

SIMATIC

S7-1200 入门手册

设备手册

前言

强大灵活的 S7-1200 简介

1

STEP 7 使工作更轻松

2

入门指南

3

简化了 PLC 概念

4

轻松创建设备配置

5

简化了编程

6

轻松实现设备间通信

7

PID 非常简单

8

便于连接 Internet 的 Web
服务器

9

轻松进行运动控制

10

轻松使用在线工具

11

IO-Link 非常简单

12

技术规范

A




用 V4.1 CPU 更换 V3.0 CPU

B

法律资讯

警告提示系统

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

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

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


合格的专业人员

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

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

按规定使用Siemens 产品

请注意下列说明：

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

商标

所有带有标记符号®的都是西门子股份有限公司的注册商标。本印刷品中的其他符号可能是一些其他商标。若第三方出于自身目的使用这些商标，将侵害其所有者的权利。

责任免除

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

前言

欢迎使用 S7-1200。SIMATIC S7-1200

紧凑型控制器是一款节省空间的模块化控制器，适合要求简单或高级逻辑、HMI 和网络功能的小型自动化系统。S7-1200

设计紧凑、成本低廉且功能强大，是控制小型应用的完美解决方案。

作为 SIMATIC“全集成自动化”(TIA, Totally Integrated Automation) 计划的一部分，S7-1200 产品系列和 TIA Portal 编程软件为您提供了满足您自动化要求所需的灵活性。

S7-1200 将帮助您轻松完成最具挑战性的任务！

SIMATIC S7-1200 控制器解决方案是专为“紧凑型”控制器类别设计的，由 SIMATIC S7-1200 控制器和 SIMATIC HMI 基本型面板组成，此二者均可使用 TIA Portal 工程软件进行编程。

由于实现了使用同一个工程软件对两种设备进行编程，开发成本得以显著降低。TIA Portal 包括用于 S7-1200 编程的 STEP 7 和用于设计基本型面板项目的 WinCC。



S7-1200 紧凑型控制器包括：

- 内置 PROFINET
- 支持运动控制的高速 I/O、对空间要求最小但需要额外 I/O 的板载模拟量输入，用于脉冲序列和脉宽应用 (页 76) 的 4 个脉冲发生器以及最多 6 个高速计数器 (页 136)
- CPU 模块中内置的板载 I/O 点提供 6 到 14 个输入点和 4 到 10 个输出点。



用于 DC、继电器或模拟 I/O 的信号模块扩展了 I/O 点数，创新的信号板卡在 CPU 的正面以提供附加 I/O (页 19)。

SIMATIC HMI 基本型面板 (页 22) 是专为 S7-1200 设计的。

本入门手册介绍了 S7-1200 PLC。

下文概述了该设备的多个特征和功能。

有关更多信息，请参见“S7-1200 可编程控制器系统手册”。有关 UL 和 FM 证书、CE 标签、C-Tick 和其它标准的信息，请参考技术规范 (页 391)。

本手册介绍了以下产品：

- STEP 7 V13 SP1 Basic 和 Professional
- S7-1200 CPU 固件版本 V4.1

文档和信息

S7-1200 和 STEP 7 提供了各种文档和其它资源，供您查找所需的技术信息。

- S7-1200 可编程控制器系统手册提供有关整个 S7-1200 产品系列的操作、编程和规范的特定信息。除系统手册外，S7-1200 入门手册还对 S7-1200 系列的功能进行了更详细的概述。

系统手册和入门手册都有电子版 (PDF)。

电子版手册既可从客户支持网站下载，也可在随每个 S7-1200 CPU 提供的文档光盘中找到。

- 通过 STEP 7 的在线信息系统，可以直接访问概念性信息和具体说明，它们介绍了编程数据包的操作和功能以及 SIMATIC CPU 的基本操作。
- My Documentation Manager 可用于访问电子版 (PDF) 的 SIMATIC 文档集，包括系统手册、入门手册和 STEP 7 信息系统。借助 My Documentation Manager，您可以通过拖放不同文档中的主题来创建自己的自定义手册。

客户支持门户 (<http://support.automation.siemens.com/CN>) 在 mySupport 下提供了指向 My Documentation Manager 的链接。

- 客户支持网站还提供了 S7-1200 和 STEP 7 的播客、常见问题解答及其它有用文档。播客采用一些侧重于特定功能或情景的指导性简短视频演示来说明 STEP 7 所提供的交互作用、便捷性及效率。请访问以下网站来获取播客集合：
 - STEP 7 Basic 网页 (<http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/step7-basic/Pages/Default.aspx>)
 - STEP 7 Professional 网页 (<http://www.automation.siemens.com/mcms/simatic-controller-software/en/step7/step7-professional/Pages/Default.aspx>)
- 您还可以关注或加入服务与支持技术论坛 (<https://www.automation.siemens.com/WW/forum/guests/Conferences.aspx?Language=en&siteid=csius&treeLang=en&groupid=4000002&extranet=standard&viewreg=WW&nodeid0=34612486>)关于产品的讨论。通过论坛，您可以与各领域的产品专家互动。
 - S7-1200 (<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=258&Language=en&onlyInternet=False>) 论坛
 - STEP 7 Basic (<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=265&Language=en&onlyInternet=False>) 论坛

服务与支持

除了文档之外，西门子公司还在 Internet 和客户支持网站 (<http://www.siemens.com/tiaportal>)上提供了专业技术知识。

如需要回答任何技术问题、培训或订购 S7

产品方面的帮助，请与西门子经销商或销售部联系。

因为西门子销售代表都经过技术培训并掌握有关操作、过程和工业以及有关您使用的各种西门子产品的最具体的知识，所以他们能够最快最高效地回答您可能遇到的任何问题。

安全信息

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

有鉴于此，西门子不断对产品和解决方案进行开发和完善。

西门子强烈建议您定期检查产品的更新和升级信息。

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

此外，还需考虑到可能使用的所有第三方产品。更多有关工业安全的信息，请访问 Internet (<http://www.siemens.com/industrialsecurity>)。

要及时了解有关产品的更新和升级信息，请订阅相关产品的实事信息。

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

目录

前言	3
1 强大灵活的 S7-1200 简介	15
1.1 S7-1200 PLC 简介	15
1.2 CPU 的扩展功能	19
1.3 S7-1200 模块	20
1.4 HMI 基本型面板	22
1.5 安装尺寸和间隙要求	23
1.6 新功能	29
2 STEP 7 使工作更轻松	33
2.1 轻松向用户程序中插入指令	34
2.2 从工具栏轻松访问收藏的指令	35
2.3 将输入或输出轻松地添加到 LAD 和 FBD 指令中	35
2.4 可扩展指令	36
2.5 轻松更改 CPU 的工作模式	37
2.6 轻松修改 STEP 7 的外观和组态	37
2.7 便于访问的项目库和全局库	38
2.8 便于选择指令版本	38
2.9 在编辑器之间轻松拖放	39
2.10 更改 DB 的调用类型	40
2.11 暂时从网络中断开设备	41
2.12 轻松实现实际“拔出”模块而不会丢失组态数据	42
3 入门指南	43
3.1 创建项目	43
3.2 为 CPU 的 I/O 创建变量	44
3.3 在用户程序中创建一个简单程序段	46
3.4 使用变量表中的 PLC 变量对指令进行寻址	48
3.5 添加“功能框”指令	49
3.6 为复杂数学等式使用 CALCULATE 指令	50

3.7	在项目中添加 HMI 设备	53
3.8	在 CPU 和 HMI 设备之间创建网络连接	54
3.9	创建 HMI 连接以共享变量	54
3.10	创建 HMI 画面	55
3.11	为 HMI 元素选择 PLC 变量	56
4	简化了 PLC 概念	57
4.1	每个扫描周期均执行的任务	57
4.2	CPU 的工作模式	59
4.3	用户程序的执行	60
4.3.1	在 RUN 模式下处理扫描周期	60
4.3.2	OB 可帮助用户构建用户程序	61
4.3.3	事件执行的优先级与排队	62
4.4	存储区、寻址和数据类型	66
4.4.1	S7-1200 支持的数据类型	67
4.4.2	对存储区进行寻址	70
4.4.3	访问一个变量数据类型的“片段”	72
4.4.4	访问带有一个 AT 覆盖的变量	74
4.5	脉冲输出	76
5	轻松创建设备配置	79
5.1	上传已连接 CPU 的组态	80
5.2	将 CPU 添加到组态中	82
5.3	更改设备	83
5.4	将模块添加到组态	83
5.5	组态控制	85
5.6	组态 CPU 和模块的运行	86
5.6.1	系统存储器和时钟存储器提供了标准功能	88
5.7	组态 CPU 的 IP 地址	91
5.8	防止对 CPU 或代码块的访问非常容易	93
5.8.1	专有技术保护	95
5.8.2	复制保护	96

6	简化了编程.....	99
6.1	轻松设计用户程序.....	99
6.1.1	使用 OB 组织用户程序.....	101
6.1.2	FB 和 FC 使模块化任务编程变得很轻松.....	103
6.1.3	数据块为程序数据提供了便捷的存储方式.....	105
6.1.4	创建新代码块.....	106
6.1.5	创建可重复使用的代码块.....	107
6.1.6	从一个代码块调用另一个代码块.....	108
6.2	易于使用的编程语言.....	108
6.2.1	梯形图 (LAD).....	108
6.2.2	功能块图 (FBD).....	109
6.2.3	SCL 概述.....	110
6.2.4	SCL 程序编辑器.....	111
6.3	功能强大的指令使编程更加轻松.....	112
6.3.1	提供您所期望的基本指令.....	112
6.3.2	比较器和移动指令.....	115
6.3.3	转换操作.....	116
6.3.4	计算指令使数字运算更加轻松.....	118
6.3.5	定时器操作.....	120
6.3.6	计数器操作.....	125
6.3.7	脉冲宽度调制 (PWM).....	127
6.4	轻松创建数据日志.....	129
6.5	轻松监视和测试用户程序.....	132
6.5.1	监视表格和强制表格.....	132
6.5.2	用于显示使用情况的交叉引用.....	132
6.5.3	用于检查调用层级的调用结构.....	133
6.5.4	用于监视硬件的诊断指令.....	134
6.5.4.1	读取 CPU 上 LED 的状态.....	134
6.5.4.2	用于读取设备诊断状态的指令.....	135
6.6	高速计数器 (HSC).....	136
6.6.1	高速计数器的使用方法.....	137
6.6.2	组态 HSC.....	146
7	轻松实现设备间通信.....	149
7.1	创建网络连接.....	150
7.2	通讯选项.....	151
7.3	V4.1 异步通信连接.....	153
7.4	PROFINET 和 PROFIBUS 指令.....	155
7.5	PROFINET.....	157
7.5.1	开放式用户通信.....	157

7.5.1.1	特殊模式.....	158
7.5.1.2	开放式用户通信指令的连接 ID.....	159
7.5.1.3	PROFINET 连接的参数.....	163
7.5.2	组态本地/伙伴连接路径.....	165
7.6	PROFIBUS.....	169
7.6.1	PROFIBUS CM 的通信服务.....	170
7.6.2	PROFIBUS CM 用户手册参考资料.....	171
7.6.3	添加 CM 1243-5 (DP 主站) 模块和 DP 从站.....	172
7.6.4	给 CM 1243-5 模块和 DP 从站分配 PROFIBUS 地址.....	173
7.7	AS-i.....	175
7.7.1	添加 AS-i 主站 CM 1243-2 和 AS-i 从站.....	176
7.7.2	为 AS-i 从站分配 AS-i 地址.....	177
7.8	S7 通信.....	180
7.8.1	GET 和 PUT 指令.....	180
7.8.2	创建 S7 连接.....	181
7.8.3	GET/PUT 连接参数分配.....	182
7.9	GPRS.....	183
7.9.1	连接到 GSM 网络.....	183
7.9.2	CP 1242-7 的应用.....	185
7.9.3	CP-1242-7 的其它属性.....	186
7.9.4	组态和电气连接.....	186
7.9.5	更多信息.....	187
7.9.6	附件.....	187
7.9.7	参考 GSM 天线手册.....	188
7.9.8	遥控组态示例.....	189
7.10	PtP、USS 和 Modbus 通信协议.....	194
7.10.1	点对点通讯.....	194
7.10.2	使用串行通信接口.....	196
7.10.3	PtP 指令.....	197
7.10.4	USS 指令.....	198
7.10.5	Modbus 指令.....	200
8	PID 非常简单.....	203
8.1	插入 PID 指令和工艺对象.....	205
8.2	PID_Compact 指令.....	207
8.3	PID_Compact 指令 ErrorBit 参数.....	212
8.4	PID_3Step 指令.....	214
8.5	PID_3Step 指令的 ErrorBit 参数.....	222
8.6	PID_Temp 指令.....	224
8.6.1	概述.....	224

8.6.2	PID_Temp 控制器的操作	229
8.6.3	级联控制器	232
8.7	PID_Temp 指令 ErrorBit 参数	236
8.8	组态 PID_Compact 和 PID_3Step 控制器	239
8.9	组态 PID_Temp 控制器	242
8.10	调试 PID_Compact 和 PID_3Step 控制器	260
8.11	调试 PID_Temp 控制器	262
9	便于连接 Internet 的 Web 服务器	275
9.1	轻松使用标准 Web 页面	276
9.2	可影响到 Web 服务器使用的限制	279
9.3	轻松创建用户定义的 Web 页面	280
9.3.1	轻松创建定制的“用户自定义”Web 页面	280
9.3.2	特定于用户定义 Web 页面的限制	282
9.3.3	用户定义的 Web 页面的组态	283
9.3.4	使用 WWW 指令	283
10	轻松进行运动控制	285
10.1	“定相”(Phasing)	291
10.2	组态脉冲发生器	293
10.3	开环运动控制	294
10.3.1	组态轴	294
10.3.2	调试	298
10.4	闭环运动控制	304
10.4.1	组态轴	304
10.4.2	调试	311
10.5	组态 TO_CommandTable_PTO	318
10.6	S7-1200 的运动控制操作	322
10.6.1	用于运动控制的 CPU 输出	322
10.6.2	用于运动控制的硬件和软件限位开关	324
10.6.3	回原点	328
10.6.3.1	使轴回原点	328
10.6.3.2	回原点参数的组态	329
10.6.3.3	主动回原点的顺序	332
10.7	运动控制指令	333
10.7.1	MC 指令概述	333
10.7.2	MC_Power (发布/阻止轴) 指令	334
10.7.3	MC_Reset (确认错误) 指令	337
10.7.4	MC_Home (使轴回原点) 指令	339

10.7.5	MC_Halt (暂停轴) 指令.....	342
10.7.6	MC_MoveAbsolute (绝对定位轴) 指令.....	344
10.7.7	MC_MoveRelative (相对定位轴) 指令.....	346
10.7.8	MC_MoveVelocity (以预定义速度移动轴) 指令.....	348
10.7.9	MC_MoveJog (在点动模式下移动轴) 指令.....	351
10.7.10	MC_CommandTable (按移动顺序运行轴命令) 指令.....	353
10.7.11	MC_ChangeDynamic (更改轴的动态设置) 指令.....	356
10.7.12	MC_WriteParam (写入工艺对象的参数) 指令.....	358
10.7.13	MC_ReadParam (读取工艺对象的参数) 指令.....	361
11	轻松使用在线工具.....	363
11.1	转到在线并连接到 CPU.....	363
11.2	与在线 CPU 交互.....	364
11.3	转到在线模式监视 CPU 中的值.....	366
11.4	显示用户程序的状态非常简单.....	367
11.5	使用监视表格监视 CPU.....	367
11.6	使用强制表格.....	369
11.7	捕获 DB 的在线值以重设起始值.....	372
11.8	上传项目元素.....	373
11.9	比较离线 CPU 与在线 CPU.....	374
11.10	显示诊断事件.....	375
11.11	设置 IP 地址和日时钟.....	375
11.12	复位为出厂设置.....	376
11.13	更新固件.....	377
11.14	将永久 IP 地址下载到在线 CPU.....	378
11.15	使用“未指定的 CPU”上传硬件配置.....	379
11.16	在 RUN 模式下下载.....	380
11.16.1	在 RUN 模式下更改程序.....	382
11.17	根据触发条件跟踪并记录 CPU 数据.....	383
12	IO-Link 非常简单.....	385
12.1	IO-Link 技术总览.....	385
12.2	IO-Link 系统的组件.....	385
12.3	上电后.....	385
12.4	IO-Link 协议.....	386
12.5	现场总线中的组态.....	386

12.6	IO-Link 和 STEP 7 程序	386
12.7	SM 1278 4xIO-Link 主站	387
A	技术规范	391
A.1	常规技术规范	391
A.2	CPU 模块	404
A.3	数字 I/O 模块	410
A.3.1	SB 1221 SB 1222 和 SB 1223 数字量输入/输出 (DI、DQ 和 DI/DQ)	410
A.3.2	SM 1221 数字量输入 (DI)	413
A.3.3	SM 1222 数字量输出 (DQ)	415
A.3.4	SM 1223 VDC 数字量输入/输出 (DI/DQ)	416
A.3.5	SM 1223 120/230 VAC 输入/继电器输出	418
A.4	数字量输入和输出的规范	420
A.4.1	24 VDC 数字量输入 (DI)	420
A.4.2	120/230 VAC 数字量 AC 输入	422
A.4.3	数字量输出 (DQ)	423
A.5	模拟量 I/O 模块	429
A.5.1	SB 1231 和 SB 1232 模拟量输入 (AI) 和输出 (AQ)	429
A.5.2	SM 1231 模拟量输入 (AI)	430
A.5.3	SM 1232 模拟量输出 (AQ)	431
A.5.4	SM 1234 模拟量输入/输出 (AI/AQ)	431
A.5.5	SM 1231 (AI)、SM 1232 (AQ) 和 SM 1234 (AI/AQ) 的接线图	432
A.6	BB 1297 电池板	433
A.7	模拟量 I/O 的规范	434
A.7.1	模拟量输入 (CPU、SM 和 SB) 的规范	434
A.7.2	输入 (AI) 的电压和电流测量范围	435
A.7.3	模拟量输入 (AI) 的阶跃响应	437
A.7.4	模拟量输入的采样时间和更新时间	438
A.7.5	模拟量输出的规范	439
A.7.6	输出 (AQ) 的电压和电流测量范围	440
A.8	RTD 模块和热电偶模块	442
A.8.1	SB 1231 RTD 和 SB 1231 TC 规范	442
A.8.2	SM 1231 RTD 规范	444
A.8.3	SM 1231 TC 规范	445
A.8.4	RTD 和 TC (SM 和 SB) 的模拟量输入规范	447
A.8.5	热电偶类型	448
A.8.6	热电偶滤波器选型和更新时间	449
A.8.7	RTD 传感器类型选型表	450
A.8.8	RTD 滤波器选型和更新时间	452
A.9	通信接口	453
A.9.1	PROFIBUS 主站/从站	453

A.9.1.1	CM 1242-5 PROFIBUS DP 从站	453
A.9.1.2	CM 1242-5 的 D 型插座的引脚分配.....	454
A.9.1.3	CM 1243-5 PROFIBUS DP 主站	455
A.9.1.4	PROFIBUS 主站 (CM 1243-5) 需要 CPU 的 24 VDC 电源	456
A.9.1.5	CM 1243-5 的 D 型插座的引脚分配.....	457
A.9.2	GPRS CP	458
A.9.2.1	CP 1242-7 GPRS	458
A.9.2.2	GSM/GPRS 天线 ANT794-4MR	460
A.9.2.3	平头天线 ANT794-3M	461
A.9.3	远程服务 (TS).....	462
A.9.4	RS485、RS232 和 RS422 通信	462
A.9.4.1	CB 1241 RS485 规范.....	462
A.9.4.2	CM 1241 RS422/485 技术规范.....	465
A.9.4.3	CM 1241 RS232 规范.....	467
A.10	工艺模块.....	469
A.10.1	SM 1278 4xIO-Link 主站 SM	469
A.10.1.1	SM 1278 4xIO-Link 主站信号模块规范	469
A.10.1.2	SM 1278 4xIO-Link 主站 SM 接线图	472
A.11	随附产品.....	474
A.11.1	PM 1207 电源模块	474
A.11.2	CSM 1277 紧凑型交换机模块.....	474
A.11.3	CM CANopen 模块	475
B	用 V4.1 CPU 更换 V3.0 CPU	477
B.1	用 V4.1 CPU 更换 V3.0 CPU.....	477
	索引	485

强大灵活的 S7-1200 简介

1.1 S7-1200 PLC 简介

S7-1200

控制器使用灵活、功能强大，可用于控制各种各样的设备以满足您的自动化需求。S7-1200

设计紧凑、组态灵活且具有功能强大的指令集，这些特点的组合使它成为控制各种应用的完美解决方案。

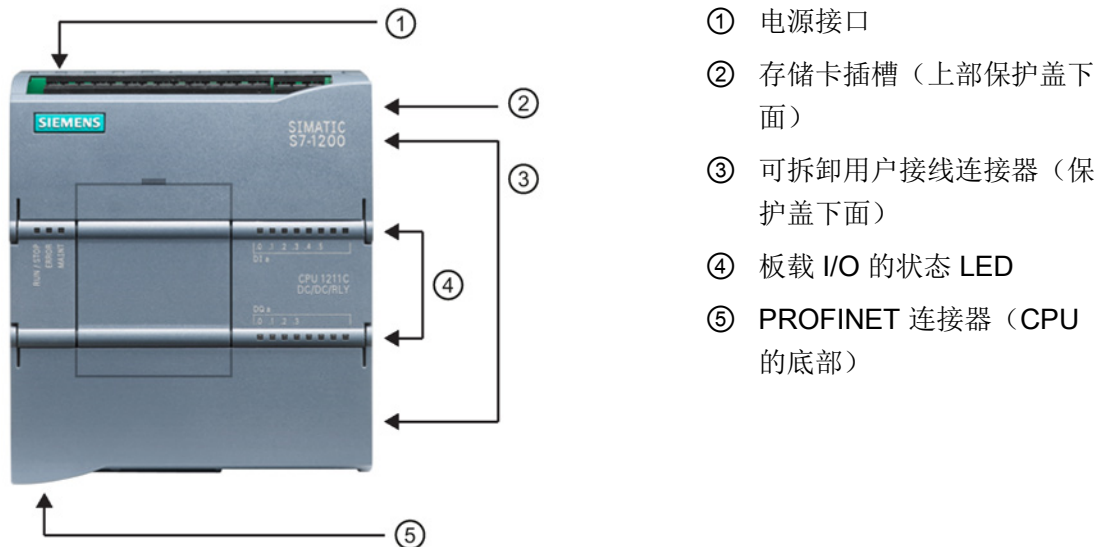
CPU 将微处理器、集成电源、输入和输出电路、内置 PROFINET、高速运动控制 I/O 以及板载模拟量输入组合到一个设计紧凑的外壳中来形成功能强大的控制器。

在您下载用户程序后，CPU 将包含监控应用中的设备所需的逻辑。CPU

根据用户程序逻辑监视输入并更改输出，用户程序可以包含布尔逻辑、计数、定时、复杂数学运算以及与其它智能设备的通信。

1.1 S7-1200 PLC 简介

CPU 提供一个 PROFINET 端口用于通过 PROFINET 网络通信。还可使用附加模块通过 PROFIBUS、GPRS、RS485、RS232、IEC、DNP3 和 WDC 网络进行通信。



有多种安全功能可用于保护对 CPU 和控制程序的访问：

- 每个 CPU 都提供密码保护 (页 93)功能，用户可以通过该功能组态对 CPU 功能的访问权限。
- 可以使用“专有技术保护” (页 95)隐藏特定块中的代码。
- 可以使用复制保护 (页 96)将程序绑定到特定存储卡或 CPU。

表格 1-1 CPU 型号的比较

特征		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
物理尺寸 (mm)		90 x 100 x 75		110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
用户存储器	工作	50 KB	75 KB	100 KB	125 KB	150 KB
	负载	1 MB		4 MB		
	保持性	10 KB				
本地板载 I/O	数字量	6 点输入/4 点输出	8 点输入/6 点输出	14 点输入/10 点输出		
	模拟量	2 路输入			2 点输入/2 点输出	
过程映像大小	输入 (I)	1024 个字节				
	输出 (Q)	1024 个字节				

特征	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
位存储器 (M)	4096 个字节		8192 个字节		
信号模块 (SM) 扩展	无	2	8		
信号板 (SB)、电池板 (BB) 或通信板 (CB)	1				
通信模块 (CM) (左侧扩展)	3				
高速计数器	总计	最多可组态 6 个使用任意内置或 SB 输入的高速计数器			
	1 MHz	-			lb.2 到 lb.5
	100/180 kHz	la.0 到 la.5			
	30/120 kHz	--	la.6 到 la.7	la.6 到 lb.5	la.6 到 lb.1
	200 kHz ³				
脉冲输出 ²	总计	最多可组态 4 个使用任意内置或 SB 输出的脉冲输出			
	1 MHz	--			Qa.0 到 Qa.3
	100 kHz	Qa.0 到 Qa.3			Qa.4 到 Qb.1
	20 kHz	--	Qa.4 到 Qa.5	Qa.4 到 Qb.	--
存储卡	SIMATIC 存储卡 (选项)				
实时时钟保持时间	通常为 20 天, 40°C 时最少为 12 天 (免维护超级电容)				
PROFINET 以太网通信端口	1			2	
实数数学运算执行速度	2.3 μs/指令				
布尔运算执行速度	0.08 μs/指令				

1 将 HSC 组态为正交工作模式时, 可应用较慢的速度。

2 对于具有继电器输出的 CPU 模块, 必须安装数字量信号 (SB) 才能使用脉冲输出。

3 与 SB 1221 DI x 24 VDC 200 kHz 和 SB 1221 DI 4 x 5 VDC 200 kHz 一起使用时最高可达 200 kHz。

不同的 CPU

型号提供了各种各样的特征和功能, 这些特征和功能可帮助用户针对不同的应用创建有效的解决方案。有关特定 CPU 的详细信息, 请参见技术规范 (页 391)。

1.1 S7-1200 PLC 简介

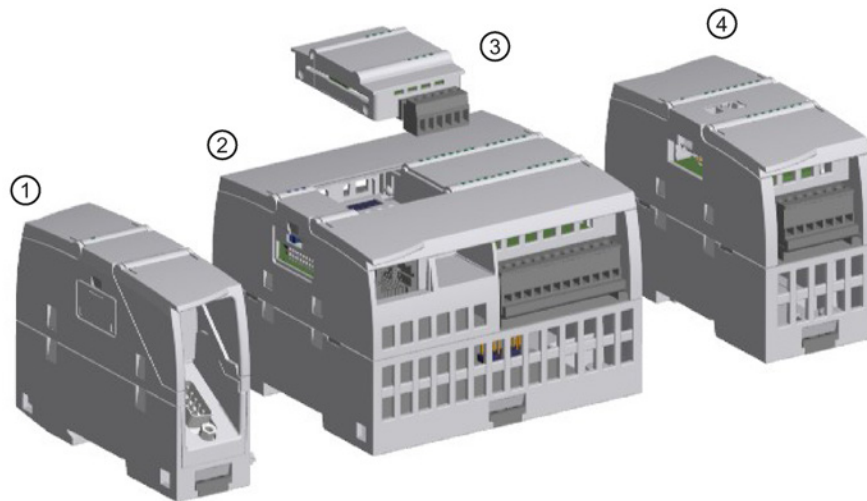
表格 1-2 S7-1200 支持的块、定时器和计数器

元素		说明
块	类型	OB、FB、FC、DB
	大小	50 KB (CPU 1211C) 75 KB (CPU 1212C) 100 KB (CPU 1214C) 125 KB (CPU 1215C) 150 KB (CPU 1217C)
	数量	最多可达 1024 个块 (OB + FB + FC + DB)
	嵌套深度	16 (从程序循环 OB 或启动 OB 开始) ; 6 (从任意中断事件 OB 开始)
	监视	可以同时监视 2 个代码块的状态
OB	程序循环	多个
	启动	多个
	延时中断	4 (每个事件 1 个)
	循环中断	4 (每个事件 1 个)
	硬件中断	50 (每个事件 1 个)
	时间错误中断	1
	诊断错误中断	1
	拔出或插入模块	1
	机架或站故障	1
	日时钟	多个
	状态	1
	更新	1
	配置文件	1
定时器	类型	IEC
	数量	仅受存储器大小限制
	存储	DB 结构, 每个定时器 16 个字节
计数器	类型	IEC
	数量	仅受存储器大小限制

元素		说明
	存储	DB 结构，大小取决于计数类型 <ul style="list-style-type: none"> • SInt 和 USInt: 3 个字节 • Int 和 UInt: 6 个字节 • DInt 和 UDInt: 12 个字节

1.2 CPU 的扩展功能

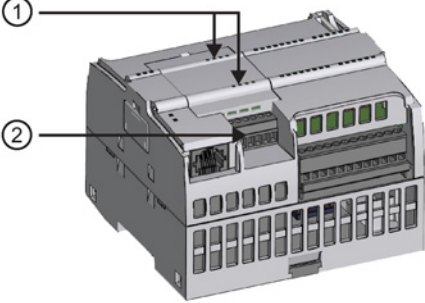
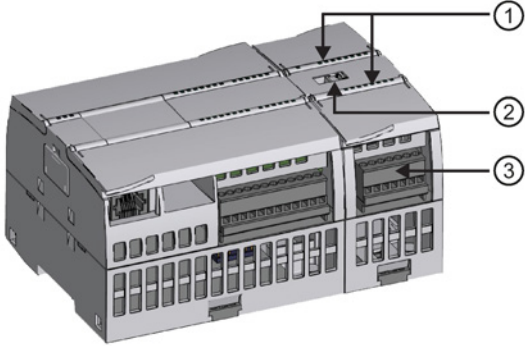
S7-1200 系列提供了各种模块和插入式板，用于通过附加 I/O 或其它通信协议来扩展 CPU 的功能。有关特定模块的详细信息，请参见技术规范 (页 391)。

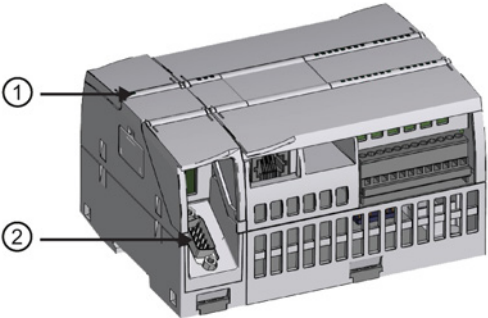


- ① 通信模块 (CM) 或通信处理器 (CP)
- ② CPU (CPU 1211C、CPU 1212C、CPU 1214C、CPU 1215C、CPU 1217C)
- ③ 信号板 (SB) (数字 SB、模拟 SB)，通信板 (CB) 或 电池板 (BB) CPU (CPU 1211C、CPU 1212C、CPU 1214C、CPU 1215C、CPU 1217C)
- ④ 信号模块 (SM) (数字 SM、模拟 SM、热电偶 SM、RTD SM、工艺 SM)

1.3 S7-1200 模块

表格 1-3 S7-1200 扩展模块

模块类型	说明
<p>CPU 支持一个插入式扩展板：</p> <ul style="list-style-type: none"> • 信号板 (SB) 可为 CPU 提供附加 I/O。SB 连接在 CPU 的前端。 • 通信板 (CB) 可以为 CPU 增加其它通信端口。 • 电池板 (BB) 可提供长期的实时时钟备份。 	<div style="text-align: center;">  </div> <p>① SB 上的状态 LED</p> <p>② 可拆卸用户接线连接器</p>
<p>信号模块 (SM) 可以为 CPU 增加其它功能。SM 连接在 CPU 右侧。</p> <ul style="list-style-type: none"> • 数字量 I/O • 模拟量 I/O • RTD 和热电偶 • SM 1278 IO-Link 主站 	<div style="text-align: center;">  </div> <p>① 状态 LED</p> <p>② 总线连接器滑动接头</p> <p>③ 可拆卸用户接线连接器</p>

模块类型	说明
<p>通信模块 (CM) 和通信处理器 (CP) 将增加 CPU 的通信选项, 例如 PROFIBUS 或 RS232/RS485 的连接性 (适用于 PtP、Modbus 或 USS) 或者 AS-i 主站。</p> <p>CP 可以提供其它通信类型的功能, 例如通过 GPRS、IEC、DNP3 或 WDC 网络连接到 CPU。</p>	
<ul style="list-style-type: none"> • CPU 最多支持三个 CM 或 CP • 各 CM 或 CP 连接在 CPU 的左侧 (或连接到另一 CM 或 CP 的左侧) 	<p>① 状态 LED</p> <p>② 通信连接器</p>

1.4 HMI 基本型面板

SIMATIC HMI 基本型面板提供了触屏式设备，用于执行基本的操作员监控任务。所有面板的保护等级均为 IP65 并通过了 CE、UL、cULus 和 NEMA 4x 认证。

可用的基本型 HMI 面板如下所述：

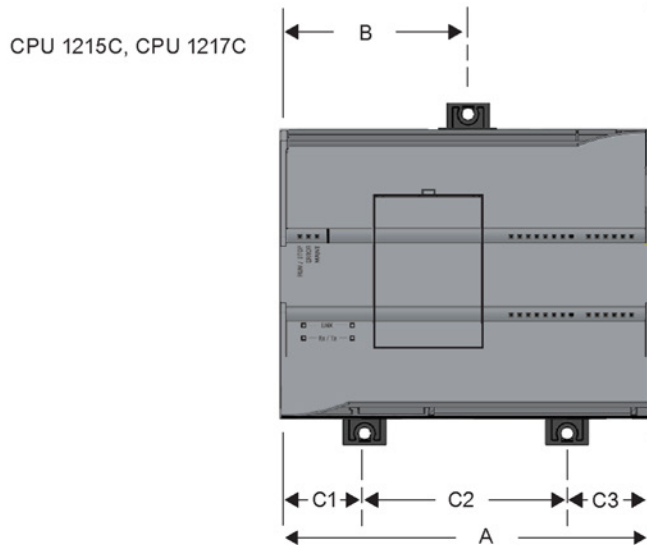
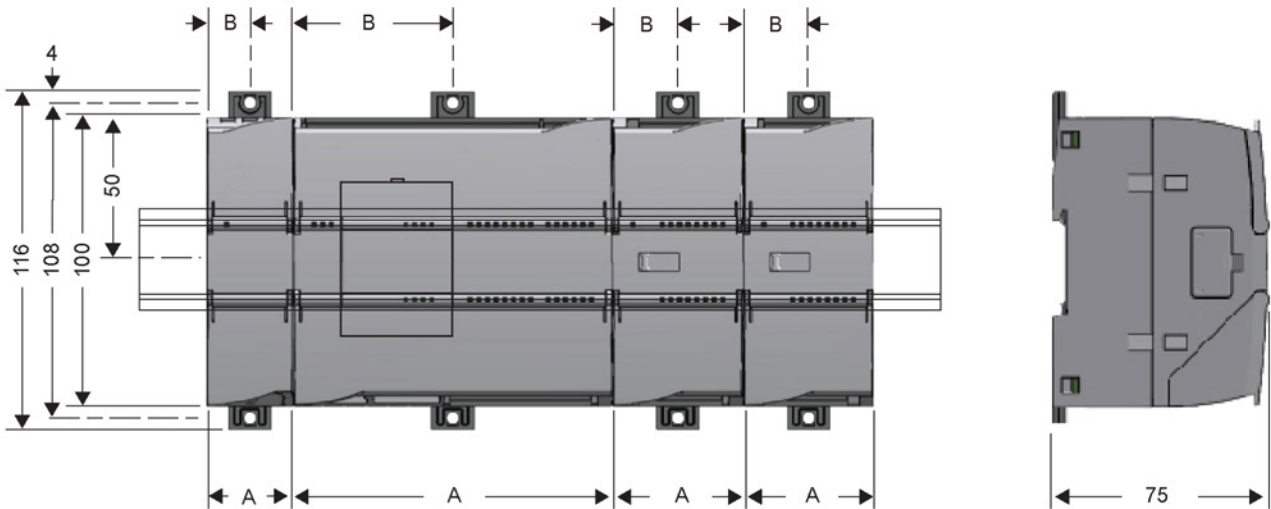
- **KTP400 Basic:** 4" 触摸屏，带 4 个可组态按键，分辨率为 480 x 272，800 个变量
- **KTP700 Basic:** 7" 触摸屏，带 8 个可组态按键，分辨率为 800 x 480，800 个变量
- **KTP700 Basic DP:** 7" 触摸屏，带 8 个可组态按键，分辨率为 800 x 480，800 个变量
- **KTP900 Basic:** 9" 触摸屏，带 8 个可组态按键，分辨率为 800 x 480，800 个变量
- **KTP1200 Basic:** 12" 触摸屏，带 10 个可组态按键，分辨率为 800 x 480，800 个变量
- **KTP 1200 Basic DP:** 12" 触摸屏，带 10 个可组态按键，分辨率为 800 x 400，800 个变量

1.5 安装尺寸和间隙要求

S7-1200 PLC 设计得易于安装。无论安装在面板上还是标准 DIN 导轨上，其紧凑型设计都有利于有效利用空间。

有关具体的安装要求和指南，请参见“S7-1200 可编程控制器系统手册”。

CPU 1211C, CPU 1212C, CPU 1214C
(measurements in mm)



1.5 安装尺寸和间隙要求

表格 1-4 安装尺寸 (mm)

S7-1200 设备		宽度 A (mm)	宽度 B (mm)	宽度 C (mm)
CPU	CPU 1211C 和 CPU 1212C	90	45	--
	CPU 1214C	110	55	--
	CPU 1215C	130	65 (顶部)	底部: C1: 32.5 C2: 65 C3: 32.5
	CPU 1217C	150	75	底部: C1: 37.5 C2: 75 C3: 37.5
信号模块	数字 8 和 16 点 模拟 2、4 和 8 点 热电偶 4 和 8 点 RTD 4 点 SM 1278 IO Link 主站	45	22.5	--
	数字量 DQ 8 x 继电器 (切换)	70	35	--
	模拟 16 点 RTD 8 点	70	35	--

S7-1200 设备		宽度 A (mm)	宽度 B (mm)	宽度 C (mm)
通信接口	CM 1241 RS232 和 CM 1241 RS422/485 CM 1243-5 PROFIBUS 主站和 CM 1242-5 PROFIBUS 从站 CM 1242-2 AS-i 主站 CP 1242-7 GPRS V2 CP 1243-7 LTE-EU CP 1243-1 DNP3 CP 1243-1 IEC CP 1243-1 CP1243-1 PCC CP 1243-8 ST7 RF120C	30	15	--
	TS (远程服务) Adapter IE Advanced ¹			
	TS (远程服务) Adapter IE Basic ¹	30	15	--
	TS 适配器	30	15	--
	TS 模块			

¹ 安装 TS (远程服务) Adapter IE Advanced 或 IE Basic 之前, 必须先连接 TS 适配器和 TS 模块。总宽度 (“宽度 A”) 为 60 mm。

每个 CPU、SM、CM 和 CP 都支持安装在 DIN 导轨或面板上。使用模块上的 DIN 导轨卡夹将设备固定到导轨上。

这些卡夹还能掰到一个伸出位置以提供将设备直接安装到面板上的螺钉安装位置。设备上 DIN 卡夹的安装孔内部尺寸是 4.3 mm。

必须在设备的上方和下方留出 25 mm 的发热区以便空气自由流通。

S7-1200 设备设计得易于安装。可以将 S7-1200 安装在面板或标准导轨上, 并且可以水平或垂直安装 S7-1200。S7-1200 尺寸较小, 用户可以有效地利用空间。

S7-1200 故障安全 CPU 不支持 PROFIBUS 或 PROFINET 分布式故障安全 I/O。

1.5 安装尺寸和间隙要求

电气设备标准将 SIMATIC S7-1200 系统分类为开放式设备。必须将 S7-1200 安装在外壳、控制柜或电控室内。

仅限获得授权的人员能打开外壳、控制柜或进入电控室。

安装时应为 S7-1200 提供干燥的环境。可以考虑使用 SELV/PELV 电路在干燥位置处提供电击防护。

安装时应按照适用的电气和建筑规范，为特定位置类别的开放式设备提供经过批准的机械和环境防护。

由于灰尘、潮湿和大气污染引起的导电性污染会导致 PLC 中发生操作和电气故障。

如果将 PLC 放在可能存在导电性污染的区域，必须采用具有适当保护等级的外壳对 PLC 实施保护。IP54

是常用于脏乱环境中电气设备外壳的一种保护等级，可能适合您的应用环境。



S7-1200 安装不当会导致发生电气故障或出现意外的机械操作。

电气故障或意外的机械操作可能会导致死亡、人员重伤和/或财产损失。
必须遵守适当操作环境的所有安装和维护说明以确保设备安全运行。

将 S7-1200 设备与热辐射、高压和电噪声隔离开

作为布置系统中各种设备的基本规则，必须将产生高压和高电噪声的设备与 S7-1200 等低压逻辑型设备隔离开。

在面板上配置 S7-1200

的布局时，请考虑发热设备并将电子式设备布置在控制柜中较凉爽区域。

少暴露在高温环境中会延长所有电子设备的使用寿命。

另外还要考虑面板中设备的布线。

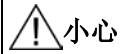
避免将低压信号线和通信电缆铺设在具有交流动力线和高能量快速开关直流线的槽中。

留出足够的空隙以便冷却和接线

S7-1200 被设计成通过自然对流冷却。

为保证适当冷却，在设备上方和下方必须留出至少 25 mm 的空隙。

此外，模块前端与机柜内壁间至少应留出 25 mm 的深度。



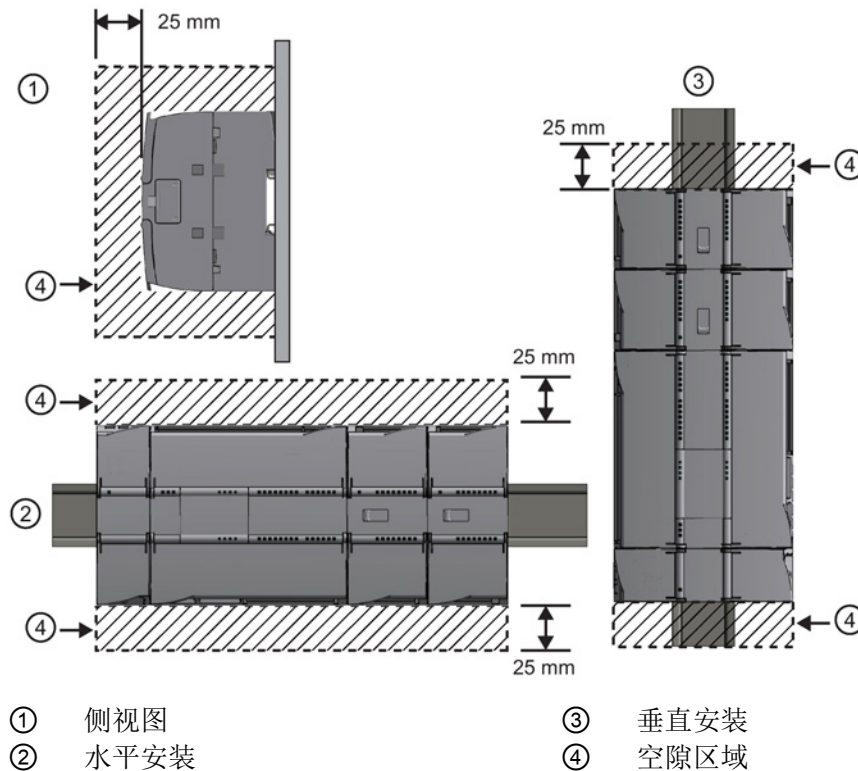
小心

对于纵向安装，允许的最大环境温度将降低 10°C。

请按下图所示调整垂直安装的 S7-1200 系统的方位。

确保正确安装 S7-1200 系统。

规划 S7-1200 系统的布局时，应留出足够的空隙以方便接线和通信电缆连接。





警告

安装或拆卸已上电的 S7-1200 或相关设备可能会导致电击或意外设备操作。

如果在安装或拆卸过程中没有断开 S7-1200 或相关设备的所有电源，则可能会由于电击或意外设备操作而导致死亡、人员重伤和/或财产损失。

务必遵守适当的安全预防措施，确保在尝试安装或拆卸 S7-1200 CPU 或相关设备前断开 S7-1200 的电源。

务必确保无论何时更换或安装 S7-1200 设备，都使用正确的模块或同等设备。



警告

S7-1200 模块安装不当可能会导致 S7-1200 中的程序工作异常。

如果更换 S7-1200 设备时使用的型号不同，或者更换后的方向或顺序与原来的不同，则可能会由于意外设备操作而导致死亡、人员重伤和/或财产损失。

请使用相同型号的设备来更换 S7-1200 设备，并确保设备的方向和位置放置正确。

1.6 新功能

以下是此版本中新增加的功能：

- 现在将 S7-1200 故障安全 CPU 和信号模块 (SM) 的硬件和固件与通过软件 (ES) 下载的安全程序结合使用，可以实现功能安全性。请参见《S7-1200 功能安全手册》(<http://support.automation.siemens.com/WW/view/en/104547552>)以获取更多信息。
- 固件版本为 V4.0 或更高版本的 S7-1200 CPU 的仿真：使用 S7-PLCSIM V13 SP1 可在仿真的 PLC 上测试 PLC 程序，而无需实际硬件。S7-PLCSIM 是一个单独安装的应用程序，在 TIA Portal 中与 STEP 7 结合使用。您可以在 STEP 7 中组态 PLC 和任何相关模块，编写应用程序逻辑，然后将硬件配置和程序下载到 S7-PLCSIM。然后可以使用 S7-PLCSIM 的工具来仿真和测试程序。请参见 S7-PLCSIM 的在线帮助以获取完整文档。请注意，不能仿真故障安全 CPU。
- 组态控制（选项处理）(页 85)：
您可以将硬件配置为最大程度的机器配置，包括在实际操作过程中可能用不到的模块。这些灵活模块的组态和标识是此版本的 STEP 7 和 S7-1200 新增的。即使没有标识的模块也不会导致错误状态。
- Web 服务器 (页 275)现在支持通过本地机架中所选模块（通信处理器）的 IP 地址以及 S7-1200 CPU 的 IP 地址进行访问。
- 增强的运动功能：
 - 模拟和 PROFIdrive 连接
 - 取模和控制环扩展参数
- 使用高速计数器 (HSC) (页 136) 进行周期测量
- SCL 编译器的性能提高
- 程序块与强制密码的动态防拷贝保护 (页 96)绑定
- 增强的 PROFINET 功能，包括支持共享设备。

1.6 新功能

- 新增编程指令：
 - EQ_Type、NE_Type、EQ_ElemType、NE_ElemType
 - IS_NULL、NOT_NULL
 - IS_ARRAY
 - Deserialize、Serialize
 - VariantGet、VariantPut、CountOfElements
 - Variant_to_DB_Any、DB_Any_To_Variant
 - GET_IM_DATA
 - RUNTIME
 - GEO2LOG、IO2MOD
 - ReadLittle、WriteLittle、ReadBig、WriteBig (仅限 SCL)
 - T_RESET、T_DIAG 和 TMAIL_C
 - PID_Temp
 - 新的 Modbus 指令 (页 200)
 - 新的点对点 (PtP) 指令 (页 197)
 - 新的 USS 指令 (页 198)

S7-1200 的新模块

各种新模块扩展了 S7-1200 CPU 的功能，因而能够灵活地满足您的自动化需要：

- 工业远程控制通信模块：可以将这些 CP 用作通信模块与 S7-1200 V4.1 CPU 配合使用。
- 故障安全 CPU 和 I/O：四个故障安全 CPU 和三个故障安全信号模块 (SM) 可与 S7-1200 V4.1 或更高版本配合使用：
 - CPU 1214FC DC/DC/DC (6ES7 214-1AF40-0XB0)
 - CPU 1214FC DC/DC/RLY (6ES7 214-1HF40-0XB0)
 - CPU 1215FC DC/DC/DC (6ES7 215-1AF40-0XB0)
 - CPU 1215FC DC/DC/RLY (6ES7 215-1HF40-0XB0)
 - SM 1226 F-DI 16 x 24 VDC (6ES7 226-6BA32-0XB0)
 - SM 1226 F-DQ 4 x 24 VDC (6ES7 226-6DA32-0XB0)
 - SM 1226 F-DQ 2 x Relay (6ES7 226-6RA32-0XB0)

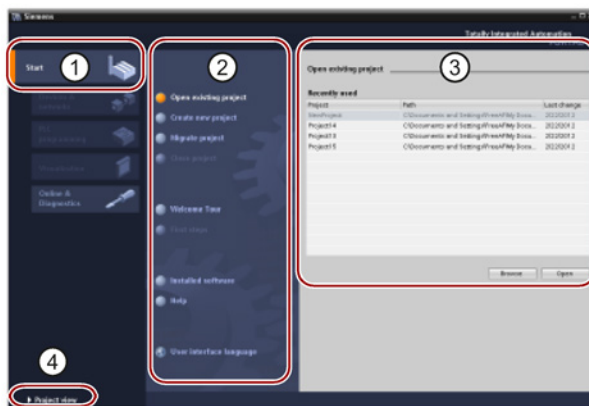
可以将同一系统中的 S7-1200 标准信号模块 (SM)、通信模块 (CM) 和信号板 (SB) 与故障安全 SM 配合使用，以完成不需要功能安全等级的应用控制功能。支持与故障安全 SM 一起使用的标准 SM 的部件编号为 (6ES7 --- ---32 0XB0) 或更高编号。

用 V4.1 CPU 更换 V3.0 CPU

如果将 S7-1200 V3.0 CPU 更换为 S7-1200 V4.1 CPU，请注意两个版本间记录的差异 (页 477)和所需的用户操作。

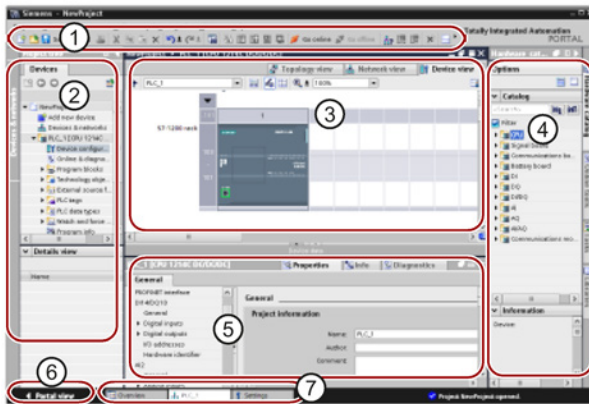
STEP 7 使工作更轻松

STEP 7 提供了一个用户友好的环境，供用户开发控制器逻辑、组态 HMI 可视化和设置网络通信。为帮助用户提高生产率，STEP 7 提供了两种不同的项目视图：根据工具功能组织的面向任务的门户集（门户视图），或项目中各元素组成的面向项目的视图（项目视图）。请选择能让您的工作最高效的视图。只需通过单击就可以切换门户视图和项目视图。



门户视图

- ① 不同任务的门户
- ② 所选门户的任务
- ③ 所选操作的选择面板
- ④ 切换到项目视图



项目视图

- ① 菜单和工具栏
- ② 项目浏览器
- ③ 工作区
- ④ 任务卡
- ⑤ 巡视窗口
- ⑥ 切换到门户视图
- ⑦ 编辑器栏

由于这些组件组织在一个视图中，所以您可以方便地访问项目的各个方面。例如，巡视窗口显示了用户在工作区中所选对象的属性和信息。当用户选择不同的对象时，巡视窗口会显示用户可组态的属性。巡视窗口包含用户可用于查看诊断信息和其它消息的选项卡。

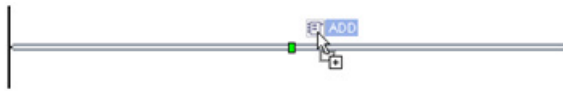
编辑器栏会显示所有打开的编辑器，从而帮助用户更快速和高效地工作。要在打开的编辑器之间切换，只需单击不同的编辑器。

2.1 轻松向用户程序中插入指令

还可以将两个编辑器垂直或水平排列在一起显示。
通过该功能可以在编辑器之间进行拖放操作。

2.1 轻松向用户程序中插入指令

STEP 7 提供了包含各种程序指令的任务卡。
这些指令按功能分组。



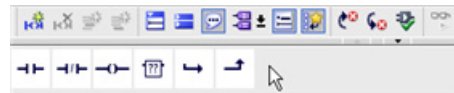
要创建程序，可将指令从任务卡拖动到程序段中。

基本指令	
名称	
▶	常规
▶	位逻辑运算
▶	定时器操作
▶	计数器操作
▶	比较器操作
▶	数学函数
▶	移动操作
▶	转换操作
▶	程序控制操作
▶	字逻辑运算
▶	移位和循环移位

2.2 从工具栏轻松访问收藏的指令

STEP 7 提供了“收藏夹”(Favorites) 工具栏，可供用户快速访问常用的指令。

只需单击指令的图标即可将其插入程序段！



(要访问指令树中的“收藏夹”，请双击该图标。)



用户可以通过添加新指令方便地自定义“收藏夹”(Favorites)。

只需将指令拖放到“收藏夹”(Favorites)。

如此单击即可插入该指令！



2.3 将输入或输出轻松地添加到 LAD 和 FBD 指令中

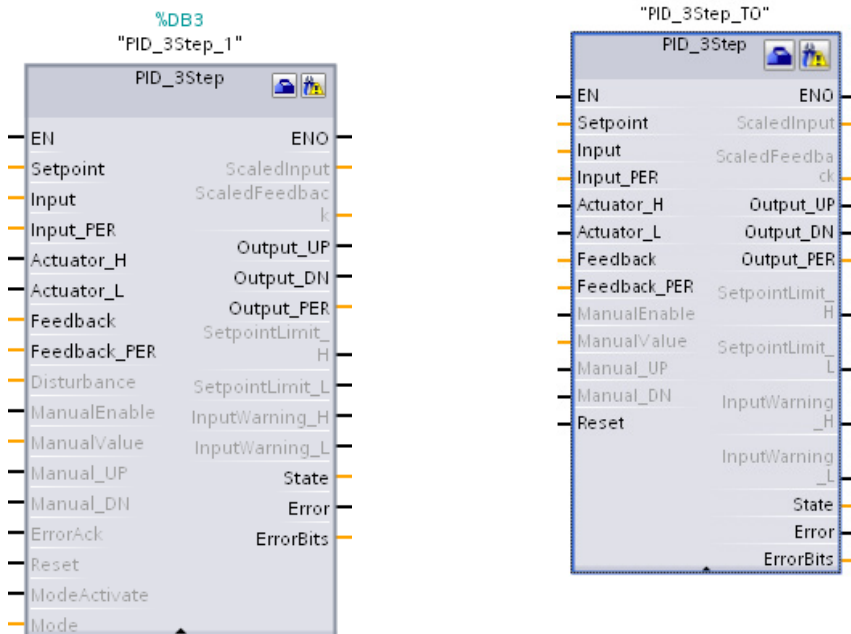


有些指令允许您另外创建输入或输出。

- 要添加输入或输出，请单击“创建”(Create) 图标，或在其中一个现有 IN 或 OUT 参数的输入短线处单击右键，并选择“插入输入”(Insert input) 命令。
- 要删除输入或输出，请在其中一个现有 IN 参数或 OUT 参数（原始输入多于两个时）的短线处单击右键，然后选择“删除”(Delete) 命令。

2.4 可扩展指令

一些更为复杂的指令是可扩展的，只显示主要输入和输出。
 要显示所有输入和输出，请单击指令底部的箭头。



2.5 轻松更改 CPU 的工作模式

该 CPU 没有用于更改工作模式（STOP 或 RUN）的物理开关。

请使用“启动 CPU”(Start CPU) 和“停止 CPU”(Stop CPU) 工具栏按钮更改 CPU 的工作模式。



在设备配置中组态 CPU 时，应组态 CPU 属性中的启动行为 (页 86)。

“在线和诊断”(Online and Diagnostics) 门户还提供了用于更改在线 CPU 工作模式的操作面板。要使用 CPU 操作员面板，必须在线连接到 CPU。“在线工具”(Online tools) 任务卡显示的操作员面板显示了在线 CPU 的工作模式。也可以通过该操作员面板更改在线 CPU 的工作模式。



使用操作员面板上的按钮更改工作模式（STOP 或 RUN）。操作员面板还提供了用于复位存储器的 MRES 按钮。

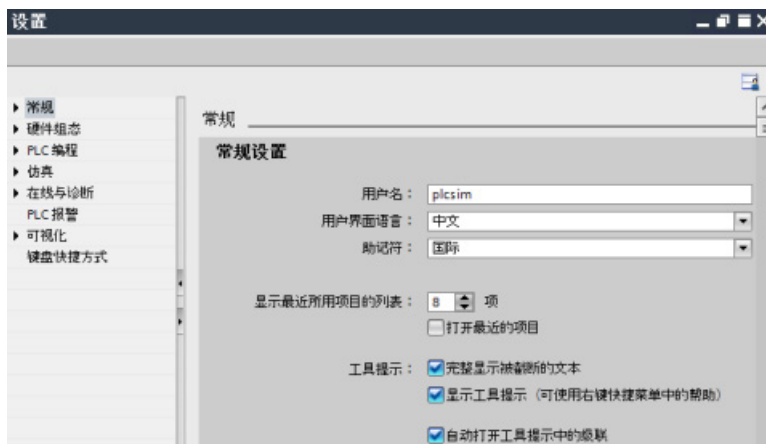
RUN/STOP 指示器的颜色指示 CPU 当前的工作模式。黄色表示 STOP 模式，而绿色表示 RUN 模式。

通过 STEP 7 中的设备组态，还可以在 CPU 上电时组态默认运行模式。

2.6 轻松修改 STEP 7 的外观和组态

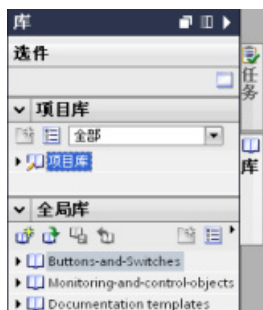
用户可以选择不同的设置，例如界面的外观、语言或项目的保存目录。

在“选项”(Options) 菜单中选择“设置”(Settings) 命令更改这些设置。



2.7 便于访问的项目库和全局库

通过全局库和项目库，可以在整个项目中或者在项目间重复使用所存储的对象。例如，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对其进行修改。可以在这些库中存储各种对象，例如 FC、FB、DB、设备配置、数据类型、监视表格、过程画面和面板。还可以将 HMI 设备的组件保存在项目中。



每个项目都有一个项目库，用于存储要在项目中多次使用的对象。该项目库是项目的一部分。

打开或关闭项目时，会相应地打开或关闭项目库；而保存项目时，也会相应地保存项目库中所做的任何更改。

用户可以创建自己的全局库，用于存储供其它项目使用的对象。创建新的全局库后，可将该库保存在计算机或网络中的某个位置。

2.8 便于选择指令版本

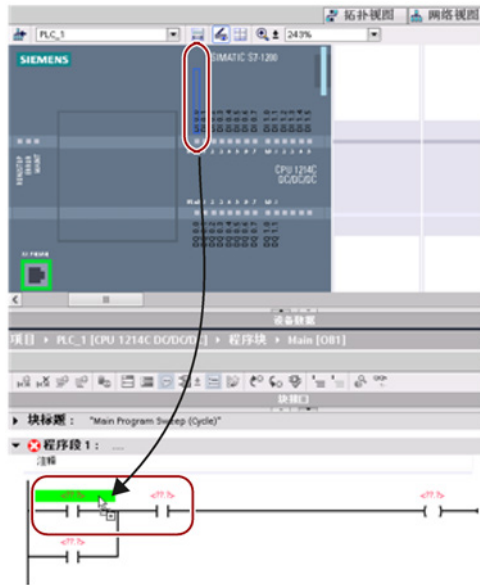
某些指令集（如 Modbus、PID 和运动指令集）经过多个开发和发布周期后形成了多种发布版本。为了有助于确保与较早项目的兼容性以及对这些项目进行移植，STEP 7 允许您选择要插入用户程序中的指令版本。



单击指令树任务卡上的图标可启用指令树的标题和列。

要更改指令版本，需从下拉列表中选择合适的版本。

2.9 在编辑器之间轻松拖放



为帮助用户快速方便地执行任务，STEP 7 允许用户将元素从一个编辑器拖放到另一个编辑器中。例如，可以将 CPU 的输入拖动到用户程序中指令的地址上。必须放大至少 200% 才能选中 CPU 的输入或输出。

请注意，变量名称不仅会在 PLC 变量表中显示，还会在 CPU 上显示。

要一次显示两个编辑器，请使用“拆分编辑器”(Split editor) 菜单命令或工具栏中的相应按钮。



要在已打开的编辑器之间切换，请单击编辑器栏中的图标。



2.10 更改 DB 的调用类型



STEP 7 允许您方便地创建或更改指令或 FB 的 DB 关联。

- 您可以在不同 DB 之间切换关联。
- 可以在单背景数据块与多背景数据块之间切换关联。
- 可以创建背景数据块（如果背景数据块丢失或不可用）。

可通过在程序编辑器中右键单击相关指令或 FB，或者通过选择“选项”(Options) 菜单中的“块调用”(Block call) 命令，来访问“更改调用类型”(Change call type) 命令。

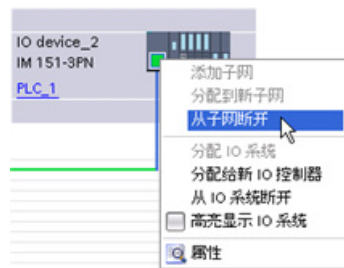
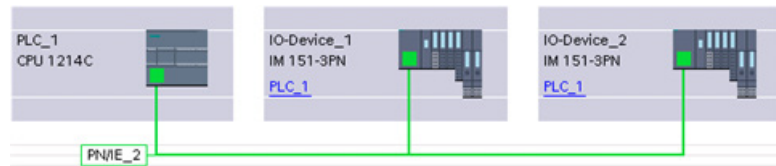


通过“调用选项”(Call options) 对话框可选择单背景数据块或多背景数据块。还可以从可用 DB 的下拉列表中选择具体 DB。

2.11 暂时从网络中断开设备

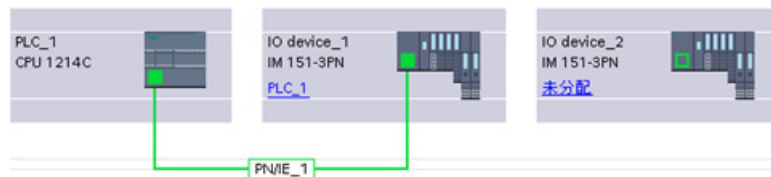
可以从子网断开网络设备。

由于不会从项目中删除相关设备的组态，因此可轻松恢复与设备的连接。



右键单击网络设备接口，然后从右键快捷菜单中选择“从子网断开”(Disconnect from subnet) 命令。

STEP 7 会重新组态网络连接，但不会从项目中删除断开的设备。删除该网络连接时，接口地址不会发生变化。



下载新的网络连接时，CPU 必须设置为 STOP 模式。

要重新连接设备，只需创建到设备端口的新网络连接。

2.12 轻松实现实际“拔出”模块而不会丢失组态数据



STEP 7

为“拔出的”模块提供了一个存储区域。用户可以从机架中拖出模块以保存该模块的组态。这些拔出的模块会随项目一同保存，从而在将来不必重新组态参数即可再次插入相应模块。

此功能的其中一种用途是用于临时维护。想想用户可能正等待一个替换模块，并计划临时使用一个不同的模块来短期替换相应模块。

此时，用户可以将组态的模块从机架拖动到“拔出的模块”(Unplugged modules) 区域，然后插入临时模块。

入门指南

3.1 创建项目

使用 STEP 7 非常容易！您可以看到开始创建项目有多么快捷。



在“开始”(Start)门户中，单击“创建新项目”(Create new project) 任务。输入项目名称并单击“创建”(Create) 按钮。



创建项目后，选择“设备和网络”(Devices & Networks) 门户。

单击“添加新设备”(Add new device) 任务。



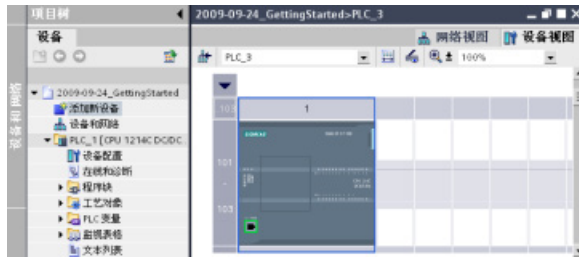
选择要添加到项目中的 CPU：

1. 在“添加新设备”(Add new device) 对话框中，单击“SIMATIC PLC”按钮。
2. 从列表中选择 CPU。
3. 单击“添加”(Add) 按钮，将所选 CPU 添加到项目中。

请注意，“打开设备视图”(Open device view) 选项已被选中。

在该选项被选中的情况下单击“添加”(Add) 将打开项目视图的“设备配置”(Device configuration)。

3.2 为 CPU 的 I/O 创建变量



设备视图显示所添加的 CPU。

3.2 为 CPU 的 I/O 创建变量

“PLC 变量”是 I/O 和地址的符号名称。用户创建 PLC 变量后，STEP 7 会将变量存储在变量表中。

项目中的所有编辑器（例如程序编辑器、设备编辑器、可视化编辑器和监视表格编辑器）均可访问该变量表。

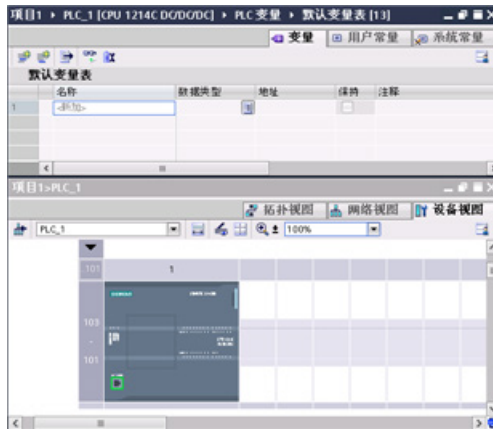


若设备编辑器已打开，请打开变量表。

您可在在编辑器栏中看到已打开的编辑器。



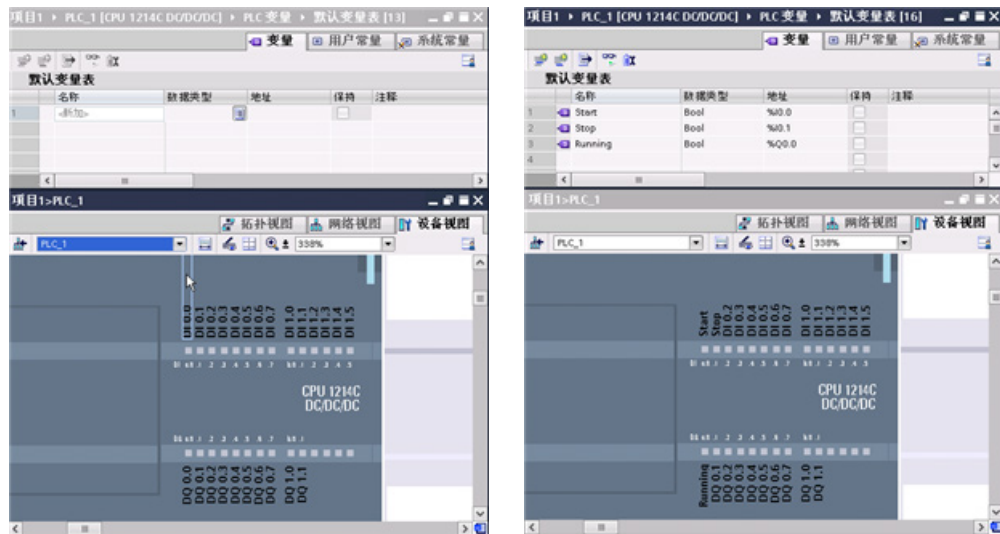
在工具栏中，单击“水平拆分编辑器空间”(Split editor space horizontally) 按钮。



STEP 7 将同时显示变量表和设备编辑器。

将设备配置放大 200% 以上，以便能清楚的查看并选择 CPU 的 I/O 点。将输入和输出从 CPU 拖动到变量表：

1. 选择 I0.0 并将其拖动到变量表的第一行。
2. 将变量名称从“I0.0”更改为“Start”。
3. 将 I0.1 拖动到变量表，并将名称更改为“Stop”。
4. 将 CPU 底部的 Q0.0 拖动到变量表，并将名称更改为“Running”。



将变量输入 PLC 变量表之后，即可在用户程序中使用这些变量。

3.3 在用户程序中创建一个简单程序段

程序代码由 CPU 依次执行的指令组成。在本实例中，使用梯形图 (LAD) 创建程序代码。LAD 程序是一系列类似梯级的程序段。



要打开程序编辑器，请按以下步骤操作：

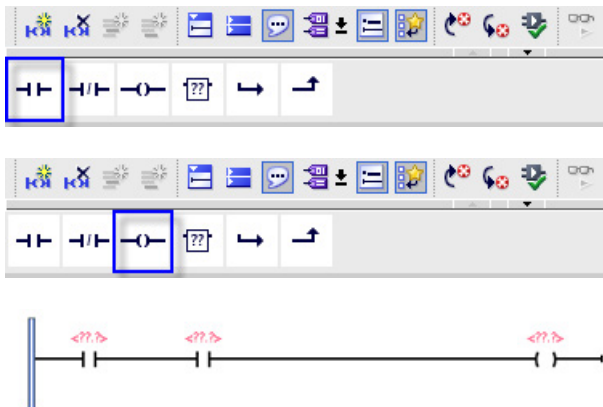
1. 在项目树中展开“程序块”(Program blocks)

文件夹以显示“Main [OB1]”块。

2. 双击“Main [OB1]”块。

程序编辑器将打开程序块 (OB1)。

使用“收藏夹”(Favorites) 上的按钮将触点和线圈插入程序段中。



1. 单击“收藏夹”(Favorites)

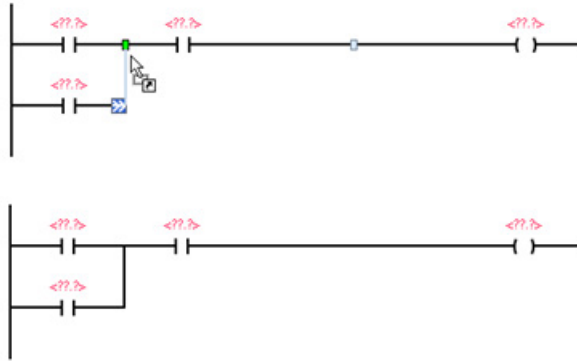
上的“常开触点”按钮向程序段添加一个触点。

2. 在本示例中，添加第二个触点。

3. 单击“输出线圈”(Output coil)

按钮插入一个线圈。

“收藏夹”(Favorites) 还提供了用于创建分支的按钮



1. 选择左侧的能流线，以指定分支的能流线。
2. 单击“打开分支”(Open branch) 图标向程序段的电源线添加分支。
3. 在打开的分支中插入另一个常开触点。
4. 将双向箭头拖动到第一梯级上两个触点之间的一个连接点位置（梯级上的绿色方块）。

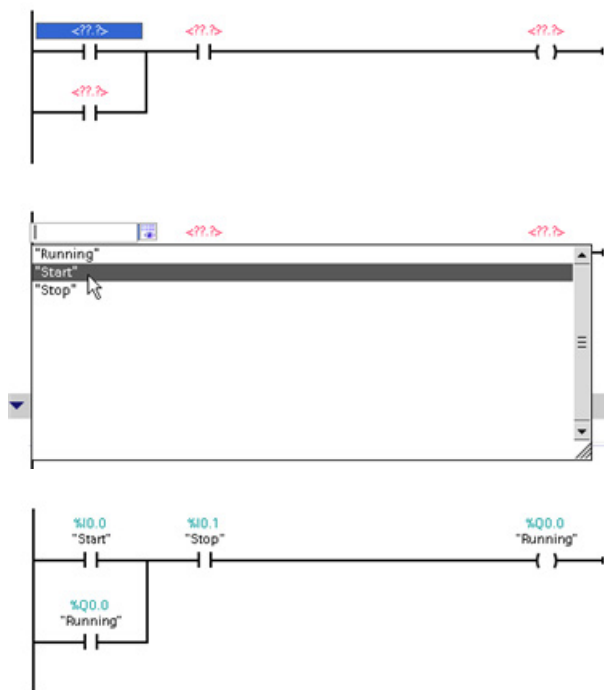
要保存项目，请单击工具栏中的“保存项目”(Save project) 按钮。

请注意，在保存前不必完成对梯级进行编辑。

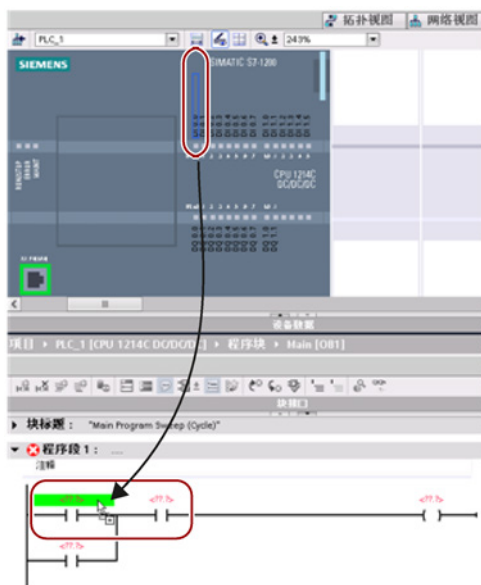
现在可以将变量名称与这些指令进行关联。

3.4 使用变量表中的 PLC 变量对指令进行寻址

使用变量表，用户可以快速输入对应触点和线圈地址的 PLC 变量。



1. 双击第一个常开触点上方的默认地址 <??.>。
2. 单击地址右侧的选择器图标打开变量表中的变量。
3. 从下拉列表中，为第一个触点选择“Start”。
4. 对于第二个触点，重复上述步骤并选择变量“Stop”。
5. 对于线圈和锁存触点，选择变量“Running”。



还可以直接从 CPU 中拖拽 I/O 地址。为此，只需拆分项目视图的工作区 (页 39)。必须将 CPU 放大 200% 以上才能选择 I/O 点。可以将“设备配置”(Device configuration) 中 CPU 上的 I/O 拖到程序编辑器的 LAD 指令上，这样不仅会创建指令的地址，还会在 PLC 变量表中创建相应条目。

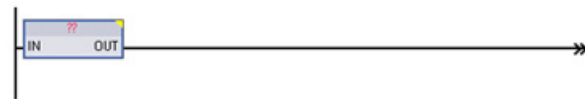
3.5 添加“功能框”指令

程序编辑器提供了一个通用“功能框”指令。

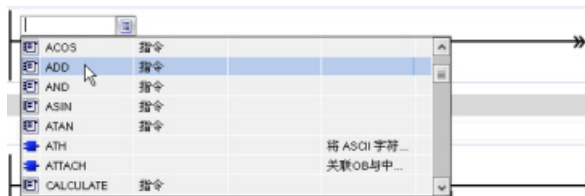
插入此功能框指令之后，可从下拉列表中选择指令类型，例如 **ADD** 指令。



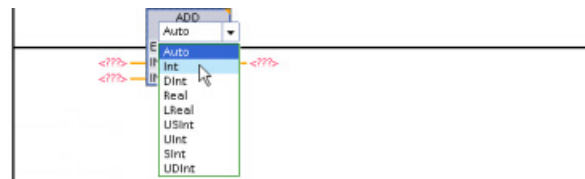
单击“收藏夹”(Favorites) 工具栏中的通用“功能框”指令。



通用“功能框”指令支持多种指令。
在本实例中，创建一个 **ADD** 指令：



1. 单击功能框指令黄色角以显示指令的下拉列表。
2. 向下滚动列表并选择 **ADD** 指令。
3. 单击“?”旁边的黄色角为输入和输出选择数据类型。



现在即可为 **ADD** 指令所用的值输入变量（或存储器地址）。



还可以为某些指令创建更多输入：

1. 单击框中的其中一个输入。
2. 单击右键以显示快捷菜单并选择“插入输入”(Insert input) 命令。

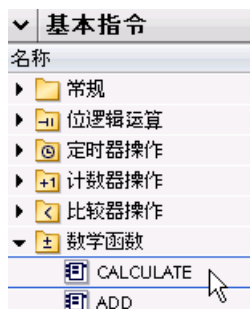


ADD 指令现在即使用三个输入。

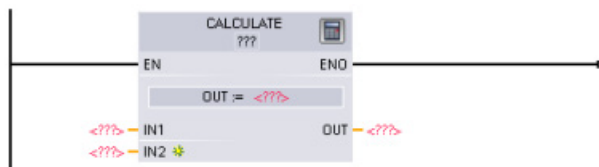
3.6 为复杂数学等式使用 CALCULATE 指令

Calculate 指令

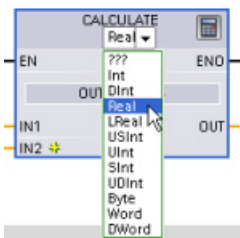
(页 118)可以根据定义的等式生成作用于多个输入参数的数学函数，从而生成结果。



在 Basic 指令树中，展开“数学函数”(Math functions) 文件夹。双击 Calculate 指令以将该指令插入用户程序中。



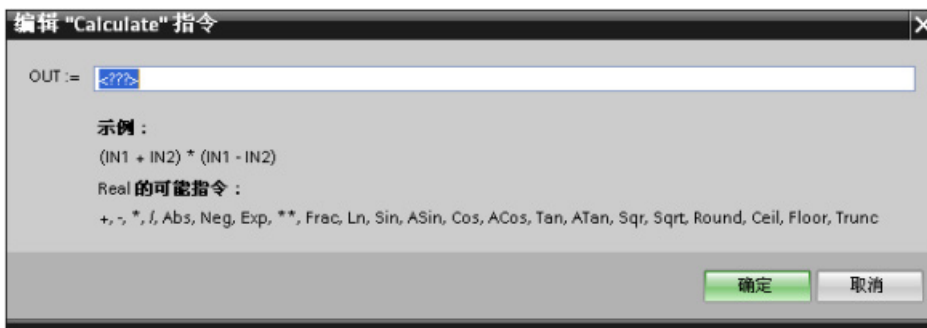
未组态的 Calculate 指令提供了两个输入参数和一个输出参数。



单击“???”并为输入参数和输出参数选择数据类型。
(所有输入参数和输出参数的数据类型必须相同。)
对于本示例，请选择“Real”数据类型。



单击“编辑等式”(Edit equation) 图标以输入等式。



对于本示例，请输入以下等式来标定原有模拟值。（“In”和“Out”标识对应于 Calculate 指令的参数。）

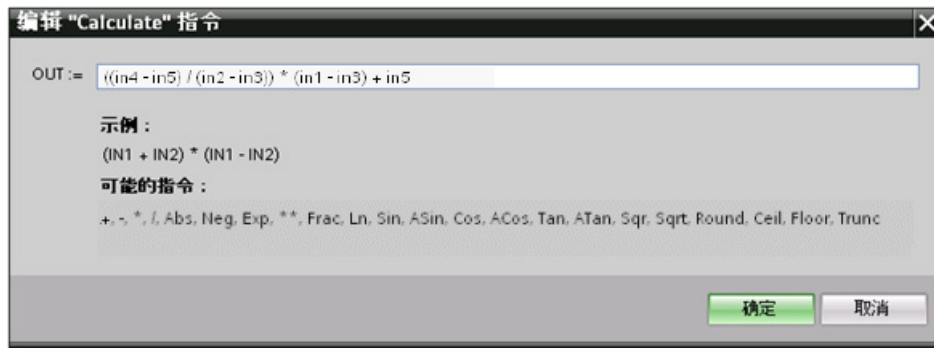
$$\text{Out}_{\text{value}} = ((\text{Out}_{\text{high}} - \text{Out}_{\text{low}}) / (\text{In}_{\text{high}} - \text{In}_{\text{low}})) * (\text{In}_{\text{value}} - \text{In}_{\text{low}}) + \text{Out}_{\text{low}}$$

$$\text{Out} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$

其中：	$\text{Out}_{\text{value}}$	(Out)	标定的输出值
	In_{value}	(in1)	模拟量输入值
	In_{high}	(in2)	标定输入值的上限
	In_{low}	(in3)	标定输入值的下限
	Out_{high}	(in4)	标定输出值的上限
	Out_{low}	(in5)	标定输出值的下限

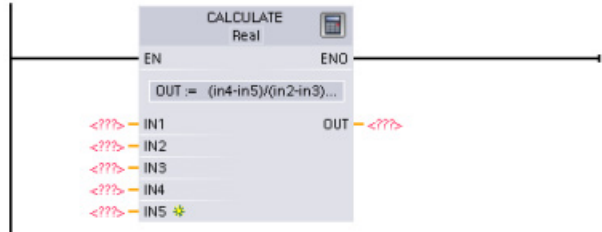
在“编辑 Calculate”(Edit Calculate) 框中，输入带有参数名称的等式：

$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$

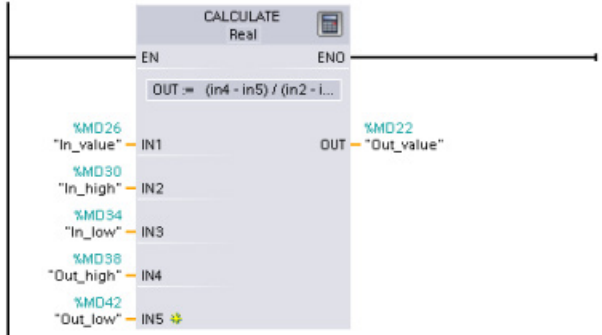


3.6 为复杂数学等式使用 *CALCULATE* 指令

单击“确定”(OK) 后，Calculate 指令就会生成指令所需的输入。



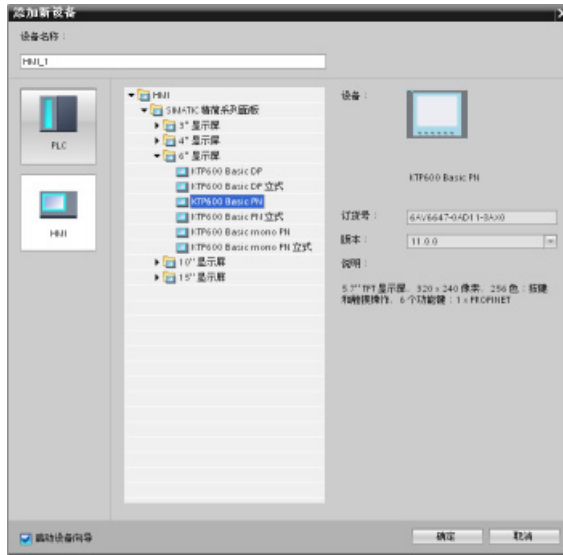
输入与参数对应的值的变量名称。



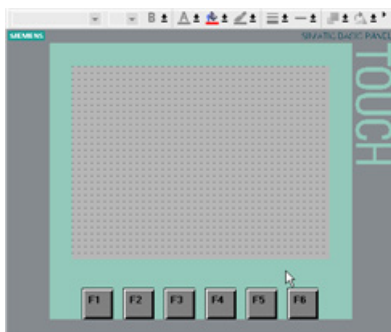
3.7 在项目中添加 HMI 设备



向项目中添加 HMI 设备非常容易！



1. 双击“添加新设备”(Add new device) 图标。
2. 在“添加新设备”(Add new device) 对话框中单击“SIMATIC HMI”按钮。
3. 从列表中选择特定的 HMI 设备。
可以运行 HMI 向导来组态 HMI 设备的画面。
4. 单击“确定”(OK) 将 HMI 设备添加到项目中。

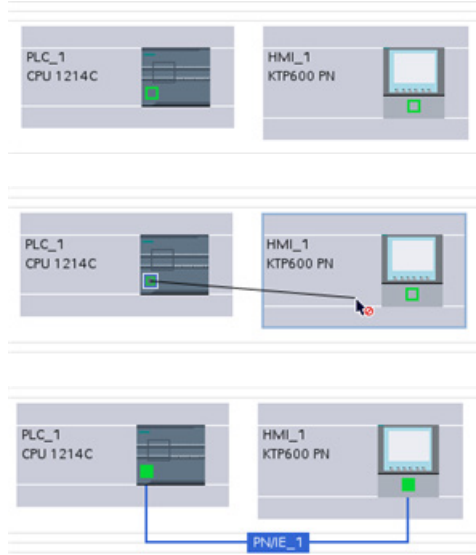


TIA Portal 将 HMI 设备添加到项目中。

TIA Portal 提供了一个 HMI 向导，可以帮助用户组态 HMI 设备的所有画面和结构。

如果未运行 HMI 向导，则 TIA Portal 将创建一个简单的默认 HMI 画面。
用户可以稍后在画面上添加其它画面或对象。

3.8 在 CPU 和 HMI 设备之间创建网络连接

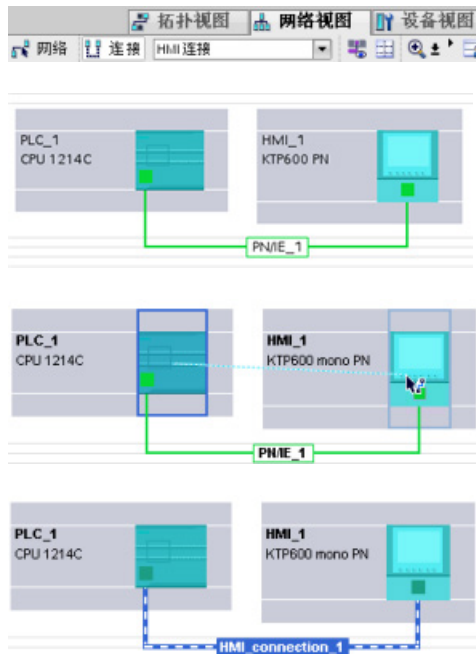


创建网络非常简单！

- 转到“设备和网络”(Devices and Networks) 并选择网络视图来显示 CPU 和 HMI 设备。
- 要创建 PROFINET 网络，只需从一个设备的绿色框拖出一条线连接到另一个设备的绿色框（以太网端口）。

随即会为这两个设备创建一个网络连接。

3.9 创建 HMI 连接以共享变量



通过在两个设备之间创建 HMI

连接，用户可以轻松地在两个设备之间共享变量。

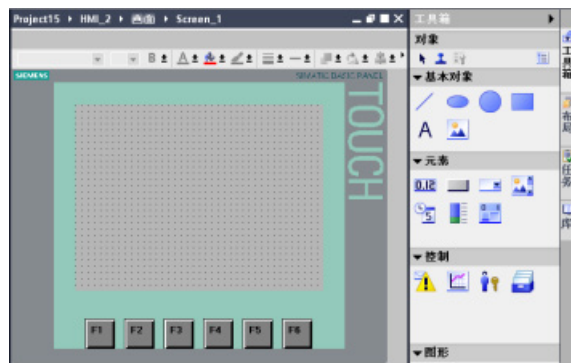
- 选择相应的网络连接，单击“连接”(Connections) 按钮并从下拉列表中选择“HMI 连接”(HMI connection)。
- HMI 连接会将相关的两个设备变为蓝色。
- 选择 CPU 设备并拖出一条线连接到 HMI 设备。
- 该 HMI 连接允许用户通过选择 PLC 变量列表对 HMI 变量进行组态。

用户可以采用其它方法创建 HMI 连接：

- 通过从 PLC 变量表、程序编辑器或设备配置编辑器将 PLC 变量拖动 HMI 画面编辑器，自动创建 HMI 连接。
- 通过使用 HMI 向导浏览到相应 PLC，自动创建 HMI 连接。

3.10 创建 HMI 画面

即使不利用 HMI 向导，组态 HMI 画面也很容易。

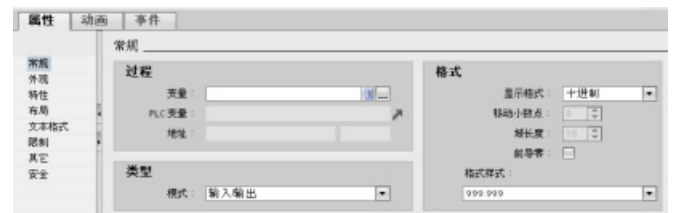
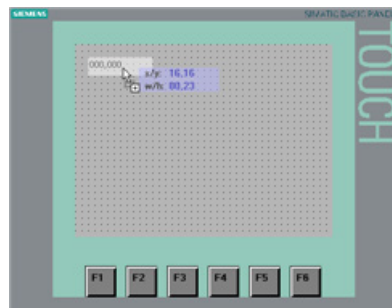


STEP 7

提供了一个标准库集合，用于插入基本形状、交互元素，甚至是标准图形。

要添加元素，只需将其中一个元素拖放到画面中。

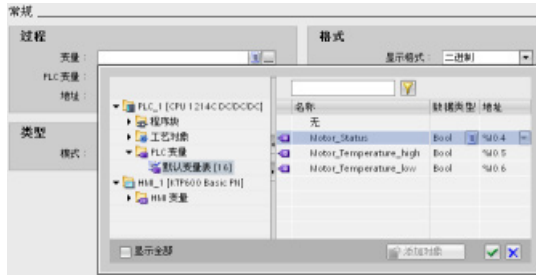
使用元素的属性（在巡视窗口中）组态该元素的外观和特性。



还可以通过从项目树或程序编辑器将 PLC 变量拖放到 HMI 画面来创建画面上的元素。PLC 变量即成为画面上的元素 然后可以使用属性来更改该元素的参数。

3.11 为 HMI 元素选择 PLC 变量

在画面上创建元素后，可使用元素的属性将 PLC 变量分配给该元素。单击变量字段旁的选择按钮来显示 CPU 的 PLC 变量。



也可以从项目树将 PLC 变量拖放到 HMI 画面中。在项目树的“详细信息”视图中显示 PLC 变量，然后将其拖放到 HMI 画面中。

简化了 PLC 概念

4.1 每个扫描周期均执行的任务

每个扫描周期都包括写入输出、读取输入、执行用户程序指令以及执行系统维护或后台处理。



该周期称为扫描周期或扫描。

在默认条件下，所有数字和模拟 I/O

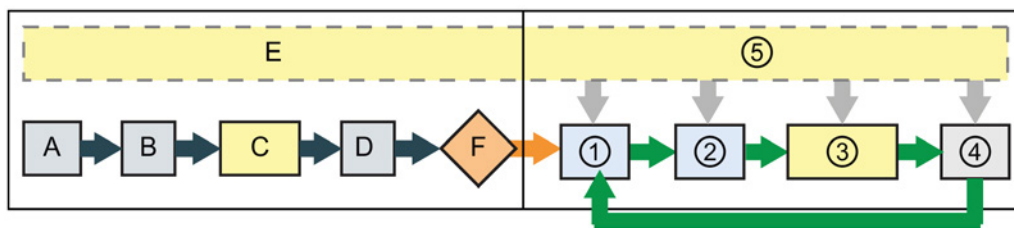
点都通过内部存储区（即过程映像）与扫描周期进行同步更新。过程映像包含

CPU、信号板和信号模块上的物理输入和输出的快照。

- CPU 仅在用户程序执行前读取物理输入，并将输入值存储在过程映像输入区。这样可确保这些值在整个用户指令执行过程中保持一致。
- CPU 执行用户指令逻辑，并更新过程映像输出区中的输出值，而不是写入实际的物理输出。
- 执行完用户程序后，CPU 将所生成的输出从过程映像输出区写入到物理输出。

这一过程通过在给定周期内执行用户指令而提供一致的逻辑，并防止物理输出点可能在过程映像输出区中多次改变状态而出现抖动。

4.1 每个扫描周期均执行的任务



STARTUP

- A 清除 I（图像）存储区
- B 使用组态的零、最后一个值或替换值初始化 Q 输出（图像）存储区，并将 PB、PN 和 AS-i 输出归零
- C 将非保持性 M 存储器和数据块初始化为初始值，并启用组态的循环中断和时间事件。
。 执行启动 OB。
- D 将物理输入的状态复制到 I 存储器
- E 将所有中断事件存储到要在进入 RUN 模式后处理的队列中
- F 启用将 Q 存储器写入到物理输出

RUN

- ① 将 Q 存储器写入物理输出
- ② 将物理输入的状态复制到 I 存储器
- ③ 执行程序循环 OB
- ④ 执行自检诊断
- ⑤ 在扫描周期的任何阶段处理中断和通信

可通过将模块从 I/O

的自动更新中删除来更改其默认行为。也可在执行指令时立即读取数字和模拟 I/O 值并将其写入模块。立即读取物理输入并不会更新过程映像输入区。立即写入物理输出会同时更新过程映像输出区和物理输出点。

4.2 CPU 的工作模式

CPU 有以下三种工作模式：STOP 模式、STARTUP 模式和 RUN 模式。CPU 前面的状态 LED 指示当前工作模式。

- 在 STOP 模式下，CPU 不执行任何程序，而用户可以下载项目。RUN/STOP LED 为黄色常亮。
- 在 STARTUP 模式下，CPU 会执行任何启动逻辑（如果存在）。在启动模式下，CPU 不会处理中断事件。RUN/STOP LED 为绿色和黄色交替闪烁。
- 在 RUN 模式下，扫描周期重复执行。在程序循环阶段的任何时刻都可能发生中断事件，CPU 也可以随时处理这些中断事件。用户可以在 RUN 模式下下载项目的某些部分。RUN/STOP LED 为绿色常亮。

CPU 支持通过暖启动方法进入 RUN 模式。暖启动不包括存储器复位，但可通过 STEP 7 控制存储器复位。存储器复位将清除所有工作存储器、保持性及非保持性存储区，将装载存储器内容复制到工作存储器，并将输出设置为组态的“对 CPU STOP 的响应”。存储器复位不会清除诊断缓冲区，也不会清除永久保存的 IP 地址。暖启动会初始化所有非保持性系统和用户数据。

可以使用 STEP 7 组态 CPU 的“上电后启动”设置以及重启方法。该组态项目出现在 CPU“设备配置”(Device Configuration) 的“启动”(Startup) 下。通电后，CPU 将执行一系列上电诊断检查和系统初始化操作。在系统初始化过程中，CPU 将删除所有非保持性位存储器，并将所有非保持性 DB 的内容重置为初始值。然后 CPU 进入适当的上电模式。某些错误会阻止 CPU 进入 RUN 模式。CPU 支持以下上电模式：STOP 模式、“暖启动后转到 RUN 模式”和“暖启动后转到上一个模式”。

注意

暖启动模式组态

CPU 因可修复故障或临时故障可能会进入 STOP

模式，前者如可替换信号模块故障，后者如电力线干扰或不稳定上电事件。

如果 CPU 已组态为“暖启动 - 断电前的模式”，则故障修复后，该 CPU 不会返回 RUN 模式，直到从 STEP 7 接收到进入 RUN 模式的新命令。没有新命令，该 STOP 模式将保持为断电前的模式。

一般情况下，STEP 7 连接中想要独立运行的 CPU 应组态为“暖启动 - RUN”，这样，移除故障条件并重新启动后，CPU 可以返回 RUN 模式。



CPU 未提供用于更改工作模式的物理开关。为更改 CPU 的工作模式，STEP 7 提供了以下工具：

- STEP 7 工具栏中的“停止”(Stop) 和“运行”(Run) 按钮
- 在线工具中的 CPU 操作面板

也可在程序中加入 STP 指令，以使 CPU 切换到 STOP 模式。这样就可以根据程序逻辑停止程序的执行。Web 服务器 (页 276)还提供用于更改运行模式的页面。

4.3 用户程序的执行

CPU 支持以下类型的代码块，使用它们可以创建有效的用户程序结构：

- 组织块 (OB) 定义程序的结构。有些 OB 具有预定义的行为和启动事件，但用户也可以创建具有自定义启动事件 (页 62)的 OB。
- 功能 (FC) 和功能块 (FB) 包含与特定任务或参数组合相对应的程序代码。每个 FC 或 FB 都提供一组输入和输出参数，用于与调用块共享数据。FB 还使用相关联的数据块（称为背景数据块）来保存执行期间程序中的其它块可使用的值状态。
- 数据块 (DB) 存储程序块可以使用的数据。

用户程序、数据及组态的大小受 CPU 中可用装载存储器和工作存储器 (页 15)的限制。对各个 OB、FC、FB 和 DB 块的数目没有特殊限制。但是块的总数限制在 1024 之内。

4.3.1 在 RUN 模式下处理扫描周期

在每个扫描周期中，CPU 都会写入输出、读取输入、执行用户程序、更新通信模块以及响应用户中断事件和通信请求。在扫描期间会定期处理通信请求。

以上操作（用户中断事件除外）按先后顺序定期进行处理。对于已启用的用户中断事件，将根据优先级按其发生顺序进行处理。对于中断事件，如果适用的话，CPU 将读取输入、执行 OB，然后使用关联的过程映像分区 (PIP) 写入输出。

系统要保证扫描周期在一定的时间段内（即最大循环时间）完成；否则将生成时间错误事件。

- 在每个扫描周期的开始，从过程映像重新获取数字量及模拟量输出的当前值，然后将其写入到 CPU、SB 和 SM 模块上组态为自动 I/O 更新（默认组态）的物理输出。通过指令访问物理输出时，输出过程映像和物理输出本身都将被更新。
- 随后在该扫描周期中，将读取 CPU、SB 和 SM 模块上组态为自动 I/O 更新（默认组态）的数字量及模拟量输入的当前值，然后将这些值写入过程映像。通过指令访问物理输入时，指令将访问物理输入的值，但输入过程映像不会更新。
- 读取输入后，系统将从第一条指令开始执行用户程序，一直执行到最后一条指令。其中包括所有的程序循环 OB 及其所有关联的 FC 和 FB。程序循环 OB 根据 OB 编号依次执行，OB 编号最小的先执行。

在扫描期间会定期处理通信请求，这可能会中断用户程序的执行。

自诊断检查包括定期检查系统和检查 I/O 模块的状态。

中断可能发生在扫描周期的任何阶段，并且由事件驱动。事件发生时，CPU 将中断扫描循环，并调用被组态用于处理该事件的 OB。OB 处理完该事件后，CPU 从中断点继续执行用户程序。

4.3.2 OB 可帮助用户构建用户程序

OB 控制用户程序的执行。CPU 中的特定事件将触发组织块的执行。OB 无法互相调用或通过 FC 或 FB 调用。只有诊断中断或时间间隔这类事件可以启动 OB 的执行。CPU 按优先等级处理 OB，即先执行优先级较高的 OB 然后执行优先级较低的 OB。最低优先等级为 1（对应主程序循环），最高优先等级为 26。

4.3.3 事件执行的优先级与排队

CPU 处理操作受事件控制。事件会触发要执行的中断 OB。
 可以在块的创建期间、设备配置期间或使用 ATTACH 或 DETACH 指令指定事件的中断 OB。有些事件定期发生，例如，程序循环或循环事件。
 而其它事件只发生一次，例如，启动事件和延时事件。
 还有一些事件则在硬件触发事件时发生，例如，输入点上的沿事件或高速计数器事件。
 诊断错误和时间错误等事件只在出现错误时发生。事件优先级和队列用于确定事件中断 OB 的处理顺序。

CPU 按照优先级顺序处理事件，1 为最低优先级，26 为最高优先级。在 S7-1200 CPU V4.0 之前的版本中，每种 OB 类型都有固定的优先级（1 到 26）。从 V4.0 开始，可为每个组态的 OB 分配优先级。优先级编号在 OB 属性的特性中进行配置。

可中断与不可中断执行模式

OB (页 61) 按照其触发事件的优先级顺序执行。从 V4.0 开始，可以将 OB 执行组态为可中断或不可中断。请注意，程序循环 OB 始终为可中断，但可将其它所有 OB 组态为可中断或不可中断。

如果设置了可中断模式，则在执行 OB 并且 OB 执行结束前发生了更高优先级的事件时，将中断正在运行的 OB，以允许更高优先级的事件 OB 运行。
 运行更高级别的事件直至结束后，才会继续执行之前中断的 OB。如果执行可中断 OB 时发生多个事件，CPU 将按照优先级顺序处理这些事件。

如果未设置可中断模式，则无论触发的 OB 在运行期间是否触发了其它任何事件，都将继续运行直至结束。

考虑以下两种情况，其中中断事件可触发循环 OB 和延时 OB。在这两种情况中，延时 OB (OB201) 没有过程映像分区分配并且以优先级 4 执行。循环 OB (OB200) 分配了 PIP1 过程映像分区并且以优先级 2 执行。下图显示了在不可中断与可中断执行模式中执行 OB 的区别：

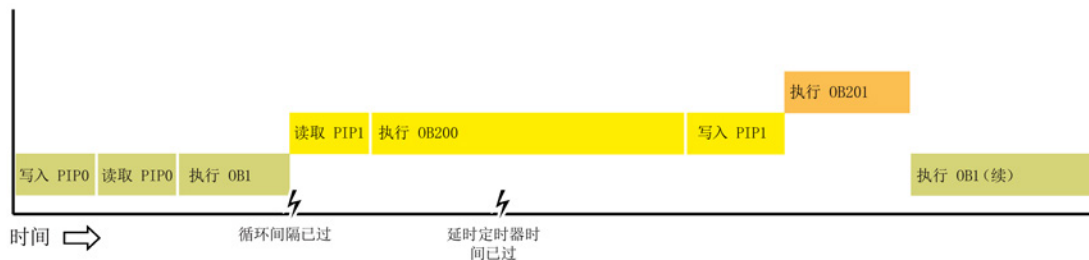


图 4-1 情况 1：不可中断 OB 执行

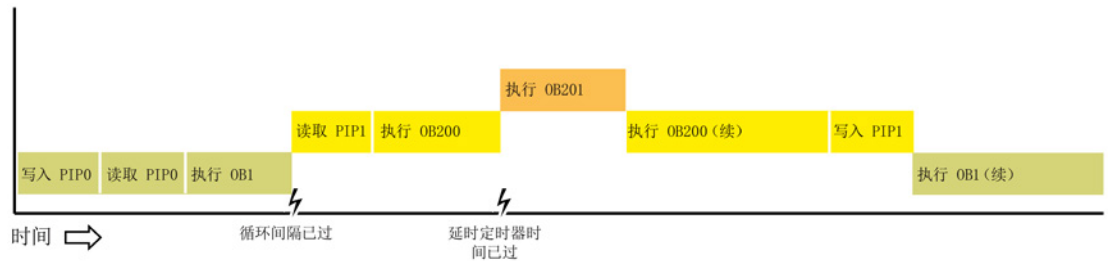


图 4-2 情况 2：可中断 OB 执行

说明

如果将 OB 执行模式组态为不可中断，则时间错误 OB 不能中断除程序循环 OB 以外的 OB。在 S7-1200 CPU V4.0 之前的版本中，时间错误 OB 可中断任何执行中的 OB。而从 V4.0 开始，如果要使时间错误 OB（或其它任何更高优先级 OB）可中断除程序循环 OB 以外的执行中 OB，必须将 OB 执行组态为可中断。

了解事件执行的优先级与排队

CPU 通过各种事件类型的不同队列限制单一来源的未决（排队的）事件数量。达到给定事件类型的未决事件限值后，下一个事件将丢失。可以使用时间错误中断 OB 响应队列溢出。

每个 CPU 事件都具有相关优先级。通常，CPU 按优先级顺序处理事件（优先级最高的最先进行处理）。对于优先级相同的事件，CPU 按照“先到先得”的原则进行处理。

表格 4-1 OB 事件

事件	允许的数量	默认 OB 优先级
程序循环	1 个程序循环事件 允许多个 OB	14
启动	1 个启动事件 ¹ 允许多个 OB	14
延时	最多 4 个时间事件 每个事件 1 个 OB	3
循环中断	最多 4 个事件 每个事件 1 个 OB	8

4.3 用户程序的执行

事件	允许的数量	默认 OB 优先级
硬件中断	最多 50 个硬件中断事件 ²	18
	每个事件 1 个 OB, 但可对多个事件使用同一个 OB	18
时间错误	1 个事件 (仅当组态时) ³	22 或 26 ⁴
诊断错误	1 个事件 (仅当组态时)	5
拔出或插入模块	1 个事件	6
机架或站故障	1 个事件	6
日时钟	最多 2 个事件	2
状态	1 个事件	4
更新	1 个事件	4
配置文件	1 个事件	4

- 1 启动事件和程序循环事件不会同时发生, 因为启动事件运行结束后程序循环事件才启动。
- 2 如果使用 DETACH 和 ATTACH 指令, 则可具有 50 个以上的硬件中断事件 OB。
- 3 可以将 CPU 组态为在超出最大扫描周期时间时保持 RUN 模式, 也可使用 RE_TRIGR 指令复位周期时间。但是, 如果同一个扫描周期第二次超出最大扫描周期时间, CPU 就会进入 STOP 模式。
- 4 新 V4.0 或 V4.1 CPU 的优先级为 22。如果是用 V3.0 CPU 更换的 V4.0 或 V4.1 CPU, 则优先级为 26: 对 V3.0 有效的优先级。无论哪种情况, 优先级字段都可以编辑, 用户可以将优先级设置为 22 到 26 之间的任何值。

有关详细信息, 请参见主题“用 V4.1 CPU 更换 V3.0 CPU (页 477)”。

另外, CPU 可识别出无关联 OB 的其它事件。下表介绍了这些事件和相应的 CPU 操作:

表格 4-2 附加事件

事件	说明	CPU 操作
I/O 访问错误	直接 I/O 读/写错误	CPU 将第一次错误记录在诊断缓冲区中并保持 RUN 模式。
最大周期时间错误	CPU 超出组态的周期时间两次	CPU 将错误记录在诊断缓冲区中并切换为 STOP 模式。

事件	说明	CPU 操作
外围设备访问错误	过程映像更新期间出现 I/O 错误	CPU 将第一次错误记录在诊断缓冲区中并保持 RUN 模式。
编程错误	程序执行错误	如果发生错误的块可进行错误处理，则更新错误结构；否则 CPU 会将错误记录在诊断缓冲区中并保持 RUN 模式。

中断等待时间

如果中断事件发生时程序循环 OB 是唯一激活的事件服务例程，则中断事件等待时间（该时间是指从通知 CPU 发生了事件到 CPU 开始执行处理该事件的 OB 中的第一条指令）约为 175 μ s。

4.4 存储区、寻址和数据类型

CPU 提供了以下用于存储用户程序、数据和组态的存储区：

- 装载存储器用于非易失性地存储用户程序、数据和组态。项目被下载到 CPU 后，首先存储在装载存储区中。该存储区位于存储卡（如存在）或 CPU 中。该非易失性存储区能够在断电后继续保持。
可以通过安装存储卡来增加数据日志的可用装载存储器的数量。
- 工作存储器是易失性存储器，用于在执行用户程序时存储用户项目的某些内容。CPU 会将一些项目内容从装载存储器复制到工作存储器中。
该易失性存储区将在断电后丢失，而在恢复供电时由 CPU 恢复。
- 保持性存储器用于非易失性地存储限量的工作存储器值。
保持性存储区用于在断电时存储所选用户存储单元的值。如果发生断电或掉电，CPU 将在上电时恢复这些保持性值。



可选的 SIMATIC

存储卡可用作存储用户程序的替代存储器，或用于传送程序。

如果使用存储卡，CPU

将运行存储卡中的程序而不是自身存储器中的程序。

检查以确定存储卡没有写保护。滑动保护开关，使其离开“Lock”位置。

将可选的 SIMATIC

存储卡用作程序卡或传送卡，以便收集数据日志文件或执行固件更新。

- 使用传送卡将项目复制到多个 CPU 中，而不使用 STEP 7。传送卡将存储的项目由卡中复制到 CPU 的存储器中。将程序复制到 CPU 后必须取出传送卡。
- 程序卡可以替代 CPU 存储器；所有 CPU 功能都由该程序卡进行控制。
插入程序卡会擦除 CPU 内部装载存储器的所有内容（包括用户程序和任何强制 I/O）。CPU 然后会执行程序卡中的用户程序。
- 还可以使用程序卡收集数据日志文件（页 129）。程序卡所能提供的存储器大于 CPU 的内部存储器。CPU 的 Web 服务器功能（页 275）允许您将数据日志文件下载到计算机中。
- 用户也可以使用存储卡执行固件更新。有关说明信息，请参见“S7-1200 可编程控制器系统手册”。

说明

程序卡**必须**保留在 CPU 中。如果取出程序卡，CPU 将切换到 STOP 模式。

4.4.1 S7-1200 支持的数据类型

数据类型用于指定数据元素的大小以及如何解释数据。

每个指令参数至少支持一种数据类型，而有些参数支持多种数据类型。

将光标停在指令的参数域上方，便可看到给定参数所支持的数据类型。

表格 4-3 S7-1200 支持的数据类型

数据类型	说明
位和位序列数据类型	<ul style="list-style-type: none"> • Bool 是布尔值或位值。 • Byte 是 8 位字节值。 • Word 是 16 位值。 • DWord 是 32 位双字节值。
整数数据类型	<ul style="list-style-type: none"> • USInt（无符号 8 位整数）和 SInt（有符号 8 位整数）可以是有符号或无符号的“短”整型（内存为 8 位或 1 个字节）。 • UInt（无符号 16 位整数）和 Int（有符号 16 位整数）可以是有符号或无符号的整型（内存为 16 位或 1 个字节）。 • UDInt（无符号 32 位整数）和 DInt（有符号 32 位整数）可以是有符号或无符号的双整型（内存为 32 位或 1 个双字节）。
实数数据类型	<ul style="list-style-type: none"> • Real 是 32 位实数或浮点值。 • LReal 是 64 位实数或浮点值。

4.4 存储区、寻址和数据类型

数据类型	说明
日期和时间数据类型	<ul style="list-style-type: none"> • Date 是包含自 1990 年 1 月 1 日开始算起的天数的 16 位日期值（与 UInt 类似）。最大日期值是 65378 (16#FF62)，该值与 2168 年 12 月 31 日相对应。所有可能的 Date 值都有效。 • DTL（日期和时间长度）是将有关日期和时间信息保存在预定义结构中的 12 字节结构。 <ul style="list-style-type: none"> - 年 (UInt): 1970 到 2554 - 月 (USInt): 1 到 12 - 日 (USInt): 1 到 31 - 工作日 (USInt): 1（星期日）到 7（星期六） - 小时 (USInt): 0 到 23 - 分 (USInt): 0 到 59 - 秒 (USInt): 0 到 59 - 纳秒 (UDInt): 0 到 999999999 • Time 是存储毫秒数（从 0 到 24 天 20 小时 31 分 23 秒 647 毫秒）的 32 位 IEC 时间值（与 Dint 类似）。所有可能的 Time 值都有效。Time 值可用于计算，可能得出负时间。 • TOD（日时钟）是包含从午夜算起的毫秒数（从 0 到 86399999）的 32 位日时钟值（与 Dint 类似）。
字符和字符串数据类型	<ul style="list-style-type: none"> • Char 是 8 位单个字符。 • String 是长度可达 254 个字符的可变长度字符串。
数组和结构数据类型	<ul style="list-style-type: none"> • Array 包含同一数据类型的多个元素。数组可以在 OB、FC、FB 和 DB 的块接口编辑器中创建。无法在 PLC 变量编辑器中创建数组。 • Struct 定义由其他数据类型组成的数据结构。Struct 数据类型可作为单个数据单元处理一组相关过程数据。 在数据块编辑器或块接口编辑器中声明 Struct 数据类型的名称和内部数据结构。 <p>数组和结构还可以集中到更大结构中。一套结构可嵌套八层。 例如，可以创建包含数组的多个结构组成的结构。</p>

数据类型	说明
PLC 数据类型	<p>PLC 数据类型是一种用户定义的数据结构，它定义用户可以在程序中多次使用的定制数据结构。创建 PLC 数据类型时，新的 PLC 数据类型会出现在 DB 编辑器和代码块接口编辑器中的数据类型选择器下拉列表中。</p> <p>可直接将 PLC 数据类型直接用作代码块接口或数据块中的数据类型。</p> <p>PLC 数据类型可用作生成使用相同数据结构的多个全局数据块的模板。</p>
指针数据类型	<ul style="list-style-type: none"> • Pointer 提供对变量地址的间接参考。它会在存储器中占用 6 个字节（48 位），可能包含以下变量信息：DB 号（或者当数据未存储在 DB 中时为 0）、CPU 中的存储区和存储器地址。 • Any 提供对数据区起始处的间接参考，并识别其长度。Any 指针使用存储器中的 10 个字节，可能包含以下信息： 数据元素的数据类型、数据元素数目、存储区或 DB 数以及数据的“Byte.Bit”起始地址。 • Variant 提供对不同数据类型或参数的变量的间接参考。Variant 指针识别结构和单独的结构组件。Variant 不会占用存储器的任何空间。

尽管以下

BCD（二进制编码的十进制数）数字格式不能用作数据类型，但它们受转换指令支持。

- BCD16 是 16 位值（-999 到 999）。
- BCD32 是 32 位值（-9999999 到 9999999）。

4.4.2 对存储区进行寻址

STEP 7 简化了符号编程。用户为数据地址创建符号名称或“变量”，作为与存储器地址和 I/O 点相关的 PLC 变量或在代码块中使用的局部变量。

要在用户程序中使用这些变量，只需输入指令参数的变量名称。为了更好地理解 CPU 的存储区结构及其寻址方式，以下段落将对 PLC 变量所引用的“绝对”寻址进行说明。

CPU 提供了以下几个选项，用于在执行用户程序期间存储数据：

- **全局存储器：** CPU 提供了各种专用存储区，其中包括输入 (I)、输出 (Q) 和位存储器 (M)。所有代码块可以无限制地访问该存储器。
- **数据块 (DB)：** 可在用户程序中加入 DB 以存储代码块的数据。
从相关代码块开始执行一直到结束，存储的数据始终存在。“全局”DB 存储所有代码块均可使用的数据，而背景 DB 存储特定 FB 的数据并且由 FB 的参数进行构造。
- **临时存储器：** 只要调用代码块，CPU 的操作系统就会分配要在执行块期间使用的临时或本地存储器 (L)。代码块执行完成后，CPU 将重新分配本地存储器，以用于执行其它代码块。

每个存储单元都有唯一的地址。用户程序利用这些地址访问存储单元中的信息。

对输入 (I) 或输出 (Q) 存储区（例如 I0.3 或 Q1.7）的引用会访问过程映像。

要立即访问物理输入或输出，请在引用后面添加“:P”（例如，I0.3:P、Q1.7:P 或 "Stop:P"）。

强制仅将固定值写入物理输入 (Ix.y:P) 或物理输出 (Qx.y:P)。要强制输入或输出，请在 PLC 变量或地址后面添加“:P”。更多信息，请参见“强制 CPU 中的变量” (页 369)。

表格 4-4 存储区

存储区	说明	强制	保持性
I 过程映像输入 I_:P ¹ (物理输入)	在扫描周期开始时从物理输入复制	否	否
	立即读取 CPU、SB 和 SM 上的物理输入点	是	否
Q 过程映像输出 Q_:P ¹ (物理输出)	在扫描周期开始时复制到物理输出	否	否
	立即写入 CPU、SB 和 SM 上的物理输出点	是	否
M 位存储器	控制和数据存储器	否	是 (可选)

存储区	说明	强制	保持性
L 临时存储器	存储区的临时数据，这些数据仅在该块的本地范围内有效	否	否
DB 数据块	数据存储区，同时也是 FB 的参数存储器	否	是 (可选)

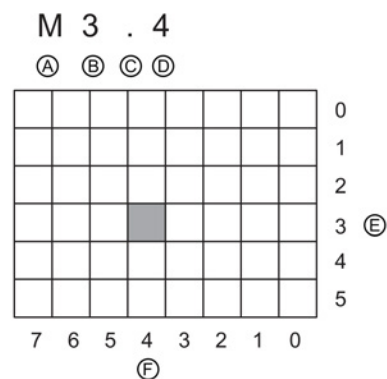
1 要立即访问（或强制）物理输入和物理输出，请在地址或变量后面添加“:P”（例如，I0.3:P、Q1.7:P 或 "Stop:P"）。

每个存储单元都有唯一的地址。用户程序利用这些地址访问存储单元中的信息。
绝对地址由以下元素组成：

- 存储区（如 I、Q 或 M）
- 要访问的数据的大小（如“B”表示 Byte 或“W”表示 Word）
- 数据地址（如 Byte 3 或 Word 3）

访问布尔值地址中的位时，不要输入大小的助记符号。

仅需输入数据的存储区、字节位置和位位置（如 0.0、Q0.1 或 M3.4）。



存储区的绝对地址：

- A 存储区标识符
- B 字节地址：字节 3
- C 分隔符（“字节.位”）
- D 位在字节中的位置（位 4，共 8 位）
- E 存储区的字节
- F 选定字节的位

本示例中，存储区和字节地址（M 代表位存储区，3 代表 Byte 3）通过后面的句点（“.”）与位地址（位 4）分隔。

对 CPU 和 I/O 模块中的 I/O 进行组态



模块	插槽	I 地址	Q 地址	类型	订货号
	103				
	102				
RS485_1	101			CM 1241 (RS485)	6ES7 2...
PLC_1	1			CPU 1214C D0D0DC	6ES7 2...
DI14/DO10	1.1	0..1	0..1	DI14/DO10	
AI2	1.2	64..67		AI2	
AO1 x 12 ...	1.3		80..81	AO1 信号板	6ES7 2...
HSC_1	1.16	1000..1...		高速计数器 (HSC)	
HSC_2	1.17			高速计数器 (HSC)	
HSC_3	1.18			高速计数器 (HSC)	
HSC_4	1.19			高速计数器 (HSC)	
HSC_5	1.20			高速计数器 (HSC)	
HSC_6	1.21			高速计数器 (HSC)	
Pulse_1	1.32			脉冲发生器 (PTOP...	
Pulse_2	1.33			脉冲发生器 (PTOP...	
PROFINET ...	X1			PROFINET 接口	
DIB x 24VDC_1	2		8	SM 1221 DIB x 24V...	6ES7 2...

向设备组态添加 CPU 和 I/O 模块时，STEP 7 会自动分配 I 地址和 Q 地址。

通过在设备组态中选择地址字段并输入新编号，可以更改默认寻址设置。

- 无论模块是否使用所有点，STEP 7 都按每组 8 点（1 字节）的方式分配数字量输入和输出。
- STEP 7 按照每组 2 点的方式分配模拟量输入和输出，其中每个模拟点占用 2 个字节（16 位）。

图中显示的示例是配有两个 SM 及一个 SB 的 CPU 1214C。在此示例中，可以将 DI8 模块的地址改为 2 来替代

8。工具可以协助您更改大小错误或与其它地址相冲突的地址范围。

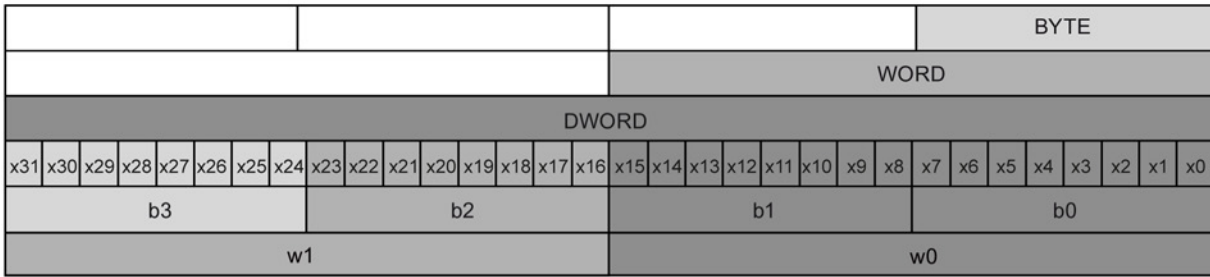
4.4.3 访问一个变量数据类型的“片段”

可以根据大小按位、字节、或字级别访问 PLC 变量和数据块变量。

访问此类数据片段的语法如下所示：

- "<PLC 变量名称>".xn（按位访问）
- "<PLC 变量名称>".bn（按字节访问）
- "<PLC 变量名称>".wn（按字访问）
- "<数据块名称>".<变量名称>.xn（按访问）
- "<数据块名称>".<变量名称>.bn（按字节访问）
- "<数据块名称>".<变量名称>.wn（按字访问）

双字大小的变量可按位 0 - 31、字节 0 - 3 或字 0 - 1 访问。一个字大小的变量可按位 0 - 15、字节 0 - 1 或字 0 访问。字节大小的变量则可按位 0 - 7 或字节 0 访问。当预期操作数为位、字节或字时，则可使用位、字节和字片段访问方式。



说明

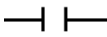

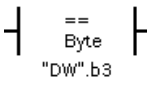
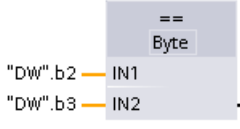
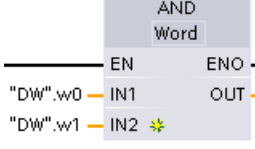
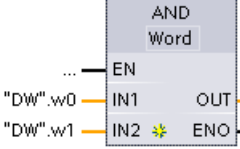
可以按片段访问的有效数据类型有：

Byte、Char、Conn_Any、Date、DInt、DWord、Event_Any、Event_Att、Hw_Any、Hw_Device、HW_Interface、Hw_Io、Hw_Pwm、Hw_SubModule、Int、OB_Any、OB_Att、OB_Cyclic、OB_Delay、OB_WHINT、OB_PCYCLE、OB_STARTUP、OB_TIMEER、ROR、OB_Tod、Port、Rtm、SInt、Time、Time_Of_Day、UDInt、UInt、USInt 和 Word。Real 类型的 PLC 变量可以按片段访问，但 Real 类型的数据块变量则不行。

示例

在 PLC 变量表中，“DW”是一个声明为 DWORD 类型的变量。

在以下示例中，显示了按位、字节和字片段的访问方式：

	LAD	FBD	SCL
按位访问	<p>"DW".x11</p> 		<pre>IF "DW".x11 THEN ... END_IF;</pre>
按字节访问	<p>"DW".b2</p> <p>== Byte</p> <p>"DW".b3</p> 		<pre>IF "DW".b2 = "DW".b3 THEN ... END_IF;</pre>
按字访问			<pre>out:= "DW".w0 AND "DW".w1;</pre>

4.4.4 访问带有一个 AT 覆盖的变量

借助 AT

变量覆盖，可通过一个不同数据类型的覆盖声明访问标准访问块中已声明的变量。

例如，可以通过 **Array of Bool** 寻址数据类型为 **Byte**、**Word** 或 **DWord** 变量的各个位。

声明

要覆盖一个参数，可以在待覆盖的参数后直接声明一个附加参数，然后选择数据类型“AT”。编辑器随即创建该覆盖，然后选择将用于该覆盖的数据类型、结构或数组。

示例

在本例中，显示一个标准访问 **FB** 的输入参数。字节变量 **B1** 将由一个布尔型数组覆盖：

[-]	■	B1	Byte	0.0
[-]	▼	OV	AT*B1*	Array[0..7] of Bool
[-]	■	OV[0]	Bool	0.0
[-]	■	OV[1]	Bool	0.1
[-]	■	OV[2]	Bool	0.2
[-]	■	OV[3]	Bool	0.3
[-]	■	OV[4]	Bool	0.4
[-]	■	OV[5]	Bool	0.5
[-]	■	OV[6]	Bool	0.6
[-]	■	OV[7]	Bool	0.7

另一个示例是 **DWord** 变量由一个 **Struct** 覆盖，其中包括字、字节和两个布尔值：

[-]	■	DW1	DWord	2.0
[-]	▼	DW1_Struct	AT*DW1*	Struct
[-]	■	W1	Word	0.0
[-]	■	B1	Byte	2.0
[-]	■	BO1	Bool	3.0
[-]	■	BO2	Bool	3.1

块接口的“偏移量”(Offset) 列中显示与原始变量相关的被覆盖数据类型的位置。

可直接在程序逻辑中指定覆盖类型的地址：

LAD	FBD	SCL
		<pre>IF #OV[1] THEN ... END_IF;</pre>
		<pre>IF #DW1_Struct.W1 = W#16#000C THEN ... END_IF;</pre>
		<pre>out1 := #DW1_Struct.B1;</pre>
		<pre>IF #OV[4] AND #DW1_Struct.BO2 THEN ... END_IF;</pre>

准则

- 只能覆盖可标准（未优化）访问的 FB 和 FC 块中的变量。
- 可以覆盖所有类型 and 所有声明部分的变量。
- 可以同使用其它块参数一样使用覆盖后的参数。
- 不能覆盖 VARIANT 类型的参数。
- 覆盖参数的大小必须小于等于被覆盖的参数。
- 必须在覆盖变量并选择关键字“AT”作为初始数据类型后立即声明覆盖变量。

4.5 脉冲输出

可将 CPU 或信号板 (SB) 组态为脉冲宽度调制 (PWM) 或脉冲串输出 (PTO)，以提供用于控制高速脉冲输出函数的四个脉冲发生器。基本运动指令使用 PTO 输出。可将每个脉冲发生器指定为 PWM 或 PTO，但不能指定为既是 PWM 又是 PTO。



用户程序中的其它指令无法使用脉冲输出。将 CPU 或 SB 的输出组态为脉冲发生器时，会从 Q 存储器中移除相应的输出地址，并且这些地址在用户程序中不能用于其它用途。如果用户程序向用作脉冲发生器的输出写入值，则 CPU 不会将该值写入到物理输出。

说明

请勿超出最大脉冲频率。

CPU 1217C 脉冲输出发生器的最大脉冲频率为 1 MHz，CPU 1211C、1212C、1214C 和 1215C 则为 100 kHz；20 kHz（对于标准 SB）；或 200 kHz（对于高速 SB）。

这四个脉冲发生器具有默认的 I/O 分配，但是，它们可组态为 CPU 或 SB 上的任意数字量输出。不能将 CPU 上的脉冲发生器分配至分布式 I/O。

组态基本运动指令时，请注意，如果被组态轴的最大速度或频率超出了此硬件限制，STEP 7 并不会提醒用户。

这可能会导致应用出现问题，因此请始终确保不会超出硬件的最大脉冲频率。

可以使用板载 CPU 输出，也可以使用可选的信号板输出。

下表列出了输出点编号（假定使用默认输出组态）。

如果更改了输出点编号，则输出点编号将为用户指定的编号。请注意，PWM 仅需要一个输出，而 PTO 的每个通道可选择使用两个输出。

如果脉冲功能不需要输出，则相应的输出可用于其它用途。

这四个脉冲发生器具有默认的 I/O 分配，但是，它们可组态为 CPU 或 SB 上的任意数字量输出。不能将 CPU 上的脉冲发生器分配给 SM 或分布式 I/O。

表格 4-5 脉冲发生器的默认输出分配

说明	脉冲	方向
PTO1		
内置 I/O	Q0.0	Q0.1
SB I/O	Q4.0	Q4.1
PWM1		
内置输出	Q0.0	-
SB 输出	Q4.0	-
PTO2		
内置 I/O	Q0.2	Q0.3
SB I/O	Q4.2	Q4.3
PWM2		
内置输出	Q0.2	-
SB 输出	Q4.2	-
PTO3		
内置 I/O	Q0.4 ¹	Q0.5 ¹
SB I/O	Q4.0	Q4.1
PWM3		
内置输出	Q0.4 ¹	-
SB 输出	Q4.1	-
PTO4		
内置 I/O	Q0.6 ²	Q0.7 ²
SB I/O	Q4.2	Q4.3
PWM4		

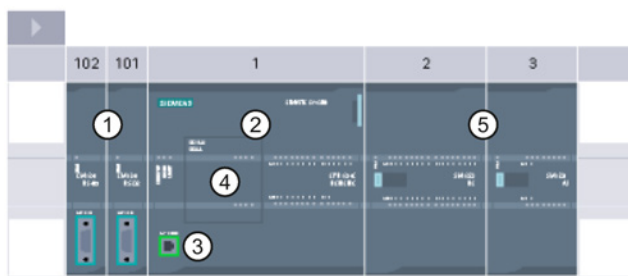
4.5 脉冲输出

说明	脉冲	方向
内置输出	Q0.6 ²	-
SB 输出	Q4.3	-

- 1 CPU 1211C 没有输出 Q0.4、Q0.5、Q0.6 或 Q0.7。因此这些输出不能在 CPU 1211C 中使用。
- 2 CPU 1212C 没有输出 Q0.6 或 Q0.7。因此这些输出不能在 CPU 1212C 中使用。
- 3 该表适用于 CPU 1211C、CPU 1212C、CPU 1214C、CPU 1215C 以及 CPU 1217C PTO/PWM 功能。

轻松创建设备配置

通过向项目中添加 CPU 和其它模块为 PLC 创建设备组态。



- ① 通信模块 (CM) 或通信处理器 (CP): 最多 3 个, 分别插在插槽 101、102 和 103 中
- ② CPU: 插槽 1
- ③ CPU 的以太网端口
- ④ 信号板 (SB)、通信板 (CB) 或电池板 (BB): 最多 1 个, 插在 CPU 中
- ⑤ 数字或模拟 I/O 的信号模块 (SM): 最多 8 个, 分别插在插槽 2 到 9 中
(CPU 1214C、CPU 1215C 和 CPU 1217C 允许使用 8 个; CPU 1212C 允许使用 2 个; CPU 1211C 不允许使用任何信号模块)

要创建设备组态, 需向项目中添加设备。

- 在门户视图中, 选择“设备和网络”(Devices & Networks) 并单击“添加设备”(Add device)。
- 在项目视图中的项目名称下, 双击“添加新设备”(Add new device)。



5.1 上传已连接 CPU 的组态

STEP 7 提供两种上传已连接 CPU 的硬件配置的方法：

- 将已连接设备作为新站上传
- 组态未指定的 CPU 并检测已连接 CPU 的硬件配置

不过需要注意的是，第一种方法将同时上传已连接 CPU 的硬件配置和软件。

将设备作为新站上传

要将已连接设备作为新站上传，请按以下步骤操作：

1. 从项目树的“在线访问”(Online access) 节点中展开通信接口。
2. 双击“更新可访问的设备”(Update accessible devices)。
3. 从检测到的设备中选择 PLC。



4. 从 STEP 7 的“在线”(Online)

菜单中，选择“将设备作为新站上传（硬件和软件）”(Upload device as new station (hardware and software)) 菜单命令。

STEP 7 将同时上传硬件配置和程序块。

检测未指定 CPU 的硬件配置

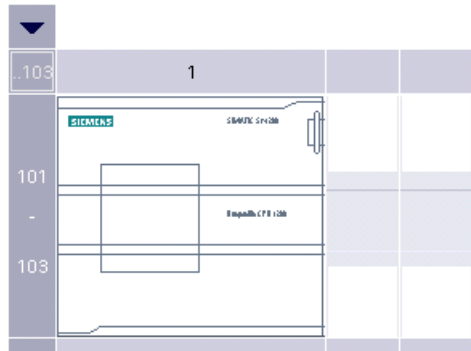


如果已连接到 CPU，则可以将该 CPU（包括所有模块）的组态上传到用户项目中。只需创建新项目并选择“未指定的 CPU”而不是选择特定的 CPU 即可。

（也可通过从“新手上路”(First steps) 中选择“创建 PLC 程序”(Create a PLC program) 完全跳过设备组态。STEP 7 即会自动创建一个未指定的 CPU。）

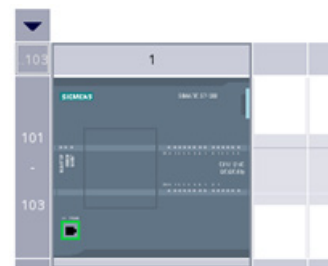
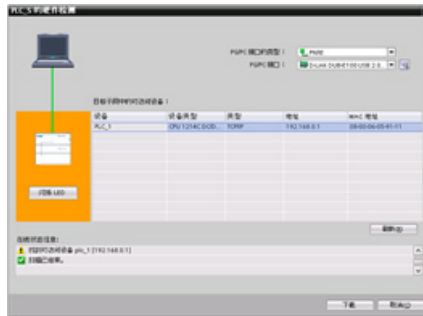
在程序编辑器中，从“在线”(Online) 菜单中选择“硬件检测”(Hardware detection) 命令。

在设备组态编辑器中，选择用于检测所连设备组态的选项。



未指定该设备。
→ 请使用 [硬件目录](#) 指定 CPU。
→ 或 [检测](#) 相连设备的组态。

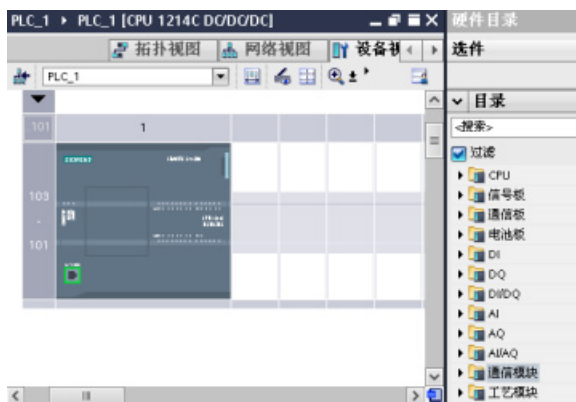
从在线对话框中选择 CPU 并单击“加载”(Load) 按钮后，STEP 7 会上传 CPU 以及所有模块（SM、SB 或 CM）的硬件配置。随后可以为 CPU 和模块 (页 86)组态参数。



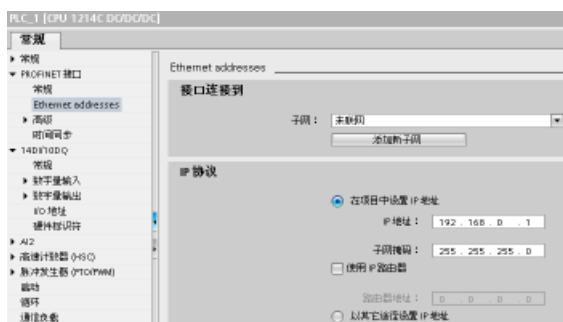
5.2 将 CPU 添加到组态中



通过将 CPU 插入到项目中创建设备组态。在“添加新设备”(Add a new device)对话框中选择 CPU 并单击“确定”(OK)，将 CPU 添加到项目中。



设备视图显示了 CPU 和机架。



通过在设备视图中选择 CPU，可在巡视窗口中显示 CPU 属性。使用这些属性组态 CPU 的运行参数 (页 86)。

说明

CPU 不具有预组态的 IP 地址。设备组态期间必须为 CPU 手动分配 IP 地址。如果 CPU 连接到网络上的路由器，则也应输入路由器的 IP 地址。

5.3 更改设备

您可以更改已组态 CPU 或模块的设备类型。

在设备组态中，右键单击设备并从上下文菜单中选择“更改设备”(Change device)。

在随后出现的对话框中，导航到您想要更换的 CPU 或模块并选择。“更改设备”(Change device) 对话框将显示两个设备之间的兼容性信息。

说明

设备更换：用 V4.1 CPU 替换 V3.0 CPU

您可以在 STEP 7 V13 中打开 STEP 7 V12 项目，用 V4.1 CPU 替换 V3.0 CPU。

无法替换 V3.0 之前版本的 CPU。用 V4.1 CPU 替换 V3.0 CPU

时，请注意两个版本在特性和行为方面的差异 (页 477)，并采取必要措施。

如果项目的 CPU 版本小于 V3.0，必须先将 CPU 升级到 V3.0，然后再将其升级到 V4.1。

5.4 将模块添加到组态

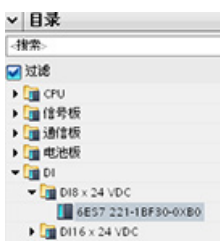


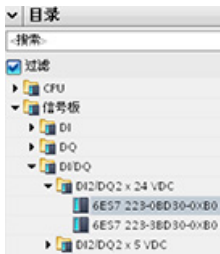


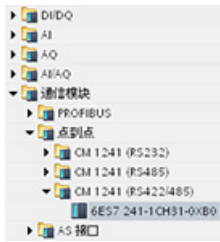


使用硬件目录将模块添加到 CPU：

- 信号模块 (SM) 提供附加的数字或模拟 I/O 点。这些模块连接在 CPU 右侧。
- 信号板 (SB) 仅为 CPU 提供几个附加的 I/O 点。SB 安装在 CPU 的前端。
- 电池板 1297 (BB) 可提供长期的实时时钟备份。BB 安装在 CPU 的前端。
- 通信板 (CB) 提供附加的通信端口（如 RS485）。CB 安装在 CPU 的前端。
- 通信模块 (CM) 和通信处理器 (CP) 提供附加的通信端口（如用于 PROFIBUS 或 GPRS）。这些模块连接在 CPU 左侧。

5.4 将模块添加到组态

要将模块插入到设备组态中，可在硬件目录中选择模块，然后双击该模块或将其拖到高亮显示的插槽中。必须将模块添加到设备组态并将硬件配置下载到 CPU 中，模块才能正常工作。

表格 5-1 将模块添加到设备组态中

模块	选择模块	插入模块	结果
SM			
SB、B B 或 CB			
CM 或 CP			

使用“组态控制”功能

(页 85)，用户可以添加信号模块和信号板到设备组态，虽然这样有可能与特定应用的实际硬件不符，但可用于共享通用用户程序、CPU 型号以及一些已组态模块的相关应用。

5.5 组态控制

在创建要用于多种不同安装的自动化解决方案（机器）时，组态控制将是非常实用的解决方案。

通过 STEP 7 和 S7-1200

的组态控制，可以为标准机器组态最高组态，并运行仅使用其中部分组态的版本（选项）。“使用 STEP 7 组态 PROFINET

(<http://support.automation.siemens.com/WW/view/en/49948856>)”手册将这些类型的项目称为“标准机器项目”。

用户可以将 STEP 7 设备组态和用户程序加载到所安装的不同 PLC 组态。

只需稍作修改，使 STEP 7 项目适合实际安装。

在启动程序块中编程的控制数据记录将通知

CPU，与组态相比实际安装中缺少哪些模块，或者哪些模块位于与组态不同的插槽中。

组态控制对模块的参数分配没有影响。

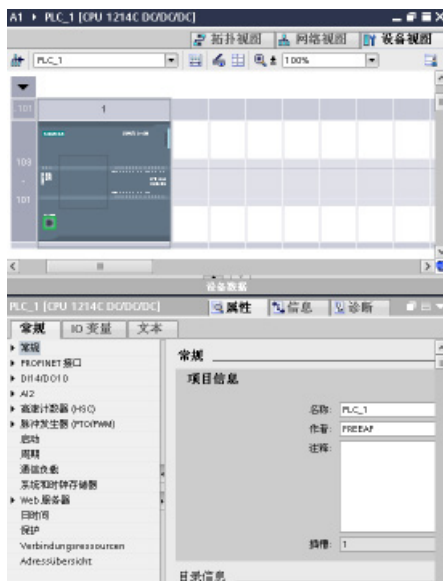
组态控制为用户提供了灵活性，只要实际组态不超过 STEP 7

的最高设备组态，就可以改变安装。

有关组态控制的说明和示例，请参见“S7-1200 可编程控制器系统手册”。

5.6 组态 CPU 和模块的运行

要组态 CPU 的运行参数，请在设备视图中选择该 CPU，并使用巡视窗口的“属性”(Properties) 选项卡。



可组态下列 CPU 属性：

- CPU 的 PROFINET IP 地址和时间同步
- CPU 在关机到开机切换之后的启动特性
- 本地（板载）数字和模拟 I/O、高速计数器 (HSC) 和脉冲发生器
- 系统时钟（时间、时区和夏令时）
- 访问 CPU 所涉及的读/写保护和密码
- 最大循环时间或固定最小循环时间以及通信负载
- “Web 服务器”(Web server) 属性

组态 CPU 从 STOP 到 RUN 的运行

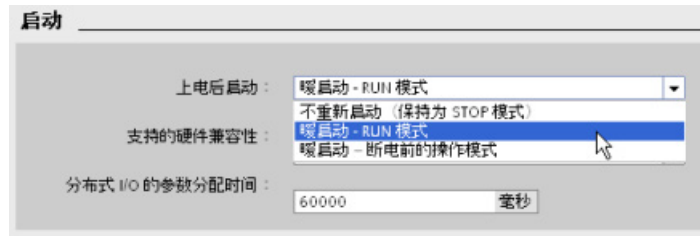
只要工作状态从 STOP 切换到 RUN，CPU 就会清除过程映像输入、初始化过程映像输出并处理启动 OB。（因此，启动 OB 中的指令此时对过程映像输入进行任何读访问，读取到的只有零，而不是当前物理输入值。）要在启动期间读取物理输入的当前状态，必须先执行立即读取操作。接着再执行启动 OB 以及任何相关的 FC 和 FB。如果存在多个启动 OB，则按照 OB 编号依次执行各启动 OB，OB 编号最小的先执行。

在启动过程中，CPU 还会执行以下任务。

- 在启动阶段，对中断进行排队但不加以处理
- 在启动阶段，不执行任何循环时间监视
- 在启动模式下，可以更改 HSC（High-Speed Counter，高速计数器）、PWM（Pulse-Width Modulation，脉冲宽度调制）以及 PtP（Point-to-Point communication，点对点通信）模块的组态
- 只有在 RUN 模式下才会真正运行 HSC、PWM 和点对点通信模块

执行完启动 OB 后，CPU 将进入 RUN 模式并在连续的扫描周期内处理控制任务。

使用 CPU 属性组态 CPU 在上电循环后的启动方式。



- 处于 STOP 模式
- 在 RUN 模式下
- 处于之前的模式（上电循环之前）

CPU 在进入 RUN 模式前执行暖启动。

暖启动会将所有非保持性存储器复位为默认初始值，但 CPU 会保留保持性存储器中存储的当前值。

说明

下载完成后 CPU 总是会执行重新启动

每次下载完项目元素（例如程序块、数据块或硬件配置），CPU 都会在下一次切换到 RUN 模式时先执行重新启动。

除清除输入、初始化输出以及初始化非保持性存储器之外，重新启动还会初始化保持性存储区。

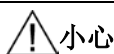
在紧随下载的重新启动完成之后，所有随后的 STOP 到 RUN 切换均会执行暖启动（不会初始化保持性存储器）。

5.6.1 系统存储器和时钟存储器提供了标准功能

使用 CPU 属性可启用“系统存储器”和“时钟存储器”的相应字节。
程序逻辑可通过这些函数的变量名称来引用它们的各个位。

- 可以将 M 存储器的一个字节分配给系统存储器。
该系统存储器字节提供了以下四个位，用户程序可通过以下变量名称引用这四个位：
 - 第一个周期：（变量名称“FirstScan”）在启动 OB 完成后的第一次扫描期间内，该位设置为 1。
（执行了第一次扫描后，“首次扫描”位将设置为 0。）
 - 诊断状态变化：（变量名称：“DiagStatusUpdate”）在 CPU 记录了诊断事件后的一个扫描周期内设置为 1。由于直到首次程序循环 OB 执行结束，CPU 才能置位“DiagStatusUpdate”位，因此用户程序无法检测在启动 OB 执行期间或首次程序循环 OB 执行期间是否发生过诊断更改。
 - 始终为 1（高）(Always 1 (high))：（变量名称“AlwaysTRUE”），该位始终设置为 1。
 - 始终为 0（低）(Always 0 (low))：（变量名称“AlwaysFALSE”），该位始终设置为 0。
- 可以将 M 存储器的一个字节分配给时钟存储器。
被组态为时钟存储器的字节中的每一位都可生成方波脉冲。时钟存储器字节提供了 8 种不同的频率，其范围从 0.5 Hz（慢）到 10 Hz（快）。
这些位可作为控制位（尤其在沿指令结合使用时），用于在用户程序中周期性触发动作。

CPU 在从 STOP 模式切换到 STARTUP 模式时初始化这些字节。时钟存储器的位在 STARTUP 和 RUN 模式下会随 CPU 时钟同步变化。



小心

覆盖系统存储器位或时钟存储器位时的风险

改写系统存储器或时钟存储器的各个位可能会破坏这些功能中的数据，同时还可能导致用户程序错误运行，进而造成设备损坏和人员伤害。

因为时钟存储器和系统存储器都不是预留的 M 存储器，所以指令或通信可以写入这些单元并破坏其中的数据。

避免向这些单元写入数据以确保这些功能正常运行，并且应始终为过程或机器使用紧急停止电路。

系统存储器组态了一个字节，其中的各个位会在发生特定事件时启用（值 = 1）。

系统存储器位

允许使用系统存储器字节

系统存储器字节的地址 (MBx) :

首次循环 :

诊断状态已更改 :

始终为 1 (高电平) :

始终为 0 (低电平) :

表格 5-2 系统存储器

7	6	5	4	3	2	1	0
保留 值 0				始终熄灭 值 0	常开 值 1	诊断状态指示 • 1: 变化 • 0: 无更改	首次扫描指示 • 1: 启动后首次扫描 • 0: 不是首次扫描

时钟存储器组态了一个字节，该字节的各个位分别按固定的时间间隔循环启用和禁用。每个时钟位都会在相应的 M 存储器位产生一个方波脉冲。这些位可作为控制位（尤其在沿指令结合使用时），用于在用户代码中周期性触发动作。

时钟存储器位

允许使用时钟存储器字节

时钟存储器字节的地址 (MBx) :

10 Hz 时钟 :

5 Hz 时钟 :

2.5 Hz 时钟 :

2 Hz 时钟 :

1.25 Hz 时钟 :

1 Hz 时钟 :

0.625 Hz 时钟 :

0.5 Hz 时钟 :

表格 5-3 时钟存储器

位号	7	6	5	4	3	2	1	0
变量名称								
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

由于时钟存储器与 CPU

周期异步运行，因此，时钟存储器的状态可能会在一个长周期中发生多次改变。

组态 I/O 和通信模块的运行

要组态信号模块 (SM)、信号板 (SB) 或通信模块 (CM) 的运行参数，请在设备视图选择相应模块，并使用巡视窗口的“属性”(Properties) 选项卡。



信号模块 (SM) 和信号板 (SB)

- 数字量 I/O: 组态各个输入，如用于沿检测和“脉冲捕捉”（瞬时高脉冲或低脉冲之后保持开启或关闭，持续一个循环）。组态输出以在 RUN 模式转换到 STOP 模式时使用冻结值或替换值。
- 模拟量 I/O: 组态各个输入的参数（如电压/电流、范围和平滑化），并启用下溢或上溢诊断。组态各个模拟量输出的参数并启用诊断，例如，短路（对于电压输出）或溢出值诊断。
- I/O 地址: 组态用于设置模块的输入和输出的起始地址。



通信模块 (CM) 和通信板 (CB)

- 端口组态: 组态通信参数，如波特率、奇偶校验、数据位、停止位以及等待时间。
- 发送和接收消息: 组态与发送和接收数据相关的选项（例如，消息起始参数和消息结束参数）也可以通过用户程序更改这些组态参数。

5.7 组态 CPU 的 IP 地址

因为 CPU 没有预组态的 IP 地址，所以必须手动分配 IP 地址。在组态 CPU 的属性时组态 PROFINET 接口的 IP 地址与其它参数。

- 在 PROFINET 网络中，制造商会为每个设备都分配一个唯一的“介质访问控制”地址（MAC 地址）以进行标识。每个设备也都必须具有一个 IP 地址。
- 子网是已连接的网络设备的逻辑分组。掩码（又称为子网掩码或网络掩码）定义子网的边界。不同子网间的唯一连接通过路由器实现。路由器是 LAN 之间的链接，它依靠 IP 地址传送和接收数据包。

将 IP 地址下载到 CPU 之前，必须先确保 CPU 的 IP 地址与编程设备的 IP 地址兼容。

可以使用 STEP 7 确定编程设备的 IP 地址：

1. 展开项目树中的“在线访问”(Online access) 文件夹以显示网络。
2. 选择连接 CPU 的网络。
3. 右键单击特定网络，以显示上下文菜单。
4. 选择“属性”(Properties) 命令。

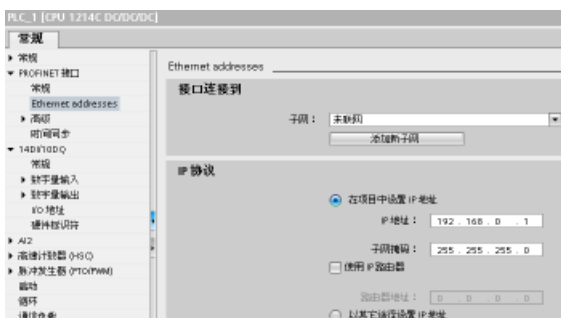
说明

CPU 的 IP 地址必须与编程设备的 IP 地址和子网掩码兼容。请咨询网络专家以确定 CPU 的适当 IP 地址和子网掩码。

5.7 组态 CPU 的 IP 地址



“属性”(Properties)窗口显示编程设备的设置。



确定 CPU 的 IP 地址和子网掩码之后，输入 CPU 和路由器（如果适用）的 IP 地址。有关更多信息，请参见“S7-1200 可编程控制器系统手册”。



完成组态后，将项目下载到 CPU 中。在下载项目前组态 CPU 和路由器（如果适用）的 IP 地址。

5.8 防止对 CPU 或代码块的访问非常容易

CPU 提供了四个安全等级，用于限制对特定功能的访问。为 CPU 组态安全等级和密码时，可以对那些不输入密码就能访问的功能和存储区进行限制。

每个等级都允许在访问某些功能时不使用密码。CPU 的默认状态是没有任何限制，也没有密码保护。要限制 CPU 的访问，可以对 CPU 的属性进行组态并输入密码。

通过网络输入密码并不会使 CPU 的密码保护受到威胁。密码保护不适用于用户程序指令的执行，包括通信功能。输入正确的密码便可访问该级别的所有功能。

PLC 到 PLC 通信（使用代码块中的通信指令）不受 CPU 中安全等级的限制。

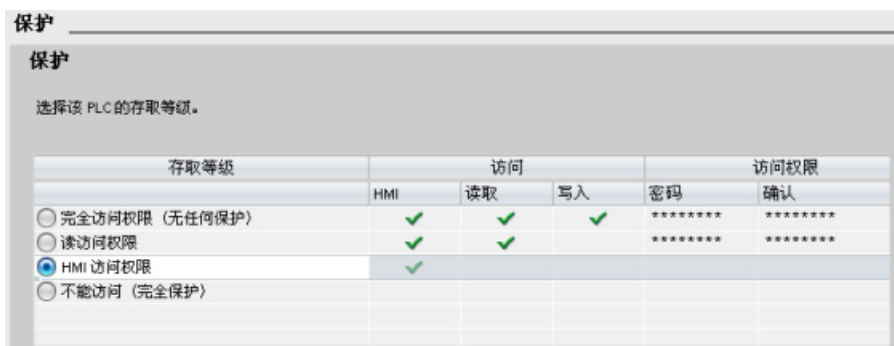
表格 5-4 CPU 的安全级别

安全等级	访问限制
完全访问（无保护）	允许完全访问，没有密码保护。
读访问	允许 HMI 访问和各种形式的 PLC 到 PLC 通信，没有密码保护。 以下情况下需要密码：修改（写入）CPU 以及更改 CPU 模式 (RUN/STOP)。
HMI 访问	允许 HMI 访问和各种形式的 PLC 到 PLC 通信，没有密码保护。 以下情况下需要密码：读取 CPU 中的数据、修改（写入）CPU 以及更改 CPU 模式 (RUN/STOP)。
无访问（完全保护）	不允许没有密码保护的访问。 进行 HMI 访问、读取 CPU 中的数据和修改（写入）CPU 时需要密码。

密码区分大小写。要组态保护级别和密码，请按以下步骤操作：

1. 在“设备组态”(Device configuration) 中，选择 CPU。
2. 在巡视窗口中，选择“属性”(Properties) 选项卡。
3. 选择“保护”(Protection) 属性以选择保护等级和输入密码。


5.8 防止对 CPU 或代码块的访问非常容易



当您将此组态下载至 CPU 时，用户将具有 HMI 访问权限，可以在无密码的情况下访问 HMI 功能。

要读取数据，用户必须输入“读访问”的已组态密码或“完全访问（无保护）”的密码。

要写入数据，用户必须输入“完全访问（无保护）”的已组态密码。

 **警告**

对受保护的 CPU 进行未授权访问

拥有 CPU 完全访问权限的用户有权限读写 PLC 变量。无论 CPU 访问级别是多少，Web 服务器用户都有权限读写 PLC 变量。未经授权访问 CPU 或将 PLC 变量更改为无效值可能会中断过程操作并可能导致死亡、严重人身伤害和/或财产损失。授权用户可以执行共模模式更改、写入 PLC 数据以及进行固件更新。Siemens 建议您遵守以下安全实践：

- 使用强密码对 CPU 访问级别和 Web 服务器用户 ID (页 276) 进行密码保护。
强密码在长度上至少为十个字符，可以是字母、数字和特殊字符的组合，不能是可在字典上找到的词，并且不能是从个人信息推断出的名字或标识符。
保管好密码并经常更改密码。
- 仅使用 HTTPS 协议启用对 Web 服务器的访问。
- 不要扩展 Web 服务器“所有人”(Everybody) 用户的默认最低权限。
- 对程序逻辑中的变量执行错误检查和范围检查，因为 Web 页面用户可将 PLC 变量更改为无效值。

连接机制

要使用 PUT/GET 指令访问远程连接伙伴，用户还必须得到许可。

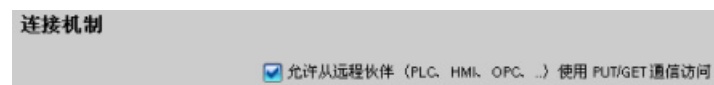
默认情况下，“允许使用 PUT/GET 通信进行访问”(Permit access with PUT/GET communication) 选项处于未启用状态。这时，只有需要对本地 CPU 和通信伙伴同时进行组态和编程的通信连接才能实现对 CPU 数据的读写访问。例如，可以通过 BSEND/BRCV 指令进行访问。

因此，本地 CPU 仅作为服务器的连接（也就是说，本地 CPU 中不存在带有通信伙伴的通信组态/编程）在 CPU 运行期间不可用，例如：

- 通过通信模块进行 PUT/GET、FETCH/WRITE 或 FTP 访问
- 从其它 S7 CPU 进行 PUT/GET 访问
- 通过 PUT/GET 通信进行 HMI 访问

如果您希望允许从客户端访问 CPU 数据，即您不希望限制 CPU 的通信服务，请按以下步骤操作：

1. 将保护访问级别组态为除“无访问（完全保护）”(No access (complete protection)) 外的任意级别。
2. 选择“允许使用 PUT/GET 通信进行访问”(Permit access with PUT/GET communication) 复选框。



当您将此组态下载至 CPU 时，CPU 将允许与远程伙伴进行 PUT/GET 通信

5.8.1 专有技术保护

专有技术保护可防止程序中的一个或多个代码块（OB、FB、FC 或 DB）受到未经授权的访问。用户创建密码以限制对代码块的访问。密码保护会防止对代码块进行未授权的读取或修改。

如果没有密码，只能读取有关代码块的以下信息：

- 块标题、块注释和块属性
- 传送参数（IN、OUT、IN_OUT、Return）
- 程序的调用结构
- 交叉引用中的全局变量（不带使用时的信息），但局部变量已隐藏

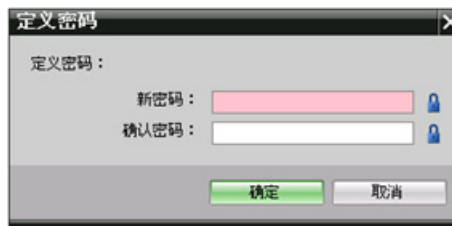
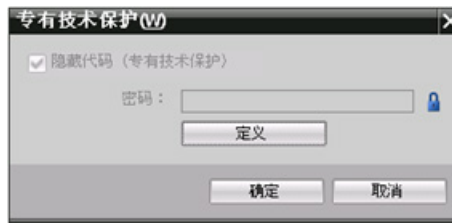
将块组态为“专有技术”保护时，只有在输入密码后才能访问块内的代码。

5.8 防止对 CPU 或代码块的访问非常容易

使用代码块的“属性”(Properties) 任务卡组态该块的专有技术保护。打开代码块后，从“属性”(Properties) 中选择“保护”(Protection)。



1. 在代码块的“属性”(Properties) 中，单击“保护”(Protection) 按钮显示“专有技术保护”(Know-how protection) 对话框。
2. 单击“定义”(Define) 按钮输入密码。输入并确认密码后，单击“确定”(OK)。



5.8.2 复制保护

附加安全特性允许捆绑程序块，以用于特定存储卡或 CPU。

该特性对于保护您的知识产权特别有用。

当您将程序块与特定设备捆绑在一起时，就会将程序或代码块限制为仅用于特定存储卡或 CPU。该特性允许您以电子方式（如通过 Internet 或通过电子邮件）或通过发送内存匣分布程序或代码块。复制保护可用于 OB (页 101)、FB (页 103) 和 FC (页 103)。S7-1200 CPU 支持三种类型的块保护：

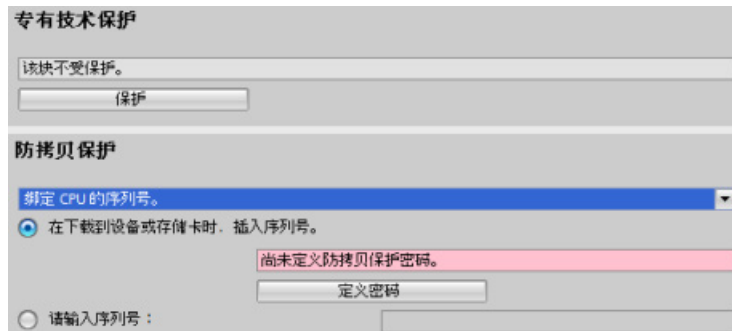
- 与 CPU 的序列号进行绑定
- 与存储卡的序列号进行绑定
- 与强制性密码动态绑定

使用代码块的“属性”(Properties) 任务卡将块捆绑到特定 CPU 或存储卡。

1. 打开代码块之后选择“保护”(Protection)。



2. 在“复制保护”(Copy protection) 任务下的下拉列表中，选择要使用的复制保护的类型。



3. 对于与 CPU 或存储卡序列号的绑定，可以在下载时插入序列号，也可以输入存储卡或 CPU 的序列号。

说明

序列号区分大小写。

对于与强制性密码的动态绑定，定义下载或复制块所必须使用的密码。

随后下载带有动态绑定的块时，必须输入可用于下载块的密码。

请注意，复制保护密码和专有技术保护 (页 95)密码是两个不同的密码。

5.8 防止对 CPU 或代码块的访问非常容易

简化了编程

6.1 轻松设计用户程序

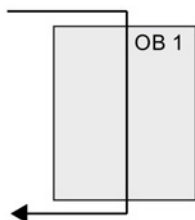
创建处理自动化任务的用户程序时，需要将程序指令插入代码块（OB、FB 或 FC）中。

为用户程序选择结构类型

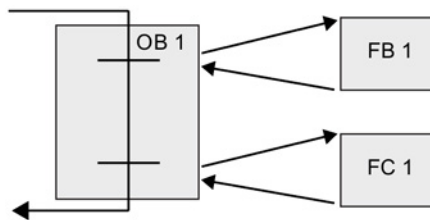
根据实际应用要求，可选择线性结构或模块化结构来创建用户程序。

- 线性程序按顺序逐条执行处理自动化任务的所有指令。
通常，线性程序将所有程序指令都放入一个程序循环 OB (如 OB 1) 中以循环执行该程序。
- 模块化程序调用可执行特定任务的特定代码块。
要创建模块化结构，需要将复杂的自动化任务划分为与过程所执行的功能任务相对应的更小的次级任务。每个代码块都为各个次级任务提供程序段。
通过从另一个块中调用其中一个代码块来构建程序。

线性结构:



模块化结构:



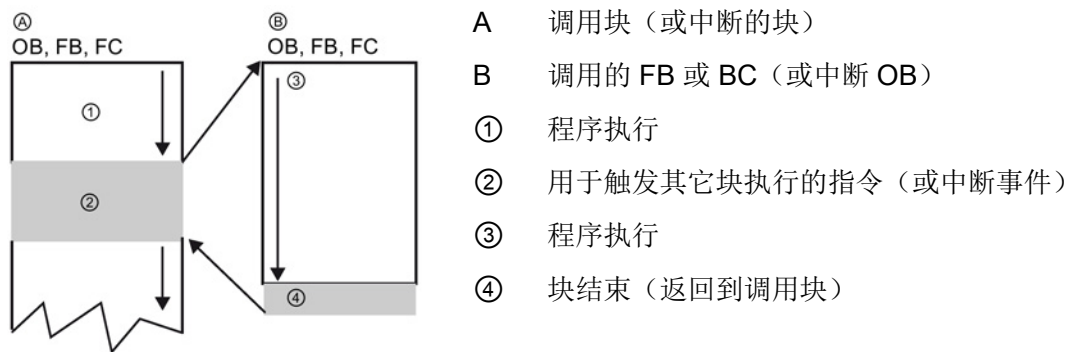
通过设计 FB 和 FC 执行通用任务，可创建模块化代码块。

然后可通过由其它代码块调用这些可重复使用的模块来构建用户程序。

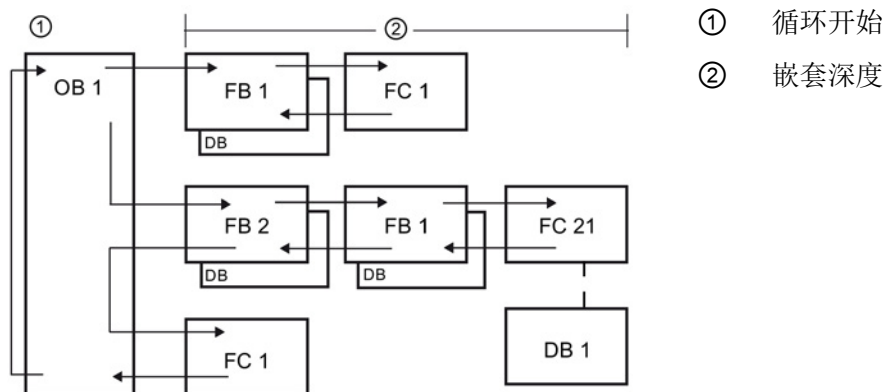
调用块将设备特定的参数传递给被调用块。当一个代码块调用另一个代码块时，CPU 会执行被调用块中的程序代码。执行完被调用块后，CPU 会继续执行调用块，并继续执行该块调用之后的指令。

6.1 轻松设计用户程序

还可以将 OB 分配给中断事件。发生该事件时，CPU 会执行相关 OB 中的程序代码。OB 的执行完成后，CPU 会在发生中断事件的时间点在用户程序中恢复执行，而该点可能在扫描过程中的任何位置。



可嵌套块调用以实现更加模块化的结构。在以下示例中，嵌套深度为 3：程序循环 OB 加 3 层对代码块的调用。



通过创建可在用户程序中重复使用的通用代码块，可简化用户程序的设计和实现。

- 可为标准任务创建能够重复使用的代码块，如用于控制泵或电机。也可以将这些通用代码块存储在可由不同的应用或解决方案使用的库中。
- 将用户程序构建到与功能任务相关的模块化组件中，可使程序的设计更易于理解和管理。模块化组件不仅有助于标准化程序设计，也有助于使更新或修改程序代码更加快速和容易。

- 创建模块化组件可简化程序的调试。
通过将整个程序构建为一组模块化程序段，可在开发每个代码块时测试其功能。
- 利用与特定功能任务相关的模块化设计，可以减少对已完成的应用程序进行调试所需的时间。

6.1.1 使用 OB 组织用户程序

组织块为程序提供结构。它们充当操作系统和用户程序之间的接口。OB 是由事件驱动的。事件（如诊断中断或时间间隔）会使 CPU 执行 OB。某些 OB 预定义了起始事件和行为。

程序循环 OB 包含用户主程序。用户程序中可包含多个程序循环 OB。RUN 模式期间，程序循环 OB 以最低优先级等级执行，可被其它事件类型中断。启动 OB 不会中断程序循环 OB，因为 CPU 在进入 RUN 模式之前将先执行启动 OB。

完成程序循环 OB 的处理后，CPU 会立即重新执行程序循环 OB。

该循环处理是用于可编程逻辑控制器的“正常”处理类型。

对于许多应用来说，整个用户程序位于一个程序循环 OB 中。

可创建其它 OB

以执行特定的功能，如用于处理中断和错误或用于以特定的时间间隔执行特定程序代码。这些 OB 会中断程序循环 OB 的执行。

6.1 轻松设计用户程序

使用“添加新块”(Add new block) 对话框在用户程序中创建新的 OB。



总是由事件驱动中断处理。发生此类事件时，CPU 会中断用户程序的执行并调用已组态用于处理该事件的 OB。完成中断 OB 的执行后，CPU 会在中断点继续执行用户程序。

CPU 按优先级确定处理中断事件的顺序。可为多个中断事件分配相同的优先级。更多相关信息，请参见组织块 (页 61)和执行用户程序 (页 60)。

创建附加 OB

可为用户程序，甚至为程序循环和启动 OB 事件创建多个 OB。使用“添加新块”(Add new block) 对话框创建 OB 并为 OB 输入名称。

如果为用户程序创建了多个程序循环 OB，则 CPU 会按数字顺序从具有最小编号（例如 OB 1）的程序循环 OB 开始执行每个程序循环 OB。例如：当第一个程序循环 OB（例如 OB 1）完成后，CPU 将执行下一个编号更高的程序循环 OB。

组态 OB 的属性

可对 OB 的属性进行修改。例如，可组态 OB 编号或编程语言。



说明

请注意，您可将局部过程映像编号分配给对应于 PIP0、PIP1、PIP2、PIP3 或 PIP4 的 OB。如果您为局部过程映像编号输入编号，则 CPU 将创建该过程映像分区。有关过程映像分区的说明，请参见主题“执行用户程序 (页 60)”。

6.1.2 FB 和 FC 使模块化任务编程变得很轻松

功能 (FC) 与子例程类似。 FC 是通常对一组输入值执行特定运算的代码块。FC 将此运算结果存储在存储单元中。使用 FC 可执行以下任务：

- 执行标准和可重复使用的运算，如数学计算
- 执行功能任务，如通过使用位逻辑运算进行单独控制

也可以在程序中的不同位置多次调用 FC。

此重复使用简化了对经常重复发生的任务的编程。

与 FB 不同，FC 不具有相关的背景 DB。FC 使用其临时存储器 (L) 保存用于计算运算的数据。不保存临时数据。要存储数据以备 FC 执行完成后使用，可将输出值赋给全局存储单元，如 M 存储器或全局 DB。

功能块 (FB) 与带存储器的子例程类似。 FB

是可通过块参数以编程方式实现其调用的代码块。FB 将输入 (IN)、输出 (OUT) 和输入/输出 (IN_OUT) 参数存储在数据块 (DB) 或“背景”DB 中的变量存储器内。背景 DB 提供与 FB 的实例（或调用）关联的一块存储区并在 FB 完成后存储数据。

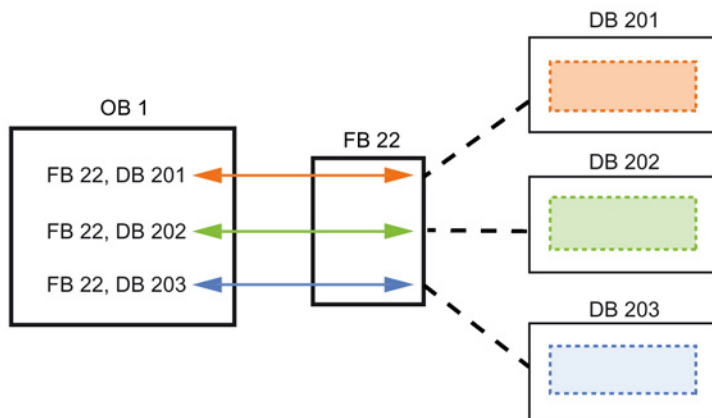
6.1 轻松设计用户程序

用户通常使用 **FB** 控制在一个扫描周期内未完成其运行的任务或设备的运行。要存储运行参数以便从一个扫描快速访问到下一个扫描，用户程序中的每一个 **FB** 都应具有一个或多个背景 **DB**。调用 **FB** 时，也会打开存储块参数的值以及用于该 **FB** 调用或“背景”的静态局部数据的背景 **DB**。这些值会在 **FB** 完成之后存储在背景数据块中。

可以给 **FB** 接口中的参数赋初值。这些值将传送到相关的背景 **DB** 中。如果未分配参数，将使用当前存储在背景 **DB** 中的值。某些情况下，必须分配参数。

可将不同的背景 **DB** 与 **FB** 的不同调用进行关联。通过背景 **DB** 可使用一个通用 **FB** 控制多个设备。通过使一个代码块对 **FB** 和背景 **DB** 进行调用，来构建程序。然后，**CPU** 执行该 **FB** 中的程序代码，并将块参数和静态局部数据存储在背景 **DB** 中。**FB** 执行完成后，**CPU** 会返回到调用该 **FB** 的代码块中。背景 **DB** 保留该 **FB** 实例的值。通过设计用于通用控制任务的 **FB**，可对多个设备重复使用 **FB**，方法是：为 **FB** 的不同调用选择不同的背景 **DB**。

下图显示了三次调用同一个 **FB** 的 **OB**，方法是针对每次调用使用一个不同的数据块。该结构使一个通用 **FB** 可以控制多个相似的设备（如电机），方法是在每次调用时为各设备分配不同的背景数据块。



每个背景 **DB** 存储单个设备的数据（如速度、加速时间和总运行时间）。在此实例中，**FB 22** 控制三个独立的设备，其中 **DB 201** 用于存储第一个设备的运行数据，**DB 202** 用于存储第二个设备的运行数据，**DB 203** 用于存储第三个设备的运行数据。

6.1.3 数据块为程序数据提供了便捷的存储方式

在用户程序中创建数据块 (DB) 以存储代码块的数据。

用户程序中的所有程序块都可访问全局 DB 中的数据，而背景 DB 仅存储特定功能块 (FB) 的数据。

用户程序可将数据存储在 CPU 的专用存储区中，如输入 (I)、输出 (Q) 和位存储器 (M)。此外，可使用数据块 (DB) 快速访问存储在程序本身中的数据。

当数据块关闭或相关代码块的执行结束时，DB 中存储的数据不会被删除。有两种类型的 DB：

- 全局 DB 存储程序中代码块的数据。任何 OB、FB 或 FC 都可访问全局 DB 中的数据。
- 背景 DB 存储特定 FB 的数据。背景 DB 中数据的结构反映了 FB 的参数 (Input、Output 和 InOut) 和静态数据。FB 的临时存储器不存储在背景 DB 中。

尽管背景 DB 反映特定 FB 的数据，然而任何代码块都可访问背景 DB 中的数据。

6.1.4 创建新代码块

要在程序中添加新的代码块，请按以下步骤操作：

1. 打开“程序块”(Program blocks) 文件夹。
2. 双击“添加新块”(Add new block)。
3. 在“添加新块”(Add new block) 对话框中单击要添加的块的类型。例如，单击“功能 (FC)”图标来添加 FC。
4. 从下拉菜单中为代码块选择编程语言。



5. 单击“确定”(OK) 将块添加到项目中。

选择“添加新对象并打开”(Add new and open) 选项（默认），让 STEP 7 在编辑器中打开新创建的块。

6.1.5 创建可重复使用的代码块



使用项目浏览器中“程序块”(Program blocks)下的“添加新块”(Add new block)对话框创建 OB、FB、FC 和全局 DB。

创建代码块时，需要为块选择编程语言。无需为 DB

选择语言，因为它仅用于存储数据。

选中“添加新对象并打开”(Add new and open)

复选框（默认），在项目视图中打开代码块。

可存储想要在库中重复使用的对象。每个项目都有一个与之相连的项目库。

除项目库外，您还可以创建可在多个项目中使用的任意数量的全局库。

由于库彼此兼容，因此可以复制库要素并将其从一个库移动到另一个库。

库可用于创建块的模板：首先将块粘贴到项目库中，随后在其中进一步开发块。

最后，将块从项目库复制到全局库。可将全局库共享给正在使用项目的其他同事。

他们可使用块并根据需要进一步调整块以满足各自的需求。

有关库操作的详细信息，请参见 STEP 7 在线帮助库主题。

6.1.6 从一个代码块调用另一个代码块



用户可以很容易地使用用户程序中的任何代码块（OB、FB 或 FC）调用 CPU 中的 FB 或 FC。

1. 打开将调用另一个块的代码块。
2. 在项目树中，选择要调用的代码块。
3. 将该块拖到所选网络中以创建一个对代码块的调用。

说明

用户程序无法调用 OB，因为 OB 是由事件驱动的 (页 62)。CPU 接收到事件时，将开始执行 OB 进行响应。

6.2 易于使用的编程语言

STEP 7 为 S7-1200 提供以下标准编程语言：

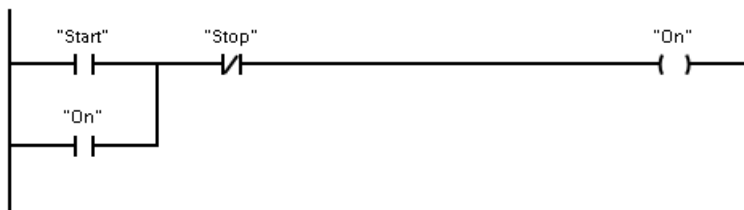
- LAD（梯形图逻辑）是一种图形编程语言。它使用基于电路图 (页 108) 的表示法。
- FBD（功能块图）是基于布尔代数 (页 109) 中使用的图形逻辑符号的编程语言。
- SCL（结构化控制语言）是一种基于文本的高级编程语言 (页 110)。

创建代码块时，应选择该块要使用的编程语言。

用户程序可以使用由任意或所有编程语言创建的代码块。

6.2.1 梯形图 (LAD)

电路图的元件（如常闭触点、常开触点和线圈）相互连接构成程序段。



要创建复杂运算逻辑，可插入分支以创建并行电路的逻辑。

并行分支向下打开或直接连接到电源线。用户可向上终止分支。

LAD 向多种功能（如数学、定时器、计数器和移动）提供“功能框”指令。

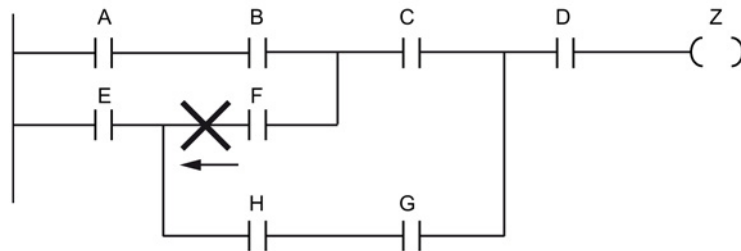
STEP 7 不限制 LAD 程序段中的指令（行和列）数。

说明

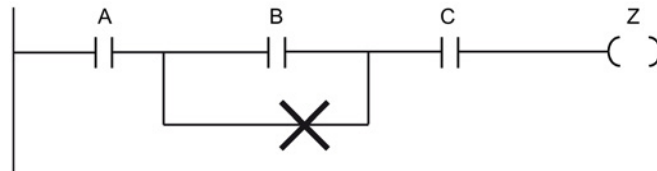
每个 LAD 程序段都必须使用线圈或功能框指令来终止。

创建 LAD 程序段时请注意以下规则：

- 不能创建可能导致反向能流的分支。



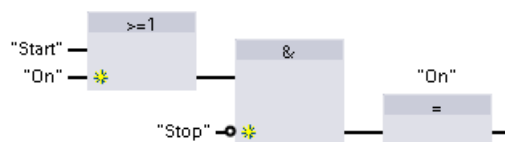
- 不能创建可能导致短路的分支。



6.2.2 功能块图 (FBD)

与 LAD 一样，FBD 也是一种图形编程语言。

逻辑表示法以布尔代数中使用的图形逻辑符号为基础。



要创建复杂运算的逻辑，在功能框之间插入并行分支。

算术功能和其它复杂功能可直接结合逻辑框表示。

STEP 7 不限制 FBD 程序段中的指令（行和列）数。

6.2.3 SCL 概述

结构化控制语言 (Structured Control Language, SCL) 是用于 SIMATIC S7 CPU 的基于 PASCAL 的高级编程语言。SCL 支持 STEP 7 的块结构。还可以将用 LAD 和 FBD 编写的程序块包括在用 SCL 编写的程序块中。

SCL 指令使用标准编程运算符，例如，用 (:=) 表示赋值，算术功能 (+ 表示相加，- 表示相减，* 表示相乘，/ 表示相除)。SCL 使用标准 PASCAL 程序控制操作，如 IF-THEN-ELSE、CASE、REPEAT-UNTIL、GOTO 和 RETURN。SCL 编程语言的语法元素可以使用 PASCAL 的任何引用方式。许多 SCL 的其它指令（如定时器和计数器）与 LAD 和 FBD 指令匹配。

由于 SCL 能像 ASCAL 一样提供条件处理、循环和嵌套控制结构，因此在 SCL 中可以比在 LAD 或 FBD 中更轻松地实现复杂的算法。

以下示例显示了用法不同的各种表达式：

<code>"C" := #A+#B;</code>	将两个局部变量的和赋给一个变量
<code>"Data_block_1".Tag := #A;</code>	为数据块变量赋值
<code>IF #A > #B THEN "C" := #A;</code>	IF-THEN 语句的条件
<code>"C" := SQRT (SQR (#A) + SQR (#B));</code>	SQRT 指令的参数

作为一种高级编程语言，SCL 使用标准语句实现基本任务：

- 赋值语句： :=
- 算术功能： +、-、* 和 /
- 全局变量寻址： "<变量名称>"（变量名称或数据块名称括在双引号内）
- 局部变量寻址： #<变量名称>（在变量名称前加“#”符号）
- 绝对寻址： %<绝对地址>，例如 %I0.0 或 %MW10

算术运算符可以处理各种数值数据类型。

结果的数据类型取决于最高有效操作数的数据类型。例如，使用 INT 操作数和 REAL 操作数的乘法运算会产生 REAL 结果值。

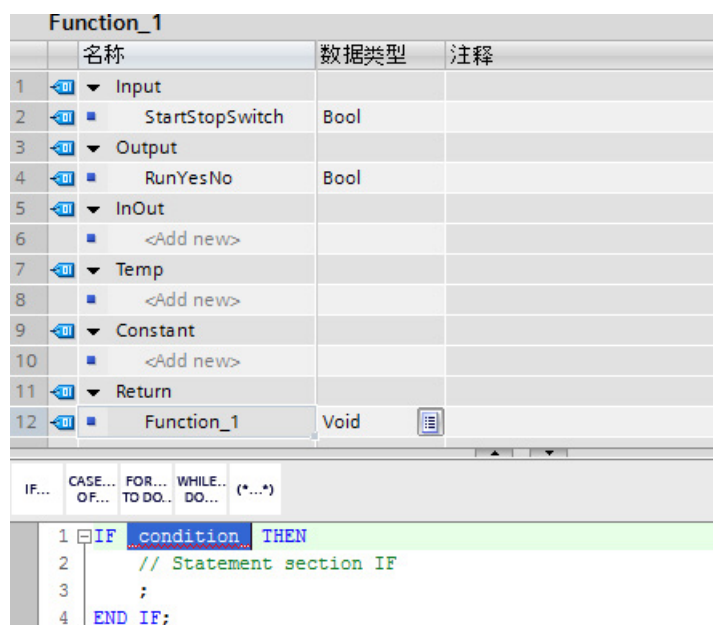
6.2.4 SCL 程序编辑器

可以在创建该块时指定任何块类型（OB、FB 或 FC）以便使用 SCL 编程语言。STEP 7 提供包含以下元素的 SCL 程序编辑器：

- 用于定义代码块参数的接口部分
- 用于程序代码的代码部分
- 包含 CPU 支持的 SCL 指令的指令树

可以直接在代码部分输入指令的 SCL 代码。

编辑器包含用于通用代码结构和注释的按钮。要了解更复杂的指令，只需从指令树拖动 SCL 指令并将其放入程序中。也可以使用任意文本编辑器创建 SCL 程序，然后将相应文件导入 STEP 7 中。



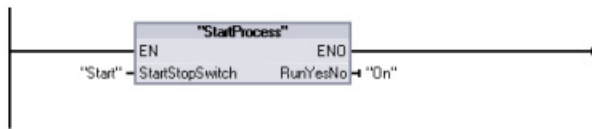
在 SCL 代码块接口部分，可以声明下列类型的参数：

- **Input、Output、InOut 和 Ret_Val:**
这些参数定义代码块的输入变量、输出变量和返回值。
执行代码块期间局部使用此处输入的变量名称。
通常不会使用变量表中的全局变量名称。
- **Static (仅适用于 FB，上述示例适用于 FC) :**
代码块使用静态变量在背景数据块中存储静态中间结果。
块会一直保留静态数据，直到多个周期后被覆盖。
块的名称（此块将其作为多重背景调用）也存储在静态局部数据中。

6.3 功能强大的指令使编程更加轻松

- Temp: 这些参数是执行代码块期间使用的临时变量。
- Constant: 这些是为代码块指定的常数值。

如果从其它代码块调用 SCL 代码块，该 SCL 代码块的参数会显示为输入或输出。



本示例中，“Start”和“On”变量（来自项目变量表）相当于 SCL 程序声明表中的“StartStopSwitch”和“RunYesNo”。

6.3 功能强大的指令使编程更加轻松

6.3.1 提供您所期望的基本指令

S7-1200 CPU 支持许多指令。这些指令在 STEP 7 的指令树中提供，分为以下几组：

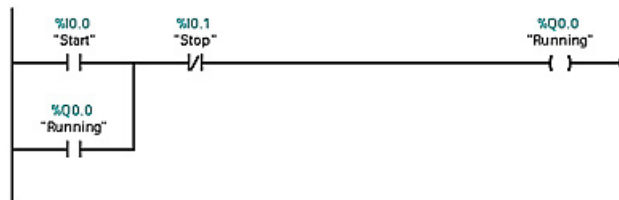
- 基本指令
- 扩展指令
- 工艺
- 通信指令

有关所有指令的完整概述，请参见“*S7-1200 Programmable Controller System Manual*”。该手册描述了许多常见指令。

位逻辑指令

位逻辑指令的基础是触点和线圈。



触点读取位的状态，而线圈则将操作的状态写入到位中。



触点可测试位的二进制状态，结果是在接通 (1) 时“有能流”，在断开 (0) 时“没有能流”。

线圈的状态反映前导逻辑的状态。

如果在多个程序位置中使用地址相同的线圈，则用户程序中最后一次运算的结果将决定输出更新期间写入物理输出中的值的状态。

常开 触点	常闭 触点	在赋的位值为 1 时，常开触点将闭合 (ON)。 在赋的位值为 0 时，常闭触点将闭合 (ON)。
"IN"	"IN"	
		

位逻辑运算的基本结构为 AND 逻辑或 OR 逻辑。以串联方式连接的触点创建 AND 逻辑程序段。以并联方式连接的触点创建 OR 逻辑程序段。

可将触点相互连接，创建用户自己的组合逻辑。

如果用户指定的输入位使用存储器标识符 I（输入）或 Q（输出），则从过程映像寄存器中读取位值。控制过程中的物理触点信号会连接到 PLC 上的输入端子。CPU

扫描已连接的输入信号并更新过程映像输入寄存器中的相应状态值。

通过在输入变量后加上“:P”（例如，“Motor_Start:P”或“I3.4:P”），可指定立即读取物理输入。对于立即读取，将直接从物理输入读取位数据值，而不是从过程映像中读取。立即读取不会更新过程映像。

输出线圈	反向输出线圈
"OUT"	"OUT"
	

记录以下通过输出和反向输出线圈的能流的输出结果：

- 如果有能流通过输出线圈，则输出位设置为 1。
- 如果没有能流通过输出线圈，则输出线圈位设置为 0。
- 如果有能流通过反向输出线圈，则输出位设置为 0。
- 如果没有能流通过反向输出线圈，则输出位设置为 1。

线圈输出指令写入输出位的值。如果用户指定的输出位使用存储器标识符 Q，则 CPU 接通或断开过程映像寄存器中的输出位，同时设置与能流状态相应的指定位。

控制执行器的输出信号连接到 PLC 的输出端子。在 RUN 模式下，CPU 系统将扫描输入信号，并根据程序逻辑处理输入状态，然后通过过程映像输出寄存器中设置新的输出状态值进行响应。在每个程序执行循环之后，CPU 都会将存储在过程映像寄存器中的新输出状态响应传送到已连接的输出端子。

通过在输出变量后加上“:P”（例如“Motor_On:P”或“Q3.4:P”），可指定立即写入物理输出。对于立即写入，会将位数据值写入到过程映像输出并直接写入到物理输出。

6.3 功能强大的指令使编程更加轻松

线圈并不局限于在程序段结尾使用。可以在 LAD 程序段的梯级中间以及触点或其它指令之间插入线圈。

NOT 触点反相器 (LAD) 带一个反向逻辑输入的 AND 功能框 (FBD) 带反向逻辑输入和输出的 AND 功能框 (FBD)

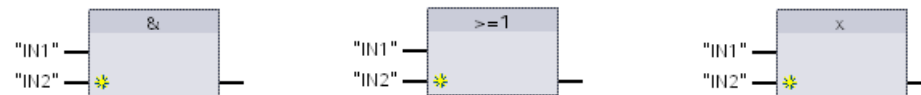


LAD NOT 触点用于对能流输入的的逻辑状态取反。

- 如果没有能流流入 NOT 触点，则会有能流流出。
- 如果有能流流入 NOT 触点，则没有能流流出。

对于 FBD 编程，可从“收藏夹”(Favorites) 工具栏或指令树中拖动“取反 RLO”(Invert RLO) 工具，然后将其放置在输入或输出端以在该功能框连接器上创建逻辑反相器。

AND 功能框 (FBD) OR 功能框 (FBD) XOR 功能框 (FBD)



- AND 功能框的所有输入必须都为 TRUE，输出才为 TRUE。
- OR 功能框只要有一个输入为 TRUE，输出就为 TRUE。
- XOR 功能框必须有奇数个输入为 TRUE，输出才为 TRUE。

在 FBD 编程中，LAD 的触点程序段由与 (&)、或 (>=1) 和异或 (x) 功能框程序段表示，可在其中为功能框输入和输出指定位值。

也可以连接到其它逻辑框，创建用户自己的逻辑组合。

在程序段中放置功能框后，可从“收藏夹”(Favorites)

工具栏或指令树中拖动“插入输入”(Insert input)

工具，然后将其放置在功能框的输入侧以添加更多输入。

也可以右键单击功能框输入连接器并选择“插入输入”(Insert input)。

功能框输入和输出可连接到其它逻辑框，也可输入未连接输入的位地址或位符号名称。

执行功能框指令时，当前输入状态会应用到二进制功能框逻辑，如果为真，功能框输出将为真。

6.3.2 比较器和移动指令

比较运算用于对数据类型相同的两个值进行比较。

表格 6-1 比较运算

指令	SCL	说明
LAD: 	<pre> out := in1 = in2; out := in1 <> in2; out := in1 >= in2; out := in1 <= in2; out := in1 > in2; out := in1 < in2; </pre>	<ul style="list-style-type: none"> • 等于 (==): 如果 IN1 等于 IN2, 则比较结果为真 • 不等于 (<>): 如果 IN1 不等于 IN2, 则比较结果为真。 • 大于或等于 (>=): 如果 IN1 大于或等于 IN2, 则比较结果为真 • 小于或等于 (<=): 如果 IN1 小于或等于 IN2, 则比较结果为真 • 大于 (>): 如果 IN1 大于 IN2, 则比较结果为真。 • 小于 (<): 如果 IN1 小于 IN2, 则比较结果为真。
FBD: 		

1 对于 LAD 和 FBD: 如果比较结果为 TRUE, 则触点将被激活 (LAD) 或者功能框输出为 TRUE (FBD)。

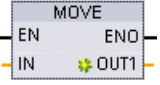
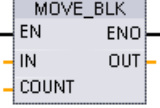
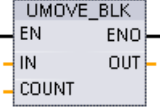
有关更多比较运算, 请参见“*S7-1200 可编程控制器系统手册*”。

移动操作用于将数据元素复制到新的存储器地址, 并可以从一种数据类型转换为另一种数据类型。移动过程不会更改源数据。

- MOVE 将存储在指定地址的数据元素复制到新地址。要添加其它输出, 请单击 OUT1 参数旁的图标。
- MOVE_BLK (可中断移动) 和 UMOVE_BLK (不可中断移动) 可将数据元素块复制到新地址。MOVE_BLK 和 UMOVE_BLK 指令具有附加的 COUNT 参数。COUNT 指定要复制的数据元素个数。每个被复制元素的字节数取决于 PLC 变量表中分配给 IN 和 OUT 参数变量名称的数据类型。

6.3 功能强大的指令使编程更加轻松

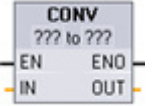
表格 6-2 MOVE、MOVE_BLK 和 UMOVE_BLK 指令

LAD/FBD	SCL	说明
	<pre>out1 := in;</pre>	<p>将存储在指定地址的数据元素复制到新地址或多个地址。要在 LAD 或 FBD 中添加其它输出，单击输出参数旁的图标。对于 SCL，请使用多个赋值语句。还可以使用任一循环结构。</p>
	<pre>MOVE_BLK(in:=_variant_in, count:=_uint_in, out=>_variant_out);</pre>	<p>将数据元素块复制到新地址的可中断移动。</p>
	<pre>UMOVE_BLK(in:=_variant_in, count:=_uint_in out=>_variant_out);</pre>	<p>将数据元素块复制到新地址的不可中断移动。</p>

有关更多移动操作，请参见“S7-1200 系统手册”。



6.3.3 转换操作

表格 6-3 转换操作

LAD/FBD	SCL	说明
	<pre>out := <data type in>_TO_<data type out>(in);</pre>	<p>将数据元素从一种数据类型转换为另一种数据类型。</p>

- 对于 LAD 和 FBD：在功能框名称下方单击，并从下拉菜单中选择数据类型。选择（转换源）数据类型之后，（转换目标）下拉列表中显示可能的转换项列表。
- 对于 SCL：通过识别输入参数 (in) 和输出参数 (out) 的数据类型来构造转换指令。例如，DWORD_TO_REAL 将 DWord 值转换为 Real 值。

表格 6-4 Round 和 Truncate 指令

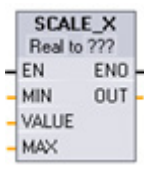
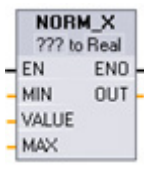
LAD/FBD	SCL	说明
	<pre>out := ROUND(in);</pre>	<p>将实数（Real 或 LReal）转换为整数。 指令将实数舍入为最接近的整数值（IEEE - 舍入为最接近值）。</p> <p>如果该数值刚好是两个连续整数的一半（例如，10.5），则指令将其取整为偶数。例如：</p> <ul style="list-style-type: none"> • ROUND (10.5) = 10 • ROUND (11.5) = 12 <p>对于 LAD/FBD，请单击指令框中的“???”来选择输出的数据类型，例如“DInt”。对于 SCL，默认输出数据类型为 DINT。要舍入为另一输出数据类型，请输入具有数据类型的显式名称的指令名称，例如 ROUND_REAL 或 ROUND_LREAL。</p>
	<pre>out := TRUNC(in);</pre>	<p>将实数（Real 或 LReal）转换为整数。 实数的小数部分被截成零（IEEE - 取整为零）。</p>

表格 6-5 上取整 (CEIL) 和 Floor 指令

LAD/FBD	SCL	说明
	<pre>out := CEIL(in);</pre>	<p>将实数（Real 或 LReal）转换为最接近的大于或等于所选实数的整数（IEEE“向正无穷取整”）。</p>
	<pre>out := FLOOR(in);</pre>	<p>将实数（Real 或 LReal）转换为最接近的小于或等于所选实数的整数（IEEE“向正无穷取整”）。</p>

6.3 功能强大的指令使编程更加轻松

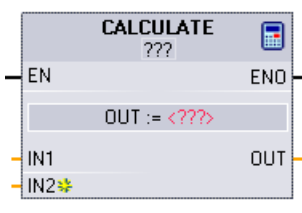
表格 6-6 SCALE_X 和 NORM_X 指令

LAD/FBD	SCL	说明
	<pre>out := SCALE_X(min:=_in_, value:=_in_, max:=_in_);</pre>	<p>按参数 MIN 和 MAX 所指定的数据类型和值范围对标准化的实参数 VALUE (其中, 0.0 <= VALUE <= 1.0) 进行标定:</p> $OUT = VALUE (MAX - MIN) + MIN$
	<pre>out := NORM_X(min:=_in_, value:=_in_, max:=_in_);</pre>	<p>标准化通过参数 MIN 和 MAX 指定的值范围内的参数 VALUE:</p> $OUT = (VALUE - MIN) / (MAX - MIN),$ <p>其中 (0.0 <= OUT <= 1.0)</p>

1 等效 SCL: $out := value (max-min) + min;$ 2 等效 SCL: $out := (value-min) / (max-min);$

6.3.4 计算指令使数字运算更加轻松

表格 6-7 CALCULATE 指令

LAD/FBD	SCL	说明
	<p>使用标准 SCL 数学表达式创建等式。</p>	<p>CALCULATE 指令可用于创建作用于多个输入上的数学函数 (IN1, IN2, .. INn), 并根据您定义的等式在 OUT 处生成结果。</p> <ul style="list-style-type: none"> • 首先选择数据类型。所有输入和输出的数据类型必须相同。 • 要添加其它输入, 请单击最后一个输入处的图标。

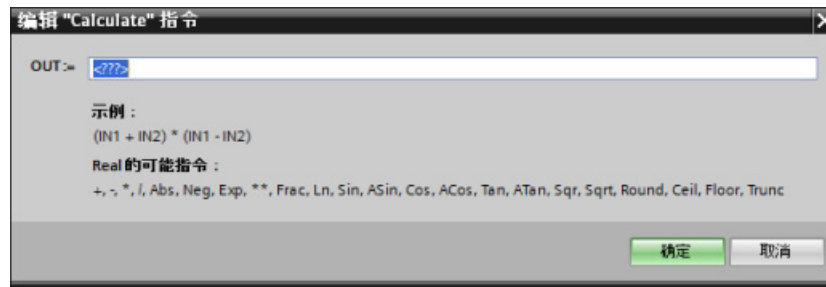
表格 6-8 参数的数据类型

参数	数据类型 ¹
IN1, IN2, ..INn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord

- ¹ IN 和 OUT 参数必须具有相同的数据类型（通过对输入参数进行隐式转换）。例如：如果 OUT 是 INT 或 REAL，则 SINT 输入值将转换为 INT 或 REAL 值

单击计算器图标可打开对话框，在其中定义数学函数。输入等式作为输入（如 IN1 和 IN2）和操作数。单击“确定”(OK) 保存函数时，对话框会自动生成 CALCULATE 指令的输入。

对话框显示一个示例，以及可根据 OUT 参数的数据类型加入的一系列指令：



说明

还必须为函数中的任何常量生成输入。然后会在指令 CALCULATE 的相关输入中输入该常量值。

通过输入常量作为输入，可将 CALCULATE 指令复制到用户程序的其它位置，从而无需更改函数。之后，不需要修改函数，就可以更改指令输入的值或变量。

当执行 CALCULATE 并成功完成计算中的所有单个运算时，ENO = 1，否则 ENO = 0。

有关 CALCULATE 指令的示例，请参见“为复杂数学等式使用 CALCULATE 指令 (页 50)”。

6.3.5 定时器操作

S7-1200 支持以下定时器

- TP 定时器可生成具有预设宽度时间的脉冲。
- TON 定时器在预设的延时过后将输出 (Q) 设置为 ON。
- TOF 定时器在预设的延时过后将输出 (Q) 设置为 ON，然后将输出复位为 OFF。
- TONR 定时器在预设的延时过后将输出 (Q) 设置为 ON。在使用复位 (R) 输入复位经过的时间之前，会一直累加多个定时时段内经过的时间。
- PT (预设定时器) 线圈会在指定的定时器中装载新的预设时间值。
- RT (预设定时器) 线圈会复位指定的定时器。

对于 LAD 和 FBD，这些指令通过功能框指令或输出线圈的形式提供。

用户程序中可以使用的定时器数仅受 CPU 存储器容量限制。每个定时器占用 16 个字节的存储器空间。

每个定时器都使用一个存储在数据块中的结构来保存定时器数据。对于 SCL，必须首先为各个定时器指令创建 DB 方可引用相应指令。对于 LAD 和 FBD，STEP 7 会在插入指令时自动创建 DB。

创建 DB 时，还可以使用多重背景数据块。由于定时器数据位于单个 DB 中，且不需要为每个定时器使用单独的 DB，因此会缩短处理定时器的处理时间。在共享的多重背景数据块中的定时器数据结构之间不存在交互作用。

表格 6-9 TP (脉冲定时器)

LAD/FBD	SCL	时序图
	<pre>"timer_db".TP(IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	

表格 6-10 TON（接通延迟定时器）

LAD/FBD	SCL	时序图
	<pre>"timer_db".TON(IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	

表格 6-11 TOF（关断延迟定时器）

LAD/FBD	SCL	时序图
	<pre>"timer_db".TOF(IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	

表格 6-12 TONR（保持型接通延迟定时器）

LAD/FBD	SCL	时序图
	<pre>"timer_db".TONR(IN:=_bool_in_, R:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	

6.3 功能强大的指令使编程更加轻松

表格 6- 13 预设定时器 -(PT)- 和重置定时器 -(RT)- 线圈命令

LAD/FBD	SCL	说明
	<pre>PRESET_TIMER(PT:=_time_in_, TIMER:=_iec_timer_in_) ;</pre>	<p>与功能框定时器或线圈定时器一起使用预设定时器 -(PT)- 和重置定时器 -(RT)- 线圈指令。</p> <p>可将这些线圈指令置于中间位置。</p> <p>线圈输出能流状态始终与线圈输入状态相同。</p> <ul style="list-style-type: none"> • 激活 -(PT)- 线圈时，指定 IEC_Timer DB 数据的 PRESET 时间元素设置为“PRESET_Tag”持续时间。 • 激活 -(RT)- 线圈时，指定 IEC_Timer DB 数据的 ELAPSED 时间元素复位为 0。
	<pre>RESET_TIMER(_iec_timer_in_);</pre>	<p>与功能框定时器或线圈定时器一起使用预设定时器 -(PT)- 和重置定时器 -(RT)- 线圈指令。</p> <p>可将这些线圈指令置于中间位置。</p> <p>线圈输出能流状态始终与线圈输入状态相同。</p> <ul style="list-style-type: none"> • 激活 -(PT)- 线圈时，指定 IEC_Timer DB 数据的 PRESET 时间元素设置为“PRESET_Tag”持续时间。 • 激活 -(RT)- 线圈时，指定 IEC_Timer DB 数据的 ELAPSED 时间元素复位为 0。

表格 6- 14 参数的数据类型

参数	数据类型	说明
功能框： IN 线圈： 能流	Bool	TP、TON 和 TONR： 功能框： 0=禁用定时器， 1=启用定时器 线圈： 无能流=禁用定时器， 能流=启用定时器 TOF： 功能框： 0=启用定时器， 1=禁用定时器 线圈： 无能流=启用定时器， 能流=禁用定时器
R	Bool	仅 TONR 功能框： 0=不重置 1= 将经过的时间和 Q 位重置为 0
功能框： PT 线圈： "PRESET_Tag"	Time	定时器功能框或线圈： 预设的时间输入
功能框： Q 线圈： DBdata.Q	Bool	定时器功能框： Q 功能框输出或定时器 DB 数据中的 Q 位 定时器线圈： 仅可寻址定时器 DB 数据中的 Q 位
功能框： ET 线圈： DBdata.ET	Time	定时器功能框： ET（经历的时间）功能框输出或定时器 DB 数据中的 ET 时间值 定时器线圈： 仅可寻址定时器 DB 数据中的 ET 时间值。

表格 6-15 PT 和 IN 参数值变化的影响

定时器	PT 和 IN 功能框参数和相应线圈参数的变化
TP	<ul style="list-style-type: none"> • 定时器运行期间，更改 PT 没有任何影响。 • 定时器运行期间，更改 IN 没有任何影响。
TON	<ul style="list-style-type: none"> • 定时器运行期间，更改 PT 没有任何影响。 • 定时器运行期间，将 IN 更改为 FALSE 会复位并停止定时器。
TOF	<ul style="list-style-type: none"> • 定时器运行期间，更改 PT 没有任何影响。 • 定时器运行期间，将 IN 更改为 TRUE 会复位并停止定时器。
TONR	<ul style="list-style-type: none"> • 定时器运行期间，更改 PT 没有任何影响，但对定时器中断后继续运行会有影响。 • 定时器运行期间，将 IN 更改为 FALSE 会停止定时器但不会复位定时器。将 IN 重新变为 TRUE 将使定时器从累积的时间值开始定时。

PT（预设时间）和 ET（经过的时间）值存储在指定 IEC_TIMER DB 数据中，以有符号双整型形式表示毫秒时间。TIME 数据使用 T# 标识符，可以简单时间单元（T#200ms 或 200）或复合时间单元（如 T#2s_200ms）的形式输入。

表格 6-16 TIME 数据类型的大小和范围

数据类型	大小	有效数值范围 ¹
TIME	32 位，以 DInt 数据的形式存储	T#-24d_20h_31m_23s_648ms 到 T#24d_20h_31m_23s_647ms 以 -2,147,483,648 ms 到 +2,147,483,647 ms 的形式存储

¹ 在定时器指令中，无法使用上面所示 TIME 数据类型的负数范围。负的 PT（预设时间）值在定时器指令执行时被设置为零。ET（经过的时间）始终为正值。

定时器编程

规划和创建用户程序时应考虑以下定时器运行说明：

- 可在同一个扫描周期内多次更新定时器。
每次执行定时器指令（TP、TON、TOF、TONR）和每次将定时器结构的 ELAPSED 或 Q 成员用作其它已执行指令的参数时，都会更新定时器。
这在需要最新时间数据（本质上是立即读取定时器）时会是一项优点。
但是，如果希望在整个程序扫描周期内保持一致的值，则请将定时器指令放置在需要这些值的其它所有指令之前，并使用定时器指令的 Q 和 ET 输出中的变量而不是定时器 DB 结构的 ELAPSED 和 Q 成员。
- 扫描期间可以不执行定时器更新。
可以在函数中启动定时器，然后在一个或多个扫描周期内不再调用该函数。
如果没有执行引用定时器结构中 ELAPSED 或 Q 成员的其它指令，则不会更新定时器。
直到再次执行定时器指令或执行将定时器结构的 ELAPSED 或 Q 用作参数的其它指令时，才会再次更新定时器。
- 尽管并不常见，但可以将同一个 DB 定时器结构分配给多个定时器指令。
通常，为避免意外交互作用，应当使每个 DB 定时器结构仅对应一个定时器指令（TP、TON、TOF、TONR）。

自复位定时器适合用于触发需要周期性发生的动作。

通常，将引用定时器位的常闭触点放置在定时器指令前面可创建自复位定时器。

该定时器网络通常位于使用该定时器位来触发动作的一个或多个依赖型网络上。

当定时器时间已到（经过的时间达到预设值）时，定时器位将在一个扫描周期内为 ON，因而可执行由该定时器位控制的依赖型网络逻辑。

下次执行定时器网络时，常闭触点将为 OFF，从而复位定时器并清除定时器位。

下次扫描期间，常闭触点将为 ON，因此将重启定时器。

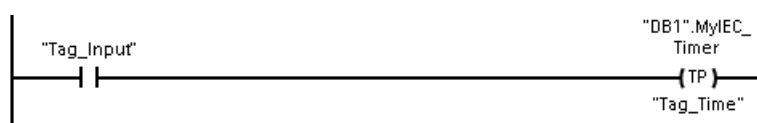
创建此类自复位定时器时，请勿将定时器 DB

结构的“Q”成员用作该定时器指令前面常闭触点的参数。

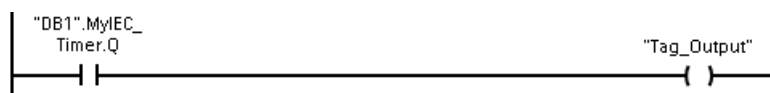
而是要使用与该定时器指令的“Q”输出相连的变量。如果访问定时器 DB 结构的 Q 成员，将导致定时器更新，且如果因常闭触点而更新定时器，该触点将立即复位该定时器。定时器指令的 Q 输出将在一个扫描周期内不为 ON，并且依赖型网络不会执行。

-(TP)-、-(TON)-、-(TOF)- 和 -(TONR)- 定时器线圈必须是网络中的最后一个指令。

如定时器示例中所示，后面网络中的触点指令会求出定时器线圈 IEC_Timer DB 数据中的 Q 位值。同样，如果要在程序中使用经过的时间值，必须访问 IEC_timer DB 数据中的 ELAPSED 元素。



当 Tag_Input 位的值由 0 转换为 1 时，脉冲定时器启动。定时器开始运行并持续 Tag_Time 时间值指定的时间。



只要定时器运行，就存在 DB1.MyIEC_Timer.Q 状态=1 且 Tag_Output 值=1。当经过 Tag_Time 值后，DB1.MyIEC_Timer.Q=0 且 Tag_Output 值=0。

6.3.6 计数器操作

可使用计数器指令对内部程序事件和外部过程事件进行计数。

- 输入参数 CU 的值从 0 变为 1 时，“加计数”计数器 (CTU) 就会加 1。
- 输入参数 CD 的值从 0 变为 1 时，“减计数”计数器 (CTD) 就会减 1。
- “加计数和减计数”计数器 (CTUD) 在加计数 (CU) 或减计数 (CD) 输入从 0 转换为 1 时加 1 或减 1。

S7-1200 还提供高速计数器 (页 136) (HSC)，用于计算发生速率快于 OB 执行速率的事件。

CU、CD 和 CTUD 指令使用软件计数器，软件计数器的最大计数速率受其所在 OB 的执行速率限制。

说明

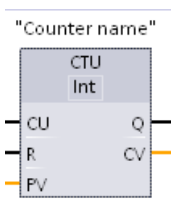
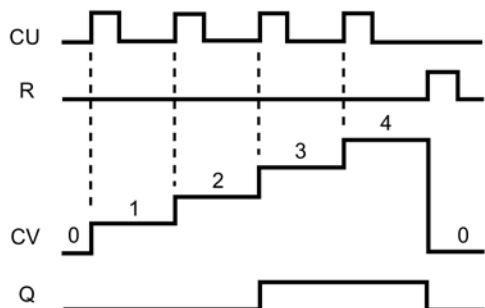
如果待计数事件的发生速率处于 OB 执行速率范围内，请使用 CTU、CTD 或 CTUD 计数器指令。如果事件的发生速率快于 OB 的执行速率，则应使用 HSC。

每个计数器都使用数据块中存储的结构来保存计数器数据。对于 SCL，必须首先为各个计数器指令创建 DB 方可引用相应指令。对于 LAD 和 FBD，STEP 7 会在插入指令时自动创建 DB。

6.3 功能强大的指令使编程更加轻松

用户程序中可以使用的计数器数仅受 CPU 存储器容量限制。各个计数器使用 3 个字节（表示 SInt 或 USInt）、6 个字节（表示 Int 或 UInt）或 12 个字节（表示 DInt 或 UInt）。

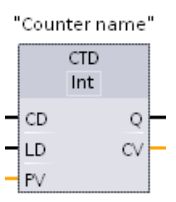
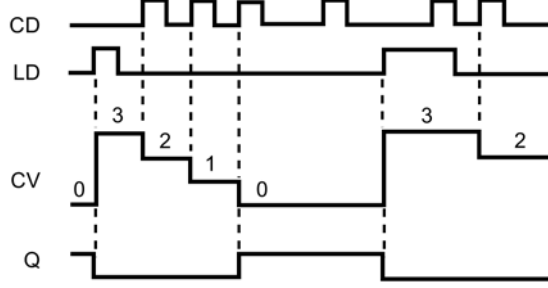
表格 6-17 CTU（加计数）计数器

LAD/FBD	SCL	操作
 <p>"Counter name"</p> <p>CTU Int</p> <p>CU Q R CV PV</p>	<pre>"ctu_db".CTU(CU:=_bool_in, R:=_bool_in, PV:=_in_, Q=>_bool_out, CV=>_out_);</pre>	 <p>CU R CV Q</p>

如上时序图显示了具有无符号整数计数值的 CTU 计数器的运行（其中 PV = 3）。

- 如果参数 CV（当前计数值）的值大于或等于参数 PV（预设计数值）的值，则计数器输出参数 Q = 1。
- 如果复位参数 R 的值从 0 变为 1，则 CV 复位为 0。

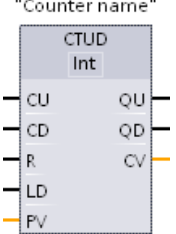
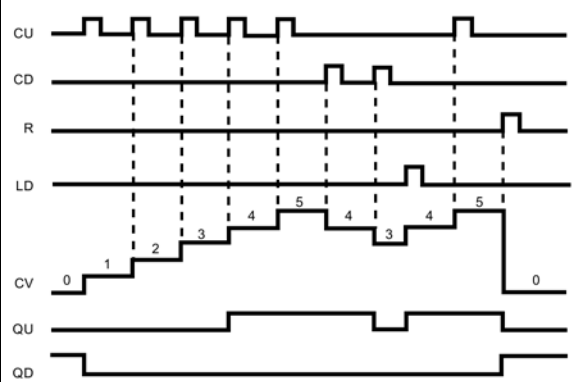
表格 6-18 CTD（减计数）计数器

LAD/FBD	SCL	操作
 <p>"Counter name"</p> <p>CTD Int</p> <p>CD Q LD CV PV</p>	<pre>"ctd_db".CTD(CD:=_bool_in, LD:=_bool_in, PV:=_in_, Q=>_bool_out, CV=>_out_);</pre>	 <p>CD LD CV Q</p>

如上时序图显示了具有无符号整数计数值的 CTD 计数器的运行（其中 PV = 3）。

- 如果参数 CV（当前计数值）的值等于或小于 0，则计数器输出参数 Q = 1。
- 如果参数 LD 的值从 0 变为 1，参数 PV 的值（预设值）将作为新的 CV 加载到计数器。

表格 6-19 CTUD（加计数和减计数）计数器

LAD/FBD	SCL	操作
<p>"Counter name"</p> 	<pre>"ctud_db".CTUD(CU:=_bool_in, CD:=_bool_in, R:=_bool_in, LD:=_bool_in, PV:=_in_, QU=>_bool_out, QD=>_bool_out, CV=>_out_);</pre>	


如上时序图显示了具有无符号整数计数值的 CTUD 计数器的运行（其中 PV = 4）。

- 如果参数 CV（当前计数值）的值等于或大于参数 PV（预设值）的值，则计数器输出参数 QU = 1。
- 如果参数 CV 的值小于或等于零，则计数器输出参数 QD = 1。
- 如果参数 LD 的值从 0 变为 1，则参数 PV 的值将作为新的 CV 加载到计数器。
- 如果复位参数 R 的值从 0 变为 1，则 CV 复位为 0。

6.3.7 脉冲宽度调制 (PWM)

CTRL_PWM 指令在扩展指令的“脉冲”组中提供。

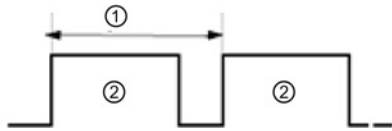
表格 6-20 CTRL_PWM 指令

LAD/FBD	SCL	说明
	<pre>"ctrl_pwm_db"(PWM:=W#16#0, ENABLE:=False, BUSY=>_bool_out_, STATUS=>_word_out_);</pre>	<p>CTRL_PWM 指令可提供占空比可变的固定循环时间输出。PWM 输出以指定频率（循环时间）启动之后将持续运行。脉冲宽度会根据需要进行变化以影响所需的控制。</p>

在代码块中插入 CTRL_PWM 指令时，可从“调用选项”(Call options) 对话框为指令创建 DB。CTRL_PWM 指令将参数信息存储在 DB 中，并控制数据块参数。

6.3 功能强大的指令使编程更加轻松

CPU 第一次进入 RUN 模式时，脉冲宽度将设置为在设备配置中组态的初始值。根据需要将值写入设备配置中指定的字长度输出 (Q) 地址 (“输出地址”/“起始地址”) 以更改脉冲宽度。使用指令 (例如, Move、Convert、数学运算或 PID) 将指定的脉冲宽度写入相应的字长度输出 (Q)。必须使用输出值的有效范围 (百分数、千分数、万分数或 S7 模拟格式)。



占空比可表示为循环时间的百分数或相对量 (例如, 0 到 1000, 或者 0 到 10000)。脉冲宽度可从 0 (无脉冲, 始终关闭) 到满刻度 (无脉冲, 始终打开) 变化。

- ① 循环时间
- ② 脉冲宽度时间

PWM 输出可在 0 到满量程之间变化, 因此可提供在许多方面都与模拟量输出相同的数字量输出。例如, PWM 输出可用于控制电机的速度, 速度范围可以从停止到全速; 也可用于控制阀的位置, 位置范围可以从闭合到完全打开。

6.4 轻松创建数据日志

控制程序可以使用 **Data log** 指令将运行数据值存储在永久性日志文件中。数据日志文件存储在闪存（CPU 或存储卡）中。日志文件数据按照标准 CSV（逗号分隔值）格式进行存储。数据记录按大小预定的循环日志文件形式组织。

Data log 指令用于在程序中创建、打开、写入记录以及关闭日志文件。

通过创建定义单个日志记录的数据缓冲区来确定要记录的程序值。

数据缓冲区用作新日志记录的临时存储空间。

运行期间，必须通过程序将新的过程值移到缓冲区中。

更新所有过程数据值之后，即可执行 **DataLogWrite**

指令，以将数据由缓冲区传送至数据日志记录中。

可以从 Web

服务器的“文件浏览器”页面打开、编辑、保存、重命名或删除数据日志文件。

必须有读取权限才能查看文件浏览器，必须有修改权限才能编辑、删除或重命名数据日志文件。

使用 **DataLog** 指令通过程序将运行期过程数据存储 CPU 的闪存中。

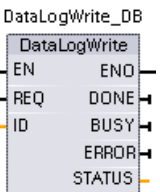
数据记录按大小预定的循环日志文件形式组织。新记录会不断添加到数据日志文件。

当数据日志文件存储的记录达到最大数量时，随后写入的记录会覆盖最早的记录。

为了避免覆盖任何数据记录，请使用 **DataLogNewFile** 指令。

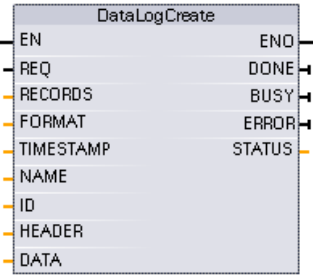
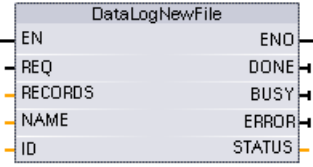
这样，新数据记录会存储在新数据日志文件中，而旧数据日志文件仍保留在 CPU 中。

表格 6-21 DataLogWrite 指令

LAD/FBD	SCL	说明
	<pre>"DataLogWrite_DB" (req:=FALSE, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>DataLogWrite 用于将数据记录写入指定的数据日志。必须打开已有目标数据日志。</p> <p>必须通过程序用当前运行期数据值装载记录缓冲区，然后执行 DataLogWrite 指令将新记录数据由缓冲区移到数据日志中。</p> <p>如果未完成 DataLogWrite 操作时出现电源故障，则可能丢失当前正传送到数据日志中的数据记录。</p>

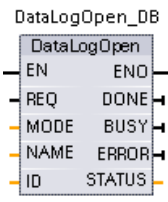
6.4 轻松创建数据日志

表格 6- 22 DataLogCreate 和 DataLogNewFile 指令

LAD/FBD	SCL	说明
<p>DataLogCreate_ DB</p> 	<pre>"DataLogCreate_DB" (req:=FALSE, records:=1, format:=1, timestamp:=1, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_variant_in_, ID:=_dword_inout_, header:=_variant_inout_, data:=_variant_inout_);</pre>	<p>DataLogCreate¹</p> <p>用于创建和初始化存储在 CPU \DataLogs 目录中的数据日志文件。采用预定的固定大小创建数据日志文件。</p>
<p>DataLogNewFile_ DB</p> 	<pre>"DataLogNewFile_DB" (req:=FALSE, records:=1, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_variant_in_, ID:=_dword_inout_);</pre>	<p>DataLogNewFile¹</p> <p>允许程序根据现有数据日志文件创建新的数据日志文件。将创建新的数据日志，并根据指定的 NAME 隐式打开该日志。从原始数据日志复制标题记录以及原始数据日志属性。隐式关闭原始数据日志文件。</p>

¹ DataLogCreate 和 DataLogNewFile 操作可能持续多个程序扫描周期。创建日志文件所需的实际时间取决于记录结构和记录数。只有在程序逻辑监测到 DONE 位变为 TRUE 后，新的数据日志才可用于其它数据日志操作。

表格 6-23 DataLogOpen 和 DataLogClose 指令

LAD/FBD	SCL	说明
	<pre>"DataLogOpen_DB" (req:=FALSE, mode:=0, name:=_variant_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>DataLogOpen 指令用于打开已有数据日志文件。必须先打开数据日志，才能向该日志写入新记录。可单独打开和关闭各个数据日志。可同时打开八个数据日志。</p>
	<pre>"DataLogClose_DB" (req:=FALSE, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>DataLogClose 指令用于关闭已打开的数据日志文件。对已关闭的数据日志执行 DataLogWrite 操作将导致错误。再次执行 DataLogOpen 操作之前，禁止对此数据日志执行写操作。切换为 STOP 模式时将关闭所有已打开的数据日志文件。</p>

6.5 轻松监视和测试用户程序

6.5.1 监视表格和强制表格

使用“监视表格”监视和修改正在由在线 CPU 执行的用户程序的值。
可在项目中创建并保存不同的监视表格以支持各种测试环境。
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。

通过监视表格，可监视 CPU 并与 CPU 交互，如同 CPU 执行用户程序一样。
不仅可以显示或更改代码块和数据块的变量值，还可以显示或更改 CPU 存储区的值，包括输入和输出 (I 和 Q)、外围设备输入 (I:P)、位存储器 (M) 和数据块 (DB)。

通过监视表格，可在 STOP 模式下启用 CPU 的物理输出 (Q:P)。例如，测试 CPU 的接线时可为输出端赋特定值。

STEP 7 还提供强制表格，用于将变量“强制”设为特定值。
有关强制的更多信息，请参见“在线和诊断”一章的 CPU 中的强制值 (页 369)一节。

说明

强制值存储在 CPU 中，而不是监视表格中。
无法强制输入（或“I”地址）。但是，可以强制外围设备输入。
要强制外围设备输入，请在地址后面添加一个“P”（例如：“On:P”）。

同时，STEP 7 还提供根据触发条件跟踪并记录程序变量 (页 383)的功能。

6.5.2 用于显示使用情况的交叉引用

巡视窗口可显示有关所选对象在整个项目中使用情况的交叉引用信息，例如用户程序、CPU 以及任何 HMI 设备。“交叉引用”(Cross-reference)
选项卡显示使用了所选对象的实例和使用该对象的其它对象。
巡视窗口还包括交叉引用中仅在线可用的块。
要显示交叉引用，请选择“显示交叉引用”(Show cross-references) 命令。
(在项目视图中，可在“工具”(Tools) 菜单中找到交叉引用。)

说明

不必关闭编辑器即可看到交叉引用信息。

可以对交叉引用中的条目进行排序。

交叉引用列表提供用户程序中存储器地址和变量的使用概况。

- 创建和更改程序时，用户始终能够掌握所使用的操作数、变量和块调用情况。
- 从交叉引用可直接跳转到操作数和变量的使用位置。
- 在程序测试或故障排除期间，系统会通知您哪个块中的哪条命令在处理哪个存储单元、哪个画面在使用哪个变量，以及哪个块被其它哪个块调用。

表格 6- 24 交叉引用的元素

列	说明
对象 (Object)	使用下级对象或被下级对象使用的对象的名称
数量	使用数量
使用位置	每个使用位置，例如，程序段
属性 (Property)	被引用对象的特定属性，例如，多重背景声明中的变量名称
作为 (as)	显示对象的更多相关信息，例如，背景数据块用作模板还是用作多重背景
访问 (Access)	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 (Path)	对象在项目树中的路径

视安装的产品而定，交叉引用表可能显示额外的列或不同的列。

6.5.3 用于检查调用层级的调用结构

调用结构描述了用户程序中块的调用层级。

其提供了以下几个方面的概要信息：所用的块、对其它块的调用、各个块之间的关系、每个块的数据要求以及块的状态。可从调用结构打开程序编辑器并对块进行编辑。

显示调用结构时会显示用户程序中使用的块的列表。STEP 7

高亮显示调用结构的第一级，并显示未被程序中的其它任何块调用的所有块。

调用结构的第一级显示 OB 以及未被 OB 调用的所有 FC、FB 和

DB。如果某个代码块调用了其它块，则被调用块将以缩进方式显示在调用块的下方。

调用结构仅显示被代码块调用的那些块。

6.5 轻松监视和测试用户程序

可以选择在调用结构中仅显示导致冲突的块。 下列情况会导致冲突：

- 块执行的任何调用具有更旧或更新代码时间戳
- 块所调用块的接口已更改
- 块所使用变量的地址和/或数据类型已更改
- 块未被 OB 直接或间接调用
- 块调用了不存在的块或缺失的块

可以将多个块调用和数据块分为一组。

可使用下拉列表来查看指向各个调用位置的链接。

还可执行一致性检查以显示时间戳冲突。

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。

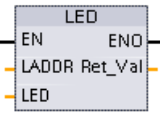
- 通过重新编译代码块可纠正大多数时间戳和接口冲突。
- 如果通过编译无法解决不一致问题，可使用“详细资料”(Details) 列中的链接转到程序编辑器中的问题源。 然后可手动消除任何不一致情况。
- 必须重新编译所有以红色标记的块。

6.5.4 用于监视硬件的诊断指令

6.5.4.1 读取 CPU 上 LED 的状态

LED 指令允许用户程序决定 CPU 上 LED 的状态。 可以使用该信息对 HMI 设备的变量进行编程。

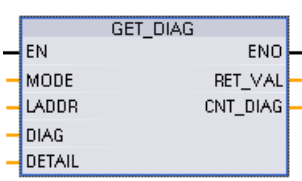
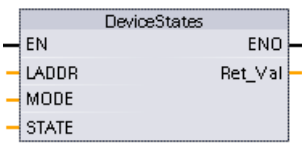
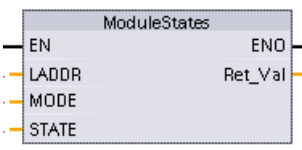
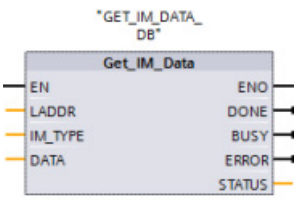
表格 6- 25 LED 指令

LAD/FBD	SCL	说明
	<pre>ret_val := LED(laddr:=_word_in_, LED:=_uint_in_);</pre>	<p>RET_VAL 为 CPU 返回以下 LED 状态</p> <ul style="list-style-type: none"> • RUN/STOP： 绿色或黄色 • 错误： 红色 • MAINT（维护）： 黄色 • 连接： 绿色 • Tx/Rx（发送/接收）： 黄色

6.5.4.2 用于读取设备诊断状态的指令

STEP 7 还包括一些用来读取网络中硬件设备所提供的状态信息的指令。

表格 6-26 诊断指令

LAD/FBD	SCL	说明
	<pre>ret_val := GET_DIAG(mode:=_uint_in_, laddr:=_word_in_, cnt_diag=>_uint_out_, diag:=_variant_inout_, detail:=_variant_inout_);</pre>	<p>GET_DIAG 指令从指定的硬件设备读取诊断信息。</p>
	<pre>ret_val := DeviceStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	<p>DeviceStates 指令读取 PROFINET 或 PROFIBUS 设备的状态。</p>
	<pre>ret_val := ModuleStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	<p>ModuleStates 指令读取 PROFINET 或 PROFIBUS 模块的状态。</p>
	<pre>"GET_IM_DATA_DB" (LADDR:=16#0, IM_TYPE:=0, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA:=_variant_inout_);</pre>	<p>使用“Get_IM_Data”指令在标识和维护 (I&M) 数据中检查指定的模块或子模块。</p>

6.6 高速计数器 (HSC)


使用高速计数器 (HSC, High-Speed Counter) 对发生速率快于 OB 执行速率的事件进行计数。计数指令在指令树的“工艺”部分中。CTRL_HSC 指令控制 HSC 的运行。

说明

如果待计数事件的发生速率处于 OB 执行速率范围内，请使用 CTU、CTD 或 CTUD 计数器指令。如果事件的发生速率快于 OB 的执行速率，则应使用 HSC。

在 CPU 的设备组态中对每个 HSC 的参数进行组态：计数模式、I/O 连接、中断分配以及是作为高速计数器还是设备来测量脉冲频率或周期。

表格 6-27 CTRL_HSC 指令

LAD/FBD	SCL	说明
	<pre>"counter_name" (HSC:=W#16#0, DIR:=FALSE, CV:=FALSE, RV:=FALSE, Period:=FALSE, New_DIR:=0, New_CV:=L#0, New_RV:=L#0, New_Period:=0, Busy=>_bool_out_, Status=> word out);</pre>	<p>每个 CTRL_HSC 指令使用数据块中存储的结构来保存计数器数据。</p> <p>对于 SCL，必须首先为各个计数器指令创建 DB 方可引用相应指令。对于 LAD 和 FBD，STEP 7 会在插入指令时自动创建 DB。</p>

CTRL_HSC 指令通常放置在触发计数器硬件中断事件时执行的硬件中断 OB 中。例如，如果 CV=RV 事件触发计数器中断，则硬件中断 OB 代码块执行 CTRL_HSC 指令，并且可通过装载 NEW_RV 值更改参考值。

说明

在 CTRL_HSC 参数中没有提供当前计数值。在高速计数器硬件的组态期间分配存储当前计数值的过程映像地址。可以使用程序逻辑直接读取计数值。返回程序的值将是读取计数器瞬间的正确计数。计数器仍将继续对高速事件计数。因此，程序使用旧的计数值完成处理前，实际计数值可能会更改。

可以通过用户程序来修改某些 HSC 参数，从而对计数过程提供程序控制：

- 将计数方向设置为 NEW_DIR 值
- 将当前计数值设置为 NEW_CV 值

- 将参考值设置为 NEW_RV 值
- 将周期值（限频率测量模式）设置为 NEW_PERIOD 值

如果执行 CTRL_HSC 指令后以下布尔标记值被设置为 1，则相应的 NEW_xxx 值将装载到计数器。执行一次 CTRL_HSC 指令可处理多个请求（同时设置多个标记）。将以下布尔标记值设置为 0 不会引起任何变化。

- 设置 DIR = 1 会装载 NEW_DIR 值。
- 设置 CV = 1 会装载 NEW_CV 值。
- 设置 RV = 1 会装载 NEW_RV 值。
- 设置 PERIOD = 1 会装载 NEW_PERIOD 值。

CTRL_HSC_EXT 指令（控制高速计数器（扩展）指令

STEP 7 和 S7-1200 CPU 还支持扩展的高速计数器指令 CTRL_HSC_EXT。

该指令允许程序精确测量指定 HSC 的输入脉冲周期。有关详细信息，请参见“S7-1200 可编程控制器系统手册”。

6.6.1 高速计数器的使用方法

高速计数器 (HSC) 能够对发生速率快于循环 OB 执行速率的事件进行计数。

如果待计数事件的发生速率慢于 OB 执行速率，则可使用 CTU、CTD 或 CTUD 标准计数器指令。如果事件的发生速率快于 OB 的执行速率，则应使用更快的 HSC 设备。CTRL_HSC 指令允许程序通过程序更改一些 HSC 参数。

例如：可以将 HSC 用作增量轴编码器的输入。
该轴编码器每转提供指定数量的计数值以及一个复位脉冲。
来自轴编码器的时钟和复位脉冲将输入到 HSC 中。

先将若干预设值中的第一个装载到 HSC 上，并且在当前计数值小于当前预设值的时段内计数器输出一直是激活的。
在当前计数值等于预设时、发生复位时以及方向改变时，HSC 会提供一个中断。

每次出现“当前计数值等于预设值”中断事件时，将装载一个新的预设值，同时设置输出的下一状态。

当出现复位中断事件时，将设置输出的第一个预设值和第一个输出状态，并重复该循环。

由于中断发生的频率远低于 HSC 的计数速率，因此能够在对 CPU 扫描周期影响相对较小的情况下实现对高速操作的精确控制。

通过提供中断，可以在独立的中断例程中执行每次的新预设值装载操作以实现简单的状态控制。此外，也可在单个中断例程中处理所有中断事件。

6.6 高速计数器 (HSC)

HSC 输入通道选择

使用以下表格并确保连接的 CPU 和 SB 输入通道可以支持过程信号中的最大脉冲速率。

说明

CPU 和 SB 输入通道（V4 或更高版本的固件）具有可组态的输入滤波时间

早期固件版本具有无法更改的固定 HSC 输入通道和固定滤波时间。

V4 或更高版本可以分配输入通道和滤波时间。

对于过程信号来说，默认的输入滤波器设置 6.4 ms 可能过慢。必须针对 HSC 应用优化 HSC 输入的数字量输入滤波时间。

表格 6-28 CPU 输入：最大频率

CPU	CPU 输入通道	1 或 2 相位模式	A/B 相正交相位模式
1211C	Ia.0 到 Ia.5	100 kHz	80 kHz
1212C	Ia.0 到 Ia.5	100 kHz	80 kHz
	Ia.6, Ia.7	30 kHz	20 kHz
1214C 和 1215C	Ia.0 到 Ia.5	100kHz	80kHz
	Ia.6 到 Ib.5	30 kHz	20 kHz
1217C	Ia.0 到 Ia.5	100 kHz	80 kHz
	Ia.6 到 Ib.1	30 kHz	20 kHz
	Ib.2 到 Ib.5 (.2+, .2- 到 .5+, .5-)	1 MHz	1 MHz

表格 6-29 SB 信号板输入：最大频率（可选信号板）

SB 信号板	SB 输入通道	1 或 2 相位模式	A/B 相正交相位模式
SB 1221, 200 kHz	Ie.0 到 Ie.3	200kHz	160 kHz
SB 1223, 200 kHz	Ie.0, Ie.1	200kHz	160 kHz
SB 1223	Ie.0, Ie.1	30 kHz	20 kHz

选择 HSC 的功能

所有 HSC 在同种计数器运行模式下的工作方式都相同。在 CPU 设备组态中为 HSC 功能属性分配计数器模式、方向控制和初始方向。

HSC 共有四种基本类型：

- 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 相正交计数器

6.6 高速计数器 (HSC)

用户可选择是否激活复位输入来使用各种 HSC 类型。

如果激活复位输入（存在一些限制，请参见下表），则它会清除当前值并在您禁用复位输入之前保持清除状态。

- 频率功能：有些 HSC 模式允许 HSC 被组态（计数类型）为报告频率而非当前脉冲计数值。有三种可用的频率测量周期：0.01、0.1 或 1.0 秒。

频率测量周期决定 HSC 计算并报告新频率值的频率。

报告频率是通过上一测量周期内总计数值确定的平均值。

如果该频率在快速变化，则报告值将是介于测量周期内出现的最高频率和最低频率之间的一个中间值。

无论频率测量周期的设置是什么，总是会以赫兹为单位来报告频率（每秒脉冲个数）。

- 计数器模式和输入：下表列出了用于与 HSC 相关的时钟、方向控制和复位功能的输入。
- 周期测量功能：周期测量通过组态的测量间隔（10ms、100ms 或 1000ms）提供。HSC_Period SDT 返回周期测量并以两个值的形式提供周期测量：ElapsedTime 和 EdgeCount。HSC 输入 ID1000 到 ID1020 不受周期测量的影响：

- ElapsedTime

是一个以纳秒为单位的无符号双精度整数值，表示测量间隔内从第一个计数事件到最后一个计数事件的时间。如果 EdgeCount = 0，则 ElapsedTime

是从上一个周期内的最后一个计数事件以来经过的时间。ElapsedTime 的范围是从 0 到 4,294,967,280 ns (0x0000 0000 到 0xFFFF FFF0)。如果值为 4,294,967,295 (0xFFFF FFFF)，则会发生溢出。从 0xFFFF FFF1 到 0xFFFF FFFE 的值是保留的。

- EdgeCount 是一个无符号的双精度整数值，表示测量间隔内计数事件的数量。

同一输入不可用于两个不同的功能，但任何未被其 HSC

的当前模式使用的输入均可用于其它用途。例如，如果 HSC1

处于使用两个内置输入但不使用第三个外部复位输入（默认分配为 I0.3）的模式，则 I0.3 可用于沿中断或 HSC 2。

表格 6-30 HSC 的计数模式

类型	输入 1	输入 2	输入 3	功能
具有内部方向控制的单相计数器	时钟	-	-	计数或频率
			复位	计数
具有外部方向控制的单相计数器	时钟	方向	-	计数或频率
			复位	计数
具有 2 个时钟输入的双相计数器	加时钟	减时钟	-	计数或频率
			复位	计数
A/B 相正交计数器	A 相	B 相	-	计数或频率
			复位 ¹	计数

¹ 对于编码器：Z 相，归位

HSC 的输入地址

组态 CPU 时，可以选择为每个 HSC 启用和组态“硬件输入”。

所有 HSC 输入必须连接到 CPU 模块上的端子，或插入 CPU 模块前方的可选信号板。

说明

如下表所示，不同 HSC 的可选信号的默认分配互相重叠。例如，HSC 1 的可选外部复位使用的输入与 HSC 2 的其中一个输入相同。

对于 V4 或更高版本的 CPU，可以在 CPU 组态期间重新分配 HSC 输入。不必使用默认输入分配。

请始终确保组态 HSC 时任何一个输入都不会被两个 HSC 使用。

下表显示了 CPU 的板载 I/O 和可选 SB 两者的默认 HSC 输入分配。（如果所选 SB 模块只有 2 个输入，则仅输入 4.0 和 4.1 可用。）

HSC 输入表定义

- **单相：** C 为时钟输入，[d] 为方向输入（可选），[R] 为外部复位输入（可选）（复位仅适用于“计数”模式。）
- **双相：** CU 为加时钟输入，CD 为减时钟输入，[R] 为外部复位输入（可选）。（复位仅适用于“计数”模式。）
- **AB 相正交：** A 为时钟 A 输入，B 为时钟 B 输入，[R] 为外部复位输入（可选）。（复位仅适用于“计数”模式。）

6.6 高速计数器 (HSC)

表格 6-31 CPU 1211C: HSC 默认地址分配

HSC 计数器模式		CPU 板载输入 (默认为 0.x)						可选 SB 输入 (默认为 4.x) ¹			
		0	1	2	3	4	5	0	1	2	3
HSC 1	单相	C	[d]		[R]			C	[d]		[R]
	双相	CU	CD		[R]			CU	CD		[R]
	AB 相	A	B		[R]			A	B		[R]
HSC 2	单相		[R]	C	[d]				[R]	C	[d]
	双相		[R]	CU	CD				[R]	CU	CD
	AB 相		[R]	A	B				[R]	A	B
HSC 3	单相					C	[d]	C	[d]		[R]
	双相										
	AB 相										
HSC 4	单相					C	[d]	C	[d]		[R]
	双相					CU	CD				
	AB 相					A	B				
HSC 5	单相							C	[d]		[R]
	双相							CU	CD		[R]
	AB 相							A	B		[R]
HSC 6	单相								[R]	C	[d]
	双相								[R]	CU	CD
	AB 相								[R]	A	B

¹ 仅具有 2 个数字量输入的 SB 只能提供输入 4.0 和 4.1。

表格 6-32 CPU 1212C: HSC 默认地址分配

HSC 计数器模式		CPU 板载输入 (默认为 0.x)								可选 SB 输入 (默认为 4.x) ¹			
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 1	单相	C	[d]		[R]					C	[d]		[R]
	双相	CU	CD		[R]					CU	CD		[R]
	AB 相	A	B		[R]					A	B		[R]
HSC 2	单相		[R]	C	[d]						[R]	C	[d]
	双相		[R]	CU	CD						[R]	CU	CD
	AB 相		[R]	A	B						[R]	A	B
HSC 3	单相					C	[d]		[R]	C	[d]		[R]
	双相					CU	CD		[R]				
	AB 相					A	B		[R]				
HSC 4	单相						[R]	C	[d]	C	[d]		[R]
	双相						[R]	CU	CD				
	AB 相						[R]	A	B				
HSC 5	单相									C	[d]		[R]
	双相									CU	CD		[R]
	AB 相									A	B		[R]
HSC 6	单相										[R]	C	[d]
	双相										[R]	CU	CD
	AB 相										[R]	A	B

¹ 仅具有 2 个数字量输入的 SB 只能提供输入 4.0 和 4.1。

6.6 高速计数器 (HSC)

表格 6-33 CPU 1214C、CPU 1215C 和 CPU1217C:
HSC 默认地址分配
(仅板载输入, 可选 SB 地址参见下表)

HSC 计数器模式		数字量输入字节 0 (默认值: 0.x)							数字量输入字节 1 (默认值: 1.x)						
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 1	单相	C	[d]		[R]										
	双相	C U	C D		[R]										
	AB 相	A	B		[R]										
HSC 2	单相		[R]	C	[d]										
	双相		[R]	C U	C D										
	AB 相		[R]	A	B										
HSC 3	单相					C	[d]		[R]						
	双相					C U	C D		[R]						
	AB 相					A	B		[R]						
HSC 4	单相						[R]	C	[d]						
	双相						[R]	C U	C D						
	AB 相						[R]	A	B						
HSC 5	单相									C	[d]	[R]			
	双相									C U	C D	[R]			
	AB 相									A	B	[R]			
HSC 6	单相												C	[d]	[R]
	双相												C U	C D	[R]
	AB 相												A	B	[R]

表格 6-34 上表 CPU 中的可选 SB: HSC 默认地址分配

HSC		可选 SB 输入 (默认值: 4.x) ¹			
		0	1	2	3
HSC 1	单相	C	[d]		[R]
	双相	CU	CD		[R]
	AB 相	A	B		[R]
HSC 2	单相		[R]	C	[d]
	双相		[R]	CU	CD
	AB 相		[R]	A	B
HSC 5	单相	C	[d]		[R]
	双相	CU	CD		[R]
	AB 相	A	B		[R]
HSC 6	单相		[R]	C	[d]
	双相		[R]	CU	CD
	AB 相		[R]	A	B

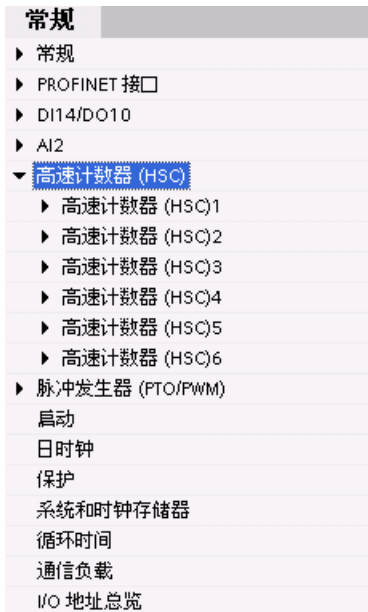
¹ 仅具有 2 个数字量输入的 SB 只能提供输入 4.0 和 4.1。

说明

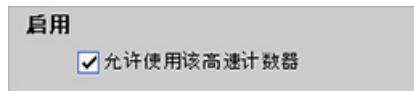
在 CPU 设备组态期间分配高速计数器设备使用的数字量 I/O 点。将数字量 I/O 点的地址分配给 HSC 设备之后, 无法通过监控表中的强制功能修改所分配的 I/O 点的地址值。


6.6 高速计数器 (HSC)

6.6.2 组态 HSC



最多可组态 6 个高速计数器。编辑 CPU 设备组态并为各个 HSC 分配 HSC 属性。通过选择该 HSC 的“启用”(Enable) 选项启用 HSC 在用户程序中使用 CTRL_HSC 和/或 CTRL_HSC_EXT 指令控制 HSC 的运行。



 警告
<p>为数字量输入通道更改滤波时间设置的风险</p> <p>如果数字量输入通道的滤波时间更改自以前的设置，则新的“0”电平输入值可能需要保持长达 20.0 ms 的累积时间，然后滤波器才会完全响应新输入。在此期间，可能不会检测到持续时间少于 20.0 ms 的短“0”脉冲事件或对其计数。滤波时间的这种更改会引发意外的机械或过程操作，这可能会导致人员死亡、重伤和/或设备损坏。</p> <p>对 CPU 循环上电以确保新的滤波时间立即生效。</p>

启用 HSC 之后组态其它参数，例如计数器功能、初始值、复位选项和中断事件。

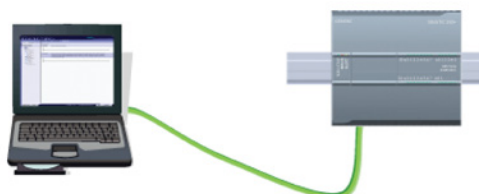


The image shows a configuration window for a High-Speed Counter (HSC). It contains several dropdown menus for setting parameters:

- 计数类型: 计数
- 运行阶段: 单相
- 输入源: 板载 CPU 输入
- 计数方向取决于: 用户程序 (内部方向控制)
- 初始计数方向: 加计数
- 频率测量期间: +/- sec

有关组态 HSC 的其他信息，请参见组态 CPU (页 86) 部分。

轻松实现设备间通信



对于 CPU 和编程设备之间的直接连接:

- 项目必须包含 CPU。
- 编程设备不是项目的一部分，但是必须运行 STEP 7。



对于 HMI 面板和 CPU 之间的直接连接，项目必须同时包括 CPU 和 HMI。



对于 CPU 和 CPU 之间的直接连接:

- 项目必须同时包含这两个 CPU。
- 必须组态这两个 CPU 之间的网络连接。

S7-1200 CPU 是 PROFINET IO 控制器，用来与编程设备上的 STEP 7 以及 HMI 设备、其它 CPU 或非西门子设备进行通信。编程设备或 HMI 与 CPU 之间的直接连接不需要以太网交换机。含有两个以上的 CPU 或 HMI 设备的网络需要以太网交换机。

通过添加 PROFIBUS CM，CPU 也可以起到 PROFIBUS 网络中的主站或从站的作用。

其它通信接口（CM、CP 或 CB）支持多种协议，如点对点（PTP）协议、Modbus 协议、USS 协议和 GPRS（调制解调器）协议、安全 CP 协议和远程控制 CP 协议。

7.1 创建网络连接

使用设备配置的“网络视图”(Network view) 在项目中的各个设备之间创建网络连接。
创建网络连接之后，使用巡视窗口的“属性”(Properties) 选项卡组态网络的参数。

表格 7-1 创建网络连接

操作	结果
选择“网络视图”(Network view) 以显示要连接的设备。	 <p>The screenshot shows the 'Network View' window with two PLC devices, PLC_1 and PLC_2, each labeled 'CPU 1214C'. The window title is '项目1 > 设备和网络'. The '网络视图' (Network View) tab is active. The network is currently empty.</p>
选择一个设备上的端口，然后将连接拖到第二个设备上的端口处。	 <p>The screenshot shows the same 'Network View' window. A blue line is being dragged from a port on PLC_1 to a port on PLC_2. The mouse cursor is at the end of the line on the second device.</p>
释放鼠标按钮以创建网络连接。	 <p>The screenshot shows the 'Network View' window with the connection between PLC_1 and PLC_2 completed. The connection is labeled 'PNIE_1' and is shown as a blue line connecting the two devices.</p>

7.2 通讯选项

S7-1200 可实现 CPU 与编程设备、HMI 和其它 CPU 之间的多种通信。



警告

如果攻击者能以物理方式访问您的网络，那么便可能读写数据。

TIA Portal、CPU 和 HMI（使用 GET/PUT 的 HMI

除外）均采用安全通信，可防止重放攻击和“中间人”攻击。

启用这种通信后，将以纯文本形式交换签名消息，这种方式允许攻击者读取数据，但可避免未经授权的数据写入操作。TIA

Portal（而非通信过程）将对受专有技术保护的块中的数据进行加密。

所有其它形式的通信（通过 PROFIBUS、PROFINET、AS-i 或其它 I/O

总线、GET/PUT、传输块 (T-block) 和通信模块 (CM) 进行的 I/O

交换）均没有安全功能。必须通过限制物理访问来保护这些形式的通信。

如果攻击者能利用这些形式的通信以物理方式访问您的网络，那么便可能读写数据。

有关安全信息和建议，请参见 Siemens 服务与支持网站上的“工业安全操作准则

(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf)”。

PROFINET

PROFINET 用于使用用户程序通过以太网与其它通信伙伴交换数据：

- 在 S7-1200 中，PROFINET 支持 16 个最多具有 256 个子模块的 IO 设备，PROFIBUS 允许使用 3 个独立的 PROFIBUS DP 主站，每个 DP 主站支持 32 个从站，每个 DP 主站最多具有 512 个模块。
- S7 通信
- 用户数据报协议 (UDP)
- ISO on TCP (RFC 1006)
- 传输控制协议 (TCP)

PROFINET IO 控制器

作为采用 PROFINET IO 的 IO 控制器，CPU 可与本地 PN 网络上或通过 PN/PN 耦合器（连接器）连接的最多 16 台 PN 设备通信。有关详细信息，请参见 PROFIBUS 和 PROFINET International (PI) (www.profinet.com)。

PROFIBUS

PROFIBUS 用于使用用户程序通过 PROFIBUS 网络与其它通信伙伴交换数据:

- 借助 CM 1242-5, CPU 作为 PROFIBUS DP 从站运行。
- 借助 CM 1243-5, CPU 作为 1 类 PROFIBUS DP 主站运行。
- PROFIBUS DP 从站、PROFIBUS DP 主站和 AS-i (左侧 3 个通信模块) 以及 PROFINET 均采用单独的通信网络, 不会相互制约。

AS-i

通过 S7-1200 CM 1243-2 AS-i 主站可将 AS-i 网络连接到 S7-1200 CPU。

CPU 至 CPU S7 通信

您可以创建与伙伴站的通信连接并使用 GET 和 PUT 指令与 S7 CPU 进行通信。

TeleService 通信

在通过 GPRS 的 TeleService 中, 安装了 STEP 7 的工程师站通过 GSM 网络 Internet 和与具有 CP 1242-7 的 SIMATIC S7-1200 站进行通信。该连接通过用作中介并连接到 Internet 的远程控制服务器运行。

IO-Link

利用 S7-1200 SM 1278 4xIO-Link 主站, 可将 IO-Link 设备与 S7-1200 CPU 相连。

7.3 V4.1 异步通信连接

通信服务概述

此 CPU 支持以下通信服务：

通信服务	功能	使用 PROFIBUS DP		使用以太网
		CM 1243-5 DP 主站模块	CM 1242-5 DP 从站模块	
PG 通信	调试、测试、诊断	√	×	√
HMI 通信	操作员控制和监视	√	×	√
S7 通信	使用已组态连接交换数据	√	×	√
路由 PG 功能	例如，跨网络边界进行测试和诊断	×	×	×
PROFIBUS DP	在主站与从站之间交换数据	√	√	×
PROFINET IO	I/O 控制器和 I/O 设备之间的数据交换	×	×	√
Web 服务器	诊断	×	×	√
SNMP (简单网络管理协议)	用于网络诊断和参数化的标准协议	×	×	√
通过 TCP/IP 的开放式通信	使用 TCP/IP 协议通过工业以太网交换数据 (使用可装载 FB)	×	×	√
通过 ISO on TCP 的开放式通信	使用 ISO on TCP 协议通过工业以太网交换数据 (使用可加载 FB)	×	×	√
通过 UDP 的开放式通信	使用 UDP 协议通过工业以太网交换数据 (使用可装载 FB)	×	×	√

可用连接

对于 PROFINET 和 PROFIBUS，CPU 最多可支持下列数量的并发异步通信连接。分配给每个类别的最大连接资源数为固定值；您无法更改这些值。但可组态 6 个“可用自由连接”以按照应用要求增加任意类别的连接数。



根据已分配的连接资源，每个设备的可用连接数如下：

	编程终端 (PG)	人机界面 (HMI)	GET/PUT 客户端/服务器	开放式用户通信	Web 浏览器
连接资源的最大数量	3 (保证支持 1 个 PG 设备)	12 (保证支持 4 个 HMI 设备)	8	8	30 (保证支持 3 个 Web 浏览器)

例如，1 个 PG 具有 3 个可用连接资源。根据当前使用的 PG 功能，该 PG 实际可能使用其可用连接资源的 1、2 或 3。在 S7-1200 中，始终保证至少有 1 个 PG，但不允许超过 1 个 PG。

另一个示例为 HMI 数，如下图所示。HMI 具有 12 个可用连接资源。根据您拥有的 HMI 类型或型号以及使用的 HMI 功能，每个 HMI 实际可能使用其可用连接资源中的 1 个、2 个或 3 个。考虑到正在使用的可用连接资源数，可以同时使用 4 个以上的 HMI。但是，要始终确保至少有 4 个 HMI。HMI 可利用其可用连接资源（每个 1 个，共 3 个）实现下列功能：

- 读取
- 写入
- 报警和诊断

示例	HMI 1	HMI 2	HMI 3	HMI 4	HMI 5	总的可用连接资源
使用的连接资源	2	2	2	3	3	12

说明

Web 服务器 (HTTP) 连接：CPU 提供用于多个 Web 浏览器的连接。此 CPU 可同时支持的浏览器数取决于给定 Web 浏览器请求/使用的连接数。

说明

开放式用户通信、S7 连接、HMI、编程设备以及 Web 服务器 (HTTP) 通信连接可以根据当前使用的功能使用多个连接资源。

7.4 PROFINET 和 PROFIBUS 指令

PROFINET 指令

TSEND_C 和 TRCV_C 指令将 TCON 和 TDISCON 指令的功能与 TSEND 或 TRCV 指令相结合，从而使 PROFINET 通信更加简单。

- TSEND_C 可与伙伴站建立 TCP 或 ISO on TCP 通信连接、发送数据，并且可以终止该连接。设置并建立连接后，CPU 会自动保持和监视该连接。TSEND_C 指令兼具 TCON、TDISCON 和 TSEND 指令的功能。
- TRCV_C 可与伙伴 CPU 建立 TCP 或 ISO-on-TCP 通信连接，可接收数据，并且可以终止该连接。设置并建立连接后，CPU 会自动保持和监视该连接。TRCV_C 指令兼具 TCON、TDISCON 和 TRCV 指令的功能。

此外，也支持 TCON、TDISCON、TSEND 和 TRCV 指令。

使用 TUSEND 和 TURCV 指令可通过 UDP 发送或接收数据。TUSEND 和 TURCV（以及 TSEND、TRCV、TCON、TDISCON）异步运行，即，作业的处理需要多次调用指令来完成。

使用 IP_CONF 指令可在用户程序中更改 IP 组态参数。IP_CONF 异步运行。
执行作业时需要多次调用指令。

PROFIBUS 指令

DPNRM_DG（读取诊断）指令以“EN 50 170 第 2 卷，PROFIBUS”所指定的格式来读取 DP 从站的当前诊断数据。

用于 PROFINET、PROFIBUS 和 AS-i 的分布式 I/O 指令

可以对 PROFINET、PROFIBUS 和 GPRS 使用以下指令。

- 使用 RDREC（读取记录）和 WRREC（写入记录）指令可在组件（例如，中央机架上的模块，或者分布式组件（PROFIBUS DP 或 PROFINET IO））中读写指定的数据记录。
- 使用 RALRM（读取报警）指令可从 DP 从站或 PROFINET IO 设备组件读取中断及其信息。输出参数中的信息包含被调用 OB 的启动信息以及中断源的信息。
- 使用 DPRD_DAT（读取一致数据）和 DPWR_DAT（写入一致数据）指令可以从 DP 标准从站或向 DP 标准从站传送 64 字节以上的一致数据区域。
- 仅针对 PROFIBUS，使用 DPNRM_DG 指令将以“EN 50 170 第 2 卷，PROFIBUS”所指定的格式来读取 DP 从站的当前诊断数据。

7.5 PROFINET

7.5.1 开放式用户通信

CPU 的集成 PROFINET 端口支持多种以太网网络上的通信标准：

- 传输控制协议 (TCP)
- ISO on TCP (RFC 1006)
- 用户数据报协议 (UDP)

表格 7-2 协议以及用于每种协议的通信指令

协议	用途示例	在接收区输入数据	通信指令	寻址类型
TCP	CPU 与 CPU 通信 帧传输	特殊模式	仅 TRCV_C 和 TRCV (V4.1 及早期指令)	将端口号分配给本地 (主动) 和伙伴 (被 动) 设备
		指定长度的数据接收	TSEND_C、TRCV_ C、TCON、TDISC ON、TSEND 和 TRCV (V4.1 及早期指令)	
ISO on TCP	CPU 与 CPU 通信 消息的分割和重组	特殊模式	仅 TRCV_C 和 TRCV (V4.1 及早期指令)	将 TSAP 分配给本地 (主动) 和伙伴 (被动) 设备
		协议控制	TSEND_C、TRCV_ C、TCON、TDISC ON、TSEND 和 TRCV (V4.1 及早期指令)	
UDP	CPU 与 CPU 通信 用户程序通信	用户数据报协议	TUSEND 和 TURCV	将端口号分配给本地 (主动) 和伙伴 (被 动) 设备, 但不是专 用连接

7.5 PROFINET

协议	用途示例	在接收区输入数据	通信指令	寻址类型
S7 通信	CPU 与 CPU 通信 从 CPU 读取数据/向 CPU 写入数据	指定长度的数据传输 和接收	GET 和 PUT	将 TSAP 分配给本地（主动） 和伙伴（被动）设备
PROFINET IO	CPU 与 PROFINET IO 设备通信	指定长度的数据传输 和接收	内置	内置

7.5.1.1 特殊模式

通常，TCP 和 ISO-on-TCP 接收指定长度的数据包（1 到 8192 字节）。但 TRCV_C 和 TRCV 通信指令还提供“特殊”通信模式，可接收可变长度的数据包（1 到 1472 字节）。

说明

如果将数据存储在“优化”DB（仅符号访问）中，则只能接收数据类型为 Byte、Char、USInt 和 SInt 的数组中的数据。

要针对特殊模式组态 TRCV_C 或 TRCV 指令，请置位 ADHOC 指令输入参数。

如果在特殊模式下并未频繁调用 TRCV_C 或 TRCV 指令，则可在一次调用中接收多个数据包。例如：如果要通过一次调用接收五个 100 字节的数据包，TCP 可将这五个数据包打包成一个 500 字节的数据包一起传送，而 ISO-on-TCP 则可将该数据包重组为五个 100 字节的数据包。

7.5.1.2 开放式用户通信指令的连接 ID

将 TSEND_C、TRCV_C 或 TCON PROFINET 指令插入到用户程序中时，STEP 7 会创建一个背景数据块，以组态设备之间的通信通道（或连接）。

使用指令的“属性”(Properties) (页 163) 组态连接的参数。这些参数中有该连接的连接 ID。

- 连接 ID 对于 CPU 必须是唯一的。创建的每个连接必须具有不同的 DB 和连接 ID。
- 本地 CPU 和伙伴 CPU 都可以对同一连接使用相同的连接 ID 编号，但连接 ID 编号不需要匹配。连接 ID 编号只与各 CPU 用户程序中的 PROFINET 指令相关。
- CPU 的连接 ID 可以使用任何数字。但是，从“1”开始按顺序组态连接 ID 可以很容易地跟踪特定 CPU 使用的连接数。

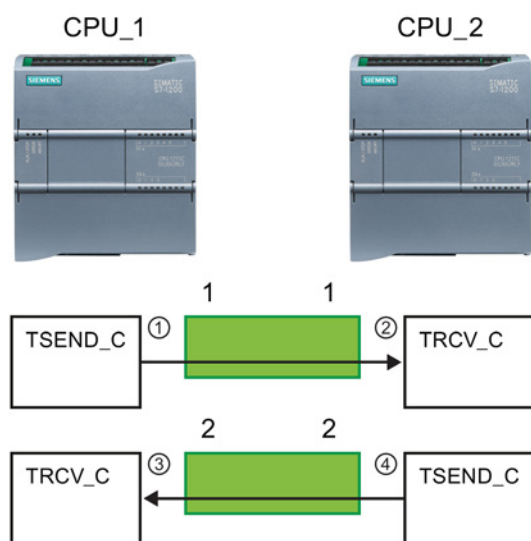
说明

用户程序中的每个 TSEND_C、TRCV_C 或 TCON 指令都创建一个新连接。为每个连接使用正确的连接 ID 非常重要。

7.5 PROFINET

以下示例显示了两个 CPU 之间的通信，这两个 CPU 使用 2 个单独的连接来发送和接收数据。

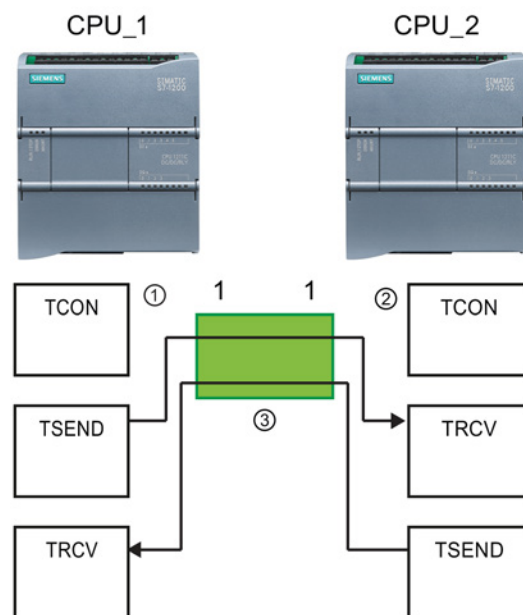
- CPU_1 中的 TSEND_C 指令通过第一个连接（CPU_1 和 CPU_2 上的“连接 ID 1”）与 CPU_2 中的 TRCV_C 链接。
- CPU_1 中的 TRCV_C 指令通过第二个连接（CPU_1 和 CPU_2 上的“连接 ID 2”）与 CPU_2 中的 TSEND_C 链接。



- ① CPU_1 上的 TSEND_C 创建一个连接并为该连接分配一个连接 ID（CPU_1 的连接 ID 1）。
- ② CPU_2 上的 TRCV_C 为 CPU_2 创建连接并分配连接 ID（CPU_2 的连接 ID 1）。
- ③ CPU_1 上的 TRCV_C 为 CPU_1 创建第二个连接并为该连接分配不同的连接 ID（CPU_1 的连接 ID 2）。
- ④ CPU_2 上的 TSEND_C 创建第二个连接并为该连接分配不同的连接 ID（CPU_2 的连接 ID 2）。

以下示例显示了两个 CPU 之间的通信，这两个 CPU 使用 1 个连接来发送和接收数据。

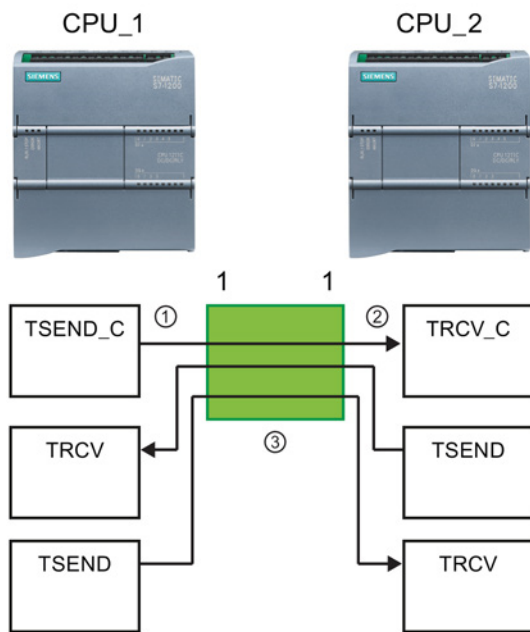
- 每个 CPU 都使用 TCON 指令来组态两个 CPU 之间的连接。
- CPU_1 中的 TSEND 指令通过由 CPU_1 中的 TCON 指令组态的连接 ID (“连接 ID 1”) 链接到 CPU_2 中的 TRCV 指令。CPU_2 中的 TRCV 指令通过由 CPU_2 中的 TCON 指令组态的连接 ID (“连接 ID 1”) 链接到 CPU_1 中的 TSEND 指令。
- CPU_2 中的 TSEND 指令通过由 CPU_2 中的 TCON 指令组态的连接 ID (“连接 ID 1”) 链接到 CPU_1 中的 TRCV 指令。CPU_1 中的 TRCV 指令通过由 CPU_1 中的 TCON 指令组态的连接 ID (“连接 ID 1”) 链接到 CPU_2 中的 TSEND 指令。



- ① CPU_1 上的 TCON
创建一个连接并在 CPU_1 上为该连接分配连接 ID (ID=1)。
- ② CPU_2 上的 TCON
创建一个连接并在 CPU_2 上为该连接分配连接 ID (ID=1)。
- ③ CPU_1 上的 TSEND 和 TRCV 使用 CPU_1 上的 TCON 创建的连接 ID (ID=1)。
CPU_2 上的 TSEND 和 TRCV 使用 CPU_2 上的 TCON 创建的连接 ID (ID=1)。

7.5 PROFINET

如以下示例所示，还可以使用单个 TSEND 和 TRCV 指令通过由 TSEND_C 或 TRCV_C 指令创建的连接进行通信。TSEND 和 TRCV 指令本身不会创建新连接，因此必须使用由 TSEND_C、TRCV_C 或 TCON 指令创建的 DB 和连接 ID。



- ① CPU_1 上的 TSEND_C 创建一个连接并为该连接分配连接 ID (ID=1)。
- ② CPU_2 上的 TRCV_C 创建一个连接并在 CPU_2 上为该连接分配连接 ID (ID=1)。
- ③ CPU_1 上的 TSEND 和 TRCV 使用 CPU_1 上的 TSEND_C 创建的连接 ID (ID=1)。CPU_2 上的 TSEND 和 TRCV 使用 CPU_2 上的 TRCV_C 创建的连接 ID (ID=1)。

7.5.1.3 PROFINET 连接的参数

TSEND_C、TRCV_C 和 TCON 指令要求指定与连接相关的参数，以连接到伙伴设备。这些参数将通过 TCP、ISO-on-TCP 和 UDP 协议的 TCON_Param 结构来分配。通常使用指令的“属性”(Properties) 中的“组态”(Configuration) 选项卡来指定这些参数。如果无法访问“组态”(Configuration) 选项卡，则必须通过程序指定 TCON_Param 结构。

TCON_Param

表格 7-3 连接描述的结构 (TCON_Param)

字节	参数和数据类型		说明
0 ... 1	block_length	UInt	长度： 64 个字节（固定）
2 ... 3	id	CONN_OUC (Word)	对该连接的引用： 值范围： 1（默认值）到 4095。在 ID 下为指令 TSEND_C、TRCV_C 或 TCON 指定该参数的值。
4	connection_type	USInt	连接类型： <ul style="list-style-type: none"> • 17: TCP（默认） • 18: ISO-on-TCP • 19: UDP
5	active_est	Bool	连接类型的 ID： <ul style="list-style-type: none"> • TCP 和 ISO-on-TCP： <ul style="list-style-type: none"> – FALSE： 被动连接 – TRUE： 主动连接（默认） • UDP： FALSE
6	local_device_id	USInt	本地 PROFINET 或工业以太网接口的 ID： 1（默认值）
7	local_tsap_id_len	USInt	所用 local_tsap_id 参数的长度（以字节表示）； 可能值： <ul style="list-style-type: none"> • TCP： 0（主动，默认值）或 2（被动） • ISO-on-TCP： 2 到 16 • UDP： 2
8	rem_subnet_id_len	USInt	该参数未使用。
9	rem_staddr_len	USInt	伙伴端点地址的长度（以字节表示）： <ul style="list-style-type: none"> • 0: 未指定（参数 rem_staddr 不相关） • 4（默认值）： 参数 rem_staddr 中的 IP 地址有效（仅对于 TCP 和 ISO-on-TCP）

7.5 PROFINET

字节	参数和数据类型	说明
10	rem_tsap_id_len USInt	所用 rem_tsap_id 参数的长度（以字节表示）；可能值： <ul style="list-style-type: none"> • TCP: 0（被动）或 2（主动，默认值） • ISO-on-TCP: 2 到 16 • UDP: 0
11	next_staddr_len USInt	该参数未使用。
12 ... 27	local_tsap_id Array [1..16] of Byte	连接的本地地址部分： <ul style="list-style-type: none"> • TCP 和 ISO-on-TCP: 本地端口号（可能值：1 到 49151；推荐值：2000...5000): <ul style="list-style-type: none"> - local_tsap_id[1] = 十六进制表示的端口号的高位字节； - local_tsap_id[2] = 十六进制表示的端口号的低位字节； - local_tsap_id[3-16] = 不相关 • ISO-on-TCP: 本地 TSAP-ID: <ul style="list-style-type: none"> - 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 • UDP: 该参数未使用。 <p>注：请确保 local_tsap_id 的每个值在 CPU 中都是唯一的。</p>
28 ... 33	rem_subnet_id Array [1..6] of USInt	该参数未使用。
34 ... 39	rem_staddr Array [1..6] of USInt	仅 TCP 和 ISO-on-TCP: 伙伴端点的 IP 地址。（与被动连接不相关。）例如，IP 地址 192.168.002.003 存储在数组的下列元素中： <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>

字节	参数和数据类型		说明
40 ... 55	rem_tsap_id	Array [1..16] of Byte	连接的伙伴地址部分 <ul style="list-style-type: none"> • TCP： 伙伴端口号。 范围： 1 到 49151； 推荐值： 2000 到 5000）： <ul style="list-style-type: none"> - rem_tsap_id[1] = 十六进制表示的端口号的高位字节 - rem_tsap_id[2] = 十六进制表示的端口号的低位字节； - rem_tsap_id[3-16] = 不相关 • ISO-on-TCP： 伙伴 TSAP-ID： <ul style="list-style-type: none"> - 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 • UDP： 该参数未使用。
56 ... 61	next_staddr	Array [1..6] of Byte	该参数未使用。
62 ... 63	spare	Word	保留： W#16#0000

7.5.2 组态本地/伙伴连接路径

本地/伙伴（远程）连接定义两个通信伙伴的逻辑分配以建立通信服务。连接定义了以下内容：

- 涉及的通信伙伴（一个主动，一个被动）
- 连接类型（例如，PLC、HMI 或设备连接）
- 连接路径

通信伙伴执行指令来设置和建立通信连接。


用户使用参数指定主动和被动通信端点伙伴。设置并建立连接后，CPU 会自动保持和监视该连接。

如果连接终止（例如，因断线），主动伙伴将尝试重新建立组态的连接。不必再次执行通信指令。

连接路径

将 TSEND_C、TRCV_C 或 TCON 指令插入用户程序后，只要选中指令的任意部分，巡视窗口都会显示连接的属性。在通信指令“属性”(Properties) 的“组态”(Configuration) 选项卡中指定通信参数。

表格 7-4 组态连接路径（使用指令的属性）


TCP、ISO-on-TCP 和 UDP	连接属性
<p>对于 TCP、ISO-on-TCP 和 UDP 以太网协议，使用指令（TSEND_C、TRCV_C 或 TCON）的“属性”(Properties) 组态“本地/伙伴”连接。</p> <p>右图显示了 ISO-on-TCP 连接的“组态”(Configuration) 选项卡中的“连接属性”(Connection properties)。</p>	

说明

组态其中一个 CPU 的连接属性时，STEP 7 允许您选择伙伴 CPU 中的特定连接 DB（如果存在），或为伙伴 CPU 创建连接 DB。必须已为该项目创建伙伴 CPU，且不能是“未指定的”CPU。

还必须将 TSEND_C、TRCV_C 或 TCON 指令插入伙伴 CPU 的用户程序中。插入指令时，应选择由组态创建的连接 DB。

表格 7-5 为 S7 通信组态连接路径（设备组态）

S7 通信 (GET 和 PUT)	连接属性
<p>对于 S7 通信，请使用网络的“设备和网络”编辑器组态本地/伙伴连接。可以单击“突出显示：连接”(Highlighted: Connection) 按钮访问“属性”(Properties)。</p> <p>“常规”(General) 选项卡中提供有多个属性：</p> <ul style="list-style-type: none"> “常规”(General) (已显示) “本地 ID”(Local ID) “特殊连接属性”(Special connection properties) “地址详细信息”(Address details) (已显示) 	 <p>The image shows two screenshots of the S7 connection configuration window. The top screenshot shows the 'General' tab with fields for 'Local ID', 'Connection path', 'Local endpoint', 'Partner endpoint', 'Interface', 'Interface type', 'Subnet', and 'Address'. The bottom screenshot shows the 'Address Details' tab with fields for 'Local endpoint', 'Partner endpoint', 'Rack slot', 'Connection resource', 'TSAP', 'SIMATIC ACC', and 'Subnet ID'.</p>

要获取更多信息以及可用通信指令的列表，请参见“PROFINET”部分的“协议” (页 157)，或“S7 通信”部分的“创建 S7 连接” (页 181)。

表格 7-6 多 CPU 连接的参数

参数	定义																
地址	分配的 IP 地址																
常规	<table border="1"> <tr> <td data-bbox="330 1453 624 1506">端点</td> <td data-bbox="624 1453 1481 1506">分配给伙伴（接收）CPU 的名称</td> </tr> <tr> <td data-bbox="330 1506 624 1559">接口</td> <td data-bbox="624 1506 1481 1559">分配给接口的名称</td> </tr> <tr> <td data-bbox="330 1559 624 1613">子网</td> <td data-bbox="624 1559 1481 1613">分配给子网的名称</td> </tr> <tr> <td data-bbox="330 1613 624 1666">接口类型</td> <td data-bbox="624 1613 1481 1666">仅 S7 通信：接口类型</td> </tr> <tr> <td data-bbox="330 1666 624 1719">连接类型</td> <td data-bbox="624 1666 1481 1719">以太网协议的类型</td> </tr> <tr> <td data-bbox="330 1719 624 1772">连接 ID</td> <td data-bbox="624 1719 1481 1772">ID 号</td> </tr> <tr> <td data-bbox="330 1772 624 1825">连接数据</td> <td data-bbox="624 1772 1481 1825">本地和伙伴 CPU 的数据存储位置</td> </tr> <tr> <td data-bbox="330 1825 624 1891">建立主动连接</td> <td data-bbox="624 1825 1481 1891">用于选择本地或伙伴 CPU 作为主动连接方的单选按钮</td> </tr> </table>	端点	分配给伙伴（接收）CPU 的名称	接口	分配给接口的名称	子网	分配给子网的名称	接口类型	仅 S7 通信：接口类型	连接类型	以太网协议的类型	连接 ID	ID 号	连接数据	本地和伙伴 CPU 的数据存储位置	建立主动连接	用于选择本地或伙伴 CPU 作为主动连接方的单选按钮
端点	分配给伙伴（接收）CPU 的名称																
接口	分配给接口的名称																
子网	分配给子网的名称																
接口类型	仅 S7 通信：接口类型																
连接类型	以太网协议的类型																
连接 ID	ID 号																
连接数据	本地和伙伴 CPU 的数据存储位置																
建立主动连接	用于选择本地或伙伴 CPU 作为主动连接方的单选按钮																

参数		定义
地址详细信息	端点	仅 S7 通信: 分配给伙伴 (接收) CPU 的名称
	机架/插槽	仅 S7 通信: 机架和插槽位置:
	连接资源	仅 S7 通信: 组态与 S7-300 或 S7-400 CPU 通信的 S7 连接时使用的 TSAP 组件
	端口 (十进制)	TCP 和 UDP: 十进制格式的伙伴 CPU 端口
	TSAP ¹ 和子网 ID:	ISO on TCP (RFC 1006) 和 S7 通信: ASCII 格式和十六进制格式的本地和伙伴 CPU TSAP

¹ 组态与 S7-1200 CPU 的 ISO-on-TCP 连接时, 请在被动通信伙伴的 TSAP 扩展中仅使用 ASCII 字符。

传输服务访问点 (TSAP)

通过 TSAP、ISO on TCP 协议和 S7 通信, 允许有多个连接访问单个 IP 地址 (最多 64K 个连接)。TSAP 可唯一标识连接到同一个 IP 地址的这些通信端点连接。

在“连接参数”(Connection Parameters) 对话框的“地址详细信息”(Address Details) 部分, 定义要使用的 TSAP。在“本地 TSAP”(Local TSAP) 域中输入 CPU 中连接的 TSAP。在“伙伴 TSAP”(Partner TSAP) 域下输入为伙伴 CPU 中的连接分配的 TSAP。

端口号

使用 TCP 和 UDP 协议时, 本地 (主动) 连接 CPU 的连接参数组态必须指定远程伙伴 (被动) 连接 CPU 的 IP 地址和端口号。

在“连接参数”(Connection Parameters) 对话框的“地址详细信息”(Address Details) 部分, 定义要使用的端口。在“本地端口”(Local Port) 域中输入 CPU 中连接的端口。在“伙伴端口”(Partner Port) 域下输入为伙伴 CPU 中的连接分配的端口。

7.6 PROFIBUS

PROFIBUS 系统使用总线主站来轮询 RS485

串行总线上以多点方式分布的从站设备。PROFIBUS

从站可以是任何处理信息并将其输出发送到主站的外围设备（I/O 传感器、阀、电机驱动器或其它测量设备）。

该从站构成网络上的被动站，因为它没有总线访问权限，只能确认接收到的消息或根据请求将响应消息发送给主站。所有 PROFIBUS

从站具有相同的优先级，并且所有网络通信都源于主站。

PROFIBUS 主站构成网络的“主动站”。PROFIBUS DP 定义两类主站。第 1

类主站（通常是中央可编程控制器 (PLC) 或运行特殊软件的

PC）处理与分配给它的从站之间的常规通信或数据交换。第 2

类主站（通常是组态设备，如用于调试、维护或诊断的膝上型计算机或编程控制台）是主要用于调试从站和诊断的特殊设备。

S7-1200 可通过 CM 1242-5 通信模块作为从站连接到 PROFIBUS 网络。CM 1242-5（DP 从站）模块可以是 DP V0/V1 主站的通信伙伴。

如果想在第三方系统中组态模块，可使用适合 CM 1242-5（DP 从站）的 GSD 文件，模块随附的 CD 或 Internet 上 Siemens 自动化客户支持

(<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=6GK72425DX300XE0&caller=view>)页面中提供了该文件。

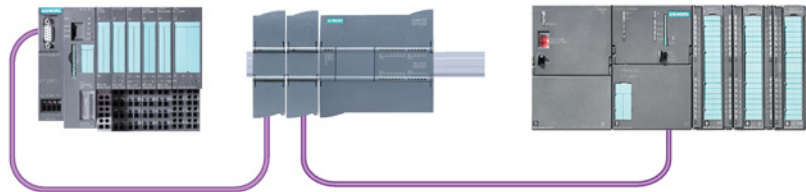
在下图中，S7-1200 是 S7-300 控制器的 DP 从站：



S7-1200 也可通过 CM 1243-5 通信模块作为主站连接到 PROFIBUS 网络。CM 1243-5（DP 主站）模块可以是 DP V0/V1 从站的通信伙伴。在下图中，S7-1200 是控制 ET200S DP 从站的主站：



如果同时安装了 CM 1242-5 和 CM 1243-5，则 S7-1200 既可充当上位 DP 主站系统的从站，又可充当下位 DP 从站系统的主站：



对于 V4.0，每站最多可组态三个 PROFIBUS CM，它们可以是 DP 主站或 DP 从站 CM 的任意组合。采用 V3.0 或更高版本的 CPU 固件时，每个 DP 主站最多可控制 32 个从站。

PROFIBUS CM 的组态数据存储在本地的 CPU 中。这样就可以在必要时方便地替换这些通信模块。

7.6.1 PROFIBUS CM 的通信服务

PROFIBUS CM 使用 PROFIBUS DP-V1 协议。

DP-V1 支持的通信类型

可通过 DP-V1 实现以下类型的通信：

- 周期性通信（CM 1242-5 和 CM 1243-5）

两个 PROFIBUS 模块支持周期性通信，因而可在 DP 从站和 DP 主站之间传送过程数据。

周期性通信由 CPU 的操作系统进行处理。此时不需要软件块。直接在 CPU 的过程映像中读取或写入 I/O 数据。

- 非周期性通信（仅限 CM 1243-5）

DP 主站模块还支持使用软件块进行非周期性通信：

- “RALRM”指令可用于处理中断。
- “RDREC”和“WRREC”指令可用于传送组态和诊断数据。

CM 1243-5 不支持的功能： SYNC/FREEZE 和 Get_Master_Diag

CM 1243-5 的其它通信服务

CM 1243-5 DP 主站模块另外还支持以下通信服务：

- S7 通信

- PUT/GET 服务

DP 主站起客户机和服务器的作用，可通过 PROFIBUS 对其它 S7 控制器或 PC 进行查询。

- PG/OP 通信

通过 PG 功能，可以从 PG 下载组态数据和用户程序，以及将诊断数据传送到 PG。

进行 OP 通信时，可用的通信伙伴有 HMI 面板、装有 WinCC flexible 的 SIMATIC 面板 PC 或者支持 S7 通信的 SCADA 系统。

7.6.2 PROFIBUS CM 用户手册参考资料

更多信息

有关 PROFIBUS CM 的详细信息，请参见设备手册。您可以在 Internet 的 Siemens 工业自动化客户支持页面上找到这些手册，相应的条目 ID 如下：


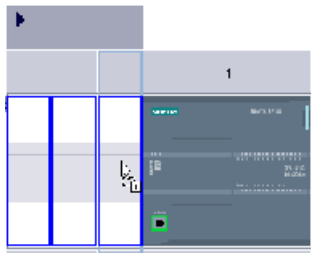
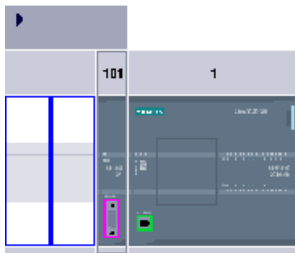
- CM 1242-5 (<http://support.automation.siemens.com/WW/view/en/49852105>)
- CM 1243-5 (<http://support.automation.siemens.com/WW/view/en/49851842>)

7.6.3 添加 CM 1243-5 (DP 主站) 模块和 DP 从站

在“设备和网络”(Devices and networks) 门户中，使用硬件目录向 CPU 添加 PROFIBUS 模块。这些模块连接在 CPU 左侧。

要将模块插入到硬件组态中，可在硬件目录中选择模块，然后双击该模块或将其拖到高亮显示的插槽中。

表格 7-7 将 PROFIBUS CM 1243-5 (DP 主站) 模块添加到设备组态



模块	选择模块	插入模块	结果
CM 1243-5 (DP 主站)			

同样也使用硬件目录添加 DP 从站。例如，要添加 ET200 S DP 从站，请在硬件目录中展开下列容器：

- 分布式 I/O
- ET200 S
- 接口模块
- PROFIBUS

接下来，从零件号列表中选择“6ES7 151-1BA02-0AB0”(IM151-1 HF)，如下图所示添加 ET200 S DP 从站。

表格 7-8 向设备组态添加 ET200 S DP 从站

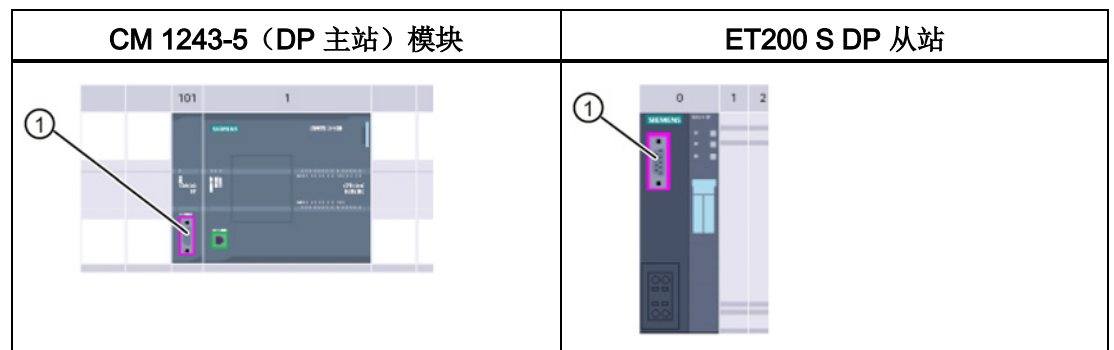
插入 DP 从站	结果
	

7.6.4 给 CM 1243-5 模块和 DP 从站分配 PROFIBUS 地址

组态 PROFIBUS 接口

组态两台 PROFIBUS 设备之间的逻辑网络连接后，便可以组态 PROFIBUS 接口的参数。为此，请单击 CM 1243-5 模块上的紫色 PROFIBUS 框，PROFIBUS 接口即显示在巡视窗口的“属性”(Properties) 选项卡中。以相同的方式组态 DP 从站 PROFIBUS 接口。

表格 7-9 组态 CM 1243-5 (DP 主站) 模块和 ET200 S DP 从站 PROFIBUS 接口



① PROFIBUS 端口

分配 PROFIBUS 地址

在 PROFIBUS 网络中，为每台设备分配了一个 PROFIBUS 地址。这个地址可以在 0 到 127 的范围内，但下列情况除外：

- 地址 0：为网络组态和/或连接到总线的编程工具保留
- 地址 1：Siemens 保留给第一个主站使用
- 地址 126：为不具有开关设置且必须通过网络重新寻址的出厂设备保留
- 地址 127：为给网络上所有设备广播消息保留，不可以分配给运转设备

因此，可用于 PROFIBUS 运转设备的地址的范围是 2 到 125。

在“属性”(Properties) 窗口中，选择“PROFIBUS 地址”(PROFIBUS address) 组态条目。STEP 7 将显示 PROFIBUS 地址组态对话框，该对话框用于分配设备的 PROFIBUS 地址。

7.6 PROFIBUS



表格 7- 10 PROFIBUS 地址的参数

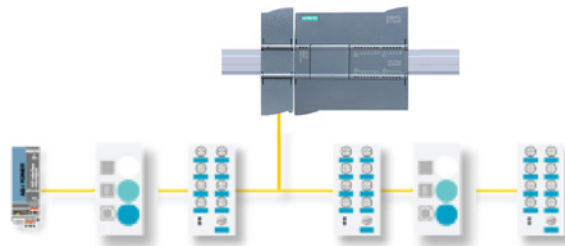
参数	说明	
子网	连接到设备的子网的名称。单击“添加新子网”(Add new subnet) 按钮以创建新的子网。默认为“未连接”(Not connected)。可以有两种连接类型： <ul style="list-style-type: none"> • 默认情况下“未连接”(Not connected) 提供本地连接。 • 网络具有两个或多个设备时，需要子网。 	
参数	地址	分配给设备的 PROFIBUS 地址
	最高地址	最高 PROFIBUS 地址基于 PROFIBUS 上的主动站（例如 DP 主站）。被动 DP 从站单独具有范围是 1 到 125 的 PROFIBUS 地址，即使最高 PROFIBUS 地址被设置为（例如）15。最高 PROFIBUS 地址与令牌传递有关（发送权限传递），并且令牌只传递给主动站。指定最高 PROFIBUS 地址可优化总线。
	传输率	组态的 PROFIBUS 网络的传输率： PROFIBUS 传输率的范围是 9.6 Kbps 到 12 Mbps。传输率设置取决于所使用的 PROFIBUS 节点的属性。传输率不应大于最慢节点所支持的传输率。通常需要为 PROFIBUS 网络上的主站设置传输率，而所有 DP 从站都将自动使用该传输率（自动波特）。

7.7 AS-i

通过 S7-1200 AS-i 主站 CM 1243-2 可将 AS-i 网络连接到 S7-1200 CPU。

执行器/传感器接口（或者说 AS-i）是自动化系统中最低级别的单一主站网络连接系统。CM 1243-2 作为网络中的 AS-i 主站。仅需一条 AS-i 电缆，即可将传感器和执行器（AS-i 从站设备）经由 CM 1243-2 连接到 CPU。CM 1243-2 可处理所有 AS-i 网络协调事务，并通过为其分配的 I/O 地址中继传输从执行器和传感器到 CPU 的数据和状态信息。根据从站类型，可以访问二进制值或模拟值。AS-i 从站是 AS-i 系统的输入和输出通道，并且只有在由 CM 1243-2 调用时才会激活。

在下图中，S7-1200 是控制 AS-i 数字量/模拟量 I/O 模块从站设备的 AS-i 主站。




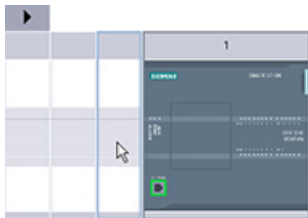
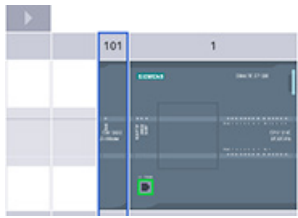
7.7 AS-i

7.7.1 添加 AS-i 主站 CM 1243-2 和 AS-i 从站

使用硬件目录将 AS-i 主站 CM1243-2 模块添加到 CPU。这些模块连接到 CPU 的左侧，并且最多可使用三个 AS-i 主站 CM1243-2 模块。

要将模块插入到硬件组态中，可在硬件目录中选择模块，然后双击该模块或将其拖到高亮显示的插槽中。

表格 7-11 向设备组态添加 AS-i 主站 CM1243-2 模块



模块	选择模块	插入模块	结果
CM 1243-2 AS-i 主站			

同样也使用硬件目录添加 AS-i 从站。例如，要添加“紧凑型数字量输入 I/O 模块”从站，请在硬件目录中展开下列容器：

- 现场设备
- AS-interface 接口从站

接下来，从零件号列表中选择“3RG9 001-0AA00”（AS-i SM-U, 4DI），并按下图所示添加“紧凑型数字量输入 I/O 模块”从站。

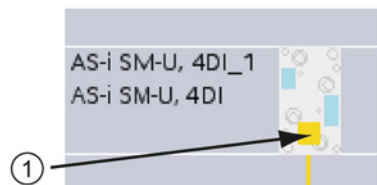
表格 7-12 向设备组态添加 AS-i 从站

插入 AS-i 从站	结果
	

7.7.2 为 AS-i 从站分配 AS-i 地址

组态 AS-i 从站接口

要组态 AS-i 接口的参数，请单击 AS-i 从站上的黄色 AS-i 框，巡视窗口的“属性”(Properties) 选项卡将显示该 AS-i 接口。



① AS-i 端口

分配 AS-i 从站地址

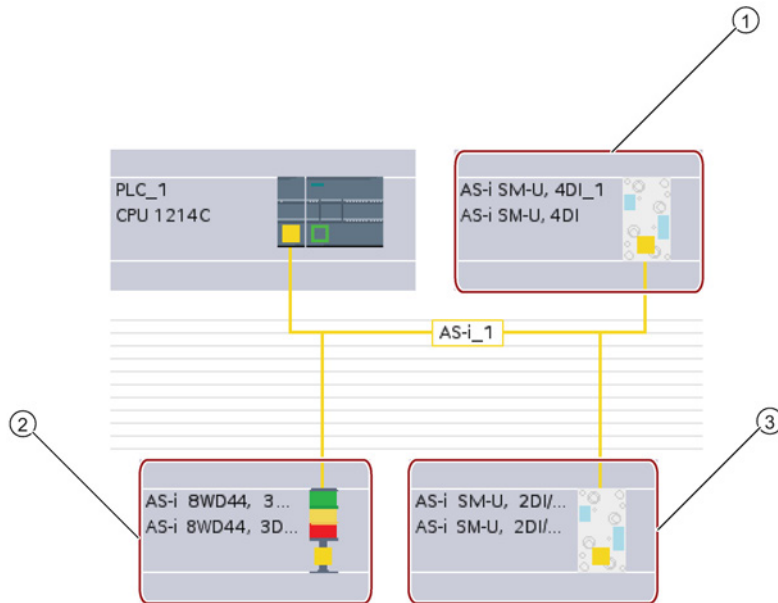
在 AS-i 网络中，每台设备都分配有一个 AS-i 从站地址。此地址的范围可从 0 到 31；但是，地址 0 只预留给新从站设备。从站地址从 1 (A 或 B) 一直到 31 (A 或 B)，总计最多 62 台从站设备。

“标准”AS-i 设备使用完整地址，其数字地址不带 A 或 B 标识。“A/B 节点”AS-i 设备的每个地址都有 A 或 B，这样 31 个地址全都可以使用两次。地址空间范围为 1A 到 31A 再加 1B 到 31B。

1 - 31 范围内的任何地址都可分配给 AS-i 从站设备；即，无论是从站从地址 21 开始，还是为第一个从站分配地址 1，都无关紧要。

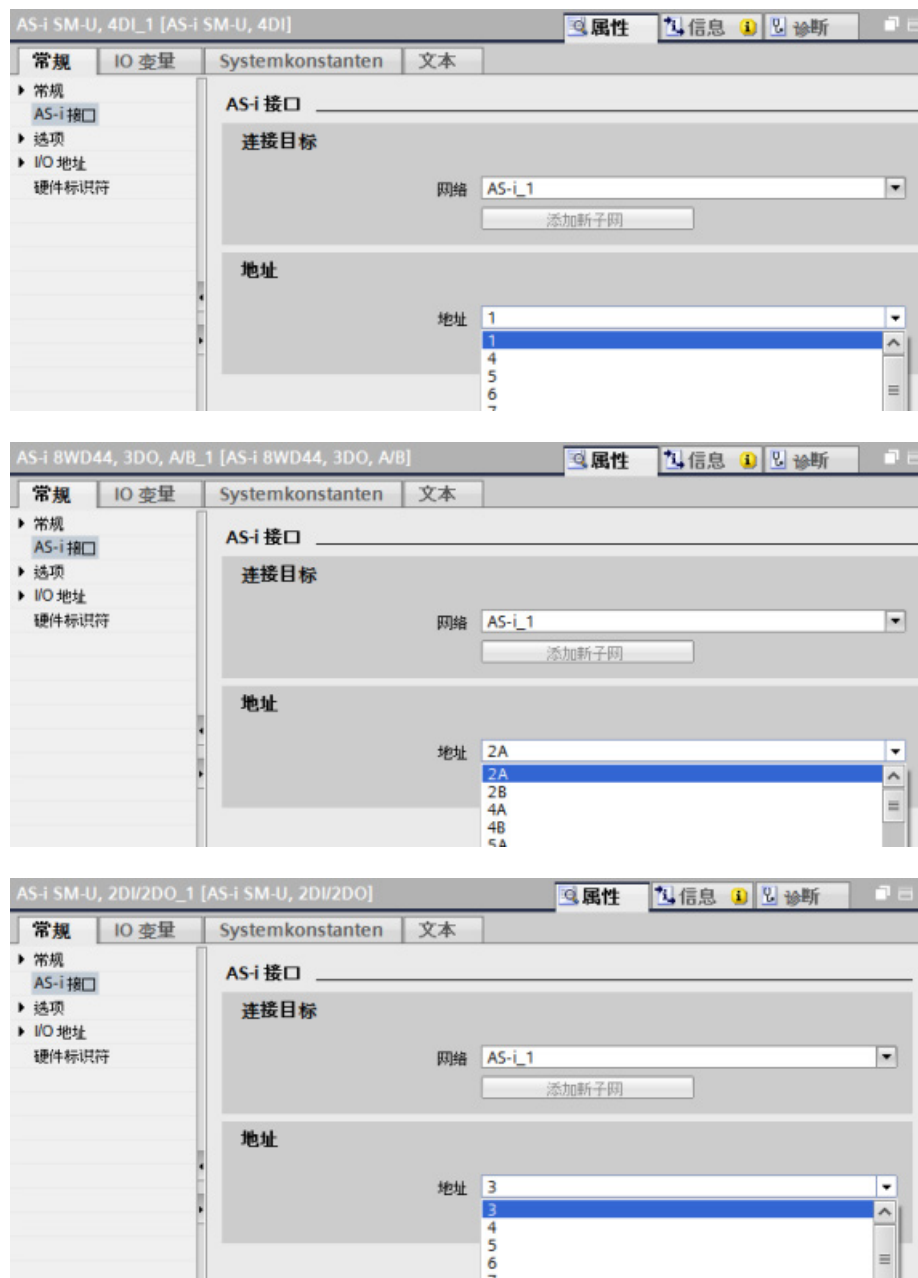
7.7 AS-i

在下面的示例中，三个 AS-i 设备的地址分别为“1”（标准类型设备）、“2A”（A/B 节点类型设备）和“3”（标准类型设备）：



- ① AS-i 从站地址 1; 设备: AS-i SM-U, 4DI; 订货号: 3RG9 001-0AA00
- ② AS-i 从站地址 2A; 设备: AS-i 8WD44, 3DO, A/B; 订货号: 8WD4 428-0BD
- ③ AS-i 从站地址 3; 设备: AS-i SM-U, 2DI/2DO; 订货号: 3RG9 001-0AC00

在此处输入 AS-i 从站地址:



表格 7- 13 AS-i 接口的参数

参数	说明
网络	设备所连接到的网络的名称
地址	为从站设备分配的 AS-i 地址范围是从 1 (A 或 B) 到 31 (A 或 B)，总计最多 62 台从站设备

7.8 S7 通信

7.8.1 GET 和 PUT 指令

可以使用 GET 和 PUT 指令通过 PROFINET 和 PROFIBUS 连接与 S7 CPU 通信。仅当在本地 CPU 属性的“保护”(Protection) 属性中为伙伴 CPU 激活了“允许使用 PUT/GET 通信进行访问”(Permit access with PUT/GET communication) 功能后，才可进行此操作：

- 访问远程 CPU 中的数据：S7-1200 CPU 在 ADDR_x 输入字段中只能使用绝对地址对远程 CPU (S7-200/300/400/1200) 的变量寻址。
- 访问标准 DB 中的数据：S7-1200 CPU 在 ADDR_x 输入字段中只能使用绝对地址对远程 S7 CPU 标准 DB 中的 DB 变量寻址。
- 访问优化 DB 中的数据：S7-1200 CPU 不能访问远程 S7-1200 CPU 的优化 DB 中的 DB 变量。
- 访问本地 CPU 中的数据：S7-1200 CPU 可使用绝对地址或符号地址分别作为 GET 或 PUT 指令的 RD_x 或 SD_x 输入字段的输入。

STEP 7 会在插入指令时自动创建该 DB。

说明

为确保数据的一致性，应始终在访问数据或启动另一读/写操作前评估已经完成的操作（对于 GET 评估 NDR = 1；对于 PUT 评估 DONE = 1）。

说明

V4.0 CPU 程序 GET/PUT 操作不会自动启用

V3.0 CPU 程序 GET/PUT 操作在 V4.0 CPU 中会自动启用。

不过，V4.0 CPU 中 V4.0 CPU 程序 GET/PUT 操作不会自动启用。要启用 GET/PUT 访问 (页 93)，必须转到 CPU“设备组态”(Device configuration)，打开巡视窗口，选择“属性”(Properties) 选项卡下的“保护”(Protection) 属性。

7.8.2 创建 S7 连接

连接机制

要使用 PUT/GET 指令访问远程连接伙伴，用户还必须得到许可。

默认情况下，“允许使用 PUT/GET 通信进行访问”(Permit access with PUT/GET communication) 选项处于未启用状态。这时，只有需要对本地 CPU 和通信伙伴同时进行组态和编程的通信连接才能实现对 CPU 数据的读写访问。例如，可以通过 BSEND/BRCV 指令进行访问。

因此，本地 CPU 仅作为服务器的连接（也就是说，本地 CPU 中不存在带有通信伙伴的通信组态/编程）在 CPU 运行期间不可用，例如：

- 通过通信模块进行 PUT/GET、FETCH/WRITE 或 FTP 访问
- 从其它 S7 CPU 进行 PUT/GET 访问
- 通过 PUT/GET 通信进行 HMI 访问

如果希望允许从客户端访问 CPU 数据，即不希望限制 CPU 的通信服务，要实现此级别的安全性，请参见“S7-1200 CPU 的访问保护 (页 93)”。

连接类型

所选的连接类型用于创建与伙伴站的通信连接。
控制器将设置、建立并自动监视该连接。

在“设备和网络”(Devices and Networks) 门户中，使用“网络视图”(Network view) 创建项目中各设备之间的网络连接。首先，请单击“连接”(Connections) 选项卡，然后使用右侧的下拉框选择连接类型（例如 S7 连接）。单击第一个设备上的绿色 (PROFINET) 框，然后拖出一条线连接到第二个设备上的 PROFINET 框。松开鼠标按钮，即可创建 PROFINET 连接。

更多相关信息，请参见“创建网络连接” (页 150)。

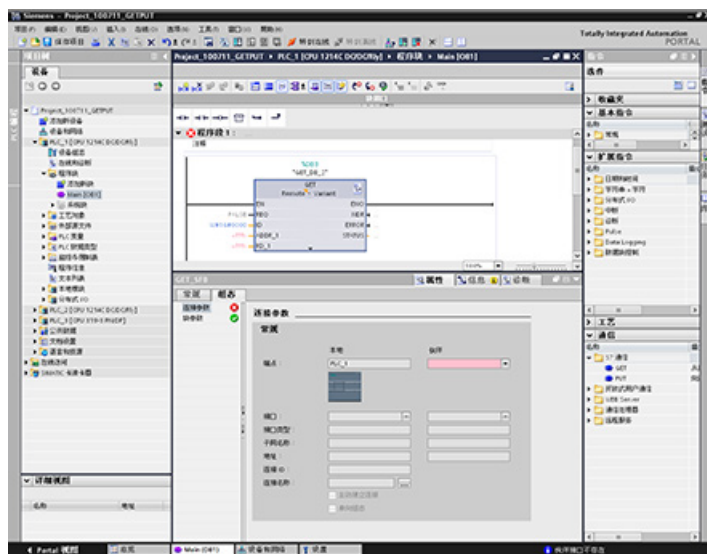


单击“突出显示：连接”(Highlighted: Connection) 按钮访问通信指令的“属性”(Properties) 组态对话框。

7.8.3 GET/PUT 连接参数分配

GET/PUT 指令连接参数分配是一项用于 S7 CPU-CPU 通信连接组态的用户辅助功能。

插入 GET 或 PUT 块后，GET/PUT 指令连接参数分配将启动：



每次选择指令的任何一部分，巡视窗口都会显示连接的属性。

在通信指令“属性”(Properties) 的“组态”(Configuration) 选项卡中指定通信参数。

插入 GET 或 PUT 块后，将自动出现“组态”(Configuration) 选项卡并立即显示“连接参数”(Connection parameters) 页面。

通过此页面，用户可组态必要的 S7 连接，组态块参数“ID”引用的参数“连接 ID”(Connection ID)。通过“块参数”(Block parameters) 页面可组态其它块参数。

说明

V4.0 CPU 程序 GET/PUT 操作不会自动启用

V3.0 CPU 程序 GET/PUT 操作在 V4.0 CPU 中会自动启用。

不过，V4.0 CPU 中 V4.0 CPU 程序 GET/PUT 操作不会自动启用。要启用 GET/PUT 访问(页 93)，必须转到 CPU“设备组态”(Device configuration)，打开巡视窗口，选择“属性”(Properties) 选项卡下的“保护”(Protection) 属性。

7.9 GPRS

7.9.1 连接到 GSM 网络

通过 GPRS 且基于 IP 的 WAN 通信

可以使用 CP 1242-7 通信处理器将 S7-1200 连接到 GSM 网络。使用 CP 1242-7 可以实现远程站与控制中心的 WAN 通信，以及站间通信。

站间通信只能通过 GSM 网络实现。

要在远程站和控制室之间进行通信，控制中心必须具备可以访问 Internet 的 PC。

CP 1242-7 支持通过 GSM 网络的以下通信服务：

- GPRS（General Packet Radio Service，通用分组无线服务）
通过 GSM 网络处理用于数据传输“GPRS”且面向数据包的服务。
- SMS（Short Message Service，短消息服务）

CP 1242-7 可以接收和发送 SMS 消息。通信伙伴可以是移动电话或 S7-1200。

CP 1242-7 适合在世界各地的工业领域使用，并且支持以下频段：

- 850 MHz
- 900 MHz
- 1,800 MHz
- 1,900 MHz

要求

站或控制中心使用的设备取决于具体的应用。

- 要与中央控制室通信或者要通过中央控制室进行通信，控制中心需要具备可以访问 Internet 的 PC。
- 除站设备之外，具有 CP 1242-7 的远程 S7-1200 站必须满足以下要求才能通过 GSM 网络进行通信：
 - 与相应的 GSM 网络供应商签订了合约
如果使用 GPRS，合约中必须允许使用 GPRS 服务。
如果存在仅通过 GSM 网络的站间直接通信，则 GSM 网络供应商必须为 CP 分配固定的 IP 地址。在这种情况下，站间通信不经过控制中心。
 - 合约中包含 SIM 卡
SIM 卡已插入 CP 1242-7 中。
 - 可在站范围内本地使用 GSM 网络

7.9.2 CP 1242-7 的应用

CP 1242-7 可用于以下应用：

遥控应用

- 通过 SMS 发送消息

通过 CP 1242-7，远程 S7-1200 站的 CPU 可以接收来自 GSM 网络的 SMS 消息，或者通过 SMS 向已组态的移动电话或 S7-1200 发送消息。

- 与控制中心的通信

远程 S7-1200 站通过 GSM 网络和 Internet 与主站中的遥控服务器进行通信。为了能够使用 GPRS 传输数据，需在主站的遥控服务器上安装“TELECONTROL SERVER BASIC”应用程序。遥控服务器使用集成的 OPC 服务器功能与更高层级的中央控制系统通信。

- S7-1200 站间通过 GSM 网络进行的通信

可通过两种不同的方式处理配有 CP 1242-7 的远程站之间的通信：

- 通过主站进行站间通信

在此组态的主站中，将要在各个 S7-1200 站之间建立一个永久的安全连接，便于各个站之间相互通信以及与遥控服务器之间进行通信。站间通信通过遥控服务器进行。CP 1242-7 在“Telecontrol”模式下运行。

- 站间直接通信

对于不经过主站的站间直接通信，使用带有固定 IP 地址的 SIM 卡，以便各站直接互相寻址。可能的通信服务和安全功能（例如 VPN）取决于网络供应商所提供的服务。CP 1242-7 在“GPRS 直接”模式下运行。

通过 GPRS 的 TeleService

可通过 GSM 网络和 Internet 在配有 STEP 7 的工程师站与配有 CP 1242-7 的远程 S7-1200 站之间建立 TeleService 连接。该连接从工程师站通过遥控服务器或 TeleService 网关（用作中介转发帧并建立相应授权）运行。这些 PC 使用“TELECONTROL SERVER BASIC”应用程序的功能。

可以将 TeleService 连接用于以下方面：

- 将组态或程序数据由 STEP 7 项目下载到工作站
- 查询工作站中的诊断数据

7.9.3 CP-1242-7 的其它属性

CP 1242-7 的其它服务和功能

- 通过 Internet 执行 CP 的日时钟同步
可以按照以下方法设置 CP 的时间：
 - 在“Telecontrol”模式下，由遥控服务器传送时间。CP 使用该时间来设置其自身的时间。
 - 在“GPRS 直接”模式下，CP 可以使用 SNTP 请求时间。

为同步 CPU 时间，可以用块从 CP 读出当前时间。

- 在存在连接问题时临时缓冲要发送的消息
- 由于可以连接到备用遥控服务器，提高了可用性
- 优化数据量（临时连接）

在 STEP 7 中，可为 CP

组态一个到遥控服务器的临时连接，用作到遥控服务器的永久连接的备选方案。在这种情况下，仅在需要时才建立到遥控服务器的连接。

- 记录数据量
记录已传送的数据量，并根据特定需要进行评估。

7.9.4 组态和电气连接

组态和模块替换

要组态模块，需要以下组态工具：

STEP 7 版本 V11.0 SP1 或更高版本

对于 STEP 7 V11.0 SP1，还需要支持包“CP 1242-7”(HSP0003001)。

要使用 GPRS 传输过程数据，请在站的用户程序中使用遥控通信指令。

CP 1242-7 的组态数据存储在本地 CPU 中。这样就可以在必要时方便地替换 CP。

每个 S7-1200 中最多可插入三个 CP 1242-7 类型的模块。这样便可建立冗余通信路径。

电气连接

- CP 1242-7 的电源
CP 具有一个用来连接外部 24 VDC 电源的单独连接。
- GSM 网络的无线接口
GSM 通信需要另外使用天线。这通过 CP 的 SMA 插座进行连接。

7.9.5 更多信息

更多信息

CP 1242-7 手册包含详细信息。您可以在 Internet 的西门子工业自动化客户支持页面上找到该手册，相应的条目 ID 如下：

45605894 (<http://support.automation.siemens.com/WW/view/zh/45605894>)

7.9.6 附件

ANT794-4MR GSM/GPRS 天线

下列天线可用于 GSM/GPRS 网络，且安装在室内和室外均可：

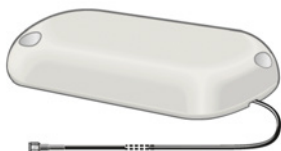
- 四频天线 ANT794-4MR



7.9 GPRS

简称	订货号	说明
ANT794-4MR	6NH9 860-1AA00	四频天线（900，1800/1900 MHz，UMTS）；防水；适合室内和室外使用；永久连接到天线的 5 m 连接电缆；SMA 连接器，包括安装支架、螺钉、墙用插座

- 平头天线 ANT794-3M



简称	订货号	说明
ANT794-3M	6NH9 870-1AA00	平头天线（900，1800/1900 MHz）；防水；适合室内和室外使用；永久连接到天线的 1.2 m 连接电缆；SMA 连接器，包括粘胶垫、可采用螺钉安装

必须单独订购天线。

7.9.7 参考 GSM 天线手册

更多信息

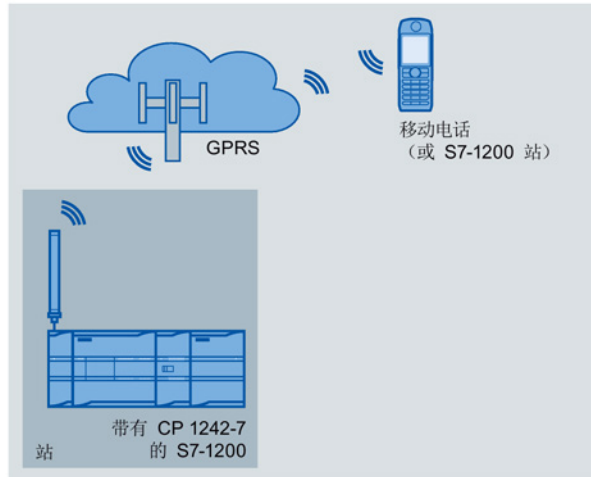
有关详细信息，请参见设备手册。您可以在 Internet 的西门子工业自动化客户支持页面上找到该手册，相应的条目 ID 如下：

23119005 (<http://support.automation.siemens.com/WW/view/zh/23119005>)

7.9.8 遥控组态示例

下文针对配有 CP 1242-7 的站提供了几个组态示例。

通过 SMS 发送消息



配有 CP 1242-7 的 SIMATIC S7-1200 可以通过 SMS 向移动电话或已组态的 S7-1200 站发送消息。

通过控制中心进行遥控

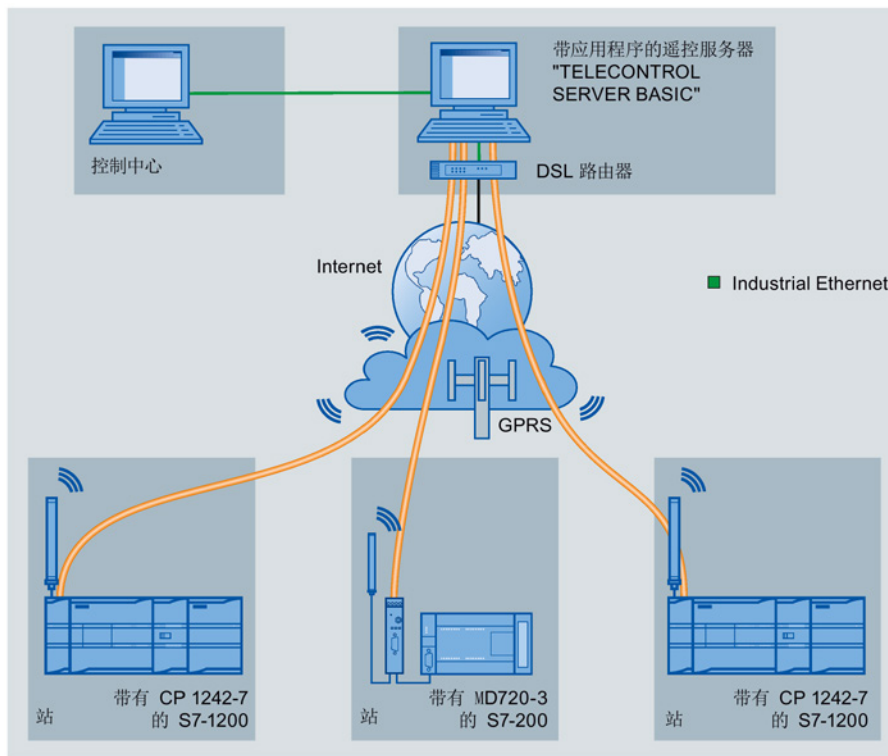


图 7-1 S7-1200 站与控制中心之间的通信

在遥控应用中，配有 CP 1242-7 的 SIMATIC S7-1200 通过 GSM 网络和 Internet 与控制中心通信。“TELECONTROL SERVER BASIC”(TCSB) 应用程序安装在主站的遥控服务器上。如此可实现以下应用：

- 工作站和控制中心之间的遥控通信

在该应用中，由工作站通过 GSM 网络和 Internet 将现场数据发送至主站的遥控服务器。遥控服务器用于监控远程站。

- 工作站和装有 OPC 客户端的控制室之间的通信

与第一种应用类似，工作站与遥控服务器进行通信。使用集成的 OPC 服务器，遥控服务器与控制室中的 OPC 客户端交换数据。

OPC 客户端和遥控服务器可位于一台计算机上，例如，当装有 WinCC 的控制中心计算机上安装了 TCSB 时，OPC 客户端和遥控服务器便可同时位于该计算机上。

- 通过控制中心进行站间通信

站间通信可通过配有 CP 1242-7 的 S7 站实现。

为了允许进行站间通信，遥控服务器会将发送站的消息转发到接收站。

站间直接通信

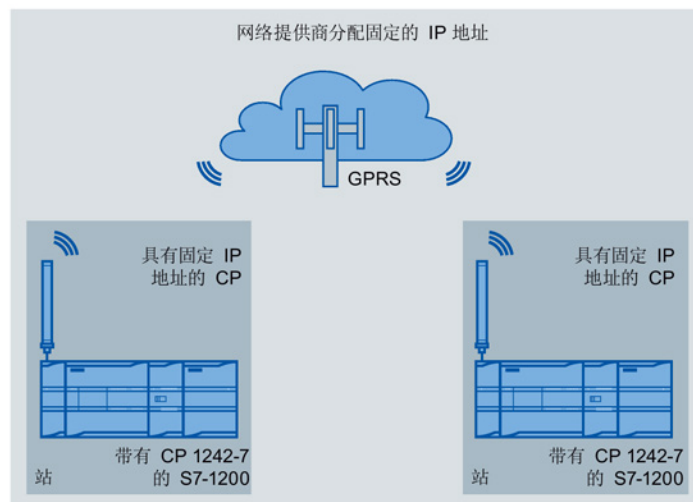


图 7-2 两个 S7-1200 站间直接通信

在此组态中，两个 SIMATIC S7-1200 站使用 CP 1242-7 通过 GSM 网络互相直接通信。每个 CP 1242-7 都有一个固定 IP 地址。GSM 网络供应商的相关服务必须允许这项规定。

通过 GPRS 的 TeleService

在通过 GPRS 的 TeleService 中，安装有 STEP 7 的工程师站通过 GSM 网络和 Internet 与 S7-1200 中的 CP 1242-7 进行通信。

由于防火墙对外部连接请求处于常闭状态，因此远程站和工程师站之间需要一个开关站。此开关站可以是一个遥控服务器，如果组态中没有遥控服务器，此开关站也可以是 TeleService 网关。

使用遥控服务器的 TeleService

连接通过遥控服务器运行。

- 工程师站和遥控服务器通过 Intranet (LAN) 或 Internet 进行连接。
- 遥控服务器和远程站通过 Intranet 或 GSM 网络进行连接。

工程师站和遥控服务器也可以是同一台计算机；换言之，也可将 STEP 7 和 TCSB 安装在同一台计算机上。

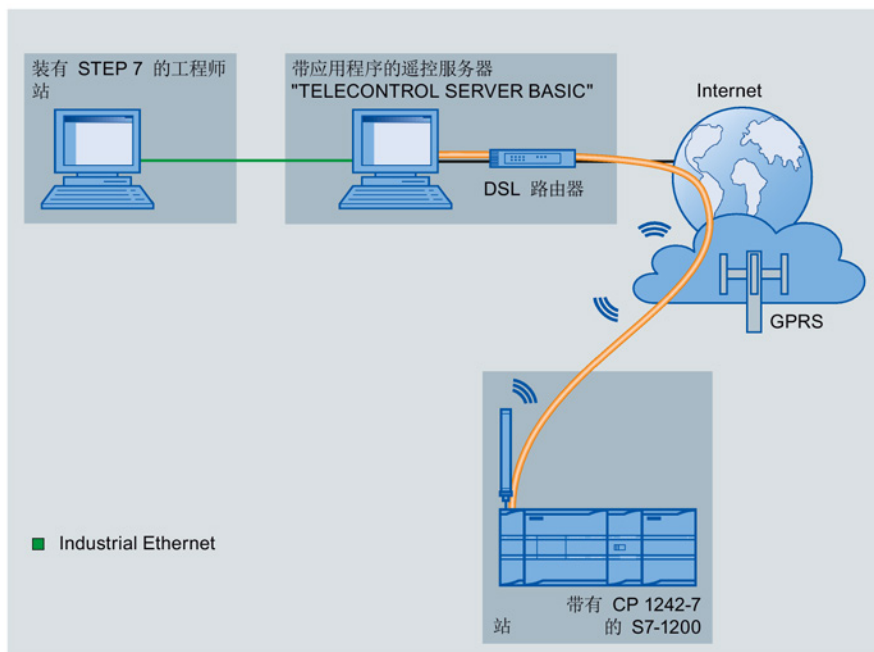


图 7-3 在使用遥控服务器的组态中通过 GPRS 的 TeleService

不使用遥控服务器的 TeleService

连接通过 TeleService 网关运行。

工程师站和 TeleService 网关间的连接可以通过 LAN 或 Internet 运行的本地连接。

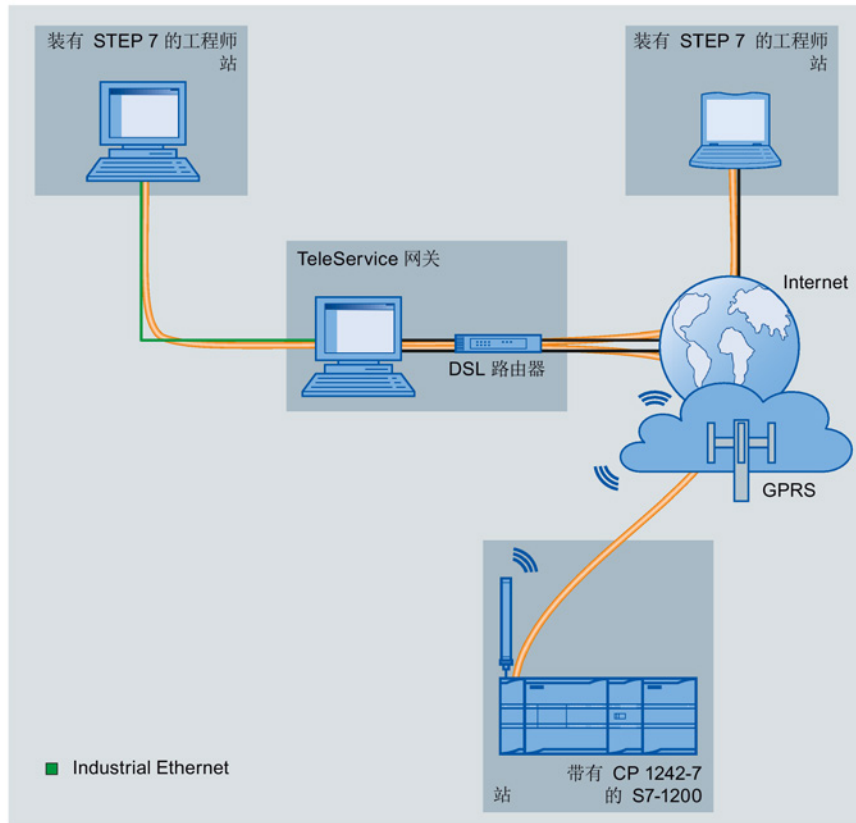


图 7-4 在使用 TeleService 网关的组态中通过 GPRS 的 TeleService

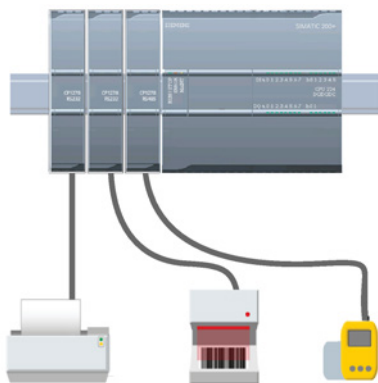
7.10 PtP、USS 和 Modbus 通信协议

7.10.1 点对点通讯

CPU 支持下列基于字符的串行协议的点对点通信 (PtP):

- PtP (页 197)
- USS (页 198)
- Modbus (页 200)

PtP 可提供最大的自由度和灵活性，但需要在用户程序中包含大量的实现。



PtP 可用于实现多种可能性:

- 能够将信息直接发送到外部设备，例如，打印机
- 能够从其它设备（例如，条码阅读器、RFID 阅读器、第三方照相机或视觉系统以及许多其它类型的设备）接收信息
- 能够与其它设备（例如，GPS 设备、第三方照相机或视觉系统、无线调制解调器以及更多其它设备）交换信息（发送和接收数据）

这种类型的 PtP 通信属于串行通信，它使用标准 UART 来支持多种波特率和奇偶校验选项。RS232 和 RS422/485 通信模块 (CM 1241) 以及 RS485 通信板 (CB 1241) 提供了用于执行 PtP 通信的电气接口。

通过 PROFIBUS 或 PROFINET 的 PtP

V4.1 版的 S7-1200 CPU 与 STEP 7 V13 SP1 一起使用可以扩展 PtP 的功能，使用户能够使用 PROFINET 或 PROFIBUS 分布式 I/O 机架与各种设备（RFID 阅读器、GPS 设备和其它设备）通信。

- PROFINET (页 157): 可以将 S7-1200 CPU 的以太网接口连接至 PROFINET 接口模块。可通过机架中 PtP 通信模块以接口模块实现与 PtP 设备的串行通信。
- PROFIBUS (页 169): 在 S7-1200 CPU 机架左边插入 PROFIBUS 通信模块。将 PROFIBUS 通信模块连接至 PROFIBUS 接口模块的机架。可通过机架中 PtP 通信模块以接口模块实现与 PtP 设备的串行通信。

出于这个原因，S7-1200 支持两组 PtP 指令：

- 早期点对点指令：这些指令适用于 V4.0 版之前的 S7-1200，并且只能通过 CM 1241 通信模块或 CB 1241 通信板进行串行通信。
- 点对点指令 (页 197): 这些指令提供早期指令的所有功能，通过这些指令还可以与 PROFINET 和 PROFIBUS 分布式 I/O 连接，可以组态分布式 I/O 机架中 PtP 通信模块与 PtP 设备之间的通信。

说明

用于 S7-1200 的 V4.1 版本时，可以对所有类型的点对点通信使用点对点指令：串行通信、基于 PROFINET 的串行通信和基于 PROFIBUS 的串行通信。STEP 7 提供早期点对点指令的目的仅是为了支持现有程序。无论对于 V4.1 CPU 或 V4.0 还是更早版本的 CPU，早期指令仍然有效。无须对之前程序的指令进行转换。

7.10.2 使用串行通信接口

以下两个通信模块 (CM, Communication Module) 和一个通信板 (CB, Communication Board) 提供了用于 PtP 通信的接口:

- CM 1241 RS232 (页 467)
- CM 1241 RS422/485 (页 465)
- CB 1241 RS485 (页 462)

最多可以连接三个 CM (类型不限) 外加一个 CB, 因而总共可提供四个通信接口。请将 CM 安装到 CPU 或另一个 CM 的左侧。将 CB 安装在 CPU 的前端。有关模块安装与卸下的信息, 请参见安装指南 (页 20)。

串行通信接口具有以下特征:

- 具有隔离的端口
- 支持点对点协议
- 通过点对点通信处理器指令进行组态和编程
- 通过 LED 显示传送和接收活动
- 显示诊断 LED (仅限 CM)
- 均由 CPU 供电: 不必连接外部电源。

请参见通信接口 (页 453) 的技术规范。

LED 指示灯

通信模块有三个 LED 指示灯:

- 诊断 LED (DIAG): 在 CPU 找到通信模块前, 诊断 LED 将一直以红色闪烁。CPU 在上电后会检查 CM, 并对其进行寻址。诊断 LED 开始以绿色闪烁。这表示 CPU 寻址到 CM, 但尚未为其提供组态。将程序下载到 CPU 后, CPU 会将组态下载到组态的 CM。执行下载到 CPU 操作后, 通信模块上的诊断 LED 应为绿色常亮。
- 发送 LED (Tx): 从通信端口向外传送数据时, 发送 LED 将点亮。
- 接收 LED (Rx): 通信端口接收数据时, 该 LED 将点亮。

通信板具有发送 LED (TxD) 和接收 LED (RxD)。但没有诊断 LED。

7.10.3 PtP 指令

通过 `Port_Config`、`Send_Config` 和 `Receive_Config` 指令，可以在用户程序中更改组态。

- `Port_Config` 用于更改端口参数，例如波特率。
- `Send_Config` 用于更改串行传输参数的组态。
- `Receive_Config` 用于更改通信端口中串行接收方参数的组态。
该指令可组态表示接收消息开始和结束的条件。`Receive_P2P` 指令将接收满足这些条件的消息。

动态组态更改不会永久存储在 CPU 中。

在上电循环后，将使用设备配置中的初始静态组态。

`Send_P2P`、`Receive_P2P` 和 `Receive_Reset` 指令用于控制 PtP 通信：

- `Send_P2P` 用于将指定的缓冲区数据传送到 CM 或 CB。
在模块以指定波特率发送数据的同时，CPU 会继续执行用户程序。
- `Receive_P2P` 用于检查 CM 或 CB 中已接收的消息。如果有消息，则将其传送到 CPU。
- `Receive_Reset` 用于复位接收缓冲区。

每个 CM 或 CB 最多可以缓冲 1K 字节。可以跨多个接收的消息来分配该缓冲区。

`Signal_Set` 和 `Signal_Get` 指令仅对 RS232 CM 有效。这些指令用于读取或设置 RS232 通信信号。

`Get_Features` 和 `Set_Features` 指令支持程序读取和设置模块功能。

7.10.4 USS 指令

S7-1200 支持 USS 协议，并提供了专门用于通过 CM 或 CB 的 RS485 端口与驱动器进行通信的指令。可使用 USS 指令控制物理驱动器和读/写驱动器参数。每个 RS485 CM 或 CB 最多支持 16 个驱动器。

- **USS_Port_Scan** 指令用于处理 CPU 与连接到某个 CM 或 CB 的所有驱动器之间的实际通信。在应用程序中分别为每个 CM 或 CB 插入不同的 **USS_Port_Scan** 指令。请确保用户程序以足够快的速度执行 **USS_Port_Scan** 指令，以防止与驱动器通信超时。**USS_Port_Scan** 指令用于程序循环 OB 或任何中断 OB。
- **USS_Drive_Control** 指令用于访问 USS 网络中的指定驱动器。**USS_Drive_Control** 指令的输入和输出参数代表驱动器的状态和控制。如果网络上有 16 个驱动器，则用户程序必须至少具有 16 个 **USS_Drive_Control** 指令，每个驱动器对应一个指令。

请确保 CPU 以控制驱动器功能所需的速率执行 **USS_Drive_Control** 指令。**USS_Drive_Control** 指令只能用于程序循环 OB。

- **USS_Read_Param** 和 **USS_Write_Param** 指令用于读取和写入远程驱动器的工作参数。这些参数控制驱动器的内部运行。有关这些参数的定义，请参见驱动器手册。

用户程序可根据需要包含任意数量的此类指令。

但在任何特定时刻，每个驱动器只能激活一个读或写请求。**USS_Read_Param** 和 **USS_Write_Param** 指令只能用于程序循环 OB。

对于与各个 CM 或 CB 相连的 USS 网络，背景数据块包含用于该网络中所有驱动器的临时存储区和缓冲区。驱动器的 USS 指令通过背景数据块来共享信息。

计算与驱动器通信所需的时间

与驱动器进行的通信与 CPU 扫描不同步。在完成一个驱动器通信事务之前，CPU 通常完成了多次扫描。

USS_Port_Scan 间隔是一个驱动器事务所需的时间。

下表列出了各个通信波特率下的最小 USS_Port_Scan 时间间隔。USS_Port_Scan 功能的调用间隔比 USS_Port_Scan 间隔短时，并不会增加事务数。

如果通信错误导致尝试 3

次才能完成事务，则驱动器超时间隔是处理该事务可能花费的时间。默认情况下，USS 协议库对每个事务最多自动进行 2 次重试。

表格 7- 14 计算时间要求

波特率	计算的最小 USS_Port_Scan 调用间隔 (毫秒)	每个驱动器的驱动器消息间隔超时 (毫秒)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

7.10.5 Modbus 指令

CPU 支持不同网络的上 Modbus 通信：

- Modbus RTU（远程终端单元）是一个标准的网络通信协议，它使用 RS232 或 RS485 电气连接在 Modbus 网络设备之间传输串行数据。可在带有一个 RS232 或 RS485 CM 或一个 RS485 CB 的 CPU 上添加 PtP（点对点）网络端口。

Modbus RTU

使用主/从网络，单个主设备启动所有通信，而从设备只能响应主设备的请求。主设备向从一个从设备地址发送请求，然后该从设备地址对命令做出响应。

- Modbus TCP（传输控制协议）是一个标准的网络通信协议，它使用 CPU 上的 PROFINET 连接器进行 TCP/IP 通信。不需要额外的通信硬件模块。

Modbus TCP 使用客户端-服务器连接作为 Modbus 通信路径。除了 STEP 7 和 CPU 之间的连接外，还可能存在多个客户端-服务器连接。

支持的混合客户端和服务器连接数最大为 CPU 所允许的最大连接值。每个 MB_SERVER 连接必须使用一个唯一的背景数据块和 IP 端口号。每个 IP 端口只能用于 1 个连接。必须为每个连接单独执行各 MB_SERVER（带有其唯一的背景数据块和 IP 端口）。



警告

如果攻击者能以物理方式访问您的网络，那么便可能读写数据。

TIA Portal、CPU 和 HMI（使用 GET/PUT 的 HMI

除外）均采用安全通信，可防止重放攻击和“中间人”攻击。

启用这种通信后，将以纯文本形式交换签名消息，这种方式允许攻击者读取数据，但可避免未经授权的数据写入操作。TIA

Portal（而非通信过程）将对受专有技术保护的块中的数据进行加密。

所有其它形式的通信（通过 PROFIBUS、PROFINET、AS-i 或其它 I/O 总线、GET/PUT、传输块 (T-block) 和通信模块 (CM) 进行的 I/O

交换）均没有安全功能。必须通过限制物理访问来保护这些形式的通信。

如果攻击者能利用这些形式的通信以物理方式访问您的网络，那么便可能读写数据。

有关安全信息和建议，请参见 Siemens 服务与支持网站上的“工业安全操作准则”。

www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf

[\(\[http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf\]\(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf\)\)](http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf)

说明

只有使用 CPU 固件版本 V1.02 或更高版本时，Modbus TCP 才能正确运行。

尝试在早期固件版本上执行 Modbus 指令会导致出错。

表格 7- 15 Modbus 指令

通信类型	指令
Modbus RTU (RS232 或 RS485)	Modbus_Comm_Load : 通过执行一次 Modbus_Comm_Load , 设置 PtP 端口参数, 如波特率、奇偶校验和流控制。为 Modbus RTU 协议组态 CPU 端口后, 该端口只能由 Modbus_Master 或 Modbus_Slave 指令使用。
	Modbus_Master : 该 Modbus 主指令使 CPU 充当 Modbus RTU 主设备, 并与一个或多个 Modbus 从设备进行通信。
	Modbus_Slave : 该 Modbus 从指令使 CPU 充当 Modbus RTU 从设备, 并与一个 Modbus 主设备进行通信。
Modbus TCP (PROFINET)	MB_CLIENT : 进行客户端-服务器 TCP 连接、发送命令消息、接收响应, 以及控制服务器断开。
	MB_SERVER : 根据要求连接至 Modbus TCP 客户端、接收 Modbus 信息及发送响应。

Modbus 指令不使用通信中断事件来控制通信用程。用户程序必须轮询 **Modbus_Master** / **Modbus_Slave** 或 **MB_CLIENT** / **MB_SERVER** 指令, 以了解传送和接收的完成情况。

Modbus TCP 客户端 (主站) 必须通过 **DISCONNECT** 参数控制客户端-服务器连接。基本的 **Modbus** 客户端操作如下所示。

1. 连接到特定服务器 (从站) IP 地址和 IP 端口号
2. 启动 **Modbus** 消息的客户端传输, 并接收服务器响应
3. 根据需要断开客户端和服务器的连接, 以便与其它服务器连接。

PID 非常简单

STEP 7 为 S7-1200 CPU 提供以下 PID 指令：

- **PID_Compact** 指令用于通过连续输入变量和输出变量控制工艺过程。
- **PID_3Step**
指令用于控制电机驱动的设备，如需要通过离散信号实现打开和关闭动作的阀门。
- **PID_Temp** 指令提供一个通用的 PID 控制器，可用于处理温度控制的特定需求。

说明

只有 CPU 从 STOP 切换到 RUN 模式后，在 RUN 模式下对 PID 组态和下载进行的更改才会生效。而在“PID 参数”(PID parameters)对话框中使用“起始值控制”(Start value control) 进行的更改立即生效。

全部三个 PID 指令（PID_Compact、PID_3Step 和 PID_Temp）都可以计算启动期间的 P 分量、I 分量以及 D 分量（如果组态为“预调节”）。
还可以将指令组态为“精确调节”，从而可对参数进行优化。用户无需手动确定参数。

说明

以恒定的采样时间间隔执行 PID 指令（最好在循环 OB 中）。

由于 PID 回路需要一段时间来响应控制值的变化，因此请勿在每个循环中都计算输出值。请勿在主程序循环 OB（如 OB 1）中执行 PID 指令。

PID 算法的采样时间表示两次输出值（控制值）计算之间的时间。
在自调节期间计算输出值，并取整为循环时间的倍数。每次调用时都会执行 PID 指令的所有其它函数。

PID 算法

PID（比例/积分/微分）控制器会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。每次进行模式切换时以及初始启动期间都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。

PID 控制器的输出值由三个分量组成：

- P（比例）：
如果通过“P”分量计算，则输出值与设定值和过程值（输入值）之差成比例。
- I（积分）：
如果通过“I”分量计算，则输出值与设定值和过程值（输入值）之差的持续时间成比例增加，以最终校正该差值。
- D（微分）：
如果通过“D”分量计算，输出值与设定值和过程值（输入值）之差的变化率成函数关系，并随该差值的变化加快而增大。从而根据设定值尽快矫正输出值。

PID 控制器使用以下公式来计算 PID_Compact 指令的输出值。

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

y	输出值	x	过程值
w	设定值	s	拉普拉斯算子
K _p	比例增益 (P 分量)	a	微分延迟系数 (D 分量)
T _i	积分作用时间 (I 分量)	b	比例作用加权 (P 分量)
T _D	微分作用时间 (D 分量)	c	微分作用加权 (D 分量)

PID 控制器使用以下公式来计算 PID_3Step 指令的输出值。

$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

y	输出值	x	过程值
w	设定值	s	拉普拉斯算子
K _p	比例增益 (P 分量)	a	微分延迟系数 (D 分量)
T _i	积分作用时间 (I 分量)	b	比例作用加权 (P 分量)
T _D	微分作用时间 (D 分量)	c	微分作用加权 (D 分量)

8.1 插入 PID 指令和工艺对象


STEP 7 提供了两个 PID 控制指令：

- **PID_Compact** 指令及其相关工艺对象提供具有调节功能的通用 PID 控制器。工艺对象中包含控制环的所有设置。
- **PID_3Step** 指令及其相关工艺对象为通过电机驱动的阀门提供具有特定设置的 PID 控制器。工艺对象中包含控制环的所有设置。PID_3Step 控制器提供两个附加的布尔型输出。

创建工艺对象之后，必须组态参数 (页 239)。



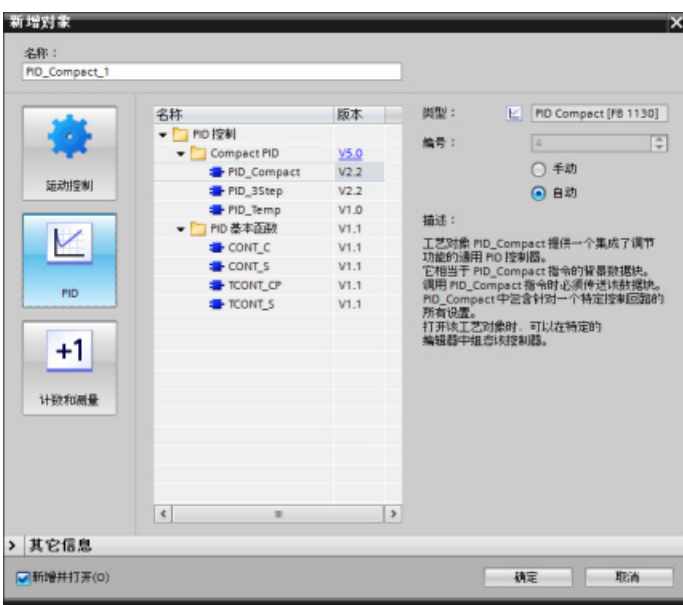
还应调整自动调节参数（启动期间的“预调节”或手动“精确调节”），以调试 PID 控制器的操作 (页 260)。

表格 8-1 插入 PID 指令和工艺对象

<p>将 PID 指令插入用户程序时，STEP 7 会自动为指令创建工艺对象和背景数据块。背景数据块包含 PID 指令要使用的所有参数。每个 PID 指令必须具有自身的唯一背景数据块才能正确工作。</p> <p>插入 PID 指令并创建工艺对象和背景数据块之后，需组态工艺对象的参数 (页 239)。</p>	
--	---

8.1 插入 PID 指令和工艺对象

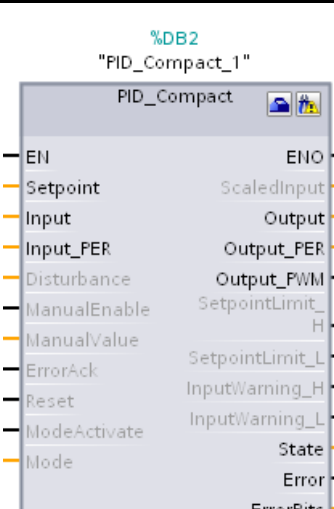
表格 8-2 (可选) 通过项目浏览器创建工艺对象

<p>还可以在插入 PID 指令之前为项目创建工艺对象。如果在将 PID 指令插入用户程序之前创建工艺对象，用户便可以在插入 PID 指令时选择工艺对象。</p>	
<p>要创建工艺对象，请在项目浏览器中双击“添加新对象”(Add new object) 图标。</p>	
<p>单击“控制”(Control) 图标并选择适用于该 PID 控制器类型 (PID_Compact 或 PID_3Step) 的工艺对象。 可以为工艺对象创建可选名称。 单击“确定”(OK) 创建工艺对象。</p>	

8.2 PID_Compact 指令

PID_Compact 指令提供自动和手动模式下具有集成自我调节功能的通用 PID 控制器。

表格 8-3 PID_Compact 指令

LAD/FBD	SCL	说明
 <p>The screenshot shows the LAD/FBD view of the PID_Compact instruction. It is a function block with the following parameters:</p> <ul style="list-style-type: none"> EN (Input), ENO (Output) Setpoint (Input), ScaledInput (Output) Input (Input), Output (Output) Input_PER (Input), Output_PER (Output) Disturbance (Input), Output_PWM (Output) ManualEnable (Input), SetpointLimit_H (Output) ManualValue (Input), SetpointLimit_L (Output) ErrorAck (Input), InputWarning_H (Output) Reset (Input), InputWarning_L (Output) ModeActivate (Input), State (Output) Mode (Input), Error (Output) ErrorBits (Output) 	<pre>"PID_Compact_1" (Setpoint:=_real_in_, Input:=_real_in_, Input_PER:=_word_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualValue:=_real_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, ScaledInput=>_real_out_, Output=>_real_out_, Output_PER=>_word_out_, Output_PWM=>_bool_out_, SetpointLimit_H=>_bool_out_, SetpointLimit_L=>_bool_out_, InputWarning_H=>_bool_out_, InputWarning_L=>_bool_out_, State=>_int_out_, Error=>_bool_out_, ErrorBits=>_dword_out_);</pre>	<p>PID_Compact 提供可在自动模式和手动模式下自我调节的 PID 控制器。PID_Compact 是具有抗积分饱和功能且对 P 分量和 D 分量加权的 PID T1 控制器。</p>

- STEP 7 会在插入指令时自动创建工艺对象和背景数据块。该背景数据块包含工艺对象的参数。
- 在 SCL 示例中，“PID_Compact_1”是背景 DB 的名称。

表格 8-4 参数的数据类型

参数和类型	数据类型	说明
Setpoint	IN	Real PID 控制器在自动模式下的设定值。（默认值：0.0）
Input	IN	Real 用户程序的变量用作过程值的源。（默认值：0.0） 如果正在使用 Input 参数，则必须设置 Config.InputPerOn = FALSE。
Input_PER	IN	Word 模拟量输入用作过程值的源。（默认值：W#16#0） 如果正在使用 Input_PER 参数，则必须设置 Config.InputPerOn = TRUE。

参数和类型		数据类型	说明
Disturbance	IN	Real	干扰变量或预控制值
ManualEnable	IN	Bool	<p>启用或禁用手动操作模式。（默认值：FALSE）：</p> <ul style="list-style-type: none"> FALSE 至 TRUE 沿激活“手动模式”，同时 State = 4，Mode 保持不变。 <p>ManualEnable = TRUE 时，无法利用 ModeActivate 的上升沿或使用调试对话框更改工作模式。</p> <ul style="list-style-type: none"> TRUE 至 FALSE 沿激活 Mode 分配的工作模式。 <p>注：建议您只使用 ModeActivate 更改工作模式。</p>
ManualValue	IN	Real	<p>手动操作的输出值。（默认值：0.0）</p> <p>可以使用从 Config.OutputLowerLimit 到 Config.OutputUpperLimit 的值。</p>
ErrorAck	IN	Bool	复位 ErrorBits 和警告输出。FALSE 至 TRUE 沿
Reset	IN	Bool	<p>重新启动控制器。（默认值：FALSE）：</p> <ul style="list-style-type: none"> FALSE 至 TRUE 沿： <ul style="list-style-type: none"> 切换到“未激活”模式 复位 ErrorBits 和警告输出 清除积分作用 保持 PID 参数 只要 Reset = TRUE，则 PID_Compact 便会保持在“未激活”模式 (State = 0)。 TRUE 至 FALSE 沿： <ul style="list-style-type: none"> PID_Compact 切换到保存在 Mode 参数中的工作模式。
ModeActivate	IN	Bool	PID_Compact 切换到保存在 Mode 参数中的工作模式。FALSE 至 TRUE 沿：
Mode	IN	Int	期望的 PID 模式；在 Mode Activate 输入的上升沿激活。
ScaledInput	OUT	Real	标定的过程值。（默认值：0.0）
Output ¹	OUT	Real	REAL 格式的输出值。（默认值：0.0）
Output_PER ¹	OUT	Word	模拟量输出值。（默认值：W#16#0）
Output_PWM ¹	OUT	Bool	<p>脉冲宽度调制的输出值。（默认值：FALSE）</p> <p>开关时间构成输出值。</p>

参数和类型		数据类型	说明
SetpointLimit_H	OUT	Bool	设定值上限。（默认值：FALSE） 如果 SetpointLimit_H = TRUE，则说明达到设定值的绝对上限 (Setpoint \geq Config.SetpointUpperLimit)。 设定值限制为 Config.SetpointUpperLimit。
SetpointLimit_L	OUT	Bool	设定值下限。（默认值：FALSE） 如果 SetpointLimit_L = TRUE，则说明达到设定值的绝对下限 (Setpoint \geq Config.SetpointLowerLimit)。 设定值限制为 Config.SetpointLowerLimit。
InputWarning_H	OUT	Bool	如果 InputWarning_H = TRUE，则说明过程值已达到或超出警告上限。（默认值：FALSE）
InputWarning_L	OUT	Bool	如果 InputWarning_L = TRUE，则说明过程值已达到或低于警告下限。（默认值：FALSE）
State	OUT	Int	PID 控制器的当前操作模式。（默认值：0） 可以使用 Mode 输入参数和 ModeActivate 的上升沿更改工作模式： <ul style="list-style-type: none"> • State = 0: 未激活 • State = 1: 预调节 • State = 2: 手动精确调节 • State = 3: 自动模式 • State = 4: 手动模式 • State = 5: 通过错误监视替换输出值

8.2 PID_Compact 指令

参数和类型	数据类型	说明
Error	OUT	Bool 如果 Error = TRUE，则该周期内至少有一条错误消息未决。（默认值：FALSE） 注： V1.x PID 中的 Error 参数是包含错误代码的 ErrorBits 字段。它现在是一个布尔标记，说明有错误发生。
ErrorBits	OUT	DWord PID_Compact 指令 ErrorBits 参数表 (页 212)定义未决的错误消息。（默认值：DW#16#0000（无错误））。ErrorBits 具有保持性并在 Reset 或 ErrorAck 的上升沿复位。 注： 在 V1.x 中，ErrorBits 参数定义为 Error 参数并且不存在。

1 您可以并行使用 Output、Output_PER 和 Output_PWM 参数的输出。

PID_Compact 控制器的操作

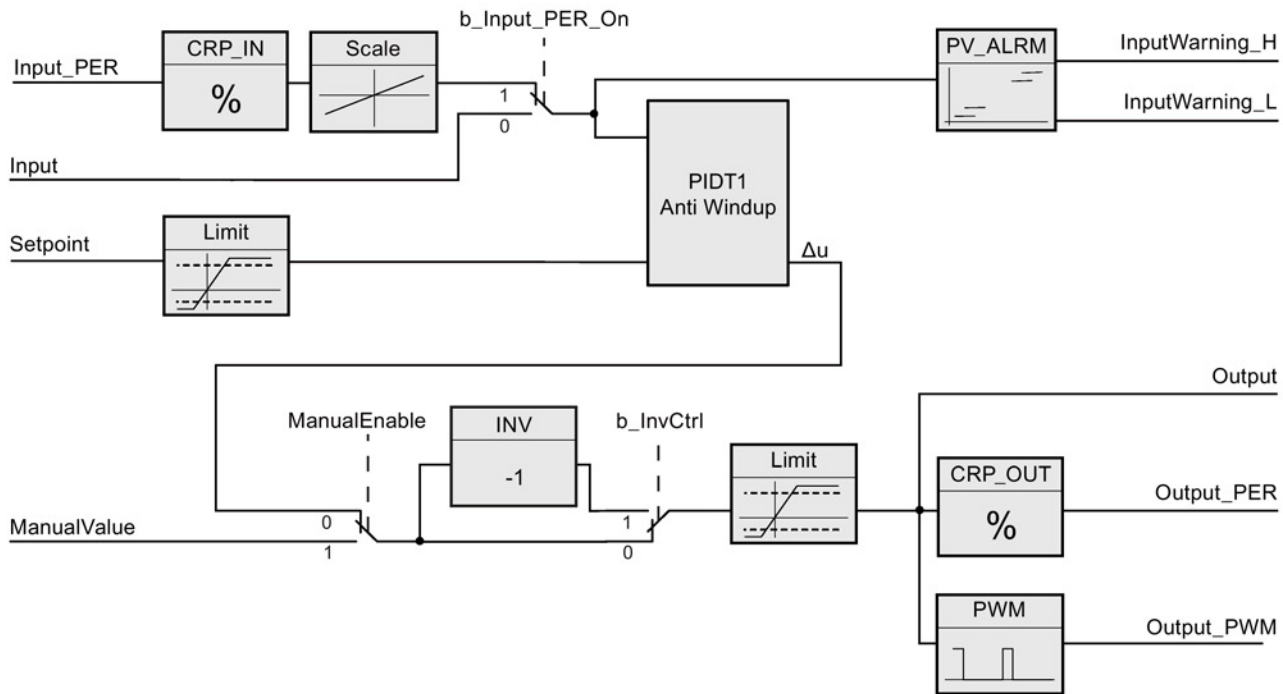


图 8-1 PID_Compact 控制器的操作

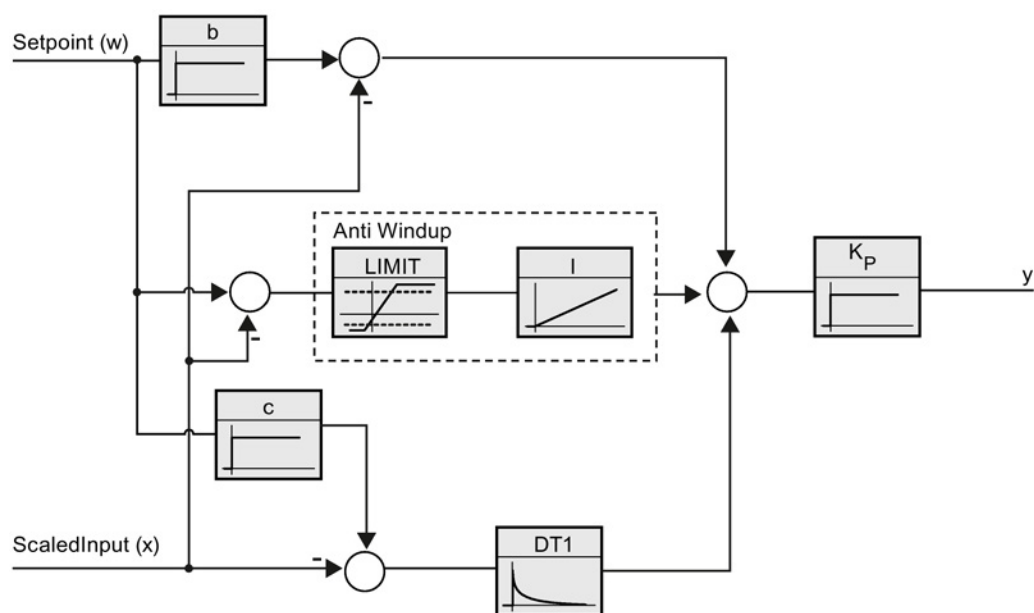


图 8-2 PID_Compact 控制器作为具有抗积分饱和功能的 PDI1 控制器时的操作

8.3 PID_Compact 指令 ErrorBit 参数

如果存在多个错误未决，则错误代码的值将通过二进制加法显示。例如，显示错误代码 0003 表示错误 0001 和 0002 未决。

表格 8-5 PID_Compact 指令 ErrorBit 参数

ErrorBit (DW#16#...)	说明
0000	无错误
0001 ^{1,2}	参数 Input 超出了过程值限值的范围。 Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2,3}	参数 Input_PER 的值无效。 请检查模拟量输入是否有错误尚未解决。
0004 ⁴	精确调节期间出错。无法保持过程值的振荡。
0008 ⁴	预调节开始时出错。过程值过于接近设定值。开始精确调节。
0010 ⁴	调节期间设定值发生改变。 注意：可在 CancelTuningLevel 变量中设置允许的设定值波动。
0020	精确调节期间不允许预调节。 注意：如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Compact 保持在精确调节模式。
0080 ⁴	预调节期间出错。输出值限值的组态不正确。 检查是否已正确组态输出值的限值以及该限值是否与控制逻辑匹配。
0100 ⁴	精确调节期间由于无效参数导致出错。
0200 ^{2,3}	参数 Input 的值无效：值的数字格式无效。
0400 ^{2,3}	输出值计算失败。检查 PID 参数。
0800 ^{1,2}	采样时间错误：循环中断 OB 的采样时间内没有调用 PID_Compact。
1000 ^{2,3}	参数 Setpoint 的值无效：值的数字格式无效。

ErrorBit (DW#16#...)	说明
10000	<p>参数 ManualValue 的值无效：值的数字格式无效。</p> <p>注意：如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Compact 使用 SubstituteOutput 作为输出值。只要在 ManualValue 参数中分配有效值，PID_Compact 便会将其用作输出值。</p>
20000	<p>变量 SubstituteValue 的值无效：值的数字格式无效。</p> <p>PID_Compact 使用输出值下限作为输出值。</p> <p>注意： 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_Compact 切换回自动模式。</p>
40000	<p>参数 Disturbance 的值无效：值的数字格式无效。</p> <p>注意：如果在错误发生前自动模式已激活且 ActivateRecoverMode = FALSE，则 Disturbance 将设置为零。PID_Compact 保持自动模式。</p> <p>注意：如果在错误发生前预调节或精确调节已激活且 ActivateRecoverMode = TRUE，则 PID_Compact 切换到 Mode 参数中保存的工作模式。</p> <p>如果当前阶段中的干扰对输出值无影响，则不会取消调节。</p>

- 1 注：如果在错误发生前自动模式已激活且 **ActivateRecoverMode = TRUE**，则 **PID_Compact** 保持自动模式。
- 2 注：如果在错误发生前预调节或精确调节已激活且 **ActivateRecoverMode = TRUE**，则 **PID_Compact** 切换到 **Mode** 参数中保存的工作模式。
- 3 注：如果在错误发生前自动模式已激活且 **ActivateRecoverMode = TRUE**，则 **PID_Compact** 输出组态的替换输出值。当错误不再处于未决状态时，**PID_Compact** 切换回自动模式。
- 4 注：如果在错误发生前 **ActivateRecoverMode = TRUE**，则 **PID_Compact** 取消调节并切换到 **Mode** 参数中保存的工作模式。

8.4 PID_3Step 指令

PID_3Step 指令用于组态具有自调节功能的 PID 控制器，这样的控制器已针对通过电机控制的阀门和执行器进行过优化。

表格 8-6 PID_3Step 指令

LAD/FBD	SCL	说明
	<pre>"PID_3Step_1" (SetpoInt:= _real_in_, Input:= _real_in_, ManualValue:= _real_in_, Feedback:= _real_in_, InputPer:= _word_in_, FeedbackPer:= _word_in_, Disturbance:= _real_in_, ManualEnable:= _bool_in_, ManualUP:= _bool_in_, ManualDN:= _bool_in_, ActuatorH:= _bool_in_, ActuatorL:= _bool_in_, ErrorAck:= _bool_in_, Reset:= _bool_in_, ModeActivate:= _bool_in_, Mode:= _int_in_, ScaledInput=> _real_out_, ScaledFeedback=> _real_out_, ErrorBits=> _dword_out_, OutputPer=> _word_out_, State=> _int_out_, OutputUP=> _bool_out_, OutputDN=> _bool_out_, SetpoIntLimitH=> _bool_out_, SetpoIntLimitL=> _bool_out_, InputWarningH=> _bool_out_, InputWarningL=> _bool_out_, Error=> _bool_out_, ErrorBits=> dword out);</pre>	<p>PID_3Step 用于组态具有自调节功能的 PID 控制器，这样的控制器已针对通过电机控制的阀门和执行器进行过优化。它提供两个布尔型输出。</p> <p>PID_3Step 是具有抗积分饱和功能且对 P 分量和 D 分量加权的 PID T1 控制器。</p>

- STEP 7 会在插入指令时自动创建工艺对象和背景数据块。该背景数据块包含工艺对象的参数。
- 在 SCL 示例中，“PID_3Step_1”是背景 DB 的名称。

表格 8-7 参数的数据类型

参数和类型		数据类型	说明
Setpoint	IN	Real	PID 控制器在自动模式下的设定值。（默认值：0.0）
Input	IN	Real	用户程序的变量用作过程值的源。（默认值：0.0） 如果正在使用 Input 参数，则必须设置 Config.InputPerOn = FALSE。
Input_PER	IN	Word	模拟量输入用作过程值的源。（默认值：W#16#0） 如果正在使用 Input_PER 参数，则必须设置 Config.InputPerOn = TRUE。
Actuator_H	IN	Bool	上端停止位阀门的数字位置反馈 如果 Actuator_H = TRUE，则阀门处于上端停止位，且不再向此方向移动。（默认值：FALSE）
Actuator_L	IN	Bool	下端停止位阀门的数字位置反馈 如果 Actuator_L = TRUE，则阀门处于下端停止位，且不再向此方向移动。（默认值：FALSE）
Feedback	IN	Real	阀门的位置反馈。（默认值：0.0） 如果正在使用 Feedback 参数，则必须设置 Config.FeedbackPerOn = FALSE。
Feedback_PER	IN	Int	阀门位置的模拟反馈。（默认值：W#16#0） 如果正在使用 Feedback_PER 参数，则必须设置 Config.FeedbackPerOn = TRUE. Feedback_PER 根据以下变量标定： <ul style="list-style-type: none"> • Config.FeedbackScaling.LowerPointIn • Config.FeedbackScaling.UpperPointIn • Config.FeedbackScaling.LowerPointOut • Config.FeedbackScaling.UpperPointOut
Disturbance	IN	Real	干扰变量或预控制值

参数和类型		数据类型	说明
ManualEnable	IN	Bool	<p>启用或禁用手动操作模式。（默认值： FALSE）：</p> <ul style="list-style-type: none"> FALSE 至 TRUE 沿激活“手动模式”，同时 State = 4， Mode 保持不变。 <p>ManualEnable = TRUE 时，无法利用 ModeActivate 的上升沿或使用调试对话框更改工作模式。</p> <ul style="list-style-type: none"> TRUE 至 FALSE 沿激活 Mode 分配的工作模式。 <p>注： 建议您只使用 ModeActivate 更改工作模式。</p>
ManualValue	IN	Real	<p>手动操作的过程值。（默认值： 0.0）</p> <p>在手动模式下，可指定阀门的绝对位置。仅当正使用 OutputPer 或者位置反馈可用时，才评估 ManualValue。</p>
ManualUP	IN	Bool	<ul style="list-style-type: none"> Manual_UP = TRUE: <ul style="list-style-type: none"> 即使使用 Output_PER 或位置反馈，阀门也会打开。如果未到达上端停止位，阀门将不再移动。 另请参见Config.VirtualActuatorLimit Manual_UP = FALSE: <ul style="list-style-type: none"> 使用 Output_PER 或位置反馈会使阀门移动到 ManualValue。否则阀门将不再移动。 <p>注： Manual_UP 和 Manual_DN 同时设为 TRUE 时，阀门不再移动。</p>
ManualDN	IN	Bool	<ul style="list-style-type: none"> Manual_DN = TRUE: <ul style="list-style-type: none"> 即使使用 Output_PER 或位置反馈，阀门也会打开。如果未到达上端停止位，阀门将不再移动。 另请参见Config.VirtualActuatorLimit Manual_DN = FALSE: <ul style="list-style-type: none"> 使用 Output_PER 或位置反馈会使阀门移动到 ManualValue。否则阀门将不再移动。
ErrorAck	IN	Bool	<p>复位 ErrorBits 和警告输出。FALSE 至 TRUE 沿</p>

参数和类型		数据类型	说明
Reset	IN	Bool	重新启动控制器。（默认值：FALSE）： <ul style="list-style-type: none"> FALSE 至 TRUE 沿： <ul style="list-style-type: none"> 切换到“未激活”模式 复位 ErrorBits 和警告输出 清除积分作用 保持 PID 参数 只要 Reset = TRUE，则 PID_3Step 便会保持在“未激活”模式 (State = 0)。 TRUE 至 FALSE 沿： <ul style="list-style-type: none"> PID_3Step 切换到保存在 Mode 参数中的工作模式。
ModeActivate	IN	Bool	PID_3Step 切换到保存在 Mode 参数中的模式。FALSE 至 TRUE 沿：
Mode	IN	Int	期望的 PID 模式；在 Mode Activate 输入的上升沿激活。
ScaledInput	OUT	Real	标定的过程值
ScaledFeedback	OUT	Real	标定的阀门位置反馈 注：对于无位置反馈的执行器，其位置由 ScaledFeedback 指示并且非常不精确。在这种情况下，ScaledFeedback 只能用于粗略估计当前位置。
Output_UP	OUT	Bool	用于打开阀门的数字输出值。（默认值：FALSE） 如果 Config.OutputPerOn = FALSE，则会使用 Output_UP 参数。
Output_DN	OUT	Bool	用于关闭阀门的数字输出值。（默认值：FALSE） 如果 Config.OutputPerOn = FALSE，则会使用 Output_DN 参数。
Output_PER	OUT	Word	模拟量输出值。 如果 Config.OutputPerOn = TRUE，则会使用 Output_PER 参数。
SetpointLimitH	OUT	Bool	设定值上限。（默认值：FALSE） 如果 SetpointLimitH = TRUE，则说明达到设定值的绝对上限 (Setpoint ≥ Config.SetpointUpperLimit)。 注：设定值限制为 (Setpoint ≥ Config.SetpointUpperLimit)。

参数和类型		数据类型	说明
SetpointLimitL	OUT	Bool	<p>设定值下限。（默认值：FALSE）</p> <p>如果 SetpointLimitL = TRUE，则说明达到设定值的绝对下限 (Setpoint \geq Config.SetpointLowerLimit)。</p> <p>注：设定值限制为 (Setpoint \geq Config.SetpointLowerLimit)。</p>
InputWarningH	OUT	Bool	<p>如果 InputWarningH = TRUE，则说明输入值已达到或超出警告上限。（默认值：FALSE）</p>
InputWarningL	OUT	Bool	<p>如果 InputWarningL = TRUE，则说明输入值已达到或超出警告下限。（默认值：FALSE）</p>
State	OUT	Int	<p>PID 控制器的当前操作模式。（默认值：0）</p> <p>可以使用 Mode 输入参数和 ModeActivate: 的上升沿更改工作模式</p> <ul style="list-style-type: none"> • State = 0: 未激活 • State = 1: 预调节 • State = 2: 手动精确调节 • State = 3: 自动模式 • State = 4: 手动模式 • State = 5: 替换输出值逼近 • State = 6: 切换时间测量 • State = 7: 错误监视 • State = 8: 通过错误监视替换输出值逼近 • State = 10: 无停止位信号的手动模式
Error	OUT	Bool	<p>如果 Error = TRUE，则至少存在一个错误消息未决。（默认值：FALSE）</p> <p>注：V1.x PID 中的 Error 参数是包含错误代码的 ErrorBits 字段。它现在是一个布尔标记，说明有错误发生。</p>
ErrorBits	OUT	DWord	<p>PID_3Step 指令 ErrorBits 参数表 (页 222) 定义未决的错误消息。（默认值：DW#16#0000（无错误））。ErrorBits 具有保持性并在 Reset 或 ErrorAck 的上升沿复位。</p> <p>注：在 V1.x 中，ErrorBits 参数定义为 Error 参数并且不存在。</p>

PID_3Step 控制器的操作

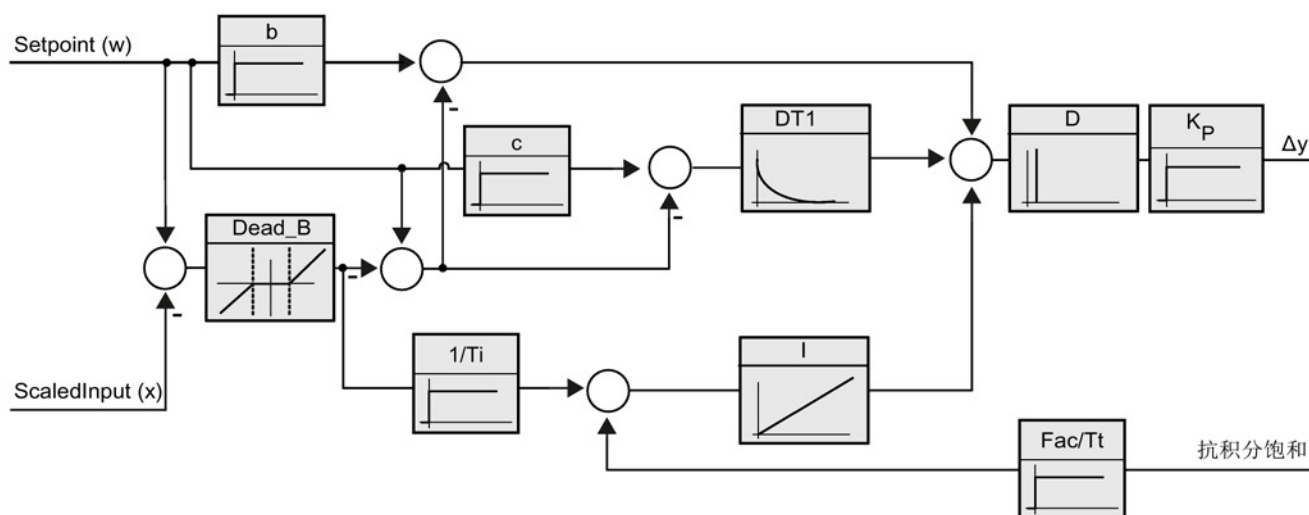


图 8-3 PID_3Step 控制器作为具有抗积分饱和功能的 PID T1 控制器时的操作

8.4 PID_3Step 指令

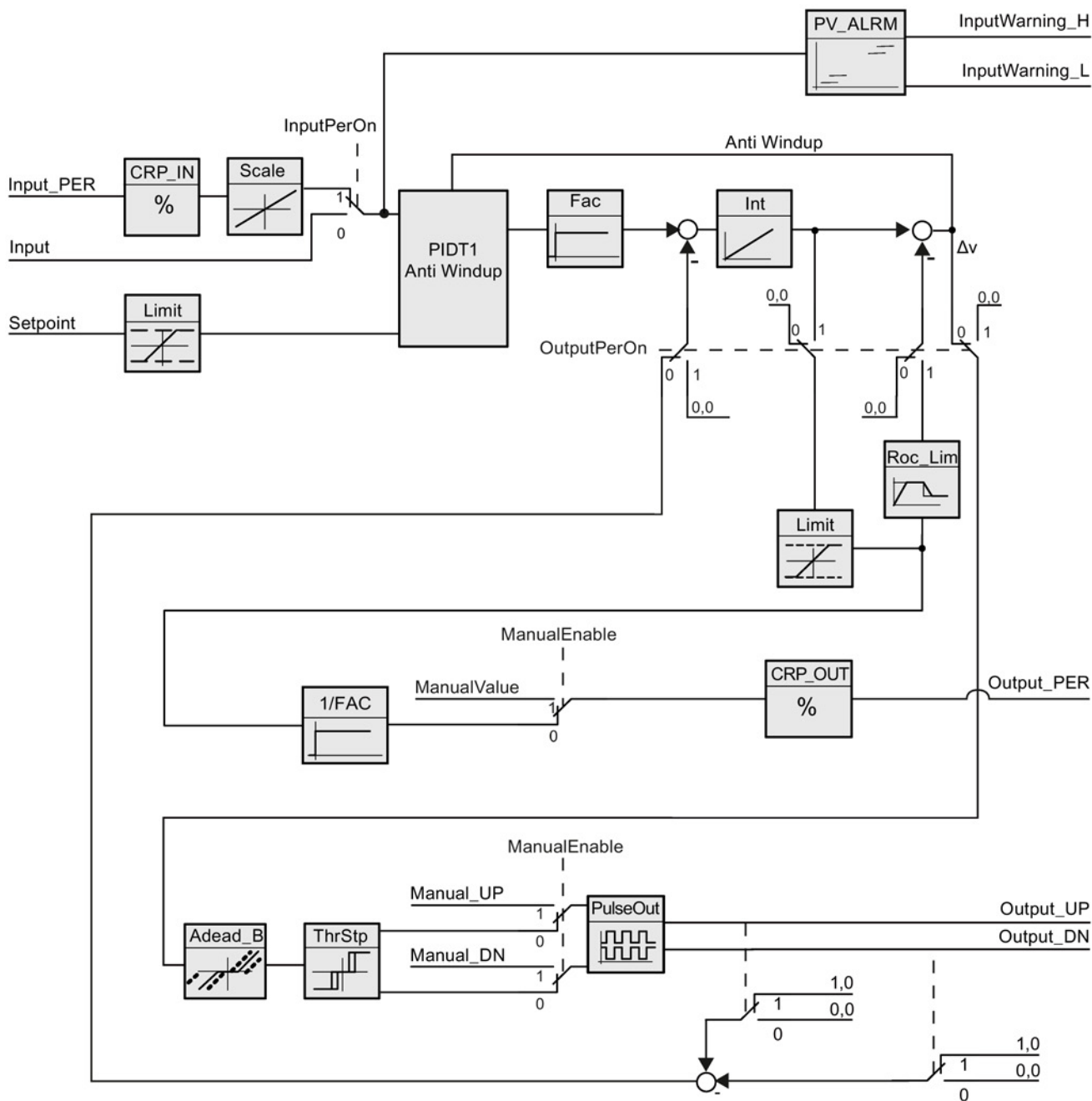


图 8-4 无位置反馈的 PID_3Step 控制器的操作

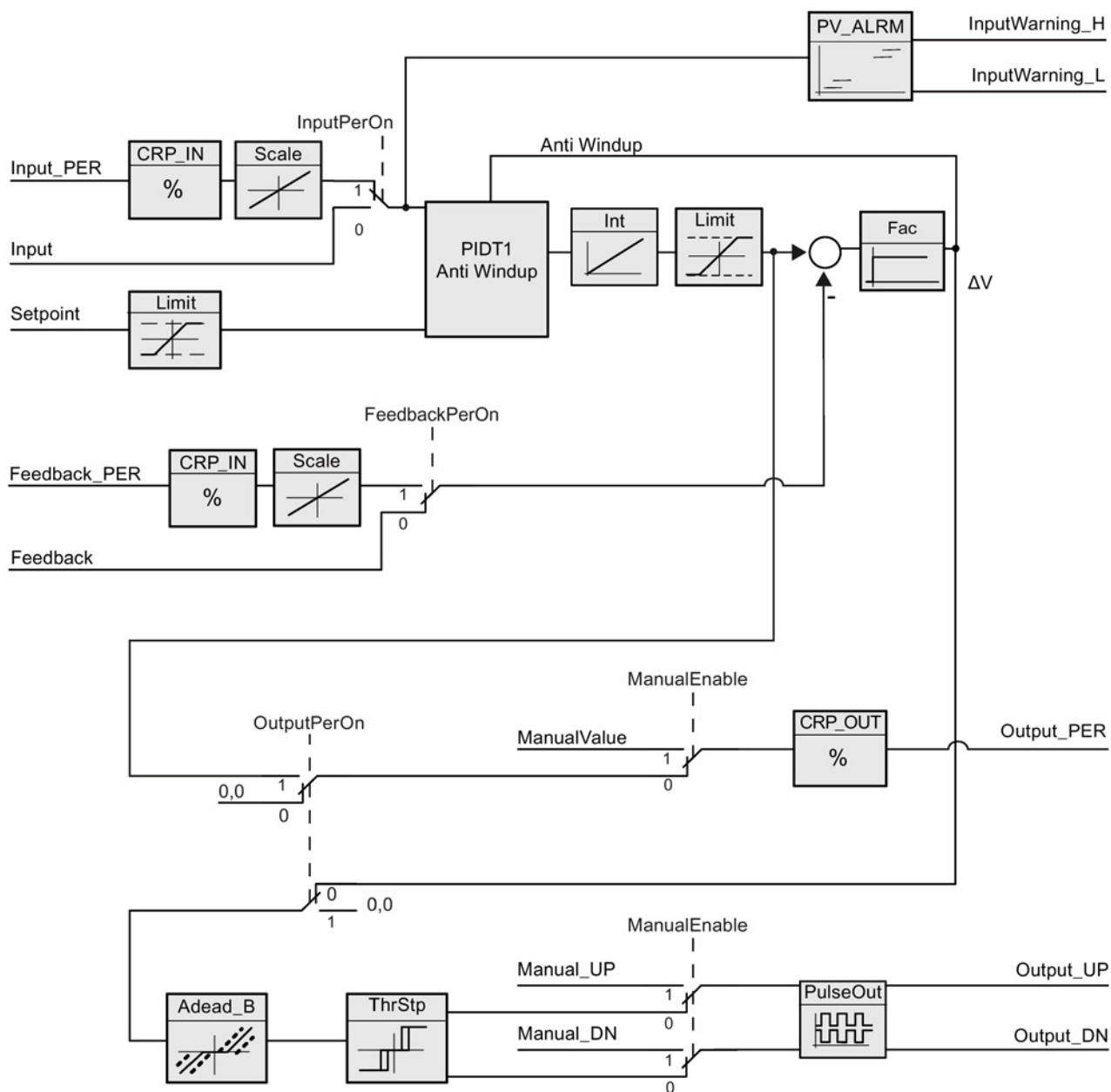


图 8-5 启用了位置反馈的 PID_3Step 控制器的操作

8.5 PID_3Step 指令的 ErrorBit 参数

如果存在多个错误未决，则错误代码的值将通过二进制加法显示。例如，显示错误代码 0003 表示错误 0001 和 0002 未决。

表格 8-8 PID_3STEP 指令的 ErrorBit 参数

ErrorBit (DW#16#...)	说明
0000	无错误
0001 ^{1,2}	参数 Input 超出了过程值限值的范围。 Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2,3}	参数 Input_PER 的值无效。 请检查模拟量输入是否有错误尚未解决。
0004 ⁴	精确调节期间出错。无法保持过程值的振荡。
0010 ⁴	调节期间设定值发生更改。 注意：可在 CancelTuningLevel 变量中设置允许的设定值波动。
0020	精确调节期间不允许预调节。 注意：如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_3Step 保持在精确调节模式。
0080 ⁴	预调节期间出错。输出值限值的组态不正确。 检查是否已正确组态输出值的限值以及该限值是否与控制逻辑匹配。
0100 ⁴	精确调节期间由于无效参数导致出错。
0200 ^{2,3}	参数 Input 的值无效：值的数字格式无效。
0400 ^{2,3}	计算输出值失败。检查 PID 参数。
0800 ^{1,2}	采样时间错误：循环中断 OB 的采样时间内没有调用 PID_3Step。
1000 ^{2,3}	参数 Setpoint 的值无效：值的数字格式无效。
2000 ^{1,2,5}	参数 Feedback_PER 的值无效。 请检查模拟量输入是否有错误尚未解决。
4000 ^{1,2,5}	参数 Feedback 的值无效：值的数字格式无效。

ErrorBit (DW#16#...)	说明
8000 ^{1,2}	<p>数字位置反馈期间出错。Actuator_H = TRUE 和 Actuator_L = TRUE。</p> <p>执行器无法移动到替代输出值，并且将保持在当前位置。 在该状态下不能使用手动模式。</p> <p>为了从此状态移动执行器，必须取消激活“执行器停止位”(Config.ActuatorEndStopOn = FALSE) 或者切换到无停止位信号的手动模式 (Mode = 10)。</p>
10000	<p>参数 ManualValue 的值无效：值的数字格式无效。</p> <p>执行器无法移动到手动值，并且将保持当前位置。</p> <p>在 ManualValue 中分配一个有效值或者在手动模式下通过 Manual_UP 和 Manual_DN 移动执行器。</p>
20000	<p>变量 SavePosition 的值无效：值的数字格式无效。</p> <p>执行器无法移动到替代输出值，并且将保持在当前位置。</p>
40000	<p>参数 Disturbance 的值无效：值的数字格式无效。</p> <p>注意：如果在错误发生前自动模式已激活且 ActivateRecoverMode = FALSE，则 Disturbance 将设置为零。PID_3Step 保持自动模式。</p> <p>注意：如果在错误发生前预调节或精确调节已激活且 ActivateRecoverMode = TRUE，则 PID_3Step 切换到 Mode 参数中保存的工作模式。</p> <p>如果当前阶段中的干扰对输出值无影响，则不会取消调节。 转换时间测量期间错误没有影响。</p>

- 1 注：如果在错误发生前自动模式已激活且 ActivateRecoverMode = TRUE，则 PID_3Step 保持自动模式。
- 2 注：如果在错误发生前已激活预调节、精确调节或转换时间测量模式，并且 ActivateRecoverMode = TRUE，则 PID_3Step 将切换到 Mode 参数中保存的工作模式。
- 3 注：如果在错误发生之前自动模式已激活并且 ActivateRecoverMode = TRUE，则 PID_3Step 将切换到“在监视错误的同时逼近替代输出值”或“错误监视”模式。
当错误不再处于未决状态时，PID_3Step 切换回自动模式。
- 4 注：如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_3Step 取消调节并切换到 Mode 参数中保存的工作模式。
- 5 执行器无法移动到替代输出值，并且将保持当前位置。在手动模式下，仅可通过 Manual_UP 和 Manual_DN 更改执行器的位置，而不可通过 ManualValue 更改。

8.6 PID_Temp 指令

8.6.1 概述

PID_Temp 指令提供一个通用的 PID 控制器，可用于处理温度控制的特定需求。

表格 8-9 PID_Temp 指令

LAD/FBD	SCL	说明
	<pre> "PID_Temp_1" (Setpoint:= _real_in_, Input:= _real_in_, Input_PER:= _int_in_, Disturbance:= _real_in_, ManualEnable:= _bool_in_, ManualValue:= _real_in_, ErrorAck:= _bool_in_, Reset:= _bool_in_, ModeActivate:= _bool_in_, Mode:= _int_in_, Master:= _dword_in Save:= _dword_in ScaledInput=> _real_out_, OutputHeat=> _real_out_, OutputCool=> _real_out_, OutputHeat_PER=> _int_out_, OutputCool_PER=> _int_out_, OutputHeat_PWM=> _bool_out_, OutputCool_PWM=> _bool_out_, SetpointLimit_H=> _bool_out_, SetpointLimit_L=> _bool_out_, InputWarning_H=> _bool_out_, InputWarning_L=> _bool_out_, State=> _int_out_, Error=> _bool_out_, ErrorBits=> _dword_out_); </pre>	<p>PID_Temp 具有以下功能：</p> <ul style="list-style-type: none"> • 使用不同执行器加热或冷却此过程 • 用于处理温度过程的集成式自动调节功能 • 级联处理取决于同一执行器的多个温度

1 STEP 7 会在插入指令时自动创建工艺对象和背景数据块。该背景数据块包含工艺对象的参数。

2 在 SCL 示例中，“PID_Temp_1”是背景 DB 的名称。

表格 8- 10 参数的数据类型

参数和类型		数据类型	说明
Setpoint	IN	Real	PID 控制器在自动模式下的设定值。（默认值：0.0）
Input	IN	Real	用户程序的变量用作过程值的源。（默认值：0.0） 如果正在使用 Input 参数，则必须设置 Config.InputPerOn = FALSE。
Input_PER	IN	Int	模拟量输入用作过程值的源。（默认值：0） 如果正在使用 Input_PER 参数，则必须设置 Config.InputPerOn = TRUE。
Disturbance	IN	Real	干扰变量或预控制值
ManualEnable	IN	Bool	启用或禁用手动操作模式。（默认值：FALSE）： <ul style="list-style-type: none"> FALSE 至 TRUE 沿激活“手动模式”，State = 4，Mode 保持不变。 ManualEnable = TRUE 时，无法利用 ModeActivate 的上升沿或使用调试对话框更改工作模式。 <ul style="list-style-type: none"> TRUE 至 FALSE 沿激活 Mode 分配的工作模式。 注：建议您只使用 ModeActivate 更改工作模式。
ManualValue	IN	Real	手动操作的输出值。（默认值：0.0） 可以使用从 Config.OutputLowerLimit 到 Config.OutputUpperLimit 的值。
ErrorAck	IN	Bool	使用 FALSE 至 TRUE 沿复位 ErrorBits 和警告输出。 （默认值：FALSE）
Reset	IN	Bool	重新启动控制器。（默认值：FALSE）： <ul style="list-style-type: none"> FALSE 至 TRUE 沿： <ul style="list-style-type: none"> 切换到“未激活”模式 复位 ErrorBits 和警告输出 清除积分作用 保持 PID 参数 只要 Reset = TRUE，则 PID_Temp 便会保持在“未激活”模式 (State = 0)。 TRUE 至 FALSE 沿： <ul style="list-style-type: none"> PID_Temp 切换到保存在 Mode 参数中的工作模式。

8.6 PID_Temp 指令

参数和类型		数据类型	说明
ModeActivate	IN	Bool	通过 FALSE 至 TRUE 沿, PID_Temp 切换到保存在 Mode 参数中的操作模式。(默认值: FALSE)
Mode	IN/OUT	Int	<p>在 Mode Activate 输入的上升沿激活。</p> <p>操作模式选择 (默认值: 0.0) :</p> <ul style="list-style-type: none"> • Mode = 0: 未激活 • Mode = 1: 预调节 • Mode = 2: 精确调节 • Mode = 3: 自动模式 • Mode = 4: 手动模式 <p>“带错误监视的替代输出值”(State = 5)。 这无法由用户来激活; 只是一种自动错误响应。</p>
Master	IN/OUT	DWord	到主站的级联连接 (AntiWindUp 和调节条件)。(默认值: DW#16#0000)
Slave	IN/OUT	DWord	<ul style="list-style-type: none"> • 位 0 - 15: 未在 PID_Temp 指令中使用 • 位 16 - 23: 限值计数器: 如果此值达到了限制, 从站会递增该值。 针对抗积分饱和和功能处理的界限内的从站数 (参见 Config.Cascade.AntiWindUpMode 参数)。 • 位 24: IsAutomatic: 如果此控制器的所有从站均处于自动模式且被处理为用于检查级联中的调节条件, 则此位设置为“1”。此位与 AllSlaveAutomaticState 参数的作用相同。 • 位 25: “IsReplacement 设定值”: 如果此控制器一个从站的“替换设定值”已激活且被处理为用于检查级联中的调节条件, 则此位设置为“1”。 反向值存储在 NoSlaveReplacementSetpoint 参数中。
ScaledInput	OUT	Real	标定的过程值。(默认值: 0.0)
OutputHeat ¹	OUT	Real	<p>REAL 格式的加热输出值。(默认值: 0.0)</p> <p>此输出值采用 Config.Output.Heat.Select 参数进行计算, 与输出选择无关。</p>
OutputCool ¹	OUT	Real	<p>REAL 格式的冷却输出值。(默认值: 0.0)</p> <p>此输出值采用 Config.Output.Cool.Select 参数进行计算, 与输出选择无关。</p>

参数和类型		数据类型	说明
OutputHeat_PER ¹	OUT	Int	<p>外设值格式的加热输出值（默认值： 0）</p> <p>只有使用 <code>Config.Output.Heat.Select = 2</code> 参数选择此输出值时，才会计算此输出值。如未选择，此输出始终为“0”。</p>
OutputCool_PER ¹	OUT	Int	<p>外设值格式的冷却输出值（默认值： 0）</p> <p>只有使用 <code>Config.Output.Cool.Select = 2</code> 参数选择此输出值时，才会计算此输出值。如未选择，此输出始终为“0”。</p>
OutputHeat_PWM ¹	OUT	Bool	<p>加热过程的脉宽调制输出值。（默认值： FALSE）</p> <p>只有使用 <code>Config.Output.Heat.Select = 1</code>（默认值）参数选择此输出值时，才会计算此输出值。如未选择，此输出始终为 FALSE。</p>
OutputCool_PWM ¹	OUT	Bool	<p>冷却过程的脉宽调制输出值。（默认值： FALSE）</p> <p>只有使用 <code>Config.Output.Cool.Select = 1</code>（默认值）参数选择此输出值时，才会计算此输出值。如未选择，此输出始终为 FALSE。</p>
SetpointLimit_H	OUT	Bool	<p>设定值上限。（默认值： FALSE）</p> <p>如果 <code>SetpointLimit_H = TRUE</code>，则说明达到设定值的绝对上限 (<code>Setpoint ≥ Config.SetpointUpperLimit</code>)。设定值限制为 <code>Config.SetpointUpperLimit</code>。</p>
SetpointLimit_L	OUT	Bool	<p>设定值下限。（默认值： FALSE）</p> <p>如果 <code>SetpointLimit_L = TRUE</code>，则说明达到设定值的绝对下限 (<code>Setpoint ≤ Config.SetpointLowerLimit</code>)。设定值限制为 <code>Config.SetpointLowerLimit</code>。</p>
InputWarning_H	OUT	Bool	<p>如果 <code>InputWarning_H = TRUE</code>，则说明过程值已达到或超出警告上限。（默认值： FALSE）</p>
InputWarning_L	OUT	Bool	<p>如果 <code>InputWarning_L = TRUE</code>，则说明过程值已达到或低于警告下限。（默认值： FALSE）</p>

8.6 PID_Temp 指令

参数和类型		数据类型	说明
State	OUT	Int	<p>PID 控制器的当前操作模式。（默认值： 0）</p> <p>可以使用 Mode 输入参数和 ModeActivate 的上升沿更改工作模式：</p> <ul style="list-style-type: none"> • State = 0: 未激活 • State = 1: 预调节 • State = 2: 精确调节 • State = 3: 自动模式 • State = 4: 手动模式 • State = 5: 通过错误监视替换输出值
Error	OUT	Bool	<p>如果 Error = TRUE，则该周期内至少有一条错误消息未决。（默认值： FALSE）</p> <p>注： V1.x PID 中的 Error 参数是包含错误代码的 ErrorBits 字段。它现在是一个布尔标记，说明有错误发生。</p>
ErrorBits	OUT	DWord	<p>PID_Temp 指令， ErrorBits 参数表 (页 236)定义未决的错误消息。（默认值： DW#16#0000（无错误））。ErrorBits 具有保持性并在 Reset 或 ErrorAck 的上升沿复位。</p> <p>注： 在 V1.x 中， ErrorBits 参数定义为 Error 参数并且不存在。</p>
Warning	OUT	DWord	<p>PID_Temp 指令， Warning 参数表定义未决的用户相关警告消息。（默认值： DW#16#0000（无警告））。</p>
WarningInternal	OUT	DWord	<p>PID_Temp 指令， WarningInternal 参数表定义未决的内部警告消息（包括所有警告）。（默认值： DW#16#0000（无内部警告））。</p>

¹ 您可以并行使用 Output、Output_PER 和 Output_PWM 参数的输出。

8.6.2 PID_Temp 控制器的操作

选择加热和/或冷却控件

用户必须首先选择除参数“ActivateCooling”中的加热输出外，是否还需要冷却设备。然后必须定义是要在参数“AdvancedCooling”中使用两个 PID 参数集（高级模式）还是仅使用一个 PID 参数集和一个额外的加热/冷却系数。

使用 CoolFactor

如果希望应用加热/冷却系数，必须手动定义该值。必须根据应用程序中的技术数据（执行器的比例增益比率（例如执行器的最大加热和冷却功率的比率））确定该值，并将其分配给参数“CoolFactor”。加热/冷却系数 2.0 表示加热设备的影响力是冷却设备的两倍。如果使用冷却系数，PID_Temp 将计算输出信号，并根据其符号，将输出信号乘以加热/冷却系数（当符号为负时）或不乘以加热/冷却系数（符号为正时）。

使用两个 PID 参数集

在调试期间，可以自动检测用于加热和冷却的不同 PID 参数集。与使用加热/冷却系数相比，这样可以提高控制性能，因为除不同的比例增益外，还可以考虑两个参数集的不同延时时间。但缺点是这要花费更多时间来进行调节。如果激活了 PID 参数切换 (Config.AdvancedCooling = TRUE)，PID_Temp 控制器将以“自动模式”检测（控制已激活），如果这时需要加热或冷却，将使用 PID 参数集进行控制。

ControlZone

使用 PID_Temp 控制器，可以在参数“ControlZone”中为每个参数集定义一个控制区。如果控制偏差（设定值 - 输入）在控制区内，PID_Temp 将使用 PID 算法来计算输出信号。

但如果控制偏差超出了定义的范围，输出将设置为最大加热或最大冷却输出值（冷却输出被激活）/最大加热输出值（冷却输出被禁用）。

用户可以使用此功能更快地达到所需的设定值，特别是对于温度变化较慢的初始加热过程。

DeadZone

通过“DeadZone”参数，可以定义 PID 算法忽略的加热和冷却控制偏差的宽度。

这意味着此范围内的控制偏差将被抑制，PID_Temp

控制器将类似于设定值，并且过程值相同。

因此，可以减少控制器对设定值的不必要干预，并节约执行器。如果要应用 DeadZone，则必须手动定义该值。自动调节功能不会自动设置 DeadZone 值。

对于不制冷的加热控制器或使用 CoolFactor 的加热/冷却控制器，DeadZone 是对称的（在 -Retain.CtrlParams.Heat.DeadZone 和

+Retain.CtrlParams.Heat.DeadZone 之间）。对于使用两个 PID

参数集的加热/冷却控制器，DeadZone 可以是对称的（在 -

Retain.CtrlParams.Cool.DeadZone 和 +Retain.CtrlParams.Heat.DeadZone 之间）。

PID_Temp 控制器操作

以下方框图说明了 PID_Temp 指令的标准和级联操作：

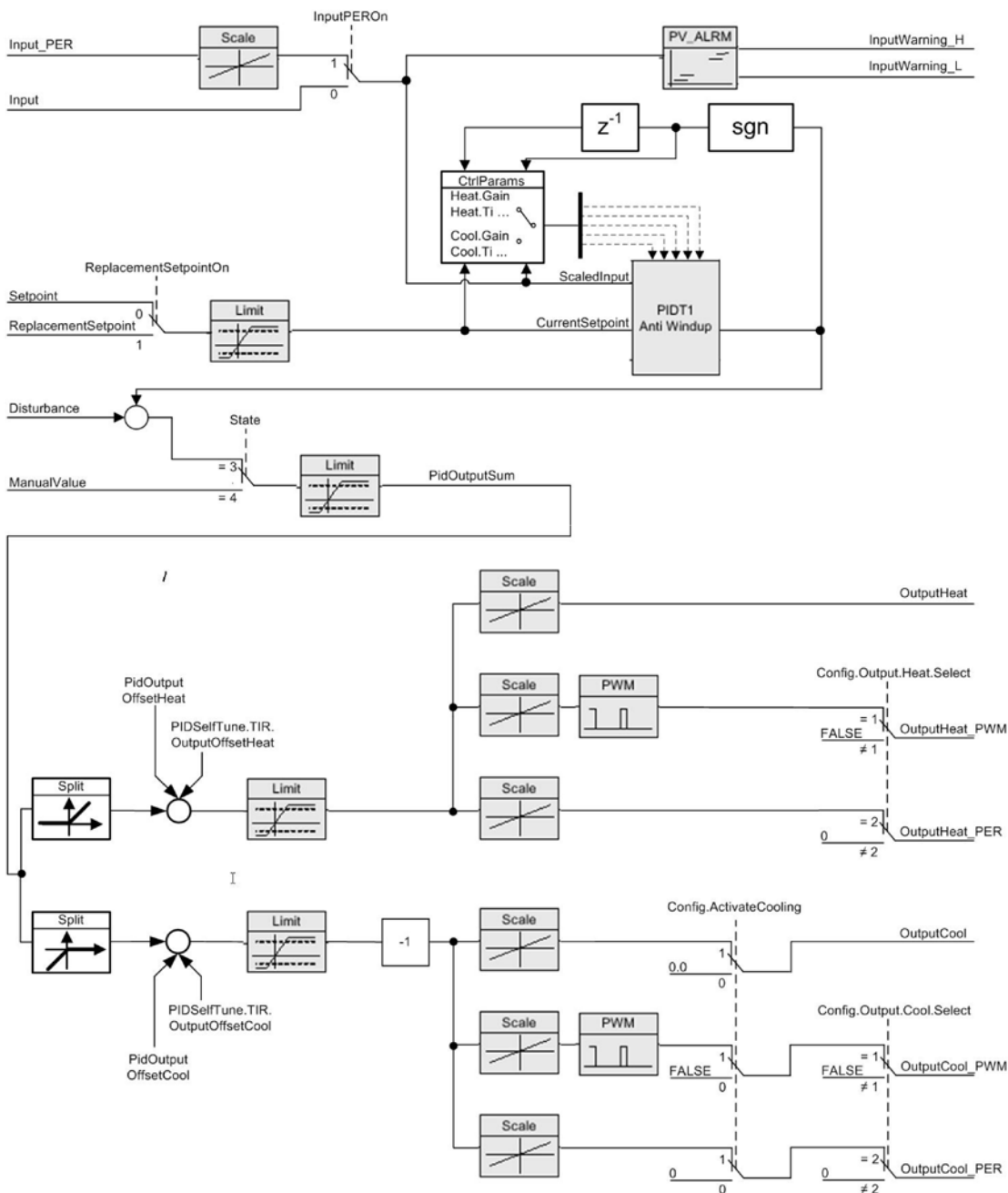


图 8-6 PID_Temp_Operation_Block_Diagram

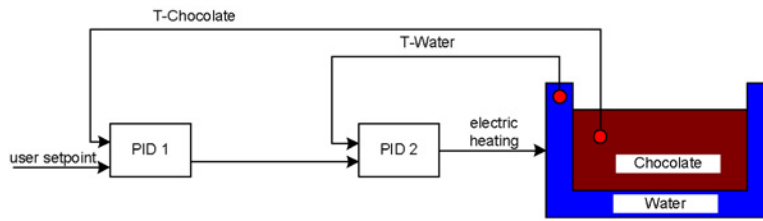


图 8-7 PID_Temp_Cascade_Operation_Block_Diagram

8.6.3 级联控制器

可以级联温度 PID 控制器来处理多个依赖相同执行器的温度。

调用顺序

必须在同一个 OB 周期内调用级联的 PID 控制器。

首先必须调用主站，然后调用控制信号流中的下一个从站，最后调用级联中的最后一个从站。PID_Temp 指令不自动检查调用顺序。

通信连接

在级联控制器时，必须连接主站和从站，使其能够互相共享信息。
必须沿信号流方向将从站的“Master”IN/OUT 参数与其主站的“Slave”IN/OUT 参数相连接。

这样会在级联中显示一个 PID_Temp 控制器连接，并且此级联具有两个子级联：
“PID_Temp1”提供设定值。

组态将“PID_Temp2”、“PID_Temp3”、“PID_Temp5”、“PID_Temp6”和“PID_Temp8”的输出连接到过程：

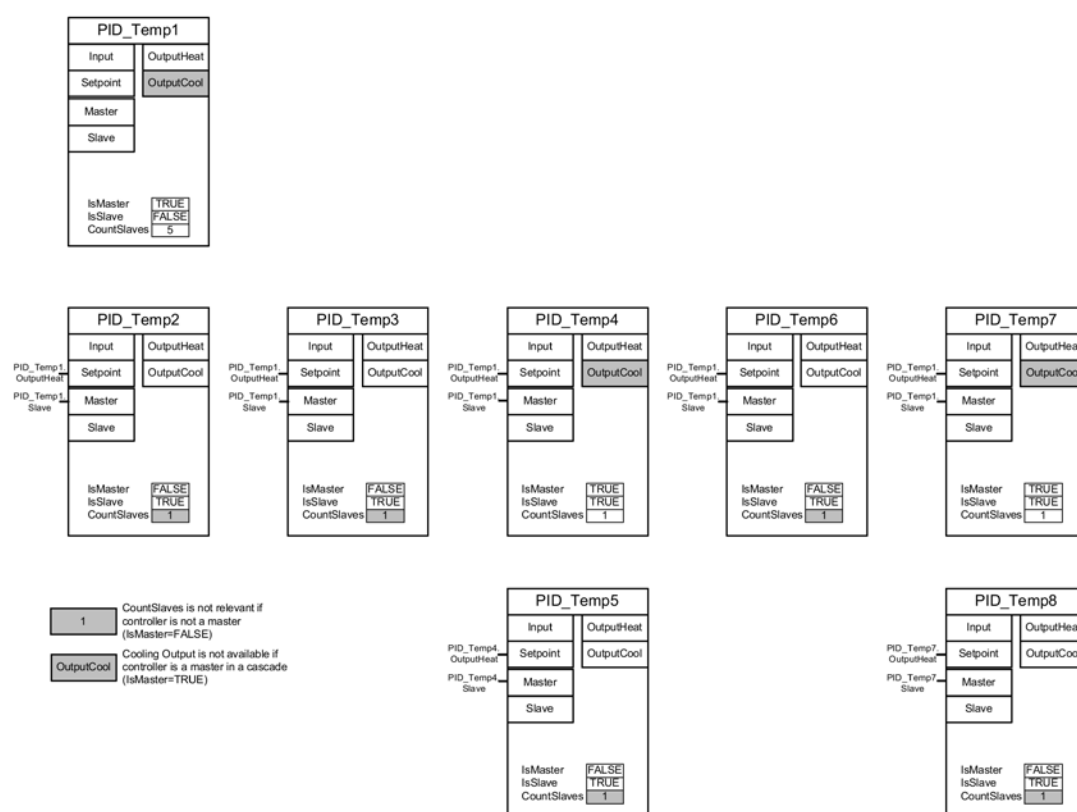


图 8-8 PID_Temp_Cascading_communication_connection

替换设定值

PID_Temp

指令在“ReplacementSetpoint”参数中提供另一个设定值输入，用户可以通过设置参数“ReplacementSetpointOn” = TRUE 将此参数激活。

在调试或调节从站控制器期间，可以使用“ReplacementSetpoint”作为设定值输入，而不必断开主站与从站之间的输出-到-设定值连接。此连接对于级联的正常运行非常重要。

通过这种方式，不必更改程序并下载即可将从站与其主站暂时分开。

用户只需激活“ReplacementSetpoint”并在完成后将其重新禁用。如果可以在“CurrentSetpoint”参数中看到设定值，则该值对 PID 算法生效。

自动调节

级联主站控制器的自动调节必须符合以下要求：

- 从其内部从站到第一个主站都已进行过调试。
- 主站的所有从站必须处于“自动模式”。
- 主站的输出必须是从站的设定值。

PID_Temp 指令将为级联中的自动调节提供以下支持：

- 如果开始自动调节主站控制器，主站将检查是否所有从站均处于“自动模式”，并检查是否为所有从站禁用了“替换设定值”功能（“ReplacementSetpointOn” = FALSE）。如果不符合这些条件，则无法自动调节主站。主站取消调节，进入“未激活”模式（如果“ActivateRecoverMode” = FALSE），或者恢复到“Mode”参数中存储的模式（如果“ActivateRecoverMode” = TRUE）。主站显示错误消息 20000hex（“级联中的主站发生错误。从站不处于自动模式或者启用了替换设定值，阻止了主站的调节。”）。
- 当所有从站均处于“自动模式”时，系统将设置参数“AllSlaveAutomaticState” = TRUE。用户可以在自己的程序中应用此参数，或者本地化错误 20000hex 的原因。
- 当为所有从站禁用“ReplacementSetpoint”时，系统将设置参数“NoSlaveReplacementSetpoint” = TRUE。用户可以在自己的程序中应用此参数，或者本地化错误 20000hex 的原因。

在使用 PID_Temp 指令调试对话框时，用户可以得到进一步的级联调节支持 (页 262)。

操作模式与错误处理

PID_Temp 控制器不允许其主站或从站切换操作模式。

这意味着当从站发生错误时，级联内的主站仍保持其当前模式。

如果有两个或更多并行从站使用此主站控制器运行，这就是一个优点，一条链中发生错误不会导致并行链关闭。

同样，如果主站发生错误，级联内的从站仍保持其当前操作模式。

但从站的进一步操作取决于主站的组态，因为从站的设定值是主站的输出。

这意味着如果使用 `"ActivateRecoverMode" = TRUE`

组态主站并且发生错误，主站将输出上一个有效值或一个替换输出值作为从站的设定值。

如果使用 `"ActivateRecoverMode" = FALSE`

组态主站，主站将切换到“未激活模式”并将所有输出设置为“0.0”，让从站使用“0.0”作为其设定值。

因为只有从站控制器能直接访问执行器，并且在主站发生错误时从站仍保持其操作模式，所以能够避免过程受损。

例如，对于塑料加工设备，从站停止工作，关闭执行器，并允许塑料在设备内部单独硬化是非常致命的错误，因为主站控制器有错误。

抗积分饱和

级联中的从站从其主站的输出获取其设定值。

如果从站达到自己的输出限制，同时主站仍看到控制偏差（设定值 - 输入），主站将冻结或减少其积分贡献以防止所谓的“积分饱和”。

在发生“积分饱和”时，主站将其积分贡献增加到了一个非常大的值，要使控制器能够重新正常响应，必须首先降低该值。此类“积分饱和”会对控制的动态性产生负面影响。

PID_Temp

通过组态主站控制器的参数 `"Config.Cascade.AntiWindUpMode"`，提供了在级联中防止这种影响的方法：

值	说明
0	禁用抗积分饱和功能。
1	将主站控制器的积分贡献比率从“界限内的从站数”减少到“现有从站数”（参数 <code>"CountSlaves"</code> ）。
2	在一个从站到达其限制后，立即冻结主站的积分贡献。仅当 <code>"Config.Cascade.IsMaster" = TRUE</code> 时适用。

8.7 PID_Temp 指令 ErrorBit 参数

如果 PID 控制器存在多个未决警告，则错误代码的值将通过二进制加法显示。
例如，显示错误代码 0003 表示错误 0001 和 0002 未决。

表格 8- 11 PID_Temp 指令 ErrorBit 参数

ErrorBit (DW#16#...)	说明
0000	无错误
0001 ^{1,2}	参数 Input 超出了过程值限值的范围。 Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2,3}	参数 Input_PER 的值无效。 请检查模拟量输入是否有错误尚未解决。
0004 ⁴	精确调节期间出错。无法保持过程值的振荡。
0008 ⁴	预调节开始时出错。过程值过于接近设定值。开始精确调节。
0010 ⁴	调节期间设定值发生改变。 注：可在 CancelTuningLevel 变量中设置允许的设定值波动。
0020	精确调节期间不允许预调节。 注：如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 保持在精确调节模式。
0040 ⁴	预调节期间出错。冷却不会减少过程值。
0080 ⁴	预调节期间出错。输出值限值的组态不正确。 检查是否已正确组态输出值的限值以及该限值是否与控制逻辑匹配。
0100 ⁴	精确调节期间由于无效参数导致出错。
0200 ^{2,3}	参数 Input 的值无效：值的数字格式无效。
0400 ^{2,3}	输出值计算失败。检查 PID 参数。
0800 ^{1,2}	采样时间错误：循环中断 OB 的采样时间内没有调用 PID_Temp。
1000 ^{2,3}	参数 Setpoint 的值无效：值的数字格式无效。

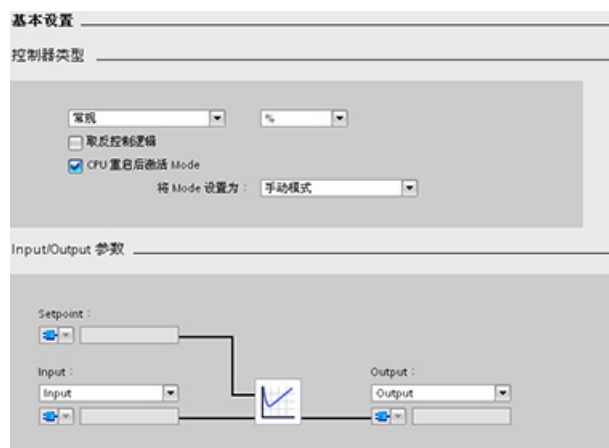
ErrorBit (DW#16#...)	说明
10000	<p>参数 ManualValue 的值无效：值的数字格式无效。</p> <p>注：如果在错误发生前 ActivateRecoverMode = TRUE，则 PID_Temp 使用 SubstituteOutput 作为输出值。只要在 ManualValue 参数中分配有效值，PID_Temp 便会将其用作输出值。</p>
20000	<p>变量 SubstituteValue 的值无效：值的数字格式无效。</p> <p>PID_Temp 使用输出值下限作为输出值。</p> <p>注： 如果在错误发生之前自动模式已激活，ActivateRecoverMode = TRUE 且错误不再处于未决状态，则 PID_Temp 切换回自动模式。</p>
40000	<p>参数 Disturbance 的值无效：值的数字格式无效。</p> <p>注：如果在错误发生前自动模式已激活且 ActivateRecoverMode = FALSE，则 Disturbance 将设置为零。PID_Temp 保持自动模式。</p> <p>注：如果在错误发生前预调节或精确调节已激活且 ActivateRecoverMode = TRUE，则 PID_Temp 切换到 Mode 参数中保存的工作模式。</p> <p>如果当前阶段中的干扰对输出值无影响，则不会取消调节。</p>
200000	<p>级联中的主站发生错误。</p> <p>从站不处于自动模式或者启用了替换设定值，阻止了主站的调节。</p>
400000	<p>当冷却正在进行时，PID 控制器不允许执行加热预调节。</p>
800000	<p>要开始进行冷却预调节，过程值必须接近设定值。</p>
1000000	<p>开始调节时出错。</p> <p>“Heat.EnableTuning”和“Cool.EnableTuning”未设置或与组态不匹配。</p>
2000000	<p>冷却预调节要求成功的加热预调节。</p>

ErrorBit (DW#16#...)	说明
4000000	开始精确调节时出错。 “Heat.EnableTuning”和“Cool.EnableTuning”不能同时设置。
8000000	PID 参数计算期间发生的错误导致参数无效（例如，负增益；当前 PID 参数保持不变并且调节无影响）。

- 1 注：如果在错误发生前自动模式已激活且 `ActivateRecoverMode = TRUE`，则 `PID_Temp` 保持自动模式。
- 2 注：如果在错误发生前预调节或精确调节已激活且 `ActivateRecoverMode = TRUE`，则 `PID_Temp` 切换到 `Mode` 参数中保存的工作模式。
- 3 注：如果在错误发生前自动模式已激活且 `ActivateRecoverMode = TRUE`，则 `PID_Compact` 输出组态的替换输出值。当错误不再处于未决状态时，`PID_Temp` 切换回自动模式。
- 4 注：如果在错误发生前 `ActivateRecoverMode = TRUE`，则 `PID_Temp` 取消调节并切换到 `Mode` 参数中保存的工作模式。

8.8 组态 PID_Compact 和 PID_3Step 控制器

工艺对象的参数可决定 PID 控制器的操作。使用该图标可打开组态编辑器。

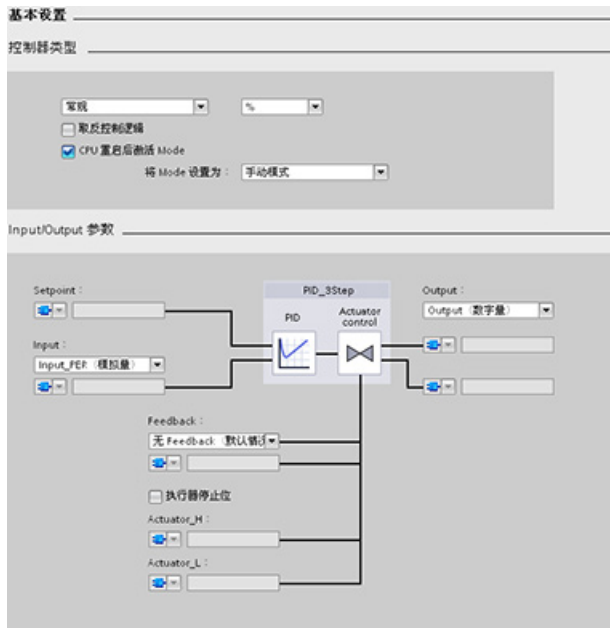


表格 8-12 PID_Compact 指令的组态设置示例

设置	说明
基础	控制器类型
	选择工程单元。
反转控制逻辑 (Invert the control logic)	<p>允许选择反作用 PID 回路。</p> <ul style="list-style-type: none"> 如果未选择该选项，则 PID 回路处于直接作用模式，在输入值小于设定值时，PID 回路的输出会增大。 如果选择了该选项，则在输入值大于设定值时，PID 回路的输出会增大。
CPU 重启后启用上一模式 (Enable last mode after CPU restart)	在复位 PID 回路之后，或在超出输入限值后回到有效范围时，重新启动 PID 回路。
输入 (Input)	为过程值选择 Input 参数或 Input_PER 参数（用于模拟量）。Input_PER 可直接来自模拟量输入模块。
输出	为输出值选择 Output 参数或 Output_PER 参数（用于模拟量）。Output_PER 可直接进入模拟量输出模块。

8.8 组态 PID_Compact 和 PID_3Step 控制器

设置	说明
过程值	<p>标定过程值的范围和限值。如果过程值低于下限或高出上限，则 PID 回路进入未激活模式，并将输出值设置为 0。</p> <p>要使用 Input_PER，必须标定模拟过程值（输入值）。</p>



表格 8-13 PID_3Step 指令的组态设置示例

设置	说明	
基础	控制器类型	选择工程单元。
	反转控制逻辑	<p>允许选择反作用 PID 回路。</p> <ul style="list-style-type: none"> 如果未选择该选项，则 PID 回路处于直接作用模式，在输入值小于设定值时，PID 回路的输出会增大。 如果选择了该选项，则在输入值大于设定值时，PID 回路的输出会增大。
	CPU 重启后激活模式	<p>在复位 PID 回路之后，或在超出输入限值后回到有效范围时，重新启动 PID 回路。</p> <p>将模式设置为：定义重新启动后用户想要 PID 跳转到的模式。</p>
	Input	为过程值选择 Input 参数或 Input_PER 参数（用于模拟量）。Input_PER 可直接来自模拟量输入模块。

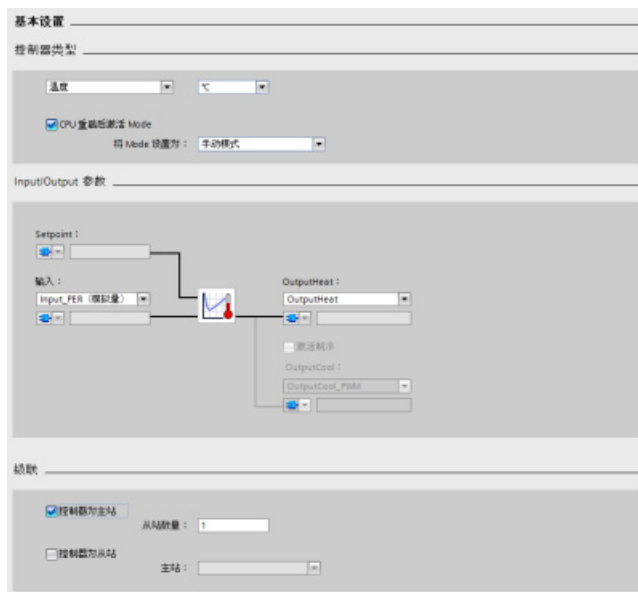
设置		说明
	输出	选择为输出值使用数字量输出 (Output_UP 和 Output_DN) 或使用模拟量输出 (Output_PER)。
	反馈	选择返回到 PID 回路的设备状态的类型： <ul style="list-style-type: none"> • 无反馈 (默认) • 反馈 • Feedback_PER
过程值	<p>标定过程值的范围和限值。如果过程值低于下限或高出上限，则 PID 回路进入未激活模式，并将输出值设置为 0。</p> <p>要使用 Input_PER，必须标定模拟过程值 (输入值)。</p>	
执行器	电机切换时间 (Motor transition time)	设置阀门从打开到关闭的时间。(可在阀门的数据表或面板上找到该值。)
	最短打开时间 (Minimum ON time)	设置阀门的最短运动时间。(可在阀门的数据表或面板上找到该值。)
	最短关闭时间 (Minimum OFF time)	设置阀门的最短暂停时间。(可在阀门的数据表或面板上找到该值。)
	对错误的响应	<p>定义检查到错误或复位 PID 回路时的阀门行为。</p> <p>如果选择使用替换位置，请输入“安全位置”(Safety position)。</p> <p>对于模拟反馈或模拟输出，应为输出选择在上限和下限之间的值。</p> <p>对于数字输出，只能选择 0% (关闭) 或 100% (打开)。</p>
	刻度位置反馈 ¹ (Scale Position Feedback)	<ul style="list-style-type: none"> • “上端停止位”(High end stop) 和“下端停止位”(Lower end stop) 定义最大正向位置 (完全打开) 和最大反向位置 (完全关闭)。“上端停止位”(High end stop) 必须大于“下端停止位”(Lower end stop)。 • “过程值上限”(High limit process value) 和“过程值下限”(Low limit process value) 定义调节模式和自动模式中阀门的上限位置和下限位置。 • “FeedbackPER” (“下限”和“上限”) 定义阀门位置的模拟反馈。“FeedbackPER 上限”必须大于“FeedbackPER 下限”。
高级	监视过程值	为过程值设置警告上限和下限。
	PID 参数	如果用户需要，可以在此窗口中输入自己的 PID 调节参数。必须选中“启用手动输入”(Enable Manual Entry) 复选框来执行此操作。

设置	说明
----	----

- 1 只有在“基本”(Basic) 设置中启用了“反馈”(Feedback) 时，才能编辑“刻度位置反馈”(Scale Position Feedback)。

8.9 组态 PID_Temp 控制器

工艺对象的参数可决定 PID 控制器的操作。使用该图标可打开组态编辑器。



表格 8-14 PID_Temp 指令的组态设置示例

设置		说明
基础	控制器类型	选择工程单元。
	CPU 重启后激活模式	在复位 PID 回路之后，或在超出输入限值后回到有效范围时，重新启动 PID 回路。 将模式设置为： 定义重新启动后用户想要 PID 跳转到的模式。
	Input	为过程值选择 Input 参数或 Input_PER 参数（用于模拟量）。Input_PER 可直接来自模拟量输入模块。

设置		说明
	OutputHeat	选择为输出值使用数字量输出 (OutputHeat 和 OutputHeat_PWM) 或使用模拟量输出 (OutputHeat_PER (analog))。
	OutputCool	选择为输出值使用数字量输出 (OutputCool 和 OutputCool_PWM) 或使用模拟量输出 (OutputCool_PER (analog))。
过程值	标定过程值的范围和限值。如果过程值低于下限或高出上限, 则 PID 回路进入未激活模式, 并将输出值设置为 0。 要使用 Input_PER, 必须标定模拟过程值 (输入值)。	
级联	控制器为主站	将控制器设置为主站并选择从站数量。
	控制器为从站	将控制器设置为从站并选择主站数量。

控制器类型

设置	TO-DB 参数	数据类型	取值范围	说明
实际数量	"PhysicalQuantity"	Int (Enum)	<ul style="list-style-type: none"> • 常规 • 温度 (=默认值) 	预选择实际单位值 无多值控制, 并且在功能视图的在线模式下无法编辑。
计量单位	"PhysicalUnit"	Int (Enum)	<ul style="list-style-type: none"> • 常规: 单位 = % • 温度: 单位 (可能的选项) = <ul style="list-style-type: none"> - °C (默认值) - °F - K 	如果更改实际数量, 选择的用户单位将设置回"0"。

设置	TO-DB 参数	数据类型	取值范围	说明
CPU 重启后 激活模式	“RunModeByStartup”	Bool	复选框	如果设置为 TRUE（默认值），在重启（电源打开，关闭，再打开）后或 PLC 从 STOP 模式转换到 RUN 模式后，控制器将切换到“Mode”变量中存储的状态”。 否则，PID_Temp 将继续处于“未激活”模式。
将模式 设置为	“模式”	Int (Enum)	模式（可能的选项）： <ul style="list-style-type: none"> • 0: 未激活 • 1: 预调节 • 2: 精确调节 • 3: “自动”模式 • 4: “手动”模式（默认值） 	工程站 (ES) 根据用户的选择设置“Mode”变量的起始值。“Mode”的默认值（存储在 TO-DB 中）为“Manual Mode”。

输入/输出参数

设置	TO-DB 参数	数据类型	取值范围	说明
设定值	设定值	Real)	Real	只能在属性页面中访问 功能视图的在线模式下无多值控制。
选择输入	“Config.InputPerOn”	Bool (Enum)	Bool	选择要使用的输入类型。 可选择： <ul style="list-style-type: none"> • FALSE: “Input” (Real) • TRUE: “Input_PER (analog)”
Input	Input 或 Input_PER	Real 或 Int	Real 或 Int	只能在属性页面中访问。 功能视图的在线模式下无多值控制。

设置	TO-DB 参数	数据类型	取值范围	说明
选择输出（加热）	“Config.Output.Heat.Select”	Int (Enum)	2 >= Config.Output.Heat.Select >= 0	<p>选择要用于加热的输出类型。</p> <p>可选择：</p> <ul style="list-style-type: none"> • “OutputHeat” (Real) • “OutputHeat_PWM” (Bool)（默认值） • “OutputHeat_PER (analog)” (Word) <p>如果用户激活了“级联”(Cascade)部分的“此控制器为主站”(This controller is a master)复选框，则设置为“OutputHeat”一次。</p>
输出（加热）	OutputHeat、OutputHeat_PER 或 OutputHeat_PWM	Real、Int 或 Bool	实型、整型或布尔型	<p>只能在属性页面中访问。</p> <p>功能视图的在线模式下无多值控制。</p>

设置	TO-DB 参数	数据类型	取值范围	说明
激活输出（冷却）	“Config.ActivateCooling”	Bool	Bool	<p>选中此复选框：</p> <ul style="list-style-type: none"> • 设置“Config.Output. Heat.PidLowerLimit = 0.0 一次。 • 设置“Config.ActivateCooling”参数为 TRUE，而不是 FALSE（未选中时的默认值）。 • 激活所有其他“输出（冷却）”控件（在“基本设置”(Basic settings) 和其它视图中）。 • 将从 PID 符号到控件的线从灰色更改为黑色。 • 禁用了“级联”(Cascade) 部分的“此控制器为主站”(This controller is a master) 复选框。 <p>注： 仅当不将控制器组态为级联的主站时才可用（禁用了“级联”(Cascade) 部分的“此控制器为主站”(This controller is a master) 复选框；“Config.Cascade.IsMaster” = FALSE）。</p>

设置	TO-DB 参数	数据类型	取值范围	说明
选择输出（冷却）	“Config.Output.Cool.Select”	Int (Enum)	2 >= Config.Output.Heat.Select >= 0	<p>选择要用于冷却的输出类型。</p> <p>可选择：</p> <ul style="list-style-type: none"> “OutputCool” (Real) “OutputCool_PWM” (Bool)（默认值） “OutputCool_PER (analog)” (Word) <p>仅当选“激活输出（冷却）”(Activate output (cooling)) 时才可用； (Config.ActivateCooling = TRUE)。</p>
输出（冷却）	OutputCool、OutputCool_PER 或 OutputCool_PWM	Real、Int 或 Bool	实型、整型或布尔型	<p>只能在属性页面中访问。</p> <p>功能视图的在线模式下无多值控制。</p>

级联参数

通过以下参数，可选择控制器作为主站或从站，并确定直接从主站控制器接收其设定值的从站控制器的数量：

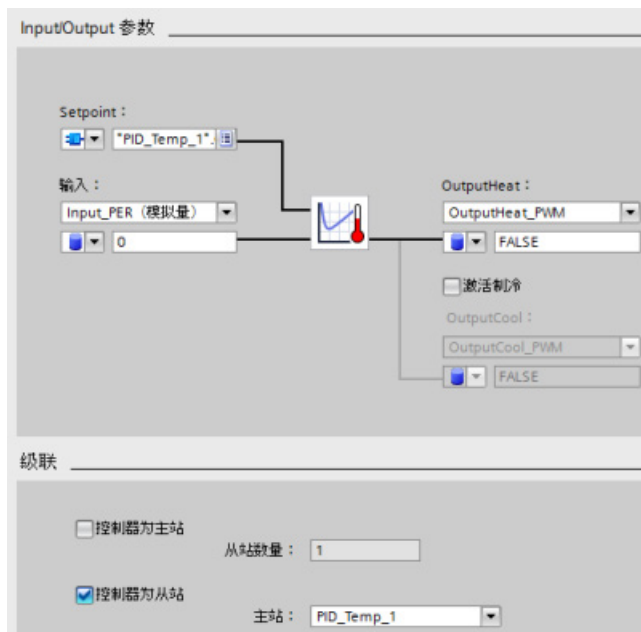
设置	TO-DB 参数	数据类型	取值范围	说明
----	----------	------	------	----

设置	TO-DB 参数	数据类型	取值范围	说明
此控制器为主站	“Config.Cascade.IsMaster”	Bool	Bool	<p>显示此控制器是否为级联中的主站。</p> <p>在选中此复选框时，执行以下操作：</p> <ul style="list-style-type: none"> 将参数“Config.Cascade.IsMaster”设置为 TRUE，而不是 FALSE（未选中时的默认值）。 将“输入/输出参数”(Input / output parameters) 部分的“选择输出（加热）”(Selection Output (heating))”设置为“OutputHeat”一次（Config.Output.Heat.Select = 0）。 启用“从站数目”(Number of Slaves) 输入域。 禁用“输入/输出参数”(Input / output parameters) 部分的“激活输出（冷却）”(Activate output (cooling))。 <p>注： 仅当禁用了此控制器的冷却输出时才可用（禁用“输入/输出参数”(Input / output parameters) 部分的“激活输出（冷却）”(Activate output (cooling)) 复选框时（Config.ActivateCooling = FALSE））。</p>

设置	TO-DB 参数	数据类型	取值范围	说明
从站数目	“Config.Cascade.CountSlaves”	Int	255 >= Config.Cascade. CountSlaves >= 1	<p>直接从该主站控制器获取其设定值的从站控制器的数目。</p> <p>PID_Temp 指令将该值与其它值一起处理以用于抗积分饱和和处理。仅当激活了“此控制器为主站”(This controller is a master) 复选框 (Config.Cascade.IsMaster = TRUE) 时, “从站数目”(Number of slaves) 才可用。</p>
此控制器为从站	“Config.Cascade.IsSlave”	Bool	Bool	<p>显示此控制器是否为级联中的从站。选中此复选框时, 将参数“Config.Cascade.IsSlave”设置为 TRUE, 而不是 FALSE (未选中时的默认值)。</p> <p>必须在属性页面中选中此复选框以启用“SelectionMaster”下拉列表。</p>

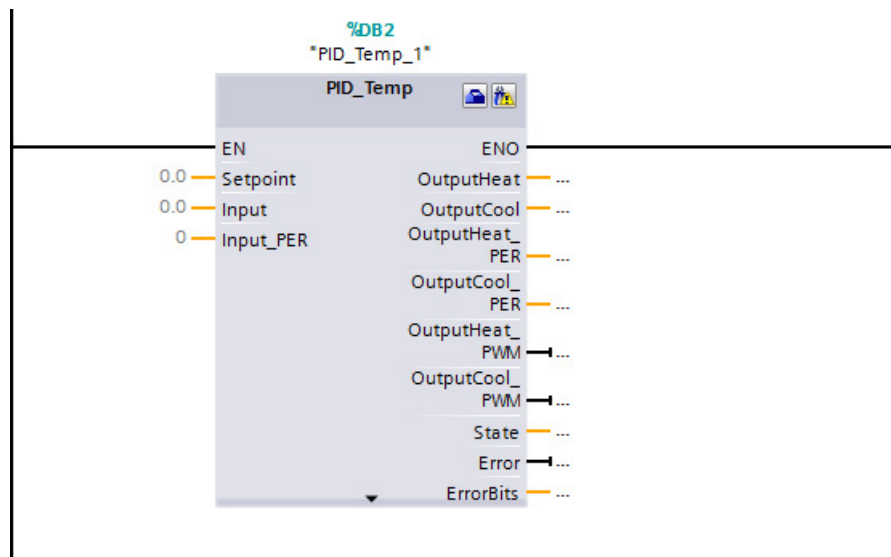
示例：级联控制器

选择“PID_Temp_1”作为主站后，在“基本设置”(Basic settings)对话框下面，可以看到从站控制器“PID_Temp_2”的“输入/输出参数”(Input / output parameters)部分和“级联”(Cascade)部分。主站和从站控制器之间已建立了连接：



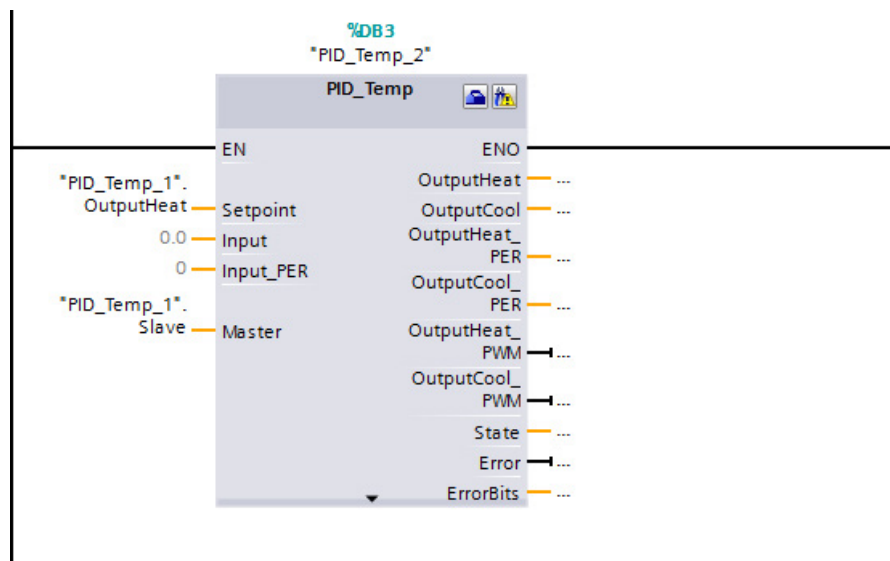
程序段 1:

在这些程序段中，通过编程编辑器在“PID_Temp_1”主站和“PID_Temp_2”从站之间建立了连接：



程序段 2:

已在“PID_Temp_1”主站的“OutputHeat”和“Slave”参数与“PID_Temp_2”从站的“Setpoint”和“Master”参数之间分别建立了连接:



温度自动调节过程

PID_Temp 指令提供两种自动调节模式:

- “预调节” (参数“Mode”= 1)
- “精确调节” (参数“Mode”= 2)

根据控制器组态，提供这些调节方法的不同版本：

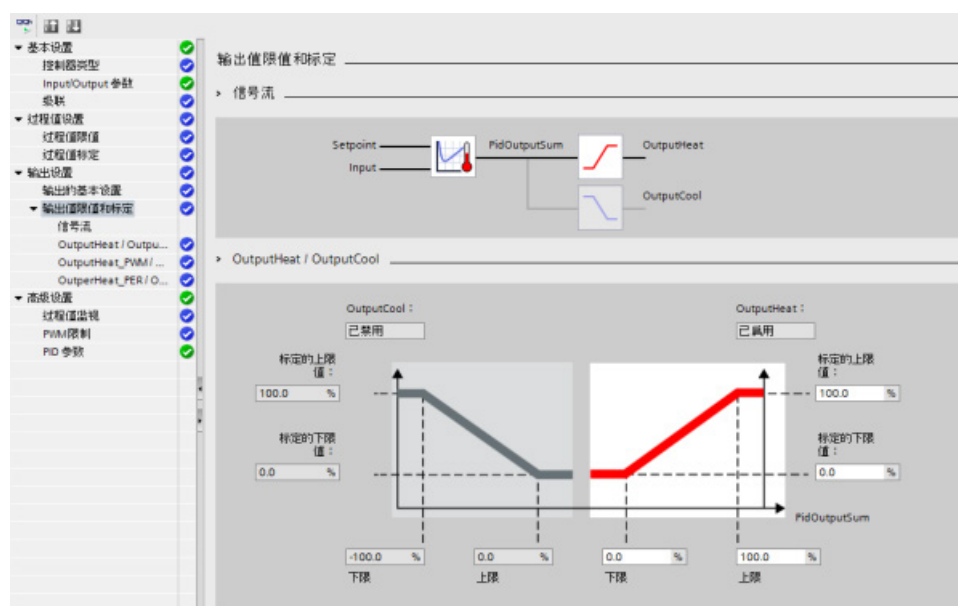
组态	具有加热输出的控制器	具有加热和冷却输出且使用冷却系数的控制器	具有加热和冷却输出且使用两个 PID 参数集的控制器
相关 TO-DB 值	<ul style="list-style-type: none"> • Config.ActivateCooling = FALSE • Config.AdvancedCooling = irrelevant 	<ul style="list-style-type: none"> • Config.ActivateCooling = TRUE • Config.AdvancedCooling = FALSE 	<ul style="list-style-type: none"> • Config.ActivateCooling = TRUE • Config.AdvancedCooling = TRUE
可用调节方法	<ul style="list-style-type: none"> • “预调节加热” • “精确调节加热” (不能使用冷却偏移量) 	<ul style="list-style-type: none"> • “预调节加热” • “精确调节加热” (可以使用冷却偏移量) 	<ul style="list-style-type: none"> • “预调节加热和冷却” • “预调节加热” • “预调节冷却” • “精确调节加热” (可以使用冷却偏移量) • “精确调节冷却” (可以使用加热偏移量)

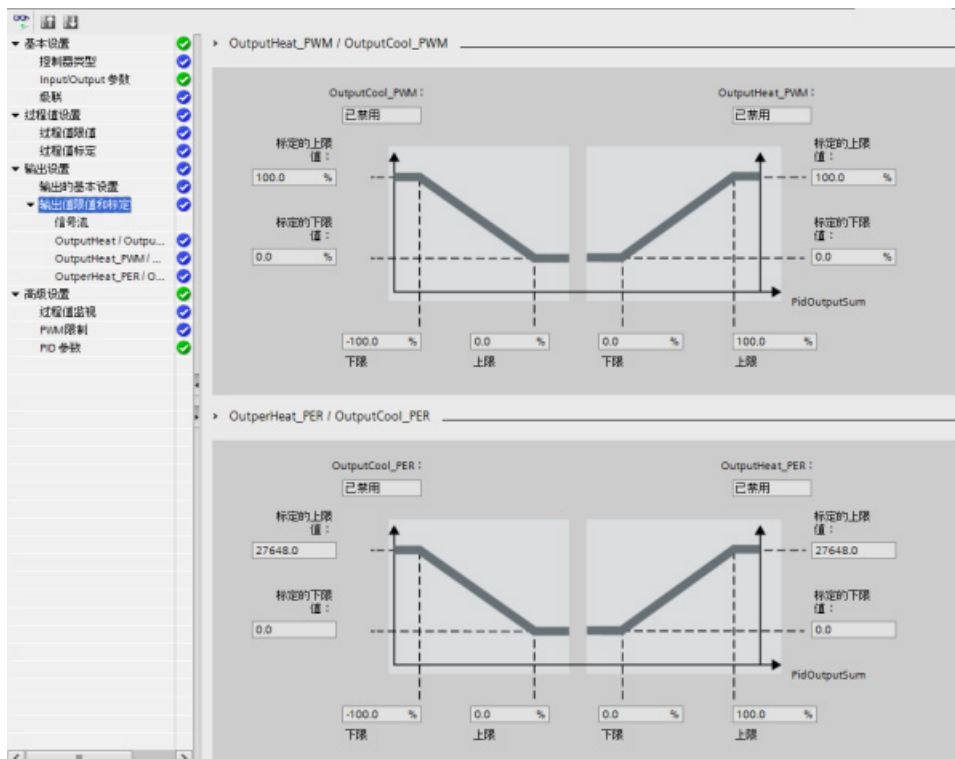
输出值限值和标定

禁用冷却激活

如果组态 PID_Temp 指令作为级联主站，“基本设置”(Basic settings) 视图中的“激活输出（冷却）”(Activate output (cooling)) 复选框将不选中并被禁用，“输出设置”(Output settings) 视图中所有依赖于冷却激活的设置也会禁用。

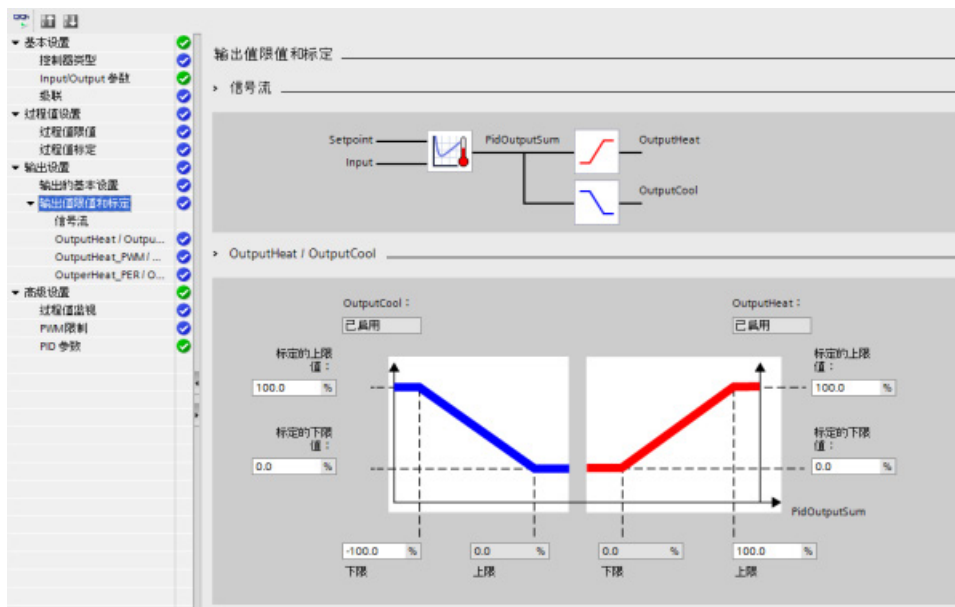
下图显示了禁用冷却时“输出设置”(Output settings) 视图中的“输出值限值和标定”(Output value limits and scaling) 部分（在“输入/输出参数”(Input / output parameters) 视图中选择了 OutputHeat_PWM 并始终启用 OutputHeat）：





启用冷却激活

下图显示了激活冷却时“输出设置”(Output settings) 视图中的“输出值限值和标定”(Output value limits and scaling) 部分（在“输入/输出参数”(Input / output parameters) 视图中选择了 OutputCool_PER 和 OutputHeat_PWM 并始终启用 OutputCool 和 OutputHeat）：





操作模式

要手动更改操作模式，用户需要设置控制器的“Mode”输入-输出参数，并通过将“ModeActivate”从 FALSE 更改为 TRUE 将其激活（触发上升沿）。

在下次模式更改之前，必须复位“ModeActivate”，它不会自动复位。

输出参数“State”显示当前操作模式，并设置为请求的“Mode”（如果可行）。

“State”参数无法直接更改；它只能由控制器通过“Mode”参数或自动操作模式更改进行更改。

8.9 组态 PID_Temp 控制器

“Mode”/“State”	名称	说明
0	未激活	<p>PID_Temp 指令：</p> <ul style="list-style-type: none"> • 禁用 PID 算法和脉宽调制 • 所有控制器输出（OutputHeat、OutputCool、OutputHeat_PWM、OutputCool_PWM、OutputHeat_PER、OutputCool_PER）设置为“0”(FALSE)，而不考虑组态的输出限值或偏移量。可通过设置 “Mode” = 0、“Reset” = TRUE 或通过发生错误进入此模式。
1	预调节（开始调节/SUT）	<p>此模式在第一次启动控制器时确定参数。</p> <p>与 PID_Compact 不同，对于 PID_Temp，必须使用“Heat.EnableTuning”和“Cool.EnableTuning”参数选择是否需要启用加热调节和/或冷却调节。</p> <p>用户可以从“未激活”、“自动”模式或“手动”模式激活“预调节”。</p> <p>如果调节成功，PID_Temp 将切换到“自动”模式。</p> <p>如果调节失败，操作模式的切换将取决于“ActivateRecoverMode”。</p>
2	精确调节（在运行时调节/TIR）	<p>此模式通过设定值确定 PID 控制器的最佳参数设置。</p> <p>与 PID_Compact 不同，对于 PID_Temp，必须使用“Heat.EnableTuning”和“Cool.EnableTuning”参数选择是否需要启用加热调节或冷却调节。</p> <p>用户可以从“未激活”、“自动”模式或“手动”模式激活“精确调节”。</p> <p>如果调节成功，PID_Temp 将切换到“自动”模式。</p> <p>如果调节失败，操作模式的切换将取决于“ActivateRecoverMode”。</p>
3	“自动”模式	<p>在“自动”模式（标准 PID 控制模式）下，PID 算法的结果确定输出值。</p> <p>如果发生错误，PID_Temp 将切换到“未激活”，并且 “ActivateRecoverMode” = FALSE。如果发生错误并且 “ActivateRecoverMode” = TRUE，操作模式的切换将取决于错误。有关更多信息，请参见 PID_Temp 指令 ErrorBit 参数 (页 236)。</p>

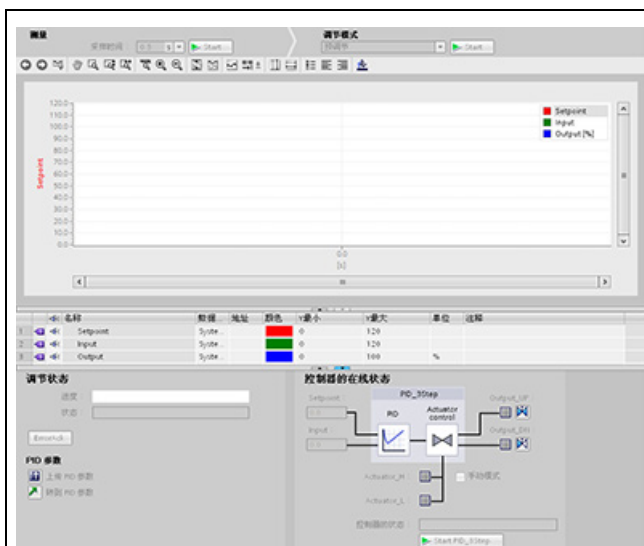
“Mode”/“State”	名称	说明
4	“手动”模式	<p>在这种模式下，PID 控制器将参数“ManualValue”的值标定、限制并传送到输出。</p> <p>PID 控制器在 PID 算法的标定中分配“ManualValue”（类似于“PidOutputSum”），所以它的值决定它对加热或冷却输出是否有效。</p> <p>可通过设置 “Mode” = 4 或 “ManualEnable”= TRUE 进入此模式。</p>
5	通过错误监视替换输出值（“恢复”模式）	<p>可以通过设置 “Mode” = 5. 激活此模式。如果在错误发生时“自动”模式处于激活状态，此模式将是控制器的自动错误响应：</p> <ul style="list-style-type: none"> • SetSubstituteOutput = FALSE（上一个有效输出值） • SetSubstituteOutput = TRUE（存储在参数“SubstituteOutput”中的值） <p>当 PID_Temp 处于“自动”模式并且 “ActivateRecoverMode” = TRUE 时，PID_Temp 将在出现以下错误时更改为此模式：</p> <ul style="list-style-type: none"> • ““Input_PER”参数的值无效。请检查模拟量输入是否出错（例如，断线）。” (ErrorBits = DW#16#0002) • ““Input”参数的值无效。值不是一个数字。” (ErrorBits = DW#16#0200) • “输出值计算失败。请检查 PID 参数。” (ErrorBits = DW#16#0400) • ““Setpoint”参数的值无效。值不是一个数字。” (ErrorBits = DW#16#1000) <p>当错误不再处于未决状态时，PID_Temp 将自动切换回“自动”模式。</p>

8.10 调试 PID_Compact 和 PID_3Step 控制器

使用调试编辑器可组态 PID 控制器，使其在启动时和操作过程中可自动调节。
要打开调试编辑器，请单击指令或项目浏览器上的图标。



表格 8-15 调试画面示例 (PID_3Step)



- 测量：
要在实时趋势中显示设定值、过程值（输入值）和输出值，请输入采样时间并单击“开始”(Start) 按钮。
 - 调节模式： 要调节 PID 循环，请选择“预调节”(Pretuning) 或“精确调节”(Fine tuning)（手动）并单击“开始”(Start) 按钮。
- PID**
控制器会运行多个阶段，以计算系统响应时间和更新时间。
通过这些值可计算相应的调节参数。
- 完成调节过程之后，可以单击调试编辑器的“PID 参数”(PID Parameters) 部分中的“上传 PID 参数”(Upload PID parameters) 按钮来存储新参数。
- 如果在调节过程中发生错误，PID 的输出值会变为 0。随后 PID 模式将被设置为“未激活”模式。状态可指示错误。

PID 起始值控制

您可以编辑 PID 组态参数的实际值，以便可以通过在线模式优化 PID 控制器的特性。

打开 PID 控制器的“工艺对象”(Technology objects) 及其“组态”(Configuration) 对象。要访问起始值控制，单击此对话框左上角的“眼镜图标”：



现在可以更改 PID 控制器组态参数的任何值，如下图所示。

可以将实际值与每个参数的项目（离线）起始值和 PLC（在线）起始值进行比较。这对于比较工艺对象数据块 (TO-DB) 的在线/离线差异以及了解在 PLC 下一次“停止到开始”转换时哪些值将用作当前值很有必要。

此外，比较图标还会通过视觉指示帮助您轻松确定在线/离线差异：



上图展示了带有比较图标的 PID 参数画面，其中显示出在线和离线项目之间有哪些值存在差异。绿色图标表示值相同，蓝色/橙色图标表示值不同。

另外，单击带有向下箭头的参数按钮，可打开一个显示每个参数的项目（离线）起始值和 PLC（在线）起始值的小窗口：

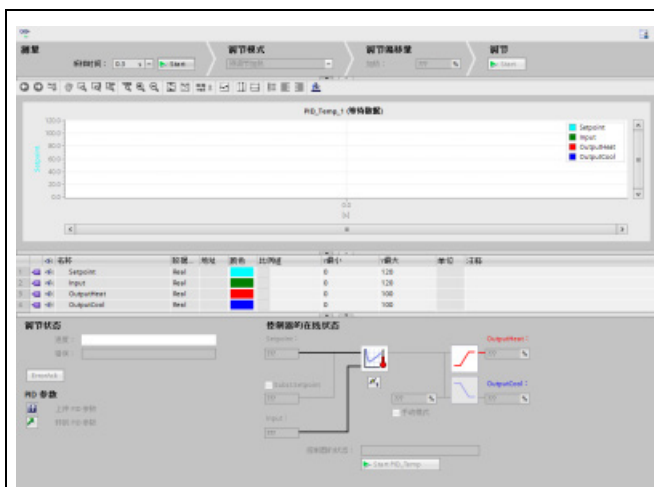


8.11 调试 PID_Temp 控制器

使用调试编辑器可组态 PID 控制器，使其在启动时和操作过程中可自动调节。
要打开调试编辑器，请单击指令或项目浏览器上的图标。



表格 8-16 调试画面示例 (PID_Temp)



测量:

要在实时趋势中显示设定值、过程值（输入值）和输出值，请输入采样时间并单击“开始”(Start)按钮。

调节模式：要调节 PID_Temp 循环，请选择“预调节”(Pretuning) 或“精确调节”(Fine

tuning) (手动) 并单击“开始”(Start) 按钮。PID 控制器会运行多个阶段，以计算系统响应时间和更新时间。通过这些值可计算相应的调节参数。完成调节过程之后，可以单击调试编辑器的“PID 参数”(PID Parameters) 部分中的“上传 PID 参数”(Upload PID parameters) 按钮来存储新参数。

如果调节期间未发生错误，则 PID 的输出值变为“0”。PID 模式将设置为“未激活”模式。状态可指示错误。

PWM 限值

使用 PID_Temp 的软件 PWM

功能控制的执行器可能需要保护，以免出现太短的脉冲持续时间（例如，可控硅继电器需要开启 20 ms 以上才能正常反应）；用户指定了一个最短时间。

执行器还可以忽略比较短的脉冲并因此影响控制质量。

需要设置一个最短的关断时间（例如，防止过热）。

要显示 PWM 限值视图，必须在工艺对象 (TO)

组态中打开功能视图，并从导航树中的“高级设置”(Advanced settings) 节点选择“PWM 限值”(PWM limits)。

如果在功能视图中打开“PWM 限值”(PWM limits)视图并激活监视（“玻璃”按钮），所有控件都会以橙色背景色显示来自 TO-DB 的在线监视值和多值控件，并且用户可以编辑这些值（前提是满足组态条件；请参见下表）。



设置	TO-DB 参数	数据类型	取值范围	说明
最短时间（加热） ^{1,2}	"Config.Output.Heat.MinimumOnTime"	Real	100000.0 >= "Config.Output.Heat.MinimumOnTime" >= 0.0	"OutputHeat_PWM"中的脉冲永不会短于该值。
最短关断时间（加热） ^{1,2}	"Config.Output.Heat.MinimumOffTime"	Real	100000.0 >= "Config.Output.Heat.MinimumOffTime" >= 0.0	"OutputHeat_PWM"中的中断永不会短于该值。

8.11 调试 PID_Temp 控制器

设置	TO-DB 参数	数据类型	取值范围	说明
最短时间（冷却） ^{1,3,4}	“Config.Output.Cool.MinimumOnTime”	Real	100000.0 >= Config.Output.Cool.MinimumOnTime >= 0.0	“OutputCool_PWM”中的脉冲永不会短于该值。
最短关断时间（冷却） ^{1,3,4}	“Config.Output.Cool.MinimumOffTime”	Real	100000.0 >= Config.Output.Cool.MinimumOffTime >= 0.0	“OutputCool_PWM”中的中断永不会短于该值。

- 1 该域显示“s”（秒）作为时间单位。
- 2 如果“基本设置”(Basic settings) 视图中的选择输出（加热）不是“OutputHeat_PWM”(Config.Output.Heat.Select = TRUE), 则应将该值设为“0.0”。
- 3 如果“基本设置”(Basic settings) 视图中的选择输出（冷却）不是“OutputCool_PWM”(Config.Output.Cool.Select = TRUE), 则应将该值设为“0.0”。
- 4 仅当选中“基本设置”(Basic settings) 视图中的“激活输出（冷却）”(Activate output (cooling)) (Config.ActivateCooling = TRUE) 时才可用。

PID 参数

下面显示了“高级设置”(Advanced settings) 视图的“PID 参数”(PID Parameters) 部分，禁用了冷却和/或“PID 参数切换”功能。

设置	TO-DB 参数	数据类型	取值范围	说明
启用 手动输入	“Retain.CtrlParams. SetByUser”	Bool	Bool	必须选中此复选框才能手动输入 PID 参数。
比例增益 (加热) ²	“Retain.CtrlParams. Heat.Gain”	Real	Gain >= 0.0	PID 加热比例增益
积分作用 时间 (加 热) ^{1,2}	“Retain.CtrlParams. Heat.Ti”	Real	100000.0 >= Ti >= 0.0	PID 加热积分作用时间。
微分作用 时间 (加 热) ^{1,2}	“Retain.CtrlParams. Heat.Td”	Real	100000.0 >= Td >= 0.0	PID 加热微分作用时间。
微分延迟 系数 (加 热) ²	“Retain.CtrlParams. Heat.TdFiltRatio”	Real	TdFiltRatio >= 0.0	PID 加热微分延迟系数， 定义微分滞后时间作 为 PID 微分时间中的系数。
比例作用 加权 (加 热) ²	“Retain.CtrlParams. Heat.PWeighting”	Real	1.0 >=PWeighting >= 0.0	PID 加热比例增益的加权 ，采用直接或环路控 制路径。

8.11 调试 PID_Temp 控制器

设置	TO-DB 参数	数据类型	取值范围	说明
微分作用 加权（加 热） ²	“Retain.CtrlParams. Heat.DWeighting”	Real	1.0 >=DWeighting >= 0.0	PID 加热微分部分的加权 ，采用直接或环路控 制路径。
PID 算法采样 时间（加 热） ^{1,2}	“Retain.CtrlParams. Heat.Cycle”	Real	100000.0 >=Cycle > 0.0	PID 控制器用于加热的内 部调用周期。 舍入为 FB 调用周期时间的整数 倍。
死区宽度 （加热） ^{2,3}	“Retain.CtrlParams. Heat.DeadZone”	Real	DeadZone>= 0.0	加热控制偏差的死区 宽度。
控制区（ 加热） ^{2,3}	“Retain.CtrlParams. Heat.ControlZone”	Real	ControlZone> 0.0	在 PID 控制处于激活状态时 用于加热的控制偏差 区的宽度。 如果控制偏差超出此 范围，输出将切换到 最大输出值。 默认值为“MaxReal” ，所以只要不执行自 动调节，控制区就会 被禁用。 禁止对控制区使用值 “0.0”；使用值“0.0” 时，PID_Temp 类似于双位置控制器 ，始终以满功率加热 或冷却。

设置	TO-DB 参数	数据类型	取值范围	说明
控制器结构（加热）	“PIDSelfTune.SUT.TuneRuleHeat”、 “PIDSelfTune.TIR.TuneRuleHeat”	Int	"PIDSelfTune.SUT.TuneRuleHeat" = 0..2, "PIDSelfTune.TIR.TuneRuleHeat" = 0..5	<p>用户可以选择加热调节算法。</p> <p>可选择：</p> <ul style="list-style-type: none"> • PID (Temperature) (默认值) ("PIDSelfTune.SUT.TuneRuleHeat" = 2) ("PIDSelfTune.TIR.TuneRuleHeat" = 0) • PID ("PIDSelfTune.SUT.TuneRuleHeat" = 0) ("PIDSelfTune.TIR.TuneRuleHeat" = 0) • PI ("PIDSelfTune.SUT.TuneRuleHeat" = 1) ("PIDSelfTune.TIR.TuneRuleHeat" = 4) <p>任何其它组合都会显示“用户自定义”，但“用户自定义”并非默认提供。</p> <p>“PID (Temperature)”是 PID_Temp 的一种新算法，可使用特定的预调节 (SUT) 方法处理温度。</p>

8.11 调试 PID_Temp 控制器

设置	TO-DB 参数	数据类型	取值范围	说明
比例增益 (冷却) 4	“Retain.CtrlParams. Cool.Gain”	Real	Gain >= 0.0	PID 冷却比例增益
积分作用 时间 (冷 却) 1,4	“Retain.CtrlParams. Cool.Ti”	Real	100000.0 >=Ti >= 0.0	PID 冷却积分作用时间
微分作用 时间 (冷 却) 1,4	“Retain.CtrlParams. Cool.Td”	Real	100000.0 >=Td >= 0.0	PID 冷却微分作用时间
微分延迟 系数 (冷 却) 4	“Retain.CtrlParams. Cool.TdFiltRatio”	Real	TdFiltRatio>= 0.0	PID 冷却微分延迟系数, 定义微分滞后时间作 为 PID 微分时间中的系数。
比例作用 加权 (冷 却) 4	“Retain.CtrlParams. Cool.PWeighting”	Real	1.0 >=PWeighting >= 0.0	PID 冷却比例增益的加权 , 采用直接或环路控 制路径。
微分作用 加权 (冷 却) 4	“Retain.CtrlParams. Cool.DWeighting”	Real	1.0 >=DWeighting >= 0.0	PID 冷却微分部分的加权 , 采用直接或环路控 制路径。
PID 算法采样 时间 (冷 却) 1,4	“Retain.CtrlParams. Cool.Cycle”	Real	100000.0 >=Cycle > 0.0	PID 控制器用于冷却的内 部调用周期。 舍入为 FB 调用周期时间的整数 倍。
死区宽度 (冷却) 3, 4	“Retain.CtrlParams. Cool.DeadZone”	Real	DeadZone>= 0.0	冷却控制偏差的死区 宽度

设置	TO-DB 参数	数据类型	取值范围	说明
控制区（冷却） ^{3,4}	“Retain.CtrlParams. Cool.ControlZone”	Real	ControlZone> 0.0	<p>在 PID 控制处于激活状态时用于冷却的控制偏差区的宽度。</p> <p>如果控制偏差超出此范围，输出将切换到最大输出值。</p> <p>默认值为“MaxReal”，所以只要不执行自动调节，控制区就会被禁用。</p> <p>禁止对控制区使用值“0.0”；使用值“0.0”时，PID_Temp 类似于双位置控制器，始终以满功率加热或冷却。</p>

8.11 调试 PID_Temp 控制器

设置	TO-DB 参数	数据类型	取值范围	说明
控制器结构 (冷却)	"PIDSelfTune.SUT.TuneRuleCool"、 "PIDSelfTune.TIR.TuneRuleCool"	Int	"PIDSelfTune.SUT.TuneRuleHeat" = 0..2, "PIDSelfTune.TIR.TuneRuleHeat" = 0..5	<p>用户可以选择冷却调节算法。</p> <p>可选择:</p> <ul style="list-style-type: none"> • PID (Temperature) (默认值) ("PIDSelfTune.SUT.TuneRuleCool" = 2) ("PIDSelfTune.TIR.TuneRuleCool = 0) • PID ("PIDSelfTune.SUT.TuneRuleCool" = 0) ("PIDSelfTune.TIR.TuneRuleCool" = 0) • PI ("PIDSelfTune.SUT.TuneRuleCool" = 1) ("PIDSelfTune.TIR.TuneRuleCool" = 4) <p>任何其它组合都会显示“用户自定义”，但“用户自定义”并非默认提供。</p> <p>“PID (Temperature)”是 PID_Temp 的一种新算法，可使用特定的预调节 (SUT) 方法处理温度。</p> <p>仅当选中/选择以下项目时可用 入门手册 “基本设置”(Basic settings)</p>

设置	TO-DB 参数	数据类型	取值范围	说明
----	----------	------	------	----

- 1 该域显示“s”（秒）作为时间单位。
- 2 仅当选中了 PID 参数中的“启用手动输入”(Enable manual entry) ("Retain.CtrlParams.SetByUser" = TRUE) 时才可用。
- 3 根据“基本设置”(Basic settings) 视图中的选择，计量单位显示在域的末尾。
- 4 仅当选中的/选择以下项目时可用：PID 参数中的“启用手动输入”(Enable manual entry) ("Retain.CtrlParams.SetByUser" = TRUE)， “基本设置”(Basic settings) 视图中的“激活输出（冷却）”(Activate output (cooling)) ("Config.ActivateCooling" = TRUE) 和“输出设置”(Output settings) 视图中的“PID 参数切换”(PID parameter switchover) (Config.AdvancedCooling = TRUE)。

PID 起始值控制

您可以编辑 PID 组态参数的实际值，以便可以通过在线模式优化 PID 控制器的特性。

打开 PID 控制器的“工艺对象”(Technology objects) 及其“组态”(Configuration) 对象。要访问起始值控制，单击此对话框左上角的“眼镜图标”：



8.11 调试 PID_Temp 控制器

现在可以更改 PID 控制器组态参数的任何值，如下图所示。

可以将实际值与每个参数的项目（离线）起始值和 PLC（在线）起始值进行比较。这对于比较工艺对象数据块 (TO-DB) 的在线/离线差异以及了解在 PLC 下一次“停止到开始”转换时哪些值将用作当前值很有必要。此外，比较图标还会通过视觉指示帮助您轻松确定在线/离线差异：



上图展示了带有比较图标的 PID 参数画面，其中显示出在线和离线项目之间有哪些值存在差异。绿色图标表示值相同，蓝色/橙色图标表示值不同。

另外，单击带有向下箭头的参数按钮，可打开一个显示每个参数的项目（离线）起始值和 PLC（在线）起始值的小窗口：

	加热	制冷
<input checked="" type="checkbox"/> 启用手动输入		
比例增益:	1.0	1.0
积分作用时间:	20.0 s	20.0 s
微分作用时间:	0.0 s	0.0 s
微分增益系数:	0.2	0.2
比例作用权重:	1.0	1.0
微分作用权重:	1.0	1.0
PID 算法采样时间:	1.0 s	1.0 s
死区宽度:	0.0 °C	0.0 °C
控制区宽度:	3.402822E+38 °C	3.402822E+38 °C
调节规则	控制器结构: PID (温度)	控制器结构: PID (温度)

便于连接 Internet 的 Web 服务器

Web 服务器允许用户通过 Web 页面来访问 CPU 的相关数据以及 CPU 中的过程数据。借助这些 Web 页面，可以通过 PC 或移动设备的 Web 浏览器访问 CPU（或支持 Web 的 CP）。已授权用户可使用标准 Web 页面执行以下功能及更多功能：

- 更改 CPU 的操作模式（STOP 和 RUN）
- 监视和修改 PLC 变量、数据块变量和 I/O 值
- 查看和下载数据日志
- 查看 CPU 的诊断缓冲区。
- 更新 CPU 的固件。

Web 服务器还允许您创建可访问 CPU 数据的用户自定义 Web 页面。您可以选择相应的 HTML 创作软件来开发这类页面。在 HTML 代码中插入预定义的“AWP”（Automation Web Programming，自动化 Web 编程）命令可访问 CPU 中的数据。

您可以使用 STEP 7 在 CPU 的设备组态中设置 Web 服务器的用户和权限级别。

Web 浏览器要求

Web 服务器支持以下 PC Web 浏览器：

- Internet Explorer 8.0
- Internet Explorer 9.0
- Mozilla Firefox 17.0.1
- Google Chrome 23.0
- Apple Safari 5.1.7 (Windows)
- Apple Safari 6.0.2 (Mac)

Web 服务器支持以下移动设备 Web 浏览器：

- Internet Explorer 6.0 及更低版本（对于 HMI 面板）
- Mobile Safari 7534.48.3 (iOS 5.0.1)
- Mobile Android Browser 2.3.4
- Mobile Google Chrome 23.0

9.1 轻松使用标准 Web 页面

有关可干扰标准或用户自定义的 Web 页面显示的浏览器相关的限制，请参见限制 (页 279) 相关主题。

9.1 轻松使用标准 Web 页面

使用标准 Web 页面非常容易！用户只需在组态 CPU 时启用 Web 服务器并为 Web 服务器用户组态执行所需任务的权限。



“启动”(Start) 页面显示所连接 CPU 的图示，并列出了有关 CPU 的常规信息。如果您启用了 Web 服务器的 CP，“启动”(Start) 页面还将显示这些 CP 并允许通过这些 CP 连接 Web 页面。如果您具有所需权限，则可以更改 CPU 的操作模式 (STOP 和 RUN) 或者使 LED 指示灯闪烁。



“变量状态”(Variable Status) 页面允许您监视或修改 CPU 中的任何 I/O 或存储器数据。必须具有“读取变量状态”的权限才能监视值，必须具有“写入变量状态”的权限才能修改值。可以输入直接地址 (如 I0.0)、PLC 变量名或特定程序块的变量。数据值会自动刷新，除非您禁用自动刷新选项。



“诊断缓冲区”页面显示诊断缓冲区，可由具有查询诊断权限的用户访问。可以选择要显示的诊断条目的范围。诊断条目会列出已发生的事件以及事件发生时 CPU 的时间和日期。选择单独的事件可显示有关该事件的详细信息。



“文件浏览器”页面允许用户查看、下载或编辑 CPU

装载存储器中的文件，例如数据日志 (页 129) 和配方。除非 CPU 的保护级别为 4，否则所有用户都可通过“文件浏览器”页面查看文件。

具有修改文件权限的用户可以删除、编辑或重命名文件。



“模块信息”页面除了显示有关站中的模块信息外，还允许您更新 CPU

或支持固件更新的其它模块中的固件。

具有查询诊断权限的用户可以查看模块信息页面。

具有执行固件更新权限的用户可以更新固件

。

其它标准 Web 页面会显示 CPU

的相关信息（如序列号、版本和产品编号）和通信参数的信息（如网络地址和通信接口的物理属性）。



警告

通过 Web 服务器对 CPU 进行未经授权的访问

未经授权访问 CPU 或将 PLC

变量更改为无效值可能会中断过程操作并可能导致死亡、严重人身伤害和/或财产损失。

由于启用 Web 服务器可让授权用户执行操作模式更改、写入 PLC 数据以及进行固件更新，Siemens 建议遵照以下安全实践：

- 仅使用 HTTPS 协议启用对 Web 服务器的访问。
- 使用可靠的密码对 Web 服务器用户 ID 进行密码保护。
强密码在长度上至少为十个字符，可以是字母、数字和特殊字符的组合，不能是可在字典上找到的词，并且不能是从个人信息推断出的名字或标识符。
保管好密码并经常更改密码。
- 不要扩展“所有人”(Everybody) 用户的默认最低权限。
- 对程序逻辑中的变量执行错误检查和范围检查，因为 Web 页面用户可将 PLC 变量更改为无效值。
- 如果您不在受保护的网路范围内，请使用安全的虚拟专用网络 (VPN) 连接到 S7-1200 PLC Web 服务器。

9.2 可影响到 Web 服务器使用的限制

下列 IT 因素可能会影响 Web 服务器的使用：

- 通常，您必须使用 CPU 的 IP 地址或具有端口号的无线路由器的 IP 地址访问标准 Web 页面或用户自定义的 Web 页面。如果 Web 浏览器不允许直接连接到 IP 地址，请咨询 IT 管理员。如果本地策略支持 DNS，您可以通过指向该地址的 DNS 条目连接到 IP 地址。
- 防火墙、代理设置和其它网站特定的限制也会限制对 CPU 的访问。请咨询 IT 管理员来解决这些问题。
- 标准 Web 页面采用 JavaScript 和 cookie。如果 Web 浏览器设置禁用了 JavaScript 或 cookie，请将其启用。如果无法启用，某些功能将受限。可根据需要来选择是否在用户自定义 Web 页面中使用 JavaScript 和 cookie。如果使用，则必须在浏览器中将其启用。
- Web 服务器支持安全套接层 (SSL)。可通过 URL `http://ww.xx.yy.zz` 或 `https://ww.xx.yy.zz` 来访问标准 Web 页面和用户定义的 Web 页面，其中，“ww.xx.yy.zz”表示 CPU 的 IP 地址。
- 西门子为 Web 服务器的安全访问提供了安全证书。可以从标准 Web 页面简介下载安全证书，并将证书导入 Web 浏览器的 Internet 选项中。如果选择不导入证书，则每次以 `https://` 形式访问 Web 服务器时都会出现安全验证提示。

连接数目

Web 服务器最多支持 30 个活动的 HTTP 连接。可以各种方式使用这 30 个连接，具体取决于您所使用的 Web 浏览器以及每页不同对象（.css 文件、图像、其它 .html 文件）的数目。

在显示页面时某些连接仍然存在；其它连接在初始连接之后便不再存在。

例如，如果使用最多支持六个持续连接的 Mozilla Firefox 8，则在 Web 服务器开始丢弃连接前可使用五个浏览器或浏览器选项卡实例。

如果页面未使用所有六个连接，则可使用更多浏览器或浏览器选项卡实例。

还要注意的，活动连接的数目会影响页面性能。

说明

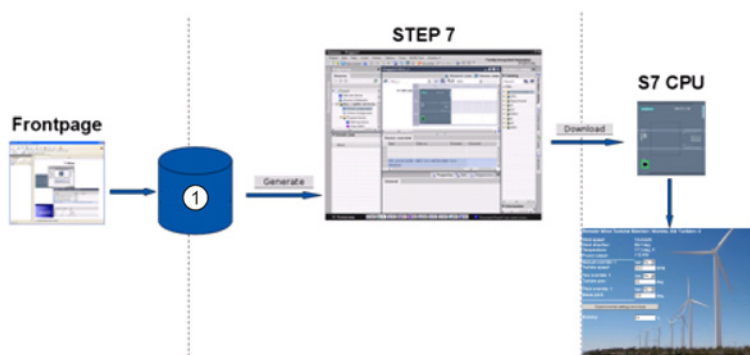
关闭 Web 服务器前注销

如果您已登录到 Web 服务器，请确保在关闭 Web 浏览器前先注销。Web 服务器最多支持 7 个并发登录。

9.3 轻松创建用户定义的 Web 页面

9.3.1 轻松创建定制的“用户自定义”Web 页面

S7-1200 Web 服务器还提供了一些方法，供您创建可融入 PLC 数据的应用特定的 HTML 页面。可以选择相应的 HTML 编辑器来创建这些页面，然后从可通过标准 Web 页面访问的位置将这些页面下载到 CPU。



① 具有嵌入式 AWP 命令的 HTML 文件

该过程涉及到以下几项任务：

- 使用 HTML 编辑器创建 HTML 页面。
- 以 HTML 注释的形式将 AWP 命令包含在 HTML 代码中： AWP 命令是一组用于访问 CPU 信息的固定命令。
- 将 STEP 7 组态为读取和处理 HTML 页面。
- 基于 HTML 页面生成程序块。
- 对 STEP 7 进行编程设计，以控制 HTML 页面的使用。

- 编译程序块并将其下载到 CPU。
- 通过 PC 或移动设备访问用户自定义的 Web 页面。

可以根据需要使用软件包来创建自己的用于 Web 服务器的 HTML 页面。请确保 HTML 代码符合 W3C（万维网联盟）的 HTML 标准。STEP 7 不会对 HTML 语法进行任何验证。

可使用能够以所见即所得或设计版式模式进行设计的软件包，但是必须能够在纯 HTML 表单中编辑 HTML 代码。大部分 Web 编写工具可以提供这种类型的编辑；否则，您始终可以使用简单文本编辑器来编辑 HTML 代码。将以下代码行包含在 HTML 页面中，以将页面的字符集设置为 UTF-8：

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

还要确保以 UTF-8 字符编码的格式保存编辑器中的文件。

使用 STEP 7 将 HTML 页面中的所有内容编译到 STEP 7 数据块中。

这些数据块由一个管理 Web 页面显示的控制数据块、一个或多个包含已编译 Web 页面的片段数据块组成。请注意，如果有大量的 HTML 网页，尤其是那些具有很多图像的页面，它们的片段 DB 需要非常大的装载存储器空间。如果 CPU 的内部装载存储器不足以容纳用户定义的 Web 页面，请使用存储卡来提供外部装载存储器。

要编写 HTML 代码程序以使用 S7-1200 中的数据，应将 AWP 命令以 HTML 注释的形式包含在代码中。完成之后，将 HTML 页面保存到 PC，并记下保存这些页面的文件夹路径。

说明

对于包含 AWP 命令的 HTML 文件，其文件大小限制为 64 KB。必须确保您的文件大小小于此限制。

刷新用户定义的 Web 页面

用户定义的 Web 页面不会自动刷新。可以根据需要选择是否编写用来刷新页面的 HTML 程序。对于显示 PLC 数据的页面，定期刷新可使数据保持最新。

对于用作数据输入格式的 HTML 页面，刷新可能会干扰用户输入数据。

如果希望整个页面自动进行刷新，可将以下命令行添加到 HTML 头文件，其中，“10”表示两次刷新间隔的时间（以秒为单位）：

```
<meta http-equiv="Refresh" content="10">
```

还可以使用 JavaScript 或其它 HTML 方法来控制页面或数据的刷新。相关信息，请参考 HTML 和 JavaScript 文档。

9.3.2 特定于用户定义 Web 页面的限制

标准 Web 页面的限制也适用于用户定义 Web 页面。此外，用户定义 Web 页面会有一些特殊考虑。

装载存储器空间

单击“生成块”(Generate blocks) 之后，用户定义的 Web 页面就会成为数据块，这一过程需要用到装载存储器空间。

如果安装了存储卡，用户定义 Web 页面的外部装载存储器空间最大容量即为存储卡的容量。

如果未安装存储卡，这些块就会占用内部装载存储器空间，根据 CPU 型号的不同而会存在限制。

可以在 STEP 7

中通过“在线和诊断”工具检查已用装载存储器空间量和可用装载存储器空间量。还可以查看 STEP 7 基于用户定义 Web 页面生成的各个块的属性，并查看装载存储器使用量。

说明

如果需要减少用户定义 Web 页面所需空间，则减少图片的使用（如果适用）。

文本字符串中的引号

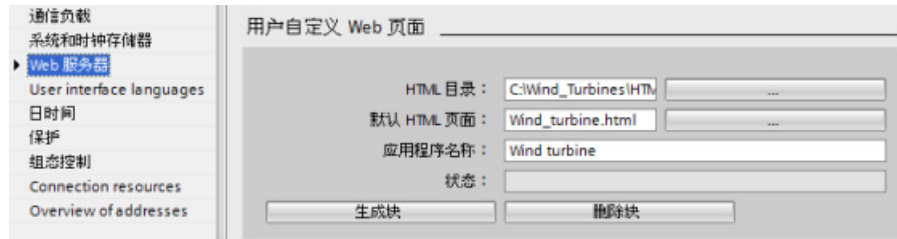
在用户自定义 Web 页面中，避免在用于任意目的的数据块变量中使用含有嵌入式单引号或双引号的文本字符串。因为在 HTML 语法中，经常将单引号或双引号用作分隔符，文本字符串内的引号会破坏用户自定义 Web 页面的显示。

对于在用户自定义 Web 页面中使用的 String 型数据块变量，应遵守以下准则：

- 不要在 STEP 7 中为数据块变量字符串值输入单引号或双引号。
- 不要让用户程序将含有引号的字符串分配给这些数据块变量。

9.3.3 用户定义的 Web 页面的组态

要组态用户自定义的 Web 页面，应编辑 CPU 的“Web 服务器”(Web server) 属性。



启用 Web 服务器功能之后，输入以下信息：

- HTML 默认起始页面的名称及当前位置，以生成用户自定义的 Web 页面的 DB。
- 应用程序的名称（可选）。应用程序名称用于对 Web 页面进一步分类或分组。在提供应用程序名称时，Web 服务器以下面的格式为用户自定义页面创建 URL：

`http[s]://ww.xx.yy.zz/awp/<应用程序名称>/<页面名称>.html`

- 包含 AWP 命令的文件的文件扩展名。STEP 7 会默认分析扩展名为 .htm、.html 或 .js 的文件。如果还有其它文件扩展名，请将其附上。
- 控制 DB 号及初始片段 DB 的标识号。

对 Web 服务器进行组态之后，单击“生成块”(Generate blocks) 按钮，从 HTML 页面生成 DB。生成 DB 之后，Web 就会成为用户程序的一部分。用于 Web 页面操作的控制数据块以及“片段”DB 包含所有 HTML 页面。

9.3.4 使用 WWW 指令

WWW 指令允许通过标准 Web 页面访问用户自定义的 Web 页面。用户程序仅必须执行 WWW 指令一次以访问用户自定义的 Web 页面。

但是，可以选择仅在某些情况下执行用户自定义的 Web 页面。

用户程序就可以根据应用程序的要求调用 WWW 指令。

表格 9-1 WWW 指令

LAD/FBD	SCL	说明
	<pre>ret_val := #WWW(ctrl_db:=_uint_in_);</pre>	<p>识别要用于用户自定义的 Web 页面的控制 DB。</p> <p>该控制数据块是 WWW 指令的输入参数，指定如片段数据块中所表示的页面内容、状态以及控制信息。</p>

9.3 轻松创建用户定义的 Web 页面

用户程序通常会将控制 DB 当作“生成块”过程生成的 DB 来直接使用，而不进行任何处理。但是，用户程序可以在控制 DB 中设置全局命令，以取消激活 Web 服务器或随后重新启用 Web 服务器。对于创建为手动片段 DB 的用户自定义的页面，用户程序必须通过控制 DB 中的请求表来控制这些页面的特性，如“*S7-1200 可编程控制器系统手册*”中所述。

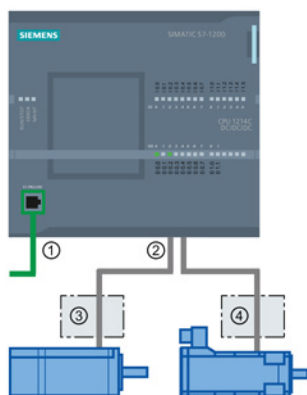
轻松进行运动控制

CPU 通过脉冲接口为步进电机和伺服电机的运行提供运动控制功能。运动控制功能负责对驱动器进行监控。

- “轴”工艺对象用于组态机械驱动器的数据、驱动器的接口、动态参数以及其它驱动器属性。
- 通过对 CPU 的脉冲输出和方向输出进行组态来控制驱动器。
- 用户程序使用运动控制指令来控制轴并启动运动任务。
- PROFINET 接口用于在 CPU 与编程设备之间建立在线连接。除了 CPU 的在线功能外，附加的调试和诊断功能也可用于运动控制。

说明

仅当 CPU 从 STOP 切换为 RUN 模式时，RUN 模式下对运动控制配置和下载的更改才会生效。



- ① PROFINET
- ② 脉冲和方向输出
- ③ 步进电机的电源部分
- ④ 伺服电机的电源部分

DC/DC/DC 型 CPU S7-1200

上配备有用于直接控制驱动器的板载输出。继电器型 CPU 需要具有用来控制驱动器的 DC 输出的信号板。

信号板 (SB, Signal Board) 将板载 I/O 扩展为包含多个附加 I/O 点。具有两个数字量输出的 SB 可用作控制一台电机的脉冲输出和方向输出。具有四个数字量输出的 SB 可用作控制两台电机的脉冲输出和方向输出。不能将内置继电器输出用作控制电机的脉冲输出。不论是使用板载 I/O、SB I/O 还是二者的组合，最多可以拥有四个脉冲发生器。

这四个脉冲发生器具有默认的 I/O 分配，但是，它们可组态为 CPU 或 SB 上的任意数字量输出。不能将 CPU 上的脉冲发生器分配至 SM 或分布式 I/O。

说明

用户程序中的其它指令无法使用脉冲串输出

将 CPU 或信号板的输出组态为脉冲发生器时（供 PWM 或运动控制指令使用），相应的输出地址不再控制输出。如果用户程序向用作脉冲发生器的输出写入值，则 CPU 不会将该值写入到物理输出。

表格 10- 1 可控制驱动器的最大数目

CPU 型号		板载 I/O; 未安装任何 SB		带 SB (2 x DC 输出)		带 SB (4 x DC 输出)	
		带方向	不带方向	带方向	不带方向	带方向	不带方向
CPU 1211C	DC/DC/DC	2	4	3	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1212C	DC/DC/DC	3	4	3	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1214C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1215C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLY	0	0	1	2	2	4
	DC/DC/RLY	0	0	1	2	2	4
CPU 1217C	DC/DC/DC	4	4	4	4	4	4

说明

最多四个脉冲发生器。

不论是使用板载 I/O、SB I/O 还是二者的组合，最多可以拥有四个脉冲发生器。

表格 10-2 CPU 输出：最大频率

CPU	CPU 输出通道	脉冲和方向输出	A/B, 正交, 上/下和脉冲/方向
1211C	Qa.0 到 Qa.3	100 kHz	100 kHz
1212C	Qa.0 到 Qa.3	100 kHz	100 kHz
	Qa.4、Qa.5	20 kHz	20 kHz
1214C 和 1215C	Qa.0 到 Qa.3	100kHz	100kHz
	Qa.4 到 Qb.1	20 kHz	20 kHz
1217C	DQa.0 到 DQa.3 (.0+, .0- 到 .3+, .3-)	1 MHz	1 MHz
	DQa.4 到 DQb.1	100 kHz	100 kHz

表格 10-3 SB 信号板输出：最大频率（可选信号板）

SB 信号板	SB 输出通道	脉冲和方向输出	A/B, 正交, 上/下和脉冲/方向
SB 1222, 200 kHz	DQe.0 到 DQe.3	200kHz	200 kHz
SB 1223, 200 kHz	DQe.0, DQe.1	200kHz	200 kHz
SB 1223	DQe.0, DQe.1	20 kHz	20 kHz

表格 10-4 脉冲输出的频率范围

脉冲输出	频率
板载	4 PTO: $2 \text{ Hz} \leq f \leq 1 \text{ MHz}$, 4 PTO: $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$, 或用于 4 个 PTO 的这些值的任意组合。 ¹²
标准 SB	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
高速 SB	$2 \text{ Hz} \leq f \leq 200 \text{ kHz}$

¹ 请参见下表，了解 CPU 1217C 的四种可能输出速度组合。

² 请参见下表，了解 CPU 1211C、CPU 1212C、CPU 1214C 或 CPU 1215C 的四种可能输出速度组合。

示例：CPU 1217C 脉冲输出速度组态

说明

使用板载差分输出，CPU 1217C 可生成高达 1 MHz 的脉冲输出。

以下示例显示了四种可能的输出速度组合：

- 示例 1：4 - 1 MHz PTO，不带方向输出
- 示例 2：1 - 1 MHz、2 - 100 kHz 和 1 - 20 kHz PTO，全部带方向输出
- 示例 3：4 - 200 kHz PTO，不带方向输出
- 示例 4：2 - 100 kHz PTO 和 2 - 200 kHz PTO，全部带方向输出

P = 脉冲 D = 方向		CPU 板载输出										高速 SB 输出				标准 SB 输出	
		1 MHz 输出 (Q)				100 kHz 输出 (Q)						200 kHz 输出 (Q)				20 kHz 输出 (Q)	
		0.0 +	0.1 +	0.2+	0.3 +	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		0.0-	0.1-	0.2-	0.3-												
示例 1: 4 - 1 MHz (无 方向 输出)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
示例 2: 1 - 1 MHz、 2 - 100 kHz 和 1 - 20 kHz (全 部带 方向输 出)	PTO1	P	D														
	PTO2					P	D										
	PTO3							P	D								
	PTO4															P	D

P = 脉冲 D = 方向	CPU 板载输出											高速 SB 输出				标准 SB 输出		
	PTO1	PTO2	PTO3	PTO4	PTO1	PTO2	PTO3	PTO4	PTO1	PTO2	PTO3	PTO4	PTO1	PTO2	PTO3	PTO4		
示例 3: 4 - 200 kHz (无 方向 输出)	PTO1												P					
	PTO2													P				
	PTO3														P			
	PTO4															P		
示例 4: 2 - 100 KHz; 2 - 200 KHz (全部带 方向输出)	PTO1					P	D											
	PTO2								P	D								
	PTO3												P	D				
	PTO4														P	D		

示例：CPU 1211C、CPU 1212C、CPU 1214C 和 CPU 1215C 脉冲输出速度组态

以下示例显示了四种可能的输出速度组合：

- 示例 1：4 - 100 kHz PTO，不带方向输出
- 示例 2：2 - 100 kHz PTO 和 2 - 20 kHz PTO，全部带方向输出
- 示例 3：4 - 200 kHz PTO，不带方向输出
- 示例 4：2 - 100 kHz PTO 和 2 - 200 kHz PTO，全部带方向输出

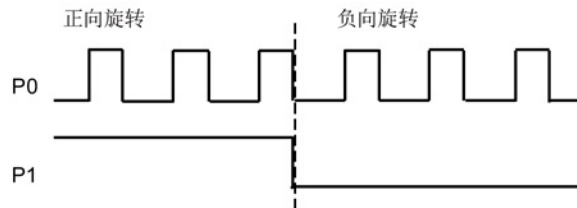
P = 脉冲 D = 方向	CPU 板载输出											高速 SB 输出				低速 SB 输出	
	100 kHz 输出 (Q)				20 kHz 输出 (Q)							200 kHz 输出 (Q)				20 kHz 输出 (Q)	
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1	
	CPU 1211C																
	CPU 1212C				CPU 1212C												
	CPU 1214C				CPU 1214C		CPU 1214C										

P = 脉冲 D = 方向		CPU 板载输出								高速 SB 输出				低速 SB 输出	
		CPU 1215C				CPU 1215C		CPU 1215C							
示例 1: 4 - 100 kHz (无 方向 输出)	PTO1	P													
	PTO2		P												
	PTO3			P											
	PTO4				P										
示例 2: 2 - 100 KHz; 2 - 20 KHz (全 部带方向 输出)	PTO1	P	D												
	PTO2			P	D										
	PTO3					P	D								
	PTO4							P	D						
示例 3: 4 - 200 kHz (无 方向 输出)	PTO1										P				
	PTO2											P			
	PTO3												P		
	PTO4													P	
示例 4: 2 - 100 KHz; 2 - 200 KHz (全 部带方向 输出)	PTO1	P	D												
	PTO2			P	D										
	PTO3										P	D			
	PTO4												P	D	

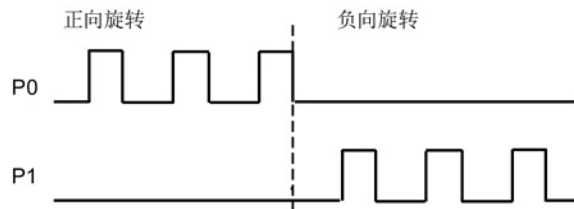
10.1 “定相”(Phasing)

步进/伺服驱动器的“定相”(Phasing) 接口有 4 个选项。选项如下：

- **PTO (脉冲 A 和方向 B)：** 如果选择 PTO (脉冲 A 和方向 B) 选项，则一个输出 (P0) 控制脉冲，另一输出 (P1) 控制方向。如果脉冲处于正向，则 P1 为高电平（激活）。如果脉冲处于负向，则 P1 为低电平（未激活）：



- **PTO (加计数 A 和减计数 B)：** 如果选择 PTO (加计数 A 和减计数 B) 选项，则一个输出 (P0) 脉冲控制正方向，另一个输出 (P1) 脉冲控制负方向：



- **PTO (A/B 相移)：** 如果选择 PTO (A/B 相移) 选项，则两个输出均以指定速度产生脉冲，但相位相差 90 度。它是一种 1X 组态，表示一个脉冲是 P0 的两次正向转换之间的时间量。这种情况下，方向由先变为高电平的输出转换决定。P0 领先 P1 表示正向。P1 领先 P0 表示负向。

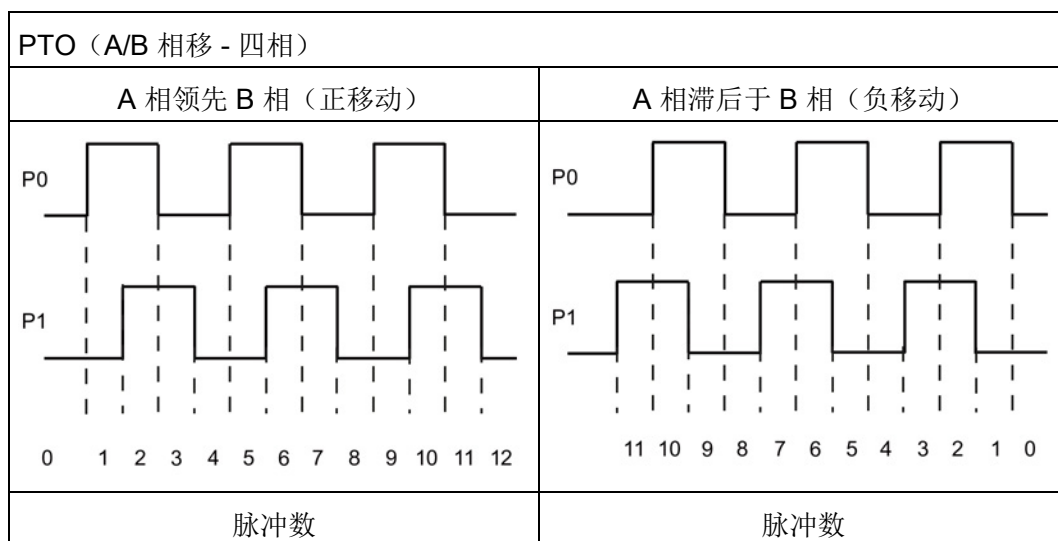
生成的脉冲数取决于 A 相的 0 到 1 的转换次数。相位关系决定了移动方向：

PTO (A/B 相移)	
A 相领先 B 相 (正移动)	A 相滞后于 B 相 (负移动)
脉冲数	脉冲数

10.1 “定相”(Phasing)

- PTO (A/B 相移 - 四相)： 如果选择 PTO (A/B 相移 - 四相) 选项，则两个输出均以指定速度产生脉冲，但相位相差 90 度。四相是一种 4X 组态，表示一个脉冲是每个输出（正向和负向）的转换。这种情况下，方向由先变为高电平的输出转换决定。P0 领先 P1 表示正向。P1 领先 P0 表示负向。

四相取决于 A 相和 B 相的正向和负向转换。您可以组态转换次数。相位关系 (A 领先 B 或 B 领先 A) 决定了移动方向。



- PTO (脉冲和方向 (已取消选择方向))： 如果在 PTO (脉冲和方向 (已取消选择方向)) 中取消方向输出，则输出 (P0) 控制脉冲。未使用输出 P1，输出 P1 可供其它程序使用。在此模式下 CPU 只接受正向运动命令。当您选择此模式时，运动控制限制进行非法负向组态。如果运动应用仅在一个方向进行，则可保存输出。单相 (一个输出) 如下图所示 (假设极性为正)：



10.2 组态脉冲发生器

1. 添加工艺对象：

- 在项目树中，展开节点“工艺对象”(Technology Objects)，然后选择“添加新对象”(Add new object)。
- 选择“轴”(Axis) 图标（必要时可以重命名），然后单击“确定”(OK) 打开轴对象的组态编辑器。
- 显示“基本参数”(Basic parameters) 下的“为轴控制选择 PTO”(Select PTO for Axis Control) 属性，然后选择所需脉冲。

说明

如果以前从未在“CPU 属性”(CPU Properties) 中组态 PTO，则将 PTO 组态为使用其中一个板载输出。

如果使用了输出信号板，则选择“设备组态”(Device configuration) 按钮以转到“CPU 属性”(CPU Properties)。在“参数分配”(Parameter assignment) 下的“脉冲选项”(Pulse options) 中，将数据源组态为信号板输出。

- 对其余的基本参数和扩展参数进行组态。

2. 对应用进行编程：将 MC_Power 指令插入代码块。

- 对于“轴”输入，请选择已创建并组态的轴工艺对象。
- 将 Enable 输入设置为 TRUE 可以使其它运动指令起作用。
- 将 Enable 输入设置为 FALSE 会取消其它运动指令。

说明

每个轴只包括一个 MC_Power 指令。

3. 插入其它运动指令，以生成所需的运动。

说明

将脉冲发生器组态为信号板输出：选择 CPU 的“脉冲发生器 (PTO/PWM)”(Pulse generators (PTO/PWM)) 属性（在“设备组态”(Device configuration) 中）并启用脉冲发生器。每个 S7-1200 CPU V1.0、V2.0、V2.1 和 V2.2 提供两个脉冲发生器。S7-1200 CPU V3.0 和 V4.0 CPU 提供四个脉冲发生器。在相同组态区域的“脉冲选项”(Pulse options) 下，选择用作以下用途的脉冲发生器：“PTO”。

说明

CPU 以 10 ms

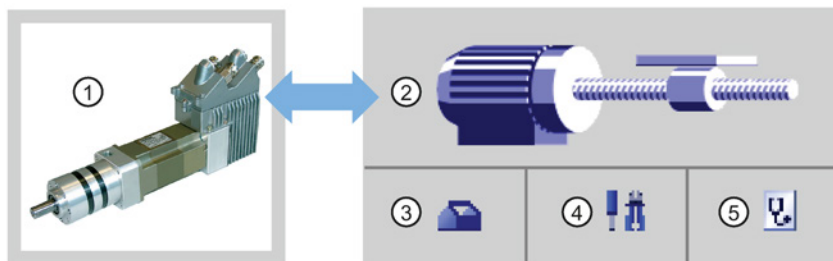
为“时间片”或时间段计算运动任务。执行一个时间片时，下一时间片会在队列中等待执行。如果中断某个轴上的运动任务（通过执行该轴的其他新运动任务），可能最多要等待 20 ms（当前时间片的剩余时间加上排队的时间片）才能执行新运动任务。

10.3 开环运动控制

10.3.1 组态轴

通过 PTO（脉冲串输出）在 PLC 和驱动器上连接开环轴。

STEP 7 为“轴”工艺对象提供组态工具、调试工具和诊断工具。



- ① 驱动器
- ② 工艺对象
- ③ 组态

- ④ 调试
- ⑤ 诊断

说明

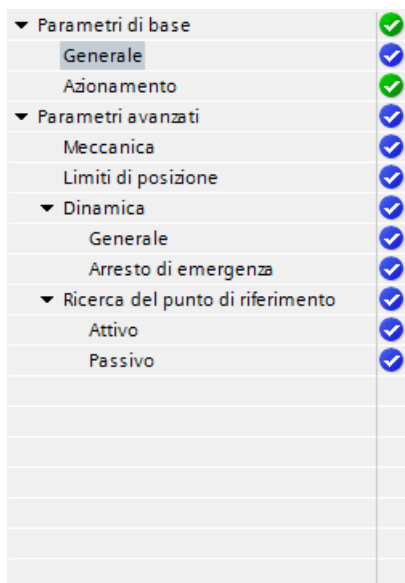
对于固件版本 V2.2 以及更早版本的 CPU，PTO 需使用高速计数器 (HSC) 的内部功能。也就是说，在其它地方无法使用相应的 HSC。

PTO 和 HSC 间的分配是固定的。如果激活 PTO1，则它将与 HSC1 连接。如果激活 PTO2，则它将与 HSC2 连接。生成脉冲时，不能监视当前值（例如，在 ID1000 中）。

S7-1200 V3.0 以及更高版本的 CPU 则无此限制；当在这些 CPU 中组态脉冲输出时，所有 HSC 仍可供程序使用。

表格 10-5 用于运动控制的 STEP 7 工具

工具	说明
组态	组态“轴”工艺对象的下列属性： <ul style="list-style-type: none"> • 要用的 PTO 的选择以及驱动器接口的组态 • 机械的属性和驱动器（机器或系统）的传动比参数 • 位置限制属性、动态属性和归位属性 在工艺对象的数据块中保存组态数据。
调试	无需创建用户程序即可测试轴的功能。启动该工具时，将显示控制面板。 控制面板上提供了下列命令： <ul style="list-style-type: none"> • 启用和禁用轴 • 在点动模式下移动轴 • 以绝对和相对方式定位轴 • 使轴归位 • 确认错误信息 可以为运动命令指定速度以及加速度/减速度。控制面板中还将显示当前的轴状态。
诊断	监视轴和驱动器的当前状态和错误信息。

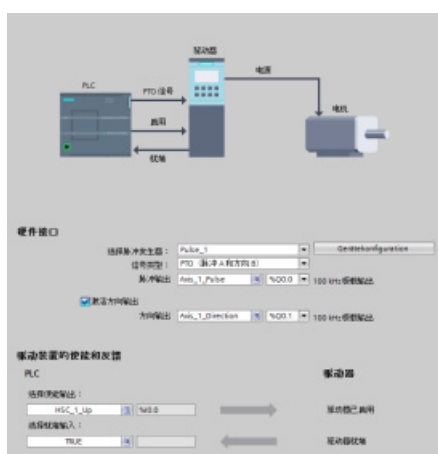


PTO

轴的树选择器不包括编码器、模数、位置监视和控制回路组态菜单。

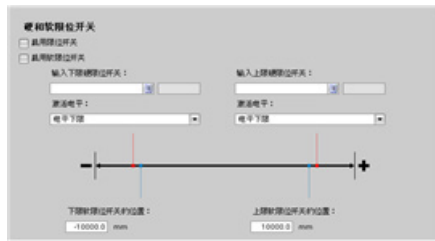


为轴创建工艺对象之后，通过定义基本参数（如 PTO 和驱动器接口的组态）来组态该轴。还可以组态轴的其他属性，例如位置限制属性、动态属性和归位属性。

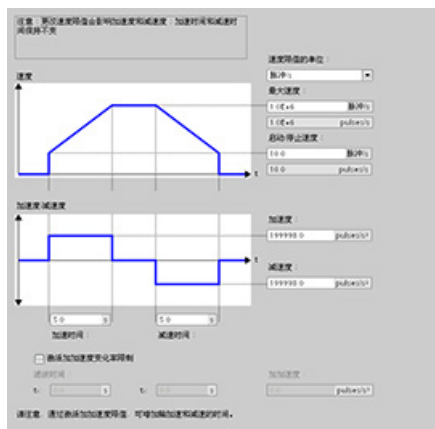


说明

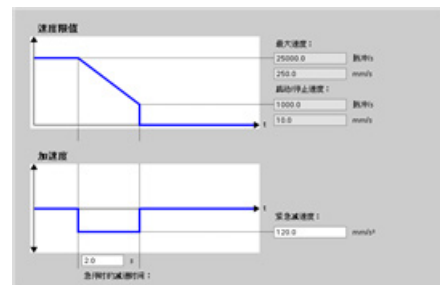
在用户程序中可以根据新量纲单位调整运动控制指令的输入参数值。



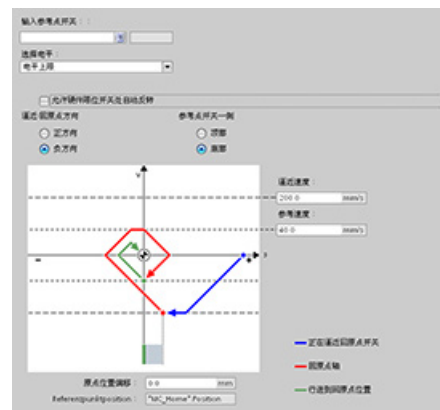
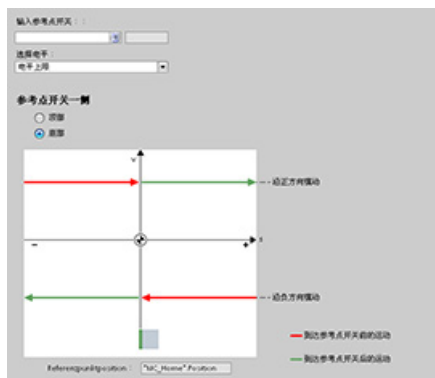
组态驱动器信号、驱动器机械装置和位置监视（硬件和软件限位开关）的属性。



组态急停命令的运动动态和行为。



还可以组态回原点行为（被动和主动）。



使用“调试”(Commissioning) 控制面板独立于用户程序对功能进行测试。

 单击“启动”(Startup) 图标对轴进行调试。

控制面板会显示轴的当前状态。

不仅可以启用和禁用轴，还可以测试轴定位（以绝对和相对方式）以及指定速度、加速度和减速度。还可以测试归位和点动任务。控制面板还可用于确认错误。

10.3.2 调试

“状态和错误位”诊断功能

诊断功能“状态和错误位”(Status and error bits) 用于监视轴的最重要状态和错误消息。当轴激活时，可以在在线模式下以“手动控制”模式和“自动控制”模式显示诊断功能。

表格 10-6 轴的状态

状态	说明
启用	轴已启用且准备好接受运动控制任务的控制。 (工艺对象的变量: <轴名称>.StatusBits.Enable)
已回原点	轴已回原点且能够执行运动控制指令“MC_MoveAbsolute”的绝对定位任务。 对于相对回原点而言，轴不必回原点。特殊情况： <ul style="list-style-type: none"> 主动回原点期间，该状态为 FALSE。 如果回原点的轴经受被动回原点，则在被动回原点期间该状态设置为 TRUE。 (工艺对象的变量: <轴名称>.StatusBits.HomingDone)
错误	“轴”工艺对象发生错误。 有关错误的更多信息，请参见自动控制模式下的运动控制指令的 ErrorID 和 ErrorInfo 参数。在手动模式下，控制面板中的“上一错误”(Last error) 字段显示更多错误原因信息。 (工艺对象的变量: <轴名称>.StatusBits.Error)
控制面板激活	在控制面板中启用了“手动控制”模式。控制面板对“轴”工艺对象具有优先控制权。不能通过用户程序来控制轴。 (工艺对象的变量: <轴名称>.StatusBits.ControlPanelActive)

表格 10-7 驱动器状态

状态	说明
驱动器准备就绪	驱动器准备好运行。 (工艺对象的变量: <轴名称>.StatusBits.DriveReady)
错误	驱动器在其准备就绪信号故障后报告了错误。 (工艺对象的变量: <轴名称>.ErrorBits.DriveFault)

表格 10-8 轴运动的状态

状态	说明
停止	轴处于停止状态。 (工艺对象的变量: <轴名称>.StatusBits.StandStill)
加速	轴在加速。 (工艺对象的变量: <轴名称>.StatusBits.Acceleration)
恒速	轴在恒速运转。 (工艺对象的变量: <轴名称>.StatusBits.ConstantVelocity)
减速	轴在减速(速度下降)。 (工艺对象的变量: <轴名称>.StatusBits.Deceleration)

表格 10-9 运动模式的状态

状态	说明
定位	轴在执行运动控制指令“MC_MoveAbsolute”或“MC_MoveRelative”或者控制面板的定位任务。 (工艺对象的变量: <轴名称>.StatusBits.PositioningCommand)
速度命令	轴在以运动控制指令“MC_MoveVelocity”或“MC_MoveJog”或者控制面板的设置速度执行任务。 (工艺对象的变量: <轴名称>.StatusBits.SpeedCommand)
回原点	轴在执行运动控制指令“MC_Home”或者控制面板的回原点任务。 (工艺对象的变量: <轴名称>.StatusBits.Homing)

表格 10-10 错误位

错误	说明
到达最小软件限位	已到达下限软件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMinReached)
超出最小软件限位	已超出下限软件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMinExceeded)
到达最大软件限位	已到达上限软件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMaxReached)

错误	说明
超出最大软件限位	已超出上限软件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMaxExceeded)
负硬件限位	已逼近下限硬件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.HwLimitMin)
正硬件限位	已逼近上限硬件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.HwLimitMax)
PTO 已使用	另一个轴正在使用相同的 PTO 并且已使用“MC_Power”启用该轴。 (工艺对象的变量: <轴名称>.ErrorBits.HwUsed)
组态错误	错误地组态了“轴”工艺对象, 或者在用户程序运行期间错误地修改了可编辑的组态数据。 (工艺对象的变量: <轴名称>.ErrorBits.ConfigFault)
常规错误	发生内部错误。 (工艺对象的变量: <轴名称>.ErrorBits.SystemFault)

“运动状态”诊断功能

诊断功能“运动状态”(Motion status) 用于监视轴的运动状态。

当轴激活时, 可以在在线模式下以“手动控制”模式和“自动控制”模式显示诊断功能。

表格 10- 11 运动状态

状态	说明
目标位置	“目标位置”(Target position) 字段指示运动控制指令“MC_MoveAbsolute”或“MC_MoveRelative”或者控制面板的激活定位任务的当前目标位置。“目标位置”(Target position) 的值仅在定位任务执行期间有效。 (工艺对象的变量: <轴名称>.MotionStatus.TargetPosition)
当前位置	“当前位置”(Current position) 字段指示当前轴位置。如果轴未回原点, 则该值是相对于轴启用位置的位置值。 (工艺对象的变量: <轴名称>.MotionStatus.Position)
当前速度	“当前速度”(Current velocity) 字段指示轴的实际速度。 (工艺对象的变量: <轴名称>.MotionStatus.Velocity)

表格 10- 12 动态限制

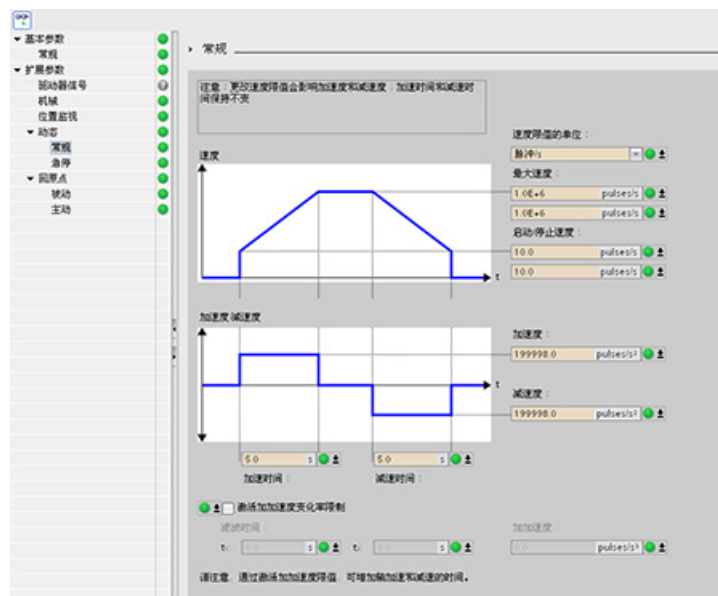
动态限制	说明
速度	“速度”(Velocity) 字段指示组态的最大轴速度。 (工艺对象的变量: <轴名称>.Config.DynamicLimits.MaxVelocity)
加速度	“加速度”(Acceleration) 字段指示当前组态的轴的加速度。 (工艺对象的变量: <轴名称>.Config.DynamicDefaults.Acceleration)
减速度	“减速度”(Deceleration) 字段指示当前组态的轴的减速度。 (工艺对象的变量: <轴名称>.Config.DynamicDefaults.Deceleration)

运动起始值控制

您可以编辑运动组态参数的实际值，以便可以在在线模式下优化过程的特性。

打开运动控制的“工艺对象”(Technology objects) 及其“组态”(Configuration) 对象。

要访问起始值控制，单击此对话框左上角的“眼镜图标”：



10.3 开环运动控制

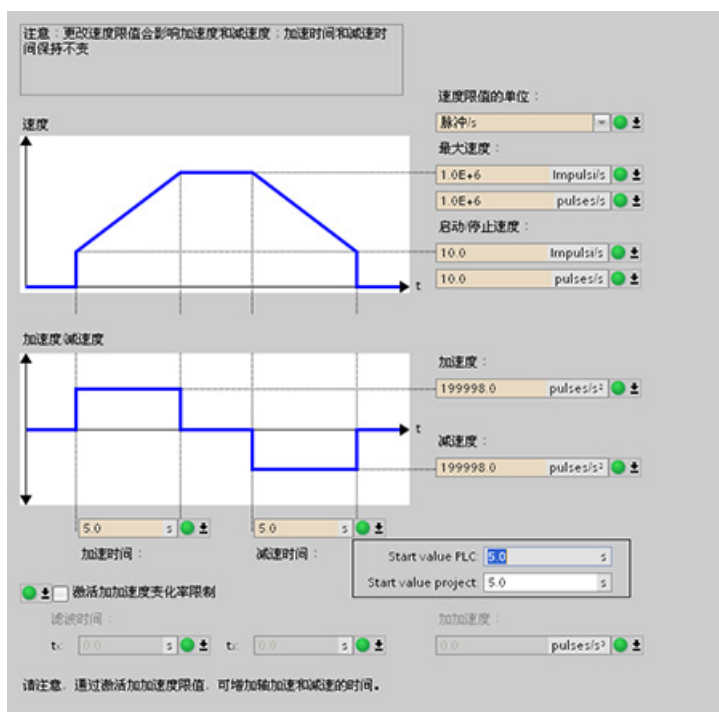
现在可以更改运动控制组态参数的任何值，如下图所示。

可以将实际值与每个参数的项目（离线）起始值和 PLC（在线）起始值进行比较。

这对于比较工艺对象数据块 (TO-DB) 的在线/离线差异以及了解在 PLC

下一次“停止到开始”转换时哪些值将用作当前值很有必要。

此外，比较图标还会通过视觉指示帮助您轻松确定在线/离线差异：



上图展示了带有比较图标的运动参数画面，其中显示了在线和离线项目之间有哪些值存在差异。绿色图标表示值相同，蓝色/橙色图标表示值不同。

另外，单击带有向下箭头的参数按钮，可打开一个显示每个参数的项目（离线）起始值和 PLC（在线）起始值的小窗口。

注意：更改速度限值会影响加速度和减速度：加速时间和减速时间保持不变。

速度

速度限值的单位：
脉冲/s

最大速度：
1.0E+6 Impuls/s
1.0E+6 pulses/s

启动/停止速度：
10.0 Impuls/s
10.0 pulses/s

加速度/减速度

加速度：
199998.0 pulses/s²

减速度：
199998.0 pulses/s²

加速时间：
5.0 s

减速时间：
5.0 s

Start value PLC: 5.0 s
Start value project: 5.1 s

加加速度：
0.0 pulses/s³

激活加速度变化率限制

滤波时间：
tc: 0.0 s

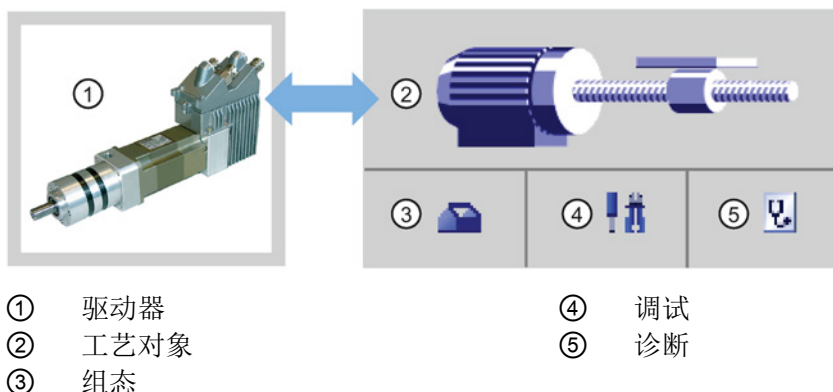
请注意，通过激活加速度限值，可增加加速和减速的时间。

10.4 闭环运动控制

10.4.1 组态轴

通过模拟驱动器或 PROFIdrive 在 PLC 和驱动器上连接闭环轴。闭环轴也需要编码器。

STEP 7 为“轴”工艺对象提供组态工具、调试工具和诊断工具。



表格 10- 13 用于闭环运动控制的 STEP 7 工具

工具	说明
组态	<p>组态“轴”工艺对象的下列属性：</p> <ul style="list-style-type: none"> • 要使用的模拟驱动器接口或 PROFIdrive 的选择以及驱动器和编码器接口的组态 • 机械的属性和驱动器与编码器（机器或系统）的传动比参数 • 位置限制属性、动态属性和归位属性 <p>在工艺对象的数据块中保存组态数据。</p>
调试	<p>无需创建用户程序即可测试轴的功能。启动该工具时，将显示控制面板。控制面板上提供了下列命令：</p> <ul style="list-style-type: none"> • 启用和禁用轴 • 在点动模式下移动轴 • 以绝对和相对方式定位轴 • 使轴归位 • 确认错误信息 <p>可以为运动命令指定速度以及加速度/减速度。控制面板中还将显示当前的轴状态。</p>
诊断	<p>监视轴和驱动器的当前状态和错误信息。</p>

说明

在用户程序中可以根据新量纲单位调整运动控制指令的输入参数值。

为轴创建工艺对象之后，通过定义基本参数（模拟驱动器或 PROFIdrive 连接和驱动器及编码器的组态）来组态该轴。

▼ Parametri di base	✓
Generale	✓
Azionamento	✓
Encoder	✓
▼ Parametri avanzati	✓
Meccanica	✓
Modulo	✓
Limiti di posizione	✓
▼ Dinamica	✓
Generale	✓
Arresto di emergenza	✓
▼ Ricerca del punto di riferimento	✓
Attivo	✓
Passivo	✓
▼ Controllo posizionamento	✓
Controllo posizionamento	✓
Errore di inseguimento	✓
Segnale di fermo	✓
Circuito di regolazione	✓

模拟驱动器或 PROFIdrive

连接的树选择器包括编码器、模数、位置监视和控制回路组态菜单。

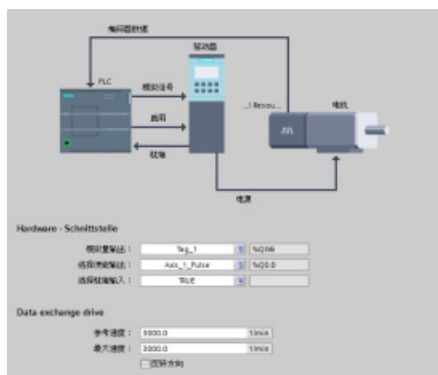
模拟驱动器接口组态



在“常规”(General)

组态对话框中，选择下列参数：

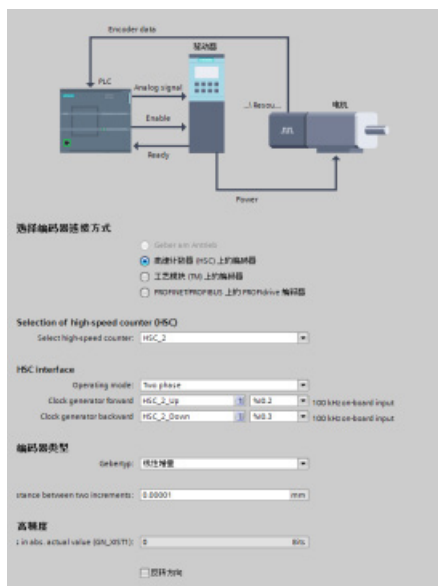
- “模拟驱动器接口”(Analog drive connection) 单选按钮
- 计量单位



在“驱动器”(Drive)

组态对话框中，选择下列参数：

- 模拟驱动器硬件输出
- 数据交换驱动器速度



在“编码器”(Encoder)

组态对话框中，选择下列参数：

- 模拟驱动器编码器耦合（例如，高速计数器 (HSC)）
- HSC 接口
- 编码器类型
- 高精度

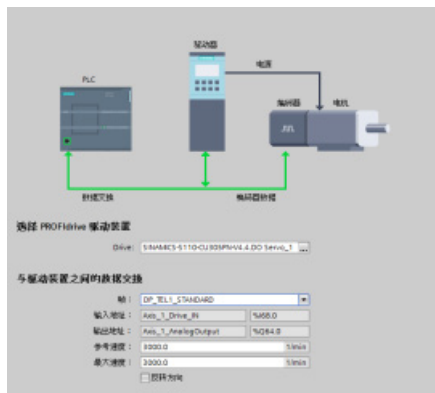
PROFIdrive 组态



在“常规”(General)

组态对话框中，选择下列参数：

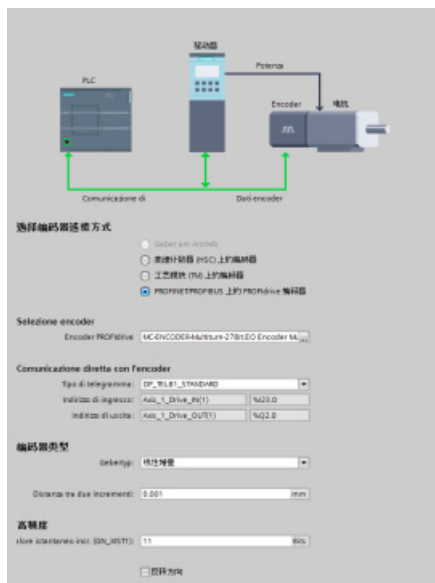
- “PROFIdrive”单选按钮
- 计量单位



在“驱动器”(Drive)

组态对话框中，选择下列参数：

- PROFIdrive 驱动器
- 与驱动器之间的数据交换



在“编码器”(Encoder)

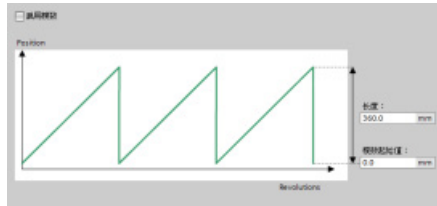
组态对话框中，选择下列参数：

- PROFIdrive 编码器耦合（例如，PROFINET 上的 PROFIdrive 编码器）
- PROFIdrive 编码器
- 与编码器之间的数据交换
- 编码器类型
- 高精度

扩展参数

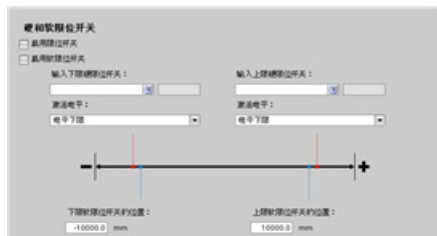
还可以组态闭环轴的以下属性：

- 模数
- 位置限制
- 动态
- 归位
- 位置监视
- 跟随误差
- 停止信号
- 控制回路



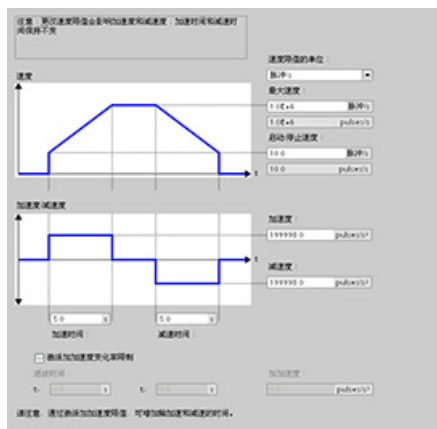
模数：

可以组态“模数”轴在循环区域中移动负载，该区域有一个起始值/起始位置和一个给定的长度。如果负载位置达到此区域的终点，则会自动重新设置为起始值。在选中“启用模数”(Enable Modulo) 复选框时，启用“长度”(Length) 和“模数起始值”(Modulo start value) 字段。

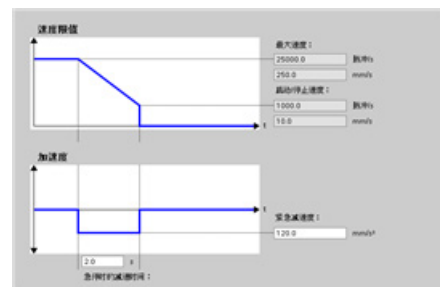


位置限制：

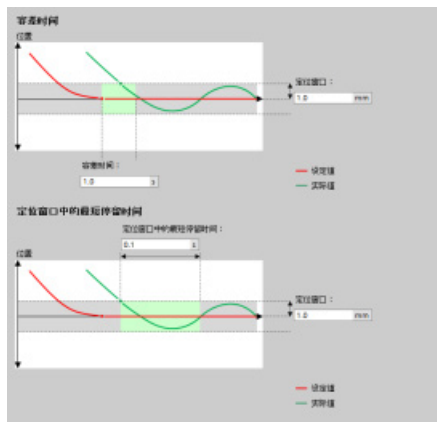
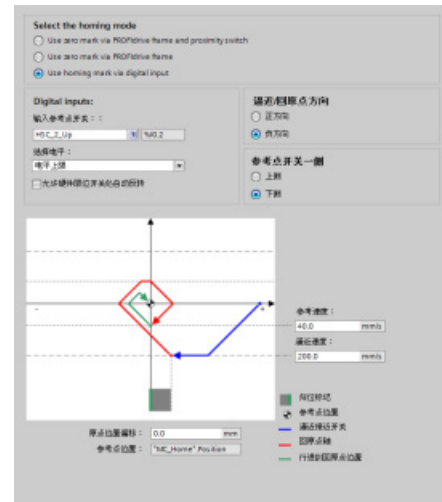
可以组态驱动器信号、驱动器机械装置和位置监视（硬限位开关和软限位开关）的属性。



动态： 可以组态急停命令的运动动态和行为。



归位：还可以组态归位行为（被动和主动）。

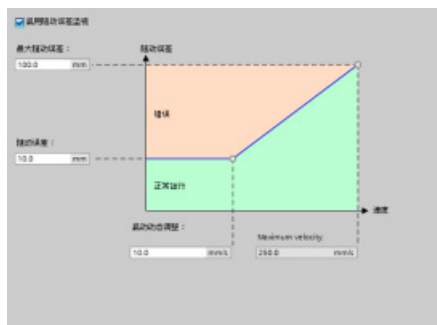


“位置监视”：

可以为定位窗口组态容差时间以及最短停留时间。

系统将以下三个参数直接与轴 TO-DB 相连：

- 定位窗口
- 容差时间
- 在定位窗口停留的最短时间

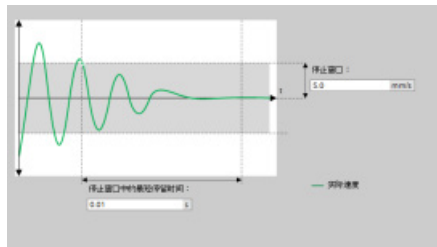


“跟随误差”：

可以组态在特定速度范围内允许的距离误差。

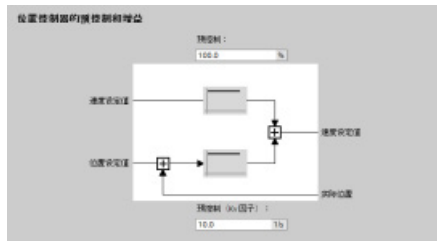
可选中“启用跟随误差监控”复选框来激活跟随误差。用户可以组态下列参数：

- 最大跟随误差
- 跟随误差
- 启动动态调整
- 最大速度



“停止信号”：用户可以组态下列参数：

- 在停止窗口停留的最短时间
- 停止窗口。



“控制回路”：可以组态被称为“预控制（Kv 因子）”的速度增益。

使用“调试”(Commissioning) 控制面板独立于用户程序对功能进行测试。

 单击“启动”(Startup) 图标对轴进行调试。

控制面板会显示轴的当前状态。

不仅可以启用和禁用轴，还可以测试轴定位（以绝对和相对方式）以及指定速度、加速度和减速度。还可以测试归位和点动任务。控制面板还可用于确认错误。

10.4.2 调试

“状态和错误位”诊断功能

诊断功能“状态和错误位”(Status and error bits) 用于监视轴的最重要状态和错误消息。

当轴激活时，可以在在线模式下以“手动控制”模式和“自动控制”模式显示诊断功能。

表格 10- 14 轴的状态

状态	说明
启用	轴已启用且准备好接受运动控制任务的控制。 (工艺对象的变量: <轴名称>.StatusBits.Enable)
已回原点	轴已回原点且能够执行运动控制指令“MC_MoveAbsolute”的绝对定位任务。 对于相对回原点而言，轴不必回原点。特殊情况： <ul style="list-style-type: none"> • 主动回原点期间，该状态为 FALSE。 • 如果回原点的轴经受被动回原点，则在被动回原点期间该状态设置为 TURE。 (工艺对象的变量: <轴名称>.StatusBits.HomingDone)

状态	说明
错误	<p>“轴”工艺对象发生错误。</p> <p>有关错误的更多信息，请参见自动控制模式下的运动控制指令的 ErrorID 和 ErrorInfo 参数。在手动模式下，控制面板中的“上一错误”(Last error) 字段显示更多错误原因信息。</p> <p>(工艺对象的变量: <轴名称>.StatusBits.Error)</p>
控制面板激活	<p>在控制面板中启用了“手动控制”模式。控制面板对“轴”工艺对象具有优先控制权。不能通过用户程序来控制轴。</p> <p>(工艺对象的变量: <轴名称>.StatusBits.ControlPanelActive)</p>

表格 10- 15 驱动器状态

状态	说明
驱动器准备就绪	<p>驱动器准备好运行。</p> <p>(工艺对象的变量: <轴名称>.StatusBits.DriveReady)</p>
错误	<p>驱动器在其准备就绪信号故障后报告了错误。</p> <p>(工艺对象的变量: <轴名称>.ErrorBits.DriveFault)</p>

表格 10- 16 轴运动的状态

状态	说明
停止	<p>轴处于停止状态。</p> <p>(工艺对象的变量: <轴名称>.StatusBits.StandStill)</p>
加速	<p>轴在加速。</p> <p>(工艺对象的变量: <轴名称>.StatusBits.Acceleration)</p>
恒速	<p>轴在恒速运转。</p> <p>(工艺对象的变量: <轴名称>.StatusBits.ConstantVelocity)</p>
减速	<p>轴在减速 (速度下降)。</p> <p>(工艺对象的变量: <轴名称>.StatusBits.Deceleration)</p>

表格 10- 17 运动模式的状态

状态	说明
定位	轴在执行运动控制指令“MC_MoveAbsolute”或“MC_MoveRelative”或者控制面板的定位任务。 (工艺对象的变量: <轴名称>.StatusBits.PositioningCommand)
速度命令	轴在以运动控制指令“MC_MoveVelocity”或“MC_MoveJog”或者控制面板的设置速度执行任务。 (工艺对象的变量: <轴名称>.StatusBits.SpeedCommand)
回原点	轴在执行运动控制指令“MC_Home”或者控制面板的回原点任务。 (工艺对象的变量: <轴名称>.StatusBits.Homing)

表格 10- 18 错误位

错误	说明
到达最小软件限位	已到达下限软件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMinReached)
超出最小软件限位	已超出下限软件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMinExceeded)
到达最大软件限位	已到达上限软件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMaxReached)
超出最大软件限位	已超出上限软件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.SwLimitMaxExceeded)
负硬件限位	已逼近下限硬件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.HwLimitMin)
正硬件限位	已逼近上限硬件限位开关。 (工艺对象的变量: <轴名称>.ErrorBits.HwLimitMax)
PTO 已使用	另一个轴正在使用相同的 PTO 并且已使用“MC_Power”启用该轴。 (工艺对象的变量: <轴名称>.ErrorBits.HwUsed)

错误	说明
组态错误	错误地组态了“轴”工艺对象，或者在用户程序运行期间错误地修改了可编辑的组态数据。 (工艺对象的变量: <轴名称>.ErrorBits.ConfigFault)
常规错误	发生内部错误。 (工艺对象的变量: <轴名称>.ErrorBits.SystemFault)

“运动状态”诊断功能

诊断功能“运动状态”(Motion status) 用于监视轴的运动状态。

当轴激活时，可以在在线模式下以“手动控制”模式和“自动控制”模式显示诊断功能。

表格 10- 19 运动状态

状态	说明
目标位置	“目标位置”(Target position) 字段指示运动控制指令“MC_MoveAbsolute”或“MC_MoveRelative”或者控制面板的激活定位任务的当前目标位置。“目标位置”(Target position) 的值仅在定位任务执行期间有效。 (工艺对象的变量: <轴名称>.MotionStatus.TargetPosition)
当前位置	“当前位置”(Current position) 字段指示当前轴位置。 如果轴未回原点，则该值是相对于轴启用位置的位置值。 (工艺对象的变量: <轴名称>.MotionStatus.Position)
当前速度	“当前速度”(Current velocity) 字段指示轴的实际速度。 (工艺对象的变量: <轴名称>.MotionStatus.Velocity)

表格 10- 20 动态限制

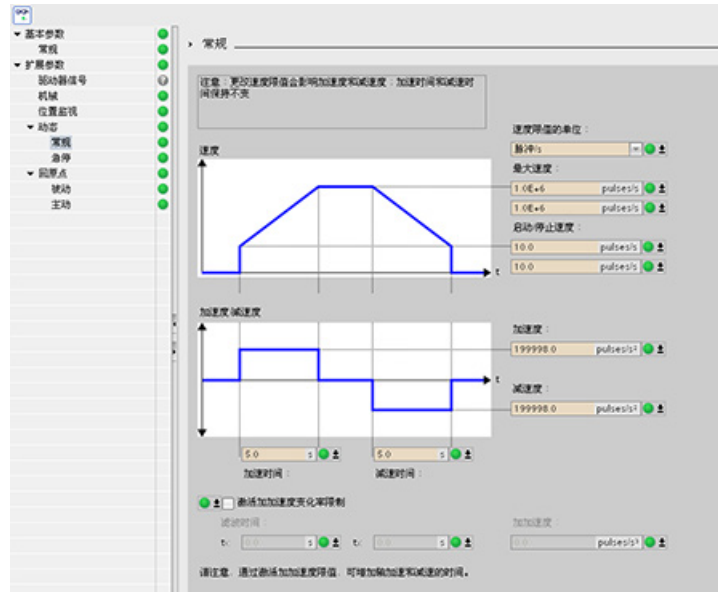
动态限制	说明
速度	“速度”(Velocity) 字段指示组态的最大轴速度。 (工艺对象的变量: <轴名称>.Config.DynamicLimits.MaxVelocity)
加速度	“加速度”(Acceleration) 字段指示当前组态的轴的加速度。 (工艺对象的变量: <轴名称>.Config.DynamicDefaults.Acceleration)
减速度	“减速度”(Deceleration) 字段指示当前组态的轴的减速度。 (工艺对象的变量: <轴名称>.Config.DynamicDefaults.Deceleration)

运动起始值控制

您可以编辑运动组态参数的实际值，以便可以在在线模式下优化过程的特性。

打开运动控制的“工艺对象”(Technology objects) 及其“组态”(Configuration) 对象。

要访问起始值控制，单击此对话框左上角的“眼镜图标”：



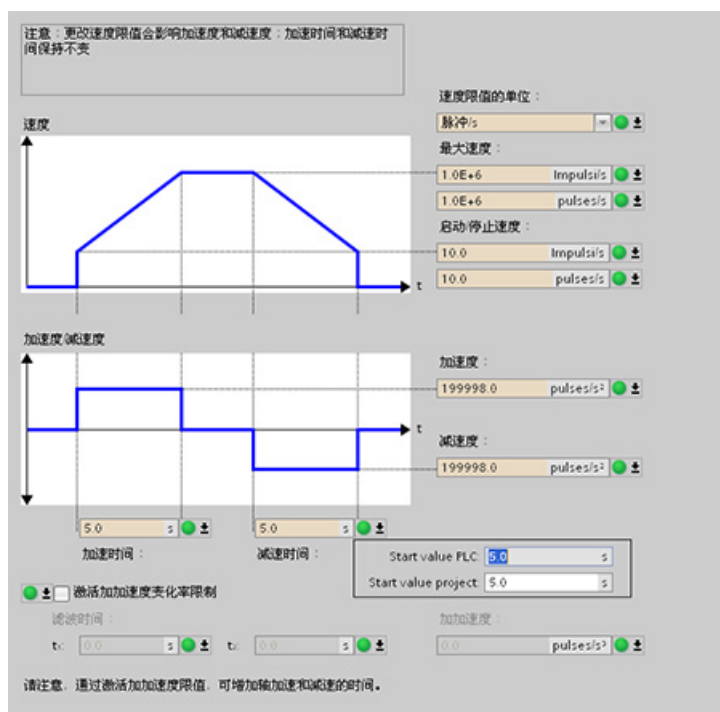
现在可以更改运动控制组态参数的任何值，如下图所示。

可以将实际值与每个参数的项目（离线）起始值和 PLC（在线）起始值进行比较。

这对于比较工艺对象数据块 (TO-DB) 的在线/离线差异以及了解在 PLC

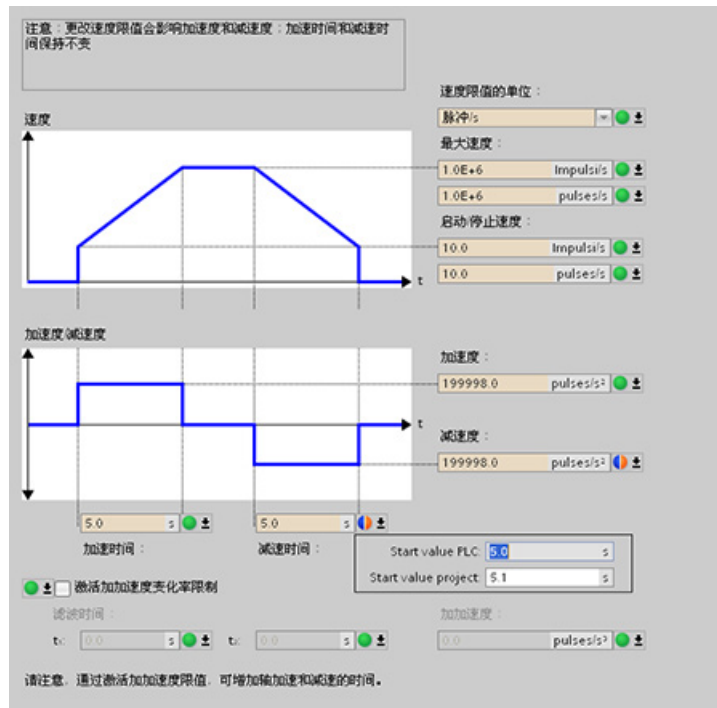
下一次“停止到开始”转换时哪些值将用作当前值很有必要。

此外，比较图标还会通过视觉指示帮助您轻松确定在线/离线差异：



上图展示了带有比较图标的运动参数画面，其中显示了在线和离线项目之间有哪些值存在差异。绿色图标表示值相同，蓝色/橙色图标表示值不同。

另外，单击带有向下箭头的参数按钮，可打开一个显示每个参数的项目（离线）起始值和 PLC（在线）起始值的小窗口。

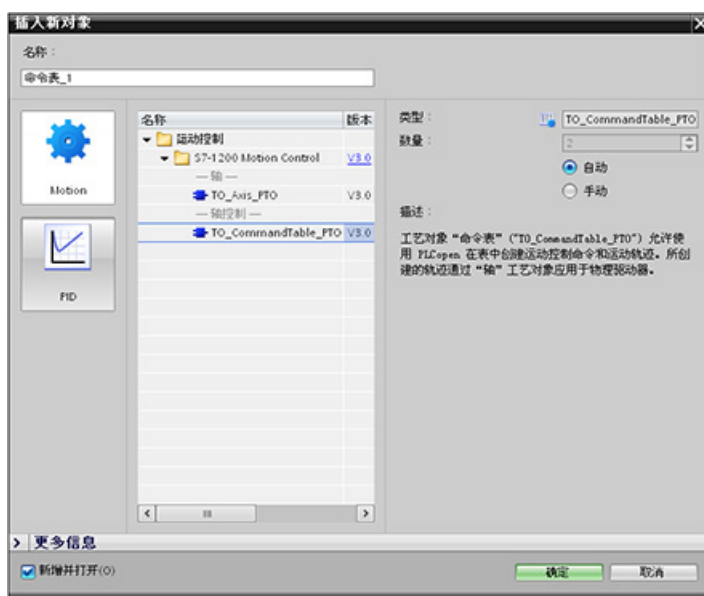


10.5 组态 TO_CommandTable_PTO

可以使用工艺对象组态 MC_CommandTable 指令。以下示例将演示如何操作。

添加工艺对象

1. 在项目树中，展开节点“工艺对象”(Technology Objects)，然后选择“添加新对象”(Add new object)。
2. 选择“CommandTable”图标（必要时可以重命名），然后单击“确定”(OK) 打开 CommandTable 对象的组态编辑器。



为应用规划步

可在“命令表”(Command Table)

组态窗口中创建所需的运动序列，并根据趋势图中的图形视图来检查结果。

可选择要用于处理命令表的命令类型。最多可输入 32 个步。

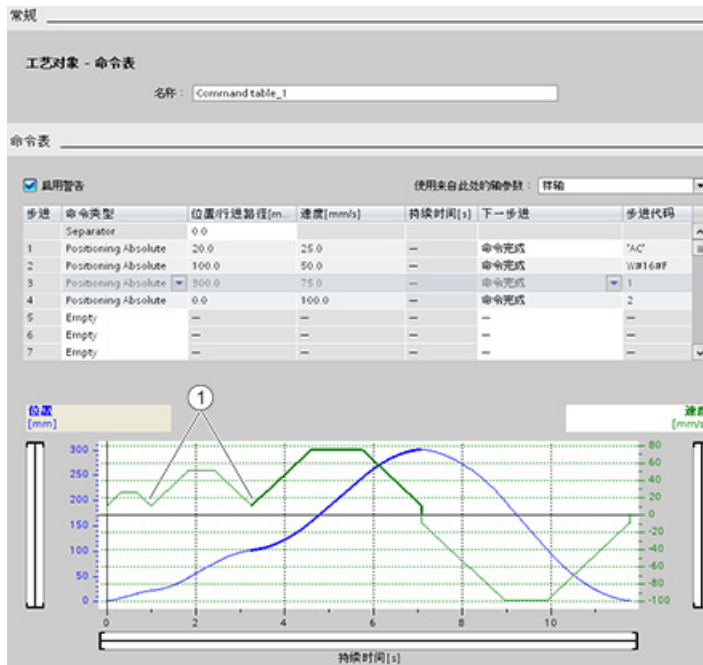
按顺序处理命令，轻松生成复杂运动轨迹。

表格 10- 21 MC_CommandTable 命令类型

命令类型	说明
Empty	空白用作占位符，以便添加任意命令。 在处理命令表时，忽略空白条目。
Halt	暂停轴。 注： 该命令仅在“Velocity setpoint”命令之后使用。
Positioning Relative	根据距离定位轴。该命令将按给定的距离和速度移动轴。
Positioning Absolute	根据位置定位轴。该命令以指定的速度将轴移到给定位置。
Velocity setpoint	按给定速度移动轴。
Wait	等待给定期间结束。“Wait”不会停止已激活的行进运动。
Separator	在选定行上方添加“分隔”线。 利用分隔线，可在单个命令表中定义多个轨迹。

10.5 组态 TO_CommandTable_PTO

在下图中，“命令完成”(Command complete) 用作到下一步的切换。
该类切换允许设备减速到启动/停止速度，然后在下一步开始时重新加速。



① 轴在两步之间减速到启动/停止速度。

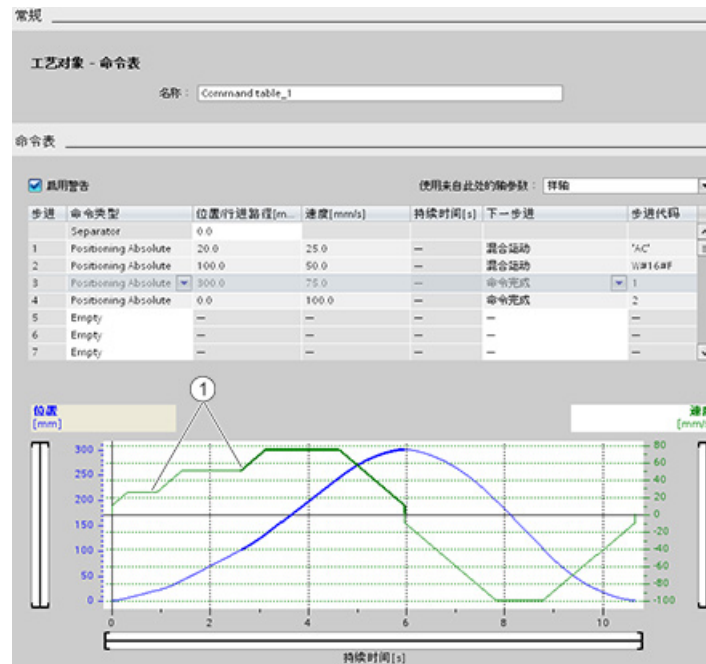
在下图中，“混合运动”(Blending motion) 用作到下一步的切换。

该类切换允许设备保持现有速度进入下一步，从而使设备平稳地从一步切换到下一步。

采用混合方式可缩短完全执行某轨迹所需的总时间。

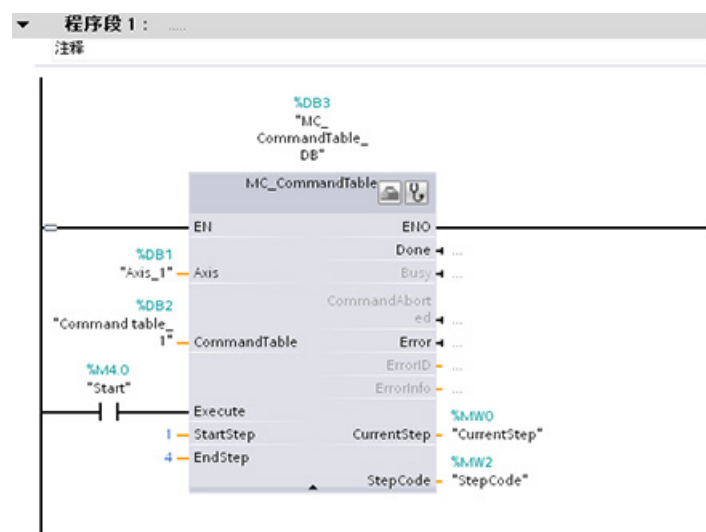
如果不采用混合方式，运行该示例将需要七秒。

如果采用混合方式，则执行时间将减少一秒，因此总时间为六秒。



① 轴继续移动，并加速或减速到下一步速度，这会节省时间和减少机械磨损。

CommandTable 的运行受 MC_CommandTable 指令控制，如下所示：



10.6 S7-1200 的运动控制操作

10.6.1 用于运动控制的 CPU 输出

CPU 提供四个脉冲输出发生器。

每个脉冲输出发生器提供一个脉冲输出和一个方向输出，用于通过脉冲接口对步进电机驱动器或伺服电机驱动器进行控制。脉冲输出为驱动器提供电机运动所需的脉冲。方向输出则用于控制驱动器的行进方向。

PTO 输出生成频率可变的方波输出。脉冲发生由通过 H/W 组态和/或 SFC/SFB 提供的组态和执行信息来控制。

在 CPU 处于 RUN

模式下时，根据用户的选择，将由存储在图像寄存器中的值或者脉冲发生器的输出来驱动数字量输出。在 STOP 模式下，PTO 发生器不控制输出。

板载 CPU 输出和信号板的输出可用作脉冲和方向输出。

在设备组态期间，可以在“属性”(Properties) 选项卡的脉冲发生器 (PTO/PWM) 中，选择板载 CPU 输出或信号板输出。只有 PTO (Pulse Train Output) 适用于运动控制。

下表显示了默认 I/O 分配；但是，可将这四个脉冲发生器组态为任意数字量输出。

说明

用户程序中的其它指令无法使用脉冲串输出。

将 CPU 或信号板的输出组态为脉冲发生器时（供 PWM 或运动控制指令使用），相应的输出地址不再控制输出。

如果用户程序向用作脉冲发生器的输出写入值，则 CPU 不会将该值写入到物理输出。

说明

可以释放 PTO 方向输出以在程序中的其它位置使用。

每个 PTO 需要分配两个输出：一个作为脉冲输出，一个作为方向输出。

可以只使用脉冲输出而不使用方向输出。

随后可以释放方向输出以用于用户程序中的其它用途。不能将输出同时用于 PTO 方向输出和用户程序。

表格 10-22 脉冲和方向输出的默认地址分配

使用运动控制的输出		
	脉冲	方向
PTO1		
内置 I/O	Q0.0	Q0.1
SB I/O	Q4.0	Q4.1
PTO2		
内置 I/O	Q0.2	Q0.3
SB I/O	Q4.2 ¹	Q4.3 ¹
PTO3		
内置 I/O	Q0.4 ²	Q0.5 ²
SB I/O	Q4.0	Q4.1
PTO4		
内置 I/O	Q0.6 ³	Q0.7 ³
SB I/O	Q4.2	Q4.3

¹ 输出 Q4.2 和 Q4.3 仅在 SB1222 DQ4 上可用。

² CPU 1211C 没有输出 Q0.4、Q0.5、Q0.6 或 Q0.7。因此这些输出不能在 CPU 1211C 中使用。

³ CPU 1212C 没有输出 Q0.6 或 Q0.7。因此这些输出不能在 CPU 1212C 中使用。

⁴ 该表适用于 CPU 1211C、CPU 1212C、CPU 1214C、CPU 1215C 以及 CPU 1217C PTO 功能。

驱动器接口

对于运动控制，可以选择将驱动器接口组态为“驱动器启用”或“驱动器准备就绪”。
使用驱动器接口时，针对“驱动器启用”可选择数字量输出，针对“驱动器准备就绪”可选择数字量输入。

说明

如果已选择 PTO (Pulse Train Output)

并将其分配给某个轴，固件将通过相应的脉冲和方向输出接管控制。

在实现上述控制功能接管后，将断开过程映像和 I/O 输出间的连接。

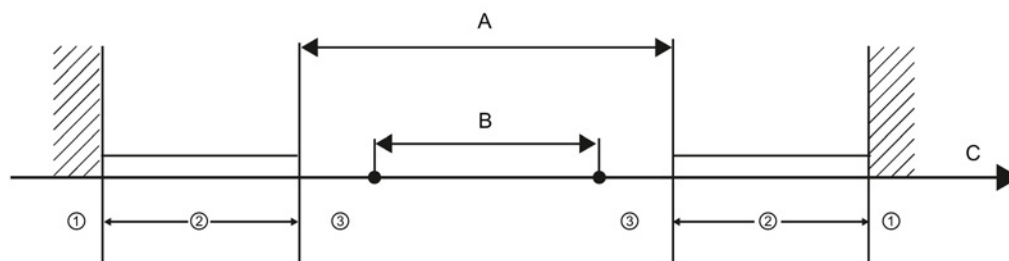
虽然用户可通过用户程序或监视表格写入脉冲输出和方向输出的过程映像，但所写入的内容不会传送到 I/O 输出。因此通过用户程序或监视表格无法监视 I/O 输出。

读取的信息只反映过程映像中的值，与 I/O 输出的实际状态并不完全一致。

对于 CPU 固件非永久使用的其它所有 CPU 输出，通常可以通过过程映像监控 I/O 输出的状态。

10.6.2 用于运动控制的硬件和软件限位开关

硬件和软件限位开关用于限制轴的“允许行程范围”和“工作范围”。



- | | | | |
|---|---------|---|----------|
| ① | 机械停止块 | A | 允许的轴行程范围 |
| ② | 硬件下限和上限 | B | 轴的工作范围 |
| ③ | 软件下限和上限 | C | 距离 |

在组态中或用户程序中使用硬件和软件限位开关之前，必须事先将其激活。
只有在轴回原点之后，才可以激活软件限位开关。

硬件限位开关

硬件限位开关确定轴的最大行程范围。硬件限位开关是物理开关元件，必须与 CPU 中具有中断功能的输入相连接。仅使用逼近后始终保持切换的硬件限位开关。

只有在返回到允许的行程范围后，该切换状态才会发生改变。

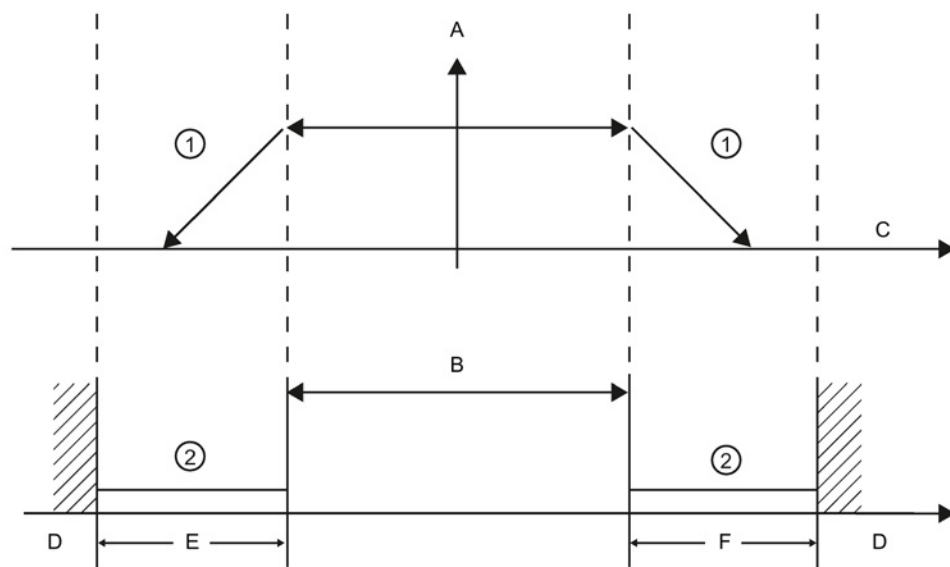
表格 10-23 硬件限值的可用输入

说明	RPS	LIM-	LIM+
内置 I/O		10.0 - 11.5	
SB I/O		14.0 - 14.3	

逼近硬件限位开关时，轴将以所组态的紧急减速度制动直到停止。

指定的紧急减速度必须足够大，才能确保在机械停止块前使轴停止。

下图显示了轴逼近硬件限位开关后的轴行为。



- ① 轴将以所组态的紧急减速度制动直到停止。
- ② 硬件限位开关产生“已逼近”状态信号的范围。
- A [速度]
- B 允许的行程范围
- C 距离
- D 机械停止块
- E 下限硬件限位开关
- F 上限硬件限位开关

**警告****对数字量输入通道的滤波时间进行更改的风险**

如果数字量输入通道的滤波时间更改自以前的设置，则新的“0”电平输入值可能需要保持长达 20.0 ms 的累积时间，然后滤波器才会完全响应新输入。

在此期间，可能不会检测到持续时间少于 20.0 ms 的短“0”脉冲事件或对其计数。

滤波时间的这种更改会引发意外的机械或过程操作，这可能会导致人员死亡、重伤和/或设备损坏。

为了确保新的滤波时间立即生效，必须关闭 CPU 电源后再开启。

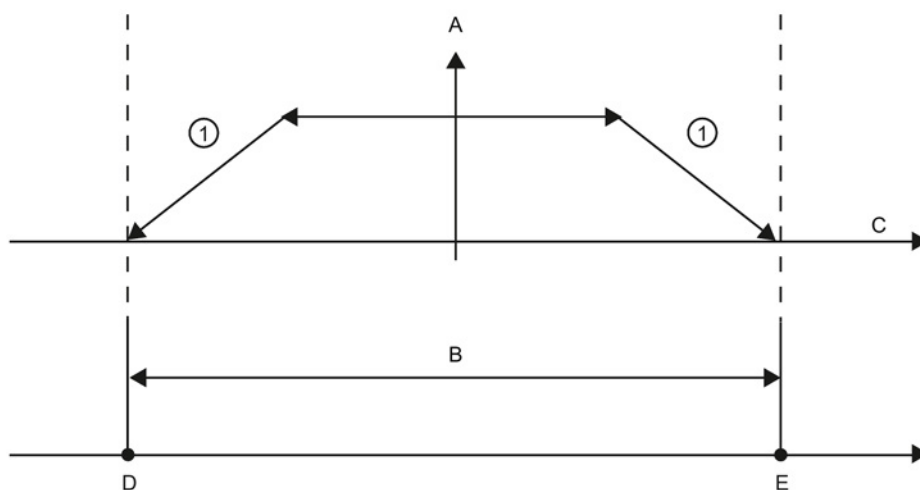
软件限位开关

软件限位开关将限制轴的“工作范围”。它们位于限制行程范围的相关硬件限位开关内。由于软件限位开关的位置可以灵活设置，所以可以根据当前的运行轨迹和具体要求来限定轴的工作范围。

与硬件限位开关不同，软件限位开关只通过软件来实现，而无需借助自身的开关元件。

如果软件限位开关激活，则在软件限位开关所在的位置将停止当前的运动。

轴将以所组态的减速度制动。下图显示了轴到达软件限位开关前的行为。



① 轴将以所组态的减速度制动直到停止。

A [速度]

B 工作范围

C 距离

D 下限软件限位开关

E 上限软件限位开关

如果机械停止块位于软件限位开关的后面并且有发生机械损坏的风险，则需要使用附加的硬件限位开关。

更多信息

用户程序可通过启用或禁用硬件和软件的限位功能来消除硬件或软件限位。可通过轴 DB 进行选择。

- 要启用或禁用硬件限位功能，请访问 DB 路径“<轴名称>/Config/PositionLimits_HW”中的“Active”变量 (Bool)。“Active”变量的状态可启用或禁用硬件限位的使用。
- 要启用或禁用软件限位功能，请访问 DB 路径“<轴名称>/Config/PositionLimits_SW”中的“Active”变量 (Bool)。此“Active”变量的状态可启用或禁用软件限位。

还可以利用用户程序修改软件限位（例如，提高机器设置的灵活性或缩短机器转换时间）。用户程序可以将新值写入 DB 路径“<轴名称>/Config/PositionLimits_SW”的“MinPosition”和“MaxPosition”变量中（采用 Real 格式的工程单位）。

10.6.3 回原点

10.6.3.1 使轴回原点

回原点是指轴坐标与实际的物理驱动器位置匹配。（如果驱动器当前位于位置 x ，则轴将被调整为位于位置 x 。）对于位置受控制的轴，位置输入与显示值指的就是这些轴坐标。

说明

轴坐标必需与实际情形相一致。

如果要确保通过驱动器也能准确到达轴的绝对目标位置，上述步骤必不可缺。

MC_Home 指令可启动轴的回原点操作。

有 4 种不同的回原点功能。

前两种功能允许用户设置轴的当前位置，后两种功能可相对于回原点参考传感器放置轴。

- **模式 0 - 绝对式直接参考：** 指令执行时，此模式将告知轴它的确切位置。

该模式将内部位置变量设置为回原点指令的 **Position** 输入的值。

此模式用于机器校准和设置。

轴位置的设置与参考点开关无关。也不会终止当前的行进运动。 **MC_Home** 指令的 **Position** 输入参数的值将被立即设置为轴的参考点。

要将参考点分配给具体的机械位置，在执行回原点操作时轴必须停止在该位置。

- **模式 1 - 相对式直接参考：**

指令执行时，该模式将使用内部位置变量并加上回原点指令的 **Position** 输入的值。

考虑到机器偏移时通常使用此模式。

轴位置的设置与参考点开关无关。也不会终止当前的行进运动。

以下语句适用于回到原点后的定位： $\text{新轴位置} = \text{当前轴位置} + \text{MC_Home 指令的 Position 参数的值}$ 。

- 模式 2 - 被动参考:

当轴在移动的过程中经过参考点开关时，当前位置将设置为回原点位置。
此功能有助于应对正常的机器磨损和齿轮间隙，从而无需对磨损进行手动补偿。

如前所述，回原点指令的 **Position**

输入将添加到参考点开关指示的位置，从而可轻松补偿回原点位置。

在被动回原点期间，指令 **MC_Home** 不会执行任何回原点运动。

用户必须通过其它运动控制指令来执行该步骤所需的行进运动。

检测到参考点开关时，将根据组态使轴回到原点。

被动回原点启动时，不会中止当前的行进运动。

- 模式 3 - 主动参考: 此模式是最精确的使轴回原点方法。

运动的初始方向和速度在工艺对象组态扩展参数 **Homing** 中进行组态。

这取决于机器的配置。

还可以确定参考点开关信号的上升沿或下降沿是否是回原点位置。

几乎所有传感器都具有一个有效范围；如果“稳态开启”位置用作回原点信号，则回原点位置可能会出现错误，因为“开启”信号有效范围将覆盖距离范围。

利用该信号的上升沿或下降沿，可得到更加精确的回原点位置。

与其它所有模式一样，回原点指令的 **Position** 输入的值将被添加到硬件参考位置。

在主动回原点模式下，**MC_Home** 指令执行所需的参考点逼近。

检测到参考点开关时，将根据组态使轴回到原点。同时终止当前的行进运动。

模式 0 和模式 1 不需要移动轴。这两种模式通常用在设置和校准中。模式 2 和模式 3 需要轴运动并经过在“轴”工艺对象中组态为参考点开关的传感器。

参考点可放在轴的工作区内或放在常规工作区外、运动范围内。

10.6.3.2 回原点参数的组态

在“回原点”(Homing) 组态窗口中，组态主动和被动回原点参数。

可以使用运动控制指令中“**Mode**”输入参数设置回原点方法。其中，**Mode = 2**

表示被动回原点，**Mode = 3** 表示主动回原点。

说明

采用以下措施之一可确保机器在发生反向时不会行进到机械停止块:

- 保持较低的逼近速度
 - 增大组态的加速度/减速度
 - 增大硬件限位开关和机械停止块间的距离
-

表格 10-24 使轴回原点的参数组态

参数	说明
输入参考点开关 (主动和被动回原点)	<p>从下拉列表框中为参考点开关选择数字量输入。输入必须具有中断功能。板载 CPU 输入和所插入信号板输入都可以选作参考点开关的输入。</p> <p>数字量输入的默认滤波时间是 6.4 ms。采用数字量输入作为参考点开关的输入时，可能引起意外减速，从而导致出现误差。</p> <p>由于速度降低和参考点开关的范围的原因，可能检测不到参考点。可以在数字量输入的设备组态的“输入滤波器”(Input filter) 中设置滤波时间。</p> <p>指定的滤波时间必须小于参考点开关的输入信号的持续时间。</p>
到达硬件限位开关后自动反转 (仅限主动回原点)	<p>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。必须组态硬件限位开关并激活反向功能。</p> <p>如果在主动回原点期间到达硬件限位开关，轴将以组态的减速度减速（不是以紧急减速度），然后反向。然后反向检测参考点开关。</p> <p>如果未激活反向功能且在主动回原点期间轴到达硬件限位开关，将因错误取消参考点逼近并按紧急减速度使轴制动。</p>
逼近方向 (主动和被动回原点)	<p>通过方向选择，可以决定主动回原点期间用于搜索参考点开关的“逼近方向”以及回原点的方向。</p> <p>回原点方向将指定执行回原点操作时轴用于逼近组态的参考点开关侧的行进方向。</p>
参考点开关 (主动和被动回原点)	<ul style="list-style-type: none"> • 主动回原点：选择将在参考点开关的左侧还是右侧对轴进行参考。根据轴的起始位置和回原点参数的组态，参考点逼近顺序可能与组态窗口中图示的顺序不同。 • 被动回原点： <ul style="list-style-type: none"> 对于被动回原点，必须由用户通过运动命令来执行回原点的行进运动。回原点发生在参考点开关的哪一侧取决于以下因素： <ul style="list-style-type: none"> - “逼近方向”组态 - “参考点开关”组态 - 被动回原点期间的当前行进方向
逼近速度 (仅限主动回原点)	<p>指定参考点逼近期间搜索参考点开关的速度。</p> <p>限值（与所选的用户单位无关）： 启动/停止速度 ≤ 逼近速度 ≤ 最大速度</p>

参数	说明
减小的速度 (仅限主动回原点)	指定轴逼近回原点参考点开关的速度。 限值(与所选的用户单位无关): 启动/停止速度 ≤ 减小的速度 ≤ 最大速度
回原点位置偏移 (仅限主动回原点)	如果期望的参考点与参考点开关的位置有偏移, 则可在该字段中指定回原点位置偏移。 如果值不等于 0, 轴回到参考点开关位置后将执行以下动作: 1. 以减小的速度使轴移动回原点位置偏移值。 2. 到达回原点位置偏移的位置后, 将该轴位置设置为绝对参考位置。 并通过运动控制指令“MC_Home”的参数“Position”, 指定该绝对参考位置。 限值(与所选的用户单位无关): -1.0e12 ≤ 回原点位置偏移 ≤ 1.0e12

表格 10-25 影响回原点的因素

影响因素:			结果:
组态 逼近方向	组态 参考点开关	当前的行进方向	回原点发生在 参考点开关
正方向	“左(负)侧”	正方向	左
		负方向	右侧
正方向	“右(正)侧”	正方向	右侧
		负方向	左
负方向	“左(负)侧”	正方向	右侧
		负方向	左
负方向	“右(正)侧”	正方向	左
		负方向	右侧

10.6.3.3 主动回原点的顺序

使用运动控制指令“MC_Home”（输入参数 Mode = 3），可以启动主动回原点。
 在这种情况下，可以通过输入参数“Position”来指定绝对参考点的坐标。
 也可以在控制面板上启动主动回原点，以便进行测试。

下图举例说明了使用以下组态参数时主动参考点逼近的特征曲线：

- “逼近方向”=“正方向逼近”
- “参考点开关”=“右（正）侧”
- “回原点位置偏移”值 > 0

表格 10- 26 MC 归位的速度特性曲线

操作		注意	
		A	逼近速度
		B	减小的速度
		C	归位位置坐标
		D	回原点位置偏移
①	搜索阶段（蓝色曲线段）： 主动回原点开始时，轴加速到组态的“逼近速度”并以该速度搜索参考点开关。		
②	参考点逼近（红色曲线段）： 检测到参考点开关时，本示例中的轴将制动并反向，以“减小的速度”在组态的参考点开关侧回原点。		
③	行进到参考点位置（绿色曲线段）： 轴回原点到参考点开关位置后，轴将以“减小的速度”行进到“参考点坐标”。 到达“参考点坐标”时，轴将立即停止在指令 MC_Home 的 Position 输入参数中指定的位置值处。		

说明

如果回原点搜索没有按照预期那样运行，请检查分配给硬件限位或参考点的输入。可能已经在设备配置中禁用了这些输入的沿中断。

请检查相关轴工艺对象的组态数据，以查看为“HW Low Limit Switch Input”、“HW High Limit Switch Input”和“Input reference point switch”分配了哪些输入（如果有）。

然后打开 CPU 的设备配置，检查所分配的每个输入。

确认是否选择了“启用上升沿检测”(Enable rising edge detection)

和“启用下降沿检测”(Enable falling edge detection)。

如果未选择这些属性，请删除轴组态中指定的输入，然后再次选择这些属性。

10.7 运动控制指令

10.7.1 MC 指令概述

运动控制指令使用相关工艺数据块和 CPU 的专用 PTO（脉冲串输出）来控制轴上的运动。

- MC_Power (页 334) 可启用和禁用运动控制轴。
- MC_Reset (页 337) 可复位所有运动控制错误。
所有可确认的运动控制错误都会被确认。
- MC_Home (页 339) 可建立轴控制程序与轴机械定位系统之间的关系。
- MC_Halt (页 342) 可取消所有运动过程并使轴停止运动。停止位置未定义。
- MC_MoveAbsolute (页 344) 可启动到某个绝对位置的运动。
达到目标位置后该作业结束。
- MC_MoveRelative (页 346) 可启动相对于起始位置的定位运动。
- MC_MoveVelocity (页 348) 可使轴以指定的速度行进。
- MC_MoveJog (页 351) 可执行用于测试和启动目的的点动模式。
- MC_CommandTable (页 353) 用于将轴命令作为一个运动序列运行。
- MC_ChangeDynamic (页 356) 用于更改轴的动态设置。
- MC_WriteParam (页 358) 用于写入选定数量的参数来通过用户程序更改轴功能。
- MC_ReadParam (页 361)
用于读取选定数量的参数，以指示在轴输入中定义的轴的当前位置、速度等。

CPU 固件级别

如果具有固件版本为 V4.1 的 S7-1200 CPU，则选择 V5.0 版本的各个运动指令。

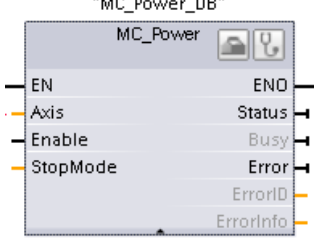
如果具有固件版本为 V4.0 或更早版本的 S7-1200 CPU，则为每个运动指令选择适用的 V4.0、V3.0、V2.0 或 V1.0 版本。

10.7.2 MC_Power（发布/阻止轴）指令

说明

如果由于错误而将轴关闭，则在消除并确认错误后会自动再次将其启用。这要求输入参数 **Enable** 的值在该过程中保持为 **TRUE**。

表格 10-27 MC_Power 指令

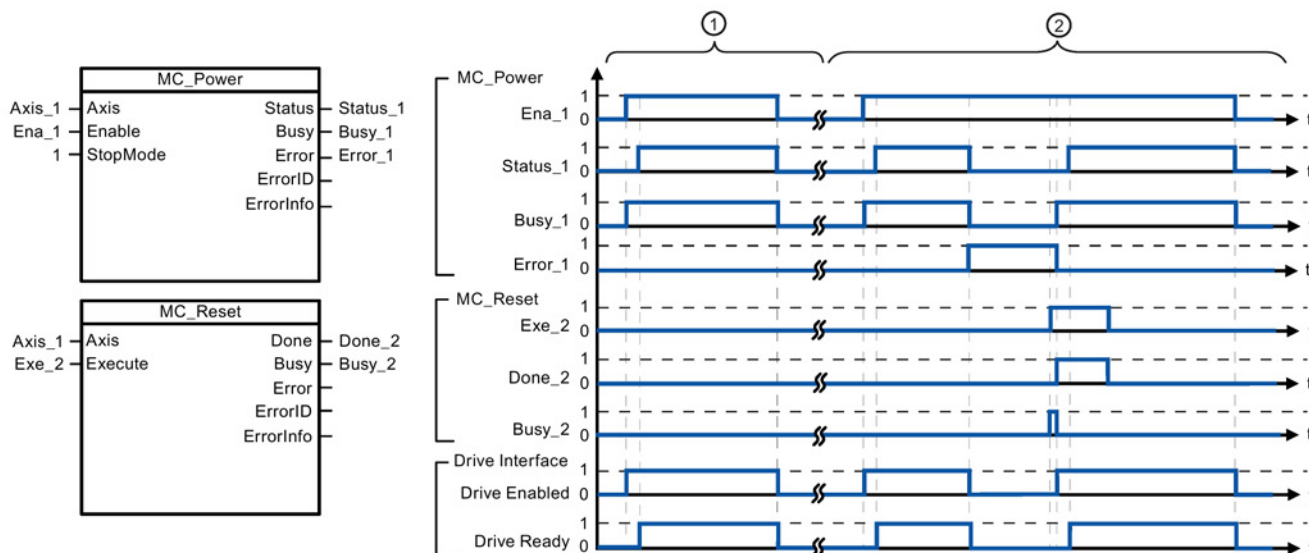
LAD/FBD	SCL	说明
	<pre>"MC_Power_DB" (Axis:= _multi_fb_in_, Enable:= _bool_in_, StopMode:= _int_in_, Status=> _bool_out_, Busy=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>MC_Power 运动控制指令可启用或禁用轴。在启用或禁用轴之前，应确保以下条件：</p> <ul style="list-style-type: none"> • 已正确组态工艺对象。 • 没有未决的启用-禁止错误。 <p>运动控制任务无法中止 MC_Power 的执行。禁用轴（输入参数 Enable = FALSE）将中止相关工艺对象的所有运动控制任务。</p>

- 1 STEP 7 会在插入指令时自动创建 DB。
- 2 在 SCL 示例中，“MC_Power_DB”是背景 DB 的名称。

表格 10- 28 MC_Power 指令的参数

参数和类型		数据类型	说明
Axis	IN	TO_Axis_1	轴工艺对象
Enable	IN	Bool	<ul style="list-style-type: none"> • FALSE（默认）： 所有激活的任务都将按照参数化的“StopMode”而中止，并且轴也会停止。 • TRUE：运动控制尝试启用轴。
StopMode	IN	Int	<ul style="list-style-type: none"> • 0: 急停： 如果禁用轴的请求未决，则轴将以组态的紧急减速度制动。轴在达到停止后被禁用。 • 1: 立即停止： 如果禁用轴的请求未决，该轴将在不减速的情况下被禁用。脉冲输出立即停止。 • 2: 通过冲击控制进行急停： 如果禁用轴的请求未决，则轴将以组态的急停减速度制动。如果激活了冲击控制，则不考虑组态的冲击。轴在达到停止后被禁用。
Status	OUT	Bool	<p>轴使能的状态：</p> <ul style="list-style-type: none"> • FALSE：轴已禁用： <ul style="list-style-type: none"> - 轴不会执行运动控制任务并且不接受任何新任务（例外：MC_Reset 任务）。 - 轴未回原点。 - 禁用时，直到轴达到停止状态，状态才会更改为 FALSE。 • TRUE：轴已启用： <ul style="list-style-type: none"> - 轴已准备好执行运动控制任务。 - 轴启用时，直到信号“驱动器就绪”(Drive ready) 进入未决，状态才会更改为 TRUE。 如果在轴组态中未组态“驱动器就绪”(Drive ready) 驱动器接口，状态会立即更改为 TRUE。
Busy	OUT	Bool	<p>FALSE：MC_Power 未激活。 TRUE：MC_Power 处于活动状态</p>
Error	OUT	Bool	<p>FALSE：无错误 TRUE：运动控制指令“MC_Power”或关联的工艺对象出错。出错原因可在“ErrorID”和“ErrorInfo”参数中找到。</p>

参数和类型		数据类型	说明
ErrorID	OUT	Word	参数“Error”的错误 ID
ErrorInfo	OUT	Word	参数“ErrorID”的错误信息 ID



- ① 启用轴，然后再次禁用轴。驱动器向 CPU 返回“驱动器就绪”(Drive ready) 信号后，可通过“Status_1”读出成功启用信息。
- ② 启用轴后，出现了导致轴被禁用的错误。该错误被消除并通过“MC_Reset”进行确认。然后再次启用该轴。

要启用组态了驱动器接口的轴，请按以下步骤操作：

1. 检查上文所述的要求。
2. 使用所需值初始化输入参数“StopMode”。将输入参数“Enable”设置为 TRUE。

“驱动器已启用”(Drive enabled) 的使能输出更改为 TRUE 以启用驱动器的电源。CPU 等待驱动器的“驱动器就绪”(Drive ready) 信号。

当“驱动器就绪”(Drive ready) 信号出现在 CPU 的已组态就绪输入中时，轴将变为启用状态。输出参数“Status”和工艺对象变量 <轴名称>.StatusBits.Enable 指示值 TRUE。

要启用未组态驱动器接口的轴，请按以下步骤操作：

1. 检查上文所述的要求。
2. 使用所需值初始化输入参数“StopMode”。将输入参数“Enable”设置为 TRUE。
轴已启用。输出参数“Status”和工艺对象变量 <轴名称>.StatusBits.Enable 指示值 TRUE。

要禁用轴，请按以下步骤操作：

1. 将轴切换到停止状态。


可在工艺对象变量 <轴名称>.StatusBits.StandStill 中确定轴何时处于停止状态。

2. 达到停止状态后将输入参数“Enable”设置为 FALSE。

3. 如果输出参数“Busy”和“Status”以及工艺对象变量 <轴名称>.StatusBits.Enable 指示值 FALSE，则禁用轴的操作已完成。

10.7.3 MC_Reset（确认错误）指令

表格 10-29 MC_Reset 指令

LAD/FBD	SCL	说明
	<pre>"MC_Reset_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Restart:= _bool_in_, Done=> _bool_out_, Busy=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>使用 MC_Reset 指令可确认“导致轴停止的运行错误”和“组态错误”。</p> <p>需要确认的错误可在“解决方法”下的“ErrorIDs 和 ErrorInfos 的列表”中找到。</p> <p>使用 MC_Reset 指令前，必须已将需要确认的未决组态错误的原因消除（例如，通过将“轴”工艺对象中的无效加速度值更改为有效值）。</p> <p>自 V3.0 及更高版本起，在 RUN 操作模式下，Restart 命令可将轴组态下载至工作存储器。</p>

- 1 STEP 7 会在插入指令时自动创建 DB。
- 2 在 SCL 示例中，“MC_Reset_DB”是背景 DB 的名称。

MC_Reset 任务无法被任何其它运动控制任务中止。新的 MC_Reset 任务不会中止任何其它已激活的运动控制任务。

表格 10- 30 MC_Reset 指令的参数

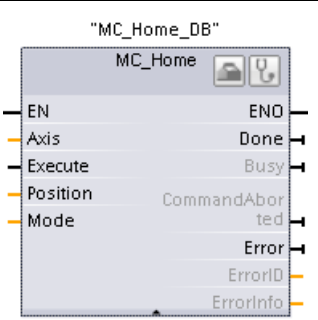
参数和类型		数据类型	说明
Axis	IN	TO_Axis_1	轴工艺对象
Execute	IN	Bool	出现上升沿时开始任务
Restart	IN	Bool	TRUE = 从装载存储器将轴组态下载至工作存储器。 只有轴处于禁用状态时才能执行该命令。
			FALSE = 确认未决错误
Done	OUT	Bool	TRUE = 错误已确认。
Busy	OUT	Bool	TRUE = 正在执行任务。
Error	OUT	Bool	TRUE = 任务执行期间出错。 出错原因可在“ErrorID”和“ErrorInfo”参数中找到。
ErrorID	OUTP	Word	参数“Error”的错误 ID
ErrorInfo	OUT	Word	参数“ErrorID”的错误信息 ID

要使用 MC_Reset 确认错误，请按以下步骤操作：

1. 检查上文所述的要求。
2. 在 Execute 输入参数出现上升沿时开始确认错误。
3. 当 Done 等于 TRUE 并且工艺对象变量 <轴名称>.StatusBits.Error 等于 FALSE 时，错误已被确认。

10.7.4 MC_Home（使轴回原点）指令

表格 10-31 MC_Home 指令

LAD/FBD	SCL	说明
	<pre>"MC_Home_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Position:=_real_in_, Mode:=_int_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>使用 MC_Home 指令可将轴坐标与实际物理驱动器位置匹配。</p> <p>轴的绝对定位需要回原点：</p> <p>为了使用 MC_Home 指令，必须先启用轴。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC_Home_DB”是背景 DB 的名称。

可使用以下类型的回原点：

- 绝对式直接回原点 (Mode = 0)：当前轴位置被设置为参数“Position”的值。
- 相对式直接回原点 (Mode = 1)：当前轴位置的偏移量为参数“Position”的值。
- 被动回原点 (Mode = 2)：在被动回原点期间，指令 MC_Home 不会执行任何回原点运动。
用户必须通过其它运动控制指令来执行该步骤所需的行进运动。
检测到参考点开关时，轴将回到原点。
- 主动回原点 (Mode = 3)：自动执行回原点步骤。

表格 10-32 MC_Home 指令的参数

参数和类型		数据类型	说明
Axis	IN	TO_Axis_PTO	轴工艺对象
Execute	IN	Bool	出现上升沿时开始任务
Position	IN	Real	<ul style="list-style-type: none"> Mode = 0、2 和 3（完成回原点操作后轴的绝对位置） Mode = 1（当前轴位置的校正值） 限值： $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$

参数和类型		数据类型	说明
Mode	IN	Int	回原点模式 <ul style="list-style-type: none"> • 0: 绝对式直接回原点 新的轴位置为参数“Position”的位置值。 • 1: 相对式直接回原点 新的轴位置为当前轴位置 + 参数“Position”的位置值。 • 2: 被动回原点 根据轴组态回原点。 回原点后，参数“Position”的值被设置为新的轴位置。 • 3: 主动回原点 按照轴组态进行参考点逼近。 回原点后，参数“Position”的值被设置为新的轴位置。
Done	OUT	Bool	TRUE = 任务完成
Busy	OUT	Bool	TRUE = 正在执行任务。
CommandAborted	OUT	Bool	TRUE = 任务在执行期间被另一任务中止。
Error	OUT	Bool	TRUE = 任务执行期间出错。 出错原因可在“ErrorID”和“ErrorInfo”参数中找到。
ErrorID	OUT	Word	参数“Error”的错误 ID
ErrorInfo	OUT	Word	参数“ErrorID”的错误信息 ID

说明

在下列情况下，轴回原点会失败：

- 通过 MC_Power 指令禁用轴
- 在自动控制和手动控制之间切换
- 主动回原点开始时（成功完成回原点操作后，可再次进行轴回原点操作。）
- 对 CPU 循环上电后
- CPU 重新启动后（RUN-to-STOP 或 STOP-to-RUN）

要使轴回原点，请按以下步骤操作：

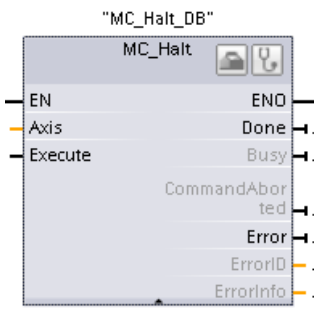
1. 检查上文所述的要求。
2. 使用相应的值初始化必要的输入参数，然后在输入参数“Execute”出现上升沿时开始回原点操作。
3. 如果输出参数“Done”和工艺对象变量 <轴名称>.StatusBits.HomingDone 指示值 TRUE，则回原点操作完成。

表格 10- 33 超驰响应

模式	说明		
0 或 1	MC_Home 任务无法被任何其它运动控制任务中止。新的 MC_Home 任务不会中止任何已激活的运动控制任务。 位置相关的运动任务在回原点后将根据新的原点位置（Position 输入参数中的值）恢复。		
2	MC_Home 任务可被下列运动控制任务中止： MC_Home 任务 Mode = 2、3：新 MC_Home 任务可中止以下已激活的运动控制任务。 MC_Home 任务 Mode = 2：位置相关的运动任务在回原点后将根据新的原点位置（Position 输入参数中的值）恢复。		
3	<table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top; width: 50%;"> MC_Home 任务可被下列运动控制任务中止： <ul style="list-style-type: none"> • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog </td> <td style="vertical-align: top; width: 50%;"> 新 MC_Home 任务可中止下列激活的运动控制任务： <ul style="list-style-type: none"> • MC_Home 模式 = 2、3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog </td> </tr> </table>	MC_Home 任务可被下列运动控制任务中止： <ul style="list-style-type: none"> • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog 	新 MC_Home 任务可中止下列激活的运动控制任务： <ul style="list-style-type: none"> • MC_Home 模式 = 2、3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog
MC_Home 任务可被下列运动控制任务中止： <ul style="list-style-type: none"> • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog 	新 MC_Home 任务可中止下列激活的运动控制任务： <ul style="list-style-type: none"> • MC_Home 模式 = 2、3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog 		

10.7.5 MC_Halt（暂停轴）指令

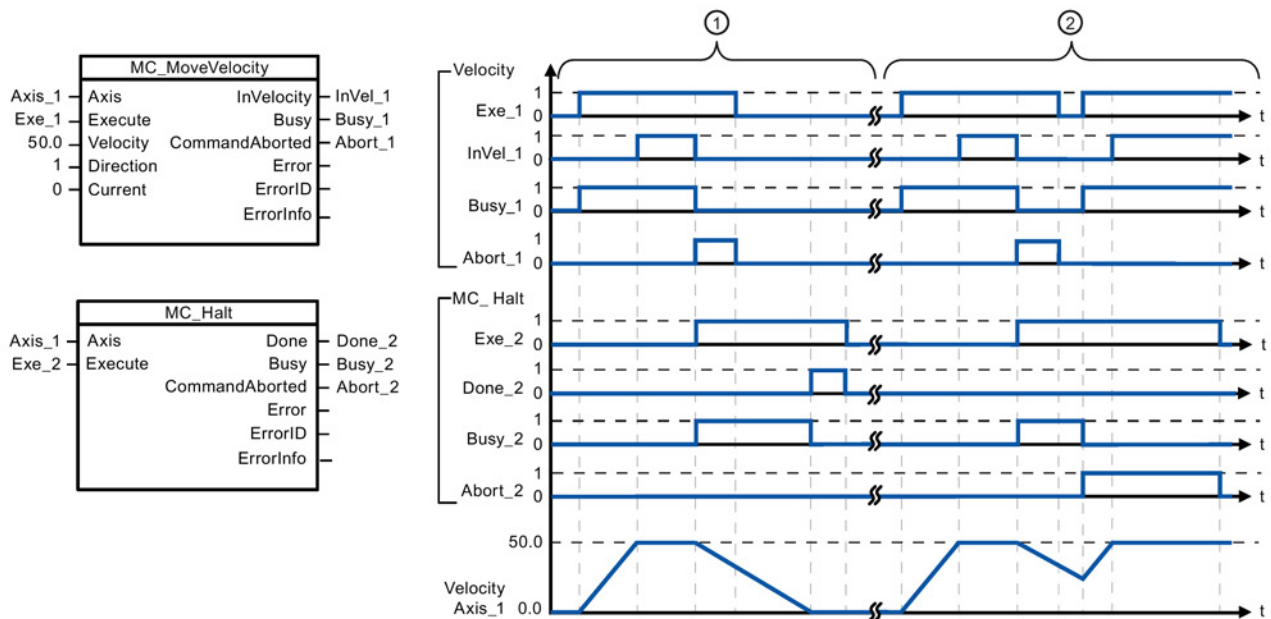
表格 10- 34 MC_Halt 指令

LAD/FBD	SCL	说明
	<pre>"MC_Halt_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Done=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>使用 MC_Halt 指令可停止所有运动并将轴切换到停止状态。停止位置未定义。</p> <p>为了使用 MC_Halt 指令，必须先启用轴。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC_Halt_DB”是背景 DB 的名称。

表格 10- 35 MC_Halt 指令的参数

参数和类型	数据类型	说明
Axis	IN	TO_Axis_1 轴工艺对象
Execute	IN	出现上升沿时开始任务
Done	OUT	TRUE = 速度达到零
Busy	OUT	TRUE = 正在执行任务。
CommandAborted	OUT	TRUE = 任务在执行期间被另一任务中止。
Error	OUT	TRUE = 任务执行期间出错。 出错原因可在“ErrorID”和“ErrorInfo”参数中找到。
ErrorID	OUT	参数“Error”的错误 ID
ErrorInfo	OUT	参数“ErrorID”的错误信息 ID



下面的值已在“动态 > 常规”(Dynamics > General) 组态窗口中组态：加速度 = 10.0，减速度 = 5.0

- ① 轴由 MC_Halt 任务进行制动，直到进入停止状态。轴的停止状态通过“Done_2”来指示。
- ② 当 MC_Halt 任务对轴进行制动处理时，另一个运动任务会中止该任务。该中止通过“Abort_2”来标识。

超驰响应

MC_Halt

任务可被下列运动控制任务中止：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

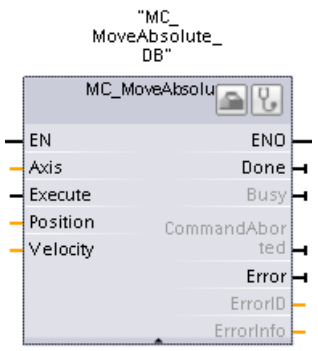
新 MC_Halt

任务可中止下列激活的运动控制任务：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.7.6 MC_MoveAbsolute（绝对定位轴）指令

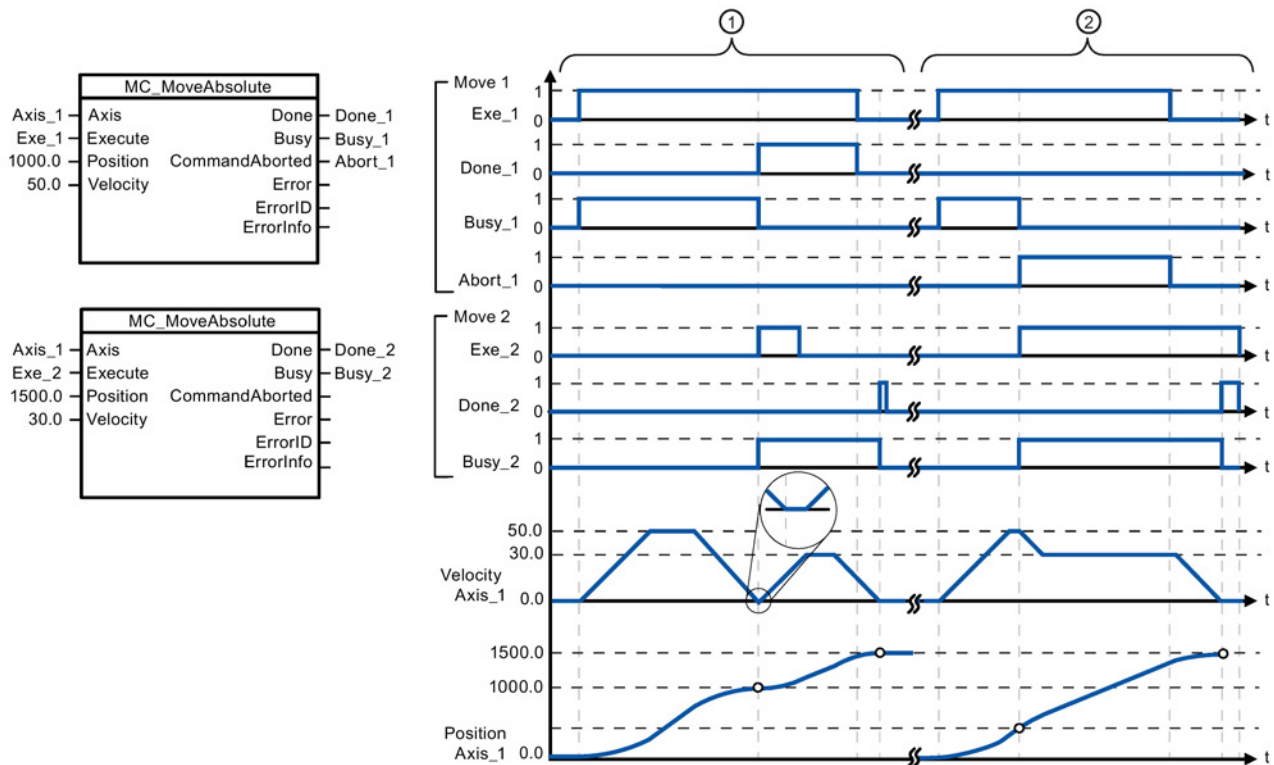
表格 10- 36 MC_MoveAbsolute 指令

LAD/FBD	SCL	说明
	<pre> "MC_MoveAbsolute_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Position:= _real_in_, Velocity:= _real_in_, Done=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_); </pre>	<p>使用 MC_MoveAbsolute 指令可启动轴到绝对位置的定位运动。</p> <p>为了使用 MC_MoveAbsolute 指令，必须先启用轴，同时必须使其回原点。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC_MoveAbsolute_DB”是背景 DB 的名称。

表格 10- 37 MC_MoveAbsolute 指令的参数

参数和类型		数据类型	说明
Axis	IN	TO_Axis_1	轴工艺对象
Execute	IN	Bool	出现上升沿时开始任务（默认值： False）
Position	IN	Real	绝对目标位置（默认值： 0.0） 0.0) 限值： $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Velocity	IN	Real	轴的速度（默认值： 10.0） 10.0) 由于组态的加速度和减速度以及要逼近的目标位置的原因，并不总是能达到此速度。 限值： 启动/停止速度 $\leq \text{Velocity} \leq$ 最大速度
Done	OUT	Bool	TRUE = 已达到绝对目标位置
Busy	OUT	Bool	TRUE = 正在执行任务。
CommandAborted	OUT	Bool	TRUE = 任务在执行期间被另一任务中止。
Error	OUT	Bool	TRUE = 任务执行期间出错。 出错原因可在“ErrorID”和“ErrorInfo”参数中找到。
ErrorID	OUT	Word	参数“Error”的错误 ID（默认值： 0000） 0000)
ErrorInfo	OUT	Word	参数“ErrorID”的错误信息 ID（默认值： 0000）



下面的值已在“动态 > 常规”(Dynamics > General) 组态窗口中组态：加速度 = 10.0，减速度 = 10.0

- ① 轴在 MC_MoveAbsolute 任务的驱动下移动到绝对位置 1000.0 处。轴到达目标位置时，通过“Done_1”对此情况进行标识。“Done_1”= TRUE 时，将启动另一个目标位置为 1500.0 的 MC_MoveAbsolute 任务。由于存在响应时间（例如，用户程序的循环时间等），轴会暂时进入停止状态（请参见放大的细节图）。轴到达新的目标位置时，通过“Done_2”对此情况进行标识。
- ② 当前 MC_MoveAbsolute 任务将由另一个 MC_MoveAbsolute 任务中止。该中止通过“Abort_1”来标识。轴随后以新的加速度移动到新的目标位置 1500.0 处。到达新的目标位置时，通过“Done_2”对此情况进行标识。

超驰响应

MC_MoveAbsolute

任务可被下列运动控制任务中止：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

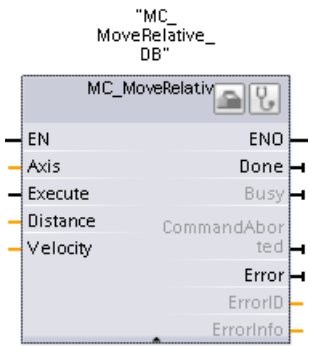
新 MC_MoveAbsolute

任务可中止下列激活的运动控制任务：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.7.7 MC_MoveRelative (相对定位轴) 指令

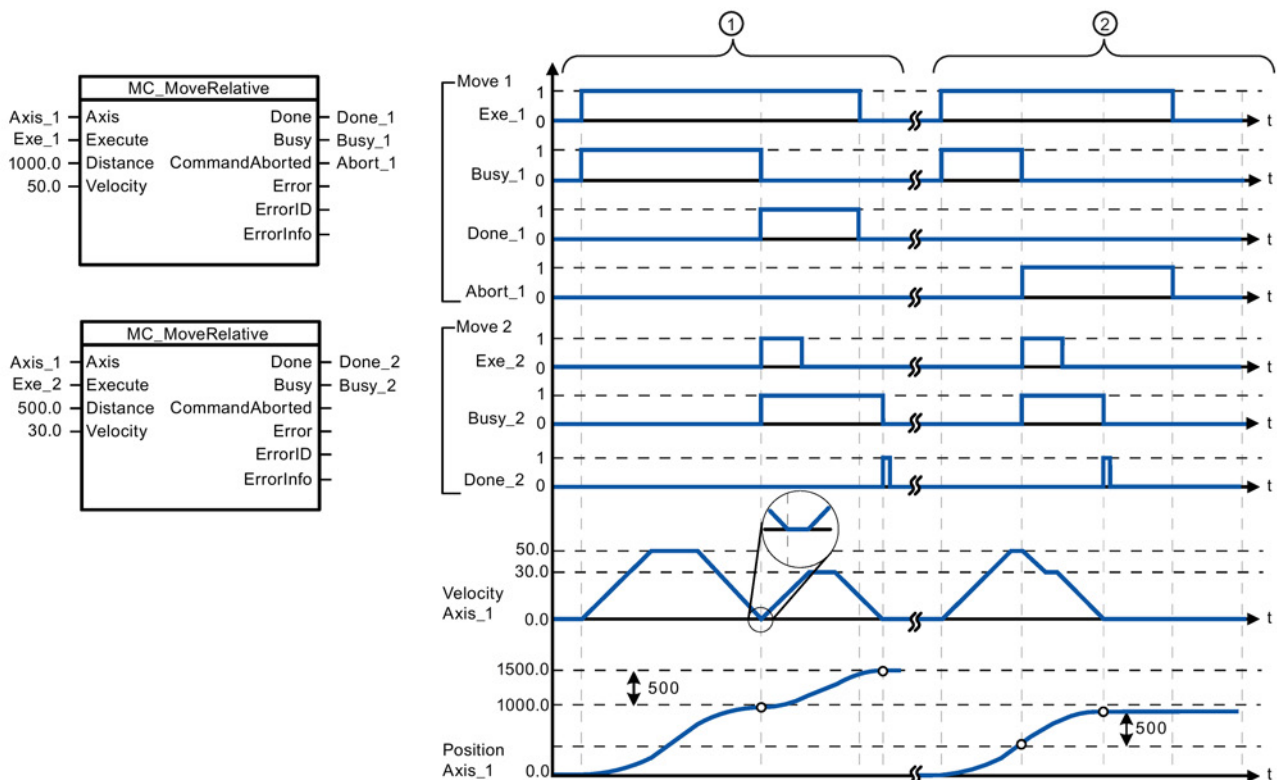
表格 10- 38 MC_MoveRelative 指令

LAD/FBD	SCL	说明
	<pre>"MC_MoveRelative_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Distance:= _real_in_, Velocity:= _real_in_, Done=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>使用 MC_MoveRelative 指令可启动相对于起始位置的定位运动。</p> <p>为了使用 MC_MoveRelative 指令，必须先启用轴。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC_MoveRelative_DB”是背景 DB 的名称。

表格 10- 39 MC_MoveRelative 指令的参数

参数和类型	数据类型	说明
Axis	IN	TO_Axis_1 轴工艺对象
Execute	IN	Bool 出现上升沿时开始任务（默认值： False）
Distance	IN	Real 定位操作的行进距离（默认值： 0.0） 0.0) 限值： $-1.0e^{12} \leq \text{Distance} \leq 1.0e^{12}$
Velocity	IN	Real 轴的速度（默认值： 10.0） 10.0) 由于组态的加速度和减速度以及要行进的距离的原因，并不总是能达到此速度。 限值： 启动/停止速度 $\leq \text{Velocity} \leq$ 最大速度
Done	OUT	Bool TRUE = 已达到目标位置
Busy	OUT	Bool TRUE = 正在执行任务。
CommandAborted	OUT	Bool TRUE = 任务在执行期间被另一任务中止。
Error	OUT	Bool TRUE = 任务执行期间出错。 出错原因可在“ErrorID”和“ErrorInfo”参数中找到。
ErrorID	OUT	Word 参数“Error”的错误 ID（默认值： 0000） 0000)
ErrorInfo	OUT	Word 参数“ErrorID”的错误信息 ID（默认值： 0000） 0000)



下面的值已在“动态 > 常规”(Dynamics > General) 组态窗口中组态：加速度 = 10.0，减速度 = 10.0

- ① 轴在 MC_MoveRelative 任务的驱动下移动 1000.0 的距离 (“Distance”)。轴到达目标位置时，通过“Done_1”对此情况进行标识。“Done_1”= TRUE 时，将启动另一个行进距离为 500.0 的 MC_MoveRelative 任务。由于存在响应时间（例如，用户程序的循环时间），轴会暂时进入停止状态（请参见放大的细节图）。轴到达新的目标位置时，通过“Done_2”对此情况进行标识。
- ② 当前 MC_MoveRelative 任务将由另一个 MC_MoveRelative 任务中止。该中止通过“Abort_1”来标识。轴随后以新的加速度移动一段新的距离 (“Distance”) 500.0。到达新的目标位置时，通过“Done_2”对此情况进行标识。

超驰响应

MC_MoveRelative

任务可被下列运动控制任务中止：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

新 MC_MoveRelative

任务可中止下列激活的运动控制任务：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.7.8 MC_MoveVelocity（以预定义速度移动轴）指令

表格 10- 40 MC_MoveVelocity 指令

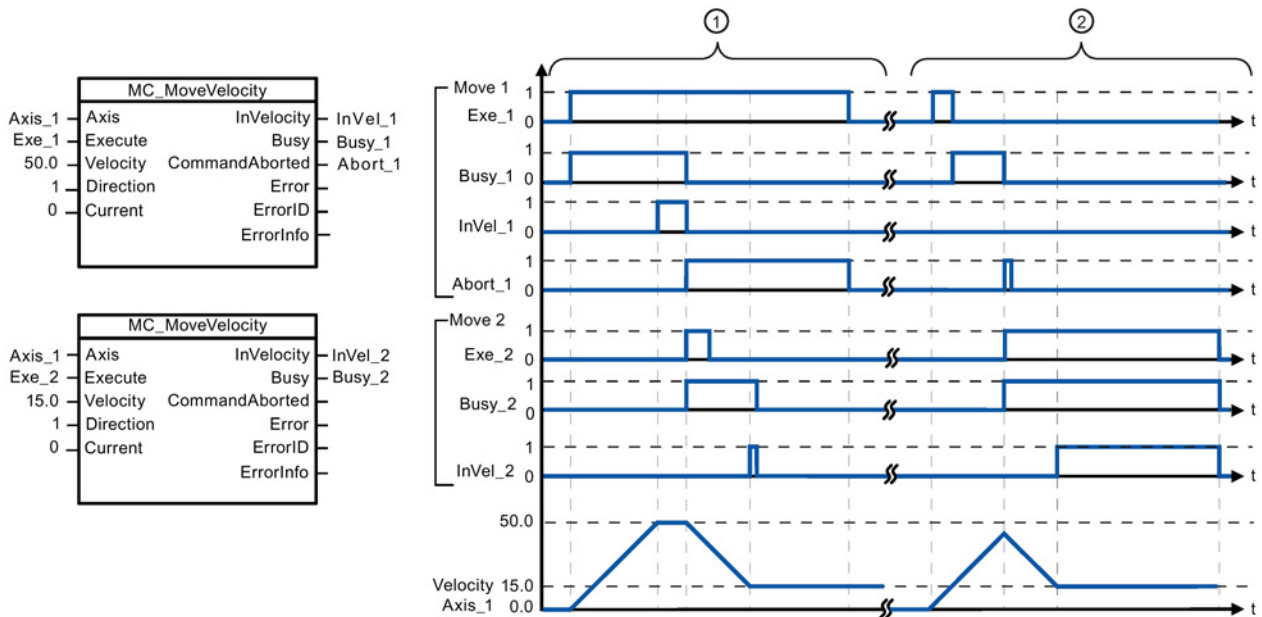
LAD/FBD	SCL	说明
	<pre>"MC_MoveVelocity_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Velocity:= _real_in_, Direction:= _int_in_, Current:= _bool_in_, InVelocity=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>使用 MC_MoveVelocity 指令以指定的速度持续移动轴。</p> <p>为了使用 MC_MoveVelocity 指令，必须先启用轴。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC_MoveVelocity_DB”是背景 DB 的名称。

表格 10- 41 MC_MoveVelocity指令的参数

参数和类型	数据类型	说明
Axis	IN	TO_Axis_1 轴工艺对象
Execute	IN	Bool 出现上升沿时开始任务（默认值： False）
Velocity	IN	Real 指定轴运动的速度（默认值： 10.0） 10.0) 限值： 启动/停止速度 ≤ Velocity ≤ 最大速度 (允许 Velocity = 0.0)
Direction	IN	Int 指定方向： <ul style="list-style-type: none"> 0: 旋转方向与参数“Velocity”中的值符号一致（默认值） 1: 正旋转方向（参数“Velocity”的值符号被忽略。） 2: 负旋转方向（参数“Velocity”的值符号被忽略。）

参数和类型		数据类型	说明
Current	IN	Bool	保持当前速度： <ul style="list-style-type: none"> • FALSE：禁用“保持当前速度”。 使用参数“Velocity”和“Direction”的值。（默认值） • TRUE：激活“保持当前速度”。 不考虑参数“Velocity”和“Direction”的值。 当轴继续以当前速度运动时，参数 "InVelocity" 返回值 TRUE.
InVelocity	OUT	Bool	TRUE： <ul style="list-style-type: none"> • 如果“Current”= FALSE： 已达到参数“Velocity”中指定的速度。 • 如果 "Current" = TRUE： 轴在启动时以当前速度运动。
Busy	OUT	Bool	TRUE = 正在执行任务。
CommandAborted	OUT	Bool	TRUE = 任务在执行期间被另一任务中止。
Error	OUT	Bool	TRUE = 任务执行期间出错。 出错原因可在“ErrorID”和“ErrorInfo”参数中找到。
ErrorID	OUT	Word	参数“Error”的错误 ID（默认值：0000） 0000)
ErrorInfo	OUT	Word	参数“ErrorID”的错误信息 ID（默认值：0000） 0000)



下面的值已在“动态 > 常规”(Dynamics > General) 组态窗口中组态： 加速度 = 10.0，减速度 = 10.0

- ① 当前 MC_MoveVelocity 任务通过“InVel_1”来指示已达到目标速度。该任务随后会被另一个 MC_MoveVelocity 任务中止。该中止通过“Abort_1”来标识。达到新的目标速度 15.0 时，将通过“InVel_2”对此情况进行指示。轴随后以新的恒定加速度继续移动。
- ② 在达到目标速度之前，当前 MC_MoveVelocity 任务会由另一个 MC_MoveVelocity 任务中止。该中止通过“Abort_1”来标识。达到新的目标速度 15.0 时，将通过“InVel_2”对此情况进行指示。轴随后以新的恒定加速度继续移动。

超驰响应

MC_MoveVelocity

任务可被下列运动控制任务中止：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

新 MC_MoveVelocity

任务可中止下列激活的运动控制任务：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

说明

速度设置为零 (Velocity = 0.0) 时的行为

“Velocity”= 0.0 的 MC_MoveVelocity 任务（如 MC_Halt 任务）可中止激活的运动任务并利用组态的减速度停止轴运动。轴停止运动后，输出参数“InVelocity”将指示 TRUE 并至少持续一个程序循环的时间。

“Busy”的值在减速运行期间为 TRUE，并且随“InVelocity”一起变为 FALSE。

如果设置了参数“Execute”= TRUE，则锁存“InVelocity”和“Busy”。

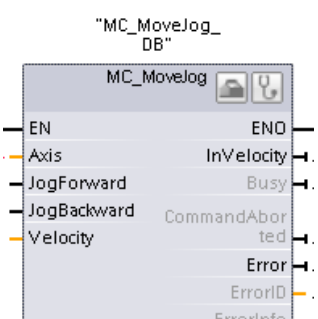
启动 MC_MoveVelocity 任务时，将设置工艺对象的状态位“SpeedCommand”。

轴停止运动后，将立即设置状态位“ConstantVelocity”。

启动新运动任务时，这两个位均会适应的新情况。

10.7.9 MC_MoveJog（在点动模式下移动轴）指令

表格 10-42 MC_MoveJog 指令

LAD/FBD	SCL	说明
	<pre>"MC_MoveJog_DB" (Axis:= _multi_fb_in_, JogForward:= _bool_in_, JogBackward:= _bool_in_, Velocity:= _real_in_, InVelocity=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>使用 MC_MoveJog 指令以指定的速度在点动模式下持续移动轴。</p> <p>该指令通常用于测试和调试。</p> <p>为了使用 MC_MoveJog 指令，必须先启用轴。</p>

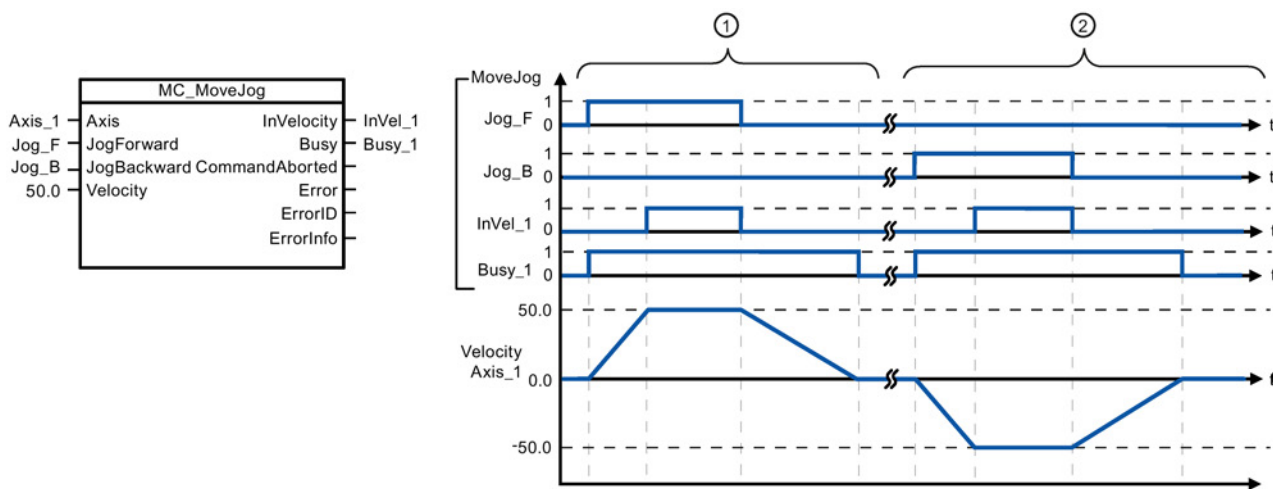
- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC_MoveJog_DB”是背景 DB 的名称。

表格 10-43 MC_MoveJog 指令的参数

参数和类型	数据类型	说明
Axis	IN	TO_Axis_1 轴工艺对象
JogForward ¹	IN	Bool 只要此参数为 TRUE，轴就会以参数“Velocity”中指定的速度沿正向移动。参数“Velocity”的值符号被忽略。（默认值：False）

参数和类型		数据类型	说明
JogBackward ¹	IN	Bool	只要此参数为 TRUE，轴就会以参数“Velocity”中指定的速度沿负向移动。参数“Velocity”的值符号被忽略。（默认值：False）
Velocity	IN	Real	点动模式的预设速度（默认值：10.0）10.0 限值：启动/停止速度 ≤ Velocity ≤ 最大速度
InVelocity	OUT	Bool	TRUE = 已达到参数“Velocity”中指定的速度。
Busy	OUT	Bool	TRUE = 正在执行任务。
CommandAborted	OUT	Bool	TRUE = 任务在执行期间被另一任务中止。
Error	OUT	Bool	TRUE = 任务执行期间出错。 出错原因可在“ErrorID”和“ErrorInfo”参数中找到。
ErrorID	OUT	Word	参数“Error”的错误 ID（默认值：0000）0000)
ErrorInfo	OUT	Word	参数“ErrorID”的错误信息 ID（默认值：0000）0000)

- ¹ 如果 JogForward 和 JogBackward 参数同时为 TRUE，则轴将以组态后的减速度停止运动。将通过参数“Error”、“ErrorID”和“ErrorInfo”指示错误。



下面的值已在“动态 > 常规”(Dynamics > General) 组态窗口中组态：加速度 = 10.0，减速度 = 5.0

- ① 通过“Jog_F”在点动模式下沿正方向移动轴。达到目标速度 50.0 时，将通过“InVelo_1”对此情况进行指示。轴会在 Jog_F 复位后再次制动直到停止。
- ② 通过“Jog_B”在点动模式下沿负方向移动轴。达到目标速度 50.0 时，将通过“InVelo_1”对此情况进行指示。轴会在 Jog_B 复位后再次制动直到停止。

超驰响应

MC_MoveJog

任务可被下列运动控制任务中止：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

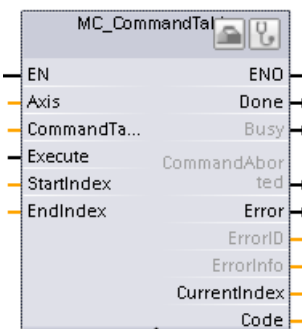
新 MC_MoveJog

任务可中止下列激活的运动控制任务：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.7.10 MC_CommandTable（按移动顺序运行轴命令）指令

表格 10- 44 MC_CommandTable 指令

LAD/FBD	SCL	说明
	<pre>"MC_CommandTable_DB" (Axis:= _multi_fb_in_, CommandTable:= _multi_fb_in_, Execute:= _bool_in_, StartIndex:= _uint_in_, EndIndex:= _uint_in_, Done=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_, CurrentIndex=> _uint_out_, Code=> _word_out_);</pre>	<p>针对电机控制轴执行一系列单个运动，这些运动可组合成一个运动序列。</p> <p>在脉冲串输出的工艺对象命令表 (TO_CommandTable_PTO) 中，可以组态这些单个的运动。</p>

- 1 STEP 7 会在插入指令时自动创建 DB。
- 2 在 SCL 示例中，“MC_CommandTable_DB”是背景 DB 的名称。

表格 10- 45 MC_CommandTable 指令的参数

参数和类型	数据类型	初始值	说明	
Axis	IN	TO_Axis_1	-	轴工艺对象
Table	IN	TO_CommandTable_1	-	命令表工艺对象
Execute	IN	Bool	FALSE	使用上升沿启动作业

参数和类型		数据类型	初始值	说明
StartIndex	IN	Int	1	从此步骤开始命令表处理 限制: $1 \leq \text{StartIndex} \leq \text{EndIndex}$
EndIndex	IN	Int	32	从此步骤结束命令表处理 限制: $\text{StartIndex} \leq \text{EndIndex} \leq 32$
Done	OUT	Bool	FALSE	MC_CommandTable 处理已成功完成
Busy	OUT	Bool	FALSE	正在运行
CommandAborted	OUT	Bool	FALSE	该任务在处理期间被另一任务中止。
Error	OUT	Bool	FALSE	处理时出错。出错原因会通过参数 ErrorID 和 ErrorInfo. 指出。
ErrorID	OUT	Word	16#0000	错误标识符
ErrorInfo	OUT	Word	16#0000	错误信息
Step	OUT	Int	0	当前在处理的步骤
Code	OUT	Word	16#0000	当前处理步骤的用户定义标识符

可在“命令表”(Command Table)

组态窗口中创建所需的运动序列，并根据趋势图中的图形视图来检查结果。



可选择要用于处理命令表的命令类型。最多可输入 32 项作业。将按顺序处理命令。

表格 10- 46 MC_CommandTable 命令类型

命令类型	说明
Empty	空白用作占位符，以便添加任意命令。在处理命令表时，忽略空白条目。
Halt	暂停轴。 注：该命令仅在“Velocity setpoint”命令之后使用。
Positioning Relative	根据距离定位轴。该命令将按给定的距离和速度移动轴。
Positioning Absolute	根据位置定位轴。该命令以指定的速度将轴移到给定位置。
Velocity setpoint	按给定速度移动轴。
Wait	等待给定期间结束。“Wait”不会停止已激活的行进运动。
Separator	在选定行上方添加“分隔”线。利用分隔线，可在单个命令表中定义多个轨迹。

执行 MC_CommandTable 的先决条件：

- 工艺对象 TO_Axis_PTO V2.0 必须已正确组态。
- 工艺对象 TO_CommandTable_PTO 必须已正确组态。
- 必须释放轴。

超驰响应

MC_CommandTable

任务可被下列运动控制任务中止：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable

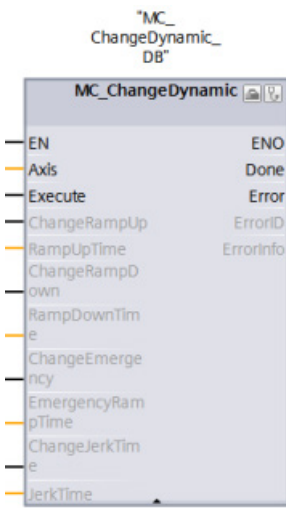
新 MC_CommandTable

任务可中止下列激活的运动控制任务：

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable
- 启动第一个“Positioning Relative”、“Positioning Absolute”、“Velocity setpoint”或“Halt”命令时的当前运动控制作业

10.7.11 MC_ChangeDynamic（更改轴的动态设置）指令

表格 10- 47 MC_ChangeDynamic 指令

LAD/FBD	SCL	说明
	<pre>"MC_ChangeDynamic_DB" (Execute:=_bool_in_, ChangeRampUp:=_bool_in_, RampUpTime:=_real_in_, ChangeRampDown:=_bool_in_, RampDownTime:=_real_in_, ChangeEmergency:=_bool_in_, EmergencyRampTime:=_real_in_, ChangeJerkTime:=_bool_in_, JerkTime:=_real_in_, Done=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>更改运动控制轴的动态设置：</p> <ul style="list-style-type: none"> 更改加速时间（加速度）值 更改减速时间（减速度）值 更改急停减速时间（急停减速度）值 更改平滑时间（冲击）值

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC_ChangeDynamic_DB”是背景 DB 的名称。

表格 10- 48 MC_ChangeDynamic 指令的参数

参数和类型	数据类型	说明
Axis	IN	TO_Axis_1 轴工艺对象
Execute	IN	Bool 出现上升沿时开始命令。默认值： FALSE
ChangeRampUp	IN	Bool TRUE = 根据输入参数“RampUpTime”更改加速时间。 默认值： FALSE
RampUpTime	IN	Real 在没有冲击限制的情况下，从静止状态加速到组态的最大速度的时间（以秒为单位）。默认值： 5.00 更改将会影响变量 <轴名称>。 Config.DynamicDefaults.Acceleration。更改的有效性显示在该变量的描述中。
ChangeRampDown	IN	Bool TRUE = 使用输入参数“RampDownTime”更改减速时间。 默认值： FALSE

参数和类型		数据类型	说明
RampDownTime	IN	Real	在没有冲击限制的情况下，轴从组态的最大速度减速到静止状态的时间（以秒为单位）。默认值： 5.00 更改将会影响变量 <轴名称>。 Config.DynamicDefaults.Deceleration。更改的有效性显示在该变量的描述中。
ChangeEmergency	IN	Bool	TRUE = 根据输入参数“EmergencyRampTime”更改急停减速时间 默认值： FALSE
EmergencyRampTime	IN	Real	在没有冲击限制的情况下，在急停模式下，轴从组态的最大速度减速到静止状态的时间（以秒为单位）。 默认值： 2.00 更改将会影响变量 <轴名称>。 Config.DynamicDefaults.EmergencyDeceleration。更改的有效性显示在该变量的描述中。
ChangeJerkTime	IN	Bool	TRUE = 根据输入参数“JerkTime”更改平滑时间。 默认值： FALSE
JerkTime	IN	Real	用于轴加速度和减速度的平滑时间（以秒为单位）。 默认值： 0.25 更改将会影响变量 <轴名称>。 Config.DynamicDefaults.Jerk。更改的有效性显示在该变量的描述中。
Done	OUT	Bool	TRUE = 更改的值已写入工艺数据块。 在更改生效时将显示变量的描述。默认值： FALSE
Error	OUT	Bool	TRUE = 命令执行期间出错。 出错原因可在“ErrorID”和“ErrorInfo”参数中找到。 默认值： FALSE
ErrorID	OUT	Word	错误标识符。默认值： 16#0000
ErrorInfo	IN	Word	错误信息。默认值： 16#0000

执行 MC_ChangeDynamic 的先决条件：

- 工艺对象 TO_Axis_PTO V2.0 必须已正确组态。
- 必须释放轴。

超驰响应

MC_ChangeDynamic 命令无法被其它任何运动控制命令中止。

新的 MC_ChangeDynamic 命令不会中止任何已激活的运动控制作业。

说明

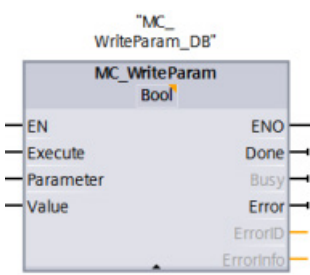
为输入参数“RampUpTime”、“RampDownTime”、“EmergencyRampTime”和“RoundingOffTime”指定的值可以使生成的轴参数“加速度”(acceleration)、“延时”(delay)、“急停延时”(emergency stop-delay) 和“冲击”(jerk) 超出允许限值。

请确保将 MC_ChangeDynamic 参数保持在轴工艺对象的动态组态设置的限制范围内。

10.7.12 MC_WriteParam (写入工艺对象的参数) 指令

使用 MC_WriteParam 指令可写入选定数量的参数来通过用户程序更改轴功能。

表格 10- 49 MC_WriteParam 指令

LAD/FBD	SCL	说明
	<pre>"MC_WriteParam_DB" (Parameter:=_variant_in_, Value:=_variant_in_, Execute:=_bool_in_, Done:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	<p>使用 MC_WriteParam 指令可写入公共参数（例如，加速度值和用户 DB 值）。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC_WriteParam_DB”是背景 DB 的名称。

可以写入公共参数。不能写入“MotionStatus”和“StatusBits”。下表列出了有效参数：

可写入参数的名称	可写入参数的名称
Actor.InverseDirection	DynamicDefaults.Acceleration
Actor.DirectionMode	DynamicDefaults.Deceleration
Actor.DriveParameter.PulsesPerDriveRevolution	DynamicDefaults.Jerk
Sensor[1].ActiveHoming.Mode	DynamicDefaults.EmergencyDeceleration

可写入参数的名称	可写入参数的名称
Sensor[1].ActiveHoming.Sidelnput	PositionLimitsHW.Active
Sensor[1].ActiveHoming.Offset	PositionLimitsHW.MaxSwitchedLevel
Sensor[1].ActiveHoming.SwitchedLevel	PositionLimitsHW.MinSwitchedLevel
Sensor[1].PassiveHoming.Mode	PositionLimitsSW.Active
Sensor[1].PassiveHoming.Sidelnput	PositionLimitsSW.MinPosition
Sensor[1].PassiveHoming.SwitchedLevel	PositionLimitsSW.MaxPosition
Units.LengthUnit	Homing.AutoReversal
Mechanics.LeadScrew	Homing.ApproachDirection
DynamicLimits.MinVelocity	Homing.ApproachVelocity
DynamicLimits.MaxVelocity	Homing.ReferencingVelocity

表格 10- 50 MC_WriteParam 指令的参数

参数和类型		数据类型	说明
PARAMNAME	IN	Variant	在其中写入值的参数名称
VALUE	IN	Variant	写入到所分配参数的值
EXECUTE	IN	Bool	启动指令。默认值： FALSE
DONE	OUT	Bool	已写入值。默认值： FALSE
BUSY	OUT	Bool	如果为 TRUE，则正在执行指令。默认值： FALSE
ERROR	OUT	Real	如果为 TRUE，则发生了错误。默认值： FALSE
ERRORID	OUT	Word	错误 ID
ERRORINFO	OUT	Word	ERRORID 的相关信息

表格 10- 51 ERRORID 和 ERRORINFO 的条件代码

ERRORID (W#16#...)	ERRORINF O (W#16#...)	说明
0	0	成功更改轴 TO-DB 参数
8410 _[1]	0028 _[1]	设置了无效参数（长度不正确的轴 TO-DB 参数）
8410 _[1]	0029 _[1]	设置了无效参数（无轴 TO-DB 参数）
8410 _[1]	002B _[1]	设置了无效参数（只读轴 TO-DB 参数）
8410 _[1]	002C _[1]	设置了有效参数，但未禁用轴
Config Error _[2]	Config Error _[2]	设置了超出范围的有效参数（公共只读轴 TO-DB 参数）
Config Error _[3]	Config Error _[3]	设置了超出范围的有效参数（公共轴 TO-DB 参数）

[1] MC_WriteParam 出错

[2] MC_Power 出错

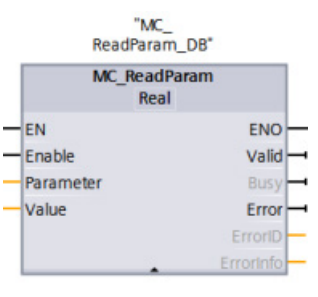
[3] MC_Power 和 MC_MoveXXX 或 MC_CommandTable 出错

10.7.13 MC_ReadParam（读取工艺对象的参数）指令

使用 MC_ReadParam

指令可读取选定数量的参数，以指示轴输入过程中定义的轴的当前位置、速度等。

表格 10- 52 MC_ReadParam 指令

LAD/FBD	SCL	说明
	<pre>"MC_ReadParam_DB" (Enable:=_bool_in_, Parameter:=_variant_in_, Value:=_variant_in_out_, Valid:=_bool_out_, Busy:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	<p>使用 MC_ReadParam 指令可读取单个状态值，与周期控制点无关。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC_ReadParam_DB”是背景 DB 的名称。

MC_ReadParam 指令通过启用来生效。

只要输入“启用”为真，指令就会将指定的“参数”读取至“值”存储位置。

每个周期控制点 (CCP) 的“MotionStatus”和“Position”值根据当前 HSC 值进行更新。

“MotionStatus”的“Velocity”值是当前时间段（更新周期 ~10ms）结束时的命令速度。

MC_ReadParam 同样可以读取该值。

如果发生错误，指令将切换到错误状态，只有“启用”输入的新上升沿才能将其复位。

表格 10- 53 MC_ReadParam 指令的参数

参数和类型	数据类型	说明	
ENABLE	IN	Bool	启动指令。默认值： FALSE
PARAMETER	IN	Variant	指向要读取的 TO 参数的指针
VALID	OUT	Bool	如果为 TRUE，则已读取该值。默认值： FALSE
BUSY	OUT	Bool	如果为 TRUE，则正在执行指令。默认值： FALSE
ERROR	OUT	Real	如果为 TRUE，则发生了错误。默认值： FALSE
ERRORID	OUT	Word	错误 ID。默认值： 0

参数和类型		数据类型	说明
ERRORINFO	OUT	Word	ERRORID. 的相关信息。默认值： 0
VALUE	INOUT	Variant	指向存储该读取值位置的指针

表格 10- 54 ERRORID 和 ERRORINFO 的条件代码

ERRORID (W#16#...)	ERRORINFO (W#16#...)	说明
0	0	成功读取参数
8410	0028	无效参数（长度不正确）
8410	0029	无效参数（无 TO-DB）
8410	0030	无效参数（不可读）
8411	0032	无效参数（值错误）

TO 参数

轴“MotionStatus”由四个值组成。可在程序运行时读取这些值，来监视这些值的变化：

变量名称	数据类型	通过 MC_ReadParam 读取
MotionStatus:	结构	-
• Position	REAL	√
• Velocity	REAL	√
• Distance	REAL	√
• TargetPosition	REAL	√

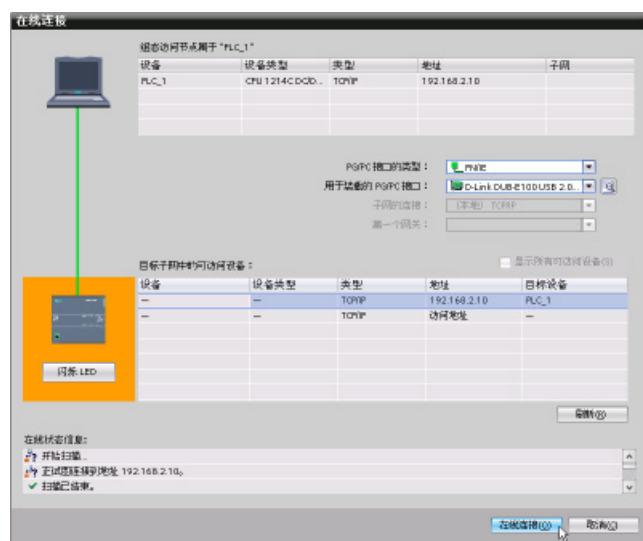
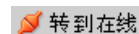
轻松使用在线工具

11.1 转到在线并连接到 CPU

装载程序和项目工程数据以及执行下列活动时，必须在编程设备和 CPU 之间建立在线连接：

- 测试用户程序
- 显示和改变 CPU 的工作模式 (页 364)
- 显示和设置 CPU 的日期和日时钟 (页 375)
- 显示模块信息
- 比较和同步 (页 374) 离线与在线程序块
- 上传和下载程序块
- 显示诊断和诊断缓冲区 (页 375)
- 通过使用监视表格 (页 367) 监视并修改值来测试用户程序
- 使用强制表格强制 CPU 中的值 (页 369)

要与组态的 CPU 建立在线连接，请单击“项目导航”树中的 CPU，并在“项目”(Project) 视图中单击“转到在线模式”(Go online) 按钮：



如果这是该 CPU 首次转到在线模式，则必须从“转到在线模式”(Go Online) 对话框中选择 PG/PC 接口的类型以及特定的 PG/PC 接口，然后才能在与该接口中发现的 CPU 建立在线连接。

现在，编程设备已连接到 CPU。橙色单元指示存在在线连接。
现在，您就可以使用“项目树”和“在线工具任务卡”中的“在线和诊断”(Online & diagnostics) 工具。

11.2 与在线 CPU 交互

项目视图下的“在线工具”(Online tools) 任务卡显示的操作员面板显示了在线 CPU 的工作模式。也可以通过该操作员面板更改在线 CPU 的工作模式。
使用操作员面板上的按钮更改工作模式 (STOP 或 RUN)。
操作员面板还提供了用于复位存储器的 MRES 按钮。

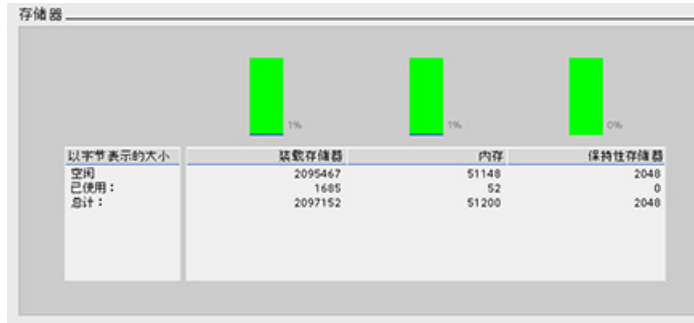


RUN/STOP 指示器的颜色指示 CPU 当前的工作模式：黄色表示 STOP 模式，而绿色表示 RUN 模式。

要使用操作员面板，必须在 STEP 7 和 CPU 之间建立在线连接。选择设备配置中的 CPU 或显示在线 CPU 中的代码块之后，可以通过“在线工具”(Online tools) 任务卡显示操作员面板。



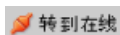
用户可以监视在线 CPU 的循环时间。



还可以查看 CPU 的存储器使用情况。

11.3 转到在线模式监视 CPU 中的值

要监视变量，必须在线连接到 CPU。只需单击工具栏中的“转到在线”(Go online) 按钮。



连接到 CPU 后，STEP 7 将工作区的标题变为橙色。
项目树显示离线项目和在线 CPU 的比较结果。绿色圆点表示 CPU 与项目同步，即二者都具有相同的组态和用户程序。
变量表会显示变量。监控表也可以显示变量以及直接地址。

名称	地址	显示格式	监视值	修改值
"On"	%I0.0	布尔型		
"Off"	%I0.1	布尔型		
"Run"	%Q0.0	布尔型		

要监视用户程序的执行并显示变量的值，请单击工具栏中的“全部监视”(Monitor all) 按钮。

名称	地址	显示格式	监视值	修改值
"On"	%I0.0	布尔型	FALSE	
"Off"	%I0.1	布尔型	FALSE	
"Run"	%Q0.0	布尔型	FALSE	

“监视值”(Monitor value) 字段中将显示每个变量的值。

11.4 显示用户程序的状态非常简单

可在 LAD 和 FBD 程序编辑器中监控多达 50 个变量的状态。使用编辑器栏显示 LAD 编辑器。

使用编辑器栏，可以在打开的编辑器之间切换视图，而无需打开或关闭编辑器。

在程序编辑器的工具栏中，单击“接通/断开监视”(Monitoring on/off)按钮，以显示用户程序的状态。



程序编辑器中的网络以绿色显示能流。

还可以右键单击指令或参数，以修改指令值。

11.5 使用监视表格监视 CPU

用户通过监视表格可以在 CPU 执行用户程序时监视或修改数据点。

这些数据点可以是输入 (I)、输出 (Q)、M 存储器、DB 或外围设备输入（如“On:P”或“I 3.4:P”）。由于监视功能只能显示从 Q

存储器写入的最后一个值，并且不会从物理输出读取实际值，因此无法准确监视物理输出（如 Q0.0:P）。

监视功能不会改变程序顺序。它为用户提供有关程序顺序的信息以及 CPU 中的程序的数据。用户也可以使用“修改值”功能来测试用户程序的执行情况。

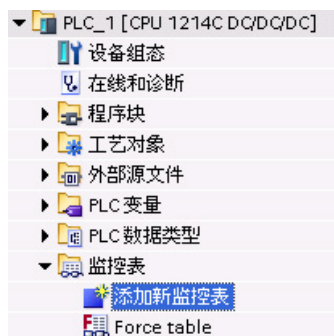
名称	地址	显示格式	监视值	使用触发器监视	使用触发器进...	修改值
1	"Start"	%I 0.0	布尔型	永久	永久	<input type="checkbox"/>
2	"Stop"	%I 0.1	布尔型	永久	永久	<input type="checkbox"/>
3	"Running"	%M 0.0	布尔型	永久	永久	<input type="checkbox"/>

说明

在设备配置期间分配高速计数器 (HSC)、脉冲宽度调制 (PWM) 和脉冲串输出 (PTO) 设备使用的数字 I/O 点。将数字 I/O 点的地址分配给这些设备之后，无法通过监视表格的“强制”功能修改所分配的 I/O 点的地址值。

使用监视表格可以监视或修改各变量的值，用户有以下选择：

- 在扫描周期开始或结束时
- CPU 切换为 STOP 模式时
- “永久”（从 STOP 切换到 RUN 后不重置值）



要创建监视表格：

1. 双击“添加新监视表格”(Add new watch table) 打开新监视表格。
2. 输入变量名称将变量添加到监视表格。

要监视变量，必须在线连接到 CPU。可使用以下选项修改变量：

- “立即修改”(Modify now) 立即修改所选地址的值一个扫描周期。
- “使用触发器修改”(Modify with trigger) 修改所选地址的值。

该功能不提供反馈来指示实际上是否修改了所选地址。

如果需要修改反馈，请使用“立即修改”(Modify now) 功能。

- “启用外围设备输出”(Enable peripheral outputs) 允许在 CPU 处于 STOP 模式时启用外围设备输出。此功能对测试输出模块的接线很有用。

可以使用监视表格顶部的按钮选择各种功能。

输入要监视的变量名称并从下拉选择项中选择一种显示格式。在线连接到 CPU 时，单击“监视”(Monitor) 按钮将在“监视值”(Monitor value) 域中显示数据点的实际值。

11.6 使用强制表格

强制表格提供了“强制”功能，能够将与外围设备输入或外围设备输出地址对应的输入或输出点的值改写成特定的值。CPU

在执行用户程序前将此强制值应用到输入过程映像并在将输出写入到模块前将其应用到输出过程映像。

说明

强制值存储在 CPU 中，而不是强制表格中。

不能强制输入（或“**I**”地址）或输出（或“**Q**”地址）。

但是，可以强制外围设备输入或外围设备输出。

强制表格将自动在地址后面添加一个“:P”（例如：“On”:P 或 “Run”:P）。

	名称	地址	显示格式	监视值	强制值	F
1	"On":P	%I0.0:P	布尔型		TRUE	<input checked="" type="checkbox"/>
2	"Off":P	%I0.1:P	布尔型			<input type="checkbox"/>
3	"Run":P	%Q0.1:P	布尔型			<input type="checkbox"/>

在“强制值”(Force value) 单元格中，输入要强制的输入值或输出值。

然后可以使用“强制”(Force) 列中的复选框启用对输入或输出的强制功能。

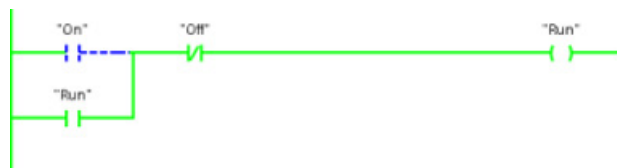
使用“启动或替换强制”(Start or replace forcing)

按钮强制设置强制表格中的变量值。单击“停止强制”(Stop forcing)

按钮重置变量值。

在强制表格中，可以监视输入的强制值的状态。但是不能监视输出的强制值。

还可以在程序编辑器中查看强制值的状态。



说明

在强制表格中强制输入或输出时，强制操作将变成项目组态的一部分。如果关闭 STEP 7，被强制元素仍会在 CPU 程序中保持激活状态，直至这些元素被清除。要清除这些被强制元素，必须使用 STEP 7 连接到在线 CPU，然后使用强制表格断开或停止对这些元素的强制功能。

CPU 允许用户在强制表格中指定物理输入或输出地址（I:P 或 Q:P）然后启动强制功能，以此来强制输入和输出点。

在程序中，物理输入的读取值被强制值覆盖。程序在处理过程中使用该强制值。程序写入物理输出时，输出值被强制值覆盖。强制值出现在物理输出端并被过程使用。

在强制表格中强制输入或输出时，强制操作将变成用户程序的一部分。

即使编程软件已关闭，强制选项在运行的 CPU

程序中仍保持激活，直到在线连接到编程软件并停止强制功能将其清除为止。

含有通过存储卡装载到另一个 CPU 的强制点的程序将继续强制程序中选择点。

如果 CPU 正在执行写保护存储卡上的用户程序，则无法通过监控表初始化或更改对 I/O 的强制，因为用户无法改写写保护用户程序中的值。

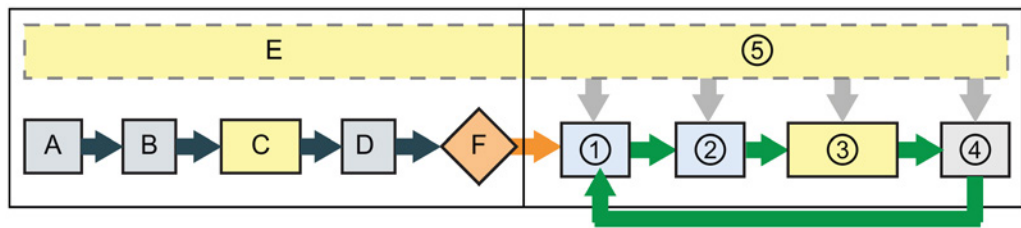
强制写保护值的任何尝试都将生成错误。

如果使用存储卡传送用户程序，则该存储卡上的所有被强制元素都将被传送到 CPU。

说明

无法强制分配给 HSC、PWM 和 PTO 的数字 I/O 点

在设备配置期间分配高速计数器 (HSC)、脉冲宽度调制 (PWM) 和脉冲串输出 (PTO) 设备使用的数字 I/O 点。将数字量 I/O 点的地址分配给这些设备之后，无法通过强制表的强制功能修改所分配的 I/O 点的地址值。



启动

- A 强制功能不影响 I 存储区的清除。
- B 强制功能不影响输出值的初始化。
- C 启动 OB 执行期间，CPU 在用户程序访问物理输入时应用强制值。
- D 不影响将中断事件存储到队列。
- E 不影响写入到输出的启用。

RUN

- ① 将 Q 存储器写入到物理输出时，CPU 在更新输出时应用强制值。
- ② 读取物理输入时，CPU 仅在将这些输入复制到 I 存储器前应用强制值。
- ③ 用户程序（程序循环 OB）执行期间，CPU 在用户程序访问物理输入或写入物理输出时应用强制值。
- ④ 强制功能不影响通信请求和自检诊断的处理。
- ⑤ 不影响在扫描周期的任何时段内处理中断。

11.7 捕获 DB 的在线值以重设起始值

可以捕获在线 CPU 中受监视的当前值，使其成为全局 DB 的起始值。

- 必须能够在线连接到 CPU。
- CPU 必须处于 RUN 模式。
- 必须已在 STEP 7 中打开相应 DB。

 使用“显示监视值的快照”(Show a snapshot of the monitored value) 按钮捕获 DB 中所选变量的当前值。然后可以将这些值复制到 DB 的“起始值”(Start value) 列。

1. 在 DB 编辑器中单击“监视所有变量”(Monitor all tags) 按钮。“监视值”(Monitor value) 列会显示当前数据值。
2. 单击“显示监视值的快照”(Show a snapshot of the monitored value) 按钮显示“快照”(Snapshot) 列中的当前值。
3. 单击“全部监视”(Monitor all) 按钮停止监视 CPU 中的数据。
4. 为变量复制“快照”(Snapshot) 列中的值。
 - 选择要复制的值。
 - 右键单击选中的值，以显示右键快捷菜单。
 - 选择“复制”(Copy) 命令。
5. 将复制的值粘贴到该变量的相应“起始值”(Start value) 列。
(右键单击单元格并从右键快捷菜单中选择“粘贴”(Paste)。)
6. 保存项目，将复制的值组态为该 DB 的新起始值。
7. 编译 DB 并将其下载到 CPU。DB 会在 CPU 进入 RUN 模式之后使用新起始值。

说明

始终从 CPU 中复制要在“监视值”(Monitor value) 列中显示的值。STEP 7 不会检查所有值是否来自于 CPU 的同一扫描周期。

11.8 上传项目元素

还可以由在线 CPU 或连接到编程设备的存储卡复制程序块。

为复制的程序块准备离线项目：

1. 添加一个与在线 CPU 匹配的 CPU 设备。
2. 展开该 CPU 节点一次，以便“程序块”(Program blocks) 文件夹可见。

要从在线 CPU

向离线项目上传程序块，请按照以下步骤操作：

1. 在离线项目中，单击“程序块”(Program blocks) 文件夹。
2. 单击“转到在线”(Go online) 按钮。
3. 单击“上传”(Upload) 按钮。
4. 在“上传”(Upload) 对话框 (页 363)中，确认所选项。

完成上传后，STEP 7

会显示项目中所有已上传的程序块。



转到在线



11.9 比较离线 CPU 与在线 CPU

可以将在线 CPU 中的代码块与项目中的代码块进行比较。如果项目中的代码块与在线 CPU 的代码块不匹配，则可通过“比较”编辑器使项目与在线 CPU 同步，具体方法可以是将项目的代码块下载到 CPU 中，或者从项目中删除在线 CPU 中不存在的块。



在项目中选择 CPU。

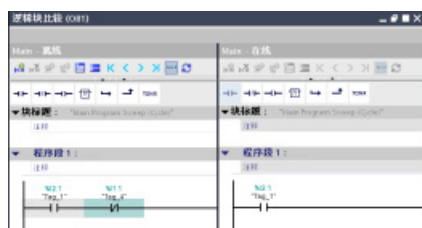
使用“比较离线/在线”(Compare Offline/online) 命令打开“比较”(Compare) 编辑器。（通过“工具”(Tools) 菜单或右键单击项目中 CPU 的方式访问该命令。）



单击某对象的“操作”(Action)

列，在删除对象、不执行任何操作或将该对象下载到设备这三项中进行选择。

单击“同步”(Synchronize) 按钮装载代码块。



在“比较目标”(Compare to)

列中右键单击一个对象，并选择“开始详细比较”(Start detailed comparison) 按钮可并排显示代码块。

详细比较功能会高亮显示在线 CPU 代码块与项目中 CPU 代码块之间的差异。

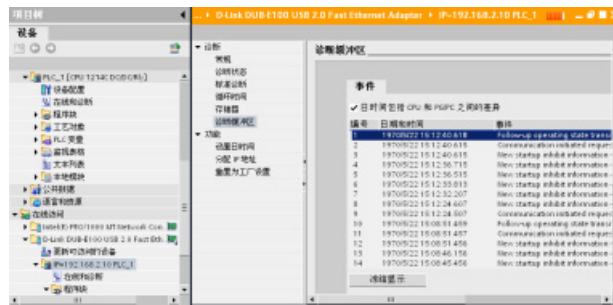
11.10 显示诊断事件

CPU 提供了一个诊断缓冲区，其中包含的每个条目对应一个诊断事件，如 CPU 工作模式切换事件，或者 CPU 或模块检测到的错误。

用户必须转到在线才能访问诊断缓冲区。

每个条目都包含了事件发生的日期和时间、事件类别及事件描述。

条目按时间顺序显示，最新发生的事件位于最上面。



在 CPU

保持通电时，该日志最多可提供 50 个最新发生的事件。

日志填满后，新事件将替换日志中最早的事件。

掉电时，将保存最新发生的十个事件。

。

11.11 设置 IP 地址和日时钟

可以设置在线 CPU 中的 IP 地址和时间。访问在线

CPU“项目树”中的“在线和诊断”(Online & diagnostics) 之后，可以显示或更改 IP 地址。

还可以显示或设置在线 CPU 的时间和日期参数。



说明

该特性仅对只具有 MAC 地址（还未分配 IP 地址）或已恢复出厂设置的 CPU 可用。

11.12 复位为出厂设置

可在以下情形下将 S7-1200 复位为原始出厂设置：

- CPU 有在线连接。
- CPU 处于 STOP 模式。

说明

如果 CPU 处于 RUN 模式，而用户要启动复位操作，则可在接受确认提示后将其切换到 STOP 模式。

步骤

要将 CPU 复位为出厂设置，请按以下步骤操作：

1. 打开 CPU 的“在线和诊断”(Online and Diagnostics) 视图。
2. 从“功能”(Functions) 文件夹中选择“复位为出厂设置”(Reset to factory settings)。
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) 复选框。
4. 单击“复位”(Reset) 按钮。
5. 单击“确定”(OK) 接受确认提示。

结果

模块会根据需要切换到 STOP 模式，并复位为出厂设置。CPU 执行以下操作：

CPU 中安装了存储卡	CPU 中未安装存储卡
<ul style="list-style-type: none"> • 清空诊断缓冲区 • 复位时间 • 从存储卡恢复工作存储器 • 将所有操作数区域设置为组态的初始值 • 将所有参数设置为其组态的值 • 根据您所做的选择，保留或删除 IP 地址。（MAC 地址是固定的，始终不变。）¹ • 如果存在控制数据记录，则将其删除 	<ul style="list-style-type: none"> • 清空诊断缓冲区 • 复位时间 • 清空工作存储器和内部装载内存 • 将所有操作数区域设置为组态的初始值 • 将所有参数设置为其组态的值 • 根据您所做的选择，保留或删除 IP 地址。（MAC 地址是固定的，始终不变。）¹ • 如果存在控制数据记录，则将其删除

CPU 中安装了存储卡	CPU 中未安装存储卡
-------------	-------------

- 1 如果选择了“保留 IP 地址”(Retain IP address), CPU 将把 IP 地址、子网掩码和路由器地址 (如果使用) 设为硬件配置中的设置, 除非已通过用户程序或其它工具修改了这些值, 这种情况下 CPU 将恢复修改后的值。

11.13 更新固件

可通过 STEP 7 在线和诊断工具更新所连接 CPU 的固件

要执行固件更新, 请执行以下步骤:

1. 打开所连接 CPU 的“在线和诊断”(Online and Diagnostics) 视图。
2. 从“Functions”文件夹中选择“固件更新”(Firmware update)。
3. 单击“浏览”(Browse) 按钮并导航至包含固件更新文件的位置。
此位置可能与您在硬盘驱动器上保存从服务与支持网站 (<http://www.siemens.com/automation/>) 下载的 S7-1200 (<http://support.automation.siemens.com/WW/view/zh/34612486/133100>) 固件更新文件的位置相同。
4. 选择与模块兼容的文件。表中会显示与所选文件兼容的模块。
5. 单击“运行更新”(Run update) 按钮。如有必要, 根据对话框更改 CPU 的工作模式。

STEP 7 在加载固件更新时会显示进程对话框。

完成后, 对话框会提示您使用新固件启动模块。

说明

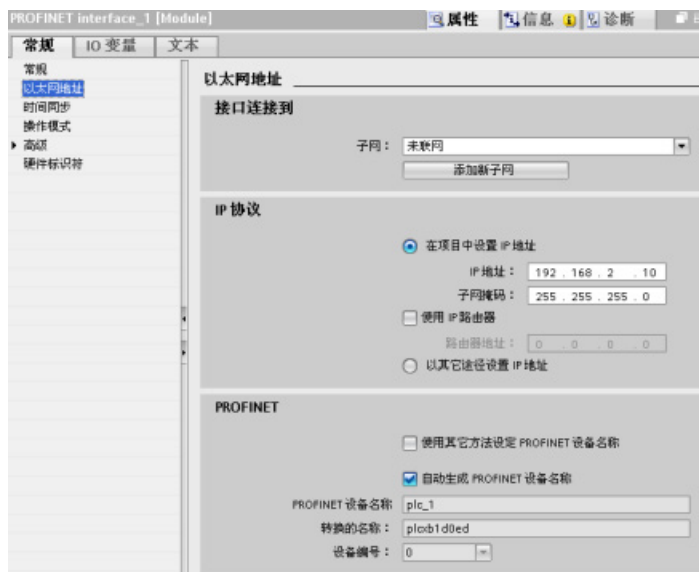
如果没有选择使用新固件启动模块, 则在通过循环上电等操作复位模块前, 先前的模块将保持激活状态。只有复位模块后, 新固件才能激活。

您还可以按以下方法之一执行固件更新:

- 使用存储卡 (页 66)
- 使用 Web 服务器“模块信息”标准 Web 页面 (页 276)

11.14 将永久 IP 地址下载到在线 CPU

要分配 IP 地址，必须执行以下任务：



- 组态 CPU (页 91) 的 IP 地址。
- 保存组态并将其下载到 CPU。

CPU 的 IP 地址和子网掩码必须与编程设备的 IP 地址和子网掩码兼容。
请咨询网络专家以确定 CPU 的 IP 地址和子网掩码。



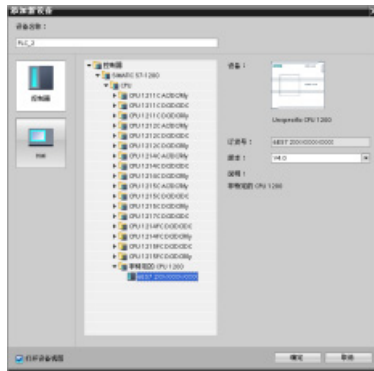
如果之前未组态 CPU，也可以使用“在线访问”设置 IP 地址。
与设备配置一起下载的 IP 地址在 PLC 循环上电时不会丢失。

下载包含 IP 地址的设备配置后，用户可以在“在线访问”(Online access) 文件夹下看到该 IP 地址。

11.15 使用“未指定的 CPU”上传硬件配置

如果用户有一个可连接到编程设备的物理 CPU，则很容易上传硬件配置。

用户必须先将 CPU 连接到编程设备，并且必须创建一个新项目。

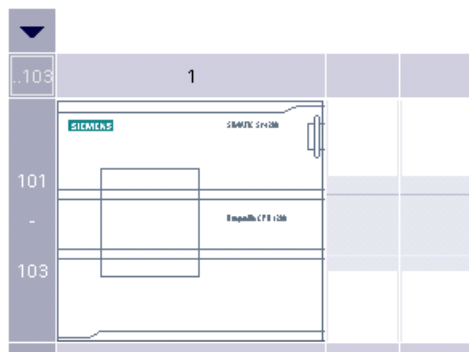


在设备配置（项目视图或门户视图）中，添加一个新设备，但要选择“未指定的 CPU”而不是选择特定的 CPU。STEP 7 即会创建一个未指定的 CPU。



创建未指定的 CPU 后，可以上传在线 CPU 的硬件配置。

- 在程序编辑器中，从“在线”(Online)菜单中选择“硬件检测”(Hardware detection) 命令。
- 在设备配置编辑器中，选择用于检测所连设备配置的选项



未指定该设备。
 → 请使用 [硬件目录](#) 指定 CPU。
 → 或 [检测](#) 相连设备的组态。

从在线对话框中选择 CPU 之后，STEP 7 会上传 CPU 以及所有模块（SM、SB 或 CM）的硬件配置。不会上传 IP 地址。必须转到“设备配置”(Device configuration) 手动组态 IP 地址。

11.16 在 RUN 模式下下载

该 CPU 支持“在 RUN 模式下下载”(Download in RUN mode)。
此功能是为了让您以对过程干扰最小的方式对控制该过程的程序进行小幅改动。
但是，执行此功能也可以对程序进行重大更改，这可能会导致损坏甚至危险情况。



警告

在 RUN 模式下下载的风险

在 RUN 模式下向 CPU 中下载更改时，这些更改将立即影响过程操作。在 RUN 模式下更改程序可能会引起意外的系统操作，进而导致人员死亡、重伤和/或设备损坏。
在 RUN 模式下执行下载的人员必须经过授权，并清楚 RUN 模式下的更改对系统运行的影响。

利用“在 RUN 模式下下载”功能，可在不切换为 STOP 模式的情况下对程序进行更改，并将其下载到 CPU 中：

- 可以在不停机的情况下对当前过程进行少量更改（例如，更改一个参数值）。
- 可利用此功能更快速地调试程序（例如，插入一段常开或常闭开关逻辑）

可在 RUN 模式下进行下列程序块和变量更改，并将其下载到 CPU 中：

- 创建、覆盖和删除函数 (FC)、函数块 (FB) 和变量表。
- 创建、删除以及覆盖数据块 (DB) 和函数块 (FB) 的背景数据块。
可添加到数据块结构并在 RUN 模式下下载它们。根据组态设置，CPU 可维持现有块变量的值并将新的数据块变量初始化为各自的初始值，或者 CPU 将所有数据块变量设置为初始值。无法在 RUN 模式下下载 Web 服务器 DB（控件或片段）。
- 覆盖组织块 (OB)；但是，不能创建或删除 OB。

在 RUN 模式下，您一次最多可下载二十个块。如果要下载的块多于二十个，必须将 CPU 置于 STOP 模式。

如果将更改下载到实际过程（相对仿真过程而言，程序调试期间可能会进行仿真），在下载前必须全面考虑可能会对机器操作员和机器造成的安全后果，这一点非常重要。

说明

如果 CPU 处于 RUN 模式，且进行了程序更改，则 STEP 7 始终会尝试先在 RUN 模式下下载。如果不希望出现这种情况，则必须将 CPU 置于 STOP 模式。

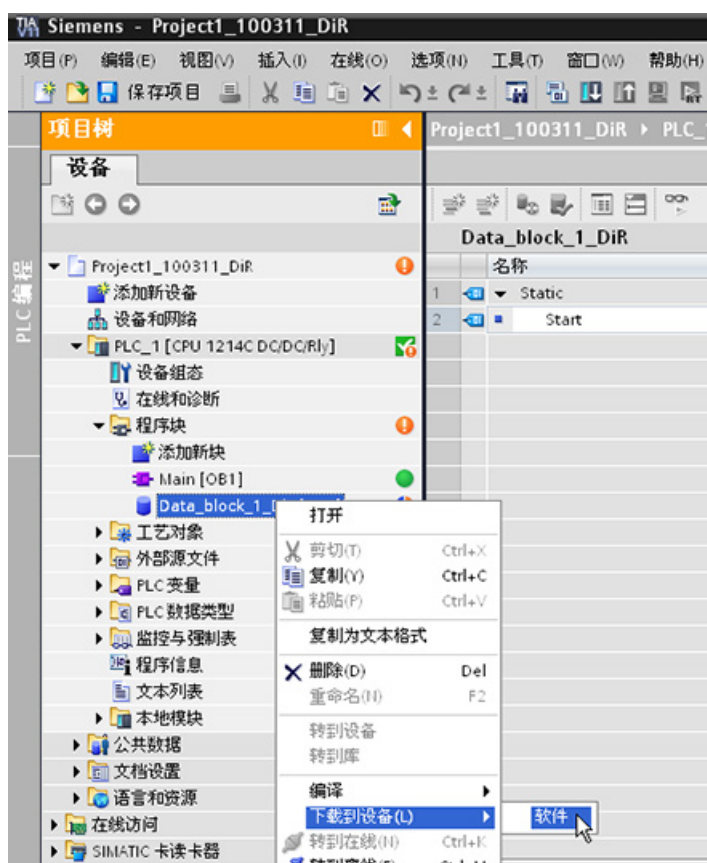
如果“在 RUN 模式下下载”不支持所做的更改，那么 STEP 7 将提示用户 CPU 必须转到 STOP 模式。

11.16.1 在 RUN 模式下更改程序

要在 RUN 模式下更改程序，首先必须确保 CPU 和程序符合先决条件，然后按照以下步骤操作：

1. 如果要在 RUN 模式下下载程序，请选择以下某种方法：

- 从“在线”(Online) 菜单中选择“下载到设备”(Download to device) 命令。
- 单击工具栏中的“下载到设备”(Download to device) 按钮。
- 在“项目树”中，右键单击“程序块”并选择“下载到设备 > 软件”(Download to device > Software) 命令。



如果程序已编译成功，STEP 7 会开始将该程序下载到 CPU 中。

2. STEP 7 将提示您加载程序或取消操作时，单击“加载”(Load) 将程序下载到 CPU。

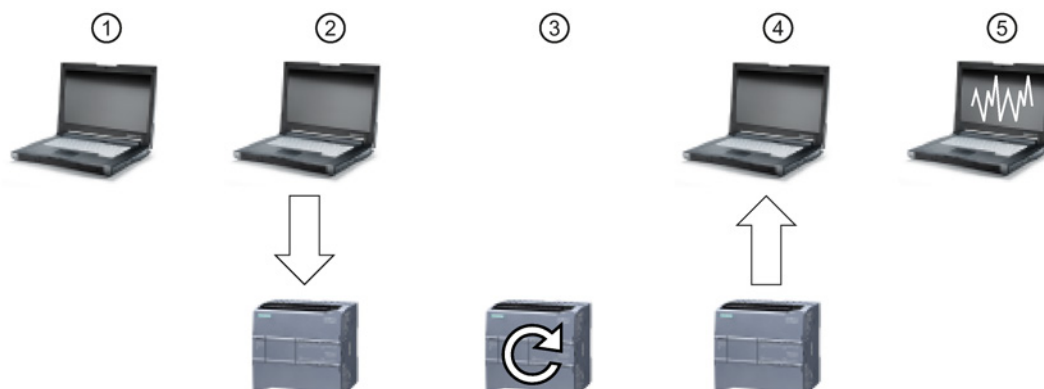
11.17 根据触发条件跟踪并记录 CPU 数据

STEP 7 提供了轨迹和逻辑分析器功能，可用于组态 PLC 要跟踪和记录的变量。

随后可将记录的轨迹数据上传到编程设备并使用 STEP 7

工具分析、管理并以图形方式显示这些数据。使用 STEP 7 项目树中的“轨迹”(Traces)文件夹创建和管理轨迹。

下图显示了轨迹功能的各个步骤：



- ① 在 STEP 7 的轨迹编辑器中组态轨迹。可以组态要记录的数据值、记录持续时间、记录频率和触发条件。
- ② 将轨迹组态从 STEP 7 传送到 PLC。
- ③ PLC 执行该程序，并在发生触发条件时开始记录轨迹数据。
- ④ 将记录的值从 PLC 传送到 STEP 7。
- ⑤ 使用 STEP 7 中的工具分析、以图形方式显示并保存该数据。

每个轨迹最大 512 KB。

访问示例

关于如何编程轨迹、如何下载组态、上传轨迹数据以及在逻辑分析器中显示数据的详细信息，请参见 STEP 7 信息系统。详细示例，请参见“使用在线和诊断功能 > 使用轨迹和逻辑分析器功能”一章。

此外，《工业自动化 SINAMICS/SIMATIC 使用轨迹和逻辑分析器功能》(<http://support.automation.siemens.com/CN/view/zh/64897128>)在线手册也是一个很好的参考。

11.17 根据触发条件跟踪并记录 CPU 数据

IO-Link 非常简单

12.1 IO-Link 技术总览

IO-Link 是适合由 PROFIBUS 用户组织 (PNO) 定义的传感器和执行器的一种创新型通信技术。IO-Link 是一项符合 IEC 61131-9 的国际标准。它基于传感器和执行器（从站）与控制器（主站）之间的点对点连接。因此，它代表的不是总线系统，而是传统点对点连接的升级。

除周期性工作数据外，扩展参数和诊断数据也由连接的传感器/执行器传输。标准传感器技术和数据传输使用同一根 3 线制连接电缆。

12.2 IO-Link 系统的组件

IO-Link 系统由 IO-Link 设备（通常为传感器、执行器或二者的组合）、标准 3 线制传感器/执行器电缆和 IO-Link 主站组成。

主站可以是采用任意设计并具有相应防护等级的设备。

IO-Link 主站可以有一个或多个端口。SM 1278 4xIO-Link Master 有四个端口。每个端口可以连接一台 IO-Link 设备或一个标准传感器/执行器。IO-Link 是一种点对点通信系统。

12.3 上电后

上电时，IO-Link 设备始终处于 SIO 模式（标准 I/O 模式）。

主站的端口可以有不同的组态。有关详细信息，请参见“S7-1200 可编程控制器系统手册”中的“IO-Link”一章。

如果将某个端口设置为 SIO 模式，主站会将此端口视作普通的数字量输入。

如果将端口设置为 IO-Link 模式（通信模式），主站会尝试寻找所连接的 IO-Link 设备。这一过程称为唤醒。

唤醒期间，主站发送已定义的信号并等待从站设备响应。

最初，主站会以可能的最高波特率尝试执行此操作。

如果不成功，主站会尝试下一较低的波特率。

主站会尝试用每种波特率对设备进行三次寻址。设备始终只支持一种已定义的波特率。

如果主站接收到响应（即，设备已被唤醒），二者即开始通信。

开始时，它们会交换通信参数，然后开始周期性交换过程数据。

如果在操作期间移除从站设备，主站会检测到通信中止，利用现场总线的特性将情况报告给控制器，并周期性尝试再次唤醒设备。

再次成功唤醒后，通信参数会被重新读出和验证（如果适用），然后再次启动周期性通信通道。

12.4 IO-Link 协议

IO-Link 系统可以交换三种类型数据：

- 周期性过程数据（过程数据输入、输出）→ 周期性数据
- 设备参数（应求数据对象）→ 非周期性数据
- 事件 → 非周期性数据

IO-Link 设备仅在收到 IO-Link 主站请求后才会发送数据。过程数据在主站的 IDLE 帧后发送，主站会明确请求设备参数数据和事件。

12.5 现场总线中的组态

IO-Link

主站作为标准现场总线节点出现在现场总线上，并通过相关网络组态器中的适当设备描述进行集成。这些文件描述了 IO-Link 主站的通信属性和其它属性，例如端口数。

它们未指明连接了哪些 IO-Link 设备。

但是，IO-Link 设备描述 (IODD) 已经定义，可实现从系统架构到 IO-Link 设备的全透明表示。借助 IODD 和 IO-Link 组态工具 S7-PCT，您可以组态将哪个 IO-Link 设备连接到 IO-Link 主站的哪个端口。

有关详细组态信息，请参见 S7-PCT 帮助系统和 *S7-1200 Programmable Controller System Manual*。

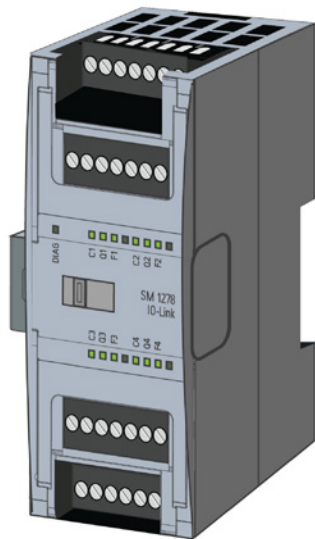
12.6 IO-Link 和 STEP 7 程序

IO-Link 主站在 STEP 7 S7-1200 控制器程序中使用 IOL_CALL 函数块 (FB) 对与 IO-Link 设备的非周期性通信进行编程。IOL_CALL FB 指示您的程序所使用的 IO-Link 主站，以及主站进行数据交换时所使用的端口。

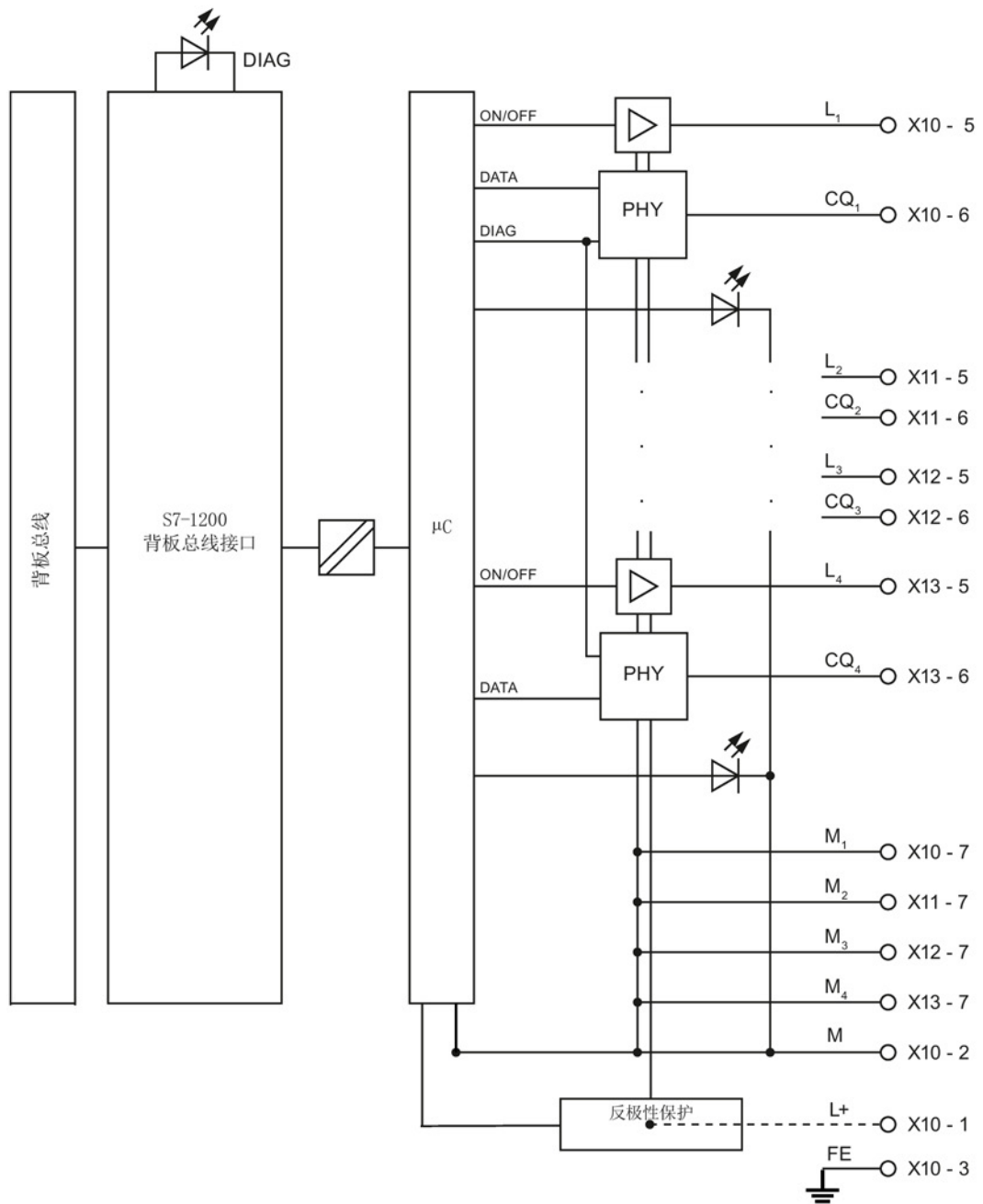
有关使用 IOL_CALL FB 的详细信息，请访问 Siemens 工业在线支持网站 (<http://support.automation.siemens.com>)。在网站的搜索框中输入“IO-Link”可访问 IO-Link 产品及其使用的相关信息。

12.7 SM 1278 4xIO-Link 主站

SM 1278 4xIO-Link Master 是一个 4 端口模块，同时具有信号模块功能和通信模块功能。每个端口均可以 IO-Link 模式、单个 24 VDC 数字量输入或 24 VDC 数字量输出方式工作。您最多可以连接四个 IO-Link 设备（3 线制连接）、四个标准执行器或标准编码器。

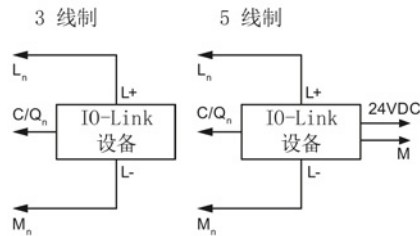


SM 1278 4xIO-Link 主站方框图

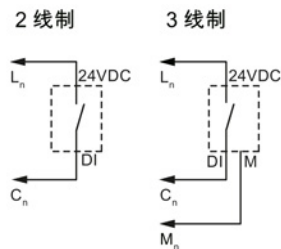


连接示例

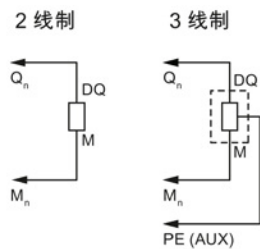
下图显示了 IO-Link 工作模式（3 线制和 5 线制）的组态，其中 $n =$ 端口号：



下图显示了 DI 工作模式（2 线制和 3 线制）的组态，其中 $n =$ 端口号：



下图显示了 DQ 工作模式（2 线制和 3 线制）的组态，其中 $n =$ 端口号：



有关使用和组态 SM 1278 4xIO-Link 主站的详细信息

有关包括图、连接、参数化、诊断报警等在内的 SM 1278 4xIO-Link Master 详细信息，请参见“S7-1200 可编程控制器系统手册”。

技术规范

A.1 常规技术规范

遵守的标准

S7-1200 自动化系统设计符合以下标准和测试规范。S7-1200 自动化系统的测试标准均基于这些标准和测试规范。

请注意，并非所有 S7-1200

型号都经过这些标准的认证，并且认证状态如果有变化，恕不另行通知。

用户有责任通过参考产品上标记的额定值来确定适用的认证。

如需更多有关按零件号排列的最新具体认证列表的信息，请咨询当地西门子代表。

CE 认证



S7-1200 自动化系统满足下列 EC

指令提出的要求和安全相关目标，并且符合欧盟的公报中列出的可编程控制器的协调欧洲标准 (EN)。

- EC 指令 2006/95/EC (低压指令) “设计用于特定电压限值内的电气设备”
 - EN 61131-2:2007 可编程控制器 - 设备要求和测试
- EC 指令 2004/108/EC (EMC 指令) “电磁兼容性”
 - 辐射标准
EN 61000-6-4:2007+A1:2011: 工业环境
 - 抗扰度标准
EN 61000-6-2:2005: 工业环境
- EC 指令 94/9/EC (ATEX) “拟用于潜在爆炸性环境的设备和保护系统”
 - EN 60079-15:2010: 保护类型“n”

可向主管部门出具的所持 CE 一致性声明文件位于以下地址：

Siemens AG
Sector Industry
I IA AS FA DH AMB
Postfach 1963
D-92209 Amberg
Germany

cULus 认证



美国保险商实验室，符合：

- 美国安全检测实验室公司：UL 508 认证（工业控制设备）
- 加拿大标准协会：CSA C22.2 第 142 号（过程控制设备）

说明

SIMATIC S7-1200 系列符合 CSA 标准。

cULus 标志表示 S7-1200 已通过美国安全检测实验室公司 (UL) 检验和认证，其符合标准 UL 508 和 CSA 22.2 第 142 号。

FM 认证



工厂共同研究协会 (FM)

认证标准类别号 3600 和 3611

批准用于:

I 类, 2 分区, 气体组别 A、B、C、D, 温度类别 T3C Ta = 60 °C

I 类, 2 区, IIC, 温度类别 T3 Ta = 60 °C

依据 CEC 18-150 的加拿大 I 类、2 分区安装

重要例外: 有关可同时使用输入或输出数的信息, 请参见技术规范。某些型号在 Ta = 60 °C 时额定值会降低。



警告

对于危险场所 I 类、2 分区和 2 区 而言, 替换组件会影响其安全性。

只能由得到授权的 Siemens 维修中心维修设备。

IECEX 认证

EN 60079-0: 易爆环境 - 一般要求

EN60079-15: 适用于易爆环境的电气设备;

防护类型“nA”

IECEX FMG14.0012X

Ex nA IIC Tx Gc

IECEX 级别信息可能与 FM 危险位置信息一起显示在产品上。

仅批准使用标有 IECEX 级别的产品。

如需更多有关按零件号排列的最新具体认证列表的信息, 请咨询当地西门子代表。

IECEX 认证不适用于继电器型号。

有关温度额定值, 请参见具体的产品铭牌。

根据 IEC 60079-15, 在合适的外壳中安装模块可提供最低级别的 IP54 保护。

ATEX 认证



ATEX 认证仅适用于 DC 型号。ATEX 认证不适用于 AC 和继电器型号。

EN 60079-0:2009: 爆炸性环境 - 一般要求

EN 60079-15:2010: 适用于潜在易爆气体环境的电气设备；
防护类型“nA”

II 3 G Ex nA IIC T4 或 T3 Gc

将模块安装在合适的机柜中，根据 EN 60529 至少要提供防护等级 IP54，或安装在可提供同等防护等级的位置。

连接的电缆和导线应在额定条件下测得的实际温度下工作。

安装应确保将瞬变电压限制在 119 V 以下。请参见本部分的“浪涌抗扰度”。

重要例外：有关可同时使用输入或输出数的信息，请参见技术规范。某些型号在 $T_a = 60\text{ }^\circ\text{C}$ 时额定值会降低。

C-Tick 认证



S7-1200 自动化系统满足 AS/NZS CISPR16 (A 类) 标准的要求。

韩国认证



S7-1200 自动化系统满足韩国认证 (KC 标志) 的要求。已被定义为 A 类设备，适合工业应用，不适合家庭应用。

欧亚关税同盟认证 (白俄罗斯、哈萨克斯坦、俄罗斯联邦)



EAC (欧亚符合性)：关税同盟 (TR CU) 技术规格的符合性声明

海事认证

S7-1200 产品定期向特定机构递交申请以便进行与特定市场和应用有关的认证。如需更多有关按零件号排列的最新具体认证列表的信息，请咨询当地西门子代表。

船级社：

- ABS (American Bureau of Shipping, 美国船级社)
- BV (Bureau Veritas, 法国船级社)
- DNV (Det Norske Veritas, 挪威船级社)
- GL (Germanischer Lloyd, 德国船级社)
- LRS (Lloyds Register of Shipping, 英国劳氏船级社)
- Class NK (Nippon Kaiji Kyokai, 日本船级社)
- 韩国船级社

工业环境

S7-1200 自动化系统设计用在工业环境中。

表格 A-1 工业环境

应用现场	辐射要求	抗扰性要求	抗噪声要求
工业	EN 61000-6-4:2007+A1:2011	EN 61000-6-2:2005	EN 61000-6-2:2005

说明

S7-1200

自动化系统旨在工业区域内使用；在住宅区内使用可能会影响无线电或电视接收。如果在居民区使用 S7-1200，必须确保射频干扰强度符合 EN 55011 的 B 类限制值。

实现 RF 干扰级别 B 的有效措施示例：

- 将 S7-1200 设备安装在接地的控制机柜中
- 在供电线路中使用噪声滤波器

确保射频干扰强度符合 EN 55011 的 B 类要求。

需要单独验收（最终的安装必须满足居民区安装的所有安全和 EMC 要求）。

电磁兼容性

电磁兼容性 (EMC) 是电气设备在电磁环境中按预期运行以及运行时电磁干扰的发射水平 (EMI) 不会干扰周围其它电气设备的能力。

表格 A- 2 抗扰度符合 EN 61000-6-2

电磁兼容性 - 抗扰度符合 EN 61000-6-2	
EN 61000-4-2 静电放电	8 kV, 对所有表面的空中放电 6 kV, 对暴露导电表面的接触放电
EN 61000-4-3 辐射、无线电频率、电磁场抗扰度测试	80 到 1000 MHz, 10 V/m, 1 kHz 时 80% AM 1.4 到 2.0 GHz, 3 V/m, 1 kHz 时 80% AM 2.0 到 2.7 GHz, 1 V/m, 1 kHz 时 80% AM
EN 61000-4-4 快速瞬变脉冲	2 kV, 5 kHz, 到交流和直流系统电源的耦合网络 2 kV, 5 kHz, 到 I/O 的耦合夹
EN 6100-4-5 浪涌抗扰度	交流系统 - 2 kV 共模, 1 kV 差模 直流系统 - 2 kV 共模, 1 kV 差模 对于直流系统, 请参见下面的浪涌抗扰度
EN 61000-4-6 传导干扰	150 kHz 到 80 MHz, 10 V RMS, 1kHz 时 80% AM
EN 61000-4-11 电压骤降	交流系统 60 Hz 时, 0% 持续 1 个周期、40% 持续 12 个周期和 70% 持续 30 个周期

浪涌抗扰度

受雷击浪涌耦合影响的布线系统必须配备外部保护。

用于评估雷击类型浪涌保护的规范之一可以在 EN 61000-4-5 中找到，其中操作限制由 EN 61000-6-2 确定。受到此标准定义的浪涌电压影响时，S7-1200 DC CPU 和信号模块需要外部保护才能保持安全运行。

下面列出了支持所需浪涌抗扰度保护的一些设备。

只有根据制造商的建议正确安装了这些设备，它们才能提供相应保护。

也可以使用由其他供应商生产、技术参数相同或更佳的设备：

表格 A- 3 支持抗浪涌保护的设备

子系统	保护设备
+24 VDC 电源	BLITZDUCTOR VT, BVT AVD 24, 零件号 918 422
工业以太网	DEHNpatch DPA M CLE RJ45B 48, 零件号 929 121
RS-485	BLITZDUCTOR XT, 基座单元 BXT BAS, 零件号 920 300
	BLITZDUCTOR XT, 模块 BXT ML2 BD HFS 5, 零件号 920 271
RS-232	BLITZDUCTOR XT, 基座单元 BXT BAS, 零件号 920 300
	BLITZDUCTOR XT, 模块 BXT ML2 BE S 12, 零件号 920 222
+24 个 VDC 数字量输入	DEHN, Inc., 型号 DCO SD2 E 24, 零件号 917 988
超过 24 个 VDC 数字量输出和传感器电 源	DEHN, Inc., 型号 DCO SD2 E 24, 零件号 917 988
模拟量 IO	DEHN, Inc., 型号 DCO SD2 E 12, 零件号 917 987
继电器输出	不需要

表格 A-4 传导和辐射发射符合 EN 61000-6-4

电磁兼容性 - 传导和辐射发射符合 EN 61000-6-4		
传导发射 EN 55011, A 类, 1 组	0.15 MHz 到 0.5 MHz	<79dB (μV) 准峰值; <66 dB (μV) 均值
	0.5 MHz 到 5 MHz	<73dB (μV) 准峰值; <60 dB (μV) 均值
	5 MHz 到 30 MHz	<73dB (μV) 准峰值; <60 dB (μV) 均值
辐射发射 EN 55011, A 类, 1 组	30 MHz 到 230 MHz	<40dB (μV/m) 准峰值; 测量距离为 10m
	230 MHz 到 1 GHz	<47dB (μV/m) 准峰值; 测量距离为 10m
	1 GHz 到 3 GHz	< 76dB (uV/m) 准峰值; 测量距离为 10m

环境条件

表格 A-5 运输与存储

环境条件 - 运输和存储	
EN 60068-2-2, 测试 Bb, 干热和 EN 60068-2-1, 测试 Ab, 寒冷	-40 °C 到 +70 °C
EN 60068-2-30, 测试 Db, 湿热	25 °C 到 55 °C, 湿度 95%
EN 60068-2-14, 测试 Na, 温度骤变	-40 °C 到 +70 °C, 停顿时间 3 小时, 5 个周期
EN 60068-2-32, 自由落体	0.3 m, 5 次, 产品包装
大气压	1080 到 660h Pa (相当于海拔 -1000 到 3500m)

表格 A-6 运行条件

环境条件 - 运行	
环境温度范围 (设备下部 25 mm 进风距离)	-20 °C 到 60 °C 水平安装 -20 °C 到 50 °C 垂直安装 湿度 95%，不结露 除非另有规定
大气压	1080 至 795 hPa (相当于海拔 -1000 到 2000m)
污染物浓度	SO ₂ : < 0.5 ppm; H ₂ S: < 0.1 ppm; RH < 60%, 不结露
	ISA-S71.04 严重度 G1、G2、G3
EN 60068-2-14, 测试 Nb, 温度变化	5 °C 到 55 °C, 3 °C/分钟
EN 60068-2-27 机械冲击	15 G, 11 ms 脉冲, 3 个轴向上 6 次冲击
EN 60068-2-6 正弦振动	DIN 导轨安装: 5-9 Hz 时 3.5 mm, 9 - 150 Hz 时 1G 面板安装: 5-9 Hz 时 7.0 mm, 9 - 150 Hz 时 2G 每个轴 10 次摆动, 每分 1 倍频程

表格 A-7 高电位绝缘测试

高电位绝缘测试	
24 VDC/5 VDC 标称电路间	520 VDC (光隔离边界的型式测试)
115 VAC/230 VAC 电路对地	1500 VAC
115 VAC/230 VAC 电路对 115 VAC/230 VAC 电路	1500 VAC
115 VAC/230 VAC 电路对 24 VAC/5 VAC 电路	1500 VAC (3000 VAC/4242 VDC 型式测试)
以太网端口对 24 VDC/5 VDC 电路和地 ¹	1500 VAC (仅限型式测试)

¹ 以太网端口隔离旨在在危险电压引起短期网络故障时限制危险情况带来的影响。它不遵照常规 AC 线电压隔离的安全要求。

保护等级

- 保护等级 II 符合 EN 61131-2 (不需要保护导线)

防护等级

- IP20 机械保护, EN 60529
- 防止手指接触经标准探针测试出的高压。需要针对灰尘、污物、水和直径小于 12.5mm 的异物施加外部保护。

额定电压

表格 A-8 额定电压


额定电压	容错
24 VDC	20.4 VDC 到 28.8 VDC
120/230 VAC	85 VAC 到 264 VAC, 47 到 63 Hz

反向电压保护

反向电压保护电路仅应用于 +24 VDC 电源的每对端子或者 CPU、信号模块 (SM) 和信号板 (SB) 上的用户输入电源。
但如果将其它端子对按相反极性接线, 仍然有可能会造成系统损坏。

S7-1200 系统中的一些 24 VDC

电源输入端口是互连的, 并且通过一个公共逻辑电路连接多个 M 端子。
例如, 在数据表中指定为“非隔离”时, 以下电路是互连的: CPU 的 24 VDC 电源、CPU 的传感器电源、SM 的继电器线圈的电源输入和非隔离模拟量输入的电源。所有非隔离的 M 端子必须连接到同一个外部参考电位。

 警告
<p>将非隔离的 M 端子连接到不同参考电位将导致意外的电流, 该电流可能导致 PLC 和任何连接设备损坏或运行不确定。</p> <p>不遵守这些准则可能会导致设备损坏或运行不确定, 而后者可能导致死亡、人员重伤和/或财产损失。</p> <p>务必确保 S7-1200 系统中的所有非隔离 M 端子都连接到同一个参考电位。</p>

DC 输出

短路保护电路不适用于 CPU、信号模块 (SM) 和信号板 (SB) 上的 DC 输出。

继电器电气使用寿命

根据抽样试验估计的典型性能数据如下。根据具体应用，实际性能可能不同。

使用适应负载的外部保护电路可增强触点的使用寿命。在感性负载和灯负载条件下，常闭触点的典型使用寿命约为常开触点的三分之一。

外部保护电路可以延长触点的寿命。

表格 A-9 典型性能数据

用于选择执行器的数据				
连续热电流		最大 2 A		
触点的开关容量和使用寿命				
对于电阻负载	电压	电流	操作循环数（典型值）	
	24 VDC	2.0 A	10 万次	
	24 VDC	1.0 A	20 万次	
	24 VDC	0.5 A	100 万次	
	48 VAC	1.5 A	150 万次	
	60 VAC	1.5 A	150 万次	
	120 VAC	2.0 A	100 万次	
	120 VAC	1.0 A	150 万次	
	120 VAC	0.5 A	200 万次	
	230 VAC	2.0 A	100 万次	
	230 VAC	1.0 A	150 万次	
	230 VAC	0.5 A	200 万次	
对于感性负载（符合		电压	电流	操作循环数（典型值）

用于选择执行器的数据			
IEC 947-5-1 DC13/AC15)	24 VDC	2.0 A	5 万次
	24 VDC	1.0 A	10 万次
	24 VDC	0.5 A	50 万次
	24 VAC	1.5 A	100 万次
	48 VAC	1.5 A	100 万次
	60 VAC	1.5 A	100 万次
	120 VAC	2.0 A	70 万次
	120 VAC	1.0 A	100 万次
	120 VAC	0.5 A	150 万次
	230 VAC	2.0 A	70 万次
	230 VAC	1.0 A	100 万次
	230 VAC	0.5 A	150 万次
激活数字量输入		可以	
切换频率			
	机械式	最大 10 Hz	
	电阻负载	最大 1 Hz	
	感性负载（符合 IEC 947-5-1 DC13/AC15）	最大 0.5 Hz	
	灯负载	最大 1Hz	

内部 CPU 内存保持性

- 保持性数据和数据日志数据的寿命：10 年
- 保持性数据掉电时，写入周期使用寿命：2 百万个周期
- 数据日志数据，每个数据日志条目最多 2 KB，写入周期使用寿命：5 亿个数据日志条目

说明

数据日志对内部 CPU 内存的影响

每次数据日志写入至少占用 2 KB 的存储空间。

如果程序频繁写入少量数据，每次写入至少占用 2 KB 的存储空间。

更好的方法是将小数据项累积在一个数据块 (DB)

中，并以更长的时间间隔将数据块写入到数据日志中。

如果程序以高频率写入了许多数据日志条目，请考虑使用可更换的 SD 存储卡。

A.2 CPU 模块

有关 S7-1200 可用模块的更完整列表，请参见“S7-1200

可编程控制器系统手册”或客户支持网站 (<http://www.siemens.com/tiaportal>)。

表格 A- 10 常规规范

常规规范		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
产品编号	AC/DC/继电器	6ES7 211-1BE40-0XB0	6ES7 212-1BE40-0XB0	6ES7 214-1BG40-0XB0	6ES7 215-1BG40-0XB0	--
	DC/DC/继电器	6ES7 211-1HE40-0XB0	6ES7 212-1HE40-0XB0	6ES7 214-1HG40-0XB0	6ES7 215-1HG40-0XB0	--
	DC/DC/DC	6ES7 211-1AE40-0XB0	6ES7 212-1AE40-0XB0	6ES7 214-1AG40-0XB0	6ES7 215-1AG40-0XB0	6ES7 217-1AG40-0XB0
尺寸 W x H x D (mm)		90 x 100 x 75		110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
重量	• AC/DC/继电器	• 420 g	• 425 g	• 475 g	• 585 g	-
	• DC/DC/继电器	• 380 g	• 385 g	• 435 g	• 550 g	-
	• DC/DC/DC	• 370 g	• 370 g	• 415 g	• 520 g	530 g
功耗	• AC/DC/继电器	• 10 W	• 11 W	• 14 W	• 14 W	-
	• DC/DC/继电器	• 8 W	• 9 W	• 12 W	• 12 W	-
	• DC/DC/DC	• 8 W	• 9 W	• 12 W	• 12 W	12 W
SM 和 CM 总线的可用电流 (5 VDC)		最大 750 mA	最大 1000 mA	最大 1600 mA	最大 1600 mA	最大 1600 mA
传感器电源的可用电流 (24 VDC)		最大 300 mA	最大 300 mA	最大 400 mA	最大 400 mA	最大 400 mA
数字量输入电流消耗 (24 VDC)		所用的每点输入 4 mA	所用的每点输入 4 mA	所用的每点输入 4 mA	使用的每点输入 4 mA	所用的每点输入 4 mA

表格 A- 11 CPU 特征

CPU 特征		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
用户存储器						
• 工作存储器		• 50 KB	• 75 KB	• 100 KB	• 125 KB	• 150 KB
• 装载存储器		• 1 MB	• 1 MB	• 4 MB	• 4 MB	• 4 MB
• 保持性存储器		• 10 KB	• 10 KB	• 10 KB	• 10 KB	• 10 KB
板载数字 I/O 请参见规范 (页 420)。		6 路输入 4 路输出	8 路输入 6 路输出	14 路输入 10 路输出	14 路输入 10 路输出	14 路输入 10 路输出
板载模拟 I/O 请参见规范 (页 434)。		2 路输入	2 路输入	2 路输入	2 路输入 2 路输出	2 路输入 2 路输出
过程映像大小						
• 输入		• 1024 个字节	• 1024 个字节	• 1024 个字节	• 1024 个字节	• 1024 个字节
• 输出		• 1024 个字节	• 1024 个字节	• 1024 个字节	• 1024 个字节	• 1024 个字节
位存储器 (M)		4096 个字节	4096 个字节	8192 个字节	8192 个字节	8192 个字节
临时 (本地) 存储器		<ul style="list-style-type: none"> • 16 KB 用于启动和程序循环 (包括相关的 FB 和 FC) • 6 KB 用于其它每个中断优先级 (包括 FB 和 FC) 				
SM 扩展		无	最多 2 个信号模块	最多 8 个信号模块	最多 8 个信号模块	最多 8 个信号模块
SB、CB 或 BB 扩展		最多 1 个	最多 1 个	最多 1 个	最多 1 个	最多 1 个
CM 扩展		最多 3 个	最多 3 个	最多 3 个	最多 3 个	最多 3 个
高速计 数器	总计	最多可组态 6 个使用任意内置或 SB 输入的高速计数器				
	1 MHz	--	--	--	--	Ib.2 到 Ib.5 (差动)
	100/180 kHz	Ia.0 到 Ia.5	Ia.0 到 Ia.5	Ia.0 到 Ia.5	Ia.0 到 Ia.5	Ia.0 到 Ia.5
	30/120 kHz	--	Ia.6 到 Ia.7	Ia.6 到 Ib.5	Ia.6 到 Ib.5	Ia.6 到 Ib.1
脉冲输 出 ²	总计	最多可组态 4 个使用任意内置或 SB 输出的脉冲输出				
	1 MHz	--	--	--	--	Qa.0 到 Qa.3 (差动)
	100 kHz	Qa.0 到 Qa.3	Qa.0 到 Qa.3	Qa.0 到 Qa.3	Qa.0 到 Qa.3	Qa.4 到 Qb.1
	30 kHz	--	Qa.4 到 Qa.5	Qa.4 到 Qb.1	Qa.4 到 Qb.1	--

CPU 特征	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
脉冲捕捉输入	6	8	14	14	14
延时中断	共 4 个, 精度为 1 ms	共 4 个, 精度为 1 ms	共 4 个, 精度为 1 ms	共 4 个, 精度为 1 ms	共 4 个, 精度为 1 ms
循环中断	共 4 个, 精度为 1 ms	共 4 个, 精度为 1 ms	共 4 个, 精度为 1 ms	共 4 个, 精度为 1 ms	共 4 个, 精度为 1 ms
沿中断 使用可选 SB 时	6 个上升沿和 6 个下降沿 10 个上升沿和 10 个下降沿	8 个上升沿和 8 个下降沿 12 个上升沿和 12 个下降沿	12 个上升沿和 12 个下降沿 16 个上升沿和 16 个下降沿	12 个上升沿和 12 个下降沿 16 个上升沿和 16 个下降沿	12 个上升沿和 12 个下降沿 16 个上升沿和 16 个下降沿
实时时钟 • 精度 • 保持时间 (免维护超级电容)	• +/- 60 秒/月 • 通常为 20 天, 40 °C 时最少为 12 天	• +/- 60 秒/月 • 通常为 20 天, 40 °C 时最少为 12 天	• +/- 60 秒/月 • 通常为 20 天, 40 °C 时最少为 12 天	• +/- 60 秒/月 • 通常为 20 天, 40 °C 时最少为 12 天	• +/- 60 秒/月 • 通常为 20 天, 40 °C 时最少为 12 天
执行速度 • 布尔运算 • 移动字 • 实数数学运算	• 0.08 μs/指令 • 1.7 μs/指令 • 2.3 μs/指令	• 0.08 μs/指令 • 1.7 μs/指令 • 2.3 μs/指令	• 0.08 μs/指令 • 1.7 μs/指令 • 2.3 μs/指令	• 0.08 μs/指令 • 1.7 μs/指令 • 2.3 μs/指令	• 0.08 μs/指令 • 1.7 μs/指令 • 2.3 μs/指令

- 1 将 HSC 组态为正交工作模式时, 可应用较慢的速度。
- 2 对于具有继电器输出的 CPU 型号, 必须安装数字量信号板 (SB) 才能使用脉冲输出。

表格 A- 12 通信

技术数据	CPU 1211C、CPU 1212C、CPU 1214C	CPU 1215C、CPU 1217C
通信 <ul style="list-style-type: none"> • 数据传输率 • 隔离（外部信号与 PLC 逻辑侧） • 电缆类型 	1 个以太网端口 <ul style="list-style-type: none"> • 10/100 Mb/s • 变压器隔离，1500 VDC • CAT5e 屏蔽电缆 	2 个以太网端口 <ul style="list-style-type: none"> • 10/100 Mb/s • 变压器隔离，1500 VDC • CAT5e 屏蔽电缆
设备	<ul style="list-style-type: none"> • 4 HMI • 1 PG 	<ul style="list-style-type: none"> • 4 HMI • 1 PG
以太网连接 ¹	8（主动或被动）	8（主动或被动）
CPU 至 CPU S7 连接 (GET/PUT)	<ul style="list-style-type: none"> • 8（客户端） • 3（服务器） 	<ul style="list-style-type: none"> • 8（客户端） • 3（服务器）

¹ 开放式用户通信连接（主动或被动）：TSEND_C、TRCV_C、TCON、TDISCON、TSEND 和 TRCV。

表格 A-13 CPU 1214C AC/DC/继电器的接线图

CPU 1214C AC/DC/继电器	
<p>The diagram shows the terminal block of a CPU 1214C AC/DC/Relay module. It includes connections for AC power (L1, N), DC power (L+, M, -), 24VDC digital inputs (DI a, DI b), analog inputs (AI), and relay outputs (DQ a, DQ b). A note indicates that the X11 connector must be gold-plated.</p>	<p>① 24 VDC 传感器电源输出。 要获得更好的抗噪声效果，即使未使用传感器电源，也可将“M”连接到机壳接地。</p>
	<p>② 对于漏型输入，将“-”连接到“M”（如图所示）。 对于源型输入，将“+”连接到“M”。</p>
	<p>注 1： X11 连接器必须镀金。 有关产品编号，请参见“<i>S7-1200 可编程控制器系统手册</i>”的附录 C“备件”。</p>
	<p>注 2： 可将 L1 或 N (L2) 端子连接到最高 240 VAC 的电压源。 N 端子可被视为 L2 并且无需接地。 L1 和 N (L2) 端子无需极化。</p>
<p>注 3： 有关 CPU 以太网端口的信息，请参见“<i>S7-1200 可编程控制器系统手册</i>”中的“设备组态”。</p>	

表格 A- 14 CPU 1214C DC/DC/DC 的接线图

CPU 1214C DC/DC/DC	
	<p>① 24 VDC 传感器电源输出。要获得更好的抗噪声效果，即使未使用传感器电源，也可将“M”连接到机壳接地。</p>
	<p>② 对于漏型输入，将“-”连接到“M”（如图所示）。对于源型输入，将“+”连接到“M”。</p>
	<p>注 1： X11 连接器必须镀金。有关产品编号，请参见“S7-1200 可编程控制器系统手册”的附录 C“备件”。</p> <p>注 2： 有关 CPU 以太网端口的信息，请参见“S7-1200 可编程控制器系统手册”中的“设备组态”。</p>

A.3 数字 I/O 模块

A.3 数字 I/O 模块

有关 S7-1200 可用模块的更完整列表，请参见“S7-1200 可编程控制器系统手册”或客户支持网站 (<http://www.siemens.com/tiaportal>)。

A.3.1 SB 1221 SB 1222 和 SB 1223 数字量输入/输出 (DI、DQ 和 DI/DQ)

表格 A- 15 SB 1221 数字量输入 (DI) 和 SB 1222 数字量输出 (DQ) 模块

常规		SB 1221 4 DI (200 kHz)	SB 1222 4 DQ (200 kHz)
产品编号		<ul style="list-style-type: none"> • 24 VDC: 6ES7 221-3BD30-0XB0 • 5 VDC: 6ES7 221-3AD30-0XB0 	<ul style="list-style-type: none"> • 24 VDC: 6ES7 222-1BD30-0XB0 • 5 VDC: 6ES7 222-1AD30-0XB0
尺寸 W x H x D (mm)		38 x 62 x 21	38 x 62 x 21
重量		35 g	35 g
功耗		<ul style="list-style-type: none"> • 24 VDC: 1.5 W • 5 VDC: 1.0 W 	0.5 W
电流消耗	SM 总线	40 mA	35 mA
	24 VDC	<ul style="list-style-type: none"> • 24 VDC: 7 mA/输入 + 20 mA • 5 VDC: 15 mA/输入 + 15 mA 	15 mA
输入/输出		4 路输入 (源型)	4 路输出 (固态 - MOSFET)

表格 A- 16 SB 1223 数字量输入/输出组合 (DI/DQ) 模块

常规		SB 1223 DI/DQ (200 kHz)	SB 1223 2 DI/2 DQ
产品编号		<ul style="list-style-type: none"> • 24 VDC: 6ES7 223-3BD30-0XB0 • 5 VDC: 6ES7 223-3AD30-0XB0 	24 VDC: 6ES7 223-0BD30-0XB0
尺寸 W x H x D (mm)		38 x 62 x 21	38 x 62 x 21
重量		35 g	40 g
功耗		<ul style="list-style-type: none"> • 24 VDC: 1.0 W • 5 VDC: 0.5 W 	24 VDC: 1.0 W
电流消耗	SM 总线	35 mA	50 mA
	24 VDC	<ul style="list-style-type: none"> • 24 VDC: 7 mA/输入 + 20 mA • 5 VDC: 15 mA/输入 + 15 mA 	所用的每点输入 4 mA
输入/输出		2 路输入 (源型) 2 路输出 (固态 - MOSFET)	2 路输入 (IEC 1 类漏型) 2 路输出 (固态 - MOSFET)

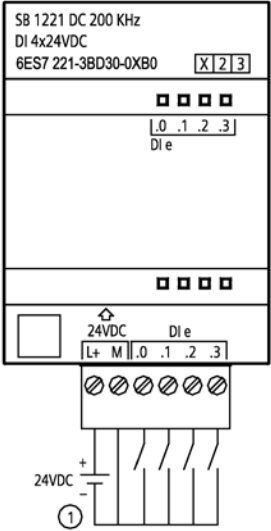
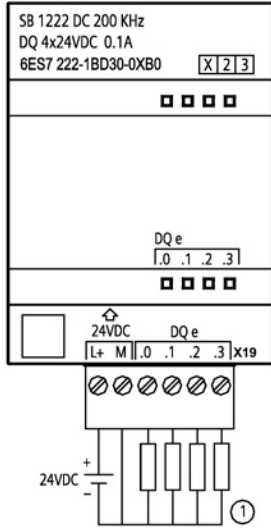
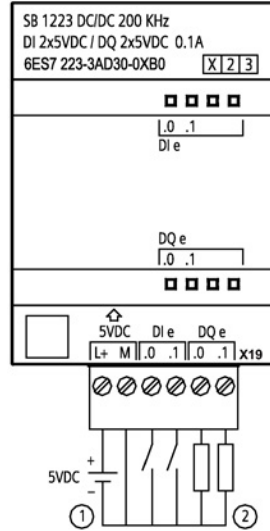
说明

高速 (200 kHz) SB 使用“源型”输入。标准 SB (20 kHz) 使用“漏型”输入。
请参见关于数字量输入和输出 (页 420) 的规范。

高速 (200 kHz) 输出 (SB 1222 和 SB 1223) 可能是源型或漏型。

对于源型输出, 将“负载”连接到“-” (如图所示)。对于漏型输出, 将“负载”连接到“+”。
因为通过同一电路来支持漏型和源型配置, 所以源型负载的激活状态与漏型负载的相反。
源型输出表现为正逻辑 (当负载有电流时, Q 位接通且 LED 亮起), 而漏型输出表现为负逻辑 (当负载有电流时, Q 位断开且 LED 熄灭)。
如果插入模块且无用户程序, 则此模块的默认值是 0V, 这意味着漏型负载将接通。

表格 A- 17 数字 SB 的接线图

SB 1221 输入模块	SB 1222 输出模块	SB 1223 输入/输出模块
<p style="text-align: center;">SB 1221 DI 4 (200 kHz)</p> 	<p style="text-align: center;">SB 1222 DQ 4 (200 kHz)</p> 	<p style="text-align: center;">SB 1223 DI 2/DQ 2 (200 kHz)</p> 
<p>① 仅支持源型输入。</p>	<p>① 对于源型输出，将“负载”连接到“-”（如图所示）。 对于漏型输出，将“负载”连接到“+”。</p> <p>因为通过同一电路来支持漏型和源型配置，所以源型负载的激活状态与漏型负载的相反。 源型输出表现为正逻辑（当负载有电流时，Q 位接通且 LED 亮起），而漏型输出表现为负逻辑（当负载有电流时，Q 位断开且 LED 熄灭）。</p> <p>如果插入模块且无用户程序，则此模块的默认值是 0 V，这意味着漏型负载将接通。</p>	<p>① 仅支持源型输入。</p> <p>② 对于源型输出，将“负载”连接到“-”（如图所示）。 对于漏型输出，将“负载”连接到“+”。</p> <p>因为通过同一电路来支持漏型和源型配置，所以源型负载的激活状态与漏型负载的相反。 源型输出表现为正逻辑（当负载有电流时，Q 位接通且 LED 亮起），而漏型输出表现为负逻辑（当负载有电流时，Q 位断开且 LED 熄灭）。</p> <p>如果插入模块且无用户程序，则此模块的默认值是 0 V，这意味着漏型负载将接通。</p>

说明

高速 (200 kHz) SB (SB 1221 和 SB 1223) 仅支持漏型输入。标准 SB 1223 仅支持源型输入。

高速 (200 kHz) 输出 (SB 1222 和 SB 1223) 可能是源型或漏型。

对于源型输出, 将“负载”连接到“-” (如图所示)。对于漏型输出, 将“负载”连接到“+”。因为通过同一电路来支持漏型和源型配置, 所以源型负载的激活状态与漏型负载的相反。

源型输出表现为正逻辑 (当负载有电流时, Q 位接通且 LED

亮起), 而漏型输出表现为负逻辑 (当负载有电流时, Q 位断开且 LED 熄灭)。

如果插入模块且无用户程序, 则此模块的默认值是 0 V, 这意味着漏型负载将接通。

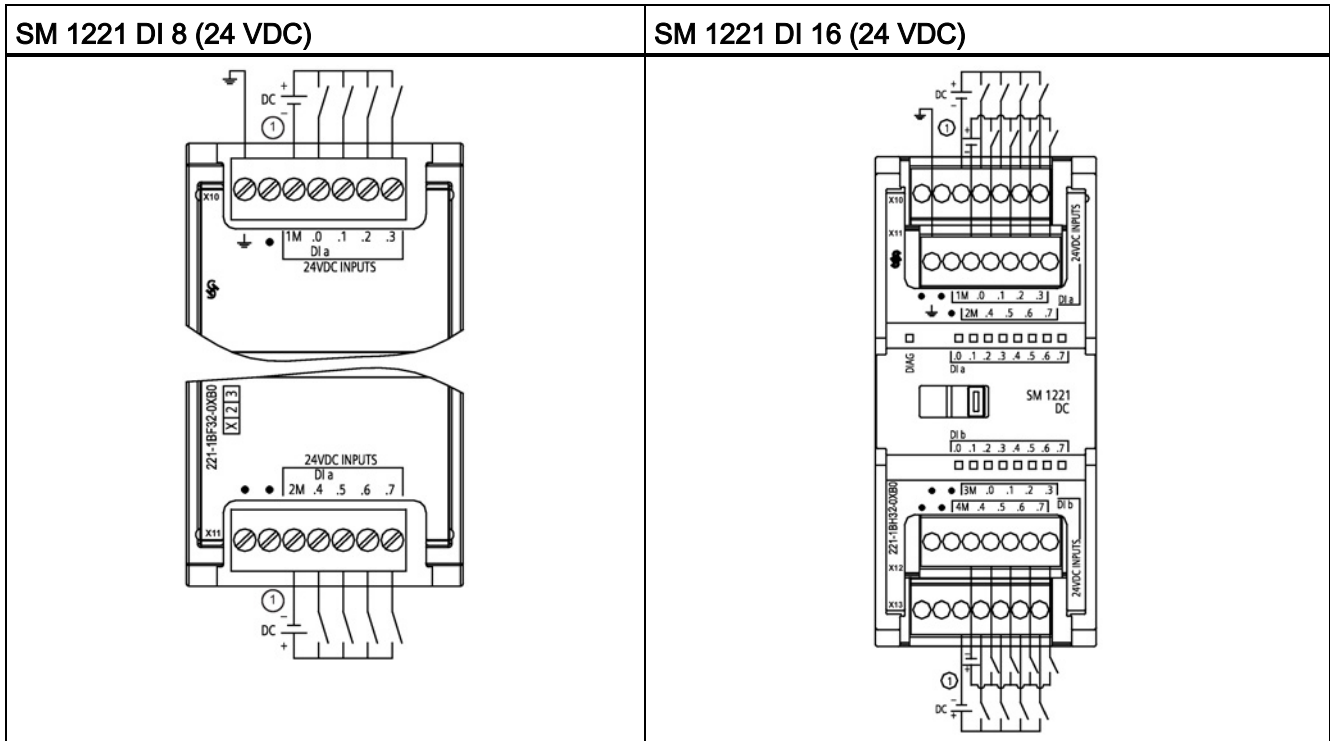
A.3.2 SM 1221 数字量输入 (DI)

表格 A- 18 SM 1221 数字量输入 (DI)

技术数据		SM 1221 DI 8 24 VDC	SM 1221 DI 16 24 VDC
产品编号		6ES7 221-1BF32-0XB0	6ES7 221-1BH32-0XB0
输入点数 (DI) 请参见规范 (页 420)。		8	16
尺寸 W x H x D (mm)		45 x 100 x 75	45 x 100 x 75
重量		170 g	210 g
功耗		1.5 W	2.5 W
电流消耗	SM 总线	105 mA	130 mA
	24 VDC	所用的每点输入 4 mA	所用的每点输入 4 mA

A.3 数字 I/O 模块

表格 A-19 SM 1221 数字量输入 (DI) 模块接线图



① 对于漏型输入，将“-”连接到“M”（如图所示）。对于源型输入，将“+”连接到“M”。

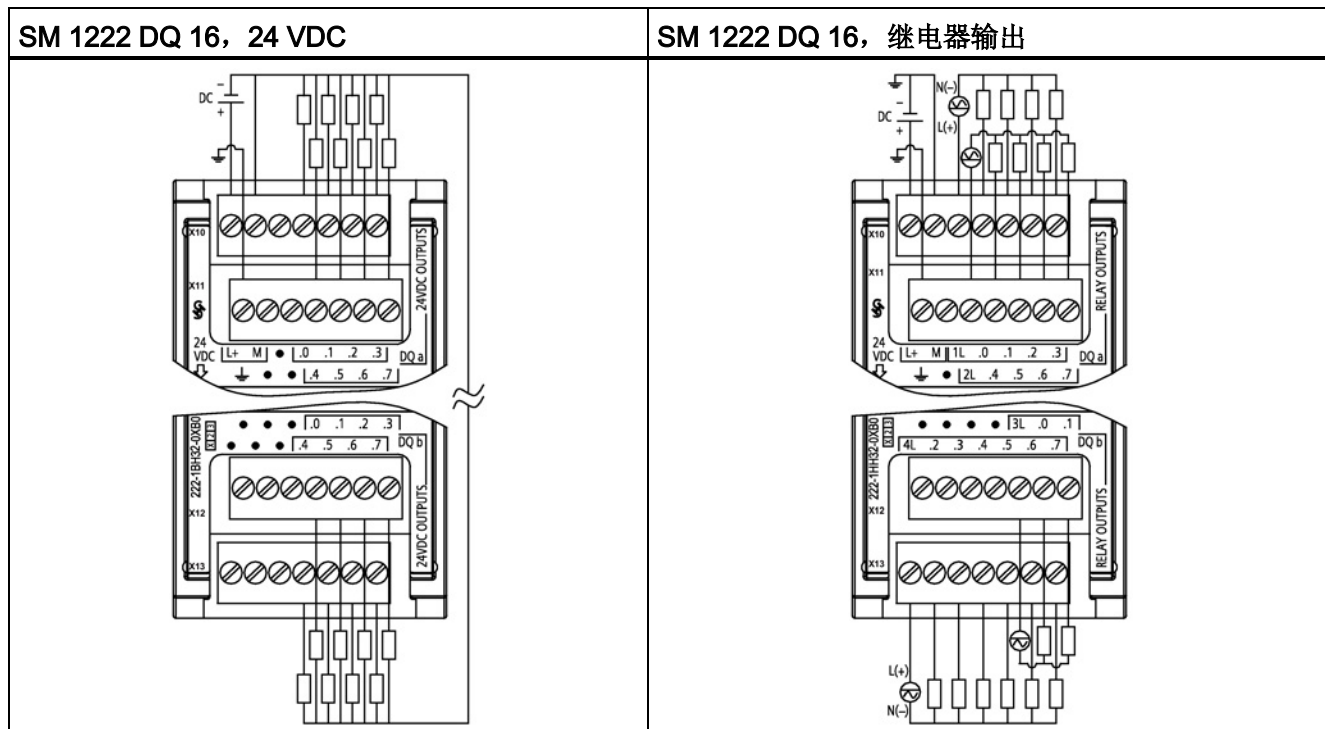
A.3.3 SM 1222 数字量输出 (DQ)

表格 A- 20 SM 1222 数字量输出 (DQ)

技术数据		SM 1222 DQ (继电器)	SM 1222 DQ (24 VDC)
产品编号		<ul style="list-style-type: none"> DQ 8: 6ES7 222-1HF32-0XB0 DQ 8: 切换: 6ES7 222-1XF32-0XB0 DQ 16: 6ES7 222-1HH32-0XB0 	<ul style="list-style-type: none"> DQ 8: 6ES7 222-1BF32-0XB0 DQ 16: 6ES7 222-1BH32-0XB0
输出点数 (DQ) 请参见规范 (页 420)。		<ul style="list-style-type: none"> 8 (DQ 8 和 DQ 8 转换) 16 (DQ 16) 	<ul style="list-style-type: none"> 8 (DQ 8) 16 (DQ 16)
尺寸 W x H x D (mm)		<ul style="list-style-type: none"> DQ 8 和 DQ 16: 45 x 100 x 75 DQ 8 转换: 70 x 100 x 75 	45 x 100 x 75
重量		<ul style="list-style-type: none"> DQ 8: 190 g DQ 8 转换: 310 g DQ 16: 260 g 	<ul style="list-style-type: none"> DQ 8: 180 g DQ 16: 220 g
功耗		<ul style="list-style-type: none"> DQ 8: 4.5 W DQ 8 转换: 5 W DQ 16: 8.5 W 	<ul style="list-style-type: none"> DQ 8: 1.5 W DQ 16: 2.5 W
电流消耗	SM 总线	<ul style="list-style-type: none"> DQ 8: 120 mA DQ 8 转换: 140 mA DQ 16: 135 mA 	<ul style="list-style-type: none"> DQ 8: 120 mA DQ 16: 140 mA
	24 VDC	<ul style="list-style-type: none"> DQ 8 和 DQ 16: 所用继电器线圈 11 mA DQ 8 转换: 所用的每个继电器线圈 16.7 mA 	<ul style="list-style-type: none"> DQ 8: -- DQ 16: --

A.3 数字 I/O 模块

表格 A-21 SM 1222 数字量输出 (DQ) 模块接线图



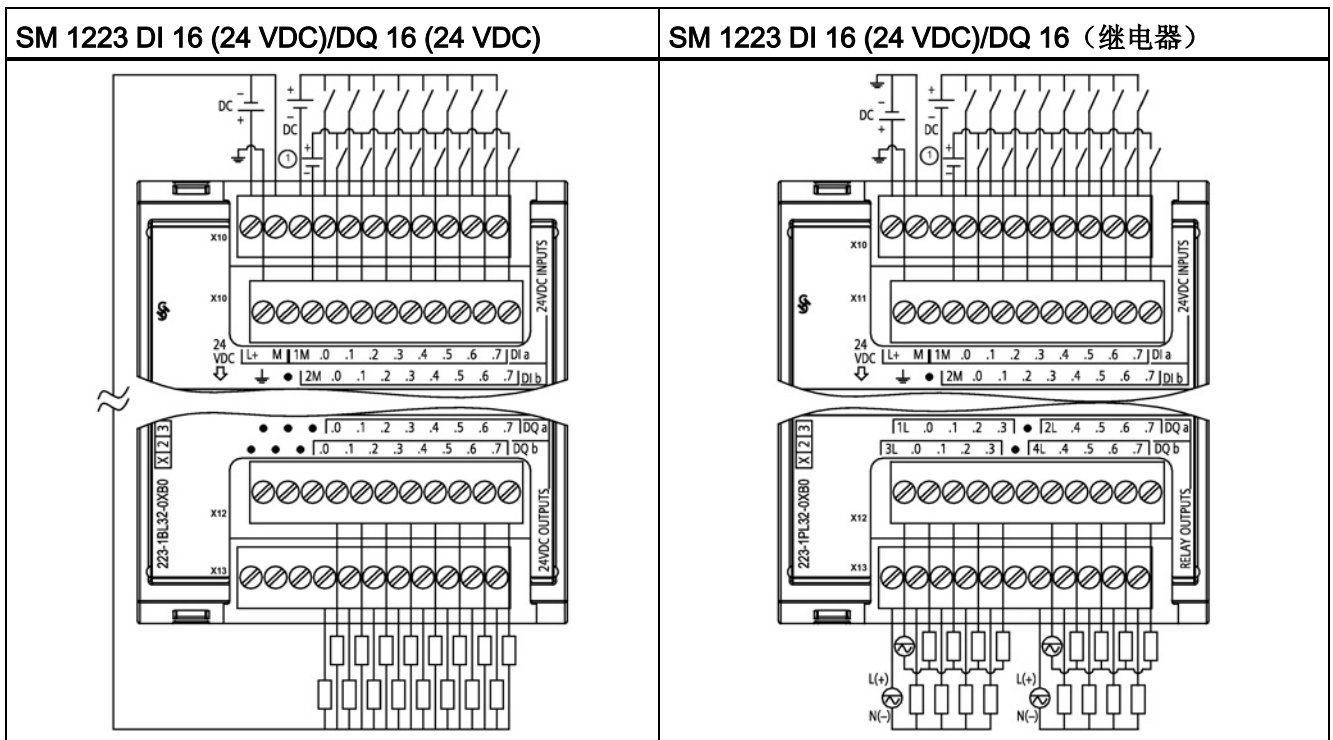
A.3.4 SM 1223 VDC 数字量输入/输出 (DI/DQ)

表格 A-22 SM 1223 数字量输入/输出组合 (DI/DQ)

技术数据	SM 1223 DI (24 VDC)/DQ (继电器)	SM 1223 DI (24 VDC)/DQ (24 VDC)
产品编号	DI 8/DQ 8: 6ES7 223-1PH32-0XB0 DI 16/DQ 16: 6ES7 223-1PL32-0XB0	DI 8/DQ 8: 6ES7 223-1BH32-0XB0 DI 8/DQ 8: 6ES7 223-1BL32-0XB0
输入/输出点数 (DI/DQ) 请参见规范 (页 420)。	<ul style="list-style-type: none"> • 输入: 8 或 16 (24 VDC) • 输出: 8 或 16 (继电器) 	<ul style="list-style-type: none"> • 输入: 8 或 16 (24 VDC) • 输出: 8 或 16 (24 VDC)
尺寸 W x H x D (mm)	<ul style="list-style-type: none"> • DI 8/DQ 8: 45 x 100 x 75 • DI 16/DQ 16: 70 x 100 x 75 	<ul style="list-style-type: none"> • DI 8/DQ 8: 45 x 100 x 75 • DI 16/DQ 16: 70 x 100 x 75
重量	<ul style="list-style-type: none"> • DI 8/DQ 8: 230 g • DI 16/DQ 16: 350 g 	<ul style="list-style-type: none"> • DI 8/DQ 8: 210 g • DI 16/DQ 16: 310 g

技术数据		SM 1223 DI (24 VDC)/DQ (继电器)	SM 1223 DI (24 VDC)/DQ (24 VDC)
功耗		<ul style="list-style-type: none"> DI 8/DQ 8: 5.5 W DI 16/DQ 16: 10 W 	<ul style="list-style-type: none"> DI 8/DQ 8: 2.5 W DI 16/DQ 16: 4.5 W
电流消耗	SM 总线	<ul style="list-style-type: none"> DI 8/DQ 8: 145 mA DI 16/DQ 16: 180 mA 	<ul style="list-style-type: none"> DI 8/DQ 8: 145 mA DI 16/DQ 16: 185 mA
	24 VDC	所用的每点输入 4 mA 所用的每个继电器线圈 11 mA	所用的每点输入 4 mA

表格 A-23 SM 1223 DI/DQ 组合模块的接线图



① 对于漏型输入，将“-”连接到“M”（如图所示）。对于源型输入，将“+”连接到“M”。

A.3 数字 I/O 模块

A.3.5 SM 1223 120/230 VAC 输入/继电器输出

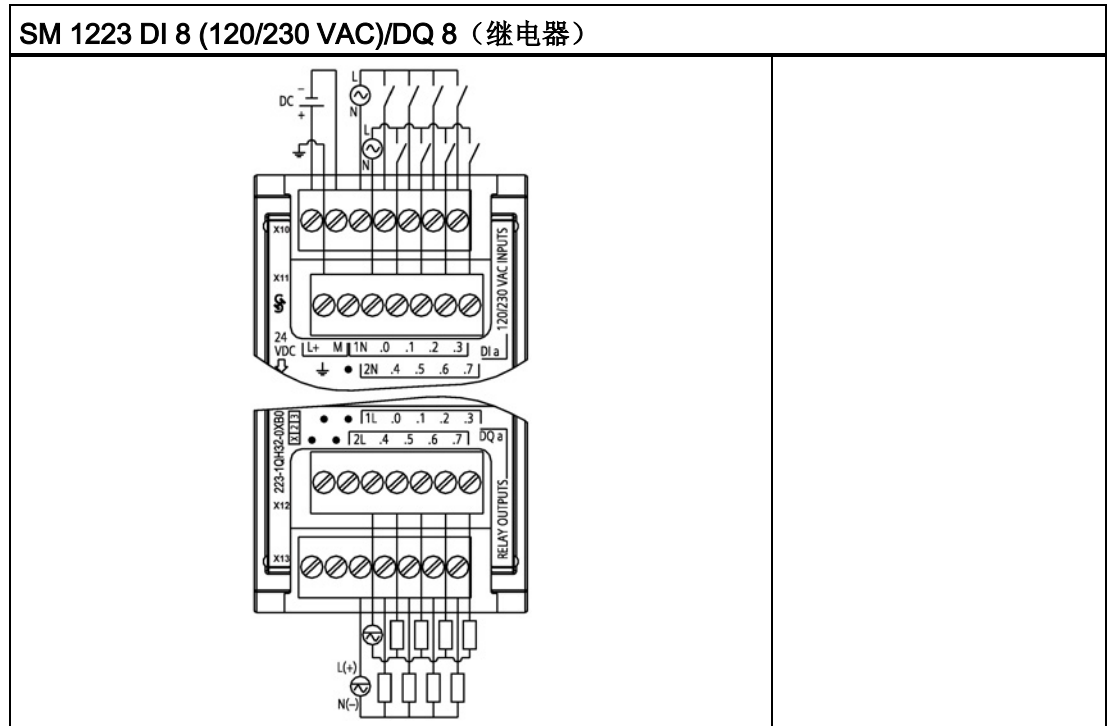
表格 A- 24 SM 1223 VAC 数字量输入/输出组合 (DI/DQ)

技术数据		SM 1223 DI (120/230 VDC)/DQ (继电器)
产品编号		DI 8/DQ 8: 6ES7 223-1QH32-0XB0
输入/输出点数 (DI/DQ)		输入: 8 (120/230 VAC) 有关 120/230 VAC 输入 (页 422), 请参见技术参数。 输出: 8 (继电器) 有关数字量输出 (页 423), 请参见技术参数。
尺寸 W x H x D (mm)		45 x 100 x 75
重量		190 g
功耗		7.5 W
电流消耗	SM 总线	120 mA
	24 VDC	所用的每个继电器线圈 11 mA

说明

SM 1223 DI 8 x 120/230 VAC, DQ 8 x 继电器信号模块 (6ES7 223-1QH32-0XB0)
获准用于 1 类, 2 分区, 气体组别 A、B、C、D, 温度类别 T4 Ta = 40 °C 中。

表格 A- 25 SM 1223 DI 8 (120/230 VAC)/DQ 8 (继电器) 接线图



A.4 数字量输入和输出的规范

A.4.1 24 VDC 数字量输入 (DI)

表格 A-26 数字量输入的规范 (DI)

技术数据	CPU、SM 和 SB	高速 SB (200 kHz)
类型	<ul style="list-style-type: none"> CPU 和 SM: IEC 类型 1 漏型 (漏型/源型) SB 1223: IEC 类型 1 漏型 (仅限漏型) 	SB 1221 200 kHz 和 SB 1223 200 kHz: 源型
额定电压	4 mA 时 24 VDC, 额定值	24 VDC SB: 7 mA 时 24 VDC, 额定值 5 VDC SB: 15 mA 时 5 VDC, 额定值
允许的连续电压	最大 30 VDC	24 VDC SB: 28.8 VDC 5 VDC SB: 6 VDC
浪涌电压	35 VDC, 持续 0.5 s	24 VDC SB: 35 VDC, 持续 0.5 s 5 VDC SB: 6 V
逻辑 1 信号 (最小)	2.5 mA 时 15 VDC	24 VDC SB: 2.9 mA 时 L+ - 10 VDC 5 VDC SB: 5.1 mA 时 L+ - 2.0 VDC
逻辑 0 信号 (最大)	1 mA 时 5 VDC	24 VDC SB: 1.4 mA 时 L+ - 5 VDC 5 VDC SB: 2.2 mA 时 L+ - 1.0 VDC
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 min	500 VAC, 持续 1 min
隔离组	<ul style="list-style-type: none"> CPU: 1 SM 1221 DI 8: 2 SM 1221 DI 16: 4 SB 1223 DI 2: 1 SM 1223: 2 	<ul style="list-style-type: none"> SB 1221 DI 4: 1 SB 1223 DI 2: 1
滤波时间	0.2、0.4、0.8、1.6、3.2、6.4 和 12.8 ms (可选择, 4 个为一组)	0.2、0.4、0.8、1.6、3.2、6.4 和 12.8 ms (可选择, 4 个为一组)

技术数据	CPU、SM 和 SB	高速 SB (200 kHz)
同时接通的输入数	<ul style="list-style-type: none"> • SM 1221 和 SM 1223 DI 8: 8 • SM 1221 和 SM 1223 DI 16: 16 • SB 1223 DI 2: 2 • CPU 1211C: 水平安装 60 °C 或垂直安装 50 °C 时为 6 个 • CPU 1212C: 水平安装 60 °C 或垂直安装 50 °C 时为 4 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 8 个 • CPU 1214C、CPU 1215C: 水平安装 60 °C 或垂直安装 50 °C 时为 7 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 14 个 • CPU 1217C: 水平安装 60 °C 或垂直安装 50 °C 时为 5 个漏型/源型输入 (无相邻点) 和 4 个不同的输入; 水平安装 55 °C 或垂直安装 45 °C 时为 14 个 	<ul style="list-style-type: none"> • SB 1221 DI 4: 4 • SB 1223 DI 2: 2
电缆长度 (米)	<ul style="list-style-type: none"> • 500 m (屏蔽); 300 m (非屏蔽) • CPU: 用于 HSC 的 50 m 屏蔽电缆 	50 m 屏蔽双绞线

说明

开关频率高于 20 kHz 时, 数字量输入接收方波甚为重要。

请考虑采取以下措施提高提供给输入的信号质量:

- 使电缆尽可能短
- 将纯漏型激励器换成漏型和源型混合的激励器
- 使用质量更好的电缆
- 将电路/组件从 24 V 减少到 5 V (如果产品以额定低电压运行。请参见“S7-1200 可编程控制器系统手册”中的完整规格)
- 在输入端连接外部负载

A.4 数字量输入和输出的规范

表格 A- 27 HSC 时钟输入频率（最大）

技术数据	单相	正交相位
CPU 1211C	100 kHz	80 kHz
CPU 1212C	100 kHz (Ia.0 到 Ia.5) 和 30 kHz (Ia.6 到 Ia.7)	80 kHz (Ia.0 到 Ia.5) 和 20 kHz (Ia.6 到 Ia.7)
CPU 1214C、CPU 1215C	100 kHz (Ia.0 到 Ia.5) 和 30 kHz (Ia.6 到 Ib.5)	80 kHz (Ia.0 到 Ia.5) 和 20 kHz (Ia.6 到 Ib.5)
CPU 1217C	1 MHz (Ib.2 到 Ib.5) 100 kHz (Ia.0 到 Ia.5) 30 kHz (Ia.6 到 Ib.1)	1 MHz (Ib.2 到 Ib.5) 80 kHz (Ia.0 到 Ia.5) 20 kHz (Ia.6 到 Ib.1)
高速 (200 kHz) SB	200 kHz	160 kHz
标准速度 SB	30 kHz	20 kHz

¹ 逻辑 1 电平 = 15 到 26 VDC

A.4.2 120/230 VAC 数字量 AC 输入

表格 A- 28 120/230 VAC 数字量输入

技术数据	SM
类型	IEC 1 类
额定电压	6 mA 时 120 VAC, 9 mA 时 230 VAC
允许的连续电压	264 VAC
浪涌电压	--
逻辑 1 信号（最小）	2.5 mA 时 79 VAC
逻辑 0 信号（最大）	1 mA 时 20 VAC
漏电流（最大值）	1 mA
隔离（现场侧与逻辑侧）	1500 VAC, 持续 1 min
隔离组 ¹	4
输入延迟时间	<ul style="list-style-type: none"> • 典型：0.2 到 12.8 ms, 用户可选择 • 最大值：--
连接 2 线制接近传感器 (Bero)（最大值）	1 mA

技术数据		SM
电缆长度	未屏蔽	300 米
	屏蔽	500 m
同时接通的输入数		8

¹ 组中通道的相位必须相同。

A.4.3 数字量输出 (DQ)

表格 A- 29 数字量输出的规范 (DQ)(DQ)

技术数据	继电器 (CPU 和 SM)	24 VDC (CPU、SM 和 SB)	200 kHz 24 VDC (SB)
类型	继电器, 干触点	固态 - MOSFET (源型)	固态 - MOSFET (漏型/源 型)
电压范围	5 到 30 VDC 或 5 到 250 VAC	20.4 到 28.8 VDC	20.4 到 28.8 VDC ¹ 4.25 到 6.0 VDC ²
最大电流时的逻辑 1 信号	--	最小 20 VDC	L+ - 1.5 V ¹ L+ - 0.7 V ²
具有 10 kΩ 负载时的逻辑 0 信号	--	CPU: 最小 20 VDC, 最大 0.1 VDC SB: 最大 0.1 VDC SM DC: 最大 0.1 VDC	最大 1.0 VDC ¹ 最大 0.2 VDC ²
电流 (最大)	2.0 A	0.5 A	0.1 A
灯负载	30 W DC/200 W AC	SB: 5 W	--
通态电阻	新设备最大为 0.2 Ω	最大 0.6 Ω	最大 11 Ω ¹ 或最大 7 Ω ²
断态电阻	--	--	最大 6 Ω ¹ 或最大 0.2 Ω ²
每点的漏泄电流	--	最大 10 μA	--

A.4 数字量输入和输出的规范

技术数据	继电器 (CPU 和 SM)	24 VDC (CPU、SM 和 SB)	200 kHz 24 VDC (SB)
脉冲串输出频率	CPU: N/A ³	CPU: 最大 100 kHz, 最小 2 Hz ⁴ SB: 最大 20 kHz, 最小 2 Hz ⁵	最大 200 kHz, 最小 2 Hz
浪涌电流	触点闭合时为 7 A	CPU: 8 A, 最长持续 100 ms SB: 5 A, 最长持续 100 ms SM: 8 A, 最长持续 100 ms	0.11 A
过载保护	×	×	×
隔离 (现场侧与逻辑侧)	线圈与触点: 1500 VAC, 持续 1 min 线圈与逻辑侧: 无	500 VAC, 持续 1 min	500 VAC, 持续 1 min
隔离组	<ul style="list-style-type: none"> • CPU 1211C: 1 • CPU 1212C: 2 • CPU 1214C: 2 • CPU 1215C: 2 • SM DQ 8: 2 • SM DQ 8 转换: 8 • SM DQ 16: 4 	<ul style="list-style-type: none"> • CPU: 1 • SB: 1 • SM (DQ 8): 1 • SM (DQ 16): 1 	1 ⁵
隔离电阻	新设备最小为 100 MΩ	--	--
断开触点间的绝缘	750 VAC, 持续 1 分钟	--	--
同时接通的输出数	CPU 1211C: 水平安装 60 °C 或垂直安装 50 °C 时为 4 个	--	--
	CPU 1212C: 水平安装 60 °C 或垂直安装 50 °C 时为 3 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 6 个	--	--
	CPU 1214C: 水平安装 60 °C 或垂直安装 50 °C 时为 5 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 10 个	--	--

技术数据	继电器 (CPU 和 SM)	24 VDC (CPU、SM 和 SB)	200 kHz 24 VDC (SB)
	CPU 1215C: 水平安装 60 °C 或垂直安装 50 °C 时为 5 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 10 个		--
	CPU 1217C: 水平安装 60 °C 或垂直安装 50 °C 时为 3 个固态 - MOSFET (源型) 输出 (无相邻点) 和 4 个不同的输出 水平安装 55 °C 或垂直安装 45 °C 时为 10 个		--
	SM 1222 DQ8: 8	SM 1222 DQ8: 8	--
	SM1222 DQ 8 转换: 水平安装 60 °C 或垂直安装 50 °C 时为 4 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 8 个		
	SM 1223 DI 8/DQ 8 继电器: 8		
	SM 1222 DQ16: 水平安装 60 °C 或垂直安装 50 °C 时为 8 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 16 个	SM 1222 DQ16: 16	--
	SM 1223 DI 16/DQ 16 继电器: 水平安装 60 °C 或垂直安装 50 °C 时为 8 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 16 个	SM 1223 DI 8/DQ 8: 8	

A.4 数字量输入和输出的规范

技术数据	继电器 (CPU 和 SM)	24 VDC (CPU、SM 和 SB)	200 kHz 24 VDC (SB)
	SM 1223 DI 8 x 120/230 VAC/DQ 8 继电器: 4x SM 1223 DI 16/DQ 16 继电器: 水平安装 60 °C 或垂直安装 50 °C 时为 4 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 8 个	SM 1223 DI 16/DQ 16: 16	
		SB 1223 DI 1 DQ 2、SM 1223 DI 2 DQ 2: 2	SB 1222 DQ 4: 水平安装 60 °C 或垂直安装 50 °C 时为 2 个 (无相邻点); 水平安装 55 °C 或垂直安装 45 °C 时为 4 个
			SB 1222 DQ 4 x 5 VDC: 4
			SB 1223 DI 2/DQ 2: 2
			SB 1223 DI 2/DQ 2: 2

技术数据	继电器 (CPU 和 SM)	24 VDC (CPU、SM 和 SB)	200 kHz 24 VDC (SB)
每个公共端的电流	SM 继电器: <ul style="list-style-type: none"> SM 1222 DQ 8 and DQ 16: 10 A SM 1222 DQ 8 转换: 2A SM 1223 DI 8/DQ 8:10 A SM 1223 DI 16/DQ 16: 8 A SM 1223 DI 8x120/230 VAC/DQ 8 继电器: 10 	SM 24 VDC <ul style="list-style-type: none"> SM 1222 DQ 16: 8 A SM 1223 DI 8/DQ 8: 4 A SM 1223 DI 16/DQ 16: 8 A 	--
电感钳位电压	--	L+ - 48 V, 1 W 损耗	无
继电器最大开关频率	1 Hz	--	--
开关延迟	最长 10 ms	CPU: <ul style="list-style-type: none"> Qa.0 到 Qa.3: 断开到接通最长为 1.0 μs; 接通到断开最长为 3.0 μs Qa.4 到 Qb.1: 断开到接通最长为 50 μs; 接通到断开最长为 200 μs SB: 断开到接通最长为 2 μ s; 接通到断开最长为 10 μ s SM: 断开到接通最长为 50 μ s; 接通到断开最长为 200 μ s	上升沿 1.5 μ s + 300 ns ¹ 下降沿 1.5 μ s + 300 ns ¹ 上升沿 200 ns + 300 ns ² 下降沿 200 ns + 300 ns ²
机械寿命 (无负载)	继电器: 10,000,000 个断开/闭合周期	--	--

A.4 数字量输入和输出的规范

技术数据	继电器 (CPU 和 SM)	24 VDC (CPU、SM 和 SB)	200 kHz 24 VDC (SB)
额定负载下的触点寿命	继电器: 100,000 个断开/闭合周期	--	--
RUN 到 STOP 时的行为	上一个值或替换值 (默 认为 0)	上一个值或替换值 (默认值 为 0)	上一个值或替换值 (默 认为 0)
电缆长度 (米)	500 m (屏蔽), 150 m (非屏蔽)	500 m (屏蔽), 150 m (非屏蔽)	50 m 屏蔽双绞线

- 1 24 VDC 200 kHz SB
- 2 5 VDC 200 kHz SB
- 3 对于具有继电器输出的 CPU 模块, 必须安装数字信号板 (SB), 以使用脉冲输出。
- 4 根据所使用的脉冲接收器和电缆的情况, 附加的负载电阻 (至少为额定电流的 10%) 可能改进脉冲信号质量和抗扰度。
- 5 SB 1223 200 kHz DI 2/DQ 2: 与输入无隔离

A.5 模拟量 I/O 模块

有关 S7-1200 可用模块的更完整列表，请参见“S7-1200

可编程控制器系统手册”或客户支持网站 (<http://www.siemens.com/tiaportal>)。

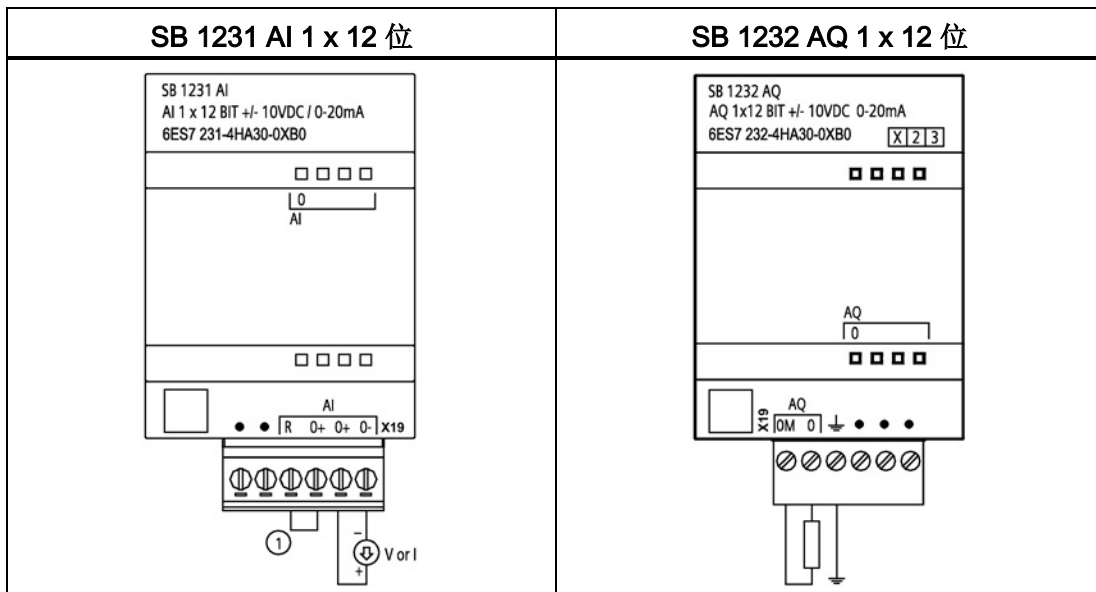
A.5.1 SB 1231 和 SB 1232 模拟量输入 (AI) 和输出 (AQ)

表格 A- 30 常规规范

技术数据	SB 1231 AI 1 x 12 位 ¹	SB 1232 AQ 1 x 12 位
产品编号	6ES7 231-4HA30-0XB0	6ES7 232-4HA30-0XB0
尺寸 W x H x D (mm)	38 x 62 x 21 mm	38 x 62 x 21 mm
重量	35 g	40 g
功耗	0.4 W	1.5 W
电流消耗 (SM 总线)	55 mA	15 mA
电流消耗 (24 VDC)	无	40 mA (无负载)
输入/输出点数	1	1
类型	电压或电流 (差动)	电压或电流

¹ 要使用 SB 1231 AI 1 x 模拟量输入，CPU 固件必须为 V2.0 或更高版本。

表格 A- 31 模拟量 SB 的接线图



① 对于电流，连接“R”和“0+”。

A.5.2 SM 1231 模拟量输入 (AI)

表格 A- 32 SM 1231 模拟量输入 (AI)

技术数据	SM 1231 AI 4 x 13 位	SM 1231 AI 8 x 13 位	SM 1231 AI 4 x 16 位
产品编号 (MLFB)	6ES7 231-4HD32-0XB0	6ES7 231-4HF32-0XB0	6ES7 231-5ND32-0XB0
输入点数	4 点输入 (AI)	8 点输入 (AI)	4 路输入
类型	电压或电流（差动），可选择，2 个为一组	电压或电流（差动），可选择，2 个为一组	电压或电流（差动）
尺寸 W x H x D (mm)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
重量	180 g	180 g	180 g
功耗	1.5 W	1.5 W	1.8 W
电流消耗 (SM 总线)	80 mA	90 mA	80 mA
电流消耗 (24 VDC)	45 mA	45 mA	65 mA

A.5.3 SM 1232 模拟量输出 (AQ)

表格 A- 33 SM 1232 模拟量输出 (AQ)

技术数据	SM 1232 AQ 2 x 14 位	SM 1232 AQ 4 x 14 位
产品编号 (MLFB)	6ES7 232-4HB32-0XB0	6ES7 232-4HD32-0XB0
输出点数和类型	2 点输出 (AQ)	4 点输出 (AQ)
尺寸 W x H x D (mm)	45 x 100 x 75	45 x 100 x 75
重量	180 g	180 g
功耗	1.5 W	1.5 W
电流消耗 (SM 总线)	80 mA	80 mA
电流消耗 (24 VDC)	45 mA (无负载)	45 mA (无负载)

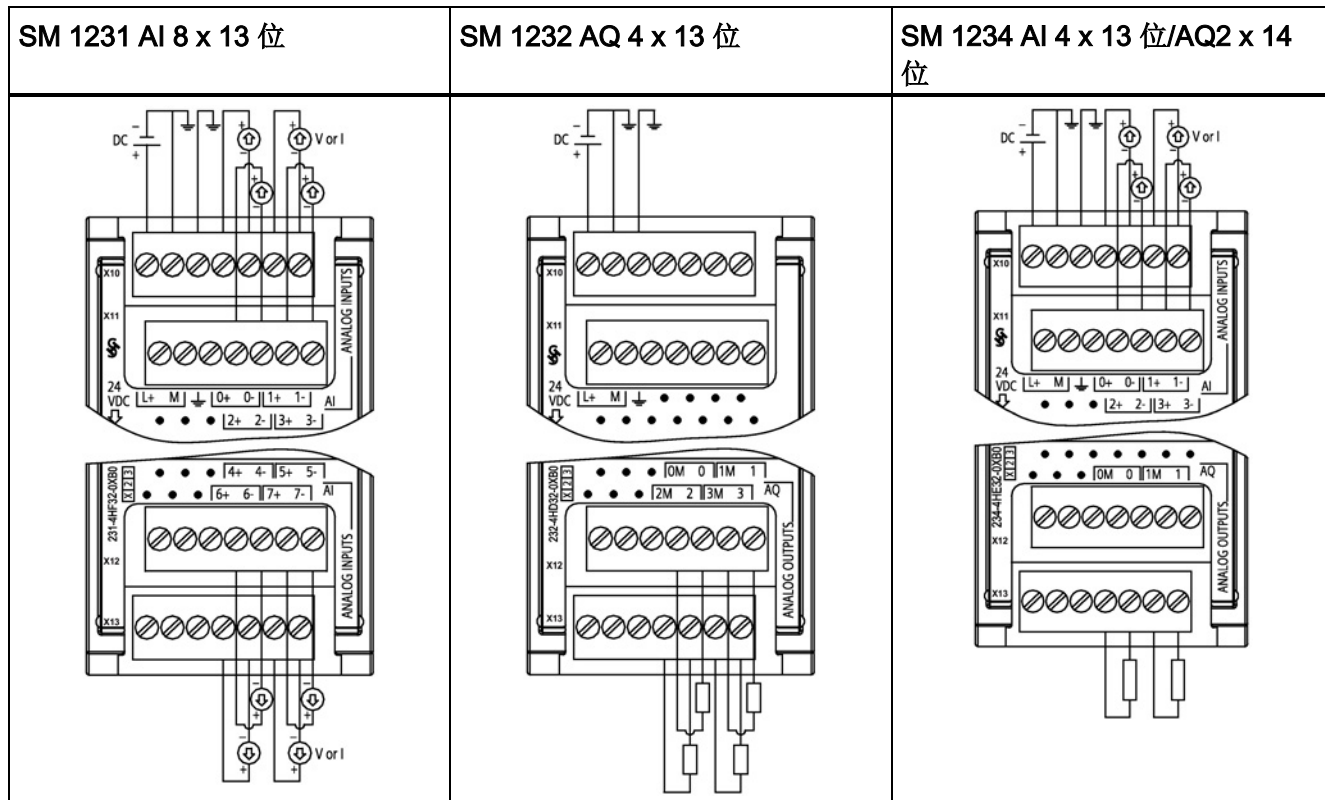
A.5.4 SM 1234 模拟量输入/输出 (AI/AQ)

表格 A- 34 SM 1234 组合模拟量输入/输出 (AI/AQ)

技术数据	SM 1234 AI 4 x 13 位/AQ 2 x 14 位
产品编号 (MLFB)	6ES7 234-4HE32-0XB0
输入点数 类型	4 点输入 (AI) 电压或电流 (差动), 可选择, 2 个为一组
输出点数 类型	2 点输出 (AQ) 电压或电流 (差动)
尺寸 W x H x D (mm)	45 x 100 x 75
重量	220 g
功耗	2.0 W
电流消耗 (SM 总线)	80 mA
电流消耗 (24 VDC)	60 mA (无负载)

A.5.5 SM 1231 (AI)、SM 1232 (AQ) 和 SM 1234 (AI/AQ) 的接线图

表格 A-35 模拟量 SM 的接线图



说明

应将未使用的电压输入通道短路。

应将未使用的电流输入通道设置为 0 到 20 mA 范围内的值，并且/或者禁用断线错误报告。

除非模块已上电且已组态，否则组态为电流模式的输入不会传导回路电流。

除非通过外部电源为发送器供电，否则电流输入通道不会工作。

A.6 BB 1297 电池板

BB 1297 电池板

表格 A- 36 常规规范

技术数据	BB 1297 电池
产品编号	6ES7 297-0AX30-0XA0
尺寸 W x H x D (mm)	38 x 62 x 21
重量	28 克
实时时钟保持时间	大约 1 年
电池类型	CR1025 ¹
CPU 的“维护”LED	指示需要更换电池。
用户程序	应用程序/系统可以评估电池状态

¹ 有关安装 BB 1297 或更换 BB 中电池的信息，请参见“S7-1200 可编程控制器系统手册”的第 2 章“安装”。

BB 1297 电池板用于实时时钟保持时间超过一个月的应用。BB 1297 电池板的功能如下：

- 在 PLC 关闭期间支持时间时钟。S7-1200 CPU 与 BB 1297 电池板结合使用，在应用关闭期间可支持最多一年的时钟保持时间。
- 一次只能使用一个 BB 1297 电池板或其它 SB。
- 不允许热插拔/热交换。仅当 CPU 关闭时，才能交换或插入 BB 1297 电池板。当关闭 CPU 并卸下 BB 1297 以交换实际电池时，内部超级电容将在用户更换电池时保持时间。
- CPU“维护”LED 在需要更换电池时会发出指示。
- 用户程序可用来监视或检查电池和电池板的状态，并允许在 HMI 或 Web 服务器上显示用户消息。

A.7 模拟量 I/O 的规范

A.7.1 模拟量输入（CPU、SM 和 SB）的规范

表格 A-37 模拟量输入 (AI) 的规范

技术数据	CPU	SB	SM
类型	电压（单侧）	电压或电流（差动）	电压或电流（差动），可选择，2 个为一组
范围	0 到 10 V	±10 V、±5 V、±2.5 V、 0 到 20 mA，或 4 mA 到 20 mA	±10 V、±5 V、±2.5 V、 0 到 20 mA，或 4 mA 到 20 mA
分辨率	10 位	11 位 + 符号位	12 位 + 符号位
满量程范围 （数据字）	0 到 27648	-27,648 到 27,648	-27,648 到 27,648
精度 （25 °C/-20 到 60 °C）	满量程的 3.0%/3.5%	满量程的 ±0.3%/±0.6%	满量程的 ±0.1%/±0.2%
过冲/下冲范围（数据字） （请参见注 1）	电压： 27,649 到 32,511	电压： 32,511 到 27,649/ -27,649 到 -32,512	电压： 32,511 到 27,649/ -27,649 到 -32,512
	电流： 不适用	电流： 32,511 到 27,649/ 0 到 -4864	电流： 32,511 到 27,649/ 0 到 -4864
上溢/下溢（数据字） （请参见注 1）	电压： 32,512 到 32,767	电压： 32,767 到 32,512/ -32,513 到 -32,768	电压： 32,767 到 32,512/ -32,513 到 -32,768
	电流： 不适用	电流： 32,767 到 32,512/ -4865 到 -32,768	电流： 32,767 到 32,512/ -4865 到 -32,768
最大耐压/耐流	35 VDC（电压）	±35 V/±40 mA	±35 V/±40 mA
平滑化 （请参见注 2）	无、弱、中或强	无、弱、中或强	无、弱、中或强
噪声抑制 （请参见注 2）	10、50 或 60 Hz	400、60、50 或 10 Hz	400、60、50 或 10 Hz

技术数据	CPU	SB	SM
测量原理	实际值转换	实际值转换	实际值转换
共模抑制	无	40 dB, DC 到 60 Hz	40 dB, DC 到 60 Hz
工作信号范围 (信号电压加共模电压)	小于 +12 V 且大于 0 V	小于 +35 V 且大于 -35 V	小于 +12 V 且大于 -12 V
负载阻抗	单端: $\geq 100\text{ K}\Omega$	差值: 220 $\text{K}\Omega$ (电压), 250 Ω (电流) 共模: 55 $\text{K}\Omega$ (电压), 55 $\text{K}\Omega$ (电流)	差值: 9 $\text{M}\Omega$ (电压), 250 Ω (电流) 共模: 4.5 $\text{M}\Omega$ (电压), 4.5 $\text{M}\Omega$ (电流)
隔离 (现场侧与逻辑侧)	无	无	无
电缆长度 (米)	100 m, 屏蔽双绞线	100 m, 屏蔽双绞线	100 m 屏蔽双绞线
诊断	上溢/下溢	上溢/下溢	上溢/下溢 24 VDC 低压
注 1: 请参见模拟量输入电压和电流的测量范围 (页 435), 确定过冲/下冲以及上溢/下溢范围。			
注 2: 请参见阶跃响应时间 (页 437), 确定平滑化和噪声抑制值。			

A.7.2 输入 (AI) 的电压和电流测量范围

表格 A- 38 模拟量输入的电压表示法 (SB 和 SM)

系统		电压测量范围				
十进制	十六进制	$\pm 10\text{ V}$	$\pm 5\text{ V}$	$\pm 2.5\text{ V}$	$\pm 1.25\text{ V}$	
32767	7FFF ¹	11.851 V	5.926 V	2.963 V	1.481 V	上溢
32512	7F00					
32511	7EFF	11.759 V	5.879 V	2.940 V	1.470 V	过冲范围
27649	6C01					
27648	6C00	10 V	5 V	2.5 V	1.250 V	额定范围
20736	5100	7.5 V	3.75 V	1.875 V	0.938 V	
1	1	361.7 μV	180.8 μV	90.4 μV	45.2 μV	
0	0	0 V	0 V	0 V	0 V	

A.7 模拟量 I/O 的规范

系统		电压测量范围				
十进制	十六进制	±10 V	±5 V	±2.5 V	±1.25 V	
-1	FFFF					
-20736	AF00	-7.5 V	-3.75 V	-1.875 V	-0.938 V	
-27648	9400	-10 V	-5 V	-2.5 V	-1.250 V	
-27649	93FF					下冲范围
-32512	8100	-11.759 V	-5.879 V	-2.940 V	-1.470 V	下溢
-32513	80FF					
-32768	8000	-11.851 V	-5.926 V	-2.963 V	-1.481 V	

1 返回 7FFF

可能由以下原因之一所致：上溢（如该表中所述），有效值可用前（例如刚上电时）或者检测到断线时。

表格 A- 39 模拟量输入的电流表示法（SB 和 SM）

系统		电流测量范围		
十进制	十六进制	0 mA 到 20 mA	4 mA 到 20 mA	
32767	7FFF	23.70 mA	22.96 mA	上溢
32512	7F00			
32511	7EFF	23.52 mA	22.81 mA	过冲范围
27649	6C01			
27648	6C00	20 mA	20 mA	额定范围
20736	5100	15 mA	16 mA	
1	1	723.4 nA	4 mA + 578.7 nA	
0	0	0 mA	4 mA	
-1	FFFF			下冲范围
-4864	ED00	-3.52 mA	1.185 mA	
-4865	ECFF			下溢
-32768	8000			

表格 A-40 模拟量输入的电压表示法 (CPU 1215C 和 CPU 1217C)

系统		电压测量范围	
十进制	十六进制	0 到 10 V	
32767	7FFF	11.851 V	溢出
32512	7F00		
32511	7EFF	11.759 V	过冲范围
27649	6C01		
27648	6C00	10 V	额定范围
20736	5100	7.5 V	
34	22	12 mV	
0	0	0 V	
负值		不支持负值	

A.7.3 模拟量输入 (AI) 的阶跃响应

下表显示了 CPU、SB 和 SM 的模拟量输入 (AI) 的阶跃响应时间。

表格 A-41 模拟量输入的阶跃响应 (ms)

平滑化选项 (采样平均)		积分时间选项			
		400 Hz (2.5 ms)	60 Hz (16.6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
无 (1 个周期) : 不求平均值	CPU	不适用	63	65	130
	SB	4.5	18.7	22.0	102
	SM	4	18	22	100
弱 (4 个周期) : 4 次采样	CPU	不适用	84	93	340
	SB	10.6	59.3	70.8	346
	SM	9	52	63	320
中 (16 个周期) : 16 次采样	CPU	不适用	221	258	1210
	SB	33.0	208	250	1240
	SM	32	203	241	1200

平滑化选项（采样平均）		积分时间选项			
		400 Hz (2.5 ms)	60 Hz (16.6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
强（32 个周期）：32 次采样	CPU	不适用	424	499	2410
	SB	63.0	408	490	2440
	SM	61	400	483	2410
采样速率	CPU	不适用	4.17	5	25
	SB	0.156	1.042	1.250	6.250

A.7.4 模拟量输入的采样时间和更新时间

表格 A-42 SM 和 CPU 的采样时间和更新时间

抑制频率（积分时间）	采样时间	所有通道的更新时间		
		4 通道 SM	8 通道 SM	CPU AI
400 Hz (2.5 ms)	0.625 ms ¹	2.5 ms	10 ms	不适用
60 Hz (16.6 ms)	4.170 ms	4.17 ms	4.17 ms	4.17 ms
50 Hz (20 ms)	5.000 ms	5 ms	5 ms	5 ms
10 Hz (100 ms)	25.000 ms	25 ms	25 ms	25 ms

¹ 8 通道 SM 的采样频率为 1.250 ms。

表格 A-43 SB 的采样时间和更新时间

抑制频率（积分时间）	采样时间	SB 更新时间
400 Hz (2.5 ms)	0.156 ms	0.156 ms
60 Hz (16.6 ms)	1.042 ms	1.042 ms
50 Hz (20 ms)	1.250 ms	1.25 ms
10 Hz (100 ms)	6.250 ms	6.25 ms

A.7.5 模拟量输出的规范

表格 A-44 模拟量输出 (SB 和 SM) 的规范

技术数据	SB	SM
类型	电压或电流	电压或电流
范围	$\pm 10\text{ V}$ 、0 到 20 mA 或 4 到 20 mA	$\pm 10\text{ V}$ 、0 到 20 mA 或 4 到 20 mA
分辨率	电压: 12 位 电流: 11 位	电压: 14 位 电流: 13 位
满量程范围 (数据字) (请参见注 1)	电压: -27,648 到 27,648 电流: 0 到 27,648	电压: -27,648 到 27,648 电流: 0 到 27,648
精度 (25 °C/-20 到 60 °C)	满量程的 $\pm 0.5\%/ \pm 1\%$	满量程的 $\pm 0.3\%/ \pm 0.6\%$
稳定时间 (新值的 95%)	电压: 300 μS (R), 750 μS (1 μF) 电流: 600 μS (1 mH), 2 ms (10 mH)	电压: 300 μS (R), 750 μS (1 μF) 电流: 600 μS (1 mH), 2 ms (10 mH)
负载阻抗	电压: $\geq 1000\ \Omega$ 电流: $\leq 600\ \Omega$	电压: $\geq 1000\ \Omega$ 电流: $\leq 600\ \Omega$
RUN 到 STOP 时的行为	上一个值或替换值 (默认值为 0)	上一个值或替换值 (默认值为 0)
隔离 (现场侧与逻辑侧)	无	无
电缆长度 (米)	100 m, 屏蔽双绞线	100 m, 屏蔽双绞线
诊断	<ul style="list-style-type: none"> • 上溢/下溢 • 对地短路 (仅限电压模式) • 断线 (仅限电流模式) 	<ul style="list-style-type: none"> • 上溢/下溢 • 对地短路 (仅限电压模式) • 断线 (仅限电流模式) • 24 VDC 低压
注 1: 请参见满量程范围的电压和电流 (页 440) 输出范围。		

A.7.6 输出 (AQ) 的电压和电流测量范围

表格 A- 45 模拟量输出的电压表示法 (SB 和 SM)

系统		电压输出范围	
十进制	十六进制	±10 V	
32767	7FFF	请参见注 1	上溢
32512	7F00	请参见注 1	
32511	7EFF	11.76 V	过冲范围
27649	6C01		
27648	6C00	10 V	额定范围
20736	5100	7.5 V	
1	1	361.7 μV	
0	0	0 V	
-1	FFFF	-361.7 μV	
-20736	AF00	-7.5 V	
-27648	9400	-10 V	
-27649	93FF		
-32512	8100	-11.76 V	
-32513	80FF	请参见注 1	下溢
-32768	8000	请参见注 1	

1 在上溢或下溢情况下，模拟量输出将采用 STOP 模式的替代值。

表格 A- 46 模拟量输出的电流表示法 (SB 和 SM)

系统		当前输出范围		
十进制	十六进制	0 mA 到 20 mA	4 mA 到 20 mA	
32767	7FFF	请参见注 1	请参见注 1	上溢
32512	7F00	请参见注 1	请参见注 1	
32511	7EFF	23.52 mA	22.81 mA	过冲范围
27649	6C01			
27648	6C00	20 mA	20 mA	额定范围

系统		当前输出范围		
十进制	十六进制	0 mA 到 20 mA	4 mA 到 20 mA	
20736	5100	15 mA	16 mA	
1	1	723.4 nA	4 mA + 578.7 nA	
0	0	0 mA	4mA	
-1	FFFF		4 mA 到 578.7 nA	下冲范围
-6912	E500		0 mA	不可能。输出值限制在 0 mA。
-6913	E4FF			
-32512	8100			
-32513	80FF	请参见注 1	请参见注 1	下溢
-32768	8000	请参见注 1	请参见注 1	

1 在上溢或下溢情况下，模拟量输出将采用 STOP 模式的替代值。

表格 A- 47 模拟量输出的电流表示法（CPU 1215C 和 CPU 1217C）

系统		当前输出范围	
十进制	十六进制	0 mA 到 20 mA	
32767	7FFF	请参见注 1	上溢
32512	7F00	请参见注 1	
32511	7EFF	23.52 mA	过冲范围
27649	6C01		额定范围
27648	6C00	20 mA	
20736	5100	15 mA	
34	22	0.0247 mA	
0	0	0 mA	
负值		不支持负值	

1 在上溢情况下，模拟量输出的行为将符合设备组态属性设置。在“对 CPU STOP 的响应”参数中，选择其中一项：“使用替换值”或“保持上一个值”。

A.8 RTD 模块和热电偶模块

热电偶 (TC) 模块 (SB 1231 TC 和 SM 1231 TC) 测量连接到模拟量输入的电压值。该值可以是 TC 测出的温度，也可以是电压。

- 如果是电压，额定范围的满量程值将是十进制数 27648。
- 如果是温度，则将度数乘 10 得到该值 (例如，25.3 度将报告为十进制数 253)。

RTD 模块 (SB 1231 RTD 和 SM 1231 RTD) 测量连接到模拟量输入的电阻值。该值可以是温度，也可以是电阻。

- 如果是电阻，额定范围的满量程值将是十进制数 27648。
- 如果是温度，则将度数乘 10 得到该值 (例如，25.3 度将报告为十进制数 253)。

RTD 模块支持采用 2 线制、3 线制和 4 线制方式连接到传感器电阻进行测量。

说明

对于没有连接传感器的激活通道，RTD 和 TC 模块都将报告 32767。

如果还启用了开路检测，模块会使相应的红色 LED 闪烁。

若使用 4 线制连接，对于 10 Ω RTD 范围，将得到最高精度。

2 线制模式的连接线电阻会导致传感器读数误差，因此无法保证精度。

说明

对模块上电后，模块将对模数转换器执行内部校准。

在此期间，模块将报告每个通道的值为 32767，直到相应通道出现有效值为止。

用户程序可能需要考虑这段初始化时间。

由于模块的组态可能改变初始化时长，因此，应验证组态中模块的行为。

如果需要，可以在用户程序中包含逻辑，以适应模块的初始化时间。

A.8.1 SB 1231 RTD 和 SB 1231 TC 规范

说明

要使用这些 TC 和 RTD SB，CPU 固件必须为 V2.0 或更高版本。

表格 A- 48 常规规范

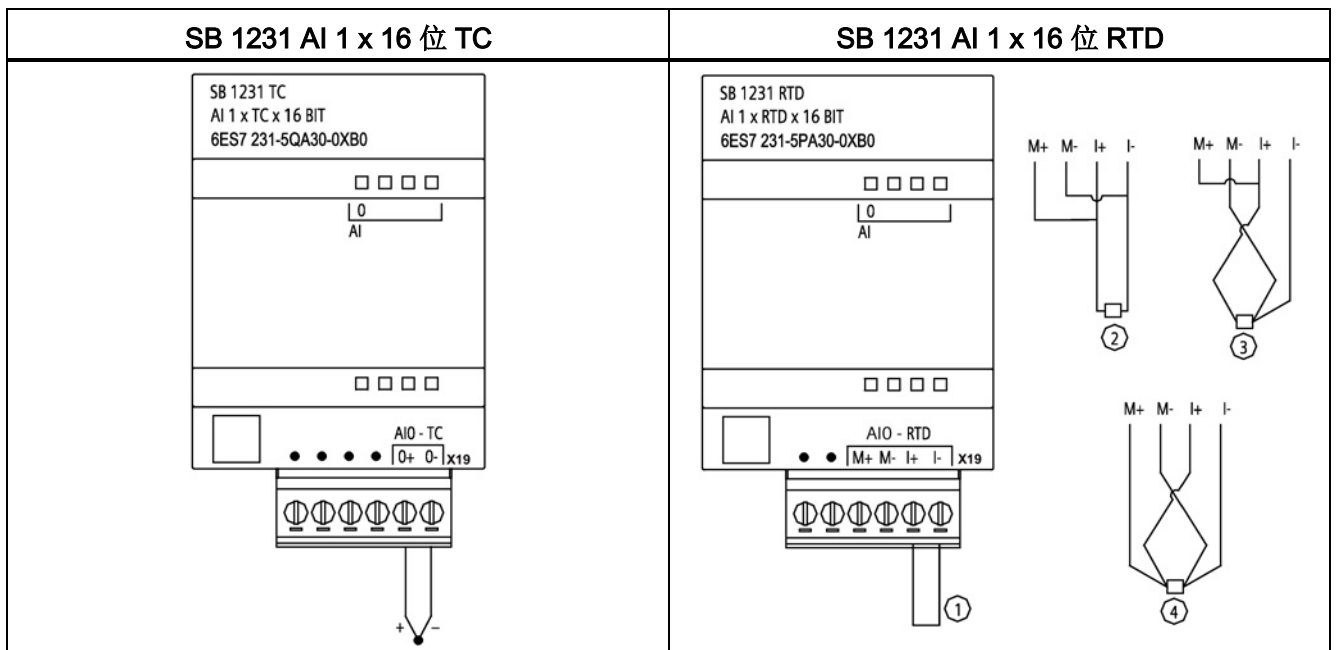
技术数据	SB 1231 AI 1 x 16 位 TC	SB 1231 AI 1 x 16 位 RTD
产品编号	6ES7 231-5QA30-0XB0	6ES7 231-5PA30-0XB0
尺寸 W x H x D (mm)	38 x 62 x 21 mm	38 x 62 x 21 mm
重量	35 g	35 g
功耗	0.5 W	0.7 W
电流消耗 (SM 总线)	5 mA	5 mA
电流消耗 (24 VDC)	20 mA	25 mA
输入 (页 447) 点数	1	1
类型	浮动 TC 和 mV	模块参考 RTD 和 Ω
诊断	<ul style="list-style-type: none"> 上溢/下溢^{1 2} 断线³ 	<ul style="list-style-type: none"> 上溢/下溢^{1 2} 断线³

1 上溢和下溢诊断报警信息将以模拟数据值的形式报告，即使在模块组态中禁用这些报警也会如此。

2 RTD: 对于电阻范围，始终会禁用下溢检测。

3 如果断线报警已禁用，但传感器接线存在开路情况，则模块可能会报告随机值。

表格 A- 49 SB 1231 TC 和 RTD 的接线图



① 环接未使用的 RTD 输入

② 2 线制 RTD ③ 3 线制 RTD ④ 4 线制 RTD

A.8.2 SM 1231 RTD 规范

表格 A- 50 常规规范

技术数据	SM 1231 AI 4 x RTD x 16 位	SM 1231 AI 8 x RTD x 16 位
产品编号	6ES7 231-5PD32-0XB0	6ES7 231-5PF32-0XB0
尺寸 W x H x D (mm)	45 x 100 x 75	70 x 100 x 75
重量	220 g	270 g
功耗	1.5 W	1.5 W
电流消耗 (SM 总线)	80 mA	90 mA
电流消耗 ¹ (24 VDC)	40 mA	40 mA
输入 (页 447) 点数 类型	4 模块参考 RTD 和 Ω	8 模块参考 RTD 和 Ω
诊断	<ul style="list-style-type: none"> • 上溢/下溢^{2 3} • 24 VDC 低压² • 断线 (仅限电流模式)⁴ 	<ul style="list-style-type: none"> • 上溢/下溢^{2 3} • 24 VDC 低压² • 断线 (仅限电流模式)⁴

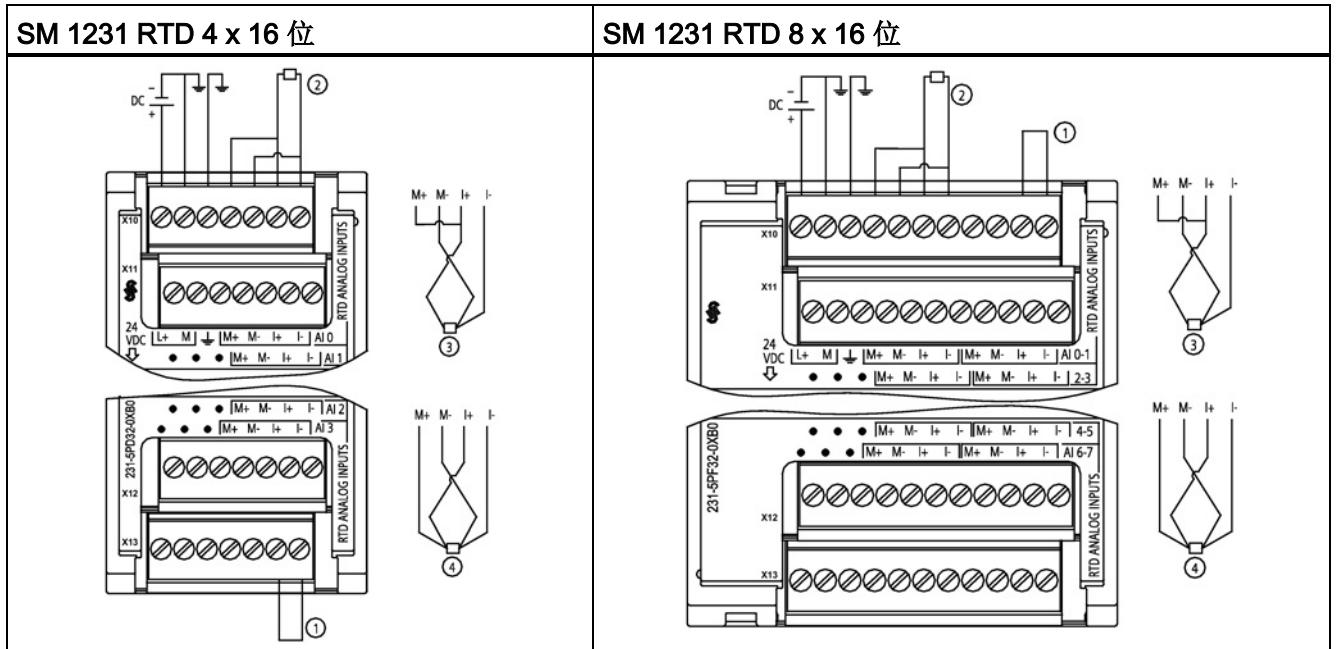
¹ 20.4 到 28.8 VDC (2 类受限制电源, 或 CPU 模块提供的传感器电源)

² 上溢、下溢和低压诊断报警信息将以模拟数据值的形式报告, 即使在模块组态中禁用这些报警也会如此。

³ 对于电阻范围, 始终会禁用下溢检测。

⁴ 如果断线报警已禁用, 但传感器接线存在开路情况, 则模块可能会报告随机值。

表格 A- 51 RTD SM 的接线图



① 环接未使用的 RTD 输入

② 2 线制 RTD

③ 3 线制 RTD

④ 4 线制 RTD

注：连接器必须镀金。请参见“S7-1200 可编程控制器系统手册”的附录 C。

A.8.3 SM 1231 TC 规范

表格 A- 52 常规规范

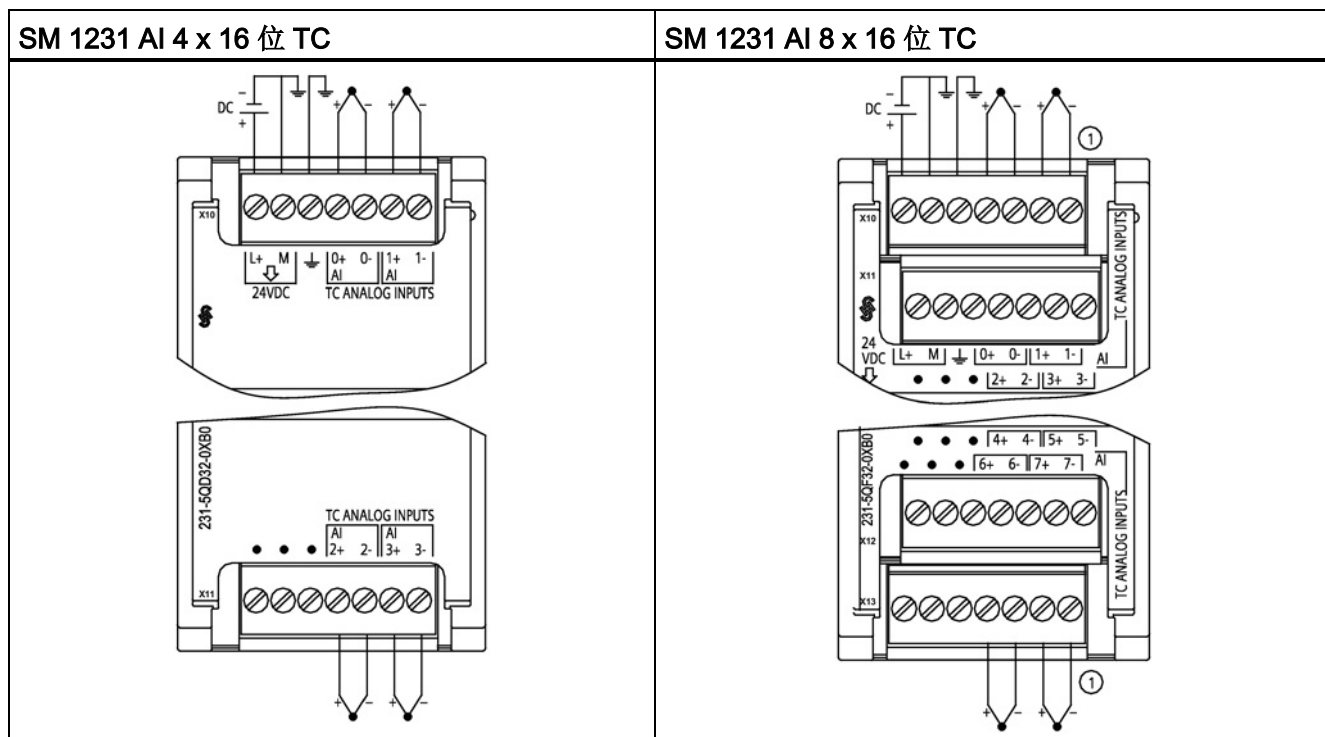
型号	SM 1231 AI 4 x 16 位 TC	SM 1231 AI 8 x 16 位 TC
产品编号	6ES7 231-5QD32-0XB0	6ES7 231-5QF32-0XB0
尺寸 W x H x D (mm)	45 x 100 x 75	45 x 100 x 75
重量	180 g	xxx g
功耗	1.5 W	1.5 W
电流消耗 (SM 总线)	80 mA	80 mA

A.8 RTD 模块和热电偶模块

型号	SM 1231 AI 4 x 16 位 TC	SM 1231 AI 8 x 16 位 TC
电流消耗 ¹ (24 VDC)	40 mA	40 mA
输入 (页 447) 点数 类型	4 浮动 TC 和 mV	8 浮动 TC 和 mV
诊断	<ul style="list-style-type: none"> 上溢/下溢² 24 VDC 低压² 断线 (仅限电流模式)³ 	<ul style="list-style-type: none"> 上溢/下溢² 24 VDC 低压² 断线 (仅限电流模式)³

- 1 20.4 到 28.8 VDC (2 类受限制电源, 或 CPU 模块提供的传感器电源)
- 2 上溢、下溢和低压诊断报警信息将以模拟数据值的形式报告, 即使在模块组态中禁用这些报警也会如此。
- 3 如果断线报警已禁用, 但传感器接线存在开路情况, 则模块可能会报告随机值。

表格 A-53 TC SM 的接线图



① SM 1231 AI 8 TC: 为清晰起见, 未显示 TC 2、3、4 和 5 的连接。

A.8.4 RTD 和 TC (SM 和 SB) 的模拟量输入规范

表格 A- 54 RTD 和 TC 模块 (SB 和 SM) 的模拟量输入

技术数据		RTD 和热电偶 (TC)
输入点数		1 (SB)、4 或 8 (SM)
类型		<ul style="list-style-type: none"> • RTD: 模块参考 RTD 和 Ω • TC: 浮动 TC 和 mV
范围		请参见 RTD/TC 类型表: <ul style="list-style-type: none"> • RTD (页 450) • TC (页 448)
分辨率	温度	0.1 °C/0.1 °F
	电阻/电压	15 位 + 符号
最大耐压		± 35 V
噪声抑制		对于所选滤波器设置 (10 Hz、50 Hz、60 Hz 或 400 Hz) 为 85 dB
共模抑制		120 VAC 时大于 120 dB
阻抗		≥ 10 M Ω
隔离	现场侧与逻辑侧	500 VAC
	现场侧与 24 VDC	SM RTD 和 SM TC: 500 VAC (不适用于 SB RTD 和 SB TC)
	24 VDC 与逻辑侧	SM RTD 和 SM TC: 500 VAC (不适用于 SB RTD 和 SB TC)
通道间隔离		<ul style="list-style-type: none"> • SM RTD: 无 (不适用于 SB RTD) • SM TC: 120 VAC (不适用于 SB TC)
精度 (25 °C/-20 到 60 °C)		请参见 RTD/TC 类型表: <ul style="list-style-type: none"> • RTD (页 450) • TC (页 448)
可重复性		$\pm 0.05\%$ FS

A.8 RTD 模块和热电偶模块

技术数据	RTD 和热电偶 (TC)
最大传感器功耗	<ul style="list-style-type: none"> • RTD: 0.5 mW • TC: 不适用
测量原理	积分型
模块更新时间	请参见 RTD/TC 滤波器选型表: <ul style="list-style-type: none"> • RTD (页 452) • TC (页 449)
冷端误差	<ul style="list-style-type: none"> • RTD: 不适用 • TC: $\pm 1.5\text{ }^{\circ}\text{C}$
电缆长度 (米)	到传感器最长为 100 米
导线电阻	<ul style="list-style-type: none"> • RTD: $20\ \Omega$, 对于 $10\ \Omega$ RTD, 最大为 $2.7\ \Omega$ • TC: 最大 $100\ \Omega$

A.8.5 热电偶类型

表格 A-55 热电偶类型 (范围和精度)

类型	低于范围最小值 ¹	额定范围下限	额定范围上限	超出范围最大值 ²	25 °C 时的额定范围 ^{3, 4} 精度	-20 °C 到 60 °C 时的额定范围 ^{3, 4} 精度
J	-210.0 °C	-150.0 °C	1200.0 °C	1450.0 °C	$\pm 0.3\text{ }^{\circ}\text{C}$	$\pm 0.6\text{ }^{\circ}\text{C}$
K	-270.0 °C	-200.0 °C	1372.0 °C	1622.0 °C	$\pm 0.4\text{ }^{\circ}\text{C}$	$\pm 1.0\text{ }^{\circ}\text{C}$
T	-270.0 °C	-200.0 °C	400.0 °C	540.0 °C	$\pm 0.5\text{ }^{\circ}\text{C}$	$\pm 1.0\text{ }^{\circ}\text{C}$
E	-270.0 °C	-200.0 °C	1000.0 °C	1200.0 °C	$\pm 0.3\text{ }^{\circ}\text{C}$	$\pm 0.6\text{ }^{\circ}\text{C}$
R & S	-50.0 °C	100.0 °C	1768.0 °C	2019.0 °C	$\pm 1.0\text{ }^{\circ}\text{C}$	$\pm 2.5\text{ }^{\circ}\text{C}$
B	0.0 °C	200.0 °C	800.0 °C	--	$\pm 2.0\text{ }^{\circ}\text{C}$	$\pm 2.5\text{ }^{\circ}\text{C}$
	--	800.0 °C	1820.0 °C	1820.0 °C	$\pm 1.0\text{ }^{\circ}\text{C}$	$\pm 2.3\text{ }^{\circ}\text{C}$
N	-270.0 °C	-200.0 °C	1300.0 °C	1550.0 °C	$\pm 1.0\text{ }^{\circ}\text{C}$	$\pm 1.6\text{ }^{\circ}\text{C}$
C	0.0 °C	100.0 °C	2315.0 °C	2500.0 °C	$\pm 0.7\text{ }^{\circ}\text{C}$	$\pm 2.7\text{ }^{\circ}\text{C}$

类型	低于范围最小值 ¹	额定范围下限	额定范围上限	超出范围最大值 ²	25 °C 时的额定范围 ^{3, 4} 精度	-20 °C 到 60 °C 时的额定范围 ^{3, 4} 精度
TXK/XK(L)	-200.0 °C	-150.0 °C	800.0 °C	1050.0 °C	±0.6 °C	±1.2 °C
电压	-32512	-27648 -80 mV	27648 80 mV	32511	±0.05%	±0.1%

- “低于范围最小值”以下的热电偶值报告为 -32768。
- “超出范围最大值”以上的热电偶值报告为 32767。
- 所有范围的内部冷端误差均为 ±1.5 °C。该误差已包括到本表的误差中。模块需要至少 30 分钟的预热时间才能满足该规范。
- 仅针对 4 通道 SM TC：若是暴露在 970 MHz 到 990 MHz 的无线电辐射频率下，精度可能会有所下降。

说明

热电偶通道

热电偶信号模块上的各个通道可组态为不同的热电偶类型（可在组态模块期间进行选择）。

A.8.6 热电偶滤波器选型和更新时间

测量热电偶时建议使用 100 ms 的积分时间。
使用更小的积分时间将增大温度读数的重复性误差。

表格 A- 56 热电偶滤波器选型和更新时间

抑制频率 (Hz)	积分时间 (ms)	更新时间 (秒)		
		单通道 SB	4 通道 SM	8 通道 SM
10	100	0.301	1.225	2.450
50	20	0.061	0.263	0.525
60	16.67	0.051	0.223	0.445
400 ¹	10	0.031	0.143	0.285

- 在选择 400 Hz 抑制频率时，要维持模块的分辨率和精度，积分时间应为 10 ms。该选择还可抑制 100 Hz 和 200 Hz 的噪声。

A.8.7 RTD 传感器类型选型表

表格 A- 57 RTD 模块支持的不同传感器的范围和精度

温度系数	RTD 类型	低于范围最小值 ¹	额定范围下限	额定范围上限	超出范围最大值 ²	25 °C 时的额定范围精度	-20 °C 到 60 °C 时的额定范围精度
Pt 0.003850 ITS90 DIN EN 60751	Pt 100 气候型	-145.00 °C	-120.00 °C	145.00 °C	155.00 °C	±0.20 °C	±0.40 °C
	Pt 10	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±1.0 °C	±2.0 °C
	Pt 50	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±0.5 °C	±1.0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0.003902 Pt 0.003916 Pt 0.003920	Pt 100	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±0.5 °C	±1.0 °C
	Pt 200	-243.0 °C	-200.0 °C	850.0 °C	1000.0 °C	±0.5 °C	±1.0 °C
	Pt 500						
	Pt 1000						
Pt 0.003910	Pt 10	-273.2 °C	-240.0 °C	1100.0 °C	1295 °C	±1.0 °C	±2.0 °C
	Pt 50	-273.2 °C	-240.0 °C	1100.0 °C	1295 °C	±0.8 °C	±1.6 °C
	Pt 100						
	Pt 500						
Ni 0.006720 Ni 0.006180	Ni 100	-105.0 °C	-60.0 °C	250.0 °C	295.0 °C	±0.5 °C	±1.0 °C
	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						

温度系数	RTD 类型	低于范围最小值 ¹	额定范围下限	额定范围上限	超出范围最大值 ²	25 °C 时的额定范围精度	-20 °C 到 60 °C 时的额定范围精度
LG-Ni 0.005000	LG-Ni 1000	-105.0 °C	-60.0 °C	250.0 °C	295.0 °C	±0.5 °C	±1.0 °C
Ni 0.006170	Ni 100	-105.0 °C	-60.0 °C	180.0 °C	212.4 °C	±0.5 °C	±1.0 °C
Cu 0.004270	Cu 10	-240.0 °C	-200.0 °C	260.0 °C	312.0 °C	±1.0 °C	±2.0 °C
Cu 0.004260	Cu 10	-60.0 °C	-50.0 °C	200.0 °C	240.0 °C	±1.0 °C	±2.0 °C
	Cu 50	-60.0 °C	-50.0 °C	200.0 °C	240.0 °C	±0.6 °C	±1.2 °C
	Cu 100						
Cu 0.004280	Cu 10	-240.0 °C	-200.0 °C	200.0 °C	240.0 °C	±1.0 °C	±2.0 °C
	Cu 50	-240.0 °C	-200.0 °C	200.0 °C	240.0 °C	±0.7 °C	±1.4 °C
	Cu 100						

1 “低于范围最小值”以下的 RTD 值报告为 -32768。

2 “超出范围最大值”以上的 RTD 值报告为 +32767。

表格 A- 58 电阻

范围	低于范围最小值	额定范围下限	额定范围上限	超出范围最大值 ¹	25 °C 时的额定范围精度	-20 °C 到 60 °C 时的额定范围精度
150 Ω	不适用	0 (0 Ω)	27648 (150 Ω)	176.383 Ω	±0.05%	±0.1%
300 Ω	不适用	0 (0 Ω)	27648 (300 Ω)	352.767 Ω	±0.05%	±0.1%
600 Ω	不适用	0 (0 Ω)	27648 (600 Ω)	705.534 Ω	±0.05%	±0.1%

1 “超出范围最大值”以上的电阻值报告为 32767。

A.8.8 RTD 滤波器选型和更新时间

表格 A- 59 滤波器选型和更新时间

噪声抑制频率 (Hz)	积分时间 (ms)	更新时间 (秒)		
		单通道 SB	4 通道 SM	8 通道 SM
10	100	4/2 线制: 0.301	4/2 线制: 1.222	4/2 线制: 2.445
		3 线制: 0.601	3 线制: 2.445	3 线制: 4.845
50	20	4/2 线制: 0.061	4/2 线制: 0.262	4/2 线制: 0.525
		3 线制: 0.121	3 线制: .505	3 线制: 1.015
60	16.67	4/2 线制: 0.051	4/2 线制: 0.222	4/2 线制: 0.445
		3 线制: 0.101	3 线制: 0.424	3 线制: 0.845
400 ¹	10	4/2 线制: 0.031	4/2 线制: 0.142	4/2 线制: 0.285
		3 线制: 0.061	3 线制: 0.264	3 线制: 0.525

¹ 在选择 400 Hz 滤波器时，要维持模块的分辨率和精度，积分时间应为 10 ms。该滤波器还可抑制 100 Hz 和 200 Hz 的噪声。

说明

对于没有连接传感器的激活通道，模块会报告 32767。如果还启用了开路检测，模块会使相应的红色 LED 闪烁。
 若使用 4 线制连接，对于 10 Ω RTD 范围，将得到最高精度。
 2 线模式的连接线电阻会导致传感器读数误差，因此无法保证精度。

A.9 通信接口

有关 S7-1200 可用模块的更完整列表，请参见“S7-1200

可编程控制器系统手册”或客户支持网站 (<http://www.siemens.com/tiaportal>)。

A.9.1 PROFIBUS 主站/从站

A.9.1.1 CM 1242-5 PROFIBUS DP 从站

表格 A- 60 CM 1242-5 的技术数据

技术数据	
产品编号	6GK7 242-5DX30-0XE0
接口	
与 PROFIBUS 的连接	9 针 D 型母连接器
连接网络组件（例如，光纤网络组件）时 PROFIBUS 接口上的最大电流消耗	5 V 时为 15 mA（仅限总线终端）*
允许的环境条件	
环境温度	
<ul style="list-style-type: none"> • 存储期间 • 运输期间 • 垂直安装（DIN 导轨水平）时的运行 • 水平安装（DIN 导轨垂直）时的运行 	<ul style="list-style-type: none"> • -40 °C 到 70 °C • -40 °C 到 70 °C • 0 °C 到 55 °C • 0 °C 到 45 °C
在 25 °C 下运行期间的最大相对湿度，无结露	95 %
防护等级	IP20
电源、电流消耗和功耗	
电源类型	DC
背板总线的电源	5 V
电流消耗（典型）	150 mA
有效功耗（典型值）	0.75 W

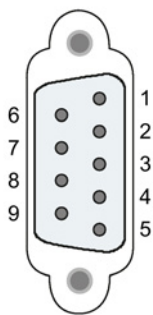
A.9 通信接口

技术数据	
电气隔离 • PROFIBUS 接口到地 • PROFIBUS 接口到内部电路	710 VDC, 持续 1 分钟
尺寸和重量	
• 宽度 • 高度 • 厚度	• 30 mm • 100 mm • 75 mm
重量 • 净重 • 含包装重量	• 115 g • 152 g

*) VP (引脚 6) 和 DGND (引脚 5) 之间连接的外部耗电装置的电流负载不得超过总线终端 15 mA 的最大电流 (防短路)。

A.9.1.2 CM 1242-5 的 D 型插座的引脚分配

PROFIBUS 接口



表格 A- 61 D 型插座的引脚分配

引脚	描述	引脚	描述
1	- 未使用 -	6	P5V2: +5V 电源
2	- 未使用 -	7	- 未使用 -
3	RxD/TxD-P: 数据线 B	8	RxD/TxD-N: 数据线 A

引脚	描述	引脚	描述
4	RTS	9	- 未使用 -
5	M5V2: 数据参考电位 (接地 DGND)	外壳	接地线

A.9.1.3 CM 1243-5 PROFIBUS DP 主站

表格 A- 62 CM 1243-5 的技术数据

技术数据	
产品编号	6GK7 243-5DX30-0XE0
接口	
与 PROFIBUS 的连接	9 针 D 型母连接器
连接网络组件 (例如, 光纤网络组件) 时 PROFIBUS 接口上的最大电流消耗	5 V 时为 15 mA (仅限总线终端) *
允许的环境条件	
环境温度	<ul style="list-style-type: none"> • 存储期间 • 运输期间 • 垂直安装 (DIN 导轨水平) 时的运行 • 水平安装 (DIN 导轨垂直) 时的运行
	<ul style="list-style-type: none"> • -40 °C 到 70 °C • -40 °C 到 70 °C • 0 °C 到 55 °C • 0 °C 到 45 °C
在 25 °C 下运行期间的最大相对湿度, 无结露	95 %
防护等级	IP20
电源、电流消耗和功耗	
电源类型	DC
电源/外部	24 V
<ul style="list-style-type: none"> • 最小值 • 最大值 	<ul style="list-style-type: none"> • 19.2 V • 28.8 V
电流消耗 (典型)	
<ul style="list-style-type: none"> • 通过 24 V DC • 通过 S7-1200 背板总线 	<ul style="list-style-type: none"> • 100 mA • 0 mA

技术数据	
有效功耗（典型值） <ul style="list-style-type: none"> 通过 24 V DC 通过 S7-1200 背板总线 	<ul style="list-style-type: none"> 2.4 W 0 W
电源 24 VDC/外部 <ul style="list-style-type: none"> 最小电缆横截面积 最大电缆横截面积 螺钉型端子的紧固扭矩 	<ul style="list-style-type: none"> 最小值： 0.14 mm² (AWG 25) 最大值： 1.5 mm² (AWG 15) 0.45 Nm (4 lb-in)
电气隔离 <ul style="list-style-type: none"> PROFIBUS 接口到地 PROFIBUS 接口到内部电路 	710 VDC, 持续 1 分钟
尺寸和重量	
<ul style="list-style-type: none"> 宽度 高度 厚度 	<ul style="list-style-type: none"> 30 mm 100 mm 75 mm
重量 <ul style="list-style-type: none"> 净重 含包装重量 	<ul style="list-style-type: none"> 134 g 171 g

*) VP（引脚 6）和 DGND（引脚 5）之间连接的外部耗电装置的电流负载不得超过总线终端 15 mA 的最大电流（防短路）。

说明

CM 1243-（PROFIBUS 主站模块）必须由 CPU 的 24 VDC 传感器电源供电。

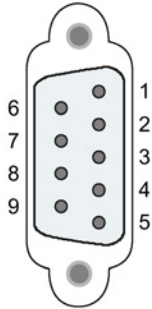
A.9.1.4 PROFIBUS 主站 (CM 1243-5) 需要 CPU 的 24 VDC 电源

说明

CM 1243-5（PROFIBUS 主站模块）必须由 CPU 的 24 VDC 传感器电源供电。

A.9.1.5 CM 1243-5 的 D 型插座的引脚分配

PROFIBUS 接口



表格 A- 63 D 型插座的引脚分配

引脚	描述	引脚	描述
1	- 未使用 -	6	VP: +5 V 电源, 仅适用于总线终端电阻, 不适合为外部设备供电
2	- 未使用 -	7	- 未使用 -
3	RxD/TxD-P: 数据线 B	8	RxD/TxD-N: 数据线 A
4	CNTR-P: RTS	9	- 未使用 -
5	DGND: 数据信号和 VP 的地	外壳	接地线

PROFIBUS 电缆

说明

接触 PROFIBUS 电缆的屏蔽层

必须接触 PROFIBUS 电缆的屏蔽层。

为此, 从 PROFIBUS 电缆末端剥去绝缘层, 然后将屏蔽层连接到功能地。

A.9.2 GPRS CP

说明

CP 1242-7 未经海事应用认证

CP 1242-7 未经海事认证。

说明

要使用这些模块，CPU 固件必须为 V2.0 或更高版本。

A.9.2.1 CP 1242-7 GPRS

表格 A-64 CP 1242-7 GPRS V2 的技术数据

技术数据	
产品编号	6GK7 242-7KX3-0XE0
无线接口	
天线连接器	SMA 插座
额定阻抗	50 欧姆
无线连接	
最大发射功率	<ul style="list-style-type: none"> • GSM 850, 类别 4: +33 dBm ±2 dBm • GSM 900, 类别 4: +33 dBm ±2 dBm • GSM 1800, 类别 1: +30 dBm ±2 dBm • GSM 1900, 类别 1: +30 dBm ±2 dBm
GPRS	多槽类别 10 B 类设备 编码方式 1...4 (GMSK)
SMS	发出模式: MO 服务: 点对点
允许的环境条件	

技术数据	
环境温度 <ul style="list-style-type: none"> • 存储期间 • 运输期间 • 垂直安装（DIN 导轨水平）时的运行 • 水平安装（DIN 导轨垂直）时的运行 	<ul style="list-style-type: none"> • -40 °C 到 70 °C • -40 °C 到 70 °C • 0 °C 到 55 °C • 0 °C 到 45 °C
在 25 °C 下运行期间的最大相对湿度，无结露	95 %
防护等级	IP20
电源、电流消耗和功耗	
电源类型	DC
电源/外部 <ul style="list-style-type: none"> • 最小值 • 最大值 	24 V <ul style="list-style-type: none"> • 19.2 V • 28.8 V
电流消耗（典型） <ul style="list-style-type: none"> • 通过 24 V DC • 通过 S7-1200 背板总线 	<ul style="list-style-type: none"> • 100 mA • 0 mA
有效功耗（典型值） <ul style="list-style-type: none"> • 通过 24 V DC • 通过 S7-1200 背板总线 	<ul style="list-style-type: none"> • 2.4 W • 0 W
24 V 直流电源 <ul style="list-style-type: none"> • 最小电缆横截面积 • 最大电缆横截面积 • 螺钉型端子的紧固扭矩 	<ul style="list-style-type: none"> • 最小值： 0.14 mm² (AWG 25) • 最大值： 1.5 mm² (AWG 15) • 0.45 Nm (4 lb-in)
电气隔离 电源单元到内部电路	710 VDC，持续 1 分钟
尺寸和重量	
<ul style="list-style-type: none"> • 宽度 • 高度 • 厚度 	<ul style="list-style-type: none"> • 30 mm • 100 mm • 75 mm

技术数据	
重量	
<ul style="list-style-type: none"> • 净重 • 含包装重量 	<ul style="list-style-type: none"> • 133 g • 170 g

A.9.2.2 GSM/GPRS 天线 ANT794-4MR

ANT794-4MR GSM/GPRS 天线的技术数据

ANT794-4MR	
产品编号	6NH9860-1AA00
移动无线网络	GSM/GPRS
频率范围	<ul style="list-style-type: none"> • 824 到 960 MHz (GSM 850、900) • 1710 到 1880 MHz (GSM 1800) • 1900 到 2200 MHz (GSM/UMTS)
特性	全向
天线增益	0 dB
阻抗	50 欧姆
驻波比 (SWR)	< 2,0
最大功率	20 W
极性	线性垂直
连接器	SMA
天线电缆长度	5 m
外部材料	硬 PVC, 抗 UV
防护等级	IP20
允许的环境条件	<ul style="list-style-type: none"> • 工作温度 • 运输/存储温度 • 相对湿度
	<ul style="list-style-type: none"> • -40 °C 到 +70 °C • -40 °C 到 +70 °C • 100 %
外部材料	硬 PVC, 抗 UV
结构	天线带 5 m 固定电缆和 SMA 公连接器

ANT794-4MR	
尺寸 (D x H) (mm)	25 x 193
重量 <ul style="list-style-type: none"> • 天线 (包括电缆) • 配件 	<ul style="list-style-type: none"> • 310 g • 54 g
安装	使用随附支架

A.9.2.3 平头天线 ANT794-3M

平头天线 ANT794-3M 的技术数据

ANT794-3M		
产品编号	6NH9870-1AA00	
移动无线网络	GSM 900	GSM 1800/1900
频率范围	890 - 960 MHz	1710 - 1990 MHz
驻波比 (VSWR)	≤ 2:1	≤ 1,5:1
回波损耗 (Tx)	≈ 10 dB	≈ 14 dB
天线增益	0 dB	
阻抗	50 欧姆	
最大功率	10 W	
天线电缆	带 SMA 公连接器的 HF 电缆 RG 174 (固定)	
电缆长度	1.2 m	
防护等级	IP64	
允许的温度范围	-40°C 到 +75°C	
易燃性	UL 94 V2	
外部材料	ABS Polylac PA-765, 浅灰色 (RAL 7035)	
尺寸 (W x L x H, 以 mm 为单位)	70.5 x 146.5 x 20.5	
重量	130 g	

A.9 通信接口

A.9.3 远程服务 (TS)

以下手册包含有关 TS Adapter IE Basic 和 TS 适配器模块的技术规范：

- 《工业软件工程工具》
模块化 TS 适配器
- 《工业软件工程工具》
TS Adapter IE Basic

有关该产品以及产品文档的详细信息，请参见 TS 适配器产品目录网站

(<https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=TS%20Adapter%20IE%20basic&tab=>)。

A.9.4 RS485、RS232 和 RS422 通信

A.9.4.1 CB 1241 RS485 规范

说明

要使用此 CB，CPU 固件必须为 V2.0 或更高版本。

表格 A- 65 常规规范

技术数据	CB 1241 RS485
订货号	6ES7 241-1CH30-1XB0
尺寸 W x H x D (mm)	38 x 62 x 21
重量	40 g

表格 A- 66 发送器和接收器

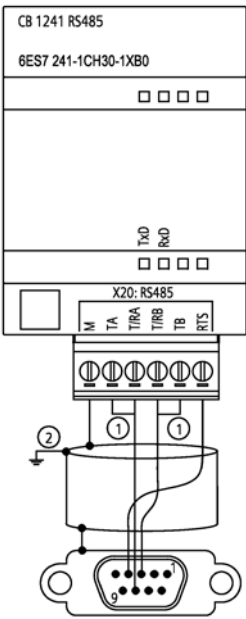
技术数据	CB 1241 RS485
类型	RS485 (2 线制半双工)
共模电压范围	-7 V 到 +12 V, 1 秒, 3 VRMS 连续
发送器差动输出电压	$R_L = 100 \Omega$ 时最小 2 V, $R_L = 54 \Omega$ 时最小 1.5 V

技术数据	CB 1241 RS485
端接和偏置	B 上 10K 对 +5 V, RS485 引脚 3 A 上 10K 对 GND, RS485 引脚 4
可选终端	短针 TB 对针 T/RB, 有效终端阻抗为 127 Ω , 连接至 RS485 针 3 短针 TA 对针 T/RA, 有效终端阻抗为 127 Ω , 连接至 RS485 针 4
接收器输入阻抗	最小 5.4K Ω , 包括终端
接收器阈值/灵敏度	最低 +/- 0.2 V, 典型滞后 60 mV
隔离 RS485 信号与机壳接地 RS485 信号与 CPU 逻辑公共端	500 VAC, 1 分钟
电缆长度, 屏蔽	最长 1000 m
波特率	300 波特, 600 波特, 1.2 Kb, 2.4 Kb, 4.8 Kb, 9.6 Kb (默认值), 19.2 Kb, 38.4 Kb, 57.6 Kb, 76.8 Kb, 115.2 Kb,
奇偶校验	无奇偶校验 (默认), 偶数, 奇数, 传号 (奇偶校验位始终设为 1), 空号 (奇偶校验位始终设为 0)
停止位的数目	1 (默认值), 2
流控制	不支持
等待时间	0 到 65535 ms

表格 A- 67 电源

技术数据	CB 1241 RS485
功率损失 (损耗)	1.5 W
最大电流消耗 (SM 总线)	50 mA
最大电流消耗 (24 VDC)	80 mA

A.9 通信接口

<p>CB 1241 RS485 (6ES7 241-1CH30-1XB0)</p> 	
<p>① 如图所示连接“TA”和“TB”以终止网络。（仅端接 RS485 网络上的终端设备。）</p>	
<p>② 使用屏蔽双绞线电缆，并将电缆屏蔽接地。</p>	

只能端接 RS485 网络的两端。不会端接或偏置这两个终端设备之间的设备。
 请参见“偏置和端接 RS485 网络连接”主题

表格 A- 68 CB 1241 RS485 (6ES7 241-1CH30-1XB0) 的连接器针脚位置

引脚	9 针连接器	X20
1	RS485/逻辑接地	--
2	RS485/未使用	--
3	RS485/TxD+	3 - T/RB
4	RS485/RTS	1 - RTS
5	RS485/逻辑接地	--
6	RS485/5 V 电源	--
7	RS485/未使用	--
8	RS485/TxD-	4 - T/RA

引脚	9 针连接器	X20
9	RS485/未使用	--
Shell		7 - M

A.9.4.2 CM 1241 RS422/485 技术规范

CM 1241 RS422/485 技术数据

表格 A- 69 常规规范

技术数据	CM 1241 RS422/485
产品编号	6ES7 241-1CH32-0XB0
尺寸 W x H x H (mm)	30 x 100 x 75
重量	155 g

表格 A- 70 发送器和接收器

技术数据	CM 1241 RS422/485
类型	RS422 或 RS485, 9 针 D 型插孔式连接器
共模电压范围	-7 V 到 +12 V, 1 秒, 3 VRMS 连续
发送器差动输出电压	$R_L = 100 \Omega$ 时最小 2 V, $R_L = 54 \Omega$ 时最小 1.5 V
端接和偏置	B 上 10K Ω 对 +5 V, PROFIBUS 引脚 3 A 上 10K Ω 对 GND, PROFIBUS 引脚 8 提供内部偏置选项, 或无内部偏置。 在所有情况下, 都需要外部终端, 请参见“偏置和端接 RS485 网络连接器”与“S7-1200 可编程控制器系统手册中的 RS422 和 RS485 组态”
接收器输入阻抗	最小 5.4K Ω , 包括终端
接收器阈值/灵敏度	最低 +/- 0.2 V, 典型滞后 60 mV

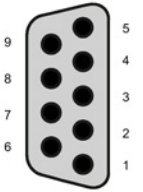
A.9 通信接口

技术数据	CM 1241 RS422/485
隔离 RS485 信号与机壳接地 RS485 信号与 CPU 逻辑公共端	500 VAC, 1 分钟
电缆长度, 屏蔽	最长 1000 m (取决于波特率)
波特率	300 波特, 600 波特, 1.2 Kb, 2.4 Kb, 4.8 Kb, 9.6 Kb (默认值), 19.2 Kb, 38.4 Kb, 57.6 Kb, 76.8 Kb, 115.2 Kb,
奇偶校验	无奇偶校验 (默认), 偶数, 奇数, 传号 (奇偶校验位始终设为 1), 空号 (奇偶校验位始终设为 0)
停止位的数目	1 (默认值), 2
流控制	对于 RS422 模式, 支持 XON/XOFF
等待时间	0 到 65535 ms

表格 A-71 电源

技术数据	CM 1241 RS422/485
功率损失 (损耗)	1.1 W
+5 VDC 电流	220 mA

表格 A-72 RS485 或 RS422 连接器（插孔式）

引脚	说明	连接器 (插孔式)	引脚	说明
1	逻辑接地或通信接地		6 PWR	+5 V 与 100 Ω 串联电阻：输出
2 TxD+	用于连接 RS422 不适用于 RS485：输出		7	未连接
3 TxD+	信号 B (RxD/TxD+)： 输入/输出		8 TXD-	信号 A (RxD/TxD-)：输入/输出
4 RTS ²	请求发送（TTL 电平）输出		9 TXD-	用于连接 RS422 不适用于 RS485：输出
5 GND	逻辑接地或通信接地		SHELL	机壳接地

1 引脚 2 和 9 仅适用于 RS422 的传送信号。

2 RTS 是 TTL 电平信号，可用于控制基于该信号进行工作的其它半双工设备。
该信号会在发送时激活，在所有其它时刻都不激活。

A.9.4.3 CM 1241 RS232 规范

表格 A-73 常规规范

技术数据	CM 1241 RS232
订货号	6ES7 241-1AH32-0XB0
尺寸 (mm)	30 x 100 x 75
重量	150 g

表格 A-74 发送器和接收器

技术数据	CM 1241 RS232
类型	RS232（全双工）
发送器输出电压	$R_L = 3K\ \Omega$ 时最小 +/- 5 V
传送输出电压	最大 +/- 15 VDC
接收器输入阻抗	最小 3 K Ω
接收器阈值/灵敏度	最低 0.8 V，最高 2.4 V 典型滞后 0.5 V

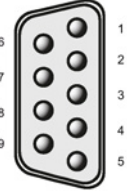
A.9 通信接口

技术数据	CM 1241 RS232
接收器输入电压	最大 +/- 30 VDC
隔离 RS 232 信号与机壳接地 RS 232 信号与 CPU 逻辑公共端	500 VAC, 1 分钟
电缆长度, 屏蔽	最长 10 m
波特率	300 波特, 600 波特, 1.2 Kb, 2.4 Kb, 4.8 Kb, 9.6 Kb (默认值), 19.2 Kb, 38.4 Kb, 57.6 Kb, 76.8 Kb, 115.2 Kb,
奇偶校验	无奇偶校验 (默认), 偶数, 奇数, 传号 (奇偶校验位始终设为 1), 空号 (奇偶校验位始终设为 0)
停止位的数目	1 (默认值), 2
流控制	硬件, 软件
等待时间	0 到 65535 ms

表格 A- 75 电源

技术数据	CM 1241 RS232
功率损失 (损耗)	1 W
+5 VDC 电流	200 mA

表格 A- 76 RS232 连接器 (公)

引脚	说明	连接器 (插头式)	引脚	说明
1 DCD	数据载波检测: 输入		6 DSR	数据设备就绪: 输入
2 RxD	从 DCE 接收数据: 输入		7 RTS	请求发送: 输出
3 TxD	传送数据到 DCE: 输出		8 CTS	允许发送: 输入
4 DTR	数据终端就绪: 输出		9 RI	振铃指示器 (未用)
5 GND	逻辑地		SHELL	机壳接地

A.10 工艺模块

A.10.1 SM 1278 4xIO-Link 主站 SM

A.10.1.1 SM 1278 4xIO-Link 主站信号模块规范

表格 A- 77 常规规范

技术数据		SM 1278 4xIO-Link 主站信号模块
产品编号		6ES7 278-4BD32-0XB0
尺寸 W x H x D (mm)		45 x 100 x 75
重量		150 g
常规信息		
	I&M 数据	√; IM0 到 IM3
供电电压		
	额定电压 (直流)	24 VDC
	直流电压下限	19.2 V; 如果使用 IO-Link, 则为 20.5 V (主站中 IO-Link 设备的供电电压必须至少为 20 V)
	直流电压上限	28.8 VDC
	反极性保护	√
输入电流		
	电流消耗	65 mA; 无负载
编码器电源		
	输出点数	4
	输出电流, 额定值	200 mA
功率损耗		
	典型功耗	1 W, 不包括端口加载
数字量输入/输出		
	电缆长度 (米)	最大 20 m (非屏蔽)
SDLC		
	电缆长度 (米)	最大 20 m (非屏蔽)

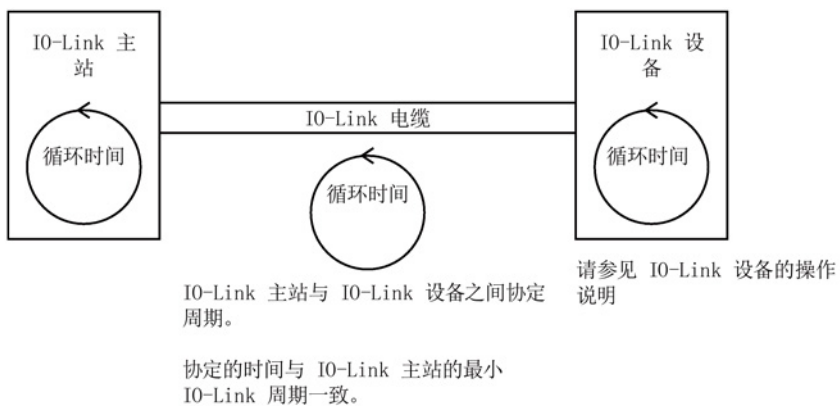
A.10 工艺模块

技术数据		SM 1278 4xIO-Link 主站信号模块
IO-Link		
	端口数	4
	可同时控制的端口数	4
	IO-Link 协议 1.0	√
	IO-Link 协议 1.1	√
工作模式		
	IO-Link	√
	DI	√
	DQ	√; 最大 100 mA
IO-Link 设备连接		
	端口类型 A	√
	传输率	4.8 kBd (COM1)
		38.4 kBd (COM2)
		230.4 kBd (COM3)
	最短周期时间	2 ms, 动态, 取决于用户数据长度
	过程数据大小, 每个端口的输入量	最大 32 个字节
	过程数据大小, 每个模块的输入量	32 个字节
	过程数据大小, 每个端口的输出量	最大 32 个字节
	过程数据大小, 每个模块的输出量	32 个字节
	设备参数的存储器大小	2 KB
	非屏蔽电缆的最大长度 (米)	20 m
中断/诊断/状态信息		
	状态显示	√
中断		
	诊断中断	√; 端口诊断仅适用于 IO-Link 模式
诊断报警		
	诊断	
	电源电压监视	√
	短路	√

技术数据		SM 1278 4xIO-Link 主站信号模块
诊断指示器 LED		
	电源电压监视	√; 红色闪烁 DIAG LED
	通道状态显示	√; 每个通道有一个绿色 LED, 用于显示通道状态 Qn (SIO 模式) 和端口状态 Cn (IO-Link 模式)
	通道诊断	√; 红色 Fn LED
	模块诊断	有; 绿色/红色 DIAG LED
电气隔离		
	电气隔离通道	
	通道之间	×
	通道和背板总线之间	√
允许的电位差		
	不同电路之间	75 VDC/60 VAC (基本绝缘)
绝缘		
	绝缘测试	707 VDC (型式测试)
环境条件		
	工作温度	
	最小值	-20 °C
	最大值	60 °C
	水平安装时的最低温度	-20 °C
	水平安装时的最高温度	60 °C
	垂直安装时的最低温度	-20 °C
	垂直安装时的最高温度	50 °C

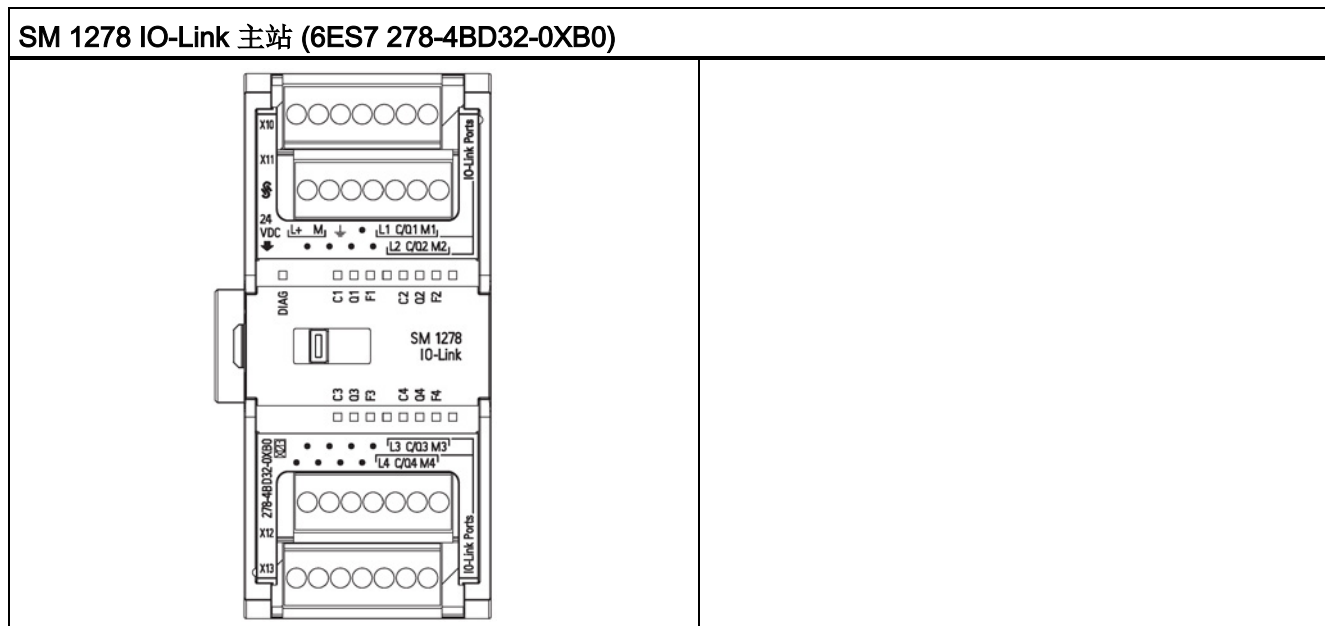
A.10 工艺模块

响应时间总览



A.10.1.2 SM 1278 4xIO-Link 主站 SM 接线图

表格 A-78 SM 1278 IO-Link 主站接线图



表格 A- 79 SM 1278 IO-Link 主站 (6ES7 278-4BD32-0XB0) 的连接器针脚位置

针脚	X10	X11	X12	X13
1	L+/24 VDC	无连接	无连接	无连接
2	M/24 VDC	无连接	无连接	无连接
3	功能性接地	无连接	无连接	无连接
4	无连接	无连接	无连接	无连接
5	L ₁	L ₂	L ₃	L ₄
6	C/Q ₁	C/QL ₂	C/Q ₃	C/QL ₄
7	ML ₁	ML ₂	M ₃	ML ₄

A.11 随附产品

A.11.1 PM 1207 电源模块

PM 1207 是用于 SIMATIC S7-1200 的电源模块。它提供以下功能：

- 输入 120/230 VAC，输出 24 VDC/2.5A
- 订货号 6ESP 332-1SH71-4AA0

有关该产品以及产品文档的详细信息，请参见 PM 1207 的产品目录网站

(<https://eb.automation.siemens.com/mall/en/de/Catalog/Product/6AG1332-1SH71-4AA0>)。

A.11.2 CSM 1277 紧凑型交换机模块

CSM1277 是工业以太网紧凑型交换机模块。可将其用于增加 S7-1200 的以太网接口，以允许与操作员面板、编程设备或其它控制器同时进行通信。它可以提供以下功能：

- 连接至工业以太网的 4 x RJ45 插座
- 位于顶部的用于外部 24 VDC 电源连接的接线板上的 3 极插头
- 用于工业以太网端口的诊断和状态显示的 LED
- 订货号 6GK7 277-1AA00-0AA0

有关该产品以及产品文档的详细信息，请参见 CSM 1277 的产品目录网站

(<https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=csm%201277&tab=>)。

A.11.3 CM CANopen 模块

CM CANopen 是一种插入式模块，可以插入到 SIMATIC S7-1200 PLC 与任意运行 CANopen 的设备之间。可以将 CM CANopen 组态为主站或从站。有两种 CM CANopen modules: CANopen 模块（订货号 021620-B）和 CANopen (Ruggedized) 模块（订货号 021730-B）。

CANopen 模块提供以下功能：

- 每个 CPU 可以连接 3 个模块
- 可连接多达 16 个 CANopen 从节点
- 每个模块提供 256 字节输入和 256 字节输出
- 3 个 LED 分别提供模块、网络和 I/O 状态的诊断信息
- 支持将 CANopen 网络组态存储在 PLC 中
- 模块可集成到 TIA Portal 组态套件的硬件目录中
- 可通过内含的 CANopen Configuration Studio 或任何其它外部 CANopen 组态工具实现 CANopen 组态
- 遵循 CANopen 通信配置文件 CiA 301 修订版 4.2 和 CiA 302 修订版 4.1
- 支持透明 CAN 2.0A，用于自定义协议处理
- 预制功能块可用于 TIA Portal 中的所有 PLC 编程
- 包含 CM CANopen 模块；适用于子网络的带螺丝端子的 DSUB。CM CANopen Configuration Studio CD 和 USB 组态电缆

有关该产品以及产品文档的详细信息，请参见 CM CANopen 的产品目录网站。

用 V4.1 CPU 更换 V3.0 CPU

B.1 用 V4.1 CPU 更换 V3.0 CPU

您可以用 V4.1 CPU 替换 V3.0 CPU (页 83)，也可使使用为 V3.0 CPU 设计的现有 STEP 7 项目，但无法通过固件更新将 V3.0 CPU 升级到 V4.1 CPU；必须替换该硬件。用 V4.1 CPU 替换 V3.0 CPU 时，最好另外也对信号和通信模块进行固件更新 (页 377)检查并实际进行更新。

说明

无法将设备从 V4.1 替换为 V3.0

您可以用 V4.1 CPU 替换 V3.0 CPU，但下载组态后，无法用 V3.0 CPU 替换 V4.1 CPU。若要查看或使用现有的 STEP 7 V3.0 项目，在更换设备之前需先将您的 STEP 7 V3.0 项目归档。

请注意，如果您尚未下载替换之后的设备组态，您可以撤消替换。下载之后，便无法撤消从 V3.0 到 V4.1 的替换。

需要注意两个 CPU 版本之间的一些组态和运行区别：

组织块

使用 V4.1，您可以将 OB 执行组态为可中断或不可中断 (页 62)。对于之前的 V3.0 CPU 项目，STEP 7 将所有 OB 默认设置为不可中断，

STEP 7 将所有 OB 属性 (页 62)设置为 V3.0 CPU STEP 7 项目中的相应值。

随后可根据需要更改中断或优先级设置。

如果没有未决诊断事件，诊断错误中断 OB 启动信息将完全参考子模块。

CPU 密码保护

STEP 7 将 V4.1 CPU 的密码保护级别 (页 93) 设置为与 V3.0 CPU 相等的密码保护级别，并将 V3.0 密码指定为 V4.1 CPU 的“完全访问（无保护）”密码：

V3.0 保护级别	V4.1 访问级别
无保护	完全访问（无保护）
写保护	读访问
写/读保护	HMI 访问

请注意，对于 V3.0，不存在 V4.1 的“无访问（完全保护）”访问级别。

Web 服务器

如果在 V3.0 项目中使用用户定义 Web 页面，那么在升级项目前，需要将项目安装文件夹中的项目存储在子文件夹“UserFilesWebserver”下。如果将用户定义页面存储在这个位置，则保存 STEP 7 项目时还将保存用户定义 Web 页面。

如果将 V3.0 CPU 替换为 V4.1 CPU，用于激活 Web 服务器的 Web 服务器项目设置和 HTTPS 设置会与版本 V3.0 中的相同。然后用户可以按需组态用户、权限、密码 (页 275) 和语言来使用 Web 服务器。如果不赋予用户更多权限，您将只能查看标准 Web 页面 (页 276) 中的内容。S7-1200 V4.1 CPU 不支持之前的预组态“admin”用户和密码。

S7-1200 V3.0 Web 服务器数据日志页面提供“下载并清除”(Download and Clear) 操作。V4.1 Web 服务器不再提供这一功能，您可以从文件浏览器页面 (页 276) 访问数据日志。不过，该 Web 服务器提供了下载、重命名和删除数据日志文件的功能。

传送卡不兼容

无法使用 V3.0 传送卡 (页 66) 将 V3.0 程序传送到 V4.1 CPU。您必须在 STEP 7 中打开 V3.0 项目，将设备更换为 V4.1 CPU (页 83)，然后将 STEP 7 项目下载到您的 V4.1 CPU。将项目更改为 V4.1 项目后，即可创建一个 V4.1 传送卡以便执行后续的程序传送。

GET/PUT 通信

V3.0 默认启用 GET/PUT 通信。当用 V4.1 CPU 替换 V3.0 CPU (页 83) 时，您可以在兼容性信息部分看到一条消息，指示 GET/PUT 已启用。

运动控制支持

S7-1200 V4.1 CPU 不支持 V1.0 和 V2.0 运动控制库。如果为具有 V1.0 或 V2.0 运动控制库的 STEP 7 项目更换设备，则设备替换编译过程中会使用兼容的 V3.0 运动控制指令 (页 333) 替换 V1.0 或 V2.0 运动控制库指令。

如果为包含两种不同运动控制指令版本 (V3.0 和 V5.0) 的 STEP 7 项目更换设备，将 V3.0 CPU 更换为 V4.1 CPU，则设备更换编译过程中会使用兼容的 V5.0 运动控制指令 (页 333)。

将 V3.0 CPU 更换为 V4.1 CPU 的设备更换期间，运动控制工艺对象 (TO) 版本不会自动从 V3.0 更改为 V5.0。如果要升级到更高版本，必须转至指令树并为项目选择所需的 S7-1200 运动控制版本，如下表中所示：

CPU 版本	允许的运动控制版本
V4.1 (运动控制 V5.0)	V5.0、V4.0 或 V3.0
V4.0 (运动控制 V4.0)	V4.0 或 V3.0
V3.0 (运动控制 V3.0)	V3.0

运动控制版本 V3.0 和 V5.0 的 TO 结构不同。所有相关的块也会更改。块接口、监控表以及跟踪都将更新为新的运动控制 V5.0 结构。有关 V3.0 CPU 和 V4.1 CPU 运动控制轴参数的区别，请参见以下两个表格：

V3.0 CPU (运动控制 V3.0)	V4.1 CPU (运动控制 V5.0)
Config.General.LengthUnit	Units.LengthUnit
Config.Mechanics.PulsesPerDriveRevolution	Actor.DriveParameter.PulsesPerDriveRevolution
Config.Mechanics.LeadScrew	Mechanics.LeadScrew
Config.Mechanics.InverseDirection	Actor.InverseDirection
Config.DynamicLimits.MinVelocity	DynamicLimits.MinVelocity
Config.DynamicLimits.MaxVelocity	DynamicLimits.MaxVelocity
Config.DynamicDefaults.Acceleration	DynamicDefaults.Acceleration
Config.DynamicDefaults.Deceleration	DynamicDefaults.Deceleration
Config.DynamicDefaults.EmergencyDeceleration	DynamicDefaults.EmergencyDeceleration

V3.0 CPU (运动控制 V3.0)	V4.1 CPU (运动控制 V5.0)
Config.DynamicDefaults.Jerk	DynamicDefaults.Jerk
Config.PositionLimits_SW.Active	PositionLimitsSW.Active
Config.PositionLimits_SW.MinPosition	PositionLimitsSW.MinPosition
Config.PositionLimits_SW.MaxPosition	PositionLimitsSW.MaxPosition
Config.PositionLimits_HW.Active	PositionLimitsHW.Active
Config.PositionLimits_HW.MinSwitchedLevel	PositionLimitsHW.MinSwitchLevel
Config.PositionLimits_HW.MaxSwitchedLevel	PositionLimitsHW.MaxSwitchLevel
Config.Homing.AutoReversal	Homing.AutoReversal
Config.Homing.Direction	Homing.ApproachDirection
Config.Homing.SideActiveHoming	Sensor[1].ActiveHoming.SideInput
Config.Homing.SidePassiveHoming	Sensor[1].PassiveHoming.SideInput
Config.Homing.Offset	Sensor[1].ActiveHoming.HomePositionOffset
Config.Homing.FastVelocity	Homing.ApproachVelocity
Config.Homing.SlowVelocity	Homing.ReferencingVelocity
MotionStatus.Position	Position
MotionStatus.Velocity	Velocity
MotionStatus.Distance	StatusPositioning.Distance
MotionStatus.TargetPosition	StatusPositioning.TargetPosition
StatusBits.SpeedCommand	StatusBits.VelocityCommand
StatusBits.Homing	StatusBits.HomingCommand

唯一重新命名的“CommandTable”参数是具有以下命令的数组：

V3.0	V4.1
Config.Command[]	Command[]

注：“Command[]”数组在 V3.0 中是“TO_CmdTab_Config_Command”类型的 UDT，在 V4.1 中则是“TO_Struct_Command”类型的 UDT。

指令变化

以下指令在参数或特性方面发生了变化：

- RDREC 和 WRREC (页 155)
- CONV (页 116)

HMI 面板通信

如果 S7-1200 V3.0 CPU 上连接一个或多个 HMI 面板 (页 22)，则与 S7-1200 V4.1 CPU 的通信将取决于使用的通信类型以及 HMI 面板的固件版本：

重新编译项目，然后将其下载到 CPU 和 HMI，并/或更新 HMI 固件。

重新编译程序块的相关要求

用 V4.1 CPU 替换 V3.0 CPU 后，您必须重新编译所有程序块，之后才能将其下载到 V4.1 CPU。此外，如果任意块采用专有技术保护 (页 95)或与某个 PLC 序列号绑定的复制保护 (页 96)，您必须先取消保护，然后再编译和下载块。

(不过，不需要取消激活与某个存储卡绑定的复制保护。)

成功编译后，可以重新组态专有技术保护和/或 PLC 序列号防拷贝保护。

请注意，如果您项目中的任意块采用了

OEM (原始设备制造商) 所提供的专有技术保护，您必须联系 OEM，获取这些块的 V4.1 版本。

更换设备后，Siemens 通常建议您先在 STEP 7

中重新编译硬件组态和软件，然后再将其下载到项目中的所有设备。

纠正编译项目时发现的所有错误，并进行重新编译，直到没有任何错误为止。

然后，可以将项目下载至 V4.1 CPU。

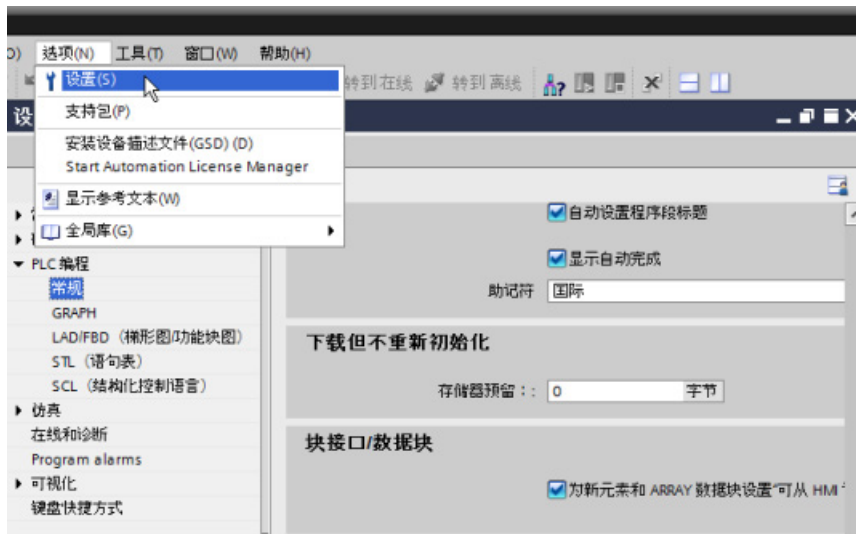
S7-1200 V3.0 项目可能不适用于 S7-1200 V4.1 CPU

S7-1200 V4.0 在每个 DB 中均增加了一个 100 字节的预留区域，以支持“下载而不重新初始化”。

在将 V3.0 项目下载至 V4.1CPU 之前，可以从 DB 中删除 100 字节的预留区域。

若要删除 100 字节的预留区域，在执行设备更换之前，请执行以下步骤：

1. 在 TIA Portal 主菜单中，选择“选项 > 设置”(Options > Settings) 菜单命令。
2. 从导航树中打开“PLC 编程 > 常规”(PLC programming > General) 节点。
3. 在“下载而不重新初始化”(Download without reinitialization) 区域，将存储器预留区域设置为 0 字节。



如果已执行设备更换，则必须分别从各个块中删除 100 字节的预留区域：

1. 在项目树中，右键单击“程序块”(Program blocks)文件夹中的数据块，并在快捷菜单中选择“属性”(Properties)。
2. 在“数据块属性”(Data block properties)对话框中，选择“下载而不重新初始化”(Download without reinitialization) 节点。
3. 将存储器预留区域设置为 0 字节。
4. 对项目中的每个数据块，均重复以上步骤。



有关 V4.1 特性的完整详细信息，请参见“S7-1200 可编程控制器系统手册”。

用 V4.1 CPU 更换 V3.0 CPU

B.1 用 V4.1 CPU 更换 V3.0 CPU

索引

A

AS-i

- AS-i 主站 CM 1243-2, 175
- 地址, 177
- 添加 AS-i 从站, 176
- 添加 AS-i 主站 CM1243-2 模块, 176

AT 变量覆盖, 74

ATEX 认证, 394

C

CALCULATE (计算), 118

用于复杂公式, 50

标定模拟值, 51

CANopen 模块

021620-B, 021630-B, 475

CB 1241 RS485, 463

CE 认证, 392

CEIL (上取整), 117

CONV (转换), 116

CPU

AS-i 地址, 177

CPU 1211C, 404

CPU 1212C, 404

CPU 1214C, 404

CPU 1215C, 404

CPU 1217C, 404

HSC 组态, 147

IP 地址, 91

PROFIBUS 地址, 173

PROFINET, 91

RUN/STOP 按钮, 37

工作模式, 59

专有技术保护, 95

比较并同步块, 374

比较表, 16

从在线 CPU 复制块, 373

以太网端口, 91

未指定的 CPU, 80, 379

处理 OB, 101

发热区, 24, 27

在线, 375

在线监视, 366

网络连接, 150

安全等级, 93, 93

设备组态, 79

访问保护, 93

启动过程, 86

启动参数, 86

诊断缓冲区, 375

转到在线, 363

组态与 HMI 的通信, 149

组态参数, 86, 90

重设 DB 的起始值, 372

复位为出厂设置, 376

信号板 (SB), 20

捕获 DB 的值, 372

监视表格, 367

调用块, 108

通信连接数, 154

通信板 (CB), 20

通信类型, 151

添加新设备, 82

添加模块, 84

密码保护, 93

- 程序执行, 57
- 强制, 369, 370
- 概述, 15
- 模拟量输入的阶跃响应时间, 437
- 操作员面板, 37, 59, 364

CPU 属性, 用户自定义的 Web 页面, 283

CSM 1277 紧凑型交换机模块, 474

C-Tick 认证, 394

CTRL_PWM 指令, 127

cULus 认证, 392

D

- DB (数据块), 105
- DC
 - 输出, 401
- DeviceStates, 135
- DTL 数据类型, 68

F

- FB (功能块), 103
- FBD (功能块图), 109
- FC (功能), 103
- FLOOR, 117
- FM 认证, 393

G

- GET, 180
- GET (从远程 CPU 读取数据)
 - 组态连接, 167
- GET_DIAG, 135
- Get_IM_Data, 135

H

HMI

- HMI 连接, 54
- 入门指南, 53, 55
- 网络连接, 54
- 画面, 55
- 组态 PROFINET 通信, 149

HMI 设备

- 网络连接, 150
- 概述, 22

HSC 时钟输入频率, 422

HSC (高速计数器)

- 组态, 147, 147
- 操作, 136, 137

HTML 页面, 用户自定义的, 280

- 页面位置, 283

HTML 页面, 用户定义的

- 开发, 281
- 刷新, 281

HTTP 连接, Web 服务器, 279

I

I 存储器

- 外围设备输入地址 (强制表格), 369
- 监视, 366
- 监视 LAD, 367
- 监控表, 366
- 强制, 369
- 强制表格, 369
- 强制操作, 370, 370

I/O

- 寻址, 72
- 监视 LAD 中的状态, 367
- 监视表格, 367
- 强制操作, 370

模拟量输入的电压表示法, 435
 模拟量输入的电流表示法, 436
 模拟量输入的阶跃响应时间, 437
 模拟量输出的电压表示法, 440
 模拟量输出的电流表示法, 440

IO-Link

上电, 385
 设备配置文件, 386
 技术总览, 385
 组件, 385
 组态, 386
 数据, 386

IO-Link 主站

方框图, 388
 连接示例, 389
 图示, 387

IO-Link 主站信号模块, 469

IP 地址, 91, 91

对在线 CPU 进行组态, 375

IP 路由器, 91

ISO on TCP

特殊模式, 158

ISO on TCP 协议, 157

ISO-on-TCP

连接 ID, 159
 连接组态, 166
 参数, 163

L

LAD (梯形图)

状态, 367, 369
 监视, 367
 监视状态或值, 366
 程序编辑器, 367
 概述, 108

LED 指示灯

LED 指令, 134
 通信接口, 196

LED (获取 LED 状态), 134

M

MAC 地址, 91

MC_ChangeDynamic (更改轴的动态设置), 356

MC_CommandTable, 353

MC_Halt (暂停轴), 342

MC_Home (使轴回原点), 339

MC_MoveAbsolute (绝对定位轴), 344

MC_MoveJog (在点动模式下移动轴), 351

MC_MoveRelative (相对定位轴), 346

MC_MoveVelocity (以预定义速度移动轴), 348

MC_Power (发布/阻止轴), 334

MC_ReadParam (读取工艺对象的参数), 361

MC_Reset (确认错误), 337

MC_WriteParam (写入工艺对象参数), 358

Modbus

版本, 38

ModuleStates, 135

MRES, 操作员面板, 37, 59, 364

My Documentation Manager, 4

N

NORM_X (标准化), 118

O

OB, (???)

OPC, 组态, 191

P

PID

PID_3Step 算法, 204

PID_3STEP (对阀门进行调节的 PID 控制器), 214

PID_Compact 算法, 204

PID_Compact (具有集成调节功能的通用 PID 控制器), 207

PID_Temp (允许处理温度控制的通用 PID 控制器), 224

调试, 260

概述, 203

PLC

CPU 概述, 15

HSC 组态, 147

入门指南, 43

专有技术保护, 95

比较并同步, 374

从在线 CPU 复制块, 373

使用块, 60, 99

变量, 44, 48

指令, 48

监视, 366

调用块, 108

添加模块, 84

强制, 369

强制操作, 370

PLC 变量

入门指南, 44, 48

PM 1207 电源模块, 474

PROFIBUS

CM 1242-5 (DP 从站) 模块, 169

CM 1243-5 (DP 主站) 模块, 169

GET, 180

PUT, 180

S7 连接, 181

从站, 169

主站, 169

地址, 173

地址, 组态, 173

网络连接, 150

通信连接数, 154

添加 CM 1243-5 (DP 主站) 模块, 172

添加 DP 从站, 172

PROFINET, 149

GET, 180

IP 地址, 91

PUT, 180

S7 连接, 181

网络连接, 150

连接 ID, 159

测试网络, 92

特殊模式, 158

通信连接数, 154

通信类型, 151

概述, 157

PROFINET RT, 157

PROFINET 接口

以太网地址属性, 91

PTO (脉冲串输出), 127

无法进行强制, 370

PtP 通信, 194

PUT, 180

PUT (将数据写入远程 CPU)

组态连接, 167

PWM

CTRL_PWM 指令, 127

PWM (脉冲宽度调制)

无法进行强制, 370

R

ROUND, 117
 RS232 和 RS485 通信模块, 196
 RTD 模块概述, 442
 RUN 模式, 59, 60
 工具栏按钮, 37
 程序执行, 57
 强制操作, 370
 操作员面板, 37, 59, 364
 RUN/STOP 按钮, 37

S

S7 通信
 组态连接, 167
 SCALE_X (标定), 118
 SCL (结构化控制语言)
 CEIL (上取整), 117
 CONV (转换), 116
 DeviceStates, 135
 FLOOR, 117
 GET_DIAG, 135
 Get_IM_Data, 135
 LED 状态, 134
 MC_ChangeDynamic (更改轴的动态设置), 356
 MC_CommandTable, 353
 MC_Halt (暂停轴), 342
 MC_Home (使轴回原点), 339
 MC_MoveAbsolute (绝对定位轴), 344
 MC_MoveJog (在点动模式下移动轴), 351
 MC_MoveRelative (相对定位轴), 346
 MC_MoveVelocity (以预定义速度移动轴), 348
 MC_Power (发布/阻止轴), 334
 MC_ReadParam (读取工艺对象的参数), 361
 MC_Reset (确认错误), 337
 MC_WriteParam (写入工艺对象参数), 358

ModuleStates, 135
 NORM_X (标准化), 118
 PID 概述, 203
 PID_3Step 算法, 204
 PID_3STEP (对阀门进行调节的 PID 控制器), 214
 PID_Compact 算法, 204
 PID_Compact (具有集成调节功能的通用 PID 控制器), 207
 PID_Temp (允许处理温度控制的通用 PID 控制器), 224
 SCALE_X (标定), 118
 Var 段, 111
 取整, 117
 程序编辑器, 111
 概述, 110
 截取, 117

SM 和 SB

比较表, 19

SMS, 189**STARTUP 模式**

程序执行, 57
 强制操作, 370

STEP 7

AS-i, 177
 HSC 组态, 147
 PROFIBUS, 173
 PROFINET, 91
 RUN/STOP 按钮, 37
 门户视图和项目视图, 33
 比较并同步, 374
 从在线 CPU 复制块, 373
 以太网端口, 91
 可扩展输入或输出, 36
 在编辑器之间拖放, 39
 网络连接, 150
 优先等级 (OB), 61

- 向 LAD 或 FBD 指令添加输入或输出, 35
 - 设备组态, 79
 - 收藏夹, 35
 - 更改设置, 37
 - 诊断缓冲区, 375
 - 拔出的模块, 42
 - 组态 CPU, 86, 90
 - 组态模块, 86, 90
 - 重设 DB 的起始值, 372
 - 捕获 DB 的值, 372
 - 监视, 366, 367
 - 添加新设备, 82
 - 添加模块, 84
 - 密码保护, 95
 - 插入指令, 34
 - 强制, 369
 - 强制操作, 370
 - 操作员面板, 37, 59, 364
- STEP 7 网页, 4
- STEP 7 编程
- 用户自定义的 Web 页面, 283
- STOP 模式, 59
- 工具栏按钮, 37
 - 强制操作, 370
 - 操作员面板, 37, 59, 364
- T**
- TCON
- 连接 ID, 159
 - 连接参数, 163
 - 组态, 166
- TCON_Param, 163
- TCP
- 协议, 157
 - 连接 ID, 159
- 连接组态, 166, 166
 - 参数, 163
 - 特殊模式, 158
- TCP/IP 通信, 149, 157
- TIA Portal
- PROFINET, 91
 - 设备组态, 79
 - 组态 CPU, 86, 90
 - 组态模块, 90
 - 添加新设备, 82
- TIA Portal, 门户视图和项目视图, 33
- TRCV
- 连接 ID, 159
- TRCV (通过以太网 (TCP) 接收数据)
- 特殊模式, 158
- TRCV_C
- 连接参数, 163
 - 特殊模式, 158
- TRCV_C 指令, 155
- TRCV_C (通过以太网 (TCP) 接收数据)
- 连接 ID, 159
- TRCV_C (通过以太网 (TCP) 接收数据)
- 组态, 166
- TRUNC (截取), 117
- TS 适配器, 19
- TSAP (传输服务访问点), 168
- 用于分配给设备的指令, 157
- TSEND
- 连接 ID, 159
- TSEND_C
- 连接参数, 163
- TSEND_C 指令, 155
- TSEND_C (通过以太网 (TCP) 发送数据)
- 连接 ID, 159
 - 组态, 166

TURCV

连接参数, 163

TURCV (通过以太网 (UDP) 接收数据)

组态, 166

TUSEND

参数, 163

TUSEND (通过以太网 (UDP) 发送数据)

组态, 166

U**UDP**

连接组态, 166

参数, 163

UDP 协议, 157

USS 协议库, 198

W

Web 页面, 用户自定义的, 280

Web 服务器, 275

HTTP 最大连接数, 279

限制, 279

Web 服务器最大连接数, 279

WWW, 283

Y

一致性检查, 133

R**入门指南**

CPU, 43

HMI, 53, 55

HMI 连接, 54

LAD 程序, 46, 49

PLC 变量, 44, 48

功能框指令, 49

代码块, 106

网络连接, 54

寻址, 48

拆分编辑器, 44, 48, 48

变量, 44, 48

项目, 43

指令, 48

程序块, 106

触点, 46

新 PLC, 43

数学运算指令, 49

G**工艺对象**

PID, 205

运动控制, 293

工艺模块, SM 1278 4xIO-Link 主站, 469

工业环境

认证, 395

工作存储器, 16, 66

工作模式, 37, 37, 59, 364

Y

与 CPU、存储卡或密码绑定, 96

S H**上传**

从在线 CPU 复制块, 373

用户程序, 373

发现, 379

M

- 门户视图, 33
 - PROFINET, 91
 - 组态 CPU, 86, 90
 - 组态以太网端口, 91
 - 组态模块, 86, 90
 - 添加新设备, 82

Z

- 子网掩码, 91

Z H

- 专有技术保护
 - 密码保护, 95
- 支持, 5

B

- 不可中断移动 (UMOVE_BLK) 指令, 115

Q

- 区别
 - 点对点指令, 195

B

- 比较并同步在线/离线 CPU, 374
- 比较表
 - CPU 型号, 16
 - HMI 设备, 22
 - 模块, 19
- 比较指令, 115

Q

- 切片 (变量化数据类型), 72

Z H

- 中断
 - 中断等待时间, 62
 - 概述, 61

S H

- 手册, 4

Q

- 气流, 27

Z H

- 长格式日期和时间数据类型, 68

P

- 片段 DB (用户自定义的 Web 页面)
 - 生成, 283

F

- 反向电压保护, 400

C

- 从在线 CPU 复制块, 373

W

- 文档, 4

J

计数器

- HSC 运行, 137
- HSC 组态, 147
- HSC (高速计数器), 136
- 大小, 19
- 数量, 18

计数器指令, 125

R

认证

- ATEX, 394
- CE, 392
- C-Tick, 394
- cULus, 392
- FM, 393
- 海事, 395
- 韩国认证, 394

Y

以太网

- CSM 1277 紧凑型交换机模块, 474
- GET, 180
- IP 地址, 91
- PUT, 180
- 网络连接, 150
- 连接 ID, 159
- 特殊模式, 158
- 通信, 149
- 通信连接数, 154
- 通信类型, 151
- 概述, 157

以太网协议, 157

- 多节点连接, 181

以太网指令

- TRCV_C, 155
- TSEND_C, 155

W

未指定的 CPU, 80, 379, 379

S H

示例, PID

- PID_3Step, 组态设置, 240
- PID_Compact, 组态设置, 239
- PID_Temp, 组态设置, 242

示例, 各种

- AT 变量覆盖, 74
- 在编辑器之间拖放, 39
- 轨迹和逻辑分析器功能, 383
- 变量化数据类型的切片, 73

示例, 运动控制

- CPU 1211C、CPU 1212C、CPU 1214C 和 CPU 1215C 脉冲输出速度组态, 289
- CPU 1217C 脉冲输出速度组态, 288
- MC 归位的速度特性曲线, 332
- 组态工艺对象运动命令表, 318

示例, 指令

- CALCULATE, 50

示例, 通信

- AS-i 从站寻址, 178
- PROFINET 通信协议, 157
- 使用公共发送和接收连接的 CPU 通信, 161
- 使用单独发送和接收连接的 CPU 通信, 160
- 通过 TSEND_C 或 TRCV_C 连接的 CPU 通信, 162
- 遥控, 189

- G**
- 功能 (FC), 103
 - 专有技术保护, 95
 - 功能块 (FB)
 - 专有技术保护, 95
 - 初始值, 103
 - 背景数据块, 103
 - 输出参数, 103
 - 功能框指令
 - 入门指南, 49
- B**
- 本地/伙伴连接, 165
- K**
- 可扩展指令, 36
 - 可视化, HMI 设备, 22
- B**
- 布尔值或位值, 71
- D**
- 电池板, BB 1297, 433
 - 电源模块
 - PM1207, 474
 - 电磁兼容性, 398
 - 电磁兼容性 (EMC), 396
- S H**
- 生成用户自定义的 Web 页面 DB, 283
- D**
- 代码块, 99
 - DB (数据块), 105
 - FB (功能块), 103
 - FC (功能), 103
 - OB 数目, 18
 - 与 CPU、存储卡或密码绑定, 96
 - 专有技术保护, 95
 - 中断, 18
 - 计数器 (数量和存储器要求), 18
 - 代码块数目, 18
 - 用户程序的大小, 18
 - 定时器 (数量和存储器要求), 18
 - 组织块 (OB), 18, 101
 - 复制保护, 96
 - 监视, 18
 - 调用块, 108
 - 嵌套深度, 18
- Y**
- 用 V4.1 CPU 更换 V3.0 CPU, 477
 - 用于显示使用情况的交叉引用, 133
 - 用户自定义的 Web 页面, 275, 280
 - 生成程序块, 283
 - 用 WWW 指令启用, 283
 - 在 STEP 7 中进行编程, 283
 - 组态, 283
 - 用户自定义的 Web 页面的控制 DB
 - WWW 指令的参数, 283
 - 用户定义的 Web 页面
 - 使用 HTML 编辑器创建, 281
 - 刷新, 281
 - 装载存储器限制, 282
 - 用户界面
 - STEP 7 项目和门户视图, 33

用户程序

- 与 CPU、存储卡或密码绑定, 96
- 从在线 CPU 复制块, 373
- 可扩展指令, 36
- 在编辑器之间拖放, 39
- 向 LAD 或 FBD 指令添加输入或输出, 35
- 收藏夹, 35
- 组织块 (OB), 101
- 密码保护, 95
- 插入指令, 34

Z H

- 主动/被动连接, 165
- 主动/被动通信
 - 连接 ID, 159
 - 参数, 163
 - 组态伙伴, 166, 182
- 主条目, 415

C H

- 出厂设置复位, 376

F

- 发现, 379
- 发现上传在线 CPU, 80
- 发送参数组态, 166, 182
- 发热区, 24, 27

D

- 动态绑定, 96

K

- 扩展 S7-1200 的能力, 19

入门手册

设备手册, 01/2015, A5E02486780-AG

S

- 扫描周期
 - 强制操作, 370, 370

G

- 过程映像
 - 状态, 367, 369
 - 监视, 367
 - 监视状态或值, 366
 - 强制, 369
 - 强制操作, 370

X

- 协议
 - ISO on TCP, 157
 - Modbus, 194
 - PROFINET RT, 157
 - TCP, 157
 - UDP, 157
 - USS, 194
 - 自由口, 194
 - 通信, 194
- 西门子技术支持, 5

Z

- 在 RUN 模式下下载
 - 从 STEP 7 启动, 382
 - 概述, 380
- 在 RUN 模式下调试, 380
- 在 RUN 模式下编辑, (? RUN ?????)
- 在线
 - IP 地址, 375
 - RUN/STOP 按钮, 37
 - 比较并同步, 374

- 发现, 379
 - 存储器使用情况监视, 365
 - 时间, 375
 - 状态, 367
 - 转到在线, 363
 - 重设 DB 的起始值, 372
 - 捕获 DB 的值, 372
 - 监视状态或值, 366
 - 监视表格, 367
 - 监控表, 366
 - 循环时间监视, 365
 - 强制, 369
 - 强制操作, 370
 - 操作员面板, 37, 59, 364
 - 在线和诊断工具
 - 在 RUN 模式下下载, 380
 - 在编辑器之间拖放, 39
- C**
- 存储区
 - 立即访问, 70
 - 对布尔值或位值进行寻址, 71
 - 过程映像, 70
 - 全局存储器, 70
 - 临时存储器, 70
 - 数据块, 70
 - 存储卡
 - 装载存储器, 66
 - 存储器
 - 工作存储器, 66
 - 外围设备输入地址 (强制表格), 369
 - 时钟存储器, 88
 - 系统存储器, 88
 - 临时存储器 (L), 70
 - 保持性存储器, 66
 - 装载存储器, 66
- 存储器使用情况监视, 在线, 365
- G**
- 轨迹功能, 383
- W**
- 网页
 - STEP 7 服务、支持和文档, 4
 - 网络
 - 网络连接, 54
 - 网络连接
 - HMI 设备, 54
 - 连接设备, 150
 - 网络通信, 149
- Y**
- 优先级
 - 处理优先级, 62
 - 优先等级, 61
- R**
- 任务卡
 - 列和标题, 38
 - 任务卡中的列和标题, 38
- Z**
- 自由口协议, 194
- X**
- 向 LAD 或 FBD 指令添加输入或输出, 35

Q

全局存储器, 70
 全局库
 USS, 198
 全局数据块, 70, 105

C H

创建 HMI 连接, 54
 创建用户自定义的 Web 页面 DB, 283
 创建用户定义的 Web 页面, 281
 创建网络连接
 PLC 之间, 150
 在 PLC 与 HMI 之间, 54

D

多节点连接
 以太网协议, 181
 连接类型, 181

A

安全性
 CPU, 93
 与 CPU、存储卡或密码绑定, 96
 代码块的专有技术保护, 95
 访问保护, 93
 复制保护, 96
 安装
 气流, 27, 27
 尺寸, 24
 发热区, 24, 24, 27, 27
 安装尺寸, 24
 冷却, 27, 27
 空隙, 27, 27
 信号模块 (SM), 20

准则, 26, 26

概述, 26

S H

设计 PLC 系统, 60, 99
 设备更换
 用 V4.1 CPU 更换 V3.0 CPU, 477
 步骤, 83
 设备组态, 79
 AS-i, 177
 AS-i 端口, 177
 PROFIBUS, 173
 PROFINET, 91
 以太网端口, 91
 发现, 80
 网络连接, 150
 更改设备类型, 83
 拔出的模块, 42
 组态 CPU, 86, 90
 组态模块, 86, 90
 添加新设备, 82
 添加模块, 84
 设备配置
 发现, 379
 设置, 37

F

访问保护, CPU, 93

X

寻址
 布尔值或位值, 71
 过程映像, 70
 存储区, 70

全局存储器, 70
 单独输入 (I) 或输出 (Q), 71
 临时存储器, 70
 数据块, 70

S H

收藏夹工具栏, 35

J

阶段, 291

F

防护等级, 400

Y

运动控制

MC_ChangeDynamic (更改轴的动态设置), 356
 MC_CommandTable, 353
 MC_Halt (暂停轴), 342
 MC_Home (使轴回原点), 339
 MC_MoveAbsolute (绝对定位轴), 344
 MC_MoveJog (在点动模式下移动轴), 351
 MC_MoveRelative (相对定位轴), 346
 MC_MoveVelocity (以预定义速度移动轴), 348
 MC_Power (发布/阻止轴), 334
 MC_ReadParam (读取工艺对象的参数), 361
 MC_Reset (确认错误), 337
 MC_WriteParam (写入工艺对象参数), 358
 回原点 (主动回原点的顺序), 332
 阶段, 291
 使组态参数回原点, 329
 使轴回原点, 328
 组态轴, 294, 304

硬件和软件限位开关, 324

概述, 285

运动控制指令, 333

J

技术支持, 5

技术数据, 391

K

块

OB 数目, 18, 62

一致性检查, 133

入门指南, 106

中断, 18, 62

从在线 CPU 复制块, 373

计数器 (数量和存储器要求), 18

功能 (FC), 60

功能块 (FB), 60

代码块数目, 18

用户程序的大小, 18

启动 OB, 62

事件, 62

定时器 (数量和存储器要求), 18

组织块 (OB), 18, 60, 61, 62

类型, 60

监视, 18

调用另一个代码块, 108

密码保护, 95

嵌套深度, 18

数据块 (DB), 60

块调用

基本, 60

块移动 (MOVE_BLK) 指令, 115

G

- 更改 STEP 7 的设置, 37
- 更改设备, 83
- 更换模块, 42
- 更新用户定义的 Web 页面, 281
- 更新固件
 - 从 STEP 7, 377

L

- 连接
 - HMI 连接, 54
 - S7 连接, 181
 - Web 服务器, 279
 - 以太网协议, 181
 - 网络连接, 54
 - 伙伴, 166, 182
 - 连接 ID, 159
 - 连接数 (PROFINET/PROFIBUS), 154
 - 组态, 163
 - 类型, 多节点连接, 181
 - 通信类型, 151

S H

- 时间
 - 对在线 CPU 进行组态, 375
- 时钟存储器字节, 90

C H

- 串行通信, 194

W

- 位逻辑, 113

X

- 系统存储器字节, 89

Z H

- 状态
 - LED 指示灯 (通信接口), 196
 - LED 指令, 134

L

- 冷却, 27

J

- 间隙, 气流和冷却, 27

Q

- 启动参数, 86

C H

- 初始值
 - 捕获和重设 DB 的起始值, 372

Z H

- 诊断
 - DeviceStates, 135
 - GET_DIAG, 135
 - Get_IM_Data, 135
 - LED 指令, 134
 - ModuleStates, 135
 - 状态指示, 89
 - 缓冲区, 375

H

环境

- 运行条件, 398
- 运输和存储条件, 398

G

规范

- CB 1241 RS485, 463
- CM 1241 RS232, 467
- CM 1241 RS422/485, 465
- CPU 1211C, 404
- CPU 1212C, 404
- CPU 1214C, 404
- CPU 1215C, 404
- CPU 1217C, 404
- SB 1221 4 DI 24 VDC 200 kHz, 410
- SB 1221 4 DI 5 VDC 200 kHz, 410
- SB 1222 4 DQ 24 VDC 200 kHz, 410
- SB 1222 4 DQ 5 VDC 200 kHz, 410
- SB 1223 2 DI/2 DQ 24 VDC, 411
- SB 1223 DI/DQ 24 VDC 200 kHz, 411
- SB 1223 DI/DQ 5 VDC 200 kHz, 411
- SB 1231 AI 1 x 12 位, 429
- SB 1231 AI 1 x 16 位 RTD, 443
- SB 1231 AI 1 x 16 位 TC, 443
- SB 1232 AQ 1 x 12 位, 429
- SM 1221 DI 16 24 VDC, 413
- SM 1221 DI 8 24 VDC, 413
- SM 1222 DQ 16 24 VDC, 415
- SM 1222 DQ 16 继电器, 415
- SM 1222 DQ 8 24 VDC, 415
- SM 1222 DQ 8 继电器, 415
- SM 1222 DQ 8 继电器切换, 415
- SM 1223 DI 120/230 VAC/DQ 继电器, 418
- SM 1223 DI 16/DQ 16 继电器, 416

- SM 1223 DI 8/DQ 8, 416, 416
- SM 1223 DI 8/DQ 8 继电器, 416
- SM 1231 AI 4 x 13 位, 430
- SM 1231 AI 4 x 16 位, 430
- SM 1231 AI 4 x 16 位 TC, 445
- SM 1231 AI 4 x RTD x 16 位, 444
- SM 1231 AI 8 x 13 位, 430
- SM 1231 AI 8 x 16 位 TC, 445
- SM 1231 AI 8 x RTD x 16 位, 444
- SM 1232 AQ 2 x 14 位, 431
- SM 1232 AQ 4 x 14 位, 431
- SM 1234 AI 4 x 13 位/AQ 2 x 14 位, 431
- SM 1278 4xIO-Link 主站, 469
- 工业环境, 395
- 认证, 392
- 电磁兼容性 (EMC), 396
- 环境条件, 398
- 常规技术数据, 391
- 输入的阶跃响应时间, 437
- 模拟量输入的电压表示法, 435
- 模拟量输入的电流表示法, 436
- 模拟量输出的电压表示法, 440
- 模拟量输出的电流表示法, 440
- 额定电压, 400

B

- 拔出的模块, 42

C H

拆分编辑器

- 入门指南, 44, 48, 48

S H

- 事件, 375

事件执行, 62

G

固件更新

从 STEP 7, 377

F

服务与支持, 5

B

变量

入门指南, 44, 48

片段, 72

监视状态或值, 366

强制操作, 370

覆盖, 74

D

定时器

大小, 18

数量, 18

S H

刷新用户定义的 Web 页面, 281

X

限制

Web 服务器, 279

用户定义的 Web 页面, 282

C

参数赋值, 103

X

线性编程, 99

Z

组态

AS-i, 177

HSC (高速计数器), 147

IP 地址, 91

PID_Compact 和 PID_3Step 指令, 239

PID_Temp 指令, 242

PROFIBUS, 173

PROFIBUS 地址, 173

PROFINET, 91

工业以太网端口, 91

用户自定义的 Web 页面, 283

发现, 80

网络连接, 150

启动参数, 86

添加模块, 84

组态参数

CPU, 86, 90

PROFINET, 91

以太网端口, 91

模块, 86, 90

组态控制 (选件处理), 85

组织块

专有技术保护, 95

处理, 101, 101

优先等级, 61

创建, 102

多个循环, 102

函数, 61

组态运行, 103

期间, 61

- X**
- 项目
- HMI 连接, 54
 - HMI 画面, 55
 - 入门指南, 43
 - 与 CPU、存储卡或密码绑定, 96
 - 比较并同步, 374
 - 网络连接, 54
 - 访问保护, 93
 - 变量, 44, 48
 - 限制对 CPU 的访问, 93
 - 保护代码块, 95
 - 添加 HMI 设备, 53
 - 程序, 48
- 项目视图, 33, 33
- PROFINET, 91
 - 设备组态, 79
 - 组态 CPU 参数, 86, 90
 - 组态以太网端口, 91
 - 组态模块, 86, 90
 - 添加新设备, 82
- Z H**
- 指令
- CALCULATE, 50
 - CALCULATE (计算), 118
 - CEIL (上取整), 117
 - CONV (转换), 116
 - CTRL_PWM, 127
 - DeviceStates, 135
 - FLOOR, 117
 - GET, 180
 - GET_DIAG, 135
 - Get_IM_Data, 135
 - HSC (高速计数器), 136, 137
 - LED 状态, 134
 - MC_ChangeDynamic (更改轴的动态设置), 356
 - MC_CommandTable, 353
 - MC_Halt (暂停轴), 342
 - MC_Home (使轴回原点), 339
 - MC_MoveAbsolute (绝对定位轴), 344
 - MC_MoveJog (在点动模式下移动轴), 351
 - MC_MoveRelative (相对定位轴), 346
 - MC_MoveVelocity (以预定义速度移动轴), 348
 - MC_Power (发布/阻止轴), 334
 - MC_ReadParam (读取工艺对象的参数), 361
 - MC_Reset (确认错误), 337
 - MC_WriteParam (写入工艺对象参数), 358
 - ModuleStates, 135
 - NORM_X (标准化), 118
 - PID_Compact (具有集成调节功能的通用 PID 控制器), 207
 - PID_Temp (允许处理温度控制的通用 PID 控制器), 224
 - PUT, 180
 - ROUND, 117
 - SCALE_X (标定), 118
 - TRCV_C, 155
 - TRUNC (截取), 117
 - TSEND_C, 155
 - WWW, 283
 - 入门指南, 48, 49
 - 不可中断移动 (UMOVE_BLK), 115
 - 比较, 115
 - 计数器, 125
 - 可扩展指令, 36
 - 在编辑器之间拖放, 39
 - 列和标题, 38
 - 向 LAD 或 FBD 指令添加输入或输出, 35
 - 收藏夹, 35
 - 运动控制, 333

块移动 (MOVE_BLK), 115
 位逻辑, 113
 状态, 367
 拖放, 34
 指令版本, 38
 标定模拟值, 51
 监视, 367
 监视状态或值, 366
 移动, 115
 添加参数, 49
 插入, 34
 强制操作, 370
 指令版本, 38

A

按移动顺序运行轴命令 (MC_CommandTable), 353

B

标定模拟值, 51
 标准 Web 页面, 275, 276

M

面板 (HMI), 22

B

背景数据块, 70

D

点对点通信, 194

L

临时存储器 (L), 70

X

选件处理 (组态控制), 85

Z H

重设 DB 的起始值, 372

F

复位为出厂设置, 376
 复制保护
 与 CPU、存储卡或密码绑定, 96

X

修改
 程序编辑器状态, 367

B

保护等级, 400
 CPU, 93
 与 CPU、存储卡或密码绑定, 96
 代码块, 95
 保持性存储器, 16, 66

X

信号板 (SB)
 SB 1221 4 DI 24 VDC 200 kHz, 410
 SB 1221 4 DI 5 VDC 200 kHz, 410
 SB 1222 4 DQ 24 VDC 200 kHz, 410
 SB 1222 4 DQ 5 VDC 200 kHz, 410
 SB 1223 2 DI/2 DQ 24 VDC, 411
 SB 1223 DI/DQ 24 VDC 200 kHz, 411
 SB 1223 DI/DQ 5 VDC 200 kHz, 411
 SB 1231 AI 1 x 12 位, 429
 SB 1231 AI 1 x 16 位 RTD, 443

- SB 1231 AI 1 x 16 位 TC, 443
 - SB 1231 RTD, 443
 - SB 1231 TC, 443
 - SB 1232 AQ 1 x 12 位, 429
 - 设备组态, 79
 - 添加模块, 84
 - 概述, 20
 - 输入的电压表示法, 435
 - 输入的电流表示法, 436
 - 模拟量输入的阶跃响应时间, 437
 - 模拟量输出的电压表示法, 440
 - 模拟量输出的电流表示法, 440
- 信号板 (SM)
- 添加新设备, 82
- 信号模块 (SM)
- SM 1221 DI 16 24 VDC, 413
 - SM 1221 DI 8 24 VDC, 413
 - SM 1222 DQ 16 24 VDC, 415
 - SM 1222 DQ 16 继电器, 415
 - SM 1222 DQ 8 24 VDC, 415
 - SM 1222 DQ 8 继电器, 415
 - SM 1222 DQ 8 继电器切换, 415
 - SM 1223 DI 1223 DI 120/230 VAC/DQ 继电器, 418
 - SM 1223 DI 16/DQ 16 继电器, 416
 - SM 1223 DI 8/DQ 8, 416, 416
 - SM 1223 DI 8/DQ 8/DQ 继电器, 416
 - SM 1231 AI 4 x 13 位, 430
 - SM 1231 AI 4 x 16 位, 430
 - SM 1231 AI 4 x 16 位 TC, 445
 - SM 1231 AI 4 x RTD x 16 位, 444
 - SM 1231 AI 8 x 13 位, 430
 - SM 1231 AI 8 x 16 位 TC, 445
 - SM 1231 AI 8 x RTD x 16 位, 444
 - SM 1232 AQ 2 x 14 位, 431
 - SM 1232 AQ 4 x 14 位, 431
 - SM 1234 AI 4 x 13 位/AQ 2 x 14 位, 431
 - SM 1278 4xIO-Link 主站, 469
 - 设备组态, 79
 - 添加新设备, 82
 - 添加模块, 84
 - 概述, 20
 - 模拟量输入的电压表示法, 435
 - 模拟量输入的电流表示法, 436
 - 模拟量输入的阶跃响应时间, 437
 - 模拟量输出的电压表示法, 440
 - 模拟量输出的电流表示法, 440
- 信息资源, 4
- ## M
- 脉冲串输出 (PTO), 127
- ## J
- 将 V3.0 CPU 升级到 V4.1 CPU, 477
- ## F
- 阀门 PID 调节, 214
- ## S H
- 首次扫描指示, 89
- ## Z
- 总线连接器, 20
- ## C
- 测试程序, 132

K

客户支持, 5

J

结构化编程, 99, 99

B

捕获在线 DB 的值, 372

R

热电偶模块概述, 442

热线, 5

H

获取 LED 状态, 134

P

配置

发现, 379

J

监视

LAD 状态, 367

LED 指令, 134

重设 DB 的起始值, 372

捕获 DB 的值, 372

监控表的 LAD 状态和使用, 366

强制表格, 369

强制操作, 370

监视表格, 367

强制, 132

监视程序, 132

监控表

监视, 366

紧凑型交换机模块, CSM 1277, 474

T

特殊模式, TCP 和 ISO on TCP, 158

G

高电位绝缘测试, 399

高速计数器

HSC, 136

无法进行强制, 370

组态, 147

操作, 137

Z H

准则

安装, 26

H

海事认证, 395

L

浪涌抗扰度, 397

B

被动/主动通信

连接 ID, 159

参数, 163

组态伙伴, 166, 182

D

调用结构, 133

T

通过 GPRS 的 TeleService, 185

通信

AS-i 地址, 177

IP 地址, 91

PROFIBUS 地址, 173

PROFINET 和 PROFIBUS, 151

TCON_Param, 163

主动/被动, 163, 166, 182

网络, 149

网络连接, 150

连接 ID, 159

连接数 (PROFINET/PROFIBUS), 154

参数, 163

组态, 163, 166, 182

通信处理器 (CP)

比较表, 19

添加模块, 84

概述, 21

通信板 (CB)

CB 1241 RS485, 463

LED 指示灯, 196

RS485, 196

比较表, 19

添加模块, 84

概述, 20

通信接口

CB 1241 RS485, 463

CM 1241 RS232, 467

RS232 和 RS485, 196

添加模块, 84

模块比较表, 19

通信模块 (CM)

CM 1241 RS232, 467

CM 1241 RS422/RS485, 465

LED 指示灯, 196

RS232 和 RS485, 196

比较表, 19

设备组态, 79

添加 AS-i 主站 CM1243-2 模块, 176

添加 CM 1243-5 (DP 主站) 模块, 172

添加新设备, 82

添加模块, 84

概述, 21

通信模块 (CM), USS 库, 198

J

继电器电气使用寿命, 401

P

排队, 62

J

接线图

CB 1241 RS 485, 464

CPU 1214C AC/DC/继电器, 408

CPU 1214C DC/DC/DC, 409

SB 1221 DI 4 200 kHz, 412

SB 1222 DQ 4 200 kHz, 412

SB 1223 DI 2/DQ 2 200 kHz, 412

SB 1231 AI 1 x 12 位, 430

SB 1231 AI 1 x 16 位 RTD, 443

SB 1231 AI 1 x 16 位 TC, 443

SB 1232 AQ 1 x 12 位, 430

SM 1221 DI 16 24 VDC, 414

SM 1221 DI 8 24 VDC, 414

SM 1222 DQ 16 24 VDC, 416
 SM 1222 DQ 16 继电器, 416
 SM 1223 DI 16 VDC/DQ 16 24 VDC, 417
 SM 1223 DI 16 VDC/DQ 16 继电器, 417
 SM 1223 DI 8 120/230 VAC/DQ 8 继电器, 419
 SM 1231 AI 8 x 13 位, 432
 SM 1231 RTD 4 x 16 位, 445
 SM 1231 RTD 8 x 16 位, 445
 SM 1232 AQ 4 x 13 位, 432
 SM 1234 AI 4 x 13 位/AQ 2 x 14 位, 432
 SM 1278 IO-Link 主站, 472

接线准则

气流和冷却空隙, 27

基本型面板 (HMI), 22

C H

常见问题解答, 4

L

逻辑分析器, 383

Y

移动指令, 115

移动顺序 (MC_CommandTable), 353

T

添加新设备

未指定的 CPU, 80, 379

检测现有硬件, 80

M

密码保护

CPU, 93

与 CPU、存储卡或密码绑定, 96

代码块, 95

对 CPU 进行访问, 93

复制保护, 96

C H

插入设备

未指定的 CPU, 80, 379

插入指令

在编辑器之间拖放, 39

收藏夹, 35

拖放, 34

L

联系信息, 5, 83

H

韩国认证, 394

Y

硬件配置, 79

AS-i, 177

AS-i 端口, 177

PROFIBUS, 173

PROFINET, 91

以太网端口, 91

发现, 80, 379

网络连接, 150

组态 CPU, 86, 90

组态模块, 86, 90

添加新设备, 82

添加模块, 84

C H

程序

- 入门指南, 46, 49
- 与 CPU、存储卡或密码绑定, 96
- 从在线 CPU 复制块, 373
- 示例程序段, 46, 49
- 优先等级, 61
- 组织块 (OB), 101
- 重设 DB 的起始值, 372
- 捕获 DB 的值, 372
- 调用块, 108
- 密码保护, 95
- 数学运算指令, 49

程序卡, 66

程序执行

- 块结构, 60
- 概述, 57

程序块

- 入门指南, 43
- 创建, 106

程序段, LAD 编程, 46

程序信息

- 在调用结构中, 133

程序结构, 99

程序编辑器

- 状态, 367
- 重设 DB 的起始值, 372
- 捕获 DB 的值, 372
- 监视, 367

D

等待时间, 62

X

循环时间监视, 365

Z H

装载存储器, 16, 66

- 用户定义的 Web 页面, 282

Q

强制, 369

- I 存储器, 369, 370
- 外围设备输入, 369, 370
- 扫描周期, 370
- 输入和输出, 370

强制表格

- 寻址外围设备输入, 369
- 强制, 369
- 强制操作, 370

B

编程

- FBD (功能块图), 109
- LAD (梯形图), 108
- PID 概述, 203
- PID_3Step 算法, 204
- PID_3STEP (对阀门进行调节的 PID 控制器), 214
- PID_Compact 算法, 204
- PID_Compact (具有集成调节功能的通用 PID 控制器), 207
- PID_Temp (允许处理温度控制的通用 PID 控制器), 224
- SCL (结构化控制语言), 110, 111
- 入门指南, 48
- 与 CPU、存储卡或密码绑定, 96
- 比较并同步代码块, 374
- 未指定的 CPU, 80, 379
- 可扩展指令, 36
- 在编辑器之间拖放, 39
- 优先等级, 61

向 LAD 或 FBD 指令添加输入或输出, 35

收藏夹, 35

拔出的模块, 42

线性, 99

结构化, 99

插入指令, 34

S H

输入和输出

 监视, 366

输出参数, 103

P

频率、时钟位, 90

L

路由器 IP 地址, 91

Y

遥控, 185

C H

触发

 跟踪, 383

触点

 编程, 46

X

新功能, 29, 29

新建项目

 HMI 连接, 54

 HMI 画面, 55

 入门指南, 43

网络连接, 54

添加 HMI 设备, 53

S H

数字信号板

 SB 1221, 410

 SB 1222, 410

 SB 1223, 411

数字信号模块

 SM 1221, 413

 SM 1222, 415

 SM 1223, 416, 418

数字量输入

 CPU、SM 和 SB 的 VDC 规范, 420

 SM 的 AC 规范, 422

数字量输出

 继电器、24 VDC CPU、SM 和 SB 的规范, 423

数学, 50, 118

数据日志

 数据日志概述, 129

数据处理块 (DHB), 105

数据块

 全局数据块, 70, 105

 组织块 (OB), 101

 背景数据块, 70

 重设起始值, 372, 372

 捕获值, 372, 372

数据类型, 67

 DTL, 68

M

模块

 比较表, 19

 发热区, 24, 27

 组态参数, 86, 90

 信号板 (SB), 20

信号模块 (SM), 20
通信处理器 (CP), 21
通信板 (CB), 20
通信模块 (CM), 21

模拟信号板

SB 1231, 429

SB 1232, 429

模拟信号模块

SM 1231, 430

SM 1231 RTD, 444

SM 1231 热电偶, 445

SM 1232, 431

SM 1234, 431

模拟量 I/O

转换为工程单位, 51

输入的电压表示法, 435

输入的电流表示法, 436

输入的阶跃响应时间, 437

输出的电压表示法, 440

输出的电流表示法, 440

模拟量输入

CPU、SB 和 SM 的规范, 434

RTD/TC SB 和 SM 的规范, 447

D

端口号

分配给通信伙伴, 157

B

播客, 4

E

额定电压, 400, 400

C

操作员面板, 22, 37, 59, 364