择显示分配列表中所选地址的交叉引用信息。

通过从快捷菜单中选择命令“显示用法”(Display usage)，可在巡视窗口中显示所选地址或各个选定位置的交叉引用。使用命令“工具 > 交叉引用”(Tools > Cross References)，也可以打开所选对象的交叉引用列表。

显示 PLC 变量表

可以从分配列表打开 PLC 变量表并编辑所使用变量的属性。

要执行此操作，请选择分配列表的地址并在快捷菜单中选择“打开编辑器”(Open editor) 命令。

启用保持性显示

通过选择“显示/隐藏保持性”(Display/hide retentivity) 工具栏按钮，可启用和禁用位存储器保持性状态的显示。

参见

分配列表中的符号 (页 500)
分配列表的布局 (页 499)

7.3.2.2 分配列表的布局

分配列表的布局

分配列表显示在按以下对象分离的多个工作窗口中：

- 输入和输出
- 位存储器
- I/O

分配列表中的每一行对应存储区的一个字节，该字节中相应的八个位（从 7 到 0）根据其访问进行标记。总之，通过“条形”指示是按字节 (B)、字 (W) 还是双字 (D) 进行访问。

显示输入、输出和位存储器

该列表显示所用的所有输入/输出字节和位存储器字节及其在 S7 用户程序中的分配。

可在此处找到分配列表中符号的说明。(页 500)













参见

分配列表简介 (页 498)

7.3.2.3 分配列表中的符号

分配列表中符号的含义

下表给出了分配列表中符号的含义：

符号	含义
	指示所选状态下的地址分配。
	指示非所选状态下的地址分配。
	指示指针起始地址和变量地址访问相同的地址范围并且它们都被选中。
	指示指针起始地址和变量地址访问相同的地址范围并且它们未被选择。
	指示所选状态下的指针分配。
	指示非所选状态下的指针分配。
	指示字节访问在使用该字节并选择了相应变量。通过快捷菜单可显示所选变量及 PLC 变量表的交叉引用信息。
	指示字节访问在使用该字节并且未选择相应变量。
	指示字访问在使用该字节并选择了相应变量。通过快捷菜单可显示所选变量及 PLC 变量表的交叉引用信息。
	指示字访问在使用该字节并且未选择相应变量。
	指示双字访问在使用该字节并选择了相应变量。通过快捷菜单可显示所选变量及 PLC 变量表的交叉引用信息。
	指示双字访问在使用该字节并且未选择相应变量。

符号	含义
背景色：灰色	指示字节、字或双字访问在使用某个字节并且硬件也在使用该地址。灰色背景色指示重叠的存储器访问。
背景色：黄色	指示硬件未使用该地址。

参见

分配列表的布局 (页 499)
分配列表简介 (页 498)

7.3.2.4 显示分配列表

要求

已经创建项目，且其中有编写好的块。

步骤

请按如下步骤显示分配列表：

1. 选择块文件夹或其中包含的一个或多个块。
2. 在“工具”(Tools) 菜单中，选择“分配列表”(Assignment list) 命令。

结果

显示所选程序的分配列表。

分配列表中的视图选项

请参考相应的视图选项，可设置这些选项以在分配列表中显示所需信息。

参见

设置分配列表的视图选项 (页 501)
分配列表的布局 (页 499)

7.3.2.5 设置分配列表的视图选项

简介

以下视图选项可用于分配列表：

- 使用的地址 (Used addresses) :
如果选中了此复选框，将显示程序中使用的地址、I/O 和指针。
- 空闲的硬件地址 (Free hardware addresses) :

如果选中了此复选框，则仅显示空闲的硬件地址。

要求

- 已经创建项目，且其中有编写好的块。
- 分配列表已打开。

步骤

请按如下步骤设置分配列表的视图选项：

1. 单击任务栏中的



符号（视图选项）的箭头。

调用结构的视图选项将会打开，并且已选中激活的视图选项前的复选标记。

2. 如果要激活或禁用视图选项，单击相应的复选框以选中或取消选中复选标记。

结果

设置了视图选项并在分配列表中显示所需信息。

7.3.2.6 分配列表中的过滤器选项

过滤器设置

您可以为分配列表定义自己的过滤器设置。可使用以下选项定义过滤器：

- 显示指定地址区的所有地址
- 显示所选地址区中已定义的单个地址，例如，“IB 0”和“IB 200”。
- 显示所选地址区中的完整区域，例如，“IB 0”到“IB 256”。

下表概要说明了所有可用选项：

选项位置	选择	符号	含义
地址区	可单独启用或按默认设置启用所有地址 (I、Q、M)	复选框被选中	分配列表仅显示已启用的地址区 (I、Q、M)
过滤区域	显示所有地址的分配	*	显示已启用地址区 (I、Q、M) 所有地址的分配。
	显示所选地址的分配，例如，输入“IB 0”和“IB 256”的分配	0;256 用分号分隔各个地址和区域。	显示已激活地址区 (I) 的所选地址的分配。
	显示所选区域的分配，例如，输入“IB 0 到 IB 100”和“IB 200 到 IB 256”的分配。	0-100;200-256 应该用连字符连接连续区域。	显示已激活地址区 (I) 的所选区域的分配。

7.3.2.7 定义分配列表的过滤器

要求

- 已经创建项目，且其中有编写好的块。
- 分配列表已打开。

定义过滤器

请按如下步骤定义分配列表的过滤器：

1. 单击任务栏中的



符号 (过滤器)。

将打开“过滤器”(Filter) 对话框。

2. 单击任务栏中的



符号 (创建新过滤器)。

将创建一个名为“Filter_1”的新过滤器。默认情况下会为该过滤器选中所有地址 (输入、输出、位存储器) 的复选框。

3. 如果要更改过滤器的名称，请单击任务栏中的下拉列表并输入一个新名称。

4. 取消选中不希望受该过滤器影响的地址的复选框。

5. 在已激活地址的过滤区域输入以下选项之一：

- 输入“*”以显示所用的所有地址
- 要显示已定义的单个地址，例如，IB 0 和 IB 25，请输入“0;25”。用分号分隔各个地址和区域。
- 要显示完整的地址区，例如，IB 0 到 IB 256，请输入“0-256”。应该用连字符连接完整的地址区。

6. 单击“确定”(OK) 确认输入。

新定义的过滤器以指定名称显示在分配列表的任务栏中。

删除过滤器

删除画面的步骤如下：

1. 单击任务栏中的



符号 (过滤器)。

将打开“过滤器”(Filter) 对话框。

2. 在任务栏的下拉列表中，选择要删除的过滤器。

3. 单击任务栏中的



符号 (删除所选过滤器)。

所选过滤器即被删除。

参见

分配列表中的过滤器选项 (页 502)

显示分配列表 (页 501)

分配列表简介 (页 498)

7.3.2.8 过滤分配列表

要求

- 已经创建项目，且其中有编写好的块。
- 分配列表已打开。

步骤

1. 单击下拉列表上的箭头。
将显示可用过滤器。
2. 选择所需的过滤器。

结果

根据所选过滤器的设置过滤分配列表。

注意

关闭项目时会保存过滤器设置。

7.3.2.9 定义位存储器的保持性存储区

简介

在分配列表中，可以为位存储器定义保持性存储区的宽度。断电之后以及上电后 STOP 切换为 RUN 时，在保持性存储器中寻址的变量内容会被保留。

可在分配列表中启用和禁用保持性位存储器的显示。如果已启用其显示，则会在“地址”(Address)列中用图标标识保持性位存储器。

要求

分配列表已打开。

步骤

要为位存储器定义保持性存储区的宽度，请按以下步骤操作：

1. 单击工具栏中的“保持性”(Retentivity)。
将打开“保持性存储区”(Retentive memory areas) 对话框。

2. 通过在输入域中输入此区域的最后一个字节（从 0 开始计数）来定义保持性存储区的宽度。请注意已分配给保持性区域的变量地址。
3. 将块装载到目标系统。在项目树中选择“程序块”(Program blocks) 文件夹，并在快捷菜单中选择“下载到设备”(Download to device) 子菜单。

结果

定义了保持性存储区的宽度。如果启用了相关的显示设置，则在“地址”(Address) 列中还将用图标指示所有变量的保持性状态。

7.3.2.10 启用保持性位存储器的显示

简介

在分配列表中，可以启用和禁用保持性位存储器的显示。如果已启用保持性的显示，则会在“地址”(Address) 列中用图标标识保持性位存储器。

要求

分配列表已打开。

步骤

请按如下说明启用或禁用保持性位存储器的显示：

1. 单击工具栏中的“显示/隐藏保持性”(Display/hide retentivity)。

结果

如果已启用保持性的显示，则会在位存储区的“地址”(Address) 列中用图标标识保持性变量。如果已禁用保持性显示，则会隐藏“地址”(Address) 列中的图标。

7.3.3 显示调用结构

7.3.3.1 调用结构简介

调用结构

调用结构描述了 S7 程序中块的调用层级。

它提供了以下项的概况：

- 所使用的块
- 到块使用位置的跳转
- 块之间的关系

- 块的局部数据要求
- 块的状态

调用结构中的信息

显示调用结构时会显示用户程序中使用的块的列表。调用结构的第一级以彩色高亮显示，并显示未被程序中的任何其它块调用的块。组织块始终在调用结构的第一级显示。功能、功能块和数据块仅当未被组织块调用时才显示在第一级。当某个块调用其它块或功能时，被调用块或功能以缩进形式列在调用块下。系统功能和系统块只有在被某个块调用时，它们才显示在调用结构中。

视图选项

以下视图选项可用于调用结构：

- 仅显示冲突 (Display conflicts only) :
如果选中了此复选框，则仅显示调用结构中的冲突。
- 组合多个调用 (Group several calls) :
如果选中了此复选框，则将多个块调用组合在一起。块调用数显示在“调用数”(Number of calls) 列中。在“详细资料”(Details) 列的下拉列表中提供有指向各种调用位置的链接。

显示块调用

单击块标题前的箭头，可显示块中的块调用。要显示所有块的调用信息，单击工具栏中的



符号 (全部展开)。

单击



符号 (全部折叠) 可隐藏全体总览。

显示交叉引用信息

通过右键单击相应块并从快捷菜单中选择“显示用法”(Display Usage) 命令，可在巡视窗口中显示块的交叉引用信息。

在程序编辑器中显示块

可从调用结构打开程序编辑器并在其中编辑块。

要执行此操作，请在调用结构中选择所需块并在快捷菜单中选择“打开”(Open) 命令。

显示已删除的块

块已删除的行会用



图标标记。

7.3.3.2 调用结构的布局

调用结构的布局

调用结构的视图由以下列组成：

列	内容/含义
调用结构	显示被调用块的总览 如果选中了“组合调用”(Group calls) 复选框，则会组合若干块调用并显示“调用数”(Number of calls) 列。
调用类型 (!)	显示调用类型，例如递归块调用。
地址	显示块的绝对地址。对于功能块，还会显示其相应背景数据块的绝对地址。
详细资料	显示调用块的网络或接口。此列中的所有信息以链接形式提供。通过此链接，可跳转到程序编辑器中的块调用位置。如果激活“组合多个调用”(Group several calls) 视图选项，则会组合若干调用，并在下拉列表中以链接形式提供这些调用。
局部数据 (在路径中)	指示完整路径的局部数据要求。 只能以符号方式进行寻址的块具有更高的局部数据要求，因为符号寻址的信息与这些块一起存储。
局部数据 (用于块)	显示块的局部数据要求。 只能以符号方式进行寻址的块具有更高的局部数据要求，因为符号寻址的信息与这些块一起存储。

参见




调用结构中的符号 (页 507)




介绍调用结构中的一致性检查 (页 510)

7.3.3.3 调用结构中的符号

调用结构中符号的含义

下表给出了调用结构中符号的含义：

符号	含义
	指示组织块 (OB)。
	指示功能块 (FB)。
	指示功能 (FC)。

符号	含义
	指示数据块 (DB)。
	指示该块已声明为多重背景。
	该对象与连接到左侧的对象之间存在着接口从属性。
	指示需要重新编译该块。
	指示需要重新编译该数据块。
	指示该对象不可用。
	指示接口导致时间戳冲突。
	指示变量导致时间戳冲突。
	指示未通过 OB 直接或间接调用该块。
	指示用专有技术保护标记对象。不能编辑具有此属性的对象。
	指示该块通常被递归调用。
	指示接口中的变量声明具有递归的从属关系： <ul style="list-style-type: none"> • 情况 1：FB1 调用 FB2，然后 FB2 调用 FB1。这些 FB 的背景数据块在接口中包含递归。 • 情况 2：多重背景 FB 使用其父 FB 的背景数据块作为全局 DB。

7.3.3.4 显示调用结构

要求

已经创建包含块的项目。

步骤

请按如下步骤显示调用结构：

1. 选择块文件夹或其中包含的一个或多个块。
2. 在“工具”(Tools) 菜单中，选择“调用结构”(Call structure) 命令。

结果

显示所选对象的调用结构。

参见

设置调用结构的视图选项 (页 509)

7.3.3.5 设置调用结构的视图选项

简介

以下视图选项可用于调用结构：

- 仅显示冲突 (Display conflicts only) :
如果选中了此复选框，则仅显示调用结构中导致冲突的块。
下列块将导致冲突：
 - 执行任何具有更旧或更新代码时间戳的调用的块。
 - 调用接口已更改的块的块。
 - 使用地址和/或数据类型已更改的变量的块。
 - 未被 OB 直接或间接调用的块。
 - 调用不再存在的块的块。
- 组合多个调用 (Group several calls) :
如果选中了此复选框，则将多个块调用和数据块访问组合在一起。块调用数显示在“数目”(Number) 列中。在“详细资料”(Details) 列的下拉列表中提供有指向各种调用位置的链接。

要求

- 已经创建项目，且其中有编写好的块。
- 调用结构已打开。

步骤

请按如下步骤设置调用结构的视图选项：

1. 单击任务栏中的



符号 (视图选项) 的箭头。

调用结构的视图选项将会打开，并且已选中激活的视图选项前的复选标记。

2. 如果要激活或禁用视图选项，单击相应的复选框以选中或取消选中复选标记。

结果

设置了视图选项并在调用结构中显示所需信息。

7.3.3.6 介绍调用结构中的一致性检查

一致性检查

若在生成程序期间或之后更改块的时间戳，将导致时间戳冲突，而这又会导致调用块和被调用块间出现不一致。

使用一致性检查

“一致性检查”功能用于在发生时间戳冲突时显示不一致内容。执行一致性检查时，不一致的块将显示在调用结构中并用相应符号进行标记。

- 通过重新编译块可纠正大多数时间戳和接口冲突。
- 如果通过编译无法解决不一致问题，则可使用“详细资料”(Details) 列中的链接转到程序编辑器中的问题源，然后手动解决所有不一致问题。
- 必须重新编译以红色标记的块。

参见

调用结构中的符号 (页 507)

7.3.3.7 检查调用结构中的块一致性

要求

- 已经创建项目，且其中有编写好的块。
- 调用结构已打开。

步骤

请按如下步骤检查块一致性：

1. 单击任务栏中的



符号 (一致性检查)。

将检查块一致性。不一致的块会相应地用符号进行标记。

2. 如果某个块不一致，请单击调用结构中块标题前的箭头。

将显示不一致的块。确切的问题位置在“详细资料”(Details) 列中以链接形式列出。

3. 单击“详细资料”(Details) 列中的相应链接可跳转到块中需要更正的位置。

4. 检查并更正块中的不一致。

5. 通过选择所需块并在快捷菜单中单击“编译”(Compile) 命令，重新编译块。

6. 通过在快捷菜单中单击“下载到设备”(Download to device) 命令，将更正后的块下载到目标系统。

结果

将检查块一致性。更正块中的不一致。更正后的块装载到了目标系统。

参见

调用结构中的符号 (页 507)

7.3.4 显示从属性结构

7.3.4.1 从属性结构简介

简介

从属性结构显示程序中每个块与其它块的从属关系。

从属性结构中的信息

显示从属性结构时会显示用户程序中使用的块的列表。块显示在最左侧，调用或使用此块的块缩进排列在其下方。

从属性结构还会用符号显示单个块的状态。

导致时间戳冲突并可能导致程序中出现不一致的对象分别以不同符号标记。

从属性结构是对象交叉引用列表的扩展。

视图选项

以下视图选项可用于从属性结构：

- 仅显示冲突 (Display conflicts only) :
如果选中了此复选框，则仅显示从属性结构中的冲突。
- 组合多个调用 (Group several calls) :
如果选中了此复选框，则将多个块调用组合在一起。块调用数以数字形式显示在“从属性结构”(dependency structure) 列中。在“详细资料”(Details) 列的下拉列表中提供有指向各种调用位置的链接。

显示从属性结构

单击块标题前的箭头，可显示调用或使用此块的块。要显示所有块的从属性结构，请单击任务栏中的



符号 (全部展开)。

单击



符号 (全部折叠) 可隐藏全体总览。

显示交叉引用信息

通过右键单击相应块并从快捷菜单中选择“显示用法”(Display Usage) 命令，可在巡视窗口中显示块的交叉引用信息。

在程序编辑器中显示块

可通过从属性结构打开程序编辑器并在其中编辑块。要执行此操作，请在从属性结构中选择所需块并在快捷菜单中选择“打开”(Open) 命令。

显示已删除的块

块已删除的行会用



图标标记。

7.3.4.2 从属性结构的布局

从属性结构的布局

从属性结构的视图由以下列组成：

列	内容/含义
从属性结构	指示程序中的每个块与其它块之间的从属关系。
调用类型 (!)	显示调用类型，例如递归块调用。
地址	显示块的绝对地址。
调用数	指示多个块调用的数目。
详细资料	显示被调用块的网络或接口。此列中的所有信息以链接形式提供。通过此链接，可跳转到程序编辑器中的块调用位置。如果激活“组合多个调用”(Group several calls) 视图选项，则会组合若干调用，并在下拉列表中以链接形式提供这些调用。






参见

从属性结构中的符号 (页 512)

7.3.4.3 从属性结构中的符号

从属性结构中符号的含义

下表给出了从属性结构中符号的含义：

符号	含义
	指示组织块 (OB)。
	指示功能块 (FB)。
	指示功能 (FC)。
	指示数据块 (DB)。
	指示该块已声明为多重背景。
	该对象与连接到左侧的对象之间存在着接口从属性。
	指示需要重新编译该块。
	指示需要重新编译该数据块。
	指示该对象不可用。
	指示接口导致时间戳冲突。
	指示此对象存在不一致。
	指示用专有技术保护标记对象。不能编辑具有此属性的对象。
	指示接口中的变量声明具有递归的从属关系： <ul style="list-style-type: none"> 情况 1：FB1 调用 FB2，然后 FB2 调用 FB1。这些 FB 的背景数据块在接口中包含递归。 情况 2：多重背景 FB 使用其父 FB 的背景数据块作为全局 DB。

7.3.4.4 显示从属性结构

要求

已经创建项目，且其中有编写好的块。

步骤

请按如下步骤显示从属性结构：

1. 选择块文件夹或其中包含的一个或多个块。
2. 在“工具”(Tools) 菜单中，选择“从属性结构”(Dependency structure) 命令。

结果

显示所选程序的从属性结构。

参见

设置从属性结构的视图选项 (页 514)

7.3.4.5 设置从属性结构的视图选项

简介

以下视图选项可用于从属性结构：

- 仅显示冲突 (Display conflicts only) :
如果选中了此复选框，则仅显示从属性结构中的冲突。
下列块将导致冲突：
 - 执行任何具有更旧或更新代码时间戳的调用的块。
 - 被接口已更改的块调用的块。
 - 使用地址和/或数据类型已更改的变量的块。
 - 未被 OB 直接或间接调用的块。
- 组合多个调用 (Group several calls) :
如果选中了此复选框，则将多个块调用组合在一起。块调用数会显示在相关列中。在“详细资料”(Details) 列的下拉列表中提供有指向各种调用位置的链接。

要求

- 已经创建项目，且其中有编写好的块。
- 从属性结构已打开。

步骤

请按如下步骤设置从属性结构的视图选项：

1. 单击任务栏中的



符号 (视图选项) 的箭头。

调用结构的视图选项将会打开，并且已选中激活的视图选项前的复选标记。

2. 如果要激活或禁用视图选项，单击相应的复选框以选中或取消选中复选标记。

结果

设置了视图选项并在调用结构中显示所需信息。

7.3.4.6 介绍从属性结构中的一致性检查

一致性检查

若在生成程序期间或之后更改块的时间戳，将导致时间戳冲突，而这又会导致调用块和被调用块间出现不一致。

使用一致性检查

“一致性检查”功能用于显示不一致内容。执行一致性检查时，不一致的块将显示在从属性结构中并用相应符号进行标记。

- 通过重新编译块可纠正大多数时间戳和接口冲突。
- 如果通过编译无法解决不一致问题，则可使用“详细资料”(Details) 列中的链接转到程序编辑器中的问题源，然后手动解决所有不一致问题。
- 必须重新编译以红色标记的块。

参见

从属性结构的布局 (页 512)
从属性结构中的符号 (页 512)


7.3.4.7 检查从属性结构中的块一致性

要求

- 已经创建项目，且其中有编写好的块。
- 从属性结构已打开。

步骤

请按如下步骤检查块一致性：

1. 单击任务栏中的
 符号 (一致性检查)。
将检查块一致性。不一致的块会相应地用符号进行标记。
2. 如果某个块不一致，请单击调用结构中块标题前的箭头。
将显示不一致的块。确切的问题位置在“详细资料”(Details) 列中以链接形式列出。
3. 检查并更正块中的不一致。
4. 通过选择所需块并在快捷菜单中单击“编译”(Compile) 命令，重新编译块。
5. 通过在快捷菜单中单击“下载到设备”(Download to device) 命令，将更正后的块下载到目标系统。

结果

将检查块一致性。更正块中的不一致。更正后的块装载到了目标系统。

参见

从属性结构中的符号 (页 512)

7.3.5 显示 CPU 资源

7.3.5.1 资源简介

简介

“资源”(Resources) 选项卡指示已组态 CPU 上用于以下对象的硬件资源：

- 使用的编程对象，
- CPU 中存储区的分配，以及
- 现有输入和输出模块的已分配输入和输出。

“资源”选项卡中提供的信息

资源选项卡概要说明了 CPU 上用于以下对象的硬件资源：

- CPU 中使用的编程对象 (例如，OB、FC、FB、DB、PLC 变量和用户定义的数据类型)，
- CPU 上可用的存储区 (工作存储器、装载存储器、保持性存储器)，其最大大小及上述编程对象使用的大小，
- 可为 CPU 组态的模块 (I/O 模块、数字输入模块、数字输出模块、模拟输入模块和模拟输出模块) 的 I/O，包括已使用的 I/O。

最大可用装载存储器的显示

可从“总计”(Total) 行“装载存储器”(Load memory) 列的下拉列表框中选择可用装载存储器的最大大小。

最大可用工作存储器的显示

可从“总计”(Total) 行“工作存储器”(Work memory) 列的下拉列表框中选择可用工作存储器的最大大小。

最大可用保持性存储器的显示

可从“总计”(Total) 行“保持性存储器”(Retentive memory) 列的下拉列表框中选择可用保持性存储器的最大大小。

注意

保持性存储器数据

保持性数据的计算中包括被指定为保持性数据的所有位存储器和数据块。

更新“资源”选项卡中的显示

单击“更新视图”(Update view) 工具栏按钮可更新对象显示。

“资源”选项卡中显示的优点

程序信息对话框的“资源”(Resources) 选项卡提供了所有对象及其使用的相应存储区的详细列表。

该选项卡还会指示资源的短缺情况并有助于避免此情况。

可标识未经过编译的块，因为这类块的大小会用一个问号指示。

参见

“最大内存空间”选项卡的布局 (页 517)

显示资源 (页 518)

选择最大可用装载存储器 (页 519)

7.3.5.2 “最大内存空间”选项卡的布局

程序信息中“资源”选项卡的布局

“资源”(Resources) 选项卡的视图由下列组成：

列	内容/含义
对象	“详细资料”(Details) 区概要说明了 CPU 中可用的编程对象，包括它们的存储器分配。
装载存储器	以百分比和绝对值形式显示 CPU 的最大装载存储器资源。 “总计”(Total) 下显示的值提供了有关装载存储器的最大可用存储空间的信息。 “已使用”(Used) 下显示的值提供有关装载存储器中实际使用的存储空间的信息。
工作存储器	以百分比和绝对值形式显示 CPU 的最大工作存储器资源。 “总计”(Total) 下显示的值提供了有关装载存储器中最大可用存储空间的信息。 “已使用”(Used) 下显示的值提供有关装载存储器中实际使用的存储空间的信息。
保持性存储器	以百分比和绝对值形式显示 CPU 中保持性存储器的最大资源。 “总计”(Total) 下显示的值提供了有关装载存储器中最大可用存储空间的信息。 “已使用”(Used) 下显示的值提供有关装载存储器中实际使用的存储空间的信息。

列	内容/含义
I/O	显示 CPU 上可用的 I/O，包括随后几列中其模块特定的可用性。 “已组态”(Configured) 中显示的值提供有关最大可用 I/O 数的信息。 “已使用”(Used) 下显示的值提供有关装载存储器中实际使用的存储空间的信息。
DI / DQ / AI / AQ	显示已组态和已使用的输入/输出数： DI = 数字输入 DQ = 数字输出 AI = 模拟输入 AQ = 模拟输出 “已组态”(Configured) 中显示的值提供有关最大可用 I/O 数的信息。 “已使用”(Used) 下显示的值提供有关实际使用的输入和输出的信息。

参见

显示资源 (页 518)

选择最大可用装载存储器 (页 519)

资源简介 (页 516)

7.3.5.3 显示资源**要求**

已经创建项目，且其中有编写好的块。

步骤

请按如下步骤显示相应 CPU 存储区的资源：

1. 选择块文件夹或其中包含的一个或多个块。
2. 在“工具”(Tools) 菜单中，选择“资源”(Resources) 命令。

结果

显示 CPU 的存储器资源。

7.3.5.4 选择最大可用装载存储器

要求

已经创建项目，且其中有编写好的块。

步骤

请按如下步骤显示最大可用装载存储器资源：

1. 通过单击图标打开“装载存储器”(Load memory) 列“总计”(Total) 域中的下拉列表。
2. 为所用的 CPU 选择一个相应的值，方法是在下拉列表框中单击该值。

结果

“总计”(Total) 域显示所选的最大存储器资源。

7.4 显示交叉引用

7.4.1 关于交叉引用的常规信息

简介

交叉引用列表提供用户程序中操作数和变量的使用概况。

交叉引用的使用

交叉引用列表有以下优点：

- 创建和更改程序时，保留已使用的操作数、变量和块调用的总览。
- 从交叉引用可直接跳转到操作数和变量的使用位置。
- 程序测试或故障排除期间，系统将给您提供以下信息：
 - 哪个块中的哪条命令处理了哪个操作数，
 - 哪个画面使用了哪个变量，
 - 哪个块被其它哪个块调用。
- 作为项目文档的一部分，交叉引用全面概述了已用的所有操作数、存储区、块、变量和画面。
- 例如，可以显示对象的使用位置以修改或删除对象。
- 可以显示已删除对象的使用位置，并在必要时进行修改。

参见

交叉引用列表的结构 (页 520)

显示交叉引用列表 (页 521)

在巡视窗口中显示交叉引用 (页 522)

7.4.2 交叉引用列表的结构

交叉引用列表的视图

交叉引用列表有两个视图。这两种视图的区别是第一列中显示的对象：

- 使用者：
显示被引用的对象。您可在此处看到对象的使用位置。
- 已使用：
显示引用对象。可以在此处查看对象的使用者。

分配的工具提示提供关于每个对象的附加信息。

交叉引用列表的结构

交叉引用列表具有以下结构：

列	内容/含义
对象	使用下级对象或被下级对象使用的对象的名称。
数量	使用次数
位置	每个使用位置，例如，程序段
属性	被引用对象的特殊属性，例如，多背景声明中的变量名称。
作为	显示对象的附加信息，例如，背景数据块用作模板或用作多重背景。
访问	访问类型，对操作数的访问是读访问 (R)、写访问 (W) 还是二者的组合。
地址	操作数的地址
类型	有关创建对象所使用的类型和语言的信息
路径	对象在项目树中的路径

视安装的产品而定，交叉引用可能显示额外的列或不同的列。

交叉引用列表的设置

可以使用工具栏中的按钮为交叉引用列表进行以下设置：

- 更新交叉引用列表
更新当前交叉引用列表。
- 进行交叉引用列表的设置
在此处选中相关复选框以指定是显示所有使用的、所有未使用的、所有定义的还是显示所有未定义的对象。如果启用“未定义对象”选项，则将显示对先前所删除的对象的引用。
- 折叠条目
通过关闭下级对象减少当前交叉引用列表中的条目。
- 展开条目

通过打开下级对象展开当前交叉引用列表中的条目。

交叉引用列表的排序

可以按升序或降序对“对象”(Object) 列 (包括其它产品特定的列) 中的条目排序。为此，请单击相关的列标题。

参见

关于交叉引用的常规信息 (页 519)

显示交叉引用列表 (页 521)

7.4.3 显示交叉引用列表

先决条件

已创建项目。

简介

根据是在 Portal 视图中还是在项目视图中以及在项目树中选择的对象，有多种显示交叉引用的方法。

在 Portal 视图中，只能显示整个 CPU 的交叉引用；在项目视图中，可以显示以下对象的交叉引用：

- “PLC”文件夹
- “块”(Blocks) 文件夹
- 各个块
- “PLC 变量”(PLC tags) 文件夹
- “变量和连接”(Tags and connections) 文件夹
- 各个变量

显示交叉引用

执行下列步骤，显示交叉引用：

1. 在 Portal 视图中选择所需的操作，例如，“PLC 编程”(Program PLC) 和“显示交叉引用”(Show cross-references) 命令，或在项目视图中选择上述某个对象，然后在“工具”(Tools) 菜单中选择“交叉引用”(Cross-references) 命令。
2. 单击“使用位置”按钮，显示交叉引用列表中所显示的对象的使用位置。
3. 单击“使用”按钮，查看交叉引用列表中所显示的对象的用户。
4. 使用工具栏中的按钮可以执行以下操作：
 - 更新交叉引用列表
 - 进行交叉引用列表的设置
 - 折叠条目

– 展开条目

5. 通过单击相关列标题可以按升序或降序对“对象”(Object) 和“地址”(Address) 列中的条目排序。
6. 要跳转到对象的使用位置，请单击所显示的链接。

参见

[关于交叉引用的常规信息 \(页 519\)](#)

[交叉引用列表的结构 \(页 520\)](#)

7.4.4 在巡视窗口中显示交叉引用

简介

巡视窗口显示在“关于”>“交叉引用”选项卡中选择的对象的相关交叉引用信息。此选项卡显示所选对象的使用位置的实例和使用所选对象的其它对象的实例。

巡视窗口还包括交叉引用中仅在线可用的块。

结构

巡视窗口以表格形式显示交叉引用信息。每列包含有关所选对象及其应用的特定信息和详细信息。下表显示在“关于”>“交叉引用”选项卡中列出的附加信息：

列	含义
对象	使用下级对象或被下级对象使用的对象的名称。
数量	使用次数
位置	每个使用位置，例如，程序段
属性	被引用对象的特殊属性，例如，多背景声明中的变量名称
作为	显示对象的附加信息，例如，背景数据块用作模板或用作多重背景。
访问	访问模式 显示操作数是被读 (R) 和/或写 (W) 操作访问。
地址	操作数的地址
类型	有关创建对象所使用的类型和语言的信息
路径	对象在项目树中的路径

视安装的产品而定，交叉引用可能显示额外的列或不同的列。

7.5 测试用户程序

7.5.1 测试用户程序的基本知识

功能

您可以测试用户程序在设备中的运行情况。然后，可以监视信号状态和变量值，并且可以在程序运行期间给变量赋值以便仿真特定情况。

要求

设备中必须已装载可执行程序。

测试选项

可使用下列测试选项：

- 使用程序状态进行测试

通过程序状态可以监视程序的运行情况。可以显示操作数的值和程序段逻辑运算的结果 (RLO)，进而可以识别和解决程序中的错误。

- 用监视表格测试

通过监视表格可以监视、修改或强制用户程序或 CPU 中的各变量的当前值。可以给各变量赋值来进行测试和在各种不同的情况下运行程序。也可以在 STOP 模式下给 CPU 的输出 I/O 分配固定值，例如用于检查接线情况。

参见

使用程序状态进行测试简介 (页 523)

有关使用监视表格测试的简介 (页 527)

7.5.2 使用程序状态进行测试

7.5.2.1 使用程序状态进行测试简介

功能

如果显示程序状态，则可以监视程序的执行。可以从程序中某一特定位置开始启用状态，然后可以获得各操作数的值的总览和逻辑运算的结果。利用它可以检查自动化系统组件的控制是否正确。



警告

如果存在功能混乱或程序错误，则在设备运行期间执行测试可能导致重大财产损失或人员伤害。在执行测试以前，请确保不会出现危险情况。

7.5.2.2 测试用户程序

7.5.2.2 开/关使用程序状态进行测试

要求

- 设备中有相同的块。
- 块已打开。

开或关程序状态

要开或关块的程序状态，请按下列步骤操作：

1. 单击工具栏中的“启用/禁用监视”(Monitoring on/off) 按钮。

从程序段中的特定点开始启用程序状态

要启动特定点处的程序状态，请按下列步骤操作：

1. 单击工具栏中的“启用/禁用监视”(Monitoring on/off) 按钮。
2. 右键单击要作为程序状态起点的变量。
3. 在快捷菜单中选择“在此处启动监视”(Start monitoring here)。

注意

使用程序状态进行测试的资源是有限的。如果没有足够的资源供当前测试使用，则将终止较早的测试。

结果

如果启用程序状态的显示，则将建立在线连接并显示程序状态。启用程序状态的显示时，系统会询问您是否要中断在线连接。

参见

程序状态的显示 (页 526)

7.5.2.2 程序测试期间编辑块

如果在使用程序状态进行测试期间编辑块，则在线监视将中断并且可以离线编辑块。如果离线时项目中没有该块，则必须先从设备将其装载到项目。编辑完该块后，还必须重新编译和装载。

要求

已启用使用程序状态进行测试。

步骤

要在使用程序状态进行测试期间编辑块，请按下列步骤操作：

1. 根据需要编辑块。

使用程序状态进行测试被中断并且在离线时启用该块（如果离线时其存在）。

2. 如果该块离线时不存在，则从设备将其装载到项目。

3. 编译块。

另请参见：[编译块 \(页 487\)](#)

4. 将块下载到设备。

另请参见：[下载块 \(页 488\)](#)

结果

此时块包含离线和在线时的修改。重新建立在线连接并继续使用程序状态进行测试。

7.5.2.2 在程序状态下修改变量

使用程序状态进行测试时，可以一次性或立即将变量修改为以下值：

- 修改为 1
将“布尔”数据类型变量的值修改为“True”。
- 修改为 0
将“布尔”数据类型变量的值修改为“False”
- 修改数值
可以为不属于“布尔”数据类型的变量输入修改值。

请注意，不能修改输入 I/O。

步骤

要在使用程序状态进行测试期间修改变量，请按以下步骤操作：

1. 右键单击要修改的变量。
2. 在快捷菜单中选择以下命令之一：
 - “修改 > 修改为 1”(Modify > Modify to 0)

- “修改 > 修改为 0”(Modify > Modify to 0)
 - “修改 > 修改值”(Modify > Modify value)
3. 如果选择“修改值”(Modify value)，则将打开“修改”(Modify) 对话框。在“修改值”(Modify value) 对话框中输入所需要的值，并单击“确定”(OK) 进行确认。

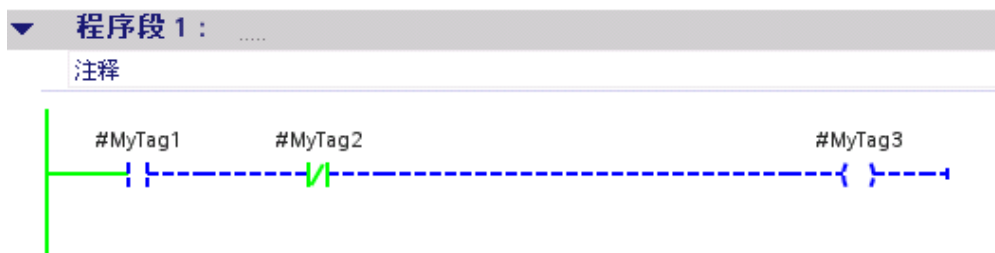
7.5.2.3 程序状态的显示

7.5.2.3 LAD 程序的程序状态显示

程序状态的显示

程序状态的显示是周期性更新的。它从所选程序段开始。

下图所示为 LAD 的程序状态显示的实例：



程序状态的表示法

基于线和符号的颜色和类型，可以快速识别程序段中各运算和线的状态。下表显示了表示法和状态间的关系：

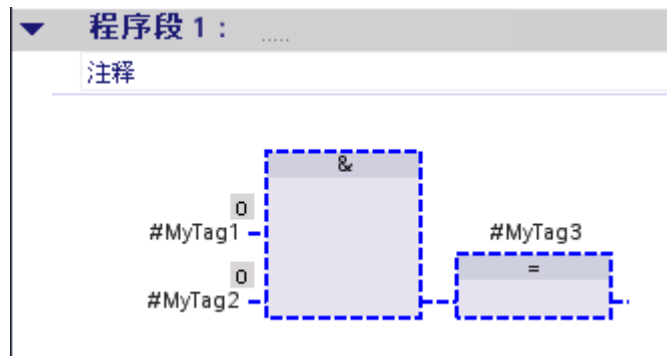
表示法	状态
绿色实线	已满足
蓝色虚线	未满足
灰色实线	未知或未执行
黑色	未互连
100 % 饱和的框中的参数	值为当前值
50 % 饱和的框中的参数	值来自以前的周期。在当前周期中程序未执行到该点

7.5.2.3 FBD 程序的程序状态显示

程序状态的显示

程序状态的显示是周期性更新的。它从所选程序段开始。

下图所示为 FBD 的程序状态显示的实例：



程序状态的表示法

基于线和符号的颜色和类型，可以快速识别程序段中各运算和线的状态。下表显示了表示法和状态间的关系：

表示法	状态
绿色实线	已满足
蓝色虚线	未满足
灰色实线	未知或未执行
黑色	未互连
100 % 饱和的框中的参数	值为当前值
50 % 饱和的框中的参数	值来自以前的周期。在当前周期中程序未执行到该点

操作数的值显示在相关操作数名称上方的一个灰色框中。

7.5.3 使用监视表格测试

7.5.3.1 有关使用监视表格测试的简介

概述

使用监视表格测试时可使用以下功能：

- **“监视变量”(Monitoring tags)**
使用该功能可以在编程设备或 PC 上显示 CPU 中或用户程序中的各个变量的当前值。
- **“修改变量”(Modifying tags)**
使用该功能可以将特定值分配给用户程序中或 CPU 中的各个变量。 使用程序状态进行测试 (页 523) 时也可以进行修改。
- **“启用外围设备输出”(Enable peripheral outputs) 和“立即修改”(Modify now)**
使用这两项功能可以将特定值分配给处于 STOP 模式下的 CPU 的各个外围设备输出。
使用这两项功能还可以检查接线情况。
- **“强制变量”(Forcing tags)**
使用该功能可以将特定值分配给用户程序中或 CPU 中的各个变量。

监视和修改变量

可以监视和修改以下变量：

- 输入、输出和位存储器
- 数据块的内容
- I/O

可能的应用

监视表格的优点是可以存储若干测试环境。这使得用户可以在调试期间或出于维修和维护目的重新进行测试。

参见

创建和编辑监视表格 (页 528)
使用监视表格测试的基本步骤 (页 530)

7.5.3.2 创建和编辑监视表格

7.5.3.2 创建监视表格

简介

如果要监视和修改变量，需要将变量输入到监视表格中。一旦创建了监视表格，即可保存、复制和打印该表格，并反复使用该表格来执行监视、修改和强制变量。

要求

项目已经打开。

步骤

要创建监视表格，请按以下步骤操作：

1. 在状态栏中单击“项目视图”(Project view)。
将显示项目视图。
2. 在项目树中，双击要为其创建监视表格的 CPU。
3. 双击“监视表格”(Watch Tables) 文件夹，然后双击命令“添加监视表格”(Add watch table)。
将添加新的监视表格。
4. 在“名称”(Name) 列或“地址”(Address) 列中，输入要监视或修改的变量的名称或绝对地址。
5. 如果要更改该默认设置，则可从“显示格式”(Display format) 列中的下拉列表选择显示格式。
6. 此时决定是要监视、修改还是强制所输入的变量，如果适用，请输入要修改或强制的期望值。

参见

调整表格 (页 119)

7.5.3.2 打开监视表格

要求

已创建监视表格。

步骤

要打开监视表格，请按以下步骤操作：

1. 打开 CPU 下的“监视表格”(Watch tables) 文件夹。
2. 在该文件夹中双击相关的监视表格。

结果

所选监视表格打开。

7.5.3.2 复制和粘贴监视表格

先决条件

已创建监视表格。

步骤

要复制监视表格，请按以下步骤操作：

1. 右键单击要复制的监视表格。
2. 在上下文菜单中，选择“复制”(Copy)。

3. 在项目树中，打开要粘贴复制的监视表格的目标 CPU 的文件夹结构。
4. 右键单击“监视表格”(Watch Tables) 文件夹。
5. 在上下文菜单中，选择“粘贴”(Paste)。
6. 还可以选择监视表格的全部内容，并将其拖放到另一个监视表格上。

结果

该监视表格的副本将粘贴到 CPU 的“监视表格”(Watch Tables) 文件夹中。

7.5.3.2 保存监视表格

先决条件

已创建监视表格。

步骤

要保存监视表格，请按以下步骤操作：

1. 如果要更改预设的表格名称，则在上下文菜单中选择“重命名”(Rename) 命令并输入新的表格名称。
2. 在“项目”(Project) 菜单中，选择“保存”(Save)。请注意，该保存操作会保存整个项目。

结果

保存监视表格和项目的内容。

注意

可以在重新测试程序时再次使用所保存的监视表格来监视、修改和强制变量。

7.5.3.3 使用监视表格测试的基本步骤

概述

按照下述步骤在监视表格中使用监视、修改和强制变量功能：

创建和编辑监视表格 (页 528)
↓
在监视表格中输入变量 (页 539)
↓
在监视表格中监视变量 (页 540)
↓

在监视表格中修改变量 (页 548)
↓
在监视表格中强制变量 (页 551)

可以在上述部分中找到有关各步骤的详细信息。

7.5.3.4 监视表格的结构

简介

监视表格包含用户为整个 CPU 定义的变量。系统会为项目中创建的每个 CPU 自动生成一个“监视表格”(Watch tables) 文件夹。通过选择“添加监视表格”(Add watch table) 命令可以在该文件夹中创建新的监视表格。

监视表格的布局

监视表格中显示的列与所用的模式有关：基本模式或扩展模式。

扩展模式下会显示下列附加列：

- 使用触发器监视
- 使用触发器修改

还可以基于操作动态更改列名称。

各列的含义

下表给出了基本模式和扩展模式下各列的含义：

模式	列	含义
基本模式		标识符列
	名称	插入变量的名称
	地址	插入变量的地址
	显示格式	所选的显示格式
	监视值	变量值，取决于所选的显示格式。
	修改数值	修改变量时所用的值。
	强制值	强制变量时所用的值。
	值	修改和强制变量时所用的值。
	 (修改)	单击相应的复选框可选择要修改的变量。
	F (强制)	选中相应的复选框可选择要强制的变量。
注释	描述变量的注释	
扩展模式下会显示下列附加列：	使用触发器监视	显示所选的监视模式
	使用触发器修改	显示所选的修改模式

7.5.3.5 基本模式和高级模式

基本模式和扩展模式之间的区别

根据指定的模式，监视表格将显示不同的列和列标题，它们可用于执行不同的操作。

可以在[监视表格的布局 \(页 531\)](#) 中找到这些列的详细列表。

在基本模式和扩展模式之间切换

可选择以下方法在基本模式和扩展模式之间切换：

- 单击图标



(显示/隐藏所有扩展模式的列)。再次单击该图标将返回到基本模式。

或者：

- 在“在线”(Online) 菜单中，选中“扩展模式”(Expanded mode) 命令对应的复选框。取消选中该复选框将返回到基本模式。

扩展模式下的功能

只有在扩展模式下才能使用以下功能：










- 使用触发器监视
- 使用触发器修改
- 启用外设输出

7.5.3.6 监视表格中的图标

监视表格中图标的含义

下表给出了监视表格中图标的含义：

图标	含义
	立即修改所有选定变量的地址一次。该命令被尽快执行一次，而不参考用户程序中定义的触发点。
	参考用户程序中定义的触发点，修改所有选定变量的地址。
	禁用外围设备输出的输出禁用命令。用户因此可以在 CPU 处于 STOP 模式时修改外围设备输出。
	显示扩展模式的所有列。如果再次单击该图标，将隐藏扩展模式的列。
	显示所有强制列。如果再次单击该图标，将隐藏强制列。
	显示所有修改列。如果再次单击该图标，将隐藏修改列。
	开始对所选变量的所有地址进行强制。如果强制功能已经在运行，则将无中断地替换先前的操作。
	停止对激活监视表格中的地址进行强制。
	在激活的监视表格中显示 CPU 中当前被强制的所有地址。
	开始对激活监视表格中的可见变量进行监视。在基本模式下，监视模式的默认设置是“永久”。在扩展模式下，可以为变量监视设置定义的触发点。
	开始对激活监视表格中的可见变量进行监视。该命令将立即执行并监视变量一次。
	显示用于选择要强制的变量的复选框。
	显示用于选择要修改的变量的复选框。
	表示所选变量的值已被修改为“1”。
	表示所选变量的值已被修改为“0”。
	表示将多次使用该地址。

图标	含义
	表示将使用该替代值。替代值是在信号输出模块故障时输出到过程的值，或在信号输入模块故障时用来替换用户程序中过程值的值。用户可以分配替代值（例如，保留旧值）。
	表示地址因已经被修改而被阻止。
	表示无法强制该地址。
	表示无法修改该地址。
	表示无法监视该地址。
	表示将强制该地址。
	表示将部分强制该地址。
	表示发生语法错误。
	表示已选择该地址但目前尚未进行修改或强制。

7.5.3.7 在监视表格中输入变量

7.5.3.7 关于“在监视表格中输入变量”的基本知识

推荐的步骤

选择要修改或监视其值的变量，然后在监视表格中输入这些变量。

输入变量时，请从外向内进行操作。即，在监视表格中先输入输入变量。然后输入受这些输入影响或影响输出的变量。最后输入输出变量。

填写监视表格实例

- 例如，如果希望监视输入位“1.0”、存储器字“5”以及输出字节“0”，则在“地址”(Address) 列中输入以下值：

I 1.0

MW 5

QB 0

或者：

- 在“名称”(Name) 列中输入相应的符号名称。
- 如果不想使用该默认设置，则还可从“显示格式”(Display format) 列的下拉列表选择所需的显示格式。
- 此时决定是要监视、修改还是强制所输入的变量。

语法检查

在监视表格中输入变量时，会在用户退出每个单元格时对单元格进行语法检查。错误的输入条目以红色标记。

注意

如果将鼠标指针放在用红色标记的单元格中，则会显示一条简要信息对错误进行更详细地说明。

参见

监视表格允许使用的地址 (页 535)

监视表格允许使用的修改值 (页 536)

7.5.3.7 监视表格允许使用的地址

在监视表格中输入操作数

下表给出了监视表格允许使用的操作数：

可用的操作数	数据类型	实例 (国际助记符)
输入/输出/位存储器	BOOL	I1.0、Q1.7、M10.1 I0.0:P ; Q0.0:P
输入/输出/位存储器	BYTE	IB1/QB10/MB100 IB1:P ; QB1:P
输入/输出/位存储器	WORD	IW1 ; QW10 ; MW100 IW2:P ; QW3:P
输入/输出/位存储器	DWORD	ID4 ; QD10 ; MD100 ID2:P ; QD1:P
数据块	BOOL	DB1.DBX1.0
数据块	BYTE	DB1.DBB1
数据块	WORD	DB1.DBW1
数据块	DWORD	DB1.DBD1

注意

无法输入“DB0...”，因为它被系统使用！

参见

关于“在监视表格中输入变量”的基本知识 (页 534)

7.5.3.7 监视表格允许使用的修改值

在监视表格中输入修改值

下表给出了在监视表格中输入强制值时允许使用的操作数：

表格 7-3 位操作数

可能的位操作数	可用的修改值实例
I1.0	True
M1.7	False
Q1.0	0
Q1.1:P	1
DB1.DBX1.1	2#0
M1.6	2#1

表格 7-4 字节操作数

可能的字节操作数	可用的修改值实例
IB1	2#00110011
MB12	B#16#1F
QB10	1F
QB11:P	'a'
DB1.DBB1	10

表格 7-5 字操作数

可能的字操作数	可用的修改值实例
IW1	2#0011001100110011
MW12	W#16#ABCD
MW14	ABCD
QW10	B#(12, 34)
QW12:P	12345
DB1.DBW1	'ab'
MW16	S5t#12s340ms
MW18	C#123

可能的字操作数	可用的修改值实例
MW9	D#2006-12-31

表格 7-6 双字操作数

可能的双字操作数	可用的修改值实例
ID1	2#00110011001100110011001100110011
MD0	1.23e4
MD4	01 Feb
QD10	Dw#16#abcdef10
QD12:P	ABCDEF10
DB1.DBD2	b#(12,34,56,78)
MD8	I# -12
MD12	L#12
MD16	123456789
MD20	123456789
MD24	T#12s345ms
MD28	Tod#1:2:34.567
MD32	P#e0.0

7.5.3.7 监视表格允许使用的强制值

在监视表格中输入强制值

下表给出了在监视表格中输入强制值时允许使用的操作数：

表格 7-7 位操作数

可能的位操作数	可用的强制值实例
I1.0:P	True
I1.1:P	False
Q1.0P	0
Q1.1:P	1
I2.0:P	2#0
I2.1:P	2#1

表格 7-8 字节操作数

可能的字节操作数	可用的强制值实例
IB1:P	2#00110011
IB2:P	B#16#1F
QB14:P	1F
QB10:P	'a'
IB3:P	10

表格 7-9 字操作数

可能的字操作数	可用的强制值实例
IW0:P	2#0011001100110011
IW2:P	W#16#ABCD
QW10:P	ABCD
QW12:P	B#(12, 34)
IW4:P	'ab'
IW6:P	12345
IW8:P	S5t#12s340ms
IW10:P	C#123
IW12:P	D#2006-12-31

表格 7-10 双字操作数

可能的双字操作数	可用的强制值实例
ID0:P	2#0011001100110011001100110011
ID4:P	1.23e4
QD10:P	1.2
QD14:P	Dw#16#abcdef10
ID8:P	ABCDEF10
ID12:P	b#(12,34,56,78)
ID16:P	I# -12
ID20:P	L#12
ID24:P	123456789
ID28:P	123456789
ID32:P	T#12s345ms
ID36:P	Tod#1:2:34.567
ID40:P	P#e0.0

7.5.3.7 显示格式简介

监视表格中的显示格式

您选择的显示格式将指定变量值的表示法。在监视表格的“显示格式”(Display formats) 列中，从下拉列表中为每个变量选择显示格式。下拉列表仅提供适合所选变量数据类型的显示格式。

下表给出了可以在监视表格中为相应的变量类型选择的显示格式实例：

数据类型	显示格式
BOOL (页 345)	例如：Bool、Hex、BCD、Octal、Bin、Dec、Dec+/-
BYTE (页 346)	例如：Hex、BCD、Octal、Bin、Dec、Dec+/-、Character
WORD (页 346)	例如：Hex、BCD、Octal、Bin、Dec、Dec+/-、Dec_Sequence、Character、SIMATIC_Timer、Date
DWORD (页 346)	例如：Hex、BCD、Octal、Bin、Dec、Dec+/-、Dec_Sequence、Character、Floating-point number、Time of day、Timer、Pointer

更多信息，请参见[数据类型 \(页 344\)](#) 的说明。

7.5.3.7 选择变量的显示格式

步骤

要在监视表格中选择变量的显示格式，请按以下步骤操作：

1. 在监视表格中输入所需的变量。
2. 在“显示格式”(Display format) 列中单击所需的单元格，然后打开下拉列表。
将显示可用的显示格式。
3. 从该下拉列表中选择一种显示格式。

7.5.3.8 在监视表格中监视变量

7.5.3.8 监视变量简介

简介

通过监视表格可以监视 CPU 中已组态的输入和输出模块的变量，具体情况取决于所选的监视和修改模式 (页 540)。要监视变量，必须存在与 CPU 的在线连接。

用于监视变量的选项

可使用以下选项监视变量：

- “全部监视”(Monitor all)
该命令用于开始对激活的监视表格中的可见变量进行监视：
 - 在基本模式下，监视模式默认设置为“永久”。
 - 在扩展模式下，监视取决于当时定义的监视模式。如果在扩展模式下更改了监视模式，然后又切换到基本模式，则还将在基本模式下继续使用该监视模式设置。
- “立即监视”(Monitor now)
该命令用于开始对激活的监视表格中的可见变量进行监视。仅立即监视变量一次。

7.5.3.8 设置监视和修改模式

简介

通过选择监视和修改模式，用户可以在监视表格中指定变量监视的触发点和持续时间。

可能的监视模式和修改模式 (监视或修改的持续时间)

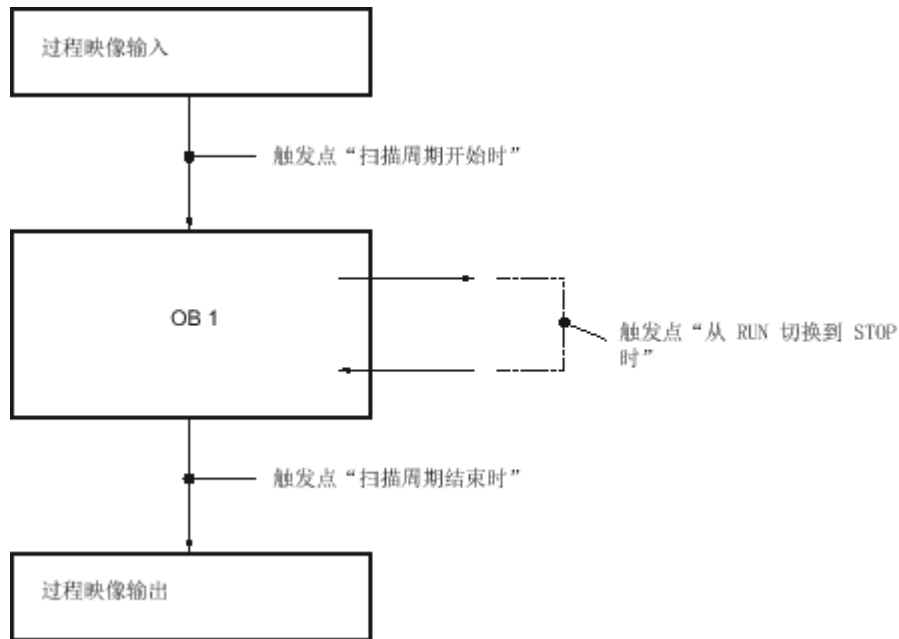
可使用下列监视模式和修改模式：

- 永久
 - 在该模式下，可以在扫描周期开始时监视输入并在扫描周期结束时监视输出。
- 仅一次，扫描周期开始时
- 仅一次，扫描周期结束时
- 永久，扫描周期开始时
- 永久，扫描周期结束时
- 仅一次，从 RUN 切换到 STOP 时
- 永久，从 RUN 切换到 STOP 时

选择触发点

触发点“扫描周期开始”(Beginning of scan cycle)、“扫描周期结束”(End of scan cycle) 和“切换到 STOP”(Switch to stop) 指定从 CPU 读取变量或在 CPU 中执行更新的时间。

下图显示了这些触发点的位置：



触发点的位置

从触发点的位置开始：

- 仅在扫描周期开始（相当于用户程序 OB 1 开始）时适合修改输入，不然在修改后会再次更新过程映像输入，从而导致修改内容被覆盖。
- 仅在扫描周期结束（相当于用户程序 OB 1 结束）时适合修改输出，不然用户程序可能会覆盖过程映像输出。
- 假如监视已激活且用户程序未覆盖修改后的值，则修改后的值将显示在“监视值”(Monitoring value) 列中。

变量的监视

修改变量时，以下说明适用于触发点：

- 将修改模式指定为“仅一次”(once only) 后，如果选择的变量无法修改，用户将收到一条消息。
- 在“永久”(permanent) 修改模式下，用户不会收到消息。

有关“立即修改”命令的说明

可使用“在线 > 立即修改”(Online > Modify now) 命令立即修改所选变量的值。该命令被尽快执行仅一次，而不参考用户程序中定义的位置（触发点）。该功能主要用于在 CPU 处于 STOP 模式时进行修改。

7.5.3.8 用于变量的“全部监视”命令

简介

使用“全部监视”(Monitor all) 命令可以启动对激活的监视表格中的可见变量进行监视。在监视表格的基本模式下，监视模式的默认设置是“永久”。在扩展模式下，可以为变量监视指定定义的触发点。在这种情况下，将参考指定的触发点监视变量。

要求

- 已创建监视表格。
- 存在与 CPU 的在线连接。

步骤

要执行“全部监视”(Monitor all) 命令，请按以下步骤操作：

1. 在监视表格中输入要监视的变量和相应的地址。
2. 通过单击工具栏中的



图标 (显示/隐藏高级设置列) 切换到扩展模式。

3. 如果想要更改变量的默认监视模式，则单击“使用触发器监视”(Monitor with trigger) 列中的相应单元格并从下拉列表中选择所需的监视模式。
4. 单击工具栏中的



图标 (全部监视)。

结果

使用所选的监视模式监视激活监视表格中的变量。

参见

监视表格中的图标 (页 533)

在监视表格中输入变量 (页 539)

基本模式和高级模式 (页 532)

7.5.3.8 用于变量的“立即监视”命令

简介

“立即监视一次”(Monitor once and now) 命令用于开始对变量立即进行监视，而不参考定义的触发点。变量值仅被读出一次并将在监视表格中显示。

要求

- 已创建监视表格。
- 存在与 CPU 的在线连接。

步骤

要执行“立即监视一次”(Monitor once and now) 命令，请按以下步骤操作：

1. 在监视表格中输入要监视的变量和相应的地址。
2. 单击工具栏中的



图标 (立即监视一次)。

结果

仅立即监视激活监视表格中的变量一次。

参见

监视表格中的图标 (页 533)
在监视表格中输入变量 (页 539)
基本模式和高级模式 (页 532)

7.5.3.9 在监视表格中修改变量

7.5.3.9 修改变量简介

简介

通过监视表格可以修改 CPU 中已组态的输入和输出模块的变量，具体情况取决于所选的监视和修改模式 (页 540)。要监视变量，必须存在与 CPU 的在线连接。

注意

修改时，请注意以下说明：

修改无法撤消。

用于修改变量的选项

可使用以下选项修改变量：

- 修改为“0”(Modify to "0")
该命令将所选地址修改为修改值“0”。
- 修改为“1”(Modify to "1")
该命令将所选地址修改为修改值“1”。
- 仅立即修改一次 (Modify once only and immediately)

该命令将激活的监视表格中的所有选定地址“仅立即修改一次”。

- 使用触发器修改

该命令使用所选的监视和修改模式 (页 540) 修改激活监视表格中的所有选定地址。

只有在扩展模式下才能使用“使用触发器修改”功能。用户将不会收到说明实际上是否已使用指定的值修改了所选地址的消息。如果需要这样的确认，则应使用“仅立即修改一次”功能。

- 启用外设输出

该命令会禁用输出禁用命令。

只有在扩展模式下且 CPU 处于 STOP 模式时，才能执行该功能。



危险

修改时存在的危险：

在设备运行期间修改变量或地址后，如果出现故障或程序错误，可能会导致人员重伤和物资损失！

在执行“修改”功能以前，请确保不会出现危险情况。

7.5.3.9 将变量修改为“0”

简介

可以独立于监视和修改模式将一次性值分配给变量并修改这些变量。与“立即触发”(Trigger now) 命令一样，该修改命令被尽快执行，而不参考用户程序中定义的位置。



危险

修改时存在的危险：

在设备运行期间修改变量或地址后，如果出现故障或程序错误，可能会导致人员重伤和物资损失！

在执行“修改”功能以前，请确保不会出现危险情况。

先决条件

- 已创建监视表格。
- 存在与 CPU 的在线连接。

步骤

要将变量修改为“0”，请按以下步骤操作：

1. 在监视表格中输入所需的地址。
2. 选择“在线 > 修改 > 修改为 0”(Online > Modify > Modify to 0) 命令以便使用指定的值修改所选的地址。

结果

所选的地址被修改为“0”。

注意
修改时，请注意以下说明：
修改无法撤消！

7.5.3.9 将变量修改为“1”

简介

可以独立于监视和修改模式将一次性值分配给变量并修改这些变量。与“立即触发”(Trigger now)命令一样，该修改命令被尽快执行，而不参考用户程序中定义的位置。



危险
修改时存在的危险：

在设备运行期间修改变量或地址后，如果出现故障或程序错误，可能会导致人员重伤和物资损失！
在执行“修改”功能以前，请确保不会出现危险情况。

先决条件

- 已创建监视表格。
- 存在与 CPU 的在线连接。

步骤

要将变量修改为“1”，请按以下步骤操作：

1. 在监视表格中输入所需的地址。
2. 选择“在线 > 修改 > 修改为 1”(Online > Modify > Modify to 0) 命令以便使用指定的值修改所选的地址。

结果

所选的地址被修改为“1”。

注意
修改时，请注意以下说明：
修改无法撤消！

7.5.3.9 用于变量的“立即修改”命令

简介

可以独立于监视和修改模式将一次性值分配给变量并立即修改这些变量。与“立即触发”(Trigger now) 命令一样，该修改命令被尽快执行，而不参考用户程序中定义的位置。



危险

修改时存在的危险：

在设备运行期间修改变量或地址后，如果出现故障或程序错误，可能会导致人员重伤和物资损失！

在执行“修改”功能以前，请确保不会出现危险情况。

要求

- 已创建监视表格。
- 存在与 CPU 的在线连接。

步骤

要立即修改变量，请按以下步骤操作：

1. 在监视表格中输入所需的地址和修改值。
2. 通过在“修改值”(Modify value) 列后的列中选中用于设定修改的复选框，选择要修改的地址。
在选中的复选框后将出现一个黄色三角形，这表示此时已选择修改该地址但尚未修改。
3. 选择“在线 > 修改 > 立即修改一次”(Online > Modify > Modify once and now) 命令以便仅使用指定的值修改所选的地址一次。

结果

仅立即修改所选的地址一次。

注意

修改时，请注意以下说明：

修改无法撤消！

7.5.3.9 用于变量的“使用触发器修改”命令

简介

可以根据定义的监视和修改模式将值分配给地址并修改这些地址。将按监视和修改模式中的指定，参考用户程序中定义的触发位置执行修改命令。



危险

修改时存在的危险：

在设备运行期间修改变量或地址后，如果出现故障或程序错误，可能会导致人员重伤和物资损失！

在执行“修改”功能以前，请确保不会出现危险情况。

要求

- 已创建监视表格。
- 存在与 CPU 的在线连接。
- 监视表格必须处于扩展模式。

步骤

要“使用触发器”修改变量，请按以下步骤操作：

1. 在监视表格中输入所需的地址和修改值。
2. 通过在“修改值”(Modify value) 列后的列中选中用于设定修改的复选框，选择要修改的地址。
在选中的复选框后将出现一个黄色三角形，这表示此时已选择修改该地址但尚未修改。

3. 使用工具栏中的



图标 (显示/隐藏所有扩展模式的列) 或“在线 > 扩展模式”(Online > Expanded mode) 命令切换到扩展模式。

将显示“使用触发器监视”(Monitor with trigger) 和“使用触发器修改”(Modify with trigger) 列。

4. 在“使用触发器修改”(Modify with trigger) 列中，从下拉列表框中选择所需修改模式。 有下列选项：
 - 永久
 - 永久，扫描周期开始时
 - 仅一次，扫描周期开始时
 - 永久，扫描周期结束时
 - 仅一次，扫描周期结束时
 - 永久，切换到 STOP 时
 - 仅一次，切换到 STOP 时
5. 使用“在线 > 修改 > 使用触发器修改”(Online > Modify > Modify with trigger) 命令启动修改。
6. 如果想使用触发器修改，则单击“是”(Yes) 确认提示。

结果

使用所选的监视和修改模式修改所选的变量。 不再显示黄色三角形。

注意
修改时，请注意以下说明：
修改无法撤消！

7.5.3.9 启用 I/O 输出

简介

“启用外围设备输出”(Enable peripheral outputs) 功能可禁用外围设备输出的输出禁用命令。用户因此可以在 CPU 处于 STOP 模式时修改外围设备输出。该功能仅在“扩展模式”下的监视表格中可用。



危险
修改时存在的危险：

在设备运行期间修改变量或地址后，如果出现故障或程序错误，可能会导致人员重伤和物资损失！
在执行“修改”功能以前，请确保不会出现危险情况。

要求

- 已创建监视表格。
- 存在与 CPU 的在线连接。
- CPU 必须已处于 STOP 模式才能启用外围设备输出。
- 监视表格必须处于扩展模式。

注意

只有在 STOP 模式下才可使用“启用外围设备输出”(Enable Peripheral Outputs) 功能。该功能会因 CPU 的运行状态变化和在线连接终止而退出。

步骤

要在 STOP 模式下启用外围设备输出，请按以下步骤操作：

1. 在监视表格中输入所需的地址和修改值。
2. 通过在“修改值”(Modify value) 列后的列中选中用于设定修改的复选框，选择要修改的地址。
在选中的复选框后将出现一个黄色三角形，这表示此时已选择修改该地址但尚未修改。

3. 使用工具栏中的



图标 (显示/隐藏所有扩展模式的列) 或“在线 > 扩展模式”(Online > Expanded mode) 命令切换到扩展模式。

将显示“使用触发器监视”(Monitor with trigger) 和“使用触发器修改”(Modify with trigger) 列。

4. 使用操作员面板将相关 CPU 切换到 STOP 模式。
5. 右键单击以打开快捷菜单并选择“启用外围设备输出”(Enable peripheral outputs)。
6. 如果想解除对外围设备输出的输出禁用命令，则单击“是”(Yes) 确认提示。
7. 使用“在线 > 修改 > 立即修改一次”(Online > Modify > Modify once and now) 命令修改外围设备输出。

结果

使用所选的修改值修改外围设备输出。不再显示黄色三角形。

启用外围设备输出

“启用外围设备输出”(Enable peripheral outputs) 功能保持激活，直到：

- 再次通过快捷菜单或“在线 > 修改 > 启用外围设备输出”(Online > Modify > Enable peripheral outputs) 命令禁用“启用外围设备输出”(Enable peripheral outputs) 命令。
- CPU 不再处于 STOP 模式。
- 在线连接被终止。

注意

修改时，请注意以下说明：

修改无法撤消！

7.5.3.10 在监视表格中强制变量

7.5.3.10 强制变量简介

简介

可以使用监视表格给用户程序中的各个变量分配永久值。该操作称为强制。

要使用强制功能，必须存在与 CPU 的在线连接且所用的 CPU 必须支持该功能。

可能的应用

可以强制在用户程序中激活的默认变量值以测试所编写的功能。在基本模式和高级模式 (页 532) 下可以执行强制。

强制变量时的注意事项

强制前，必须检查该步骤的[安全预防措施](#) (页 551)。



危险 防止人身伤害和物资损失！

请注意，执行“强制”功能时操作不当可能会：

- 危及人员生命或健康
 - 导致机器或整个设备损坏
-

用于强制变量的选项

可使用以下选项强制变量：

- 强制为“0”(Force to "0")
该命令将所选地址强制为强制值“0”。
- 强制为“1”(Force to "1")
该命令将所选地址强制为强制值“1”。
- 全部强制 (Force all)
该命令用于开始对激活监视表格中的所选地址进行强制。
- 停止强制 (Stop forcing)
该命令用于停止对激活监视表格中的所有地址进行强制。
- 显示强制值 (Show force values)
该命令用于在激活的监视表格中显示 CPU 中被强制的所有地址。

强制变量时的局限

强制值时请注意以下限制：

- 强制功能始终与所用 CPU 的操作数范围有关。
- S7-1200 产品线的 CPU 只能强制 I/O，即：“Tag_1”:P、“QW0:P”或“IW0:P”。请注意，禁止将“Tag_1”:P 作为位存储器的符号名。

强制变量时的独特之处

请注意，变量的强制会覆盖 CPU 中的值，即使在终止与 CPU 的在线连接后，仍然会继续强制变量。

- **停止强制**
终止在线连接不足以停止强制操作！要停止强制，必须选择“在线 > 强制 > 停止强制”(Online > Force > Stop forcing) 命令。此后，才不再强制激活监视表格中的可见变量。
- **显示所有强制值**

要停止强制变量，必须将变量插入激活的监视表格中。如果已经在强制包含在另一个监视表格中的变量，则必须使用“在线 > 强制 > 显示强制值”(Online > Force > Show forced values) 命令将这些变量插入激活监视表格的末尾。这些变量在激活的监视表格中可见后，才能使用“在线 > 强制 > 停止强制”(Online > Force > Stop forcing) 命令停止强制。

- **停止个别变量的强制**

“在线 > 强制 > 停止强制”(Online > Force > Stop forcing) 命令始终应用于激活监视表格中显示的所有变量。要停止个别变量的强制，必须在激活的监视表格中清除这些变量的复选标记，并使用“在线 > 强制 > 全部强制”(Online > Force > Force all) 命令重新启动强制。

7.5.3.10 强制变量时的安全预防措施

强制变量时的安全预防措施

因为强制功能允许用户永久干预过程，所以必须遵守以下注意事项：



危险 防止人身伤害和物资损失！

请注意，执行“强制”功能时操作不当可能会：

- 危及人员生命或健康
- 导致机器或整个设备损坏



小心 防止人身伤害和物资损失

- 在启动“强制”功能之前，应确保当前没有其他人正在对同一 CPU 执行此功能。
- 只能通过单击



图标（停止强制）或使用“在线 > 强制 > 停止强制”(Online > Force > Stop forcing) 命令停止强制。关闭激活的监视表格并不会停止强制！

- 强制无法撤消！
- 考虑“[修改变量 \(页 543\)](#)”和“[强制变量 \(页 549\)](#)”的区别。
- 如果 CPU 不支持“强制”功能，则无法选择相关图标。
- 如果通过选择“在线 > 修改 > 启用外围设备输出”(Online > Modify > Enable peripheral outputs) 命令并选中相关复选框启用外围设备输出，则所有被强制的输出模块将输出它们的强制值。

7.5.3.10 将变量强制为“0”

简介

可以使用强制功能给用户程序中的各个变量分配永久值。

强制变量时的注意事项

强制前，必须检查强制变量时的安全预防措施 (页 551)。



危险

防止人身伤害和物资损失！

请注意，执行“强制”功能时操作不当可能会：

- 危及人员生命或健康
 - 导致机器或整个设备损坏
-

要求

- 已创建监视表格。
- 存在与 CPU 的在线连接。
- 所用的 CPU 支持强制功能。

步骤

要将变量强制为“0”，请按以下步骤操作：

1. 在监视表格中输入所需的地址。
2. 选择“在线 > 强制 > 强制为 0”(Online > Force > Force to 0) 命令以便使用指定的值强制所选的地址。
3. 单击“是”(Yes) 确认下一个对话框。

结果

所选的地址被强制为“0”。不再显示黄色三角形。红色的“F”显示在第一列中，表示变量处于强制状态。

停止强制

要停止强制，请按以下步骤操作：

1. 选择“在线 > 强制 > 停止强制”(Online > Force > Stop forcing) 命令。
2. 单击“是”(Yes) 确认下一个对话框。

结果

停止对所选值的强制。第一列中的红色“F”不再显示。黄色三角形再次重新出现在相关复选框后面，指示此时选择了强制该地址但尚未强制。

注意

强制时，请注意以下说明：

- 强制无法撤消！
 - 终止在线连接不会停止强制！
 - 要停止强制，强制的地址必须在激活的监视表格中可见。必要时，使用“在线 > 强制 > 显示强制值”(Online > Force > Show force values) 命令在激活的监视表格中显示所有被强制的地址。
-

7.5.3.10 将变量强制为“1”

简介

可以使用强制功能给用户程序中的各个变量分配永久值。

强制变量时的注意事项

强制前，必须检查强制变量时的安全预防措施 (页 551)。



危险

防止人身伤害和物资损失！

请注意，执行“强制”功能时操作不当可能会：

- 危及人员生命或健康
 - 导致机器或整个设备损坏
-

要求

- 已创建监视表格。
- 存在与 CPU 的在线连接。
- 所用的 CPU 支持强制功能。

步骤

要将变量强制为“1”，请按以下步骤操作：

1. 在监视表格中输入所需的地址。
2. 选择“在线 > 强制 > 强制为 1”(Online > Force > Force to 0) 命令以便使用指定的值强制所选的地址。
3. 单击“是”(Yes) 确认下一个对话框。

结果

所选的地址被强制为“1”。不再显示黄色三角形。红色的“F”显示在第一列中，表示变量处于强制状态。

停止强制

要停止强制，请按以下步骤操作：

1. 选择“在线 > 强制 > 停止强制”(Online > Force > Stop forcing) 命令。
2. 单击“是”(Yes) 确认下一个对话框。

结果

停止对所选值的强制。第一列中的红色“F”不再显示。黄色三角形再次重新出现在相关复选框后面，指示此时选择了强制该地址但尚未强制。

注意

强制时，请注意以下说明：

- 强制无法撤消！
 - 终止在线连接不会停止强制！
 - 要停止强制，强制的地址必须在激活的监视表格中可见。必要时，使用“在线 > 强制 > 显示强制值”(Online > Force > Show force values) 命令在激活的监视表格中显示所有被强制的地址。
-

7.5.3.10 用于变量的“全部强制”命令

简介

可以使用强制功能给用户程序中的各个变量分配永久值。

如果强制已激活，则“在线 > 强制 > 全部强制”(Online > Force > Force all) 命令会无中断地替换该强制操作。不再强制未选择的所有强制地址。

强制变量时的注意事项

强制前，必须检查强制变量时的安全预防措施 (页 551)。



危险

防止人身伤害和物资损失！

请注意，执行“强制”功能时操作不当可能会：

- 危及人员生命或健康
 - 导致机器或整个设备损坏
-

要求

- 已创建监视表格。
- 存在与 CPU 的在线连接。
- 所用的 CPU 支持强制功能。

步骤

要使用“在线 > 全部强制”(Online > Force all) 命令强制变量，请按以下步骤操作：

1. 打开监视表格并单击



图标 (显示/隐藏所有强制列) 以显示强制列。

2. 在监视表格中输入所需的地址和强制值。
3. 通过在“强制值”(Force value) 列后的列中选中用于设定强制的复选框，选择要强制的地址。
黄色三角形将出现在所选复选框后面，表示此时选择了强制该地址但尚未强制。
4. 选择“在线 > 强制 > 全部强制”(Online > Force > Force all) 命令以便使用指定的值强制所选的地址。
5. 单击“是”(Yes) 确认下一个对话框。

结果

所选的地址被强制为指定值。不再显示黄色三角形。红色的“F”显示在第一列中，表示变量处于强制状态。

停止强制

要停止强制，请按以下步骤操作：

1. 选择“在线 > 强制 > 停止强制”(Online > Force > Stop forcing) 命令。
2. 单击“是”(Yes) 确认下一个对话框。

结果

停止对所选地址的强制。第一列中的红色“F”不再显示。黄色三角形再次重新出现在相关复选框后面，指示此时选择了强制该地址但尚未强制。

注意

强制时，请注意以下说明：

- 强制无法撤消！
 - 终止在线连接不会停止强制！
 - 要停止强制，强制的地址必须在激活的监视表格中可见。必要时，使用“在线 > 强制 > 显示强制值”(Online > Force > Show force values) 命令在激活的监视表格中显示所有被强制的地址。
-

7.5.3.10 显示强制值

简介

可以使用强制功能给用户程序中的各个变量分配永久值。

您可以在一个或多个监视表格中定义这些变量。“显示强制值”(Show force values) 命令可用于确保能够在监视表格中显示 CPU 中被强制的所有变量。只有变量在一个以上的监视表格中被强制时，该命令才可用。如果变量仅在一个监视表格中被强制，则该命令不会激活。

强制变量时的注意事项

强制前，必须检查强制变量时的安全预防措施 (页 551)。



危险 防止人身伤害和物资损失！

请注意，执行“强制”功能时操作不当可能会：

- 危及人员生命或健康
 - 导致机器或整个设备损坏
-

要求

- 已定义要在其中强制变量的监视表格。
- 存在与 CPU 的在线连接。
- 所用的 CPU 支持强制功能。

步骤

要在监视表格中显示 CPU 中被强制的所有变量，请按以下步骤操作：

1. 打开一个现有的要在其中强制变量的监视表格。
2. 单击



图标 (显示/隐藏所有强制列) 以显示强制列。

3. 选择“在线 > 强制 > 显示强制值”(Online > Force > Show force values) 命令在当前监视表格中显示 CPU 中被强制的所有变量。

结果

在打开的监视表格中显示 CPU 中被强制的所有变量。其它监视表格中定义的变量插入在当前监视表格的底部。红色的“F”显示在第一列中，表示变量处于强制状态。

停止强制

要停止强制，请按以下步骤操作：

1. 选择“在线 > 强制 > 停止强制”(Online > Force > Stop forcing) 命令。

- 单击“是”(Yes) 确认“停止强制”(Stop forcing) 对话框。

结果

停止对所选值的强制。第一列中的红色“F”不再显示。黄色三角形再次重新出现在相关复选框后面，指示此时选择了强制该地址但尚未强制。

注意

强制时，请注意以下说明：

- 强制无法撤消！
 - 终止在线连接不会停止强制！
 - 要停止强制，强制的地址必须在激活的监视表格中可见。必要时，使用“在线 > 强制 > 显示强制值”(Online > Force > Show force values) 命令在激活的监视表格中显示所有被强制的地址。
-

7.5.3.10 停止强制

简介

停止强制变量前请注意下列说明：

- 强制无法撤消！
- 终止在线连接不会停止强制！
- 要停止强制，强制的地址必须在激活的监视表格中可见。如果想要同时停止强制所有变量，则必要时可使用“在线 > 强制 > 显示强制值”(Online > Force > Show force values) 命令在激活的监视表格中显示所有被强制的地址。

强制变量时的注意事项

强制前，必须检查强制变量时的安全预防措施 (页 551)。



危险

防止人身伤害和物资损失！

请注意，执行“强制”功能时操作不当可能会：

- 危及人员生命或健康
 - 导致机器或整个设备损坏
-

要求

- 已创建要在其中强制变量的监视表格。
- 存在与 CPU 的在线连接。
- 所用的 CPU 支持强制功能。

步骤

要停止强制变量，请按以下步骤操作：

1. 打开监视表格并单击



图标（显示/隐藏所有强制列）以显示强制列。

2. 使用“在线 > 强制 > 显示强制值”(Online > Force > Show force values) 命令显示 CPU 中被强制的所有地址。
3. 选择“在线 > 强制 > 停止强制”(Online > Force > Stop forcing) 命令以停止强制所显示的地址。
4. 单击“是”(Yes) 确认“停止强制”(Stop forcing) 对话框。

结果

停止对所选值的强制。第一列中的红色“F”不再显示。黄色三角形再次重新出现在相关复选框后面，指示此时选择了强制该地址但尚未强制。

7.6 编程实例

7.6.1 LAD 编程实例

7.6.1.1 编程实例概述

实际应用

“编程实例”一章通过实际应用加强您对各 LAD 指令的理解。为此，提供以下自动化任务：

- [使用位逻辑指令控制传送带 \(页 559\)](#)
- [使用位逻辑指令检测传送带方向 \(页 560\)](#)
- [使用计数器和比较指令检测存储区域的填充量 \(页 562\)](#)
- [使用数学函数计算方程式 \(页 564\)](#)
- [使用定时器和位逻辑指令控制室温 \(页 565\)](#)

使用的指令

助记符	指令	描述
--- --- (页 617)	位逻辑指令	常开触点
--- /--- (页 618)	位逻辑指令	常闭触点
--- NOT --- (页 619)	位逻辑指令	取反逻辑运算结果
---()--- (页 620)	位逻辑指令	输出线圈
---(R)--- (页 622)	位逻辑指令	复位输出
---(S)--- (页 623)	位逻辑指令	置位输出
--- P --- (页 628)	位逻辑指令	查询操作数的信号上升沿
CTUD (页 644)	计数器	加计数和减计数
CMP >= (页 652)	比较器	查询第一个比较值是否大于或等于第二个比较值。
CMP < (页 657)	比较器	查询第一个比较值是否小于第二个比较值。
ADD (页 663)	数学函数	加
MUL (页 666)	数学函数	乘
DIV (页 668)	数学函数	除
TP (页 635)	定时器指令	生成脉冲

7.6.1.2 控制传送带实例

控制传送带

下图所示为可以用电气方式激活的传送带。在传送带的开始处有两个按钮开关：S1 用于启动，S2 用于停止。在传送带的末端还有两个按钮开关：S3 用于启动，S4 用于停止。从任意一端都可启动或停止传送带。

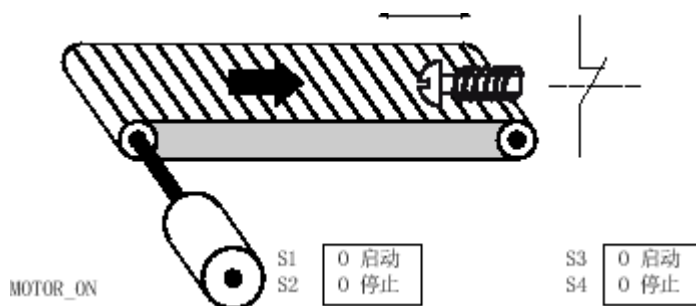


图 7-2

实施

下表显示了所用变量的定义：

名称	数据类型	注释
StartSwitch_Left	布尔型	传送带左侧的启动开关
StopSwitch_Left	布尔型	传送带左侧的停止开关
StartSwitch_Right	布尔型	传送带右侧的启动开关
StopSwitch_Right	布尔型	传送带右侧的停止开关
MOTOR_ON	布尔型	启动传送带电机

以下程序段显示了为解决该任务而编写的 LAD 程序：

程序段 1：启动开关“S1”或“S3”按下时，传送带电机接通。

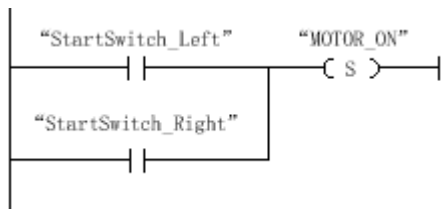


图 7-2

程序段 2：停止开关“S2”或“S4”按下时，传送带电机关闭。

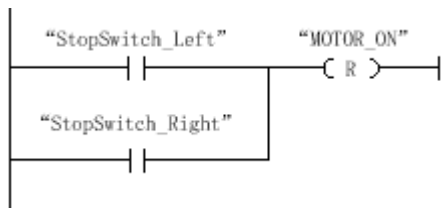


图 7-2

参见

- (R)---：复位输出 (页 622)
- (S)---：置位输出 (页 623)
- |/---：常开触点 (页 617)

7.6.1.3 检测传送带方向实例

检测传送带方向

下图所示为配有两个光电垒 (PEB1 和 PEB2) 的传送带。它们用于检测包裹在传送带上的移动方向。

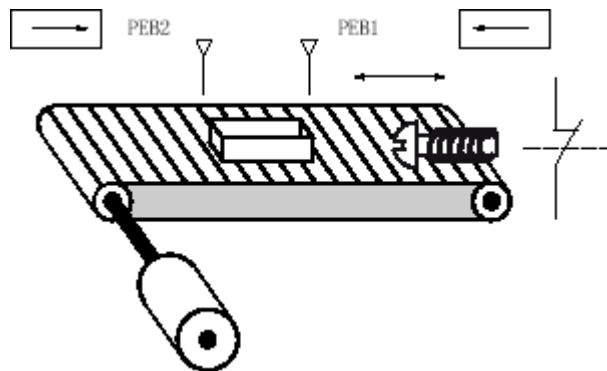


图 7-2

实施

下表显示了所用变量的定义：

名称	数据类型	注释
PEB1	布尔型	光电垒 1
PEB2	布尔型	光电垒 2
RIGHT	布尔型	向右移动时显示
LEFT	布尔型	向左移动时显示
CM1	布尔型	沿位存储器 1
CM2	布尔型	沿位存储器 2

以下程序段显示了为解决该任务而编写的 LAD 程序：

程序段 1：如果“PEB1”处的信号从“0”变为“1”(上升沿)并且此时“PEB2”的信号状态为“0”，则传送带上的包裹向左移动。

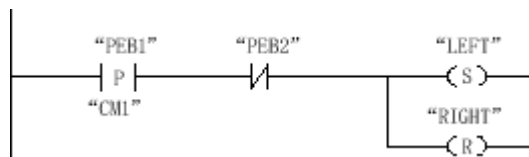


图 7-2

程序段 2：如果“PEB2”处的信号从“0”变为“1”(上升沿)并且此时“PEB1”的信号状态为“0”，则传送带上的包裹向右移动。

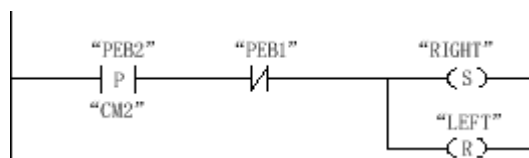


图 7-2

参见

- (R)--- : 复位输出 (页 622)
- (S)--- : 置位输出 (页 623)
- [P]-- : 扫描操作数的信号上升沿 (页 628)
- /|--- : 常闭触点 (页 618)

7.6.1.4 检测存储区域的填充量实例

检测存储区域的填充量

下图显示的系统中包含两条传送带和一个临时存储区域，而临时存储区域位于两条传送带之间。传送带 1 将包裹传送到存储区域。传送带 1 末端靠近存储区域的光电垒负责检测传送到存储区域的包裹数量。传送带 2 将包裹从临时存储区域传输到装载台，卡车从此处取走包裹并传送给用户。存储区域出口处的光电垒负责检测离开存储区域进入装载台的包裹数量。显示面板上的五个灯用于指示临时存储区域的利用情况。

重新启动传送带时，当前计数值将被设置为存储区域内现有的包裹数量。

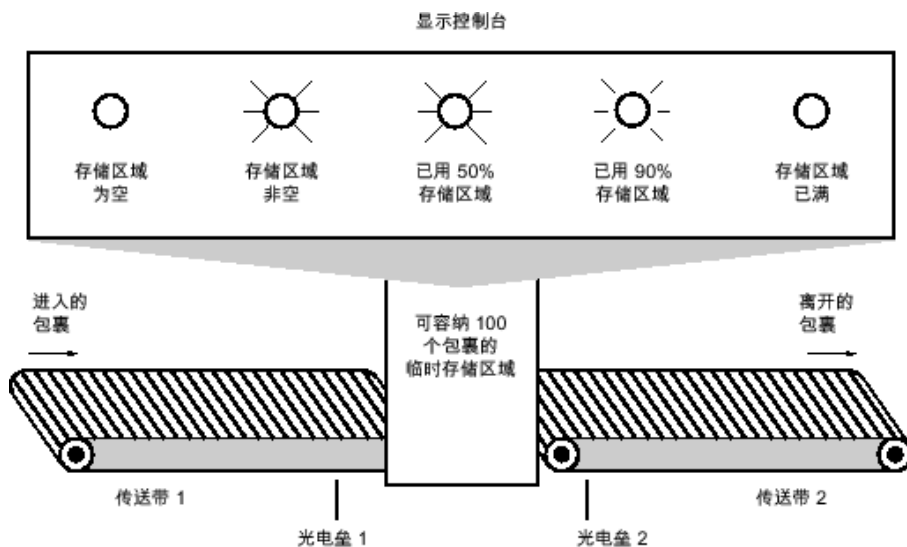


图 7-2

实施

下表显示了所用变量的定义：

名称	数据类型	注释
PEB1	BOOL	光电垒 1
PEB2	BOOL	光电垒 2
RESET	BOOL	重置计数器
LOAD	BOOL	将计数器设置为“PV”参数的值
STOCK	UINT	重新启动时的库存

名称	数据类型	注释
PACKAGECOUNT	UINT	存储区域中的包裹数 (当前计数值)
STOCK_PACKAGE	BOOL	当前计数值大于或等于变量“STOCK”的值时置位。
STOR_EMPTY	BOOL	指示灯：存储区域为空
STOR_NOT_EMPTY	BOOL	指示灯：存储区域非空
STOR_50%_FULL	BOOL	指示灯：已用 50% 存储区域
STOR_90%_FULL	BOOL	指示灯：已用 90% 存储区域
STOR_FULL	BOOL	指示灯：存储区域满
VOLUME_50	UINT	比较值：50 个包裹
VOLUME_90	UINT	比较值：90 个包裹
VOLUME_100	UINT	比较值：100 个包裹

以下程序段显示了为激活指示灯而编写的 LAD 程序：

程序段 1：包裹传递到存储区域时，“PEB1”的信号状态从“0”切换到“1”(上升沿)。 “PEB1”上有上升沿时，启用“加”计数器，“PACKAGECOUNT”的当前计数值增加 1。

包裹从存储区域传递到装载台时，“PEB2”的信号状态从“0”切换到“1”(上升沿)。 “PEB2”上有上升沿时，启用“减”计数器，“PACKAGECOUNT”的当前计数值减少 1。

只要存储区域中没有包裹 (“PACKAGECOUNT”=“0”)，“STOR_EMPTY”变量的信号状态即会设置为“1”且“存储区域为空”指示灯接通。

“RESET”变量的信号状态置位为“1”时，会将当前计数值复位为“0”。

如果“LOAD”变量的信号状态置位为“1”，则会将当前计数值设置为“STOCK”变量的值。 如果当前计数值大于或等于“STOCK”变量的值，则“STOCK_PACKAGE”变量的信号状态为“1”。

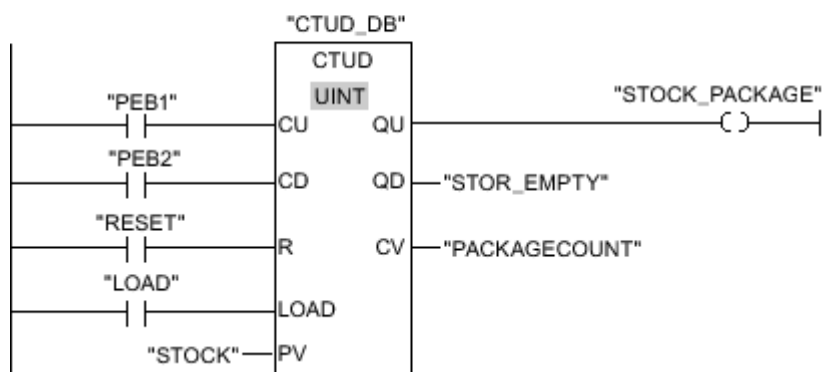


图 7-2

程序段 2：只要存储区域中有包裹，“STOR_NOT_EMPTY”变量的信号状态即会设置为“1”且“存储区域非空”指示灯接通。



图 7-2

程序段 3：如果存储区域中的包裹数大于或等于 50，则“已用 50% 存储区域”指示灯将接通。

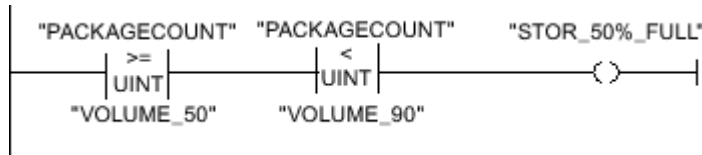
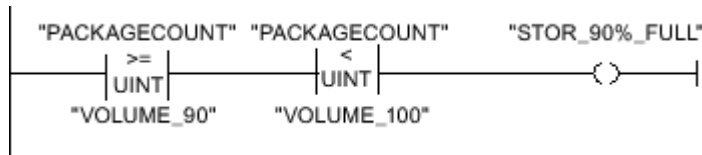
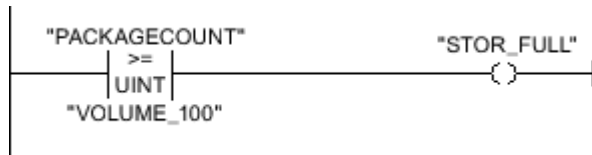


图 7-2

程序段 4：如果存储区域中的包裹数大于或等于 90，则“已用 90% 存储区域”指示灯将接通。



程序段 5：如果存储区域中的包裹数达到 100，则“存储区域满”指示灯将接通。



参见

- ()---: 输出线圈 (页 620)
- CTUD : 加计数和减计数 (IEC) (页 644)
- CMP >= : 大于或等于 (页 652)
- |/---: 常开触点 (页 617)
- |/|---: 常闭触点 (页 618)

7.6.1.5 计算方程式实例

计算方程式

此实例程序显示如何使用三个数学函数来产生与下列方程式相同的结果：

$$RESULT = ((A + B) \times 15) / E$$

实施

下表显示了所用变量的定义：

名称	数据类型	注释
A	INT	被加数
B	INT	加数
C	INT	第一个中间结果
15	INT	乘数
D	INT	第二个中间结果
E	INT	除数
RESULT	INT	最终结果

以下程序段显示了为计算方程式而编写的 LAD 程序：

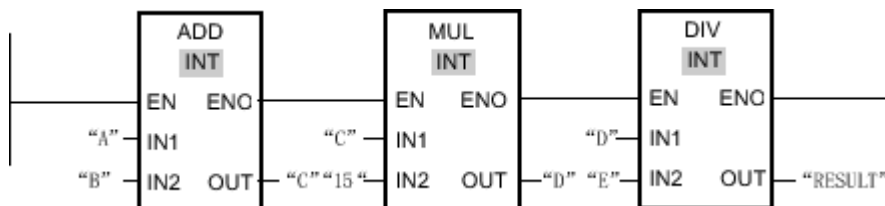


图 7-2

操作数“A”的值与操作数“B”的值相加。和存储在操作数“C”中。操作数“C”的值乘以“15”。乘法结果存储在操作数“D”中。操作数“D”中存储的值然后除以操作数“E”的值。最终结果存储在操作数“RESULT”中。

参见

ADD : 加 (页 663)

MUL : 乘 (页 666)

DIV : 除 (页 668)

7.6.1.6 控制室温实例

控制室温

在冷库中，温度必须维持在零摄氏度以下。通过传感器监视温度波动。如果温度上升到零摄氏度以上，制冷系统将在预定的时间内接通。在此期间，“制冷系统接通”指示灯一直亮。

满足下列条件之一时，制冷系统和指示灯将关闭：

- 传感器报告温度回落到零摄氏度以下。
- 预设的制冷时间已过。
- 已按下按钮开关“停止”。

如果预设的制冷时间已过，但冷库中的温度仍然过高，则可以按下按钮开关“复位”重新启动制冷系统。

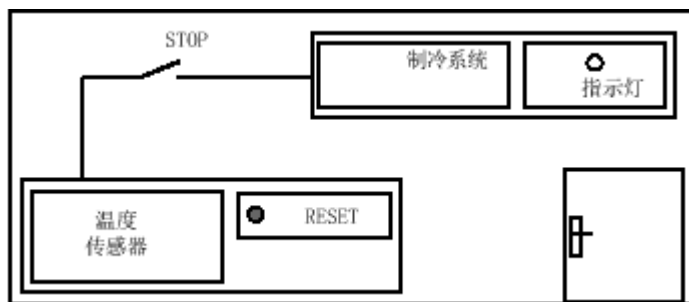


图 7-2

实施

下表显示了所用变量的定义：

名称	数据类型	注释
Sensor	BOOL	温度传感器信号
Reset	BOOL	重新启动
Stop	BOOL	制冷系统关闭。
MaxCoolTime	TIME	预定的制冷时间 该变量在“DB_Cool”数据块中定义。
CurrCoolTime	TIME	当前所耗的制冷时间 该变量在“DB_Cool”数据块中定义。
CoolSystem	BOOL	制冷系统接通。
Lamp	BOOL	“制冷系统接通”指示灯接通。

以下程序段显示了为控制室温而编写的 LAD 程序：

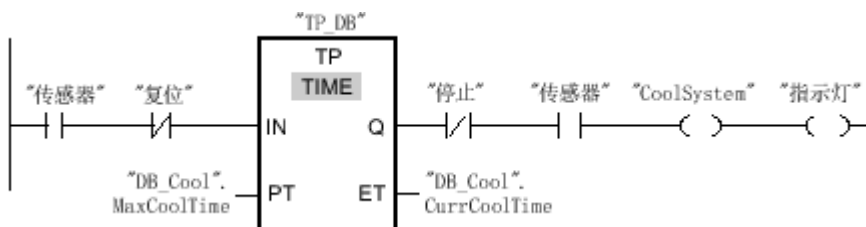


图 7-2

当冷库中的温度上升到零摄氏度以上时，“传感器”操作数的信号状态从“0”切换到“1”(上升沿)。在定时器功能的“IN”输入端有上升沿时，预定的制冷时间将启动，并且制冷系统和指示灯接通。如果冷库中的温度回落到零摄氏度以下，传感器的信号状态将切换回“0”。这将关闭制冷系统和指示灯。

如果传感器没有发出温度下降的信号，制冷系统和指示灯最迟在预定的制冷时间用完后关闭。在这种情况下，可以按下按钮开关“重新启动”重新启动制冷过程。按下按钮然后松开该按钮会在“IN”输入端产生新的上升沿，这将重新启动制冷系统。

可以随时按下按钮开关“停止”关闭制冷系统和指示灯。

参见

---()---: 输出线圈 (页 620)
 ---|/---: 常开触点 (页 617)
 ---|/|---: 常闭触点 (页 618)
 --|NOT|--: 取反逻辑运算结果 (页 619)
 TON: 接通延迟 (IEC) (页 637)

7.6.2 FBD 编程实例

7.6.2.1 编程实例概述

实际应用

“编程实例”一章通过实际应用加强您对各 FBD 指令的理解。为此，提供以下自动化任务：

- 使用位逻辑指令控制传送带 (页 568)
- 使用位逻辑指令检测传送带方向 (页 569)
- 使用计数器和比较指令检测存储区域的填充量 (页 570)
- 使用数学函数计算方程式 (页 573)
- 使用定时器和位逻辑指令控制室温 (页 575)

使用的指令

助记符	指令	描述
& (页 742)	位逻辑指令	AND 逻辑运算
>=1 (页 743)	位逻辑指令	OR 逻辑运算
= (页 749)	位逻辑指令	赋值
	位逻辑指令	取反二进制输入
R (页 751)	位逻辑指令	复位输出
S (页 752)	位逻辑指令	置位输出
P (页 757)	位逻辑指令	查询操作数的信号上升沿
CTUD (页 773)	计数器	加计数和减计数

助记符	指令	描述
CMP >= (页 780)	比较器	查询第一个比较值是否大于或等于第二个比较值。
CMP < (页 785)	比较器	查询第一个比较值是否小于第二个比较值。
ADD (页 791)	数学函数	加
MUL (页 794)	数学函数	乘
DIV (页 795)	数学函数	除
TP (页 764)	定时器指令	生成脉冲

参见

取反二进制值 (页 748)

7.6.2.2 控制传送带实例

控制传送带

下图所示为可以用电气方式激活的传送带。在传送带的开始处有两个按钮开关：S1 用于启动，S2 用于停止。在传送带的末端还有两个按钮开关：S3 用于启动，S4 用于停止。从任意一端都可启动或停止传送带。

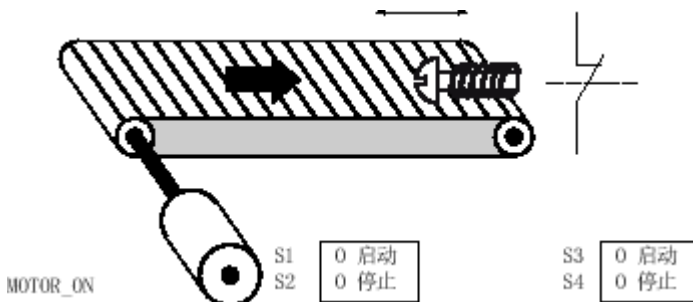


图 7-2

实施

下表显示了所用变量的定义：

名称	数据类型	注释
StartSwitch_Left	布尔型	传送带左侧的启动开关
StopSwitch_Left	布尔型	传送带左侧的停止开关
StartSwitch_Right	布尔型	传送带右侧的启动开关
StopSwitch_Right	布尔型	传送带右侧的停止开关
MOTOR_ON	布尔型	启动传送带电机

以下程序段显示了为完成该任务而编写的 FBD 程序：

程序段 1：启动开关“S1”或“S3”按下时，传送带电机接通。



图 7-2

程序段 2：停止开关“S2”或“S4”按下时，传送带电机关闭。



图 7-2

参见

- R*：复位输出 (页 751)
- S*：置位输出 (页 752)
- OR* 逻辑运算 (页 743)

7.6.2.3 检测传送带方向实例

检测传送带方向

下图所示为配有两个光电垒 (PEB1 和 PEB2) 的传送带。它们用于检测包裹在传送带上的移动方向。

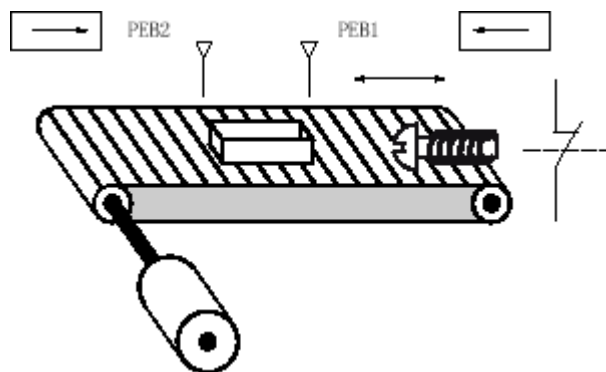


图 7-2

实施

下表显示了所用变量的定义：

名称	数据类型	注释
PEB1	布尔型	光电垒 1

名称	数据类型	注释
PEB2	布尔型	光电垒 2
RIGHT	布尔型	向右移动时显示
LEFT	布尔型	向左移动时显示
CM1	布尔型	沿位存储器 1
CM2	布尔型	沿位存储器 2

以下程序段显示了为解决该任务而编写的 FBD 程序：

程序段 1：如果“PEB1”处的信号从“0”变为“1”(上升沿) 并且此时“PEB2”的信号状态为“0”，则传送带上的包裹向左移动。

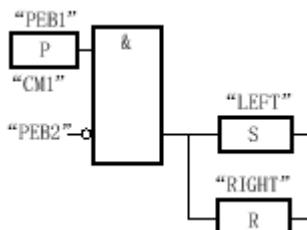


图 7-2

程序段 2：如果“PEB2”的信号状态从“0”变为“1”(上升沿) 并且此时“PEB1”的信号状态为“0”，则传送带上的包裹向右移动。

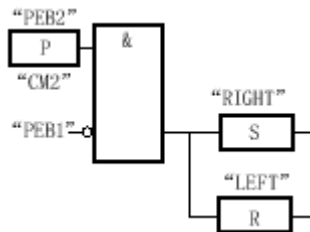


图 7-2

参见

- R：复位输出 (页 751)
- S：置位输出 (页 752)
- P：扫描操作数的信号上升沿 (页 757)
- 取反二进制值 (页 748)
- AND 逻辑运算 (页 742)

7.6.2.4 检测存储区域的填充量实例

检测存储区域的填充量

下图显示的系统中包含两条传送带和一个临时存储区域，而临时存储区域位于两条传送带之间。传送带 1 将包裹传送到存储区域。传送带 1 末端靠近存储区域的光电垒负责检测传送到

存储区域的包裹数量。传送带 2 将包裹从临时存储区域传输到装载台，卡车从此处取走包裹并传送给用户。存储区域出口处的光电垒负责检测离开存储区域进入装载台的包裹数量。显示面板上的五个灯用于指示临时存储区域的利用情况。

重新启动传送带时，当前计数值将被设置为存储区域内现有的包裹数量。

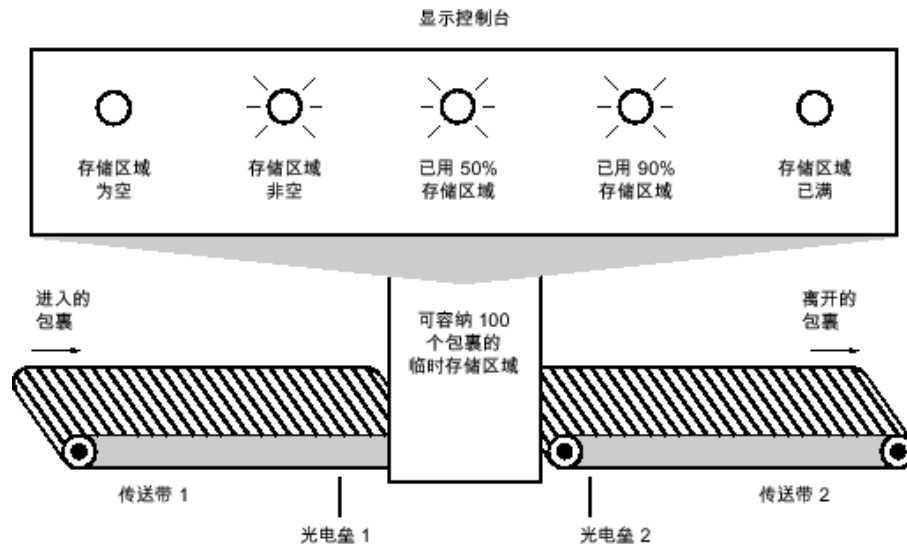


图 7-2

实施

下表显示了所用变量的定义：

名称	数据类型	注释
PEB1	BOOL	光电垒 1
PEB2	BOOL	光电垒 2
RESET	BOOL	重置计数器
LOAD	BOOL	将计数器设置为“CV”参数的值
STOCK	UINT	重新启动时的库存
PACKAGECOUNT	UINT	存储区域中的包裹数（当前计数值）
STOCK_PACKAGE	BOOL	当前计数值大于或等于变量“STOCK”的值时置位。
STOR_EMPTY	BOOL	指示灯：存储区域为空
STOR_NOT_EMPTY	BOOL	指示灯：存储区域非空
STOR_50%_FULL	BOOL	指示灯：已用 50% 存储区域
STOR_90%_FULL	BOOL	指示灯：已用 90% 存储区域
STOR_FULL	BOOL	指示灯：存储区域满
VOLUME_50	UINT	比较值：50 个包裹

名称	数据类型	注释
VOLUME_90	UINT	比较值：90 个包裹
VOLUME_100	UINT	比较值：100 个包裹

以下程序段显示了为激活指示灯而编写的 FBD 程序：

程序段 1：包裹传递到存储区域时，“PEB1”的信号状态从“0”切换到“1”(上升沿)。“PEB1”上有上升沿时，启用“加”计数器，“PACKAGECOUNT”的当前计数值增加 1。

包裹从存储区域传递到装载台时，“PEB2”的信号状态从“0”切换到“1”(上升沿)。“PEB2”上有上升沿时，启用“减”计数器，“PACKAGECOUNT”的当前计数值减少 1。

只要存储区域中没有包裹 (“PACKAGECOUNT”=“0”)，“STOR_EMPTY”变量的信号状态即会设置为“1”且“存储区域为空”指示灯接通。

“RESET”变量的信号状态置位为“1”时，会将当前计数值复位为“0”。

如果“LOAD”变量的信号状态置位为“1”，则会将当前计数值设置为“STOCK”变量的值。如果当前计数值大于或等于“STOCK”变量的值，则“STOCK_PACKAGE”变量的信号状态为“1”。

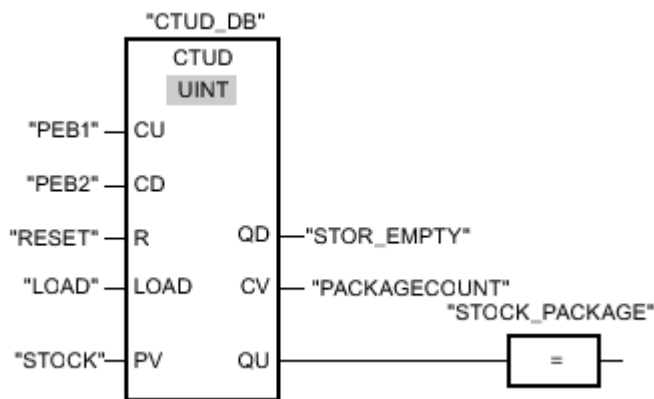


图 7-2

程序段 2：只要存储区域中有包裹，“STOR_NOT_EMPTY”变量的信号状态即会设置为“1”且“存储区域非空”指示灯接通。

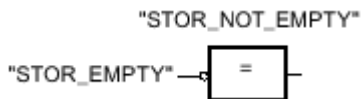


图 7-2

程序段 3：如果存储区域中的包裹数大于或等于 50，则“已用 50 % 存储区域”指示灯将接通。

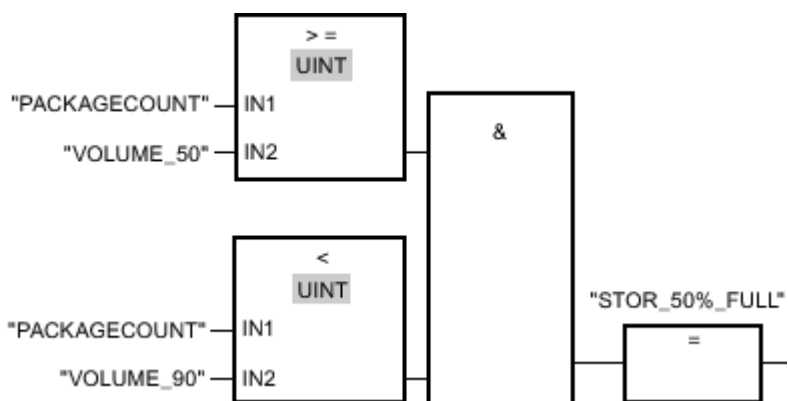


图 7-2

程序段 4：如果存储区域中的包裹数大于或等于 90，则“已用 90% 存储区域”指示灯将接通。

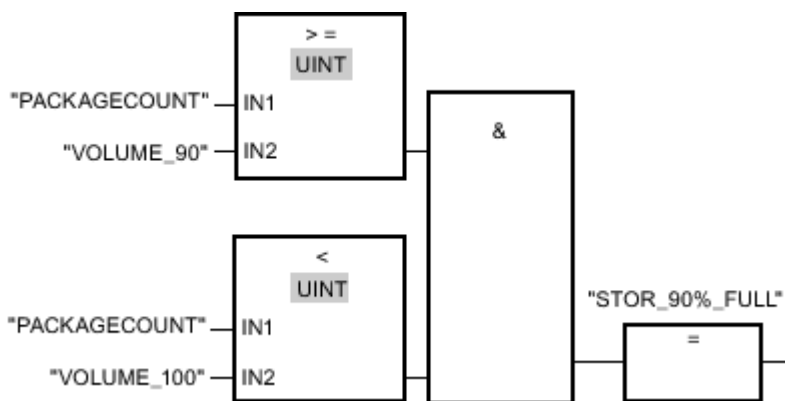
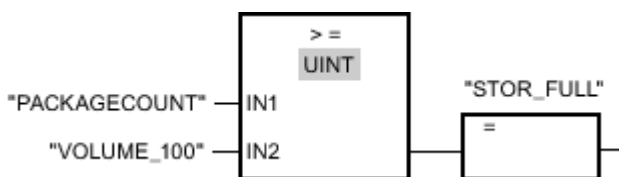


图 7-2

程序段 5：如果存储区域中的包裹数达到 100，则“存储区域满”指示灯将接通。



参见

- := 赋值 (页 749)*
- CTUD : 加计数和减计数 (IEC) (页 773)*
- CMP >= : 大于或等于 (页 780)*
- 取反二进制值 (页 748)*
- OR 逻辑运算 (页 743)*

7.6.2.5 计算方程式实例

计算方程式

此实例程序显示如何使用三个数学函数来产生与下列方程式相同的结果：

$$\text{RESULT} = ((A + B) \times 15) / E$$

实施

下表显示了所用变量的定义：

名称	数据类型	注释
A	INT	被加数
B	INT	加数
C	INT	第一个中间结果
15	INT	乘数
D	INT	第二个中间结果
E	INT	除数
RESULT	INT	最终结果

以下程序段显示了为计算方程式而编写的 FBD 程序：

程序段 1：操作数“A”的值与操作数“B”的值相加。和存储在操作数“C”中。操作数“C”的值乘以“15”。乘法结果存储在操作数“D”中。操作数“D”中存储的值然后除以操作数“E”的值。最终结果存储在操作数“RESULT”中。

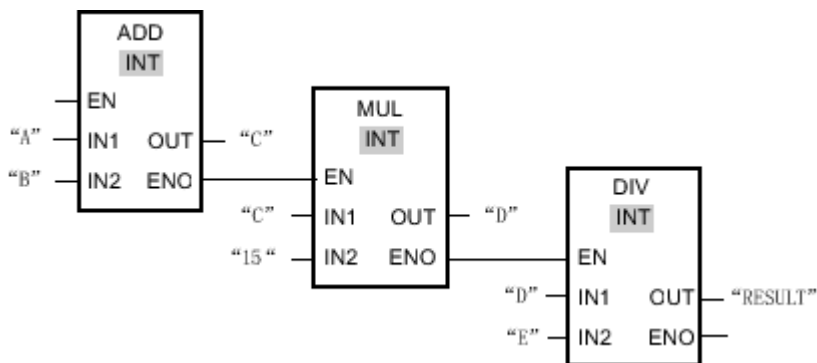


图 7-2

参见

ADD：加 (页 791)

MUL：乘 (页 794)

DIV：除 (页 795)

7.6.2.6 控制室温实例

控制室温

在冷库中，温度必须维持在零摄氏度以下。通过传感器监视温度波动。如果温度上升到零摄氏度以上，制冷系统将在预定的时间内接通。在此期间，“制冷系统接通”指示灯一直亮。

满足下列条件之一时，制冷系统和指示灯将关闭：

- 传感器报告温度回落到零摄氏度以下。
- 预设的制冷时间已过。
- 已按下按钮开关“停止”。

如果预设的制冷时间已过，但冷库中的温度仍然过高，则可以按下按钮开关“复位”重新启动制冷系统。

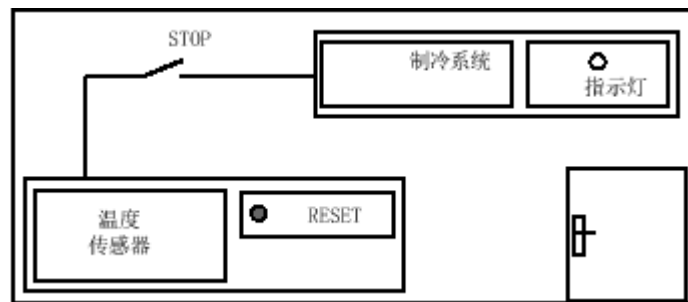


图 7-2

实施

下表显示了所用变量的定义：

名称	数据类型	注释
Sensor	BOOL	温度传感器信号
Reset	BOOL	重新启动
Stop	BOOL	制冷系统关闭。
MaxCoolTime	TIME	预设的制冷时间 该变量在“DB_Cool”数据块中定义。
CurrCoolTime	TIME	当前所耗的制冷时间 该变量在“DB_Cool”数据块中定义。
CoolSystem	BOOL	制冷系统接通。
Lamp	BOOL	“制冷系统接通”指示灯接通。

以下程序段显示了为控制室温而编写的 FBD 程序：

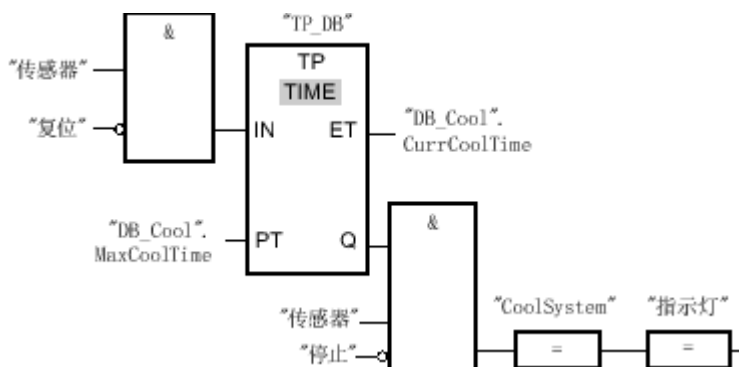


图 7-2

当冷库中的温度上升到零摄氏度以上时，“传感器”操作数的信号状态从“0”切换到“1”（上升沿）。在定时器功能的“IN”输入端有上升沿时，预定的制冷时间将启动，并且制冷系统和指示灯接通。如果冷库中的温度回落到零摄氏度以下，传感器的信号状态将切换回“0”。这将关闭制冷系统和指示灯。

如果传感器没有发出温度下降的信号，制冷系统和指示灯最迟在预定的制冷时间用完后关闭。在这种情况下，可以按下按钮开关“重新启动”重新启动制冷过程。按下按钮然后松开该按钮会在“IN”输入端产生新的上升沿，这将重新启动制冷系统。

可以随时按下按钮开关“停止”关闭制冷系统和指示灯。

参见

TONR：保持型接通延迟 (IEC) (页 768)

=：赋值 (页 749)

取反二进制值 (页 748)

AND 逻辑运算 (页 742)

OR 逻辑运算 (页 743)

TON：接通延迟 (IEC) (页 765)

7.7 使用工艺对象

7.7.1 使用运动控制

7.7.1.1 运动控制指南

“轴”工艺对象在控制器中创建一个轴映像，并且适用于控制步进电机和具有脉冲接口的伺服驱动器。“轴”工艺对象通过运动控制语句进行控制。在用户程序中可以调用以下命令（语句）：

- MC_Power：启用/禁用轴
- MC_Reset：确认错误
- MC_Home：使轴回原点，设置参考点

- MC_Halt : 停止轴
- MC_MoveAbsolute : 绝对定位轴
- MC_MoveRelative : 相对定位轴
- MC_MoveVelocity : 以速度预设值移动轴
- MC_MoveJog : 在点动模式下移动轴

要求

要使用“轴”工艺对象，必须创建具有 CPU S7-1200 的项目。

步骤

要使用“轴”工艺对象，请按以下步骤操作：

1. [添加工艺对象“轴” \(页 579\)](#)
2. [组态工艺对象 \(页 580\)](#)
3. [下载到 CPU \(页 596\)](#)
4. [在调试窗口中对轴执行功能测试 \(页 597\)](#)
5. [创建用户程序 \(页 587\)](#)
6. [对轴控制执行诊断 \(页 599\)](#)

参见

MC_Power : 启用、禁用轴 (页 930)
MC_Reset : 确认错误 (页 931)
MC_Home : 归位轴，设置归位位置 (页 932)
MC_Halt : 停止轴 (页 935)
MC_MoveAbsolute : 轴的绝对定位 (页 936)
MC_MoveRelative : 轴的相对定位 (页 937)
MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)
MC_MoveJog : 在点动模式下移动轴 (页 940)

7.7.1.2 轴 - 基本知识

7.7.1.2 轴和驱动器之间的区别

术语“轴”特指用“轴”工艺对象表示的驱动器工艺映像。“轴”工艺对象是用户程序与驱动器之间的接口。该工艺对象接收用户程序中的运动控制命令、执行这些命令并监视其运行情况。运动控制命令在用户程序中通过运动控制语句启动。

术语“驱动器”特指由步进电机与动力部分或伺服驱动器与具有脉冲接口的转换器组成的机电装置。驱动器由“轴”工艺对象通过 CPU S7-1200 的脉冲发生器控制。

参见

MC_Power : 启用、禁用轴 (页 930)
MC_Reset : 确认错误 (页 931)
MC_Home : 归位轴, 设置归位位置 (页 932)
MC_Halt : 停止轴 (页 935)
MC_MoveAbsolute : 轴的绝对定位 (页 936)
MC_MoveRelative : 轴的相对定位 (页 937)
MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)
MC_MoveJog : 在点动模式下移动轴 (页 940)

7.7.1.2 回原点

位置控制轴的位置输入和显示均参考轴坐标系。轴坐标系必须与轴的实际物理位置同步 (回原点)。

注意

请注意以下事项：

- 在下列情况下轴的回原点状态会丢失：
 - CPU 存储器复位
 - 断电 -> 上电
 - CPU 重新启动
 - 在运动控制语句“MC_Power”中复位轴启用 (该状态下封锁轴)
- 在非回到原点状态下, 除了运动控制语句“MC_MoveAbsolute”外, 可以执行所有运动命令。

轴回原点由运动控制语句“MC_Home”启动。回原点期间, 参考点坐标设置在定义的轴机械位置处。

回原点模式

- **主动回原点**
在主动回原点模式下, 运动控制语句“MC_Home”执行所需要的参考点逼近。将取消其它所有激活的运动。
- **被动回原点**
在被动回原点模式下, 运动控制语句“MC_Home”不执行参考点逼近。不取消其它激活的运动。逼近参考点开关必须由用户通过运动控制语句或由机械运动执行。
- **绝对式直接回原点**
无论参考凸轮位置为何, 都设置轴位置。不取消其它激活的运动。立即激活“MC_Home”语句中的“Position”参数的值作为轴的参考点和位置值。轴必须处于停止状态才能将参考点准确分配到机械位置。
- **相对式直接回原点**

无论参考凸轮位置为何，都设置轴位置。不取消其它激活的运动。适用于参考点和轴位置的规则：

新的轴位置 = 当前轴位置 + “Position”参数的值。

立即激活参考点和轴位置的值。轴必须处于停止状态才能将参考点准确分配到机械位置。

参见

MC_Home : 归位轴, 设置归位位置 (页 932)

组态 - 回原点 (页 585)

7.7.1.3 添加工艺对象“轴”

要在项目浏览器中添加“轴”工艺对象，请按以下步骤操作：

要求

已创建具有 CPU S7-1200 的项目。

步骤

1. 在项目浏览器中打开 CPU 文件夹。
2. 打开工艺对象文件夹。
3. 双击“添加新对象”(Add new object) 对象。
将打开“添加新对象”(Add new object) 对话框。
4. 在“名称”(Name) 输入域中键入该工艺对象的专用名称。
5. 单击“轴”(Axis)。
已选择所需的“TO_AXIS_PTO”类型。
6. 如果想要更改推荐的数据块编号，则选择“手动”(Manual) 选项。
7. 如果想要为该工艺对象补充用户信息，则单击“更多信息”(Further information)。
8. 单击“确定”(OK) 以添加该工艺对象。单击“取消”(Cancel) 以放弃输入。

结果

创建了新工艺对象并保存在项目浏览器中的“工艺对象”(Technological objects) 文件夹中。

注意

可以选中该对话框底部的“添加并打开新块”(Add and open new block) 复选框。这将在添加操作完成后打开工艺对象的组态。

7.7.1.4 组态

7.7.1.4 组态工艺对象

在组态窗口中组态运动控制工艺对象的属性。要打开工艺对象的组态窗口，请按以下步骤操作：

1. 在项目浏览器中打开所选工艺对象组。
2. 双击“组态”(Configuration) 对象。

组态窗口的图标

组态的区域导航中的图标显示有关组态完成情况的更多信息：

✔	<p>组态包含默认值且已完成。 组态仅包含默认值。使用这些默认值即可使用工艺对象，无需进一步更改。</p>
✔	<p>组态包含自定义值且已完成 组态输入域中的所有值都有效且至少一个默认值被修改。</p>
✘	<p>组态不完整或不正确 至少一个输入域或下拉列表框未包含值或包含无效值。相应域或下拉列表框以红色背景显示。单击弹出错误消息可找出错误原因。</p>

参见

- 组态 - 常规 (页 580)
- 组态 - 驱动器信号 (页 582)
- 组态 - 机械 (页 582)
- 组态 - 位置监视 (页 583)
- 组态 - 常规动态响应 (页 584)
- 组态 - 动态急停 (页 584)
- 组态 - 回原点 (页 585)

7.7.1.4 基本参数

7.7.1.4 组态 - 常规

在“常规”(General) 组态窗口中组态“轴”(Axis) 工艺对象的基本属性。

轴名称 (Axis name)

在该域中定义轴和工艺对象的名称。“轴”工艺对象以该名称列出在项目浏览器中。

硬件接口 (Hardware interface)

工艺对象通过 PTO (Pulse Train Output, 脉冲串输出) 产生用于步进电机控制的脉冲。脉冲通过永久分配的数字输出输出到步进电机的动力部分。

对于具有集成继电器输出的 CPU, 无法输出脉冲信号, 因为继电器不支持该信号切换频率。要允许此类 CPU 使用 PTO, 必须安装具有数字输出的信号板。

注意

PTO 在内部需要高速计数器 (HSC, High-Speed Counter) 功能。用户无法以任何其它方式使用相应的高速计数器。

第一个 PTO 与第一个高速计数器互连; 第二个 PTO 与第二个高速计数器互连。

要组态所选的 PTO, 请按以下步骤操作:

1. 从“PTO 选择驱动器控制”(PTO selection drive control) 下拉列表框中选择“Pulse_1”或“Pulse_2”。
2. 单击“设备配置”(Device configuration)。
3. 在区域浏览器中, 打开“脉冲发生器”(Pulse generators) 组。
4. 在区域浏览器中, 单击所选 PTO。
将显示该 PTO 的组态。
5. 选中“启用该脉冲发生器”(Enable this pulse generator) 复选框。
6. 在“脉冲发生器作为”(Pulse generator as) 下拉列表框中选择“PTO”条目。
7. 转到“输出源”(Output source) 下拉列表框并选择 CPU 的集成输出或信号板的输出。

如果任何其它应用程序都未使用相应的高速计数器, 则“常规”(General) 轴组态中的 PTO 域不会以红色背景输出。否则, 根据错误消息更正组态。

自定义单位 (User-defined unit)

从下拉列表框中选择用于轴定位的测量系统。

注意事项

如果后来更改了该测量系统, 则值可能无法在所有组态窗口中都正确转换。在这种情况下, 检查所有轴参数的组态。

参见

组态工艺对象 (页 580)
组态 - 驱动器信号 (页 582)
组态 - 机械 (页 582)
组态 - 位置监视 (页 583)
组态 - 常规动态响应 (页 584)
组态 - 动态急停 (页 584)

组态 - 回原点 (页 585)

7.7.1.4 高级参数

7.7.1.4 组态 - 驱动器信号

在“驱动器信号”(Drive signals) 组态窗口中可以组态驱动器的启用信号和“准备就绪”反馈信号。如果驱动器不提供任何这类接口，则不必设置这些参数。

参见

组态工艺对象 (页 580)
组态 - 常规 (页 580)
组态 - 机械 (页 582)
组态 - 位置监视 (页 583)
组态 - 常规动态响应 (页 584)
组态 - 动态急停 (页 584)
组态 - 回原点 (页 585)

7.7.1.4 组态 - 机械

在“机械”(Mechanics) 组态窗口中组态驱动器的机械属性。

电机每转的脉冲数 (Pulses per motor revolution)

在该域中组态步进电机每转需要的脉冲数。

电机每转的距离 (Distance per motor revolution)

在该域中，组态电机每转带动单元的机械系统行进的距离。

取反方向感 (Inverting the sense of direction)

如果机械传动系统方向与步进电机的方向感相反，则选中“取反方向感”(Invert sense of direction) 复选框。

参见

组态工艺对象 (页 580)
组态 - 常规 (页 580)
组态 - 驱动器信号 (页 582)
组态 - 位置监视 (页 583)
组态 - 常规动态响应 (页 584)
组态 - 动态急停 (页 584)
组态 - 回原点 (页 585)

7.7.1.4 组态 - 位置监视

在“位置监视”(Position monitoring) 组态窗口中组态轴的软件和硬件限位开关。

激活硬件限位开关 (Activate hardware limit switches)

选中该复选框将激活上限和下限硬件限位开关。轴逼近或通过硬件限位开关后，将以急停减速度减速直到停止。回原点步骤 (页 585) 激活期间以急停减速度制动可以替换为到达硬件限位开关后轴反向。

激活软件限位开关 (Activate software limit switches)

选中该复选框将激活上限和下限软件限位开关。激活的运动在轴到达软件限位开关的位置后将停止。工艺对象报告故障。确认故障后，轴可以返回到其工作范围。

注意事项

激活软件限位开关设置仅影响回原点的轴。

下限/上限硬件限位开关输入 (Lower / upper HW limit switch input)

从下拉列表框中为下限或上限硬件限位开关选择数字输入。输入必须支持报警功能。集成的 CPU 输入和信号板的输入可以选作硬件限位开关的输入。

信号位选择 (Signal level selection)

选择硬件限位开关的触发信号位 (“下限位”(Lower level)/“上限位”(Upper level))。在“下限位”(Lower level) 时，输入信号在轴到达或通过硬件限位开关后为 FALSE。在“上限位”(Upper level) 时，输入信号在轴到达或通过硬件限位开关后为 TRUE。

下限/上限 SW 限位开关输入 (Lower / upper SW limit switch input)

在这些域中指定相应的软件限位开关的位置。

参见

组态工艺对象 (页 580)
组态 - 常规 (页 580)
组态 - 驱动器信号 (页 582)
组态 - 机械 (页 582)
组态 - 常规动态响应 (页 584)
组态 - 动态急停 (页 584)
组态 - 回原点 (页 585)

7.7.1.4 动态响应

7.7.1.4 组态 - 常规动态响应

在“常规动态响应”(General dynamic response) 组态窗口中，可以设置轴的最大速度、启动/停止速度和最大加速度/减速度。

速度限值的单位 (Unit for velocity limit)

从下拉列表框中选择速度限值的物理单位。

速度 (Velocity)

在这些域中定义轴的最大速度和启动/停止速度。

加速度/减速度 (Acceleration/deceleration)

在“加速时间”(Ramp-up time) 或“加速度”(Acceleration) 域中设置加速度值。在“减速时间”(Ramp-down time) 或“减速度”(Deceleration) 域中设置减速度值。

注意

对速度限值做的所有更改都影响轴的加速度/减速度值。加速时间和减速时间保持不变。

参见

组态工艺对象 (页 580)
组态 - 常规 (页 580)
组态 - 驱动器信号 (页 582)
组态 - 机械 (页 582)
组态 - 位置监视 (页 583)
组态 - 动态急停 (页 584)
组态 - 回原点 (页 585)

7.7.1.4 组态 - 动态急停

在“动态急停”(Dynamic emergency stop) 组态窗口中组态轴的最大急停减速度。发生错误时，轴以该减速度运行到停止。

速度 (Velocity)

“常规动态响应”(General dynamic response) 组态窗口中组态的速度值将再次显示在该信息区域中。

减速度 (Deceleration)

在“急停减速时间”(Emergency stop ramp-down time) 或“急停减速度”(Emergency stop deceleration) 域中设置急停的减速度限值。

参见

组态工艺对象 (页 580)

组态 - 常规 (页 580)

组态 - 驱动器信号 (页 582)

组态 - 机械 (页 582)

组态 - 位置监视 (页 583)

组态 - 常规动态响应 (页 584)

组态 - 回原点 (页 585)

7.7.1.4 组态 - 回原点

在“回原点”(Homing) 组态窗口中组态主动和被动回原点参数。激活的参考点逼近的回原点模式通过语句“MC_Home”Mode = 3 的方式设置；被动回原点通过 Mode = 2 的方式设置。

参考凸轮输入 (Reference cam input) (主动和被动回原点)

从下拉列表框中为参考点开关选择数字输入。输入必须支持报警功能。可以为参考点开关选择集成的 CPU 输入和信号板的输入。

允许在硬件限位开关处反向 (Allow reversal at the hardware limit switch) (仅限主动回原点)

激活该复选框可将硬件限位开关用作指示参考点逼近的反向凸轮。轴在主动回原点期间到达硬件限位开关后，将以设置的减速度减速，然后反向。然后反向检测参考点开关。如果该功能未激活且轴在主动回原点期间到达硬件限位开关，将因错误取消参考点逼近并按急停减速度斜坡停止轴。

逼近方向 (Approach direction) (仅限主动回原点)

在此指定用于检测参考点开关的逼近方向。

参考凸轮 (Reference cam) (主动和被动回原点)

选择参考点开关的左侧或右侧作为回原点位置。

注意

根据轴的起始位置，参考点逼近可能与组态窗口中的图中显示的顺序不同。

启动速度 (Startup velocity) (仅限主动回原点)

在该域中，指定参考点逼近期间检测参考点开关的速度。

逼近速度 (Approach velocity) (仅限主动回原点)

在该域中，指定轴逼近代表回原点的参考点开关的速度。

参考点位移 (Reference point shift) (仅限主动回原点)

参考点开关的位置和参考点位置不同时，在该域中输入相应的参考点位移。轴以逼近速度逼近回原点位置。

绝对参考点坐标在“MC_Home”语句的“位置”(Position) 参数中指定。

参见

组态工艺对象 (页 580)
组态 - 常规 (页 580)
组态 - 驱动器信号 (页 582)
组态 - 机械 (页 582)
组态 - 位置监视 (页 583)
组态 - 常规动态响应 (页 584)
组态 - 动态急停 (页 584)
回原点 (页 578)
MC_Home : 归位轴，设置归位位置 (页 932)

7.7.1.5 编程

7.7.1.5 运动控制语句概述

运动控制语句是用户程序和“轴”工艺对象间的接口。运动控制语句用于将命令传送到命令处理和监视的工艺对象。

可以从运动控制语句的输出参数和从项目浏览器的“诊断”中读取工艺对象的状态和错误事件。以下运动控制语句可用：

- MC Power : 启用、禁用轴 (页 930)
- MC Reset : 确认错误 (页 931)
- MC Home : 归位轴，设置归位位置 (页 932)
- MC Halt : 停止轴 (页 935)
- MC MoveAbsolute : 轴的绝对定位 (页 936)
- MC MoveRelative : 轴的相对定位 (页 937)
- MC MoveVelocity : 以预设的旋转速度移动轴 (页 939)
- MC MoveJog : 在点动模式下移动轴 (页 940)

7.7.1.5 创建用户程序

以下部分介绍了如何创建具有用于轴控制的基本组态的用户程序。要插入的运动控制语句用于控制轴的所有运动。

要求

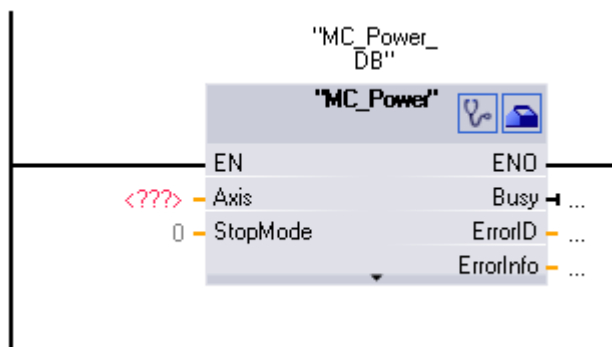
- 已创建并组态工艺对象且没有错误。

创建和测试用户程序前，建议使用控制面板(页 597)测试轴和驱动器功能。

步骤

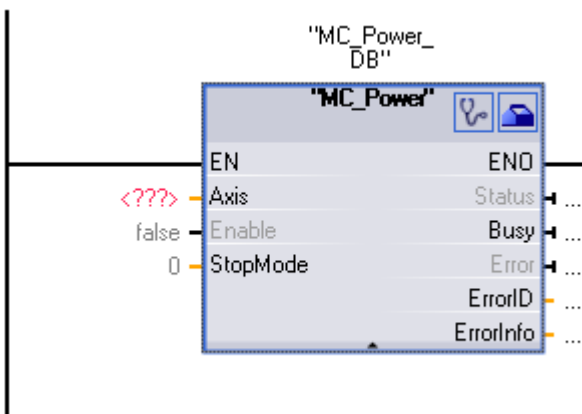
1. 在项目浏览器中，双击程序块（程序块必须在循环程序中调用）。
块在程序编辑器中打开并显示所有可用语句。
2. 打开高级语句和“运动控制”(Motion Control) 文件夹。
3. 将“MC_Power”语句拖放至块中选择的程序段中。
将打开用于定义背景数据块的对话框。
4. 单击“单个背景”(Single instance) 并选择是自动还是手动定义背景数据块编号。
5. 单击“确定”(OK)。

运动控制语句“MC_Power”插入到该程序段中。



程序编辑器提供以下控制选项：

如果想要打开工艺对象的诊断对话框，则单击听诊器形图标。单击文本框图标可打开工艺对象的组态窗口。单击下箭头图标可查看运动控制语句的其它参数。



必须组态所有标有“<???”的参数。黑体显示的参数是实际使用运动控制语句需要的。灰显的参数是可选的。

6. 双击 <???” 并从地址列表中选择“轴”(Axis) 工艺对象。
7. 根据步骤 3 到 6 将运动控制语句“MC_Reset”、“MC_Home”、“MC_Halt”、“MC_MoveAbsolute”、“MC_MoveRelative”、“MC_MoveVelocity”和“MC_MoveJog”插入其它程序段。

结果

已在用户程序中创建用于轴控制的基本组态。在用户程序的其它部分，控制运动控制语句的参数值。有关参数的其它信息，请参见运动控制语句说明。

参见

MC_Power : 启用、禁用轴 (页 930)
MC_Reset : 确认错误 (页 931)
MC_Home : 归位轴，设置归位位置 (页 932)
MC_Halt : 停止轴 (页 935)
MC_MoveAbsolute : 轴的绝对定位 (页 936)
MC_MoveRelative : 轴的相对定位 (页 937)
MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)
MC_MoveJog : 在点动模式下移动轴 (页 940)
调试 (页 597)

7.7.1.5 监视激活的命令

通过 Busy、Done、InVelocity 和 CommandAborted 输出参数监视运动控制语句的命令状态。

运动控制命令激活时：输出参数 Busy 的值为 TRUE，而在命令完成后，Busy 的值为 FALSE。Done、InVelocity 和 CommandAborted 输出参数指示状态，至少持续一个周期。只要输入参数 Execute 设置为 TRUE，就在保持模式下显示这些状态消息。

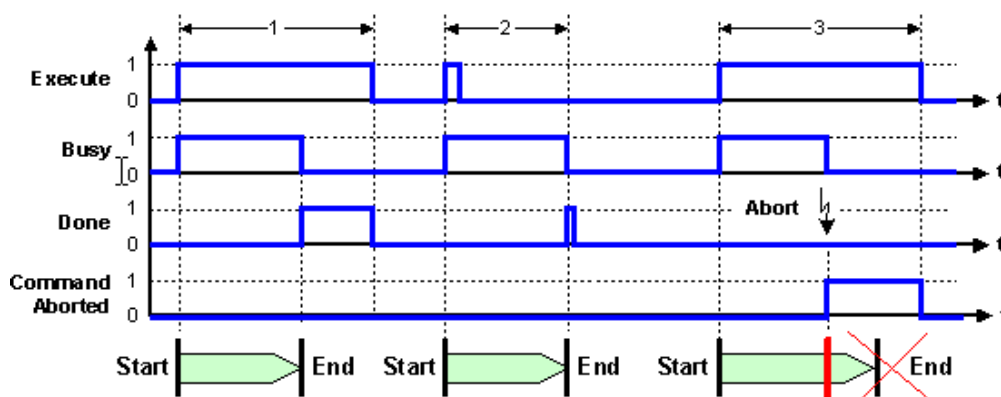
具有输出参数 Done 的运动控制语句

以下运动控制语句的命令具有定义的结论：

- MC_Reset
- MC_Home
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative

输出参数 Done 返回值 TRUE，以表明成功完成该命令。

以下部分基于三个实例阶段说明了命令状态的行为：



阶段 1

- 输入参数 Execute = TRUE 的上升沿时启动命令。
- 命令执行期间输出参数显示 Busy = TRUE.
- 命令执行完后，输出参数 Busy 变为 FALSE 且输出参数 Done 变为 TRUE。
- 如果 Execute = FALSE，输出参数 Done 变为 FALSE。

阶段 2

- 输入参数 Execute 的上升沿时启动命令。
- 命令执行期间输出参数显示 Busy = TRUE.
- 命令执行完后，输出参数 Busy 变为 FALSE. 且输出参数 Done 变为 TRUE。

阶段 3

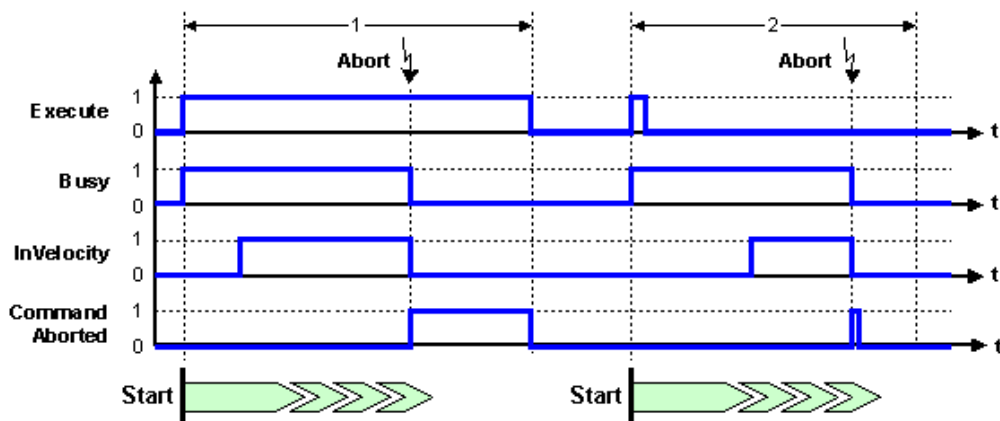
- 输入参数 Execute = TRUE 的上升沿时启动命令。
- 命令执行期间输出参数显示 Busy = TRUE.
- 命令执行被另一个命令取消。
- 取消命令执行后，输出参数 CommandAborted 从 FALSE 变为 TRUE。
- 命令执行完后，输出参数 Busy 变为 FALSE。
- 如果 Execute = FALSE，输出参数 CommandAborted 变为 FALSE。

具有输出参数 Done 的运动控制语句

以下运动控制语句的命令没有定义的结论：

- MC_MoveVelocity
- MC_MoveJog

列出的运动控制语句通过输出参数 InVelocity 指示第一次达到选择的运行状态。以下部分基于运动控制语句“MC_MoveVelocity”以两个实例阶段说明了命令状态的行为：



阶段 1

- 输入参数 Execute = TRUE 的上升沿时启动命令。
- 命令执行期间输出参数显示 Busy = TRUE.
- 达到目标速度后，InVelocity 输出参数立即设置为 TRUE。
- 该命令被另一个命令取消：
 - 输出参数 Busy 变为 FALSE。
 - 输出参数 InVelocity 变为 FALSE.
 - 输出参数 CommandAborted 返回值 TRUE，以指示该命令被取消。
- 如果 Execute = FALSE，输出参数 CommandAborted 变为 FALSE。

阶段 2

- 输入参数 Execute = TRUE 的上升沿时启动命令。
- 命令执行期间输出参数显示 Busy = TRUE.
- 达到目标速度后，InVelocity 输出参数立即设置为 TRUE。
- 该命令被另一个命令取消：
 - 输出参数 Busy 变为 FALSE。
 - 输出参数 InVelocity 变为 FALSE.
 - 输出参数 CommandAborted 在一个执行周期中返回值 TRUE，以指示该命令被取消。

7.7.1.5 断电和重新启动后运动控制命令的行为

每次断电 (断开电源 -> 接通电源) 和重新启动 (RUN-STOP -> STOP-RUN) 后都将初始化工艺对象。

必须使用运动控制语句“MC_Power”重新启用轴。

如果需要回原点，则必须使用运动控制语句“MC_Home”重新参考轴。

断电或重新启动前不会初始化激活的命令。

7.7.1.5 运动控制语句的错误显示

运动控制语句在输出参数 Error、ErrorID und 和 ErroInfo 中指示工艺对象的所有错误。

输出参数 Error, ErrorID 和 ErrorInfo 中的错误显示

输出参数 Error = TRUE 说明运动控制语句无法执行该命令。错误原因通过输出参数 ErrorID 的值指示。输出参数 ErrorInfo 的值返回更多错误原因信息。我们区分以下用于错误指示的错误类别：

- **运行故障时不停止轴**
运行故障是指运动控制语句执行期间发生的故障。运行故障时不停止轴仅在触发运动控制语句中指示。不进行错误确认的情况下纠正错误后，就可重新启动运动控制语句。
- **运行故障时停止轴**
运行故障是指运动控制语句执行期间发生的故障。运行故障时停止轴在触发运动控制语句和“MC_Power”语句中指示。以组态的急停减速度停止轴。只有通过“MC_Reset”语句确认错误后，工艺对象才能接受新的运动命令。
- **运动控制语句组态错误**
组态错误是指运动控制语句的输入参数中有导致错误的不正确条目。该错误仅在触发运动控制语句中指示。不进行错误确认的情况下纠正错误后，就可重新启动运动控制语句。
- **组态错误**
如果轴参数组态有错误，则发生组态错误。该错误在触发运动控制语句和“MC_Power”语句中指示。下列选项可用于故障排除：
 - 再次将工艺对象下载到控制器。
 - 在线更正组态错误。
通过“MC_Reset”语句确认错误。
通过“MC_Power”语句再次启用轴。
- **内部错误**
该错误在触发运动控制语句和“MC_Power”语句中指示。必要时，可通过重新启动控制器来复位错误。

参见

错误 ID 和错误信息列表 (页 591)

7.7.1.5 错误 ID 和错误信息列表

错误 ID/错误信息列表

下表列出了可以显示在运动控制语句中的错误 ID 和错误信息。

运行故障时停止轴

错误 ID	错误信息	描述	解决方法
8000		驱动器错误，“驱动器准备就绪”故障	

错误 ID	错误信息	描述	解决方法
	0001	-	提供驱动器信号
8001		下限软件限位开关已被触发	
	000E	轴已到达该限位开关	通过“MC_Reset”语句确认错误；设置运动命令使轴沿正方向移动，脱离软件限位开关的范围
	0010	轴已通过该限位开关	
8002		上限软件限位开关已被触发	
	000E	轴已到达该限位开关	通过“MC_Reset”语句确认错误；设置运动命令使轴沿负方向移动，脱离软件限位开关的范围
	0010	轴已通过该限位开关	
8003		下限硬件限位开关已被触发	
	000E	轴已到达该限位开关	通过“MC_Reset”语句确认错误；设置运动命令使轴沿正方向移动，脱离硬件限位开关的范围
8004		上限硬件限位开关已被触发	
	000E	轴已到达该限位开关	通过“MC_Reset”语句确认错误；设置运动命令使轴沿负方向移动，脱离硬件限位开关的范围
8005		PTO 和 HSC 已被其它轴占用	
	0001	-	修改 PTO 和 HSC 的组态并将其下载到控制器

运行故障时不停止轴

错误 ID	错误信息	描述	解决方法
8200		轴未启用	
	0001	-	启用轴；重新启动命令
8201		轴已被另一个“MC_Power”语句启用	
	0001	-	仅通过一个“MC_Power”语句启用轴
8202		超出并行激活运动控制命令数的最大值（所有运动控制工艺对象最多 256 条命令）	
	0001	-	减少并行激活命令数；重新启动命令
8203		轴当前在“手动控制”模式（控制面板）下运行	
	0001	-	退出“手动控制”；重新启动命令
8204		轴未回原点	
	0001	-	通过“MC_Home”语句使轴回原点；重新启动命令
8205		轴当前由用户程序控制（该错误仅显示在控制面板中）	
	0001	-	

块参数错误

错误 ID	错误信息	描述	解决方法
8400 运动控制语句的“位置”(Position) 参数值不正确			
	0005	该值超出值范围 (大于 $1e^{12}$)	更正“位置”值；重新启动命令
	0006	该值超出值范围 (小于 $-1e^{12}$)	更正“位置”值；重新启动命令
8401 运动控制语句的“距离”(Distance) 参数值不正确			
	0005	该值超出值范围 (大于 $1e^{12}$)	更正“距离”值；重新启动命令
	0006	该值超出值范围 (小于 $-1e^{12}$)	更正“距离”值；重新启动命令
8402 运动控制语句的“速度”(Velocity) 参数值不正确			
	0008	该速度小于启动/停止速度	更正“速度”值；重新启动命令
	0009	超出最大速度	更正“速度”值；重新启动命令
8403 运动控制语句的“方向”(Direction) 参数值不正确			
	000F	选取值无效	更正选取值；重新启动命令
8404 运动控制语句“ MC_Home ”的“ Mode ”参数值不正确			
	000F	选取值无效	更正选取值；重新启动命令
8405 运动控制语句“ MC_Power ”的“ StopMode ”参数值不正确			
	000F	选取值无效	更正选取值；重新启动命令

组态错误

错误 ID	错误信息	描述	解决方法
8600 脉冲发生器 (PTO) 的组态无效			
	000D	地址无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
8601 高速计数器 (HSC, High-Speed Counter) 的组态无效			
	000D	地址无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
8602 “驱动器启用”(Drive enable) 输出的组态无效			
	000D	地址无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
8603 “驱动器准备就绪”(Drive ready) 输入的组态无效			
	000D	地址无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
8604 “电机每转的脉冲数”(Pulses per motor revolution) 值无效			

错误 ID	错误信息	描述	解决方法
	000A	该值小于或等于 0	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
8605	“电机每转的距离”(Distance per motor revolution) 值无效		
	0002	值使用了无效的数字格式	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	000A	该值小于或等于 0	
8606	“启动/停止速度”(Start / stop velocity) 值无效		
	0002	值的数字格式无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0003	值超出硬件限值	
	0004	值小于硬件限值	
8607	“最大速度”(maximum velocity) 值无效		
	0002	值的数字格式无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0003	值超出硬件限值	
	0004	值小于硬件限值	
8608	“加速度”(Acceleration) 值无效		
	0002	值的数字格式无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0003	值超出硬件限值	
	0004	值小于硬件限值	
8609	“减速度”(Deceleration) 值无效		
	0002	值的数字格式无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0003	值超出硬件限值	
	0004	值小于硬件限值	
860A	“急停减速度”(Emergency stop deceleration) 值无效		
	0002	值的数字格式无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0003	值超出硬件限值	
	0004	值小于硬件限值	
860B	下限软件限位开关的位置值无效		
	0002	值使用了无效的数字格式	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0005	该值超出值范围 (大于 $1e^{12}$)	
	0006	该值超出值范围 (小于 $-1e^{12}$)	
	0007	下限软件限位开关的位置值大于上限软件限位开关的位置值	
860C	上限软件限位开关的位置值无效		

错误 ID	错误信息	描述	解决方法
	0002	值的数字格式无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0005	该值超出值范围 (大于 $1e^{12}$)	
	0006	该值超出值范围 (小于 $-1e^{12}$)	
860D		下限硬件限位开关的地址无效	
	000B	下降沿的地址无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	000C	上升沿的地址无效	
860E		上限硬件限位开关的地址无效	
	000B	下降沿的地址无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	000C	上升沿的地址无效	
860F		“参考点位移”(reference point shift) 值无效	
	0002	值的数字格式无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0005	该值超出值范围 (大于 $1e^{12}$)	
	0006	该值超出值范围 (小于 $-1e^{12}$)	
8610		“启动速度”(startup velocity) 值无效	
	0002	值的数字格式无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0008	该速度小于启动/停止速度	
	0009	超出最大速度	
8611		“逼近速度”(approach velocity) 值无效	
	0002	值的数字格式无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	0008	该速度小于启动/停止速度	
	0009	超出最大速度	
8612		参考点开关的地址无效	
	000B	下降沿的地址无效	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴
	000C	上升沿的地址无效	
8613		虽然已禁用硬件限位开关，但还是激活了在该硬件限位开关处轴反向	
	0001	-	修改该组态并将其下载到控制器；通过调用“MC_Power”语句重新启用轴

内部错误

错误 ID	错误信息	描述	解决方法
8FFF		内部错误	

错误 ID	错误信息	描述	解决方法
	F001	-	通过断电和上电的方式重新启动 CPU
	F002	-	
	F003	-	
	F004	-	
	F005	-	
	F006	-	
	F007	-	
	F008	-	

参见

运动控制语句的错误显示 (页 591)

7.7.1.6 下载到 CPU

新的或修改的工艺对象组态必须下载到在线模式的 CPU。以下菜单和快捷菜单命令可用于下载操作：

- **菜单命令在线 > 下载到设备 (Online > Download to device)**
将工艺对象的组态、编译后的硬件数据和其余的软件项目数据下载到设备。
- **菜单命令在线 > 到设备的高级下载 (Online > Advanced download to device)**
与所选设备建立在线连接并将编译后的硬件和软件项目数据 (包括工艺对象的组态) 下载到设备。

已在项目浏览器中选择 CPU 对象。

- **下载到设备 > 全部 (Download to Device > All) 快捷菜单命令**
将工艺对象的组态、编译后的硬件数据和其余的软件项目数据下载到设备。
- **下载到设备 > 软件 (Download to Device > Software) 快捷菜单命令**
将修改后的工艺对象组态和修改后的块下载到设备。仅将修改的对象传送到设备。
- **下载到设备 > 软件 (所有块) (Download to Device > Software (all blocks)) 快捷菜单命令**
将所有块和工艺对象 (包括未更改的对象) 下载到设备。

已在项目浏览器中选择“程序块”(Program blocks) 对象。

- **下载到设备 > 软件 (Download to Device > Software) 快捷菜单命令**
将修改后的工艺对象组态和修改后的块下载到设备。仅将修改的对象传送到设备。
- **下载到设备 > 软件 (所有块) (Download to Device > Software (all blocks)) 快捷菜单命令**
将所有块和工艺对象 (包括未更改的对象) 下载到设备。

7.7.1.7 调试

控制面板用于测试轴和驱动器功能或在手动模式下移动轴。只有与 CPU 建立在线连接后，才能使用控制面板。

“手动控制”(Manual control) 按钮

单击“手动控制”(Manual control) 可在手动控制模式下移动轴。通过在用户程序的运动控制语句“MC_Power”中清除轴启用来启动。在“手动控制”模式下，控制面板获得轴和驱动器功能的控制优先权。用户程序不能影响轴功能。



警告

只能为一根轴激活手动控制。可以在自动控制模式下移动第二根轴，但这可能导致出现危险情况。

在这种情况下，将第二根轴设置为不运行。

“自动模式”(Auto mode) 按钮

单击“自动模式”(Auto mode) 可结束“手动控制”模式。控制优先权再次传给控制器。

运动控制语句“MC_Power”输入参数“Enable”的 FALSE -> TRUE 沿时重新启用轴并通过运动控制语句“MC_Home”启动轴回原点。

“启用”(Enable) 按钮

单击“启用”(Enable) 可在手动控制模式下移动轴。

“锁定”(Lock) 按钮

单击“锁定”(Lock) 可将轴临时锁定在手动控制模式下。

“命令”(Command) 区域

仅在轴启用后才可以执行“命令”(Command) 区域中的操作。可以选择以下命令输入之一：

- **点动 (Jog)**
该命令相当于用户程序中的运动控制命令“MC_MoveJog”。
- **定位 (Positioning)**
该命令相当于用户程序中的运动控制命令“MC_MoveAbsolute”和“MC_MoveRelative”。必须使轴回原点以便进行绝对定位。
- **回原点 (Homing)**
该命令相当于用户程序中的运动控制命令“MC_Home”。

根据选择，画面显示不同的设定值输入域和不同的命令启动按钮。

“轴状态”(Axis status) 区域

如果“手动控制”模式激活，则“轴状态”(Axis status) 区域中显示轴和驱动器的当前状态。轴的当前轴位置和速度显示在“实际值”(Actual values) 中。

单击“确认”(Acknowledge) 可确认所有清除的错误。

上一错误 (Last error)

在“手动控制”模式下，“上一错误”(Last error) 域指示最新错误和相应的详细错误信息。消除错误原因后，单击“确认”(Acknowledge) 可清除错误条目。

7.7.1.8 诊断

7.7.1.8 状态和错误位

诊断功能“状态和错误位”(Status and error bits) 用于监视轴的重要状态和故障消息。诊断功能的显示可以在在线模式下以“手动控制”模式和“自动模式”获得。状态错误消息具有下列含义：

轴的状态

状态	描述
已启用	轴已启用且准备好接受运动控制命令。
已回原点	轴已回原点且能够执行定位命令，例如，“MC_MoveAbsolute”语句的定位命令。对于相对回原点，轴不必回原点。 通过“MC_Home”语句主动回原点期间，该状态为 FALSE。
错误	轴工艺对象出错。根据错误类型，停止轴或拒绝命令。有关错误的更多信息，可以在自动模式下在运动控制语句的 ErrorID 和 ErrorInfo 参数中获得。在手动模式下，控制面板中的“上一错误”(Last error) 域显示更多错误原因信息。
控制面板激活	在控制面板中启用了“手动控制”模式。控制面板获得轴工艺对象的控制优先权。用户程序不能影响轴工艺对象。

驱动器状态

状态	描述
驱动器准备就绪	驱动器准备就绪。
错误	驱动器在其准备就绪信号故障后报告了错误。

轴运动的状态

状态	描述
停止	轴处于停止状态。
加速度	轴在加速。
恒速	轴在恒速运转。
减速度	轴在减速 (速度下降)。

运动模式的状态

状态	描述
定位	轴正在执行定位命令 (例如, 通过“MC_MoveAbsolute”或“MC_MoveRelative”语句)。
以预置的速度运动	轴正在执行以下运动控制语句的命令 : <ul style="list-style-type: none"> • MC_MoveVelocity • MC_MoveJog • MC_Halt
回原点	轴正在执行回原点命令 (例如, 通过“MC_Home”语句)。

错误位

错误	描述
软件限位开关已到达负端位置	到达软件限位开关的负端位置 ; 停止轴。
软件限位开关已超出负端位置	超出软件限位开关的负端位置 ; 停止轴。
软件限位开关已到达正端位置	到达软件限位开关的正端位置 ; 停止轴。
软件限位开关已超出正端位置	超出软件限位开关的正端位置 ; 停止轴。
硬件限位开关处于负端位置	到达或超出硬件限位开关的负端位置 ; 以急停减速度停止轴。
硬件限位开关处于正端位置	到达或超出硬件限位开关的正端位置 ; 以急停减速度停止轴。
PTO 和 HSC 已在使用中	轴无法访问分配的脉冲发生器和高速计数器。 停止并锁定轴。
组态错误	因组态错误, 工艺对象无法执行命令。 停止并锁定轴
内部错误	发生内部错误。 停止并锁定轴。

7.7.1.8 运动状态

诊断功能“运动状态”(Motion status) 用于监视轴的运动命令。 诊断功能的显示可以在在线模式下以“手动控制”模式和“自动模式”获得。 显示的域有下列含义 :

运动状态

状态	描述
目标位置	“目标位置”(Target position) 指示定位命令 (例如, 通过“MC_MoveAbsolute”或“MC_MoveRelative”语句) 的当前目标位置。只有轴已回原点且定位命令激活时, 才显示目标位置。
当前位置	“当前位置”(Current position) 域指示当前轴位置。该域仅指示回原点的轴的当前位置。
当前速度	“当前速度”(Current velocity) 域指示实际轴速度。

动态限制

动态限制	描述
速度	“速度”(Velocity) 域指示组态的最大轴速度。
加速度	“加速度”(Acceleration) 域指示组态的最大轴加速度。
减速度	“减速度”(Deceleration) 域指示组态的最大轴减速度。

7.7.2 使用 PID Compact

7.7.2.1 “PID Compact”使用指南

“PID_Compact”工艺对象用于实现自动和手动模式下都可自我优化调节的 PID 控制器。

在控制回路中, PID 控制器连续采集受控变量的实际测量值, 并将其与期望设定值进行比较。PID 控制器基于所生成的系统偏差计算控制器输出, 尽可能快速稳定地将受控变量调整到设定值。在 PID 控制器中, 控制器输出值通过以下三个分量进行计算:

- **比例分量**
比例分量计算的控制器输出值与系统偏差成比例。
- **积分分量**
积分分量计算的控制器输出值随着控制器输出的持续时间而增加, 最终补偿控制器输出。
- **微分分量**
PID 控制器的微分分量随着系统偏差变化率的增加而增加。受控变量将尽快调整到设定值。系统偏差的变化率减小时, 微分分量也会随之减小。

工艺对象在“初始启动时自调节”期间自行计算 PID 控制器的比例、积分和微分分量。可通过“运行中自调节”对这些参数进行进一步优化。

要求

要使用“PID Compact”工艺对象，必须创建具有 CPU S7-1200 的项目。

操作步骤

要使用“PID Compact”工艺对象，请按以下步骤操作：

1. [添加“PID Compact”工艺对象 \(页 601\)](#)
2. [在用户程序中调用“PID Compact”指令 \(页 607\)](#)
3. [组态工艺对象 \(页 602\)](#)
4. [下载到 CPU \(页 608\)](#)
5. [调试 \(页 610\)](#)
6. [保存优化的 PID 参数 \(页 616\)](#)

7.7.2.2 添加“PID Compact”工艺对象

可通过以下方法在项目中创建“PID Compact”工艺对象：

在项目浏览器中添加“PID Compact”工艺对象

添加工艺对象时，会为“PID_Compact”指令创建一个背景数据块。工艺对象的组态存储在该背景数据块中。要在项目浏览器中添加“PID Compact”工艺对象，请按以下步骤操作：

要求

已创建具有 CPU S7-1200 的项目。

操作步骤

1. 在项目树中打开 CPU 文件夹。
2. 打开“工艺对象”(Technology objects) 文件夹。
3. 双击“添加新对象”(Add new object) 对象。
将打开“添加新对象”(Add new object) 对话框。
4. 在“名称”(Name) 输入域中输入该工艺对象的专用名称。
5. 单击“PID 控制器”(PID controller) 按钮。
6. 如果要更改背景数据块的推荐数据块编号，请选择“手动”(Manual) 选项。
7. 如果想要为该工艺对象添加用户信息，请单击“更多信息”(Further information)。
8. 如果要添加该工艺对象，则单击“确定”(OK)。如果要放弃输入内容，则单击“取消”(Cancel)。

结果

新工艺对象已创建，并存储在项目树的“工艺对象”(Technology objects) 文件夹中。

注意

可以选中该对话框底部的“添加并打开新块”(Add and open new block) 复选框。这将在添加操作完成后打开工艺对象的组态。

在程序编辑器中通过插入“PID_Compact”指令添加工艺对象。

在程序编辑器中添加“PID_Compact”指令时，将创建背景数据块。该背景数据块与“PID Compact”工艺对象相对应。工艺对象的组态存储在该背景数据块中。要在程序编辑器中添加“PID Compact”工艺对象，请按以下步骤操作：

要求

已创建具有 CPU S7-1200 的项目。

操作步骤

1. 打开要在其中插入“PID_Compact”指令的循环中断 OB。
要确保控制器的扫描时间恒定，必须在循环中断 OB 中调用“PID_Compact”指令。
2. 在“指令”(Instructions) 窗口中，打开“扩展指令”(Extended instructions) 组和“PID”文件夹。
3. 选择“PID_Compact”指令，并将其拖动到块中。
“调用选项”(Call options) 对话框随之打开。
4. 在“名称”(Name) 输入域中输入该工艺对象的专用名称。
5. 如果要更改背景数据块的推荐数据块编号，请选择“手动”(Manual)。
6. 如果要添加背景数据块或工艺对象，则单击“确定”(OK)。

结果

插入了“PID_Compact”指令，并创建了新工艺对象。该对象存储在项目浏览器的“工艺对象”(Technology objects) 文件夹中。

参见

编程 (页 607)

7.7.2.3 组态

7.7.2.3 组态工艺对象

可在组态窗口和巡视窗口中组态工艺对象“PID Compact”的属性。

- 在组态窗口中：适用于所有组态属性
- 在巡视窗口中：适用于运行期间所需的组态属性

在组态窗口中组态

要打开工艺对象的组态窗口，请按以下步骤操作：

1. 在项目树中打开所需工艺对象组。
2. 双击“组态”(Configuration) 对象。




在巡视窗口中组态

要打开工艺对象“PID Compact”的巡视窗口，请按以下步骤操作：

1. 在项目树中打开“程序块”(Program blocks) 文件夹。
2. 双击要调用“PID Compact”指令的块 (循环中断 OB)。
该块将在工作区中打开。
3. 单击“PID_Compact”指令。
4. 在巡视窗口中，依次选择“属性”(Properties) 和“组态”(Configuration) 选项卡。

组态窗口的图标

组态区域导航以及巡视窗口中的图标显示有关组态完成情况的详细信息：

	组态包含默认值且已完成。 组态仅包含默认值。通过这些默认值即可使用工艺对象，而无需进一步更改。
	组态包含用户定义的值且已完成。 组态的所有输入域中均包含有效值，而且至少更改了一个默认值。
	组态不完整或有缺陷。 至少一个输入域或下拉列表框不包含任何值或者包含的值无效。相应域或下拉列表框的背景为红色。单击这些域或下拉列表框时，弹出的错误消息便会指出错误原因。

参见

基本参数 (页 603)
实际值标定 (页 604)
输入监视 (页 605)
PWM 限制 (页 605)
受控值限制 (页 606)
PID 参数 (页 606)

7.7.2.3 基本参数

在巡视窗口或组态窗口的“默认设置”(Default settings) 中，组态工艺对象“PID Compact”的基本属性。

控制系统类型

预先选择“控制系统类型”(Type of control system)，然后选择所需的控制系统类型单元。

如果受控值的增加会引起实际值的减小 (例如, 由于阀位开度增加而使水位下降, 或者由于冷却性能增加而使温度降低), 请选中“控制方向反向”(Inversion of the control direction) 复选框。

输入/输出分配

在本部分, 将有效值分配给适用于“PID_Compact”指令的设定值、实际值和受控变量的输入和输出参数。只能在巡视窗口中组态这些输入和输出参数。

- **设定值**

在下拉列表框中, 选择是使用功能块的值还是背景数据块的值。

- **实际值**

在下拉列表框中, 选择是使用输入参数“Input”还是“Input_PER”。

- 如果要使用用户程序中的实际值, 请使用“Input”。此外, 在下拉列表框中, 选择是使用功能块的值还是背景数据块的值。

- 如果要使用模拟量输入的实际值, 请使用“Input_PER”。

在以下域中, 指定适用于实际值的参数。

- **受控变量**

在下拉列表框中, 为“PID_Compact”指令选择要组态的受控变量。有以下可选方式:

- **输出**

使用用户程序的变量作为受控变量输出。

- **Output_PER**

使用模拟量输出作为受控变量输出。

- **Output_PWM**

使用数字量开关输出, 并通过脉宽调制对其进行控制。受控变量通过变量开关时间形成。

在以下域中, 指定适用于受控变量输出的参数。

参见

组态工艺对象 (页 602)

实际值标定 (页 604)

输入监视 (页 605)

PWM 限制 (页 605)

受控值限制 (页 606)

PID 参数 (页 606)

7.7.2.3 实际值标定

在“实际值标定”(Actual value scaling) 组态窗口中, 组态实际值的标定, 并指定实际值绝对限值。

标定

通过上下限值对指定实际值的标定。

每一对值均包含模拟量输入的模拟值以及相应标定点的物理值。根据默认设置的组态，还可使用用户程序的过程值来代替模拟量输入的模拟值。

上限和下限

指定实际值的绝对上限和下限。只要在运行期间超过上限或低于下限，调节控制过程就会停止，同时受控变量设置为 0%。

“默认设置”按钮

如果要为标定值和上下限值装载默认设置值，以及要替换现有值，请单击“默认设置”(Default setting) 按钮。

参见

组态工艺对象 (页 602)

基本参数 (页 603)

输入监视 (页 605)

PWM 限制 (页 605)

受控值限制 (页 606)

PID 参数 (页 606)

7.7.2.3 高级参数

7.7.2.3 输入监视

在“输入监视”(Input monitoring) 组态窗口中，组态实际值的警告上限和下限。如果在运行期间超出或低于其中一个警告限值，则将在“PID_Compact”指令的以下参数中显示一条警告：

- 如果超出警告上限，则警告出现在输出参数“InputWarning_H”中
- 如果低于警告下限，则警告出现在输出参数“InputWarning_L”中

参见

组态工艺对象 (页 602)

基本参数 (页 603)

实际值标定 (页 604)

PWM 限制 (页 605)

受控值限制 (页 606)

PID 参数 (页 606)

7.7.2.3 PWM 限制

在“PWM 限制”(PWM limitations) 组态窗口中，组态脉宽调制允许的最小 ON 和 OFF 开关时间。如果要减少开关周期数，请延长最小 ON 和 OFF 时间。

PWM 限制的组态会影响受控变量的输出“Output_PWM”。受控变量的开关输出“Output_PWM”通过脉宽调制的开关操作来实现。脉宽调制的 ON 和 OFF 开关时间是“PID_Compact”指令调用周期的倍数。

注意

脉宽调制的开关输出由“PID_Compact”指令控制。不使用 CPU 中集成的脉冲发生器。

参见

组态工艺对象 (页 602)

基本参数 (页 603)

实际值标定 (页 604)

输入监视 (页 605)

受控值限制 (页 606)

PID 参数 (页 606)

7.7.2.3 受控值限制

在“受控限值”(Manipulated value limits) 组态窗口中，组态受控变量的绝对限值。在手动模式或自动模式下，都不要超过或低于受控变量的绝对限值。如果在手动模式下指定了一个超出限值范围的受控变量，CPU 会将有效值限制为组态的限值。

注意

如果使用受控变量输出“Output_PWM”，则只能控制正受控变量。

参见

组态工艺对象 (页 602)

基本参数 (页 603)

实际值标定 (页 604)

输入监视 (页 605)

PWM 限制 (页 605)

PID 参数 (页 606)

7.7.2.3 PID 参数

在“PID 参数”(PID parameters) 组态窗口中，组态 PID 控制器的参数。如果要手动指定工艺对象的 PID 参数，请使用该组态窗口。为此，请选中“使用手动 PID 参数”(Use manual PID parameters) 复选框，并输入所需的 PID 参数。

视初始情况而定，调用时将显示以下值：

- 添加了工艺对象后
添加了工艺对象后，将显示工艺对象的默认值。
- 单击了“将 PID 参数装载到项目中”(Load PID parameters into project) 图标后
在调试窗口中单击了该图标后，当前正在 CPU 中使用的 PID 参数将装载到项目并显示在组态中。

参见

组态工艺对象 (页 602)
基本参数 (页 603)
实际值标定 (页 604)
输入监视 (页 605)
PWM 限制 (页 605)
受控值限制 (页 606)

7.7.2.4 编程

7.7.2.4 在用户程序中调用“PID_Compact”指令

要求

“PID Compact”控制器的采样时间由调用间隔确定。为了严格遵守采样时间，必须在循环中断组织块中调用“PID_Compact”指令。在项目中添加所需的循环中断 OB。将所需采样时间值组态为循环中断 OB 的“时帧”。

操作步骤

要在用户程序中调用“PID_Compact”指令，请按以下步骤操作：

1. 在项目树中打开“程序块”(Program blocks) 文件夹。
2. 双击所需的定时中断 OB。
该块将在工作区中打开。
3. 在“指令”(Instructions) 窗口中，打开“扩展指令”(Extended instructions) 组和“PID”文件夹。
4. 选择“PID_Compact”指令，并将其拖动到循环中断 OB 中。
“调用选项”(Call options) 对话框随之打开。
5. 根据初始情况，执行以下操作：
 - 已添加了工艺对象“PID Compact”：
单击“取消”(Cancel) 关闭“调用选项”(Call options) 对话框。
在工作区中，双击背景数据块的输入域。
从列表中选择该工艺对象。
 - 尚未添加工艺对象“PID Compact”：
“调用选项”(Call options) 对话框打开。
在“名称”(Name) 输入域中，键入工艺对象的专用名称。
如果要更改背景数据块的建议数据块编号 (在工艺对象“PID Compact”中，工艺对象和背景数据块相同)，请选择“手动”(manual)。
要添加背景数据块或工艺对象，请单击“确定”(OK) 确认输入内容。

结果

插入了“PID_Compact”指令，并创建了新工艺对象。在项目树的“工艺对象”(Technology objects)文件夹中创建了该对象。

参见

PID_Compact : 具有自调节功能的 PID 控制器 (页 926)

下载到 CPU (页 608)

对工艺对象的变量进行读/写访问 (页 608)

7.7.2.4 下载到 CPU

必须将新的或修改的工艺对象组态下载到在线模式的 CPU。以下菜单和快捷菜单命令可用于下载操作：

- **菜单命令在线 > 下载到设备 (Online > Download to device)**
将工艺对象组态、编译后的硬件数据和其余的软件项目数据下载到设备。
- **菜单命令在线 > 到设备的高级下载 (Online > Advanced download to device)**
与所选设备建立在线连接并将编译后的硬件和软件项目数据（包括工艺对象的组态）下载到设备。

已在项目浏览器中选择 CPU 对象

- **下载到设备 > 全部 (Download to Device > All) 快捷菜单命令**
将工艺对象组态、编译后的硬件数据和其余的软件项目数据下载到设备。
- **下载到设备 > 软件 (Download to Device > Software) 快捷菜单命令**
将修改后的工艺对象组态和修改后的块下载到设备。仅将修改的对象传送到设备
- **下载到设备 > 软件 (所有块) (Download to Device > Software (all blocks)) 快捷菜单命令**
将所有块和工艺对象（包括未更改的对象）下载到设备

已在项目浏览器中选择“程序块”(Program blocks) 对象

- **下载到设备 > 软件 (Download to Device > Software) 快捷菜单命令**
将修改后的工艺对象组态和修改后的块下载到设备。仅将修改的对象传送到设备
- **下载到设备 > 软件 (所有块) (Download to Device > Software (all blocks)) 快捷菜单命令**
将所有块和工艺对象（包括未更改的对象）下载到设备

参见

在用户程序中调用“PID_Compact”指令 (页 607)

7.7.2.4 对工艺对象的变量进行读/写访问

使用背景数据块（对应于工艺对象“PID Compact”）的变量，可以访问基本组态数据、PID 控制器工作模式，以及工艺对象的其它状态显示。

下表列出了可用的变量，并对其作用和含义进行了说明：

变量	数据类型	说明	
sRet.i_Mode	INT	工作模式的更改由沿触发。	
		0	选择“未激活”模式（停止控制器）
		1	选择“初始启动时自调节”模式
		2	选择“运行中自调节”模式
		3	选择“自动”模式
		4	选择“手动”模式
sPid_Cmpt.r_Sp_Hlm	REAL	设定值绝对上限（默认设置为组态的实际值绝对上限）	
sPid_Cmpt.r_Sp_Llm	REAL	设定值绝对下限（默认设置为组态的实际值绝对下限）	
sPid_Cmpt.r_Pv_Norm_IN_1	REAL	模拟量输入“Input_PER”的模拟值下限	
sPid_Cmpt.r_Pv_Norm_IN_2	REAL	模拟量输入“Input_PER”的模拟值上限	
sPid_Cmpt.r_Pv_Norm_OUT_1	REAL	标定点下限的物理值	
sPid_Cmpt.r_Pv_Norm_OUT_2	REAL	标定点上限的物理值	
sPid_Cmpt.r_Lmn_Hlm	REAL	输出参数“Output”的受控变量上限	
sPid_Cmpt.r_Lmn_Llm	REAL	输出参数“Output”的受控变量下限	
sPid_Cmpt.r_Lmn_Pwm_PPTm	REAL	脉宽调制的最小 ON 时间（秒）	
sPid_Cmpt.r_Lmn_Pwm_PBTm	REAL	脉宽调制的最小 OFF 时间（秒）	
sPid_Cmpt.b_InvCtrl	BOOL	FALSE	不反转控制方向
		TRUE	反转控制方向
sPid_Cmpt.b_Input_PER_On	BOOL	FALSE	输入参数“Input_PER”未激活
		TRUE	输入参数“Input_PER”已激活
sPid_Cmpt.b_LoadBackup	BOOL	激活备份参数集。如果优化失败，可通过置位该位重新激活先前的 PID 参数。	
sPid_Calc.d_CycCountEnd	DINT	“PID_Compact”指令在完成当前自调节前所执行的估计最大处理周期数。	
sPid_Calc.d_CycCounter	DINT	“PID_Compact”指令自启动自调节以来所执行的当前处理周期数。	

注意

在“未激活”模式下编辑本表列出的变量，以防 PID 控制器出现故障。通过将变量“si_Mode”的值设置为“0”，可强制“未激活”模式。

7.7.2.5 调试

7.7.2.5 调试

调试窗口有助于您调试 PID 控制器。可以在趋势视图中监视设定值、实际值以及受控变量的值随时间轴的变化。调试窗口支持以下功能：

- 使用“初始启动时自调节”(Self tuning during initial start) 优化控制器
- 使用“运行中自调节”(Self tuning in the operating point) 优化控制器
如果要精确地调整控制器优化，请使用“运行中自调节”(Self tuning in the operating point)。
- 在趋势视图中监视当前闭环控制
- 通过指定手动受控变量对过程进行测试

只有与 CPU 建立了在线连接后，才能使用调试窗口。

基本处理操作

- 在“更新周期”(Updating cycle) 下拉列表框中，选择所需的更新周期。
调试窗口中显示的所有值都将以所选的更新周期进行更新。
- 如果要使用调试功能，请单击“开启测量功能”(Measuring on) 图标。
随后便开始记录值。设定值、实际值以及受控变量的当前值将记录在趋势视图中。可以对调试窗口进行操作。
- 如果要终止调试功能，请单击“关闭测量功能”(Measuring off) 图标。
可以继续对趋势视图中记录的值进行分析。

调试窗口关闭后，趋势视图中的记录操作将终止，所记录的值将被删除。

参见

“初始启动时自调节”模式 (页 610)

“运行中自调节”模式 (页 611)

使用趋势视图 (页 612)

“手动”模式 (页 615)

保存优化的 PID 参数 (页 616)

7.7.2.5 “初始启动时自调节”模式

下面说明如何在工艺对象“PID Compact”的调试窗口中执行“初始启动时自调节”。

要求

- “PID_Compact”指令在循环中断 OB 中调用。
- 与 CPU 建立了在线连接，并且 CPU 处于“RUN”模式。
- 已通过“开启测量功能”(Measuring on) 图标启用了调试窗口的功能。
- 未选中“手动受控变量”(Manual manipulated variable) 复选框。

- 设定值和实际值均在组态的限值范围内 (请参见“输入监视”组态)。
- 设定值和实际值的差值大于组态的“输入监视”限值的 50%。

操作步骤

要执行“初始启动时自调节”，请按以下步骤操作：

1. 在调试窗口的“优化”(Optimization) 部分选择“初始启动时自调节”(Self tuning during initial start) 选项。
2. 单击“启动自调节”(Start self tuning) 图标。

自调节功能启动。“状态”(Status) 域显示当前步骤以及可能出现的所有错误。进度条指示当前步骤的进度。

注意

如果进度条到达了 100%，并且需要停止自调节功能执行，请单击“停止控制器”(Stop controller) 图标。检查工艺对象的组态，必要时，重新启动自调节功能。

结果

如果执行了自调节且没有错误，则 PID 参数得到了优化。PID 控制器切换到自动模式，并使用优化的参数。在电源接通以及重启 CPU 期间，优化的 PID 参数保持不变。

参见

调试 (页 610)
“运行中自调节”模式 (页 611)
使用趋势视图 (页 612)
“手动”模式 (页 615)
保存优化的 PID 参数 (页 616)

7.7.2.5 “运行中自调节”模式

下面说明如何在工艺对象“PID Compact”的调试窗口中执行“运行中自调节”。

要求

- “PID_Compact”指令在循环中断 OB 中调用。
- 与 CPU 建立了在线连接，并且 CPU 处于“RUN”模式。
- 已通过“开启测量功能”(Measuring on) 图标启用了调试窗口的功能。
- 未选中“手动受控变量”(Manual manipulated variable) 复选框。
- 设定值和实际值均在组态的限值范围内 (请参见“输入监视”组态)。
- 设定值和实际值的差值小于“输入监视”限值的 50%。

如果该差值大于 50%，则执行“初始启动时自调节”。之后将自动执行“运行中自调节”。

操作步骤

要执行“运行中自调节”，请按以下步骤操作：

1. 在调试窗口的“优化”(Optimization) 部分选择“运行中自调节”(Self tuning in the operating point) 选项。
2. 单击“启动自调节”(Start self tuning) 图标。

自调节功能启动。“状态”(Status) 域显示当前步骤以及可能出现的所有错误。进度条指示当前步骤的进度。

注意

如果进度条到达了 100%，并且需要停止自调节功能执行，请单击“停止控制器”(Stop controller) 图标。检查工艺对象的组态，必要时请重新启动自调节功能。

结果

如果执行了自调节且没有错误，则 PID 参数得到了优化。PID 控制器切换到自动模式，并使用优化的参数。在电源接通以及重启 CPU 期间，优化的 PID 参数保持不变。

参见

调试 (页 610)

“初始启动时自调节”模式 (页 610)

使用趋势视图 (页 612)

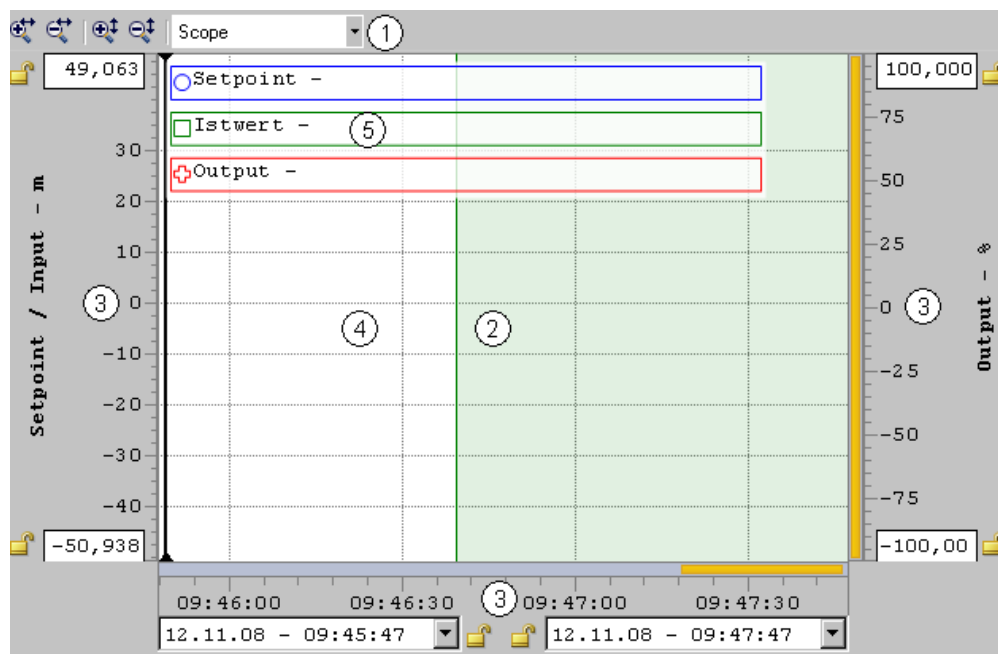
“手动”模式 (页 615)

保存优化的 PID 参数 (页 616)

7.7.2.5 使用趋势视图

趋势视图用于以图形方式描述设定值、实际值以及受控变量。趋势视图的值以所选的更新周期进行更新。单击“开启测量功能”(Measurement on) 开始记录趋势值，单击“关闭测量功能”(Measurement off) 停止记录趋势值。

趋势视图的元素



- ① 选择显示模式
- ② 趋势视图
- ③ 对轴进行移动和标定的区域
- ④ 标尺
- ⑤ 标尺上趋势值的图例

选择显示模式

可选择以下显示模式来显示趋势记录：

- 条状图（连续显示）
新趋势值在趋势视图右侧输入。之前的趋势值向左侧滚动。时间轴无法移动。
- 示波器图（跳动式区域显示）
新趋势值在趋势视图中自左向右输入。如果到达趋势视图的右侧边缘，监视区域将向右移动一个视图宽度。时间轴可以在监视区域的限值范围内移动。
- 扫描图（滚动显示）
新趋势值依照滚动显示的方式在趋势视图中显示。趋势值从左向右输入。上一次滚动显示的趋势值在写入位置处被覆盖。时间轴无法移动。
- 静态图（静态区域显示）
中断趋势值写入。在后台执行新趋势值的记录。在先前的整个记录时间段内，时间轴可以进行移动。

可见区与记录区的比例显示在比例画面中：



比例画面的宽度表示记录的完整持续时间。画面的黄色部分表示在趋势视图中可见的记录部分。画面的灰色部分表示在趋势视图中不可见的记录部分。











在“静态”显示模式下，通过用鼠标拖动比例画面的可见范围，可将其沿轴移动。

趋势视图

在趋势视图中显示设定值、实际值和受控变量的相应趋势图。这些趋势图除颜色不同外，还分别用符号（称为图例）进行标识。

移动和标定轴

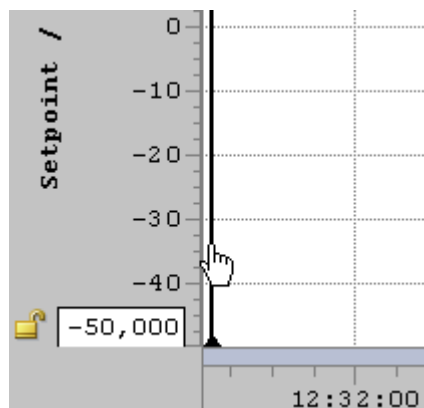
设定值、实际值与受控变量的轴以及时间轴可以分别进行移动和标定。鼠标左键和鼠标右键分配了不同的功能。可以使用以下图标和鼠标操作：

	向上或向下移动设定值/实际值轴或者受控变量轴。 如果锁定了上标定值或下标定值，则无法移动该轴。
	向右或向左移动时间轴。 如果锁定了左或右标定值，则无法移动该轴。
	对设定值轴、实际值轴或受控变量轴进行拉伸和压缩。 <ul style="list-style-type: none"> 如果没有锁定任何标定值，则对称拉伸或压缩轴的标定。 拉伸或压缩轴时，锁定的标定值保持不变。
	拉伸和压缩时间轴 <ul style="list-style-type: none"> 如果没有限定任何标定值，则对称拉伸或压缩时间轴的标定。 拉伸或压缩轴时，锁定的标定值保持不变。
	对设定值/实际值轴或者受控变量轴进行拉伸和压缩。 下标定值不会由于拉伸或压缩而改变。
	对设定值/实际值轴或者受控变量轴进行拉伸和压缩。 上标定值在拉伸或压缩期间不变。
	拉伸和压缩时间轴。 右标定值不会由于拉伸或压缩而改变。
	拉伸和压缩时间轴。 左标定值不会由于拉伸或压缩而改变。
 50,000	输入一个标定值。
 50,000	使用挂锁图标可以锁定当前标定值。只能锁定相应轴上的一个值。
	双击趋势视图可对其中的设定值、实际值以及受控变量的标定和位置进行优化。
	双击轴区可恢复轴的默认位置和标定。

使用标尺

使用一个或多个标尺分析趋势轮廓的离散值。

标尺停驻位置在趋势图区域的左边。另一个停驻位置在趋势图区域的上边。但是，无法显示这些标尺的值。



将鼠标移动到趋势图区域的左边，注意鼠标指针的变化情况。将垂直标尺拖动到要分析的测量趋势图上。

趋势值在标尺上以左对齐的方式输出。标尺位置对应的时间显示在标尺下方。

活动标尺的趋势值在图例中显示。如果将多个标尺拖动到趋势图区域，则最后一个标尺为激活状态。活动标尺以相应的彩色符号



指示。

单击未激活的标尺便可将其重新激活。

按住 ALT 键的同时单击不再需要的标尺，便可将其删除。

参见

调试 (页 610)

“初始启动时自调节”模式 (页 610)

“运行中自调节”模式 (页 611)

“手动”模式 (页 615)

保存优化的 PID 参数 (页 616)

7.7.2.5 “手动”模式

下面说明如何在工艺对象“PID Compact”的调试窗口中使用“手动”工作模式。

要求

- “PID_Compact”指令在循环中断 OB 中调用。
- 与 CPU 建立了在线连接，并且 CPU 处于“RUN”模式。
- 已通过“开启测量功能”(Measuring on) 图标启用了调试窗口的功能。

操作步骤

如果希望通过指定手动受控变量来测试过程，请使用调试窗口中的“手动模式”。要指定手动受控变量，请按以下步骤操作：

1. 在“当前值”(Current values) 区域中，选中“手动受控变量”(Manual manipulated variable) 复选框。

控制器输出与受控变量输出之间的连接将断开。PID 控制器继续在自动模式下运行。

闭环控制的上一个当前受控变量保持激活，作为受控变量输出中的手动受控变量。

2. 在“输出”(Output) 域中以 % 值的形式输入所需的受控变量。
3. 单击控制图标



。

结果

受控变量被写入 CPU 并立即激活。

注意

由于 PID 控制器继续监视实际值，因而任何一次超出实际值限值范围的情况都会显示在“状态”(Status) 域中，同时将禁用 PID 控制器。

如果希望再次由 PID 控制器指定受控变量，则清除“手动受控变量”(Manual manipulated variable) 复选框。

参见

调试 (页 610)
“初始启动时自调节”模式 (页 610)
“运行中自调节”模式 (页 611)
使用趋势视图 (页 612)
保存优化的 PID 参数 (页 616)

7.7.2.5 保存优化的 PID 参数

PID 控制器在 CPU 中进行优化。如果希望在将项目数据重新装载到 CPU 时，再次使用已在 CPU 中优化过的 PID 参数，则将 PID 参数保存在项目中。请执行以下步骤：

要求

- 与 CPU 建立了在线连接，并且 CPU 处于“RUN”模式。
- 已通过“开启测量功能”(Measuring on) 按钮启用了调试窗口的功能。

操作步骤

1. 在调试窗口中，单击“将 PID 参数装载到项目中”(Load PID parameters into project) 图标。

结果

当前激活的 PID 参数存储在项目数据中。

参见

调试 (页 610)
 “初始启动时自调节”模式 (页 610)
 “运行中自调节”模式 (页 611)
 使用趋势视图 (页 612)
 “手动”模式 (页 615)

7.8 参考

7.8.1 指令

7.8.1.1 LAD

7.8.1.1 位逻辑

7.8.1.1 : 常开触点

符号

<操作数>



参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、L、D (页 293)	要查询其信号状态的操作数。

描述

常开触点的激活取决于相关操作数的信号状态。如果操作数的信号状态为“1”，常开触点将闭合。能流从左侧电源线通过该常开触点流到右侧电源线，并且该运算的输出的信号状态设置为“1”。

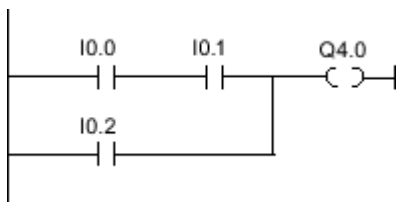
如果操作数的信号状态为“0”，则常开触点不会被激活。到右侧电源线的能流中断，并且该运算的输出的信号状态复位为“0”。

两个或多个常开触点串联时通过 AND 逻辑逐位相连。串联时，所有触点都闭合后才产生能流。
常开触点并联时通过 OR 逻辑相连。并联时，有一个触点闭合就会产生能流。

放置

“常开触点”运算可以放置在程序段中的任何位置。

实例



满足下列条件之一时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 I .2 的信号状态为“1”。

参见

- 将操作数插入 LAD 指令 (页 430)
- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 控制传送带实例 (页 559)
- 检测存储区域的填充量实例 (页 562)
- 控制室温实例 (页 565)

7.8.1.1 --|/|---: 常闭触点

符号

<操作数>



参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、L、D (页 293)	要查询其信号状态的操作数。

说明

常闭触点的激活取决于相关操作数的信号状态。操作数的信号状态为“1”时，触点“断开”并且到右侧电源线的能流中断。在这种情况下，运算的输出的信号状态为“0”。

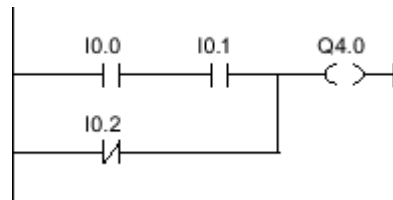
如果操作数的信号状态为“0”，常闭触点将“闭合”。能流通过该常闭触点流到右侧电源线，并且该运算的输出的信号状态设置为“1”。

两个或多个常闭触点串联时通过 AND 逻辑逐位相连。串联时，所有触点都闭合后才产生能流。
常闭触点并联时通过 OR 逻辑相连。并联时，有一个触点闭合就会产生能流。

放置

该运算可以放置在程序段中的任何位置。

实例



满足下列条件之一时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 I 0.2 的信号状态为“0”。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[检测传送带方向实例 \(页 560\)](#)

[检测存储区域的填充量实例 \(页 562\)](#)

[控制室温实例 \(页 565\)](#)

7.8.1.1 --|NOT|--：取反逻辑运算结果

符号

--| NOT |--

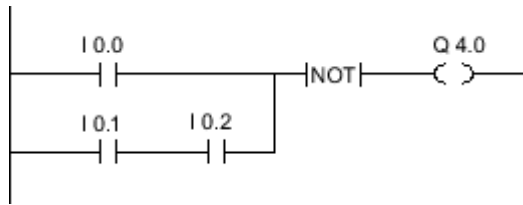
描述

可以使用“取反逻辑运算结果”运算取反逻辑运算结果 (RLO) 的信号状态。该运算输入的信号状态为“1”时，运算输出的信号状态将为“0”。该运算输入的信号状态为“0”时，运算输出的信号状态将为“1”。

放置

“取反逻辑运算结果”运算可以放置在程序段中的任何位置。

实例



满足下列条件之一时，将复位输出 Q 4.0：

- 输入 I.0 的信号状态为“1”。
- 输入 I.1 与 I.2 的信号状态为“1”。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 控制室温实例 (页 565)*
- 将操作数插入 LAD 指令 (页 430)*

7.8.1.1 ---

符号

<操作数>
--- () ---

参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、L、D (页 293)	RLO 为“1”时置位的操作数。

描述

可以使用“输出线圈”运算置位指定操作数的位。线圈输入端的逻辑运算结果 (RLO) 为“1”时，指定操作数的信号状态将被置位为“1”。线圈输入端的信号状态为“0”时，指定操作数的位被复位为“0”。

该运算不会影响 RLO。线圈输入端的 RLO 会立即发送到输出。

放置

“输出线圈”运算可以放置在程序段中的任何位置。通过使用分支，多个线圈可以相互嵌套放置。

实例

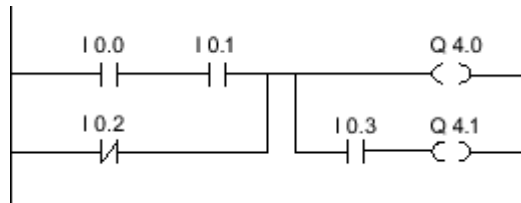


图 7-2

满足下列条件之一时，将置位输出 Q 4.0：

- 输入 I0.0 与 I0.1 的信号状态为“1”。
- 输入 I0.2 的信号状态为“0”。

满足下列条件之一时，将置位输出 Q 4.1：

- 输入 I0.0 与 I0.1 以及输入 I0.3 的信号状态为“1”。
- 输入 I0.2 的信号状态为“0”且输入 I0.3 的信号状态为“1”。

参见

插入 LAD 元素 (页 417)

将操作数插入 LAD 指令 (页 430)

检测存储区域的填充量实例 (页 562)

控制室温实例 (页 565)

更改 LAD 元素 (页 425)

7.8.1.1 --

符号

<操作数>

-- (/) --

参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、D、L (页 293)	RLO 为“0”时置位的操作数。

描述

“取反线圈”运算将逻辑运算结果 (RLO) 取反并将取反结果分配给指定的操作数。线圈输入端的 RLO 为“1”时，复位操作数。线圈输入端的 RLO 为“0”时，操作数的信号状态置位为“1”。

该运算不会影响 RLO。线圈输入端的 RLO 将立即发送到线圈输出端。

放置

“取反线圈”运算可以放置在程序段中的任何位置。

实例

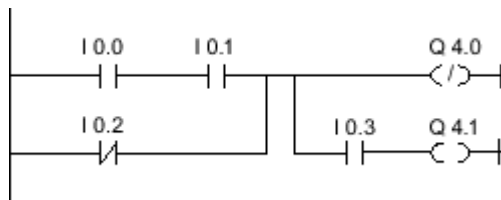


图 7-2

满足下列条件之一时，将复位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 I .2 的信号状态为“1”。

参见

插入 LAD 元素 (页 417)
 将操作数插入 LAD 指令 (页 430)
 更改 LAD 元素 (页 425)

7.8.1.1 ---

符号

<操作数>
 ---(R)---

参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、L、D (页 293)	RLO 为“1”时复位的操作数。

描述

可以使用“复位输出”运算将指定操作数的信号状态设置为“0”。

只有线圈输入端的逻辑运算结果 (RLO) 为“1”时，才会执行该运算。如果能流通过线圈 (RLO 为“1”)，那么指定的操作数将被置位为“0”。如果线圈输入端的逻辑运算结果为“0”(没有信号流过线圈)，则指定操作数的信号状态将保持不变。

该运算不会影响 RLO。线圈输入端的 RLO 将立即发送到线圈输出端。

放置

“复位输出”运算可以放置在程序段中的任何位置。

实例

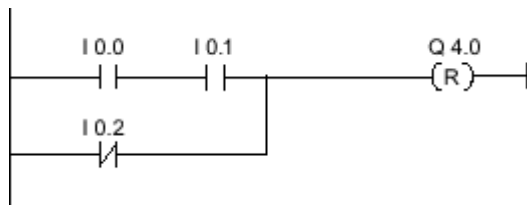


图 7-2

满足下列条件之一时，将复位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 为“1”
- 输入 I 0.2 的信号状态为“0”。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

控制传送带实例 (页 559)

检测传送带方向实例 (页 560)

7.8.1.1 ---

符号

<操作数>

---(S)---

参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、L、D (页 293)	RLO 为“1”时置位的操作数。

描述

可以使用“置位输出”运算将指定操作数的信号状态置位为“1”。

只有线圈输入端的逻辑运算结果为“1”时，才会执行该运算。如果能流通过线圈 (RLO 为“1”)，那么指定的操作数将被置位为“1”。如果线圈输入端的逻辑运算结果为“0” (没有信号流过线圈)，则指定操作数的信号状态将保持不变。

该运算不会影响 RLO。线圈输入端的 RLO 会立即发送到输出。

放置

“置位输出”运算可以放置在程序段中的任何位置。

实例

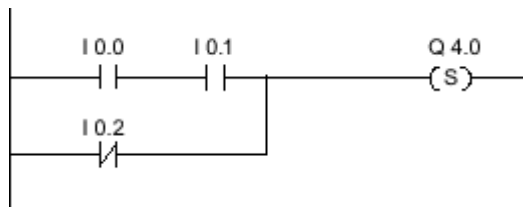


图 7-2

满足下列条件之一时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 I 0.2 的信号状态为“0”。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 将操作数插入 LAD 指令 (页 430)*
- 控制传送带实例 (页 559)*
- 检测传送带方向实例 (页 560)*

7.8.1.1 SET_BF：置位位域

符号

<Operand1>
 —(SET_BF)
 <Operand2>

参数	数据类型	存储区	描述
<操作数 1>	BOOL	I、Q、M、D、L (页 293)	指向要置位的第一个位的指针
<操作数 2>	UINT	常数	要置位的位数

描述

可以使用“置位位域”操作从某个指定地址开始的多个位的置位。要置位的位数决定 <操作数 2> 的值。要置位的第一个位的地址用 <操作数 1> 定义。如果 <操作数 2> 的值大于所选字节中的位数，则将置位下一个字节的位。在明确复位这些位（例如，通过另一个运算）之前，它们会保持置位。

只有使能输入的信号状态为“1”时，才执行该运算。如果输入的信号状态为“0”，则不执行该运算。

放置

“置位位域”运算可以放置在程序段的右端。

实例

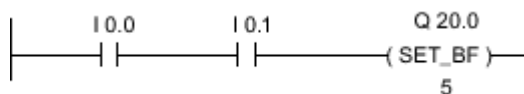


图 7-2

如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则置位输出 Q 20.0、Q 20.1、Q 20.2、Q 20.3 和 Q 20.4。

参见

插入 LAD 元素 (页 417)
更改 LAD 元素 (页 425)
将操作数插入 LAD 指令 (页 430)

7.8.1.1 RESET_BF：复位位数组

符号

<Operand1>
 —(RESET_BF)
 <Operand2>

参数	数据类型	存储区	描述
<操作数 1>	BOOL	I、Q、M、D、L (页 293)	指向要复位的第一个位的指针
<操作数 2>	UINT	常数	要复位的位位数

描述

可以使用“复位位域”运算复位从某个地址开始的多个位。要复位的位数决定 <操作数 2> 的值。要复位的第一个位的地址用 <操作数 1> 定义。如果 <操作数 2> 的值大于所选字节中的位数，则将复位下一个字节的位。在明确置位这些位（例如，通过另一个运算）之前，它们会保持复位。

只有使能输入的信号状态为“1”时，才执行该运算。如果输入的信号状态为“0”，则不会执行该运算。

放置

“复位位域”运算可以放置在程序段的右端。

实例

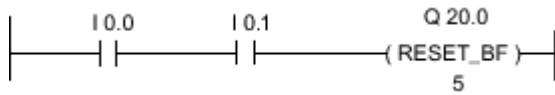


图 7-2

如果输入 I0.0 与 I0.1 的信号状态为“1”，则复位输出 Q 20.0、Q 20.1、Q 20.2、Q 20.3 和 Q 20.4。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)

7.8.1.1 SR：置位复位触发器

符号

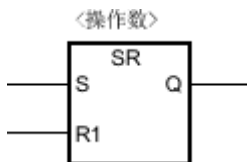


图 7-2

参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、L、D (页 293)	指定的操作数。
S	BOOL	I、Q、M、L、D	置位使能
R1	BOOL	I、Q、M、L、D	复位使能
Q	BOOL	I、Q、M、L、D	指定操作数的信号状态

描述

可以使用“置位复位触发器”运算基于输入 S 和 R1 的信号状态置位或复位指定操作数的位。如果输入 S 的信号状态为“1”且输入 R1 的信号状态为“0”，指定的操作数将置位为“1”。如果输入 S 的信号状态为“0”且输入 R1 的信号状态为“1”，指定的操作数将复位为“0”。

输入 R1 优先于输入 S。如果两个输入 S 和 R1 的信号状态均为“1”，指定操作数的信号状态将复位为“0”。

如果两个输入 R1 和 S 的信号状态为“0”，则不执行该运算。操作数的信号状态因此保持不变。操作数的当前信号状态传送到输出 Q 并且可供查询。

放置

“置位复位触发器”运算可以放置在程序段的右端。

实例

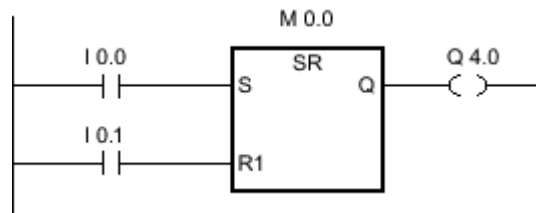


图 7-2

满足以下条件时，将置位存储器 M 0.0 和输出 Q 4.0：

- 输入 I 0.0 的信号状态为“1”。
- 输入 I 0.1 的信号状态为“0”。

满足下列条件之一时，将复位存储器 M 0.0 和输出 Q 4.0：

- 输入 I 0.0 的信号状态为“0”且输入 I 0.1 的信号状态为“1”。
- 输入 I 0.0 和 I 0.1 的信号状态均为“1”。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

7.8.1.1 RS：复位置位触发器

符号

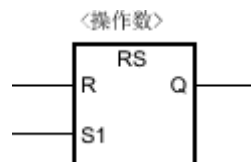


图 7-2

参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、L、D (页 293)	指定的操作数。
R	BOOL	I、Q、M、L、D	复位使能
S1	BOOL	I、Q、M、L、D	置位使能
Q	BOOL	I、Q、M、L、D	指定操作数的信号状态

描述

可以使用“复位置位触发器”运算基于输入 R 和 S1 的信号状态置位或复位指定操作数的位。如果输入 R 的信号状态为“1”且输入 S1 的信号状态为“0”，指定的操作数将复位为“0”。如果输入 R 的信号状态为“0”且输入 S1 的信号状态为“1”，指定的操作数将置位为“1”。

输入 S1 优先于输入 R。如果两个输入 R 和 S1 的信号状态均为“1”，指定操作数的信号状态将置位为“1”。

如果两个输入 R 和 S1 的信号状态为“0”，则不执行该运算。操作数的信号状态因此保持不变。操作数的当前信号状态传送到输出 Q 并且可供查询。

放置

“复位置位触发器”运算可以放在程序段的右端。

实例

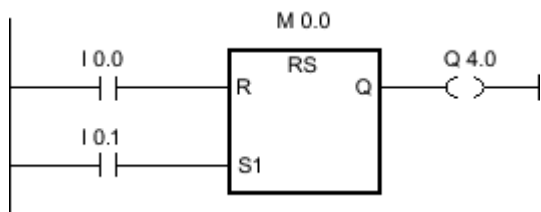


图 7-2

满足以下条件时，将置位存储器 M 0.0 和输出 Q 4.0：

- 输入 I 0.0 的信号状态为“0”且输入 I 0.1 的信号状态为“1”。
- 输入 I 0.0 和 I 0.1 的信号状态均为“1”。

满足以下条件时，将复位存储器 M 0.0 和输出 Q 4.0：

- 输入 I 0.0 的信号状态为“1”。
- 输入 I 0.1 的信号状态为“0”。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

7.8.1.1 --[P]--：扫描操作数的信号上升沿

符号

<操作数 1>

--[P]--

<操作数 2>

参数	数据类型	存储区	描述
<操作数 1>	BOOL	I、Q、M、L、D (页 293)	要查询的信号
<操作数 2>	BOOL	I、Q、M、L、D	保存上一次查询的信号状态的沿存储器位。

描述

可以使用“扫描操作数的信号上升沿”运算确定指定操作数 (<操作数 1>) 的信号状态是否从“0”变为“1”。该运算比较 <操作数 1> 的当前信号状态与保存在 <操作数 2> 中的上一次查询的信号状态。如果该运算检测到逻辑运算结果从“0”变为“1”，则说明出现了上升沿。

如果检测到上升沿，运算的输出的信号状态将为“1”。在任何其它情况下，运算的输出的信号状态为“0”。

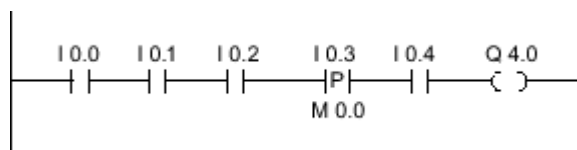
注意

沿存储器位的地址在程序中最多只能使用一次，否则，该存储器位会被覆盖。这将影响沿评估，并且结果会变得不明确。

放置

“扫描操作数的信号上升沿”运算可以放置在程序段中的任何位置。

实例



满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 以及 I 0.2 的信号状态为“1”。
- 输入 I 0.3 有上升沿。
- 输入 I 0.4 的信号状态为“1”。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

检测传送带方向实例 (页 560)

7.8.1.1 --|N|-- : 扫描操作数的信号下降沿

符号

<操作数 1>

--|N|--

<操作数 2>

参数	数据类型	存储区	描述
<操作数 1>	BOOL	I、Q、M、L、D (页 293)	要查询的信号
<操作数 2>	BOOL	I、Q、M、L、D	保存上一次查询的信号状态的沿存储器位。

描述

可以使用“扫描操作数的信号下降沿”运算查询指定操作数 (<操作数 1>) 的信号状态是否从“1”变为了“0”。该运算比较 <操作数 1> 的当前信号状态与保存在 <操作数 2> 中的上一次查询的信号状态。如果该运算检测到逻辑运算结果从“1”变为“0”，则说明存在下降沿。

如果检测到下降沿，运算的输出的信号状态将为“1”。

如果没有下降沿，到右侧电源线的信号状态将复位为“0”。

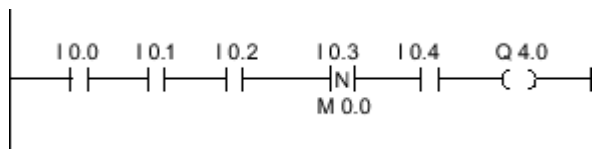
注意

沿存储器位的地址在程序中最多只能使用一次，否则，该存储器位会被覆盖。这将影响沿评估，并且结果会变得不明确。

放置

“扫描操作数的信号下降沿”运算可以放置在程序段中的任何位置。

实例



满足下列条件时，将置位输出 Q 4.0：

- 输入 I0.0 与 I0.1 以及 I0.2 的信号状态为“1”。
- 输入 I0.3 有下降沿。
- 输入 I0.4 的信号状态为“1”。

参见

插入 LAD 元素 (页 417)
 更改 LAD 元素 (页 425)
 将操作数插入 LAD 指令 (页 430)

7.8.1.1 --**符号**

<操作数 1>
 — (P) —
 <操作数 2>

参数	数据类型	存储区	描述
<操作数 1>	BOOL	I、Q、M、D、L (页 293)	上升沿置位的操作数。
<操作数 2>	BOOL	I、Q、M、D	沿存储器位

描述

可以使用“在信号上升沿置位操作数”运算在能流从“0”变为“1”时置位指定操作数。该运算将当前逻辑运算结果 (RLO) 与保存在沿位存储器中的上次查询的运算结果进行比较。如果该运算检测到能流从“0”变为“1”，则说明出现了上升沿。

如果出现了上升沿，<操作数 1> 的信号状态将置位为“1”一个程序周期。在任何其它情况下，操作数的信号状态为“0”。

注意

沿存储器位的地址在程序中最多只能使用一次，否则，该存储器位会被覆盖。这将影响沿评估，并且结果会变得不明确。

该运算不会影响 RLO。线圈输入端的 RLO 将立即发送到线圈输出端。

放置

“在信号上升沿置位操作数”运算可以放置在程序段的中间或者右端。通过使用分支，多个线圈可以相互嵌套放置。

实例

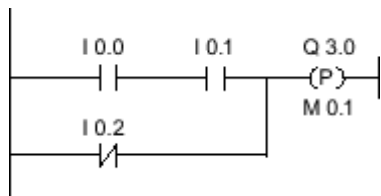


图 7-2

线圈输入端的信号状态从“0”切换到“1”时（上升沿），输出 Q 3.0 将置位一个程序周期。在任何其它情况下，输出 Q 3.0 的信号状态为“0”。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)

7.8.1.1 --

符号

<操作数 1>
— (N) —
<操作数 2>

参数	数据类型	存储区	描述
<操作数 1>	BOOL	I、Q、M、D、L (页 293)	下降沿置位的操作数。
<操作数 2>	BOOL	I、Q、M、D	沿存储器位

描述

可以使用“在信号下降沿置位操作数”运算在能流从“1”变为“0”时置位指定操作数。该运算将当前逻辑运算结果 (RLO) 与保存在沿位存储器中的上次查询的运算结果进行比较。如果该运算检测到能流从“1”变为“0”，则说明出现了下降沿。

检测到下降沿时，<操作数 1> 的信号状态将置位为“1”一个程序周期。在任何其它情况下，操作数的信号状态为“0”。

注意

沿存储器位的地址在程序中最多只能使用一次，否则，该存储器位会被覆盖。这将影响沿评估，并且结果会变得不明确。

该运算不会影响 RLO。线圈输入端的 RLO 将立即发送到线圈输出端。

放置

“在信号下降沿置位操作数”运算可以放置在程序段的中间或者右端。通过使用分支，多个线圈可以相互嵌套放置。

实例

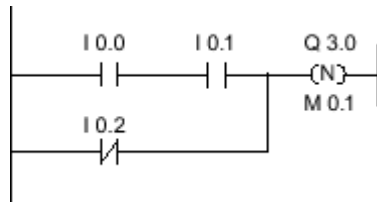


图 7-2

线圈输入端的信号状态从“1”切换到“0”时（下降沿），输出 Q 3.0 将置位一个程序周期。在任何其它情况下，输出 Q 3.0 的信号状态为“0”。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

7.8.1.1 P_TRIG：在信号上升沿置位输出

符号



参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、L、D (页 293)	保存上一次查询的 RLO 的沿存储器位。
CLK	BOOL		当前 RLO
Q	BOOL		沿评估的结果

描述

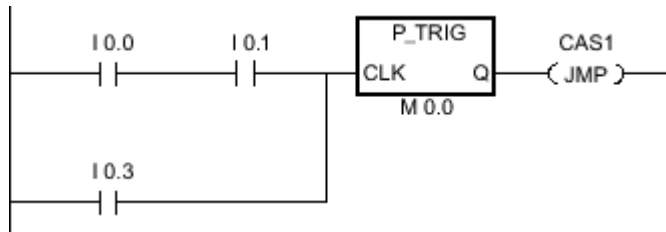
您可以使用“在信号上升沿置位输出”运算查询逻辑运算结果的信号状态是否从“0”变为了“1”。该运算比较逻辑运算结果 (RLO) 的当前信号状态与保存在沿位存储器中的上一次查询的信号状态。如果该运算检测到 RLO 从“0”变为“1”，则说明出现了上升沿。

如果检测到上升沿，运算的输出的信号状态将为“1”。在任何其它情况下，运算的输出的信号状态为“0”。

放置

“在信号上升沿置位输出”运算可以放置在程序段的中间。

实例



全部逻辑运算的 RLO 均保存在沿位存储器 M 0.0 中。如果 RLO 的信号状态从“1”变为“0”，则程序将跳转到跳转标签 CAS1。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)

7.8.1.1 N_TRIG : 在信号下降沿置位输出

符号



参数	数据类型	存储区	描述
<操作数>	BOOL	I、Q、M、L、D (页 293)	保存上一次查询的 RLO 的沿存储器位。
CLK	BOOL		当前 RLO
Q	BOOL		沿评估的结果

描述

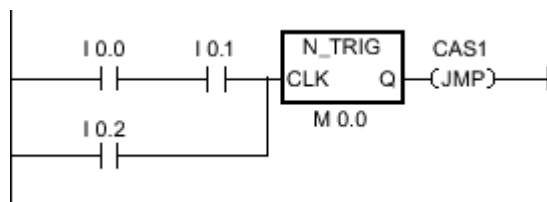
可以使用“在信号下降沿置位输出”运算检测逻辑运算结果 (RLO) 的信号状态是否从“1”变为“0”。该运算比较逻辑运算结果的当前信号状态与保存在沿存储器位中的上一次查询的信号状态。如果该运算检测到 RLO 从“1”变为“0”，则说明存在下降沿。

如果检测到下降沿，运算的输出的信号状态将为“1”。在任何其它情况下，运算的输出的信号状态为“0”。

放置

“在信号下降沿置位输出”运算可以放置在程序段中间。

实例



前导位逻辑运算的 RLO 保存在沿存储器位 M 0.0 中。如果检测到 RLO 的信号状态从“1”变为“0”，程序将跳转到跳转标签 CAS1。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

7.8.1.1 定时器

7.8.1.1 TP：脉冲生成

符号

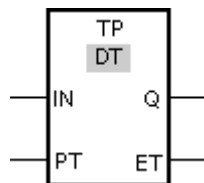


图 7-2

参数	数据类型	存储区	描述
IN	BOOL	I、Q、M、D、L (页 293)	启动输入
PT	TIME	I、Q、M、D、L 或常数	脉冲的持续时间。PT 必须为正值。
Q	BOOL	I、Q、M、D、L	脉冲输出
ET	TIME	I、Q、M、D、L	已耗时间

描述

可以使用“生成脉冲”操作置位输出 Q 一段预设的时间。当输入 IN 的逻辑运算结果 (RLO) 从“0”变为“1”时，启动该操作。启动该操作后，为 PT 编程的时间即开始计时。无论输入信号接下来是什么，输出 Q 都将置位一段时间，即 PT。即使检测到新的上升沿，只要 PT 在计时，输出 Q 的信号状态就不会受到影响。

可以通过输出 ET 查询当前定时器功能已运行了多长时间。该时间从 T#0s 开始计时并在达到为 PT 定时器设置的值后结束。只要定时器 PT 在计时且输入 IN 的信号状态为“1”，就可以查询输出 ET 的值。

插入“生成脉冲”操作时，将创建一个保存操作数据的背景数据块。

放置

“生成脉冲”操作需要一个用于沿评估的前导逻辑运算。该运算可以放置在程序段的中间或者末端。

脉冲图

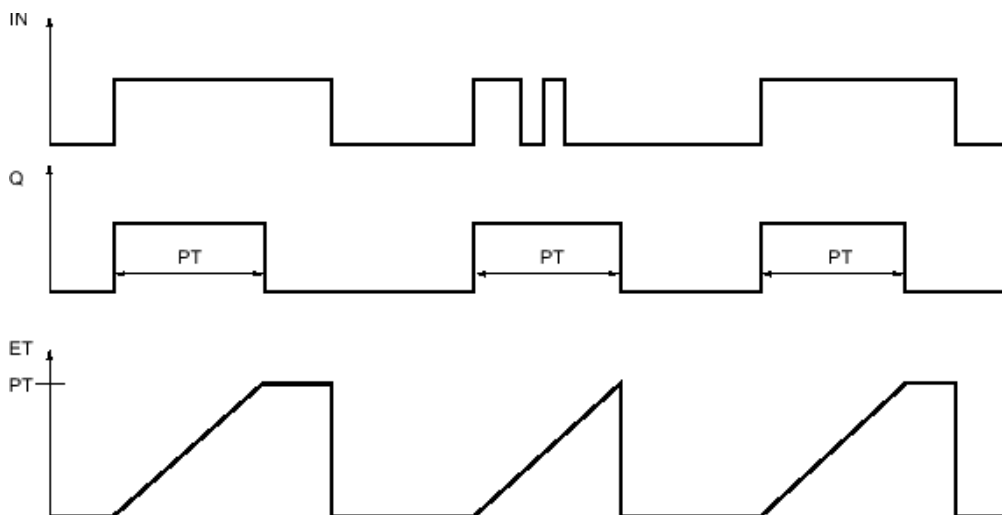


图 7-2

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 将操作数插入 LAD 指令 (页 430)*
- 选择 LAD 元素的数据类型 (页 419)*

7.8.1.1 TON : 接通延迟

符号

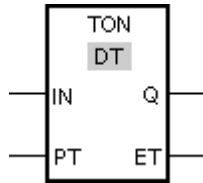


图 7-2

参数	数据类型	存储区	描述
IN	BOOL	I、Q、M、D、L (页 293)	启动输入
PT	TIME	I、Q、M、D、L 或常数	输入 IN 上升沿延迟的时间。
Q	BOOL	I、Q、M、D、L	延迟了时间 PT 的输出。
ET	TIME	I、Q、M、D、L	已耗时间

描述

可以使用“接通延迟”操作使上升沿延迟 PT 端设置的时间。输入 IN 的逻辑运算结果 (RLO) 从“0”变为“1”(上升沿) 时, 执行“接通延迟”操作。启动该操作后, 为 PT 设置的时间即开始计时。PT 时间用完时, 输出 Q 的信号状态为“1”。只要启动输入仍为“1”, 输出 Q 就保持置位。如果启动输入的信号状态从“1”变为“0”, 则复位输出 Q。在启动输入端检测到新的上升沿时, 该定时器功能将再次启动。

输出 ET 提供从输入 IN 的最近一次上升沿后所耗时间。该时间从 T#0s 开始计时并在达到为 PT 定时器设置的值后结束。只要输入 IN 的信号状态为“1”, 就可通过输出 ET 查询已耗时间。

插入“接通延迟”操作时, 将创建一个保存操作数据的背景数据块。

放置

“接通延迟”操作需要一个用于沿评估的前导逻辑运算。该运算可以放置在程序段的中间或者末端。

脉冲图

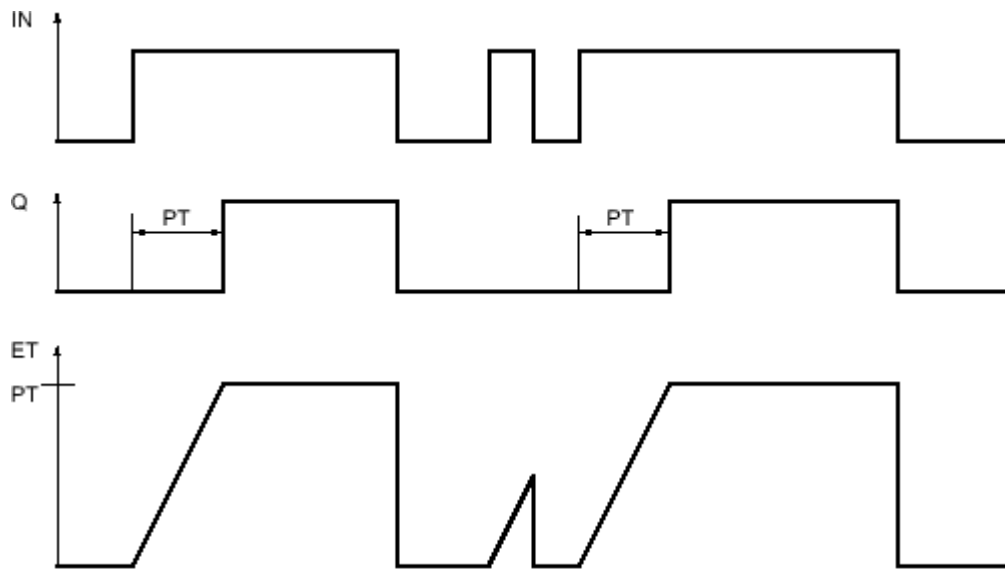


图 7-2

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 选择 LAD 元素的数据类型 (页 419)
- 将操作数插入 LAD 指令 (页 430)
- 控制室温实例 (页 565)

7.8.1.1 TOF : 关断延迟

符号

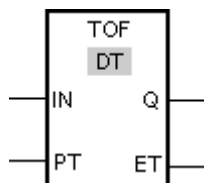


图 7-2

参数	数据类型	存储区	描述
IN	BOOL	I、Q、M、D、L (页 293)	启动输入
PT	TIME	I、Q、M、D、L 或常数	输入 IN 下降沿延迟的时间。PT 必须为正值。
Q	BOOL	I、Q、M、D、L	延迟了时间 PT 的输出。
ET	TIME	I、Q、M、D、L	已耗时间

描述

可以使用“关断延迟”操作使下降沿延迟 PT 端设置的时间。输入 IN 的逻辑运算结果 (RLO) 从“0”变为“1”时，输出 Q 置位。当输入 IN 的信号状态切换回“0”时，PT 端设置的时间开始计时。只要 PT 端设置的时间在运行，输出 Q 就保持置位。PT 时间用完时，输出 Q 将复位。如果输入 IN 的信号状态在 PT 端设置的时间用完前变为“1”，则定时器将复位。输出 Q 的信号状态仍将为“1”。

可以通过输出 ET 查询当前定时器功能已运行了多长时间。该时间从 T#0s 开始计时并在达到为 PT 定时器设置的值后结束。PT 端设置的时间用完时，输出 ET 保持设置为当前值，直到输入 IN 重新变回“1”。如果输入 IN 在时间 PT 用完前变为“1”，输出 ET 将复位为值 T#0。

插入“关断延迟”操作时，将创建一个保存操作数据的背景数据块。

放置

“关断延迟”操作需要一个用于沿评估的前导逻辑运算。该运算可以放置在程序段的中间或者末端。

脉冲图

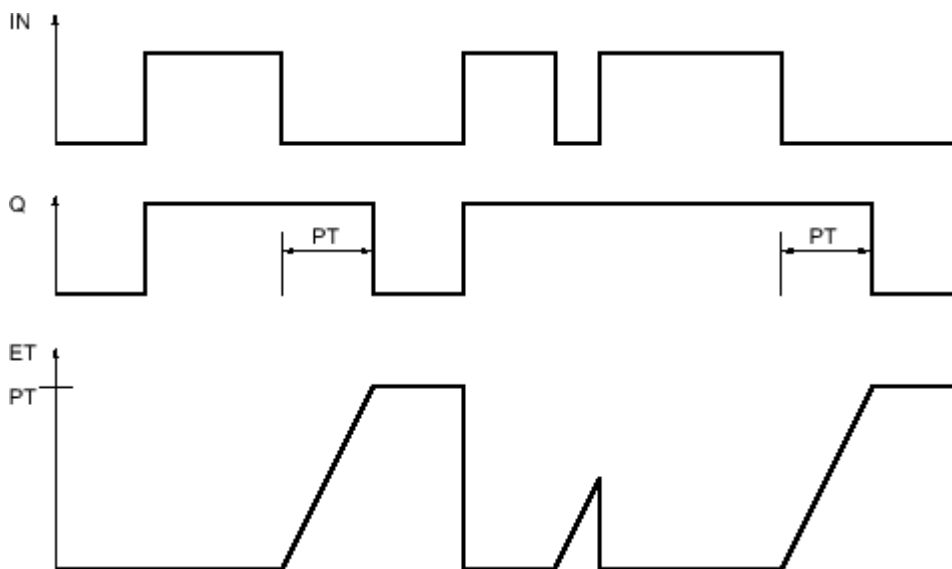


图 7-2

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 选择 LAD 元素的数据类型 (页 419)
- 将操作数插入 LAD 指令 (页 430)

7.8.1.1 TONR : 时间累加器

符号

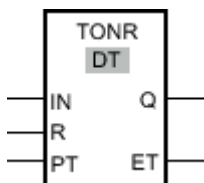


图 7-2

参数	数据类型	存储区	描述
IN	BOOL	I、Q、M、D、L (页 293)	启动输入
R	BOOL	I、Q、M、D、L	复位输入
PT	TIME	I、Q、M、D、L 或常数	时间记录的最长持续时间
Q	BOOL	I、Q、M、D、L	PT 时间用完时置位的输出。
ET	TIME	I、Q、M、D、L	累计的时间

描述

可以使用“时间累加器”操作在参数 PT 设置的时段内累计时间值。输入 IN 的信号状态变为“1”时，执行该操作且 PT 端设置的时间开始计时。在 PT 端设置的时间计时期间，累加输入 IN 的信号状态为“1”时记录的时间值。累计的时间写入到输出 ET 并可在其中查询。PT 时间用完时，输出 Q 的信号状态为“1”。

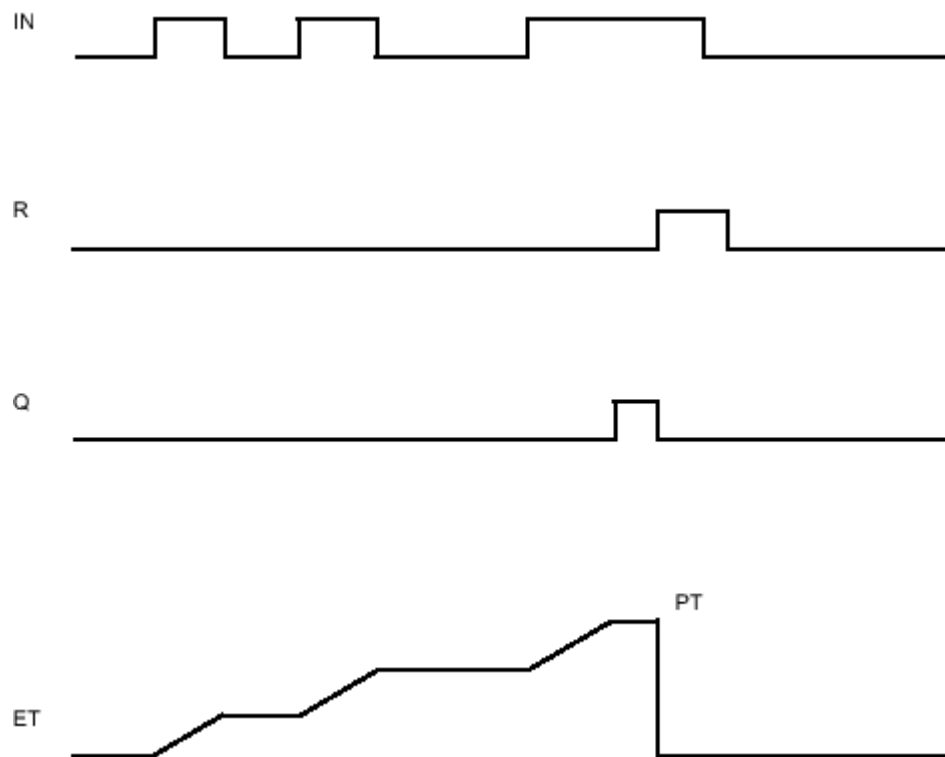
无论启动输入端的信号状态是什么，输入 R 都会复位输出 ET 和 Q。

插入“时间累加器”操作时，将创建一个保存操作数据的背景数据块。

放置

“时间累加器”操作需要一个前导逻辑运算。该运算可以放置在程序段的中间或者末端。

脉冲图



参见

[插入 LAD 元素 \(页 417\)](#)
[更改 LAD 元素 \(页 425\)](#)
[选择 LAD 元素的数据类型 \(页 419\)](#)
[将操作数插入 LAD 指令 \(页 430\)](#)

7.8.1.1 计数器

7.8.1.1 CTU：加计数

符号

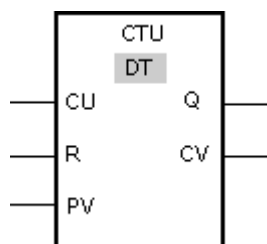


图 7-2

参数 英语	数据类型	存储区	描述
CU	BOOL	I、Q、M、D、L (页 293)	计数输入
R	BOOL	I、Q、M、D、L	复位输入
PV	SINT、INT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L 或常数	预置计数值
Q	BOOL	I、Q、M、D、L	计数器状态
CV	SINT、INT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L	实际计数值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“加计数”操作使输出 CV 的值递增。如果输入 CU 的信号状态从“0”变为“1”(上升沿), 则执行该操作且输出 CV 的当前计数值加“1”。首次执行该操作时, 输出 CV 的当前计数值设置为“0”。每检测到一个上升沿, 计数值就会递增, 直到其达到输出 CV 指定数据类型的上限。达到上限时, 输入 CU 的信号状态就不再影响该操作。

可通过输出 Q 查询计数器状态。输出 Q 的信号状态由输入参数 PV 决定。如果当前计数值大于或等于参数 PV 的值, 则输出 Q 的信号状态设置为“1”。在任何其它情况下, 输出 Q 的信号状态为“0”。

输入 R 的信号状态变为“1”时, 输出 CV 的值复位为“0”。只要输入 R 的信号状态为“1”, 输入 CU 的信号状态就不会影响该操作。

插入“加计数”操作时, 将创建一个保存操作数据的背景数据块。

放置

“加计数”操作需要一个用于沿评估的前导逻辑运算。该运算可以放置在程序段的中间或者末端。

实例

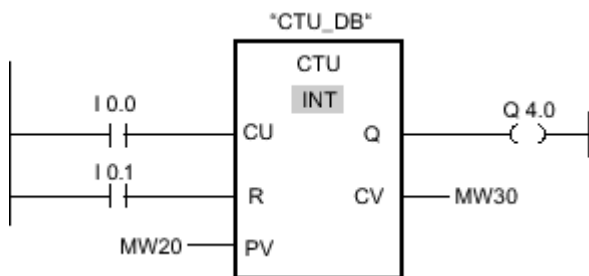


图 7-2

当输入 I0.0 的信号状态从“0”变为“1”时, 执行“加计数”操作且输出 MW30 的当前计数值加“1”。每检测到一个上升沿, 计数值就会递增, 直到达到指定数据类型的上限值 (32767)。

参数 MW20 中的值用于确定输出 Q 4.0 的上限。只要当前计数值大于或等于参数 MW20 中的值, 输出 Q 4.0 的信号状态就为“1”。在任何其它情况下, 输出 Q 4.0 的信号状态为“0”。

参见

插入 LAD 元素 (页 417)
 更改 LAD 元素 (页 425)
 选择 LAD 元素的数据类型 (页 419)
 将操作数插入 LAD 指令 (页 430)

7.8.1.1 CTD : 减计数

符号

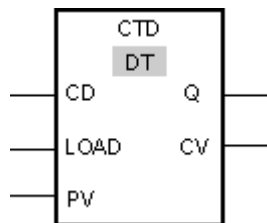


图 7-2

参数 英语	数据类型	存储区	描述
CD	BOOL	I、Q、M、D、L (页 293)	计数输入
LOAD	BOOL	I、Q、M、D、L	装载输入
PV	SINT、UINT、DINT、USINT、UINT、UDINT	I、Q、M、D、L 或常数	预置计数值
Q	BOOL	I、Q、M、D、L	计数器状态
CV	SINT、UINT、DINT、USINT、UINT、UDINT	I、Q、M、D、L	实际计数值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“减计数”操作使输出 CV 的值递减。如果输入 CD 的信号状态从“0”变为“1”(上升沿), 则执行该操作且输出 CV 的当前计数值减“1”。首次执行该操作时, 输出 CV 的当前计数值设置为“0”。每检测到一个上升沿, 计数值就会递减, 直到达到指定数据类型的下限值。达到下限值时, 输入 CD 的信号状态就不再影响该操作。

可通过输出 Q 查询计数器状态。如果当前计数值小于或等于“0”, 则输出 Q 的信号状态设置为“1”。在任何其它情况下, 输出 Q 的信号状态为“0”。

输入 LOAD 的信号状态变为“1”时, 输出 CV 的值将设置为参数 PV 的值。只要输入 LOAD 的信号状态为“1”, 输入 CD 的信号状态就不会影响该操作。

插入“减计数”操作时, 将创建一个保存操作数据的背景数据块。

放置

“减计数”操作需要一个用于沿评估的前导逻辑运算。该运算可以放置在程序段的中间或者末端。

实例

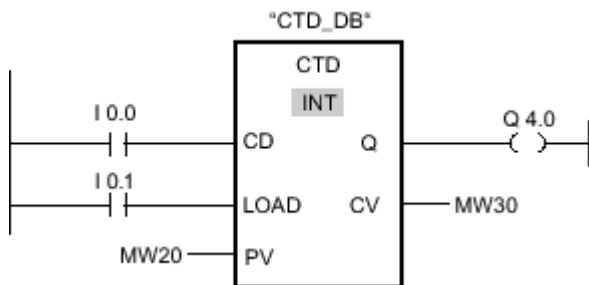


图 7-2

如果输入 I 0.0 的信号状态从“0”变为“1”，则执行“减计数”操作且输出 MW30 的值减“1”。每检测到一个上升沿，计数值就会递减，直到达到指定数据类型的下限值 (-32768)。

只要当前计数值小于或等于“0”，输出 Q 4.0 的信号状态就为“1”。在任何其它情况下，输出 Q 4.0 的信号状态为“0”。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 选择 LAD 元素的数据类型 (页 419)*
- 将操作数插入 LAD 指令 (页 430)*

7.8.1.1 CTUD : 加计数和减计数

符号

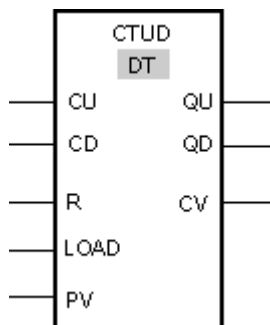


图 7-2

参数 英语	数据类型	存储区	描述
CU	BOOL	I、Q、M、D、L (页 293)	加计数输入
CD	BOOL	I、Q、M、D、L	减计数输入
R	BOOL	I、Q、M、D、L	复位输入
LOAD	BOOL	I、Q、M、D、L	装载输入
PV	SINT、UINT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L 或常数	预置计数值
QU	BOOL	I、Q、M、D、L	递增计数器的状态
QD	BOOL	I、Q、M、D、L	递减计数器的状态
CV	SINT、UINT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L	实际计数值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“加计数和减计数”操作使输出 CV 的计数值递增或递减。如果输入 CU 的信号状态从“0”变为“1”(上升沿),则当前计数值加“1”并存储在输出 CV 中。如果输入 CD 的信号状态从“0”变为“1”(上升沿),则输出 CV 的计数值减“1”。如果在一个程序周期内输入 CU 和 CD 都有上升沿,则输出 CV 的当前计数值保持不变。

计数值可以一直递增,直到其达到输出 CV 指定数据类型的上限值。如果达到上限值,出现了上升沿时计数值也不再递增。达到指定数据类型的下限值时,计数值不再递减。

输入 LOAD 的信号状态变为“1”时,输出 CV 的计数值将设置为参数 PV 的值。只要输入 LOAD 的信号状态变为“1”,输入 CU 和 CD 的信号状态就不会影响该操作。

输入 R 的信号状态变为“1”时,计数值设置为“0”。只要输入 R 的信号状态变为“1”,输入 CU 和 CD 和 LOAD 的信号状态变化都不能影响“加计数和减计数”操作。

可通过输出 QU 查询递增计数器状态。如果当前计数值大于或等于参数 PV 的值,则输出 QU 的信号状态为“1”。在任何其它情况下,输出 QU 的信号状态为“0”。

可通过输出 QD 查询递减计数器状态。如果当前计数值小于或等于“0”,则输出 QD 的信号状态为“1”。在任何其它情况下,输出 QD 的信号状态为“0”。

插入“加计数和减计数”操作时,将创建一个保存操作数据的背景数据块。

放置

“加计数和减计数”操作需要一个用于沿评估的前导逻辑运算。该运算可以放置在程序段的中间或者末端。

实例

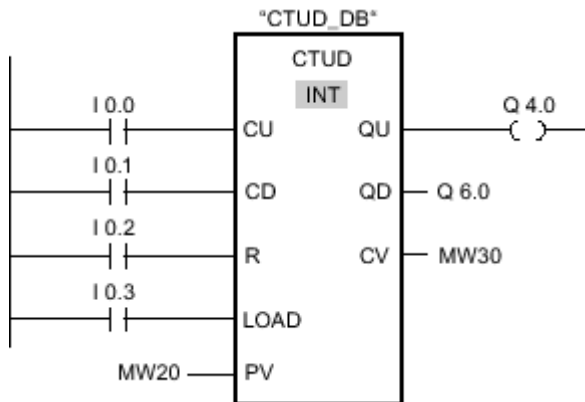


图 7-2

如果输入 I0.0 或输入 I0.1 的信号状态从“0”变为“1”(上升沿),则执行“加计数和减计数”操作。输入 I0.0 有上升沿时,当前计数值加“1”并存储在输出 MW30 中。输入 I0.1 有上升沿时,当前计数值减“1”并存储在输出 MW30 中。输入 CU 每出现一个上升沿,计数值将递增,直到其达到上限值 32767 为止。输入 I0.1 每出现一个上升沿,计数值将递减,直到其达到下限值 -32768 为止。

只要当前计数值大于或等于输入 MW20 中的值,输出 Q 4.0 的信号状态即为“1”。在任何其它情况下,输出 Q 4.0 的信号状态为“0”。

只要当前计数值小于或等于“0”,输出 Q 6.0 的信号状态就为“1”。在任何其它情况下,输出 Q 6.0 的信号状态为“0”。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 选择 LAD 元素的数据类型 (页 419)
- 将操作数插入 LAD 指令 (页 430)
- 检测存储区域的填充量实例 (页 562)

7.8.1.1 高速计数器

7.8.1.1 CTRL_HSC : 在 LAD 中控制高速计数器

符号

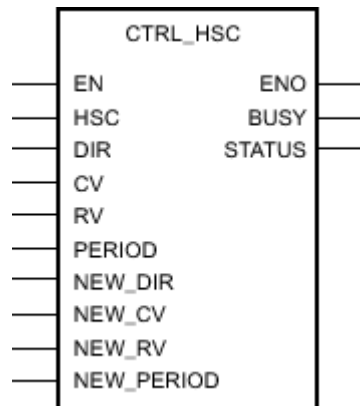


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
HSC	HW_HSC	L、D 或常数	高速计数器的硬件地址 (HW-ID)
DIR	BOOL	I、Q、M、L、D	启用新的计数方向 (请参见 NEW_DIR)
CV	BOOL	I、Q、M、L、D	启用新的计数值 (请参见 NEW_CV)
RV	BOOL	I、Q、M、L、D	启用新的参考值 (请参见 NEW_RV)
PERIOD	BOOL	I、Q、M、L、D	启用新的频率测量周期 (请参见 NEW_PERIOD)
NEW_DIR	INT	I、Q、M、L、D	DIR = TRUE 时装载的计数方向。
NEW_CV	DINT	I、Q、M、L、D	CV = TRUE 时装载的计数值。
NEW_RV	DINT	I、Q、M、L、D	CV = TRUE 时装载的参考值。
NEW_PERIOD	INT	I、Q、M、L、D	PERIOD = TRUE 时装载的频率测量周期。

参数	数据类型	存储区	描述
BUSY	BOOL	I、Q、M、L、D	处理状态
STATUS	WORD	I、Q、M、L、D	运行状态

描述

使用“控制高速计数器”操作，可以通过将新值装载到计数器来进行参数设置和控制 CPU 支持的高速计数器。只有在硬件配置中启用要控制的高速计数器后，才能无错误地执行该操作。在程序中每个高速计数器只能插入和执行一个“控制高速计数器”操作。用户需要输入高速计数器的硬件标识符 (HW-ID)，在输入 HSC 中分配它的值。

在输入 HSC 只能指定数据类型为“HW_HSC”的变量。硬件数据类型“HW_HSC”的长度为一个 WORD。

使用“控制高速计数器”操作可以将以下参数值装载到高速计数器：

- 计数方向 (NEW_DIR)：计数方向定义高速计数器是加计数还是减计数。计数方向在输入 NEW_DIR 通过以下值来定义：1 = 加计数，-1 = 减计数。
只有通过程序在参数中设置方向控制后，才能使用“控制高速计数器”操作更改计数方向。
输入 NEW_DIR 指定的计数方向将在输入 DIR 的位置位时装载到高速计数器。
- 计数值 (NEW_CV)：计数值是高速计数器开始计数时使用的初始值。计数值的范围为 - 2147483648 到 2147483647。
输入 NEW_CV 指定的计数值会在输入 CV 的位置位时装载到高速计数器。
- 参考值 (NEW_RV)：参考值是高速计数器可以达到的最大值。参考值的范围为 - 2147483648 到 2147483647。
输入 NEW_RV 指定的参考值会在输入 RV 的位置位时装载到高速计数器。
- 频率测量周期 (NEW_PERIOD)：频率测量周期在输入 NEW_PERIOD 通过以下值来指定：10 = 0.01s，100 = 0.1s，1000 = 1s。
在指定高速计数器的参数中设置“频率测量”功能后，可更新该周期。
输入 NEW_PERIOD 指定的周期值在输入 PERIOD 的位置位时装载到高速计数器。

只有输入 EN 的信号状态为“1”时，才执行“控制高速计数器”操作。只要该操作在执行，输出 BUSY 的位就会置位。该操作执行完后，输出 BUSY 的位立即复位。

只有输入 EN 的信号状态为“1”且执行该操作期间没有出错时，才置位使能输出 ENO。

插入“控制高速计数器”操作时，将创建一个保存操作数据的背景数据块。

参数 STATUS

通过输出 STATUS，可以查询“控制高速计数器”操作执行期间是否出错。下表列出了输出 STATUS 输出的值的含义：

错误代码 (十六进制)	描述
0	无错误
80A1	高速计数器的硬件标识符无效
80B1	计数方向 (NEW_DIR) 无效

错误代码 (十六进制)	描述
80B2	计数值 (NEW_CV) 无效
80B3	参考值 (NEW_RV) 无效
80B4	频率测量周期 (NEW_PERIOD) 无效
80C0	多次访问高速计数器

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

组态高速计数器 (页 243)

给硬件中断 OB 分配参数 (页 384)

7.8.1.1 比较

7.8.1.1 CMP == : 等于

符号



参数	数据类型	存储区	描述
操作数 1	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“等于”操作确定第一个比较值是否等于第二个比较值。要比较的两个值必须为相同的数据类型。

如果比较为真，则指令的逻辑运算结果为“1”。该 RLO 通过以下运算连接到当前整条路径的 RLO：

- 串联比较运算时，采用 AND。
- 并联比较运算时，采用 OR。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'AA'	'AA'	1
'Hello World'	'HelloWorld'	0
'AA'	'aa'	0

放置

“等于”运算可以放置在程序段的左端或中间。

实例

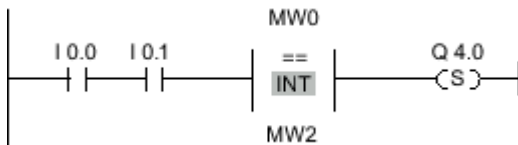


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 比较运算的结果为真 (MW0 = MW2)。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 将操作数插入 LAD 指令 (页 430)*
- 选择 LAD 元素的数据类型 (页 419)*

7.8.1.1 CMP <> : 不等于

符号



参数	数据类型	存储区	描述
操作数 1	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“不等于”运算确定第一个比较值与第二个比较值是否不相等。要比较的两个值必须为相同的数据类型。

如果比较为真，则 RLO 为“1”。该 RLO 通过以下运算连接到当前整条路径的 RLO：

- 串联比较运算时，采用 AND。
- 并联比较运算时，采用 OR。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'AA'	'aa'	1
'Hello World'	'HelloWorld'	1
'AA'	'AA'	0

放置

“不等于”运算可以放置在程序段的左端或者中间。

实例

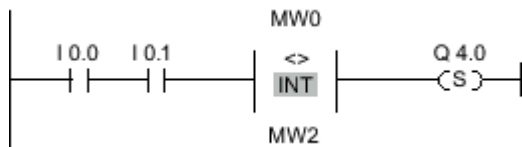


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I0.0 与 I0.1 的信号状态为“1”。
- 比较运算的结果为真 (MW0 <=> MW2)。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 CMP >= : 大于或等于

符号



参数	数据类型	存储区	描述
操作数 1	USINT、UINT、UDINT、SINT、INT、DINT、REAL、CHAR、STRING、TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、UDINT、SINT、INT、DINT、REAL、CHAR、STRING、TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“大于或等于”操作确定第一个比较值是否大于或等于第二个比较值。 要比较的两个值必须为相同的数据类型。

如果比较为真，则 RLO 为“1”。该 RLO 通过以下运算连接到当前整条路径的 RLO：

- 串联比较运算时，采用 AND。
- 并联比较运算时，采用 OR。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。如果较长字符串的左侧部分和较短字符串相同，则认为较长字符串更大。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'BB'	'AA'	1
'AAA'	'AA'	1
'Hello World'	'Hello World'	1
'Hello World'	'HelloWorld'	0
'AA'	'aa'	0
'AAA'	'a'	0

在比较时间值时，如果 <操作数 1> 端的时间大于（晚于）或等于 <操作数 2> 端的时间，则 RLO 为“1”。

放置

“大于或等于”运算可以放置在程序段的左端或者中间。

实例

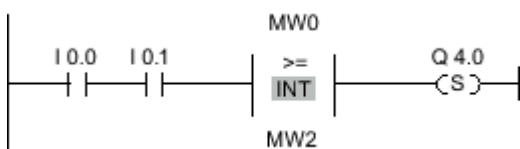


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 比较运算的结果为真 (MW0 >= MW2)。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

[检测存储区域的填充量实例 \(页 562\)](#)

7.8.1.1 CMP <= : 小于或等于

符号



参数	数据类型	存储区	描述
操作数 1	USINT、UINT、UDINT、SINT、INT、DINT、REAL、CHAR、STRING、TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、UDINT、SINT、INT、DINT、REAL、CHAR、STRING、TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“小于或等于”操作确定第一个比较值是否小于或等于第二个比较值。要比较的两个值必须为相同的数据类型。

如果比较为真，则 RLO 为“1”。该 RLO 通过以下运算连接到当前整条路径的 RLO：

- 串联比较运算时，采用 AND。
- 并联比较运算时，采用 OR。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。如果较长字符串的左侧部分和较短字符串相同，则认为较短字符串更小。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'AA'	'aa'	1
'AAA'	'a'	1
'Hello World'	'Hello World'	1
'HelloWorld'	'Hello World'	0
'BB'	'AA'	0
'AAA'	'AA'	0

在比较时间值时，如果 <操作数 1> 端的时间小于（早于）或等于 <操作数 2> 端的时间，则 RLO 为“1”。

放置

“小于或等于”运算可以放置在程序段的左端或者中间。

实例

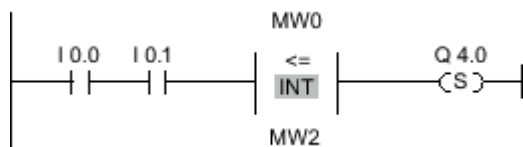


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 比较运算的结果为真 (MW0 <= MW2)。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

7.8.1.1 CMP > : 大于

符号



参数	数据类型	存储区	描述
操作数 1	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“大于”操作确定第一个比较值是否大于第二个比较值。要比较的两个值必须为相同的数据类型。

如果比较为真，则 RLO 为“1”。该 RLO 通过以下运算连接到当前整条路径的 RLO：

- 串联比较运算时，采用 AND。
- 并联比较运算时，采用 OR。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。如果较长字符串的左侧部分和较短字符串相同，则认为较长字符串更大。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'BB'	'AA'	1
'AAA'	'AA'	1
'AA'	'aa'	0
'AAA'	'a'	0

在比较时间值时，如果 <操作数 1> 端的时间大于（晚于）<操作数 2> 端的时间，则 RLO 为“1”。

放置

“大于”运算可以放置在程序段的左端或者中间。

实例

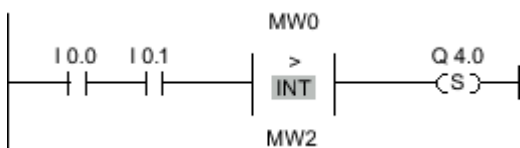


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 比较运算的结果为真 (MW0 > MW2)。

参见

- [插入 LAD 元素 \(页 417\)](#)
- [更改 LAD 元素 \(页 425\)](#)
- [将操作数插入 LAD 指令 \(页 430\)](#)

选择 LAD 元素的数据类型 (页 419)

7.8.1.1 CMP < : 小于

符号



参数	数据类型	存储区	描述
操作数 1	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“小于”运算确定第一个比较值是否小于第二个比较值。要比较的两个值必须为相同的数据类型。

如果比较为真，则 RLO 为“1”。该 RLO 通过以下运算连接到当前整条路径的 RLO：

- 串联比较运算时，采用 AND。
- 并联比较运算时，采用 OR。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。如果较长字符串的左侧部分和较短字符串相同，则认为较短字符串更小。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'AA'	'aa'	1
'AAA'	'a'	1
'BB'	'AA'	0

<操作数 1>	<操作数 2>	运算的 RLO
'AAA'	'AA'	0

在比较时间值时，如果 <操作数 1> 端的时间小于（早于）<操作数 2> 端的时间，则 RLO 为“1”。

放置

“小于”运算可以放置在程序段的左端或者中间。

实例

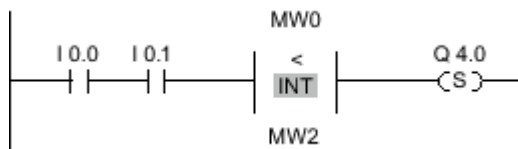


图 7-2

满足下列条件时，将置位输出 Q 4.0：

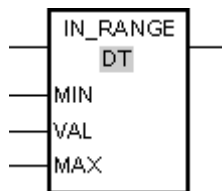
- 输入 I0.0 与 I0.1 的信号状态为“1”。
- 比较运算的结果为真 (MW0 < MW2)。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 将操作数插入 LAD 指令 (页 430)*
- 选择 LAD 元素的数据类型 (页 419)*

7.8.1.1 IN_RANGE：值在范围内

符号



参数	数据类型	存储区	描述
功能框输入	BOOL	I、Q、M、L、D (页 293)	上一逻辑运算的结果
功能框输出	BOOL	I、Q、M、L、D	比较结果

参数	数据类型	存储区	描述
MIN	SINT、INT、DINT、 USINT、UINT、 UDINT、REAL	I、Q、M、L、D 或常数	取值范围的下限
VAL	SINT、INT、DINT、 USINT、UINT、 UDINT、REAL	I、Q、M、L、D 或常数	比较值
MAX	SINT、INT、DINT、 USINT、UINT、 UDINT、REAL	I、Q、M、L、D 或常数	取值范围的上限

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“值在范围内”操作确定输入 VAL 的值是否在指定的值范围内。使用参数 MIN 和 MAX 指定取值范围的限值。查询时，“值在范围内”操作将输入 VAL 的值与参数 MIN 和 MAX 的值相比较并将结果发送到该功能框输出。如果输入 VAL 的值满足比较公式 $MIN \leq VAL \leq MAX$ ，则功能框输出的信号状态为“1”。如果该比较的结果不为真，则功能框输出的信号状态为“0”。

如果功能框输入的信号状态为“0”，则不执行“值在范围内”操作。

只有要比较的值具有相同的数据类型且互连了功能框输出时，才能执行该比较功能。

放置

“值在范围内”运算可以放置在程序段的左端或者中间。

实例

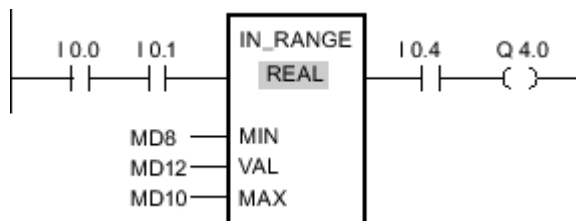


图 7-2

满足下列所有条件时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 MD12 的值在输入 MD8 和 MD10 的当前值所指定的取值范围内 ($MIN \leq VAL \leq MAX$)。
- 输入 I 0.4 的信号状态为“1”。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

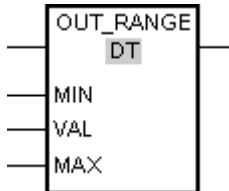
将操作数插入 LAD 指令 (页 430)

选择 LAD 元素的数据类型 (页 419)

GetError (页 887)

7.8.1.1 OUT_RANGE : 值超出范围

符号



参数	数据类型	存储区	描述
功能框输入	BOOL	I、Q、M、L、D (页 293)	上一逻辑运算的结果
功能框输出	BOOL	I、Q、M、L、D	比较结果
MIN	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	取值范围的下限
VAL	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	比较值
MAX	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	取值范围的上限

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“值超出范围”操作查询输入 VAL 的值是否超出指定范围。使用参数 MIN 和 MAX 指定取值范围的限值。查询时，“值超出范围”操作将输入 VAL 的值与参数 MIN 和 MAX 的值相比较并将结果发送到该功能框输出。如果输入 VAL 的值满足比较 $MIN > VAL$ 或 $VAL > MAX$ ，则功能框输出的信号状态为“1”。如果该比较的结果不为真，则功能框输出的信号状态为“0”。

如果功能框输入端的信号状态为“0”，则不执行“值超出范围”操作。

只有要比较的值具有相同的数据类型且互连了功能框输出时，才能执行该比较功能。

放置

“值超出范围”操作可以放置在程序段的左端或中间。

实例

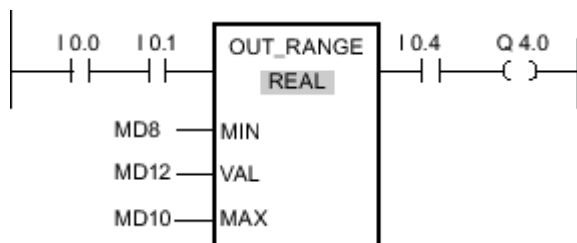


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 MD12 的值在由输入 MD8 和 MD10 的值所设定的取值范围之外 (MIN > VAL 或 VAL > MAX)。
- 输入 I 0.4 的信号状态为“1”。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

选择 LAD 元素的数据类型 (页 419)

GetError (页 887)

7.8.1.1 ---| OK |---：检查有效性

符号

<操作数>
—| OK |—

参数	数据类型	存储区	描述
<操作数>	REAL	I、Q、M、L、D (页 293) 或常数	要检查的值。

描述

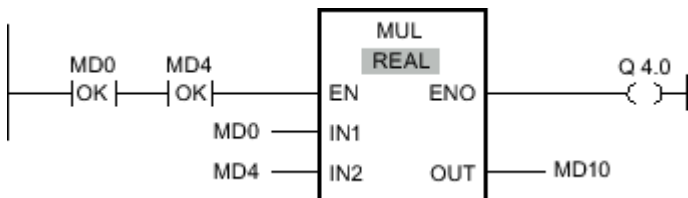
可使用“检查有效性”操作检查某个变量的值 (<操作数>) 是否为有效的浮点数。当操作的输入端信号状态为“1”时，每个程序循环都进行查询。当查询时该处的变量值是有效的浮点数时，右侧电源线的输出为信号状态“1”。在任何其它情况下，“检查有效性”操作的该输出的信号状态为“0”。

可以同时使用“检查有效性”操作和 EN 机制。如果将功能框 OK 连接到使能输入 EN，则只有值的有效性查询结果为正时，才会置位使能输入。可以使用此功能来确保只有指定变量的值为有效浮点数时才启用某个操作。

放置

“检查有效性”操作可以放置在程序段的左端或中间。

实例



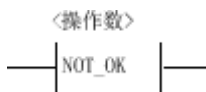
当 MD0 和 MD4 显示为有效的浮点数时，会激活“乘”(MUL) 运算并置位输出 ENO。执行“乘”(MUL) 运算时，输入 MD0 的值将与 MD4 的值相乘。相乘的结果存储在输出 MD10 中。如果操作执行过程中未发生错误，则输出 ENO 和 Q 4.0 的信号状态设置为“1”。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)

7.8.1.1 ---I NOT_OK I--- : 检查无效性

符号



参数	数据类型	存储区	描述
<操作数>	REAL	I、Q、M、L、D (页 293) 或常数	要检查的值。

描述

可使用“检查无效性”操作检查某个变量的值 (<操作数>) 是否为无效的浮点数。当操作的输入端信号状态为“1”时，每个程序循环都进行查询。当查询时此处的变量值是无效的浮点数且左侧电源线的信号状态为“1”时，右侧电源线的输出为信号状态“1”。在任何其它情况下，“检查无效性”操作的输出端的信号状态为“0”。

放置

“检查无效性”操作可以放置在程序段的左端或中间。

实例

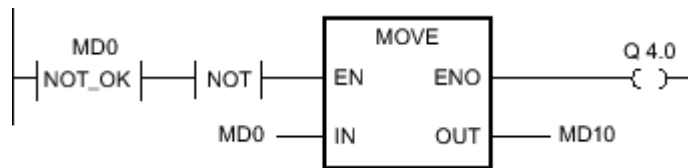


图 7-2

当输入 MD0 的值是无效浮点数时，则不执行“移动值”操作 (MOVE)。输出 Q 4.0 的信号状态被复位为“0”。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

7.8.1.1 数学

7.8.1.1 ADD : 加

符号

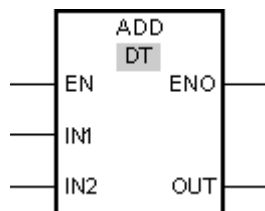


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	被加数
IN2	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	加数
OUT	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D	相加的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“加”运算将输入 IN1 的值与输入 IN2 的值相加并通过输出 OUT 查询和 (OUT =IN1 +IN2)。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 操作结果超出输出 OUT 指定的数据类型的允许范围。
- 输入变量为 REAL 数据类型且无效。

放置

“加”运算可以放置在程序段中的任何位置。

实例

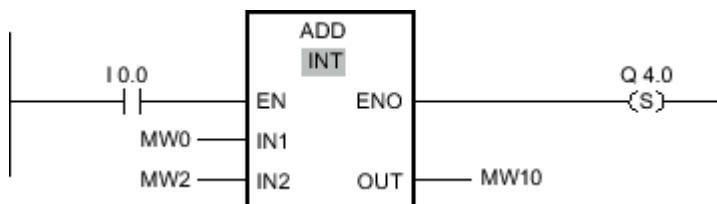


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“加”运算。输入 MW0 的值将与输入 MW2 的值相加。相加结果放在输出 MW10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 选择 LAD 元素的数据类型 (页 419)*
- 将操作数插入 LAD 指令 (页 430)*
- 插入附加输入和输出 (页 426)*
- 计算方程式实例 (页 564)*

7.8.1.1 SUB : 减

符号

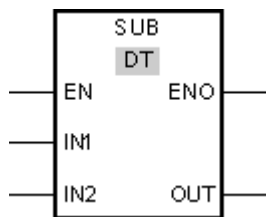


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	被减数
IN2	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	减数。
OUT	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D	相减的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可使用“减”运算从输入 IN1 的值中减去输入 IN2 的值并通过输出 OUT 查询差值 ($OUT = IN1 - IN2$)。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 操作结果超出输出 OUT 指定的数据类型的允许范围。
- 输入变量为 REAL 数据类型且无效。

放置

“减”运算可以放置在程序段中的任何位置。

实例

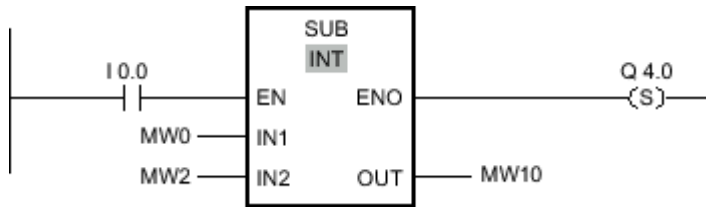


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“减”运算。从输入 MW0 的值中减去输入 MW2 的值。相减结果放在输出 MW10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 MUL : 乘

符号

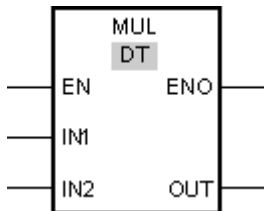


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	被乘数
IN2	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	乘数
OUT	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D	乘法结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可使用“乘”运算将输入 IN1 的值与输入 IN2 的值相乘并通过输出 OUT 查询乘积 ($OUT = IN1 * IN2$)。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 结果超出输出 OUT 指定的数据类型的允许范围。
- 输入变量为 REAL 数据类型且无效。

放置

“乘”运算可以放置在程序段中的任何位置。

实例

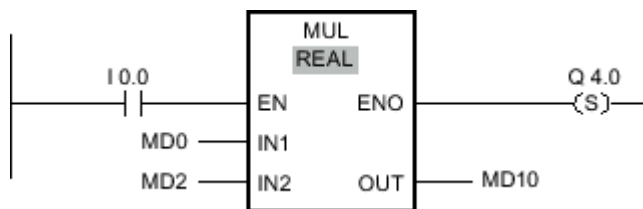


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“乘”运算。用输入 MD2 的值乘以输入 MD0 的值。相乘结果存储在输出 MD10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[插入附加输入和输出 \(页 426\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

[计算方程式实例 \(页 564\)](#)

7.8.1.1 DIV：除

符号

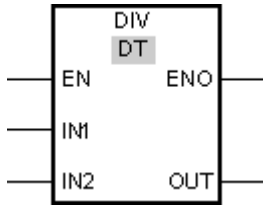


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	被除数
IN2	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	除数
OUT	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D	除法结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可使用“除”运算用输入 IN1 的值除以输入 IN2 的值并通过输出 OUT 查询商 (OUT =IN1/IN2)。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 操作结果超出输出 OUT 指定的数据类型的允许范围。
- 输入变量为 REAL 数据类型且无效。

放置

“除”运算可以放置在程序段中的任何位置。

实例

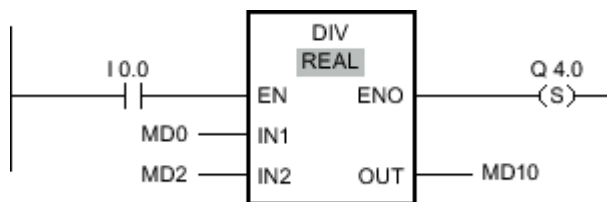


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“除”运算。用输入 MD0 的值除以输入 MD2 的值。相除结果放在输出 MD10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

[计算方程式实例 \(页 564\)](#)

7.8.1.1 MOD : 返回除法的余数

符号

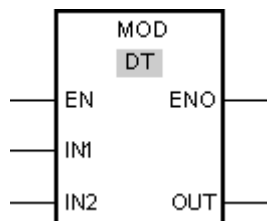


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、INT、DINT、USINT、UINT、UDINT	I、Q、M、L、D 或常数	被除数
IN2	SINT、INT、DINT、USINT、UINT、UDINT	I、Q、M、L、D 或常数	除数
OUT	SINT、INT、DINT、USINT、UINT、UDINT	I、Q、M、L、D	除法的余数

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可使用“返回除法的余数”运算用输入 IN1 的值除以输入 IN2 的值并通过输出 OUT 查询余数。
 只有输入 EN 的信号状态为“1”时，才会执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。
 如果输入 EN 的信号状态为“0”，则不会执行该操作。在这种情况下，将复位输出 ENO。

放置

“返回除法的余数”运算可以放置在程序段中的任何位置。

实例

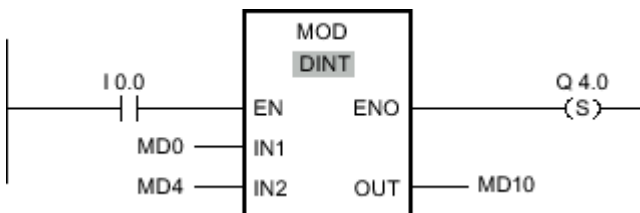


图 7-2

如果输入 I0.0 的信号状态为“0”，则执行“返回除法的余数”运算。用输入 MD0 的值除以输入 MD4 的值。除法的余数放在输出 MD10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 NEG : 创建补码

符号

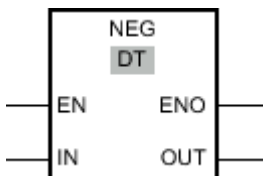


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出

参数	数据类型	存储区	描述
IN	SINT、INT、DINT、REAL	I、Q、M、L、D 或常数	输入值
OUT	SINT、INT、DINT、REAL	I、Q、M、L、D	输入值的补码

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“创建补码”操作更改输入 IN 的值符号，并通过输出 OUT 查询结果。例如，如果输入 IN 为正值，则该值的负等效值将发送到输出 OUT。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 操作结果超出输出 OUT 指定的数据类型的允许范围。
- 输入变量为 REAL 数据类型且无效。

放置

“创建补码”运算可以放置在程序段中的任何位置。

实例

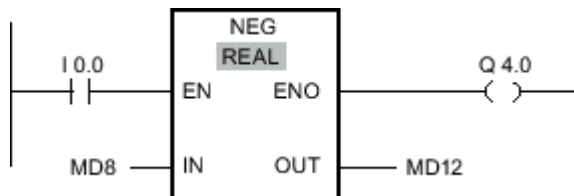


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“创建补码”运算。输入 MD8 的值的符号被更改，且结果放在输出 MD12 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

[插入 LAD 元素 \(页 417\)](#)

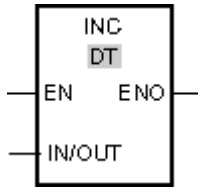
[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

7.8.1.1 INC：递增

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN/OUT	SINT、USINT、INT、UINT、DINT、UDINT	I、Q、M、D、L	要递增的值。

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“递增”运算将输出 IN/OUT 的变量更改为下一个较高的值并查询结果。只有使能输入端 EN 的信号状态为“1”时，才能启动“递增”运算。如果执行过程中未发生溢出错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则不执行该运算。在这种情况下，将复位使能输出 ENO。

放置

“递增”运算可以放置在程序段中的任何位置。

实例

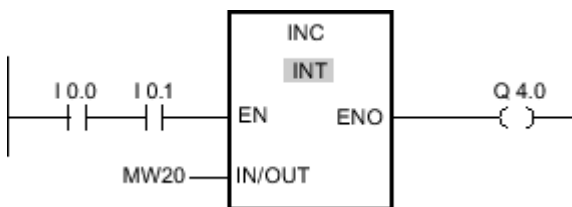


图 7-2

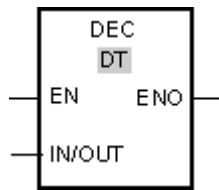
如果输入 I0.0 和 I0.1 的信号状态为“1”，则输出 MW20 处的值将递增“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 DEC : 递减

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN/OUT	SINT、USINT、INT、UINT、DINT、UDINT	I、Q、M、D、L	要递减的值。

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

使用“递减”运算可以将输出 IN/OUT 的变量的值更改为下一个较低的值并查询结果。当使能输入 EN 的信号状态为“1”时，开始执行“递减”运算。如果执行过程中未发生下溢错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则不执行该运算。在这种情况下，将复位使能输出 ENO。

放置

“递减”运算可以放置在程序段中的任何位置。

实例

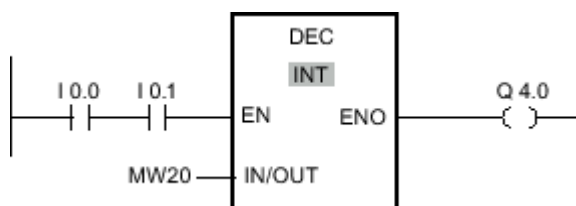


图 7-2

如果输入 I 0.0 和 I 0.1 的信号状态为“1”，则输出 MW20 的值将递减“1”并置位输出 Q 4.0。

参见

插入 LAD 元素 (页 417)

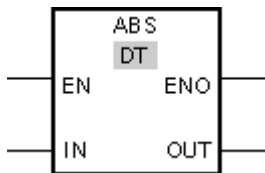
更改 LAD 元素 (页 425)

选择 LAD 元素的数据类型 (页 419)

将操作数插入 LAD 指令 (页 430)

7.8.1.1 ABS : 计算绝对值

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	SINT、INT、DINT、REAL	I、Q、M、L、D 或常数	输入值
OUT	SINT、INT、DINT、REAL	I、Q、M、L、D	输入值的绝对值

描述

可以使用“计算绝对值”运算计算输入 IN 所指定值的绝对值。运算结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 指定的 REAL 变量的值不是有效浮点数。

放置

“计算绝对值”运算可以放置在程序段中的任何位置。

实例

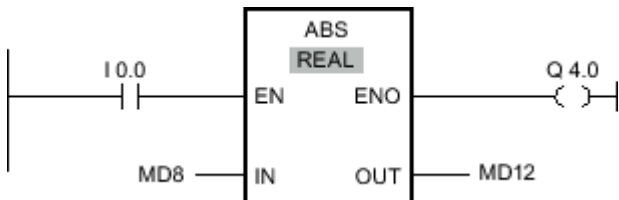


图 7-2

IN	MD8 = - 6.234
----	---------------

OUT	MD12 = 6.234
-----	--------------

如果输入 I0.0 的信号状态为“1”，则执行“计算绝对值”运算。该运算会计算输入 MD8 的值的绝对量，并将结果发送到输出 MD12。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

选择 LAD 元素的数据类型 (页 419)

7.8.1.1 MIN：提取最小值

符号

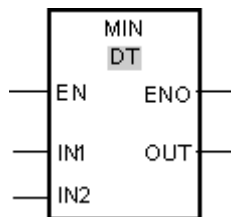


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	第一个输入值
IN2	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	第二个输入值
OUT	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

“提取最小值”运算会比较输入 IN1 的值和输入 IN2 的值，并将较小值写入到输出 OUT。只有为所有参数设置的变量具有相同数据类型时，才会执行该运算。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果在运算执行期间未发生错误，则使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 指定的变量不具有相同的数据类型。
- 输入变量为 REAL 数据类型且无效。

放置

“提取最小值”运算可以放置在程序段中的任何位置。

实例

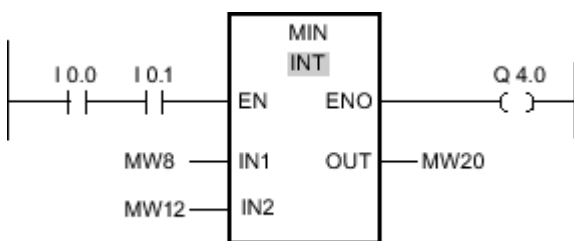


图 7-2

IN1	MW8 = 12,666
IN2	MW12 = 14,444
OUT	MW20 = 12,666

如果输入 I0.0 与 I0.1 的信号状态为“1”，则执行“提取最小值”运算。该运算会比较输入 MW8 的值和输入 MW12 的值，并选择较小值 (MW8)。该值被复制到输出 MW20。如果运算执行过程中未发生错误，则置位输出 Q4.0。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 将操作数插入 LAD 指令 (页 430)*
- 选择 LAD 元素的数据类型 (页 419)*

7.8.1.1 MAX：确定最大值

符号

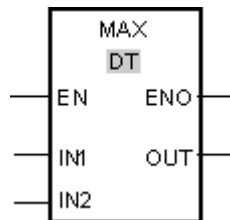


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	第一个输入值
IN2	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	第二个输入值
OUT	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

“提取最大值”运算会比较输入 IN1 和 IN2 的值，并将较大值写入到输出 OUT。只有为所有参数设置的变量具有相同数据类型时，才会执行该运算。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果在运算执行期间未发生错误，则使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出的信号状态为“0”：

- 输入 EN 的信号状态为“0”。
- 指定的变量不具有相同的数据类型。
- 输入变量为 REAL 数据类型且无效。

放置

“提取最大值”运算可以放置在程序段中的任何位置。

实例

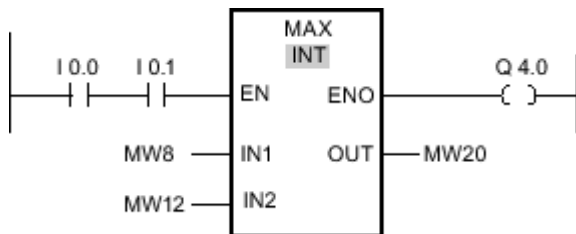


图 7-2

IN1	MW8 = 12,666
IN2	MW12 = 14,444
OUT	MW20 = 14,444

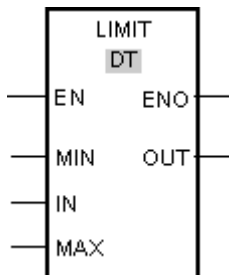
如果输入 I0.0 和 I0.1 的信号状态为“1”，则执行“提取最大值”运算。该运算会比较输入 MW8 和 MW12 的值，并选择较大值 (MW12)。该值被复制到输出 MW20。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 选择 LAD 元素的数据类型 (页 419)
- 将操作数插入 LAD 指令 (页 430)

7.8.1.1 LIMIT : 设置限值

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出

参数	数据类型	存储区	描述
MIN	SINT、USINT、INT、 UINT、DINT、 UDINT、REAL	I、Q、M、L、D 或常数	下限
IN	SINT、USINT、INT、 UINT、DINT、 UDINT、REAL	I、Q、M、L、D 或常数	输入值
MAX	SINT、USINT、INT、 UINT、DINT、 UDINT、REAL	I、Q、M、L、D 或常数	上限
OUT	SINT、USINT、INT、 UINT、DINT、 UDINT、REAL	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“设置限值”运算，利用输入 MIN 和 MAX 的值来限制输入 IN 的值。如果输入 IN 的值满足 $MIN < IN < MAX$ 的条件，则将该值复制到输出 OUT。如果不满足该条件且输入值低于下限，则将输出设置为输入 MIN 的值。如果超出上限，输出 OUT 将设置为输入 MAX 的值。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果在运算执行期间未发生错误，则使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 指定的变量不具有相同的数据类型。
- 某个输入变量具有无效值。
- 输入 MIN 的值大于输入 MAX 的值。

放置

“设置限值”运算可以放置在程序段中的任何位置。

实例

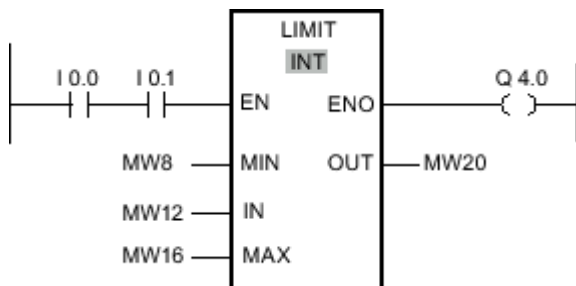


图 7-2

MIN	MW8 = 12000
IN	MW12 = 8000
MAX	MW16 = 16000
OUT	MW20 = 12000

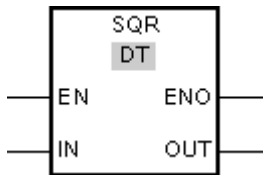
如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则执行“设置限值”运算。将输入 MW12 的值与输入 MW8 和 MW16 的值进行比较。由于输入 MW12 的值小于下限，因此将输入 MW8 的值复制到输出 MW20。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 选择 LAD 元素的数据类型 (页 419)
- 将操作数插入 LAD 指令 (页 430)

7.8.1.1 SQR : 计算平方

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	输入值
OUT	REAL	I、Q、M、L、D	输入值的平方

描述

可以使用“计算平方”运算计算输入 IN 的值的平方，并通过输出 OUT 查询结果。
 只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算平方”运算可以放置在程序段中的任何位置。

实例

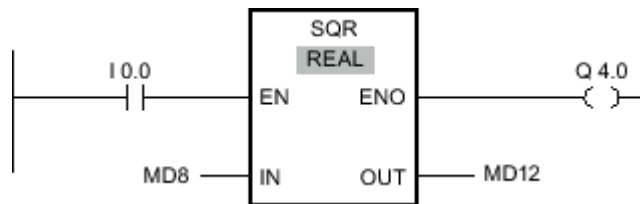


图 7-2

IN	MD8 = 5
OUT	MD12 = 25

如果输入 I 0.0 的信号状态为“1”，则执行“计算平方”运算。该运算计算输入 MD8 的值的平方，并将结果发送到输出 MD12。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

[插入 LAD 元素 \(页 417\)](#)

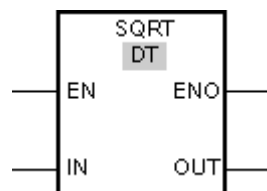
[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

7.8.1.1 SQRT：计算平方根

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	输入值
OUT	REAL	I、Q、M、L、D	输入值的平方根

描述

可以使用“计算平方根”运算计算输入 IN 的值的平方根，并通过输出 OUT 查询结果。如果输入值大于零，则运算结果为正。如果输入值小于零，则输出 OUT 返回一个无效浮点数。如果输入 IN 的值为“-0”，则结果也是“-0”。

只有使能输入的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。
- 输入 IN 的值为负值。

放置

“计算平方根”运算可以放置在程序段中的任何位置。

实例

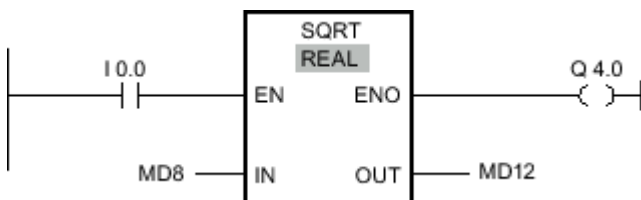


图 7-2

IN	MD8 = 25
OUT	MD12 = 5

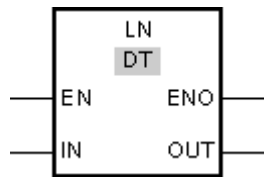
如果输入 I 0.0 的信号状态为“1”，则执行“计算平方根”运算。该运算计算输入 MD8 的值的平方根，并将结果发送到输出 MD12。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 将操作数插入 LAD 指令 (页 430)*
- 选择 LAD 元素的数据类型 (页 419)*

7.8.1.1 LN：计算自然对数

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	输入值：
OUT	REAL	I、Q、M、L、D	运算结果

描述

可以使用“计算自然对数”运算计算输入 IN 的自然对数，以 e ($e=2.718282e+00$) 为底。运算结果放在输出 OUT 中，可供查询。如果输入值大于零，则运算结果为正。如果输入值小于零，则输出 OUT 返回一个无效浮点数。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。
- 输入 IN 的值为负值。

放置

“计算自然对数”运算可以放置在程序段中的任何位置。

实例

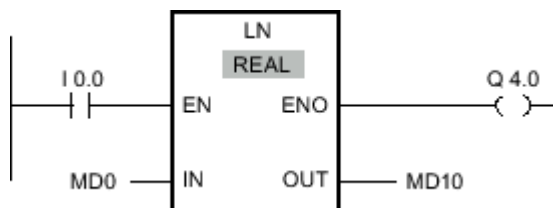


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“计算自然对数”运算。该运算计算输入 MD0 的值的自然对数，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 LAD 元素 (页 417)

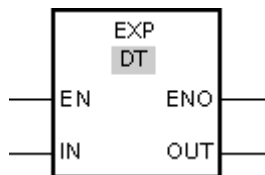
更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

选择 LAD 元素的数据类型 (页 419)

7.8.1.1 EXP : 计算指数值

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	输入值
OUT	REAL	I、Q、M、L、D	输出值

描述

可以使用“计算指数值”运算计算底数为 e ($e = 2.718282e+00$) 且幂为输入 IN 上设置值的指数值。运算结果放在输出 OUT 中，可供查询 ($OUT = e^{IN}$)。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算指数值”运算可以放置在程序段中的任何位置。

实例

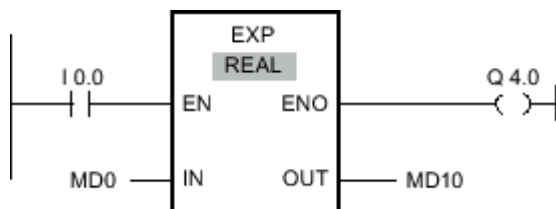


图 7-2

如果输入 I0.0 的信号状态为“1”，则执行“计算指数值”运算。该运算会计算以 e 为底且以输入 MD0 的值为幂的指数值，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q4.0。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

选择 LAD 元素的数据类型 (页 419)

7.8.1.1 SIN : 计算正弦值

符号

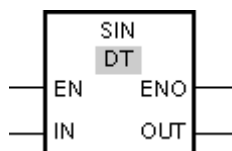


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	以弧度为单位的角度值
OUT	REAL	I、Q、M、L、D	指定角度的正弦

描述

可以使用“计算正弦值”运算计算输入 IN (以弧度为单位指定的角度) 的正弦值。运算结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算正弦值”运算可以放置在程序段中的任何位置。

实例

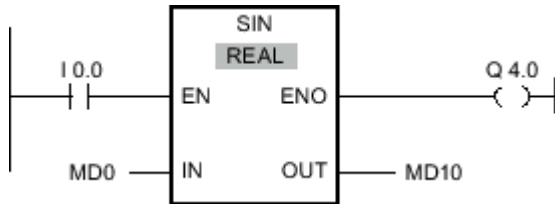


图 7-2

IN	MD0 = +1.570796e+00 ($\pi/2$)
OUT	MD10 = 1

如果输入 I 0.0 的信号状态为“1”，则执行“计算正弦值”运算。此运算会计算在输入 MD0 (指定的角度) 的正弦值，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 将操作数插入 LAD 指令 (页 430)*
- 选择 LAD 元素的数据类型 (页 419)*

7.8.1.1 COS : 计算余弦值

符号

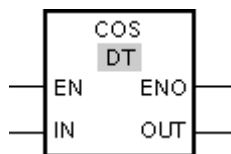


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	以弧度为单位的角度值
OUT	REAL	I、Q、M、L、D	指定角度的余弦

描述

可以使用“计算余弦值”运算计算一个角度的余弦。在输入 IN 指定以弧度为单位的角度值，运算结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算余弦值”运算可以放置在程序段中的任何位置。

实例

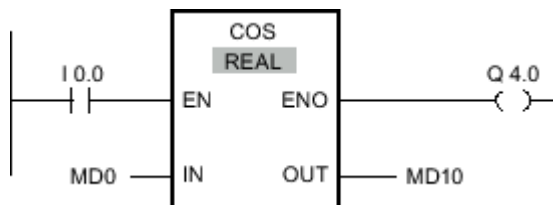


图 7-2

IN	MD0 = +1.570796e+00 ($\pi/2$)
OUT	MD10 = 0

如果输入 I 0.0 的信号状态为“1”，则执行“计算余弦值”运算。此运算会计算在输入 MD0 (指定的角度) 的余弦值，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

7.8.1.1 TAN : 计算正切值

符号

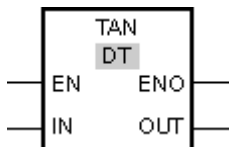


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	以弧度为单位的角度值
OUT	REAL	I、Q、M、L、D	指定角度的正切

描述

可以使用“计算正切值”运算计算一个角度的正切。在输入 IN 指定以弧度为单位的角度值，运算结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算正切值”运算可以放置在程序段中的任何位置。

实例

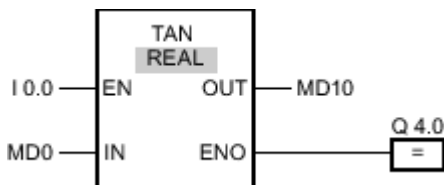


图 7-2

IN	MD0 = +3.141593e++00 (π)
OUT	MD10 = 0

如果输入 I0.0 的信号状态为“1”，则执行“计算正切值”运算。此运算会计算在输入 MD0 (指定的角度) 的正切值，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

选择 LAD 元素的数据类型 (页 419)

7.8.1.1 ASIN : 计算反正弦值

符号

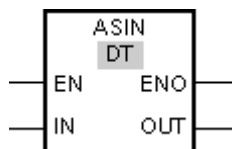


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	正弦值
OUT	REAL	I、Q、M、L、D	以弧度为单位的角度值

描述

可以使用“计算反正弦值”运算根据输入 IN 指定的正弦值计算对应的角度值。只能给输入 IN 指定在范围 -1 到 +1 内的有效浮点数。计算出的角度值以弧度为单位放在输出 OUT 中，该值应该在 $-\pi/2$ 到 $+\pi/2$ 范围内。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。
- 输入 IN 的值超出允许的范围 (-1 到 +1)。

放置

“计算反正弦值”运算可以放置在程序段中的任何位置。

实例

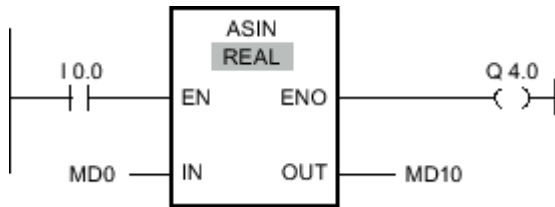


图 7-2

IN	MD0 = 1
OUT	MD10 = +1.570796e+00 ($\pi/2$)

如果输入 I0.0 的信号状态为“1”，则执行“计算反正弦值”运算。该运算会计算与输入 MD0 的正弦值对应的角度值。运算结果放在输出 MD10 中。如果运算执行过程中未发生错误，则置位输出 Q4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 ACOS : 计算反余弦值

符号

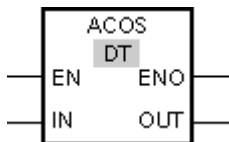


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	余弦值
OUT	REAL	I、Q、M、L、D	以弧度为单位的角度值

描述

可以使用“计算反余弦值”运算计算输入 IN 指定的余弦值对应的角度值。只能给输入 IN 指定在范围 -1 到 +1 内的有效浮点数。计算出的角度值以弧度为单位放在输出 OUT 中，该值应该在 0 到 $+\pi$ 范围内。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。
- 输入 IN 的值超出允许的范围（-1 到 +1）。

放置

“计算反余弦值”运算可以放置在程序段中的任何位置。

实例

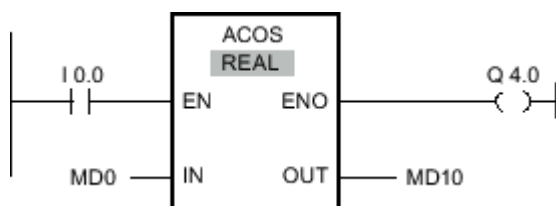


图 7-2

IN	MD0 = 0
OUT	MD10 = +1.570796e+00 ($\pi/2$)

如果输入 I 0.0 的信号状态为“1”，则执行“计算反余弦值”运算。该运算会计算与输入 MD0 的余弦值对应的角度值。运算结果放在输出 MD10 中。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

7.8.1.1 ATAN：计算反正切值

符号

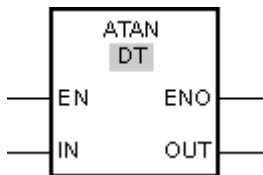


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	正切值
OUT	REAL	I、Q、M、L、D	以弧度为单位的角度值

描述

可以使用“计算反正切值”运算计算输入 IN 指定的正切值对应的角度值。只能给输入 IN 指定有效的浮点数。计算出的角度值以弧度为单位放在输出 OUT 中，该值应该在 $-\pi/2$ 到 $+\pi/2$ 范围内。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算反正切值”运算可以放置在程序段中的任何位置。

实例

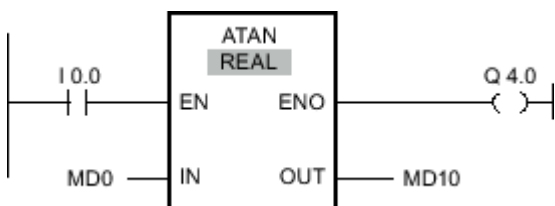


图 7-2

IN	MD0 = 1
----	---------

OUT	MD10 =+0.785398e++00 ($\pi/4$)
-----	----------------------------------

如果输入 I 0.0 的信号状态为“1”，则执行“计算反正切值”运算。该运算会计算与输入 MD0 的正切值对应的角度值。运算结果放在输出 MD10 中。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 LAD 元素 (页 417)

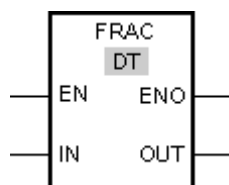
更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

选择 LAD 元素的数据类型 (页 419)

7.8.1.1 FRAC：提取小数

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	要确定其小数位的值。
OUT	REAL	I、Q、M、L、D	输入 IN 的值的的小数位

描述

可以使用“提取小数”运算获取输入 IN 的值的的小数位。查询的结果放在输出 OUT 中，可供查询。例如，如果输入 IN 的值为 123.4567，则输出 OUT 返回值 0.4567。输入 EN 的信号状态为“1”时，将启动该运算。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理该运算期间发生错误，例如，输入端没有有效浮点数。

放置

“提取小数”运算可以放置在程序段中的任何位置。

实例

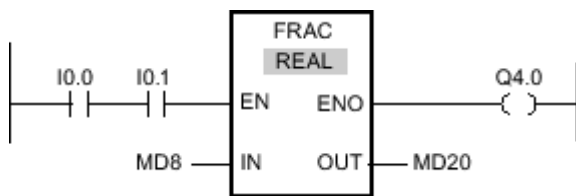


图 7-2

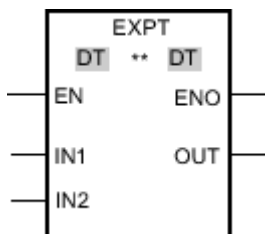
如果输入 I0.0 与 I0.1 的信号状态为“1”，则启动“提取小数”运算。输入 MD8 的值的小数位将会复制到输出 MD20。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 选择 LAD 元素的数据类型 (页 419)
- 将操作数插入 LAD 指令 (页 430)

7.8.1.1 EXPT：取幂

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	REAL	I、Q、M、L、D 或常数	底数值
IN2	REAL、INT、UINT、USINT、SINT、DINT、UDINT	I、Q、M、L、D 或常数	对底数进行幂运算所用的值
OUT	REAL	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

通过“取幂”运算，以输入 IN2 指定的值为指数，对输入 IN1 进行幂运算。该运算的结果放在输出 OUT 中，可供查询 ($OUT = IN1^{IN2}$)。

只能为输入 IN1 分配有效的浮点数。还允许使用整数设置 IN2 输入。

只有使能输入端 EN 的信号状态为“1”时，才执行“取幂”运算。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理该运算期间发生错误，例如，发生溢出。

放置

“取幂”运算可以放置在程序段中的任何位置。

实例

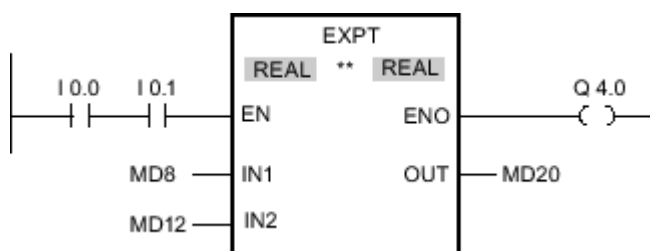


图 7-2

如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则执行“取幂”运算。以输入 MD12 指定的值为指数，对输入 MD8 的值进行幂运算。结果放在输出 MD20 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

7.8.1.1 移动

7.8.1.1 MOVE：移动值

符号

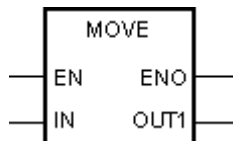


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	所有基本数据类型、DTL、STRUCT、ARRAY	I、Q、M、D、L 或常数	源值
OUT1	所有基本数据类型、DTL、STRUCT、ARRAY	I、Q、M、D、L	目标地址

描述

可以使用“移动值”操作将输入 IN 的操作数的数据传送到输出 OUT1 的操作数。始终沿地址升序方向进行传送。

如果启用 IEC 测试，则输入 IN 和输出 OUT1 的变量必须具有相同的数据类型。如果未启用 IEC 测试，则该操作的输入 IN 和输出 OUT1 的操作数宽度必须相同。

只有当输入 IN 和输出 OUT1 的变量的数组元素具有相同数据类型时，才能复制整个数组。

对于“移动值”操作，可插入附加输出。在这种情况下，输入 IN 的操作数的数据将传送到所有可用的输出。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。在这种情况下，输出 ENO 的信号状态也为“1”。

如果输入 EN 的信号状态为“0”，则使能输出 ENO 将复位为“0”。

“移动块”(MOVE_BLK) 和“移动块不可中断”(UMOVE_BLK) 操作也可用于复制 ARRAY 数据类型的变量。可以使用 S_CONV 操作复制 STRING 数据类型的变量。

放置

“移动值”操作可以放置在程序段中的任何位置。

实例

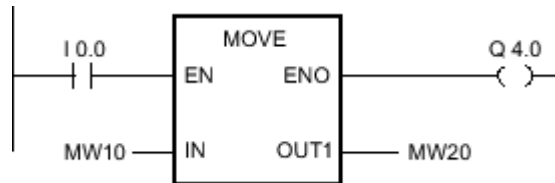


图 7-2

IN	MW10 = 0011 1111 1010 1111
OUT1	MW20 = 0011 1111 1010 1111

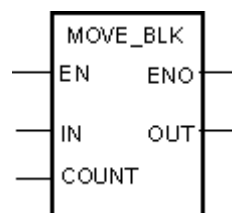
如果输入 I 0.0 的信号状态为“1”，则执行“移动值”操作。该操作将输入操作数 (MW10) 的数据复制到输出操作数 (MW20) 并将输出 Q 4.0 的信号状态置位为“1”。

参见

插入 LAD 元素 (页 417)
 更改 LAD 元素 (页 425)
 插入附加输入和输出 (页 426)
 将操作数插入 LAD 指令 (页 430)

7.8.1.1 MOVE_BLK：移动块

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	ARRAY	D、L	要复制的源区域的第一个元素。
COUNT	UINT	I、Q、M、D、L 或常数	要从源区域复制到目标区域的元素个数。
OUT	ARRAY	D、L	源区域数据将复制到的目标区域的第一个元素。

描述

可以使用“移动块”操作将一个存储区（源区域）的内容复制到另一个存储区（目标区域）。要复制到目标区域的元素个数由参数 COUNT 指定。要复制的元素的宽度由输入 IN 的元素宽度定义。复制操作沿地址升序方向进行。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 复制的数据量超出输出 OUT 存储区所提供的数据量。

放置

“移动块”操作可以放置在程序段中的任何位置。

实例

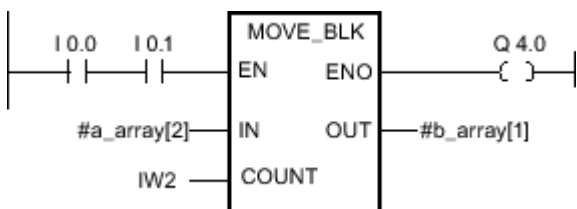


图 7-2

IN	变量“a_array”是 ARRAY 数据类型，由 5 个 INT 数据类型的元素组成。
COUNT	IW2 = 3
OUT	变量“b_array”是 ARRAY 数据类型，由 6 个 INT 数据类型的元素组成。

如果输入 I0.0 与 I0.1 的信号状态为“1”，则执行“移动块”操作。该操作从变量“a_array”选择三个 INT 元素 (a_array[2..4]) 并将其复制到输出变量“b_array”(b_array[1..3])。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q4.0。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 插入附加输入和输出 (页 426)*
- 将操作数插入 LAD 指令 (页 430)*

7.8.1.1 UMOVE_BLK : 移动块不可中断

符号

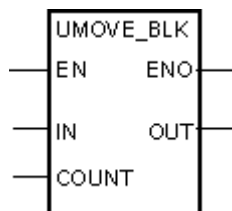


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	ARRAY	D、L	要复制的源区域的第一个元素。
COUNT	UINT	I、Q、M、D、L 或常数	要从源区域复制到目标区域的元素个数。
OUT	ARRAY	D、L	源区域数据将复制到的目标区域的第一个元素。

描述

可以使用“移动块不可中断”操作将一个存储区（源区域）的内容无中断地复制到另一个存储区（目标区域）。要复制到目标区域的元素个数由参数 COUNT 指定。要复制的元素的宽度由输入 IN 的元素宽度定义。

源区域内容沿地址升序方向复制到目标区域。此复制操作不能被其它操作系统任务打断。这就是执行“移动块不可中断”操作期间 CPU 的报警响应时间增加的原因。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 复制的数据量超出输出 OUT 可用的数据量。

放置

“移动块不可中断”操作可以放置在程序段中的任何位置。

实例

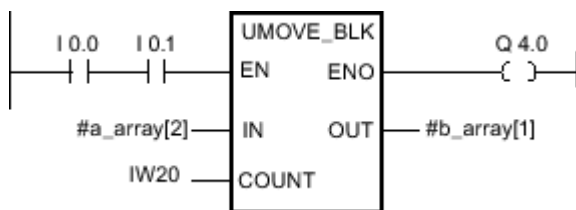


图 7-2

IN	变量“a_array”是 ARRAY 数据类型，由 5 个 INT 数据类型的元素组成。
COUNT	IW20 = 3
OUT	变量“b_array”是 ARRAY 数据类型，由 6 个 INT 数据类型的元素组成。

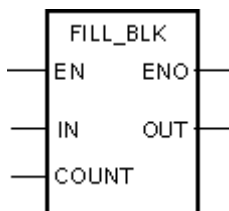
如果输入 I0.0 与 I0.1 的信号状态为“1”，则执行“移动块”操作。该操作从变量“a_array”选择三个 INT 元素 (a_array[2..4]) 并将其复制到输出变量“b_array”(b_array[1..3])。此复制操作不能被其它操作系统任务打断。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 插入附加输入和输出 (页 426)

7.8.1.1 FILL_BLK：填充块

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	ARRAY	D、L 或常数	用于填充目标区域的元素。

参数	数据类型	存储区	描述
COUNT	UINT	I、Q、M、D、L 或常数	重复的复制操作次数
OUT	ARRAY	D、L	目标区域中开始填充的地址。

描述

可以通过“填充块”操作使用输入 IN 的值填充存储区（目标区域）。从输出 OUT 指定的地址开始填充目标区域。重复的复制操作次数由参数 COUNT 指定。执行该操作时，将选择输入 IN 的值并将其按照参数 COUNT 指定的重复次数复制到目标区域。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 复制的数据量超出输出 OUT 存储区所提供的的数据量。

放置

“填充块”操作可以放置在程序段中的任何位置。

实例

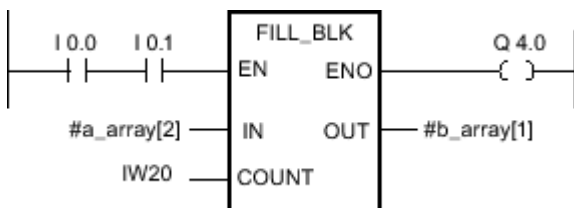


图 7-2

IN	变量“a_array”是 ARRAY 数据类型，由 4 个 WORD 数据类型的元素组成（WORD 类型的 ARRAY[1..4]）。
COUNT	IW20=3
OUT	变量“b_array”是 ARRAY 数据类型，由 5 个 WORD 数据类型的元素组成（WORD 类型的 ARRAY[1..5]）。

如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则执行“填充块”操作。该操作将变量“a_array”的第二个元素（a_array[2]）复制到输出变量“b_array”（b_array[1..3]）三次。如果操作执行过程中未发生错误，则输出 ENO 和 Q 4.0 的信号状态设置为“1”。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

插入附加输入和输出 (页 426)

7.8.1.1 UFILL_BLK：填充块不可中断

符号

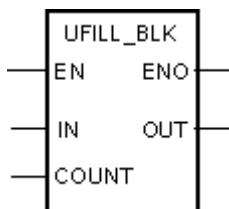


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	ARRAY	D、L 或常数	用于填充目标区域的元素。
COUNT	UINT	I、Q、M、D、L 或常数	重复的复制操作次数
OUT	ARRAY	D、L	目标区域中开始填充的地址。

描述

可以通过“填充块不可中断”操作使用输入 IN 的值不中断地填充存储区 (目标区域)。从输出 OUT 指定的地址开始填充目标区域。重复的复制操作次数由参数 COUNT 指定。执行该操作时，将选择输入 IN 的值并将其按照参数 COUNT 指定的重复次数复制到目标区域。

此复制操作不能被其它操作系统任务打断。这就是执行“填充块不可中断”操作期间 CPU 的报警响应时间增加的原因。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 复制的数据量超出输出 OUT 存储区所提供的数据量。

放置

“填充块不可中断”操作可以放置在程序段中的任何位置。

实例

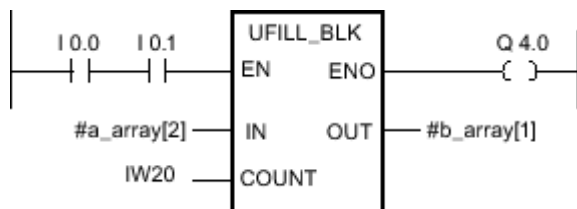


图 7-2

IN	变量“a_array”是 ARRAY 数据类型，由 4 个 WORD 数据类型的元素组成 (WORD 类型的 ARRAY[1..4])。
COUNT	IW20=3
OUT	变量“b_array”是 ARRAY 数据类型，由 5 个 WORD 数据类型的元素组成 (WORD 类型的 ARRAY[1..5])。

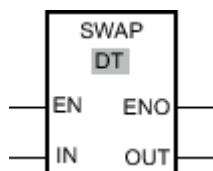
如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则执行“填充块”操作。该操作将变量“a_array”的第二个元素 (a_array[2]) 复制到输出变量“b_array”(b_array[1..3]) 三次。此复制操作不能被其它操作系统任务打断。如果操作执行过程中未发生错误，则输出 ENO 和 Q 4.0 的信号状态设置为“1”。

参见

插入 LAD 元素 (页 417)
 更改 LAD 元素 (页 425)
 将操作数插入 LAD 指令 (页 430)
 插入附加输入和输出 (页 426)

7.8.1.1 SWAP : 交换

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	WORD、DWORD	I、Q、M、L、D 或常数	要交换其字节的操作数。

参数	数据类型	存储区	描述
OUT	WORD、DWORD	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“交换”操作更改输入 IN 变量内的字节顺序，并通过输出 OUT 查询结果。

下图显示了如何使用“交换”操作交换 DWORD 数据类型变量的字节：

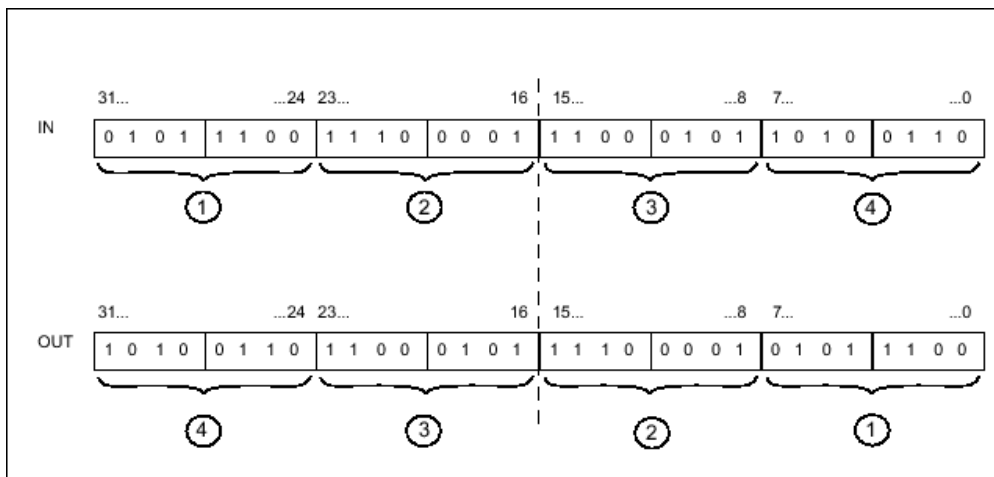


图 7-2

只有使能输入 EN 的信号状态为“1”时，才执行“交换”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”或执行该操作期间出错，将复位使能输出 ENO。

放置

“交换”操作可以放置在程序段中的任何位置。

实例

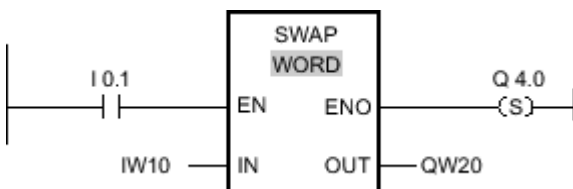


图 7-2

IN	IW10 = 0000 1111 0101 0101
OUT	QW20 = 0101 0101 0000 1111

如果输入 I 0.1 的信号状态为“1”，则执行“交换”操作。字节的排列被更改并存储在 QW20 变量中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

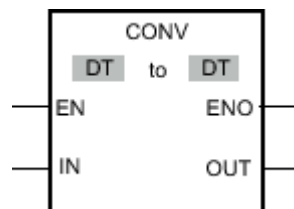
选择 LAD 元素的数据类型 (页 419)

插入附加输入和输出 (页 426)

7.8.1.1 转换

7.8.1.1 CONVERT：转换值

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	BYTE、WORD、DWORD、SINT、INT、DINT、USINT、UINT、UDINT、BCD16、BCD32、REAL	I、Q、M、D、L 或常数	要转换的值。
OUT	BYTE、WORD、DWORD、SINT、INT、DINT、USINT、UINT、UDINT、BCD16、BCD32、REAL	I、Q、M、D、L	转换结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

“转换”操作读取参数 IN 的数据，然后根据指定的数据类型对其进行转换。

仅当使能输入 EN 的信号状态为“1”时，才能启动“转换”操作。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“转换”操作可以放置在程序段中的任何位置。

实例

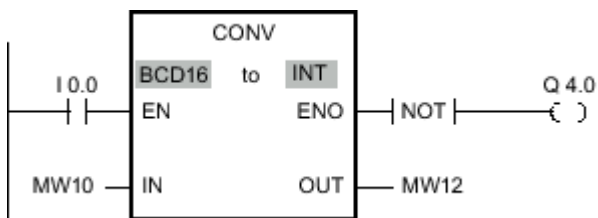


图 7-2

如果输入 I 0.0 = 1，则会以三位 BCD 码数字的形式读取 MW10 的数据并将其转换为整数 (16 位)。结果存储在 MW12 中。如果未执行转换 (ENO = EN = 0)，则输出 Q 4.0 为“1”。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 将操作数插入 LAD 指令 (页 430)*
- 选择 LAD 元素的数据类型 (页 419)*

7.8.1.1 ROUND：舍入为双精度整数

符号

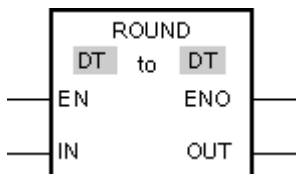


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	要取整的输入值。
OUT	INT、USINT、UINT、SINT、UDINT、DINT、REAL	I、Q、M、L、D	输出值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“取整数字值”运算将输入 IN 的值取整为最接近的整数。该运算将输入 IN 的值解释为浮点数并将其转换为最接近的双精度整数。如果输入值恰好是相邻偶数和奇数的平均数，则选择偶数。运算结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“取整数字值”运算可以放置在程序段中的任何位置。

实例

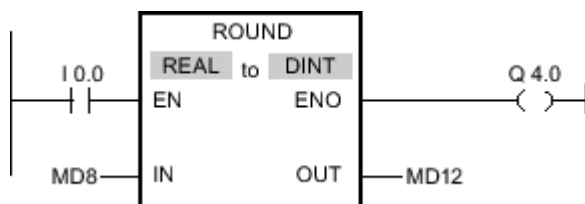


图 7-2

IN	MD8 = 0.50000000
OUT	MD12 = 0

如果输入 I 0.0 的信号状态为“1”，则执行“取整数字值”运算。MD8 输入端的浮点数将取整到最接近的偶数双精度整数并发送到输出 OUT。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 更改 LAD 元素 (页 425)
- 插入 LAD 元素 (页 417)
- 选择 LAD 元素的数据类型 (页 419)
- 将操作数插入 LAD 指令 (页 430)

7.8.1.1 CEIL : 浮点数向上取整

符号

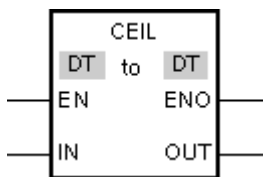


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	输入值
OUT	INT、USINT、UINT、SINT、UDINT、DINT、REAL	I、Q、M、L、D	输出值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“浮点数向上取整”运算将输入 IN 的值向上取整为相邻整数。该运算将输入 IN 的值解释为浮点数并将其向上转换为相邻的整数。运算结果放在输出 OUT 中，可供查询。输出值可以大于或等于输入值。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“浮点数向上取整”运算可以放置在程序段中的任何位置。

实例

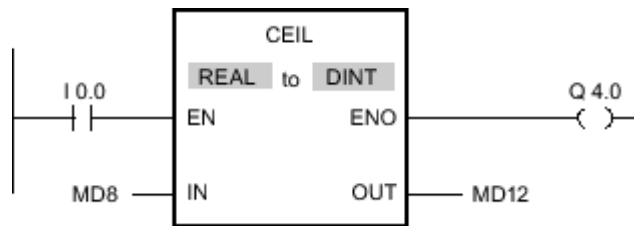


图 7-2

IN	MD8 = 0.50000000
OUT	MD12 = 1

如果输入 I0.0 的信号状态为“1”，则执行“浮点数向上取整”运算。输入 MD8 的浮点数将向上取整为相邻整数并发送到输出 MD12。如果运算执行过程中未发生错误，则置位输出 Q4.0。

参见

更改 LAD 元素 (页 425)

插入 LAD 元素 (页 417)

选择 LAD 元素的数据类型 (页 419)

将操作数插入 LAD 指令 (页 430)

7.8.1.1 FLOOR：浮点数向下取整

符号

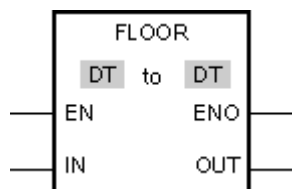


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	输入值
OUT	INT、USINT、UINT、SINT、UDINT、DINT、REAL	I、Q、M、L、D	输出值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“浮点数向下取整”运算将输入 IN 的值向下取整为相邻整数。该运算将输入 IN 的值解释为浮点数并将其向下转换为相邻的整数。运算结果放在输出 OUT 中，可供查询。输出值可以小于或等于输入值。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“浮点数向下取整”运算可以放置在程序段中的任何位置。

实例

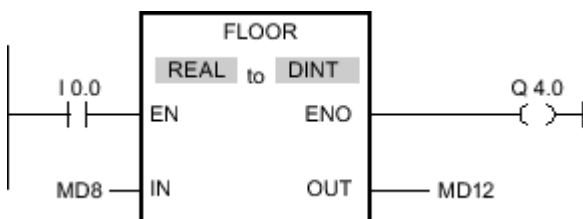


图 7-2

IN	MD8 = 0.50000000
OUT	MD12 = 0

如果输入 I 0.0 的信号状态为“1”，则执行“浮点数向下取整”运算。输入 MD8 的浮点数将向下取整为相邻整数并发送到输出 MD12。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- [插入 LAD 元素 \(页 417\)](#)
- [更改 LAD 元素 \(页 425\)](#)
- [将操作数插入 LAD 指令 \(页 430\)](#)
- [选择 LAD 元素的数据类型 \(页 419\)](#)

7.8.1.1 TRUNC：截取整数

符号

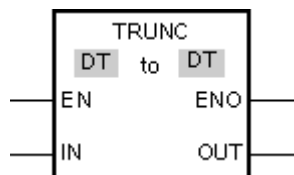


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	输入值
OUT	INT、USINT、UINT、SINT、UDINT、DINT、REAL	I、Q、M、L、D	输入值的整数部分

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“截取整数”运算得出整数而不对输入 IN 的值进行舍入。输入 IN 的值被视为浮点数。该运算仅选择浮点数的整数部分，并将其发送到输出 OUT 中，不带小数位。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“截取整数”运算可以放置在程序段中的任何位置。

实例

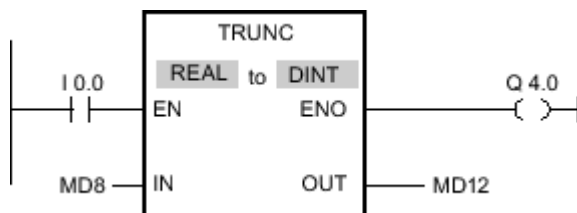


图 7-2

IN	MD8 = 0.50000000
OUT	MD12 = 0

如果输入 I 0.0 的信号状态为“1”，则执行“截取整数”运算。输入 MD8 浮点数的整数部分转换为整数并发送到输出 MD12。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 SCALE_X：缩放

符号

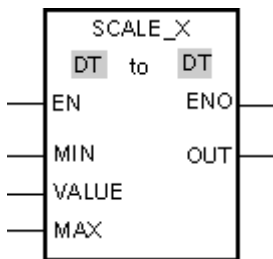


图 7-2

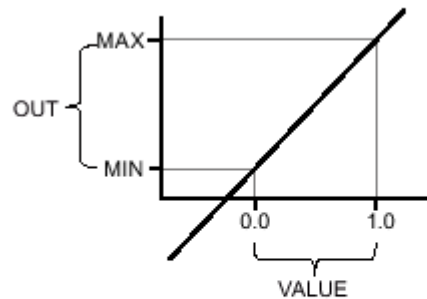
参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
VALUE	REAL	I、Q、M、D、L 或常数	要标定的值。
MIN	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	取值范围的下限
MAX	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	取值范围的上限
OUT	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L	标定的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“标定”运算通过将输入 VALUE 的值映射到指定的取值范围对该值进行标定。执行“标定”运算时，会将输入 VALUE 的浮点数值标定到由参数 MIN 和 MAX 定义的取值范围。标定结果为整数，并存储在输出 OUT 中。

下图所示为如何标定值的实例：



只有使能输入端 EN 的信号状态为“1”时，才执行“标定”运算。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 MIN 的值大于或等于输入 MAX 的值。
- 根据 IEEE-754 标准，指定的 REAL 变量的值超出标准化数的范围。
- 发生溢出。
- 输入 VALUE 的值为 NaN (无效算术运算的结果)。

放置

“标定”运算可以放置在程序段中的任何位置。

实例

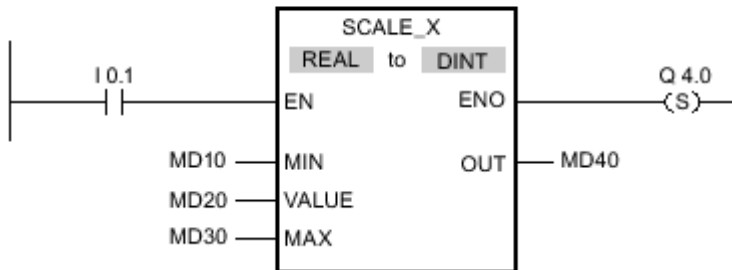


图 7-2

VALUE	MD20 = 0.5
MIN	MD10 = 10
MAX	MD30 = 30
OUT	MD40 = 20

如果输入 I 0.1 的信号状态为“1”，则执行“标定”运算。输入 MD20 的值标定到输入 MD10 和 MD30 的值所定义的取值范围。结果存储在输出 MD40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 NORM_X：标准化

符号

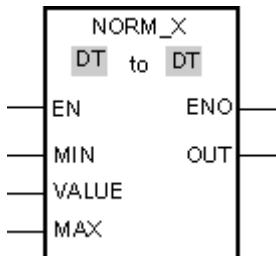


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
VALUE	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	要标准化的值。
MIN	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	取值范围的下限
MAX	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	取值范围的上限
OUT	REAL	I、Q、M、D、L	标准化结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“标准化”运算通过将输入 VALUE 的变量值映射到线性标尺对其进行标准化。可以使用参数 MIN 和 MAX 定义 (应用于该标尺的) 取值范围的限值。根据标准化值在该取值范围内的位置, 计算结果并以浮点数形式存储在输出 OUT 中。如果要标准化的值等于输入 MIN 的值, 输出 OUT 将返回值“0.0”。如果要标准化的值等于输入 MAX 的值, 输出 OUT 将返回值“1.0”。

下图所示为如何标准化值的实例：

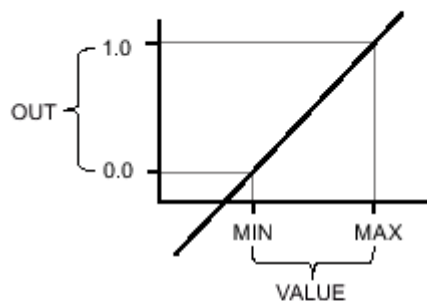


图 7-2

只有使能输入 EN 的信号状态为“1”时，才执行“标准化”运算。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 MIN 的值大于或等于输入 MAX 的值。
- 根据 IEEE-754 标准，指定的 REAL 变量的值超出标准化数的范围。
- 发生溢出。
- 输入 VALUE 的值为 NaN (无效算术运算的结果)。

放置

“标准化”运算可以放置在程序段中的任何位置。

实例

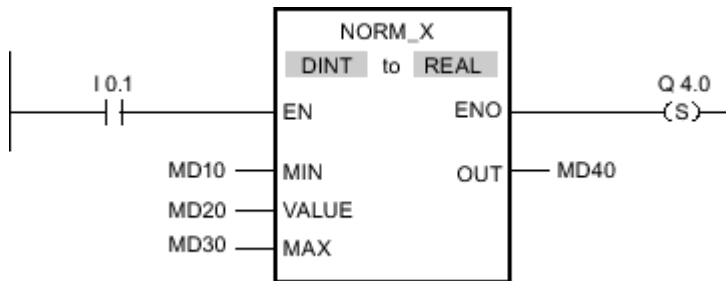


图 7-2

VALUE	MD20 = 20
MIN	MD10 = 10
MAX	MD30 = 30
OUT	MD40 = 0.5

如果输入 I 0.1 的信号状态为“1”，则执行“标准化”运算。输入 MD20 的值分配到输入 MD10 和 MD30 的值所定义的取值范围。根据定义的取值范围标准化输入 MD20 的变量值。结果以浮点数形式存储在输出 MD40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)*
- 更改 LAD 元素 (页 425)*
- 将操作数插入 LAD 指令 (页 430)*
- 选择 LAD 元素的数据类型 (页 419)*

7.8.1.1 程序控制

7.8.1.1 ---

符号

<跳转标签>

--- (JMP)

描述

可以使用“为 1 时块中跳转 (有条件)”操作中断程序的线性执行并在另一个程序段中继续执行。目标程序段必须标有跳转标签。需要指定该跳转标签的名称以便执行该操作。指定的跳转标签位于该操作的上方。

指定的跳转标签必须在执行该操作的块中。指定的名称在块中只能出现一次。

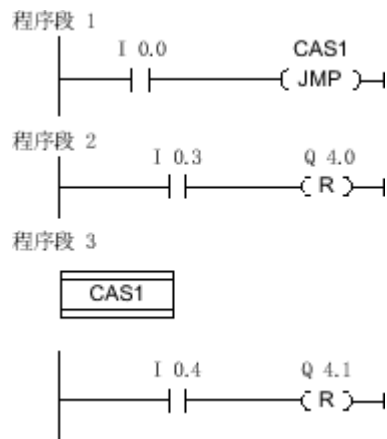
如果该操作的输入端的逻辑运算结果 (RLO) 为“1”，则跳转到跳转标签所标识的程序段。跳转方向可以朝向更大或更小的程序段编号。

如果该操作的输入不为真 (RLO = 0)，则程序继续在下一个程序段中执行。

放置

“为 1 时块中跳转 (有条件)”操作只能放置在程序段的右侧。

实例



如果输入 I 0.0 的信号状态为“1”，则执行“为 1 时块中跳转 (有条件)”操作。程序的线性执行被中断并在跳转标签 CAS1 标识的“程序段 3”中继续执行。如果输入 I 0.4 的信号状态为“1”，则复位输出 Q 4.1。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

7.8.1.1 ---

符号

<跳转标签>

---(JMPN)

描述

可以使用“为 0 时块中跳转 (有条件)”操作，在该操作的输入端的逻辑运算结果为“0”时中断程序的线性执行并在另一个程序段中继续执行。目标程序段必须标有跳转标签。需要指定该跳转标签的名称以便执行该操作。指定的跳转标签位于该操作的上方。

指定的跳转标签必须在执行该操作的块中。指定的名称在块中只能出现一次。

如果该操作的输入端的逻辑运算结果 (RLO) 为“0”，则跳转到跳转标签所标识的程序段。跳转方向可以朝向更大或更小的程序段编号。

如果该操作的输入端的逻辑运算结果为“1”，则程序在下一个程序段中继续执行。

放置

“为 0 时块中跳转 (有条件)”操作需要一个前导逻辑运算并且只能放置在程序段的右侧。

实例

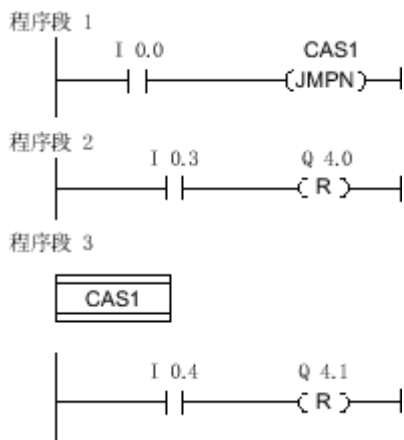


图 7-2

如果输入 I 0.0 的信号状态为“0”，则执行“为 0 时块中跳转 (有条件)”操作。程序的线性执行被中断并在跳转标签 CAS1 标识的“程序段 3”中继续执行。如果输入 I 0.4 的信号状态为“1”，则复位输出 Q 4.1。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

7.8.1.1 LABEL : 跳转标签

符号

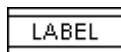


图 7-2

描述

可以使用“跳转标签”指定跳转的目标程序段，程序应在目标程序段中继续执行。跳转标签的名称可以包含字母、数字或下划线。

跳转标签以及指定了该跳转标签的操作必须位于同一个块中。跳转标签的名称在块中只能分配一次。

每个跳转标签可以跳转到多个位置。

实例

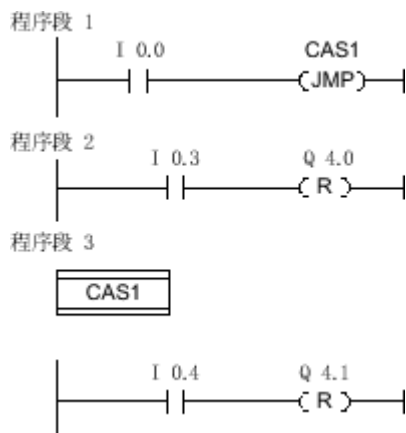


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“为 1 时块中跳转 (有条件)”操作。程序的线性执行被中断并在跳转标签 CAS1 标识的“程序段 3”中继续执行。如果输入 I 0.4 的信号状态为“1”，则复位输出 Q 4.1。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

7.8.1.1 --

符号

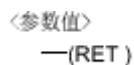


图 7-2

参数值	数据类型	存储区	描述
TRUE	-	-	调用功能的状态设置为“1”。
FALSE	-	-	调用功能的状态设置为“0”。
<操作数>	BOOL	I、Q、M、L、D (页 293)	调用功能的状态设置为指定操作数的信号状态。

描述

可以使用“返回”操作停止块的执行。只有左侧接头的信号状态为“1”时，才会执行该操作。如果满足该条件，将在当前被调用块中终止程序执行并在调用块（例如，在调用 OB 中）中的调用功能后继续执行。调用功能的状态由“返回”操作的参数决定。该参数可以为以下值：

- TRUE：调用功能的输出 ENO 置位为“1”。
- FALSE：调用功能的输出 ENO 复位为“0”。
- <操作数>：调用功能的输出 ENO 由指定操作数的信号状态决定。

如果组织块被“返回”操作终止，则 CPU 继续执行系统程序。

如果“返回”操作输入端的信号状态为“0”，则不执行该操作。在这种情况下，程序继续在被调用块的下一个程序段中执行。

放置

“返回”操作可以放置在程序段的左端或者右端。

实例

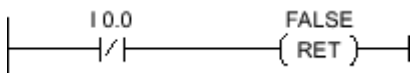


图 7-2

如果输入 I 0.0 的信号状态为“0”，则执行“返回”操作。被调用块中的程序执行被终止，并继续在调用块中执行。调用功能的输出 ENO 的信号状态复位为“0”。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

7.8.1.1 逻辑运算

7.8.1.1 AND : AND 逻辑运算

符号

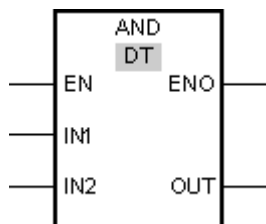


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	逻辑运算的第一个值
IN2	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	逻辑运算的第二个值
OUT	BYTE、WORD、DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“AND 逻辑运算”将输入 IN1 的值与输入 IN2 的值通过 AND 逻辑逐位运算，并通过输出 OUT 查询结果。

执行该运算时，输入 IN1 的值的位 0 与输入 IN2 的值的位 0 通过 AND 逻辑进行运算。结果存储在输出 OUT 的位 0 中。对指定值的所有其它位都执行相同的逻辑运算。

仅当该逻辑运算中的两个位的信号状态均为“1”时，结果位的信号状态才为“1”。如果该逻辑运算的两个位中有一个位的信号状态为“0”，则对应的结果位将复位。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。在这种情况下，输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“AND 逻辑运算”可以放置在程序段中的任何位置。

实例

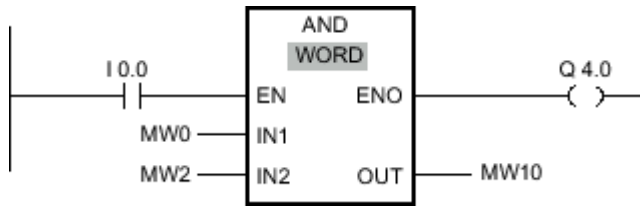


图 7-2

IN1	MW0 = 01010101 01010101
IN2	MW2 = 00000000 00001111
OUT	MW10= 00000000 00000101

如果输入 I 0.0 的信号状态为“1”，则执行“AND 逻辑运算”。输入 MW0 的值与输入 MW2 的值通过 AND 逻辑进行运算。逐位运算得出结果并发送到输出 MW10 中。输出 ENO 和输出 Q 4.0 的信号状态置位为“1”。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 OR : OR 逻辑运算

符号

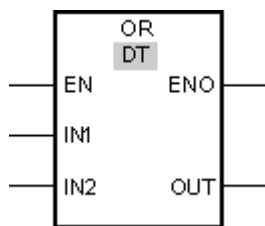


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	逻辑运算的第一个值

参数	数据类型	存储区	描述
IN2	BYTE、WORD、 DWORD	I、Q、M、L、D 或常数	逻辑运算的第二个值
OUT	BYTE、WORD、 DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“OR 逻辑运算”将输入 IN1 的值与输入 IN2 的值通过 OR 逻辑逐位运算，并通过输出 OUT 查询结果。

执行该运算时，输入 IN1 的值的位 0 与输入 IN2 的值的位 0 通过 OR 逻辑进行运算。结果存储在输出 OUT 的位 0 中。对指定变量的所有位都执行相同的逻辑运算。

只要该逻辑运算中的两个位中至少有一个位的信号状态为“1”，结果位的信号状态就为“1”。如果该逻辑运算的两个位的信号状态均为“0”，则对应的结果位将复位。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。在这种情况下，输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“OR 逻辑运算”可以放置在程序段中的任何位置。

实例

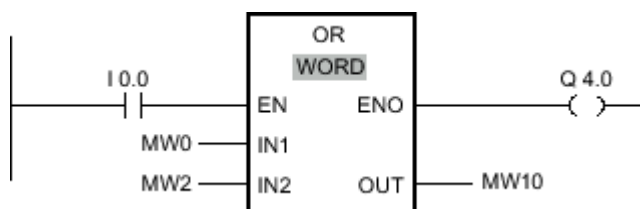


图 7-2

IN1	MW0 = 01010101 01010101
IN2	MW2 = 00000000 00001111
OUT	MW10 = 01010101 01011111

如果输入 I 0.0 的信号状态为“1”，则执行“OR 逻辑运算”。输入 MW0 的值与输入 MW2 的值通过 OR 逻辑进行运算。逐位运算得出结果并发送到输出 MW8 中。输出 ENO 和输出 Q 4.0 的信号状态置位为“1”。

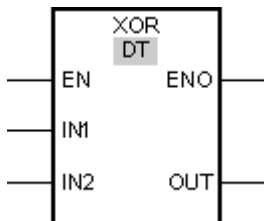
参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)
 将操作数插入 LAD 指令 (页 430)
 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 XOR：异或逻辑运算

符号



参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	BYTE、WORD、 DWORD	I、Q、M、L、D 或常数	逻辑运算的第一个值
IN2	BYTE、WORD、 DWORD	I、Q、M、L、D 或常数	逻辑运算的第二个值
OUT	BYTE、WORD、 DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“异或逻辑运算”将输入 IN1 的值与输入 IN2 的值通过异或逻辑逐位运算，并通过输出 OUT 查询结果。

执行该运算时，输入 IN1 的值的位 0 与输入 IN2 的值的位 0 通过异或逻辑进行运算。结果存储在输出 OUT 的位 0 中。对指定值的所有其它位都执行相同的逻辑运算。

当该逻辑运算中的两个位中有一个位的信号状态为“1”时，结果位的信号状态为“1”。如果该逻辑运算的两个位的信号状态均为“1”或“0”，则对应的结果位将复位。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。在这种情况下，输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“异或逻辑运算”可以放置在程序段中的任何位置。

实例

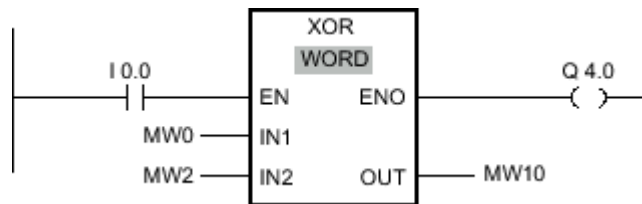


图 7-2

IN1	MW0 = 01010101 01010101
IN2	MW2 = 00000000 00001111
OUT	MW10 = 01010101 01011010

如果输入 I 0.0 的信号状态为“1”，则执行“异或逻辑运算”。输入 MW0 的值与输入 MW2 的值通过异或逻辑进行运算。逐位运算得出结果并发送到输出 MW8 中。输出 ENO 和输出 Q 4.0 的信号状态置位为“1”。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

7.8.1.1 INV：创建反码

符号

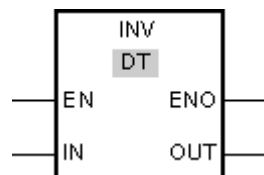


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出

参数	数据类型	存储区	描述
IN	BYTE、WORD、 DWORD、USINT、 UINT、UDINT、 SINT、INT、DINT	I、Q、M、D、L 或常数	输入值
OUT	BYTE、WORD、 DWORD、USINT、 UINT、UDINT、 SINT、INT、DINT	I、Q、M、D、L	输入 IN 的值的反码

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“创建反码”运算对输入 IN 的各位的信号状态取反。执行该运算时，输入 IN 的值与一个十六进制掩码（表示 16 位数的 W#16#FFFF 或表示 32 位数的 DW#16#FFFF FFFF）通过异或逻辑进行运算。这会将各个位的信号状态取反并且结果存储在输出 OUT 中。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。在这种情况下，输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“创建反码”运算可以放置在程序段中的任何位置。

实例

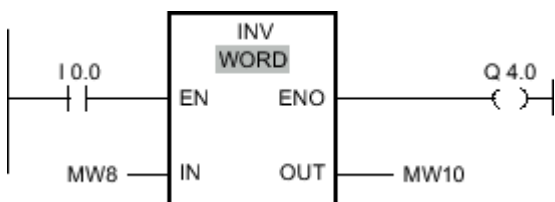


图 7-2

IN	MW8 = W#16#000F
OUT	MW10 = W#16#FFF0

如果输入 I 0.0 的信号状态为“1”，则执行“创建反码”运算。该运算对输入 MW8 的各个位的信号状态取反并将结果写入输出 MW10。输出 ENO 和输出 Q 4.0 的信号状态置位为“1”。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

选择 LAD 元素的数据类型 (页 419)

7.8.1.1 DECO : 解码

符号

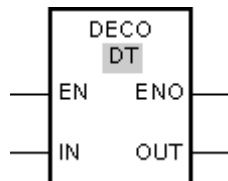


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L。	使能输出
IN	UINT	I、Q、M、D、L 或常数	输入值
OUT	BYTE、WORD、 DWORD	I、Q、M、D、L	输出值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“解码”运算在输出值中置位一个位，该位通过输入值指定。

“解码”运算读取输入 IN 的值，然后在输出值中置位一个位，该位位置与读取的值一致。输出值中的其它位以零填充。当输入 IN 的值大于 31 时，将执行以 32 为模的模运算。

只有使能输入 EN 的信号状态为“1”时，才能启动“解码”运算。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“解码”运算可以放置在程序段中的任何位置。

实例

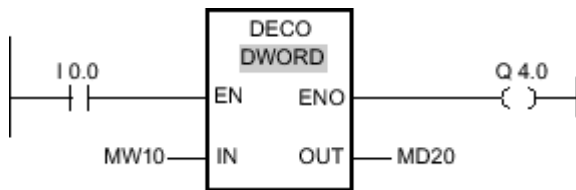


图 7-2

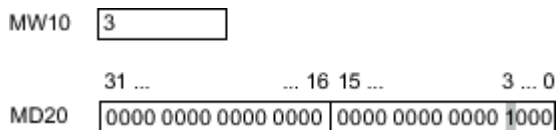


图 7-2

如果输入 I0.0 的信号状态为“1”，则执行“解码”运算。该运算通过输入 MW10 的值读取位号“3”，然后将输出 MD20 的值的第三位位置。

如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 ENCO : 编码

符号

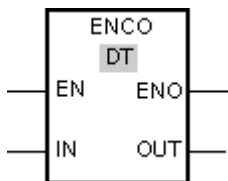


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L。	使能输出
IN	BYTE、WORD、DWORD	I、Q、M、D、L 或常数	输入值
OUT	INT	I、Q、M、D、L	输出值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“编码”运算读取输入值中最低有效置位位的位号并将其发送到输出 OUT。

“编码”运算选择输入 IN 值的最低有效位并将该位号写入输出 OUT 的变量中。

只有使能输入 EN 的信号状态为“1”时，才能启动“编码”运算。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“编码”运算可以放置在程序段中的任何位置。

实例

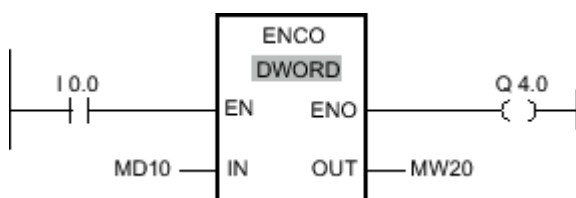


图 7-2

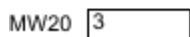
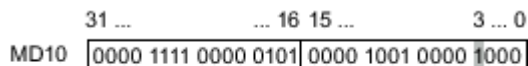


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“编码”运算。该运算选择输入 MD10 的最低有效置位位并将位位置 3 写入输出 MW20 的变量中。

如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

[插入 LAD 元素 \(页 417\)](#)

[更改 LAD 元素 \(页 425\)](#)

[将操作数插入 LAD 指令 \(页 430\)](#)

[选择 LAD 元素的数据类型 \(页 419\)](#)

7.8.1.1 SEL : 选择

符号

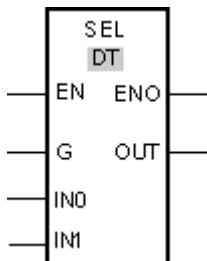


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
G	BOOL	I、Q、M、D、L	选择开关
IN0	BYTE、WORD、 DWORD、SINT、 USINT、INT、UINT、 DINT、UDINT、 REAL、TIME、CHAR	I、Q、M、D、L 或常数	第一个输入值
IN1	BYTE、WORD、 DWORD、SINT、 USINT、INT、UINT、 DINT、UDINT、 REAL、TIME、CHAR	I、Q、M、D、L 或常数	第二个输入值
OUT	BYTE、WORD、 DWORD、SINT、 USINT、INT、UINT、 DINT、UDINT、 REAL、TIME、CHAR	I、Q、M、D、L	所选输入的值

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

“选择”操作依靠一个开关（参数 G）选择输入 IN0 或 IN1 中的一个并将其数据复制到输出 OUT。如果参数 G 的信号状态为“0”，则复制输入 IN0 的值。如果参数 G 的信号状态为“1”，则将输入 IN1 的值复制到输出 OUT。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”或执行该操作期间出错，将复位使能输出 ENO。

放置

“选择”运算可以放置在程序段中的任何位置。

实例

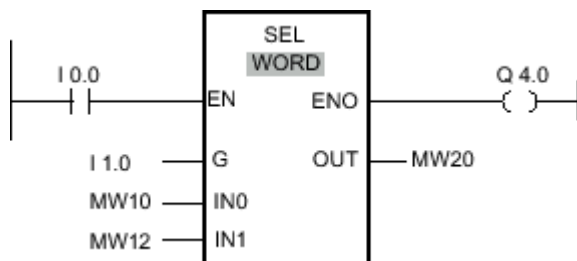


图 7-2

G	I 1.0 = 1
IN0	MW10 = W#16#0000
IN1	MW12 = W#16#FFFF
OUT	MW20 = W#16#FFFF

如果输入 I 0.0 的信号状态为“1”，则执行“选择”操作。根据输入 I 1.0 的信号状态，将选择输入 MW12 的值并复制到输出 MW20。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

选择 LAD 元素的数据类型 (页 419)

7.8.1.1 MUX：多路复用

符号

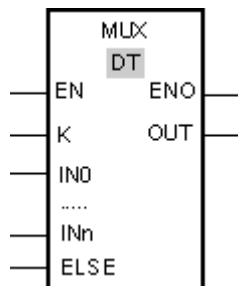


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
K	UINT	I、Q、M、D、L	指定要复制哪个输入的数据。
IN0	BYTE、WORD、 DWORD、INT、 DINT、UINT、 UDINT、SINT、 USINT、REAL、 CHAR、TIME	I、Q、M、D、L 或常数	第一个可用的输入
INn	BYTE、WORD、 DWORD、INT、 DINT、UINT、 UDINT、SINT、 USINT、REAL、 CHAR、TIME	I、Q、M、D、L 或常数	可用的输入
ELSE	BYTE、WORD、 DWORD、INT、 DINT、UINT、 UDINT、SINT、 USINT、REAL、 CHAR、TIME	I、Q、M、D、L 或常数	指定 $K > n$ 时要复制的值。
OUT	BYTE、WORD、 DWORD、INT、 DINT、UINT、 UDINT、SINT、 USINT、REAL、 CHAR、TIME	I、Q、M、D、L	要将值复制到的输出。

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“多路复用”操作将所选输入的数据复制到输出 OUT。功能框 MUX 的可选输入数可以扩展。输入会在该功能框中自动编号。从 IN0 起开始编号，每个新输入的编号连续递增。可以使用参数 K 确定应该将哪个输入的数据复制到输出 OUT。如果参数 K 的值大于可用输入数，则将参数 ELSE 的数据复制到输出 OUT，并将使能输出 ENO 设置为信号状态“0”。

只有所有输入的变量和输出 OUT 的变量具有相同数据类型时，才能执行“多路复用”操作。参数 K 例外，因为只能为其指定整数。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，将复位使能输出 ENO：

- 使能输入 EN 的信号状态为“0”。

- 参数 K 的值大于可用输入数。
- 操作处理期间出错。

放置

“多路复用”运算可以放置在程序段中的任何位置。

实例

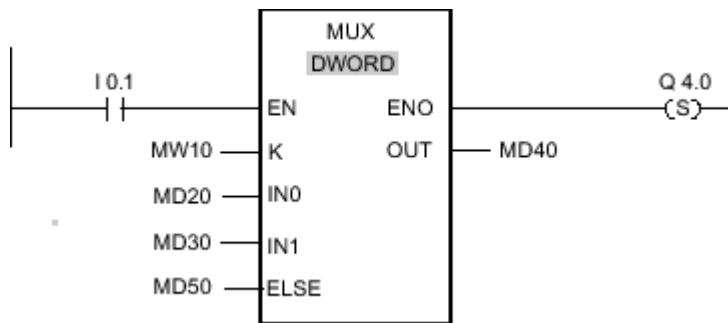


图 7-2

K	MW10 = 1
IN0	MD20 = DW#16#00000000
IN1	MD30 = DW#16#FFFFFFFF
ELSE	MD50 = DW#16#FFFF0000
OUT	MD40 = DW#16#FFFFFFFF

如果输入 I 0.1 的信号状态为“1”，则执行“多路复用”操作。根据输入 MW10 的值，将复制输入 MD30 的值并将其分配给输出 MD40 的变量。如果操作执行过程中未发生错误，则置位输出 ENO 和 Q 4.0。

参见

插入 LAD 元素 (页 417)

更改 LAD 元素 (页 425)

将操作数插入 LAD 指令 (页 430)

选择 LAD 元素的数据类型 (页 419)

7.8.1.1 移位和循环

7.8.1.1 SHR：右移

符号

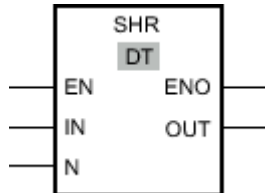


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	BYTE、WORD、 DWORD、USINT、 UINT、UDINT、 SINT、INT、DINT	I、Q、M、L、D 或常数	要移位的值。
N	UINT	I、Q、M、L、D 或常数	将移位的位数。
OUT	BYTE、WORD、 DWORD、USINT、 UINT、UDINT、 SINT、INT、DINT	I、Q、M、L、D	移位操作的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“右移”操作将输入 IN 的变量数据逐位右移，并通过输出 OUT 查询结果。使用参数 N 设置指定将移位的位数。

参数 N 的值为“0”时，输入 IN 的值将被复制到输出 OUT 的变量中。

当参数 N 的值大于位数时，输入 IN 的变量值将按其可用位数向右移位。

无符号值移位时，用零填充变量左侧区域中空出的位。如果指定值有符号，则用符号位的信号状态填充空出的位。

下图显示了一个整型数据类型的变量数据是如何右移四位的：

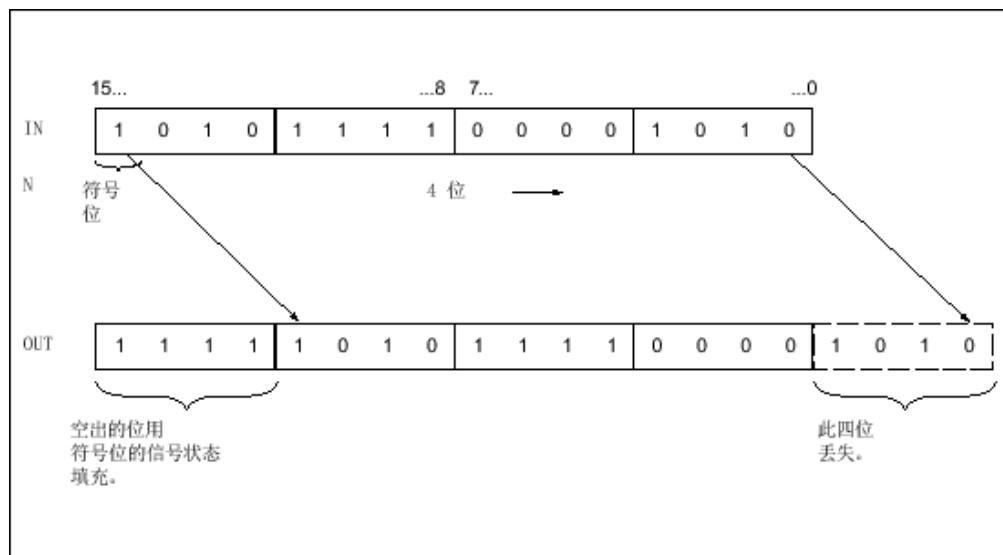


图 7-2

只有使能输入 EN 的信号状态为“1”时，才执行“右移”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

右移操作可以放置在程序段中的任何位置。

实例

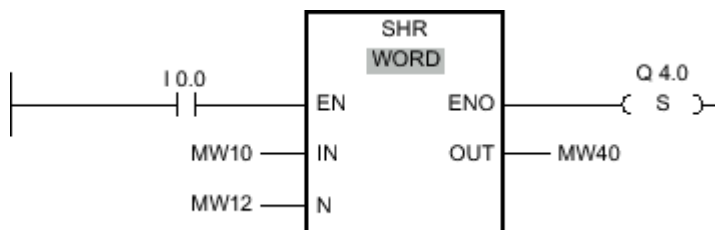


图 7-2

IN	MW10 = 0011 1111 1010 1111
N	MW12 = 3
OUT	MW40 = 0000 0111 1111 0101

如果输入 I 0.0 的信号状态为“1”，则执行“右移”操作。变量 MW10 中的数据右移三位。结果放在输出 MW40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 SHL : 左移

符号

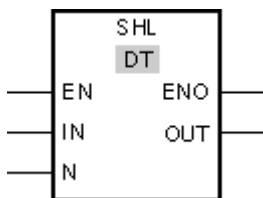


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT	I、Q、M、L、D 或常数	要移位的值。
N	UINT	I、Q、M、L、D 或常数	将移位的位数。
OUT	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT	I、Q、M、L、D	移位操作的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“左移”操作将输入 IN 中的变量数据逐位左移，并通过输出 OUT 查询结果。使用参数 N 设置指定将移位的位数。

参数 N 的值为“0”时，输入 IN 的值将被复制到输出 OUT 的变量中。

当参数 N 的值大于位数时，输入 IN 的变量值将按其可用位数向左移位。

用零填充变量右侧部分因移位空出的位。

下图显示了一个 WORD 数据类型的变量数据是如何左移六位的：

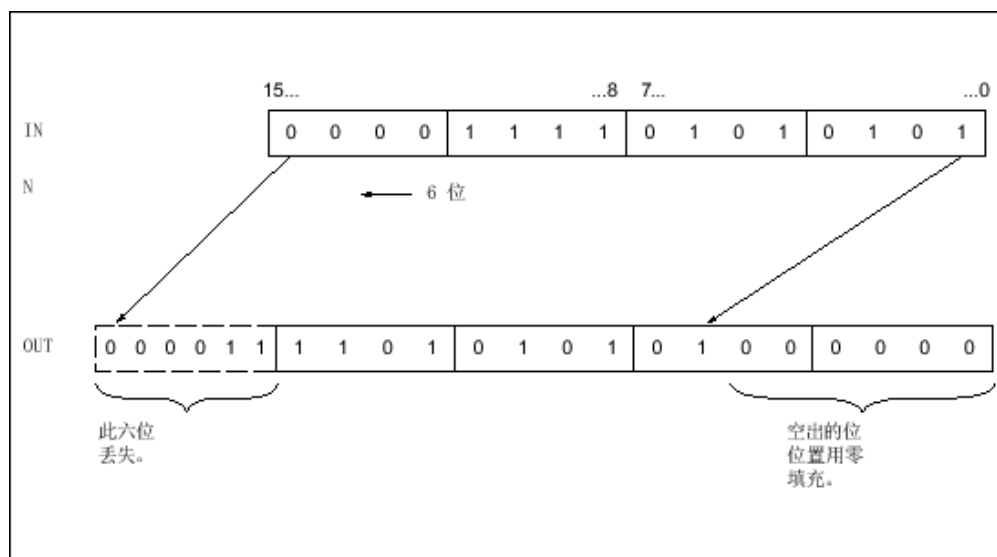


图 7-2

只有使能输入 EN 的信号状态为“1”时，才执行“左移”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“左移”操作可以放置在程序段中的任何位置。

实例

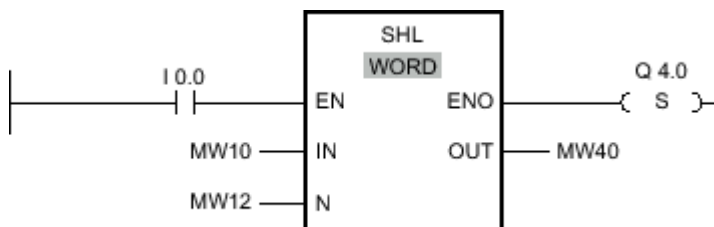


图 7-2

IN	MW10 = 0011 1111 1010 1111
N	MW12 = 4
OUT	MW40 = 1111 1010 1111 0000

如果输入 I 0.0 的信号状态为“1”，则执行“左移”操作。变量 MW10 的数据左移四位。结果放在输出 MW40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 ROR : 循环右移

符号

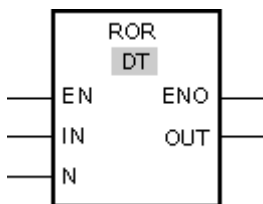


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	要循环移位的值。
N	UINT	I、Q、M、L、D 或常数	将循环移位的位数。
OUT	BYTE、WORD、DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“循环右移”操作将输入 IN 的变量数据逐位循环右移，并通过输出 OUT 查询结果。参数 N 指定将循环移位的位数。用挤出的位填充因循环移位空出的位。

参数 N 的值为“0”时，输入 IN 的值将被复制到输出 OUT 的变量中。

当参数 N 的值大于位数时，输入 IN 的变量值将按其可用位数进行循环移位。

下图显示了一个 DWORD 数据类型的变量数据是如何循环右移三位的：

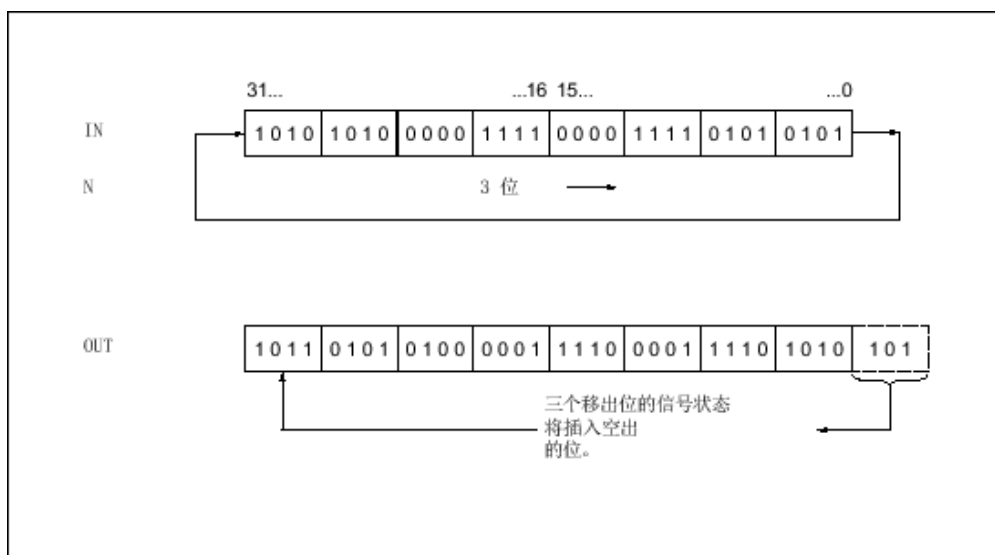


图 7-2

只有使能输出 ENO 的信号状态为“1”时，才执行“循环右移”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“循环右移”操作可以放置在程序段中的任何位置。

实例

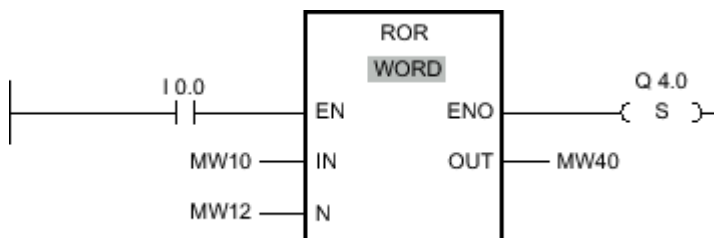


图 7-2

IN	MW10 = 0000 1111 1001 0101
N	MW12 = 5
OUT	MW40 = 1010 1000 0111 1100

如果输入 I 0.0 的信号状态为“1”，则执行“循环右移”操作。变量 MW10 的数据将循环右移五位。结果放在输出 MW40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 LAD 元素 (页 417)
- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.1 ROL : 循环左移

符号

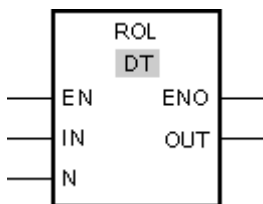


图 7-2

参数	数据类型	存储区	描述
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	要循环移位的值。
N	UINT	I、Q、M、L、D 或常数	将循环移位的位数。
OUT	BYTE、WORD、DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

描述

可以使用“循环左移”操作将输入 IN 的变量数据逐位循环左移，并通过输出 OUT 查询结果。参数 N 指定将循环移位的位数。用挤出的位填充因循环移位空出的位。

参数 N 的值为“0”时，输入 IN 的值将被复制到输出 OUT 的变量中。

当参数 N 的值大于位数时，输入 IN 的变量值将按其可用位数进行循环移位。

下图显示了一个 DWORD 数据类型的变量数据是如何循环左移三位的：

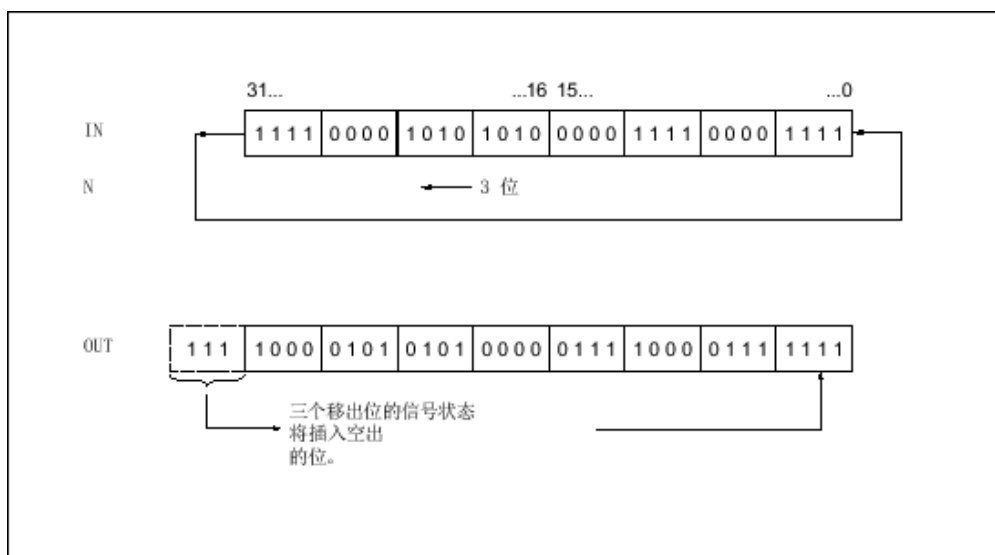


图 7-2

只有使能输出 ENO 的信号状态为“1”时，才执行“循环左移”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“循环左移”操作可以放置在程序段中的任何位置。

实例

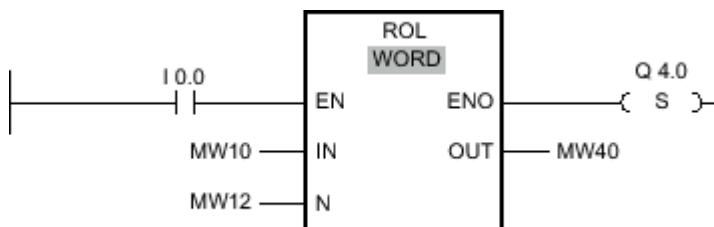


图 7-2

IN	MW10 = 1010 1000 1111 0110
OUT	MW12 = 5
N	MW40 = 0001 1110 1101 0101

如果输入 I 0.0 的信号状态为“1”，则执行“循环左移”操作。变量 MW10 的数据将循环左移五位。结果放在输出 MW40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 LAD 元素 (页 417)

- 更改 LAD 元素 (页 425)
- 将操作数插入 LAD 指令 (页 430)
- 选择 LAD 元素的数据类型 (页 419)

7.8.1.2 FBD

7.8.1.2 位逻辑

7.8.1.2 AND 逻辑运算

符号

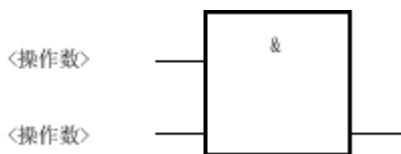


图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	操作数表示将查询哪个位的信号状态。

说明

可以使用“AND”逻辑运算来查询两个或更多指定操作数的信号状态，并根据 AND 运算真值表评估这些信号状态。

如果所有操作数的信号状态都为“1”，则满足条件，并且运算返回结果“1”。如果其中一个操作数的信号状态为“0”，则不满足条件，并且该运算生成结果“0”。

如果“AND”逻辑运算是逻辑串的第一个运算，则它会将其信号状态的查询结果存入 RLO 位。

如果“AND”逻辑运算不是逻辑串的第一个运算，则它会将其信号状态的查询结果与 RLO 位中存储的值进行逻辑组合。该逻辑组合将根据 AND 运算真值表来执行。

实例



图 7-2

当输入 I 0.0 与 I 0.1 的信号状态都为“1”时，输出 Q 4.0 置位。

参见

AND 运算真值表 (页 743)
 插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 将操作数插入 FBD 指令 (页 457)
 检测传送带方向实例 (页 569)
 控制室温实例 (页 575)

7.8.1.2 AND 运算真值表

逻辑运算结果

信号状态 I1.0	信号状态 I1.1	逻辑运算结果
1	1	1
0	1	0
1	0	0
0	0	0

参见

AND 逻辑运算 (页 742)

7.8.1.2 OR 逻辑运算

符号



图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	操作数表示将查询哪个位的信号状态。

说明

可以使用“OR”逻辑运算来查询两个或更多指定操作数的信号状态，并根据 OR 运算真值表评估这些信号状态。

如果有一个操作数的信号状态为“1”，则满足条件，并且运算返回结果“1”。如果所有操作数的信号状态都为“0”，则不满足条件，并且该运算生成结果“0”。

如果“OR”逻辑运算是逻辑串的第一个运算，则它会将其信号状态的查询结果存入 RLO 位。

如果“OR”逻辑运算不是逻辑串的第一个运算，则它会将其信号状态的查询结果与 RLO 位中存储的值进行逻辑组合。该逻辑组合将根据 OR 运算真值表来执行。

实例

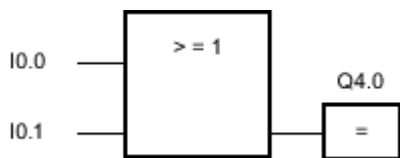


图 7-2

当输入 I0.0 或 I0.1 的信号状态为“1”时，输出 Q 4.0 置位。

参见

- OR 运算真值表 (页 744)*
- 插入 FBD 元素 (页 447)*
- 更改 FBD 元素 (页 454)*
- 将操作数插入 FBD 指令 (页 457)*
- 控制传送带实例 (页 568)*
- 检测存储区域的填充量实例 (页 570)*
- 控制室温实例 (页 575)*

7.8.1.2 OR 运算真值表

逻辑运算结果

信号状态 I1.0	信号状态 I1.1	逻辑运算结果
1	0	1
0	1	1
1	1	1
0	0	0

参见

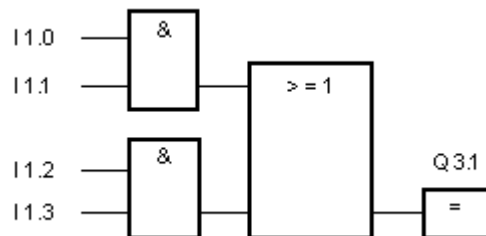
- OR 逻辑运算 (页 743)*

7.8.1.2 先与后或逻辑运算和先或后与逻辑运算

说明

使用“先与后或”逻辑运算，可以根据 OR 运算真值表查询信号状态的结果。对于先与后或逻辑运算，至少有一个 AND 逻辑运算结果为真时，信号状态才为 1。

实例



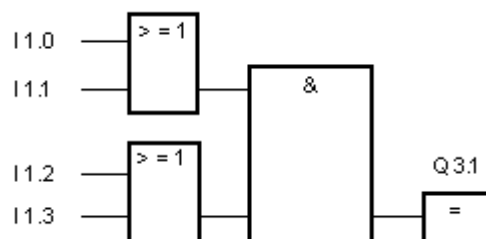
当至少有一个 AND 逻辑运算结果为真时，输出 Q 3.1 置位。

当 AND 逻辑运算结果都不为真时，输出 Q 3.1 复位。

说明

使用“先或后与”逻辑运算，可以根据 AND 运算真值表查询信号状态的结果。对于先或后与逻辑运算，全部 OR 逻辑运算结果必须都为真，信号状态才为 1。

实例



当两个 OR 逻辑运算结果都为真时，输出 Q 3.1 置位。

当有一个或两个 OR 逻辑运算结果不为真时，输出 Q 3.1 复位。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 X：异或逻辑运算

符号

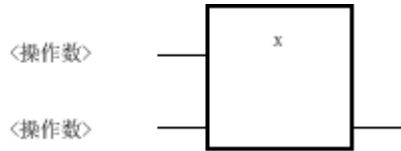


图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	操作数表示将查询哪个位的信号状态。

说明

可以使用“异或”运算来根据异或运算真值表查询信号状态的结果。

对于“异或”运算，当两个指定操作数之一的信号状态为“1”时，其信号状态为“1”。XOR 元素用于查询两个以上操作数时，如果有奇数个查询的操作数返回结果“1”，则公共结果为“1”。

实例

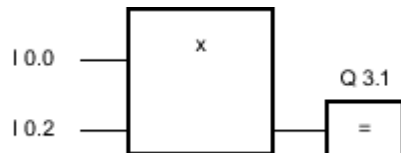


图 7-2

当输入 I 0.0 或 I 0.2 只有一个信号状态为“1”时，输出 Q 3.1 置位。

参见

- 异或运算真值表 (页 746)
- 插入 FBD 元素 (页 447)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 异或运算真值表

逻辑运算结果

信号状态 I1.0	信号状态 I1.1	逻辑运算结果
1	0	1
0	1	1

信号状态 I1.0	信号状态 I1.1	逻辑运算结果
1	1	0
0	0	0

参见

更改 FBD 元素 (页 454)

7.8.1.2 插入二进制输入**符号**

图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	操作数表示将查询哪个位的信号状态。

说明

可以使用“插入二进制输入”操作在标记后“与”、“或”或者“异或”功能框中再插入一个二进制输入。

实例

图 7-2

当输入 I 1.0、I 1.1 和 I 1.2 的信号状态都为“1”时，输出 Q 4.0 置位。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 取反二进制值

符号



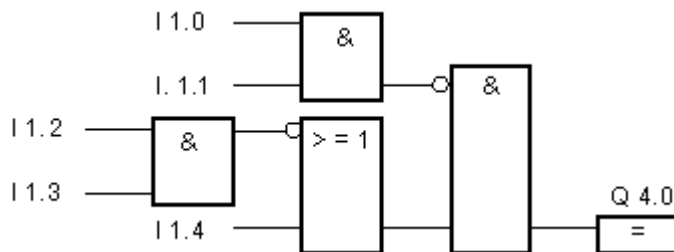
说明

可以使用“取反二进制值”操作将逻辑运算的结果取反。

如果逻辑运算的结果被取反，请记住下列规则：

- 如果是对“与”功能框或者“或”功能框的第一个输入处的逻辑运算结果取反，则不进行嵌套。
- 如果被取反的对象不是“或”功能框的第一个输入处的逻辑运算结果，则在此“或”逻辑运算中将包括在此输入之前的全部二进制逻辑运算。
- 如果被取反的对象不是“与”功能框的第一个输入处的逻辑运算结果，则在此“与”逻辑运算中将包括在此输入之前的全部二进制逻辑运算。

实例



满足下列条件时，将置位输出 Q 4.0：

- 输入 I 1.0 与 I 1.1 的信号状态为“0”。
- 输入 I 1.2 与 I 1.3 的信号状态为“0”，或者输入 I 1.4 的信号状态为“1”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

检测传送带方向实例 (页 569)

检测存储区域的填充量实例 (页 570)

控制室温实例 (页 575)

7.8.1.2 =: 赋值

符号



图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	RLO 为“1”时置位的操作数。

说明

可以使用“赋值”操作置位指定操作数的位。如果该功能框输入端的逻辑运算结果 (RLO) 的信号状态为“1”，指定的操作数的信号状态将被置位为“1”。如果该功能框输入端的信号状态为“0”，指定操作数的位将复位为“0”。

该操作不会影响 RLO。功能框输入端的 RLO 将直接传送到功能框输出上。

放置

“赋值”操作可以放置在逻辑串中的任何位置。

实例

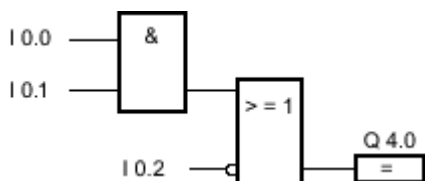


图 7-2

满足下列条件之一时，将置位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 I 0.2 的信号状态为“0”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

检测存储区域的填充量实例 (页 570)

控制室温实例 (页 575)

7.8.1.2 /=: 赋值取反

符号



参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	RLO 为“0”时置位的操作数。

说明

“赋值取反”操作将逻辑运算结果 (RLO) 取反并将取反结果分配给功能框上方的操作数。如果功能框输入端的 RLO 为“1”，则复位二进制操作数。如果功能框输入端的 RLO 为“0”，则将二进制操作数的信号状态置位为“1”。

该操作不会影响 RLO。功能框输入端的 RLO 将直接传送到功能框输出上。

放置

“赋值取反”操作可以放置在逻辑串中的任何位置。

实例

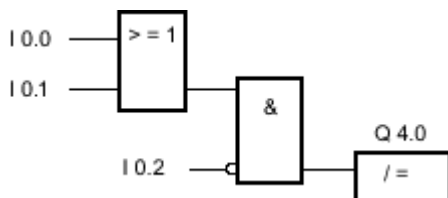


图 7-2

满足下列条件时，将复位输出 Q 4.0：

- 输入 I0.0 或 I0.1 的信号状态为“1”。
- 输入 I0.2 的信号状态为“0”。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 R：复位输出

符号



图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	RLO 为“0”时复位的操作数。

说明

可以使用“复位输出”操作将指定操作数的信号状态复位为“0”。

只有当功能框输入端的逻辑运算结果 (RLO) 为“1”时，才会执行该操作。如果功能框输入端的信号状态为“1”，指定的操作数将被复位为“0”。如果功能框输入端的逻辑运算结果为“0”，则指定操作数的信号状态保持不变。

该运算不会影响 RLO。功能框输入端的 RLO 将直接传送到功能框输出上。

放置

“复位输出”操作可以放置在逻辑串中的任何位置。

实例

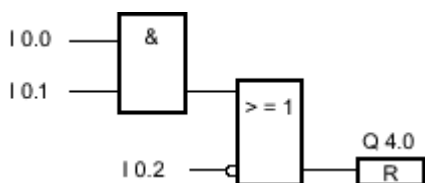


图 7-2

满足下列条件之一时，将复位输出 Q 4.0：

- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 I 0.2 的信号状态为“0”。

参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 将操作数插入 FBD 指令 (页 457)
 控制传送带实例 (页 568)
 检测传送带方向实例 (页 569)

7.8.1.2 S：置位输出

符号



图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	RLO 为“1”时置位的操作数。

说明

可以使用“置位输出”操作将指定操作数的信号状态置位为“1”。

只有当功能框输入端的逻辑运算结果 (RLO) 为“1”时，才会执行该操作。如果功能框输入端的信号状态为“1”，指定的操作数将被置位为“1”。如果功能框输入端的逻辑运算结果为“0”，则指定操作数的信号状态保持不变。

该运算不会影响 RLO。功能框输入端的 RLO 将直接传送到功能框输出上。

放置

“置位输出”操作可以放置在逻辑串中的任何位置。

实例

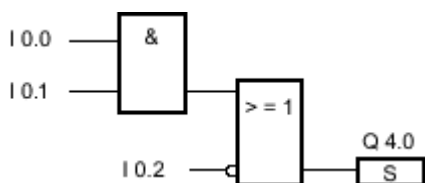


图 7-2

满足下列条件之一时，将置位输出 Q 4.0：

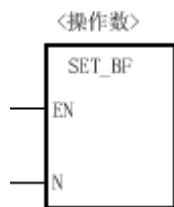
- 输入 I 0.0 与 I 0.1 的信号状态为“1”。
- 输入 I 0.2 的信号状态为“0”。

参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 将操作数插入 FBD 指令 (页 457)
 控制传送带实例 (页 568)
 检测传送带方向实例 (页 569)

7.8.1.2 SET_BF : 置位位域

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	上一逻辑运算的结果
<操作数>	BOOL	I、Q、M、D、L	指向要置位的第一个位的指针
N	UINT	常数	要置位的位数

说明

可以使用“置位位域”操作从某个指定地址开始的多个位的置位。您可以使用参数 N 的值来指定将要置位的位数。要置位的第一个位的地址用操作数 (<操作数>) 定义。如果参数 N 的值大于所选字节中的位数，则将置位下一个字节的位。在明确复位这些位 (例如，通过另一个操作) 之前，它们会保持置位。

只有输入 EN 的信号状态为“1”时，才会执行该操作。如果输入 EN 的信号状态为“0”，则不执行该操作。

放置

“置位位域”操作只能放置在逻辑串的末端。

实例

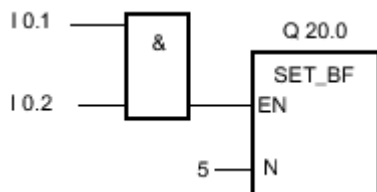


图 7-2

如果输入 I 0.1 与 I 0.2 的信号状态为“1”，则置位输出 Q 20.0、Q 20.1、Q 20.2、Q 20.3 和 Q 20.4。

参见

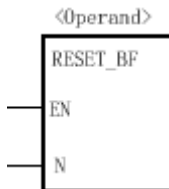
插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 RESET_BF : 复位位域

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	上一逻辑运算的结果
<操作数>	BOOL	I、Q、M、D、L	指向要复位的第一个位的指针
N	UINT	常数	要复位的位的编号

说明

可以使用“复位位域”操作复位从某个指定地址开始的多个位的置位。您可以使用参数 N 的值来指定将要复位的位数。要复位的第一个位的地址用操作数 (<操作数>) 定义。如果参数 N 的值大于所选字节中的位数，则将复位下一个字节的位。在明确置位这些位 (例如, 通过另一个操作) 之前, 它们会保持复位。

只有输入 EN 的信号状态为“1”时, 才会执行该操作。如果输入 EN 的信号状态为“0”, 则不执行该操作。

放置

“复位位域”操作只能放置在逻辑串的末端。

实例

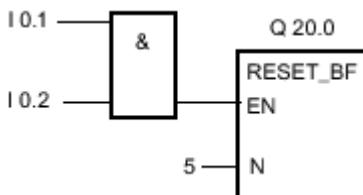


图 7-2

如果输入 I 0.0 与 I 0.1 的信号状态为“1”, 则复位输出 Q 20.0、Q 20.1、Q 20.2、Q 20.3 和 Q 20.4。

参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 将操作数插入 FBD 指令 (页 457)

7.8.1.2 SR : 置位复位触发器

符号



图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	指定的操作数。
S	BOOL	I、Q、M、D、L	启用置位
R1	BOOL	I、Q、M、D、L	启用复位
Q	BOOL	I、Q、M、D、L	指定操作数的信号状态

说明

可以使用“置位复位触发器”操作基于输入 S 和 R1 的信号状态置位或复位指定操作数的位。如果输入 S 的信号状态为“1”且输入 R1 的信号状态为“0”，指定的操作数将置位为“1”。如果输入 S 的信号状态为“0”且输入 R1 的信号状态为“1”，指定的操作数将复位为“0”。

输入 R1 优先于输入 S。如果两个输入 S 和 R1 的信号状态均为“1”，指定操作数的信号状态将复位为“0”。

如果两个输入 R 和 S1 的信号状态为“0”，则不执行该操作。操作数的信号状态因此保持不变。

操作数的当前信号状态传送到输出 Q 并且可在那进行查询。

放置

“置位复位触发器”操作可以放置在逻辑串的中间或末端。

实例

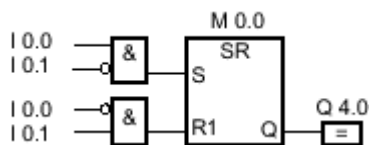


图 7-2

满足以下条件时，将置位存储器 M 0.0 和输出 Q 4.0：

- 输入 I 0.0 的信号状态为“1”。
- 输入 I 0.1 的信号状态为“0”。

满足以下条件时，将复位存储器 M 0.0 和输出 Q 4.0：

- 输入 I 0.0 的信号状态为“0”。
- 输入 I 0.1 的信号状态为“1”。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 RS：复位置位触发器

符号



图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	指定的操作数。
R	BOOL	I、Q、M、D、L	启用复位
S1	BOOL	I、Q、M、D、L	启用置位
Q	BOOL	I、Q、M、D、L	指定操作数的信号状态

说明

可以使用“复位置位触发器”操作基于输入 R 和 S1 的信号状态置位或复位指定操作数的位。如果输入 R 的信号状态为“1”且输入 S1 的信号状态为“0”，指定的操作数将复位为“0”。如果输入 R 的信号状态为“0”且输入 S1 的信号状态为“1”，指定的操作数将置位为“1”。

输入 S1 优先于输入 R。如果两个输入 R 和 S1 的信号状态均为“1”，指定操作数的信号状态将置位为“1”。

如果两个输入 R 和 S1 的信号状态为“0”，则不执行该操作。操作数的信号状态因此保持不变。操作数的当前信号状态传送到输出 Q 并且可在那进行查询。

放置

“复位置位触发器”操作可以放置在逻辑串的中间或末端。

实例

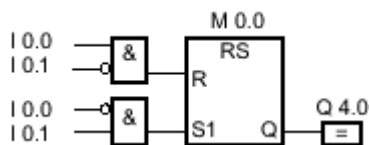


图 7-2

满足以下条件时，将复位位存储器 M 0.0 和输出 Q 4.0：

- 输入 I 0.0 的信号状态为“1”。
- 输入 I 0.1 的信号状态为“0”。

满足以下条件时，将置位位存储器 M 0.0 和输出 Q 4.0：

- 输入 I 0.0 的信号状态为“0”。
- 输入 I 0.1 的信号状态为“1”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 P：扫描操作数的信号上升沿

符号



参数	数据类型	存储区	说明
<操作数 1>	BOOL	I、Q、M、D、L (页 293)	要查询的信号
<操作数 2>	BOOL	I、Q、M、D	保存上一次查询的信号状态的沿存储器位。

说明

可以使用“扫描操作数的信号上升沿”操作确定指定操作数 (<操作数 1>) 的信号状态是否从“0”变为“1”。该运算比较 <操作数 1> 的当前信号状态与保存在 <操作数 2> 中的上一次查询的信号状态。如果该操作检测到逻辑运算结果从“0”变为“1”，则说明出现了一个上升沿。

如果检测到上升沿，该操作的输出的信号状态将为“1”。在任何其它情况下，操作的输出的信号状态为“0”。

注意

沿存储器位的地址在程序中最多只能使用一次，否则，该存储器位会被覆盖。这将影响沿评估，并且结果会变得不明确。

放置

“扫描操作数的信号上升沿”操作可以放置在逻辑串中的任何位置。

实例

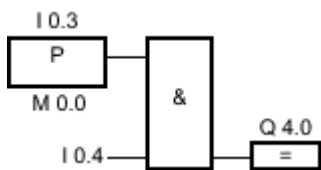


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.3 出现了一个上升沿。
- 输入 I 0.4 的信号状态为“1”。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 检测传送带方向实例 (页 569)

7.8.1.2 N：扫描操作数的信号下降沿

符号



参数	数据类型	存储区	说明
<操作数 1>	BOOL	I、Q、M、D、L (页 293)	要查询的信号

参数	数据类型	存储区	说明
<操作数 2>	BOOL	I、Q、M、D	保存上一次查询的信号状态的沿存储器位。

说明

可以使用“扫描操作数的信号下降沿”操作查询指定操作数（<操作数 1>）信号状态从“1”到“0”的变化。该运算比较 <操作数 1> 的当前信号状态与保存在 <操作数 2> 中的上一次查询的信号状态。如果该操作检测到逻辑运算结果从“1”变为“0”，则说明出现了一个下降沿。

如果检测到下降沿，该操作的输出的信号状态将为“1”。

如果未检测到下降沿，操作的输出的信号状态将为“0”。

注意

沿存储器位的地址在程序中最多只能使用一次，否则，该存储器位会被覆盖。这将影响沿评估，并且结果会变得不明确。

放置

“扫描操作数的信号下降沿”操作可以放置在逻辑串中的任何位置。

实例

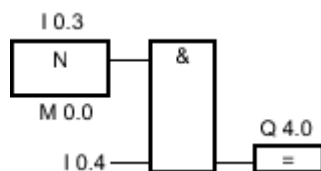


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.3 出现了一个下降沿。
- 输入 I 0.4 的信号状态为“1”。

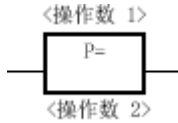
参见

插入 FBD 元素 (页 447)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 P= : 在信号上升沿置位操作数

符号



参数	数据类型	存储区	说明
<操作数 1>	BOOL	I、Q、M、D、L (页 293)	上升沿置位的操作数。
<操作数 2>	BOOL	I、Q、M、D	沿存储器位

说明

可以使用“在信号上升沿置位操作数”操作检测逻辑运算结果 (RLO) 从“0”到“1”的变化。查询时，该操作将当前 RLO 与沿存储器位中保存的上次查询的 RLO 进行比较。如果该操作检测到 RLO 从“0”变为“1”，则说明出现了一个上升沿。

如果检测到上升沿，<操作数 1> 的信号状态将置位为“1”一个程序周期。在任何其它情况下，操作数的信号状态为“0”。

注意

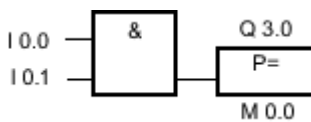
沿存储器位的地址在程序中最多只能使用一次，否则，该存储器位会被覆盖。这将影响沿评估，并且结果会变得不明确。

该运算不会影响 RLO。功能框输入端的 RLO 将直接传送到功能框输出。

放置

“在信号上升沿置位操作数”操作可以放置在逻辑串的中间或末端。

实例



如果功能框输入端的信号状态从“0”变为“1”(上升沿)，输出 Q 3.0 将置位一个程序周期。在任何其它情况下，输出 Q 3.0 的信号状态为“0”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)
将操作数插入 FBD 指令 (页 457)

7.8.1.2 N= : 在信号下降沿置位操作数

符号



参数	数据类型	存储区	说明
<操作数 1>	BOOL	I、Q、M、D、L (页 293)	下降沿置位的操作数。
<操作数 2>	BOOL	I、Q、M、D、L	沿存储器位

说明

可以使用“在信号下降沿置位操作数”操作检测逻辑运算结果 (RLO) 从“1”到“0”的变化。查询时，该操作将当前 RLO 与沿存储器位中保存的上次查询的 RLO 进行比较。如果该操作检测到 RLO 从“1”变为“0”，则说明出现了一个下降沿。

如果检测到下降沿，<操作数 1> 的信号状态将置位为“1”一个程序周期。在任何其它情况下，操作数的信号状态为“0”。

注意

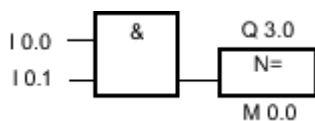
沿存储器位的地址在程序中最多只能使用一次，否则，该存储器位会被覆盖。这将影响沿评估，并且结果会变得不明确。

该运算不会影响 RLO。功能框输入端的 RLO 将直接传送到功能框输出。

放置

“在信号下降沿置位操作数”操作可以放置在逻辑串的中间或末端。

实例



如果功能框输入端的信号状态从“1”变为“0”(下降沿), 输出 Q 3.0 将置位一个程序周期。在任何其它情况下, 输出 Q 3.0 的信号状态为“0”。

参见

- 插入 FBD 元素 (页 447)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 P_TRIG : 在信号上升沿置位输出

符号



参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	保存上一次查询的 RLO 的沿存储器位。
CLK	BOOL	I、Q、M、D、L	当前 RLO。
Q	BOOL	I、Q、M、D、L	沿评估的结果

说明

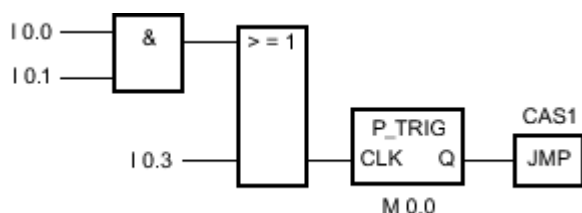
您可以使用“在信号上升沿置位输出”操作查询逻辑运算结果的信号状态从“0”到“1”的变化。该运算比较逻辑运算结果的当前信号状态与保存在沿存储器位中的上一次查询的信号状态。如果该操作检测到 RLO 从“0”变为“1”, 则说明出现了一个上升沿。

如果检测到上升沿, 该操作的输出的信号状态将为“1”。在任何其它情况下, 操作的输出的信号状态为“0”。

放置

“在信号上升沿置位输出”操作可以放置在逻辑串的中间或末端。

实例



前导位逻辑运算的 RLO 保存在沿存储器位 M 0.0 中。如果检测到 RLO 的信号状态从“0”变为“1”, 程序将跳转到标签 CAS1。

参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 将操作数插入 FBD 指令 (页 457)

7.8.1.2 N_TRIG : 在信号下降沿置位输出

符号



图 7-2

参数	数据类型	存储区	说明
<操作数>	BOOL	I、Q、M、D、L (页 293)	保存上一次查询的 RLO 的沿存储器位。
CLK	BOOL	I、Q、M、D、L	当前 RLO
Q	BOOL	I、Q、M、D、L	沿评估的结果

说明

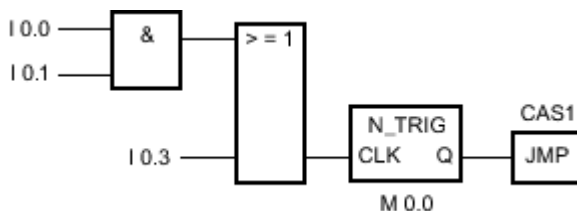
可以使用“在信号下降沿置位输出”操作检测逻辑运算结果 (RLO) 的信号状态从“1”到“0”的变化。该运算比较逻辑运算结果的当前信号状态与保存在沿存储器位中的上一次查询的信号状态。如果该操作检测到能流从“1”变为“0”，则说明出现了一个下降沿。

如果检测到下降沿，该操作的输出的信号状态将为“1”。在任何其它情况下，操作的输出的信号状态为“0”。

放置

“在信号下降沿置位输出”操作可以放置在逻辑串的中间或末端。

实例



前导位逻辑运算的 RLO 保存在沿存储器位 M 0.0 中。如果检测到 RLO 的信号状态从“1”变为“0”，程序将跳转到跳转标签 CAS1。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 定时器

7.8.1.2 TP : 脉冲定时器

符号

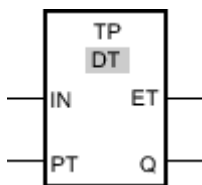


图 7-2

参数	数据类型	存储区	说明
IN	BOOL	I、Q、M、D、L (页 293)	启动输入
PT	TIME	I、Q、M、D、L 或常数	脉冲的持续时间。PT 必须为正值。
ET	TIME	I、Q、M、D、L	已耗时间
Q	BOOL	I、Q、M、D、L	脉冲输出

说明

可以使用“生成脉冲”操作置位输出 Q 预设的一段时间。当输入 IN 的逻辑运算结果 (RLO) 从“0”变为“1”时，启动该操作。启动该操作后，为 PT 编程的时间即开始计时。无论输入信号在接下来的过程中为何，输出 Q 都将置位一段时间，即 PT。即使检测到新的上升沿，只要定时器 PT 在计时就不会影响输出 Q 的信号状态。

可以通过输出 ET 查询当前定时器功能已运行了多长时间。该时间从 T#0s 开始计时并在达到为 PT 定时器设置的值后结束。只要定时器 PT 在运行且输入 IN 的信号状态为“1”，就可以查询输出 ET 的值。

插入“生成脉冲”操作时，将创建一个保存操作数据的背景数据块。

放置

“生成脉冲”操作需要一个用于沿评估的前导逻辑操作。该操作可以放置在逻辑串的中间或末端。

脉冲图

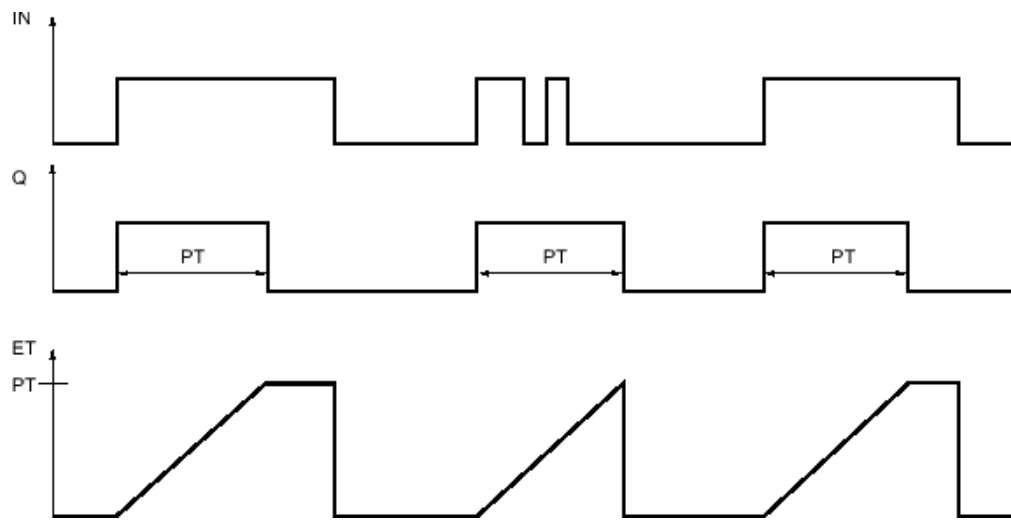


图 7-2

参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 选择 FBD 元素的数据类型 (页 419)
 将操作数插入 FBD 指令 (页 457)

7.8.1.2 TON : 接通延迟

符号

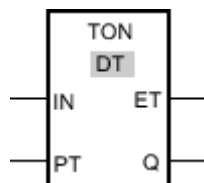


图 7-2

参数	数据类型	存储区	说明
IN	BOOL	I、Q、M、D、L (页 293)	启动输入
PT	TIME	I、Q、M、D、L 或常数	输入 IN 上升沿延迟的时间。
ET	TIME	I、Q、M、D、L	已耗时间
Q	BOOL	I、Q、M、D、L	延迟了时间 PT 的输出。

说明

可以使用“接通延迟”操作使上升沿延迟 PT 端设置的时间。输入 IN 的逻辑运算结果 (RLO) 从“0”变为“1”(上升沿) 时, 执行“接通延迟”操作。启动该操作后, 为 PT 设置的时间即开始计时。PT 时间用完时, 输出 Q 的信号状态为“1”。只要启动输入仍为“1”, 输出 Q 就保持置位。如果启动输入的信号状态从“1”变为“0”, 则复位输出 Q。在启动输入端检测到新的上升沿时, 该定时器功能将再次启动。

输出 ET 提供从输入 IN 的最近一次上升沿后经过的时间。该时间从 T#0s 开始计时并在达到为 PT 定时器设置的值后结束。只要输入 IN 的信号状态为“1”, 就可通过输出 ET 查询过去的时间。如果输入 IN 变为“0”, 输出 ET 将被复位为值 T#0。

插入“接通延迟”操作时, 将创建一个保存操作数据的背景数据块。

放置

“接通延迟”操作需要一个用于沿评估的前导逻辑操作。该操作可以放置在逻辑串的中间或末端。

脉冲图

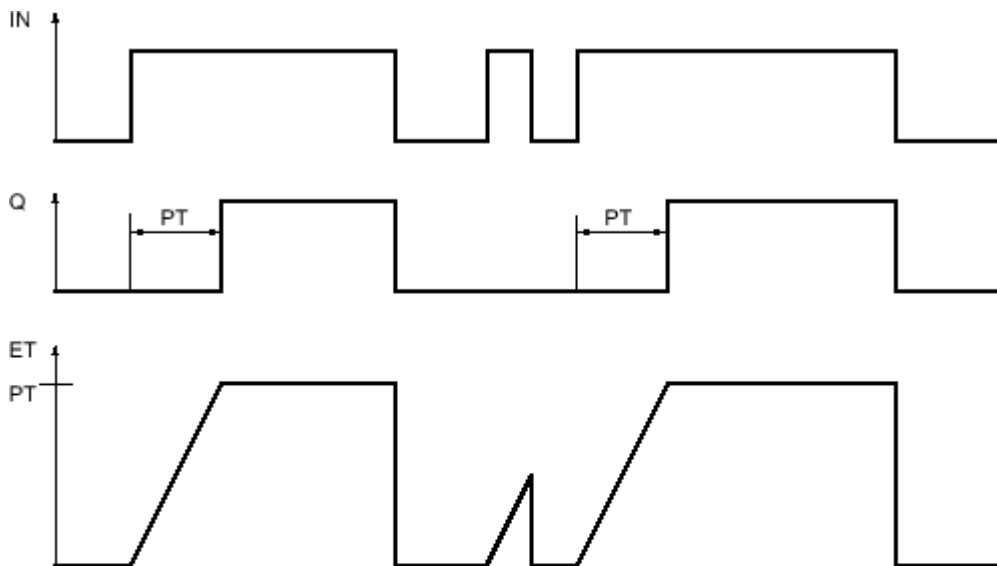


图 7-2

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)
 选择 FBD 元素的数据类型 (页 419)
 将操作数插入 FBD 指令 (页 457)
 控制室温实例 (页 575)

7.8.1.2 TOF : 关断延迟

符号

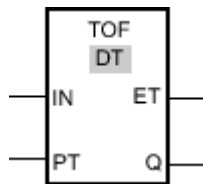


图 7-2

参数	数据类型	存储区	说明
IN	BOOL	I、Q、M、D、L (页 293)	启动输入
PT	TIME	I、Q、M、D、L 或常数	输入 IN 下降沿延迟的时间。PT 必须为正值。
ET	TIME	I、Q、M、D、L	已耗时间
Q	BOOL	I、Q、M、D、L	延迟了时间 PT 的输出。

说明

可以使用“关断延迟”操作使下降沿延迟 PT 端设置的时间。输入 IN 的逻辑运算结果 (RLO) 从“0”变为“1”时，输出 Q 置位。如果输入 IN 的信号状态切换回“0”，PT 端设置的时间开始计时。只要 PT 端设置的时间仍在计时，输出 Q 就保持置位。PT 时间计时完后，输出 Q 将复位。如果输入 IN 的信号状态在 PT 端设置的时间计时完前变为“1”，则定时器将复位。输出 Q 的信号状态仍将为“1”。

可以通过输出 ET 查询当前定时器功能已运行了多长时间。该时间从 T#0s 开始计时并在达到为 PT 定时器设置的值后结束。PT 端设置的时间用完时，输出 ET 保持设置为当前值，直到输入 IN 重新变回“1”。

插入“关断延迟”操作时，将创建一个保存操作数据的背景数据块。

放置

“关断延迟”操作需要一个用于沿评估的前导逻辑操作。该操作可以放置在逻辑串的中间或末端。

脉冲图

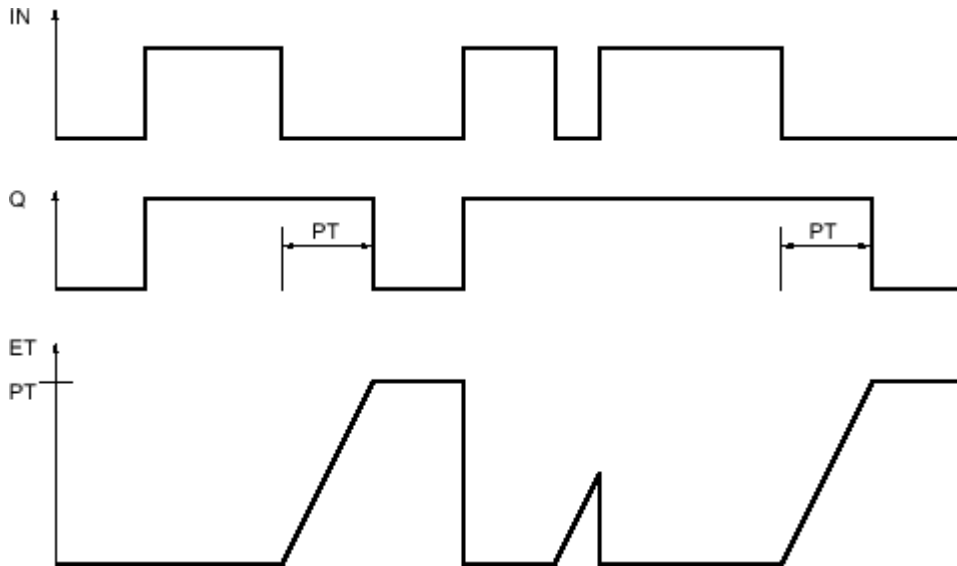


图 7-2

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 选择 FBD 元素的数据类型 (页 419)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 TONR : 保持型接通延迟

符号

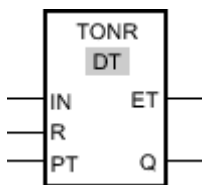


图 7-2

参数	数据类型	存储区	说明
IN	BOOL	I、Q、M、D、L (页 293)	启动输入
R	BOOL	I、Q、M、D、L	复位输入
PT	TIME	I、Q、M、D、L 或常数	时间记录的最长持续时间
ET	TIME	I、Q、M、D、L	累计的时间

参数	数据类型	存储区	说明
Q	BOOL	I、Q、M、D、L	PT 时间计时完时置位的输出。

说明

可以使用“时间累加器”操作在参数 PT 设置的时段内累计时间值。输入 IN 的信号状态变为“1”时，执行该操作且 PT 端设置的时间开始计时。在 PT 端设置的时间计时期间，累加输入 IN 的信号状态为“1”时记录的时间值。累计的时间写入到输出 ET 并可在其中查询。PT 时间用完时，输出 Q 的信号状态为“1”。

无论启动输入端的信号状态为何，输入 R 都会复位定时器功能和输出 Q。

插入“时间累加器”操作时，将创建一个保存操作数据的背景数据块。

放置

“时间累加器”操作需要一个前导逻辑操作。该操作可以放置在逻辑串的中间或末端。

脉冲图

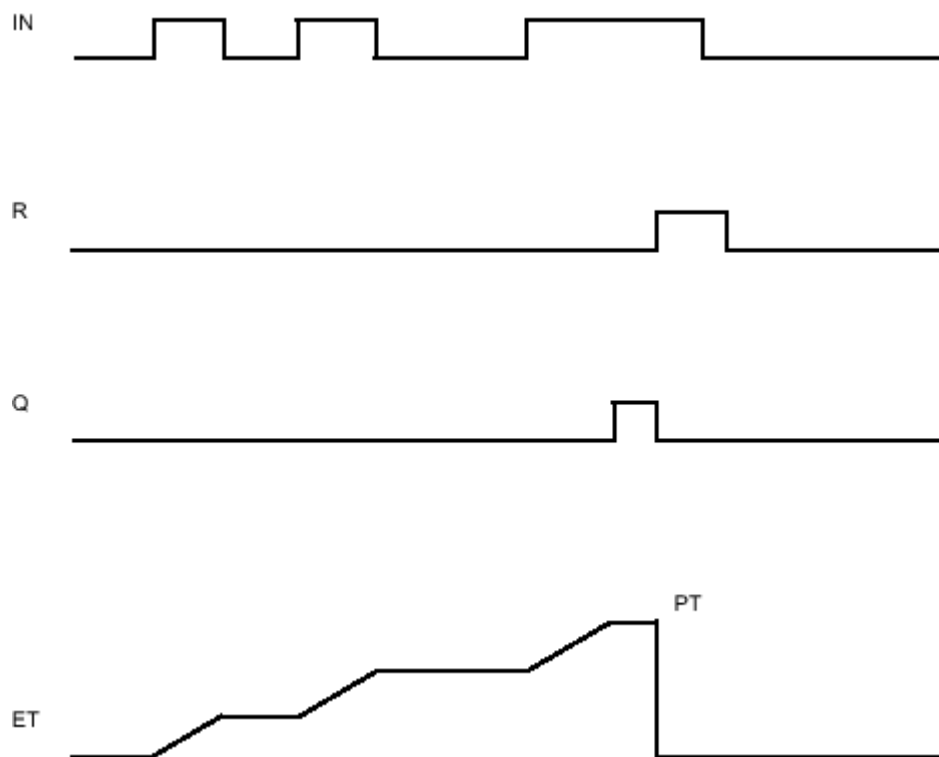


图 7-2

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

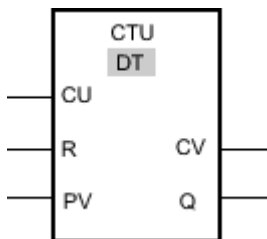
选择 FBD 元素的数据类型 (页 419)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 计数器

7.8.1.2 CTU：加计数

符号



参数 英语	数据类型	存储区	说明
CU	BOOL	I、Q、M、D、L (页 293)	计数输入
R	BOOL	I、Q、M、D、L	复位输入
PV	SINT、INT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L 或常数	预置计数值
CV	SINT、INT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L	实际计数值
Q	BOOL	I、Q、M、D、L	计数器状态

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“加计数”操作使输出 CV 的值递增。如果输入 CU 的信号状态从“0”变为“1”(上升沿), 则执行该操作且输出 CV 的当前计数值加“1”。首次执行该操作时, 输出 CV 的当前计数值设置为“0”。每检测到一个上升沿, 计数值就会连续递增, 直到其达到输出 CV 指定数据类型的上限。达到上限时, 输入 CU 的信号状态就不再影响该操作。

可通过输出 Q 查询计数器状态。输出 Q 的信号状态由输入参数 PV 决定。如果当前计数值大于或等于参数 PV 的值, 则输出 Q 的信号状态设置为“1”。在任何其它情况下, 输出 Q 的信号状态为“0”。

输入 R 的信号状态变为“1”时, 输出 CV 的值复位为“0”。只要输入 R 的信号状态为“1”, 输入 CU 的信号状态就不会影响该操作。

插入“加计数”操作时，将创建一个保存操作数据的背景数据块。

放置

“加计数”操作需要一个用于沿评估的前导逻辑操作。该操作可以放置在逻辑串的中间或末端。

实例

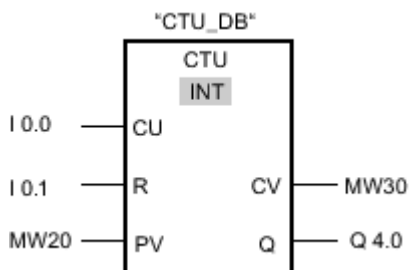


图 7-2

当输入 I 0.0 的信号状态从“0”变为“1”时，执行“加计数”操作且输出 MW30 的当前计数值加“1”。每检测到一个上升沿，计数值就会递增，直到达到指定数据类型上限值 (32767)。

参数 MW20 中的值用于确定输出 Q 4.0 的上限。只要当前计数值大于或等于参数 MW20 中的值，输出 Q 4.0 的信号状态就为“1”。在任何其它情况下，输出 Q 4.0 的信号状态为“0”。

参见

[插入 FBD 元素 \(页 447\)](#)

[更改 FBD 元素 \(页 454\)](#)

[将操作数插入 FBD 指令 \(页 457\)](#)

[选择 FBD 元素的数据类型 \(页 419\)](#)

7.8.1.2 CTD：减计数

符号

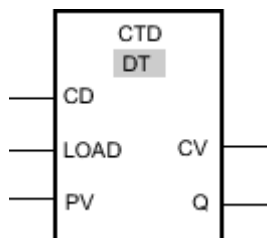


图 7-2

参数 英语	数据类型	存储区	说明
CD	BOOL	I、Q、M、D、L (页 293)	计数输入

参数 英语	数据类型	存储区	说明
LOAD	BOOL	I、Q、M、D、L	装载输入
PV	SINT、UINT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L 或常数	预置计数值
Q	BOOL	I、Q、M、D、L	计数器状态
CV	SINT、UINT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L	实际计数值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“减计数”操作使输出 CV 的值递减。如果输入 CD 的信号状态从“0”变为“1”(上升沿), 则执行该操作且输出 CV 的当前计数值减“1”。首次执行该操作时, 输出 CV 的当前计数值设置为“0”。每检测到一个上升沿, 计数值就会递减, 直到达到指定数据类型的下限值。达到下限值时, 输入 CD 的信号状态就不再影响该操作。

可通过输出 Q 查询计数器状态。如果当前计数值小于或等于“0”, 则输出 Q 的信号状态设置为“1”。在任何其它情况下, 输出 Q 的信号状态为“0”。

输入 LOAD 的信号状态变为“1”时, 输出 CV 的值将设置为参数 PV 的值。只要输入 LOAD 的信号状态为“1”, 输入 CD 的信号状态就不会影响该操作。

插入“减计数”操作时, 将创建一个保存操作数据的背景数据块。

放置

“减计数”操作需要一个用于沿评估的前导逻辑操作。该操作可以放置在逻辑串的中间或末端。

实例

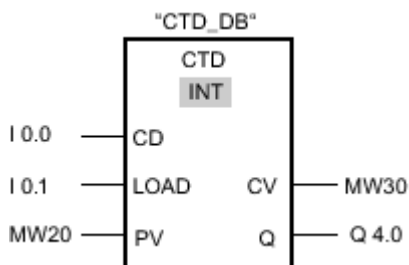


图 7-2

如果输入 I 0.0 的信号状态从“0”变为“1”, 则执行“减计数”操作且输出 MW30 的值减“1”。每检测到一个上升沿, 计数值就会递减, 直到达到指定数据类型的下限值 (-32768)。

只要当前计数值小于或等于“0”, 输出 Q 4.0 的信号状态就为“1”。在任何其它情况下, 输出 Q 4.0 的信号状态为“0”。

参见

更改 FBD 元素 (页 454)

插入 FBD 元素 (页 447)
 将操作数插入 FBD 指令 (页 457)
 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 CTUD：加计数和减计数

符号

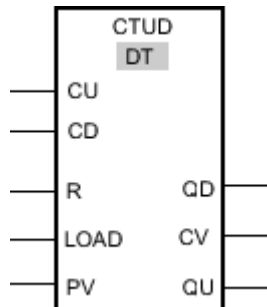


图 7-2

参数 英语	数据类型	存储区	说明
CU	BOOL	I、Q、M、D、L (页 293)	加计数输入
CD	BOOL	I、Q、M、D、L	减计数输入
R	BOOL	I、Q、M、D、L	复位输入
LOAD	BOOL	I、Q、M、D、L	装载输入
PV	SINT、UINT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L 或常数	预置计数值
QU	BOOL	I、Q、M、D、L	递增计数器的状态
QD	BOOL	I、Q、M、D、L	递减计数器的状态
CV	SINT、UINT、DINT、 USINT、UINT、UDINT	I、Q、M、D、L	实际计数值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“加计数和减计数”操作使输出 CV 的计数值递增或递减。如果输入 CU 的信号状态从“0”变为“1”(上升沿),则当前计数值加“1”并存储在输出 CV 中。如果输入 CD 的信号状态从“0”变为“1”(上升沿),则输出 CV 的计数值减“1”。如果在一个程序周期内输入 CU 和 CD 都出现了一个上升沿,则输出 CV 的当前计数值保持不变。

计数值可以一直递增，直到其达到输出 CV 指定数据类型的上限值。如果达到上限值，出现上升沿时计数值也不再递增。达到指定数据类型的下限值时，计数值不再递减。

输入 LOAD 的信号状态变为“1”时，输出 CV 的计数值将设置为参数 PV 的值。只要输入 LOAD 的信号状态变为“1”，输入 CU 和 CD 的信号状态就不会影响该操作。

输入 R 的信号状态变为“1”时，计数值设置为“0”。只要输入 R 的信号状态变为“1”，输入 CU、CD 和 LOAD 的信号状态变化都不能影响“加计数和减计数”操作。

可通过输出 QU 查询递增计数器状态。如果当前计数值大于或等于参数 PV 的值，则输出 QU 的信号状态为“1”。在任何其它情况下，输出 QU 的信号状态为“0”。

可通过输出 QD 查询递减计数器状态。如果当前计数值小于或等于“0”，则输出 QD 的信号状态为“1”。在任何其它情况下，输出 QD 的信号状态为“0”。

插入“加计数和减计数”操作时，将创建一个保存操作数据的背景数据块。

放置

“加计数和减计数”操作需要一个用于沿评估的前导逻辑操作。该操作可以放置在逻辑串的中间或末端。

实例

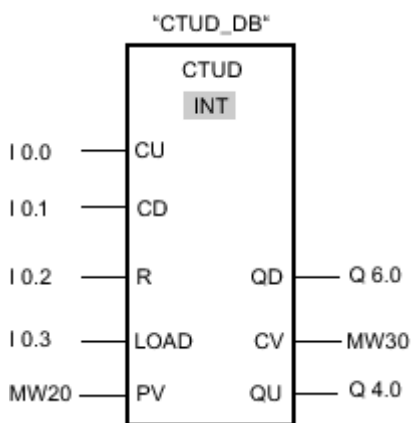


图 7-2

如果输入 I 0.0 或输入 I 0.1 的信号状态从“0”变为“1”(上升沿)，则执行“加计数和减计数”操作。输入 I 0.0 出现了一个上升沿时，当前计数值加“1”并存储在输出 MW30 中。输入 I 0.1 出现了一个上升沿时，当前计数值减“1”并存储在输出 MW30 中。输入 CU 出现了一个上升沿时，计数值将递增，直到其达到上限值 32767 为止。输入 I 0.1 出现了一个上升沿时，计数值将递减，直到其达到下限值 -32768 为止。

只要当前计数值大于或等于输入 MW20 中的值，输出 Q 4.0 的信号状态即为“1”。在任何其它情况下，输出 Q 4.0 的信号状态为“0”。

只要当前计数值小于或等于“0”，输出 Q 6.0 的信号状态就为“1”。在任何其它情况下，输出 Q 6.0 的信号状态为“0”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)
 选择 FBD 元素的数据类型 (页 419)
 检测存储区域的填充量实例 (页 570)

7.8.1.2 高速计数器

7.8.1.2 CTRL_HSC : 在 FBD 中控制高速计数器

符号

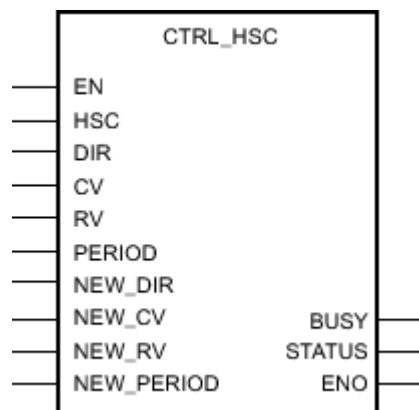


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
HSC	HW_HSC	L、D 或常数	高速计数器的硬件地址 (HW-ID)
DIR	BOOL	I、Q、M、L、D	启用新的计数方向 (请参见 NEW_DIR)
CV	BOOL	I、Q、M、L、D	启用新的计数值 (请参见 NEW_CV)
RV	BOOL	I、Q、M、L、D	启用新的参考值 (请参见 NEW_RV)
PERIOD	BOOL	I、Q、M、L、D	启用新的频率测量周期 (请参见 NEW_PERIOD)
NEW_DIR	INT	I、Q、M、L、D	DIR = TRUE 时装载的计数方向。

参数	数据类型	存储区	说明
NEW_CV	DINT	I、Q、M、L、D	CV = TRUE 时装载的计数值。
NEW_RV	DINT	I、Q、M、L、D	CV = TRUE 时装载的参考值。
NEW_PERIOD	INT	I、Q、M、L、D	PERIOD = TRUE 时装载的频率测量周期。
BUSY	BOOL	I、Q、M、L、D	处理状态
STATUS	WORD	I、Q、M、L、D	运行状态

说明

使用“控制高速计数器”操作，可以通过将新值装载到计数器来进行参数设置和控制 CPU 支持的高速计数器。只有启用要控制的高速计数器后，才能执行该操作。在程序中每个高速计数器只能插入和执行一个“控制高速计数器”操作。用户需要输入高速计数器的硬件标识符 (HW-ID)，在输入 HSC 中分配它的值。

使用“控制高速计数器”操作可以将以下参数值装载到高速计数器：

- 计数方向 (NEW_DIR)：计数方向定义高速计数器是加计数还是减计数。计数方向通过输入 NEW_DIR 的以下值来定义：1 = 加计数，-1 = 减计数。
只有通过程序在参数中设置方向控制后，才能使用“控制高速计数器”操作更改计数方向。
输入 NEW_DIR 指定的计数方向将在输入 DIR 的位被置位时装载到高速计数器。
- 计数值 (NEW_CV)：计数值是高速计数器开始计数时使用的初始值。计数值的范围为 - 2147483648 到 2147483647。
输入 NEW_CV 指定的计数值会在输入 CV 的位被置位时装载到高速计数器。
- 参考值 (NEW_RV)：参考值是高速计数器可以达到的最大值。参考值的范围为 - 2147483648 到 2147483647。
输入 NEW_RV 指定的参考值会在输入 RV 的位被置位时装载到高速计数器。
- 频率测量周期 (NEW_PERIOD)：频率测量周期通过输入 NEW_PERIOD 的以下值来指定：10 = 0.01s，100 = 0.1s，1000 = 1s。
在指定高速计数器的参数中设置“频率测量”功能后，可更新该周期。
输入 NEW_PERIOD 指定的周期值在输入 PERIOD 的位被置位时装载到高速计数器。

只有输入 EN 的信号状态为“1”时，才执行“控制高速计数器”操作。只要该操作在执行，输出 BUSY 的位就会被置位。该操作执行完后，输出 BUSY 的位立即复位。

只有输入 EN 的信号状态为“1”且执行该操作期间没有出错时，才置位使能输出 ENO。

插入“控制高速计数器”操作时，将创建一个保存操作数据的背景数据块。

参数 STATUS

通过输出 STATUS，可以查询“控制高速计数器”操作执行期间是否出错。下表列出了输出 STATUS 的输出值含义：

错误代码 (十六进制)	说明
0	无错误
80A1	高速计数器的硬件标识符无效
80B1	计数方向 (NEW_DIR) 无效
80B2	计数值 (NEW_CV) 无效
80B3	参考值 (NEW_RV) 无效
80B4	频率测量周期 (NEW_PERIOD) 无效
80C0	多次访问高速计数器

参见

更改 FBD 元素 (页 454)

插入 FBD 元素 (页 447)

将操作数插入 FBD 指令 (页 457)

选择 FBD 元素的数据类型 (页 419)

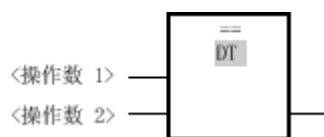
组态高速计数器 (页 243)

给硬件中断 OB 分配参数 (页 384)

7.8.1.2 比较

7.8.1.2 CMP == : 相等

符号



参数	数据类型	存储区	说明
操作数 1	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“等于”运算查询第一个比较值是否等于第二个比较值。 要比较的两个值必须为相同的数据类型。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。 从左到右执行比较。 第一个不同的字符决定比较结果。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'AA'	'AA'	1
'Hello World'	'HelloWorld'	0
'AA'	'aa'	0

放置

“等于”运算可以放置在逻辑串中的任何位置。

实例



图 7-2

满足下列条件时，将置位输出 Q 4.0：

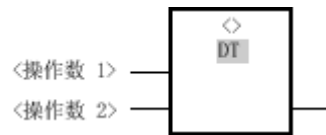
- 比较运算的结果为真 (MW0 = MW2)。
- 输入 I 0.0 的信号状态为“1”。

参见

- 插入 FBD 元素 (页 447)*
- 更改 FBD 元素 (页 454)*
- 将操作数插入 FBD 指令 (页 457)*
- 选择 FBD 元素的数据类型 (页 419)*

7.8.1.2 CMP <> : 不相等

符号



参数	数据类型	存储区	说明
操作数 1	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“不等于”运算查询第一个比较值是否不等于第二个比较值。要比较的两个值必须为相同的数据类型。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'AA'	'aa'	1
'Hello World'	'HelloWorld'	1
'AA'	'AA'	0

放置

“不等于”运算可以放置在逻辑串中的任何位置。

实例

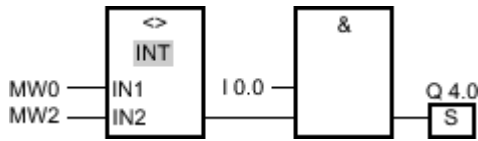


图 7-2

满足下列条件时，将置位输出 Q 4.0：

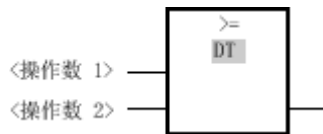
- 比较运算的结果为真 (MW0 \neq MW2)。
- 输入 I 0.0 的信号状态为“1”。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 CMP >= : 大于或等于

符号



参数	数据类型	存储区	说明
操作数 1	USINT、UINT、UDINT、SINT、INT、DINT、REAL、CHAR、STRING、TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、UDINT、SINT、INT、DINT、REAL、CHAR、STRING、TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“大于或等于”运算确定第一个比较值是否大于或等于第二个比较值。要比较的两个值必须为相同的数据类型。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。如果较长字符串的左侧部分和较短字符串相同，则认为较长字符串更大。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'BB'	'AA'	1
'AAA'	'AA'	1
'Hello World'	'Hello World'	1
'Hello World'	'HelloWorld'	0
'AA'	'aa'	0
'AAA'	'a'	0

在比较时间值时，如果 <操作数 1> 端的时间大于（晚于）或等于 <操作数 2> 端的时间，则 RLO 为“1”。

放置

“大于或等于”操作可以放置在逻辑串中的任何位置。

实例

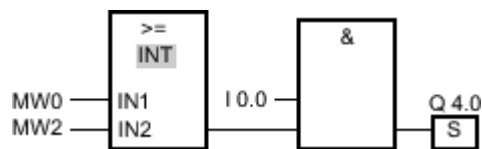


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 比较运算的结果为真 (MW0 >= MW2)。
- 输入 I 0.0 的信号状态为“1”。

参见

[插入 FBD 元素 \(页 447\)](#)

[更改 FBD 元素 \(页 454\)](#)

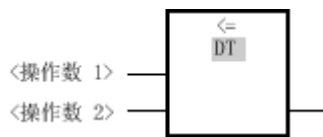
[将操作数插入 FBD 指令 \(页 457\)](#)

[选择 FBD 元素的数据类型 \(页 419\)](#)

[检测存储区域的填充量实例 \(页 570\)](#)

7.8.1.2 CMP <= : 小于或等于

符号



参数	数据类型	存储区	说明
操作数 1	USINT、UINT、UDINT、SINT、INT、DINT、REAL、CHAR、STRING、TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、UDINT、SINT、INT、DINT、REAL、CHAR、STRING、TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“小于或等于”运算确定第一个比较值是否小于或等于第二个比较值。要比较的两个值必须为相同的数据类型。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。如果较长字符串的左侧部分和较短字符串相同，则认为较短字符串更小。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'AA'	'aa'	1
'AAA'	'a'	1
'Hello World'	'Hello World'	1
'HelloWorld'	'Hello World'	0
'BB'	'AA'	0
'AAA'	'AA'	0

在比较时间值时，如果 <操作数 1> 端的时间小于（早于）或等于 <操作数 2> 端的时间，则 RLO 为“1”。

放置

“小于或等于”运算可以放置在逻辑串中的任何位置。

实例

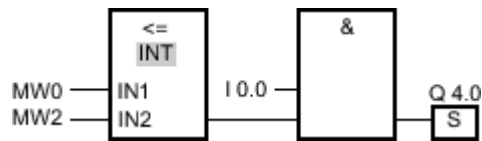


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 比较运算的结果为真 (MW0 <= MW2)。
- 输入 I 0.0 的信号状态为“1”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

选择 FBD 元素的数据类型 (页 419)

7.8.1.2 CMP > : 大于

符号



参数	数据类型	存储区	说明
操作数 1	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“大于”运算查询第一个比较值是否大于第二个比较值。要比较的两个值必须为相同的数据类型。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。如果较长字符串的左侧部分和较短字符串相同，则认为较长字符串更大。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'BB'	'AA'	1
'AAA'	'AA'	1
'AA'	'aa'	0
'AAA'	'a'	0

在比较时间值时，如果 <操作数 1> 端的时间大于（晚于）<操作数 2> 端的时间，则 RLO 为“1”。

放置

“大于”运算可以放置在逻辑串中的任何位置。

实例

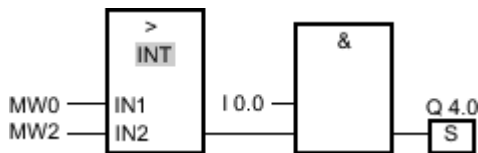


图 7-2

满足下列条件时，将置位输出 Q 4.0：

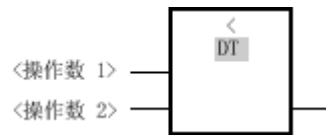
- 比较运算的结果为真 (MW0 > MW2)。
- 输入 I 0.0 的信号状态为“1”。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 CMP < : 小于

符号



参数	数据类型	存储区	说明
操作数 1	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D (页 293) 或常数	要比较的第一个值
操作数 2	USINT、UINT、 UDINT、SINT、INT、 DINT、REAL、 CHAR、STRING、 TIME、DTL	I、Q、M、L、D 或常数	要比较的第二个值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“小于”运算查询第一个比较值是否小于第二个比较值。要比较的两个值必须为相同的数据类型。

在比较字符串时，通过字符 ASCII 码比较字符（例如“a”大于“A”）。从左到右执行比较。第一个不同的字符决定比较结果。如果较长字符串的左侧部分和较短字符串相同，则认为较短字符串更小。

下表显示了字符串比较实例：

<操作数 1>	<操作数 2>	运算的 RLO
'AA'	'aa'	1
'AAA'	'a'	1
'BB'	'AA'	0
'AAA'	'AA'	0

在比较时间值时，如果 <操作数 1> 端的时间小于（早于）<操作数 2> 端的时间，则 RLO 为“1”。

放置

“小于”运算可以放置在逻辑串中的任何位置。

实例

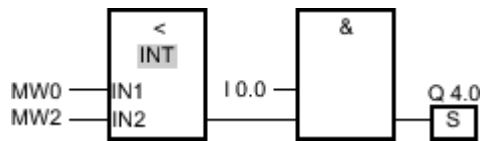


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 比较运算的结果为真 (MW0 < MW2)。
- 输入 I 0.0 的信号状态为“1”。

参见

插入 FBD 元素 (页 447)

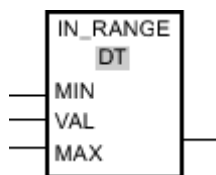
更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

选择 FBD 元素的数据类型 (页 419)

7.8.1.2 IN_RANGE：值在范围内

符号



参数	数据类型	存储区	说明
MIN	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D (页 293) 或常数	取值范围的下限
VAL	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	比较值
MAX	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	取值范围的上限
功能框输出	BOOL	I、Q、M、L、D	比较结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“值在范围内”运算确定输入 VAL 的值是否在指定的取值范围内。使用参数 MIN 和 MAX 指定取值范围的限值。查询时，“值在范围内”运算将输入 VAL 的值与参数 MIN 和 MAX 的值相比较并将结果发送到该功能框输出。如果输入 VAL 的值满足比较 $MIN \leq VAL \leq MAX$ ，则功能框输出的信号状态为“1”。如果该比较结果不为真，则功能框输出的信号状态为“0”。

只有要比较的值具有相同的数据类型且互连了功能框输出时，才能执行该比较功能。

放置

“值在范围内”运算可以放置在逻辑串的始端或中间。

实例

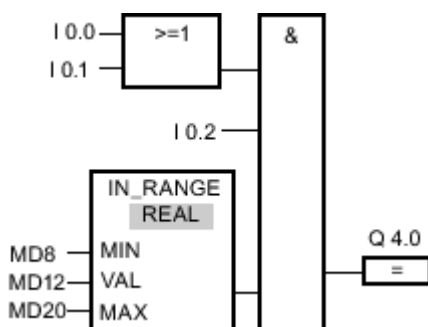


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.0 或 I 0.1 的信号状态为“1”。
- 输入 I 0.2 的信号状态为“1”。
- 输入 MD12 的值在输入 MD8 和 MD20 的当前值所指定的取值范围内 ($MIN \leq VAL \leq MAX$)。

参见

[插入 FBD 元素 \(页 447\)](#)

[更改 FBD 元素 \(页 454\)](#)

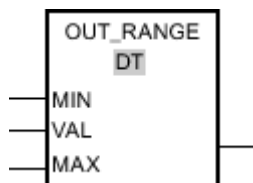
[将操作数插入 FBD 指令 \(页 457\)](#)

[选择 FBD 元素的数据类型 \(页 419\)](#)

[GetError \(页 887\)](#)

7.8.1.2 OUT_RANGE：值超出范围

符号



参数	数据类型	存储区	说明
MIN	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D (页 293) 或常数	取值范围的下限
VAL	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	比较值
MAX	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	取值范围的上限
功能框输出	BOOL	I、Q、M、L、D	比较结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“值超出范围”运算查询输入 VAL 的值是否超出指定范围。使用参数 MIN 和 MAX 指定取值范围的限值。进行查询时，“值超出范围”运算将输入 VAL 的值与参数 MIN 和 MAX 的值相比较并将结果发送到该功能框输出。如果输入 VAL 的值满足比较 $MIN > VAL$ 或 $VAL > MAX$ ，则功能框输出的信号状态为“1”。如果该比较结果不为真，则功能框输出的信号状态为“0”。

只有要比较的值具有相同的数据类型且互连了功能框输出时，才能执行该比较功能。

放置

“值超出范围”运算可以放置在逻辑串的始端或中间。

实例

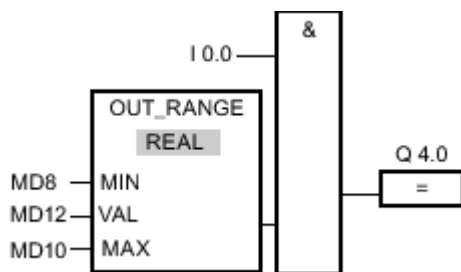


图 7-2

满足下列条件时，将置位输出 Q 4.0：

- 输入 I 0.0 的信号状态为“1”。
- 输入 MD12 的值在由输入 MD8 和 MD10 的值所设定的取值范围之外 ($MIN > VAL$ 或 $VAL > MAX$)。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)

选择 FBD 元素的数据类型 (页 419)

GetError (页 887)

7.8.1.2 OK：测试有效性

符号



参数	数据类型	存储区	说明
<操作数>	REAL	I、Q、M、L、D (页 293) 或常数	要检查的值。

说明

可使用“检查有效性”操作检查某个变量的值 (<操作数>) 是否为有效的浮点数。每个程序周期都进行查询。如果在查询时该变量的值是一个有效浮点数，则右侧连接的输出为信号状态“1”。在任何其它情况下，“检查有效性”操作的该输出的信号状态为“0”。

可以同时使用“检查有效性”操作和 EN 机制。如果将功能框 OK 连接到使能输入 EN，则只有值的有效性查询结果为正时，才会置位使能输入。可以使用此功能来确保只有指定变量的值为有效浮点数时才启用某个操作。

放置

“检查有效性”操作可以放置在逻辑串的始端或中间。

实例

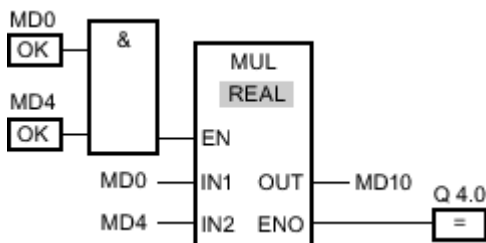


图 7-2

如果 MD0 和 MD4 为有效浮点数，则会执行“乘”运算 (MUL) 并置位输出 ENO。执行“乘”(MUL) 运算时，输入 MD0 的值将与 MD4 的值相乘。相乘的结果存储在输出 MD10 中。如果运算执行过程中未发生错误，则输出 ENO 和 Q 4.0 的信号状态将置为“1”。

参见

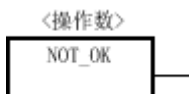
插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)
 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 NOT_OK : 检查无效性

符号



参数	数据类型	存储区	说明
<操作数>	REAL	I、Q、M、L、D (页 293) 或常数	要检查的值。

说明

可使用“检查无效性”运算检查某个变量的值 (<操作数>) 是否为无效的浮点数。每个程序周期都进行查询。如果在查询时该变量的值是一个无效浮点数并且该功能框的输入端的信号状态为“1”，则输出端的信号状态为“1”。在任何其它情况下，“检查无效性”运算的输出端的信号状态为“0”。

放置

“检查无效性”运算可以放置在逻辑串的始端或中间。

实例

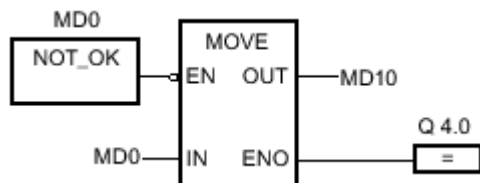


图 7-2

如果输入 MD0 的值为无效浮点数，则不会执行“移动值”(MOVE) 操作。输出 Q 4.0 的信号状态复位为“0”。

参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 将操作数插入 FBD 指令 (页 457)
 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 数学

7.8.1.2 ADD : 加

符号

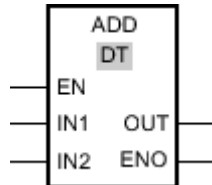


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	被加数
IN2	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	加数
OUT	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D	相加的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可使用“加”运算将输入 IN1 的值与输入 IN2 的值相加并通过输出 OUT 查询总和 ($OUT = IN1 + IN2$)。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 运算结果超出输出 OUT 指定的数据类型的允许范围。
- 输入变量为 REAL 数据类型且无效。

放置

“加”运算可以放置在逻辑串中的任何位置。

实例

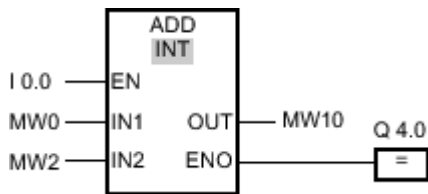


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“加”运算。输入 MW0 的值将与输入 MW2 的值相加。相加结果存储在输出 MW10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 SUB : 减

符号

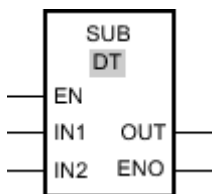


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	被减数
IN2	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	减数。

参数	数据类型	存储区	说明
OUT	SINT、INT、DINT、 USINT、UINT、 UDINT、REAL	I、Q、M、L、D	相减的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可使用“减”运算从输入 IN1 的值中减去输入 IN2 的值并通过输出 OUT 查询差值 ($OUT = IN1 - IN2$)。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 运算结果超出输出 OUT 指定的数据类型的允许范围。
- 输入变量为 REAL 数据类型且无效。

放置

可将“减”运算放置在逻辑串中的任何位置。

实例

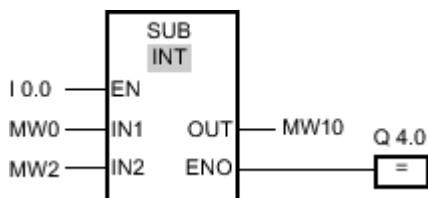


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“减”运算。从输入 MW0 的值中减去输入 MW2 的值。相减结果存储在输出 MW10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

[插入 FBD 元素 \(页 447\)](#)

[更改 FBD 元素 \(页 454\)](#)

[将操作数插入 FBD 指令 \(页 457\)](#)

[选择 FBD 元素的数据类型 \(页 419\)](#)

[粘贴附加输入和输出 \(页 456\)](#)

[计算方程式实例 \(页 573\)](#)

7.8.1.2 MUL : 乘

符号

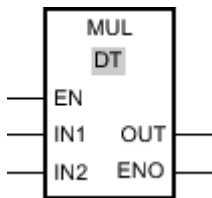


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	被乘数
IN2	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D 或常数	乘数
OUT	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、L、D	乘法结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可使用“乘”运算将输入 IN1 的值与输入 IN2 的值相乘并通过输出 OUT 查询乘积 (OUT =IN1*IN2)。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 结果超出输出 OUT 指定的数据类型的允许范围。
- 输入变量为 REAL 数据类型且无效。

放置

“乘”运算可以放置在逻辑串中的任何位置。

实例

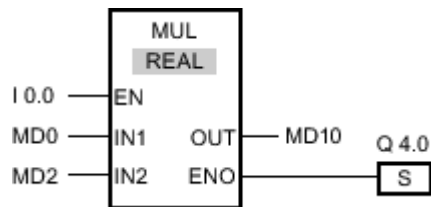


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“乘”运算。用输入 MD2 的值乘以输入 MD0 的值。相乘结果存储在输出 MD10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

粘贴附加输入和输出 (页 456)

计算方程式实例 (页 573)

选择 FBD 元素的数据类型 (页 419)

7.8.1.2 DIV : 除

符号

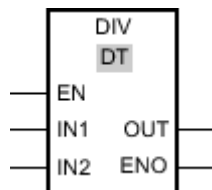


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	被除数
IN2	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	除数

参数	数据类型	存储区	说明
OUT	SINT、USINT、INT、 UINT、DINT、 UDINT、REAL	I、Q、M、L、D	除法结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可使用“除”运算用输入 IN1 的值除以输入 IN2 的值并通过输出 OUT 查询商 (OUT =IN1/IN2)。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 运算结果超出输出 OUT 指定的数据类型的允许范围。
- 输入变量为 REAL 数据类型且无效。

放置

可将“除”运算放置在逻辑串中的任何位置。

实例

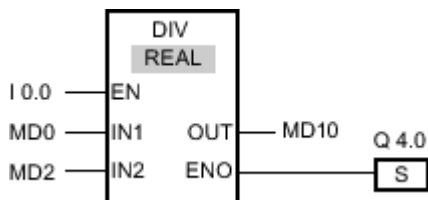


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“除”运算。用输入 MD0 的值除以输入 MD2 的值。相除结果存储在输出 MD10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 MOD : 返回除法的余数

符号

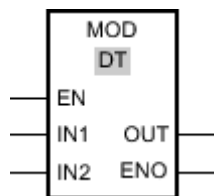


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、INT、DINT、USINT、UINT、UDINT	I、Q、M、L、D 或常数	被除数
IN2	SINT、INT、DINT、USINT、UINT、UDINT	I、Q、M、L、D 或常数	除数
OUT	SINT、INT、DINT、USINT、UINT、UDINT	I、Q、M、L、D	除法的余数

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可使用“返回除法的余数”运算用输入 IN1 的值除以输入 IN2 的值并通过输出 OUT 查询余数。

只有输入 EN 的信号状态为“1”时，才会执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果输入 EN 的信号状态为“0”，则不会执行该运算。在这种情况下，将复位输出 ENO。

放置

“返回除法的余数”运算可以放置在逻辑串中的任何位置。

实例

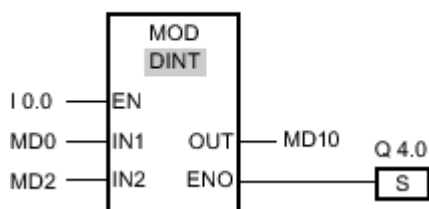


图 7-2

如果输入 I0.0 的信号状态为“0”，则执行“返回除法的余数”运算。用输入 MD0 的值除以输入 MD2 的值。除法的余数放在输出 MD10 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 NEG : 创建补码

符号

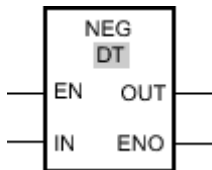


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	SINT、INT、DINT、REAL	I、Q、M、L、D 或常数	输入值
OUT	SINT、INT、DINT、REAL	I、Q、M、L、D	输入值的补码

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“创建补码”操作更改输入 IN 的值符号，并通过输出 OUT 查询结果。例如，如果输入 IN 为正值，则该值的负等效值将发送到输出 OUT。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果操作执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 操作结果超出输出 OUT 指定的数据类型的允许范围。

- 输入变量为 REAL 数据类型且无效。

放置

“创建补码”操作可以放置在逻辑串中的任何位置。

实例

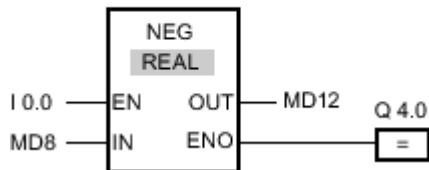


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“创建补码”操作。输入 MD8 的值的符号被更改，且结果放在输出 MD12 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

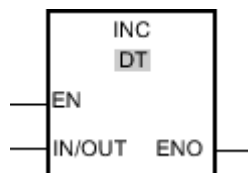
选择 FBD 元素的数据类型 (页 419)

粘贴附加输入和输出 (页 456)

计算方程式实例 (页 573)

7.8.1.2 INC : 递增

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN/OUT	SINT、USINT、INT、UINT、DINT、UDINT	I、Q、M、D、L	要递增的值。

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

使用“递增”运算可以将输出 IN/OUT 的变量的值更改为下一个较高的值并查询结果。只有 EN 使能输入端的信号状态为“1”时，才能启动“递增”运算。如果执行过程中未发生溢出错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则不执行该运算。在这种情况下，将复位使能输出 ENO。

放置

“递增”运算可以放置在逻辑串中的任何位置。

实例

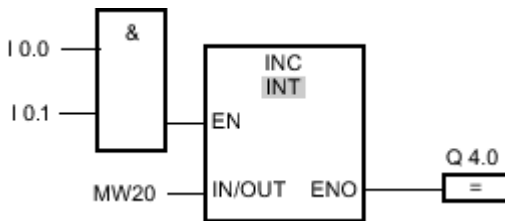


图 7-2

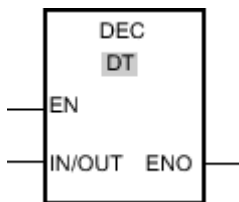
如果输入 I 0.0 和 I 0.1 的信号状态为“1”，则输出 MW20 的值将递增“1”并置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 DEC : 递减

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入

参数	数据类型	存储区	说明
ENO	BOOL	I、Q、M、D、L	使能输出
IN/OUT	SINT、USINT、INT、 UINT、DINT、UDINT	I、Q、M、D、L	要递减的值。

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

使用“递减”运算可以将输出 IN/OUT 的变量值更改为下一个较低的值并查询结果。当使能输入 EN 的信号状态为“1”时，开始执行“递减”运算。如果执行过程中未发生下溢错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则不执行该运算。在这种情况下，将复位使能输出 ENO。

放置

“递减”运算可以放置在逻辑串中的任何位置。

实例

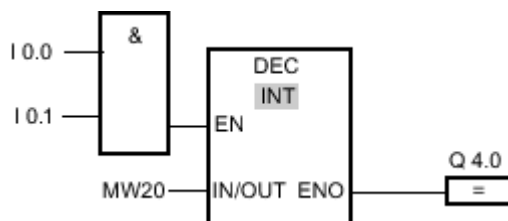


图 7-2

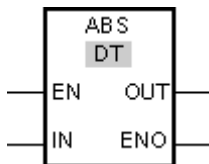
如果输入 I 0.0 和 I 0.1 的信号状态为“1”，则输出 MW20 的值将递减“1”并置位输出 Q 4.0。

参见

[插入 FBD 元素 \(页 447\)](#)
[更改 FBD 元素 \(页 454\)](#)
[选择 FBD 元素的数据类型 \(页 419\)](#)
[将操作数插入 FBD 指令 \(页 457\)](#)
[粘贴附加输入和输出 \(页 456\)](#)
[计算方程式实例 \(页 573\)](#)

7.8.1.2 ABS : 计算绝对值

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	SINT、INT、DINT、REAL	I、Q、M、D、L 或常数	输入值
OUT	SINT、INT、DINT、REAL	I、Q、M、D、L	输入值的绝对值

说明

可以使用“计算绝对值”运算计算输入 IN 所指定值的绝对值。运算结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 指定的 REAL 变量的值不是有效浮点数。

放置

“计算绝对值”运算可以放置在逻辑串中的任何位置。

实例

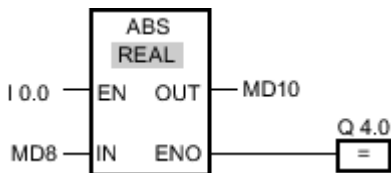


图 7-2

IN	MD8 = -6.234
OUT	MD10 = 6.234

如果输入 I0.0 的信号状态为“1”，则执行“计算绝对值”运算。该运算会计算输入 MD8 的值的绝对值，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

选择 FBD 元素的数据类型 (页 419)

粘贴附加输入和输出 (页 456)

计算方程式实例 (页 573)

7.8.1.2 MIN：提取最小值

符号

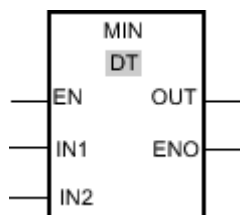


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	第一个输入值
IN2	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	第二个输入值
OUT	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

“提取最小值”操作会比较输入 IN1 的值和输入 IN2 的值，并将较小值写入到输出 OUT。只有为所有参数设置的变量具有相同数据类型时，才会执行该运算。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果在操作执行期间未发生错误，则使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 指定的变量不具有相同的数据类型。
- 输入变量为 REAL 数据类型且无效。

放置

“提取最小值”操作可以放置在逻辑串中的任何位置。

实例

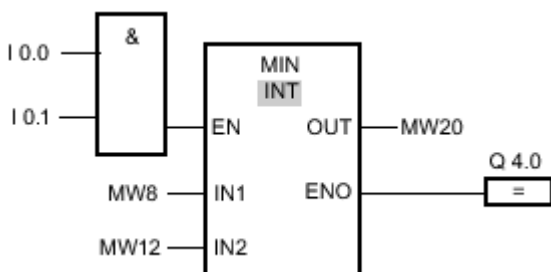


图 7-2

IN1	MW8 = 12666
IN2	MW12 = 14444
OUT	MW20 = 12666

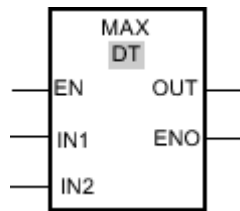
如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则执行“提取最小值”操作。该操作会比较输入 MW8 的值和输入 MW12 的值，并选择较小值 (MW8)。该值被复制到输出 MW20。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 MAX : 提取最大值

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	第一个输入值
IN2	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D 或常数	第二个输入值
OUT	SINT、USINT、INT、UINT、DINT、UDINT、REAL	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

“提取最大值”操作会比较输入 IN1 和 IN2 的值，并将较大值写入到输出 OUT。只有为所有参数设置的变量具有相同数据类型时，才会执行该运算。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果在操作执行期间未发生错误，则使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出的信号状态为“0”：

- 输入 EN 的信号状态为“0”。
- 指定的变量不具有相同的数据类型。
- 输入变量为 REAL 数据类型且无效。

放置

“提取最大值”操作可以放置在逻辑串中的任何位置。

实例

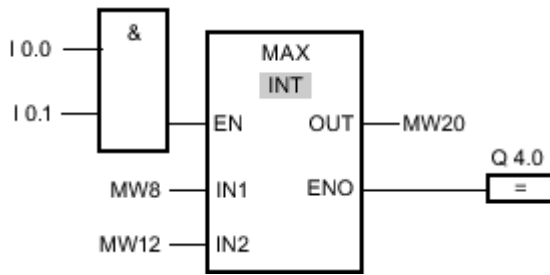


图 7-2

IN1	MW8 = 12666
IN2	MW12 = 14444
OUT	MW20 = 14444

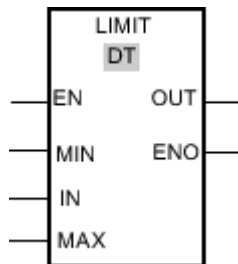
如果输入 I0.0 和 I0.1 的信号状态为“1”，则执行“提取最大值”操作。该操作会比较输入 MW8 和 MW12 的值，并选择较大值 (MW12)。该值被复制到输出 MW20。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 LIMIT : 设置限值

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入

参数	数据类型	存储区	说明
ENO	BOOL	I、Q、M、L、D	使能输出
MIN	SINT、USINT、INT、 UINT、DINT、 UDINT、REAL	I、Q、M、L、D 或常数	下限
IN	SINT、USINT、INT、 UINT、DINT、 UDINT、REAL	I、Q、M、L、D 或常数	输入值
MAX	SINT、USINT、INT、 UINT、DINT、 UDINT、REAL	I、Q、M、L、D 或常数	上限
OUT	SINT、USINT、INT、 UINT、DINT、 UDINT、REAL	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“设置限值”操作，利用输入 MIN 和 MAX 的值来限制输入 IN 的值。如果输入 IN 的值满足 $MIN < IN < MAX$ 的条件，则将该值复制到输出 OUT。如果不满足该条件且输入值低于下限，则将输出设置为输入 MIN 的值。如果超出上限，输出 OUT 将设置为输入 MAX 的值。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果在操作执行期间未发生错误，则使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 指定的变量不具有相同的数据类型。
- 某个输入变量具有无效值。
- 输入 MIN 的值大于输入 MAX 的值。

放置

“设置限值”操作可以放置在逻辑串中的任何位置。

实例

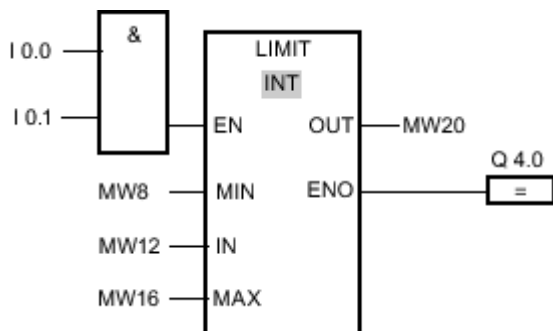


图 7-2

MIN	MW8 = 12000
IN	MW12 = 8000
MAX	MW16 = 16000
OUT	MW20 = 12000

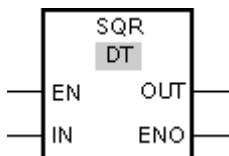
如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则执行“设置限值”操作。将输入 MW12 的值与输入 MW8 和 MW16 的值进行比较。由于输入 MW12 的值小于下限，因此将输入 MW8 的值复制到输出 MW20。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 粘贴附加输入和输出 (页 456)
- 选择 FBD 元素的数据类型 (页 419)
- 计算方程式实例 (页 573)

7.8.1.2 SQR：计算平方

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入

参数	数据类型	存储区	说明
ENO	BOOL	I、Q、M、D、L	使能输出
IN	REAL	I、Q、M、D、L 或常数	输入值
OUT	REAL	I、Q、M、D、L	输入值的平方

说明

可以使用“计算平方”运算计算输入 IN 的值的平方，并通过输出 OUT 查询结果。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算平方”运算可以放置在逻辑串中的任何位置。

实例

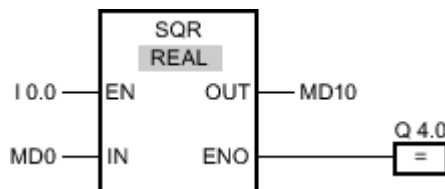


图 7-2

IN	MD0 = 5
OUT	MD10 = 25

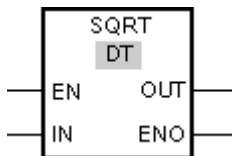
如果输入 I 0.0 的信号状态为“1”，则执行“计算平方”运算。该运算会计算输入 MD0 的值的平方，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

[插入 FBD 元素 \(页 447\)](#)
[更改 FBD 元素 \(页 454\)](#)
[将操作数插入 FBD 指令 \(页 457\)](#)
[选择 FBD 元素的数据类型 \(页 419\)](#)
[粘贴附加输入和输出 \(页 456\)](#)
[计算方程式实例 \(页 573\)](#)

7.8.1.2 Sqrt : 计算平方根

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	REAL	I、Q、M、D、L 或常数	编号
OUT	REAL	I、Q、M、D、L	数字的平方根

说明

可以使用“计算平方根”运算计算输入 IN 的值的平方根，并通过输出 OUT 查询结果。如果输入值大于零，则运算结果为正。如果输入值小于零，则输出 OUT 返回一个无效浮点数。如果输入 IN 的值为“-0”，则结果也是“-0”。

只有使能输入的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。
- 输入 IN 的值为负值。

放置

“计算平方根”运算可以放置在逻辑串中的任何位置。

实例

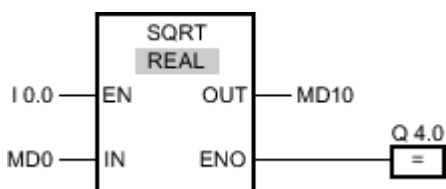


图 7-2

IN	MD0 = 25
----	----------

OUT	MD10 = 5
-----	----------

如果输入 I 0.0 的信号状态为“1”，则执行“计算平方根”运算。该运算会计算输入 MD0 的值的平方根，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

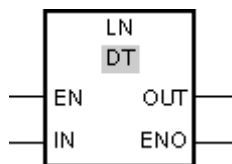
选择 FBD 元素的数据类型 (页 419)

粘贴附加输入和输出 (页 456)

计算方程式实例 (页 573)

7.8.1.2 LN：计算自然对数

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	REAL	I、Q、M、D、L 或常数	输入值
OUT	REAL	I、Q、M、D、L	运算结果

说明

可以使用“计算自然对数”运算计算输入 IN 的自然对数，以 e ($e=2.718282e+00$) 为底。运算结果放在输出 OUT 中，可供查询。如果输入值大于零，则运算结果为正。如果输入值小于零，则输出 OUT 返回一个无效浮点数。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。
- 输入 IN 的值为负值。

放置

“计算自然对数”运算可以放置在逻辑串中的任何位置。

实例

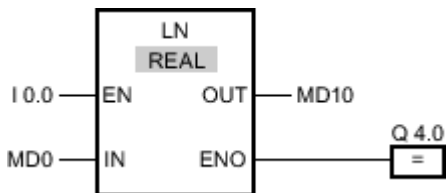


图 7-2

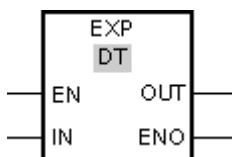
如果输入 I 0.0 的信号状态为“1”，则执行“计算自然对数”运算。该运算会计算输入 MD0 的值的自然对数，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 EXP : 计算指数值

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	REAL	I、Q、M、D、L 或常数	输入值
OUT	REAL	I、Q、M、D、L	输出值

说明

可以使用“计算指数值”运算计算底数为 e (e = 2.718282e+00) 且幂为输入 IN 上设置值的指数值。运算结果放在输出 OUT 中，可供查询 (OUT = e^{IN})。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算指数值”运算可以放置在逻辑串中的任何位置。

实例

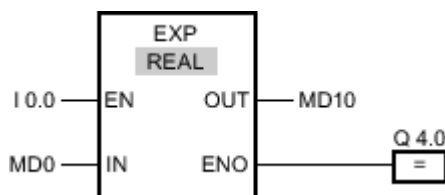


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“计算指数值”运算。该运算会计算以 e 为底且以输入 MD0 的值为幂的指数值，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

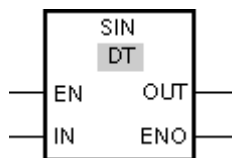
选择 FBD 元素的数据类型 (页 419)

粘贴附加输入和输出 (页 456)

计算方程式实例 (页 573)

7.8.1.2 SIN：计算正弦值

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出

参数	数据类型	存储区	说明
IN	REAL	I、Q、M、L、D 或常数	以弧度为单位的角度值
OUT	REAL	I、Q、M、L、D	指定角度的正弦

说明

可以使用“计算正弦值”运算计算输入 IN (以弧度为单位指定的角度) 的正弦值。运算结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算正弦值”运算可以放置在逻辑串中的任何位置。

实例

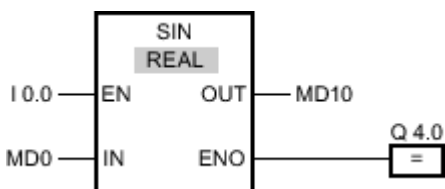


图 7-2

IN	MD0 = +1.570796e+00 ($\pi/2$)
OUT	MD10 = 1

如果输入 I 0.0 的信号状态为“1”，则执行“计算正弦值”运算。此运算会计算在输入 MD0 (指定的角度) 的正弦值，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 选择 FBD 元素的数据类型 (页 419)
- 将操作数插入 FBD 指令 (页 457)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 COS : 计算余弦值

符号

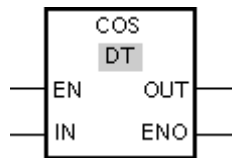


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	以弧度为单位的角度值
OUT	REAL	I、Q、M、L、D	指定角度的余弦

说明

可以使用“计算余弦值”运算计算一个角度的余弦。在输入 IN 指定以弧度为单位的角度值，运算结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算余弦值”运算可以放置在逻辑串中的任何位置。

实例

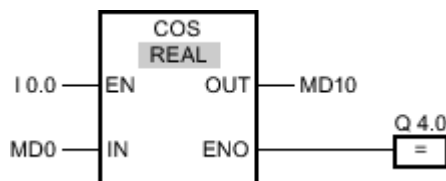


图 7-2

IN	MD0 = +1.570796e+00 ($\pi/2$)
OUT	MD10 = 0

如果输入 I0.0 的信号状态为“1”，则执行“计算余弦值”运算。此运算会计算在输入 MD0 (指定的角度) 的余弦值，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 选择 FBD 元素的数据类型 (页 419)
 将操作数插入 FBD 指令 (页 457)
 粘贴附加输入和输出 (页 456)
 计算方程式实例 (页 573)

7.8.1.2 TAN : 计算正切值

符号

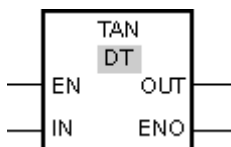


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	以弧度为单位的角度值
OUT	REAL	I、Q、M、L、D	指定角度的正切

说明

可以使用“计算正切值”运算计算一个角度的正切。在输入 IN 指定以弧度为单位的角度值，运算结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算正切值”运算可以放置在逻辑串中的任何位置。

实例

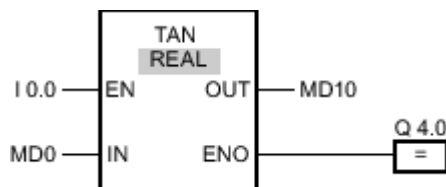


图 7-2

IN	MD0 = +3.141593e++00 (π)
OUT	MD10 = 0

如果输入 I 0.0 的信号状态为“1”，则执行“计算正切值”运算。此运算会计算在输入 MD0 (指定的角度) 的正切值，并将结果发送到输出 MD10。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

选择 FBD 元素的数据类型 (页 419)

将操作数插入 FBD 指令 (页 457)

粘贴附加输入和输出 (页 456)

计算方程式实例 (页 573)

7.8.1.2 ASIN : 计算反正弦值

符号

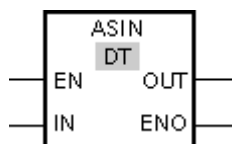


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	正弦值
OUT	REAL	I、Q、M、L、D	以弧度为单位的角度值

说明

可以使用“计算反正弦值”运算根据输入 IN 指定的正弦值计算与该值对应的角度值。只能给输入 IN 指定在范围 -1 到 +1 内的有效浮点数。计算出的角度值以弧度为单位存储在输出 OUT 中，并且该值可以在 $-\pi/2$ 到 $+\pi/2$ 范围内。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。
- 输入 IN 的值超出允许的范围 (-1 到 +1)。

放置

“计算反正弦值”运算可以放置在程序段中的任何位置。

实例

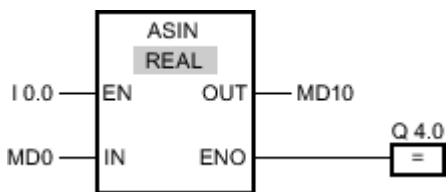


图 7-2

IN	MD0 = 1
OUT	MD10 = +1.570796e+00 ($\pi/2$)

如果输入 I 0.0 的信号状态为“1”，则执行“计算反正弦值”运算。该运算会计算与输入 MD0 的正弦值对应的角度值。运算结果放在输出 MD10 中。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)*
- 更改 FBD 元素 (页 454)*
- 选择 FBD 元素的数据类型 (页 419)*
- 将操作数插入 FBD 指令 (页 457)*
- 粘贴附加输入和输出 (页 456)*
- 计算方程式实例 (页 573)*

7.8.1.2 ACOS : 计算反余弦值

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	余弦值
OUT	REAL	I、Q、M、L、D	以弧度为单位的角度值

说明

可以使用“计算反余弦值”运算通过输入 IN 指定的余弦值计算与该值对应的角度值。只能给输入 IN 指定在范围 -1 到 +1 内的有效浮点数。计算出的角度值以弧度为单位存储在输出 OUT 中，并且该值可以在 0 到 $+\pi$ 范围内。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。
- 输入 IN 的值超出允许的范围 (-1 到 +1)。

放置

“计算反余弦值”运算可以放置在逻辑串中的任何位置。

实例

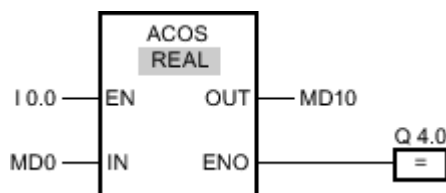


图 7-2

IN	MD0 = 0
----	---------

OUT	MD10 = +1.570796e+00 ($\pi/2$)
-----	----------------------------------

如果输入 I 0.0 的信号状态为“1”，则执行“计算反余弦值”运算。该运算会计算与输入 MD0 的余弦值对应的角度值。运算结果放在输出 MD10 中。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 选择 FBD 元素的数据类型 (页 419)
- 将操作数插入 FBD 指令 (页 457)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 ATAN : 计算反正切值

符号

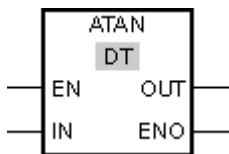


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	正切值
OUT	REAL	I、Q、M、L、D	以弧度为单位的角度值

说明

可以使用“计算反正切值”运算根据输入 IN 指定的正切值计算与该值对应的角度值。在输入 IN 只能指定有效的浮点数。计算出的角度值以弧度为单位放在输出 OUT 中，该值应该在 $-\pi/2$ 到 $+\pi/2$ 范围内。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 IN 的值不是有效浮点数。

放置

“计算反正切值”运算可以放置在逻辑串中的任何位置。

实例

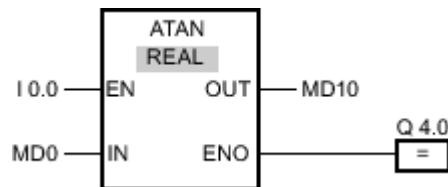


图 7-2

IN	MD0 = 1
OUT	MD10 = +0.785398e++00 ($\pi/4$)

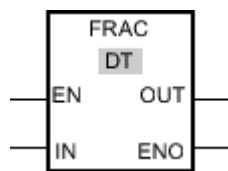
如果输入 I 0.0 的信号状态为“1”，则执行“计算反正切值”运算。该运算会计算与输入 MD0 的正切值对应的角度值。运算结果放在输出 MD10 中。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 选择 FBD 元素的数据类型 (页 419)
 将操作数插入 FBD 指令 (页 457)
 粘贴附加输入和输出 (页 456)
 计算方程式实例 (页 573)

7.8.1.2 FRAC：提取小数

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	REAL	I、Q、M、L、D 或常数	要确定其小数位的值。

参数	数据类型	存储区	说明
OUT	REAL	I、Q、M、L、D	输入 IN 的 值的小数位

说明

可以使用“提取小数”运算获取输入 IN 的 值的小数位。查询的结果存储在输出 OUT 中，可供查询。例如，如果输入 IN 的 值为 123.4567，则输出 OUT 返回值 0.4567。输入 EN 的信号状态为“1”时，将启动该运算。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理该运算期间发生错误，例如，输入端没有有效浮点数。

放置

“提取小数”运算可以放置在逻辑串中的任何位置。

实例

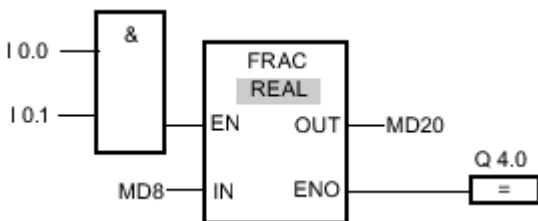


图 7-2

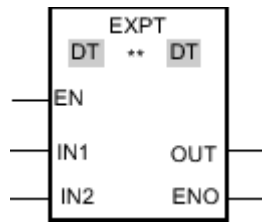
如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则启动“提取小数”运算。输入 MD8 的 值的小数位会复制到输出 MD20。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 选择 FBD 元素的数据类型 (页 419)
- 将操作数插入 FBD 指令 (页 457)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 EXPT : 取幂

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	REAL	I、Q、M、L、D 或常数	底数值
IN2	REAL、INT、UINT、USINT、SINT、DINT、UDINT	I、Q、M、L、D 或常数	对底数进行幂运算所用的值
OUT	REAL	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“取幂”运算使用输入 IN2 指定的值为幂对输入 IN1 的值进行幂运算。运算结果放在输出 OUT 中，可供查询 ($OUT = IN1^{IN2}$)。

输入 IN1 的值必须是有效浮点数。还允许使用整数设置输入 IN2。

执行“取幂”运算时要求使能输入 EN 的信号状态为“1”。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理该运算期间发生错误，例如，发生溢出。

放置

“取幂”运算可以放置在逻辑串中的任何位置。

实例

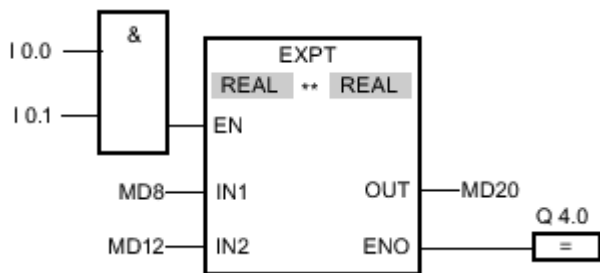


图 7-2

如果输入 I0.0 与 I0.1 的信号状态为“1”，则执行“取幂”运算。输入 MD8 的值为使用输入 MD12 指定的值为幂进行幂运算。结果存储在输出 MD20 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 选择 FBD 元素的数据类型 (页 419)
- 将操作数插入 FBD 指令 (页 457)
- 粘贴附加输入和输出 (页 456)
- 计算方程式实例 (页 573)

7.8.1.2 移动

7.8.1.2 MOVE : 复制值

符号

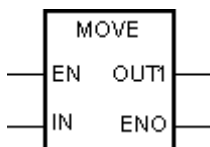


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	所有基本数据类型、DTL、STRUCT、ARRAY	I、Q、M、D、L 或常数	源值

参数	数据类型	存储区	说明
OUT1	所有基本数据类型、DTL、STRUCT、ARRAY	I、Q、M、D、L	目标地址

说明

可以使用“移动值”操作将输入 IN 的操作数的数据传送到输出 OUT1 的操作数。始终沿地址升序方向进行传送。

如果启用 IEC 测试，则输入 IN 和输出 OUT1 的变量必须具有相同的数据类型。如果未启用 IEC 测试，则该操作的输入 IN 和输出 OUT1 的操作数宽度必须相同。

只有当输入 IN 和输出 OUT1 的变量的数组元素具有相同数据类型时，才能复制整个数组。

对于“移动值”操作，可插入附加输出。在这种情况下，输入 IN 的操作数的数据将传送到所有可用的输出。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。在这种情况下，输出 ENO 的信号状态也为“1”。

如果输入 EN 的信号状态为“0”，则使能输出 ENO 将复位为“0”。

“移动块”(MOVE_BLK) 和“移动块不可中断”(UMOVE_BLK) 操作也可用于复制 ARRAY 数据类型的变量。可以使用 S_CONV 操作复制 STRING 数据类型的变量。

放置

“移动值”操作可以放置在逻辑串中的任何位置。

实例

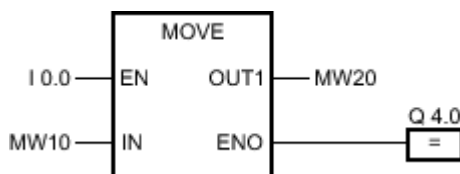


图 7-2

IN	MW10 = 0011 1111 1010 1111
OUT1	MW20 = 0011 1111 1010 1111

如果输入 I 0.0 的信号状态为“1”，则执行“移动值”操作。该操作将输入操作数 (MW10) 的数据复制到输出操作数 (MW20) 并将输出 Q 4.0 的信号状态置位为“1”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

粘贴附加输入和输出 (页 456)

7.8.1.2 MOVE_BLK : 复制区域

符号

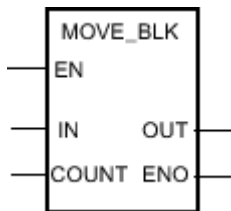


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	ARRAY	D、L	要复制的源区域的第一个元素。
COUNT	UINT	I、Q、M、D、L 或常数	要从源区域复制到目标区域的元素个数。
OUT	ARRAY	D、L	源区域数据将复制到的目标区域的第一个元素。

说明

可以使用“移动块”操作将一个存储区（源区域）的数据复制到另一个存储区（目标区域）。要复制到目标区域的元素个数由参数 COUNT 指定。要复制的元素的宽度由输入 IN 的元素宽度定义。复制操作沿地址升序方向进行。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 复制的数据量超出输出端 OUT 存储区所能容纳的数据量。

放置

“移动块”操作可以放置在逻辑串中的任何位置。

实例

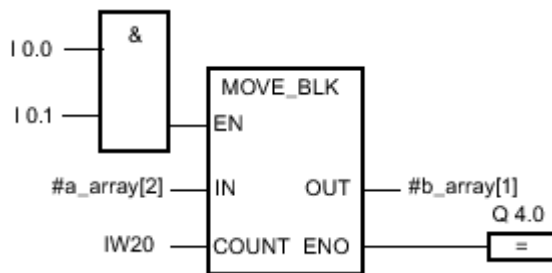


图 7-2

IN	变量“a_array”是 ARRAY 数据类型，由 5 个 INT 数据类型的元素组成。
COUNT	IW20 = 3
OUT	变量“b_array”是 ARRAY 数据类型，由 6 个 INT 数据类型的元素组成。

如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则执行“移动块”操作。该操作从变量“a_array”选择三个 INT 元素 (a_array[2..4]) 并将其复制到输出变量“b_array”(b_array[1..3])。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 将操作数插入 FBD 指令 (页 457)
 粘贴附加输入和输出 (页 456)

7.8.1.2 UMOVE_BLK：复制区域不可中断

符号

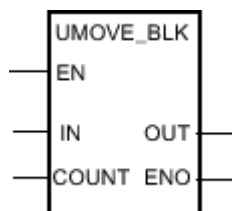


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入

参数	数据类型	存储区	说明
ENO	BOOL	I、Q、M、D、L	使能输出
IN	ARRAY	D、L	要复制的源区域的第一个元素。
COUNT	UINT	I、Q、M、D、L 或常数	要从源区域复制到目标区域的元素个数。
OUT	ARRAY	D、L	源区域数据将复制到的目标区域的第一个元素。

说明

可以使用“移动块不可中断”操作将一个存储区（源区域）的数据不中断地复制到另一个存储区（目标区域）。要复制到目标区域的元素个数由参数 COUNT 指定。要复制的元素的宽度由输入 IN 的元素宽度定义。

源区域数据沿地址升序方向复制到目标区域。此复制操作不能被其它操作系统任务打断。这就是执行“移动块不可中断”操作期间 CPU 的报警响应时间增加的原因。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 复制的数据量超出输出端 OUT 所能容纳的数据量。

放置

“移动块不可中断”操作可以放置在逻辑串中的任何位置。

实例：使用复杂数据类型

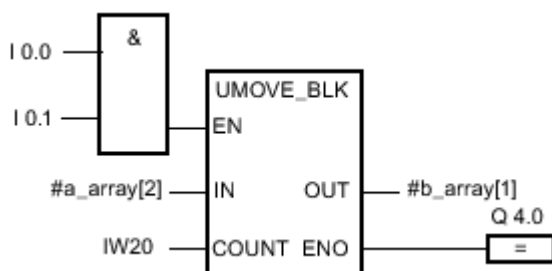


图 7-2

IN	变量“a_array”是 ARRAY 数据类型，由 5 个 INT 数据类型的元素组成。
COUNT	IW2 = 3
OUT	变量“b_array”是 ARRAY 数据类型，由 6 个 INT 数据类型的元素组成。

如果输入 I0.0 与 I0.1 的信号状态为“1”，则执行“移动块”操作。该操作从变量“a_array”选择三个 INT 元素 (a_array[2..4]) 并将其复制到输出变量“b_array”(b_array[1..3])。此复制操作不能被其它操作系统任务打断。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)
更改 FBD 元素 (页 454)
将操作数插入 FBD 指令 (页 457)
粘贴附加输入和输出 (页 456)

7.8.1.2 FILL_BLK：填充区域

符号

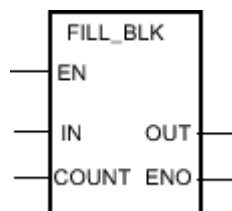


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	ARRAY	D、L 或常数	用于填充目标区域的元素。
COUNT	UINT	I、Q、M、D、L 或常数	重复的复制操作次数
OUT	ARRAY	D、L	目标区域中开始填充的地址。

说明

可以通过“填充块”操作使用输入 IN 的值填充存储区 (目标区域)。从输出 OUT 指定的地址开始填充目标区域。重复的复制操作次数由参数 COUNT 指定。执行该操作时，将选择输入 IN 的值并将其按照参数 COUNT 指定的重复次数复制到目标区域。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。

- 复制的数据量超出输出端 OUT 存储区所能容纳的数据量。

放置

“填充块”操作可以放置在逻辑串中的任何位置。

实例

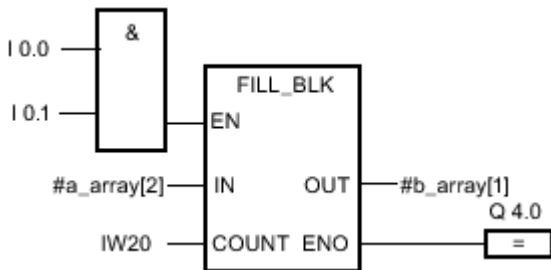


图 7-2

IN	变量“a_array”是 ARRAY 数据类型，由 4 个 WORD 数据类型的元素组成 (WORD 数据类型的 ARRAY[1..4])。
COUNT	IW20=3
OUT	变量“b_array”是 ARRAY 数据类型，由 5 个 WORD 数据类型的元素组成 (WORD 数据类型的 ARRAY[1..5])。

如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则执行“填充块”操作。该操作将变量“a_array”的第二个元素 (a_array[2]) 复制到输出变量“b_array”(b_array[1..3]) 三次。如果操作执行过程中未发生错误，则输出 ENO 和 Q 4.0 的信号状态设置为“1”。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 粘贴附加输入和输出 (页 456)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 UFILL_BLK：填充区域不可中断

符号

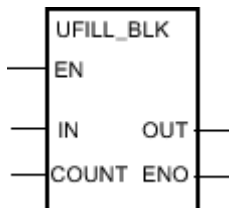


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	ARRAY	D、L 或常数	用于填充目标区域的元素。
COUNT	UINT	I、Q、M、D、L 或常数	重复的复制操作次数
OUT	ARRAY	D、L	目标区域中开始填充的地址。

说明

可以通过“填充块不可中断”操作使用输入 IN 的值不中断地填充存储区（目标区域）。从输出 OUT 指定的地址开始填充目标区域。重复的复制操作次数由参数 COUNT 指定。执行该操作时，将选择输入 IN 的值并将其按照参数 COUNT 指定的重复次数复制到目标区域。

此复制操作不能被其它操作系统任务打断。这就是执行“填充块不可中断”操作期间 CPU 的报警响应时间增加的原因。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果运算执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 复制的数据量超出输出端 OUT 存储区所能容纳的数据量。

放置

“填充块不可中断”操作可以放置在逻辑串中的任何位置。

实例

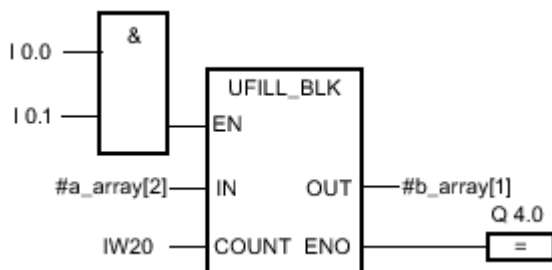


图 7-2

IN	变量“a_array”是 ARRAY 数据类型，由 4 个 WORD 数据类型的元素组成（WORD 数据类型的 ARRAY[1..4]）。
COUNT	IW20=3

OUT	变量“b_array”是 ARRAY 数据类型，由 5 个 WORD 数据类型的元素组成 (WORD 数据类型的 ARRAY[1..5])。
-----	--

如果输入 I 0.0 与 I 0.1 的信号状态为“1”，则执行“填充块”操作。该操作将变量“a_array”的第二个元素 (a_array[2]) 复制到输出变量“b_array”(b_array[1..3]) 三次。此复制操作不能被其它操作系统任务打断。如果操作执行过程中未发生错误，则输出 ENO 和 Q 4.0 的信号状态设置为“1”。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 粘贴附加输入和输出 (页 456)

7.8.1.2 SWAP : 交换

符号

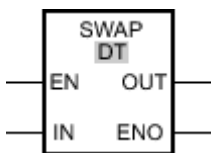


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	WORD、DWORD	I、Q、M、L、D 或常数	要交换其字节的操作数。
OUT	WORD、DWORD	I、Q、M、L、D	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“交换”操作更改输入 IN 变量内的字节顺序，并通过输出 OUT 查询结果。

下图显示了如何使用“交换”操作交换 DWORD 数据类型变量的字节：

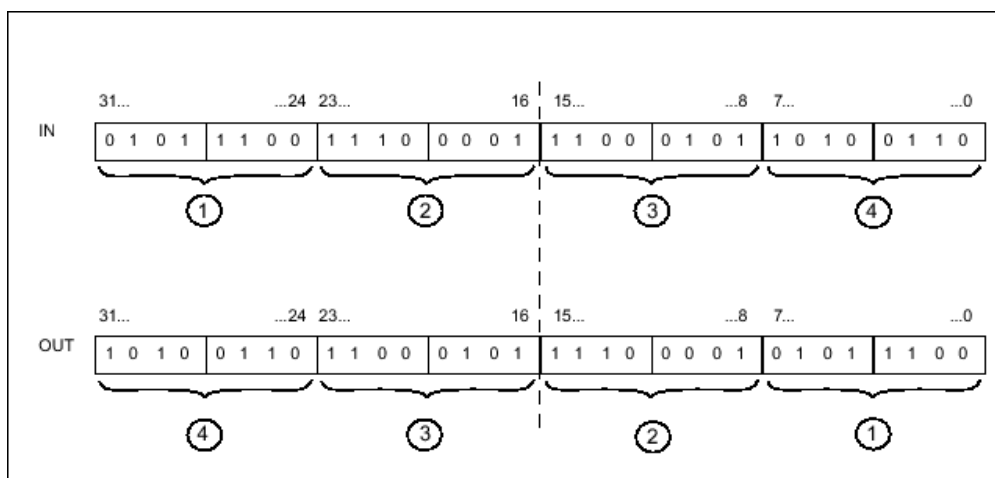


图 7-2

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果操作执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果输入 EN 的信号状态为“0”，则使能输出 ENO 将复位为“0”。

放置

“交换”操作可以放置在逻辑串中的任何位置。

实例

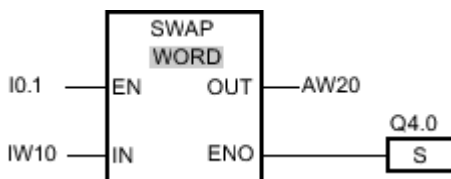


图 7-2

IN	IW10 = 0000 1111 0101 0101
OUT	QW20 = 0101 0101 0000 1111

如果输入 I 0.1 的信号状态为“1”，则执行“交换”操作。字节的排列被更改并存储在变量 QW20 中。如果操作执行过程中未发生错误，则输出 ENO 和 Q 4.0 的信号状态设置为“1”。

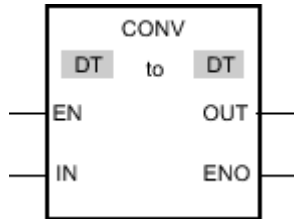
参见

插入 FBD 元素 (页 447)
 更改 FBD 元素 (页 454)
 选择 FBD 元素的数据类型 (页 419)
 将操作数插入 FBD 指令 (页 457)
 粘贴附加输入和输出 (页 456)

7.8.1.2 转换

7.8.1.2 CONVERT : 转换值

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	BYTE、WORD、 DWORD、SINT、 INT、DINT、USINT、 UINT、UDINT、 BCD16、BCD32、 REAL	I、Q、M、D、L 或常数	要转换的值。
OUT	BYTE、WORD、 DWORD、SINT、 INT、DINT、USINT、 UINT、UDINT、 BCD16、BCD32、 REAL	I、Q、M、D、L	转换结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

“转换”操作读取参数 IN 的数据，然后根据指定的数据类型对其进行转换。

只有使能输入 EN 的信号状态为“1”时，才能启动“转换”操作。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“转换”操作可以放置在逻辑串中的任何位置。

实例

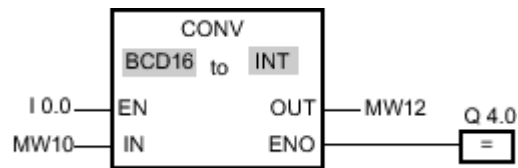


图 7-2

如果输入 I 0.0 = 1，则会以三位 BCD 码数字的形式读取 MW10 的数据并将其转换为整数（16 位）。结果存储在 MW12 中。如果执行转换（ENO = EN = 1），则输出 Q 4.0 为“1”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

选择 FBD 元素的数据类型 (页 419)

7.8.1.2 ROUND：舍入为双整型

符号

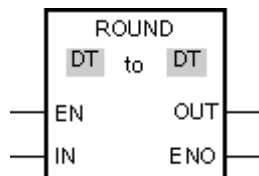


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	REAL	I、Q、M、D、L 或常数	要取整的输入值。
OUT	INT、USINT、UINT、 SINT、UDINT、 DINT、REAL	I、Q、M、D、L	输出值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“取整数字值”操作将输入 IN 的值取整为最接近的整数。该操作将输入 IN 的值解释为浮点数并将其转换为最接近的双整数。如果输入值恰好是相邻偶数和奇数的平均数，则选择偶数。操作结果放在输出 OUT 中，可供查询。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果操作执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“取整数字值”操作可以放置在逻辑串中的任何位置。

实例

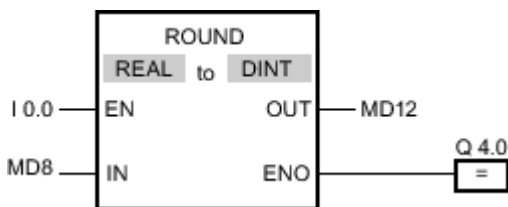


图 7-2

IN	MD8 = 0.50000000
OUT	MD12 = 0

如果输入 I 0.0 的信号状态为“1”，则执行“取整数字值”操作。输入 MD8 的浮点数将取整到最近的偶数双整数并发送到输出 OUT。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 CEIL：浮点数向上取整

符号

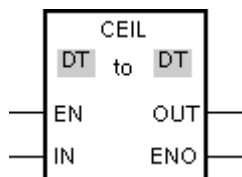


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	REAL	I、Q、M、D、L 或常数	输入值
OUT	INT、USINT、UINT、SINT、UDINT、DINT、REAL	I、Q、M、D、L	输出值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“浮点数向上取整”操作将输入 IN 的值向上取整为相邻整数。该操作将输入 IN 的值解释为浮点数并将其向上转换为相邻的整数。运算结果放在输出 OUT 中，可供查询。输出值可以大于或等于输入值。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果操作执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“浮点数向上取整”操作可以放置在逻辑串中的任何位置。

实例

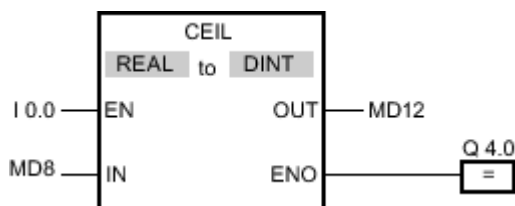


图 7-2

IN	MD8 = 0.50000000
OUT	MD12 = 1

如果输入 I 0.0 的信号状态为“1”，则执行“浮点数向上取整”操作。输入 MD8 的浮点数将向上取整为相邻整数并发送到输出 MD12。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 选择 FBD 元素的数据类型 (页 419)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 FLOOR : 浮点数向下取整

符号

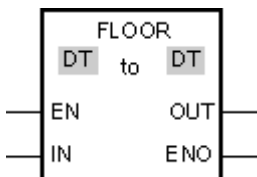


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	REAL	I、Q、M、D、L 或常数	输入值
OUT	INT、USINT、UINT、SINT、UDINT、DINT、REAL	I、Q、M、D、L	输出值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“浮点数向下取整”操作将输入 IN 的值向下取整为相邻整数。该操作将输入 IN 的值解释为浮点数并将其向下转换为相邻的整数。运算结果放在输出 OUT 中，可供查询。输出值可以小于或等于输入值。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果操作执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“浮点数向下取整”操作可以放置在逻辑串中的任何位置。

实例

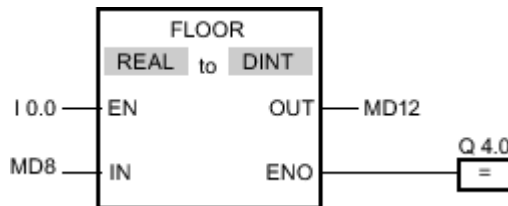


图 7-2

IN	MD8 = 0.50000000
OUT	MD12 = 0

如果输入 I 0.0 的信号状态为“1”，则执行“浮点数向下取整”操作。输入 MD8 的浮点数将向下取整为相邻整数并发送到输出 MD12。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

选择 FBD 元素的数据类型 (页 419)

插入 FBD 元素 (页 447)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 TRUNC：截取整数

符号

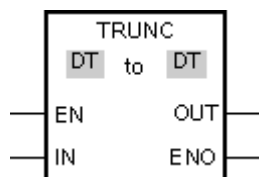


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出

参数	数据类型	存储区	说明
IN	REAL	I、Q、M、D、L 或常数	输入值
OUT	INT、USINT、UINT、 SINT、UDINT、 DINT、REAL	I、Q、M、D、L	输入值的整数部分

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“截取整数”操作得出整数而不对输入 IN 的值进行舍入。输入 IN 的值被视为浮点数。该操作仅选择浮点数的整数部分，并将其发送到输出 OUT 中，不带小数位。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果操作执行过程中未发生错误，则输出 ENO 的信号状态为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 处理过程中发生溢出之类的错误。

放置

“截取整数”操作可以放置在逻辑串中的任何位置。

实例

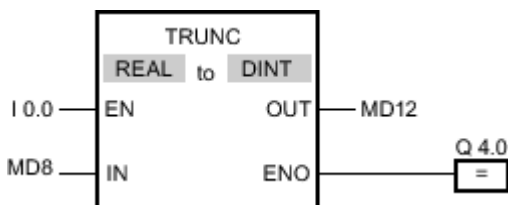


图 7-2

IN	MD8 = 0.50000000
OUT	MD12 = 0

如果输入 I 0.0 的信号状态为“1”，则执行“截取整数”操作。输入 MD8 浮点数的整数部分转换为整数并发送到输出 MD12。如果运算执行过程中未发生错误，则置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 SCALE_X：缩放

符号

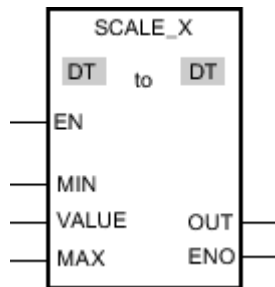


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
VALUE	REAL	I、Q、M、D、L 或常数	要标定的值。
MIN	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	取值范围的下限
MAX	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	取值范围的上限
OUT	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L	标定的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“标定”操作通过将输入 VALUE 的值映射到指定的取值范围对该值进行标定。执行“标定”操作时，会将输入 VALUE 的浮点数值标定到由参数 MIN 和 MAX 定义的取值范围。标定结果为整数，并存储在输出 OUT 中。

下图显示了如何标定值的实例：

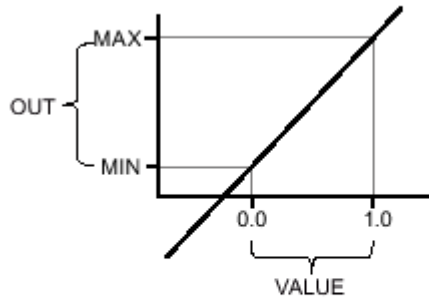


图 7-2

只有使能输入 EN 的信号状态为“1”时，才执行“标定”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 MIN 的值大于或等于输入 MAX 的值。
- 根据 IEEE-754 标准，指定的 REAL 变量的值超出标准化数的范围。
- 发生溢出。
- 输入 VALUE 的值为 NaN (无效算术运算的结果)。

放置

“标定”操作可以放置在逻辑串中的任何位置。

实例

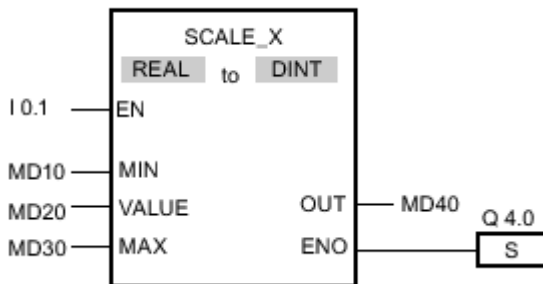


图 7-2

VALUE	MD20 = 0.5
-------	------------

MIN	MD10 = 10
MAX	MD30 = 30
OUT	MD40 = 20

如果输入 I 0.1 的信号状态为“1”，则执行“标定”操作。输入 MD20 的值标定到输入 MD10 和 MD30 的值所定义的取值范围。结果存储在输出 MD40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

选择 FBD 元素的数据类型 (页 419)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 NORM_X：标准化

符号

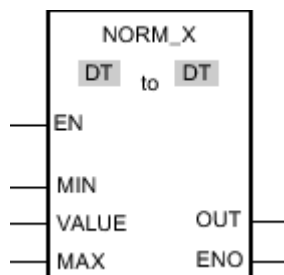


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
VALUE	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	要标准化的值。
MIN	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	取值范围的下限
MAX	SINT、INT、DINT、USINT、UINT、UDINT、REAL	I、Q、M、D、L 或常数	取值范围的上限
OUT	REAL	I、Q、M、D、L	标准化结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“标准化”操作通过将输入 VALUE 的变量值映射到线性标尺对其进行标准化。可以使用参数 MIN 和 MAX 定义（应用于该标尺的）取值范围的限值。根据标准化值在该取值范围内的位置，计算输出 OUT 的结果并以浮点数形式存储计算结果。如果要标准化的值等于输入 MIN 的值，输出 OUT 将返回值“0.0”。如果要标准化的值等于输入 MAX 的值，输出 OUT 将返回值“1.0”。

下图显示了如何标准化值的实例：

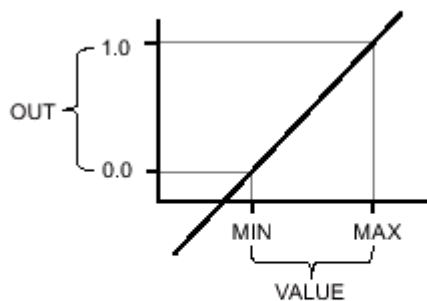


图 7-2

只有使能输入 EN 的信号状态为“1”时，才执行“标准化”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，使能输出 ENO 将返回信号状态“0”：

- 输入 EN 的信号状态为“0”。
- 输入 MIN 的值大于或等于输入 MAX 的值。
- 根据 IEEE-754 标准，指定的 REAL 变量的值超出标准化数的范围。
- 发生溢出。
- 输入 VALUE 的值为 NaN（无效算术运算的结果）。

放置

“标准化”操作可以放置在逻辑串中的任何位置。

实例

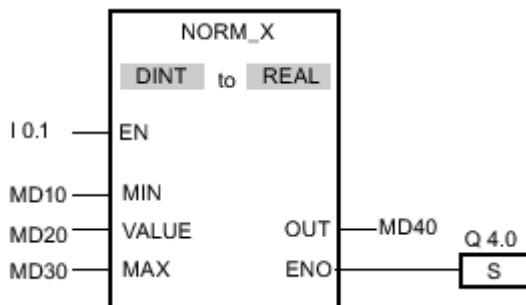


图 7-2

VALUE	MD20 = 20
MIN	MD10 = 10
MAX	MD30 = 30
OUT	MD40 = 0.5

如果输入 I 0.1 的信号状态为“1”，则执行“标准化”操作。MD20 输入端的值在输入 MD10 和 MD30 的值所定义的取值范围内。根据定义的取值范围标准化输入 MD20 的变量值。结果以浮点数形式存储在输出 MD40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

选择 FBD 元素的数据类型 (页 419)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 程序控制

7.8.1.2 JMP：为 1 时块中跳转（有条件）

符号



图 7-2

说明

可以使用“为 1 时块中跳转（有条件）”操作中断程序的线性执行并在另一个程序段中继续执行。目标程序段必须标有跳转标签。需要指定该跳转标签的名称以便执行该操作。指定的跳转标签位于该操作的上方。

指定的跳转标签必须在执行该操作的块中。指定的名称在块中只能出现一次。

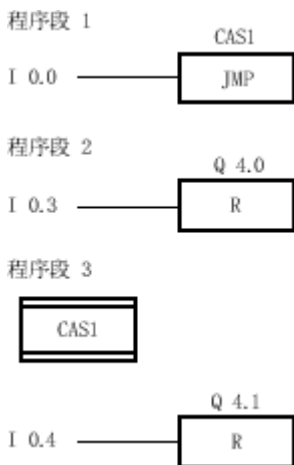
如果该操作的输入端的逻辑运算结果 (RLO) 为“1”，则跳转到跳转标签所标识的程序段。跳转方向可以朝向更大或更小的程序段编号。

如果不满足该操作的输入端的条件 (RLO = 0)，程序则从下一个程序段继续执行。

放置

“为 1 时块中跳转 (有条件)”操作只能放置在程序段的右侧。

实例



如果输入 I 0.0 的信号状态为“1”，则执行“为 1 时块中跳转 (有条件)”操作。程序的线性执行被中断并在跳转标签 CAS1 标识的“程序段 3”中继续执行。如果输入 I 0.4 的信号状态为“1”，则复位输出 Q 4.1。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 JMPN : 为 0 时块中跳转 (有条件)

符号



图 7-2

说明

可以使用“为 0 时块中跳转 (有条件)”操作，在该操作的输入端的逻辑运算结果为“0”时中断程序的线性执行并在另一个程序段中继续执行。目标程序段必须标有跳转标签。需要指定该跳转标签的名称以便执行该操作。指定的跳转标签位于该操作的上方。

指定的跳转标签必须在执行该操作的块中。指定的名称在块中只能出现一次。

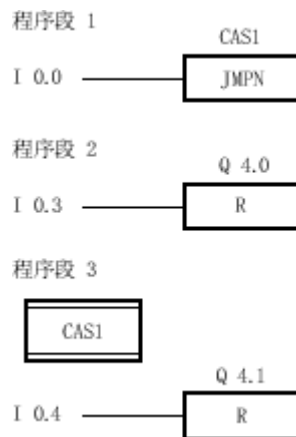
如果该操作的输入端的逻辑运算结果 (RLO) 为“0”，则跳转到跳转标签所标识的程序段。跳转方向可以朝向更大或更小的程序段编号。

如果不满足该操作的输入条件 (RLO = 1)，程序则从下一个程序段继续执行。

放置

“为 0 时块中跳转 (有条件)”操作需要一个前导逻辑操作，只能放置在逻辑串的末端。

实例



如果输入 I 0.0 的信号状态为“0”，则执行“为 0 时块中跳转 (有条件)”操作。程序的线性执行被中断并在跳转标签 CAS1 标识的“程序段 3”中继续执行。如果输入 I 0.4 的信号状态为“1”，则复位输出 Q 4.1。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 LABEL : 跳转标签

符号



图 7-2

说明

可以使用“跳转标签”指定跳转的目标程序段，程序应在目标程序段中继续执行。跳转标签的名称可以包含字母、数字或下划线。

跳转标签以及其中指定了该跳转标签的操作必须位于同一个块中。跳转标签的名称在块中只能分配一次。

每个跳转标签可以跳转到多个位置。

实例

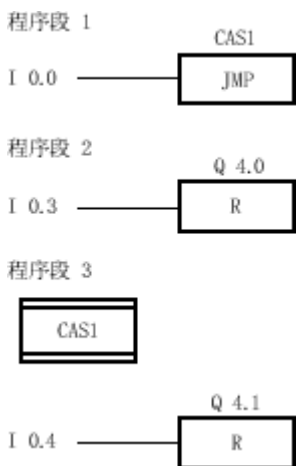


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“为 1 时块中跳转 (有条件)”操作。程序的线性执行被中断并在跳转标签 CAS1 标识的“程序段 3”中继续执行。如果输入 I 0.4 的信号状态为“1”，则复位输出 Q 4.1。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 RET : 返回

符号



图 7-2

参数值	数据类型	存储区	说明
TRUE	-	-	调用功能的状态设置为“1”。
FALSE	-	-	调用功能的状态设置为“0”。
<操作数>	BOOL	I、Q、M、L、D (页 293)	调用功能的状态设置为指定操作数的信号状态。

说明

可以使用“返回”操作停止块的执行。只有左侧接头的信号状态为“1”时，才会执行该操作。如果满足该条件，将在当前被调用块中终止程序执行并在调用块（例如，调用 OB）中的调用功能完成后继续执行。调用功能的状态由“返回”操作的参数决定。该参数可以为以下值：

- TRUE：调用功能的输出 ENO 置位为“1”。
- FALSE：调用功能的输出 ENO 复位为“0”。
- <操作数>：调用功能的输出 ENO 由指定操作数的信号状态决定。

如果组织块被“返回”操作终止，则 CPU 继续执行系统程序。

如果“返回”操作输入端的信号状态为“0”，则不执行该操作。在这种情况下，程序继续在被调用块的下一个程序段中执行。

放置

“返回”操作可以放置在逻辑串的始端或末端。

实例

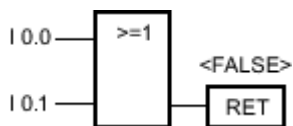


图 7-2

如果输入 I0.0 或 I0.1 的信号状态为“0”，则执行“返回”操作。被调用块中的程序执行被终止，并继续在被调用块中执行。调用功能的输出 ENO 的信号状态复位为“0”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 逻辑运算

7.8.1.2 AND：AND 逻辑运算

符号

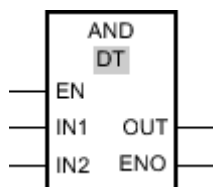


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	逻辑运算的第一个值
IN2	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	逻辑运算的第二个值
OUT	BYTE、WORD、DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“AND 逻辑运算”将输入 IN1 的值与输入 IN2 的值通过 AND 逻辑逐位运算，并通过输出 OUT 查询结果。

执行该运算时，输入 IN1 的值的位 0 与输入 IN2 的值的位 0 通过 AND 逻辑进行运算。结果存储在输出 OUT 的位 0 中。对指定值的所有其它位都执行相同的逻辑运算。

只有该逻辑运算中的两个位的信号状态均为“1”时，结果位的信号状态才为“1”。如果该逻辑运算的两个位中有一个位的信号状态为“0”，则对应的结果位将复位。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。在这种情况下，输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“AND 逻辑运算”可以放置在逻辑串中的任何位置。

实例

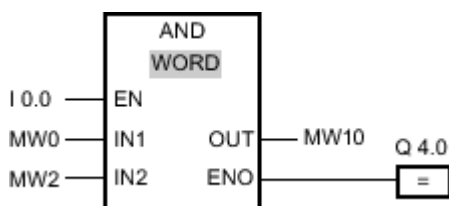


图 7-2

IN1	MW0 = 01010101 01010101
IN2	MW2 = 00000000 00001111
OUT	MW10= 00000000 0000101

如果输入 I0.0 的信号状态为“1”，则执行“AND 逻辑运算”。输入 MW0 的值与输入 MW2 的值通过 AND 逻辑进行运算。逐位运算得出结果并发送到输出 MW10 中。输出 ENO 和输出 Q4.0 的信号状态置位为“1”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

选择 FBD 元素的数据类型 (页 419)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 OR : OR 逻辑运算

符号

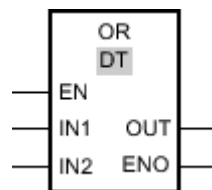


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	逻辑运算的第一个值
IN2	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	逻辑运算的第二个值
OUT	BYTE、WORD、DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“OR 逻辑运算”将输入 IN1 的值与输入 IN2 的值通过 OR 逻辑逐位运算，并通过输出 OUT 查询结果。

执行该运算时，输入 IN1 的值的位 0 与输入 IN2 的值的位 0 通过 OR 逻辑进行运算。结果存储在输出 OUT 的位 0 中。对指定变量的所有位都执行相同的逻辑运算。

只要该逻辑运算中的两个位中至少有一个位的信号状态为“1”，结果位的信号状态就为“1”。如果该逻辑运算的两个位的信号状态均为“0”，则对应的结果位将复位。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。在这种情况下，输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“OR 逻辑运算”可以放置在逻辑串中的任何位置。

实例

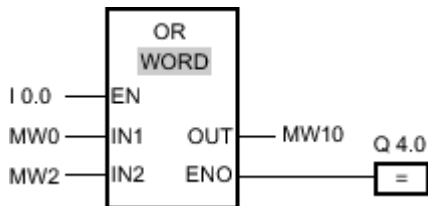


图 7-2

IN1	MW0 = 01010101 01010101
IN2	MW2 = 00000000 00001111
OUT	MW10= 01010101 01011111

如果输入 I 0.0 的信号状态为“1”，则执行“OR 逻辑运算”。输入 MW0 的值与输入 MW2 的值通过 OR 逻辑进行运算。逐位运算得出结果并发送到输出 MW10 中。输出 ENO 和输出 Q 4.0 的信号状态置位为“1”。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 XOR：异或逻辑运算

符号

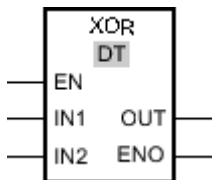


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN1	BYTE、WORD、 DWORD	I、Q、M、L、D 或常数	逻辑运算的第一个值
IN2	BYTE、WORD、 DWORD	I、Q、M、L、D 或常数	逻辑运算的第二个值
OUT	BYTE、WORD、 DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“异或逻辑运算”将输入 IN1 的值与输入 IN2 的值通过异或逻辑逐位运算，并通过输出 OUT 查询结果。

执行该运算时，输入 IN1 的值的位 0 与输入 IN2 的值的位 0 通过异或逻辑进行运算。结果存储在输出 OUT 的位 0 中。对指定值的所有其它位都执行相同的逻辑运算。

当该逻辑运算中的两个位中有一个位的信号状态为“1”时，结果位的信号状态为“1”。如果该逻辑运算的两个位的信号状态均为“1”或“0”，则对应的结果位将复位。

只有使能输入 EN 的信号状态为“1”时，才执行该运算。在这种情况下，输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“异或逻辑运算”可以放置在逻辑串中的任何位置。

实例

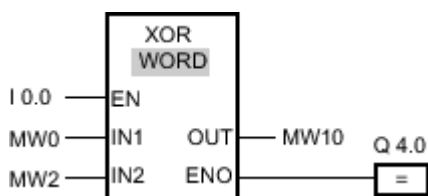


图 7-2

IN1	MW0 = 01010101 01010101
IN2	MW2 = 00000000 00001111
OUT	MW10 = 01010101 01011010

如果输入 I 0.0 的信号状态为“1”，则执行“异或逻辑运算”。输入 MW0 的值与输入 MW2 的值通过异或逻辑进行运算。逐位运算得出结果并发送到输出 MW10 中。输出 ENO 和输出 Q 4.0 的信号状态置位为“1”。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 选择 FBD 元素的数据类型 (页 419)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 INV : 创建反码

符号

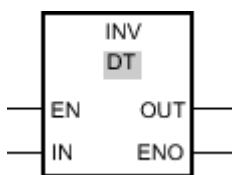


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
IN	BYTE、WORD、DWORD、USINT、UINT、UDINT、SINT、INT、DINT	I、Q、M、D、L 或常数	输入值
OUT	BYTE、WORD、DWORD、USINT、UINT、UDINT、SINT、INT、DINT	I、Q、M、D、L	输入 IN 的值的反码

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“创建反码”运算对输入 IN 的各位的信号状态取反。执行该运算时，输入 IN 的值与一个十六进制掩码 (表示 16 位数的 W#16#FFFF 或表示 32 位数的 DW#16#FFFF FFFF) 通过异或逻辑进行运算。这会将各个位的信号状态取反并且结果存储在输出 OUT 中。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。在这种情况下，输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“创建反码”运算可以放置在逻辑串中的任何位置。

实例

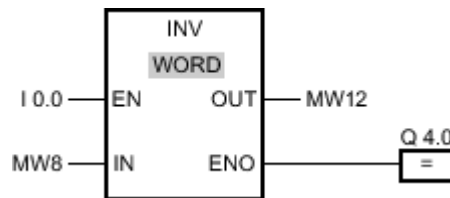


图 7-2

IN	MW8 = W#16#000F
OUT	MW12 = W#16#FFF0

如果输入 I 0.0 的信号状态为“1”，则执行“创建反码”运算。该运算对输入 MW8 的各个位的信号状态取反并将结果写入输出 MW12。输出 ENO 和输出 Q 4.0 的信号状态置位为“1”。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

选择 FBD 元素的数据类型 (页 419)

7.8.1.2 DECO : 解码

符号

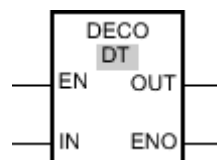


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L。	使能输出
IN	UINT	I、Q、M、D、L 或常数	输入值
OUT	BYTE、WORD、DWORD	I、Q、M、D、L	结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“解码”运算在输出值中设置一个位，该位通过输入值指定。

“解码”运算读取输入 IN 的值，然后在输出值中设置一个位，该位位置与读取的值一致。输出值中的其它位以零填充。当输入 IN 的值大于 31 时，将执行以 32 为模的模运算。

只有使能输入 EN 的信号状态为“1”时，才能启动“解码”运算。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“解码”运算可以放置在逻辑串中的任何位置。

实例

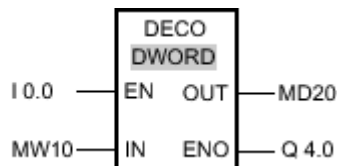


图 7-2

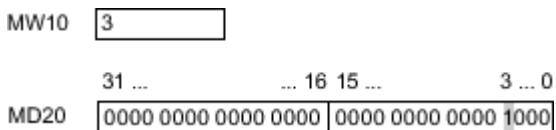


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“解码”运算。该运算通过输入 MW10 的值读取位号“3”，然后将输出 MD20 的值的第三位置位。

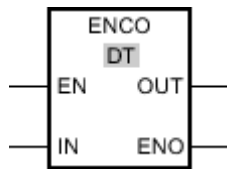
如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 选择 FBD 元素的数据类型 (页 419)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 ENCO : 编码

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L。	使能输出
IN	BYTE、WORD、 DWORD	I、Q、M、D、L 或常数	输入值
OUT	INT	I、Q、M、D、L	输出值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“编码”运算读取输入值中最低有效置位位的位号并将其存储在输出 OUT 中。

“编码”运算选择输入 IN 值的最低有效位并将该位号写入输出 OUT 的变量中。

只有使能输入 EN 的信号状态为“1”时，才能启动“编码”运算。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“编码”运算可以放置在逻辑串中的任何位置。

实例

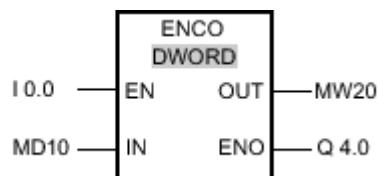


图 7-2

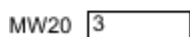
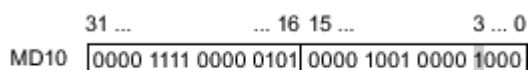


图 7-2

如果输入 I 0.0 的信号状态为“1”，则执行“编码”运算。该运算选择输入 MD10 的最低有效置位位并将位位置 3 写入输出 MW20 的变量中。

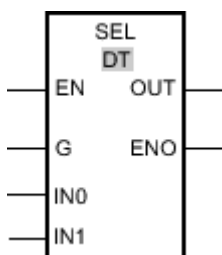
如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 更改 FBD 元素 (页 454)
- 插入 FBD 元素 (页 447)
- 选择 FBD 元素的数据类型 (页 419)
- 将操作数插入 FBD 指令 (页 457)

7.8.1.2 SEL : 选择

符号



参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
G	BOOL	I、Q、M、D、L	选择开关
IN0	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、CHAR	I、Q、M、D、L 或常数	第一个输入值
IN1	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、CHAR	I、Q、M、D、L 或常数	第二个输入值
OUT	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT、REAL、TIME、CHAR	I、Q、M、D、L	所选输入的值

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

“选择”操作依靠一个开关（参数 G）选择输入 IN0 或 IN1 中的一个并将其数据复制到输出 OUT。如果参数 G 的信号状态为“0”，则复制输入 IN0 的值。如果参数 G 的信号状态为“1”，则将输入 IN1 的值复制到输出 OUT。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”或执行该操作期间出错，将复位使能输出 ENO。

放置

“选择”操作可以放置在逻辑串中的任何位置。

实例

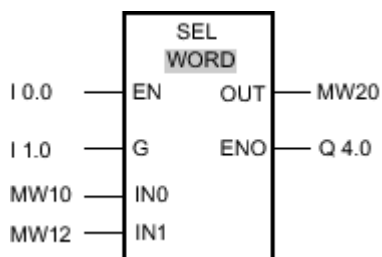


图 7-2

G	I 1.0 = 1
IN0	MW10 = W#16#0000
IN1	MW12 = W#16#FFFF
OUT	MW20 = W#16#FFFF

如果输入 I 0.0 的信号状态为“1”，则执行“选择”操作。根据输入 I 1.0 的信号状态，将选择输入 MW12 的值并复制到输出 MW20。如果操作执行过程中未发生错误，则置位输出 ENO 和 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

选择 FBD 元素的数据类型 (页 419)

7.8.1.2 MUX：多路复用

符号

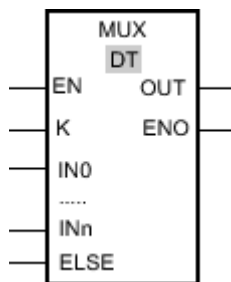


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、D、L (页 293)	使能输入
ENO	BOOL	I、Q、M、D、L	使能输出
K	UINT	I、Q、M、D、L	指定要复制哪个输入的数据。
IN0	BYTE、WORD、DWORD、INT、DINT、UINT、UDINT、SINT、USINT、REAL、CHAR、TIME	I、Q、M、D、L 或常数	第一个可用的输入
INn	BYTE、WORD、DWORD、INT、DINT、UINT、UDINT、SINT、USINT、REAL、CHAR、TIME	I、Q、M、D、L 或常数	可用的输入
ELSE	BYTE、WORD、DWORD、INT、DINT、UINT、UDINT、SINT、USINT、REAL、CHAR、TIME	I、Q、M、D、L 或常数	指定 K > n 时要复制的值。
OUT	BYTE、WORD、DWORD、INT、DINT、UINT、UDINT、SINT、USINT、REAL、CHAR、TIME	I、Q、M、D、L	要将值复制到的输出。

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“多路复用”操作将所选输入的数据复制到输出 OUT。功能框 MUX 的可选输入数可以扩展。输入会在该功能框中自动编号。从 IN0 起开始编号，每个新输入的编号连续递增。可以使用参数 K 确定应该将哪个输入的数据复制到输出 OUT。如果参数 K 的值大于可用输入数，则将参数 ELSE 的数据复制到输出 OUT，并将使能输出设置为信号状态“0”。

只有所有输入的变量与输出 OUT 的变量具有相同数据类型时，才能执行“多路复用”操作。此处的例外情况是，参数 K 只能指定为整数。

只有使能输入 EN 的信号状态为“1”时，才执行该操作。如果执行过程中未发生错误，则输出 ENO 的信号状态也为“1”。

如果满足下列条件之一，将复位使能输出 ENO：

- 使能输入 EN 的信号状态为“0”。
- 参数 K 的值大于可用输入数。
- 操作处理期间出错。

放置

“多路复用”操作可以放置在逻辑串中的任何位置。

实例

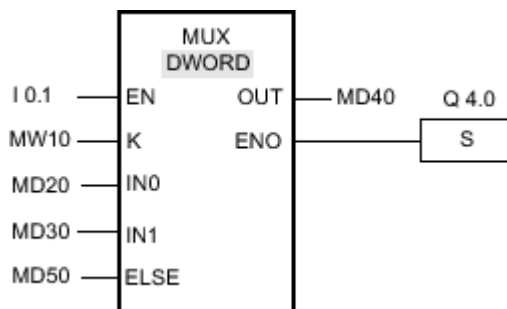


图 7-2

K	MW10 = 1
IN0	MD20 = DW#16#00000000
IN1	MD30 = DW#16#FFFFFFFF
ELSE	MD50 = DW#16#FFFF0000
OUT	MD40 = DW#16#FFFFFFFF

如果输入 I 0.1 的信号状态为“1”，则执行“多路复用”操作。根据输入 MW10 的值，将复制输入 MD30 的值并将其分配给输出 MD40 的变量。如果操作执行过程中未发生错误，则置位输出 ENO 和 Q 4.0。

参见

插入 FBD 元素 (页 447)

- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 移位和循环

7.8.1.2 SHR：右移

符号

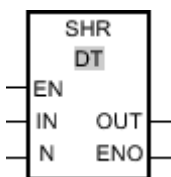


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT	I、Q、M、L、D 或常数	要移位的值。
N	UINT	I、Q、M、L、D 或常数	将移位的位数。
OUT	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT	I、Q、M、L、D	移位操作的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“右移”操作将输入 IN 的变量数据逐位右移，并通过输出 OUT 查询结果。使用参数 N 设置指定将移位的位数。

参数 N 的值为“0”时，输入 IN 的值将被复制到输出 OUT 的变量中。

当参数 N 的值大于位数时，输入 IN 的变量值将按其可用位数向右移位。

无符号值移位时，用零填充变量左侧区域中空出的位。如果指定值有符号，则用符号位的信号状态填充空出的位。

下图显示了一个整型数据类型的变量数据是如何右移四位的：

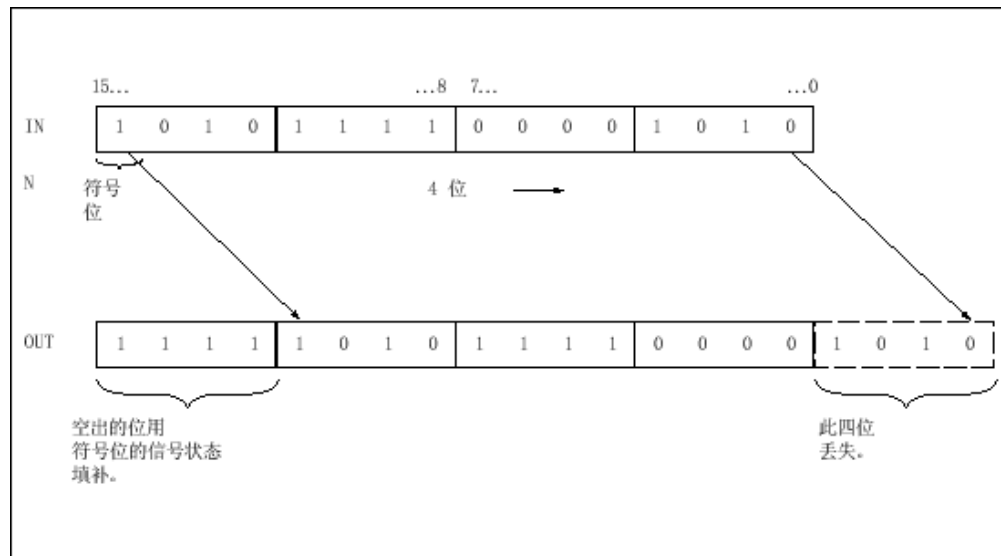


图 7-2

只有使能输入 EN 的信号状态为“1”时，才执行“右移”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“右移”操作可以放置在逻辑串中的任何位置。

实例

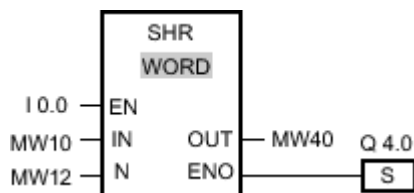


图 7-2

IN	MW10 = 0011 1111 1010 1111
N	MW12 = 3
OUT	MW40 = 0000 0111 1111 0101

如果输入 I 0.0 的信号状态为“1”，则执行“右移”操作。变量 MW10 中的数据右移三位。结果放在输出 MW40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

- 插入 FBD 元素 (页 447)
- 更改 FBD 元素 (页 454)
- 将操作数插入 FBD 指令 (页 457)
- 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 SHL : 左移

符号

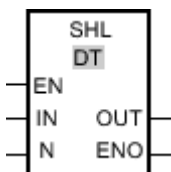


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT	I、Q、M、L、D 或常数	要移位的值。
N	UINT	I、Q、M、L、D 或常数	将移位的位数。
OUT	BYTE、WORD、DWORD、SINT、USINT、INT、UINT、DINT、UDINT	I、Q、M、L、D	移位操作的结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“左移”操作将输入 IN 中的变量数据逐位左移，并通过输出 OUT 查询结果。使用参数 N 设置指定将移位的位数。

参数 N 的值为“0”时，输入 IN 的值将被复制到输出 OUT 的变量中。

当参数 N 的值大于位数时，输入 IN 的变量值将按其可用位数向左移位。

用零填充变量右侧部分因移位空出的位。

下图显示了一个 WORD 数据类型的变量数据是如何左移六位的：

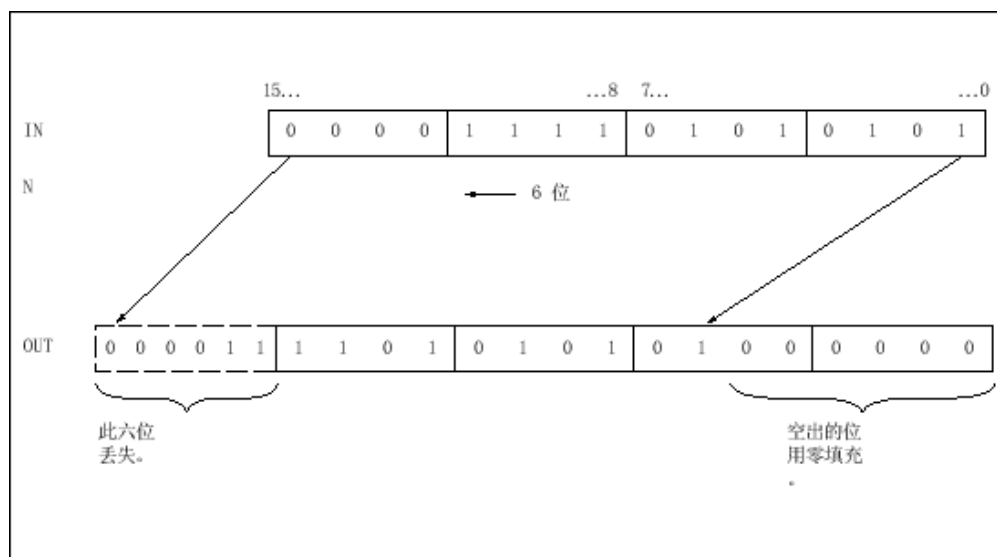


图 7-2

只有使能输入 EN 的信号状态为“1”时，才执行“左移”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“左移”操作可以放置在逻辑串中的任何位置。

实例

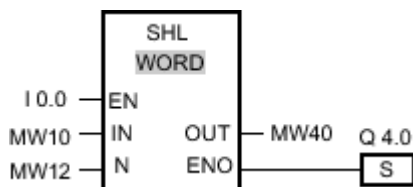


图 7-2

IN	MW10 = 0011 1111 1010 1111
N	MW12 = 4
OUT	MW40 = 1111 1010 1111 0000

如果输入 I 0.0 的信号状态为“1”，则执行“左移”操作。变量 MW10 的数据左移四位。结果放在输出 MW40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)
 选择 FBD 元素的数据类型 (页 419)

7.8.1.2 ROR：循环右移

符号

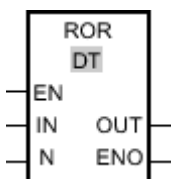


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	要循环移位的值。
N	UINT	I、Q、M、L、D 或常数	将循环移位的位数。
OUT	BYTE、WORD、DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“循环右移”操作将输入 IN 的变量数据逐位循环右移，并通过输出 OUT 查询结果。参数 N 指定将循环移位的位数。用挤出的位填充因循环移位空出的位。

参数 N 的值为“0”时，输入 IN 的值将被复制到输出 OUT 的变量中。

当参数 N 的值大于位数时，输入 IN 的变量值将按其可用位数进行循环移位。

下图显示了一个 DWORD 数据类型的变量数据是如何循环右移三位的：

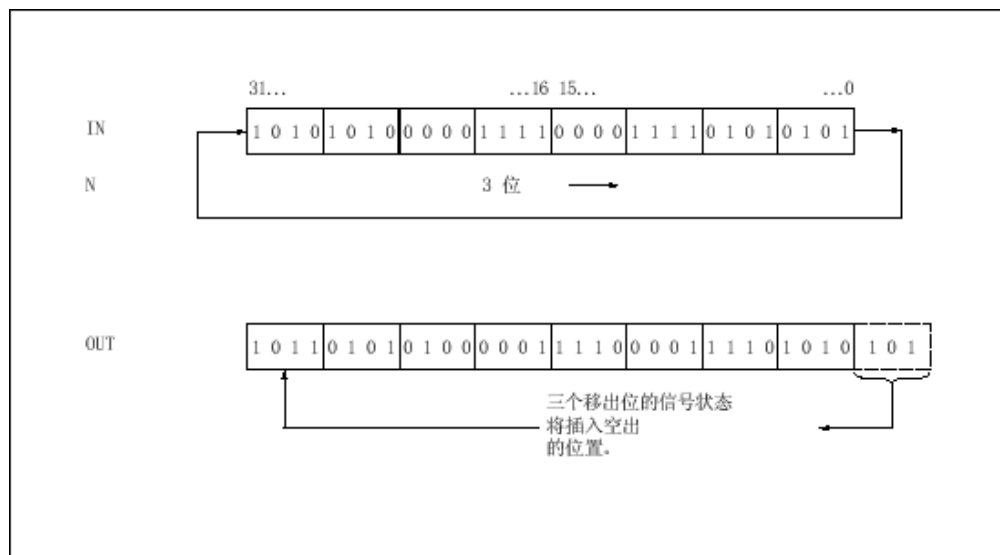


图 7-2

只有使能输出 ENO 的信号状态为“1”时，才执行“循环右移”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态复位为“0”。

放置

“循环右移”操作可以放置在逻辑串中的任何位置。

实例

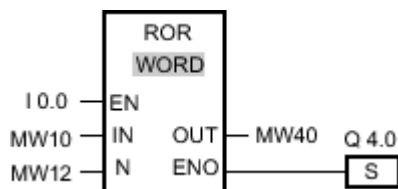


图 7-2

IN	MW10 = 0000 1111 1001 0101
N	MW12 = 5
OUT	MW40 = 1010 1000 0111 1100

如果输入 I 0.0 的信号状态为“1”，则执行“循环右移”操作。变量 MW10 的数据将循环右移五位。结果放在输出 MW40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

选择 FBD 元素的数据类型 (页 419)

将操作数插入 FBD 指令 (页 457)

7.8.1.2 ROL : 循环左移

符号

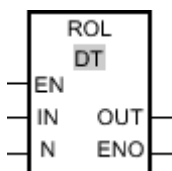


图 7-2

参数	数据类型	存储区	说明
EN	BOOL	I、Q、M、L、D (页 293)	使能输入
ENO	BOOL	I、Q、M、L、D	使能输出
IN	BYTE、WORD、DWORD	I、Q、M、L、D 或常数	要循环移位的值。
N	UINT	I、Q、M、L、D 或常数	将循环移位的位数。
OUT	BYTE、WORD、DWORD	I、Q、M、L、D	运算结果

可以从“DT”下拉列表中选择用于该操作的数据类型。

说明

可以使用“循环左移”操作将输入 IN 的变量数据逐位循环左移，并通过输出 OUT 查询结果。参数 N 指定将循环移位的位数。用挤出的位填充因循环移位空出的位。

如果参数 N 的值为“0”，输入 IN 的值将被复制到输出 OUT 的变量中。

如果参数 N 大于可用的位数，输入 IN 的变量值将按可用位数进行循环移位。

下图显示了一个 DWORD 数据类型的变量数据是如何循环左移三位的：

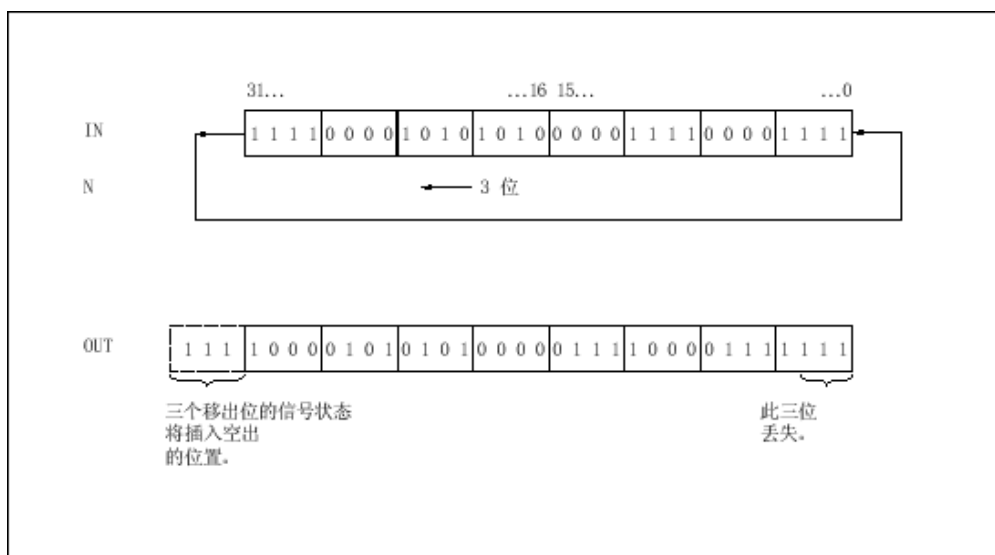


图 7-2

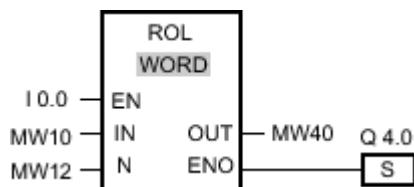
只有使能输出 ENO 的信号状态为“1”时，才执行“循环左移”操作。在这种情况下，使能输出 ENO 的信号状态也为“1”。

如果使能输入 EN 的信号状态为“0”，则使能输出 ENO 的信号状态也为“0”。

放置

“循环左移”操作可以放置在逻辑串中的任何位置。

实例



IN	MW10 = 1010 1000 1111 0110
OUT	MW12 = 5
N	MW40 = 0001 1110 1101 0101

如果输入 I 0.0 的信号状态为“1”，则执行“循环左移”操作。变量 MW10 的数据将循环左移五位。结果放在输出 MW40 中。如果运算执行过程中未发生错误，则输出 ENO 的信号状态将为“1”并置位输出 Q 4.0。

参见

插入 FBD 元素 (页 447)

更改 FBD 元素 (页 454)

将操作数插入 FBD 指令 (页 457)

选择 FBD 元素的数据类型 (页 419)

7.8.2 扩展指令

7.8.2.1 时钟和日历

7.8.2.1 日时钟功能

7.8.2.1 WR_SYS_T

描述

可使用 WR_SYS_T 设置 CPU 时钟的日期和时间。在指令的输入 IN 指定 DTL 格式的日期和时间。在输出 RET_VAL 中，可以查询指令执行期间是否出错。

不能使用“WR_SYS_T”指令发送有关本地时区或夏令时的信息。

参数

参数	声明	数据类型	存储区	说明
IN	IN	DTL	M、D、L	日期和时间
RET_VAL	OUT	INT	M、D、L	指令的状态

参数 RET_VAL

错误代码 (W#16#...)	描述
0000	无错误
8081	年无效
8082	月无效
8083	日无效
8084	小时信息无效
8085	分钟信息无效
8086	秒信息无效

错误代码 (W#16#...)	描述
8087	纳秒信息无效
80B0	实时时钟发生了故障。

7.8.2.1 RD_SYS_T

描述

可使用 RD_SYS_T 读取 CPU 时钟的当前日期和当前时间。数据以 DTL 格式放在指令的输出 OUT 中。得出的值不包含有关本地时区或夏令时的信息。在输出 RET_VAL 中，可以查询指令执行期间是否出错。

参数

参数	声明	数据类型	存储区	说明
OUT	OUT	DTL	M、D、L	CPU 的日期和时间
RET_VAL	OUT	INT	M、D、L	指令的状态

参数 RET_VAL

错误代码 (W#16#...)	描述
0000	无错误
8222	结果超出允许的取值范围
8223	结果无法以指定的数据类型保存

7.8.2.1 RD_LOC_T

说明

可使用 RD_LOC_T 从 CPU 时钟读取当前本地时间，并在输出 OUT 以 DTL 格式输出该值。在 CPU 时钟的组态中设置的时区和夏令时开始时间以及标准时间，其相关信息均包括在本地时间信息中。

参数

参数	声明	数据类型	存储区	说明
OUT	OUT	DTL	M、D、L	本地时间
RET_VAL	OUT	INT	M、D、L	指令的状态

参数 RET_VAL

错误代码 (W#16#...)	描述
0000	无错误
8080	无法读取本地时间。

7.8.2.1 T_CONV

说明

使用 T_CONV 可将输入 IN 的值转换成输出 OUT 指定的数据格式。可实现下列转换：

- 时间 (TIME) 到数字值 (DINT) 的转换
- 数值 (DINT) 到时间 (TIME) 的转换

通过选择指令输入和输出的数据类型来决定转换的类型。可通过输出 OUT 查询转换结果。

参数

参数	声明	数据类型	存储区	说明
IN	IN	TIME、DINT	M、D、L	要转换的值
OUT	OUT	TIME、DINT	M、D、L	转换结果

7.8.2.1 T_ADD

描述

使用 T_ADD 可将输入 IN1 的时间与输入 IN2 的时间相加。通过输出 OUT 查询结果。可以对下列格式进行相加操作：

- 时间段 (TIME) 与时间段 (TIME) 相加。结果可以输出到 TIME 格式的变量中。
- 时间段 (TIME) 与时间点 (DTL) 相加。结果可以输出到 DTL 格式的变量中。

通过选择指令输入和输出的数据类型来决定输入 IN1 和输出 OUT 的格式。在输入 IN2 中，只能指定 TIME 格式的时间。

参数

参数	声明	数据类型	存储区	描述
IN1	IN	TIME、DTL	M、D、L	被加数
IN2	IN	TIME	M、D、L	被加数
OUT	OUT	TIME、DTL	M、D、L	相加的结果

7.8.2.1 T_SUB

说明

使用 T_SUB 可将输入 IN1 的时间与输入 IN2 的时间相减。通过输出 OUT 查询差值。可以对下列格式进行相减操作：

- 时间段 (TIME) 与时间段 (TIME) 相减。结果可以输出到 TIME 格式的变量中。
- 从某时间点 (DTL) 减去一个时间段 (TIME)。结果可以输出到 DTL 格式的变量中。

通过选择指令输入和输出的数据类型来决定输入 IN1 和输出 OUT 的格式。在输入 IN2 中，只能指定 TIME 格式的时间。

参数

参数	声明	数据类型	存储区	描述
IN1	IN	TIME、DTL	M、D、L	被减数
IN2	IN	TIME	M、D、L	减数
OUT	OUT	TIME、DTL	M、D、L	相减的结果

7.8.2.1 T_DIFF

描述

使用 T_DIFF 可将输入 IN1 的时间与输入 IN2 的时间相减。以 TIME 格式通过输出 OUT 输出结果。在输入 IN1 和 IN2 中只能指定 DTL 格式的值。

如果在输入 IN2 中指定的时间大于在输入 IN1 中指定的时间，则结果将以负值的形式通过输出 OUT 输出。如果该指令的结果超出允许范围，则此结果将限制为相应的值，并且使能输出 ENO 将设置为“0”。

参数

参数	声明	数据类型	存储区	说明
IN1	IN	DTL	M、D、L	被减数
IN2	IN	DTL	M、D、L	减数
OUT	OUT	TIME	M、D、L	TIME 格式的差值

7.8.2.2 STRING 和 CHAR

7.8.2.2 S_CONV

描述

使用 S_CONV 可将输入 IN 的值转换成在输出 OUT 中指定的数据格式。可实现下列转换：

- 字符串 (STRING) 转换为数字值：
在输入 IN 中指定的字符串的所有字符都将进行转换。允许的字符为数字 0 到 9、小数点以及加号和减号。字符串的第一个字符可以是有效数字或符号。前导空格和指数表示将被忽略。无效字符可能会中断字符转换。此时，使能输出 ENO 将设置为“0”。
可通过选择输出 OUT 的数据类型来决定转换的输出格式。
- 数字值转换为字符串 (STRING)：
通过选择输入 IN 的数据类型来决定要转换的数字值格式。必须在输出 OUT 中指定一个有效的 STRING 数据类型的变量。转换后的字符串长度取决于输入 IN 的值。由于第一个字节包含字符串的最大长度，第二个字节包含字符串的实际长度，因此转换的结果从字符串的第三个字节开始存储。输出正数字值时不带符号。
- 复制字符串：
如果在指令的输入和输出均输入 STRING 数据类型，则输入 IN 的字符串将被复制到输出 OUT。如果输入 IN 字符串的实际长度超出输出 OUT 字符串的最大长度，则将复制 IN 字符串中完全适合 OUT 的字符串的那部分，并且使能输出 ENO 将设置为“0”值。

参数

将字符串转换为数字值的参数：

参数	声明	数据类型	存储区	说明
IN	IN	STRING	D、L 或常数	要转换的值

参数	声明	数据类型	存储区	说明
OUT	OUT	USINT、 SINT、 UINT、INT、 UDINT、 DINT、REAL	I、Q、M、D、L	转换结果

将数字值转换为字符串的参数：

参数	声明	数据类型	存储区	说明
IN	IN	USINT、 SINT、 UINT、INT、 UDINT、 DINT、REAL	I、Q、M、D、L 或常数	要转换的值
OUT	OUT	STRING	D、L	转换结果

用于复制字符串的参数：

参数	声明	数据类型	存储区	说明
IN	IN	STRING	D、L 或常数	要复制的值
OUT	OUT	STRING	D、L	复制操作的结果

实例

下表列出了字符串转换为数字值的实例：

IN (STRING)	OUT (数据类型)	OUT (值)	ENO 状态
'123'	INT/DINT	123	1
'-00456'	INT/DINT	-456	1
'123.45'	INT/DINT	123	1
'+2345'	INT/DINT	2345	1
'00123AB'	INT/DINT	123	1
'123'	REAL	123.0	1
'123.45'	REAL	123.45	1
'1.23e-4'	REAL	1.23	1
'1.23E-4'	REAL	1.23	1
'12,345.67'	REAL	12345.67	1

IN (STRING)	OUT (数据类型)	OUT (值)	ENO 状态
'3.4e39'	REAL	3.4	1
'-3.4e39'	REAL	-3.4	1
'1.17549e-38'	REAL	1.17549	1
'12345'	SINT	0	0
'A123'	-/-	0	0
"	-/-	0	0
'++123'	-/-	0	0
'+-123'	-/-	0	0

实例

下表列出了数字值转换为字符串的实例：

IN (值)	IN (数据类型)	OUT (STRING)	ENO 状态
123	UINT	'123'	1
0	UINT	'0'	1
12345678	UDINT	'12345678'	1
-Inf ¹⁾	REAL	'Inf'	0
+Inf ²⁾	REAL	'Inf'	0
NaN ³⁾	REAL	'NaN'	0

1) -Inf：表示值为负无穷大的浮点数。
2) +Inf：表示值为正无穷大的浮点数。
3) NaN：作为无效数学运算结果的返回值。

参见

STRING (页 353)

7.8.2.2 STRG_VAL

描述

使用 STRG_VAL 可将字符串转换为数字值。在输入 IN 中指定要转换的字符串。可通过选择输出 OUT 的数据类型来决定输出值的格式。通过输出 OUT 查询结果。

从参数 P 中指定位置的字符开始转换。例如，如果参数 P 中指定的值为“1”，则将从指定字符串的第一个字符开始转换。转换允许的字符为数字 0 到 9、小数点、逗号小数点、符号“E”和“e”以及加号和减号字符。无效字符可能会中断转换。此时，使能输出 ENO 将设置为“0”。

使用参数 FORMAT 可指定要如何解释字符串中的字符。也可以使用“STRG_VAL”指令来转换和表示指数值。只能为参数 FORMAT 指定 USINT 数据类型的变量。

下表列出了参数 FORMAT 的可能值及其含义：

值 (W#16#...)	表示法	小数点表示法
0000	小数	"."
0001		" , "
0002	指数	" . "
0003		" , "
0004 到 FFFF	无效值	

参数

参数	声明	数据类型	存储区	说明
IN	IN	STRING	D、L 或常数	要转换的字符串
FORMAT	IN	WORD	I、Q、M、D、L 或常数	字符的输出格式
P	IN_OUT	UINT	I、Q、M、D、L	转换的起始字符
OUT	OUT	USINT、SINT、 UINT、INT、 UDINT、DINT、 REAL	I、Q、M、D、L	转换结果

实例

下表列出了字符串转换为数字值的实例：

IN (STRING)	FORMAT (W#16#...)	OUT (数据类型)	OUT (值)	ENO 状态
'123'	0000	INT/DINT	123	1
'-00456'	0000	INT/DINT	-456	1
'123.45'	0000	INT/DINT	123	1
'+2345'	0000	INT/DINT	2345	1
'00123AB'	0000	INT/DINT	123	1
'123'	0000	REAL	123.0	1
'-00456'	0001	REAL	-456.0	1

IN (STRING)	FORMAT (W#16#....)	OUT (数据类型)	OUT (值)	ENO 状态
'+00456'	0001	REAL	456.0	1
'123.45'	0000	REAL	123.45	1
'123.45'	0001	REAL	12345.0	1
'123,45'	0000	REAL	12345.0	1
'123,45'	0001	REAL	123.45	1
'.00123AB'	0001	REAL	123.0	1
'1.23e-4'	0000	REAL	1.23	1
'1.23E-4'	0000	REAL	1.23	1
'1.23E-4'	0002	REAL	1.23E-4	1
'12,345.67'	0000	REAL	12345.67	1
'12,345.67'	0001	REAL	12.345	1
'3.4e39'	0002	REAL	W#16#7F800000	1
'-3.4e39'	0002	REAL	W#16#FF800000	1
'1.1754943e-38'	0002	REAL	0.0	1
'12345'	-/-	SINT	0	0
'A123'	-/-	-/-	0	0
"	-/-	-/-	0	0
'++123'	-/-	-/-	0	0
'+-123'	-/-	-/-	0	0

参见

STRING (页 353)

7.8.2.2 VAL_STRG

描述

使用 VAL_STRG 可将数字值转换为字符串。在输入 IN 中指定要转换的值。通过选择数据类型来决定数字值的格式。通过输出 OUT 查询转换结果。

通过参数 P 可指定从字符串中的哪个字符开始写入结果。例如，如果参数 P 中指定的值为“2”，则将从字符串的第二个字符开始保存转换值。

通过参数 SIZE 可以指定字符串中写入的字符数。这要从参数 P 中指定的字符开始算起。如果由参数 P 和 SIZE 定义的长度不够，则使能输出 ENO 将设置为“0”。如果输出值比指定长度短，则结果将以右对齐方式写入字符串。空字符位置将填入空格。

转换允许的字符为数字 0 到 9、小数点、逗号小数点、符号“E”和“e”以及加号和减号字符。无效字符可能会中断转换。此时，使能输出 ENO 将设置为“0”。

使用参数 FORMAT 可指定在转换期间如何解释数字值以及如何将其写入字符串。只能为参数 FORMAT 指定 USINT 数据类型的变量。

下表列出了参数 FORMAT 的可能值及其含义：

值 (W#16#...)	表示法	符号	小数点表示法
0000	小数	"_"	"."
0001			","
0002	指数		"."
0003			","
0004	小数	"+”和“-”	"."
0005			","
0006	指数		"."
0007			","
0008 到 FFFF	无效值		

通过参数 PREC 可定义转换浮点数时的小数位。REAL 数据类型的数字值所支持的最大精度为 7 位。如果要转换的值为整数，则使用参数 PREC 来指定小数点的位置。

参数

参数	声明	数据类型	存储区	说明
IN	IN	USINT、SINT UINT、INT UDINT、DINT REAL	I、Q、M、D、L 或 常数	要转换的值
SIZE	IN	USINT	I、Q、M、D、L 或 常数	字符位数
PREC	IN	USINT	I、Q、M、D、L 或 常数	小数位数
FORMAT	IN	WORD	I、Q、M、D、L 或 常数	字符的输出格式
P	IN_OUT	UINT	I、Q、M、D、L	开始写入结果的字符。
OUT	OUT	STRING	D、L	转换结果

实例

下表列出了数字值转换为字符串的实例。

IN (值)	IN (数据类型)	P	SIZE	FORMAT (W#16#....)	PREC	OUT (STRING)	ENO 状态
123	UINT	16	10	0000	0	xxxxxxxx123 C	1
0	UINT	16	10	0000	2	xxxxxxx0.00 C	1
12345678	UDINT	16	10	0000	3	x12345.678 C	1
12345678	UDINT	16	10	0001	3	x12345.678 C	1
123	INT	16	10	0004	0	xxxxxxx+123 C	1
-123	INT	16	10	0004	0	xxxxxxx-123 C	1
-0.00123	REAL	16	10	0004	4	xxx-0.0012 C	1
-0.00123	REAL	16	10	0006	4	-1.2300E-3 C	1
-Inf ¹⁾	REAL	16	10	-/-	4	xxxxxxx-INF C	0
+Inf ²⁾	REAL	16	10	-/-	4	xxxxxxx+INF C	0
NaN ³⁾	REAL	16	10	-/-	4	xxxxxxxxNaN C	0
12345678	UDINT	16	6	-/-	3	xxxxxxxxxxx C	0

“x”表示空格
 1) -Inf：表示值为负无穷大的浮点数。
 2) +Inf：表示值为正无穷大的浮点数。
 3) NaN：作为无效数学运算结果的返回值。

参见

STRING (页 353)

7.8.2.2 指令

7.8.2.2 LEN

描述

STRING 类型的变量包含两个长度：最大长度和当前长度（即当前有效字符的数量）。每个变量的字符串最大长度在 STRING 关键字的方括号中指定。当前长度表示实际使用的字符位置数。当前长度必须小于或等于最大长度。字符串占用的字节数为最大长度加 2。

可使用“LEN”指令查询在输入 IN 中指定的字符串的当前长度，并在输出 OUT 以数字值的形式将其输出。空字符串 (") 的长度为零。

如果操作处理期间出错，则将输出空字符串。

参数

参数	声明	数据类型	存储区	说明
IN	IN	STRING	D、L 或常数	字符串
OUT	OUT	INT	I、Q、M、D、L	有效字符数

参见

STRING (页 353)

7.8.2.2 CONCAT

说明

CONCAT 连接字符串参数 IN1 和 IN2 以形成一个字符串，并放在 OUT 中。目标字符串必须足够长，否则结果字符串将被截短并且使能输出 ENO 将设置为“0”值。

如果指令处理期间出错并且可以写入到输出 OUT，则将输出空字符串。

参数

参数	声明	数据类型	存储区	说明
IN1	IN	STRING	D、L 或常数	字符串
IN2	IN	STRING	D、L 或常数	字符串
OUT	OUT	STRING	D、L	结果字符串

参见

STRING (页 353)

7.8.2.2 LEFT

描述

可使用 LEFT 从输入 IN 字符串的第一个字符开始提取出一部分字符串。通过参数 L 指定要提取的字符数。以 STRING 格式通过输出 OUT 输出提取的字符。

如果要提取的字符数大于字符串的当前长度，则输出 OUT 将返回输入字符串。参数 L 的值为“0”时或者输入值为空字符串时都将返回空字符串。如果参数 L 的值为负值，则将输出空字符串并且使能输出 ENO 将设置为“0”值。

如果指令处理期间出错并且可以写入到输出 OUT，则将输出空字符串。

参数

参数	声明	数据类型	存储区	描述
IN	IN	STRING	D、L 或常数	字符串
L	IN	INT	I、Q、M、D、L 或常数	要提取的字符数
OUT	OUT	STRING	D、L	提取的部分字符串

参见

STRING (页 353)

7.8.2.2 RIGHT**描述**

可使用 RIGHT 从输入 IN 字符串的最后一个字符开始提取出一部分字符串。通过参数 L 指定要提取的字符数。以 STRING 格式通过输出 OUT 输出提取的字符。

如果要提取的字符数大于字符串的当前长度，则输出 OUT 将返回输入字符串。参数 L 的值为“0”时或者输入值为空字符串时都将返回空字符串。如果参数 L 的值为负值，则将输出空字符串并且使能输出 ENO 将设置为“0”值。

如果指令处理期间出错并且可以写入到输出 OUT，则将输出空字符串。

参数

参数	声明	数据类型	存储区	描述
IN	IN	STRING	D、L 或常数	字符串
L	IN	INT	I、Q、M、D、L 或常数	要提取的字符数
OUT	OUT	STRING	D、L	提取的部分字符串

参见

STRING (页 353)

7.8.2.2 MID

说明

可使用 MID 将输入 IN 字符串的一部分提取出来。可通过参数 P 指定要提取的第一个字符的位置。通过参数 L 指定要提取的字符串长度。提取的部分字符串通过输出 OUT 输出。

执行该指令时应遵守以下规则：

- 如果要提取的字符数超出输入 IN 字符串的当前长度，则将输出从字符位置 P 开始到该字符串末尾的这一部分字符串。
- 如果通过参数 P 指定的字符位置超出输入 IN 字符串的当前长度，则将通过输出 OUT 输出空字符串并且使能输出 ENO 将设置为“0”值。
- 如果参数 L 或 P 的值等于零或为负值，则将通过输出 OUT 输出空字符串并且使能输出 ENO 将设置为“0”值。

如果指令处理期间出错并且可以写入到输出 OUT，则将输出空字符串。

参数

参数	声明	数据类型	存储区	说明
IN	IN	STRING	D、L 或常数	字符串
P	IN	INT	I、Q、M、D、L 或常数	要提取的第一个字符的位置（第一个字符 = 1）
L	IN	INT	I、Q、M、D、L 或常数	要提取的字符串的长度
OUT	OUT	STRING	D、L	提取的部分字符串

参见

STRING (页 353)

7.8.2.2 DELETE

说明

可使用 DELETE 将输入 IN 字符串的一部分删除。可通过参数 P 指定要删除的第一个字符的位置。可使用参数 L 指定要删除的字符数。剩余部分的字符串通过输出 OUT 以 STRING 格式输出。

执行该指令时应遵守以下规则：

- 如果参数 L 或 P 的值等于零，则输出 OUT 将返回输入字符串。
- 如果参数 P 的值大于输入 IN 字符串的当前长度，则输出 OUT 将返回输入字符串。
- 如果要删除的字符数大于输入 IN 字符串的长度，则将输出空字符串。
- 如果参数 L 或 P 的值为负值，则将输出空字符串并且使能输出 ENO 将设置为“0”值。

如果指令处理期间出错并且可以写入到输出 OUT，则将输出空字符串。

参数

参数	声明	数据类型	存储区	说明
IN	IN	STRING	D、L 或常数	字符串
L	IN	INT	I、Q、M、D、L 或常数	要删除的字符数
P	IN	INT	I、Q、M、D、L 或常数	要删除的第一个字符的位置
OUT	OUT	STRING	D、L	结果字符串

参见

STRING (页 353)

7.8.2.2 INSERT

说明

从字符串 1 的某个字符位置开始插入字符串 2，并将结果存储在目标字符串中。可使用参数 P 指定要插入字符的字符位置。以 STRING 格式通过输出 OUT 输出结果。

执行该指令时应遵守以下规则：

- 如果参数 P 的值超出输入 IN1 字符串的当前长度，则输入 IN2 的字符串将附加到输入 IN1 的字符串的后面。
- 如果参数 P 的值为负值或等于零，则将通过输出 OUT 输出空字符串。使能输出 ENO 将设置为“0”值。
- 如果结果字符串比在输出 OUT 中指定的变量长，则结果字符串将被限制为有效长度。使能输出 ENO 将设置为“0”值。

如果指令处理期间出错并且可以写入到输出 OUT，则将输出空字符串。

参数

参数	声明	数据类型	存储区	说明
IN1	IN	STRING	D、L 或常数	字符串
IN2	IN	STRING	D、L 或常数	要插入的字符串
P	IN	INT	I、Q、M、D、L 或常数	插入位置
OUT	OUT	STRING	D、L	结果字符串

参见

STRING (页 353)

7.8.2.2 REPLACE

说明

可使用 REPLACE 将输入 IN1 的字符串替换为输入 IN2 的字符串。可通过参数 P 指定要替换的第一个字符的位置。通过参数 L 指定要替换的字符数。以 STRING 格式通过输出 OUT 输出结果。

执行该指令时应遵守以下规则：

- 如果参数 L 的值等于零，则输出 OUT 将返回输入 IN1 的字符串。
- 如果 P 等于 1，则将从输入 IN1 字符串的第一个字符开始（包括该字符）对其进行替换。
- 如果参数 P 的值超出输入 IN1 字符串的当前长度，则输入 IN2 的字符串将附加到输入 IN1 的字符串的后面。
- 如果参数 P 的值为负值或等于零，则将通过输出 OUT 输出空字符串。使能输出 ENO 将设置为“0”值。
- 如果结果字符串比在输出 OUT 中指定的变量长，则结果字符串将被限制为有效长度。使能输出 ENO 将设置为“0”值。

如果指令处理期间出错并且可以写入到输出 OUT，则将输出空字符串。

参数

参数	声明	数据类型	存储区	说明
IN1	IN	STRING	D、L 或常数	要替换其中字符的字符串。
IN2	IN	STRING	D、L 或常数	含有要插入的字符的字符串。
L	IN	INT	I、Q、M、D、L 或常数	要替换的字符数
P	IN	INT	I、Q、M、D、L 或常数	要替换的第一个字符的位置
OUT	OUT	STRING	D、L	结果字符串

参见

STRING (页 353)

7.8.2.2 FIND

说明

可使用 FIND 来搜索输入 IN1 的字符串以查找特定字符或特定字符串。在输入 IN2 中指定要搜索的值。搜索从左向右进行。将通过输出 OUT 输出第一个搜索结果的位置。如果搜索未返回任何匹配值，则将通过输出 OUT 输出“0”值。

如果指令处理期间出错，则将输出空字符串。

参数

参数	I/O	数据类型	存储区	说明
IN1	IN	STRING	D、L 或常数	被搜索的字符串
IN2	IN	STRING、 CHAR	D、L 或常数 (对于 CHAR 还包括 I、Q、 M)	要搜索的字符
OUT	OUT	INT	I、Q、M、 D、L	字符位置

参见

STRING (页 353)

7.8.2.3 程序控制

7.8.2.3 RE_TRIGR

说明

使用 RE_TRIGR 可重新启动 CPU 循环监视。然后循环时间监视将以 CPU 组态中设置的时间重新启动。通过重新启动循环监视时间，可防止触发错误或者 CPU 切换到 STOP。

“RE_TRIGR”指令可用于优先等级为 1 的块（例如，程序循环 OB）以及其中调用的块。

如果在优先级更高的块（例如硬件中断、诊断错误中断或循环中断）中调用该指令，则使能输出 ENO 将设置为“0”，并且“RE_TRIGR”不会执行。

参数

“RE_TRIGR”指令没有参数。

参见

事件和 OB (页 298)

7.8.2.3 STP

描述

使用 STP 可将 CPU 切换到 STOP 模式，从而终止程序执行。从 RUN 切换到 STOP 的效果取决于 CPU 的组态。

使能输入 EN 为“1”时，CPU 将切换到“STOP”模式并且终止程序执行。不会评估“ENO”的信号状态。

使能输入 EN 为“0”时，ENO 也为“0”。

参数

“STP”指令没有参数。

7.8.2.3 GetError

描述

通过“GetError”指令可查询块内发生的错误。如果块执行期间系统信号出错，则在输出 ERROR 的变量中保存有关发生的第一个错误的详细信息。

在输出 ERROR 中只能指定“ErrorStruct”系统数据类型的变量。“ErrorStruct”系统数据类型指定了存储错误相关信息的具体结构。使用额外的指令可评估此结构并编写适当的响应。消除第一个错误后，该指令会输出关于发生的下一个错误的信息。

参数

参数	声明	数据类型	存储区	说明
ERROR	OUT	ErrorStruct	D、L	错误信息

“ErrorStruct”数据类型

下表列出了“ErrorStruct”数据类型的变量可包含的信息：

结构组件	数据类型	说明
ERROR_ID	WORD	错误 ID
FLAGS	BYTE	显示块调用期间是否出错。 16#01: 块调用期间出错。 16#00: 块调用期间未出错。
REACTION	BYTE	默认响应： 0: 忽略（写入错误）， 1: 以替代值“0”继续（读取错误）， 2: 跳过指令（系统错误）

结构组件	数据类型	说明																																																						
BLOCK_TYPE	BYTE	出错块的类型： 1: OB 2: FC 3: FB																																																						
PAD_0	BYTE	用于划分 ErrorStruct 单独结构区域的内部字节。该字节的内容不相关。																																																						
CODE_BLOCK_NUMBER	UINT	代码块的编号																																																						
ADDRESS	UDINT	对内部存储器的引用																																																						
MODE	BYTE	访问模式：根据具体的访问类型，可输出以下信息：																																																						
		<table border="1"> <thead> <tr> <th>模式</th> <th>(A)</th> <th>(B)</th> <th>(C)</th> <th>(D)</th> <th>(E)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>1</td> <td></td> <td></td> <td></td> <td></td> <td>偏移量</td> </tr> <tr> <td>2</td> <td></td> <td></td> <td>区域</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>位置</td> <td>范围</td> <td></td> <td>编号</td> <td></td> </tr> <tr> <td>4</td> <td></td> <td></td> <td>区域</td> <td></td> <td>偏移量</td> </tr> <tr> <td>5</td> <td></td> <td></td> <td>区域</td> <td>DB 号</td> <td>偏移量</td> </tr> <tr> <td>6</td> <td>PtrNo./ Acc</td> <td></td> <td>区域</td> <td>DB 号</td> <td>偏移量</td> </tr> <tr> <td>7</td> <td>PtrNo./ Acc</td> <td>插槽号/ 范围</td> <td>区域</td> <td>DB 号</td> <td>偏移量</td> </tr> </tbody> </table>	模式	(A)	(B)	(C)	(D)	(E)	0						1					偏移量	2			区域			3	位置	范围		编号		4			区域		偏移量	5			区域	DB 号	偏移量	6	PtrNo./ Acc		区域	DB 号	偏移量	7	PtrNo./ Acc	插槽号/ 范围	区域	DB 号	偏移量
		模式	(A)	(B)	(C)	(D)	(E)																																																	
		0																																																						
		1					偏移量																																																	
		2			区域																																																			
		3	位置	范围		编号																																																		
		4			区域		偏移量																																																	
		5			区域	DB 号	偏移量																																																	
6	PtrNo./ Acc		区域	DB 号	偏移量																																																			
7	PtrNo./ Acc	插槽号/ 范围	区域	DB 号	偏移量																																																			
PAD_1	BYTE	用于划分 ErrorStruct 单独结构区域的内部字节。该字节的内容不相关。																																																						
OPERAND_NUMBER	UINT	机器命令的操作数																																																						
POINTER_NUMBER_LOCATION	UINT	(A) 内部指针																																																						
SLOT_NUMBER_SCOPE	UINT	(B) 内部存储器中的存储区																																																						
AREA	BYTE	(C) 存储区： L：16#40 – 4E、86、87、8E、8F、C0 – CE E：16#81 A：16#82 M：16#83 DB：16#84、85、8A、8B																																																						
PAD_2	BYTE	用于划分 ErrorStruct 单独结构区域的内部字节。该字节的内容不相关。																																																						
DB_NUMBER	UINT	(D) 数据块编号																																																						

结构组件	数据类型	说明
OFFSET	UDINT	(E) 操作数的相对地址

只有使能输入 EN 返回信号状态“1”且存在出错信息时，“GetError”指令的使能输出 ENO 才会置位。如果这些条件中有任何一个未满足，则其余的程序执行不受“GetError”指令影响。

“GetError”指令也可用于向调用块转发有关出错状态的报警。为此，该指令必须放置在被调用块的最后一个程序段中。

注意

“GetError”指令将启用块内的局部错误处理。如果将“GetError”插入块的程序代码中，则出错时任何预定义的系统响应都将被忽略。

实例

下图显示了该实例的 LAD 程序。

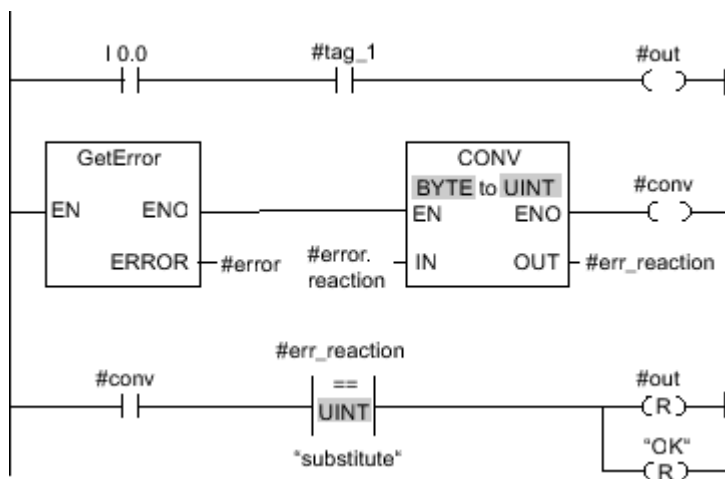


图 7-2

发生错误时，“GetError”指令将错误信息返回到输出 ERROR 局部创建的“#error”结构中。将对错误信息进行转换并使用“等于”比较运算进行评估。有关错误类型的信息为分配给该指令的第一个比较值。对于第二个比较值，变量“替换值”中指定了值“1”。如果该错误为读取错误，则满足比较指令的条件。此时，输出“#out”和“OK”将复位。

参见

局部错误处理 (页 470)

7.8.2.3 GetErrorID

描述

通过“GetErrorID”指令可查询块内发生的错误。如果块执行期间系统信号出错，则在输出“ID”的变量中保存所发生的第一个错误的错误 ID。在输出“ID”只能指定 WORD 数据类型的变量。消除第一个错误后，该指令会输出发生的下一个错误的错误 ID。

只有“GetErrorID”指令的输入返回信号状态“1”且存在错误信息时，该指令的输出才会被置位。如果这些条件中有任何一个未满足，则其余的程序执行不受“GetErrorID”影响。

“GetErrorID”指令也可用于向调用块发送有关错误状态的报警。为此，该指令必须放置在被调用块的最后一个程序段中。

注意

“GetErrorID”指令将启用块内的局部错误处理。如果将“GetErrorID”插入块的程序代码中，则出错时任何预定义的系统响应都将被忽略。

参数

参数	声明	数据类型	存储区	说明
ID	OUT	WORD	I、Q、M、D、L	错误 ID

“ID”参数

ID (十六进制)	ID (十进制)	说明
2503	9475	指针无效
2522	9506	读取错误：操作数超出有效范围
2523	9507	写入错误：操作数超出有效范围
2524	9508	读取错误：操作数无效
2525	9509	写入错误：操作数无效
2528	9512	读取错误：数据对齐
2529	9513	写入错误：数据对齐
2530	9520	写入错误：数据块
253A	9530	全局数据块不存在
253C	9532	故障信息或功能不存在
253D	9533	系统功能不存在
253E	9534	故障信息或功能块不存在

ID (十六进制)	ID (十进制)	说明
253F	9535	系统块不存在
2575	9589	程序嵌套深度出错
2576	9590	局部数据分配出错
2942	10562	读取错误：输入
2943	10563	写入错误：输出

参见

局部错误处理 (页 470)

7.8.2.4 通信

7.8.2.4 开放式用户通信

7.8.2.4 TSEND_C

描述

TSEND_C 是异步指令且具有以下功能：

- 设置并建立通信连接：
TSEND_C 可设置并建立 TCP 或 ISO-on-TCP 通信连接。设置并建立连接后，CPU 会自动保持和监视该连接。
在参数 CONNECT 中指定的连接描述用于设置通信连接。要建立连接，参数 CONT 的值必须设置为“1”。连接成功建立后，参数 DONE 在一个周期内设置为“1”。
若 CPU 转到 STOP 模式，将终止现有连接并删除所设置的相应连接。必须再次执行 TSEND_C，才能重新设置并建立该连接。
有关可建立的通信连接数的信息，请参考 CPU 的技术规范。
- 通过现有通信连接发送数据：
通过参数 DATA 可指定要发送的区域。这包括要发送数据的地址和长度。在参数 REQ 中检测到上升沿时执行发送作业。使用参数 LEN 指定通过一个发送作业可发送的最大字节数。在发送作业完成前不允许编辑要发送的数据。如果发送作业成功执行，则参数 DONE 将设置为“1”。参数 DONE 的信号状态为“1”并不表示是确认通信伙伴已读取发送数据。
- 终止通信连接：
参数 CONT 设置为“0”时，将终止通信连接。

参数 COM_RST 设置为“1”时，将再次执行 TSEND_C。这会终止现有通信连接并建立新连接。如果再次执行该指令时正在传送数据，可能会导致数据丢失。

参数

参数	声明	数据类型	存储区	说明
REQ	INPUT	BOOL	I、Q、M、D、L	在上升沿启动发送作业。
CONT	INPUT	BOOL	I、Q、M、D、L	控制通信连接： <ul style="list-style-type: none"> 0: 断开通信连接 1: 建立并保持通信连接
LEN	INPUT	UINT	I、Q、M、D、L 或常数	要通过作业发送的最大字节数。
CONNECT	IN_OUT	TCON_Param	D	指向连接描述的指针 请参见： 创建和分配连接参数 (页 218)
DATA	IN_OUT	VARIANT	I、Q、M、D、L	指向发送区的指针，该发送区包含要发送数据的地址和长度。
COM_RST	IN_OUT	BOOL	I、Q、M、D、L	重新启动该指令： <ul style="list-style-type: none"> 0: 不相关 1: 该指令重新启动完成后，将导致现有连接终止并建立一个新连接。
DONE	OUTPUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或仍在执行 1: 作业已完成且未出错
BUSY	OUTPUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或已完成。 1: 作业尚未完成。无法启动新作业。
ERROR	OUTPUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 无错误 1: 出现错误
STATUS	OUTPUT	WORD	I、Q、M、D、L	指令的状态

参数 BUSY、DONE 和 ERROR

可使用参数 BUSY、DONE、ERROR 和 STATUS 检查作业状态。参数 BUSY 指示处理状态。通过参数 DONE 可检查作业是否已成功执行。如果 TSEND_C 执行期间出错，参数 ERROR 将置位。错误信息会在参数 STATUS 中输出。

下表说明了参数 BUSY、DONE 和 ERROR 之间的关系：

BUSY	DONE	ERROR	说明
1	-	-	正在处理作业。
0	1	0	作业已成功完成。

BUSY	DONE	ERROR	说明
0	0	1	由于出错，导致作业结束。在参数 STATUS 中指明出错原因。
0	0	0	未分配新作业。

参数 ERROR 和 STATUS

ERROR	STATUS (W#16#...)	描述
0	0000	作业已完成且未出错
0	7000	无激活的作业处理
0	7001	<ul style="list-style-type: none"> 开始执行作业 建立连接 等待连接伙伴
0	7002	正在发送数据
0	7003	连接已终止
0	7004	连接已建立并受到监视，无激活的作业处理。
1	8085	参数 LEN 的值为 0 或大于最大允许值。
1	8086	参数 CONNECT 超出允许范围。
1	8087	已达到最大连接数；无法建立更多连接。
1	8088	参数 LEN 的值大于 DATA 中指定的区域的长度。 用于接收数据的区域太小。
1	8089	参数 CONNECT 未指向数据块。
1	8091	超出最大嵌套深度。
1	809A	参数 CONNECT 指向的域与连接描述的长度不匹配。
1	809B	连接描述中本地设备的 ID 与 CPU 不匹配。
1	80A1	通信错误： <ul style="list-style-type: none"> 尚未建立指定的连接。 正在终止指定的连接。无法通过此连接进行传送。 正在重新初始化接口。
1	80A3	正在尝试终止不存在的连接。
1	80A7	通信错误：在 TSEND_C 完成前调用了 TDISCON。
1	80B2	参数 CONNECT 指向使用关键字 UNLINKED 生成的数据块。
1	80B3	不一致的参数： <ul style="list-style-type: none"> 连接描述错误 本地端口已经存在于其它连接描述中 连接描述中的 ID 与作为参数指定的 ID 不同

ERROR	STATUS (W#16#...)	描述
1	80B4	使用 ISO-on-TCP 协议 (connection_type = B#16#12) 建立被动连接 (active_est = FALSE) 时，违反了以下一个或两个条件：“local_tsap_id_len >= B#16#02”和/或“local_tsap_id[1] = B#16#E0”。
1	80C3	所有连接资源都在使用。
1	80C4	临时通信错误： <ul style="list-style-type: none"> • 此时无法建立连接。 • 接口当前正在接收新参数。 • 正通过 TDISCON 指令删除已组态连接。
1	8722	CONNECT 参数：源区域无效。DB 中不存在该区域。
1	873A	CONNECT 参数：无法访问连接描述（例如，DB 不存在）。
1	877F	CONNECT 参数：内部错误。

参见

开放式用户通信的基本知识 (页 213)

面向连接协议的工作原理 (页 276)

通信连接的参数 (页 277)

通信块的常规状态信息 (页 921)

7.8.2.4 TRCV_C

描述

TRCV_C 是异步指令且具有以下功能：

- 设置并建立通信连接：

TRCV_C 可设置并建立 TCP 或 ISO-on-TCP 通信连接。设置并建立连接后，CPU 会自动保持和监视该连接。

使用参数 CONNECT 指定的连接描述，可以设置通信连接。要建立连接，参数 CONT 的值必须设置为“1”。连接成功建立后，参数 DONE 在一个周期内设置为“1”。

若 CPU 转到 STOP 模式，将终止现有连接并删除所设置的相应连接。必须再次执行 TRCV_C，才能重新设置并建立该连接。

有关可建立的通信连接数的信息，请参考 CPU 的技术规格。
- 通过现有通信连接接收数据：

如果参数 EN_R 的值设置为“1”，则启用数据接收。接收到的数据将输入到接收区中。根据所用的协议选项，通过参数 LEN 指定接收区长度（如果 LEN <> 0），或者通过参数 DATA 的长度信息来指定（如果 LEN = 0）。

成功接收数据后，参数 DONE 的信号状态为“1”。如果数据传送过程中出错，参数 DONE 将设置为“0”。
- 终止通信连接：

参数 CONT 设置为“0”时，将终止通信连接。

置位参数 COM_RST 时，将再次执行 TRCV_C。这会终止现有通信连接并建立新连接。如果再次执行该指令时正在接收数据，可能会导致数据丢失。

TRCV_C 的接收模式

下表说明了接收数据如何输入到接收区中。

协议选项	接收区中数据的可用性	具有连接描述的 "connection_type"参数	参数 LEN	参数 RCVD_LEN
TCP (特殊模式)	数据立即可用。	B#16#11	0	1 到 x (x: 参数 DATA 定义的长度)
TCP (指定长度的数据接收)	只要接收到的数据达到参数 LEN 指定的数据长度，数据就立即可用。	B#16#11	1 到 8192	与参数 LEN 的值相同
ISO-on-TCP (协议控制的数据传输)	只要接收到的数据达到参数 LEN 指定的数据长度，数据就立即可用。	B#16#12	<ul style="list-style-type: none"> 如果使用了 CP，为 1 到 1452。 如果未使用 CP，为 1 到 8192。 	与参数 LEN 的值相同

TCP (特殊模式)

特殊模式仅在使用 TCP 协议选项时才可用。要设置特殊模式，需要将参数 LEN 的值指定为 0。接收区长度由参数 DATA 中的指针定义。参数 RCVD_LEN 中实际接收的数据长度必须与参数 DATA 中定义的长度相同。最多可接收 8192 个字节。

TCP (指定长度的数据接收)

使用参数 LEN 的值指定数据接收的长度。只要已完全接收到参数 LEN 中指定长度的数据，参数 DATA 中指定的数据在接收区中就立即可用。

ISO-on-TCP (协议控制的数据传输)

使用 ISO-on-TCP 协议选项时，将以协议控制的方式传输数据。

接收区由参数 LEN 和 DATA 定义。

参数

参数	声明	数据类型	存储区	说明
EN_R	IN	BOOL	I、Q、M、D、L	接收使能
CONT	IN	BOOL	I、Q、M、D、L	控制通信连接： <ul style="list-style-type: none"> 0: 断开通信连接 1: 建立并保持通信连接

参数	声明	数据类型	存储区	说明
LEN	IN	UINT	I、Q、M、D、L 或常数	接收区长度 (字节)
CONNECT	IN_OUT	TCON_Param	D	指向连接描述的指针 请参见： 创建和分配连接参数 (页 218)
DATA	IN_OUT	VARIANT	I、Q、M、D、L	指向接收区的指针
COM_RST	IN_OUT	BOOL	I、Q、M、D、L	重新启动该指令： <ul style="list-style-type: none"> 0: 不相关 1: 完成该指令的重新启动，会导致现有连接终止
DONE	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或仍在执行 1: 作业已完成且未出错
BUSY	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或已完成 1: 作业尚未完成。无法启动新作业
ERROR	OUT	BOOL	I、Q、M、D、L	ERROR 状态参数： <ul style="list-style-type: none"> 0: 无错误 1: 出现错误
STATUS	OUT	WORD	I、Q、M、D、L	指令的状态
RCVD_LEN	OUT	WORD	I、Q、M、D、L	实际接收到的数据量 (字节)

参数 BUSY、DONE 和 ERROR

可使用参数 BUSY、DONE、ERROR 和 STATUS 检查作业状态。参数 BUSY 指示处理状态。通过参数 DONE 可检查作业是否已成功执行。如果 TRCV_C 执行期间出错，参数 ERROR 将置位。错误信息会在参数 STATUS 中输出。

下表说明了参数 BUSY、DONE 和 ERROR 之间的关系：

BUSY	DONE	ERROR	说明
1	-	-	正在处理作业。
0	1	0	作业已成功完成。
0	0	1	由于出错，导致作业结束。出错原因在参数 STATUS 中输出。
0	0	0	未分配新作业。

参数 ERROR 和 STATUS

ERROR	STATUS (W#16#...)	描述
0	0000	作业已完成且未出错
0	7000	无激活的作业处理
0	7001	<ul style="list-style-type: none"> 开始执行作业 建立连接 等待连接伙伴
0	7002	正接收数据
0	7003	正在终止连接
0	7004	<ul style="list-style-type: none"> 连接已建立且处于受监视状态 无激活的作业处理
1	8085	参数 LEN 的值为 0 或大于最大允许值。
1	8086	参数 CONNECT 超出允许范围。
1	8087	已达到最大连接数；无法建立更多连接
1	8088	参数 LEN 的值大于 DATA 中指定的区域的长度。 用于接收数据的区域太小。
1	8089	参数 CONNECT 未指向数据块。
1	8091	超出最大嵌套深度。
1	809A	参数 CONNECT 指向的域与连接描述的长度不匹配。
1	809B	连接描述中本地设备的 ID (local_device_id) 与 CPU 不匹配。
1	80A1	通信错误： <ul style="list-style-type: none"> 尚未建立指定的连接。 正在终止指定的连接。无法通过此连接进行传送。 正在重新初始化接口。

参见

开放式用户通信的基本知识 (页 213)

面向连接协议的工作原理 (页 276)

通信连接的参数 (页 277)

通信块的常规状态信息 (页 921)

7.8.2.4 其它

7.8.2.4 TCON

描述

使用 TCON 可设置并建立通信连接。设置并建立连接后，CPU 会自动保持和监视该连接。TCON 是异步指令。

将使用为参数 CONNECT 和 ID 指定的连接数据来设置通信连接。要建立该连接，必须在参数 REQ 中检测到上升沿。如果成功建立连接，参数 DONE 将设置为“1”。

可能的连接数

有关可能的通信连接数的信息，请参考 CPU 的技术规格。

使用 TCP 和 ISO-on-TCP 的连接

两个通信伙伴都调用“TCON”指令来设置并建立通信连接。用户需要在参数中指定哪个伙伴是主动通信端点以及哪个是被动通信端点。

例如，如果连接由于断线或远程通信伙伴而中止，主动伙伴会尝试重新建立组态的连接。用户不必再次调用 TCON。

执行“TDISCON”指令时或 CPU 切换到 STOP 模式后，会终止现有连接并删除所设置的相应连接。要再次设置并建立连接，需重新执行 TCON。

参数

参数	声明	数据类型	存储区	说明
REQ	IN	BOOL	I、Q、M、D、L	在上升沿出现时启动作业以建立 ID 中指定的连接。
ID	IN	CONN_OUC	L、D 或常数	引用已分配的连接。 范围：W#16#0001 到 W#16#0FFF
CONNECT	IN_OUT	TCON_Param	D	指向连接描述的指针 请参见： 创建和分配连接参数 (页 218)
DONE	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或仍在执行 1: 作业已完成且未出错
BUSY	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或已完成。 1: 作业尚未完成。无法启动新作业
ERROR	OUT	BOOL	I、Q、M、D、L	ERROR 状态参数： <ul style="list-style-type: none"> 0: 无错误 1: 出现错误
STATUS	OUT	WORD	I、Q、M、D、L	指令的状态

参数 BUSY、DONE 和 ERROR

可使用参数 BUSY、DONE、ERROR 和 STATUS 检查作业状态。参数 BUSY 指示处理状态。通过参数 DONE 可检查作业是否已成功执行。如果执行 TCON 期间出错，则参数 ERROR 将置位。错误信息会在参数 STATUS 中输出。

下表说明了参数 BUSY、DONE 和 ERROR 之间的关系：

BUSY	DONE	ERROR	描述
1	-	-	正在处理作业。
0	1	0	作业已成功完成。
0	0	1	由于出错，导致作业结束。出错原因在参数 STATUS 中输出。
0	0	0	未分配新作业。

参数 ERROR 和 STATUS

ERROR	STATUS (W#16#...)	说明
0	0000	连接已成功建立
0	7000	无激活的作业处理
0	7001	启动作业执行，建立连接。
0	7002	正在建立连接（与 REQ 无关）
1	8086	参数 ID 超出允许范围
1	8087	已达到最大连接数；无法建立更多连接
1	8089	参数 CONNECT 未指向数据块。
1	809A	参数 CONNECT 指向的域与连接描述的长度不匹配。
1	809B	连接描述中本地设备的 ID 与 CPU 不匹配。
1	80A0	错误代码 W#16#80A1 和 W#16#80A2 对应的组错误
1	80A1	连接或端口已被用户使用。
1	80A2	本地端口或分布式端口已被系统使用。
1	80A3	正尝试重新建立现有连接。
1	80A4	分布式连接端点的 IP 地址无效，即它与本地伙伴的 IP 地址重复。
1	80A7	通信错误：在 TCON 完成前执行了 TDISCON。
1	80B3	不一致的参数分配：错误代码 W#16#80A0 到 W#16#80A2、W#16#80A4、W#16#80B4 到 W#16#80B9 对应的组错误

ERROR	STATUS (W#16#...)	说明
1	80B4	用户违反了以下一个或多个使用 ISO-on-TCP 协议选项 (connection_type = B#16#12) 建立被动连接的条件： <ul style="list-style-type: none"> • local_tsap_id_len >= B#16#02 • local_tsap_id[1] = B#16#E0 • local_tsap_id_len >= B#16#03 时，local_tsap_id[1] 为 ASCII 字符。 • local_tsap_id[1] 为 ASCII 字符且 local_tsap_id_len >= B#16#03。
1	80B6	连接描述数据块的“connection_type”参数存在参数分配错误
1	80B7	连接描述数据块的以下参数之一出错：block_length、local_tsap_id_len、rem_subnet_id_len、rem_staddr_len、rem_tsap_id_len、next_staddr_len
1	80B8	本地连接描述中的参数与参数 ID 不同。
1	80C3	所有连接资源都在使用。
1	80C4	临时通信错误： <ul style="list-style-type: none"> • 此时无法建立连接。 • 接口当前正在接收新参数。 • 正通过 TDISCON 指令删除已组态连接。

参见

开放式用户通信的基本知识 (页 213)

面向连接协议的工作原理 (页 276)

通信连接的参数 (页 277)

通信块的常规状态信息 (页 921)

7.8.2.4 TDISCON

描述

使用 TDISCON 可终止通信连接。在参数 REQ 中检测到上升沿时，即会启动终止通信连接的作业。在参数 ID 中，输入要终止的连接的引用。TDISCON 是异步指令。

执行 TDISCON 后，为 TCON 指定的 ID 不再有效，因此不能再用于发送或接收。

参数

参数	声明	数据类型	存储区	说明
REQ	IN	BOOL	I、Q、M、D、L	在上升沿出现时启动作业以终止 ID 中指定的连接。
ID	IN	CONN_OUC	L、D 或常数	引用通过 TCON 建立的连接。 取值范围：W#16#0001 到 W#16#0FFF

参数	声明	数据类型	存储区	说明
DONE	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或仍在执行 1: 作业已完成且未出错
BUSY	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或已完成 1: 作业尚未完成。无法启动新作业。
ERROR	OUT	BOOL	I、Q、M、D、L	ERROR 状态参数： <ul style="list-style-type: none"> 0: 无错误 1: 出现错误
STATUS	OUT	WORD	I、Q、M、D、L	指令的状态

参数 BUSY、DONE 和 ERROR

可使用参数 BUSY、DONE、ERROR 和 STATUS 检查作业状态。参数 BUSY 指示处理状态。通过参数 DONE 可检查作业是否已成功执行。如果执行 TDISCON 期间出错，则参数 ERROR 将置位。错误信息会在参数 STATUS 中输出。

下表说明了参数 BUSY、DONE 和 ERROR 之间的关系：

BUSY	DONE	ERROR	描述
1	-	-	正在处理作业。
0	1	0	作业已成功完成。
0	0	1	由于出错，导致作业结束。出错原因在参数 STATUS 中输出。
0	0	0	未分配新作业。

参数 ERROR 和 STATUS

ERROR	STATUS (W#16#...)	说明
0	0000	已成功终止连接
0	7000	无激活的作业处理
0	7001	启动作业执行，正在终止连接。
0	7002	正在终止连接（与 REQ 无关）。
1	8086	参数 ID 超出允许的地址范围。
1	80A3	正在尝试终止不存在的连接。
1	80C4	临时通信错误：接口正在接收新参数或正在建立连接。

参见

开放式用户通信的基本知识 (页 213)

面向连接协议的工作原理 (页 276)

通信连接的参数 (页 277)

通信块的常规状态信息 (页 921)

7.8.2.4 TSEND

描述

使用 TSEND 可通过已有的通信连接发送数据。TSEND 是异步指令。

用户使用参数 DATA 指定发送区。这包括要发送数据的地址和长度。在参数 REQ 中检测到上升沿时执行发送作业。使用参数 LEN 指定通过一个发送作业可发送的最大字节数。在发送作业完成前不允许编辑要发送的数据。如果发送作业成功执行，则参数 DONE 将设置为“1”。参数 DONE 的信号状态为“1”并不表示是确认通信伙伴已读出了发送数据。

参数

参数	声明	数据类型	存储区	说明
REQ	IN	BOOL	I、Q、M、D、L	在上升沿启动发送作业。
ID	IN	CONN_OUT C	L、D 或常数	引用通过 TCON 建立的连接。 取值范围：W#16#0001 到 W#16#0FFF
LEN	IN	UINT	I、Q、M、D、L	要通过作业发送的最大字节数。
DATA	IN_OUT	VARIANT	I、Q、M、D、L	指向发送区的指针，该发送区包含要发送数据的地址和长度。该地址引用： <ul style="list-style-type: none"> • 过程映像输入 • 过程映像输出 • 一个存储器位 • 一个数据块
DONE	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 作业尚未启动或仍在执行 • 1: 作业已完成且未出错
BUSY	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 作业尚未启动或已完成 • 1: 作业尚未完成。无法启动新作业。
ERROR	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 无错误 • 1: 出现错误
STATUS	OUT	WORD	I、Q、M、D、L	指令的状态

参数 BUSY、DONE 和 ERROR

可使用参数 BUSY、DONE、ERROR 和 STATUS 检查作业状态。参数 BUSY 指示处理状态。通过参数 DONE 可检查作业是否已成功执行。如果执行 TSEND 期间出错，则参数 ERROR 将置位。错误信息会在参数 STATUS 中输出。

下表说明了参数 BUSY、DONE 和 ERROR 之间的关系：

BUSY	DONE	ERROR	描述
1	-	-	正在处理作业。
0	1	0	作业已成功完成。
0	0	1	由于出错，导致作业结束。在参数 STATUS 中指明出错原因。
0	0	0	未分配新作业。

注意

由于 TSEND 是异步指令，所以需要在参数 DONE 或参数 ERROR 的值变为“1”前，保持发送区中的数据一致。

参数 ERROR 和 STATUS

ERROR	STATUS (W#16#...)	说明
0	0000	发送作业已完成且未出错。
0	7000	无激活的作业处理。
0	7001	启动作业执行，正在发送数据。 处理该作业期间，操作系统访问 DATA 发送区中的数据。
0	7002	作业正在执行（与 REQ 无关）。 处理该作业期间，操作系统访问 DATA 发送区中的数据。
1	8085	参数 LEN 的值为 0 或大于最大允许值。
1	8086	参数 ID 超出允许的地址范围。
1	8088	参数 LEN 的值大于在 DATA 中指定的区域的长度。
1	80A1	通信错误： <ul style="list-style-type: none"> • 尚未建立指定的连接。 • 正在终止指定的连接。无法通过此连接进行传送。 • 正在重新初始化接口。

ERROR	STATUS (W#16#...)	说明
1	80C3	<ul style="list-style-type: none"> 具有该 ID 的块正在一个具有不同优先级的组中处理。 内部缺乏资源。
1	80C4	临时通信错误： <ul style="list-style-type: none"> 此时无法建立与伙伴的连接。 接口正在接收新参数设置或正在建立连接。

参见

开放式用户通信的基本知识 (页 213)

面向连接协议的工作原理 (页 276)

通信连接的参数 (页 277)

通信块的常规状态信息 (页 921)

7.8.2.4 TRCV

描述

使用 TRCV 可通过已有的通信连接接收数据。TRCV 是异步指令。

当参数 EN_R 的值设置为“1”时，启用数据接收。接收到的数据将输入到接收区中。根据所用的协议选项，通过参数 LEN 指定接收区长度（如果 LEN <> 0），或者通过参数 DATA 的长度信息来指定（如果 LEN = 0）。

成功接收数据后，参数 NDR 的值设置为“1”。可在参数 RCVD_LEN 中查询实际接收的数据量。

TRCV 的接收模式

下表说明了接收数据如何输入到接收区中。

协议选项	接收区中数据的可用性	具有连接描述的“connection_type”参数	参数 LEN	参数 RCVD_LEN
TCP (特殊模式)	数据立即可用。	B#16#11	0	1 到 x (x: 参数 DATA 定义的长度)
TCP (指定长度的数据接收)	只要接收到的数据达到参数 LEN 指定的数据长度，数据就立即可用。	B#16#11	1 到 8192	与参数 LEN 的值相同
ISO-on-TCP (协议控制的数据传输)	只要接收到的数据达到参数 LEN 指定的数据长度，数据就立即可用。	B#16#12	<ul style="list-style-type: none"> 如果使用了 CP，为 1 到 1452。 如果未使用 CP，为 1 到 8192。 	与参数 LEN 的值相同

TCP (特殊模式)

特殊模式仅在使用 TCP 协议选项时才可用。要设置特殊模式，需要将参数 LEN 的值指定为 0。接收区长度由参数 DATA 中的指针定义。参数 RCVD_LEN 中实际接收的数据长度必须与参数 DATA 中定义的长度相同。最多可接收 8192 个字节。

TCP (指定长度的数据接收)

使用参数 LEN 的值指定数据接收的长度。只要已完全接收到参数 LEN 中指定长度的数据，参数 DATA 中指定的数据在接收区中就立即可用。

ISO-on-TCP (协议控制的数据传输)

使用 ISO-on-TCP 协议选项时，将以协议控制的方式传输数据。

接收区由参数 LEN 和 DATA 定义。

参数

参数	声明	数据类型	存储区	说明
EN_R	IN	BOOL	I、Q、M、D、L	接收使能
ID	IN	CONN_OUT	L、D 或常数	引用通过 TCON 建立的连接。 取值范围：W#16#0001 到 W#16#0FFF
LEN	IN	UINT	I、Q、M、D、L 或常数	接收区长度 (字节)
DATA	IN_OUT	VARIANT	I、Q、M、D	指向接收区的指针
NDR	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或仍在执行 1: 作业已完成且未出错
BUSY	OUT	BOOL	I、Q、M、D、L	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或已完成 1: 作业尚未完成。无法启动新作业
ERROR	OUT	BOOL	I、Q、M、D、L	ERROR 状态参数： <ul style="list-style-type: none"> 0: 无错误 1: 出现错误
STATUS	OUT	WORD	I、Q、M、D、L	指令的状态
RCVD_LEN	OUT	UINT	I、Q、M、D、L	实际接收到的数据量 (字节)

参数 BUSY、NDR 和 ERROR

可使用参数 BUSY、NDR、ERROR 和 STATUS 检查作业状态。参数 BUSY 指示处理状态。通过参数 NDR 可检查作业是否已成功执行。如果执行 TRCV 期间出错，则参数 ERROR 将置位。错误信息会在参数 STATUS 中输出。

下表说明了参数 BUSY、NDR 和 ERROR 之间的关系：

BUSY	NDR	ERROR	说明
1	-	-	正在处理作业。
0	1	0	作业已成功完成。
0	0	1	由于出错，导致作业结束。 出错原因在参数 STATUS 中输出。
0	0	0	未分配新作业。

注意

由于 TRCV 是异步指令，因此仅当参数 NDR 的值设置为“1”时，接收区中的数据才一致。

参数 ERROR 和 STATUS

ERROR	STATUS (W#16#...)	说明
0	0000	作业已成功执行。 在参数 RCVD_LEN 中输出已接收数据的当前长度。
0	7000	块未做好接收准备。
0	7001	块已做好接收准备，接收作业已激活。
0	7002	中间调用，接收作业正在执行。 注：处理作业期间，数据会写入接收区。这意味着，如果发生错误，就会导致接收区中的数据不一致。
1	8085	<ul style="list-style-type: none"> 参数 LEN 大于最大允许值。 参数 LEN 或 DATA 的值在第一次调用后发生改变。
1	8086	参数 ID 超出允许的地址范围。
1	8088	<ul style="list-style-type: none"> 接收区过小。 参数 LEN 的值大于参数 DATA 指定的接收区的大小。
1	80A1	通信错误： <ul style="list-style-type: none"> 尚未建立指定的连接。 正在终止指定的连接。无法通过该连接执行接收作业。 正在重新初始化连接。
1	80B3	参数分配不一致
1	80C3	<ul style="list-style-type: none"> 具有该 ID 的块正在一个具有不同优先级的组中处理。 内部缺乏资源。
1	80C4	临时通信错误： <ul style="list-style-type: none"> 此时无法建立与伙伴的连接。 接口正在接收新参数设置或正在建立连接。

参见

开放式用户通信的基本知识 (页 213)

面向连接协议的工作原理 (页 276)

通信连接的参数 (页 277)

通信块的常规状态信息 (页 921)

7.8.2.4 点对点**7.8.2.4 PORT_CFG****描述**

使用 PORT_CFG 可动态组态点对点通信端口的通信参数。

在硬件配置中设置端口的初始静态组态。可通过执行“PORT_CFG”指令更改该组态。使用 PORT_CFG 可更改以下通信参数设置：

- 奇偶校验
- 波特率
- 每个字符的位数
- 停止位的数目
- 流控制的类型和属性

“PORT_CFG”指令所做的更改不会永久存储在目标系统中。

可通过电气连接 RS-232 (半双工和全双工) 和 RS-485 (半双工) 传送串行数据。

参数

参数	声明	数据类型	说明
REQ	IN	BOOL	在上升沿激活组态更改
PORT	IN	PORT (UINT)	通信端口的 ID (模块 ID)
PROTOCOL	IN	UINT	传输协议： <ul style="list-style-type: none"> • 0: 点对点通信协议 • 1..n : 用于将来定义特定的传输协议

参数	声明	数据类型	说明
BAUD	IN	UINT	端口的波特率： <ul style="list-style-type: none"> • 1: 300 波特 • 2: 600 波特 • 3: 1200 波特 • 4: 2400 波特 • 5: 4800 波特 • 6: 9600 波特 (默认值) • 7: 19200 波特 • 8: 38400 波特 • 9: 57600 波特 • 10: 76800 波特 • 11: 115200 波特
PARITY	IN	UINT	端口的奇偶校验： <ul style="list-style-type: none"> • 1: 无奇偶校验 (默认值) • 2: 偶校验 • 3: 奇校验 • 4: 传号校验 • 5: 空号校验
DATABITS	IN	UINT	每个字符的位数： <ul style="list-style-type: none"> • 1: 每个字符 8 位 (默认值) • 2: 每个字符 7 位
STOPBITS	IN	UINT	停止位的数目： <ul style="list-style-type: none"> • 1: 1 个停止位 (默认值) • 2: 2 个停止位
FLOWCTRL	IN	UINT	数据流控制： <ul style="list-style-type: none"> • 1: 无 (默认值) • 2: XON/XOFF • 3: 硬件流控制 (RTS 始终激活) • 4: 硬件流控制 (可在传输期间禁用 RTS)
XONCHAR	INPUT	CHAR	指示用作 XON 字符的字符。默认设置是字符 DC1 (11H)。
XOFFCHAR	INPUT	CHAR	指示用作 XOFF 字符的字符。默认设置是字符 DC3 (13H)。
WAITIME	IN	UINT	指定开始传输后 XON 或 CTS 的等待时间。 所指定的值必须大于 0。默认设置是 2000 毫秒。
DONE	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 作业尚未启动或仍在执行 • 1: 作业已完成且未出错
ERROR	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 无错误 • 1: 出现错误

参数	声明	数据类型	说明
STATUS	OUT	WORD	指令的状态

参数 STATUS

STATUS (W#16#...)	说明
80A0	指定的协议无效。
80A1	指定的波特率无效。
80A2	指定的奇偶校验率无效。
80A3	指定的每个字符的位数无效。
80A4	指定的停止位个数无效。
80A5	指定的流控制类型无效。
80A6	参数 WAITTIME 的值不正确 如果启用数据流控制，参数 WAITTIME 的值必须大于 0。

参见

通信块的常规状态信息 (页 921)
点对点通信 (页 249)

7.8.2.4 SEND_CFG

说明

使用 SEND_CFG 可动态组态点对点通信端口的串行传送参数。SEND_CFG 执行后将丢弃所有等待传送的消息。

在硬件配置中设置端口的初始静态组态。可通过执行“SEND_CFG”指令更改该组态。使用 SEND_CFG 可更改以下传送参数设置：

- 激活 RTS 到开始传送之间的时间
- 结束传送到激活 RTS 之间的时间
- 定义中断的位时间数

“SEND_CFG”指令所做的更改不会永久存储在目标系统中。

可通过电气连接 RS-232 (半双工和全双工) 和 RS-485 (半双工) 传送串行数据。

参数

参数	声明	数据类型	说明
REQ	IN	BOOL	在上升沿激活组态更改
PORT	IN	PORT (UINT)	通信端口的 ID (HW ID)
RTSONDLY	IN	UINT	激活 RTS 后到开始传输要经过的时间。 该参数的有效值如下： <ul style="list-style-type: none"> • 0 (默认值) • 0 到 65535 ms (步长为 1 ms) 该参数不适用于 RS-485 模块。
RTSOFFDLY	IN	UINT	传输结束后到禁用 RTS 要经过的时间。 该参数的有效值如下： <ul style="list-style-type: none"> • 0 (默认值) • 0 到 65535 ms (步长为 1 ms) 该参数不适用于 RS-485 模块。
BREAK	IN	UINT	指定中断的位时间数，在消息开始时发送这些位时间数。 默认设置为 12 个位时间。最多可指定 25000 个位时间。
IDLELINE	IN	UINT	指定在消息开始时发送的中断后线路空闲信号的位时间数。 默认设置为 12 个位时间。最多可指定 25000 个位时间。
DONE	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 作业尚未启动或仍在执行 • 1: 作业已完成且未出错
ERROR	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 无错误 • 1: 出现错误
STATUS	OUT	WORD	指令的状态

参数 STATUS

STATUS (W#16#...)	说明
80B0	不允许某个传输中断的组态。
80B1	指定的中断时间超出允许的最多 25000 个位时间。
80B2	指定的线路空闲时间超出允许的最多 25000 个位时间。

参见

通信块的常规状态信息 (页 921)

点对点通信 (页 249)

7.8.2.4 RCV_CFG**描述**

使用 RCV_CFG 可动态组态点对点通信端口的串行接收参数。可使用该指令来组态条件，以指定发送消息的开始和结束。符合这些条件的消息接收可通过“RCV_PTP”指令来启用。

在硬件配置的属性中设置端口的初始静态组态。在用户程序中执行“RCV_CFG”指令可以更改该组态。“RCV_CFG”指令所做的更改不会永久存储在目标系统中。

RCV_CFG 指令执行后将丢弃所有等待传送的消息。

参数

参数	声明	数据类型	说明
REQ	IN	BOOL	在上升沿激活组态更改。
PORT	IN	PORT (UINT)	通信端口的 ID (HW ID)
CONDITIONS	IN	CONDITIONS	用户自定义的数据结构，定义开始和结束条件。
DONE	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 作业尚未启动或仍在执行 • 1: 作业已完成且未出错
ERROR	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 无错误 • 1: 出现错误。
STATUS	OUT	WORD	指令的状态

消息接收的开始和结束条件。

可使用 CONDITIONS 数据类型来定义消息传送的开始和结束条件。开始和结束条件告知接收方消息传送何时完成以及下一次消息传送何时开始。用户可为此定义一个或多个开始条件和结束条件。如果指定多个开始或结束条件，则会通过 OR 逻辑指令来处理这些条件。

如果组态的开始条件适用，则接收方识别到消息开始。以下条件可定义为消息接收的开始条件：

- 开始字符：在出现某个特定字符时识别到消息开始。该字符被存储为消息的第一个字符。在开始字符前收到的所有字符均会被拒绝。
- 任意字符：任何字符都可以定义消息开始。该字符被存储为消息的第一个字符。
- 线路中断：如果接收的数据流中断超过一个字符时间，接收方将识别消息。

- 线路空闲：如果发送传输线路在空闲一段时间（该时间以位时间为单位）后发生诸如接收字符之类的事件，将识别到消息开始。
- 字符串（序列）：当数据流中出现指定的字符序列时，将检测到消息开始。可以指定最多四个字符序列，每个序列最多包含五个字符。

如果组态的结束条件适用，则接收方识别到消息结束。以下条件可定义为消息接收的结束条件：

- 响应超时：如果字符接收超出所选的最长持续时间，将自动识别到消息结束。最长持续时间在参数 RCVTIME 中定义。只要上一次传输完成，且 RCV_PTP 指令启用消息接收，即开始对定义的时间进行倒计时。如果在定义的时间 (RCVTIME) 内没有收到任何字符，则 RCV_PTP 指令将报告错误。
- 消息超时：如果字符接收超出所选的最长持续时间，将自动识别到消息结束。最长持续时间在参数 MSGTIME 中定义。只要收到消息的第一个字符，即开始对定义的时间进行倒计时。
- 字符串内超时：如果接收两个连续字符之间的时间间隔比参数 CHARGAP 的值更长，识别到消息结束。
- 最大长度：在超过参数 MAXLEN 中定义的消息长度时识别到消息结束。
- 读取消息长度 (N+LEN+M)：在达到特定消息长度时识别到消息结束。该长度可通过以下参数值来计算：
 - N：特定字符在消息中的位置，长度域从该位置开始。
 - LENGTHSIZE：长度域大小（字节）
 - LENGTHM：长度域后的结束字符数。计算消息长度时不考虑这些字符。
- 字符串：在接收到定义的字符序列时识别到消息结束。该字符串最多可包含 5 个字符。对于字符串中的每个字符，都可以使用位位置定义在计算中是考虑还是忽略该字符。

CONDITIONS 数据类型

下表给出了 CONDITIONS 数据类型的变量的结构：

参数	数据类型	说明
START	STRUCT	开始条件

参数	数据类型	说明
STARTCOND	UINT	<p>指定开始条件。</p> <p>可将开始条件指定成一个 16 位的十六进制值。开始条件的值可以是：</p> <ul style="list-style-type: none"> • 1: 开始字符 • 2: 任意字符 (默认值) • 4: 线路中断 • 8: 线路空闲 • 16: 字符串 1 • 32: 字符串 2 • 64: 字符串 3 • 128: 字符串 4 <p>还可以在参数 STARTCOND 中定义多个开始条件。为此，需要指定各个条件的值的总和。例如，如果要定义“线路空闲”或“字符串 1”或“字符串 4”作为开始条件，则必须指定值“152”。</p>
IDLETIME	UINT	<p>指定开始接收前的最长线路空闲时间。</p> <p>该参数的有效值如下：</p> <ul style="list-style-type: none"> • 40 个位时间 (默认值) • 0 到 2500 个位时间
STARTCHAR	BYTE	<p>指定开始字符。只有组态的开始条件是“开始字符”时，才会启用该设置。</p> <p>该参数的有效值如下：</p> <ul style="list-style-type: none"> • 02 (STX)：默认设置 • B#16#00 到 B#16#FF
SEQ[1].CTL	BYTE	<p>字符串 1：各字符的控制序列</p> <p>可以使用字符的位位置来定义要考虑或忽略字符串中的哪些字符。要评估字符就必须置位相应的位。</p> <ul style="list-style-type: none"> • 位 0：1 个字符 • 位 1：2 个字符 • 位 2：3 个字符 • 位 3：4 个字符 • 位 4：5 个字符 <p>通过复位相应的位来忽略某个字符。</p>
SEQ[1].STR	CHAR[5]	字符串 1：开始字符 (5 个字符)
SEQ[2].CTL	BYTE	字符串 2：忽略/比较各字符的控制序列
SEQ[2].STR	CHAR[5]	字符串 2：开始字符 (5 个字符)
SEQ[3].CTL	BYTE	字符串 3：忽略/比较各字符的控制序列
SEQ[3].STR	CHAR[5]	字符串 3：开始字符 (5 个字符)
SEQ[4].CTL	BYTE	字符串 4：忽略/比较各字符的控制序列
SEQ[4].STR	CHAR[5]	字符串 4：开始字符 (5 个字符)

参数	数据类型	说明
END	STRUCT	结束条件

参数	数据类型	说明
ENDCOND	UINT	<p>指定结束条件。</p> <p>可将结束条件指定成一个 16 位的十六进制值。结束条件的值可以是：</p> <ul style="list-style-type: none"> • 1: 响应超时 • 2: 消息超时 • 4: 字符串内超时 • 8: 最大长度 • 16: N+LEN+M • 32: 字符串 1 <p>还可以在参数 ENDCOND 中定义多个结束条件。为此，需要指定各个结束条件的值的总和。例如，如果要定义结束条件“最大长度”或“序列 1”，则必须指定值“40”。</p>
MAXLEN	UINT	<p>指定消息中的最大字符数。</p> <p>该参数的有效值如下：</p> <ul style="list-style-type: none"> • 1 个字符（默认值） • 0 到 4132 个字符 <p>只有在参数 ENDCOND 中设置了“最大长度”结束条件时，才会启用该设置。</p>
N	UINT	<p>消息中长度域的偏移量</p> <p>该参数的有效值如下：</p> <ul style="list-style-type: none"> • 0 个字符（默认值） • 0 到 4131 个字符 <p>只有在参数 ENDCOND 中设置了“N+LEN+M”结束条件时，才会启用该设置。</p>
LENGTHSIZE	UINT	<p>长度域大小（字节）</p> <p>该参数的有效值如下：</p> <ul style="list-style-type: none"> • 0 个字节（默认值） • 1 个字节 • 2 个字节 • 4 个字节 <p>只有在参数 ENDCOND 中设置了“N+LEN+M”结束条件时，才会启用该设置。</p>
LENGTHM	UINT	<p>指定跟在长度域后但不包含在消息长度中的结束字符数。</p> <p>该参数的有效值如下：</p> <ul style="list-style-type: none"> • 0 个字符（默认值） • 0 到 255 个字符 <p>只有在参数 ENDCOND 中设置了“N+LEN+M”结束条件时，才会启用该设置。</p>
RCVTIME	UINT	<p>指定字符接收的最长持续时间。</p> <p>该参数的有效值如下：</p> <ul style="list-style-type: none"> • 200 ms（默认值） • 0 到 65535 ms（步长为 1 ms） <p>只有在参数 ENDCOND 中设置了“响应超时”结束条件时，才会启用该设置。</p>
MSGTIME	UINT	<p>指定消息接收的最长持续时间。</p> <p>该参数的有效值如下：</p> <ul style="list-style-type: none"> • 200 ms（默认值）

参数 STATUS

STATUS (W#16#...)	说明
80C0	开始条件出错
80C1	<ul style="list-style-type: none"> 结束条件出错 未定义结束条件
80C2	已启用接收中断
80C3	设置了“最大长度”结束条件时，在参数 MAXLEN 中输入的值等于 0 或大于 4132。
80C4	设置了“N+LEN+M”结束条件时，在参数 N 中输入的值大于 4131。
80C5	设置了“N+LEN+M”结束条件时，在参数 LENGTHSIZE 中输入的值等于 0 或无效。
80C6	设置了“N+LEN+M”结束条件时，在参数 LENGTHHM 中输入的值大于 255。
80C7	设置了“N+LEN+M”结束条件时，计算出的消息长度大于 4132。
80C8	设置了“响应超时”结束条件时，在参数中 RCVTIME 输入的值等于 0。
80C9	设置了“字符串内超时”结束条件时，在参数 CHARGAP 中输入的值等于 0 或大于 2500。
80CA	设置了“线路空闲”结束条件时，在参数 IDLETIME 中输入的值等于 0 或大于 2500。
80CB	尽管设置了“字符串”设置作为结束条件，但字符串的所有字符均被标记为“不相关”。
80CC	尽管设置了“字符串”设置作为开始条件，但字符串的所有字符均被标记为“不相关”。

参见

通信块的常规状态信息 (页 921)

点对点通信 (页 249)

7.8.2.4 SEND_PTP

描述

使用 SEND_PTP 可启动数据传送。“SEND_PTP”指令不执行实际的数据传送。传送缓冲区中的数据将传送到相关通信伙伴。通信伙伴处理实际传送。

参数

参数	声明	数据类型	说明
REQ	IN	BOOL	在上升沿启用请求的传送
PORT	IN	PORT (UINT)	通信端口的 ID (模块 ID)
BUFFER	IN	VARIANT	指向传送缓冲区的起始地址。

参数	声明	数据类型	说明
LENGTH	IN	UINT	传送缓冲区的长度
PTRCL	IN	BOOL	指定所连接的通信伙伴上或扩展模块中，用于可自由编程的通信或用于西门子提供的一些特定协议的传送缓冲区。
DONE	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 作业尚未启动或仍在执行 • 1: 作业已完成且未出错
ERROR	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 无错误 • 1: 出现错误
STATUS	OUT	WORD	指令的状态

参数 STATUS

STATUS (W#16#...)	说明
80D0	传送进行期间收到新的发送请求。
80D1	传送中断，因为在指定的等待时间内未确认 CTS 信号。
80D2	发送请求中断，因为通信伙伴 (DCE) 发出信号表示不愿意接收 (DSR)。
8080	端口号无效。
7000	发送操作未激活。
7001	发送操作正在处理第一个调用。
7002	发送操作正在处理后续调用 (第一个调用之后的查询)。

参见

通信块的常规状态信息 (页 921)

点对点通信 (页 249)

7.8.2.4 RCV_PTP

描述

使用 RCV_PTP 可启用发送消息的接收。必须单独启用每条消息。只有相关通信伙伴确认消息后，发送的数据才会传送到接收区中。

参数

参数	声明	数据类型	说明
EN_R	IN	BOOL	在上升沿启用接收
PORT	IN	PORT (UINT)	通信端口的 ID (HW ID)
BUFFER	IN	VARIANT	指向接收缓冲区的起始地址
LENGTH	OUT	UINT	接收缓冲区中消息的长度
NDR	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 作业尚未启动或仍在执行 1: 作业已完成且未出错
ERROR	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> 0: 无错误 1: 出现错误
STATUS	OUT	WORD	指令的状态

参数 STATUS

STATUS (W#16#....)	说明
80E0	因接收缓冲区已满，消息被终止。
80E1	因出现奇偶校验错误，消息被终止。
80E2	因出现帧错误，消息被终止。
80E3	因出现超限错误，消息被终止。
80E4	因计算的消息长度 (N+LEN+M) 超出接收缓冲区大小，消息被终止。
8080	端口号无效。
0094	因接收到最大字符长度，消息被终止。
0095	因指定时间内未收到完整消息，消息被终止。
0096	因指定时间内未收到下一个字符，消息被终止。
0097	因指定时间内未收到第一个字符，消息被终止。
0098	因已满足“n+len+m”长度条件，消息被终止。
0099	因已满足结束序列，消息被终止。

参见

通信块的常规状态信息 (页 921)

点对点通信 (页 249)

7.8.2.4 RCV_RST

说明

使用 RCV_RST 可删除通信伙伴的接收缓冲区。

参数

参数	声明	类型	说明
REQ	IN	BOOL	在上升沿激活接收缓冲区的删除操作
PORT	IN	UINT	通信端口的 ID (模块 ID)
DONE	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 作业尚未启动或仍在执行 • 1: 作业已完成且未出错
ERROR	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 无错误 • 1: 出现错误
STATUS	OUT	WORD	指令的状态 (有关通信块状态的常规信息在参数 STATUS 中输出。)

参见

通信块的常规状态信息 (页 921)

点对点通信 (页 249)

7.8.2.4 SGN_GET

描述

使用 SGN_GET 可查询 RS-232 通信模块的多个信号的当前状态。

参数

参数	声明	数据类型	说明
REQ	IN	BOOL	在上升沿启用查询
PORT	IN	PORT (UINT)	通信端口的 ID (HW ID)
NDR	OUT	BOOL	如果新数据准备好发送且指令已无错执行，则置位一个周期。
DTR	OUT	BOOL	数据终端就绪，模块就绪
DSR	OUT	BOOL	数据设备就绪，通信伙伴就绪

参数	声明	数据类型	说明
RTS	OUT	BOOL	发送请求，模块已做好发送准备
CTS	OUT	BOOL	允许发送，通信伙伴可从模块接收数据（模块对 RTS = ON 的响应）。
DCD	OUT	BOOL	数据载波检测，接收的信号电平
RING	OUT	BOOL	振铃显示，来电显示
ERROR	OUT	BOOL	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 无错误 • 1: 出现错误
STATUS	OUT	WORD	指令的状态

参数 STATUS

STATUS (W#16#....)	说明
80F0	通信模块是 RS-485 模块且没有信号可用。
80F1	由于启用了 H/W 流控制，因此没有可设置的信号。
80F2	因为模块是 DTE 设备，所以 DSR 无法置位。
80F3	因为模块是 DCE 设备，所以 DTR 无法置位。

参见

通信块的常规状态信息 (页 921)

点对点通信 (页 249)

7.8.2.4 SGN_SET

描述

使用 SGN_SET (设置 RS-232 信号)，用户可设置 RS-232 通信模块输出信号的状态。

参数

参数	声明	数据类型	初始值	说明
REQ	IN	BOOL	FALSE	在上升沿激活动作
PORT	IN	PORT (UINT)	0	通信端口的 ID (模块 ID)

参数	声明	数据类型	初始值	说明
SIGNAL	IN	BYTE	FALSE	指定要设置的信号： <ul style="list-style-type: none"> • 设置 01H = RTS • 设置 02H = DTR • 设置 04H = DSR
RTS	IN	BOOL	FALSE	发送请求，模块已做好发送准备
DTR	IN	BOOL	FALSE	数据终端就绪，模块就绪
DSR	IN	BOOL	FALSE	数据设备就绪（仅适用于 DCE 类型的接口）
DONE	OUT	BOOL	FALSE	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 作业尚未启动或仍在执行 • 1: 作业已完成且未出错
ERROR	OUT	BOOL	FALSE	状态参数，可具有以下值： <ul style="list-style-type: none"> • 0: 无错误 • 1: 出现错误
STATUS	OUTPUT	WORD	0	指令的状态

参数 STATUS

STATUS (W#16#...)	说明
80F0	通信模块是 RS-485 模块且没有信号可用。
80F1	由于启用了 H/W 流控制，因此没有可设置的信号。
80F2	因为模块是 DTE 设备，所以 DSR 无法置位。
80F3	因为模块是 DCE 设备，所以 DTR 无法置位。

参见

通信块的常规状态信息 (页 921)

点对点通信 (页 249)

7.8.2.4 通信块的常规状态信息

有关通信块执行状态的常规信息

下表列出了通信块的参数 STATUS 中可输出的常规信息：

STATUS (W#16#...)	说明
8070	所有内部实例存储器都在使用
8080	为通信端口输入的标识符无效
8081	超时、模块错误、内部错误
8090	消息长度无效、模块无效、消息无效
8091	参数化消息中的版本错误
8092	参数化消息中的记录长度无效

7.8.2.5 中断

7.8.2.5 ATTACH

描述

使用 ATTACH 可为事件分配组织块 (OB)。

在参数 OB_NR 中输入组织块的符号名称或数字名称。然后此组织块将分配给由参数 EVENT 指定的事件。

如果在无错执行 ATTACH 指令之后发生参数 EVENT 中的事件，则会调用由参数 OB_NR 指定的组织块并执行其程序。

通过参数 ADD 可指定应取消还是保留先前对其它事件进行的组织块分配。如果参数 ADD 的值为“0”，则现有分配会被当前分配替换。

参数

参数	声明	数据类型	存储区	说明
OB_NR	IN	OB_ATT	M、D、L	组织块
EVENT	IN	EVENT_ATT	M、D、L	事件
ADD	IN	BOOL	M、D、L	对先前分配的影响： <ul style="list-style-type: none"> • ADD=0 (默认)：该事件将取代先前为此 OB 分配的所有事件。 • ADD=1：该事件将添加到此 OB 的先前事件分配中。
RET_VAL	OUT	INT	M、D、L	指令的状态

参数 RET_VAL

RET_VAL (W#16#....)	描述
0	无错误
8090	OB 不存在
8091	OB 类型错误
8093	事件不存在

7.8.2.5 DETACH**描述**

使用 DETACH，可在运行期间取消对一个或多个事件进行的组织块分配。

如果选择了单个事件，则将取消对该事件进行的 OB 分配。当前存在的所有其它分配仍保持激活状态。

如果未选择任何事件，则当前对事件的所有现有组织块分配都将取消。

在参数 OB_NR 中输入组织块的符号名称或数字名称。然后，将取消把该组织块分配给由 EVENT 参数指定的事件。

参数

参数	声明	数据类型	存储区	说明
OB_NR	IN	OB_ATT	M、D、L	组织块
EVENT	IN	EVENT_ATT	M、D、L	事件
RET_VAL	OUT	INT	M、D、L	指令的状态

参数 RET_VAL

RET_VAL (W#16#....)	描述
0	无错误
1	不存在任何分配 (警告)
8090	OB 不存在
8091	OB 类型错误

RET_VAL (W#16#...)	描述
8093	事件不存在

7.8.2.5 延时中断

7.8.2.5 SRT_DINT

描述

在参数 DTIME 组态的延迟时间过后，可使用 SER_DINT 启动用于调用延迟中断 OB 的延迟中断。延迟时间从使能输入 EN 上生成下降沿时开始算起。

如果延迟时间的减计数中断，则不会执行在参数 OB_NR 中指定的组织块。

精确度

如果没有中断事件延迟调用，则调用“SRT_DINT”指令与启动延时中断 OB 之间的时间最多比分配的延迟时间少一毫秒。

参数

参数	声明	数据类型	存储区	说明
OB_NR	IN	OB_ATT	M、D、L	延迟时间后要执行的 OB 的编号
DTIME	IN	TIME	M、D、L 或常数	延迟时间（1 到 60000 ms） 您可以实现更长的延迟时间，例如在时间延迟中断 OB 中使用计数器。
SIGN	IN	WORD	M、D、L 或常数	注：调用时必须为此参数赋值。但是，该值没有任何意义。
RET_VAL	OUT	INT	M、D、L	指令的状态

参数 RET_VAL

OUT (W#16#...)	说明
0000	无错误
8090	参数 OB_NR 不正确
8091	参数 DTIME 不正确

7.8.2.5 CAN_DINT

描述

可使用 CAN_DINT 取消已启动的延时中断，同时也取消通常会在组态的延迟时间后执行的延时中断 OB 的调用。使用参数 OB_NR 可指定要取消调用的组织块的编号。

参数

参数	声明	数据类型	存储区	说明
OB_NR	IN	OB_ATT	M、D、L	要取消调用的 OB 的编号
RET_VAL	OUT	INT	M、D、L	指令的状态

参数 RET_VAL

OUT (W#16#...)	说明
0000	无错误
8090	参数 OB_NR 不正确

7.8.2.5 异步语句

7.8.2.5 DIS_AIRT

描述

可使用 DIS_AIRT 延迟处理其优先级高于当前组织块优先级的中断 OB。

可在组织块中多次调用 DIS_AIRT。DIS_AIRT 调用由操作系统进行计数。每次执行 DIS_AIRT 都会使处理越来越延迟。要取消延迟，需要执行“EN_AIRT”指令。要取消所有延迟，EN_AIRT 的执行次数必须与 DIS_AIRT 的调用次数相等。

可在 DIS_AIRT 指令的参数 RET_VAL 中查询延迟次数。如果参数 RET_VAL 的值为“0”，则无延迟。

参数

参数	声明	类型	存储区	说明
RET_VAL	OUT	INT	M、D、L	延迟次数

7.8.2.5 EN_AIRT**描述**

发生中断时，可使用 EN_AIRT 启用由“DIS_AIRT”指令延迟的组织块处理。

每次执行 EN_AIRT 都会取消已被操作系统记录的由 DIS_AIRT 调用产生的处理延迟。要取消所有延迟，EN_AIRT 的执行次数必须与 DIS_AIRT 的调用次数相等。例如，如果调用了 DIS_AIRT 五次并因此延迟处理五次，则需要调用“EN_AIRT”指令五次以取消全部五次延迟。

可在“EN_AIRT”指令的参数 RET_VAL 中查询执行 EN_AIRT 后尚未启用的中断延迟数。参数 RET_VAL 的值为“0”表示由 DIS_AIRT 启用的所有延迟均已取消。

参数

参数	声明	类型	存储区	说明
RET_VAL	OUT	INT	M、D、L	组态的延迟次数

7.8.2.6 PID**7.8.2.6 PID_Compact：具有自调节功能的 PID 控制器****说明**

“PID_Compact”指令提供可在自动和手动模式下进行自调节的 PID 控制器。

要求

以恒定的采样时间间隔（最好在循环中断 OB 中）调用“PID_Compact”指令。

参数

参数	声明	数据类型	初始值	说明	
Setpoint	INPUT	REAL	0.0	PID 控制器在自动模式下的设定值	
				在“PID_Compact”调用的巡视窗口中组态要使用输入“Input”还是“Input_PER”。	
Input	INPUT	REAL	0.0	作为实际值源的用户程序变量	
Input_PER	INPUT	WORD	W#16#0	作为实际值源的模拟输入	
ManualEnable	INPUT	BOOL	FALSE	<ul style="list-style-type: none"> 出现 FALSE -> TRUE 沿时选择“手动模式” 出现 TRUE -> FALSE 沿时选择最近激活的工作模式 	
ManualValue	INPUT	REAL	0.0	手动模式下的调节变量	
Reset	INPUT	BOOL	FALSE	重新启动控制器。 下列规则适用于 Reset = TRUE 时： <ul style="list-style-type: none"> “未激活”工作模式 调节变量 = 0 复位控制器的中间值（保留 PID 参数） 	
ScaledInput	OUTPUT	REAL	0.0	已标定的实际值的输出	
				输出“Output”、“Output_PER”和“Output_PWM”可同时使用。	
Output	OUTPUT	REAL	0.0	用于输出调节变量的用户程序变量	
Output_PER	OUTPUT	WORD	W#16#0	用于输出调节变量的模拟输出	
Output_PWM	OUTPUT	BOOL	FALSE	使用脉宽调制输出调节变量的开关输出	
SetpointLimit_H	OUTPUT	BOOL	FALSE	TRUE 达到或超出设定值的绝对上限。在 CPU 中，设定值被限制为已组态的实际值绝对上限。	
SetpointLimit_L	OUTPUT	BOOL	FALSE	TRUE 达到或超出设定值的绝对下限。在 CPU 中，设定值被限制为已组态的实际值绝对下限。	
InputWarning_H	OUTPUT	BOOL	FALSE	TRUE 实际值已达到或超出警告上限。	
InputWarning_L	OUTPUT	BOOL	FALSE	TRUE 实际值已达到或低于警告下限。	
State	OUTPUT	INT	0	PID 控制器的当前工作模式：	
				0	未激活（调节变量设置为 0）
				1	初始启动期间的自调节
				2	工作点中的自调节
				3	自动模式
				4	手动模式

参数	声明	数据类型	初始值	说明	
Error	OUTPUT	DWORD	W#32#0	错误消息	
				0000 0000	没有任何错误。
				> 0000 0000	一个或多个错误处于待决状态。PID 控制器将进入“未激活”模式。请参考“错误消息”以分析激活的错误。

采样时间的监视

“PID_Compact”指令用于测量两次调用之间的时间间隔并评估监视采样时间的结果。在每次模式切换时以及初始启动期间会生成采样时间的平均值。该值用作监视功能的参考并用于块中的计算。监视包括两次调用之间的当前测量时间和定义的控制器的采样时间的平均值。

下列情况下会将 PID 控制器设置为“未激活”工作模式：

- 新平均值 $\geq 1.5 \times$ 原平均值
- 新平均值 $\leq 0.5 \times$ 原平均值
- 当前采样时间 $\geq 2 \times$ 当前平均值或 0

工作模式

以下部分介绍了“PID_Compact”指令的工作模式的影响。

工作模式	说明
不活动	在组态“PID 紧凑型”工艺对象后首次将用户程序下载到 CPU 时，PID 控制器保持在“未激活”工作模式下。在这种情况下，将在调试窗口中执行“初始启动期间的自调节”。 在运行期间，如果发生错误或在调试窗口中单击“控制器停止”(Controller stop) 图标，PID 控制器会切换为“未激活”工作模式。 选择备选工作模式后会确认激活的错误。
初始启动期间的自调节/ 工作点中的自调节	在调试窗口中调用此功能时，将执行“初始启动期间的自调节”或“工作点中的自调节”工作模式。
自动模式	在自动模式下，“PID_Compact”工艺对象会按照指定的参数更正控制回路。满足下列条件时控制器将切换为自动模式： <ul style="list-style-type: none"> • 初始启动期间的自调节成功完成。 • 工作点中的自调节成功完成。 • 如果在“PID 参数”(PID parameters) 组态窗口中选中了“使用手动 PID 参数”(Use manual PID parameters) 复选框：“sRet.i_Mode”变量将设置为 3。
手动模式	如果 PID 控制器工作在手动模式下，则可手动设置调节变量。可通过如下方式选择手动模式： <ul style="list-style-type: none"> • 参数“ManualEnable”的值为 TRUE • 在调试窗口中选“手动调节变量”(Manual manipulated variable) 复选框

参数“Error”的错误消息

Error (W#32#...)	说明
0000 0000	没有任何错误。
0000 0001	实际值超出组态的实际值限制。
0000 0002	参数“Input_PER”中有无效值。 请检查模拟量输入是否有错误。
0000 0004	“工作点中的自调节”期间出错。 实际值的波动无法保持。
0000 0008	启动“初始启动期间的自调节”时出错。 实际值非常接近设定值。 请启动工作点中的自调节。
0000 0010	自调节期间设定值发生改变。
0000 0020	“初始启动期间的自调节”处于自动模式并且在“工作点中的自调节”期间不允许进行。
0000 0040	“工作点中的自调节”期间出错。 设定值非常接近调节变量限制。
0000 0080	调节变量限制尚未正确组态。
0000 0100	导致参数无效的自调节期间出错。
0000 0200	参数“Input”的值无效： <ul style="list-style-type: none"> • 值超出数字范围（值小于 $-1e^{12}$ 或大于 $1e^{12}$） • 值的数字格式无效
0000 0400	参数“Output”的值无效： <ul style="list-style-type: none"> • 值超出数字范围（值小于 $-1e^{12}$ 或大于 $1e^{12}$） • 值的数字格式无效
0000 0800	采样时间错误：在循环程序中调用了“PID_Compact”指令或循环中断的设置已更改。
0000 1000	参数“Setpoint”的值无效： <ul style="list-style-type: none"> • 值超出数字范围（值小于 $-1e^{12}$ 或大于 $1e^{12}$） • 值的数字格式无效

如果多个错误处于待决状态，将通过二进制加法显示错误代码值。例如，显示错误代码 0000 0003 表示错误 0000 0001 和 0000 0002 处于待决状态。

参见

在用户程序中调用“PID_Compact”指令 (页 607)
组态 (页 603)

7.8.2.7 运动控制

7.8.2.7 MC_Power：启用、禁用轴

说明

运动控制语句“MC_Power”可启用或禁用轴。

先决条件

- 已正确组态工艺对象。
- 没有待决的启用-禁止错误。
- 在用户程序中，针对每个轴仅调用一次运动控制语句“MC_Power”。

运动控制命令的覆盖特性

MC_Power 命令无法通过任何其它运动控制命令取消。

Enable = TRUE 时，MC_Power 命令不会取消任何其它运动控制命令。

Enable = FALSE 时，MC_Power 命令将取消此工艺对象上的其它所有运动控制命令。

参数

参数	声明	数据类型	初始值	说明	
Axis	INPUT	TO_Axis_PTO	-	轴工艺对象	
Enable	INPUT	BOOL	FALSE	TRUE	运动控制尝试启用轴。如果没有待决的启用-禁止错误，则启用轴。
				FALSE	根据“StopMode”的参数设置取消所有当前命令。
StopMode	INPUT	INT	0	0	紧急停止 禁用时，轴会以组态的紧急停止减速率进行减速。 轴在变为静止状态后被禁用。
				1	紧急停止 禁用轴时，将会毫无延迟地立即阻止轴。脉冲输出立即停止。
Status	OUTPUT	BOOL	FALSE	轴的使能状态	
				FALSE	无使能信号。轴已禁用 轴不执行任何运动命令且不接受任何新运动命令。 轴未归位。 禁用轴时，在状态更改为 FALSE 之前可能有延迟。
				TRUE	已启用 如果有待决的启用-禁止错误，则不会启用系统。
Busy	OUTPUT	BOOL	FALSE	TRUE	命令正在执行。

参数	声明	数据类型	初始值	说明	
Error	OUTPUT	BOOL	FALSE	FALSE	工艺对象执行完毕且未出错。
				TRUE	命令启动过程出错。命令被拒绝。出错原因可在 "ErrorID" 中找到。有关错误的详细信息，请参考 "ErrorInfo"。
ErrorID	OUTPUT	WORD	0000	输出参数 "Error" 的错误 ID (页 591)	
ErrorInfo	OUTPUT	WORD	0000	输出参数 "ErrorID" 的错误 ID (页 591)	

注意事项

消除错误并进行确认后，通过 Enable = TRUE 重新启用由于错误关闭的轴。

禁用轴的建议步骤

要禁用轴：

1. 停止驱动器。
2. 驱动器停止后禁用轴 (Enable = FALSE)。

参见

错误 ID 和错误信息列表 (页 591)

MC_Reset : 确认错误 (页 931)

MC_Home : 归位轴，设置归位位置 (页 932)

MC_Halt : 停止轴 (页 935)

MC_MoveAbsolute : 轴的绝对定位 (页 936)

MC_MoveRelative : 轴的相对定位 (页 937)

MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)

MC_MoveJog : 在点动模式下移动轴 (页 940)

运动控制指南 (页 576)

创建用户程序 (页 587)

7.8.2.7 MC_Reset : 确认错误

说明

运动控制语句 "MC_Reset" 用于确认需要确认的所有运动控制错误。通过循环上电或将项目数据重新下载到模块可确认致命错误。

如果已消除错误原因，则运动控制语句的参数 Error、ErrorID 和 ErrorInfo 的值会在确认时复位。

先决条件

- 无

运动控制语句的覆盖特性

MC_Reset 命令无法通过任何其它运动控制命令取消。

新的 MC_Reset 命令不会取消任何当前的运动控制命令。

参数

参数	声明	数据类型	初始值	说明
Axis	INPUT	TO_Axis_PTO	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	在上升沿启动命令
Done	OUTPUT	BOOL	FALSE	TRUE 错误已确认。
Busy	OUTPUT	BOOL	FALSE	TRUE 命令正在执行。
Error	OUTPUT	BOOL	FALSE	FALSE 命令启动过程未出错。
				TRUE 命令启动过程出错。命令被拒绝。出错原因可在 "ErrorID" 中找到。有关错误的详细信息，请参考 "ErrorInfo"。
ErrorID	OUTPUT	WORD	0000	输出参数 "Error" 的错误 ID (页 591)
ErrorInfo	OUTPUT	WORD	0000	输出参数 "ErrorID" 的错误 ID (页 591)

参见

错误 ID 和错误信息列表 (页 591)

MC_Power : 启用、禁用轴 (页 930)

MC_Home : 归位轴，设置归位位置 (页 932)

MC_Halt : 停止轴 (页 935)

MC_MoveAbsolute : 轴的绝对定位 (页 936)

MC_MoveRelative : 轴的相对定位 (页 937)

MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)

MC_MoveJog : 在点动模式下移动轴 (页 940)

运动控制指南 (页 576)

创建用户程序 (页 587)

7.8.2.7 MC_Home : 归位轴，设置归位位置

说明

运动控制语句“MC_Home”通过测量系统在控制系统与机械系统之间建立基于位置的关系。轴的绝对定位需要归位。可执行以下类型的归位：

- 主动归位

自动执行归位步骤。

- 被动归位
归位步骤必须由用户执行。
- 直接绝对归位
设置绝对归位位置。
- 直接相对归位
相对于当前位置设置归位位置。

先决条件

- 轴已启用。

运动控制语句的覆盖特性

新的 MC_Home 命令 (Mode = 0、1、2) 会取消当前运行的以下运动控制命令：

- MC_Home 命令 (Mode = 2)

基于位置的运动命令会按照新位置继续。

新的 MC_Home 命令 (Mode = 3) 会取消当前运行的以下运动控制命令：

- MC_Home 命令 (Mode = 0、1、2、3)
- MC_Halt 命令
- MC_MoveAbsolute 命令
- MC_MoveRelative 命令
- MC_MoveVelocity 命令
- MC_MoveJog 命令

参数

参数	声明	数据类型	初始值	说明
Axis	INPUT	TO_Axis_PTO	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	在上升沿启动命令 也可在轴归位后启动该命令。
Position	INPUT	REAL	0.0	<ul style="list-style-type: none"> • 到达归位位置时的绝对位置 (Mode = 2 和 3) • 位置设定值 (Mode = 1) • 位置校正值 (Mode = 2)

参数	声明	数据类型	初始值	说明	
Mode	INPUT	INT	0	归位模式	
				0	直接绝对归位。 新的轴位置为参数 Position 的值。
				1	直接相对归位。 新的轴位置为当前轴位置加上参数 Position 的值。
				2	被动归位。 基于轴组态的归位。参数 Position 的值用作归位位置坐标。
				3	主动参照 基于轴组态的归位步骤。 参数 Position 的值用作归位位置坐标。
Done	OUTPUT	BOOL	FALSE	TRUE	命令已完成
Busy	OUTPUT	BOOL	FALSE	TRUE	命令正在执行。
CommandAborted	OUTPUT	BOOL	FALSE	TRUE	该命令由另一命令取消或由于执行期间出错而取消。
Error	OUTPUT	BOOL	FALSE	FALSE	命令启动过程未出错。
				TRUE	命令启动过程出错。命令被拒绝。出错原因可在 "ErrorID" 中找到。有关错误的详细信息，请参考 "ErrorInfo"。
ErrorID	OUTPUT	WORD	0000	输出参数 "Error" 的错误 ID (页 591)	
ErrorInfo	OUTPUT	WORD	0000	输出参数 "ErrorID" 的错误 ID (页 591)	

注意

在下列情况下轴归位会丢失：

- 通过 "MC_Power" 语句将轴禁用
- 断电后 (断电 -> 上电)
- CPU 重新启动后 (RUN-STOP -> STOP-RUN)

参见

错误 ID 和错误信息列表 (页 591)
 MC_Power : 启用、禁用轴 (页 930)
 MC_Reset : 确认错误 (页 931)
 MC_Halt : 停止轴 (页 935)
 MC_MoveAbsolute : 轴的绝对定位 (页 936)
 MC_MoveRelative : 轴的相对定位 (页 937)

MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)

MC_MoveJog : 在点动模式下移动轴 (页 940)

回原点 (页 578)

组态 - 回原点 (页 585)

运动控制指南 (页 576)

创建用户程序 (页 587)

7.8.2.7 MC_Halt : 停止轴

说明

运动控制语句“MC_Halt”可停止所有运动并使轴停止。未定义停止位置。该命令在轴变为静止状态后完成或被新的运动命令取消。

先决条件

- 轴已启用。

运动控制语句的覆盖特性

新的 MC_Halt 命令可取消当前运行的以下运动控制命令：

- MC_Home 命令 (Mode = 3)
- MC_Halt 命令
- MC_MoveAbsolute 命令
- MC_MoveRelative 命令
- MC_MoveVelocity 命令
- MC_MoveJog 命令

参数

参数	声明	数据类型	初始值	说明
Axis	INPUT	TO_Axis_PTO	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	在上升沿启动命令
Done	OUTPUT	BOOL	FALSE	TRUE 速度达到零
Busy	OUTPUT	BOOL	FALSE	TRUE 命令正在执行。
CommandAborted	OUTPUT	BOOL	FALSE	TRUE 该命令由另一命令取消或由于执行期间出错而取消。
Error	OUTPUT	BOOL	FALSE	FALSE 命令启动过程未出错。
				TRUE 命令启动过程出错。命令被拒绝。出错原因可在 "ErrorID" 中找到。有关错误的详细信息，请参考“ErrorInfo”。
ErrorID	OUTPUT	WORD	0000	输出参数“Error”的错误 ID (页 591)

参数	声明	数据类型	初始值	说明
ErrorInfo	OUTPUT	WORD	0000	输出参数“ErrorID”的错误 ID (页 591)

参见

错误 ID 和错误信息列表 (页 591)
 MC_Power : 启用、禁用轴 (页 930)
 MC_Reset : 确认错误 (页 931)
 MC_Home : 归位轴, 设置归位位置 (页 932)
 MC_MoveAbsolute : 轴的绝对定位 (页 936)
 MC_MoveRelative : 轴的相对定位 (页 937)
 MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)
 MC_MoveJog : 在点动模式下移动轴 (页 940)
 运动控制指南 (页 576)
 创建用户程序 (页 587)

7.8.2.7 MC_MoveAbsolute : 轴的绝对定位

说明

运动控制语句“MC_MoveAbsolute”可启动轴到某个绝对位置的定位运动。该命令在到达目标位置时终止。

先决条件

- 轴已启用。
- 轴已归位。

运动控制语句的覆盖特性

新的 MC_MoveAbsolute 命令可取消当前运行的以下运动控制命令：

- MC_Home 命令 (Mode = 3)
- MC_Halt 命令
- MC_MoveAbsolute 命令
- MC_MoveRelative 命令
- MC_MoveVelocity 命令
- MC_MoveJog 命令

参数

参数	声明	数据类型	初始值	说明
Axis	INPUT	TO_Axis_PTO	-	轴工艺对象

参数	声明	数据类型	初始值	说明
Execute	INPUT	BOOL	FALSE	在上升沿启动命令
Position	INPUT	REAL	0.0	目标位置 (负值或正值)
Velocity	INPUT	REAL	10.0	所需最大速度 由于组态的加速度和减速度以及目标位置的原因，通常都达不到最大速度。 速度必须大于或等于组态的启动/停止速度。
Done	OUTPUT	BOOL	FALSE	TRUE 目标位置已到达
Busy	OUTPUT	BOOL	FALSE	TRUE 命令正在执行。
CommandAborted	OUTPUT	BOOL	FALSE	TRUE 该命令由另一命令取消或由于执行期间出错而取消。
Error	OUTPUT	BOOL	FALSE	FALSE 命令启动过程未出错。
				TRUE 命令启动过程出错。命令被拒绝。出错原因可在 "ErrorID" 中找到。有关错误的详细信息，请参考 "ErrorInfo"。
ErrorID	OUTPUT	WORD	0000	输出参数 "Error" 的错误 ID (页 591)
ErrorInfo	OUTPUT	WORD	0000	输出参数 "ErrorID" 的错误 ID (页 591)

参见

错误 ID 和错误信息列表 (页 591)
 MC_Power : 启用、禁用轴 (页 930)
 MC_Reset : 确认错误 (页 931)
 MC_Home : 归位轴，设置归位位置 (页 932)
 MC_Halt : 停止轴 (页 935)
 MC_MoveRelative : 轴的相对定位 (页 937)
 MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)
 MC_MoveJog : 在点动模式下移动轴 (页 940)
 运动控制指南 (页 576)
 创建用户程序 (页 587)

7.8.2.7 MC_MoveRelative : 轴的相对定位

说明

运动控制语句 "MC_MoveRelative" 可启动相对于起始位置的定位运动。

先决条件

- 轴已启用。

运动控制语句的覆盖特性

新的 MC_MoveRelative 命令可取消当前运行的以下运动控制命令：

- MC_Home 命令 (Mode = 3)
- MC_Halt 命令
- MC_MoveAbsolute 命令
- MC_MoveRelative 命令
- MC_MoveVelocity 命令
- MC_MoveJog 命令

参数

参数	声明	数据类型	初始值	说明
Axis	INPUT	TO_Axis_PTO	-	轴工艺对象
Execute	INPUT	BOOL	FALSE	在上升沿启动命令
Distance	INPUT	REAL	0.0	定位距离 (负值或正值)
Velocity	INPUT	REAL	10.0	所需最大速度 由于组态的加速度和减速度以及目标位置的原因，通常都达不到最大速度。 速度必须大于或等于组态的启动/停止速度。
Done	OUTPUT	BOOL	FALSE	TRUE 目标位置已到达
Busy	OUTPUT	BOOL	FALSE	TRUE 命令正在执行
CommandAborted	OUTPUT	BOOL	FALSE	TRUE 该命令由另一命令取消或由于执行期间出错而取消。
Error	OUTPUT	BOOL	FALSE	FALSE 命令启动过程未出错。
				TRUE 命令启动过程出错。命令被拒绝。出错原因可在 "ErrorID" 中找到。有关错误的详细信息，请参考"ErrorInfo"。
ErrorID	OUTPUT	WORD	0000	输出参数"Error"的错误 ID (页 591)
ErrorInfo	OUTPUT	WORD	0000	输出参数"ErrorID"的错误 ID (页 591)

参见

错误 ID 和错误信息列表 (页 591)

MC_Power : 启用、禁用轴 (页 930)

MC_Reset : 确认错误 (页 931)

MC_Home : 归位轴，设置归位位置 (页 932)

MC_Halt : 停止轴 (页 935)

MC_MoveAbsolute : 轴的绝对定位 (页 936)

MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)

MC_MoveJog : 在点动模式下移动轴 (页 940)
 运动控制指南 (页 576)
 创建用户程序 (页 587)

7.8.2.7 MC_MoveVelocity : 以预设的旋转速度移动轴

说明

运动控制语句“MC_MoveVelocity”可使轴以预设速度移动。

先决条件

- 轴已启用。

运动控制语句的覆盖特性

新的 MC_MoveVelocity 命令可取消当前运行的以下运动控制命令：

- MC_Home 命令 (Mode = 3)
- MC_Halt 命令
- MC_MoveAbsolute 命令
- MC_MoveRelative 命令
- MC_MoveVelocity 命令
- MC_MoveJog 命令

参数

参数	声明	数据类型	初始值	说明	
Axis	INPUT	TO_Axis_PTO	-	轴工艺对象	
Execute	INPUT	BOOL	FALSE	在上升沿启动命令	
Velocity	INPUT	REAL	10.0	轴的最终速度	
				0	允许
				>0 且 < 启动/停止速度	不允许
				>= 启动/停止速度	允许
Direction	INPUT	INT	0	方向命令	
				0	旋转方向与参数“velocity”值的符号一致
				1	正旋转方向
				2	负旋转方向

参数	声明	数据类型	初始值	说明	
Current	INPUT	BOOL	FALSE	保持当前速度	
				FALSE	“保持当前速度”已禁用。
				TRUE	保持当前的速度和方向。不考虑参数 "Velocity" 和 "Direction"。 当轴继续以当前速度运动时，参数 "InVelocity" 返回值 TRUE。
InVelocity	OUTPUT	BOOL	FALSE	TRUE <ul style="list-style-type: none"> "Current" = FALSE : 已达到参数 "Velocity" 中定义的速度并保持该速度。 "Current" = TRUE : 保持当前速度。 	
Busy	OUTPUT	BOOL	FALSE	TRUE 命令正在执行	
CommandAborted	OUTPUT	BOOL	FALSE	TRUE 该命令由另一命令取消或由于执行期间出错而取消。	
Error	OUTPUT	BOOL	FALSE	FALSE 命令启动过程未出错。	
				TRUE 命令启动过程出错。命令被拒绝。出错原因可在 "ErrorID" 中找到。有关错误的详细信息，请参考 "ErrorInfo"。	
ErrorID	OUTPUT	BOOL	0000	输出参数 "Error" 的错误 ID (页 591)	
ErrorInfo	OUTPUT	BOOL	0000	输出参数 "ErrorID" 的错误 ID (页 591)	

参见

- 错误 ID 和错误信息列表 (页 591)
- MC_Power : 启用、禁用轴 (页 930)
- MC_Reset : 确认错误 (页 931)
- MC_Home : 归位轴，设置归位位置 (页 932)
- MC_Halt : 停止轴 (页 935)
- MC_MoveAbsolute : 轴的绝对定位 (页 936)
- MC_MoveRelative : 轴的相对定位 (页 937)
- MC_MoveJog : 在点动模式下移动轴 (页 940)
- 运动控制指南 (页 576)
- 创建用户程序 (页 587)

7.8.2.7 MC_MoveJog : 在点动模式下移动轴

说明

通过运动控制语句“MC_MoveJog”可在点动模式下使用闭环速度控制来移动轴。此运动控制语句可用于测试和调试目的。

先决条件

- 轴已启用。

运动控制语句的覆盖特性

新的 MC_MoveJog 命令可取消当前运行的以下运动控制命令：

- MC_Home 命令 (Mode = 3)
- MC_Halt 命令
- MC_MoveAbsolute 命令
- MC_MoveRelative 命令
- MC_MoveVelocity 命令
- MC_MoveJog 命令

参数

参数	声明	数据类型	初始值	说明	
Axis	INPUT	TO_Axis_PTO	-	轴工艺对象	
JogForward	INPUT	BOOL	FALSE	此参数为 TRUE 时，轴以预设的“Velocity”速度正向移动。	
JogBackward	INPUT	BOOL	FALSE	此参数为 TRUE 时，轴以预设的“Velocity”速度反向移动。	
				如果两个输入参数均为 TRUE，则轴会停止且输出参数“ErrorID”会给出错误消息。	
Velocity	INPUT	REAL	10.0	点动模式的预设速度	
InVelocity	OUTPUT	BOOL	FALSE	已达到参数“Velocity”中定义的速度并保持该速度。	
Busy	OUTPUT	BOOL	FALSE	命令正在执行。	
CommandAborted	OUTPUT	BOOL	FALSE	该命令由另一命令取消或由于执行期间出错而取消。	
Error	OUTPUT	BOOL	FALSE	FALSE	命令启动过程未出错。
				TRUE	命令启动过程出错。命令被拒绝。出错原因可在 "ErrorID" 中找到。有关错误的详细信息，请参考“ErrorInfo”。
ErrorID	OUTPUT	WORD	0000	输出参数“Error”的 错误 ID (页 591)	
ErrorInfo	OUTPUT	WORD	0000	输出参数“ErrorID”的 错误 ID (页 591)	

参见

[错误 ID 和错误信息列表 \(页 591\)](#)
[MC_Power : 启用、禁用轴 \(页 930\)](#)
[MC_Reset : 确认错误 \(页 931\)](#)
[MC_Home : 归位轴，设置归位位置 \(页 932\)](#)

MC_Halt : 停止轴 (页 935)
MC_MoveAbsolute : 轴的绝对定位 (页 936)
MC_MoveRelative : 轴的相对定位 (页 937)
MC_MoveVelocity : 以预设的旋转速度移动轴 (页 939)
 运动控制指南 (页 576)
 创建用户程序 (页 587)

7.8.2.8 脉冲

7.8.2.8 CTRL_PWM

说明

通过“CTRL_PWM”指令，可使用软件启用和禁用 CPU 所支持的脉冲发生器。

注意

脉冲发生器只能在设备配置中进行参数化，而不能通过“CTRL_PWM”指令进行参数化。因此，任何会对 CPU 产生影响的参数更改都必须在 CPU 处于 STOP 模式下时进行。

在输入 PWM 中输入要通过该指令控制的脉冲发生器的硬件 ID。只有在硬件配置中启用指定的脉冲发生器后，才能无错误地执行该指令。

在输入“PWM”中只能指定数据类型为“HW_PWM”的变量。硬件数据类型“HW_PWM”的长度为一个 WORD。

置位该指令的输入 ENABLE 的位时，将启用脉冲发生器。如果 ENABLE 的值为 TRUE，脉冲发生器将生成具有设备配置中所定义属性的脉冲。输入 ENABLE 的位复位或 CPU 切换到 STOP 时，将禁用脉冲发生器并且不再生成脉冲。

只有输入 EN 的信号状态为“1”时，才会执行“CTRL_PWM”指令。

由于在执行 CTRL_PWM 指令时 S7-1200 会启用脉冲发生器，因此 S7-1200 的 BUSY 值始终为 FALSE。

只有使能输入“EN”的信号状态为“1”且指令执行期间没有发生错误时，使能输出 ENO 才会置位。

参数

参数	声明	数据类型	存储区	说明
PWM	IN	HW_PWM	L、D 或常数	脉冲发生器的硬件标识 (HW ID)
ENABLE	IN	BOOL	I、Q、M、L、D 或常数	ENABLE = TRUE 时启用脉冲发生器，ENABLE = FALSE 时禁用脉冲发生器。

参数	声明	数据类型	存储区	说明
BUSY	OUT	BOOL	I、Q、M、L、D	处理状态
STATUS	OUT	BOOL	I、Q、M、L、D	指令的状态

参数 STATUS

错误代码 (十六进制)	说明
0	无错误
80A1	脉冲发生器的硬件 ID 无效

可视化过程

8.1 移植

8.1.1 移植

简介

在 WinCC V10.5 中可继续使用 WinCC flexible 2008 的项目。可以移植“精简面板”型的 HMI 设备。对于其它的 HMI 设备，必须首先在 WinCC flexible 2008 中更改 HMI 设备类型。

不能将项目直接从 WinCC flexible 2008 SP1 移植到 WinCC V10.5。如果要在 WinCC V10.5 中继续使用此类项目，则必须首先将其移植到 WinCC flexible 2008。

ProTool Pro 和早期 WinCC flexible 版本的项目不能直接移植到 WinCC V10.5。如果要在 WinCC V10.5 中继续使用此类项目，需要首先将其移植到 WinCC flexible 2008，然后更改 HMI 设备类型。

在 WinCC V10.5 中不能继续使用 WinCC Classic 的项目。

参见

移植基础知识 (页 945)

移植期间的对象支持 (页 946)

移植 WinCC flexible 2008 项目 (页 949)

数据类型的移植 (页 951)

8.1.2 移植基础知识

简介

请注意，在从 WinCC flexible 2008 移植项目时，WinCC 10.5 仅支持以下 HMI 设备类型：

- KTP400 Basic mono PN
- KTP400 Basic mono PN Portrait
- KTP600 Basic mono PN

- KTP600 Basic mono PN Portrait
- KTP600 Basic color PN
- KTP600 Basic color PN Portrait
- KTP1000 Basic PN
- TP1500 Basic PN

WinCC V10.5 仅支持这些 HMI 设备类型提供的功能。

仅 SIMATIC S7 1200 通信驱动程序可用于 WinCC 10.5。因此，不会移植 SIMATIC S7 300/400 和 SIMATIC S7 200 通信驱动程序。

由于设备和通信驱动程序的选择限制而不被移植的其它功能，会在以下章节中进行说明。

移植前的调整

如果所移植项目中的 HMI 设备发生更改，则移植前需要重新编译项目。编译过程将调整画面和画面元素的大小。

移植 STEP 7 中集成的项目

STEP 7 中集成的项目不能直接移植到 WinCC。要移植集成项目，必须先将其从集成中取出。先在 WinCC flexible 中打开项目，然后选择菜单命令“项目”>“从 STEP 7 复制”。WinCC flexible 项目将会从集成中取出。然后移植 WinCC flexible 项目。

参见

移植 (页 945)

8.1.3 移植期间的对象支持

简介

在从 WinCC flexible 2008 移植项目期间，所有可用且可映射到新项目环境的对象类型和函数会被完全移植。

如果对象可以移植，但其引用的对象或功能不能移植，则将移植对象本身。其引用会在移植期间丢失。

实例：

已在 WinCC flexible 2008 中创建了一个项目，要将该项目同时用于 MP 277 和 KTP 1000 类型的目标设备。切换设备会启用/禁用特定设备类型所支持的功能。

用于 MP 277 HMI 设备的项目包含一个 VB 脚本，该脚本可通过在运行系统中单击按钮启动。当针对 KTP 1000 HMI 设备编译该项目时，会获得有关无效功能的警告。

如果要将该项目移植到 WinCC V10.5，则需要设置 KTP 1000 HMI 设备。在随后的移植过程中，不会移植 VB 脚本。对 VB 脚本的引用将丢失。按钮被移植。

支持的对象类型

支持移植以下对象类型：

- 动画
- 调度程序
- 用户管理
- 画面
- 画面模板
- 数据类型
由于不会移植连接，因此会将外部变量的数据类型映射到内部变量的数据类型。更多信息，请参见“[数据类型的移植 \(页 951\)](#)”。
- 事件
- 函数列表
- 图形列表
- 显示和操作元素
受支持的 HMI 设备上可用的所有显示和操作元素都可以移植。
- 报警
- 报警类别
- 报警组
- 项目语言
- 配方
- 运行系统语言
- 运行系统脚本
- 系统函数
- 文本
- 文本列表
- 变量
- 周期 (移植不支持用户定义的周期)。

下载到 HMI 设备的设置

以太网接口的设置包括在移植中。会移植最近用于下载的计算机名称或 IP 地址。选择“下载到设备时使用设备地址”选项，以便在移植后可将项目下载到 HMI 设备。选择此选项时，会使用当前连接的 HMI 设备的 IP 地址。因此下载项目时无需更改目标地址。

还会移植以下用于下载的选项：

- 覆盖用户管理
- 覆盖配方数据

如果在 WinCC flexible 中未进行传送便更改了 HMI 设备类型，则需要更新上载到 WinCC 的设置。在项目树中选择 HMI 设备并从快捷菜单中选择“属性”。会打开 HMI 设备的“属性”对话框。选择“加载”组并验证界面设置。单击“确定”，确认设置。

移植后的限制和调整

在移植支持的对象类型的过程中，会有一些限制。

变量的移植

将不会移植到 PLC 的连接，因为 WinCC flexible 2008 和 WinCC V10.5 中的可组态控件不相同。移植期间，外部变量将会映射到内部变量。移植和组态连接后，需将这些内部变量与 PLC 变量相连接。

[数据类型的移植 \(页 951\)](#) 中提供了有关移植外部变量的数据类型的其它信息。

变量、画面和画面对象的名称

在 WinCC flexible 2008 中，保存在不同文件夹中的变量可能具有相同的名称。在 WinCC 中，已组态的 HMI 设备上的变量名称必须是唯一的。这意味着在移植期间，会对来自不同文件夹而名称相同的变量进行重命名。

基于以下方案进行重命名：

移植之前	移植之后
Folder_1/Tag_1	Folder_1/Tag_1
Folder_1/Tag_2	Folder_1/Tag_2
Folder_2/Tag_1	Folder_2/Tag_1#Mig1
Folder_2/Tag_2	Folder_2/Tag_2#Mig1
Folder_3/Tag_1	Folder_3/Tag_1#Mig2
Folder_3/Tag_2	Folder_3/Tag_2#Mig2

根据此方案会对画面和画面对象进行重命名。

对象位置超出有效范围

在 WinCC flexible 中用于指定画面中对象位置的取值范围，比 WinCC 中的范围大。如果用于指定显示对象或操作员控件位置的值超出有效的取值范围，则移植会将其设置为最大或最小值。

报警组的移植

移植过程将只移植实际使用的报警组。

ID 为 1-31 的报警组将以 1:1 移植。而对于 ID > 31 的报警组，其 ID 在移植期间会根据移植个数逐步递增。这意味着，对于 ID > 31 的报警组，将报警组名称分配给 ID 时，移植后 ID 可能会发生改变。

还应注意以下几点：

移植同名的报警组时，要移植的第一个报警组的名称将保持不变。所有其它同名报警组的名称将带有连续数字形式的扩展。

对已删除对象的引用

不会移植对已删除对象的引用。如果在移植期间检测到对已删除对象的引用，则会将其记录下来。

不支持的对象类型

移植不支持以下对象类型：

- 连接
- 区域指针
- 对已删除对象的引用
- 库
- 字典
- 更改日志
- 项目版本
- 用户定义的周期

以后可使用 SIMATIC S7 1200 通信驱动程序，在 WinCC V10.5 中组态区域指针和连接。

参见

数据类型的移植 (页 951)

移植 (页 945)

8.1.4 移植 WinCC flexible 2008 项目

简介

移植项目时，WinCC flexible 2008 项目数据会载入 WinCC 10.5 的新项目中。因此，项目移植过程会自动创建新项目。不能向现有项目进行移植。

移植过程可在起始视图和项目视图中启动。

只能在新启动的 TIA Portal 中进行移植。移植后保存项目，并重新启动 TIA Portal。

要求

- 在起始视图中。
- 未打开项目。
- WinCC flexible 2008 项目可用。
- 项目不能在 WinCC flexible 2008 中打开。

步骤

移植项目的步骤如下：

1. 选择“开始”>“移植项目”操作。



2. 在“源路径”框中，浏览到要移植的项目。



3. 选择 WinCC flexible 项目文件“*.hmi”。
4. 必要时更改要创建的项目的信息。例如，更改项目名称或项目路径。要移植的数据会在新项目中创建。
5. 单击“移植”。
 - 创建新项目并启动数据移植：
 - 移植进度即显示在移植窗口中。
 - 如果移植期间出现错误，则输出窗口中会出现报警。
6. 移植完毕后，保存项目。

移植完成时，对于每个移植的 HMI 设备，您都会在项目树中找到新创建的设备。这些设备包含移植的数据，例如画面、报警和变量。

项目移植成功后，将出现报告移植成功的通知。

打开移植日志

有关移植的信息存储在日志文件中。阅读此文件的步骤如下：

1. 选择项目。
2. 打开“属性”菜单。
3. 单击“项目历史”。
4. 单击日志文件。

参见

移植 (页 945)

8.1.5 数据类型的移植

简介

下表介绍从 WinCC flexible 2008 中的数据类型到 WinCC 中的数据类型的映射。

数据类型的移植

移植期间内部数据类型按如下方式进行映射：

WinCC flexible 2008 的内部数据类型	WinCC 的内部数据类型
Bool	Bool
Char	SByte
Byte	UByte
Int	Short
UInt	UShort
Long	Long
ULong	ULong
Float	Float
Double	Double
String	WString
DateTime	DateTime

S7 300/400 的数据类型按如下方式映射到内部数据类型：

S7 300/400 数据类型	WinCC 的内部数据类型
Bool	Bool

S7 300/400 数据类型	WinCC 的内部数据类型
Char	SByte
Byte	UByte
Int	Short
Word	UShort
DInt	Long
DWord	ULong
Real	Float
Double	Double
String	WString
StringChar	WString
Time	Long
Timer	ULong
Date and Time	DateTime
Date	DateTime
Time of Day	DateTime
Counter	UShort

S7 200 的数据类型按如下方式映射到内部数据类型：

S7 200 数据类型	WinCC 的内部数据类型
Bool	Bool
Char	SByte
Byte	UByte
Int	Short
Word	UShort
DInt	Long
DWord	ULong
Real	Float
Double	Double
StringChar	WString
Timer	ULong

参见

移植 (页 945)

8.2 使用画面

8.2.1 基础知识

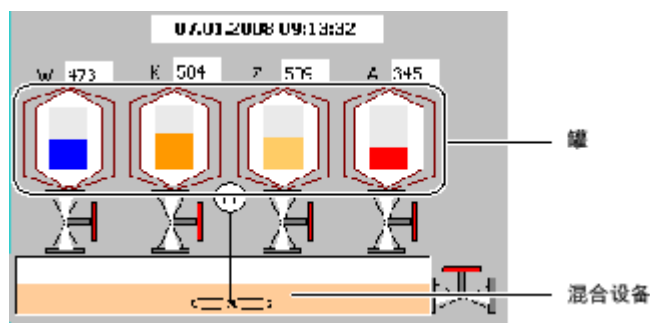
8.2.1.1 画面基础知识

简介

在 WinCC 中，可以创建操作员用来控制和监视机器设备和工厂的画面。创建画面时，所包含的对象模板将在显示过程、创建设备图像和定义过程值方面为您提供支持。

应用实例

下图显示了在 WinCC 中创建的一个画面。利用此画面，操作员可以操作和监视果汁生产系统的混合设备。果汁主要成分由各个罐提供给混合单元。此画面会指示各个罐的填充量。



画面设计

将需要用其表示过程的对象插入到画面。对该对象进行组态使之符合过程要求。

画面可以包含静态和动态元素。

- 静态元素 (例如文本或图形对象) 在运行时不改变它们的状态。此混合设备实例中所示的罐标签 (W、K、Z、A) 是静态元素。
- 动态元素根据过程改变它们的状态。通过下列方式显示当前过程值：
 - 显示从 PLC 的存储器中输出
 - 以字母数字、趋势图和棒图的形式显示 HMI 设备存储器中输出的过程值

HMI 设备上的输入域也认作是动态对象。混合设备实例中的罐填充量值是动态对象。

通过变量可以在控制器和 HMI 设备之间交换过程值和操作员输入值。

画面属性

画面布局由正在组态的 HMI 设备的特征确定。它对应于该设备用户界面的布局。诸如画面分辨率、字体和颜色等的画面属性也由所选 HMI 设备的特征确定。如果设定的 HMI 设备有功能键，则此画面将显示这些功能键。

功能键是 HMI 设备上的键。在 WinCC 中可分配一或多个功能。当操作员在 HMI 设备上按下相关键时，会触发这些功能。

可以为功能键分配全局或局部功能。

- 全局功能键始终触发同样的操作，而与当前显示的画面无关。
- 局部分配的功能键会触发不同的操作，这取决于操作员站上当前显示的画面。这种分配只适用于已在其中定义了功能键的画面。

打开画面

要使操作员能在运行系统中调用画面，必须在操作过程中集成各组态的画面。可以使用不同的方法来组态这些功能：

- 使用“画面”编辑器组态用于打开其它画面的按钮和功能键。
- 使用“全局画面”编辑器组态全局分配的功能键。

参见

任务卡 (页 955)

步骤 (页 959)

有关使用模板的基础知识 (页 963)

可用于特定 HMI 设备的画面 (页 954)

对象概述 (页 969)

关于文本列表的基础知识 (页 1005)

图形列表的基本原理 (页 1012)

使用功能键 (页 1028)

有关使用层的基础知识 (页 1038)

关于库的基础知识 (页 1044)

棒图 (页 1055)

画面浏览的基础知识 (页 1078)

8.2.1.2 可用于特定 HMI 设备的画面

简介

HMI 设备的功能决定着 WinCC 中的项目可视化和编辑器的功能范围。

创建项目时，为项目选择相应的 HMI 设备。使用“项目”窗口更改 HMI 设备类型或添加更多设备。

下列画面属性由所选 HMI 的功能确定：

- 布局
- 屏幕分辨率

- 颜色深度
- 字体
- 可用的对象

设备布局

画面中的设备布局组成您所组态的 HMI 设备的图像。例如，画面的设备布局显示 HMI 设备上所有可用的功能键。

颜色深度

可以将颜色分配给画面对象。可能的颜色数由所选 HMI 设备支持的颜色深度和特定颜色决定。

字体

在包含静态或动态文本的所有画面对象中，其文本的外观都是可以自定义的。例如，可以分别标识画面中各个文本的优先级。例如，选择字体、字体样式和大小，并设置诸如下划线等附加效果。

哪些字体可用取决于所选的 HMI 设备。所选的字体确定哪些字体属性可用。



文本标记（如字形）和效果的设置始终针对画面对象的整个文本而言。就是说，例如，可以以粗体格式显示整个标题，但是不能以粗体格式显示标题的单个字符或单词。

可用的对象

部分画面对象并不能为所有的 HMI 设备全局组态。这些画面对象不会显示在工具箱中。例如，对于 TP 170 触摸面板单元，无法组态任何按钮。

参见

画面基础知识 (页 953)

8.2.1.3 元素和基本设置

8.2.1.3 任务卡

简介

在“画面”编辑器中提供以下任务卡：

- 工具：显示对象和操作对象

- 动画：动态组态的模板
- 布局：帮助进行显示方面的自定义操作
- 库：管理项目库和全局库

注意
精简面板



“动画”任务卡不可用于精简面板。

工具

“工具”任务卡在不同的窗格中包含以下对象：

- 基本对象
- 元素
- 控件
- 用户控件（可选）
- 图形

通过拖放操作将对象从窗格插入画面。可供选择的对象由所组态的 HMI 设备的特征决定。以下图标用于更改显示模式：

图标	含义
	以列表形式显示对象。
	以图形形式显示对象。

动画

“动画”任务卡的窗格中包含可以实现画面对象动态化的可行方法：

- 移动：对角线、直接、水平、垂直
- 显示：外观、可见性
- 其它：可操作性

通过拖放操作将动画从“移动”、“显示”和“其它”窗格添加到画面对象。

布局

“布局”任务卡包含以下用于显示对象和元素的窗格：

- 缩放：用于选择详细视图
- 层：管理画面的对象。它们以树形视图显示，包含有关可见性和活动层的信息。
- 网格：指定对象是否与网格对齐，并设置网格的网格大小。
- 超出范围的对象：可见区域外的对象显示名称、位置和类型。

库

“库”任务卡分别以单独的窗格显示以下库：

- 项目库：项目库与项目一起存储。
- 全局库：全局库存储在组态 PC 上指定路径的单独文件中。

参见

画面基础知识 (页 953)


移动视图 (页 957)

缩放视图 (页 958)

8.2.1.3 移动视图

简介

以下选项可使在工作区域中仅显示整个画面的一部分：


- 使用“画面”编辑器的
图标。
图标。
- 使用“布局”任务卡的“缩放”选项板中整个画面的缩微视图。

要求

- 画面打开。
- 视图只显示一个画面区域。

步骤

移动视图的步骤如下：

1. 单击工作区右下角的
图标，并按下鼠标左键。
会显示整个画面的缩微视图。一个橙色的框显示了当前所选的区域。
2. 按住鼠标按钮并将框拖动到想要的区域。

注意

将画面对象从可视区拖到当前隐藏的区域时，画面就会滚动。

参见

任务卡 (页 955)

8.2.1.3 缩放视图

简介

要更清晰地查看小画面区域的细节，可以使用缩放工具在工作区域中放大画面。最大缩放量可达 800%。

可提供下列功能：

- 借助选择边框进行自由缩放。
- 输入缩放因子。
- 使用“布局”任务卡的“缩放”对象组中的



和



图标可逐渐放大。

要求

画面打开。

步骤

缩放视图的步骤如下：

1. 单击

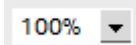


工具栏按钮。

2. 用鼠标在画面上绘制一个选择框。

放开鼠标按钮后，选择框所围起的区域会缩放到恰好能放入工作区的程度。

或者，也可以在“缩放”



工具栏中直接输入以 [%] 为单位的缩放比率。

此工具栏还包含



和



图标。这些图标用于逐步放大或缩小视图。

或者，也可以使用“布局”任务卡更改画面的视图。

结果

所选的画面区域被放大。

参见

任务卡 (页 955)

8.2.1.4 使用画面

8.2.1.4 步骤

步骤

要创建画面，需要进行下列初始步骤：

- 规划过程可视化的结构：画面的数量及其体系结构。

实例：子过程在单独的画面中可视化，再合并到主画面中。

- 定义画面浏览控制策略。
- 调整模板。

在 WinCC 中存储的用于所选 HMI 设备的模板适用于所有项目画面。使用功能键，可在模板对象中执行集中组态并为 HMI 设备全局分配功能键。对于某些 HMI 设备，也可以将集成到每个画面的对象存储在固定窗口中。针对每个 HMI 设备都可以创建和管理自己的多个模板。

- 创建画面。使用下列选项进行有效的画面创建：

- 在“画面浏览”编辑器中创建画面结构
- 使用库
- 使用层

注意

精简面板

“画面浏览”编辑器不可用于精简面板。

参见

画面基础知识 (页 953)

创建新画面 (页 959)

管理画面 (页 961)

定义项目的起始画面 (页 962)

8.2.1.4 创建新画面

简介

可通过以下两种方法设置画面：

- 不使用模板

所有更改仅适用于当前画面。

- 基于模板

在模板中，对使用此模板的所有画面集中执行所有更改。对于具有功能键的 HMI 设备，可将模板中的功能键集中分配给使用给定模板的所有画面。在“全局画面”编辑器中全局分配的功能键适用于 HMI 设备的所有画面。

有关同时使用模板和全局元素的可能性，将在“模板使用的基础知识”一章中介绍。

要求

- 已创建项目。
- 打开巡视窗口。

步骤

创建画面的步骤如下：

1. 在项目树中双击“添加画面”。
画面在项目中生成并出现在视图中。画面属性即显示在巡视窗口中。
2. 输入有意义的画面名称。
3. 在巡视窗口中组态画面属性：
 - 指定画面是否基于模板以及基于哪个模板。
 - 设置“背景色”和“画面号”。
 - 在“信息文本”下指定说明文本。
 - 在工程系统的“层”下指定要显示的层。
 - 在“动画”下指定动态画面更新。
 - 选择“事件”，以定义在调用和退出画面或在其他事件时要在运行系统中执行哪些功能。

注意

HMI 设备相关性

不是所有 HMI 设备都支持画面的“可见性”动画。

结果

在项目中创建了画面。现在可以从工具箱窗口添加对象和控制元素。

参见

步骤 (页 959)

8.2.1.4 管理画面

简介

可以将项目中的画面移动到其它组，也可以复制、重命名和删除画面。

将画面移动到组中

WinCC 允许创建多个画面，并将您的画面划归到这些画面中。

将画面移动到组中的步骤如下：

1. 选择项目树中的“画面”文件夹。
2. 从快捷菜单中选择“添加组”命令。
名为“Group_x”的文件夹随即插入。
3. 在项目树中选择画面。
4. 将画面拖放到所需的组中。
画面即移动到该组中。

复制画面

在 WinCC 中，可以剪切画面，然后将其粘贴到相同的或不同的组中。

复制画面的步骤如下：

1. 在项目树中选择画面。
2. 在快捷菜单中选择“复制”命令将画面复制到剪贴板。
3. 在项目树中，选择画面插入位置。
4. 从快捷菜单中选择“粘贴”来插入画面。
画面的副本随即插入。副本的名称由原画面名称和追加的连续编号组成。

或者，也可以在按住 <Ctrl> 键的同时将画面拖动到所需位置。

注意

如果复制的画面带有多个设备和项目的互连模板，则也会复制该模板。不使用任何现有匹配模板。这一点对通过拖放动作复制画面尤其如此。

重命名画面

重命名画面的步骤如下：

1. 在项目树中选择画面。
2. 从快捷菜单中选择“重命名”。
3. 输入新名称。
不要使用特殊字符 ?、"、/、\、*、<、>。
4. 按下 <ENTER>。

也可以使用 <F2> 功能键来重命名画面。

删除画面

删除画面的步骤如下：

1. 在项目树中选择画面。
2. 从快捷菜单中选择“删除”。

画面及其所有对象均从当前项目删除。

参见

步骤 (页 959)

8.2.1.4 定义项目的起始画面

简介

起始画面是在运行系统中启动项目时打开的初始画面。可为每个 HMI 设备定义不同的起始画面。操作员可从此起始画面开始调用其它画面。

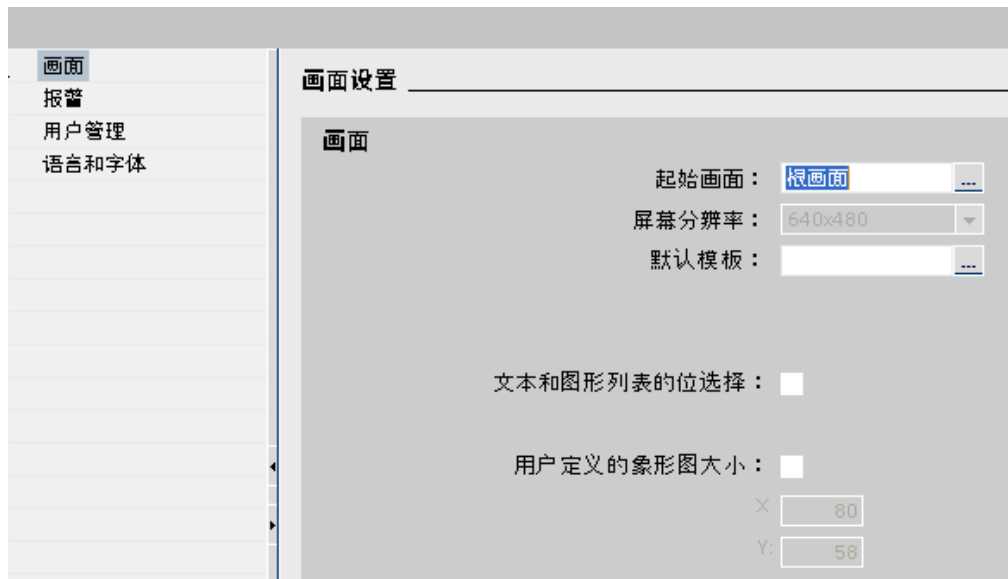
要求

项目包含要用作起始画面的画面。

在“运行系统设置”编辑器中定义起始画面

步骤如下：

1. 在项目树中，双击“运行系统设置”。
已打开“运行系统设置”编辑器。
2. 双击“画面”。



3. 在“起始画面”下选择所需画面。

结果

启动运行系统时，HMI 上会打开该起始画面。

参见

步骤 (页 959)

8.2.1.5 使用模板

8.2.1.5 有关使用模板的基础知识

模板和全局画面

每个 HMI 设备在项目中都有一个模板，将设备的属性考虑进来。在添加画面时会用到模板。在模板中，可组态将在基于此模板的所有画面中显示的对象。

请注意以下规则：

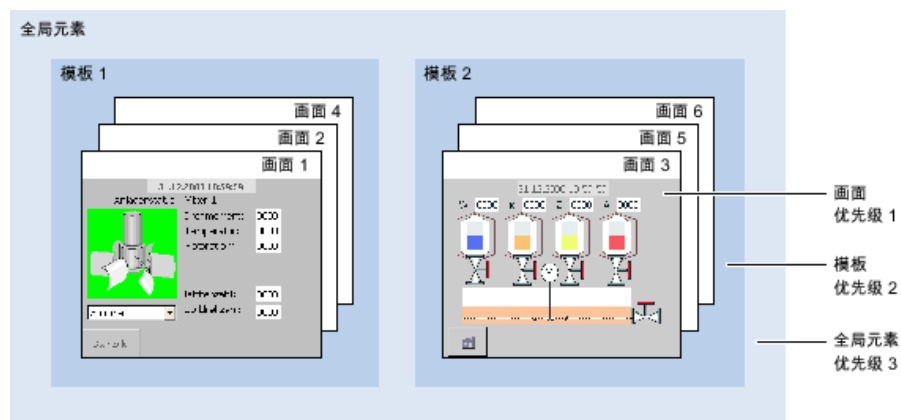
- 一个画面只能基于一个模板。
- 一个设备可以创建几个模板。

无论使用哪个模板，都可为 HMI 设备指定用于所有画面的全局元素。可作为全局对象的“报警窗口”和“报警指示器”对象在“全局画面”编辑器中组态。

以下规则适用：

- 画面中的组态具有优先级 1，优先于模板的组态。
- 模板中的组态具有优先级 2，优先于全局画面中对象的组态。
- 全局画面中对象的组态具有优先级 3。

下图显示组态的优先级顺序：



模板的对象

在模板中可指定将应用于基于此模板的所有画面的以下功能和对象：

- 操作员控件：可以将也用于画面的相同操作员控件插入模板。
- 功能键的分配：使用功能键在模板中为 HMI 设备局部分配功能键。此分配会覆盖可能的全局分配。
- 固定窗口：某些设备（如 MP 370）支持在画面的上部分显示适用于所有画面的固定窗口。和模板相比，固定窗口独自占据了画面的一个区域。

应用实例

- 要将“ActivateScreen”功能分配给模板中的功能键。操作员使用此键在运行系统中切换到另一画面。此组态适用于基于该模板的所有画面。
- 可将包含贵公司标志的图形添加到此模板中。该标志将出现在基于此模板的所有画面中。

注意

如果模板中的对象与画面中的对象具有相同的位置，则模板对象被覆盖。

固定窗口

注意

可用于特定设备

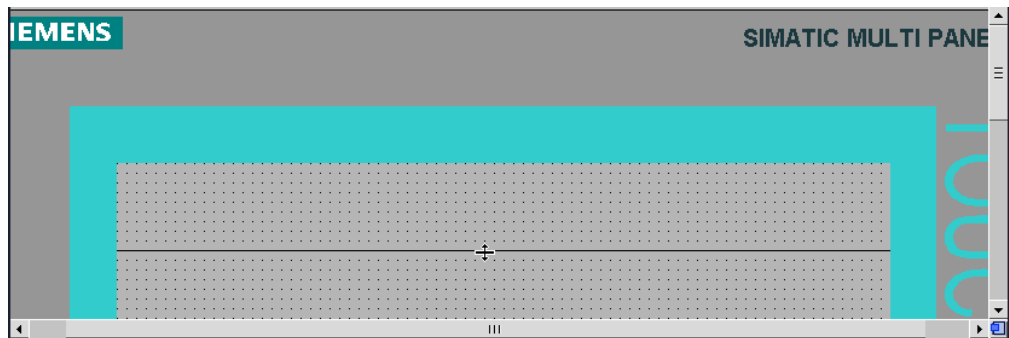
固定窗口不可用于精简面板。

组态固定窗口的步骤如下：

1. 打开带有固定窗口的 HMI 设备的画面或模板。
2. 使用鼠标（光标形状：



) 向下拖动可编辑画面区域的顶边。



此 HMI 设备的所有画面随即就都可共享此边界上方的固定窗口区域。所有组态的画面对象以由固定窗口高度确定的因子向下移动。

3. 在固定窗口中组态各元素，使其适合您的要求。

4. 固定窗口的内容出现在每个画面和模板中。

固定窗口的默认高度是“0”。最大高度是画面的最大高度减 10。

在画面中显示模板

编辑画面时，可使用现有模板并在画面中显示它。

在画面中显示模板的步骤如下：

1. 在“选项”>“设置”>“可视化”菜单中，激活“在画面中显示模板”选项。

注意

使用“全局画面”编辑器创建“报警窗口”和“报警指示器”对象。

参见

自定义模板 (页 965)

创建新模板 (页 966)

管理模板 (页 967)

全局画面 (页 968)

画面基础知识 (页 953)

8.2.1.5 自定义模板

简介

基于此模板的所有画面都将包含在此模板中组态的功能键和对象。对模板中对象或软键分配的更改将应用于基于此模板的画面中的所有对象实例。

在模板中组态局部功能键。使用“全局画面”编辑器创建全局功能键。局部功能键优先于全局功能键。

注意

HMI 设备相关性

功能键并非在所有 HMI 设备上均可用。

要求

已创建项目。

步骤

创建模板的步骤如下：

1. 在“项目”窗口中，在“画面管理”>“模板”下双击要更改的模板。

模板在工作区打开。

2. 在此模板中，自定义要由基于此模板的画面共享的功能键和对象。

结果

基于此模板的所有画面共享在此模板中组态的对象。这也同样适用于在模板中设置的软键分配。更改每个单独画面的特殊键分配。

参见

有关使用模板的基础知识 (页 963)

8.2.1.5 创建新模板

简介

一个画面始终只能基于一个模板。对于一个 HMI 设备也可以创建多个模板。一个模板不能基于另一个模板。新模板的名称会自动分配。

在模板中，可集中修改对象和功能键。模板中的更改会在基于此模板的所有画面中立即生效。

注意

HMI 设备相关性

功能键并非在所有 HMI 设备上均可用。

要求

- 已创建项目。
- 打开巡视窗口。

步骤

创建模板的步骤如下：

1. 在项目树中选择“画面管理”>“模板”，然后双击“添加模板”。
模板即在项目中创建出来，并出现在视图中。模板的属性显示在巡视窗口中。
2. 在巡视窗口中组态模板：
 - 在“常规”下指定模板名称。
 - 添加模板对象
 - 在工程系统中指定要在“层”下显示的层。
3. 组态功能键。

结果

模板创建在项目中。现在可从工具箱中添加对象和控制元素并分配功能键。

参见

有关使用模板的基础知识 (页 963)

8.2.1.5 管理模板

简介

在“项目”窗口中，可以移动、复制、重命名和删除项目中的模板。

将模板移动到组中。

使用 WinCC 创建多个组，并在这些组中对模板进行排序。

将模板移动到组中的步骤如下：

1. 选择项目树中的“画面管理”文件夹。
2. 从快捷菜单中选择“添加组”命令。
即插入了名为“Group_x”的文件夹。
3. 在项目导航中选择模板。
4. 将模板拖放至所需的组中。
模板即移动到此组。

复制模板

在 WinCC 中，可使用“复制”和“粘贴”复制模板。

复制模板的步骤如下：

1. 在项目导航中选择模板，然后从快捷菜单中选择“复制”。
2. 在项目导航中选择此模板要粘贴到的位置。
3. 从快捷菜单中选择“粘贴”以插入模板。
系统会自动将唯一名称分配给此副本。

或者，您也可以在按住 <Ctrl> 键的同时将模板拖动到所需位置。

删除模板

删除模板的步骤如下：

1. 在项目导航中，选择要删除的模板。
2. 从快捷菜单中选择“删除”。
模板及其所有对象即从当前项目中删除。

将模板分配给画面

将模板分配给画面的步骤如下：

1. 在项目导航中，选择要将模板分配到的画面。
2. 在巡视窗口中选择“常规”。
3. 在“模板”下选择所需模板。

所选模板及其中包含的所有对象即分配给画面。

参见

有关使用模板的基础知识 (页 963)

8.2.1.5 全局画面

简介

全局画面用于将以下对象或功能分配给 HMI 设备的所有画面：

- 功能：为带有功能键的设备分配功能键
- 报警指示器和控制对象：报警窗口、报警指示器

全局画面中使用的对象和功能键在“全局画面”编辑器的各巡视窗口中组态。

功能键

在“全局画面”编辑器中可全局分配 HMI 设备的功能键。除非在使用的模板中或画面本身中具有另一个分配，否则此全局分配适用于 HMI 设备的所有画面。请注意以下规则：

- 画面中的局部分配优先于模板中的局部分配和“全局画面”编辑器中的全局分配。
- 模板中的局部分配优先于“全局画面”编辑器中的全局分配。

报警指示器和控制对象

仅可将报警窗口和报警指示器作为全局元素插入到画面中。

注意

如果在画面或模板中组态了固定窗口，则不要将报警窗口和报警指示器放在固定窗口区域中。否则，报警窗口和报警指示器将不会在运行系统中显示。在“全局画面”编辑器中看不见固定窗口。

参见

有关使用模板的基础知识 (页 963)

8.2.2 使用对象

8.2.2.1 对象概述

简介

对象是用于设计项目画面的图形元素。

“工具”任务卡包含了可用于 HMI 设备的所有对象。激活“任务卡”选项可以显示带有菜单命令“视图”的任务卡。

工具箱所含的选项板因当前激活的编辑器而异。如果“画面”编辑器打开，则工具箱包含以下选项板：

- “基本对象”

基本对象包括图形对象（如“线”或“圆”）和标准控制元素（如“文本域”或“图形显示”）。

- “元素”

元素包括标准控制元素，如“IO 域”或“按钮”。

- “控件”

控件用于提供高级功能。它们也动态地代表过程操作，例如趋势视图和配方视图。

- “图形”

图形以目录树结构的形式分解为主题。各种文件夹包含以下图形表示：

- 机器人和工厂区域
- 测量设备
- 控制元素
- 标志
- 建筑物

您可以创建指向您自己的图形文件夹的链接。外部图形位于这些文件夹和子文件夹中。它们显示在工具箱中，并通过链接集成到项目中。

- “库”任务卡

除了显示和操作员控件，还有库对象可供使用。它们位于“库”任务卡的选项板中。库包含预组态的对象，如管道、泵或预组态的按钮的图形。也可以将库对象的多个实例集成到项目，而不必重新组态。









WinCC 软件包中包含了库。

注意












可用于特定 HMI 设备

一些工具箱对象或者只能发挥有限的功能或者根本不可用。这取决于正在组态的 HMI 设备。对象不可用的属性显示为禁用状态，因此无法选择。








基本对象

图标	对象	说明
	“线”	-
	“折线”	折线是开放对象。即使起点和终点坐标相同，它们包围的区域也不能填充。如果要填充多边形，请选择“多边形”对象。
	“多边形”	-
	“椭圆”	-
	“圆”	-
	“矩形”	-
	“文本框”	一行或多行文本。字体和布局是可调整的。
	“图形视图”	显示来自外部图形程序的图形并插入 OLE 对象。可使用以下图形格式：“*.emf”、“*.wmf”、“*.dib”、“*.bmp”、“*.jpg”、“*.jpeg”、“*.gif”和“*.tif”。

元素

图标	对象	说明
	“I/O 域”	输出变量的值，和/或将值写入变量。 可以为显示在 I/O 域中的变量值定义限值。要在运行系统中隐藏操作员输入，请激活“隐藏输入”。
	“按钮”	根据组态执行函数列表或脚本。
	“符号 I/O 域”	输出变量的值，和/或将值写入变量。根据变量值显示相关文本列表中的文本。
	“图形 I/O 域”	输出变量的值，和/或将值写入变量。根据变量值显示相关图形列表中的图形。
	“日期/时间域”	输出系统日期和时间，或输出变量中的时间和日期。操作员可输入新值。显示格式是可调整的。
	“棒图”	棒图以带刻度的棒图形式显示 PLC 值。
	“开关”	在两个定义的状态之间切换。可用文本或图形给开关加标签。
	“符号库”	用于添加基于同名控件的画面对象。
	“滚动条”	显示来自 PLC 的当前值，或将数字值发送到 PLC。
	“量表”	显示数字值。 量表的外观是可调整的。
	“时钟”	以模拟或数字格式显示系统时间。

控件

图标	对象	描述
	“报警视图”	显示来自报警缓冲区或报警日志的选定报警或报警事件。
	“趋势视图”	以来自 PLC 或日志的值表示多条曲线。
	“用户视图”	允许管理员在 HMI 设备上管理用户。 还允许没有管理员权限的操作员更改密码。
	“状态强制”	此功能赋予操作员对所连 SIMATIC S7 中的各地址区域的直接读/写访问权。
	“配方视图”	显示数据记录，并允许编辑。
	“浏览按钮”	转到当前画面层级结构的根画面。 转到画面层级结构同一层级中当前画面左侧的下一个画面。 转到父画面。 转到层级结构中较低层级最左侧的画面。 转到画面层级结构同一层级中当前画面右侧的下一个画面。
	“动态浏览按钮”	将按钮位置预留给动态浏览按钮。如果画面在画面层级结构中有子画面，则会在该位置自动生成动态浏览按钮。

参见

[画面基础知识 \(页 953\)](#)
[用于编辑对象的选项 \(页 972\)](#)
[插入对象 \(页 973\)](#)
[删除对象 \(页 975\)](#)
[定位对象 \(页 975\)](#)
[调整对象的大小 \(页 977\)](#)
[选择多个对象 \(页 978\)](#)
[排列对象 \(页 980\)](#)
[向前或向后移动对象 \(页 982\)](#)
[显示画面区域外的对象 \(页 983\)](#)
[旋转对象 \(页 984\)](#)
[翻转对象 \(页 985\)](#)
[插入同一类型的多个对象 \(标记工具 \) \(页 986\)](#)
[重新定位多个对象并调整其大小 \(页 987\)](#)
[外部图形 \(页 988\)](#)
[管理外部图形 \(页 989\)](#)
[将外部图像存储在图形库中。\(页 991\)](#)
[关于组的基础知识 \(页 993\)](#)

实例：插入和组态矩形 (页 1002)

8.2.2.2 用于编辑对象的选项

简介

对象是用于设计项目画面的图形元素。

可用以下选项来编辑对象：

- 使用快捷菜单复制、粘贴或删除对象。如果将某对象复制到画面中，但画面中已包含同名对象，则将更改该对象的名称。
- 保持所插入对象的标准大小或在插入时自定义对象大小。
- 更改对象的属性，例如大小。
- 定位对象
- 将对象移动到其它对象的前面或后面。
- 旋转对象
- 镜像对象
- 更改对象的默认属性
- 定义对象的 Tab 顺序
- 图章：插入多个相同类型的对象
- 同时选择多个对象
- 重新定位多个对象并调整其尺寸
- 可以将外部图形分配给对象，例如在图形视图中。

只能查看先前存储在 WinCC 项目的图形浏览器中的图像。

可按如下方式将图形保存在图形浏览器中：

- 从“图形”对象组拖放到工作区的图像
- 下列格式的图形文件：*.bmp、*.dib、*.ico、*.emf、*.wmf、*.gif、*.tif、*.jpeg 或 *.jpg
- OLE 对象

既可以创建新的 OLE 对象，也可以将现有的图形文件另存为 OLE 对象。要保存 OLE 对象，必需在组态计算机上安装了 OLE 兼容的图形程序。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.3 插入对象

简介

可在“画面”或“报表”编辑器中将对象从工具箱或 WinCC 图形文件夹添加到画面或报表中。使用鼠标将对象拖入工作区。粘贴对象时，可以保持对象原始大小，也可以进行缩放。

每个用户都可以定义自己的标准对象大小。

对于要定义的线条和由线条组成的对象，可在插入工作区时定义其大小。

注意 精简面板

“报表”编辑器不可用于精简面板。

要求

打开“工具”任务卡。

以标准大小插入对象

执行下列步骤，按标准大小插入对象：

1. 在“工具箱”任务卡中选择所需的图形对象，或在 WinCC 图形文件夹中选择所需的图形。
当工作区上移动光标时，光标会变成带有附加对象图标的十字准线。
2. 单击工作区中要插入对象或图形的位置。
对象就以标准大小插入到了工作区中的所需位置。
3. 可以通过拖动所选矩形的尺寸控制点来调整对象大小。
也可以在巡视窗口“布局”的“位置和大小”区域中输入所需要的大小（以像素为单位）。
4. 可以在巡视视图中指定其它对象属性。
要添加更多图像对象，重复步骤 1 和 2。

插入对象并同时选择其大小

插入对象并同时选择其大小的步骤如下：

1. 在“工具箱”任务卡中选择所需的图形对象，或在 WinCC 图形文件夹中选择所需的图形。
2. 将光标移动到工作区中要插入对象的位置。
鼠标指针变成附带对象图标的十字准线。
3. 按下鼠标左键并将对象拖动至所需要的大小。
松开鼠标按钮，以所需大小将对象粘贴到工作区中的所需位置。
要添加更多图像对象，重复步骤 1 和 2。

插入线条

插入线条的步骤如下：

1. 在“工具”任务卡中选择所需图形对象。
2. 在工作区中单击作为线条起点的位置。
这将确定线条的起点。
3. 在工作区中移动鼠标指针。
这样会在起点和光标位置之间绘制一条线。
4. 在工作区中单击作为线条终点的位置。
这将确定线条的终点。

插入多边形或折线

插入线对象的步骤如下：

1. 在“工具”任务卡中选择所需对象“折线”或“多边形”。
2. 在工作区中单击作为对象起点的位置。
这将确定对象的起点。
3. 单击工作区中要定义对象拐角的位置。
4. 针对对象的每个其它转角重复步骤 3。
5. 双击工作区中要定义最后一个对象拐角的位置。
这将确定多边形或折线的所有点。

注意

精简面板

“多边形”和“折线”对象不可用于精简面板。

注意

如果要插入同类型的多个对象，可使用“标记”功能。这样可避免在每次插入对象前都必须在“工具”任务卡中重新选择该对象。为此，在“工具”任务卡的工具栏中选择



图标。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.4 删除对象

简介

可以单独删除对象或选择多个同时删除。

要求

已打开包含至少一个对象的工作区。

步骤

删除对象的步骤如下：

1. 选择要删除的对象。
为了删除多个对象，按住 <Shift> 键并逐个选择要删除的对象。
2. 从快捷菜单中选择“删除”。

结果

选定对象即删除。

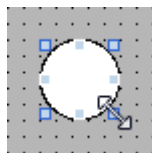
参见

对象概述 (页 969)
实例：插入和组态矩形 (页 1002)

8.2.2.5 定位对象

简介

选择对象时，对象被带有调整大小拖动点矩形包围。此矩形是选择矩形。使用选择矩形的左上角的坐标定义对象位置。



注意

如果对象位于工作区外（即其中一个坐标是负值），则在运行系统中看不到此对象。

可通过以下方式重新定位对象：

- 将对象拖动到新位置
- 使用箭头键移动对象

- 在巡视窗口中更改“位置”属性

网格

网格显示在工作区中。如果要使用鼠标定位对象或者更改其大小，对象会自动与网格对齐。如果按住 <Alt> 键，对象则不再与网格对齐。

注意

对报表而言，网格和“网格对齐”功能始终处于活动状态。

对于画面，可按如下方式激活和禁用网格和“网格对齐”功能：

- 在“选项”>“设置”>“可视化”菜单中
- 在“画面”编辑器中：打开“布局”任务卡并在“网格”选项板中激活所需功能。

要求

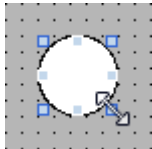
已打开包含至少一个对象的工作区。

步骤

更改对象位置的步骤如下：

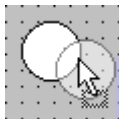
1. 选择想要移动的对象。

用带有尺寸控制点的矩形将所选对象框起。



2. 左键单击对象，并按住鼠标按钮。
3. 将鼠标指针移动到新位置。

对象的轮廓随鼠标的移动而移动，定位对象的新位置。



对象开始时保持在其原位置。

4. 现在释放鼠标按钮。

对象将移动到选择矩形轮廓所指示的位置。

- 激活“网格对齐”功能时，对象将与网格对齐。
- 如果拖动对象时按住 <ALT> 键，则禁用该功能。

或：

1. 在巡视窗口中，选择“属性”>“布局”。
2. 在“位置和大小”下输入位置的 X 和 Y 值。

结果

对象显示在其新的位置。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.6 调整对象的大小

简介

选择对象时，带有尺寸控制点的矩形将对象围起来。下列选项可用于调整对象的大小：

- 使用鼠标拖动调整尺寸控制点。
- 在巡视窗口中修改“大小”属性。

网格

网格显示在工作区中。如果要使用鼠标定位对象或者更改其大小，对象会自动与网格对齐。如果按住 <Alt> 键，对象则不再与网格对齐。

注意

对报表而言，网格和“网格对齐”功能始终处于活动状态。

对于画面，可按如下方式启用和禁用网格和“网格对齐”功能：

- 在“选项”>“设置”>“可视化”菜单中
- 在“画面”编辑器中：打开“布局”任务卡并在“网格”选项板中激活所需功能。

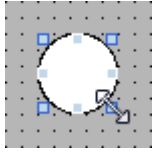
要求

已打开包含至少一个对象的工作区。

步骤

调整对象大小的步骤如下：

1. 选择想要调整其大小的对象。
显示选择矩形。下图显示了一个已选对象：



2. 将矩形的尺寸控制点拖动到一个新位置。

对象的大小已改变。

- 如果设置“网格对齐”功能，则对象大小将与网格图案对齐。
- 按下 <ALT> 后，在拖动对象期间该功能禁用。

注意

要让对象按一定比例缩放，可在使用鼠标更改大小时按住 <Shift> 键。

或：

1. 在巡视窗口中选择“布局”。
2. 在“位置和大小”下输入对象的大小。

结果

对象将以新大小显示。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.7 选择多个对象

简介

选择想相互对齐或更改全局属性的所有对象。这一操作过程称为“多项选择”。

“巡视”窗口显示所选对象的所有属性。

现在可使用多种方法一次选择多个对象：

- 在对象周围绘制一个选择框。
- 按住 <Shift> 键，然后单击所需对象。

多项选择的选择框

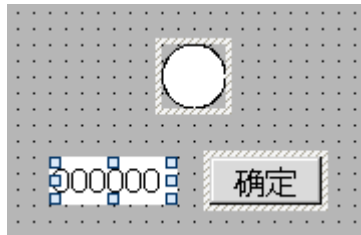
选择框包围多项选择的所有对象。这种选择框与框起单个对象的矩形相当。

选择框不可见。进行多项选择时，将显示下列框：

- 参考对象以其周围的矩形指示。
- 其它所选对象以短划线边框指示。

指定参考对象

参考对象是用于定位其它对象的对象。一个带有句柄的矩形框住引用对象。下图显示了一个参考对象及两个其它所选对象：



参考对象的指定有以下可选方式：

- 通过多项选择来选择对象。最先选择的对象就是参考对象。
- 在对象周围绘制一个选择框。参考对象被自动编译。如果希望在选择的对象中指定一个不同的对象作为参考对象，可单击所需的对象。该操作不会取消多项选择。

要求

已打开包含至少两个对象的工作区。

使用选择框选择多个对象

执行下列步骤，使用选择框选择多个对象：

1. 将鼠标指针置于工作区中要选择的某一个对象附近。
2. 按住鼠标按钮，在要选择的对象周围绘制选择框。

或：

1. 按住 <Shift> 键。
2. 连续单击相关对象。

所有选择的对象都由框标识。

最先选择的对象被识别为参考对象。

注意

要从多项选择中删除某个对象，可按住 <SHIFT> 并再次单击相关对象。

结果

将选择多个对象。其中一个对象被识别为参考对象。现在可以进行下列步骤：

- 更改所有对象的对象属性
- 拖动选择框来放大或缩小所有对象，从而按相同比率调整所有对象的大小
- 以一个组的形式移动所有对象
- 将其它对象与参考对象对齐

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)




8.2.2.8 排列对象**步骤**

对齐对象的步骤如下：

1. 通过多项选择来选择对象。
2. 指定作为参考对象的对象。
3. 从工具栏或快捷菜单中选择命令（参见下表）。
所选对象将对齐。

对齐对象

将所选对象与参考对象对齐。

图标	描述
	将所选对象与参考对象的左边缘对齐。
	将所选对象与参考对象的垂直中心坐标轴对齐。
	将所选对象与参考对象的右边缘对齐。
	将所选对象与参考对象的上边缘对齐。
	将所选对象与参考对象的水平中心坐标轴对齐。
	将所选对象与参考对象的下边缘对齐。
	将所选对象与参考对象的中心点对齐。

均匀分布对象

至少需要选择三个对象。不需要参考对象。

1. 选择对象。
2. 单击“横向等距分布”或“纵向等距分布”按钮。
等间距分布所选对象。

以下画面显示了如何等间距分布所选对象：



图标	描述
	水平等间距分布对象。 最左侧和最右侧的对象位置保持不变。所有其它对象在这两个对象之间均匀分布。
	垂直等间距分布对象。 最上面和最下面（右侧和左侧）的对象位置保持不变。所有其它对象在这两个对象之间均匀分布。

使对象大小一致

将所选对象的高度或宽度按照参考对象的高度或宽度进行调整。

1. 选择对象。

2. 单击



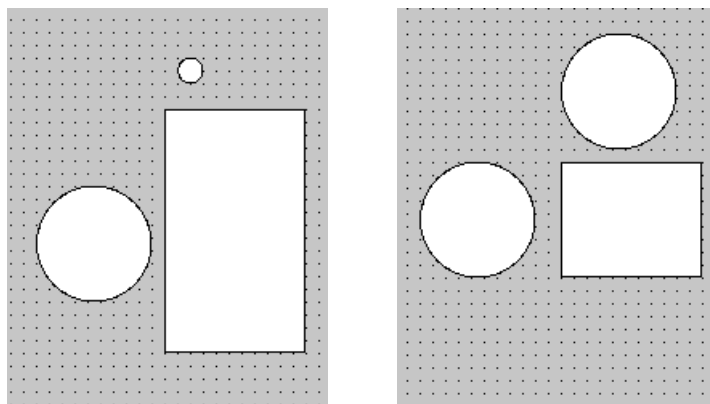
或





按钮。

所选对象大小互相匹配。

以下画面显示了如何将所选对象调整到参考对象的高度：



图标	描述
	将所选对象与参考对象的宽度对齐。
	将所选对象与参考对象的高度对齐。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.9 向前或向后移动对象

简介

可使用所选对象快捷菜单或工具栏中的“顺序”功能将所选对象移动到对象层内其它对象的前面或后面。

注意

ActiveX 控件始终位于对象层的前面 (.NET 属性)。





要求

已经打开一个画面，该画面某层中包含有多个对象。

步骤

更改层内对象顺序的步骤如下：

1. 选择要向前或向后移动的对象。
2. 从快捷菜单中选择“排序”命令和下列命令之一：

图标	描述
	将所选对象移动到同一层所有其它对象的前面
	将所选对象移动到同一层最下面的位置
	将所选对象上移一个位置
	将所选对象下移一个位置

其它方法

步骤如下：

1. 打开“布局”任务卡的“层”选项板。
2. 浏览到所需对象。
3. 按住鼠标按钮并将树形结构中的对象拖动到层中的所需位置。
4. 现在释放鼠标按钮。

结果

对象向前或向后移动。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.10 显示画面区域外的对象

简介

如果将对象分配到可组态区域之外的位置，则这些对象将隐藏起来。“布局”任务卡中“可见区域外的对象”选项板的各项功能用于将这些对象移回画面中。

要求

- 打开一个包含可组态区域外对象的画面。
- “布局”任务卡已打开。

步骤

步骤如下：

1. 打开“布局”任务卡的“区域外的对象”选项板。

将出现可组态区域外对象的列表。

2. 选择要移回画面的对象。
3. 单击“范围外的对象”选项板中的“移动到画面”。

或者，打开“布局”任务卡的“层”选项板。区域外的对象以



图标指示。如果单击此图标，对象将移回画面中。

结果

对象将显示在可组态区域的右边缘。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.11 旋转对象

简介

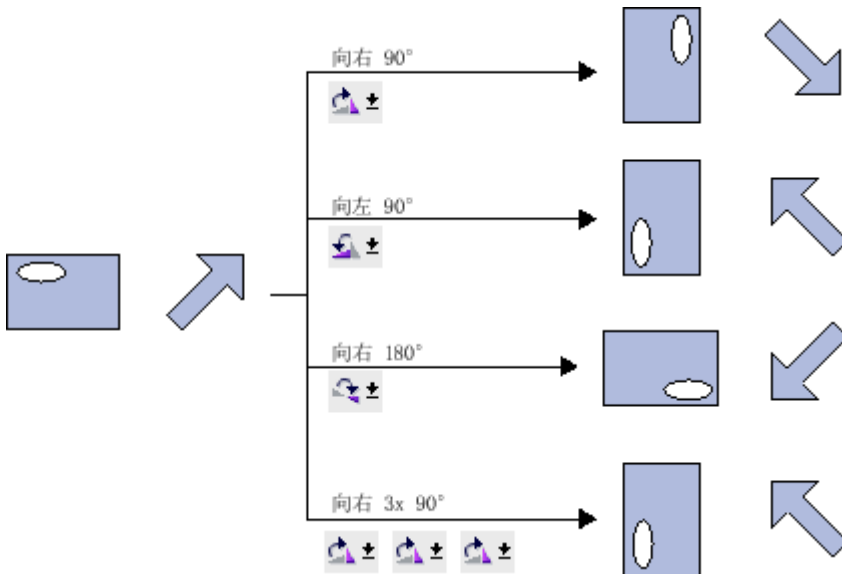
可以使适当的对象围绕其中心轴以 90 度为增量按顺时针或逆时针旋转。

注意

并非所有对象都可以旋转。某些可在画面中旋转的对象在报表中却不能旋转。

还可以使用多项选择功能旋转多个对象。某些 WinCC 对象 (如按钮) 不能旋转。

在旋转的对象中，对象中的元素对齐方式将有所变化。下图显示了在使用不同命令旋转对象时矩形和椭圆的状态：



要求

已打开包含至少一个对象的工作区。

步骤

旋转对象的步骤如下：

1. 选择要旋转的对象。
2. 单击下列工具栏图标之一：



，将对象围绕其中心点顺时针旋转。旋转角度为 90 度。



，将对象围绕其中心点逆时针旋转。旋转角度为 90 度。



，将对象顺时针旋转 180 度。

结果

对象在其新的角度位置显示。

参见

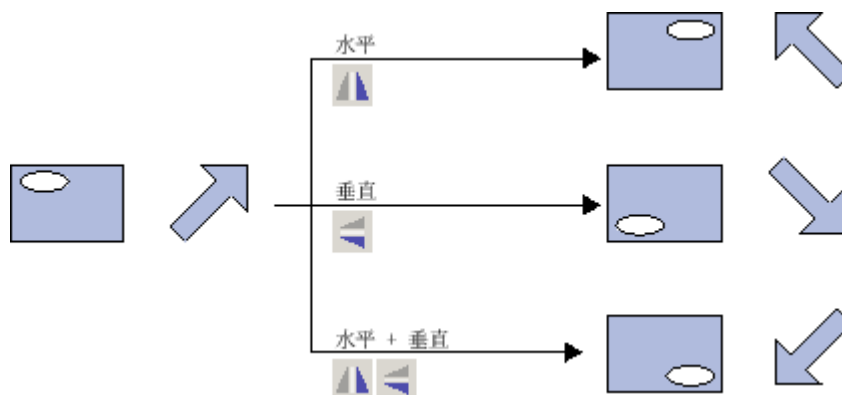
对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.12 翻转对象

简介

可以沿对象的垂直或水平中心坐标轴将其翻转。翻转对象时，对象中的元素对齐方式将有所变化。下图显示了在使用不同命令翻转对象时矩形和椭圆的状态：



要求

已打开包含至少一个对象的画面。

步骤

执行下列步骤，翻转对象：

1. 选择要翻转的对象。
2. 单击快捷菜单中的“翻转”命令并选择一个显示的选项：



，将所选对象沿其垂直中心轴翻转。



，将所选对象沿其水平中心轴翻转。

结果

对象在其翻转后的位置显示。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.13 插入同一类型的多个对象 (标记工具)

简介

WinCC 提供直接连续“标记”或插入多个同类型对象的可能性。因此，不必每次都重新选择对象。此外，您还可以将插入的对象放大。

要求

打开“工具”任务卡。

插入多个同类型对象

插入多个同类型对象的步骤如下：

1. 在“工具”任务卡中选择要插入的对象。
2. 单击“工具”任务卡工具栏中的



图标。

“标记”功能即激活。

3. 要以标准大小插入对象，单击工作区中相关的插入位置。

要以非标准大小插入对象，将鼠标指针置于工作区中的所需位置。按下鼠标左键并将对象拖动至所需要的大小。

在您释放鼠标按钮之后，对象即会插入到工作区中。

4. 重复步骤 3 以插入更多相同类型的对象。
5. 再次单击



图标。

“标记”功能即禁用。

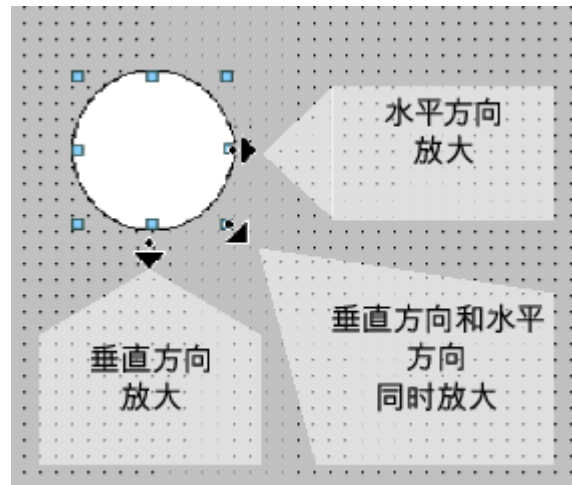
注意

可以使用鼠标拖放 +<CTRL> 功能复制现有的对象。在这种情况下，现有对象不被移动。而是将该对象的副本粘贴到新的位置。

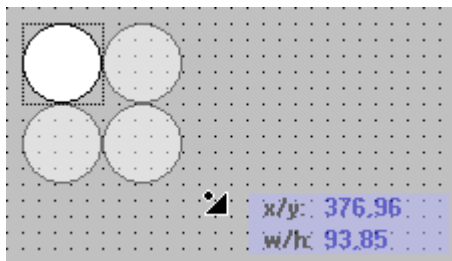
插入并放大对象

插入并放大对象的步骤如下：

1. 从“工具”任务卡插入所需对象。
2. 按 <Ctrl> 键，并将光标置于下图所示的控制点之一。



3. 按住鼠标左键的同时，向右和/或向下拖动拖动点。
4. 如果持续移动光标，则对象会根据可用空间放大。



参见

对象概述 (页 969)
实例：插入和组态矩形 (页 1002)

8.2.2.14 重新定位多个对象并调整其大小

可能的修改

选择多个对象后，可以按如下方式进行编辑：

- 使用鼠标移位

- 要更改标记对象的绝对位置，可将鼠标指针置于对象上方，然后按住鼠标对多项选择进行移位操作。
- 要按照相同比率调整所有对象的大小，可拖动参考对象的大小调整拖动点。
- 使用工具栏的图标在工作区移动
 - 更改标记对象彼此之间的相对位置
 - 对齐标记对象的高度和宽度
- 使用工作区的快捷菜单命令移动
 - 更改标记对象彼此之间的相对位置
 - 对齐标记对象的高度和宽度

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.15 外部图形

简介

在 WinCC 中，会用到通过外部图形编辑器创建的图形。要使用这些图形，可将它们存储在 WinCC 项目的图形浏览器中。

可按如下方式将图形保存在图形浏览器中：

- 将图形对象从“图形”窗格拖放到工作区中时，这些图形对象会自动存储在图形浏览器中。图形名称按其创建顺序编号，例如“Graphic_1”。使用 <F2> 功能键可将图形重命名。
- 存为下列格式的图形文件：
.bmp、.ico、*.emf、*.wmf、*.gif、*.tif、*.jpeg 或 *.jpg
- 存为嵌入 WinCC 中并链接到外部图形编辑器的 OLE 对象。通过 OLE 链接，可从 WinCC 打开外部图形编辑器。链接的对象使用该图形编辑器编辑。只有在外部图形编辑器已安装在 PC 上并且支持 OLE 时，OLE 链接才生效。

使用图形浏览器中的图形

图形浏览器中的图形在画面中有如下用处：

- 用在图形视图中
- 用在图形列表中
- 用作按钮/功能键的标签

透明图形

在 WinCC 中，也会用到具有透明背景的图形。将具有透明背景的图形插入 WinCC 的图形对象时，这种透明属性会被为图形对象指定的背景色所取代。所选背景色与图形紧密链接。如果在 WinCC 的另一图形对象中使用此图形，则该对象的背景色将与第一个组态的图形对象的背

景色相同。如果以其它背景色使用该图形，则将该图像以其它名称再次加入图形浏览器中。当该图形用在相应的 WinCC 图形对象中时，即组态了其它背景色。

管理图形

WinCC 随附了大量的图形、图标和符号，例如：

- 机器和工厂区域
- 测量设备
- 控制元素
- 标志
- 建筑物

在“图形”窗格的“工具箱”窗口中，图形对象按主题分类保存在“WinCC 图形文件夹”中。WinCC 图形文件夹链接不能删除、编辑或重命名。

“图形”窗格也用于管理外部图形。它具有以下功能：

- 创建图形文件夹的链接
该文件夹及其子文件夹中的外部图形对象显示在工具箱中，从而集成到项目中。
- 文件夹链接
 - 编辑
 - 重命名
 - 更新
 - 删除
- 在 WinCC 中打开编辑外部图形所需的程序。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.16 管理外部图形

简介

要在 WinCC 中使用的外部图形可通过“图形”窗格中的“工具”任务卡在“画面”编辑器中进行管理。

要求

- 打开“画面”编辑器。
- 打开“工具”任务卡。

为在工具箱中显示外部图形：

- 图形文件可用。
- 图像文件具有以下格式：

.bmp、.ico、*.emf、*.wmf、*.gif、*.tif、*.jpeg 或 *.jpg

创建文件夹链接

创建文件夹链接的步骤如下：

1. 单击“我的图形文件夹”。
“创建文件夹链接”对话框即打开。该对话框会提供文件夹链接名称的建议。
2. 根据需要编辑名称。选择包含图形对象的路径。
3. 单击“确定”，确认输入。

新文件夹链接即添加到“图形”对象组中。位于目标文件夹和子文件夹的外部图形显示在工具箱中。

编辑文件夹链接

编辑文件夹链接的步骤如下：

1. 选择要编辑的文件夹链接。
2. 从快捷菜单中选择“编辑链接...”命令。
“创建文件夹链接”对话框即打开。
3. 根据需要编辑文件夹链接的名称和路径。
4. 单击“确定”，确认输入。

重命名文件夹链接

文件夹链接重命名的步骤如下：

1. 选择要重命名的文件夹链接。
2. 从快捷菜单中选择“重命名”。
3. 为新文件夹链接分配一个名称。

删除文件夹链接

删除文件夹链接的步骤如下：

1. 选择要删除的文件夹链接。
2. 在快捷菜单中选择“删除”。

编辑外部图形

编辑外部图形的步骤如下：

1. 选择要编辑的图形。
2. 从快捷菜单中选择“打开画面编辑器...”。

这将打开与图形对象文件相关联的画面编辑器。

在 WinCC 中编辑图形文件夹

在 WinCC 中编辑图形文件夹的步骤如下：

1. 选择要编辑的图形。

2. 从快捷菜单中选择“打开父文件夹”命令。
打开“Windows 资源管理器”。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.17 将外部图像存储在图形库中。

简介

要在画面中显示在外部图形程序中所创建的图形，首先要将这些图形存储到 WinCC 项目的图形浏览器中。

要求

- 已经创建一个画面。
- 在画面中插入了图形视图。
- 打开图形视图的巡视窗口。

在图像浏览器中存储外部图形的要求：

- 图形可用。

要将 OLE 对象存储在浏览器中：

- 在组态计算机上安装 OLE 兼容的图形程序。

保存图形文件

保存图形文件的步骤如下：

1. 打开 Windows 资源管理器。
2. 选择要存储的图形。
3. 将图形拖放到图形浏览器中。

创建新图形并保存为 OLE 对象

步骤如下：

1. 在图形视图的巡视窗口中单击“常规”组。
2. 打开图形选择列表。
3. 单击



4. “插入对象”对话框随即打开。

注意

此外，还将打开“外部应用程序正在运行...”对话框。该对话框将在您退出外部应用程序后关闭。

5. 在“插入对象”对话框中，选择“新建”和对象类型。“设置”>“OLE 设置”中的设置决定将显示哪些对象类型。
6. 单击“确定”。将打开关联的图形程序。

完成创建图形后，通过“文件”>“关闭”或“文件”>“关闭并返回到 WinCC”结束图形编程软件。

图形将以图形编程软件标准格式进行存储，并被添加至图形浏览器。

将创建的图形插入 WinCC

注意

当创建为 OLE 对象的新图形对象被保存到外部图形程序时，在 WinCC 中可能不会被直接接受。

步骤如下：

1. 重新打开用于插入图形的对话框。
2. 从“插入对象”对话框中，选择“从文件创建”。
3. 单击“浏览”按钮。
4. 找到创建的图形并选择该图形。

将现有图形对象保存为 OLE 对象

请按如下步骤进行操作：

1. 在图形视图的巡视窗口中单击“常规”组。
2. 打开图形选择列表。
3. 单击



。

4. “插入对象”对话框随即打开。

注意

此外，还将打开“外部应用程序正在运行...”对话框。该对话框将在您退出外部应用程序后关闭。

5. 从“插入对象”对话框中，选择“从文件创建”。
6. 单击“浏览”按钮。
7. 使用该对话框帮助您浏览到保存图形文件的文件夹。

要导入图形文件，请注意以下大小限制：

"*.bmp" ≤ 4 MB

"*.jpg" ≤ 1 MB

"*.jpeg" ≤ 1 MB

结果

现在正将图像文件存储到图像浏览器中。可以通过图形显示在画面上查看它，或将其作为列表元素添加到图像列表。

可双击库中的 OLE 对象将其打开，以在相应的图形编辑器中进行编辑。完成图形编辑后，通过“文件”>“关闭”或“文件”>“关闭并返回到 WinCC”结束图形编程软件。更改将被应用至 WinCC。

参见

对象概述 (页 969)

实例：插入和组态矩形 (页 1002)

8.2.2.18 使用对象组

8.2.2.18 关于组的基础知识

简介

组是使用“编组”功能组合在一起的若干个对象。可以按与编辑任何其它对象相同的方式编辑组。

概述

WinCC 提供下列方法来编辑多个对象：

- [多项选择 \(页 978\)](#)
- [创建对象组 \(页 994\)](#)

编辑模式

要分别编辑组中的对象，请在“布局”任务卡“层”选项板中选择对象。

分层组

可通过添加更多对象或组来扩展一个组。该组随即扩大为包括新对象并将按层次分为主组和子组或对象。这种分层组必须逐级分解。也可以按对象或组加入该组中的顺序来进行分解。这些分层组的分解与创建需要的步骤数相同。

包围对象的矩形

对于组，仅为整个组显示一个包围对象的矩形。但是，对于多项选择，围绕所有对象的矩形都会显示出来。

层

组的所有对象都位于同一层面。

参见

取消编组 (页 995)
将对象添加到组 (页 996)
从组中删除对象 (页 996)
编辑组中的对象 (页 998)
对象概述 (页 969)
实例：插入和组态矩形 (页 1002)

8.2.2.18 创建对象组

简介

“编组”命令用于将多个对象组合成一个组。

可以更改组的大小和位置。以下规则适用：

- 重新定位组时，系统会自动调整组合对象的位置坐标。编组的对象相对于组的位置不受影响。
- 系统按照组大小变化的比例自动调整编组的对象的长度和宽度。
- 要成比例地更改组的大小，可按住 <Shift> 键并拖动对象周围的矩形直到得到所需的大小。

注意

要创建分层组，可像处理对象那样组织各个组。

要求

- 打开一个至少包含两个对象的画面。

创建对象组

步骤如下：

1. 选择要组织在一个组中的所有对象。
2. 从快捷菜单中选择“编组”>“编组”命令。

组中对象的周围会显示矩形。

编组组中的对象

步骤如下：

1. 选择要编辑的组。
2. 从快捷菜单中选择“编组”>“编辑组”命令。

用红色边框突出显示要编辑的组。

3. 选择要组合成子组的组对象。

4. 从快捷菜单中选择“编组”>“编组”命令。
含这些对象的子组即创建完毕。

将对象添加到现有组

请按如下步骤进行操作：

1. 选择要添加对象的组。
2. 按住 <Shift> 键并选择要添加到组的对象。
3. 从快捷菜单中选择“编组”>“添加到组”命令。

对象将被添加至该组。

其它方法

也可以在“布局”任务卡中编辑组。还可以在“层”窗格中通过拖放轻松地编辑嵌套组。

结果

所选的对象即组合在一个组中。多项选择矩形变为环绕在组中对象周围的矩形。只显示组的拖动点。该组位于活动层中。

参见

关于组的基础知识 (页 993)

实例：插入和组态矩形 (页 1002)

8.2.2.18 取消编组

简介

选择“取消编组”命令将组拆分成原来的对象组成。

要求

- 打开包含一个组的画面。

取消编组

步骤如下：

1. 选择组。
2. 从快捷菜单中选择“编组”>“取消编组”命令。

对组中的组取消编组

执行下列步骤，将组中的组取消编组：

1. 选择高层级的组。
2. 从快捷菜单中选择“编组”>“编辑组”命令。

用红色边框突出显示要编辑的组。

3. 选择较低层级的组。
4. 从快捷菜单中选择“编组”>“取消编组”命令。

将对较低层级的组取消编组。对象即分配给上一层级的组中。

其他方法

也可以在“布局”任务卡中编辑组。使用拖放方法，还可以在“层”窗格中轻松编辑嵌套组。

参见

关于组的基础知识 (页 993)

实例：插入和组态矩形 (页 1002)

8.2.2.18 将对象添加到组

简介

“将对象添加到组”命令用于在不事先取消编组的情况下将对象添加到组中。

要求

打开至少包含一个对象的组。

将对象添加到组

步骤如下：

1. 选择组。
2. 按住 <Shift> 键并选择要添加到组的对象。
3. 从快捷菜单中选择“编组”>“添加到组”命令。

该组将包含初始对象和新添加的对象。添加的对象排列在组的前面。

其他方法

也可以在“布局”任务卡中编辑组。使用拖放方法，还可以在“层”窗格中轻松编辑嵌套组。

参见

关于组的基础知识 (页 993)

实例：插入和组态矩形 (页 1002)

8.2.2.18 从组中删除对象

简介

“从组中删除对象”命令用于在事先不取消编组的情况下将各个对象从组中删除。

从组中删除对象

步骤如下：

注意

要编辑组中的对象，不必先将其从组中删除。可以分别编辑组中的对象。

要求

- 打开包含一个组的画面。

从组中删除对象

执行下列步骤，从组中删除对象：

1. 选择组。
2. 从快捷菜单中选择“编组”>“编辑组”命令。
用红色边框突出显示要编辑的组。
3. 选择要从组中删除的组内的所有对象。
4. 从快捷菜单中选择“编组”>“从组中删除”命令。

对象将从组中删除。

注意

即使组内的所有对象都被删除，组也不会被自动删除。

从组中删除对象

要从组中和画面中删除对象，可执行下列步骤：

1. 选择组。
2. 从快捷菜单中选择“编组”>“编辑组”命令。
用红色边框突出显示要编辑的组。
3. 选择所要删除的位于组中的对象。
4. 从快捷菜单中选择“删除”。

注意

即使组内的所有对象都被删除，组也不会被自动删除。

其他方法

也可以在“布局”任务卡中编辑组。使用拖放方法，还可以在“层”窗格中轻松编辑嵌套组。

参见

关于组的基础知识 (页 993)

实例：插入和组态矩形 (页 1002)

8.2.2.18 编辑组中的对象

简介

可以分别编辑组中的对象。

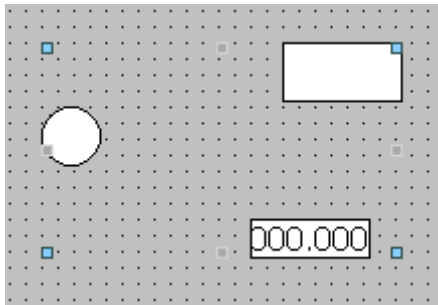
要求

打开包含一个组的画面。

编辑编组的对象

步骤如下：

1. 选择组。



组的属性即显示在巡视窗口中。

2. 在“布局”中更改编组的对象的位置和大小。
3. 在“其它”下更改组的名称。

指定活动层并在运行系统中激活组的可见性。

修改组中对象的属性

步骤如下：

1. 选择组。
2. 在巡视窗口中选择要更改属性的对象。



将显示对象的属性。

3. 更改对象属性。

结果

虽然编辑单个对象，但它仍然是组的元素。这些更改不会影响组中的其它对象。

参见

关于组的基础知识 (页 993)
实例：插入和组态矩形 (页 1002)

8.2.2.19 组态键盘访问

8.2.2.19 键盘访问概述

简介

对于未配备鼠标的键盘单元，操作员应使用 <Tab> 键来激活控制对象。可以通过设置键盘输入来使过程更易于运行并确保操作员输入所有必要的值。如果使用键盘，可通过 <Tab> 键以特定顺序激活对象并输入必要的值。

对于未配备按键的 HMI 设备，可以通过将“SimulateSystemKey”系统函数组态为功能键来仿真 <Tab> 键。

操作员权限和操作员控制启用

如果将某个对象组态为通过 <Tab> 键实现操作员输入，则该对象必须同时具有操作员权限和操作员控制启用。

编辑 Tab 顺序

Tab 顺序是在创建控制对象时自动确定的。Tab 顺序的编号按库的创建顺序分配。

在下列情况下会更改 Tab 顺序：

- 操作员直接改到某个特定控制对象。
- 画面需要特定顺序

要更改 Tab 顺序，需切换到 Tab 顺序模式。在该模式下，Tab 顺序编号显示在控制对象的左上方。隐藏对象的 Tab 顺序编号也会显示出来。通过鼠标可以编辑这些编号的分布。

注意

其它功能在 Tab 顺序模式下不能使用。

参见

定义对象的操作员权限和操作员控制启用 (页 1000)
设置 Tab 顺序 (页 1001)

8.2.2.19 定义对象的操作员权限和操作员控制启用

简介

如果将某个对象组态为通过 <Tab> 键实现操作员输入，则该对象必须同时具有操作员权限和操作员控制启用。

要求

已打开包含至少一个对象的画面。

步骤

为对象指定操作员权限和操作员控制启用的步骤如下：

1. 选择对象。
2. 在巡视窗口中选择“安全”组。
3. 选择“权限”下的操作员权限。
4. 激活该权限。

结果

操作员可以在运行系统中使用 <Tab> 键来选择该对象。

参见

键盘访问概述 (页 999)

8.2.2.19 设置 Tab 顺序

简介

在运行系统中可使用 <Tab> 键到达所有可操作对象。 可以使用“Tab 顺序”命令来定义操作员在运行系统中激活对象的顺序。

注意

在“输出”或“两种状态”模式下，在运行系统中无法使用 <Tab> 键到达对象。

可以选择在运行系统中按如下方式操作画面：

- 使用 <Tab> 键
- 使用鼠标
- 使用组态的热键

要求

- 活动画面中包含可操作对象。
- 未选择任何对象。
- 对象已启用，可以在运行系统中使用，并具有相应操作员权限。

步骤

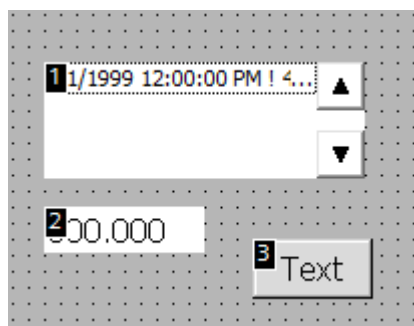
步骤如下：

1. 在“编辑”菜单中选择“编辑 Tab 顺序”命令。

Tab 顺序模式即激活。 所有可用对象的 Tab 顺序编号均显示出来。 隐藏对象的 Tab 顺序编号也会显示出来。

2. 在 Tab 顺序编辑模式下，单击各个可访问的对象，单击的顺序即为这些对象在运行系统中通过 <Tab> 激活的顺序。

下图显示了在画面中定义的 Tab 顺序。 在运行系统中，<Tab> 键首先激活报警视图（编号 1），然后激活 I/O 域（编号 2），接着激活按钮（编号 3）：



3. 要从 Tab 顺序中排除某一画面对象，请按住组合键 <Shift+Ctrl> 并单击所需对象。

Tab 顺序编号不再显示在该画面对象中。现在从 Tab 顺序中排除该画面对象。剩余的 Tab 顺序编号会自动减 1。

4. 要在 Tab 顺序中重新输入已排除的画面对象，请重复步骤 3。
画面对象将被作为 Tab 顺序中的第一个对象进行输入。

结果

在运行系统中，操作员可使用 <Tab> 键按照定义的顺序激活对象。

参见

键盘访问概述 (页 999)

8.2.2.20 实例

8.2.2.20 实例：插入和组态矩形

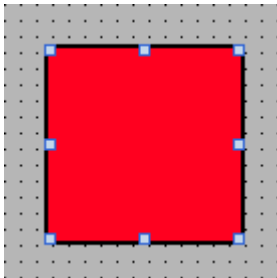
任务

在本实例中，要在画面中插入一个矩形。可组态下列属性：

- 名称 = "MyRectangle"
- 位置 = (20, 20)
- 大小 = (100,100)
- 颜色 = 红色
- 黑色边框 2 个像素宽

原理

矩形是封闭对象，可以用颜色或图案填充。可以调整矩形的高度和宽度，以允许进行水平和垂直对齐。



概述

创建矩形，请执行以下步骤：

- 插入矩形

- 组态矩形

参见

实例：插入矩形 (页 1003)

实例：组态矩形 (页 1003)

对象概述 (页 969)

8.2.2.20 实例：插入矩形

任务

在本实例中，插入并重命名矩形。请不要在名称中使用特殊字符？、"、/、\、*、<、>。

要求

- 画面打开。
- 打开巡视窗口。
- 打开“工具”任务卡。

步骤

插入矩形的步骤如下：

1. 在“工具”任务卡中单击“基本对象”选项板。
2. 将“矩形”对象拖放到画面上。
3. 在巡视窗口中选择“其它”。
4. 键入新名称“MyRectangle”。

结果

矩形现在已插入并已命名为“MyRectangle”。该矩形具有“矩形”对象的默认属性。

参见

实例：插入和组态矩形 (页 1002)

8.2.2.20 实例：组态矩形

任务

在此实例中，按如下方式组态矩形：

- 颜色 = 红色
- 黑色边框 2 个像素宽
- 位置 = (20, 20)
- 大小 = (100,100)

改变矩形的颜色

更改矩形颜色的步骤如下：

1. 选择矩形。
2. 在巡视窗口中“外观”下的“背景”区域中，选择红色作为“颜色”。
3. 在“背景”区域中，选择“实心”作为“填充图案”。
4. 在“边框”区域中，选择黑色作为“颜色”。
5. 在“边框”区域的“宽度”中，输入值“2”。
6. 在“边框”区域中，选择“实心”作为“样式”。

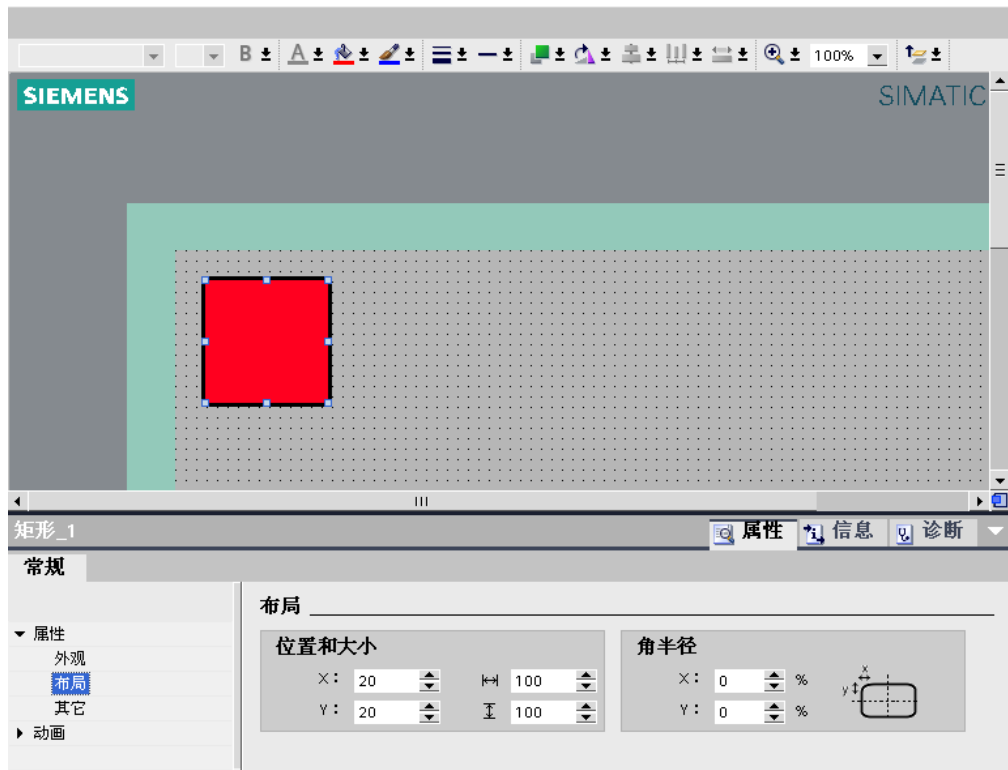
中间结果

矩形为红色，带有宽度为两个像素的黑色边框。

对矩形重新定位和调整大小

更改矩形位置和大小步骤如下：

1. 选择矩形。
2. 在巡视窗口中选择“布局”。



3. 在“位置和大小”下，将 X 坐标和 Y 坐标设置为“20”。
4. 为高度和宽度设置“100”。

结果

矩形放置在坐标 (20, 20) 处，宽度和高度均为 100 个像素。

参见

实例：插入和组态矩形 (页 1002)

8.2.3 使用文本列表和图形列表

8.2.3.1 使用文本列表

8.2.3.1 关于文本列表的基础知识

简介

在文本列表中，文本被分配给变量的值。例如，在组态期间，可以将文本列表分配给符号 I/O 域。这样将为对象提供要显示的文本。

文本列表在“文本列表”编辑器中创建。可在使用文本列表的对象上组态该文本列表和变量之间的接口。

注意

运行系统相关性

哪些对象可分配文本列表取决于运行系统。

应用

例如，文本列表用于显示符号 I/O 域中的下拉列表。

可以为文本列表中的文本组态多种语言。文本随后会在运行系统中以所设语言显示。

注意

HMI 设备相关性

文本列表是否可用取决于所使用的 HMI 设备。

功能

如果符号 I/O 域是显示域，则相关联的文本会因组态变量的值而异。如果符号 I/O 域是输入域，则当操作员在运行系统中选择了相应的文本时组态变量将采用相关联的值。

注意

运行系统相关性

对于未分配文本的变量值，其显示取决于运行系统：

- 显示和操作对象保持为空白。
 - 显示三个星号 ***。
-

多语言文本

可以为文本列表中的文本组态多种语言。为此，可在“项目”窗口的“语言支持”>“项目语言”下设置语言。

组态步骤

例如，要在符号 I/O 域中显示文本，需要执行以下步骤：

1. 创建文本列表
2. 为文本列表分配值/值范围
3. 在显示对象（例如，符号 I/O 域）中分配文本列表

参见

画面基础知识 (页 953)

创建文本列表 (页 1006)

将文本和值分配给范围文本列表 (页 1008)

将文本和值分配给位文本列表 (页 1009)

将文本和值分配给位号文本列表 (页 1010)

组态带有文本列表的对象 (页 1011)

8.2.3.1 创建文本列表

简介

通过文本列表，可将特定文本分配给值并在运行系统中输出（例如，在符号 I/O 域中）。可以指定符号 I/O 域的类型（例如，指定为纯输入域）。

注意

运行系统相关性

可以选择的 I/O 域类型取决于所用的运行系统。

列表有以下类型：

- 范围

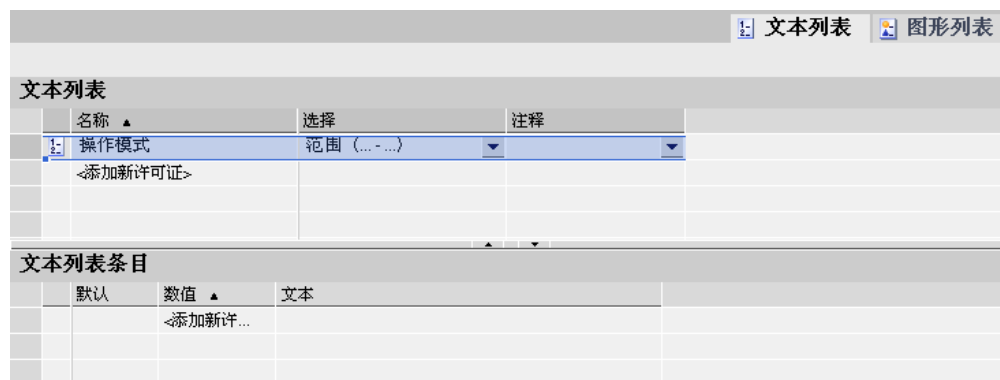
- 位
- 位号

使用“文本和图形列表”编辑器定义文本列表的内容和属性。

步骤

创建文本列表的步骤如下：

1. 打开“文本和图形列表”编辑器，然后在项目树中打开“文本列表”选项卡。



2. 单击上方工作区的第一个空表格行。
文本列表的巡视窗口即打开。
3. 为文本列表分配可指示其功能的名称。
4. 在“选择”下选择文本列表类型。
 - 范围 (... - ...)：当变量的值在指定范围内时，将显示文本列表中的文本。
 - 位 (0、1)：变量值为 0 时，显示文本列表中的某个文本。变量值为 1 时，则显示文本列表中的另一文本。
 - 位号 (0-31)：当变量具有指定位号的值时，将显示文本列表中的文本。
5. 输入文本列表的注释。

注意

HMI 设备相关性

文本列表是否可用取决于所使用的 HMI 设备。

注意

请不要在文本列表的文本中使用分号。分号是控制字符，会自动从文本中删除。

结果

文本列表创建完毕。

参见

关于文本列表的基础知识 (页 1005)

8.2.3.1 将文本和值分配给范围文本列表

简介

对于每个范围文本列表，都可指定值范围及其对应的显示文本。

要求

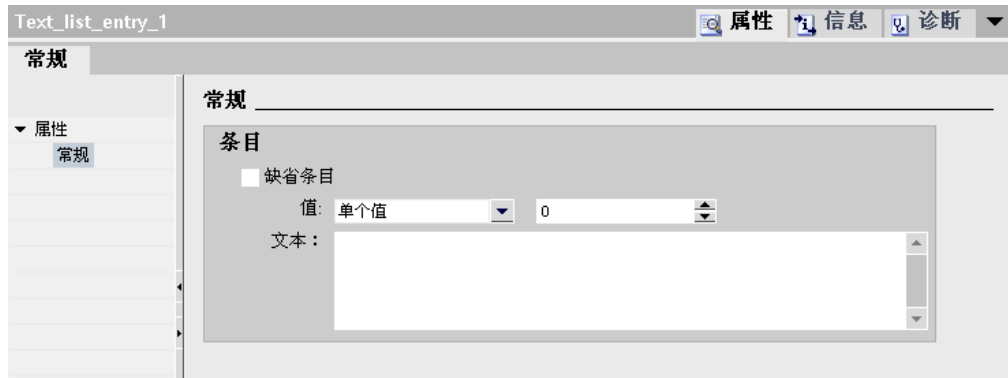
- 打开“文本和图形列表”编辑器。
- 打开“文本列表”选项卡。
- 已创建并选择一个范围文本列表。

步骤

将值和文本分配给范围文本列表的步骤如下：

1. 单击“列表项”编辑区域中的第一个空行。

此列表项的巡视窗口随即打开。



2. 选择取值范围类型：



- “范围”：最小变量值到最大变量值，例如 $1 \leq \text{值} \leq 21$
- “单个值”：恰好一个变量值，例如值 = 21

3. 在“值”下输入要显示文本列表中的文本所对应的值或值范围。
4. 在“文本”下输入当变量等于指定值或处于指定值范围内时将在运行系统中显示的文本。
5. 如果要在变量具有未定义的值时显示某个输入的文本，则激活“默认条目”。每个列表只能有一个默认条目。
6. 为同一个文本列表中的更多取值范围创建更多对应列表项。

注意

请不要在文本列表的文本中使用下列特殊字符之一：?、"、/、\、*、<、> 和分号。控制字符在文本中会自动删除。

结果

范围文本列表创建完毕。运行系统中将显示的文本被分配给可能的值。

参见

[关于文本列表的基础知识 \(页 1005\)](#)

8.2.3.1 将文本和值分配给位文本列表

简介

对于每个文本列表，都可指定位值及其对应的显示文本。

要求

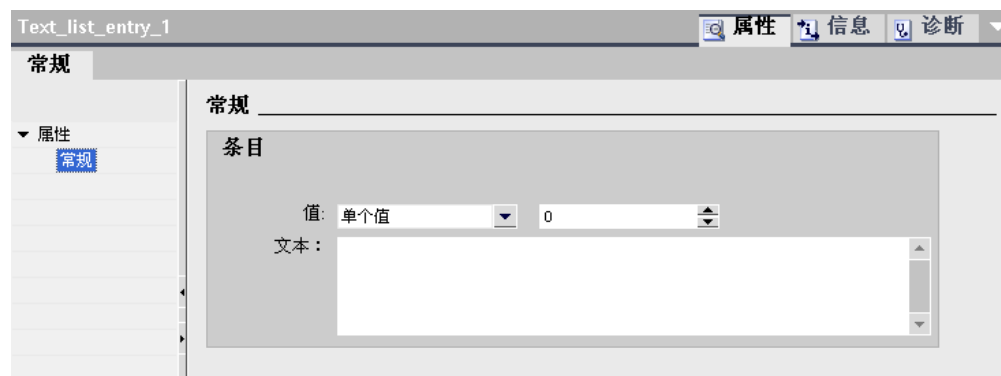
- 打开“文本和图形列表”编辑器。
- 打开“文本列表”选项卡。
- 已创建并选择一个位文本列表。

步骤

将值和文本分配给位文本列表的步骤如下：

1. 单击“列表项”编辑区域中的第一个空行。

此列表项的巡视窗口即打开。



2. 在“值”下输入“0”。
3. 输入当变量值为“0”时在运行系统中显示的文本。

注意

请不要在文本列表的文本中使用下列特殊字符之一：？、"、/、\、*、<、> 和分号。控制字符在文本中会自动删除。

4. 单击“列表项”编辑区域中的下一个空行。
下一个列表项的巡视窗口即打开。
5. 在“值”下输入“1”。
6. 输入当变量值为“1”时在运行系统中显示的文本。

结果

位文本列表创建完毕。运行系统中将显示的文本被分配给可能的值“0”和“1”。

参见

关于文本列表的基础知识 (页 1005)

8.2.3.1 将文本和值分配给位号文本列表**简介**

对于每个位号文本列表，都可指定位号及其对应的显示文本。

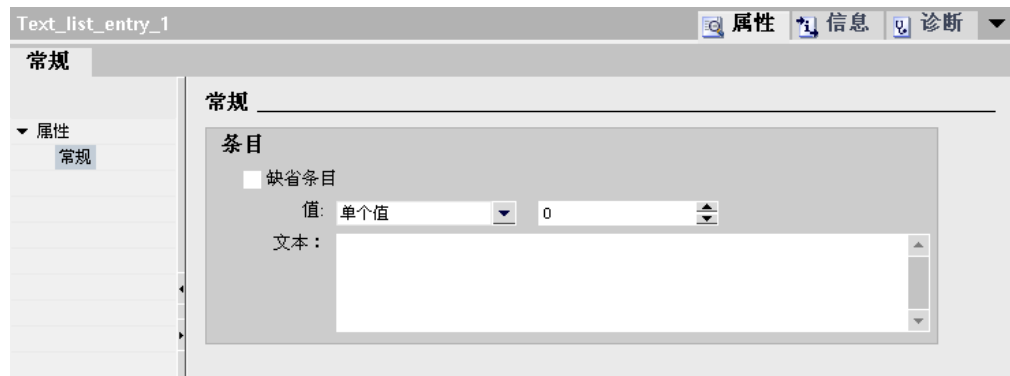
要求

- 打开“文本和图形列表”编辑器。
- 打开“文本列表”选项卡。
- 已创建并选择一个位号文本列表。

步骤

将值和文本分配给位号文本列表的步骤如下：

1. 单击“列表项”编辑区域中的第一个空行。
此列表项的巡视窗口即打开。
2. 在“值”下输入要显示文本列表中的文本所对应的位号。



3. 输入当变量具有指定位号时在运行系统中显示的文本。
4. 如果要在变量具有未定义的值时显示某个输入的文本，则激活“默认条目”。每个列表只能有一个默认条目。
5. 为同一个文本列表中的更多位号创建更多对应列表项。

注意

请不要在文本列表的文本中使用下列特殊字符之一：？、"、/、\、*、<、> 和分号。控制字符在文本中会自动删除。

结果

位号文本列表创建完毕。运行系统中将显示的文本被分配给指定的位号。

参见

关于文本列表的基础知识 (页 1005)

8.2.3.1 组态带有文本列表的对象

简介

在显示和操作对象中指定输出值以及值应用，该对象将在运行系统显示文本列表中的文本。根据需要组态这些对象的属性。

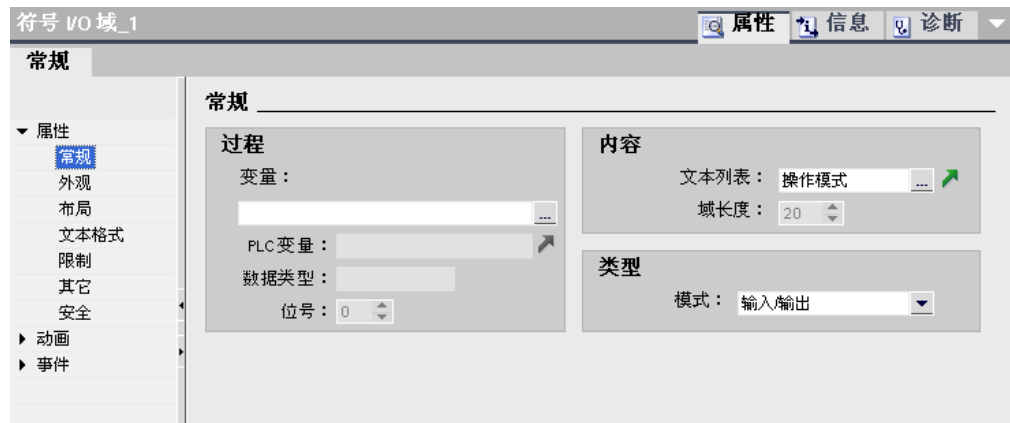
要求

- 已在“文本和图形列表”编辑器中创建一个文本列表。
- 已定义文本列表的值并已为其分配文本。
- 打开“画面”编辑器。
- 具有符号 I/O 域的画面已打开，并已编辑对象。

步骤

执行下列步骤，在运行系统中符号 I/O 域内显示文本列表中的文本：

1. 在“其它”下为对象分配可指示其功能的名称。
2. 在“文本列表”域中的“常规”下选择要在运行系统中显示其中文本的文本列表。



3. 在“模式”域中选择对象要显示的数据。

注意

运行系统相关性

符号 I/O 域可用的域类型因运行系统而异。

4. 在“可见条目”下指定项目中可见的条目数。
5. 在“特性”下激活“退出时应用值”。
在运行系统中，退出对象时将会应用这些值。

结果

当变量具有指定值时，在运行系统的符号 I/O 域中会显示文本列表中定义的文本。

参见

[关于文本列表的基础知识 \(页 1005\)](#)

8.2.3.2 使用图形列表

8.2.3.2 图形列表的基本原理

简介

可能的变量值被分配给图形列表中的特定图形。组态期间，可以为按钮或图形 I/O 域创建图形列表。这样将为对象提供要显示的图形。

图形列表在“文本和图形列表”编辑器中创建。可在使用图形列表的对象上组态该图形列表和变量之间的接口。

应用

可以组态图形列表以用于以下情况：

- 带图形 I/O 域的下拉列表
- 按钮的状态特定的图形

图形列表中的图形可以组态为多种语言。图形随后会在运行系统中以所设语言显示。

注意
可用于特定 HMI 设备

图形列表是否可用取决于所使用的 HMI 设备。

图形源

图形可以从下列来源添加到图形列表中：

- 通过从图形浏览器中选择
- 通过选择现有文件
可使用下列文件类型：
.bmp、.ico、*.cur、*.emf、*.wmf、*.gif、*.tif、*.jpeg 和 *.jpg。
- 通过创建新文件

功能

如果图形 I/O 域是显示域，则相关联的图形会因组态变量的值而异。如果图形 I/O 域是输入域，则当操作员在运行系统中选择了图形时组态变量将采用相关联的值。

注意
可用于特定运行系统

对于未分配图形的变量值，其显示取决于运行系统：

- 显示和操作对象保持为空
 - 显示已定义的默认图形
-

组态步骤

例如，要在图形 I/O 域中显示图形，需要完成以下任务：

1. 创建图形列表
2. 为图形列表分配值/值范围
3. 在显示对象（例如，图形 I/O 域）中分配图形列表

参见

画面基础知识 (页 953)
创建图形列表 (页 1014)
将图形和值分配给范围图形列表 (页 1015)

将图形和值分配给位图形列表 (页 1016)
 将图形和值分配给位号图形列表 (页 1017)
 组态带有图形列表的对象 (页 1018)

8.2.3.2 创建图形列表

简介

可以通过图形列表将特定图形分配给变量值，并在运行系统中的图形 I/O 域中输出。例如，将图形 I/O 域的类型指定为纯输出域。

注意

运行系统相关性

可以选择的 I/O 域类型取决于所用的运行系统。

注意

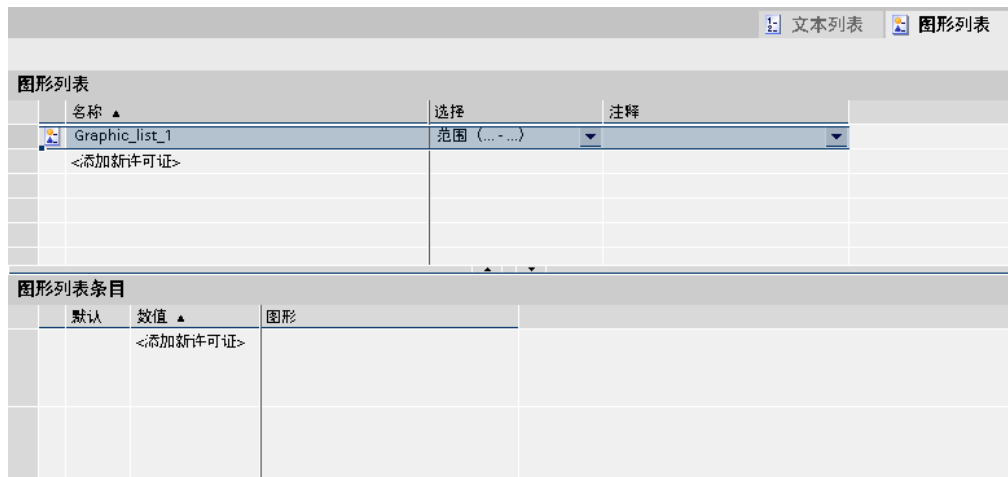
HMI 设备相关性

图形列表是否可用取决于所使用的 HMI 设备。

步骤

执行下列步骤，创建图形列表：

1. 打开“文本和图形列表”编辑器，然后在项目树中打开“图形列表”选项卡。



2. 单击上方工作区的第一个空表格行。

图形列表的巡视窗口随即打开。



3. 为图形列表分配一个有意义的名称。
4. 在“选择”下选择图形列表类型：
 - 范围 (... - ...)：当变量的值在指定范围内时，将显示图形列表中的图形。
 - 位 (0、1)：变量值为 0 时，显示图形列表中的某一图形。变量值为 1 时，则显示图形列表中的另一图形。
 - 位号 (0-31)：当变量值为分配的位号时，将显示图形列表中的图形。
5. 输入图形列表的注释。

结果

范围图形列表创建完毕。

参见

图形列表的基本原理 (页 1012)

8.2.3.2 将图形和值分配给范围图形列表

简介

对于每个范围图形列表，可指定在哪个值范围显示哪个图形。

要求

- 打开“文本和图形列表”编辑器。
- 打开“图形列表”选项卡。
- 已创建和选择一个范围图形列表。

步骤

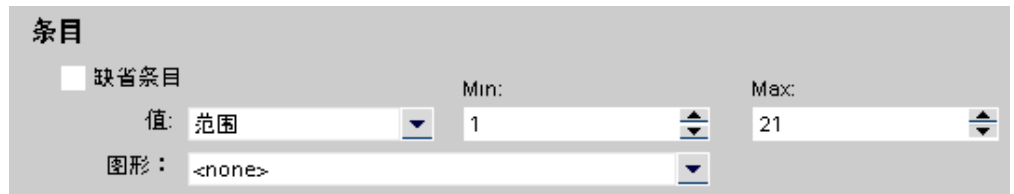
将值和图形分配给范围图形列表的步骤如下：

1. 单击“列表项”编辑区域中的第一个空行。
此列表项的巡视窗口随即打开。

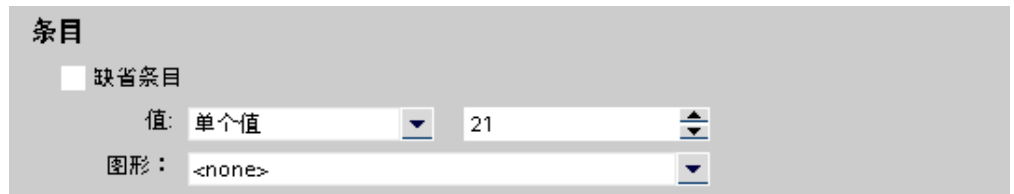


2. 在“值”下选择取值范围类型：

- “范围”：变量最小值到变量最大值，例如 $1 \leq \text{值} \leq 21$



- “单个值”：恰好一个变量值，例如值 = 21



3. 在“值”下输入要显示图形列表中的图形所对应的值或值范围。

4. 在“图形”下选择当变量具有指定值或处于指定值范围内时将在运行系统中显示的图形。

5. 如果要在变量具有未定义的值时显示某个指定的条目，则激活“默认条目”。每个列表只能有一个默认条目。

结果

范围图形列表创建完毕。运行系统中将显示的图形被分配给可能的值。

参见

图形列表的基本原理 (页 1012)

8.2.3.2 将图形和值分配给位图形列表

简介

对于每个图形列表，可指定在哪个位值显示哪个图形。

要求

- 打开“文本和图形列表”编辑器。

- 打开“图形列表”选项卡。
- 已创建和选择一个位图形列表。

步骤

将值和图形分配给位图形列表的步骤如下：

1. 单击“列表项”编辑区域中的第一个空行。
此列表项的巡视窗口随即打开。



2. 在“值”下输入“0”。
3. 选择当变量值为“0”时在运行系统中显示的图形。
4. 单击“列表项”编辑区域中的下一个空行。
下一个列表项的巡视窗口即打开。
5. 在“值”下输入“1”。
6. 选择当变量值为“1”时在运行系统中显示的图形。

结果

位图形列表创建完毕。运行系统中显示的图形分配给了值“0”和“1”。

参见

图形列表的基本原理 (页 1012)

8.2.3.2 将图形和值分配给位号图形列表

简介

对于每个位号图形列表，可以指定在哪个位号显示哪个图形。

要求

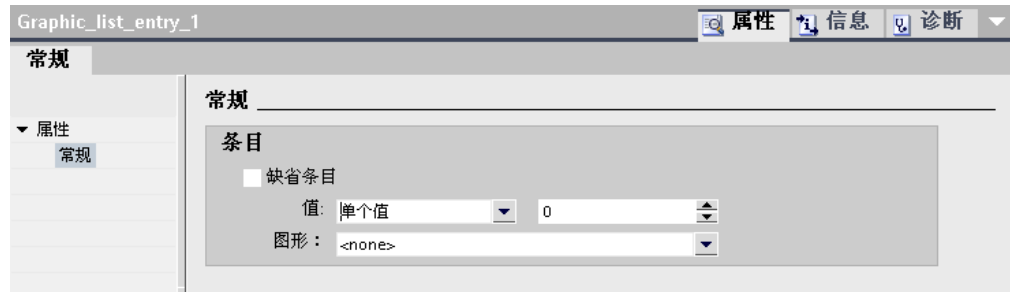
- 打开“文本和图形列表”编辑器。
- 打开“图形列表”选项卡。
- 已创建和选择一个位号图形列表。

步骤

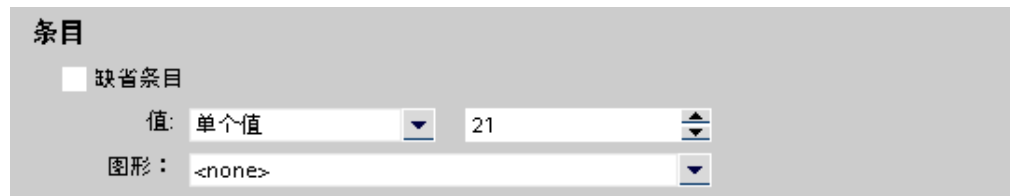
将值和图形分配给位号图形列表的步骤如下：

1. 单击“列表项”编辑区域中的第一个空行。

此列表项的巡视窗口即打开。



2. 在“值”下输入要显示图形列表中的图形所对应的位号。



3. 在“图形”下选择当变量具有指定位号值时将在运行系统中显示的图形。
4. 如果要在变量具有未定义的值时显示某个指定的条目，则激活“默认条目”。每个列表只能有一个默认条目。
5. 为同一个图形列表中的其它位号创建对应的列表项。

结果

位号图形列表创建完毕。运行系统中显示的图形分配给了指定位号。

参见

图形列表的基本原理 (页 1012)

8.2.3.2 组态带有图形列表的对象

简介

在显示和操作对象中指定图形列表的输出值和值应用，此对象将在运行系统中显示图形列表中的图形。根据需要组态这些对象的属性。

要求

- 已在“文本和图形列表”编辑器中创建一个图形列表。已定义值。图形已分配给值。
- 打开“画面”编辑器。
- 具有图形 I/O 域的画面已打开，并已编辑对象。

步骤

执行下列步骤，在运行系统中图形 I/O 域内显示图形列表中图形：

1. 在“其它”下为对象分配可指示其功能的名称。
2. 在“图形列表”域中的“常规”下选择要在运行系统中显示其中指定图形的图形列表。



3. 在“类型”下的“模式”域中选择对象要显示的数据。

注意 运行系统相关性

图形 I/O 域可用的域类型因运行系统而异。

结果

当变量具有指定值时，运行系统的图形 I/O 域中会显示图形列表中定义的图形。

参见

图形列表的基本原理 (页 1012)

8.2.4 画面动态化

8.2.4.1 关于动态化的基础知识

简介

在 WinCC 中，可将对象属性进行动态化，用以实现工厂到 HMI 设备的映射并显示过程顺序。例如罐的映射，罐中液位的上升或下降与过程值相关。

对象属性的动态化

可以将动态属性分配给任何图形对象。为此可使用预定义的动画。

操作员控制对象对事件（如鼠标单击）的反应。可以组态系统函数对事件作出反应。更多信息，请参考“[使用系统函数 \(页 1231\)](#)”。

动态化选项取决于涉及的对象。复制对象时，其动态化功能也会随之复制。

动画

注意

HMI 设备相关性

“控制启用”动画并非适用于每个 HMI 设备。

WinCC 帮助您通过预定义的动画来实现动态化。如果要使对象属性动态化，首先使用工具箱或对象巡视窗口的工具组态所需要的动画。然后在巡视窗口中根据项目要求自定义动画。

动画可用于实现以下对象属性的动态化：

- 外观
更改对象的外观，如颜色。
- 位置
将画面对象动画化。
- 可见性
决定对象是显示还是隐藏。
- 操作员控制对象的可操作性
启用或锁定对象操作。

可在工具箱中自定义动画的默认属性，其方法与自定义其它图形对象默认属性的方法相同。

参见

[组态新动画 \(页 1021\)](#)

8.2.4.2 元素和基本设置

简介

在 WinCC 中，可使用动态对象将工厂映射到 HMI 设备并显示过程顺序。

对于画面中组态的所有图形对象，都可以实现动态化。实际可以使用的动态化功能和事件取决于所选的对象。

注意

HMI 设备相关性

“控制启用”动画并非适用于每个 HMI 设备。

“动画”任务卡

“动画”任务卡包含用于组态动画的以下组和元素：

- 移动：水平、垂直、对角线和直接移动
- 布局：外观和可见性
- 其它：激活对象

注意 精简面板

“动画”任务卡不可用于精简面板。

巡视窗口

巡视窗口显示在“动画”组中为选定对象组态的所有动画：

- 对角线、水平、垂直或直接移动：将画面对象动画化。
- 设计：对象更改其颜色和闪烁属性等外观。
- 可见性：决定对象是显示还是隐藏。
- 激活对象：决定对象是否能够使用。

8.2.4.3 动画

8.2.4.3 组态新动画

简介

可使用简单的鼠标操作来组态图形对象的动画。对于每个对象，只能组态同类型的一个动画。

要求

- 已打开包含至少一个动态对象的画面。
- 打开巡视窗口。
- 显示“工具箱”窗口。

步骤

组态动画的步骤如下：

1. 打开要实现动态化的对象的巡视窗口或属性列表。
2. 在“动画”组中，双击“新建动画”项。
3. 选择要使用的动画。
4. 单击“添加”。

其它方法

1. 在“动画”任务卡中打开包含所需动画的对象组。
2. 将动画拖动到要实现动态化的对象上。
或者：从工具箱中选择所需动画。然后单击要实现动态化的对象。

注意

精简面板

“动画”任务卡不可用于精简面板。

结果

动画即出现在对象的巡视窗口中。现在可以组态动画。

管理组态的动画

在巡视窗口中，“动画”快捷菜单包含以下用于管理动画的菜单项：

- “复制动画”：此菜单项用于复制选定的动画。
- “插入动画”：此菜单项用于将动画从剪贴板插入另一对象的巡视窗口。
- “删除动画”：此选项用于删除不再需要的动画。

参见

对象外观动态化 (页 1022)

组态移动 (页 1024)

对象可见性动态化 (页 1026)

用于对象组和多项选择的动画 (页 1027)

关于动态化的基础知识 (页 1019)

8.2.4.3 对象外观动态化

简介

可以动态控制对象的前景色、背景色以及其闪烁属性。在运行系统中，对象的外观由变量值的变化进行控制。变量获得某个值或在某个范围内改变其值时，对象根据其组态改变颜色或闪烁。

外观变化

可以组态多少种外观取决于变量数据类型的设置：

- “范围”：
考虑变量的范围值。可以定义任意数目的值或范围来控制对象的外观。不能定义重叠的范围。
- “多个位”

使用“多个位”时，考虑变量的值。变量的值可以是介于 0 至 31 之间的整数。可将其中的每个值分配给一种对象外观形式。“多位”数据类型不允许定义范围，而只能定义“从”列中的值。

- “单个位”

仅考虑指定的变量位。可为对象外观定义两种布局方式，即“位 = 1”和“位 = 0”时的布局方式。“单个位”数据类型不允许定义范围。

注意

您可以使控制元素的外观随操作员输入而变。为此，在 I/O 域中使用相同的变量来保存输入和控制外观。

实例：在运行系统中，I/O 域的背景色随输入而变。

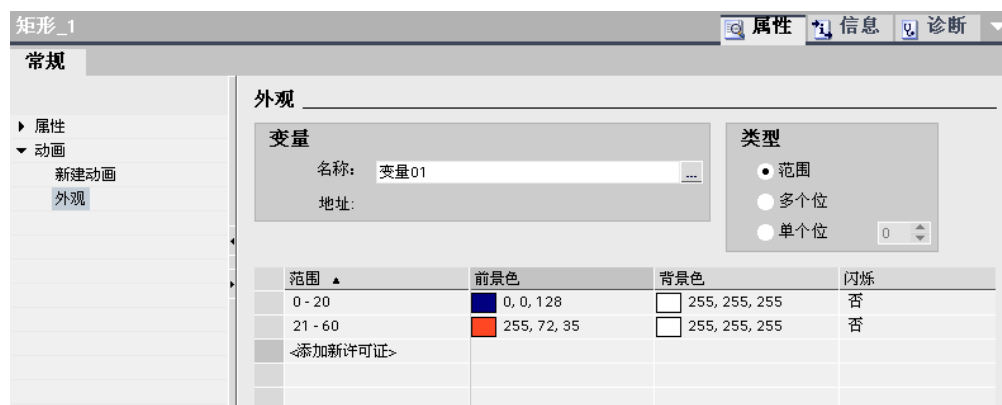
要求

- 已打开包含至少一个动态对象的画面。
- 打开巡视窗口。
- 显示“工具箱”窗口。

步骤

设置对象闪烁属性的步骤如下：

1. 选择要动态控制的画面对象。
对象属性即显示在巡视窗口中。
2. 在巡视窗口中，单击“动画”>“新建动画”。
选定对象可用的动画即显示出来。
3. 选择“外观”动画。
将显示动画的参数。
4. 单击“变量”打开变量列表，然后选择要用于控制对象外观的变量。
变量即出现在“变量”域中。
5. 选择变量的数据类型。用来控制对象外观的数值或范围的数目取决于变量的数据类型。
6. 单击“值”或“范围”列的第一个单元格。



7. 输入变量值：

在“值”列中为“单个位”和“多个位”输入变量值。

在“范围”列中输入变量范围，例如 0 - 60。

8. 在“前景色”和“背景色”列中，选择当变量达到设定值或范围时对象在运行系统中将显示的前景色和背景色。
9. 在“闪烁”列表中为对象选择一种闪烁模式。
10. 如果要为其它变量值或变量范围定义外观，则重复步骤 6 至 10。

结果

在运行系统中，对象的闪烁响应和颜色根据变量中返回的过程值动态改变。

参见

组态新动画 (页 1021)

8.2.4.3 组态移动

简介

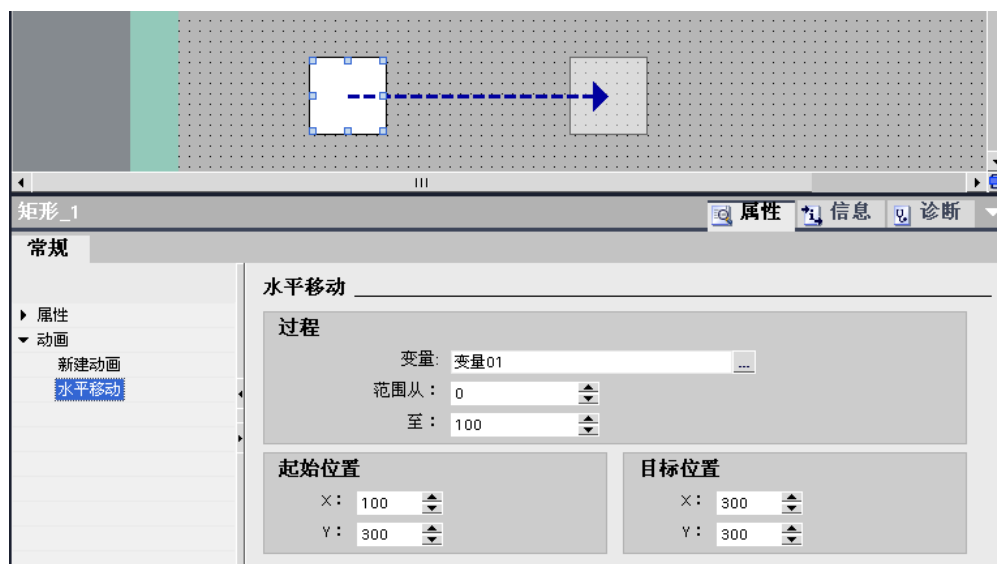
可以将动态对象组态为沿特定路径移动，或者相对于当前位置移动特定数目的像素。对象的移动通过变量进行控制。变量每更新一次，对象便移动一步。

每个对象只能设定一种类型的移动。可使用以下移动类型：

- “对角线移动”、“水平移动”或“垂直移动”：
对象沿定义了起点和终点的路径移动。对象沿该路径的相对位置由变量值确定。
- “直接移动”
对象沿着 X 和 Y 坐标轴移动特定数目的像素。移动量由两个变量的绝对值定义。

图形设计支持

“画面”编辑器可以图形方式编辑对角线移动、水平移动和垂直移动。组态期间，画面上会显示对象的透明副本。此副本通过箭头连接到源对象。将对象副本移动到所需目标时，系统将与移动对应的像素值自动保存到巡视窗口中。



要求

- 已打开包含至少一个动态对象的画面。
- 巡视窗口已打开。
- 显示“工具箱”窗口。

组态“对角线移动”、“水平移动”或“垂直移动”

组态动画的步骤如下：

1. 选择要动态控制的画面对象。
对象属性即显示在巡视窗口中。
2. 在巡视窗口中，单击“动画”>“新建动画”。
选定对象可用的动画即显示出来。
3. 选择移动类型。
将显示动画的参数。
4. 选择用于控制移动的变量。
对象的透明副本显示在工作区中，该副本与源对象通过箭头互连。
5. 将对象副本移动到相关的目标。系统自动输入巡视窗口中最后位置的像素值。
6. 根据需要自定义变量的值范围。

组态“直接移动”

组态“直接移动”动画的步骤如下：

1. 选择要动态控制的画面对象。
对象属性即显示在巡视窗口中。
2. 在巡视窗口中，单击“动画”>“新建动画”。

选定对象可用的动画即显示出来。

3. 选择“直接移动”。
4. 选择用于控制 X 方向偏移的变量。
5. 选择用于控制 Y 方向偏移的变量。

结果

在运行系统中，对象会随着用于控制移动的变量值的每个变化而移动。移动的方向对应于组态的移动类型。

参见

组态新动画 (页 1021)

8.2.4.3 对象可见性动态化

简介

可以动态控制对象的可见性。如果变量值在组态的范围内，对象可在运行系统中显示或隐藏。“简单配方视图”和“简单报警视图”对象始终可见。

应用实例

通过“可见性”属性的动态化，可以向画面输出报警，例如，在变量值超出临界范围时触发报警。当变量值返回到非临界范围时，报警被清除。

要求

- 打开含要在运行系统中显示或隐藏的对象画面。
- 打开巡视窗口。

步骤

1. 选择要动态控制的控制元素。
对象属性即显示在巡视窗口中。
2. 在巡视窗口中，单击“动画”组中“新建动画”。
选定对象可用的动画即显示出来。
3. 选择“可见性”动画。
将显示动画的参数。
4. 设置所需“状态”：“可见”或“隐藏”。
如果没有定义用于控制状态的变量，则会在运行系统启动时设置所选状态。
5. 要使用变量控制对象的显示，请选择变量并定义范围。
只要变量值在设定范围内，便会应用设定的对象状态。

结果

对象在运行系统中根据变量值而显示或隐藏。

参见

组态新动画 (页 1021)

8.2.4.3 用于对象组和多项选择的动画

将动画应用于对象组

可以为对象组组态对象组中各对象所支持的所有动画。如果为对象组组态动画，则此动画将应用于支持此动画的所有单个对象。

更改用于多个对象的动画

对于多项选择，为参考对象组态的动画会显示在巡视窗口中。这些动画可以照常更改。所做更改将应用于多项选择中支持所组态动画的所有对象。这意味着，即使尚未组态动画的对象也将拥有参考对象的动画。

应用实例

同时选择一个按钮和一个圆。按钮是参考对象。该按钮已经组态了“外观”动画，所以它会出现在该多项选择的巡视窗口中。如果激活“动画”组中的“闪烁”属性，则“外观”动画的设置将同时应用于按钮和圆。

为多个对象组态新动画

如果为多项选择的对象组态新动画，则此动画将应用于支持所组态动画的所有选定对象。如果新动画替换现有动画，则会出现安全提示。

应用实例

选择一个圆和一个矩形。圆已经组态了“对角线移动”动画。为该多项选择组态“水平移动”动画。此动画将应用于矩形，因为矩形尚未组态“移动”类型的动画。对于圆，系统会提示您确认是否要用新的“水平移动”动画替换现有“对角线移动”动画。

参见

组态新动画 (页 1021)

8.2.5 使用功能键

8.2.5.1 使用功能键

简介

功能键是 HMI 设备上实际按键，可以对这些键的功能进行组态。可为“按下按键”和“释放按键”事件组态函数列表。

可以为功能键分配全局或局部功能。

注意

可用于特定 HMI 设备

功能键并非在所有 HMI 设备上均可用。

全局功能键

全局功能键始终触发同样的操作，而不管所显示的画面为何。

在“全局画面”编辑器中组态全局功能键。全局分配适用于设定的 HMI 设备的所有画面。

全局功能键可极大地减少设计工作量，这是因为无需为各个画面分配这些全局键。

画面中的局部功能键

画面中的局部功能键可触发各画面中不同的操作。这种分配只适用于那些已在其中定义了功能键的画面。

使用局部功能键，可覆盖模板中的全局功能键和局部功能键。

注意

即使带有局部功能键的画面与警报视图或警报窗口重叠，功能键在运行时也依然有效。带有小显示屏的 HMI 设备更有可能出现这种情况。

模板中的局部功能键

在模板中分配的局部功能键对基于该模板的所有画面有效。在每个画面中，这些功能键可触发不同的操作。在“画面”编辑器模板中分配模板的功能键。模板中的功能键局部分配可覆盖功能键的全局分配。

热键分配

可以为控制对象分配热键，如按钮。可用热键取决于 HMI 设备。

注意










可为功能键分配局部或全局操作。如果还为功能键分配了热键，则热键功能可在运行系统中执行。

图形

当功能键直接置于显示的内容旁，可为其分配图形，以使功能键的功能更为清晰。

显示分配

表格 8-1 下表列出了用符号来显示各功能键的分配：

功能键	描述
	未分配
	全局分配
	在模板中局部分配
	局部分配
	局部分配（模板的局部分配可覆盖全局分配）
	局部分配（局部分配可覆盖全局分配）
	局部分配（局部分配可覆盖模板的局部分配）
	局部分配（局部分配可覆盖模板的局部分配，模板的局部分配已覆盖全局分配）
	使用画面浏览分配按钮

注意
精简面板

“画面浏览”编辑器不可用于精简面板。

参见

画面基础知识 (页 953)
全局分配功能键 (页 1030)
功能键的局部分配 (页 1031)
将功能键分配给函数 (页 1032)
为功能键分配操作员权限 (页 1034)
将功能键分配给图形 (页 1035)
实例：将功能键用于画面浏览 (页 1036)

8.2.5.2 全局分配功能键

简介

在“全局画面”编辑器中定义功能键的全局分配。全局分配适用于设定的 HMI 设备的所有画面。

注意
可用于特定 HMI 设备

功能键并非在所有 HMI 设备上均可用。

要求

- 已打开项目。
- 巡视窗口已打开。

步骤

执行下列步骤，为功能键分配基于画面的功能：

1. 要打开“全局画面”编辑器，请双击“项目”窗口的“画面管理”组中的“全局画面”。
2. 选择所需功能键。
功能键的属性显示在巡视窗口中。



3. 在“常规”下，为功能键组态图形和操作员权限。
4. 在“事件”下，为所需的事件组态函数列表。

结果

如果局部分配不覆盖全局分配，则在设定的 HMI 设备的所有画面中，功能键分配随输入的不同而不同。

参见

使用功能键 (页 1028)

8.2.5.3 功能键的局部分配

简介

可以全局分配功能键和局部分配功能键。功能键的局部分配仅对在其中定义局部分配的画面或模板有效。下列局部功能键可用：

- 画面的局部功能键
为每个画面的功能键分配不同的功能。这种分配只适用于已在其中定义了功能键的画面。
- 模板的局部功能键
在模板中分配功能键。分配适用于基于该模板的所有画面，且不会被其中一个画面的局部分配所覆盖。

功能键的局部分配将覆盖功能键的全局分配。

注意

可用于特定 HMI 设备

功能键并非在所有 HMI 设备上均可用。

使用现有分配

在巡视窗口中，使用现有分配的选项包括：

- 在模板中：“使用全局分配”

- 在画面中：
 - 基于模板的画面：“使用局部模板”
 - 不基于模板的画面：“使用全局分配”

“使用局部模板”选项包括使用模板中的局部分配和使用全局分配。

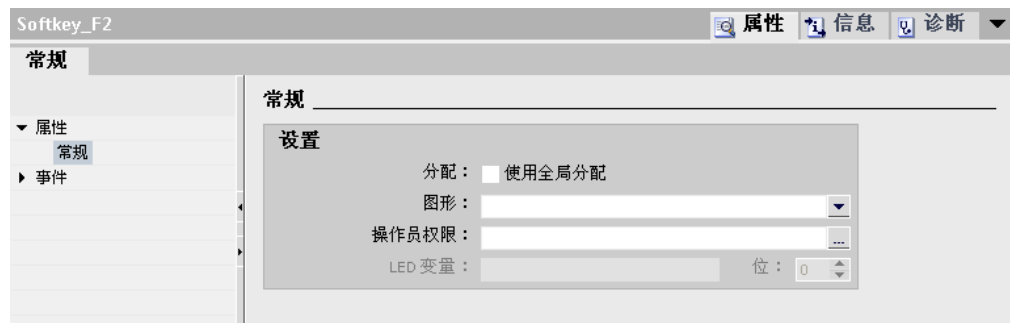
要求

- 已打开要局部分配功能键的画面或模板。
- 巡视窗口已打开。

步骤

步骤如下：

1. 在画面或模板中选择所需功能键。
功能键的属性显示在巡视窗口中。
2. 在巡视窗口中单击“常规”。



3. 禁用“使用局部模板”或“使用全局分配”选项。
4. 在“常规”下，为功能键组态图形和操作员权限。
5. 在“事件”下，为所需的事件组态函数列表。

结果

在画面或模板中为功能键分配组态的功能。

参见

[使用功能键 \(页 1028\)](#)

8.2.5.4 将功能键分配给函数

简介

功能键可以具有两种状态：

- 按下：由“按下按键”事件定义。
- 释放：由“释放按键”事件定义。

这两个事件均在功能键的巡视窗口中进行组态。可向任意事件分配包含系统函数或脚本的函数列表。运行系统中，该函数列表的执行是事件驱动的。

注意
可用于特定 HMI 设备

功能键并非在所有 HMI 设备上均可用。

注意
精简面板

脚本不可用于精简面板。

要求

要为功能键分配全局功能：

- “全局画面”编辑器已打开。

要为功能键分配局部功能：

- 想要在其中分配功能键的画面已打开。

如果要在模板中局部分配功能键：

- 想要在其中分配功能键的模板已打开。
- 巡视窗口已打开。

步骤

步骤如下：

1. 选择想要定义的功能键。
功能键的属性显示在巡视窗口中。
2. 在巡视窗口的“常规”组中的“属性”下，为所需结果组态函数列表。

结果

运行系统中，操作员按下或释放功能键时，函数列表被执行。

参见

使用功能键 (页 1028)

8.2.5.5 为功能键分配操作员权限

简介

在 WinCC，可以在运行系统中为功能键分配操作员权限。这允许在组态项目时将功能键的访问限制于特定人员或操作员组。只有经过授权的人员才能更改运行系统中的重要参数和设置。

要避免操作员错误和增强设备或机器的安全性，需要进行组态访问保护。

注意**可用于特定 HMI 设备**

功能键并非在所有 HMI 设备上均可用。

要求

- 已定义用户组。

要保护全局功能键：

- “全局画面”编辑器已打开。

如果要保护画面或模板的局部功能键：

- 包含该功能键的画面或模板已打开。
- 巡视窗口已打开。

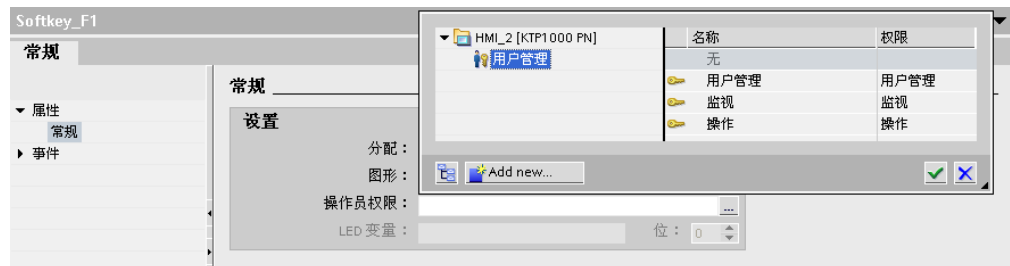
步骤

请按如下步骤进行操作：

1. 选择相关的功能键。

功能键的属性显示在巡视窗口中。

2. 在巡视窗口中单击“常规”。



3. 在“授权”列表中，选择希望允许其具有对功能键的运行系统访问权限的用户组。

结果

操作员授权已组态。

参见

使用功能键 (页 1028)

8.2.5.6 将功能键分配给图形

简介

为了使键的功能更清晰，可以在功能键旁的画面中插入图形。只能将图形分配给那些与 HMI 设备的画面边缘相接的功能键。

注意

可用于特定 HMI 设备

功能键并非在所有 HMI 设备上均可用。

要求

要将图形分配给全局功能键：

- “全局画面”编辑器已打开。

如果要将图形分配给画面或模板中的局部功能键：

- 包含相应功能键的画面或模板已打开。
- 巡视窗口已打开。
- 已经为功能键创建了图形。

步骤

请按如下步骤进行操作：

1. 选择相关的功能键。

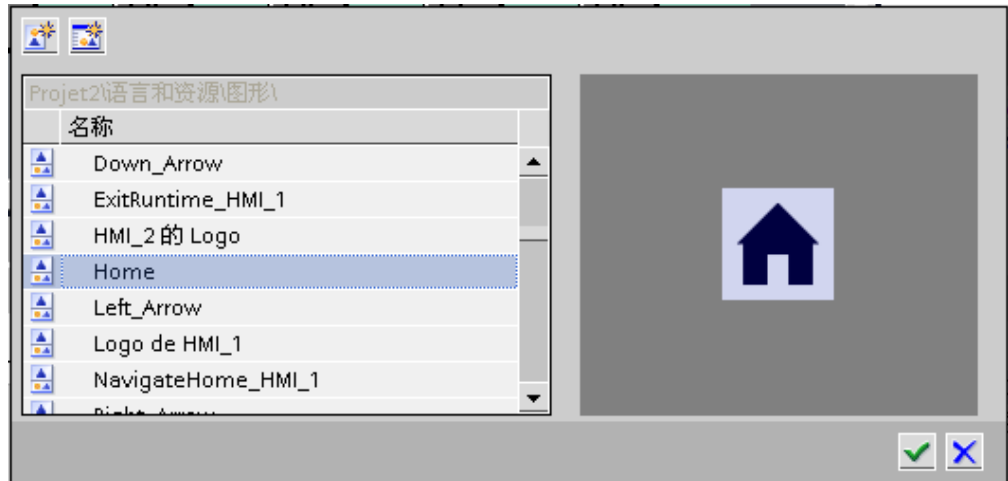
功能键的属性显示在巡视窗口中。

2. 在巡视窗口中单击“常规”。



3. 单击列表的“图形”区域。

显示 WinCC 项目的图形浏览器。左侧的窗格显示存储在浏览器中的外部图形。右侧的窗格显示在浏览器中选择的图形的预览。



使用



和



图标，可以显示缩略图形式的集合或列表形式的集合。

要打开并编辑具有相关图形程序的 OLE 对象，请双击该对象。

4. 在浏览器中单击所需图形，或将相关图形存储在图形浏览器中。
右窗格中将显示图形预览。
5. 单击“选择”，将图形添加到画面中。
单击“清除”，将图形从画面中删除。

结果

图形显示在功能键旁。

参见

使用功能键 (页 1028)

8.2.5.7 实例：将功能键用于画面浏览

任务

在本实例中，在画面中创建一个局部功能键。操作员按下该功能键时，将使画面切换至预定义的画面，例如“Boiler 2”。

注意

可用于特定 HMI 设备

功能键并非在所有 HMI 设备上均可用。

要求

- 想要在其中分配功能键的画面已打开。
- 已经创建了“Boiler 2”画面。
- 巡视窗口已打开。

步骤

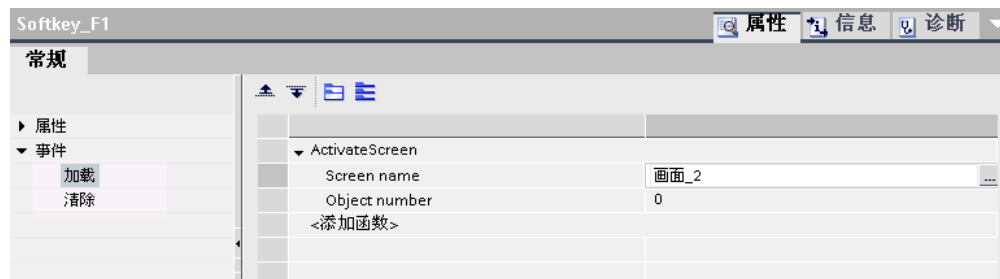
执行下列步骤，使用“ActivateScreen”函数：

1. 选择所需功能键。
功能键的属性显示在巡视窗口中。
2. 单击“常规”。
3. 要覆盖全局分配，禁用“使用局部模板”选项。
4. 在“事件”下，单击“按下按键”。



5. 从列表中选择“ActivateScreen”系统函数。

该“ActivateScreen”函数将出现在“函数列表”对话框中，还将出现“画面名称”和“对象号”参数。



6. 从“画面名称”列表中选择“Boiler 2”画面。

结果

在运行时操作员通过按下所选的功能键切换至“Boiler 2”画面。

参见

使用功能键 (页 1028)

8.2.6 使用层

8.2.6.1 有关使用层的基础知识

层

要在一个画面中完成各对象的分类编辑，请使用层。一个画面由 32 个层组成，可为层指定任何名称。如果将对象分配给层，也就定义了画面深度。第 0 层的对象位于画面背景中，第 31 层的对象位于前景中。



同一层的各对象也按层次排列。创建画面时，先插入的对象位于层的最下层。每插入一个对象，即会将该对象向前放一个位置。可在一层中向前移动对象或向后移动对象。

层技术的原理

32 个层中只有一个层当前处于活动状态的。添加到画面的新对象始终被分配给激活的层。活动层的编号会在“层”工具栏和画面的巡视窗口中指示出来。

当打开画面时，将显示画面的所有 32 层。在画面的巡视窗口中，可以隐藏活动层之外的其它所有层。从而可以很明确地编辑活动层中的对象。

可使用画面的巡视窗口定义这些层是否仅显示给工程系统。如果要隐藏画面中的数据，则此功能很有用。

在“布局”任务卡“层”选项板的树形视图中，通过拖放和右键快捷菜单来管理层和对象。

应用实例

例如，可在以下情况中使用层：

- 编辑时隐藏对象标签，
- 进一步组态画面时隐藏某些对象（如报警窗口）。

参见

画面基础知识 (页 953)

在层之间移动对象 (页 1039)

设置激活层 (页 1040)

显示和隐藏层 (页 1041)

重命名层 (页 1042)

8.2.6.2 在层之间移动对象

简介

默认情况下，新对象会插入活动层中。但您随后可以将对象分配给另一个层。

要求

- 打开了含有对象的画面。
- 打开巡视窗口。

将对象移动到另一层

将对象移动到另一层的步骤如下：

1. 在画面上选择对象。
对象属性即显示在巡视窗口中。
2. 单击“其它”。
3. 在“层”下，输入对象要移动到的层。

或者，也可以从“布局”任务卡中选择对象，然后将其拖动到所需的层。

更改对象的顺序

步骤如下：

1. 在画面上选择对象。
对象属性即显示在巡视窗口中。
2. 要前后移动对象，请从快捷菜单选择“顺序”>“向后移动”或“向前移动”命令。

或者，也可以使用工具栏中的



或



图标。

将对象分配给相邻层。

步骤如下：

1. 在画面上选择对象。

对象属性即显示在巡视窗口中。

2. 要将对象分配给紧邻的低层或高层，请从快捷菜单中选择“顺序”>“向后移动”或“向前移动”命令。

或者，也可以使用工具栏中的



或



图标。

结果

对象被分配给所选的层并位于该层的顶部。

参见

有关使用层的基础知识 (页 1038)

8.2.6.3 设置激活层

简介

画面对象将分配到 32 个层中的一个。画面中始终会有一个活动层。添加到画面的新对象始终被分配给激活的层。

激活层的编号在“层”工具栏中指示。活动层在“布局”任务卡的“层”选项板中以



图标来指示。

开始设计时，第 0 层为激活层。必要时可以在组态期间激活另外的层。

要求

- 已打开包含至少一个对象的画面。
- 打开活动画面的巡视窗口。

激活不同的层

激活其它层的步骤如下：

1. 在当前画面的巡视窗口中，单击“层”。
2. 在“设置”下的“活动层”域中输入层号。

其它方法

步骤如下：

1. 在“布局”任务卡中选择“层”选项板。
2. 从右键快捷菜单中选择“活动层”命令。

结果

具有指定编号的层现在激活。

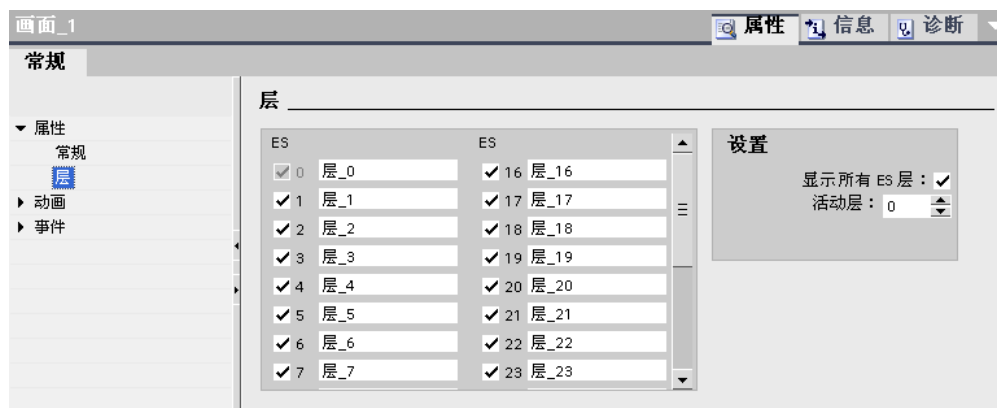
参见

有关使用层的基础知识 (页 1038)

8.2.6.4 显示和隐藏层

简介

可以根据需要显示或隐藏画面的层。指定在工程系统中显示的层。打开画面时，始终都会显示所有层。



要求

- 画面打开。
- 打开“布局”任务卡。

隐藏或显示层

隐藏或显示层的步骤如下：

1. 在“布局”任务卡的“层”选项板中选择要隐藏或显示的层。
2. 单击对应层旁边的一个图标：

— 
隐藏显示的层

— 

显示隐藏的层

注意

活动层将无法隐藏。

使用巡视窗口显示或隐藏层

使用巡视窗口显示或隐藏层的步骤如下：

1. 单击不包含对象的画面区域。

画面属性即显示在巡视窗口中。

2. 在巡视窗口中单击“层”。



3. 在列表中，启用要显示的层或禁用要隐藏的层。

如果为某一层激活“所有 ES 层”，则该层中的对象将显示在工程系统中。

结果

层根据您的设置显示。

参见

有关使用层的基础知识 (页 1038)

8.2.6.5 重命名层

简介

创建画面时，默认情况下 32 个层会连续编号。为了更加明确，您可以根据自己的要求将层重命名。

要求

- 画面打开。

- 打开“布局”任务卡。

在“布局”任务卡中重命名层

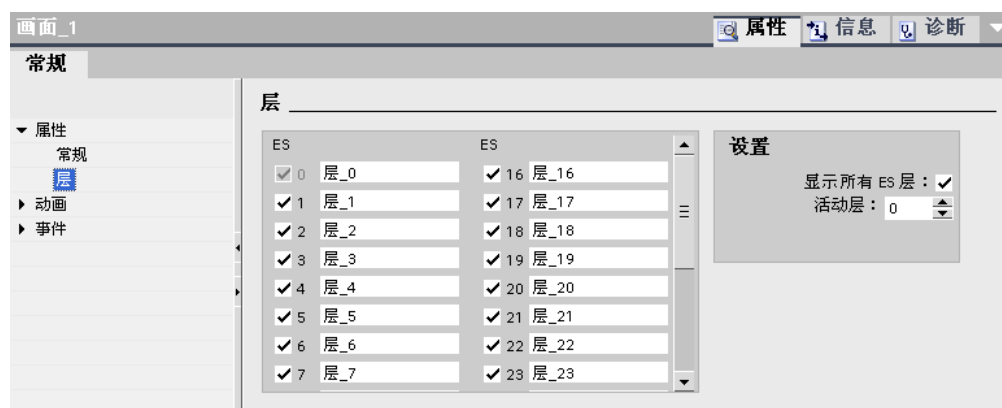
在“布局”任务卡中重命名层的步骤如下：

1. 在“层”选项板中选择要重命名的层。
2. 从快捷菜单中选择“重命名”。
现在可以编辑层的名称。
3. 输入新名称。

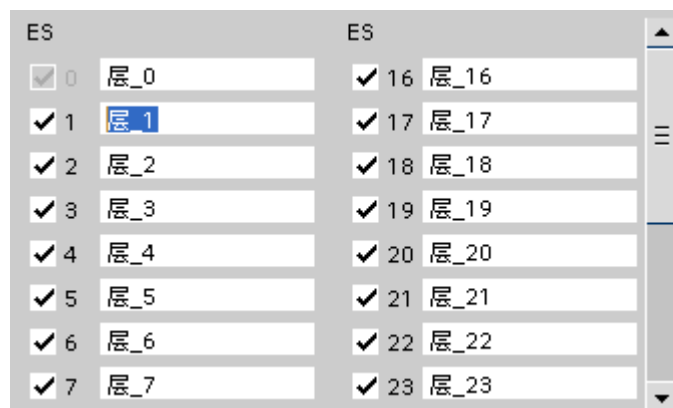
在巡视窗口中重命名层

在巡视窗口中重命名层的步骤如下：

1. 单击不包含对象的画面区域。
画面属性即显示在巡视窗口中。
2. 在巡视窗口中单击“层”。



3. 输入新的层名称。



结果

层将以新名称显示。

参见

有关使用层的基础知识 (页 1038)

8.2.7 使用库

8.2.7.1 关于库的基础知识

简介

将经常使用的所有对象存储在库中。存储在库中的对象只需要组态一次。然后便可以在需要时随时重复使用。库对象扩大了可用画面对象的数量，并通过多次使用现成对象，提高了组态时的效率。

WinCC 软件包附带了包含各种对象（例如“电机”或“阀”对象等）的综合库。尽管如此，还是可以定义自己的库对象。

在“库”任务卡中管理库。如果存储了库对象，此处会提供以下库：

- 项目库
- 全局库



注意

在“工具”任务卡的“元素”选项板中有符号库。

库对象

库中可包含所有 WinCC 对象。实例：

- 完整的 HMI 设备
- 画面
- 包括变量和函数的显示和控制对象
- 包括变量和函数的对象组
- 图形
- 变量
- 报警
- 文本和图形列表
- 日志
- 脚本

要在项目中使用库对象，将对象和所有引用的对象复制到项目中。复制的对象丢失与库的互连。库中的改变不影响任何已复制的库对象。

注意

精简面板

脚本或日志不可用于精简面板。

项目库

每个项目都有一个库。项目库的对象与项目数据一起存储，只可用于在其中创建库的项目。将项目移动到不同的计算机时，包含了在其中创建的项目库。

如果项目库的库对象要用在其它对象中，可将该库对象移动或复制到全局库。

全局库

除了来自项目库的对象外，也可以将来自全局库的对象合并到项目中。全局库独立于项目数据，它存储在扩展名为 *.al10 的单独文件中。

如果要使用另一个项目的全局库，请打开相应的库。

一个项目可以访问多个全局库。一个全局库可以同时用于多个项目中。

如果在一个项目中更改了某个库对象，则在所有打开了该库的项目中，该库都会随之更改。

在全局库中，还能找到 WinCC 软件包随附的库。

类别

要将库对象按主题分类，可以按类别拆分库，也可以创建几个全局库。例如，一个特定的全局库可以包含组态电机控件所需的所有对象。另一个全局库可以包含组态泵控件所需的所有对象。

部件

库元素的组件或元素会按照元素类型显示在“部件”选项板中：

- 如果将单个对象分配给库，则此对象会显示在“部件”选项板中。
- 如果一组几个对象分配给库，则该组（而非组中包含的对象）会显示在“部件”选项板中。
- 当选择多个对象分配给库时，则只有其中一个对象及其名称显示在“全局库”下。而“部件”选项板包含通过此对象插入的所有对象。
- 如果一整个画面及其所有对象分配给库，则画面的名称会显示在“全局库”下和“部件”选项板中。而不会显示单个对象。如果在全局库中更改此元素的名称，则“部件”选项板中的画面名称会保持不变。

参见

画面基础知识 (页 953)
显示库对象 (页 1046)
管理库对象 (页 1048)
将对象存储在库中 (页 1049)
插入库对象 (页 1050)
创建全局库 (页 1051)
保存全局库 (页 1052)
打开全局库 (页 1053)

8.2.7.2 显示库对象

简介

库以文件夹形式显示在相应的选项板中。库中包含的元素显示在文件夹和“元素”选项板中。

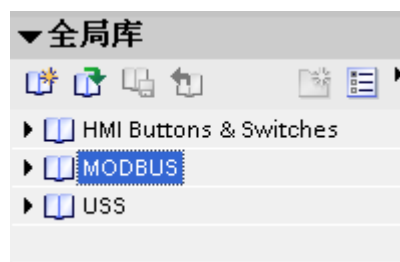
要求

- 库中至少已创建一个库对象。
- 打开“库”任务卡。

显示库元素

显示库中所含元素的步骤如下：

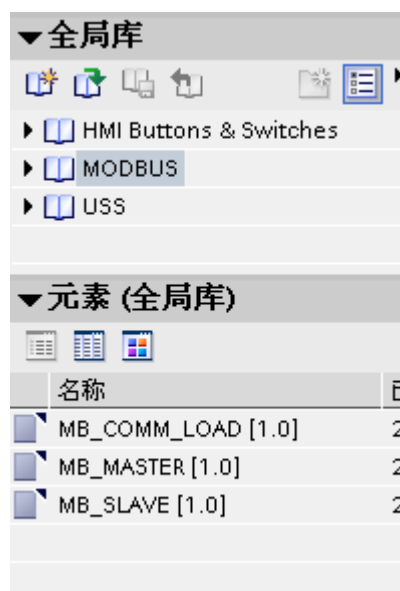
1. 在相应的选项板中选择要显示其库对象的库。



2. 单击



。所包含的库对象即显示在“元素”选项板中。



3. 单击下列图标之一：

图标	描述
	详细模式的元素视图
	列表模式的元素视图
	总览模式的元素视图，带有图标

如果通过多项选择将几个对象分配给库，则只有其中一个对象显示在“元素”选项板中。此元素的单个组件显示在“部件”选项板中。

显示库元素的部件

显示通过多项选择分配给库的对象的步骤如下：

1. 在相应的选项板中选择要查看其中元素组件的库。

2. 单击



。

3. 所包含的库对象即显示在“元素”选项板中。

4. 选择元素。

“部件”选项板即显示组成元素的对象。



结果

库对象会根据组态进行显示。面板的组件也会显示出来。

参见

关于库的基础知识 (页 1044)

8.2.7.3 管理库对象

简介

始终可以在库类别内复制或移动库对象。删除不需要的库对象。

要求

- 已经打开包含多个类别和至少一个对象的库。
- 库对象被显示。

移动库对象

移动库对象的步骤如下：

1. 选择库对象。
2. 将对象拖放至相关的类别中。

库对象被移动。

复制库对象

复制库对象的步骤如下：

1. 选择库对象。
2. 从快捷菜单中选择“复制”。
3. 选择目标类别。
4. 从快捷菜单中选择“粘贴”。

库对象的副本被粘贴到目标类别中。

删除库对象

删除库对象的步骤如下：

1. 选择要删除的库对象。
2. 从快捷菜单中选择“删除”。

库对象被删除。

结果

根据所选的操作，库对象被移动、复制或删除。

参见

关于库的基础知识 (页 1044)

8.2.7.4 将对象存储在库中

简介

画面、变量、图形对象和报警等所有 WinCC 对象都可以存储在库中。可以通过拖放将相应的对象从工作区、项目树或详细视图移动到库中。在已经划分类别的库中，可以将对象直接添加到指定类别。

下面将介绍如何将显示和操作员控件从工作区插入库中。

要求

- 打开“画面”编辑器。
- 要保存到库的对象显示在工作区域中。
- 显示创建的库。

步骤

将对象添加到库的步骤如下：

1. 在“画面”编辑器的工作区域中选择对象。
2. 将对象从工作区域拖放到所需的库。

鼠标指针变成附带对象图标的十字准线。



用同样的方法将其它 WinCC 对象 (如画面) 从项目树拖放到库中。

结果

对象被保存到库中，以在组态的多个实例中进一步使用。



参见

关于库的基础知识 (页 1044)

8.2.7.5 插入库对象

简介

可将库对象拖放到工作区或项目树中。

系统始终为插入的库对象分配一个包含对象类型和连续编号的名称。如果插入的对象已存在，可使用对话框指定是替换现有对象还是以新名称插入。如果不想替换现有对象，请输入新名称。

无法插入 HMI 设备不支持的库对象。

注意

如果从库插入带有互连模板的画面，则模板也会被插入。不使用任何现有匹配模板。

下面，您会看到编辑器工作区域中的一个库对象示例。

要求

- 打开“库”任务卡。
- 已选择包含此对象的库。
- 要在其中插入库对象的编辑器已打开。

步骤

在编辑器中插入库对象的步骤如下：

1. 选择要插入库中的库对象。
2. 将库对象拖放到工作区域中要插入对象的位置。
库文件夹随即插入。
3. 在画面中选择库对象，并使用选择标记调整其大小。

参见

关于库的基础知识 (页 1044)

8.2.7.6 创建全局库

简介

在全局库中，存储要在组态中多次使用的已组态对象。
如果要在多个项目中使用对象，则应创建全局库。

要求

- 已经打开项目。
- 打开“库”任务卡。

步骤

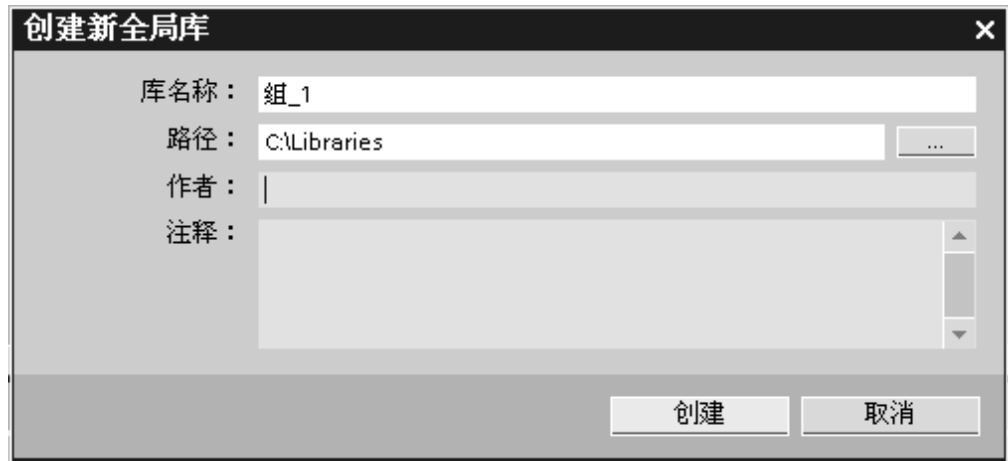
创建全局库的步骤如下：

1. 在“库”任务卡的“全局库”选项板中，单击



。

“创建新全局库”对话框随即显示。



2. 输入新库的名称，并选择新库的存储路径。
3. 单击“保存”。

结果

新库显示在“全局库”选项板中。可以将对象保存到库中。

参见

关于库的基础知识 (页 1044)

8.2.7.7 保存全局库

简介

共享库存储在硬盘驱动器中的一个单独文件中。该文件包含全局库的对象，包括引用的对象。例如对为 I/O 域组态的变量的引用也存储在该库中。

当在未保存的情况下关闭 WinCC 或项目时，WinCC 会提示保存全局库。也可以在组态期间存储全局库，而不必存储整个项目。

要求

- 已经打开包含至少一个库的项目。
- 打开“库”任务卡。

存储库

存储全局库的步骤如下：

1. 选择要保存的全局库。



2. 在“库”任务卡的“全局库”选项板中，单击



。

或者也可以在快捷菜单中选择“保存库”命令。

如果要将全局库保存在另一个文件夹中，在快捷菜单中选择“另存为”命令，选择要保存新库的路径并输入文件名。

结果

全局库以当前文件名或指定的文件名保存。

参见

关于库的基础知识 (页 1044)

8.2.7.8 打开全局库

简介

在 WinCC 中，全局库存储在单独的文件中。通过打开相关库，可以在所有项目中使用全局库。

要求

- 已保存了全局库。
- 已打开要使用全局库的项目。
- 打开“库”任务卡。

步骤

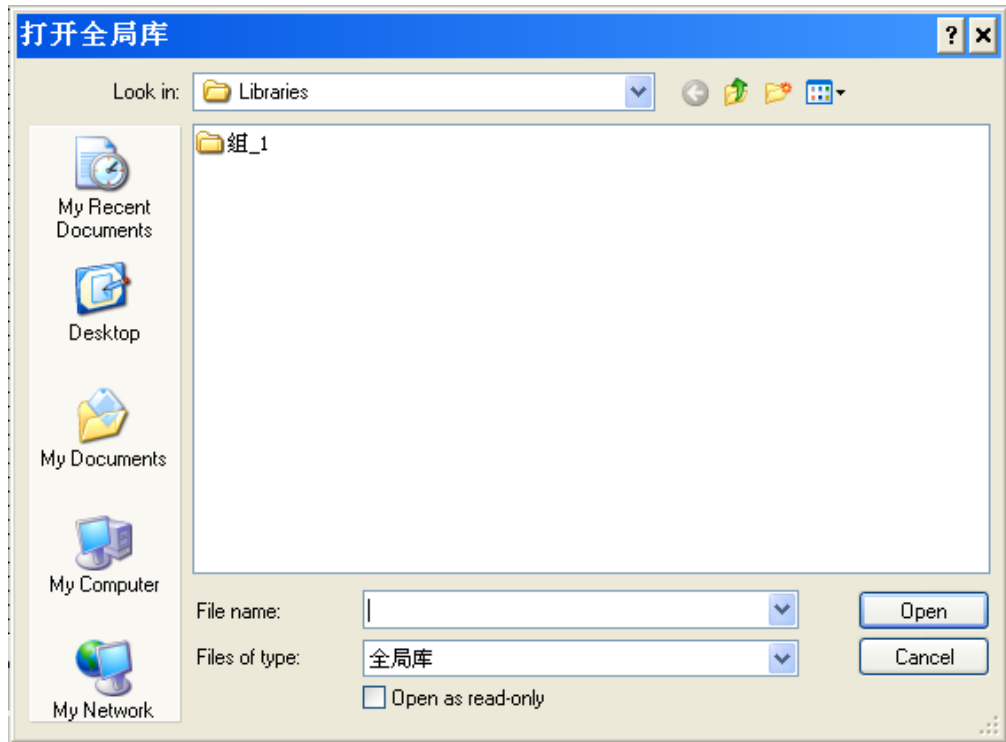
打开全局库的步骤如下：

1. 在“全局库”选项板中，单击



“打开全局库”图标。

“打开全局库”对话框随即显示。



2. 选择用于保存库的路径、库文件夹和文件名。
3. 单击“打开”。
4. 如果要更改库，则禁用“只读打开”。

注意

如果要从几个项目访问全局库，则打开全局库时，全局库必须为只读。为此，在“打开全局库”对话框中启用“只读打开”选项。只要全局库成为只读，就会禁止从其它项目进行访问。

结果

WinCC 在“全局库”选项板中显示打开的全局库。

参见

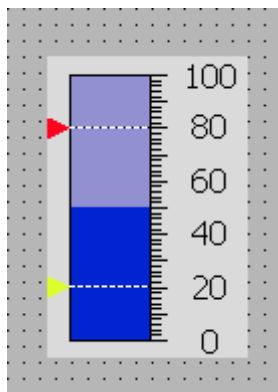
关于库的基础知识 (页 1044)

8.2.8 显示操作对象

8.2.8.1 棒图

应用

过程值通过“棒图”对象以图形方式显示。可以添加标尺来标注棒图显示。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型等设置。具体地说，可以修改下列属性：

- 颜色转变：指定超出限制值后显示颜色的变化。
- 显示限制线/限制标记：将组态的限制显示为线条或标记。
- 定义棒图分段：定义棒图比例尺分级。
- 定义比例尺分级：定义棒图比例尺的细分、刻度标记和间隔。

颜色转变

在“属性”窗口“属性”组中的“外观”下定义颜色变化的方式。

颜色转变	描述
“按段”	颜色逐段变化。
“整个棒图”	整个棒图的颜色发生变化。

显示限制线和限制标记

通过“线条”和“标记”属性将棒图中已组态的限制在运行系统中显示为线条或标记。步骤如下：

1. 在巡视窗口中单击“外观”。
2. 在“限制”区域中激活“线条”和“标记”选项。

定义棒图分段

使用“细分数量”属性定义段数，通过标尺上的主分级将棒图分割成该段数。

使用“间隔”属性分割主分级之间的距离。该值是两个邻近主分级之间值的差。步骤如下：

1. 在巡视窗口中单击“标尺”。
2. 激活“显示标尺”。
3. 在“设置”区域的“细分数量”下选择相应值。
4. 在“设置”区域的“标记标签”下选择相应值。
5. 在“大间距”区域的“间隔”下选择相应值。

参见

画面基础知识 (页 953)

用户视图 (页 1056)

日期/时间域 (页 1057)

I/O 域 (页 1058)

椭圆 (页 1060)

图形视图 (页 1061)

图形 I/O 域 (页 1062)

圆 (页 1063)

趋势视图 (页 1064)

直线 (页 1066)

报警视图 (页 1067)

报警窗口 (页 1068)

报警指示器 (页 1071)

矩形 (页 1071)

配方视图 (页 1072)

开关 (页 1073)

按钮 (页 1074)

符号 I/O 域 (页 1075)

文本域 (页 1077)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.2 用户视图

应用

“用户视图”对象用于设置和管理用户和授权。

Administrator	Administrator group
Meister	Group_2
Operator_1	Operator
PLC User	Unauthorized
<New user>	

注意

在组中不要使用简单用户视图。

布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 行数：指定可见条目的最大数量。

行数

在巡视窗口中指定运行系统中显示的用户视图的行数。只有在属性“自动调整大小”处于激活状态时，行数的设置才会生效。

1. 在巡视窗口中单击“常规”组。
2. 在“设置”下的“行数”域中输入整数值。
3. 在巡视窗口中，激活“属性”>“布局”>“大小”。
4. 在“尺寸”区域中，激活“自动调整大小”。

参见

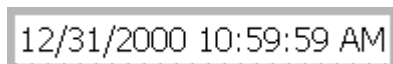
棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.3 日期/时间域

应用

“日期/时间域”对象显示了系统时间和系统日期。“日期/时间域”的外观取决于在 HMI 设备中设置的语言。



12/31/2000 10:59:59 AM

布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 显示系统时间：指定显示的系统时间。
- 使用变量：指定显示连接变量的时间。
- 日期/时间长格式：此设置定义日期和时间的显示格式。

显示系统时间

在巡视窗口中指定 HMI 设备上“日期/时间域”中显示的时间。

1. 在巡视窗口中单击“常规”组。
2. 激活“格式”区域中的“系统时间”选项。

使用变量

互连变量的时间显示在日期/时间域中。

1. 在巡视窗口中单击“常规”组。
2. 在“格式”区域中，选择“DateTime”数据类型的变量，如内部变量。有关可用于与其它 PLC 连接的数据类型的相关信息可参见相应的通信驱动程序文档。

日期/时间长格式

在巡视窗口“常规”组下的“格式”区域中指定日期和时间的格式。

选项	描述
“允许”	完整地显示日期和时间，如“2000 年 12 月 31 日，星期日，上午 10:59:59”
“禁止”	以简短形式显示日期和时间，如“12/31/2000 10:59:59 AM”

参见

棒图 (页 1055)

画面基础知识 (页 953)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.4 I/O 域

应用

“I/O 域”对象用于输入和显示过程值。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 模式：指定在运行系统中对象的响应。
- 显示格式：指定 I/O 域中输入和输出值的显示格式。
- 隐藏输入：指定在输入过程中是正常显示输入值还是加密输入值。

注意 报表

只有“输出”模式可用于报表。

“隐藏输入”选项不可用于报表。

模式

在巡视窗口的“常规”组的“I/O 域类型”区域中指定 I/O 域的响应。

模式	描述
“输入”	在运行系统中，只能在 I/O 域中输入值。
“输入/输出”	在运行系统中，可以在 I/O 域中输入和输出值。
“输出”	I/O 域仅用于输出值。

布局

输入和输出值的显示格式在巡视窗口“常规”组下的“格式”区域中指定。

布局	
“二进制”	以二进制形式输入和输出数值
“十进制”	以十进制形式输入和输出值
“十六进制”	以十六进制形式输入和输出值
“日期”	输入和输出日期信息。格式依赖于在 HMI 设备上的语言设置。
“日期/时间”	输入和输出日期和时间信息。格式依赖于在 HMI 设备上的语言设置。
“时间”	输入和输出时间。格式依赖于在 HMI 设备上的语言设置。
“字符串”	输入和输出字符串。
“时间”	输入和输出时间。

隐藏输入

在运行系统中，可以正常或加密（例如密码的隐藏输入）显示输入内容。在隐藏输入过程中，系统使用“*”显示每个字符。输入值的数据格式不能识别。

1. 在巡视窗口中单击“特征”。
2. 在“域”区域中，激活“隐藏输入”选项。

对于 OP 73、OP 73micro、OP 77A、TP 177A、TP 177micro 避免输出域重叠

如果在画面中将几个 I/O 域组态为具有透明背景的输出域，则这些 I/O 域可能会重叠。一个域的透明部分会遮盖另一个域的数字。这会导致运行系统出现显示问题。为避免此类重叠，应在“属性”>“布局”下的对象属性中将 I/O 域的边距设置为零。激活“按内容调整对象”复选框。

参见

棒图 (页 1055)

画面基础知识 (页 953)

8.2.8.5 椭圆

应用

“椭圆”是可用颜色或图案填充的闭合对象。



布局

在巡视窗口中，可自定义设置对象位置、几何形状、样式、边框和颜色。具体地说，可以修改下列属性：

- 半径 X：指定椭圆对象的水平半径。
- 半径 Y：指定椭圆对象的垂直半径。

半径 X：

在巡视窗口中指定“椭圆”对象的水平半径。以像素为单位输入该值。

1. 在“属性”下单击“布局”。
2. 在“水平”下输入介于 0 和 2500 之间的值。

半径 Y

在巡视窗口中指定“椭圆”对象的垂直半径。以像素为单位输入该值。

1. 在“属性”下单击“布局”。
2. 在“垂直”下输入介于 0 和 2500 之间的值。

参见

棒图 (页 1055)

8.2.8.6 图形视图

应用

“图形视图”对象用于显示图形。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 图形：指定在对象中显示的图形文件。
- 伸展图形：指定带图形对象的自动尺寸延伸。
- 透明色：指定图形是否采用透明色。

插入图形

“图形视图”对象中可使用下列图形格式：*.bmp、*.tif、*.png、*.ico、*.emf、*.wmf、*.gif、*.jpg 或 *.jpeg。也可在图形视图中使用作为 OLE 对象的图形。

1. 在巡视窗口中单击“常规”。
2. 在“图形”的下拉列表中选择要添加的图形。
右窗格中将显示图形预览。
3. 单击“分配”在图形视图中插入图形。单击“删除”从图形显示中删除图形。

伸展图形

在巡视窗口中指定是否将图形视图中显示的图形伸展到运行系统中图形视图的大小。

1. 在巡视窗口中单击“布局”。
2. 从“适合大小”区域中选择下列选项之一：
 - 按内容调整对象大小
 - 按对象大小调整内容

透明色

该属性定义了要显示的图形是否采用透明色。仅当该对象被分配了 3D 边框或没有分配边框时才可以使透明色。

1. 在巡视窗口中单击“布局”。
2. 在“颜色”区域中激活“透明色”。

注意

在 WinCC 画面中使用位图需要在面板 HMI 设备上进行用于高性能可视化的“透明色”设置。在相关显示对象的属性中禁用“透明色”设置可提高可视化性能。此限制尤其适用于将位图用作背景图像的情况。

注意

精简面板

“透明色”属性不能用于精简面板。

参见

画面基础知识 (页 953)
可用于特定 HMI 设备的画面 (页 954)
棒图 (页 1055)

8.2.8.7 图形 I/O 域

应用

“图形 I/O 域”对象可用于组态一份实现图形文件的显示和选择的列表。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 模式：指定在运行系统中对象的响应。
- 滚动条类型：指定滚动条的图形布局。

注意

精简面板

滚动条不可用于精简面板。

注意 报表

只有“输出”模式可用于报表。

滚动条不可用于报表。

模式

可在巡视窗口的“常规”>“类型”>“模式”下指定“图形 I/O 域”对象的响应。

模式	描述
“输入”	“图形 I/O 域”对象只用于选择图形。
“输入/输出”	“图形 I/O 域”对象用于选择和显示图形。
“输出”	“图形 I/O 域”对象只用于显示图形。
“两种状态”	“图形 I/O 域”对象仅用于显示图形，并且最多只能具有两种状态。

滚动条类型

可在巡视窗口的“外观”>“滚动条”>“类型”下指定显示窗口中对图形表示的响应。

类型	描述
“永久”	滚动条始终可见。
“无滚动条”	滚动条不可见。
“单击后可见”	滚动条在鼠标单击后可见。

参见

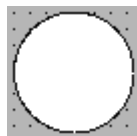
棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.8 圆

应用

“圆”是封闭对象，可以用颜色或图案填充。



布局

在巡视窗口中，可自定义设置对象位置、几何形状、样式、边框和颜色。具体地说，可以修改下列属性：

- 半径：指定圆的尺寸。

圆角

在巡视窗口中指定“圆”对象的半径。以像素为单位输入该值。

1. 单击“布局”。
2. 在“半径”区域中输入介于 0 到 2500 之间的数值。

参见

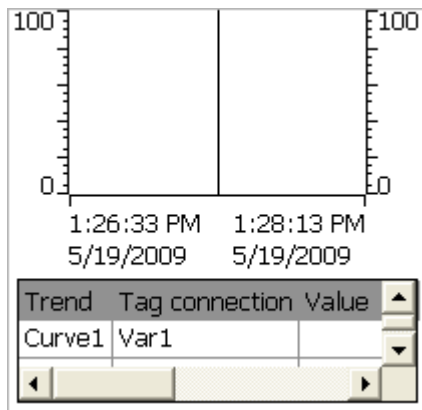
棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.9 趋势视图

应用

趋势视图指当前过程或日志的变量值以趋势的形式表达的图形。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 显示数值表、标尺和网格：指定除了坐标系外，是否还显示数值表、标尺或网格以提高可读性。
- 工具栏：定义控制元素的显示。

显示数值表、标尺和网格

为了提高可读性，可以在运行系统中显示数值表、标尺和网格。

1. 激活“布局”下的“显示标尺”选项。
2. 激活“表格”下的“显示表格”选项。

3. 激活“表格”下的“显示网格”选项。

工具栏

在巡视窗口的“工具栏”下指定控制元素的布局。可以指定不显示控制元素、将控制元素显示为工具栏或将控制元素显示为按钮。

注意

精简面板

控制元素不可用于精简面板。

按钮	工具栏	简述	描述
		“转到开始位置”	向后翻页到趋势记录的开始处。显示开始趋势记录的起始值。
		“放大”	放大所显示的时间区域。
		“缩小”	缩小所显示的时间区域。
		“标尺向后”	将标尺向后移动。
		“标尺向前”	将标尺向前移动。
		“向后”	回滚一个显示宽度。
		“向前”	向前滚动一个显示宽度。
		“标尺”	显示或隐藏标尺。标尺显示相应位置的 X,Y 坐标值。
		“启动/停止”	停止记录趋势或继续记录趋势。

组态行为

显示列标题

趋势视图中表格的布局取决于控制面板中的视图设置。根据设置的不同，表格标题可能会被截断。可在控制面板的“显示”>“布局”下找到此设置。为了正确显示列标题，在“窗口和按钮”中将显示方式设置为“Windows 经典”样式。

这种情况仅在组态期间发生。列标题在运行系统中可正确显示。

一致性测试

如果在涉及曲线显示的一致性检查期间输出窗口中显示警告或错误，单击快捷菜单上的“跳转至出错处/变量”并不能总是跳转到准确的错误位置。在某些情况下，仅曲线显示错误的原因。

添加、组态和删除趋势

在趋势视图巡视窗口的“属性”>“趋势”下管理趋势视图的趋势。可在不同趋势视图之间复制趋势。

参见

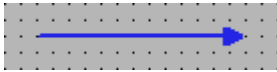
棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.10 直线

应用

“直线”对象是一个开放的对象。直线长度和斜率由包围对象的矩形的高度和宽度定义。



布局

在巡视窗口中，可自定义对象位置、形状、样式和颜色的设置。具体地说，可以修改下列属性：

- 线样式
- 线始端和线末端

线样式

在巡视窗口中，“属性”组的“外观”下定义直线的显示方式。例如，如果选择“实线”，则直线将不间断地显示出来。

注意

可用于特定 HMI 设备

可用的线样式取决于 HMI 设备。

线始端和线末端

在巡视窗口中，“属性”组的“外观”下的“线端点”中，定义线的起点和终点。

例如，使用箭头作为起点和终点。可用的起点和终点与设备有关。

参见

棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.11 报警视图

应用

报警将在报警视图或 HMI 设备的报警窗口中显示。

以下画面包含一个简单报警视图：



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。

注意

可选择的字体取决于在运行系统设置中组态的“语言和字体”。

具体地说，可以修改下列属性：




- 控制元素：定义报警显示的操作员控件。
- 报警类别：此设置定义了报警视图中将显示哪些报警类别。
- 列：指定运行系统中显示的列。

注意

如果具有不同的报警类别输出，则在运行系统中报警先按报警类别排序，然后再按报警发生时间排序。

控制元素

在巡视窗口的“显示”>“设置”下，指定可用于在运行系统中控制报警显示的控制元素。下表显示了报警视图中的控制元素以及它们的作用：

按钮		功能
“信息文本”		显示报警的信息文本。
“确认”		确认报警。
"Loop-In-Alarm"		切换到包含已发生错误的信息的画面。

选择报警类别

1. 在巡视窗口中单击“属性”。
2. 在“报警类别”下，激活要在运行系统的报警视图中显示的报警类别。

定义列

在巡视窗口中，定义要在运行系统的报警视图中显示的列。

1. 在巡视窗口中单击“属性”>“列”。
2. 在“列”下激活要在运行系统中显示的列。

组态行为

显示列标题

报警视图的布局取决于控制面板中的视图设置。根据设置的不同，列标题可能会被截断。可在控制面板的“显示”>“布局”选项卡下找到此设置。为了正确显示列标题，在“窗口和按钮”中将显示方式设置为“Windows 经典”样式。

这种情况仅在组态期间发生。列标题在运行系统中可正确显示。

注意

在工程系统中可以动态控制对象的可见性，例如，在巡视窗口的“动画”组中。在运行系统中，“简单报警视图”不支持动画。如果已组态某个动画，并希望执行某项操作（例如执行项目一致性检查），则系统会在“输出”窗口中显示一条错误报警。

参见

棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.12 报警窗口

应用

报警将在报警视图或 HMI 设备的报警窗口中显示。报警窗口的布局和操作与报警视图的类似。报警窗口中的下列特性与报警视图相同：

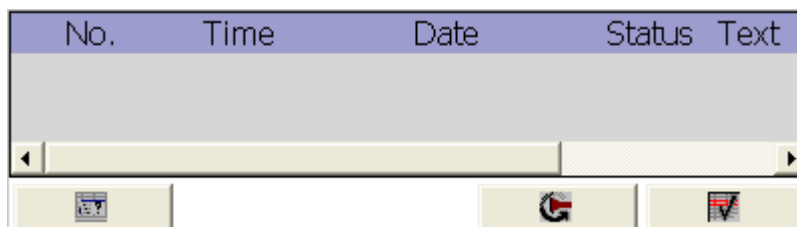
- 简单报警窗口
- 高级报警窗口
- 报警行

注意
精简面板

只有简单报警窗口可用于精简面板。

报警窗口在“全局画面”编辑器中组态。

报警窗口独立于过程画面。通过组态，报警窗口可以设置成一接收到新的、未确认的报警就自动打开。如果可行，可将报警窗口组态为只有在所有报警都经确认之后才关闭。下图显示了一个高级报警窗口：



注意

在工程系统中，可以实现动态化操作，例如在巡视窗口的“动画”组中动态化对象的可见性。在运行系统中，“简单报警窗口”对象不支持动画。如果已组态某个动画，并希望执行某项操作（例如执行项目一致性检查），则系统会在“输出”窗口中显示一条错误报警。

注意

无法编组“报警窗口”对象。

布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。可以按照与组态报警视图相同的方式组态报警窗口。另外，还可修改下列属性：




- 固定报警窗口：指定在画面更改后报警窗口仍保留原焦点。
- 窗口：定义在运行系统中报警窗口的操作员输入和响应。

注意

如果具有不同的报警类别输出，则在运行系统中报警先按报警类别排序，然后再按报警发生时间排序。

控制元素

在巡视窗口的“显示”>“设置”下，指定可用于在运行系统中控制报警显示的控制元素。下表显示了报警窗口中的控制元素以及它们的作用：

按钮		功能
“信息文本”		显示报警的信息文本。
“确认”		确认报警。
"Loop-In-Alarm"		切换到包含已发生错误的信息的画面。

运行系统中的访问保护

在报警视图属性的“属性”>“安全”组中组态访问保护。如果登录用户具有所需的权限，则该用户可以使用报警视图中的操作员控件确认和编辑报警。如果登录用户没有所需的权限或者没有用户登录，则单击“确认”或“编辑”按钮或者双击报警行将打开登录对话框。

注意

精简面板

访问控制不可用于精简面板。

激活报警窗口的焦点

要使报警窗口在画面更改后不丢失焦点，请选择下列选项：

1. 在巡视窗口的“属性”组中单击“模式”。
2. 启用“标签”。

窗口

在巡视窗口中，“窗口”区域的“模式”类别中定义报警窗口的响应。下表显示了可能的属性：

选项	功能
自动显示	例如，发生系统报警时自动显示报警窗口。
可关闭	设置时间结束后窗口再次关闭。可在报警设置中定义显示持续时间。
模式的	报警窗口与确认相链接，如：必须确认报警。如果焦点为模式报警窗口，则无法使用后面画面中的按钮。但为功能键组态的功能可执行。
可调整大小	可更改运行系统中报警窗口的大小。

参见

棒图 (页 1055)
可用于特定 HMI 设备的画面 (页 954)

8.2.8.13 报警指示器

应用

报警指示器是根据组态显示当前错误或显示需要进行确认的错误的图形符号。报警指示器在“全局画面”编辑器中组态。下图包含一个报警指示器：



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 报警类别：确定报警指示器显示之处的报警类别。
- 运行系统中的操作员控件：定义运行系统中导致报警窗口打开的操作员操作。

报警类别

在巡视窗口的“常规”>“报警类别”下定义使用报警指示器显示哪些报警类别。报警类别包括警告或错误。

定义运行系统中的操作员控件

在巡视窗口的“事件”下定义报警窗口对于哪些事件可见或隐藏。

1. 选择画面中的报警指示器。
2. 在巡视窗口中，单击“事件”组中的“单击”或“闪烁时单击”。
3. “函数列表”打开。单击函数列表的第一行。将打开项目中可用的系统函数和脚本列表。
4. 选择“报警”下的“ActivateScreen”系统函数。在“对象名称”下，从选择列表中选择报警窗口的名称。在“布局”下，定义报警窗口应该是显示、隐藏还是在两种状态之间切换。

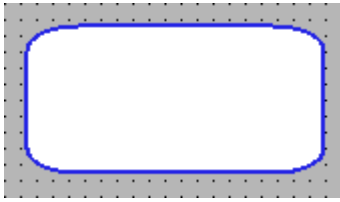
参见

棒图 (页 1055)
可用于特定 HMI 设备的画面 (页 954)

8.2.8.14 矩形

应用

“矩形”是可用颜色填充的闭合对象。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 转角 X 半径和转角 Y 半径：指定矩形顶点与圆角起点之间的水平距离和垂直距离。

转角 X 半径和转角 Y 半径

可以根据需要将“矩形”对象的转角定义为圆角。如果将属性“转角 X 半径”和“转角 Y 半径”的值设置为 100%，则矩形会显示为椭圆。只要这两个属性之一的值为 0%，就会显示不带圆角的正常矩形。

1. 单击“属性”组中的“布局”。
2. 在“转角半径”区域中为“转角 X 半径”输入一个数值。输入值是矩形一半宽度的百分比。
3. 在“转角半径”区域中为“转角 Y 半径”输入一个数值。输入值是矩形一半高度的百分比。

参见

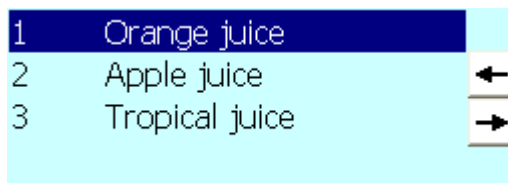
棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.15 配方视图

应用

“配方视图”对象用于显示和修改配方。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 控制元素：指定配方视图的菜单命令。

控制元素

在巡视窗口的“属性”>“按钮”下，组态在运行系统中用于操作配方视图的菜单命令。

菜单命令	描述
“信息文本”	调用为所选配方组态的信息文本。
“新建记录”	在配方中创建新的配方记录。
“删除记录”	删除选定记录。
“保存”	以当前名称保存修改过的记录。
“另存为”	以新名称保存已修改的记录。
“写入 PLC”	将当前值发送给 PLC。
“从 PLC 中读取”	从 PLC 中读取当前值。

组态行为

注意

例如，在工程系统中，可以在“属性”窗口的“动画”组中动态控制对象的可见性。在运行系统中，“简单配方视图”不支持动画。如果已组态某个动画，并希望执行某项操作（例如检查项目的一致性），则系统会在“输出”窗口中显示一条错误报警。

参见

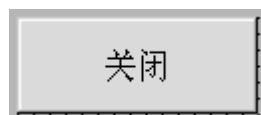
棒图 (页 1055)
可用于特定 HMI 设备的画面 (页 954)

8.2.8.16 开关

应用

“开关”对象用于组态开关，以便在运行期间在两种预定义的状态之间进行切换。可通过标签或图形将“开关”对象的当前状态做可视化处理。

下图显示了一个“开关”类型的开关。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以自定义下列属性：

- 类型：定义对象的图形表示。

类型

可在巡视窗口“常规”组的“设置”区域中指定开关。

类型	描述
“开关”	“开关”的两种状态均按开关的形式显示。开关的位置指示当前状态。在运行期间通过滑动开关来改变状态。 对于这种类型，可在“开关方向”中指定开关的运动方向。
“带有文本的开关”	该开关显示为一个按钮。其当前状态通过标签显示。在运行期间，单击该按钮即可启动开关。
“带有图形的开关”	该开关显示为一个按钮。其当前状态通过图形显示。在运行期间，单击该按钮即可启动开关。

注意 精简面板

“开关”类型不可用于精简面板。

参见

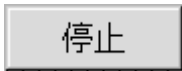
棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.17 按钮

应用

“按钮”对象允许用户组态一个对象，借助该对象操作员可在运行系统中执行任何可组态的功能。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 类型：定义对象的图形表示。
- 文本/图形：定义图形视图是静态的还是动态的。
- 定义热键：定义操作员用来启动该按钮的按键或快捷键。

注意

只能为带有按键的 HMI 设备定义热键。

类型

在巡视窗口中“常规”下的“类型”区域中定义按钮显示。

类型	描述
“不可见”	该按钮在运行系统中不可见。
“文本”	该按钮的当前状态通过标签显示。
“图形”	该按钮的当前状态通过图形显示。

文本/图形

“类型”属性设置用于定义显示是静态的还是动态的。在巡视窗口“常规”组的“文本”或“图形”区域中定义显示。

类型	选项	描述
“文本”	“文本”	“OFF 状态文本”用于指定当状态为“OFF”时将在按钮中显示的文本。如果启用“ON 状态文本”，则可为“ON”状态输入文本。
	“文本列表”	按钮的文本取决于状态。根据状态显示文本列表中的条目。
“图形”	“图形”	“OFF 状态图形”用于指定当状态为“OFF”时将在按钮中显示的图形。如果启用“ON 状态图形”，则可为“ON”状态输入图形。
	“图形列表”	按钮的图形取决于状态。根据状态显示图形列表中的条目。

定义热键

在巡视窗口中，定义操作员可用于在运行系统中控制按钮的按键或组合键。

1. 在“属性”窗口中单击“常规”组。
2. 在“热键”区域的选择列表中选择按键或组合键。

参见

棒图 (页 1055)
可用于特定 HMI 设备的画面 (页 954)

8.2.8.18 符号 I/O 域

应用

可使用“符号 I/O 域”对象来组态运行系统中用于文本输入和输出的选择列表。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 模式：指定在运行系统中对象的响应。
- 文本列表：指定链接到对象的文本列表。
- 选择列表的按钮：指定对象具有可打开选择列表的按钮。

注意 报表

只有“输出”模式可用于报表。

“选择列表的按钮”选项不可用于报表。

模式

在巡视窗口“常规”组的“类型”区域中指定符号 I/O 域的响应。

模式	描述
“输出”	符号 I/O 域用于输出数值。
“输入”	符号 I/O 域用于输入数值。
“输入/输出”	符号 I/O 域用于数值的输入和输出。
“两种状态”	符号 I/O 域仅用于输出数值，且最多可具有两种状态。该域在两个预定义的文本之间切换。例如，可用于显示一个阀的两种状态：关闭或打开。

注意

符号 I/O 域所能具有的特性取决于运行系统。

文本列表

在巡视窗口中，指定与符号 I/O 域相链接的文本列表。

1. 在巡视窗口中激活“常规”。
2. 在“文本列表”的“视图”区域打开选择列表。选择文本列表。

选择列表的按钮

“选择列表的按钮”属性用于显示一个可打开选择列表的按钮。

1. 在巡视窗口中激活“布局”。

2. 在“设置”区域中激活“选择列表的按钮”选项。

注意
精简面板

“选择列表的按钮”选项不可用于精简面板。

参见

棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.8.19 文本域

应用

“文本域”是可以颜色填充的封闭对象。



布局

在巡视窗口中，自定义对象的位置、形状、样式、颜色和字体类型。具体地说，可以修改下列属性：

- 文本：指定文本域的文本。
- 文本域的大小：指定是否将对象尺寸调整到最长列表条目所需要的距离。

文本

在巡视窗口中指定文本域的文本。

1. 在巡视窗口中单击“常规”组。
2. 在“文本”区域中输入任意长度的文本。对于跨越多行的文本，可通过按 <Enter>、使用组合键 <Shift + Enter> 或组合键 <Ctrl + Enter> 设置分行。

文本域的大小

在巡视窗口中，可定义是否根据最长列表条目所需要的空间来调整对象大小。

1. 在巡视窗口中，单击“属性”组中的“布局”。
2. 在“尺寸”区域中，激活“自动调整大小”选项。
3. 另外，可以手动设置大小。在“大小”区域中，禁用“自动调整大小”选项。按下鼠标按钮并将文本域拖动至所需要的大小。

参见

棒图 (页 1055)

可用于特定 HMI 设备的画面 (页 954)

8.2.9 为精简面板组态画面浏览

8.2.9.1 画面浏览的基础知识

画面切换浏览的类型

对于包含多个子过程的生产过程，将组态多个画面。在运行系统中，以下选项可使操作员从一个画面切换到下一个画面：

- 为画面切换分配按钮
- 使用局部功能键组态画面切换

步骤

在创建画面切换之前，请先定义工厂结构并从中得出要组态的画面切换。如果您尚未在项目向导中定义起始画面，请在运行系统设置中创建起始画面。

参见

画面基础知识 (页 953)

组态画面切换 (页 1078)

8.2.9.2 组态画面切换

简介

运行期间，要在 HMI 设备的画面之间进行切换，需使用拖放功能在画面上组态按钮、功能键或同时组态二者。

可按以下方法组态画面切换：

- 要在画面中创建一个可调用另一个画面的按钮，请将第二个画面从项目树拖放到打开的画面。
- 要在画面中组态一个可调用另一个画面的功能键，请将第二个画面从项目树拖放到此功能键。

注意

如果在画面的巡视窗口中将动画的“可见性”设置为“隐藏”，则无法在运行系统中调用此画面。

要求

- 必须具备至少有两个画面的 HMI 设备。
- 要在其中调用画面切换的画面必须已经打开。

将画面切换分配给按钮

执行下列步骤，将按钮分配给画面切换：

1. 将要按钮打开的画面从项目树拖放到已打开的画面。

系统将添加一个按钮并标以画面名称。

2. 在巡视窗口中，单击“事件”类别中的“单击”。

“ActivateScreen”函数将显示在“函数列表”组中。该函数有两个参数：

- “画面名称”

“画面名称”参数包含在运行系统中切换画面时所要调用的画面名称。

- “对象号”。

此可选的“对象号”参数表示对象的 Tab 顺序号，通过它设置画面切换后焦点的位置。

3. 如果需要，还可以在“对象号”属性中定义对象的 Tab 顺序号。画面改变后，会在该对象上设置一个焦点。也可以指定一个包含此对象号的变量。

将画面切换分配给功能键

步骤如下：

1. 将要用功能键打开的画面从项目树拖放到该功能键上。

组态的功能键显示一个黄色的三角形。

2. 在巡视窗口的“事件”类别中，单击“按下”。

“ActivateScreen”函数将显示在“函数列表”组中。该函数有两个参数：

- “画面名称”

“画面名称”参数包含在运行系统中切换画面时所要调用的画面名称。

- “对象号”。

此可选的“对象号”参数表示对象的 Tab 顺序号，通过它设置画面切换后焦点的位置。

3. 如果需要，还可以在“对象号”属性中定义对象的 Tab 顺序号。画面改变后，会在该对象上设置一个焦点。也可以指定一个包含此对象号的变量。

结果

在运行系统中，操作员可以使用按钮或功能键切换到指定的画面。一旦指定了对象号，则具有此对象号的对象将在画面改变后获得焦点。

参见

画面浏览的基础知识 (页 1078)

8.3 使用变量

8.3.1 基础知识

8.3.1.1 有关变量的基础知识

简介

在项目中使用变量来传送数据。变量具有一个在项目中使用的地址和符号名称。地址用于与自动化系统进行通信。

WinCC 使用两种类型的变量：

- 过程变量
- 内部变量

通过提供变量组，WinCC 为变量的管理提供了方便。

原理

在 WinCC 中，由过程提供值的变量称为过程变量或外部变量。对于过程变量，变量的属性用于定义 WinCC 用来与自动化系统通信的连接以及如何交换数据。

不是由过程提供值的变量（内部变量）不会连接到自动化系统。这在变量的“连接”属性中由“内部变量”条目标识。

为了更加清晰，也可以将变量编译到组中。创建变量组时，在“项目”窗口中将显示一个项目树。可像使用 Windows 资源管理器那样来浏览该项目树。

8.3.1.2 内部变量

简介

内部变量不具有与 PLC 的连接。

原理

内部变量存储在 HMI 设备的内存中。因此，只有这台 HMI 设备能够对内部变量进行读写访问。例如，可以创建内部变量来执行本地计算。

可以将 HMI 数据类型用于内部变量。

可使用以下 HMI 数据类型：

HMI 数据类型	数据格式
SByte	有符号 8 位数

HMI 数据类型	数据格式
UByte	无符号 8 位数
Short	有符号 16 位数
UShort	无符号 16 位数
Long	有符号 32 位数
ULong	无符号 32 位数
Float	32 位 IEEE 754 浮点数
Double	64 位 IEEE 754 浮点数
Bool	二进制变量
WString	文本变量，16 位字符集
DateTime	日期和时间的格式为： “DD.MM.YYYY hh:mm:ss”

8.3.1.3 外部变量

简介

外部变量使得自动化系统的组件之间（例如 HMI 设备和 PLC 之间）能够进行通信（数据交换）。

原理

外部变量是 PLC 中所定义的存储单元的映像。无论是 HMI 设备还是 PLC，都可对该存储位置进行读写访问。

可使用符号寻址访问 PLC。由于该设置允许您通过符号表访问 PLC，因此简化了组态。更多信息，请参见“符号访问的基础知识 (页 1082)”部分。

由于外部变量是在 PLC 中定义的存储位置的映像，因而它能采用的数据类型取决于与 HMI 设备相连的 PLC。

数据类型

用于使用外部变量的各 HMI 数据类型适用于 WinCC。此外，也可以使用针对所连接的 PLC 专门设计的其它数据类型。即使具有相同的名称，各数据类型也可能不同。例如，用于通过 SIMATIC S7 协议连接的“Byte”数据类型的版本是“无符号”的。而 WinCC 中 HMI 数据类型“SByte”的格式是“有符号”的。将 HMI 数据的数据类型和已连接的 PLC 的数据类型转换为相互匹配。

注意

在 HMI 运行系统组态期间，除外部变量之外，还存在用于在 HMI 设备和 PLC 之间进行通信的区域指针。可以在“连接”编辑器中设置和启用区域指针。

更新系统函数中的变量值

系统函数始终从运行系统存储器中获取外部变量的值。请求变量值后，将从 PLC 中读取当前值并写入运行系统存储器。然后，会将变量值更新为设置周期时间。系统函数首先访问在上一个扫描周期检查点从 PLC 读取的变量值。

8.3.1.4 符号访问的基础知识**简介**

利用符号访问，可输入一个符号名称作为 PLC 中的目的地，代替输入直接地址。编写 PLC 程序时定义符号名称。

- 在 PLC 变量表中定义全局 PLC 变量的符号名称。
- 在相应的块中，为逻辑块或全局数据块中的局部变量指定符号名称。

更多信息，请参见“符号编程 (页 333)”。

这样，将 HMI 变量互连到 PLC 时，只需选择 PLC 变量表或数据块中的符号名称即可。

当 PLC 程序中的寻址发生更改时，仅仅是符号表中分配给符号名称的地址发生变化。而符号名称以及与 HMI 设备的互连则不会有改变。不必在组态 HMI 设备时匹配此变化。

要求

要实现 HMI 变量的符号访问，必须满足以下要求：

- PLC 变量必须在 PLC 程序中，或者局部变量必须在程序块中。
- 必须为 PLC 中的局部变量编译程序块，以便其在 HMI 设备上可用。

符号访问的工作原理

首先，为 WinCC 中的 PLC 控件编写控制器程序。在编写程序时使用符号声明。

完成后，编译 PLC 程序。编译完成后，符号名称即可用于 HMI 设备。

如果以后要更改 PLC 程序，则需要重新执行编译以使更改可用于 HMI 设备。

然后，为 HMI 设备创建组态。将 HMI 变量与 PLC 变量表或 PLC 程序块中的符号直接互连。

优势

通过在 PLC 中使用符号声明，采用的 CPU 中的数据得到有效地组织。该设置提高了 PLC 的性能。

符号访问还提高了 HMI 设备和 PLC 之间通信的性能。

HMI 设备的组态不依赖于 PLC 端的变化，因为 PLC 中的地址发生改变时符号名称并不会随之改变。

限制

符号访问存在下列限制：

- 地址指针化功能不可用，只能通过指针变量进行指针化操作。
- HMI 变量的数据类型和链接到该变量的 PLC 变量的数据类型必须相同。这意味着只能读取和写入变量的整个值。例如，在 PLC 中进行相应编程来改变变量中触发 HMI 报警的各个位。可在有关 PLC 编程的文档中查找到附加信息。

对更改的反应

对于符号访问，并不需要编译及下载所有的项目更改。PLC 程序发生以下更改时，才需要编译和下载项目。

对 PLC 变量的更改

当 PLC 变量发生更改时，需要对 PLC 程序进行编译和下载。

如果只是 PLC 变量的地址偏移发生改变（例如从 %MB5 到 %MB10 时），则无需对项目进行编译和下载。当地址区域、数据类型或变量名称发生更改时，也需要对项目进行编译和下载。

程序块中的更改

如果更改数据块的名称，则无需对 PLC 程序进行编译和下载。如果对程序块进行额外的更改，则总是需要对 PLC 程序进行编译和下载。

如果进行以下更改，则需要将项目编译并下载到 HMI 设备：

- 更改了数据块中某个变量的名称并且该变量已经与 HMI 变量建立了互连。
- 已在数据块中创建了一个结构并且更改了该结构中的某个名称。
- 更改数据块的编号时。
- 更改了互连到 HMI 设备的变量的数据类型时。
- 更改变量数组的名称时。
- 更改数组的限制时。

当更改、添加或删除块中某个未与 HMI 设备建立互连的变量时，无需将项目编译和下载到 HMI 设备。

8.3.2 使用变量

8.3.2.1 创建变量

8.3.2.1 创建外部变量

简介

使用外部变量，可通过 PLC 变量访问 PLC 中的地址。可通过链接到 HMI 设备的 PLC 中的符号表来执行访问。不能在 PLC 中直接指定地址。在 PLC 中创建 PLC 变量时会自动在符号表中创建条目。创建 HMI 变量后，可通过“PLC 变量”域直接访问符号表。可使用变量的符号名称来选择 PLC 变量。

有关符号访问的更多信息，请参见“[符号编程 \(页 333\)](#)”部分。

要求

- 已经打开项目。
- 项目中包含 PLC。
- 已组态 HMI 设备和 PLC 之间的连接。
- 项目中至少具有一个 PLC 变量。
- 巡视窗口已打开。

步骤

要创建外部变量，请执行如下操作：

1. 在项目树中打开“HMI 变量”文件夹，然后双击“HMI 变量”。打开“HMI 变量”编辑器。
2. 双击编辑器表“名称”列中的“添加”。创建新变量。
3. 在巡视窗口中，选择“常规”类别，必要时，在“名称”域中输入一个唯一的变量名称。此变量名称在整个项目中必须是唯一的。
4. 在“连接”框中选择到所需 PLC 的连接。如果需要的连接未显示，则必须先创建与 PLC 的连接。在“硬件和网络”编辑器中创建连接。
5. 在“PLC 变量”域中，单击



按钮，在对象列表中选择所需的 PLC 变量。单击



按钮以确认做出的选择。

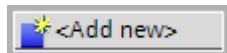


6. 在“采集周期”域中，选择更新变量所需的周期时间。

7. 为了进行归档，请在“注释”域中输入有关变量使用的信息。

另外，还可以在“HMI 变量”编辑器的表中直接组态所有变量属性。要查看隐藏列，可使用快捷菜单激活列标题。

也可以通过在对象列表中单击



按钮直接在使用处创建新变量。然后可以在“属性”窗口中组态该新变量。

结果

外部变量已创建完毕并已链接到 PLC 中的 PLC 变量。在附加的步骤中，可通过其它方式组态变量（例如，值改变时设置一个事件）。

参见

符号编程 (页 333)

8.3.2.1 创建内部变量

简介

对于内部变量，必须至少设置名称和数据类型。选择“内部变量”项，而不是与 PLC 连接。为了进行归档，最好为每个变量输入注释。

要求

已经打开项目。

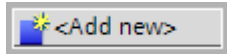
步骤

1. 在项目树中打开“HMI 变量”文件夹，然后双击“HMI 变量”。
打开“HMI 变量”编辑器。
2. 如果巡视窗口未打开，则可以在“视图”菜单中选择“巡视窗口”命令。

3. 在巡视窗口“属性”组的“常规”下，在“名称”域中输入一个唯一的变量名称。此变量名称在整个项目中必须是唯一的。
4. 在“连接”域中选择“内部变量”作为连接。
5. 在“数据类型”域中选择所需的数据类型。
6. 在“长度”域中，必须根据所选的数据类型指定将存储在变量中的最大字符数。长度由数字变量的数据类型自动定义。
7. 还可以输入关于变量使用的注释。为此，单击“属性”组中的“注释”，并输入注释文本。

另外，还可以在“HMI 变量”编辑器的表中直接组态所有变量属性。要查看隐藏列，可使用快捷菜单激活列标题。

也可以通过在对象列表中单击



按钮直接在使用处创建新变量。然后可以在“属性”窗口中组态该新变量。

结果

内部变量已创建。现在，可在项目中使用该变量。

现在还可以组态变量，例如，定义起始值和限制值。

8.3.2.1 创建多个变量

简介

在“HMI 变量”编辑器中，通过自动填充表格中变量下面的行，可以创建大量相同的变量。

使用自动填充功能时，变量名称和内存位置将自动递增。

也可使用自动填充功能以单一变量属性填充下方的表格单元，从而修改相应的变量。

如果对已经在“HMI 变量”编辑器中填充的单元格应用自动填充功能，则这些单元格将被覆盖。

自动填充每次操作仅限 100 个变量。

要求

- 已经打开项目。
- “HMI 变量”编辑器已打开。
- 将要用作其他变量模板的变量已组态。

步骤

1. 如果想要创建新的变量，在“名称”列中标记要用作新变量模板的变量。

如果想要将某个变量的属性复制给其下方的变量，选择包含该属性的单元格。

所选择的单元格将以高亮颜色显示，并将在其右下角显示一个小蓝方块。如果将鼠标移动到此方块上面，则光标变为黑色十字。

HMI 变量	
名称 ▲	连接
Motor_State	<Internal tag> ...
<添加新许可证> +	

- 按住鼠标按钮，将此方块拖动到下面要自动填充的单元格上面。
标记将扩展到覆盖该区域。

HMI 变量	
名称 ▲	连接
Motor_State	<Internal tag> ...
<添加新许可证>	
+ (dotted border)	

- 现在释放鼠标按钮。所有标记的单元格将被自动填充。
将在标记区域的所有空白单元格中创建新变量。

HMI 变量	
名称 ▲	连接
Motor_State	<Internal tag> ...
Motor_State_1	<Internal tag>
Motor_State_2	<Internal tag>
Motor_State_3	<Internal tag>
Motor_State_4	<Internal tag>
Motor_State_5	<Internal tag>

结果

根据所选择的单元格的不同，这个功能可以自动填充单个属性或创建新变量。

8.3.2.2 编辑变量

8.3.2.2 编辑变量

简介

始终都可以对变量进行重命名、复制或删除操作。
如果重命名变量，则新名称必须在整个项目中唯一。

如果使用“复制”命令将变量复制到剪贴板，也会复制链接到该变量的对象和引用。例如，如果变量链接到某个 PLC 连接，则也会复制该连接。

如果使用“插入”命令将变量添加到另一个设备，则将添加变量，而不添加连接的引用。只插入这些引用的对象名。如果在目标系统中存在相同名称和有效属性的引用，则现有的引用将连接到所复制的变量上。

如果使用“扩展插入”命令将变量添加到另一个设备，则将一起添加变量及其链接的对象和引用。如果目标系统中已有同名对象和引用，则插入的对象和引用的名称将增加一个数字。

要求

- 想要重命名、复制或删除的变量必须存在。
- 打开“变量”编辑器。

重命名变量

1. 在“名称”域中，在“HMI 变量”编辑器表中选择变量。
2. 从快捷菜单中选择“重命名”。
3. 输入新名称。

变量以新的名称显示。

此外，也可以在使用点的“对象”窗口中重命名变量。

复制变量

1. 在“HMI 变量”编辑器或“详情”窗口中选择一个或多个变量。
2. 从快捷菜单中选择“复制”。
3. 单击要插入变量的点。例如，在同一个设备中单击另一个变量组，或在另一个设备中单击“HMI 变量”编辑器。
4. 从快捷菜单中选择“插入”或“扩展插入”命令。如上所述插入变量。

删除变量

1. 在“HMI 变量”编辑器中选择一个或多个变量。
2. 在“工具”菜单中选择“交叉引用”命令。在“交叉引用”编辑器中，可查看变量用于何处。采用该方式，可发现删除变量将对项目产生哪些影响。
3. 在变量的弹出式菜单中选择“删除”。

所有标记的变量将被删除。

8.3.2.2 更改变量组态

简介

可随时修改变量，以适应不同的项目要求。

原理

WinCC 提供几种方法更改变量组态。

- “HMI 变量”编辑器

如果需要一个表格来提供多个变量的总览，则使用“HMI 变量”编辑器来组态变量。在“HMI 变量”编辑器中，可执行诸如此类的任务：比较和调整多个变量的属性或按变量属性对变量进行排序。

- 对象列表

如果要直接在使用时修改变量，可使用对象列表。为此，在对象列表中选择变量，然后使用快捷菜单打开变量的属性窗口。

如果改变变量属性，而这种改变引起与另一属性的冲突，该属性将以颜色高亮显示，以提醒用户注意。例如，如果将变量连接到不支持该数据类型的另一 PLC 上，就有可能发生这种情况。

8.3.2.2 同时组态多个变量

简介

在 WinCC 中，可在单个操作中将相同属性分配给多个变量。这样可以提高编程效率。

要求

- 已经创建了要组态的变量。
- 打开巡视窗口。

步骤

1. 在“HMI 变量”编辑器中，选择要同时组态的所有变量。

如果所选择的属性对于所有变量都相同，则此属性的设置出现在巡视窗口中。否则，相关域将保留空白。

2. 可以在巡视窗口中或直接在“HMI 变量”编辑器中定义公共属性。

结果

所有标记的变量都将被重新组态。

要编辑变量间彼此不同的变量属性，只要清除多选即可。

8.3.2.2 在一个画面中同时使用多个变量

简介

在 WinCC 中，可以在单个操作中在一个画面上创建与变量链接的多个 I/O 域。因此，对于过程变量，可以为变量创建一个网络和连接。这样可以提高编程效率。

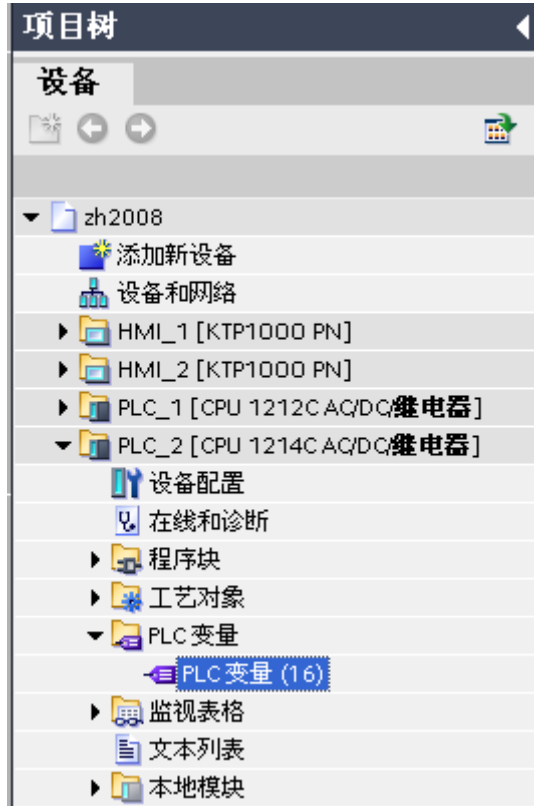
要求

- 已创建几个变量。

- 画面打开。

步骤

1. 在项目树中，在“变量与连接”下选择所需的变量组。



2. 选择项目树底部的详细视图。详细视图显示所选变量组中存在的变量。



3. 在详细视图中，按住 <Ctrl> 或 <Shift> 键同时选择所需要的变量。
4. 将变量拖动到画面中。对于每个变量，此操作创建连接到该变量的 I/O 域。

注意

将过程变量拖放到详细视图窗口中时，会在“硬件和网络”编辑器中创建网络和连接。

8.3.2.3 组态变量

8.3.2.3 变量限制

简介

可限制数字变量的值范围。

原理

可为数字变量指定由上限和下限所定义的取值范围。

当过程值超出变量的限制值时，可产生模拟量报警输出。

如果过程值超出取值范围的上限或下限，可产生模拟量报警输出或触发函数列表。如果操作员输入的变量值在组态的值范围外，则拒绝输入。不会接受该值。

注意

在“模拟量报警”编辑器中输入越限时的模拟量报警文本。

应用实例

例如，在变量值进入临界范围时使用限制值来及时警告操作员。

8.3.2.3 定义变量的限制值

简介

对于数字变量，可通过定义下限和上限来指定取值范围。

可以组态当超出其中一个限制时，系统发送模拟量报警。

另外，组态系统在变量值低于或高于其所组态的数值范围时处理函数列表。

要求

- 已创建了想要为其设置限制值的变量。
- 巡视窗口已打开，其中显示该变量的属性。

步骤

要定义变量限制，请按如下步骤进行操作：

1. 在巡视窗口的“属性”组中单击“限制”。如果要将其中的一个限制定义为常量值，使用



按钮选择“常量”。在相关域中输入数字。

如果要将其中的一个限制定义为变量值，使用



按钮选择“HMI 变量”。在相关域中输入值。使用对象列表为此限制定义变量。

2. 如果要在值越限时输出模拟量报警，启用限制旁的“创建模拟量报警”选项。
使用“模拟量报警”编辑器输入模拟量报警文本。
3. 要设置变量的其它限制，使用相应的设置重复步骤 2 和 3。

其它方法

此外，还可以直接在“HMI 变量”编辑器表中组态上限和下限以及相关模拟量报警的显示。要查看隐藏列，可使用快捷菜单激活列标题。

结果

已经为所选变量设置了由上限和下限所定义的取值范围。根据组态，越限时在运行系统中输出模拟量报警。如果超过或低于值范围，则执行函数列表。

组态函数列表

可按以下方式针对超出取值范围的情况组态函数列表：

1. 如果要在值下降到值范围以下时启动函数列表，则在“事件”组中单击“超出下限”。在该对话框中创建函数列表。
2. 如果要在值超过值范围时启动函数列表，则在“事件”组中单击“超出上限”。在该对话框中创建函数列表。

8.3.2.3 变量的起始值

运行系统启动时的变量值

可以为数字变量组态起始值。运行系统启动时变量将预置为该值。采用这种方式，可确保运行系统启动时变量具有已定义的状态。

对于外部变量，起始值将显示在 HMI 设备上，直到由 PLC 或输入所覆盖。

应用实例

可将默认值分配给 I/O 域。输入期望的默认值作为与 I/O 域链接的变量的起始值。

8.3.2.3 定义变量的起始值

简介

在 WinCC 中，可组态数字变量的起始值，即运行系统启动时采用的值。

要求

- 已创建要为其定义起始值的变量。
- 巡视窗口已打开，并显示该变量属性。

步骤

要组态起始值，请按如下步骤进行操作：

1. 在巡视窗口的“属性”组中单击“设置”。
2. 输入期望的“起始值”。

其它方法

另外，也可以直接在“HMI 变量”编辑器中组态起始值。要查看隐藏列，可使用快捷菜单激活列标题。

结果

为变量选择的起始值将传送到项目。

8.3.2.3 在运行系统中更新变量值

简介

变量包含运行系统运行时可以改变的过程值。内部变量和外部变量的数值改变在处理上有所不同。

原理

运行系统启动时，变量的值等于起始值。在运行系统中，可以改变变量值。

在运行系统中，提供下列选项更改变量值：

- 通过运行用于数值更改的系统函数，例如“SetTag”系统函数。
- 通过输入，例如在 I/O 域中输入。
- 在 PLC 中外部变量的值改变。

更新外部变量值

外部变量值有下面几种更新方式：

- 循环操作

如果选择“循环操作”采集模式，则只要变量显示在画面中，即会在运行系统中得到更新。采集周期决定 HMI 设备上变量值的更新周期。循环采集基于所选的扫描周期。

- 循环连续

如果选择“循环连续”采集模式，则变量将在运行系统中连续更新，即使变量不在当前打开的画面上仍是如此。例如，如果将变量组态为当变量值变化时触发函数列表，则为该变量激活此设置。

只能将“循环连续”设置用于那些确实必须连续更新的变量。频繁读取操作将增加通信的负担。
- 请求时

如果选择“请求时”采集模式，则不循环更新变量。例如，只有当使用“Update Tag”系统函数请求时才更新变量。

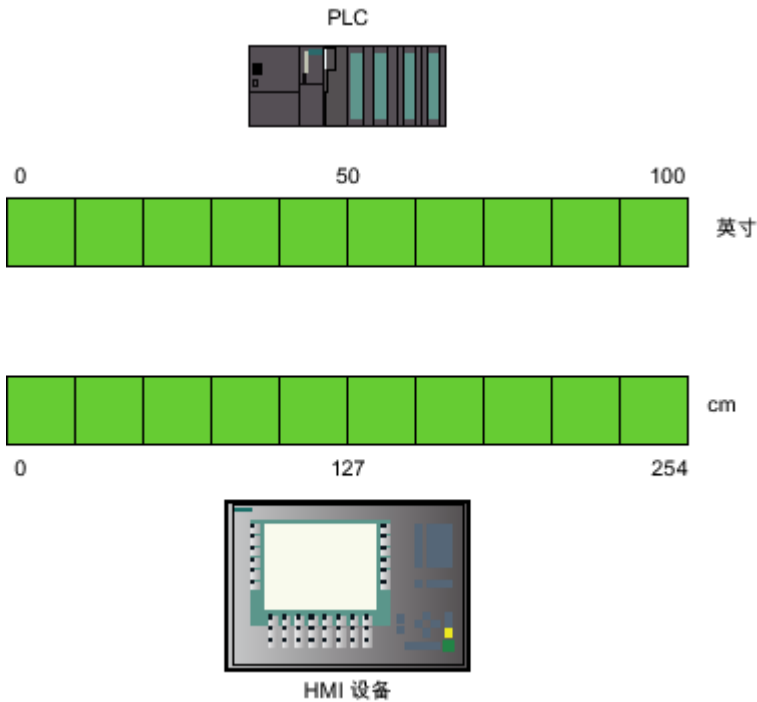
8.3.2.3 变量的线性转换

简介

数字数据类型可使用线性转换进行处理。外部变量在 PLC 中的过程值可映射到项目中的特定取值范围。

原理

为了对变量进行线性转换，必须在 HMI 设备和 PLC 上各指定一个数值范围。数值范围将相互线性映射。



一旦来自 HMI 设备的数据写入外部变量，它将被自动映射到 PLC 的数值范围。一旦来自 HMI 设备的数据从外部变量读取，将反向完成相应的转换。

注意

也可使用系统函数“LinearScaling”和“InverseLinearScaling”来自动转换过程值。

应用实例

用户输入以厘米为单位的长度尺寸，但 PLC 期望输入英寸值。输入的值在传送到控制器之前自动进行转换。使用线性转换，PLC 上的值范围 [0 - 100] 可映射到 HMI 设备上的值范围 [0 - 254]。

8.3.2.3 线性变量标定

简介

为了对变量进行线性标定，必须在 HMI 设备和 PLC 上各指定一个数值范围。数值范围将相互线性映射。

要求

- 要应用线性标定的外部变量必须存在。
- 巡视窗口已打开，其中显示该变量的属性。

步骤

要将线性标定应用于变量，执行下列步骤：

1. 在巡视窗口的“属性”组中单击“线性标定”。
2. 单击“启用”打开线性标定。
例如，为了进行测试，可使用该选项临时关闭线性标定。先前为线性标定所进行的设置保持不变。
3. 在“PLC”区域中，输入要应用于 PLC 上过程值的值范围起始值和结束值。
4. 在“HMI 设备”区域中，输入要应用于 HMI 设备上过程值的值范围起始值和结束值。

结果

在运行系统中，自动将数据从一个值范围映射到另一个值范围。

注意

也可以使用“LinearScaling”和“InverseLinearScaling”系统函数自动转换过程值。

8.3.2.3 将变量连接到另一个 PLC

简介

在 WinCC 中，可随时更改变量与 PLC 的连接。例如，更改工厂组态时，需要此操作。根据所选的 PLC，可能需要修改变量的组态。必须更改的变量属性将以高亮颜色显示。

要求

- 希望改变其连接的外部变量必须已经存在。
- 与 PLC 的连接必须已经存在。
- 该变量的“属性”窗口已打开。

步骤

要更改变量与 PLC 的连接，请执行以下步骤：

1. 在巡视窗口的“属性”组中单击“常规”。
2. 从“常规”区域中选择新连接。

要更改的变量属性将在“HMI 变量”编辑器和巡视窗口中以高亮颜色显示。

3. 改变变量所有高亮显示的属性，以满足新 PLC 的要求。

结果

外部变量连接到了新 PLC。

8.3.2.3 变量的间接寻址

原理

在指针化（间接寻址的一种类型）中，首先确定运行时所使用的变量。定义一个变量列表专门用于指针变量。在运行时从变量列表中选择相关的变量。变量的选择取决于索引变量的值。

在运行系统中，系统首先读取索引变量的值。随后将访问在变量列表的相应位置中指定的变量。

应用实例

使用间接寻址，可组态下列情况：

操作员从选择列表中选择多台机器中的一台。根据操作员的选择，来自所选机器的数据将显示在输出域中。

要组态这类情况，可为符号 I/O 域组态索引变量。在 I/O 域组态指针变量。组态指针变量的变量列表，以反映选择列表的结构。

如果操作员选择另一台设备，则索引变量的值将改变。此时，选择域将显示变量列表（指针变量）中新索引值指向的变量的内容。

8.3.2.3 间接变量寻址

简介

使用间接寻址时，在运行系统中首先确定使用的变量。定义一个变量列表，而不是单个变量。列表条目由索引值和所使用变量的名称组成。使用索引变量，可以控制将访问变量列表中的哪一条目。

要求

- 希望对其进行间接寻址的变量必须已经存在。
- 索引变量必须存在。
- 变量列表中包含的变量必须已经存在。
- 巡视窗口已打开，并显示该变量属性。

步骤

要间接寻址变量，请执行以下步骤：

1. 在巡视窗口的“属性”组中单击“指针化”。
2. 单击“启用”打开间接寻址。
例如，为了进行测试，可使用该选项临时关闭间接寻址。先前为间接寻址所进行的设置保持不变。
3. 选择“索引变量”或使用对象列表定义新变量。
4. 在变量列表中单击“变量”列的第一个条目。
5. 选择变量作为列表条目或使用对象列表定义新变量。
“索引”列中的条目将自动生成。
6. 对希望添加到变量列表中的所有变量，重复步骤 5。
7. 如有必要，使用拖放功能来改变列表中的条目顺序。

结果

在运行系统中，系统将通过索引变量中的当前值动态访问变量列表中的相应变量。

8.3.2.3 使用变量触发函数

简介

可以使用变量值作为运行时某个动作的触发事件。要在运行系统中启动操作，为变量组态函数列表。在函数列表中添加一个或多个系统函数。组态的事件发生时，将会处理函数列表。

下列事件可用于变量：

- 变量值变化
变量值的每次改变都将触发函数列表的执行。
如果变量包含数组，只要数组中的一个元素变化，就会处理函数列表。

- 超出变量的上限
在高于上限时执行函数列表。
- 超出变量的下限
在低于下限时执行函数列表。

要求

- 希望将其值用作事件的变量已经存在。
- 巡视窗口已打开，其中显示该变量的属性。

步骤

要组态带有函数列表的变量，请执行以下步骤：

1. 在巡视窗口的“事件”组中，单击要为其创建函数列表的事件。
显示与所选择的事件相关的函数列表。
2. 单击带有“<无函数>”条目的行。第二个表格列包含选择按钮。
3. 单击选择按钮，选择系统函数。
4. 定义参数值。

结果

当组态的事件在运行系统中发生时，将会处理函数列表。

8.3.2.3 定义变量的采集周期

简介

在运行系统中，外部变量值可通过与变量链接的 PLC 来更改。为了确保在 PLC 改变变量值时通知 HMI 设备，必须在 HMI 上更新变量值。在过程画面中显示变量时，将定期更新变量值。定期更新的时间间隔由采集周期进行设置。也可以连续更新。

要求

- 已创建要为其定义采集周期的变量。
- 巡视窗口已打开，并显示该变量属性。

步骤

要组态变量的采集周期，请执行以下步骤：

1. 在巡视窗口的“属性”组中，单击“常规”。
2. 如果要在画面中显示变量或记录变量时定期更新变量，则选择“循环操作”作为采集模式。
或者：
如果要在画面中未显示变量时仍定期更新变量，则选择“循环连续”作为采集模式。
例如，为变量选择“循环连续”模式，该变量针对自身值变化而组态了函数列表，且不在画面中直接可见。

3. 在“采集周期”域中选择所需要的周期，或使用对象列表定义新的采集周期。

另外，还可以直接在“HMI 变量”编辑器中组态采集周期。要查看隐藏列，可使用快捷菜单激活列标题。

注意

对于必须连续更新的变量，只能使用“循环连续”设置。频繁读取操作会加重通信负载。

结果

在运行系统中以所选择的采集周期更新所组态的变量。

8.3.3 使用数组

8.3.3.1 关于数组的基础知识

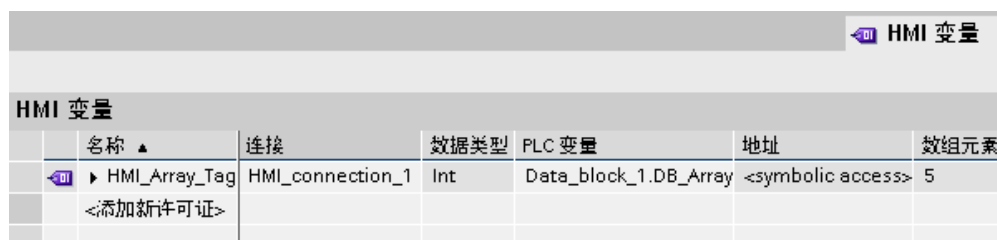
定义

数据类型相同的数组数据连续排列，并可通过索引在地址空间内寻址以允许访问这些数据。为了寻址数据的各个数组元素，数组使用从“1”开始的整数索引。每个数组元素的属性都是相同的，并且都在 PLC 程序的数据块中的数组变量处进行组态。



Data_block_1					
	名称	数据类型	初始值	保持性	注释
1	▼ Static			<input type="checkbox"/>	
2	▼ DB_Array	Array [1 .. 5] of int ▼		<input type="checkbox"/>	
3	DB_Array[1]	Int	0	<input type="checkbox"/>	
4	DB_Array[2]	Int	0	<input type="checkbox"/>	
5	DB_Array[3]	Int	0	<input type="checkbox"/>	
6	DB_Array[4]	Int	0	<input type="checkbox"/>	
7	DB_Array[5]	Int	0	<input type="checkbox"/>	

然后，可以将 HMI 变量连接到数据块中的数组变量。HMI 变量与数组之间建立镜像，因此它继承了数据块中数组变量的属性。



HMI 变量						
	名称 ▲	连接	数据类型	PLC 变量	地址	数组元素
	▶ HMI_Array_Tag	HMI_connection_1	Int	Data_block_1.DB_Array	<symbolic access>	5
	<添加新许可证>					

优势

可使用单个数组变量一次组态多个具有相同属性的数组元素。然后，可以在组态中将每个数组元素用作任何其它变量。

限制

下列限制适用于数组的使用：

- 并非所有 HMI 设备都支持数组变量。
- 在属性方面：例如，不是所有数组元素都有限制。

应用实例

可在下列情况下使用数组变量：

- 要在分析趋势中编组过程值：例如，将过程值映射到在不同时间点获得的趋势。
- 要访问在趋势中编组的特定值：例如，通过逐步增加索引变量输出所有分析趋势值。
- 要使用连续位号组态离散量报警。
- 要将机器数据记录保存到单个变量中。

用于运行时的许可证规则

在 WinCC Runtime 中，将一个数组变量计为一个 PowerTag，而不管数组元素的数目是多少。

特性



警告 系统负载增加且性能下降

对单个数组元素的读或写访问始终包括对数组变量的所有数组元素进行读或写访问。与传输基础数据类型的数据相比，较大数组与 PLC 之间的数据传输通常要花费更长的时间。这可能会导致通信过载并最终导致通信中断。

实例：

- 组态由 100 个数组元素组成的数据类型为“Real”的数组变量。
- 如果更改一个长度为四个字节的数组元素，则会将 100 x 4 个字节写入 PLC。



小心
数组变量中的数据不一致

在时间 t_1 读取整个数组。如果更改了某个数组元素，则在数组中，修改的数组元素将进行替换。在时间 $t_3 > t_1$ 将数组写回 PLC。如果数组在时间 t_2 时再次进行更改，则在时间 t_3 时， t_2 时的更改值会被时间 t_1 时的值改写。因此在时间 t_3 时，数组数据会出现不一致。

应始终阻止 HMI 设备和 PLC 同时向同一数组变量写入值。使用配方数据记录的同步传送以将数组变量与 PLC 同步。

8.3.3.2 创建数组变量

简介

创建数组变量以组态具有相同数据类型的大量变量。数组元素保存到连续的地址空间中。在所连接的 PLC 的 PLC 程序内，于数据块中创建数组变量。然后，将数组变量连接到 HMI 变量。

要求

- 项目包含 PLC。
- “HMI 变量”编辑器已打开。

步骤

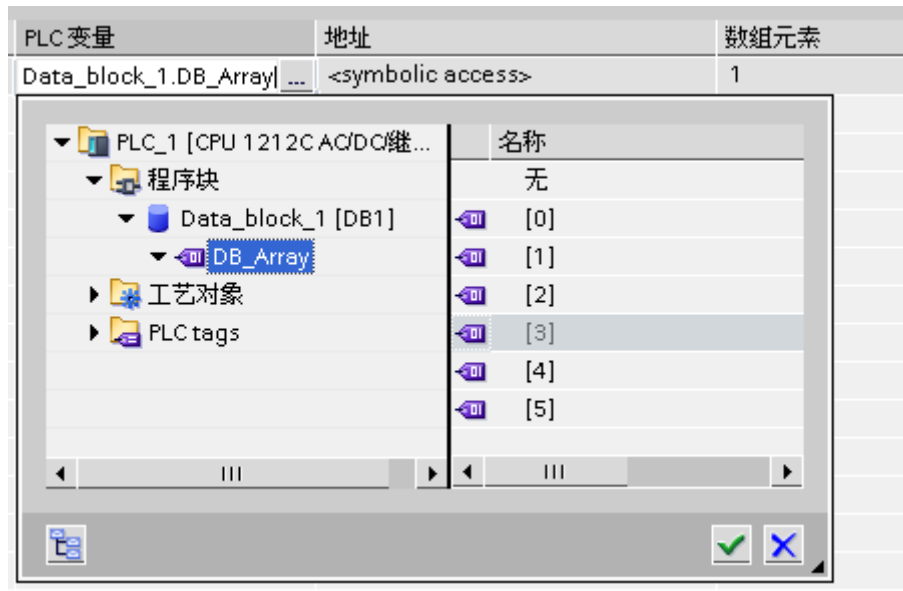
要创建数组变量，执行下列步骤：

1. 在项目树中打开 PLC 节点，然后双击“程序块”>“添加新块”。将打开具有此名称的对话框。
2. 使用该对话框创建一个数据块，然后在“程序块”编辑器中打开该数据块。
3. 在“名称”列的第一个空白域中输入数组变量的名称。
4. 选择“type 类型的 Array[lo .. hi]”作为数据类型。
5. 在数据类型中，以相应的用于编排数组元素的数字来替换“lo”和“hi”。用数组变量的期望数据类型替换“type”。注意数据类型规范中的空格。
实例：“int 类型的 Array[1 .. 5]”将生成一个含有 5 个同名元素的数组变量，从 1 到 5 为其连续编号。数组元素的数据类型为“整型”。

Data_block_1					
	名称	数据类型	初始值	保持性	注释
1	▼ Static			<input type="checkbox"/>	
2	▼ DB_Array	Array [1 .. 5] of int ▼		<input type="checkbox"/>	
3	DB_Array[1]	Int	0	<input type="checkbox"/>	
4	DB_Array[2]	Int	0	<input type="checkbox"/>	
5	DB_Array[3]	Int	0	<input type="checkbox"/>	
6	DB_Array[4]	Int	0	<input type="checkbox"/>	
7	DB_Array[5]	Int	0	<input type="checkbox"/>	

6. 选择菜单命令“编辑”>“编译”。编译数据块，然后就可用于项目。

7. 在“HMI 变量”编辑器中创建一个新的变量。
8. 在“连接”域中选择到 PLC 的连接。
9. 在“PLC 变量”域中，打开选择对话框并在相应的 PLC 下选择已创建的数据块。在右侧窗口中选择数组变量并对选择进行确认。



结果

已创建数组变量并连接到 HMI 变量。HMI 变量会在 HMI 设备上建立 PLC 数组的镜像。自动分配 PLC 存储空间。数组元素的属性从上层数组变量继承而来。

8.3.3.3 数组示例

简介

数组变量将多个变量分组，以形成一个数据结构，例如，由 100 个数组元素组成的数据结构。数组元素是可在组态中的任何位置使用的变量。还可在以下位置使用数组变量：

- 在“报警”编辑器中
- 在“配方”编辑器中
- 用于地址指针化
- 在趋势视图中

实例

可以组态含相应数目的数组元素的数组变量，以处理多个同一数据类型的变量。

- 例如，可通过指针化索引变量间接访问单个数组元素。
- 使用这些索引变量来操作和监视数组元素。
- 使用数组变量创建多个配方元素，这些配方元素自动与相应的数组元素互连。

- 通过配方同步数组变量和 PLC。

8.3.4 使用周期

8.3.4.1 关于周期的基础知识

简介

周期用于控制运行中定期发生的操作。一般应用包括采集周期和更新周期。

原理

在运行系统中，定期执行的操作由周期来控制。周期的典型应用包括：

- 采集外部变量
采集周期决定 HMI 设备何时从 PLC 读取外部变量的过程值。对采集周期进行设置，使其适合过程值的改变速率。例如，烤炉的温度变化明显比电气驱动的速度慢。
不要将采集周期设置得太小，因为这将不必要地增加过程的通信负载。
- 刷新画面
更新周期决定画面的刷新频率。

注意

可用于特定设备

更新周期在精简面板上是固定的。

周期的最小可能值取决于项目所使用的 HMI 设备。对于大多数 HMI 设备，该数值为 100 毫秒。其他所有周期值总是最小值的整数倍。

应用实例

周期可用于下列任务：

- 定期更新变量。
- 注意维护间隔时间。

8.3.5 显示变量

8.3.5.1 在画面中输出变量值

简介

在运行时，可以以趋势的形式将变量值输出到操作员设备的画面中。趋势是变量在运行时所采用值的图形表示。趋势用“趋势显示”图形对象来表示。趋势显示的过程值由 PLC 从正在进行的过程中载入。

要显示的值在固定的、可组态的周期内分别确定。周期性触发趋势适合于表示连续曲线，如电机运行温度的变化。

显示的值

需要在画面中组态趋势视图，以便在 HMI 设备上显示变量值。组态趋势视图时，可指定要显示哪些变量值。

可通过定义周期时间来控制趋势显示的更新。

8.3.5.2 为来自 PLC 的值组态趋势显示

简介

可使用趋势视图以图形形式表示过程期间的变量取值。

要求

- 画面打开。
- 巡视窗口打开，并显示趋势视图属性。

步骤

要组态趋势视图，执行下列步骤：

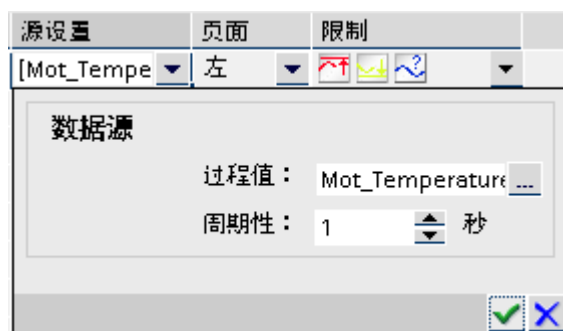
1. 将“趋势视图”对象从“控制”组中的工具箱添加到画面。



2. 在巡视窗口中，从“属性”组中选择“趋势”类别，然后在“名称”列中双击“<添加>”。



3. 在“名称”列中为趋势分配一个名称。
4. 在“样式”列中，使用选择按钮打开“样式”对话框，然后选择线的样式。
5. 在“趋势值”列中选择趋势值的数目。
6. 在“设置”列中，使用选择按钮打开“数据源”对话框，选择为趋势提供值的变量。
指定从 PLC 中读取变量的周期。



7. 可以在巡视窗口的对话框中进行其它设置。例如，可以在“表格”类别中选择“显示表格”选项，以便在趋势视图下显示数值表。

注意

如果按住 <CTRL> 键同时双击趋势视图，则激活趋势视图。在激活模式下，可以在数值表的表标题中设置列的宽度和列的位置。为激活趋势视图，缩放因子必须设置为 100 %。

结果

在运行系统中，所选变量的值被显示在组态的趋势视图中。

8.4 使用报警

8.4.1 基础知识

8.4.1.1 WinCC 中的报警系统

简介

报警系统允许您在 HMI 设备上显示和记录运行状态和工厂中出现或发生的故障。

报警可能有下列内容：

编号	时间	日期	报警文本	状态	报警类别
5	12:50:24 :590	24.02.2 007	锅炉压力高于上限。	进入 离开	警告： 颜色为红色

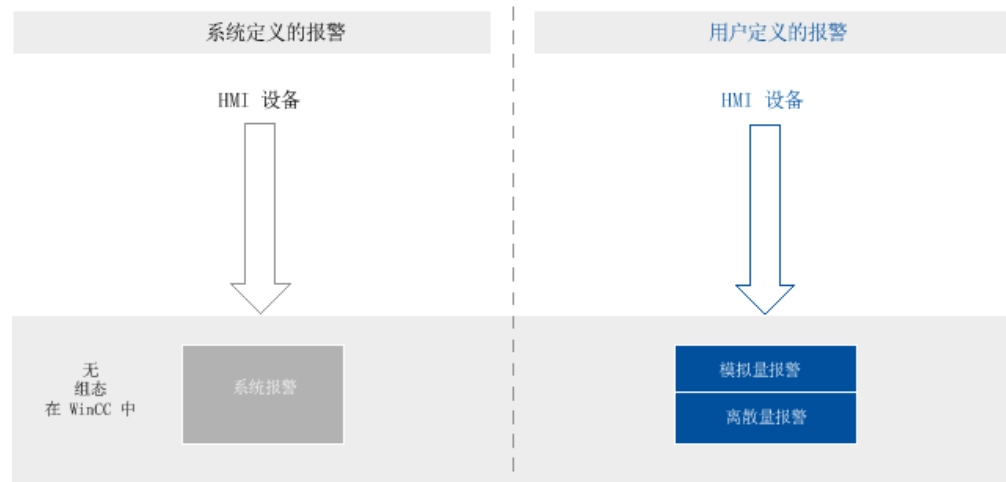
WinCC 中的报警系统

报警系统可处理多种报警类型。报警过程可划分为系统定义的报警和用户定义的报警：

- 用户定义的报警用于监视工厂过程。
- 系统定义的报警用于监视 HMI 设备。

检测到的报警事件显示在 HMI 设备上。有目标地访问报警和有关单个报警的补充信息可确保迅速定位和清除故障。这可减少停机，甚至完全避免停机。

下图显示了报警系统的结构：



参见

[报警类型概述 \(页 1107\)](#)
[报警状态 \(页 1110\)](#)
[确认模型 \(页 1114\)](#)
[报警组 \(页 1115\)](#)
[确认报警 \(页 1113\)](#)
[报警组件和属性 \(页 1115\)](#)
[运行系统的报警 \(页 1138\)](#)
[报警的系统函数和事件 \(页 1144\)](#)
[报警组态概述 \(页 1117\)](#)

8.4.1.2 报警类型

8.4.1.2 报警类型概述

简介

报警类型适用于监视工厂的多个方面。来自各个报警类型的报警均以不同方法组态和触发。可以在“报警”编辑器的相关选项卡上，使用各个报警类型来组态报警。

WinCC 中的报警类型

WinCC 支持下列报警类型：
用户定义的报警

- **模拟量报警**
 - 模拟量报警用于监视是否超出限制值。
- **离散量报警**
 - 离散量报警用于监视状态。
- **PLC 报警**
 - 在 STEP 7 中组态 PLC 报警。
 - 然后在 WinCC 中进一步处理 PLC 报警。

注意
可用于特定设备

PLC 报警不可用于精简面板。

系统定义的报警

- **系统定义的 PLC 报警**
 - 系统定义的 PLC 报警由诊断报警 (SIMATIC S7) 和系统错误 (SFM) 组成
 - 系统定义的 PLC 报警用于监视 PLC。

注意
可用于特定设备

系统定义的 PLC 报警不可用于精简面板。

- **系统报警**
 - 系统报警属于 HMI 设备，并导入到项目中。
 - 系统报警用于监视 HMI 设备。

参见

WinCC 中的报警系统 (页 1106)

8.4.1.2 系统定义的报警

8.4.1.2 系统报警

报警的实例

- “在文本模式下，行打印的缓冲区溢出”
- “产生了内部错误。结束运行系统”

描述

系统报警指示系统状态，以及 HMI 设备和系统之间的通信错误。

在“运行系统设置”>“报警”下，指定系统报警显示在 HMI 设备上的时间。

支持

HMI 设备手册中包含可能的系统报警以及原因和可能的补救措施的列表。如果您因为 HMI 设备上的系统报警而与在线支持联系，则需要提供报警编号以及系统报警中使用的变量。

8.4.1.2 用户定义的报警

8.4.1.2 模拟量报警

描述

模拟量报警指示过程期间超出限制值的情况。

实例

果汁混合工厂中的搅拌器的速度不能太高或太低。可组态模拟量报警来监视搅拌器的速度。如果超过了搅拌器速度的上下限值，则会在 HMI 设备上输出报警，例如包含下列报警文本：“搅拌器速度太低”。

参见

离散量报警 (页 1109)

8.4.1.2 离散量报警

描述

离散量报警指示当前过程的状态。

实例

果汁混合工厂由几个装配料的罐组成。为确保水、浓缩果汁、糖和香料的混合比正确，入口中的阀应在合适的时间打开和关闭。应监视此操作。

为所有阀门状态组态合适的离散量报警。如果在这四个罐中的一个罐上的阀门打开或关闭，则显示报警，如“水阀已关闭”。

因此，操作员可以监视工厂是否正常生产。

参见

模拟量报警 (页 1109)

8.4.1.3 报警状态

简介

报警在运行系统中呈现不同的报警状态。用户参考报警状态对过程执行的状况进行分析和报告。

注意
精简面板

报告和记录不可用于精简面板。

描述

每个报警有一个报警状态。报警状态由下列事件组成：

- **进入 (I)**
满足触发报警的条件。显示报警，如“锅炉压力太高”。
- **离开 (O)**
不再满足触发报警的条件。因为锅炉已通风，所以不再显示报警。
- **确认 (A)**
操作员确认报警。

无需确认的报警

下表显示了不需要确认的报警的报警状态：

显示文本	状态	描述
I	进入	满足报警条件。
IO	离开	不再满足报警条件。

需要确认的报警

下表显示了需要确认的报警的报警状态：

显示文本	状态	描述
I	进入	满足报警条件。
IO	离开， 未确认	不再满足报警条件。操作员未确认报警。
IOA	离开， 随后确认	不再满足报警条件。此后操作员确认报警。
IA	进入， 已确认	满足报警条件。操作员确认报警。

显示文本	状态	描述
IAO	离开， 但先确认	不再满足报警条件。仍满足条件时操作员确认报警。

这些状态的每次出现都会在 HMI 设备上显示和记录，并打印协议。

注意

可以根据需要组态报警状态的显示文本。

参见

WinCC 中的报警系统 (页 1106)

8.4.1.4 报警类别

简介

在工厂中发生许多报警。它们具有不同的重要性。可以将项目的报警分配给报警类别，以便向用户清楚地显示哪些报警最重要。

描述

报警类别定义如何显示报警。报警类别指定用户是否需要确认该报警类别的报警以及如何确认报警。

在 WinCC 中生成强制确认的新报警类别。

注意

报警类别的显示模式的选择取决于 HMI 设备上的选项。

如何使用报警类别的实例

- 报警“风扇 1 的速度达到公差范围上限”具有报警类别“警告”。以蓝色文本和白色背景显示此报警。此报警不需要确认。
- 将报警“风扇 2 的速度超过警告范围上限”分配给“错误”报警类别。报警在运行系统中以红色背景显示并频繁闪烁。用户确认前，报警将一直显示。

使用报警类别

可使用下列报警类别为项目定义报警的确认模型和显示：

- 预定义的报警类别

不能删除预定义的报警类别，并且只能在有限的范围内对其进行编辑。已为每个 HMI 设备创建多达四个报警类别。

- 自定义报警类别

可以创建新报警类别，组态想要如何显示报警，并为来自此报警类别的报警定义确认模型。自定义报警类别的可能数量取决于项目中使用的运行系统。

- 项目范围报警类别

项目范围的报警类别显示在项目树的“共享数据”>“报警类别”下，并且可用于 HMI 设备的报警。项目范围的报警类别源于 STEP 7 中的报警组态。如果需要，可在 WinCC 中创建额外的项目范围报警类别。

注意

精简面板

项目范围的报警类别不可用于精简面板。

预定义的报警类别

在 WinCC 中已为每个 HMI 设备创建下列报警类别：

用户定义报警的报警类别

- “警告”

“警告”报警类别用于显示常规状态以及过程中的日常操作的离散量和模拟量报警。用户不确认来自此报警类别的报警。

- “错误”

“错误”报警类别用于显示过程中的关键或危险状态，或者越限情况的离散量和模拟量报警。用户必须确认来自此报警类别的报警。

系统定义报警的报警类别

- “系统”

“系统”报警类别包含显示 HMI 设备和 PLC 的状态的报警。

- “诊断”

“诊断”报警类别包含显示 SIMATIC S7 中的状态和报警的报警。用户不确认来自此报警类别的报警。

注意

精简面板

“诊断”报警类别不可用于精简面板。

8.4.1.5 确认

8.4.1.5 确认报警

简介

为确保工厂操作员登记某一报警，将此报警组态为一直显示到操作员确认。过程中显示严重或危险状态的报警必须进行确认。

描述

报警的确认是一个根据需要加以记录和报告的事件。确认报警将报警的报警状态从“进入”更改为“已确认”。操作员确认报警时，表明确认他已处理了触发报警的状态。

注意

精简面板

报告和记录不可用于精简面板。

触发确认报警

在运行系统中可通过如下方式来触发确认报警：

- 授权的用户在 HMI 设备上确认
- 无需操作员输入，系统通过以下方式自动确认
 - 变量
 - 函数列表中的系统函数或脚本
 - PLC

注意

精简面板

脚本不可用于精简面板。

确认属于一组的报警

为使报警系统在运行系统中更清楚和易于使用，可以组态属于一组的报警，以使用户可通过单步操作员操作同时进行确认。

由 PLC 确认

如果需要，通过 PLC 自动确认离散量报警。通过“确认变量 PLC”中的位触发确认。在组态阶段定义位和变量。

在 HMI 设备上确认报警

在运行系统中，用户根据组态情况通过下列方式之一确认报警：

- 使用 HMI 设备上的确认按钮 <ACK>
- 使用报警视图中的按钮
- 使用组态的功能键或画面中的按钮

注意

HMI 设备上的确认按钮 <ACK>

为确保仅由获得授权的用户处理严重报警，可通过分配相应的权限，防止未经授权的用户使用 HMI 设备上的“ACK”按钮以及报警中的操作控件和显示对象。

注意

HMI 设备相关性

确认键 <ACK> 并非在所有 HMI 设备上均可用。

参见

WinCC 中的报警系统 (页 1106)

确认模型 (页 1114)

8.4.1.5 确认模型

概述

可定义报警类别的确认模型。根据此确认模型确认分配给此报警类别的报警。在 WinCC 中使用下列确认模型：

- 无需确认的单个报警
此报警无需确认即可进入和离开。没有来自系统的可见响应。
- 需要进入确认的单个报警
触发报警的事件一出现，就必须确认此报警。此报警保持未决状态，直到被确认。

多个报警的确认模型

为一起确认多个报警，将这些报警分配给一个报警组。

如果用户在运行系统中确认属于某一报警组中的报警，则来自此报警组的所有报警将同时得到确认。

参见

WinCC 中的报警系统 (页 1106)

确认报警 (页 1113)

8.4.1.6 报警组

简介

在工厂中会发生来自不同区域和过程的许多报警。可使用报警组一起确认具有相同原因的报警。

报警组

可将相关报警编译到报警组中。可使用报警组监视工厂的各个部分，并根据需要一起确认相关报警。如果确认来自此报警组的一个报警，则也确认此报警组中的所有其它报警。

报警组可包含来自不同报警类别的报警。只有将需要确认的报警分配给报警组才有意义。

使用报警组

对于诸如下列报警，编译报警组是个好主意：

- 由相同故障引起的报警。
- 相同类型的报警
- 来自同一机器单元的报警，如“驱动器 XY 故障”
- 来自同一过程的相关部分的报警，如“冷却水供水故障”

在运行系统中的显示

在运行系统中，“报警组”列显示报警所属报警组的编号。

参见

WinCC 中的报警系统 (页 1106)

8.4.2 使用报警

8.4.2.1 报警组件和属性

概述

在 WinCC 中组态报警的组成部分。下表显示报警的基本组成部分：

报警类别	报警编号	时间	日期	报警状态	报警文本	报警组	信息文本
警告	1	11:09:14	06.08.2007	IO	达到最大速度	2	此报警是 ...
系统	110001	11:25:58	06.08.2007	I	切换到“在线”模式	0	此报警是 ...

报警类别

报警类别，如“警告”或“错误”。报警类别定义报警的下列内容：

- 确认模型
- 在运行系统中的外观（例如颜色）
- 记录

注意

精简面板

记录不可用于精简面板。

报警编号

通过唯一报警编号标识报警。报警编号由系统分配。必要时，可以将报警编号更改为连续的报警编号，以标识项目中相关联的报警。

时间和日期

每个报警都有一个时间戳，显示触发报警的时间和日期。

报警状态

报警具有事件“进入”、“离开”和“确认”。针对每个事件，都输出一个新报警和报警的当前状态。

报警文本

报警文本描述报警的原因。

报警文本可包含当前值的输出域。可插入的值取决于正在使用的运行系统。报警状态变化时保留值。

报警组

报警组将单个报警捆绑在一起。

信息文本

可以为每个报警组态单独的信息文本；用户可以在运行系统中显示该信息文本。

参见

组态报警输出概述 (页 1129)

WinCC 中的报警系统 (页 1106)

使用消息类别组态报警确认 (页 1136)

8.4.2.2 组态报警

8.4.2.2 报警组态概述

组态报警的步骤

在 WinCC 中组态报警包括下列步骤：

1. 编辑和创建报警类别
使用报警类别定义在运行系统中如何显示报警并定义报警的确认模型。
2. 创建变量
 - 在“变量”编辑器中创建项目的变量。
 - 可为变量定义限值。如果超过这些限值，则触发报警。
 - 如果需要，也可以在“变量”编辑器中创建报警。
3. 创建离散量报警和模拟量报警
 - 可创建离散量报警和模拟量报警，并为它们分配要监视的变量、报警类别、报警组和其它属性。
 - 也可以将系统函数或脚本分配给报警事件。
4. 输出组态的报警
要输出组态的报警，在“画面”编辑器中组态在运行系统中输出报警的显示和操作人员控制对象。

其它组态任务

视项目要求而定，组态报警可能需要其它任务：

1. 创建报警组
根据报警之间的关联将项目的报警分配给报警组，如按问题原因（电源故障）或错误来源（电机 1）分组。
2. 组态 Loop-In-Alarm
组态了 Loop-In-Alarm，以便切换到包含已接收报警相关信息的画面。

参见

创建报警类别 (页 1118)
组态报警组 (页 1119)
WinCC 中的报警系统 (页 1106)
在“变量”编辑器中组态离散量报警 (页 1127)
组态离散量报警 (页 1119)
组态模拟量报警 (页 1121)
将输出域添加到报警文本 (页 1124)
设置报警文本格式 (页 1125)
组态报警循环 (页 1125)

8.4.2.2 创建报警类别

简介

在“报警”编辑器的“报警类别”选项卡中创建报警类别。默认情况下为每个项目提供四个预定义的报警类别。可以根据需要创建更多自定义报警类别。最多可以创建 32 个报警类别。

要求

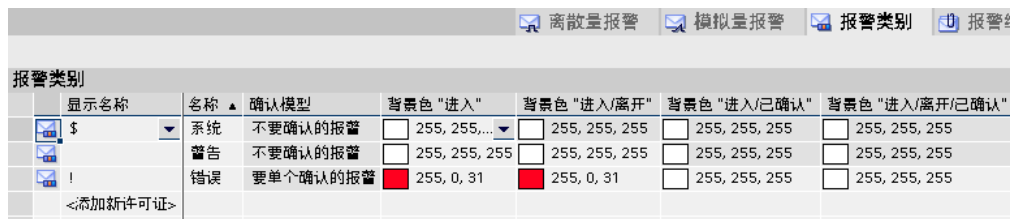
- 已创建项目。
- “报警”编辑器打开。
- 打开巡视窗口。

步骤

要创建报警类别，请按如下步骤进行操作：

1. 单击“报警类别”选项卡。

显示四个预定义的报警类别和现有自定义报警类别。预定义报警类别表显示在下面：



显示名称	名称	确认模型	背景色 "进入"	背景色 "进入/离开"	背景色 "进入/已确认"	背景色 "进入/离开/已确认"
\$	系统	不要确认的报警	255, 255,...	255, 255, 255	255, 255, 255	255, 255, 255
	警告	不要确认的报警	255, 255, 255	255, 255, 255	255, 255, 255	255, 255, 255
!	错误	要单个确认的报警	255, 0, 31	255, 0, 31	255, 255, 255	255, 255, 255
<添加新许可证>						

2. 双击工作区中表的第一个空行。

创建新报警类别。将固定的 ID 自动分配给每个新报警类别。

新报警类别的属性显示在巡视窗口中。

3. 在巡视窗口中的“常规”下组态报警类别。输入“名称”和“显示名称”。根据 HMI 设备的不同，可对记录或自动发送电子邮件进行设置。
4. 在巡视窗口中的“确认”下定义此报警类别的确认模型。
5. 在巡视窗口中的“状态”下，更改默认文本。
此文本指示在运行系统中的报警状态。
6. 在巡视窗口中的“颜色”下，更改默认颜色。根据 HMI 设备的不同，还可以更改闪烁特征。

这些设置定义在运行系统中如何显示来自此报警类别的报警。

注意

要在运行系统中显示以颜色编码的报警类别，必须在“项目树”>“运行系统设置”>“报警设置”下选择“使用报警类别颜色”选项。在 WinCC 的新项目中该选项已选择。

参见

[报警组态概述 \(页 1117\)](#)

8.4.2.2 组态报警组

简介

在“报警”编辑器的“报警组”选项卡上创建报警组。报警组是单个报警的汇编。通过关联（如问题原因或错误来源）将报警分配到报警组中。如果在运行系统中确认来自此报警组的一个报警，则将自动确认此报警组中的所有其它报警。

要求

- 已创建项目。
- “报警”编辑器打开。
- 打开巡视窗口。

创建新报警组

1. 单击“报警组”选项卡。
显示现有报警组。



2. 双击工作区中表的第一个空行。
创建了一个新的报警组。新报警组的属性显示在巡视窗口中。
3. 输入报警组的名称。
4. 保存项目。

结果

已创建报警组。要在运行系统中一起确认多个报警，将这些关联的报警分配给一个报警组。

参见

报警组态概述 (页 1117)

8.4.2.2 组态离散量报警

简介

离散量报警指示工厂中的状态变化，离散量报警由 PLC 触发。例如，它们指示阀门打开或关闭。

下段内容描述在“报警”编辑器中的组态过程。另外，还可以在“变量”编辑器中组态离散量报警。

要求

- “报警”编辑器打开。
- 打开巡视窗口。
- 已创建所需的报警类别和报警组。

步骤

要组态离散量报警，请按如下步骤进行操作：

1. 单击“离散量报警”选项卡。
2. 要创建新离散量报警，在表的第一个空行中双击。
创建了一个新的离散量报警。
3. 在巡视窗口中的“常规”下组态报警。
 - 如果需要，更改报警的对象名称。
 - 如果需要，选择报警类别和报警组。
 - 如果需要，输入报警文本并在报警文本中插入输出域。
4. 在巡视窗口中，在“属性”>“触发源”下选择触发报警的变量和位。 请注意下列信息：
 - 使用数据类型“UShort”或“Short”。
 - 触发变量位仅用于报警。
 - 不要将触发变量用于别的方面。
 - 如果要通过 PLC 确认报警，将此变量用作 PLC 确认变量。

注意

如果指定的触发位已用作其它操作的触发源，则在下次编译前，系统不检查此多重分配。

注意事项

在指定位时，请注意对所使用的 PLC 中的位进行计数的方法。有关详细信息，请参阅 PLC 在线帮助中的“通信”部分。

注意

如果选择中的对象尚不存在，则直接在对象列表中创建它，并在以后更改其属性。

结果

创建了离散量报警。

状态相关报警文本

将离散量报警的值与文本列表关联，以便为通过离散量报警显示的两个状态输出相关报警文本。

离散量报警的附加设置

将离散量报警分配给报警类别

要将报警分配给报警组，请按如下步骤进行操作：

1. 选择离散量报警。
2. 在巡视窗口中的“常规”下选择报警组。

创建信息文本

要将报警分配给信息文本，请执行以下步骤：

1. 选择离散量报警。
2. 在巡视窗口中选择“属性”>“信息文本”，并输入所需文本。

组态事件驱动的任务

要组态事件驱动的任务，请按如下步骤进行操作：

1. 选择离散量报警。
2. 在巡视窗口中，选择“事件”。
3. 为所需的事件组态函数列表。

参见

报警组态概述 (页 1117)

8.4.2.2 组态模拟量报警

简介

模拟量报警指示超出限制值。例如，如果电机速度下降到某一值以下，则触发模拟量报警。

要求

- “报警”编辑器打开。
- 打开巡视窗口。
- 已创建所需的报警类别和报警组。

步骤

要组态模拟量报警，请按如下步骤进行操作：




1. 单击“模拟量报警”选项卡。
2. 要创建新模拟量报警，在表的第一个空行中双击。
创建了一个新的模拟量报警。
3. 在巡视窗口中的“常规”下组态报警。
 - 如果需要，更改报警的对象名称。

- 如果需要，选择报警类别和报警组。



4. 输入报警文本。
5. 您可以逐个字符地处理报警文本的格式，并在其中插入变量值或文本列表的输出域。
6. 在巡视窗口中，在“属性”>“触发源”下选择触发报警的变量。不要将触发变量用于别的方面。

组态模拟量报警的限值

1. 选择模拟量报警。
2. 要使用常数作为限值，在巡视窗口中的“触发源”>“值”下单击左按钮  并选择“字符串”选项。
3. 输入所需的限值。
4. 要使用变量作为限值，在巡视窗口中的“触发源”>“值”下单击左按钮  并选择“变量”选项。
右按钮  出现。
5. 使用该按钮选择所要使用的变量。
“同步”指示这些限值是否与在“变量”编辑器中为此报警组态的限值同步。
6. 选择触发模式：
 - “上限”：超过此限值时触发报警。
 - “下限”：低于此限值时触发报警。

注意

如果选择中包括的对象尚不存在，则在对象列表中创建它，并在以后更改其属性。

模拟量报警的可选设置

设置延迟时间

要设置延迟时间，请按如下步骤进行操作：

1. 在巡视窗口中，选择“属性”>“触发源”。
2. 在“延迟”下输入时间段。只有当延迟时间过去之后触发条件仍存在时才触发报警。

设置滞后

注意

如果过程值围绕限值波动，则会多次触发与此故障关联的报警。在该情形下，请组态迟滞或延迟时间。

要输入滞后，请按如下步骤进行操作：

1. 在巡视窗口中，选择“属性”>“触发源”。
2. 在“滞后”>“模式”下，选择考虑滞后情况的报警状态的变化。
3. 在“滞后”>“值”下输入常数值。
4. 要将滞后值定义为限值的百分比，启用“%”复选框。

自动报告

1. 选择模拟量报警。
2. 在巡视窗口中，选择“属性”>“过程”，并启用“报告”。如果在报警设置下启用“报告”，此报警将记录在报警报告中。

注意

精简面板

报告不可用于精简面板。

创建信息文本

1. 选择模拟量报警。
2. 在巡视窗口中选择“属性”>“信息文本”，并输入所需文本。

组态事件驱动的任务

1. 选择模拟量报警。
2. 在巡视窗口中选择“事件”，并为所需事件组态新函数列表。

参见

报警组态概述 (页 1117)

8.4.2.2 将输出域添加到报警文本

简介

在 WinCC 中，可将选择域添加到报警文本中。这些域显示文本列表或变量的内容。

要求

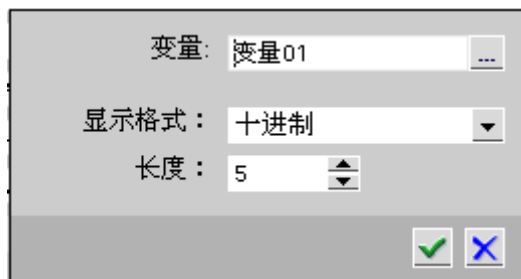
- “报警”编辑器打开。
- 已经创建了报警。

步骤

要将输出域添加到报警文本，请按如下步骤进行操作：

1. 选择要编辑的报警。
2. 将光标放在报警文本中的所需位置。
3. 要直接输出变量值，请从快捷菜单中选择“插入变量输出域”命令。

“插入变量输出域”对话框随即打开。



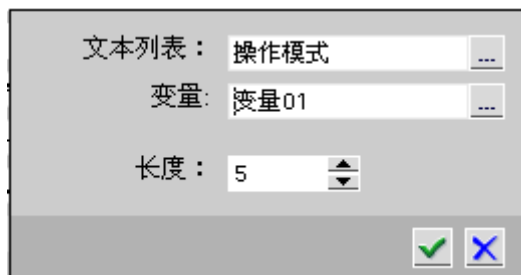
4. 从“变量”域中选择所需的变量。从“显示格式”下选择格式，从“长度”下选择输出域的文本长度。
5. 单击



接受输入。

6. 要使用文本列表以符号输出变量值，请从快捷菜单中选择“输入文本列表输出域”。

“输入文本列表输出域”对话框随即打开。



7. 从“文本列表”下选择所需的文本列表，从“变量”下选择所需的变量。在“长度”下定义输出域的长度。
8. 单击



接受输入。

注意

报警文本中变量输出域的顺序取决于语言。运行系统语言的顺序用于在 CSV 日志中记录报警。更改以某一种语言表示的输出域的变量，可导致所修改的输出域出现在以所有其它语言表示的报警文本的结尾。这更改了日志中输出域的顺序。

精简面板

记录不可用于精简面板。

参见

报警组态概述 (页 1117)

8.4.2.2 设置报警文本格式

要求

- “报警”编辑器打开。
- 已创建离散量或模拟量报警。

步骤

要设置报警文本格式，执行如下操作：

1. 在巡视窗口中选择要为其设置格式的报警文本部分。
2. 从报警文本的快捷菜单中选择“格式”。
3. 激活该报警文本所需的格式。

结果

在运行系统中以所选择的格式显示报警文本。

参见

报警组态概述 (页 1117)

8.4.2.2 组态报警循环

简介

组态了 Loop-In-Alarm，以便切换到包含已接收报警相关信息的画面。

要求

- 已创建项目。
- 已创建要由 Loop-In-Alarm 调用的画面。
- “报警”编辑器打开。

步骤

要为报警组态 Loop-In-Alarm，请按如下步骤进行操作：

1. 单击包含要为其组态 Loop-In-Alarm 的报警的选项卡。
2. 选择报警。
3. 在巡视窗口中，选择“事件”>“Loop-In-Alarm”。
4. 选择“ActivateScreen”系统函数。



5. 选择 Loop-In-Alarm 要调用的画面作为参数。

注意

如果要为带有“报警行”表单的报警视图组态 Loop-In-Alarm，使用“EditAlarm”系统函数触发“Loop-In-Alarm”事件。报警行没有按钮。

结果

如果报警显示在运行系统中，并且您单击报警视图的 Loop-In-Alarm 按钮，则会打开一个画面，其中包含有关出现的错误的信息。

参见

[报警组态概述 \(页 1117\)](#)

8.4.2.2 “变量”编辑器中的报警

8.4.2.2 在“变量”编辑器中组态离散量报警

简介

在 WinCC 中，可在“变量”编辑器中与触发变量一起创建和编辑离散量和模拟量报警。

注意

如果在“变量”编辑器中进行删除、移动或复制，则更改在“报警”编辑器中也生效。

要求

“变量”编辑器中的“HMI 报警”选项卡已打开。

步骤

要在“变量”编辑器中组态离散量报警，请按如下步骤进行操作：

1. 单击工作区顶部表格的第一行。
创建变量。
2. 根据需要，组态数据类型为 ULong 或 UShort 的内部或外部变量。
3. 选择变量，并单击工作区底部的“离散量报警”选项卡。
4. 双击表格的第一个空行。
为变量创建新离散量报警。如果您选择了错误的数据类型，则将在离散量报警中选择该变量。
5. 在巡视窗口中的“属性”>“文本”下输入报警文本。
6. 也可以将输出域添加到报警文本。
7. 选择报警类别。
8. 在巡视窗口中，在“属性”>“触发源”下选择触发离散量报警的变量的触发位。
9. 要创建用于监视变量的其它离散量报警，重复上述组态步骤。

注意

只能使用一个报警过程监视一个变量。因此，应为变量创建模拟量报警或离散量报警。

结果

在“变量”编辑器中创建组态的离散量报警，并在“报警”和“变量”编辑器中显示它们。

参见

在“变量”编辑器中组态模拟量报警 (页 1128)

报警组态概述 (页 1117)

8.4.2.2 在“变量”编辑器中组态模拟量报警

简介

在 WinCC 中，可在“变量”编辑器中与触发变量一起创建和编辑离散量和模拟量报警。最多可为变量创建四个限值。使用模拟量报警监视这些限值。

注意

可用于特定设备

最多可为精简面板组态两个限值。

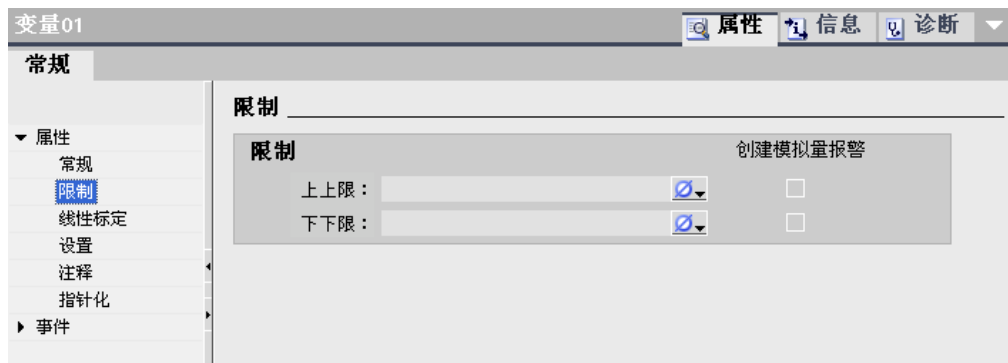
要求

“变量”编辑器中的“HMI 报警”选项卡已打开。

步骤

要在“变量”编辑器中组态模拟量报警，请按如下步骤进行操作：

1. 单击工作区顶部表格的第一行。
创建变量。
2. 根据需要组态内部或外部变量。
3. 在巡视窗口中单击“属性”>“列”。



4. 选择所需限值，即“常数”或“HMI 变量”。如果选择“HMI 变量”，则打开对象列表。选择希望使用的变量。
5. 启用此限值的“创建模拟量报警”。
6. 对于所需要的限值，重复最后两个组态步骤。
7. 单击工作区底部的“模拟量报警”选项卡。
为所选择的限值创建模拟量报警。

8. 选择模拟量报警。
9. 在巡视窗口中的“属性”下输入报警文本。
也可以将输出域添加到报警文本。
- 1 选择报警类别。默认报警类别是“错误”。
- 0.

注意

请注意以下要点：

- 只能使用一个报警过程监视一个变量。因此，应为变量创建模拟量报警或离散量报警。
 - 如果在“变量”编辑器中进行删除、移动或复制，则更改在“报警”编辑器中也生效。
 - 在变量的巡视窗口中创建模拟量报警时，限值与“报警”编辑器同步。
-

结果

在“变量”编辑器中创建组态的模拟量报警，并在“报警”和“变量”编辑器中显示它们。

参见

在“变量”编辑器中组态离散量报警 (页 1127)

8.4.2.3 组态报警输出

8.4.2.3 组态报警输出概述

组态报警输出的步骤

在 WinCC 中按以下步骤组态报警输出：

1. 创建报警视图

使用“画面”编辑器中的显示和控制对象在运行系统中显示报警。也可为已记录的报警组态报警视图。

2. 组态确认

在“画面”编辑器中，可设置将触发确认的操作员操作。

3. 组态报告

可以在“报告”编辑器中创建报告以打印运行系统中的报警。在“调度程序”、“画面”、“报警”或“变量”编辑器中，定义触发报表打印的时间和方式。

注意

精简面板

报告和记录不可用于精简面板。

其它组态任务

根据项目要求，可能需要执行其它任务来组态报警视图：

1. 设置权限

为确保只有获得授权的操作员才能处理报警，可为报警视图及 HMI 设备上的功能键分配权限。

2. 组态报警视图的过滤

可在“画面”编辑器中组态运行系统中的报警过滤。也可以组态仅显示选定报警的报警视图。

3. 组态操作员输入报警

在“画面”编辑器中，组态 HMI 设备的操作员控件上的操作员输入报警。将输出预组态的操作员输入报警用于操作员操作。操作员操作包括报警显示的确认、禁用和抑制。

参见

报警组件和属性 (页 1115)

显示报警 (页 1130)

组态报警视图 (页 1131)

组态报警窗口 (页 1134)

组态报警指示器 (页 1135)

8.4.2.3 显示报警

在 HMI 设备上显示报警的选项

WinCC 提供了下列选项用于在 HMI 设备上显示报警：

- 报警视图

报警视图在画面中组态。根据已组态的大小，可以同时显示多个报警。可以组态具有不同内容的多个报警视图。

- 报警窗口

报警窗口在“全局画面”编辑器中组态。根据已组态的大小，报警窗口可同时显示多个报警。报警窗口的关闭和重新打开均可通过事件触发。要在组态期间隐藏报警窗口，可在其自身层级上创建一个报警窗口。

其它信号

- 报警指示器

报警指示器是可组态的图标。当报警到达时，报警指示器将显示在 HMI 设备上。在“全局元素”编辑器中组态报警指示器。

报警指示器具有两种状态：

- 闪烁：至少存在一条未确认的待决报警。
- 静态：报警已确认，但其中至少有一条尚未取消激活。

报警指示器还将根据 HMI 设备显示未决报警数。

- 电子邮件通知

当带有特定报警类别的报警到达时，若要通知除操作员之外的人员（如工程师），可将该报警类别分配给电子邮件地址。

注意
精简面板

电子邮件通知不可用于精简面板。

- 系统函数
可以为报警相关的事件组态一个函数列表。事件发生时，必须在运行系统中执行这些函数。
在 WinCC 中使用针对报警的系统函数，而不是通过工具栏，来控制报警视图或报警窗口。

在运行系统中显示预定义的报警类别

下表显示了在报警视图中用于显示预定义报警类别的符号：

报警类别	显示的图标
诊断报警	S7
错误	!
系统	\$
警告	<无符号>

注意
精简面板

诊断报警不可用于精简面板。

参见

组态报警输出概述 (页 1129)

8.4.2.3 组态报警视图

简介

当前报警或报警日志中的报警将显示在“报警视图”显示和控制对象中。报警视图显示所有报警类别的报警。

注意 精简面板

报警日志不可用于精简面板。

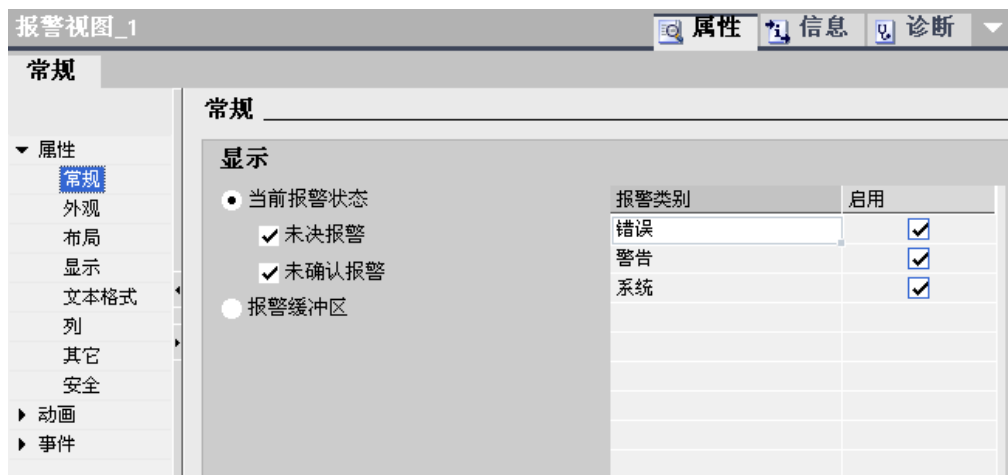
要求

- 画面在“画面”编辑器中打开。
- 打开“工具”任务卡。
- 项目树已打开。

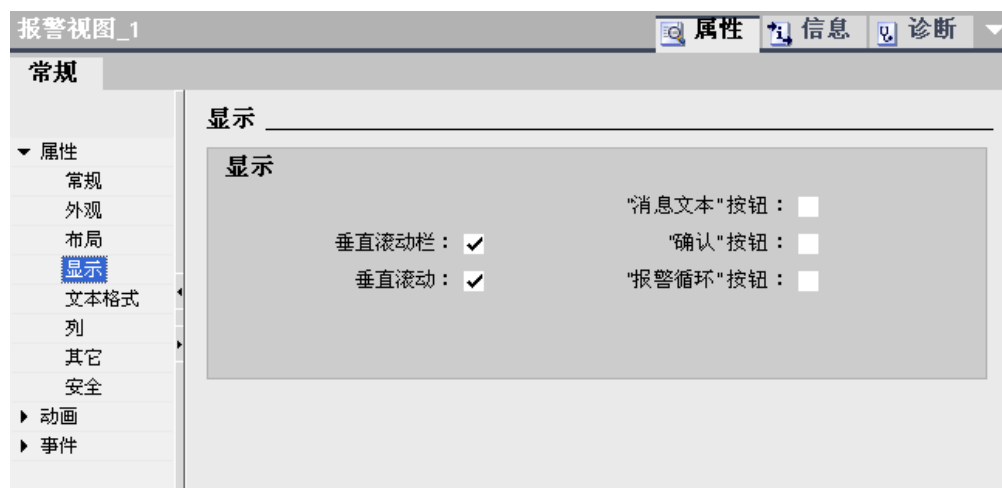
为报警视图组态报警

要指定将显示在报警视图中的报警，请按如下步骤进行操作：

1. 在“工具”任务卡的“控件”下，将报警视图添加到画面。
2. 选择报警视图。
3. 在巡视窗口的“常规”>“显示”下，确定要显示报警类别、报警事件还是报警日志。



4. 在“报警类别”下，启用要在报警视图中显示的报警类别。
5. 在巡视窗口中单击“显示”。
6. 在“用于显示区的控制变量”下，定义返回报警显示起始日期的变量。



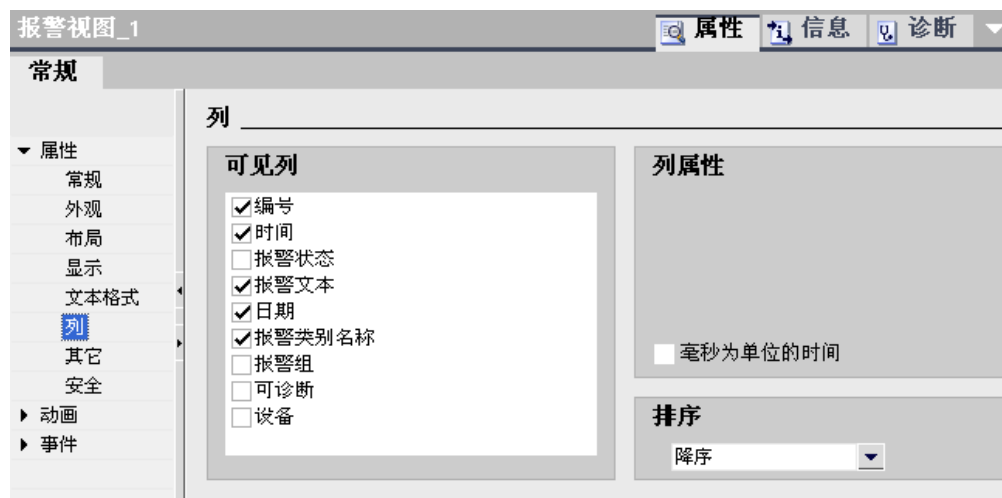
注意
精简面板

“用于显示区的控制变量”属性不可用于精简面板。

组态报警视图的布局

要指定报警如何在报警视图中显示，请按如下步骤进行操作。

1. 在项目树中单击“布局”。
2. 在“每个报警的行数”下，指定每个报警将要显示的行数。
3. 在“显示”区域中，启用可用于 HMI 设备的控制元素。
4. 在巡视窗口中单击“属性”>“列”。



5. 在“可见列”下，选择将在报警视图中输出的列。
6. 在“排序”下，选择报警的排序顺序。
7. 在“属性列”下，定义列的属性。

注意

在报警视图的快捷菜单中选择“编辑”命令，以激活报警视图。在激活模式下，可以设置列宽和位置。要启用报警视图，将缩放因子设置为 100 %。

结果

不同报警组的报警于运行时在报警视图中输出。

参见

组态报警输出概述 (页 1129)

8.4.2.3 组态报警窗口

简介

报警窗口显示当前报警。在“全局画面”编辑器中组态并打开报警窗口（与当前画面无关）。即使报警处于未决状态且已显示，仍然可以使用 HMI 设备。报警窗口的显示和组态与报警视图类似。

要在组态期间隐藏报警窗口，可在其自身层级上创建一个报警窗口。

要求

- “全局画面”编辑器已打开。
- 已显示“工具”任务卡。
- 打开巡视窗口。

步骤

组态报警窗口的步骤如下：

1. 在“工具”任务卡中，将“报警窗口”对象从“控件”组插入到模板中。
2. 报警窗口的组态方式与报警视图相同。
3. 在巡视窗口的“模式”>“窗口”下，选择在运行系统中报警窗口如何反应以及如何操作。
4. 启用“模式”>“已启用”。

如果已启用此选项，则在运行系统中进行画面切换后，报警窗口将保留原焦点。此选项很重要，因为系统不支持使用 <Ctrl+TAB> 在画面和不同窗口之间进行来回切换。

结果

在运行时，所选报警类别的报警将显示在报警窗口中。

参见

组态报警输出概述 (页 1129)

8.4.2.3 组态报警指示器

简介

报警指示器使用警告三角来表示报警处于未决状态或要求确认。如果产生组态的报警类别中的报警，则显示报警指示器。

报警指示器具有两种状态：

- 闪烁：至少存在一条未确认的待决报警。
- 静态：至少有一个已确认的报警尚未离开。

当用户操作报警指示器时，您可以组态运行系统来打开报警窗口。

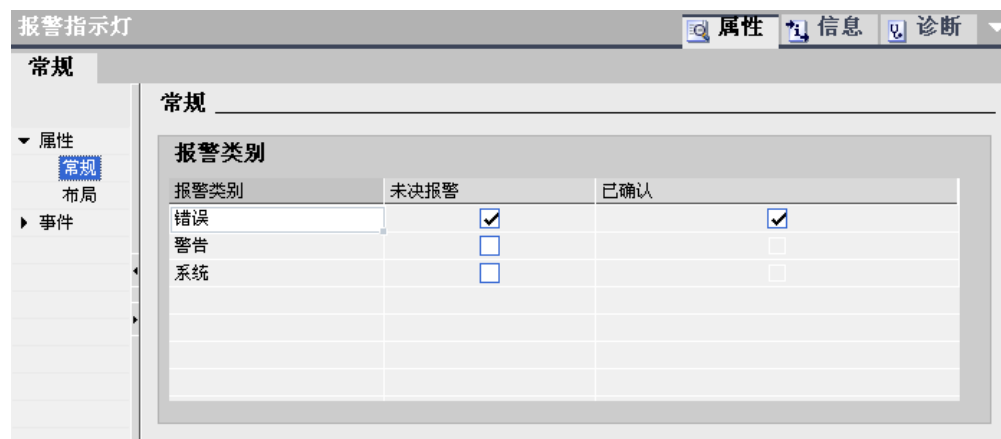
要求

- “全局画面”编辑器已打开。
- 打开“工具”任务卡。
- 打开巡视窗口。

步骤

组态报警指示器的步骤如下：

1. 在“工具”任务卡中，将“报警指示器”对象从“控件”组拖动到工作区中。
2. 选择报警指示器。
3. 在巡视窗口的“常规”>“报警类别”下，选择将为其显示报警指示器的报警类别。



您自己创建的报警类别也会显示。

输入报警指示器是否要显示未决的报警和/或那些等待确认的报警。

4. 为报警指示器的事件组态系统函数“ShowAlarmWindow”。

注意

如果在画面或模板中组态了固定窗口，则不要将报警窗口和报警指示器放在固定窗口附近。否则，在运行系统中将不显示报警窗口和报警指示器。在“全局元素”编辑器中不会看到固定窗口。

结果

如果在运行系统中所选报警类别中的报警处于未决状态或需要确认，则显示报警指示器。当用户操作报警指示器时，报警窗口将打开。

参见

组态报警输出概述 (页 1129)

8.4.2.4 确认报警

8.4.2.4 使用消息类别组态报警确认

简介

使用报警类别定义是否应确认所有报警类别的报警，以及如何确认这些报警。将报警分配到一个报警类别时，定义是否必须确认报警。

要求

- “报警”编辑器打开。
- 已创建所需报警类别。
- 已创建所需报警。

为报警类别选择确认模型

已为预定义报警类别设置了确认模型。可以仅为自定义报警类别设置确认模型。为此，请按如下步骤进行操作：

1. 在“报警”编辑器中，单击“报警类别”选项卡，然后选择所需报警类别。
2. 在巡视窗口的“确认”>“设置”下选择所需的确认模型。

将报警分配到需要确认的报警类别

将报警分配到需要确认的报警类别的步骤如下。

1. 在“报警”编辑器中，单击所需报警类型的选项卡，然后选择所需报警。
2. 在巡视窗口的“常规”下，选择该报警所从属的报警类别。

参见

报警组件和属性 (页 1115)
组态报警确认 (页 1137)

8.4.2.4 组态报警确认

简介

始终使用报警类别为报警指定确认要求。以下选项还可以用于确认：

- 通过系统函数确认报警
- 由 PLC 确认离散量报警
- 将报警确认发送到 PLC
- 组态用于确认报警的按钮

要求

- “报警”编辑器打开。
- 已创建所需报警类别。
- 已创建所需报警。

通过系统函数确认报警

要通过系统函数确认报警，请按如下步骤进行操作：

1. 在“报警”编辑器中选择所需报警。
2. 在巡视窗口的“事件”下单击“进入”事件。
函数列表打开。
3. 在“函数列表”对话框中为“进入”事件设置“SetBit”系统函数。
4. 输入“BOOL”数据类型的确认变量作为“变量（进/出）”参数。
更多信息，请参考“使用系统函数”部分。

由 PLC 确认离散量报警

1. 在“报警”编辑器中，单击“离散量报警”选项卡并选择所需离散量报警。
2. 在巡视窗口中，选择“属性”>“确认”。



3. 在“PLC”下，选择 PLC 将用来确认报警的变量和位。

将报警确认发送到 PLC

要将报警确认发送给 PLC，请执行以下步骤：

1. 在“报警”编辑器中，单击“离散量报警”选项卡并选择所需离散量报警。

2. 在巡视窗口中，选择“属性”>“确认”。
3. 在“HMI”下，选择要被报警确认置位的变量和位。

组态用于确认报警的按钮

要组态用于确认报警的按钮，请按如下步骤进行操作：

1. 选择按钮或功能键。
2. 为“单击”事件设置“AlarmViewAcknowledgeAlarm”系统函数。

更多详细信息，请参见“使用画面”部分。

参见

使用消息类别组态报警确认 (页 1136)

8.4.3 操作运行系统中的报警

8.4.3.1 运行系统的报警

报警

在 HMI 设备上，报警指示系统、过程或 HMI 设备本身所发生的事件或状态。状态接收后将报告。

报警可触发下列报警事件：

- 进入
- 离开
- 确认
- 报警循环

组态工程师将定义哪些报警必须由用户进行确认。

报警可能包含以下信息：

- 日期
- 时间
- 报警文本
- 故障位置
- 状态
- 报警类别
- 报警编号
- 报警组

报警类别

报警可分为以下几类。

- 警告
此类报警通常指示设备状态，例如“电机已启动”。此类报警不需要进行确认。
- 错误
该类报警必须始终进行确认。错误报警通常指示设备的关键错误，例如“电机温度过高”。
- 系统
系统报警指示 HMI 设备本身的状态或事件。
系统报警提供关于诸如发生操作错误或通信故障等的信息。
- 自定义报警类别
该报警类别的属性必须在组态中定义。

报警缓冲区

报警事件将存储在断电易失的内部缓冲区中。该报警缓冲区的大小取决于 HMI 设备类型。

报警视图

报警视图显示了在报警缓冲区中选择的报警或报警事件。报警事件是否需要确认由组态指定。

报警窗口

报警窗口显示所有未决的报警或特定报警类别的等待确认的报警。报警窗口将在新报警出现时显示。

可以组态报警显示的顺序。您可以选择按发生时间的升序或降序显示报警。报警窗口也可以设置为显示故障的确切地点，以及报警事件的日期和时间。通过组态，可以过滤显示内容，仅显示包含指定字符串的报警。

报警指示器

报警指示器是一个图标，当激活指定报警类别的报警时，该图标便会显示在屏幕上。

报警指示器可采用以下两种状态之一：

- 闪烁：至少存在一条未确认的待决报警。
- 静态：报警已确认，但其中至少有一条尚未离开。显示的数字指示未决报警的数目。

参见

WinCC 中的报警系统 (页 1106)
运行系统中的简单报警视图、简单报警窗口 (页 1140)
运行系统中的报警指示器 (页 1141)
确认报警 (页 1142)

8.4.3.2 运行系统中的简单报警视图、简单报警窗口

应用

简单报警视图显示了在报警缓冲区或报警日志中选择的报警或报警事件。简单报警窗口的布局和操作与简单报警视图中的保持一致。

注意

无法使用脚本动态操作“单个报警视图”对象。

例如，在工程系统中，可以在巡视窗口的“动画”组中动态控制对象的可见性。在运行系统中，“简单报警视图”不支持动画。如果已组态某个动画，并希望执行某项操作（例如执行项目一致性检查），则系统会在输出窗口中输出一条错误消息。

注意

精简面板

记录不可用于精简面板。



布局

根据组态的不同，在报警视图中，系统使用不同的列来显示关于报警或报警事件的信息。

为区分不同的报警类别，报警视图中的第一列包含一个图标：

符号	报警类别
!	错误
空	警告
取决于组态	自定义报警类别
\$	系统










操作

根据组态方式的不同，可使用报警视图执行以下操作：

- 确认报警
- 编辑报警

控制元素

按钮具有下列功能：

按钮	功能
	确认报警。
	编辑报警。
	显示报警的信息文本。
	在单独的窗口（即报警文本窗口）中显示所选报警的完整报警文本。 在报警文本窗口中，可以查看其所需空间超出“报警”视图中可用空间的报警文本。使用  按钮关闭报警文本窗口。
	向上滚动一个报警。
	在报警视图中向上滚动一页。
	在报警视图中向下滚动一页。
	向下滚动一个报警。

控制元素的格式

在操作面板 OP 73micro 和 TP 177micro 上，简单报警视图具有一个按钮，用于在单独的窗口中显示消息文本。该按钮在工程系统中简单报警的组态期间并不显示。

简单报警视图中按钮的显示取决于所组态的大小。因此应该检查 HMI 设备上是否存在所有需要的按钮。

参见

运行系统的报警 (页 1138)

8.4.3.3 运行系统中的报警指示器

应用

如果指定报警类别中的报警处于未决状态或需要确认，则显示报警指示器。



布局

报警指示器的状态可以为以下两种之一：

- 闪烁：至少存在一条未确认的待决报警。
- 静态：报警已确认，但其中至少有一条尚未取消激活。数字指示排队等候的报警个数。

操作

根据组态的不同，操作报警指示器时可能会打开报警窗口。符号库中的图标只能通过鼠标或触摸屏操作。

注意 精简面板

鼠标操作不可用于精简面板。

参见

运行系统的报警 (页 1138)

8.4.3.4 确认报警

简介

在运行系统中，可以根据项目组态设置确认报警。可以通过以下方式确认报警：

- 使用显示和控制对象按钮
- 使用 HMI 设备上的“ACK”按键
- 使用单独组态的功能键或按钮

如果为单个控件组态了操作员权限，则报警只能由获得授权的用户确认。

要在运行系统中自动确认报警，应使用系统函数和脚本，同时选中“通过 PLC 确认”选项。

注意 精简面板

脚本不可用于精简面板。

确认类别

在运行系统中，可分别确认报警或同时确认多个报警。它们区别如下：



- 单个确认
使用按钮或功能键确认一个报警。
- 确认报警组
使用按钮或功能键确认一个报警组的所有报警。

要求

- 报警显示在 HMI 设备上。

步骤

要确认报警，请执行以下步骤：

1. 选择报警。
2. 如果正在使用报警视图或报警窗口，则单击
按钮。
3. 如果正在使用简单报警视图或简单报警窗口，则单击
按钮。
4. 按 HMI 设备上的“ACK”按钮以确认报警行中的报警。

注意 精简面板

只有简单报警视图和简单报警窗口可用于精简面板。

结果

报警状态变为“已确认”。如果触发报警的条件不再适用，则报警状态还会变为“离开”，且不再显示在 HMI 设备上。

参见

运行系统的报警 (页 1138)

8.4.4 参考

8.4.4.1 报警的系统函数和事件

系统函数

系统函数是预定义的函数，用户即使没有任何编程知识，也可使用这些函数在运行系统中执行许多任务。系统函数可在函数列表或脚本中使用。

注意

精简面板

脚本不可用于精简面板。

下表显示了所有可用于显示和编辑报警的系统函数。

系统函数	作用
EditAlarm	为所有选定的报警触发 Loop-In-Alarm 事件。
ClearAlarmBuffer	删除 HMI 设备报警缓冲区中的报警。
ClearAlarmBufferProtoolLegacy	与“ClearAlarmBuffer”类似的函数。为确保兼容性，该系统函数仍然保留，以便使用以前的 ProTool 编号方式。
AlarmViewEditAlarm	为在指定报警视图中选定的所有报警触发 Loop-In-Alarm 事件。
AlarmViewAcknowledgeAlarm	确认在指定报警视图中选定的报警。
AlarmViewShowOperatorNotes	显示在指定报警画面中选定报警的已组态信息文本。
AcknowledgeAlarm	确认选择的所有报警。
SetAlarmReportMode	确定是否将报警自动报告到打印机上。
ShowAlarmWindow	隐藏或显示 HMI 设备上的报警窗口。
ShowSystemAlarm	将已传递参数的值显示为 HMI 设备上的系统报警。

注意

精简面板

“SetAlarmReportMode”和“ShowSystemAlarm”系统函数不可用于精简面板。

有关系统函数的详细信息，请参见“可视化过程 > 使用系统函数 > 参考 > 系统函数”部分。

报警及其显示和控制对象的事件

在运行系统中，为响应报警及其显示和控制对象，将发生以下事件。可以为每个事件组态一个函数列表。

对象	可组态的事件
报警	进入 离开 确认 Loop-In-Alarm
报警视图	启用 禁用 建立 报警编号已更改
报警指示器	单击 闪烁时单击

注意
精简面板

“建立”和“报警编号已更改”事件不可用于精简面板。

有关事件的详细信息，请参见“可视化过程 > 使用系统函数 > 参考 > 系统函数 > 事件”部分。

参见

WinCC 中的报警系统 (页 1106)
系统报警 (页 1145)

8.4.4.2 系统报警

简介

HMI 设备上的系统报警提供关于 HMI 设备和 PLC 内部状态的信息。

以下概括说明了系统报警产生的原因以及排除出错原因的方法。

注意
可用于特定 HMI 设备

本节中所介绍的一些系统报警适用于每台 HMI 设备 (基于其功能范围)。

注意

系统报警在报警视图中输出。系统报警将以当前在 HMI 设备上所设置的语言输出。

系统报警参数

系统报警可能包含加密参数。故障排除时各参数是相关的，因为它们都对运行系统软件的源代码进行引用。这些参数在“错误代码：”文本之后输出。

系统报警的含义

下表包含了所有系统报警。系统报警分为几个系列：

表格 8-2 10000 - 打印机报警

编号	效果/原因	解决方法
10000	由于某个未知原因，打印作业无法启动或取消。没有正确设置打印机。或：无访问网络打印机的权限。数据传输过程中断电。	检查打印机设置、电缆连接和电源。 再次设置打印机。获得网络打印机使用权限。 如果故障仍未排除，请联系我们的热线！
10001	尚未安装任何打印机或尚未设置默认打印机。	请安装打印机和/或将其选择为默认打印机。
10002	用于打印的图形缓冲区已溢出。至多可缓冲两个图像。	请等待一个打印作业完成，然后再触发其它打印作业。
10003	现在可再次对图像进行缓冲。	--
10004	在文本模式下，行打印的缓冲区溢出（例如，报警）。至多可缓冲 1000 行。	请等待一个打印作业完成，然后再触发其它打印作业。
10005	文本行现在可再次缓冲。	--
10006	Windows 打印系统报错。请查阅输出文本和错误 ID 号，以确定可能的原因。什么也没有打印或打印出现故障。	重复该操作。

表格 8-3 20000 - 全局脚本报警

编号	效果/原因	解决方法
20010	指定脚本行中发生错误。脚本的执行因此而中止。请注意在这之前可能已发生的系统报警。	在组态中选择所指定的脚本行。请确保使用允许的变量类型。使用系统函数检查参数编号和类型是否正确。
20011	由指定脚本调用的脚本产生了错误。因此中止执行被调用的脚本。请注意在这之前可能已发生的系统报警。	在组态中，选择已经由指定脚本直接或间接调用的脚本。 请确保使用允许的变量类型。 使用系统函数检查参数编号和类型是否正确。
20012	项目数据不一致。因此不能生成脚本。	重新编译组态。
20013	WinCC Runtime 的脚本组件安装不正确。因此，不能执行任何脚本。	在您的 PC 上重新安装 WinCC Runtime。 使用“项目”>“生成器”>“生成”来重新生成项目并将其下载到 HMI 设备。
20014	系统函数返回一个未写入任何返回变量的值。	选择在组态中指定的脚本。 检查是否为脚本名分配了一个值。

编号	效果/原因	解决方法
20015	在短时间内接接触发的脚本太多。当排队等待处理的脚本多于 20 个时，将拒绝所有后续脚本。此时，不执行报警中所指示的脚本。	请检查是什么触发了脚本。延长时间，例如触发脚本的变量的轮询时间。

表格 8-4 30000 - IFwSetValue 的报警：SetValue()

编号	效果/原因	解决方法
30010	变量不能接受函数结果，例如当其超出取值范围时。	检查系统函数参数的数据类型。
30011	系统函数不能执行，因为在参数中给函数分配了一个无效的值或类型。	检查无效参数的参数值和数据类型。如果使用变量作为参数，则请对其值进行检查。
30012	系统函数不能执行，因为在参数中给函数分配了一个无效的值或类型。	检查无效参数的参数值和数据类型。如果使用变量作为参数，则请对其值进行检查。

表格 8-5 40000 - 线性转换报警

编号	效果/原因	解决方法
40010	无法执行系统函数，因为参数不能转换为通用的数据类型。	请检查组态中的参数类型。
40011	无法执行系统函数，因为参数不能转换为通用的数据类型。	请检查组态中的参数类型。

表格 8-6 50000 - 数据服务器报警

编号	效果/原因	解决方法
50000	HMI 接收数据速度比其处理数据速度快。因此，将在所有数据处理完毕之后才又开始接收数据。然后将重新开始数据交换。	--
50001	数据交换已经重新开始。	--

表格 8-7 60000 - Win32 函数报警

编号	效果/原因	解决方法
60000	该报警由“DisplaySystemAlarms”函数生成。要显示的文本被作为参数传送给函数。	--
60010	不能按所定义的方向复制文件，因为两个文件中有一个文件当前已打开或源/目标路径无法使用。这可能是当前的 Windows 用户没有其中某个文件的访问权限。	重新启动系统函数或检查源/目标文件的路径。使用 Windows NT/2000/XP：运行 WinCC Runtime 的用户必须具有文件的访问权限。
60011	试图将文件复制到自身。这可能是 Windows 用户没有访问其中某个文件的权限。	检查源/目标文件的路径。使用基于 NTFS 文件系统的 Windows NT/2000/XP：运行 WinCC Runtime 的用户必须具有文件的访问权限。

表格 8-8 70000 - Win32 函数报警

编号	效果/原因	解决方法
70010	应用程序不能启动，因为在指定的路径中无法找到该程序，或没有足够的存储空间。	检查所指定的路径中是否存在该应用程序，或关闭其它应用程序。
70011	不能修改系统时间。 该错误报警只在连接了区域指针“日期/时间 PLC”时才会出现。可能的原因： <ul style="list-style-type: none"> 在作业信箱中传递了无效时间。 Windows 用户没有权限修改系统时间。 如果系统报警中的第一个参数显示有数值 13，则第二个参数指示包含有错误数值的字节。	检查要设置的时间。 使用 Windows NT/2000/XP：运行 WinCC Runtime 的用户应具有更改操作系统系统时间的权限。
70012	执行选项“运行系统和操作系统”的函数“StopRuntime”时出现错误。 Windows 和 WinCC flexible Runtime 没有关闭。 可能原因之一是其它程序不能关闭。	关闭所有当前运行的程序。 然后关闭 Windows。
70013	不能修改系统时间，因为输入值无效。可能使用了不正确的分隔符。	检查要设置的时间。
70014	不能修改系统时间。可能的原因： <ul style="list-style-type: none"> 传递了一个无效时间。 Windows 用户没有权限修改系统时间。 Windows 拒绝设置请求。	检查要设置的时间。 使用 Windows NT/2000/XP：运行 WinCC Runtime 的用户具有更改操作系统系统时间的权限。
70015	不能读取系统时间，因为 Windows 拒绝读取函数。	--
70016	试图通过系统函数或作业选择画面。因为所指定的画面号不存在，所以不能执行该操作。 或者：由于没有足够的系统存储空间而不能生成画面。 或：画面被堵塞。 或：画面选择未被正确执行。	比较系统函数或作业中的画面号是否与所组态的画面号一致。 为画面分配编号。 检查画面选择的细节，以及是否禁止特定的用户使用画面。
70017	未从区域指针中读取日期/时间，因为 PLC 中所设置的地址不可用或尚未设置。	更改 PLC 中的地址或在其中设置地址。
70018	确认密码列表已经成功导入。	--
70019	确认密码列表已经成功导出。	--
70020	确认激活了报警报表。	--
70021	确认禁用报警报表。	--
70022	确认启动“导入密码列表”操作。	--
70023	确认启动“导出密码列表”操作。	--
70024	系统函数中变量值超出范围。 系统函数的计算不能完成。	检查期望的计算并进行更正。
70025	系统函数中变量值超出范围。 系统函数的计算不能完成。	检查期望的计算并进行更正。

编号	效果/原因	解决方法
70026	内部画面存储器没有存储其它画面。 不能选择其它画面。	--
70027	已经启动 RAM 文件系统的备份。	--
70028	RAM 中的文件已复制到闪存。 RAM 中的文件已复制到闪存。重启后，所保存的这些文件将被复制回 RAM 文件系统。	--
70029	RAM 文件系统的备份失败。 没有为 RAM 文件系统备份副本。	检查“控制面板”>“OP”对话框中的设置。使用“永久存储”选项卡上的“保存文件”按钮来保存 RAM 文件系统。
70030	为系统函数组态的参数有错误。 不能建立到新的 PLC 的连接。	将为系统函数组态的参数与为 PLC 组态的参数进行比较，并加以更正。
70031	在系统函数中组态的 PLC 不是 S7 PLC。 不能建立到新的 PLC 的连接。	将为系统函数组态的 S7 PLC 名称参数与为 PLC 组态的参数进行比较，并加以更正。
70032	组态为 Tab 顺序中该编号的对象在所选画面里不可用。 画面发生改变，但焦点仍设置在第一个对象上。	检查 Tab 顺序的编号，并加以更正。
70033	不能发送电子邮件，因为与 SMTP 服务器的 TCP/IP 连接不再存在。 该系统事件仅在第一次尝试后生成。当随后的电子邮件发送失败时，不再生成系统报警。在此期间，如果电子邮件成功发送，则不会再次生成报警。 WinCC Runtime 的中央电子邮件组件每隔固定间隔（1 分钟）就会尝试建立与 SMTP 服务器的连接并发送剩余的电子邮件。	检查与 SMTP 服务器的网络连接并重新连接。
70034	连接中断之后，可以重新建立与 SMTP 服务器的 TCP/IP 连接。 然后发送队列中的电子邮件。	--
70036	没有组态任何用于发送电子邮件的 SMTP 服务器。 尝试连接 SMTP 服务器失败，无法发送电子邮件。 WinCC Runtime 在首次尝试发送电子邮件后生成系统事件。	组态 SMTP 服务器： 在 WinCC 工程系统中 使用“设备设置”>“设备设置”。 在 Windows CE 操作系统中 使用“控制面板”>“Internet 设置”>“电子邮件”>“SMTP 服务器”
70037	电子邮件由于未知原因而无法发送。 电子邮件的内容将丢失。	检查电子邮件参数，如“收件人”等。
70038	SMTP 服务器拒绝发送或转发电子邮件，因为接收方的域对于服务器是未知的，或因为 SMTP 服务器需要进行身份认证。 电子邮件的内容丢失。	检查收件人地址的域。或： 如有可能，禁用 SMTP 服务器上的身份验证。 WinCC Runtime 目前没有使用 SMTP 身份验证。
70039	电子邮件地址的语法不正确或包含非法字符。 电子邮件的内容被丢弃。	检查接收方的电子邮件地址。
70040	电子邮件地址的语法不正确或包含非法字符。	--
70041	导入用户管理因错误而中断。 未导入任何内容。	检查用户管理，或再次将其下载到面板。

8.4 使用报警

编号	效果/原因	解决方法
70042	执行系统函数时超出了变量的数值范围。 未进行系统函数计算。	检查期望的计算并进行更正。
70043	执行系统函数时超出了变量的数值范围。 未进行系统函数计算。	检查期望的计算并进行更正。

表格 8-9 80000 - 归档报警

编号	效果/原因	解决方法
80001	对指定日志的填充已达到定义的大小 (达到某个百分比), 因而必须存储在别处。	通过执行“move”或“copy”函数存储文件或表格。
80002	指定日志中的行已丢失。	--
80003	记录复制过程不成功。 在这种情况下, 也建议检查所有随后出现的系统报警。	--
80006	因为不能进行记录, 将导致永久不能执行该功能。	对于数据库, 检查相关数据源是否存在, 并重新启动系统。
80009	复制操作已经成功完成。	--
80010	由于在 WinCC 中没有正确输入存储位置, 将导致该功能永久丧失。	如果需要完整的功能, 请重新组态日志的存储位置并重新启动系统。
80012	日志条目存储在缓冲区中。 如果将数值读入缓冲区比数据的物理写入 (例如使用硬盘) 快, 则可能导致超载, 从而停止进行记录。	减少归档的值。 或: 增加记录周期。
80013	超载状态不再适用。 归档将重新恢复对所有数值的记录。	--
80014	同样的操作被连续触发两次。 因为过程已经进行, 所以操作将只执行一次。	--
80015	该系统报警用于将 DOS 或数据库错误报告给用户。	--
80016	日志被系统函数“CloseAllLogs”分开, 到来的条目超出所定义的缓冲区大小。 缓冲区中的所有条目都将被删除。	重新连接日志。
80017	到来事件数引起缓冲区溢出。 其它原因也会导致此类情况, 例如同时激活多个复制作业。 缓冲区中的所有复制作业都将被删除。	停止复制操作。
80019	WinCC 和所有日志之间的连接被取消, 例如, 在运行“CloseAllLogs”系统函数之后。 条目会被写入缓冲区, 并在重新建立连接后写入到日志中。 没有与存储位置的连接, 存储介质可能已被替换。	--
80020	已经超过同时可激活的复制作业的最大数目。 复制不被执行。	等待直到当前复制作业已完成, 然后重新启动上一个复制作业。

编号	效果/原因	解决方法
80021	试图删除仍在进行复制的日志。删除没有被执行。	等待直到当前复制作业已完成，然后重新启动上一个操作。
80022	试图使用系统函数“StartSequenceLog”来为没有组态为分段循环日志的日志启动日志分段。没有创建任何日志分段文件。	在项目中，检查下列内容： <ul style="list-style-type: none"> 系统函数“StartSequenceLog”是否已正确组态？ HMI 设备上是否已正确提供了变量参数的值？
80023	试图将日志复制到其自身。未能复制日志。	在项目中，检查下列内容： <ul style="list-style-type: none"> 系统函数“CopyLog”是否已正确组态？ HMI 设备上是否已正确提供了变量参数的值？
80024	当目标日志已经包含数据（“模式”参数）时，系统函数“CopyLog”不允许进行复制。日志不被复制。	在组态中编辑“CopyLog”系统函数。在启动系统函数之前，删除目标日志文件。
80025	您已取消复制作业。 在当前时刻之前写入的数据将被保留。未删除目标日志文件（如果已组态）。 取消操作将在目标日志的末尾，以错误条目 \$RT_ERR \$ 进行报告。	--
80026	在所有的日志初始化后，输出该报警。数值将从此刻起写入日志。在此之前，即使 WinCC Runtime 在运行，也没有任何条目写入日志。	--
80027	内部闪存已被指定为日志的存储位置。这是不允许的。 不会为该日志写入任何数值并且不会创建日志文件。	将存储卡或网络路径组态为存储位置。
80028	报警返回一个状态报告，指示当前正在初始化日志。不记录任何值，直到输出报警 80026。	--
80029	在报警中指定的日志数目不能进行初始化。初始化日志。 有错误的日志文件不可用于作业的记录。	评估与此报警相关联的其它系统报警。检查组态、ODBC（开放数据库连通性）和指定的驱动器。
80030	当前日志文件的结构与期望的结构不匹配。 该日志的记录进程停止。	提前手工删除现有的日志数据。
80031	CSV 格式的日志已损坏。 日志无法再用。	删除有错误的文件。
80032	日志可以分配有事件。一旦日志满，就将触发这些事件。如果 WinCC Runtime 已经启动，而日志已满，则不会触发事件。 指定的日志不再记录数据，因为该日志已满。	退出 WinCC Runtime。删除日志并重新启动 WinCC Runtime。 或者： 组态包含与该事件相同操作的按钮。单击按钮。
80033	在数据日志文件中设置了“已定义的系统”作为数据源名称。这将导致一个错误。数据未被写入到数据库日志，而是被记录到 CSV 日志中。	重新安装 MSDE。
80034	在日志的初始化过程中产生了错误。试图创建表格作为备份。该动作已经成功了。已经对损坏的日志文件的表格进行了备份，且清空的日志也已经重新启动。	保存或删除备份以释放内存。

编号	效果/原因	解决方法
80035	在日志的初始化过程中产生了错误。试图创建表格的备份，但未能成功。没有进行任何记录或备份。	保存或删除备份以释放内存。
80044	因运行系统关闭或停电导致日志导出中断。重新启动运行系统时检测到需要恢复导出。	自动恢复导出。
80045	日志导出因到服务器的连接错误或服务器本身错误而中断。	自动重复导出。检查以下内容： <ul style="list-style-type: none"> 与服务器的连接是否完好？ 服务器是否在运行？ 服务器上是否有足够的可用空间？
80046	目标文件或相关文件夹无法在服务器上进行创建。	检查服务器上是否有足够的空间以及您是否具有保存日志文件所需的权限。
80047	导出日志时不能读取该日志。	检查是否正确插入了存储介质。
80049	准备导出日志时不能重命名该日志。无法完成作业。	检查是否正确插入了存储介质以及该存储介质上是否有足够的空间。
80050	未关闭要导出的日志。无法完成作业。	确保在使用系统函数“ExportLog”之前调用系统函数“CloseAll Logs”。根据要求更改组态。

表格 8-10 90000 - FDA 报警

编号	效果/原因	解决方法
90024	因存储介质上缺少用于日志的空间，从而导致不能记录操作员操作。因此将不会执行操作员操作。	插入一个空的存储介质或使用“ExportLog”换出服务器上的日志文件，以获取更多的可用空间。
90025	因归档状态错误而导致无法记录用户操作。因此将不会执行用户操作。	检查是否正确插入了存储介质。
90026	因日志关闭而不能记录操作员操作。因此将不会执行操作员操作。	在执行任何其它操作之前，使用“OpenAllLogs”系统函数重新打开日志。根据要求更改组态。
90028	密码错误。	输入正确的密码。
90029	执行当前操作时关闭了运行系统（可能由于断电所致）或使用的存储介质与“检查跟踪”不兼容。如果“检查跟踪”属于另一个项目或已经归档，则不适合使用。	确保使用正确的存储介质。
90030	执行当前操作时关闭了运行系统（可能是由于断电而导致的）。	--
90031	执行当前操作时关闭了运行系统（可能是由于电源故障而导致的）。	--
90032	存储介质上的日志空间用尽。	插入一个空的存储介质或使用“ExportLog”系统函数保存服务器上的日志文件，以获得更多的可用空间。
90033	存储介质上无更多的日志空间。从现在起，将不再执行需要记录的更多操作员操作。	为了释放内存，请使用“ExportLog”系统函数保存服务器上的日志文件。 或： 插入一个空的存储介质。
90039	没有执行该操作所需的权限。	更换或者升级您的授权。

编号	效果/原因	解决方法
90040	因强制用户操作而导致“检查跟踪”被关闭。	使用“StartLog”系统函数再次激活“检查跟踪”。
90041	在没有登录用户的情况下执行了必须记录的用户操作。	只有具有相应的权限才可执行需要记录的用户操作。通过为操作对象组态权限来更改组态。
90044	由于存在另一个未决的用户操作，从而导致需要确认的用户操作受阻。	必要时重复该用户动作。

表格 8-11 110000 - 离线函数报警

编号	效果/原因	解决方法
110000	已经更改了操作模式。现在设置为“离线”模式。	--
110001	已经更改了操作模式。现在设置为“在线”模式。	--
110002	未更改操作模式。	检查与 PLC 的连接。 检查在 PLC 中“协调”区域指针的地址范围是否存在。
110003	指定 PLC 的操作模式已经由系统函数“SetConnectionMode”更改。 现在的操作模式为“离线”。	--
110004	指定 PLC 的操作模式已经由系统函数“SetConnectionMode”更改。 现在的操作模式为“在线”。	--
110005	虽然整个系统处于“离线”模式，但试图使用系统函数“SetConnectionMode”将指定 PLC 切换为“在线”模式。不允许该转换。PLC 将仍然保持“离线”模式。	将整个系统切换为“在线”模式并再次执行系统函数。
110006	区域指针“项目 ID”的内容与在 WinCC 中组态的项目 ID 不匹配。因此，WinCC Runtime 将关闭。	检查： <ul style="list-style-type: none"> 在 PLC 中输入的项目 ID 是否正确？ 在 WinCC 中输入的项目 ID 是否正确？

表格 8-12 120000 - 趋势报警

编号	效果/原因	解决方法
120000	不能显示趋势图，因为您组态的坐标或趋势图不正确。	更改组态。
120001	不能显示趋势图，因为您组态的坐标或趋势图不正确。	更改组态。
120002	趋势没有显示，因为分配的变量试图访问的 PLC 地址无效。	检查： <ul style="list-style-type: none"> PLC 中是否存在用于该变量的数据区域？ 组态的地址是否正确？ 变量值的范围是否正确？

8.4 使用报警

表格 8-13 130000 - 系统信息报警

编号	效果/原因	解决方法
130000	操作不被执行。	关闭所有其它程序。 删除硬盘中不再需要的文件。
130001	操作不被执行。	删除硬盘中不再需要的文件。
130002	操作不被执行。	关闭所有其它程序。 删除硬盘中不再需要的文件。
130003	没有找到任何数据介质。操作被取消。	检查： <ul style="list-style-type: none"> • 是否访问了正确的数据介质？ • 是否插入了数据介质？
130004	数据介质被写保护。操作被取消。	检查： <ul style="list-style-type: none"> • 是否访问了正确的数据介质？ • 文件是否为写保护的？
130005	该文件属性为只读。操作被取消。	检查是否访问了正确的文件。如有必要，更改文件属性。
130006	对文件的访问失败。操作被取消。	检查： <ul style="list-style-type: none"> • 是否访问了正确的文件？ • 文件是否存在？ • 是否同时有另一操作阻止了对文件的访问？
130007	网络连接中断。 不能通过网络连接保存或读取记录。	检查网络连接并排除出错原因。
130008	存储卡无法使用。 记录不能保存到存储卡或无法从存储卡中读取。	插入存储卡。
130009	指定的文件夹不在存储卡上。 当 HMI 设备关闭后，该文件夹下的任何文件都不能进行备份。	插入存储卡。
130010	最大嵌套深度可能用完，例如，一个脚本中的数值变化可能会导致其下一个脚本的调用，而接下来第二个脚本中的数值变化又会导致它的下一个脚本调用，依此类推。 不支持所组态的功能。	检查组态。

表格 8-14 140000 - 连接报警：chns7：连接类型 + 设备

编号	效果/原因	解决方法
140000	已建立与 PLC 的在线连接。	--
140001	已断开与 PLC 的在线连接。	--
140003	不执行任何变量更新或写操作。	检查连接以及 PLC 是否已接通。 在控制面板中，通过“设置 PG/PC 接口”检查参数定义。 重新启动系统。

编号	效果/原因	解决方法
140004	不执行任何变量更新或写操作，因为存取点或模块组态不正确。	验证连接，并检查 PLC 是否已接通。 在控制面板中，通过“设置 PG/PC 接口”检查存取点或模块组态 (MPI、PPI、PROFIBUS)。 重新启动系统。
140005	不执行任何变量更新或写操作，因为 HMI 设备的地址不正确 (可能太高)。	使用不同的 HMI 设备地址。 验证连接，并检查 PLC 是否已接通。 在控制面板中，通过“设置 PG/PC 接口”检查参数定义。 重新启动系统。
140006	不执行任何变量更新或写操作，因为波特率不正确。	在 WinCC 中选择不同的波特率 (根据模块、配置文件、通信伙伴等)。
140007	没有更新或写入变量，因为总线配置文件不正确 (请参阅 %1)。 无法将下列参数写入注册表中： 1：Tslot 2：Tqui 3：Tset 4：MinTsdr 5：MaxTsdr 6：Trdy 7：Tid1 8：Tid2 9：间隙因子 10：重试限制	检查用户定义的总线配置文件。 检查连接，并检查 PLC 是否已接通。 在控制面板中，通过“设置 PG/PC 接口”检查参数定义。 重新启动系统。
140008	不执行任何变量更新或写操作，因为项目数据不正确。下列参数不能写入注册表： 0：常规错误 1：版本错误 2：配置文件不能写入注册表。 3：子网类型不能写入注册表。 4：目标轮询时间不能写入注册表。 5：最高地址 (HSA) 错误。	检查连接以及 PLC 是否已接通。 在控制面板中，通过“设置 PG/PC 接口”检查参数定义。 重新启动系统。
140009	不执行任何变量更新或写操作，因为未找到用于 S7 通信的模块。	在控制面板中使用“设置 PG/PC 接口”重新安装模块。
140010	找不到 S7 通信伙伴，因为 PLC 已关闭。 DP/T： 没有在控制面板的“设置 PG/PC 接口”下选择“PG/PC 是唯一主站”选项。	接通 PLC。 DP/T： 如果仅有一台主站连接到网络上，则在“设置 PG/PC 接口”中禁用“PG/PC 是唯一主站”选项。 如果有多个主站连接到网络上，则启用该选项。切勿更改任何设置，否则将导致总线错误。
140011	不执行任何变量更新或写操作，因为通信已经中断。	检查连接以及通信伙伴是否接通。
140012	初始化存在问题 (当在任务管理器中关闭 WinCC Runtime 时)。 或者： 另一个使用不同总线参数的应用程序 (例如，STEP7) 正在运行，而驱动程序无法使用新的总线参数 (如传输率) 启动。	重新启动 HMI 设备。 或者： 先启动 WinCC Runtime，然后再启动其它应用程序。

8.4 使用报警

编号	效果/原因	解决方法
140013	MPI 电缆已断开，因而不存在任何电源供应。	检查连接。
140014	所组态的总线地址已被另一应用程序使用。	更改 PLC 组态中的 HMI 设备地址。
140015	错误的传输率 或者： 错误的总线参数（例如，HSA） 或者： OP 地址 > HSA 或者：错误的中断向量（中断未到达驱动程序）	更正相关参数。
140016	硬件不支持该组态中断。	更改中断号。
140017	设置的中断正被另一驱动程序使用。	更改中断号。
140018	一致性检查被 SIMOTION Scout 禁用。只显示一个相应注释。	再次启用 SIMOTION Scout 的一致性检查，并再次将项目下载到 PLC 中。
140019	SIMOTION Scout 正在将新项目下载到 PLC。与 PLC 的连接被取消。	等待直至重新组态结束。
140020	PLC 中的版本与项目版本（FWX 文件）不匹配。与 PLC 的连接被取消。	可用下列方法纠正： 使用 SIMOTION Scout 将当前版本下载到 PLC。 使用 WinCC ES 重新生成项目，关闭 WinCC Runtime，然后使用新的组态重新启动。

表格 8-15 150000 - 连接报警：chnAS511：连接

编号	效果/原因	解决方法
150000	不再读取或写入数据。可能的原因： <ul style="list-style-type: none"> • 电缆出现故障。 • PLC 没有反应或出现故障等。 • 用于连接的端口错误。 • 系统过载 	确保电缆已插入，PLC 能运作，并使用了正确的端口。 如果系统报警持续显示，请重新启动系统。
150001	连接重新建立，因为引起中断的原因已经消除。	--

表格 8-16 160000 - 连接报警：IVar (WinLC)/OPC：连接

编号	效果/原因	解决方法
160000	不再读取或写入数据。可能的原因： <ul style="list-style-type: none"> • 电缆出现故障。 • PLC 没有反应或出现故障等。 • 用于连接的端口错误。 • 系统过载 	确保电缆已插入，PLC 能运作，并使用了正确的端口。 如果系统报警持续显示，请重新启动系统。
160001	连接重新建立，因为引起中断的原因已经消除。	--
160010	不存在与服务器的任何连接，因为无法确定服务器标识 (CLS-ID)。无法读出或写入值。	检查访问权限。

编号	效果/原因	解决方法
160011	不存在与服务器的任何连接，因为无法确定服务器标识 (CLS-ID)。 无法读出或写入值。	检查： <ul style="list-style-type: none"> • 服务器名称是否正确？ • 计算机名称是否正确？ • 服务器是否已注册？
160012	不存在与服务器的任何连接，因为无法确定服务器标识 (CLS-ID)。 无法读出或写入值。	例如，检查 <ul style="list-style-type: none"> • 服务器是否名称正确。 • 计算机名称是否正确。 • 服务器是否已注册。 高级用户请注意： 可以根据 HRESULT 解读该值。
160013	指定服务器作为 InProc 服务器启动。未提供此功能，这样做可能导致未知结果，因为服务器与 WinCC Runtime 运行在相同的过程区域中。	将服务器组态为 OutProc 服务器或本地服务器。
160014	在 PC/MP 上只能启动一个 OPC 服务器项目。试图启动第二个项目时，输出一条报警信息。 第二个项目没有 OPC 服务器功能，因而外部数据源不能将它视为 OPC 服务器。	切勿启动计算机上具有 OPC 服务器功能的第二个项目。

表格 8-17 170000 - S7 诊断报警

编号	效果/原因	解决方法
170000	S7 诊断事件不显示，因为不可能使用该设备登录到 S7 诊断功能。不支持该服务。	--
170001	无法查看 S7 诊断缓冲区，因为与 PLC 的通信已经中断。	将 PLC 设置为在线模式。
170002	无法查看 S7 诊断缓冲区，因为诊断缓冲区 (SSL) 中的读操作由于出现错误而被取消。	--
170003	不能显示 S7 诊断事件。系统返回内部错误 %2。	--
170004	不能显示 S7 诊断事件。系统返回内部错误，错误类别为 %2，错误代码为 %3。	--
170007	不能读取 S7 诊断缓冲区 (SSL)，因为发生内部错误 (错误类别 %2，错误代码 %3)，该操作被取消。	--

表格 8-18 180000 - 其它/通用报警

编号	效果/原因	解决方法
180000	组件/OCX 接收的项目数据的版本 ID 不被支持。	安装更新的组件。
180001	由于并行运行的操作太多，系统出现过载。并非所有操作均可执行，某些操作将被拒绝。	可以使用下列方法进行纠正： <ul style="list-style-type: none"> • 降低报警的生成速度 (轮询)。 • 延长脚本和函数的启动间隔。 如果警告出现得更频繁， 请重新启动 HMI 设备。

8.4 使用报警

编号	效果/原因	解决方法
180002	不能激活屏幕键盘。可能的原因： 文件“TouchInputPC.exe”由于安装错误而没有注册。	重新安装 WinCC Runtime。

表格 8-19 190000 - 变量报警

编号	效果/原因	解决方法
190000	可能变量没有更新。	--
190001	在上一个错误原因已排除之后，变量被更新（返回到正常操作）。	--
190002	变量没有更新，因为与 PLC 的通信已经中断。	选择系统函数“SetOnline”以进入在线模式。
190004	变量没有更新，因为所组态的变量地址不存在。	检查组态。
190005	变量没有更新，因为组态的 PLC 类型中不存在该变量。	检查组态。
190006	变量没有更新，因为不能将 PLC 类型映射到变量的数据类型中。	检查组态。
190007	变量值没有修改，因为与 PLC 的连接已经中断或变量处于离线状态。	设置在线模式或者重新连接到 PLC。
190008	超出为该变量组态的阈值范围，例如由于 <ul style="list-style-type: none"> • 输入的值 • 系统函数 • 脚本 	遵守变量的组态或当前阈值。
190009	试图将数值分配给此变量，但该数值超出了此数据类型允许的取值范围。 实例： 为“Byte”类型的变量分配“260” 为“String”类型的变量分配“-3”	遵守变量数据类型的数值范围。
190010	为该变量写入过多的值（例如，在一个脚本触发的循环中）。 数值将丢失，因为缓冲区中最多只能存储 100 个动作。	可用下列方法纠正： <ul style="list-style-type: none"> • 延长多次写动作之间的时间间隔。 • 在使用“确认 HMI”组态 HMI 设备上的确认时，请勿使用超过 6 个字长的数组变量。
190011	可能原因 1： 输入的数值无法写入到组态的 PLC 变量中，因为该值超出上下限范围。 系统拒绝该输入并恢复原先的数值。 可能原因 2： 与 PLC 的连接被中断。	请确保输入的数值不超出控制变量的数值范围。 检查至 PLC 的连接。
190012	不能将数值从源格式转换为目标格式，例如： 试图将数值分配给计数器，但该数值超出了 PLC 规定的有效数值范围。 要为整型变量分配一个字符串类型的值。	请检查变量的数值范围或数据类型。

编号	效果/原因	解决方法
190013	用户输入了一个长于该变量的字符串。该字符串自动缩短为允许的长度。	只能输入不超过允许变量长度的字符串。

表格 8-20 190100 - 区域指针报警

编号	效果/原因	解决方法
190100	区域指针没有更新，因为为其组态的地址不存在。 类型 1 警告 2 错误 3 PLC 确认 4 HMI 设备确认 5 LED 映射 6 趋势请求 7 趋势传送 1 8 趋势传送 2 编号： 在 WinCC ES 中显示的连续编号。	检查组态。
190101	区域指针没有更新，因为不能将 PLC 类型映射到区域指针类型。 参数类型和编号： 参见报警 190100	--
190102	在上一个错误原因已排除之后，区域指针被更新（返回到正常操作）。参数类型和编号：参见报警 190100。	--

表格 8-21 200000 - PLC 协调报警

编号	效果/原因	解决方法
200000	不执行协调，因为 PLC 中组态的地址不存在或尚未设置。	更改 PLC 中的地址或在其中设置地址。
200001	协调被取消，因为不能对在 PLC 中组态的地址进行写访问。	在 PLC 可进行写操作的区域中，更改或设置地址。
200002	此时不执行协调，因为区域指针的地址格式与内部存储格式不匹配。	内部错误
200003	协调可再次执行，因为上一个错误已经消除（返回到正常操作）。	--
200004	协调可能不执行。	--
200005	不再读取或写入数据。可能的原因： <ul style="list-style-type: none"> • 电缆出现故障。 • PLC 没有反应或出现故障等。 • 系统过载 	确保电缆已插入，PLC 能运作。 如果系统报警一直存在，请重新启动系统。

8.4 使用报警

表格 8-22 200100 - PLC 用户版本报警

编号	效果/原因	解决方法
200100	没有执行协调，因为 PLC 中组态的地址不存在或尚未设置。	更改 PLC 中的地址或在其中设置地址。
200101	协调被取消，因为不能对在 PLC 中组态的地址进行写访问。	在 PLC 可进行写操作的区域中，更改或设置地址。
200102	此时不执行协调，因为区域指针的地址格式与内部存储格式不匹配。	内部错误
200103	协调可再次执行，因为上一个错误已经消除（返回到正常操作）。	--
200104	协调可能不执行。	--
200105	不再读取或写入数据。可能的原因： <ul style="list-style-type: none"> • 电缆出现故障。 • PLC 没有反应或出现故障等。 • 系统过载 	确保电缆已插入，PLC 能运作。 如果系统报警一直存在，请重新启动系统。

表格 8-23 210000 - PLC 作业报警

编号	效果/原因	解决方法
210000	作业没有处理，因为在 PLC 中组态的地址不存在或尚未设置。	更改 PLC 中的地址或在其中设置地址。
210001	作业没有处理，因为无法对 PLC 中组态的地址进行读/写访问。	在 PLC 可进行读/写访问的区域中，更改或设置地址。
210002	作业未执行，因为区域指针的地址格式与内部存储格式不匹配。	内部错误
210003	再次处理作业缓冲区，因为上一个错误已经消除（返回到正常操作）。	--
210004	可能不处理作业缓冲区。	--
210005	触发了具有非法编号的控制请求。	检查 PLC 程序。
210006	试图执行该控制请求时出错。结果是不执行该控制请求。查看后续/先前的系统报警。	检查控制请求的参数。重新编译组态。

表格 8-24 220000 - WinCC 通道适配器报警

编号	效果/原因	解决方法
220001	变量没有下载，因为相关的通信驱动程序/HMI 设备不支持下载布尔/离散型数据。	更改组态。
220002	变量没有下载，因为相关的通信驱动程序/HMI 设备不支持对字节型数据的写访问。	更改组态。
220003	不能装载通信驱动程序。驱动程序可能没有安装。	通过重新安装 WinCC Runtime 安装驱动程序。

编号	效果/原因	解决方法
220004	通信终止且不传送任何更新数据，因为电缆没有连接或出现故障等。	检查连接。
220005	建立通信。	--
220006	指定 PLC 与指定端口之间的连接已激活。	--
220007	与指定 PLC 的连接在指定端口处中断。	检查： <ul style="list-style-type: none"> • 电缆是否插入 • PLC 是否正常 • 使用的端口是否正确。 • 组态是否正常（端口参数、协议设置、PLC 地址）。 如果系统报警一直存在，请重新启动系统。
220008	通信驱动程序不能访问或打开指定端口。另外一个应用程序可能正在使用该端口，或是使用的端口在目标系统中尚且无法使用。不能与 PLC 进行任何通信。	关闭所有访问该端口的应用程序，然后重启计算机。使用系统的另外一个端口。

表格 8-25 230000 - 视图报警

编号	效果/原因	解决方法
230000	输入的数值不被接受。系统拒绝该输入并恢复先前的数值。 可能是以下原因之一 <ul style="list-style-type: none"> • 超出数值范围 • 输入了非法字符 • 超过了允许的最多用户数。 	输入有效值或删除不需要的用户。
230002	当前登录的用户没有所需权限。因此，系统拒绝该输入并恢复原来的值。	以具有适当权限的用户身份登录。
230003	无法切换到指定画面，因为画面不可用/未组态。保持选定当前画面。	组态画面并检查画面选择函数。
230005	变量值的范围超出 I/O 域。 保持变量的原始值。	在输入值的时候要遵守该变量的数值范围。
230100	在使用 Web 浏览器浏览时，系统可能会返回一个操作员感兴趣的消息。 Web 浏览器可以继续运行，但不能（完整）显示新网页。	浏览至另一个页面。
230200	与 HTTP 通道的连接由于出现错误而中断。该错误由另一个系统报警详细解释。 不再交换数据。	检查网络连接。 检查服务器的组态。
230201	与 HTTP 通道的连接已经建立。 完成数据交换。	--

编号	效果/原因	解决方法
230202	<p>WININET.DLL 已经检测到一个错误。当尝试与服务器连接失败或服务器拒绝无授权客户机的连接时，通常产生该错误。</p> <p>当连接使用 SSL 加密时，未知的服务器证书也可能导致该错误。</p> <p>报警文本将提供详细信息。</p> <p>该文本始终以所安装的 Windows 的语言显示，因为文本是由 Windows 操作系统返回的。</p> <p>不再交换过程值。</p>	<p>取决于原因：</p> <p>当连接失败或者出现超时错误时：</p> <ul style="list-style-type: none"> • 检查网络连接和网络。 • 检查服务器地址。 • 检查 WebServer 是否确实在目标站上运行。 <p>授权错误时：</p> <ul style="list-style-type: none"> • 组态的用户名和/或密码与服务器上的用户名和/或密码不匹配。使其一致 <p>当服务器证书被拒绝时：</p> <p>证书以“未知 CA ()”标记：</p> <ul style="list-style-type: none"> • 在项目中忽略该项，或者 • 安装一个使用客户端计算机已知根证书签名的证书。 <p>证书日期无效：</p> <ul style="list-style-type: none"> • 在项目中忽略该项，或者 • 在服务器上安装具有有效日期的证书。 <p>无效 CN (Common Name, 通用名称；或者，Computer Name, 计算机名称)：</p> <ul style="list-style-type: none"> • 在项目中忽略该项，或者 • 安装一个名称与服务器地址名相对应的证书。
230203	<p>虽然可进行与服务器的连接，但 HTTP 服务器仍然拒绝连接，因为</p> <ul style="list-style-type: none"> • WinCC Runtime 没有在服务器上运行，或者 • 不支持 HTTP 通道 (503 服务不可用)。 <p>只有在 Webserver 不支持 HTTP 通道时，才会发生其它错误。报警文本的语言取决于 Webserver。</p> <p>数据没有交换。</p>	<p>出现错误：503 服务不可用：</p> <p>检查 WinCC Runtime 是否正在服务器上运行，并检查是否支持 HTTP 通道。</p>
230301	<p>产生了内部错误。英语文本更详细地解释了该错误。这可能是因存储空间不够所引起的。</p> <p>OCX 不起作用。</p>	--
230302	<p>无法解析远程服务器的名称。</p> <p>连接失败。</p>	<p>检查组态的服务器地址。</p> <p>检查网络的 DNS 服务是否可用。</p>
230303	<p>所寻址的计算机上没有远程服务器在运行。</p> <p>服务器地址不正确。</p> <p>连接失败。</p>	<p>检查组态的服务器地址。</p> <p>检查目标计算机是否正在运行远程服务器。</p>
230304	<p>所寻址计算机上的远程服务器与 VNCOCX 不兼容。</p> <p>连接失败。</p>	<p>使用兼容的远程服务器。</p>
230305	<p>身份认证失败，因为密码不正确。</p> <p>连接失败。</p>	<p>组态正确的密码。</p>

编号	效果/原因	解决方法
230306	连接到远程服务器时出错。此类情况可能是由于网络问题所导致的。 连接失败。	检查： <ul style="list-style-type: none"> 总线电缆是否已插入。 网络是否存在问题。
230307	至远程服务器的连接被终止，因为 <ul style="list-style-type: none"> 远程服务器关机，或者 用户指示服务器关闭所有连接。 连接被关闭。	--
230308	该报警提供有关连接状态的信息。 尝试建立连接。	--

表格 8-26 240000 - 授权报警

编号	效果/原因	解决方法
240000	WinCC Runtime 以演示模式运行。 未安装授权或授权不正确。	安装授权。
240001	WinCC Runtime 以演示模式运行。 安装的版本组态了过多变量。	装载一个合适的授权/授权组。
240002	WinCC Runtime 以有时间限制的紧急授权运行。	恢复完整的授权。
240004	读取紧急授权时出错。 WinCC Runtime 以演示模式运行。	重新启动 WinCC Runtime，安装授权或修复授权（请参见软件保护调试说明）。
240005	自动化许可证管理器检测到内部系统故障。 可能的原因： <ul style="list-style-type: none"> 文件有损坏 安装有缺陷 没有可供自动化许可证管理器使用的空间等 	重新启动 HMI 设备或 PC。如果问题未得到解决，请删除自动化许可证管理器并重新安装。

表格 8-27 250000 - S7 强制报警

编号	效果/原因	解决方法
250000	在“状态强制”指定行中的变量未进行更新，因为针对该变量组态的地址不可用。	检查所设置的地址，然后确认是否已在 PLC 中设置该地址。
250001	在“状态强制”指定行中的变量未进行更新，因为针对该变量组态的 PLC 类型不存在。	检查所设置的地址。
250002	“状态强制”中指定行内的变量未被更新，因为无法将 PLC 类型映射到该数据类型中。	检查所设置的地址。
250003	与 PLC 的连接失败。变量未更新。	检查至 PLC 的连接。检查 PLC 是否已经接通以及是否在线。

表格 8-28 260000 - 密码报警

编号	效果/原因	解决方法
260000	系统中已经输入了未知用户或未知密码。 当前用户将从系统中注销。	以具有有效密码的用户登录到系统。
260001	登录的用户没有足够的权限来执行系统受保护的功能。	以具有足够授权的用户登录到系统。
260002	该报警由系统函数“TrackUserChange”触发。	--
260003	用户已经从系统中注销。	--
260004	输入到用户视图中的用户名已经存在于用户管理中。	选择另一个用户名，因为用户名在用户管理中必须唯一。
260005	放弃输入。	输入更短的用户名。
260006	放弃输入。	使用更短或更长的密码。
260007	输入的登录超时值超出有效范围（0 到 60 分钟）。 拒绝新值，保持原值。	输入的登录超时值必须在 0 到 60 分钟之间。
260008	在 WinCC 中尝试读取由 ProTool V 6.0 创建的 PTProRun.pwl 文件。 由于格式不兼容，文件读操作已取消。	--
260009	您已试图删除用户“管理员”或“PLC 用户”。这些用户是用户管理的固定组件，无法删除。	如果由于已超出最大允许的用户数而需要删除用户，请删除其它用户。
260012	在“更改密码”对话框中输入的密码不相同。 密码未更改。用户将注销。	您必须重新登录系统。然后输入相同的密码两次，才能更改密码。
260013	在“更改密码”对话框中输入的密码无效，因为该密码已经在使用。 密码未更改。用户将注销。	您必须重新登录系统。然后输入一个以前未用过的新密码。
260014	您已经连续三次尝试登录失败。 您将被锁定并被分配到组 0 中。	您可使用正确密码登录系统。只有管理员可以改变组分配。
260023	您输入的密码不符合必要的密码规则。	输入一个包括至少一个数字的密码。
260024	您输入的密码不符合必要的密码规则。	输入一个包括至少一个字符的密码。
260025	您输入的密码不符合必要的密码规则。	输入一个包括至少一个特殊字符的密码。
260028	当运行系统启动、尝试进行登录或尝试更改 SIMATIC Logon 用户的密码时，系统将尝试访问 SIMATIC Logon 服务器。 如果尝试登录，则新用户不登录。如果在此之前已经有一个不同的用户登录，则注销此用户。	检查到 SIMATIC Logon 服务器的连接及其配置，例如： 1. 端口号 2. IP 地址 3. 服务器名称 4. 功能传送电缆 或者使用一个本地的用户。

编号	效果/原因	解决方法
260029	SIMATIC Logon 用户并未与任何一个或者多个组关联。 新用户不登录。如果在此之前已经有一个不同的用户登录，则注销此用户。	检查 SIMATIC Logon 服务器上的用户数据以及项目的组态情况。一个用户只能分配给一个组。
260030	SIMATIC Logon 用户不能在 SIMATIC Logon 服务器上更改其密码。新密码可能不符合服务器上的密码规则，或者用户无权更改密码。 旧密码仍然生效，并且注销用户。	再次登录并选择另一个密码。检查 SIMATIC Logon 服务器上的密码规则。
260031	用户不能登录到 SIMATIC Logon 服务器上。用户名或者密码错误，或者用户权限有限不能登录。 新用户不登录。如果在此之前已经有一个不同的用户登录，则注销此用户。	再试一次。如有必要，检查 SIMATIC Logon 服务器上密码数据。
260032	由于帐户已经锁定，所以用户不能登录到 SIMATIC Logon 服务器上。 新用户不登录。如果在此之前已经有一个不同的用户登录，则注销此用户。	检查 SIMATIC Logon 服务器上的用户数据。
260033	不能执行更改密码或者登录用户的动作。	检查 SIMATIC Logon 服务器的组态。
260034	上一个登录操作尚未完成。因此无法调用用户操作或者登录对话框。 登录对话框未打开。用户操作未被执行。	请等待，直到该操作完成。
260035	上一次修改密码的尝试未完成。因此无法调用用户操作或者登录对话框。 登录对话框未打开。用户操作未被执行。	请等待，直到该操作完成。
260036	SIMATIC Logon 服务器的授权不够。登录权限不够。	检查 SIMATIC Logon 服务器上的许可证授权。
260037	SIMATIC Logon 服务器上无许可证授权。不能登录。 用户不能通过 SIMATIC Logon 服务器登录，只能通过本地用户登录。	检查 SIMATIC Logon 服务器上的许可证授权。

表格 8-29 270000 - 系统事件

编号	效果/原因	解决方法
270000	变量没有在报警中显示，因为它试图访问 PLC 中的无效地址。	检查变量的数据区域是否存在于 PLC 中，组态的地址是否正确以及变量的数值范围是否正确。
270001	对排队等待输出的报警数量，有一个与设备相关的限制值（参见操作说明）。报警数量超出该限制值。 视图将不能包含所有报警。 但是，所有报警都已写入报警缓冲区。	--
270002	视图中显示了关于某个日志的报警，因为当前项目中没有任何数据可用于该日志。 输出用于报警的通配符。	如有必要，删除旧的日志数据。

8.4 使用报警

编号	效果/原因	解决方法
270003	服务无法建立，因为太多设备要用该服务。 最多只能有 4 台设备执行该操作。	减少想要使用该服务的 HMI 设备的数目。
270004	无法访问永久缓冲区。无法恢复或保存报警。	如果下次启动时问题仍然存在，请联系客户支持（删除闪存内容）。
270005	永久缓冲区损坏：无法恢复报警。	如果下次启动时问题仍然存在，请联系客户支持（删除闪存内容）。
270006	项目已修改：无法从永久缓冲区恢复报警。	项目已生成并被再次下载到 HMI 设备；下次启动设备时，应该不会再出现该错误。
270007	组态问题妨碍恢复过程（缺少 DLL、文件夹未知等）。	更新操作系统，然后再次将项目下载到 HMI 设备。

表格 8-30 280000 - DPHMI 报警连接

编号	效果/原因	解决方法
280000	连接重新建立，因为引起中断的原因已经消除。	--
280001	不再读取或写入数据。可能的原因： <ul style="list-style-type: none"> • 电缆出现故障 • PLC 没有反应或出现故障等。 • 连接端口错误 • 系统过载 	检查： <ul style="list-style-type: none"> • 电缆是否插入 • PLC 是否正常 • 端口是否正确 如果系统报警一直存在，请重新启动系统。
280002	所使用的连接需要 PLC 中的一个功能块。功能块已经响应。现在启用通信。	--
280003	所使用的连接需要 PLC 中的一个功能块。功能块没有响应。	检查： <ul style="list-style-type: none"> • 电缆是否插入 • PLC 是否正常 • 端口是否正确 如果系统报警持续显示，请重新启动系统。 补救措施取决于错误代码： <ol style="list-style-type: none"> 1：功能块必须在响应容器中设置 COM 位。 2：功能块禁止在响应容器中设置 ERROR 位。 3：功能块必须在指定时间（超时时间）之内响应。 4：建立到 PLC 的在线连接。
280004	与 PLC 的连接被中断。目前没有任何数据交换。	检查 WinCC 中的连接参数。 确保电缆已插入、PLC 正常运行且使用了正确的端口。 如果仍显示该系统报警，请重新启动系统。

表格 8-31 290000 - 配方系统事件

编号	效果/原因	解决方法
290000	配方变量不能进行读或写操作。它被赋给启动值。 如有必要，最多可为四个出现故障的变量将报警输入到报警缓冲区中。此后，输出报警 290003。	检查组态，确定是否已经在 PLC 中设置了地址。
290001	试图为配方变量赋予超出该类型允许范围的值。 如有必要，可以为多达四个出现故障的变量将报警输入到报警缓冲区中。此后，输出报警 290004。	遵守数据类型的取值范围。
290002	不能将一个值从源格式转换为目标格式。 如有必要，可以为多达四个出现故障的变量将报警输入到报警缓冲区中。此后，输出报警 290005。	检查变量的数值范围或类型。
290003	当报警编号 290000 被触发超过五次时，该报警将被输出。 在这种情况下，报警将不再产生。	检查组态，确定是否已经在 PLC 中设置了变量地址。
290004	当报警编号 290001 被触发超过五次时，该报警将被输出。 在这种情况下，报警将不再产生。	遵守数据类型的取值范围。
290005	当报警编号 290002 被触发超过五次时，该报警将被输出。 在这种情况下，报警将不再产生。	检查变量的数值范围或类型。
290006	输入的值超出了为该变量组态的阈值范围。	遵守变量的组态或当前阈值。
290007	当前正在处理的配方的源和目标结构之间存在差异。源结构包含目标结构中不可用的附加的数据配方变量，因此不能进行赋值。 拒绝该值。	将指定配方中的指定数据配方变量从项目中删除。
290008	当前正在处理的配方的源和目标结构之间存在差异。目标结构包含源结构中不可用的附加的数据配方变量。 指定的数据配方变量被赋给它的启动值。	将指定的数据配方变量插入源结构中。
290010	为该配方组态的存储单元被禁用。 可能原因： 非法字符、写保护、数据介质已满或不存在。	检查组态的存储单元。
290011	指定编号的记录不存在。	检查该编号的来源（常量或变量值）。
290012	具有指定编号的配方不存在。	检查该编号的来源（常量或变量值）。
290013	试图保存记录，但该记录号已经存在。 动作没有执行。	可用下列方法纠正： <ul style="list-style-type: none"> • 检查该编号的来源（常量或变量值）。 • 首先，删除该记录。 • 更改“覆盖”函数参数。

编号	效果/原因	解决方法
290014	指定要导入的文件找不到。	检查： <ul style="list-style-type: none"> • 文件名 • 确保文件位于指定文件夹中。
290020	报警报告将记录：从 HMI 设备下载到 PLC 已经开始。	--
290021	报警报告将记录：从 HMI 设备下载到 PLC 已完成。	--
290022	报警报告将记录：从 HMI 设备下载到 PLC 因错误而取消。	检查在组态中是否存在以下情况： <ul style="list-style-type: none"> • 在 PLC 中组态了变量地址 • 存在配方号 • 存在记录号 • 设置了“覆盖”函数参数
290023	报警报告将记录：从 PLC 下载到 HMI 设备已经开始。	--
290024	报警报告将记录：从 PLC 下载到 HMI 设备已完成。	---
290025	报警报告将记录：从 PLC 下载到 HMI 设备因错误而取消。	检查在组态中是否存在以下情况： <ul style="list-style-type: none"> • 在 PLC 中组态了变量地址 • 存在配方号 • 存在记录号 • 设置了“覆盖”函数参数
290026	虽然记录在当前并不空闲，但仍试图读出/写入该记录。 在配方已经组态为同步下载的情况下，可能会出现这种错误。	将记录状态设置为零。
290027	目前无法连接到 PLC。因此，记录无法读出，也无法写入。 可能的原因： 没有与 PLC 建立任何物理连接（例如电缆没有插入、电缆出现故障）或 PLC 已关闭。	检查至 PLC 的连接。
290030	在选择含有配方视图（在其中已选择记录）的画面之后，输出该报警。	重新装载存储位置中的记录或保留当前值。
290031	在进行保存时，检测到具有指定编号的记录已经存在。	覆盖该记录或取消该操作。
290032	在导出记录时，检测到指定名称的文件已经存在。	覆盖文件或取消操作。
290033	在删除记录之前出现确认请求。	--
290040	发生了无法对它进行更详细描述的错误代码为 1% 的记录错误。 操作被取消。 可能是 PLC 上没有正确安装该记录。	检查存储位置、记录、“数据记录”区域指针以及与 PLC 的连接（在必要时）。 在短暂等待之后，重新启动该操作。 如果错误仍存在，请联系“客户支持”。将相关的错误代码发送给“客户支持”。

编号	效果/原因	解决方法
290041	无法保存记录或文件，因为存储单元已满。	删除不再需要的文件。
290042	试图同时执行多个配方操作。最后一个操作不被执行。	在短暂等待之后，再次触发该操作。
290043	在存储记录之前出现确认请求。	--
290044	配方的数据存储已被破坏，且被删除。	--
290050	报警报告：已经开始导出记录。	--
290051	报警报告：已经完成导出记录。	--
290052	报警报告：因错误而取消导出记录。	确存储位置中记录的结构与 HMI 设备上的当前配方结构完全相同。
290053	报警报告：已经开始导入记录。	--
290054	报警报告：已经完成导入记录。	--
290055	报警报告：因错误而取消导入记录。	确存储位置中记录的结构与 HMI 设备上的当前配方结构完全相同。
290056	当读/写指定行/列中的值时出错。 操作被取消。	检查指定的行/列。
290057	指定配方的变量已从“离线”切换为“在线”模式。 现在，一旦配方中的变量有更改，就会立即下载到 PLC 中。	--
290058	指定配方的变量已从“在线”切换为“离线”模式。 对该配方中变量的修改将不再立即传送到 PLC 中，但是必须通过下载记录明确地传送到 PLC 中。	--
290059	报警报告：已保存指定记录。	--
290060	报警报告：指定记录存储器已清空。	--
290061	报警报告：因错误而取消清空记录存储器。	--
290062	记录号超过了最大值 65536。 无法创建该记录。	选择其他编号。
290063	这发生在系统函数“ExportDataRecords”的参数“Overwrite”设置为“否”时。 试图用一个已存在的文件名保存配方。 导出被取消。	检查系统函数“ExportDataRecords”。
290064	报警报告：已经开始删除记录。	--
290065	报警报告：已经成功完成删除记录。	--
290066	在删除记录之前出现确认请求。	--
290068	安全性请求确认是否应删除配方中的所有记录。	--
290069	安全性请求确认是否应删除配方中的所有记录。	--
290070	指定记录不在导入文件中。	检查记录号或记录名的来源（常量或变量值）。

8.4 使用报警

编号	效果/原因	解决方法
290071	在编辑记录值期间，输入了小于配方变量下限的数值。 拒绝输入值。	输入值不能超出配方变量限制值范围。
290072	在编辑记录值期间，输入了大于配方变量上限的数值。 拒绝输入值。	输入值不能超出配方变量限制值范围。
290073	由于未知错误，操作（例如，保存记录）失败。 该错误对应于大型配方视图中的状态报警 IDS_OUT_CMD_EXE_ERR。	--
290074	在进行保存时，检测到具有指定编号的记录已存在，但是已经使用其它名称进行保存。	覆盖该记录、更改记录号或取消该操作。
290075	具有此名称的记录已经存在。 该记录未保存。	请选择不同的记录名。
290110	由于出错而无法设置默认值。	--
290111	配方子系统不能使用。配方视图无内容，因此不能执行配方相关的功能。 可能的原因： <ul style="list-style-type: none"> • 装载配方时出错。 • 在 ES 中改变了配方的结构。再次下载该项目时，配方并不随之进行传送。这意味着，新项目数据不会被传送到设备的旧配方上。 	再次将项目（包括配方）下载到设备（在对话框中选择相应选项）。

表格 8-32 300000 - Alarm_S 报警

编号	效果/原因	解决方法
300000	过程监视组态错误（如采用 PDiag 或 S7-Graph）：排队的报警比 CPU 规定的要多。PLC 无法管理更多的 ALARM_S 报警并将其报告给 HMI 设备。	更改 PLC 组态。
300001	ALARM_S 未在此 PLC 上注册。	选择支持 ALARM_S 服务的控制器。

表格 8-33 310000 - 报表系统事件

编号	效果/原因	解决方法
310000	尝试并行打印的报告太多。 一次只能输出一个日志文件到打印机；因此该打印作业被拒绝。	等待直至上一个激活的日志打印完成。 如有必要，重复打印作业。
310001	触发打印机时发生错误。报表没有打印或打印出错。	判断与该报警有关的其它系统报警。 如有必要，重复打印作业。

表格 8-34 320000 - 报警

编号	效果/原因	解决方法
320000	移动由另一个设备指示。 移动不再受控制。	在其它显示单元上取消选择移动并在所需显示单元上选择移动控制画面。
320001	网络太复杂。 无法指示故障地址。	以 STL 显示模式查看网络。
320002	没有选择诊断报警消息 (出错)。 无法选择与报警消息相关的单元。	选择来自 ZP_ALARM 报警画面的诊断报警。
320003	对于所选单元, 不存在报警消息 (错误)。该详细视图无法显示任何网络。	在总览画面中选择故障单元。
320004	PLC 无法读取所需的信号状态。无法找到故障地址。	检查 HMI 设备单元上的组态与下载的 PLC 程序之间的一致性。
320005	项目包含未安装的 ProAgent 元素。无法执行 ProAgent 诊断功能。	为运行项目, 请安装 ProAgent 选件包。
320006	您尝试执行当前组合体中不支持的功能。	检查所选单元的类型。
320007	在网络中未找到错误触发地址。 ProAgent 无法显示任何故障地址。	将细节画面切换为 STL 显示模式, 并检查地址的状态和禁用地址。
320008	保存在组态中的诊断数据与 PLC 中的不同步。 ProAgent 只能显示诊断单元。	再次将项目下载到 HMI 设备。
320009	保存在组态中的诊断数据与 PLC 中的不同步。 诊断画面可正常操作。 ProAgent 可能无法显示所有诊断文本。	再次将项目下载到 HMI 设备。
320010	组态中保存的诊断文本与 STEP7 中的不同步。 ProAgent 诊断数据不是最新的。	再次将项目下载到 HMI 设备。
320011	具有相应 DB 号和 FB 号的单元不存在。 无法执行该功能。	检查函数“SelectUnit”的参数以及项目中所选的单元。
320012	不再支持“步进序列模式”对话框。	为您的项目使用相应的标准画面中的步进序列画面 ZP_STEP。可使用 ZP_STEP 作为画面名称来调用函数“FixedScreenSelection”, 而不是调用 Overview_Step_Sequence_Mode 函数。
320014	ProAgent 无法评估所选 PLC。 找不到分配有系统函数“EvaluateAlarmDisplayFault”的报警视图。	检查系统函数“EvaluateAlarmDisplayFault”的参数。

表格 8-35 330000 - GUI 报警

编号	效果/原因	解决方法
330022	在 HMI 设备上打开了太多对话框。	关闭 HMI 设备上所有不需要的对话框。
330026	在显示的天数之后, 密码将过期。	输入新的密码。

表格 8-36 350000 - GUI 报警

编号	效果/原因	解决方法
350000	在必须的期限内，没有得到 PROFIsafe 软件包。 与 F-CPU 的通信出现问题。 结束 RT。	检查无线局域网连接。
350001	在必须的期限内，没有得到 PROFIsafe 软件包。 与 F-CPU 的通信出现问题。 重新建立了 PROFIsafe 连接。	检查无线局域网连接。
350002	产生了内部错误。 终止运行系统。	内部错误
350003	与 F-CPU 建立连接的反馈。 立即激活紧急关闭按钮。	--
350004	设置 PROFIsafe 通信，连接被清除。 可以结束运行系统。 立即禁用紧急关闭按钮。	--
350005	为 F 从站配置错误地址。 没有 PROFIsafe 连接。	在 WinCC ES 中检查并且修改 F 从站的地址。
350006	在登录之前，“Acknowledgement”和“Panic”功能中的确认按钮并未经过测试。 不能登录到有效的范围上。	依次按下“Acknowledgement”和“Panic”位置的两个确认按钮。
350008	组态了错误的故障安全按钮编号。 没有 PROFIsafe 连接。	更改项目中故障安全按钮的编号。
350009	设备处于超调模式。 由于转发器检测失败，可能无法检测位置。	退出超调模式。
350010	内部错误：设备没有故障安全按钮。	将设备返还。 世界范围内的联系人

参见

报警的系统函数和事件 (页 1144)

8.5 使用配方

8.5.1 基础知识

8.5.1.1 有关配方的基础知识

简介

配方是同一类数据的集合，如机器参数设置或生产数据。

实例：

- 将产品转化为一个不同产品变量时，所需的机器参数设置。
- 导致不同最终产品不同组合的产品组件。

配方有固定的数据结构。在组态阶段，对配方的结构进行一次性定义。一个配方中包含有多个配方数据记录。这些数据记录仅在数值方面有所不同，而非结构。配方存储在 HMI 设备或外部存储介质上。例如，如果将生产数据存储在服务器上的数据库中，则可以在运行时通过 CSV 文件导入生产数据。

注意

精简面板

无法为精简面板导出或导入配方。无外部存储介质。

配方数据记录以单步的方式在 HMI 设备和 PLC 之间完整地传送。

使用配方

在下列情况下可以使用配方：

- 手动生产
可以选择所需的配方数据并将其显示在 HMI 设备上。可以根据需要修改配方数据并将其保存在 HMI 设备上。将配方数据传送到 PLC 中。
- 自动生产
控制程序启动 PLC 和 HMI 设备之间的配方数据传送。还可以从 HMI 设备上启动传送。随后生产过程将自动进行。无需显示或修改这些数据。
- Teach-in 模式
可以优化系统中已手动优化过的生产数据，例如坐标位置或填充量。从而将确定值传送给 HMI 设备，并保存在一个配方数据记录中。以后可以将已保存的配方数据回传给 PLC。

输入并修改配方数据

可以在单个配方数据记录中输入数据，并根据需要进行修改。有下列选项：

- 组态期间的数据项

如果生产数据已存在，则可以在配方组态期间在“配方”编辑器中输入数据。

- 运行期间的数据项

如果不得不频繁地修改生产数据，可以直接在运行系统中以如下方式进行修改：

- 在 HMI 设备上直接输入数据。
- 在机器上直接设置参数。然后将数据从 PLC 传送到 HMI 设备，并保存在配方中。

参见

“配方”编辑器 (页 1178)

简单配方视图 (页 1181)

常规组态步骤 (页 1184)

简单配方视图的描述 (基本版、高级版) (页 1192)

使用配方的实例 (页 1174)

配方的结构 (页 1174)

显示配方 (页 1176)

配方的数据流 (页 1176)

使用 PLC 同步配方数据记录 (页 1177)

8.5.1.2 使用配方的实例

例如，配方用于制造业和机械工程。下面的实例说明了通过 WinCC 工程系统的配方功能可以实现的几个典型应用：

- 机械设备参数分配

配方的一个应用领域就是制造工业中机械设备参数的分配：机械设备按指定的尺寸切割木板并进行钻孔。导轨和钻子必须根据木板的尺寸向新位置移动。所需的位置数据作为数据记录存储在配方中。如果要采用新的木板尺寸，需要使用“Teach in”模式重新分配机械设备参数。将新的位置数据直接从 PLC 传送到 HMI 设备，然后将其保存为新数据记录。

- 批量生产

食品加工工业中的批量生产代表配方的另一个应用领域：果汁工厂中的配料站可以生产出不同口味的果汁、蜜露和水果饮料。它们的配料始终相同，只是混合比不同。每种口味对应于一个配方。每种混合比对应于一条数据记录。触摸按钮时，一种混合比所需的全部数据都可以传送到机械设备控制器。

参见

有关配方的基础知识 (页 1173)

8.5.1.3 配方的结构

简介

以果汁工厂中的配料站为例介绍配方的基本结构。

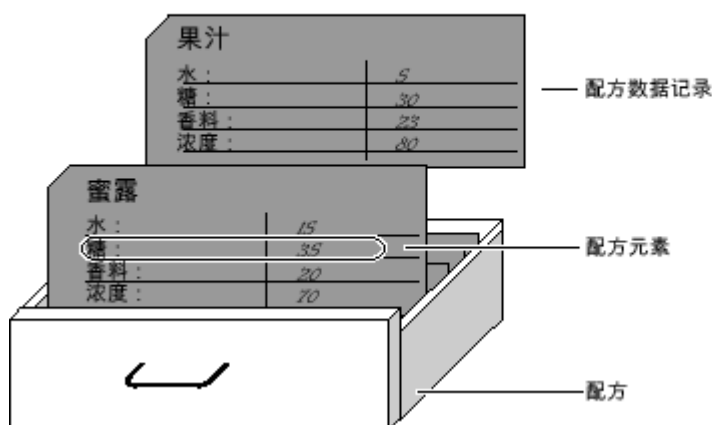
在一个 HMI 设备中可能存在多个不同的配方。配方可以比作一个包含多个索引卡的索引卡盒。对于制造一个产品系列该索引卡盒包含多个不同的索引卡。针对每样产品所需的所有数据包含在一个索引卡中。

实例：

在一个软饮料工厂中，同一个配方需要用于不同的口味。各种饮料产品包括果汁饮料、果汁和蜜露。

配方

配方包含不同种类饮料的所有配方数据记录。



配方数据记录

每个索引卡代表了一个制造一种产品所需的配方数据记录。

配方条目

在同一个抽屉中的每个索引卡都拥有相同的结构。所有的索引卡包含有用于不同配料的应用范围。每个应用范围对应于一个配方条目。因此，一个配方中的所有记录均含有相同的条目。不过，各记录中的各个条目的值并不相同。

实例：

所有的饮料都包含以下成分：

- 水
- 浓度
- 糖
- 香料

不过，在生产果汁饮料、果汁或蜜露时，其各自所对应的记录在糖的使用量方面不同。

参见

有关配方的基础知识 (页 1173)

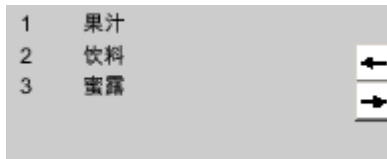
8.5.1.4 显示配方

简介

需要组态配方视图以显示配方。可在配方视图中更改配方值，以此来影响制造过程或机器。

配方视图

配方视图是现成的 WinCC 显示和操作员控件，可用于管理配方数据记录。配方视图始终是画面的一部分。配方视图以表格形式显示配方数据记录。可根据自己的特定需求来调整外观和可用操作。



如果要在项目中使用配方视图编辑配方，则值将保存在配方数据记录中。直至使用了相关的操作员控件，才开始在 HMI 设备和 PLC 之间传送这些值。

参见

有关配方的基础知识 (页 1173)

8.5.1.5 配方的数据流

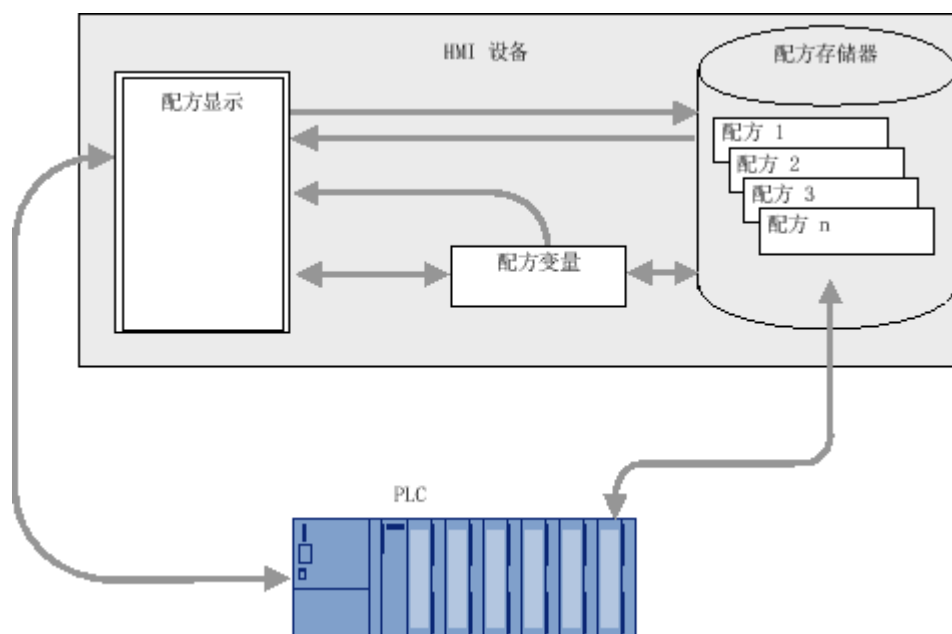
组件间的相互作用

在运行时，下列组件间会相互作用：

- 配方视图
在 HMI 设备上的配方视图中显示和编辑配方。
在配方视图中，可以显示和编辑 HMI 设备内部存储器中的配方数据记录。
- HMI 设备配方存储器
配方以配方数据记录的形式保存在 HMI 设备的配方存储器中。
- 配方变量
配方变量中包含有配方数据。

数据流概述

下图说明了配方中的数据流：



要将配方数据记录传送到 PLC，请使用配方视图中的“写入 PLC”按钮或具有系统函数“RecipeViewSetDataRecordToPLC”的操作员控件。

参见

有关配方的基础知识 (页 1173)

8.5.1.6 使用 PLC 同步配方数据记录

概述

在 HMI 设备和 PLC 之间传送配方数据记录时，两个同级通讯对象都会访问彼此的公共通讯区。配方数据记录始终直接传送。变量的值直接从已组态的地址中写入或读出，不需要放在剪贴板上。

数据传送类型

在 PLC 和 HMI 设备之间传送配方数据记录有两种方法：

- 非同步传送
- 通过“数据记录”区域指针同步传送。

注意 同步传送

同步传送用于防止控制程序中数据在任一方向上被意外覆盖。

同步传送的要求

同步传送的要求：

- 必须在“通讯”>“连接”编辑器中为所需的连接设立“数据记录”区域指针。
- HMI 在“配方”编辑器中的配方属性里指定要与 HMI 设备同步数据记录传送的 PLC。

同步传送

在同步传送的情况下，PLC 和 HMI 设备均在共享数据区中设置状态位。

同步数据记录传送是一种非常有用的解决方案，例如在以下情况下：

- 对于配方数据记录的传送，PLC 是“主动方”。
- PLC 会判断包含配方编号和名称的信息，同样也会判断配方数据记录的编号和名称。
- 配方数据记录的传送通过下列 PLC 作业启动：
 - “Set_data_record_in_PLC”
 - “Get_data_record_from_PLC”

参见

有关配方的基础知识 (页 1173)

8.5.2 元素和基本设置

8.5.2.1 “配方”编辑器

简介

可以在“配方”编辑器中创建、组态和编辑配方、配方条目和配方数据记录。也可以使用“配方”编辑器在配方数据记录中输入值。

“配方”编辑器的结构

在表编辑器的上部创建配方。也可以在这里或在巡视窗口中组态配方。

表编辑器的下部有以下选项卡：

- 元素
使用此处提供的表单元格定义所选配方的元素。
- 数据记录
使用此处提供的表单元格定义所选配方数据记录的值。



然后可以在巡视窗口中组态所选配方、配方元素或配方数据记录。您将在“组态配方”部分找到关于配方组件组态的更多注释。

配方设置

有关配方有以下设置：

设置	描述
配方名	这用于在 HMI 设备中唯一地标识配方。
显示名称	显示在运行系统中的相应位置（例如配方视图中）。显示名称可使用多种语言组态。输入可以直接分配给产品的有意义的名称或描述，例如“Fruit_juice_Orange”。
配方号	这用于在 HMI 设备中唯一地标识配方。
版本	显示关于配方的信息。配方最后的更改日期和时间默认设置。
路径	定义配方的存储位置。
信息文本	运行系统中操作员会看见的关于配方元素的信息文本。

注意 精简面板

“路径”设置不可用于精简面板。

配方元素设置

您可以在“元素”选项卡中进行以下设置：

设置	描述
配方元素名	在配方中唯一地标识配方元素。输入您可以唯一分配的有意义的名称或标签，例如机器的轴标签或配料（如“Flavoring”）。
显示名称	显示在运行系统中的相应位置（例如配方视图中）。显示名称可使用多种语言组态。输入可以直接分配给产品的有意义的名称或描述，例如“Fruit_juice_Orange”。
分配的变量	为每个配方元素分配一个在运行系统中用于存储配方数据记录值的变量。
文本列表	文本分配给文本列表中的一个值或值范围。例如，您可以在输出域中显示此文本。 分配的变量必须为“integer”型，文本列表才能在配方元素中使用。变量值必须在文本列表的取值范围之内。
默认值	在创建新配方数据记录时作为默认条目使用。
小数点	确定用于在运行系统中显示配方数据记录值的小数位数。

配方数据记录设置

可以在“数据记录”选项卡中进行以下设置：

设置	描述
配方数据记录名	在配方中唯一地标识配方数据记录。
显示名称	显示在运行系统中的相应位置（例如配方视图中）。显示名称可使用多种语言组态。输入可以直接分配给产品的有意义的名称或标签，例如“Product_numbers”。
配方数据记录号	在配方中唯一地标识配方数据记录。
输入的值	组态期间可以在配方数据记录中输入值。将项目下载到 HMI 设备时，同时会下载配方数据记录。如果 HMI 设备上已经有数据记录，则提示将会出现，并且记录将会被根据下载设置进行覆盖。
注释	关于配方数据记录的注释

参见

有关配方的基础知识 (页 1173)

8.5.3 在运行系统中显示和编辑配方

8.5.3.1 简单配方视图

配方视图

简单配方视图是现成的显示元素和操作员控件，可用于管理配方数据记录。配方视图以表格形式显示配方数据记录。

简单配方视图的操作员控件是可组态的。

在配方视图中显示和输入的值将保存在配方数据记录中。配方数据记录通过系统函数与 PLC 进行交换。

显示的布局

简单配方视图由三个区域组成：

- 配方列表
- 数据记录列表
- 元素列表

在简单配方视图中，每一区域均独立地显示在 HMI 设备上。根据组态的不同，简单配方视图以配方列表开始。

下图给出了数据记录列表的一个实例。



数值的显示

注意事项

在后台更改配方数据记录

适用于配方数据记录处理：

如果控制作业更改了配方数据记录值，配方视图不会自动更新。

若要更新配方视图，请重新选择相应的配方数据记录。

参见

有关配方的基础知识 (页 1173)

简单配方视图的组态选项 (页 1182)

运行系统中配方视图的特性 (页 1183)

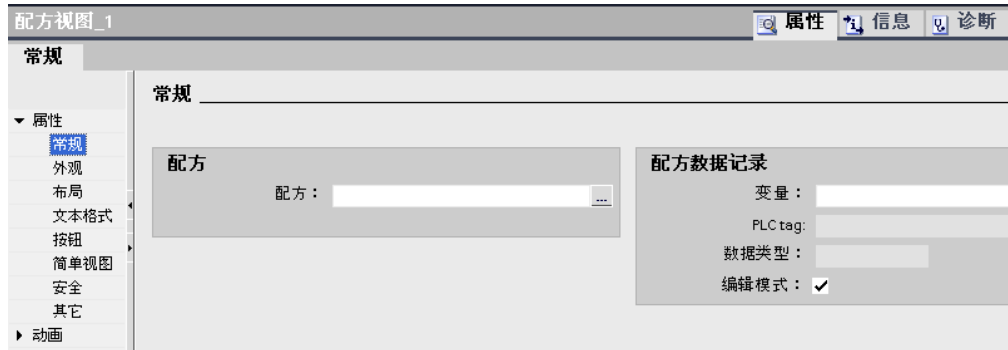
8.5.3.2 简单配方视图的组态选项

可在配方视图巡视窗口中定义配方视图的行为和显示的配方。

只显示配方数据记录值

如果只是出于检查的目的在配方视图中显示配方数据，请按如下步骤进行操作：

1. 在“常规”组中，取消选取“编辑模式”。



此时不能编辑配方数据记录。

在配方视图上组态一个事件：

要为配方视图组态事件，请按如下步骤进行操作：

1. 在“画面”编辑器中选择已添加到画面中的配方视图。
配方视图的属性即显示在巡视窗口中。
2. 在巡视窗口中，在“属性”>“事件”下单击要组态的事件。
“函数列表”对话框打开。
3. 为所选事件组态函数列表。

组态的事件发生时，将会处理函数列表。

注意

HMI 设备相关性

仅能为用于简单配方视图的 OP 77A 和 TP 177A HMI 设备组态事件。

简单配方视图的约束：

注意

无法使用脚本动态操作“简单配方视图”对象。

例如，在工程系统中，可以在巡视窗口的“动画”组中动态控制对象的可见性。在运行系统中，“简单配方视图”不支持动画。如果已组态某个动画，并希望执行某项操作（例如执行项目一致性检查），则系统会在“输出”窗口中显示一条错误报警。

简单配方视图中不包含以下功能：

- 同步变量
- 将配方和数据记录保存至一个变量或从一个变量中读出

参见

简单配方视图 (页 1181)

8.5.3.3 运行系统中配方视图的特性

画面切换

如果要切换到另一个画面，但尚未在配方视图中保存修改过的配方数据，您将被提示保存配方数据。显示配方名和配方数据记录的名称以指出尚未保存的配方数据。

切换至包含配方视图（带有加载的配方数据）的画面时，配方数据会自动更新。

创建、修改、复制或者删除配方记录

如果试图创建新的配方数据记录而配方数据记录已经存在，则画面中会出现系统报警。

使用功能键操作配方视图

可使用功能键来操作配方视图，例如，如果 HMI 设备不具有触摸功能时。可将 SaveDataRecord 等功能分配给 HMI 设备上的功能键。

在导入配方数据后显示

如果在导入配方数据过程中打开配方视图，则仅会显示已完全导入的配方数据。配方视图不会随着数据导入自动更新。要显示完整的配方数据视图，请仅在系统通知您配方数据已成功导入后打开配方视图。或者，在导入过程成功完成后更新配方视图。

注意

精简面板

配方数据导入和导出操作不可用于精简面板。

更新用于配方和配方数据记录的变量

根据组态的不同，可将当前的配方数据记录或编号保存到一个变量中。只有满足下列条件时，该变量才会被更新：

- 已加载了配方数据记录。
- 在加载期间，未退出带有配方视图的画面。

此操作可能需要一些时间。

参见

简单配方视图 (页 1181)

8.5.4 组态配方

8.5.4.1 常规组态步骤

创建新配方时执行以下组态步骤：

步骤	描述
1	定义配方的结构。
2	根据配方结构创建配方变量 为这些变量分配过程名称。
3	创建配方。
4	输入配方必需的属性： <ul style="list-style-type: none"> • 配方的存储位置 • 匹配配方变量值和配方数据记录 • 与 PLC 同步
5	创建配方元素并输入必需的属性： <ul style="list-style-type: none"> • 如有需要，为配方数据条目输入数值。 • 输入配方条目的特定语言的显示名称。 • 将配方条目连接至变量。
6	创建配方数据记录。 输入配方数据记录的特定语言的显示名称。
7	组态一个带配方视图的画面或者一个配方画面。

注意 精简面板

存储位置选择操作不可用于精简面板。配方始终存储在配方存储器中。

配方画面不可用于精简面板。

参见

有关配方的基础知识 (页 1173)

创建新配方 (页 1186)

组态简单配方视图 (页 1185)

8.5.4.2 组态配方的显示

8.5.4.2 组态简单配方视图

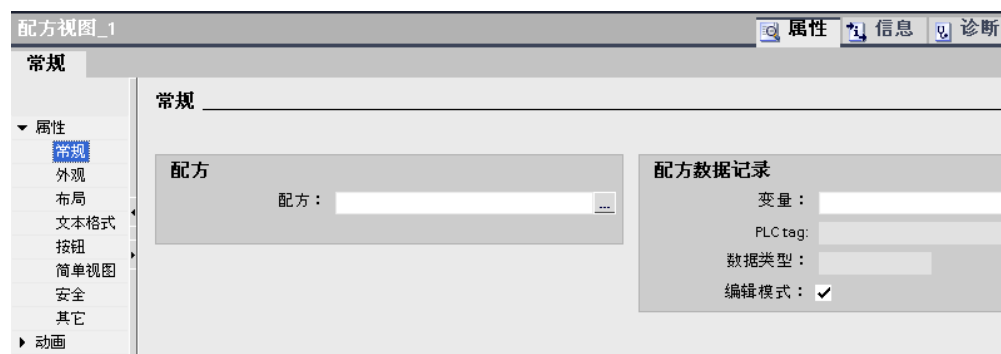
要求

- 已经创建了配方。
- 打开“画面”编辑器。
- 已创建并打开了画面。

步骤

要组态简单配方视图，请按如下步骤进行操作：

1. 将配方视图粘贴到画面中。该配方视图将出现在“工具”任务卡中的“控件”下。
2. 在“常规”>“显示类型”下选择“简单视图”。



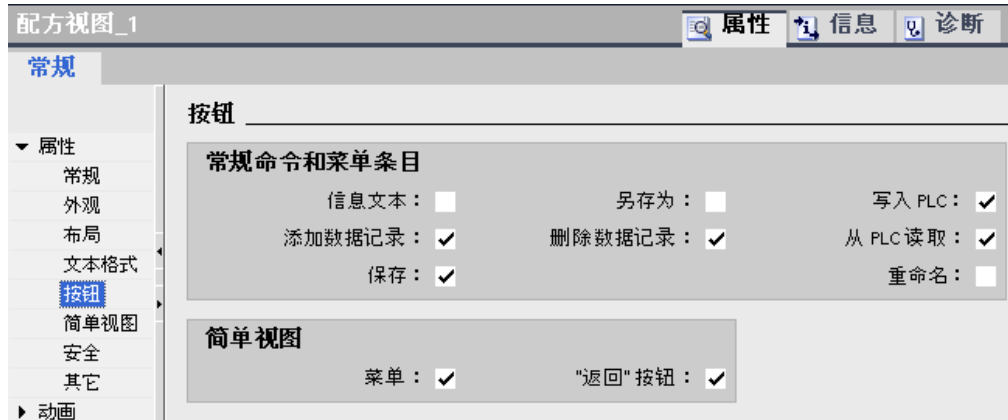
3. 如果只想在配方视图中显示特定配方的配方数据记录，则在“配方”下选择该配方。
4. 如果只想在配方视图中显示配方数据，则在“数据记录”区域中取消选取“编辑模式”。
5. 可以在“属性”>“外观”和“属性”>“布局”下为配方视图定义更多选项。
6. 选择“属性”>“简单视图”，以选择所需的位置、域长度和行数。

选择“位置”>“顶部”可在配方条目第一行显示配方值。

选择“位置”>“底部”可在配方条目最后一行显示配方值。



7. 在“属性”>“按钮”下，指定运行系统的配方视图中可使用的菜单命令。



结果

组态了一个简单的配方视图。可以使用配方视图在运行时显示和编辑配方数据。

参见

常规组态步骤 (页 1184)

8.5.4.3 创建并编辑配方

8.5.4.3 创建新配方

简介

要创建完整的配方，首先要创建一个新配方并分配相应的配方元素，然后在配方数据记录中定义相关值。

要求

- 已创建了用于配方的变量。
- 打开“配方”编辑器。

创建配方

按以下步骤创建配方：

1. 在“配方”编辑器表格的第一个空行中单击“添加”。
新配方即创建并显示在一行中。
2. 在“名称”下为配方输入描述性名称。
该名称用于在项目中明确地标识该配方。
3. 选择“显示名称”以输入要在运行系统中显示的特定语言的名称。
4. 在“编号”中选择配方号。
该编号用于在 HMI 设备中清楚标识该配方。
系统会为该配方自动分配一个版本，用以指示最后更改日期和时间。或者，也可以输入与配方相关的特定信息。
5. 在“数据介质”中指定配方数据记录的存储位置。提供的选项因所用的具体 HMI 设备而异。

注意 精简面板

“数据存储”类别不可用于精简面板。配方数据记录存储在内部闪存中。

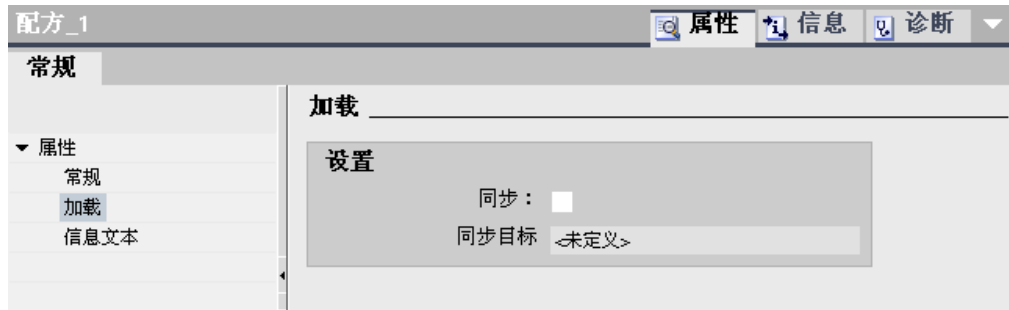
6. 输入在运行系统中操作员会看见的信息文本。
7. 要在运行系统中将在 I/O 域中组态的配方变量与配方视图进行匹配，请在巡视窗口中的“属性”>“选项”下选择“同步变量”。
8. 禁用“离线变量”以指定在编辑 I/O 域时，配方变量应被自动传送到 PLC。



注意 精简面板

因为无法通过精简面板在画面的 I/O 域中另外使用配方变量，所以“选项”类别不可用。

9. 在“属性”>“下载”下启用“同步”，以使用区域指针在运行时监视配方数据下载。
1. 在“同步...”下选择相关的 PLC。
- 0.



创建配方元素

要创建配方元素，请按如下步骤进行操作：

1. 单击“元素”选项卡。
2. 在表格编辑器的第一个空行中单击“添加”。
一个新的配方元素即创建出来。
3. 在“名称”下为元素输入描述性名称。
该名称用于在配方中唯一地标识元素。
4. 在“显示名称”下为元素输入特定语言的显示名称。
显示名称会出现在运行系统中的相应位置（例如配方视图中）。
5. 在“变量”下选择要链接到配方元素的变量。
在运行系统中，将配方数据记录的值保存在此变量中。

成分		数据记录				
	名称	显示名称	变量	缺省值	小数位数	信息文本
	水	水	HMI_Tag_1	0	0	
	<添加新许可证>					

6. 输入信息文本。
在运行系统中，这些信息文本将显示给操作员。
7. 在“默认值”下，输入创建新配方数据记录时要用作默认条目的值。
8. 要将文本分配给值或值范围，应选择此处的相关文本列表。
分配的变量必须为“integer”型，文本列表才能在配方元素中使用。变量值必须在文本列表的取值范围之内。
存储在文本列表中的文本会显示在运行系统的相应域（例如输出域）中。
9. 在“小数位数”列中设置，运行时显示的数据记录值的精确小数位数。
1. 根据需要为配方创建多个配方条目。可能的最大配方条目数目取决于所用的 HMI 设备。
- 0.

成分		数据记录			
	名称	显示名称	变量	缺省值	小数位数
	水	水	HMI_Tag_1	0	0
	酸	酸	HMI_Tag_2	0	0
	糖	糖	HMI_Tag_3	0	0
	味儿	味儿	HMI_Tag_4	...	0

以已知配方值创建配方数据记录

要创建配方元素，请按如下步骤进行操作：

1. 单击“数据记录”选项卡。
2. 在表格编辑器的第一个空行中单击“添加”。

一个新的数据记录即创建出来。配方数据记录以单独的列显示配方中创建的每个配方元素。

成分		数据记录				
	名称	显示名称	编号	水	酸	糖
	Recipe_data_rec..	Recipe_data_reco...	1	0	0	0
	<添加新许可证>					

3. 在“名称”下输入描述性的名称。
该名称用于在配方中唯一地标识数据记录。
4. 在“显示名称”下输入特定语言的名称。
显示名称会出现在运行系统中的相应位置（例如配方视图中）。
5. 在“编号”下输入配方数据记录编号。
配方数据记录编号用于在配方中唯一地标识配方数据记录。
6. 如果在组态阶段已经知道了配方的值，可以为每个配方元素输入相关的值。

成分		数据记录				
	名称	显示名称	编号	水	酸	糖
	果汁	果汁	1	30	70	45
	<添加新许可证>					

7. 根据需要为配方创建多个数据记录。

成分		数据记录				
	名称	显示名称	编号	水	酸	糖
	果汁	果汁	1	30	70	45
	蜜露	蜜露	2	50	50	45
	饮料	饮料	3	70	30	3
	<添加新许可证>					

在运行系统中输入值

下列选项可用于在运行期间在配方数据记录中输入值：

- 直接从 PLC 传送数据（Teach-in 模式）
- 从 CSV 文件中导入值

- 在 HMI 设备上输入值

注意
精简面板

导入值操作不可用于精简面板。

结果

整个配方组态完毕。

带有日期或时间标记的配方数据记录

如果在配方中指定日期或时间，下面的内容必须一致：

- 组态计算机上的时间和日期系统设置
- 目标系统中的时间和日期系统设置

下载到目标系统后，检查目标系统中配方的日期或时间戳。

参见

常规组态步骤 (页 1184)
编辑配方 (页 1190)
管理配方 (页 1191)

8.5.4.3 编辑配方

用途

如希望修正、扩展或删除配方的各个部分。

要求

- 至少已创建一个配方。
- 打开“配方”编辑器。

更改配方设置

要更改配方设置，请按如下步骤进行操作：

1. 在“配方”编辑器中选择要更改的配方。
 巡视窗口即打开。

2. 在巡视窗口中更改配方组态。

以相同方法更改配方元素和配方数据记录。

更改配方值

要更改配方值，请按如下步骤进行操作：

1. 选择要更改其值的配方。
2. 单击“数据记录”选项卡。
3. 在值列中输入新值。

添加配方元素

要在配方中添加更多配方元素，请按如下步骤进行操作：

1. 在“配方”编辑器中选择要添加更多元素的配方。
2. 单击“元素”选项卡。
3. 在第一个空白行中单击“添加”。
配方元素即创建出来。
4. 组态配方元素。

以相同方法添加配方数据记录。

参见

创建新配方 (页 1186)

8.5.4.3 管理配方

要求

- 已经创建了含配方元素和配方数据记录的配方。
- 打开“配方”编辑器。

重命名配方

对配方、配方条目和配方数据记录的内部名称和显示名称加以区分。

要重命名配方元素，请执行如下操作：

1. 选择要重命名的配方。
巡视窗口即打开。
2. 从快捷菜单中选择“重命名”命令。
3. 输入新名称。

以相同方法在相关选项卡中重命名配方元素和配方数据记录。

注意

也可以通过选择“语言支持”>“项目文本”来更改配方和配方数据记录的显示名称。此选择在已经使用多种语言进行了组态设置时适用。

复制并粘贴配方

要复制并粘贴配方，请执行如下操作：

1. 选择要复制的配方。
2. 从快捷菜单中选择“复制”命令。
3. 在表格第一个空白行中，通过快捷菜单选择“粘贴”命令。

复制的配方即粘贴到表格中。

以相同方法在相关选项卡中复制配方元素和配方数据记录。

如果具有该名称的配方数据记录已经存在，则将在所复制配方数据记录的名称后面添加一个数字。以确保每个名称的唯一性。只能在同一配方中复制或粘贴配方数据记录。

删除配方

要删除配方，请执行如下操作：

1. 选择要删除的配方。
2. 从快捷菜单中选择“删除”命令。

配方即删除。

以相同方法在相关选项卡中删除配方元素和配方数据记录。

注意

删除配方时，配方中包含的配方数据记录也同时被删除。

注意

删除某个配方元素时，配方数据记录中的相关值也会随之删除。分配的变量仍然保留。

参见

创建新配方 (页 1186)

8.5.5 在运行系统中使用配方

8.5.5.1 使用简单配方视图

8.5.5.1 简单配方视图的描述 (基本版、高级版)

布局

简单配方视图由以下显示区域组成：

- 配方列表
- 数据记录列表
- 元素列表

下面将介绍这一应用程序：



在简单配方视图中，每一区域均独立地显示在 HMI 设备上。可使用快捷菜单操作其中的每一个显示区域。

简单配方视图始终以配方列表开始。

操作

下列选项可用于根据组态使用简单配方视图：

- 创建、修改、复制或者删除配方记录
- 从 PLC 中读取配方数据记录或将其传送到 PLC

使用显示区域和快捷菜单

可在显示区域和快捷菜单之间进行切换来操作简单配方视图。

下表说明显示区域的操作信息。

按钮	按键	功能
	<Enter>	相邻的最低显示区域打开，即数据记录列表或元素列表。
	<ESC>	打开先前的显示区域。
	<Right>	打开显示区域的快捷菜单。
	<Up>/<Down>	选择上一个/下一个条目。
	<Pg Up>/<Pg Down>	将显示向上或向下移动一页。
	<Home>/<End>	选择第一个/最后一个条目。第一个/最后一个条目已选择。

下表说明快捷菜单的操作信息：

按钮	按键	功能
	<ESC>	关闭菜单。 打开显示区域。
	输入菜单命令的编号	执行相应的菜单命令。

简单配方视图的快捷菜单

可以在每个显示区域中单击



按钮来调用命令的选择列表。命令选择列表包含了当前显示区域中可用的命令。编号将分配给每个命令。当您输入这个编号是，该命令将被执行。

配方列表中的快捷菜单

菜单命令	功能
新建	为所选配方创建新的配方数据记录。 如果组态了初始值，则它会显示在输入域中。
显示信息文本	显示为简单配方视图组态的信息文本。
打开	打开所选配方的记录列表。

数据记录列表中的快捷菜单

菜单命令	功能
新建	创建新的配方数据记录。 如果组态了初始值，则它会显示在输入域中。
删除	删除显示的记录。
另存为	所选数据记录会以不同的名称保存。将打开一个对话框，可在其中输入名称。
重命名	重命名所选的数据记录。将打开一个对话框，可在其中输入名称。
打开	随即会打开所选数据记录的元素列表。
上一步	配方列表打开。

元素列表中的快捷菜单

菜单命令	功能
保存	重命名所选记录。
到 PLC	将所选数据记录的显示值从 HMI 设备传送到 PLC。
来自 PLC	在 HMI 设备的配方视图中显示 PLC 中的配方值。
另存为	该数据记录会以新的名称保存。将打开一个对话框，可在其中输入名称。
	按 <ESC> 键打开数据记录列表。

注意

HMI 设备相关性

下列菜单命令也可以在 TP 177A 和 OP 77A HMI 设备以及所有精简面板上组态：

数据记录列表中的快捷菜单

菜单命令	功能
到 PLC	将所选数据记录的显示值从 HMI 设备传送到 PLC。
来自 PLC	在 HMI 设备的配方视图中显示 PLC 中的配方值。
显示信息文本	显示为简单配方视图组态的信息文本。

元素列表中的快捷菜单

菜单命令	功能
显示信息文本	显示为简单配方视图组态的信息文本。
重命名	重命名所选的数据记录。将打开一个对话框，可在其中输入名称。
上一步	数据记录列表打开。

参见

有关配方的基础知识 (页 1173)
 管理配方数据记录 (基本版、高级版) (页 1195)
 从 PLC 读取配方数据记录 (基本版、高级版) (页 1197)
 将配方数据记录传送到 PLC (基本版、高级版) (页 1198)

8.5.5.1 管理配方数据记录 (基本版、高级版)

配方数据记录管理

下列选项可用于根据组态管理简单配方视图：

- 创建新的配方数据记录
- 复制配方数据记录
- 编辑配方数据记录
- 删除配方数据记录

创建新的配方数据记录

要创建新配方数据记录，请按如下步骤进行操作：

1. 在欲创建新配方数据记录的 HMI 设备上选择配方。
2. 从配方列表的快捷菜单中选择“新建”命令。
 将创建具有下一个可用编号的新数据记录。
 随即打开该新配方数据记录的元素列表。
3. 输入配方数据记录各元素的值。

组态数据可能已经包含了配方数据记录的默认值。

4. 从元素列表的快捷菜单中选择“保存”命令。

将打开“另存为”对话框。

5. 输入配方数据记录的名称和编号。

6. 单击“确定”按钮。

结果

新的配方数据记录将保存在所选的配方中。如果配方数据记录已经存在，一个系统报警将输出到画面上。

复制配方数据记录

要复制配方数据记录，请按如下步骤进行操作：

1. 在 HMI 设备上选择想要在其中复制现有配方数据记录的配方。
2. 在 HMI 设备上，选择要保存其副本的配方数据记录。
3. 从数据记录列表的快捷菜单中选择“另存为”命令。

将打开“另存为”对话框。配方数据记录会使用下一个空闲配方数据记录号自动编号。

4. 在名称下输入记录的名称。

5. 单击“确定”按钮。

结果

配方数据记录以新的名称保存。

修改配方数据记录

要更改配方数据记录，请按如下步骤进行操作：

1. 在欲修改一个已存的配方数据记录的 HMI 设备上选择配方。
2. 选择想要在 HMI 设备上编辑的配方数据记录。
3. 选择配方数据记录。

随即显示该配方数据记录的元素列表。

4. 使用新值替换旧值。

5. 从元素列表的快捷菜单中选择“保存”命令。

结果

修改的值应用到配方数据记录中。

删除配方数据记录

要删除配方数据记录，请按如下步骤进行操作：

1. 在 HMI 设备上选择想要从中删除现有配方数据记录的配方。
2. 选择想要在 HMI 设备上删除的配方数据记录。
3. 从数据记录列表的快捷菜单中选择“删除”命令。

4. 确认此安全提示，以删除数据记录。

结果

删除配方数据记录。

参见

简单配方视图的描述 (基本版、高级版) (页 1192)

8.5.5.1 从 PLC 读取配方数据记录 (基本版、高级版)

简介

在运行系统中，可直接在工厂中更改 HMI 设备上存储的配方值。如果在工厂中打开阀门比在配方中指定更加直接，则上述操作适用。在 HMI 设备中保存的配方数据记录值可能与 PLC 中的值不再匹配。

可以从 PLC 读取配方变量值并将其写入配方数据记录。

将读取的值写入至当前显示在 HMI 设备上的配方记录中。

步骤

要从 PLC 中读取配方数据记录，请按如下步骤进行操作：

1. 在 HMI 设备上打开配方。
数据记录列表打开。
2. 选择要将来自 PLC 的值应用到其中的配方数据记录元素列表。
3. 从元素列表的快捷菜单中选择“来自 PLC”命令。
从 PLC 中读取数值，并显示在当前配方数据记录中。
4. 如果要保存数值，请选择“保存”或“另存为”命令。

结果

从 PLC 读取值，显示在 HMI 设备上，并保存到配方数据记录中。

注意 精简面板

通过精简面板，还可以为数据记录列表组态“来自 PLC”菜单命令：这种情况下，还可以在数据记录列表中选择“来自 PLC”菜单命令。

参见

简单配方视图的描述 (基本版、高级版) (页 1192)

8.5.5.1 将配方数据记录传送到 PLC (基本版、高级版)

简介

为了使在配方视图中更改的数据记录的值生效，必须将这些值传送到 PLC。显示在配方视图中的值将始终传送到 PLC 中。

步骤

要将配方数据记录传送到 PLC 中，请按如下步骤进行操作：

1. 打开要使用的配方。
数据记录列表打开。
2. 选择要将其值传送到 PLC 的配方数据记录的元素列表。
3. 从元素列表的快捷菜单中选择“到 PLC”命令。

结果

配方数据记录的值被传送到 PLC。

注意 精简面板

通过精简面板，还可以为数据记录列表组态“传送到 PLC”菜单命令：这种情况下，还可以在数据记录列表中选择“传送到 PLC”菜单命令。

参见

简单配方视图的描述 (基本版、高级版) (页 1192)

8.6 使用用户管理

8.6.1 用户管理的应用领域

原理

访问保护用于控制对运行系统中的数据和函数的访问。此功能将防止应用程序进行未经授权的操作。在项目创建期间，与安全相关的操作已限制为特定的用户组。为此，需要设置拥有特有访问权（即所谓的权限）的用户和用户组。然后组态操作安全相关的对象所需的权限。例如，操作员仅对特定的操作员控件具有访问权限。例如，调试人员在运行系统中具有无限制的权限。

定义

在 WinCC 用户管理中集中管理用户、用户组和权限。将用户和用户组与项目一起传送到 HMI 设备。通过用户视图在 HMI 设备中管理用户和密码。

应用实例

组态“服务”权限以便仅服务技术人员有权访问组态参数。将权限分配给“服务技术人员”用户组。这就允许该组的所有成员都能设置受保护的组态参数。

小心

访问保护不保护错误的操作。您的任务是确保经过相应培训并获得授权的人员才能设计、调试、操作和维护工厂和设备。

访问保护不适用于定义日常工作以及监控其遵守情况。

参见

用户工作区域 (页 1200)

有关用户管理的基础知识 (页 1205)

具有访问保护的對象 (页 1221)

示例：组态带登录对话框的按钮 (页 1222)

用户管理的形式 (页 1199)

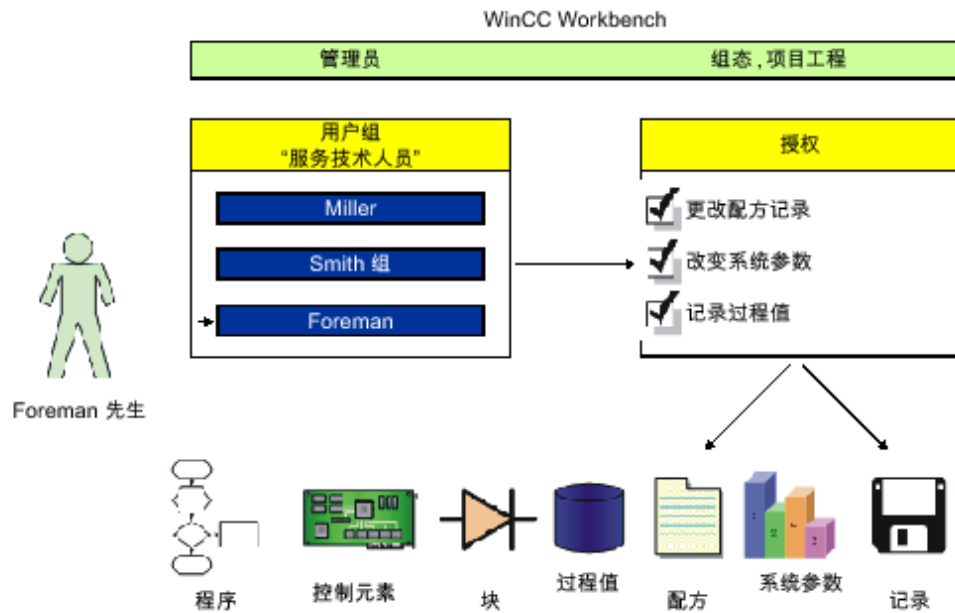
实例：用户管理的结构 (页 1224)

8.6.2 用户管理的形式

简介

对于制造业中的项目，设备制造商处的人员与终端客户（设备操作员）处的人员是有区别的。

设备制造商允许用户（例如，Mr. Foreman）在应用程序或 HMI 设备内进行特定的访问。但是，在终端客户端根本不存在用户 Foreman。另一方面，在组态期间 OEM 无法识别终端客户端的用户及其作业。在终端客户端，通常在调试后设置最终用户。



原理

为最小化管理所需的工作，将通过用户组分配权限而不是直接分配给单个用户。

用户组将根据共同的作业组合已组态的权限。例如，服务作业所需的所有权限都收集在“服务技术人员”组中。当创建应当对服务负责的用户时，只需将其分配到“服务技术人员”组。

所提供的“用户显示”用于管理运行系统中的用户；它被链接到应用程序或 HMI 设备中的组态。利用“用户显示”可创建和删除用户，并为其分配权限。

用户管理将用户的管理与权限的组态分离开来。这样可确保访问保护的灵活性。

在工程系统中的组态阶段，可为用户管理设置默认值。

参见

- 用户管理的应用领域 (页 1198)*
- 示例：组态带登录对话框的按钮 (页 1222)*
- 实例：用户管理的结构 (页 1224)*

8.6.3 元素和基本设置

8.6.3.1 用户工作区域

简介

“用户”工作区以表格形式列出用户和用户组。在此可以管理用户并将其分配给用户组。

原理

工作区包括“用户”和“组”表。

Users						
名称	密码	自动注销	退出登录时间	编号	注释	
Administrator <添加新许可证>	*****	<input checked="" type="checkbox"/>	5	1	将用户 'Administrator' 分配给 '管...	

组						
成员属于	名称	编号	显示名称	Password aging	注释	
•	Administrator group	1	管理员组	<input type="checkbox"/>	'管理员' 组初始被授予所有...	
○	Users	2	用户	<input type="checkbox"/>	'用户' 组初始被授予 '操作'...	
<添加新许可证>						

“用户”表显示已存在的用户。当在此表中选择用户时，“组”表中的“成员属于”列将显示该用户所属的用户组。

参见

- 用户管理的应用领域 (页 1198)*
- 用户组工作区域 (页 1201)*
- “用户”用户管理 (页 1202)*
- “用户组”用户管理 (页 1203)*
- 用户管理的设置 (页 1204)*
- 示例：组态带登录对话框的按钮 (页 1222)*
- 实例：用户管理的结构 (页 1224)*

8.6.3.2 用户组工作区域

简介

“用户组”工作区显示了组的列表及其权限。您可以管理用户组并为其分配权限。

原理

工作区包括“组”和“权限”表。

组					
名称	编号	显示名称	Password aging	注释	
Administrator group	1	管理员组	<input type="checkbox"/>	'管理员' 组初始被授予所有权...	
Users	2	用户	<input type="checkbox"/>	'用户' 组初始被授予 '操作' 权限。	
<添加新许可证>					

权限					
已启用	名称	显示名称	编号	注释	
<input checked="" type="checkbox"/>	操作	操作	3	'操作' 授权。	
<input checked="" type="checkbox"/>	用户管理	用户管理	1	授权 '用户管理' 在运行系统中管理用户视...	
<input checked="" type="checkbox"/>	监视	监视	2	'监视' 权限。	
<添加新许可证>					

“组”表显示已存在的用户组。在该表中选择用户组时，“权限”表的“激活”列将显示为该用户组分配的权限。

用户组和权限的数量由用户管理指定。名称和描述则由您来指定。

预定义的权限的编号是固定的。虽然应确保这些权限始终具有唯一的编号，但可自由编辑您自己所创建的权限。

参见

用户工作区域 (页 1200)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.3.3 “用户”用户管理

简介

可在“运行系统用户管理”编辑器的“用户”选项卡中创建用户，并将其分配给用户组。“用户”选项卡是 WinCC 中用户管理的一部分。

打开

要打开“用户”选项卡，请双击项目窗口中的“运行系统用户管理”。

菜单栏

菜单栏中包含操作 WinCC 所需的所有命令。

工具栏

工具栏包含菜单中最重要的命令。

工作区

在工作区中管理用户：

- 创建或删除用户。

- 将用户分配给用户组。

注意

一个用户只能分配给一个用户组。

巡视窗口

选择了一个用户后，可在“常规”组中更改密码。在“自动退出”组中，指定用户是否要从 HMI 设备自动注销。

参见

用户工作区域 (页 1200)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.3.4 “用户组”用户管理

简介

可在“用户管理”编辑器的“用户组”选项卡中创建用户和权限。“用户组”选项卡是 WinCC 中用户管理的一部分。

打开

在项目窗口中双击“用户管理”。打开“用户组”选项卡。

菜单栏

菜单栏中包含操作 WinCC 所需的所有命令。如果快捷菜单可用，则它们会显示在菜单命令旁边。

工具栏

工具栏包含菜单中最重要的命令。

工作区

在工作区中管理用户组和授权：

- 可创建新用户组和权限或删除它们。
- 将权限分配给用户组。

属性窗口

选择用户组或权限后，可在“常规”组中编辑名称。还可在“注释”组中输入简短的描述。

参见

用户工作区域 (页 1200)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.3.5 用户管理的设置

简介

在“运行系统设置”>“用户管理”编辑器中，为运行系统中的用户及其密码组态安全设置。

打开

在项目窗口中双击“运行系统设置”编辑器。单击“用户管理”。

工作区

在运行系统的工作区中对密码的有效性进行设置。例如，确定密码的复杂性。

在运行系统中的作用

根据组态情况，运行系统中的安全设置具有以下作用。

- “运行时服务”组
 - “启用登录尝试限制”复选框已选中

在“错误登录尝试的次数”域中输入的数值，可用于定义在用户被分配到“未经授权”组之前允许其进行登录尝试的次数。

“启用登录尝试限制”复选框未选中

在运行系统中，用户进行登录尝试的次数不受限制。
 - “错误登录尝试的次数”域

例如，如果您在此域中输入“4”，并且第四次登录尝试失败，则用户会被自动分配到“未经授权”组。

输入 1 到 9 次尝试。
 - “只能使用密码进行登录”复选框

如果此复选框已选中，则将通过密码来验证用户。不需要用户名。

要使用户与密码相匹配，则只能组态一次密码。
- “层级”组
 - “用户管理的组特定权限”复选框

当此复选框已选中时，管理员只能管理其组号小于或等于管理员自身组号的组。

例如，组号为 5 的管理员只能管理其组号小于或等于 5 的组。这表示，管理员只能将用户分配到组号小于或等于 5 的组中。
- “密码”组
 - “启用密码有效天数”复选框已选中

密码在经过“密码有效期（天数）”域中设置的天数后过期。

在“用户组”编辑器中选择“密码有效天数”列。这使您能够按组指定密码是否应过期，或者密码生成数是否应保存。对于其密码有效天数未启用的组来说，密码永远不会过期。
 - “预警告时间（天数）”域

在密码过期之前的指定天数通知用户密码即将过期。
 - 生成密码域

如果用户更改密码，则新密码必须不同于先前指定数量的密码。密码生成数范围为 1 至 5。

- “密码复杂性”组
 - “必须包含特殊字符”复选框已选中
在任何情况下，用户都必须输入至少包含一个特殊字符的密码。
 - “必须包含数字”复选框已选中
用户输入的密码必须包含至少一个数字（可位于任意位置处）。
 - “最小密码长度”域
用户必须输入一个最小长度为“最小密码长度”域中指定的长度的密码。
密码的最小长度为 3 个字符。

参见

用户工作区域 (页 1200)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.4 设置用户管理

8.6.4.1 有关用户管理的基础知识

原理

本节将介绍四个目标组。相应地组织主题。目标组分别用于使用用户管理的不同组。

1. 管理员 OEM
2. 管理员 RT
3. 设计者
4. 操作员

作为管理员 OEM，您可在工程系统（例如，设备生产商的工程系统）中创建运行系统的用户组、用户和权限。

作为管理员 RT，您可以在运行系统中通过“用户视图”管理用户。

作为项目工程师，您可将权限分配给工程系统中的用户组。此外，还可为对象组态权限。

作为操作员，您可以登录到运行系统中。您只有拥有所需的权限，才可以访问受保护的對象。

注意

管理员 RT 目标组作为预定义的用户组“管理员组”已经存在于运行系统的用户管理中。为了更好地理解，下面将不使用预定义的用户组和权限。

参见

用户管理的应用领域 (页 1198)
创建权限 (页 1206)
运行系统中的用户 (页 1212)
访问保护 (页 1221)
示例：组态带登录对话框的按钮 (页 1222)
实例：用户管理的结构 (页 1224)

8.6.4.2 管理运行系统用户

8.6.4.2 创建权限

简介

创建权限并将其分配给一或多个用户组。

要求

“用户组”工作区打开。

步骤

1. 在“权限”表格中双击“添加...”。
2. 输入“归档数据”作为权限的名称。
3. 输入简短的描述作为“注释”。

参见

有关用户管理的基础知识 (页 1205)
创建用户组 (页 1206)
分配权限 (页 1207)
创建用户 (页 1208)
将用户分配到用户组 (页 1209)
用户管理 (页 1209)
管理用户组 (页 1211)
示例：组态带登录对话框的按钮 (页 1222)
实例：用户管理的结构 (页 1224)

8.6.4.2 创建用户组

简介

创建用户组，这样就不需要将权限分配给每个单个用户。创建用户组，分配权限，然后将用户分配给用户组。

注意

用户组的名称在项目中必须唯一。否则，输入将不被接受。

要求

“用户组”工作区打开。

步骤

1. 在“组”表格中双击“添加...”。
2. 输入“操作员”作为用户组的名称。
3. 输入简短的描述作为“注释”。

注意

用户组的名称与语言相关。以多种语言指定该名称并于运行时在语言之间进行切换。

参见

创建权限 (页 1206)
示例：组态带登录对话框的按钮 (页 1222)
实例：用户管理的结构 (页 1224)

8.6.4.2 分配权限

简介

为用户组分配权限时，所有分配的用户都拥有此权限。

要求

- “归档数据”权限已创建。
- “操作员”用户组已创建。
- “用户组”工作区打开。

步骤

1. 在“组”表中单击“操作员”用户组。“权限”表格显示所有权限。
2. 在“权限”表格中激活“归档数据”权限。

注意事项

“归档数据”权限只是一个名称，与函数“Archiving”没有任何关系。您必须自己建立该关系。为此，可在控制按钮上组态系统函数“StartArchiving”，并选择“归档数据”作为“权限”。

参见

创建权限 (页 1206)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.4.2 创建用户

简介

创建用户，这样在传送项目后用户可在运行系统中以此用户名登录。

或者，也可以通过“用户视图”在运行系统中创建和更改用户。

当在登录时输入的用户名与运行系统中的用户匹配时，登录将会成功。此外，输入的密码也必须与该用户的存储密码一致。

注意

在具有专业版运行系统的 HMI 设备上，还要注意大写。

注意

用户名必须在项目中唯一。否则，输入将不被接受。


注意事项

要使所创建的用户拥有权限，必须将其分配到用户组，然后为该用户组分配权限。

要求

“用户”工作区域打开。

步骤

1. 在“用户”表中双击“添加...”。
2. 输入“Foreman”作为用户名。
3. 在“密码”列中单击

按钮。显示用于输入密码的对话框。
4. 输入用户的密码。
5. 要确认密码，请在下面的域中再次输入密码。
6. 使用



图标关闭此对话框。

7. 如果用户将在特定时间段之后注销，则激活“自动注销”。
8. 单击“注销时间”列。“注销时间”的预设值为 5 分钟。
9. 输入简短的描述作为“注释”。

参见

创建权限 (页 1206)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.4.2 将用户分配到用户组

简介

将用户分配到用户组时，该用户将拥有该用户组的权限。

注意

一个用户只能分配给一个用户组。在一致性检查和生成项目期间会对分配情况进行检查。

要求

- “Foreman”用户已创建。
- “操作员”用户组已创建。
- “用户”工作区域打开。

步骤

1. 在“用户”表中单击“Foreman”用户。“组”表格显示所有用户组。
2. 在“组”表格中激活“操作员”用户组。

参见

创建权限 (页 1206)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.4.2 用户管理

简介

在工作区域中，可以管理用户，并将其分配给用户组。

要求

“用户”工作区域打开。

更改用户的名称

1. 在“用户”表格中，双击“名称”列中的域以更改用户名。
2. 更改用户名。
3. 按下 <Return> 确认输入。

或者，也可以在工作区中选择用户；然后，该用户的属性将显示在“巡视”窗口中。在“属性”选项卡中，选择“常规”组。在“名称”域中更改用户名称。

更改用户密码

1. 在“用户”表格的“密码”列中，单击



按钮。用于输入密码的对话框将打开。

2. 在“输入密码”域中，输入新密码。
3. 在“确认密码”域中再次输入新密码。
4. 按下 <Return> 确认输入。

或者，也可以在工作区中选择用户；然后，该用户的属性将显示在“巡视”窗口中。在“属性”选项卡中，选择“常规”组。在“密码”区域中更改密码。

显示不可见的列

1. 将鼠标光标置于“用户”表的标题上。
2. 单击鼠标右键打开快捷菜单，然后进行诸如启用“退出时间”列的显示之类的操作。

更改用户的退出时间

1. 在“用户”区域中，双击“退出时间”列中的域以更改退出时间。
2. 更改退出时间。
3. 按下 <Return> 确认输入。

或者，也可以在工作区中选择用户；然后，该用户的属性将显示在“巡视”窗口中。在“属性”选项卡中，选择“自动退出”组。在“退出时间”域中更改退出时间。

删除用户

1. 选择要删除的用户行。
2. 使用鼠标右键打开快捷菜单，并选择“删除”命令。

注意

预定义的用户不能被删除。

参见

创建权限 (页 1206)
示例：组态带登录对话框的按钮 (页 1222)
实例：用户管理的结构 (页 1224)

8.6.4.2 管理用户组

简介

可在工作区域中管理用户组以及分配要在运行系统中使用的权限。

要求

“用户组”工作区打开。

更改用户组的名称

1. 在“组”表格中，双击“名称”列中的域以更改用户组的名称。
2. 更改用户组的名称。
3. 按下 <Return> 确认输入。

或者，也可以在工作区中选择用户组；然后，该用户组的属性将显示在“属性”窗口中。在“属性”窗口中，选择“常规”组。在“名称”域中更改名称。

注意

预定义的用户组不能删除。

显示不可见的列

1. 将鼠标光标置于“用户”表的标题上。
2. 单击鼠标右键打开快捷菜单，然后进行诸如启用“显示名称”列的显示之类的操作。

更改用户组的显示名称

1. 在“组”表格中，双击“显示名称”列中的域以更改用户组的显示名称。
2. 改变用户组的显示名称。
3. 按下 <Return> 确认输入。

或者，也可以在工作区中选择用户组；然后，该用户组的属性将显示在“属性”窗口中。在“属性”窗口中，选择“常规”组。在“显示名称”域中更改显示名称。

删除用户组

1. 标记要删除的用户组行。
2. 使用鼠标右键打开快捷菜单，并选择“删除”命令。

注意

预定义的用户组不能删除。

更改权限的名称

1. 在“权限”表格中，双击“名称”列中的域以更改权限的名称。
2. 更改权限名称。
3. 按下 <Return> 确认输入。

或者，也可以在工作区中选择权限，该权限的属性将显示在属性视图中。在“属性”窗口中，选择“常规”组。在“显示名称”域中更改权限名称。

删除权限

1. 标记要删除的权限行。
2. 使用鼠标右键打开快捷菜单，并选择“删除”命令。

注意

预定义的权限不能删除。

参见

创建权限 (页 1206)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.4.3 在运行系统中管理用户

8.6.4.3 运行系统中的用户

原理

在工程系统中创建用户和用户组，并为其分配权限。可为对象组态权限。在下载至 HMI 设备后，所有组态了权限的对象都会得到保护，以免在运行时受到未授权的访问。

用户视图

如果在工程系统中组态了用户视图，则在下载到 HMI 设备后可在此用户视图中管理用户。

小心

在用户视图中的更改将在运行系统中立即生效。在运行系统中进行的更改不会更新到工程系统中。在从工程系统下载到 HMI 设备时，用户视图中的所有更改将在用户提示后根据下载设置被覆盖。

有些 HMI 设备不支持用户视图。它们仅支持“登录”和“退出”功能：只有用户“管理员”可以登录和退出。“管理员”只能分配到用户组“管理员组”。

导出和导入用户数据

通过系统函数将 HMI 设备上的现有用户及密码导出，然后导入到其它的操作面板。这可以确保不同 HMI 设备的用户管理具有相同的状态。

参见

有关用户管理的基础知识 (页 1205)
用户视图 (页 1213)
组态用户视图 (页 1215)
创建用户 (页 1216)
在运行系统中管理用户 (页 1217)
解锁用户 (页 1219)
以用户身份登录 (页 1219)
示例：组态带登录对话框的按钮 (页 1222)
实例：用户管理的结构 (页 1224)

8.6.4.3 用户视图

用途

在工程系统中组态用户视图，以便还可以在运行系统中管理用户。

简介

在工程系统中创建用户和用户组，并将其下载到 HMI 设备。具有“用户管理”权限的用户对用户视图具有无限制的访问权限。这允许他们管理所有用户。所有其他用户只拥有对用户视图的有限访问权限，因此只能进行自我管理。

小心

在用户视图中的更改将在运行系统中立即生效。在运行系统中进行的更改不会更新到工程系统中。

在将用户管理下载到 HMI 设备时，用户视图中的所有更改都会被覆盖。

结构

用户视图的每一行将显示下列内容：

- 用户
- 加密密码
- 相应的用户组
- 退出时间



注意

如果没有用户登录，则用户视图为空。登录后会显示各个域的内容。

管理员的用户视图

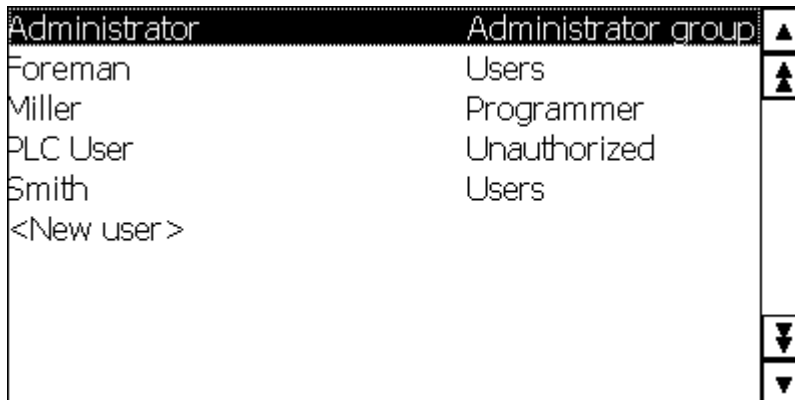


图 8-1

管理员登录时，用户视图中显示所有的用户。管理员可以更改用户名和密码。还可以创建新用户，并将其分配到现有的用户组。

用户的用户视图

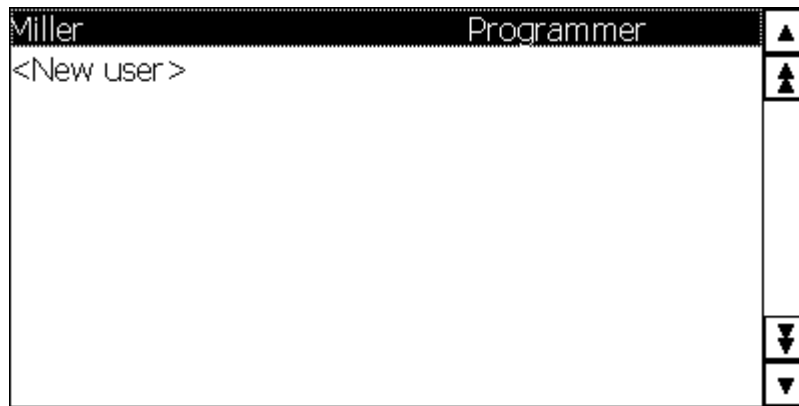


图 8-1

如果没有管理员登录，则用户视图仅显示登录的用户。用户可更改自己的密码。用户必须以管理员的身份登录才能更改名称。

参见

运行系统中的用户 (页 1212)
示例：组态带登录对话框的按钮 (页 1222)
实例：用户管理的结构 (页 1224)

8.6.4.3 组态用户视图

简介

在工程系统中组态用户视图，以便还可以在运行系统中管理用户。

要求

已经创建一个画面。

步骤

1. 从工具箱的“控件”类别中选择“用户视图”对象。
2. 将“用户视图”对象拖放到画面中。
3. 在“用户视图”的属性窗口中单击“常规”组。
4. 指定“用户视图”的外观。例如，选择“行数”和“字体”。

参见

运行系统中的用户 (页 1212)
示例：组态带登录对话框的按钮 (页 1222)
实例：用户管理的结构 (页 1224)

8.6.4.3 创建用户

简介

在运行系统中创建一个用户，以便用户可以用其用户名称登录。

或者，也可以在工程系统中创建用户，并将其下载到 HMI 设备。

只有当登录时输入的用户名与运行系统中的用户匹配时，登录才会成功。此外，登录时输入的密码也必须与运行系统中的用户的密码匹配。

注意

在具有专业版运行系统的 HMI 设备上，还要注意大写。

将用户分配到用户组。于是，用户便拥有了该用户组的权限。

注意事项

运行系统用户必须分配给用户组。用户组是在工程系统中创建的。用户组的名称与语言相关。

要求

- 用户视图打开。
- 用户组“组 2”已创建。

步骤

1. 单击用户视图中的“<新建用户>”。一个对话框随即打开。
2. 输入“Foreman”作为用户名。
3. 按下 <Return> 键。
4. 单击“密码”。
5. 输入用户的密码。
6. 按下 <Return> 键。密码是隐藏的。
7. 单击“组”列。
8. 选择“组 2”作为“组”。
9. 按下 <Return> 键。
- 1 单击“退出时间”列。
- 0.
- 1 输入用户自动退出的时间。
- 1.

参见

运行系统中的用户 (页 1212)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.4.3 在运行系统中管理用户

简介

如果您已经在工程系统中组态了用户视图，则可在运行系统中管理用户和用户组。

小心

在用户视图中进行的更改将在运行系统中立即生效。在运行系统中进行的更改不会更新到工程系统中。在从工程系统下载到 HMI 设备时，用户视图中的所有更改将在用户提示后根据下载设置被覆盖。

要求

- 已激活运行系统。
- 用户视图打开。
- 您拥有预定义的“用户管理”权限。

注意事项

如果您没有“用户管理”权限，则只能更改您自己的密码和注销时间。

更改用户的名称

1. 在用户显示的“用户”列中输入一个新用户名。
2. 按下 <Return> 确认输入。

注意事项

然后，用户便无法使用运行系统中先前的密码进行登录。如果删除名称并按 <Return> 键，则用户将被删除。

在基本用户显示中更改用户密码

1. 在用户显示的“密码”列中输入一个新密码。
2. 按下 <Return> 确认输入。

注意事项

然后，用户便无法使用运行系统中先前的密码进行登录。

如果在基本用户视图中删除密码并按 <Return> 键，则消息将会生成。该消息指明密码不在定义的限制内。

在复杂用户显示中更改用户密码

复杂用户显示仅在所选设备中提供。

1. 在用户显示的“密码”列中输入一个新密码。
2. 按下 <Return> 确认输入。

注意事项

然后，用户便无法使用运行系统中先前的密码进行登录。

如果在复杂用户视图中删除密码并按 <Return> 键，则用户将被删除。

更改用户的退出时间

1. 在用户显示的“注销时间”列中输入一个新注销时间。
2. 按下 <Return> 确认输入。

删除用户

1. 单击要删除的用户名称。
2. 删除名称。
3. 按下 <Return> 键。

注意事项

用户不能再登录到运行系统。

将用户分配到不同的用户组

1. 激活相应用户的用户组域。
2. 选择一个用户组。
3. 按下 <Return> 确认选择。

参见

运行系统中的用户 (页 1212)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.4.3 解锁用户

启用锁定用户

激活“运行系统设置”>“用户管理”中的“激活登录尝试次数限制”复选框。
在“无效登录尝试次数”域中输入数字 3。

如果用户尝试登录三次都宣告失败（例如，输入不正确的密码），则会将其分配到“未经授权”组中。该用户将丧失所有权限。该用户仍然可以登录，但不再拥有任何权限。只有拥有管理员权限的用户才能将未经授权的用户重新分配到用户组。

参见

运行系统中的用户 (页 1212)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.4.3 以用户身份登录

简介

通常情况下，通过一个特定的按钮以用户身份登录。为此，将显示登录对话框。

操作有访问保护的按钮时，系统会在 OP 73micro、TP 177micro、OP 73、OP 77A 和 TP 177A HMI 设备上始终打开登录对话框：

注意

默认情况下，访问受保护对象时，如果出现下列情况将显示登录对话框：

- 在运行系统中没有用户登录。
- 登录用户没有所需权限。

要求

- 在“运行系统设置”>“用户管理”下，
 - 已选中“启用登录尝试次数限制”复选框。
 - 在“无效登录尝试次数”域中输入数字 3。
- 已对名为“登录”的按钮组态了“ShowLogonDialog”系统函数。

步骤

1. 单击“登录”按钮。将显示登录对话框。

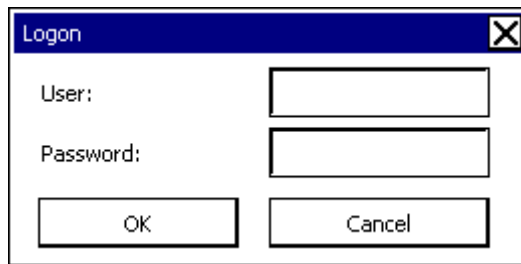


图 8-1

2. 输入您在用户管理中指定的用户名，例如“Foreman”。
如果在您之前已有用户登录，则会显示其用户名。
3. 输入相应的密码。该输入为隐藏。
4. 单击“确定”关闭该对话框。

登录成功

如果输入用户名“Foreman”，并且输入的密码与存储的密码相符，则在运行系统中以用户“Foreman”的身份登录。您已拥有用户“Foreman”的权限。

当用户“Foreman”访问受保护对象时，只有在用户“Foreman”拥有所需权限的情况下，才会授予对此受保护对象的访问权限。才会立即执行所组态的函数。

如果您在成功登录后没有所需权限，则将显示相应的错误消息。但是，您仍然在运行系统中登录。

登录未成功

显示一条出错消息。

为了保持安全性，您或在您之前登录的用户不再拥有任何权限。然而仍然可以访问不受保护的對象。但是，用户视图将不显示任何条目。在下次成功登录后用户视图和权限将会改变。

如果第三次登录尝试失败，则会将用户分配到“未经授权”组。因此，不要组态具有此显示名称的用户组。

如果调用“Log off”函数或用户的退出时间到期，则用户将退出。

参见

运行系统中的用户 (页 1212)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.4.4 组态访问保护

8.6.4.4 访问保护

简介

您可以为一个对象组态授权以保护对它的访问。然后，所有拥有此权限的登录用户便可访问该对象。如果用户没有操作对象的权限，则登录对话框将会自动显示。

注意

在“用户管理”下提供多个系统函数，以使用户能够在诸如 PLC 中编辑用户、密码和用户组。

参见

有关用户管理的基础知识 (页 1205)
具有访问保护的對象 (页 1221)
示例：组态带登录对话框的按钮 (页 1222)
实例：用户管理的结构 (页 1224)

8.6.4.4 具有访问保护的對象

简介

可为下列对象组态权限：

- 日期/时间域
- I/O 域
- 图形 I/O 域
- 配方视图
- 开关
- 按钮
- 符号 I/O 域

参见

用户管理的应用领域 (页 1198)
示例：组态带登录对话框的按钮 (页 1222)
实例：用户管理的结构 (页 1224)

8.6.5 参考

8.6.5.1 具有访问保护的對象

简介

可为下列对象组态权限：

- 日期/时间域
- I/O 域
- 图形 I/O 域
- 配方视图
- 开关
- 按钮
- 符号 I/O 域

参见

用户管理的应用领域 (页 1198)

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.6 实例

8.6.6.1 示例：组态带登录对话框的按钮

任务

在下面的实例中，将对一个按钮组态函数“ShowLogonDialog”。这样，不同的用户可以在换班时登录到运行系统中。在该过程中，先前登录的用户退出。

注意

默认状态下，登录对话框在运行时不显示，除非访问受保护的對象，或者没有用户登录或登录的用户没有所需权限。

要求

- 已经创建一个画面。
- 画面中已创建了按钮。

步骤

1. 单击画面中的按钮。
2. 在巡视窗口的“属性”选项卡的“事件”组中单击“释放”。
3. 在“函数列表”表格中的第一行上，单击“无函数”条目。
4. 从“用户管理”组选择系统函数“ShowLogonDialog”。

结果

如果用户在运行系统中单击该按钮，则调用函数“ShowLogonDialog”。当调用函数“ShowLogonDialog”时，将显示登录对话框。用户以其用户名和密码登录。

参见

用户管理的应用领域 (页 1198)

实例：用户管理的结构 (页 1224)

实例：记录登录和退出事件 (页 1223)

8.6.6.2 实例：记录登录和退出事件

任务

在下列实例中将函数“TraceUserChange”组态到事件“用户切换”。

原理

用户登录或退出时调用函数“TraceUserChange”。在调用函数时，将输出带有相关用户信息的系统消息。

可以归档该系统消息。在归档时，系统消息被加上日期标志和时间标志。这就确保了可以对登录 HMI 设备的用户、登录时间以及登录持续时间进行跟踪。

要求

- 已创建装有高级版运行系统的 HMI 设备。

步骤

1. 在项目视图中双击“调度程序”。
2. 在活动域的表格中双击“添加...”。
3. 输入“登录协议”作为“名称”。
4. 选择“用户切换”作为“事件”。
5. 在“函数列表”表格中的第一行上，单击“无函数”条目。
6. 从“用户管理”组选择系统函数“TraceUserChange”。

结果

在用户登录或退出时，输出系统消息。

参见

示例：组态带登录对话框的按钮 (页 1222)

实例：用户管理的结构 (页 1224)

8.6.6.3 用户管理实例

8.6.6.3 实例：用户管理的结构

任务

在下面的实例中，为不同用户和用户组建立用户管理。本实例针对制造工程业的典型要求配置文件。

原理

在设备或项目中包含了完全不同的组。每组人员都保护其自身数据和函数不被他人访问。为此，创建用户并将其分配到用户组。

通过用户组可以产生不同的视图。

实例：

- 组织视图：调试人员、操作员、轮班 I、轮班 II
- 技术视图：坐标轴控件、工具转换器、北面设备、南面设备

下面的实例针对组织视图。

每个用户组都具有与访问保护相关的特殊要求：对于指定应用事例，用户组具有操作权限。例如，程序员改变配方数据记录。

在本实例中，创建用户 Miller、Group Smith 和 Foreman 并分配给不同用户组。

Miller 女士是工程系统的程序员。Group Smith 是调试人员组。Mr. Foreman 是操作员。

要求

- 已经创建了新项目。
- “运行系统用户管理”编辑器打开。

步骤概述

使用用户管理的实例中含有以下步骤：

1. 创建权限：设计者指定访问保护所需的权限。
2. 组态权限：设计者指定可以操作哪些对象，可以执行哪些函数。
3. 创建用户组并分配权限：管理员与设计者一起创建用户组。设计者使用权限指定可操作对象和更改参数的用户。
4. 创建用户并将其分配到用户组：管理员管理用户。

结果

最终将获得以下用户、用户组和权限的用户管理结构：

用户			用户组	授权			
Miller	Smith	Foreman	身份	更改配方记录	更改系统参数	更改过程参数	管理
			管理员组				x
X			程序员	X			
	X		调试工程师	X	X	X	
		X	操作员	x			

属于“操作员”用户组的用户“Foreman”具有对组态的“至配方视图”按钮的访问权限。

注意

也可以创建多个具有不同操作权限的“操作员”用户，例如“操作员级别 1”、“操作员级别 2”。

参见

- 示例：组态带登录对话框的按钮 (页 1222)
- 实例：创建和组态权限 (页 1225)
- 实例：组态具有访问保护的按钮 (页 1226)
- 实例：创建用户组并分配权限： (页 1227)
- 实例：创建用户并分配到用户组 (页 1229)
- 实例：记录登录和退出事件 (页 1223)

8.6.6.3 实例：创建和组态权限

任务

下例说明如何创建权限

步骤

1. 打开“用户组”工作区。
2. 在“权限”表格中双击“添加...”。
3. 输入“更改配方数据记录”作为权限的“名称”。
4. 重复步骤 2 和 3 创建其它权限：“更改系统参数”、“更改过程参数”。

结果

Users 用户组					
组					
	名称	编号 ▲	显示名称 ▼	Password aging	注释
	Administrator group	1	管理员组	<input type="checkbox"/>	'管理员' 组初始被授予所有权...
	Users	2	用户	<input type="checkbox"/>	'用户' 组初始被授予 '操作' 权限。
	<添加新许可证>				
权限					
	已启用	名称	显示名称	编号	注释
	<input checked="" type="checkbox"/>	操作	操作	3	'操作' 授权。
	<input checked="" type="checkbox"/>	用户管理	用户管理	1	授权 '用户管理' 在运行系统
	<input checked="" type="checkbox"/>	监视	监视	2	'监视' 权限。
	<input checked="" type="checkbox"/>	更改配方数据记录	更改配方数据记录	4	
	<input checked="" type="checkbox"/>	更改系统参数	更改系统参数	5	
	<input checked="" type="checkbox"/>	更改过程参数	更改过程参数	6	
		<添加新许可证>			

参见

实例：用户管理的结构 (页 1224)

8.6.6.3 实例：组态具有访问保护的按钮

任务

在下例中，将使用系统函数创建用于更改画面的按钮。保护“至配方视图”按钮防止未经授权的操作。为此，在“至配方视图”按钮组态“更改配方数据记录”权限。

要求

- 已创建“更改配方数据记录”权限。
- 已创建“配方”画面。
- 已创建并打开“启动”画面。
- 已在“启动”画面中创建和标记了按钮。

步骤

1. 选择属性窗口中的“属性”组，然后单击“常规”。
2. 输入“至配方视图”作为文本。
3. 单击“属性”视图的“事件”组中的“单击”。
4. 在“函数列表”表格中的第一行上，单击“无函数”条目。
5. 从“画面”组中选择“ActivateScreen”系统函数。
6. 在“画面名称”域中单击



按钮。用于选择画面的对话框打开。

7. 选择“配方”画面，然后使用



按钮关闭对话框。

8. 在属性视图中选择“属性”组，然后单击“安全性”。
9. 选择“更改配方数据记录”作为“权限”。

结果

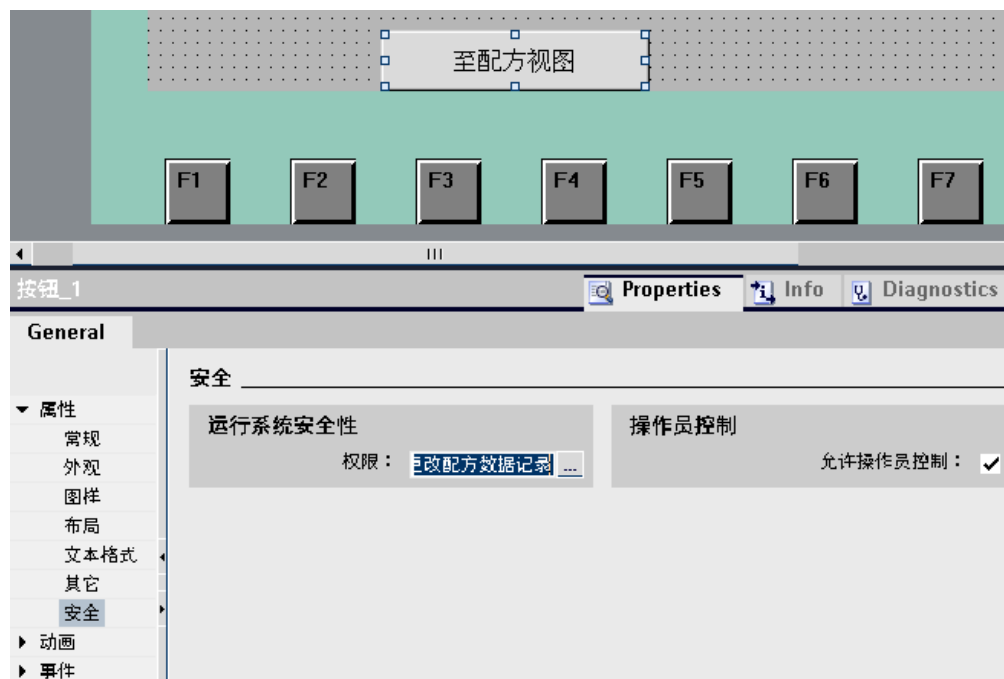


图 8-1

对“至配方视图”按钮的访问进行保护。例如，如果用户“Smith”单击运行系统中的按钮，则调用函数“配方视图”画面。前提条件是用户“Smith”已正确登录，并拥有所需要的权限。“配方”画面包含配方视图及其它画面对象。

如果已登录的用户不具有所需权限，或没有用户登录，则显示“登录对话框”。

参见

实例：用户管理的结构 (页 1224)

8.6.6.3 实例：创建用户组并分配权限：

任务

在下面的实例中，将创建用户组，并为其分配权限。

步骤

1. 打开“用户组”工作区。
2. 在“组”表格中双击“添加...”。
3. 输入“程序员”作为“名称”。
4. 重复步骤 2 和 3 创建“调试员”和“操作员”用户组。
5. 在“组”表格中单击“管理员”。
6. 在“权限”表格中激活“更改系统参数”权限。

中间结果

组					
名称	编号	显示名称	Password aging	注释	
Administrator group	1	管理员组	<input type="checkbox"/>	'管理员' 组初始被授予所有...	
Users	2	用户	<input type="checkbox"/>	'用户' 组初始被授予 '操作' 权...	
Programmer	3	组_1	<input type="checkbox"/>		
Commissioner	4	组_1	<input type="checkbox"/>		
Operator	5	组_1	<input type="checkbox"/>		
<添加新许可证>					

权限					
已启用	名称	显示名称	编号	注释	
<input checked="" type="checkbox"/>	操作	操作	3	'操作' 授权。	
<input checked="" type="checkbox"/>	用户管理	用户管理	1	授权 '用户管理' 在运行系统...	
<input checked="" type="checkbox"/>	监视	监视	2	'监视' 权限。	
<input type="checkbox"/>	更改配方数据记录	更改配方数据记录	4		
<input checked="" type="checkbox"/>	更改系统参数	更改系统参数	5		
<input type="checkbox"/>	更改过程参数	更改过程参数	6		
<添加新许可证>					

图 8-1

步骤

1. 在“组”表格中单击“操作员”。
2. 在“权限”表格中激活“更改配方数据记录”权限。
3. 在“组”表格中单击“调试员”。
4. 在“权限”表格中激活权限“更改配方数据记录”、“更改系统参数”和“更改过程参数”。
5. 在“组”表格中单击“程序员”。
6. 在“权限”表格中激活“更改配方数据记录”权限。

结果

Users 用户组					
组					
	名称	编号 ▲	显示名称	Password aging	注释
	Administrator group	1	管理员组	<input type="checkbox"/>	'管理员' 组初始被授予所有权...
	Users	2	用户	<input type="checkbox"/>	'用户' 组初始被授予 '操作' 权...
	Programmer	3	组_1	<input type="checkbox"/>	
	Commissioner	4	组_1	<input type="checkbox"/>	
	Operator	5	组_1	<input type="checkbox"/>	
	<添加新许可证>				
权限					
	已启用	名称	显示名称	编号	注释
	<input checked="" type="checkbox"/>	操作	操作	3	'操作' 授权。
	<input type="checkbox"/>	用户管理	用户管理	1	授权 '用户管理' 在运行系统...
	<input type="checkbox"/>	监视	监视	2	'监视' 权限。
	<input checked="" type="checkbox"/>	更改配方数据记录	更改配方数据...	4	
	<input type="checkbox"/>	更改系统参数	更改系统参数	5	
	<input type="checkbox"/>	更改过程参数	更改过程参数	6	
		<添加新许可证>			

参见


实例：用户管理的结构 (页 1224)

8.6.6.3 实例：创建用户并分配到用户组

任务

在下面的实例中，创建用户并为其分配用户组。在输入名称后，立即按字母顺序排序用户。

步骤

1. 打开“用户”工作区。
2. 在“用户”表格中双击“添加...”。
3. 输入“Miller”作为用户名。
4. 在“密码”列中单击  按钮。显示用于输入密码的对话框。
5. 输入“miller”作为密码。
6. 要确认密码，请在下面的域中再次输入密码。
7. 使用



图标关闭此对话框。

- 在“组”表格中激活“程序员”用户组。



中间结果

Users						
名称	密码	自动注销	退出登录时间	编号	注释	
Administrator	*****	<input checked="" type="checkbox"/>	5	1	将用户 'Administrator'...	
Miller	*****	<input checked="" type="checkbox"/>	5	2		
<添加新许可证>						

组						
成员属于	名称	编号	显示名称	Password aging	注释	
<input type="radio"/>	Administrator gr...	1	管理员组	<input type="checkbox"/>	'管理员' 组初始被...	
<input type="radio"/>	Users	2	用户	<input type="checkbox"/>	'用户' 组初始被授...	
<input checked="" type="radio"/>	Programmer	3	组_1	<input type="checkbox"/>		
<input type="radio"/>	Commissioner	4	组_1	<input type="checkbox"/>		
<input type="radio"/>	Operator	5	组_1	<input type="checkbox"/>		
<添加新许可证>						

图 8-1

步骤

- 在“用户”表格中双击“添加...”。
- 输入“Smith”作为用户名。
- 在“密码”列中单击  按钮。显示用于输入密码的对话框。
- 输入“smith”作为密码。
- 要确认密码，请在下面的域中再次输入密码。
- 使用  图标关闭此对话框。
- 在“组”表格中激活“调试员”用户组。
- 对用户“Foreman”重复步骤 2 到 6。
- 在“组”表格中激活“操作员”用户组。

结果

Users						
名称	密码	自动注销	退出登录时间	编号	注释	
Administrator	*****	<input checked="" type="checkbox"/>	5	1	将用户 'Administrator'...	
Miller	*****	<input checked="" type="checkbox"/>	5	2		
Smith	*****	<input checked="" type="checkbox"/>	5	3		
Foreman	*****	<input checked="" type="checkbox"/>	5	4		
<添加新许可证>						

组						
成员属于	名称	编号	显示名称	Password aging	注释	
<input type="radio"/>	Administrator gr...	1	管理员组	<input type="checkbox"/>	'管理员' 组初始被...	
<input type="radio"/>	Users	2	用户	<input type="checkbox"/>	'用户' 组初始被授...	
<input type="radio"/>	Programmer	3	组_1	<input type="checkbox"/>		
<input type="radio"/>	Commissioner	4	组_1	<input type="checkbox"/>		
<input checked="" type="radio"/>	Operator	5	组_1	<input type="checkbox"/>		
<添加新许可证>						

参见

实例：用户管理的结构 (页 1224)

8.7 使用系统函数

8.7.1 基础知识

8.7.1.1 系统函数

简介

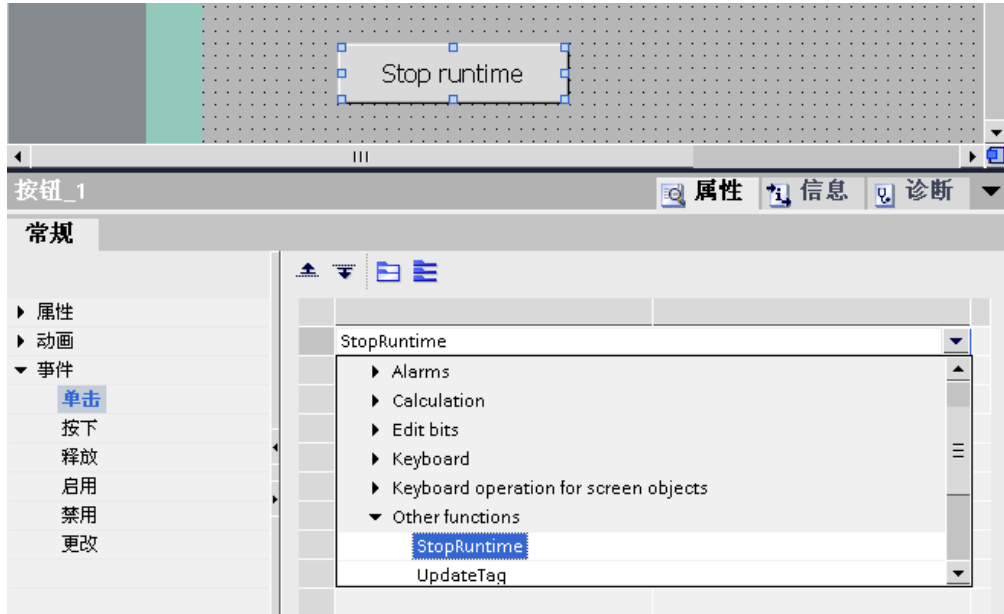
系统函数是预定义函数，在运行期间可用来执行许多任务，甚至不需要任何编程知识，例如：

- 计算，如以特定值或可变值增加变量值。
- 记录功能，如启动过程值日志。
- 设置，例如改变 PLC 或将 PLC 中的位置位。
- 报警，如一个不同用户登录后。

应用

在函数列表中使用系统函数。系统函数是预定义函数且不能更改。

组态函数列表时，从选择列表选择按类别排序的系统函数：



WinCC 中的每个系统函数均与一个对象和一个事件存在逻辑链接。一旦事件产生，系统函数即被触发。

语言相关性

系统函数的名称取决于设置的项目语言。项目设计者能立即识别其功能。

可用性

只能在函数列表中组态系统函数，这些系统函数受选定的 HMI 设备支持。如果将一个项目用于多个操作单元，操作单元不支持的系统函数将以不同颜色标记出来。

事件

对象和所选函数决定可定义哪些事件作为触发源来执行系统函数。

例如，“更改值”、“超出下限”和“超出上限”事件与“变量”对象相关联。而“装载”、“清除”事件则与“画面”对象相关。

参见

在 HMI 设备的当前显示上更换操作模式。(页 1238)

系统函数的使用(页 1233)

关于动态化的基础知识(页 1019)

系统函数对设备的依赖性(页 1241)

系统函数(页 1244)

8.7.1.2 系统函数的使用

简介

在运行系统中组态事件发生时，将处理函数列表。操作员可以触发事件，例如按下 HMI 设备上的功能键。系统还可以触发一个事件，例如，在过程值小于指定限制时。

应用

可以在能对事件作出反应的所有对象上组态系统函数。可以直接在函数列表中使用系统函数，从而控制顺序。

- WinCC Runtime 的函数列表

函数列表中的系统函数每次处理一行。为避免等待时间，将同时执行 WinCC Runtime 中运行时间较长的系统函数。例如，即使前一个系统函数还未完成，后一个系统函数也可以先被执行。

有关函数列表的组态实例，请参考“[在 HMI 设备的当前显示上更换操作模式。\(页 1238\)](#)”。

参见

[系统函数 \(页 1231\)](#)

[在 HMI 设备的当前显示上更换操作模式。\(页 1238\)](#)

8.7.2 使用函数列表

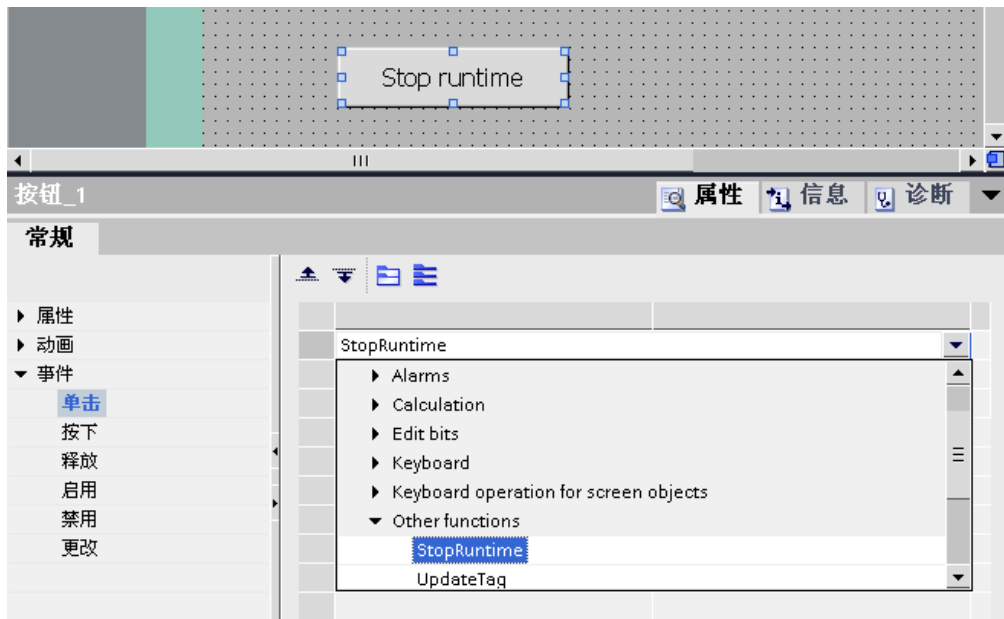
8.7.2.1 关于函数列表的基础知识

简介

发生组态的事件时，可以使用函数列表执行多个系统函数。

原理

为对象（例如画面对象或变量）的事件组态函数列表。可用事件取决于所选择的对象和 HMI 设备。



事件仅在项目处于运行时产生。这些事件包括：

- 变量的数值改变
- 按钮按下
- 运行系统激活

可为每个事件恰好组态一个函数列表。

注意

函数列表中的可组态系统函数的选择取决于所选取的 HMI 设备。

参见

[函数列表的属性 \(页 1234\)](#)

[组态函数列表 \(页 1235\)](#)

[编辑函数列表 \(页 1237\)](#)

[在运行系统中执行函数列表 \(页 1238\)](#)

[在 HMI 设备的当前显示上更换操作模式。 \(页 1238\)](#)

[系统函数对设备的依赖性 \(页 1241\)](#)

8.7.2.2 函数列表的属性

可用于特定 HMI 设备

同一项目可使用不同的 HMI 设备。更改项目中的 HMI 设备时，所选择的 HMI 设备不支持的所有系统函数将以黄色标记。在运行系统中不执行不支持的系统函数。

状态信息

组态期间项目数据在后台测试。在每个函数列表中，状态信息返回各自的系统函数的状态。

状态信息有下列含义：

- 橙色：运行时函数列表未执行，因为至少有一个系统函数尚未获得全部参数。
- 黄色：函数列表在运行时执行。然而，函数列表包含至少一个不受 HMI 设备支持的系统函数（例如由于设备类型改变）。

执行系统函数

函数列表中的系统函数在运行系统中按从上至下的顺序执行。为了避免等待时间，需要较长的运行时间（例如文件操作）的系统函数同时处理。例如，即使前一个系统函数还未完成，后一个系统函数也可以先被执行。

为了避免编写按顺序和按条件执行的过程，可以使用具有循环、条件语句和取消请求的脚本。

注意

可用于特定设备

脚本不可用于精简面板。

参见

关于函数列表的基础知识 (页 1233)
在 HMI 设备的当前显示上更换操作模式。 (页 1238)

8.7.2.3 组态函数列表

简介

可以通过从下拉列表中选择系统函数来组态函数列表。系统函数按照类别排列在下拉列表中。

要求

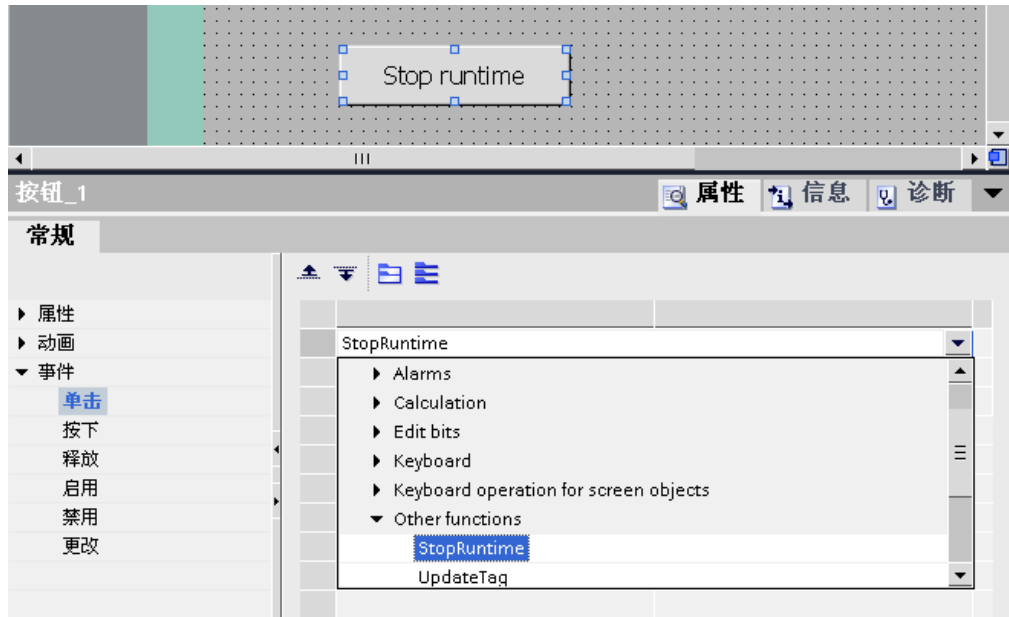
对象至少有一个可组态的事件。

步骤

执行下列步骤，组态函数列表：

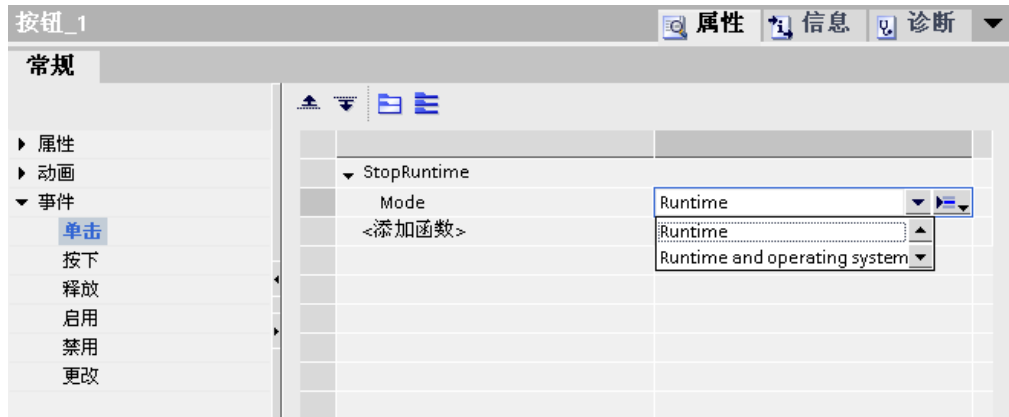
1. 打开 WinCC 中对象所在的编辑器。
2. 选择对象。
3. 在巡视窗口的“事件”下，单击想要组态函数列表的事件。
4. 从巡视窗口的下拉列表中选择“<无函数>”条目。
5. 从下拉列表中选择所需系统函数。

6. 也可以输入系统函数的名称。



将系统函数输入到函数列表中。

7. 如果系统函数有参数，指定参数值。



8. 如果想要添加其他系统函数或函数到函数列表，重复步骤 4 至 7。

结果

函数列表组态已经完成。除了组态的事件，函数列表状态也显示在巡视窗口中。在运行系统中发生组态的事件时，函数列表从上至下执行一遍。

参见

[编辑函数列表 \(页 1237\)](#)

[关于函数列表的基础知识 \(页 1233\)](#)

[在 HMI 设备的当前显示上更换操作模式。 \(页 1238\)](#)

8.7.2.4 编辑函数列表

简介

函数列表可以按如下方法进行编辑：

- 更改系统函数的执行顺序
- 删除系统函数

更多信息，请参考“[组态函数列表 \(页 1235\)](#)”。

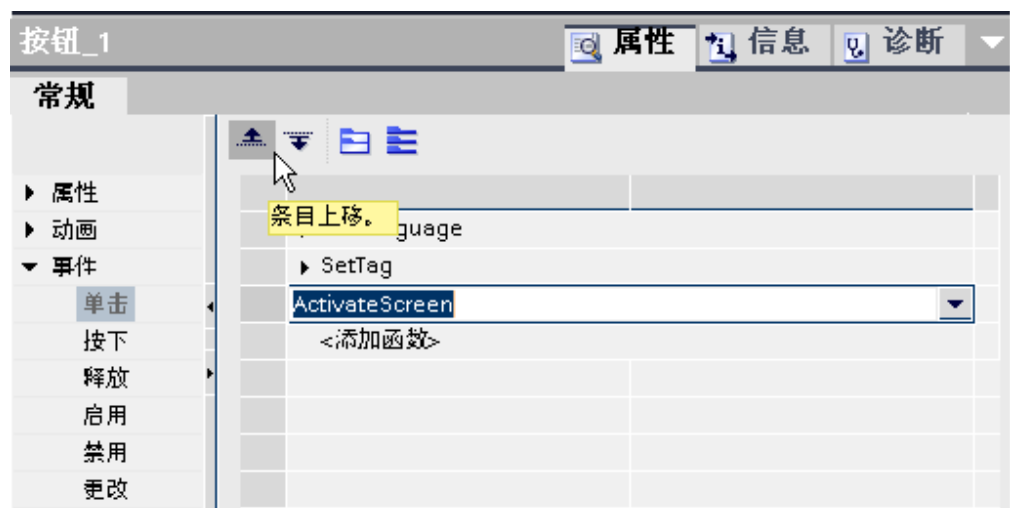
要求

函数列表组态已经完成。

步骤

执行下列步骤，编辑函数列表：

1. 打开 WinCC 中对象所在的编辑器。
2. 选择对象。
3. 在巡视窗口的“事件”下，单击要编辑函数列表的事件。
4. 从下拉列表中选择系统函数以更改函数列表的执行顺序。
5. 在巡视窗口中，一直单击相应箭头，直到将系统函数移动到所需位置。



6. 要从函数列表中删除系统函数，选择各个系统函数，然后使用快捷菜单中的“删除”命令。

其他方法

如果要同时移动多个系统函数，可在函数列表中将其一起选定。在按下并按住 <Ctrl> 的同时选择所需系统函数。

也可以使用拖放功能来移动。

参见

[关于函数列表的基础知识 \(页 1233\)](#)

[在 HMI 设备的当前显示上更换操作模式。 \(页 1238\)](#)

8.7.2.5 在运行系统中执行函数列表

原理

运行时函数列表从上至下完成。由于分为同步完成和异步完成两种形式，因此完成期间不必等待。系统通过判断系统函数的不同运行时间来进行区别。脚本总是独立于运行系统，同时进行处理。如果系统函数返回出错状态，函数列表的完成被取消。

同步完成

同步完成时，函数列表中的系统函数依次执行。必须在上一个系统函数完成后，下一个才能执行。

异步完成

执行文件操作(诸如存储和读取)的系统函数，比那些设置变量数值之类的系统函数需要更长的运行时间。

因此，需要较长运行时间的系统函数采用异步执行方式。当系统函数写入存储介质时，例如配方记录，下一个系统函数已开始执行。因为系统函数是并行完成的，避免了HMI设备的等待。

参见

关于函数列表的基础知识 (页 1233)

在HMI设备的当前显示上更换操作模式。(页 1238)

8.7.3 实例

8.7.3.1 在HMI设备的当前显示上更换操作模式。

任务

在本实例中，使用系统函数“SetDeviceMode”在HMI设备上切换“在线”和“离线”操作模式。也可在HMI设备上显示当前设置的操作模式。

要求

已经创建一个画面。

设置

在该实例中需要一个变量和一个文本列表以及下列设置：

Tag:

名称	PLC 连接	类型
操作模式	否	Bool

文本列表：

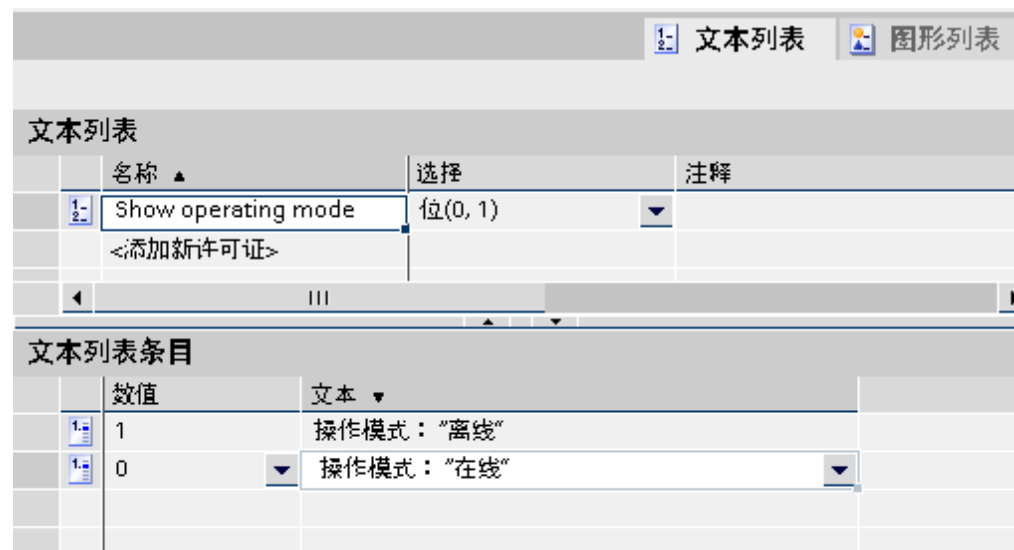
名称	包含	数值
Show operating mode	Boolean (0/1)	1: 操作模式：“在线” 0: 操作模式：“离线”

步骤

1. 创建上述变量“Operating mode”。



2. 创建上述文本列表“ShowOperatingMode”。



3. 打开过程画面并添加一个可将操作模式改为“在线”的按钮。
4. 在巡视窗口的“事件”组中，单击“按下”。
5. 对“按下”事件组态系统函数“SetDeviceMode”。该系统函数可在“设置”下的选择列表中找到。
6. 从选择列表中为参数“模式”选择条目“在线”。
7. 对“按下”事件组态系统函数“SetBit”。该系统函数可在“位处理”下的选择列表中找到。
8. 从选择列表中为参数“变量”选择变量“操作模式”。



9. 在过程画面中添加一个可将操作模式改为“离线”的按钮。

1. 重复步骤 4 至 7。从选择列表中为参数“模式”选择条目“离线”。组态系统函数“ResetBit”代替系统函数“SetBit”。



中间结果

可使用这两个按钮切换 HMI 设备的操作模式。

想要在 HMI 设备的输出域中显示当前设置的操作模式。

步骤

1. 在过程画面中，创建“符号 I/O 域”，并在巡视窗口的“常规”选项卡中进行以下设置。
 - 选择“输出”作为“模式”。
 - 选择文本列表“显示操作模式”作为“文本列表”。
 - 选择“操作模式”作为“变量”。



结果

使用按钮更改操作模式时，始终显示 HMI 设备上当前设置的操作模式。

参见

[系统函数 \(页 1231\)](#)
[关于函数列表的基础知识 \(页 1233\)](#)
[系统函数对设备的依赖性 \(页 1241\)](#)

8.7.4 参考

8.7.4.1 系统函数对设备的依赖性

系统函数的可用性

下表显示了 HMI 设备上系统函数和脚本的可用性。

技术数据如有更改，恕不另行通知。

精简面板

	KTP400 mono PN	KTP600 mono PN	KTP600 color PN	KTP1000 PN	TP1500 PN
脚本	否	否	否	否	否
Logoff (页 1244)	是	是	是	是	是
AdjustContrast (页 1244)	是	是	否	是	是
ActivateScreen (页 1245)	是	是	是	是	是

	KTP400 mono PN	KTP600 mono PN	KTP600 color PN	KTP1000 PN	TP1500 PN
ActivateScreenByNumber (页 1246)	是	是	是	是	是
ActivateCleanScreen (页 1246)	是	是	是	是	是
ActivatePreviousScreen (页 1247)	是	是	是	是	是
UpdateTag (页 1248)	是	是	是	是	是
Logon (页 1248)	是	是	是	是	是
EditAlarm (页 1249)	是	是	是	是	否
ScreenObjectCursorDown (页 1250)	是	是	是	是	是
ScreenObjectCursorUp (页 1249)	是	是	是	是	是
ScreenObjectPageDown (页 1251)	是	是	是	是	是
ScreenObjectPageUp (页 1250)	是	是	是	是	是
IncreaseFocusedValue (页 1252)	是	是	是	是	否
IncreaseTag (页 1252)	是	是	是	是	是
GoToHome (页 1253)	是	是	是	是	否
GoToEnd (页 1253)	是	是	是	是	否
InvertBit (页 1254)	是	是	是	是	是
InverseLinearScaling (页 1254)	是	是	是	是	是
CalibrateTouchScreen (页 1255)	是	是	是	是	是
TrendViewScrollForward (页 1255)	是	是	是	是	是
TrendViewScrollBack (页 1256)	是	是	是	是	是
TrendViewExtend (页 1256)	是	是	是	是	是
TrendViewCompress (页 1257)	是	是	是	是	是
TrendViewRulerForward (页 1257)	是	是	是	是	是
TrendViewRulerBackward (页 1258)	是	是	是	是	是
TrendViewSetRulerMode (页 1259)	是	是	是	是	是
TrendViewStartStop (页 1259)	是	是	是	是	是
TrendViewBackToBeginning (页 1260)	是	是	是	是	是
GetUserName (页 1260)	是	是	是	是	是
GetGroupNumber (页 1261)	是	是	是	是	是
GetPassword (页 1261)	是	是	是	是	是
LinearScaling (页 1262)	是	是	是	是	是
ClearAlarmBuffer (页 1263)	是	是	是	是	是
ClearAlarmBufferProtoolLegacy (页 1263)	是	是	是	是	是

	KTP400 mono PN	KTP600 mono PN	KTP600 color PN	KTP1000 PN	TP1500 PN
AlarmViewEditAlarm (页 1264)	是	是	是	是	是
AlarmViewAcknowledgeAlarm (页 1265)	是	是	是	是	是
AlarmViewShowOperatorNotes (页 1265)	是	是	是	是	是
AcknowledgeAlarm (页 1266)	是	是	是	是	否
RecipeViewNewDataRecord (页 1266)	是	是	是	是	是
RecipeViewGetDataRecordFromPLC (页 1267)	是	是	是	是	是
RecipeViewDeleteDataRecord (页 1267)	是	是	是	是	是
RecipeViewMenu (页 1268)	是	是	是	是	是
RecipeViewOpen (页 1269)	是	是	是	是	是
RecipeViewSetDataRecordToPLC (页 1269)	是	是	是	是	是
RecipeViewSaveDataRecord (页 1270)	是	是	是	是	是
RecipeViewSaveAsDataRecord (页 1270)	是	是	是	是	是
RecipeViewRenameDataRecord (页 1271)	是	是	是	是	是
RezeptanzeigeZeigeHilfetext (页 1271)	是	是	是	是	是
RecipeViewBack (页 1272)	是	是	是	是	是
ResetBit (页 1273)	是	是	是	是	是
ButtonPress (页 1273)	是	是	是	是	否
ButtonRelease (页 1274)	是	是	是	是	否
ShiftAndMask (页 1274)	是	是	是	是	是
PageDown (页 1276)	是	是	是	是	否
PageUp (页 1276)	是	是	是	是	否
SetDeviceMode (页 1276)	是	是	是	是	是
SetBit (页 1277)	是	是	是	是	是
SetBitWhileKeyPressed (页 1277)	是	是	是	是	否
SetLanguage (页 1278)	是	是	是	是	是
SetConnectionMode (页 1279)	是	是	是	是	是
SetTag (页 1279)	是	是	是	是	是
SimulateSystemKey (页 1281)	是	是	是	是	否
SimulateTag (页 1281)	是	是	是	是	是
StopRuntime (页 1282)	是	是	是	是	是

	KTP400 mono PN	KTP600 mono PN	KTP600 color PN	KTP1000 PN	TP1500 PN
TraceUserChange (页 1282)	是	是	是	是	是
DecreaseFocusedValue (页 1283)	是	是	是	是	否
DecreaseTag (页 1283)	是	是	是	是	是
ChangeConnection (页 1284)	是	是	是	是	是
ShowLogonDialog (页 1285)	是	是	是	是	是
ShowOperatorNotes (页 1285)	是	是	是	是	是
ShowAlarmWindow (页 1286)	是	是	是	是	是

参见

[系统函数 \(页 1231\)](#)

[关于函数列表的基础知识 \(页 1233\)](#)

[在 HMI 设备的当前显示上更换操作模式。 \(页 1238\)](#)

8.7.4.2 系统函数

8.7.4.2 Logoff

应用

在 HMI 设备上注销当前用户

语法

Logoff

在脚本中是否可用：如果组态的设备支持脚本，则可用 (Logoff)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

--

8.7.4.2 AdjustContrast

应用

将 HMI 设备的显示对比度改变一个级别

语法

AdjustContrast (Adjust)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (AdjustContrast)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Adjust

指定如何更改对比度：

0 (hmiDecrease) = 减少：将对比度降低一个级别。

1 (hmiIncrease) = 增加：将对比度增加一个级别。

应用实例

目的

需要分别用于增加和降低画面对比度的按钮。

关于组态的注意事项

对“按下”事件组态两个按钮并组态系统函数“AdjustContrast”。分配参数“Increase”和“Decrease”。

HMI 设备上的步骤

在运行时按下两个按钮之一时，对比度增加或降低一个级别。

8.7.4.2 ActivateScreen

应用

将画面切换到指定的画面。

使用“ActivateScreenByNumber”系统函数可以从根画面切换到固定窗口，反之亦然。

语法

ActivateScreen (Screen name, Object number)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ActivateScreen)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Screen name

要切换到的画面的名称。

Object number

画面切换后在指定画面中获得焦点的操作员控件元素。操作员控件元素的编号在组态期间使用 TAB 顺序确定。

在指定为“0”时：

如果调用该系统函数时焦点位于固定窗口，则固定窗口保留焦点。

如果调用该系统函数时焦点位于根画面，则指定画面中的第一个操作员控件元素获得焦点。

注意

如果将“到达边界”事件分配给“ActivateScreen”系统函数，则只有数值“0”对“Object number”参数有效。活动对象不是由对象号定义的，而是由画面更改之前其 X 位置定义的。

8.7.4.2 ActivateScreenByNumber

应用

根据变量值将画面切换到另一画面。

画面由其画面号标识。

语法

ActivateScreenByNumber (Screen number, Object number)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ActivateScreenByNumber)。更多信息，请参考[系统函数对设备的依赖性 \(页 1241\)](#)。”

参数

Screen number

包含目标画面的画面编号的变量。

如果需要从根画面切换到固定窗口，则指定“0”或“-1”：

0 = 从根画面切换到固定窗口

-1 = 从固定窗口切换到根画面

Object number

画面切换后在指定画面中获得焦点的画面对象的编号。操作员控件元素的编号在组态期间使用 TAB 顺序确定。

在指定为“0”时：

- 如果调用该系统函数时焦点位于固定窗口，则固定窗口保留焦点。
- 如果调用该系统函数时焦点位于根画面，则指定画面中的第一个操作员控件元素获得焦点。

8.7.4.2 ActivateCleanScreen

应用

在 HMI 设备上激活洁屏画面。在指定的时间内禁用 HMI 设备的显示屏。

当取消激活 HMI 设备的显示屏时，可对其进行清洁，而不会因无意间触发了触摸函数而受到影响。

语法

ActivateCleanScreen (Time period)

在脚本中是否可用：否

参数

Time period

禁用显示屏的时间段。剩余时间将以进度条显示。

取值范围为 10 到 300 秒。

注意

系统函数 ActivateCleanScreen 无法仿真。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 ActivatePreviousScreen

应用

将画面切换到在当前画面之前激活的画面。如果先前没有激活任何画面，则画面切换不执行。

最近调用的 10 个画面被保存。当切换到不再保存的画面时，会输出系统报警。

注意

如果要使用该函数，则必须在浏览结构中使用画面更改。

语法

ActivatePreviousScreen

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ActivatePreviousScreen)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

--

8.7.4.2 UpdateTag

应用

从 PLC 读取带有指定更新标识符的变量的当前值。 .

语法

UpdateTag (Update ID)

在脚本中是否可用： 否

参数

Update ID

为将要更新的变量分配的更新标识符。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 Logon

应用

在 HMI 设备上登录当前用户

语法

Logon (Password, User name)

在脚本中是否可用： 如果组态的设备支持脚本，则可用 (Logon)。 更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Password

从中读取用户登录密码的变量。

如果用户已经登录，则变量中的密码会删除。

User name

从中读取用户登录用户名的变量。

8.7.4.2 EditAlarm

应用

为选择的所有报警触发“编辑”事件。

如果要编辑的报警尚未被确认，则在调用该系统函数时自动确认。

系统函数可用于以下功能键：

语法

EditAlarm

在脚本中是否可用：如果组态的设备支持脚本，则可用 (EditAlarm)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

--

8.7.4.2 ScreenObjectCursorUp

应用

在给定的画面对象中执行<Up>键功能。

当不能使用画面对象的集成按钮时使用该系统函数。系统函数可用于以下画面对象：

- 用户视图
- 报警视图
- 配方视图

语法

ScreenObjectCursorUp (Screen object)

在脚本中是否可用：否

参数

画面对象

要触发其键功能的画面对象的名称。

注意

下面所列的 HMI 设备不支持画面对象的此功能：OP 73、OP 73micro、OP 77A、TP 177A 和 TP 177micro。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 ScreenObjectCursorDown

应用

在给定的画面对象中执行<Down>键功能。

当不能使用画面对象的集成按钮时使用该系统函数。系统函数可用于以下画面对象：

- 用户视图
- 报警视图
- 配方视图

语法

ScreenObjectCursorDown (Screen object)

在脚本中是否可用：否

参数

画面对象

要触发其键功能的画面对象的名称。

注意

下面所列的 HMI 设备不支持画面对象的此功能：OP 73、OP 73micro、OP 77A、TP 177A 和 TP 177micro。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 ScreenObjectPageUp

应用

在给定的画面对象中执行<Page Up>键功能。

当不能使用画面对象的集成按钮时使用该系统函数。系统函数可用于以下画面对象：

- 用户视图
- 报警视图
- 配方视图

语法

ScreenObjectPageUp (Screen object)

在脚本中是否可用：否

参数

Screen object

要触发其键功能的画面对象的名称。

注意

下面所列的 HMI 设备不支持画面对象的此功能：OP 73、OP 73micro、OP 77A、TP 177A 和 TP 177micro。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 ScreenObjectPageDown

应用

在给定的画面对象中执行<Page Down>键功能。

当不能使用画面对象的集成按钮时使用该系统函数。系统函数可用于以下画面对象：

- 用户视图
- 报警视图
- 配方视图

语法

ScreenObjectPageDown (Screen object)

在脚本中是否可用：否

参数

画面对象

要触发其键功能的画面对象的名称。

注意

下面所列的 HMI 设备不支持画面对象的此功能：OP 73、OP 73micro、OP 77A、TP 177A 和 TP 177micro。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 IncreaseFocusedValue

应用

将给定值添加到与具有当前焦点的输入域(下拉式列表、图形选择列表、滚动条)连接的变量的值上。

语法

IncreaseFocusedValue (Value)

在脚本中是否可用：否

参数

Value

为其添加给定值的变量。

8.7.4.2 IncreaseTag

应用

将给定值添加到变量值上。

$X = X + a$

注意

系统函数使用同一变量作为输入和输出值。如果想要使用此系统函数转换数值，可使用辅助变量。可使用系统函数“SetTag”将辅助变量分配给变量值。

如果在报警事件中组态了函数但变量未在当前画面中使用，则无法确保在 PLC 中使用实际的变量值。通过设置“循环连续”采集模式可以改善这种情况。

语法

IncreaseTag (Tag, Value)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (IncreaseTag)。更多信息，请参考“系统函数对设备的依赖性 (页 1241)”。

参数

变量

为其添加给定值的变量。

值

作为加数的数值。

8.7.4.2 GoToHome

应用

在 HMI 设备上执行<Home>键功能：

当 HMI 设备在默认情况下不具有该功能时使用此系统函数。

系统函数可用于以下功能键：

语法

GoToHome

在脚本中是否可用：如果组态的设备支持脚本，则可用 (GoToHome)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

--

8.7.4.2 GoToEnd

应用

在 HMI 设备上执行<End>键功能：

当 HMI 设备在默认情况下不具有该功能时使用此系统函数。

系统函数可用于以下功能键：

语法

GoToEnd

在脚本中是否可用：如果组态的设备支持脚本，则可用 (GoToEnd)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

--

8.7.4.2 InvertBit

应用

对给定的“Bool”型变量的值取反。

- 如果变量具有值 1 (真), 它将被设置为 0 (假)。
- 如果变量具有值 0 (假), 它将被设置为 1 (真)。

语法

InvertBit (Tag)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (InvertBit)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

变量

要设置其位的变量。

8.7.4.2 InverseLinearScaling

应用

使用线性函数 $X = (Y - b) / a$ ，将通过给定变量 Y 的值计算得出的数值赋给变量 X。

变量 X 和 Y 不能相同。与此函数相反的系统函数是“LinearScaling”。

注意

变量 X 和 Y 不能相同。如果要将变量转换为其自身，必须使用一个辅助变量。

可用系统函数“SetTag”将要转换变量的值分配给辅助变量。

语法

InverseLinearScaling (X, Y, b, a)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (InverseLinearScaling)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

X

要为其分配通过线性方程式计算得出的值的变量。

Y

其值用于计算的变量。

b

其值作为减数。

a

其值作为除数。

8.7.4.2 CalibrateTouchScreen

应用

调用用于校准触摸屏的程序。

在校准过程中，将会提示触摸屏显示上的五个位置。在 30 秒内触摸屏显示以确认校准过程。如果在该时间间隔内没有完成校准，校准设置被放弃。用户提示为英语。

首次启动操作员设备时使用该系统函数。

语法

CalibrateTouchScreen

在脚本中是否可用：如果组态的设备支持脚本，则可用 (CalibrateTouchScreen)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

--

注意

系统函数 CalibrateTouchScreen 无法仿真。

8.7.4.2 TrendViewScrollForward

应用

在趋势视图中向右滚动一个显示宽度。

语法

TrendViewScrollForward (Screen object)

在脚本中是否可用：否

参数

Screen object

在其中进行前滚的趋势视图的名称。

注意

TP 177A 和 TP 177micro HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 TrendViewScrollBack

应用

在趋势视图中向左回滚一个显示宽度。

语法

TrendViewScrollBack (Screen object)

在脚本中是否可用：否

参数

Screen object

在其中进行回滚的趋势视图的名称。

注意

TP 177A 和 TP 177micro HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 TrendViewExtend

应用

减少在趋势视图中显示的时间段。

语法

TrendViewExtend (Screen object)

在脚本中是否可用：否

参数

Screen object

要在其中减少显示时间的趋势视图的名称。

注意

TP 177A 和 TP 177micro HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 TrendViewCompress

应用

增加在趋势视图中显示的时间段。

语法

TrendViewCompress (Screen object)

在脚本中是否可用：否

参数

Screen object

要增加显示时间段的趋势视图的名称。

注意

TP 177A 和 TP 177micro HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 TrendViewRulerForward

应用

在趋势视图中向前(右)移动标尺。

注意

为了能够移动标尺，必须将标尺打开。可以使用系统函数“TrendViewSetRulerMode”实现此操作。

语法

TrendViewRulerForward (Screen object)

在脚本中是否可用：否

参数

Screen object

要在其中前移标尺的趋势视图名称。

注意

TP 177A 和 TP 177micro HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 TrendViewRulerBackward

应用

在趋势视图中向后(左)移动标尺。

注意

为了能够移动标尺，必须将标尺打开。可以使用系统函数“TrendViewSetRulerMode”实现此操作。

语法

TrendViewRulerBackward (Screen object)

在脚本中是否可用：否

参数

Screen object

要在其中向后移动标尺的趋势视图的名称。

注意

TP 177A 和 TP 177micro HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 TrendViewSetRulerMode**应用**

在趋势视图中隐藏或显示标尺。标尺显示与 X 值相关联的 Y 值。

语法

TrendViewSetRulerMode (Screen object)

在脚本中是否可用：否

参数**Screen object**

要在其中隐藏或显示标尺的趋势视图名称。

注意

TP 177A 和 TP 177micro HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 TrendViewStartStop**应用**

停止或继续趋势视图中的趋势记录。

语法

TrendViewStartStop (Screen object)

在脚本中是否可用：否

参数

Screen object

要开始或停止其中的趋势记录的趋势视图的名称。

注意

TP 177A 和 TP 177micro HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 TrendViewBackToBeginning

应用

在趋势视图中向后翻页到趋势记录的开始处。趋势记录的起始值将显示在此处。

语法

TrendViewBackToBeginning (Screen object)

在脚本中是否可用：否

参数

Screen object

要增加显示时间段的趋势视图的名称。

注意

TP 177A 和 TP 177micro HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 GetUserName

应用

在给定的变量中写入当前登录到 HMI 设备的用户的用户名。

如果给出的变量具有控制连接，则用户名在 PLC 上也可用。该系统函数可以用来实现一些与用户相关的函数。

语法

GetUserName (Tag)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (GetUserName)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Tag

要将用户名写入其中的变量。

8.7.4.2 GetGroupNumber

应用

读取登录到 HMI 设备的用户所属组的编号，并将其写入给定的变量。

语法

GetGroupNumber (Tag)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (GetGroupNumber)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Tag

要将组的编号写入其中的变量。

8.7.4.2 GetPassword

应用

在给定的变量中写入当前登录到 HMI 设备的用户的密码。

注意

确保给定变量的值未显示在项目中的其它位置。

注意

无法读取 SIMATIC Logon 用户的密码。

语法

GetPassword (Tag)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (GetPassword)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Tag

要将结果写入其中的变量。

8.7.4.2 LinearScaling

应用

为变量 Y 赋值，该变量通过线性函数 $Y = (a * X) + b$ 利用给定变量 X 的值计算得出。
与此函数相反的系统函数是“InverseLinearScaling”。

注意

变量 X 和 Y 不能相同。如果要将变量转换为其自身，必须使用一个辅助变量。

可用系统函数“SetTag”将要转换变量的值分配给辅助变量。

语法

LinearScaling (Y, X, a, b)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (LinearScaling)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Y

要为其分配通过线性方程式计算得出的值的变量。

X

其值用于计算的变量。

a

作为乘数的数值。

b

作为加数的数值。

8.7.4.2 ClearAlarmBuffer

应用

删除 HMI 设备报警缓冲区中的报警。

注意

尚未确认的报警也被删除。

语法

ClearAlarmBuffer (Alarm class number)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ClearAlarmBuffer)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Alarm class number

确定要从报警缓冲区中删除的报警：

0 (hmiAll) = 所有报警/事件

1 (hmiAlarms) = 错误

2 (hmiEvents) = 警告

3 (hmiSystem) = 系统事件

4 (hmiS7Diagnosis) = S7 诊断报警

注意

可用于特定设备

S7 诊断报警不可用于精简面板。

8.7.4.2 ClearAlarmBufferProtoolLegacy

应用

该系统函数用来确保兼容性。

它具有与系统函数“ClearAlarmBuffer”相同的功能，但使用旧的 ProTool 编号方式。

语法

ClearAlarmBufferProtoolLegacy (Alarm class number)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ClearAlarmBufferProtoolLegacy)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Alarm class number

将要删除其消息的报警类别号：

-1 (hmiAllProtoolLegacy) = 所有报警/事件

0 (hmiAlarmsProtoolLegacy) = 错误

1 (hmiEventsProtoolLegacy) = 警告

2 (hmiSystemProtoolLegacy) = 系统事件

3 (hmiS7DiagnosisProtoolLegacy) = S7 诊断报警

注意

可用于特定设备

S7 诊断报警不可用于精简面板。

8.7.4.2 AlarmViewEditAlarm

应用

对给定报警画面内选择的全部报警，触发“编辑”事件。

当不能使用 ActiveX 控件的集成按钮时，使用该系统函数。

可以在“编辑”事件上依次组态系统函数。例如，可以切换到出现报警的过程画面。

注意

如果要编辑的报警尚未被确认，则在调用该系统函数时自动确认。

语法

AlarmViewEditAlarm (Screen object)

在脚本中是否可用：否

参数

Screen object

在其中触发事件的报警画面的名称。

注意

下面所列的 HMI 设备不支持画面对象的此功能：OP 73、OP 73micro、OP 77A、TP 177A 和 TP 177micro。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 AlarmViewAcknowledgeAlarm

应用

确认在给定报警视图中选择的报警。

当不能使用 ActiveX 控件的集成按钮时，使用该系统函数。

语法

AlarmViewAcknowledgeAlarm (Screen object)

在脚本中是否可用：否

参数**Screen object**

在其中触发事件的报警画面的名称。

注意

下面所列的 HMI 设备不支持画面对象的此功能：OP 73、OP 73micro、OP 77A、TP 177A 和 TP 177micro。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 AlarmViewShowOperatorNotes

应用

在给定报警画面中显示组态好的所选报警的操作员注释。

语法

AlarmViewShowOperatorNotes (Screen object)

在脚本中是否可用：否

参数

Screen object

在其中触发事件的报警画面的名称。

注意

下面所列的 HMI 设备不支持画面对象的此功能：OP 73、OP 73micro、OP 77A、TP 177A 和 TP 177micro。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 AcknowledgeAlarm

应用

确认选择的所有报警。

该系统函数用于 HMI 设备没有 ACK 键时或报警屏幕的集成键不能使用时。

系统函数可用于以下功能键：

语法

AcknowledgeAlarm

在脚本中是否可用：否

参数

--

8.7.4.2 RecipeViewNewDataRecord

应用

在给定配方视图中创建新数据记录。

语法

RecipeViewNewDataRecord (Screen object)

在脚本中是否可用：否

参数

Screen object

在其中创建新配方数据记录的配方视图的名称。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RecipeViewGetDataRecordFromPLC

应用

将 PLC 中当前装载的数据记录传送至 HMI 设备并在配方视图中显示。

语法

RecipeViewGetDataRecordFromPLC (Screen object)

在脚本中是否可用：否

参数

Screen object

配方视图的名称，在其中显示来自 PLC 的配方数据记录。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RecipeViewDeleteDataRecord

应用

删除当前显示在配方视图中的数据记录。

语法

RecipeViewDeleteDataRecord (Screen object)

在脚本中是否可用：否

参数

Screen object

配方视图的名称，在其中删除显示的配方数据记录。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RecipeViewMenu

应用

打开指定简单配方视图的菜单。

该系统函数只能用于简单配方视图。

语法

RecipeViewMenu (Screen object)

在脚本中是否可用：否

参数

Screen object

要打开菜单的配方视图的名称。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RecipeViewOpen

应用

显示给定配方视图中的数据记录值。如果配方数据记录值已显示在操作员设备上，系统函数不被执行。

该系统函数在组态了简单配方视图后使用。在简单配方视图中，每次只在操作员设备上显示一个选择列表。系统函数“RecipeViewBack”用于显示前一选择列表。

语法

RecipeViewOpen (Screen object)

在脚本中是否可用：否

参数

Screen object

要显示其配方数据记录的配方视图的名称。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RecipeViewSetDataRecordToPLC

应用

将当前显示在配方视图中的配方数据记录传送到 PLC。

语法

RecipeViewSetDataRecordToPLC (Screen object)

在脚本中是否可用：否

参数

Screen object

配方视图的名称，将其配方数据记录传送到相连接 PLC。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RecipeViewSaveDataRecord

应用

保存当前显示在配方视图中的配方数据记录。

语法

RecipeViewSaveDataRecord (Screen object)

在脚本中是否可用：否

参数

Screen object

保存其配方数据记录的配方视图的名称。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RecipeViewSaveAsDataRecord

应用

以新名称保存当前在配方视图中显示的数据记录。

语法

RecipeViewSaveAsDataRecord (Screen object)

在脚本中是否可用：否

参数

Screen object

配方视图的名称，其配方数据记录以新名称保存。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RecipeViewRenameDataRecord

应用

重命名在给定配方视图中选择的数据记录。

该系统函数只能用于简单配方视图。

语法

RecipeViewRenameDataRecord (Screen object)

在脚本中是否可用：否

参数

Screen object

配方视图的名称，其中的配方数据记录要重命名。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RezepturanzeigeZeigeHilfetext

应用

显示给定配方视图的已组态信息文本。

语法

RecipeViewShowOperatorNotes (Screen object)

在脚本中是否可用：否

参数

Screen object

显示其已组态的帮助文本的配方视图的名称。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 RecipeViewBack

应用

返回到给定配方视图中的上一个选择列表。

如果配方的选择域显示在 HMI 设备上，则系统函数无效。运行时，选择列表的操作顺序：

- 配方名
- 数据记录名
- RecipeDataRecordValues

该系统函数在组态了简单配方视图后使用。在简单配方视图中，每次只在操作员设备上显示一个选择列表。系统函数“RecipeViewOpen”用于显示配方数据记录值。

语法

RecipeViewBack (Screen object)

在脚本中是否可用：否

参数

Screen object

在其中触发命令的配方视图的名称。

注意

OP 77A 和 TP 177A HMI 设备不支持画面对象的此功能。

参见

[系统函数对设备的依赖性 \(页 1241\)](#)

8.7.4.2 ResetBit**应用**

将“Bool”型变量的值设置为 0 (假)。

语法

ResetBit (Tag)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ResetBit)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数**变量**

设置为 0 (假) 的 BOOL 型变量。

8.7.4.2 ButtonPress**应用**

触发给定画面对象上的“按下”事件。

例如，当希望使用操作员设备的功能键来操作屏幕按钮时，可使用该系统函数。

注意

系统函数“ButtonPress”和“ButtonRelease”必须始终一起组态。因此，如果在某一功能键的“按下”事件上组态了系统函数“ButtonPress”，则在同一功能键的“释放”事件上也要组态系统函数“ButtonRelease”。

语法

ButtonPress (Screen object)

在脚本中是否可用：否

参数**Screen object**

要触发其事件的画面对象的名称。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 ButtonRelease

应用

触发给定画面对象上的“释放”事件。

例如，当希望使用操作员设备的功能键来操作屏幕按钮时，可使用该系统函数。

注意

系统函数“ButtonPress”和“ButtonRelease”必须始终一起组态。因此，如果在某一功能键的“按下”事件上组态了系统函数“ButtonPress”，则在同一功能键的“释放”事件上组态了系统函数“ButtonRelease”。

语法

ButtonRelease (Screen object)

在脚本中是否可用：否

参数

Screen object

要触发其事件的画面对象的名称。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 ShiftAndMask

应用

此系统函数将源变量的输入位组合格式转换成目标变量的输出位组合格式。这包括移动位和掩码位。

注意

如果源变量和目标变量位数不同，则在目标变量中使用系统函数会导致超出取值范围。

语法

ShiftAndMask (Source tag, Target tag, Bits to shift, Bits to mask)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ShiftAndMask)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Source tag

变量包括输入位组合格式。允许整型变量，例如“Byte”、“Char”、“Int”、“UInt”、“Long”和“ULong”。

示例：16 位整型源变量，设置当前实际值为 72：0000000001001000.

Bits to shift

输入位组合格式向右侧移动的位数。负值意味着将输入位组合格式向左侧移动。

实例：“Bits to shift 的值为“+3”。当调用该系统函数时，输入位组合格式向右移动 3 位：0000000000001001.

左侧用“0”填充。在右侧截掉 3 位。新的十进制值为“9”。

注意

如果源变量为有符号的整型数据类型，且带有符号“-”，则最左侧位为“1”。符号位向右侧移动后，该位将由“0”填充。符号变为“+”。

Bits to mask

掩码位是一个整数。其位组合格式用于与移动的输入位组合格式相乘。示例：整数“2478”的位组合格式为“0000100110101110”。

您可通过三种不同途径输入掩码位：

- 十六进制：首先输入前缀“0h”或者“0H”，随后是可选空格，以便于阅读。然后将位组合格式分为四个组 (0000)(1001)(1010)(1110) 且将每个块改为 16 位编码：(0)(9)(A)(E)。只允许使用字符 0-9, A-F, a-f：“0h 09AE”。
- 二进制：首先输入前缀“0h”或者“0H”，随后是可选空格，以便于阅读。然后将二进制位组合格式分为四个块 0000 1001 1010 1110 且用空格隔开以便检查：只允许使用字符“0”或者“1”：“0b 0000 1001 1010 1110”。
- 十进制：直接输入值“2478”，不加任何前缀。

Target tag (输出)

输出位组合格式保存在变量中。允许整型变量，例如“Byte”、“Char”、“Int”、“UInt”、“Long”和“ULong”。

示例：移动的输入位组合格式通过逻辑 AND 逐位运算与位掩码相乘：0000000000001001。结果为十进制值“8”，保存在目标变量中。

请注意以下事项：

- 源变量和目标变量位数相同。

- 要移动的位数应少于源变量和目标变量中的位数。
- 要掩码的位不能多于源变量和目标变量的位数。

8.7.4.2 PageUp

应用

在 HMI 设备上执行 <PageUp> 键功能。

系统函数可用于以下功能键：

语法

PageUp

在脚本中是否可用：如果组态的设备支持脚本，则可用 (PageUp)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

-

8.7.4.2 PageDown

应用

在 HMI 设备上执行 <Pagedown> 键功能：

系统函数可用于以下功能键：

语法

PageDown

在脚本中是否可用：如果组态的设备支持脚本，则可用 (PageDown)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

-

8.7.4.2 SetDeviceMode

应用

切换 HMI 设备上的运行模式。可以有列操作类型：“在线”、“离线”和“传送”

语法

SetDeviceMode (Operating mode)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (SetDeviceMode)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Operating mode

确定 HMI 设备的运行模式：

0 (hmiOnline) = 在线：建立至 PLC 的连接。

1 (hmiOffline) = 离线：断开至 PLC 的连接。

2 (hmiTransfer) = 传送：项目可以从组态计算机传送到 HMI 设备。

注意

如果将 PC 用作 HMI 设备，则当传送完成后切换运行模式时，运行系统软件将退出。

8.7.4.2 SetBit

应用

将“Bool”型变量的值设置为 1 (真)。

语法

SetBit (Tag)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (SetBit)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Tag

要将其值设置为 1 (真) 的 BOOL 型变量。

8.7.4.2 SetBitWhileKeyPressed

应用

只要用户按下已组态的键，给定变量中的位即设置为 1(真)。

在改变了给定位之后，系统函数将整个变量传送回 PLC。但是并不检查变量中的其它位是否同时改变。在变量被传送回 PLC 之前，操作员和 PLC 可能只能读该变量。仅应使用该函数访问 BOOL 类型的变量，以避免发生同时访问相同变量的问题。

注意

即使该键保持按下状态，与事件“释放”有关的所有函数均将通过为某个键所组态的画面变化来立即执行。

如果为某个功能键组态了系统函数“SetBitWhileKeyPressed”，则该位在画面改变后将立即复位。该动作是必需的，因为在画面改变后键赋值也发生了变化。

如果 PLC 支持 BOOL 型变量，不要使用该系统函数。而使用系统函数“SetBit”。

语法

SetBitWhileKeyPressed (Tag, Bit)

在脚本中是否可用：否

参数**Tag**

要将其中的一个位暂时设置为 1(真)的变量。只要 PLC 允许，可以仅使用 BOOL 类型的变量。

Bit

要暂时设置为 1(真)的位的编号。

注意

要实现可靠的功能，必须保证与实际过程值一起使用的变量的更新。因此，应在 IO 域中组态变量或将函数分配给画面元素（如按钮）。

如果为函数组态了短事件（如报警的激活），则只能通过设置连续读取的变量访问实际过程值。

8.7.4.2 SetLanguage**应用**

切换 HMI 设备上的语言。所有组态的文本和系统事件以新设置的语言显示在 HMI 设备上。

语法

SetLanguage (Language)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (SetLanguage)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数**Language**

确定在 HMI 设备上设置了哪种语言。有下列规范：

- -1 (hmiToggle) = 切换：切换到下一种语言。组态期间在“项目语言”编辑器中确定顺序。

- 在“项目语言”编辑器的“语言切换顺序”下指定的编号：切换到带有给定编号的语言。
- 按照 VBScript5 参考的语言缩写：这样可切换到与指定语言代码相对应的语言，例如“de-DE”为德语(德国)或“en-US”为英语(美国)。

在 VBScript 基本信息的主题“区域图标标识号 (LCID) 图”下具有语言缩写总览。

8.7.4.2 SetTag

应用

将新值赋给给定的变量。

注意

该系统函数可用于根据变量类型分配字符串和数字。

语法

SetTag (Tag, Value)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (SetTag)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

变量

为其分配给定值的变量。

Value

为给定变量所赋的值。

注意

“SetTag”系统函数只能在建立连接后执行。

8.7.4.2 SetConnectionMode

应用

连接或断开给定的连接。

注意

只有在 HMI 设备上设置了“在线”运行模式后，才能建立至 PLC 的连接。可以使用系统函数“SetDeviceMode”实现此操作。

语法

SetConnectionMode (Mode, Connection)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (SetConnectionMode)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数**Mode**

确定是建立还是断开至 PLC 的连接：

0 (hmiOnline) = 在线：建立连接。

1 (hmiOffline) = 离线：断开连接。

Connection

与 HMI 设备相连的 PLC。在连接编辑器中指定 PLC 的名称。

在脚本中多次使用系统函数

如果在脚本中将系统函数“SetConnectionMode”用于不同的连接，则可能无法正确执行所有的函数。为防止出现此类情况，请执行以下步骤：

1. 使用起始值“0”创建“BOOL”类型的变量。
2. 在变量的“数值改变”事件上组态“SetConnectionMode”函数。例如，如果您打算断开 3 个连接，则必须组态该函数三次。
3. 在脚本中，为变量应用“InvertBit”函数。

应用实例

此系统函数的两个典型应用实例如下：

- 测试

只要没有 PLC 与 HMI 设备相连，在 HMI 设备上进行测试期间将不会有错误消息输出。如果 HMI 设备与 PLC 相连，则它与 PLC 的连接可通过按键来建立。

- 启动

要为一个系统组态多个 PLC。首先，将除其中一个之外的所有 PLC 组态为“离线”。在启动第一个 PLC 之后，与其它每个 PLC 的连接都可通过按键来建立。用这种方法，可以依次启动其它 PLC。

8.7.4.2 SimulateSystemKey

应用

模拟系统键的特性。如果系统键（例如“ACK”键、“Input”键或数字键区）在 HMI 设备上不可用，则请使用此系统函数。

语法

SimulateSystemKey (System key)

在脚本中是否可用：否

参数

System Key

将要模拟其特性的系统键。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 SimulateTag

应用

仿真变量和动态对象（如文本列表）的特性，而无需将 HMI 设备与 PLC 相连。

例如，可使用该系统函数来演示项目的功能。

注意事项

只有数据类型为整型的变量可用于仿真。但数据类型为整型和长整型的变量可与 OP 73、OP 73 micro、OP 77A、TP 177A 和 TP 177micro 一起使用。

语法

SimulateTag (Tag, Cycle, Maximum value, Minimum value, Value)

在脚本中是否可用：否

参数

Tag

其值要改变的变量。

Cycle

与基本周期 200 毫秒相乘的因子。该周期定义变量值何时由指定值改变。允许周期在 1 和 32767 之间。

Maximum value

变量在仿真过程中可以假设的最大值。最大值必须大于最小值，但必须小于或等于 32767。如果达到了最大值，则在下一个更新周期之后将变量值设置为最小值。

Minimum value

变量在仿真过程中可以假设的最小值。最小值必须小于最大值，但必须大于或等于- 32768。如果达到了最小值，则在下一个更新周期之后将变量值设置为最大值。

Value

在每一周期执行期间变量值根据其进行改变的值。负值将减少变量值。允许值在 -32768 和 32767 之间。

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 StopRuntime**应用**

退出运行系统软件，从而退出运行在 HMI 设备上的项目。

语法

StopRuntime (Mode)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (StopRuntime)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数**Mode**

确定在退出运行系统后操作系统是否关闭。

0 (hmiStopRuntime) = 运行系统：操作系统不关闭

1 (hmiStopRuntimeAndOperatingSystem) = 运行系统和操作系统：操作系统关闭(对于 WinCE 不适用)

8.7.4.2 TraceUserChange**应用**

输出显示当前哪个用户正登录到 HMI 设备的系统事件。

此系统函数只能用于调度程序。

语法

TraceUserChange

在脚本中是否可用：否

参数

--

8.7.4.2 DecreaseFocusedValue**应用**

从变量的值中减去给定值，该变量与输入域或具有当前焦点的下拉式列表、图形选择列表、滚动条相连接。

语法

DecreaseFocusedValue (Value)

在脚本中是否可用：否

参数

Value

从变量值中减去的数值。

8.7.4.2 DecreaseTag**应用**

从变量值中减去给定的值。

$X = X - a$

注意

系统函数使用同一变量作为输入和输出值。如果想要使用此系统函数转换数值，可使用辅助变量。可使用系统函数“SetTag”将辅助变量分配给变量值。

如果在报警事件中组态了函数但变量未在当前画面中使用，则无法确保在 PLC 中使用实际的变量值。通过设置“循环连续”采集模式可以改善这种情况。

语法

DecreaseTag (Tag, Value)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (DecreaseTag)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

变量

要减去给定值的变量。

Value

其值作为减数。

8.7.4.2 ChangeConnection

应用

断开与正在使用的、与当前使用的 PLC 之间的连接，使用另一个地址建立与另一个 PLC 的连接。

注意

更改为另一个地址时，确保该地址尚未被其它 HMI 设备使用。

支持以下地址类型：

- IP 地址

支持以下 PLC 类型：

- SIMATIC S7 1200

语法

ChangeConnection (Connection, Address, Slot, Rack)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ChangeConnection)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Connection

要断开的连接的名称。例如，该名称在“连接”编辑器中进行组态时设置。

Address

要与其建立连接的 PLC 的 IP 地址。

注意

使用变量指定地址。对象列表显示所有数据类型的变量。仅选择下列数据类型的变量：

- 以太网连接：“String”数据类型
-

Slot

要与其建立连接的 PLC 的插槽。

Rack

要与其建立连接的 PLC 的机架。

应用实例

目的

想要在多台机器上操作一台 HMI 设备。项目中组态了多个 PLC，通过按键方式切换至想要的 PLC。当切换 PLC 时，断开与正在使用的 PLC 的连接。然后重新建立与具有其它地址参数的新的 PLC 的连接。为能访问新的 PLC 的值，需要为使用的 PLC 组态相同的变量。

默认情况下，使用创建项目时所给定的 PLC。

1. 在“连接”编辑器中输入 PLC 的名称和地址。
2. 在过程画面中组态一个按钮。
3. 对“按下”事件组态系统函数“ChangeConnection”。
4. 输入连接名称和 PLC 地址作为参数。

8.7.4.2 ShowLogonDialog

应用

在 HMI 设备上打开用户可用于登录到 HMI 设备的对话框。

语法

ShowLogonDialog

在脚本中是否可用：否

参数

--

参见

系统函数对设备的依赖性 (页 1241)

8.7.4.2 ShowOperatorNotes

应用

显示所选对象的已组态的信息文本。

如果功能键上组态了该函数，则将显示当前具有焦点的画面对象的帮助文本。如果操作员注意事项是为画面本身组态的，那么可以通过按下<Enter>或双击帮助窗口切换到该文本。

如果按钮上组态了该函数，则只显示当前画面的帮助文本。如果在按钮自身上组态了帮助文本，则最初只显示此按钮的帮助文本。按下<Enter>或双击帮助窗口可以切换到当前画面的操作员注意事项。

注意

在帮助窗口打开期间，无法使用任何其它画面对象。要使用画面对象，请关闭帮助窗口。

关闭帮助窗口

可按照下列方式关闭帮助窗口：

使用按键：

- 通过再次按<HELP>键
- 通过按<ESC>键

使用触摸屏：

- 单击



按钮：

语法

ShowOperatorNotes (Display mode)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ShowOperatorNotes)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Display mode

确定隐藏或显示所组态的帮助文本：

0 (hmiOff) = 关：隐藏所组态的帮助文本

1 (hmiOn) = 开：显示所组态的帮助文本

-1 (hmiToggle) = 切换：在两种模式之间切换。

8.7.4.2 ShowAlarmWindow

应用

隐藏或显示 HMI 设备上的报警窗口。

语法

ShowAlarmWindow (Object name, Display mode)

在脚本中是否可用：如果组态的设备支持脚本，则可用 (ShowAlarmWindow)。更多信息，请参考“[系统函数对设备的依赖性 \(页 1241\)](#)”。

参数

Object name

要隐藏或显示的报警画面的名称。

Display mode

确定隐藏或显示报警窗口：

0 (hmiOff) = 关：隐藏报警画面

1 (hmiOn) = 开：显示报警画面

-1 (hmiToggle) = 切换：在两种模式之间切换。

8.7.4.3 事件

8.7.4.3 已清除

描述

当 HMI 设备上的当前画面被清除时发生。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 已装载

描述

画面切换后，将所有组态的显示操作对象载入活动画面中时，发生该事件。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

注意

启用画面切换，以确保画面打开后与控件建立连接。

8.7.4.3 启用

描述

用户使用组态的 TAB 顺序选择显示或操作对象时发生该事件。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

注意

例如，如果用户使用鼠标单击按钮，将触发“单击”事件。想要触发“启用”事件的用户，必须使用 TAB 键来选择按钮。

8.7.4.3 调整

描述

如果显示和操作人员控制对象的状态发生变化，将发生该事件。

例如，如果用户按下该键，则对象状态发生变化。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 在对话框打开时

描述

该事件在模式对话框打开时触发。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 在对话框关闭时

描述

该事件在模式对话框关闭时触发。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 用户改变

描述

发生在用户退出 HMI 设备或另一用户登录 HMI 设备时。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 画面切换

描述

画面切换后，将所有组态的显示操作对象载入画面中时，发生该事件。
如果想要在画面切换到某一画面期间执行其它系统函数，可使用事件“已装载”。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 禁用

描述

用户从显示操作对象获得焦点时发生该事件。
可以使用组态的 TAB 顺序或通过使用鼠标执行其它操作来禁用画面对象。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

注意

当关闭画面时，不执行画面的“取消激活”事件上的系统函数或脚本。

8.7.4.3 按下

描述

用户使用鼠标左键单击按钮以及用户按 <RETURN> 或 <SPACE> 时发生该事件。
用户右键单击符号库中的对象时发生该事件。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 离开

描述

当取消激活报警时发生。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 进入

描述

触发报警并显示在报警视图中时发生该事件。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 违反“上限误差”限制

描述

超出变量的“上限误差”限制时发生该事件。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 违反“下限误差”限制

描述

违反变量的“下限误差”限制时发生该事件。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 单击

描述

用户使用鼠标单击显示操作对象或使用手指触碰触摸屏时发生了该事件。

如果单击了错误的对象，请按照如下步骤防止处理已组态的函数列表：

- 在按住鼠标按钮的同时，将鼠标指针从该对象移开。在鼠标指针离开该对象后，立即释放鼠标按钮。这样，函数列表将不会执行。
- 在触摸屏上，必须用手指触摸显示屏直到产生反应，例如画面切换。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 报警回路

描述

当用户在报警视图中选择报警后单击“Loop-In-Alarm”按钮或双击报警之后，即会发生该事件。

可以使用事件“Loop-In-Alarm”组态系统函数，例如切换到发生报警的画面。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 释放

描述

用户释放按钮。

只要按钮保持按下状态，该事件便不会产生。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 报警缓冲区溢出

描述

达到所组态的报警缓冲区大小。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 确认

描述

在用户确认报警时发生该事件。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 运行系统停止

描述

用户在 HMI 设备上退出运行系统软件时产生该事件。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

注意

不能在“运行系统停止”事件上组态任何脚本。

8.7.4.3 按下按键

描述

用户按键盘上的键时发生该事件。

操作员不使用 <F10> 和 <ALT+PRINT> 键执行过程操作。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 释放按键

描述

用户释放键盘上的键时发生该事件。

操作员不使用 <F10> 和 <ALT+PRINT> 键执行过程操作。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 切换

描述

用户切换显示操作对象时发生该事件，例如从“开”切换到“关”。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.7.4.3 值改变

描述

当对象的值或者数组元素的值改变时发生。

变量值的更改由 PLC 或用户触发，例如输入新值时。

注意

请注意，该事件的可用性取决于 HMI 设备和对象类型。

8.8 使用调度程序

8.8.1 调度程序的应用领域

定义

可以使用调度程序组态在后台运行的任务。通过将系统函数链接到触发源来创建任务。当触发事件发生时，将会执行所链接的函数。

应用实例

调度程序用于自动执行受事件控制的任务。例如，可以使用任务自动监视用户操作。

参见

使用任务和触发源 (页 1295)

调度程序 (页 1295)

示例：用户更改后更新用户 (页 1299)

8.8.2 使用任务和触发源

简介

任务由触发源和任务类型组成。

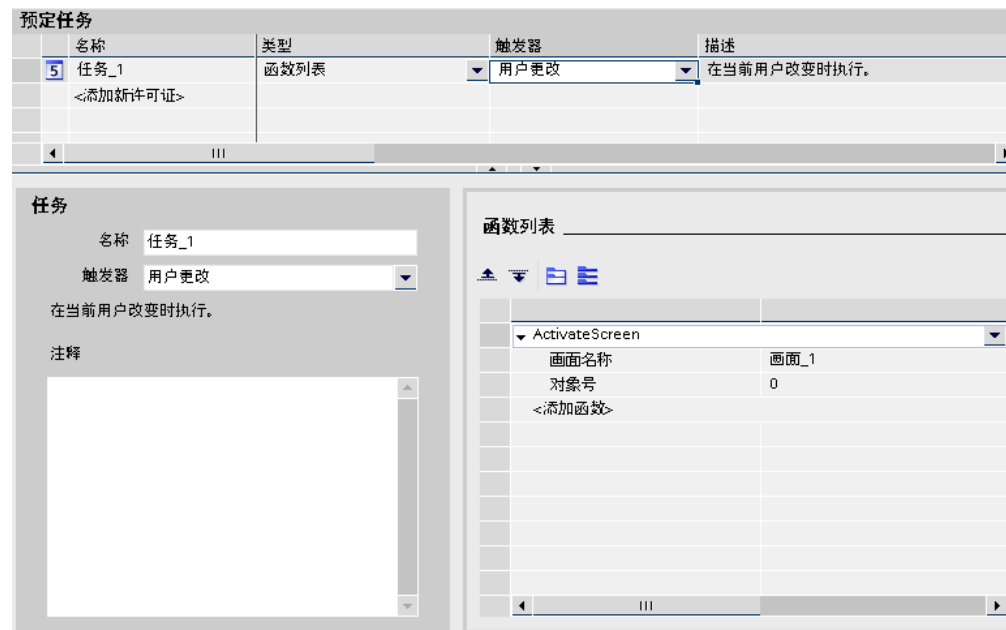


图 8-1

启动任务

由触发源控制，调度程序启动连接到触发源的任务。

参见

调度程序的应用领域 (页 1294)
示例：用户更改后更新用户 (页 1299)

8.8.3 元素和基本设置

8.8.3.1 调度程序

简介

在“调度程序”中，通过组态触发源和任务类型来规划任务。

打开

双击“调度程序”以在项目视图中将其打开。

工作区

工作区显示计划好的任务。

参见

[“调度程序”编辑器的工作区域 \(页 1296\)](#)

[使用事件触发规划任务 \(页 1298\)](#)

[调度程序的应用领域 \(页 1294\)](#)

[函数列表 \(页 1297\)](#)

[触发源 \(页 1297\)](#)

示例：[用户更改后更新用户 \(页 1299\)](#)

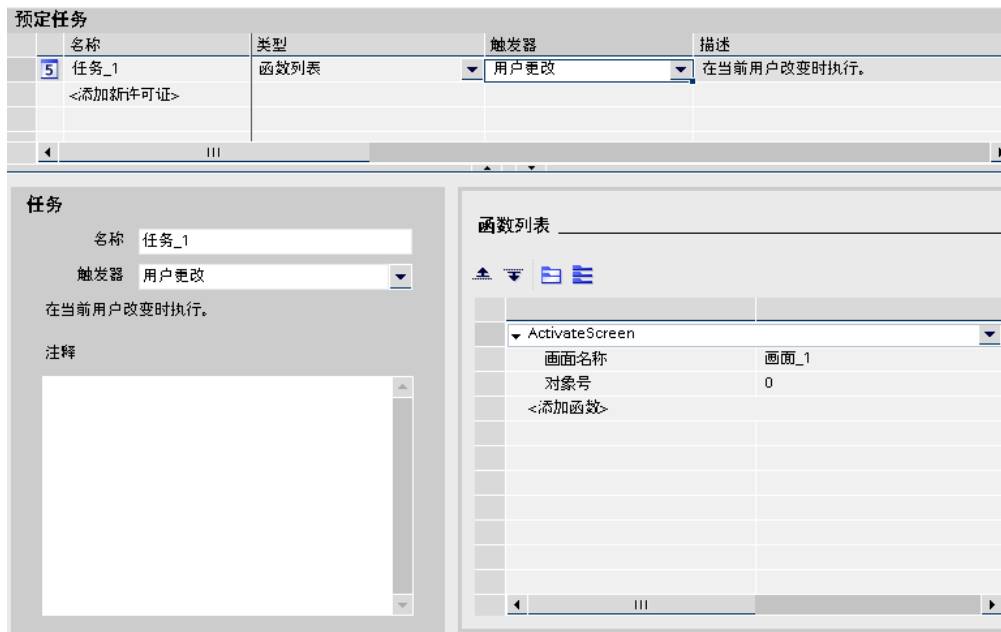
8.8.3.2 “调度程序”编辑器的工作区域

简介

工作区显示计划的任務，任务由触发源和任务类型（如函数列表）组成。

结构

工作区包括任务表、“调度程序”区域和函数列表。



任务表显示指定任务及其属性，如触发源。选择任务类型和触发源。将名称和注释分配到任务。调度程序编译任务的说明。

“调度程序”区域还将任务与触发源一起显示。“调度程序”区域随所选的触发源的不同而有所变化。

使用函数列表组态在任务中执行的系统函数。

注意

编译的说明提供任务简介，包括任务触发的时机。可以使用工具提示获得有关用户界面元素的更详细信息。为此，将鼠标指针移动到所选元素上方。

参见

调度程序 (页 1295)

示例：用户更改后更新用户 (页 1299)

系统函数 (页 1244)

8.8.3.3 函数列表**函数列表**

由“触发器”启动函数列表。函数列表是逐行执行的。每行包含一个系统函数。可为每个任务恰好组态一个函数列表。

注意

函数列表中可以组态的系统函数取决于选择的 HMI 设备。

参见

调度程序 (页 1295)

示例：用户更改后更新用户 (页 1299)

8.8.3.4 触发源**简介**

触发源与任务相链接，并构成任务的触发事件。在发生触发事件时执行该任务。

事件触发

任务链接到系统事件时，将由事件触发任务。例如，运行系统停止是系统事件。

禁用任务

例如，在重新组态期间将暂时禁用与此触发源关联的任务。在工程系统中而不是在运行系统中禁用任务。

也可以使用该触发源使先前已组态的系统事件可用。实例：将系统事件“关闭”计划用于任务“A”。该系统事件对于另一任务“B”就不再可用。选择“禁用”作为任务“A”的触发源，可使“运行系统停止”系统事件再次可用。

注意

可用的系统事件和触发源取决于正在使用的 HMI 设备。

参见

事件 (页 1287)

调度程序 (页 1295)

示例：用户更改后更新用户 (页 1299)

8.8.3.5 使用事件触发规划任务

简介

规划一项在用户更改时生成画面切换的任务。

要求

- “调度程序”工作区域已打开。
- 已创建“启动”画面。

步骤

1. 在任务区域的表中单击“<添加>”。
2. 输入“用户更改时画面切换”作为“名称”。
3. 选择“用户更改”作为“触发源”。
4. 在“函数列表”表格中的第一行上，单击“<添加函数>”条目。
5. 选择“Screens/ActivateScreen”作为函数。
6. 在画面名称域中选择“启动”画面。

结果

使用“用户更改”事件执行任务。新用户成功登录后，将调用“启动”画面。

参见

调度程序 (页 1295)

示例：用户更改后更新用户 (页 1299)

8.8.4 实例

8.8.4.1 示例：用户更改后更新用户

任务

组态显示已登录用户的 I/O 域。组态一个在登录用户更改时更新 I/O 域的任务。

要求

- 已创建了具有运行系统高级版和面板的 HMI 设备。
- 创建了“String”类型的“CurrentUser”变量。
- 画面已创建并打开。
- 已在画面中创建了 I/O 域。

步骤

1. 单击“I/O 域”对象。
2. 在巡视窗口的“属性”选项卡中单击“常规”组。
 - 选择“字符串”作为“显示格式”。
 - 选择“CurrentUser”作为“变量”。
 - 选择“输出”作为模式。
3. 转到“调度程序”的工作区域。
4. 在任务区域的表中单击“添加...”。
5. 输入“CurrentUser”作为“名称”。
6. 选择“用户更改”作为“触发源”。
7. 在“函数列表”表的第一行中，单击条目“无函数”。
8. 从“用户管理”组选择系统函数“ReadUserName”。
9. 选择“CurrentUser”作为“变量”。

结果

新用户成功登录后，将调用“ReadUserChange”函数。“CurrentUser”变量将更新并显示在新登录用户的 I/O 域中。

如果某个用户未成功登录，则该用户将被退出。I/O 域中将继续显示以前登录的用户，直到有新用户成功登录为止。

参见

使用任务和触发源 (页 1295)
调度程序的应用领域 (页 1294)
调度程序 (页 1295)
“调度程序”编辑器的工作区域 (页 1296)

函数列表 (页 1297)
触发源 (页 1297)
使用事件触发规划任务 (页 1298)

8.9 使用连接

8.9.1 基础知识

8.9.1.1 通信的基础知识

简介

两个通信伙伴之间的数据交换被称为通信。通信伙伴可以通过直连电缆连接或网络互连。

通信伙伴

通信伙伴可以是任何能与网络中其它节点通信和交换数据的节点。在 WinCC 环境中，下列节点可作为通信伙伴：

- 自动化系统中的中央模块和通信模块
- HMI 设备和 PC 中的通信处理器

通信伙伴间传送的数据可以用于不同用途：

- 过程控制
- 过程数据采集
- 报告过程中的状态

参见

通信原理 (页 1300)
HMI 上的 SNMP 和 MIB (页 1302)
组态连接 (页 1303)
连接的区域指针 (页 1309)
调试组件 (页 1328)
联网设备 (页 1302)

8.9.1.2 通信原理

简介

WinCC 使用变量和区域指针控制 HMI 设备和 PLC 之间的通信。

使用变量通信

在 WinCC 中，在“变量”编辑器中集中管理变量。变量包括外部变量和内部变量。外部变量用于通信，代表 PLC 上已定义内存位置的映像。HMI 和 PLC 都可以对此存储位置进行读写访问。这些读写操作可以是周期性的，也可以是事件触发的。

在组态中，创建指向特定 PLC 地址的变量。HMI 从已定义地址读取该值，然后将其显示出来。操作员还可以在 HMI 设备上输入值，随后这些值会写入相关 PLC 地址。

使用区域指针通信

区域指针在“连接”编辑器中集中管理。区域指针用于交换特定用户数据区的数据。区域指针是参数域。在运行系统中，WinCC 接收来自这些参数域的、有关 PLC 中数据区位置和大小信息。在通信过程中，PLC 和 HMI 设备交替访问这些数据区，以进行读、写操作。根据对存储在这些数据区中的数据进行分析，PLC 和 HMI 设备触发一些定义的操作。

WinCC 使用以下区域指针：

- 数据记录
- 日期/时间
- 协调
- 控制请求
- 日期/时间 PLC
- 项目 ID
- 画面号

各种区域指针的可用性由所用的 HMI 决定。

WinCC 和自动化系统之间的通信

使用 WinCC 进行工业通信意味着数据是使用变量和区域指针进行交换。要采集数据，HMI 用通信驱动程序将请求消息发送到自动化系统。自动化系统(AS)在响应帧中将请求的数据返回 HMI。

通信驱动程序

通信驱动程序是在自动化系统和 HMI 设备之间建立连接的软件组件。因此，通信驱动程序可为 WinCC 中的变量提供过程值。

可以根据所采用的 HMI 设备和连接的通信伙伴选择接口、配置文件和传输速度。

参见

通信的基础知识 (页 1300)

8.9.1.3 HMI 上的 SNMP 和 MIB

SNMP

SNMP (简单网络管理协议) 是用于监视网络组件或终端设备 (例如, HMI 设备) 的 Internet 标准协议。SNMP 是 TCP/IP 协议组的一部分, 以客户机/服务器模式工作。

SNMP 由 Internet 工程任务小组 (IETF, Internet Engineering Task Force) 开发。HMI 设备支持版本 1 (SNMPv1) 和基于团体的 SNMP 版本 2 (SNMPv2c)。

HMI 设备有几种 SNMP 代理。SNMP 代理提供有关设备组态的信息。此信息在数据结构 (即管理信息库 (MIB, Management Information Base)) 中进行管理。

MIB

MIB 是由多种不同 SNMP 变量组成的标准化数据结构。这些设备使用 MIB II (RFC1213)。支持“公共”团体读写 HMI 设备上的 SNMP 变量。

参见

通信的基础知识 (页 1300)

8.9.2 使用连接

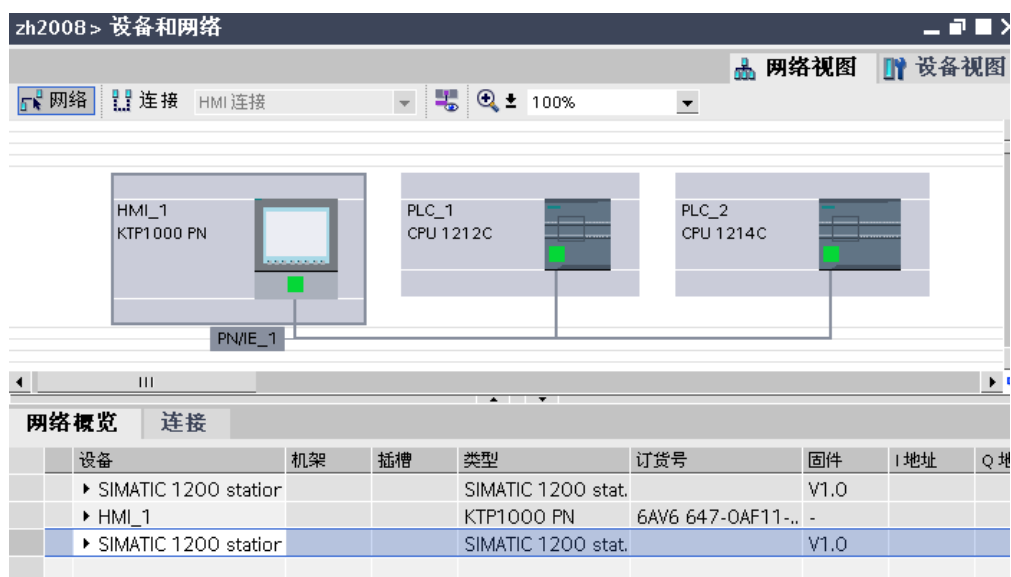
8.9.2.1 联网设备

简介

提供用于组态连接的“设备和网络”编辑器。可以在编辑器中联网设备。也可以为设备和接口组态和分配参数。随后组态联网设备之间所需的连接。

联网设备

“设备和网络”为网络组态提供图形区域和表格区域。在图形区域中, 可以使用拖放方法联网项目中的设备。表格区域提供设备及其组件的概述。



联网设备时自动创建所需子网。自动为接口设置一致的地址参数。

用绿线表示设备联网。显示所用子网的名称。

更多信息，请参见“[联网设备 \(页 204\)](#)”部分。

参见

[联网设备 \(页 204\)](#)
[组态连接 \(页 1303\)](#)
[通信的基础知识 \(页 1300\)](#)

8.9.2.2 组态连接

简介

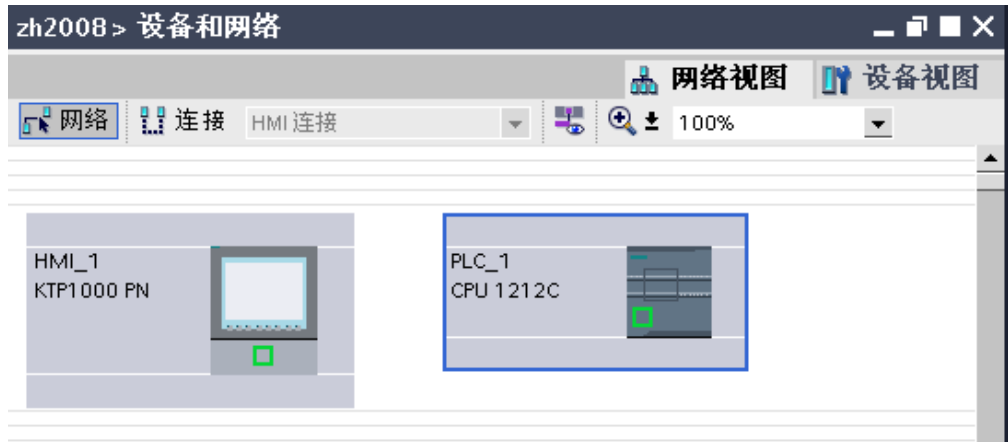
可以在“设备和网络”编辑器中组态与通信伙伴之间的连接。要连接的设备必须具有相同类型的接口。

要求

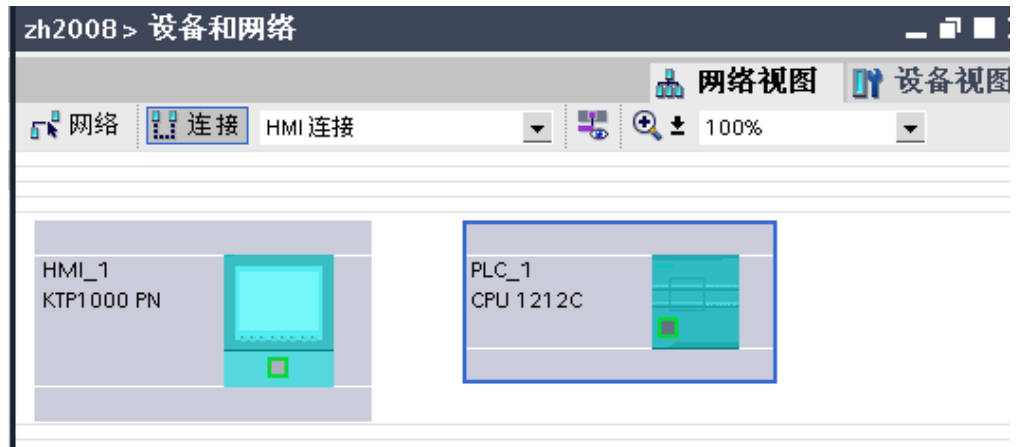
- 项目已在 WinCC 中创建并打开。
- 通信伙伴已插入项目。

步骤

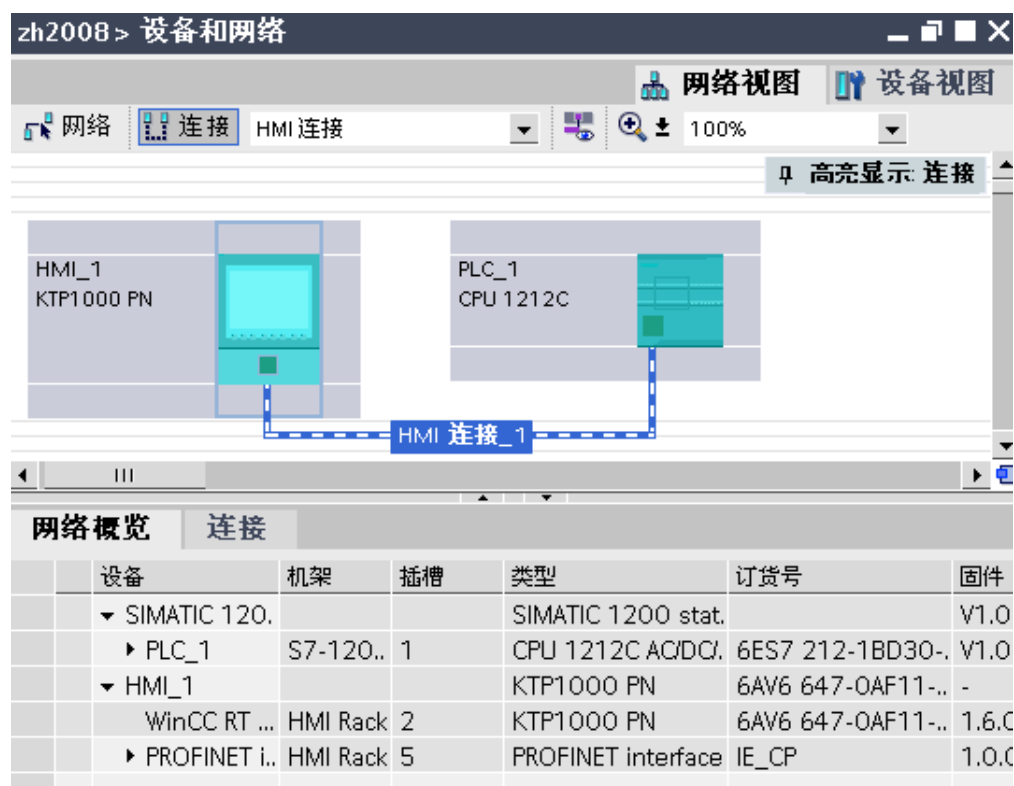
1. 在项目树中双击“设备和网络”项。将打开具有此名称的编辑器。
2. 在网络视图以图形形式显示项目中可用的通信伙伴。



3. 单击“连接”选项卡。 可用于连接的设备将以高亮颜色显示。



4. 将鼠标指针放在代表第一个通信伙伴的绿色方块上，按住鼠标左键，将连接拖动至所需的通信伙伴。将创建连接，同时自动创建相应的子网和适合于连接的参数。连接包含一个局部连接名称，作为唯一标识连接的标识符。



5. 在网络视图中，单击代表连接伙伴的绿色方块。在巡视窗口的“属性”>“以太网地址”下显示 IP 地址和子网掩码。根据要求更改地址参数。
6. 对于每个附加连接伙伴，重复步骤 5。
要上载项目，在 HMI 设备上设置的地址参数必须与 WinCC 项目中的地址参数相同。更多信息，请参考 [装载设置 \(页 1335\)](#)。
进行更改时确保在网络内保持一致。
7. 要打开编辑器的表格区域，单击图形区域下方的小箭头按钮。



8. 创建的连接也显示在编辑器的表格区域中的“连接”选项卡上。使用表格监视连接参数和更改连接伙伴。只能在表格中更改连接的局部名称。
9. 保存项目。

还可以使用“创建新连接”对话框组态连接。为此，在“连接”选项卡的图形区域中选择一个以高亮颜色显示的设备，然后从快捷菜单中选择“创建新连接”命令。

结果

新连接已创建。连接参数已组态。



小心 以太网通信

在基于以太网的通信中，是由最终用户负责其数据网络的安全。不能保证设备在所有情况下都能正常运行；例如，遭受蓄意攻击就会导致设备过载。

参见

自动创建连接 (页 1306)

优化组态 (页 1308)

联网设备 (页 1302)

8.9.2.3 自动创建连接

简介

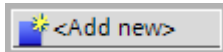
可以使用 WinCC 自动创建和组态 HMI 设备和 PLC 之间的连接。

要求

- 已创建 WinCC 项目。
- 已经打开项目。
- 项目包含 SIMATIC PLC。

使用设备向导创建连接

1. 在起始视图中，单击“开始”>“可视化”，然后单击



按钮打开“添加新设备”对话框。

2. 选择所需 HMI 设备，然后选择“启动设备向导”选项。
3. 单击“确定”。设备向导随即打开。



4. 在“PLC 连接”页面上，打开选择对话框并选择 SIMATIC PLC。
5. 确认对 PLC 的选择。将创建设备之间的连接。

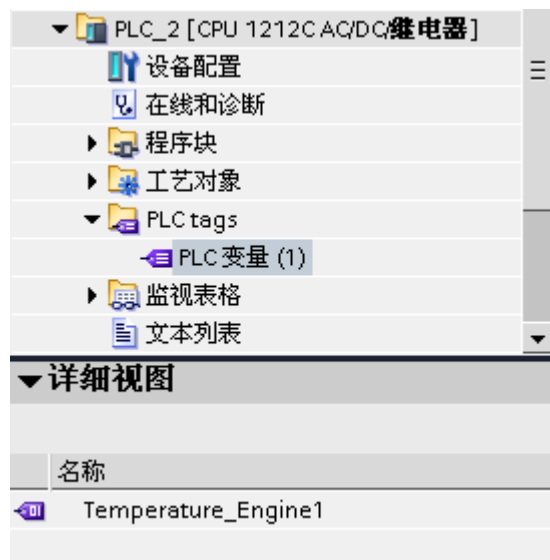


6. 根据需要在设备向导中进行附加设置，然后单击“完成”。
在设备向导中进行的设置将被应用于项目。

使用 PLC 变量创建连接

如果 PLC 变量在项目中可用，则可使用该变量在画面中自动创建连接。

1. 在“画面”编辑器中打开所需画面。
2. 在项目树中选择“PLC 变量”文件夹。现有 PLC 变量将显示在详细视图中。



3. 将 PLC 变量从详细视图拖动到打开的画面中。
将在画面中创建一个 I/O 域，并通过 HMI 变量链接到 PLC 变量。还将自动创建 HMI 设备和 PLC 之间的连接。

参见

组态连接 (页 1303)

8.9.2.4 优化组态

采集周期和更新时间

在工程软件中所定义的“区域指针”和变量的采集周期是可获得的实际更新时间的决定性因素。

更新时间等于采集周期 + 传送时间 + 处理时间。

要获得最佳的更新时间，在组态期间请记住以下几点：

- 根据需要设置数据区的最大和最小值。
- 定义相关的数据区。可通过设置一个较大数据区域而不是若干小区域来优化更新时间。
- 如果所选择的采集周期太短，将会不利于整体性能。对采集周期进行设置，使其适合过程值的改变速率。例如，炉温变化速度比电驱动速度慢得多。采集周期的指导值：约为 1 秒。
- 要优化更新时间，可以不进行大量变量的周期性传送。这样采用的将不是周期传送，而是受事件控制的自发且有目的的变量传输方式（作业信箱）。
- 将报警或画面的变量不留间隙地写入单个数据区中。
- 要想可靠地识别 PLC 中的数据变化，选用的采样周期时间必须小于实际数据变化的时间周期。
- 将传输率设置为可能的最高值。

画面

切实可行的画面更新率取决于要可视化的数据类型和数据量。

只对实际需要更短刷新周期的对象组态短采集周期。

趋势

使用位触发的趋势时，如果在“趋势传送区”(Trend transfer area) 中设置了组位，则 HMI 设备始终更新在此区域中设置了其位的所有趋势。然后将这些位复位。

只有当 HMI 复位了所有位之后，才可重新设置 S7 程序中的组位。

注意

可用于特定设备

位触发的趋势不可用于精简面板。

作业信箱

如果连续而快速地传送大量的作业信箱，将会导致 HMI 和 PLC 之间的通信过载。

HMI 设备通过在作业信箱的第一个数据字中输入数值 0 来确认接收到作业信箱。现在，HMI 处理作业，这需要更长的时间。如果立即在作业信箱中再输入新作业，则可能需要过一段时间 HMI 设备才能处理下一个作业。下一个作业信箱只有在系统提供足够的计算能力时才会被接受。

TCP/IP (以太网)的超时响应

由于使用了 TCP/IP 协议，大约一分钟后才可检测到连接故障。如果不请求变量（例如，当前画面中无输出变量），则不能可靠地检测通信故障。

为每个 PLC 组态区域指针协调。这确保了在通信故障发生后大约两分钟内即可识别到，即使是在上述情景中也是如此。

参见

组态连接 (页 1303)

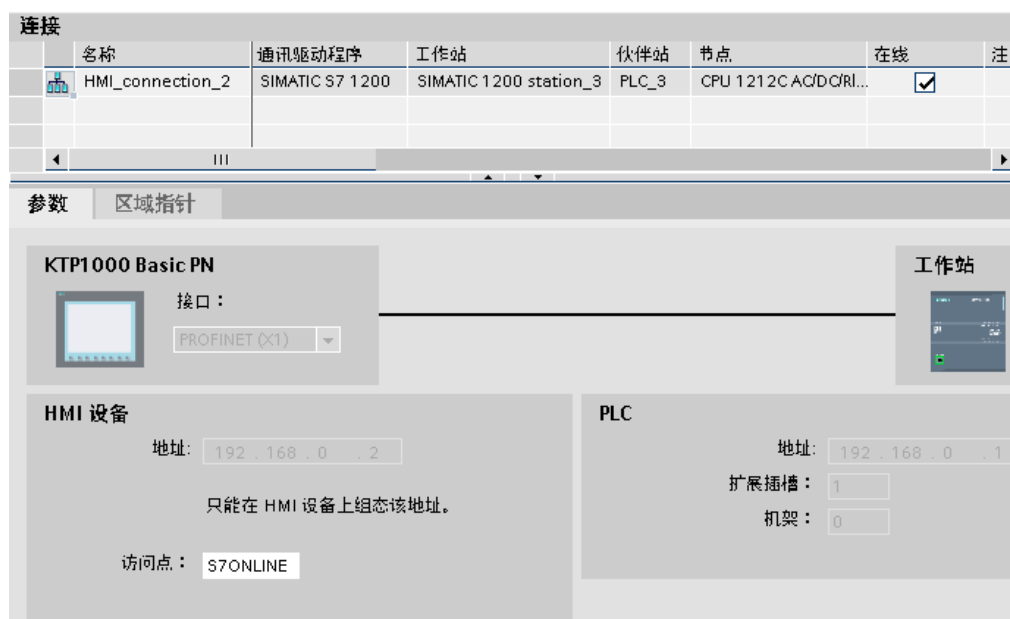
8.9.3 用户数据区

8.9.3.1 连接的区域指针

简介

使用“连接”编辑器的“区域指针”选项卡可以组态可用区域指针的使用。

要组态区域指针，打开“连接”编辑器，然后打开“区域指针”选项卡。



结构

“区域指针”选项卡含有两个区域指针表格。表格的上面部分包含可以为每个可用连接单独创建和启用的区域指针。

“HMI 设备的全局区域指针”表格包含只在项目中创建一次且只能用于一个连接的区域指针。

参数		区域指针					
激活		显示名称	PLC 变量	地址	长度	采集模式	采集周期
	<input type="checkbox"/>	作业邮箱			4	循环连续	<未定义>
	<input type="checkbox"/>	协调			1	循环连续	<未定义>
	<input type="checkbox"/>	数据记录			5	循环连续	<未定义>
	<input type="checkbox"/>	日期/时间			6	循环连续	<未定义>

HMI 设备的全局区域指针							
连接		显示名称	PLC 变量	地址	长度	采集模式	采集周期
<未定义>	...	日期/时间 PLC			6	循环连续	<未定义>
<未定义>		画面号			5	循环连续	<未定义>
<未定义>		项目 ID			1	循环连续	<未定义>

更多信息，请参见“关于区域指针的常规信息 (页 1312)”。

参见

[通信的基础知识 \(页 1300\)](#)

[组态区域指针 \(页 1310\)](#)

8.9.3.2 组态区域指针

简介

使用区域指针访问 PLC 中的数据区。数据区保存在 PLC 中。区域指针长度大于或等于 1 时，可以将对应的数据区设为全局数据块或背景数据块中的变量数组。如果区域指针的长度为 1，则可以使用 PLC 变量。根据要使用的区域指针的长度组态数据块中的变量。区域指针的长度单位是 16 位字。例如，如果要使用长度为“5”的区域指针，则需要在数据块中创建包含 5 个元素的数组。有关其它详细信息，请参见“编程 PLC”部分中的“编程数据块”。

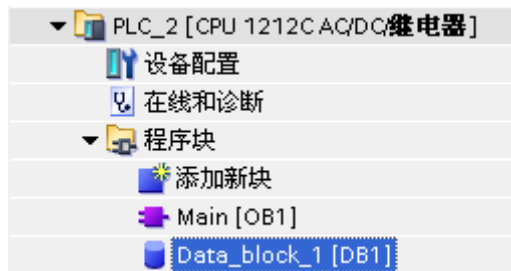
要求

- 项目包含 PLC。
- 已组态 PLC 和 HMI 设备之间的连接。
- PLC 程序包含全局数据块。

步骤

要组态区域指针，请执行以下步骤：

1. 在项目树中展开 PLC 视图，打开“程序块”文件夹，然后双击全局数据块。数据块将打开。



2. 在“名称”列中输入变量名称。
3. 在“数据类型”列中选择“type 类型的 Array[lo .. hi]”作为数据类型。用数组的维数的下限值和上限值替换方括号中的“lo”和“hi”条目。

实例：如果组态长度为“4”的区域指针，则输入值“0”替换方括号中的“lo”，输入值“3”替换“hi”。

4. 用“word”数据类型替换“type”名称。含有 4 个变量的数组的完整数据类型如下所示：“word 类型的 Array[0 .. 3]”。

变量数组将在确认了输入内容之后进行创建。

Data_block_1					
	名称	数据类型	初始值	保持性	注释
1	▼ Static			<input type="checkbox"/>	
2	▼ Job_mailbox	Array [0 .. 3] of word		<input type="checkbox"/>	
3	Job_mailbox[0]	Word	W#16#0000	<input type="checkbox"/>	
4	Job_mailbox[1]	Word	W#16#0000	<input type="checkbox"/>	
5	Job_mailbox[2]	Word	W#16#0000	<input type="checkbox"/>	
6	Job_mailbox[3]	Word	W#16#0000	<input type="checkbox"/>	
7				<input type="checkbox"/>	

5. 在项目树中选择 PLC，然后在快捷菜单中选择“编译”>“软件”命令。数据块将被编译。编译之后，变量数组可在项目中进一步使用。
6. 在项目树中展开 HMI 设备的视图，然后双击“连接”条目。“连接”编辑器将打开。



7. 打开“区域指针”选项卡并启用所需区域指针。通过在“连接”域中选择连接来启用全局区域指针。
8. 单击“PLC 变量”域中的浏览按钮。对象列表将打开。在对象列表中浏览到数据块，然后在右侧窗口中选择变量。

参数		区域指针				
激活	显示名称	PLC 变量	地址	长度	采集模式	
<input type="checkbox"/>	日期/时间			6	循环连续	
<input type="checkbox"/>	数据记录			5	循环连续	
<input type="checkbox"/>	协调			1	循环连续	
<input checked="" type="checkbox"/>	作业邮箱	Data_block_1_Job_mailbox ...	<symbolic access>	4	循环连续	

HMI 设备的全局区域指针						
连接	显示名称	PLC 变量	地址	长度	采集模式	采集周期
<未定义> ...	画面号			5	循环连续	<未定义>
<未定义>	日期/时间 PLC			6	循环连续	<未定义>
<未定义>	项目 ID			1	循环连续	<未定义>

组态长度为“1”的区域指针时不需要数组变量。选择“Word”数据类型以在数据块中创建变量。在组态期间设置附加参数，如采集周期。

结果

区域指针将被启用并被链接至全局数据块中的 PLC 变量。

参见

连接的区域指针 (页 1309)

8.9.3.3 区域指针

8.9.3.3 关于区域指针的常规信息

简介

使用区域指针访问 PLC 中的数据区。PLC 和 HMI 设备通过读写这些数据区域的数据进行交互通信。根据对存储的数据进行分析，PLC 和 HMI 设备可触发定义的交互操作。

精简面板支持下列区域指针：

- 作业信箱
- 项目 ID
- 画面编号
- 数据记录
- 日期/时间
- 日期/时间 PLC
- 协调

应用

在使用区域指针之前，应在“连接”►“区域指针”中组态并启用该区域指针。

参数		区域指针					
激活	显示名称	PLC 变量	地址	长度	采集模式	采集周期	
<input type="checkbox"/>	作业信箱			4	循环连续	<未定义>	
<input type="checkbox"/>	协调			1	循环连续	<未定义>	
<input type="checkbox"/>	数据记录			5	循环连续	<未定义>	
<input type="checkbox"/>	日期/时间			6	循环连续	<未定义>	

HMI 设备的全局区域指针							
连接	显示名称	PLC 变量	地址	长度	采集模式	采集周期	
<未定义>	日期/时间 PLC			6	循环连续	<未定义>	
<未定义>	画面号			5	循环连续	<未定义>	
<未定义>	项目 ID			1	循环连续	<未定义>	

- 激活
启用区域指针。
- 显示名称
由 WinCC 指定的区域指针的名称。
- 控制变量
在此处选择您已将其组态为区域指针的数据区的 PLC 变量或变量数组。
- 地址
由于符号访问的原因，不会在此域中输入地址。
- 长度
WinCC 指定区域指针的长度。
- 采集周期
在此域中定义一个采集周期，以允许 HMI 设备周期性地读取区域指针。极短的采集时间可能会对 HMI 设备性能有负面影响。
- 注释
储存注释，例如对区域指针的使用情况进行描述。

访问数据区

此表介绍了 PLC 和 HMI 设备对数据区的读 (R) 和写 (W) 访问。

数据区	适用操作	HMI 设备	PLC
画面号	由 PLC 进行评估以确定活动的画面。	W	R
数据记录	同步传送数据记录	R/W	R/W
日期/时间	将日期和时间由 HMI 设备传送至 PLC	W	R
日期/时间 PLC	将日期和时间由 PLC 传送至 HMI 设备	R	W
协调	用控制程序请求 HMI 设备状态	W	R
项目 ID	运行系统检查 WinCC 项目 ID 与 PLC 中的项目 ID 是否一致。	R	W
作业信箱	通过控制程序触发 HMI 设备功能	R/W	R/W

以下部分将介绍区域指针及其关联的作业信箱。

参见

- “画面编号”区域指针 (页 1314)
- “项目 ID”区域指针 (页 1318)
- “作业信箱”区域指针 (页 1319)
- “数据信箱”区域指针 (页 1321)
- 连接的区域指针 (页 1309)
- 组态区域指针 (页 1310)
- “日期/时间”区域指针 (页 1315)
- “日期/时间 PLC”区域指针 (页 1316)
- “协调”区域指针 (页 1317)

8.9.3.3 “画面编号”区域指针

功能

HMI 设备将 HMI 设备上调用的画面的信息存储在“画面编号”区域指针中。

这允许将当前画面的内容与 HMI 设备传送到 PLC。然后，PLC 可触发特定的反应，比如调用不同的画面。

应用

必须首先通过选择“通信”▶“区域指针”来设置并激活“画面编号”区域指针，然后才能使用它。您只能创建“画面号”(Screen number) 区域指针的一个实例和一个 PLC。

画面号会自动传送给 PLC。也就是说，当在 HMI 设备上激活新画面时，新的画面总是会传送到 PLC。因此，不必组态采集周期。

结构

区域指针是 PLC 内存中的一个数据区，它具有 5 个字长的固定长度。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1. 字	当前画面类型															
2. 字	当前画面号															
3. 字	保留															
第 4 个字	当前域号															
5. 字	保留															

- 当前画面类型
 - “1”表示根画面，或
 - “4”表示固定窗口
- 当前画面号
 - 1 至 32767
- 当前域号

1 至 32767

注意
可用于特定设备
固定窗口不可用于精简面板。

参见

关于区域指针的常规信息 (页 1312)

8.9.3.3 “日期/时间”区域指针

功能

该区域指针用于将日期和时间从 HMI 设备传送到 PLC。

PLC 将控制作业“41”写入作业信箱。

当判断控制作业时，HMI 设备将其当前日期和时间写入在“日期/时间”区域指针中组态的数据区内。

将数据类型“DTL”与通信驱动程序 S7 1200 一起使用。数据类型“DTL”的变量的长度为 12 字节，并在预定义的结构中保存日期和时间信息。

“DTL”数据类型具有下列结构：

Byte	组成部分	数据类型	取值范围
0	年	UINT	1970 至 2554
1			
2	月	USINT	0 至 12
3	变量	USINT	1 至 31
4	星期	USINT	1 (星期日) 至 7 (星期六) 在值输入中不考虑工作日。
5	小时	USINT	0 至 23
6	分钟	USINT	0 至 59
7	秒钟	USINT	0 至 59
8	纳秒	UDINT	0 至 999,999,999
9			
10			
11			

参见

关于区域指针的常规信息 (页 1312)

8.9.3.3 “日期/时间 PLC”区域指针

功能

该区域指针用于将日期和时间从 PLC 传送到 HMI 设备。如果 PLC 为时间主站，则使用该区域指针。

PLC 装载该区域指针的数据区。

HMI 设备在组态的采集时间周期内读取数据，并自行同步。

注意

为日期/时间 PLC 区域指针设置足够长的采集周期以避免对 HMI 设备的性能造成负面影响。
建议：如果过程允许，可将采集周期设置为 1 分钟。

日期/时间 PLC 是全局区域指针，只能在项目中组态一次。

将数据类型“DTL”与通信驱动程序 S7 1200 一起使用。数据类型“DTL”的变量的长度为 12 字节，并在预定义的结构中保存日期和时间信息。

“DTL”数据类型具有下列结构：

Byte	组成部分	数据类型	取值范围
0	年	UINT	1970 至 2554
1			
2	月	USINT	0 至 12
3	变量	USINT	1 至 31
4	星期	USINT	1 (星期日) 至 7 (星期六) 在值输入中不考虑工作日。
5	小时	USINT	0 至 23
6	分钟	USINT	0 至 59
7	秒钟	USINT	0 至 59
8	纳秒	UDINT	0 至 999 999 999
9			
10			
11			

HMI 设备不支持使用纳秒。在运行系统中处理时，将忽略纳秒级的值。

参见

关于区域指针的常规信息 (页 1312)

8.9.3.3 “协调”区域指针

功能

“协调”区域指针用于实现以下功能：

- 在控制程序中检测 HMI 设备的启动
- 在控制程序中检测 HMI 设备的当前操作模式
- 在控制程序中检测 HMI 设备是否做好了通信准备

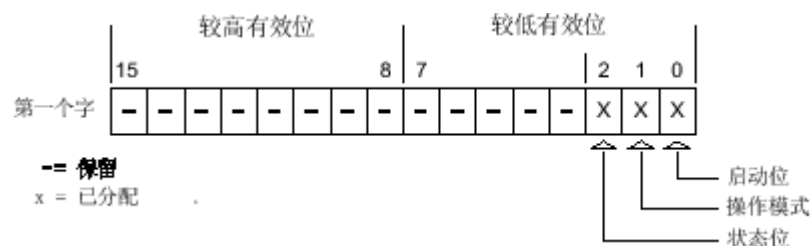
“协调”区域指针的长度为一个字。

应用

注意

每当 HMI 设备更新区域指针后，总是写入整个协调区域。
因此，PLC 程序不能在协调区域进行任何修改。

“协调”区域指针中的位分配



启动位

在启动过程中，HMI 设备将启动位暂时设置为“0”。启动后，将该位永久地设置为“1”。

操作模式

当用户将 HMI 设备切换为离线后，操作模式位将被设置为 1。在 HMI 设备的正常操作中，操作模式位的状态为“0”。可通过查询此位来了解 HMI 设备的当前操作模式。

状态位

HMI 设备以约为一秒的时间间隔对状态位取反。可通过在 PLC 程序中查询此位来检查与 HMI 设备的连接是否仍然存在。

在 PLC 中处理

对于 PLC 程序中的较简单评估，在使用 SIMATIC S7 1200 通信驱动程序时将 Bool 数组用于此区域指针。必须映射完整的 16 位字区域指针。为此，组态“bool 类型的 Array [0 .. 15]”数据类型的变量。

参见

关于区域指针的常规信息 (页 1312)

8.9.3.3 “项目 ID”区域指针

功能

运行系统启动时，它会检查 HMI 设备是否连接到了正确的 PLC。当对多台 HMI 设备进行操作时，该检查非常重要。

HMI 设备将 PLC 中所存储的值与组态数据中的指定值进行比较。这样可确保组态数据和 PLC 程序的兼容性。

缺少兼容性将导致相应的报警，且运行系统将不会启动。

应用

为了使用该区域指针，必须在组态时进行如下设置：

- 指定组态数据的版本。允许值在 1 和 255 之间。
在“设备设置”▶“画面”编辑器的“项目 ID”中输入该版本。
- 在此处选择您已将其组态为区域指针的数据区的 PLC 变量或变量数组。

连接失败

与组态了“项目 ID”区域指针的设备之间的连接失败将会导致项目中的所有其它连接都切换为“离线”。

此行为有以下要求：

- 项目中组态了多个连接。
- 至少在一个连接中使用了“项目 ID”区域指针。

使连接“离线”的原因：

- 找不到 PLC。
- 已在工程系统中将连接切换为离线。

参见

关于区域指针的常规信息 (页 1312)

8.9.3.3 “作业信箱”区域指针

功能

PLC 可使用作业信箱将作业传送到 HMI 设备以在 HMI 设备上触发相应的操作。包括的功能举例如下：

- 显示画面
- 设置日期和时间

数据结构

作业信箱的第一个字中含有作业号。根据作业信箱的不同，最多可传送三个参数。

字	最高有效位	最低有效位
n+0	0	作业号
n+1	参数 1	
n+2	参数 2	
n+3	参数 3	

如果作业的第一个字不等于 0，则 HMI 设备会对作业信箱进行评估。这就意味着，必须首先在作业信箱中输入参数，然后再输入作业号。

当 HMI 设备接受该作业信箱时，第一个字将被重新设置为 0。作业信箱的执行通常不会在此时完成。

作业信箱

所有作业邮箱及其参数都列在下面。“编号”列中为作业信箱的作业号。仅当 HMI 设备在线时才能由 PLC 触发作业信箱。

编号	功能	
14	设置时间(以 BCD 码编码)	
	参数 1	左字节：- 右字节：小时(0-23)
	参数 2	左字节：分钟(0-59) 右字节：秒钟(0-59)
	参数 3	-
15	设置日期(以 BCD 码编码)	
	参数 1	左字节：- 右字节：星期 (1-7：星期天-星期六)
	参数 2	左字节：日期(1-31) 右字节：月份(1-12)
	参数 3	左字节：年份

编号	功能	
14	设置时间(以 BCD 码编码)	
23	用户登录	
	用户以“PLC user”名称登录 HMI 设备，组编号在参数 1 中传输。 只有项目中存在传送的组号时，才能登录。	
	参数 1	组号 1 - 255
	参数 2、3	-
24	用户退出	
	退出当前登录的用户。 (该功能对应于系统函数“logoff”)	
	参数 1、2、3	-
40	将日期/时间传送到 PLC	
	两个连续作业之间至少应间隔 5 秒，否则 HMI 设备将过载。	
	参数 1、2、3	-
41	将日期/时间传送到 PLC	
	两个连续作业之间至少应间隔 5 秒，否则 HMI 设备将过载。	
	参数 1、2、3	-
46	更新变量	
	使 HMI 设备从 PLC 中读取其更新 ID 与参数 1 中所传送的值相匹配的变量的当前值。 (功能对应于系统函数“UpdateTag”。)	
	参数 1	1 - 100
49	清除事件缓存	
	参数 1、2、3	-
50	清除出错报警缓存	
	参数 1、2、3	-
51	显示选择	
	参数 1	画面号
	参数 2	-
	参数 3	域号
69	从 PLC 中读取数据记录 ¹⁾	
	参数 1	配方号 (1-999)
	参数 2	数据记录号 (1-65535)
	参数 3	0：不覆盖现有数据记录 1：覆盖现有数据记录
70	写入来自 PLC 的数据记录 ¹⁾	

编号	功能	
14	设置时间(以 BCD 码编码)	
	参数 1	配方号 (1-999)
	参数 2	数据记录号 (1-65535)
	参数 3	-

参见

关于区域指针的常规信息 (页 1312)

8.9.3.3 “数据记录”区域指针

8.9.3.3 “数据信箱”区域指针

功能

在 HMI 设备和 PLC 之间传送数据记录时，两个伙伴都可以在控制器上访问公共通信区。

数据传送类型

在 HMI 设备和 PLC 之间传送数据记录有以下两种方法：

- 非同步传送
- 同步传送数据记录

数据记录始终直接传送。即，可直接将变量值写入到为该变量组态的地址或从相应的地址读出，而无需通过中间存储器来重新定位数据值。

启动数据记录的传送

触发传送有以下三种方法：

- 操作员在配方视图输入
- 作业信箱
也可由 PLC 触发数据记录的传送。
- 由组态的函数触发

如果通过作业信箱触发数据记录传送，也会更新配方视图中的数据。因此，通过触发 PLC 作业传送数据记录时应避免操作配方视图。如果已开始编辑数据记录并触发 PLC 作业以传送数据记录，则此 PLC 作业将会遭到拒绝。

参见

关于区域指针的常规信息 (页 1312)

非同步传送 (页 1322)

同步传送 (页 1322)

操作员在配方显示中启动的传送顺序 (页 1324)
由作业信箱 SIMATIC S7 触发后的传送顺序 (页 1325)
由组态的函数触发后的传送顺序 (页 1326)
传送数据记录时出错的可能原因 (页 1327)

8.9.3.3 非同步传送

如果选择在 HMI 设备和 PLC 之间异步传送数据记录，则无需通过公用数据区执行协调。因此，不必在组态期间设置数据区。

异步数据记录传送是一种非常有用的备选方法，例如在以下情况下：

- 系统能够排除通讯伙伴任意覆盖数据的风险。
- PLC 不需要有关配方号和数据记录的信息。
- 数据记录的传送由操作员在 HMI 设备上触发。

读取值

触发读取作业后，从 PLC 地址读取值，然后传送到 HMI 设备。

- 由操作员在配方视图中触发：
值被下载到 HMI 设备上。例如，您可以处理、编辑或保存这些值。
- 由函数或 PLC 作业触发：
值被立即保存到数据卷中。

写入值

触发写入作业后，值被写入 PLC 地址中。

- 由操作员在配方视图中触发：
当前值被写入 PLC。
- 由函数或 PLC 作业触发：
将当前值从数据介质写入 PLC。

参见

“数据信箱”区域指针 (页 1321)

8.9.3.3 同步传送

如果选择同步传送，两个通信伙伴均在公用数据区设置状态位。使用此机制可防止任意覆盖控制程序中的数据。

应用

同步数据记录传送是一种非常有用的解决方案，例如在以下情况下：

- 在传送数据记录时，PLC 是“主动方”。
- PLC 对有关配方号和数据记录的信息进行评估。

- 数据记录的传送由作业信箱触发。

要求

为了实现 HMI 设备和 PLC 之间数据记录的同步传送，组态时下列要求必须满足：

- 已设置区域指针：“连接”▶“区域指针”编辑器
- 已在配方中指定要与 HMI 设备同步传送数据记录的 PLC。“配方”编辑器，巡视窗口中配方的属性视图，“属性”▶“下载”组。

数据区的结构

数据区域的长度是固定的：为 5 个字。数据区域的结构：

	15	0
1. 字	当前配方号(1 - 999)	
2. 字	当前数据记录号 (0 - 65535)	
3. 字	保留	
4. 字	状态(0、2、4、12)	
5. 字	保留	

- 状态

状态字(字 4)可采用以下值：

数值		含义
十进制	二进制	
0	0000 0000	允许传送，数据记录空闲
2	0000 0010	传送忙碌
4	0000 0100	传送完成，没有错误
12	0000 1100	传送完成，出现错误

参见

“数据信箱”区域指针 (页 1321)

8.9.3.3 操作员在配方显示中启动的传送顺序

由操作员在配方显示中启动 PLC 读取操作

步骤	操作	
1	检查：Status word = 0?	
	是	否
2	HMI 设备在数据记录中输入要读取的配方号和“激活传送”状态，并将数据记录号设置为 0。	中止，出现系统报警。
3	HMI 设备从 PLC 读取值并将这些值显示在配方显示中。 如果配方有同步变量，也会将从 PLC 中读取的值写入这些变量中。	
4	HMI 设备设置“传送完成”状态。	
5	控制程序必须将状态字复位为 0 以允许进一步传送。	

向操作员在配方显示中启动的 PLC 写入

步骤	操作	
1	检查：Status word = 0?	
	是	否
	HMI 设备在数据记录中输入要写入的配方和数据记录号以及“激活传送”状态。	中止，出现系统报警。
2	HMI 设备将当前值写入 PLC。 如果配方有同步变量，则会在配方显示和变量之间同步更改后的值，然后将其写入 PLC。	
3	HMI 设备设置“传送完成”状态。	
4	如果需要，控制程序现在可判断传送的数据。	
5	控制程序必须将状态字复位为 0 以允许进一步传送。	

注意

状态字只能通过 HMI 设备进行设置。PLC 只能将状态字复位至零。

注意

如果满足下面列出的任一条件，则在检测到数据不一致情况时，PLC 只能评估配方和数据记录号。

- 数据信箱状态被设置为“传送完成”。
- 数据信箱状态被设置为“传送无差错完成”。

参见

“数据信箱”区域指针 (页 1321)

8.9.3.3 由作业信箱 SIMATIC S7 触发后的传送顺序

HMI 设备和 PLC 之间的数据记录传送可由任何站发起。

PLC 作业编号 69 和编号 70 可供此类传送使用。

编号 69：从 PLC 读取数据信箱(“PLC → DAT”)

作业信箱编号 69 将数据信箱从 PLC 传送到 HMI 设备。该作业信箱的结构如下：

	最高有效位	最低有效位
字 1	0	69
字 2	配方号 (1-999)	
字 3	数据信箱号 (1-65,535)	
字 4	不覆盖现有数据信箱：0 覆盖现有数据信箱：1	

编号 70：将数据信箱写入 PLC (“DAT → PLC”)

作业信箱编号 70 将数据信箱从 HMI 设备传送到 PLC。该作业信箱的结构如下：

	最高有效位	最低有效位
字 1	0	70
字 2	配方号 (1-999)	
字 3	数据信箱号 (1-65,535)	
字 4	—	

使用作业信箱“PLC → DAT” (编号 69) 从 PLC 读取时的顺序

步骤	操作	
1	检查：Status word = 0?	
	是	否
2	HMI 设备在数据信箱中输入于作业中指定的配方和数据信箱号以及“激活传送”状态。	中止，没有返回消息。
3	HMI 设备读取值，并将其存储在于作业信箱中指定的数据信箱中。	

步骤	操作
4	<ul style="list-style-type: none"> 如果在作业中选择了“覆盖”，将在无任何确认提示的情况下覆盖现有数据信箱。 HMI 设备设置“传送完成”状态。 如果在作业中选择了“不覆盖”，并且数据信箱已经存在，则 HMI 设备将中止该作业，并在数据信箱的状态字中输入 0000 1100。
5	要允许继续传送，PLC 程序必须将状态字再次设置为 0。

使用作业信箱“DAT → PLC” (编号 70) 向 PLC 写入时的顺序

步骤	操作	
1	检查：Status word = 0?	
	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">是</td> <td style="text-align: center;">否</td> </tr> </table>	是
是	否	
2	HMI 设备在数据信箱中输入于作业中指定的配方和数据信箱号以及“激活传送”状态。 中止，没有返回消息。	
3	HMI 设备从数据介质取出于作业中指定的数据信箱值，并将其写入 PLC。	
4	HMI 设备设置“传送完成”状态。	
5	PLC 程序现在可判断传送的数据。 要允许继续传送，PLC 程序必须将状态字再次设置为 0。	

参见

“数据信箱”区域指针 (页 1321)

8.9.3.3 由组态的函数触发后的传送顺序

使用组态的函数从 PLC 读取

步骤	操作	
1	检查：Status word = 0?	
	<table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">是</td> <td style="text-align: center;">否</td> </tr> </table>	是
是	否	
2	HMI 设备在数据记录中输入于函数中指定的配方和数据记录号以及“激活传送”状态。 中止，出现系统报警。	
3	HMI 设备从 PLC 读取值并将其存储在于函数中指定的数据记录中。	

注意

如果满足下面列出的任一条件，则在检测到数据不一致情况时，PLC 只能评估配方和数据记录号。

- 数据信箱状态被设置为“传送完成”。
- 数据信箱状态被设置为“传送无差错完成”。

对因出错而中止的传送的反应

如果数据记录的传送因出错而中止，则 HMI 设备会作出如下反应：

- 由操作员在配方显示中触发
在配方视图的状态栏中显示信息并输出系统报警
- 由函数触发
输出系统报警
- 由 PLC 作业触发
HMI 设备上无返回消息。

不过，可通过查询数据记录中的状态字来判断传送状态。

注意

可用于特定设备

配方视图的状态栏中的注释不可用于精简面板。

参见

“数据信箱”区域指针 (页 1321)

8.9.4 调试组件

8.9.4.1 调试组件

将 PLC 程序传送到 PLC

1. 使用合适的电缆连接 PC 和 CPU。
2. 将程序文件下载到 CPU。
3. 然后将 CPU 设置为“运行”。

将项目传送到 HMI 设备

1. HMI 设备必须处于传送模式才能接受项目传送。

可能的情景：

- 初始启动

HMI 设备尚未包含任何初始启动阶段的组态数据。必须将运行所需要的项目数据和运行系统软件从组态计算机传送到设备：HMI 设备自动更改为传送模式。

“传送模式”消息出现在 HMI 设备上：

– 重新调试

重新调试意味着重写 HMI 设备上的现有项目数据。

有关相应的详细说明，请参阅 HMI 设备手册。

2. 检查 WinCC 项目中的报警设置是否满足您的要求。
3. 在将项目传送到 HMI 设备之前检查下载设置。更多信息，请参考“[装载设置 \(页 1335\)](#)”。
4. 要将项目传送到 HMI 设备，在 HMI 设备的项目树中打开快捷菜单，然后选择命令“下载到设备”>“软件”。
 - 项目被自动编译。
 - 在“装载预览”窗口中，可以指定是否应覆盖现有用户数据。

5. 单击“下载”。将执行下载。

下载完成后，将在组态计算机上显示一条消息。

HMI 设备上将显示起始画面。

连接 PLC 和 HMI 设备

1. 使用合适的电缆连接 PLC 和 HMI 设备。
2. HMI 设备上将显示连接消息。请注意，用户可在 WinCC 中编辑系统事件文本。

注意事项

调试设备时，要始终遵守 HMI 设备手册中与安全相关的信息。

由设备（如手机）产生的 RF 辐射可能会导致意外的操作状态。

参见

[通信的基础知识 \(页 1300\)](#)

8.10 导入和导出项目数据

8.10.1 导入和导出项目数据

简介

WinCC 使您能够将数据从一个项目中导出，然后再将其导入至另一个项目。

可以导入或导出以下项目数据：

- 配方

导出和导入可减少工作量。例如，可使用在之前项目中创建的数据，而无需在配方中创建新数据记录。

导出项目数据

导出数据时，所有信息都存储在 CSV 文件中。

例如，可以设置用于导出的选项并确定列表分隔符。

编辑项目数据

可在 Excel 或文本编辑器中编辑导入文件。

启动 Excel 并从“文件”菜单中选择“打开”。从“文件类型”列表中选择“文本文件 (*.prn、*.txt 或 *.csv)”类型。请勿通过双击在 Excel 中打开导入文件，因为这样将会损坏文件的数据结构并阻止其导入。

在常规的文本编辑器中打开该文件，检查文件格式是否正确。

导入项目数据

导入项目数据时，这些数据会显示在 HMI 设备上。

例如，可以设置用于导入的选项并确定列表分隔符。

在导入 CSV 文件过程中将检查导入文件的语法。不会检查属性的含义或属性之间的依存关系。将在编译期间报告错误。

参见

导出配方 (基本版、高级版) (页 1330)

导入配方 (基本版、高级版) (页 1332)

8.10.2 导入和导出配方

8.10.2.1 导出配方 (基本版、高级版)

简介

WinCC 具有从配方导出数据记录的功能。

要求

- 要导出的 WinCC 项目已打开。
- 在项目中已创建配方。
- “配方”编辑器已打开。

导出配方

1. 在“配方”编辑器中，选择含有所要导出的数据记录的配方。

2. 单击



“导出”对话框随即打开。

The screenshot shows the 'Export' dialog box with the following settings:

- 配方选择 (Recipe Selection):** Orange
- 内容选择 (Content Selection):** All, Data record number. Range: 自 1 到 5.
- 文件选择 (File Selection):** C:\Users\Documents\Recipe_Orange.csv
- 数据分隔 (Data Delimitation):** 列表分隔符 (List Delimiter): ,; 十进制分隔符 (Decimal Delimiter): .
- Buttons:** 导出 (Export), 取消 (Cancel)

所选择的配方将显示在“配方选择”之下。

- 在“内容选择”下，指定是要导出所有数据记录还是仅导出所选择的数据记录。
- 在“文件选择”下，指定存储配方数据记录的文件。
- 在“数据分隔”下，指定列表分隔符和十进制分隔符。
- 单击“导出”。

导出将启动。

结果

导出的数据已写入到 CSV 文件中。CSV 文件存储在指定文件夹中。

参见

导入配方 (基本版、高级版) (页 1332)
配方数据的格式 (基本版、高级版) (页 1333)

8.10.2.2 导入配方 (基本版、高级版)

简介

配方以其名称来标识。这意味着配方名称必须是唯一的。指定导入期间现有数据记录是否被同名记录覆盖。

要求

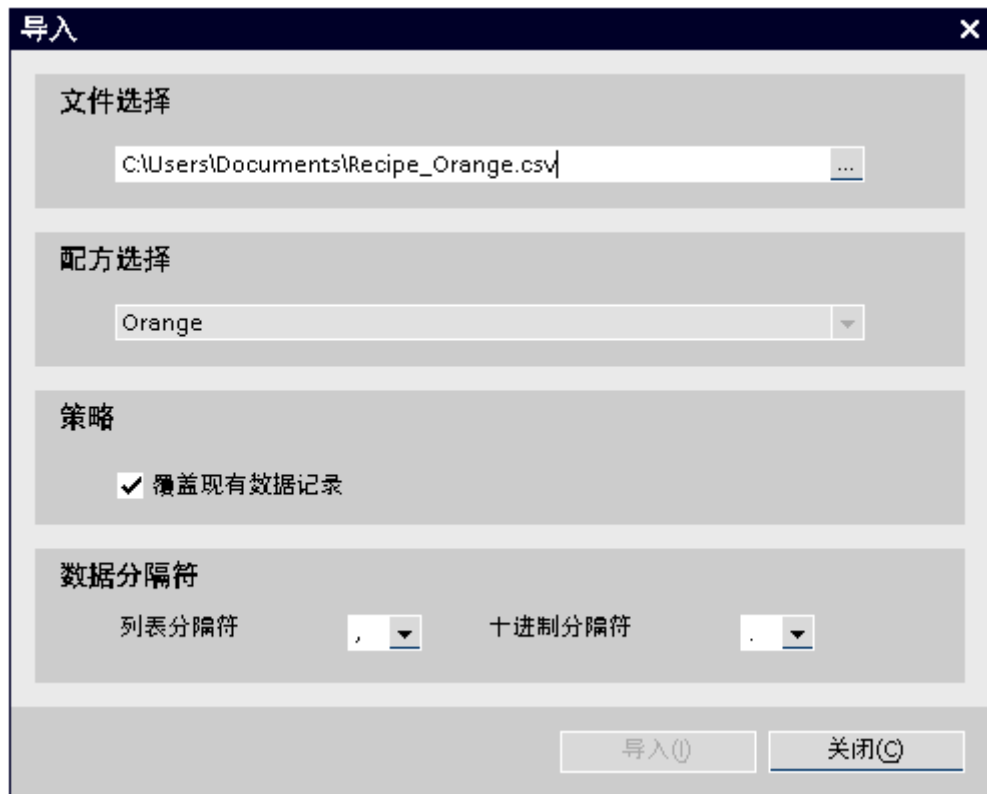
- 已创建包含至少一个配方的 CSV 文件。
- 要导入的 WinCC 项目已打开。
- “配方”编辑器已打开，其中包含至少一个配方。

导入配方

1. 在“配方”编辑器中，选择含有所要导入的数据记录的配方。
2. 单击



“导入”对话框随即打开。



所选择的配方将显示在“配方选择”之下。

3. 在“文件选择”下，选择要导入的文件。
4. 在“策略”下，指定现有数据记录是否被同名记录覆盖。

5. 在“数据分隔”下，选择要在 CSV 文件中使用的列表分隔符和十进制分隔符。
6. 单击“导入”。
导入将启动。

结果

在所选配方中创建数据记录。根据“策略”的设置，现有数据记录被 CSV 文件中的同名记录覆盖。

参见

导出配方 (基本版、高级版) (页 1330)
配方数据的格式 (基本版、高级版) (页 1333)

8.10.2.3 配方数据的格式 (基本版、高级版)

简介

本节介绍用于导入配方的文件的所需格式。包含配方数据的文件必须以“*.csv”格式提供：

配方数据的结构

导入文件的结构是固定的。下面的实例显示配方的结构，该配方包含两个配方元素，每个元素包含两个数据记录，每个数据记录包含两个值：

```
List separator=<List separator>
Decimal symbol=<Decimal separator><List separator>
<Name of the recipe><List separator><List separator><Line break>
LANGID_<ID of the language><List separator>
<Name data record 1><List separator>
<Name data record 2><List separator><Line break>
<Number recipe><List separator>
<Number data record 1><List separator>
<Number data record 2><List separator><Line break>
<Tag recipe element 1><List separator>
<Value data record 1><List separator>
<Value data record 2><List separator><Line break>
<Tag recipe element 2><List separator>
<Value data record 1><List separator>
<Value data record 2><List separator><Line break>
```

语言 ID

在十进制符号中使用“Windows 语言 ID”，例如，“1033”表示英语。更多信息，请参考 Windows 操作系统文档。

8.11 编译和装载

8.11.1 编译和装载项目

8.11.1.1 编译和加载项目概述

概述

正如在 WinCC 中组态项目一样，项目是在后台进行编译的。编译会生成可在相应 HMI 设备上运行的文件。

如果编译期间出现错误，则 WinCC 会提供支持来查找错误并进行更正。

更正所有问题之后，将编译后的项目装载到要运行该项目的 HMI 设备中。

启动项目的生产操作之前，使用 HMI 设备快捷菜单中的“编译”>“软件（完全编译）”命令对项目进行完全编译。如果在项目中使用连接到控制变量的 HMI 变量，则使用快捷菜单中的“编译”>“软件”命令对所有修改的块进行编译，然后再编译 HMI 设备。

为了减少当前组态下编译更新内容的时间，建议时常使用“编译”>“软件（完全编译）”命令。

8.11.1.2 编译项目

简介

必须先对项目进行编译，然后它才能够在 HMI 设备上运行。将编译过的项目随后装载到一个或多个 HMI 设备上。

正如在 WinCC 中组态项目一样，对项目所做的更改是在后台进行编译的。每次装载项目时均会自动编译项目。这可确保始终装载项目的最新版本。

注意

只要运行系统在工程师站上运行，后台中正在进行的编译就会中断。

必要时也可以手动启动项目的编译。

WinCC 会在编译期间检查项目的一致性。项目中的错误位置将在“巡视”窗口中列出。可从“巡视”窗口中的条目直接跳转到错误源。检查所发现的错误并对其进行更正。

编译的范围

初始编译涵盖整个项目。初始编译后，仅对项目所做的更改进行编译。

不过，可始终运行项目的完全编译。

要求

- 项目已经打开。

步骤

执行下列步骤，编译项目：

1. 如果要同时为多个 HMI 设备编译项目，可以在项目树中使用多选方式选择所有所需 HMI 设备。
2. 在 HMI 设备的快捷菜单中选择“编译”>“软件”命令。
3. 如果要编译整个项目：在快捷菜单中选择“编译”>“软件（编译全部）”命令。

结果

针对所有选定的 HMI 设备对项目进行了编译。编译期间出现的任何错误均显示在“巡视”窗口中。

8.11.1.3 装载项目

8.11.1.3 加载项目概述

概述

在将项目的增量数据装载到一个或几个 HMI 设备之前，会自动对这些数据进行编译。这可确保始终传送项目的最新版本。启动装载过程之前，可激活“覆盖全部”选项。

在每个 HMI 设备的属性中定义装载设置。

启动项目的生产操作之前，使用 HMI 设备快捷菜单中的“编译”>“软件（完全编译）”命令对项目进行完全编译。如果在项目中使用连接到控制变量的 HMI 变量，则使用快捷菜单中的“编译”>“软件”命令对所有修改的块进行编译，然后再编译 HMI 设备。

为了减少活动组态期间编译增量内容所需的时间量，建议时常使用“编译”>“软件（完全编译）”命令。

8.11.1.3 装载设置

简介

在每个 HMI 设备的属性中，进行装载设置。

要求

已在项目中创建了 HMI 设备。

步骤

执行下列步骤，以进行装载项目方面的设置：

1. 在 HMI 设备的快捷菜单中选择“属性”命令。
2. 在 HMI 设备的属性中单击“装载”。
将显示用于装载的设置。
3. 选择 HMI 设备上的同一个接口集。
4. 为连接设置参数。所示参数取决于选定的接口。
5. 检查“加载名称”复选框的设置。默认情况下会激活“加载名称”复选框。在这种情况下，所组态对象的名称与项目一起进行编译和装载。如果禁用该复选框，则在 HMI 设备上编译项目所需的空间会更少。
6. 指定是否应覆盖 HMI 设备上已有的用户管理数据和配方。

可在每次装载前在“装载预览”下更改这两个设置。

结果

用于装载的设置已指定。这些设置将在项目下次装载到对应的 HMI 设备时生效。

8.11.1.3 加载项目

简介

必须先将项目装载到 HMI 设备，然后它才可以在 HMI 设备上运行。

实际装载数据之前所执行的操作：

- 已编译了项目。编译期间出现的任何错误均显示在“巡视”窗口中。如果双击错误消息，WinCC 会在相应的编辑器中打开无效组态。

重新装载项目之前，清除所有显示的错误。

- 成功完成编译后会显示装载预览。预览会显示下列信息，这些信息以 HMI 设备分隔：
 - 装载的各个步骤
 - 装载时生效的预设置。可在启动下一个装载操作之前一次性编辑这些设置，也可以直接在预览中对其进行编辑。
 - 可能出现的任何警告。即使项目出现警告，也可以对项目进行编译和装载。功能可能会在运行时受到限制。
 - 可能出现的任何错误。如果项目出现错误，则无法编译和装载项目。

启动项目的生产操作之前，使用 HMI 设备快捷菜单中的“编译”>“软件（完全编译）”命令对项目进行完全编译。如果在项目中使用连接到控制变量的 HMI 变量，则使用快捷菜单中的“编译”>“软件”命令对所有修改的块进行编译，然后再编译 HMI 设备。

要求

- 已在项目中创建了 HMI 设备。
- HMI 设备连接到了工程师站。连接与“装载”组设置相对应。

步骤

执行下列步骤，装载项目：

1. 如果要将项目同时装载到多个 HMI 设备，可以使用多选方式在项目树中选择所有所需 HMI 设备。
2. 在 HMI 设备的快捷菜单中选择“装载到目标系统”>“软件”命令。

已编译了项目。

如果显示错误：双击列出的错误以进行更正。错误原因显示在组态中。清除所有错误后，重新启动装载。

成功完成编译后，“装载预览”对话框会打开。

3. 检查所显示的预设置并根据需要进行更改。
4. 单击“更新”按钮以查看这些更改将如何影响装载过程。预览即会更新。
如果预览指示可成功执行装载，则单击“装载”以将项目装载到所选 HMI 设备。

结果

项目装载到所有选定的 HMI 设备。在此过程中，将更新所有的现有项目。

如果在装载期间出现错误或警告，则相应的报警会显示在“巡视”窗口中的“装载”之下。

成功装载项目后，其即可在 HMI 设备上执行。

8.11.1.3 装载项目期间的错误消息

8.11.1.3 更新 HMI 设备的操作系统 (基本版)

简介

将 WinCC 项目传送到 HMI 设备时，WinCC 将检查 HMI 设备操作系统版本是否与项目中使用的操作系统版本匹配。如果操作系统版本不匹配，则会在开始装载前显示一条警告。

如果检测到错误的操作系统版本，将提供更新操作系统的机会。

注意事项

如果更新操作系统，则会丢失 HMI 设备上的所有配方和用户数据，并且无法恢复。

请事先执行备份以保存这些数据。

要求

HMI 设备已连接到组态计算机。

已正确组态组态计算机与 HMI 设备之间的接口。

在装载过程中更新操作系统

1. 如果 HMI 设备操作系统版本与组态不匹配，装载过程中将在“装载”对话框中显示一条消息。此对话框允许立即更新操作系统。
要开始更新，在“装载”对话框中单击“是”。
将开始操作系统更新。更新完成后，项目将被装载到 HMI 设备。
2. 更新期间将删除配方、用户数据和一些系统参数。取消更新可保留数据。在“装载”对话框中单击“否”以取消。将取消操作系统更新和装载过程。

取消更新后，可以备份指定数据，然后更新 HMI 设备的操作系统。

备份配方和用户数据

1. 在项目树中选择 WinCC 项目。
2. 选择菜单命令“在线”>“设备维护”>“备份”。“SIMATIC ProSave”对话框随即打开。
3. 从“数据类型”框中选择“配方”。
4. 在“另存为”框中，输入目标，或使用该对话框浏览到所需位置。
5. 单击“开始备份”按钮。将备份配方数据。
6. 重复上述步骤，备份用户数据。从“数据类型”框中选择“用户管理”。

在进行数据备份后更新操作系统

1. 在项目树中选择所需的 HMI 设备。
2. 选择菜单命令“在线”>“装载到设备”。
将打开“装载”对话框，并开始装载过程。
在“装载”对话框中，将询问您是否要更新操作系统。
3. 单击“是”以确认有关更新操作系统的提示。
操作系统将会更新。更新完成后，项目将被装载到 HMI 设备。

恢复备份的配方和用户数据

更新操作系统并将项目装载到 HMI 设备后，恢复所备份的数据。

1. 在项目树中选择 WinCC 项目。
2. 选择菜单命令“在线”>“设备维护”>“恢复”。“SIMATIC ProSave”对话框随即打开。
3. 单击“打开”框中的浏览按钮，然后在打开的对话框中浏览到所需数据的位置。
4. 选择含有备份的配方数据的文件，然后单击“打开”。
5. 在“SIMATIC ProSave”对话框中单击“开始恢复”。
恢复将会开始。
6. 重复上述步骤以恢复用户数据。

8.11.1.3 调整项目，使其适合于另一个 HMI 设备

简介

将 WinCC 项目传送到 HMI 设备时，WinCC 将检查 HMI 设备是否与项目中所使用的 HMI 设备的类型兼容。如果 HMI 设备的类型不匹配，则会在下载开始前显示一条消息。

下载将中止。

调整项目，使其适合于 HMI 设备

需要相应地调整项目，以便能够将项目下载到所连接的 HMI 设备。

- 将新 HMI 设备添加到项目树中。从 HMI 设备选择中选择正确类型的 HMI 设备。
- 将组态的组件从以前使用的 HMI 设备复制到新创建的 HMI 设备。
可以将许多组件直接复制到项目树中和详细视图中。
例如，可以使用快捷菜单将完整的“画面”文件夹复制到新 HMI 设备的画面文件夹中。
- 使用详细视图复制在项目树内不能使用快捷菜单中的“复制”命令的条目。
- 例如，在项目树中选择“配方”条目。配方显示在详细视图中。
- 在详细视图中选择配方，然后将其拖动到新 HMI 设备的“配方”条目。配方将会被复制。也可以在详细视图中选择多个对象。
- 组态无法复制的组件，如连接、区域指针和报警。
- 随时保存项目。
- 编译完整项目。
- 编译成功完成后，将项目下载到 HMI 设备。

链接引用

对链接对象的引用将被复制。复制链接对象后，会再次链接引用。

实例：

复制一个对象已链接到变量的画面。将画面添加到新 HMI 设备后，将在各个对象处输入变量名称。变量名称以红色标记，因为引用已打开。然后，将变量复制和插入到新 HMI 设备中时，打开的引用将会关闭。变量名称的红色标记会消失。

要完成对 PLC 中连接对象的引用，首先需要组态与 PLC 的连接。

使用信息区域

为 HMI 设备编译项目时，错误和警告将显示在巡视窗口的“信息”选项卡中。可以使用快捷菜单命令“转到”直接转到可以更正错误或警告的位置。

从上到下处理整个错误和警告列表。

编译成功完成后，将项目下载到 HMI 设备。

8.11.1.3 建立与 HMI 设备的连接

简介

要将 WinCC 项目装载到 HMI 设备，必须在组态计算机和 HMI 设备之间建立连接，还必须正确组态连接。如果无法建立连接，则将中止装载过程。

在组态计算机和 HMI 设备之间建立连接

1. 检查 HMI 设备和组态计算机之间的电缆连接情况。
2. 在 WinCC 中打开“设备和网络”编辑器并启动网络视图。
3. 在网络视图中选择子网，然后检查子网的设置。
4. 在网络视图或设备视图中选择 HMI 设备的接口，然后在巡视窗口中检查连接参数。
5. 打开 HMI 设备，在装载程序中按下“控制面板”按钮。“控制面板”随即打开。
6. 在“控制面板”中按下两次“传送”。“传送设置”对话框打开。
7. 检查设置，然后按“高级”。“PROFIBUS 设置”对话框将打开。
8. 检查高级设置，然后单击“确定”关闭对话框。

重要设置

检查连接设置时请仔细检查以下参数：

- 网络地址或站地址
- 所选择的传输率
- 总线上的主站，仅允许一个主站。

如果使用可组态适配器建立连接，还请检查适配器的设置，例如传输率、总线上的主站。

8.11.1.4 运行系统启动

8.11.1.4 在工程师站上启动运行系统

简介

当在 WinCC 中执行组态的同时，可以在工程师站上启动运行系统中的项目。然而，所谓的在线组态会受到某些限制。

当运行系统在工程师站上处于激活状态时，无法在后台编译项目。运行系统关闭后将项目装载到 HMI 设备时，会自动编译项目的增量数据。也可以手动启动编译。

在运行系统中启动项目时，在“组态”编辑器内为 HMI 设备而存储在项目中的设置生效。

要求

在工程师站上已打开项目。

步骤

执行下列步骤，在工程师站上启动运行系统：

1. 在项目树中选择所需的 HMI 设备。
2. 选择菜单命令“在线”>“启动运行系统”。
3. 如果希望在工程师站上启动运行系统之后编辑项目数据，请在 HMI 设备的快捷菜单中选择“编译”>“软件”命令。

更新的项目会显示在工程师站上的运行系统中。

8.11.1.4 在 HMI 设备上启动运行系统

简介

项目一装载到 HMI 设备，即可在运行系统中启动项目。

在运行系统中启动项目时，在“组态”编辑器内为 HMI 设备而存储在项目中的设置生效。

要求

WinCC Runtime 已安装在 HMI 设备上。

运行系统在精简面板上启动

在一段由组态指定的延迟之后，装载的项目会在精简面板上自动启动。

如果要在精简面板上手动启动项目，请在 HMI 设备的装载程序中按下“启动”按钮。

HMI 设备启动后，会显示装载程序。

8.11.2 仿真项目

8.11.2.1 仿真基础知识

简介

根据已定义的变量值，在工程师站上使用仿真器测试组态性能。这使您可以在进行生产操作之前快速找到任何逻辑上的组态错误。

可采用两种不同的方式启动仿真器：

- 菜单命令“在线”>“仿真运行系统”
- 菜单命令“开始”>“程序”>“西门子自动化”>“仿真”

要求

仿真/运行系统组件必须已安装在工程师站上，以供仿真之用。

应用领域

可使用仿真器来测试 HMI 系统的以下功能，例如：

- 检查限制级别和报警输出
- 中断的一致性
- 组态的中断仿真
- 组态的警告
- 组态的错误消息
- 检查状态显示
- 数字量和模拟量 I/O：
 - 默认设置
 - 读取
 - 修改

工作原理

可使用仿真器，以各种数值来仿真项目中的变量分配。

过程值和报警均保存到仿真日志中。退出仿真模式时，生成的数据会从该日志中删除。

8.11.2.2 仿真项目

简介

通过以下两种方法之一来仿真项目：

- 无连接的 PLC
在为仿真 WinCC Runtime 而读取的仿真表中，更改区域指针和变量的值。
- 有连接的 PLC，但无运行过程
通过直接在运行系统中运行项目来仿真项目。变量和区域指针变为激活状态。这样可以在运行系统中对组态的 HMI 设备创建一个可靠的仿真程序。

要求

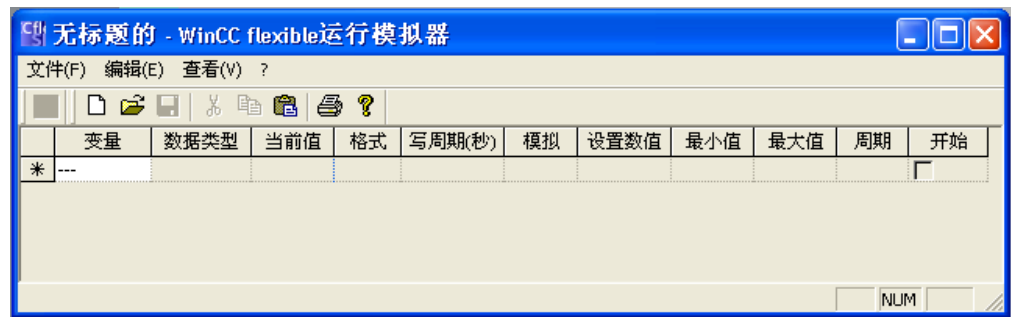
已创建了具有变量和区域指针的项目。

首次仿真项目

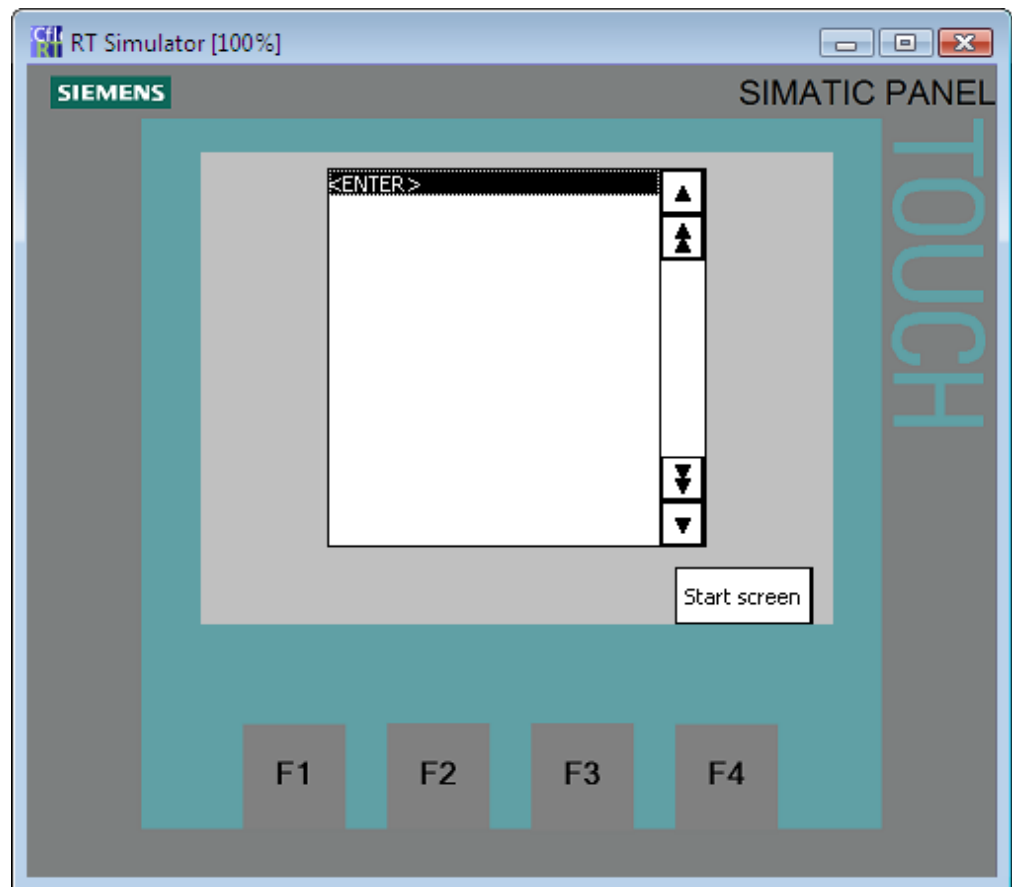
执行下列步骤，使用仿真表仿真项目：

1. 打开项目。
2. 选择菜单命令“在线”>“仿真运行系统”>“使用变量仿真器”。

首次仿真项目时，仿真器将启动一张新的空仿真表。



项目同时会在运行系统中打开：



可使用 <Alt + Tab> 键在仿真表和运行系统之间切换。

3. 在仿真表中单击“变量”列内的单元格，并从下拉列表中选择仿真表内已组态的变量。可选择要以此方式仿真的所有变量。

下拉列表中包含所有已组态的变量。在 WinCC 的“变量”编辑器中创建一个新变量。

利用该仿真器最多可仿真 300 个变量。

4. 在“仿真”列的下拉列表中，选择所需的仿真模式。



5. 在相应列中更改变量和区域指针的值。
6. 激活“启动”复选框以启动此变量的仿真。
7. 观察更改后的值如何影响项目。

通过此方法，可仿真每个单独的变量或同时仿真几个变量。

8. 使用菜单命令“文件”>“保存”来保存用于仿真的设置。输入名称以保存文件。会自动为该文件指定扩展名 *.six。

注意

无法仿真以下系统函数：

- CalibrateTouchScreen
- ActivateCleanScreen

管理仿真器数据

如果保存了之前仿真中的数据，则可在以后的某个时间点打开文件并再次仿真项目。这要求仿真表中的变量和区域指针仍在项目中。

执行下列步骤，打开仿真文件：

1. 选择菜单命令“在线”>“仿真运行系统”>“使用变量仿真器”。
2. 在仿真表中，选择菜单命令“文件”>“打开”。
3. 选择相应的仿真表并单击“打开”。

仿真器会装载已存储的数据。

启用和禁用变量

可启动和停止每个单独变量的仿真，使离线组态到在线组态的转换无波动。要完成此操作，请激活“启动”复选框。

启用变量后，会计算仿真的值并将其传递到 WinCC 仿真器。

删除变量

执行下列步骤，从仿真表中删除变量：

1. 选择变量的名称。

2. 选择菜单命令“编辑”>“剪切”。
删除的变量即会从表中移除。

8.11.2.3 使用变量仿真器

有关仿真表的注意事项

仿真表具有以下列：

列	描述
变量	指定要仿真的变量。
数据类型	显示所选变量的数据类型
当前值	显示已定义变量的仿真值
格式	指定所选格式，变量值将以此格式进行仿真： <ul style="list-style-type: none"> • 十进制 (1、2、3、4 ...) • 十六进制 (03CE、01F3 ...) • 二进制 (0 和 1)
写入周期	指定所选的时间间隔，变量的当前值将按此间隔进行仿真。如果输入值 2，变量的当前值将每隔 2 秒钟显示一次。
仿真	显示仿真期间处理变量值的方法。
设置值	为相应变量设置所选值。仿真从指定的值开始。
minWert 和 maxWert	指定变量的取值范围。为此范围设置最小值和最大值。默认最小值为 -32767，最大值为 32767。
期限	包括针对“增量”和“减量”仿真模式重复变量值的期限。
启动	根据先前输入的信息启动变量的仿真。

仿真模式

仿真器具有六个不同的仿真模式。在仿真期间，为已组态的变量提供接近实际的值。

仿真模式	描述
正弦	更改变量值以形成正弦曲线。参数被形象化为一个周期性的非线性函数。
随机	提供随机生成的值。用随机函数的方式改变变量值。
增量	变量值持续增至某个指定的最大值。达到最大值后再次从最小值开始。值趋势与正锯齿曲线相对应。
减量	变量值持续降至某个指定的最小值。达到最小值后再次从最大值开始。值曲线与负锯齿曲线相对应。

仿真模式	描述
移位	将一个置位按一位地连续移位。总是复位前一个位。例如，这使您可以测试 HMI 设备的报警。
<显示>	静态显示当前变量值。

实例：以“移位”仿真模式仿真变量

执行下列步骤，以“移位”仿真模式仿真变量：

1. 打开要仿真的项目。
2. 选择菜单命令“在线”>“仿真运行系统”>“使用变量仿真器”。
仿真表即会打开。
3. 在“变量”列中，选择来自项目的一个变量。
4. 在“格式”列中选择“二进制”。
5. 在“写入周期”列中输入值“1”。
6. 在“仿真”列中选择“移位”仿真模式。
7. 在“设置值”列中输入值“1”。
8. 通过“启动”复选框启用变量。

结果

仿真器逐位测试所选变量，如下所示：

仿真值	报警字节
设置起始值	00000001
1. 仿真值	00000010
2. 仿真值	00000100
3. 仿真值	00001000
....	...

在运行系统中，查看所需的报警是否以给定值输出。

8.11.3 维护 HMI 设备

8.11.3.1 ProSave

简介

ProSave 服务工具随 WinCC 一起提供。ProSave 功能集成到工程师站上的 WinCC 用户界面中。

ProSave 也可作为单独的程序（“独立模式”）安装在计算机上，且无需安装 WinCC。

功能范围

ProSave 提供了向 HMI 设备传送文件时所需要的全部功能。

- 数据备份和备份数据的恢复
- 操作系统更新
- 通信设置

工程师站上的集成操作

默认情况下，ProSave 由 WinCC 安装程序安装在工程师站上。在 WinCC 内，ProSave 功能的整个范围都集成到菜单“在线”>“设备维护”中。

在计算机上的独立操作

ProSave 也可通过 WinCC 光盘安装在计算机上且无需安装 WinCC，例如用于技术维护目的。在这种情况下，需要在“通信设置”中指定希望传送自/至哪个 HMI 设备以及用于文件传送的模式。

例如，使用 ProSave 可从原始 HMI 设备保存项目并在替换设备上对其进行恢复，而无需安装 WinCC。

如果是在 WinCC 外使用 ProSave，则可以更改用户界面的语言。要选择语言，请在 ProSave 中使用“语言”菜单命令。必须重新启动 ProSave，才能使语言切换生效。

8.11.3.2 备份 HMI 设备数据

简介

定期备份 HMI 设备的数据。

系统出现故障后或替换设备时，数据备份可使您快速恢复操作。系统只是将备份数据传送到新 HMI 设备中，从而恢复原始状态。

使用 WinCC 或 ProSave 进行数据备份

从连接了 HMI 设备的工程师站工作时，可使用 WinCC 备份和恢复此 HMI 设备的数据。

可选择在计算机上使用 ProSave 且无需安装 WinCC，藉此方便地执行中央数据备份。

数据备份的范围

备份和恢复操作取决于 HMI 设备的类型，可以包含下列项目数据：

- 完全备份（取决于 HMI 设备：运行系统、固件、操作系统映像、组态、配方、密码）
- 仅配方
- 仅密码

备份 HMI 数据时将产生扩展名为 *.psb 的备份文件。

只要 HMI 设备和存储介质之间存在适当的连接，就可以备份到任何存储介质，例如数据服务器。

注意

仅在用相同组态软件组态的操作员设备上对项目数据使用恢复功能。

注意

执行完整数据备份和恢复操作时，请注意以下几点：

- 执行完整数据恢复时，设备和操作系统上原有的所有数据都将被删除，并且不能撤消。
 - 如果数据的恢复中断，则在再次触发恢复之前，操作系统首先需要通过“复位为出厂设置”重新装载到 HMI 设备上。
-

8.11.3.3 备份和恢复 HMI 设备上的数据

简介

定期备份 HMI 设备的数据。

从连接了 HMI 设备的工程师站工作时，可使用 WinCC 备份和恢复此 HMI 设备的数据。

可选择在计算机上使用 ProSave 且无需安装 WinCC，藉此方便地执行中央数据备份。

要求

- HMI 设备与工程师站互连或者计算机与 ProSave 连接。
- 已在项目树中选择了应备份或恢复其数据的 HMI 设备。
- 已在 HMI 设备的属性中正确地设置了用于装载的设置。
- 使用特殊的存储介质（例如数据服务器）时：HMI 设备连接了此存储介质。

步骤

请按如下步骤备份数据：

1. 从“在线”>“设备维护”菜单中选择“备份”命令。
2. 选择备份的范围：“完整备份”、“配方”或“用户管理”。
3. 单击“...”，在“选择备份文件”对话框中选择存储位置，并指定文件名。
4. 单击“确定”。

这将启动数据备份。备份可能要花费一些时间。

数据恢复过程

1. 从“在线”>“设备维护”菜单中选择“恢复”命令。
2. 单击“...”，并在“打开”对话框中选择存储位置和文件。
在“内容”区域中指出哪个 HMI 设备是数据备份的源设备以及数据备份的范围。
3. 单击“确定”。

这将启动恢复过程。该过程可能要花费一些时间。

8.11.3.4 更新操作系统

简介

如果 HMI 设备的操作系统具有与组态不兼容的版本，则会中止组态的装载。将出现一条报警，指示操作系统必须更新。

更新操作系统

与工程师站互连的 HMI 设备的操作系统可以使用该工程师站上的 WinCC 进行更新。

可选择在 PC 上使用 ProSave 且无需安装 WinCC，藉此更新 HMI 设备的操作系统。

更新操作系统时，目标系统上的所有数据都将被删除。

如果要在更新操作系统后使用存储于内部闪存上的用户数据（例如密码和配方），请在启动更新之前，将这些数据导出到外部数据介质。执行更新后，将数据导回到 HMI 设备。

“复位为出厂设置”

如果操作系统更新被过早地终止，则 HMI 设备上将不再有任何操作系统。此时，装载操作系统的唯一方法是“复位为出厂设置”。

工程师站和 HMI 设备之间用于操作系统更新的通信受 HMI 设备的操作系统控制。使用“复位为出厂设置”，工程站将改为与 HMI 设备的引导装载程序通信。

该过程可能要花费一些时间。

在 WinCC 中执行完“复位为出厂设置”后，关闭 HMI 设备，然后再次启动。利用此方法，可通过引导装载程序与 HMI 设备建立联系。

注意

在通过以太网复位出厂设置之前，需要将 HMI 设备的 MAC 地址和组态 PC 上的 PG/PC 接口设置为以太网 TCP/IP。

通过“开始”>“系统控制”>“设置 PG/PC 接口”更改 PG/PC 接口的设置。在“应用程序的存取点”域中选择“S7ONLINE (STEP7) -> TCP/IP”。

8.11.4 参考

8.11.4.1 装载项目期间的错误消息

装载期间的可能问题

向 HMI 设备装载项目时，输出窗口中将显示关于装载过程进度的状态消息。

在向 HMI 设备装载项目时出现的任何问题的原因通常是传输率不正确，或工程师站上同时使用了其它驱动程序（例如，如果连接到了网络）。

下面列出了最常见的装载失败案例。

串行传送终止。

可能的纠正方法：选择更低的传输率。

传送终止，并发出兼容性冲突报警。

可能的原因	解决方法
组态软件的版本和 HMI 设备操作系统之间发生冲突。	使 HMI 设备的操作系统与组态软件的版本同步。 使用“在线”菜单中的“设备维护”>“更新操作系统”命令来更新操作系统。 有关附加信息，请参考 HMI 设备的操作说明。 另外，还可以使用“ProSave”服务工具来更新操作系统。有关附加信息，请参考 ProSave 的在线帮助。
组态计算机连接了非法设备(例如 PLC)。	检查并更正电缆连接和通信参数。

传送操作未进行

可能的原因	解决方法
不能建立与 HMI 设备的连接(输出窗口中的报警)	检查组态计算机与 HMI 设备之间的物理连接。HMI 设备还必须处于传送模式(远程控制情况除外)。
默认通信驱动程序未在 Windows 设备管理器中列出。	在设备管理器的属性窗口中检查 COM 连接的设备状态。

MPI/DP 下载不执行。

可能的原因	解决方法
用于下载的 CP 被设为“组态模式”(例如，在使用 SIMATIC NET CD 7/2001 时)。	转到“设置 PC 站”工具并将 CP 设为“PG 模式”。 检查网络参数 (传输率、MPI 地址)。 从 WinCC ES 中执行下载。 将 CP 重新设为“组态模式”。
在编程设备/PC 面板中，S7ONLINE 存取点不在物理设备 (例如 CP5611 (MPI)) 上。例如安装了 SIMATIC NET CD 7/2001 时可能会发生这种问题。	转到“PG/PC 面板”工具或者若正在使用 SIMATIC NET CD 7/2001，则转到“设置 PC 站”工具，然后将 S7ONLINE 存取点置于所需要的设备上。 检查网络参数 (传输率、MPI 地址)。 从 WinCC ES 中执行下载。 将 S7ONLINE 存取点重新定位回原来的设备。

组态过于复杂

可能的原因	解决方法
组态中所包括的选定设备的不同对象和选项过多。	清除某个类型的所有对象 (例如所有图形指示器)。

8.12 运行系统中的操作**8.12.1 基础知识****8.12.1.1 概述****组态和过程控制阶段**

HMI 设备用于在过程及生产自动化中操作和监视任务。通过 HMI 设备上的设备画面可以清楚直观地总览当前过程。包含有设备画面的 HMI 项目是在组态阶段创建的。

过程控制阶段，将项目传送至 HMI 设备。对于过程控制阶段的另一个要求是 HMI 设备必须在线连接到 PLC。然后可在工作过程进行期间执行过程控制阶段、操作员控制和监视。

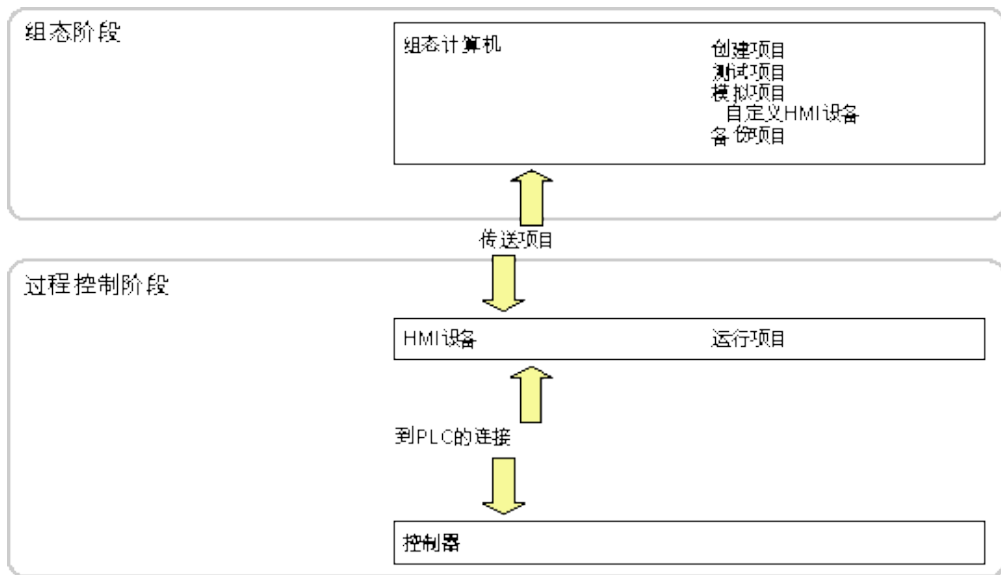


图 8-1 组态和过程控制阶段

将项目下载到 HMI 设备

可通过以下步骤将项目下载到 HMI 设备：

- 从组态 PC 下载
- 使用 ProSave 从 PC 上恢复项目

在这种情况下，已归档的项目从 PC 下载到 HMI 设备。不必在 PC 上安装组态软件。

这些步骤可用于项目的初次调试和再次调试。

初次调试和再次调试

- 当第一次调试 HMI 设备时，HMI 设备上不存在任何项目。
当操作系统更新完毕之后，HMI 设备也处于这种状态。
- 当再次调试时，HMI 设备上的所有项目都将被取代。

参见

- 运行系统软件中的设置 (页 1354)*
- 操作员控制项目的概述 (页 1362)*
- 棒图 (页 1371)*
- 概述 (页 1386)*
- 概述 (页 1394)*
- 配方的结构 (页 1403)*
- 运行系统中的变量 (页 1353)*
- 运行系统中的系统函数 (页 1353)*

8.12.1.2 运行系统中的变量

定义

变量对应于 HMI 设备上的指定存储区域，将向其中写入值和/或从中读取值。该操作可由 PLC 启动或由操作员通过 HMI 设备启动。

参见

概述 (页 1351)

8.12.1.3 运行系统中的系统函数

应用

在运行系统中，系统函数用于下列目的：

- 控制过程。
- 利用 HMI 设备的属性。
- 在线模式下，组态 HMI 设备上的系统设置。

WinCC 中的每个系统函数均与一个对象和一个事件存在逻辑链接。一旦事件产生，系统函数即被触发。

系统函数

这些默认系统函数用于在运行系统中执行多种任务，例如：

- 计算，如以特定值或可变值增加变量值。
- 记录功能，如启动过程值日志。
- 设置，如 PLC 改变或置位 PLC 中的位。
- 报警，如一个不同用户登录后。

事件

对象和所选函数决定可触发执行系统函数的事件。

例如，“更改值”、“超出下限”和“超出上限”事件与“变量”对象相关联。而“装载”、“清除”事件则与“画面”对象相关。

参见

概述 (页 1351)

8.12.2 调试项目

8.12.2.1 运行系统软件中的设置

打开 WinCC 组态软件，并对运行系统软件进行以下设置：

在 PLC 上显示

在 WinCC 中，组态运行系统中所生成项目的可视化表示。对于精简面板，画面分辨率是固定的。如果图片大于组态的画面分辨率，就会出现滚动条。

要关闭任务栏，选择“开始”菜单中的“设置”>“任务栏”命令。在“任务栏属性”对话框中，禁用“总在最前”和“自动隐藏”属性。

对话框字体

对话框文本以标准字体显示。通过在“设备设置”编辑器中选择“语言和字体”来定义默认字体。

禁用程序切换

锁定程序切换，以防止操作员在运行系统中调用其它应用程序。

在“设备设置”编辑器中单击“窗口”，然后选择选项“禁用程序切换”。另请隐藏 Windows 任务栏。

注意

停止运行系统

如果锁定了程序切换，则一定要组态软键或按钮以调用“StopRuntime”系统函数。否则，将无法关闭运行系统或 Windows。

屏幕保护程序

屏幕保护程序对于大多数新式的屏幕已不再需要，事实上，它甚至会造成损坏。一旦视频信号在指定时间内没有改变，这些监视器将切换到休眠模式。常规的屏保将会阻止它，从而缩短监视器的使用寿命。

注意

经批准的屏幕保护程序

如果您确实希望使用屏幕保护程序，请注意，只允许在运行系统中使用标准的 Windows 屏幕保护程序。

确保在安装了运行系统软件的 PC 上设置正确的时区。要在 Windows 中设置时区，请选择“开始”>“设置”>“控制面板”>“日期和时间”。

参见

概述 (页 1351)

加载项目 (页 1355)

关闭项目 (页 1357)

8.12.2.2 加载项目

概述

装载项目可能有多种情况：

- 运行系统软件与组态软件安装在同一系统上。
- 运行系统软件与组态软件安装在不同的系统上。必须将项目从组态计算机装载到目标系统。为了进行传送，必须将 HMI 设备连接到组态 PC。另一项要求是 HMI 设备上和 WinCC 中的传送模式必须匹配。

注意

根据组态的具体情况，装载过程中可能会出现安全提示。HMI 设备上的配方数据和密码列表在提示后被覆盖。

组态软件和运行系统软件安装在同一系统上

如果组态软件和运行系统软件安装在同一系统上，请执行以下步骤：

1. 创建并编译项目。
2. 直接从激活的组态软件中启动运行系统。从“在线”菜单中选择“启动运行系统”命令。
3. 如果已经组态了相应的通信，就可以使用控制器在线测试和操作项目。

组态软件与运行系统软件安装在不同的系统上

如果组态软件和运行系统软件安装在不同的系统上，请执行以下步骤：

1. 创建并编译项目。更多信息，请参考“[编译项目 \(页 1334\)](#)”。
2. 通过电缆下载文件：

使用标准电缆将 HMI 设备连接到组态计算机以匹配所需的传送模式，然后启动 HMI 设备。
3. 将 HMI 设备设置为传送模式。

要启动传送模式，请按下装载程序中的“传送”按钮。也可以将系统函数“SetDeviceMode”分配给操作员控件。
4. 将项目从组态计算机装载到目标设备。有关详细信息，请参考“[装载项目 \(页 1336\)](#)”。

注意

如果 HMI 设备是 PC，不必使用装载程序就可以传送已编译文件，例如通过以太网。双击 PC 上的相应文件以启动运行系统。

参见

[运行系统软件中的设置 \(页 1354\)](#)

在 HMI 设备的当前显示上更换操作模式。(页 1238)

8.12.2.3 在工程师站上启动运行系统

简介

当在 WinCC 中执行组态的同时，可以在工程师站上启动运行系统中的项目。然而，所谓的在线组态会受到某些限制。

当运行系统在工程师站上处于激活状态时，无法在后台编译项目。运行系统关闭后将项目装载到 HMI 设备时，会自动编译项目的增量数据。也可以手动启动编译。

在运行系统中启动项目时，在“组态”编辑器内为 HMI 设备而存储在项目中的设置生效。

要求

在工程师站上已打开项目。

步骤

执行下列步骤，在工程师站上启动运行系统：

1. 在项目树中选择所需的 HMI 设备。
2. 选择菜单命令“在线”>“启动运行系统”。
3. 如果希望在工程师站上启动运行系统之后编辑项目数据，请在 HMI 设备的快捷菜单中选择“编译”>“软件”命令。

更新的项目会显示在工程师站上的运行系统中。

8.12.2.4 在 HMI 设备上启动运行系统

简介

项目一装载到 HMI 设备，即可在运行系统中启动项目。

在运行系统中启动项目时，在“组态”编辑器内为 HMI 设备而存储在项目中的设置生效。

要求

WinCC Runtime 已安装在 HMI 设备上。

运行系统在精简面板上启动

在一段由组态指定的延迟之后，装载的项目会在精简面板上自动启动。

如果要在精简面板上手动启动项目，请在 HMI 设备的装载程序中按下“启动”按钮。

HMI 设备启动后，会显示装载程序。

8.12.2.5 测试项目

简介

可使用以下选项测试 WinCC 项目：

在组态计算机上测试项目。

- 仿真器

仿真器用来测试带内部变量和过程变量的 WinCC 项目。更多信息，请参考“[仿真项目 \(页 1342\)](#)”。

仿真器使您可以测试以下内容：

- 在未连接到 PLC 时，对组态的离线测试。
- 在连接到 PLC 且过程处于非激活状态时，对组态的在线测试。
- 演示项目的实施。

在 HMI 设备上测试项目

- 在 HMI 设备上对项目的离线测试

离线测试意味着，HMI 设备与 PLC 之间的通信在测试持续期间处于断开状态。您可以操作 HMI 设备，但无法与 PLC 交换数据，反之亦然。通过将系统函数“[SetDeviceMode \(页 1276\)](#)”分配给操作员控件，在 HMI 设备上设置“离线”模式。

- 在 HMI 设备上对项目的在线测试

在线测试意味着，HMI 设备与 PLC 之间的通信在测试持续期间处于保持状态。使用基于组态的 HMI 设备来控制设备。通过将系统函数“[SetDeviceMode](#)”分配给操作员控件，在 HMI 设备上设置“在线”模式。

步骤

执行下列步骤，在 PLC 没有连接到组态 PC 的情况下仿真项目：

1. 创建一个项目，稍后它将和一个互连 PLC 一起运行。
2. 保存并编译项目。
3. 直接从活动组态软件中启动仿真器。选择菜单命令“在线”>“仿真运行系统”>“使用变量仿真器”。
首次仿真项目时，仿真器将启动一张新的空仿真表。将打开已为项目创建的仿真表。
仿真表“*.six”包含仿真所需的所有设置。更多信息，请参考“[使用变量仿真器 \(页 1345\)](#)”。
4. 在仿真表中对项目的变量和区域指针进行任何更改。

使用 <ALT+TAB> 快捷键在仿真表和运行系统之间切换。

使用菜单命令“文件”>“保存”来保存用于仿真的设置。输入保存文件的名称。会自动为文件名分配扩展名“*.six”。

8.12.2.6 关闭项目

简介

定义在用户程序中关闭运行系统的步骤：

步骤

按以下步骤退出运行系统：

1. 当运行系统运行时，可以使用关闭符号或任务管理器将其关闭。
2. 当运行系统运行时，按下对应的按钮来关闭运行系统。运行系统的关闭需要专门组态。

参见

运行系统软件中的设置 (页 1354)

8.12.2.7 备份和恢复 HMI 设备上的数据

简介

定期备份 HMI 设备的数据。

从连接了 HMI 设备的工程师站工作时，可使用 WinCC 备份和恢复此 HMI 设备的数据。

可选择在计算机上使用 ProSave 且无需安装 WinCC，藉此方便地执行中央数据备份。

要求

- HMI 设备与工程师站互连或者计算机与 ProSave 连接。
- 已在项目树中选择了应备份或恢复其数据的 HMI 设备。
- 已在 HMI 设备的属性中正确地设置了用于装载的设置。
- 使用特殊的存储介质（例如数据服务器）时：HMI 设备连接了此存储介质。

步骤

请按如下步骤备份数据：

1. 从“在线”>“设备维护”菜单中选择“备份”命令。
2. 选择备份的范围：“完整备份”、“配方”或“用户管理”。
3. 单击“...”，在“选择备份文件”对话框中选择存储位置，并指定文件名。
4. 单击“确定”。

这将启动数据备份。备份可能要花费一些时间。

数据恢复过程

1. 从“在线”>“设备维护”菜单中选择“恢复”命令。
2. 单击“...”，并在“打开”对话框中选择存储位置和文件。
在“内容”区域中指出哪个 HMI 设备是数据备份的源设备以及数据备份的范围。
3. 单击“确定”。

这将启动恢复过程。该过程可能要花费一些时间。

8.12.3 运行系统中的语言

8.12.3.1 运行系统中的语言

使用多种运行系统语言

用户可以决定在特定 HMI 设备的运行系统中使用哪些项目语言。HMI 设备上可同时使用的运行系统语言的数目取决于设备。为了使操作员可以在运行时切换语言，必须组态相应的操作员控件。

运行系统启动时，项目根据最近的语言设置进行显示。首次启动运行系统时，显示语言为“语言顺序”中编号最小的语言。

在组态期间设置运行系统语言

可在“语言和字体”编辑器中指定以下内容：

- 可用作 HMI 设备运行系统语言的项目语言。
- 语言切换的次序。

参见

设置运行语言 (页 1359)

设置运行语言的字体 (页 1360)

组态语言切换 (页 1361)

运行系统中亚洲和东方语言的特点 (页 1361)

8.12.3.2 设置运行语言

简介

“语言和字体”编辑器显示项目中可用的所有项目语言。可以选择用作 HMI 设备运行系统语言的项目语言。此外，指定语言切换的顺序。

要求

在“项目语言”编辑器中启用了多种语言。

步骤

1. 打开“设备设置”下的“语言和字体”编辑器。
2. 在“运行系统语言”列中，选择运行系统首次启动时要使用的语言。
在“语言顺序”列中，为选中的语言分配编号“0”。
3. 在“运行系统语言”列中，选择在切换语言时下一个将激活的语言。
在“语言顺序”列中，为选中的语言分配编号“1”。

语言与字体						
运行语言与字体选择						
运行系统语言	语言顺序	语言名	固定字体 0	缺省字体	组态的字体 0	
<input checked="" type="checkbox"/>	0	English (USA)	Tahoma	Tahoma, 11px	<None>	

- 按照切换语言时语言被激活的先后顺序选择其它语言。
如果所选语言的编号超出了可装载到 HMI 设备上的编号，则表格背景将改变颜色。
- 如果要更改语言的顺序，选择所需行然后选择快捷菜单命令“向上”或“向下”。

语言与字体						
运行语言与字体选择						
运行系统语言	语言顺序	语言名	固定字体 0	缺省字体	组态的字体 0	
<input checked="" type="checkbox"/>	0	English (USA)	Tahoma	Tahoma, 11px	<None>	

结果

允许的运行系统语言和经编译的项目一起传送到 HMI 设备。

首次启动运行系统时，项目的显示语言为“语言顺序”中编号最小的语言。

如果已经组态了通过“SetLanguage”系统函数进行语言切换，则指定的编号序列将确定语言切换的次序。

8.12.3.3 设置运行语言的字体

简介

可在“语言和字体”编辑器中指定用于在 HMI 设备上显示各种运行系统语言文本的字体。如果无法定义特定字体，则将在所有文本中使用默认字体。

WinCC 只提供 HMI 设备支持的字体。

在 HMI 设备的操作系统中默认字体还用于对话框文本显示。如果不能显示完整的对话框文本或标题，则选择较小的字体作为默认设置。

要求

在“项目语言”编辑器中启用了多种语言。

步骤

- 打开“设备设置”下的“语言和字体”编辑器。
- 在“运行系统语言”列启用要在 HMI 设备上显示的语言。

在“固定字体 0”列中，WinCC 显示运行系统中默认使用的字体。

3. 组态期间，在“已组态字体 0”列中，为每种希望可用的语言选择一种其它字体。

在传送操作过程中，这些字体会传送到 HMI 设备上。

4. 在“标准字体”列中，选择不能为文本选择字体时默认使用的字体。

结果

选定语言的项目文本以选定字体在 HMI 设备上显示。

8.12.3.4 组态语言切换

简介

如果希望在 HMI 设备上使用多种运行系统语言，则需要组态语言切换。操作员要在各运行系统语言间切换就必须组态语言切换。

语言切换的方法

可以按下列方法组态语言切换：

- 直接语言选择

每种语言均通过单独的按钮进行设置。在这种情况下，需要为每种运行系统语言创建一个按钮。为每种字体组态“SetLanguage”系统函数，使用语言编号或语言 ID 作为参数。

- 语言切换

操作员可使用按钮在所有语言间切换。

使用“Toggle”参数为该按钮组态“SetLanguage”系统函数。语言将按照“语言和字体”编辑器中指定的顺序启用。

不管使用哪种方法，必须将按钮名称翻译为所使用的那种语言。也可组态输出域以显示当前的语言设置。

8.12.3.5 运行系统中亚洲和东方语言的特点

简介

组态亚洲语言时，在运行系统中操作时应注意一些特点。

注意

组态期间只能使用组态计算机支持的亚洲字体。

亚洲字符集的内存要求

使用亚洲语言时所需的内存空间当然更大。因此，在编译项目时要注意任何错误消息。

输入东方和亚洲字符 (非 ANSI)

无法通过精简面板向运行系统中输入东方和亚洲字符。

亚洲字符的翻译

使用 Sm@rtAcess 和 Sm@rtService 时，只能使用 HMI 设备能够识别的那些字符。为了能够使用亚洲字符，必须在工程系统中对这些字符进行组态。另外组态的字符在 HMI 设备上需要额外的内存空间。注意 HMI 设备上可用内存空间的大小。

亚洲字符集的字号

在运行系统中，使用至少 10 磅字号来显示为亚洲语言创建的项目的文本。如果使用更小的字号，亚洲字符将变得字迹模糊。本规则同样适用于“语言和字体”编辑器中使用的默认字体。

亚洲语言的文本域长度

处理包含亚洲语言的多语言项目时，请为文本域的长度留出余量。域内容可能会隐藏一部分，具体取决于字体和字号。

1. 在巡视窗口中打开“属性”>“外观”文本框。
2. 在“适合大小”下禁用“自动调整大小”选项。
3. 验证运行系统中的显示是否正常。

8.12.4 操作项目

8.12.4.1 基础知识

8.12.4.1 操作员控制项目的概述

所有基本 HMI 设备都具有触摸屏。某些基本 HMI 设备具有功能键。使用触摸屏和功能键操作控制面板或 HMI 设备上运行的项目。



危险 错误操作

项目中包含的某些操作可能要求操作员对特定设备有深入的了解。

请确保只允许经过培训的专业人员操作设备。

操作触摸屏

小心

对触摸屏的损坏

尖头或锋利的物体可能会损坏触摸屏的塑料表面。

请始终只用手指或触摸笔操作触摸屏。

触发意外动作

同时触摸几个操作员控件可能会触发意外的动作。

请每次只触摸屏幕上的一个操作员控件。

操作员控件是 HMI 设备屏幕上的触摸敏感式符号。

触摸的操作与机械按键的操作基本相同。通过用手指触摸操作员控件来将其激活。

注意

HMI 设备只要检测到操作员控件被触摸就会返回可见反馈。

可见反馈是独立的，与 PLC 的通信无关。因此，可见反馈信号并不指示相关动作是否已真正执行。

操作员控件的实例：

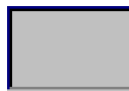
- 按钮

按钮可呈现下列状态：

“未触摸”



“触摸”



- 隐藏按钮

隐形按钮的特点是默认情况下在选择后并不显示。在这种情况下不会提供任何视觉操作反馈。

然而，组态工程师可组态隐形按钮，以便在触摸后将按钮的轮廓显示为线条。该轮廓将保持可见，直到选择其它的操作员控件。

- I/O 域

触摸了 I/O 域后（例如，为了输入密码），会显示屏幕键盘作为可见反馈。

根据 HMI 设备和所组态的操作员控件，系统会针对是输入数字值还是字母数字值而显示不同的屏幕键盘。

输入完成后，屏幕键盘将自动再次隐藏。

操作功能键

可以为功能键分配全局或局部功能：

- 采用全局功能分配的功能键

采用全部功能分配方式的功能键在 HMI 设备或 PLC 上始终触发相同的动作，无论当前显示的画面如何。例如，激活画面或关闭报警窗口就属于这类动作。

- 采用局部功能分配的功能键

采用局部功能分配方式的功能键与画面相关，因此，仅在活动画面中才生效。

分配给功能键的功能可根据画面的不同而变化。

在一个画面中，功能键只能分配有一种功能，即全局功能或局部功能。局部功能分配优先于全局功能分配。

参见

概述 (页 1351)

屏幕键盘的一般功能 (页 1364)

在 KTP400 Basic 上输入数据 (页 1365)







在 KTP600、KTP1000、TP1500 Basic 上输入数据 (页 1367)

显示信息文本 (页 1369)

设置项目语言 (页 1371)

8.12.4.1 屏幕键盘的一般功能

以下按键在所有基本 HMI 设备的屏幕键盘上都可用：

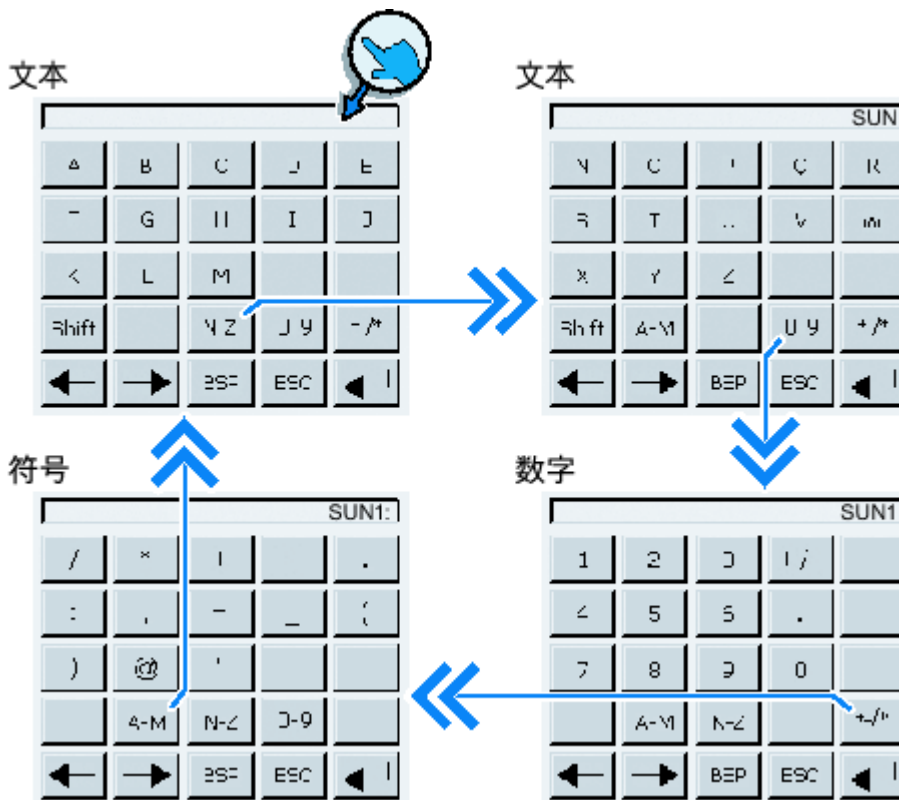
	光标向左
	光标向右
	删除字符
	取消输入
	确认输入
	显示信息文本。该按键只在为操作员控件配置了信息文本时才显示。

参见

操作员控制项目的概述 (页 1362)




8.12.4.1 在 KTP400 Basic 上输入数据



由于 KTP400 Basic 的显示屏很小，因而其屏幕键盘和输入概念不同于其它基本 HMI 设备。



触摸需要输入的操作员控件时，屏幕键盘会出现在 HMI 设备触摸屏上。

KTP400 的屏幕键盘有四个视图。使用屏幕键盘第四行中的按钮进行输入时，可以更改视图：

按钮	视图的更改
	输入文本，字符“A”到“M”
	输入文本，字符“N”到“Z”
	输入数字“0”到“9”(有符号/无符号以及含/不含小数位)

按钮	视图的更改
	输入特殊字符
	输入文本 (切换到小写字母)

注意

作业邮箱无效

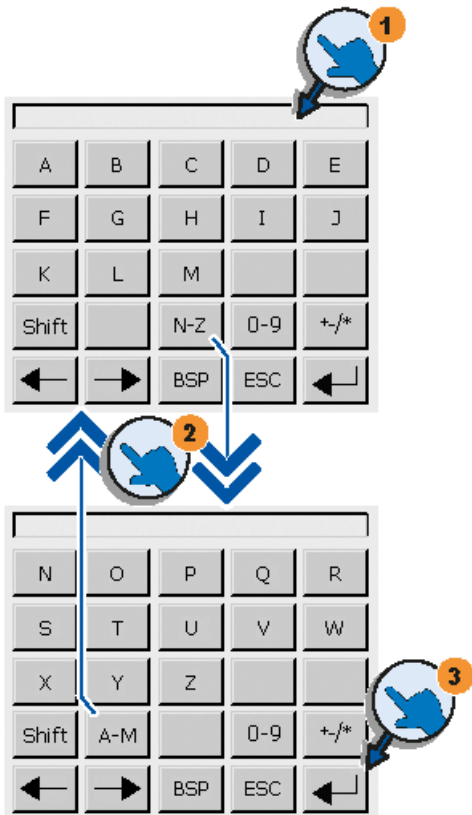
屏幕键盘打开时，PLC 作业 51“选择画面”无效。

键布局

字母数字键盘布局为单一语言。

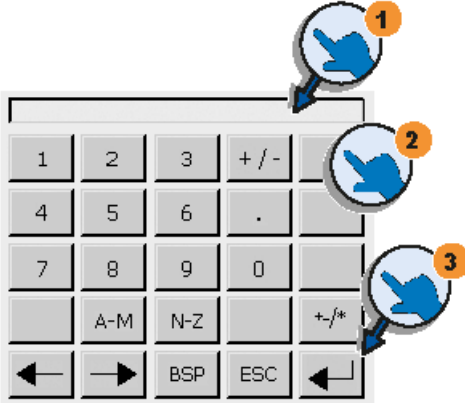
在项目内切换语言不会对字母数字屏幕键盘造成任何影响。

输入字母数字值



1. 在屏幕上触摸所需的操作员控件。
字母数字屏幕键盘将打开。
2. 输入数值。视设置而定，HMI 设备会输出有声信号。
可以使用键 <N-Z> 和 <A-M> 更改屏幕键盘的视图。
使用 <Shift> 键可以输入小写字母。
3. 按 <Return> 键确认输入，或使用 <ESC> 取消输入。
其中任一项操作均会关闭屏幕键盘。

输入数字值



1. 在屏幕上触摸所需的操作员控件。
数字屏幕键盘将打开。
2. 输入数值。视设置而定，HMI 设备会输出有声信号。
可以使用 <N-Z> 和 <A-M> 键更改屏幕键盘的视图，以输入十六进制表示的数字。
3. 按 <Return> 键确认输入，或使用 <ESC> 取消输入。
其中任一项操作均会关闭屏幕键盘。

检查数字限制值

可为变量分配限制值。输入任何超出此限制值范围的值都会受到拒绝。如果组态了报警视图，将触发系统事件，并且会再次显示原始值。

数字值的小数位

组态工程师可定义数字文本框小数位的位数。当在此类 I/O 域中输入值时，系统会检查小数位的位数。

- 超出限制值的小数位将被忽略。
- 未使用的小数位将用“0”条目进行填充。

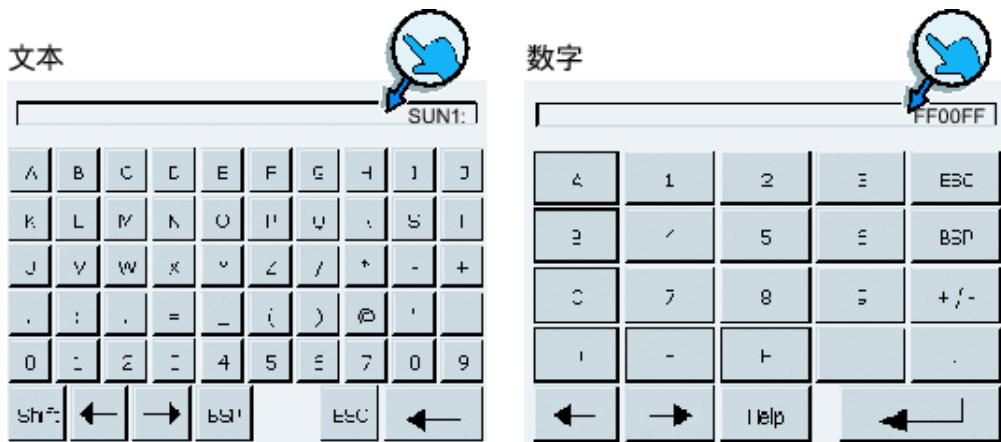
参见

操作员控制项目的概述 (页 1362)

8.12.4.1 在 KTP600、KTP1000、TP1500 Basic 上输入数据

字母数字屏幕键盘

触摸需要输入的操作员控件时，屏幕键盘会出现在 HMI 设备触摸屏上。



注意

作业邮箱无效

屏幕键盘打开时，PLC 作业 51“选择画面”无效。

键布局

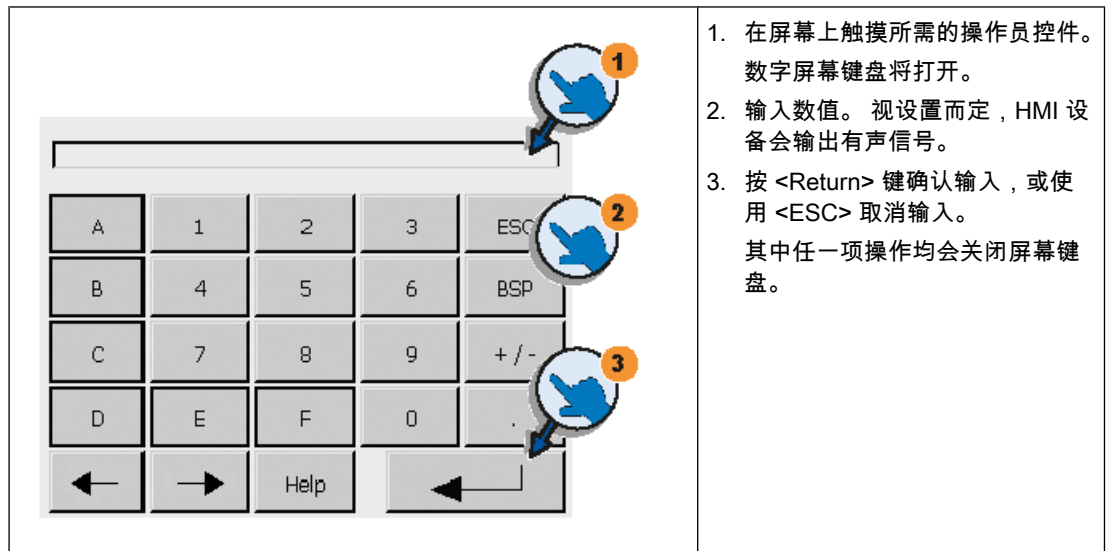
字母数字键盘布局为单一语言。

在项目内切换语言不会对字母数字屏幕键盘造成任何影响。

输入字母数字值

1. 在屏幕上触摸所需的操作员控件。字母数字屏幕键盘将打开。
2. 输入数值。视设置而定，HMI 设备会输出有声信号。
使用 <Shift> 键可以输入小写字母。
3. 按 <Return> 键确认输入，或使用 <ESC> 取消输入。
其中任一项操作均会关闭屏幕键盘。

输入数字值



1. 在屏幕上触摸所需的操作员控件。数字屏幕键盘将打开。
2. 输入数值。视设置而定，HMI 设备会输出有声信号。
3. 按 <Return> 键确认输入，或使用 <ESC> 取消输入。其中任一项操作均会关闭屏幕键盘。

检查数字限制值

可为变量分配限制值。输入任何超出此限制值范围的值都会受到拒绝。如果组态了报警视图，将触发系统事件，并且会再次显示原始值。

数字值的小数位

组态工程师可定义数字文本框小数位的位数。在此类型 I/O 域中输入值时，系统会检查小数位的位数。

- 超出限制值的小数位将被忽略。
- 未使用的小数位将用“0”条目进行填充。

参见

操作员控制项目的概述 (页 1362)

8.12.4.1 显示信息文本

应用

组态工程师使用信息文本来提供附加信息和操作说明。组态工程师可以组态画面和操作员控件上的信息文本。

I/O 域的信息文本可以包含要输入值的信息等内容。



图 8-2 I/O 域的信息文本实例

步骤

执行下列步骤，打开操作员控件的信息文本：

1. 触摸所需的操作员控件。

屏幕键盘将打开。根据



键的外观，您可以辨别出信息文本是组态用于操作员控件还是组态用于当前画面。

2. 触摸屏幕键盘上的



键。

将显示操作员控件的信息文本。如果所选画面对象没有信息文本，则显示当前画面的信息文本（如果已组态）。

可使用



和



滚动查看较长信息文本的内容。

注意

在显示的信息文本间切换

组态工程师可组态 I/O 域及相关画面的信息文本。通过触摸信息文本窗口可在这两种信息文本之间切换。

3. 按下



关闭所显示的信息文本。

其它步骤

根据组态的不同，也可通过所组态的操作员控件调用信息文本。

参见

操作员控制项目的概述 (页 1362)

8.12.4.1 设置项目语言

简介

HMI 设备支持多语言项目。如果要在运行期间更改 HMI 设备上的语言设置，则必须已组态了相应的操作元素。

项目启动时使用的语言总是上一个会话中设置的语言。

要求

- HMI 设备上必须提供项目所需的语言。
- 语言切换功能必须逻辑上链接到已组态的操作元素，如按钮。

选择语言

可以随时更改项目语言。切换语言之后，语言相关对象将立即以新语言显示在画面中。

可使用以下方法切换语言：

- 使用所组态的操作元素将语言从列表中的一种切换到另一种。
- 使用所组态的操作元素直接设置想要的语言。

参见

操作员控制项目的概述 (页 1362)

8.12.4.2 操作对象

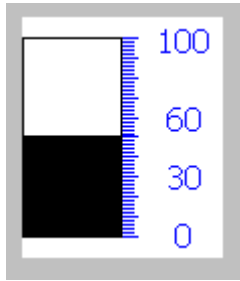
8.12.4.2 棒图

应用

棒图是一种动态显示对象。棒图以矩形区域显示来自 PLC 的值。棒图使您一眼即可识别出以下内容：

- 当前值与组态的限制值之间的差距
- 是否达到设置点值

棒图可以显示诸如填充量或批计数等值。



布局

棒图布局取决于组态：

- 棒图可以提供一系列值
- 可以用线指示组态的限制值
- 超出限制值或未达限制值时，可通过颜色变化进行报告

参见

概述 (页 1351)

概述 (页 1377)

概述 (页 1382)

概述 (页 1383)

概述 (页 1384)

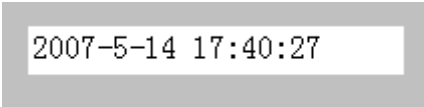
8.12.4.2 日期/时间域

8.12.4.2 概述

应用

“日期/事件框”可能具有下列运行系统功能：

- 输出日期和时间
- 组合的输入和输出；操作员可以在此处编辑输出值，以重置日期和时间。



2007-5-14 17:40:27

布局

日期/时间域的外观取决于在 HMI 设备中设置的语言。

日期可以详细形式显示(例如，2003 年 12 月 31 日，星期二)，也可以简短形式显示(31.12.2003)。

操作

根据组态的不同，可以使用下列方法操作日期/时间域：

- 标准操作：修改日期和时间。

运行系统特性

输入值时如果操作员没有注意语法或输入了无效的值，系统将拒绝这些值并将原始值（加上期间消耗的时间）显示在日期/时间域中，同时 HMI 设备上将显示一条系统报警消息。

8.12.4.2 触摸和键操作

触摸操作

执行下列步骤，操作日期/时间域：

1. 在 HMI 设备的触摸屏上，触摸日期/时间域。将自动显示屏幕键盘。
2. 使用屏幕键盘输入所需值。
3. 按 <ENTER> 确认输入的内容，或按 <ESC> 取消。在确认或取消所输入内容后，屏幕键盘会自动消失。

键操作

例如，根据组态的 Tab 顺序，通过一次或多次按下



按键激活日期/时间域。将有带颜色的边框标识出域内容的选中状态。

步骤

执行下列步骤，操作日期/时间域：

1. 使用光标键定位光标并输入值。
2. 按下



。对象变为特殊的编辑模式。在该域中，任何时候都只有一个字符被标记。

- 使用光标键



/



来滚动字符表。

- 使用光标键



/



可定位到下一个或前一个光标位置。

3. 用



确认输入内容或用



放弃输入。

8.12.4.2 I/O 域

8.12.4.2 概述

应用

在 I/O 域中输入数字或字母数字值。例如，数字值可以是基准温度 80 (数值)，字母数字值可以是用户名“Service”(文本)。



布局

I/O 域的外观取决于组态：

- 数字 I/O 域
用于输入十进制、十六进制或二进制格式的数字。
- 字母 I/O 域
用于输入字符串。
- 用于日期和时间输入的 I/O 域
用于输入日历日期或时间信息。格式取决于设置的组态。
- 用于密码输入的 I/O 域
用于密码的隐藏输入。输入的字符串以占位符(*)显示。

操作

根据组态的不同，可以使用下列方法操作 I/O 域：

- 标准操作：在 I/O 域中输入值。
- 事件：操作 I/O 域 (例如，激活 I/O 域) 时将触发事件。您可将函数列表的处理组态到事件中。

运行系统特性

数字值的限制值测试

可为变量分配限制值。如果输入值超出此限制值，将不会接受该值，例如在限制值为 78 时输入 80。在这种情况下，如果组态了报警窗口，HMI 设备将发出系统报警。并将再次显示原始值。

数字值的小数位

组态工程师可定义数字文本框中小数位的位数。在此类 I/O 域中输入值时，系统会检查小数位的位数。

- 将忽略超出限制的小数位。
- 空的小数位将用“0”填充。

隐藏输入

在隐藏输入过程中，系统使用“*”显示每个字符。输入值的数据格式不能识别。

输入域之间切换时的特性

在同一画面中切换到其它输入域并且显示画面键盘时，在关闭屏幕键盘之前，不会为前一个域执行“退出域”事件。

8.12.4.2 触摸和键操作

触摸操作

执行下列步骤，操作 IO 域：

1. 在 HMI 设备的触摸屏上触摸 IO 域。将自动显示屏幕键盘。
2. 使用屏幕键盘输入所需值。
3. 按 <ENTER> 确认输入的内容，或按 <ESC> 取消。

在确认或取消所输入内容后，屏幕键盘会自动消失。

键操作

例如，根据组态的 Tab 顺序，通过一次或多次按下



按键激活 IO 域。将有带颜色的边框标识出域内容的选中状态。

步骤

执行下列步骤，操作 IO 域：

1. 使用



和光标键定位光标。

2. 此步骤会取消域内容选择。输入所需值。

3. 按下



。对象变为特殊的编辑模式。在该域中，任何时候都只有一个字符被标记。

- 使用光标键



/



来滚动字符表。

- 使用光标键



/



将光标移动到下一个或前一个输入位置。

- 4. 用



确认输入内容或用



放弃输入。

注意

使用按键



将输入键切换到字符模式，这样可以输入十六进制数值的字符“A”到“F”。

8.12.4.2 图形视图

应用

图形视图显示图形情况。



布局

图形的外观取决于组态。例如，图形视图可自动根据图形的大小进行调整。

注意

如果在 WinCC 屏幕中使用带有“透明色”设置的位图，则需要面板 HMI 设备具有高性能显示器。通过禁用相应图形对象属性中的“透明色”设置来增强性能。此限制尤其适用于将位图用作背景图像的情况。

操作

图形显示只用于显示，不能进行操作。

参见

棒图 (页 1371)

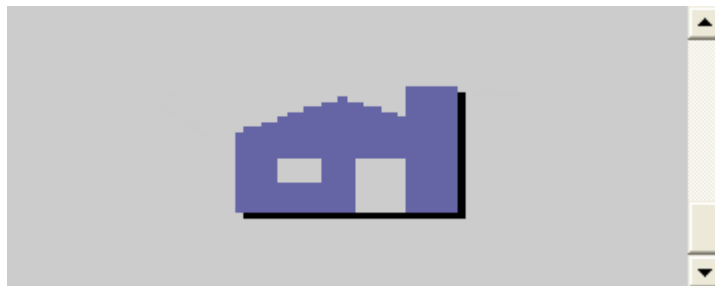
8.12.4.2 图形 I/O 域**8.12.4.2 概述****应用**

图形 IO 域可以有如下列运行系统功能：

- 图形列表条目的输出
- 组合输入和输出

用作输出域的实例：

为指示阀的运行时状态，图形 IO 域输出关闭的或打开的阀的图像。

**操作**

根据组态的不同，可以使用下列方法操作图形 IO 域：

- 标准操作：从图形列表中选择一条目。
- 事件：操作图形 IO 域(如，激活图形 IO 域)时将触发事件。您可将函数列表的操作组态到事件中。

运行系统特性

如果图形 IO 域显示仙人掌图像，表明还未在项目中为特定值定义图形输出。
 图形 IO 域的内容通过改变颜色来显示其当前激活。
 输出域以平面方式显示 3D 边框。

参见

棒图 (页 1371)
 触摸和键操作 (页 1378)









8.12.4.2 触摸和键操作

触摸操作

在 HMI 设备的触摸屏上触摸图形 IO 域。现已启用选择模式。
 使用滚动条选择图形对象。
 触摸所选图形对象以保存它，或者触摸其它画面对象来放弃当前选择。

键操作

如何在键盘设备上使用图形 IO 域：

步骤		步骤	
1	选择图形 IO 域	例如 	图形 IO 域被标记。
2	启用选择模式		现已启用选择模式。
3	选择一个条目	   	逐行移动光标。
4	接受选择		所选择条目现已生效。选择模式关闭。
	或 取消选择		将恢复原始值。

参见

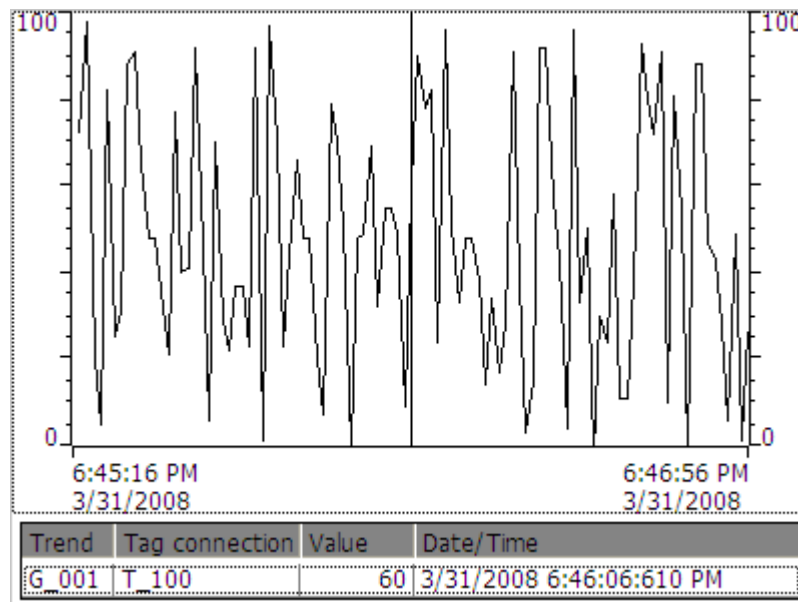
概述 (页 1377)

8.12.4.2 趋势视图

8.12.4.2 概述

应用

趋势视图是一种动态显示对象。如果 HMI 设备支持，趋势视图可以持续显示实际的过程数据和记录中的过程数据。



布局

趋势视图的布局基于组态。例如，趋势视图可以同时显示多个曲线，以允许用户比较不同的过程序列。如果显示的过程值超出或低于组态的限制值，可以通过改变曲线中的颜色来显示限制值超界。

标尺还可以简化从趋势视图进行过程值的读取。标尺显示对应 X 值的 Y 值。










操作

根据组态的不同，您能够：

- 扩展显示的时间区域。
- 缩小显示的时间区域。
- 回滚一个显示宽度。
- 前滚一个显示宽度。
- 停止或继续趋势记录。

操作员控件

按钮具有下列功能：

操作员控件	功能
	回退到趋势记录的开始位置。趋势记录的起始值将显示在此处。
	缩放所显示的时间区域
	缩小所显示的时间区域
	向后移动标尺 (向左)。
	向前移动标尺(向右)。
	回滚一个显示宽度 (向左)。
	前滚一个显示宽度 (向右)。
	显示或隐藏标尺。标尺显示相应位置的 X,Y 坐标值。
	停止或继续趋势记录

参见

棒图 (页 1371)

触摸和键操作 (页 1380)

8.12.4.2 触摸和键操作

步骤

在 HMI 设备的触摸屏上，触摸趋势视图的相关操作员控件。



步骤

通过已组态的 Tab 顺序，使用



键选择趋势视图。

下表列出了可用的快捷键：

键	功能
 + 	回退到趋势记录的开始位置。趋势记录的起始值将显示在此处。

键	功能
 + 	增大显示的时间段。
 + 	减少显示的时间段。
 +  + 	向后移动标尺 (向左)。
 +  + 	向前移动标尺(向右)。
 + 	回滚一个显示宽度 (向左)。
 + 	前滚一个显示宽度 (向右)。

参见

概述 (页 1379)

8.12.4.2 按钮

8.12.4.2 概述

应用

按钮是 HMI 设备屏幕上的虚拟键，具有一项或多项功能。



布局

按钮的布局取决于按钮类型。

- 带文本按钮：按钮上显示的文本提供有关按钮状态的信息。
- 带图形按钮：按钮上显示的图形提供有关按钮状态的信息。
- 不可见：该按钮在运行期间不可见。

操作

根据组态的不同，可以使用下列方法操作按钮：

- 标准操作：单击按钮。
- 事件：操作按钮(例如，单击按钮)时将触发事件。您可将函数列表的操作组态到事件中。

运行系统特性

操作之后可能会有可视化反馈。然而，还要注意，可视化反馈信息只指示操作已完成，而不指示是否真正执行了组态的功能。

参见

棒图 (页 1371)

触摸和键操作 (页 1382)

8.12.4.2 触摸和键操作

步骤

在 HMI 设备的触摸屏上，触摸按钮。

步骤

在键盘设备上操作按钮的步骤如下：

1. 使用光标键选择按钮，例如



2. 然后，按下



或



键。

参见

[概述 \(页 1382\)](#)

8.12.4.2 开关

8.12.4.2 概述

应用

开关是一种操作元素，以两种状态显示对象：“已按下”和“已释放”。开关可以用信号告知无法从 HMI 设备看到的系统组件(例如电机)的状态。也可以在 HMI 设备处改变系统组件的状态。



布局

开关的布局取决于开关类型。

- 开关：开关具有滚动条。该滚动条的位置提供了有关开关状态的信息。
- 带文本的开关：开关上显示的文本提供了有关开关状态的信息。
- 带图形的开关：开关上显示的图形提供了有关开关状态的信息。

操作

根据组态的不同，可以使用下列方法操作开关：

- 标准操作：单击开关。
- 事件：操作开关(例如，单击开关)时将触发事件。您可将函数列表的操作组态到事件中。

运行系统特性

一个开关具有两种稳定状态：激活开关时，它切换至另一种状态。开关保持该状态，直至下一次操作。

参见

[棒图 \(页 1371\)](#)

[触摸和键操作 \(页 1384\)](#)

8.12.4.2 触摸和键操作

触摸操作

因其类型的不同，开关的触摸操作也不相同：

- 如果开关显示有滚动条：
在 HMI 设备的触摸屏上将滚动条拖动到新位置，或在滚动条范围内双击。
- 如果开关只显示有文本或图形对象：
在 HMI 设备的触摸屏上触摸开关。

键操作

在键盘设备上操作开关的步骤如下：

- 使用光标键选择开关，例如



- 然后，按下



或



键。

参见

概述 (页 1383)

8.12.4.2 符号 I/O 域

8.12.4.2 概述

应用

可将运行系统中的下列功能分配给符号 IO 域：

- 文本列表条目的输出
- 组合输入和输出

用作组合 IO 域的实例：

可以从文本列表中选择文本“电机关”或“电机开”来控制运行系统中的电机。将根据所做选择启动或关闭电机。符号 IO 域直观显示电机的当前状态。

操作模式：“在线”

操作

根据组态的不同，提供了以下操作符号 IO 域的选项：

- 标准操作：从文本列表中选择一条目。
- 事件：通过操作符号 IO 域（例如通过启用它）触发事件。您可将函数列表的操作组态到事件中。

运行系统特性

如果对应的条目没有在项目数据中定义，符号 IO 域的选项列表将显示空的文本行。通过更改符号 IO 域内容的颜色，在 HMI 设备上指示激活状态。

参见

棒图 (页 1371)

触摸和键操作 (页 1385)

8.12.4.2 触摸和键操作

触摸操作







在 HMI 设备的触摸屏上，触摸符号 IO 域。选项列表会显示默认条目。



如果选择列表显示滚动条：在 HMI 设备的触摸屏上触摸滚动条。在触摸屏上触摸滚动条并将其拖动到所需位置。

通过在屏幕上触摸条目来选择条目并接受相应的变量值。将关闭选择列表并显示该条目。焦点仍落在符号 IO 域上。

键操作

在键盘设备上操作符号 IO 域的步骤如下：

步骤		步骤	
1	选择符号 IO 域	例如 	符号 IO 域被标记。
2	打开选择列表		下拉列表框打开。
3	选择一个条目	   	逐行移动光标。

步骤		步骤	
4	接受选择 或		所选择条目现已生效。选择列表关闭。
	取消选择		将恢复原始值。选择列表关闭。

参见

概述 (页 1384)

8.12.4.3 项目安全

8.12.4.3 概述

安全系统的设计

组态工程师可通过实施某种安全系统来保护项目操作。

安全系统基于权限、用户组和用户。

如果使用带有访问保护的操作员控件，HMI 设备会首先要求您登录。显示登录画面，在其中输入用户名和密码。登录后，可以操作那些您拥有所需权限的操作员控件。

组态工程师可通过单独的操作员控件设置登录对话框。

组态工程师也可以相同方式设置一个进行注销的操作员控件。在注销之后，将不能再操作分配有访问保护的對象；如要操作，请重新登录。

用户组和权限

用户组由组态工程师针对具体项目创建。默认情况下，所有项目中都包含“管理员”组和“用户”组。系统会为用户组分配权限。逐一为项目中的各个对象和功能定义某项操作所需的权限。

用户和密码

只能将各个用户分配给单个用户组。

以下人员可创建用户并为用户分配密码：

- 组态工程师在组态期间
- HMI 设备管理员
- 具有 HMI 设备的用户管理权限的用户

无论在哪个用户组中，每个用户都可更改自己的密码。

注销时间

在系统中为每个用户指定注销时间。如果任意两个用户操作（如输入值或切换画面）之间的时间间隔超过此注销时间，则用户将被自动注销。然后，用户必须再次登录才能继续操作分配有访问保护的對象。

备份与恢复

用户数据将被加密并保存在 HMI 设备中，以防因电源故障而丢失。

可备份和恢复在 HMI 设备上设置的用户、密码、组分配和注销时间。这样就不必在另一台 HMI 设备上重新输入所有数据。

注意事项

在下列情况下，当前有效的用户数据将被覆盖：

- 根据新的项目下载的设置。
 - 恢复备份的项目时
 - 通过操作员控件导入用户管理时。新下载或恢复的用户数据和密码立即生效。
-

参见

概述 (页 1351)

简单用户视图 (页 1387)

用户登录 (页 1388)

用户注销 (页 1390)

创建用户 (页 1390)

更改用户 (页 1392)

删除用户 (页 1393)

8.12.4.3 简单用户视图

应用

在带有小型显示屏的 HMI 设备上，简单用户视图用来在 HMI 设备上显示用户。

Administrator	Administrator group
Foreman	Users
Miller	Prgrammer
PLC User	未授权
Smith	Users
<新建用户>	

注意

无法使用脚本动态操作“简单用户视图”对象。

布局

外观取决于权限。

- HMI 设备系统上的所有用户显示在管理员或具有管理员权限用户的“用户”视图中。
- 对于不具有管理员权限的用户，仅显示他/她自己的用户条目。

操作

根据组态的不同，您能够：

- 管理用户，例如，创建用户、删除用户。
- 更改现有的用户数据。
- 导出或导入用户数据。

注意

一台 HMI 设备最多支持 100 名用户。这种限制条件不适用于 PC。在 PC 中，用户的最大数量受到 PC 的物理内存的限制。

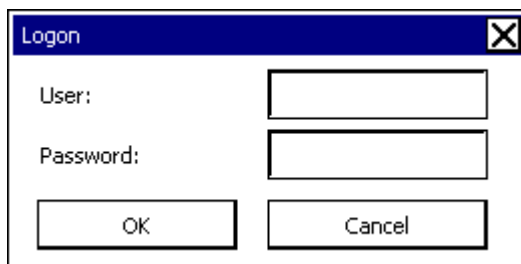
参见

概述 (页 1386)

8.12.4.3 用户登录

登录对话框

使用登录对话框登录到 HMI 设备的安全系统。在登录对话框中输入用户名和密码。



下列情况下登录对话框会打开：

- 使用具有访问保护的操作员控件。
- 按下已组态用于显示登录对话框的操作员控件。
- 在简单用户视图中选择条目“<ENTER>”时。
- 在扩展用户视图中选择空白条目时。
- 根据组态，启动项目时会自动显示登录对话框。

要求

登录对话框打开。




触摸操作的步骤

请按如下步骤进行操作：

1. 输入用户名和密码。
触摸相应的文本框。将显示字母数字屏幕键盘。
2. 选择“确定”确认登录。

键操作的步骤

请按如下步骤进行操作：

1. 通过按

键在登录对话框内选择“用户”输入域。
2. 使用系统键输入用户名。
使用

键将数字小键盘切换到字母模式以输入字母。
3. 使用

键选择“密码”输入域。
4. 使用系统键输入密码。
5. 按“确定”确认输入。

注意

用户名不区分大小写。

密码区分大小写。

结果

成功登录到安全系统后，可在 HMI 设备上执行具有访问保护和经授权的功能。

如果输入的密码错误且已组态了报警视图，则会输出报警。

参见

概述 (页 1386)

8.12.4.3 用户注销

要求

您已登录到 HMI 设备的安全系统。

步骤

您可以采取下列方式注销：

- 按下为注销而组态的操作员控件。
- 如果您不再操作项目并且超出注销时间，那么您将被自动注销。

如果您输入了错误的密码，您也将被自动注销。

结果

您不再登录到项目。要使用具有访问保护的操作员控件，首先必须重新登录。

参见

概述 (页 1351)

概述 (页 1386)

8.12.4.3 创建用户

要求

- 用户视图已打开。
- 您拥有用户管理权限或是管理员。
- 用户组已创建。

注意事项

运行系统用户必须分配给用户组。用户组是在工程系统中创建的。用户组的名称与语言相关。

注意事项



不能在密码中使用以下字符：

- 空格
- 特殊字符 * ? . % / \ ' "

在简单用户视图中创建用户

请按如下步骤进行操作：

1. 单击用户视图中的“<新建用户>”条目。一个对话框随即打开。

2. 输入所需用户名和密码。
触摸相应的文本框。将显示字母数字屏幕键盘。
3. 单击组的文本框。一个对话框随即打开。
4. 将用户分配到组。选择

和

以滚动选择列表。
5. 触摸下拉列表框中的所需条目。
所选条目即作为输入被接受。
6. 触摸文本框“注销时间”(Logoff time)。将显示屏幕键盘。
7. 输入注销时间 (0 至 60 分钟)。值 0 表示“不自动注销”。
8. 按“确定”确认输入。

结果

创建了新用户。

参见

[概述 \(页 1386\)](#)

8.12.4.3 更改用户

要求

用户视图已打开。

您的权限级别决定可以更改哪些数据：

- 您是管理员或具有用户管理权限的用户。在这些情况下，您可更改用户视图中 HMI 设备上所有用户的数据：
 - 用户名
 - 组分配
 - 密码
 - 注销时间
- 您不具有用户管理权限。这种情况下，您只能更改自己的用户数据：
 - 密码
 - 注销时间 (如果已组态)

注意

您只能为“Admin”用户更改注销时间和密码。

您只能为“PLC_User”更改注销时间。该用户用于通过 PLC 进行登录。

注意事项

在用户视图中进行的更改将在运行系统中立即生效。在运行系统中进行的更改不会更新到工程系统中。

在将用户管理下载到 HMI 设备时，用户视图中的所有更改都会被覆盖。

在简单用户视图中更改用户数据

请按如下步骤进行操作：

1. 在用户视图中，触摸想要更改其数据的用户。
2. 使用与创建用户时完全相同的步骤来输入数据。

在高级用户视图中更改用户数据

请按如下步骤进行操作：

1. 在用户视图中，触摸想要更改其数据的用户。

2. 使用与创建用户时完全相同的步骤来输入数据。

结果

用户数据已更改。

参见

概述 (页 1386)

8.12.4.3 删除用户

要求

- 已打开包含用户视图的画面。
- 您必须以管理员权限登录或拥有删除用户的用户管理权限。

注意事项

在用户视图中进行的更改将在运行系统中立即生效。在运行系统中进行的更改不会更新到工程系统中。

在将用户管理下载到 HMI 设备时，用户视图中的所有更改都会被覆盖。

触摸操作的步骤

请按如下步骤进行操作：

1. 在用户视图中触摸要删除的用户。
2. 删除用户名。

键操作的步骤

请按如下步骤进行操作：

1. 使用



键或光标键选择用户视图。

2. 通过光标键在用户视图中选择用户。

3. 按



以删除用户。

结果

用户已删除并且不能再登录到项目。

参见

概述 (页 1386)

8.12.4.4 运行报警

8.12.4.4 概述

报警

在 HMI 设备上，报警指示系统、过程或 HMI 设备本身所发生的事件或状态。状态接收后将报告。

报警可触发下列报警事件：

- 进入
- 离开
- 确认

组态工程师将定义哪些报警必须由用户进行确认。

报警可能包含以下信息：

- 日期
- 时间
- 报警文本
- 事件文本
- 故障位置
- 状态
- 报警类别
- 报警编号
- 报警组
- 支持诊断

报警类别

报警可分为以下几类。此选择取决于 HMI 设备。

- 警告
此类报警通常指示设备状态，例如“电机已启动”。此类报警不需要进行确认。
- 错误
该类报警必须始终进行确认。该类报警通常显示设备的关键错误，例如“电机温度过高”。
- 系统
系统报警指示 HMI 设备本身的状态或事件。

系统报警提供关于诸如发生操作错误或通信故障等的信息。

- 诊断报警

SIMATIC 诊断报警显示 SIMATIC S7 控制器中的状态和事件。

注意

可用于特定设备

诊断报警不可用于精简面板。

- STEP 7 报警类别

在 STEP 7 中组态的报警类别也可用于 HMI 设备。

注意

可用于特定设备

STEP 7 报警类别不可用于精简面板。

- 自定义报警类别

该报警类别的属性必须在组态中定义。

报警缓冲区

报警事件将存储在断电易失的内部缓冲区中。该报警缓冲区的大小取决于 HMI 设备类型。

报警日志

当报警记录激活后，报警事件直接从打印机上打印输出。

可以为每个报警分别设置记录功能。系统向打印机输出“进入”和“离开”报警事件。

“系统”类别的报警向打印机输出必须通过相应的报警缓冲区来启动。这样便将报警缓冲区的内容输出到打印机了。为了启动该打印功能，需要在项目中组态相应的控制对象。

注意

可用于特定设备

报警日志不可用于精简面板。

报警日志

报警事件保存在报警日志中，假设已组态了日志文件。日志文件的容量受限于存储介质和系统限制。

注意

可用于特定设备

报警日志不可用于精简面板。

报警视图

报警视图显示了在报警缓冲区或报警日志中选择的报警或事件。报警事件是否需要确认由组态指定。通过组态，可以过滤显示内容，仅显示包含指定字符串的报警。

报警窗口

如果组态了报警窗口，则它会显示所有未决的报警或特定报警类别中等待确认的报警。报警窗口将在新报警出现时显示。

可以组态报警显示的顺序。将显示当前报警或最后一个报警。报警窗口也可以设置为显示故障的确切地点，以及报警事件的日期和时间。通过组态，可以过滤显示内容，仅显示包含指定字符串的报警。

报警指示器

报警指示器是一个图标，当激活指定报警类别的报警时，该图标便会显示在屏幕上。

报警指示器可采用以下两种状态之一：

- 闪烁：至少存在一条未确认的待决报警。
- 静态：报警已确认，但其中至少有一条尚未离开。显示的数字指示未决报警的数目。

参见

概述 (页 1351)

检测未决报警 (页 1398)

显示报警的信息文本 (页 1399)

确认报警 (页 1400)

编辑报警 (页 1401)

简单报警视图、报警窗口 (页 1396)

8.12.4.4 简单报警视图、报警窗口

应用

简单报警视图显示了在报警缓冲区或报警日志中选择的报警或报警事件。简单报警窗口的布局和操作与简单报警视图的一致。

注意

无法使用脚本动态操作“单个报警视图”对象。

例如，在工程系统中，可以在属性窗口的“动画”组中动态控制对象的可见性。在运行系统中，“简单报警视图”不支持动画。如果已组态某个动画，并希望执行某项操作（例如执行项目一致性检查），则系统会在输出窗口中显示一条错误报警。

**布局**

根据组态的不同，在报警视图中，系统使用不同的列来显示关于报警或报警事件的信息。

为区分不同的报警类别，报警视图中的第一列包含一个图标：

符号	报警类别
!	错误
(空)	警告
(取决于组态)	自定义报警类别
\$	系统

操作







根据组态的不同，您能够：

- 确认报警
- 编辑报警

操作员控件

按钮具有下列功能：

按钮	功能
!	确认报警
↓	编辑报警
?	显示报警的信息文本

按钮	功能
	在单独的窗口(即报警文本窗口)中显示所选报警的完整报警文本。 在报警文本窗口中，可以查看其所需空间超出“报警”视图中可用空间的报警文本。使用下列图标关闭报警文本窗口： 
	向上滚动一个报警
	在报警视图中向上滚动一页
	在报警视图中向下滚动一页
	向下滚动一个报警

操作员控件的布局

简单报警视图中按钮的外观取决于所组态的大小。因此，应检查 HMI 设备上所有必需的按钮是否可用。

参见

概述 (页 1394)

8.12.4.4 检测未决报警

简介

通过以下信息可识别是否出现了必须确认的报警：

- 对带有按键的 HMI 设备：



键的 LED 亮起。

- 取决于组态：画面上显示了报警指示器。

组态决定是否必须对报警进行确认。这也由报警所属报警类别定义。

“ACK”键中的指示灯

LED 是通过键盘集成到 HMI 设备上的



键中的。如果有需要确认的报警，并且必须确认该报警，则 LED 亮起。

当确认了所有需要确认的报警后，该 LED 熄灭。

报警指示器

报警指示器是一个图形符号，它根据组态指示未决报警或需要确认的报警。



图 8-3 具有三个未决报警的报警指示器

布局

可假设报警指示器的状态为以下两种之一：

- 闪烁：
只要存在未决且需要确认的报警，报警指示器就将一直闪烁。显示的数字指示未决报警的数目。通过操作报警指示器，项目工程师可以组态要执行的特定功能。
- 静态：报警已确认，但其中至少有一条尚未离开。

运行系统特性

显示对话框

例如，显示的报警指示器视图被登录对话框、帮助对话框或报警文本窗口所覆盖。关闭这些对话框后，报警指示器即可见。

参见

概述 (页 1394)

8.12.4.4 显示报警的信息文本

触摸操作的步骤

执行下列步骤，显示信息文本：

1. 在报警视图或报警窗口中触摸相关的报警。
报警即被选中。

2. 触摸简单报警视图中的



按钮或高级报警视图中的



。

如果已组态，则将显示分配给该报警的信息文本。

3. 通过



按钮关闭显示信息文本的画面。

键操作的步骤

执行下列步骤，显示信息文本：

1. 在报警视图中选择所期望的报警。

2. 按



键。

如果已组态，则将显示分配给该报警的信息文本。

3. 按



键关闭信息文本。

参见

[概述 \(页 1394\)](#)

8.12.4.4 确认报警

要求

待确认的报警显示在报警窗口或报警视图中。

触摸操作的步骤

要确认报警，请执行以下步骤：

1. 在报警视图或报警窗口中触摸相关的报警。

报警即被选中。

2. 触摸简单报警视图中的



按钮或高级报警视图中的



。

键操作的步骤

报警视图和报警窗口均有一个 Tab 顺序，利用这一顺序可使用键盘来选择操作员控件以及上次选择的报警。

要确认报警，请执行以下步骤：

1. 使用



键选择所需的报警视图或报警窗口。



2. 选择所需报警。相应地使用



、



、

- 
 或

 键。
 3. 按

 键。

备用操作

根据组态的不同，也可使用功能键确认报警。

结果

报警即被确认。如果报警属于某个报警组，将会确认该相应组的所有报警。

参见

概述 (页 1394)

8.12.4.4 编辑报警

简介

组态工程师可为每个报警分配附加功能。在处理报警时将执行这些功能。

注意

在编辑未确认的报警时，将对其自动进行确认。

要求

要编辑的报警显示在报警窗口或报警视图中。

触摸操作的步骤

执行下列步骤，编辑报警：

1. 在报警视图或报警窗口中触摸相关的报警。报警即被选中。
2. 触摸简单报警视图中的



按钮或增强型报警视图中的



按钮。

键操作的步骤

执行下列步骤，编辑报警：

1. 使用



选择所需的报警视图或报警窗口。

2. 选择所需报警。使用



、



、



或



键。

3. 继续按



键，直到选中简单报警视图中的



按钮或扩展报警视图中的



按钮。

4. 按下



键确认输入。

结果

系统执行报警的附加功能。有关此主题的更多信息，请参见您的设备文档。

参见

[概述 \(页 1394\)](#)

8.12.4.5 操作配方

8.12.4.5 配方的结构

配方

可将生产产品系列的配方集合当成文件柜。用于制造产品的配方相当于文件柜中的抽屉。

实例：

在生产果汁的工厂中，需要满足不同口味的配方。例如，有橙味、葡萄味、苹果味和樱桃味的配方。



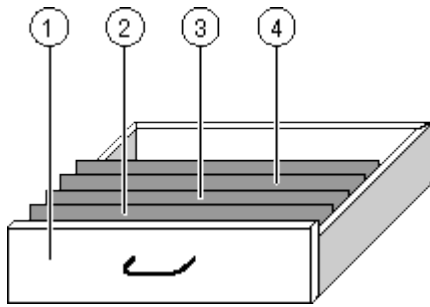
① 文件柜	配方集合	果汁厂配方
② 抽屉	配方	橙味饮料
③ 抽屉	配方	葡萄味饮料
④ 抽屉	配方	苹果味饮料
⑤ 抽屉	配方	樱桃味饮料

配方数据记录

文件柜抽屉装满了搁置的文件夹。抽屉中搁置的文件夹代表制造产品的不同种类时所需的记录。

实例：

例如，苹果味产品的不同种类可能是软饮料、果汁或蜜露。



①	抽屉	配方	苹果味饮料产品的不同种类
②	搁置的文件夹	配方数据记录	苹果饮料
③	搁置的文件夹	配方数据记录	苹果蜜露
④	搁置的文件夹	配方数据记录	苹果汁

元素

在显示文件柜的图中，每个搁置的文件夹均包含相同的页数。搁置的文件夹中的每一页对应于配方数据记录中的一个元素。一份配方的所有记录均含有相同的元素。不过，各记录中的各个元素的值并不相同。

实例：

所有饮料均含有相同的成份：水、浓缩物、糖和香料。不过，软饮料、果汁或蜜露在生产时所记录的使用的糖含量不同。

参见

[概述 \(页 1351\)](#)

[项目中的配方 \(页 1404\)](#)

[简单配方视图 \(页 1406\)](#)

[创建配方数据记录 \(页 1409\)](#)

[编辑配方数据记录 \(页 1410\)](#)

[删除配方数据记录 \(页 1411\)](#)

[从 PLC 读取配方数据记录 \(页 1412\)](#)

[将配方数据记录传送至 PLC \(页 1412\)](#)

8.12.4.5 项目中的配方

概述

如果在项目中使用配方，以下组件将会相互影响：

- HMI 设备配方内存
配方以数据记录的形式保存在 HMI 设备的配方内存中。
- 配方视图/配方画面
在 HMI 设备上，在配方视图或配方画面中显示和编辑配方。

- HMI 设备内部存储器中的配方数据记录在配方视图中进行显示和编辑。
- 在配方画面中显示和编辑配方变量的值。

注意

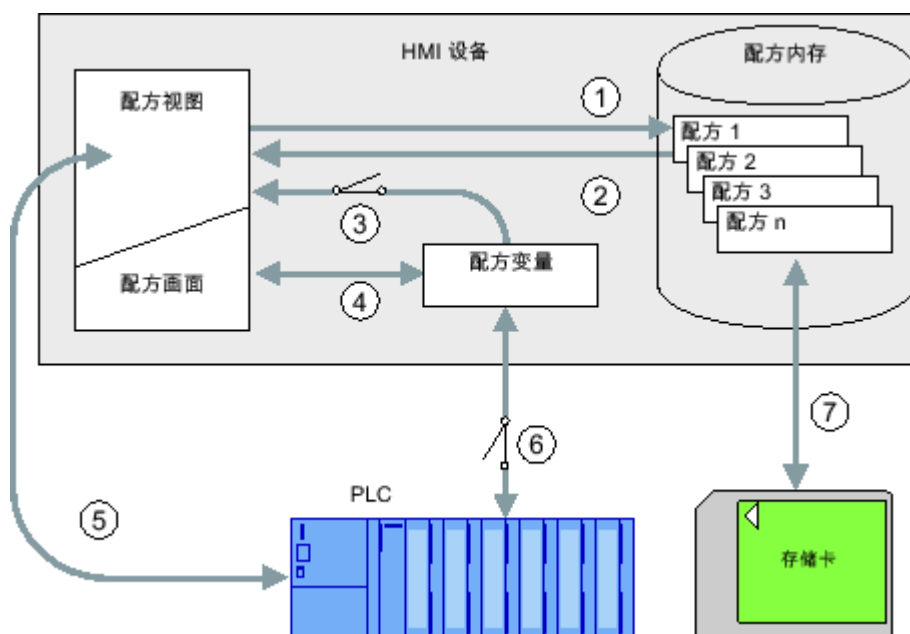
可在各种配方中组态相同的配方变量。如果修改了某个配方变量的值，则同步处理会更改该变量在所有其它配方中的值。

- 配方变量

配方变量包含配方数据。在配方画面中编辑配方时，配方值将保存在配方变量中。在哪一点与 PLC 交换配方变量值取决于组态。

数据流

下图说明了具有配方的项目中的数据流：



- ① 编辑、保存或删除配方数据记录
- ② 显示配方数据记录
- ③ 同步或不同步配方变量
- ④ 在配方画面中显示和编辑配方变量
- ⑤ 将配方视图中的记录写入 PLC 或从 PLC 读取记录并在配方视图中进行显示。
- ⑥ 在线或离线将配方变量传送到 PLC
- ⑦ 将配方数据记录导出或导入存储卡

参见

配方的结构 (页 1403)

8.12.4.5 简单配方视图

布局

简单配方视图由三个区域组成：

- 配方列表
- 数据记录列表
- 元素列表

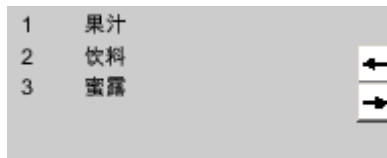


图 8-4 简单配方视图 - 带有数据记录列表的实例

在简单配方视图中，每一区域均独立地显示在 HMI 设备上。可使用快捷菜单操作其中的每一个显示区域。

简单配方视图始终以配方列表开始。

操作

根据组态的不同，可以按照如下步骤使用简单配方视图：

- 创建、修改、复制或者删除配方记录
- 从 PLC 中读取配方数据记录或将其传送到 PLC



简单配方视图的操作员控件

可在显示区域和快捷菜单之间进行切换来操作简单配方视图。

下表说明显示区域的操作信息。

	按键式	功能
触摸条目		相邻的最低显示区域打开，即数据记录列表或元素列表。
		打开先前的显示区域。
		打开显示区域的快捷菜单。

下表说明快捷菜单的操作信息：

	按键	功能
		关闭菜单。 打开显示区域。
触摸菜单命令	输入菜单命令的编号	执行相应的菜单命令。

简单配方视图的快捷菜单

通过按







按钮或






键，可调用各视图区域的快捷菜单。快捷菜单列出了激活视图区域中可用的命令。编号将分配给每个命令。可通过输入编号来执行命令。也可使用系统键来执行特定命令。




范围取决于 HMI 设备。

- 配方列表



编号	菜单命令	键	功能
0	新建	 + 	为所选配方创建新的配方数据记录。 如果组态了初始值，则它会显示在输入域中。
1	显示信息文本		显示为简单配方视图组态的信息文本。
2	打开		打开所选配方的记录列表。

- 数据记录列表



	菜单命令	键	功能
	新建	 + 	创建新的配方数据记录。 如果组态了初始值，则它会显示在输入域中。
	删除		删除显示的记录。
	另存为		所选数据记录会以不同的名称保存。将打开一个对话框，可在其中输入名称。
	重命名		重命名所选的数据记录。将打开一个对话框，可在其中输入名称。

	菜单命令	键	功能
	打开		随即会打开所选数据记录的元素列表。
	上一步		配方列表打开。
	到 PLC		将所选数据记录的显示值从 HMI 设备传送到 PLC。
	来自 PLC		在 HMI 设备的配方视图中显示 PLC 中的配方值。
	显示信息文本		显示为简单配方视图组态的信息文本。

• 元素列表

	菜单命令	键	功能
	保存		重命名所选记录。
	传送到 PLC		将所选数据记录的显示值从 HMI 设备传送到 PLC。
	来自 PLC		在 HMI 设备的配方视图中显示来自 PLC 的配方值。
	另存为		该数据记录会以新的名称保存。将打开一个对话框，可在其中输入名称。
	显示信息文本		显示为简单配方视图组态的信息文本。
	重命名		重命名所选的数据记录。将打开一个对话框，可在其中输入名称。
	上一步		数据记录列表打开。

简单配方视图的鼠标控制或触摸屏控制

1. 从配方视图中选择所期望的配方。
2. 单击  按钮。
快捷菜单打开。
3. 选择所需的菜单命令。
执行相应的菜单命令。
4. 此外，也可以在配方视图中打开所期望的配方。
显示数据记录列表。
5. 打开期望的数据记录。也可使用  按钮打开快捷菜单，并选择一个菜单命令。

执行相应的菜单命令。

使用带简单配方视图的键盘

1. 根据需要按



键多次，以选择简单配方视图。

2. 使用光标键选择期望的配方。

3. 按下



键。

快捷菜单打开。

4. 根据需要按



光标键多次，以选择菜单命令。

5. 按下



键确认菜单命令。

6. 此外，也可按下期望的菜单命令的编号。

执行相应的菜单命令。

参见

配方的结构 (页 1403)

8.12.4.5 创建配方数据记录

简介

在配方列表或记录列表中创建新的配方数据记录。然后在元素列表中输入新记录的值并保存该记录。

要求

已显示具有简单配方视图的画面。

步骤

执行下列步骤，创建配方数据记录：

1. 如果配方列表包含几个配方：选择要为其创建新配方数据记录的配方。
2. 打开配方列表菜单。
3. 选择“0 新建”菜单命令。

创建一条新记录。

随即打开该新记录的元素列表。

4. 输入数据记录各元素的值。

根据组态情况，可将默认值分配给记录的各个变量。

5. 打开元素列表的菜单，然后选择“0 保存”菜单命令。
6. 为新记录输入名称。
7. 确认输入。

如果将现有数据记录的编号分配给新数据记录，则会覆盖现有的数据记录。

结果

新的配方数据记录已经保存到所选的配方中。

参见

配方的结构 (页 1403)

8.12.4.5 编辑配方数据记录

简介

编辑配方数据记录的值并将其保存到配方视图中。

与 PLC 同步

要在简单配方视图中显示来自 PLC 的当前配方值，首先必须使用菜单命令“2 来自 PLC”从 PLC 读取元素列表中的实际值。

仅在使用菜单命令“1 到 PLC”将修改的数据记录传送到 PLC 之后，才能激活配方视图中发生更改的值。

要求

已显示具有配方视图的画面。

步骤

执行下列步骤，复制配方数据记录：

1. 如果配方列表包含多个配方：选择包含相关配方数据记录的配方。
2. 打开数据记录列表。
3. 选择想要编辑的配方数据记录。
4. 打开元素列表。
5. 按要求编辑元素值。
6. 使用菜单命令“0 保存”保存更改。

结果

编辑后的配方数据记录保存到所选配方中。

参见

配方的结构 (页 1403)

8.12.4.5 删除配方数据记录

简介

可删除所有不需要的数据记录。

要求

已显示具有简单配方视图的画面。

触摸操作的步骤

执行下列步骤，删除新配方数据记录：

1. 如果配方列表包含多个配方：选择包含相关配方数据记录的配方。
2. 打开数据记录列表。
3. 选择想要删除的数据记录。
4. 打开菜单。
5. 选择菜单命令“1 删除”。

键操作的步骤

执行下列步骤，删除新配方数据记录：

1. 如果配方列表包含多个配方：选择包含相关配方数据记录的配方。
2. 打开数据记录列表。
3. 选择想要删除的数据记录。
4. 按下



。

结果

删除数据记录。

参见

配方的结构 (页 1403)

8.12.4.5 从 PLC 读取配方数据记录

简介

配方元素的值通过变量与 PLC 进行交换。

可直接在激活项目内的设备级别编辑已保存到 HMI 设备中配方的值；如果设备中阀的打开程度大于配方中所示程度，则可能属于上述情况。HMI 设备上变量的值有可能与 PLC 中的值不再匹配。

从 PLC 读取值并将这些值输出到配方视图，以同步配方值。

要求

已显示具有简单配方视图的画面。

步骤

要将配方数据记录传送到 PLC 中，请执行以下步骤：

1. 如果配方列表包含多个配方：选择包含相关配方数据记录的配方。
2. 选择要应用来自 PLC 的值的配方数据记录元素列表。
3. 打开菜单。
4. 选择菜单命令“2 来自 PLC”。
从 PLC 读取值。
5. 使用菜单命令“0 保存”将显示的值保存到 HMI 设备。

结果

这些值已从 PLC 中读出，当前在 HMI 设备上可见并且已保存到了所选的配方数据记录中。

注意 精简面板

通过精简面板，还可以为数据记录列表组态“来自 PLC”菜单命令：这种情况下，还可以在数据记录列表中选择“来自 PLC”菜单命令。

参见

配方的结构 (页 1403)

8.12.4.5 将配方数据记录传送到 PLC

简介

必须将值传送到 PLC，以激活为此过程更改的配方数据记录。

配方视图中显示的值将传送到 PLC 中。

要求

已显示具有简单配方视图的画面。

步骤

要将配方数据记录传送到 PLC 中，请执行以下步骤：

1. 如果配方列表包含多个配方：选择包含相关配方数据记录的配方。
2. 选择要将其值传送到 PLC 的配方数据记录的元素列表。
3. 打开菜单。
4. 选择菜单命令“1 到 PLC”。

结果

配方数据记录的值已传送到 PLC 并在过程中激活。

注意 精简面板

通过精简面板，还可以为数据记录列表组态“传送到 PLC”菜单命令：这种情况下，还可以在数据记录列表中选择“传送到 PLC”菜单命令。

参见

配方的结构 (页 1403)

8.13 性能特点

8.13.1 工程系统

工程系统

工程系统中的组态受主内存资源的限制。根据操作系统的不同，WinCC 最多使用 2 GB 的 RAM。

在并行运行多个需要大量内存的应用程序时，需要在 PC 上安装 2 GB 以上的 RAM。

下面所述的组态对主内存的装载具有重要影响：

- 动画量大
- 使用大图形对象
- 一个项目组态中有多台设备

HMI 设备的性能规范

下表可帮助评估您的项目是否符合 HMI 设备的性能规范。

[精简面板 \(页 1414\)](#)

8.13.2 精简面板

简介

下表可帮助评估您的项目是否符合 HMI 设备的性能规范。

指定的最大值不可附加。不能保证设备上的组态操作可以在完全系统限制下运行。

除了指定的限制，还需要考虑组态存储器资源的限制。

精简面板

	KTP400 Basic	KTP600 Basic	KTP1000 Basic	TP1500 Basic
变量				
项目中的变量数目	128	128	256	256
PowerTag 的数目	--	--	--	--
每个数组中的元素数目	100	100	100	100
局部变量的数目	--	--	--	--
报警				
报警类别的数目	32	32	32	32
离散量报警的数目	200	200	200	200
模拟量报警的数目	15	15	15	15
报警的字符长度	80	80	80	80
每个报警的过程值数目	8	8	8	8
报警缓冲区的大小	256	256	256	256
队列中的报警事件数目	64	64	64	64
画面				
画面数目	50	50	50	50
每个画面的域数目	30	30	30	30
每个画面的变量数目	30	30	30	30
每个画面的复杂对象的数目	5	5	5	5
配方				
配方的数目	5	5	5	5

	KTP400 Basic	KTP600 Basic	KTP1000 Basic	TP1500 Basic
每个配方中的元素数目	20	20	20	20
每条数据记录的用户数据长度 (以字节为单位)	--	--	--	--
每个配方的数据记录数目	20	20	20	20
项目中配方元素的数目	--	--	--	--
内部闪存中为数据记录保留的存储空间	40 KB	40 KB	40 KB	40 KB
记录				
记录数	--	--	--	--
每个记录文件中的条目数 (包括所有的记录段) ¹⁾	--	--	--	--
记录段的数目	--	--	--	--
变量记录的周期性触发	--	--	--	--
可记录的变量数目 ⁴⁾	--	--	--	--
趋势				
趋势数目	25	25	25	25
文本列表和图形列表				
图形列表的数目	100	100	100	100
文本列表的数目	150	150	150	150
总列表数	150	150	150	150
每个文本或图形列表的条目数	30	30	30	30
图形对象的数目	500	500	500	500
文本元素的数目	500	500	500	500
脚本				
脚本数目	--	--	--	--
通信				
连接数目	4	4	4	4
基于“SIMATIC HMI http 协议”的连接数目	--	--	--	--
帮助系统				
帮助文本中的字符数	320	320	320	320
语言				
运行系统语言种数	5	5	5	5
调度器				
任务	--	--	--	--
用户管理				

	KTP400 Basic	KTP600 Basic	KTP1000 Basic	TP1500 Basic
用户组	50	50	50	50
授权	32	32	32	32
密码	50	50	50	50
项目				
项目文件“.fwx”的大小	512 KB	512 KB	1024 KB	1024 KB

1)	所有分段循环日志的条目数适用于“分段循环日志”记录方法。不得超过循环日志数乘以本记录中的数据记录数得到的乘积。
----	---

参见

工程系统 (页 1413)

8.14 显示交叉引用

8.14.1 关于交叉引用的常规信息

简介

交叉引用列表提供有关对象在项目中的使用情况的概述。

交叉引用的使用

交叉引用列表提供以下好处：

- 创建和更改程序时，保留已使用的对象、变量和报警等的概述。
- 可以从交叉引用直接跳转到对象的使用位置。
- 可在调试时了解以下信息：
 - 在特定画面中使用的对象。
 - 在特定显示中显示的报警和配方。
 - 在特定报警或对象中使用的变量。
- 作为项目文档的一部分，交叉引用提供所有对象、报警、配方、变量和画面的全面概述。
- 例如，可以显示对象的使用位置，以便修改或删除对象。
- 可以显示已删除对象的使用点，然后进行修改。

8.14.2 显示交叉引用列表

要求

已创建项目。

简介

可以在项目视图中显示下列对象的交叉引用：

- HMI 设备
- 所有文件夹
- 所有编辑器

显示交叉引用

执行下列步骤，显示交叉引用：

1. 在项目视图中，选择所需条目，然后选择快捷菜单命令“交叉引用”。
将显示交叉引用列表。
2. 单击“使用位置”按钮，显示交叉引用列表中所显示的对象的使用位置。
3. 单击“使用”按钮，查看交叉引用列表中所显示的对象的用户。
4. 使用工具栏中的按钮可以执行以下操作：
 - 更新交叉引用列表
 - 进行交叉引用列表的设置
 - 折叠条目
 - 展开条目
5. 通过单击相应的列标题，可以按升序或降序对“对象”列中的条目排序。
6. 要转到特定对象的使用位置，请单击显示的链接。

8.14.3 交叉引用列表的结构

交叉引用列表的视图

交叉引用列表有两个视图。这两个视图之间的差异在于第一列中所显示的对象：

- 使用位置：
显示被引用的对象。您可在此处看到对象的使用位置。
- 使用的：
显示引用对象。此处显示对象的用户。

分配的工具提示提供关于每个对象的附加信息。

交叉引用列表的结构

交叉引用列表具有以下结构：

列	内容/含义
对象	使用较低层级对象或正被较低层级对象使用的对象的名称。
次数	使用次数
使用位置	每个使用位置，例如对象或事件
属性	被引用对象的功能，例如用于数据记录或过程值的变量
已连接到	对象与之连接的 PLC 变量。
类型	对象类型
路径	对象路径

视安装的产品而定，交叉引用可能显示额外的列或不同的列。

交叉引用列表的设置

可以使用工具栏中的图标为交叉引用列表进行以下设置：

- 更新交叉引用列表
更新当前交叉引用列表。
- 进行交叉引用列表的设置
可在此处指定显示所有已使用对象、所有未使用对象、所有已定义对象还是所有未定义对象。如果启用“未定义对象”选项，也将显示对先前所删除的对象的引用。
- 折叠条目
通过关闭较低层级对象，减少当前交叉引用列表中的条目。
- 展开条目
通过打开较低层级对象，展开当前交叉引用列表中的条目。

交叉引用列表的排序

可以按升序或降序对“对象”列中的条目排序。单击相应列标题进行排序。

8.14.4 在巡视窗口中显示交叉引用

简介

巡视窗口显示在“关于”>“交叉引用”选项卡中选择的对象的相关交叉引用信息。此选项卡显示所选对象的使用位置的实例和使用所选对象的其它对象的实例。

结构

巡视窗口以表格形式显示交叉引用信息。每列包含有关所选对象及其应用的特定信息和详细信息。下表显示在“关于”>“交叉引用”选项卡中列出的附加信息：

列	含义
对象	使用较低层级对象或正被较低层级对象使用的对象的名称。
次数	使用次数
使用位置	每个使用位置，例如对象或事件
属性	被引用对象的功能，例如用于数据记录或过程值的变量
已连接到	对象与之连接的 PLC 变量。
类型	对象类型
路径	对象路径

视安装的产品而定，交叉引用可能显示额外的列或不同的列。

使用在线和诊断功能

9.1 有关在线模式的常规信息

在线模式

在线模式下，编程设备/PC 和一个或多个设备之间会建立在线连接。

要将程序和组态数据装载到设备以及要执行下列操作，编程设备/PC 与设备之间必须建立在线连接：

- 测试用户程序
- 显示和切换 CPU 的工作模式
- 显示和设置 CPU 的日期和日时钟
- 显示模块信息
- 比较在线块和离线块
- 硬件诊断

在建立在线连接之前，必须通过工业以太网接口连接编程设备/PC 和设备。然后才能在在线和诊断视图中或使用“在线工具”(Online tools) 任务卡访问设备上的数据。设备的当前在线状态由项目树中该设备右边的图标指示。在相关的工具提示中可找到各个状态图标的含义。

注意

某些在线功能取决于安装软件的功能范围或者项目是否已打开。

在线显示

成功建立在线连接之后，用户界面将发生如下变化：

- 激活窗口的标题栏背景色现在为橙色。
- 相关站点的非激活窗口的标题栏下方现在有一条橙色线。
- 状态栏的右侧会显示一个橙色条。
- 项目树中显示了相关工作模式的对象或站的诊断符号。
- “诊断 > 设备信息”(Diagnostics > Device information) 区域置于巡视窗口的前景。

9.2 在线访问

项目的在线访问

项目树的“在线访问”(Online access) 区域中显示了所有可用的网络访问方式，编程设备/PC 使用这些网络访问方式可与所连接的目标系统建立在线连接。

每个接口图标都提供了相应接口的状态信息。还可以显示可访问设备以及显示和编辑接口属性。

状态显示

接口的当前状态通过其名称右边的图标指示。可以在工具提示中查看图标的含义。

显示或更新可访问设备

如果要在编程设备/PC 上显示所有可在线访问的设备，用户有以下选择：

- 在项目树中显示编程设备/PC 的某个接口上的可访问设备。在项目树中还可以显示有关各个可访问设备的附加信息。
- 在列表中显示所有接口的可访问设备。

接口的属性

对于每个接口，都可以显示其属性，而且有时可修改属性，例如网络类型、地址和状态。

参见

显示可访问设备 (页 1422)

打开接口的属性 (页 1423)

9.3 显示可访问设备

可访问设备

可访问设备是指所有连接到编程设备/PC 接口上并且已经开启的设备。也可显示那些使用当前安装的产品仅允许进行有限组态或根本无法组态的设备。

在项目树中显示编程设备/PC 某个接口上的可访问设备

要显示编程设备/PC 单个接口上的可访问设备，请按以下步骤操作：

1. 在项目树中打开“在线访问”(Online access) 文件夹。
2. 单击接口左侧的箭头，在接口下方排列显示所有对象。
3. 双击接口下方的“更新可访问的设备”(Update accessible devices) 命令。

所有可通过该接口访问的设备都将显示在项目树中。

在列表中显示可访问设备

要在总览列表中显示所有可用接口上的可访问设备，请按以下步骤操作：

1. 在“在线”(Online) 菜单中，选择“可访问设备”(Accessible devices) 命令。
将显示“可访问设备”(Accessible devices) 对话框。
2. If your programming device / PC has multiple interfaces, choose the interface you require, for example Ethernet, in the "PG/PC interface to show accessible devices for:" box.
如果接口上没有可用设备，则编程设备/PC 和设备间的连线将显示为断开。如果可以访问设备，连线将显示为未断开，并且编程设备/PC 所选接口上的可访问设备将显示在列表中。
3. 如果安装了新设备，请单击“刷新”(Refresh) 按钮。
4. 要转到项目树中的某个设备，请从可访问设备列表中选择该设备并单击“在项目树中显示”(Show in project tree) 按钮。
与所选设备连接的接口将在项目树中显示为选中状态。

在项目树中显示有关可访问设备的附加信息

要在项目树中显示有关可访问设备的附加信息，请按以下步骤操作：

1. 在项目树中单击某个可访问设备左侧的箭头。
将显示已知设备的所有在线可用数据，例如块和系统数据。

9.4 打开接口的属性

简介

对于每个接口，都可以显示其属性，而且有时可修改属性，例如网络类型、地址和状态。

步骤

要打开属性，请按以下步骤操作：

1. 在项目树中，右键单击“在线访问”(Online access) 下方的所需接口。
2. 从快捷菜单中选择“属性”(Properties) 命令。

将打开“属性”(Properties) 对话框。在该对话框的左侧，可以看到区域导航。可以在区域导航的各个条目中查看当前参数设置，并在必要时进行更改。

9.5 设置以太网接口的参数

9.5.1 设置工业以太网接口的参数

工业以太网接口的参数设置中的选项

设置工业以太网接口的参数时，用户可选择以下参数：

- 取决于操作系统的参数

9.5 设置以太网接口的参数

工业以太网接口的一些参数在操作系统中进行设置，对所有连接设备均有效。这些参数设置仅在此处显示，但可以在操作系统的网络设置中进行更改。

- 可在软件中设置的参数

注意

请注意，更改接口参数会直接影响操作系统和编程设备/PC。请记住，只有拥有足够的用户权限才能更改某些参数设置。

工业以太网接口的参数

下表概括了通过操作系统进行设置并可由用户更改的工业以太网接口参数。

无法更改的参数设置	可以设置的参数
MAC 地址	IE-PG 访问以及 TCP/IP 的快速确认
DHCP 服务器激活/禁用	IE-PG 访问以及 TCP/IP 的超时
APIPA 激活/禁用	LLDP
IP 地址	网络适配器的附加动态 IP 地址
子网掩码	
DNS 地址	
DHCP 地址	

参见

[显示操作系统参数 \(页 1424\)](#)
[将 PG/PC 接口连接到子网 \(页 1425\)](#)
[设置以太网接口的参数 \(页 1425\)](#)
[分配临时 IP 地址 \(页 1426\)](#)
[管理临时 IP 地址 \(页 1427\)](#)
[用户权限的影响 \(页 99\)](#)

9.5.2 显示操作系统参数

以太网接口是操作系统的一部分。因此，网络适配器的所有参数都可以在操作系统的网络设置中进行调整。

可以在软件中显示下列参数：

- 网络适配器的物理地址
- DHCP 服务器激活或禁用时分配的 IP 地址
- 操作系统激活或禁用时分配的私有 IP 地址
- 当前静态 IP 地址

- 分配的子网掩码
- DNS 地址
- DHCP 地址

如果要修改参数设置，请参考操作系统或网络适配器的文档。

显示以太网接口的当前参数

要显示以太网接口的当前参数，请按以下步骤操作：

1. 在项目树的“在线访问”(Online access) 中选择以太网接口。
2. 在接口的快捷菜单中选择“属性”(Properties) 命令。
将打开用于组态接口的对话框。
3. 在区域导航中选择“组态 > 工业以太网”(Configurations > Industrial Ethernet)。

参见

[设置以太网接口的参数 \(页 1425\)](#)

9.5.3 将 PG/PC 接口连接到子网

如果创建了多个子网，可以指定以太网接口所连接的子网。

步骤

要选择与以太网接口连接的子网，请按以下步骤操作：

1. 在项目树的“在线访问”(Online access) 中选择以太网接口。
2. 在接口的快捷菜单中选择“属性”(Properties) 命令。
将打开用于组态接口的对话框。
3. 转到“常规 > 分配”(General > Assignment)，并在“子网的连接”(Connection to subnet) 下拉列表中选择编程设备/PC 的以太网接口所要连接的子网。
4. 单击“确定”(OK) 关闭对话框。

9.5.4 设置以太网接口的参数

可以在软件中直接调整与网络协议有关的参数设置。

要求

您必须有足够的用户权限。

另请参见：[用户权限的影响 \(页 99\)](#)。

步骤

要更改与网络协议有关的参数设置，请按以下步骤操作：

1. 在项目树的“在线访问”(Online access) 中选择以太网接口。
2. 在接口的快捷菜单中选择“属性”(Properties) 命令。
将打开用于组态接口的对话框。
3. 选择“组态 > IE-PG 访问”(Configurations > IE-PG access) 来调整与网络管理相关的协议设置。
 - 选中“快速确认”(Fast acknowledge) 复选框以更快地响应较小的网络包。
 - 从“超时”(Timeout) 下拉列表中选择可用于检测网络节点的最长时间。
4. 要激活 LLDP 协议并更精确地发现网络拓扑，请选中“组态 > LLDP”(Configurations > LLDP) 中的“LLDP 激活”(LLDP active) 复选框。
5. 选择“组态 > TCP/IP”(Configurations > TCP/IP) 以针对运行期间的网络通信量调整 TCP/IP 协议。
 - 选中“快速确认”(Fast acknowledge) 复选框以更快地响应较小的网络包。
 - 从“超时”(Timeout) 下拉列表中选择可用于与网络节点通信的最长时间。

参见

用户权限的影响 (页 99)

显示操作系统参数 (页 1424)

9.5.5 分配临时 IP 地址

添加动态 IP 地址

如果某个设备的 IP 地址与网络适配器的 IP 地址不属于同一个子网，则需要先给适配器另外分配一个 IP 地址，其子网地址与设备的相同。只有这样才能在该设备与编程设备/PC 之间进行通信。

要执行在线操作时，如果编程设备/PC 的当前 IP 地址尚未处于正确的子网，则系统将自动建议您分配临时 IP 地址。

在重启编程设备/PC 或您手动删除临时分配的 IP 地址之前，该地址一直有效。

注意

用户需要有足够的权限才能分配临时 IP 地址。

另请参见：[用户权限的影响 \(页 99\)](#)

参见

管理临时 IP 地址 (页 1427)

9.5.6 管理临时 IP 地址

如果设备的 IP 地址与网络适配器的当前静态 IP 地址不属于同一个子网，则可以给适配器分配一个合适的动态 IP 地址，其子网地址与设备的相同。

您可以显示所有临时分配的地址以及删除这些地址。

要求

要删除这些地址，您需要有足够的权限。

步骤

要显示和删除临时分配的地址，请按以下步骤操作：

1. 在项目树的“在线访问”(Online access) 中选择以太网接口。
2. 在接口的快捷菜单中选择“属性”(Properties) 命令。
将打开用于组态接口的对话框。
3. 选择“组态 > IE-PG 访问”(Configurations > IE-PG access)。
将显示一个包含已分配的 IP 地址的表格。
4. 单击“删除项目特定的 IP 地址”(Delete project-specific IP addresses) 按钮以一次性删除所有 IP 地址。

参见

用户权限的影响 (页 99)

9.6 建立和取消在线连接

要求

- 至少安装有一个 PG/PC 接口并且该接口与某个设备间存在物理连接，例如通过以太网电缆建立的连接。
- 该设备的 IP 地址必须与编程设备/PC 的 IP 地址位于同一个子网。

转到在线

要建立在线连接，请按以下步骤操作：

1. 在项目树中，选择要建立在线连接的设备。
2. 在“在线”(Online) 菜单中，选择“转到在线”(Go online) 命令。
如果该设备曾经连接到某个特定 PG/PC 接口，则会自动与之前的 PG/PC 接口建立在线连接。如果之前没有连接，则打开“转到在线”(Go online) 对话框。
3. 如有必要，从“转到在线”(Go online) 对话框的“PG/PC 接口将在线连接到”(PG/PC interface to go online with) 下拉列表中选择要连接到设备的接口。
所有可通过所选接口访问的设备将显示在“目标子网中的可访问设备”(Accessible devices in target subnet) 下。
4. 可选：单击“更新”(Update) 按钮以重新显示可访问设备的列表。

9.6 建立和取消在线连接

5. 可选：单击图形左侧的“闪烁 LED”(Flash LED) 按钮以执行 LED 闪烁测试。使用该功能，可以检查是否选择了正确的设备。并非所有设备都支持 LED 闪烁测试。
6. 选择节点并单击“转到在线”(Go online) 按钮。
将建立与所选目标设备的在线连接。

结果

建立在线连接后，编辑器的标题栏变为橙色。编辑器的标题栏和状态栏中也将显示一个橙色活动栏。

将存储该连接路径，以便将来进行连接尝试。除非要选择新的连接路径，否则不必再打开“转到在线”(Go online) 对话框。

注意

如果未显示任何可访问设备，请为 PG/PC 接口选择其它网络访问或检查该接口的设置。

取消在线连接

要断开现有在线连接，请按以下步骤操作：

1. 在项目树中选择要断开连接的设备。
2. 在“在线”(Online) 菜单中，选择“转到离线”(Go offline) 命令。

参见

分配临时 IP 地址 (页 1426)
用户权限的影响 (页 99)

源文件

已用文档列表

- STEP 7 V10.5 SP2 (12/2009, zh-CHS)

索引

*

*.bmp, 990
*.emf, 990
*.gif, 990
*.ico, 990
*.jpeg, 990
*.jpg, 990
*.tif, 990
*.wmf, 990

A

ABS, 674, 802
ACK, 1398
AcknowledgeAlarm, 1266
ACOS, 690, 819
ActivateCleanScreen, 1246
ActivatePreviousScreen, 1247
ActivateScreen, 1245
ActivateScreenByNumber, 1246
ADD, 564, 573, 663, 791
ADD 指令
 EN/ENO 保护电路, 332
AdjustContrast, 1244
Admin, 1393
AlarmViewAcknowledgeAlarm, 1265
AlarmViewEditAlarm, 1264
AlarmViewShowOperatorNotes, 1265
AND, 721, 849
 先与后或逻辑运算, 745
 逻辑运算, 568, 569, 575, 742, 849
ARRAY
 全局数据块中的声明, 475
 块接口中的声明, 403
 实例, 355
 数据访问, 430, 459
 格式, 354
 用作参数, 430, 459
ASCII TSAP, 283
ASCII 代码表, 283
ASIN, 689, 817
ATAN, 692, 820
ATTACH, 922

Automation License Manager, 58

B

BOOL, 345
ButtonPress, 1273
ButtonRelease, 1274
BYTE, 346

C

CalibrateTouchScreen, 1255
CAN_DINT, 925
CEIL, 708, 837
ChangeConnection, 1284
CHAR, 351
Char
 数组变量, 53
ClearAlarmBuffer, 1263
ClearAlarmBufferProtoolLegacy, 1263
CONVERT, 705, 834
COS, 686, 815
Counter
 加计数和减计数, 570
CPU
 从硬件目录中选择, 229
 使用 STP 切换到 STOP 模式, 887
 切换工作模式, 270
 属性, 231
 所有存储器的使用率, 265
 插入信号板, 202
 显示当前 LED 状态, 265
 读出诊断缓冲区, 267
CPU 存储区
 显示, 518
CPU 属性, 232
CPU 控制面板
 显示区域, 265
CPU 模式
 显示, 95
CSV 文件, 1330, 1332
CTD, 643, 771
CTRL_HSC, 647, 775

CTRL_PWM, 942
CTU, 641, 770
CTUD, 562, 570, 644, 773

D

DEC, 673, 800
DECO, 727, 855
DecreaseFocusedValue, 1283
DecreaseTag, 1283
DETACH, 923
DINT, 349
DIS_AIRT, 925
DIV, 564, 573, 668, 795
DTL, 352
DTL 数据类型
 限制, 54
DWORD, 346
 说明, 21

E

EditAlarm, 1249
EN/ENO 机制, 332
EN_AIRT, 926
ENCO, 728, 857
EXP, 684, 812
EXPT, 694, 823

F

FB, 320
FB1030, 304
FB1031, 304
FB1071, 304
FB1072, 304
FB1073, 304
FB1081, 304
FB1082, 304
FB1100, 304
FB1101, 304
FB1102, 304
FB1103, 304
FB1104, 304
FB1105, 304
FB1107, 304
FB1108, 304
FB1130, 304
FBD, 440
FBD 元素
 删除, 455
 剪切, 455
 复制, 455

插入, 447, 448
插入和编辑操作数, 457
插入操作的规则, 446
 替换, 454
 粘贴, 455

FBD 指令

 实例, 567

FBD 程序段

 分支, 462
 分支的规则, 462
 插入分支, 463
 插入块调用, 421, 451
 插入所提供的全局库中的元素, 422, 452
 断开分支, 463
 程序状态, 527

FBT1, 304
FBT1083, 304
FBT2, 304
FC, 320
FC1070, 304
FC1080, 304
FILL_BLK, 700, 829
FLOOR, 709, 838
FRAC, 693, 821

G

GET_ERROR, 470
GET_ERRORID, 470
GetError, 887
 错误信息, 887
GetErrorID, 890
GetGroupNumber, 1261
GetPassword, 1261
GetUserName, 1260
GoToEnd, 1253
GoToHome, 1253

H

HMI 画面, 89
HMI 设备, 71
 传送项目, 1328
 可用的系统函数, 1241
 性能特点, 1414
 恢复数据, 1347, 1348, 1358
 数据备份, 1347, 1348, 1358
 更改设备类型, 1339
 系统限制, 1414
 调试, 1328
 连接, 1329
HMI 设备类型
 更改, 1339
HMI 连接, 221

HMI 设备
 调试, 1352
 重新调试, 1352
 HSC, 241
 HTML 浏览器, 969

I

I 地址, 237
 I/O
 立即写, 431, 460
 立即读, 431, 460
 I/O 地址, 237, 297
 I/O 域, 969
 布局, 1059
 应用, 1058
 数据格式, 1059
 模式, 1059
 隐藏输入, 1059
 I/O 访问错误, 296
 I/O 输入, 293
 I/O 输出, 293
 IEC 定时器
 复位, 26, 27
 IEC 时间, 351
 IEC 检查
 功能, 360
 设置, 361
 说明, 21
 IN_RANGE, 658, 786
 INC, 672, 799
 IncreaseFocusedValue, 1252
 IncreaseTag, 1252
 INT, 348
 INV, 725, 854
 InverseLinearScaling, 1254
 InvertBit, 1254
 IO 域, 1374
 IO 域, 1375
 布局, 1374
 应用, 1374
 操作, 1374
 特性, 1374
 触摸操作, 1375
 输入域之间的变更, 1375
 IP 地址
 分配, 272
 ISO-on-TCP
 TSAP, 281
 特征, 277

J

JMP, 717, 845

JMPN, 718, 846

L

LABEL, 719, 847
 LAD, 409
 LAD 元素
 删除, 425
 剪切, 425
 复制, 425
 插入, 417, 418, 425
 插入和编辑操作数, 428
 插入操作的规则, 416
 替换, 425
 LAD 程序段
 交叉点, 434
 关闭分支, 433
 分支, 432
 删除交叉点, 436
 并行分支的规则, 432
 插入, 420, 450
 插入交叉点, 435
 插入分支, 433
 插入块调用, 421, 451
 插入所提供的全局库中的元素, 422, 452
 断开分支, 434
 禁止的互连, 417
 程序状态, 526
 重新安排交叉点, 435
 LED ACK, 1398
 LEFT, 881
 LEN, 880
 LIMIT, 678, 806
 LinearScaling, 1262
 LN, 683, 811
 LogOff, 1244
 Logoff, 1244
 Logon, 1248
 LREAL, 22, 25

M

MAX, 677, 805
 MIB, 1302
 MIN, 675, 803
 MOD, 669, 797
 MOVE, 696, 824
 MOVE_BLK, 697, 826
 MUL, 564, 573, 666, 794
 MUX, 731, 860

N

N, 630, 632, 758, 761
N_TRIG, 634, 763
NEG, 670, 798
NORM_X, 714, 843
NOT, 619, 750
NOT OK, 662, 790

O

OB 1, 313
OB 80, 20, 319
OB 82, 318
OK, 661, 789
OLE 对象
 从文件创建, 972
 存储在图像浏览器中, 991
 重新创建, 972
OR, 722, 851
 先或后与逻辑运算, 745
 逻辑运算, 568, 743, 851
OUT_RANGE, 660, 787

P

P, 560, 569, 628, 631, 757, 760
P_TRIG, 633, 762
PageDown, 1276
PageUp, 1276
Password
 层级, 1204
PG/PC 接口
 在线连接, 1428
PID_Compact, 37
PLC
 传送配方数据记录, 1412
 变量, 1096
 读取配方数据记录, 1412
PLC 变量
 PLC 变量表, 364
 PLC 变量表的结构, 363
 保持特性, 369
 删除, 372, 408
 可用的地址和数据类型, 365
 声明, 366, 367
 复制, 372
 定义, 337
 对行排序, 374
 属性, 370, 371
 插入表格行, 373, 408, 481
 更改, 368
 监视, 371

规则, 365, 366
设置保持特性, 369
重新链接, 368
PLC 报警, 1108
 系统定义的, 1108
 组态, 1117
PLC 编程
 登陆选项卡, 65
PORT_CFG, 907
Portal 视图, 64, 104
ProSave, 1347
 安装, 54
 语言切换, 1347

Q

Q 地址, 237

R

R, 559, 560, 568, 569, 622, 751
RCV_CFG, 47, 911
RCV_PTP, 917
RD_SYS_T, 871
RE_TRIGR, 886
REAL, 350
 说明, 21
RecipeViewBack, 1272
RecipeViewDeleteDataRecord, 1267
RecipeViewGetDataRecordFromPLC, 1267
RecipeViewMenu, 1268
RecipeViewNewDataRecord, 1266
RecipeViewOpen, 1269
RecipeViewRenameDataRecord, 1271
RecipeViewSaveAsDataRecord, 1270
RecipeViewSaveDataRecord, 1270
RecipeViewSetDataRecordToPLC, 1269
RESET_BF, 625, 754
ResetBit, 1273
RET, 719, 848
RezeptanzeigeZeigeHilfetext, 1271
RIGHT, 882
ROL, 740, 868
ROR, 738, 866
ROUND, 706, 835
RS, 627, 756
RS-232/PPI 多主站编程电缆, 249
RT, 26, 27
RUN, 292

S

S, 559, 560, 568, 569, 623, 752

- S_CONV, 874
 - S7 CPU
 - 工作存储器, 293
 - 操作数区域, 293
 - 装载存储器, 293
 - SCALE_X, 712, 841
 - ScreenObjectCursorDown, 1250
 - ScreenObjectCursorUp, 1249
 - ScreenObjectPageDown, 1251
 - ScreenObjectPageUp, 1250
 - SD 卡
 - 添加读卡器, 151
 - 简介, 150
 - 访问, 152
 - 选择卡类型, 152
 - SEL, 730, 858
 - SEND_CFG, 909
 - SEND_PTP, 916
 - SET_BF, 624, 753
 - SetBit, 1277
 - SetBitWhileKeyPressed, 1277
 - SetConnectionMode, 1279
 - SetDeviceMode, 1276
 - SetLanguage, 1278
 - SetTag, 1279
 - SFB100, 304
 - SFB101, 304
 - SFB102, 304
 - SFB103, 304
 - SFB110, 304
 - SFB111, 304
 - SFB112, 304
 - SFB113, 304
 - SFB114, 304
 - SFB115, 304
 - SFB116, 304
 - SFB117, 304
 - SFB122, 304
 - SFC151, 304
 - SFC154, 304
 - SFC156, 304
 - SFC157, 304
 - SFC158, 304
 - SFC159, 304
 - SFC192, 304
 - SFC193, 304
 - SFC198, 304
 - SFC199, 304
 - SFC200, 304
 - SFC201, 304
 - SFC202, 304
 - SFC204, 304
 - SFC205, 304
 - SFC206, 304
 - SFC208, 304
 - SFC210, 304
 - SFC211, 304
 - SFC212, 304
 - SFC214, 304
 - SFC215, 304
 - SFC216, 304
 - SFC218, 304
 - SFC32, 304
 - SFC33, 304
 - SFC41, 304
 - SFC42, 304
 - SFC43, 304
 - SFC46, 304
 - SGN_GET, 919
 - SGN_SET, 920
 - SHL, 736, 864
 - ShowAlarmWindow, 1286
 - ShowLogonDialog, 1285
 - ShowOperatorNotes, 1285
 - SHR, 734, 862
 - SIMATIC PLC, 71
 - SIMATIC S7
 - 优化组态, 1308
 - SimulateTag, 1281
 - SIN, 685, 813
 - SINT, 347
 - SmartClient 显示, 969
 - SNMP, 1302
 - SQR, 680, 808
 - SQRT, 681, 810
 - SR, 626, 755
 - SRT_DINT, 924
 - Start.exe, 15
 - STARTUP
 - STARTUP 操作, 290
 - 功能, 288
 - 暖启动, 289
 - STOP, 292
 - StopRuntime, 1282
 - STP, 887
 - STRG_VAL, 876
 - STRING, 353
 - STRUCT
 - 全局数据块中的声明, 476
 - 块接口中的声明, 404
 - 数据访问, 430, 459
 - 用作参数, 430, 459
 - 结构, 355
 - SUB, 665, 792
 - SWAP, 703, 832
- T
- T_ADD, 872
 - T_CONV, 872
 - T_DIFF, 873
 - TAN, 688, 816

TCON, 898
TCON_Param, 278
TCP
 特征, 276
 端口号, 280
TDISCON, 900
TIA Portal, 61
 启动, 101
 退出, 101
TIME, 351
TOF, 638, 767
TON, 637, 765
TONR, 565, 575, 640, 768
TP, 635, 764
TraceUserChange, 1282
TRCV, 904
TRCV_C, 894
TrendViewBackToBeginning, 1260
TrendViewCompress, 1257
TrendViewExtend, 1256
TrendViewRulerBackward, 1258
TrendViewRulerForward, 1257
TrendViewScrollBack, 1256
TrendViewScrollForward, 1255
TrendViewSetRulerMode, 1259
TrendViewStartStop, 1259
TRUNC, 711, 839
TSAP
 ASCII 代码表, 283
 结构, 281
TSAP 分配
 实例, 283
TSEND, 902
TSEND_C, 891

U

UART 数据传输, 245
UDINT, 350
UDT1001, 304
UDT65, 304
UDT651, 304
UDT652, 304
UDT653, 304
UDT654, 304
UFILL_BLK, 702, 830
UINT, 349
UMOVE_BLK, 699, 827
UpdateTag, 1248
USINT, 348

V

VAL_STRG, 878

VARIANT, 357
VOID, 357

W

WinCC flexible 运行系统
 关闭, 1357
WinCC flexible 项目
 移植, 949
WORD, 346
WR_SYS_T, 870

X

XE * MERGEFORMAT, 911
XOFF, 18
XON, 18
XOR, 724, 746, 852

—

一致
 对象大小, 981
一致性
 插槽规则, 189
一致性检查, 510, 515
 简介, 510, 515

下

下载

将块下载到可访问节点, 493
将块下载到存储卡, 494
设备中的块, 490, 491
配方, 1190
项目, 1352

不

不等于, 651, 779

与

与 PLC
 使用 PLC 的 HMI 设备, 1329

专

专有技术保护, 496
禁用, 497
设置, 497

东

东方字符
HMI 设备上的输入, 1362

丢

丢失字体, 48

中

中断驱动的程序执行, 314

乘

乘, 564, 573, 666, 794

事

事件

值改变, 1294
切换, 1294
单击, 1291
取消激活, 1289
变量, 1097
启用, 1288
在对话框关闭时, 1289
在对话框打开时, 1288
巡视窗口, 51
已清除, 1287
报警回路, 1291
报警缓冲区溢出, 1292
按下, 1290
按下按键, 1293
改变用户, 1289
用于函数列表, 1097
画面切换, 1289
确认, 1292
离开, 1290
组态, 1182
调整, 1288
运行系统停止, 1293
进入, 1290
释放, 1292
限定值, 1291

限定值, 1291

事件驱动的任务

组态:事件驱动的任务, 1121

二

二进制值

取反, 748

二进制输入

插入, 747

亚

亚洲字符

HMI 设备上的输入, 1362
存储空间要求, 1361
翻译, 1362

亚洲语言

字号, 1362
文本域长度, 1362
配置, 1362

交

交叉引用

使用, 519, 1416
巡视窗口, 51
显示, 521, 1417
简介, 519, 1416

交叉引用列表, 520, 1417

对列排序, 520, 1417
显示, 521, 1417
概述, 521, 1417
结构, 520, 1417
视图, 520, 1417
设置, 520, 1417

交叉引用列表概述, 521, 1417

交叉引用列表的视图, 520, 1417

交叉点

删除, 436
定义, 434
插入, 435
重新安排, 435

产

产品支持, 186

启用, 187

从

- 从属性结构, 511
 - 布局, 512
 - 显示, 513
 - 符号的含义, 512
 - 简介, 511
 - 设置视图选项, 514

以

- 以太网
 - 组态设备, 272
- 以太网接口
 - 显示参数, 1425

任

- 任务, 1295
 - 取消激活, 1297
- 任务卡, 67, 955
 - 功能, 113
 - 在线工具, 262
 - 硬件目录, 184

仿

- 仿真, 56
 - PLC 连接, 56
 - 运行系统, 1341

休

- 休眠, 14

优

- 优先等级, 312
- 优先级
 - 局部错误处理, 470
- 优化组态
 - SIMATIC S7, 1308

传

- 传输数据
 - “项目 ID”区域指针, 1318
 - 不需要同步, 1322
 - 作业信箱, 1325

- 作业信箱区域指针, 1319
- 协调区域指针, 1317
- 操作员在配方视图中输入, 1324
- 数据记录是指针, 1321
- 由组态的函数触发, 1326
- 需要同步, 1322

传送

- 移动值, 696
- 配方数据记录, 1412
- 错误消息, 1350
- 传送数据
 - 区域指针, 1312
 - 区域指针画面号, 1314
 - 可能的出错原因, 1327
 - 日期/时间区域指针, 1315, 1316
- 传送项目
 - HMI 设备, 56, 1328
 - 配方数据记录, 56

位

位号

- 读取, 728, 857

位域

- 复位, 625, 754
- 设置, 624, 753

位存储器

- 启用保持性位存储器的显示, 505

位置

- 对象, 972
- 对象的, 975
- 编辑多个对象, 987

位逻辑指令

- AND 逻辑运算, 568, 569, 575
- OR 逻辑运算, 568
- 复位输出, 560, 568, 569
- 常开触点, 560, 562, 565
- 常闭触点, 560, 562, 565
- 查询操作数的信号上升沿, 560, 569
- 置位输出, 560, 568, 569
- 赋值, 570, 575
- 输出线圈, 562, 565

位逻辑运算

- AND 逻辑运算, 742
- OR 逻辑运算, 743
- 先与后或逻辑运算, 745
- 先或后与逻辑运算, 745
- 取反二进制值, 748
- 取反线圈, 621
- 取反逻辑运算结果, 619
- 在信号上升沿置位操作数, 760
- 在信号上升沿置位输出, 633, 762
- 在信号下降沿置位操作数, 631, 632, 761
- 在信号下降沿置位输出, 634, 763

- 复位位域, 625
- 复位置位触发器, 627, 755, 756
- 复位输出, 622, 751
- 常开触点, 617
- 常闭触点, 618
- 异或逻辑运算, 746
- 扫描操作数的信号上升沿, 628, 757
- 扫描操作数的信号下降沿, 630, 758
- 插入二进制输入, 747
- 置位位域, 624, 753, 754
- 置位复位触发器, 626
- 置位输出, 623, 752
- 赋值, 749
- 赋值取反, 750
- 输出线圈, 620
- 重置 IEC 定时器, 26, 27

余

- 余弦
 - 计算余弦值, 686, 815

作

- 作业信箱
 - 传输数据, 1325

使

- 使用, 1174
 - 关于配方, 1174
 - 画面浏览功能键, 1036
 - 系统函数, 1233
- 使用变量, 1058
- 使用实例项目, 16
- 使用键盘
 - 程序编辑器, 396

保

- 保存
 - 全局库, 1052
 - 外部图像, 991
- 保护
 - 带密码的功能键, 1034
- 保护原理, 176
- 保护等级, 254
- 保持型接通延迟, 565, 575
- 保持性, 295
 - 说明, 20

- 保持性位存储器
 - 启用显示, 505
- 保持性存储器, 516
- 保持特性
 - PLC 变量, 369
 - 全局数据块, 478
 - 背景数据块, 479
 - 设置, 369, 478, 480

信

- 信号板, 201
 - 插入, 202
- 信号板, 移动, 21
- 信息文本, 1116
 - 创建, 1121, 1123
 - 显示, 1369
- 信息系统
 - 信息系统的组件, 123
 - 工具提示, 125
 - 弹出项, 125

修

- 修改
 - 高速计数器, 647, 775
- 修改数值
 - 可用的操作数, 536

值

- 值
 - 移动, 696
 - 移动操作, 824
 - 转换, 705
- 值改变, 1294

允

- 允许的字符
 - 用户视图, 55

元

- 元素列表, 1181

入

- 入门向导, 65

Portal 视图和项目视图, 64
创建项目, 69

全

全局分配
 功能键, 52, 1028, 1030
全局变量, 363
全局库, 422, 452, 1044
 使用元素, 172
 使用元素和部件视图, 164
 使用所提供的全局库, 175
 使用过滤视图, 168, 175
 保存, 170, 1052
 共享使用, 169
 关闭, 171
 创建, 169
 创建文件夹, 171
 删除, 171
 删除元素, 174
 基本知识, 169
 打开, 170, 1053
 添加元素, 172
 编辑元素, 173
 说明, 21
全局数据块, 321
 保持特性, 478
 符号访问, 478
 设置保持特性, 478
全局软键, 1030

关

关断延迟, 638, 767
关键字, 338
关闭
 运行系统, 1357
关闭 Windows XP, 14

减

减, 665, 792

函

函数列表, 1097, 1233, 1297
 可用于特定 HMI 设备, 1234
 同步完成, 1238
 属性, 1234
 异步完成, 1238
 执行, 1235
 状态信息, 1235

系统函数, 1235
编辑, 1237
脚本, 1235
运行时完成, 1238

分

分布
 对象均匀, 980
分支
 关闭, 433
 定义, 432
 插入, 433, 463
 断开, 434, 463
分配
 功能键, 1028, 1030, 1031
 功能键的函数, 1032
 图形到功能键, 1035
 层的对象, 1039
 授权, 1207, 1227
 用户组, 1209, 1229
分配 IP 地址, 19
分配列表
 删除过滤器, 503
 启用保持性位存储器的显示, 505
 定义过滤器, 503
 布局, 499
 显示, 501
 显示位存储器的实例, 499
 显示输入和输出的实例, 499
 符号的含义, 500
 简介, 498
 设置视图选项, 502
 过滤, 504
 过滤器选项, 502
分配参数
 硬件, 188
分配符号, 238

切

切换, 1294
 运行系统语言之间, 1361
切换工作模式, 270
切换时间, 250

列

列
 报警视图, 1068

创

- 创建, 1187
 - HMI 设备上的配方数据记录, 1195, 1196
 - 信息文本, 1121, 1123
 - 内部变量, 1085
 - 外部变量, 1084
 - 报警组, 1119
 - 授权, 1206, 1225
 - 数组, 1101
 - 文件夹链接, 990
 - 模板, 966
 - 用户, 1208, 1229, 1388, 1390
 - 用户组, 1206, 1227
 - 画面, 960
 - 组, 994
 - 配方, 1187
 - 配方数据记录, 1409
- 创建功能 (FC), 375
- 创建打印预览, 145
- 创建监视表格, 528
- 创建项目, 69

初

- 初始值
 - 更改, 405
 - 查看初始值, 405

删

- 删除, 15
 - 分配列表中的过滤器, 503
 - 动画, 1022
 - 变量, 1088
 - 对象, 975
 - 授权, 1212
 - 文件夹链接, 990
 - 模板, 967
 - 用户, 1210, 1393
 - 用户组, 1211
 - 画面, 961
 - 硬件组件, 198
 - 组中的对象, 996
 - 运行系统中的用户, 1218
 - 配方数据记录, 1411

前

- 前景色
 - 动态化, 1022

剪

- 剪切
 - 动画, 1022

功

- 功能
 - 功能键的分配, 1032
 - 更新变量值, 1082
- 功能 (FC), 320
- 功能块 (FB)
 - 创建, 375
 - 初始值, 320
 - 定义, 320
 - 背景数据块, 320
 - 输出参数, 327
- 功能块图, 440
- 功能范围
 - ProSave, 1347
- 功能键, 52, 1028
 - 全局分配, 52, 1030
 - 分配函数, 1032
 - 分配图形, 1035
 - 局部分配, 52, 1031
 - 用于画面浏览, 1036
 - 用密码保护, 1034

加

- 加, 564, 573, 663, 791

动

- 动态化, 1019
 - 对象属性, 1019
 - 对象的外观, 1022
 - 对象的移动, 1024
 - 对象的颜色, 1022
 - 闪烁, 1022
- 动画, 1020
 - 删除, 1022
 - 剪切, 1022
 - 多项选择, 1027
 - 插入, 1022
 - 用于对象组, 1027
 - 组态, 1021

助

助记符
更改, 22

区

区域指针, 1312
作业信箱, 1319
协调, 1317
数据记录, 1178, 1321
日期/时间, 54, 1315, 1316
日期/时间 PLC, 54
画面号, 1314
连接编辑器, 1309
项目 ID, 1318

半

半径 X : , 1060
半径 Y, 1060

单

单个背景
定义, 324
实例, 324
更正调用类型, 424, 454
单击, 1291

参

参数
分配, 327
输出参数, 327
适用于 CPU, 232
隐藏参数, 431, 461
参数传递, 326, 328
规则, 328
参数类型
VARIANT, 357
VOID, 357
概述, 356
简介, 344
参考对象
定义, 979
参考点开关, 38

双

双字, 346

反

反余弦
计算反余弦值, 690, 819
计算反余弦值, 690, 819
反正切
计算反正切值, 692, 820
反正弦
计算反正弦值, 689, 817
反码
创建, 725, 854

取

取幂, 694, 823
取整
取整数字值, 706, 835
取消
组, 995
取消激活, 1289
取消编组
组, 995

变

变量

PLC 变量和局部变量, 337
事件, 1097
保持特性, 369
内部变量, 1080
创建内部变量, 1085
创建外部变量, 1084
删除, 1088
名称, 1084, 1085
声明 ARRAY, 403
声明 STRUCT, 404
复制, 1088
外部变量, 1081
对象列表, 1085
寻址, 1084
指针化, 1096
改变 PLC, 1096
数据类型, 1085
显示值, 1104
更新, 1093, 1098
最大长度, 55
概述, 336
注释, 1085

- 监视全部, 542
- 移植, 951
- 立即监视一次, 542
- 索引变量, 1096, 1097
- 线性标定, 1095
- 线性转换, 1094
- 组态, 1088
- 组态多个变量, 1089
- 至 PLC 的连接, 1084
- 设置保持特性, 369
- 起始值, 1092, 1093
- 运行系统中, 1093
- 连续读, 1094
- 采集周期, 1093, 1098, 1103
- 采集模式, 1093
- 重命名, 1088
- 长度, 1085
- 间接寻址, 1096, 1097
- 限制值, 1091
- 限制范围, 1091
- 变量值
 - 输出, 1104
- 变量列表
 - 间接寻址, 1096, 1097
- 变量声明
 - 保留关键字, 338
 - 删除变量, 372, 408, 481
 - 变量声明的用途, 399
 - 在末尾插入表行, 408, 482
 - 块接口, 398
 - 声明变量, 401
 - 对行排序, 374
 - 插入表格行, 373, 408, 481
 - 更改初始值, 405
 - 更改数据类型, 405
 - 自动填充单元格, 373

可

- 可用于特定设备
 - 画面的, 955
- 可能的出错原因
 - 传送数据, 1327
- 可视化 (Visualization)
 - 登陆选项, 65

同

- 同步
 - 与 PLC, 1178
 - 传输数据, 1322

名

- 名称
 - 数组元素, 52

启

- 启动
 - 组织块, 313
- 启动 OB
 - 描述, 313
- 启动信息, 312
- 启动参数, 291
- 启动在线和诊断视图, 260
- 启动程序, 313
- 启动设置, 102
- 启用, 1288

周

- 周期负载, 235

回

- 回读连接参数功能, 281

固

- 固定窗口, 964

图

- 图像文件
 - 存储在图像浏览器中, 991
- 图像格式, 991
- 图形
 - 伸展, 1061
 - 使用来自图形浏览器的, 988
 - 具有透明背景, 989
 - 功能键的分配, 1035
 - 图形视图, 1061
 - 按钮, 1075
 - 插入, 973
 - 管理, 989
- 图形 I/O 域, 969
 - 布局, 1062
 - 模式, 1063
- 图形 IO 域, 1377
 - 应用, 1377
 - 操作, 1377

- 特性, 1378
- 图形列表
 - 基本原理, 1012
 - 输出组态数据, 1018
- 图形对象, 969
- 图形浏览器, 991
 - 使用图形, 988
- 图形视图, 969, 1376
 - 伸展图形, 1061
 - 图形, 1061
 - 布局, 1061, 1376
 - 应用, 1061, 1376
 - 操作, 1377
 - 透明色, 1061
- 圆
 - 圆, 969
 - 圆角, 1064
 - 布局, 1064
 - 应用, 1063
 - 圆角, 1064
- 在
 - 在基本模式和扩展模式之间切换, 532
 - 在对话框关闭时, 1289
 - 在对话框打开时, 1288
 - 在线功能, 95
 - 在线和诊断
 - 登陆选项卡, 65
 - 在线和诊断视图, 256
 - 语言, 19
 - 输入和输出, 19
 - 在线工具, 257, 262
 - 在线显示
 - 橙色, 260
 - 在线模式, 95, 1421
 - 取消在线连接, 1428
 - 将所有可访问设备显示为列表, 1422
 - 显示接口上的可访问设备, 1422
 - 转到在线, 1427
 - 在线访问, 1422
 - 在线运行, 18
 - 在线连接, 1421
 - 在设备视图中联网
 - 步骤, 208
 - 联网选项, 208
 - 在运行系统中创建
 - 用户, 1216
 - 在运行系统中的作用
 - 运行系统中的用户管理设置, 1204

地

- 地址
 - 分配, 238
- 地址范围
 - 更改, 238

块

块

- 一致性检查, 510, 515
- 下载到设备, 488, 490
- 从存储卡上传块, 495
- 从设备上传, 488, 494
- 使用库, 377
- 保存, 386
- 关闭, 386
- 在程序编辑器中编译, 487
- 在线删除, 387
- 在项目树中编译, 487
- 填充, 700, 829
- 填充不可中断, 702, 830
- 复制, 378, 379
- 将块下载到可访问节点, 493
- 将块下载到存储卡, 494
- 属性, 381
- 插入, 378, 379
- 显示属性, 383
- 比较, 138, 482
- 激活符号访问, 336
- 离线删除, 387
- 移动, 697
- 移动不可中断, 699, 827
- 移动操作, 826
- 符号声明, 335
- 符号访问, 335
- 类型, 311
- 编辑属性, 383
- 调用结构中的一致性检查, 510
- 输入标题, 380
- 输入注释, 380
- 重命名, 386
- 块一致性
 - 在从属性结构中检查, 515
 - 检查, 510
- 块参数, 381
 - 块接口, 398
 - 基本知识, 384
- 块属性
 - 功能, 381
 - 显示, 383
 - 概述, 381
 - 编辑, 383

块接口

- 删除变量, 372, 408, 481
- 功能, 390
- 变量声明的用途, 399
- 声明变量, 401
- 定义, 398
- 对行排序, 374
- 插入行, 373, 408, 481
- 显示, 395
- 更改初始值, 405
- 更改数据类型, 405
- 自动填充单元格, 373
- 隐藏, 395

块文件夹, 20, 374

块比较

- 同步比较动作, 140
- 同步滚动, 485
- 基本知识, 482
- 导航, 484
- 指定动作, 140
- 显示和隐藏相同对象, 138
- 更改块, 485
- 更新比较结果, 139
- 比较编辑器, 136
- 详细比较, 483
- 详细比较的表示法, 483
- 运行详细比较, 139
- 进行, 138

块注释

- 显示, 395
- 隐藏, 395

块调用

- 以单个背景或多重背景的方式调用, 323
- 单个背景, 324
- 基本知识, 322
- 多重背景, 325
- 嵌套深度, 323
- 插入, 421, 451
- 更新, 486
- 更正调用类型, 424, 454
- 顺序, 323

基

- 基本数据类型, 344
- 基本模式, 531

填

填充

- 块, 700, 829
- 块不中断, 702
- 块不可中断, 830

声

声明

- STRUCT, 476
- 变量的符号声明, 335
- 声明 ARRAY, 403, 475
- 声明 STRUCT, 404

备

备份, 1387

- HMI 设备数据, 1347, 1348, 1358

复

复位

- IEC 定时器, 26, 27
- 位域, 625, 754
- 输出, 568, 622

复位为出厂设置

- 操作系统, 1349

复位输出, 559

复制

- 值, 696
- 变量, 1088
- 块, 697, 826
- 块不中断, 699
- 报警指示灯, 50
- 报警窗口, 50
- 报警视图, 50
- 模板, 967
- 画面, 961
- 硬件组件, 199

复杂数据类型

- 概述, 352

外

外观

- 对象动态化, 1022

外部图像文件

- 创建文件夹链接, 989
- 删除文件夹链接, 989
- 存储在图像浏览器中, 991
- 管理, 972
- 编辑文件夹链接, 989
- 重命名文件夹链接, 989

多

- 多路复用, 731, 860

多边形, 969
 多重背景
 定义, 325
 更正调用类型, 424, 454
 说明, 21
 多项选择, 192, 978

大

大于, 655, 783
 大于或等于, 570, 652, 780

字

字体
 丢失字体, 48
 字号
 亚洲语言, 1362
 字符, 351
 字符串, 353
 字节
 交换, 703, 832
 字逻辑运算
 AND 逻辑运算, 721, 849
 OR 逻辑运算, 722, 851
 创建反码, 725, 854
 多路复用, 731, 860
 异或逻辑运算, 724, 852
 编码, 728, 857
 解码, 727, 855
 选择, 858

存

存储
 对象在库中, 1049
 存储位置, 103
 存储区
 保持性存储区, 295
 工作存储器, 293
 装载存储器, 293
 存储卡
 显示属性, 153
 添加读卡器, 151
 简介, 150
 访问, 152
 选择卡类型, 152
 存储器位 (M), 293
 存储器双字 (MD), 293
 存储器复位
 基础知识, 292
 执行, 271

存储器字 (MW), 293
 存储器字节 (MB), 293

安

安全指南, 129
 安全系统, 1386
 安全须知
 切换信息文本, 1370
 后台中的配方数据记录, 1181
 安装, 15
 ProSave, 54

定

定义
 参考对象, 979
 棒图分段, 1056
 用于分配列表, 503
 定义热键, 1075
 定时器指令
 保持型接通延迟, 565, 575
 定时器操作
 关断延迟, 638, 767
 接通延迟, 637, 765
 时间累加器, 640, 768
 生成脉冲, 635, 764

实

实例
 FBD 指令, 567
 使用位逻辑指令控制传送带, 559, 568
 使用位逻辑指令检测传送带方向, 560, 569
 使用定时器和位逻辑指令控制室温, 565, 575
 使用数学函数计算方程式, 564, 573
 使用计数器和比较指令检测存储区域的填充量, 562, 570
 对于监视表格中的修改值条目, 536
 对于监视表格中的强制值条目, 537
 报警类别的应用, 1111
 改变操作模式和显示操作模式, 1238
 梯形逻辑指令, 558
 用于显示分配列表中的位存储器, 499
 用于显示分配列表中的输入和输出, 499
 系统报警, 1109
 实例:
 离散量报警, 1109
 实例:模拟量报警, 1109
 实时时钟, 271

密

- 密码, 496, 1386
 - 为功能键, 1034
 - 保护块, 496
 - 在运行系统中更改, 1217
 - 备份, 1387
 - 密码复杂性, 1205
 - 恢复, 1387
 - 更改, 497, 1210
- 密码列表, 1386

对

- 对数
 - 计算自然对数, 683, 811
- 对象
 - Tab 顺序, 972
 - 从文件创建 OLE 对象, 972
 - 分配给层, 1039
 - 创建新 OLE 对象, 972
 - 删除, 975
 - 动态移动, 1024
 - 均匀分布, 980
 - 外观动态化, 1022
 - 存储在库中, 1049
 - 对齐, 980
 - 排列, 972, 982
 - 插入, 972, 973
 - 插入多次, 986
 - 插入相同类型, 972
 - 放大, 986
 - 旋转, 972, 984
 - 编组, 994
 - 编辑组中的对象, 998
 - 翻转, 972
 - 自定义默认属性, 972
 - 访问保护, 1221, 1222
 - 调整大小, 972, 977
 - 选择多个对象, 972, 978
 - 重新定位多个对象并调整其大小, 972, 987
 - 重新定位对象, 972, 975
- 对象列表
 - 变量, 1085
- 对象大小
 - 一致, 981
- 对象属性
 - 动态化, 1019
- 对象组
 - 删除对象, 996
 - 编辑组中的对象, 998
- 对齐
 - 对象对齐, 980

寻

- 寻址, 297
 - 常规, 237
 - 指针化, 1096
 - 更改, 238
 - 符号, 1084
 - 间接寻址变量, 1096, 1097

导

- 导入
 - 用户数据, 1213
 - 配方, 1332
 - 配方数据的结构, 1333
- 导入配方
 - 导入结构, 1333
- 导出
 - 用户数据, 1213
 - 配方, 1330

将

- 将数据下载到 PLC
 - 错误消息, 55
- 将项目数据装载到设备, 93

小

- 小于, 657, 785
- 小于或等于, 570, 654, 782
- 小数位, 1367, 1369
 - 提取, 693, 821

尺

- 尺寸
 - 对象, 972
 - 编辑多个对象, 987

局

- 局部分配
 - 功能键, 52, 1028, 1031
- 局部变量
 - 定义, 337
 - 属性, 407
- 局部数据
 - 块接口, 398

局部数据位 (L), 293
局部数据双字 (LD), 293
局部数据字 (LW), 293
局部数据字节 (LB), 293
局部错误处理, 470
 GetError, 887
 GetErrorID, 890
 优先级, 470
 指令, 470
 错误信息, 470

层

层

将对象分配给层, 1039
重命名, 1043

屏

屏幕保护程序, 14
屏幕键盘, 123
 KTP400 Basic, 1365
 KTP600、KTP1000、TP1500 Basic, 1367
 字母数字, 1366, 1368
 数字, 1367, 1369
 语言行为, 55
 键布局, 1366, 1368

属

属性
 函数列表, 1234
属性 (CPU), 232

巡

巡视窗口, 68
 事件, 51
 交叉引用, 51
 布局, 111
 硬件和网络编辑器, 183
 诊断选项卡, 257

工

工作区, 67
 功能, 110
 调度程序, 1296
 调整, 117
工作存储器, 293, 516

工作模式
 RUN, 292
 STARTUP, 288
 STOP, 292
 简介, 287
 转换, 288
工作步骤
 创建画面, 959
工具栏, 389
 顺序, 972, 982
工程系统
 性能特点, 1413
工艺对象, 84

已

已清除, 1287
已组态
 报警循环, 1125

布

布局

I/O 域, 1059
IO 域, 1374
图形 I/O 域, 1062
图形视图, 1061, 1376
圆, 1064
开关, 1073, 1383
报警指示器, 1071, 1142
报警视图, 1067, 1068, 1069, 1140, 1397
按钮, 1074, 1382
文本域, 1077
日期/时间域, 1058, 1372
棒图, 1055
椭圆, 1060
用户视图, 1057
直线, 1066
矩形, 1072
符号 I/O 域, 1076
简单用户视图, 1388
简单配方视图, 1192, 1406
调用结构, 507
趋势视图, 1064, 1379
配方视图, 1072

帮

帮助

全文搜索, 127
导航, 127
帮助主题的标识, 69, 124

打印信息, 128
 打开, 126
 搜索, 127
 帮助指示器, 969

常

常开触点, 559, 560, 562, 565, 617
 常见问题解答, 186
 常规插槽规则, 189
 常量
 PLC 变量表的结构, 363
 基本知识, 343
 常闭触点, 560, 562, 565, 618

平

平方
 计算, 680, 808
 平方根
 计算, 681, 810

库

库, 79, 1044
 任务卡, 162
 使用元素和部件视图, 164
 使用过滤视图, 168, 175
 保存, 1052
 全局库, 169
 基本知识, 161
 存储对象, 1049
 打开, 1053
 添加和使用块, 377
 版本控制, 1052
 类型, 161
 项目库, 165
 库对象, 1044, 1045
 插入, 1050

应

应用
 I/O 域, 1058
 IO 域, 1374
 图形 I/O 域, 1062
 图形 IO 域, 1377
 图形视图, 1061, 1376
 圆, 1063
 开关, 1073, 1383
 报警指示器, 1141
 按钮, 1074, 1382

文本域, 1077
 日期/时间域, 1057, 1372
 棒图, 1055
 椭圆, 1060
 用户视图, 1056
 直线, 1066
 矩形, 1071
 符号 I/O 域, 1075
 符号 IO 域, 1384
 简单报警窗口, 1140, 1396
 简单报警视图, 1140, 1396
 简单用户视图, 1387
 趋势视图, 1379
 配方视图, 1072
 应用领域
 用户管理, 1198
 调度程序, 1294

延

延时中断
 取消, 925
 延时中断 OB, 315
 延迟时间
 设置, 1123

开

开关, 969, 1383
 布局, 1073, 1383
 应用, 1383
 操作, 1383
 特性, 1383
 类型, 1074
 开始卸载, 60
 开始安装, 58
 开始登陆选项, 65
 开放式用户通信
 TCON_Param, 278
 TSAP, 281
 使用的协议, 276
 创建连接, 218
 删除互连, 220
 启动连接参数分配, 218
 回读功能, 281
 常规, 213
 指令, 213
 更改参数值, 280
 端口号, 280
 连接参数, 216
 连接参数分配, 214
 连接建立, 214
 连接描述, 277

异

异或

- 逻辑运算, 852

异步

- 传输数据, 1322

强

强制值

- 可用的操作数, 537

待

- 待机模式, 14

循

循环

- 操作, 1093

- 连续, 1093

循环 OB

- 描述, 313

循环中断 OB

- 分配参数, 385

- 描述, 316

循环时间, 20, 234

- 显示测量的, 264

- 显示组态的, 263

循环监视时间, 违反, 20

循环移位

- 循环右移, 738, 866

- 循环左移, 740, 868

循环程序执行

- 中断选项, 313

- 编程, 313

性

性能特点

- HMI 设备, 1414

- 工程系统, 1413

恢

恢复, 1387

- HMI 设备数据, 1347, 1348, 1358

恢复数据

- HMI 设备, 1347, 1348, 1358

手

- 手册, 186

打

打印

- 打印网络视图, 182

- 打印范围, 182

- 打印设备视图, 182

- 打印输出的形式, 182

- 打印项目, 146

打开

- 全局库, 1053

扩

扩展指令, 87

- ATTACH, 922

- CAN_DINT, 925

- CTRL_PWM, 942

- DETACH, 923

- DIS_AIRT, 925

- EN_AIRT, 926

- LEFT, 881

- LEN, 880

- PORT_CFG, 907

- RCV_PTP, 917

- RIGHT, 882

- S_CONV, 874

- SEND_CFG, 909

- SEND_PTP, 916

- SGN_GET, 919

- SGN_SET, 920

- SRT_DINT, 924

- STP, 887

- STRG_VAL, 876

- TCON, 898

- TDISCON, 900

- TRCV, 904

- TRCV_C, 894

- TSEND, 902

- TSEND_C, 891

- VAL_STRG, 878

- 使用 RE_TRIGR 进行暖启动, 886

- 扩展模式, 531

折

- 折线, 969

报

报告

- 登录, 1223

报警

- 事件, 1145
- 巡视窗口, 51
- 显示, 1067
- 确认, 1400
- 系统函数, 1144
- 组态, 1121, 1127
- 组成部分, 1115
- 编辑, 1401
- 运行系统中, 1138, 1394

报警事件

- 确认, 1110
- 离开, 1110
- 进入, 1110

报警回路, 1291

报警循环

- 已组态, 1125

报警打印

- 组态:报警打印, 1129

报警指示器, 969, 1071, 1130, 1141, 1398

- 事件, 1071
- 布局, 1071, 1142
- 应用, 1141
- 报警类别, 1071
- 操作, 1142
- 组态, 1135
- 运行系统中, 1139, 1396

报警指示灯, 50

报警文本, 1116

- 格式, 1125
- 特殊字符, 53
- 输出域, 53

报警日志

- 运行系统中, 1395

报警状态, 1116

- 已确认, 1110
- 离开, 1110
- 进入, 1110

报警窗口, 50, 1130

- 组态, 1134
- 运行系统中, 1139, 1396

报警类别, 1111, 1116

- 使用, 1111
- 布局, 1140, 1397
- 自定义, 1112
- 运行系统中, 1139, 1394
- 项目范围, 1112
- 预定义, 1111

报警类型, 1107

报警系统, 1106

报警组, 1115, 1116

- 创建, 1119

- 组态, 1119

报警缓冲区

- 运行系统中, 1139, 1395

报警缓冲区溢出, 1292

报警编号, 1116

报警视图, 50, 969, 1067, 1068, 1130, 1141, 1398

- 列, 1068
- 布局, 1067, 1069, 1140, 1397
- 报警文本窗口, 1141, 1398
- 控制元素, 1067, 1070
- 操作, 1141, 1398
- 组态, 1132

指

指令窗口

- 功能, 394
- 扩大工作区, 394

指定

- 用于装载的设置, 1336

指数值

- 计算, 684, 812

指针

- VARIANT, 357

指针化, 1096

按

按下, 1290

按下按键, 1293

按钮, 969, 1065, 1382

- 图形, 1075
- 定义热键, 1075
- 布局, 1074, 1382
- 应用, 1074, 1382
- 操作, 1382
- 文本, 1075
- 特性, 1382
- 类型, 1075
- 组态, 1222, 1226

授

授权, 58

- 分配, 1207, 1227
- 创建, 1206
- 删除, 1212
- 更改名称, 1212
- 管理, 1211

排

排列

画面中的对象, 972, 982

接

接线测试, 527

接通延迟, 637, 765

控

控制元素

报警视图, 1067, 1070

简单报警视图, 1141

趋势视图, 1065

配方视图, 1072

提

提取

小数位, 693, 821

最大值, 677, 805

最小值, 675, 803

限值, 678, 806

插

插入

动画, 1022

对象, 986

库对象, 1050

插入图形, 973

插入对象, 972, 973

矩形, 1003

输入, 426, 456

输出, 426, 456

输出域, 1124

插入输入, 426, 456

插槽

分配, 192

机架, 191

选择, 192

插槽规则

常规, 189

握

握手, 248

搜

搜索

硬件目录, 190

操

操作

IO 域, 1374

图形 IO 域, 1377

图形视图, 1377

开关, 1383

报警指示器, 1142

按钮, 1382

日期/时间域, 1373

符号 IO 域, 1385

简单报警视图, 1141, 1397

简单用户视图, 1388

简单配方视图, 1193, 1406

趋势视图, 1379

操作员在配方视图输入

传输数据, 1324

操作员控件

简单报警视图, 1397

趋势视图, 1379

操作数

在信号上升沿置位, 631, 760

在信号下降沿置位, 632, 761

扫描操作数的信号上升沿, 628, 757

扫描操作数的信号下降沿, 630, 758

插入, 428, 429, 457, 458

编辑, 428, 457

操作数区域, 293

操作数选择项, 429, 458

操作特性

使用日时钟功能, 240

使用时钟存储器, 253

操作系统, 309

在 HMI 设备上更新, 1349

操作配方

传送数据记录, 1198

创建配方数据记录, 1195, 1196

读取配方数据记录, 1197

收

收藏夹

功能, 390

显示, 395

程序编辑器, 420, 421, 449, 450

隐藏, 395

改

改变操作模式和显示操作模式(实例), 1238
改变用户, 1289

故

故障设备, 257

数

数字量输入, 38

数学函数

乘, 564, 573, 666, 794
减, 665, 792
创建补码, 670, 798
加, 564, 573, 663, 791
取幂, 694, 823
提取小数, 693, 821
提取最大值, 677, 805
提取最小值, 675, 803
提取限值, 806
计算余弦值, 686, 815
计算反余弦值, 690, 819
计算反正切值, 692, 820
计算反正弦值, 689, 817
计算平方, 680, 808
计算平方根, 681, 810
计算指数值, 684, 812
计算正切值, 688, 816
计算正弦值, 685, 813
计算绝对值, 674, 802
计算自然对数, 683, 811
设置限值, 678
返回除法的余数, 669, 797
递减, 673, 800
递增, 672, 799
除, 564, 573, 668, 795

数据

发送, 891
接收, 904

数据位 (DBX), 293

数据信箱

对于配方, 1178

数据双字 (DBD), 293

数据块

全局数据块, 321, 472
创建, 376, 474
在线监视数据值, 472, 480
在线监视数据值, 472, 480
声明 ARRAY, 475
声明 STRUCT, 476
声明表, 472
编程, 472, 475, 477

背景数据块, 472

数据块声明

在末尾插入表行, 408, 482
插入行, 373, 408, 481

数据块编辑器

对行排序, 374

数据备份

HMI 设备, 1347, 1348, 1358

数据字 (DBW), 293

数据字节 (DBB), 293

数据对象, 344

数据流, 1405

数据流控制, 247

数据类型

ARRAY, 354
BOOL, 345
BYTE, 346
CHAR, 351
DINT, 349
DTL, 352
DWORD, 346
INT, 348
LREAL, 22, 25
REAL, 350
SINT, 347
STRING, 353
STRUCT, 355
TIME, 351
UDINT, 350
UINT, 349
USINT, 348
VARIANT, 357
VOID, 357
WORD, 346
关于转换的说明, 21
内部变量, 1080
参数类型, 356
基本, 344
复杂, 352
数据类型兼容性, 360
更改, 405
移植, 951
简介, 344
转换, 360

数据记录

传送, 1198
读取, 1197

数据记录列表, 1181

数组, 53, 354, 1099, 1101

创建, 1101

参见 ARRAY, 403, 475

参见 STRUCT, 476

间接寻址, 1096, 1099

数组元素, 1099

HMI 变量的使用位置, 52
名称, 52

数组变量, 1099
Char, 53

整

整型, 347, 348, 349, 350
16 位, 348, 349
32 位, 349, 350
8 位, 347, 348
创建, 711
截取, 839
浮点数向上取整, 708, 837
浮点数向下取整, 709, 838

文

文件夹链接

创建, 990
删除, 990
更新, 989
编辑, 990
重命名, 990

文本

按钮, 1075
文本域, 1077

文本列表

在配方数据记录中使用, 1180
用户自定义, 154
符号 I/O 域, 1076
简介, 154
系统定义的, 154

文本域, 969

尺寸, 1077
布局, 1077
应用, 1077
文本, 1077

文本域长度

亚洲语言, 1362

文档设置, 145

旋

旋转

对象, 972, 984

无

无人值守模式, 15
无需确认的单个报警, 1114

日

日时钟

在在线和诊断视图中设置日时钟, 271

日时钟功能

基本知识, 240
日时钟格式, 240
时钟参数, 241
设置日时钟, 240
读取日时钟, 240

日期/时间

区域指针, 1315

日期/时间域, 969

使用变量, 1058
布局, 1058, 1372
应用, 1057, 1372
操作, 1373
显示系统时间, 1058
格式, 1058
特性, 1373

时

时钟, 969

时钟存储器, 253
时间戳, 1116
时间累加器, 768
时间错误中断 OB, 18, 20, 319

显

显示

CPU 存储区, 518
交叉引用, 521, 1417
从属性结构, 513
信息文本, 1369
分配列表, 501
最大可用装载存储器, 519
棒图上的限制线, 1055
程序信息, 498
装载存储器, 519
调用结构, 508
显示 CPU 的存储器类型, 265
显示值
作为趋势, 1104
显示可访问设备, 95
显示在线状态, 1428
显示系统时间, 1058
显示通信接口, 17

暖

暖启动, 289

更

更改

字节的排列, 703, 832
 密码, 1210
 对象大小, 977
 用户, 1217, 1392
 用户组的显示名称, 1211
 运行系统中的密码, 1217
 运行系统中的注销时间, 1218
 运行系统中的用户组, 1218

更改名称

授权, 1212
 用户, 1210
 用户组, 1211

更改轴组态, 38

更新, 1299

HMI 设备上的操作系统, 1349
 变量, 1093, 1098
 更改用户后更新用户, 1299

更新周期, 1103

替

替换

模块, 201

最

最大值

提取, 677, 805

最大可用装载存储器

显示, 519

最大循环时间, 234

RE_TRIGR, 886

最大长度

变量, 55

最小值

提取, 675, 803

最小循环时间, 234

服

服务与支持, 186

启用, 187

未

未指定的 CPU, 275

未插入模块区域, 194

未插入的模块, 194

未经授权, 1220

机

机架, 191

插入模块, 196

表格视图, 192

设备概览, 192

权

权限, 1386

创建, 1225

查

查找和替换, 148

使用搜索功能, 149

启动搜索, 149

替换搜索关键字, 150

附加搜索选项, 149

标

标准化, 714, 843

标定, 712, 841

标记, 986

格

格式, 1058

报警文本, 1125

梯

梯形逻辑, 409

梯形逻辑指令

实例, 558

梯级

使用, 436

删除, 438

插入, 437

检

检查

- 浮点数的无效性, 662, 790
- 浮点数的无效性, 662, 790
- 浮点数的有效性, 661, 789

棒

棒图, 969, 1371

- 布局, 1055
- 应用, 1055
- 显示限制线, 1055
- 颜色转变, 1055

棒图分段

- 定义, 1056

椭

椭圆, 969

- 半径 X : , 1060
- 半径 Y, 1060
- 布局, 1060
- 应用, 1060

模

模块

- 删除, 198
- 复制, 199
- 寻址, 237
- 插入, 196, 197
- 插入, 196, 197
- 替换, 201
- 模块的日时钟, 271
- 确定诊断状态, 267
- 移动, 200
- 组态和分配, 18
- 选择, 192

模块信息

- 显示, 95

模块寻址, 297

模块标题, 191

模式

- I/O 域, 1059
- 图形 I/O 域, 1063
- 符号 I/O 域, 1076

模拟量报警, 1108, 1109

- 组态, 1117, 1121, 1128
- 限制值高于, 1091

模拟量报警类型, 1108

模板, 965, 1028

创建, 966

删除, 967

复制, 967

插入, 967

移动, 967

重命名, 967

正

正切

计算正切值, 688, 816

正弦

计算正弦值, 685, 813

比

比较器

大于或等于, 570

小于或等于, 570

比较器操作

不等于, 651

值在范围内, 658

值超出范围, 660

大于, 655

大于或等于, 652

小于, 657

小于或等于, 654

检查无效性, 662

检查有效性, 661

等于, 649

比较器运算

不等于, 779

值在范围内, 786

值超出范围, 787

大于, 783

大于或等于, 780

小于, 785

小于或等于, 782

检查无效性, 790

检查有效性, 789

等于, 777

比较块, 95

注

注销

用户, 1390

注销时间, 1387

在运行系统中更改, 1218

流

流控制, 18

浮

浮点数, 25, 350
 检查无效性, 662, 790
 检查有效性, 661, 789
 浮点数向上取整, 837
 浮点数向下取整, 838

消

消息
 发送, 250
 指定开始, 251
 指定结束, 47, 252

添

添加
 对象到组, 996

滚

滚动条, 969

滞

滞后
 设置, 1123

激

激活
 符号访问, 336

点

点对点通信 (PtP), 245
 自由口协议, 245

特

特性
 IO 域, 1374

图形 IO 域, 1378
 开关, 1383
 按钮, 1382
 日期/时间域, 1373
 符号 IO 域, 1385
 简单配方视图, 1194, 1407
 配方视图, 1183
 特殊字符
 报警文本, 53

状

状态
 模块有故障, 267
 确定和显示在线, 258
 状态信息
 函数列表, 1235
 状态强制, 969

环

环境过滤器, 190

生

生成脉冲, 635, 764

用

用户, 1299, 1386
 Admin, 1393
 分配用户组, 1209
 创建, 1208, 1229, 1388
 创建, 触摸操作, 1388, 1390
 删除, 1210, 1393
 启用, 1219
 在运行系统中创建, 1216
 在运行系统中删除, 1218
 更改, 1217
 更改, 触摸操作, 1392
 更改名称, 1210
 更改用户后更新, 1299
 注销, 1390
 登录, 1219
 管理, 1209, 1217
 用户数据
 区域, 297
 备份, 1387
 导入, 1213
 导出, 1213
 恢复, 1387
 用户更改, 1299

用户权限引起的限制, 99, 100
识别, 99

用户界面

Portal 视图, 104
任务卡, 113
巡视窗口, 111
工作区, 110
视图, 103
详细视图, 114
语言, 117
调整表格, 119
项目树, 107
项目视图, 105

用户界面的语言, 117

用户界面语言, 117

用户程序

功能, 309
查找错误, 467
用户管理, 1198, 1224
应用领域, 1198

用户组, 1386

分配, 1229
创建, 1206, 1227
删除, 1211
在运行系统中更改, 1218
更改名称, 1211
更改显示名称, 1211
未经授权, 1220
管理, 1211
管理权限, 1211

用户自定义文本列表

创建, 155
编辑, 155
编辑取值范围和文本, 156

用户视图, 969, 1212, 1213, 1215

允许的字符, 55
布局, 1057
应用, 1056
组态, 1215
行数, 1057

由

由组态的函数触发

传输数据, 1326

电

电子邮件通知, 1130

画

画面

创建, 960
创建画面的工作步骤, 959
删除, 961
可用于特定设备, 955
复制, 961
字体, 955
插入, 961
移动, 961
组态, 960
缩放, 958
重命名, 961
画面切换, 1289
画面对象, 89
画面模板, 1030

登

登录

不成功, 1220
报告, 1223
用户, 1219

登录对话框

组态, 1222

监

监视

“全部监视”命令, 542
“立即监视一次”命令, 542

监视表格

使用触发器修改, 36
允许用于修改值的操作数, 536
允许用于强制值的操作数, 537
创建, 528
可能的应用, 527
各列的含义, 531
在基本模式和扩展模式之间切换, 532
在存在激活的控制作业期间装载数据块, 36
基本模式, 531
基本步骤, 530
填写监视表格实例, 534
复制, 529
对同一个 CPU 的多重访问, 36
对浮点数取整, 36
布局, 531
扩展模式, 531
概述, 530
测试接线, 527
测试选项总览, 527
语法检查, 534

直

直线, 969
 布局, 1066
 应用, 1066
 线始端, 1066
 线末端, 1066
 线样式, 1066

矩

矩形, 969
 布局, 1072
 应用, 1071
 插入, 1003
 插入和组态, 1002
 组态, 1003
 转角 X 半径, 1072
 转角 Y 半径, 1072

硬

硬件
 检测, 19
 组态和分配参数, 188
 编辑参数, 74, 203
 编辑属性, 74, 203
 硬件 ID, 237
 硬件中断 OB
 分配参数, 384
 描述, 317
 硬件和网络编辑器, 72
 硬件控制的数据流控制, 248
 硬件目录, 72
 任务卡, 184
 浏览, 190
 添加设备, 195
 选择硬件组件, 229
 硬件编辑器
 功能, 177
 巡视窗口, 183
 硬件目录, 184
 组成部分, 177
 设备视图, 180
 硬件诊断, 256
 硬件配置
 添加模块, 73, 196
 添加设备, 195
 硬件限位开关, 38

确

确定日时钟, 271
 确定诊断状态, 267
 确认, 1110, 1113, 1292
 报警, 1400
 组态, 1136, 1137
 错误报警, 1400
 确认模型, 1114

离

离开, 1110, 1290
 离散量报警, 1108, 1109
 组态, 1117, 1120, 1127
 离散量报警类型, 1108

移

移位
 右移, 734, 862
 左移, 736, 864
 移动
 交换, 703
 填充块, 700, 829
 填充块不可中断, 702, 830
 模板, 967
 画面, 961
 硬件组件, 200
 移动值, 824
 移动块, 697
 移动块不可中断, 699, 827
 移动, 信号板, 21
 移动操作
 交换, 832
 值, 824
 块不可中断, 827
 移动块, 826
 移植
 事前调整, 946
 外部变量, 951
 显示历史, 147
 显示日志文件, 148
 移植项目, 949
 简介, 945

程

程序
 启动设置, 102
 存储位置, 103
 起始视图, 101

程序信息

- 在从属性结构中, 511
- 在分配列表中, 498
- 在调用结构中, 505
- 在选项卡中, 516
- 显示, 498
- 视图, 498

程序执行

- 中断驱动, 314
- 循环, 313

程序控制

- 为 0 时块中跳转 (有条件), 846
- 为 1 时块中跳转 (有条件), 845
- 跳转标签, 847
- 返回, 719, 848

程序控制操作

- 为 0 时块中跳转 (有条件), 718
- 为 1 时块中跳转 (有条件), 717

程序段

- 使用, 412, 443
- 关闭, 415, 445
- 删除, 414, 445
- 复制, 414, 444
- 打开, 415, 445
- 插入, 413, 414, 443, 444
- 输入标题, 415, 446
- 输入注释, 415, 446
- 选择, 413, 444

程序段注释

- 显示, 395
- 隐藏, 395

程序状态

- FBD, 527
- LAD, 526
- 修改变量, 525
- 关闭, 524
- 功能, 523
- 打开, 524
- 编辑块, 525
- 说明, 21

程序编辑器, 78

- “指令”任务卡, 391
- “测试”任务卡, 392
- 使用键盘, 396
- 功能, 388
- 块接口, 390
- 工具栏, 389
- 常规设置, 397
- 指令窗口, 394
- 收藏夹, 390
- 结构, 388

空

空功能框

- 插入 FBD 元素, 448
- 插入 LAD 元素, 418

站

站

- 删除, 198
- 复制, 199
- 移动, 200

端

端口号, 280

符

符号

- 在从属性结构中, 512
- 在分配列表中, 500
- 在调用结构中, 507
- 寻址, 1084
- 报警类别, 1131
- 用于工作模式, 260
- 用于比较, 259
- 用于诊断, 259
- 重叠图标, 259

符号 I/O 域, 969

- 布局, 1076
- 文本列表, 1076
- 模式, 1076

符号 IO 域, 1384

- 应用, 1384
- 操作, 1385
- 特性, 1385

符号常量, 21

符号库, 969

符号编程

- 显示绝对地址, 334
- 概述, 333

符号访问, 1082

- 优势, 1082
- 全局数据块, 478
- 激活, 336
- 限制, 1083

等

等于, 649, 777

简

简单报警窗口

应用, 1140, 1396

简单报警视图, 1140, 1396

应用, 1140, 1396

控制元素, 1141

操作, 1141, 1397

操作员控件, 1397

简单用户视图, 1387

布局, 1388

应用, 1387

操作, 1388

简单配方视图, 1181

布局, 1192, 1406

操作, 1193, 1406

特性, 1194, 1407

约束, 1183

菜单命令, 1194, 1406, 1407

管

管理

图形, 989

用户, 1209

用户组, 1211

运行系统中的用户, 1217

类

类型, 1075

精

精简面板

运行系统启动, 1341, 1356

系

系统函数, 1231

AcknowledgeAlarm, 1266

ActivateCleanScreen, 1246

ActivatePreviousScreen, 1247

ActivateScreen, 1245

ActivateScreenByNumber, 1246

AdjustContrast, 1244

AlarmViewAcknowledgeAlarm, 1265

AlarmViewEditAlarm, 1264

AlarmViewShowOperatorNotes, 1265

ButtonPress, 1273

ButtonRelease, 1274

CalibrateTouchScreen, 1255

ChangeConnection, 1284

ClearAlarmBuffer, 1263

ClearAlarmBufferProtoolLegacy, 1263

DecreaseFocusedValue, 1283

DecreaseTag, 1283

EditAlarm, 1249

GetGroupNumber, 1261

GetPassword, 1261

GetUserName, 1260

GoToEnd, 1253

GoToHome, 1253

IncreaseFocusedValue, 1252

IncreaseTag, 1252

InverseLinearScaling, 1254

InvertBit, 1254

LinearScaling, 1262

Logoff, 1244

Logon, 1248

PageDown, 1276

PageUp, 1276

RecipeViewBack, 1272

RecipeViewDeleteDataRecord, 1267

RecipeViewGetDataRecordFromPLC, 1267

RecipeViewMenu, 1268

RecipeViewNewDataRecord, 1266

RecipeViewOpen, 1269

RecipeViewRenameDataRecord, 1271

RecipeViewSaveAsDataRecord, 1270

RecipeViewSaveDataRecord, 1270

RecipeViewSetDataRecordToPLC, 1269

ResetBit, 1273

RezeptanzeigeZeigeHilfetext, 1271

ScreenObjectCursorDown, 1250

ScreenObjectCursorUp, 1249

ScreenObjectPageDown, 1251

ScreenObjectPageUp, 1250

SetBit, 1277

SetBitWhileKeyPressed, 1277

SetConnectionMode, 1279

SetDeviceMode, 1276

SetLanguage, 1278

SetTag, 1279

ShowAlarmWindow, 1286

ShowLogonDialog, 1285

ShowOperatorNotes, 1285

SimulateTag, 1281

StopRuntime, 1282

TraceUserChange, 1282

TrendViewBackToBeginning, 1260

TrendViewCompress, 1257

TrendViewExtend, 1256

TrendViewRulerBackward, 1258

TrendViewRulerForward, 1257

TrendViewScrollBack, 1256

TrendViewScrollForward, 1255

TrendViewSetRulerMode, 1259

TrendViewStartStop, 1259

- UpdateTag, 1248
 - 使用, 1233
 - 函数列表, 1235
 - 可用, 1241
 - 在函数列表中, 1233
 - 在脚本中, 1233
 - 应用, 1232, 1233
- 系统存储器
 - 操作数区域, 293
 - 诊断缓冲区, 273, 297
 - 过程映像输入/输出, 295
- 系统定义的 PLC 报警, 1108
- 系统定义的文本列表
 - 修改文本, 156
 - 编辑, 156
- 系统报警, 1108, 1109
 - 参数, 1146
 - 组态, 1117
- 系统相关信息
 - 制造商信息, 262
 - 模块信息, 262
- 系统限制
 - HMI 设备, 1414

索

- 索引变量, 1096, 1097

线

线圈

- 为 0 时块中跳转 (有条件), 718
- 为 1 时块中跳转 (有条件), 717
- 取反线圈, 621
- 在信号上升沿置位操作数, 631
- 在信号下降沿置位操作数, 632
- 复位输出, 622
- 置位输出, 623
- 输出线圈, 562, 620
- 重置 IEC 定时器, 26, 27

线始端

- 直线, 1066

线性标定

- 变量、线性标定, 1095
- 变量、线性转换, 1094

线性编程, 310

线末端

- 直线, 1066

线样式

- 直线, 1066

组

组

- 创建, 994
- 删除对象, 996
- 取消, 995
- 取消编组, 995
- 添加对象, 996
- 编辑, 994
- 组态, 1121, 1129, 1215
 - PLC 报警, 1117
 - 事件, 1182
 - 事件驱动的任务, 1123
 - 动画, 1021
 - 变量, 1088
 - 多个变量, 1089
 - 报警确认, 1136, 1137
 - 报警组, 1119
 - 报警视图, 1132
 - 按钮, 1222, 1226
 - 模拟量报警, 1117, 1121, 1128
 - 用于来自 PLC 的值的趋势视图, 1104
 - 用户视图, 1215
 - 画面, 960
 - 登录对话框, 1222
 - 矩形, 1003
 - 硬件, 188
 - 离散量报警, 1117, 1120, 1127
 - 系统报警, 1117
 - 访问保护, 1226
 - 配方视图, 1185
- 组态具有以太网的网络, 211
 - IP 地址和子网掩码之间的关系, 212
 - 创建专用子网, 212
 - 设置 IP 地址, 211
 - 设置子网掩码, 211
 - 链接网络, 213
- 组态阶段, 1351
- 组成部分
 - 报警, 1115
- 组织块, 312
 - 创建, 374
 - 功能, 312
 - 启动信息, 312
 - 调用, 312

结

- 结构, 355, 1224
 - 交叉引用列表的, 520, 1417
 - 从属性结构的, 512
 - 资源选项卡, 517
- 结构化编程, 310

绝

绝对值
 计算, 674, 802

编

编码, 728, 857
编程, 362
 线性, 310
 结构化, 310
编程语言, 78
 FBD, 440
 LAD, 409
 更改, 468
 更改操作的规则, 468

编组

对象, 994

编译

在脉冲发生器被禁用时, 20
地址参数, 54
更正编译错误, 488
用户程序, 486, 487
软件, 486, 487
项目, 1334

编辑

函数列表, 1237
报警, 1401
文件夹链接, 990
组, 994
组中的对象, 998
配方数据记录, 1410

编辑打印模板, 145**编辑网络连接**

从网络中断开, 209
复制子网, 210
复制子网和设备, 211

编辑语言

使用, 157
切换, 19
指定, 159

缩**缩放**

画面, 958

网**网络概览**

基本功能, 207
用于编辑网络概览表格的基本功能, 207

网络编辑器

功能, 177
巡视窗口, 183
硬件目录, 184
组成部分, 177
网络视图, 178
网络视图, 72
 添加设备, 195
 网络编辑器, 178
网络访问, 1422
 打开属性, 1423

置**置位**

位域, 624, 753
在信号上升沿置位操作数, 631, 760
在信号上升沿置位输出, 633, 762
在信号下降沿置位操作数, 632, 761
在信号下降沿置位输出, 634, 763
输出, 568, 623

置位输出, 559**翻****翻转**

对象, 972

联**联网**

通信伙伴, 1302

联网设备

不存在子网时的联网, 77, 205
与现有子网联网, 206
同时联网多个接口, 205
组态网络的基础, 204
编辑接口参数, 209
编辑网络参数, 209
要求, 208
通信类型, 204
项目内的网络, 204

背**背景数据块**

保持特性, 479
创建, 321
定义, 321
设置保持特性, 480

背景色
 动态化, 1022

脉

脉冲发生器
 启用, 942
 禁用, 942
 编译硬件, 20
 诊断数据, 19
 限制, 38

脚

脚本
 在函数列表中, 1235

自

自动
 报告, 1123
自定义报警类别, 1112
自由格式的注释
 删除, 440, 467
 插入, 438, 466
 简介, 438, 466
 编辑, 439, 466

范

范围
 值在范围内, 658, 786
 值超出范围, 660, 787

获

获得用户权限
 临时获得, 100
 以管理员权限登录操作系统, 100
 使用 Windows 用户帐户控制, 100

菜

菜单命令
 简单配方视图, 1194, 1407

补

补码
 创建, 670, 798

表

表格
 调整, 119
 键盘快捷键, 121

装

装载, 143
 在设备上装载块, 255
 指定设置, 1336
 项目数据, 94
装载存储器, 293, 516
 显示, 519

规

规划任务, 1295
规则
 更改编程语言, 468

视

视图选项
 用于分配列表, 502
 设置从属性结构, 514
 设置调用结构, 509

解

解码, 727, 855
解锁
 用户, 1219

触

触发器
 复位置位, 756
 置位置位, 626, 627, 755
触发源, 1297
 事件触发, 1297
触摸操作
 IO 域, 1375

计

计数

- 减计数, 643, 771
- 加和减, 644, 773
- 加和减, 644, 773
- 加计数, 641, 770
- 加计数和减计数, 562, 570

计数器

- 加计数和减计数, 562

计数器, 高速, 241

计数器操作

- 减计数, 643, 771
- 加计数, 641, 770
- 加计数和减计数, 644, 773
- 控制高速计数器, 647, 775
- 高速计数器, 647, 775

计数器模式, 242

计数器输入, 242

计算

- 余弦值, 686, 815
- 反余弦值, 690, 819
- 反正切值, 692, 820
- 反正弦值, 689, 817
- 平方, 680, 808
- 指数值, 684, 812
- 正切值, 688, 816
- 正弦值, 685, 813
- 绝对值, 674, 802
- 自然对数, 683, 811

设

设备

- 删除, 198
- 复制, 199
- 故障设备, 257
- 添加到硬件配置, 195
- 移动, 200
- 联网, 1302

设备信息, 257

设备和网络

- 登陆选项, 65

设备地址, 237

设备概览, 192

- 地址范围, 237

设备视图, 72

- 插入信号板, 201
- 插入模块, 73, 196
- 未插入模块区域, 194
- 机架, 191
- 硬件编辑器, 180
- 编辑参数, 74, 203
- 编辑属性, 74, 203

表格视图, 192

设备概览, 192

设置

FBD, 411, 442

LAD, 411, 442

从属性结构的视图选项, 514

分配列表的视图选项, 502

常规, 397

延迟时间, 1123

更改, 398, 412, 442

滞后, 1123

语言, 1371

设置以太网接口的参数, 1425

删除动态 IP 地址, 1427

参数设置中的选项, 1423

可修改的参数, 1424

添加动态 IP 地址, 1426

连接到子网, 1425

设置助记符, 409

设置语言, 1371

访

访问保护, 1221, 1222

组态, 1226

诊

诊断

模块特定, 270

诊断文本, 19

诊断缓冲区

基本知识, 273

基础知识, 297

组织结构, 273

诊断错误中断 OB, 318

详

详细视图, 114

语

语言切换, 1361

ProSave, 1347

语言行为

屏幕键盘, 55

请

请求时, 1093

读

读出

- 配方数据记录, 1412
- 读出诊断缓冲区, 267

调

调度程序

- 事件触发, 1297, 1298
- 函数列表, 1297
- 工作区, 1296
- 应用领域, 1294
- 禁用任务, 1297
- 触发源, 1297

调整, 1288

调整项目

- 适合于不同的 HMI 设备, 1339

调用层级

- 嵌套深度, 323
- 顺序, 323

调用结构, 505

- 图标的含义, 533
- 布局, 507
- 显示, 508
- 符号的含义, 507
- 简介, 505
- 设置视图选项, 509

调用结构中的符号, 533

调试

- HMI 设备, 1352

资

资源, 516

- 保持性存储器, 516
- 保持性存储器数据, 516
- 工作存储器, 516
- 简介, 516
- 装载存储器, 516

资源选项卡

- 结构, 517

赋

赋值, 570, 575, 620, 749

- 取反, 750

起

起始值

- 变量, 1092, 1093
- 起始地址, 237, 297
- 起始画面, 962
- 起始视图, 101

趋

趋势视图, 969, 1379

- 为来自 PLC 的值组态, 1104
- 布局, 1064, 1379
- 应用, 1064, 1379
- 按钮, 1065
- 操作, 1379
- 操作员控件, 1379

跳

跳转

- 为 0 时块中跳转 (有条件), 718, 846
- 为 1 时块中跳转 (有条件), 717, 845
- 跳转标签, 719, 847
- 返回, 719, 848
- 跳转标签, 719, 847

转

转换, 834

- 值, 705
- 数据类型, 360

转换器

- 取整数字值, 706, 835
- 截取整数, 711, 839
- 标准化, 714, 843
- 标定, 712, 841
- 浮点数向上取整, 708, 837
- 浮点数向下取整, 709, 838
- 转换, 834
- 转换值, 705
- 转角 X 半径, 1072
- 转角 Y 半径, 1072

软

软件控制的数据流控制, 248

- 软键, 1028, 1031
- 全局, 1030

输

输入

- 删除, 427, 456

- 输入 (I), 293

- 输入字 (IW), 293

- 输入字节 (IB), 293

输出

- 删除, 427, 456

- 在信号上升沿置位, 633, 762

- 在信号下降沿置位, 634, 763

- 复位, 560, 568, 569, 622, 751

- 插入, 426, 456

- 设置, 560, 568, 569, 623, 752

- 输出 (O), 293

- 输出双字 (QD), 293

输出域

- 指针变量, 53

- 插入, 1124

- 输出字 (QW), 293

- 输出字节 (QB), 293

- 输出线圈, 565, 620

边

边沿

- 在信号上升沿置位操作数, 631, 760

- 在信号上升沿置位操作数, 631, 760

- 在信号上升沿置位输出, 633, 762

- 在信号下降沿置位操作数, 632, 761

- 在信号下降沿置位输出, 634, 763

- 扫描操作数的信号上升沿, 628, 757

- 扫描操作数的信号下降沿, 630, 758

- 查询操作数的信号上升沿, 560, 569

过

过滤器

- 为分配列表定义, 503

- 分配列表, 504

- 删除, 503

- 在分配列表中, 502

- 硬件目录, 190

- 选择, 504

- 过程控制阶段, 1351

过程数据区

- 模块寻址, 297

- 模块起始地址, 297

过程映像, 297

- I/O 访问错误, 296

- 基础知识, 295

- 更新, 296

运

运动控制, 87

- 控制面板的响应时间, 37

- 调用指令, 39

- 通过 MC_Reset 确认错误, 37

- 错误 ID 和错误信息, 39

运行系统

- 仿真, 1341

- 函数列表的完成, 1238

- 用户管理, 1198

- 起始画面, 962

- 运行系统中的用户管理设置, 1204

- 运行系统停止, 1293

运行系统启动

- 精简面板, 1341, 1356

运行系统设置

- 用户管理, 1204

- 运行系统语言, 1359

- 字体, 1360

- 选择, 1359

进

- 进入, 1110, 1290

连

连接, 221

- 创建, 218

- 删除, 220

- 离线, 1318

- 连接 ID, 18

连接参数分配

- 启动, 218

- 常规, 214

- 概述, 214

- 连接参数, 216

连接描述

- 数据块, 277

- 更改参数值, 280

- 结构, 278

- 连接资源, 222

连续读

- 变量, 1094

选

- 选择, 730, 858

- 多个对象, 972, 978

透

透明

在图形中, 989

透明色, 1061

面板 HMI 上的显示, 1062

面板 HMI 上的显示, 1377

递

递减, 673, 800

递增, 672, 799

通

通信

使用区域指针, 1301

使用变量, 1301

发送数据, 902

周期负载, 235

建立连接, 891, 898

接收数据, 904

终止连接, 891, 900

通信伙伴, 1300

联网, 1302

通信协议

定义, 248

概述, 248

通信模块, 246

属性, 246

通信端口

组态, 246

通信负载, 235

通信驱动程序, 1301

逻

逻辑路径

使用, 464

删除, 465

插入, 465

逻辑运算

通过 AND, 721, 849

通过 OR, 722, 851

通过异或, 724, 852

逻辑运算结果

取反, 619

配

配方, 1173, 1174, 1175, 1333, 1403

使用, 1174

基本信息, 1174

基础知识, 1173

导入, 1332

导出, 1330

应用实例: 机械设备参数分配, 1174

应用实例: 批量生产, 1174

控制, 1404

数据流, 1405

数据记录, 1175, 1403

文本列表的使用, 1180

新建, 1187

条目, 1175

配方元素

新建, 1187

配方列表, 1181

配方数据记录

与 PLC 同步, 1410

从 PLC 读取, 1412

传送到 PLC, 1412

传送项目, 56

创建, 1409

删除, 1411

在 HMI 设备上创建, 1195, 1196

文本列表的使用, 1180

新建, 1187

编辑, 1410

配方视图, 969, 1181

仅显示数值, 1182

使用功能键操作, 1183

可组态的事件, 1182

布局, 1072

应用, 1072

控制元素, 1072

更新, 1181

特性, 1183

画面切换特性, 1183

简单, 1181

组态, 1185

配方数据记录, 1176, 1184

采

采集周期

变量, 1093, 1098, 1103

采集模式

变量, 1093

释

释放, 1292

重

- 重叠图标, 259
- 重命名
 - 变量, 1088
 - 层, 1043
 - 模板, 967
 - 画面, 961
- 重新调试
 - HMI 设备, 1352
- 重新链接
 - PLC 变量, 368

量

- 量表, 969

锁

- 锁定用户
 - 启用, 1219

错

- 错误 ID, 890
- 错误信息, 470, 887
- 错误处理
 - 局部错误处理, 470
 - 系统响应, 469
- 错误报警
 - 确认, 1400
- 错误消息
 - 传送, 1350
 - 将数据下载到 PLC, 55

键

- 键操作
 - IO 域, 1375
 - 日期/时间域, 1373
- 键盘快捷键
 - 窗口, 120
 - 编辑文本, 122
 - 表格, 121
 - 项目树, 121
 - 项目编辑, 119

闪

- 闪烁, 1022

间

- 间接寻址, 1096

限

- 限值
 - 提取, 678, 806
- 限制值
 - 变量, 1091
- 限制值检查, 1367, 1369
- 限制值高于
 - 模拟量报警, 1091
- 限制范围
 - 变量, 1091
- 限定值, 1291

除

- 除, 564, 573, 668, 795
- 除法的余数
 - 返回, 669, 797

隐

- 隐藏参数, 431, 461
- 隐藏输入, 1059

需

- 需要进入确认的单个报警, 1114

项

- 项目
 - 下载, 1352
 - 保存, 134
 - 保护块, 496
 - 关闭, 135
 - 创建, 133
 - 删除, 135
 - 同步比较动作, 140
 - 基本知识, 133
 - 多个连接, 1318
 - 定义比较动作, 140

- 打开, 134
 - 显示和隐藏相同的比较对象, 138
 - 更新比较结果, 139
 - 比较, 138
 - 比较编辑器, 136
 - 移植, 949
 - 编译, 1334
 - 编译项目数据, 142
 - 装载项目数据, 143
 - 运行详细比较, 139
 - 项目库, 1044
 - 使用元素, 166
 - 使用元素和部件视图, 164
 - 使用过滤视图, 168, 175
 - 创建文件夹, 165
 - 删除元素, 168
 - 基本知识, 165
 - 添加元素, 165
 - 编辑元素, 167
 - 项目数据
 - 导出配方, 1330
 - 有关编译的信息, 141
 - 有关装载的信息, 142
 - 编译, 142
 - 装载, 143
 - 装载到可访问设备, 144
 - 项目文本
 - 显示参考文本, 160
 - 翻译各文本, 159
 - 翻译成项目语言, 157
 - 项目树, 67
 - 功能, 107
 - 显示, 395
 - 添加设备, 195
 - 键盘快捷键, 121
 - 隐藏, 395
 - 项目范围的报警类别, 1112
 - 项目视图, 66, 105
 - 项目语言
 - 使用, 157
 - 指定, 158
- 预
- 预定义的报警类别, 1111
- 颜
- 颜色转变
 - 棒图, 1055
- 高
- 高速计数器
 - 工作原理, 241
 - 常规, 241
 - 组态, 244
 - 诊断数据, 19
- 默
- 默认字体, 1360