

# SIEMENS

## SIMATIC

### S7 S7-1200 可编程控制器




#### 系统手册

前言	
产品概述	1
STEP 7 编程软件	2
安装	3
PLC 概念	4
设备配置	5
编程概念	6
基本指令	7
扩展指令	8
工艺指令	9
通信	10
Web 服务器	11
通信处理器	12
远程服务通信 (SMTP 电子邮件)	13
在线和诊断工具	14
技术规范	A
计算功率预算	B
订货号	C

## 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。

 <b>危险</b>
表示如果不采取相应的小心措施， <b>将会</b> 导致死亡或者严重的人身伤害。
 <b>警告</b>
表示如果不采取相应的小心措施， <b>可能</b> 导致死亡或者严重的人身伤害。
 <b>小心</b>
带有警告三角，表示如果不采取相应的小心措施，可能导致轻微的人身伤害。
<b>小心</b>
不带警告三角，表示如果不采取相应的小心措施，可能导致财产损失。
<b>注意</b>
表示如果不注意相应的提示，可能会出现不希望的结果或状态。


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

### 合格的专业人员

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。

### Siemens 产品

请注意下列说明：

 <b>警告</b>
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。标签中的其他符号可能是一些其他商标，这是出于保护所有权利的目的由第三方使用而特别标示的。

### 责任免除

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

# 前言

## 手册用途

S7-1200 系列是一款可编程逻辑控制器 (PLC, Programmable Logic Controller)，可以控制各种自动化应用。S7-1200 设计紧凑、成本低廉且具有功能强大的指令集，这些特点使它成为控制各种应用的完美解决方案。S7-1200 型号和基于 Windows 的编程工具提供了解决自动化问题时需要的灵活性。

本手册提供了有关 S7-1200 PLC 的安装和编程信息，其主要用户是具备可编程逻辑控制器基本知识的工程师、编程人员、安装人员和电工人员。

## 所需的基本知识

要理解本手册，需要具备自动化和可编程逻辑控制器的基本知识。

## 手册适用范围

本手册介绍了以下产品：

- STEP 7 V11 Basic 和 Professional
- S7-1200 CPU 固件版本 V3.0

有关本手册中所述 S7-1200 产品的完整列表，请参见 技术规范 (页 755)。

## 证书、CE 标签、C 标记和其它标准

请参见 技术规范 (页 755) 以获取更多信息。

## 服务与支持

除了文档之外，我们还在 Internet 的客户支持网站 (<http://www.siemens.com/automation/>) 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。

## 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 系统手册提供了有关完整 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) 和印刷版。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 提供的配套光盘中找到。

- 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 用于访问电子版 (PDF) 的 SIMATIC 文档集，其中包括 STEP 7 的系统手册、入门手册和信息系统。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



# 目录

前言 .....	3
<b>1 产品概述 .....</b>	<b>21</b>
1.1 S7-1200 PLC 简介 .....	21
1.2 CPU 的扩展功能 .....	24
1.3 S7-1200 模块 .....	27
1.4 新功能 .....	28
1.5 HMI 基本型面板 .....	29
<b>2 STEP 7 编程软件 .....</b>	<b>31</b>
2.1 系统要求 .....	31
2.2 使工作更轻松的不同视图 .....	32
2.3 易于使用的工具 .....	33
2.3.1 将指令插入用户程序中 .....	33
2.3.2 从“收藏夹”工具栏调用指令 .....	34
2.3.3 使用简单指令创建复杂等式 .....	34
2.3.4 向 LAD 或 FBD 指令添加输入或输出 .....	36
2.3.5 可扩展指令 .....	37
2.3.6 选择指令的版本 .....	37
2.3.7 修改 STEP 7 的外观和组态 .....	38
2.3.8 在编辑器之间拖放 .....	38
2.3.9 更改 CPU 的工作模式 .....	39
2.3.10 更改 DB 的调用类型 .....	40
2.3.11 暂时从网络中断开设备 .....	40
2.3.12 从组态中虚拟拔出设备 .....	42
<b>3 安装 .....</b>	<b>43</b>
3.1 S7-1200 设备安装准则 .....	43
3.2 功率预算 .....	44
3.3 安装和拆卸步骤 .....	46
3.3.1 S7-1200 设备的安装尺寸 .....	46
3.3.2 安装和拆卸 CPU .....	49
3.3.3 安装和拆卸 SB、CB 或 BB .....	51
3.3.4 安装和拆卸 SM .....	53
3.3.5 安装和拆卸 CM 或 CP .....	55
3.3.6 拆卸和重新安装 S7-1200 端子板连接器 .....	56
3.3.7 安装和卸下扩展电缆 .....	57

3.3.8	TS (远程服务) 适配器 .....	59
3.3.8.1	连接远程服务适配器 .....	59
3.3.8.2	安装 SIM 卡 .....	60
3.3.8.3	安装 TS 适配器单元 .....	61
3.3.8.4	将 TS 适配器安装到墙壁上 .....	62
3.4	接线准则 .....	63
<b>4</b>	<b>PLC 概念 .....</b>	<b>69</b>
4.1	用户程序的执行 .....	69
4.1.1	CPU 的工作模式 .....	71
4.1.2	在 RUN 模式下处理扫描周期 .....	74
4.1.3	组织块 (OB) .....	75
4.1.4	事件执行的优先级与排队 .....	76
4.1.5	监视循环时间 .....	81
4.1.6	CPU 存储器 .....	83
4.1.6.1	系统和时钟存储器 .....	85
4.1.7	诊断缓冲区 .....	87
4.1.8	日时钟 .....	87
4.1.9	组态从 RUN 切换到 STOP 时的输出 .....	88
4.2	数据存储、存储区、I/O 和寻址 .....	88
4.2.1	访问 S7-1200 的数据 .....	88
4.3	模拟值的处理 .....	94
4.4	数据类型 .....	94
4.4.1	Bool、Byte、Word 和 DWord 数据类型 .....	96
4.4.2	整数数据类型 .....	97
4.4.3	浮点型实数数据类型 .....	97
4.4.4	时间和日期数据类型 .....	98
4.4.5	字符和字符串数据类型 .....	100
4.4.6	数组数据类型 .....	101
4.4.7	数据结构数据类型 .....	103
4.4.8	PLC 数据类型 .....	103
4.4.9	指针数据类型 .....	103
4.4.9.1	“Pointer”指针数据类型 .....	104
4.4.9.2	“Any”指针数据类型 .....	105
4.4.9.3	“Variant”指针数据类型 .....	107
4.4.10	访问一个变量数据类型的“片段” .....	107
4.4.11	访问带有一个 AT 覆盖的变量 .....	109
4.5	使用存储卡 .....	111
4.5.1	在 CPU 中插入存储卡 .....	112
4.5.2	将项目复制到存储卡之前组态 CPU 的启动参数 .....	113
4.5.3	将存储卡用作“传送”卡 .....	113
4.5.4	将存储卡用作“程序”卡 .....	116

4.5.5	固件更新 .....	118
4.6	丢失密码后恢复 .....	120
<b>5</b>	<b>设备配置 .....</b>	<b>121</b>
5.1	插入 CPU .....	122
5.2	检测未指定的 CPU 的组态 .....	123
5.3	将模块添加到组态 .....	124
5.4	组态 CPU 的运行 .....	125
5.5	组态模块的参数 .....	127
5.6	组态 CPU 以进行通信 .....	128
5.6.1	创建网络连接 .....	128
5.6.2	组态本地/伙伴连接路径 .....	129
5.6.3	PROFINET 连接的参数 .....	132
5.6.4	分配 Internet 协议 (IP) 地址 .....	135
5.6.4.1	为编程设备和网络设备分配 IP 地址 .....	135
5.6.4.2	检查编程设备的 IP 地址 .....	137
5.6.4.3	在线给 CPU 分配 IP 地址 .....	138
5.6.4.4	为项目中的 CPU 组态 IP 地址 .....	139
5.6.5	测试 PROFINET 网络 .....	142
5.6.6	查找 CPU 上的以太网 (MAC) 地址 .....	143
5.6.7	组态网络时间协议同步 .....	144
5.6.8	PROFINET 设备启动时间、命名和地址分配 .....	146
<b>6</b>	<b>编程概念 .....</b>	<b>147</b>
6.1	设计 PLC 系统的指南 .....	147
6.2	构建用户程序 .....	148
6.3	使用块来构建程序 .....	149
6.3.1	组织块 (OB) .....	150
6.3.2	功能 (FC) .....	152
6.3.3	功能块 (FB) .....	152
6.3.4	数据块 (DB) .....	153
6.4	了解数据一致性 .....	154
6.5	编程语言 .....	155
6.5.1	梯形图 (LAD) .....	156
6.5.2	功能块图 (FBD) .....	157
6.5.3	SCL .....	157
6.5.4	LAD、FBD 和 SCL 的 EN 和 ENO .....	164
6.6	保护 .....	166
6.6.1	CPU 的访问保护 .....	166
6.6.2	专有技术保护 .....	167

6.6.3	复制保护.....	168
6.7	下载程序的元素.....	169
6.8	从 CPU 上传.....	170
6.8.1	复制项目元素.....	170
6.8.2	使用比较功能.....	171
6.9	调试和测试程序.....	172
6.9.1	监视和修改 CPU 中的数据.....	172
6.9.2	监视表格和强制表格.....	172
6.9.3	用于显示使用情况的交叉引用.....	173
6.9.4	用于检查调用层级的调用结构.....	174
<b>7</b>	<b>基本指令.....</b>	<b>177</b>
7.1	位逻辑.....	177
7.1.1	位逻辑触点和线圈.....	177
7.1.2	置位和复位指令.....	180
7.1.3	上升沿和下降沿指令.....	183
7.2	定时器.....	185
7.3	计数器.....	193
7.4	比较.....	200
7.4.1	比较.....	200
7.4.2	范围内和范围外指令.....	201
7.4.3	OK 和 Not OK 指令.....	202
7.5	数学.....	203
7.5.1	计算指令.....	203
7.5.2	加法、减法、乘法和除法指令.....	204
7.5.3	求模指令.....	205
7.5.4	取反指令.....	206
7.5.5	递增和递减指令.....	207
7.5.6	绝对值指令.....	207
7.5.7	最小值和最大值指令.....	208
7.5.8	Limit 指令.....	209
7.5.9	浮点型算术运算指令.....	210
7.6	移动.....	212
7.6.1	移动和块移动指令.....	212
7.6.2	FieldRead 和 FieldWrite 指令.....	215
7.6.3	填充指令.....	217
7.6.4	交换指令.....	218
7.7	转换.....	220
7.7.1	CONV 指令.....	220
7.7.2	SCL 的转换指令.....	221
7.7.3	取整和截取指令.....	224

7.7.4	上取整和下取整指令 .....	225
7.7.5	标定和标准化指令 .....	226
7.8	程序控制 .....	229
7.8.1	SCL 程序控制语句概述 .....	229
7.8.2	IF-THEN 语句 .....	230
7.8.3	CASE 语句 .....	231
7.8.4	FOR 语句 .....	232
7.8.5	WHILE-DO 语句 .....	233
7.8.6	REPEAT-UNTIL 语句 .....	234
7.8.7	CONTINUE 语句 .....	235
7.8.8	EXIT 语句 .....	236
7.8.9	GOTO 语句 .....	236
7.8.10	RETURN 语句 .....	237
7.8.11	跳转和标签指令 .....	237
7.8.12	JMP_LIST 指令 .....	238
7.8.13	SWITCH 指令 .....	239
7.8.14	RET 执行控制指令 .....	241
7.8.15	重新触发扫描循环看门狗指令 .....	242
7.8.16	停止扫描循环指令 .....	243
7.8.17	获取错误指令 .....	243
7.9	字逻辑指令 .....	248
7.9.1	AND、OR 和 XOR 指令 .....	248
7.9.2	取反指令 .....	249
7.9.3	编码和解码指令 .....	249
7.9.4	选择、多路复用和多路分用指令 .....	251
7.10	移位和循环 .....	254
7.10.1	移位指令 .....	254
7.10.2	循环指令 .....	255
<b>8</b>	<b>扩展指令 .....</b>	<b>257</b>
8.1	日期和日时钟 .....	257
8.1.1	日期和时间指令 .....	257
8.1.2	设置和读取系统时钟 .....	260
8.1.3	运行时间计时器指令 .....	261
8.1.4	SET_TIMEZONE 指令 .....	263
8.2	字符串和字符 .....	265
8.2.1	String 数据概述 .....	265
8.2.2	S_MOVE 指令 .....	266
8.2.3	字符串转换指令 .....	266
8.2.3.1	字符串到值以及值到字符串的转换 .....	266
8.2.3.2	字符串到字符和字符到字符串的转换 .....	277
8.2.3.3	ASCII 到十六进制和十六进制到 ASCII 的转换 .....	279
8.2.4	字符串操作指令 .....	281

8.2.4.1	LEN .....	282
8.2.4.2	CONCAT .....	282
8.2.4.3	LEFT、RIGHT 和 MID .....	284
8.2.4.4	DELETE .....	285
8.2.4.5	INSERT .....	286
8.2.4.6	REPLACE .....	287
8.2.4.7	FIND.....	289
8.3	分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface) .....	290
8.3.1	分布式 I/O 指令 .....	290
8.3.2	RDREC 和 WRREC .....	291
8.3.3	RALRM .....	294
8.3.4	RDREC、WRREC 和 RALRM 的 STATUS 参数 .....	297
8.3.5	DPRD_DAT 和 DPWR_DAT .....	302
8.3.6	DPNRM_DG.....	305
8.4	中断.....	308
8.4.1	附加和分离指令 .....	308
8.4.2	循环中断.....	311
8.4.2.1	SET_CINT (Set 循环中断) .....	311
8.4.2.2	QRY_CINT (Query 循环中断) .....	313
8.4.3	延时中断.....	315
8.4.4	异步事件中断.....	317
8.5	诊断 (PROFINET 或 PROFIBUS) .....	318
8.5.1	诊断指令.....	318
8.5.2	分布式 I/O 的诊断事件.....	318
8.5.3	LED 指令.....	319
8.5.4	DeviceStates 指令.....	321
8.5.5	ModuleStates 指令.....	322
8.5.6	GET_DIAG 指令.....	324
8.6	脉冲.....	333
8.6.1	CTRL_PWM 指令.....	333
8.6.2	脉冲输出的作用.....	335
8.6.3	组态 PWM 的脉冲通道 .....	336
8.7	数据记录.....	338
8.7.1	数据日志记录结构 .....	338
8.7.2	控制数据日志的程序指令.....	339
8.7.2.1	DataLogCreate.....	339
8.7.2.2	DataLogOpen.....	342
8.7.2.3	DataLogClose .....	344
8.7.2.4	DataLogWrite .....	346
8.7.2.5	DataLogNewFile.....	347
8.7.3	使用数据日志.....	350
8.7.4	对数据日志文件大小的限制 .....	351

8.7.5	数据日志示例程序.....	354
8.8	数据块控制 .....	358
8.8.1	READ_DBL、WRIT_DBL（读取或写入装载存储器中的 DB）.....	358
8.9	“扩展”指令的常见错误代码 .....	361
<b>9</b>	<b>工艺指令 .....</b>	<b>363</b>
9.1	高速计数器 .....	363
9.1.1	高速计数器的使用方法 .....	365
9.1.2	组态 HSC .....	373
9.2	PID 控制.....	374
9.2.1	插入 PID 指令和工艺对象 .....	376
9.2.2	PID_Compact 指令 .....	378
9.2.3	PID_Compact 指令 ErrorBit 参数.....	383
9.2.4	PID_3STEP 指令.....	384
9.2.5	PID_3STEP 指令 ErrorBit 参数.....	391
9.2.6	组态 PID 控制器 .....	393
9.2.7	调试 PID 控制器 .....	396
9.3	运动控制.....	397
9.3.1	组态轴 .....	401
9.3.2	组态 TO_CommandTable_PTO.....	404
9.3.3	运动控制指令.....	407
9.3.3.1	MC_Power 指令.....	407
9.3.3.2	MC_Reset 指令 .....	410
9.3.3.3	MC_Home 指令 .....	411
9.3.3.4	MC_Halt 指令 .....	414
9.3.3.5	MC_MoveAbsolute 指令 .....	416
9.3.3.6	MC_MoveRelative 指令 .....	418
9.3.3.7	MC_MoveVelocity 指令 .....	420
9.3.3.8	MC_MoveJog 指令 .....	423
9.3.3.9	MC_CommandTable 指令 .....	425
9.3.3.10	MC_ChangeDynamic .....	428
9.3.4	S7-1200 的运动控制操作.....	430
9.3.4.1	用于运动控制的 CPU 输出 .....	430
9.3.4.2	用于运动控制的硬件和软件限位开关 .....	432
9.3.4.3	回原点 .....	435
9.3.4.4	冲击限制.....	440
9.3.5	调试.....	441
9.3.6	监视激活的命令 .....	444
9.3.6.1	监视具有输出参数“Done”的 MC 指令 .....	444
9.3.6.2	监视 MC_Velocity 指令 .....	449
9.3.6.3	监视 MC_MoveJog 指令 .....	453

<b>10</b>	<b>通信</b> .....	<b>457</b>
10.1	支持的异步通信连接数.....	458
10.2	PROFINET.....	459
10.2.1	本地/伙伴连接.....	459
10.2.2	开放式用户通信.....	461
10.2.2.1	PROFINET 指令的连接 ID.....	461
10.2.2.2	协议.....	463
10.2.2.3	特殊模式.....	464
10.2.2.4	TCP 和 ISO on TCP.....	465
10.2.2.5	UDP.....	481
10.2.2.6	T_CONFIG.....	486
10.2.2.7	指令的公共参数.....	493
10.2.3	与编程设备通信.....	495
10.2.3.1	建立硬件通信连接.....	495
10.2.3.2	配置设备.....	496
10.2.3.3	分配 Internet 协议 (IP) 地址.....	497
10.2.3.4	测试 PROFINET 网络.....	497
10.2.4	HMI 到 PLC 通信.....	497
10.2.4.1	组态两个设备之间的逻辑网络连接.....	498
10.2.5	PLC 到 PLC 通信.....	499
10.2.5.1	组态两个设备之间的逻辑网络连接.....	500
10.2.5.2	组态两台设备间的本地/伙伴连接路径.....	500
10.2.5.3	组态传送 (发送) 和接收参数.....	501
10.2.6	配置 CPU 和 PROFINET IO 设备.....	503
10.2.7	诊断.....	507
10.2.8	分布式 I/O 指令.....	507
10.2.9	诊断指令.....	508
10.2.10	分布式 I/O 的诊断事件.....	508
10.3	PROFIBUS.....	508
10.3.1	通信模块 PROFIBUS.....	509
10.3.1.1	连接到 PROFIBUS.....	509
10.3.1.2	PROFIBUS CM 的通信服务.....	510
10.3.1.3	PROFIBUS CM 的其它属性.....	511
10.3.1.4	PROFIBUS 组态示例.....	513
10.3.2	配置 DP 主站和从站设备.....	514
10.3.2.1	添加 CM 1243-5 (DP 主站) 模块和 DP 从站.....	514
10.3.2.2	组态两台 PROFIBUS 设备之间的逻辑网络连接.....	514
10.3.2.3	给 CM 1243-5 模块和 DP 从站分配 PROFIBUS 地址.....	515
10.3.3	分布式 I/O 指令.....	516
10.3.4	诊断指令.....	517
10.3.5	分布式的诊断事件.....	517
10.4	AS-i.....	517
10.4.1	组态 AS-i 主站和从站设备.....	517



10.4.1.1	添加 AS-i 主站 CM 1243-2 和 AS-i 从站 .....	518
10.4.1.2	组态两个 AS-i 设备之间的逻辑网络连接 .....	518
10.4.1.3	组态 AS-i 主站 CM1243-2 的属性 .....	519
10.4.1.4	为 AS-i 从站分配 AS-i 地址 .....	520
10.4.2	在用户程序和 AS-i 从站之间交换数据 .....	521
10.4.2.1	STEP 7 基本组态 .....	521
10.4.2.2	使用 STEP 7 组态从站 .....	522
10.4.3	分布式 I/O 指令 .....	524
10.4.4	使用 AS-i 在线工具 .....	524
10.5	S7 通信 .....	526
10.5.1	GET 和 PUT 指令 .....	526
10.5.2	创建 S7 连接 .....	530
10.5.3	组态两台设备间的本地/伙伴连接路径 .....	531
10.5.4	GET/PUT 连接参数分配 .....	531
10.5.4.1	连接参数 .....	532
10.5.4.2	组态 CPU 间的 S7 连接 .....	535
<b>11</b>	<b>Web 服务器 .....</b>	<b>541</b>
11.1	启用 Web 服务器 .....	542
11.2	标准 Web 页面 .....	543
11.2.1	通过 PC 访问标准 Web 页面 .....	543
11.2.2	标准 Web 页面的布局 .....	545
11.2.3	简介 .....	547
11.2.4	Start .....	548
11.2.5	Identification .....	549
11.2.6	Diagnostic Buffer .....	549
11.2.7	模块信息 .....	550
11.2.8	Communication .....	553
11.2.9	Variable Status .....	554
11.2.10	Data Logs .....	556
11.2.11	更新固件 .....	559
11.3	用户定义的 Web 页面 .....	561
11.3.1	创建 HTML 页面 .....	562
11.3.2	S7-1200 Web 服务器支持的 AWP 命令 .....	563
11.3.2.1	读取变量 .....	564
11.3.2.2	写入变量 .....	565
11.3.2.3	读取特殊变量 .....	567
11.3.2.4	写入特殊变量 .....	569
11.3.2.5	对变量引用使用别名 .....	570
11.3.2.6	定义枚举类型 .....	571
11.3.2.7	通过枚举类型引用 CPU 变量 .....	572
11.3.2.8	创建片段 .....	573
11.3.2.9	导入片段 .....	574

11.3.2.10	组合定义.....	575
11.3.2.11	处理包含特殊字符的变量名称.....	575
11.3.3	组态用户定义 Web 页面的使用.....	577
11.3.4	针对用户定义 Web 页面编写 WWW 指令.....	579
11.3.5	将程序块下载到 CPU.....	580
11.3.6	通过 PC 访问用户定义 Web 页面.....	581
11.3.7	特定于用户定义 Web 页面的限制.....	581
11.3.8	用户定义 Web 页面示例.....	582
11.3.8.1	用于监控风力发电机的 Web 页面.....	582
11.3.8.2	读取和显示控制器数据.....	584
11.3.8.3	使用枚举类型.....	585
11.3.8.4	将用户输入写入控制器.....	586
11.3.8.5	写入特殊变量.....	587
11.3.8.6	引用：远程风力发电机监视 Web 页面的 HTML listing.....	587
11.3.8.7	STEP 7 中示例 Web 页面的组态.....	592
11.3.9	创建多语言用户定义 Web 页面.....	594
11.3.9.1	创建文件夹结构.....	594
11.3.9.2	设置语言切换.....	595
11.3.9.3	组态 STEP 7 以使用多语言页面结构.....	597
11.3.10	高级用户定义 Web 页面控制.....	597
11.4	限制.....	601
11.4.1	禁用 JavaScript 时的受限功能.....	602
11.4.2	不允许使用 cookie 时受限的功能.....	603
11.4.3	导入 Siemens 安全证书.....	603
11.4.4	将 CSV 格式的数据日志导入非 USA/UK 版本的 Microsoft Excel 中.....	604
<b>12</b>	<b>通信处理器.....</b>	<b>607</b>
12.1	使用串行通信接口.....	607
12.2	偏置和端接 RS485 网络连接.....	608
12.3	点对点 (PtP) 通信.....	609
12.3.1	点对点指令.....	610
12.3.1.1	点对点指令的公共参数.....	610
12.3.1.2	PORT_CFG 指令.....	613
12.3.1.3	SEND_CFG 指令.....	615
12.3.1.4	RCV_CFG 指令.....	616
12.3.1.5	SEND_PTP 指令.....	622
12.3.1.6	RCV_PTP 指令.....	625
12.3.1.7	RCV_RST 指令.....	627
12.3.1.8	SGN_GET 指令.....	628
12.3.1.9	SGN_SET 指令.....	629
12.3.2	组态通信端口.....	630
12.3.2.1	管理流控制.....	632
12.3.3	组态传送 (发送) 和接收参数.....	634

12.3.3.1	组态传送（发送）参数 .....	634
12.3.3.2	组态接收参数.....	635
12.3.4	设计 PtP 通信 .....	643
12.3.4.1	轮询架构.....	644
12.3.5	示例：点对点通信.....	645
12.3.5.1	组态通信模块.....	646
12.3.5.2	组态 RS422 和 RS485.....	648
12.3.5.3	编写 STEP 7 程序.....	651
12.3.5.4	组态终端仿真器 .....	652
12.3.5.5	运行示例程序.....	652
12.4	通用串行接口 (USS) 通信 .....	653
12.4.1	使用 USS 协议的要求.....	654
12.4.2	USS_DRV 指令 .....	657
12.4.3	USS_PORT 指令 .....	660
12.4.4	USS_RPM 指令.....	661
12.4.5	USS_WPM 指令 .....	662
12.4.6	USS 状态代码 .....	664
12.4.7	常规驱动器设置信息.....	666
12.5	Modbus 通信 .....	670
12.5.1	Modbus RTU 和 TCP 通信概述 .....	670
12.5.2	Modbus TCP .....	673
12.5.2.1	MB_CLIENT (Modbus TCP).....	673
12.5.2.2	MB_SERVER (Modbus TCP).....	680
12.5.2.3	MB_SERVER 示例：多个 TCP 连接 .....	686
12.5.2.4	MB_CLIENT 示例 1：通过公共 TCP 连接发送多个请求.....	687
12.5.2.5	MB_CLIENT 示例 2：通过不同的 TCP 连接发送多个请求 .....	688
12.5.2.6	MB_CLIENT 示例 3：输出映像写入请求 .....	690
12.5.2.7	MB_CLIENT 示例 4：协调多个请求 .....	690
12.5.3	Modbus RTU .....	691
12.5.3.1	MB_COMM_LOAD .....	692
12.5.3.2	MB_MASTER.....	695
12.5.3.3	MB_SLAVE .....	701
12.5.3.4	Modbus RTU 主站示例程序.....	708
12.5.3.5	Modbus RTU 从站示例程序.....	709
12.6	采用 CP 1242-7 的遥控和远程服务 .....	710
12.6.1	连接到 GSM 网络 .....	710
12.6.2	CP 1242-7 的应用 .....	711
12.6.3	CP 的其它属性 .....	713
12.6.4	附件 .....	714
12.6.5	遥控组态示例.....	715

<b>13</b>	<b>远程服务通信 (SMTP 电子邮件)</b>	<b>721</b>
13.1	TM_Mail 发送电子邮件指令	721
<b>14</b>	<b>在线和诊断工具</b>	<b>729</b>
14.1	状态 LED	729
14.2	转到在线并连接到 CPU	731
14.3	在线为 PROFINET IO 设备分配名称	732
14.4	设置 IP 地址和日时钟	734
14.5	复位为出厂设置	734
14.6	在线 CPU 的 CPU 操作员面板	735
14.7	监视循环时间和存储器使用情况	736
14.8	显示 CPU 中的诊断事件	736
14.9	比较离线 CPU 与在线 CPU	737
14.10	监视和修改 CPU 中的值	738
14.10.1	转到在线模式监视 CPU 中的值	739
14.10.2	显示程序编辑器中的状态	739
14.10.3	捕获 DB 的在线值以重设起始值	740
14.10.4	使用监视表格来监视和修改 CPU 中的值	741
14.10.4.1	监视或修改 PLC 变量时使用触发器	743
14.10.4.2	在 STOP 模式下启用输出	743
14.10.5	CPU 中的强制值	744
14.10.5.1	使用强制表格	744
14.10.5.2	强制功能的操作	745
14.11	在 RUN 模式下下载	746
14.11.1	“在 RUN 模式下下载”的先决条件	747
14.11.2	在 RUN 模式下更改程序	747
14.11.3	下载所选块	749
14.11.4	其它块中存在编译错误时下载选定的单个块	750
14.11.5	下载失败时的系统响应	751
14.11.6	在 RUN 模式下下载程序	751
<b>A</b>	<b>技术规范</b>	<b>755</b>
A.1	常规技术规范	755
A.2	CPU 1211C	762
A.2.1	常规规范和特性	762
A.2.2	CPU 1211C 支持的定时器、计数器和代码块	764
A.2.3	数字量输入和输出	767
A.2.4	模拟量输入	769
A.2.4.1	CPU 内置模拟量输入的阶跃响应	769

A.2.4.2	CPU 内置模拟端口的采样时间 .....	770
A.2.5	CPU 1211C 的接线图 .....	770
A.3	CPU 1212C .....	774
A.3.1	常规规范和特性 .....	774
A.3.2	CPU 1212C 支持的定时器、计数器和代码块 .....	776
A.3.3	数字量输入和输出 .....	779
A.3.4	模拟量输入 .....	781
A.3.4.1	CPU 内置模拟量输入的阶跃响应 .....	782
A.3.4.2	CPU 内置模拟端口的采样时间 .....	782
A.3.5	CPU 1212C 的接线图 .....	783
A.4	CPU 1214C .....	786
A.4.1	常规规范和特性 .....	786
A.4.2	CPU 1214C 支持的定时器、计数器和代码块 .....	788
A.4.3	数字量输入和输出 .....	791
A.4.4	模拟量输入 .....	793
A.4.4.1	CPU 内置模拟量输入的阶跃响应 .....	794
A.4.4.2	CPU 内置模拟端口的采样时间 .....	794
A.4.5	CPU 1214C 的接线图 .....	795
A.5	CPU 1215C .....	799
A.5.1	常规规范和特性 .....	799
A.5.2	CPU 1215C 支持的定时器、计数器和代码块 .....	801
A.5.3	数字量输入和输出 .....	803
A.5.4	模拟量输入和输出 .....	805
A.5.4.1	模拟量输入规范 .....	805
A.5.4.2	CPU 内置模拟量输入的阶跃响应 .....	806
A.5.4.3	CPU 内置模拟端口的采样时间 .....	807
A.5.4.4	模拟量输出规格 .....	807
A.5.5	CPU 1215C 的接线图 .....	808
A.6	数字信号模块 (SM) .....	813
A.6.1	SM 1221 数字量输入规范 .....	813
A.6.2	SM 1222 8 点数字量输出技术规格 .....	815
A.6.3	SM 1222 16 点数字量输出技术规格 .....	817
A.6.4	SM 1223 数字量输入/输出 VDC 规范 .....	823
A.6.5	SM 1223 数字量输入/输出 AC 规范 .....	829
A.7	模拟信号模块 (SM) .....	833
A.7.1	SM 1231 模拟量输入模块规范 .....	833
A.7.2	SM 1232 模拟量输出模块规范 .....	838
A.7.3	SM 1234 模拟量输入/输出模块规范 .....	840
A.7.4	模拟量输入的阶跃响应 .....	844
A.7.5	模拟量输入的采样时间和更新时间 .....	844
A.7.6	模拟量输入的电压测量范围 .....	845
A.7.7	模拟量输入的电流测量范围 .....	846

A.7.8	输出 (AQ) 的电压和电流测量范围 (SB 和 SM)	846
A.8	热电偶和 RTD 信号模块 (SM)	848
A.8.1	SM 1231 热电偶	848
A.8.1.1	热电偶的基本操作	851
A.8.1.2	SM 1231 热电偶选型表	852
A.8.2	SM 1231 RTD	854
A.8.2.1	SM 1231 RTD 选型表	857
A.9	数字信号板 (SB)	861
A.9.1	SB 1221 200 kHz 数字量输入规范	861
A.9.2	SB 1222 200 kHz 数字量输出规范	863
A.9.3	SB 1223 200 kHz 数字量输入/输出规范	866
A.9.4	SB 1223 2 X 24 VDC 输入/2 X 24 VDC 输出规范	870
A.10	模拟信号板 (SB)	874
A.10.1	SB 1231 1 路模拟量输入规范	874
A.10.2	SB 1232 1 路模拟量输出规范	877
A.10.3	模拟量输入和输出的测量范围	879
A.10.3.1	模拟量输入的阶跃响应	879
A.10.3.2	模拟量输入的采样时间和更新时间	879
A.10.3.3	模拟量输入的电压测量范围	880
A.10.3.4	模拟量输入的电流测量范围	881
A.10.3.5	输出 (AQ) 的电压和电流测量范围 (SB 和 SM)	881
A.10.4	热电偶 SB	883
A.10.4.1	SB 1231 1 路热电偶模拟量输入规范	883
A.10.4.2	热电偶的基本操作	884
A.10.5	RTD SB	887
A.10.5.1	SB 1231 1 路模拟量 RTD 输入的规范	887
A.10.5.2	SB 1231 RTD 选型表	891
A.11	BB 1297 电池板	893
A.12	通信接口	895
A.12.1	PROFIBUS	895
A.12.1.1	CM 1242-5	895
A.12.1.2	CM 1243-5	897
A.12.2	GPRS	899
A.12.2.1	CP 1242-7	899
A.12.3	CM 1243-2 AS-i 主站	903
A.12.3.1	AS-i 主站 CM 1243-2 的技术数据	903
A.12.3.2	AS-i 主站 CM 1243-2 的电气连接	905
A.12.4	RS232、RS422 和 RS485	906
A.12.4.1	CB 1241 RS485 规范	906
A.12.4.2	CM 1241 RS232 规范	909
A.12.4.3	CM 1241 RS422/485 技术规格	910
A.13	远程服务 (TS 适配器和 TS 适配器模块)	912

A.14	SIMATIC 存储卡 .....	912
A.15	输入仿真器 .....	913
A.16	I/O 扩展电缆 .....	914
A.17	随附产品 .....	915
A.17.1	PM 1207 电源模块.....	915
A.17.2	CSM 1277 紧凑型交换机模块.....	915
<b>B</b>	<b>计算功率预算 .....</b>	<b>917</b>
<b>C</b>	<b>订货号.....</b>	<b>921</b>
C.1	CPU 模块 .....	921
C.2	信号模块 (SM)、信号板 (SB) 以及电池板 (BB).....	921
C.3	通信 .....	923
C.4	其它模块.....	924
C.5	存储卡 .....	925
C.6	Basic HMI 设备 .....	925
C.7	备件和其它硬件 .....	925
C.8	编程软件.....	926
C.9	文档.....	927
	<b>索引 .....</b>	<b>929</b>





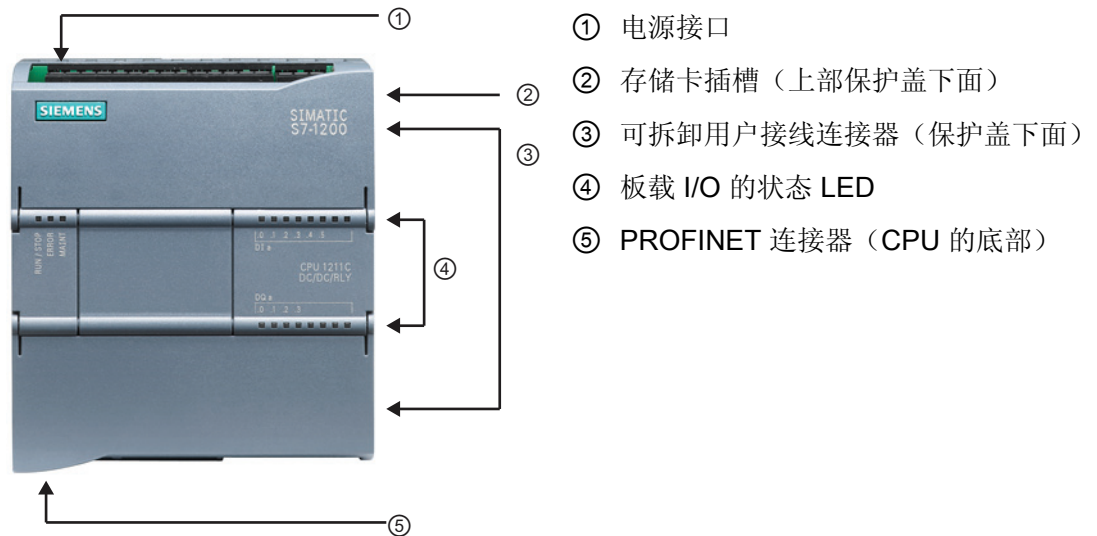
## 产品概述

### 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。

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 或 RS232 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 166)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 167)隐藏特定块中的代码。
- 可以使用 复制保护 (页 168)将程序绑定到特定存储卡或 CPU。

1.1 S7-1200 PLC 简介

表格 1-1 CPU 型号的比较

特征		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C
物理尺寸 (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75	130 x 100 x 75
用户存储器	工作	30 KB	50 KB	75 KB	100 KB
	负载	1 MB	1 MB	4 MB	4 MB
	保持性	10 KB	10 KB	10 KB	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	14 点输入/10 点输出
	模拟量	2 路输入	2 路输入	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	1024 个字节	1024 个字节	1024 个字节
	输出 (Q)	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
信号模块 (SM) 扩展		无	2	8	8
信号板 (SB)、电池板 (BB) 或通信板 (CB)		1	1	1	1
通信模块 (CM) (左侧扩展)		3	3	3	3
高速计数器	总计	3 个内置 I/O, SB 为 5 个	4 个内置 I/O, SB 为 6 个	6	6
	单相	3 个, 100 kHz SB: 2 个, 30 kHz	3 个, 100 kHz 1 个, 30 kHz SB: 2 个, 30 kHz	3 个, 100 kHz 3 个, 30 kHz	3 个, 100 kHz 3 个, 30 kHz
		正交相位	3 个, 80 kHz SB: 2 个, 20 kHz	3 个, 80 kHz 1 个, 20 kHz SB: 2 个, 20 kHz	3 个, 80 kHz 3 个, 20 kHz
脉冲输出 <sup>1</sup>		4	4	4	4
存储卡		SIMATIC 存储卡 (选件)			
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)			

特征	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C
PROFINET	1 个以太网通信端口			2 个以太网通信端口
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			

<sup>1</sup> 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。

每个 CPU 提供专用的 HMI 连接，以支持最多 3 个 HMI 设备。支持的 HMI 总数受组态中 HMI 面板类型的影响。例如，可以将最多 3 个 SIMATIC 基本面板连接到 CPU，或者最多可以连接两个 SIMATIC 精智面板与一个附加基本面板。

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 755)。

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	30 KB (CPU 1211C) 50 KB (CPU 1212C) 64 KB (CPU 1214C 和 CPU 1215C)
	数量	一共多达 1024 个块 (OB + FB + FC + DB)
	FB、FC 和 DB 的地址范围	1 到 65535 (例如 FB 1 到 FB 65535)
	嵌套深度	16 (从程序循环或启动 OB 开始)；4 (从延时中断、日时钟中断、循环中断、硬件中断、时间错误中断或诊断错误中断 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：OB 1、OB 200 到 OB 65535
	启动	多个：OB 100、OB 200 到 OB 65535
	延时中断和循环中断	4 <sup>1</sup> (每个事件 1 个)：OB 200 到 OB 65535
	硬件中断 (沿和 HSC)	50 (每个事件 1 个)：OB 200 到 OB 65535
	时间错误中断	1: OB 80
	诊断错误中断	1: OB 82

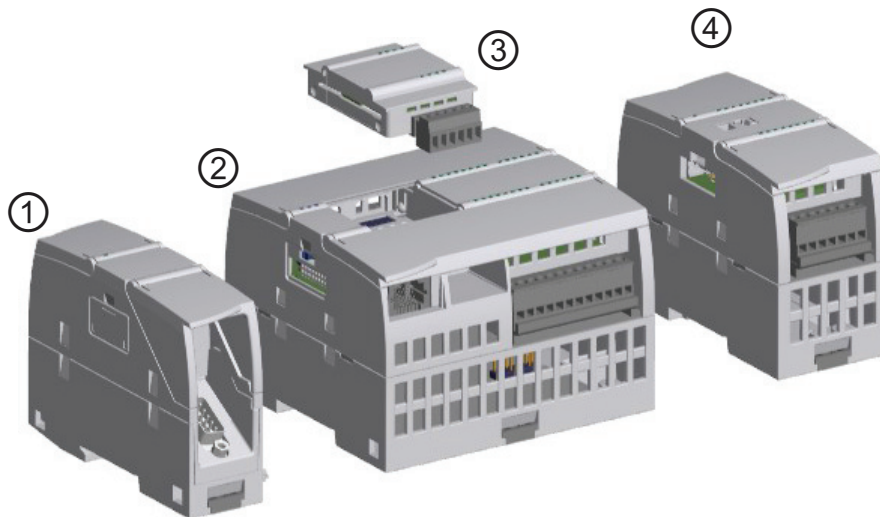
1.2 CPU 的扩展功能

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

1 延时中断和循环中断在 CPU 中使用相同的资源。延时中断和循环中断的总和只能为 4 个。不能有 4 个延时中断和 4 个循环中断。

## 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 755)。



- ① 通信模块 (CM)、通信处理器 (CP) 或 TS 适配器
- ② CPU
- ③ 信号板 (SB)、通信板 (CB) 或电池板 (BB)
- ④ 信号模块 (SM)

表格 1-3 数字量信号模块和信号板

类型	仅输入	仅输出	输入/输出组合
③ 数字量 SB	<ul style="list-style-type: none"> <li>• 4 x 24VDC 输入, 200 kHz</li> <li>• 4 x 5VDC 输入, 200 kHz</li> </ul>	<ul style="list-style-type: none"> <li>• 4 x 24VDC 输出, 200 kHz</li> <li>• 4 x 5VDC 输出, 200 kHz</li> </ul>	<ul style="list-style-type: none"> <li>• 2 x 24VDC 输入/2 x 24VDC 输出</li> <li>• 2 x 24VDC 输入/2 x 24VDC 输出, 200 kHz</li> <li>• 2 x 5VDC 输入/2 x 5VDC 输出, 200 kHz</li> </ul>
④ 数字量 SM	<ul style="list-style-type: none"> <li>• 8 x 24VDC 输入</li> </ul>	<ul style="list-style-type: none"> <li>• 8 x 24VDC 输出</li> <li>• 8 x 继电器输出</li> <li>• 8 x 继电器输出 (切换)</li> </ul>	<ul style="list-style-type: none"> <li>• 8 x 24VDC 输入/8 x 24VDC 输出</li> <li>• 8 x 24VDC 输入/8 x 继电器输出</li> <li>• 8 x 120/230VAC 输入/8 x 继电器输出</li> </ul>
	<ul style="list-style-type: none"> <li>• 16 x 24VDC 输入</li> </ul>	<ul style="list-style-type: none"> <li>• 16 x 24VDC 输出</li> <li>• 16 x 继电器输出</li> </ul>	<ul style="list-style-type: none"> <li>• 16 x 24VDC 输入/16 x 24VDC 输出</li> <li>• 16 x 24VDC 输入/16 x 继电器输出</li> </ul>

表格 1-4 模拟量信号模块和信号板

类型	仅输入	仅输出	输入/输出组合
③ 模拟量 SB	<ul style="list-style-type: none"> <li>• 1 x 12 位模拟量输入</li> <li>• 1 x 16 位 RTD</li> <li>• 1 x 16 位热电偶</li> </ul>	<ul style="list-style-type: none"> <li>• 1 x 模拟量输出</li> </ul>	-
④ 模拟量 SM	<ul style="list-style-type: none"> <li>• 4 x 模拟量输入</li> <li>• 4 x 模拟量输入 x 16 位</li> <li>• 8 x 模拟量输入</li> <li>• 热电偶: <ul style="list-style-type: none"> <li>- 4 x 16 位 TC</li> <li>- 8 x 16 位 TC</li> </ul> </li> <li>• RTD: <ul style="list-style-type: none"> <li>- 4 x 16 位 RTD</li> <li>- 8 x 16 位 RTD</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 2 x 模拟量输出</li> <li>• 4 x 模拟量输出</li> </ul>	<ul style="list-style-type: none"> <li>• 4 x 模拟量输入/2 x 模拟量输出</li> </ul>

1.2 CPU 的扩展功能

表格 1-5 通信接口

模块	类型	说明
① 通信模块 (CM)	RS232	全双工
	RS422/485	全双工 (RS422) 半双工 (RS485)
	PROFIBUS 主站	DPV1
	PROFIBUS 从站	DPV1
	AS-i 主站 (CM 1243-2)	AS-Interface 接口
① 通信处理器 (CP)	调制解调器连接性	GPRS
① 通信板 (CB)	RS485	半双工
① TeleService	TS Adapter IE Basic <sup>1</sup>	连接到 CPU
	TS Adapter GSM	GSM/GPRS
	TS Adapter Modem	调制解调器
	TS Adapter IE-ISDN	ISDN
	TS Adapter RS232	RS232

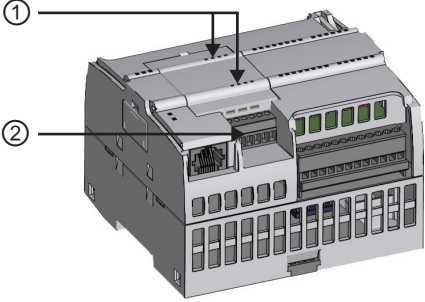
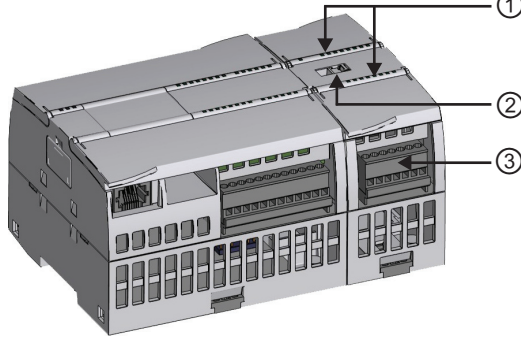
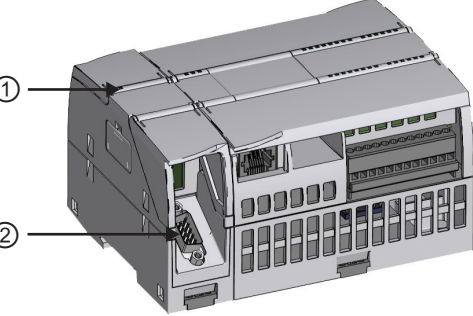
<sup>1</sup> TS 适配器允许用户将各种通信接口连接到 CPU 的 PROFINET 端口。将 TS 适配器安装在 CPU 左侧，然后将 TS 适配器模块（最多 3 个）连入 TS 适配器。

表格 1-6 其它板

模块	说明
③ 电池板	从 CPU 前端的扩展板接口插入。提供长期的实时时钟备份。

### 1.3 S7-1200 模块

表格 1-7 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"> <li>• 信号板 (SB) 可为 CPU 提供附加 I/O。SB 连接在 CPU 的前端。</li> <li>• 通信板 (CB) 可以为 CPU 增加其它通信端口。</li> <li>• 电池板 (BB) 可提供长期的实时时钟备份。</li> </ul>		<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"> <li>• 数字量 I/O</li> <li>• 模拟量 I/O</li> <li>• RTD 和热电偶</li> </ul>		<p>① 状态 LED</p>
		<p>② 总线连接器</p>
		<p>③ 可拆卸用户接线连接器</p>
<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。CP 可以提供其它类型的通信功能, 例如通过 GPRS 网络连接 CPU。</p> <ul style="list-style-type: none"> <li>• CPU 最多支持 3 个 CM 或 CP</li> <li>• 各 CM 或 CP 连接在 CPU 的左侧 (或连接到另一 CM 或 CP 的左侧)</li> </ul>		<p>① 状态 LED</p>
		<p>② 通信连接器</p>

## 1.4 新功能

以下是此版本中新添加的功能：

- 一个标准 Web 服务器页面，用于执行 CPU 固件更新 (页 559)
- 能够使用三个 PROFIBUS DP CM 1243-5 主站模块或三个 AS-i CM 1243-2 主站模块

---

### 说明

要使用三个 AS-i 模块作为主站，必须更新 AS-i 模块的固件。

---

## 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：

- 新的和改进的 CPU：
  - 新的 CPU 1215C DC/DC/DC、CPU 1215C DC/DC/继电器和 CPU 1215C AC/DC/继电器提供了 100KB 的工作存储器、双以太网和模拟量输出。
  - 新的和改进的 CPU 1211C、CPU 1212C 和 CPU 1214C 具有更短的处理时间、可使用 4 个 PTO（CPU 1211C 需要信号板）、更大的保持性存储器 (10 KB) 以及更长的日时钟保持时间 (20 天)。
- 新 I/O 信号模块：SM 1231 AI 4 x 16 位提供了更高的采样率而且增加了位数。
- 新的电池板 (BB 1297) 可提供长期的实时时钟备份。BB 1297 可插入 S7-1200 CPU（固件版本 3.0 及更高版本）的信号板插槽中。

要使用新模块，必须使用 STEP 7 V11 SP2 Update 3 或更高版本（Basic 或 Professional），而且必须通过 STEP 7“选项 > 支持包”(Options > Support Packages) 菜单命令为新模块下载硬件支持包 (HSP)。通过西门子服务和支持网站 (<http://support.automation.siemens.com/CN>) 按照相关指示将模块添加到 STEP 7 (TIA Portal) 的硬件目录中。

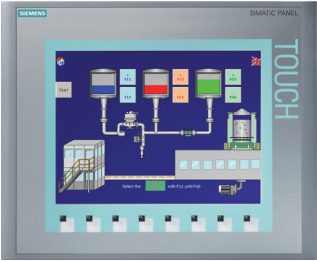


## 1.5 HMI 基本型面板

由于可视化逐渐成为大多数机器设计的标准组件，所以 SIMATIC HMI 基本型面板提供了用于执行基本操作员监控任务的触摸屏设备。所有面板的保护等级均为 IP65 并通过了 CE、UL、cULus 和 NEMA 4x 认证。

HMI 基本型面板	说明	技术数据
 <p>KP 300 Basic PN</p>	<p>3.6" 薄膜键盘，带 10 个可自由组合的按键</p> <ul style="list-style-type: none"> <li>• 单色 (STN, 黑/白)</li> <li>• 87 mm x 31 mm (3.6")</li> <li>• 背光灯颜色可调 (白色、绿色、黄色或红色)</li> <li>• 分辨率: 240 x 80</li> </ul>	<ul style="list-style-type: none"> <li>• 250 个变量</li> <li>• 50 个过程画面</li> <li>• 200 个报警</li> <li>• 25 条曲线</li> <li>• 40 KB 配方存储器</li> <li>• 5 个配方, 20 条数据记录, 20 个条目</li> </ul>
 <p>KTP 400 Basic PN</p>	<p>4" 触摸屏，带 4 个触摸键</p> <ul style="list-style-type: none"> <li>• 单色 (STN, 灰度)</li> <li>• 76.79 mm x 57.59 mm (3.8") 纵向或横向</li> <li>• 分辨率: 320 x 240</li> </ul>	<ul style="list-style-type: none"> <li>• 250 个变量</li> <li>• 50 个过程画面</li> <li>• 200 个报警</li> <li>• 25 条曲线</li> <li>• 40 KB 配方存储器</li> <li>• 5 个配方, 20 条数据记录, 20 个条目</li> </ul>
 <p>KTP 600 Basic PN</p>	<p>6" 触摸屏，带 6 个触摸键</p> <ul style="list-style-type: none"> <li>• 彩色 (TFT, 256 色) 或单色 (STN, 灰度)</li> <li>• 115.2 mm x 86.4 mm (5.7") 纵向或横向</li> <li>• 分辨率: 320 x 240</li> </ul>	<ul style="list-style-type: none"> <li>• 500 个变量</li> <li>• 50 个过程画面</li> <li>• 200 个报警</li> <li>• 25 条曲线</li> <li>• 40 KB 配方存储器</li> <li>• 5 个配方, 20 条数据记录, 20 个条目</li> </ul>

1.5 HMI 基本型面板

HMI 基本型面板	说明	技术数据
	<p>10" 触摸屏，带 8 个触摸键</p> <ul style="list-style-type: none"> <li>• 彩色 (TFT, 256 色)</li> <li>• 211.2 mm x 158.4 mm (10.4")</li> <li>• 分辨率: 640 x 480</li> </ul>	<ul style="list-style-type: none"> <li>• 500 个变量</li> <li>• 50 个过程画面</li> <li>• 200 个报警</li> <li>• 25 条曲线</li> <li>• 40 KB 配方存储器</li> <li>• 5 个配方, 20 条数据记录, 20 个条目</li> </ul>
<p>KTP 1000 Basic PN</p>	<p>15" 触摸屏</p> <ul style="list-style-type: none"> <li>• 彩色 (TFT, 256 色)</li> <li>• 304.1 mm x 228.1 mm (15.1")</li> <li>• 分辨率: 1024 x 768</li> </ul>	<ul style="list-style-type: none"> <li>• 500 个变量</li> <li>• 50 个过程画面</li> <li>• 200 个报警</li> <li>• 25 条曲线</li> <li>• 40 KB 配方存储器 (集成闪存)</li> <li>• 5 个配方, 20 条数据记录, 20 个条目</li> </ul>
	<p>TP 1500 Basic PN</p>	

## STEP 7 编程软件

STEP 7 软件提供了一个用户友好的环境，供用户开发、编辑和监视控制应用所需的逻辑，其中包括用于管理和组态项目中所有设备（例如控制器和 HMI 等设备）的工具。为了帮助用户查找需要的信息，STEP 7 提供了内容丰富的在线帮助系统。

STEP 7 提供了标准编程语言，用于方便高效地开发适合用户具体应用的控制程序。

- LAD（梯形图逻辑）是一种图形编程语言。它使用基于电路图 (页 156) 的表示法。
- FBD（功能块图）是基于布尔代数 (页 157) 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。

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。

---

### 说明

STEP 7 是 TIA Portal 中的编程和组态软件。除了包括 STEP 7 外，TIA Portal 中还包括设计和执行运行过程可视化的 WinCC，以及 WinCC 以及 STEP 7 的在线帮助。

---

## 2.1 系统要求

要在运行 Windows XP 或 Windows 7 操作系统的 PC 上安装 STEP 7 软件，必须以管理员权限登录。

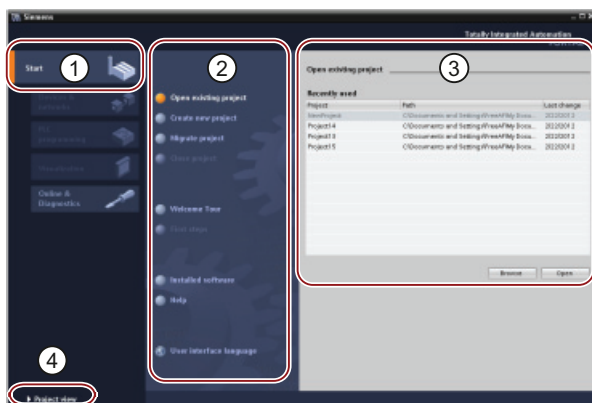
表格 2-1 系统要求

硬件/软件	要求
处理器类型	Pentium M 1.6 GHz 或类似
RAM	1 GB
可用硬盘空间	系统驱动器 C:\ 上 2 GB

硬件/软件	要求
操作系统	<ul style="list-style-type: none"> <li>Windows XP Professional SP3</li> <li>Windows 2003 Server R2 StdE SP2</li> <li>Windows 7 Home Premium (仅限 STEP 7 Basic, 不支持 STEP 7 Professional)</li> <li>Windows 7 (Professional、Enterprise、Ultimate)</li> <li>Windows 2008 Server StdE R2</li> </ul>
图形卡	32 MB RAM 24 位颜色深度
屏幕分辨率	1024 x 768
网络	20 Mbit/s 以太网或更快
光驱	DVD-ROM

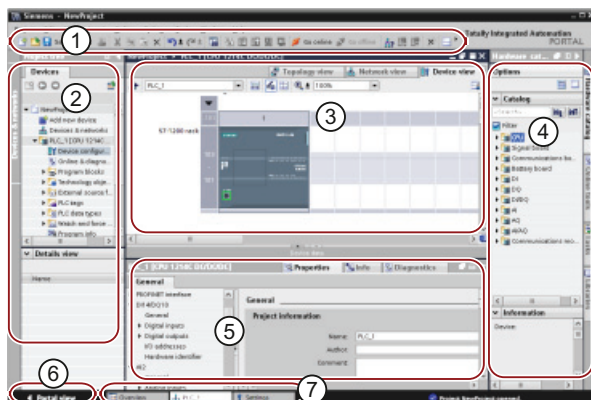
## 2.2 使工作更轻松的不同视图

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。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。

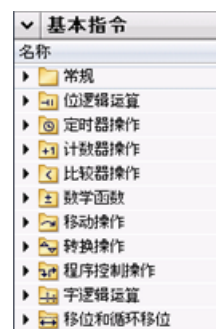
## 2.3 易于使用的工具

### 2.3.1 将指令插入用户程序中

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。

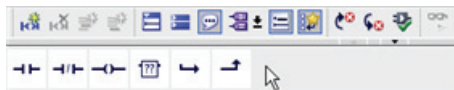


要创建程序，可将指令从任务卡拖动到程序段中。

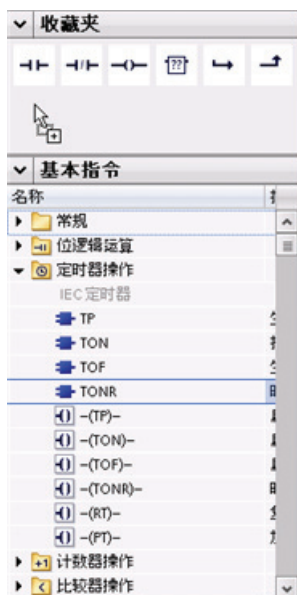


### 2.3.2 从“收藏夹”工具栏调用指令

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。)



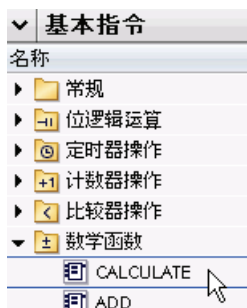
用户可以通过添加新指令方便地自定义“收藏夹”(Favorites)。只需将指令拖放到“收藏夹”(Favorites)。

如此单击即可插入该指令!

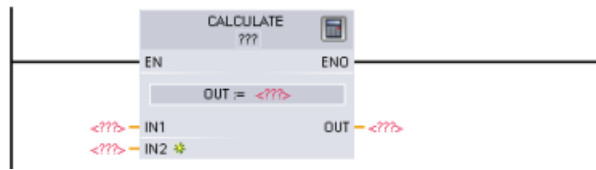


### 2.3.3 使用简单指令创建复杂等式

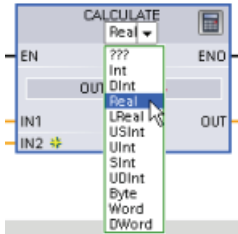
Calculate 指令可以根据定义的等式生成作用于多个输入参数的数学函数, 从而生成结果。



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。



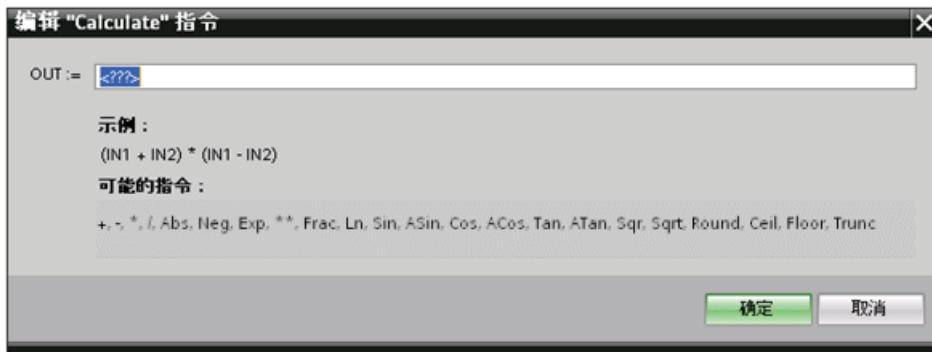
未组态的 **Calculate** 指令提供了两个输入参数和一个输出参数。



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。）

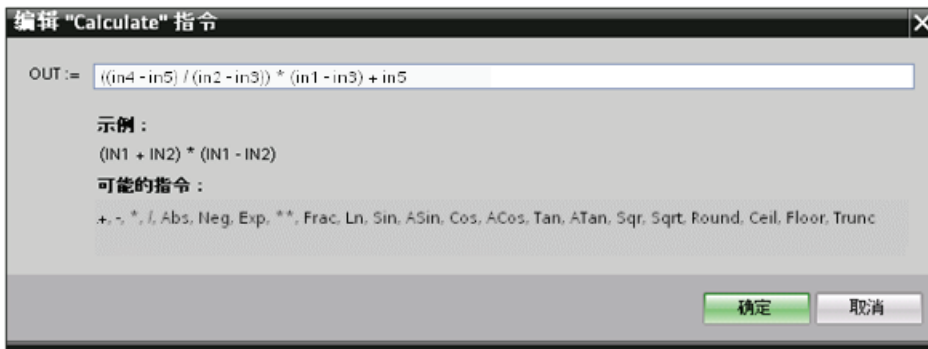
$$\text{Out}_{\text{value}} = ((\text{Out}_{\text{high}} - \text{Out}_{\text{low}}) / (\text{In}_{\text{high}} - \text{In}_{\text{low}})) * (\text{In}_{\text{value}} - \text{In}_{\text{low}}) + \text{Out}_{\text{low}}$$

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

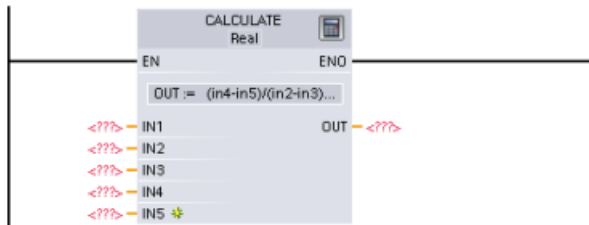
其中:	<b>Out<sub>value</sub></b>	<b>(Out)</b>	标定的输出值
	<b>In<sub>value</sub></b>	<b>(in1)</b>	模拟量输入值
	<b>In<sub>high</sub></b>	<b>(in2)</b>	标定输入值的上限
	<b>In<sub>low</sub></b>	<b>(in3)</b>	标定输入值的下限
	<b>Out<sub>high</sub></b>	<b>(in4)</b>	标定输出值的上限
	<b>Out<sub>low</sub></b>	<b>(in5)</b>	标定输出值的下限

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

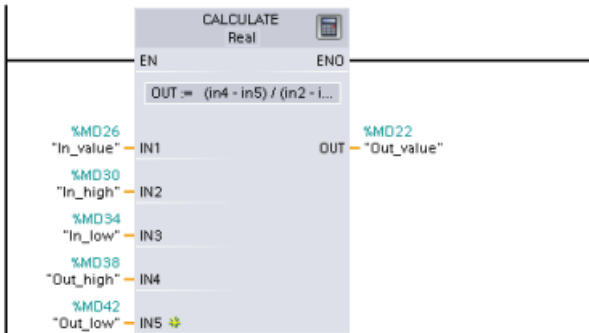
$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$




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。



### 2.3.4 向 LAD 或 FBD 指令添加输入或输出

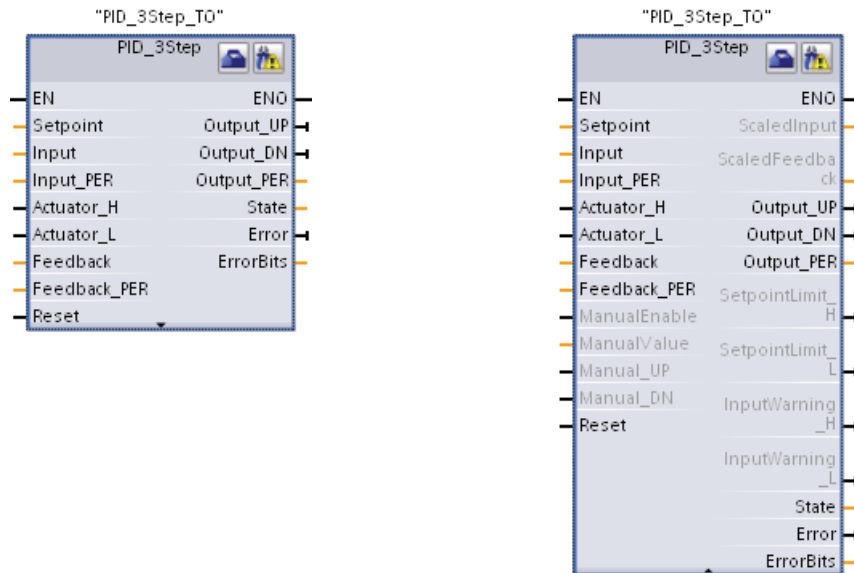
 有些指令允许您另外创建输入或输出。

- 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。



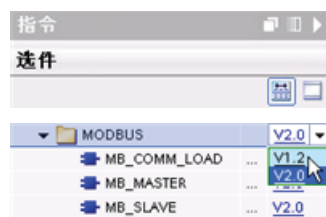
### 2.3.5 可扩展指令

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



### 2.3.6 选择指令的版本

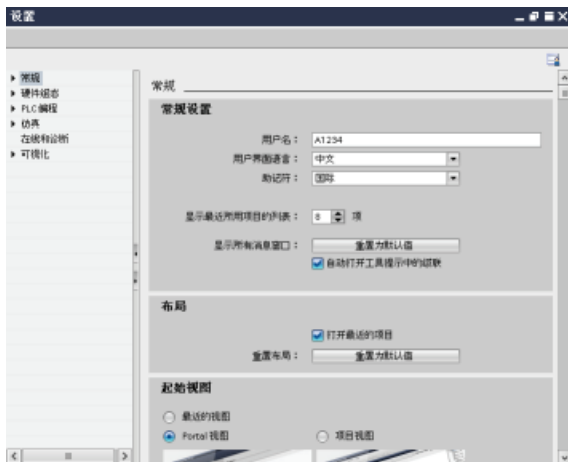
某些指令集（如 Modbus、PID 和运动指令集）经过多个开发和发布周期后形成了多种发布版本。为了有助于确保与较早项目的兼容性以及对这些项目进行移植，STEP 7 允许您选择要插入用户程序中的指令版本。



单击指令树任务卡上的图标可启用指令树的标题和列。

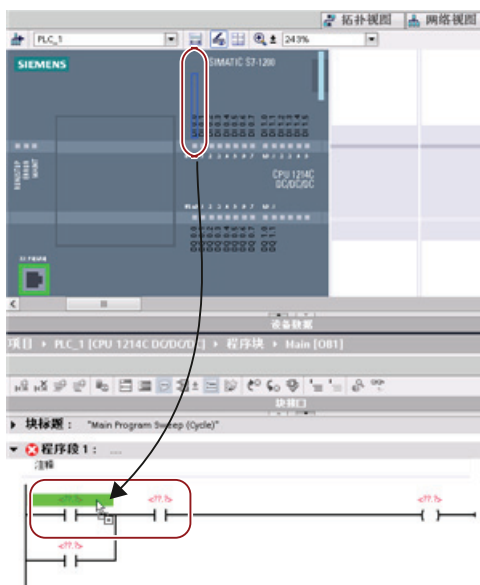
要更改指令版本，需从下拉列表中选择合适的版本。

### 2.3.7 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。

### 2.3.8 在编辑器之间拖放



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 的输入拖放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。



## 2.3.9 更改 CPU 的工作模式

参见

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。

“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。

有关上电时组态默认工作模式的信息，请参见 S7-1200 系统手册中的 CPU 工作模式 (页 71)。

## 2.3.10 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。

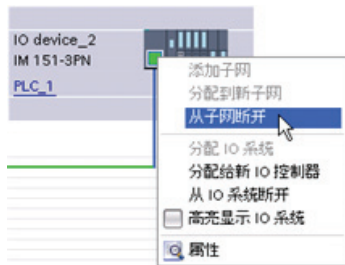


通过“调用选项”(Call options) 对话框可选择单背景数据块或多背景数据块。还可以从可用 DB 的下拉列表中选择具体 DB。

## 2.3.11 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。

### 2.3.12 从组态中虚拟拔出设备



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。

## 安装

### 3.1 S7-1200 设备安装准则

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。



**警告**

SIMATIC S7-1200 PLC 是敞开式控制器。需要将 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。

不遵守这些安装要求可能会导致死亡、人员重伤和/或财产损失。

安装 S7-1200 PLC 时务必遵守这些要求。

#### 将 S7-1200 设备与热辐射、高压和电噪声隔离开

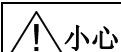
作为布置系统中各种设备的基本规则，必须将产生高压和高电噪声的设备与 S7-1200 等低压逻辑型设备隔离开。

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

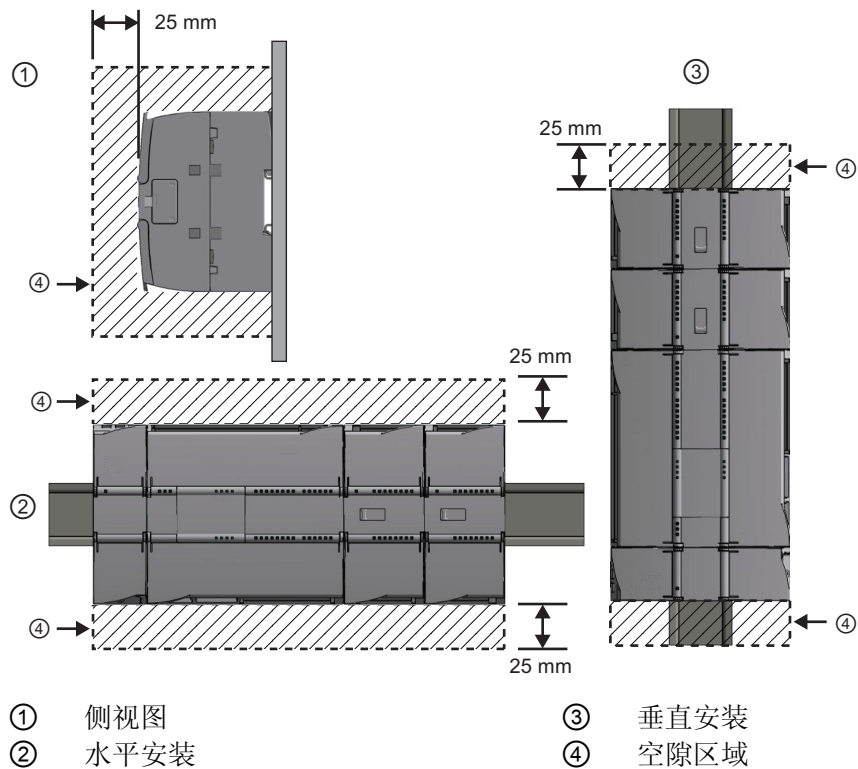
S7-1200 被设计成通过自然对流冷却。为保证适当冷却，在设备上方和下方必须留出至少 25 mm 的空隙。此外，模块前端与机柜内壁间至少应留出 25 mm 的深度。



**小心**

垂直安装时，允许的最大环境温度将降低 10 摄氏度。请按下图所示调整垂直安装的 S7-1200 系统的方位。

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。



### 3.2 功率预算

CPU 有一个内部电源，用于为 CPU、信号模块、信号板和通信模块供电以及用于满足其它 24 VDC 用户的功率要求。

有关 CPU 所提供的 5 VDC 逻辑预算以及信号模块、信号板和通信模块的 5 VDC 功率要求的信息，请参考技术规范 (页 755)。请参考“计算功率预算”(页 917)来确定 CPU 可以为您的配置提供多少电能（或电流）。

CPU 提供 24 VDC 传感器电源，该电源可以为输入点、信号模块上的继电器线圈电源或其它要求供给 24 VDC。如果您的 24 VDC 功率要求超出该传感器电源的预算，则必须给系统增加外部 24 VDC 电源。有关具体 CPU 的 24 VDC 传感器电源功率预算，请参考技术规范 (页 755)。

**说明**

CM 1243-5 (PROFIBUS 主站模块) 需要由 CPU 的 24 VDC 传感器电源供电。



如果需要外部 24 VDC 电源，请确保该电源不要与 CPU 的传感器电源并联。为提高电噪声防护能力，建议连接不同电源的公共端 (M)。

 **警告**

将外部 24 VDC 电源与 24 VDC 传感器电源并联会导致这两个电源之间有冲突，因为每个电源都试图建立自己首选的输出电压电平。

该冲突可能使其中一个电源或两个电源的寿命缩短或立即出现故障，从而导致 PLC 系统的运行不确定。运行不确定可能导致死亡、人员重伤和/或财产损失。

DC 传感器电源和任何外部电源应分别给不同位置供电。

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、SM 的继电器线圈的电源输入或非隔离模拟输入的电源。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**

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。

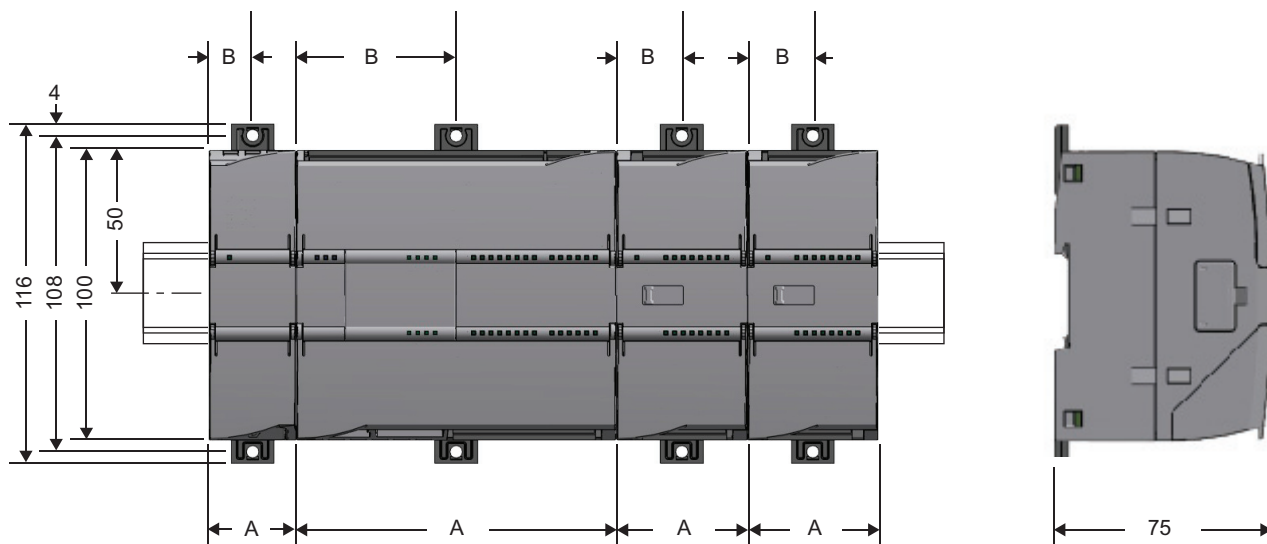
不遵守这些准则可能会导致设备损坏或运行不确定，而后者可能导致死亡、人员重伤和/或财产损失。

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。

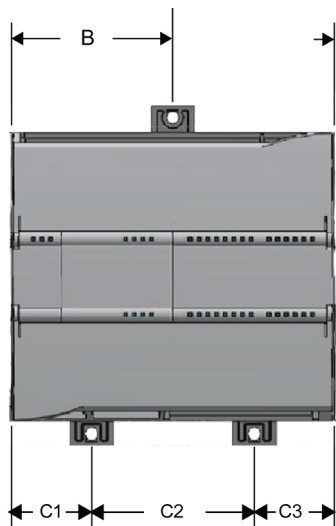
### 3.3 安装和拆卸步骤

#### 3.3.1 S7-1200 设备的安装尺寸

CPU 1211C, CPU 1212C, CPU 1214C



CPU 1215C



表格 3-1 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
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	45	22.5	--
	数字量 DQ 8 x 继电器 (切换)	70	22.5	--
	模拟 16 点 RTD 8 点	70	35	--
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	30	15	--
	TS Adapter IE Basic	60 <sup>1</sup>	15	--

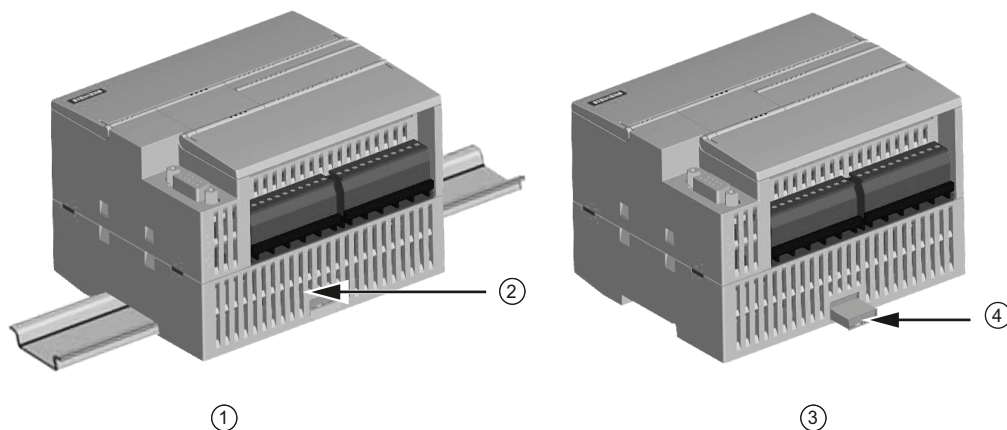
<sup>1</sup> 由于必须安装带有 TS 适配器的 TS 适配器模块，因此总宽度（“宽度 A”）为 60 mm。

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。

安装和拆卸 S7-1200 设备

CPU 可以很方便地安装到标准 DIN 导轨或面板上。可使用 DIN 导轨卡夹将设备固定到 DIN 导轨上。这些卡夹还能掰到一个伸出位置以提供设备面板安装时所用的螺钉安装位置。



- ① DIN 导轨安装
- ③ 面板安装
- ② DIN 导轨卡夹处于锁紧位置
- ④ 卡夹处于伸出位置用于面板安装

在安装或拆卸任何电气设备之前，请确保已关闭相应设备的电源。同时，还要确保已关闭所有相关设备的电源。

**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 模块安装不当可能导致 S7-1200 中的程序工作异常。

如果不是用相同型号、方向或顺序来更换 S7-1200 设备，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。

 <b>警告</b>
请勿在易燃或易爆环境中断开连接设备。 在易燃或易爆环境中断开连接设备可能会引起火灾或爆炸，从而导致死亡、人员重伤和/或财产损失。

<b>小心</b>
静电放电可能会损坏设备或 CPU 上的卡槽。 在拿放设备时，请与已接地的导电垫接触或使用接地腕带。

### 3.3.2 安装和拆卸 CPU

可以将 CPU 安装到 DIN 导轨或面板上。

#### 说明

将全部通信模块连接到 CPU 上，然后将该组件作为一个单元来安装。在安装 CPU 之后分别安装信号模块。

将该单元安装到 DIN 导轨或面板上时，应考虑以下几点：

- 若是 DIN 导轨安装，确保 CPU 和相连 CM 的上部 DIN 导轨卡夹处于锁紧（内部）位置而下部 DIN 导轨卡夹处于伸出位置。
- 将设备安装到 DIN 导轨上后，将下部 DIN 导轨卡夹推到锁紧位置以将设备锁定在 DIN 导轨上。
- 若是面板安装，确保将 DIN 导轨卡夹推到伸出位置。

要将 CPU 安装到面板上，请按以下步骤操作：

1. 按照 安装尺寸 (mm) (页 46) 表中所示的尺寸，执行定位、钻孔和攻丝以准备安装孔 (M4)。
2. 确保 CPU 和所有 S7-1200 设备都与电源断开。

3.3 安装和拆卸步骤

3. 从模块上掰出安装卡夹。确保 CPU 上部和下部的 DIN 导轨卡夹都处于伸出位置。
4. 使用带弹簧和平垫圈的 Pan Head M4 螺钉将模块固定到面板上。不要使用平头螺钉。

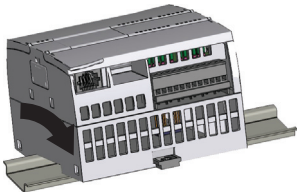
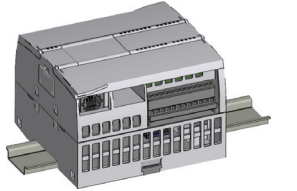
**说明**

螺钉类型将由安装时的材料决定。应施加适当的扭矩，直到弹簧垫圈变平。避免对安装螺钉施加过多扭矩。不要使用平头螺钉。

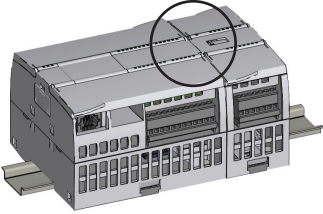
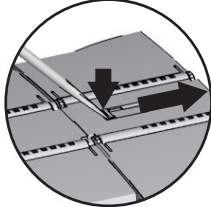
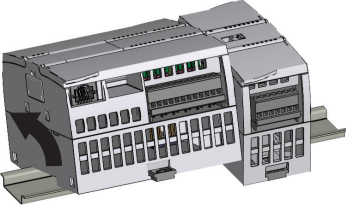
**说明**

如果系统处在多振动环境或采用垂直安装，则通过面板安装 S7-1200 将能提供更高的防护等级。

表格 3-2 将 CPU 安装在 DIN 导轨上

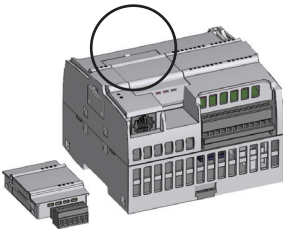
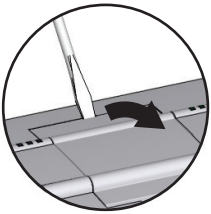
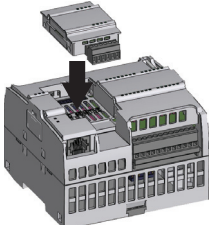
任务	步骤
	<ol style="list-style-type: none"> <li>1. 安装 DIN 导轨。每隔 75 mm 将导轨固定到安装板上。</li> <li>2. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>3. 将 CPU 挂到 DIN 导轨上方。</li> <li>4. 拉出 CPU 下方的 DIN 导轨卡夹以便能将 CPU 安装到导轨上。</li> <li>5. 向下转动 CPU 使其在导轨上就位。</li> </ol>
	<ol style="list-style-type: none"> <li>6. 推入卡夹将 CPU 锁定到导轨上。</li> </ol>

表格 3-3 将 CPU 从 DIN 导轨上卸下

任务	步骤
	 <ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 从 CPU (页 56) 断开 I/O 连接器、接线和电缆。</li> <li>3. 将 CPU 和所有相连的通信模块作为一个完整单元拆卸。所有信号模块应保持安装状态。</li> <li>4. 如果 SM 已连接到 CPU，则需要缩回总线连接器：                         <ul style="list-style-type: none"> <li>- 将螺丝刀放到信号模块上方的小接头旁。</li> <li>- 向下按使连接器与 CPU 相分离。</li> <li>- 将小接头完全滑到右侧。</li> </ul> </li> </ol>
	<ol style="list-style-type: none"> <li>5. 卸下 CPU：                         <ul style="list-style-type: none"> <li>- 拉出 DIN 导轨卡夹从导轨上松开 CPU。</li> <li>- 向上转动 CPU 使其脱离导轨，然后从系统中卸下 CPU。</li> </ul> </li> </ol>

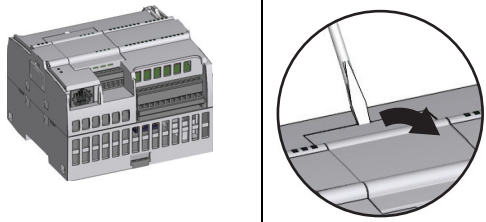
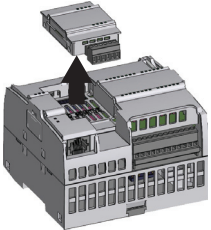
### 3.3.3 安装和拆卸 SB、CB 或 BB

表格 3-4 安装 SB、CB 或 BB 1297

任务	步骤
	<ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 卸下 CPU 上部和下部的端子板盖板。</li> <li>3. 将螺丝刀插入 CPU 上部接线盒盖背面的槽中。</li> <li>4. 轻轻将盖撬起并从 CPU 上卸下。</li> <li>5. 将模块直接向下放入 CPU 上部的安装位置中。</li> <li>6. 用力将模块压入该位置直到卡入就位。</li> <li>7. 重新装上端子板盖子。</li> </ol>
 	

3.3 安装和拆卸步骤

表格 3-5 拆卸 SB、CB 或 BB 1297

任务	步骤
	<ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 卸下 CPU 上部和下部的端子板盖板。</li> <li>3. 将螺丝刀插入模块上部的槽中。</li> <li>4. 轻轻将模块撬起使其与 CPU 分离。</li> <li>5. 将模块直接从 CPU 上部的安装位置中取出。</li> </ol>
	<ol style="list-style-type: none"> <li>6. 将盖板重新装到 CPU 上。</li> <li>7. 重新装上端子板盖子。</li> </ol>

安装或更换 BB 1297 电池板中的电池

BB 1297 要求的电池型号为 CR1025。电池未随 BB 1297 一起提供，必须由用户另行购买。要安装或更换电池，请执行以下步骤：

1. 在 BB 1297 中，将电池正极朝上，负极靠近印刷电路板来安装新电池。
2. BB 1297 已准备好安装到 CPU 中。按照上述安装指示安装 BB 1297。

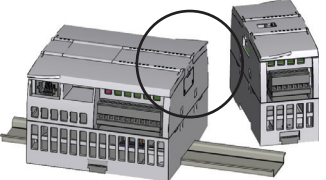
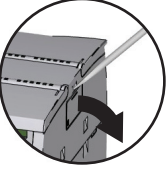
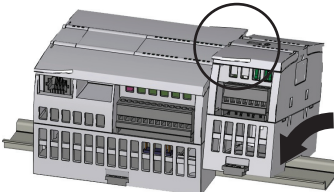

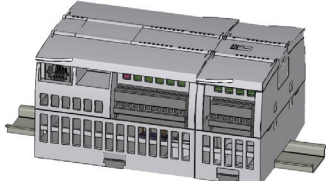
更换 BB 1297 中的电池：

1. 按照上述拆卸指示将 BB 1297 从 CPU 中取出。
2. 使用小号螺丝刀小心地取下旧电池。将电池从卡夹下部推出。
3. 安装新的 CR1025 替换电池时，使电池正极朝上，负极靠近印刷电路板。
4. 按照上述安装指示重新安装 BB 1297 电池板。



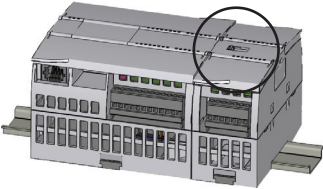
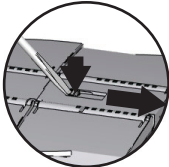
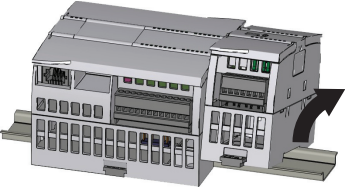
### 3.3.4 安装和拆卸 SM

表格 3-6 安装 SM

任务	步骤	
		<p>在安装 CPU 之后安装 SM。</p> <ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 卸下 CPU 右侧的连接器盖。</li> <li>3. 将螺丝刀插入盖上方的插槽中。</li> <li>4. 将其上方的盖轻轻撬出并卸下盖。收好盖以备再次使用。</li> </ol>
		<p>将 SM 连接到 CPU:</p> <ol style="list-style-type: none"> <li>1. 将 SM 装在 CPU 旁边。</li> <li>2. 将 SM 挂到 DIN 导轨上方。</li> <li>3. 拉出下方的 DIN 导轨卡夹以便将 SM 安装到导轨上。</li> <li>4. 向下转动 CPU 旁的 SM 使其就位并推入下方的卡夹将 SM 锁定到导轨上。</li> </ol>
	<p>伸出总线连接器即为 SM 建立了机械和电气连接。</p> <ol style="list-style-type: none"> <li>1. 将螺丝刀放到 SM 上方的小接头旁。</li> <li>2. 将小接头滑到最左侧，使总线连接器伸到 CPU 中。</li> </ol> <p>要接着信号模块再安装信号模块，请按照相同的步骤操作。</p>	

3.3 安装和拆卸步骤

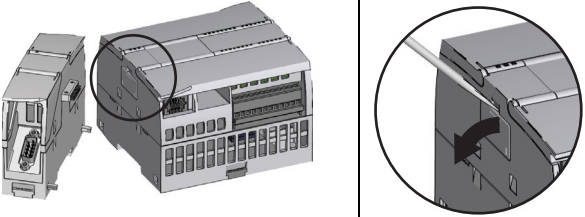
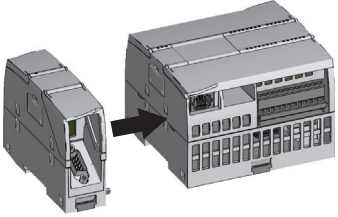
表格 3-7 卸下 SM

任务	步骤
	<p>可以在不卸下 CPU 或其它 SM 处于原位时卸下任何 SM。</p> <ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 将 I/O 连接器和接线从 SM (页 56) 上卸下。</li> <li>3. 缩回总线连接器。                     <ul style="list-style-type: none"> <li>- 将螺丝刀放到 SM 上方的小接头旁。</li> <li>- 向下按使连接器与 CPU 相分离。</li> <li>- 将小接头完全滑到右侧。</li> </ul> </li> </ol>
	<p>如果右侧还有 SM，则对该 SM 重复该步骤。</p>
	<p>卸下 SM:</p> <ol style="list-style-type: none"> <li>1. 拉出下方的 DIN 导轨卡夹从导轨上松开 SM。</li> <li>2. 向上转动 SM 使其脱离导轨。从系统中卸下 SM。</li> <li>3. 如有必要，用盖子盖上 CPU 的总线连接器以避免污染。</li> </ol> <p>要拆除信号模块旁的信号模块，请按照相同的步骤操作。</p>

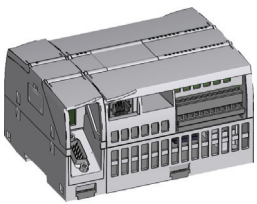
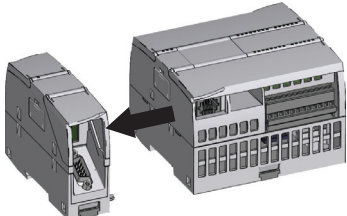
### 3.3.5 安装和拆卸 CM 或 CP

将全部通信模块连接到 CPU 上，然后将该组件作为一个单元来安装，如 安装和拆卸 CPU (页 49) 中所示。

表格 3-8 安装 CM 或 CP

任务	步骤
	<ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 请首先将 CM 连接到 CPU 上，然后再将整个组件作为一个单元安装到 DIN 导轨或面板上。</li> <li>3. 卸下 CPU 左侧的总线盖：                     <ul style="list-style-type: none"> <li>- 将螺丝刀插入总线盖上方的插槽中。</li> <li>- 轻轻撬出上方的盖。</li> </ul> </li> </ol>
	<ol style="list-style-type: none"> <li>4. 卸下总线盖。收好盖以备再次使用。</li> <li>5. 将 CM 或 CP 连接到 CPU 上：                     <ul style="list-style-type: none"> <li>- 使 CM 的总线连接器和接线柱与 CPU 上的孔对齐。</li> <li>- 用力将两个单元压在一起直到接线柱卡入到位。</li> </ul> </li> <li>6. 将 CPU 和 CP 安装到 DIN 导轨或面板上。</li> </ol>

表格 3-9 拆卸 CM 或 CP

任务	步骤
	<p>将 CPU 和 CM 作为一个完整单元从 DIN 导轨或面板上卸下。</p> <ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 拆除 CPU 和 CM 上的 I/O 连接器和所有接线及电缆。</li> <li>3. 对于 DIN 导轨安装，将 CPU 和 CM 上的下部 DIN 导轨卡夹掰到伸出位置。</li> </ol>
	<ol style="list-style-type: none"> <li>4. 从 DIN 导轨或面板上卸下 CPU 和 CM。</li> <li>5. 用力抓住 CPU 和 CM，并将它们分开。</li> </ol>

### 3.3 安装和拆卸步骤

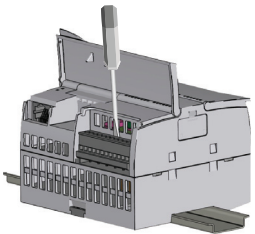
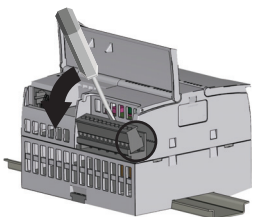
**小心**

请不要使用工具来分离这两个模块，因为这可能会损坏单元。

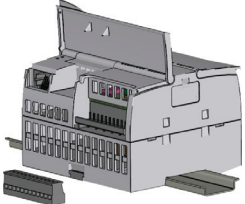
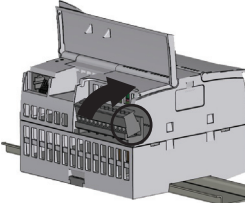
#### 3.3.6 拆卸和重新安装 S7-1200 端子板连接器

CPU、SB 和 SM 模块提供了方便接线的可拆卸连接器。

表格 3- 10 拆卸连接器

任务	步骤
	<p>通过卸下 CPU 的电源并打开连接器上的盖子，准备从系统中拆卸端子板连接器。</p> <ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 查看连接器的顶部并找到可插入螺丝刀头的槽。</li> <li>3. 将螺丝刀插入槽中。</li> <li>4. 轻轻撬起连接器顶部使其与 CPU 分离。连接器从夹紧位置脱离。</li> <li>5. 抓住连接器并将其从 CPU 上卸下。</li> </ol>
	

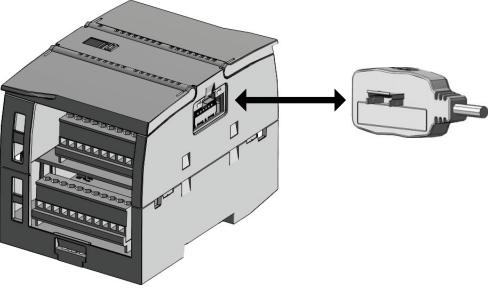
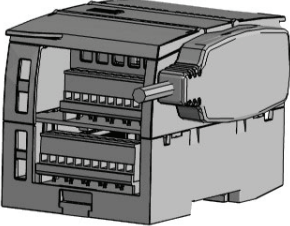
表格 3-11 安装连接器

任务	步骤
	<p>通过断开 CPU 的电源并打开连接器的盖子，准备端子板安装的组件。</p> <ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 使连接器与单元上的插针对齐。</li> <li>3. 将连接器的接线边对准连接器座沿的内侧。</li> <li>4. 用力按下并转动连接器直到卡入到位。</li> </ol>
	<p>仔细检查以确保连接器已正确对齐并完全啮合。</p>

### 3.3.7 安装和卸下扩展电缆

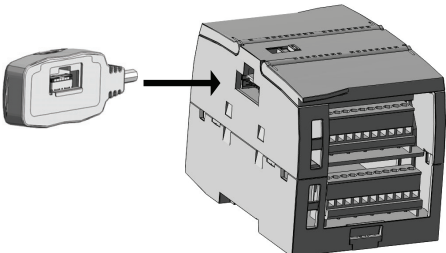
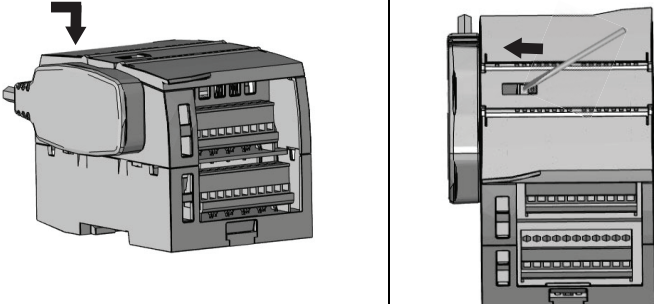
S7-1200 扩展电缆可用来更灵活地组态 S7-1200 系统的布局。每个 CPU 系统只允许使用一条扩展电缆。可以将扩展电缆安装在 CPU 和第一个 SM 之间，或者安装在任意两个 SM 之间。

表格 3-12 安装和卸下扩展电缆的公连接器

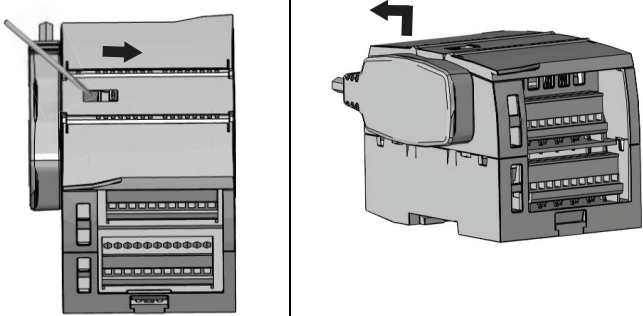
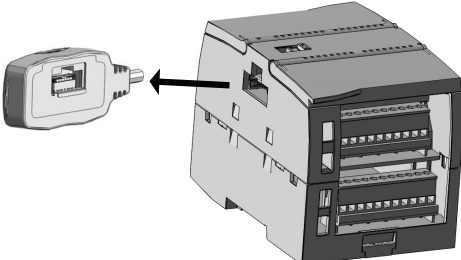
任务	步骤
	<p>要安装公连接器：</p> <ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 将公连接器按压到信号模块或 CPU 右侧的总线连接器中。</li> </ol> <p>要卸下公连接器：</p> <ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 拔出公连接器，使其从信号模块或 CPU 上松开。</li> </ol>
	

3.3 安装和拆卸步骤

表格 3-13 安装扩展电缆的母连接器

任务	步骤
	<ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 将母连接器放到信号模块左侧的总线连接器上。</li> <li>3. 将母连接器的钩伸端滑入总线连接器处的外壳，并轻轻按下，使钩咬合。</li> </ol>
	<ol style="list-style-type: none"> <li>4. 将连接器锁定到位：                     <ul style="list-style-type: none"> <li>- 将螺丝刀放到信号模块上方的小接头旁。</li> <li>- 将小接头完全滑到左侧。</li> </ul> </li> </ol> <p>要使连接器啮合，必须将连接器小接头一直向左滑动。必须将连接器小接头锁定到位。</p>

表格 3-14 卸下扩展电缆的母连接器

任务	步骤
	<ol style="list-style-type: none"> <li>1. 确保 CPU 和所有 S7-1200 设备都与电源断开。</li> <li>2. 解除锁定连接器：                     <ul style="list-style-type: none"> <li>- 将螺丝刀放到信号模块上方的小接头旁。</li> <li>- 轻轻按下连接器，将小接头完全滑到右侧。</li> </ul> </li> <li>3. 轻轻向上提起连接器，使钩伸端分离。</li> </ol>
	<ol style="list-style-type: none"> <li>4. 卸下母连接器。</li> </ol>

### 3.3.8 TS（远程服务）适配器

#### 3.3.8.1 连接远程服务适配器

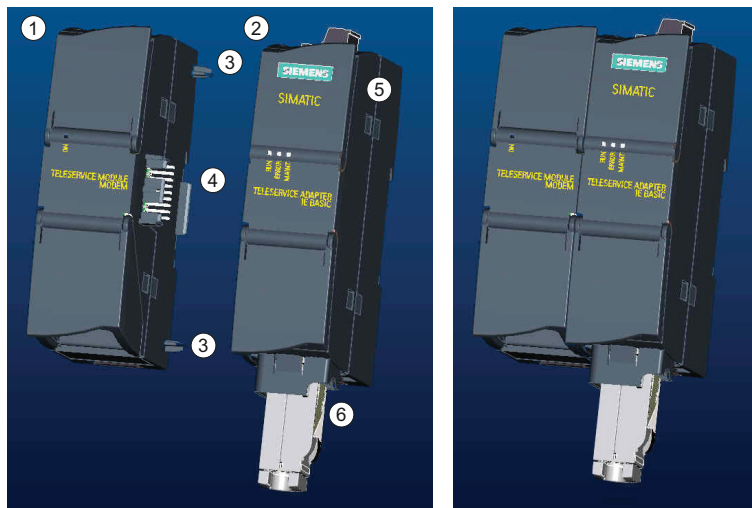
安装 TS（远程）Adapter IE Basic 之前，必须先连接 TS 适配器和一个 TS 模块。

可用的 TS 模块：

- TS 模块 RS232
- TS 模块 Modem
- TS 模块 GSM
- TS 模块 ISDN

**小心**

如果接触 TS 模块的插头连接器 ④ 的触点，则可能损坏 TS 模块。请遵守 ESD 准则，以免静电放电损坏 TS 模块。在连接 TS 模块和 TS 适配器之前，请确保它们都处于空闲状态。



- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>① TS 模块</li> <li>② TS 适配器</li> <li>③ 部件</li> </ul> | <ul style="list-style-type: none"> <li>④ TS 模块的插头连接器</li> <li>⑤ 无法打开</li> <li>⑥ 以太网端口</li> </ul> |
|---|--|



### 3.3 安装和拆卸步骤

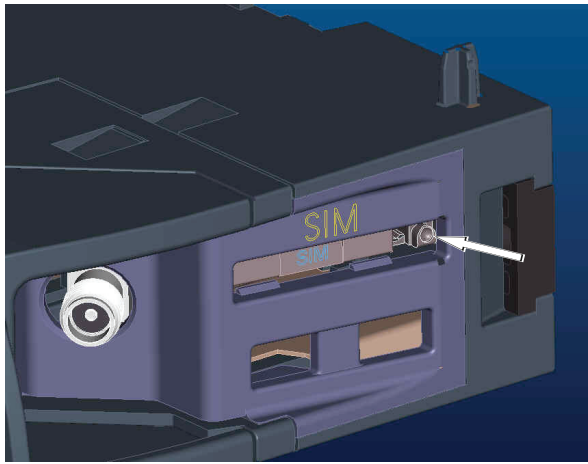
<b>小心</b>
在连接 TS 模块和 TS 适配器基本单元之前，确保触针 ④ 没有弯曲。连接时，确保插头连接器和导销部件位置正确。
只能把一个 TS 模块连接到 TS 适配器中。请勿将 TS 适配器强行连接到不同设备，如 S7-1200 CPU。请勿更改连接器的机械构造，也不要卸下或损坏导销部件。

#### 3.3.8.2 安装 SIM 卡

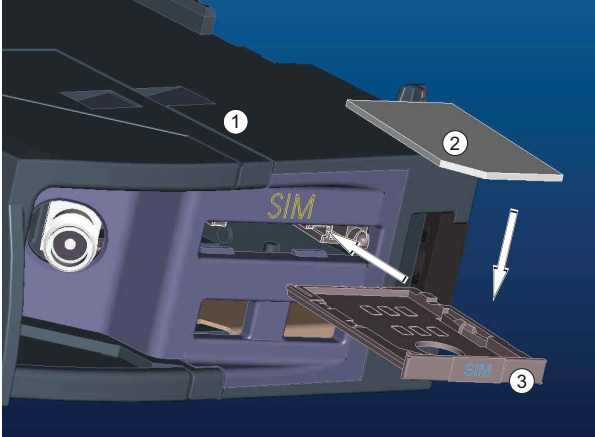
将 SIM 卡插槽置于 TS module GSM 下方。

<b>注意</b>
只有在 TS module GSM 断电情况下才能卸下或插入 SIM 卡。

表格 3- 15 安装 SIM 卡

步骤	任务
使用尖物按压 SIM 卡托上的弹出按钮（箭头方向），取出 SIM 卡托。	



步骤	任务				
<p>如图所示将 SIM 卡放入 SIM 卡托，然后将 SIM 卡托放回卡槽中。</p>				①	TS Module GSM
			②	SIM 卡	
			③	SIM 卡托	

**说明**

确保卡托中的 SIM 卡朝向正确。否则，SIM 卡无法与模块连接，弹出按钮也可能无法弹出卡托。

**3.3.8.3**

**安装 TS 适配器单元**

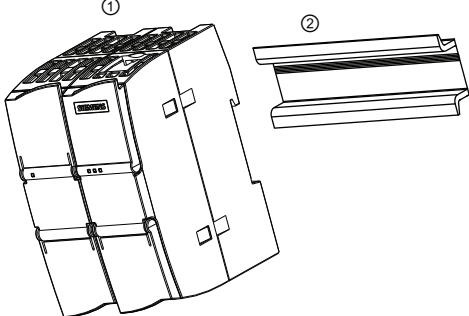
先决条件： 必须已将 TS 适配器和 TS 模块连接在一起，且必须已安装 DIN 导轨。


**说明**

如果垂直安装 TS 单元或在剧烈振动环境中进行安装，TS 模块可能与 TS 适配器断开连接。在 DIN 导轨上使用端盖 8WA1 808 以确保模块保持连接状态。

3.3 安装和拆卸步骤

表格 3- 16 安装和拆卸 TS 适配器

任务	步骤
	<p>安装：</p> <ol style="list-style-type: none"> <li>1. 将连有 TS 模块的 TS 适配器 ① 挂在 DIN 导轨 ② 上。</li> <li>2. 向后旋转单元，直到咬合为止。</li> <li>3. 推入每个模块上的 DIN 导轨卡夹，将每个模块固定在导轨上。</li> </ol> <p>拆卸：</p> <ol style="list-style-type: none"> <li>1. 从 TS 适配器下方卸下模拟电缆和以太网电缆。</li> <li>2. 断开 TS 适配器的电源。</li> <li>3. 用螺丝刀松开两个模块上的导轨卡夹。</li> <li>4. 向上旋转单元，将其从导轨上卸下。</li> </ol>

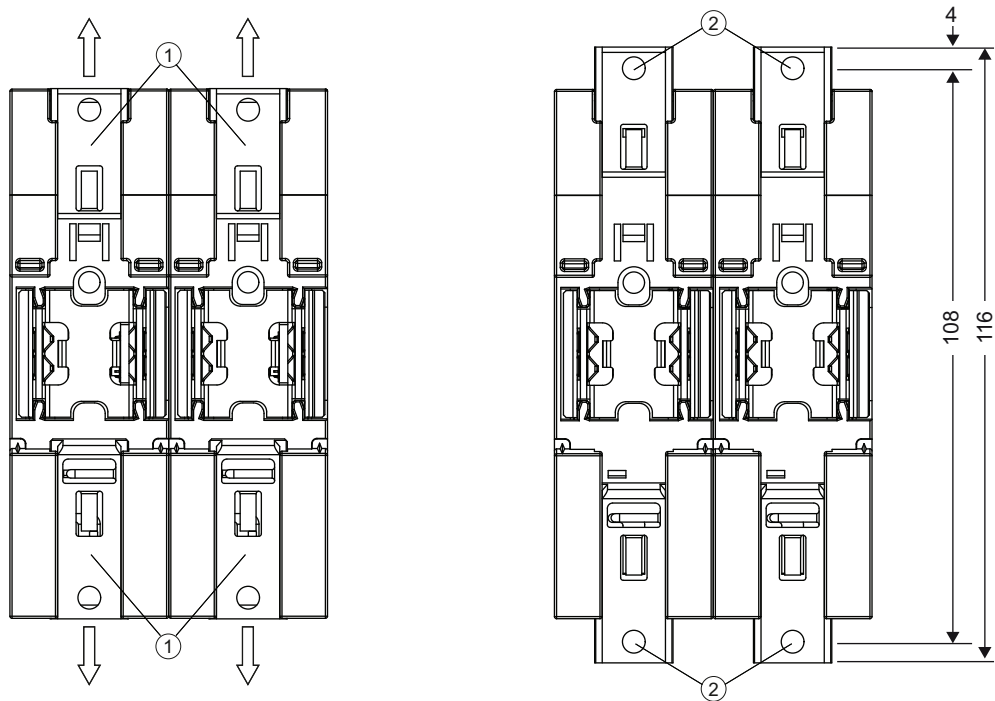
 <b>警告</b>
断开单元的电源之前，先卸下模拟电缆和以太网电缆以断开 TS 适配器的接地连接。

3.3.8.4 将 TS 适配器安装到墙壁上

先决条件： 必须已连接 TS 适配器和 TS 模块。

1. 沿箭头方向将连接滑块 ① 朝 TS 适配器和 TS 模块的后方推，直至其咬合。
2. 用螺钉将 TS 适配器和 TS 模块固定到指定安装墙上标有 ② 的位置。

下图为 TS 适配器的后视图，在两个位置都有连接滑块 ①：



- ① 连接滑块
- ② 墙式安装用的钻孔

### 3.4 接线准则

所有电气设备的正确接地和接线非常重要，因为这有助于确保实现最佳系统运行以及为您的应用和 S7-1200 提供更好的电噪声防护。请参考技术规范 (页 755)以查看 S7-1200 的接线图。

#### 先决条件

在对任何电气设备进行接地或者接线之前，请确保设备的电源已经断开。同时，还要确保已关闭所有相关设备的电源。

### 3.4 接线准则

确保在对 S7-1200 和相关设备接线时遵守所有适用的电气规程。请根据所有适用的国家和地方标准来安装和操作所有设备。请联系当地的管理机构确定哪些规范和标准适用于您的具体情况。



**警告**

安装已上电的 S7-1200 或相关设备或者为这些设备接线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 或相关设备前断开 S7-1200 的电源。

在您规划 S7-1200 系统的接地和接线时，务必考虑安全问题。电子控制设备（如 S7-1200）可能会失灵和导致正在控制或监视的设备出现意外操作。因此，应采取一些独立于 S7-1200 的安全措施以防止可能的人员受伤或设备损坏。



**警告**

控制设备在不安全情况下运行时可能会出现故障，从而导致受控设备的意外操作。这种意外操作可能会导致死亡、人员重伤和/或财产损失。

应使用紧急停止功能、机电超控功能或其它独立于 S7-1200 的冗余安全功能。

### 绝缘准则

S7-1200 交流电源和 I/O 与交流电路的边界经过设计，经验证可以在交流线路电压与低压电路之间实现安全隔离。根据各种适用的标准，这些边界包括双重或加强绝缘，或者基本绝缘加辅助绝缘。跨过这些边界的组件（例如，光耦合器、电容器、变压器和继电器）已通过安全隔离认证。满足这些要求的绝缘边界在 S7-1200 产品数据页中被标识为具有 1500 VAC 或更高的绝缘度。该标识是通过准许的方法采用  $(2U_e + 1000 \text{ VAC})$  或等效电压进行常规工厂测试得来的。S7-1200 的安全隔离边界已通过高达 4242 VDC 的典型试验。

根据 EN 61131-2，集成有交流电源的 S7-1200 的传感器电源输出、通信电路和内部逻辑电路属于 SELV（安全超低电压）电路。

要维持 S7-1200 低压电路的安全特性，到通信端口、模拟电路以及所有 24 V 额定电源和 I/O 电路的外部连接必须由合格的电源供电，该电源必须满足各种标准对 SELV、PELV、二类、限制电压或受限电源的要求。

 **警告**

若使用非隔离或单绝缘电源通过交流线路给低压电路供电，可能会导致本来应当可以安全触摸的电路出现危险电压，例如，通信电路和低压传感器线路。

这种意外的高压可能会引起电击而导致死亡、人员重伤和/或财产损失。

只应当使用合格的高压转低压整流器作为可安全接触的限压电路的供电电源。

### S7-1200 的接地准则

将应用设备接地的最佳方式是确保 S7-1200 和相关设备的所有公共端和接地连接在同一个点接地。该点应该直接连接到系统的大地接地。

所有地线应尽可能地短且应使用大线径，例如，2 mm<sup>2</sup> (14 AWG)。

确定接地点时，应考虑安全接地要求和保护性中断装置的正常运行。

### S7-1200 的接线准则

规划 S7-1200 的接线时，应提供一个可同时切断 S7-1200 CPU 电源、所有输入电路和所有输出电路电力供应的隔离开关。请提供过流保护（例如，熔断器或断路器）以限制电源线中的故障电流。考虑在各输出电路中安装熔断器或其它电流限制器提供额外保护。

为所有可能遭雷电冲击的线路安装合适的浪涌抑制设备。

避免将低压信号线和通信电缆铺设在具有交流线和高能量快速开关直流线的槽中。始终成对布线，中性线或公共线与火线或信号线成对。

使用尽可能短的电线并确保线径适合承载所需电流。CPU 和 SM 连接器接受 2 mm<sup>2</sup> 到 0.3 mm<sup>2</sup> (14 AWG 到 22 AWG) 的线径。SB 连接器接受 1.3 mm<sup>2</sup> 到 0.3 mm<sup>2</sup> (16 AWG 到 22 AWG) 的线径。使用屏蔽线以便最好地防止电噪声。通常在 S7-1200 端将屏蔽层接地能获得最佳效果。

在给通过外部电源供电的输入电路接线时，应在电路中安装过流保护装置。由 S7-1200 的 24 VDC 传感器电源供电的电路不需要外部保护，因为该传感器电源的电流已经受到限制。

所有 S7-1200 模块都有供用户接线的可拆卸连接器。要防止连接器松动，请确保连接器固定牢靠并且导线被牢固地安装到连接器中。为避免损坏连接器，小心不要将螺丝拧得

### 3.4 接线准则

过紧。CPU 和 SM 连接器螺钉的最大扭矩为 0.56 N·m (5 英寸-磅)。SB 连接器螺钉的最大扭矩为 0.33 N·m (3 英寸-磅)。

为了有利于防止安装中出现意外的电流，S7-1200 在某些点提供绝缘边界。在您规划系统的接线时，应考虑这些绝缘边界。有关所提供的绝缘程度和绝缘边界位置的信息，请参见技术规范。不要相信额定值小于 1500 VAC 的绝缘边界是安全边界。

#### 灯负载的使用准则

由于接通浪涌电流大，灯负载会损坏继电器触点。该浪涌电流通常是钨灯稳态电流的 10 到 15 倍。对于在应用期间将进行大量开关操作的灯负载，建议安装可更换的插入式继电器或浪涌限制器。

#### 感性负载的使用准则

应当为感性负载安装抑制电路，限制在关闭控制输出时的电压上升。抑制电路可保护输出，防止关闭感性负载时产生的高压导致其过早损坏。此外，抑制电路还能限制开关感性负载时产生的电噪声。布置一个外部抑制电路使其从电路上跨接在负载两端并且在位置上接近负载，这样对降低电气噪声最有效。

S7-1200 的 DC 输出包括内部抑制电路，该电路足以满足大多数应用中感性负载的要求。由于 S7-1200 继电器输出触点可用于开关直流或交流负载，所以未提供内部保护。

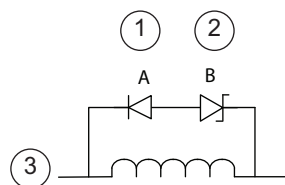
---

#### 说明

给定的抑制电路是否有效取决于实际的应用，必须针对具体应用检验其有效性。务必确保抑制电路中使用的所有元件都适合您的具体应用。

---

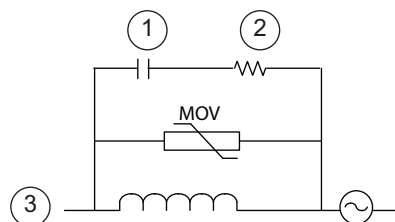
## 用于开关 DC 感性负载的 DC 或继电器输出的典型抑制电路



- ① 1N4001 二极管或同等元件
- ② 8.2 V 稳压二极管（直流输出），  
36 V 稳压二极管（继电器输出）
- ③ 输出点

在大多数应用中，在 DC 感性负载两端增加一个二极管 (A) 就可以了，但如果您的应用要求更快的关闭时间，则建议再增加一个稳压二极管 (B)。请确保正确选择稳压二极管，以适合输出电路中的电流。

## 用于开关 AC 感性负载的继电器输出的典型抑制电路



- ① 0.1  $\mu$ F
- ② 100 到 120  $\Omega$
- ③ 输出点

使用继电器输出开关 115 V/230 VAC 负载时，请在 AC 负载两端并联一个额定值适当的电阻-电容金属氧化物变阻器 (MOV)。请确保 MOV 的工作电压至少比额定线电压高出 20%。

### 3.4 接线准则



## PLC 概念

### 4.1 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的 OB。在事件执行的优先级与排队 (页 76) 中说明了有效 OB 号的范围。
- 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）来保存执行期间的值状态，程序中的其它块可以使用这些值状态。有效的 FC 和 FB 数目范围从 1 到 65535。
- 数据块 (DB) 存储程序块可以使用的数据。有效的 DB 数目范围从 1 到 65535。

用户程序的执行顺序是：从一个或多个在进入 RUN 模式时运行一次的可选启动组织块 (OB) 开始，然后执行一个或多个循环执行的程序循环 OB。OB 也可以与中断事件（可以是标准事件或错误事件）相关联，并在相应的标准或错误事件发生时执行。

功能 (FC) 或功能块 (FB) 是指可从 OB 或其它 FC/FB 调用的程序代码块，可下至以下嵌套深度：

- 16（从程序循环 OB 或启动 OB 开始）
- 4（从延时中断、循环中断、日时钟中断、硬件中断、时间错误中断或诊断错误中断 OB 开始）

FC 不与任何特定数据块 (DB) 相关联，而 FB 与 DB 直接相关并使用 DB 来传送参数以及存储中间值和结果。


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。

每个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后台处理。该周期称为扫描周期或扫描。

4.1 用户程序的执行

只有在通电时才会对模块（SM、SB、BB、CB、CM 或 CP）进行检测和注册。

- 不支持通电时在中央机架中插入或拔出模块（热插拔）。切勿在 CPU 通电时在中央机架中插入或拔出模块。

 <b>警告</b>
<p>在 CPU 通电时在中央机架中插入或拔出模块（SM、SB、BB、CD、CM 或 CP）可能导致不可预知的行为，从而导致设备受损和/或人员受伤。</p> <p>务必确保在中央机架中插入或拔出模块前已切断 CPU 和中央机架的电源。</p>

- 可在 CPU 通电时插入或拔出 SIMATIC 存储卡。但在 CPU 处于 RUN 模式时插入或拔出存储卡会使 CPU 进入 STOP 模式。

<b>小心</b>
<p>在 CPU 处于 RUN 模式时插入或拔出存储卡会使 CPU 进入 STOP 模式，这可能导致受控的设备或过程受损。</p> <p>只要插入或拔出存储卡，CPU 就立即进入 STOP 模式。在插入或拔出存储卡前，务必确保 CPU 当前未控制任何机器或过程。因此务必要为您的应用或过程安装急停电路。</p>

- 如果在 CPU 处于 RUN 模式时在分布式 I/O 机架（PROFINET 或 PROFIBUS）中插入或拔出模块，CPU 将在诊断缓冲区中生成一个条目并保持在 RUN 模式。

在默认组态中，所有本地数字量和模拟量 I/O 点都通过内部存储区（即过程映像）与扫描周期同步更新。过程映像包含物理输入和输出（CPU、信号板和信号模块上的物理 I/O 点）的快照。

CPU 执行以下任务：

- CPU 将过程映像输出区中的输出值写入到物理输出。
- 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。

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。

用户可以指定是否将数字量和模拟量 I/O 点自动更新并存储到过程映像中。如果在设备视图中插入模块，则其数据将默认存储到 CPU 的过程映像中。CPU 在过程映像更新期间自动处理模块和过程映像间的数据交换。要从过程映像自动更新中删除数字量或模拟量点，请在设备配置中选择相应的设备，查看“属性”(Properties) 选项卡，在必要时展开以查找所需 I/O 点，然后选择“IO 地址/硬件标识符”(IO addresses/HW identifier)。然后将

“过程映像:”(Process image:) 对应的条目从“循环 PI”(Cyclic PI) 更改为“---”。要将这些点重新添加到过程映像自动更新中, 请将该选项再更改为“循环 PI”(Cyclic PI)。

可以在指令执行时立即读取物理输入值和立即写入物理输出值。无论 I/O 点是否被组态为存储到过程映像中, 立即读取功能都将访问物理输入的当前状态而不更新过程映像输入区。立即写入物理输出功能将同时更新过程映像输出区(如果相应 I/O 点组态为存储到过程映像中)和物理输出点。如果想要程序不使用过程映像, 直接从物理点立即访问 I/O 数据, 则在 I/O 地址后加后缀“:P”。

CPU 支持 PROFINET 和 PROFIBUS 网络(页 457)上的分布式 I/O。

### 4.1.1 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 不执行程序。您可以下载项目。
- 在 STARTUP 模式下, 执行一次启动 OB(如果存在)。在 STARTUP 模式下不处理任何中断事件。
- 在 RUN 模式下会重复执行程序循环 OB。在 RUN 模式中的任何时刻都可能发生中断事件并对其进行处理。在 RUN 模式下可以下载(页 746)项目中的某些部分。

CPU 支持通过暖启动进入 RUN 模式。暖启动不包括存储器复位。在暖启动时, 所有非保持性系统及用户数据都将被初始化。保留保持性用户数据。

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, 并将装载存储器复制到工作存储器。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 地址值。

---

#### 说明

将一个或多个 DB 从 STEP 7 V11 下载到 S7-1200 V2 CPU 时, 这些 DB 的保持性值和非保持性值将设置为其起始值。下次切换到 RUN 时会执行暖启动, 将所有非保持性值设置为其起始值, 将所有保持性数据设置为其保存值。

若将项目元素(如设备组态、代码块或 DB)从 STEP 7 V10.5 下载到任何 S7-1200 CPU 或从 STEP 7 V11 下载到 S7-1200 V1 CPU(或已组态为 V1 CPU 的 V2 CPU), 则在下次切换到 RUN 模式时会将项目中的所有 DB 复位为起始值。

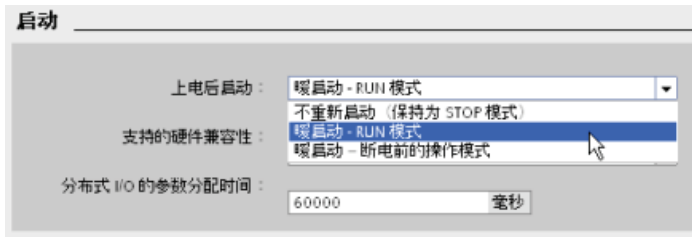
---

可组态 CPU 中“上电后启动”(startup after POWER ON) 设置。该组态项出现在 CPU“设备组态”(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存储器, 并将所有非保持性 DB 的内容重置为装载存储器的初始值。CPU 将保留保持性存储器

4.1 用户程序的执行

和保持性 DB 中的内容，然后切换到相应的工作模式。检测到的某些错误会阻止 CPU 进入 RUN 模式。CPU 支持以下组态选项：

- 不重新启动（保持为 STOP 模式）
- 暖启动 - RUN 模式
- 暖启动 - 断电前的模式



**小心**

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。

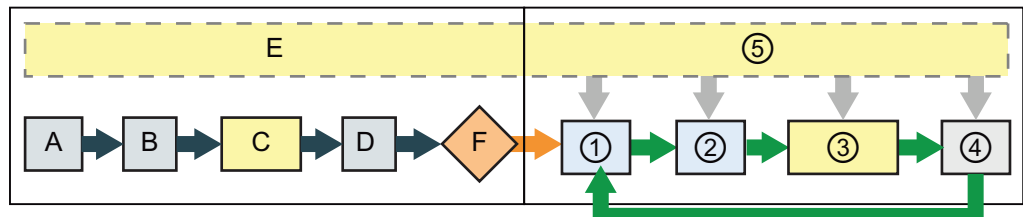
一般情况下，STEP 7 连接中想要独立运行的 CPU 应组态为“暖启动 - RUN”，这样，移除故障条件并重新启动后，CPU 可以返回 RUN 模式。

可以使用编程软件在线工具中的“STOP”或“RUN”命令更改当前工作模式。也可在程序中包含 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。

- 在 STOP 模式下，CPU 处理所有通信请求（如果适用）并执行自诊断。CPU 不执行用户程序，过程映像也不会自动更新。

只有在 CPU 处于 STOP 模式时，才能下载项目。

- 在 STARTUP 和 RUN 模式下，CPU 执行下图所示的任务。



#### STARTUP

- A 清除 I（映像）存储区
- B 使用上一个值或替换值对输出执行初始化
- C 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 Q 存储器到物理输出的写入操作

#### 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

## STARTUP 过程

只要工作模式从 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。通过“启动 OB”中的指令对过程映像输入进行任何的读访问，都只会读取零值，而不是读取当前物理输入值。因此，要在启动模式下读取物理输入的当前状态，必须执行立即读取操作。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。

每个启动 OB 都包含帮助您确定保持性数据和时钟有效性的启动信息。可以在启动 OB 中编写指令，以检查这些启动值，从而采取适当的措施。启动 OB 支持以下启动位置：

表格 4-1 启动 OB 支持的启动位置

输入	数据类型	说明
LostRetentive	Bool	如果保持性数据存储区丢失，该位为真
LostRTC	Bool	如果时钟（实时时钟）丢失，该位为真

## 4.1 用户程序的执行

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。

### 参见

停止扫描循环指令 (页 243)

在线 CPU 的 CPU 操作员面板 (页 735)

### 4.1.2 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。

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。

### 4.1.3 组织块 (OB)

OB 控制用户程序的执行。每个 OB 的 OB 编号必须唯一。200 以下的默认 OB 编号被保留。其它 OB 编号必须大于或等于 200。

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。只有启动事件（例如，诊断中断或时间间隔）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（对应时间错误中断）。

OB 控制以下操作：

- 程序循环 OB 在 CPU 处于 RUN 模式时循环执行。主程序块是程序循环 OB。用户在其中放置控制程序的指令以及调用其它用户块。允许使用多个程序循环 OB，它们按编号顺序执行。OB 1 是默认设置。其它程序循环 OB 必须标识为 OB 200 或更大。
- 启动 OB 在 CPU 的操作模式从 STOP 切换到 RUN 时执行一次，包括处于 RUN 模式时和执行 STOP 到 RUN 切换命令时上电。之后将开始执行主“程序循环”OB。允许有多个启动 OB。OB 100 是默认设置。其它启动 OB 必须为 OB 200 或更大。
- 循环中断 OB 以指定的时间间隔执行。循环中断 OB 将按用户定义的时间间隔（例如，每隔 2 秒）中断循环程序执行。在任何给定的时间，最多可组态总共 4 个时间延时事件和循环事件，每个组态的时间延时和循环事件只允许使用一个 OB。该 OB 必须是 OB 200 或更大。
- 硬件中断 OB 在发生相关硬件事件时执行，包括内置数字输入端的上升沿和下降沿事件以及 HSC 事件。硬件中断 OB 将中断正常的循环程序执行来响应硬件事件信号。可以在硬件配置的属性中定义事件。每个组态的硬件事件只允许对应一个 OB。该 OB 必须是 OB 200 或更大。

## 4.1 用户程序的执行

- 当超出最大循环时间或者发生时间错误事件时，将执行时间错误中断 OB。OB 80 将处理时间错误中断。它将在触发后执行，同时会中断正常循环程序执行或其它所有事件 OB 的执行。以下介绍了触发时间错误中断的事件以及 CPU 对这些事件的响应：
  - 超出最大循环时间：在 CPU 属性中组态最大循环时间。如果 OB 80 不存在，则在超出最大时间后，CPU 将切换为 STOP 模式。
  - 时间错误：如果 OB 80 不存在，则 CPU 将保持为 RUN 模式。如果在 CPU 完成首次执行前，发生时间事件丢失或重复发生、发生队列溢出，或者发生一个事件 OB（时间延时事件、时间事件或者循环中断），则将导致时间错误。

发生任何上述事件都将生成一个描述相应事件的诊断缓冲区条目。无论是否存在 OB 80，都将生成诊断缓冲区条目。

- 诊断错误中断 OB 在检测到和报告诊断错误时执行。如果具有诊断功能的模块发现错误（如果模块已启用诊断错误中断），诊断 OB 将中断正常的循环程序执行。OB 82 是唯一支持诊断错误事件的 OB。如果希望 CPU 在接收到该类型的错误时进入 STOP 模式，可以在 OB 82 中加入 STP 指令（使 CPU 进入 STOP 模式）。如果程序中没有诊断 OB，则 CPU 会忽略该错误（保持 RUN 模式）。

### 4.1.4 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。有些事件在出现硬件触发的变化时发生，例如，输入点上的沿事件或高速计数器事件。还有些事件只有在出现错误时才发生，例如，诊断错误和时间错误事件。事件优先级和队列用于确定事件中断 OB 的处理顺序。

程序循环事件在每个程序循环（扫描）期间发生一次。在程序循环期间，CPU 写入输出、读取输入和执行程序循环 OB。程序循环事件是必需的，并且一直启用。您可以不为程序循环事件选择程序循环 OB，也可选择多个 OB。程序循环事件触发后，将执行编号最小的程序循环 OB（通常是 OB 1）。在程序循环中，会依次（按编号顺序）执行其它程序循环 OB。

用户可通过循环中断事件组态中断 OB 在组态的扫描时间执行。在创建 OB 并将其选为循环中断 OB 时组态初步扫描时间。循环事件此后可中断程序循环并执行循环中断 OB（循环事件的优先级比程序循环事件的优先级高）。

只能将一个循环中断 OB 连接到一个循环事件。

可以为每个循环事件分配相移，从而扫描时间相同的循环中断彼此错开一定的相移量执行。默认相移是 0。要更改初始相移，或更改循环事件的初始扫描时间，请单击项目树中



的循环中断 OB，单击“属性”(Properties)，然后单击“循环中断”(Cyclic interrupt)，并输入新的初始值。还可以用 Query 循环中断 (QRY\_CINT) 和 Set 循环中断 (SET\_CINT) 指令在程序中查询并更改扫描时间和相移。SET\_CINT 指令设置的扫描时间和相移不会在上电循环或切换到 STOP 模式的过程中保持不变；扫描时间和相移值会在上电循环或切换到 STOP 模式后返回到初始值。CPU 共支持四个循环和延时中断事件。

启动事件在从 STOP 切换到 RUN 模式时发生一次，并触发启动 OB 执行。可以为启动事件选择多个 OB。启动 OB 按编号顺序执行。

用户可以通过延时中断事件组态中断 OB 在指定的延时时间过后执行。延时时间使用 SRT\_DINT 指令指定。延时事件将中断程序循环以执行延时中断 OB。只能将一个延时中断 OB 连接到一个延时事件。CPU 支持四个延时事件。

硬件中断事件在硬件有变化时触发，例如，输入点上的上升沿/下降沿事件或者 HSC (High Speed Counter, 高速计数器) 事件。可以为每个硬件中断事件选择一个中断 OB。硬件事件在设备配置中启用。在硬件配置中或在用户程序中使用 ATTACH 指令为事件指定 OB。CPU 支持多个硬件中断事件。具体事件数取决于 CPU 型号和输入点数。

时间和诊断错误中断事件在 CPU 检测到错误时触发。这些事件的优先级比其它中断事件的优先级高，因此可以中断延时、循环和硬件中断事件的执行。对一个时间错误和诊断错误中断事件只能指定一个中断 OB。

### 了解事件执行的优先级与排队

单一来源的未决（排队的）事件数量通过各种事件类型的不同队列加以限制。达到给定事件类型的未决事件限值后，下一个事件将丢失。有关队列溢出的更多信息，请参见后面的“了解时间错误事件”部分。

每个 CPU 事件都具有相关优先级。不能更改 OB 的优先级。通常，会按优先级的顺序处理事件（先处理优先级最高的事件）。优先级相同的事件按“先到先得”的原则进行处理。

表格 4-2 OB 事件

事件	OB 编号	允许的数量	启动事件	OB 优先级
程序循环	OB 1、OB 200 到 OB 65535	1 个程序循环事件 允许多个 OB	<ul style="list-style-type: none"> <li>启动 OB 结束</li> <li>最后一个程序循环 OB 结束</li> </ul>	1
启动	OB 100、OB 200 到 OB 65535	1 个启动事件 <sup>1,2</sup> 允许多个 OB	从 STOP 切换到 RUN	1
Time	OB 200 到 OB	可达 4 个时间事件 <sup>3</sup>	安排延时 OB 事件	3

## 4.1 用户程序的执行

事件	OB 编号	允许的数量	启动事件	OB 优先级
	65535	每个事件 1 个 OB	安排循环 OB 事件	7
过程	OB 200 到 OB 65535	可达 50 个过程事件 <sup>4</sup> 每个事件 1 个 OB	沿： • 上升沿事件：最多 16 个 • 下降沿事件：最多 16 个	5
			对于 HSC： • CV=PV：最多 6 个 • 方向更改：最多 6 个 • 外部复位：最多 6 个	6
诊断错误	OB 82	1 个事件（仅当已加载 OB 82 时）	模块发送错误	9
时间错误	OB 80	1 个事件（仅当已加载 OB 80 时） <sup>5</sup>	• 超出最大循环时间 • 在 CPU 执行完第一个中断之前开始执行第二个中断（循环中断或时间延时中断）	26

- 启动事件和程序循环事件永远不会同时发生，因为在启动事件运行完成之后才会启动程序循环事件（由操作系统控制）。
- 仅诊断错误事件 (OB 82) 会中断启动事件。其它事件会排队等到启动事件完成后进行处理。
- CPU 提供总共 4 个由延时 OB 和循环 OB 共享的时间事件。用户程序中的延时和循环 OB 数量不能超过 4 个。
- 如果使用 DETACH 和 ATTACH 指令，会有多于 50 个过程事件。
- 可以将 CPU 组态为在超出最大扫描循环时间时保持 RUN 模式，也可使用 RE\_TRIGR 指令重设循环时间。但是，如果在同一个扫描循环中第二次超出最大扫描循环时间，CPU 就会进入 STOP 模式。

开始执行优先级为 2 到 25 的 OB 之后，除 OB 80（时间错误事件，优先级为 26）外，出现其它事件并不会中断对该 OB 的处理。所有其它事件将排队等候处理，以允许完成当前 OB 的处理。

### 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 μs。

## 了解时间错误事件

出现几种不同时间错误情况中的任何一种都会引起时间错误事件。所支持的时间错误有以下几种：

- 超出最大循环时间
- 请求的 OB 无法启动
- 发生队列溢出

如果程序循环在指定的最大扫描周期时间内未完成，就会出现超出最大循环时间这种情况。有关最大循环时间情况、如何组态最大扫描周期时间以及如何复位循环定时器的更多信息，请参见“S7-1200 系统手册中的监视循环时间 (页 81)”部分。

如果循环中断、延时中断或日时钟中断请求 OB，但请求的 OB 已经在执行，就会出现请求的 OB 无法启动这种情况。

如果中断的出现频率超过其处理频率，就会出现发生队列溢出这种情况。各种事件类型的未决（排队的）事件数量通过不同的队列加以限制。如果某个事件在相应的队列已满时发生，将生成时间错误事件。

所有时间错误事件都会触发 OB 80（如果存在）的执行。如果用户程序中不包含 OB 80，则由 CPU 的设备配置决定 CPU 对时间错误的响应：

- 默认时间错误（如在 CPU 完成第一个循环中断前开始第二个循环中断）组态是 CPU 保持 RUN 模式。
- 针对超出最大时间的默认组态是 CPU 切换为 STOP 模式。

可以使用 RE\_TRIGR 指令重设最大循环时间。但是，如果在同一程序循环中出现两次“超出最大循环时间”情况且没有复位循环定时器，则无论 OB 80 是否存在，CPU 都将切换到 STOP 模式。请参见“S7-1200 系统手册中的监视循环时间 (页 81)”部分。

OB 80 包含的启动信息可帮助您确定生成时间错误的事件和 OB。可以在 OB 80 中编写指令，以检查这些启动值以及采取适当的措施。

表格 4-3 OB 80 的启动信息

输入	数据类型	说明
fault_id	BYTE	16#01 - 超出最大循环时间 16#02 - 请求的 OB 无法启动 16#07 和 16#09 - 发生队列溢出
csg_OBnr	OB_ANY	出错时正在执行的 OB 的编号
csg_prio	UINT	导致错误的 OB 的优先级

#### 4.1 用户程序的执行

创建新项目时，不存在时间错误中断 OB 80。如果需要，请在树形结构的“程序块”(Program blocks) 下双击“添加新块”(Add new block)，然后依次选择“组织块”(Organization block)、“时间错误中断”(Time error interrupt)，将时间错误中断 OB 80 添加到项目中。

#### 了解诊断错误事件

模拟（本地）、PROFINET 和 PROFIBUS 设备都能够检测并报告诊断错误。发生或清除几种不同诊断错误情况中的任何一种都会引起诊断错误事件。所支持的诊断错误有以下几种：

- 无用户电源
- 超出上限
- 超出下限
- 断路
- 短路

诊断错误事件可触发 OB 82（如果存在）的执行。如果 OB 82 不存在，CPU 将忽略该错误。创建新项目时，不存在诊断错误中断 OB 82。如果需要，请在树形结构的“程序块”(Program blocks) 下双击“添加新块”(Add new block)，然后依次选择“组织块”(Organization block)、“诊断错误中断”(Diagnostic error interrupt)，将诊断错误中断 OB 82 添加到项目中。

---

#### 说明

##### 多通道本地模拟设备（I/O、RTD 和热电偶）的诊断错误

OB 82 诊断错误中断一次只能报告一个通道的诊断错误。

如果多通道设备的两个通道出现错误，则第二个错误只会在以下情况触发 OB 82：第一个通道错误已清除，由第一个错误触发的 OB 82 已执行完毕，并且第二个错误仍然存在。

---

OB 82 包含的启动信息可帮助您确定事件发生原因是错误的出现还是清除所致，以及确定报告错误的设备和通道。可以在 OB 82 中编写指令，以检查这些启动值以及采取适当的措施。

表格 4-4 OB 82 的启动信息

输入	数据类型	说明
IOstate	WORD	设备的 IO 状态： <ul style="list-style-type: none"> <li>• 如果组态正确，则位 0 = 1，如果组态不再正确，则 = 0。</li> <li>• 如果出现错误（如断线），则位 4 = 1。（如果没有错误，则位 4 = 0。）</li> <li>• 如果组态不正确，则位 5 = 1，如果组态再次正确，则 = 0。</li> <li>• 如果出现 I/O 访问错误，则位 6 = 1。有关存在访问错误的 I/O 的硬件标识符，请参见 laddr。（如果没有错误，则位 6 = 0。）</li> </ul>
laddr	HW_ANY	报告错误的设备或功能单元的硬件标识符 <sup>1</sup>
channel	UINT	通道号
multierror	BOOL	如果存在多个错误，参数值为 TRUE

<sup>1</sup> 梯形图输入包含返回错误的设备或功能单元的硬件标识符。硬件标识符是在设备或网络视图中插入组件时自动分配的，它出现在 PLC 变量的“常量”(Constants) 选项卡中。还会自动为硬件标识符分配名称。不能更改这些 PLC 变量的“常量”(Constants) 选项卡中的条目。

### 4.1.5 监视循环时间

循环时间是指 CPU 操作系统在 RUN 模式下执行循环阶段所需的时间。CPU 提供了两种监视循环时间的方法：

- 最大扫描周期时间
- 固定最小扫描周期时间

扫描周期监视在启动事件完成后开始。此功能的组态出现在 CPU“设备配置”(Device Configuration) 的“循环时间”(Cycle time) 下。

CPU 始终监视扫描周期，并在超出最大扫描周期时间时做出响应。如果超出组态的最大扫描周期时间，将生成错误，并按以下两种方法之一对该错误进行处理：

- 如果用户程序不包含 OB 80，则 CPU 会生成错误并进入 STOP 模式。（可以将该 CPU 组态更改为忽略该时间错误并保持 RUN 模式。默认组态是 CPU 进入 STOP 模式。）
- 如果用户程序包含 OB 80，则 CPU 会执行 OB 80

RE\_TRIGR 指令（重新触发循环时间监视）可用来复位记录循环时间的定时器。不过，该指令只有在程序循环 OB 中执行时才起作用；如果在 OB 80 中执行，RE\_TRIGR 指令

## 4.1 用户程序的执行

将被忽略。如果在同一程序循环中两次超出最大扫描周期时间，且期间未执行 RE\_TRIGR 指令，则 CPU 将立即切换到 STOP 模式。如果反复执行 RE\_TRIGR 指令，可能会导致死循环或扫描时间非常长。

通常，扫描周期会尽快执行，当前扫描周期一完成，下一个扫描周期就会开始。视用户程序和通信任务而定，扫描周期的时间段在各次扫描中有所不同。为了消除这种差异，CPU 支持一种可选的固定最小扫描周期时间（也称为固定扫描周期）。如果启用了此可选功能并且固定最小扫描周期时间的单位为 ms，则 CPU 将使完成每次 CPU 扫描的最小循环时间保持在  $\pm 1$  ms 的范围内。

如果 CPU 完成正常扫描周期的时间小于指定的最小循环时间，则 CPU 将用额外的扫描周期时间执行运行诊断和/或处理通信请求。这样，CPU 将始终花费固定的时间量来完成扫描周期。

如果 CPU 在指定的最小循环时间内未完成扫描周期，CPU 将正常完成扫描（包括通信处理），并且不会因超出最小扫描时间而引起任何系统响应。下表定义了循环时间监视功能的值范围和默认值。

表格 4-5 循环时间的范围

循环时间	值范围 (ms)	默认值
最大扫描周期时间 <sup>1</sup>	1 到 6000	150 ms
固定最小扫描周期时间 <sup>2</sup>	1 到最大扫描周期时间	禁用

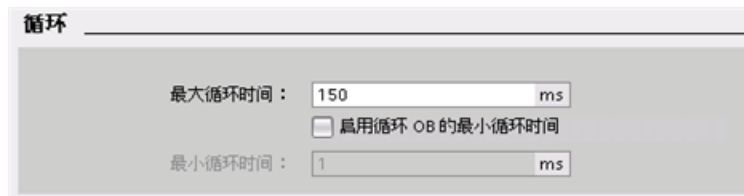
<sup>1</sup> 最大扫描周期时间始终启用。请组态一个 1 ms 到 6000 ms 之间的周期时间。默认值为 150 ms。

<sup>2</sup> 固定最小扫描周期时间是可选的，默认情况下被禁用。必要时，可组态一个 1 ms 到 6000 ms 之间的周期时间。

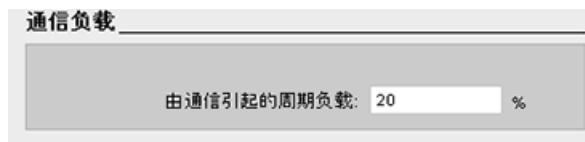
### 组态循环时间和通信负载

利用设备配置中的 CPU 属性可以组态以下参数：

- 循环时间 (Cycle time)：可以输入最大扫描周期时间。也可以输入固定最小扫描周期时间。



- 通信负载 (Communications load)：可以组态一个百分比时间，专门用于通信任务。



有关扫描周期的更多信息，请参见 监视循环时间 (页 81)。

## 4.1.6 CPU 存储器

### 存储器管理

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。存储卡支持的存储空间比 CPU 内置的存储空间更大。
- 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。

要显示当前项目的存储器使用情况，请右键单击相应 CPU（或其中的某个块），然后从上下文菜单中选择“资源”(Resources)。要显示当前 CPU 的存储器使用情况，请双击“在线和诊断”(Online and diagnostics)，展开“诊断”(Diagnostics)，然后选择“存储器”(Memory)。

## 保持性存储器

通过将某些数据标记为保持性数据可以避免在出现电源故障后导致数据丢失。以下数据可以组态为保持性数据：

- **位存储器 (M)：** 可以在 PLC 变量表或分配列表中定义位存储器的具体存储器宽度。保持性位存储器总是从 MB0 开始向上连续贯穿指定的字节数。通过 PLC 变量表或在分配列表中通过单击“保持性”(Retain) 工具栏图标指定该值。输入从 MB0 开始保留的 M 字节个数。
- **函数块 (FB) 的变量：** 如果创建 FB 时选择了“优化”(Optimized)，则该 FB 的接口编辑器中将包含一个“保持性”(Retain) 列。在该列中，可以为每个变量分别选择“保持性”(Retentive)、“非保持性”(Non-Retentive) 或“在 IDB 中设置”(Set in IDB)。在程序编辑器中放置该 FB 时创建的背景 DB 中也将显示该保持性列。在优化的 FB 中，如果在变量的“保持性”(Retain) 选项中选择“在 IDB 中设置”(Set in IDB)（在背景数据块中设置），那么只能更改背景 DB 接口编辑器中某个变量的保持性状态。

如果创建 FB 时选择了“标准 - 与 S7-300/400 兼容”(Standard - compatible with S7-300/400)，则该 FB 的接口编辑器中不会包括“保持性”(Retain) 列。在程序编辑器中插入该 FB 时创建的背景 DB 会显示“保持性”(Retain) 列，并且该列可以编辑。在这种情况下，为任何变量选择“保持性”(Retain) 选项都会导致选择所有变量。同样，为任何变量取消选择该选项都会导致取消选择所有变量。对于组态为“标准 - 与 S7-300/400 兼容”(Standard - compatible with S7-300/400) 的 FB，可以在背景 DB 编辑器中更改保持性状态，但所有变量会同时设置为相同的保持性状态。

创建 FB 后，“标准 - 与 S7-300/400 兼容”(Standard - compatible with S7-300/400) 的选项将无法更改。创建 FB 时，只能选择该选项。要确定现有 FB 是否组态了“优化”(Optimized) 或“标准 - 与 S7-300/400 兼容”(Standard - compatible with S7-300/400)，请在项目树中右键单击该 FB，选择“属性”(Properties)，然后选择“特性”(Attributes)。如果“优化块访问”(Optimized block access) 复选框为选中状态，则说明块已优化。否则为标准块且与 S7-300/400 CPU 兼容。

- **全局数据块的变量：** 在保持性状态分配方面，全局 DB 与 FB 类似。根据块访问设置情况，用户可以定义全局数据块的单个变量或所有变量的保持性状态。
  - 如果在 DB 创建时选择“优化”(Optimized)，则可以设置每个单独变量的保持性状态。
  - 如果在创建 DB 时选择“标准 - 与 S7-300/400 兼容”(Standard - compatible with S7-300/400)，则该保持性状态的设置将适用于该 DB 的所有变量；即变量要么都具有保持性，要么都没有。

总共 10240 个字节的数据可以具有保持性。要了解可用保持性字节数，请在 PLC 变量表或分配列表中单击“保持性”(Retain) 工具栏图标。尽管这里是为 M 存储器指定保持性范



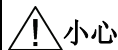
围的地方，但第二个箭头会指示可用于 M 和 DB 的总剩余存储空间。请注意，要保证该值的准确性，必须编译带有保持性变量的所有数据块。

#### 4.1.6.1 系统和时钟存储器

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。）
  - 诊断状态已更改（变量名称：“DiagStatusUpdate”），在 CPU 记录了诊断事件后的一个扫描周期内，该位设置为 1。由于直到首次程序循环 OB 执行结束，CPU 才能置位“诊断图形已更改”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（尤其在与沿指令结合使用时），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

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。

4.1 用户程序的执行

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 4-6 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 4-7 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。

#### 4.1.7 诊断缓冲区

CPU 支持的诊断缓冲区包含有与诊断事件一一对应的条目。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。此日志最多可提供 50 个最近发生的事件。日志填满后，新事件将替换日志中最早的事件。掉电时，将保存事件。

诊断缓冲区中记录以下事件类型：

- 所有系统诊断事件；例如，CPU 错误和模块错误
- CPU 的每次状态切换（每次上电、每次切换到 STOP 模式、每次切换到 RUN 模式）

必须在线访问 诊断缓冲区 (页 736)。从“在线和诊断”(Online & diagnostics) 视图中，在“诊断 > 诊断缓冲区”(Diagnostics > Diagnostics buffer) 下查找诊断缓冲区。

#### 4.1.8 日时钟

CPU 支持日时钟。在 CPU 断电期间，超级电容器提供时钟继续运行所需的电能。超级电容器在 CPU 通电时充电。在 CPU 通电至少 24 小时之后，超级电容器所具有的电量通常足以维持时钟运行 20 天。

STEP 7 将日时钟设置为系统时间，它有一个初始的默认值或者遵循出厂值。若要使用日时钟，必须进行设置。诸如用于诊断缓冲区条目、数据日志文件和数据日志条目的时间戳都是基于系统时间。从在线 CPU 的“在线和诊断”(Online & diagnostics) 视图中的“设置日时钟”功能 (页 734) 下设置日时钟。然后，STEP 7 从您设置的时间中加上或者减去 Windows 操作系统与 UTC（世界协调时间）的偏差来计算系统时间。如果您的 Windows 操作系统的时区和夏令时的设置与您所处的区域相一致，则将日时钟设置为当前的本地时间会产生 UTC 的系统时间。

STEP 7 中包含读写系统时间 (RD\_SYS\_T 和 WR\_SYS\_T)、读取本地时间 (RD\_LOC\_T) 和设置时区 (SET\_TIMEZONE) 的指令 (页 260)。RD\_LOC\_T 指令使用您在 CPU 的一般属性 (页 125) 的“日时钟”(Time of day) 组态中所设置的时区和夏令时偏移

量来计算本地时间。这些设置可以设置您本地时间的时区、选择性地设置夏令时并指定夏令时的开始时间和结束时间。您也可以通过使用 `SET_TIMEZONE` 指令来设定这些设置。

### 4.1.9 组态从 RUN 切换到 STOP 时的输出

可以组态 CPU 处于 STOP 模式时数字量输出和模拟量输出的特性。可以将 CPU、SB 或 SM 的任何输出设置为冻结值或使用替换值：

- 替换特定的输出值（默认）：为 CPU、SB 或 SM 设备的每个输出（通道）分别输入替换值。

数字输出通道的默认替换值为 OFF，而模拟输出通道的默认替换值为 0。

- 冻结输出以保持上一个状态：工作模式从 RUN 切换到 STOP 时，输出将保留当前值。上电后，输出被设置为默认的替换值。

可以在“设备配置”(Device Configuration) 中组态输出的行为。选择相应的设备，然后使用“属性”(Properties) 选项卡组态每个设备的输出。

CPU 从 RUN 切换到 STOP 后，CPU 将保留过程映像，并根据组态写入相应的数字和模拟输出值。

## 4.2 数据存储、存储区、I/O 和寻址

### 4.2.1 访问 S7-1200 的数据

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
- PLC 变量表：在 STEP 7 PLC 变量表中，可以输入特定存储单元的符号名称。这些变量在 STEP 7 程序中为全局变量，并允许用户使用应用程序中有具体含义的名称进行命名。

- **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")。

表格 4-8 存储区

存储区	说明	强制	保持性
I 过程映像输入 I_:P <sup>1</sup> (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 <sup>1</sup> (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)
L 临时存储器	存储块的临时数据，这些数据仅在该块的本地范围内有效	否	否
DB 数据块	数据存储器，同时也是 FB 的参数存储器	否	是 (可选)

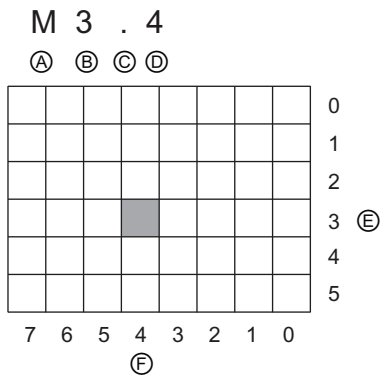
<sup>1</sup> 要立即访问 (读取或写入) 物理输入和物理输出，请在地址或变量后面添加“:P” (例如，I0.3:P、Q1.7:P 或 “Stop:P”)。

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：

- 存储区标识符 (如 I、Q 或 M)
- 要访问的数据的大小 (“B”表示 Byte、“W”表示 Word 或“D”表示 DWord)
- 数据的起始地址 (如字节 3 或字 3)

4.2 数据存储、存储区、I/O 和寻址

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 and 位位置（如 0.0、Q0.1 或 M3.4）。



- 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。

访问 CPU 存储区中的数据

STEP 7 简化了符号编程。通常，既可在 PLC 变量（数据块）中创建变量，也可在 OB、FC 或 FB 顶部的接口中创建变量。这些变量包括名称、数据类型、偏移量和注释。另外，可以在数据块中指定起始值。在编程时，通过在指令参数中输入变量名称，可以使用这些变量。也可以选择指令参数中输入绝对操作数（存储区、大小和偏移量）。以下各部分的实例介绍了如何输入绝对操作数。程序编辑器会自动在绝对操作数前面插入 % 字符。可以在程序编辑器中将视图切换到以下几种视图之一：符号、符号和绝对，或绝对。

**I（过程映像输入）：** CPU 仅在每个扫描周期的循环 OB 执行之前对外围（物理）输入点进行采样，并将这些值写入到输入过程映像。可以按位、字节、字或双字访问输入过程映像。允许对过程映像输入进行读写访问，但过程映像输入通常为只读。

表格 4-9 I 存储器的绝对地址

位	I[字节地址].[位地址]	I0.1
字节、字或双字	I[大小][起始字节地址]	IB4、IW5 或 ID12

通过在地址后面添加“:P”，可以立即读取 CPU、SB 或 SM 的数字和模拟输入。使用 I\_:P 访问与使用 I 访问的区别是，前者直接从被访问点而非输入过程映像获得数据。这种 I\_:P

访问称为“立即读”访问，因为数据是直接从源而非副本获取的，这里的副本是指在上次更新输入过程映像时建立的副本。

因为物理输入点直接从与其连接的现场设备接收值，所以不允许对这些点进行写访问。即，与可读或可写的 I 访问不同的是，I\_:P 访问为只读访问。

I\_:P 访问也仅限于单个 CPU、SB 或 SM 所支持的输入大小（向上取整到最接近的字节）。例如，如果 2 DI/2 DQ SB 的输入被组态为从 I4.0 开始，则可按 I4.0:P 和 I4.1:P 形式或者按 IB4:P 形式访问输入点。不会拒绝 I4.2:P 到 I4.7:P 的访问形式，但没有任何意义，因为这些点未使用。但不允许 IW4:P 和 ID4:P 的访问形式，因为它们超出了与该 SB 相关的字节偏移量。

使用 I\_:P 访问不会影响存储在输入过程映像中的相应值。

表格 4-10 I 存储器的绝对地址（立即）

位	I[字节地址].[位地址]:P	I0.1:P
字节、字或双字	I[大小][起始字节地址]:P	IB4:P、IW5:P 或 ID12:P

**Q（过程映像输出）：** CPU 将存储在输出过程映像中的值复制到物理输出点。可以按位、字节、字或双字访问输出过程映像。过程映像输出允许读访问和写访问。

表格 4-11 Q 存储器的绝对地址

位	Q[字节地址].[位地址]	Q1.1
字节、字或双字	Q[大小][起始字节地址]	QB5、QW10、QD40

通过在地址后面添加“:P”，可以立即写入 CPU、SB 或 SM 的物理数字和模拟输出。使用 Q\_:P 访问与使用 Q 访问的区别是，前者除了将数据写入输出过程映像外还直接将数据写入被访问点（写入两个位置）。这种 Q\_:P 访问有时称为“立即写”访问，因为数据是被直接发送到目标点；而目标点不必等待输出过程映像的下一次更新。

因为物理输出点直接控制与其连接的现场设备，所以不允许对这些点进行读访问。即，与可读或可写的 Q 访问不同的是，Q\_:P 访问为只写访问。

Q\_:P 访问也仅限于单个 CPU、SB 或 SM 所支持的输出大小（向上取整到最接近的字节）。例如，如果 2 DI/2 DQ SB 的输出被组态为从 Q4.0 开始，则可按 Q4.0:P 和 Q4.1:P 形式或者按 QB4:P 形式访问输出点。不会拒绝 Q4.2:P 到 Q4.7:P 的访问形式，但没有任何意义，因为这些点未使用。但不允许 QW4:P 和 QD4:P 的访问形式，因为它们超出了与该 SB 相关的字节偏移量。

4.2 数据存储、存储区、I/O 和寻址

使用 Q<sub>·</sub>:P 访问既影响物理输出，也影响存储在输出过程映像中的相应值。

表格 4- 12 Q 存储器的绝对地址（立即）

位	Q[字节地址].[位地址]:P	Q1.1:P
字节、字或双字	Q[大小][起始字节地址]:P	QB5:P、QW10:P 或 QD40:P

**M（位存储区）：** 针对控制继电器及数据的位存储区（M 存储器）用于存储操作的中间状态或其它控制信息。可以按位、字节、字或双字访问位存储区。M 存储器允许读访问和写访问。

表格 4- 13 M 存储器的绝对地址

位	M[字节地址].[位地址]	M26.7
字节、字或双字	M[大小][起始字节地址]	MB20、MW30、MD50

**临时（临时存储器）：** CPU 根据需要分配临时存储器。CPU 在代码块启动（对于 OB）或被调用（对于 FC 或 FB）时为其分配临时存储器。为代码块分配临时存储器时，可能会重复使用其它 OB、FC 或 FB 先前使用的相同临时存储单元。CPU 在分配临时存储器时不会对其进行初始化，因而临时存储器可能包含任何值。

临时存储器与 M 存储器类似，但有一个主要的区别：M 存储器在“全局”范围内有效，而临时存储器在“局部”范围内有效：

- M 存储器：任何 OB、FC 或 FB 都可以访问 M 存储器中的数据，也就是说这些数据可以全局性地用于用户程序中的所有元素。
- 临时存储器：只有创建或声明了临时存储单元的 OB、FC 或 FB 才可以访问临时存储器中的数据。临时存储单元是局部有效的，并且不会被其它代码块共享，即使在代码块调用其它代码块时也是如此。例如：当 OB 调用 FC 时，FC 无法访问对其进行调用的 OB 的临时存储器。

CPU 为三个 OB 优先级组中的每一个都提供了临时（本地）存储器：

- 16 KB 用于启动和程序循环（包括相关的 FB 和 FC）
- 4 KB 用于标准中断事件（包括 FB 和 FC）
- 4 KB 用于错误中断事件（包括 FB 和 FC）

只能通过符号寻址的方式访问临时存储器。

**DB（数据块）：** DB 存储器用于存储各种类型的数据，其中包括操作的中间状态或 FB 的其它控制信息参数，以及许多指令（如定时器和计数器）所需的数据结构。可以按



位、字节、字或双字访问数据块存储器。读/写数据块允许读访问和写访问。只读数据块只允许读访问。

表格 4-14 DB 存储器的绝对地址

位	DB[数据块编号].DBX[字节地址].[位地址]	DB1.DBX2.3
字节、字或双字	DB[数据块编号].DB [大小][起始字节地址]	DB1.DBB4、 DB10.DBW2、 DB20.DBD8

### 说明

指定绝对地址时，STEP 7 会为此地址加上“%”字符前缀，以指示其为绝对地址。编程时，可以输入带或不带“%”字符的绝对地址（例如 %I0.0 或 I.0）。如果忽略，则 STEP 7 将加上“%”字符。

## 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				
PS485_1	101			CM 1241 (PS485)	6ES7...
▼ PLC_1	1			CPU 1214C D-OD-ODC	6ES7...
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...
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	
DI8 x 24VDC_1	2			SM 1221 DI8 x 24V...	6ES7...

向组态画面添加 CPU 和 I/O 模块时，系统会自动分配 I 地址和 Q 地址。通过在组态画面中选择地址域并键入新编号，可以更改默认寻址设置。

- 无论模块是否使用所有的点，都按 8 点组（1 字节）的方式分配数字量输入和输出。
- 模拟输入和输出按每组 2 点（4 个字节）的方式进行分配。

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。

## 4.3 模拟值的处理

模拟量信号模块可以提供输入信号，或等待表示电压范围或电流范围的输出值。这些范围是  $\pm 10V$ 、 $\pm 5V$ 、 $\pm 2.5V$  或  $0 - 20mA$ 。模块返回的值是整数值，其中，0 到 27648 表示电流的额定范围，-27648 到 27648 表示电压的额定范围。任何该范围之外的值即表示上溢或下溢。有关详细信息，请参见 模拟量输入表示法 (页 845) 和 模拟量输出表示法 (页 846)。

在控制程序中，很可能需要以工程单位使用这些值，例如表示体积、温度、重量或其它数量值。要以工程单位使用模拟量输入，必须首先将模拟值标准化为由 0.0 到 1.0 的实数（浮点）值。然后，必须将其标定为其表示的工程单位的最小值和最大值。对于要转换为模拟量输出值的以工程单位表示的值，应首先将以工程单位表示的值标准化为 0.0 和 1.0 之间的值，然后将其标定为 0 到 27648 之间或 -27648 到 27648 之间（取决于模拟模块的范围）的值。STEP 7 为此提供了 NORM\_X 和 SCALE\_X 指令 (页 226)。还可以使用 CALCULATE 指令 (页 203) 来标定模拟值 (页 34)。

## 4.4 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。

形参指的是指令上标记该指令要使用的数据位置的标识符（例如：ADD 指令的 IN1 输入）。实参指的是包含指令要使用的数据的存储单元（含“%”字符前缀）或常量（例如，%MD400 "Number\_of\_Widgets"）。用户指定的实参的数据类型必须与指令指定的形参所支持的数据类型之一匹配。

指定实参时，必须指定变量（符号）或者绝对（直接）存储器地址。变量将符号名（变量名）与数据类型、存储区、存储器偏移量和注释关联在一起，并且可以在 PLC 变量编辑器或块（OB、FC、FB 和 DB）的接口编辑器中进行创建。如果输入一个没有关联变量的绝对地址，使用的地址大小必须与所支持的数据类型相匹配，而默认变量将在输入时创建。

除 **String** 外，所有数据类型都可在 PLC 变量编辑器和块接口编辑器中使用。String 只能在块接口编辑器中使用。还可以为许多输入参数输入常数值。

- 位和位序列 (页 96): **Bool** (布尔值或位值)、**Byte** (8 位字节值)、**Word** (16 位值)、**DWord** (32 位双字值)
- 整型 (页 97)
  - **USInt** (无符号 8 位整数)、**SInt** (有符号 8 位整数)、
  - **UInt** (无符号 16 位整数)、**Int** (有符号 16 位整数)
  - **UDInt** (无符号 32 位整数)、**DInt** (有符号 32 位整数)
- 浮点实数 (页 97): **Real** (32 位实数值或浮点值)、**LReal** (64 位实数值或浮点值)
- 时间和日期 (页 98): **Time** (32 位 IEC 时间值)、**Date** (16 位日期值)、**TOD** (32 位日时钟值)、**DT** (64 位日期和时间值)
- 字符和字符串 (页 100): **Char** (8 位单字符)、**String** (最长 254 个字符的可变长度字符串)
- 数组 (页 101)
- 数据结构 (页 103): **Struct**
- PLC 数据类型 (页 103)
- 指针 (页 104): **Pointer**、**Any**、**Variant**

尽管以下 BCD 数字格式不能用作数据类型，但它们受转换指令支持。

表格 4- 15 BCD 格式的大小和范围

格式	大小 (位)	数字范围	常量输入示例
BCD16	16	-999 到 999	123, -123
BCD32	32	-9999999 到 9999999	1234567, -1234567

## 4.4.1 Bool、Byte、Word 和 DWord 数据类型

表格 4-16 位和位序列数据类型

数据类型	位大小	数值类型	数值范围	常数示例	地址示例
Bool	1	布尔运算	FALSE 或 TRUE	TRUE、1	I1.0 Q0.1 M50.7 DB1.DBX2.3 Tag_name
		二进制	0 或 1	0, 2#0	
		八进制	8#0 或 8#1	8#1	
		十六进制	16#0 或 16#1	16#1	
Byte	8	二进制	2#0 到 2#11111111	2#00001111	IB2 MB10 DB1.DBB4 Tag_name
		无符号整数	0 到 255	15	
		八进制	8#0 到 8#377	8#17	
		十六进制	B#16#0 到 B#16#FF	B#16#F、16#F	
Word	16	二进制	2#0 到 2#1111111111111111	2#111100001111000 0	MW10 DB1.DBW2 Tag_name
		无符号整数	0 到 65535	61680	
		八进制	8#0 到 8#177777	8#170360	
		十六进制	W#16#0 到 W#16#FFFF、 16#0 到 16#FFFF	W#16#F0F0、 16#F0F0	
DWord	32	二进制	2#0 到 2#1111111111111111 1111111111	2#11110000111111 100001111	MD10 DB1.DBD8 Tag_name
		无符号整数	0 到 4294967295	15793935	
		八进制	8#0 到 8#3777777777	8#74177417	
		十六进制	DW#16#0000_0000 到 DW#16#FFFF_FFFF、 16#0000_0000 到 16#FFFF_FFFF	DW#16#F0FF0F、 16#F0FF0F	

### 4.4.2 整数数据类型

表格 4-17 整型数据类型 (U = 无符号, S = 短, D = 双)

数据类型	位大小	数值范围	常数示例	地址示例
USInt	8	0 到 255	78, 2#01001110	MB0、DB1.DBB4、 Tag_name
SInt	8	-128 到 127	+50, 16#50	
UInt	16	0 到 65,535	65295, 0	MW2、DB1.DBW2、 Tag_name
Int	16	-32,768 到 32,767	30000, +30000	
UDInt	32	0 到 4,294,967,295	4042322160	MD6、DB1.DBD8、 Tag_name
DInt	32	-2,147,483,648 到 2,147,483,647	-2131754992	

### 4.4.3 浮点型实数数据类型

如 ANSI/IEEE 754-1985 标准所述, 实 (或浮点) 数以 32 位单精度数 (Real) 或 64 位双精度数 (LReal) 表示。单精度浮点数的精度最高为 6 位有效数字, 而双精度浮点数的精度最高为 15 位有效数字。在输入浮点常数时, 最多可以指定 6 位 (Real) 或 15 位 (LReal) 有效数字来保持精度。

表格 4-18 浮点型实数数据类型 (L = 长浮点型)

数据类型	位大小	数值范围	常数示例	地址示例
Real	32	-3.402823e+38 到 -1.175 495e-38、 ±0、 +1.175 495e-38 到 +3.402823e+38	123.456, -3.4, 1.0e-5	MD100、DB1.DBD8、 Tag_name
LReal	64	-1.7976931348623158e+308 到 -2.2250738585072014e-308、 ±0、 +2.2250738585072014e-308 到 +1.7976931348623158e+308	12345.123456789e40 、 1.2E+40	DB_name.var_name 规则: <ul style="list-style-type: none"> <li>不支持直接寻址</li> <li>可在 OB、FB 或 FC 块接口数组中进行分配</li> </ul>

## 4.4 数据类型

计算涉及到包含非常大和非常小数字的一长串数值时，计算结果可能不准确。如果数字相差 10 的  $x$  次方，其中  $x > 6$  (Real) 或 15 (LReal)，则会发生上述情况。例如 (Real):  
 $100\,000\,000 + 1 = 100\,000\,000$ .

## 4.4.4 时间和日期数据类型

表格 4-19 时间和日期数据类型

数据类型	大小	范围	常量输入示例
Time	32 位	T#-24d_20h_31m_23s_648ms 到 T#24d_20h_31m_23s_647ms 存储形式: -2,147,483,648 ms 到 +2,147,483,647 ms	T#5m_30s T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630ms 500h10000ms 10d20h30m20s630ms
日期	16 位	D#1990-1-1 到 D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31
Time_of_Day	32 位	TOD#0:0:0.0 到 TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (时间和日期 长型)	12 个字 节	最小: DTL#1970-01-01-00:00:00.0 最大: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

## Time

TIME 数据作为有符号双整数存储，被解释为毫秒。编辑器格式可以使用日期 (d)、小时 (h)、分钟 (m)、秒 (s) 和毫秒 (ms) 信息。

不需要指定全部时间单位。例如，T#5h10s 和 500h 均有效。

所有指定单位值的组合值不能超过以毫秒表示的时间日期类型的上限或下限 (-2,147,483,648 ms 到 +2,147,483,647 ms)。

## 日期

DATE 数据作为无符号整数值存储，被解释为添加到基础日期 1990 年 1 月 1 日的天数，用以获取指定日期。编辑器格式必须指定年、月和日。

## TOD

TOD (TIME\_OF\_DAY) 数据作为无符号双整数值存储，被解释为自指定日期的凌晨算起的毫秒数（凌晨 = 0 ms）。必须指定小时（24 小时/天）、分钟和秒。可以选择指定小数秒格式。

## DTL

DTL（日期和时间长型）数据类型使用 12 个字节的结构保存日期和时间信息。可以在块的临时存储器或者 DB 中定义 DTL 数据。必须在 DB 编辑器的“起始值”(Start value) 列为所有组件输入一个值。

表格 4-20 DTL 的大小和范围

长度 (字节)	格式	值范围	值输入的示例
12	时钟和日历 年-月-日:小时:分钟: 秒.纳秒	最小: DTL#1970-01-01- 00:00:00.0 最大: DTL#2554-12-31- 23:59:59.999 999 999	DTL#2008-12-16- 20:30:20.250

DTL 的每一部分均包含不同的数据类型和值范围。指定值的数据类型必须与相应部分的数据类型相一致。

表格 4-21 DTL 结构的元素

Byte	组件	数据类型	值范围
0	年	UINT	1970 到 2554
1			
2	月	USINT	1 到 12
3	日	USINT	1 到 31
4	工作日 <sup>1</sup>	USINT	1（星期日）到 7（星期六） <sup>1</sup>
5	小时	USINT	0 到 23
6	分	USINT	0 到 59

## 4.4 数据类型

Byte	组件	数据类型	值范围
7	秒	USINT	0 到 59
8	纳秒	UDINT	0 到 999 999 999
9			
10			
11			

<sup>1</sup> 工作日不包括在值条目内。

## 4.4.5 字符和字符串数据类型

表格 4-22 字符和字符串数据类型

数据类型	大小	范围	常量输入示例
Char	8 位	ASCII 字符编码: 16#00 到 16#FF	'A', 't', '@'
String	n+ 2 字节	n = (0 到 254 字符字节)	'ABC'

## Char

Char 数据在存储器中占一个字节，可以存储以 ASCII 格式编码的单个字符。编辑器语法在 ASCII 字符的前面和后面使用一个单引号字符。可以使用可见字符和控制字符。在 String 数据类型的说明中给出了有效的控制字符表。

## String

CPU 支持使用 String 数据类型存储一串单字节字符。String 数据类型包含总字符数（字符串中的字符数）和当前字符数。String 类型提供了多达 256 个字节，用于存储最大总字符数（1 个字节）、当前字符数（1 个字节）以及最多 254 个字符（每个字符占 1 个字节）。

可以对 IN 类型的指令参数使用带单引号的文字串（常量）。例如，'ABC'是由三个字符组成的字符串，可用作 S\_CONV 指令中 IN 参数的输入。还可通过在 OB、FC、FB 和 DB 的块接口编辑器中选择数据类型“字符串”来创建字符串变量。无法在 PLC 变量编辑器中创建字符串。

可以通过在关键字“String”后输入方括号来指定以字节为单位的最大字符串长度（从数据类型下拉列表中选择数据类型“String”之后）。例如，“MyString String[10]”指定 MyString



的最大长度为 10 个字节。如果不包含带有最大长度的方括号，则假定最大长度为 254 字节。

以下示例定义了一个最大字符数为 10 而当前字符数为 3 的 String。这表示该 String 当前包含 3 个单字节字符，但可以扩展到包含最多 10 个单字节字符。

表格 4-23 String 数据类型示例

总字符数	当前字符数	字符 1	字符 2	字符 3	...	字符 10
10	3	'C' (16#43)	'A' (16#41)	'T' (16#54)	...	-
字节 0	字节 1	字节 2	字节 3	字节 4	...	字节 11

ASCII 控制字符可用于 Char 和 String 数据中。下表给出了控制字符语法的示例。

表格 4-24 有效的 ASCII 控制字符

控制字符	ASCII 十六进制值	控制功能	示例
\$L 或 \$l	0A	换行	'\$LText'、'\$0AText'
\$N 或 \$n	0A 和 0D	线路中断 新行显示字符串中的两个字符。	'\$NText'、'\$0A\$0DText'
\$P 或 \$p	0C	换页	'\$PText'、'\$0CText'
\$R 或 \$r	0D	回车 (CR)	'\$RText'、'\$0DText'
\$T 或 \$t	09	制表符	'\$TText'、'\$09Text'
\$\$	24	美元符号	'100\$\$'、'100\$24'
\$'	27	单引号	'\$'Text\$'、'\$27Text\$27'

#### 4.4.6 数组数据类型

##### 数组

可以创建包含多个相同数据类型元素的数组。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。

要在块接口编辑器中创建数组，请为数组命名并选择数据类型“Array [lo .. hi] of type”，然后根据如下说明编辑“lo”、“hi”和“type”：

4.4 数据类型

- lo - 数组的起始（最低）下标
- hi - 数组的结束（最高）下标
- type - 数据类型之一，例如 BOOL、SINT、UDINT

表格 4- 25 ARRAY 数据类型规则

数据类型	数组语法		
ARRAY	Name [index1_min..index1_max, index2_min..index2_max] of <数据类型>		
	<ul style="list-style-type: none"> <li>• 全部数组元素必须是同一数据类型。</li> <li>• 索引可以为负，但下限必须小于或等于上限。</li> <li>• 数组可以是一维到六维数组。</li> <li>• 用逗号分隔多维索引的最小最大值声明。</li> <li>• 不允许使用嵌套数组或数组的数组。</li> <li>• 数组的存储器大小 = （一个元素的大小 * 数组中的元素的总数）</li> </ul>		
	数组索引	有效索引数据类型	数组索引规则
常量或变量	USInt, SInt, UInt, Int, UDInt, DInt	<ul style="list-style-type: none"> <li>• 限值： -32768 到 +32767</li> <li>• 有效： 常量和变量混合</li> <li>• 有效： 常量表达式</li> <li>• 无效： 变量表达式</li> </ul>	

- 示例： 数组声明    ARRAY[1..20] of REAL            一维， 20 个元素
- ARRAY[-5..5] of INT            一维， 11 个元素
- ARRAY[1..2, 3..4] of CHAR        二维， 4 个元素
- 示例： 数组地址    ARRAY1[0]                    ARRAY1 元素 0
- ARRAY2[1,2]                 ARRAY2 元素 [1,2]
- ARRAY3[i,j]                 如果 i =3 且 j=4， 则对 ARRAY3 的元素 [3, 4] 进行寻址

#### 4.4.7 数据结构数据类型

可以用数据类型“**Struct**”来定义包含其它数据类型的数据结构。 **Struct** 数据类型可用来以单个数据单元方式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命名 **Struct** 数据类型并声明内部数据结构。

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。

**Struct** 变量以偶字节地址开始，并占用到下一个字边界的内存。

#### 4.4.8 PLC 数据类型

PLC 数据类型可用来定义可以在程序中多次使用的数据结构。 可以通过打开项目树的“**PLC 数据类型**”分支并双击“添加新数据类型”项来创建 **PLC** 数据类型。 在新创建的 **PLC** 数据类型项上，两次单击可重命名默认名称，双击则会打开 **PLC** 数据类型编辑器。

可使用在数据块编辑器中的相同编辑方法创建自定义 **PLC** 数据类型结构。 为任何必要的数据类型添加新的行，以创建所需数据结构。

如果创建新的 **PLC** 数据类型，则该新 **PLC** 数据类型名称会出现在 **DB** 编辑器和代码块接口编辑器中的数据类型选择下拉列表中。

PLC 数据类型的可能应用：

- 可将 **PLC** 数据类型直接用作代码块接口或数据块中的数据类型。
- **PLC** 数据类型可用作模板，以创建多个使用相同数据结构的全局数据块。

例如，**PLC** 数据类型可能是混合颜色的配方。用户可以将该 **PLC** 数据类型分配给多个数据块。之后，每个数据块都会调节变量，以创建特定颜色。

#### 4.4.9 指针数据类型

指针数据类型（**Pointer**、**Any** 和 **Variant**）可用于 **FB** 和 **FC** 代码块的块接口表中。 可以从块接口数据类型下拉列表中选择指针数据类型。

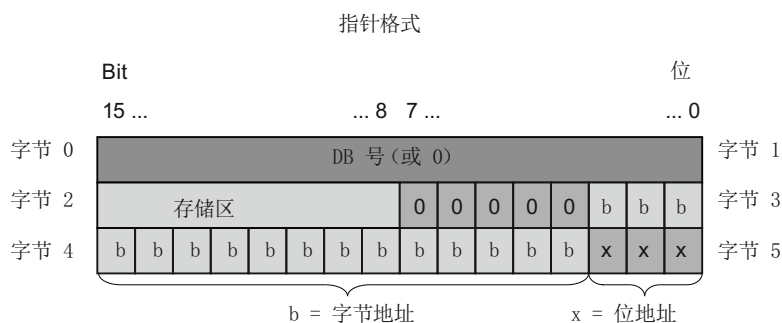
还可以将 **Variant** 数据类型用作指令参数。

4.4 数据类型

4.4.9.1 “Pointer”指针数据类型

数据类型 Pointer 指向特殊变量。它会在存储器中占用 6 个字节（48 位），可能包含以下信息：

- DB 编号或 0（如果该数据未存储在 DB 中）
- CPU 中的存储区
- 变量地址



可以使用指令声明以下三种类型的指针：

- 区域内部的指针：包含变量的地址数据
- 跨区域指针：包含存储区中数据以及变量地址数据
- DB 指针：包含数据块编号以及变量地址

表格 4-26 指针类型：

类型	格式	示例输入
区域内部的指针	P#Byte.Bit	P#20.0
跨区域指针	P#Memory_area_Byte.Bit	P#M20.0
DB 指针	P#Data_block.Data_element	P#DB10.DBX20.0

可以输入没有前缀 (P #) 的 Pointer 类型的参数。您的输入将自动转换为指针格式。

表格 4-27 Pointer 数据中的存储区编码：

十六进制代码	数据类型	说明
b#16#81	I	输入存储区
b#16#82	Q	输出存储区
b#16#83	M	标记存储区

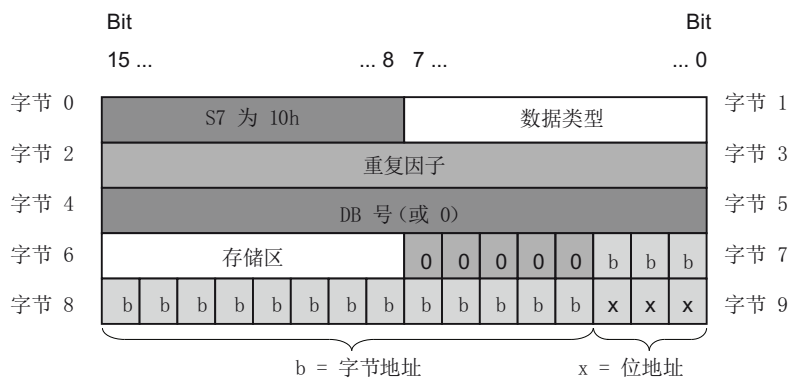
十六进制代码	数据类型	说明
b#16#84	DBX	数据块
b#16#85	DIX	背景数据块
b#16#86	L	本地数据
b#16#87	V	上一本地数据

#### 4.4.9.2 “Any”指针数据类型

指针数据类型 ANY (“Any”) 指向数据区的起始位置，并指定其长度。ANY 指针使用存储器中的 10 个字节，可能包含以下信息：

- 数据类型：数据元素的数据类型
- 重复因子：数据元素数目
- DB 号：存储数据元素的数据块
- 存储区：CPU 中存储数据元素的存储区
- 起始地址：数据的“Byte.Bit”起始地址

下图给出了 ANY 指针的结构：



指针无法检测 ANY 结构。只能将其分配给局部变量。

## 4.4 数据类型

表格 4-28 ANY 指针的格式和示例：

格式	条目示例	说明
P#Data_block.Memory_area Data_address 类型号	P#DB 11.DBX 20.0 INT 10	全局 DB 11 中从 DBB 20.0 开始的 10 个字
P#Memory_area Data_address 类型号	P#M 20.0 BYTE 10	从 MB 20.0 开始的 10 个字节
	P#I 1.0 BOOL 1	输入 I1.0

表格 4-29 ANY 指针中的数据类型编码

十六进制代码	数据类型	说明
b#16#00	Null	Null 指针
b#16#01	Bool	位
b#16#02	Byte	字节, 8 位
b#16#03	Char	8 位字符
b#16#04	Word	16 位字
b#16#05	Int	16 位整数
b#16#37	SInt	8 位整数
b#16#35	UInt	16 位无符号整数
b#16#34	USInt	8 位无符号整数
b#16#06	DWord	32 位双字
b#16#07	DInt	32 位双整数
b#16#36	UDInt	32 位无符号双整数
b#16#08	Real	32 位浮点数
b#16#0B	Time	Time
b#16#13	String	字符串

表格 4- 30 ANY 指针中的存储区编码:

十六进制代码	存储区	说明
b#16#81	I	输入存储区
b#16#82	Q	输出存储区
b#16#83	M	标记存储区
b#16#84	DBX	数据块
b#16#85	DIX	背景数据块
b#16#86	L	本地数据
b#16#87	V	上一本地数据

#### 4.4.9.3 “Variant”指针数据类型

Variant 数据类型可以指向不同数据类型的变量或参数。Variant 指针可以指向结构和单独的结构元素。Variant 指针不会占用存储器的任何空间。

表格 4- 31 Variant 指针的属性

长度 (字节)	表示方式	格式	示例输入
0	符号	操作数	MyTag
		DB_name.Struct_name.element_name	MyDB.Struct1.pressure1
	绝对	操作数	%MW10
		DB_number.Operand Type Length	P#DB10.DBX10.0 INT 12

#### 4.4.10 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)

4.4 数据类型

- "<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。一个字大小的的变量可按位 0 - 15、字节 0 - 2 或字 0 访问。字节大小的变量则可按位 0 - 8 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。

																BYTE															
																WORD															
DWORD																															
x31	x30	x29	x28	x27	x26	x25	x24	x23	x22	x21	x20	x19	x18	x17	x16	x15	x14	x13	x12	x11	x10	x9	x8	x7	x6	x5	x4	x3	x2	x1	x0
b3								b2								b1								b0							
w1																w0															

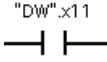

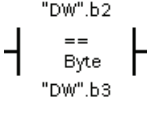

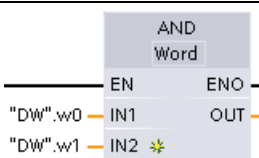
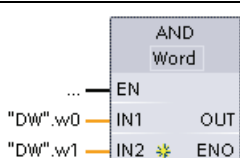
说明

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\_TIMEERROR、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、字节和字片的访问方式：

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

## 参见

SCL (页 157)

### 4.4.11 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。

4.4 数据类型

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
▼ AT	AT "B1" Array [0..7] of Bool
■ AT[0]	Bool
■ AT[1]	Bool
■ AT[2]	Bool
■ AT[3]	Bool
■ AT[4]	Bool
■ AT[5]	Bool
■ AT[6]	Bool
■ AT[7]	Bool


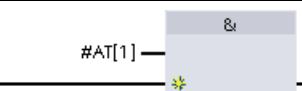
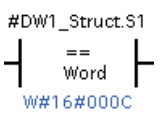
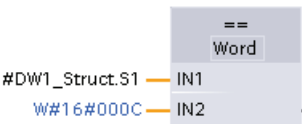

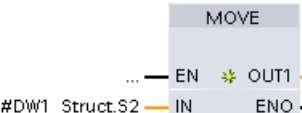
表格 4-32 使用布尔型数组覆盖一个字节

7	6	5	4	3	2	1	0
AT[0]	AT[1]	AT[2]	AT[3]	AT[4]	AT[5]	AT[6]	AT[7]

另一个示例是，使用 Struct 类型覆盖 DWord 变量：

■ DW1	DWord
▼ DW1_Struct	AT "DW1" Struct
■ S1	Word
■ S2	Byte
■ S3	Byte

在程序逻辑中可以直接寻址覆盖的类型：

LAD	FBD	SCL
		<pre>IF #AT[1] THEN ... END_IF;</pre>
		<pre>IF (#DW1_Struct.S1 = W#16#000C) THEN ... END_IF;</pre>
		<pre>out1 := #DW1_Struct.S2;</pre>

## 规则

- 只能覆盖可标准访问的 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。
- 覆盖变量必须在所覆盖的变量之后立即声明，并使用关键字“AT”进行标识。

## 参见

SCL (页 157)

## 4.5 使用存储卡

### 注意

CPU 仅支持预格式化的 SIMATIC 存储卡 (页 912)。

在将程序复制到格式化的存储卡之前，请删除存储卡中以前保存的所有程序。

将存储卡用作传送卡或程序卡。复制到存储卡中的任何程序均包括所有代码块和数据块、所有工艺对象和设备配置。复制的程序不包含强制值。

- 用传送卡 (页 113) 将程序复制到 CPU 的内部装载存储器中，而不使用 STEP 7。插入传送卡后，CPU 首先会擦除内部装载存储器中的用户程序以及任何强制值，然后将程序由传送卡复制到内部装载存储器中。传送过程完成后，必须取出传送卡。

在密码丢失或忘记密码时 (页 120)，可使用空传送卡访问受密码保护的 CPU。插入空传送卡会删除 CPU 内部装载存储器中受密码保护的程序。随后可以将新的程序下载到 CPU 中。

- 将程序卡 (页 116) 用作 CPU 的外部装载存储器。在 CPU 中插入程序卡将擦除 CPU 内部装载存储器的所有内容 (用户程序和所有强制值)。CPU 然后执行外部装载存储器 (程序卡) 中的程序。如果将数据下载到插有程序卡的 CPU，将仅更新外部装载存储器 (程序卡)。

因为 CPU 的内部装载存储器在插入程序卡时已被擦除，所以**必须**将程序卡保留在 CPU 上。如果取出程序卡，CPU 将切换到 STOP 模式。(错误 LED 闪烁，指示程序卡已取出。)

## 4.5 使用存储卡

存储卡上复制的程序包括代码块、数据块、工艺对象和设备配置。存储卡不包含任何强制值。强制值并不属于程序的组成部分，但存储在装载存储器中，也就是存储在 CPU 的内部装载存储器或者外部装载存储器（程序卡）中。如果在 CPU 中插有程序卡，STEP 7 将仅对程序卡上的外部装载存储器应用强制值。

下载 固件更新 (页 118)时，也会使用存储卡。

### 4.5.1 在 CPU 中插入存储卡

#### 小心

静电放电可能会损坏存储卡或 CPU 上的卡槽。

在操控存储卡时，请先接触接地传导垫和/或佩戴接地腕带。将存储卡存放在导电容器内。



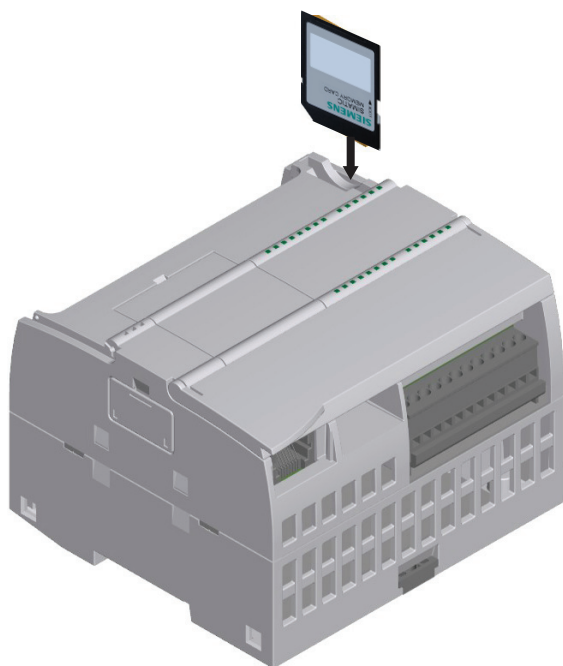
检查以确定存储卡没有写保护。滑动保护开关，使其离开“Lock”位置。

#### 小心

如果将存储卡（无论组态为程序卡还是传送卡）插入到正在运行的 CPU，CPU 将立即进入 STOP 模式，这可能导致受控的设备或过程受损。在插入或拔出存储卡前，务必确保 CPU 当前未控制任何机器或过程。因此务必要为您的应用或过程安装急停电路。

#### 说明

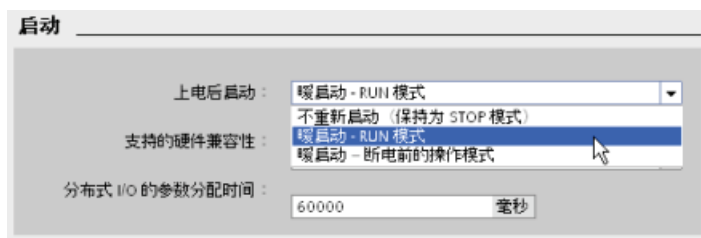
如果在 CPU 处于 STOP 模式时插入存储卡，则诊断缓冲区将显示一条消息提示存储卡评估已经启动。下次 CPU 切换到 RUN 模式、使用 MRES 复位 CPU 存储器或者 CPU 循环上电时，CPU 会评估存储卡。



要插入存储卡，需打开 CPU 顶盖，然后将存储卡插入到插槽中。推弹式连接器可以轻松地插入和取出。  
存储卡要求正确安装。

### 4.5.2 将项目复制到存储卡之前组态 CPU 的启动参数

将程序复制到传送卡或程序卡时，程序中包含了 CPU 的启动参数。将程序复制到传送卡之前，请始终确保组态了 CPU 在循环上电后的工作模式。选择 CPU 是在 STOP 模式、RUN 模式还是上一个模式（通电周期之前）下启动。



### 4.5.3 将存储卡用作“传送”卡

#### 小心

静电放电可能会损坏存储卡或 CPU 上的卡槽。  
在操控存储卡时，请先接触接地传导垫和/或佩戴接地腕带。将存储卡存放在导电容器内。

## 创建传送卡

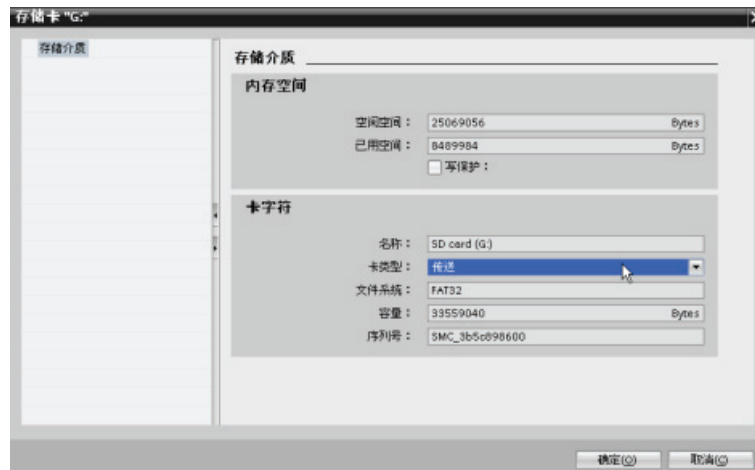
请务必牢记在将程序复制到存储卡之前 组态 CPU 的启动参数 (页 113)。要创建传送卡，请按以下步骤操作：

1. 将空白的 SIMATIC 存储卡插入与计算机相连的 SD 卡读卡器/写卡器中。

如果要重复使用包含用户程序或其它固件更新程序的 SIMATIC 存储卡，那么在重新使用该存储卡之前**必须**删除程序文件。使用 Windows 资源管理器显示存储卡的内容，删除“S7\_JOB.S7S”文件以及任何现有“数据日志”文件夹和目录文件夹（如“SIMATIC.S7S”或“FWUPDATE.S7S”）。

2. 在项目树中（项目视图），展开“SIMATIC 卡读卡器”(SIMATIC Card Reader) 文件夹，然后选择读卡器。
3. 右键单击读卡器中存储卡对应的驱动器盘符，然后从右键快捷菜单中选择“属性”(Properties)，显示“存储卡”(Memory card) 对话框。
4. 在“存储卡”(Memory card) 对话框中，从“卡类型”(Card type) 下拉菜单中选择“传送”(Transfer)。

此时，STEP 7 将创建空传送卡。如果要创建空传送卡以便在 丢失 CPU 密码 (页 120) 后恢复，请从读卡器中移除传送卡。



5. 通过项目树中选择 CPU 设备（例如 PLC\_1 [CPU 1214 DC/DC/DC]），将该 CPU 设备拖动到存储卡来添加程序。（另一种方法是复制 CPU 设备，并将其粘贴到存储卡中。）将 CPU 设备复制到存储卡时，“装载预览”(Load preview) 对话框会打开。

6. 在“装载预览”(Load preview) 对话框中，单击“装载”(Load) 按钮，以将 CPU 设备复制到存储卡。
7. 在对话框显示一条消息指示 CPU 设备（程序）已正确装载时，单击“完成”(Finish) 按钮。

## 使用传送卡



### 警告

**插入存储卡之前，请检查并确认 CPU 当前并未执行任何操作。**

插入存储卡会使 CPU 切换到 STOP 模式，这可能会影响在线操作或机器的运行。意外的过程操作或机器操作可能会导致死亡、人身伤害和/或财产损失。

在插入存储卡前，请务必确保 CPU 处于离线模式且处于安全状态。

要将程序传送到 CPU，请按以下步骤操作：

1. 将传送卡插入 CPU 中 (页 112)。如果 CPU 处于 RUN 模式，它将转至 STOP 模式。维护 (MAINT) LED 闪烁，表示需要对存储卡进行评估。
2. 对 CPU 循环上电以评估存储卡。另一种重启 CPU 的办法是通过 STEP 7 执行 STOP-RUN 切换或存储器复位 (MRES)。
3. 重启并对存储卡进行评估后，CPU 会将程序复制到内部装载存储器。  
RUN/STOP LED 呈绿色和黄色交替闪烁，表示正在复制程序。等到 RUN/STOP LED 为黄色常亮且 MAINT LED 闪烁时，表示复制过程已完成。然后可以取出存储卡。
4. 重启 CPU（通过恢复供电或另一种重启方法），以评估传送到内部装载存储器的新程序。

CPU 随后进入您为项目组态的启动模式（RUN 或 STOP）。

### 说明

将 CPU 设置为 RUN 模式之前，必须先取出传送卡。

#### 4.5.4 将存储卡用作“程序”卡

##### 小心

静电放电可能会损坏存储卡或 CPU 上的卡槽。

在操控存储卡时，请先接触接地传导垫和/或佩戴接地腕带。将存储卡存放在导电容器内。



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。在将程序元素复制到程序卡之前，请删除存储卡中以前保存的所有程序。

#### 创建程序卡

存储卡被用作程序卡时，它就是 CPU 的外部装载存储器。如果取出程序卡，CPU 的内部装载存储器会是空的。

##### 说明

如果在 CPU 中插入空存储卡，然后通过对 CPU 循环上电、执行 STOP 到 RUN 切换或者执行存储器复位 (MRES) 来进行储存卡评估，则 CPU 内部装载存储器中的程序和强制值将复制到存储卡中。（此时存储卡就是程序卡。）复制完成后，将擦除 CPU 内部装载存储器中的程序。CPU 随后进入组态的启动模式 (RUN 或 STOP)。

请务必牢记在将项目复制到程序卡之前 组态 CPU 的启动参数 (页 113)。要创建程序卡，请按以下步骤操作：

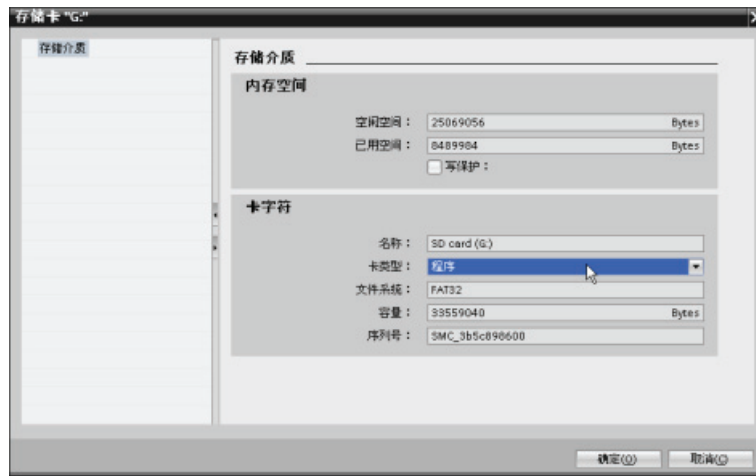
1. 将空白的 SIMATIC 存储卡插入与计算机相连的 SD 卡读卡器/写卡器中。

如果要重复使用包含用户程序或其它固件更新程序的 SIMATIC 存储卡，那么在重新使用该存储卡之前**必须**删除程序文件。使用 Windows 资源管理器显示存储卡的内容，删除“S7\_JOB.S7S”文件以及任何现有“数据日志”文件夹和目录文件夹（如“SIMATIC.S7S”或“FWUPDATE.S7S”）。

2. 在项目树中（项目视图），展开“SIMATIC 卡读卡器”(SIMATIC Card Reader) 文件夹，然后选择读卡器。
3. 右键单击读卡器中存储卡对应的驱动器盘符，然后从右键快捷菜单中选择“属性”(Properties)，显示“存储卡”(Memory card) 对话框。



4. 在“存储卡”(Memory card)对话框中，从下拉菜单中选择“程序”(Program)。



5. 通过在项目树中选择 CPU 设备（例如 PLC\_1 [CPU 1214 DC/DC/DC]），将该 CPU 设备拖动到存储卡来添加程序。（另一种方法是复制 CPU 设备，并将其粘贴到存储卡中。）将 CPU 设备复制到存储卡时，“装载预览”(Load preview)对话框会打开。
6. 在“装载预览”(Load preview)对话框中，单击“装载”(Load)按钮，以将 CPU 设备复制到存储卡。
7. 在对话框显示一条消息指示 CPU 设备（程序）已正确装载时，单击“完成”(Finish)按钮。

### 将程序卡用作 CPU 的装载存储器



#### 警告

**插入存储卡之前，请检查并确认 CPU 当前并未执行任何操作。**

插入存储卡会使 CPU 切换到 STOP 模式，这可能会影响在线操作或机器的运行。意外的过程操作或机器操作可能会导致死亡、人身伤害和/或财产损失。

在插入存储卡前，请务必确保 CPU 处于离线模式且处于安全状态。

要对 CPU 使用程序卡，请按以下步骤操作：

1. 将程序卡插入 CPU。如果 CPU 处于 RUN 模式，则它将切换到 STOP 模式。维护 (MAINT) LED 闪烁，表示需要对存储卡进行评估。
2. 对 CPU 循环上电以评估存储卡。另一种重启 CPU 的办法是通过 STEP 7 执行 STOP-RUN 切换或存储器复位 (MRES)。
3. CPU 重启并对程序卡进行评估后，将擦除其内部装载存储器。

CPU 随后进入您为 CPU 组态的启动模式 (RUN 或 STOP)。

## 4.5 使用存储卡

程序卡必须保留在 CPU 中。取出程序卡将导致 CPU 的内部装载存储器中不会留下任何程序。

### 警告

如果取出程序卡，CPU 将失去外部装载存储器，并生成一条错误消息。CPU 切换到 STOP 模式并且错误 LED 闪烁。

控制设备在不安全情况下运行时可能会出现故障，从而导致受控设备的意外操作。这种意外运行可能会导致死亡、严重的人员伤害和/或设备损坏。

### 4.5.5 固件更新

#### 小心

静电放电可能会损坏存储卡或 CPU 上的卡槽。

在操控存储卡时，请先接触接地传导垫和/或佩戴接地腕带。将存储卡存放在导电容器内。

从客户支持 (<http://www.siemens.com/automation/>) 中下载固件更新程序时，会使用存储卡。在此 Web 站点中，导航至“自动化技术 > 自动化系统 > SIMATIC 工业自动化系统 > PLC > 模块化控制器 SIMATIC S7 > SIMATIC S7-1200”(Automation Technology > Automation Systems > SIMATIC Industrial Automation Systems > PLC > Modular controllers SIMATIC S7 > SIMATIC S7-1200)。然后继续导航至需要更新的特定类型的模块。在“支持”(Support) 下，单击“软件下载”(Software Downloads) 链接以继续。

此外，还可以直接访问 S7-1200 下载网页

(<http://support.automation.siemens.com/WW/view/zh/34612486/133100>)。

#### 说明

不能通过固件更新来更新 S7-1200 CPU V2.2 或 S7-1200 V3.0 之前的版本。

#### 小心

请勿使用 Windows 的格式化程序或其它格式化程序来重新格式化存储卡。

如果使用 Microsoft Windows 的格式化程序重新格式化了 Siemens 存储卡，那么 S7-1200 CPU 将无法再使用该存储卡。

要将固件更新下载到存储卡中，请执行以下步骤：

1. 将空白的 24 MB SIMATIC MC 存储卡插入与计算机相连的 SD 卡读卡器/写卡器中。

您可重复使用包含用户程序或其它固件更新程序的 SIMATIC 存储卡，但您必须删除该存储卡上的一些文件。

**小心**

请勿删除存储卡上的“\_\_LOG\_\_”和“crdinfo.bin”隐藏文件。

存储卡必须包含“\_\_LOG\_\_”和“crdinfo.bin”文件。如果删除了这些文件，将无法在 CPU 中使用该存储卡。

要重复使用存储卡，**必须**在下载固件更新前删除“S7\_JOB.S7S”文件以及任何现有“数据日志”文件夹或任何文件夹（如“SIMATIC.S7S”或“FWUPDATE.S7S”）。但是，**请勿删除**“\_\_LOG\_\_”和“crdinfo.bin”文件。（这些通常是隐藏文件且必不可少。）可以使用 Windows 资源管理器，显示存储卡中的内容并删除相关文件和文件夹。

2. 选择模块所对应的固件更新的自解压文件 (.exe)，然后将其下载到您的计算机上。双击更新程序文件，将该文件的目标路径设置为 SIMATIC 存储卡的根目录，然后开始解压缩。解压缩完成之后，存储卡的根目录中将包含一个“FWUPDATE.S7S”目录和一个“S7\_JOB.S7S”文件。

要安装固件更新，请执行以下步骤：

 **警告**

在安装固件更新之前，请确定 CPU 当前未执行任何进程。

安装固件更新程序时 CPU 将切换到 STOP 模式，这可能会影响在线操作或机器的运行。意外的过程操作或机器操作可能会导致死亡、人身伤害和/或财产损失。

在插入存储卡前，请务必确保 CPU 处于离线模式且处于安全状态。

1. 将存储卡插入 CPU 中。如果 CPU 处于 RUN 模式，则 CPU 将切换到 STOP 模式。维护 (MAINT) LED 闪烁，表示需要对存储卡进行评估。
2. 对 CPU 进行通电以启动固件更新程序。重启 CPU 另一种方式，可通过 STEP 7 执行 STOP-RUN 切换或存储器复位 (MRES)。

**注意**

要完成对该模块的固件更新，必须确保该模块始终接通外部的 24 VDC 电源。

## 4.6 丢失密码后恢复

CPU 重启之后，将开始执行固件更新。RUN/STOP LED 呈绿色和黄色交替闪烁，表示正在复制更新程序。等到 RUN/STOP LED 为黄色常亮且 MAINT LED 闪烁时，表示复制过程已完成。然后必须取出存储卡。

3. 取出存储卡后，再次重新启动 CPU（通过重新通电或其它重新启动方法）以装载新固件程序。

用户程序和硬件配置将不受固件更新的影响。CPU 通电后，CPU 将进入组态后的启动状态。（如果 CPU 的启动模式已组态为“暖启动 - 断电前的模式”，CPU 将处于 STOP 模式，因为 CPU 的前一个状态为 STOP。）

### 说明

#### 更新多个连接到 CPU 的模块

如果硬件配置包含多个与存储卡上单个固件更新文件相对应的模块，则 CPU 将按组态顺序（即按模块在 STEP 7 设备组态中的位置的升序）对所有适用模块（CM、SM、SB）应用更新。

如果已将多个模块的多个固件更新下载到存储卡，则 CPU 将按这些更新下载到存储卡的顺序应用更新。

## 4.6 丢失密码后恢复

如果用户丢失受密码保护的 CPU 的密码，则可使用空传送卡删除受密码保护的程序。空传送卡将擦除 CPU 内部的装载存储器。随后可以将新的用户程序从 STEP 7 下载到 CPU 中。

有关创建和使用空传送卡的信息，请参见 传送卡 (页 113)部分。



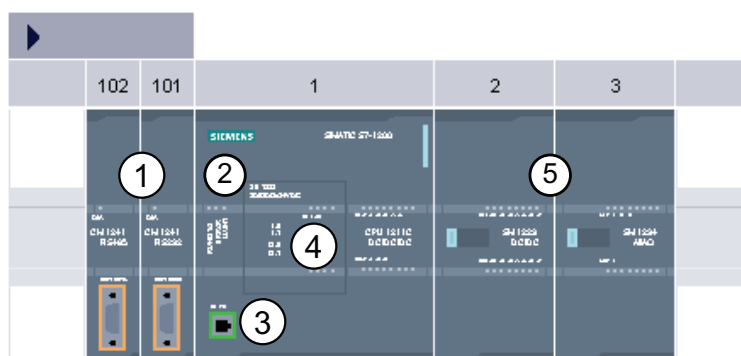
### 警告

如果将传送卡插入正在运行的 CPU 中，CPU 将进入 STOP 模式。控制设备在不安全情况下运行时可能会出现故障，从而导致受控设备的意外操作。这种意外运行可能会导致死亡、严重的人员伤害和/或设备损坏。

将 CPU 设置为 RUN 模式之前，必须先取出传送卡。

## 设备配置

通过向项目中添加 CPU 和其它模块，可以为 PLC 创建设备配置。



- 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 和 CPU 1215C 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)

要创建设备配置, 需向项目中添加设备。

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



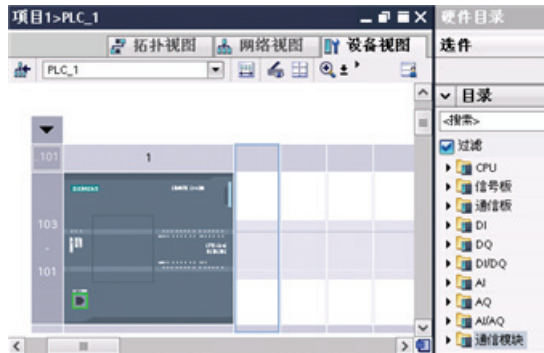
## 5.1 插入 CPU

通过将 CPU 插入到项目中，可创建设备配置。 确保插入了列表中的正确型号和固件版本。 通过从“添加新设备”(Add new device) 对话框中选择 CPU，可创建机架和 CPU。

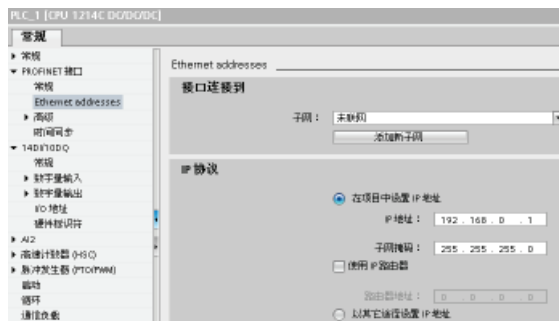
“添加新设备”对话框



硬件配置的设备视图



通过在设备视图中选择 CPU，可在巡视窗口中显示 CPU 属性。



### 说明

CPU 不具有预组态的 IP 地址。 设备配置期间必须为 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。

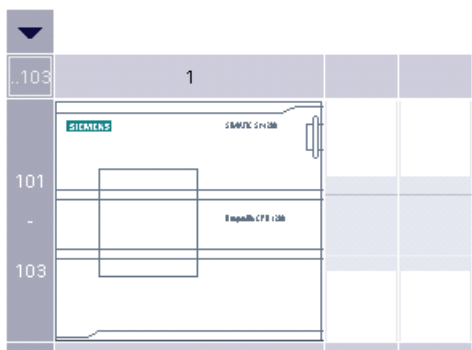
## 5.2 检测未指定的 CPU 的组态



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) 完全跳过设备配置。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。

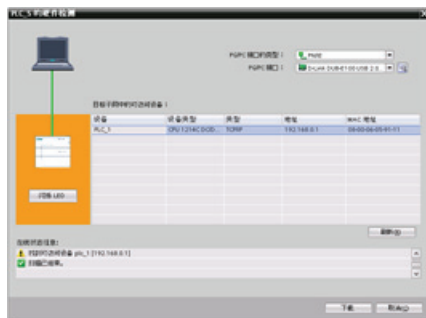
在设备配置编辑器中，选择用于检测所连设备配置的选项。



**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 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 和模块组态参数。





### 5.3 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。

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、BB 或 CB			
CM 或 CP			



## 5.4 组态 CPU 的运行

要组态 CPU 的运行参数，在设备视图（整个 CPU 周围的蓝色轮廓）中选择 CPU，并使用巡视窗口的“属性”(Properties) 选项卡。

要组态输入滤波时间，选择“数字量输入”(Digital Inputs)。数字量输入的默认滤波时间是 6.4 ms。

每一个输入点都有一个适用于所有应用的滤波器组态：过程输入、中断、脉冲捕捉和 HSC 输入。



### 说明

如果 HSC 未组态为使用一个点，则将采用此屏幕中所选的滤波器设置。如果 HSC 组态为使用一个输入点，则该点的滤波器设置将自动设为 800 ns，并且不受此屏幕中组态的影响。

### 警告

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。

表格 5-2 CPU 属性

属性	说明
PROFINET 接口	设置 CPU 的 IP 地址和时间同步
DI、DO 和 AI	组态本地（板载）数字量和模拟量 I/O 的特性（例如，数字量输入滤波时间和对 CPU 停止的数字量输出响应）。

5.4 组态 CPU 的运行

属性	说明
高速计数器 (页 363)和 脉冲发生器 (页 335)	<p>启用并组态高速计数器 (HSC, High-Speed Counter) 以及用于脉冲串运行 (PTO, Pulse-Train Operation) 和脉冲宽度调制 (PWM, Pulse-Width Modulation) 的脉冲发生器</p> <p>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), 会从 Q 存储器中移除相应的输出地址 (Q0.0、Q0.1、Q4.0 和 Q4.1), 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。</p>
启动 (页 71)	<p><b>上电后启动:</b> 选择进行关到开转换之后 CPU 的特性, 如在 STOP 模式下启动或在暖启动后转到 RUN 模式</p> <p><b>支持的硬件兼容性:</b> 组态所有系统组件 (SM、SB、CM、CP 和 CPU) 的替换策略:</p> <ul style="list-style-type: none"> <li>• 允许可接受的替换</li> <li>• 允许任何替换 (默认)</li> </ul> <p>各模块内部均包含基于 I/O 数量、电气兼容性以及其它对应比较点的替换兼容性要求。例如, 16 通道的 SM 是 8 通道 SM 的可接受替换设备, 但 8 通道 SM 不是 16 通道 SM 的可接受替换设备。如果选择“允许可接受的替换”, 则 STEP 7 会实施替换规则; 否则, STEP 7 将允许任何替换。</p> <p><b>分布式 I/O 的参数分配时间:</b> 组态将分布式 I/O 切换到在线状态 所允许的最长时间 (默认值: 60000 ms)。(在启动期间, CM 和 CP 会从 CPU 接收供电和通信参数。该分配时间是连接到 CM 或 CP 的 I/O 切换到在线状态所允许的时间。)</p> <p>无论分配时间是多少, 分布式 I/O 切换为在线状态后, CPU 会立即进入 RUN 模式。如果分布式 I/O 未在这一时间内切换为在线状态, 则 CPU 仍会在没有分布式 I/O 的情况下进入 RUN 模式。</p> <p><b>注:</b> 如果组态使用 CM 1243-5 (PROFIBUS 主站), 不要将此参数设置为低于 15 秒 (15000 ms), 以确保模块切换到在线状态。</p>
周期 (页 81)	定义最大循环时间或固定的最小循环时间
通信负载	分配专门用于通信任务的 CPU 时间百分比
系统和时钟存储器 (页 85)	启用一个字节用于“系统存储器”功能, 并启用一个字节用于“时钟存储器”功能 (其中每个位都按预定义频率打开和关闭)。
Web 服务器 (页 541)	启用和组态 Web 服务器功能。
时间	选择时区并组态夏令时
保护 (页 166)	设置用于访问 CPU 的读/写保护和密码

属性	说明
连接资源 (页 458)	提供可用于 CPU 的通信连接汇总以及已组态的连接数。
地址总览	提供已为 CPU 组态的 I/O 地址的摘要。

## 5.5 组态模块的参数

要组态模块的运行参数，请在设备视图中选择模块，并使用巡视窗口的“属性”(Properties) 选项卡组态模块的参数。

### 组态信号模块 (SM) 或信号板 (SB)



- 数字量 I/O：可组态各个输入用于上升沿检测或下降沿检测（将每个检测分别与一个事件和硬件中断进行关联），并用于在输入过程映像的下一次更新期间进行“脉冲捕捉”（瞬时脉冲之后停留）输出可使用冻结值或替换值。
- 模拟量 I/O：为各个输入组态参数，如测量类型（电压或电流）、范围和平滑化，也可启用下溢或上溢诊断。模拟量输出提供诸如输出类型（电压或电流）之类的参数，也可用于诊断，例如，短路（针对电压输出）或上/下限诊断。请勿在“属性”(Properties) 对话框中组态以工程单位表示的模拟量输入和模拟量输出的范围。必须按照主题“模拟值的处理 (页 94)”的说明在程序逻辑中进行相应处理。
- I/O 诊断地址：组态用于设置模块的输入和输出的起始地址

5.6 组态 CPU 以进行通信

组态通信接口 (CM、CP 或 CB)



根据通信接口的类型组态网络参数。


5.6 组态 CPU 以进行通信

5.6.1 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 5-3 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。	
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。	

操作	结果
释放鼠标按钮以创建网络连接。	

### 5.6.2 组态本地/伙伴连接路径

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。

表格 5-4 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>	

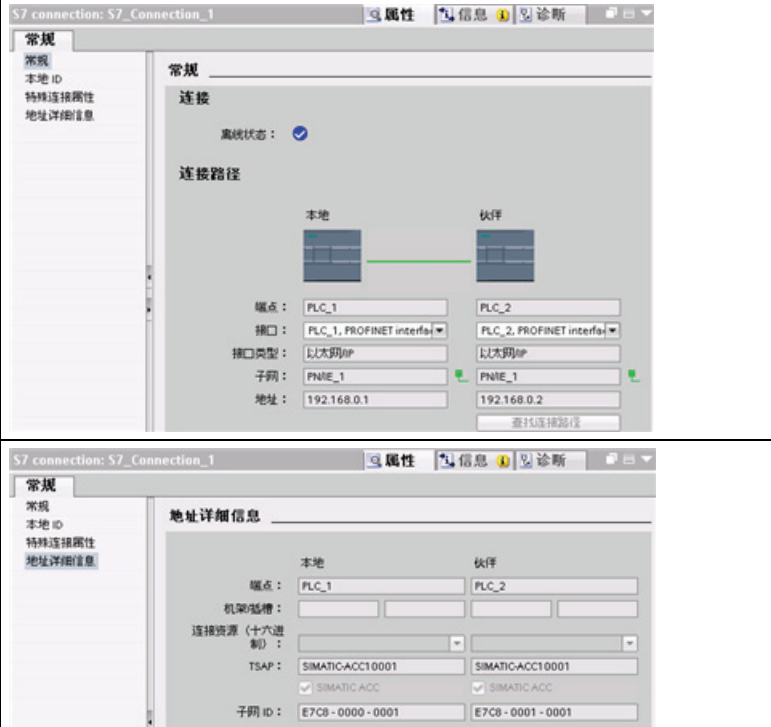
#### 说明

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。

5.6 组态 CPU 以进行通信

表格 5-5 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。可单击“高亮显示：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"> <li>• “常规”(General) (已显示)</li> <li>• “本地 ID”(Local ID)</li> <li>• “特殊连接属性”(Special connection properties)</li> <li>• “地址详细信息”(Address details) (已显示)</li> </ul>	

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 463)，或“S7 通信”部分的“创建 S7 连接”(页 530)。

表格 5-6 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="288 1565 584 1615">端点</td> <td data-bbox="584 1565 1439 1615">分配给伙伴（接收）CPU 的名称</td> </tr> <tr> <td data-bbox="288 1615 584 1664">接口</td> <td data-bbox="584 1615 1439 1664">分配给接口的名称</td> </tr> <tr> <td data-bbox="288 1664 584 1713">子网</td> <td data-bbox="584 1664 1439 1713">分配给子网的名称</td> </tr> <tr> <td data-bbox="288 1713 584 1762">接口类型</td> <td data-bbox="584 1713 1439 1762">仅 S7 通信：接口类型</td> </tr> <tr> <td data-bbox="288 1762 584 1812">连接类型</td> <td data-bbox="584 1762 1439 1812">以太网协议的类型</td> </tr> <tr> <td data-bbox="288 1812 584 1861">连接 ID</td> <td data-bbox="584 1812 1439 1861">ID 号</td> </tr> <tr> <td data-bbox="288 1861 584 1933">连接数据</td> <td data-bbox="584 1861 1439 1933">本地和伙伴 CPU 的数据存储位置</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
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														

参数		定义
	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
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 <sup>1</sup> 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

<sup>1</sup> 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。

### 5.6.3 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 结构来指定。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。

表格 5-7 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"> <li>• 17: TCP（默认）</li> <li>• 18: ISO-on-TCP</li> <li>• 19: UDP</li> </ul>
5	active_est	Bool	连接类型的 ID： <ul style="list-style-type: none"> <li>• TCP 和 ISO-on-TCP： <ul style="list-style-type: none"> <li>– FALSE：被动连接</li> <li>– TRUE：主动连接（默认）</li> </ul> </li> <li>• UDP：FALSE</li> </ul>
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"> <li>• TCP：0（主动，默认值）或 2（被动）</li> <li>• ISO-on-TCP：2 到 16</li> <li>• UDP：2</li> </ul>
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"> <li>• 0: 未指定（参数 rem_staddr 不相关）</li> <li>• 4（默认值）：参数 rem_staddr 中的 IP 地址有效（仅对于 TCP 和 ISO-on-TCP）</li> </ul>



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"> <li>• TCP: 0（被动）或 2（主动，默认值）</li> <li>• ISO-on-TCP: 2 到 16</li> <li>• UDP: 0</li> </ul>
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"> <li>• 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"> <li>– local_tsap_id[1] = 十六进制表示的端口号的高位字节；</li> <li>– local_tsap_id[2] = 十六进制表示的端口号的低位字节；</li> <li>– local_tsap_id[3-16] = 不相关</li> </ul> </li> <li>• ISO-on-TCP: 本地 TSAP-ID: <ul style="list-style-type: none"> <li>– local_tsap_id[1] = B#16#E0;</li> <li>– 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）；</li> <li>– local_tsap_id[3-16] = TSAP 扩展，可选</li> </ul> </li> <li>• UDP: 该参数未使用。</li> </ul> <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： <pre>rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = 不相关</pre>

5.6 组态 CPU 以进行通信

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"> <li>• TCP: 伙伴端口号。范围: 1 到 49151; 推荐值: 2000 到 5000):                             <ul style="list-style-type: none"> <li>- rem_tsap_id[1] = 十六进制表示的端口号的高位字节</li> <li>- rem_tsap_id[2] = 十六进制表示的端口号的低位字节</li> <li>;</li> <li>- rem_tsap_id[3-16] = 不相关</li> </ul> </li> <li>• ISO-on-TCP: 伙伴 TSAP-ID:                             <ul style="list-style-type: none"> <li>- rem_tsap_id[1] = B#16#E0</li> <li>- 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)</li> <li>- rem_tsap_id[3-16] = TSAP 扩展, 可选</li> </ul> </li> <li>• UDP: 该参数未使用。</li> </ul>
56 ... 61	next_staddr	Array [1..6] of Byte	该参数未使用。
62 ... 63	spare	Word	保留: W#16#0000

参见

组态本地/伙伴连接路径 (页 129)

## 5.6.4 分配 Internet 协议 (IP) 地址

### 5.6.4.1 为编程设备和网络设备分配 IP 地址

如果编程设备使用板载适配器卡连接到工厂 LAN（可能是万维网），则 CPU 与编程设备板载适配器卡的 IP 地址网络 ID 和子网掩码必须完全相同。网络 ID 是 IP 地址的第一部分（前三个八位位组）（例如，**211.154.184.16**），它决定用户所在的 IP 网络。子网掩码的值通常为 **255.255.255.0**；然而由于您的计算机处于工厂 LAN 中，子网掩码可能有不同的值（例如，**255.255.254.0**）以设置唯一的子网。子网掩码通过与设备 IP 地址进行数学 AND 运算来确定 IP 子网的边界。

---

#### 说明

在万维网环境下，编程设备、网络设备和 IP 路由器可与全世界通信，但必须分配唯一的 IP 地址以避免与其它网络用户冲突。请联系公司 IT 部门熟悉工厂网络的人员分配 IP 地址。

---

如果编程设备使用连接到独立网络的以太网转 USB 适配器卡，则 CPU 与编程设备的以太网转 USB 适配器卡的 IP 地址网络 ID 和子网掩码必须完全相同。网络 ID 是 IP 地址的第一部分（前三个八位位组）（例如，**211.154.184.16**），它决定用户所在的 IP 网络。子网掩码的值通常为 **255.255.255.0**。子网掩码通过与设备 IP 地址进行数学 AND 运算来确定 IP 子网的边界。

---

#### 说明

当不想将 CPU 连入公司 LAN 时，非常适合使用以太网转 USB 适配器。在首次测试或调试测试期间，这种安排尤其实用。

---

5.6 组态 CPU 以进行通信

表格 5-8 分配以太网地址

编程设备适配器卡	网络类型	Internet 协议 (IP) 地址	子网掩码
板载适配器卡	连接到工厂 LAN (可能是万维网)	CPU 与编程设备板载适配器卡的网络 ID 必须完全相同。 网络 ID 是 IP 地址的第一部分 (前三个八位位组) (例如, <b>211.154.184.16</b> ), 它决定用户所在的 IP 网络。	CPU 和板载适配器卡的子网掩码必须完全相同。 子网掩码的值通常为 <b>255.255.255.0</b> ; 然而由于您的计算机处于工厂 LAN 中, 子网掩码可能有不同的值 (例如, <b>255.255.254.0</b> ) 以设置唯一的子网。子网掩码通过与设备 IP 地址进行数学 AND 运算来确定 IP 子网的边界。
以太网转 USB 适配器卡	连接到独立网络	CPU 与编程设备以太网转 USB 适配器卡的网络 ID 必须完全相同。 网络 ID 是 IP 地址的第一部分 (前三个八位位组) (例如, <b>211.154.184.16</b> ), 它决定用户所在的 IP 网络。	CPU 与编程设备以太网转 USB 适配器卡的子网掩码必须完全相同。 子网掩码的值通常为 <b>255.255.255.0</b> 。子网掩码通过与设备 IP 地址进行数学 AND 运算来确定 IP 子网的边界。

使用桌面上的“网上邻居”(My Network Places) 分配或检查编程设备的 IP 地址

用户可使用以下菜单选项来分配或检查编程设备的 IP 地址:

- (右键单击) “网上邻居”(My Network Places)
- “属性”(Properties)
- (右键单击) “本地连接”(Local Area Connection)
- “属性”(Properties)

在“本地连接属性”(Local Area Connection Properties) 对话框中，在“此连接使用下列项目：”(This connection uses the following items:) 区域向下滚动到“Internet 协议 (TCP/IP)”(Internet Protocol (TCP/IP))。单击“Internet 协议 (TCP/IP)”(Internet Protocol (TCP/IP))，然后单击“属性”(Properties) 按钮。选择“自动获得 IP 地址 (DHCP)”(Obtain an IP address automatically (DHCP)) 或“使用下面的 IP 地址”(Use the following IP address) (可输入静态 IP 地址)。

#### 说明

动态主机配置协议 (DHCP, Dynamic Host Configuration Protocol) 通过 DHCP 服务器在编程设备上电时自动为其分配 IP 地址。

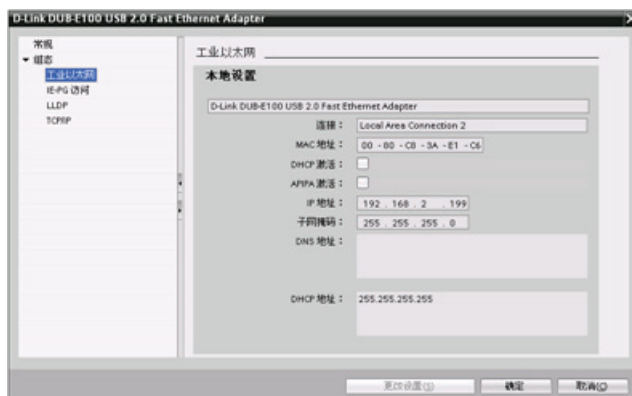
### 5.6.4.2

#### 检查编程设备的 IP 地址

可以使用以下菜单选项来检查编程设备的 MAC 地址和 IP 地址：

1. 在“项目树”(Project tree) 中展开“在线访问”(Online access)。
2. 右键单击所需网络并选择“属性”(Properties)。
3. 在网络对话框中展开“组态”(Configurations) 并选择“工业以太网”(Industrial Ethernet)。

将显示编程设备的 MAC 地址和 IP 地址。



### 5.6.4.3 在线给 CPU 分配 IP 地址

可以在线为网络设备分配 IP 地址。这在进行初始设备配置时尤其有用。

1. 在“项目树”(Project tree) 中, 使用以下菜单选项检查是否还没有给 CPU 分配任何 IP 地址:

- “在线访问”(Online access)
- <设备所在网络的适配器卡>
- “更新可访问的设备”(Update accessible devices)



注意: 如果显示 MAC 地址, 而非 IP 地址, 表示未分配 IP 地址。

2. 在所需可访问设备下双击“在线和诊断”(Online & diagnostics)。

3. 在“在线和诊断”(Online & diagnostics) 对话框中, 选择以下菜单项:

- “功能”(Functions)
- “分配 IP 地址”(Assign IP address)



4. 在“IP 地址”(IP address) 区域输入新 IP 地址, 然后单击“分配 IP 地址”(Assign IP address) 按钮。



5. 在“项目树”(Project tree) 中，使用以下菜单选项检查新的 IP 地址是否已分配给了 CPU:

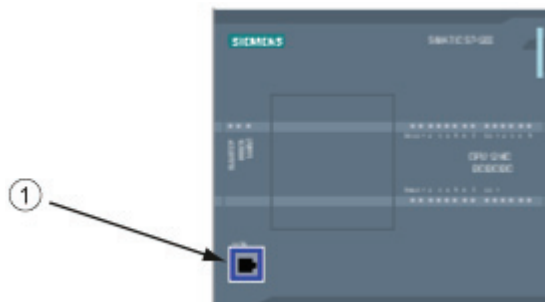
- “在线访问”(Online access)
- <设备所在网络的适配器>
- “更新可访问的设备”(Update accessible devices)



#### 5.6.4.4 为项目中的 CPU 组态 IP 地址

##### 组态 PROFINET 接口

要为 PROFINET 接口组态参数，请选择 CPU 上的绿色 PROFINET 框。巡视窗口中的“属性”(Properties) 选项卡会显示 PROFINET 端口。



① PROFINET 端口

##### 组态 IP 地址

**以太网 (MAC) 地址:** 在 PROFINET 网络中，制造商会为每个设备都分配一个“介质访问控制”地址 (MAC 地址) 以进行标识。MAC 地址由六组数字组成，每组两个十六进制数，这些数字用连字符 (-) 或冒号 (:) 分隔并按传输顺序排列 (例如 01-23-45-67-89-AB 或 01:23:45:67:89:AB)。

**IP 地址:** 每个设备也都必须具有一个 Internet 协议 (IP) 地址。该地址使设备可以在更加复杂的路由网络中传送数据。

每个 IP 地址分为四段，每段占 8 位，并以点分十进制格式表示 (例如，211.154.184.16)。IP 地址的第一部分用于表示网络 ID (您正位于什么网络中?)，地

5.6 组态 CPU 以进行通信

址的第二部分表示主机 ID（对于网络中的每个设备都是唯一的）。IP 地址 192.168.x.y 是一个标准名称，视为未在 Internet 上路由的专用网的一部分。

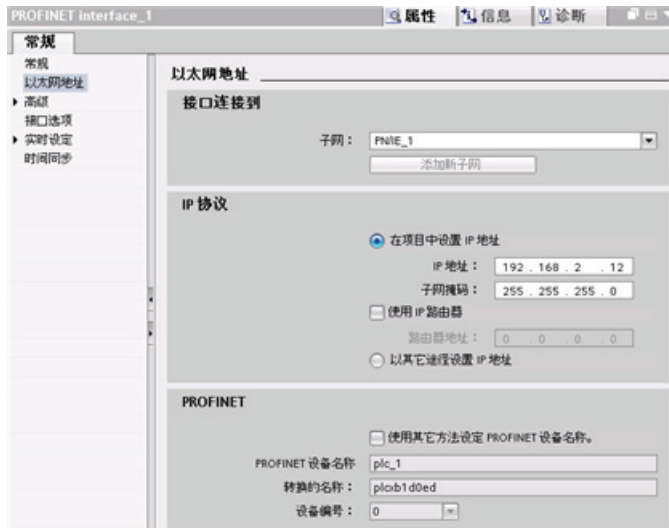
**子网掩码：**子网是已连接的网络设备的逻辑分组。在局域网 (LAN, Local Area Network) 中，子网中的节点往往彼此之间的物理位置相对接近。掩码（称为子网掩码或网络掩码）定义 IP 子网的边界。

子网掩码 255.255.255.0 通常适用于小型本地网络。这就意味着此网络中的所有 IP 地址的前 3 个八位位组应该是相同的，该网络中的各个设备由最后一个八位位组（8 位域）来标识。举例来说，在小型本地网络中，为设备分配子网掩码 255.255.255.0 和 IP 地址 192.168.2.0 到 192.168.2.255。

不同子网间的唯一连接通过路由器实现。如果使用子网，则必须部署 IP 路由器。

**IP 路由器：**路由器是 LAN 之间的链接。通过使用路由器，LAN 中的计算机可向其它任何网络发送消息，这些网络可能还隐含着其它 LAN。如果数据的目的地不在 LAN 内，路由器会将数据转发给可将数据传送到其目的地的另一个网络或网络组。

路由器依靠 IP 地址来传送和接收数据包。



**IP 地址属性：**在“属性”(Properties) 窗口中，选择“以太网地址”(Ethernet addresses) 组态条目。STEP 7 会显示以太网地址组态对话框，该对话框可将软件项目与接收该项目的 CPU 的 IP 地址相关联。



表格 5-9 IP 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"> <li>• 默认情况下“未连接”(Not connected) 提供本地连接。</li> <li>• 网络具有两个或多个设备时，需要子网。</li> </ul>	
IP 协议	IP 地址	为 CPU 分配的 IP 地址
	子网掩码	分配的子网掩码
	使用 IP 路由器	单击该复选框以指示 IP 路由器的使用
	路由器地址	为路由器分配的 IP 地址（如果适用）

**说明**

下载项目时会组态所有 IP 地址。如果 CPU 不具有预组态的 IP 地址，必须将该项目与目标设备的 MAC 地址相关联。如果 CPU 连接到网络中的路由器，则也必须输入路由器的 IP 地址。

“使用其它方法设置 IP 地址”(Set IP address using a different method) 单选按钮允许用户在线更改 IP 地址，或在下载程序之后通过“T\_CONFIG”指令进行更改。这种 IP 地址分配方法仅适用于 CPU。

**警告**

下载启用了“使用其它方法设置 IP 地址”(Set IP address using a different method) 选项的硬件配置后，不能将 CPU 操作模式从 RUN 切换到 STOP，或者从 STOP 切换到 RUN。

在这些情况下，用户设备将保持运行，如果未采取适当的预防措施，则可能导致意外的机器或过程操作，从而导致死亡、严重人身伤害或财产损失。

确保先设置 CPU IP 地址，然后在实际的自动化环境中使用 CPU。这可以通过将 STEP 7 编程包、S7-1200 工具或连接的 HMI 设备与 T\_CONFIG 指令配合使用来完成。

**警告**

在线更改 CPU 的 IP 地址或通过用户程序更改时，可能会出现 PROFINET 网络停止的情况。

如果 CPU 的 IP 地址更改为子网外的 IP 地址，PROFINET 网络将失去通信，并会停止所有数据交换。可将用户设备组态为在这些情况下保持运行。如果未采取适当的预防措施，丢失 PROFINET 通信可能会导致意外的机器或过程操作，从而导致死亡、严重人身伤害或财产损失。

如果必须手动更改 IP 地址，则应确保新 IP 地址在子网范围内。

参见

T\_CONFIG (页 486)

### 5.6.5 测试 PROFINET 网络

在完成组态后，下载项目 (页 169)到 CPU 中。下载项目时会组态所有 IP 地址。



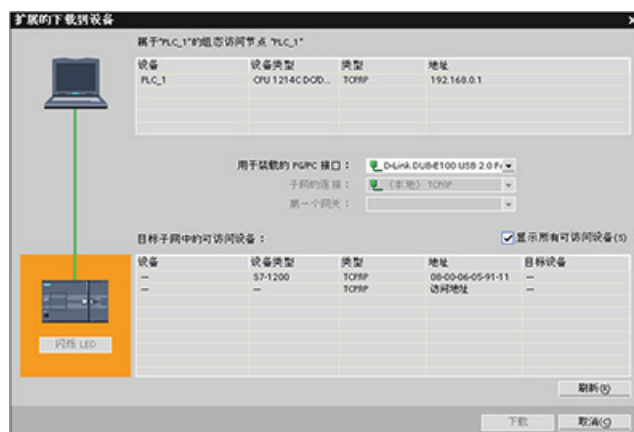
#### 在线为设备分配 IP 地址

S7-1200 CPU 不具有预组态的 IP 地址。必须手动为 CPU 分配 IP 地址：

- 要在线为设备分配 IP 地址，请参见“设备组态：在线为 CPU 分配 IP 地址” (页 138) 了解此逐步操作过程。
- 要在项目中分配 IP 地址，必须在设备配置中组态 IP 地址，保存配置并将其下载到 PLC。请参见“设备组态：为项目中的 CPU 组态 IP 地址” (页 139)，获取更多信息。

### 使用“扩展的下载到设备”(Extended download to device) 对话框测试所连接的网络设备

S7-1200 CPU“下载到设备”(Download to device) 功能及其“扩展的下载到设备”(Extended download to device) 对话框可以显示所有可访问的网络设备，以及是否为所有设备都分配了唯一的 IP 地址。要显示全部可访问和可用的设备以及为其分配的 MAC 地址或 IP 地址，请选中“显示所有可访问设备”(Show all accessible devices) 复选框。



如果所需网络设备不在此列表中，则说明由于某种原因而中断了与该设备的通信。必须检查设备和网络是否有硬件和/或组态错误。

### 5.6.6 查找 CPU 上的以太网 (MAC) 地址

在 PROFINET 网络中，“介质访问控制”地址（MAC 地址）是制造商为了标识网络接口而分配的标识符。MAC 地址通常用制造商的注册标识号进行编码。

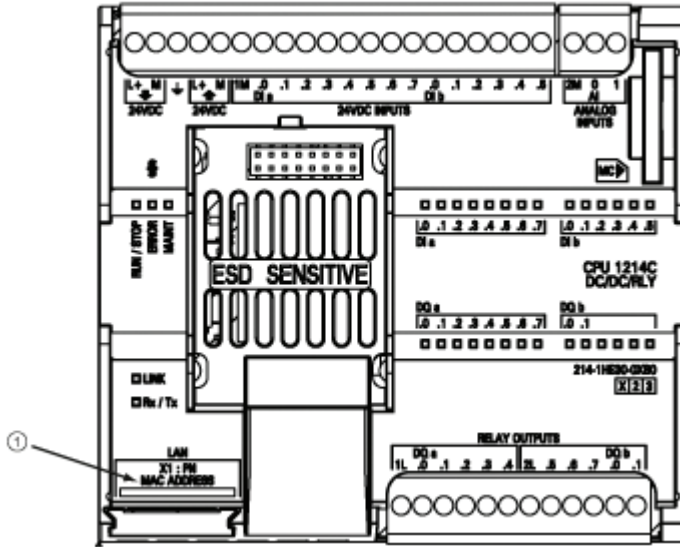
外观良好、按标准 (IEEE 802.3) 格式印制的 MAC 地址由六组数字组成，每组两个十六进制数，这些数字组用连字符 (-) 或冒号 (:) 分隔并按传输顺序排列（例如 01-23-45-67-89-ab 或 01:23:45:67:89:ab）。

#### 说明

每个 CPU 在出厂时都已装载了一个永久、唯一的 MAC 地址。您无法更改 CPU 的 MAC 地址。

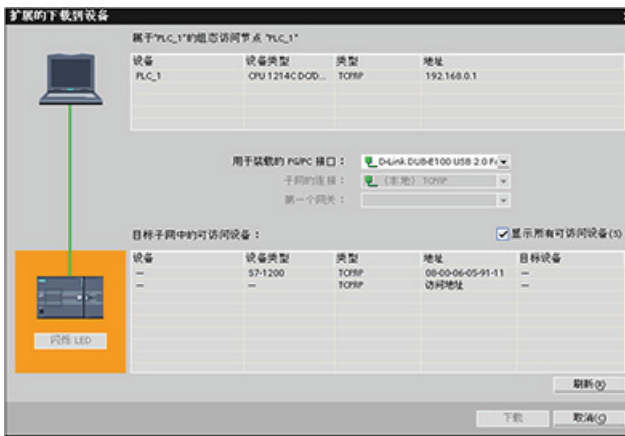
MAC 地址印在 CPU 正面左下角位置。请注意，必须提起下面的门才能看到 MAC 地址信息。

5.6 组态 CPU 以进行通信



① MAC 地址

最初，CPU 没有 IP 地址，只有工厂安装的 MAC 地址。PROFINET 通信要求为所有设备都分配唯一的 IP 地址。



可以使用 CPU“下载到设备”(Download to device) 功能及其“扩展的下载到设备”(Extended download to device) 对话框，显示所有可访问的网络设备以确保已经为所有设备分配了唯一的 IP 地址。此对话框可显示所有可访问和可用的设备以及所分配的 MAC 地址或 IP 地址。在识别缺少所需唯一 IP 地址的设备时，MAC 地址就十分重要。

5.6.7 组态网络时间协议同步

网络时间协议 (NTP, Network Time Protocol) 被广泛用于使计算机系统的时钟与 Internet 时间服务器同步。在 NTP 模式中，CP 按固定时间间隔将日时钟查询（客户机模式中）发送到子网 (LAN) 的 NTP 服务器。根据服务器的响应，来计算最可靠、最准确的时间，并同步工作站的日时钟。

这种模式的优点是可以跨子网同步时间。

需要组态多达四个 NTP 服务器的 IP 地址。更新时间间隔定义各时间查询之间的时间间隔（单位为秒）。时间间隔的值范围在 10 秒到一天之间。

在 NTP 模式下，通常会传送 UTC（Universal Time Coordinated，协调世界时）；与 GMT（Greenwich Mean Time，格林威治标准时间）相对应。

在“属性”(Properties) 窗口中，选择“时间同步”(Time synchronization) 组态条目。STEP 7 显示“时间同步”(Time synchronization) 组态对话框：

#### 说明

下载项目时会组态所有 IP 地址。

表格 5- 10 时间同步参数

参数	定义
启用使用网络时间协议 (NTP) 服务器的日时钟同步 (Enable time-of-day synchronization using Network Time Protocol (NTP) servers)	单击该复选框可启用使用 NTP 服务器的日时钟同步。
服务器 1 (Server 1)	为网络时间服务器 1 分配的 IP 地址
服务器 2 (Server 2)	为网络时间服务器 2 分配的 IP 地址
服务器 3 (Server 3)	为网络时间服务器 3 分配的 IP 地址
服务器 4 (Server 4)	为网络时间服务器 4 分配的 IP 地址
时间同步间隔 (Time synchronization interval)	时间间隔值（秒）

### 5.6.8 PROFINET 设备启动时间、命名和地址分配

PROFINET IO 可以延长系统的启动时间（可组态超时）。设备较多和设备较慢都会影响切换到 RUN 模式的时间。

您可以在您的 S7-1200 PROFINET 网络中选用最多以下数量的 PROFINET IO 设备：

- 在 V3.0 中，您可选用最多 16 个 IO 设备。
- 在 V2.2 中，您可选用最多 8 个 IO 设备。

每个站（或 IO 设备）会在启动时单独启动，这会影响总的 CPU 启动时间。如果将可组态的超时值设定过低，就可能没有足够的总 CPU 启动时间让所有站完成启动。若发生这种情况，会导致假的站错误。

在“启动”(Startup) 下的“CPU 属性”(CPU Properties) 中，可以找到“分布式 I/O 的参数分配时间”(Parameter assignment time for distributed I/O)（超时）。默认的可组态超时为 60,000 ms（1 分钟）；用户可以组态该时间。

### STEP 7 中的 PROFINET 设备命名及寻址

所有 PROFINET 设备**必须**都具有设备名称和 IP 地址。使用 STEP 7 定义设备名称并组态 IP 地址。使用 PROFINET DCP（Discovery and Configuration Protocol，发现和组态协议）将设备名称下载到 IO 设备。

### 系统启动时的 PROFINET 地址分配

控制器会向网络广播设备名称，设备会以其 MAC 地址进行响应。然后，控制器会使用 PROFINET DCP 协议为设备分配 IP 地址：

- 如果 MAC 地址具有已组态的 IP 地址，则相应的站执行启动。
- 如果 MAC 地址不具有组态的 IP 地址，则 STEP 7 会分配项目中组态的地址，之后，相应的站会执行启动。
- 如果这一过程出现问题，则会产生站错误，且不会进行启动。这种情况会导致超出可组态的超时值。

## 编程概念

### 6.1 设计 PLC 系统的指南

设计 PLC 系统时，可从若干方法和标准中进行选择。下列常规指南可应用到许多设计项目中。当然，必须遵守您自己公司程序的指令、自身培训以及当地已被接受的实践。

表格 6-1 设计 PLC 系统的指南

建议步骤	任务
对过程或机器进行分区	将过程或机器划分为彼此独立的部分。这些分区会确定控制器之间的边界，并影响功能描述规范和资源的分配。
创建功能规范	写下过程或机器的每一部分（如 I/O 点）的操作说明、操作的功能描述、在允许进行每个执行器（如螺线管、电机或驱动器）的操作之前必须实现的状态、操作员界面的描述以及过程或机器其它部分的任何接口。
设计安全电路	<p>出于安全考虑，标识任何可能需要硬接线逻辑的设备。请记住，控制设备在不安全方式下可能会出现故障，可能会造成意外启动或机械运转变化。其中意外或错误的机械运转可能会导致人员的身体伤害或重大的财产损失，请考虑实施机电替代装置（其独立于 PLC 运行）以防止不安全的运行。安全电路的设计中应包含以下任务：</p> <ul style="list-style-type: none"> <li>• 标识任何可能造成危险的不正确或意外的执行器操作。</li> <li>• 标识可确保操作不危险的条件，并确定如何独立于 PLC 检测这些条件。</li> <li>• 标识上电和断电时 PLC 如何影响过程，并标识检测错误的方式和时间。此信息仅用于设计正常和预期的异常操作。出于安全考虑，不应依赖此“最佳情况”方案。</li> <li>• 设计可独立于 PLC 来阻止危险运行的手动或机电安全替代装置。</li> <li>• 从独立于 PLC 的电路提供相应状态信息，以便程序和任何操作员界面具有必要的信息。</li> <li>• 标识针对过程安全运行的任何其它安全相关要求。</li> </ul>
规划系统安全	确定访问相关过程所需的保护 (页 166) 级别。可以对 CPU 和程序块进行密码保护，以防受到未经授权的访问。
指定操作员站	<p>根据功能规范的要求，创建以下操作员站的绘图：</p> <ul style="list-style-type: none"> <li>• 显示与过程或机器相关的每个操作员站的位置的总览图。</li> <li>• 操作员站中设备的机械布局图，如显示屏、开关和灯。</li> <li>• 包含 PLC 和信号模块中相关 I/O 的电气图。</li> </ul>

建议步骤	任务
创建组态图	<p>根据功能规范的要求，创建控制设备的组态图：</p> <ul style="list-style-type: none"> <li>• 显示与过程或机器相关的每个 PLC 位置的总览图。</li> <li>• 每个 PLC 和任何 I/O 模块的机械布局图，其中包括任何控制柜及其它设备。</li> <li>• 每个 PLC 和任何 I/O 模块的电气图，其中包括设备型号、通信地址和 I/O 地址。</li> </ul>
创建符号名称的列表	创建绝对地址的符号名称列表。不仅包括物理 I/O 信号，也包括要在程序中使用的其它元素（如变量名）。

## 6.2 构建用户程序

创建用于自动化任务的用户程序时，需要将程序的指令插入代码块中：

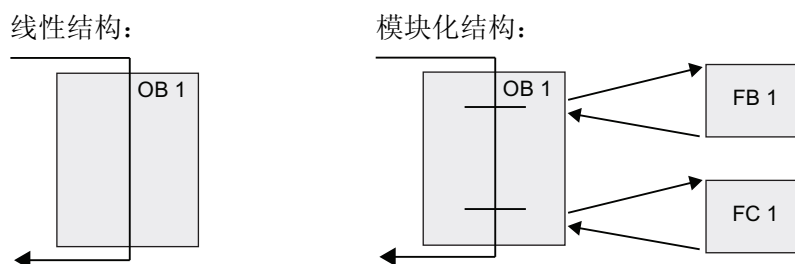
- 组织块 (OB) 对应于 CPU 中的特定事件，并可中断用户程序的执行。用于循环执行用户程序的默认组织块 (OB 1) 为用户程序提供基本结构，是唯一一个用户必需的代码块。如果程序中包括其它 OB，这些 OB 会中断 OB 1 的执行。其它 OB 可执行特定功能，如用于启动任务、用于处理中断和错误或者用于按特定的时间间隔执行特定的程序代码。
- 功能块 (FB) 是从另一个代码块 (OB、FB 或 FC) 进行调用时执行的子例程。调用块将参数传递到 FB，并标识可存储特定调用数据或该 FB 实例的特定数据块 (DB)。更改背景 DB 可使通用 FB 控制一组设备的运行。例如，借助包含每个泵或阀门的特定运行参数的不同背景数据块，一个 FB 可控制多个泵或阀。
- 功能 (FC) 是从另一个代码块 (OB、FB 或 FC) 进行调用时执行的子例程。FC 不具有相关的背景 DB。调用块将参数传递给 FC。FC 中的输出值必须写入存储器地址或全局 DB 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用于创建用户程序：

- 线性程序按顺序逐条执行用于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用于循环执行程序的组织块 (OB 1) 中。
- 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的工艺功能相对应的更小的次级任务。每个代码块都为每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。





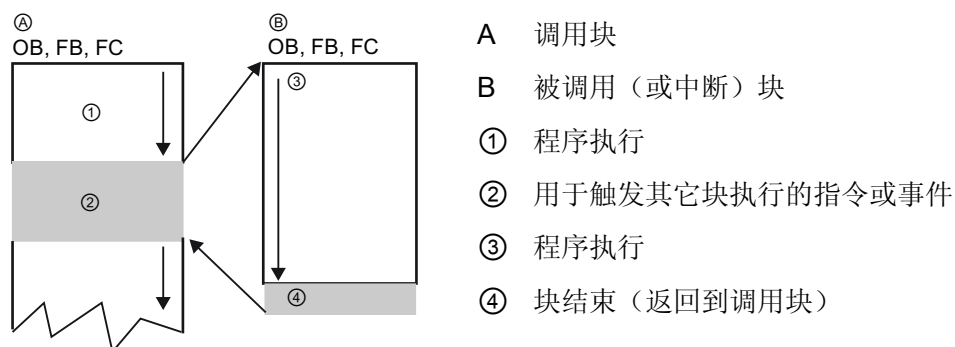
通过创建可在用户程序中重复使用的通用代码块，可简化用户程序的设计和实现。使用通用代码块具有许多优点：

- 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。
- 创建与特定工艺功能相关的模块化组件，有助于简化对已完成应用程序的调试，并减少调试过程中所用的时间。

## 6.3 使用块来构建程序

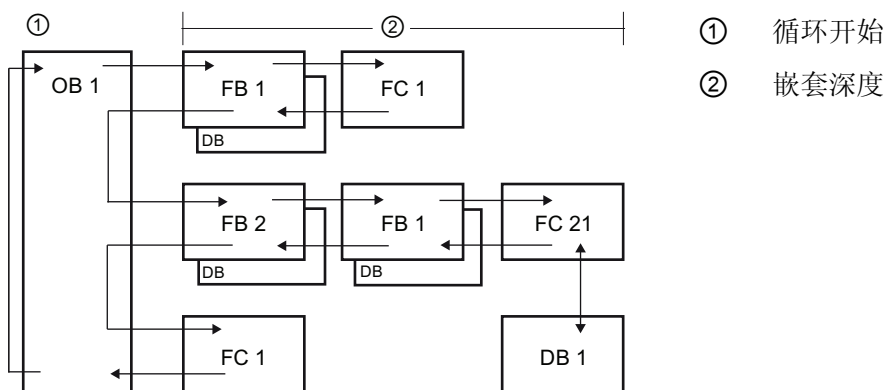
通过设计 FB 和 FC 执行通用任务，可创建模块化代码块。然后可通过由其它代码块调用这些可重复使用的模块来构建程序。调用块将设备特定的参数传递给被调用块。

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。继续执行该块调用之后的指令。



### 6.3 使用块来构建程序

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 4：程序循环 OB 加 3 层对代码块的调用。



#### 6.3.1 组织块 (OB)

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，可被其它各种类型的程序处理中断。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。

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 根据分配给各个 OB 的优先级来确定中断事件的处理顺序。每个事件都具有一个特定的处理优先级。各 OB 的执行顺序取决于它们在优先等级中的相应优先级。多个中断事件可合并为优先级等级。更多相关信息，请参见 PLC 概念一章，执行用户程序小节 (页 69)。

### 在某等级的 OB 内创建附加 OB

可为用户程序，甚至为程序循环和启动 OB 等级创建多个 OB。使用“添加新块”(Add new block) 对话框创建 OB。输入 OB 的名称并输入 200 或更大的数作为 OB 编号。

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 将执行下一个更高的程序循环 OB（例如 OB 200）。

## 组态 OB 的运行



可修改 OB 的运行参数。例如，可为延时 OB 或循环 OB 组态时间参数。

## 6.3.2 功能 (FC)

功能 (FC) 是通常用于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器位置。例如，可使用 FC 执行标准运算和可重复使用的运算（例如数学计算）或者执行工艺功能（如使用位逻辑运算执行独立的控制）。FC 也可以在程序中的不同位置多次调用。此重复使用简化了对经常重复发生的任务的编程。

FC 不具有相关的背景数据块 (DB)。对于用于计算该运算的临时数据，FC 采用了局部数据堆栈。不保存临时数据。要长期存储数据，可将输出值赋给全局存储器位置，如 M 存储器或全局 DB。

## 6.3.3 功能块 (FB)

功能块 (FB) 是使用背景数据块保存其参数和静态数据的代码块。FB 具有位于数据块 (DB) 或“背景”DB 中的变量存储器。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。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。随后在同一扫描周期或其它扫描周期中调用该功能块时可使用这些值。

## 可重复使用的代码块和关联的存储区

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 都具有一个或多个背景 DB。调用 FB 时，也需要指定包含块参数以及用于该调用或 FB “实例”的静态局部数据的背景 DB。FB 完成执行后，背景 DB 将保留这些值。

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。

FB 将 Input、Output 和 InOut 以及静态参数存储在背景数据块中。

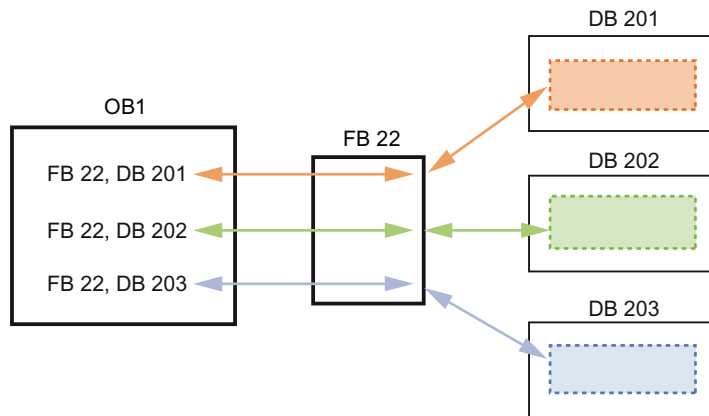
### 在背景数据块中分配起始值

背景数据块存储每个参数的默认值和起始值。起始值提供在执行 FB 时使用的值。然后可在用户程序执行期间修改起始值。

FB 接口还提供一个“默认值”(Default value) 列，使您能够在编写程序代码时为参数分配新的起始值。然后将 FB 中的这个默认值传给关联背景数据块中的起始值。如果不在 FB 接口中为参数分配新的起始值，则将背景数据块中的默认值复制到起始值。

### 使用带多个 DB 的单个 FB

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。

### 6.3.4 数据块 (DB)

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。

## 6.4 了解数据一致性

相关代码块执行完成后，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。）

---

**说明**

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。

---

可将 DB 组态为只读：

1. 在项目浏览器中右键单击相应 DB，然后在右键快捷菜单中选择“属性”(Properties)。
2. 在“属性”(Properties) 对话框中选择“特性”(Attributes)。
3. 选择“在设备中写保护数据块”(Data block write-protected in the device) 选项并单击“确定”(OK)。

### 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。

## 6.4 了解数据一致性

CPU 为所有基本数据类型（例如 Word 或 DWord）和所有系统定义的结构（例如 IEC\_TIMERS 或 DTL）保持数据一致性。值的读/写操作无法中断。（例如，在读写四

字节的 DWord 之前，CPU 会防止对该 DWord 值进行访问。) 为确保程序循环 OB 和中断 OB 无法同时写入同一个存储单元，在程序循环 OB 中的读/写操作完成之前，CPU 不会执行中断 OB。

如果用户程序共享存储器中在程序循环 OB 和中断 OB 之间生成的多个值，用户程序还必须确保在修改或读取这些值时保持一致性。可以在程序循环 OB 中使用 DIS\_AIRT（禁用报警中断）和 EN\_AIRT（启用报警中断）指令，以防止对共享值进行访问。

- 在代码块中插入 DIS\_AIRT 指令，以确保在读/写操作期间无法执行中断 OB。
- 插入读/写能够被中断 OB 更改的值的指令。
- 在顺序结尾插入 EN\_AIRT 指令，以取消 DIS\_AIRT，并允许执行中断 OB。

HMI 设备或另一个 CPU 发出的通信请求也能够中断程序循环 OB 的执行。通信请求也会导致与数据一致性相关的问题。CPU 确保基本数据类型始终由用户程序指令执行一致地读取和写入。由于通信会周期性地中断用户程序，因而不能保证 HMI 能够同时更新 CPU 中的多个值。例如，给定 HMI 画面上显示的值可能来自 CPU 的不同扫描周期。

PtP (Point-to-Point, 点到点) 指令、PROFINET 指令 (例如, TSEND\_C 和 TRCV\_C) 和 PROFINETS 分布式 I/O 指令和 PROFIBUS 分布式 I/O 指令 (页 290) 可用于传送被中断的数据缓冲区。通过避免对程序循环 OB 和中断 OB 中的缓冲区进行任何读/写操作，可以确保数据缓冲区的数据一致性。如果需要在中断 OB 中修改这些指令的缓冲区值，请使用 DIS\_AIRT 指令延迟所有中断 (中断 OB 或源自 HMI 或另一个 CPU 的通信中断)，直到执行了 EN\_AIRT 指令。

---

#### 说明

使用 DIS\_AIRT 指令延迟中断 OB 的处理，直到执行了 EN\_AIRT 指令，以此影响用户程序的中断等待时间 (从事件发生到执行中断 OB 的时间)。

---

## 6.5 编程语言

STEP 7 为 S7-1200 提供以下标准编程语言：

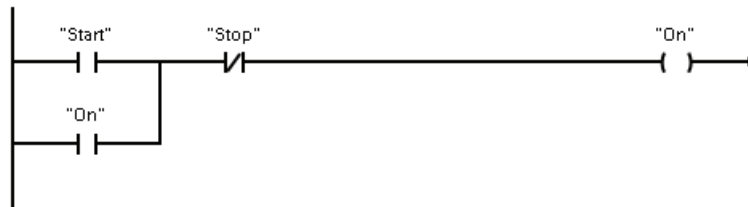
- LAD (梯形图逻辑) 是一种图形编程语言。它使用基于电路图 (页 156) 的表示法。
- FBD (功能块图) 是基于布尔代数 (页 157) 中使用的图形逻辑符号的编程语言。
- SCL (结构化控制语言) 是一种基于文本的高级编程语言 (页 157)。

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。

### 6.5.1 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。

---

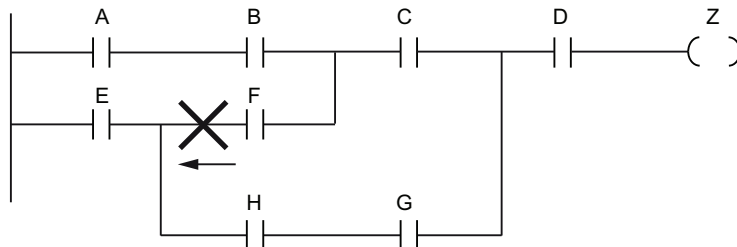
#### 说明

每个 LAD 程序段都必须使用线圈或功能框指令来终止。

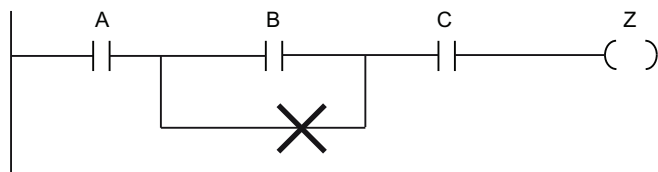
---

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。



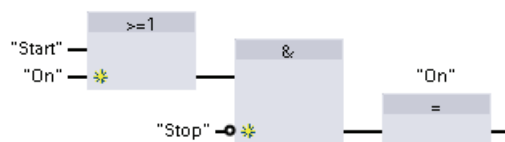
- 不能创建可能导致短路的分支。





## 6.5.2 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。

## 6.5.3 SCL

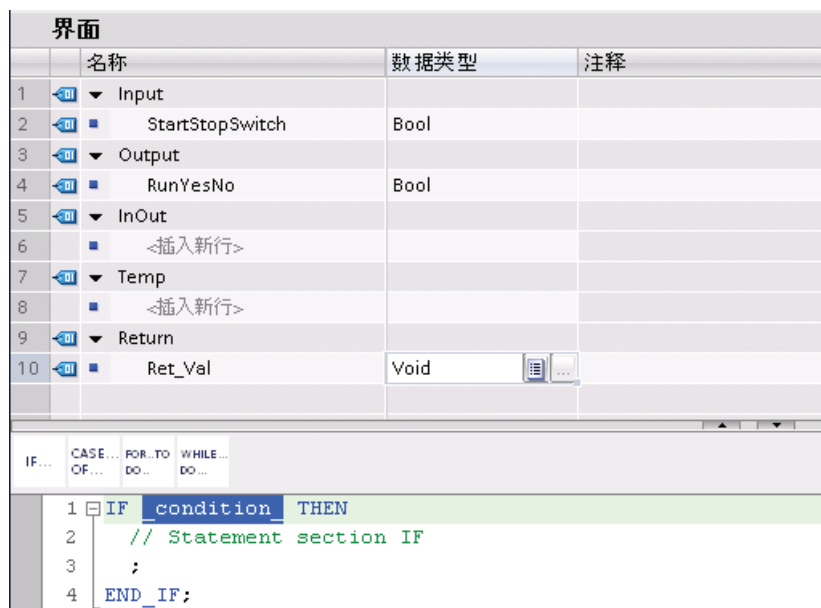
结构化控制语言 (SCL, Structured Control Language) 是用于 SIMATIC S7 CPU 的基于 PASCAL 的高级编程语言。SCL 支持 STEP 7 的块结构 (页 149)。还可以在用 SCL 编写的程序块中包括用 LAD 和 FBD 编写的程序块。

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 也使用标准的 PASCAL 程序控制操作，如 IF-THEN-ELSE、CASE、REPEAT-UNTIL、GOTO 和 RETURN。SCL 编程语言中的语法元素还可以使用所有的 PASCAL 参考。许多 SCL 的其它指令（如定时器和计数器）与 LAD 和 FBD 指令匹配。有关特定指令的更多信息，请参见基本指令 (页 177) 和扩展指令 (页 257) 章节中的特定指令。

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

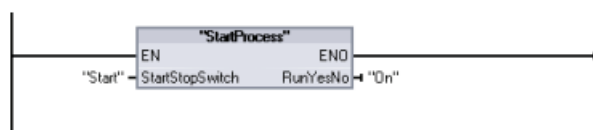
可以直接在代码部分输入指令的 SCL 代码。要了解更复杂的指令，只需从指令树拖动 SCL 指令并将其放入程序中。也可以使用任意文本编辑器创建 SCL 程序，然后将相应文件导入 STEP 7 中。



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。
- **静态（仅适用于 FB，上述示例适用于 FC）：** 静态变量用于在背景数据块中存储静态中间结果。静态数据会一直保留，直到多个周期后被覆盖。块的名称（在此代码块中作为多重背景调用）也存储在静态局部数据中。
- **温度：** 这些参数是执行代码块期间使用的临时变量。

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”。

### 构造 SCL 表达式

SCL 表达式是用于计算值的公式。表达式由操作数和运算符（如 \*、/、+ 或 -）组成。操作数可以是变量、常量或表达式。

表达式的计算按一定的顺序进行，具体由以下因素决定：

- 每个运算符均具有预定义的优先级，首先执行优先级最高的运算。
- 按从左至右的顺序处理优先级相同的运算符。
- 可使用圆括号指定要一起计算的一系列运算符。

表达式的结果可用于将值分配给程序使用的变量、用作由控制语句使用的条件、用作其它 SCL 指令的参数或者用于调用代码块。

表格 6-2 SCL 中的运算符

类型	操作	操作员	优先级
圆括号	(表达式)	(,)	1
数学	乘方	**	2
	符号（一元加号）	+	3
	符号（一元减号）	-	3
	倍增	*	4
	除法	/	4
	取模	MOD	4
	加法	+	5
	减法	-	5
比较	小于	<	6
	小于或等于	<=	6
	大于	>	6
	大于或等于	>=	6
	等于	=	7
	不等于	<>	7
位逻辑	取反（一元）	NOT	3
	AND 逻辑运算	AND 或 &	8
	异或逻辑运算	XOR	9
	OR 逻辑运算	OR	10
赋值	赋值	:=	11

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 /
- 全局变量的寻址: "<变量名称>" (变量名称或数据块名称括在双引号内)
- 局部变量的寻址: #<变量名称> (在变量名称前加"#"符号)

以下示例显示了用法不同的各种表达式。

<code>"C" := #A+#B;</code>	将两个局部变量之和赋值给一个变量
<code>"Data_block_1".Tag := #A;</code>	为数据块变量赋值
<code>IF #A &gt; #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

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。

## 控制语句

控制语句是 SCL 表达式的一种专用类型，可用于执行以下任务：

- 程序分支
- 重复 SCL 编程代码的某些部分
- 跳转到 SCL 程序的其它部分
- 按条件执行

SCL 控制语句包括 IF-THEN、CASE-OF、FOR-TO-DO、WHILE-DO、REPEAT-UNTIL、CONTINUE、GOTO 和 RETURN。

一条语句通常占一行代码。可以在一行中输入多条语句，或者可将一条语句断开成多行代码以使代码易于阅读。分隔符（如制表符、换行符和多余空格）在语法检查期间会被忽略。END 语句可终止控制语句。

以下示例显示的是 FOR-TO-DO 控制语句。（两种形式的代码在语法上均有效。）

```
FOR x := 0 TO max DO sum := sum + value(x); END_FOR;
FOR x := 0 TO max DO
    sum := sum + value(x);
END_FOR;
```

还可以为控制语句提供标签。用语句前的逗号将标签隔开：

**Label:** <语句>;

有关完整的 SCL 编程语言参考，请参见 STEP 7 在线帮助。

## 条件

条件是一个比较表达式或逻辑表达式，其结果为 **BOOL** 类型（值为 **TRUE** 或 **FALSE**）。以下示例显示了各种类型的条件。

<code>#Temperature &gt; 50</code>	关系表达式
<code>#Counter &lt;= 100</code>	
<code>#CHAR1 &lt; 'S'</code>	
<code>(#Alpha &lt;&gt; 12) AND NOT #Beta</code>	比较和逻辑表达式
<code>5 + #Alpha</code>	算术表达式

条件可以使用算术表达式：

- 如果结果是非零的任何值，则表达式的条件为 **TRUE**。
- 如果结果为零，则表达式的条件为 **FALSE**。

## 寻址

与 LAD 和 FBD 一样，SCL 允许用户在用户程序中使用变量地址（符号寻址）或绝对地址。SCL 还允许使用变量作为数组索引。

### 绝对寻址

`I0.0`  
`MB100`

### 符号寻址

<code>"PLC_Tag_1"</code>	PLC 变量表中的变量
<code>"Data_block_1".Tag_1</code>	数据块中的变量
<code>"Data_block_1".MyArray[#i]</code>	数据块数组中的数组元素

## 使用 PEEK 和 POKE 指令进行索引寻址

SCL 提供 PEEK 和 POKE 指令，可以用来读取或写入数据块、I/O 或存储器。而您提供操作中具体字节偏移量或位偏移量的参数。

## 说明

与数据块一起使用 PEEK 和 POKE 指令时，必须使用标准（未优化的）数据块。同时需要注意 PEEK 和 POKE 指令仅用于传输数据。它们无法识别地址中的数据类型。

```
PEEK(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_);
```

读取引用数据块、I/O 或存储区中由 byteOffset 引用的字节。

引用数据块示例：

```
%MB100 := PEEK(area:=16#84,
               dbNumber:=1, byteOffset:=#i);
```

引用 IB3 输入示例：

```
%MB100 := PEEK(area:=16#81,
               dbNumber:=0, byteOffset:=#i); // when
#i = 3
```

```
PEEK_WORD(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_);
```

读取引用数据块、I/O 或存储区中由 byteOffset 引用的字。

示例：

```
%MW200 := PEEK_WORD(area:=16#84,
                    dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
            dbNumber:=_in_,
            byteOffset:=_in_);
```

读取引用数据块、I/O 或存储区中由 byteOffset 引用的双字。

示例：

```
%MD300 := PEEK_DWORD(area:=16#84,
                     dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_);
```

读取引用数据块、I/O 或存储区中由 bitOffset 和 byteOffset 引用的布尔值。

示例：

```
%MB100.0 := PEEK_BOOL(area:=16#84,
                      dbNumber:=1, byteOffset:=#ii,
                      bitOffset:=#j);
```

```
POKE (area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_,
      value:=_in_);
```

向由 `byteOffset` 引用的引用数据块、I/O 或存储区中写入值 (Byte、Word 或 DWord)

引用数据块示例:

```
POKE(area:=16#84, dbNumber:=2,
      byteOffset:=3, value:="Tag_1");
```

引用 QB3 输出示例:

```
POKE (area:=16#82, dbNumber:=0,
      byteOffset:=3, value:="Tag_1");
```

向由 `bitOffset` 和 `byteOffset` 引用的引用数据块、I/O 或存储区中写入布尔值

示例:

```
POKE_BOOL(area:=16#84, dbNumber:=2,
           byteOffset:=3, bitOffset:=5,
           value:=0);
```

将引用源数据块、I/O 或存储区从引用字节偏移量开始的共“count”个字节，写入引用目标数据块、I/O 或存储区中引用的 `byteOffset` 区域

示例:

```
POKE_BLK(area_src:=16#84,
          dbNumber_src:=#src_db,
          byteOffset_src:=#src_byte,
          area_dest:=16#84,
          dbNumber_dest:=#src_db,
          byteOffset_dest:=#src_byte,
          count:=10);
```

```
POKE_BOOL(area:=_in_,
          dbNumber:=_in_,
          byteOffset:=_in_,
          bitOffset:=_in_,
          value:=_in_);
```

```
POKE_BLK(area_src:=_in_,
          dbNumber_src:=_in_,
          byteOffset_src:=_in_,
          area_dest:=_in_,
          dbNumber_dest:=_in_,
          byteOffset_dest:=_in_,
          count:=_in_);
```

对于 PEEK 和 POKE 指令，“area”、“area\_src”和“area\_dest”参数可以使用以下值。对于数据块以外的其它区域，`dbNumber` 参数必须为 0。

16#81	I
16#82	Q
16#83	M
16#84	DB

### 从 SCL 程序中调用其它代码块

要调用用户程序中的其它代码块，只需使用参数输入 FB 或 FC 的名称（或绝对地址）。对于 FB，还必须提供 FB 待调用的背景数据块。

<DB 名称> (参数列表)	作为单个背景调用
<#背景名称> (参数列表)	作为多重背景调用

```
"MyDB" (MyInput:=10, MyInOut:="Tag1");
```

<FC 名称> (参数列表)	标准调用
<操作数>:=<FC 名称> (参数列表)	在表达式中调用

```
"MyFC" (MyInput:=10, MyInOut:="Tag1");
```

还可将块从导航树中拖动到 SCL 程序编辑器中，然后完成参数分配。

### 6.5.4 LAD、FBD 和 SCL 的 EN 和 ENO

#### 确定指令的“能流”（EN 和 ENO）

特定指令（如数学和移动指令）为 EN 和 ENO 提供参数。这些参数与 LAD 或 FBD 中的能流有关并确定在该扫描期间是否执行指令。SCL 还允许用户为代码块设置 ENO 参数。

- EN（使能输入）是布尔输入。要执行功能框指令，能流 (EN = 1) 必须出现在此输入端。如果 LAD 框的 EN 输入直接连接到左侧电源线，将始终执行该指令。
- ENO（使能输出）是布尔输出。如果该功能框在 EN 输入端有能流且正确执行了其功能，则 ENO 输出会将能流 (ENO = 1) 传递到下一个元素。如果执行功能框指令时检测到错误，则在产生该错误的功能框指令处终止该能流 (ENO = 0)。

表格 6-3 EN 和 ENO 的操作数

程序编辑器	输入/输出	操作数	数据类型
LAD	EN, ENO	能流	Bool
FBD	EN	I、I:P、Q、M、DB、Temp、能流	Bool
	ENO	能流	Bool



程序编辑器	输入/输出	操作数	数据类型
SCL	EN <sup>1</sup>	TRUE, FALSE	Bool
	ENO <sup>2</sup>	TRUE, FALSE	Bool

<sup>1</sup> EN 仅适用于 FB。

<sup>2</sup> ENO 可以与 SCL 代码块一起使用。代码块完成时，用户必须组态 SCL 编译器来设置 ENO。

### 通过组态 SCL 来设置 ENO

要组态 SCL 编译器以设置 ENO，请按以下步骤操作：

1. 从“选项”(Options) 菜单中选择“设置”(Settings) 命令。
2. 展开“PLC 编程”(PLC programming) 属性并选择“SCL (结构化控制语言)”(SCL (Structured Control Language))。
3. 选择“自动设置 ENO”(Set ENO automatically) 选项。

### Ret\_Val 或 Status 参数对 ENO 的影响

某些指令（如通信指令或字符串转换指令）提供一个输出参数，其中包含有关指令处理的信息。例如，某些指令提供通常为 Int 数据类型的 Ret\_Val（返回值）参数，其中包含 -32768 到 +32767 范围内的状态信息。还有些指令提供通常为 Word 数据类型的 Status 参数，其中存储十六进制值 16#0000 到 16#FFFF 范围内的状态信息。Ret\_Val 或 Status 参数中存储的数字值确定该指令的 ENO 状态。

- Ret\_Val: 介于 0 至 32767 的值通常设置 ENO = 1（即 TRUE）。介于 -32768 至 -1 的值通常设置 ENO = 0（即 FALSE）。要评估 Ret\_Val，将表示法更改为十六进制。
- Status: 介于 16#0000 至 16#7FFF 的值通常设置 ENO = 1（即 TRUE）。介于 16#8000 至 16#FFFF 的值通常设置 ENO = 0（即 FALSE）。

需要多次扫描才能执行的指令通常提供 Busy 参数 (Bool)，用于表示指令处于活动状态，但尚未完成执行。此类指令通常还提供 Done 参数 (Bool) 和 Error 参数 (Bool)。Done 表示指令已完成且无错误，而 Error 表示指令已完成，但存在错误情况。

- Busy = 1（即 TRUE）时，ENO = 1（即 TRUE）。
- Done = 1（即 TRUE）时，ENO = 1（即 TRUE）。
- Error = 1（即 TRUE）时，ENO = 0（即 FALSE）。

参见

OK 和 Not OK 指令 (页 202)

## 6.6 保护

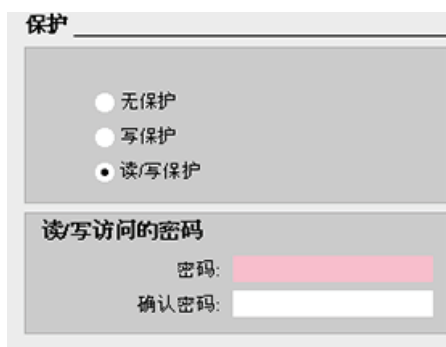
### 6.6.1 CPU 的访问保护

CPU 提供了三个安全等级，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。

密码区分大小写。

要组态密码，请按以下步骤操作：

1. 在“设备配置”(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。



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。HMI 功能同样也不受限制。

表格 6-4 CPU 的安全级别

安全等级	访问限制
无保护	允许完全访问，没有密码保护。
写保护	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)。
读/写保护	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)。

## 6.6.2 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：

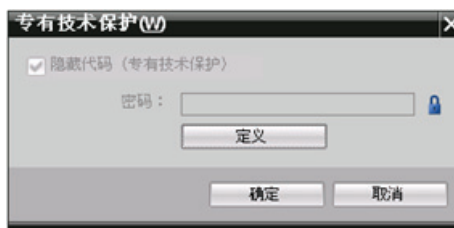
- 块标题、块注释和块属性
- 传送参数（IN、OUT、IN\_OUT、Return）
- 程序的调用结构
- 交叉引用中的全局变量（不带使用时的信息），但局部变量已隐藏

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。

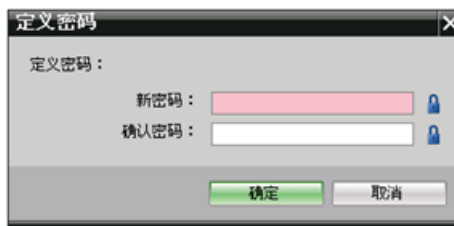
使用代码块的“属性”(Properties) 任务卡组态该块的专有技术保护。打开代码块后，从“属性”(Properties) 中选择“保护”(Protection)。



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)。



### 6.6.3 复制保护

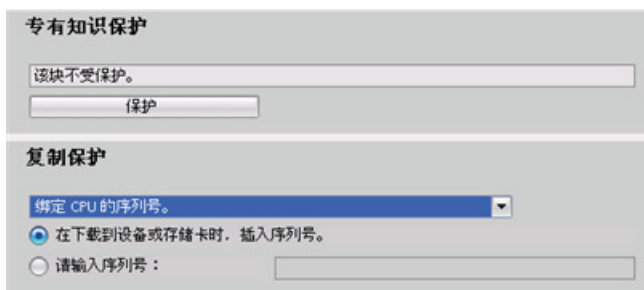
附加安全特性允许捆绑程序或代码块，以用于特定存储卡或 CPU。该特性对于保护您的知识产权特别有用。当您程序或块与特定设备捆绑在一起时，就会将程序或代码块限制为仅用于特定存储卡或 CPU。该特性允许您以电子方式（如通过 Internet 或通过电子邮件）或通过发送内存匣分布程序或代码块。

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，选择将代码块捆绑到存储卡或特定 CPU 的相应选项。



3. 选择复制保护的类型并输入存储卡或 CPU 的序列号。

### 说明

序列号区分大小写。

## 6.7 下载程序的元素

可将项目的元素从编程设备下载到 CPU。下载项目时，CPU 会将用户程序（OB、FC、FB 和 DB）存储在永久存储器中。



## 6.8 从 CPU 上传

可从以下任何位置将项目从编程设备下载到 CPU:

- “项目树”(Project tree): 右键单击程序元素, 然后单击上下文相关的“下载”(Download) 选择项。
- “在线”(Online) 菜单: 单击“下载到设备”(Download to device) 选择项。
- 工具栏: 单击“下载到设备”(Download to device) 图标。

## 6.8 从 CPU 上传

### 6.8.1 复制项目元素

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 731)中, 确认所选项。

此外，还可以按照以下步骤操作：

1. 在项目浏览器中展开“在线访问”(Online access) 的节点，以选择在线 CPU 中的程序块：
2. 展开网络节点并双击“更新可访问设备”(Update accessible devices)。
3. 展开 CPU 的节点。
4. 从在线 CPU 中拖动“程序块”(Program blocks) 文件夹，然后放入您离线项目的“程序块”(Program blocks) 文件夹中。
5. 在“上传预览”(Upload preview) 对话框中选择“继续”(Continue) 框，然后单击“从设备上传”(Upload from device) 按钮。



完成上传后，将在离线区域中显示所有程序块、工艺块和变量。



### 说明

可以将在线 CPU 中的程序块复制到现有程序中。离线项目的“程序块”(Program-blocks) 文件夹不需要为空。但是，现有程序将被删除并用在线 CPU 中的用户程序替代。

## 6.8.2

### 使用比较功能

使用 STEP 7 中的“比较”编辑器 (页 737)，可以查找在线和离线项目之间的差异。此功能在从 CPU 进行上传之前非常有用。

## 6.9 调试和测试程序

### 6.9.1 监视和修改 CPU 中的数据

如下表所示，可以监视和修改在线 CPU 中的值。

表格 6-5 使用 STEP 7 监视和修改数据

编辑器	监视	修改	强制
监视表格	支持	支持	无
强制表格	支持	不支持	支持
程序编辑器	支持	支持	不支持
变量表	支持	不支持	不支持
DB 编辑器	支持	不支持	不支持



通过监视表格监视

通过 LAD 编辑器监视

有关 监视和修改 CPU 中的数据 (页 738)的更多信息，请参见“在线和诊断”一章。

### 6.9.2 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 745) 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”）。

---

### 6.9.3 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 6-6 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 (Quantity)	使用数量

列	说明
位置 (Location)	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 (Address)	操作数的地址
类型 (Type)	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

#### 6.9.4 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。

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。



## 基本指令



### 7.1 位逻辑

#### 7.1.1 位逻辑触点和线圈

使用 LAD 和 FBD 处理布尔逻辑非常高效。SCL 不但非常适合处理复杂的数学计算和项目控制结构，而且也可以使用 SCL 处理布尔逻辑。

#### LAD 触点

表格 7-1 常开触点和常闭触点

LAD	SCL	说明
"IN" 	<pre>IF in THEN     Statement; ELSE     Statement; END IF;</pre>	常开触点和常闭触点：可将触点相互连接并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 I 端子。CPU 扫描已连接的输入信号并持续更新过程映像输入寄存器中的相应状态值。
"IN" 	<pre>IF NOT (in) THEN     Statement; ELSE     Statement; END_IF;</pre>	通过在 I 偏移量后追加“:P”，可指定立即读取物理输入（例如：“%I3.4:P”）。对于立即读取，直接从物理输入读取位数据值，而非从过程映像中读取。立即读取不会更新过程映像。

表格 7-2 参数的数据类型

参数	数据类型	说明
IN	Bool	分配位


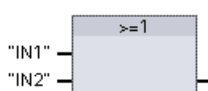
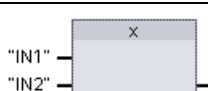
- 在赋的位值为 1 时，常开触点将闭合 (ON)。
- 在赋的位值为 0 时，常闭触点将闭合 (ON)。
- 以串联方式连接的触点创建 AND 逻辑程序段。
- 以并联方式连接的触点创建 OR 逻辑程序段。

**FBD、AND、OR 和 XOR 功能框**

在 FBD 编程中，LAD 触点程序段变为与 (&)、或 (>=1) 和异或 (x) 功能框程序段，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并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。

表格 7-3 AND、OR 和 XOR 功能框

FBD	SCL <sup>1</sup>	说明
	<pre>out := in1 AND in2;</pre>	AND 功能框的所有输入必须都为“真”，输出才为“真”。
	<pre>out := in1 OR in2;</pre>	OR 功能框只要有一个输入为“真”，输出就为“真”。
	<pre>out := in1 XOR in2;</pre>	XOR 功能框必须有奇数个输入为“真”，输出才为“真”。

<sup>1</sup> 对于 SCL： 必须将运算的结果赋给要用于其它语句的变量。

表格 7-4 参数的数据类型

参数	数据类型	说明
IN1, IN2	Bool	输入位

## NOT 逻辑反相器

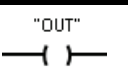

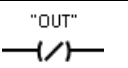
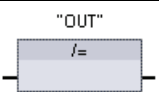
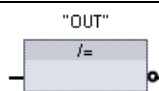
表格 7-5 NOT 逻辑反相器

LAD	FBD	SCL	说明
	 	NOT	<p>对于 FBD 编程，可从“收藏夹”(Favorites) 工具栏或指令树中拖动“取反二进制输入”(Negate binary input) 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。</p> <p>LAD NOT 触点取反能流输入的逻辑状态。</p> <ul style="list-style-type: none"> <li>• 如果没有能流流入 NOT 触点，则会有能流流出。</li> <li>• 如果有能流流入 NOT 触点，则没有能流流出。</li> </ul>

## 输出线圈和赋值功能框

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，同时将指定的位设置为等于能流状态。控制执行器的输出信号连接到 CPU 的 Q 端子。在 RUN 模式下，CPU 系统将连续扫描输入信号，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 系统会将存储在过程映像寄存器中的新的输出状态响应传送到已连接的输出端子。

表格 7-6 输出线圈 (LAD) 和输出赋值功能框 (FBD)

LAD	FBD	SCL	说明
		<code>out := &lt;布尔表达式&gt;;</code>	<p>在 FBD 编程中，LAD 线圈变为分配 (= 和 /=) 功能框，可在其中为功能框输出指定位地址。功能框输入和输出可连接到其它功能框逻辑，用户也可以输入位地址。</p> <p>通过在 Q 偏移量后加上“:P”，可指定立即写入物理输出（例如：“%Q3.4:P”）。对于立即写入，将位数据值写入过程映像输出并直接写入物理输出。</p>
		<code>out := NOT &lt;布尔表达式&gt;;</code>	
			

表格 7-7 参数的数据类型

参数	数据类型	说明
OUT	Bool	分配位

7.1 位逻辑

- 如果有能流通过输出线圈或启用了 FBD“=”功能框，则输出位设置为 1。
- 如果没有能流通过输出线圈或未启用 FBD“=”赋值功能框，则输出位设置为 0。
- 如果有能流通过反向输出线圈或启用了 FBD“/=”功能框，则输出位设置为 0。
- 如果没有能流通过反向输出线圈或未启用 FBD“/=”功能框，则输出位设置为 1。

7.1.2 置位和复位指令

置位和复位 1 位

表格 7-8 S 和 R 指令

LAD	FBD	SCL	说明
		不提供	S（置位）激活时，OUT 地址处的数据值设置为 1。S 未激活时，OUT 不变。
		不提供	R（复位）激活时，OUT 地址处的数据值设置为 0。R 未激活时，OUT 不变。

- 1 对于 LAD 和 FBD：这些指令可放置在程序段的任何位置。
- 2 对于 SCL：必须在应用程序内编写代码来复制该函数。

表格 7-9 参数的数据类型

参数	数据类型	说明
IN（或连接到触点/门逻辑）	Bool	要监视的位位置
OUT	Bool	要置位或复位的位位置



## 置位和复位位域

表格 7-10 SET\_BF 和 RESET\_BF 指令

LAD <sup>1</sup>	FBD	SCL	说明
		不提供	SET_BF 激活时，为从地址 OUT 处开始的“n”位分配数据值 1。SET_BF 未激活时，OUT 不变。
		不提供	RESET_BF 为从地址 OUT 处开始的“n”位写入数据值 0。RESET_BF 未激活时，OUT 不变。

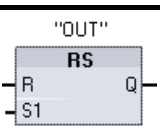
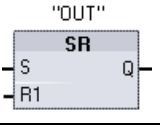
- 1 对于 LAD 和 FBD：这些指令必须是分支中最右端的指令。
- 2 对于 SCL：必须在应用程序内编写代码来复制该函数。

表格 7-11 参数的数据类型

参数	数据类型	说明
OUT	Bool	要置位或复位的位域的起始元素（例如 #MyArray[3]）
n	常数 (UInt)	要写入的位数

## 置位优先和复位优先位锁存

表格 7-12 RS 和 SR 指令

LAD/FBD	SCL	说明
	不提供	RS 是置位优先锁存，其中置位优先。如果置位 (S1) 和复位 (R) 信号都为真，则输出地址 OUT 将为 1。
	不提供	SR 是复位优先锁存，其中复位优先。如果置位 (S) 和复位 (R1) 信号都为真，则输出地址 OUT 将为 0。

- 1 对于 LAD 和 FBD：这些指令必须是分支中最右端的指令。
- 2 对于 SCL：必须在应用程序内编写代码来复制该函数。

表格 7- 13 参数的数据类型

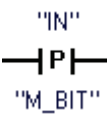

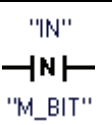

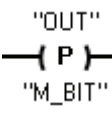
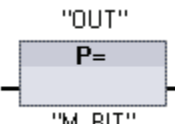
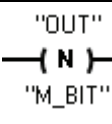
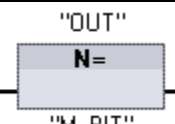
参数	数据类型	说明
S, S1	Bool	置位输入；1 表示优先
R, R1	Bool	复位输入；1 表示优先
OUT	Bool	分配的位输出“OUT”
Q	Bool	遵循“OUT”位的状态

OUT 参数指定置位或复位的位地址。可选 OUT 输出 Q 反映“OUT”地址的信号状态。

指令	S1	R	“OUT”位
RS	0	0	先前状态
	0	1	0
	1	0	1
	1	1	1
SR	<b>S</b>	<b>R1</b>	
	0	0	先前状态
	0	1	0
	1	0	1
	1	1	0



## 7.1.3 上升沿和下降沿指令

表格 7-14 上升沿和下降沿跳变检测

LAD	FBD	SCL	说明
		不提供	<p><b>LAD:</b> 在分配的“IN”位上检测到正跳变（断到通）时，该触点的状态为 TRUE。该触点逻辑状态随后与能流输入状态组合以设置能流输出状态。P 触点可以放置在程序段中除分支结尾外的任何位置。</p> <p><b>FBD:</b> 在分配的输入位上检测到正跳变（关到开）时，输出逻辑状态为 TRUE。P 功能框只能放置在分支的开头。</p>
		不提供	<p><b>LAD:</b> 在分配的输入位上检测到负跳变（开到关）时，该触点的状态为 TRUE。该触点逻辑状态随后与能流输入状态组合以设置能流输出状态。N 触点可以放置在程序段中除分支结尾外的任何位置。</p> <p><b>FBD:</b> 在分配的输入位上检测到负跳变（开到关）时，输出逻辑状态为 TRUE。N 功能框只能放置在分支的开头。</p>
		不提供	<p><b>LAD:</b> 在进入线圈的能流中检测到正跳变（关到开）时，分配的位“OUT”为 TRUE。能流输入状态总是通过线圈后变为能流输出状态。P 线圈可以放置在程序段中的任何位置。</p> <p><b>FBD:</b> 在功能框输入连接的逻辑状态中或输入位赋值中（如果该功能框位于分支开头）检测到正跳变（关到开）时，分配的位“OUT”为 TRUE。输入逻辑状态总是通过功能框后变为输出逻辑状态。P= 功能框可以放置在分支中的任何位置。</p>
		不提供	<p><b>LAD:</b> 在进入线圈的能流中检测到负跳变（开到关）时，分配的位“OUT”为 TRUE。能流输入状态总是通过线圈后变为能流输出状态。N 线圈可以放置在程序段中的任何位置。</p> <p><b>FBD:</b> 在功能框输入连接的逻辑状态中或在输入位赋值中（如果该功能框位于分支开头）检测到负跳变（通到断）时，分配的位“OUT”为 TRUE。输入逻辑状态总是通过功能框后变为输出逻辑状态。N= 功能框可以放置在分支中的任何位置。</p>

<sup>1</sup> 对于 SCL：必须在应用程序内编写代码来复制该函数。

表格 7- 15 P\_TRIG 和 N\_TRIG 指令

LAD/FBD	SCL	说明
	不提供	<p>在 CLK 输入状态 (FBD) 或 CLK 能流输入 (LAD) 中检测到正跳变 (断到通) 时, Q 输出能流或逻辑状态为 TRUE。</p> <p>在 LAD 中, P_TRIG 指令不能放置在程序段的开头或结尾。在 FBD 中, P_TRIG 指令可以放置在除分支结尾外的任何位置。</p>
	不提供	<p>在 CLK 输入状态 (FBD) 或 CLK 能流输入 (LAD) 中检测到负跳变 (通到断) 时, Q 输出能流或逻辑状态为 TRUE。</p> <p>在 LAD 中, N_TRIG 指令不能放置在程序段的开头或结尾。在 FBD 中, N_TRIG 指令可以放置在除分支结尾外的任何位置。</p>

1 对于 SCL: 必须在应用程序内编写代码来复制该函数。

表格 7- 16 参数的数据类型 (P 和 N 触点/线圈、P=、N=、P\_TRIG 和 N\_TRIG)

参数	数据类型	说明
M_BIT	Bool	保存输入的前一个状态的存储器位
IN	Bool	要检测其跳变沿的输入位
OUT	Bool	指示检测到跳变沿的输出位
CLK	Bool	要检测其跳变沿的能流或输入位
Q	Bool	指示检测到沿的输出

所有沿指令均使用存储器位 (M\_BIT) 存储要监视的输入信号的前一个状态。通过将输入的状态与存储器位的状态进行比较来检测沿。如果状态指示在关注的方向上有输入变化, 则会在输出写入 TRUE 来报告沿。否则, 输出会写入 FALSE。

**说明**

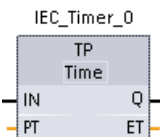
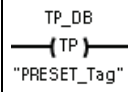
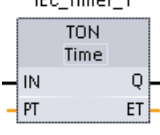
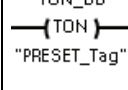
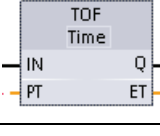
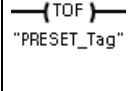
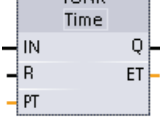
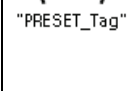
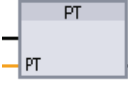
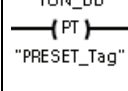

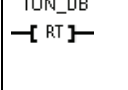
沿指令每次执行时都会对输入和存储器位值进行评估, 包括第一次执行。在程序设计期间必须考虑输入和存储器位的初始状态, 以允许或避免在第一次扫描时进行沿检测。

由于存储器位必须从一次执行保留到下一次执行, 所以应该对每个沿指令都使用唯一的位, 并且不应在程序中的任何其它位置使用该位。还应避免使用临时存储器和可受其它系统功能 (例如 I/O 更新) 影响的存储器。仅将 M、全局 DB 或静态存储器 (在背景 DB 中) 用于 M\_BIT 存储器分配。

## 7.2 定时器

使用定时器指令可创建编程的时间延时。用户程序中可以使用的定时器数仅受 CPU 存储器容量限制。每个定时器均使用 16 字节的 IEC\_Timer 数据类型的 DB 结构来存储功能框或线圈指令顶部指定的定时器数据。STEP 7 会在插入指令时自动创建该 DB。

表格 7-17 定时器指令

LAD/FBD 功能框	LAD 线圈	SCL	说明
		<pre>"IEC_Timer_0_DB".TP (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	TP 定时器可生成具有预设宽度时间的脉冲。
		<pre>"IEC_Timer_0_DB".TON (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	TON 定时器在预设的延时过后将输出 Q 设置为 ON。
		<pre>"IEC_Timer_0_DB".TOF (   IN:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	TOF 定时器在预设的延时过后将输出 Q 重设为 OFF。
		<pre>"IEC_Timer_0_DB".TONR (   IN:=_bool_in_,   R:=_bool_in_,   PT:=_time_in_,   Q=&gt;_bool_out_,   ET=&gt;_time_out_);</pre>	TONR 定时器在预设的延时过后将输出 Q 设置为 ON。在使用 R 输入重置经过的时间之前，会跨越多个定时时段一直累加经过的时间。
仅 FBD: 		(无对应的 SCL 指令)	PT (预设定时器) 线圈会在指定的 IEC_Timer 中装载新的 PRESET 时间值。
仅 FBD: 		(无对应的 SCL 指令)	RT (复位定时器) 线圈会复位指定的 IEC_Timer。

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“IEC\_Timer\_0\_DB”是背景 DB 的名称。

表格 7- 18 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 7- 19 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"> <li>定时器运行期间, 更改 PT 没有任何影响。</li> <li>定时器运行期间, 更改 IN 没有任何影响。</li> </ul>
TON	<ul style="list-style-type: none"> <li>定时器运行期间, 更改 PT 没有任何影响。</li> <li>定时器运行期间, 将 IN 更改为 FALSE 会复位并停止定时器。</li> </ul>
TOF	<ul style="list-style-type: none"> <li>定时器运行期间, 更改 PT 没有任何影响。</li> <li>定时器运行期间, 将 IN 更改为 TRUE 会复位并停止定时器。</li> </ul>
TONR	<ul style="list-style-type: none"> <li>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。</li> <li>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 改回 TRUE 将使定时器从累积的时间值开始定时。</li> </ul>

PT (预设时间) 和 ET (经过的时间) 值以表示毫秒时间的有符号双精度整数形式存储在指定的 IEC\_TIMER DB 数据中。TIME 数据使用 T# 标识符, 可以简单时间单元 (T#200ms 或 200) 和复合时间单元 (如 T#2s\_200ms) 的形式输入。

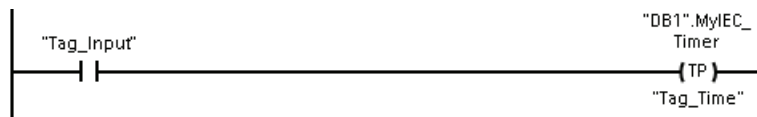
表格 7- 20 TIME 数据类型的大小和范围

数据类型	大小	有效数值范围 <sup>1</sup>
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

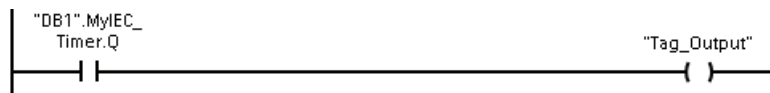
<sup>1</sup> 在定时器指令中，无法使用上面所示 TIME 数据类型的负数范围。负的 PT（预设时间）值在定时器指令执行时被设置为 0。ET（经过的时间）始终为正值。

### 定时器线圈示例

-(TP)-、-(TON)-、-(TOF)- 和 -(TONR)- 定时器线圈必须是 LAD 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。

**重置定时器 -(RT)- 和预设定时器 -(PT)- 线圈**

这些线圈指令可与功能框或线圈定时器一起使用并可放置在中间位置。线圈输出能流状态始终与线圈输入状态相同。若 -(RT)- 线圈激活，指定 IEC\_Timer DB 数据中的 ELAPSED 时间元素将重置为 0。若 -(PT)- 线圈激活，指定 IEC\_Timer DB 数据中的 PRESET 时间元素将重置为 0。

**说明**

在 FB 中放置定时器指令时，可以选择“多重背景数据块”(Multi-instance data block) 选项。各定时器结构名称可以对应不同的数据结构，但定时器数据包含在同一个数据块中，无需为每个定时器都使用一个独立的数据块。这样可减少处理定时器所需的处理时间和数据存储空间。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。

**定时器的运行**

表格 7-21 IEC 定时器的类型

定时器	时序图
<p><b>TP:</b> 脉冲定时器</p> <p>TP 定时器可生成具有预设宽度时间的脉冲。</p>	
<p><b>TON:</b> 接通延迟定时器</p> <p>TON 定时器在预设的延时过后将输出 Q 设置为 ON。</p>	



定时器	时序图
<p><b>TOF:</b> 关断延迟定时器</p> <p>TOF 定时器在预设的延时过后将输出 Q 重置为 OFF。</p>	
<p><b>TONR:</b> 保持型接通延迟定时器</p> <p>TONR 定时器在预设的延时过后将输出 Q 设置为 ON。在使用 R 输入重置经过的时间之前，会跨越多个定时时段一直累加经过的时间。</p>	

### 说明

在 CPU 中，没有给任何特定的定时器指令分配专门的资源。每个定时器使用 DB 存储器中其自身的结构和一个连续运行的内部 CPU 定时器来执行定时。

当由于 TP、TON、TOF 或 TONR 指令的输入上出现沿跳变而启动定时器时，连续运行的内部 CPU 定时器的值将被复制到为该定时器指令分配的 DB 结构的 START 成员中。该起始值在定时器继续运行期间将保持不变，随后将在每次更新定时器时使用。每次启动定时器时，都会从内部 CPU 定时器将一个新的起始值加载到定时器结构中。

更新定时器时，将从内部 CPU 定时器的当前值中减去上述起始值以确定经过的时间。再将经过的时间与预设值进行比较以确定定时器 Q 位的状态。然后在为该定时器分配的 DB 结构中，更新 ELAPSED 和 Q 成员。注意，经过的时间将停留在预设值上（达到预设值后定时器便不会继续累加经过的时间）。

当且仅当满足以下条件时才会执行定时器更新：

- 已执行定时器指令（TP、TON、TOF 或 TONR）
- 某个指令直接引用 DB 中定时器结构的“ELAPSED”成员
- 某个指令直接引用 DB 中定时器结构的“Q”成员

## 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。

## RUN-STOP-RUN 切换或 CPU 循环上电后保留时间数据

如果从运行模式阶段切换到停止模式或 CPU 循环上电并启动了新运行模式阶段，则存储在之前运行模式阶段中的定时器数据将丢失，除非将定时器数据结构指定为具有保持性（TP、TON、TOF 和 TONR 定时器）。

将定时器指令放到程序编辑器中后，如果接受调用选项对话框中的默认设置，则将自动分配一个**无法实现具有保持性**的背景数据块。要使定时器数据具有保持性，必须使用全局数据块或**多重背景数据块**。

## 指定全局数据块将定时器数据存储为保持性数据

无论将定时器放在什么位置（OB、FC 或 FB），该选项都有效。

### 1. 创建一个全局数据块：

- 在项目树中双击“添加新块”(Add new block)。
- 单击数据块 (DB) 图标
- 对于“类型”(Type)，选择“全局数据块”(global DB)。
- 如果希望能够将该数据块中各数据元素选择为具有保持性，则确保选中数据块类型“优化”(Optimized) 框。另一个数据块类型选项“标准 - 与 S7-300/400 兼容”(Standard - compatible with S7-300/400) 仅允许将所有 DB 数据元素都设置为具有保持性或没有保持性。
- 单击“确定”(OK)

### 2. 向该数据块中添加定时器结构：

- 在新的全局数据块中，添加 IEC\_Timer 数据类型的静态变量。
- 在“保持性”(Retain) 列中，选中相应框以使该结构具有保持性。
- 重复此过程为要存储在该数据块中的所有定时器创建结构。可以将每个定时器结构放置在独立的全局数据块中，也可以将多个定时器结构放置在同一个全局数据块中。除定时器外，还可以将其它静态变量放置在该全局数据块中。将多个定时器结构放置在同一个全局数据块中可减少总的块数。
- 可根据需要重命名定时器结构。

### 3. 打开程序块来选择保持性定时器的放置位置（OB、FC 或 FB）。

### 4. 将定时器指令放置在所需位置。

5. 在调用选项对话框出现后，单击“取消”按钮。
6. 在新的定时器指令上方，输入上面所创建全局数据块和定时器结构的名称（请勿使用助手浏览）（例如：“Data\_block\_3.Static\_1”）。

### 指定多重背景数据块以将定时器数据存储为保持性数据

该选项仅对于将定时器放置在 FB 中有效。

该选项取决于创建 FB 时是否选中了“优化”块访问（仅允许符号访问）。创建 FB 后，便无法将复选框设置更改为“优化”(Optimized)；必须在创建 FB 时，在从树中选择“添加新块”(Add new block) 后出现的第一个屏幕上正确选择该复选框。要检查现有 FB 访问属性的组态情况，请在项目树中右键单击该 FB，选择“属性”(Properties)，然后选择“特性”(Attributes)。

如果创建 FB 时选中了“优化”(Optimized) 框（仅允许符号访问）：

1. 打开 FB 进行编辑。
2. 将定时器指令放在 FB 中的所需位置。
3. “调用选项”(Call options) 对话框出现后，单击“多重背景”(Multi instance) 图标。仅在将该指令放置于 FB 中后，“多重背景”(Multi instance) 选项才可用。
4. 如有需要，请在“调用选项”(Call options) 对话框中重命名定时器。
5. 单击“确定”(OK)。定时器指令将出现在编辑器中，而 IEC\_TIMER 结构将出现在“FB 接口”(FB Interface) 的“静态”(Static) 下。
6. 如有必要，打开 FB 接口编辑器（可能需要单击小箭头以展开视图）。
7. 在“静态”(Static) 下，找到刚刚创建的定时器结构。
8. 在此定时器结构的“保持性”(Retain) 列中，改为选择“保持性”(Retain)。此后只要从另一程序块调用此 FB，都将利用此接口定义（包含标有保持性的定时器结构）创建背景数据块。

如果创建 FB 时选中了“标准 - 与 S7-300/400 兼容”(Standard - compatible with S7-300/400) 框（允许符号访问和直接访问）：

1. 打开 FB 进行编辑。
2. 将定时器指令放在 FB 中的所需位置。
3. “调用选项”(Call options) 对话框出现后，单击“多重背景”(Multi instance) 图标。仅在将该指令放置于 FB 中后，“多重背景”(Multi instance) 选项才可用。
4. 如有需要，请在“调用选项”(Call options) 对话框中重命名定时器。

5. 单击“确定”(OK)。定时器指令将出现在编辑器中，而 IEC\_TIMER 结构将出现在“FB 接口”(FB Interface) 的“静态”(Static) 下。
6. 打开将使用此 FB 的块。
7. 将此 FB 置于所需的位置。如此将为该 FB 创建一个背景数据块。
8. 打开将 FB 放入编辑器时创建的背景数据块。
9. 在“静态”(Static) 下，找到所需的定时器结构。在此定时器结构的“保持性”(Retain) 列中，选中相应框使该结构具有保持性。

## 7.3 计数器

表格 7-22 计数器指令

LAD/FBD	SCL	说明
	<pre>"IEC_Counter_0_DB".CTU(   CU:=_bool_in,   R:=_bool_in,   PV:=_int_in,   Q=&gt;_bool_out,   CV=&gt;_int_out);</pre>	<p>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。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。用户在编辑器中放置计数器指令时分配相应的数据块。</p> <ul style="list-style-type: none"> <li>• CTU 是加计数器</li> </ul>
	<pre>"IEC_Counter_0_DB".CTD(   CD:=_bool_in,   LD:=_bool_in,   PV:=_int_in,   Q=&gt;_bool_out,   CV=&gt;_int_out);</pre>	<ul style="list-style-type: none"> <li>• CTD 是减计数器</li> <li>• CTUD 是加减计数器</li> </ul>
	<pre>"IEC_Counter_0_DB".CTUD(   CU:=_bool_in,   CD:=_bool_in,   R:=_bool_in,   LD:=_bool_in,   PV:=_int_in,   QU=&gt;_bool_out,   QD=&gt;_bool_out,   CV=&gt;_int_out);</pre>	

- 1 对于 LAD 和 FBD：从指令名称下的下拉列表中选择计数值数据类型。
- 2 STEP 7 会在插入指令时自动创建 DB。
- 3 在 SCL 示例中，“IEC\_Counter\_0\_DB”是背景 DB 的名称。

表格 7- 23 参数的数据类型

参数	数据类型 <sup>1</sup>	说明
CU, CD	Bool	加计数或减计数, 按加或减一计数
R (CTU, CTUD)	Bool	将计数值重置为零
LD (CTD, CTUD)	Bool	预设值的装载控制
PV	SInt, Int, DInt, USInt, UInt, UDInt	预设计数值
Q, QU	Bool	CV >= PV 时为真
QD	Bool	CV <= 0 时为真
CV	SInt, Int, DInt, USInt, UInt, UDInt	当前计数值

<sup>1</sup> 计数值的数值范围取决于所选的数据类型。如果计数值是无符号整型数, 则可以减计数到零或加计数到范围限值。如果计数值是有符号整数, 则可以减计数到负整数限值或加计数到正整数限值。

用户程序中可以使用的计数器数仅受 CPU 存储器容量限制。计数器占用以下存储器空间:

- 对于 SInt 或 USInt 数据类型, 计数器指令占用 3 个字节。
- 对于 Int 或 UInt 数据类型, 计数器指令占用 6 个字节。
- 对于 DInt 或 UDInt 数据类型, 计数器指令占用 12 个字节。

这些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的 OB 的执行速率限制。指令所在的 OB 的执行频率必须足够高, 以检测 CU 或 CD 输入的所有跳变。要了解更快的计数操作, 请参见 CTRL\_HSC 指令 (页 363)。

#### 说明

在 FB 中放置计数器指令后, 可以选择多重背景数据块选项, 各计数器结构名称可以对应不同的数据结构, 但计数器数据包含在同一个数据块中, 从而无需每个计数器都使用一个单独的数据块。这减少了计数器所需的处理时间和数据存储空间。在共享的多重背景数据块中的计数器数据结构之间不存在交互作用。

## 计数器的运行

表格 7-24 CTU 计数器的运行

计数器	运行
<p>当参数 CU 的值从 0 变为 1 时，CTU 计数器会使计数值加 1。CTU 时序图显示了计数值为无符号整数时的运行（其中，PV = 3）。</p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 如果复位参数 R 的值从 0 变为 1，则当前计数值重置为 0。</li> </ul>	<p>The diagram shows four signals over time: CU (count up), R (reset), CV (current value), and Q (output). CU has four pulses. R has one pulse. CV starts at 0, increments to 1, 2, 3, and 4. Q is high when CV is 3 or 4. A vertical dashed line marks the start of the R pulse, after which CV resets to 0.</p>

表格 7-25 CTD 计数器的运行

计数器	运行
<p>当参数 CD 的值从 0 变为 1 时，CTD 计数器会使计数值减 1。CTD 时序图显示了计数值为无符号整数时的运行（其中，PV = 3）。</p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 如果参数 LOAD 的值从 0 变为 1，则参数 PV（预设值）的值将作为新的 CV（当前计数值）装载到计数器。</li> </ul>	<p>The diagram shows four signals over time: CD (count down), LOAD, CV (current value), and Q (output). CD has four pulses. LOAD has two pulses. CV starts at 0, decrements to -1, and then increments to 3. Q is high when CV is 0 or -1. A vertical dashed line marks the start of the first LOAD pulse, after which CV is loaded with the value 3.</p>

表格 7- 26 CTUD 计数器的运行

计数器	运行
<p>当加计数 (CU) 输入或减计数 (CD) 输入从 0 转换为 1 时，CTUD 计数器将加 1 或减 1。CTUD 时序图显示了计数值为无符号整数时的运行（其中 PV = 4）。</p> <ul style="list-style-type: none"> <li>• 如果参数 CV 的值大于等于参数 PV 的值，则计数器输出参数 QU = 1。</li> <li>• 如果参数 CV 的值小于或等于零，则计数器输出参数 QD = 1。</li> <li>• 如果参数 LOAD 的值从 0 变为 1，则参数 PV 的值将作为新的 CV 装载到计数器。</li> <li>• 如果复位参数 R 的值从 0 变为 1，则当前计数值重置为 0。</li> </ul>	<p>The timing diagram illustrates the operation of a CTUD counter. It shows several input signals: CU (Add Count), CD (Decrement Count), R (Reset), and LOAD (Load). The counter value (CV) is shown as a staircase function that increases when CU is active and decreases when CD is active. The counter value is limited by the preset value (PV = 4), so it cannot exceed 4 or go below 0. The output signals QU (Upper Limit Reached) and QD (Lower Limit Reached) are shown as digital signals that become active (1) when the counter value reaches its upper or lower limit, respectively. The diagram shows the counter starting at 0, increasing to 5, then decreasing to 3, increasing to 5 again, and finally being reset to 0.</p>

### RUN-STOP-RUN 切换或 CPU 循环上电后保留计数器数据

如果从运行模式阶段切换到停止模式或 CPU 循环上电并启动了新运行模式阶段，则存储在之前运行模式阶段中的计数器数据将丢失，除非将定时器数据结构指定为具有保持性（CTU、CTD 和 CTUD 计数器）。

将计数器指令放到程序编辑器中后，如果接受调用选项对话框中的默认设置，则将自动分配一个无法实现具有保持性的背景数据块。要使计数器数据具有保持性，必须使用全局数据块或双重背景数据块。

### 指定全局数据块将计数器数据存储为保持性数据

无论将计数器放在什么位置（OB、FC 或 FB），该选项都有效。

1. 创建一个全局数据块：



- 在项目树中双击“添加新块”(Add new block)。
  - 单击数据块 (DB) 图标
  - 对于“类型”(Type)，选择“全局数据块”(global DB)。
  - 如果希望能够将该数据块中的各个项选择为具有保持性，则确保选中“仅符号访问”(symbolic-access-only) 框。
  - 单击“确定”(OK)
2. 向该数据块添加计数器结构：
- 在新的全局数据块中，添加使用以下计数器数据类型之一的新静态变量。务必要考虑到想要用于预设值和计数值的类型。

计数器数据类型	预设值和计数值的相应类型
IEC_Counter	INT
IEC_SCounter	SINT
IEC_DCounter	DINT
IEC_UCounter	UINT
IEC_USCounter	USINT
IEC_UDCounter	UDINT

1. 在“保持性”(Retain) 列中，选中相应框以使该结构具有保持性。
  - 重复此过程为要存储在该数据块中的所有计数器创建结构。可以将每个计数器结构放置在独立的全局数据块中，也可以将多个计数器结构放置在同一个全局数据块中。除计数器外，还可以将其它静态变量放置在该全局数据块中。将多个计数器结构放置在同一个全局数据块中可减少总的块数。
  - 可根据需要重命名计数器结构。
2. 打开程序块来选择保持性计数器的放置位置 (OB、FC 或 FB)。
3. 将计数器指令放置在所需位置。
4. 在调用选项对话框出现后，单击“取消”按钮。您现在应该看到新的计数器指令，在指令名称的上面和下面均显示“???”。
5. 在新的计数器指令上方，输入上面所创建全局数据块和计数器结构的名称（请勿使用助手浏览）（例如：“Data\_block\_3.Static\_1”）。这需要填入对应的预设值和计数值类型（例如：UInt 对应于 IEC\_UCounter 结构）。

### 指定多重背景数据块以将计数器数据存储为保持性数据

该选项仅对于将计数器放置在 FB 中有效。

该选项取决于是否将 FB 创建为仅符号访问。创建 FB 后，便无法更改“仅符号访问”(Symbolic access only) 复选框的设置；必须在创建 FB 时，在从树中选择“添加新块”(Add new block) 后出现的第一个屏幕上正确选择该复选框。要查看此框针对现有 FB 的组态情况，请在项目树中右键单击该 FB，选择“属性”(properties)，然后选择“特性”(Attributes)。

如果创建 FB 时选中了“仅符号访问”(Symbolic access only) 框：

1. 打开 FB 进行编辑。
2. 将计数器指令放在 FB 中的所需位置。
3. “调用选项”(Call options) 对话框出现后，单击“多重背景”(Multi instance) 图标。仅在将该指令放置于 FB 中后，“多重背景”(Multi instance) 选项才可用。
4. 如有需要，请在“调用选项”(Call options) 对话框中重命名计数器。
5. 单击“确定”(OK)。计数器指令将出现在编辑器中并且预设值和计数值的类型为 INT，而 IEC\_COUNTER 结构将出现在“FB 接口”(FB Interface) 的“静态”(Static) 下。
6. 如有需要，请在计数器指令中将类型从 INT 更改为其它类型之一。计数器结构将相应更改。

计数器指令中显示的类型（对于预设值和计 FB 接口中显示的对应的结构类型数值）

INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

1. 如有必要，打开 FB 接口编辑器（可能需要单击小箭头以展开视图）。
2. 在“静态”(Static) 下，找到刚刚创建的计数器结构。
3. 在此计数器结构的“保持性”(Retain) 列中，改为选择“保持性”(Retain)。此后只要从另一程序块调用此 FB，都将利用此接口定义（包含标有保持性的计数器结构）创建背景数据块。

如果创建 FB 时未选中“仅符号访问”(Symbolic access only) 框：

1. 打开 FB 进行编辑。
2. 将计数器指令放在 FB 中的所需位置。
3. “调用选项”(Call options) 对话框出现后，单击“多重背景”(Multi instance) 图标。 仅在将该指令放置于 FB 中后，“多重背景”(Multi instance) 选项才可用。
4. 如有需要，请在“调用选项”(Call options) 对话框中重命名计数器。
5. 单击“确定”(OK)。 计数器指令将出现在编辑器中并且预设值和计数值的类型为 INT，而 IEC\_COUNTER 结构将出现在“FB 接口”(FB Interface) 的“静态”(Static) 下。
6. 如有需要，请在计数器指令中将类型从 INT 更改为其它类型之一。 计数器结构将相应更改。

计数器指令中显示的类型（对于预设值和计 FB 接口中显示的对应的结构类型数值）

INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

1. 打开将使用此 FB 的块。
2. 将此 FB 置于所需的位置。 如此将为该 FB 创建一个背景数据块。
3. 打开将 FB 放入编辑器时创建的背景数据块。
4. 在“静态”(Static) 下，找到所需的计数器结构。 在此计数器结构的“保持性”(Retain) 列中，选中相应框使该结构具有保持性。

## 7.4 比较

### 7.4.1 比较

表格 7-27 比较指令

LAD	FBD	SCL	说明
		<pre> out := in1 = in2;  Or  IF in1 = in2   THEN out := 1;   ELSE out := 0; END IF;                     </pre>	<p>比较数据类型相同的两个值。该 LAD 触点比较结果为 TRUE 时，则该触点会被激活。如果该 FBD 功能框比较结果为 TRUE，则功能框输出为 TRUE。</p>

1 对于 LAD 和 FBD：单击指令名称（如“==”），以从下拉列表中更改比较类型。单击“???”并从下拉列表中选择数据类型。

表格 7-28 参数的数据类型

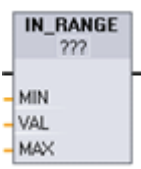
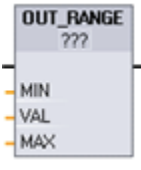
参数	数据类型	说明
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, String, Char, Time, DTL, 常数	要比较的值

表格 7-29 比较说明

关系类型	满足以下条件时比较结果为真 ...
=	IN1 等于 IN2
<>	IN1 不等于 IN2
>=	IN1 大于或等于 IN2
<=	IN1 小于或等于 IN2
>	IN1 大于 IN2
<	IN1 小于 IN2

## 7.4.2 范围内和范围外指令

表格 7-30 范围内和范围外指令

LAD/FBD	SCL	说明
	<pre>out := IN_RANGE(min, val, max);</pre>	测试输入值是在指定的值范围之内还是之外。 如果比较结果为 TRUE，则功能框输出为 TRUE。
	<pre>out := OUT_RANGE(min, val, max);</pre>	

1 对于 LAD 和 FBD：单击“???”并从下拉列表中选择数据类型。

表格 7-31 参数的数据类型

参数	数据类型 <sup>1</sup>	说明
MIN, VAL, MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, 常数	比较器输入

1 输入参数 MIN、VAL 和 MAX 的数据类型必须相同。

- 满足以下条件时 IN\_RANGE 比较结果为真：MIN ≤ VAL ≤ MAX
- 满足以下条件时 OUT\_RANGE 比较结果为真：VAL < MIN 或 VAL > MAX

### 7.4.3 OK 和 Not OK 指令

表格 7- 32 OK 和 Not OK 指令

LAD	FBD	SCL	说明
		不提供	测试输入数据参考是否为符合 IEEE 规范 754 的有效实数。
		不提供	

<sup>1</sup> 对于 LAD 和 FBD：如果该 LAD 触点为 TRUE，则激活该触点并传递能流。如果该 FBD 功能框为 TRUE，则功能框输出为 TRUE。

表格 7- 33 参数的数据类型

参数	数据类型	说明
IN	Real, LReal	输入数据

表格 7- 34 操作

指令	满足以下条件时 REAL 数测试结果为 TRUE:
OK	输入值为有效实数 <sup>1</sup>
NOT_OK	输入值不是有效实数 <sup>1</sup>

<sup>1</sup> 如果 Real 或 LReal 类型的值为 +/- INF（无穷大）、NaN（不是数字）或者非标准化的值，则其无效。非标准化的值是非常接近于 0 的数字。CPU 在计算中用 0 替换非标准化的值。

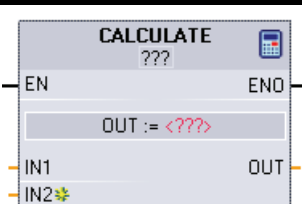
参见

LAD、FBD 和 SCL 的 EN 和 ENO (页 164)

## 7.5 数学

### 7.5.1 计算指令

表格 7-35 CALCULATE 指令

LAD/FBD	SCL	说明
	使用标准 SCL 数学表达式创建等式。	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。 <ul style="list-style-type: none"> <li>• 首先选择数据类型。所有输入和输出的数据类型必须相同。</li> <li>• 要添加其它输入, 请单击最后一个输入处的图标。</li> </ul>

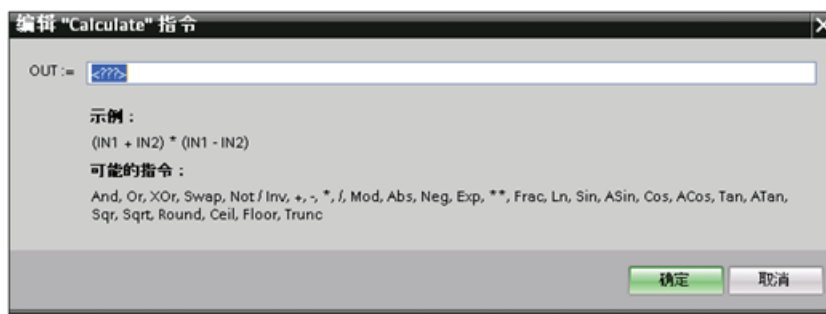
表格 7-36 参数的数据类型

参数	数据类型 <sup>1</sup>
IN1, IN2, ..INn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord

<sup>1</sup> 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。

编辑器底部显示有一个示例及可包含的数学运算列表。



**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**。

### 7.5.2 加法、减法、乘法和除法指令

表格 7-37 加法、减法、乘法和除法指令

LAD/FBD	SCL	说明
	<pre> out := in1 + in2; out := in1 - in2; out := in1 * in2; out := in1 / in2;                     </pre>	<ul style="list-style-type: none"> <li>• <b>ADD</b>: 加法 (<math>IN1 + IN2 = OUT</math>)</li> <li>• <b>SUB</b>: 减法 (<math>IN1 - IN2 = OUT</math>)</li> <li>• <b>MUL</b>: 乘法 (<math>IN1 * IN2 = OUT</math>)</li> <li>• <b>DIV</b>: 除法 (<math>IN1 / IN2 = OUT</math>)</li> </ul> <p>整数除法运算会截去商的小数部分以生成整数输出。</p>

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7-38 参数的数据类型（LAD 和 FBD）

参数	数据类型 <sup>1</sup>	说明
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, 常数	数学运算输入
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	数学运算输出

1 参数 IN1、IN2 和 OUT 的数据类型必须相同。



要添加 **ADD** 或 **MUL** 输入，请单击“创建”(Create) 图标，或在其中一个现有 **IN** 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，请在其中一个现有 **IN** 参数（多于两个原始输入时）的输入短线处单击右键，并选择“删除”(Delete) 命令。



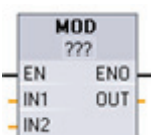
启用数学指令 (EN = 1) 后, 指令会对输入值 (IN1 和 IN2) 执行指定的运算并将结果存储在通过输出参数 (OUT) 指定的存储器地址中。运算成功完成后, 指令会设置 ENO = 1。

表格 7-39 ENO 状态

ENO	说明
1	无错误
0	数学运算结果值可能超出所选数据类型的有效数值范围。返回适合目标大小的结果的最低有效部分。
0	除数为 0 (IN2 = 0): 结果未定义, 返回 0。
0	Real/LReal: 如果其中一个输入值为 NaN (不是数字), 则返回 NaN。
0	ADD Real/LReal: 如果两个 IN 值均为 INF, 但符号不同, 则这是非法运算并返回 NaN。
0	SUB Real/LReal: 如果两个 IN 值均为 INF, 且符号相同, 则这是非法运算并返回 NaN。
0	MUL Real/LReal: 如果一个 IN 值为零而另一个为 INF, 则这是非法运算并返回 NaN。
0	DIV Real/LReal: 如果两个 IN 值均为零或 INF, 则这是非法运算并返回 NaN。

### 7.5.3 求模指令

表格 7-40 MOD 指令

LAD/FBD	SCL	说明
	<pre>out := in1 MOD in2;</pre>	<p>可以使用 MOD 指令返回整数除法运算的余数。用输入 IN1 的值除以输入 IN2 的值, 在输出 OUT 中返回余数。</p>

1 对于 LAD 和 FBD: 单击“???”并从下拉菜单中选择数据类型。

表格 7-41 参数的数据类型

参数	数据类型 <sup>1</sup>	说明
IN1 和 IN2	SInt, Int, DInt, USInt, UInt, UDIInt, 常数	求模输入
OUT	SInt, Int, DInt, USInt, UInt, UDIInt	求模输出

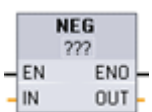
1 IN1、IN2 和 OUT 参数的数据类型必须相同。

表格 7- 42 ENO 值

ENO	说明
1	无错误
0	值 IN2 = 0, OUT 被赋以零值

### 7.5.4 取反指令

表格 7- 43 NEG 指令

LAD/FBD	SCL	说明
	<code>-(in);</code>	使用 NEG 指令可将参数 IN 的值的算术符号取反并将结果存储在参数 OUT 中。

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7- 44 参数的数据类型

参数	数据类型 <sup>1</sup>	说明
IN	SInt, Int, DInt, Real, LReal, Constant	数学运算输入
OUT	SInt, Int, DInt, Real, LReal	数学运算输出



1 参数 IN 和 OUT 的数据类型必须相同。

表格 7- 45 ENO 状态

ENO	说明
1	无错误
0	结果值超出所选数据类型的有效数值范围。 以 SInt 为例：NEG (-128) 的结果为 +128，超出该数据类型的最大值。

## 7.5.5 递增和递减指令

表格 7-46 INC 和 DEC 指令

LAD/FBD	SCL	说明
	<code>in_out := in_out + 1;</code>	递增有符号或无符号整数值： IN_OUT 值 +1 = IN_OUT 值
	<code>in_out := in_out - 1;</code>	递减有符号或无符号整数值： IN_OUT 值 - 1 = IN_OUT 值

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7-47 参数的数据类型

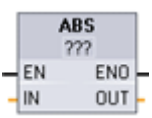
参数	数据类型	说明
IN/OUT	SInt, Int, DInt, USInt, UInt, UDIInt	数学运算输入和输出

表格 7-48 ENO 状态

ENO	说明
1	无错误
0	结果值超出所选数据类型的有效数值范围。 以 SInt 为例：INC (+127) 的结果为 +128，超出该数据类型的最大值。

## 7.5.6 绝对值指令

表格 7-49 ABS 指令

LAD/FBD	SCL	说明
	<code>out := ABS(in);</code>	计算参数 IN 的有符号整数或实数的绝对值并将结果存储在参数 OUT 中。

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7- 50 参数的数据类型

参数	数据类型 <sup>1</sup>	说明
IN	SInt, Int, DInt, Real, LReal	数学运算输入
OUT	SInt, Int, DInt, Real, LReal	数学运算输出

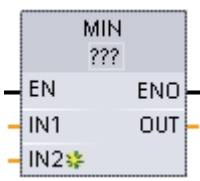
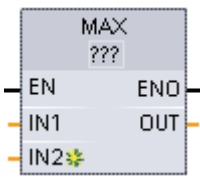
<sup>1</sup> 参数 IN 和 OUT 的数据类型必须相同。

表格 7- 51 ENO 状态

ENO	说明
1	无错误
0	数学运算结果值超出所选数据类型的有效数值范围。 以 SInt 为例：ABS (-128) 的结果为 +128，超出该数据类型最大值。

### 7.5.7 最小值和最大值指令

表格 7- 52 MIN 和 MAX 指令

LAD/FBD	SCL	说明
	<pre>out:= MIN(     in1:=_variant_in_,     in2:=_variant_in_     [...in32]);</pre>	MIN 指令用于比较两个参数 IN1 和 IN2 的值并将最小（较小）值分配给参数 OUT。
	<pre>out:= MAX(     in1:=_variant_in_,     in2:=_variant_in_     [...in32]);</pre>	MAX 指令用于比较两个参数 IN1 和 IN2 的值并将最大（较大）值分配给参数 OUT。

<sup>1</sup> 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7- 53 参数的数据类型

参数	数据类型 <sup>1</sup>	说明
IN1, IN2 [...IN32]	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, 常数	数学运算输入（最多 32 个输入）
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	数学运算输出

<sup>1</sup> IN1、IN2 和 OUT 参数的数据类型必须相同。



要添加输入，请单击“创建”(Create) 图标，或在其中一个现有 IN 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，请在其中一个现有 IN 参数（多于两个原始输入时）的输入短线处单击右键，并选择“删除”(Delete) 命令。

表格 7- 54 ENO 状态

ENO	说明
1	无错误
0	仅适用于 Real 数据类型： <ul style="list-style-type: none"> <li>一个或多个输入不是实数 (NaN)。</li> <li>结果 OUT 为 +/- INF（无穷大）。</li> </ul>

## 7.5.8 Limit 指令

表格 7- 55 LIMIT 指令

LAD/FBD	SCL	说明
	<pre>LIMIT (MN:= _variant_in_,         IN:= _variant_in_,         MX:= _variant_in_,         OUT:= _variant_out_);</pre>	Limit 指令用于测试参数 IN 的值是否在参数 MIN 和 MAX and if not, clamps the value at MIN or MAX. 指定的值范围内

<sup>1</sup> 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7- 56 参数的数据类型

参数	数据类型 <sup>1</sup>	说明
MN, IN 和 MX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, 常数	数学运算输入
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	数学运算输出

<sup>1</sup> 参数 MN、IN、MX 和 OUT 的数据类型必须相同。

如果参数 IN 的值在指定的范围内，则 IN 的值将存储在参数 OUT 中。如果参数 IN 的值超出指定的范围，则 OUT 值为参数 MIN 的值（如果 IN 值小于 MIN 值）或参数 MAX 的值（如果 IN 值大于 MAX 值）。

表格 7- 57 ENO 状态

ENO	说明
1	无错误
0	Real: 如果 MIN、IN 和 MAX 的一个或多个值是 NaN（不是数字），则返回 NaN。
0	如果 MIN 大于 MAX，则将值 IN 分配给 OUT。

SCL 示例:

- MyVal := LIMIT(MN:=10,IN:=53, MX:=40); //结果: MyVal = 40
- MyVal := LIMIT(MN:=10,IN:=37, MX:=40); //结果: MyVal = 37
- MyVal := LIMIT(MN:=10,IN:=8, MX:=40); //结果: MyVal = 10

## 7.5.9 浮点型算术运算指令

使用浮点指令可编写使用 Real 或 LReal 数据类型的数学运算程序:



- SQR: 平方 ( $IN^2 = OUT$ )
- SQRT: 平方根 ( $\sqrt{IN} = OUT$ )
- LN: 自然对数 ( $LN(IN) = OUT$ )
- EXP: 自然指数 ( $e^{IN} = OUT$ ), 其中底数  $e = 2.71828182845904523536$
- EXPT: 一般指数 ( $IN1^{IN2} = OUT$ )

EXPT 参数 IN1 和 OUT 的数据类型始终相同，必须为其选择 Real 或 LReal 类型。可以从众多数据类型中为指数参数 IN2 选择数据类型。

- FRAC: 分数（浮点数 IN 的小数部分 = OUT）

- SIN: 正弦 ( $\sin(\text{IN 弧度}) = \text{OUT}$ )  
ASIN: 反正弦 ( $\arcsine(\text{IN}) = \text{OUT 弧度}$ ), 其中  $\sin(\text{OUT 弧度}) = \text{IN}$
- COS: 余弦 ( $\cos(\text{IN 弧度}) = \text{OUT}$ )  
ACOS: 反余弦 ( $\arccos(\text{IN}) = \text{OUT 弧度}$ ), 其中  $\cos(\text{OUT 弧度}) = \text{IN}$
- TAN: 正切 ( $\tan(\text{IN 弧度}) = \text{OUT}$ )  
ATAN: 反正切 ( $\arctan(\text{IN}) = \text{OUT 弧度}$ ), 其中  $\tan(\text{OUT 弧度}) = \text{IN}$

表格 7- 58 浮点型数学运算指令示例

LAD/FBD	SCL	说明
	<code>out := SQR(in);</code> 或 <code>out := in * in;</code>	平方: $\text{IN}^2 = \text{OUT}$ 例如: 如果 $\text{IN} = 9$ , 则 $\text{OUT} = 81$ 。
	<code>out := in1 ** in2;</code>	普通指数: $\text{IN1}^{\text{IN2}} = \text{OUT}$ 例如: 如果 $\text{IN1} = 3$ 且 $\text{IN2} = 2$ , 则 $\text{OUT} = 9$ 。

- 1 对于 LAD 和 FBD: 单击“???” (指令名称旁), 并从下拉菜单中选择数据类型。
- 2 对于 SCL: 还可以使用基本的 SCL 数学运算符来创建数学表达式。

表格 7- 59 参数的数据类型

参数	数据类型	说明
IN, IN1	Real, LReal, Constant	输入
IN2	SInt, Int, DInt, USInt, UInt,UDInt, Real, LReal, Constant	EXPT 指数输入
OUT	Real, LReal	输出

表格 7- 60 ENO 状态

ENO	指令	条件	结果 (OUT)
1	全部	无错误	有效结果
0	SQR	结果超出有效 Real/LReal 范围	+INF
		IN 为 +/- NaN (不是数字)	+NaN
	SQRT	IN 为负数	-NaN

ENO	指令	条件	结果 (OUT)
		IN 为 +/- INF (无穷大) 或 +/- NaN	+/- INF 或 +/- NaN
	LN	IN 为 0.0、负数、-INF 或 -NaN	-NaN
		IN 为 +INF 或 +NaN	+INF 或 +NaN
	EXP	结果超出有效 Real/LReal 范围	+INF
		IN 为 +/- NaN	+/- NaN
	SIN, COS, TAN	IN 为 +/- INF 或 +/- NaN	+/- INF 或 +/- NaN
	ASIN, ACOS	IN 超出 -1.0 到 +1.0 的有效范围	+NaN
		IN 为 +/- NaN	+/- NaN
	ATAN	IN 为 +/- NaN	+/- NaN
	FRAC	IN 为 +/- INF 或 +/- NaN	+NaN
	EXPT	IN1 为 +INF 且 IN2 不是 -INF	+INF
		IN1 为负数或 -INF	如果 IN2 为 Real/LReal, 则为 +NaN, 否则为 -INF
		IN1 或 IN2 为 +/- NaN	+NaN
		IN1 为 0.0 且 IN2 为 Real/LReal (只能为 Real/LReal)	+NaN

## 7.6 移动

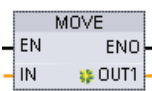
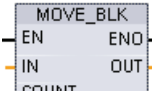
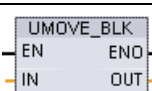
### 7.6.1 移动和块移动指令

使用移动指令可将数据元素复制到新的存储器地址并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 指令用于将单个数据元素从参数 IN 指定的源地址复制到参数 OUT 指定的目标地址。
- 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。



表格 7-61 MOVE、MOVE\_BLK 和 UMOVE\_BLK 指令

LAD/FBD	SCL	说明
	<code>out1 := in;</code>	将存储在指定地址的数据元素复制到新地址或多个地址。 <sup>1</sup>
	<pre>MOVE_BLK(     in:=_variant_in,     count:=_uint_in,     out=&gt;_variant_out);</pre>	将数据元素块复制到新地址的可中断移动。
	<pre>UMOVE_BLK(     in:=_variant_in,     count:=_uint_in,     out=&gt;_variant_out);</pre>	将数据元素块复制到新地址的不可中断移动。

<sup>1</sup> MOVE 指令：要在 LAD 或 FBD 中添加其它输出，请单击输出参数旁的“创建”(Create) 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。

表格 7-62 MOVE 指令的数据类型

参数	数据类型	说明
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, Array, Struct, DTL, Time	源地址
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, Array, Struct, DTL, Time	目标地址



要添加 MOVE 输出，请单击“创建”(Create) 图标，或右键单击现有 OUT 参数之一的输出短线，并选择“插入输出”(Insert output) 命令。

要删除输出，请在其中一个现有 OUT 参数（多于两个原始输出时）的输出短线处单击右键，并选择“删除”(Delete) 命令。

表格 7- 63 MOVE\_BLK 和 UMOVE\_BLK 指令的数据类型

参数	数据类型	说明
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal Byte, Word, DWord	源起始地址
COUNT	UInt	要复制的数据元素数
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord	目标起始地址

**说明****数据复制操作规则**

- 要复制 Bool 数据类型，请使用 SET\_BF、RESET\_BF、R、S 或 输出线圈 (LAD) (页 180)
- 要复制单个基本数据类型，请使用 MOVE
- 要复制基本数据类型数组，请使用 MOVE\_BLK 或 UMOVE\_BLK
- 要复制结构，请使用 MOVE
- 要复制字符串，请使用 S\_MOVE (页 266)
- 要复制字符串中的单个字符，请使用 MOVE
- MOVE\_BLK 和 UMOVE\_BLK 指令不能用于将数组或结构复制到 I、Q 或 M 存储区。

MOVE\_BLK 和 UMOVE\_BLK 指令在处理中断的方式上有所不同：

- 在 MOVE\_BLK 执行期间**排队并处理**中断事件。在中断 OB 子程序中未使用移动目标地址的数据时，或者虽然使用了该数据，但目标数据不必一致时，使用 MOVE\_BLK 指令。如果 MOVE\_BLK 操作被中断，则最后移动的一个数据元素在目标地址中是完整并且一致的。MOVE\_BLK 操作会在中断 OB 执行完成后继续执行。
- 在 UMOVE\_BLK 完成执行前**排队但不处理**中断事件。如果在执行中断 OB 子程序前移动操作必须完成且目标数据必须一致，则使用 UMOVE\_BLK 指令。更多信息，请参阅 数据一致性 (页 154)部分。

执行 MOVE 指令之后，ENO 始终为真。

表格 7- 64 ENO 状态

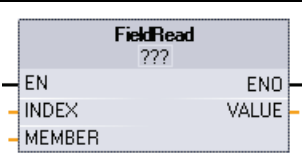

ENO	条件	结果
1	无错误	成功复制了全部的 COUNT 个元素。
0	源 (IN) 范围或目标 (OUT) 范围超出可用存储区。	复制适当的元素。不复制部分元素。

## 7.6.2 FieldRead 和 FieldWrite 指令

### 说明

STEP 7 V10.5 不支持数组索引或多维数组形式的变量引用。FieldRead 和 FieldWrite 指令曾用于为一维数组提供变量数组索引操作。STEP 7 V11 支持数组索引和多维数组形式的变量。STEP 7 V11 中包含了 FieldRead 和 FieldWrite，以便向后兼容使用了这两个指令的程序。

表格 7- 65 FieldRead 和 FieldWrite 指令

LAD/FBD	SCL	说明
	<pre>value := member[index];</pre>	FieldRead 用于从第一个元素由 MEMBER 参数指定的数组中读取索引值为 INDEX 的数组元素。数组元素的值将传送到 VALUE 参数指定的位置。
	<pre>member[index] : = value;</pre>	WriteField 用于将 VALUE 参数指定的位置上的值传送给第一个元素由 MEMBER 参数指定的数组。该值将传送给由 INDEX 参数指定数组索引的数组元素。

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7- 66 参数的数据类型

参数和类型		数据类型	说明
Index	输入	DInt	要读取或写入的数组元素的索引号
Member <sup>1</sup>	输入	数组元素类型： Bool、Byte、Word、DWord、 Char、SInt、Int、Dint、USInt、 UInt、UDInt、Real、LReal	在全局数据块或块接口中定义的一维数组的第一个元素的位置。 例如： 如果将数组索引指定为 [-2..4]， 则第一个元素的索引为 -2，而不是 0。
值 <sup>1</sup>	输出	Bool、Byte、Word、DWord、 Char、SInt、Int、Dint、USInt、 UInt、UDInt、Real、LReal	将指定的数组元素复制到位置 (FieldRead) 被复制到指定的数组元素的值的位置 (FieldWrite)

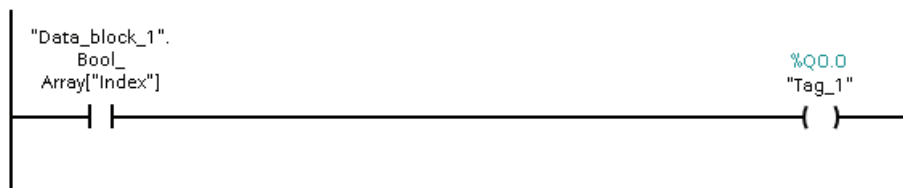
<sup>1</sup> MEMBER 参数和 VALUE 参数指定的数组元素的数据类型必须相同。

如果满足下列条件之一，则使能输出 ENO = 0:

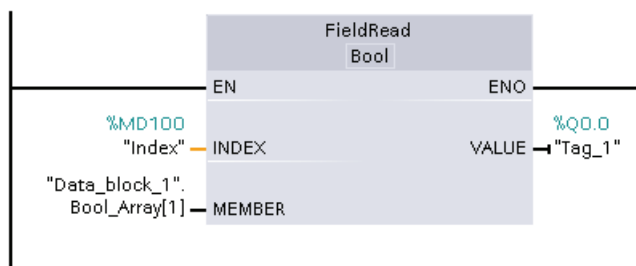
- EN 输入的信号状态为“0”
- 在 MEMBER 参数引用的数组中未定义 INDEX 参数指定的数组元素
- 处理过程中发生溢出之类的错误

### 通过数组索引访问数据

要通过变量访问数组中的元素， 仅需在程序逻辑中将该变量用作数组索引即可。 例如， 以下程序段中通过 PLC 变量“Index”引用的“Data\_block\_1”内布尔数组的布尔值来设置输出。



使用变量数组索引的逻辑结构与之前使用 FieldRead 指令的方法相同:



可以使用变量数组索引逻辑替换 FieldWrite 和 FieldRead 指令。

SCL 中没有 FieldRead 或 FieldWrite 指令，但支持通过变量对数组进行间接寻址：

```
#Tag_1 := "Data_block_1".Bool_Array[#Index];
```

### 7.6.3 填充指令

表格 7-67 FILL\_BLK 和 UFILL\_BLK 指令

LAD/FBD	SCL	说明
	<pre>FILL_BLK (     in:=_variant_in,     count:=int,     out=&gt;_variant_out);</pre>	可中断填充指令： 使用指定数据元素的副本填充地址范围
	<pre>UFILL_BLK (     in:=_variant_in,     count:=int    out =&gt;_variant_out);</pre>	不中断填充指令： 使用指定数据元素的副本填充地址范围

表格 7-68 参数的数据类型

参数	数据类型	说明
IN	SInt, Int, DIntT, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord	数据源地址
COUNT	USInt, UInt	要复制的数据元素数
OUT	SInt, Int, DIntT, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord	数据目标地址

说明

数据填充操作规则

- 要使用 BOOL 数据类型填充，请使用 SET\_BF、RESET\_BF、R、S 或输出线圈 (LAD)
- 要使用单个基本数据类型填充，请使用 MOVE
- 要使用基本数据类型填充数组，请使用 FILL\_BLK 或 UFILL\_BLK
- 要填充字符串中的单个字符，请使用 MOVE
- FILL\_BLK 和 UFILL\_BLK 指令不能用于将数组填充到 I、Q 或 M 存储区。

FILL\_BLK 和 UFILL\_BLK 指令可将源数据元素 IN 复制到通过参数 OUT 指定初始地址的目标中。复制过程不断重复并填充相邻的一组地址，直到副本数等于 COUNT 参数。

FILL\_BLK 和 UFILL\_BLK 指令在处理中断的方式上有所不同：

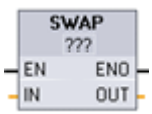
- 在 FILL\_BLK 执行期间**排队并处理**中断事件。在中断 OB 子程序中未使用移动目标地址的数据时，或者虽然使用了该数据，但目标数据不必一致时，使用 FILL\_BLK 指令。
- 在 UFILL\_BLK 完成执行前**排队但不处理**中断事件。如果在执行中断 OB 子程序前移动操作必须完成且目标数据必须一致，则使用 UFILL\_BLK 指令。

表格 7-69 ENO 状态

ENO	条件	结果
1	无错误	IN 元素成功复制到全部的 COUNT 个目标中。
0	目标 (OUT) 范围超出可用存储区	复制适当的元素。不复制部分元素。

### 7.6.4 交换指令

表格 7-70 SWAP 指令

LAD/FBD	SCL	说明
	<pre>out := SWAP(in);</pre>	用于反转二字节和四字节数据元素的字节顺序。不改变每个字节中的位顺序。执行 SWAP 指令之后，ENO 始终为 TRUE。

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7-71 参数的数据类型

参数	数据类型	说明
IN	Word, DWord	有序数据字节 IN
OUT	Word, DWord	反转有序数据字节 OUT

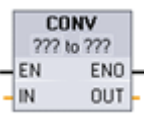
示例 1	参数 IN = MB0 (执行前)		参数 OUT = MB4, (执行后)	
地址	MB0	MB1	MB4	MB5
W#16#1234	12	34	34	12
WORD	MSB	LSB	MSB	LSB

示例 2	参数 IN = MB0 (执行前)				参数 OUT = MB4, (执行后)			
地址	MB0	MB1	MB2	MB3	MB4	MB5	MB6	MB7
DW#16# 12345678	12	34	56	78	78	56	34	12
DWORD	MSB			LSB	MSB			LSB

## 7.7 转换

### 7.7.1 CONV 指令

表格 7-72 转换 (CONV) 指令

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

- 1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。
- 2 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 7-73 参数的数据类型

参数	数据类型	说明
IN	位串 1, SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32	输入值
OUT	位串 1, SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCE32	转换为新数据类型的输入值

- 1 该指令不允许您选择位串 (Byte、Word、DWord)。要为指令参数输入数据类型 Byte、Word 或 DWord 的操作数，选择位长度相同的无符号整型。例如为 Byte 选择 USInt、为 Word 选择 UInt 或为 DWord 选择 UDInt。

选择 (转换源) 数据类型之后，(转换目标) 下拉列表中 will 显示可能的转换项列表。与 BCD16 进行转换仅限于 Int 数据类型。与 BCD32 进行转换仅限于 DInt 数据类型。

表格 7-74 ENO 状态

ENO	说明	结果 (OUT)
1	无错误	有效结果
0	IN 为 +/- INF 或 +/- NaN	+/- INF 或 +/- NaN
0	结果超出 OUT 数据类型的有效范围	OUT 被设置为 IN 的最低有效字节



## 7.7.2 SCL 的转换指令

## SCL 的转换指令

表格 7-75 从 Bool、Byte、Word 或 DWord 进行转换

数据类型	指令	结果
Bool	BOOL_TO_BYTE, BOOL_TO_WORD, BOOL_TO_DWORD, BOOL_TO_INT, BOOL_TO_DINT	值被传送到目标数据类型的最低有效位。
Byte	BYTE_TO_BOOL	最低位被传送到目标数据类型。
	BYTE_TO_WORD, BYTE_TO_DWORD	值被传送到目标数据类型的低位字节。
	BYTE_TO_SINT, BYTE_TO_USINT	值被传送到目标数据类型。
	BYTE_TO_INT, BYTE_TO_UINT, BYTE_TO_DINT, BYTE_TO_UDINT	值被传送到目标数据类型的最低有效字节。
Word	WORD_TO_BOOL	最低位被传送到目标数据类型。
	WORD_TO_BYTE	源值的低位字节被传送到目标数据类型。
	WORD_TO_DWORD	值被传送到目标数据类型的低位字节。
	WORD_TO_SINT, WORD_TO_USINT	源值的低位字节被传送到目标数据类型。
	WORD_TO_INT, WORD_TO_UINT	值被传送到目标数据类型。
	WORD_TO_DINT, WORD_TO_UDINT	值被传送到目标数据类型的低位字节。
DWord	DWORD_TO_BOOL	最低位被传送到目标数据类型。
	DWORD_TO_BYTE, DWORD_TO_WORD, DWORD_TO_SINT, DWORD_TO_USINT, DWORD_TO_INT, DWORD_TO_UINT	源值的低位字节被传送到目标数据类型。
	DWORD_TO_DINT, DWORD_TO_UDINT, DWORD_TO_REAL	值被传送到目标数据类型。

表格 7-76 从短整型 (SInt 或 USInt) 进行转换

数据类型	指令	结果
SInt	SINT_TO_BOOL	最低位被传送到目标数据类型。
	SINT_TO_BYTE	值被传送到目标数据类型
	SINT_TO_WORD, SINT_TO_DWORD, SINT_TO_INT, SINT_TO_DINT	值被传送到目标数据类型的低位字节。

7.7 转换

数据类型	指令	结果
	SINT_TO_USINT, SINT_TO_UINT, SINT_TO_UDINT, SINT_TO_REAL, SINT_TO_LREAL, SINT_TO_CHAR, SINT_TO_STRING	值被转换。
USInt	USINT_TO_BOOL	最低位被传送到目标数据类型。
	USINT_TO_BYTE	值被传送到目标数据类型
	USINT_TO_WORD, USINT_TO_DWORD, USINT_TO_INT, USINT_TO_UINT, USINT_TO_DINT, USINT_TO_UDINT	值被传送到目标数据类型的低位字节。
	USINT_TO_SINT, USINT_TO_REAL, USINT_TO_LREAL, USINT_TO_CHAR, USINT_TO_STRING	值被转换。

表格 7-77 从整型 (Int 或 UInt) 进行转换

数据类型	指令	结果
Int	INT_TO_BOOL	最低位被传送到目标数据类型。
	INT_TO_BYTE, INT_TO_DWORD, INT_TO_SINT, INT_TO_USINT, INT_TO_UINT, INT_TO_UDINT, INT_TO_REAL, INT_TO_LREAL, INT_TO_CHAR, INT_TO_STRING	值被转换。
	INT_TO_WORD	值被传送到目标数据类型。
	INT_TO_DINT	值被传送到目标数据类型的低位字节。
UInt	UINT_TO_BOOL	最低位被传送到目标数据类型。
	UINT_TO_BYTE, UINT_TO_SINT, UINT_TO_USINT, UINT_TO_INT, UINT_TO_REAL, UINT_TO_LREAL, UINT_TO_CHAR, UINT_TO_STRING	值被转换。
	UINT_TO_WORD, UINT_TO_DATE	值被传送到目标数据类型。
	UINT_TO_DWORD, UINT_TO_DINT, UINT_TO_UDINT	值被传送到目标数据类型的低位字节。

表格 7-78 从双整型 (Dint 或 UDInt) 进行转换

数据类型	指令	结果
Dint	DINT_TO_BOOL	最低位被传送到目标数据类型。
	DINT_TO_BYTE, DINT_TO_WORD, DINT_TO_SINT, DINT_TO_USINT, DINT_TO_INT, DINT_TO_UINT, DINT_TO_UDINT, DINT_TO_REAL, DINT_TO_LREAL, DINT_TO_CHAR, DINT_TO_STRING	值被转换。
	DINT_TO_DWORD, DINT_TO_TIME	值被传送到目标数据类型。
UDInt	UDINT_TO_BOOL	最低位被传送到目标数据类型。
	UDINT_TO_BYTE, UDINT_TO_WORD, UDINT_TO_SINT, UDINT_TO_USINT, UDINT_TO_INT, UDINT_TO_UINT, UDINT_TO_DINT, UDINT_TO_REAL, UDINT_TO_LREAL, UDINT_TO_CHAR, UDINT_TO_STRING	值被转换。
	UDINT_TO_DWORD, UDINT_TO_TOD	值被传送到目标数据类型。

表格 7-79 从实数 (Real 或 LReal) 进行转换

数据类型	指令	结果
Real	REAL_TO_DWORD, REAL_TO_LREAL	值被传送到目标数据类型。
	REAL_TO_SINT, REAL_TO_USINT, REAL_TO_INT, REAL_TO_UINT, REAL_TO_DINT, REAL_TO_UDINT, REAL_TO_STRING	值被转换。
LReal	LREAL_TO_SINT, LREAL_TO_USINT, LREAL_TO_INT, LREAL_TO_UINT, LREAL_TO_DINT, LREAL_TO_UDINT, LREAL_TO_REAL, LREAL_TO_STRING	值被转换。

表格 7-80 从 Time、DTL、TOD 或 Date 进行转换



数据类型	指令	结果
Time	TIME_TO_DINT	值被传送到目标数据类型。
DTL	DTL_TO_DATE, DTL_TO_TOD	值被转换。
TOD	TOD_TO_UDINT	值被转换。
Date	DATE_TO_UINT	值被转换。

表格 7- 81 从 Char 或 String 进行转换

数据类型	指令	结果
Char	CHAR_TO_SINT, CHAR_TO_USINT, CHAR_TO_INT, CHAR_TO_UINT, CHAR_TO_DINT, CHAR_TO_UDINT	值被转换。
	CHAR_TO_STRING	值被传送到字符串的第一个字符。
String	STRING_TO_SINT, STRING_TO_USINT, STRING_TO_INT, STRING_TO_UINT, STRING_TO_DINT, STRING_TO_UDINT, STRING_TO_REAL, STRING_TO_LREAL	值被转换。
	STRING_TO_CHAR	字符串的第一个字符被复制到 Char。

### 7.7.3 取整和截取指令

表格 7- 82 ROUND 和 TRUNC 指令

LAD/FBD	SCL	说明
	<pre>out := ROUND (in);</pre>	<p>将实数转换为整数。默认数据类型为 DINT。当输出为除 DINT 以外的有效数据类型时，必须显式声明；例如，ROUND_REAL 或 ROUND_LREAL。</p> <p>实数的小数部分舍入为最接近的整数值（IEEE - 舍入为最接近值）。如果该数值刚好是两个连续整数的一半（例如，10.5），则将其取整为偶数。例如：</p> <ul style="list-style-type: none"> <li>• ROUND (10.5) = 10</li> <li>• ROUND (11.5) = 12</li> </ul>
	<pre>out := TRUNC(in);</pre>	<p>TRUNC 用于将实数转换为整数。实数的小数部分被截成零（IEEE - 取整为零）。</p>

1 对于 LAD 和 FBD：单击“???”（指令名称旁），并从下拉菜单中选择数据类型。

表格 7- 83 参数的数据类型

参数	数据类型	说明
IN	Real, LReal	浮点型输入
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	取整或截取后的输出

表格 7- 84 ENO 状态

ENO	说明	结果 (OUT)
1	无错误	有效结果
0	IN 为 +/- INF 或 +/- NaN	+/- INF 或 +/- NaN

### 7.7.4 上取整和下取整指令

表格 7- 85 CEIL 和 FLOOR 指令

LAD/FBD	SCL	说明
	<code>out := CEIL(in);</code>	将实数 (Real 或 LReal) 转换为大于或等于所选实数的最小整数 (IEEE“向正无穷取整”)。
	<code>out := FLOOR(in);</code>	将实数 (Real 或 LReal) 转换为小于或等于所选实数的最大整数 (IEEE“向负无穷取整”)。

1 对于 LAD 和 FBD: 单击“???” (指令名称旁), 并从下拉菜单中选择数据类型。

表格 7- 86 参数的数据类型

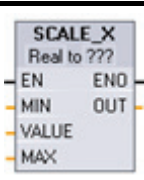
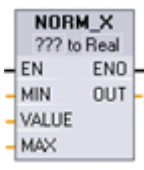
参数	数据类型	说明
IN	Real, LReal	浮点型输入
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	转换后的输出

表格 7- 87 ENO 状态

ENO	说明	结果 (OUT)
1	无错误	有效结果
0	IN 为 +/- INF 或 +/- NaN	+/- INF 或 +/- NaN

### 7.7.5 标定和标准化指令

表格 7-88 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，<math>0.0 \leq \text{VALUE} \leq 1.0</math>）进行标定：</p> $\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}$
	<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> $\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN}),$ <p>其中 <math>(0.0 \leq \text{OUT} \leq 1.0)</math></p>

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7-89 参数的数据类型

参数	数据类型 <sup>1</sup>	说明
MIN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	输入范围的最小值
VALUE	SCALE_X: Real, LReal NORM_X: SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	要标定或标准化的输入值
MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	输入范围的最大值
OUT	SCALE_X: SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal NORM_X: Real, LReal	标定或标准化后的输出值

1 对于 SCALE\_X：参数 MIN、MAX 和 OUT 的数据类型必须相同。  
对于 NORM\_X：参数 MIN、VALUE 和 MAX 的数据类型必须相同。

**说明****SCALE\_X 参数 VALUE 应限制为 (0.0 <= VALUE <= 1.0)**

如果参数 VALUE 小于 0.0 或大于 1.0:

- 线性标定运算会生成一些小于 MIN 参数值或大于 MAX 参数值的 OUT 值，作为 OUT 值，这些数值在 OUT 数据类型值范围内。此时，SCALE\_X 执行会设置 ENO = TRUE。
- 还可能会生成一些不在 OUT 数据类型值范围内的标定数值。此时，OUT 参数值会被设置为一个中间值，该中间值等于被标定实数在最终转换为 OUT 数据类型之前的最低有效部分。在这种情况下，SCALE\_X 执行会设置 ENO = FALSE。

**NORM\_X 参数 VALUE 应限制为 (MIN <= VALUE <= MAX)**

如果参数 VALUE 小于 MIN 或大于 MAX，线性标定运算会生成小于 0.0 或大于 1.0 的标准化 OUT 值。在这种情况下，NORM\_X 执行会设置 ENO = TRUE。

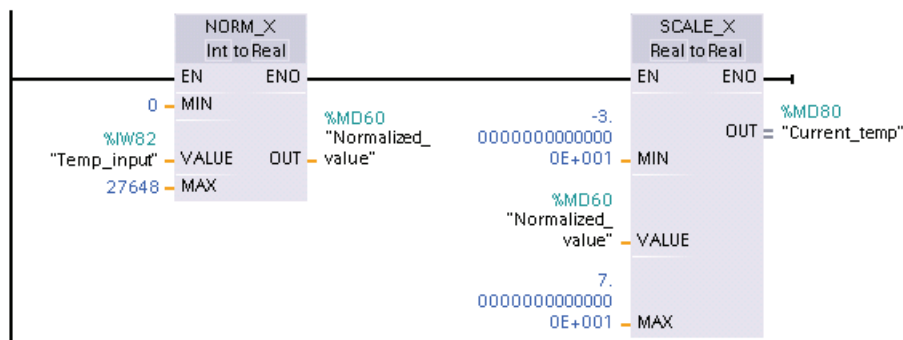
表格 7- 90 ENO 状态

ENO	条件	结果 (OUT)
1	无错误	有效结果
0	结果超出 OUT 数据类型的有效范围	中间结果：实数在最终转换为 OUT 数据类型前的最低有效部分。
0	参数 MAX <= MIN	SCALE_X: 用实数 VALUE 的最低有效部分填充 OUT 大小。 NORM_X: 扩展 VALUE 数据类型的 VALUE 来填充双字大小。
0	参数 VALUE = +/- INF 或 +/- NaN	VALUE 写入 OUT

**示例 (LAD): 标准化和标定模拟量输入值**

来自电流输入型模拟量信号模块或信号板的模拟量输入的有效值在 0 到 27648 范围内。假设模拟量输入代表温度，其中模拟量输入值 0 表示 -30.0 摄氏度，27648 表示 70.0 摄氏度。

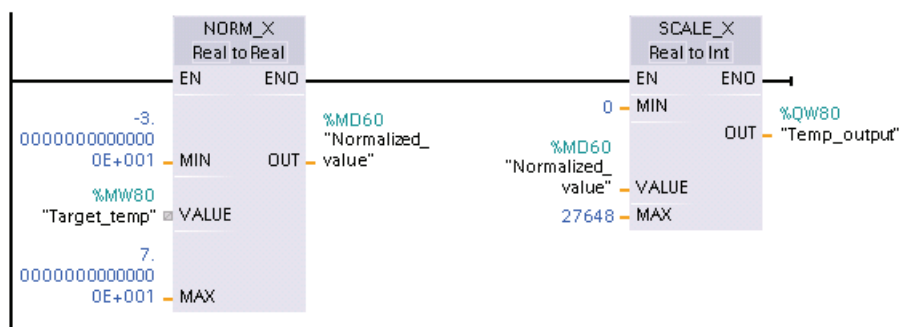
要将模拟值转换为对应的工程单位，应将输入标准化为 0.0 到 1.0 之间的值，然后再将其标定为 -30.0 到 70.0 之间的值。结果值是用模拟量输入（以摄氏度为单位）表示的温度：



请注意，如果模拟量输入来自电压型模拟量信号模块或信号板，则 NORM\_X 指令的 MIN 值是 -27648，而不是 0。

**示例 (LAD): 标准化和标定模拟量输出值**

要在电流输出型模拟量信号模块或信号板中设置的模拟量输出的有效值必须在 0 到 27648 范围内。假设模拟量输出表示温度设置，其中模拟量输入值 0 表示 -30.0 摄氏度，27648 表示 70.0 摄氏度。要将存储器中的温度值（范围是 -30.0 到 70.0）转换为 0 到 27648 范围内的模拟量输出值，必须将以工程单位表示的值标准化为 0.0 到 1.0 之间的值，然后将其标定为 0 到 27648 范围内的模拟量输出值：



请注意，如果模拟量输出应用到电压型模拟量信号模块或信号板，则 SCALE\_X 指令的 MIN 值是 -27648，而不是 0。

有关电压和电流形式的模拟量输出表示法 (页 845)和模拟量输出表示法 (页 846)的详细信息，请参见技术规范。



## 7.8 程序控制

### 7.8.1 SCL 程序控制语句概述

结构化控制语言 (SCL, Structured Control Language) 提供三类用于结构化用户程序的程序控制语句：

- 选择语句：选择语句可将程序执行转移到备选语句序列。
- 循环：可以使用迭代语句控制循环执行。迭代语句指定应根据某些条件重复执行的程序部分。
- 程序跳转：程序跳转是指立刻跳转到特定的跳转目标，因而跳转到同一块内的其它语句。

这些程序控制语句都使用 PASCAL 编程语言的语法。

表格 7- 91 SCL 程序控制语句类型

程序控制语句	说明	
选择	IF-THEN 语句 (页 230)	用将程序执行转移到两个备选分支之一（取决于条件为 TRUE 还是 FALSE）
	CASE 语句 (页 231)	用于选择执行 $n$ 个备选分支之一（取决于变量值）
循环	FOR 语句 (页 232)	只要控制变量在指定值范围内，就重复执行某一语句序列
	WHILE-DO 语句 (页 233)	只要仍满足执行条件，就重复执行某一语句序列
	REPEAT-UNTIL 语句 (页 234)	重复执行某一语句序列，直到满足终止条件为止
程序跳转	CONTINUE 语句 (页 235)	停止执行当前循环迭代
	EXIT 语句 (页 236)	无论是否满足终止条件，都会随时退出循环
	GOTO 语句 (页 236)	使程序立即跳转到指定标签
	IF-THEN 语句 (页 230)	使程序立刻退出正在执行的块，返回到调用块

参见

RETURN 语句 (页 237)

## 7.8.2 IF-THEN 语句

IF-THEN 语句是条件语句，可控制程序流，根据逻辑表达式的 Bool 值的结果决定是否执行一组语句。您还可以使用括号嵌套或结构化多条 IF-THEN 语句的执行。

表格 7- 92 IF-THEN 语句的元素

SCL	说明
<pre>IF "condition" THEN     statement_A;     statement_B;     statement_C; ;</pre>	<p>如果“condition”为 TRUE 或 1，则执行后面的语句，直到遇到 END_IF 语句为止。</p> <p>如果“condition”为 FALSE 或 0，则跳转到 END_IF 语句（除非程序包含可选的 ELSIF 或 ELSE 语句）。</p>
<pre>[ELSIF "condition-n" THEN     statement_N; ;]</pre>	<p>可选的 ELSEIF<sup>1</sup> 语句提供其它要评估的条件。例如：如果 IF-THEN 语句中的“condition”为 FALSE，则程序将评估“condition-n”。如果“condition-n”为 TRUE，则执行“statement_N”。</p>
<pre>[ELSE     statement_X; ;]</pre>	<p>可选的 ELSE 语句提供 IF-THEN 语句的“condition”为 FALSE 时将要执行的语句。</p>
<pre>END_IF;</pre>	<p>END_IF 语句用于终止 IF-THEN 指令。</p>

<sup>1</sup> 可以在一条 IF-THEN 语句中包含多条 ELSIF 语句。

表格 7- 93 IF-THEN 语句的变量

变量	说明
“condition”	必需。逻辑表达式为 TRUE (1) 或 FALSE (0)。
“statement_A”	可选。“condition”为 TRUE 时要执行的一条或多条语句。
“condition-n”	可选。可选 ELSIF 语句要评估的逻辑表达式。
“statement_N”	可选。ELSIF 语句的“condition-n”为 TRUE 时要执行的一条或多条语句。
“statement_X”	可选。IF-THEN 语句的“condition”为 FALSE 时要执行的一条或多条语句。

IF 语句按照下列规则执行：

- 执行第一个逻辑表达式为 TRUE 的语句序列。不执行其余语句序列。
- 如果无布尔型表达式为 TRUE，则执行 ELSE 引入的语句序列（或者，如果 ELSE 分支不存在，则不执行语句序列）。
- 不限制 ELSIF 语句的数量。

#### 说明

与 IF 语句相比，使用一个或多个 ELSIF 分支存在一定的优势，就是不用再评估有效表达式后面的逻辑表达式。从而，可缩短程序的运行时间。

### 7.8.3 CASE 语句

表格 7-94 CASE 语句的元素

SCL	说明
<pre> CASE "Test_Value" OF   "ValueList": Statement[; Statement, ...]   "ValueList": Statement[; Statement, ...] [ELSE Else-statement[; Else-statement, ...]] END_CASE;</pre>	CASE 语句根据表达式的值来选择执行多组语句中的一组。

表格 7-95 参数

参数	说明
"Test_Value"	必需。任何 Int 数据类型的数字表达式
"ValueList"	必需。单个值、或逗号分隔的值或值范围的列表。（使用两个句点定义值范围：2..8）下例说明了不同变型的值列表： 1: Statement_A; 2, 4: Statement_B; 3, 5..7,9: Statement_C;
Statement	必需。“Test_Value”与值列表中任何一个值匹配时执行的一条或多条语句
Else-statement	可选。与“ValueList”中的任何一个值都不匹配时执行的一条或多条语句

CASE 语句按照下列规则执行：

- 选择表达式必须返回一个 Int 类型的值。
- 处理 CASE 语句时，程序会检查选择表达式的值是否包含在指定的值列表中。如果找到匹配项，则执行分配给该列表的语句成分。
- 如果未找到匹配项，则执行 ELSE 后面的程序段，如果不存在 ELSE 分支，则不执行任何语句。

CASE 语句可以嵌套使用。每个嵌套的 CASE 语句必须具有相关联的 END\_CASE 语句。

```

CASE var1 OF
    1 : var2 := "A";
    2 : var2 := "B";
ELSE
    CASE var3 OF
        65..90: var2 := "UpperCase";
        97..122: var2 := "LowerCase";
    ELSE
        var2:= "SpecialCharacter";
    END_CASE;
END_CASE;

```

### 7.8.4 FOR 语句

表格 7-96 FOR 语句的元素

SCL	说明
<pre> FOR "control_variable" := "begin" TO "end" [BY "increment"] DO     statement; ; END_FOR; </pre>	<p>FOR 语句用于在控制变量处于指定的值范围内时重复执行某一语句序列。使用 FOR 定义循环时需要指定初始值和最终值。这两个值的数据类型必须与控制变量的相同。</p>

表格 7-97 参数

参数	说明
"control_variable"	必需。整型 (Int 或 DInt)，用作循环计数器
"begin"	必需。指定控制变量初始值的简单表达式

参数	说明
“end”	必需。确定控制变量最终值的简单表达式
“increment”	可选。每次循环后“control variable”的变化量。“increment”与“control variable”具有相同的数据类型。如果未指定“increment”的值，则每次循环之后，运行变量的值加 1。不能在执行 FOR 语句期间更改“increment”。

FOR 语句的执行方式如下：

- 循环开始时，控制变量设置为初始值（初始分配），每次重复进行循环时，控制变量会增加指定增量（正增量）或减少指定增量（负增量），直至达到最终值。
- 每次执行完循环之后，会检查该条件（达到最终值）以确定是否满足该条件。如果满足条件，则执行语句序列，否则将跳过该循环及其语句序列。

定义 FOR 语句的规则：

- 控制变量的数据类型只能是 Int 或 DInt。
- 可以省略语句 BY [increment]。如果未指定增量，则自动默认为 +1。

要结束循环而不考虑“condition”表达式的状态，请使用 EXIT 语句 (页 236)。EXIT 语句将执行紧随 END\_FOR 语句之后的语句。

使用 CONTINUE 语句 (页 235) 可跳过某个 FOR 循环的后续语句，并继续执行循环，同时检查是否满足终止条件。

### 7.8.5 WHILE-DO 语句

表格 7-98 WHILE 语句

SCL	说明
<pre>WHILE "condition" DO     Statement;     Statement;     ...; END WHILE;</pre>	<p>WHILE 语句执行一系列语句，直到给定条件为 TRUE。</p> <p>可以嵌套使用 WHILE 循环。END_WHILE 语句与最后执行的 WHILE 指令配对。</p>

表格 7-99 参数

参数	说明
“condition”	必需。值为 TRUE 或 FALSE 的逻辑表达式。（“null”条件被视为 FALSE。）
Statement	可选。在条件值为 TRUE 之前执行的一条或多条语句。

**说明**

WHILE 语句先评估“condition”的状态，然后执行语句。要执行语句一次或多次而不考虑“condition”的状态，请使用 REPEAT 语句。

WHILE 语句按照下列规则执行：

- 每次循环执行循环体之前，评估执行条件。
- 只要执行条件的值为 TRUE，就重复执行 DO 后面的循环体。
- 一旦值变为 FALSE，则立即跳过循环，去执行循环后面的语句。

要结束循环而不考虑“condition”表达式的状态，请使用 EXIT 语句 (页 236)。EXIT 语句执行紧跟 END\_WHILE 语句的语句

使用 CONTINUE 语句可跳过 WHILE 循环后面的语句，并在检查是否满足终止条件后决定是否继续执行循环。

### 7.8.6 REPEAT-UNTIL 语句

表格 7- 100 REPEAT 指令

SCL	说明
<pre>REPEAT     Statement; ; UNTIL "condition" END_REPEAT;</pre>	REPEAT 语句执行一组语句，直到给定条件为 TRUE。 可以嵌套使用 REPEAT 循环。END_REPEAT 语句始终与最后执行的 Repeat 指令配对。

表格 7- 101 参数

参数	说明
Statement	可选。在条件值为 TRUE 之前执行的一条或多条语句。
"condition"	必需。一个或多个用以下两种方式表达的表达式：值为 TRUE 或 FALSE 的数字表达式或字符串表达式。“null”条件被视为 FALSE。

**说明**

在循环的首次迭代过程中，REPEAT 语句在执行相关语句（即使“condition”为 FALSE）后评估“condition”的状态。要在执行这些语句前查看“condition”的状态，请使用 WHILE 语句。

要结束循环而不考虑“condition”表达式的状态，请使用 EXIT 语句 (页 236)。EXIT 语句执行紧跟 END\_REPEAT 语句的语句

使用 CONTINUE 语句 (页 235) 可跳过某个 REPEAT 循环的后续语句，并继续执行循环，同时检查是否满足终止条件。

## 7.8.7 CONTINUE 语句

表格 7- 102 CONTINUE 语句

SCL	说明
<pre>CONTINUE   Statement; ;</pre>	<p>CONTINUE 语句跳过程序循环（FOR、WHILE、REPEAT）后面的语句，并在检查是否满足终止条件后决定是否继续执行循环。如果不满足，则继续执行循环。</p>

CONTINUE 语句按照下列规则执行：

- 该语句立即终止循环体的执行。
- 根据是否满足重复执行循环的条件，决定是再次执行循环体还是退出迭代语句而去执行紧随其后的语句。
- 在 FOR 语句中，在执行 CONTINUE 语句后控制变量立即增加指定的增量。

只能在循环中使用 CONTINUE 语句。在嵌套循环中，CONTINUE 总是立即指向包含它的循环。CONTINUE 通常与 IF 语句一起使用。

如果要退出循环而不考虑终止测试情况，请使用 EXIT 语句。

下例说明了使用 CONTINUE 语句来避免计算值的百分数时发生被 0 除的错误：

```
FOR x = 0 TO 10 DO
  IF value[i] = 0 THEN CONTINUE; END_IF;
  p := part / value[i] * 100;
  s := INT_TO_STRING(p);
  percent=CONCAT(IN1:=s, IN2:="%");
END_FOR;
```

### 7.8.8 EXIT 语句

表格 7- 103 EXIT 指令

SCL	说明
<b>EXIT;</b>	EXIT 语句用于随时退出循环（FOR、WHILE 或 REPEAT），而不考虑是否满足终止条件。

EXIT 语句按照下列规则执行：

- 该语句会立即退出该退出语句所处的重复语句。
- 继续执行该循环后面（例如 END\_FOR 之后）的程序。

在循环中使用 EXIT 语句。在嵌套循环中，EXIT 语句将处理权返回到下一更高嵌套级。

```
FOR i = 0 TO 10 DO
CASE 值[i, 0] OF
  1..10: 值 [i, 1]:="A";
  11..40: 值 [i, 1]:="B";
  41..100: 值 [i, 1]:="C";
ELSE
EXIT;
END_CASE;
END_FOR;
```

### 7.8.9 GOTO 语句

表格 7- 104 GOTO 语句

SCL	说明
<b>GOTO JumpLabel;</b> <b>Statement;</b> <b>... ;</b> <b>JumpLabel: Statement;</b>	GOTO 语句通过跳转到同一块中的某个标签来跳过语句。 跳转标签（“JumpLabel”）和 GOTO 语句必须在同一个块中。跳转标签的名称只能在块中分配一次。每个跳转标签都可以是多条 GOTO 语句的跳转目标。

不能跳转到循环部分（FOR、WHILE 或 REPEAT）。可以在循环中进行跳转。

在以下示例中：根据操作数“Tag\_value”的值在对应跳转标签定义的位置继续执行程序。

如果“Tag\_value”等于 2，则会在跳转标签“MyLabel2”位置继续执行，并会跳过“MyLabel1”。

```
CASE "Tag_value" OF
  1 : GOTO MyLabel1;
  2 : GOTO MyLabel2;
ELSE GOTO MyLabel3;
END_CASE;
```



```

MyLabel1: "Tag_1" := 1;
MyLabel2: "Tag_2" := 1;
MyLabel3: "Tag_4" := 1;

```

### 7.8.10 RETURN 语句

表格 7- 105 RETURN 指令

SCL	说明
RETURN;	Return 指令用于无条件退出正在执行的代码块。程序执行返回到调用块或操作系统（退出 OB 时）。

RETURN 指令示例：  
IF "Error" <> 0 THEN  
RETURN;  
END\_IF;

#### 说明

执行最后一条指令后，代码块自动返回到调用块。不要在代码块末尾插入 RETURN 指令。

### 7.8.11 跳转和标签指令

表格 7- 106 JMP、JMPN 和 LABEL 指令

LAD	FBD	SCL	说明
		请参见 GOTO (页 236) 语句。	如果有能流通过 JMP 线圈 (LAD)，或者 JMP 功能框的输入为真 (FBD)，则程序将从指定标签后的第一条指令继续执行。
			如果没有能流通过 JMPN 线圈 (LAD)，或者 JMPN 功能框的输入为假 (FBD)，则程序将从指定标签后的第一条指令继续执行。
			JMP 或 JMPN 跳转指令的目标标签。

- 1 通过在 LABEL 指令中直接键入来创建标签名称。可以使用参数助手图标来选择 JMP 和 JMPN 标签名称字段可用的标签名称。也可在 JMP 或 JMPN 指令中直接键入标签名称。

7.8 程序控制

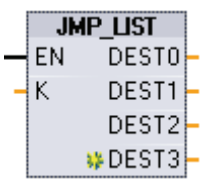
表格 7- 107 参数的数据类型

参数	数据类型	说明
Label_name	标签标识符	跳转指令以及相应跳转目标程序标签的标识符

- 各标签在代码块内必须唯一。
- 可以在代码块中进行跳转，但不能从一个代码块跳转到另一个代码块。
- 可以向前或向后跳转。
- 可以在同一代码块中从多个位置跳转到同一标签。

7.8.12 JMP\_LIST 指令

表格 7- 108 JMP\_LIST 指令

LAD/FBD	SCL	说明
	<pre> CASE k OF   0: GOTO dest0;   1: GOTO dest1;   2: GOTO dest2;   [n: GOTO destn;] END_CASE;         </pre>	<p>JMP_LIST 指令用作程序跳转分配器，控制程序段的执行。根据 K 输入的值跳转到相应的程序标签。程序从目标跳转标签后面的程序指令继续执行。如果 K 输入的值超过（标签数 - 1），则不进行跳转，继续处理下一程序段。</p>

表格 7- 109 参数的数据类型

参数	数据类型	说明
K	UInt	跳转分配器控制值
DEST0, DEST1, ..., DESTn.	程序标签	与特定 K 参数值对应的跳转目标标签： 如果 K 的值等于 0，则跳转到分配给 DEST0 输出的程序标签。 如果 K 的值等于 1，则跳转到分配给 DEST1 输出的程序标签，以此类推。如果 K 输入的值超过（标签数 - 1），则不进行跳转，继续处理下一程序段。

对于 LAD 和 FBD：在程序中第一次放置 JMP\_LIST 功能框，该功能框有两个跳转标签输出。可以添加或删除跳转目标。



单击功能框内的创建图标（位于最后一个 DEST 参数的左侧）可添加新的跳转标签输出。



- 右键单击输出短线，并选择“插入输出”(Insert output) 命令。
- 右键单击输出短线，并选择“删除”(Delete) 命令。

### 7.8.13 SWITCH 指令

表格 7- 110 SWITCH 指令

LAD/FBD	SCL	说明
	不提供	SWITCH 指令用作程序跳转分配器，控制程序段的执行。根据 K 输入的值与分配给指定比较输入的值的比较结果，跳转到与第一个为“真”的比较测试相对应的程序标签。如果比较结果都不为 TRUE，则跳转到分配给 ELSE 的标签。程序从目标跳转标签后面的程序指令继续执行。

- 1 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。
- 2 对于 SCL：使用 IF-THEN 语句进行比较。

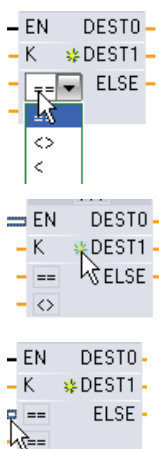
表格 7- 111 参数的数据类型

参数	数据类型 <sup>1</sup>	说明
K	SInt、Int、DInt、USInt、UInt、UDInt、Real、LReal、Byte、Word、DWord、Time、TOD、Date	常用比较值输入
==, <>, <, <=, >, >=	SInt、Int、DInt、USInt、UInt、UDInt、Real、LReal、Byte、Word、DWord、Time、TOD、Date	分隔比较值输入，获得特定比较类型
DEST0、DEST1, ..., DESTn。ELSE	程序标签	与特定比较对应的跳转目标标签： 首先处理 K 输入下面的第一个比较输入，如果 K 值与该输入的比较结果为“真”，则跳转到分配给 DEST0 的标签。下一比较测试使用接下来的下一个输入，如果比较结果“真”，则跳转到分配给 DEST1 的标签。依次对其它比较进行类似的处理，如果比较结果都不为“真”，则跳转到分配给 ELSE 输出的标签。

<sup>1</sup> K 输入和比较输入 (==, <>, <, <=, >, >=) 的数据类型必须相同。

### 添加输入、删除输入和指定比较类型

在程序中第一次放置 LAD 或 FBD SWITCH 功能框时，该功能框有两个比较输入。可以分配比较类型以及添加输入/跳转目标，如下所示。



单击功能框内的比较运算符，并从下拉列表中选择新运算符。

单击功能框中的创建图标（位于最后一个 DEST 参数的左侧）可添加新的比较目标参数。

- 右键单击输入短线，并选择“插入输入”(Insert input) 命令。
- 右键单击输入短线并选择“删除”(Delete) 命令。

表格 7- 112 SWITCH 功能框数据类型和允许的比较运算

数据类型	比较	运算符语法
Byte、Word、DWord	等于	==
	不等于	<>
SInt、Int、DInt、USInt、 UInt、UDInt、Real、 LReal、Time、TOD、Date	等于	==
	不等于	<>
	大于或等于	>=
	小于或等于	<=
	大于	>
	小于	<

### SWITCH 功能框放置规则

- 比较输入前可以不连接 LAD/FBD 指令。
- 由于没有 ENO 输出，因此，在一个程序段中只允许使用一条 SWITCH 指令，并且 SWITCH 指令必须是程序段中的最后一个运算。



#### 7.8.14 RET 执行控制指令

可选的 RET 指令用于终止当前块的执行。当且仅当有能流通过 RET 线圈 (LAD)，或者当 RET 功能框的输入为真 (FBD) 时，则当前块的程序执行将在该点终止，并且不执行 RET 指令以后的指令。如果当前块为 OB，则参数“Return\_Value”将被忽略。如果当前块为 FC 或 FB，则将参数“Return\_Value”的值作为被调用功能框的 ENO 值传回到调用例程。

不要求用户将 RET 指令用作块中的最后一个指令；该操作是自动完成的。一个块中可以有多个 RET 指令。

有关 SCL，请参见 RETURN (页 237) 语句。

表格 7- 113 Return\_Value (RET) 执行控制指令

LAD	FBD	SCL	说明
		RETURN;	终止当前块的执行

表格 7- 114 参数的数据类型

参数	数据类型	说明
Return_Value	Bool	RET 指令的“Return_value”参数被分配给调用块中块调用功能框的 ENO 输出。

以下是在 FC 代码块中使用 RET 指令的示例步骤：

1. 创建新项目并添加 FC:
2. 编辑该 FC:
  - 从指令树添加指令。
  - 添加一个 RET 指令，包括参数“Return\_Value”的以下值之一：  
TRUE、FALSE，或用于指定所需返回值的存储位置。
  - 添加更多的指令。
3. 从 MAIN [OB1] 调用 FC。

MAIN 代码块中 FC 功能框的 EN 输入必须为真，才能开始执行 FC。

执行了有能流通过 RET 指令的 FC 后，该 FC 的 RET 指令所指定的值将出现在 MAIN 代码块中 FC 功能框的 ENO 输出上。

### 7.8.15 重新触发扫描循环看门狗指令

表格 7- 115 RE\_TRIGR 指令

LAD/FBD	SCL	说明
	RE_TRIGR();	RE_TRIGR（重新触发扫描时间监视狗）用于延长扫描循环监视狗定时器生成错误前允许的最大时间。

RE\_TRIGR 指令用于在单个扫描循环期间重新启动扫描循环监视狗定时器。结果是从最后一次执行 RE\_TRIGR 功能开始，使允许的最大扫描周期延长一个最大循环时间段。

#### 说明

对于 S7-1200 CPU 固件版本 2.2 之前的版本，RE\_TRIGR 限制为从程序循环 OB 执行，并可能用于无限期地延长 PLC 扫描时间。如果从启动 OB、中断 OB 或错误 OB 执行 RE\_TRIGR，则不会复位监视狗定时器且 ENO = FALSE。

对于固件版本 2.2 及以上版本，可从任何 OB（包括启动、中断和错误 OB）执行 RE\_TRIGR。但是，PLC 扫描时间最长只能延长到已组态最大循环时间的 10 倍。

## 设置 PLC 最大循环时间

可以在设备配置的“循环时间”(Cycle time) 下组态最大扫描循环时间值。

表格 7- 116 循环时间值

循环时间监视	最小值	最大值	默认值
最大循环时间	1 ms	6000 ms	150 ms

## 监视狗超时

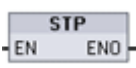
如果最大扫描循环定时器在扫描循环完成前达到预置时间，则会生成错误。如果用户程序中包含错误处理代码块 OB 80，则 CPU 将执行 OB 80，用户可以在其中添加程序逻辑以创建具体响应。如果不包含 OB 80，则忽略第一个超时条件并且 CPU 切换到 STOP。

如果在同一程序扫描中第二次发生最大扫描时间超时（2 倍的最大循环时间值），则触发错误使 CPU 切换到 STOP 模式。

在 STOP 模式下，用户程序停止执行，而 CPU 系统通信和系统诊断仍继续执行。

### 7.8.16 停止扫描循环指令

表格 7- 117 STP 指令

LAD/FBD	SCL	说明
	STP ();	STP（停止扫描循环）将 CPU 置于 STOP 模式。CPU 处于 STOP 模式时，将停止程序执行并停止过程映像的物理更新。

有关详细信息，请参见：组态从 RUN 切换到 STOP 时的输出 (页 88)。


如果 EN = TRUE，CPU 将进入 STOP 模式，程序执行停止，并且 ENO 状态无意义。否则，EN = ENO = 0。

### 7.8.17 获取错误指令

获取错误指令提供有关程序块执行错误的信息。如果在代码块中添加了 GetError 或 GetErrorID 指令，则可在程序块中处理程序错误。

### GetError

表格 7- 118 GetError 指令

LAD/FBD	SCL	说明
	<pre>GET_ERROR(Out);</pre>	指示发生本地程序块执行错误，并用详细错误信息填充预定义的错误数据结构。

表格 7- 119 参数的数据类型

参数	数据类型	说明
ERROR	ErrorStruct	错误数据结构： 可以重命名该结构，但不能重命名结构中的成员。

表格 7- 120 ErrorStruct 数据结构的元素


结构组件	数据类型	说明						
ERROR_ID	Word	错误 ID						
FLAGS	Byte	显示块调用期间是否出错。 <ul style="list-style-type: none"> <li>• 16#01: 块调用期间出错。</li> <li>• 16#00: 块调用期间未出错。</li> </ul>						
REACTION	Byte	默认响应： <ul style="list-style-type: none"> <li>• 0: 忽略（写入错误），</li> <li>• 1: 以替代值“0”继续（读取错误），</li> <li>• 2: 跳转指令（系统错误）</li> </ul>						
CODE_ADDRESS	CREF	有关块地址和类型的信息						
BLOCK_TYPE	Byte	出错块的类型： <ul style="list-style-type: none"> <li>• 1: OB</li> <li>• 2: FC</li> <li>• 3: FB</li> </ul>						
CB_NUMBER	UInt	代码块的编号						
OFFSET	UDInt	对内部存储器的引用						
MODE	Byte	访问模式： 根据具体的访问类型，可输出以下信息：						
		<table border="1" style="width: 100%; text-align: center;"> <tr> <td>模式</td> <td>(A)</td> <td>(B)</td> <td>(C)</td> <td>(D)</td> <td>(E)</td> </tr> </table>	模式	(A)	(B)	(C)	(D)	(E)
模式	(A)	(B)	(C)	(D)	(E)			



结构组件		数据类型	说明				
			0				
			1				偏移
			2			区域	
			3	位置	范围		编号
			4			区域	偏移
			5			区域	DB 编号 偏移
			6	伙伴编号 / 访问		区域	DB 编号 偏移
			7	伙伴编号 / 访问	插槽号 / 范围	区域	DB 编号 偏移
OPERAND_NUMBER		UInt	机器命令的操作数				
POINTER_NUMBER_LOCATION		UInt	(A) 内部指针				
SLOT_NUMBER_SCOPE		UInt	(B) 内部存储器中的存储区				
DATA_ADDRESS		NREF	有关操作数地址的信息				
	AREA	Byte	(C) 存储区： <ul style="list-style-type: none"> <li>• L: 16#40 – 4E、86、87、8E、8F、C0 – CE</li> <li>• I:16#81</li> <li>• Q: 16#82</li> <li>• M: 16#83</li> <li>• DB: 16#84、85、8A、8B</li> </ul>				
	DB_NUMBER	UInt	(D) 数据块编号				
	OFFSET	UDInt	(E) 操作数的相对地址				

## GetErrorID

表格 7- 121 GetErrorID 指令

LAD/FBD	SCL	说明
	<pre>GET_ERR_ID () ;</pre>	指示发生程序块执行错误，并报告错误的 ID（标识符代码）。

表格 7- 122 参数的数据类型

参数	数据类型	说明
ID	Word	ErrorStruct ERROR_ID 成员的错误标识符值

表格 7- 123 Error\_ID 值

ERROR_ID 十六进制值	ERROR_ID 十进制值	程序块执行错误
0	0	无错误
2503	9475	未初始化指针错误
2522	9506	操作数超出范围读取错误
2523	9507	操作数超出范围写入错误
2524	9508	无效区域读取错误
2525	9509	无效区域写入错误
2528	9512	数据分配读取错误（位赋值不正确）
2529	9513	数据分配写入错误（位赋值不正确）
2530	9520	DB 受到写保护
253A	9530	全局 DB 不存在
253C	9532	版本错误或 FC 不存在
253D	9533	指令不存在
253E	9534	版本错误或 FB 不存在
253F	9535	指令不存在
2575	9589	程序嵌套深度错误
2576	9590	局部数据分配错误
2942	10562	物理输入点不存在
2943	10563	物理输出点不存在

## 操作

默认情况下，CPU 通过将错误记录到诊断缓冲区来响应块执行错误。但是，如果在代码块中放置一个或多个 `GetError` 或 `GetErrorID` 指令，即将该块设置为在块内处理错误。在这种情况下，CPU 不在诊断缓冲区中记录错误。而是在 `GetError` 或 `GetErrorID` 指令的输出中报告错误信息。可以使用 `GetError` 指令读取详细错误信息，或使用 `GetErrorID` 指令只读取错误标识符。因为后续错误往往只是第一个错误的结果，所以第一个错误通常最重要。

在块内第一次执行 `GetError` 或 `GetErrorID` 指令将返回块执行期间检测到的第一个错误。在块启动到执行 `GetError` 或 `GetErrorID` 期间随时都可能发生该错误。随后执行 `GetError` 或 `GetErrorID` 将返回上次执行 `GetError` 或 `GetErrorID` 以来发生的第一个错误。不保存错误历史，执行任一指令都将使 PLC 系统重新捕捉下一个错误。

可以在数据块编辑器和块接口编辑器中添加 `GetError` 指令所使用的 `ErrorStruct` 数据类型，从而程序逻辑可以访问这些值。从数据类型下拉列表中选择 `ErrorStruct` 以添加该结构。您可以使用唯一的名称创建多个 `ErrorStruct` 元素。不能重命名 `ErrorStruct` 的成员。

## ENO 指示的错误条件

如果 `EN = TRUE` 且 `GetError` 或 `GetErrorID` 执行，则：

- `ENO = TRUE` 表示发生代码块执行错误并提供错误数据
- `ENO = FALSE` 表示未发生代码块执行错误

可以将错误响应程序逻辑连接到在发生错误后激活的 `ENO`。如果存在错误，该输出参数会将错误数据存储到程序能够访问这些数据的位置。

`GetError` 和 `GetErrorID` 可用来将错误信息从当前执行块（被调用块）发送到调用块。将该指令放置在被调用块程序的最后一个程序段中可以报告被调用块的最终执行状态。


## 7.9 字逻辑指令

### 7.9.1 AND、OR 和 XOR 指令

表格 7- 124 AND、OR 和 XOR 指令

LAD/FBD	SCL	说明
	<code>out := in1 AND in2;</code>	AND: 逻辑 AND
	<code>out := in1 OR in2;</code>	OR: 逻辑 OR
	<code>out := in1 XOR in2;</code>	XOR: 逻辑异或

1 对于 LAD 和 FBD: 单击“???”并从下拉菜单中选择数据类型。

 要添加输入, 请单击“创建”(Create) 图标, 或在其中一个现有 IN 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, 请在其中一个现有 IN 参数 (多于两个原始输入时) 的输入短线处单击右键, 并选择“删除”(Delete) 命令。

表格 7- 125 参数的数据类型


参数	数据类型	说明
IN1, IN2	Byte, Word, DWord	逻辑输入
OUT	Byte, Word, DWord	逻辑输出

1 所选数据类型将 IN1、IN2 和 OUT 设置为相同的数据类型。

IN1 和 IN2 的相应位值相互组合, 在参数 OUT 中生成二进制逻辑结果。执行这些指令之后, ENO 总是为 TRUE。

## 7.9.2 取反指令

表格 7- 126 INV 指令

LAD/FBD	SCL	说明
	不提供	计算参数 IN 的二进制反码。通过对参数 IN 各位的值取反来计算反码（将每个 0 变为 1，每个 1 变为 0）。执行该指令后，ENO 总是为 TRUE。

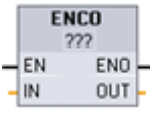
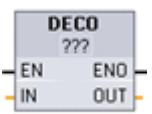
1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7- 127 参数的数据类型

参数	数据类型	说明
IN	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	要取反的数据元素
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	取反后的输出

## 7.9.3 编码和解码指令

表格 7- 128 ENCO 和 DECO 指令

LAD/FBD	SCL	说明
	<code>out := ENCO(_in_);</code>	<p>将位序列编码成二进制数</p> <p>ENCO 指令将参数 IN 转换为与参数 IN 的最低有效设置位的位位置对应的二进制数，并将结果返回给参数 OUT。如果参数 IN 为 0000 0001 或 0000 0000，则将值 0 返回给参数 OUT。如果参数 IN 的值为 0000 0000，则 ENO 设置为 FALSE。</p>
	<code>out := DECO(_in_);</code>	<p>将二进制数解码成位序列</p> <p>DECO 指令通过将参数 OUT 中的相应位位置设置为 1（其它所有位设置为 0）解码参数 IN 中的二进制数。执行 DECO 指令之后，ENO 始终为 TRUE。</p> <p>注：DECO 指令的默认数据类型为 DWORD。在 SCL 中，将指令名称更改为 DECO_BYTE 或 DECO_WORD 可解码字节或字值，并分配到字节或字变量或地址。</p>

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

7.9 字逻辑指令

表格 7- 129 参数的数据类型

参数	数据类型	说明
IN	ENCO: Byte, Word, DWord	ENCO: 要编码的位序列
	DECO: UInt	DECO: 要解码的值
OUT	ENCO: Int	ENCO: 编码后的值
	DECO: Byte, Word, DWord	DECO: 解码后的位序列

表格 7- 130 ENCO 的 OUT 参数

ENO	条件	结果 (OUT)
1	无错误	有效位号
0	IN 为零	OUT 设置为零

DECO 参数 OUT 的数据类型选项 (Byte、Word 或 DWord) 限制参数 IN 的可用范围。如果参数 IN 的值超出可用范围, 将执行求模运算, 如下所示提取最低有效位。

DECO 参数 IN 的范围:

- 3 位 (值 0-7) IN 用于设置 Byte OUT 中 1 的位位置
- 4 位 (值 0-15) IN 用于设置 Word OUT 中 1 的位位置
- 5 位 (值 0-31) IN 用于设置 DWord OUT 中 1 的位位置

表格 7- 131 示例

DECO IN 值		DECO OUT 值 (解码单个位位置)	
Byte OUT 8 位	最小 IN	0	00000001
	最大 IN	7	10000000
Word OUT 16 位	最小 IN	0	0000000000000001
	最大 IN	15	1000000000000000
DWord OUT 32 位	最小 IN	0	00000000000000000000000000000001
	最大 IN	31	10000000000000000000000000000000

## 7.9.4 选择、多路复用和多路分用指令

表格 7- 132 SEL（选择）指令

LAD/FBD	SCL	说明
	<pre>out := SEL(     g:=_bool_in,     in0:=-_variant_in,     in1:=-_variant_in);</pre>	SEL 根据参数 G 的值将两个输入值之一分配给参数 OUT。

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

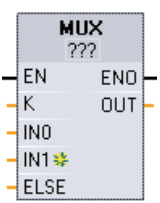
表格 7- 133 SEL 指令的数据类型

参数	数据类型 <sup>1</sup>	说明
G	Bool	<ul style="list-style-type: none"> <li>• 0 选择 IN0</li> <li>• 1 选择 IN1</li> </ul>
IN0, IN1	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Char	输入
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Char	输出

1 输入变量和输出变量必须为相同的数据类型。

**条件代码：** 执行 SEL 指令之后，ENO 始终为 TRUE。

表格 7- 134 MUX（多路复用）指令

LAD/FBD	SCL	说明
	<pre>out := MUX(     k:=_unit_in,     in1:=variant_in,     in2:=variant_in,     [...in32:=variant_in,     ]     inelse:=variant_in);</pre>	MUX 根据参数 K 的值将多个输入值之一复制到参数 OUT。如果参数 K 的值大于 (INn- 1)，则会将参数 ELSE 的值复制到参数 OUT。

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。



要添加输入，请单击“创建”(Create) 图标，或在其中一个现有 IN 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，请在其中一个现有 IN 参数（多于两个原始输入时）的输入短线处单击右键，并选择“删除”(Delete) 命令。

表格 7- 135 MUX 指令的数据类型

参数	数据类型	说明
K	UInt	<ul style="list-style-type: none"> <li>• 0 选择 IN1</li> <li>• 1 选择 IN2</li> <li>• n 选择 INn</li> </ul>
IN0, IN1, .. INn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Char	输入
ELSE	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Char	输入替换值（可选）
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Char	输出

1 输入变量和输出变量必须为相同的数据类型。

表格 7- 136 DEMUX（多路分用）指令

LAD/FBD	SCL	说明
	<pre> DEMUX (     k:=_unit_in,     in:=variant_in,     out1:=variant_in,     out2:=variant_in,     [...out32:=variant_in     ,]     outelse:=variant_in) ;                 </pre>	DEMUX 将分配给参数 IN 的位置值复制到多个输出之一。参数 K 的值选择将哪一输出作为 IN 值的目标。如果 K 的值大于数值 (OUTn - 1)，则会将 IN 值复制到分配给 ELSE 参数的位置。

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

要添加输出，请单击创建图标，或在其中一个现有 OUT 参数的输出短线处单击右键，并选择“插入输出”(Insert output) 命令。要删除输出，请在其中一个现有 OUT 参数（多于两个原始输出时）的输出短线处单击右键，并选择“删除”(Delete) 命令。





要添加输出，请单击“创建”(Create) 图标，或在其中一个现有 OUT 参数的输出短线处单击右键，并选择“插入输出”(Insert output) 命令。

要删除输出，请在其中一个现有 OUT 参数（多于两个原始输出时）的输出短线处单击右键，并选择“删除”(Delete) 命令。

表格 7- 137 DEMUX 指令的数据类型

参数	数据类型 <sup>1</sup>	说明
K	UInt	选择器的值： <ul style="list-style-type: none"> <li>• 0 选择 OUT1</li> <li>• 1 选择 OUT2</li> <li>• n 选择 OUTn</li> </ul>
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	输入
OUT0, OUT1, .. OUTn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	输出
ELSE	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Char	K 大于 (OUTn - 1) 时的替换输出

<sup>1</sup> 输入变量和输出变量必须为相同的数据类型。

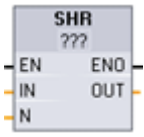
表格 7- 138 MUX 和 DEMUX 指令的 ENO 状态

ENO	条件	结果 (OUT)
1	无错误	MUX: 将选择的 IN 值复制到 OUT DEMUX: 将 IN 值复制到选择的 OUT
0	MUX: K 大于输入数 -1	<ul style="list-style-type: none"> <li>• 不提供 ELSE: OUT 不变,</li> <li>• 提供 ELSE, 将 ELSE 值分配给 OUT</li> </ul>
	DEMUX: K 大于输出数 -1	<ul style="list-style-type: none"> <li>• 不提供 ELSE: 输出不变,</li> <li>• 提供 ELSE, 将 IN 值复制到 ELSE</li> </ul>

## 7.10 移位和循环

### 7.10.1 移位指令

表格 7- 139 SHR 和 SHL 指令

LAD/FBD	SCL	说明
	<pre> out := SHR(     in:=_variant_in_,     n:=_uint_in_); out := SHL(     in:=_variant_in_,     n:=_uint_in_);                     </pre>	使用移位指令（SHL 和 SHR）移动参数 IN 的位序列。结果将分配给参数 OUT。参数 N 指定移位的位数： <ul style="list-style-type: none"> <li>• SHR: 右移位序列</li> <li>• SHL: 左移位序列</li> </ul>

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

表格 7- 140 参数的数据类型

参数	数据类型	说明
IN	Byte, Word, DWord	要移位的位序列
N	UInt	要移位的位数
OUT	Byte, Word, DWord	移位操作后的位序列

- 若 N=0，则不移位。将 IN 值分配给 OUT。
- 用 0 填充移位操作清空的位位置。
- 如果要移位的位数 (N) 超过目标值中的位数（Byte 为 8 位、Word 为 16 位、DWord 为 32 位），则所有原始位值将被移出并用 0 代替（将 0 分配给 OUT）。
- 对于移位操作，ENO 总是为 TRUE。

表格 7- 141 Word 数据的 SHL 示例

自右插入零，使 Word 的位左移 (N = 1)			
IN	1110 0010 1010 1101	首次移位前的 OUT 值:	1110 0010 1010 1101
		首次左移后:	1100 0101 0101 1010
		第二次左移后:	1000 1010 1011 0100
		第三次左移后:	0001 0101 0110 1000

## 7.10.2 循环指令

表格 7- 142 ROR 和 ROL 指令

LAD/FBD	SCL	说明
	<pre> out := ROL(     in:=_variant_in_,     n:=_uint_in); out := ROR(     in:=_variant_in_,     n:=_uint_in); </pre>	<p>循环指令（ROR 和 ROL）用于将参数 IN 的位序列循环移位。结果分配给参数 OUT。参数 N 定义循环移位的位数。</p> <ul style="list-style-type: none"> <li>• ROR: 循环右移位序列</li> <li>• ROL: 循环左移位序列</li> </ul>

1 对于 LAD 和 FBD: 单击“???”并从下拉菜单中选择数据类型。

表格 7- 143 参数的数据类型

参数	数据类型	说明
IN	Byte, Word, DWord	要循环移位的位序列
N	UInt	要循环移位的位数
OUT	Byte, Word, DWord	循环移位操作后的位序列

- 若 N=0, 则不循环移位。将 IN 值分配给 OUT。
- 从目标值一侧循环移出的位数据将循环移位到目标值的另一侧, 因此原始位值不会丢失。
- 如果要循环移位的位数 (N) 超过目标值中的位数 (Byte 为 8 位、Word 为 16 位、DWord 为 32 位), 仍将执行循环移位。
- 执行循环指令之后, ENO 始终为 TRUE。

表格 7- 144 Word 数据的 ROR 示例

将各个位从右侧循环移出到左侧 (N = 1)			
IN	0100 0000 0000 0001	首次循环移位前的 OUT 值:	0100 0000 0000 0001
		首次循环右移后:	1010 0000 0000 0000
		第二次循环右移后:	0101 0000 0000 0000



## 扩展指令

### 8.1 日期和日时钟

#### 8.1.1 日期和时间指令

日期和时间指令用于设计日历和时间计算。

- T\_CONV 用于转换时间值的数据类型：(Time 转换为 DInt) 或 (DInt 转换为 Time)
- T\_ADD 将 Time 值与 DTL 值相加：(Time + Time = Time) 或 (DTL + Time = DTL)
- T\_SUB 将 Time 值与 DTL 值相减：(Time - Time = Time) 或 (DTL - Time = DTL)
- T\_DIFF 提供两个 DTL 值的差作为 Time 值：DTL - DTL = Time
- T\_COMBINE 将 Date 值和 Time\_and\_Date 值组合在一起生成 DTL 值

有关 DTL 和 Time 数据的结构信息，请参见 Time 和 Date 数据类型 (页 98)部分。

表格 8-1 T\_CONV (时间转换) 指令

LAD/FBD	SCL	说明
	<pre>out := T_CONV(    in:=_varian t_in);</pre>	T_CONV 将 Time 数据类型转换为 DInt 数据类型，或将 DInt 数据类型转回 Time 数据类型。

1 对于 LAD 和 FBD：单击“???”并从下拉菜单中选择数据类型。

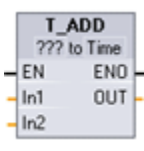
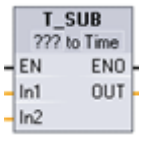
表格 8-2 T\_CONV 参数的数据类型

参数和类型	数据类型	说明	
IN	IN	DInt, Time	输入 Time 值或 DInt 值
OUT	OUT	DInt, Time	经过转换的 DInt 值或 Time 值

## 扩展指令

### 8.1 日期和日时钟

表格 8-3 T\_ADD (时间相加) 和 T\_SUB (时间相减) 指令

LAD/FBD	SCL	说明
	<pre>out := T_ADD(   in1:=_variant_in,   in2:=_time_in);</pre>	<p>T_ADD 将输入 IN1 的值 (DTL 或 Time 数据类型) 与输入 IN2 的 Time 值相加。参数 OUT 提供 DTL 或 Time 值结果。允许以下两种数据类型的运算:</p> <ul style="list-style-type: none"> <li>• Time + Time = Time</li> <li>• DTL + Time = DTL</li> </ul>
	<pre>out := T_SUB(   in1:=_variant_in,   in2:=_time_in);</pre>	<p>T_SUB 从 IN1 (DTL 或 Time 值) 中减去 IN2 的 Time 值。参数 OUT 以 DTL 或 Time 数据类型提供差值。允许以下两种数据类型的运算:</p> <ul style="list-style-type: none"> <li>• Time - Time = Time</li> <li>• DTL - Time = DTL</li> </ul>

1 对于 LAD 和 FBD: 单击“???”并从下拉菜单中选择数据类型。

表格 8-4 T\_ADD 和 T\_SUB 参数的数据类型

参数和类型	数据类型	说明	
IN1 <sup>1</sup>	IN	DTL, Time	DTL 或 Time 值
IN2	IN	Time	要加上或减去的 Time 值
OUT	OUT	DTL, Time	DTL 或 Time 的和值或差值

1 从指令名称下方提供的下拉列表中选择 IN1 的数据类型。所选的 IN1 数据类型同时也会设置参数 OUT 的数据类型。

表格 8-5 T\_DIFF (时差) 指令

LAD/FBD	SCL	说明
	<pre>out := T_DIFF(   in1:=_DTL_in,   in2:=_DTL_in);</pre>	<p>T_DIFF 从 DTL 值 (IN1) 中减去 DTL 值 (IN2)。参数 OUT 以 Time 数据类型提供差值。</p> <ul style="list-style-type: none"> <li>• DTL - DTL = Time</li> </ul>

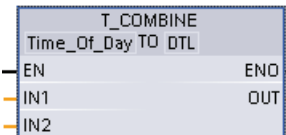
表格 8-6 T\_DIFF 参数的数据类型

参数和类型		数据类型	说明
IN1	IN	DTL	DTL 值
IN2	IN	DTL	要减去的 DTL 值
OUT	OUT	Time	Time 差

条件代码：ENO = 1 表示未发生错误。ENO = 0 且参数 OUT = 0 的错误：

- DTL 值无效
- Time 值无效

表格 8-7 T\_COMBINE（组合时间值）指令

LAD/FBD	SCL	说明
	<pre> out := CONCAT_DATE_TOD(   In1 := _date_in,   In2 := _tod_in); </pre>	T_COMBINE 将 Date 值和 Time_of_Day 值组合在一起生成 DTL 值。

1 请注意，在扩展指令中，T\_COMBINE 指令相当于 SCL 中的 CONCAT\_DATE\_TOD 函数。




表格 8-8 T\_COMBINE 参数的数据类型

参数和类型		数据类型	说明
IN1	IN	Date	要组合的 Date 值必须在 DATE#1990-01-01 和 DATE#2089-12-31 之间
IN2	IN	Time_of_Day	要组合的 Time_of_Day 值
OUT	OUT	DTL	DTL 值

### 8.1.2 设置和读取系统时钟

时钟指令用于设置和读取 CPU 系统时钟。使用数据类型 DTL (页 98) 提供日期和时间值。

表格 8-9 系统时间指令

LAD/FBD	SCL	说明
	<pre>ret_val := WR_SYS_T(     in:=_DTL_in_);</pre>	WR_SYS_T (写入系统时间) 使用参数 IN 中的 DTL 值设置 CPU 时钟。该时间值不包括本地时区或夏令时偏移量。
	<pre>ret_val := RD_SYS_T(     out=&gt;_DTL_out);</pre>	RD_SYS_T (读取系统时间) 从 CPU 中读取当前系统时间。该时间值不包括本地时区或夏令时偏移量。
	<pre>ret_val := RD_LOC_T(     out=&gt;_DTL_out);</pre>	RD_LOC_T (读取本地时间) 以 DTL 数据类型提供 CPU 的当前本地时间。该时间值反映了就夏令时 (如果已经组态) 进行过适当调整的本地时区。

表格 8-10 参数的数据类型

参数和类型		数据类型	说明
IN	IN	DTL	要在 CPU 系统时钟内设置的时间
RET_VAL	OUT	Int	执行条件代码
OUT	OUT	DTL	RD_SYS_T: 当前 CPU 系统时间 RD_LOC_T: 当前本地时间。包括对夏令时的任何调整 (如果已经组态)

- 通过使用用户在设备组态常规选项卡“时间”(Time of day) 参数中设置的时区和夏令时偏移量计算本地时间。
- 时区组态是相对于 UTC 或 GMT 时间的偏移量。
- 夏令时组态指定夏令时开始时的月份、星期、日期和小时。
- 标准时间组态也会指定标准时间开始时的月份、星期、日期和小时。
- 时区偏移量始终会应用到系统时间值。只有在夏令时有效时才会应用夏令时偏移量。

**说明**

**夏令时和标准起始时间组态**

CPU 设备组态的“夏令时开始”(Start for daylight saving time) 的“时间”(Time of day) 属性必须是本地时间。



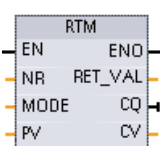
**条件代码：** ENO = 1 表示未发生错误。 ENO = 0 表示发生了执行错误，同时在 RET\_VAL 输出中提供条件代码。

表格 8- 11 条件代码

RET_VAL (W#16#...)	说明
0000	当前的本地时间为标准时间。
0001	夏令制时间已组态，当前的本地时间为夏令制时间。
8080	本地时间不可用
8081	非法年份值
8082	非法月份值
8083	非法日期值
8084	非法小时值
8085	非法分钟值
8086	非法秒数值
8087	非法纳秒值
80B0	实时时钟发生了故障。

### 8.1.3 运行时间计时器指令

表格 8- 12 RTM 指令

LAD/FBD	SCL	说明
	<pre>RTM(NR:=_uint_in_,     MODE:=_byte_in_,     PV:=_dint_in_,     CQ=&gt;_bool_out_,     CV=&gt;_dint_out_);</pre>	RTM（运行时间计时器）指令可以设置、启动、停止和读取 CPU 中的运行时间小时计时器。

表格 8- 13 参数的数据类型

参数和类型		数据类型	说明
NR	IN	UInt	运行时间计时器编号：（可能值： 0..9）
MODE	IN	Byte	RTM 执行模式编号： <ul style="list-style-type: none"> <li>• 0 = 获取值（然后状态值写入 CQ，当前值写入 CV）</li> <li>• 1 = 启动（从上一计数值开始）</li> <li>• 2 = 停止</li> <li>• 4 = 设置（设为 PV 中指定的值）</li> <li>• 5 = 设置（设为 PV 中指定的值），然后启动</li> <li>• 6 = 设置（设为 PV 中指定的值），然后停止</li> <li>• 7 = 将 CPU 中的所有 RTM 值保存到 MC（存储卡）</li> </ul>
PV	IN	DInt	指定运行时间计时器的预设小时值
RET_VAL	OUT	Int	功能结果/错误消息
CQ	OUT	Bool	运行时间计时器的状态（1 = 正在运行）
CV	OUT	DInt	指定计时器的当前运行小时值

CPU 最多可运行 10 个运行小时计时器来跟踪关键控制子系统的运行小时数。必须对每个定时器执行一次 RTM 分别启动小时计时器。CPU 从运行模式切换为停止模式时，所有运行小时计时器都将停止。还可以使用 RTM 执行模式 2 停止各个的定时器。

CPU 从停止模式切换为运行模式时，必须对每个已启动的定时器执行一次 RTM 来重新启动小时计时器。运行时间计时器值大于 2147483647 小时后，将停止计时并发出“上溢”错误。必须为每个定时器执行一次 RTM 指令，以复位或修改定时器。

CPU 电源故障或循环上电会导致断电过程将当前运行时间计时器值保存在保持性存储器中。在 CPU 上电时，所存储的运行时间计时器值将重新加载到定时器，之前的运行时间小时总数不会丢失。必须重启运行时间计时器才能累加额外的运行时间。

用户程序还可以使用 RTM 执行模式 7 将运行时间计时器值保存在存储卡中。执行 RTM 模式 7 时的所有定时器的状态将保存在存储卡中。由于小时定时器会在程序运行过程中或启动或停止，随着时间的推移，这些存储值就可能出错。因此，必须周期性更新存储卡值，以捕获重要的运行事件。在存储卡中保存 RTM 值的好处是，在替代 CPU 中插入存储卡时，就可以在其中使用程序和所保存的 RTM 值。如果未将 RTM 值保存在存储卡中，则会丢失定时器值（在替代 CPU 中）。

**说明**

**避免过度调用执行存储卡写操作的程序**

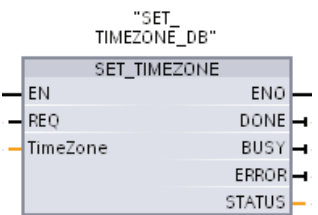
尽可能减少闪存卡写操作，以延长存储卡的使用寿命。

表格 8- 14 条件代码

RET_VAL (W#16#....)	说明
0	无错误
8080	运行时间计时器编号错误
8081	负值已传递给参数 PV
8082	操作小时计数器溢出
8091	输入参数 MODE 包含非法值。
80B1	无法将值保存到 MC (MODE=7)

### 8.1.4 SET\_TIMEZONE 指令

表格 8- 15 SET\_TIMEZONE 指令

LAD/FBD	SCL	说明
	<pre>"SET_TIMEZONE_DB" (     REQ:=_bool_in,     Timezone:=_struct_in,     DONE=&gt;_bool_out_,     BUSY=&gt;_bool_out_,     ERROR=&gt;_bool_out_,     STATUS=&gt;_word_out_);</pre>	设置本地时区和夏令时参数，以用于将 CPU 系统时间转换为本地时间。

1 在 SCL 示例中，“SET\_TIMEZONE\_DB”是背景 DB 的名称。

表格 8- 16 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	REQ=1: 执行功能
Timezone	IN	TimeTransformationRule	将系统时间转换为本地时间的规则
DONE	OUT	Bool	功能执行完毕
BUSY	OUT	Bool	功能忙

参数和类型		数据类型	说明
ERROR	OUT	Bool	检测到错误
STATUS	OUT	Word	功能结果/错误消息

要手动组态 CPU 的时区参数，请使用设备组态“常规”(General) 选项卡中的“时间”(Time of day) 属性。

通过程序使用 SET\_TIMEZONE 指令设置本地时间组态。“TimeTransformationRule”结构的参数用于指定本地时区以及在标准时间和夏令时之间自动切换的时间。

表格 8- 17 "TimeTransformationRule" 结构

参数	数据类型	说明
Bias	Int	UTC 与本地时间的时差 [min]
DaylightBias	Int	冬季时间与夏季时间的时差 [min]
DaylightStartMonth	USInt	夏令时的月份
DaylightStartWeek	USInt	夏令时的星期： <ul style="list-style-type: none"> <li>• 1 = 该月的第一周</li> <li>• ...</li> <li>• 5 = 该月的最后一周</li> </ul>
DaylightStartWeekday	USInt	夏令时的周几： <ul style="list-style-type: none"> <li>• 1 = 星期日</li> <li>• ...</li> <li>• 7 = 星期六</li> </ul>
DaylightStartHour	USInt	夏令时的小时
StandardStartMonth	USInt	转换为冬季时间的月份
StandardStartWeek	USInt	转换为冬季时间的周： <ul style="list-style-type: none"> <li>• 1 = 该月的第一周</li> <li>• ...</li> <li>• 5 = 该月的最后一周</li> </ul>
StandardStartWeekday	USInt	冬季时间的周几： <ul style="list-style-type: none"> <li>• 1 = 星期日</li> <li>• ...</li> <li>• 7 = 星期六</li> </ul>

参数	数据类型	说明
StandardStartHour	USInt	冬季时间的小时
Time Zone Name	STRING [80]	时区名称： (GMT +01:00) 柏林、伯尔尼、布鲁塞尔、罗马、斯德哥尔摩和越南

## 8.2 字符串和字符

### 8.2.1 String 数据概述

#### 字符串数据类型

**String** 数据被存储成 2 个字节的标头后跟最多 254 个 ASCII 码字符组成的字符字节。**String** 标头包含两个长度。第一个字节是初始化字符串时方括号中给出的最大长度，默认值为 254。第二个标头字节是当前长度，即字符串中的有效字符数。当前长度必须小于或等于最大长度。**String** 格式占用的存储字节数比最大长度大 2 个字节。

#### 初始化 String 数据

在执行任何字符串指令之前，必须将 **String** 输入和输出数据初始化为存储器中的有效字符串。

#### 有效 String 数据

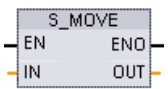
有效字符串的最大长度必须大于 0 但小于 255。当前长度必须小于等于最大长度。

字符串无法分配给 I 或 Q 存储区。

有关详细信息，请参见：**String** 数据类型的格式 (页 100)。

### 8.2.2 S\_MOVE 指令

表格 8- 18 字符串移动指令

LAD/FBD	SCL	说明
	<pre>out := in;</pre>	将源 IN 字符串复制到 OUT 位置。S_MOVE 的执行并不影响源字符串的内容。

表格 8- 19 参数的数据类型

参数	数据类型	说明
IN	String	源字符串
OUT	String	目标地址

如果输入 IN 中字符串的实际长度超过输出 OUT 存储的字符串最大长度，则会复制 OUT 字符串能容纳的部分 IN 字符串。

### 8.2.3 字符串转换指令

#### 8.2.3.1 字符串到值以及值到字符串的转换

可以使用以下指令将数字字符串转换为数值或将数值转换为数字字符串：

- S\_CONV 用于将数字字符串转换成数值或将数值转换成数字字符串
- STRG\_VAL 使用格式选项将数字字符串转换成数值
- VAL\_STRG 使用格式选项将数值转换成数字字符串

## S\_CONV (字符串到值的转换)

表格 8-20 字符串转换指令

LAD/FBD	SCL	说明
	<pre>out := &lt;Type&gt;_TO_&lt;Type&gt;(in);</pre>	<p>将字符串转换成相应的值，或将值转换成相应的字符串。S_CONV 指令没有输出格式选项。因此，S_CONV 指令比 STRG_VAL 指令和 VAL_STRG 指令更简单，但灵活性更差。</p>

- 1 对于 LAD/FBD：单击“???”并从下拉列表中选择数据类型。
- 2 对于 SCL：从扩展指令中选择 S\_CONV，然后应答数据类型转换的提示信息。STEP 7 随后会显示相应的转换指令。

表格 8-21 数据类型 (字符串到值)

参数和类型		数据类型	说明
IN	IN	String	输入字符串
OUT	OUT	String, Char, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	输出数值

字符串参数 IN 的转换从首个字符开始，并一直进行到字符串的结尾，或者一直进行到遇到第一个不是“0”到“9”、“+”、“-”或“.”的字符为止。结果值将在参数 OUT 中指定的位置提供。如果输出数值不在 OUT 数据类型的范围内，则参数 OUT 设置为 0，并且 ENO 设置为 FALSE。否则，参数 OUT 将包含有效的结果，并且 ENO 设置为 TRUE。

输入 String 格式规则：

- 如果在 IN 字符串中使用小数点，则必须使用“.”字符。
- 允许使用逗号字符“,”作为小数点左侧的千位分隔符，并且逗号字符会被忽略。
- 忽略前导空格。

**S\_CONV (值到字符串的转换)**

表格 8-22 数据类型 (值到字符串)

参数和类型		数据类型	说明
IN	IN	String, Char, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	输入数值
OUT	OUT	String	输出字符串

整数值、无符号整数值或浮点值 **IN** 在 **OUT** 中被转换为相应的字符串。在执行转换前，参数 **OUT** 必须引用有效字符串。有效字符串由第一个字节中的最大字符串长度、第二个字节中的当前字符串长度以及后面字节中的当前字符串字符组成。转换后的字符串将从第一个字符开始替换 **OUT** 字符串中的字符，并调整 **OUT** 字符串的当前长度字节。**OUT** 字符串的最大长度字节不变。

被替换的字符数取决于参数 **IN** 的数据类型和数值。被替换的字符数必须在参数 **OUT** 的字符串长度范围内。**OUT** 字符串的最大字符串长度（第一个字节）应大于或等于被转换字符的最大预期数目。下表列出了所支持的各种数据类型要求的最大可能字符串长度。

表格 8-23 每种数据类型的最大字符串长度

IN 数据类型	OUT 字符串中被转换字符的最大数目	示例	包括最大及当前长度字节在内的总字符串长度
USInt	3	255	5
SInt	4	-128	6
UInt	5	65535	7
Int	6	-32768	8
UDInt	10	4294967295	12
DInt	11	-2147483648	13


输出 **String** 格式规则：

- 写入到参数 **OUT** 的值不使用前导“+”号。
- 使用定点表示法（不可使用指数表示法）。
- 参数 **IN** 为 **Real** 数据类型时，使用句点字符“.”表示小数点。



## STRG\_VAL 指令

表格 8-24 字符串转换成值指令

LAD/FBD	SCL	说明
	<pre>"STRG_VAL" (   in:=_string_in,   format:=_word_in,   p:=uint_in,   out=&gt;_variant_out);</pre>	将数字字符串转换为相应的整型或浮点型表示法。

1 对于 LAD/FBD：单击“???”并从下拉列表中选择数据类型。

表格 8-25 STRG\_VAL 指令的数据类型

参数和类型		数据类型	说明
IN	IN	String	要转换的 ASCII 字符串
FORMAT	IN	Word	输出格式选项
P	IN	UInt, Byte, USInt	IN：指向要转换的第一个字符的索引（第一个字符 = 1）
OUT	OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	转换后的数值

转换从字符串 IN 中的字符偏移量 P 位置开始，并一直进行到字符串的结尾，或者一直进行到遇到第一个不是“+”、“-”、“.”、“;”、“e”、“E”或“0”到“9”的字符为止。结果放置在参数 OUT 中指定的位置。

必须在执行前将 String 数据初始化为存储器中的有效字符串。

以下定义了 STRG\_VAL 指令的 FORMAT 参数。未使用的位位置必须设置为零。

8.2 字符串和字符

表格 8-26 STRG\_VAL 指令的格式

位 16								位 8	位 7							位 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f	r

f = 表示法格式                      1= 指数表示法  
   0 = 定点表示法  
 r = 小数点格式                      1 = “,” (逗号字符)  
   0 = “.” (句点字符)

表格 8-27 FORMAT 参数的值


FORMAT (W#16#)	表示法格式	小数点表示法
0000 (默认)	定点	"."
0001		","
0002	指数	"."
0003		","
0004 到 FFFF	非法值	

STRG\_VAL 转换的规则:

- 如果使用句点字符“.”作为小数点，则小数点左侧的逗号“,”将被解释为千位分隔符字符。允许使用逗号字符并且会将其忽略。
- 如果使用逗号字符“,”作为小数点，则小数点左侧的句点“.”将被解释为千位分隔符字符。允许使用句点字符并且会将其忽略。
- 忽略前导空格。

## VAL\_STRG 指令

表格 8-28 值转换成字符串的运算

LAD/FBD	SCL	说明
	<pre>"VAL_STRG" (   in:=_variant_in,   size:=_usint_in,   prec:=_usint_in,   format:=_word_in,   p:=uint_in,   out=&gt;_string_out);</pre>	将整数值、无符号整数值或浮点值转换为相应的字符串表示法。

1 对于 LAD/FBD：单击“???”并从下拉列表中选择数据类型。

表格 8-29 VAL\_STRG 指令的数据类型

参数和类型		数据类型	说明
IN	IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	要转换的值
SIZE	IN	USInt	要写入 OUT 字符串的字符数
PREC	IN	USInt	小数部分的精度或大小。不包括小数点。
FORMAT	IN	Word	输出格式选项
P	IN	UInt, Byte, USInt	IN：指向要替换的第一个 OUT 字符串字符的索引（第一个字符 = 1）
OUT	OUT	String	转换后的字符串

参数 IN 表示的值将被转换为参数 OUT 所引用的字符串。在执行转换前，参数 OUT 必须为有效字符串。

转换后的字符串将从字符偏移量计数 P 位置开始替换 OUT 字符串中的字符，一直到参数 SIZE 指定的字符数。SIZE 中的字符数必须在 OUT 字符串长度范围内（从字符位置 P 开始计数）。该指令对于将数字字符嵌入到文本字符串中很有用。例如，可以将数字“120”放入字符串“Pump pressure = 120 psi”中。

参数 PREC 用于指定字符串中小数部分的精度或位数。如果参数 IN 的值为整数，则 PREC 指定小数点的位置。例如，如果数据值为 123 且 PREC = 1，则结果为“12.3”。对于 Real 数据类型，支持的最大精度为 7 位。

8.2 字符串和字符

如果参数 P 大于 OUT 字符串的当前大小，则会添加空格，一直到位置 P，并将该结果附加到字符串末尾。如果达到了最大 OUT 字符串长度，则转换结束。

以下定义了 VAL\_STRG 指令的 FORMAT 参数。未使用的位位置必须设置为零。

表格 8-30 VAL\_STRG 指令的格式

位 16							位 8	位 7								位 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	s	f	r

s = 数字符号字符

1= 使用符号字符“+”和“-”

0 = 仅使用符号字符“-”

f = 表示法格式

1= 指数表示法

0 = 定点表示法

r = 小数点格式

1 = “,” (逗号字符)

0 = “.” (句点字符)

表格 8-31 FORMAT 参数的值

FORMAT (WORD)	数字符号字符	表示法格式	小数点表示法
W#16#0000	仅“-”	定点	“.”
W#16#0001			“,”
W#16#0002		指数	“.”
W#16#0003			“,”
W#16#0004	“+”和“-”	定点	“.”
W#16#0005			“,”
W#16#0006		指数	“.”
W#16#0007			“,”
W#16#0008 到 W#16#FFFF	非法值		

参数 OUT 字符串的格式规则如下：

- 如果转换后的字符串小于指定的大小，则会在字符串的最左侧添加前导空格字符。
- 如果 FORMAT 参数的符号位为 FALSE，则会将无符号和有符号整型值写入输出缓冲区，且不带前导“+”号。必要时会使用“-”号。  
<前导空格><无前导零的数字>!'<PREC 数字>
- 如果符号位为 TRUE，则会将无符号和有符号整型值写入输出缓冲区，且总是带前导符号字符。  
<前导空格><符号><不带前导零的数字>!'<PREC 数字>
- 如果 FORMAT 被设置为指数表示法，则会按以下方式将 Real 数据类型的值写入输出缓冲区：  
<前导空格><符号><数字> !' <PREC 数字>'E' <符号><无前导零的数字>
- 如果 FORMAT 被设置为定点表示法，则会按以下方式将整型、无符号整型和实型值写入输出缓冲区：  
<前导空格><符号><不带前导零的数字>!'<PREC 数字>
- 小数点左侧的前导零会被隐藏，但与小数点相邻的数字除外。
- 小数点右侧的值被舍入为 PREC 参数所指定的小数点右侧的位数。
- 输出字符串的大小必须比小数点右侧的位数多至少三个字节。
- 输出字符串中的值为右对齐。

## ENO 报告的条件

在转换操作期间遇到错误时，将返回以下结果：

- ENO 设置为 0。
- OUT 设置为 0，或者如字符串到值的转换示例中所示。
- OUT 不变，或者如 OUT 为字符串时的示例中所示。

表格 8-32 ENO 状态

ENO	说明
1	无错误
0	非法或无效参数；例如，访问一个不存在的 DB
0	非法字符串，要求该字符串的最大长度为 0 或 255

8.2 字符串和字符

ENO	说明
0	非法字符串，当前长度大于最大长度
0	转换后的数值对于指定的 OUT 数据类型而言过大。
0	OUT 参数的最大字符串大小必须足够大，以接受参数 SIZE 所指定的字符数（从字符位置参数 P 开始）。
0	非法 P 值，P=0 或 P 大于当前字符串长度
0	参数 SIZE 必须大于参数 PREC。

表格 8-33 S\_CONV 字符串到值的转换示例

IN 字符串	OUT 数据类型	OUT 值	ENO
"123"	Int 或 DInt	123	TRUE
"-00456"	Int 或 DInt	-456	TRUE
"123.45"	Int 或 DInt	123	TRUE
"+2345"	Int 或 DInt	2345	TRUE
"00123AB"	Int 或 DInt	123	TRUE
"123"	Real	123.0	TRUE
"123.45"	Real	123.45	TRUE
"1.23e-4"	Real	1.23	TRUE
"1.23E-4"	Real	1.23	TRUE
"12,345.67"	Real	12345.67	TRUE
"3.4e39"	Real	3.4	TRUE
"-3.4e39"	Real	-3.4	TRUE
"1.17549e-38"	Real	1.17549	TRUE
"12345"	SInt	0	FALSE
"A123"	不适用	0	FALSE
""	不适用	0	FALSE
"++123"	不适用	0	FALSE
"+-123"	不适用	0	FALSE

表格 8-34 S\_CONV 值到字符串的转换示例

数据类型	IN 值	OUT 字符串	ENO
UInt	123	"123"	TRUE
UInt	0	"0"	TRUE
UDInt	12345678	"12345678"	TRUE
Real	-INF	"INF"	FALSE
Real	+INF	"INF"	FALSE
Real	NaN	"NaN"	FALSE

表格 8-35 STRG\_VAL 转换示例

IN 字符串	FORMAT (W#16#....)	OUT 数据类型	OUT 值	ENO
"123"	0000	Int 或 DInt	123	TRUE
"-00456"	0000	Int 或 DInt	-456	TRUE
"123.45"	0000	Int 或 DInt	123	TRUE
"+2345"	0000	Int 或 DInt	2345	TRUE
"00123AB"	0000	Int 或 DInt	123	TRUE
"123"	0000	Real	123.0	TRUE
"-00456"	0001	Real	-456.0	TRUE
"+00456"	0001	Real	456.0	TRUE
"123.45"	0000	Real	123.45	TRUE
"123.45"	0001	Real	12345.0	TRUE
"123,45"	0000	Real	12345.0	TRUE
"123,45"	0001	Real	123.45	TRUE
".00123AB"	0001	Real	123.0	TRUE
"1.23e-4"	0000	Real	1.23	TRUE
"1.23E-4"	0000	Real	1.23	TRUE
"1.23E-4"	0002	Real	1.23E-4	TRUE
"12,345.67"	0000	Real	12345.67	TRUE
"12,345.67"	0001	Real	12.345	TRUE

8.2 字符串和字符

IN 字符串	FORMAT (W#16#....)	OUT 数据类型	OUT 值	ENO
"3.4e39"	0002	Real	+INF	TRUE
"-3.4e39"	0002	Real	-INF	TRUE
"1.1754943e-38" (及更小值)	0002	Real	0.0	TRUE
"12345"	不适用	SInt	0	FALSE
"A123"	不适用	不适用	0	FALSE
""	不适用	不适用	0	FALSE
"++123"	不适用	不适用	0	FALSE
"+-123"	不适用	不适用	0	FALSE

下面的 VAL\_STRG 转换示例均基于按以下方式初始化的 OUT 字符串:

"Current Temp = xxxxxxxxxxx C"

其中字符“x”表示为转换后的值分配的空格字符。

表格 8- 36 VAL\_STRG 转换示例

数据类型	IN 值	P	SIZE	FORMAT (W#16#....)	PREC	OUT 字符串	ENO
UInt	123	16	10	0000	0	Current Temp = xxxxxxxx123 C	TRUE
UInt	0	16	10	0000	2	Current Temp = xxxxxx0.00 C	TRUE
UDInt	12345678	16	10	0000	3	Current Temp = x12345.678 C	TRUE
UDInt	12345678	16	10	0001	3	Current Temp = x12345,678 C	TRUE
Int	123	16	10	0004	0	Current Temp = xxxxxxx+123 C	TRUE
Int	-123	16	10	0004	0	Current Temp = xxxxxxx-123 C	TRUE
Real	-0.00123	16	10	0004	4	Current Temp = xxx-0.0012 C	TRUE



数据类型	IN 值	P	SIZE	FORMAT (W#16#....)	PREC	OUT 字符串	ENO
Real	-0.00123	16	10	0006	4	Current Temp = - 1.2300E-3 C	TRUE
Real	-INF	16	10	不适用	4	Current Temp = xxxxxxx-INF C	FALSE
Real	+INF	16	10	不适用	4	Current Temp = xxxxxxx+INF C	FALSE
Real	NaN	16	10	不适用	4	Current Temp = xxxxxxxxNaN C	FALSE
UDInt	12345678	16	6	不适用	3	Current Temp = xxxxxxxxxxxxx C	FALSE

### 8.2.3.2 字符串到字符和字符到字符串的转换

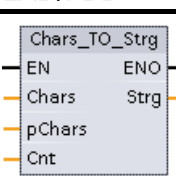
Chars\_TO\_Strg 将 ASCII 字符字节数组复制到字符串中。

Strg\_TO\_Chars 将 ASCII 字符串复制到字符字节数组中。

#### 说明

只允许将零基数组类型 (Array [0..n] of Char) 或 (Array [0..n] of Byte) 作为指令 Chars\_TO\_Strg 的输入参数 Chars，或作为指令 Strg\_TO\_Chars 的 IN\_OUT 参数 Chars。

表格 8-37 Chars\_TO\_Strg 指令

LAD/FBD	SCL	说明
	<pre>Chars_TO_Strg(   Chars:=_variant_in_,   pChars:=_dint_in_,   Cnt:=_uint_in_,   Strg=&gt;_string_out_);</pre>	<p>将字符数组的全部或部分复制到字符串。</p> <p>执行 Chars_TO_Strg 之前必须声明输出字符串。之后，Chars_TO_Strg 操作会覆盖该字符串。</p> <p>可以使用所有受支持的最大长度 (1..254) 的字符串。</p> <p>Chars_TO_Strg 操作不会更改字符串的最大长度值。达到最大字符串长度后，将停止从数组复制到字符串。</p> <p>字符数组中的 nul 字符“\$00”或 16#00 值起分隔符的作用，用于结束向字符串复制字符的操作。</p>

8.2 字符串和字符

表格 8- 38 参数的数据类型 (Chars\_TO\_Strg)

参数和类型		数据类型	说明
Chars	IN	Variant	Chars 参数是要转换为字符串的零基字符数组 [0..n] 的指针。可以在 DB 中声明数组，也可以在块接口中将其声明为本地变量。 示例: "DB1".MyArray 指向 DB1 中的 MyArray [0..10] of Char 元素值。
pChars	IN	Dint	数组中要复制的第一个字符的元素编号。默认值为数组元素 [0]。
Cnt	IN	UInt	要复制的字符数: 0 表示全部
Strg	OUT	String	目标字符串

表格 8- 39 Strg\_TO\_Chars 指令

LAD/FBD	SCL	说明
	<pre>Strg_TO_Chars(   Strg:=_string_in_,   pChars:=_dint_in_,   Cnt=&gt;_uint_out_,   Chars:=_variant_inout_);</pre>	<p>将整个输入字符串 Strg 复制到 IN_OUT 参数 Chars. 的字符数组中。</p> <p>该操作会从 pChars 参数指定的数组元素编号开始覆盖字节。</p> <p>可以使用所有受支持的最大长度 (1..254) 的字符串。</p> <p>结束分隔符不会被写入; 这由用户负责。要在最后写入的数组字符后面设置结束分隔符, 应使用下一数组元素编号 [pChars+Cnt]。</p>

表格 8- 40 参数的数据类型 (Strg\_TO\_Chars)

参数和类型		数据类型	说明
Strg	IN	String	源字符串
pChars	IN	Dint	写入目标数组的第一个字符串字符的数组元素编号

参数和类型		数据类型	说明
Chars	IN_OUT	Variant	Chars 参数是从输入字符串复制的零基字符数组 [0..n] 的指针。可以在 DB 中声明数组，也可以在块接口中将其声明为本地变量。 示例: "DB1".MyArray 指向 DB1 中的 MyArray [0..10] of Char 元素值。
Cnt	OUT	UInt	已复制的字符数

表格 8-41 ENO 状态

ENO	说明
1	无错误
0	Chars_TO_Strg: 尝试将多于字符串声明中的最大长度字节允许的字符字节复制到输出字符串
0	Chars_TO_Strg: 在输入字符字节数组中找到 nul 字符 (16#00) 值。
0	Strg_TO_Chars: 尝试将多于元素数目限制所允许的字符字节复制到输出数组

### 8.2.3.3 ASCII 到十六进制和十六进制到 ASCII 的转换

使用 ATH (ASCII 到十六进制) 和 HTA (十六进制到 ASCII) 指令进行 ASCII 字符字节 (仅字符 0 到 9 和 大写 A 到 F) 与相应的 4 位十六进制半字节之间的转换。

表格 8-42 ATH 指令

LAD/FBD	SCL	说明
	<pre>ret_val := ATH(     in:=_variant_in_,     n:=_int_in_,     out=&gt;_variant_out_);</pre>	将 ASCII 字符转换为压缩的十六进制数字。

表格 8-43 ATH 指令的数据类型

参数类型		数据类型	说明
IN	IN	Variant	指向 ASCII 字符字节数组的指针
N	IN	UInt	要转换的 ASCII 字符字节数

8.2 字符串和字符

参数类型		数据类型	说明
RET_VAL	OUT	Word	执行条件代码
OUT	OUT	Variant	指向转换后的十六进制字节数组的指针

转换从参数 IN 指定的位置开始，并持续 N 个字节。结果放置在 OUT 指定的位置。只能转换有效的 ASCII 字符 0 到 9 和大写 A 到 F。任何其它字符都将被转换为零。

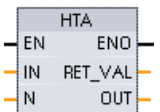
8 位 ASCII 编码的字符将被转换为 4 位十六进制半字节。两个 ASCII 字符可存储在单个字节中。

参数 IN 和 OUT 指定的是字节数组而不是十六进制 String 数据。ASCII 字符将被转换，并以其被读取的顺序放置在十六进制输出中。如果 ASCII 字符数为奇数，则在最后一个转换的十六进制数字的最右侧半字节中放置零。

表格 8-44 ASCII 到十六进制 (ATH) 转换示例

IN 字符字节	N	OUT 值	ENO
'0123'	4	W#16#0123	TRUE
'123AFx1a23'	10	16#123AF01023	FALSE
'a23'	3	W#16#A230	TRUE

表格 8-45 HTA 指令

LAD/FBD	SCL	说明
	<pre>ret_val := HTA(     in:=_variant_in_,     n:=_uint_in_,     out=&gt;_variant_out_);</pre>	将压缩的十六进制数字转换为相应的 ASCII 字符字节。

表格 8-46 HTA 指令的数据类型

参数和类型		数据类型	说明
IN	IN	Variant	指向输入字节数组的指针
N	IN	UInt	要转换的字节数（每个输入字节有两个 4 位半字节，并会生成 2N 个 ASCII 字符）
RET_VAL	OUT	Word	执行条件代码
OUT	OUT	Variant	指向 ASCII 字符字节数组的指针

转换从参数 IN 指定的位置开始，并持续 N 个字节。每个 4 位半字节都会转换为单个 8 位 ASCII 字符，并会生成 2N 个 ASCII 字符输出字节。全部 2N 个输出字节都会被写为 ASCII 字符 0 到 9 以及大写的 A 到 F。参数 OUT 指定一个字节数组，而不是字符串。

十六进制字节的每个半字节将按其读入的顺序转换为一个字符（首先转换十六进制数字最左侧的半字节，然后转换该字节最右侧的半字节）。

表格 8- 47 十六进制到 ASCII (HTA) 转换示例

IN 值	N	OUT 字符字节	ENO (执行 HTA 之后, ENO 始终为 TRUE)
W#16#0123	2	'0123'	TRUE
DW#16#123AF012	4	'123AF012'	TRUE

表格 8- 48 ATH and HTA 条件代码

RET_VAL (W#16#...)	说明	ENO
0000	无错误	TRUE
0007	无效的 ATH 输入字符: 发现不属于 ASCII 字符 0-9、小写 a-f 或大写 A-F 的字符	FALSE
8101	非法或无效的输入指针, 例如, 访问一个不存在的数据块。	FALSE
8120	输入字符串的格式无效, 即, 最大值 = 0、最大值 =255、当前值 > 最大值或允许的指针长度 < 最大值	FALSE
8182	输入缓冲区对于 N 来说过小	FALSE
8151	数据类型不允许用于输入缓冲区	FALSE
8301	非法或无效的输出指针, 例如, 访问一个不存在的数据块。	FALSE
8320	输出字符串的格式无效, 即, 最大值 = 0、最大值 =255、当前值 > 最大值或允许的指针长度 < 最大值	FALSE
8382	输出缓冲区对于 N 来说过小	FALSE
8351	数据类型不允许用于输出缓冲区	FALSE

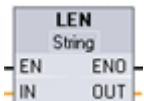
## 8.2.4 字符串操作指令

控制程序可以使用以下字符串和字符指令为操作员显示和过程日志创建消息。

8.2 字符串和字符

8.2.4.1 LEN

表格 8-49 长度指令

LAD/FBD	SCL	说明
	<pre>out := LEN(in);</pre>	LEN（长度）提供输出 OUT 处的字符串 IN 的当前长度。空字符串的长度为零。

表格 8-50 参数的数据类型

参数和类型		数据类型	说明
IN	IN	String	输入字符串
OUT	OUT	Int, DInt, Real, LReal	IN 字符串的有效字符数

表格 8-51 ENO 状态

ENO	条件	OUT
1	没有无效字符串条件	有效字符串长度
0	IN 的当前长度超出 IN 的最大长度	当前长度被设置为 0
	IN 的最大长度不在分配的存储范围内	
	IN 的最大长度为 255（非法长度）	

8.2.4.2 CONCAT

表格 8-52 连接字符串指令

LAD/FBD	SCL	说明
	<pre>out := CONCAT(in1, in2);</pre>	CONCAT（连接字符串）将字符串参数 IN1 和 IN2 连接成一个字符串，并在 OUT 输出。连接后，字符串 IN1 是组合字符串的左侧部分，而 IN2 是其右侧部分。

表格 8-53 参数的数据类型

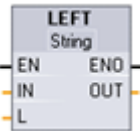
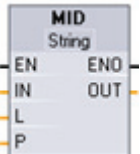
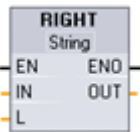
参数和类型		数据类型	说明
IN1	IN	String	输入字符串 1
IN2	IN	String	输入字符串 2
OUT	OUT	String	组合字符串（字符串 1 + 字符串 2）

表格 8-54 ENO 状态

ENO	条件	OUT
1	未检测到错误	有效字符
0	连接后的结果字符串比 OUT 字符串的最大长度长	复制结果字符串字符直到达到 OUT 的最大长度为止
	IN1 的当前长度超出 IN1 的最大长度，IN2 的当前长度超出 IN2 的最大长度，或 OUT 的当前长度超出 OUT 的最大长度（无效字符串）	当前长度被设置为 0
	IN1、IN2 或 OUT 的最大长度不在分配的存储范围内	
	IN1 或 IN2 的最大长度为 255，或 OUT 的最大长度为 0 或 255	

8.2.4.3 LEFT、RIGHT 和 MID

表格 8- 55 左侧、右侧和中间子串操作

LAD/FBD	SCL	说明
	<pre>out := LEFT(in, L);</pre>	<p>LEFT（左侧子串）提供由字符串参数 IN 的前 L 个字符所组成的子串。</p> <ul style="list-style-type: none"> <li>• 如果 L 大于 IN 字符串的当前长度，则在 OUT 中返回整个 IN 字符串。</li> <li>• 如果输入是空字符串，则在 OUT 中返回空字符串。</li> </ul>
	<pre>out := MID(in, L, p);</pre>	<p>MID（中间子串）提供字符串的中间部分。中间子串为从字符位置 P（包括该位置）开始的 L 个字符的长度。</p> <p>如果 L 和 P 的和超出字符串参数 IN 的当前长度，则返回从字符位置 P 开始并一直到 IN 字符串结尾的子串。</p>
	<pre>out := RIGHT(in, L);</pre>	<p>RIGHT（右侧子串）提供字符串的最后 L 个字符。</p> <ul style="list-style-type: none"> <li>• 如果 L 大于 IN 字符串的当前长度，则在参数 OUT 中返回整个 IN 字符串。</li> <li>• 如果输入是空字符串，则在 OUT 中返回空字符串。</li> </ul>

表格 8- 56 参数的数据类型

参数和类型	数据类型	说明
IN	IN	String 输入字符串
L	IN	Int 要创建的子串的长度： <ul style="list-style-type: none"> <li>• LEFT 使用字符串最左侧的字符数</li> <li>• RIGHT 使用字符串最右侧的字符数</li> <li>• MID 使用字符串中从位置 P 开始的字符数</li> </ul>
P	IN	Int 仅限 MID： 要复制的第一个子串字符的位置 P= 1, 表示 IN 字符串的起始字符位置
OUT	OUT	String 输出字符串



表格 8-57 ENO 状态

ENO	条件	OUT
1	未检测到错误	有效字符
0	<ul style="list-style-type: none"> <li>• L 或 P 小于或等于 0</li> <li>• P 大于 IN 的最大长度</li> <li>• IN 的当前长度超出 IN 的最大长度，或者 OUT 的当前长度超出 OUT 的最大长度</li> <li>• IN 或 OUT 的最大长度不在分配的存储范围内</li> <li>• IN 或 OUT 的最大长度为 0 或 255</li> </ul>	当前长度被设置为 0
	要复制的子串长度 (L) 比 OUT 字符串的最大长度长。	复制字符，直到达到 OUT 的最大长度为止
	仅限 MID: L 或 P 小于或等于 0	当前长度被设置为 0
	仅限 MID: P 大于 IN 的最大长度	
	IN1 的当前长度超出 IN1 的最大长度，或者 IN2 的当前长度超出 IN2 的最大长度（无效字符串）	当前长度被设置为 0
	IN1、IN2 或 OUT 的最大长度不在分配的存储范围内	
IN1、IN2 或 OUT 的最大长度为 0 或 255（非法长度）		

### 8.2.4.4 DELETE

表格 8-58 删除子串指令

LAD/FBD	SCL	说明
	<pre>out := DELETE(in, L, P);</pre>	<p>从字符串 IN 中删除 L 个字符。从字符位置 P（包括该位置）处开始删除字符，剩余字符串在参数 OUT 中输出。</p> <ul style="list-style-type: none"> <li>• 如果 L 等于零，则在 OUT 中返回输入字符串。</li> <li>• 如果 L 与 P 的和大于输入字符串的长度，则一直删除到该字符串的末尾。</li> </ul>

8.2 字符串和字符

表格 8-59 参数的数据类型

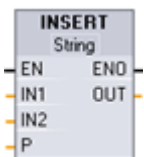
参数和类型		数据类型	说明
IN	IN	String	输入字符串
L	IN	Int	要删除的字符数
P	IN	Int	要删除的第一个字符的位置： IN 字符串的第一个字符的位置编号为 1
OUT	OUT	String	输出字符串

表格 8-60 ENO 状态

ENO	条件	OUT
1	未检测到错误	有效字符
0	P 大于 IN 的当前长度	将 IN 复制到 OUT 且不删除任何字符
	删除字符后的结果字符串比 OUT 字符串的最大长度长	复制结果字符串字符，直到达到 OUT 的最大长度为止
	L 小于 0，或者 P 小于或等于 0	当前长度被设置为 0
	IN 的当前长度超出 IN 的最大长度，或者 OUT 的当前长度超出 OUT 的最大长度	
	IN 或 OUT 的最大长度不在分配的存储范围内	
	IN 或 OUT 的最大长度为 0 或 255	

8.2.4.5 INSERT

表格 8-61 插入子串指令

LAD/FBD	SCL	说明
	<pre>out := INSERT(in1, in2, p);</pre>	<p>将字符串 IN2 插入字符串 IN1。在位置 P 的字符后开始插入。</p>

表格 8-62 参数的数据类型

参数和类型		数据类型	说明
IN1	IN	String	输入字符串 1
IN2	IN	String	输入字符串 2
P	IN	Int	字符串 IN1 中字符串 IN2 插入点前的最后一个字符位置 字符串 IN1 的第一个字符的位置编号为 1。
OUT	OUT	String	结果字符串

表格 8-63 ENO 状态

ENO	条件	OUT
1	未检测到错误	有效字符
0	P 大于 IN1 的长度	IN2 紧接最后一个 IN1 字符与 IN1 连接
	P 小于 0	当前长度被设置为 0
	插入后的结果字符串比 OUT 字符串的最大长度长	复制结果字符串字符，直到达到 OUT 的最大长度为止
	IN1 的当前长度超出 IN1 的最大长度，IN2 的当前长度超出 IN2 的最大长度，或 OUT 的当前长度超出 OUT 的最大长度（无效字符串）	当前长度被设置为 0
	IN1、IN2 或 OUT 的最大长度不在分配的存储范围内	
IN1 或 IN2 的最大长度为 255，或 OUT 的最大长度为 0 或 255		

### 8.2.4.6 REPLACE

表格 8-64 替换子串指令

LAD/FBD	SCL	说明
	<pre> out := REPLACE(     in1:=_string_in_,     in2:=_string_in_,     L:=_int_in_,     p:=_int_in_);         </pre>	替换字符串参数 IN1 中的 L 个字符。使用字符串参数 IN2 中的替换字符，从字符串 IN1 的字符位置 P（包括该位置）开始替换。

8.2 字符串和字符

表格 8-65 参数的数据类型

参数和类型		数据类型	说明
IN1	IN	String	输入字符串
IN2	IN	String	替换字符的字符串
L	IN	Int	要替换的字符数
P	IN	Int	要替换的第一个字符的位置
OUT	OUT	String	结果字符串

如果参数 L 等于零，则在字符串 IN1 的位置 P 处插入字符串 IN2，而不删除字符串 IN1 中的任何字符。

如果 P 等于 1，则用字符串 IN2 的字符替换字符串 IN1 的前 L 个字符。

表格 8-66 ENO 状态

ENO	条件	OUT
1	未检测到错误	有效字符
0	P 大于 IN1 的长度	IN2 紧接最后一个 IN1 字符与 IN1 连接
	P 在 IN1 范围内，但 IN1 中剩余的字符数小于 L	IN2 从位置 P 开始替换 IN1 的后端字符
	替换后的结果字符串比 OUT 字符串的最大长度长	复制结果字符串字符，直到达到 OUT 的最大长度为止
	IN1 的最大长度为 0	IN2 字符被复制到 OUT
	L 小于 0，或者 P 小于或等于 0	当前长度被设置为 0
	IN1 的当前长度超出 IN1 的最大长度，IN2 的当前长度超出 IN2 的最大长度，或 OUT 的当前长度超出 OUT 的最大长度	
	IN1、IN2 或 OUT 的最大长度不在分配的存储范围内	
IN1 或 IN2 的最大长度为 255，或 OUT 的最大长度为 0 或 255		

## 8.2.4.7 FIND

表格 8-67 查找子串指令

LAD/FBD	SCL	说明
	<pre>out := FIND(     in1:=_string_in_,     in2:=_string_in);</pre>	<p>提供由 IN2 指定的子串在字符串 IN1 中的字符位置。从左侧开始搜索。在 OUT 中返回 IN2 字符串第一次出现的字符位置。如果在字符串 IN1 中没有找到字符串 IN2，则返回零。</p>

表格 8-68 参数的数据类型

参数和类型	数据类型	说明	
IN1	IN	String	在该字符串内搜索
IN2	IN	String	搜索该字符串
OUT	OUT	Int	字符串 IN1 中第一个搜索匹配项的字符位置

表格 8-69 ENO 状态

ENO	条件	OUT
1	未检测到错误	有效字符位置
0	IN2 大于 IN1	字符位置被设置为 0
	IN1 的当前长度超出 IN1 的最大长度，或者 IN2 的当前长度超出 IN2 的最大长度（无效字符串）	
	IN1 或 IN2 的最大长度不在分配的存储范围内	
	IN1 或 IN2 的最大长度为 255	

## 8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

### 8.3.1 分布式 I/O 指令

可对 PROFINET、PROFIBUS 或 AS-i 使用以下分布式 I/O 指令：

- RDREC 指令 (页 291)： 可以从模块或设备中读取编号为 INDEX 的数据记录。
- WRREC 指令 (页 291)： 可以将编号为 INDEX 的数据记录传送到由 ID 定义的模块或设备。
- RALRM 指令 (页 294)： 可以从模块或设备接收中断及其全部相关信息，并将此信息提供给它的输出参数。
- DPRD\_DAT 指令 (页 302)： 必须利用 DPRD\_DAT 指令从模块或设备读取超过 64 个字节的一致性数据区。
- DPWR\_DAT 指令 (页 302)： 必须利用 DPWR\_DAT 指令将超过 64 个字节的一致性数据区写入模块或设备。

DPNRM\_DG 指令 (页 305)只能用于 PROFIBUS。可以“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。

### 8.3.2 RDREC 和 WRREC

可以对 PROFINET、PROFIBUS 和 AS-Interface 使用 RDREC（读取记录）和 WRREC（写入记录）指令。

表格 8-70 RDREC 和 WRREC 指令

LAD/FBD	SCL	说明
<p>"RDREC_DB"</p>	<pre>"RDREC_DB" (   req:=_bool_in_,   ID:=_word_in_,   index:=_dint_in_,   mlen:=_uint_in_,   valid=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_dword_out_,   len=&gt;_uint_out_,   record:=_variant_inout_);</pre>	<p>使用 RDREC 指令从通过 ID 寻址的组件（如中央机架或分布式组件（PROFIBUS DP 或 PROFINET IO））读取编号为 INDEX 的数据记录。在 MLEN 中分配要读取的最大字节数。目标区域 RECORD 的选定长度至少应该为 MLEN 个字节。</p>
<p>"WRREC_DB"</p>	<pre>"WRREC_DB" (   req:=_bool_in_,   ID:=_word_in_,   index:=_dint_in_,   len:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_dword_out_,   record:=_variant_inout_);</pre>	<p>使用 WRREC 指令将记录号为 INDEX 的数据 RECORD 传送到通过 ID 寻址的 DP 从站/PROFINET IO 设备组件，如中央机架上的模块或分布式组件（PROFIBUS DP 或 PROFINET IO）。</p> <p>分配要传送的数据记录的字节长度。因此，源区域 RECORD 的选定长度至少应该为 LEN 个字节。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“RDREC\_DB”和“WRREC\_DB”是背景 DB 的名称。

8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

表格 8- 71 RDREC 和 WRREC 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	REQ = 1: 传送数据记录
ID	IN	HW_IO (Word)	<p>DP 从站/PROFINET IO 组件 (模块或子模块) 的逻辑地址:</p> <ul style="list-style-type: none"> <li>• 对于输出模块, 必须将位 15 置位 (例如, 对于地址 5: ID:=DW#16#8005)。</li> <li>• 对于组合模块, 应指定两个地址中的较小者。</li> </ul> <p><b>注:</b> 可以通过以下两种方法之一来确定设备 ID:</p> <ul style="list-style-type: none"> <li>• 通过选择下列“网络视图”(Network view) 选项:                             <ul style="list-style-type: none"> <li>- “设备” (灰色框)</li> <li>- 设备的“属性”</li> <li>- “硬件标识符”</li> </ul> <b>注:</b> 然而, 并非所有设备都会显示硬件标识符。                         </li> <li>• 通过选择下列“项目树”(Project tree) 菜单选项:                             <ul style="list-style-type: none"> <li>- PLC 变量</li> <li>- 默认变量表</li> <li>- “系统常量”选项卡</li> </ul>                             将显示所有已组态的设备硬件标识符。                         </li> </ul>
INDEX	IN	Byte, Word, USInt, UInt, SInt, Int, DInt	数据记录号
MLEN	IN	Byte, USInt, UInt	要获取的数据记录信息的最大长度 (字节) (RDREC)
VALID	OUT	Bool	新数据记录已接收并且有效 (RDREC)。上一请求已完成且没有出错后, VALID 位将保持为 TRUE 一个扫描周期时间。
DONE	OUT	Bool	已传送数据记录 (WRREC)。上一请求已完成且没有出错后, DONE 位将保持为 TRUE 一个扫描周期时间。
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• BUSY = 1: 读取 (RDREC) 或写入 (WRREC) 过程尚未终止。</li> <li>• BUSY = 0: 数据记录传送已完成。</li> </ul>



## 8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

参数和类型		数据类型	说明
ERROR	OUT	Bool	ERROR = 1: 出现读取 (RDREC) 或写入 (WRREC) 错误。上一请求因错误而终止后, ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个扫描周期内有效。
STATUS	OUT	DWord	块状态或错误信息
LEN	OUT (RDREC) IN (WRREC)	UInt	<ul style="list-style-type: none"> <li>已获取的数据记录信息的长度 (RDREC)</li> <li>要传送的数据记录的最大长度 (字节) (WRREC)</li> </ul>
RECORD	IN_OUT	Variant	<ul style="list-style-type: none"> <li>已获取的数据记录的目标区域 (RDREC)</li> <li>数据记录 (WRREC)</li> </ul>

RDREC 和 WRREC 指令以异步方式运行, 即, 处理过程跨越多个指令调用。以 REQ = 1 调用 RDREC 或 WRREC 来启动作业。

通过输出参数 BUSY 和输出参数 STATUS 的两个中间字节显示作业状态。输出参数 BUSY 的值为 FALSE 时, 说明数据记录的传送完成。

输出参数 VALID (RDREC) 或 DONE (WRREC) 为 TRUE 时 (只持续一个扫描周期), 表示数据记录已成功传送到目标区域 RECORD (RDREC) 或目标设备 (WRREC)。使用 RDREC 时, 输出参数 LEN 包含所获取数据的长度 (字节)。

输出参数 ERROR (只在 ERROR = TRUE 时持续一个扫描周期) 表示发生数据记录传送错误。在这种情况下, 输出参数 STATUS (只在 ERROR = TRUE 时持续一个扫描周期) 包含错误信息。

由硬件设备制造商定义数据记录。有关数据记录的详细信息, 请参见硬件设备制造商的设备文档。

**说明**

如果已通过 GSD 文件 (GSD 修订版 3 及更高版本) 组态 DPV1 从站且 DP 主站的 DP 接口已设置为“S7 兼容”(S7 compatible), 则可能无法在用户程序中通过“RDREC”从 I/O 模块读取数据记录, 或通过“WRREC”将记录写入 I/O 模块。这种情况下, DP 主站寻址错误的插槽 (组态的插槽 + 3)。

解决方法: 将 DP 主站的接口设置为“DPV1”。

**说明**

“RDREC”和“WRREC”指令的接口与“符合 IEC 61131-3 的 PROFIBUS 准则、PROFIBUS 通信和代理函数块”中定义的“RDREC”和“WRREC”FB 完全相同。

8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

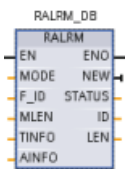
**说明**

如果使用“RDREC”或“WRREC”读或写 PROFINET IO 的数据记录，则 INDEX、MLEN 和 LEN 参数中的负值将被解释为无符号 16 位整型值。

**8.3.3 RALRM**

可以对 PROFINET 和 PROFIBUS 使用 RALRM（读取报警）指令。

表格 8-72 RALRM 指令

LAD/FBD	SCL	说明
	<pre>"RALRM_DB" (     mode:=_int_in_,     f_ID:=_word_in_,     mlen:=_uint_in_,     new=&gt;_bool_out_,     status=&gt;_dword_out_,     ID=&gt;_word_out_,     len=&gt;_uint_out_,     tinfo:=_variant_inout_,     ainfo:=_variant_inout_);</pre>	<p>使用 RALRM（读取报警）指令从 PROFIBUS 或 PROFINET I/O 模块/设备读取诊断中断信息。</p> <p>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。</p> <p>在中断 OB 中调用 RALRM，可返回导致中断的事件的相关信息。在 S7-1200 中，仅支持诊断中断 (OB82)。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“RALRM\_DB”是背景 DB 的名称。

## 8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

表格 8-73 参数的数据类型

参数和类型		数据类型	说明
MODE	IN	Byte, USInt, SInt, Int	工作模式
F_ID	IN	HW_IO (Word)	<p>组件（模块）的逻辑起始地址，将从该地址接收中断</p> <p><b>注：</b> 可以通过以下两种方法之一来确定设备 ID：</p> <ul style="list-style-type: none"> <li>• 通过选择下列“网络视图”(Network view) 选项： <ul style="list-style-type: none"> <li>- “设备”（灰色框）</li> <li>- 设备的“属性”</li> <li>- “硬件标识符”</li> </ul> <p><b>注：</b> 并非所有设备都会显示硬件标识符。</p> </li> <li>• 通过选择下列“项目树”(Project tree) 菜单选项： <ul style="list-style-type: none"> <li>- PLC 变量</li> <li>- 默认变量表</li> <li>- “系统常量”选项卡</li> <li>- 将显示所有已组态的设备硬件标识符。</li> </ul> </li> </ul>
MLEN	IN	Byte, USInt, UInt	要接收的数据中断信息的最大长度（字节）。如果 MLEN 为 0，则允许接收的数据中断信息量与 AINFO 目标区域中提供的数据中断信息量相同。
NEW	OUT	Bool	已接收新中断。
STATUS	OUT	DWord	RALRM 指令的状态。有关详细信息，请参见“RDREC、WRREC 和 RALRM 的 STATUS 参数”(页 297)。
ID	OUT	HW_IO (Word)	<p>导致诊断中断的 I/O 模块的硬件标识符</p> <p><b>注：</b> 有关如何确定设备 ID 的说明，请参见参数 F_ID。</p>
LEN	OUT	DWord, UInt, UDIInt, DInt, Real, LReal	已接收的 AINFO 中断信息的长度
TINFO	IN_OUT	Variant	任务信息： OB 启动和管理信息的目标范围。TINFO 的长度始终为 32 个字节。
AINFO	IN_OUT	Variant	中断信息： 头信息和附加中断信息的目标区域。对于 AINFO，如果 MLEN 大于 0，则提供的长度至少为 MLEN 个字节。AINFO 的长度是可变的。

8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

**说明**

如果在启动事件不是 I/O 中断的 OB 中调用“RALRM”，该指令在输出中提供的信息会相应减少。

在不同的 OB 中调用“RALRM”时，务必要使用不同的背景数据块。如果评估在关联中断 OB 外部调用“RALRM”得出的数据，则应对每个 OB 启动事件单独使用一个数据块。

**说明**

“RALRM”指令的接口与“符合 IEC 61131-3 的 PROFIBUS 准则、PROFIBUS 通信和代理函数块”中定义的“RALRM”FB 完全相同。

**调用 RALRM**

可以在三种不同的操作模式 (MODE) 下调用 RALRM 指令。

表格 8-74 RALRM 指令操作模式

MODE	说明
0	<ul style="list-style-type: none"> <li>• ID 包含触发中断的 I/O 模块的硬件标识符。</li> <li>• 输出参数 NEW 被设置为 TRUE。</li> <li>• LEN 产生 0 输出。</li> <li>• AINFO 和 TINFO 不使用任何信息进行更新。</li> </ul>
1	<ul style="list-style-type: none"> <li>• ID 包含触发中断的 I/O 模块的硬件标识符。</li> <li>• 输出参数 NEW 被设置为 TRUE。</li> <li>• LEN 产生输出，数量为返回的 AINFO 数据的字节数。</li> <li>• AINFO 和 TINFO 使用中断相关信息进行更新。</li> </ul>
2	<p>如果分配给输入参数 F_ID 的硬件标识符触发了中断，则：</p> <ul style="list-style-type: none"> <li>• ID 包含触发中断的 I/O 模块的硬件标识符。应与 F_ID 处的值相同。</li> <li>• 输出参数 NEW 被设置为 TRUE。</li> <li>• LEN 产生输出，数量为返回的 AINFO 数据的字节数。</li> <li>• AINFO 和 TINFO 使用中断相关信息进行更新。</li> </ul>

## 8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

**说明**

如果为过短的 TINFO 或 AINFO 指定目标区域，则 RALRM 无法返回完整信息。

MLEN 可限制所返回 AINFO 数据的量。

有关如何解释 TINFO 和 AINFO 数据的信息，请参见 STEP 7 在线信息系统的 AINFO 参数和 TINFO 参数。

**8.3.4 RDREC、WRREC 和 RALRM 的 STATUS 参数**

输出参数 STATUS 包含被解释为 ARRAY[1...4] OF BYTE 的错误信息，其结构如下：

表格 8-75 STATUS 输出数组

数组元素	名称	说明
STATUS[1]	Function_Nu m	<ul style="list-style-type: none"> <li>• B#16#00（如果无错误）</li> <li>• DPV1-PDU 的功能 ID：如果发生错误，则对 B#16#80 进行或运算（对于读取数据记录：B#16#DE；对于写入数据记录：B#16#DF）。如果未使用 DPV1 协议元素，则输出 B#16#C0。</li> </ul>
STATUS[2]	Error Decode	错误 ID 的位置
STATUS[3]	Error_Code_1	错误 ID
STATUS[4]	Error_Code_2	制造商特定的错误 ID 扩展

表格 8-76 STATUS[2] 值

Error_decode (B#16#....)	来源	说明
00 到 7F	CPU	无错误或无警告
80	DPV1	因不符合 IEC 61158-6 而出错
81 到 8F	CPU	B#16#8x 表示在指令的第“x”个调用参数中存在错误。
FE、FF	DP 配置文件	配置文件特定错误

表格 8-77 STATUS[3] 值

Error_decode (B#16#....)	Error_code_1 (B#16#....)	解释 (DVP1)	说明
00	00		无错误, 无警告
70	00	保留, 拒绝	初始调用; 未传送活动数据记录
	01	保留, 拒绝	初始调用; 已开始传送数据记录
	02	保留, 拒绝	中间调用; 已激活数据记录传送
80	90	保留, 通过	逻辑起始地址无效
	92	保留, 通过	Variant 指针的类型非法
	93	保留, 通过	通过 ID 或 F_ID 寻址的 DP 组件未组态。
	96		“RALRM (页 294)”不能提供 OB 启动信息、管理信息、标题信息或附加中断信息。 对于 OB 4x、55、56、57、82 和 83, 可以使用“DPNRM_DG (页 305)”指令异步读取相关 DP 从站的当前诊断消息帧 (OB 启动信息中的地址信息)。
	A0	读取错误	读取模块时得到否定确认。
	A1	写错误	写入模块时得到否定确认
	A2	模块故障	第 2 层出现 DP 协议错误 (例如, 从站故障或总线故障)
	A3	保留, 通过	<ul style="list-style-type: none"> <li>• PROFIBUS DP: 直接数据链路映射器或用户接口/用户出现 DP 协议错误</li> <li>• PROFINET IO: 常规 CM 错误</li> </ul>
	A4	保留, 通过	通信总线上的通信中断
	A5	保留, 通过	-
	A7	保留, 通过	DP 从站或模块已被占用 (临时错误)。
	A8	版本冲突	DP 从站或模块报告出现不兼容版本。
	A9	特性不受支持	特性不受 DP 从站或模块支持
	AA 到 AF	用户特定	DP 从站或模块报告其应用中出现制造商特定错误。请检查 DP 从站或模块的制造商提供的文档。
B0	索引无效	模块中出现未知数据记录; 非法数据记录编号 ≥ 256	

## 8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

Error_decode (B#16#....)	Error_code_1 (B#16#....)	解释 (DVP1)	说明
	B1	写长度错误	<p>RECORD 参数中的长度信息不正确。</p> <ul style="list-style-type: none"> <li>对于“RALRM”：AINFO 中的长度错误</li> </ul> <p><b>注意：</b>要立即访问有关如何解释“AINFO”返回缓冲区的信息，请参见 STEP 7 的在线信息系统。</p> <ul style="list-style-type: none"> <li>对于“RDREC (页 291)”和“WRREC (页 291)”：“MLEN”中的长度错误</li> </ul>
	B2	无效插槽	组态的插槽未被占用。
	B3	类型冲突	实际模块类型与指定的模块类型不匹配。
	B4	无效区域	DP 从站或模块报告对无效区域的访问。
	B5	状态冲突	DP 从站或模块未就绪
	B6	访问被拒绝	DP 从站或模块拒绝访问。
	B7	无效范围	DP 从站或模块报告参数或值的范围无效。
	B8	无效参数	DP 从站或模块报告参数无效。
	B9	无效类型	<p>DP 从站或模块报告类型无效：</p> <ul style="list-style-type: none"> <li>对于“RDREC (页 291)”：缓冲区过小（无法读取子网）</li> <li>对于“WRREC (页 291)”：缓冲区过小（无法写入子网）</li> </ul>
	BA 到 BF	用户特定	DP 从站或模块在访问时报告制造商特定错误。请检查 DP 从站或模块的制造商提供的文档。
	C0	读限制冲突	<ul style="list-style-type: none"> <li>对于“WRREC (页 291)”：仅当 CPU 处于 STOP 模式时才能写入数据。 <b>注意：</b>这意味着无法通过用户程序写入数据。只能使用 PG/PC 在线写入数据。</li> <li>对于“RDREC (页 291)”：模块可发送数据记录，但没有数据，或仅当 CPU 处于 STOP 模式时才能读取数据。 <b>注意：</b>如果仅当 CPU 处于 STOP 模式时才能读取数据，则用户程序无法进行评估。在这种情况下，只能使用 PG/PC 在线读取数据。</li> </ul>

8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

Error_decode (B#16#....)	Error_code_1 (B#16#....)	解释 (DVP1)	说明
	C1	写限制冲突	针对相同数据记录向模块发送的上一次写请求的数据尚未被该模块处理。
	C2	资源忙	模块正在处理 CPU 允许的最多作业数。
	C3	资源不可用	所需操作资源当前已被占用。
	C4		内部临时错误。无法执行作业。 重复作业。如果此错误经常发生，请检查安装的电气干扰源。
	C5		DP 从站或模块不可用
	C6		由于取消优先级而使数据记录传送被取消。
	C7		作业由于 DP 主站的暖启动或冷启动而中止。
	C8 到 CF		DP 从站或模块报告制造商特定资源错误。请检查 DP 从站或模块的制造商提供的文档。
	Dx	用户特定	DP 从站特定。参见 DP 从站的说明。
81	00 到 FF		初始调用参数错误 (对于“RALRM (页 294)”: MODE) MODE)
	00		非法工作模式
82	00 到 FF		第二个调用参数错误
88	00 到 FF		第八个调用参数错误 (对于“RALRM (页 294)”: TINFO) <b>注意:</b> 要立即访问有关如何解释“TINFO”返回缓冲区的信息, 请参见 STEP 7 的在线信息系统。
	01		语法 ID 错误
	23		超出数量结构或目标区域过小
	24		范围 ID 错误
	32		DB/DI 号超出用户范围
	3A		区域 ID DB/DI 的 DB/DI 号为 NULL, 或指定的 DB/DI 不存在。
89	00 到 FF		第九个调用参数错误 (对于“RALRM (页 294)”: AINFO) <b>注意:</b> 要立即访问有关如何解释“AINFO”返回缓冲区的信息, 请参见 STEP 7 的在线信息系统。



## 8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

Error_decode (B#16#....)	Error_code_1 (B#16#....)	解释 (DVP1)	说明
	01		语法 ID 错误
	23		超出数量结构或目标区域过小
	24		范围 ID 错误
	32		DB/DI 号超出用户范围
	3A		区域 ID DB/DI 的 DB/DI 号为 NULL, 或指定的 DB/DI 不存在。
8A	00 到 FF		第 10 个调用参数错误
8F	00 到 FF		第 15 个调用参数错误
FE、FF	00 到 FF		配置文件特定错误

**数组元素 STATUS[4]**

出现 DPV1 错误时, DP 主机将 STATUS[4] 传递给 CPU 和指令。如果没有 DPV1 错误, 该值将被设置为 0, 但对于 RDREC 有以下例外情况:


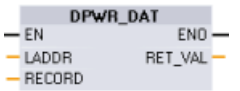
- 如果  $MLEN > RECORD$  中的目标区域长度, 则 STATUS[4] 包含 RECORD 中的目标区域长度。
- 如果实际数据记录长度  $< MLEN < RECORD$  中的目标区域长度, 则 STATUS[4] = MLEN。
- 如果必须设置  $STATUS[4] > 255$ , 则 STATUS[4] = 0

在 PROFINET IO 中, STATUS[4] 的值为 0。

### 8.3.5 DPRD\_DAT 和 DPWR\_DAT

可以对 PROFINET 和 PROFIBUS 使用 DPRD\_DAT (读取一致性数据) 和 DPWR\_DAT (写入一致性数据) 指令。

表格 8-78 DPRD\_DAT 和 DPWR\_DAT 指令

LAD/FBD	SCL	说明
	<pre>ret_val := DPRD_DAT(     laddr:=_word_in_,     record=&gt;_variant_out_);</pre>	<p>使用 DPRD_DAT 指令可读取 DP 标准从站 /PROFINET IO 设备的一致性数据。如果数据传递过程中未出错，则已读取的数据将被输入到通过 RECORD 参数设置的目标区域中。目标区域的长度必须与通过 STEP 7 为所选模块组态的长度相同。调用 DPRD_DAT 指令时，只能访问已组态的起始地址下的一个模块/DP 标识的数据。</p>
	<pre>ret_val := DPWR_DAT(     laddr:=_word_in_,     record:=_variant_in_);</pre>	<p>使用 DPWR_DAT 指令可将 RECORD 中的数据一致性地传送到已寻址的 DP 标准从站/PROFINET IO 设备。源区域的长度必须与通过 STEP 7 为所选模块组态的长度相同。</p>

CPU 最多支持 64 个字节的一致性数据。对于超过 64 个字节的一致性数据区域，必须使用 DPRD\_DAT 和 DPWR\_DAT 指令。如果需要，可对 1 个字节或更大的数据区域使用这些指令。如果访问被拒绝，将出现错误代码 W#16#8090。

#### 说明

如果正在对一致性数据使用 DPRD\_DAT 和 DPWR\_DAT 指令，则必须从过程映像自动更新中删除该一致性数据。有关详细信息，请参见“PLC 概念：用户程序的执行”(页 69)。

表格 8-79 参数的数据类型

参数和类型		数据类型	说明
LADDR	IN	HW_IO (Word)	<ul style="list-style-type: none"> <li>模块的“I”区域中已组态的起始地址，将从该处读取数据 (DPRD_DAT)</li> <li>模块的过程映像输出区域中已组态的起始地址，数据将被写入该地址 (DPWR_DAT)</li> </ul> 必须以十六进制格式输入地址（例如，输入或输出地址 100 表示：LADDR:=W#16#64）。
RECORD	OUT	Variant	已读取的用户数据的目标区域 (DPRD_DAT) 或要写入的用户数据的源区域 (DPWR_DAT)。此区域的大小必须与通过 STEP 7 为所选模块组态的区域大小完全相同。只允许使用数据类型 Byte。
RET_VAL	OUT	Int	如果在此函数已激活情况下发生错误，则返回值中将包含一个错误代码。

### DPRD\_DAT 操作

目标区域的长度必须与使用 STEP 7 为所选模块组态的长度相同。如果数据传送期间未出现错误，则会将已读取的数据输入到由 RECORD 标识的目标区域中。

如果从具有模块化设计或具有多个 DP 标识符的 DP 标准从站读取数据，则通过指定组态的起始地址，每次 DPRD\_DAT 调用只能访问一个模块/DP 标识符的数据。

### DPWR\_DAT 操作

将 RECORD 中的数据一致性地传送到已寻址的 DP 标准从站/PROFINET IO 中。数据以同步方式传送，即，指令完成时写入过程即完成。

源区域的长度必须与通过 STEP 7 为所选模块组态的长度相同。

如果 DP 标准从站具有模块化设计，则只能访问 DP 从站的一个模块。

表格 8-80 DPRD\_DAT 和 DPWR\_DAT 错误代码

错误代码	说明
0000	未出错
808x	外部 DP 接口模块出现系统错误

8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

错误代码	说明
8090	适用于下列情况之一： <ul style="list-style-type: none"> <li>没有为指定的逻辑基址组态模块。</li> <li>忽略了有关一致性数据长度的限制。</li> <li>没有以十六进制格式在 LADDR 参数中输入起始地址。</li> </ul>
8092	在 Any 引用中指定了一个非 Byte 的类型。
8093	LADDR 中指定的逻辑地址处不存在可以从其中读取 (DPRD_DAT) 一致性数据或向其写入 (DPWR_DAT) 一致性数据的 DP 模块/PROFINET IO 设备。
80A0	访问 I/O 设备时检测到访问错误 (DPRD_DAT)。
80A1	访问 I/O 设备时检测到访问错误 (DPWR_DAT)。
80B0	外部 DP 接口模块出现从站故障
80B1	指定的目标区域的长度 (DPRD_DAT) 或源区域的长度 (DPWR_DAT) 与通过 STEP 7 Basic 组态的用户数据的长度不同。
80B2、80B3、80C2、80Fx	外部 DP 接口模块出现系统错误 (DPRD_DAT) 和 (DPWR_DAT)
87xy、808x	外部 DP 接口模块出现系统错误 (DPRD_DAT)
85xy	外部 DP 接口模块出现系统错误 (DPWR_DAT)
80C0	模块尚未读取数据 (DPRD_DAT)。
80C1	模块尚未处理模块中前一次写入作业的数据 (DPWR_DAT)。
8xyy <sup>1</sup>	常规错误信息

有关常规错误代码的更多信息，请参见“扩展指令，分布式 I/O：RDREC、WRREC 和 RALRM 的错误信息” (页 297)。

**说明**

如果访问 DPV1 从站，这些从站的错误信息可从 DP 主站转发到指令。

## 8.3.6 DPNRM\_DG

可以对 PROFIBUS 使用 DPNRM\_DG (读取诊断数据) 指令。

表格 8- 81 DPNRM\_DG 指令

LAD/FBD	SCL	说明
	<pre>ret_val := DPNRM_DG(   req:=_bool_in_,   laddr:=_word_in_,   record=&gt;_variant_out_,   busy=&gt;_bool_out_);</pre>	<p>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。在顺利完成数据传输后，已读取的数据被输入到由 RECORD 指定的目标区域。</p>

表格 8- 82 DPNRM\_DG 指令的参数数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	REQ=1: 读取请求
LADDR	IN	HW_DPSLAVE	<p>所组态的 DP 从站诊断地址: 必须是该站点的地址, 而不是 I/O 设备的地址。在“网络”(Network) 视图的“设备组态”(Device configuration) 中, 选择相应站 (不是设备的图像) 来确定诊断地址。</p> <p>输入十六进制格式的地址。例如, 诊断地址 1022 表示为 LADDR:=W#16#3FE。</p>
RET_VAL	OUT	Int	如果在此函数已激活情况下发生错误, 则返回值中将包含一个错误代码。如果没有错误, 则将实际传输的数据长度输入 RET_VAL。
RECORD	OUT	Variant	<p>已读取的诊断数据的目标区域。只允许 Byte 数据类型。要读取的数据记录或目标区域的最小长度为 6。要发送的数据记录的最大长度为 240。</p> <p>标准从站可提供大于 240 字节的诊断数据, 最高可达 244 字节。在这种情况下, 将前 240 字节传送到目标区域, 并在数据中设置溢出位。</p>
BUSY	OUT	Bool	BUSY=1: 读取作业还未完成

通过在调用 DPNRM\_DG 指令时将值 1 赋给输入参数 REQ 启动读取作业。读取作业异步执行, 即需要多次调用 DPNRM\_DG 指令。此作业的状态由输出参数 RET\_VAL 和 BUSY 指示。

8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

表格 8- 83 从站诊断数据结构

字节	说明
0	站状态 1
1	站状态 2
2	站状态 3
3	主站号
4	供应商 ID (高字节)
5	供应商 ID (低字节)
6 ...	附加从站诊断信息

表格 8- 84 DPNRM\_DG 指令的错误代码

错误代码	说明	限制
0000	无错误	-
7000	首次调用时 REQ = 0: 没有激活的数据传输; BUSY 的值为 0。	-
7001	首次调用时 REQ = 1: 没有激活的数据传输; BUSY 的值为 1。	分布式 I/O
7002	临时调用 (与 REQ 无关): 数据传送已经激活; BUSY 的值为 1。	分布式 I/O
8090	指定的逻辑基址无效: 无基址。	-
8092	在 Any 引用中指定的类型不是 Byte。	-
8093	<ul style="list-style-type: none"> <li>通过 LADDR 指定的模块不允许使用此指令 (适用于 S7-1200 的 S7-DP 模块允许使用)。</li> <li>LADDR 指定 I/O 设备, 而不是站。在“网络”(Network) 视图的“设备组态”(Device configuration) 中, 选择相应站 (不是设备的图像) 来确定 LADDR 的诊断地址。</li> </ul>	-
80A2	<ul style="list-style-type: none"> <li>第 2 层出现 DP 协议错误 (例如, 从站故障或总线故障)</li> <li>对于 ET200S, 无法在 DPV0 模式下读取数据记录。</li> </ul>	分布式 I/O
80A3	用户接口/用户中出现 DP 协议错误	分布式 I/O
80A4	通信总线上出现通信故障	CPU 与外部 DP 接口模块之间发生错误。

## 8.3 分布式 I/O (PROFINET、PROFIBUS 或 AS-Interface)

错误代码	说明	限制
80B0	<ul style="list-style-type: none"> <li>相应模块类型不支持此指令。</li> <li>该模块不能识别数据记录。</li> <li>不允许使用数据目录号 241。</li> </ul>	-
80B1	在 RECORD 参数中指定的长度不正确。	指定长度 > 记录长度
80B2	组态的插槽未被占用。	-
80B3	实际模块类型与要求的模块类型不匹配。	-
80C0	无诊断信息。	-
80C1	模块尚未处理其中同一数据记录前一次写入作业的数据。	-
80C2	模块正在处理 CPU 允许的最多作业数。	-
80C3	所需资源（存储器等）当前被占用。	-
80C4	内部临时错误。作业无法处理。 请重复该作业。如果此错误频繁发生，请检查系统是否存在电干扰源。	-
80C5	分布式 I/O 不可用。	分布式 I/O
80C6	数据记录传送因优先等级中止（重启或后台）而停止	分布式 I/O
8xyy <sup>1</sup>	常规错误代码	

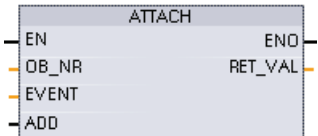
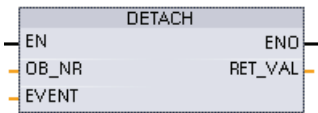
有关常规错误代码的更多信息，请参见“扩展指令，分布式 I/O：RDREC、WRREC 和 RALRM 的错误信息” (页 297)。

## 8.4 中断

### 8.4.1 附加和分离指令

使用 ATTACH 和 DETACH 指令可激活和禁用于中断事件驱动的子程序。

表格 8- 85 ATTACH 和 DETACH 指令

LAD/FBD	SCL	说明
	<pre>ret_val := ATTACH(     ob_nr:=_int_in_,     event:=_event_att_in_,     add:=_bool_in_);</pre>	ATTACH 启用响应硬件中断事件的中断 OB 子程序执行。
	<pre>ret_val := DETACH(     ob_nr:=_int_in_,     event:=_event_att_in_);</pre>	DETACH 禁用响应硬件中断事件的中断 OB 子程序执行。

表格 8- 86 参数的数据类型

参数和类型		数据类型	说明
OB_NR	IN	OB_ATT	组织块标识符：从使用“添加新块”(Add new block) 功能创建的可用硬件中断 OB 中进行选择。双击该参数域，然后单击助手图标可查看可用的 OB。
EVENT	IN	EVENT_ATT	事件标识符：从在 PLC 设备组态中为数字输入或高速计数器启用的可用硬件中断事件中进行选择。双击该参数域，然后单击助手图标可查看这些可用事件。
ADD (仅限 ATTACH)	IN	Bool	<ul style="list-style-type: none"> <li>• ADD = 0 (默认值)：该事件将取代先前为此 OB 附加的所有事件。</li> <li>• ADD = 1：该事件将添加到先前为此 OB 附加的事件中。</li> </ul>
RET_VAL	OUT	Int	执行条件代码



## 硬件中断事件

CPU 支持以下硬件中断事件：

- 上升沿事件（所有内置 CPU 数字量输入和 SB 数字量输入）
  - 数字输入从 OFF 切换为 ON 时会出现上升沿，以响应连接到输入的现场设备的信号变化。
- 下降沿事件（所有内置 CPU 数字量输入和 SB 数字量输入）
  - 数字输入从 ON 切换为 OFF 时会出现下降沿。
- 高速计数器 (HSC) 当前值 = 参考值 (CV = RV) 事件 (HSC 1 至 6)
  - 当前计数值从相邻值变为与先前设置的参考值完全匹配时，会生成 HSC 的 CV = RV 中断。
- HSC 方向变化事件 (HSC 1 至 6)
  - 当检测到 HSC 从增大变为减小或从减小变为增大时，会发生方向变化事件。
- HSC 外部复位事件 (HSC 1 至 6)
  - 某些 HSC 模式允许分配一个数字输入作为外部复位端，用于将 HSC 的计数值重置为零。当该输入从 OFF 切换为 ON 时，会发生此类 HSC 的外部复位事件。

## 在设备组态期间启用硬件中断事件

必须在设备组态中启用硬件中断。如果要在组态或运行期间附加此事件，则必须在设备组态中为数字输入通道或 HSC 选中启用事件框。

PLC 设备组态中的复选框选项：

- 数字量输入
  - 启用上升沿检测
  - 启用下降沿检测
- 高速计数器 (HSC)
  - 启用此高速计数器
  - 生成计数器值等于参考计数值的中断
  - 生成外部复位事件的中断
  - 生成方向变化事件的中断

### 向用户程序添加新硬件中断 OB 代码块

默认情况下，第一次启用事件时，没有任何 OB 附加到该事件。这可通过标签“HW 中断:”(HW interrupt:) 设备组态 "<未连接>" 来进行标识。只有硬件中断 OB 能附加到硬件中断事件。所有现有的硬件中断 OB 都会出现在“HW 中断:”(HW interrupt:) 下拉列表中。如果未列出任何 OB，则必须按下列步骤创建类型为“硬件中断”的 OB。在项目树的“程序块”(Program blocks) 分支下：

1. 双击“添加新块”(Add new block)，选择“组织块 (OB)”(Organization block (OB))，然后选择“硬件中断”(Hardware interrupt)。
2. 也可以重命名 OB、选择编程语言 (LAD 或 FBD) 以及选择块编号 (切换为手动并选择与建议块编号不同的块编号)。
3. 编辑该 OB，添加事件发生时要执行的已编程响应。可以从此 OB 调用最多嵌套四层深的 FC 和 FB。

### OB\_NR 参数

所有现有的硬件中断 OB 名称都会出现在“HW 中断:”(HW interrupt:) 下拉列表和 ATTACH / DETACH 参数 OB\_NR 下拉列表中。

### EVENT 参数

启用某个硬件中断事件时，将为该事件分配一个唯一的默认事件名称。可以通过编辑“事件名称:”(Event name:) 编辑框更改该事件的名称，但该名称必须唯一。这些事件名称将成为“常量”(Constants) 变量表中的变量名称，并显示在 ATTACH 和 DETACH 指令框的 EVENT 参数下拉列表中。变量的值是用于标识事件的内部编号。

### 常规操作

每个硬件事件都可附加到一个硬件中断 OB 中，在发生该硬件中断事件时将排队执行该硬件中断 OB。在组态或运行期间可附加 OB 事件。

用户可以在组态时将 OB 附加到已启用的事件或使其与该事件分离。要在组态过程中向事件附加一个 OB，必须使用“HW 中断:”(HW interrupt:) 下拉列表 (单击右侧的向下箭头)，然后从可用的硬件中断 OB 列表选择一个 OB。从该列表中选择相应的 OB 名称，或者选择“<未连接>”( <not connected>) 以删除该附加关系。

也可以在运行期间附加或分离已启用的硬件中断事件。在运行期间使用 ATTACH 或 DETACH 程序指令 (如有必要可多次使用) 将已启用的中断事件附加到相应的 OB 或与其分离。如果当前未附加到任何 OB (选择了设备组态中的“<未连接>”( <not connected>) 选项或由于执行了 DETACH 指令)，则将忽略已启用的硬件中断事件。

## DETACH 操作

使用 DETACH 指令将特定事件或所有事件与特定 OB 分离。如果指定了 EVENT，则仅将该事件与指定的 OB\_NR 分离；当前附加到此 OB\_NR 的任何其它事件仍保持附加状态。如果未指定 EVENT，则分离当前附加到 OB\_NR 的所有事件。

## 条件代码


表格 8- 87 条件代码

RET_VAL (W#16#...)	ENO	说明
0000	1	无错误
0001	1	没有要分离的事件（仅 DETACH）
8090	0	OB 不存在
8091	0	OB 类型错误
8093	0	事件不存在

## 8.4.2 循环中断

### 8.4.2.1 SET\_CINT（Set 循环中断）

表格 8- 88 SET\_CINT（Set 循环中断指令）

LAD/FBD	SCL	说明
	<pre>ret_val := SET_CINT(   ob_nr:=_int_in_,   cycle:=_udint_in_,   phase:=_udint_in_);</pre>	设置特定的中断 OB 以开始循环中断程序扫描过程。

表格 8- 89 参数的数据类型

参数和类型		数据类型	说明
OB_NR	IN	OB_CYCLIC	OB 号（允许使用符号名称）
CYCLE	IN	UDInt	时间间隔（微秒）

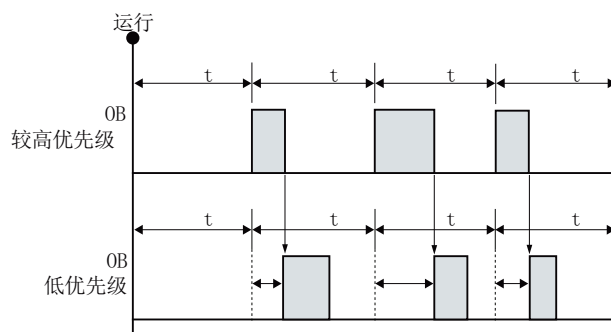
参数和类型		数据类型	说明
PHASE	IN	UDInt	相移（微秒）
RET_VAL	OUT	Int	执行条件代码

时间参数示例：

- 如果 CYCLE 时间 = 100 us，则由 OB\_NR 引用的中断 OB 将每隔 100 us 中断一次循环程序扫描。中断 OB 在执行后将执行控制交回程序扫描过程，从而继续从中断位置开始扫描。
- 如果 CYCLE 时间 = 0，则中断事件被禁用，并且不会执行中断 OB。
- PHASE（相移）时间是 CYCLE 时间间隔开始前的指定延迟时间。可使用相移来控制优先级较低的 OB 的执行时间。

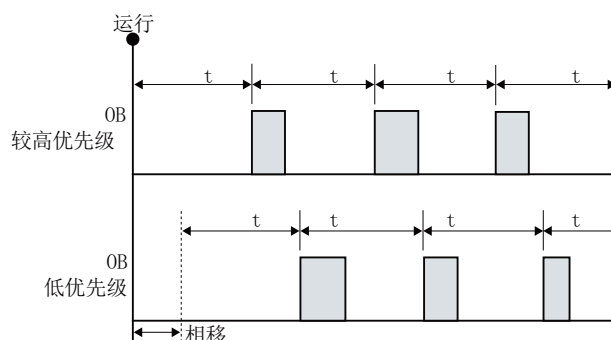
如果以相同的时间间隔调用优先级较高和优先级较低的 OB，则只有在优先级较高的 OB 完成处理后会调用优先级较低的 OB。低优先级 OB 的执行起始时间会根据优先级较高的 OB 的处理时间来延迟。

没有相移的 OB 调用



如果希望以固定的时间周期来执行优先级较低的 OB，则相移时间应大于优先级较高的 OB 的处理时间。

有相移的 OB 调用

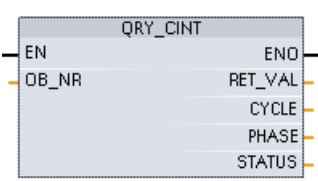


表格 8-90 条件代码

RET_VAL (W#16#...)	说明
0000	无错误
8090	OB 不存在或类型错误
8091	无效周期时间
8092	无效相移时间
80B2	OB 未附加事件

### 8.4.2.2 QRY\_CINT (Query 循环中断)

表格 8-91 QRY\_CINT (Query 循环中断)

LAD/FBD	SCL	说明
	<pre>ret_val := QRY_CINT(   ob_nr:=_int_in_,   cycle=&gt;_udint_out_,   phase=&gt;_udint_out_,   status=&gt;_word_out_);</pre>	<p>获取循环中断 OB 的参数和执行状态。返回的值早在执行 QRY_CINT 时便已存在。</p>

表格 8-92 参数的数据类型

参数和类型		数据类型	说明
OB_NR	IN	OB_CYCLIC	OB 号 (允许使用类似 OB_MyOBName 的符号名称)
RET_VAL	OUT	Int	执行条件代码
CYCLE	OUT	UDInt	时间间隔 (微秒)
PHASE	OUT	UDInt	相移 (微秒)
STATUS	OUT	Word	<p>循环中断状态代码:</p> <ul style="list-style-type: none"> <li>位 0 到 4, 请参见下面的 STATUS 表</li> <li>其它位, 始终为 0</li> </ul>

表格 8- 93 STATUS 参数

位	值	说明
0	0	CPU RUN 期间
	1	启动过程中
1	0	中断已启用。
	1	中断已通过 DIS_IRT 指令禁用。
2	0	中断未激活或已过期。
	1	中断已激活。
4	0	通过 OB_NR 标识的 OB 不存在。
	1	通过 OB_NR 标识的 OB 存在。
其它位		始终为 0

如果发生错误，RET\_VAL 显示相应的错误代码，并且参数 STATUS = 0。

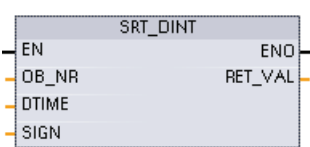
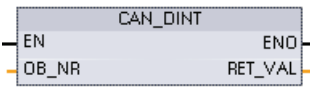

表格 8- 94 RET\_VAL 参数

RET_VAL (W#16#....)	说明
0000	无错误
8090	OB 不存在或类型错误。
80B2	OB 未附加事件。

### 8.4.3 延时中断

可使用 SRT\_DINT 和 CAN\_DINT 指令启动和取消延时中断处理过程，或使用 QRY\_DINT 指令查询中断状态。每个延时中断都是一个在指定的延迟时间过后发生的一次性事件。如果在延迟时间到期前取消延时事件，则不会发生程序中断。

表格 8-95 SRT\_DINT、CAN\_DINT 和 QRY\_DINT 指令

LAD/FBD	SCL	说明
	<pre>ret_val := SRT_DINT(   ob_nr:=_int_in_,   dtime:=_time_in_,   sign:=_word_in_);</pre>	SRT_DINT 启动延时中断，在参数 DTIME 指定的延迟过后执行 OB。
	<pre>ret_val := CAN_DINT(   ob_nr:=_int_in_);</pre>	CAN_DINT 取消已启动的延时中断。在这种情况下，将不执行延时中断 OB。
	<pre>ret_val := QRY_DINT(   ob_nr:=_int_in_,   status=&gt;_word_out_);</pre>	QRY_DINT 查询通过 OB_NR 参数指定的延时中断的状态。

表格 8-96 参数的数据类型

参数和类型		数据类型	说明
OB_NR	IN	OB_DELAY	将在延迟时间过后启动的组织块 (OB)：从使用“添加新块”(Add new block) 项目树功能创建的可用延时中断 OB 中进行选择。双击该参数域，然后单击助手图标可查看可用的 OB。
DTIME <sup>1</sup>	IN	Time	延迟时间值（1 到 60000 ms）
SIGN <sup>1</sup>	IN	Word	S7-1200 不使用：接受任何值。为避免发生错误，必须指定一个值。
RET_VAL	OUT	Int	执行条件代码
STATUS	OUT	Word	QRY_DINT 指令：所指定延时中断 OB 的状态，请参见下文表格

<sup>1</sup> 仅限 SRT\_DINT

**操作**

SRT\_DINT 指令指定延迟时间、启动内部延迟时间定时器以及将延时中断 OB 子程序与延时超时事件相关联。指定的延迟时间过去后，将生成可触发相关延时中断 OB 执行的程序中断。在指定的延时发生之前执行 CAN\_DINT 指令可取消进行中的延时中断。激活延时和循环中断事件的总次数不得超过四次。

**在项目中添加延时中断 OB 子程序**

只有延时中断 OB 可分配给 SRT\_DINT 和 CAN\_DINT 指令。新项目中不存在延时中断 OB。必须将延时中断 OB 添加到项目中。要创建延时中断 OB，请按以下步骤操作：

1. 在项目树的“程序块”(Program blocks) 分支中双击“添加新块”(Add new block)，选择“组织块 (OB)”(Organization block (OB))，然后选择“延时中断”(Time delay interrupt)。
2. 可以重命名 OB、选择编程语言或选择块编号。如果要分配与自动分配的编号不同的块编号，请切换到手动编号模式。
3. 编辑延时中断 OB 子程序，并创建要在发生延时超时事件时执行的已编程响应。可从延时中断 OB 调用其它最多嵌套四层深的 FC 和 FB 代码块。
4. 编辑 SRT\_DINT 和 CAN\_DINT 指令的 OB\_NR 参数时，将可以使用新分配的延时中断 OB 名称。

**QRY\_DINT 参数 STATUS**

表格 8-97 如果存在错误 (REL\_VAL <> 0)，则 STATUS = 0。

位	值	说明
0	0	在 RUN 模式下
	1	在启动过程中
1	0	中断已启用。
	1	中断已禁用。
2	0	中断未激活或已过期。
	1	中断已激活。
4	0	不存在具有 OB_NR 中所指定 OB 号的 OB。
	1	存在具有 OB_NR 中所指定 OB 号的 OB。
其它位		始终为 0



## 条件代码

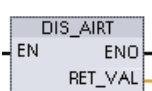
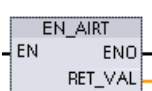
表格 8-98 SRT\_DINT、CAN\_DINT 和 QRY\_DINT 的条件代码

RET_VAL (W#16#...)	说明
0000	未出错
8090	不正确的参数 OB_NR
8091	不正确的参数 DTIME
80A0	未启动延时中断。

## 8.4.4 异步事件中断

使用 DIS\_AIRT 和 EN\_AIRT 指令可禁用和启用报警中断处理过程。

表格 8-99 DIS\_AIRT 和 EN\_AIRT 指令

LAD/FBD	SCL	说明
	DIS_AIRT();	DIS_AIRT 可延迟新中断事件的处理。可在 OB 中多次执行 DIS_AIRT。
	EN_AIRT();	对先前使用 DIS_AIRT 指令禁用的中断事件处理，可使用 EN_AIRT 来启用。每一次 DIS_AIRT 执行都必须通过一次 EN_AIRT 执行来取消。必须在同一个 OB 中或从同一个 OB 调用的任意 FC 或 FB 中完成 EN_AIRT 执行后，才能再次启用此 OB 的中断。

 警告

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。

## 8.5 诊断 (PROFINET 或 PROFIBUS)

表格 8- 100 参数的数据类型

参数和类型		数据类型	说明
RET_VAL	OUT	Int	延迟次数 = 队列中的 DIS_AIRT 执行次数。

由操作系统会统计 DIS\_AIRT 执行的次数。在特别通过 EN\_AIRT 指令再次取消之前或者在已完成处理当前 OB 之前，这些执行中的每一个都保持有效。例如：如果通过五次 DIS\_AIRT 执行禁用中断五次，则在再次启用中断前，必须通过五次 EN\_AIRT 执行来取消禁用。

再次启用中断事件后，将处理 DIS\_AIRT 生效期间发生的中断；或者在完成执行当前 OB 后，立即处理中断。

参数 RET\_VAL 表示禁用中断处理的次数，即已排队的 DIS\_AIRT 执行的个数。只有当参数 RET\_VAL = 0 时，才会再次启用中断处理。

## 8.5 诊断 (PROFINET 或 PROFIBUS)

### 8.5.1 诊断指令

以下诊断指令适用于 PROFINET 或 PROFIBUS:

- GET\_DIAG 指令 (页 324): 可读取指定设备的诊断信息。
- DeviceStates 指令 (页 321): 可获取 I/O 子系统中分布式 I/O 设备的运行状态。
- ModuleStates 指令 (页 322): 可获取分布式 I/O 设备中各模块的运行状态。
- LED 指令 (页 319): 可读取 I/O 设备的 LED 状态。

### 8.5.2 分布式 I/O 的诊断事件

#### 说明

对于 PROFIBUS IO 系统，除非硬件兼容性设置为允许可接受替换模块 (页 125)，并且有一个或多个模块丢失或者不是已组态模块的可接受替换模块，否则 CPU 在下载或循环上电后将转到 RUN 模式。

如下表所示，CPU 支持对分布式 I/O 系统中的组件组态的诊断。只要发生下面提到的错误，诊断缓冲区就会生成一个日志条目。

表格 8- 101 PROFINET 和 PROFIBUS 诊断事件的处理

错误类型	成为站的诊断信息?	在诊断缓冲区中生成条目?	CPU 的操作模式
诊断错误	是	是	保持 RUN 模式
机架或站故障	是	是	保持 RUN 模式
I/O 访问错误 <sup>1</sup>	否	是	保持 RUN 模式
外围设备访问错误 <sup>2</sup>	否	是	保持 RUN 模式
插/拔事件	是	是	保持 RUN 模式

1 I/O 访问错误示例原因：已移除的一个模块。

2 外围设备访问错误示例原因：非周期性地与没有通信的子模块进行通信。

可对每个站使用 GET\_DIAG 指令 (页 324) 来获取相应的诊断信息。用户借此可通过编程来处理设备错误，并根据需要将 CPU 切换为 STOP 模式。采用此方法时，您需要指定从哪个硬件设备读取状态信息。

GET\_DIAG 指令使用站的“L 地址”(LADDR) 来获取整个站点的健康状况。此 L 地址可在“网络组态”(Network Configuration) 视图找到，或者也可选择整个站机架（整个灰色区域）并在站的“属性”(Properties) 选项卡中获取。对于各模块的 LADDR，既可在该模块的属性中查看（在设备组态中），也可在 CPU 的默认变量表中查看。

### 8.5.3 LED 指令

表格 8- 102 LED 指令

LAD/FBD	SCL	说明
	<pre>ret_val := LED(     laddr:=_word_in_,     LED:=_uint_in_);</pre>	使用 LED 指令可读取某 CPU 或接口上 LED 的状态。通过 RET_VAL 输出返回指定 LED 的状态。

表格 8- 103 参数的数据类型

参数和类型		数据类型	说明	
LADDR	IN	HW_IO	CPU 或接口的标识号 <sup>1</sup>	
LED	IN	UInt	LED 标识号	
			1	RUN/STOP

8.5 诊断 (PROFINET 或 PROFIBUS)

参数和类型		数据类型	说明		
			2	出错	颜色 1 = 红色
			3	维护	颜色 1 = 黄色
			4	冗余	不适用
			5	链接	颜色 1 = 绿色
			6	Tx/Rx	颜色 1 = 黄色
RET_VAL	OUT	Int	LED 的状态		

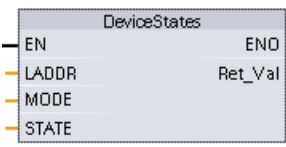
<sup>1</sup> 例如，可从该参数的下拉列表中选择 CPU（如“PLC\_1”）或 PROFINET 接口。

表格 8- 104 RET\_VAL 的状态

RET_VAL (W#16#...)	说明	
0 到 9 LED 状态	0	LED 不存在
	1	灭
	2	颜色 1 常亮
	3	颜色 2 常亮
	4	颜色 1 以 2 Hz 的频率闪烁
	5	颜色 2 以 2 Hz 的频率闪烁
	6	颜色 1 和 2 以 2 Hz 的频率交替闪烁
	7	颜色 1 点亮 (Tx/Rx)
	8	颜色 2 点亮 (Tx/Rx)
	9	LED 状态不可用
8091	由 LADDR 标识的设备不存在	
8092	由 LADDR 标识的设备不支持 LED	
8093	LED 标识符未定义	
80Bx	由 LADDR 标识的 CPU 不支持 LED 指令	

### 8.5.4 DeviceStates 指令

表格 8- 105 DeviceStates 指令

LAD/FBD	SCL	说明
	<pre>ret_val := DeviceStates(     laddr:=hw_io_in_,     mode:=_uint_in_,     state:=_variant_inout_);</pre>	<p>DeviceStates 获取 I/O 子系统的 I/O 设备运行状态。指令执行后，STATE 参数将以位列表形式包含各个 I/O 设备的错误状态（针对指定的 LADDR 和 MODE）。此信息与在 STEP 7 诊断视图中看到的设备状态一致。</p>

表格 8- 106 参数的数据类型

参数和类型		数据类型	说明
LADDR	IN	HW_IOSYSTEM	逻辑地址：（I/O 系统的标识符）
MODE	IN	UInt	状态类型： <ul style="list-style-type: none"> <li>• 1: 设备的组态处于激活状态或尚未完成。</li> <li>• 2: 设备故障</li> <li>• 3: 被禁止的设备</li> <li>• 4: 设备存在</li> </ul>
RET_VAL	OUT	Int	执行条件代码
STATE <sup>1</sup>	InOut	Variant	接收每个设备的错误状态的缓冲区：为 STATE 参数选择的数据类型可以是任何位类型（Bool、Byte、Word 或 DWord），也可以是位类型的数组 <ul style="list-style-type: none"> <li>• 摘要位：位 0 = 1，前提时 I/O 设备的状态位中有一个为 1</li> <li>• 状态位：根据所选 MODE，站编号为 n 的 I/O 设备的状态。例如，MODE = 2 且位 3 = 1 表示站 3 有故障。</li> </ul>

<sup>1</sup> 对于 PROFIBUS-DP，状态信息的长度为 128 位。对于 PROFINET I/O，长度为 1024 位。

指令执行后，STATE 参数将以位列表形式包含各个 I/O 设备的错误状态（针对分配的 LADDR 和 MODE）。

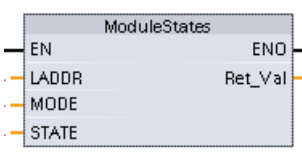
8.5 诊断 (PROFINET 或 PROFIBUS)

表格 8- 107 条件代码

RET_VAL (W#16#...)	说明
0	无错误
8091	LADDR 不存在。
8092	LADDR 未寻址 I/O 系统。
8093	为 STATE 参数指定的数据类型无效： 有效数据类型为 (Bool、Byte、Word 或 Dword) 或者 (Bool、Byte、Word 或 Dword) 的数组
80Bx	CPU 不支持对此 LADDR 使用 DeviceStates 指令。
8452	完整的状态数据对于分配的 STATE 参数来说过大。 STATE 缓冲区包含部分结果。

8.5.5 ModuleStates 指令

表格 8- 108 ModuleStates 指令

LAD/FBD	SCL	说明
	<pre>ret_val := ModuleStates(     laddr:=_word_in_,     mode:=_uint_in_,     state:=_variant_inout);</pre>	<p>ModuleStates 获取 I/O 模块的运行状态。指令执行后，STATE 参数将以位列表形式包含各个 I/O 模块的错误状态（针对指定的 LADDR 和 MODE）。此信息与在 STEP 7 诊断视图中看到的模块状态一致。</p>

表格 8- 109 参数的数据类型

参数和类型		数据类型	说明
LADDR	IN	HW_DEVICE	逻辑地址 (I/O 模块的标识符)
MODE	IN	UInt	状态类型： <ul style="list-style-type: none"> <li>• 1: 模块的组态处于激活状态或尚未完成。</li> <li>• 2: 模块有故障</li> <li>• 3: 模块已禁用</li> <li>• 4: 模块存在</li> </ul>

参数和类型		数据类型	说明
RET_VAL	OUT	Int	状态 (条件代码)
STATE <sup>1</sup>	InOut	Variant	<p>接收每个模块的错误状态的缓冲区：用于 STATE 参数的数据类型可以是任何位类型 (Bool、Byte、Word 或 DWord)，也可以是位类型的数组。</p> <ul style="list-style-type: none"> <li>• 摘要位：位 0 = 1，前提是 I/O 模块的状态位中有一个为 1</li> <li>• 状态位：根据所选 MODE，插槽编号为 <i>n</i> 的 I/O 模块的状态。例如，MODE = 2 且位 3 = 1 表示站 3 有故障。</li> </ul>

<sup>1</sup> 最多可分配 128 位。所需位数取决于 I/O 模块的使用情况。

表格 8- 110 条件代码


RET_VAL ( W#16#...)	说明
0	无错误
8091	由 LADDR 标识的模块不存在。
8092	由 LADDR 标识的模块未寻址 I/O 设备。
8093	STATE 参数的数据类型无效：有效数据类型为 (Bool、Byte、Word 或 Dword) 或者 (Bool、Byte、Word 或 Dword) 的数组。
80Bx	该 CPU 不支持对此 LADDR 使用 ModuleStates 指令。
8452	完整的状态数据对于分配的 STATE 参数来说过大。STATE 缓冲区包含部分结果。

### 8.5.6 GET\_DIAG 指令

#### 说明

可以使用“GET\_DIAG”指令读出硬件对象的诊断信息。硬件对象通过 LADDR 参数进行选择。使用 MODE 参数选择要读出的诊断信息。

表格 8- 111 GET\_DIAG 指令

LAD/FBD	SCL	说明
	<pre>ret_val := GET_DIAG(     mode:=_uint_in_,     laddr:=_word_in_,     cnt_diag=&gt;_uint_out_,     diag:=_variant_inout_,     detail:=_variant_inout_);</pre>	<p>从指定的硬件设备读取诊断信息。</p>

#### 参数

下表列出了“GET\_DIAG”指令的参数：

表格 8- 112 参数的数据类型

参数和类型		数据类型	说明
MODE	IN	UInt	使用 MODE 参数选择要输出的诊断数据。
LADDR	IN	HW_ANY (Word)	设备的硬件 ID
RET_VAL	OUT	Int	指令的状态
CNT_DIAG	OUT	UInt	输出诊断详细信息数量
DIAG	InOut	Variant	指向用于存储所选模式的诊断信息的数据区。
DETAILS	InOut	Variant	指向用于存储与所选模式一致的诊断详细信息的数据区。



## MODE 参数

根据 MODE 参数的值，在 DIAG、CNT\_DIAG 和 DETAILS 输出参数中输出不同的诊断数据。

表格 8- 113 MODE 参数

MODE	说明	DIAG	CNT_DIAG	DETAILS
0	以 DWord 格式输出模块的所有支持诊断信息，其中位 X=1 表示支持模式 X。	所支持模式的位字符串 (DWord 格式)，其中位 X=1 表示支持模式 X。	0	-
1	输出已寻址硬件对象的固有状态。	诊断状态：与 DIS 结构一致的输出。（注意：请参见下面的“DIS 结构”信息以及本部分结尾处的 GET_DIAG 指令示例。）	0	-
2	输出已寻址硬件对象所有从属模块的状态。	输出与 DNN 结构一致的诊断数据。（注意：请参见下面的“DNN 结构”信息以及本部分结尾处的 GET_DIAG 指令示例。）	0	与 DiagnosticsDetails 结构一致的模块状态信息。

## DIS 结构

在 MODE 参数 =1 的情况下，诊断信息的输出与 DIS 结构一致。下表列出了各个参数值的含义：

表格 8- 114 诊断信息源 (DIS) 的结构

参数	数据类型	值	说明
MaintenanceState	DWord	枚举	
		0	不需要维护
		1	模块或设备已禁用。
		2	-
		3	-

8.5 诊断 (PROFINET 或 PROFIBUS)

参数	数据类型	值	说明
		4	-
		5	需要维护
		6	要求维护
		7	错误
		8	附属模块中的状态未知/错误
		9	-
		10	输入/输出不可用。
Componentstate Detail	DWord	位数组	模块子模块的状态： <ul style="list-style-type: none"> <li>位 0 到位 15： 模块的状态消息</li> <li>位 16 到位 31： CPU 的状态消息</li> </ul>
		0 到 2 (枚举)	附加信息： <ul style="list-style-type: none"> <li>位 0： 无附加信息</li> <li>位 1： 不允许传送</li> </ul>
		3	位 3 = 1： 至少一个通道支持诊断限定符。
		4	位 4 = 1： 至少一个通道或一个组件需要维护
		5	位 5 = 1： 至少一个通道或一个组件要求维护
		6	位 6 = 1： 至少一个通道或一个组件有错误
		7 到 10	-
		11 到 14	位 11 = 1： PNIO - 子模块正确 位 12 = 1： PNIO - 更换模块 位 13 = 1： PNIO - 错误模块 位 14 = 1： PNIO - 模块已断开
		15	-

参数	数据类型	值	说明
		16 到 31	CPU 生成的模块状态信息： 位 16 = 1： 模块已禁用 位 17 = 1： CiR 操作激活 位 18 = 1： 输入不可用 位 19 = 1： 输出不可用 位 20 = 1： 溢出诊断缓冲区 位 21 = 1： 诊断不可用 位 22 - 31： 保留（始终为 0）
OwnState	Uint16	枚举	OwnState 参数的值描述了模块的维护状态。
		0	无故障
		1	模块或设备已禁用。
		2	需要维护
		3	要求维护
		4	错误
		5	无法从 CPU 访问模块或设备（对于 CPU 下的模块和设备有效）。
		6	输入/输出不可用。
7	-		
IO State	Uint16	位数组	模块的 I/O 状态
		0	位 0 = 1： 不需要维护
		1	位 1 = 1： 模块或设备已禁用。
		2	位 2 = 1： 需要维护
		3	位 3 = 1： 要求维护
		4	位 4 = 1： 错误
		5	位 5 = 1： 无法从 CPU 访问模块或设备（对于 CPU 下的模块和设备有效）。
		6	输入/输出不可用。
		7	限定符；如果位 0、2 或 3 置位，则位 7 = 1
		8 到 15	保留（始终为 0）

8.5 诊断 (PROFINET 或 PROFIBUS)

参数	数据类型	值	说明
OperatingState	UInt16	枚举	
		0	-
		1	处于 STOP 状态/固件更新
		2	处于 STOP 状态/复位存储器
		3	处于 STOP 状态/自启动
		4	处于 STOP 状态
		5	存储器复位
		6	处于 START 状态
		7	处于 RUN 状态
		8	-
		9	处于 HOLD 状态
		10	-
		11	-
		12	模块有故障
		13	-
		14	无电源
		15	CiR
		16	处于 STOP 状态/无 DIS
		17	输入
		18	
19			
20			

**DiagnosticsDetail 结构**

在 MODE 参数=2 的情况下，诊断详细信息的输出与 DiagnosticsDetail 结构一致。下表列出了各个参数值的含义：

表格 8- 115 诊断详细信息的结构

参数	数据类型	说明
ChannelNumber	UInt	通道号
Properties	Word	
ALID	UInt	报警的标识 ID
Qualifier	DWord	诊断数据限定符
ErrorType	UDInt	通道错误类型
ExtErrorType	UDInt	扩展通道错误类型
AddValue_1	UInt	附加值
AddValue_2	UInt	附加值
AddValue_3	UInt	附加值
AddValue_4	UInt	附加值

**DNN 结构**

在 MODE 参数=2 的情况下，诊断详细信息的输出与 DNN 结构一致。下表列出了各个参数值的含义：

表格 8- 116 诊断导航节点 (DNN) 的结构

参数	数据类型	值	说明
SubordinateState	UINT	Enum	从属模块的状态（请参见 DIS 结构的参数 OwnState）
SubordinateIOState	WORD	Bitarray	从属模块的输入和输出状态（请参见 DIS 结构的参数 IO State）
DNNmode	WORD	Bitarray	<ul style="list-style-type: none"> <li>• 位 0 = 0: 诊断已启用</li> <li>• 位 0 = 1: 诊断已禁用</li> <li>• 位 1 到 15: 保留</li> </ul>

## RET\_VAL 参数

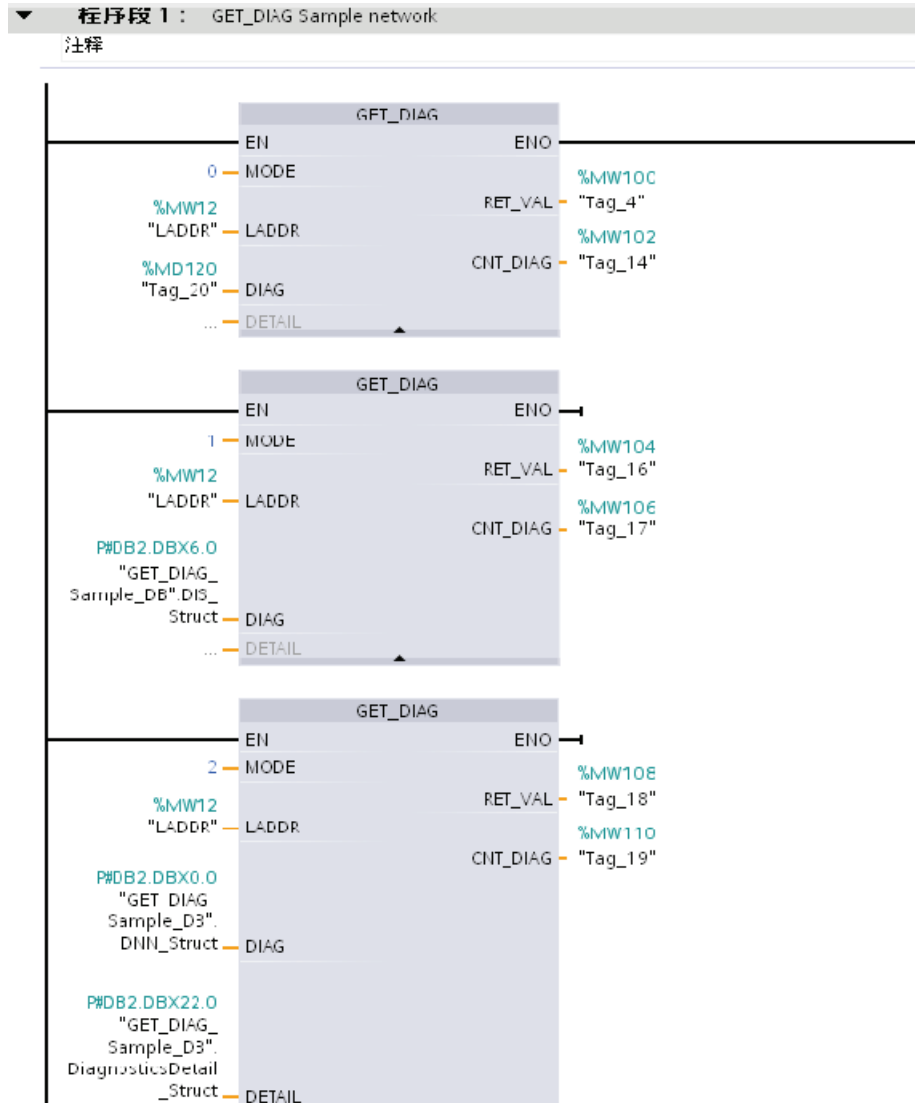
表格 8- 117 RET\_VAL 参数的错误代码

错误代码 (W#16#...)	说明
0	无错误
n	DETAILS 参数中的数据区过小。无法输出诊断数据的所有详细信息。
8080	不支持 MODE 参数中的值。
8081	所选模式 (参数 MODE) 不支持 DIAG 参数中的类型。
8082	所选模式 (参数 MODE) 不支持 DETAILS 参数中的类型。
8090	LADDR 不存在。
8091	CHANNEL 参数中的所选通道不存在。
80C1	并行执行的资源不足

示例

下面的梯形逻辑程序段和 DB 显示了如何使用三种结构的三种模式：

- DIS
- DiagnosticsDetail
- DNN



8.5 诊断 (PROFINET 或 PROFIBUS)

名称	数据类型	偏...	启动值	保持	在 HMI 中...	注释
Static						
DNN_Struct	DNN	0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
SubordinateState	UInt	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
SubordinateIOState	Word	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
DNNmode	Word	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
DIS_Struct	DIS	6.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
MaintenanceState	DWord	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
ComponentStateDetail	DWord	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
OwnState	UInt	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
IOState	Word	10.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
OperatingState	UInt	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
DiagnosticsDetail_Struct	DiagnosticsDetail	22.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
ChannellNumber	UInt	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Channel number
Properties	Word	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
ALID	UInt	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Alarm identifier
Qualifier	DWord	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
ErrorType	UDInt	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Channel error type
ExtErrorType	UDInt	16.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	extended channel error type
AddValue_1	UInt	20.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value
AddValue_2	UInt	22.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value
AddValue_3	UInt	24.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value
AddValue_4	UInt	26.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	additional value

- ① DNN
- ② DIS
- ③ DiagnosticsDetail

说明

在 DB 中，必须手动键入数据类型，以访问三种结构中的每一种；没有下拉列表选择。如下所示，准确键入数据类型：

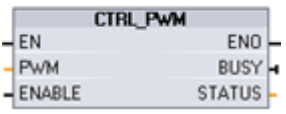
- DNN
- DIS
- DiagnosticsDetail



## 8.6 脉冲

### 8.6.1 CTRL\_PWM 指令

表格 8- 118 CTRL\_PWM（脉宽调制）指令

LAD/FBD	SCL	说明
	<pre>"CTRL_PWM_DB" (   PWM:=_word_in_,   enable:=_bool_in_,   busy=&gt;_bool_out_,   status=&gt;_word_out_);</pre>	<p>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）启动之后将连续运行。脉冲宽度会根据需要进行变化以影响所需的控制。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“CTRL\_PWM\_DB”是背景 DB 的名称。

### CTRL\_HSC

表格 8- 119 参数的数据类型

参数和类型		数据类型	说明
PWM	IN	HW_PWM (Word)	PWM 标识符：已启用的脉冲发生器的名称将变为“常量”(constant) 变量表中的变量，并可用作 PWM 参数。（默认值：0）
ENABLE	IN	Bool	1 = 启动脉冲发生器 0 = 停止脉冲发生器
BUSY	OUT	Bool	功能忙（默认值：0）
STATUS	OUT	Word	执行条件代码（默认值：0）

CTRL\_PWM 指令将参数信息存储在 DB 中。数据块参数不是由用户单独更改的，而是由 CTRL\_PWM 指令进行控制。

通过将其变量名称用于 PWM 参数，指定要使用的已启用脉冲发生器。

EN 输入为 TRUE 时，PWM\_CTRL 指令根据 ENABLE 输入的值启动或停止所标识的 PWM。脉冲宽度由相关 Q 字输出地址中的值指定。

由于 CPU 在 CTRL\_PWM 指令执行后处理请求，所以参数 BUSY 总是报告 FALSE。如果检测到错误，则 ENO 设置为 FALSE 且参数 STATUS 包含条件代码。

CPU 第一次进入 RUN 模式时，脉冲宽度将设置为在设备组态中组态的初始值。根据需  
要将值写入设备组态中指定的 Q 字位置（“输出地址”/“起始地址：”），以更改脉冲宽  
度。使用指令（如移动、转换、数学）或 PID 功能框将所需脉冲宽度写入相应的 Q 字。  
必须使用 Q 字值的有效范围（百分数、千分数、万分数或 S7 模拟格式）。

#### 说明

##### 无法强制分配给 PWM 和 PTO 的数字量 I/O 点

在设备组态期间分配脉冲宽度调制 (PWM, Pulse-Width Modulation) 和脉冲串输出 (PTO, Pulse-Train Output) 设备使用的数字量 I/O 点。将数字 I/O 点分配给这些设备之后，无法通过监视表格强制功能修改所分配的 I/O 点的地址值。

表格 8- 120 STATUS 参数的值

STATUS	说明
0	无错误
80A1	PWM 标识符未寻址到有效的 PWM。

表格 8- 121 公共条件代码

条件代码 <sup>1</sup>	说明
8022	存储区对于输入太小
8023	存储区对于输出太小
8024	输入区非法
8025	输出区非法
8028	输入位赋值非法
8029	输出位赋值非法
8030	输出区是只读 DB。
803A	DB 不存在。

<sup>1</sup> 如果执行代码块时发生某个常见错误，则 CPU 将进入 STOP 模式，除非在该代码块中使用 GetError 或 GetErrorID 指令编写程序来响应错误。

## 8.6.2 脉冲输出的作用



- ① 循环时间  
② 脉冲宽度

脉冲宽度可表示为循环时间的百分数（0 到 100）、千分数（0 到 1000）、万分数（0 到 10000）或 S7 模拟格式。

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。

由于 PWM 输出可从 0 到满刻度变化，因此可提供在许多方面都与模拟输出相同的数字输出。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。

有两种脉冲发生器可用于控制高速脉冲输出功能：PWM 和脉冲串输出 (PTO, Pulse train output)。PTO 由运动控制指令使用。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。

这两种脉冲发生器映射到特定的数字输出，如下表所示。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。无论是在 CPU 上还是在连接的信号板上，PTO1/PWM1 都使用前两个数字输出，PTO2/PWM2 使用接下来的两个数字输出。请注意，PWM 仅需要一个输出，而 PTO 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。

### 注意

#### 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），这将从 Q 存储器中移除相应的输出地址（Q0.0、Q0.1、Q4.0 和 Q4.1），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。

表格 8- 122 脉冲发生器的默认输出分配

说明	脉冲	方向
PTO 0		
内置 I/O	Q0.0	Q0.1
SB I/O	Q4.0	Q4.1
PWM 0		

说明	脉冲	方向
内置输出	Q0.0	-
SB 输出	Q4.0	-
PTO 1		
内置 I/O	Q0.2	Q0.3
SB I/O	Q4.2	Q4.3
PWM 1		
内置输出	Q0.2	-
SB 输出	Q4.2	-
PTO 2		
内置 I/O	Q0.4 <sup>1</sup>	Q0.5 <sup>1</sup>
SB I/O	Q4.0	Q4.1
PWM 2		
内置输出	Q0.4 <sup>1</sup>	-
SB 输出	Q4.1	-
PTO 3		
内置 I/O	Q0.6 <sup>2</sup>	Q0.7 <sup>2</sup>
SB I/O	Q4.2	Q4.3
PWM 3		
内置输出	Q0.6 <sup>2</sup>	-
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 以及 CPU 1215C PTO/PWM 功能。

### 8.6.3 组态 PWM 的脉冲通道

要准备 PWM 操作，首先通过选择 CPU 来组态设备配置中的脉冲通道，然后组态脉冲发生器 (PTO/PWM)，并选择 PWM1 或 PWM2。启用脉冲发生器（复选框）。如果启用一

个脉冲发生器，将为该特定脉冲发生器分配一个唯一的默认名称。可以通过在“名称：”(Name:)编辑框中编辑此名称来对其进行更改。已启用的脉冲发生器的名称将成为“常量”(constant) 变量表中的变量，并可用作 CTRL\_PWM 指令的 PWM 参数。

#### 注意

脉冲输出发生器的最大脉冲频率对于 CPU 的数字量输出为 100 KHz，而对于 SB 的数字量输出为 20 KHz，对于高速 SB 的数字量输出为 200 KHz。可是，当组态了最大速度或频率超出此硬件限制的轴时，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。

可按如下方式重命名脉冲发生器、添加注释以及分配参数：

- 脉冲发生器可用作： PWM 或 PTO（选择 PWM）
- 输出源： 板载 CPU 或 SB
- 时间基数： 毫秒或微秒
- 脉冲宽度格式：
  - 百分数（0 到 100）
  - 千分数（0 到 1000）
  - 万分数（0 到 10000）
  - S7 模拟格式（0 到 27648）
- 循环时间： 输入循环时间值。该值只能在“设备配置”(Device configuration) 中更改。
- 初始脉冲宽度： 输入初始脉冲宽度值。可在运行期间更改脉冲宽度值。

输入起始地址以组态输出地址。输入要在其中查找脉冲宽度值的 Q 字地址。

#### 注意

##### 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），这将从 Q 存储器中移除相应的输出地址（Q0.0、Q0.1、Q4.0 和 Q4.1），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。

对于 PWM1，默认位置是 QW1000；而对于 PWM2，默认位置是 QW1002。该位置的值控制脉冲宽度，并且在每次 CPU 从 STOP 切换到 RUN 模式时都会初始化为上面指定的“初始脉冲宽度：”(Initial pulse width:) 值。在运行期间更改该 Q 字值会引起脉冲宽度变化。

## 8.7 数据记录

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。

使用内置的 **PLC Web** 服务器管理数据日志文件。通过“数据日志”(Data Logs) 标准 **Web** 页面下载最新记录、所有记录、清除记录或删除日志文件。将数据日志文件传送到 **PC** 后，可使用标准电子表格工具（如 **Excel**）分析数据。

### 8.7.1 数据日志记录结构

**DataLogCreate** 指令的 **DATA** 和 **HEADER** 参数分配日志记录的所有数据元素的数据类型和列标题说明。

#### **DataLogCreate** 指令的 **DATA** 参数

**DATA** 参数指向用作新日志记录临时缓冲区的存储器，必须将其分配给 **M** 或 **DB** 位置。

可以分配整个 **DB**（源自创建 **DB** 时分配的 **PLC** 数据类型），也可分配部分 **DB**（指定的 **DB** 元素可以是任何数据类型、数据类型结构、**PLC** 数据类型或数据数组）。

结构数据类型限制为单嵌套级。所声明数据元素的总数应与标题参数中指定的列数相对应。可以分配的最大数据元素个数为 **253**（带时间戳）或 **255**（不带时间戳）。这一限制使记录始终处于 **Excel** 工作表的 **256** 列限制范围内。

**DATA** 参数可在“标准”（与 **S7-300/400** 兼容）或“优化”**DB** 类型中分配保持性数据元素或非保持性数据元素。

要写入数据日志记录，首先必须用新过程值装载临时 **DATA** 记录，然后执行 **DataLogWrite** 指令将新记录值保存到 **Datalog** 文件中。

#### **DataLogCreate** 指令的 **HEADER** 参数

**HEADER** 参数指向 **CSV** 文件中编码的数据矩阵的第一行的列标题名称。**HEADER** 数据必须位于 **DB** 或 **M** 存储器，且字符必须遵守标准 **CSV** 格式规则，各列名称用逗号分隔。数据类型可以是字符串、字节数组或字符数组。字符/字节数组的大小可以增加，其中字

字符串被限制为最多 255 个字节。HEADER 参数是可选参数。如果未分配 HEADER 参数，则不会在数据日志文件中创建标题行。

## 8.7.2 控制数据日志的程序指令

### 8.7.2.1 DataLogCreate

表格 8- 123 DataLogCreate 指令

LAD/FBD	SCL	说明
	<pre>"DataLogCreate_DB" (   req:=_bool_in_,   records:=_udint_in_,   format:=_uint_in_,   timestamp:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   name:=_string_inout_,   ID:=_dword_inout_,   header:=_variant_inout_,   data:=_variant_inout_);</pre>	<p>创建和初始化数据日志文件。文件创建到 PLC \DataLogs 目录下，由 NAME 参数命名，并且已隐式打开以便执行写操作。可以使用 Data log 指令通过程序将运行过程数据存储到 CPU 的闪存中。</p> <p>STEP 7 会在插入指令时自动创建关联的背景数据块。</p>

1 在 SCL 示例中，“DataLogCreate\_DB”是背景数据块的名称。

表格 8- 124 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	通过由低到高的（上升沿）信号启动操作。（默认值： False）
RECORDS	IN	UDint	覆盖最旧条目前循环数据日志可存储的最大数据记录数：不包括标题记录。PLC 装载存储器的可用空间必须足够大，以确保成功创建数据日志。（默认值 - 1）
FORMAT	IN	UInt	数据日志格式： <ul style="list-style-type: none"> <li>• 0 - 内部格式（不支持）</li> <li>• 1 - 逗号分隔值“csv-eng”（默认值）</li> </ul>

8.7 数据记录

参数和类型		数据类型	说明
TIMESTAMP	IN	UInt	<p>数据时间戳格式： 不需要日期和时间字段的列标题。 时间戳使用系统时间（世界协调时间 - UTC）， 不使用本地时间。</p> <ul style="list-style-type: none"> <li>• 0 - 无时间戳</li> <li>• 1 - 日期和时间戳（默认值）</li> </ul>
NAME	IN	Variant	<p>数据日志名称： 用户可在此提供名称。 此变量仅支持 <b>String</b> 数据类型， 且只可位于本地、DB 或 M 存储器。（默认值： ''）</p> <p>该字符串引用还用作数据日志文件名。 名称中的字符必须遵守 <b>Windows</b> 文件系统命名限制。 字符 \ / : * ? " &lt; &gt;   及空格字符均不许使用。</p>
ID	输入/输出	DWord	<p>数据日志数字标识符： 存储该生成值以便与其它数据日志指令配合使用。 ID 参数仅用作 <b>DataLogCreate</b> 指令的输出。（默认值： 0）</p> <p>该参数不支持符号名称访问。</p>
HEADER	输入/输出	Variant	<p>指向 <b>CSV</b> 文件中编码的数据矩阵的第一行的数据日志列标题名称。（默认值： 空值）。</p> <p><b>HEADER</b> 数据必须位于 <b>DB</b> 或 <b>M</b> 存储器。</p> <p>字符必须遵守标准 <b>CSV</b> 格式规则， 各列名称用逗号分隔。 数据类型可以是字符串、字节数组或字符数组。 字符/字节数组的大小可以增加， 其中字符串被限制为最多 <b>255</b> 个字节。</p> <p><b>HEADER</b> 参数是可选参数。 如果未设置 <b>HEADER</b> 参数， 则不会在数据日志文件中创建标题行。</p>
DATA	输入/输出	Variant	<p>指向记录数据结构、用户自定义类型 (UDT) 或数组。 记录数据必须位于 <b>DB</b> 或 <b>M</b> 存储器。</p> <p><b>DATA</b> 参数指定数据日志记录的各个数据元素（列）及其数据类型。 结构数据类型限制为单嵌套级。 所声明数据元素的个数应与标题参数中指定的列数相对应。 可以分配的最大数据元素个数为 <b>253</b>（带时间戳）或 <b>255</b>（不带时间戳）。 这一限制使记录始终处于 <b>Excel</b> 工作表的 <b>256</b> 列限制范围内。</p>
DONE	OUT	Bool	<p>上一请求已完成且没有出错后， <b>DONE</b> 位将保持为 <b>TRUE</b> 一个扫描周期时间。（默认值： <b>False</b>）</p>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 - 无操作正在进行</li> <li>• 1 - 操作正在进行</li> </ul>



参数和类型		数据类型	说明
ERROR	OUT	Bool	上一请求因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个扫描周期内有效。
STATUS	OUT	Word	执行条件代码（默认值：0）

采用基于参数 RECORDS 和 DATA 预定的固定大小创建数据日志文件。数据记录被组织成循环日志文件。新记录将附加到数据日志文件中，直到已存储 RECORDS 参数指定的最大记录数为止。写入的下一条记录将覆盖最早的记录。另一记录写入操作将覆盖下一条最早的数据记录，依此类推。

#### 说明

如果要防止覆盖任何数据记录，则可在当前数据日志已存储最大记录数后，使用 DataLogNewFile 指令基于当前数据日志创建新数据日志。新数据记录将存储到新数据日志文件中。旧数据日志文件及记录数据仍保存在闪存中。

存储器资源使用情况：

- 数据日志仅占用装载存储器。
- 对数据日志的总数并没有设置限制。组合起来的所有数据日志的大小受装载存储器可用资源的限制。一次只能打开八个数据日志。
- RECORDS 参数的最大可能数值是 UDint 数的限值 (4,294,967,295)。RECORD 参数的实际限值取决于单个记录的大小、其它数据日志的大小及装载存储器的可用资源。此外，Excel 对 Excel 工作表中允许的行数也有一定限制。

#### 说明

DataLogCreate 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。程序逻辑监视并捕捉到 DataLogCreate DONE 位转换到 TRUE 状态后，新的数据日志才可用于其它数据日志操作。

表格 8- 125 ERROR 和 STATUS 的值

ERROR	STATUS (W#16#....)	说明
0	0000	无错误
0	7000	无 REQ 沿时调用：BUSY = 0, DONE = 0
0	7001	有 REQ 沿时首次调用（工作）：BUSY = 1, DONE = 0
0	7002	第 N 次调用（工作）：BUSY = 1, DONE = 0
1	8070	所有内部实例存储器都在使用。

8.7 数据记录

ERROR	STATUS (W#16#....)	说明
1	807F	内部错误
1	8090	文件名无效
1	8091	名称参数不是字符串引用。
1	8093	数据日志已存在。
1	8097	请求的文件长度超出文件系统最大值。
1	80B3	可用装载存储器空间不足。
1	80B4	MC（存储卡）受写保护。
1	80C1	打开的文件过多：最多只允许同时打开 8 个数据日志文件。
1	8253	记录计数无效
1	8353	格式选择无效
1	8453	时间戳选择无效
1	8B24	HEADER 区域分配无效：例如，指向本地存储器
1	8B51	HEADER 参数数据类型无效
1	8B52	HEADER 参数数据元素过多
1	8C24	DATA 区域分配无效：例如，指向本地存储器
1	8C51	DATA 参数数据类型无效
1	8C52	DATA 参数数据元素过多

8.7.2.2 DataLogOpen

表格 8- 126 DataLogOpen 指令

LAD/FBD	SCL	说明
	<pre>"DataLogOpen_DB" (   req:=_bool_in_,   mode:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   name:=_string_inout_,   ID:=_dword_inout_);</pre>	<p>打开已有数据日志文件。打开数据日志后，才可向该日志写入新记录。可单独打开和关闭各个数据日志。最多可同时打开八个数据日志。</p> <p>STEP 7 会在插入指令时自动创建关联的背景数据块。</p>

<sup>2</sup> 在 SCL 示例中，“DataLogOpen\_DB”是背景 DB 的名称。

表格 8- 127 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	通过由低到高的（上升沿）信号启动操作。（默认值： False）
MODE	IN	UInt	工作模式： <ul style="list-style-type: none"> <li>• 0 - 附加到现有数据（默认值）</li> <li>• 1 - 清除所有现有记录</li> </ul>
NAME	IN	Variant	现有数据日志的名称： 此变量仅支持 String 数据类型，且只可位于本地、DB 或 M 存储器。（默认值： ''）
ID	输入/输出	DWord	数据日志的数字标识符。（默认值： 0） <b>注：</b> 该参数不支持符号名称访问。
DONE	OUT	Bool	上一请求已完成且没有出错后，DONE 位将保持为 TRUE 一个扫描周期时间。（默认值： False）
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 - 无操作正在进行</li> <li>• 1 - 操作正在进行</li> </ul>
ERROR	OUT	Bool	上一请求因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个扫描周期内有效。
STATUS	OUT	Word	执行条件代码（默认值： 0）

可提供已有数据日志的 NAME 或 ID（ID 参数作为输入）。如果同时提供这两个参数，但有效的 ID 与 NAME 数据日志不对应，则使用 ID，而忽略 NAME。

NAME 必须是 DataLogCreate 指令创建的数据日志的名称。如果只提供 NAME 且 NAME 指定一个有效数据日志，将返回对应的 ID（ID 参数作为输出）。

#### 说明

##### 数据日志文件的一般用法

- 执行 DataLogCreate 和 DataLogNewFile 操作后会自动打开数据日志文件。
- PLC 执行 RUN-STOP 切换或 PLC 循环上电后会自动关闭数据日志文件。
- 必须打开了数据日志文件，才能执行新的 DataLogWrite 操作。
- 最多可同时打开八个数据日志文件。可能存在八个以上数据日志文件，但必须关闭一些数据日志文件，使打开的文件数不超过八个。

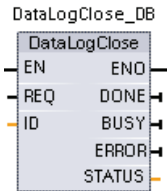
8.7 数据记录

表格 8- 128 ERROR 和 STATUS 的值

ERROR	STATUS (W#16#)	说明
0	0000	无错误
0	0002	警告： 数据日志文件已通过该应用程序打开
0	7000	无 REQ 沿时调用： BUSY = 0, DONE = 0
0	7001	有 REQ 沿时首次调用（工作）： BUSY = 1, DONE = 0
0	7002	第 N 次调用（工作）： BUSY = 1, DONE = 0
1	8070	所有内部实例存储器都在使用。
1	8090	数据日志定义与现有数据日志文件不一致。
1	8091	名称参数不是字符串引用。
1	8092	数据日志不存在。
1	80C0	数据日志文件被锁定。
1	80C1	打开的文件过多： 最多只允许同时打开 8 个数据日志文件。

8.7.2.3 DataLogClose

表格 8- 129 DataLogClose 指令

LAD/FBD	SCL	说明
	<pre>"DataLogClose_DB" (     req:=_bool_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     ID:=_dword_inout_);</pre>	<p>关闭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。</p> <p>切换到 STOP 模式时将关闭所有已打开的数据日志文件。</p> <p>STEP 7 会在插入指令时自动创建关联的背景数据块。</p>

<sup>2</sup> 在 SCL 示例中，“DataLogClose\_DB”是背景 DB 的名称。

表格 8- 130 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	通过由低到高的（上升沿）信号启动操作。（默认值：False）
ID	输入/输出	DWord	数据日志的数字标识符。仅用作 DataLogClose 指令的输入。（默认值：0） <b>注：</b> 该参数不支持符号名称访问。
DONE	OUT	Bool	上一请求已完成且没有出错后，DONE 位将保持为 TRUE 一个扫描周期时间。
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 – 无操作正在进行</li> <li>• 1 - 操作正在进行</li> </ul>
ERROR	OUT	Bool	上一请求因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个扫描周期内有效。
STATUS	OUT	Word	执行条件代码（默认值：0）

表格 8- 131 ERROR 和 STATUS 的值

ERROR	STATUS (W#16#)	说明
0	0000	无错误
0	0001	数据日志未打开
0	7000	无 REQ 沿时调用：BUSY = 0, DONE = 0
0	7001	有 REQ 沿时首次调用（工作）：BUSY = 1, DONE = 0
0	7002	第 N 次调用（工作）：BUSY = 1, DONE = 0
1	8092	数据日志不存在。

8.7.2.4 DataLogWrite

表格 8- 132 DataLogWrite 指令

LAD/FBD	SCL	说明
	<pre>"DataLogWrite_DB" (     req:=_bool_in_,     done=&gt;_bool_out_,     busy=&gt;_bool_out_,     error=&gt;_bool_out_,     status=&gt;_word_out_,     ID:=_dword_inout_);</pre>	<p>将数据记录写入指定的数据日志。已有目标数据日志处于打开状态时，才可执行 DataLogWrite 操作。</p> <p>STEP 7 会在插入指令时自动创建关联的背景数据块。</p>

2 在 SCL 示例中，“DataLogWrite\_DB”是背景 DB 的名称。

表格 8- 133 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	通过由低到高的（上升沿）信号启动操作。（默认值： False）
ID	输入/输出	DWord	数据日志数字标识符。仅用作 DataLogWrite 指令的输入。（默认值： 0） <b>注：</b> 该参数不支持符号名称访问。
DONE	OUT	Bool	上一请求已完成且没有出错后，DONE 位将保持为 TRUE 一个扫描周期时间。
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0 - 无操作正在进行</li> <li>1 - 操作正在进行</li> </ul>
ERROR	OUT	Bool	上一请求因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个扫描周期内有效。
STATUS	OUT	Word	执行条件代码（默认值： 0）

记录缓冲区的存储器地址和数据结构通过 DataLogCreate 指令的 DATA 参数组态。必须通过程序用当前运行过程值装载记录缓冲区，然后执行 DataLogWrite 指令将新记录数据从缓冲区移动到数据日志。

ID 参数用于标识数据日志和数据记录组态。ID 号在创建数据日志时生成。

如果循环数据日志文件中存在空记录，则将写入下一条可用的空记录。如果所有记录均不为空，则将覆盖最早的记录。

<b>小心</b>
<p><b>CPU 电源故障时数据日志数据丢失的可能性</b></p> <p>如果未完成 DataLogWrite 操作时发生电源故障，则当前正向数据日志传送的数据记录可能会丢失。</p>

表格 8- 134 ERROR 和 STATUS 的值

ERROR	STATUS (W#16#)	说明
0	0000	无错误
0	0001	表明数据日志已满： 创建的各数据日志只能存储指定的最大记录数。如果写入最后一条记录后达到最大记录数， 则下一写操作将覆盖最早的记录。
0	7000	无 REQ 沿时调用： BUSY = 0, DONE = 0
0	7001	有 REQ 沿时首次调用（工作）： BUSY = 1, DONE = 0
0	7002	第 N 次调用（工作）： BUSY = 1, DONE = 0
1	8070	所有内部实例存储器都在使用。
1	8092	数据日志不存在。
1	80B0	数据日志文件未打开（仅针对显式打开模式）。

### 8.7.2.5 DataLogNewFile

表格 8- 135 DataLogNewFile 指令

LAD/FBD	SCL	说明
	<pre>"DataLogNewFile_DB" (   req:=_bool_in_,   records:=_udint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   name:=_DataLog_out_,   ID:=_dword_inout_);</pre>	<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。</p> <p>STEP 7 会在插入指令时自动创建关联的背景数据块。</p>

2 在 SCL 示例中，“DataLogNewFile\_DB”是背景 DB 的名称。

表格 8- 136 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	通过由低到高的（上升沿）信号启动操作。（默认值： False）
RECORDS	IN	UDInt	覆盖最旧条目前循环数据日志可存储的最大数据记录数。（默认值： 1） 不包括标题记录。 CPU 装载存储器的可用空间必须足够大，以确保成功创建数据日志。
NAME	IN	Variant	数据日志名称： 用户可在此提供名称。此变量仅支持 String 数据类型，且只可位于本地、DB 或 M 存储器。（默认值： ''） 该字符串引用还用作数据日志文件名。名称中的字符必须遵守 Windows 文件系统命名限制。字符 \ / : * ? " < >   及空格字符均不许使用。
ID	输入/输出	DWord	数据日志数字标识符（默认值： 0）： <ul style="list-style-type: none"> <li>• 执行时，ID 输入标识有效数据日志。 将从该数据日志复制新数据日志组态。</li> <li>• 执行后，ID 参数成为返回新建数据日志文件的 ID 的输出。</li> </ul> <b>注：</b> 该参数不支持符号名称访问。
DONE	OUT	Bool	上一请求已完成且没有出错后，DONE 位将保持为 TRUE 一个扫描周期时间。
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 – 无操作正在进行</li> <li>• 1 - 操作正在进行</li> </ul>
ERROR	OUT	Bool	上一请求因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个扫描周期内有效。
STATUS	OUT	Word	执行条件代码（默认值： 0）

如果数据日志已满或被视为已完成，但您不想丢失数据日志中存储的任何数据，则可执行 DataLogNewFile 指令。可以根据此已写满的数据日志文件的结构创建一个新的空数据日志文件。 将从原始数据日志复制标题记录以及原始数据日志属性（DATA 记录缓冲区、数据格式和时间戳设置）。 隐式关闭原始数据日志文件并隐式打开新数据日志文件。

**DataLogWrite 参数触发器：** 程序必须监视每个 DataLogWrite 操作的 ERROR 和 STATUS 参数。 如果写入最后一条记录时数据日志已满，DataLogWrite ERROR 位 = 1 且 DataLogWrite STATUS 字 = 1。这些 ERROR 和 STATUS 值仅在一个扫描周期时间



内有效，因此监视逻辑必须使用 **ERROR = 1** 作为定时门来捕捉 **STATUS** 值，然后测试 **STATUS** 是否为 1（数据日志已满）。

**DataLogNewFile** 操作：当程序逻辑获得数据日志已满信号时，此状态用于激活 **DataLogNewFile** 操作。必须使用现有（通常已满）的已打开数据日志的 ID 执行 **DataLogNewFile**，但要使用新的唯一 **NAME** 参数。**DataLogNewFile** 操作完成后，将返回新的数据日志 ID 值（作为输出参数），该值与新的数据日志名称相对应。新数据日志文件隐式打开并可存储新记录。针对新数据日志文件的新的 **DataLogWrite** 操作必须使用 **DataLogNewFile** 操作返回的 ID 值。

#### 说明

**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。程序逻辑监视并捕捉到 **DataLogNewFile DONE** 位转换到 **TRUE** 状态后，新的数据日志才可用于其它数据日志操作。

表格 8- 137 ERROR 和 STATUS 的值

ERROR	STATUS (W#16#)	说明
0	0000	无错误
0	7000	无 REQ 沿时调用： <b>BUSY = 0, DONE = 0</b>
0	7001	有 REQ 沿时首次调用（工作）： <b>BUSY = 1, DONE = 0</b>
0	7002	第 N 次调用（工作）： <b>BUSY = 1, DONE = 0</b>
1	8070	所有内部实例存储器都在使用。
1	8090	文件名无效
1	8091	名称参数不是字符串引用。
1	8092	数据日志不存在。
1	8093	数据日志已存在。
1	8097	请求的文件长度超出文件系统最大值。
1	80B3	可用装载存储器空间不足。
1	80B4	<b>MC</b> 受写保护。
1	80C1	打开的文件过多。

### 8.7.3 使用数据日志

数据日志文件以逗号分隔值格式 (\*.csv) 存储在永久性闪存中。可以使用 PLC Web 服务器功能或通过取出 PLC 存储卡并将其插入标准 PC 读卡器中来查看数据日志。

#### 使用 PLC Web 服务器功能查看数据日志

如果 PLC PROFINET 端口和 PC 连接到网络，则可使用 PC Web 浏览器（如 Microsoft Internet Explorer 或 Mozilla Firefox）访问内置 PLC Web 服务器。运行 PLC Web 服务器时，PLC 可以处于运行模式或停止模式。如果 PLC 处于运行模式，则当 PLC Web 服务器通过网络传送日志数据时，控制程序会继续执行。

Web 服务器访问：

1. 在目标 CPU 的设备配置中启用 Web 服务器 (页 542)。
2. 通过 PROFINET 网络将 PC 连接到 PLC (页 543)。
3. 登录内置 Web 服务器 (页 545)。
4. 通过“数据日志”(Data Logs) 标准 Web 页面 (页 556) 下载最新记录、所有记录、清除记录或删除日志文件。
5. 将数据日志文件的副本下载到 PC 后，可使用电子表格应用程序（如 Excel）打开 .csv 文件。

#### 查看 PLC 存储卡中的数据日志

如果 S7-1200 CPU 中插入了“程序”型 S7-1200 存储卡，则可以取出该存储卡，然后将该卡插入 PC 或 PG 上的标准 SD（安全数码卡）卡槽或 MMC（多媒体卡）卡槽中。取出存储卡后，PLC 将处于停止模式，因此不执行控制程序。

在 Windows 资源管理器中导航至存储卡中的 \DataLog 目录。所有 \*.csv 数据日志文件都位于该目录下。

复制数据日志文件，然后将副本放到 PC 的本地驱动器中。接着，可以使用 Excel 打开 \*.csv 文件的本地副本，而不是存储在存储卡中的原始文件。

#### 小心

可以借助 PC 读卡器复制 S7-1200 存储卡中的数据日志文件但不要修改或删除这些文件。推荐使用标准 Web 服务器数据日志页面工具来查看、下载（复制）、清除（删除数据）和删除数据日志文件。Web 服务器可为您管理存储卡文件并有助于防止数据被意外修改或删除。

如果直接通过 Windows 资源管理器浏览存储卡文件系统，则可能意外删除/修改数据日志或其它系统文件，这样一来可能会损坏文件或使存储卡无法使用。

### 使用 Web 浏览器查看数据日志

即使没有使用 Web 服务器功能，也可使用 Internet Explorer 或 Mozilla Firefox 之类的 Web 浏览器直接查看数据日志。只需在浏览器的地址栏中输入以下文本，就可以使用 CPU 的 IP 地址和 STEP 7 中数据日志文件的实际名称进行查看而非“MyDataLog”。

**http://192.168.0.1/DataLog.html?FileName=MyDataLog.csv**

还可以通过第三方的文件收集工具访问固定地址的数据日志文件。

### 8.7.4 对数据日志文件大小的限制

数据日志文件与程序、程序数据、组态数据、用户定义的 Web 页面和 PLC 系统数据共享 PLC 装载存储器空间。使用内部装载存储器的大型程序需要大量的装载存储空间，因而可能造成用于数据日志文件的可用空间不足。在这种情况下，可以使用“程序卡”来增加装载存储器容量。S7-1200 CPU 既可以使用内部装载存储器也可以使用外部装载存储器，但不能同时使用。

有关如何创建“程序卡”的详细信息，请参见“存储卡”章节 (页 116)。

#### 单个数据日志文件的最大大小

单个数据日志文件的最大大小不可超过装载存储器（内部或外部）大小的 25%。如果应用程序需要更多的数据日志条目，则可以使用“DataLogNewFile”指令在第一个文件中的所有记录都已填充后创建一个新文件。有关单个数据日志文件的最大大小的信息，请参见下表。

表格 8- 138 装载存储器大小和单个数据日志文件的最大大小

数据区	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	数据存储
内部装载存储器 闪存	1 MB (单个数据日志文件最大 250 KB)	1 MB (单个数据日志文件最大 250 KB)	4 MB (单个数据日志文件最大 500 KB)	4 MB (单个数据日志文件最大 500 KB)	用户程序和程序数据、组态数据、数据日志、用户定义 Web 页面以及 PLC 系统数据
外部装载存储器 可选“程序卡”闪存卡	2 MB、12 MB 或 24 MB，具体取决于 SD 卡的容量 (使用 2 MB 卡时，单个数据日志文件最大 500 KB) (使用 24 MB 卡时，单个数据日志文件最大 6 MB)				

### 确定装载存储器可用空间的大小

1. 建立 STEP 7 和目标 S7-1200 PLC 之间的在线连接。
2. 下载要添加数据日志操作的程序。
3. 根据需要创建任何可选的用户定义 Web 页面。（用于访问数据日志的标准 Web 页面存储在 PLC 固件中，不占用装载存储器的空间）。
4. 使用 在线和诊断工具可获得装载存储器大小以及装载存储器可用空间的百分数 (页 736)。
5. 将装载存储器大小乘以可用空间百分数即得到当前可用的装载存储器空间。

### 所有数据日志最大总大小

正常运行期间操作系统会使用 and 释放存储空间，因此装载存储器可用空间量会随之变化。应将所有数据日志文件的总大小限制为可用空间的一半。

### 计算单个数据日志记录的存储器要求

日志数据以 CSV（逗号分隔值）文件格式存储为若干字符字节。下表给出了存储各数据类型所需的字节数。

表格 8- 139 CSV 文件数据大小

数据类型	字节数（数据字节加上逗号分隔符字节）
------	--------------------

Bool	2
Byte	5
Word	7
DWord	12
Char	4

**数据类型 字节数（数据字节加上逗号分隔符字节）**

String	<p><b>示例 1: MyString[10]</b>            最大字符串大小分配为 10 个字符。            文本字符 + 自动填充的空格字符 = 10 个字节            成对的双引号 + 逗号字符 = 3 个字节            10 + 3 = 13 个字节（总共）</p> <p><b>示例 2: Mystring2</b>            如果未使用方括号指定大小，则默认分配 254 个字节。            文本字符 + 自动填充的空格字符 = 254 个字节            成对的双引号 + 逗号字符 = 3 个字节            254 + 3 = 257 个字节（总共）</p>
USInt	5
UInt	7
UDInt	12
SInt	5
Int	7
DInt	12
Real	16
LReal	25
Time	15
DTL	24

**DataLogCreate DATA** 参数指向一个结构，该结构用于为一条数据日志记录指定数据字段数和各数据字段的数据类型。上表给出了各种数据类型在 CSV 文件中所需的字节数。将给出的数据类型的出现次数乘以该数据类型所需的字节数。对记录中的每种数据类型执行此操作，然后对字节数求和即得到数据记录的总大小。另加上一个字节（行尾字符占用）。

数据日志记录的大小 = 所有数据字段所需字节数之和 + 1（行尾字符）。

### 计算整个数据日志文件的存储器要求

**DataLogCreate** 指令的 **RECORDS** 参数用于设置数据日志文件中可存储的最大记录数。创建数据日志文件时，分配最大存储器大小。

数据日志文件大小 = (一条记录的字节数) x (记录数)。

### 8.7.5 数据日志示例程序

该数据日志示例程序未显示从动态过程获取采样值必需的所有程序逻辑，但显示了数据日志指令的重要操作。所使用的日志文件的结构和数目取决于过程控制要求。

#### 说明

##### 数据日志文件的一般用法

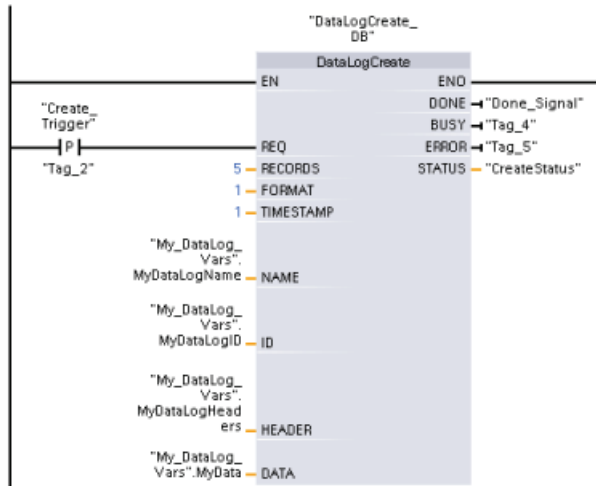
- 执行 **DataLogCreate** 和 **DataLogNew** 文件操作后会自动打开数据日志文件。
- PLC 执行 RUN-STOP 切换或 PLC 循环上电后会自动关闭数据日志文件。
- 必须打开了数据日志文件，才能执行 **DataLogWrite** 操作。
- 最多可同时打开八个数据日志文件。可能存在八个以上数据日志文件，但必须关闭一些数据日志文件，使打开的文件数不超过八个。

#### 示例数据日志程序

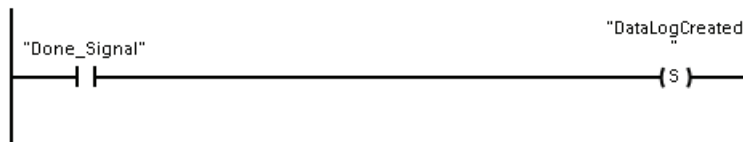
在数据块中创建示例数据日志名称、标题文本和 **MyData** 结构。三个 **MyData** 变量临时存储新的采样值。通过执行 **DataLogWrite** 指令将这些 DB 位置的过程采样值传送到数据日志文件。

My_Datalog_Vars				
	名称	数据类型	启动值	
1	Static			
2	MyNEWDatalogName	String	'MyNEWDatalog'	
3	MyDatalogName	String	'MyDatalog'	
4	MyDatalogID	DWord	0	
5	MyDatalogHeaders	String	'Count,Temperature,Pressure'	
6	MyData	Struct		
7	MyCount	Int	0	
8	MyTemperature	Real	0.0	
9	MyPressure	Real	0.0	

程序段 1 REQ 上升沿启动数据日志创建过程。



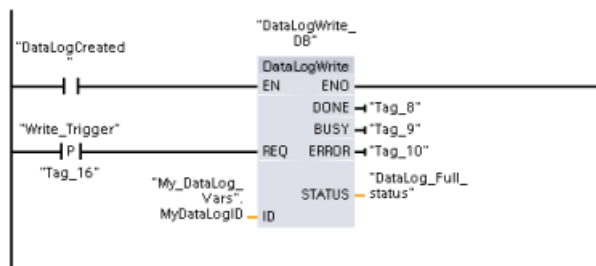
程序段 2 捕获 DataLogCreate 的 DONE 输出，考虑到该指令仅在一个扫描周期内有效。



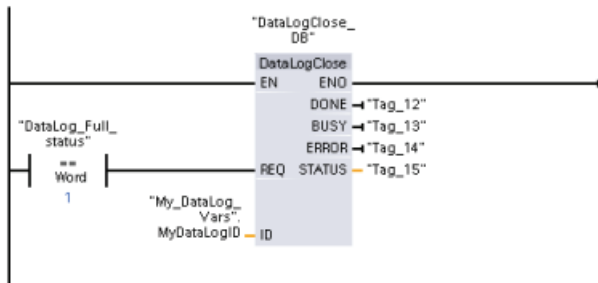
程序段 3 上升沿信号触发何时将新过程值存储在 MyData 结构中。



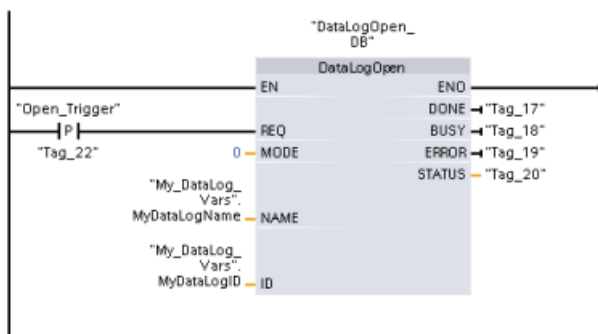
程序段 4 EN 输入状态取决于何时完成 DataLogCreate 操作。创建操作将跨越多个扫描周期，并且必须在执行写入操作之前完成。REQ 输入的上升沿信号是触发已启用写入操作的事件。



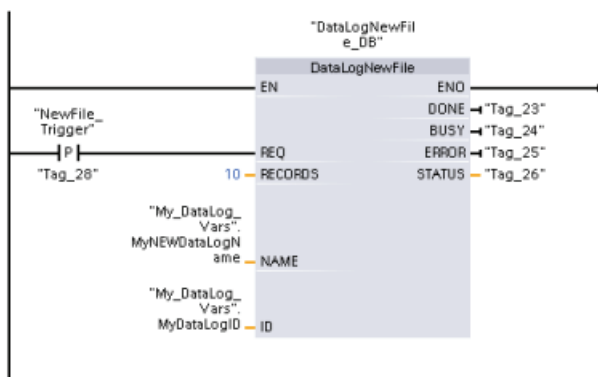
**程序段 5** 在写入最后一条记录后关闭数据日志。执行写入最后一条记录的 DataLogWrite 操作后，将通过 DataLogWrite STATUS 输出为 1 来指示日志文件已写满状态。



**程序段 6** DataLogOpen REQ 输入的上升信号沿会模拟用户按下 HMI 上的按钮打开数据日志文件的操作。如果打开所有记录都填满了过程数据的数据日志文件，则下一个 DataLogWrite 操作会覆盖最早的记录。您也许想保存之前的数据日志并创建新数据日志，如程序段 7 中所示。

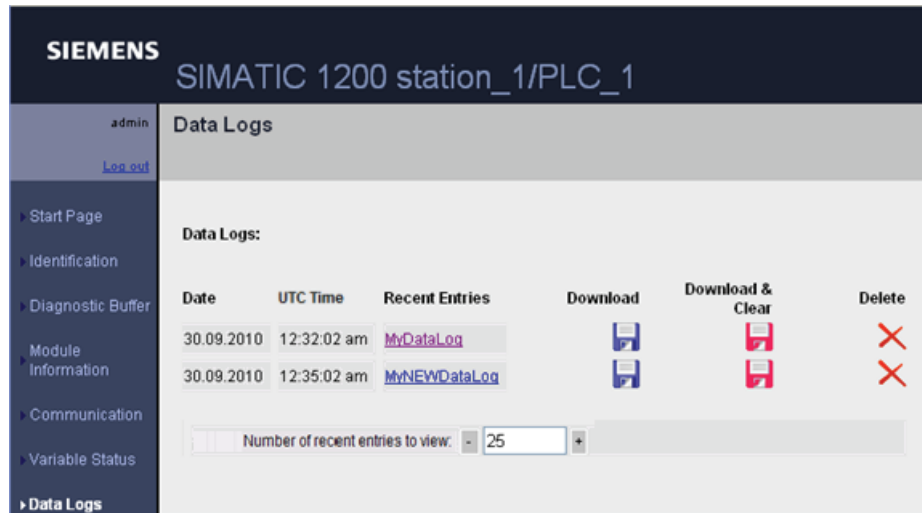


**程序段 7** ID 参数是 IN/OUT 类型。首先应现有数据日志的 ID 值，以便能复制其结构。完成 DataLogNewFile 操作后，会将新数据记录的唯一 ID 新值写回 ID 参考位置。未显示所需的 DONE 位 = TRUE 捕获，有关 DONE 位逻辑的示例，请参见程序段 1、2 和 4。





通过 S7-1200 CPU Webserver 看到的由示例程序创建的数据日志文件



表格 8- 140 用 Excel 查看的已下载 .csv 文件示例

在最多五条记录的文件中写入了两条记录		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	1	9/29/2010	21:01:46	5	5.00E+00	5.00E+00
	3	2	9/29/2010	21:01:47	5	5.00E+00	5.00E+00
	4	//END					
5							
最多五条记录的数据日志文件中有五条记录		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	1	9/30/2010	20:26:56	1	9.86E+01	3.52E+01
	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
7							
将又一条记录写入上述已写满的文件后，第六次写入操作会用第六条数据覆盖最早的第一条记录。另一次写入操作会用第七条记录覆盖第二条记录，依此类推。		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	6	9/30/2010	20:32:03	6	9.86E+01	3.58E+01
	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
7							

## 8.8 数据块控制

### 8.8.1 READ\_DBL、WRIT\_DBL（读取或写入装载存储器中的 DB）

表格 8- 141 READ\_DBL 和 WRIT\_DBL 指令

LAD/FBD		说明
	<pre>READ_DBL (     req:=_bool_in_,     srcblk:=_variant_in_,     busy=&gt;_bool_out_,     dstblk=&gt;_variant_out_);</pre>	<p>将 DB 的全部或部分起始值从装载存储器复制到工作存储器的目标 DB 中。</p> <p>在复制期间，装载存储器的内容不变。</p>
	<pre>WRIT_DBL (     req:=_bool_in_,     srcblk:=_variant_in_,     busy=&gt;_bool_out_,     dstblk=&gt;_variant_out_);</pre>	<p>将 DB 全部当前值或部分值从工作存储器复制到装载存储器的目标 DB 中。</p> <p>在复制期间，工作存储器的内容不变。</p>

表格 8- 142 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	BOOL	如果 BUSY = 0，则高电平信号会启动操作。
SRCBLK	IN	VARIANT	READ_DBL: 指向装载存储器中源数据块的指针 WRIT_DBL: 指向工作存储器中源数据块的指针
RET_VAL	OUT	INT	执行条件代码
BUSY	OUT	BOOL	BUSY = 1 表示读取/写入过程尚未完成。
DSTBLK	OUT	VARIANT	READ_DBL: 指向工作存储器中目标数据块的指针 WRIT_DBL: 指向装载存储器中目标数据块的指针

通常，DB 存储在装载存储器（闪存）和工作存储器 (RAM) 中。起始值（初始值）始终存储在装载存储器中，当前值始终存储在工作存储器中。READ\_DBL 可用于将一组起始值从装载存储器复制到工作存储器中程序引用的 DB 的当前值。可使用 WRIT\_DBL 将存储在内部装载存储器或存储卡中的起始值更新为工作存储器中的当前值。

---

#### 说明

##### 避免过多 WRIT\_DBL 闪存写入操作

WRIT\_DBL 指令会在闪存（内部装载存储器或存储卡）内执行写入操作。WRIT\_DBL 应该用于诸如生产流程更改等的非经常性更新。

---

事先必须已通过 STEP 7 创建了 READ\_DBL 和 WRIT\_DBL 要使用的数据块，才能使用这些指令。如果源数据块被创建成“标准”类型，则目标数据块也必须为“标准”类型。如果源数据块被创建成“优化”类型，则目标数据块也必须为“优化”类型。

如果 DB 为标准 DB，则可指定一个变量名称或 P# 值。P# 值允许指定和复制指定大小（字节、字或双字）的任意数量的元素。因此，可复制 DB 的全部或部分内容。如果 DB 是优化 DB，则只能指定一个变量名称；无法使用 P# 操作符。如果为标准或优化 DB（或者其它工作存储器类型）指定变量名称，则会复制此变量引用的所有内容。可以是用户定义类型、数组或基本元素。如果 DB 是标准 DB 而不是优化 DB，则这些指令只能使用数据类型结构。如果它是优化存储器中的结构，则必须使用用户定义类型 (UDT)。仅用户定义类型可确保源结构和目标结构的“数据类型”完全相同。

---

#### 说明

##### 在“优化”DB 中使用结构（数据类型 Struct）

在“优化”DB 中使用 Struct 数据类型时，首先必须为 Struct 创建一个用户定义的数据类型 (UDT)。然后通过 UDT 组态源 DB 和目标 DB。UDT 确保针对两个 DB Struct 内的数据类型保持一致。

对于“标准”DB,使用 Struct 时无需创建 UDT。

---

READ\_DBL 和 WRIT\_DBL 相对于循环程序扫描异步执行。处理期间需要多次调用 READ\_DBL 和 WRIT\_DBL 指令。通过 REQ = 1 来调用指令启动 DB 传输作业，然后监视 BUSY 和 RET\_VAL 输出以确定数据传输的完成时间以及是否正确。

为确保数据的一致性，请勿在 READ\_DBL 处理过程中修改目标区域或在 WRIT\_DBL 处理过程中修改来源区域（即，不要在 BUSY 参数为 TRUE 时修改）。

8.8 数据块控制

SRCBLK 和 DSTBLK 参数限制:

- 数据块必须先创建，然后才可引用。
- BOOL 类型的 VARIANT 指针长度必须可被 8 整除。
- 源指针和目标指针中 STRING 类型的 VARIANT 指针的长度必须相同。

配方和机器设置信息

可使用 READ\_DBL 和 WRIT\_DBL 指令来管理配方或机器设置信息。虽然可以限制写入的次数以防止闪存损耗，但以上方法实际上是另一种归档那些值不经常更改的保持性数据的方法。这样即可在提供给常规掉电保持性数据的容量的基础上有效增加保持性存储器的容量，至少可满足不经常更改值的需要。可使用 WRIT\_DBL 指令将配方信息或机器设置信息从工作存储器保存到装载存储器，并使用 READ\_DBL 指令将这些信息从装载存储器提取到工作存储器。

表格 8- 143 条件代码

RET_VAL (W#16#...)	说明
0000	无错误
0081	警告：源区域小于目标区域。源数据已完全复制，目标区域中的额外字节未更改。
7000	REQ = 0 时调用：BUSY = 0
7001	REQ = 1 时首次调用（工作）：BUSY = 1
7002	第 N 次调用（工作）：BUSY = 1
8051	数据块类型错误
8081	源区域大于目标区域。目标区域已满，源数据中的剩余字节被忽略。
8251	源数据块类型错误
82B1	缺少源数据块
82C0	源 DB 正在被其它语句或通信功能编辑。
8551	目标数据块类型错误
85B1	缺少目标数据块
85C0	目标 DB 正在被其它语句或通信功能编辑。
80C3	当前已有超过 50 个的 READ_DBL 或 WRIT_DBL 语句排队等候执行。

## 8.9 “扩展”指令的常见错误代码

表格 8- 144 扩展指令的常见错误代码

条件代码 (W#16#...) <sup>1</sup>	说明
8022	存储区对于输入太小
8023	存储区对于输出太小
8024	输入区非法
8025	输出区非法
8028	输入位赋值非法
8029	输出位赋值非法
8030	输出区是只读 DB。
803A	DB 不存在。

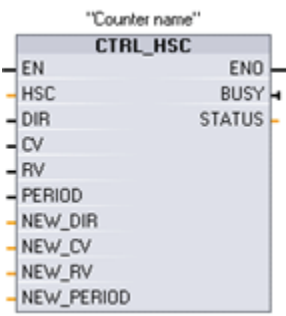
- <sup>1</sup> 如果执行代码块时发生以上错误之一，则 CPU 将进入 STOP 模式，除非在该代码块中使用 `GetError` 或 `GetErrorID` 指令并编写程序来响应错误。

### 8.9 “扩展”指令的常见错误代码

## 工艺指令

## 9.1 高速计数器

表格 9-1 CTRL\_HSC 指令

LAD/FBD	SCL	说明
	<pre>"CTRL_HSC_0_DB" (   hsc:=_hw_hsc_in_,   dir:=_bool_in_,   cv:=_bool_in_,   rv:=_bool_in_,   period:=_bool_in_,   new_dir:=_int_in_,   new_cv:=_int_in_,   new_rv:=_dint_in_,   new_period:=_int_in_,   busy:=_bool_out_,   status:=_word_out_);</pre>	<p>每个 CTRL_HSC 指令都使用 DB 中存储的结构来保存数据。在编辑器中放置 CTRL_HSC 指令后分配 DB。</p>

- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“CTRL\_HSC\_0\_DB”是背景 DB 的名称。

表格 9-2 参数的数据类型

参数和类型	数据类型	说明	
HSC	IN	HW_HSC	HSC 标识符
DIR <sup>1,2</sup>	IN	Bool	1 = 请求新方向
CV <sup>1</sup>	IN	Bool	1 = 请求设置新的计数器值
RV <sup>1</sup>	IN	Bool	1 = 请求设置新的参考值
PERIOD <sup>1</sup>	IN	Bool	1 = 请求设置新的周期值 (仅限频率测量模式)
NEW_DIR	IN	Int	新方向: 1= 向上, -1= 向下
NEW_CV	IN	DInt	新计数器值
NEW_RV	IN	DInt	新参考值
NEW_PERIOD	IN	Int	以秒为单位的新周期值: 0.01、0.1 或 1 (仅限频率测量模式)

## 9.1 高速计数器

参数和类型		数据类型	说明
BUSY <sup>3</sup>	OUT	Bool	功能忙
STATUS	OUT	Word	执行条件代码

- 1 如果不请求更新参数值，则将忽略相应的输入值。
- 2 仅当组态的计数方向设置为“用户程序（内部方向控制）”(User program (internal direction control)) 时，DIR 参数才有效。用户在 HSC 设备组态中确定如何使用该参数。
- 3 对于 CPU 或 SB 上的 HSC，BUSY 参数的值始终为 0。

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。

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（仅限频率测量模式）设置为 NEW\_PERIOD 值

如果执行 CTRL\_HSC 指令后以下布尔标记值置位为 1，则相应的 NEW\_xxx 值将装载到计数器。CTRL\_HSC 指令执行一次可处理多个请求（同时设置多个标记）。

- DIR = 1 是装载 NEW\_DIR 值的请求，0 = 无变化
- CV = 1 是装载 NEW\_CV 值的请求，0 = 无变化
- RV = 1 是装载 NEW\_RV 值的请求，0 = 无变化
- PERIOD = 1 是装载 NEW\_PERIOD 值的请求，0 = 无变化

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。

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。返回给程序的值将是读取计数器瞬间的正确计数。但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。

**条件代码：** 发生错误时，ENO 设置为 0，并且 STATUS 输出包含条件代码。



表格 9-3 STATUS 值 (W#16#)

STATUS	说明
0	无错误
80A1	HSC 标识符没有对 HSC 寻址
80B1	NEW_DIR 的值非法
80B2	NEW_CV 的值非法
80B3	NEW_RV 的值非法
80B4	NEW_PERIOD 的值非法
80C0	多路访问高速计数器
80D0	CPU 硬件配置中未启用高速计数器 (HSC)

### 9.1.1

#### 高速计数器的使用方法

高速计数器 (HSC) 对发生速率快于 OB 执行速率的事件进行计数。如果待计数事件的发生速率处于 OB 执行速率范围内, 则可使用 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。CTRL\_HSC 指令允许用户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。

先是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。(或者, 所有中断事件也可在单个中断例程中进行处理。)

## 9.1 高速计数器

表格 9-4 最大频率 (KHz)

HSC		单相	双相和 AB 正交
HSC1	CPU	100 KHz	80 KHz
	高速 SB	200 KHz	160 KHz
	SB	30 KHz	20 KHz
HSC2	CPU	100 KHz	80 KHz
	高速 SB	200 KHz	160 KHz
	SB	30 KHz	20 KHz
HSC3	CPU	100 KHz	80 KHz
HSC4	CPU	30 KHz	20 KHz
HSC5	CPU	30 KHz	20 KHz
	高速 SB	200 KHz	160 KHz
	SB	30 KHz	20 KHz
HSC6	CPU	30 KHz	20 KHz
	高速 SB	200 KHz	160 KHz
	SB	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。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

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。

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 处于使用内置输入但不使用外部复位 (IO.3) 的模式，则 IO.3 可以用于沿中断或 HSC2。

表格 9-5 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	(可选：方向)	-	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 <sup>1</sup>	? 计数

<sup>1</sup> 对于编码器：Z 相，归位

## HSC 的输入地址

### 说明

在设备配置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 点的地址分配给这些设备之后，无法通过监视表格中的强制功能修改所分配的 I/O 点的地址值。

9.1 高速计数器

组态 CPU 时，可以选择启用和组态每个 HSC。CPU 会根据其组态自动为每个 HSC 分配输入地址。（某些 HSC 允许选择是使用 CPU 的板载输入还是使用 SB 的输入。）

**注意**

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。

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 1211C 的板载 I/O 和 SB 两者的 HSC 输入分配。（如果 SB 只有 2 个输入，则仅输入 4.0 和 4.1 可用。）

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

表格 9-6 CPU 1211C 的 HSC 输入分配

HSC		CPU 板载输入 (0.x)						SB 输入（默认为 4.x） <sup>3</sup>			
		0	1	2	3	4	5	0	1	2	3
HSC 1 <sup>1</sup>	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 <sup>1</sup>	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]				
	双相					CU	CD				
	AB 相					A	B				
HSC 5 <sup>2</sup>	单相							C	[d]		[R]
	双相							CU	CD		[R]
	AB 相							A	B		[R]

HSC		CPU 板载输入 (0.x)						SB 输入 (默认为 4.x) <sup>3</sup>			
		0	1	2	3	4	5	0	1	2	3
HSC 6 <sup>2</sup>	单相								[R]	C	[d]
	双相								[R]	CU	CD
	AB 相								[R]	A	B

1 HSC 1 和 HSC 2 可组态为使用板载输入或 SB 输入。

2 HSC 5 和 HSC 6 只能使用 SB 输入。HSC 6 只能使用 4 输入 SB。

3 仅具有 2 个数字量输入的 SB 只能提供输入 4.0 和 4.1。

下表显示了 CPU 1212C 的板载 I/O 和 SB 两者的 HSC 输入分配。（如果 SB 只有 2 个输入，则仅输入 4.0 和 4.1 可用。）

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

表格 9-7 CPU 1212C 的 HSC 输入分配

HSC		CPU 板载输入 (0.x)								SB 输入 (4.x) <sup>3</sup>			
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 1 <sup>1</sup>	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 <sup>1</sup>	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]				
	双相					CU	CD		[R]				
	AB 相					A	B		[R]				
HSC 4	单相						[R]	C	[d]				
	双相						[R]	CU	CD				

9.1 高速计数器

HSC		CPU 板载输入 (0.x)								SB 输入 (4.x) <sup>3</sup>			
		0	1	2	3	4	5	6	7	0	1	2	3
	AB 相						[R]	A	B				
HSC 5 <sup>2</sup>	单相									C	[d]		[R]
	双相									CU	CD		[R]
	AB 相									A	B		[R]
HSC 6 <sup>2</sup>	单相										[R]	C	[d]
	双相										[R]	CU	CD
	AB 相										[R]	A	B

- 1 HSC 1 和 HSC 2 可组态为使用板载输入或 SB 输入。
- 2 HSC 5 和 HSC 6 只能使用 SB 输入。HSC 6 只能使用 4 输入 SB。
- 3 仅具有 2 个数字量输入的 SB 只能提供输入 4.0 和 4.1。

以下两个表给出了 CPU 1214C 的板载 I/O 和可选 SB（如果已安装）两者的 HSC 输入分配。

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

表格 9-8 CPU 1214C 和 CPU 1215C（仅板载输入）的 HSC 输入分配

HSC		数字量输入 0（默认值：0.x）								数字量输入 1（默认值：1.x）					
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 1 <sup>1</sup>	单相	C	[d]		[R]										
	双相	CU	CD		[R]										
	AB 相	A	B		[R]										
HSC 2 <sup>1</sup>	单相		[R]	C	[d]										
	双相		[R]	CU	CD										
	AB 相		[R]	A	B										
HSC 3	单相					C	[d]		[R]						

HSC		数字量输入 0 (默认值: 0.x)							数字量输入 1 (默认值: 1.x)						
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
	双相					CU	CD		[R]						
	AB 相					A	B		[R]						
HSC 4	单相							[R]	C	[d]					
	双相							[R]	CU	CD					
	AB 相							[R]	A	B					
HSC 5 <sup>1</sup>	单相									C	[d]	[R]			
	双相									CU	CD	[R]			
	AB 相									A	B	[R]			
HSC 6 <sup>1</sup>	单相												C	[d]	[R]
	双相												CU	CD	[R]
	AB 相												A	B	[R]

<sup>1</sup> HSC 1、HSC 2、HSC 5 和 HSC 6 可组态为使用板载输入或 SB 输入。

表格 9-9 SB 的 HSC 输入分配

HSC <sup>1</sup>		SB 输入 (默认值: 4.x) <sup>2</sup>			
		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

9.1 高速计数器

HSC 1	SB 输入 (默认值: 4.x) <sup>2</sup>			
	0	1	2	3

- 1 对于 CPU 1214C: HSC 1、HSC 2、HSC 5 和 HSC 6 可组态为使用板载输入或 SB 输入。
- 2 仅具有 2 个数字量输入的 SB 只能提供输入 4.0 和 4.1。

访问 HSC 的当前值

说明

启用脉冲发生器作为 PTO 时，会向该 PTO 分配一个对应的 HSC。HSC1 分配给 PTO1，HSC2 分配给 PTO2。分配的 HSC 完全属于 PTO 通道，并且禁用 HSC 的正常输出。HSC 值仅用于内部功能。生成脉冲时，不能监视当前值（例如，在 ID1000 中）。

CPU 将每个 HSC 的当前值存储在一个输入 (I) 地址中。下表列出了为每个 HSC 的当前值分配的默认地址。可以通过在设备配置中修改 CPU 的属性来更改当前值的 I 地址。

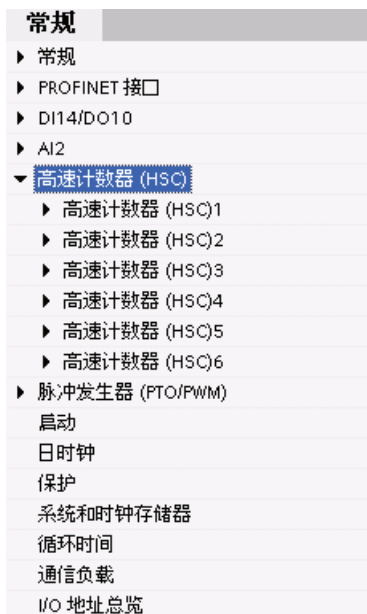
表格 9- 10 HSC 的当前值

HSC	数据类型	默认地址
HSC1	DInt	ID1000
HSC2	DInt	ID1004
HSC3	DInt	ID1008
HSC4	DInt	ID1012
HSC5	DInt	ID1016
HSC6	DInt	ID1020



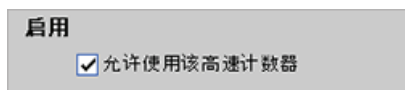
## 9.1.2

## 组态 HSC



CPU 允许用户组态最多 6 个高速计数器。用户可编辑 CPU 的“属性”(Properties) 来组态各个 HSC 的参数。在用户程序中使用 CTRL\_HSC 指令控制 HSC 的运行。

通过选择该 HSC 的“启用”(Enable) 选项启用特定的 HSC。



## 说明

当启用高速计数器并为其选择输入点时，这些点的输入滤波设置将组态为 800 ns。每一个输入点都有一个适用于所有应用的滤波器组态：过程输入、中断、脉冲捕捉和 HSC 输入。


 警告

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。

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 screenshot shows a configuration window for HSC with the following settings:

- 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

有关对 HSC 进行组态的信息，请参见 组态 CPU (页 125) 部分。

## 9.2 PID 控制

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。

### 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。

两个 PID 指令（PID\_3Step 和 PID\_Compact）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 <sub>p</sub>	比例增益 (P 分量)	a	微分延迟系数 (D 分量)
T <sub>i</sub>	积分作用时间 (I 分量)	b	比例作用加权 (P 分量)
T <sub>D</sub>	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 <sub>p</sub>	比例增益 (P 分量)	a	微分延迟系数 (D 分量)
T <sub>i</sub>	积分作用时间 (I 分量)	b	比例作用加权 (P 分量)
T <sub>D</sub>	微分作用时间 (D 分量)	c	微分作用加权 (D 分量)


### 9.2.1 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。工艺对象包括控制环的所有设置。
- 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。工艺对象包括控制环的所有设置。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 393)。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 396)。

表格 9- 11 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，应 组态工艺对象的参数 (页 393)。</p>	
---	---

表格 9- 12 (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。通过在将 PID 指令插入用户程序之前创建工艺对象，用户可以在插入 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类型的 PID 控制器 (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>	

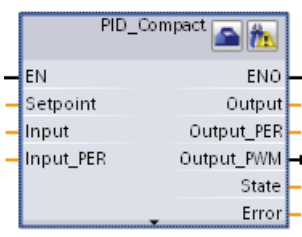
### 9.2.2 PID\_Compact 指令

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 <sub>p</sub>	比例增益 (P 分量)	a	微分延迟系数 (D 分量)
T <sub>i</sub>	积分作用时间 (I 分量)	b	比例作用加权 (P 分量)
T <sub>d</sub>	微分作用时间 (D 分量)	c	微分作用加权 (D 分量)

表格 9- 13 PID\_Compact 指令

LAD/FBD	SCL	说明
	<pre>"PID_Compact_1" (   Setpoint:=_real_in_,   Input:=_real_in_,   Input_PER:=_word_in_,   ManualEnable:=_bool_in_,   ManualValue:=_real_in_,   Reset:=_bool_in_,   ScaledInput=&gt;_real_out_,   Output=&gt;_real_out_,   Output_PER=&gt;_word_out_,   Output_PWM=&gt;_bool_out_,   SetpointLimit_H=&gt;_bool_out_,   SetpointLimit_L=&gt;_bool_out_,   InputWarning_H=&gt;_bool_out_,   InputWarning_L=&gt;_bool_out_,   State=&gt;_int_out_,   Error=&gt;_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 PIDT1 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 9- 14 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	过程值。默认值： 0.0 还必须设置 sPid_Cmpt.b_Input_PER_On = FALSE。
Input_PER	IN	Word	模拟过程值（可选）。默认值： W#16#0 还必须设置 sPid_Cmpt.b_Input_PER_On = TRUE。
ManualEnable	IN	Bool	启用或禁用手动操作模式。默认值： FALSE: <ul style="list-style-type: none"> <li>• PID_Compact V1.0 和 V1.2: 当 CPU 切换到 RUN 时, 如果 ManualEnable = TRUE, 则 PID_Compact 在手动模式下启动。将 PID_Compact 置于手动模式无需从 FALSE 切换到 TRUE。</li> <li>• PID_Compact V1.1: 当 CPU 切换到 RUN 并且 ManualEnable = TRUE 时, PID Compact 在上一个状态下启动。将 PID_Compact 置于手动模式需要从 TRUE 切换到 FALSE 再切换到 TRUE。</li> </ul>
ManualValue	IN	Real	手动操作的过程值。默认值： 0.0
Reset	IN	Bool	Reset 参数用于重新启动控制器。默认值： FALSE 请参见下面的“复位响应”部分以查看 PID_Compact V1.1 和 V1.0 的复位响应图。
ScaledInput	OUT	Real	标定的过程值。默认值： 0.0
Output <sup>1</sup>	OUT	Real	输出值。默认值： 0.0
Output_PER <sup>1</sup>	OUT	Word	模拟量输出值。默认值： W#16#0
Output_PWM <sup>1</sup>	OUT	Bool	脉冲宽度调制的输出值。默认值： FALSE
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。默认值： FALSE
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。默认值： FALSE
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

参数和类型		数据类型	说明
InputWarning_L	OUT	Bool	如果 InputWarning_L = TRUE，则说明过程值已达到警告下限。默认值： FALSE
State	OUT	Int	PID 控制器的当前操作模式。默认值： 0 使用 sRet.i_Mode 更改模式。 <ul style="list-style-type: none"> <li>• State = 0: 未激活</li> <li>• State = 1: 预调节</li> <li>• State = 2: 手动精确调节</li> <li>• State = 3: 自动模式</li> <li>• State = 4: 手动模式</li> </ul>
ErrorBits	OUT	DWord	PID_Compact 指令 ErrorBits 参数表 (页 383)定义错误消息。 默认值： DW#16#0000 (无错误)

<sup>1</sup> 参数 Output、Output\_PER 和 Output\_PWM 的输出可并行使用。

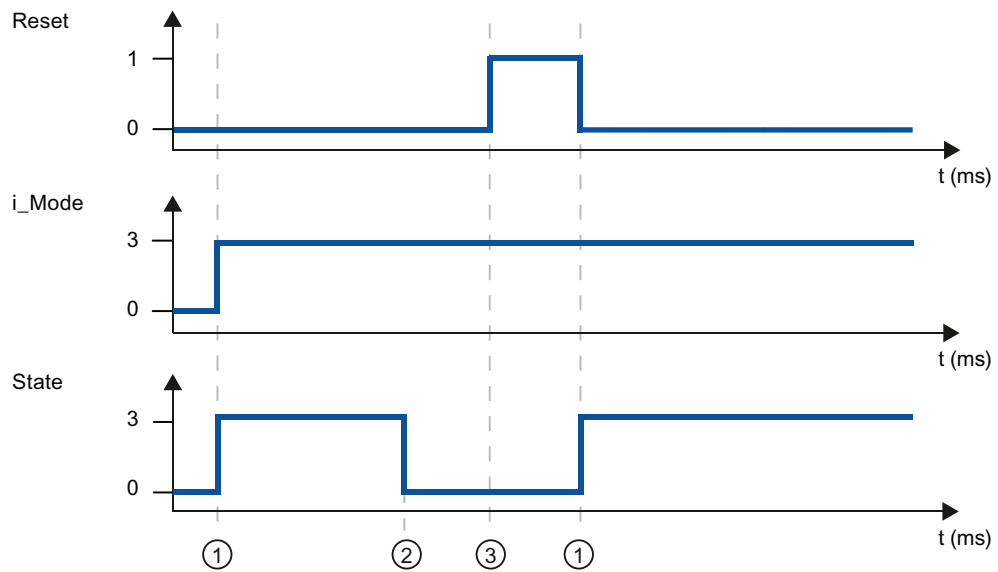
## 复位响应

复位响应是否为 TRUE 取决于 PID\_Compact 指令的版本。

### 复位响应 PID\_Compact 1.1

“复位”的上升沿将复位错误和警告并清除积分作用。“复位”的下降沿将触发对最近激活的操作模式的更改。

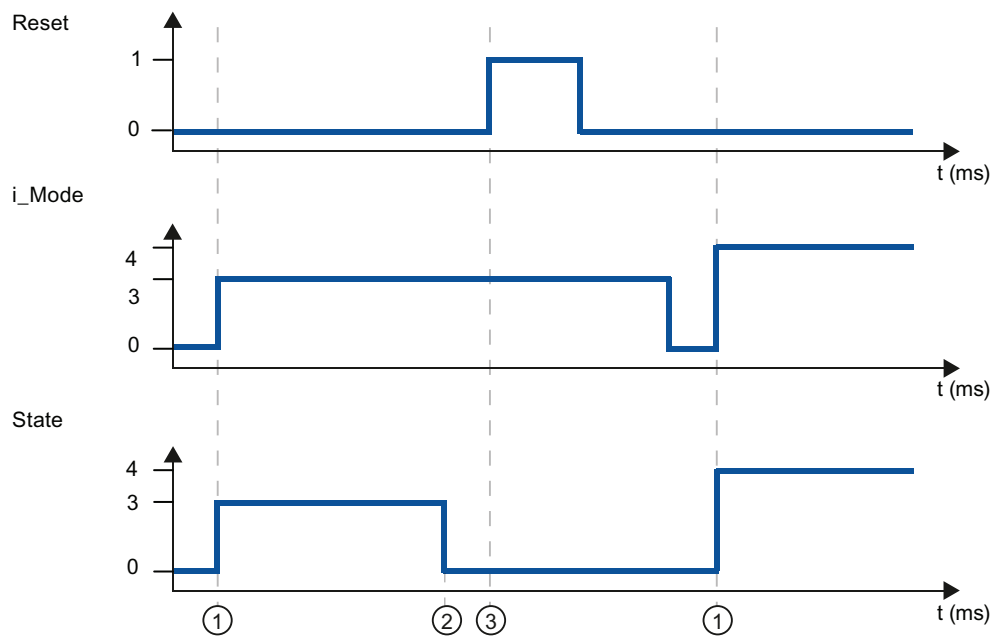




- ① 激活
- ② 错误
- ③ 复位

#### 复位响应 PID\_Compact V1.0

“复位”的上升沿将复位错误和警告并清除积分作用。控制器直到 i\_Mode 的下一个沿时才会重新激活。



- ① 激活
- ② 错误
- ③ 复位

**PID\_Compact 控制器的操作**

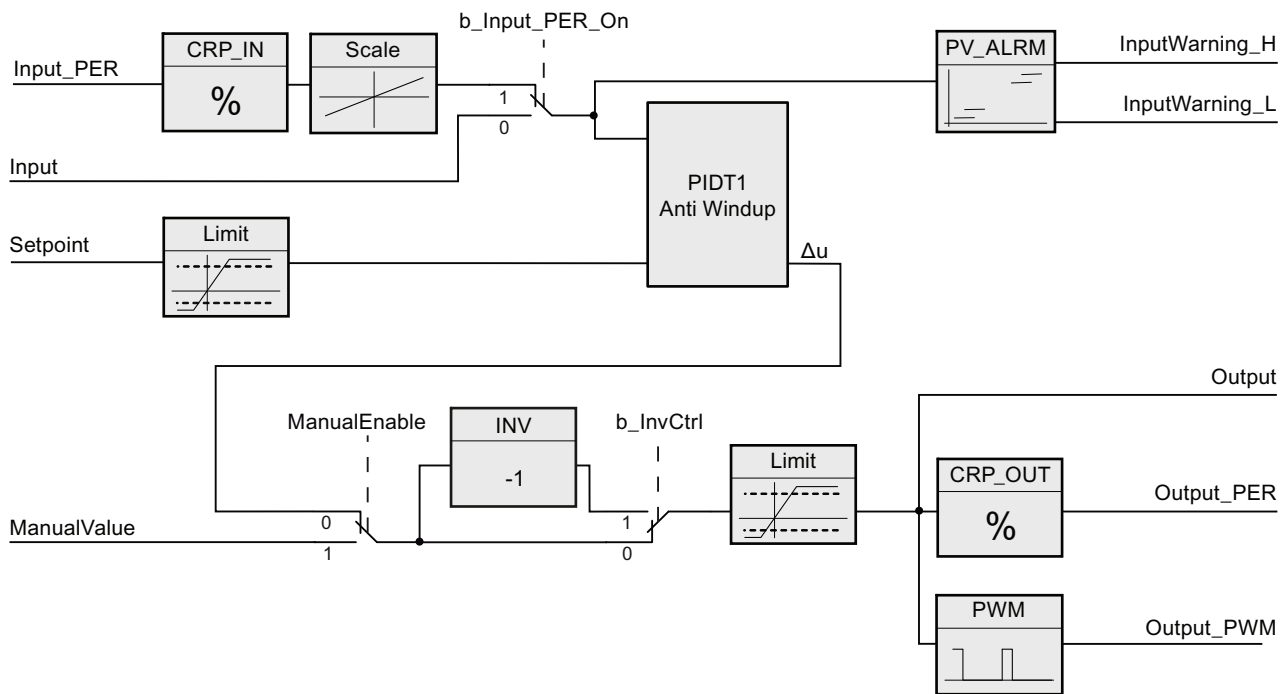


图 9-1 PID\_Compact 控制器的操作

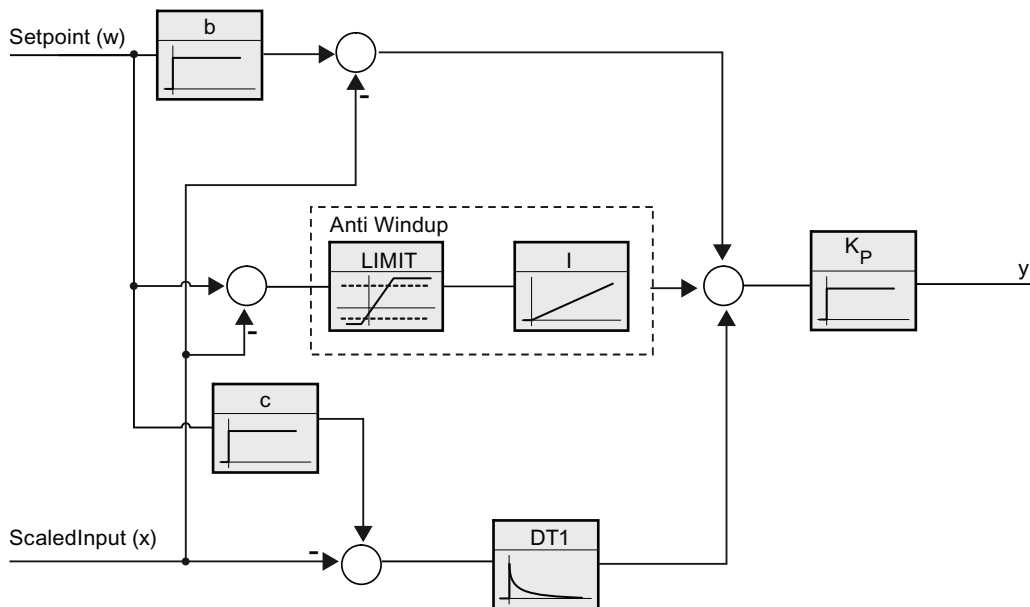


图 9-2 PID\_Compact 控制器作为具有抗积分饱和功能的 PIDT1 控制器时的操作

### 9.2.3 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 9- 15 PID\_Compact 指令 ErrorBit 参数

ErrorBit (DW#16#...)	说明
0000	无错误
0001	“Input”参数超出过程值限值。 Input > sPid_Cmpt.r_Pv_HlMor Input < sPid_Cmpt.r_Pv_LlM 在消除错误之前不能再次启动执行器。
0002	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	精确调节时出错，无法保持过程值振荡。
0008	开始预调节时出错。过程值过于接近设定值。开始精确调节。
0010	在控制器调节期间设定值已更改。
0020	在自动模式下或精确调节期间可能无法执行预调节。
0040	精确调节时出错，设定值过于接近设定值限值。
0080	输出值限值的组态不正确。 进行检查，确定是否正确配置输出值的限值，并匹配控件工作时采用的方向。
0100	控制器调节期间出错导致参数无效。
0200	参数“Input”的值无效：值的数字格式无效。
0400	计算输出值失败。检查 PID 参数。
0800	采样时间错误：在循环中断 OB 的采样时间内未调用 PID_Compact。
1000	参数“Setpoint”的值无效：值的数字格式无效。

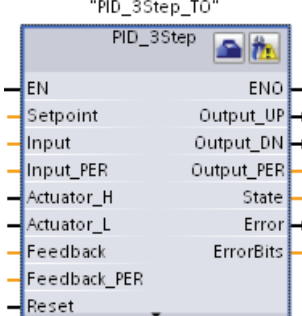
### 9.2.4 PID\_3STEP 指令

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 <sub>p</sub>	比例增益 (P 分量)	a	微分延迟系数 (D 分量)
T <sub>i</sub>	积分作用时间 (I 分量)	b	比例作用加权 (P 分量)
T <sub>d</sub>	微分作用时间 (D 分量)	c	微分作用加权 (D 分量)

表格 9- 16 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_,   ManualEnable:= _bool_in_,   ManualUP:= _bool_in_,   ManualDN:= _bool_in_,   ActuatorH:= _bool_in_,   ActuatorL:= _bool_in_,   Reset:= _bool_in_,   ScaledInput=&gt; _real_out_,   ScaledFeedback=&gt; _real_out_,   ErrorBits=&gt; _dword_out_,   OutputPer=&gt; _word_out_,   State=&gt; _int_out_,   OutputUP=&gt; _bool_out_,   OutputDN=&gt; _bool_out_,   SetpoIntLimitH=&gt; _bool_out_,   SetpoIntLimitL=&gt; _bool_out_,   InputWarningH=&gt; _bool_out_,   InputWarningL=&gt; _bool_out_,   Error=&gt; _bool_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 PIDT1 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 9- 17 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	过程值。默认值： 0.0 还必须设置 Config.InputPEROn = FALSE。
Input_PER	IN	Word	模拟过程值（可选）。默认值： W#16#0 还必须设置 Config.InputPEROn = TRUE。
ManualEnable	IN	Bool	启用或禁用手动操作模式。默认值： FALSE <ul style="list-style-type: none"> <li>在从 FALSE 变为 TRUE 的信号沿出现时，PID 控制器会切换为手动模式，State = 4 和 Retain.Mode 保持不变。</li> <li>在从 TRUE 变为 FALSE 的信号沿出现时，PID 控制器会切换为上一次的激活操作模式且 State = Retain.Mode。</li> </ul>
ManualUP	IN	Bool	在手动模式下，每个上升沿都会将阀门打开总动作范围的 5%，或者保持最小电机动作时间。仅当未使用 Output_PER 并且无位置反馈时，才评估 ManualUP。默认值： FALSE <ul style="list-style-type: none"> <li>如果 Output_PER 为 FALSE，则手动输入打开 Output_UP，持续时间对应于设备的 5% 运动量。</li> <li>如果 Config.ActuatorEndStopOn 为 TRUE，则在 Actuator_H 为 TRUE 时 Output_UP 不打开。</li> </ul>
ManualDN	IN	Bool	在手动模式下，每个上升沿都会将阀门关闭总动作范围的 5%，或者保持最小电机动作时间。仅当未使用 Output_PER 并且无位置反馈时，才评估 ManualDN。默认值： FALSE <ul style="list-style-type: none"> <li>如果 Output_PER 为 FALSE，则手动输入打开 Output_DN，持续时间对应于设备的 5% 运动量。</li> <li>如果 Config.ActuatorEndStopOn 为 TRUE，则在 Actuator_L 为 TRUE 时 Output_DN 不打开。</li> </ul>
ManualValue	IN	Real	手动操作的过程值。默认值： 0.0 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。默认值： 0.0
Feedback	IN	Real	阀门的位置反馈。默认值： 0.0 要使用 Feedback，应设置 Config.FeedbackPerOn = FALSE。

参数和类型		数据类型	说明
Feedback_PER	IN	Word	阀门位置的模拟反馈。默认值：W#16#0 要使用 Feedback_PER，应设置 Config.FeedbackPerOn = TRUE。通过以下参数标定 Feedback_PER： <ul style="list-style-type: none"> <li>• Config.FeedbackScaling.LowerPointIn</li> <li>• Config.FeedbackScaling.UpperPointIn</li> <li>• Config.FeedbackScaling.LowerPointOut</li> <li>• Config.FeedbackScaling.UpperPointOut</li> </ul>
Actuator_H	IN	Bool	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	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
Reset	IN	Bool	重新启动 PID 控制器。默认值：FALSE 如果为 FALSE 至 TRUE 沿： <ul style="list-style-type: none"> <li>• “未激活”操作模式</li> <li>• 输入值 = 0</li> <li>• 复位控制器的中间值。（保留 PID 参数。）</li> </ul> 如果为 TRUE 至 FALSE 沿，则更改为最近激活的模式。
ScaledInput	OUT	Real	标定的过程值
ScaledFeedback	OUT	Real	标定的阀门位置
Output_PER	OUT	Word	模拟量输出值。如果 Config.OutputPerOn = TRUE，则会使用 Output_PER 参数。
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。
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。在 CPU 中，设定值被限制为已组态的实际值绝对上限。

参数和类型		数据类型	说明
SetpointLimitL	OUT	Bool	设定值下限。默认值：FALSE 如果 SetpointLimitL = TRUE，则说明达到设定值的绝对下限。在 CPU 中，设定值被限制为已组态的过程值绝对下限。
InputWarningH	OUT	Bool	如果 InputWarningH = TRUE，则说明输入值已达到或超出警告上限。默认值：FALSE
InputWarningL	OUT	Bool	如果 InputWarningL = TRUE，则说明输入值已达到或超出警告下限。默认值：FALSE
State	OUT	Int	PID 控制器的当前操作模式。默认值：0 使用 Retain.Mode 更改操作模式： <ul style="list-style-type: none"> <li>• State = 0: 未激活</li> <li>• State = 1: 预调节</li> <li>• State = 2: 手动精确调节</li> <li>• State = 3: 自动模式</li> <li>• State = 4: 手动模式</li> <li>• State = 5: 替换输出值逼近</li> <li>• State = 6: 切换时间测量</li> <li>• State = 7: 通过错误监视替换输出值逼近</li> <li>• State = 8: 错误监视</li> </ul>
Error	OUT	Bool	如果 Error = TRUE，则至少一条错误消息未决。默认值：FALSE
ErrorBits	OUT	DWord	PID_3STEP 指令 ErrorBits 参数表 (页 391)定义错误消息。默认值：DW#16#0000 (无错误)

9.2 PID 控制

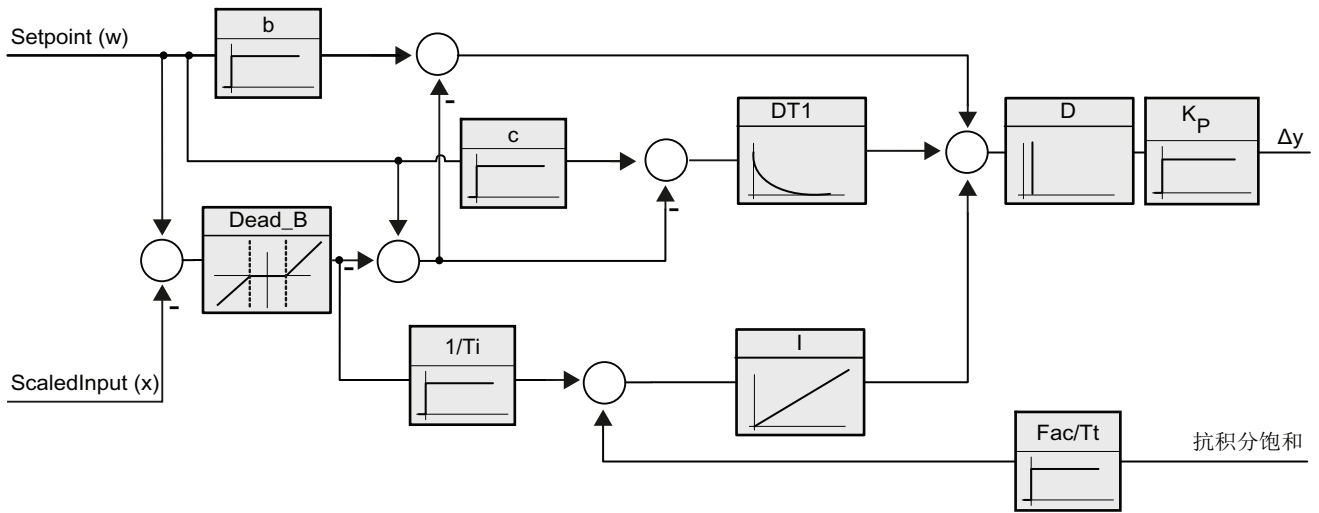


图 9-3 PID\_3Step 控制器作为具有抗积分饱和功能的 PIDT1 控制器时的操作



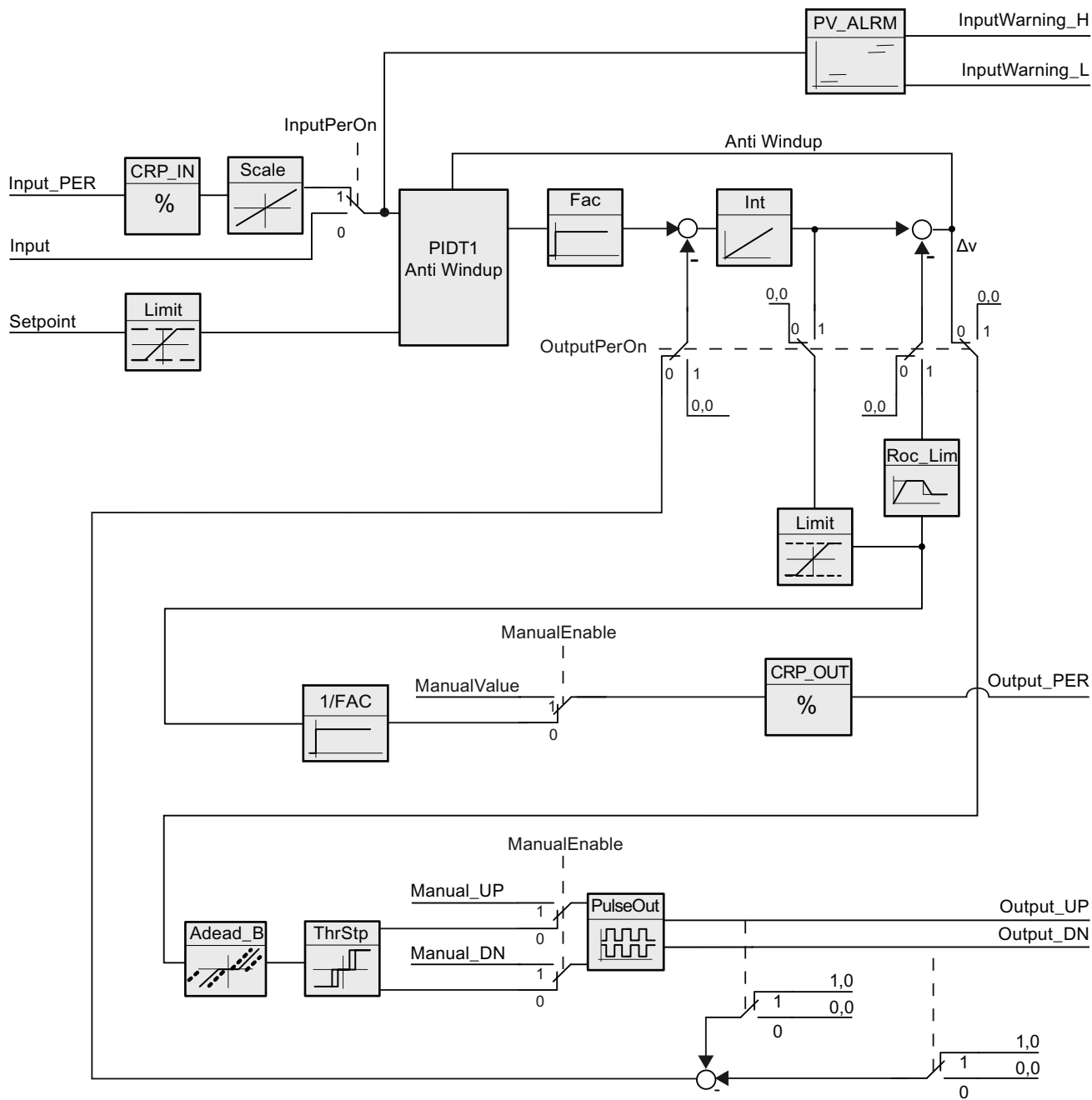


图 9-4 无位置反馈的 PID\_3Step 控制器的操作

9.2 PID 控制

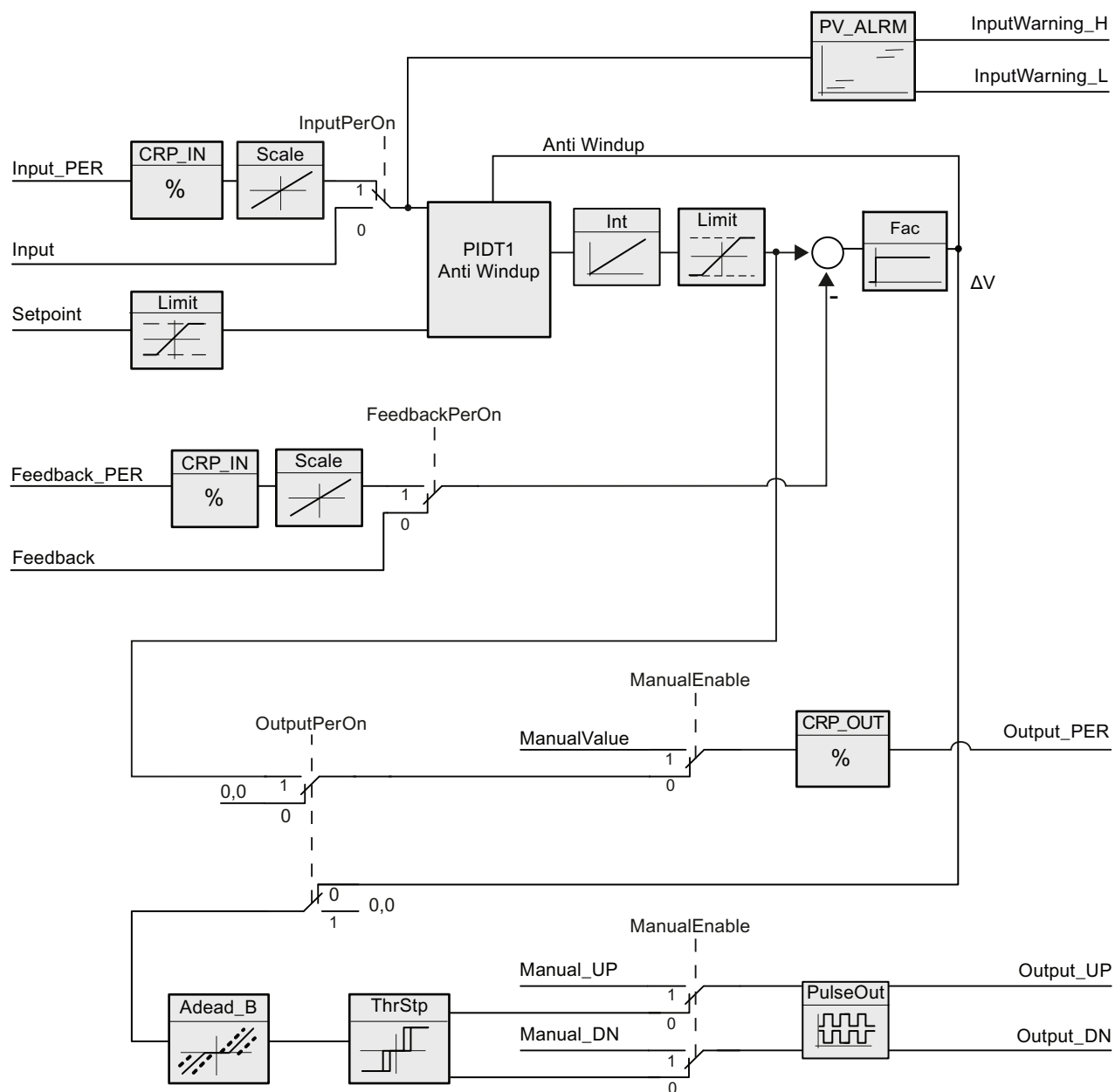


图 9-5 启用了位置反馈的 PID\_3Step 控制器的操作

### 9.2.5 PID\_3STEP 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 9- 18 PID\_3STEP 指令 ErrorBit 参数

ErrorBit (DW#16#...)	说明
0000	无错误
0001	<p>“Input”参数超出过程值限值：  <math>Input &gt; Config.InputUpperLimit</math> 或者  <math>Input &lt; Config.InputLowerLimit</math></p> <p>如果 <math>ActivateRecoverMode = TRUE</math> 并且 <math>ErrorBehaviour = 1</math>，执行器会移动至替换输出值。如果 <math>ActivateRecoverMode = TRUE</math> 并且 <math>ErrorBehaviour = 0</math>，执行器将在其当前位置停止。如果 <math>ActivateRecoverMode = FALSE</math>，执行器将在其当前位置停止。</p> <p>PID_3STEP V1.1：可在手动模式下移动执行器。            PID_3STEP V1.0：在该状态下不能使用手动模式。在消除错误之前不能再次启动执行器。</p>
0002	<p>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。</p> <p>如果在出错之前自动模式处于活动状态，<math>ActivateRecoverMode = TRUE</math> 并且错误不再为未决错误，PID_3STEP 会切换回自动模式。</p>
0004	精确调节时出错，无法保持过程值振荡。
0008	开始预调节时出错。过程值过于接近设定值。开始精确调节。
0010	在精确调节期间不能更改设定值。
0020	在自动模式下或精确调节期间可能无法执行预调节。
0040	精确调节时出错，设定值过于接近设定值限值。
0080	<p>在预调节时出错。输出值限值的组态不正确。</p> <p>进行检查，确定是否正确配置输出值的限值，并匹配控件工作时采用的方向。</p>
0100	精确调节期间出错导致参数无效。

ErrorBit (DW#16#...)	说明
0200	<p>参数“Input”的值无效：值的数字格式无效。</p> <p>如果在出错之前自动模式处于活动状态，<b>ActivateRecoverMode = TRUE</b> 并且错误不再为未决错误，PID_3STEP 会切换回自动模式。</p>
0400	<p>计算输出值失败。检查 PID 参数。</p>
0800	<p>采样时间错误：在循环中断 OB 的采样时间内未调用 PID_3STEP。</p> <p>如果在出错之前自动模式处于活动状态，<b>ActivateRecoverMode = TRUE</b> 并且错误不再为未决错误，PID_3STEP 会切换回自动模式。</p>
1000	<p>参数“Setpoint”的值无效：值的数字格式无效。</p> <p>如果在出错之前自动模式处于活动状态，<b>ActivateRecoverMode = TRUE</b> 并且错误不再为未决错误，PID_3STEP 会切换回自动模式。</p>
2000	<p>参数 <b>Feedback_PER</b> 的值无效。</p> <p>请检查模拟量输入是否有错误尚未解决。</p> <p>执行器不能移动至替换输出值并且不能从当前位置移动。在该状态下不能使用手动模式。必须禁用位置反馈 (<b>Config. FeedbackOn = FALSE</b>) 以从该状态移动执行器。</p> <p>如果在出错之前自动模式处于活动状态，<b>ActivateRecoverMode = TRUE</b> 并且错误不再为未决错误，PID_3STEP 会切换回自动模式。</p>

ErrorBit (DW#16#...)	说明
4000	<p>参数 Feedback 的值无效。值的数字格式无效。</p> <p>执行器不能移动至替换输出值并且不能从当前位置移动。在该状态下不能使用手动模式。必须禁用位置反馈 (Config.FeedbackOn = FALSE) 以从该状态移动执行器。</p> <p>如果在出错之前自动模式处于活动状态, ActivateRecoverMode = TRUE 并且错误不再为未决错误, PID_3STEP 会切换回自动模式。</p>
8000	<p>数字位置反馈出错。Actuator_H = TRUE 以及 Actuator_L = TRUE。</p> <p>执行器不能移动至替换输出值并且不能从当前位置移动。在该状态下不能使用手动模式。</p> <p>必须禁用“Endstop 信号执行器”(Config.ActuatorEndStopOn = FALSE) 才能从该状态移动执行器。</p> <p>如果在出错之前自动模式处于活动状态, ActivateRecoverMode = TRUE 并且错误不再为未决错误, PID_3STEP 会切换回自动模式。</p>

## 9.2.6 组态 PID 控制器

工艺对象的参数决定 PID 控制器的操作。使用该图标可打开组态编辑器。

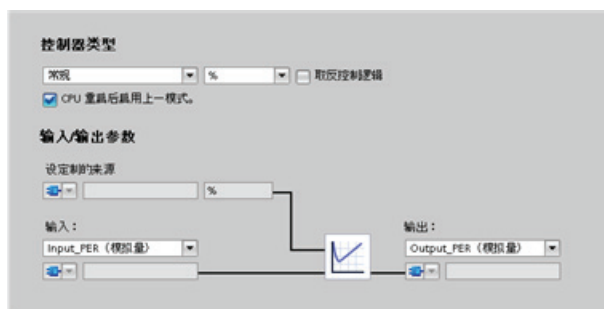


图 9-6 PID\_Compact 的组态编辑器 (基本设置)

9.2 PID 控制

表格 9- 19 PID\_Compact 指令的示例组态设置

设置	说明
基本	<p>控制器类型 (Controller type)</p> <p>选择工程单元。</p> <p>反转控制逻辑 (Invert the control logic)</p> <p>允许选择反作用 PID 回路。</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul> <p>CPU 重启后启用上一模式 (Enable last mode after CPU restart)</p> <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>输入 (Input)</p> <p>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。</p> <p>输出 (Output)</p> <p>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。</p>
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>

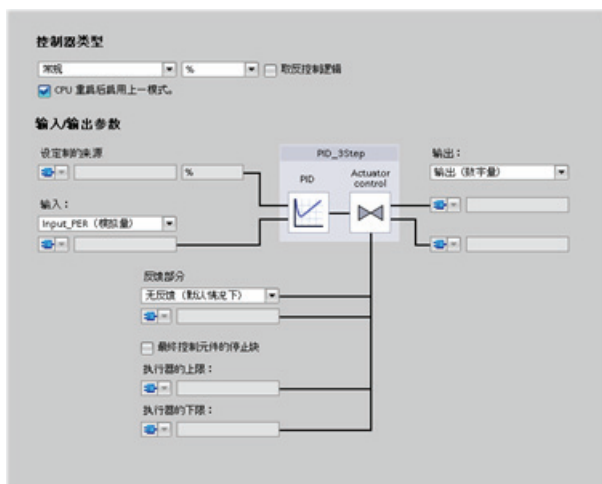


图 9-7 PID\_3Step 的组态编辑器（基本设置）

表格 9-20 PID\_3Step 指令的示例组态设置

设置	说明	
基本	控制器类型 (Controller type)	选择工程单元。
	反转控制逻辑 (Invert the control logic)	允许选择反作用 PID 回路。 <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul>
	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 (Output)	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 (Feedback)	选择返回到 PID 回路的设备状态的类型： <ul style="list-style-type: none"> <li>无反馈（默认）</li> <li>反馈</li> <li>Feedback_PER</li> </ul>
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）。	
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查看该值。）
	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查看该值。）
	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查看该值。）

设置		说明
	出错行为 (Error behavior)	定义检查到错误或复位 PID 回路时的阀门行为。如果选择使用替换位置，请输入“安全位置”(Safety position)。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。对于数字输出，只能选择 0%（关闭）或 100%（打开）。
	刻度位置反馈 <sup>1</sup> (Scale Position Feedback)	<ul style="list-style-type: none"> <li>“停止上限”(High stop) 和“停止下限”(Lower limit stop) 定义最大正向位置（完全打开）和最大反向位置（完全关闭）。“停止上限”(High stop) 必须大于“停止下限”(Lower limit stop)。</li> <li>“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。</li> <li>“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”。</li> </ul>

<sup>1</sup> 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)。


### 9.2.7 调试 PID 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 9- 21 组态画面示例 (PID\_3Step)

	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 调节模式：要调节 PID 循环，请选择“预调节”(Pretuning) 或“微调”(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。</li> </ul> <p>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。</p>
--	--

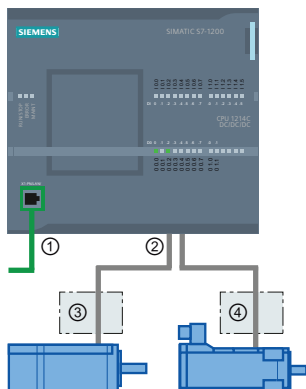
## 9.3 运动控制

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 2 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 4 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。

**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），会从 Q 存储器中移除相应的输出地址（Q0.0 到 Q0.3, Q4.0 到 Q4.3），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。

表格 9- 22 可控制驱动器的最大数目

CPU 型号		未安装任何 SB	带 SB (2 x DC 输出)	带 SB (4 x DC 输出)
CPU 1211C	DC/DC/DC	2	2	2
	AC/DC/RLY	0	1	2
	DC/DC/RLY	0	1	2
CPU 1212C	DC/DC/DC	2	2	2
	AC/DC/RLY	0	1	2
	DC/DC/RLY	0	1	2
CPU 1214C	DC/DC/DC	2	2	2
	AC/DC/RLY	0	1	2
	DC/DC/RLY	0	1	2

CPU 型号		未安装任何 SB	带 SB (2 x DC 输出)	带 SB (4 x DC 输出)
CPU 1215C	DC/DC/DC	4	4	4
	AC/DC/RLY	0	1	2
	DC/DC/RLY	0	1	2

表格 9-23 脉冲输出的频率范围

脉冲输出	频率
板载	2 PTO: $2 \text{ Hz} \leq f \leq 100 \text{ KHz}$
	2 PTO: $2 \text{ Hz} \leq f \leq 20 \text{ KHz}$
标准 SB	$2 \text{ Hz} \leq f \leq 20 \text{ KHz}$
高速 (200 KHz) SB	MC V2 指令: $2 \text{ Hz} \leq f \leq 200 \text{ KHz}$ MC V1 指令: $2 \text{ Hz} \leq f \leq 100 \text{ KHz}$ <sup>1</sup>

<sup>1</sup> MC V1 指令支持的最大频率为 100 KHz。

<b>注意</b>
-----------

脉冲输出发生器的最大脉冲频率对于 CPU 的数字量输出为 100 KHz，而对于标准 SB 的数字量输出为 20 KHz，对于高速 SB 的数字量输出为 200 KHz（或对于 MC V1 指令为 100 KHz）。
--

### 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”(Technological 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。请注意为脉冲和方向分配的两个 Q 输出。
- 

**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。“Pulse\_1”和“Pulse\_3”是信号板上提供的两个唯一脉冲输出。

---

- 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 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”。

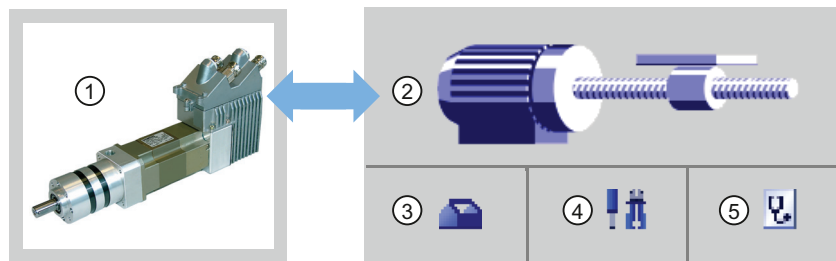
---

**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。

**9.3.1****组态轴**

STEP 7 为“轴”工艺对象提供了组态工具、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**

PTO 需要使用高速计数器 (HSC) 的内部功能。也就是说，在其它地方无法使用相应的高速计数器。

PTO 和 HSC 间的分配是固定的。激活 PTO1 时，它将与 HSC1 连接。如果激活 PTO2，则它将与 HSC2 连接。这仅适用于 S7-1200 V1.0、V2.0、V2.1 和 V2.2 CPU。S7-1200 V3.0 CPU 没有此限制。

生成脉冲时，无法监视当前值（例如，在 ID 1000 中）。

表格 9- 24 用于运动控制的 STEP 7 工具

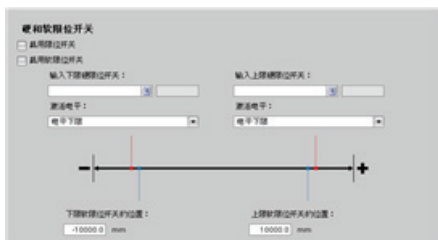
工具	说明
组态	<p>组态“轴”工艺对象的下列属性：</p> <ul style="list-style-type: none"> <li>• 要用的 PTO 的选择以及驱动器接口的组态</li> <li>• 机械的属性和驱动器（机器或系统）的传动比参数</li> <li>• 位置限制属性、动态属性和回原点属性</li> </ul> <p>在工艺对象的数据块中保存组态数据。</p>
调试	<p>无需创建用户程序即可测试轴的功能。启动该工具时，将显示控制面板。控制面板上提供了下列命令：</p> <ul style="list-style-type: none"> <li>• 启用和禁用轴</li> <li>• 在点动模式下移动轴</li> <li>• 以绝对和相对方式定位轴</li> <li>• 使轴回原点</li> <li>• 确认错误信息</li> </ul> <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>



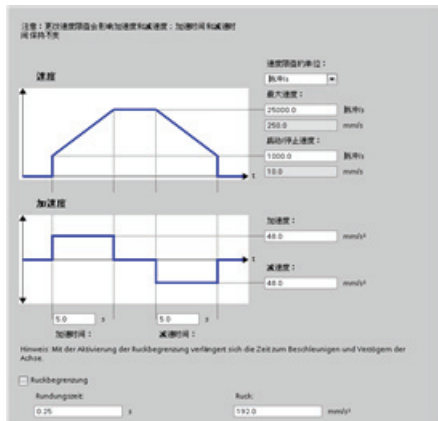
为轴创建工艺对象之后，通过定义基本参数（如 PTO 和驱动器接口的组态）来组态该轴。还可以组态轴的其它属性，例如位置限制属性、动态属性和回原点属性。

**注意**

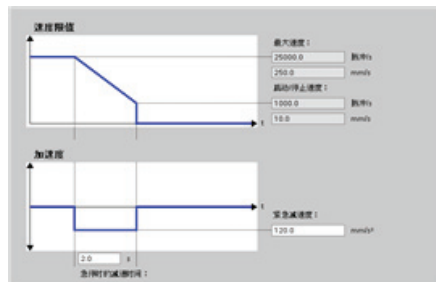
在用户程序中可以根据新量纲单位调整运动控制指令的输入参数的值。



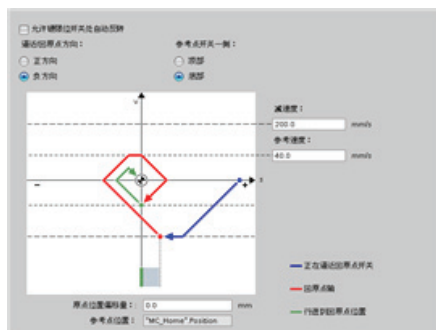
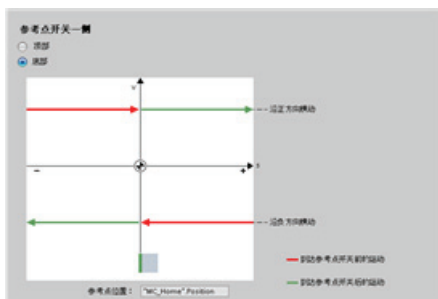
组态驱动器信号、驱动器机械装置和位置监视（硬件和软件限位开关）的属性。



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。

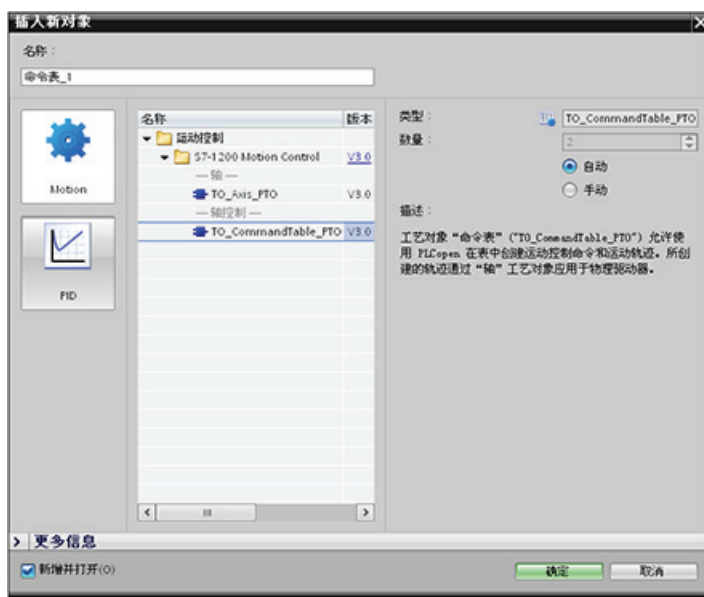
控制面板会显示轴的当前状态。不仅可以启用和禁用轴，还可以测试轴定位（以绝对和相对方式）以及指定速度、加速度和减速度。还可以测试回原点和点动任务。控制面板还可用于确认错误。

### 9.3.2 组态 TO\_CommandTable\_PTO

可以使用工艺对象组态 CommandTable 指令。

#### 添加工艺对象

1. 在项目树中，展开节点“工艺对象”(Technological 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 9- 25 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。



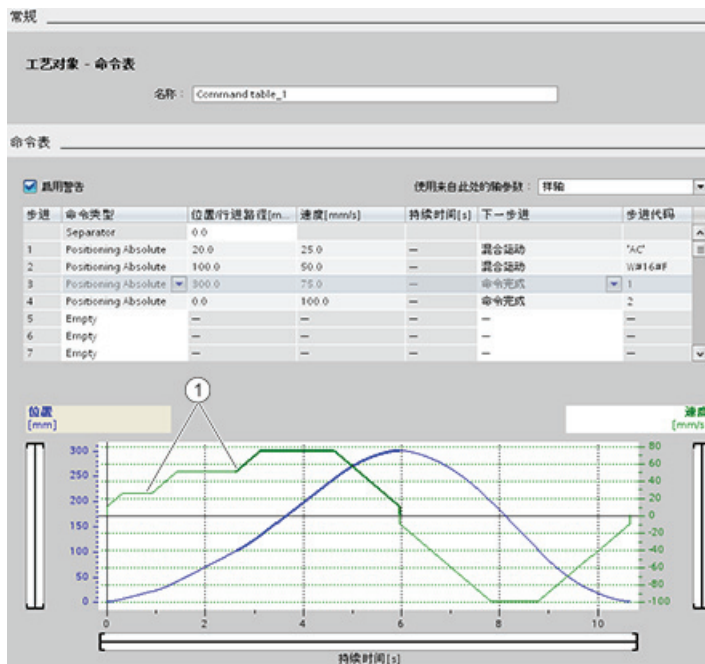
命令类型	说明
Positioning Relative	根据距离定位轴。该命令将按给定的距离和速度移动轴。
Positioning Absolute	根据位置定位轴。该命令以指定的速度将轴移到给定位置。
Velocity setpoint	按给定速度移动轴。
Wait	等待给定期间结束。“Wait”不会停止已激活的行进运动。
Separator	在选定行上方添加“分隔”线。利用分隔线，可在单个命令表中定义多个轨迹。

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。



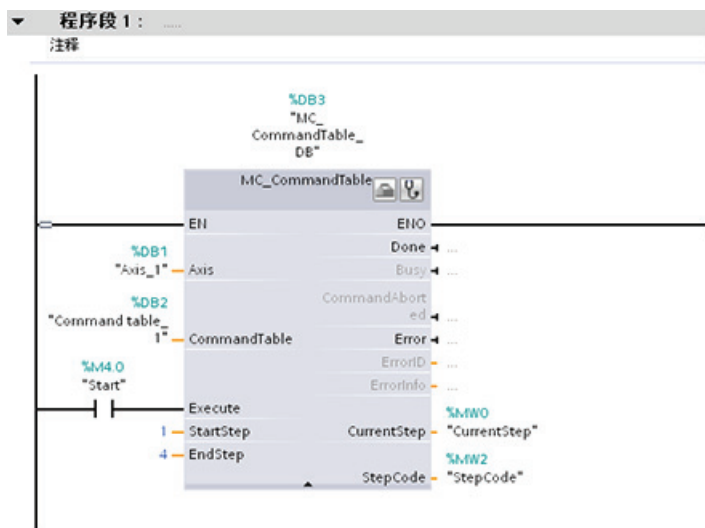
① 轴在两步之间减速到启动/停止速度。

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：



### 9.3.3 运动控制指令

#### 说明

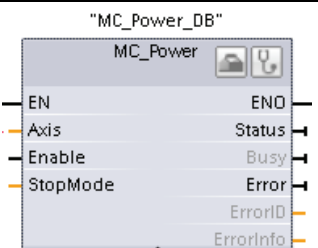
CPU 以 10 ms 为“时间片”或时间段计算运动任务。执行一个时间片时，下一时间片会在队列中等待执行。如果中断某个轴上的运动任务（通过执行该轴的其他新运动任务），可能最多要等待 20 ms（当前时间片的剩余时间加上排队的时间片）才能执行新运动任务。

#### 9.3.3.1 MC\_Power 指令

#### 注意

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。

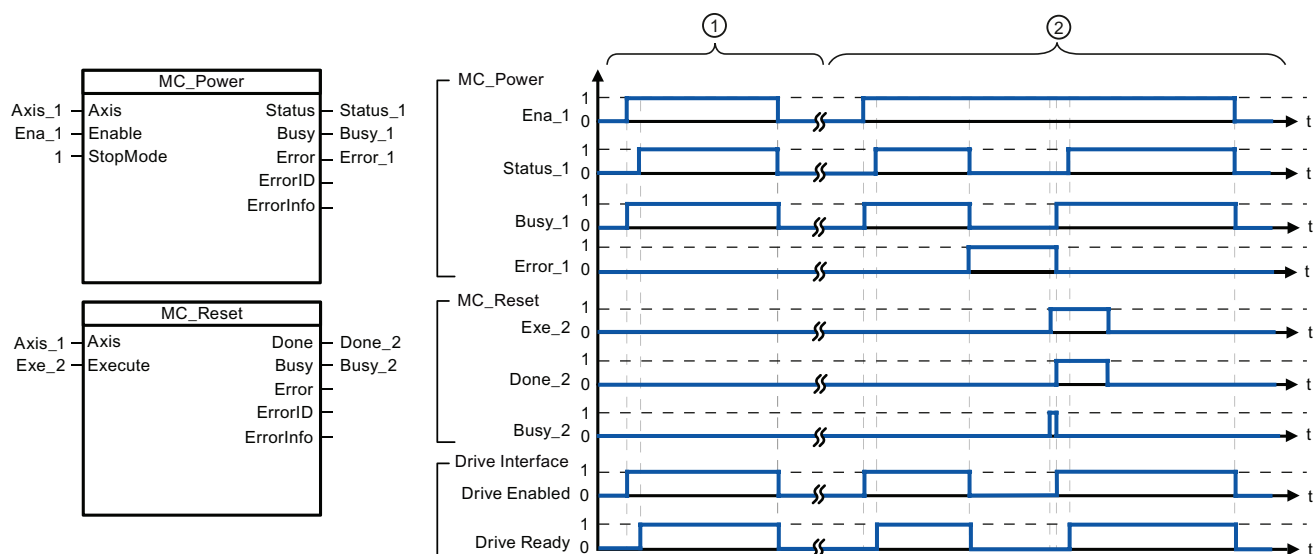
表格 9- 26 MC\_Power 指令

LAD/FBD	SCL	说明
	<pre>"MC_Power_DB" (   Axis:= _multi_fb_in_,   Enable:= _bool_in_,   StopMode:= _int_in_,   Status=&gt; _bool_out_,   Busy=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _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"> <li>• 已正确组态工艺对象。</li> <li>• 没有未决的启用-禁止错误。</li> </ul> <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 9- 27 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"> <li>• 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。</li> <li>• TRUE：运动控制尝试启用轴。</li> </ul>
StopMode	IN	Int	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> <li>• 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。</li> </ul>
Status	OUT	Bool	轴使能的状态： <ul style="list-style-type: none"> <li>• FALSE：轴已禁用：               <ul style="list-style-type: none"> <li>– 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）。</li> <li>– 轴未回原点。</li> <li>– 禁用时，直到轴达到停止状态，状态才会更改为 FALSE。</li> </ul> </li> <li>• TRUE：轴已启用：               <ul style="list-style-type: none"> <li>– 轴已准备好执行运动控制任务。</li> <li>– 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。</li> </ul> </li> </ul>
Busy	OUT	Bool	FALSE: MC_Power 未激活。 TRUE: MC_Power 处于活动状态
Error	OUT	Bool	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。
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。

9.3.3.2 MC\_Reset 指令

表格 9- 28 MC\_Reset 指令

LAD/FBD	SCL	说明
	<pre> "MC_Reset_DB" (   Axis:=_multi_fb_in_,   Execute:=_bool_in_,   Restart:=_bool_in_,   Done=&gt;_bool_out_,   Busy=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_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”。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>

- STEP 7 会在插入指令时自动创建 DB。
- 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 9- 29 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。

### 9.3.3.3 MC\_Home 指令

表格 9-30 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=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; 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。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>

- 1 STEP 7 会在插入指令时自动创建 DB。
- 2 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 9-31 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

参数和类型		数据类型	说明
Position	IN	Real	<ul style="list-style-type: none"> <li>Mode = 0、2 和 3（完成回原点操作后轴的绝对位置）</li> <li>Mode = 1（当前轴位置的校正值）</li> </ul> 限值: $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Mode	IN	Int	回原点模式 <ul style="list-style-type: none"> <li>0: 绝对式直接回原点 新的轴位置为参数“Position”的位置值。</li> <li>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。</li> <li>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。</li> <li>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。</li> </ul>
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：

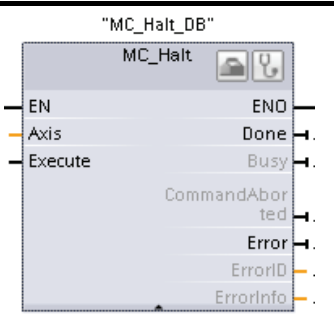
1. 检查上文所述的要求。
2. 使用相应的值初始化必要的输入参数，然后在输入参数“Execute”出现上升沿时开始回原点操作。
3. 如果输出参数“Done”和工艺对象变量 <轴名称>.StatusBits.HomingDone 指示值 TRUE，则回原点操作完成。

表格 9- 32 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	<p>MC_Home 任务可被下列运动控制任务中止：</p> <p>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。</p> <p>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。</p>	
3	<p>MC_Home 任务可被下列运动控制任务中止：</p> <ul style="list-style-type: none"> <li>• MC_Home Mode = 3</li> <li>• MC_Halt</li> <li>• MC_MoveAbsolute</li> <li>• MC_MoveRelative</li> <li>• MC_MoveVelocity</li> <li>• MC_MoveJog</li> </ul>	<p>新 MC_Home 任务可中止下列激活的运动控制任务：</p> <ul style="list-style-type: none"> <li>• MC_Home 模式 = 2、3</li> <li>• MC_Halt</li> <li>• MC_MoveAbsolute</li> <li>• MC_MoveRelative</li> <li>• MC_MoveVelocity</li> <li>• MC_MoveJog</li> </ul>

9.3.3.4 MC\_Halt 指令

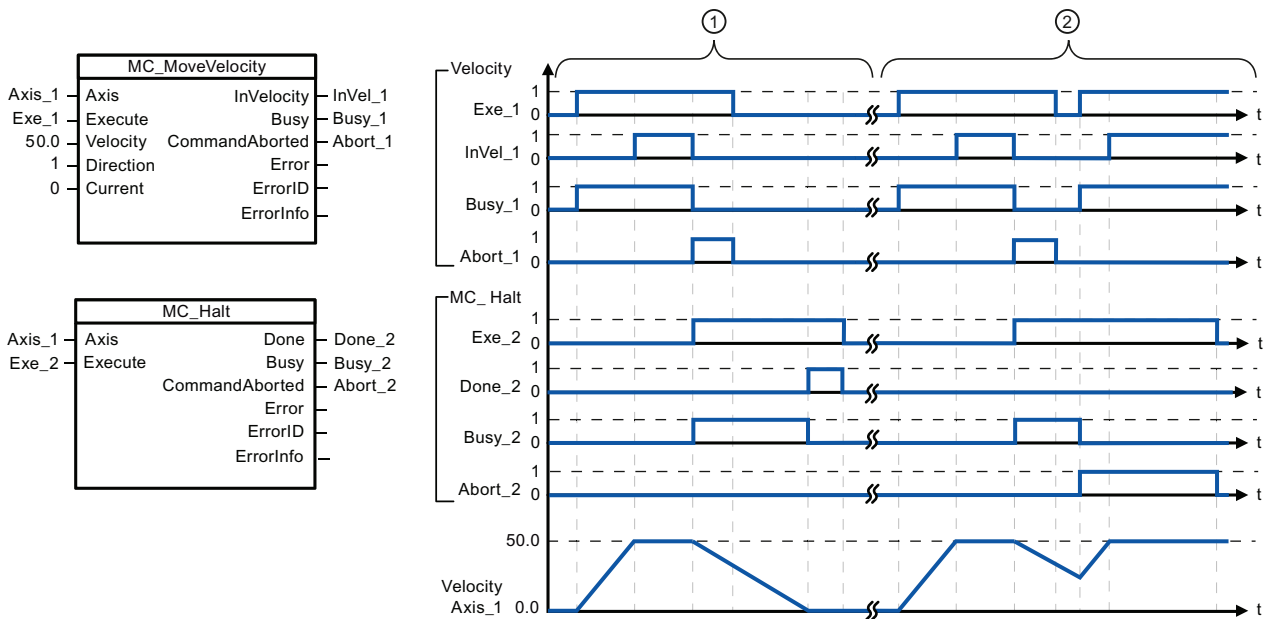
表格 9-33 MC\_Halt 指令

LAD/FBD	SCL	说明
	<pre>"MC_Halt_DB" (   Axis:= _multi_fb_in_,   Execute:= _bool_in_,   Done=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; _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 9-34 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	Bool 出现上升沿时开始任务
Done	OUT	Bool TRUE = 速度达到零
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



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

9.3.3.5 MC\_MoveAbsolute 指令

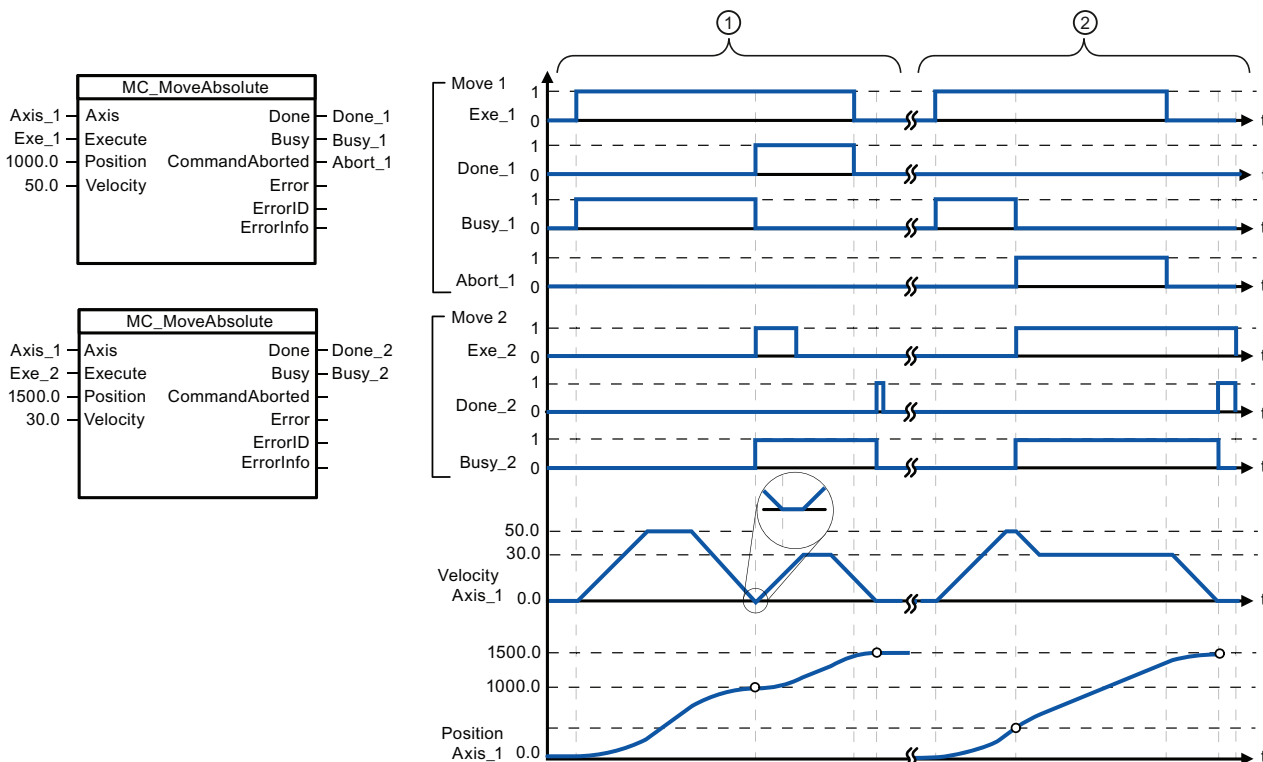
表格 9- 35 MC\_MoveAbsolute 指令

LAD/FBD	SCL	说明
	<pre>"MC_MoveAbsolute_DB" (     Axis:= _multi_fb_in_,     Execute:= _bool_in_,     Position:= _real_in_,     Velocity:= _real_in_,     Done=&gt; _bool_out_,     Busy=&gt; _bool_out_,     CommandAborted=&gt; _bool_out_,     Error=&gt; _bool_out_,     ErrorID=&gt; _word_out_,     ErrorInfo=&gt; _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 9- 36 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） 限值： $-1.0e^{12} \leq \text{Position} \leq 1.0e^{12}$
Velocity	IN	Real 轴的速度（默认值： 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）
ErrorInfo	OUT	Word 参数“ErrorID”的错误信息（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`

9.3.3.6 MC\_MoveRelative 指令

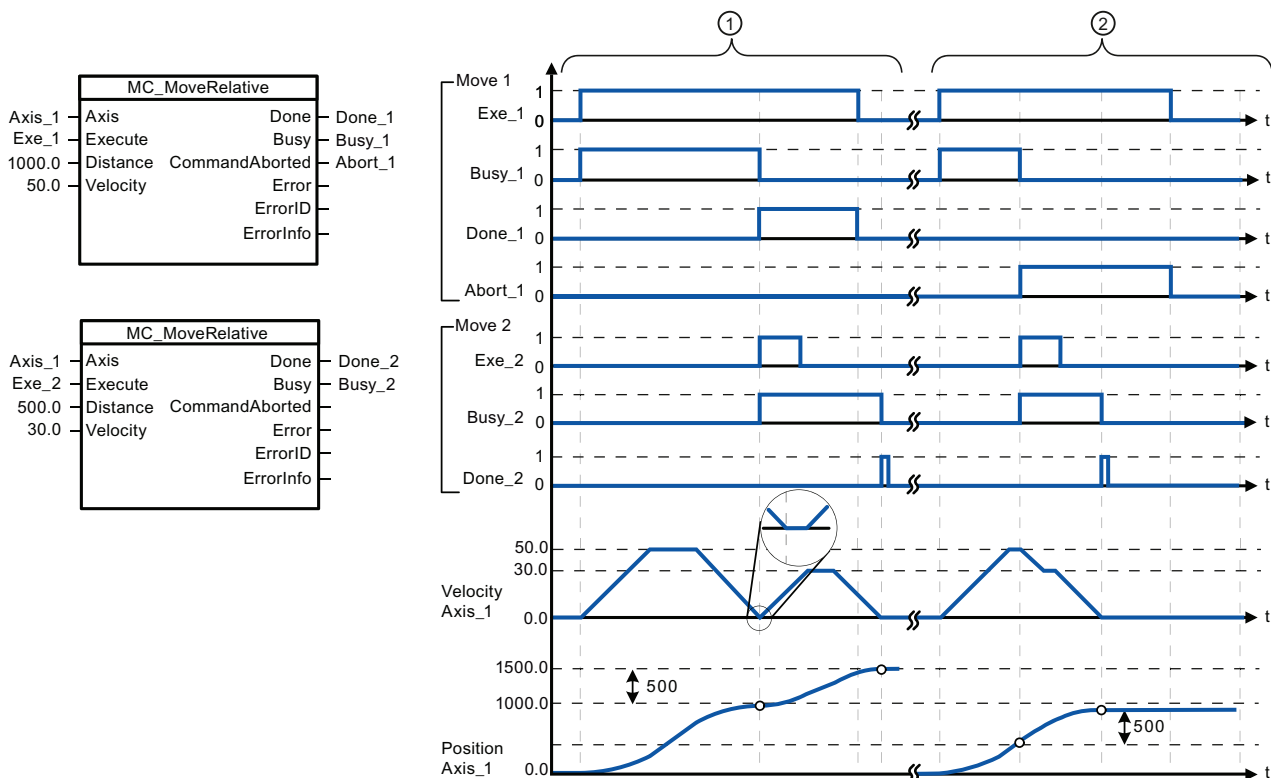
表格 9- 37 MC\_MoveRelative 指令

LAD/FBD	SCL	说明
	<pre>"MC_MoveRelative_DB" (     Axis:= _multi_fb_in_,     Execute:= _bool_in_,     Distance:= _real_in_,     Velocity:= _real_in_,     Done=&gt; _bool_out_,     Busy=&gt; _bool_out_,     CommandAborted=&gt; _bool_out_,     Error=&gt; _bool_out_,     ErrorID=&gt; _word_out_,     ErrorInfo=&gt; _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 9- 38 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） 限值： $-1.0e^{12} \leq \text{Distance} \leq 1.0e^{12}$
Velocity	IN	Real 轴的速度（默认值： 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）
ErrorInfo	OUT	Word 参数“ErrorID”的错误信息（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\_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

9.3.3.7 MC\_MoveVelocity 指令

表格 9- 39 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=&gt; _bool_out_,     Busy=&gt; _bool_out_,     CommandAborted=&gt; _bool_out_,     Error=&gt; _bool_out_,     ErrorID=&gt; _word_out_,     ErrorInfo=&gt; 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>

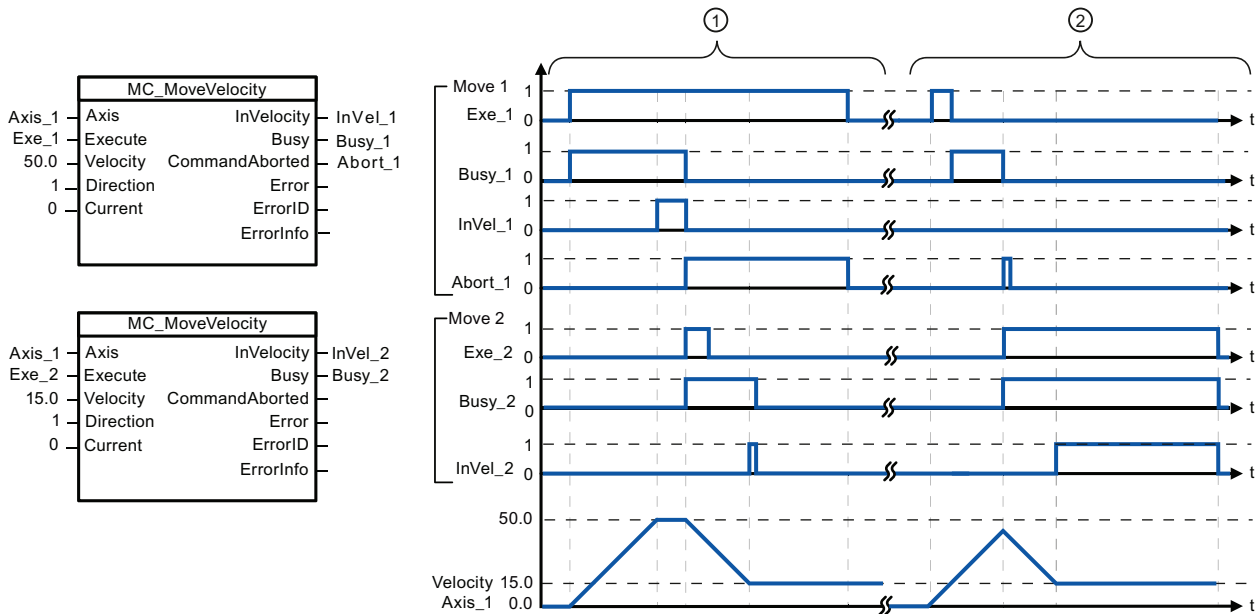
- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC\_MoveVelocity\_DB”是背景 DB 的名称。

表格 9- 40 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） 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"> <li>0: 旋转方向与参数“Velocity”中的值符号一致（默认值）</li> <li>1: 正旋转方向（参数“Velocity”的值符号被忽略。）</li> <li>2: 负旋转方向（参数“Velocity”的值符号被忽略。）</li> </ul>



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"> <li>• FALSE：禁用“保持当前速度”。使用参数“Velocity”和“Direction”的值。（默认值）</li> <li>• TRUE：激活“保持当前速度”。不考虑参数“Velocity”和“Direction”的值。</li> </ul> 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"> <li>• 如果“Current”= FALSE：已达到参数“Velocity”中指定的速度。</li> <li>• 如果 "Current" = TRUE：轴在启动时以当前速度运动。</li> </ul>
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）
ErrorInfo	OUT	Word	参数“ErrorID”的错误信息（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 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

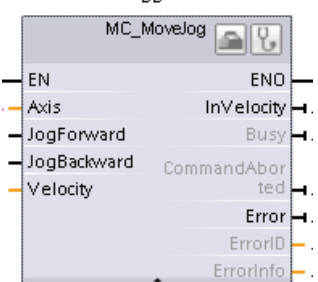
“Velocity”= 0.0 的 MC\_MoveVelocity 任务（如 MC\_Halt 任务）可中止激活的运动任务并利用组态的减速度停止轴运动。轴停止运动后，输出参数“InVelocity”将指示 TRUE 并至少持续一个程序循环的时间。

“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。

## 9.3.3.8 MC\_MoveJog 指令

表格 9- 41 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=&gt; _bool_out_,   Busy=&gt; _bool_out_,   CommandAborted=&gt; _bool_out_,   Error=&gt; _bool_out_,   ErrorID=&gt; _word_out_,   ErrorInfo=&gt; 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。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>

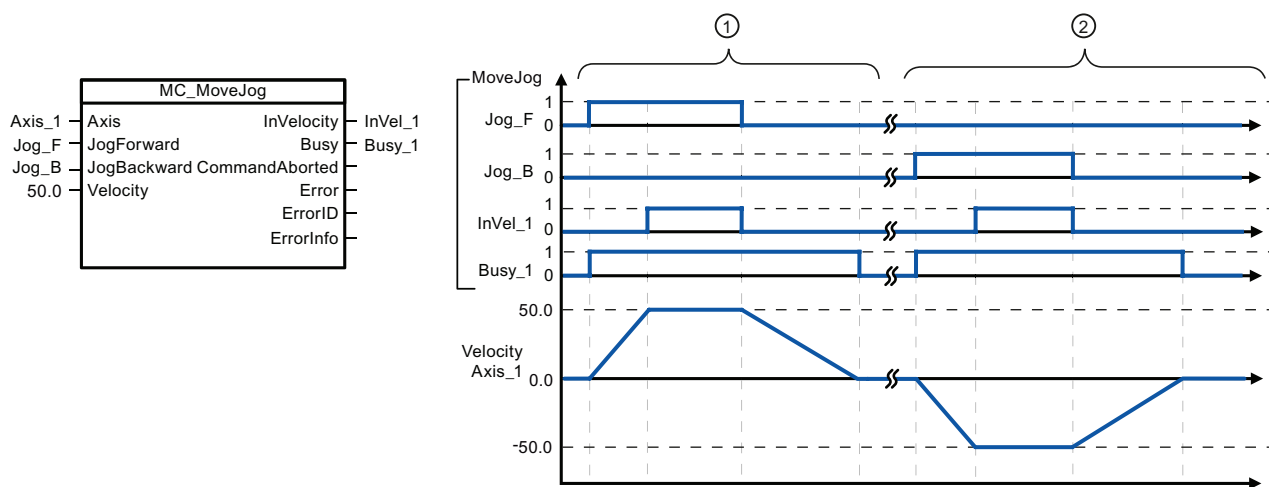
- STEP 7 会在插入指令时自动创建 DB。
- 在 SCL 示例中，“MC\_MoveJog\_DB”是背景 DB 的名称。

表格 9- 42 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 <sup>1</sup>	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 <sup>1</sup>	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） 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）
ErrorInfo	OUT	Word	参数“ErrorID”的错误信息（默认值：0000）

<sup>1</sup> 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

9.3.3.9 MC\_CommandTable 指令

表格 9- 43 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=&gt;_bool_out_,   Busy=&gt;_bool_out_,   CommandAborted=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_,   CurrentIndex=&gt;_uint_out_,   Code=&gt;_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>

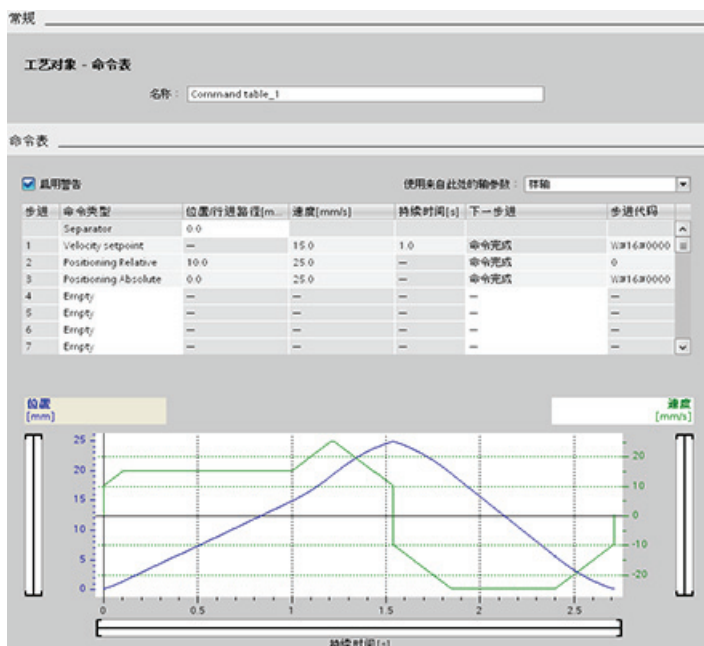
- 1 STEP 7 会在插入指令时自动创建 DB。
- 2 在 SCL 示例中，“MC\_CommandTable\_DB”是背景 DB 的名称。

表格 9- 44 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

参数和类型		数据类型	初始值	说明
Execute	IN	Booll	FALSE	使用上升沿启动作业
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 9- 45 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：

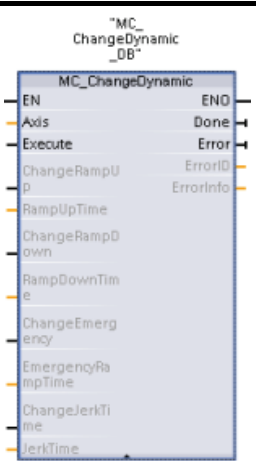
- MC\_Home Mode = 3
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative
- MC\_MoveVelocity
- MC\_MoveJog
- MC\_CommandTable

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

9.3.3.10 MC\_ChangeDynamic

表格 9- 46 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=&gt;_bool_out_,   Error=&gt;_bool_out_,   ErrorID=&gt;_word_out_,   ErrorInfo=&gt;_word_out_);         </pre>	<p>更改运动控制轴的动态设置:</p> <ul style="list-style-type: none"> <li>更改加速时间（加速度）值</li> <li>更改减速时间（减速度）值</li> <li>更改急停减速时间（急停减速度）值</li> <li>更改平滑时间（冲击）值</li> </ul>

- 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 9- 47 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 = 根据输入参数“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。

**9.3.4 S7-1200 的运动控制操作**

**9.3.4.1 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。

脉冲输出和方向输出彼此互相分配的关系保持不变。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。

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。

表格 9- 48 脉冲和方向输出的地址分配

使用运动控制的输出		
	脉冲	方向
PTO 0		
内置 I/O	Q0.0	Q0.1
SB I/O	Q4.0	Q4.1
PTO 1		
内置 I/O	Q0.2	Q0.3
SB I/O	Q4.2	Q4.3

使用运动控制的输出		
PTO 2		
内置 I/O	Q0.4 <sup>1</sup>	Q0.5 <sup>1</sup>
SB I/O	Q4.0	Q4.1
PTO 3		
内置 I/O	Q0.6 <sup>2</sup>	Q0.7 <sup>2</sup>
SB I/O	Q4.2	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 以及 CPU 1215C 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

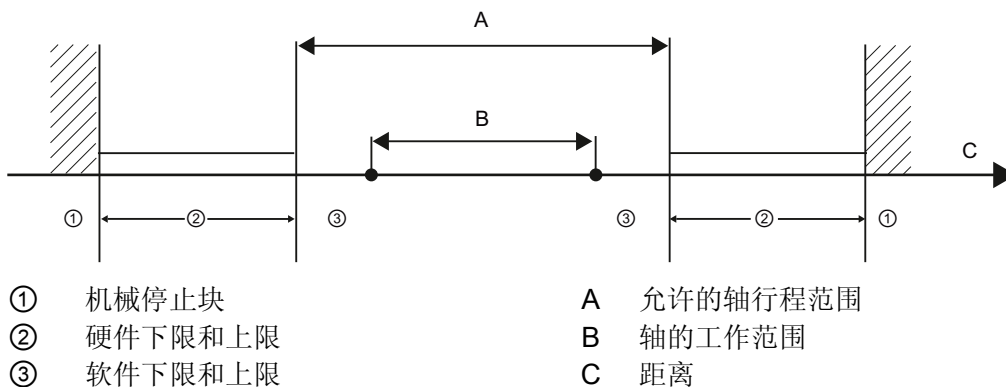
如果已选择 PTO (Pulse Train Output) 并将其分配给某个轴，固件将通过相应的脉冲和方向输出接管控制。

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。

9.3.4.2 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”。



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

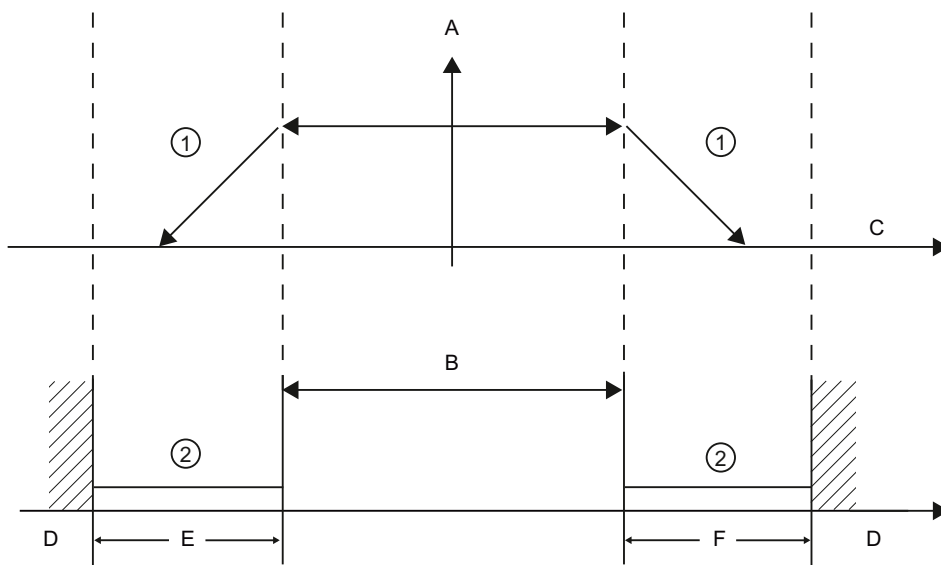
硬件限位开关确定轴的最大行程范围。硬件限位开关是物理开关元件，必须与 CPU 中具有中断功能的输入相连接。仅使用逼近后始终保持切换的硬件限位开关。只有在返回到允许的行程范围后，该切换状态才会发生改变。

表格 9- 49 脉冲发生器的可用输入

说明	RPS	LIM-	LIM+
PTO 0			
内置 I/O		10.0 - 11.5	
SB I/O		14.0 - 14.3	
PTO 1			
内置 I/O		10.0 - 11.5	
SB I/O		14.0 - 14.3	
PTO 2			
内置 I/O		10.0 - 11.5	
SB I/O		14.0 - 14.3	
PTO 3			

说明	RPS	LIM-	LIM+
内置 I/O		I0.0 - I1.5	
SB I/O		I4.0 - I4.3	

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**

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。

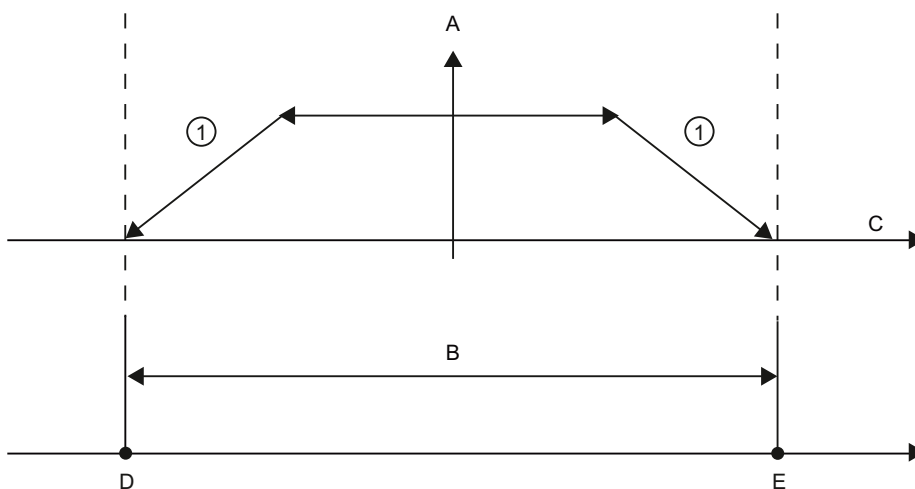
滤波时间的这种更改会引发意外的机械或过程操作，这可能会导致人员死亡、重伤和/或设备损坏。

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/PositonLimits\_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/Position Limits\_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）。

### 9.3.4.3

#### 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。

### 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。

#### 注意

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 9-50 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。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。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"> <li>• 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。</li> <li>• 被动回原点：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"> <li>- “逼近方向”组态</li> <li>- “参考点开关”组态</li> <li>- 被动回原点期间的当前行进方向</li> </ul> </li> </ul>
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>

9.3 运动控制

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 9-51 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

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 9- 52 MC 回原点的速度特性曲线

操作		备注	
		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**

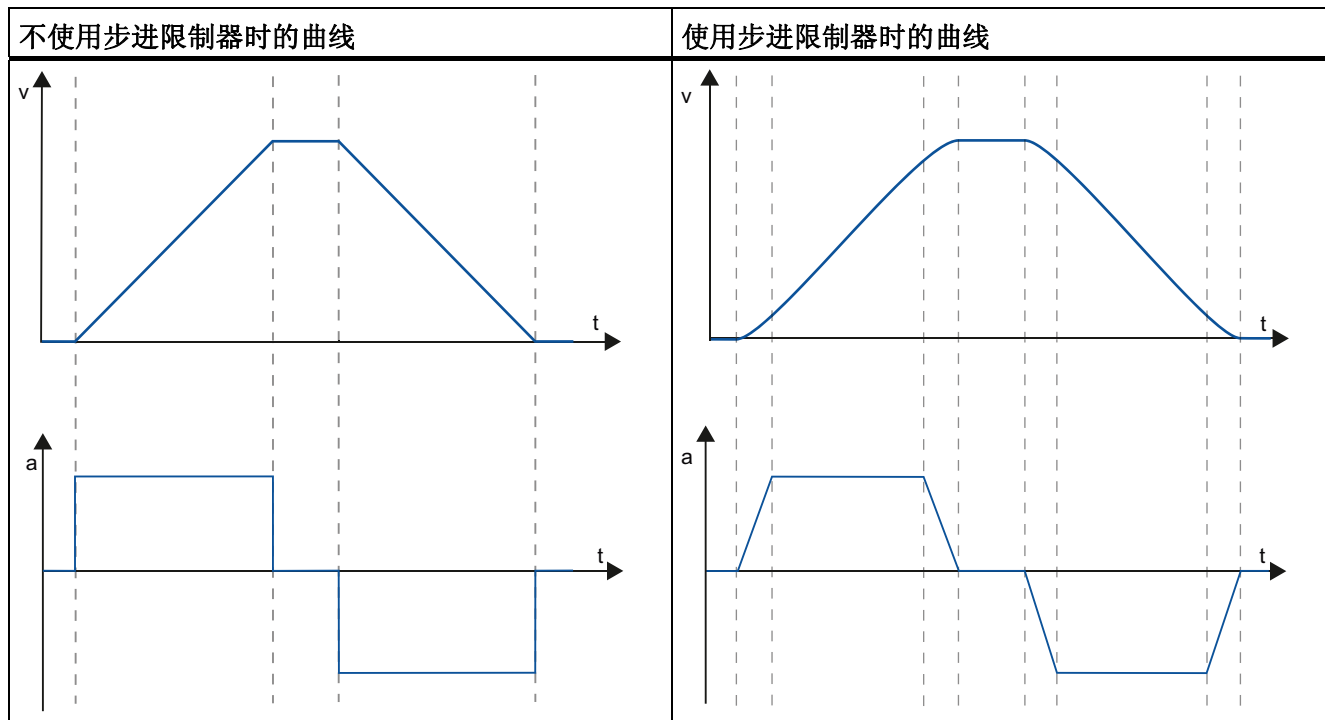
如果回原点搜索没有按照预期的那样运行，请检查分配给硬件限位或参考点的输入。可能已经在设备配置中禁用了这些输入的沿中断。

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。

9.3.4.4 冲击限制

利用冲击限制可在加速和减速斜坡期间减小机械装置上的应力。步进限制器处于激活状态时，加速度和减速度的值不会突然发生变化；该值会在转换阶段进行调整。下图显示了不使用冲击限制和使用冲击限制时的速度和加速度曲线。

表格 9-53 冲击限制



冲击限制使轴运动的速度曲线变得“平滑”。例如，这可以确保传送带实现软启动和软制动。

### 9.3.5 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 9- 54 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"> <li>主动回原点期间，该状态为 FALSE。</li> <li>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。</li> </ul> (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 9- 55 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)

9.3 运动控制

表格 9- 56 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 9- 57 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 9- 58 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 和 HSC 已使用	另一个轴正在使用相同的 PTO 和 HSC 并且该轴已启用“MC_Power”。 (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 9- 59 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 9- 60 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)

### 9.3.6 监视激活的命令

#### 9.3.6.1 监视具有输出参数“Done”的 MC 指令

具有输出参数“Done”的运动控制指令通过输入参数“Execute”启动，并且具有明确的结论（例如，对于运动控制指令“MC\_Home”：回原点已成功）。任务完成后，轴处于停止状态。

- 如果任务已成功完成，则输出参数“Done”的值为 TRUE。
- 输出参数“Busy”、“CommandAborted”和“Error”发出信号，指示任务仍在处理、已中止或有未决的错误。运动控制指令“MC\_Reset”无法中止，所以没有输出参数“CommandAborted”。
  - 在运动控制任务处理期间，输出参数“Busy”的值为 TRUE。如果任务已完成、中止或因错误停止，则输出参数“Busy”的值将变为 FALSE。无论输入参数“Execute”的信号状态是什么，都会发生这种变化。
  - 输出参数“Done”、“CommandAborted”和“Error”的值至少在一个周期内都为 TRUE。当输入参数“Execute”设置为 TRUE 时，将锁存这些状态消息。

以下运动控制指令的任务具有明确的结论：

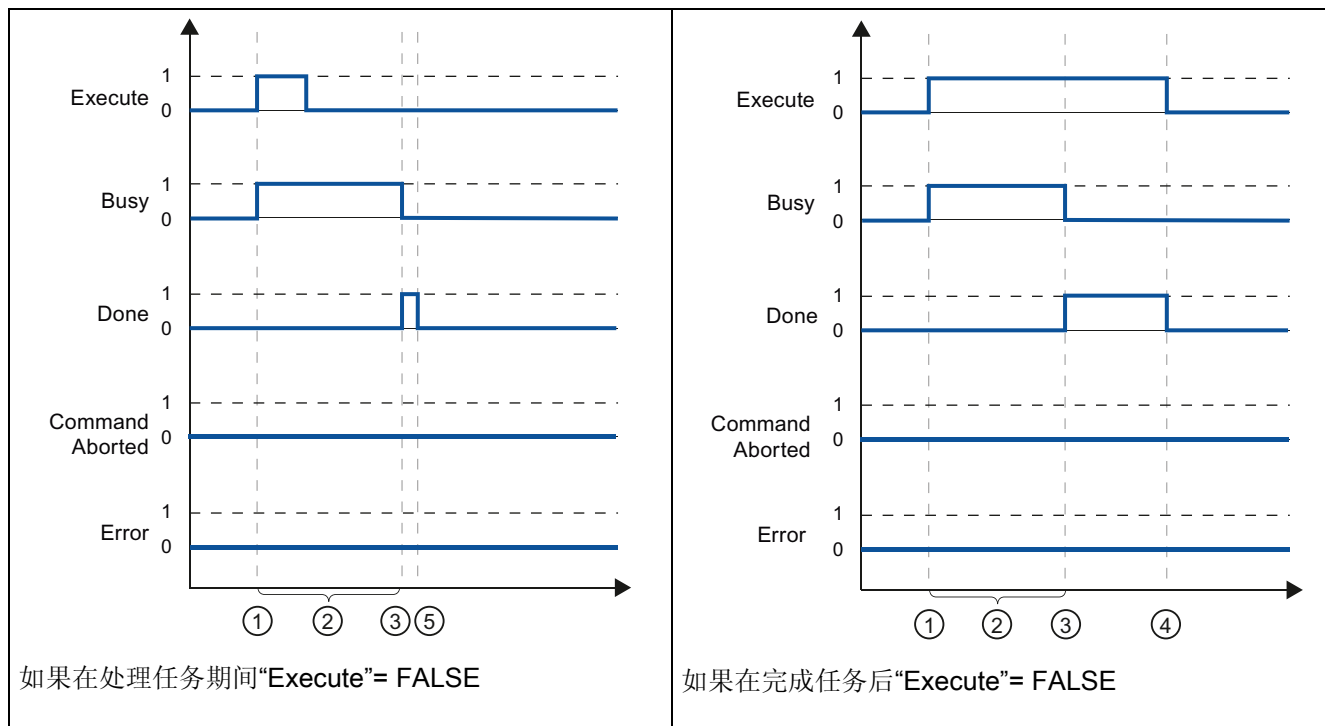
- MC\_Reset
- MC\_Home
- MC\_Halt
- MC\_MoveAbsolute
- MC\_MoveRelative



下图针对各种示例情况显示了状态位的操作。

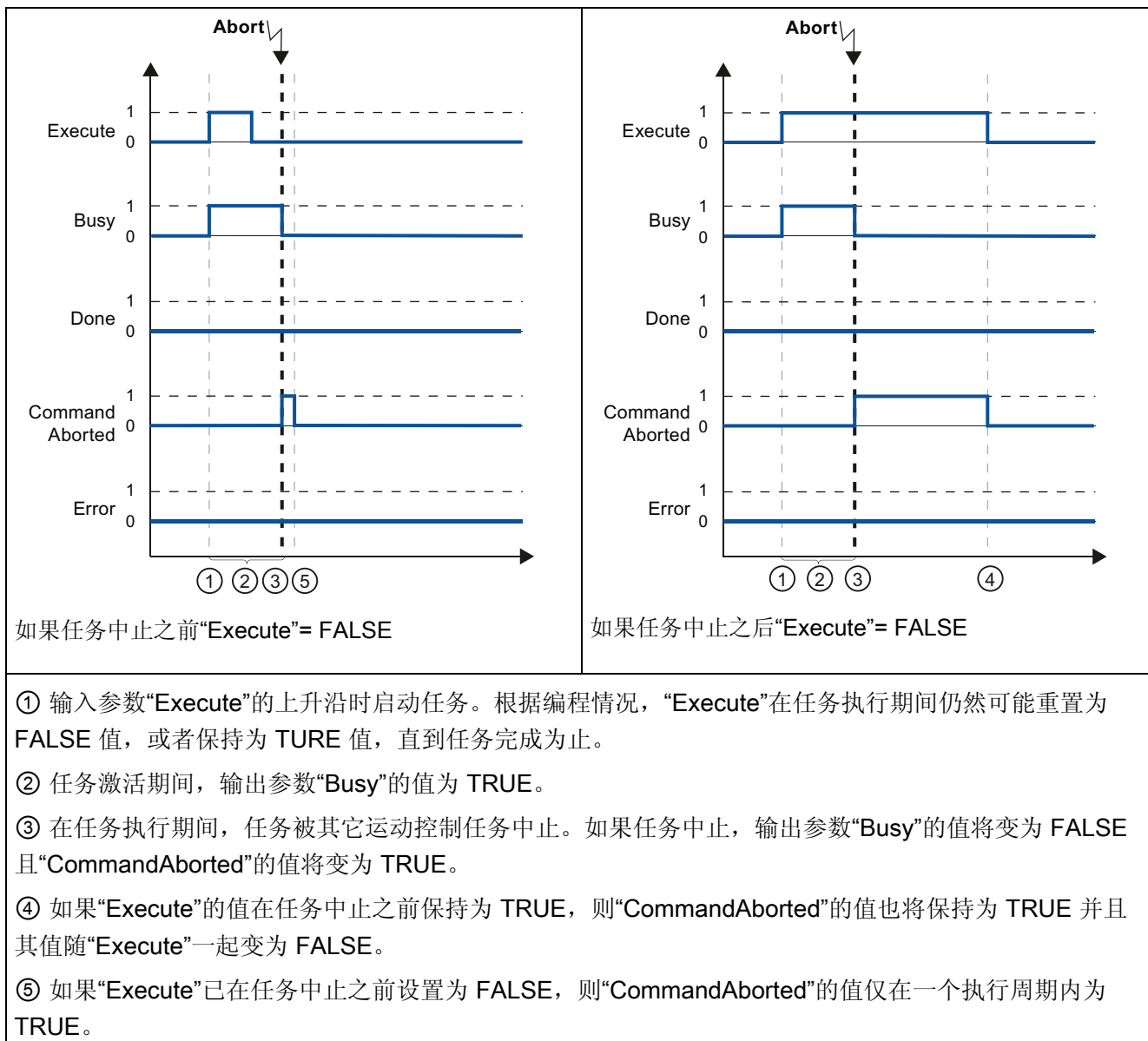
- 第一个示例显示了已完成的任务的轴行为。如果运动控制任务已在对其下结论前完全执行，则通过输出参数“Done”的 TURE 值对此进行指示。输入参数“Execute”的信号状态影响输出参数“Done”中的显示持续时间。
- 第二个示例显示了已中止的任务的轴行为。如果运动控制任务在执行期间中止，则通过输出参数“CommandAborted”的 TURE 值对此进行指示。输入参数“Execute”的信号状态影响输出参数“CommandAborted”中的显示持续时间。
- 第三个示例显示了出现错误时的轴行为。如果在运动控制任务执行期间出错，则通过输出参数“Error”的 TURE 值对此进行指示。输入参数“Execute”的信号状态影响输出参数“Error”中的显示持续时间。

表格 9- 61 示例 1 - 任务完成执行

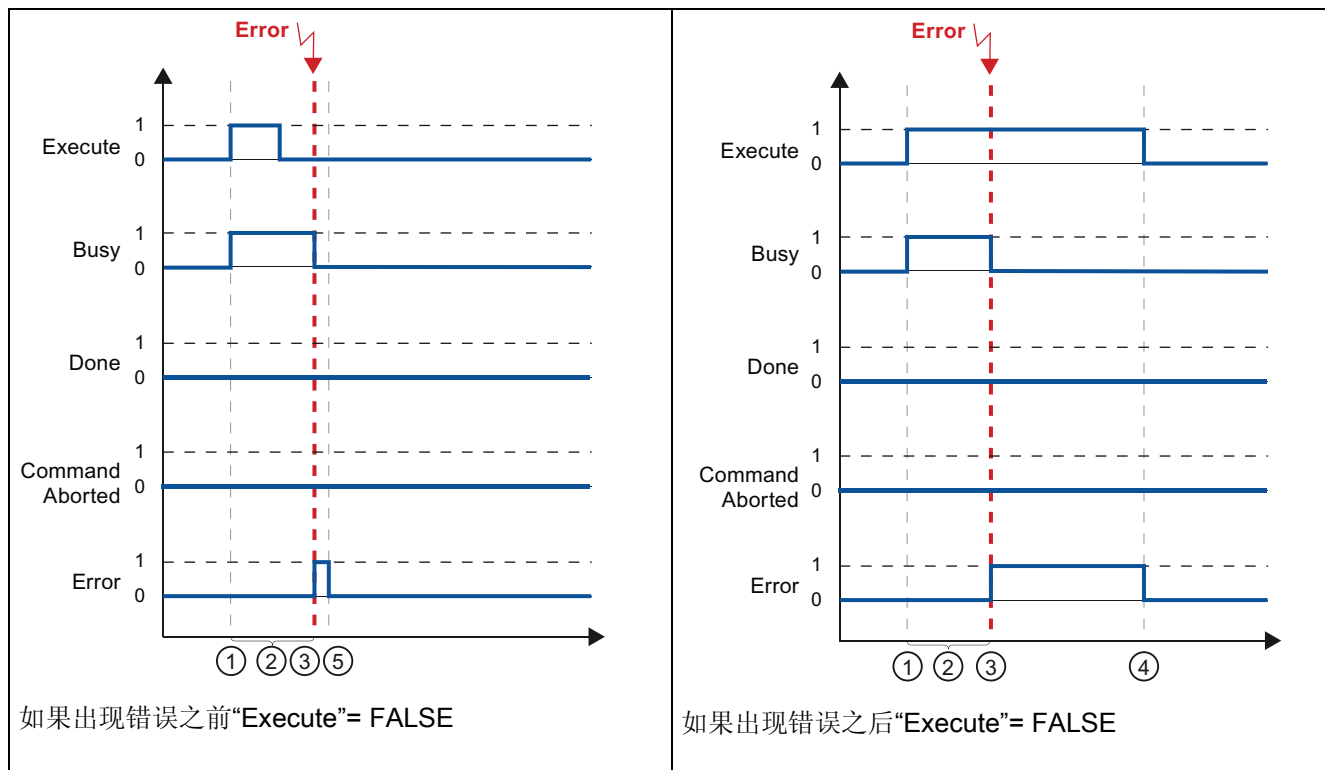


- ① 输入参数“Execute”的上升沿时启动任务。根据编程情况，“Execute”在任务执行期间仍然可能重置为 FALSE 值，或者保持为 TRUE 值，直到任务完成为止。
- ② 任务激活期间，输出参数“Busy”的值为 TRUE。
- ③ 任务结束后（例如，对于运动控制指令“MC\_Home”：回原点已成功），输出参数“Busy”的值变为 FALSE，并且“Done”的值变为 TRUE。
- ④ 如果“Execute”的值在任务完成之前保持为 TRUE，则“Done”的值也将保持为 TRUE 并且其值随“Execute”一起变为 FALSE。
- ⑤ 如果“Execute”已在任务完成之前设置为 FALSE，则“Done”的值仅在一个执行周期内为 TRUE。

表格 9- 62 示例 2 - 中止任务



表格 9- 63 示例 3 - 任务执行期间出错



- ① 输入参数“Execute”的上升沿时启动任务。根据编程情况，“Execute”在任务执行期间仍然可能重置为 FALSE 值，或者保持为 TRUE 值，直到任务完成为止。
- ② 任务激活期间，输出参数“Busy”的值为 TRUE。
- ③ 任务执行期间出错。出错时，输出参数“Busy”的值将变为 FALSE 且“Error”的值将变为 TRUE。
- ④ 如果“Execute”的值在出错之前保持为 TRUE，则“Error”的值也将保持为 TRUE 并且其值仅随“Execute”一起变为 FALSE。
- ⑤ 如果“Execute”已在出错之前设置为 FALSE，则“Error”的值仅在一个执行周期内为 TRUE。

### 9.3.6.2 监视 MC\_Velocity 指令

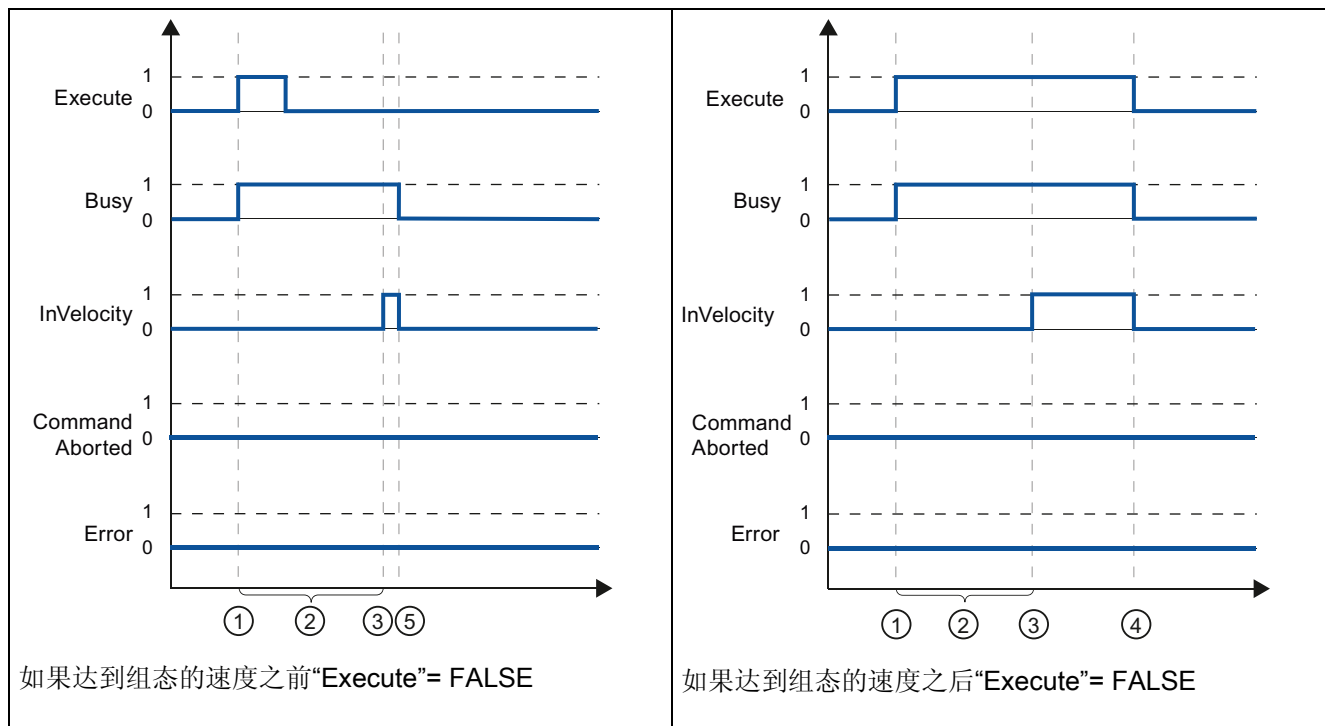
运动控制指令“MC\_MoveVelocity”的任务持续以指定的速度执行。

- 未明确定义运动控制指令“MC\_MoveVelocity”任务的结束。第一次达到设置的速度且轴恒速运转时，就实现了任务目标。如果达到设置的速度，则通过输出参数“InVelocity”的 TURE 值对此进行指示。
- 已达到设置的速度且输入参数“Execute”的值已设置为 FALSE 时，任务完成。然而，在任务完成时轴运动尚未完成。例如，可以使用运动控制任务“MC\_Halt”停止轴运动。
- 输出参数“Busy”、“CommandAborted”和“Error”发出信号，指示任务仍在处理、已中止或有未决的错误。
  - 在运动控制任务执行期间，输出参数“Busy”的值为 TRUE。如果任务已完成、中止或因错误停止，则输出参数“Busy”的值将变为 FALSE。无论输入参数“Execute”的信号状态是什么，都会发生这种变化。
  - 当输出参数“InVelocity”、“CommandAborted”和“Error”的条件满足时，它们的值至少在一个周期内都为 TRUE。当输入参数“Execute”设置为 TRUE 时，将锁存这些状态消息。

下图针对各种示例情况显示了状态位的操作。

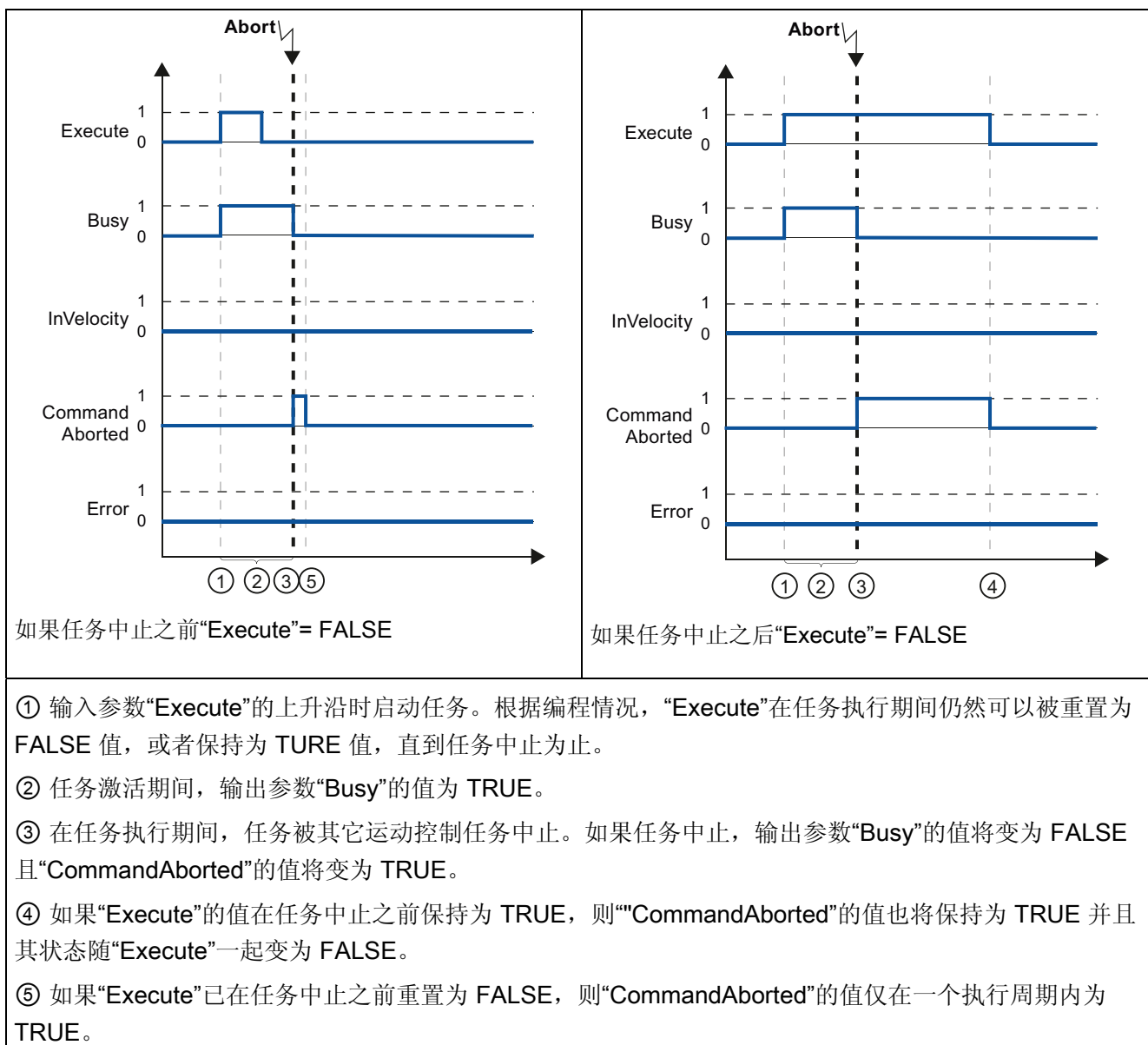
- 第一个示例显示了轴达到设置的速度时的行为。如果运动控制任务已在达到设置的速度前完成执行，则通过输出参数“InVelocity”的 TURE 值对此进行指示。输入参数“Execute”的信号状态影响输出参数“InVelocity”中的显示持续时间。
- 第二个示例显示了在达到设置的速度之前中止任务时的轴行为。如果运动控制任务在达到设置速度前中止，则通过输出参数“CommandAborted”的 TURE 值对此进行指示。输入参数“Execute”的信号状态影响输出参数“CommandAborted”中的显示持续时间。
- 第三个示例显示了在达到设置的速度之前出错时的轴行为。如果在运动控制任务执行期间，达到设置速度之前出错，则通过输出参数“Error”的 TURE 值对此进行指示。输入参数“Execute”的信号状态影响输出参数“Error”中的显示持续时间。

表格 9- 64 示例 1 - 如果达到设置的速度



- ① 输入参数“Execute”的上升沿时启动任务。根据编程情况，“Execute”的值可在达到设置的速度前重置为 FALSE，也可在达到设置的速度后重置为 FALSE。
- ② 任务激活期间，输出参数“Busy”的值为 TRUE。
- ③ 达到设置的速度时，输出参数“InVelocity”的值变为 TRUE。
- ④ 如果“Execute”的值在达到设置的速度后仍保持为 TRUE，则任务将保持激活状态。“InVelocity”和“Busy”的值保持为 TRUE 并且其状态仅随“Execute”一起变为 FALSE。
- ⑤ 如果“Execute”已在达到设置的速度前重置为 FALSE，则任务将在达到设置的速度时完成。“InVelocity”的值仅在一个执行周期内为 TRUE，并且随“Busy”一起变为 FALSE。

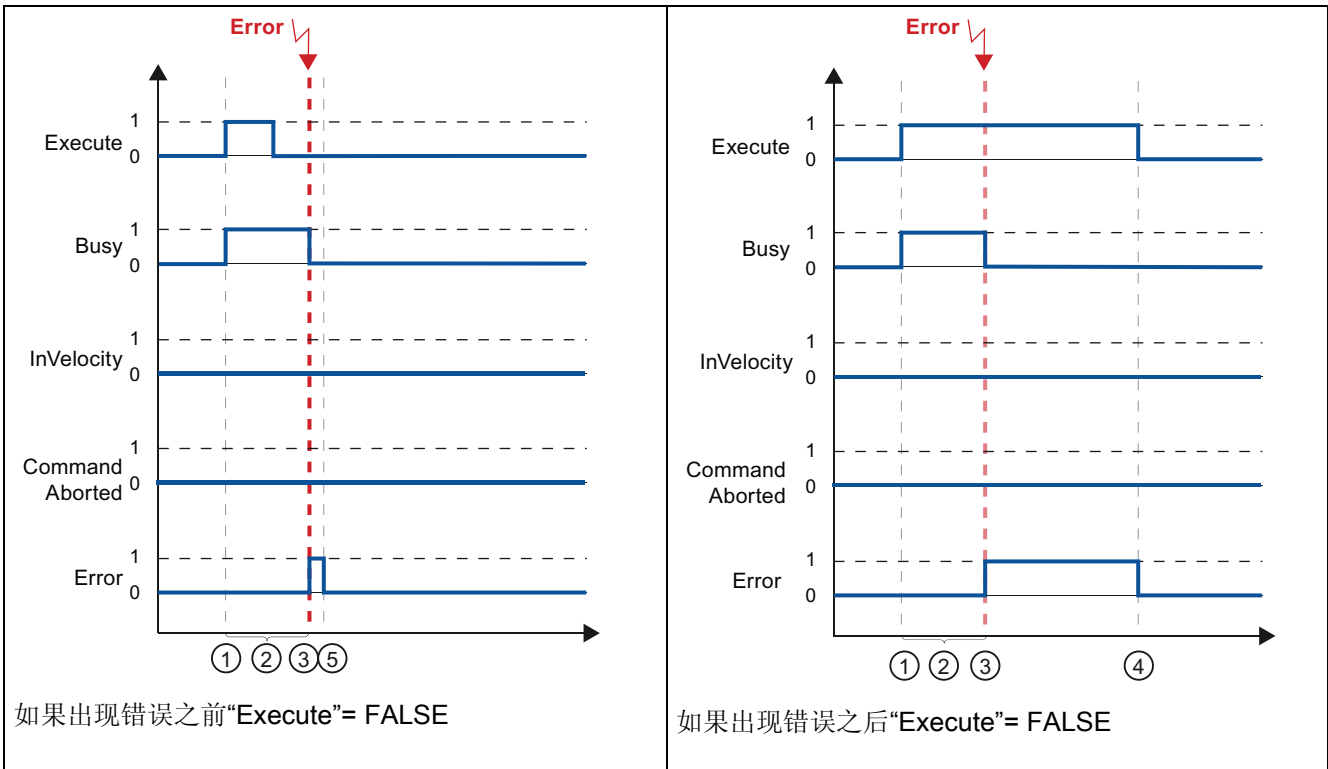
表格 9- 65 示例 2 - 如果任务在达到设置速度前中止

**说明**

在以下条件下，输出参数“CommandAborted”不指示出现中止：

- 已达到设置的速度，输入参数“Execute”的值为 FALSE，并且已启动一个新的运动控制任务。
- 达到设置的速度并且输入参数“Execute”的值为 FALSE 时，任务完成。所以不会将新任务的启动指示为中止。

表格 9-66 示例 3 - 如果在达到设置的速度之前出错



- ① 输入参数“Execute”的上升沿时启动任务。根据编程情况，“Execute”在任务执行期间仍然可以被重置为 FALSE 值，或者保持为 TRUE 值，直到出现错误为止。
- ② 任务激活期间，输出参数“Busy”的值为 TRUE。
- ③ 任务执行期间出错。出错时，输出参数“Busy”的值将变为 FALSE 且“Error”的值将变为 TRUE。
- ④ 如果“Execute”的值在出错之前保持为 TRUE，则“Error”的值也将保持为 TRUE 并且其状态仅随“Execute”一起变为 FALSE。
- ⑤ 如果“Execute”已在出错之前重置为 FALSE，则“Error”的值仅在一个执行周期内为 TRUE。

**说明**

在以下条件下，输出参数“Error”不指示出现错误：

- 已达到设置的速度，输入参数“Execute”的值为 FALSE，并且发生轴错误（例如，逼近软件限位开关）。
- 达到设置的速度并且输入参数“Execute”的值为 FALSE 时，任务完成。任务完成后，轴错误仅在运动控制指令“MC\_Power”中指示。



### 9.3.6.3 监视 MC\_MoveJog 指令

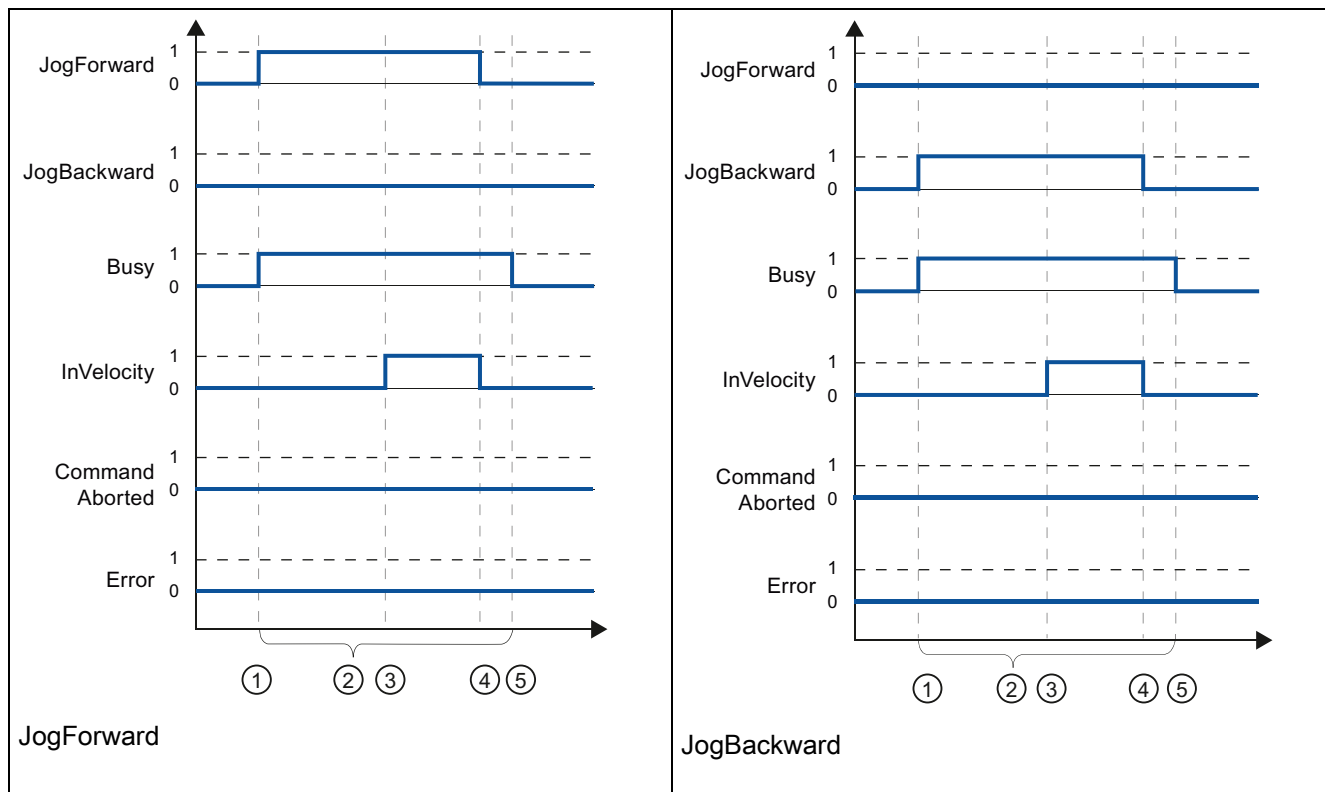
运动控制指令“MC\_MoveJog”的任务是实现点动操作。

- 运动控制任务“MC\_MoveJog”的结束没有明确的定义。第一次达到设置的速度且轴恒速运转时，就实现了任务目标。如果达到设置的速度，则通过输出参数“InVelocity”的 TURE 值对此进行指示。
- 输入参数“JogForward”或“JogBackward”的值已设置为 FALSE 并且轴已停止时，命令完成。
- 输出参数“Busy”、“CommandAborted”和“Error”发出信号，指示任务仍在处理、已中止或有未决的错误。
  - 在运动控制任务处理期间，输出参数“Busy”的值为 TRUE。如果任务已完成、中止或因错误停止，则输出参数“Busy”的值将变为 FALSE。
  - 只要轴在以设置的速度运转，输出参数“InVelocity”的值就为 TRUE。输出参数“CommandAborted”和“Error”保持该状态至少一个周期。只要输入参数“JogForward”或“JogBackward”设置为 TRUE，就锁存这些状态消息。

下图针对各种示例情况显示了状态位的操作。

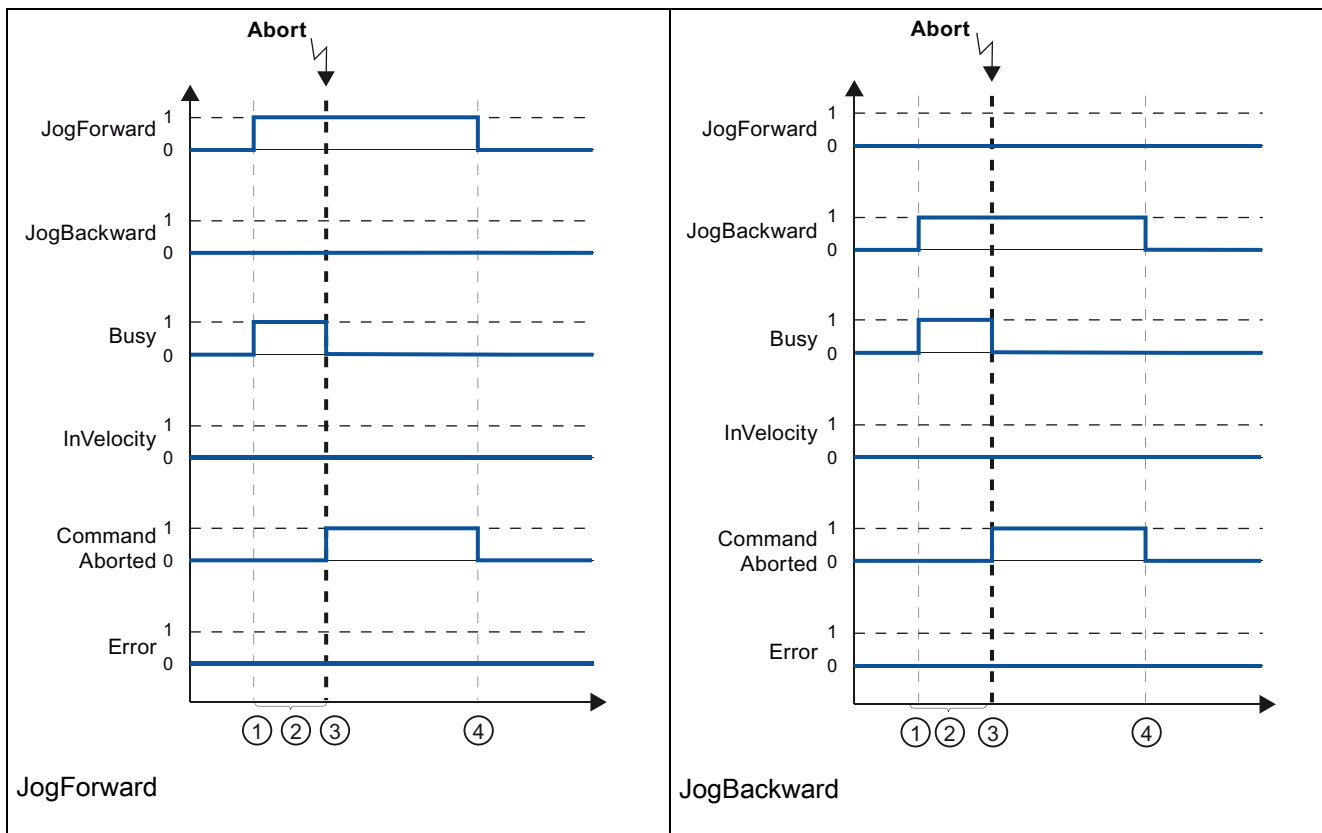
- 第一个示例显示了达到并保持设置的速度时的轴行为。如果运动控制任务已在达到设置的速度前完成执行，则通过输出参数“InVelocity”的 TURE 值对此进行指示。
- 第二个示例显示了任务中止时的轴行为。如果运动控制任务在执行期间中止，则通过输出参数“CommandAborted”的 TURE 值对此进行指示。该行为与是否达到设置的速度无关。
- 第三个示例显示了出现错误时的轴行为。如果在运动控制任务执行期间出错，则通过输出参数“Error”的 TURE 值对此进行指示。该行为与是否达到设置的速度无关。

表格 9- 67 示例 1 - 如果达到并保持设置的速度



- ① 输入参数“JogForward”或“JogBackward”的上升沿时启动任务。
- ② 任务激活期间，输出参数“Busy”的值为 TRUE。
- ③ 达到设置的速度时，输出参数“InVelocity”的值变为 TRUE。
- ④ 输入参数“JogForward”或“JogBackward”的值重置为 FALSE 时，轴运动结束。轴开始减速。结果，轴不再恒速运转并且输出参数“InVelocity”的状态变为 FALSE。
- ⑤ 如果轴已停止，则运动控制任务完成并且输出参数“Busy”的值将变为 FALSE。

表格 9- 68 示例 2 - 如果任务在执行期间中止

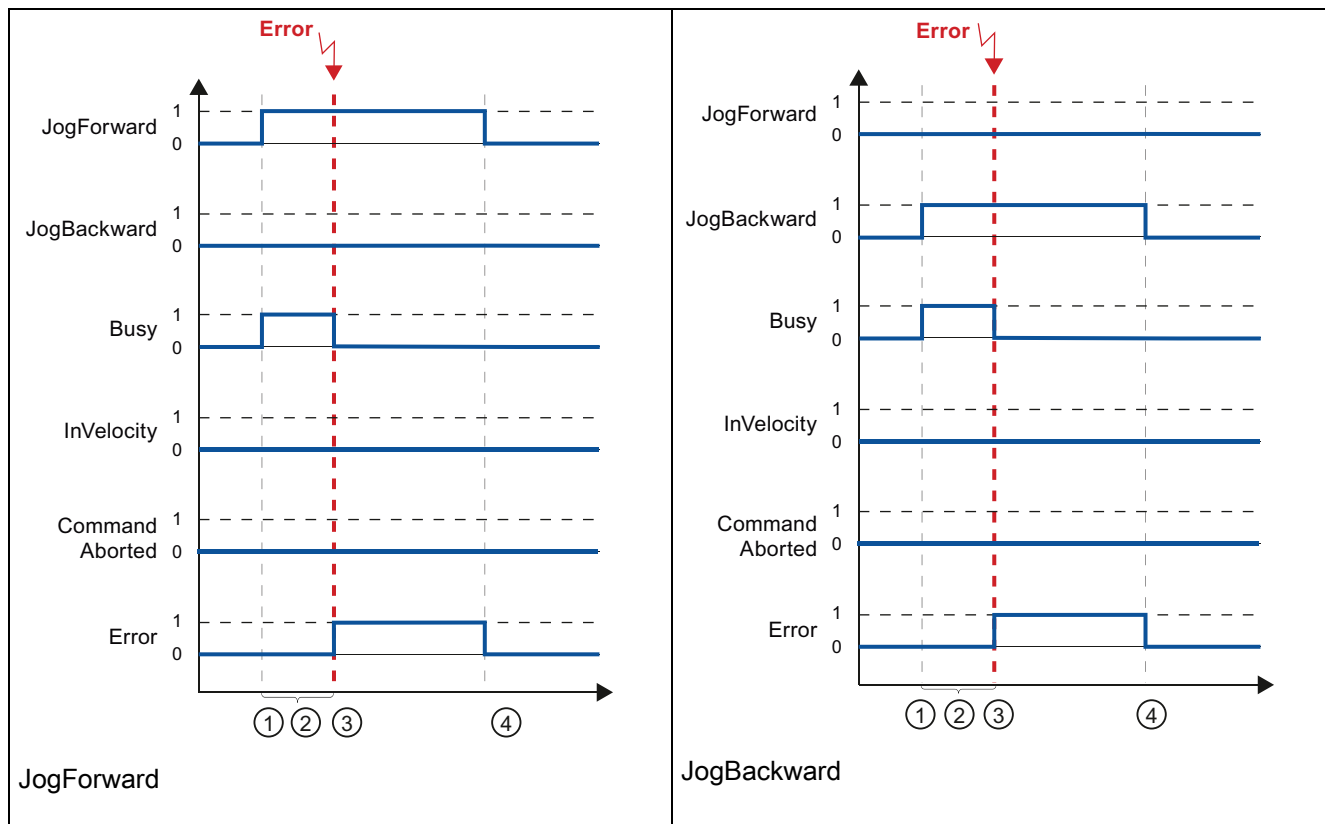


- ① 输入参数“JogForward”或“JogBackward”的上升沿时启动任务。
- ② 任务激活期间，输出参数“Busy”的值为 TRUE。
- ③ 在任务执行期间，任务被其它运动控制任务中止。如果任务中止，输出参数“Busy”的值将变为 FALSE 且“CommandAborted”的值将变为 TRUE。
- ④ 输入参数“JogForward”或“JogBackward”的值重置为 FALSE 时，输出参数“CommandAborted”的值将变为 FALSE。

**说明**

如果满足以下所有条件，则在输出参数“CommandAborted”中指示任务中止并且仅持续一个执行周期：  
 输入参数“JogForward”和“JogBackward”的值为 FALSE（但轴仍在减速），并且新的运动控制任务已启动。

表格 9- 69 示例 3 - 如果在任务执行期间出错



- ① 输入参数“JogForward”或“JogBackward”的上升沿时启动任务。
- ② 任务激活期间，输出参数“Busy”的值为 TRUE。
- ③ 任务执行期间出错。出错时，输出参数“Busy”的值将变为 FALSE 且“Error”的值将变为 TRUE。
- ④ 输入参数“JogForward”或“JogBackward”的值重置为 FALSE 时，输出参数“Error”的值将变为 FALSE。

**说明**

如果满足以下所有条件，则在输出参数“Error”中指示出错并且仅持续一个执行周期：  
 输入参数“JogForward”和“JogBackward”的值为 FALSE（但轴仍在减速），并且发生新错误（例如，逼近软件限位开关）。

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

## 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：

- CPU 提供以下 PROFINET 和 PROFIBUS 支持：
  - 在 V3.0 中，PROFINET 支持 16 个最多具有 256 个子模块的 IO 设备。  
PROFIBUS 允许 3 个独立的 PROFIBUS DP 主站，支持 32 个最多具有 512 个子模块的 IO 设备。
  - 在 V2.2 中，PROFINET 支持 8 个最多具有 128 个子模块的 IO 设备（如果配置了八个或更少的 PROFIBUS 从站或子模块）。PROFIBUS 在单个主站上最多支持 16 个 PROFIBUS IO 设备，每个 IO 设备最多具有 256 个子模块。
- S7 通信
- 用户数据报协议 (UDP)
- ISO on TCP (RFC 1006)
- 传输控制协议 (TCP)

## PROFINET RT IO 控制器

作为采用 PROFINET RT 的 IO 控制器，CPU 在本地 PN 网络上或通过 PN/DP 耦合器（连接器）提供以下支持。有关详细信息，请参见 PROFIBUS 和 PROFINET International (PI) ([www.profinet.com](http://www.profinet.com))。

- 在 V3.0 中，S7-1200 最多可与 16 个 PN 设备通信。
- 在 V2.2 中，S7-1200 最多可与 8 个 PN 设备通信。

## 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。

### 10.1 支持的异步通信连接数

- 在 V3.0 中，PROFIBUS DP 从站、PROFIBUS DP 主站以及 ASi（3 个左侧通信模块）和 PROFINET 是独立的。
- 在 V2.2 中，CPU 提供以下 PROFINET 和 PROFIBUS 支持：
  - 16 台设备和 256 个子模块，以及最多 16 台 PROFIBUS DP 从站设备和 256 个子模块（如果未组态任何 PROFINET IO 设备或子模块）。

---

#### 说明

在 V2.2 中共有 16 台 PROFINET 和 PROFIBUS 设备，其中包括以下内容：

- 通过 PROFIBUS DP 主站 (CM 1243-5) 连接的 PROFIBUS DP 从站模块
- 任何与 CPU 相连接的 PROFIBUS DP 从站模块 (CM 1242-5)
- 通过 PROFINET 端口与 CPU 连接的任何 PROFINET 设备

例如，对于采用一个 CM 1243-5 主站模块和两个 CM 1242-5 从站模块，共三个 PROFIBUS CM 的配置，PROFIBUS DP 主站 (CM 1243-5) 能够访问的最大从站模块的数量将降为 14。

---

- 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

### 远程服务通信

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。

## 10.1 支持的异步通信连接数

对于 PROFINET 和 PROFIBUS，CPU 最多可支持下列数量的并发异步通信连接：

- 8 个用于开放式用户通信（主动或被动）的连接：TSEND\_C、TRCV\_C、TCON、TDISCON、TSEND 和 TRCV。
- 用于服务器 GET/PUT 数据的 3 个 CPU 至 CPU S7 连接
- 用于客户端 GET/PUT 数据的 8 个 CPU 至 CPU S7 连接

---

#### 说明

S7-1200、S7-300 和 S7-400 CPU 使用 GET 和 PUT 指令进行 CPU 至 CPU S7 通信。S7-200 CPU 使用 ETHx\_XFER 指令进行 CPU 至 CPU S7 通信。

---

- **HMI 连接：** CPU 提供专用的 HMI 连接，以支持最多 3 个 HMI 设备。（最多可以有 2 个 SIMATIC 精智面板。）支持的 HMI 总数受组态中 HMI 面板类型的影响。例如，可以将最多 3 个 SIMATIC 基本面板连接到 CPU，或者最多可以连接两个 SIMATIC 精智面板与一个附加基本面板。
- **PG 连接：** CPU 提供连接以支持 1 个编程设备 (PG)。
- **Webserver (HTTP) 连接：** CPU 提供用于 Webserver 的连接。

## 10.2 PROFINET

### 10.2.1 本地/伙伴连接

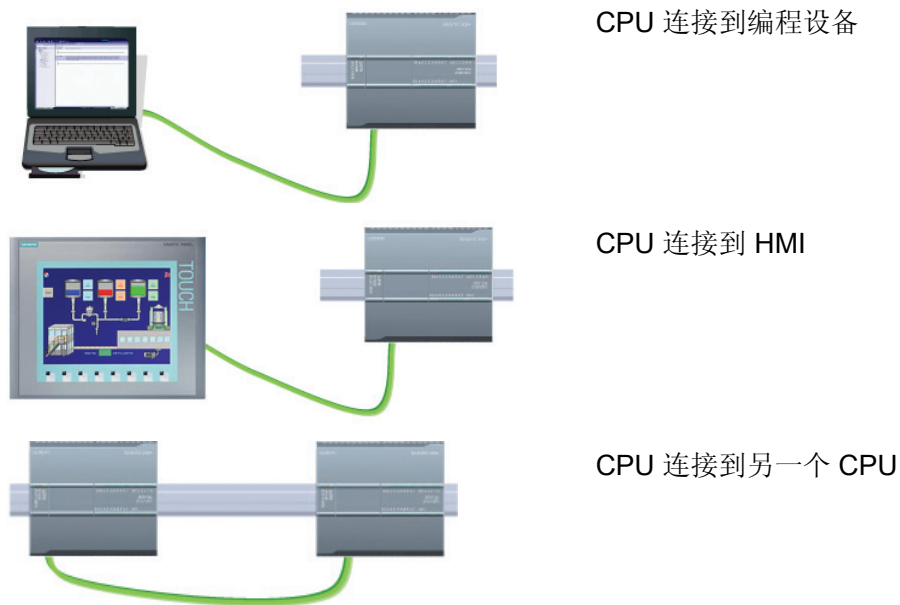
本地/伙伴（远程）连接定义两个通信伙伴的逻辑分配以建立通信服务。连接定义了以下内容：

- 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。有关组态连接参数的信息，请参见“组态本地/伙伴连接” (页 129)部分。

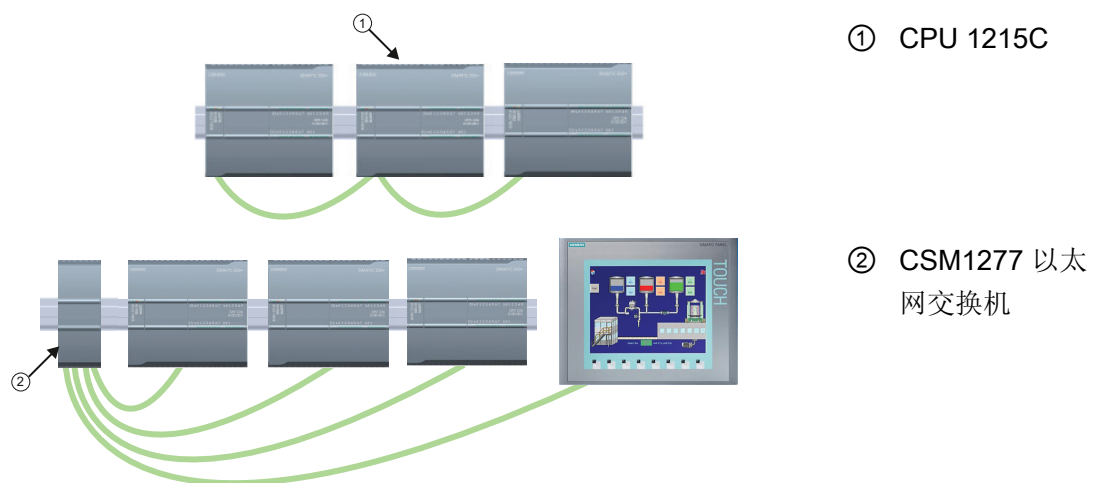
如果连接终止（例如，因断线），主动伙伴将尝试重新建立组态的连接。不必再次执行通信指令。

CPU 可使用标准 TCP 通信协议与其它 CPU、编程设备、HMI 设备和非 Siemens 设备通信。



以太网交换

CPU 1211C、1212C 和 1214C 上的 PROFINET 端口不包含以太网交换设备。编程设备或 HMI 与 CPU 之间的直接连接不需要以太网交换机。不过，含有两个以上的 CPU 或 HMI 设备的网络需要以太网交换机。





CPU 1215C 没有内置 2 端口以太网交换机。您可使用具有 CPU 1215C 和另两个 S7-1200 CPU 的网络。也可以使用安装在机架上的 CSM1277 4 端口以太网交换机来连接多个 CPU 和 HMI 设备。

## 10.2.2 开放式用户通信

### 10.2.2.1 PROFINET 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) 来组态连接的参数。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。

---

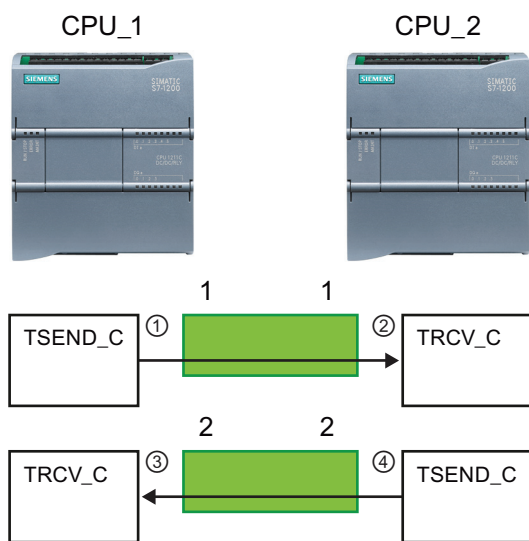
#### 说明

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。

---

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。

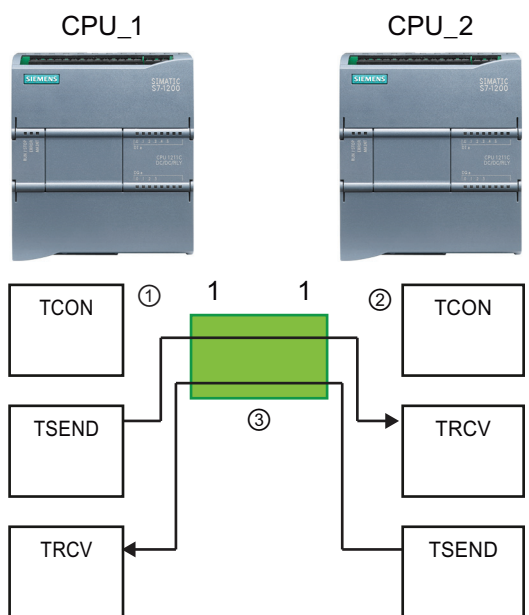
- CPU\_1 中的 TSEND\_C 指令通过第一个连接（CPU\_1 和 CPU\_2 上的“连接 ID 1”）与 CPU\_2 中的 TRCV\_V 链接。
- CPU\_1 中的 TRCV\_C 指令通过第二个连接（CPU\_1 和 CPU\_2 上的“连接 ID 2”）与 CPU\_2 中的 TSEND\_C 链接。



- 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)。

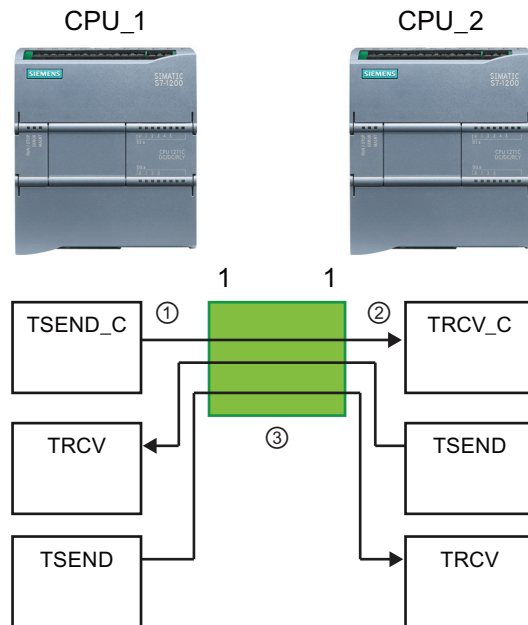
以下示例显示了两个 CPU 之间的通信，这两个 CPU 使用 1 个连接来发送和接收数据。

- 每个 CPU 都使用 TCON 指令来组态两个 CPU 之间的连接。
- CPU\_2 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)。

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)。

## 参见

组态本地/伙伴连接路径 (页 129)

### 10.2.2.2 协议

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 10-1 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	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、 TDISCON、TSEND 和 TRCV	
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	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、 TDISCON、TSEND 和 TRCV	
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
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 RT	CPU 与 PROFINET IO 设 备通信	指定长度的数据传输 和接收	内置	内置

### 10.2.2.3 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

如果将数据存储于“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, 可将 LEN 参数设置为 65535 (0xFFFF)。

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。

#### 10.2.2.4 TCP 和 ISO on TCP

传输控制协议 (TCP) 是由 RFC 793 描述的一种标准协议：传输控制协议。TCP 的主要用途是在过程对之间提供可靠、安全的连接服务。该协议有以下特点：

- 由于它与硬件紧密相关，因此它是一种高效的通信协议
- 它适合用于中等大小或较大的数据量（最多 8192 字节）
- 它为应用带来了更多的便利，特别是对于错误恢复、流控制和可靠性。
- 它是一种面向连接的协议
- 它可以非常灵活地用于只支持 TCP 的第三方系统
- 有路由功能
- 只能应用静态数据长度。
- 消息会被确认。
- 使用端口号对应用程序寻址。
- 大多数用户应用协议（例如 TELNET 和 FTP）都使用 TCP。
- 由于使用 SEND/RECEIVE 编程接口的缘故，需要编程来进行数据管理。

基于传输控制协议 (TCP) 的国际标准组织 (ISO) (RFC 1006) (ISO on TCP) 是一种能够将 ISO 应用移植到 TCP/IP 网络的机制。该协议有以下特点：

- 它是与硬件关系紧密的高效通信协议
- 它适合用于中等大小或较大的数据量（最多 8192 字节）
- 与 TCP 相比，它的消息提供了数据结束标识符并且它是面向消息的。
- 具有路由功能；可用于 WAN
- 可用于实现动态数据长度。
- 由于使用 SEND/RECEIVE 编程接口的缘故，需要编程来进行数据管理。

通过传输服务访问点 (TSAP, Transport Service Access Point)，TCP 协议允许有多个连接访问单个 IP 地址（最多 64K 个连接）。借助 RFC 1006，TSAP 可唯一标识连接到同一个 IP 地址的这些通信端点连接。

## TSEND\_C 和 TRCV\_C

TSEND\_C 指令兼具 TCON、TDISCON 和 TSEND 指令的功能。TRCV\_C 指令兼具 TCON、TDISCON 和 TRCV 指令的功能。（有关这些指令的详细信息，请参见“TCON、TDISCON、TSEND 和 TRCV (页 473)”。）

最少可传送 (TSEND\_C) 或接收 (TRCV\_C) 1 字节的数据，最多 8192 字节。

TSEND\_C 不支持传送布尔位置的数据，TRCV\_C 也不会在布尔位置中接收数据。有关使用这些指令传送数据的信息，请参阅 数据一致性 (页 154) 部分。

---

### 说明

#### 初始化通信参数

插入 TSEND\_C 或 TRCV\_C 指令之后，可使用该指令 (页 129) 的“属性”(Properties) 来组态通信参数。在巡视窗口为通信伙伴输入参数时，STEP 7 会在指令的背景数据块中输入相应数据。

如果要使用多重背景数据块，必须在两个 CPU 上手动组态该 DB。

---

表格 10-2 TSEND\_C 和 TRCV\_C 指令

LAD/FBD	SCL	说明
	<pre>"TSEND_C_DB" (   req:=_bool_in_,   cont:=_bool_in_,   len:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   connect:=_struct_inout_,   data:=_variant_inout_,   com rst:=_bool_inout_);</pre>	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。
	<pre>"TRCV_C_DB" (   en_r:=_bool_in_,   cont:=_bool_in_,   len:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   rcvd_len=&gt;_uint_out_,   connect:=_struct_inout_,   data:=_variant_inout_,   com rst:=_bool_inout_);</pre>	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。

1 STEP 7 会在插入指令时自动创建 DB。

表格 10-3 TSEND\_C 和 TRCV\_C 参数的数据类型

参数和类型		数据类型	说明
REQ (TSEND_C)	IN	Bool	控制参数 REQ 在上升沿启动具有 CONNECT 中所述连接的发送作业。
EN_R (TRCV_C)	IN	Bool	启用接收的控制参数：EN_R = 1 时，TRCV_C 准备接收。处理接收作业。
CONT	IN	Bool	<ul style="list-style-type: none"> <li>0: 断开连接</li> <li>1: 建立并保持连接</li> </ul>
LEN	IN	UInt	要发送 (TSEND_C) 或接收 (TRCV_C) 的最大字节数： <ul style="list-style-type: none"> <li>默认 = 0: DATA 参数确定要发送 (TSEND_C) 或接收 (TRCV_C) 的数据长度。</li> <li>特殊模式 = 65535: 设置可变长度的数据接收 (TRCV_C)。</li> </ul>

参数和类型		数据类型	说明
CONNECT	IN_OUT	TCON_Param	指向连接描述的指针
DATA	IN_OUT	Variant	<ul style="list-style-type: none"> <li>包含要发送数据的地址和长度 (TSEND_C)</li> <li>包含接收数据的起始地址和最大长度 (TRCV_C)。</li> </ul>
COM_RST	IN_OUT	Bool	允许重新启动指令： <ul style="list-style-type: none"> <li>0: 不相关</li> <li>1: 完成函数块的重新启动，现有连接将终止。</li> </ul>
DONE	OUT	Bool	<ul style="list-style-type: none"> <li>0: 作业尚未开始或仍在运行。</li> <li>1: 作业无错完成。</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0: 作业完成。</li> <li>1: 作业尚未完成。无法触发新作业。</li> </ul>
ERROR	OUT	Bool	状态参数，可具有以下值： <ul style="list-style-type: none"> <li>0: 无错误</li> <li>1: 处理时出错。STATUS 提供错误类型的详细信息。</li> </ul>
STATUS	OUT	Word	包括错误信息的状态信息。（请参见下表中的“错误和状态参数”。）
RCVD_LEN (TRCV_C)	OUT	Int	实际接收到的数据量（字节）

**说明**

TSEND\_C 指令需要通过 REQ 输入参数的上升沿来启动发送作业。然后，BUSY 参数在处理期间会设置为 1。发送作业完成时，将通过 DONE 或 ERROR 参数被设置为 1 并持续一个扫描周期进行指示。在此期间，将忽略 REQ 输入参数的上升沿。

**说明**

LEN 参数的默认设置 (LEN = 0) 使用 DATA 参数来确定要传送的数据的长度。确保 TSEND\_C 指令传送的 DATA 的大小与 TRCV\_C 指令的 DATA 参数的大小相同。



## TSEND\_C 操作

下列功能说明了 TSEND\_C 指令的操作：

- 要建立连接，请在  $CONT = 1$  时执行 TSEND\_C。
- 成功建立连接后，TSEND\_C 便会置位 DONE 参数一个周期。
- 要终止通信连接，请在  $CONT = 0$  时执行 TSEND\_C。连接将立即中止。这还会影响接收站。将在接收站关闭该连接，并且接收缓冲区内的数据可能会丢失。
- 要通过建立的连接发送数据，请在 REQ 的上升沿执行 TSEND\_C。发送操作成功执行后，TSEND\_C 便会置位 DONE 参数一个周期。
- 要建立连接并发送数据，请在  $CONT = 1$  且  $REQ = 1$  时执行 TSEND\_C。发送操作成功执行后，TSEND\_C 便会置位 DONE 参数一个周期。

## TRCV\_C 操作

下列功能说明了 TRCV\_C 指令的操作：

- 要建立连接，请在参数  $CONT = 1$  时执行 TRCV\_C。
- 要接收数据，请在参数  $EN\_R = 1$  时执行 TRCV\_C。参数  $EN\_R = 1$  且  $CONT = 1$  时，TRCV\_C 连续接收数据。
- 要终止连接，请在参数  $CONT = 0$  时执行 TRCV\_C。连接将立即中止且数据可能丢失。

TRCV\_C 处理与 TRCV 指令相同的接收模式。下表说明了在接收区输入数据的方法。

表格 10-4 将数据输入接收区

协议选项	将数据输入接收区	参数 "connection_type"	LEN 参数的值	RCVD_LEN 参数的值 (字节)
TCP	特殊模式	B#16#11	65535	1 到 1472
TCP	指定长度的数据接收	B#16#11	0 (推荐) 或 1 到 8192, 65535 除外	1 到 8192
ISO on TCP	特殊模式	B#16#12	65535	1 到 1472
ISO on TCP	协议控制	B#16#12	0 (推荐) 或 1 到 8192, 65535 除外	1 到 8192

**说明****特殊模式**

使用 TCP 或 ISO on TCP 协议时可以存在“特殊模式”。用户通过将“65535”分配给 LEN 参数来设置特殊模式。接收区与 DATA 构成的区域相同。接收数据的长度将输出到参数 RCVD\_LEN 中。

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。

**说明****将包含“特殊模式”的 S7-300/400 STEP 7 项目导入 S7-1200 中**

在 S7-300/400 STEP 7 项目中，通过将“0”分配给 LEN 参数来选择“特殊模式”。在 S7-1200 中，用户通过将“65535”分配给 LEN 参数来设置特殊模式。

如果将包含“特殊模式”的 S7-300/400 STEP 7 项目导入 S7-1200 中，则必须将 LEN 参数更改为“65535”。

**说明**

由于 TSEND\_C 采用异步处理，所以在 DONE 参数值或 ERROR 参数值为 TRUE 前，必须保持发送方区域中的数据一致。

对于 TSEND\_C，参数 DONE 状态为 TRUE 表示数据已成功发送。但并不表示连接伙伴 CPU 实际读取了接收缓冲区。

由于 TRCV\_C 采用异步处理，因此仅当参数 DONE = 1 时，接收方区域中的数据才一致。

表格 10-5 TSEND\_C 和 TRCV\_C 指令的 BUSY、DONE 和 ERROR 参数

BUSY	DONE	ERROR	说明
TRUE	不相关	不相关	正在处理作业。
FALSE	TRUE	FALSE	作业已成功完成。
FALSE	FALSE	TRUE	作业因错结束。出错原因可在 STATUS 参数中找到。
FALSE	FALSE	FALSE	未分配新作业。

## Error 和 Status 参数

表格 10-6 TSEND\_C 和 TRCV\_C 指令的 ERROR 和 STATUS 的条件代码

ERROR	STATUS	说明
0	0000	作业已无错执行
0	7000	无激活的作业处理
0	7001	启动作业处理，正在建立连接，正在等待连接伙伴
0	7002	正在发送或接收数据
0	7003	正终止连接
0	7004	连接已建立并受到监视，无激活的作业处理
1	8085	LEN 参数的值大于允许的最大值。
1	8086	CONNECT 参数超出允许范围。
1	8087	已达到最大连接数；无法建立更多连接。
1	8088	LEN 参数对于在 DATA 中指定的存储区无效。
1	8089	CONNECT 参数未指向数据块。
1	8091	超出最大嵌套深度。
1	809A	CONNECT 参数指向的字段与连接描述的长度不匹配。
1	809B	连接描述中的 local_device_id 与 CPU 的不匹配。
1	80A1	通信错误： <ul style="list-style-type: none"> <li>• 尚未建立指定的连接</li> <li>• 当前正在终止指定的连接；无法通过该连接传输</li> <li>• 正在重新初始化接口</li> </ul>
1	80A3	正在尝试终止不存在的连接
1	80A4	远程伙伴连接的 IP 地址无效。例如，远程伙伴的 IP 地址与本地伙伴的 IP 地址相同。
1	80A5	连接 ID 已被使用。
1	80A7	通信错误：在 TSEND_C 完成前调用了 TDISCON。
1	80B2	参数 CONNECT 指向使用关键字 UNLINKED 生成的数据块。

ERROR	STATUS	说明
1	80B3	不一致的参数： <ul style="list-style-type: none"> <li>• 连接描述错误</li> <li>• 本地端口（参数 local_tsap_id）已在另一个连接描述中存在。</li> <li>• 连接描述中的 ID 与作为参数指定的 ID 不同</li> </ul>
1	80B4	使用 ISO on TCP (connection_type = B#16#12) 建立被动连接时，条件代码 80B4 提示您输入的 TSAP 不符合下列某一项地址要求： <ul style="list-style-type: none"> <li>• 若是本地 TSAP 长度为 2 个字节且首字节的 TSAP ID 值为 E0 或 E1（十六进制），第二字节必须为 00 或 01。</li> <li>• 如果本地 TSAP 长度为 3 个或更多字节，且首字节的 TSAP ID 值为 E0 或 E1（十六进制），则第二字节必须为 00 或 01，且所有其它字节必须为有效的 ASCII 字符。</li> <li>• 如果本地 TSAP 长度为 3 个或更多字节，且首字节的 TSAP ID 值既不为 E0 也不为 E1（十六进制），则 TSAP ID 的所有字节都必须为有效的 ASCII 字符。</li> </ul> 有效 ASCII 字符的字节值为 20 到 7E（十六进制）。
1	80B7	所传送数据的数据类型和/或长度与伙伴 CPU 上用于写入该数据的区域不相符。
1	80C3	所有连接资源都在使用。
1	80C4	临时通信错误： <ul style="list-style-type: none"> <li>• 此时无法建立连接</li> <li>• 接口正在接收新参数</li> <li>• TDISCON 当前正在删除已组态连接。</li> </ul>
1	8722	CONNECT 参数：源区域无效：DB 中不存在该区域。
1	873A	CONNECT 参数：无法访问连接描述（例如，DB 不可用）
1	877F	CONNECT 参数：内部错误，如无效的 ANY 引用
1	893A	参数包含未装载的 DB 的编号。

## 以太网连接协议

每个 CPU 都集成了一个支持标准 PROFINET 通信的 PROFINET 端口。TSEND\_C、TRCV\_C、TSEND 和 TRCV 指令均支持 TCP 和 ISO on TCP 以太网协议。

更多相关信息，请参见“设备配置：组态本地/伙伴连接路径 (页 129)”。

## 参见

PROFINET 连接的参数 (页 132)

## TCON、TDISCON、TSEND 和 TRCV

## 使用 TCP 和 ISO on TCP 协议的以太网通信

## 说明

## TSEND\_C 和 TRCV\_C 指令

为帮助简化 PROFINET/以太网通信的编程，TSEND\_C 指令和 TRCV\_C 指令兼具 TCON、TDISCON、TSEND 和 TRCV 指令的功能：

- TSEND\_C 兼具 TCON、TDISCON 和 TSEND 指令的功能。
- TRCV\_C 兼具 TCON、TDISCON 和 TRCV 指令的功能。

以下指令控制通信过程：

- TCON 在客户机与服务器 (CPU) PC 之间建立 TCP/IP 连接。
- TSEND 和 TRCV 发送和接收数据。
- TDISCON 断开连接。

最少可传送 (TSEND) 或接收 (TRCV) 一个字节的的数据，最多 8192 字节。TSEND 不支持传送布尔位置的数据，TRCV 也不会布尔位置中接收数据。有关使用这些指令传送数据的信息，请参阅 数据一致性 (页 154)部分。

TCON、TDISCON、TSEND 和 TRCV 异步运行，即，作业处理需要多次执行指令来完成。例如，执行参数 REQ = 1 的 TCON 指令来启动用于设置和建立连接的作业。然后，另外执行 TCON 来监视作业进度并使用参数 DONE 来测试作业是否完成。

下表给出了 BUSY、DONE 和 ERROR 之间的关系。使用该表可以确定当前作业状态。

表格 10-7 BUSY、DONE 和 ERROR 参数之间的交互作用

BUSY	DONE	ERROR	说明
TRUE	不相关	不相关	正在处理作业。
FALSE	TRUE	FALSE	作业已成功完成。
FALSE	FALSE	TRUE	作业因错结束。出错原因可在 STATUS 参数中找到。
FALSE	FALSE	FALSE	未分配新作业。

## TCON 和 TDISCON

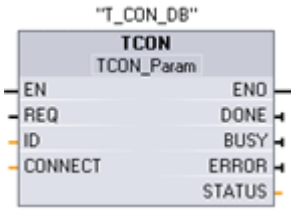
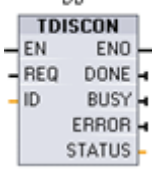
## 说明

## 初始化通信参数

插入 TCON 指令之后，可使用该指令 (页 129) 的“属性”(Properties) 来组态通信参数。在巡视窗口为通信伙伴输入参数时，STEP 7 会在指令的背景数据块中输入相应数据。

如果要使用多重背景数据块，必须在两个 CPU 上手动组态该 DB。

表格 10-8 TCON 和 TDISCON 指令

LAD/FBD		说明
	<pre>"TCON_DB" (   req:=_bool_in_,   ID:=_undef_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   connect:= struct inout );</pre>	TCP 和 ISO on TCP: TCON 启动从 CPU 到通信伙伴的通信连接。
	<pre>"TDISCON_DB" (   req:=_bool_in_,   ID:=_word_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_);</pre>	TCP 和 ISO on TCP: TDISCON 终止从 CPU 到通信伙伴的通信连接。

1 STEP 7 会在插入指令时自动创建 DB。

表格 10-9 TCON 和 TDISCON 参数的数据类型

参数和类型	数据类型	说明	
REQ	IN	Bool	控制参数 REQ 启动用于建立通过 ID 指定的连接的作业。该作业在上升沿时启动。
ID	IN	CONN_OUC (Word)	引用要建立的 (TCON) 或终止的 (TDISCON)、连接到远程伙伴或在用户程序和操作系统通信层之间的连接。ID 必须与本地连接描述中的相关参数 ID 相同。 值范围: W#16#0001 到 W#16#0FFF

参数和类型		数据类型	说明
CONNECT (TCON)	IN_OUT	TCON_Param	指向连接描述的指针
DONE	OUT	Bool	<ul style="list-style-type: none"> <li>0: 作业尚未开始或仍在运行。</li> <li>1: 作业无错完成。</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0: 作业完成。</li> <li>1: 作业尚未完成。无法触发新作业。</li> </ul>
ERROR	OUT	Bool	状态参数，可具有以下值： <ul style="list-style-type: none"> <li>0: 无错误</li> <li>1: 处理时出错。STATUS 提供错误类型的详细信息。</li> </ul>
STATUS	OUT	Word	包括错误信息的状态信息。（请参见下表中的错误和状态条件代码。）

两个通信伙伴都执行 TCON 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，因断线或远程通信伙伴原因），主动伙伴将尝试重新建立组态的连接。不必再次执行 TCON。

执行 TDISCON 指令或 CPU 切换到 STOP 模式后，会终止现有连接并删除所设置的连接。要设置和重新建立连接，必须再次执行 TCON。

表格 10- 10 ERROR 和 STATUS 指令的 TCON 和 TDISCON 的条件代码

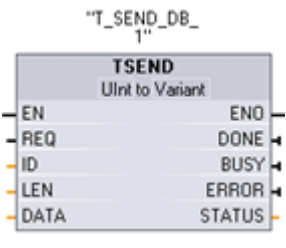
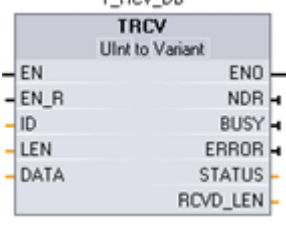
ERROR	STATUS	说明
0	0000	连接已成功建立。
0	7000	无激活的作业处理
0	7001	启动作业处理；正在建立连接 (TCON) 或正在终止连接 (TDISCON)
0	7002	后续调用（与 REQ 无关）；正在建立连接 (TCON) 或正在终止连接 (TDISCON)
1	8086	参数 ID 超出允许的地址范围。
1	8087	TCON: 已达到最大连接数；无法建立更多连接。
1	809B	TCON: 连接描述中的 local_device_id 与 CPU 不匹配。
1	80A1	TCON: 连接或端口已被用户占用。
1	80A2	TCON: 本地端口或远程端口已被系统占用。

ERROR	STATUS	说明
1	80A3	正在尝试重新建立现有连接 (TCON) 或终止不存在的连接 (TDISCON)。
1	80A4	TCON: 远程连接端点的 IP 地址无效; 可能与本地 IP 地址匹配。
1	80A5	TCON: 连接 ID 已被使用。
1 ( )	80A7	TCON: 通信错误: 在 TCON 完成前执行了 TDISCON。TDISCON 必须先完全终止 ID 引用的连接。
1	80B4	TCON: 使用 ISO on TCP (connection_type = B#16#12) 建立被动连接时, 条件代码 80B4 提示您输入的 TSAP 不符合下列某一项地址要求: <ul style="list-style-type: none"> <li>• 如果本地 TSAP 长度为 2 个字节且首字节的 TSAP ID 值为 E0 或 E1 (十六进制), 则第二字节必须为 00 或 01。</li> <li>• 如果本地 TSAP 长度为 3 个或更多字节, 且首字节的 TSAP ID 值为 E0 或 E1 (十六进制), 则第二字节必须为 00 或 01, 且所有其它字节必须为有效的 ASCII 字符。</li> <li>• 如果本地 TSAP 长度为 3 个或更多字节, 且首字节的 TSAP ID 值既不为 E0 也不为 E1 (十六进制), 则 TSAP ID 的所有字节都必须为有效的 ASCII 字符。</li> </ul> 有效 ASCII 字符的字节值为 20 到 7E (十六进制)。
1	80B6	TCON: 参数 connection_type 的参数分配错误
1	80B7	TCON: 所传送数据的数据类型和/或长度与伙伴 CPU 上用于写入该数据的区域不相符。
1)	80B8	TCON: 本地连接描述中的参数与参数 ID 不同。
1	80C3	TCON: 所有连接资源都在使用。
1	80C4	临时通信错误: <ul style="list-style-type: none"> <li>• 此时无法建立连接 (TCON)。</li> <li>• TDISCON (TCON) 当前正在删除已组态连接。</li> <li>• 当前正在建立连接 (TDISCON)。</li> <li>• 接口正在接收新参数 (TCON 和 TDISCON)。</li> </ul>



## TSEND 和 TRCV

表格 10- 11 TSEND 和 TRCV 指令

LAD/FBD	SCL	说明
	<pre>"TSEND_DB" (   req:=_bool_in_,   ID:=_word_in_,   len:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   data:=_variant_inout_);</pre>	TCP 和 ISO on TCP: TSEND 通过从 CPU 到伙伴站的通信连接发送数据。
	<pre>"TRCV_DB" (   en_r:=_bool_in_,   ID:=_word_in_,   len:=_uint_in_,   ndr=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   rcvd_len=&gt;_uint_out_,   data:=_variant_inout_);</pre>	TCP 和 ISO on TCP: TRCV 通过从伙伴站到 CPU 的通信连接接收数据。

1 STEP 7 会在插入指令时自动创建 DB。

表格 10- 12 TSEND 和 TRCV 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	TSEND: 在上升沿启动发送作业。 传送通过 DATA 和 LEN 指定的区域中的数据。
EN_R	IN	Bool	TRCV: 允许 CPU 进行接收; EN_R = 1 时, TRCV 准备接收。 处理接收作业。
ID	IN	CONN_OUC (Word)	引用相关的连接。 ID 必须与本地连接描述中的相关参数 ID 相同。 值范围: W#16#0001 到 W#16#0FFF
LEN	IN	UInt	要发送 (TSEND) 或接收 (TRCV) 的最大字节数: <ul style="list-style-type: none"> <li>默认 = 0: DATA 参数确定要发送 (TSEND) 或接收 (TRCV) 的数据长度。</li> <li>特殊模式 = 65535: 设置可变长度的数据接收 (TRCV)。</li> </ul>

参数和类型		数据类型	说明
DATA	IN_OUT	Variant	指向发送 (TSEND) 或接收 (TRCV) 数据区的指针；数据区包含地址和长度。该地址引用 I 存储器、Q 存储器、M 存储器或 DB。
DONE	OUT	Bool	TSEND: <ul style="list-style-type: none"> <li>0: 作业尚未开始或仍在运行。</li> <li>1: 无错执行作业。</li> </ul>
NDR	OUT	Bool	TRCV: <ul style="list-style-type: none"> <li>NDR = 0: 作业尚未启动或仍在运行。</li> <li>NDR = 1: 作业已成功完成。</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>BUSY = 1: 作业尚未完成。无法触发新作业。</li> <li>BUSY = 0: 作业已完成。</li> </ul>
ERROR	OUT	Bool	ERROR = 1: 处理期间出错。STATUS 提供错误类型的详细信息
STATUS	OUT	Word	包括错误信息的状态信息。（请参见下表中的错误和状态条件代码。）
RCVD_LEN	OUT	Int	TRCV: 实际接收到的数据量（字节）

### 说明

TSEND 指令需要通过 REQ 输入参数的上升沿来启动发送作业。然后，BUSY 参数在处理期间会设置为 1。发送作业完成时，将通过 DONE 或 ERROR 参数被设置为 1 并持续一个扫描周期进行指示。在此期间，将忽略 REQ 输入参数的上升沿。

### TRCV 操作

TRCV 指令将收到的数据写入到通过以下两个变量指定的接收区：

- 指向区域起始位置的指针
- 如果不为 0 则为区域长度或 LEN 上提供的值

### 说明

LEN 参数的默认设置 (LEN = 0) 使用 DATA 参数来确定要传送的数据的长度。确保 TSEND 指令传送的 DATA 的大小与 TRCV 指令的 DATA 参数的大小相同。

接收所有作业数据后，TRCV 会立即将其传送到接收区并将 NDR 设置为 1。

表格 10- 13 将数据输入接收区

协议选项	将数据输入接收区	参数“connection_type”	LEN 参数的值	RCVD_LEN 参数的值 (字节)
TCP	特殊模式	B#16#11	65535	1 到 1472
TCP	指定长度的数据接收	B#16#11	0 (推荐) 或 1 到 8192, 65535 除外	1 到 8192
ISO on TCP	特殊模式	B#16#12	65535	1 到 1472
ISO on TCP	协议控制	B#16#12	0 (推荐) 或 1 到 8192, 65535 除外	1 到 8192

**说明****特殊模式**

使用 TCP 或 ISO on TCP 协议时可以存在“特殊模式”。用户通过将“65535”分配给 LEN 参数来设置特殊模式。接收区与 DATA 构成的区域相同。接收数据的长度将输出到参数 RCVD\_LEN 中。接收数据块后，TRCV 会立即将数据写入接收区并将 NDR 设置为 1。

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。

**说明****将包含“特殊模式”的 S7-300/400 STEP 7 项目导入 S7-1200 中**

在 S7-300/400 STEP 7 项目中，通过将“0”分配给 LEN 参数来选择“特殊模式”。在 S7-1200 中，用户通过将“65535”分配给 LEN 参数来设置特殊模式。

如果将包含“特殊模式”的 S7-300/400 STEP 7 项目导入 S7-1200 中，则必须将 LEN 参数更改为“65535”。

表格 10- 14 ERROR 和 STATUS 指令的 TSEND 和 TRCV 的条件代码

ERROR	STATUS	说明
0	0000	<ul style="list-style-type: none"> <li>发送作业无错完成 (TSEND)</li> <li>已接受新数据：在 RCVD_LEN 中显示已接收数据的当前长度 (TRCV)。</li> </ul>
0	7000	<ul style="list-style-type: none"> <li>无激活的作业处理 (TSEND)</li> <li>块未准备好接收 (TRCV)</li> </ul>

ERROR	STATUS	说明
0	7001	<ul style="list-style-type: none"> <li>启动作业处理，正在发送数据：在执行此处理期间，操作系统访问 DATA 发送区中的数据 (TSEND)。</li> <li>块准备接收，接收作业已激活 (TRCV)。</li> </ul>
0	7002	<ul style="list-style-type: none"> <li>后续指令执行（与 REQ 无关），正在处理作业：在执行此处理期间，操作系统访问 DATA 发送区中的数据 (TSEND)。</li> <li>后续指令执行，正在处理接收作业：数据在执行此处理期间写入接收区。因此，错误可能导致接收区中的数据不一致 (TRCV)。</li> </ul>
1	8085	<ul style="list-style-type: none"> <li>LEN 参数的值大于允许的最大值 (TSEND) 和 (TRCV)。</li> <li>自第一次指令执行 (TRCV) 以来，LEN 或 DATA 参数发生变化。</li> </ul>
1	8086	ID 参数不在允许的地址范围内。
1	8088	LEN 参数大于 DATA 中指定的存储区。
1	80A1	通信错误： <ul style="list-style-type: none"> <li>尚未建立指定的连接 (TSEND 和 TRCV)。</li> <li>当前正在终止指定的连接。无法通过该连接执行传送或接收作业 (TSEND 和 TRCV)。</li> <li>正在重新初始化接口 (TSEND)。</li> <li>接口正在接收新参数 (TRCV)。</li> </ul>
1	80C3	内部缺乏资源：具有该 ID 的块正在一个具有不同优先级的组中处理。
1	80C4	临时通信错误： <ul style="list-style-type: none"> <li>此时无法建立与通信伙伴的连接。</li> <li>接口正在接收新参数设置或当前正在建立连接。</li> </ul>

### 以太网连接协议

每个 CPU 都集成了一个支持标准 PROFINET 通信的 PROFINET 端口。TSEND\_C、TRCV\_C、TSEND 和 TRCV 指令均支持 TCP 和 ISO-on-TCP 以太网协议。

更多相关信息，请参见“设备组态：组态本地/伙伴连接路径 (页 129)”。

### 参见

PROFINET 连接的参数 (页 132)

### 10.2.2.5 UDP

UDP 是由 RFC 768 描述的一种标准协议：用户数据报协议。UDP 提供了一种一个应用程序向另一个应用程序发送数据报可采用的机制；但是，数据的传输得不到保证。该协议有以下特点：

- 由于该协议与硬件紧密相关，因此它是一种快速通信协议
- 适合用于小数据量到中等数据量（最多 2048 字节）
- UDP 是比 TCP 更加简单的传输控制协议，其薄层占用资源非常少
- 可以非常灵活地与许多第三方系统一起使用
- 有路由功能
- 使用端口号指引数据报
- 不确认消息：需要负责错误恢复和安全性的应用程序
- 由于使用 SEND/RECEIVE 编程接口的缘故，需要编程来进行数据管理

UDP 支持广播通信。要使用广播，必须组态 ADDR 组态的 IP 地址部分。例如：IP 地址为 192.168.2.10、子网掩码为 255.255.255.0 的 CPU 将使用广播地址 192.168.2.255。

### TUSEND 和 TURCV

以下指令控制 UDP 通信过程：

- TCON 在客户机与服务器 (CPU) PC 之间建立通信连接。
- TUSEND 和 TURCV 发送和接收数据。
- TDISCON 断开客户机与服务器之间的通信。

有关 TCON 和 TDISCON 通信指令的更多信息，请参见“TCP 和 ISO-on-TCP”部分中的 TCON、TDISCON、TSEND 和 TRCV (页 473)。

表格 10- 15 TUSEND 和 TURCV 指令

LAD/FBD	SCL	说明
	<pre>"TUSEND_DB" (   req:=_bool_in_,   ID:=_word_in_,   len:=_uint_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   data:=_variant_inout_);</pre>	<p>TUSEND 指令通过 UDP 将数据发送到参数 ADDR 指定的远程伙伴。</p> <p>要启动用于发送数据的作业，请调用 REQ = 1 的 TUSEND 指令。</p>
	<pre>"TURCV_DB" (   en_r:=_bool_in_,   ID:=_word_in_,   len:=_uint_in_,   ndr=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   rcvd_len=&gt;_uint_out_,   data:=_variant_inout_);</pre>	<p>TURCV 指令通过 UDP 接收数据。参数 ADDR 显示发送方地址。TURCV 成功完成后，参数 ADDR 将包含远程伙伴（发送方）的地址。</p> <p>TURCV 不支持特殊模式。</p> <p>要启动用于接收数据的作业，请调用 EN_R = 1 的 TURCV 指令。</p>

1 STEP 7 会在插入指令时自动创建 DB。

TCON、TDISCON、TUSEND 和 TURCV 异步运行，即，作业处理需要多次执行指令来完成。

表格 10- 16 TUSEND 和 TURCV 参数的数据类型

参数和类型	数据类型	说明	
REQ (TUSEND)	IN	Bool	在上升沿启动发送作业。传送通过 DATA 和 LEN 指定的区域中的数据。
EN_R (TURCV)	IN	Bool	<ul style="list-style-type: none"> <li>0: CPU 无法接收。</li> <li>1: 允许 CPU 进行接收。TURCV 指令准备接收，并处理接收作业。</li> </ul>
ID	IN	Word	引用用户程序与操作系统通信层之间的相关连接。ID 必须与本地连接描述中的相关参数 ID 相同。 值范围：W#16#0001 到 W#16#0FFF。
LEN	IN	UInt	要发送 (TUSEND) 或接收 (TURCV) 的字节数。 <ul style="list-style-type: none"> <li>默认值为 0。DATA 参数确定要发送或接收的数据长度。</li> <li>否则为值范围：1 到 1472</li> </ul>

参数和类型		数据类型	说明
DONE (TUSEND)	IN	Bool	状态参数 DONE (TUSEND): <ul style="list-style-type: none"> <li>0: 作业尚未开始或仍在运行。</li> <li>1: 作业无错完成。</li> </ul>
NDR (TURCV)	OUT	Bool	状态参数 NDR (TURCV): <ul style="list-style-type: none"> <li>0: 作业尚未开始或仍在运行。</li> <li>1: 作业已成功完成。</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>1: 作业尚未完成。无法触发新作业。</li> <li>0: 作业已完成。</li> </ul>
ERROR	OUT	Bool	状态参数，可具有以下值: <ul style="list-style-type: none"> <li>0: 无错误</li> <li>1: 处理时出错。STATUS 提供错误类型的详细信息。</li> </ul>
STATUS	OUT	Word	包括错误信息的状态信息。（请参见下表中的错误和状态条件代码。）
DATA	IN_OUT	Variant	发送区 (TUSEND) 或接收区 (TURCV) 的地址: <ul style="list-style-type: none"> <li>过程映像输入表</li> <li>过程映像输出表</li> <li>存储器位</li> <li>数据块</li> </ul>
ADDR	IN_OUT	Variant	指向接收方（对于 TUSEND）或发送方（对于 TURCV）的地址的指针（例如，P#DB100.DBX0.0 byte 8）。该指针可指向任何存储区。 需要 8 字节的结构，具体如下: <ul style="list-style-type: none"> <li>前 4 个字节包含远程 IP 地址。</li> <li>接下来的 2 个字节指定远程端口号。</li> <li>最后 2 个字节保留。</li> </ul>

作业状态由输出参数 BUSY 和 STATUS 指示。STATUS 与以异步方式工作的指令的 RET\_VAL 输出参数一致。

下表给出了 BUSY、DONE (TUSEND)、NDR (TURCV) 和 ERROR 之间的关系。通过该表格，用户可以确定指令（TUSEND 或 TURCV）的当前状态或者发送（传送）/接收过程完成的时间。

表格 10- 17 BUSY、DONE (TUSEND)/NDR (TURCV) 和 ERROR 参数的状态

BUSY	DONE / NDR	ERROR	说明
TRUE	不相关	不相关	正在处理作业。
FALSE	TRUE	FALSE	作业已成功完成。
FALSE	FALSE	TRUE	作业因错结束。 错误原因可在 STATUS 参数中找到。
FALSE	FALSE	FALSE	未给该指令分配（新）作业。

- 1 由于指令以异步方式工作：对于 TUSEND，在 DONE 参数值或 ERROR 参数值为 TRUE 前，必须保持发送方区域中的数据一致。对于 TURCV，仅当 NDR 参数值为 TRUE 时，接收方区域中的数据才一致。

表格 10- 18 TUSEND 和 TURCV 指令的 ERROR 和 STATUS 的条件代码

ERROR	STATUS	说明
0	0000	<ul style="list-style-type: none"> <li>发送作业无错完成 (TUSEND)。</li> <li>接受了新数据。在 RCVD_LEN 中显示已接收数据的当前长度 (TURCV)。</li> </ul>
0	7000	<ul style="list-style-type: none"> <li>无激活的作业处理 (TUSEND)</li> <li>块未准备好接收 (TURCV)</li> </ul>
0	7001	<ul style="list-style-type: none"> <li>启动作业处理，正在发送数据 (TUSEND)：在执行此处理期间，操作系统访问 DATA 发送区中的数据。</li> <li>块准备接收，接收作业已激活 (TURCV)。</li> </ul>
0	7002	<ul style="list-style-type: none"> <li>后续指令执行（与 REQ 无关），正在处理作业 (TUSEND)：在执行此处理期间，操作系统访问 DATA 发送区中的数据。</li> <li>后续指令执行，正在处理作业：在执行此处理期间，TURCV 指令将数据写入接收区。因此，错误可能导致接收区中的数据不一致。</li> </ul>
1	8085	LEN 参数值大于最大允许值，其值为 0 (TUSEND)，或者自第一次执行指令 (TURCV) 以来更改了 LEN 或 DATA 参数的值。
1	8086	ID 参数不在允许的地址范围内。
1	8088	<ul style="list-style-type: none"> <li>LEN 参数大于 DATA 中指定的存储区 (TUSEND) 或接收区 (TURCV)。</li> <li>接收区过小 (TURCV)。</li> </ul>
1	8089	ADDR 参数未指向数据块。



ERROR	STATUS	说明
1	80A1	通信错误： <ul style="list-style-type: none"> <li>• 尚未建立用户程序和操作系统通信层之间的指定连接。</li> <li>• 当前正在终止用户程序和操作系统通信层之间的指定连接。无法通过该连接执行传送 (TUSEND) 或接收作业 (TURCV)。</li> <li>• 正在重新初始化接口。</li> </ul>
1	80A4	远程连接端点的 IP 地址无效；可能与本地 IP 地址匹配 (TUSEND)。
1	80B3	<ul style="list-style-type: none"> <li>• 设置的协议（连接说明中的 <code>connection_type</code> 参数）不是 UDP。请使用 TSEND 或 TRCV 指令。</li> <li>• ADDR 参数：端口号的设置无效 (TUSEND)</li> </ul>
1	80C3	<ul style="list-style-type: none"> <li>• 具有该 ID 的块正在一个具有不同优先级的组中处理。</li> <li>• 内部缺乏资源</li> </ul>
1	80C4	临时通信错误： <ul style="list-style-type: none"> <li>• 此时无法建立用户程序和操作系统通信层之间的连接 (TUSEND)。</li> <li>• 接口正在接收新参数 (TUSEND)。</li> <li>• 当前正在重新启动连接 (TURCV)。</li> </ul>

### 以太网连接协议

每个 CPU 都集成了一个支持标准 PROFINET 通信的 PROFINET 端口。TUSEND 和 TURCV 指令支持 UDP 以太网协议。

更多相关信息，请参见“设备配置”一章中的“组态本地/伙伴连接路径”（页 129）。

操作

两个伙伴在 UDP 通信中均为被动方。下表给出“TCON\_Param”数据类型的典型参数起始值。端口号 (LOCAL\_TSAP\_ID) 以 2 字节形式表示。允许使用除 161、34962、34963 和 34964 外的所有端口。

表格 10- 19 "TCON\_Param"数据类型参数值

TCON 指令	TCON“UDP Conn DB”																																																																																					
	<table border="1"> <thead> <tr> <th colspan="5">UDP_Conn_DB</th> </tr> <tr> <th>名称</th> <th>数据类型</th> <th>偏移量</th> <th>启动值</th> <th>注释</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Static</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>TCON_Param</td> <td>0.0</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>BLOCK_LENGTH</td> <td>UInt</td> <td>0.0</td> <td>64 byte length of SDT</td> </tr> <tr> <td>4</td> <td>ID</td> <td>CONN_OUC</td> <td>2.0</td> <td>1 reference to the connection</td> </tr> <tr> <td>5</td> <td>CONNECTION_TYPE</td> <td>USInt</td> <td>4.0</td> <td>19 17: TCMIP, 18: ISO on TCP</td> </tr> <tr> <td>6</td> <td>ACTIVE_EST</td> <td>Bool</td> <td>5.0</td> <td>false active/passive connection establishment</td> </tr> <tr> <td>7</td> <td>LOCAL_DEVICE_ID</td> <td>USInt</td> <td>6.0</td> <td>1 local IE interface</td> </tr> <tr> <td>8</td> <td>LOCAL_TSAP_ID_LEN</td> <td>USInt</td> <td>7.0</td> <td>2 byte length of local TSAP id/port number</td> </tr> <tr> <td>9</td> <td>REM_SUBNET_ID_LEN</td> <td>USInt</td> <td>8.0</td> <td>0 byte length of remote subnet id</td> </tr> <tr> <td>10</td> <td>REM_STADDR_LEN</td> <td>USInt</td> <td>9.0</td> <td>0 byte length of remote IP address</td> </tr> <tr> <td>11</td> <td>REM_TSAP_ID_LEN</td> <td>USInt</td> <td>10.0</td> <td>0 byte length of remote port/TSAP id</td> </tr> <tr> <td>12</td> <td>NEXT_STADDR_LEN</td> <td>USInt</td> <td>11.0</td> <td>0 byte length of next station address</td> </tr> <tr> <td>13</td> <td>LOCAL_TSAP_ID</td> <td>Array[1..16] of Byte</td> <td>12.0</td> <td></td> </tr> <tr> <td>14</td> <td>LOCAL_TSAP_ID[1]</td> <td>Byte</td> <td></td> <td>B#16#07 TSAP id/local port number</td> </tr> <tr> <td>15</td> <td>LOCAL_TSAP_ID[2]</td> <td>Byte</td> <td></td> <td>B#16#D0</td> </tr> </tbody> </table>	UDP_Conn_DB					名称	数据类型	偏移量	启动值	注释	1	Static				2	TCON_Param	0.0			3	BLOCK_LENGTH	UInt	0.0	64 byte length of SDT	4	ID	CONN_OUC	2.0	1 reference to the connection	5	CONNECTION_TYPE	USInt	4.0	19 17: TCMIP, 18: ISO on TCP	6	ACTIVE_EST	Bool	5.0	false active/passive connection establishment	7	LOCAL_DEVICE_ID	USInt	6.0	1 local IE interface	8	LOCAL_TSAP_ID_LEN	USInt	7.0	2 byte length of local TSAP id/port number	9	REM_SUBNET_ID_LEN	USInt	8.0	0 byte length of remote subnet id	10	REM_STADDR_LEN	USInt	9.0	0 byte length of remote IP address	11	REM_TSAP_ID_LEN	USInt	10.0	0 byte length of remote port/TSAP id	12	NEXT_STADDR_LEN	USInt	11.0	0 byte length of next station address	13	LOCAL_TSAP_ID	Array[1..16] of Byte	12.0		14	LOCAL_TSAP_ID[1]	Byte		B#16#07 TSAP id/local port number	15	LOCAL_TSAP_ID[2]	Byte		B#16#D0
UDP_Conn_DB																																																																																						
名称	数据类型	偏移量	启动值	注释																																																																																		
1	Static																																																																																					
2	TCON_Param	0.0																																																																																				
3	BLOCK_LENGTH	UInt	0.0	64 byte length of SDT																																																																																		
4	ID	CONN_OUC	2.0	1 reference to the connection																																																																																		
5	CONNECTION_TYPE	USInt	4.0	19 17: TCMIP, 18: ISO on TCP																																																																																		
6	ACTIVE_EST	Bool	5.0	false active/passive connection establishment																																																																																		
7	LOCAL_DEVICE_ID	USInt	6.0	1 local IE interface																																																																																		
8	LOCAL_TSAP_ID_LEN	USInt	7.0	2 byte length of local TSAP id/port number																																																																																		
9	REM_SUBNET_ID_LEN	USInt	8.0	0 byte length of remote subnet id																																																																																		
10	REM_STADDR_LEN	USInt	9.0	0 byte length of remote IP address																																																																																		
11	REM_TSAP_ID_LEN	USInt	10.0	0 byte length of remote port/TSAP id																																																																																		
12	NEXT_STADDR_LEN	USInt	11.0	0 byte length of next station address																																																																																		
13	LOCAL_TSAP_ID	Array[1..16] of Byte	12.0																																																																																			
14	LOCAL_TSAP_ID[1]	Byte		B#16#07 TSAP id/local port number																																																																																		
15	LOCAL_TSAP_ID[2]	Byte		B#16#D0																																																																																		

TUSEND 指令通过 UDP 将数据发送到“TADDR\_Param”数据类型中指定的远程伙伴。TURCV 指令通过 UDP 接收数据。成功执行 TURCV 指令之后，“TADDR\_Param”数据类型会显示远程伙伴（发送方）的地址。

表格 10- 20 "TADDR\_Param"数据类型参数值

TUSEND 指令	TUSEND“UDP ADDR DB”																																																							
	<table border="1"> <thead> <tr> <th colspan="5">Send_UDP_ADDR</th> </tr> <tr> <th>名称</th> <th>数据类型</th> <th>偏移量</th> <th>启动值</th> <th>注释</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Static</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>TADDR_Param</td> <td>0.0</td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>Array[1..4] of USInt</td> <td>0.0</td> <td></td> <td>remote station address</td> </tr> <tr> <td>4</td> <td>USInt</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td>5</td> <td>USInt</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td>6</td> <td>USInt</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td>7</td> <td>USInt</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td>8</td> <td>USInt</td> <td>4.0</td> <td>0</td> <td>remote port number</td> </tr> <tr> <td>9</td> <td>Word</td> <td>6.0</td> <td>0</td> <td>unused, has to be 0</td> </tr> </tbody> </table>	Send_UDP_ADDR					名称	数据类型	偏移量	启动值	注释	1	Static				2	TADDR_Param	0.0			3	Array[1..4] of USInt	0.0		remote station address	4	USInt		0		5	USInt		0		6	USInt		0		7	USInt		0		8	USInt	4.0	0	remote port number	9	Word	6.0	0	unused, has to be 0
Send_UDP_ADDR																																																								
名称	数据类型	偏移量	启动值	注释																																																				
1	Static																																																							
2	TADDR_Param	0.0																																																						
3	Array[1..4] of USInt	0.0		remote station address																																																				
4	USInt		0																																																					
5	USInt		0																																																					
6	USInt		0																																																					
7	USInt		0																																																					
8	USInt	4.0	0	remote port number																																																				
9	Word	6.0	0	unused, has to be 0																																																				

10.2.2.6 T\_CONFIG

在用户程序中通过 T\_CONFIG 指令可更改 PROFINET 端口的 IP 组态参数。该指令可用于永久更改或设置以下特性：

- 站名
- IP 地址

- 子网掩码
- 路由器地址

### 说明

使用位于“以太网地址”(Ethernet address) 页面的 CPU“属性”(Properties) 中的“使用其它方法设置 IP 地址”(页 492)(Set IP address using a different method) 单选按钮或者在下载程序之后使用“T\_CONFIG”指令，可以在线更改 IP 地址。这种指定 IP 地址的方法仅适用于 CPU。

使用位于“以太网地址”(Ethernet address) 页面的 CPU“属性”(Properties) 中的“使用其它方法设置 PROFINET 设备名称”(页 493)(Set PROFINET device name using a different method) 单选按钮或在下载程序之后使用“T\_CONFIG”指令，可以在线更改 PROFINET 设备的名称。这种指定 PROFINET 设备名称的方法仅适用于 CPU。



### 警告

使用 T\_CONFIG 更改 IP 组态参数之后，CPU 会重启。CPU 将转到 STOP 模式、暖启动，然后返回 RUN 模式。

控制设备在不安全情况下运行时可能会出现故障，从而导致受控设备的意外操作。这种意外运行可能会导致死亡、严重的人员伤害和/或设备损坏。

请确保 CPU 由于 T\_CONFIG 指令执行而重启时，过程会进入安全状态。

表格 10-21 T\_CONFIG 指令

LAD/FBD	SCL	说明
	<pre>"T_CONFIG_DB" (   req:=_bool_in_,   interface:=_word_in_,   conf_Data:=_variant_in_,   done=&gt;_bool_out_,   busy=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_dword_out_,   err_loc=&gt;_word_out_);</pre>	<p>在用户程序中使用 T_CONFIG 指令可更改 IP 组态参数。</p> <p>T_CONFIG 异步运行。执行作业时需要多次调用指令。</p>

表格 10-22 参数的数据类型

参数和类型	数据类型	说明	
REQ	Input	Bool	在上升沿时启动该指令。
INTERFACE	Input	HW_Interface	网络接口的 ID
CONF_DATA	Input	Variant	引用组态数据的结构；CONF_DATA 通过系统数据类型 (SDT) 来定义。

参数和类型		数据类型	说明
DONE	Output	Bool	<ul style="list-style-type: none"> <li>0: 作业尚未启动或仍在运行。</li> <li>1: 作业已无错执行。</li> </ul>
BUSY	Output	Bool	<ul style="list-style-type: none"> <li>0: 作业已完成。</li> <li>1: 作业尚未完成。无法触发新作业。</li> </ul>
ERROR	Output	Bool	状态参数，可具有以下值： <ul style="list-style-type: none"> <li>0: 无错误</li> <li>1: 处理时出错。STATUS 提供错误类型的详细信息。</li> </ul>
STATUS	Output	DWord	包括错误信息的状态信息。（请参见下表中的错误和状态条件代码。）
ERR_LOC	Output	DWord	故障位置（错误参数的 ID 字段和 ID 子字段）

IP 组态信息与上面所述参数 CONF\_DATA 中的 Variant 指针一起存储在 CONF\_DATA 数据块中。T\_CONFIG 指令的成功执行以 IP 组态数据传送到网络接口宣告结束。错误被分配给 STATUS 输出参数。

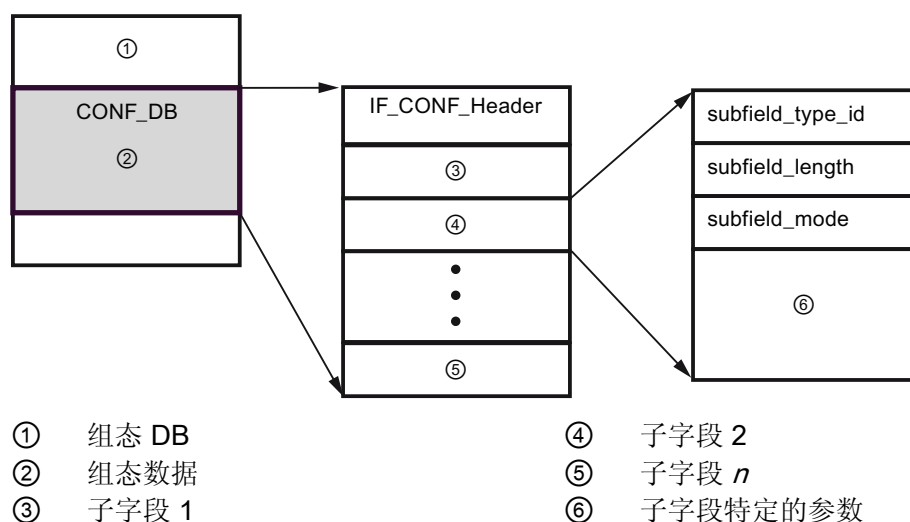
表格 10-23 ERROR 和 STATUS 的条件代码

ERROR	STATUS	说明
0	00000000	无错误
0	00700000	作业尚未完成 (BUSY = 1)。
0	00700100	开始执行作业
0	00700200	临时调用（与 REQ 无关）
1	C08xyy00	常规故障
1	C0808000	接口标识参数 LADDR 无效。
1	C0808100	给接口标识参数 LADDR 分配了不受支持的硬件接口。
1	C0808200	CONF_DATA 参数错误：Variant 指针的数据类型与数据类型 Byte 不匹配。
1	C0808300	CONF_DATA 参数错误：区域指针不在 Variant 指针的 DB 中。
1	C0808400	CONF_DATA 参数错误：Variant 指针的长度错误。
1	C0808600	保留
1	C0808700	CONF_DATA 数据块长度与 IP 组态不一致
1	C0808800	CONF_DATA 数据块 field_type_id 的参数无效。（仅允许 field_type_id = 0。）

ERROR	STATUS	说明
1	C0808900	CONF_DATA 数据块 field_type_id 的参数无效或多次使用。
1	C0808A00	IP 组态参数的 LEN 长度或 subfield_cnt 错误
1	C0808B00	IP 组态 ID 参数无效或不被支持。
1	C0808C00	错误地放置了 IP 组态子块（子块错误、顺序错误或多次使用）。
1	C0808D00	语句 LEN 子块的长度无效。
1	C0808E00	子块模式中的参数的值无效。
1	C0808F00	IP 组态与前一子块之间存在子块冲突。
1	C0809000	子字段的参数是写保护的（例如：通过组态指定参数，或启用了 PNIO 模式）。
1	C0809100	保留
1	C0809400	尚未定义或者可能没有使用 IP 组态子块中的参数。
1	C0809500	IP 组态子块中的参数与其它参数不一致。
1	C080C200	无法执行指令。某些情况下会发生此错误，例如，与接口的通信丢失。
1	C080C300	没有足够的资源。某些情况下会发生此错误，例如，使用不同参数多次调用该指令。
1	C080C400	通信故障。该错误可能会暂时出现，需要重新执行用户程序。
1	C080D200	PROFINET 接口不支持执行该指令。

## CONF\_DATA 数据块

下图显示了待传送的组态数据在组态 DB 中的存储情况。



CONF\_DB 的组态数据由一个包含字段头 (IF\_CONF\_Header) 的字段和多个子字段构成。IF\_CONF\_Header 提供以下元素：

- field\_type\_id (数据类型 UInt) : 零
- field\_id (数据类型 UInt) : 零
- subfield\_cnt (数据类型 UInt) : 子字段数

各子字段又由字段头 (subfield\_type\_id、subfield\_length、subfield\_mode) 和子字段特定的参数组成。各子字段必须由偶数个字节组成。subfield\_mode 支持值 1。

#### 说明

目前仅允许一个字段 (IF\_CONF\_Header)。其参数 field\_type\_id 和 field\_id 的值必须为零。其它具有不同 field\_type\_id 和 field\_id 值的字段用于将来扩展。

在 IF\_CONF\_Header 字段中，目前仅允许两个子字段“addr” (IP 地址) 和“nos” (Name of station)。

表格 10-24 支持的子字段

subfield_type_id	数据类型	说明
30	IF_CONF_V4	IP 参数: IP 地址、子网掩码、路由器地址
40	IF_CONF_NOS	PROFINET IO 设备名称 (Name of station)

表格 10-25 IF\_CONF\_V4 数据类型的元素

名称	数据类型	起始值	说明
Id	UInt	30	subfield_type_id
len	UInt	18	subfield_length
mode	UInt	1	subfield_mode (1: 永久)
InterfaceAddresses	IP_V4	-	接口地址
ADDR	Array [1..4] of Byte		
	ADDR[1]	Byte	b#16#C8 IP 地址高位字节: 200
	ADDR[2]	Byte	b#16#0C IP 地址高位字节: 12
	ADDR[3]	Byte	b#16#01 IP 地址低位字节: 1

名称	数据类型	起始值	说明
ADDR[4]	Byte	b#16#90	IP 地址低位字节: 144
SubnetMask	IP_V4	-	子网掩码
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte	b#16#FF	子网掩码高位字节: 255
ADDR[2]	Byte	b#16#FF	子网掩码高位字节: 255
ADDR[3]	Byte	b#16#FF	子网掩码低位字节: 255
ADDR[4]	Byte	b#16#00	子网掩码低位字节: 0
DefaultRouter	IP_V4	-	默认路由器
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte	b#16#C8	路由器高位字节: 200
ADDR[2]	Byte	b#16#0C	路由器高位字节: 12
ADDR[3]	Byte	b#16#01	路由器低位字节: 1
ADDR[4]	Byte	b#16#01	路由器低位字节: 1

表格 10- 26 IF\_CONF\_NOS 数据类型的元素

名称	数据类型	起始值	说明
id	UInt	40	subfield_type_id
len	UInt	246	subfield_length
mode	UInt	1	subfield_mode (1: 永久)
Nos (Name of station)	Array[1..240] of Byte	0	站名称: 必须从第一个字节开始填充 ARRAY。如果 ARRAY 比要指定的站名称长, 则必须在实际站名称后输入零字节 (符合 IEC 61158-6-10)。否则, 将拒绝 nos, 并且“T_CONFIG (页 486)”指令会在 STATUS 中输入错误代码 DW#16#C0809400。如果用零填充第一个字节, 则将删除站名称。

站名称有以下限制:

- 站名称中的名称部分, 即两个点之间的字符串, 不得超过 63 个字符。
- 不可使用特殊字符, 如元音变音、括号、下划线、斜线、空格等。破折号是唯一允许使用的特殊字符。
- 站名称不得以“-”字符开始或结尾。

- 站名称不得以数字开头。
- 不允许站名称形式 n.n.n.n (n = 0 ... 999)。
- 站名称不得以字符串“port-xyz”或“port-xyz-abcde” (a、b、c、d、e、x、y、z = 0 ... 9) 开头。

**说明**

还可以创建少于 240 个字节的 ARRAY“nos”，但不能少于 2 个字节。在这种情况下，必须相应调整“len”（子字段长度）。

**如何更改 IP 参数**

在以下示例中，更改了“addr”子字段下的 "InterfaceAddress"（IP 地址）、"SubnetMask" 和 "DefaultRouter"（IP 路由器）。要在下载程序之后使用“T\_CONFIG”指令更改 IP 地址，必须在“以太网地址”(Ethernet address) 页面的 CPU“属性”(Properties) 中单击“使用其它方法设置 IP 地址”(Set IP address using a different method) 单选按钮。

表格 10-27 如何更改 IP 参数

CONF_DATA_1			
名称	数据类型	启动值	
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	Fieldid	UInt	0
6	SubfieldCount	UInt	1
7	addr	IF_CONF_v4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	ADDR[1]	Byte	192
14	ADDR[2]	Byte	168
15	ADDR[3]	Byte	2
16	ADDR[4]	Byte	30
17	SubnetMask	IP_V4	
18	ADDR	array [1..4] of Byte	
19	ADDR[1]	Byte	255
20	ADDR[2]	Byte	255
21	ADDR[3]	Byte	255
22	ADDR[4]	Byte	0
23	DefaultRouter	IP_V4	
24	ADDR	array [1..4] of Byte	
25	ADDR[1]	Byte	192
26	ADDR[2]	Byte	168
27	ADDR[3]	Byte	2
28	ADDR[4]	Byte	1





## 如何更改 IP 参数和 PROFINET IO 设备名称

在以下示例中，更改了“addr”和“nos”(Name of station) 这两个子字段。要在下载程序之后使用“T\_CONFIG”指令更改 PROFINET 设备的名称，必须在“以太网地址”(Ethernet address) 页面的 CPU“属性”(Properties) 中单击“使用其它方法设置 PROFINET 设备名称”(Set PROFINET device name using a different method) 单选按钮。

表格 10-28 如何更改 IP 参数和 PROFINET IO 设备名称

CONF_DATA_2			
	名称	数据类型	启动值
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	FieldId	UInt	0
6	SubfieldCount	UInt	2
7	addr	IF_CONF_v4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	Subnetmask	IP_V4	
14	ADDR	array [1..4] of Byte	
15	DefaultRouter	IP_V4	
16	ADDR	array [1..4] of Byte	
17	nos	IF_CONF_NIOS	
18	Id	UInt	40
19	Length	UInt	246
20	Mode	UInt	1
21	NOS	array [1..240] of Byte	



### 10.2.2.7 指令的公共参数

#### REQ 输入参数

许多开放式用户通信指令使用 REQ 输入在由低电平向高电平切换时启动操作。REQ 输入在指令执行一次的时间内必须为高电平 (TRUE)，不过 REQ 输入可以在所需时间内一直保持为 TRUE。在 REQ 输入为 FALSE 时执行指令以便能复位 REQ 输入的历史状态之前，该指令不会启动其它操作。只有这样，指令才能检测低电平到高电平的跳变以启动下一个操作。

在程序中放置这些指令之一后，STEP 7 会提示用户指定背景数据块。对每个指令调用使用一个唯一的背景数据块。这样可确保每个指令都能正确地处理诸如 REQ 等输入。

## ID 输入参数

这是对 STEP 7 中“设备和网络”(Devices and networks) 的“网络视图”(Network view) 中的“本地 ID (十六进制)”(Local ID (hex)) 的引用，并且是要用于该通信块的网络的 ID。ID 必须与本地连接描述中的相关参数 ID 相同。

## DONE、NDR、ERROR 和 STATUS 输出参数

这些指令提供说明完成状态的输出：

表格 10- 29 开放式用户通信指令输出参数

参数	数据类型	默认值	说明
DONE	Bool	FALSE	设置为 TRUE 并持续执行一次所需的时间，以表明上一请求已经完成且没有出现错误；否则为 FALSE。
NDR	Bool	FALSE	设置为 TRUE 并持续执行一次所需的时间，以表明请求的动作已经完成且没有出现错误并已接收新的数据；否则为 FALSE。
BUSY	Bool	FALSE	激活时设置为 TRUE 以表明： <ul style="list-style-type: none"> <li>• 作业尚未完成。</li> <li>• 无法触发新作业。</li> </ul> 作业完成时设置为 FALSE。
ERROR	Bool	FALSE	设置为 TRUE 并持续执行一次所需的时间，以表明上一请求已经完成但出现了错误，相应的错误代码在 STATUS 中；否则为 FALSE。
STATUS	Word	0	结果状态： <ul style="list-style-type: none"> <li>• 如果设置了 DONE 或 NDR 位，则 STATUS 被设置为 0 或信息代码。</li> <li>• 如果设置了 ERROR 位，则 STATUS 被设置为一个错误代码。</li> <li>• 如果没有设置以上任何一位，则指令会返回说明功能当前状态的状态结果。</li> </ul> STATUS 在该功能执行期间一直保持其值。

## 说明

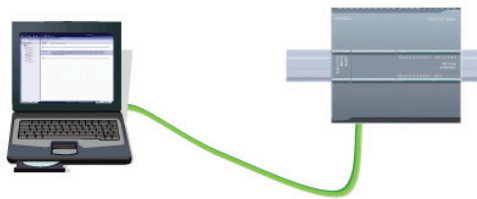
请注意，DONE、NDR 和 ERROR 仅置位一个执行周期的时间。

### 被动 ISO 和 TCP 通信的 TSAP 和端口号限制

如果使用“TCON”指令设置并建立被动通信连接，则下列端口地址将受到限制，不应该使用：

- ISO TSAP（被动）：
  - 01.00, 01.01, 02.00, 02.01, 03.00, 03.01
  - 10.00, 10.01, 11.00, 11.01, ... BF.00、BF.01
- TCP 端口（被动）： 5001, 102, 123, 20, 21, 25, 34962, 34963, 34964, 80
- UDP 端口（被动）： 161, 34962, 34963, 34964

## 10.2.3 与编程设备通信



CPU 可以与网络上的 STEP 7 编程设备进行通信。

在 CPU 和编程设备之间建立通信时请考虑以下几点：

- 组态/设置： 需要进行硬件配置。
- 一对一通信不需要以太网交换机；网络中有两个以上的设备时需要以太网交换机。

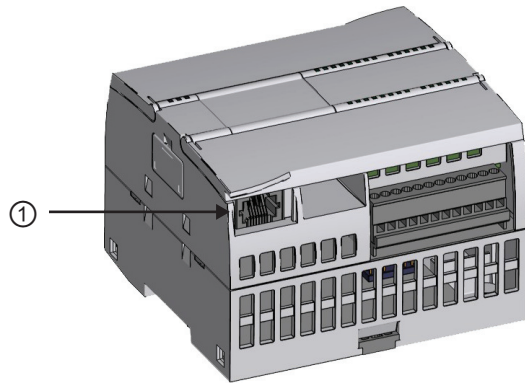
### 10.2.3.1 建立硬件通信连接

PROFINET 接口可在编程设备和 CPU 之间建立物理连接。由于 CPU 内置了自动跨接功能，所以对该接口既可以使用标准以太网电缆，又可以使用跨接以太网电缆。将编程设备直接连接到 CPU 时不需要以太网交换机。

要在编程设备和 CPU 之间创建硬件连接，请按以下步骤操作：

1. 安装 CPU (页 49)。

2. 将以太网电缆插入下图所示的 PROFINET 端口中。
3. 将以太网电缆连接到编程设备上。



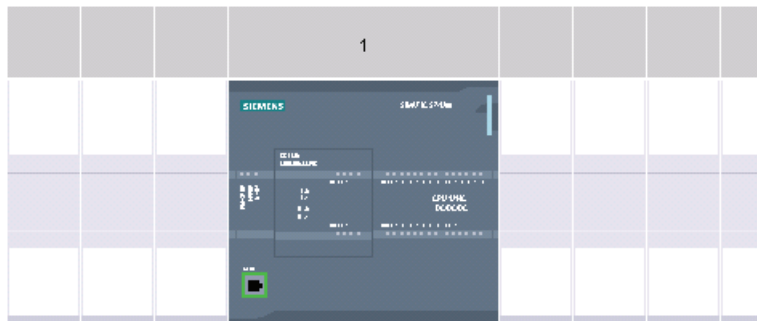
① PROFINET 端口

可选配张力消除装置以加固 PROFINET 连接。

### 10.2.3.2 配置设备

如果已使用 CPU 创建项目，则在 STEP 7 中打开项目。

如果没有，请创建项目并在机架中插入 CPU (页 122)。在下面的项目中，“设备视图”(Device View) 中显示了 CPU。



### 10.2.3.3 分配 Internet 协议 (IP) 地址

#### 分配 IP 地址

在 PROFINET 网络中，每个设备还必须具有一个 Internet 协议 (IP) 地址。该地址使设备可以在更加复杂的路由网络中传送数据：

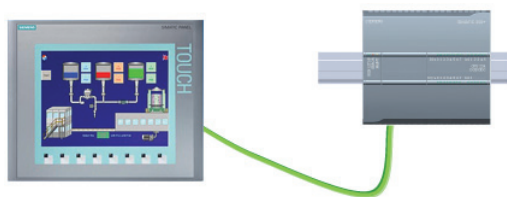
- 如果用户具有使用连接到工厂 LAN 的板载适配器卡或连接到独立网络的以太网转 USB 适配器卡的编程设备或其它网络设备，则必须为它们分配 IP 地址。更多相关信息，请参见“为编程设备和网络设备分配 IP 地址”(页 135)。
- 还可以在线为 CPU 或网络设备分配 IP 地址。这在进行初始设备配置时尤其有用。更多相关信息，请参见“在线为 CPU 分配 IP 地址”(页 135)。
- 组态项目中的 CPU 或网络设备后，可以组态 PROFINET 接口的参数及其 IP 地址。更多相关信息，请参见“为项目中的 CPU 组态 IP 地址”(页 138)。

### 10.2.3.4 测试 PROFINET 网络

完成组态后，必须将项目下载到 CPU 中。下载项目时会组态所有 IP 地址。

CPU“下载到设备”(Download to device) 功能及其“扩展的下载到设备”(Extended download to device) 对话框可以显示所有可访问的网络设备，以及是否为所有设备都分配了唯一的 IP 地址。更多相关信息，请参见“测试 PROFINET 网络”(页 142)

## 10.2.4 HMI 到 PLC 通信



CPU 支持通过 PROFINET 端口与 HMI 通信。设置 CPU 和 HMI 之间的通信时必须考虑以下要求：

组态/设置：

- 必须组态 CPU 的 PROFINET 端口与 HMI 连接。
- 必须已设置和组态 HMI。

- HMI 组态信息是 CPU 项目的一部分，可以在项目内部进行组态和下载。
- 一对一通信不需要以太网交换机；网络中有两个以上的设备时需要以太网交换机。

#### 说明

机架安装的 CSM1277 4 端口以太网交换机可用于连接 CPU 和 HMI 设备。CPU 上的 PROFINET 端口不包含以太网交换设备。

支持的功能：

- HMI 可以对 CPU 读/写数据。
- 可基于从 CPU 重新获取的信息触发消息。
- 系统诊断

表格 10- 30 组态 HMI 与 CPU 之间的通信时所需的步骤

步骤	任务
1	<p>建立硬件通信连接</p> <p>通过 PROFINET 接口建立 HMI 和 CPU 之间的物理连接。由于 CPU 内置了自动跨接功能，所以对该接口既可以使用标准以太网电缆，又可以使用跨接以太网电缆。连接一个 HMI 和一个 CPU 不需要以太网交换机。</p> <p>更多相关信息，请参见“与编程设备通信： 建立硬件通信连接” (页 495)。</p>
2	<p>配置设备</p> <p>更多相关信息，请参见“与编程设备通信： 组态设备” (页 496)。</p>
3	<p>组态 HMI 与 CPU 之间的逻辑网络连接</p> <p>更多相关信息，请参见“HMI 与 PLC 通信： 组态两个设备之间的逻辑网络连接” (页 498)。</p>
4	<p>在项目中组态 IP 地址</p> <p>使用相同的组态过程；但必须为 HMI 和 CPU 组态 IP 地址。</p> <p>更多相关信息，请参见“设备配置： 为项目中的 CPU 组态 IP 地址” (页 139)。</p>
5	<p>测试 PROFINET 网络</p> <p>必须为每个 CPU 和 HMI 设备都下载相应的组态。</p> <p>更多相关信息，请参见“设备配置： 测试 PROFINET 网络” (页 142)。</p>

#### 10.2.4.1 组态两个设备之间的逻辑网络连接

使用 CPU 配置机架后，您即准备好组态网络连接。

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 ISO-on-TCP 连接）。

要创建 PROFINET 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。

有关详细信息，请参见“设备配置：创建网络连接”(页 128)。

## 10.2.5 PLC 到 PLC 通信



通过使用 TSEND\_C 和 TRCV\_C 指令，一个 CPU 可与网络中的另一个 CPU 进行通信。

设置两个 CPU 之间的通信时必须考虑以下事宜：

- 组态/设置：需要进行硬件配置。
- 支持的功能：向对等 CPU 读/写数据
- 一对一通信不需要以太网交换机；网络中有两个以上的设备时需要以太网交换机。

表格 10-31 组态两个 CPU 之间的通信时所需的步骤

步骤	任务
1	<p>建立硬件通信连接</p> <p>通过 PROFINET 接口建立两个 CPU 之间的物理连接。由于 CPU 内置了自动跨接功能，所以对该接口既可以使用标准以太网电缆，又可以使用跨接以太网电缆。连接两个 CPU 时不需要以太网交换机。</p> <p>更多相关信息，请参见“与编程设备通信：建立硬件通信连接”(页 495)。</p>
2	<p>配置设备</p> <p>必须组态项目中的两个 CPU。</p> <p>更多相关信息，请参见“与编程设备通信：组态设备”(页 496)。</p>
3	<p>组态两个 CPU 之间的逻辑网络连接</p> <p>更多相关信息，请参见“PLC 与 PLC 通信：组态两个设备之间的逻辑网络连接”(页 500)。</p>

步骤	任务
4	在项目中组态 IP 地址 使用相同的组态过程；但必须为两个 CPU（例如，PLC_1 和 PLC_2）组态 IP 地址。 更多相关信息，请参见“设备配置：为项目中的 CPU 组态 IP 地址”（页 139）。
5	组态传送（发送）和接收参数 必须在两个 CPU 中均组态 TSEND_C 和 TRCV_C 指令，才能实现两个 CPU 之间的通信。 更多相关信息，请参见“组态两个 CPU 之间的通信：组态传送（发送）和接收参数”（页 501）。
6	测试 PROFINET 网络 必须为每个 CPU 都下载相应的组态。 更多相关信息，请参见“设备配置：测试 PROFINET 网络”（页 142）。

### 10.2.5.1 组态两个设备之间的逻辑网络连接

使用 CPU 配置机架后，您即准备好组态网络连接。

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 ISO-on-TCP 连接）。

要创建 PROFINET 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。

有关详细信息，请参见“设备配置：创建网络连接”（页 128）。

### 10.2.5.2 组态两台设备间的本地/伙伴连接路径

#### 组态常规参数

在通信指令的“属性”(Properties) 组态对话框中指定通信参数。只要选中了该指令的任何一部分，此对话框就会出现在页面底部附近。

更多相关信息，请参见“设备配置：组态本地/伙伴连接路径”（页 129）。

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 或端口。在“本地 TSAP”(Local TSAP) 字段中输入 CPU 中连接的 TSAP 或端口。在“伙伴 TSAP”(Partner TSAP) 字段下输入为伙伴 CPU 中的连接分配的 TSAP 或端口。



### 10.2.5.3 组态传送（发送）和接收参数

通信块（例如 TSEND\_C 和 TRCV\_C）用于建立两个 CPU 之间的连接。在 CPU 可进行 PROFINET 通信前，必须组态传送（或发送）消息和接收消息的参数。这些参数决定了在向目标设备传送消息或从目标设备接收消息时的通信工作方式。

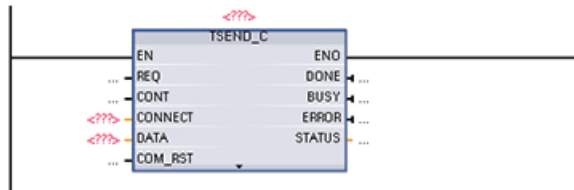
#### 组态 TSEND\_C 指令传送（发送）参数

#### TSEND\_C 指令

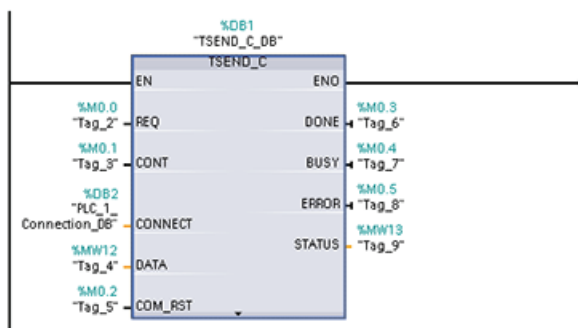
TSEND\_C 指令 (页 466) 可创建与伙伴站的通信连接。通过该指令可设置和建立连接，并会在通过指令断开连接前一直自动监视该连接。TSEND\_C 指令兼具 TCON、TDISCON 和 TSEND 指令的功能。

通过 STEP 7 中的设备配置，可以组态 TSEND\_C 指令传送数据的方式。首先，从“通信”(Communications) 文件夹的“指令”(Instructions) 任务卡中将该指令插入程序中。

TSEND\_C 指令将与“调用选项”(Call options) 对话框一起显示，在该对话框中可以分配用于存储该指令参数的 DB。



可以为输入和输出分配变量存储位置，如下图所示：



### 组态常规参数

在 TSEND\_C 指令的“属性组态”(Properties configuration) 对话框中指定通信参数。只要选中了 TSEND\_C 指令的任何一部分，此对话框就会出现在页面底部附近。

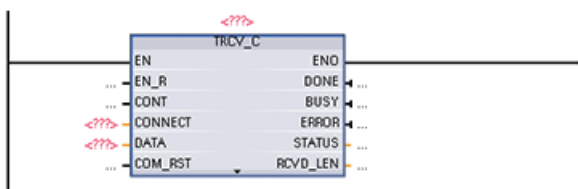
### 组态 TRCV\_C 指令接收参数

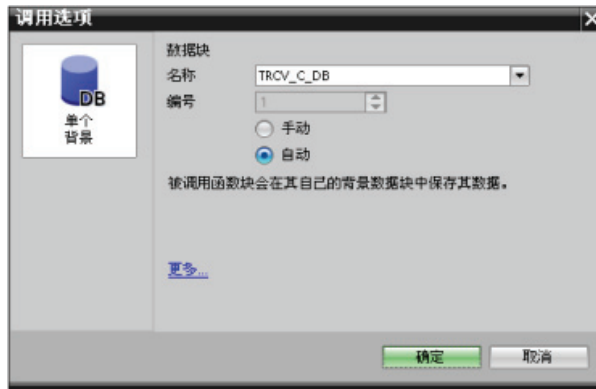
### TRCV\_C 指令

TRCV\_C 指令 (页 466)可创建与伙伴站的通信连接。通过该指令可设置和建立连接，并会在通过指令断开连接前一直自动监视该连接。TRCV\_C 指令兼具 TCON、TDISCON 和 TRCV 指令的功能。

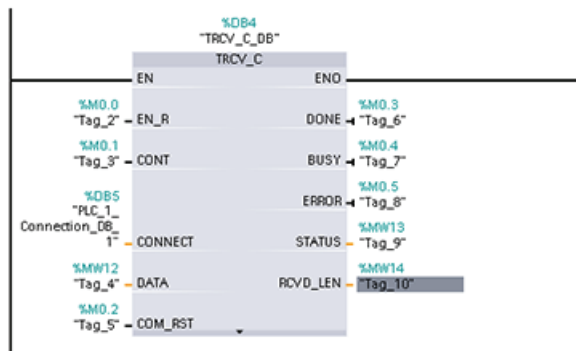
通过 STEP 7 中的 CPU 组态，可以组态 TRCV\_C 指令接收数据的方式。首先，从“通信”(Communications) 文件夹的“指令”(Instructions) 任务卡中将该指令插入程序中。

TRCV\_C 指令将与“调用选项”(Call options) 对话框一起显示，在该对话框中可以分配用于存储该指令参数的 DB。





可以为输入和输出分配变量存储位置，如下图所示：



## 组态常规参数

在 TRCV\_C 指令的“属性组态”(Properties configuration) 对话框中指定通信参数。只要选中了 TRCV\_C 指令的任何一部分，此对话框就会出现在页面底部附近。

## 10.2.6 配置 CPU 和 PROFINET IO 设备

### 添加 PROFINET IO 设备



在“设备和网络”(Devices and Networks) 门户中，使用硬件目录添加 PROFINET IO 设备。

#### 说明

要添加一个 PROFINET IO 设备，可以使用 STEP 7 Professional/Basic, V11 或更高版本。

例如，在硬件目录中展开下列容器以添加 ET200S IO 设备：分布式 I/O、ET200S、接口模块和 PROFINET。然后可从 ET200S 设备（按零件号排序）列表中选择接口模块和添加 ET200S IO 设备。

表格 10- 32 向设备组态添加 ET200S IO 设备

插入 IO 设备	结果
	

现在可将 PROFINET IO 设备连接到 CPU：

1. 右键单击设备上的“未分配”(Not assigned) 链接，然后从上下文菜单中选择“分配新的 IO 控制器”(Assign new IO controller) 以显示“选择 IO 控制器”(Select IO controller) 对话框。
2. 从项目的 IO 控制器列表中选择 S7-1200 CPU（本例中为“PLC\_1”）。
3. 单击“确定”(OK) 创建网络连接。

## 组态逻辑网络连接

使用 CPU 配置机架后，您即准备好组态网络连接。

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。要创建 PROFINET 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“设备组态：创建网络连接”(页 128)。

## 分配 CPU 和设备名称

设备之间的网络连接还会将 PROFINET IO 设备分配给 CPU，从而 CPU 能够控制相应设备。要更改该分配，请单击 PROFINET IO 设备上显示的“PLC 名称”(PLC Name)。将打开一个对话框，允许用户从当前 CPU 上断开 PROFINET IO 设备以重新分配设备，或根据需保持不分配状态。

PROFINET 网络中的设备在分配名称后才可与 CPU 连接。如果 PROFINET 设备尚未分配名称，或要更改该设备的名称，则可使用“网络视图”(Network view) 为 PROFINET 设

备分配名称。可通过右键单击 PROFINET IO 设备并选择“分配设备名称”(Assign device name) 来实现。

对于各 PROFINET IO 设备，必须在 STEP 7 项目（使用“在线和诊断”(Online & diagnostics) 工具）和 PROFINET IO 设备组态存储器（例如 ET200 S 接口模块组态存储器）中为该设备分配相同的名称。如果名称缺失或两个位置中的名称不匹配，则 PROFINET IO 数据交换模式将不会运行。更多相关信息，请参见“在线和诊断工具：在线为 PROFINET 设备分配名称 (页 732)”。

## 分配 IP 地址

在 PROFINET 网络中，每个设备还必须具有一个 Internet 协议 (IP) 地址。该地址使设备可以在更加复杂的路由网络中传送数据：

- 如果用户具有使用连接到工厂 LAN 的板载适配器卡或连接到独立网络的以太网转 USB 适配器卡的编程设备或其它网络设备，则必须为它们分配 IP 地址。更多相关信息，请参见“为编程设备和网络设备分配 IP 地址” (页 135)。
- 还可以在线为 CPU 或网络设备分配 IP 地址。这在进行初始设备配置时尤其有用。更多相关信息，请参见“在线为 CPU 分配 IP 地址” (页 138)。
- 组态项目中的 CPU 或网络设备后，可以组态 PROFINET 接口的参数及其 IP 地址。更多相关信息，请参见“为项目中的 CPU 组态 IP 地址” (页 139)。

## 组态 IO 循环时间

CPU 会在“IO 循环”期间为 PROFINET IO 设备提供新数据。可以单独组态每台设备的更新时间，更新时间可确定在 CPU 和设备之间交换数据的时间间隔。

在 PROFINET 网络上每台设备的默认设置中，由 STEP 7 根据要交换的数据量和分配给控制器的设备数自动计算“IO 循环”更新时间。如果不希望自动计算更新时间，则可以更改此设置。

在 PROFINET IO 设备的“属性”(Properties) 组态对话框中指定“IO 循环”(IO cycle) 参数。只要选中了该指令的任何一部分，此对话框就会出现在页面底部附近。

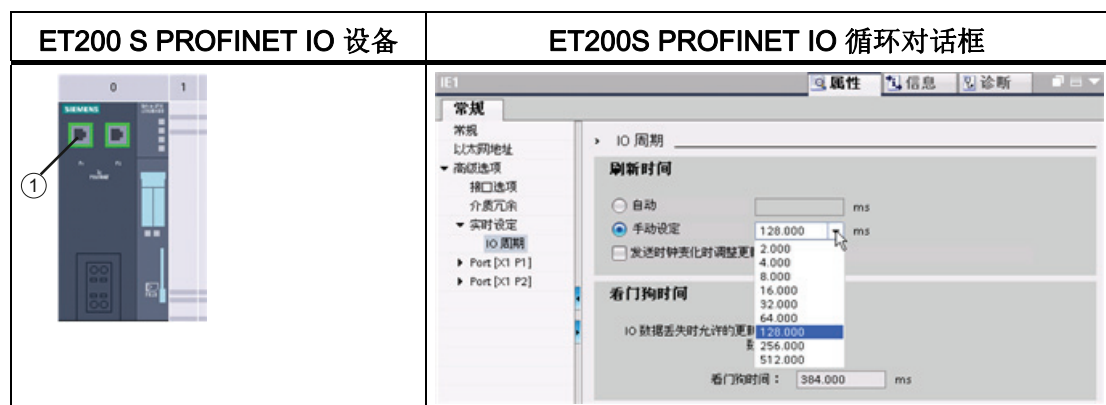
在 PROFINET IO 设备的“设备视图”(Device view) 中单击 PROFINET 端口。在“PROFINET 接口”(PROFINET Interface) 对话框中，通过以下菜单选项访问“IO 循环”(IO cycle) 参数：

- “高级选项”(Advanced options)
- “实时设置”(Realtime settings)
- “IO 循环”(IO cycle)

通过以下选项定义 IO 循环“更新时间”(Update time)：

- 要自动计算合适的更新时间，请选择“自动”(Automatic)。
- 要亲自设置更新时间，请选择“可进行设置”(Can be set) 并输入所需更新时间（单位为 ms）。
- 为了确保发送时钟与更新时间之间的一致性，应激活“发送时钟变化时调整更新时间”(Adapt update time when send clock changes) 选项。该选项可确保不会将更新时间设置为小于发送时钟的时间。

表格 10- 33 组态 ET200S PROFINET IO 循环时间



① PROFINET 端口

## 10.2.7 诊断

### 诊断中断组织块 (OB82)

如果具有诊断功能的模块启用了诊断中断并且检测到其诊断状态发生了变化，则在以下情况下，该模块会向 CPU 发送诊断中断请求：

- 该模块检测到问题（例如断路）或需要对组件进行维护，或两种情况都有（进入事件）。
- 问题已经解决或不复存在，或者没有其它组件需要维护（离开事件）。

如果 OB82 不存在，则将这些错误写入诊断缓冲区。CPU 不会有任何动作，也不会切换到 STOP 模式。

如果 OB82 存在，则操作系统会调用 OB82 以响应进入事件。必须创建 OB82，因为该 OB 可用于组态本地错误处理并可对进入事件作出更加详细的响应。

如果使用具有 DPV1 功能的 CPU，则可以通过 RALRM 指令获取有关中断的更多信息，该指令可提供比 OB82 的启动信息更为具体的信息。

### 外围设备访问报警

这些错误写入诊断缓冲区。CPU 不会有任何动作，也不会切换到 STOP 模式。写入诊断缓冲区的错误包括：

- 模块故障
- 模块不匹配
- 模块丢失

### IO 访问错误

这些错误写入诊断缓冲区。CPU 不会有任何动作，也不会切换到 STOP 模式。

## 10.2.8 分布式 I/O 指令

请参见“分布式 I/O（PROFINET、PROFIBUS 或 AS-i）”（页 290），以了解有关如何将分布式 I/O 指令用于这些通信网络的信息。

### 10.2.9 诊断指令

请参见“诊断（PROFINET 或 PROFIBUS）”：“诊断指令”（页 318），以了解有关如何将诊断指令用于这些通信网络的信息。

### 10.2.10 分布式 I/O 的诊断事件

请参见“诊断（PROFINET 或 PROFIBUS）”：“分布式 I/O 的诊断事件”（页 318），以了解有关如何将该诊断信息用于这些通信网络的信息。

## 10.3 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。

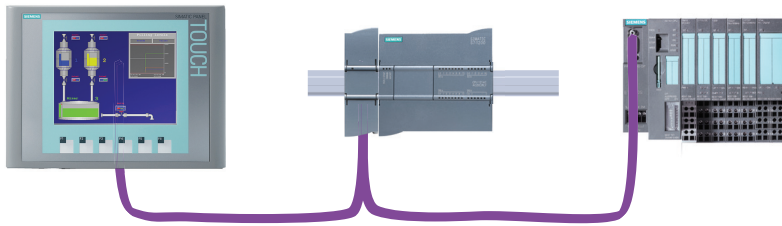
PROFIBUS 主站构成网络的“主动站”。PROFIBUS DP 定义两类主站。第 1 类主站（通常是中央可编程控制器 (PLC) 或运行特殊软件的 PC）处理与分配给它的从站之间的常规通信或数据交换。第 2 类主站（通常是组态设备，如用于调试、维护或诊断的膝上型计算机或编程控制台）是主要用于调试从站和诊断的特殊设备。

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。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 既可充当更高级 DP 主站系统的从站，又可充当更低级 DP 从站系统的主站。



对于 V3.0，每站最多可配置三个 PROFIBUS CM，它们可以是 DP 主站或 DP 从站 CM 的任意组合。V3.0 实现中的每个 DP 主站最多可控制 32 个从站。

对于 V2.2，每个站最多可配置三个 PROFIBUS CM，其中只有一个可为 DP 主站。在 V2.2 实现中，一个 DP 主站最多可控制 16 个从站。

## 10.3.1 通信模块 PROFIBUS

### 10.3.1.1 连接到 PROFIBUS

#### 将 S7-1200 连接到 PROFIBUS DP

可以用以下通信模块将 S7-1200 连接到 PROFIBUS 现场总线系统：

- CM 1242-5  
作为 DP 从站运行
- CM 1243-5  
作为 1 类 DP 主站运行

如果将 CM 1242-5 和 CM 1243-5 安装在一起，则 S7-1200 可以同时执行以下功能：

- 更高级别 DP 主站系统的从站  
和
- 较低级别 DP 主站系统的主站

### 10.3.1.2 PROFIBUS CM 的通信服务

#### 总线协议

PROFIBUS CM 使用 PROFIBUS DP-V1 协议。

#### S7-1200 的 PROFIBUS 通信伙伴

两个 PROFIBUS CM 模块允许 S7-1200 与以下通信伙伴交换数据。

- CM 1242-5  
CM 1242-5 (DP 从站) 可以成为以下 DP V0/V1 主站的通信伙伴：
  - SIMATIC S7-1200、S7-300、S7-400、S7 模块化嵌入式控制器
  - DP 主站模块和分布式 IO SIMATIC ET200
  - SIMATIC PC 站
  - SIMATIC NET IE/PB Link
  - 各家供应商提供的可编程控制器
- CM 1243-5  
CM 1243-5 (DP 主站) 可以成为以下 DP V0/V1 从站的通信伙伴：
  - 分布式 I/O SIMATIC ET200
  - 配备 CM 1242-5 的 S7-1200 CPU
  - 带有 PROFIBUS DP 模块 EM 277 的 S7-200 CPU
  - SINAMICS 变频器
  - 各家供应商提供的驱动器和执行器
  - 各家供应商提供的传感器
  - 具有 PROFIBUS 接口的 S7-300/400 CPU
  - 配备 PROFIBUS CP (例如 CP 342-5) 的 S7-300/400 CPU
  - 配备 PROFIBUS CP 的 SIMATIC PC 站

## 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 不支持功能：

SYNC/FREEZE

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。

### 10.3.1.3 PROFIBUS CM 的其它属性

#### 组态和模块替换

可以在 V11.0 和更高版 STEP 7 中组态模块、网络和连接。

如果要在第三方系统中组态该模块，可以在模块随附的 CD 上或通过 Internet 在西门子自动化客户支持页面上找到 CM 1242-5（DP 从站）的 GSD 文件。

PROFIBUS CM 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这些通信模块。

每个站最多可组态三个 PROFIBUS CM。

### 电气连接

- 电源
  - CM 1242-5 通过 SIMATIC 站的背板总线供电。
  - CM 1243-5 具有一个用来连接 24 V DC 电源的独立连接器。

- PROFIBUS

PROFIBUS 连接器的 RS-485 接口是 9 针 D 型母连接器。

还可以选择通过光总线终端 OBT 或光链接模块 OLM 连接到光纤 PROFIBUS 网络。

### 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 的西门子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:  
49852105 (<http://support.automation.siemens.com/WW/view/zh/49852105>)
- CM 1243-5:  
49851842 (<http://support.automation.siemens.com/WW/view/zh/49851842>)

### 10.3.1.4 PROFIBUS 组态示例

下文给出了将 CM 1242-5 用做 PROFIBUS 从站和将 CM 1243-5 用做 PROFIBUS 主站的组态示例。

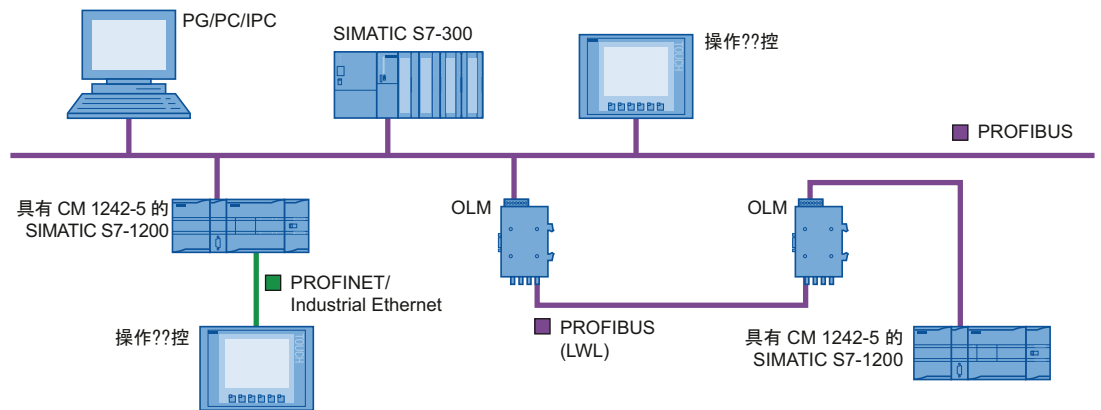


图 10-1 CM 1242-5 用做 PROFIBUS 从站的组态示例

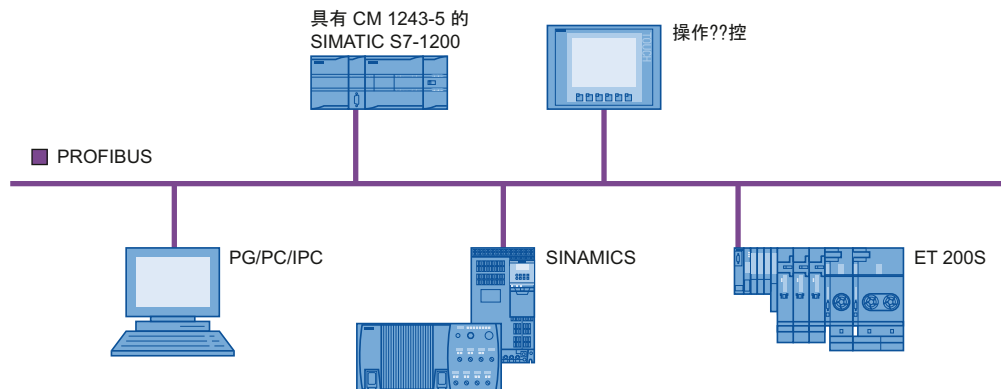



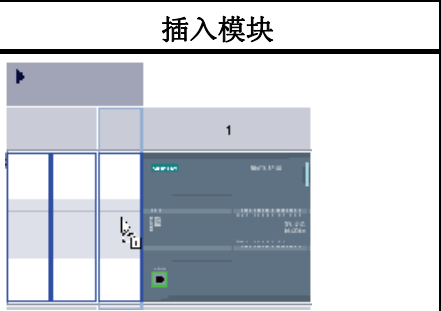
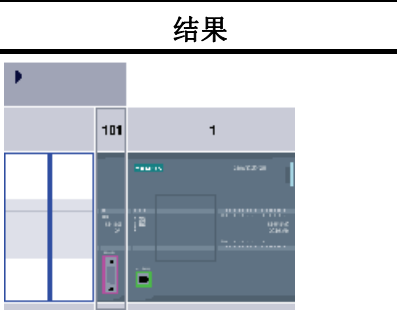
图 10-2 CM 1243-5 用做 PROFIBUS 主站的组态示例

### 10.3.2 配置 DP 主站和从站设备

#### 10.3.2.1 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 PROFINET 模块。这些模块连接在 CPU 左侧。要将模块插入到硬件配置中，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 10- 34 将 PROFIBUS CM 1243-5（DP 主站）模块添加到设备组态

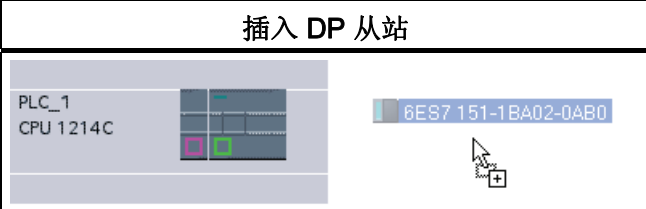
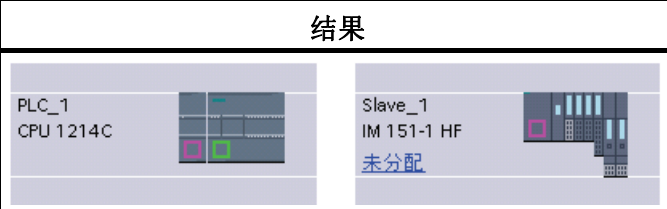
模块	选择模块	插入模块	结果
CM 1243-5 (DP 主站)			

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 10- 35 向设备组态添加 ET200 S DP 从站

插入 DP 从站	结果
	

#### 10.3.2.2 组态两台 PROFIBUS 设备之间的逻辑网络连接

组态 CM 1243-5（DP 主站）模块后，便可以组态网络连接。

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。要创建 PROFIBUS 连接, 请选择第一台设备上的紫色 (PROFIBUS) 框。拖出一条线连接到第二台设备上的 PROFIBUS 框。释放鼠标按钮, 即可创建 PROFIBUS 连接。

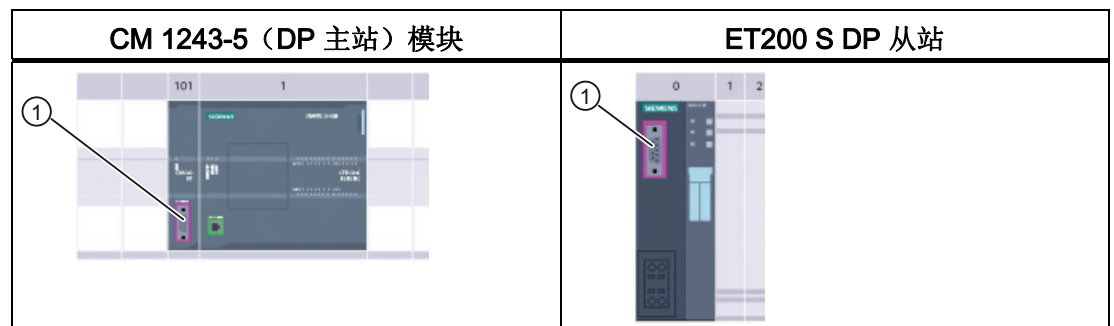
有关详细信息, 请参见“设备配置: 创建网络连接”(页 128)。

### 10.3.2.3 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 10- 36 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。



表格 10- 37 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"> <li>• 默认情况下“未连接”(Not connected) 提供本地连接。</li> <li>• 网络具有两个或多个设备时, 需要子网。</li> </ul>	
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)。

### 10.3.3 分布式 I/O 指令

请参见“分布式 I/O (PROFINET、PROFIBUS 或 AS-i)” (页 290), 以了解有关如何将分布式 I/O 指令用于这些通信网络的信息。



### 10.3.4 诊断指令

请参见“诊断（PROFINET 或 PROFIBUS）”：“诊断指令”（页 318），以了解有关如何将诊断指令用于这些通信网络的信息。

### 10.3.5 分布式的诊断事件

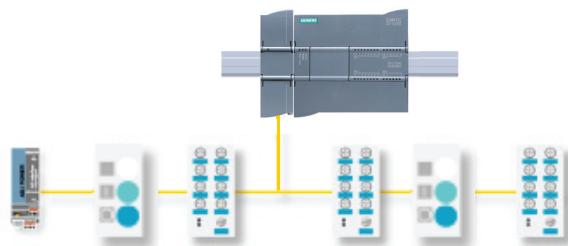
请参见“诊断（PROFINET 或 PROFIBUS）”：“分布式 I/O 的诊断事件”（页 318），以了解有关如何将该诊断信息用于这些通信网络的信息。

## 10.4 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 操作面板和数字量/模拟量 I/O 模块从站设备的 AS-i 主站。



### 10.4.1 组态 AS-i 主站和从站设备

AS-i 主站 CM 1243-2 作为通信模块集成到 S7-1200 自动化系统中。

有关 AS-i 主站 CM 1243-2 的详细信息，请参见“SIMATIC S7-1200 的 AS-i 主站 CM 1243-2 和 AS-i 数据解耦装置 DCM 1271”手册 (<http://support.automation.siemens.com/WW/view/en/50414115/133300>)。

### 10.4.1.1 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 10- 38 向设备组态添加 AS-i 主站 CM1243-2 模块

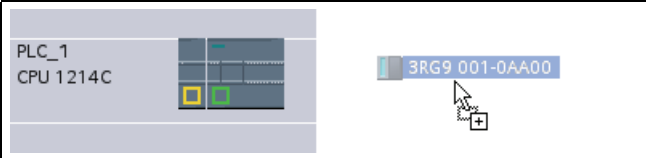
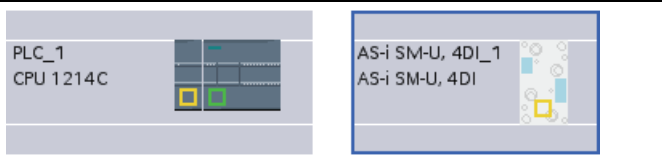
模块	选择模块	插入模块	结果
CM 1243-2 AS-i 主站			

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 10- 39 向设备组态添加 AS-i 从站

插入 AS-i 从站	结果
	

### 10.4.1.2 组态两个 AS-i 设备之间的逻辑网络连接

组态 AS-i 主站 CM1243-2 后，便可以组态网络连接。

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。要创建 AS-i 连接，在第一个设备上选择黄色的 (AS-i) 框。拖出一条线连接到第二个设备上的 AS-i 框。松开鼠标按钮，即可创建 AS-i 连接。

更多相关信息，请参见“设备组态：创建网络连接”(页 128)。

### 10.4.1.3 组态 AS-i 主站 CM1243-2 的属性

要组态 AS-i 接口的参数，请单击 AS-i 主站 CM1243-2 模块上的黄色 AS-i 框，巡视窗口的“属性”(Properties) 选项卡将显示该 AS-i 接口。

在 STEP 7 巡视窗口中，可以查看、组态以及更改常规信息、地址和操作参数：

表格 10- 40 AS-i 主站 CM1243-2 模块属性

属性	说明
常规	AS-i 主站 CM 1243-2 的名称
操作参数	AS-i 主站的响应参数
I/O 地址	从站 I/O 地址的地址区域
AS-i 接口 (X1)	分配的 AS-i 网络

#### 说明

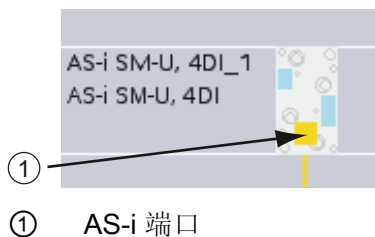
“AS-i 组态故障的诊断中断”(Diagnostic interrupt for faults in the AS-i configuration) 和“自动地址编程”(Automatic address programming) 始终处于激活状态，因此呈灰显。



### 10.4.1.4 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。

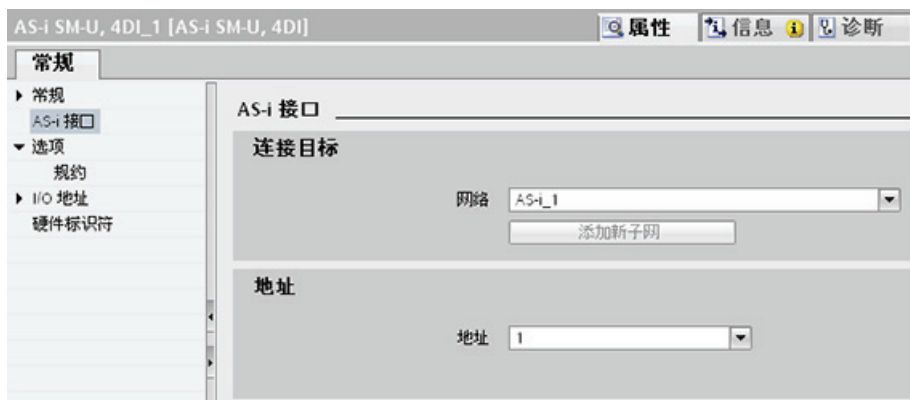


#### 分配 AS-i 从站地址

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。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。

在此处输入 AS-i 从站地址。



表格 10- 41 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

## 10.4.2 在用户程序和 AS-i 从站之间交换数据

### 10.4.2.1 STEP 7 基本组态

AS-i 主站在 CPU 的 I/O 区域中预留一个 62 字节的数据区。在此将按照字节访问数字量数据；对于每个从站，都有一个字节的输入数据和一个字节的输出数据。

并在 AS-i 主站 CM 1243-2 的巡视窗口中，指示 AS-i 数字量从站到所分配字节数据位的 AS-i 连接分配。



I 地址	O 地址	AS-i 地址	硬件 ID
		0	335
2	2	1A	336
33	33	1B	337
3	3	2A	338
34	34	2B	339
4	4	3A	340
35	35	3B	341
5	5	4A	342
36	36	4B	343
6	6	5A	344
37	37	5B	345
7	7	6A	346

可以通过相应位逻辑运算（如“AND”）的显示 I/O 地址或位分配，访问用户程序中 AS-i 从站的数据。

#### 说明

如果未使用 STEP 7 组态 AS-i 从站，则自动激活“系统分配”(System assignment)。如果不组态任何从站，则必须使用在线功能“ACTUAL > EXPECTED”通知 AS-i 主站 CM1243-2 有关实际总线组态的信息。

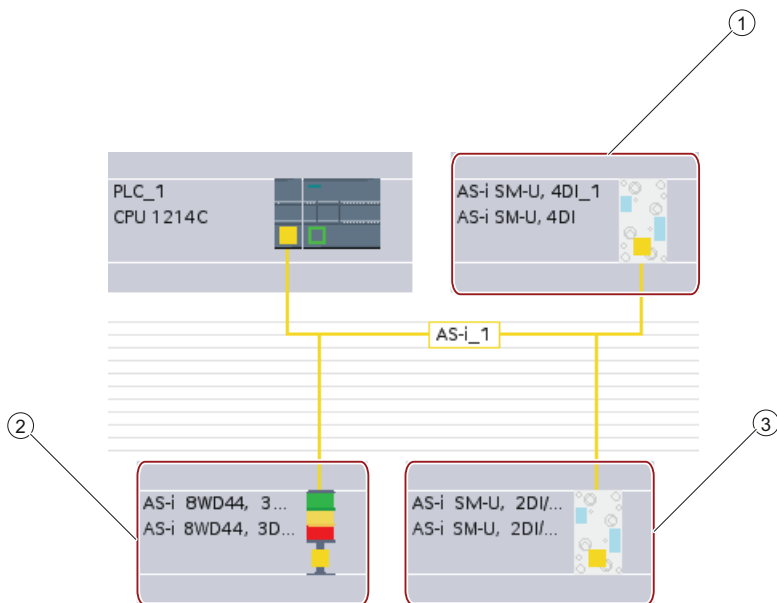
#### 更多信息

有关 AS-i 主站 CM 1243-2 的详细信息，请参见“SIMATIC S7-1200 的 AS-i 主站 CM 1243-2 和 AS-i 数据解耦装置 DCM 1271”手册 (<http://support.automation.siemens.com/WW/view/en/50414115/133300>)。

## 10.4.2.2 使用 STEP 7 组态从站

## 传输 AS-i 数字值

在循环操作中，CPU 通过 AS-i 主站 CM1243-2 访问 AS-i 从站的数字量输入和输出。可以通过 I/O 地址或数据记录传输访问数据。



- ① AS-i 从站地址 1
- ② AS-i 从站地址 2A
- ③ AS-i 从站地址 3

在此将按照字节访问数字量数据（即，每个 AS-i 数字量从站都对应一个字节）。在 STEP 7 中组态 AS-i 从站时，将在相应 AS-i 的巡视窗口中显示访问用户程序中数据的 IO 地址。

上述 AS-i 网络中的数字量输入模块（AS-i SM-U、4DI）已分配了从站地址 1。单击该数字量输入模块，设备“属性”(Properties) 的“AS 接口”(AS interface) 选项卡将显示从站地址，如下所示：

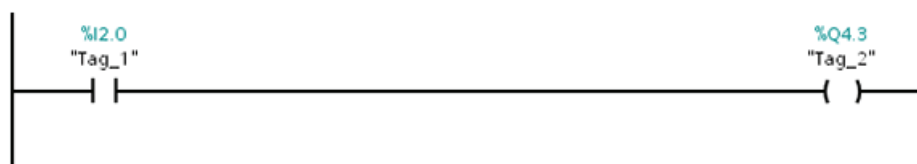


上述 AS-i 网络中的数字量输入模块（AS-i SM-U、4DI）已分配了 I/O 地址 2。单击该数字量输入模块，设备“属性”(Properties) 的“I/O 地址”(I/O addresses) 选项卡将显示 I/O 地址，如下所示：



可以通过对 I/O 地址进行相应位逻辑运算（如“AND”）或位分配，来访问用户程序中 AS-i 从站的数据。以下一段小程序举例说明了如何进行分配：

在本程序中将轮询输入 I2.0。在 AS-i 系统中，该输入属于从站 1（第 2 个输入字节，第 0 位）。随后设置的输出 Q4.3 对应于 AS-i 从站 3（第 4 个输出字节，第 3 位）



### 传输 AS-i 模拟值

如果在 STEP 7 中已将该 AS-i 从站组态为模拟量从站，那么就可以通过 CPU 的过程映像访问 AS-i 从站的模拟量数据。

如果没有在 STEP 7 中组态模拟量从站，那么只能通过非周期性函数（数据记录接口）访问 AS-i 从站的数据。在 CPU 的用户程序中，可以使用 RDREC（读取数据记录）和 WRREC（写入数据记录）分布式 I/O 指令读取和写入 AS-i 调用。

---

**说明**

在 S7 站的启动过程中，可以通过 AS-i 主站 CM1243-2 上的 CPU 传输通过 STEP 7 指定并下载到 S7 站中的 AS-i 从站的组态信息。并会覆盖由“系统分配”在线功能 (页 521) ("ACTUAL -> EXPECTED") 确定的所有现有组态信息。

---

**更多信息**

有关 AS-i 主站 CM 1243-2 的详细信息，请参见“SIMATIC S7-1200 的 AS-i 主站 CM 1243-2 和 AS-i 数据解耦装置 DCM 1271”手册 (<http://support.automation.siemens.com/WW/view/en/50414115/133300>)。

**10.4.3 分布式 I/O 指令**

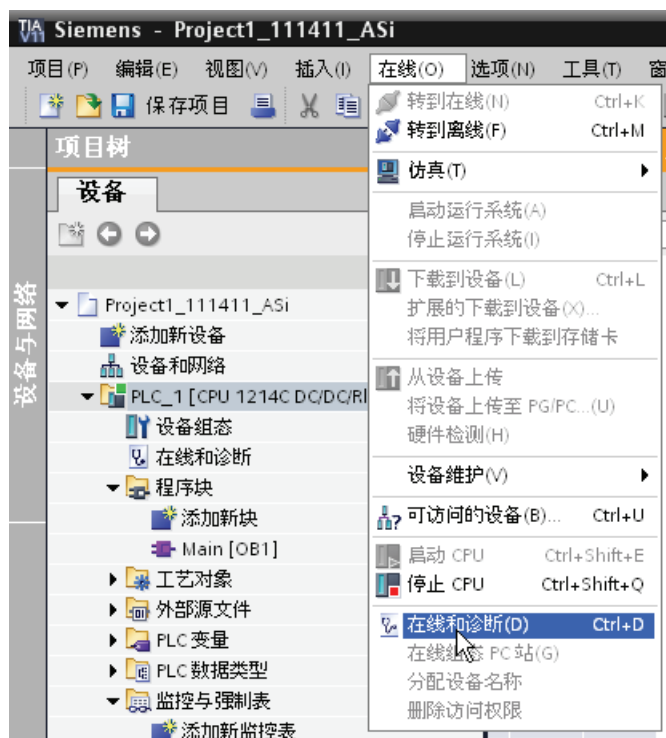
请参见“分布式 I/O（PROFINET、PROFIBUS 或 AS-i）”(页 290)，以了解有关如何将分布式 I/O 指令用于这些通信网络的信息。

**10.4.4 使用 AS-i 在线工具****在线更改 AS-i 操作模式**

必须在线查看和更改 AS-i 的操作模式。

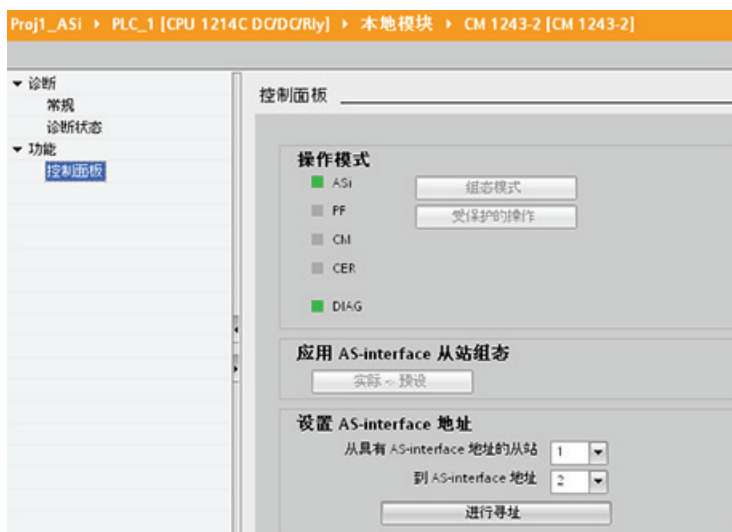
要转到在线模式，必须先在“设备组态”(Device Configuration) 中选择 AS-i 主站 CM1243-2 模块，然后单击工具栏中的“转到在线”(Go online) 按钮。然后，从“在线”(Online) 菜单中选择“在线和诊断”(Online and diagnostics) 命令。





有 2 种 AS-i 操作模式：

- 保护模式：
  - 不能更改 AS-i 从站设备和 CPU I/O 的地址。
  - 绿色“CM” LED 熄灭。
- 组态模式：
  - 可以在 AS-i 从站设备和 CPU I/O 地址进行相应更改。
  - 绿色“CM” LED 亮起。



在“设置 AS-i 地址”(Set AS-i address) 字段中，可以更改 AS-i 从站地址。对于尚未分配地址的新从站，其地址始终为 0。在为其分配地址之前，它会被主站检测为尚未分配地址的新从站，并且不包括在常规通信范围之内。

### 组态错误

黄色“CER” LED 亮起时，表示 AS-i 从站设备组态中出现错误。选择“ACTUAL > EXPECTED”按钮，用 AS-i 现场网络从站设备的组态覆盖 AS-i 主站 CM1243-2 模块从站设备的组态。

## 10.5 S7 通信

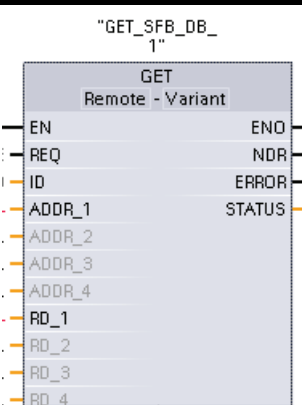
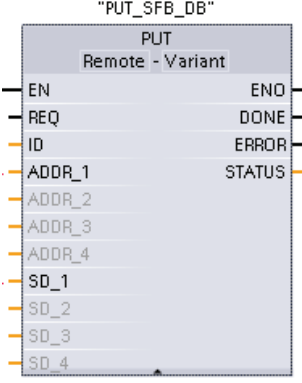
### 10.5.1 GET 和 PUT 指令

可以使用 GET 和 PUT 指令通过 PROFINET 和 PROFIBUS 连接与 S7 CPU 通信：

- 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。

表格 10-42 GET 和 PUT 指令

LAD/FBD	SCL	说明
	<pre>"GET_DB" (   req:=_bool_in_,   ID:=_word_in_,   ndr=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   addr_1:=_remote_inout_,   [...addr_4:=_remote_inout_,]   rd_1:=_variant_inout_   [...rd_4:=_variant_inout_]);</pre>	<p>使用 GET 指令从远程 S7 CPU 中读取数据。远程 CPU 可处于 RUN 或 STOP 模式下。</p> <p>STEP 7 会在插入指令时自动创建该 DB。</p>
	<pre>"PUT_DB" (   req:=_bool_in_,   ID:=_word_in_,   done=&gt;_bool_out_,   error=&gt;_bool_out_,   status=&gt;_word_out_,   addr_1:=_remote_inout_,   [...addr_4:=_remote_inout_,]   sd_1:=_variant_inout_   [...sd_4:=_variant_inout_]);</pre>	<p>使用 PUT 指令将数据写入远程 S7 CPU。远程 CPU 可处于 RUN 或 STOP 模式下。</p> <p>STEP 7 会在插入指令时自动创建该 DB。</p>

表格 10-43 参数的数据类型

参数和类型	数据类型	说明	
REQ	Input	Bool	通过由低到高的（上升沿）信号启动操作。
ID	Input	CONN_PRG (Word)	S7 连接 ID（十六进制）

参数和类型		数据类型	说明
NDR (GET)	Output	Bool	新数据就绪： <ul style="list-style-type: none"> <li>0: 请求尚未启动或仍在运行</li> <li>1: 已成功完成任务</li> </ul>
DONE (PUT)	Output	Bool	DONE: <ul style="list-style-type: none"> <li>0: 请求尚未启动或仍在运行</li> <li>1: 已成功完成任务</li> </ul>
ERROR STATUS	Output Output	Bool Word	<ul style="list-style-type: none"> <li>ERROR=0 STATUS 值： <ul style="list-style-type: none"> <li>0000H: 既没有警告也没有错误</li> <li>&lt;&gt; 0000H: 警告，STATUS 提供详细信息</li> </ul> </li> <li>ERROR = 1 出现错误。STATUS 提供有关错误性质的详细信息。</li> </ul>
ADDR_1	InOut	远程	指向远程 CPU 中存储待读取 (GET) 或待发送 (PUT) 数据的存储区。
ADDR_2	InOut	远程	
ADDR_3	InOut	远程	
ADDR_4	InOut	远程	
RD_1 (GET) SD_1 (PUT)	InOut	Variant	指向本地 CPU 中存储待读取 (GET) 或待发送 (PUT) 数据的存储区。
RD_2 (GET) SD_2 (PUT)	InOut	Variant	允许的数据类型： Bool（只允许单个位）、Byte、Char、Word、Int、DWord、DInt 或 Real。
RD_3 (GET) SD_3 (PUT)	InOut	Variant	注： 如果该指针访问 DB，则必须指定绝对地址，如： P# DB10.DBX5.0 Byte 10
RD_4 (GET) SD_4 (PUT)	InOut	Variant	在此情况下，10 代表 GET 或 PUT 的字节数。

必须确保 ADDR\_x (远程 CPU) 与 RD\_x 或 SD\_x (本地 CPU) 参数的长度 (字节数) 和数据类型相匹配。标识符“Byte”之后的数字是 ADDR\_x、RD\_x 或 SD\_x 参数引用的字节数。

---

#### 说明

通过 GET 指令可接收的字节总数或者通过 PUT 指令可发送的字节总数有一定的限制。具体限制取决于使用了四个可用地址和存储区中的多少:

- 如果仅使用 ADDR\_1 和 RD\_1/SD\_1, 则一个 GET 指令可获取 222 个字节, 一个 PUT 指令可发送 212 个字节。
- 如果使用 ADDR\_1、RD\_1/SD\_1、ADDR\_2 和 RD\_2/SD\_2, 则一个 GET 指令总共可获取 218 个字节, 一个 PUT 指令总共可发送 196 个字节。
- 如果使用 ADDR\_1、RD\_1/SD\_1、ADDR\_2、RD\_2/SD\_2、ADDR\_3 和 RD\_3/SD\_3, 则一个 GET 指令总共可获取 214 个字节, 一个 PUT 指令总共可获取 180 个字节。
- 如果使用 ADDR\_1、RD\_1/SD\_1、ADDR\_2、RD\_2/SD\_2、ADDR\_3、RD\_3/SD\_3、ADDR\_4、RD\_4/SD\_4, 则一个 GET 指令总共可获取 210 个字节, 一个 PUT 指令总共可发送 164 个字节。

各个地址和存储区参数的字节数之和必须小于等于定义的限值。如果超出这些限值, 则 GET 或 PUT 指令将返回错误。

---

在 REQ 参数的上升沿出现时, 读操作 (GET) 或写操作 (PUT) 将装载 ID、ADDR\_1 和 RD\_1 (GET) 或 SD\_1 (PUT) 参数。

- 对于 GET: 从下次扫描开始, 远程 CPU 会将请求的数据返回接收区 (RD\_x)。当读操作顺利完成时, NDR 参数设置为 1。只有在完成前一个操作后, 才能开始新的操作。
- 对于 PUT: 本地 CPU 开始将数据发送 (SD\_x) 到远程 CPU 中的存储位置 (ADDR\_x)。写操作顺利完成后, 远程 CPU 返回执行确认。然后, PUT 指令的 DONE 参数设置为 1。只有在完成前一个操作后, 才能开始新的写操作。

---

#### 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)。

---

ERROR 和 STATUS 参数提供有关读 (GET) 或写 (PUT) 操作的状态信息。

表格 10- 44 错误信息

ERROR	STATUS (十进制)	说明
0	11	<ul style="list-style-type: none"> <li>• 由于前一个作业还没有结束，所以不能执行新作业。</li> <li>• 正在以较低优先级处理此作业。</li> </ul>
0	25	通信已启动。正在处理作业。
1	1	通讯故障，如： <ul style="list-style-type: none"> <li>• 未装载连接描述（本地或远程）</li> <li>• 连接被中断（例如： 电缆断线、CPU 关闭或 CM/CB/CP 处于 STOP 模式）</li> <li>• 没有建立到通信伙伴的连接</li> </ul>
1	2	来自伙伴设备的否定应答。无法执行任务。
1	4	发送区指针（GET 的 RD_x，或 PUT 的 SD_x）出错，包括数据长度或数据类型。
1	8	在伙伴 CPU 上发生访问错误
1	10	无法访问本地用户存储器（例如，尝试访问已经删除的数据块）
1	12	调用 SFB 时： <ul style="list-style-type: none"> <li>• 指定了不属于 GET 或 PUT 的背景数据块</li> <li>• 未指定背景数据块，而是指定了一个共享数据块</li> <li>• 未发现背景数据块（装载新的背景数据块）</li> </ul>
1	20	<ul style="list-style-type: none"> <li>• 超出并行作业/实例的最大数量</li> <li>• 当 CPU 处于 RUN 模式时，实例过载</li> </ul> 首次执行 GET 或 PUT 指令时可能出现此状态
1	27	CPU 中没有相应的 GET 或 PUT 指令。

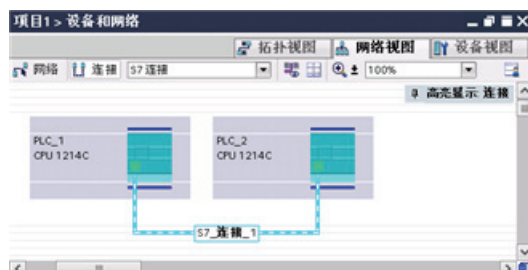
## 10.5.2 创建 S7 连接

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 128)。



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

### 10.5.3 组态两台设备间的本地/伙伴连接路径

#### 组态常规参数

在通信指令的“属性”(Properties) 组态对话框中指定通信参数。只要选中了该指令的任何一部分，此对话框就会出现在页面底部附近。

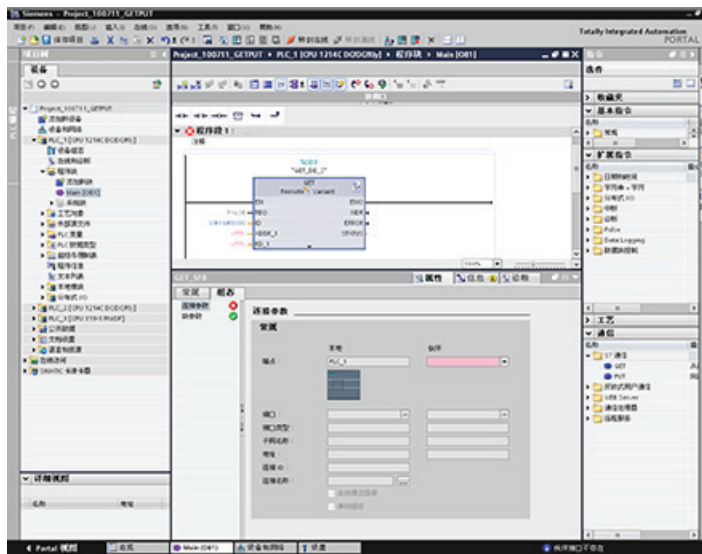
更多相关信息，请参见“设备配置: 组态本地/伙伴连接路径(页 129)”。

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 或端口。在“本地 TSAP”(Local TSAP) 字段中输入 CPU 中连接的 TSAP 或端口。在“伙伴 TSAP”(Partner TSAP) 字段下输入为伙伴 CPU 中的连接分配的 TSAP 或端口。

### 10.5.4 GET/PUT 连接参数分配

GET/PUT 指令连接参数分配是一项用于 CPU 间 S7 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：

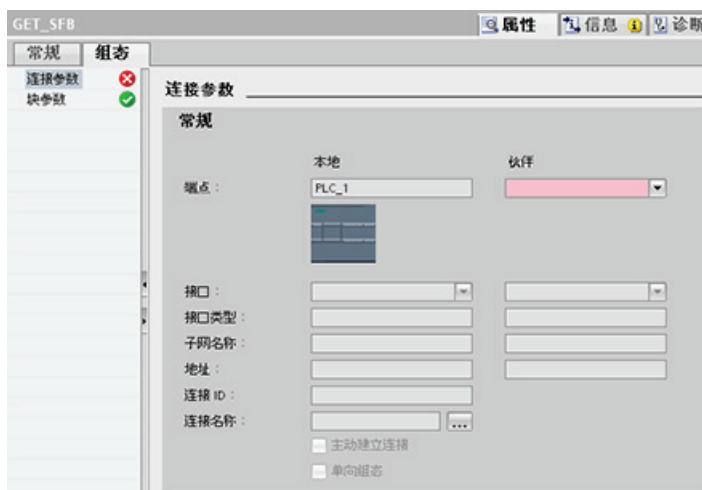


每次选择指令的任何一部分，巡视窗口都会显示连接的属性。可以在通信指令“属性”(Properties)的“组态”(Configuration)选项卡中指定通信参数。

### 10.5.4.1 连接参数

在“连接参数”(Connection parameters)页面中，可以组态必要的 S7 连接，以及组态由 GET/PUT 块参数“ID”引用的参数“连接 ID”(Connection ID)。页面内容包括有关本地端点的信息，用户可在页面中定义本地接口。用户还可定义伙伴端点。

通过“块参数”(Block parameters)页面可组态其它块参数。





表格 10-45 连接参数：常规定义

参数	定义
连接参数： 常规	<p>“本地端点”： 分配给本地 CPU 的名称</p> <p>“伙伴端点”： 分配给伙伴（远程）CPU 的名称</p> <p>注： 在“伙伴端点”(Partner End point) 下拉列表中，系统将显示当前项目中所有可能的 S7 连接伙伴以及选项“未指定”(unspecified)。未指定伙伴是指当前不在 STEP 7 项目中的通信伙伴（例如，第三方设备通信伙伴）。</p>
接口	<p>分配给接口的名称</p> <p>注： 用户可通过更改本地和伙伴接口来更改连接</p>
接口类型	接口类型
子网名称	分配给子网的名称
地址	<p>分配的 IP 地址</p> <p>注： 用户可为“未指定”通信伙伴指定一个第三方设备远程地址。</p>
连接 ID	ID 号： 由 GET/PUT 连接参数分配自动生成
连接名称	本地和伙伴 CPU 的数据存储位置： 由 GET/PUT 连接参数分配自动生成
主动连接建立	用于选择本地 CPU 作为主动连接的复选框
单向	<p>指定单向或双向连接的复选框；只读</p> <p>注： 在 PROFINET GET/PUT 连接中，本地与伙伴设备都可以作为服务器或客户端。这样就可以进行双向连接，并取消选中“单向”(One-way) 复选框。</p> <p>在某些情况下，PROFIBUS GET/PUT 连接中的伙伴设备只能作为服务器（例如 S7-300），并选中“单向”(One-way) 复选框。</p>

## 连接 ID 参数

共有三种更改系统定义连接 ID 的方法：

1. 用户可在 GET/PUT 块中直接更改当前 ID。如果新 ID 属于已存在的连接，则连接将更改。

2. 用户可在 GET/PUT 块中直接更改当前 ID，但不能有新 ID。系统已创建新 S7 连接。
3. 可通过“连接概况”(Connection overview) 对话框更改当前的 ID：用户的输入与相应 GET/PUT 块中的 ID 参数同步。

### 说明

GET/PUT 块的参数“ID”不是连接名称，而是一个数字表达式，其写法类似于以下示例：**W#16#1**

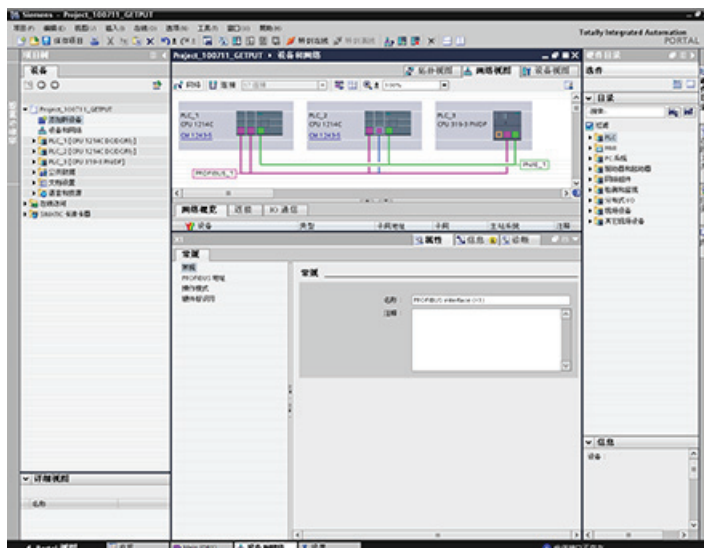
### 连接名称参数

可以通过特殊用户控件（“连接概况”(Connection overview) 对话框）编辑连接名称。该对话框提供所有可用 S7 连接，可以选择这些连接作为当前 GET/PUT 通信的备选方式。用户可在此表中创建全新的连接。单击“连接名称”(Connection name) 字段右侧的按钮，可启动“连接概况”(Connection overview) 对话框。



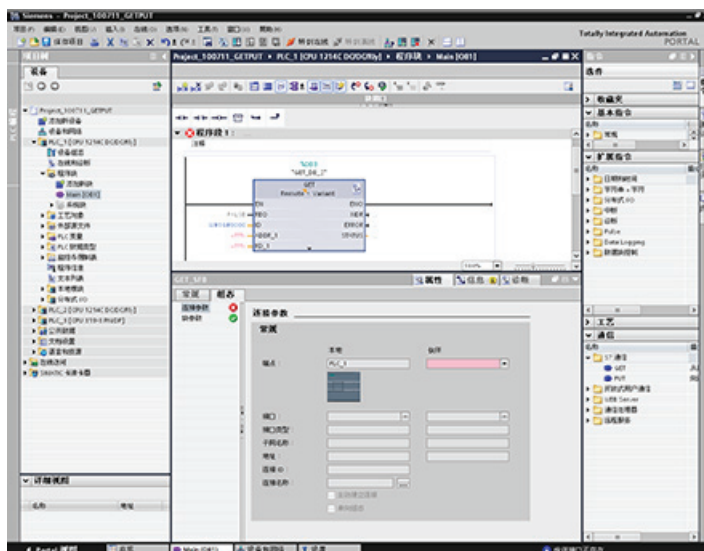
### 10.5.4.2 组态 CPU 间的 S7 连接

假设 PLC\_1、PLC\_2 和 PLC\_3 的组态如下图所示，为“PLC\_1”插入 GET 或 PUT 块。



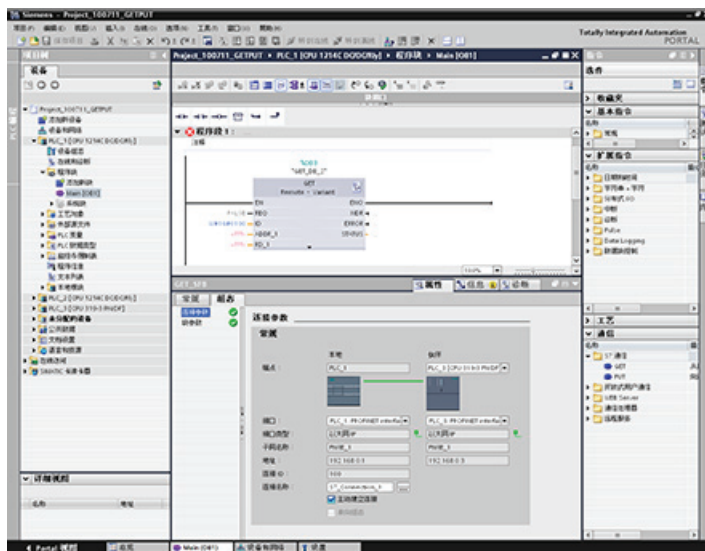
对于 GET 或 PUT 指令，将在巡视窗口中自动显示“属性”(Properties) 选项卡，且包含以下菜单选项：

- “组态”(Configuration)
- “连接参数”(Connection parameters)



组态 PROFINET S7 连接

对于“伙伴端点”，请选择“PLC\_3”。



系统将进行以下更改以对此进行响应：

表格 10- 46 连接参数： 常规值

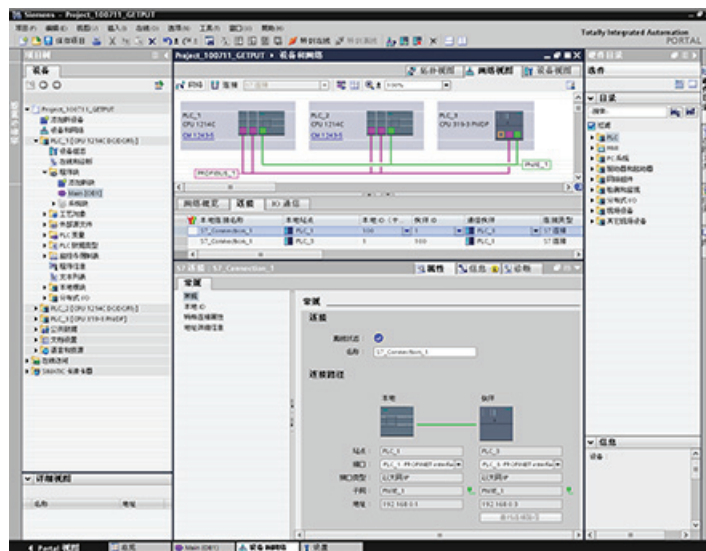
参数	定义
连接参数： 常规	<p>端点</p> <p>“本地端点”中为只读的“PLC_1”。</p> <p>“伙伴端点”字段中为“PLC_3[CPU319-3PN/DP]”：</p> <ul style="list-style-type: none"> <li>• 颜色从红色变为白色</li> <li>• 显示“伙伴”设备图像。</li> <li>• PLC_1 和 PLC_3 设备图像之间有一条连接线（绿色以太网线）。</li> </ul>
接口	<p>“本地接口”中为“CPU1214C DC/DC/DC、PROFINET interface (R0/S1)”。</p> <p>“伙伴接口”中为：“CPU319-3PN/DP、PROFINET interface (R0/S2)”。</p>
接口类型	<p>“本地接口类型”中为“Ethernet/IP”；控件为只读。</p> <p>“伙伴接口类型”中为“Ethernet/IP”；控件为只读。</p> <p>本地和伙伴“接口类型”（绿色以太网图标）旁显示接口类型的图像。</p>
子网名称	<p>“本地子网名称”中为“PN/IE_1”；控件为只读。</p> <p>“伙伴子网名称”中为“PN/IE_1”；控件为只读。</p>

参数	定义
地址	“本地地址”中为本地 IP 地址；控件为只读。 “伙伴地址”中为伙伴 IP 地址；控件为只读。
连接 ID	“连接 ID”中为“100”。 在程序编辑器中，Main [OB1] 中的 GET/PUT 块“连接 ID”值也为“100”。
连接名称	“连接名称”中为默认的连接名称（例如，“S7_Connection_1”）；控件已启用。
主动连接建立	选中并启用，以选择本地 CPU 作为主动连接。
单向	只读且取消选中。 注：“PLC_1” (S7-1200 CPU 1214CDC/DC/Rly) 和“PLC_3” (S7-300 CPU 319-3PN/DP) 在 PROFINET GET/PUT 连接中都可以作为服务器和客户端，实现双向连接。

属性视图树中的 GET/PUT 图标也将从红色变为绿色。

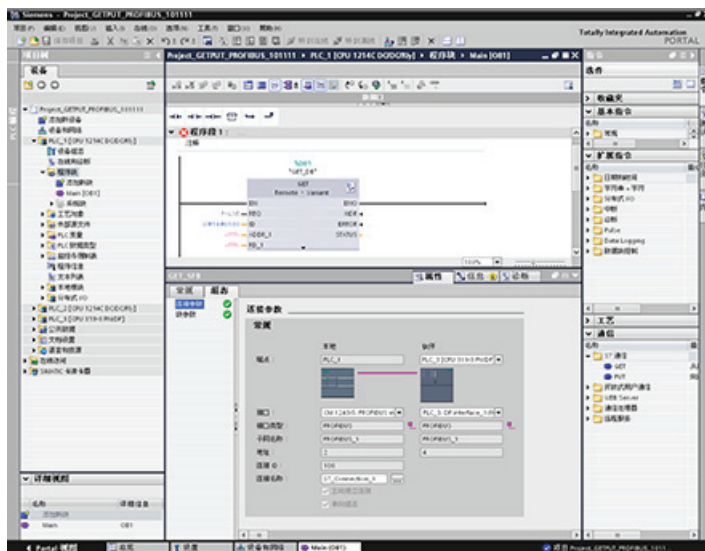
## 完成 PROFINET S7 连接

在“网络视图”(Network view) 中，将在“PLC\_1”和“PLC\_3”之间的“连接”(Connections) 表中显示 S7 双向连接。



### 组态 PROFIBUS S7 连接

对于“伙伴端点”，请选择“PLC\_3”。



系统将进行以下更改以对此进行响应：

表格 10- 47 连接参数： 常规值

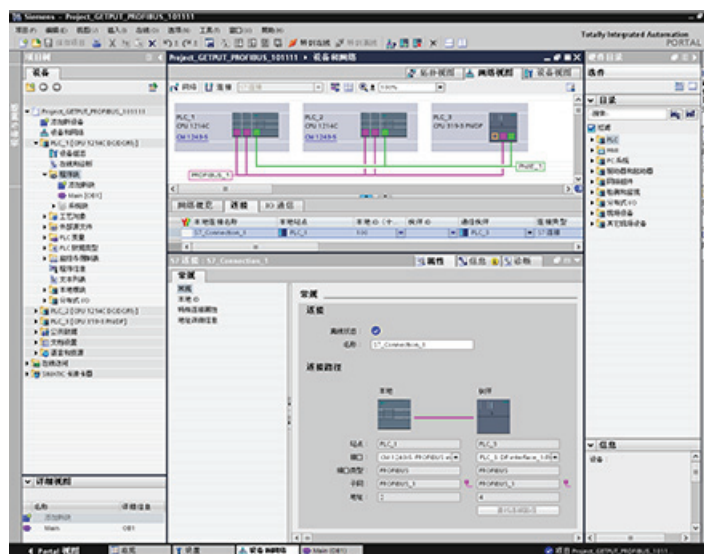
参数	定义
连接参数： 常规	<p>端点</p> <p>“本地端点”中为只读的“PLC_1”。</p> <p>“伙伴端点”字段中为“PLC_3[CPU319-3PN/DP]”：</p> <ul style="list-style-type: none"> <li>• 颜色从红色变为白色</li> <li>• 显示“伙伴”设备图像。</li> <li>• PLC_1 和 PLC_3 设备图像之间有一条连接线（紫色的 PROFIBUS 线）。</li> </ul>
接口	<p>“本地接口”中为“CPU1214C DC/DC/DC、PROFIBUS interface (R0/S1)”。</p> <p>“伙伴接口”中为：“CPU319-3PN/DP、PROFIBUS interface (R0/S2)”。</p>
接口类型	<p>“本地接口类型”中为“PROFIBUS”；控件为只读。</p> <p>“伙伴接口类型”中为“PROFIBUS”；控件为只读。</p> <p>本地和伙伴“接口类型”（紫色的 PROFIBUS 图标）旁显示接口类型的图像。</p>

参数	定义
子网名称	“本地子网名称”中为“PROFIBUS_1”；控件为只读。 “伙伴子网名称”中为“PROFIBUS_1”；控件为只读。
地址	“本地地址”中为本地 IP 地址；控件为只读。 “伙伴地址”中为伙伴 IP 地址；控件为只读。
连接 ID	“连接 ID”中为“100”。 在程序编辑器中，Main [OB1] 中的 GET/PUT 块“连接 ID”值也为“100”。
连接名称	“连接名称”中为默认的连接名称（例如，“S7_Connection_1”）；控件已启用。
主动连接建立	只读，选中并启用，以选择本地 CPU 作为主动连接。
单向	只读且选中。 注：“PLC_3” (S7-300 CPU319-3PN/DP) 在 PROFIBUS GET/PUT 连接中只能作为服务器（无法同时作为客户端），只能进行单向连接。

属性视图树中的 GET/PUT 图标也将变为绿色。

## 完成 PROFIBUS S7 连接

在“网络视图”(Network view) 中，将在“PLC\_1”和“PLC\_3”之间的“连接”(Connections) 表中显示 S7 单向连接。







## Web 服务器

S7-1200 的 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。

### 标准 Web 页面

S7-1200 包含可通过 PC 的 Web 浏览器进行访问 (页 543) 的标准 Web 页面:

- 简介 (页 547) - 标准 Web 页面的进入点
- Start Page (页 548) - 有关 CPU 的常规信息
- Identification (页 549) - 有关 CPU 的详细信息, 包括序列号、订单号和版本号
- Module Information (页 550) - 有关本地机架中的模块的信息
- Communication (页 553) - 有关网络地址、通信接口的物理属性和通信统计的信息
- Diagnostic Buffer (页 549) - 诊断缓冲区
- Variable Status (页 554) - CPU 变量和 I/O, 可通过地址或 PLC 变量名称进行访问
- Data Logs (页 556) - 存储在 CPU 内部或存储卡中的数据日志文件
- 更新固件 (页 559) - 更新 CPU 中的固件

这些页面内置于 S7-1200。有关标准 Web 页面以及如何对其进行访问的详细信息, 请参见 标准 Web 页面 (页 543) 部分。

### 用户定义的 Web 页面

S7-1200 还支持您创建可访问 CPU 数据的用户定义的 Web 页面。可以使用所选的 HTML 创作软件来开发这类页面, 并且可将预定义的“AWP” (Automation Web Programming, 自动化 Web 编程) 命令包含在 HTML 代码中以访问 CPU 数据。有关开发用户定义的 Web 页面以及在 STEP 7 中进行相关组态和编程的具体信息, 请参见 用户定义的 Web 页面 (页 561) 一章。

## Web 浏览器要求

以下 Web 浏览器支持 Web 服务器：

- Internet Explorer 8.0 或更新版本
- Mozilla Firefox 3.0 或更新版本
- Opera 11.0 或更新版本

有关可干扰标准或用户定义的 Web 页面显示的浏览器相关的限制，请参见 限制 (页 601) 部分。

## 11.1 启用 Web 服务器

在 STEP 7 中，通过“设备组态”(Device Configuration) 为要连接的 CPU 启用 Web 服务器。

要启用 Web 服务器，请按以下步骤操作：

1. 在设备组态视图中选择 CPU。
2. 在巡视窗口中，从 CPU 属性中选择“Web 服务器”(Web server)。
3. 选中“在该模块上启用 Web 服务器”(Enable Web server on this module) 复选框。
4. 如果需要对 Web 服务器进行安全访问，请选中“仅允许使用 HTTPS 访问”(Permit access only with HTTPS) 复选框。



### 警告

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“admin”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：

- 只使用 HTTPS 协议启用对 Web 服务器的访问权限。
- 使用可靠的密码对 CPU 进行密码保护 (页 166)。可靠的密码在长度上至少为八个字符，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。
- 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。

下载设备组态后，可使用标准 Web 页面访问 CPU。如果针对“自动更新”(Automatic update) 选择了“启用”(Enable)，则标准 Web 页面每十秒刷新一次。

如果创建了用户定义的 Web 页面，则可通过标准 Web 页面菜单访问这些页面。

---

#### 说明

如果正在“在 RUN 模式下进行下载”(页 746)，那么在下载完成之前，标准和用户定义的 Web 页面不会更新数据值，也不允许写入任何数据值。下载期间，进行的任何写入数据值尝试都将被丢弃。

---

## 11.2 标准 Web 页面

### 11.2.1 通过 PC 访问标准 Web 页面

要通过 PC 访问 S7-1200 的标准 Web 页面，请按以下步骤操作：

1. 请确保 S7-1200 和 PC 位于同一个以太网中，或二者之间直接使用标准以太网电缆进行连接。
2. 打开 Web 浏览器，输入 URL“http://ww.xx.yy.zz”，其中“ww.xx.yy.zz”与 S7-1200 CPU 的 IP 地址相对应。

Web 浏览器将打开“简介”(Introduction) 页面。

---

#### 说明

如果 Internet 访问不允许直接连接到 IP 地址，请咨询 IT 管理员。Web 环境或操作系统可能造成其它限制(页 601)。

---

也可将 Web 浏览器设置为指向特定的标准 Web 页面。为此，应输入“http://ww.xx.yy.zz/<页面>.html”形式的 URL，其中 <页面> 为以下标准 Web 页面之一：

- start (页 548) - 有关 CPU 的常规信息
- identification (页 549) - 有关 CPU 的详细信息，包括序列号、订单号和版本号
- module (页 550) - 有关本地机架中的模块的信息
- communication (页 553) - 有关网络地址、通信接口的物理属性和通信统计的信息
- diagnostic (页 549) - 诊断缓冲区
- variable (页 554) - CPU 变量和 I/O，可通过地址或 PLC 变量名称进行访问
- datalog (页 556) - 存储在 CPU 内部或存储卡中的数据日志文件

- [updatefirmware](#) (页 559) - 能够通过文件更新 CPU 中的固件的页面
- [index](#) (页 547) - 进入标准 Web 页面的简介页面

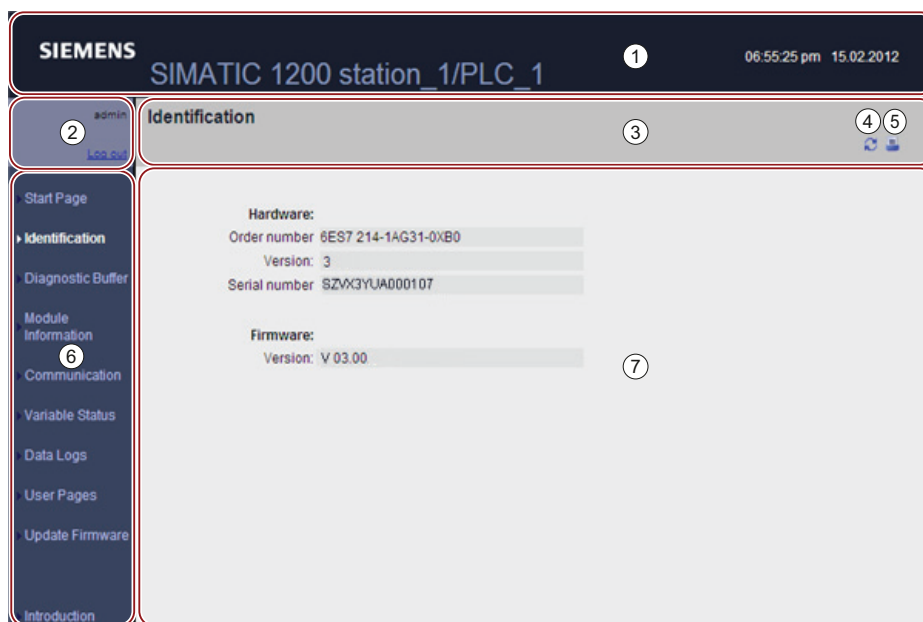
例如，如果输入“<http://www.xx.yy.zz/communication.html>”，浏览器将会显示通信页面。

### 安全访问

可以使用 [https://](#)（而不是 [http://](#)）对标准 Web 页面进行安全访问。使用 [https://](#) 连接到 S7-1200 时，网站将通过数字证书对会话进行加密。数据将安全传输，而不会被任何人看到。通常，您会收到安全警告，可以按“是”(Yes) 继续浏览标准 Web 页面。要避免每次安全访问时都出现安全警告，可以将 Siemens 软件证书导入 Web 浏览器 (页 603)。

## 11.2.2 标准 Web 页面的布局

每个标准 Web 页面的布局均相同，都具有如下所示的导航链接和页面控件：



- ① Web 服务器标题
- ② 登录或注销
- ③ 包含您正在查看的页面的名称的标准 Web 页面标题。本示例为 CPU 的 **Identification** 页面。有些标准 Web 页面（如模块信息页面）还会在此处显示导航路径，前提是您可以访问该类型的多个画面。
- ④ 刷新图标：对于具有自动更新功能的页面，可用来启用或禁用自动更新功能；对于不具有自动更新功能的页面，可以使页面用当前数据进行更新
- ⑤ 打印图标：准备并显示所显示页面提供的信息的可打印版本
- ⑥ 用来切换到其它页面的导航区
- ⑦ 正在查看的特定标准 Web 页面的内容区域。本示例为 CPU 的 **Identification** 页面。

### 说明

#### 打印标准 Web 页面

打印标准 Web 页面内容时请注意，打印的内容有时可能与显示的页面不同。例如，“诊断缓冲区”(Diagnostic buffer) 页面的打印件可能包含并未显示在该页面上的新诊断条目。如果未启用自动刷新，显示出的内容是最初显示页面时的诊断事件，而打印件包含的内容则是执行打印功能时的诊断事件。

## 登录

查看标准 Web 页面中的数据不需要进行登录。要执行特定操作（如更改控制器的操作模式或向存储器写入值以及更新 CPU 固件），必须以“admin”用户的身份登录。

A screenshot of a login form. It has two input fields: 'Name' and 'Password'. Below the 'Password' field is a blue 'Login' button.

登录框位于每个页面的左上角附近。

要以“admin”用户的身份登录，请按以下步骤操作：

1. 在“名称”(Name) 字段中输入“admin”。
2. 如果在 Password 域中组态了 CPU 密码，应输入 CPU 密码；否则，按 Enter 键。

现在，您便以“admin”用户的身份登录。



### 警告

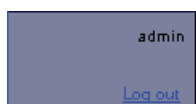
未经授权访问 CPU 或将 PLC 变量更改为无效值可能会中断过程操作并可能导致死亡、严重人身伤害和/或财产损失。

由于启用 Web 服务器可让“admin”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：

- 只使用 HTTPS 协议启用对 Web 服务器的访问权限。
- 使用可靠的密码对 CPU 进行密码保护 (页 166)。可靠的密码在长度上至少为八个字符，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。
- 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。

如果在登录时遇到错误，返回“简介”(Introduction) 页面 (页 547)，然后下载 Siemens 安全证书 (页 603)。随后便可成功登录。

## 注销



要注销“admin”用户，只需单击任意页面的“注销”(Log out) 链接。

也可以在未登录的情况下继续访问和查看标准 Web 页面，但不能执行仅限于“admin”用户的操作。每个标准 Web 页面说明都定义了需要以“admin”身份登录才能执行的操作（如果有）。

### 11.2.3 简介

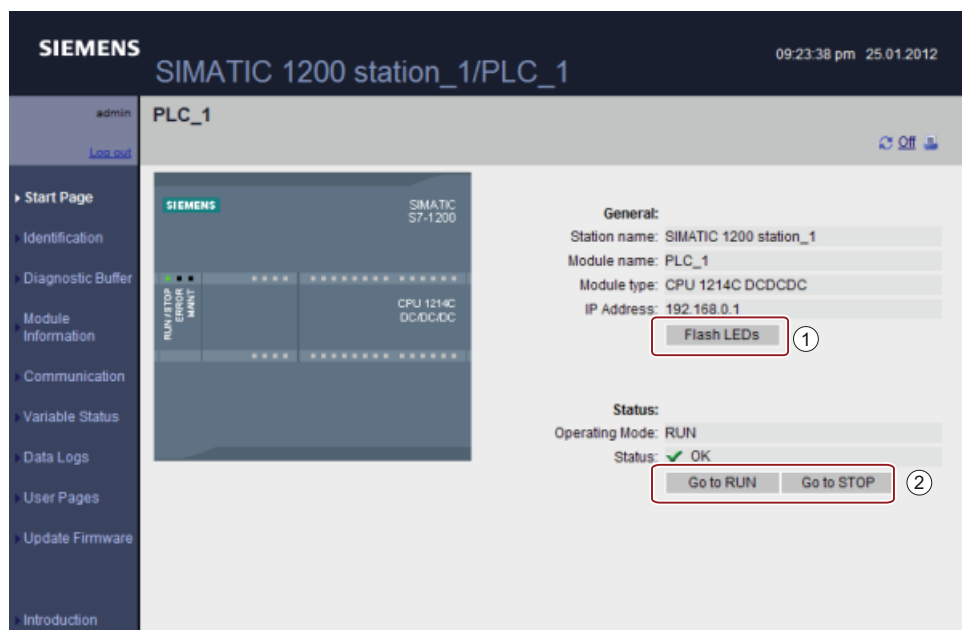
“简介”(Introduction) 页面是进入 S7-1200 标准 Web 页面的欢迎画面。



在该页面中单击“Enter”可访问 S7-1200 标准 Web 页面。屏幕上方是有用的 Siemens Web 网站的链接以及下载 Siemens 安全证书 (页 603) 的链接。

### 11.2.4 Start

Start 页面显示所连接 CPU 的图示，并列出有关 CPU 的常规信息。如果以“admin”用户的身份登录，还可以更改 CPU 的操作模式并使 LED 闪烁。



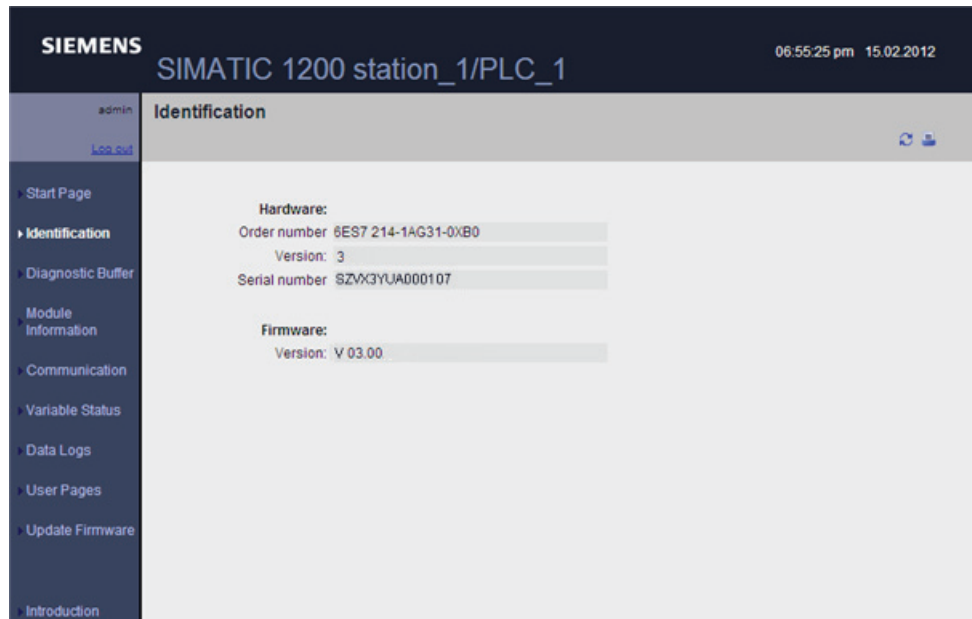
① 和 ② 仅当您以“admin”用户的身份登录时，才会在“启动”(Start)页面上显示用于使 LED 闪烁以及更改操作模式的按钮。



## 11.2.5 Identification

Identification 页面显示 CPU 的标识特性：

- 序列号
- 订货号
- 版本信息



“标识”(Identification) 页面与您是否以“admin”身份登录无关。

## 11.2.6 Diagnostic Buffer

Diagnostic Buffer 页面会显示诊断事件。可以从选择器中选择要显示的诊断缓冲区条目范围（1 到 25 或 26 到 50）。页面的顶部会显示这些条目以及事件发生时对应的 CPU 时间和日期。时间为来自 CPU “时间”(Time of day) (页 87) 的系统时间。

可以从页面顶部选择任何单独的条目，以在页面底部显示有关该条目的详细信息。

Number	Time	Date	Event
1	01:55:06.490 pm	26.01.2012	CPU info: Follow-on operating mode change
2	01:55:06.488 pm	26.01.2012	CPU info: Communication initiated request: WARM RESTART
3	01:55:06.488 pm	26.01.2012	CPU info: New startup information
4	01:54:57.888 pm	26.01.2012	CPU info: New startup information
5	01:54:57.788 pm	26.01.2012	CPU info: Communication initiated request: STOP
6	08:53:29.180 pm	25.01.2012	CPU info: Follow-on operating mode change
7	08:53:29.178 pm	25.01.2012	CPU info: Communication initiated request: WARM RESTART
8	08:53:29.178 pm	25.01.2012	CPU info: New startup information
9	08:53:26.978 pm	25.01.2012	CPU info: New startup information
10	08:53:23.478 pm	25.01.2012	CPU info: New startup information
11	08:53:22.262 pm	25.01.2012	CPU info: New startup information
12	08:53:22.162 pm	25.01.2012	CPU info: Communication initiated request: STOP
13	08:52:43.600 pm	25.01.2012	CPU info: Follow-on operating mode change

Details: 1 Event ID: 16# 02:400C

CPU info: Follow-on operating mode change  
Power-on mode set: WARM RESTART to RUN (if CPU was in RUN before power off)

Pending startup inhibit(s):  
- No startup inhibit set

CPU changes from STARTUP to RUN mode

HW\_ID= 00052 - Operating mode control

“诊断缓冲区”(Diagnostic buffer) 页面与您是否以“admin”身份登录无关。

## 11.2.7 模块信息

Module Information 页面提供有关本地机架中所有模块的信息。屏幕顶部显示模块的摘要，屏幕底部显示所选模块的状态和标识。

## 状态显示

SIEMENS SIMATIC 1200 station\_1/PLC\_1 03:49:55 pm 26.01.2012

admin Module information Slot  Filter

[Log out](#) SIMATIC 1200 station\_1 - SIMATIC 1200 station\_1 [Off](#) [Print](#)

Slot	Status	Name	Order number	I address	Q address	Comment
1	✓	<a href="#">PLC_1</a>	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	---	---	
2	✓	<a href="#">DQ16 x 24VDC_1</a>	<a href="#">Details</a> 6ES7 222-1BH30-0XB0	---	8	
3	✓	<a href="#">DI16/DQ16 x 24VDC_1</a>	<a href="#">Details</a> 6ES7 223-1BL30-0XB0	12	12	

Start Page Identification Diagnostic Buffer

**Module information**

Communication Variable Status Data Logs User Pages Update Firmware Introduction

**Status Identification**

The module is parameterized, and it exchanges data with the CPU. The module did not detect any parameter assignment errors, channel errors, or hardware related errors.

## 标识显示

SIEMENS SIMATIC 1200 station\_1/PLC\_1 03:51:25 pm 26.01.2012

admin Module information Slot  Filter

[Log out](#) SIMATIC 1200 station\_1 - SIMATIC 1200 station\_1 [Off](#) [Print](#)

Slot	Status	Name	Order number	I address	Q address	Comment
1	✓	<a href="#">PLC_1</a>	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	---	---	
2	✓	<a href="#">DQ16 x 24VDC_1</a>	<a href="#">Details</a> 6ES7 222-1BH30-0XB0	---	8	
3	✓	<a href="#">DI16/DQ16 x 24VDC_1</a>	<a href="#">Details</a> 6ES7 223-1BL30-0XB0	12	12	

Start Page Identification Diagnostic Buffer

**Module information**

Communication Variable Status Data Logs User Pages Update Firmware Introduction

**Status Identification**

Hardware Version: 1  
Firmware Version: R.3.0.0  
Serial Number: SZVX3YUA000107

## 深入展开

可以选择顶部的链接来深入展开特定模块的模块信息。具有子模块的模块包含每个子模块的链接。显示的信息类型会根据所选模块的不同而异。例如，模块信息对话框最初会

显示 SIMATIC 1200 站的名称、状态指示灯和注释。如果深入展开 CPU，模块信息将显示 CPU 型号提供的数字量和模拟量输入输出的名称（例如“DI14/DO10”和“AI2”）、I/O 地址信息、状态指示灯、插槽号和注释。

Slot	Status	Name	Order number	I address	Q address	Comment
1.16	✓	HSC_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1000	---	
1.17	✓	HSC_2	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1004	---	
1.18	✓	HSC_3	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1008	---	
1.19	✓	HSC_4	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1012	---	
1.20	✓	HSC_5	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1016	---	
1.21	✓	HSC_6	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	1020	---	
1.2	✓	AI2_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	64	---	
1.1	✓	DI14/DO10_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	0	0	
1.32	✓	Pulse_1	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	---	1000	
1.33	✓	Pulse_2	<a href="#">Details</a> 6ES7 214-1AG31-0XB0	---	1002	

在深入展开的同时，模块信息页面会显示访问的路径。可以单击该路径中的任意链接返回到更高级别。

**Module Information**  
SIMATIC 1200 station 1 - SIMATIC 1200 station 1 - PLC\_1

## 对域排序

如果列表显示多个模块，则可单击域的列标题，以按该域对列表进行升序或降序排列。

Name	Details
AI2_1	<a href="#">Details</a>
DI14/DO10_1	<a href="#">Details</a>
HSC_1	<a href="#">Details</a>
HSC_2	<a href="#">Details</a>
HSC_3	<a href="#">Details</a>
HSC_4	<a href="#">Details</a>
HSC_5	<a href="#">Details</a>
HSC_6	<a href="#">Details</a>
Pulse_1	<a href="#">Details</a>
Pulse_2	<a href="#">Details</a>

## 过滤模块信息

可以过滤模块信息列表中的任何域。从下拉列表中选择要过滤数据的域名称。在相关文本框中输入文本并单击 **Filter** 链接。列表会进行更新，以显示与过滤标准相对应的模块。

## 状态信息

Module Information 页面底部的 **Status** 选项卡显示顶部所选模块的当前状态的说明。

## Identification

Identification 选项卡显示所选模块的序列号和修订版本号。

“模块信息”(Module Information) 页面与您是否以“admin”身份登录无关。

## 11.2.8 Communication

Communication 页面显示所连 CPU 的参数以及通信统计数据。Parameter 选项卡显示 CPU 的 MAC 地址、CPU 的 IP 地址和 IP 设置以及物理属性。Statistics 选项卡显示发送和接收通信统计数据。

### Communication: Parameter 显示

The screenshot displays the 'Communication' page for a SIMATIC 1200 station. The page is titled 'SIEMENS SIMATIC 1200 station\_1/PLC\_1' and shows the time '04:02:50 pm 26.01.2012'. The user is logged in as 'admin'. The page is divided into two tabs: 'Parameter' (selected) and 'Statistics'. The 'Parameter' tab shows the following information:

- Network connection:**
  - MAC address: 08-00-06-05-91-11
  - Name: picxb1d0ed
- IP parameter:**
  - IP Address: 192.168.0.1
  - Subnet mask: 255.255.255.0
  - Default router: 0.0.0.0
  - IP settings: Retained IP address is set inside of project
- Physical properties:**

Port number	Link status	Settings	Mode
1	OK	automatic	100 MBit/s full-duplex

## Communication: Statistics 显示

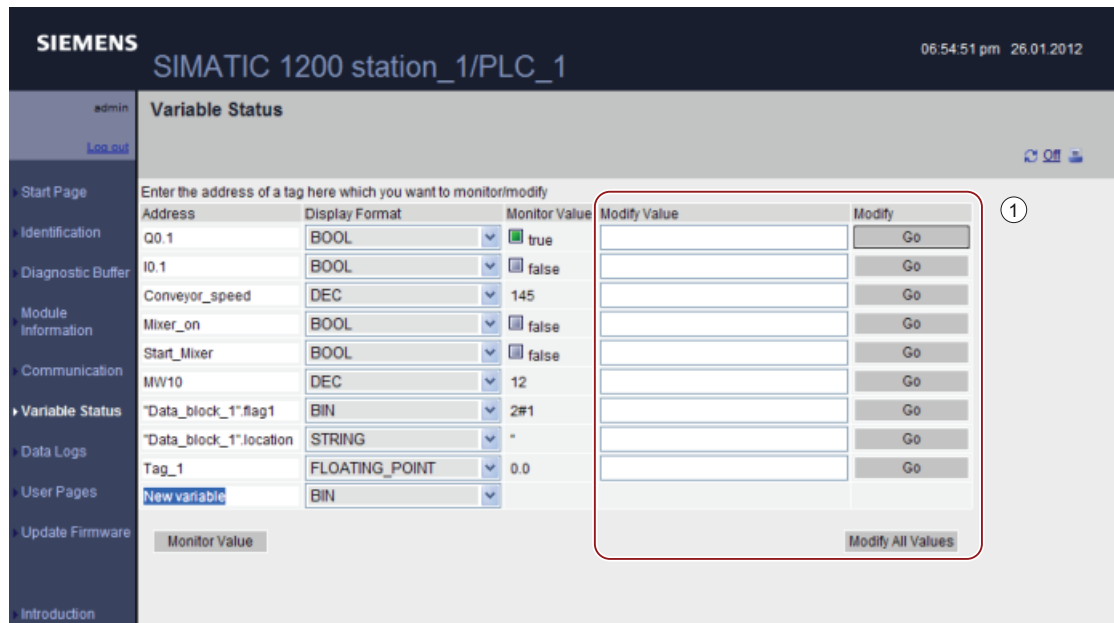


“通信”(Communication) 页面与您是否以“admin”身份登录无关。

## 11.2.9 Variable Status

Variable Status 页面允许您查看 CPU 中的任何 I/O 或存储器数据。可以输入直接地址（如 I0.0）、PLC 变量名或特定数据块的变量。对于数据块变量，应使用双引号将数据块名称括起来。可以为每个监视值选择数据的显示格式。可以继续输入和指定值，只要所需值的数量不超过页面的限制。默认情况下，会自动显示监视值并进行刷新，除非您单击页面右上方区域的“Off”图标。如果已禁用刷新，则可单击“On”重新启用自动刷新。

以“admin”身份登录时，还可以修改数据值。在相应的“Modify Value”域中输入任何要设置的值。单击值旁边的“Go”按钮将该值写入 CPU。还可以输入多个值，然后单击“Modify All Values”将所有值写入 CPU。



① 仅当以“admin”用户的身份登录时，才能看到和访问“修改值”(Modify Value) 功能。

如果离开“Variable Status”页面然后返回，则“Variable Status”页面不会保留您的输入内容。可以为页面加书签，然后返回该书签，这样便可看到相同的条目。如果不为页面加书签，则必须重新输入变量。

## 说明

使用标准的“变量状态”页时，请注意以下问题：

- “变量状态”页不允许用户修改超过 198 个字符的字符串。
- 使用指数表示法在“变量状态”页输入 Real 或 LReal 数据类型的值时：
  - 要输入具有正指数（如 +3.402823e+25）的实数值（Real 或 LReal），请按以下格式之一输入值：
    - +3.402823e25
    - +3.402823e+25
  - 要输入具有负指数（如 +3.402823e-25）的实数值（Real 或 LReal），请按如下形式输入值：
    - +3.402823e-25
  - 确保指数表示法实数值的尾数部分包含小数点。不包含小数点将导致值被改为意外的整数值。例如，请输入 -1.0e8，而不是 -1e8。
- “变量状态”页只支持输入 15 位数的 LReal 值（小数点位置不限）。输入 15 位以上的值会导致舍入错误。

“变量状态”页的限制：

- 每页可输入的最大变量数为 50。
- 与 **Variable Status** 页面对应的 URL 的最大字符数为 2083。可以在浏览器的地址栏中查看表示当前变量页面的 URL。
- 对于字符显示格式，如果实际 CUP 值不是浏览器所解释的有效 ASCII 字符，则页面会显示十六进制值。

---

**说明**

如果某个变量名称因包含特殊字符而被拒绝作为“变量状态”(Variable Status) 页面上的条目，则可用双引号将该变量名括起来。大多数情况下，“变量状态”(Variable Status) 页面随后便能识别该变量名称。

---

### 11.2.10

#### Data Logs

Data Logs 页面允许您查看或下载指定数目的数据日志条目。以“admin”身份登录时，还可以在下载之后清空或删除这些条目。Web 服务器会以逗号分隔值 (CSV) 文件格式将数据日志下载到您的 PC。

“数据日志”(Data Logs) 页面最多显示 40 个数据日志文件。如果存在 40 个以上的数据日志文件，“数据日志”(Data Logs) 页面将显示前 40 个最先创建的文件。

---

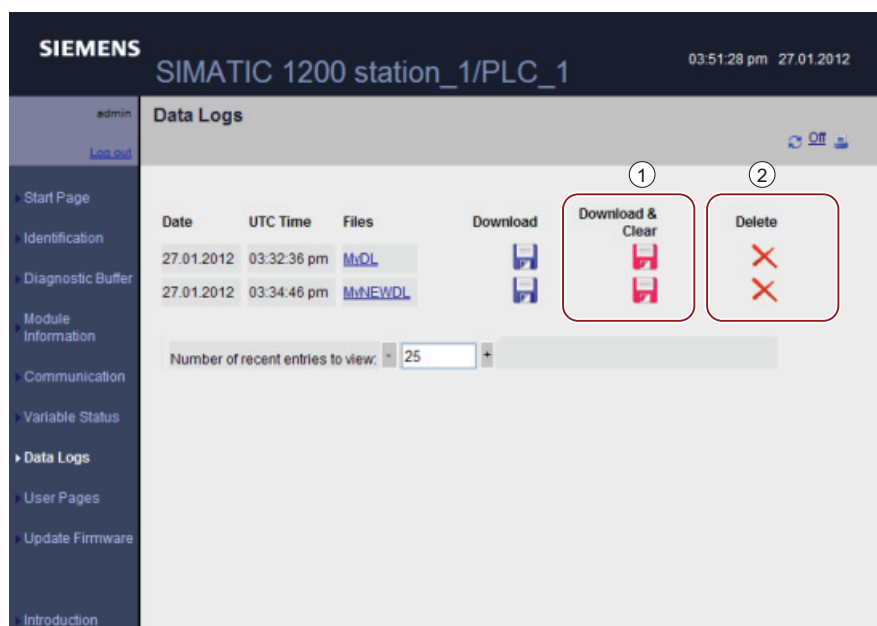
**说明**

**数据日志时间戳显示为系统时间，而不是本地时间**

CPU 在系统时间 (页 87) 中写入数据日志条目的时间戳并且 Web 服务器的“数据日志”(Data Logs) 页面显示采用系统时间的数据日志的时间戳。

---





- ① 如果未以“admin”用户的身份登录，则“下载并清除”(Download & Clear) 选项不可用。
- ② 如果未以“admin”用户的身份登录，则“删除”(Delete) 选项不可用。

#### 说明

数据日志文件采用 USA/UK 的逗号分隔值格式 (CSV)。要在非 USA/UK 系统上的 Excel 中打开该文件，必须采用特定设置将其导入 Excel 中 (页 604)。

#### 最新条目： 下载指定数量的最新数据条目

设定要下载的最新条目的数目，然后单击数据日志名称以开始下载指定数目的记录。在输出 .csv 文件中，会按条目的降序对数据条目进行排序。Windows 会提示您打开或保存日志文件。

默认情况下，要查看的最新条目数默认设置为 25。您可以在“要查看的最新条目数”(Number of recent entries to view) 字段中更改该值，方法是输入所需的值，也可以使用“+”或“-”按钮来递增或递减该值。

#### 说明

DataLogCreate (页 339) 指令的 Records 参数定义每个数据日志文件的最大条目数。

### 下载包含所有数据条目的日志文件

要下载整个日志文件，请单击特定日志文件对应的“下载”(Download) 图标。Windows 会提示您打开或保存日志文件。

在输出 .csv 文件中包含所有数据条目，并且会按条目编号的升序进行排序，但数据日志已满或旧条目（编号较低的条目）被新条目（编号较高的条目）覆盖的情况除外。

### 下载和清空日志文件

要下载日志文件，然后清除所有数据条目，必须以“admin”用户的身份登录。然后单击特定日志文件对应的“下载并清空”(Download & Clear) 图标。Windows 会提示您打开或保存日志文件。

下载完成后，将在 PLC 存储的数据日志文件的标题条目之后插入新的“//END”行。这样可以有效清空数据日志供将来内部 PLC 处理之用，但对该文件的后续下载会在第一个“//END”行上方插入新数据条目。

---

#### 说明

##### 数据日志 .csv 文件“//END”标记

“//END”.csv 文件结束标记仅用于前（最大条目数 - 1）个条目，以标识文件的逻辑结束位置。逻辑结束标记之后，文件可能会包含被 Excel 解释为附加数据条目的数据。应搜索第一个“//END”，然后删除它及其后的所有条目。如果不存在逻辑结束标记，可以通过条目编号对数据行进行排序。

---

### 删除日志文件

要删除日志文件，必须以“admin”用户的身份登录。然后单击特定日志文件对应的“删除”(Delete) 图标。Web 服务器随后便会删除所选的日志文件。

### 更多信息

有关如何使用数据日志指令进行编程的信息，请参见 数据记录 (页 338) 一章。

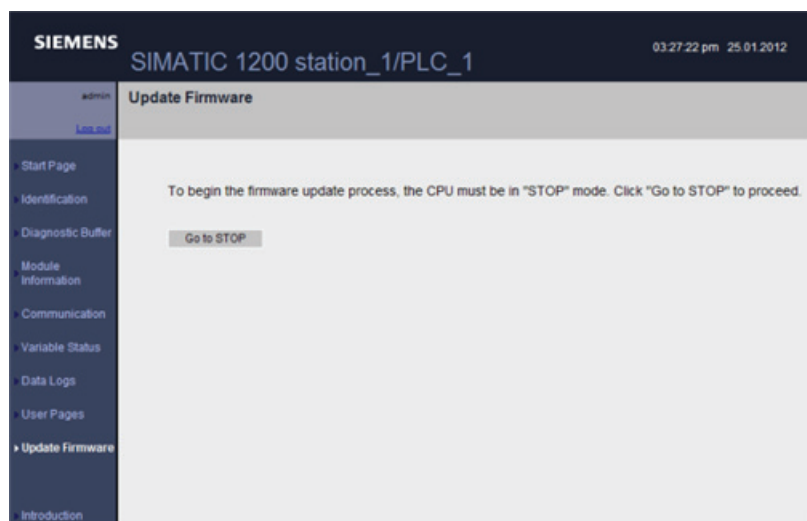
## 11.2.11 更新固件

“更新固件”(Update Firmware) 页面可让“admin”用户通过文件更新 CPU 固件。

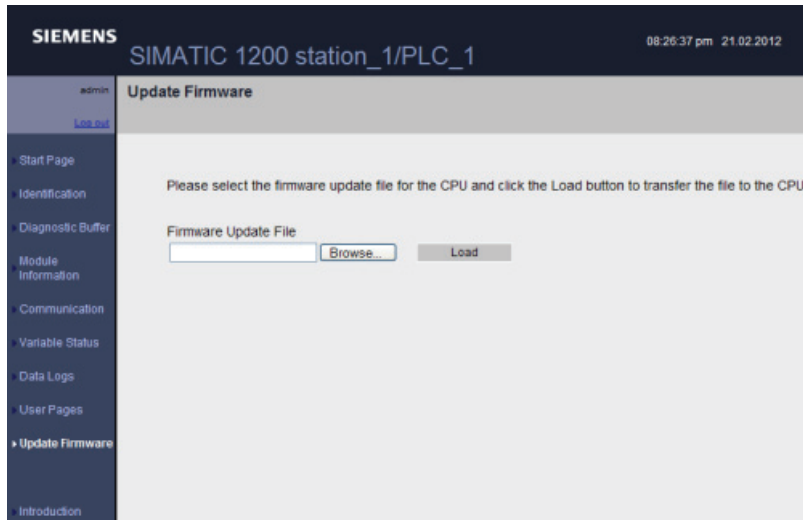
### 说明

只能使用“更新固件”(Update Firmware) 功能更新版本 3.0 及更高版本的 S7-1200 CPU。  
Web 服务器使用“https”协议执行固件更新。

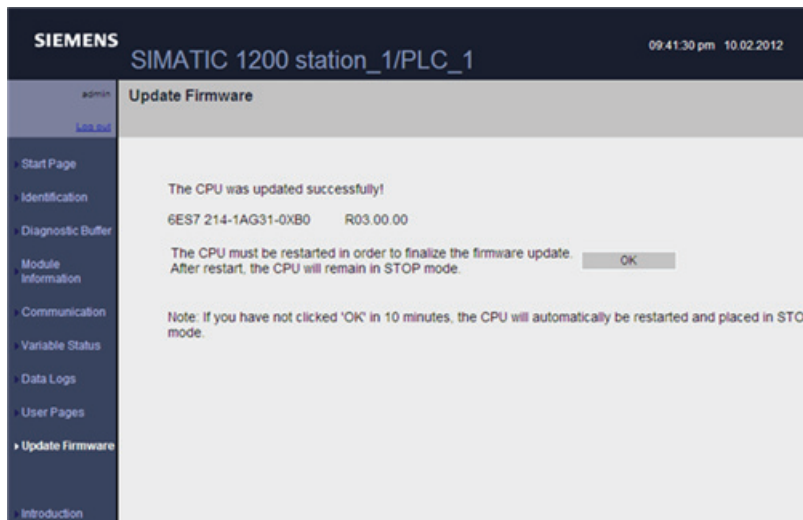
CPU 必须处于 STOP 模式下才能执行固件更新。如果 CPU 尚未处于 STOP 模式下，Web 服务器会提示将 CPU 切换为停止模式。



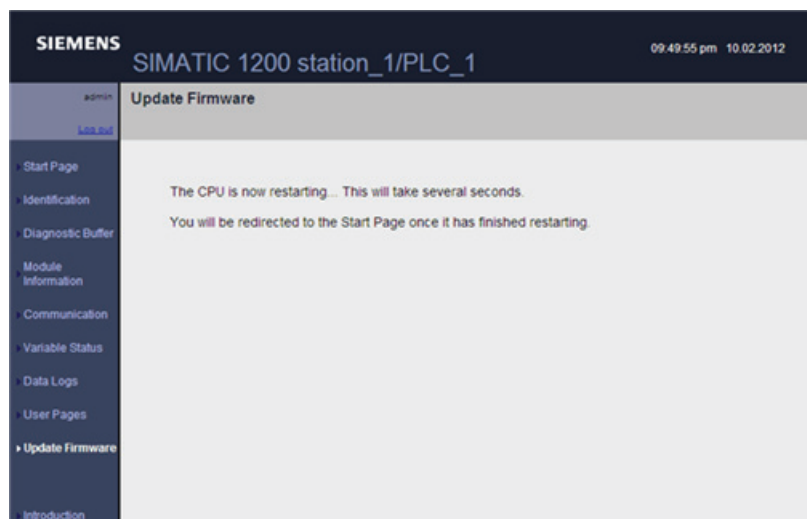
当 CPU 处于 STOP 模式下时，可导航至某个文件并选择该文件，从中加载固件版本更新。固件更新可在客户支持网站 (<http://support.automation.siemens.com/CN>) 上使用。在从网站将相应固件更新下载到计算机后，就可在计算机上浏览至某个文件并选择该文件进行更新。



在更新期间，“更新固件”(Update Firmware) 页面会显示消息，表明更新正在进行。在更新完成后，“更新固件”(Update Firmware) 页面会显示更新的固件的订货号和版本号，并提示您确认是否重启 CPU。



如果在十分钟内未做出响应，CPU 将会自动重启。

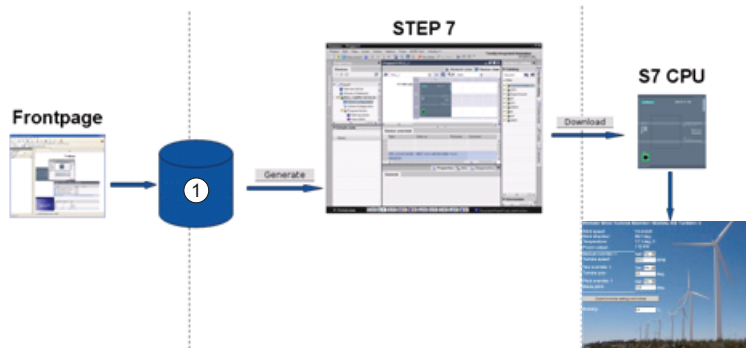


## 11.3 用户定义的 Web 页面

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 页面菜单访问的位置将这些页面下载到 CPU。该过程涉及到以下几项任务：

- 使用 HTML 编辑器（如 Microsoft Frontpage）创建 HTML 页面。（页 562）
- 将 AWP 命令包含在 HTML 代码的 HTML 注释中（页 563）：AWP 命令是 Siemens 提供用于访问 CPU 信息的固定命令集。
- 将 STEP 7 组态为读取和处理 HTML 页面（页 577）
- 基于 HTML 页面生成块（页 577）
- 对 STEP 7 进行编程设计，以控制 HTML 页面的使用（页 579）
- 编译程序块并将其下载到 CPU（页 580）
- 通过 PC 访问用户定义的 Web 页面（页 581）

该过程的图示如下：



① 具有嵌入式 AWP 命令的 HTML 文件

### 11.3.1 创建 HTML 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（页 581）。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（页 111）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（页 563）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。

## 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。

### 11.3.2 S7-1200 Web 服务器支持的 AWP 命令

S7-1200 Web 服务器提供了以 HTML 注释形式嵌入用户定义的 Web 页面中的 AWP 命令，这些命令具有以下用途：

- 读取变量 (页 564)
- 写入变量 (页 565)
- 读取特殊变量 (页 567)
- 写入特殊变量 (页 569)
- 定义枚举类型 (页 571)
- 为枚举类型分配变量 (页 572)
- 创建片段数据块 (页 573)

#### 一般语法

除读取变量的命令之外，AWP 命令的语法如下：

```
<!-- AWP_ <command name and parameters> -->
```

AWP 命令与典型的 HTML 表单命令一起使用时，可将变量写入 CPU。

接下来的各页面在介绍 AWP 命令时采用如下惯例：

- 方括号 [] 中包含的项为可选项。
- 尖括号 < > 中包含的项是要指定的参数值。
- 引号是命令的文字部分。它们必须按所示的形式出现。
- 根据具体用法，变量或数据块名称中的特殊字符 必须进行转义或用引号括号来 (页 575)。

使用文本编辑器或 HTML 编辑模式可将 AWP 命令插入页面中。

## AWP 命令汇总

接下来的各主题将详细介绍每个 AWP 命令的用法，但此处先对这些命令进行简单汇总：

## 读取变量

```
:=<Varname>:
```

## 写入变量

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

该 AWP 命令只是声明 Name 子句中的变量可写入。HTML 代码将按 HTML 表单中 <input>、<select> 或其它 HTML 语句中的名称写入变量。

## 读取特殊变量

```
<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->
```

## 写入特殊变量

```
<!-- AWP_In_Variable Name='<Type>:<Name>' [Use='<Varname>']-->
```

## 定义枚举类型

```
<!--
  AWP_Enum_Def Name='<Enum type name>' Values='<Value>, <Value>, ...
  . ' -->
```

## 引用枚举类型

```
<!-- AWP_Enum_Ref Name='<VarName>' Enum='<EnumType>' -->
```

## 创建片段

```
<!-- AWP_Start_Fragment Name='<Name>' [Type=<Type>] [ID=<id>] -->
```

## 导入片段

```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

## 11.3.2.1

## 读取变量

用户定义的 Web 页面可以从 CPU 中读取变量（PLC 变量）。

## 语法

```
:=<Varname>:
```



## 参数

<Varname>	要读取的变量，可以是 STEP 7 程序中的 PLC 变量名称、数据块变量、I/O 或可寻址存储器。对于存储器、I/O 地址或别名 (页 575)，请勿使用引号将变量名称括起来。对于 PLC 变量，请使用双引号将变量名称括起来。对于数据块变量，只用双引号将块名称括起来。变量名称位于引号外。请注意，应使用数据块名称，而不是数据块编号。
-----------	---

## 示例

```

:= "Conveyor_speed" := "My_Data_Block".flag1:
:= I0.0:
:= MW100:

```

### 读取具有别名的变量的示例

```

<!-- AWP_Out_Variable Name='flag1' Use=' "My_Data_Block".flag1' -->
:= flag1:

```

### 说明

主题 对变量引用使用别名 (页 570) 中介绍了如何定义 PLC 变量和数据块变量的别名。

如果变量名称或数据块名称包含特殊字符，则必须使用附加引号或转义字符，如主题 处理包含特殊字符的变量名称 (页 575) 中所述。

### 11.3.2.2

#### 写入变量

用户定义的页面可以将数据写入 CPU。这是通过 AWP 命令识别 CPU 中可从 HTML 页面写入的变量来实现的。该变量必须通过 PLC 变量名称或块变量名称指定。可以在一条语句中声明多个变量名称。要将数据写入 CPU，应使用标准 HTTP POST 命令。

典型用法是：使用与可写入的 CPU 变量对应的文本输入域或选择列表选项来在 HTML 页面中设计表单。与所有用户定义的页面相同，之后将通过 STEP 7 生成块，从而在 STEP 7 程序中包含这些块。admin 用户随后访问该页面并在输入字段中输入数据或从选择列表中选择选项时，Web 服务器会将输入转换为变量的相应数据类型，并将该值写入 CPU 中的变量。请注意，HTML 输入域和 HTML 选择列表的 name 子句所使用的语法通常也用于 AWP\_In\_Variable 命令的 name 子句。通常用单引号将名称括起来，但如果引用数据块，则应将数据块名称用双引号括起来。

有关表单管理的详细信息，请参考 HTML 文档。

## 语法

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

## 参数

<Varname1>	如果未提供 Use 子句，则 Varname1 表示要写入的变量。它可以是 STEP 7 程序中的 PLC 变量名称，或者是特定数据块中的变量。 如果提供了 Use 子句，则 Varname1 为 <Varname2> 中引用的变量的备用名称 (页 570)。它是 HTML 页面中的本地名称。
<Varname2>	如果提供了 Use 子句，则 Varname2 表示要写入的变量。它可以是 STEP 7 程序中的 PLC 变量名称，或者是特定数据块中的变量。

对于 Name 子句和 Use 子句，必须用单引号将整个名称括起来。在单引号中，用双引号将 PLC 变量括起来，并用双引号将数据块名称括起来。数据块名称括在双引号之内，但数据块变量名称不在双引号之内。请注意，对于数据块变量，应使用块名称，而不是数据块编号。

## HTML 输入域的用法示例

```
<!-- AWP_In_Variable Name='"Target_Level"' -->
<form method="post">
<p>Input Target Level: <input name='"Target_Level"' type="text"
/> </p>
</form>

<!-- AWP_In_Variable Name='"Data_block_1".Braking' -->
<form method="post">
<p>Braking: <input name='"Data_block_1".Braking' type="text" />
%</p>
</form>

<!-- AWP_In_Variable Name='"Braking"'
Use='"Data_block_1".Braking' -->
<form method="post">
<p>Braking: <input name='"Braking"' type="text" /> %</p>
</form>
```

## HTML 选择列表的用法示例

```
<!-- AWP_In_Variable Name='"Data_block_1".ManualOverrideEnable'-->
<form method="post">
<select name='"Data_block_1".ManualOverrideEnable'>
<option value=":"Data_block_1".ManualOverrideEnable:> </option>
<option value=1>Yes</option>
```

```
<option value=0>No</option>  
</select><input type="submit" value="Submit setting" /></form>
```

---

**说明**

只有 admin 用户才能将数据写入 CPU。如果用户未以 admin 用户的身份登录，则会忽略这些命令。

---

如果变量名称或数据块名称包含特殊字符，则必须使用附加引号或转义字符，如主题“处理包含特殊字符的变量名称 (页 575)”中所述。

**11.3.2.3****读取特殊变量**

Web 服务器可以从 PLC 中读取要存储在 HTTP 响应标头的特殊变量中的值。例如，您可能要从 PLC 变量中读取路径名，以使用 HEADER:Location 特殊变量将 URL 重新定向到其它位置。

**语法**

```
<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->
```

## 参数

<Type>	<p>特殊变量的类型有：</p> <p>HEADER</p> <p>COOKIE_VALUE</p> <p>COOKIE_EXPIRES</p>
<Name>	<p>要获取所有 HEADER 变量名称的列表，请参考 HTTP 文档。下面给出了几个示例：</p> <p>Status: 响应代码</p> <p>Location: 重定向的路径</p> <p>Retry-After: 预计服务在多长时间对请求客户机不可用</p> <p>对于 COOKIE_VALUE 和 COOKIE_EXPIRES 类型，&lt;Name&gt; 是特定 cookie 的名称。</p> <p>COOKIE_VALUE:name: 指定的 cookie 的值</p> <p>COOKIE_EXPIRES:name: 指定的 cookie 的到期时间（以秒为单位）</p> <p>必须用单引号或双引号将 Name 子句括起来。</p> <p>如果未指定 Use 子句，则特殊变量名称与 PLC 变量名称相对应。用单引号将完整的 Name 子句括起来，用双引号将 PLC 变量括起来。特殊变量名称和 PLC 变量名称必须完全匹配。</p>
<Varname>	<p>要读入的变量对应的 PLC 变量或数据块变量的名称</p> <p>必须用单引号将 Varname 括起来。在单引号中，用双引号将 PLC 变量或数据块名称括起来。数据块名称括在双引号之内，但数据块变量名称不在双引号之内。请注意，对于数据块变量，应使用块名称，而不是数据块编号。</p>

## 示例

```
<!-- AWP_Out_Variable Name='"HEADER:Status"' -->
```

本示例中，HTTP 特殊变量“HEADER:Status”会接收 PLC 变量“HEADER:Status”的值。如果未指定 Use 子句，则 PLC 变量表中的名称必须与特殊变量的名称完全匹配。

```
<!-- AWP_Out_Variable Name='HEADER:Status' Use='"Status"' -->
```

本示例中，HTTP 特殊变量“HEADER:Status”会接收 PLC 变量“Status”的值。

如果变量名称或数据块名称包含特殊字符，则必须使用附加引号或转义字符，如主题 处理包含特殊字符的变量名称 (页 575)中所述。

### 11.3.2.4 写入特殊变量

Web 服务器可以将 HTTP 请求标头中特殊变量的值写入 CPU。例如，可以将与用户自定义网页相关的 cookie 信息、正在访问页面的用户信息或标题信息存储在 STEP 7 中。Web 服务器可用于访问指定的特殊变量，您以 admin 用户的身份登录后可以将这些变量写入 CPU。

#### 语法

```
<!-- AWP_In_Variable Name='<Type>:<Name>' [Use='<Varname>']-->
```

#### 参数

<Type>	<p>特殊变量的类型有：</p> <p>HEADER</p> <p>SERVER</p> <p>COOKIE_VALUE</p>
<Name>	<p>上面定义的类型中的特殊变量，如下示例中所示：</p> <p>HEADER:Accept: 可接受的内容类型</p> <p>HEADER:User-Agent: 有关发起请求的用户代理的信息。</p> <p>SERVER:current_user_id: 当前用户的 id；如果没有用户登录，则该变量为 0</p> <p>SERVER:current_user_name: 当前用户的名称</p> <p>COOKIE_VALUE:&lt;name&gt;: 指定的 cookie 的值</p> <p>用单引号将 Name 子句括起来。</p> <p>如果未指定 Use 子句，则特殊变量名称与 PLC 变量名称相对应。用单引号将完整的 Name 子句括起来，用双引号将 PLC 变量括起来。特殊变量名称和 PLC 变量名称必须完全匹配。</p> <p>要获取所有 HEADER 变量名称的列表，请参考 HTTP 文档。</p>
<Varname>	<p>要写入特殊变量的 STEP 7 程序中的变量名，它可以是 PLC 变量名称或数据块变量。</p> <p>必须用单引号将 Varname 括起来。在单引号中，用双引号将 PLC 变量或数据块名称括起来。数据块名称括在双引号之内，但数据块变量名称不在双引号之内。请注意，对于数据块变量，应使用块名称，而不是数据块编号。</p>

## 示例

```
<!-- AWP_In_Variable Name='\"SERVER:current_user_id\"' -->
```

在本示例中，Web 页面将 HTTP 特殊变量“SERVER:current\_user\_id”的值写入名称为“SERVER:current\_user\_id”的 PLC 变量中。

```
<!-- AWP_In_Variable Name=SERVER:current_user_id'
Use='\"my_userid\"' -->
```

在本示例中，Web 页面将 HTTP 特殊变量“SERVER:current\_user\_id”的值写入名称为“my\_userid”的 PLC 变量中。

## 说明

只有 admin 用户才能将数据写入 CPU。如果用户未以 admin 用户的身份登录，则会忽略这些命令。

如果变量名称或数据块名称包含特殊字符，则必须使用附加引号或转义字符，如主题“处理包含特殊字符的变量名称 (页 575)”中所述。

## 11.3.2.5

## 对变量引用使用别名

可以在用户定义的 Web 页面中为 In\_Variable 或 Out\_Variable 使用别名。例如，您可以在 HTML 页面中使用不同于 CPU 中的符号名称，也可以使 CPU 中的变量与特殊变量等同。AWP 的 Use 子句就可以实现这一功能。

## 语法

```
<-- AWP_In_Variable Name='<Varname1>' Use='<Varname2>' -->
<-- AWP_Out_Variable Name='<Varname1>' Use='<Varname2>' -->
```

## 参数

<Varname1>	必须将别名或特殊变量名称 Varname1 用单引号或双引号括起来。
<Varname2>	要为其分配别名的 PLC 变量的名称。该变量可以是 PLC 变量、数据块变量或特殊变量。必须用单引号将 Varname2 括起来。在单引号中，用双引号将 PLC 变量、特殊变量或数据块名称括起来。数据块名称括在双引号之内，但数据块变量名称不在双引号之内。请注意，对于数据块变量，应使用块名称，而不是数据块编号。

## 示例

```
<-- AWP_In_Variable Name='SERVER:current_user_id'
Use='"Data_Block_10".server_user' -->
```

本示例中，特殊变量 SERVER:current\_user\_id 将被写入数据块“Data\_Block\_10”中的变量“server\_user”。

```
<-- AWP_Out_Variable Name='Weight'
Use='"Data_Block_10".Tank_data.Weight' -->
```

本示例中，可以在用户定义的 Web 页面的其余部分仅由“Weight”引用数据块结构成员 Data\_Block\_10.Tank\_data.Weight 中的值。

```
<-- AWP_Out_Variable Name='Weight' Use='"Raw_Milk_Tank_Weight"' -
->
```

在本示例中，可以在用户定义的 Web 页面的其余部分仅由“Weight”引用 PLC 变量“Raw\_Milk\_Tank\_Weight”中的值。

如果变量名称或数据块名称包含特殊字符，则必须使用附加引号或转义字符，如主题 处理包含特殊字符的变量名称 (页 575)中所述。

## 11.3.2.6 定义枚举类型

可以使用 AWP 命令在用户定义的页面中定义枚举类型并分配元素。

## 语法

```
<!-- AWP_Enum_Def Name='<Enum type name>' Values='<Value>,
<Value>,... ' -->
```

## 参数

<Enum type name>	枚举类型的名称，用单引号或双引号括起来。
<Value>	<constant>:<name> constant 表示枚举类型分配的数字值。总数不受限制。 name 是分配给枚举元素的值。

请注意，应将整个枚举值分配字符串用单引号括起来，而将每个单独的枚举类型元素分配用双引号括起来。对于用户定义的 Web 页面，枚举类型定义在全局范围内有效。如果已在语言文件夹 (页 594)中创建用户定义的 Web 页面，则对于语言文件夹中的所有页面，枚举类型定义在全局范围内有效。

## 示例

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank is full", 2:"Tank is empty"' -->
```

## 11.3.2.7 通过枚举类型引用 CPU 变量

可以将 CPU 中的变量分配给枚举类型。在执行读操作 (页 564) 或写操作 (页 565) 时，可在用户定义的 Web 页面中的其它位置使用该变量。在读操作中，Web 服务器将用相应的枚举文本值替换从 CPU 中读取的数字值。在写操作中，Web 服务器将在向 CPU 写入值之前用与文本对应的枚举整型值来替换文本值。

## 语法

```
<!-- AWP_Enum_Ref Name='<Varname>' Enum="<EnumType>" -->
```

## 参数

<Varname>	与枚举类型相关联的 PLC 变量或数据块变量的名称，或 PLC 变量的别名 (页 570) (如已声明)。 必须用单引号将 Varname 括起来。在单引号中，用双引号将 PLC 变量或数据块名称括起来。请注意，对于数据块变量，应使用块名称，而不是数据块编号。数据块名称括在双引号之内，但数据块变量名称不在双引号之内。
<EnumType>	枚举类型的名称，必须用单引号或双引号括起来。

枚举类型引用的适用范围为当前片段。

## 声明示例

```
<!-- AWP_Enum_Ref Name='Alarm' Enum="AlarmEnum" -->
```

## 变量读取的用法示例

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank is full", 2:"Tank is empty"' -->
<!-- AWP_Enum_Ref Name='Alarm' Enum="AlarmEnum" -->
...
<p>The current value of "Alarm" is := "Alarm":</p>
```

如果 CPU 中“Alarm”的值为 2，则 HTML 页面会显示“The current value of “Alarm” is Tank is empty”，这是因为枚举类型定义 (页 571) 将文本字符串“Tank is empty”分配给了数字值 2。



### 变量写入的用法示例

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank is full", 2:"Tank is empty"' -->
<!-- AWP_In_Variable Name='Alarm' -->
<!-- AWP_Enum_Ref Name='Alarm' Enum='AlarmEnum' -->
...
<form method="POST">
<p><input type="hidden" name="Alarm" value="Tank is full" /></p>
<p><input type="submit" value='Set Tank is full' /></p>
</form>
```

由于枚举类型定义(页 571)将“Tank is full”分配给了数字值 1，因此，值 1 被写入 CPU 中的 PLC 变量“Alarm”。

请注意，AWP\_In\_Variable 声明中的 Name 子句必须与 AWP\_Enum\_Ref 声明中的 Name 子句完全对应。

### 在变量写入过程中使用别名的用法示例

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank is full", 2:"Tank is empty"' -->
<!-- AWP_In_Variable Name='Alarm'
Use='Data_block_4.Motor1.Alarm'-->
<!-- AWP_Enum_Ref Name='Alarm' Enum='AlarmEnum' -->
...
<form method="POST">
<p><input type="hidden" name="Alarm" value="Tank is full" /></p>
<p><input type="submit" value='Set Tank is full' /></p>
</form>
```

由于枚举类型定义(页 571)将“Tank is full”分配给了数字值 1，因此，值 1 被写入 CPU 中与数据块“Data\_Block\_4”的 PLC 变量“Motor1.Alarm”对应的别名“Alarm”。

如果变量名称或数据块名称包含特殊字符，则必须使用附加引号或转义字符，如主题处理包含特殊字符的变量名称(页 575)中所述。

#### 11.3.2.8 创建片段

单击 Web 服务器的“CPU 属性”(CPU Properties) 中的“生成块”(Generate blocks) 时，STEP 7 会将用户定义的 Web 页面转换和存储为控制 DB 和片段 DB。可以为特定页面或特定页面部分建立特殊片段。可以使用 AWP 命令“Start\_Fragment”通过名称和编号来标识这些片段。执行 AWP\_Start\_Fragment 命令之后，页面中的所有内容都属于该片段，直到再发出 AWP\_Start\_Command 或到达文件结尾。

## 语法

```
<!-- AWP_Start_Fragment Name='<Name>' [Type=<Type>] [ID=<id>] -->
```

## 参数

<Name>	<p>文本字符串： 片段 DB 的名称</p> <p>片段名称必须以字母或下划线开头，并由字母、数字和下划线组成。 片段名称是以下形式的正则表达式： [a-zA-Z_][a-zA-Z_0-9]*</p>
<Type>	<p>“manual”或“automatic”</p> <p><b>manual:</b> STEP 7 程序必须请求该片段并可相应地做出响应。 必须使用 STEP 7 和控制 DB 变量来控制片段的操作。</p> <p><b>automatic:</b> Web 服务器自动处理片段。</p> <p>如果未指定类型参数，则默认设置为“automatic”。</p>
<id>	<p>整型标识号。 如果未指定 ID 参数，则 Web 服务器会默认分配一个数字。 对于手动片段，应将 ID 设为较小的数字。 ID 是 STEP 7 程序控制手动片段的方法。</p>
<模式>	<p>“可见”或“隐藏”</p> <p><b>可见:</b> 片段的内容将显示在用户定义的 Web 页面上。</p> <p><b>隐藏:</b> 片段的内容将不会显示在用户定义的 Web 页面上。</p> <p>如果未指定类型参数，则默认设置为“可见”(visible)。</p>

## 手动片段

如果为用户定义的 Web 页面或页面部分创建手动片段，则 STEP 7 程序必须控制片段的发送时间。 手动控制时，STEP 7 程序必须在控制 DB 中为用户定义的页面设置相应的参数，然后针对修改的控制 DB 调用 WWW 指令。 要了解控制 DB 的结构以及如何处理各个页面和片段，请参见主题 高级用户定义 Web 页面控制 (页 597)。

## 11.3.2.9

## 导入片段

可以根据部分 HTML 代码创建指定的片段，然后将该片段导入用户定义的 Web 页面集的其他位置。 例如，有一个包含起始页面的用户定义的 Web 页面集，还有几个可通过该起始页面中的链接访问的其他 HTML 页面。 假定每个单独的页面都要在页面上显示公司徽标。 那么，您可以创建片段 (页 573)，用来加载公司徽标的图像。 每个单独的 HTML 页

面都可以导入该片段以显示公司徽标。您可以使用 `AWP Import_Fragment` 命令来执行该操作。片段的 HTML 代码只存在于一个片段中，但您可以根据需要将该片段 DB 多次导入所选的众多 Web 页面中。

## 语法

```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

## 参数

<Name>	文本字符串：要导入的片段 DB 的名称
--------	---------------------

## 示例

用来创建显示图像的片段的 HTML 代码摘录：

```
<!-- AWP_Start_Fragment Name='My_company_logo' --><p></p>
```

导入了显示徽标图像的片段的其它 .html 文件中的 HTML 代码摘录：

```
<!-- AWP_Import_Fragment Name='My_company_logo' -->
```

这两个 .html 文件（一个文件创建片段，另一个文件导入片段）都位于您在 STEP 7 中组态用户定义的页面 (页 577) 时所定义的文件夹结构中。

### 11.3.2.10 组合定义

声明要在用户定义的 Web 页面中使用的变量时，可以将变量声明与变量的别名 (页 570) 组合在一起。还可以在一条语句中声明多个 `In_Variable` 以及多个 `Out_Variable`。

## 示例

```
<!-- AWP_In_Variable Name='Level', Name='Weight',  
Name='Temp' -->  
<--! AWP_Out_Variable Name='HEADER:Status', Use='Status',  
Name='HEADER:Location', Use='Location',  
Name='COOKIE_VALUE:name', Use='my_cookie' -->  
<!-- AWP_In_Variable Name='Alarm' Use='Data_block_10.Alarm' -->
```

### 11.3.2.11 处理包含特殊字符的变量名称

在用户定义的 Web 页面中指定变量名称时，如果变量名称中包含具有特殊含义的字符，则需要特别注意。

## 读取变量

可以使用以下语法 读取变量 (页 564):

**:=<Varname>:**

在读取变量时下列规则适用:

- 对于 PLC 变量表中的变量名称, 应将变量名称用双引号括起来。
- 对于作为数据块变量的变量名称, 应将该数据块名称用双引号括起来。变量在引号外。
- 对于直接 I/O 地址、存储器地址或别名等变量名称, 请勿使用引号将读取变量括起来。
- 对于包含反斜杠的变量名称或数据块变量名称, 应在反斜杠前面再加一个反斜杠。
- 如果变量名称或数据块变量名称包含冒号、小于号、大于号或 & 号, 则为读取的变量定义不含特殊字符的别名, 并通过该别名读取变量。在 Use 子句的变量名称中, 在冒号前加上反斜杠。

表格 11-1 读取变量示例

数据块名称	变量名称	读取命令
不适用	ABC:DEF	<code>&lt;!--AWP_Out_Variable Name='special_tag' Use ='"ABC:DEF"' --&gt; :=special_tag:</code>
不适用	T\	<code>:= "T\\":</code>
不适用	A \B 'C :D	<code>&lt;!--AWP_Out_Variable Name='another_special_tag' Use='\"A \\B \\'C :D\"' --&gt; :=another_special_tag:</code>
不适用	a<b	<code>&lt;!--AWP_Out_Variable Name='a_less_than_b' Use='\"a&lt;b\"' --&gt; :=a less than b:</code>
Data_block_1	Tag_1	<code>:= "Data_block_1".Tag_1:</code>
Data_block_1	ABC:DEF	<code>&lt;!-- AWP_Out_Variable Name='special_tag' Use='\"Data_block_1\".ABC\\:DEF' --&gt; :=special_tag:</code>
DB A' B C D\$ E	Tag	<code>:= "DB A' B C D\$ E".Tag:</code>
DB:DB	Tag:Tag	<code>&lt;!--AWP_Out_Variable Name='my_tag' Use = '\"DB:DB\".Tag\\:Tag' --&gt; :=my_tag:</code>

## Name 和 Use 子句

AWP 命令 AWP\_In\_Variable、AWP\_Out\_Variable、AWP\_Enum\_Def、AWP\_Enum\_Ref、AWP\_Start\_Fragment 和 AWP\_Import\_Fragment 含有 Name 子句。<input> 和 <select> 等 HTML 表单命令也含有 name 子句。AWP\_In\_Variable 和 AWP\_Out\_Variable 还可以含有 Use 子句。对于所有命令，在特殊字符处理方面 Name 和 Use 子句的语法相同：

- 为 Name 或 Use 子句提供的文本必须用单引号括起来。如果括起来的名称是 PLC 变量或数据块名称，则应使用单引号将整个子句括起来。
- 在 Name 或 Use 子句中，数据块名称和 PLC 变量名称必须用双引号括起来。
- 如果变量名称或数据块名称包含单引号字符或反斜杠，应使用反斜杠将该字符转义。反斜杠在 AWP 命令编译器中是转义字符。

表格 11-2 Name 子句示例

数据块名称	变量名称	Name 子句可选形式
不适用	ABC'DEF	Name=' "ABC\'DEF" '
不适用	A \B 'C :D	Name=' "A \\B \'C :D" '
Data_block_1	Tag_1	Name=' "Data_block_1".Tag_1'
Data_block_1	ABC'DEF	Name=' "Data_block_1".ABC\'DEF'
Data_block_1	A \B 'C :D	Name=' "Data_block_1".A \\B \'C :D'
DB A' B C D\$ E	Tag	Name=' "DB A\' B C D\$ E".Tag'

Use 子句遵从 Name 子句的约定。

### 说明

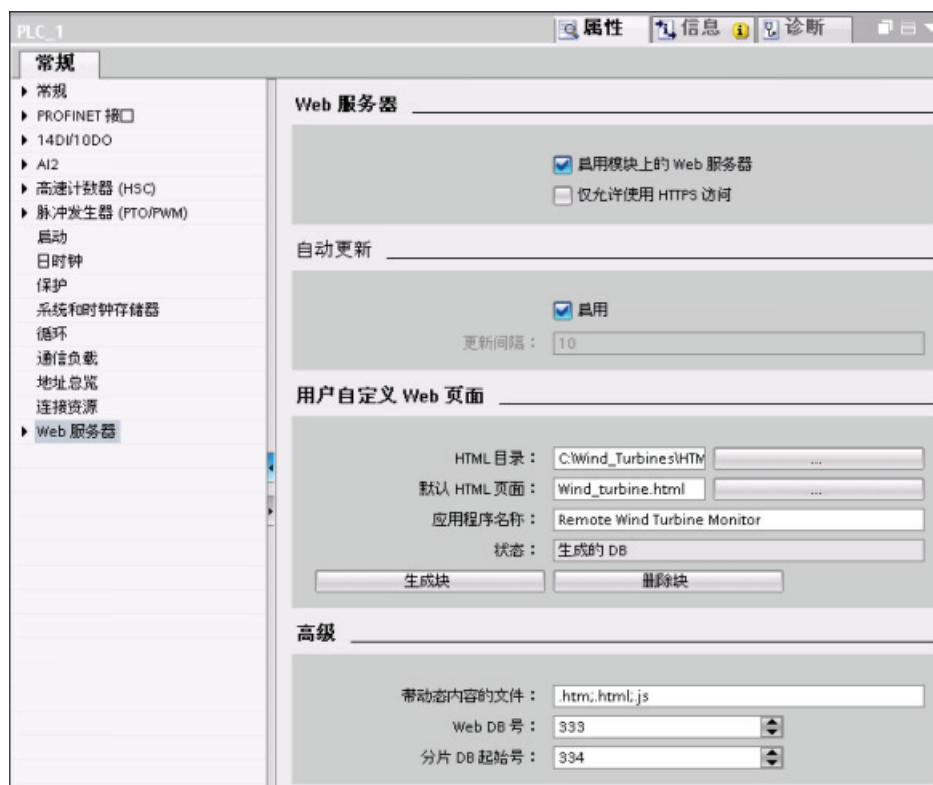
无论在 HTML 页面中使用什么字符，都应将 HTML 页面的字符集设置为 UTF-8 并在编辑器中使用 UTF-8 字符编码对其进行保存。

## 11.3.3 组态用户定义 Web 页面的使用

要通过 STEP 7 组态用户定义的 Web 页面，请执行以下步骤：

1. 在设备组态视图中选择 CPU。

2. 在巡视窗口中显示该 CPU 的“Web server”属性。



3. 如果尚未选择，则选中“Enable Web server”复选框。
4. 选择“仅允许使用 HTTPS 访问”(Permit access only with HTTPS) 确保 Web 服务器使用加密的通信，并提高 Web 可访问 CPU 的安全性。
5. 输入或浏览到 PC 上保存 HTML 默认页面（起始页面）的文件夹的名称。
6. 输入该默认页面的名称。
7. 为应用程序提供名称（可选）。应用程序名称用于对 Web 页面进一步分类或分组。如果应用程序名称存在，URL 将以以下格式出现：`http://ww.xx.yy.zz/awp/<应用程序名称>/<页面名称>.html`。
8. 指定用以检查是否存在 AWP 命令的文件扩展名。默认情况下，STEP 7 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。
9. 保留默认的 Web DB 编号，或输入适合的编号。这是控制 Web 页面显示的控制 DB 的 DB 编号。
10. 保留默认的片段 DB 起始编号，或输入适合的编号。这是首个包含 Web 页面的片段 DB。

## 生成程序块

单击“生成块”(Generate blocks) 按钮时，STEP 7 会基于指定的 HTML 源目录下的 HTML 页面生成数据块，以及一个用于 Web 页面操作的控制数据块。可以根据应用的需要设置这些属性 (页 579)。STEP 7 还会生成一组片段数据块，以保存所有 HTML 页面的显示。在您生成数据块时，STEP 7 会更新属性，以显示控制数据块编号和首个片段数据块编号。生成该数据块之后，用户定义的 Web 页面就会成为 STEP 7 程序的一部分。与这些页面对应的块会出现在项目导航树中“程序块”(Program blocks) 下“系统块”(System blocks) 文件夹中的“Web 服务器”(Web server) 文件夹中。

## 删除程序块

要删除先前生成的数据块，请单击“Delete data blocks”按钮。STEP 7 将从项目中删除与用户定义的 Web 页面相对应的控制数据块和所有片段数据块。


### 11.3.4 针对用户定义 Web 页面编写 WWW 指令

STEP 7 用户程序必须包含并执行 WWW 指令，以便能够通过标准 Web 页面访问用户定义 Web 页面。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。在组态用户定义 Web 页面 (页 577) 时单击“Create blocks”按钮后，STEP 7 便会创建控制数据块。

## 编写 WWW 指令

要通过标准 Web 页面访问用户定义的 Web 页面，STEP 7 程序必须执行 WWW 指令。您可能希望用户定义的 Web 页面仅在应用程序要求和首选项指定的一些情况下可用。在这种情况下，程序逻辑可控制何时调用 WWW 指令。

表格 11-3 WWW 指令

LAD/FBD	SCL	说明
	<pre>ret_val := WWW(     ctrl_db:=_uint_in_);</pre>	提供从标准 Web 页面访问用户定义的 Web 页面的权限

必须提供控制数据块输入参数 (CTRL\_DB)，该参数对应于控制 DB 的整数 DB 编号。为用户定义的 Web 页面创建块后，可在 CPU 的“Web 服务器”(Web Server) 属性中找到此控制 DB 块编号 (称为 Web DB 编号)。输入整数 DB 编号作为 WWW 指令的 CTRL\_DB 参数。返回值 (RET\_VAL) 包含函数结果。请注意，WWW 指令异步执行，

RET\_VAL 输出的初始值可能为 0，但这不能说明后来不会发生错误。程序会检查控制 DB 的状态，以确定应用程序是否已成功启动，或者通过随后对 WWW 的调用来检查 RET\_VAL。

表格 11-4 返回值

RET_VAL	说明
0	无错误
16#00yx	<p>x: 相关位声称的请求处于等待状态:</p> <p>x=1: 请求 0</p> <p>x=2: 请求 1</p> <p>x=4: 请求 2</p> <p>x=8: 请求 3</p> <p>可以对 x 值进行逻辑或运算，以说明多个请求处于等待状态。例如，如果 x = 6，则说明请求 1 和请求 2 处于等待状态。</p> <p>y: 0: 没有错误; 1: 存在错误，并且已在控制 DB 中置位“last_error”</p>
16#803a	未装载控制 DB。
16#8081	控制 DB 的类型、格式或版本错误。
16#80C1	没有资源可用于初始化 Web 应用程序。

## 控制 DB 的使用

单击“生成块”(Generate blocks) 后，STEP 7 将创建控制数据块，并在用户定义的 Web 页面属性中显示控制 DB 编号。您可以在项目导航树中的“程序块”(Program blocks) 文件夹下找到该控制 DB。

通常情况下，STEP 7 程序会直接使用“生成块”(Generate blocks) 过程中创建的控制 DB，而不进行任何处理。但是，STEP 7 用户程序可以在控制 DB 中设置全局命令，以禁用 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 (页 577) 的用户定义页面，STEP 7 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。有关这些高级任务的信息，请参见主题 高级用户定义 Web 页面控制 (页 597)。

### 11.3.5 将程序块下载到 CPU

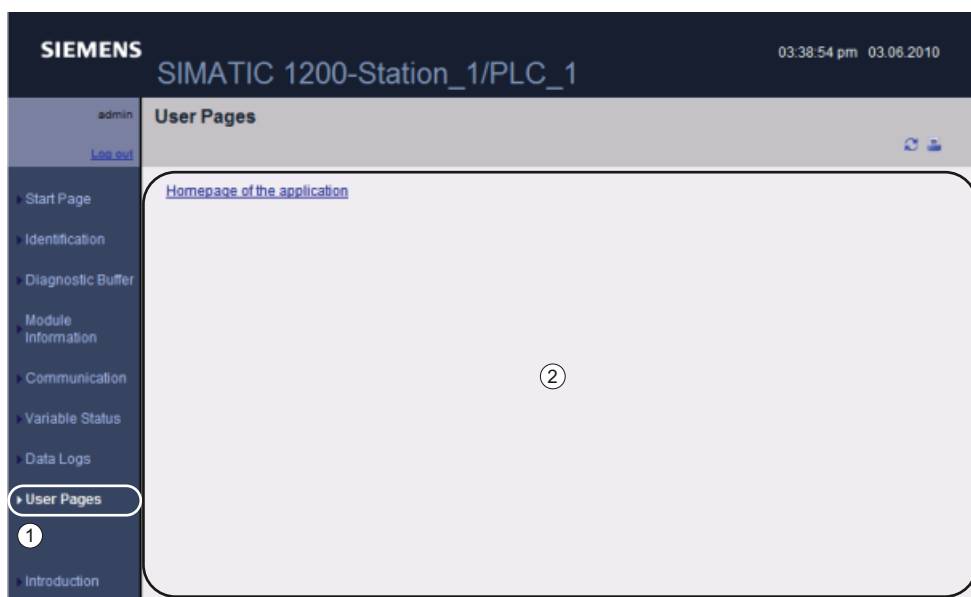
生成用户定义的 Web 页面块后，它们与所有其它程序块一样成为 STEP 7 程序的一部分。可以按照正常过程将这些程序块下载到 CPU。注意，只能在 CPU 处于 STOP 模式时下载用户定义的网页程序块。



### 11.3.6 通过 PC 访问用户定义 Web 页面

用户可以通过 标准 Web 页面 (页 543)访问用户定义 Web 页面。标准 Web 页面的左侧菜单中显示一个“用户页面”(User Pages) 链接，该页面中显示一些到其它页面的链接。单击“用户页面”(User Pages) 链接后，Web 浏览器将转到提供默认页面的入口的页面。在用户定义内容区域中，导航视特定页面的设计情况而定。

Web 服务器在标准 Web 页面显示画面的框架内显示用户定义页面内容。即，顶部的标题框和左侧的登录和导航框都仍在固定位置。



① 到用户定义 Web 页面的默认页面的链接

② 用户定义 Web 页面的内容区域

### 11.3.7 特定于用户定义 Web 页面的限制

标准 Web 页面的限制 (页 601)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。

### 11.3 用户定义的 Web 页面

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

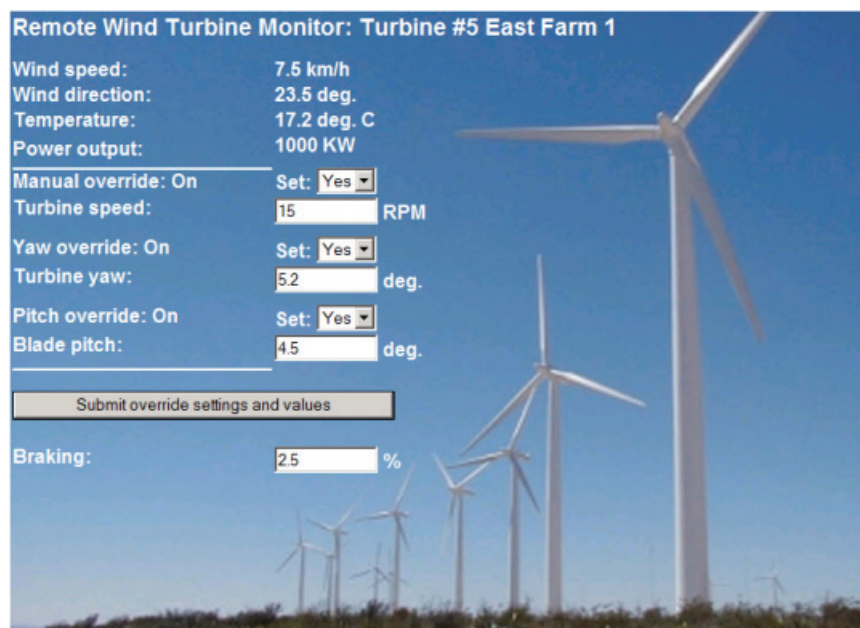
如果需要减少用户定义 Web 页面所需空间，则减少图片的使用（如果适用）。

---

## 11.3.8 用户定义 Web 页面示例

### 11.3.8.1 用于监控风力发电机的 Web 页面

可以考虑将用于远程监控风力发电机的 Web 页面作为用户定义 Web 页面示例：



---

#### 说明

此示例页面采用的是英语，当然，您在开发自己的 HTML 页面时可以使用任何语言。

---

## 说明

在此应用中，风电厂中的每台风力发电机都配备一个 S7-1200 用于控制涡轮机。在 STEP 7 程序中，各风力发电机都有一个数据块，其中包含该风力发电机及其位置的特定数据。

用户定义 Web 页面可用于通过 PC 远程访问涡轮机。用户可以连接到特定风力发电机的 CPU 的标准 Web 页面并访问用户定义“Remote Wind Turbine Monitor”Web 页面，以查看该涡轮机的数据。admin 用户还可以通过该 Web 页面将涡轮机置于手动模式，控制涡轮机速度、偏航和桨距变量。此外，无论涡轮机处于手动还是自动控制模式，admin 用户都可设置制动值。

STEP 7 程序会检查用于替代自动控制的布尔值，如果为真，则涡轮机速度、偏航和桨距将使用用户输入值。否则，程序将忽略这些值。

## 使用的文件

此用户定义 Web 页面示例包含三个文件：

- **Wind\_turbine.html:** 这是实现上面所示显示画面的 HTML 页面，该页面使用 AWP 命令访问控制器数据。
- **Wind\_turbine.css:** 这是包含该 HTML 页面格式样式的级联样式表。可以选择是否使用级联样式表，但使用它可以简化 HTML 页面的开发。
- **Wind\_turbine.jpg:** 这是 HTML 页面使用的背景图片。当然，可以选择是否在用户定义 Web 页面中使用图片，使用图片会额外占用 CPU 的存储空间。

这些文件没有随安装程序一起提供，但在这里将其作为示例来介绍。

## 实现

HTML 页面使用 AWP 命令从 PLC 读取值 (页 564) (用于显示字段)，并将值写入 PLC (页 565) (用于用户输入的数据)。此页面还使用 AWP 命令进行枚举类型定义 (页 571) 和引用 (页 572) 以处理 ON/OFF 设置。

页面的第一部分显示一个标题行，其中包括风力发电机编号和位置。

**Remote Wind Turbine Monitor: Turbine #5 East Farm 1**

页面的下一部分显示风力发电机所处的环境条件。这些字段由该涡轮机现场的 I/O 提供数据，提供的数据包括风速、风向和当前温度。

Wind speed:	7.5 km/h
Wind direction:	23.5 deg.
Temperature:	17.2 deg. C

接下来，页面显示从 S7-1200 读取的涡轮机的功率输出。

Power output:	1000 KW
---------------	---------

以下部分介绍了涡轮机的手动控制，手动控制会替代由 S7-1200 负责的标准的自动控制。各类型如下所述：

- 手动替代：启用对涡轮机设置的手动替代。只有手动替代设置为真时，STEP 7 用户程序才会对涡轮机速度、偏航或桨距使用手动设置。

Manual override: On	Set: Yes
Turbine speed:	15 RPM

- 偏航替代：启用对偏航设置的手动替代，对偏航使用手动设置。手动替代和偏航替代都为真时，STEP 7 用户程序才会应用偏航设置。

Yaw override: On	Set: Yes
Turbine yaw:	52 deg.

- 桨距替代：启用对叶片桨距的手动替代。手动替代和桨距替代都为真时，STEP 7 用户程序才会应用叶片桨距设置。

Pitch override: On	Set: Yes
Blade pitch:	45 deg.

HTML 页面包含一个用于将替代设置传送给控制器的提交按钮。

Submit override settings and values
-------------------------------------

制动用户输入字段提供手动制动设置百分数。无需启用手动替代，STEP 7 用户程序便会接受制动值。

Braking:	25 %
----------	------

此外，HTML 页面还使用 AWP 命令来写入特殊变量 (页 569) (其包含正在访问该页面的用户的用户 ID) 到 PLC 变量表中的一个变量。

### 11.3.8.2 读取和显示控制器数据

“Remote Wind Turbine Monitor”HTML 页面使用多个 AWP 命令从控制器读取数据 (页 564)，并将其显示在该页面中。例如，考虑一下用于显示该示例 Web 页面的下面部分所示功率输出的 HTML 代码：

Power output:	1000 KW
---------------	---------

## HTML 代码示例

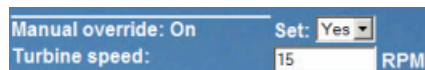
从“Remote Wind Turbine Monitor”HTML 页面中摘录的以下部分在表格行的左侧单元中显示文本“Power Output:”，读取功率输出变量并在该表格行的右侧单元中将其与单位文本 KW 显示在一起。

AWP 命令 :=“Data\_block\_1”.PowerOutput: 执行读取操作。请注意，数据块通过名称而不是通过数据块编号来引用（即，通过“Data\_block\_1”而不是“DB1”）。

```
<tr style="height:2%;">
<td>
<p>Power output:</p>
</td>
<td>
<p style="margin-bottom:5px;"> :=“Data_block_1”.PowerOutput:
KW</p>
</td>
</tr>
```

### 11.3.8.3 使用枚举类型

%“Remote Wind Turbine Monitor”HTML 页面在三种情况下使用枚举类型：HTML 页面显示代表布尔值的“ON”或“OFF”以及用户设置布尔值。“ON”枚举类型使值为 1，“OFF”枚举类型使值为 0。例如，考虑一下用于在 "Data\_block\_1".ManualOverrideEnable 值中使用枚举类型读取和写入 Manual Override Enable 设置的 HTML 代码：



## HTML 代码示例

从“Remote Wind Turbine Monitor”HTML 页面中摘录的以下部分显示如何声明值为“Off”和“On”（分别代表 0 和 1）的“OverrideStatus”枚举类型，以及随后设置对数据块（名为 Data\_block\_1）中 ManualOverrideEnable 布尔变量的 OverrideStatus 的枚举类型引用。请注意，对于每个 AWP\_Enum\_Ref 声明，如果 HTML 页面通过枚举类型写入变量，则必须有一个对应的数据块变量或 PLC 变量的 AWP\_In\_Variable 声明。

```
<!-- AWP_In_Variable Name='“Data_block_1”.ManualOverrideEnable' -->

<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->

<!-- AWP_Enum_Ref Name='“Data_block_1”.ManualOverrideEnable'
Enum="OverrideStatus" -->
```

其中，HTML 页面在表格单元中包含一个用于显示 `ManualOverrideEnable` 的当前状态的显示字段。该页面通常仅使用标准读取变量命令，但由于使用了先前声明和引用的枚举类型，所以该页面显示“Off”或“On”，而不是 0 或 1。

```
<td style="width:24%; border-top-style: Solid; border-top-width: 2px; border-top-color: #ffffff;">
<p>Manual override: :="Data_block_1".ManualOverrideEnable:</p>
</td>
```

该 HTML 页面包含一个下拉选择列表，供用户更改 `ManualOverrideEnable` 的值。该选择列表在选择列表中显示文本“`Yes`”和“`No`”。使用枚举类型时，“`Yes`”关联到枚举类型的值“`On`”，“`No`”关联到值“`Off`”。选项为空时保持 `ManualOverrideEnable` 的值不变。

```
<select name='"Data_block_1".ManualOverrideEnable'>
<option value='"Data_block_1".ManualOverrideEnable:'> </option>
<option value="On">Yes</option>
<option selected value="Off">No</option>
</select>
```

选择列表包含在 HTML 页面上的表单中。当用户单击提交按钮时，该页面将发布表单，如果用户已选择“`Yes`”，则会将值“1”写入 `Data_block_1` 中的布尔型 `ManualOverrideEnable`；如果用户选择的是“`No`”，则写入值“0”。

#### 11.3.8.4 将用户输入写入控制器

“Remote Wind Turbine Monitor”HTML 页面包含多个用于将数据写入控制器的 AWP 命令（页 565）。HTML 页面将 `AWP_In_Variables` 声明为布尔变量，以便管理员用户可以将风力发电机置于手动控制模式，并且可以启用涡轮机速动的手动替代、偏航替代和/或叶片桨距替代。该页面还使用 `AWP_In_Variables` 来让 `admin` 用户随后为涡轮机速度、偏航、桨距和制动百分数设置浮点值。该页面使用 HTTP 表单发布命令将 `AWP_In_Variables` 写入控制器。

例如，考虑一下用于手动设置制动值的 HTML 代码：



#### HTML 代码示例

从“Remote Wind Turbine Monitor”HTML 页面中摘录的以下部分首先声明 `"Data_block_1"` 的 `AWP_In_Variable`，该变量使 HTML 页面可以写入到数据块 `"Data_block_1"` 中的任何变量。该页面的表格行的左侧单元中显示文本“`Braking:`”。表格行的右侧单元是一个字段，可接受用户为“`Data_block_1`”的“`Braking`”变量输入的值。此用户输入值在 HTML 表单中，该表单使用 HTTP 方法“`POST`”将输入的文本数据传送到 CPU。然后，页面会从控制器读取实际制动值，并在数据输入字段中显示该值。

`admin` 用户随后可以通过此页面将制动值写入制动控制 CPU 中的数据块。

```
<!-- AWP_In_Variable Name='"Data_block_1"' -->
```

```

...
<tr style="vertical-align: top; height: 2%;">
<td style="width: 22%;"><p>Braking:</p></td>
<td>
<form method="POST">
<p><input name=' "Data_block_1".Braking' size="10" type="text">
%</p>
</form>
</td>
</tr>

```

**说明**

请注意，如果用户定义的页面具有用于字符串数据类型可写入数据块变量的数据输入字段，则在该字段中输入字符串值时，用户必须用单引号将字符串括起来。

**说明**

请注意，如果在 AWP\_In\_Variable 声明中声明整个数据块，例如 <!-- AWP\_In\_Variable Name="Data\_block\_1" -->，则通过用户定义 Web 页面会写入该数据块内的所有变量。如果要使数据块中的所有变量均为可写入变量，请使用此方法。而如果仅想通过用户定义 Web 页面写入特定数据块变量，则应在声明中对其进行具体声明，例如 <!-- AWP\_In\_Variable Name="Data\_block\_1".Braking' -->。

**11.3.8.5****写入特殊变量**

“Remote Wind Turbine Monitor”Web 页面将特殊变量 SERVER:current\_user\_id 写入到 CPU 中的 PLC 变量。在这种情况下，PLC 变量值包含正在访问“Remote Wind Turbine Monitor”Web 页面的用户的用户 ID。目前，admin 用户的用户 ID 为 1，所以 PLC 变量值设置为 1。

该特殊变量由 Web 页面写入 PLC，无需用户干预。

**HTML 代码示例**

```

<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID"-
-->

```

**11.3.8.6****引用：远程风力发电机监视 Web 页面的 HTML listing****Wind\_turbine.html**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"><!--
该测试程序将仿真一个 Web 页面，用于监视和控制一个风力发电机
STEP 7 中所需的 PLC 变量和数据块变量：

```

```
PLC 变量:
User_ID: Int
```

```
数据块:
Data_block_1
```

```
Data_Block_1 中的变量:
Location: String
TurbineNumber: Int
WindSpeed: Real
WindDirection: Real
Temperature: Real
PowerOutput: Real
ManualOverrideEnable: Bool
TurbineSpeed: Real
YawOverride: Bool
Yaw: Real
PitchOverride: Bool
Pitch: Real
Braking: Real
```

用户自定义的网页中将显示 PLC 数据的当前值，并提供一个选择列表，以便使用枚举类型分配设置 3 个布尔值。“提交”按钮将提交所选择的布尔值以及用于涡轮机速度、偏航和桨距的数据条目字段。无需点击“提交”按钮即可设置制动值。

使用该页面时，无需运行实际的 STEP 7 程序。理论上来说，如果设置了相关布尔值，则 STEP 7 程序将只计算涡轮机速度、偏航和桨距的值。对 STEP 7 的唯一要求是，通过该页面所生成数据块的 DB 编号调用 WWW 指令。

```
-->
<!-- AWP_In_Variable Name='Data_block_1' -->
<!-- AWP_In_Variable Name='Data_block_1'.ManualOverrideEnable' -->
-->
<!-- AWP_In_Variable Name='Data_block_1'.PitchOverride' -->
<!-- AWP_In_Variable Name='Data_block_1'.YawOverride' -->
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID"-->
-->
<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->
-->
<!-- AWP_Enum_Ref Name='Data_block_1'.ManualOverrideEnable'
Enum="OverrideStatus" -->
<!-- AWP_Enum_Ref Name='Data_block_1'.PitchOverride'
Enum="OverrideStatus" -->
<!-- AWP_Enum_Ref Name='Data_block_1'.YawOverride'
Enum="OverrideStatus" -->

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"><link rel="stylesheet" href="Wind_turbine.css">
<title>Remote Wind Turbine Monitor 页面</title>
```



```

</head>
<body>
<table style="background-image: url('./Wind_turbine.jpg'); width:
904px; height: 534px;" cellpadding="0" cellspacing="2"><tr
style="height: 2%;"><td colspan="2">
<h2>Remote Wind Turbine Monitor: Turbine
#:="Data_block_1".TurbineNumber: :="Data_block_1".Location:</h2>
</td>

<tr style="height: 2%;"><td style="width: 24%;"><p>Wind
speed:</p></td>
<td><p> :="Data_block_1".WindSpeed: km/h</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 24%;"><p>Wind direction:</p></td>
<td><p> :="Data_block_1".WindDirection: deg.</p></td>
</tr>

<tr style="height: 2%;"><td style="width:
24%;"><p>Temperature:</p></td>
<td><p> :="Data_block_1".Temperature: deg. C</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 24%;"><p>Power output:</p></td>
<td><p style="margin-bottom:5px;"> :="Data_block_1".PowerOutput:
KW</p>
</td>
</tr>

<form method="POST" action="">
<tr style="height: 2%;" >
<td style="width=24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manual override: :="Data_block_1".ManualOverrideEnable:</p>
</td>
<td class="Text">Set:

<select name="'Data_block_1".ManualOverrideEnable'>
<option value=' :="Data_block_1".ManualOverrideEnable:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;"><td style="width:
24%;"><p>Turbine speed:</p></td>
<td>

```

```

<p style="margin-bottom:5px;"><input
name=' "Data_block_1".TurbineSpeed' size="10"
value=' :="Data_block_1".TurbineSpeed:' type="text"> RPM</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;">
<p>Yaw override: :="Data_block_1".YawOverride: </p>
</td>
<td class="Text">Set:

<select name=' "Data_block_1".YawOverride'>
<option value=' :="Data_block_1".YawOverride:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;">
<p>Turbine yaw:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name=' "Data_block_1".Yaw'
size="10" value=' :="Data_block_1".Yaw:' type="text"> deg.</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;">
<p>Pitch override: :="Data_block_1".PitchOverride: </p>
</td>
<td class="Text">Set:

<select name=' "Data_block_1".PitchOverride'>
<option value=' :="Data_block_1".PitchOverride:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width=24%; border-bottom-style: Solid; border-bottom-
width: 2px; border-bottom-color: #ffffff;">
<p>Blade pitch:</p>
</td>
<td>

```

```

<p style="margin-bottom:5px;"><input name='"Data_block_1".Pitch'
size="10" value':"Data_block_1".Pitch:' type="text"> deg.</p>
</td>

</tr>
<tr style="height: 2%;">
<td colspan="2"><br>
<input type="submit" value="Submit override settings and values">
</td>
</tr>
</form>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 24%;"><p>Braking:</p></td>
<td>
<form method="POST" action="">
<p> <input name='"Data_block_1".Braking' size="10"
value':"Data_block_1".Braking:' type="text"> %</p>
</form>
</td>
</tr>
<tr><td></td></tr>

</table>
</body>
</html>

```

### Wind\_turbine.css

```

H2 {
font-family: Arial;
font-weight: bold;
font-size: 14.0pt;
color: #FFFFFF;
margin-top:0px;
margin-bottom:10px;
}
P {
font-family: Arial;
font-weight: bold;
color: #FFFFFF;
font-size: 12.0pt;
margin-top:0px;
margin-bottom:0px;
}
TD.Text {
font-family: Arial;
font-weight: bold;
color: #FFFFFF;
font-size: 12.0pt;
margin-top:0px;
margin-bottom:0px;
}

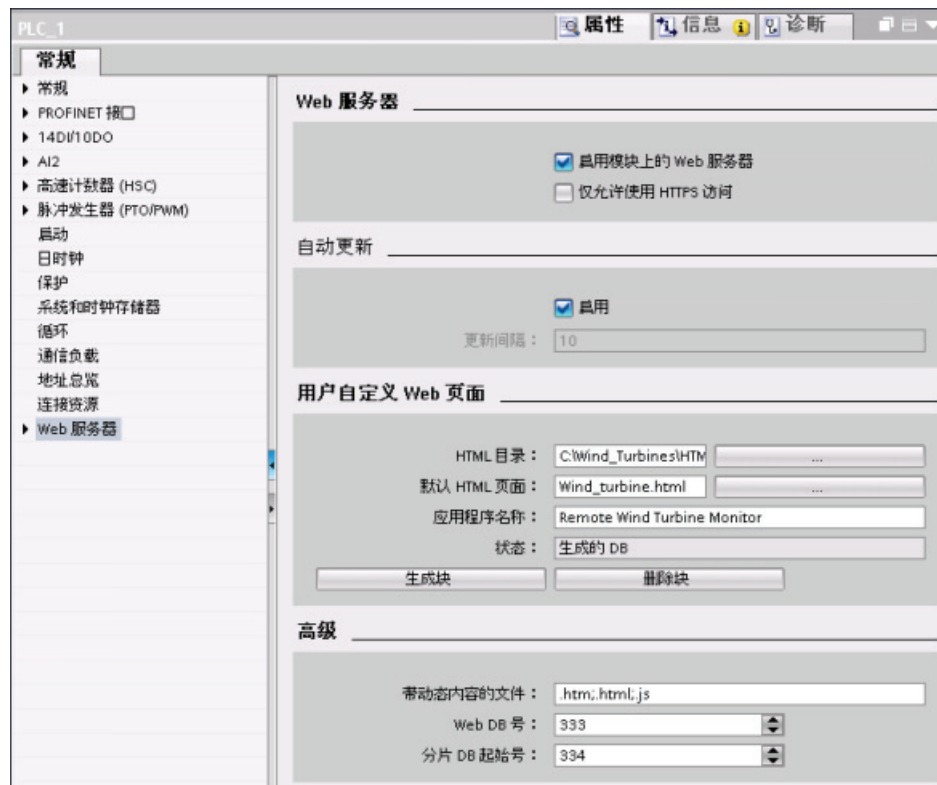
```

}

#### 11.3.8.7 STEP 7 中示例 Web 页面的组态

要包括“Remote Wind Turbine Monitor”HTML 页面并将其作为 S7-1200 的用户定义 Web 页面，需要在 STEP 7 中对 HTML 页面的相关数据进行组态并基于该 HTML 页面创建数据块。

访问控制风力发电机的 S7-1200 的“CPU 属性”(CPU Properties)，并在 Web 服务器的用户定义 Web 页面属性中输入组态信息：



## 组态字段

- “HTML 目录”(HTML directory): 此字段指定计算机上保存默认页面（主页或起始页面）的文件夹的完全限定路径名。使用“...”按钮可浏览至所需文件夹。
- “默认 HTML 页面”(Default HTML page): 此字段指定默认页面的文件名或 HTML 应用的主页。使用“...”按钮可选择所需文件。在此示例中，WindTurbine.html 是默认 HTML 页面。Remote Wind Turbine Monitor 示例仅包含一个页面，但在其它用户定义应用中，默认页面可以通过默认页面中的链接调用其它页面。在 HTML 代码中，默认页面必须引用与 HTML 源文件夹有关的其它页面。
- “应用名称”(Application name): 此可选字段包含一个名称，Web 浏览器在显示该页面时会将其显示在地址字段中。在此示例中，名称为“Remote Wind Turbine Monitor”，但可以是任何名称。

其它字段无需组态。

## 最终步骤

要按照所组态的方式使用 Remote Wind Turbine Monitor，需要生成块、以生成的控制 DB 的编号为输入参数来编写 WWW 指令 (页 579)、下载程序块并将 CPU 置于运行模式。

当操作员以后访问控制风力发电机的 S7-1200 的标准 Web 页面时，便可通过导航条上的“用户页面”(User Pages) 链接访问“Remote Wind Turbine Monitor”Web 页面。此时，可通过此页面监控风力发电机。

### 11.3.9 创建多语言用户定义 Web 页面

Web 服务器提供了一些用于提供以下语言形式的用户定义 Web 页面的方法：

- 德语 (de)
- 英语 (en)
- 西班牙语 (es)
- 法语 (fr)
- 意大利语 (it)
- 中文 (zh)

可通过在与各语言对应的文件夹结构 (页 594) 中创建 HTML 页面并在页面中设置名为“siemens\_automation\_language”的特定 cookie (页 595) 来实现。Web 服务器会响应此 cookie，并切换到相应语言文件夹中的默认页面。

#### 11.3.9.1 创建文件夹结构

要提供多语言用户定义 Web 页面，请在 HTML 目录下创建一个文件夹结构。文件夹名称应为特定的两个字母，必须按下面的方式命名：



还可在该目录下创建页面所需的任何其它文件夹，例如，图片文件夹或脚本文件夹。

可以包括语言文件夹的任何一种子集。不必包括所有六种语言。在语言文件夹中，创建并设置相应语言形式的 HTML 页面。

### 11.3.9.2 设置语言切换

Web 服务器使用名为“siemens\_automation\_language”的 cookie 来执行语言切换。此 cookie 在 HTML 页面中定义和设置，Web 服务器解释该 cookie 并相应地使用同名的语言文件夹中的相应语言来显示页面。HTML 页面必须包含 JavaScript，这样才可将此 cookie 设置为预定义语言标识符之一：“de”、“ed”、“es”、“fr”、“it”或“zh”。

例如，如果 HTML 页面将 cookie 设置为“de”，则 Web 服务器将切换到“de”文件夹并显示具有 STEP 7 组态 (页 597)过程中定义的默认 HTML 页面名称的页面。

#### 示例

下面的示例使用各语言文件夹中名为“langswitch.html”的默认 HTML 页面。此外，HTML 目录下有一个名为“script”的文件夹。script 文件夹包含一个名为“lang.js”的 JavaScript 文件。各 langswitch.html 页面使用此 JavaScript 来设置语言 cookie“siemens\_automation\_language”。

#### “en”文件夹中的“langswitch.html”的 HTML

HTML 页面的报头将语言设置为英语，将字符集设置为 UTF-8，并设置 JavaScript 文件 lang.js 的路径。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Language switching english page</title>
<script type="text/javascript" src="script/lang.js" ></script>
```

文件正文使用选择列表供用户在德语和英语之间进行选择。英语（“en”）是预选的语言。用户更改语言时，该页面将调用具有所选选项值的 DoLocalLanguageChange() JavaScript 函数。

```
<!-- Language Selection -->
<table>
  <tr>
    <td align="right" valign="top" nowrap>
      <!-- change language immediately on selection change -->
      <select name="Language"
        onchange="DoLocalLanguageChange(this)"
        size="1">
        <option value="de" >German</option>
```

```

                <option value="en" selected >English</option>
            </select>
        </td>
    </tr>
</table><!-- Language Selection End-->

```

### “de”文件夹中的“langswitch.html”的 HTML

除语言被设置为德语外，德语 langswitch.html 页面的报头与英语的相同。

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="de"><meta http-
equiv="Content-Type" content="text/html; charset=utf-8">
<title>Sprachumschaltung Deutsche Seite</title>
<script type="text/javascript" src="script/lang.js" ></script>
</head>

```

除所选语言的默认值为德语（“de”）外，德语页面中的 HTML 与英语页面中的完全相同。

```

<!-- Language Selection -->
<table>
    <tr>
        <td align="right" valign="top" nowrap>
            <!-- change language immediately on change of the
selection -->
            <select name="Language"
                onchange="DoLocalLanguageChange(this)"
                <size="1">
                <option value="de" selected >Deutsch</option>
                <option value="en" >Englisch</option>
            </select>
        </td>
    </tr>
</table><!-- Language Selection End-->

```

### “script”文件夹中的 JavaScript“lang.js”

函数“DoLocalLanguageChange()”在 lang.js 文件中。该函数调用“SetLangCookie()”函数，然后重新加载显示该 HTML 页面的窗口。

函数“SetLangCookie()”构造一个分配，将选择列表中的值分配给文档的“siemens\_automation\_language”cookie。该函数还会设置应用程序的路径，以便被切换页面（非请求页面）接收该 cookie 的值。

可以选择在注释部分中，该页面对 cookie 的到期值进行设置。

```

function DoLocalLanguageChange(oSelect) {
    SetLangCookie(oSelect.value);
    top.window.location.reload();
}

```



```
function SetLangCookie(value) {
    var strval = "siemens_automation_language=";
    // 这是 cookie, Web 服务器据此
    // 检测所需的语言。
    // Web 服务器需要此名称。
    strval = strval + value;
    strval = strval + "; path=/ ";
    // 设置应用程序路径, 否则
    // 该路径将被设置为请求页面,
    // 此页面将得不到 cookie。
    /* OPTIONAL
    如果此 cookie 的有效期应长于
    当前的浏览器会话, 则使用期限设置:
    var now = new Date();
    var endttime = new Date(now.getTime() + expiration);
    strval = strval + "; expires=" +
        endttime.toGMTString() + ";";
    */
    document.cookie = strval;
}
```

### 11.3.9.3 组态 STEP 7 以使用多语言页面结构

组态多语言用户定义 Web 页面的过程与 组态用户定义 Web 页面 (页 577) 的常规过程类似。为各语言创建文件夹后, 将 HTML 目录设置为包含各语言文件夹的文件夹。而不要将 HTML 目录设置为某一个语言文件夹。

选择默认 HTML 页面时, 导航到相应的语言文件夹并选择将作为起始页面的 HTML 页面。随后生成块并将这些块下载到 CPU 时, Web 服务器将显示所组态的语言文件夹中的起始页面。

例如, 如果此处所显示的文件夹结构为 C:\, 则 HTML 目录设置为 C:\html, 如果要选择英语作为初始页面显示语言, 则导航到作为默认 HTML 页面设置的 en\langswitch.html。



### 11.3.10 高级用户定义 Web 页面控制

为用户定义 Web 页面生成数据块时, STEP 7 会创建一个控制 DB, 该控制 DB 用于控制用户定义页面的显示和与该页面的交互。STEP 7 还会创建一组分别代表各页面的片段 DB。在正常情况下, 不需要知道控制 DB 的结构或处理控制 DB 的方法。

例如，如果要开关 Web 应用程序或处理各手动片段，可使用控制 DB 变量和 WWW 指令来实现。

### 控制 DB 的结构

控制 DB 是一种全面的数据结构，可以在编写 STEP 7 用户程序时访问。这里仅介绍了一部分控制数据块变量。

### Commandstate 结构

“Commandstate”是包含 Web 服务器的全局命令和全局状态的结构。

#### “Commandstate”结构中的全局命令

全局命令通常应用于 Web 服务器。可通过控制 DB 参数取消激活或重启 Web 服务器。

块变量	数据类型	说明
init	BOOL	评估控制 DB 并初始化 Web 应用程序
deactivate	BOOL	取消激活 Web 应用程序

#### Commandstate 结构中的全局状态

全局状态通常应用于 Web 服务器，其包含 Web 应用程序的状态信息。

块变量	数据类型	说明
initializing	BOOL	Web 应用程序正在读取控制 DB
error	BOOL	无法初始化 Web 应用程序
deactivating	BOOL	Web 应用程序正在终止
deactivated	BOOL	Web 应用程序已终止
initialized	BOOL	Web 应用程序已初始化

### 请求表

请求表是包含应用于各片段 DB 的命令和状态的结构数组。如果通过“手动”类型 AWP\_Start\_Fragment (页 573) 命令创建片段，则 STEP 7 用户程序必须通过控制 DB 控制这些页面。请求状态为只读，提供当前片段的相关信息。使用请求命令可控制当前片段。

块变量	数据类型	说明
requesttab	ARRAY [ 1 .. 4 ] OF STRUCT	用于各片段 DB 控制的结构数组。 Web 服务器一次最多可处理四个片段。 Web 服务器处理多个片段或通过多个浏览器会话处理片段时，特定片段的数组索引是任意的。

### requesttab 结构的结构成员

块变量	数据类型	说明
page_index	UINT	当前 Web 页面的编号
fragment_index	UINT	当前片段的编号 - 可以设置为其它片段
// 请求命令		
continue	BOOL	启用发送当前页面/片段，从下一片段开始继续发送
repeat	BOOL	启用重新发送当前页面/片段，从同一片段开始继续发送
abort	BOOL	关闭 http 连接并且不进行发送
finish	BOOL	发送此片段；页面已完成 - 不处理任何其它片段
// 请求状态		请求状态为只读
idle	BOOL	空闲，但处于激活状态
waiting	BOOL	片段等待被启用
sending	BOOL	片段正在发送
aborting	BOOL	用户已终止当前请求

### 操作

只要程序对控制 DB 进行了更改，就必须调用 WWW 指令并将已修改的控制 DB 编号作为该指令的参数。当 STEP 7 用户程序执行 WWW 指令(页 579)时，全局命令和请求命令生效。

STEP 7 用户程序可明确地设置 `fragment_index`，从而使 Web 服务器通过请求命令处理指定片段。否则，在 WWW 指令执行时，Web 服务器将处理当前页面的当前片段。

使用 `fragment_index` 的可能方法包括：

- 处理当前片段：保持 `fragment_index` 不变并设置 `continue` 命令。
- 跳过当前片段：将 `fragment_index` 设置为 0 并设置 `continue` 命令。
- 将当前片段替换为其它片段：将 `fragment_index` 设置为新片段 ID 并设置 `continue` 命令。

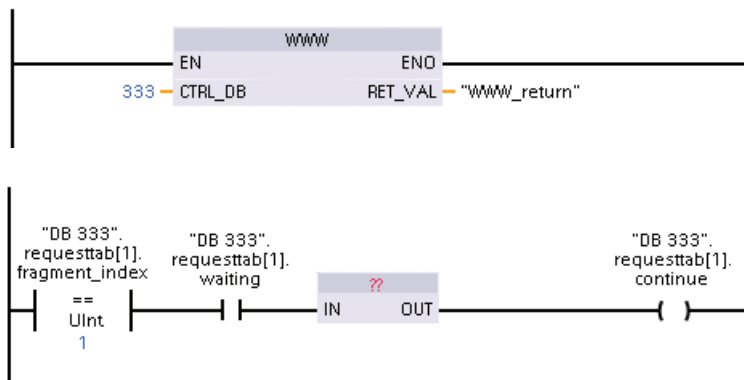
要检查全局状态或请求状态是否发生变化，STEP 7 用户程序必须调用 WWW 指令来评估这些状态的当前值。典型的方法是定期调用 WWW 指令，一直到出现特定状态。

### 说明

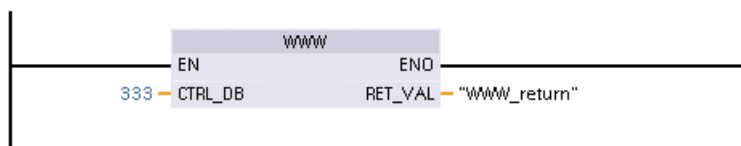
如果 STEP 7 用户程序设置了多条请求命令，则 WWW 指令仅会按优先级执行某一条命令，具体的顺序如下：`abort`、`finish`、`repeat`、`continue`。处理结束后，WWW 指令将清除所有请求命令。

### 示例

下面的示例显示了一个 STEP 7 用户程序，该程序检查在调用 WWW 指令后，ID 为 1 的片段是否进入等待状态。程序也可以等待其它应用程序特定条件的出现。然后执行该片段所需的任何操作，例如，设置数据块变量、执行计算或其它应用程序特定任务。然后，设置 `continue` 变量，以便 Web 服务器执行此片段。



当程序调用使用这一修改控制 DB 的 WWW 指令时，Web 浏览器将显示具有此片段的用户定义 Web 页面。



请注意，这是一个简化示例，要检查的片段可以是数组中的四个 `requesttab` 结构的任意一个。

## 11.4 限制

下列 IT 因素可能会影响 Web 服务器的使用：

- 通常，必须使用 CPU 的 IP 地址来访问标准 Web 页面或用户定义的 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 浏览器中禁用了 JavaScript 或 cookie，请将其启用。如果无法启用，某些功能将受限 (页 602)。可根据需要来选择是否在用户定义的 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 (页 547) 下载安全证书，并将证书导入 Web 浏览器的 Internet 选项中 (页 603)。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；在初始连接后其它连接将被释放。

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。

同时要知道页面性能会受到活动连接数目的影响。

### 11.4.1 禁用 JavaScript 时的受限功能

标准 Web 页面使用 HTML、JavaScript 和 cookie 来实现。如果站点限制使用 JavaScript 和 cookie，请将其启用，以使页面正常运行。如果无法为 Web 浏览器启用 JavaScript，将无法运行受 JavaScript 控制的功能。

#### 常规

这些页面不会动态更新。要查看刷新的数据，必须使用“刷新”(Refresh) 图标 (页 545) 手动刷新页面。

#### “诊断缓冲区”(Diagnostic Buffer) 页面

- 显示事件详细信息：使用 JavaScript 时，可在诊断缓冲区中选择一行，以在下面部分查看其详细信息。如果不使用 JavaScript，必须单击诊断缓冲区条目的事件字段超链接，才能在下面部分查看事件数据。
- 更改要查看的诊断缓冲区条目的范围：使用 JavaScript 时，可使用顶部的下拉列表来选择要查看的诊断缓冲区条目的范围，而页面会自动进行更新。如果不使用 JavaScript，则需使用顶部的下拉列表来选择要查看的诊断缓冲区条目的范围，但是，之后必须单击“转至”(Go) 链接，才能使用从下拉列表中选择的范围来更新诊断缓冲区页面。

请注意，仅当未启用 JavaScript 时，才能显示“转至”(Go) 和事件字段超链接。启用 JavaScript 时，不需要使用“转至”(Go) 和事件字段超链接，因此也不会显示它们。

---

#### 说明

Opera V11.0 浏览器不支持“转到”(Go) 按钮或者超链接诊断条目。在 Opera V11.0 中，如果禁用了 JavaScript，则无法查看事件详情和更改范围。

---

#### “模块信息”(Module Information) 页面

- 不能过滤数据。
- 不能对字段进行排序。

#### “变量状态”(Variable Status) 页面

- 输入每个变量之后，必须手动将焦点置于“New variable”行，才能输入新变量。
- 选择显示格式并不会自动将数据值显示更改为选定的格式。必须单击“Monitor value”按钮才能将显示内容刷新为新格式。

## Data Logs 页面

- 无法单击“Recent entries”下的文件名来打开或保存日志文件。但是，可以用“Download”图标实现这一功能。
- Data Logs 页面不会进行刷新。
- 用于增加或减少近期条目数的“+”按钮和“-”按钮不起作用。
- 在近期条目数中直接输入数值不会对条目数进行相应的设置。但在 Mozilla Firefox 中，如果尝试在该字段中输入数值，则画面将转到空白页。如要恢复“数据日志”(Data Logs) 显示，则必须在导航面板中重新选择“数据日志”(Data Logs)。而近期条目字段中的数值保持不变。

请注意，可以退出“数据日志”(Data Logs) 页面，然后重新进入该页面，从而获取最新的 25 个条目。

### 11.4.2 不允许使用 cookie 时受限的功能

Web 浏览器不允许使用 cookies 时，您无法以“admin”用户名登录。

### 11.4.3 导入 Siemens 安全证书

可以将 Siemens 安全证书导入 Internet 选项，这样，当您在 Web 浏览器中输入 `https://ww.xx.yy.zz`（其中“ww.xx.yy.zz”是 CPU 的 IP 地址）时，就不会出现安全验证提示。如果使用 `http:// URL`，而不是 `https:// URL`，则不需要下载并安装该证书。

## 下载证书

使用简介页面 (页 547) 中的“下载证书”(download certificate) 链接将 Siemens 安全证书下载到 PC 中。该过程会因所用 Web 浏览器而异：

[download certificate](#)

## 将证书导入 Internet Explorer

1. 单击“简介”(Introduction) 页面中的“下载证书”(download certificate) 链接。将弹出“File Download - Security Warning”对话框。
2. 单击“File Download - Security Warning”对话框中的“Open”打开文件。将出现“Certificate”对话框。

3. 在“Certificate”对话框中，单击“Install Certificate”按钮以启动 Certificate Import Wizard。
4. 按照“Certificate Import Wizard”对话框的提示导入证书，并使操作系统自动选择证书存储位置。

### 将证书导入 Mozilla Firefox

1. 单击 Intro 页面中的“download certificate”链接。将弹出“Opening MiniWebCA\_Cer.crt”对话框。
2. 单击“Opening MiniWebCA\_Cer.crt”对话框中的“Save file”。将出现“Downloads”对话框。
3. 双击“Downloads”对话框中的“MiniWebCA\_Cer.crt”。如果已多次尝试下载，则会显示多个副本。只需双击其中一个“MiniWebCA\_Cer.crt”条目即可。
4. 如果提示打开可执行文件，请单击“OK”。
5. 如果出现“Open File - Security Warning”对话框，请单击“Open”。将出现“Certificate”对话框。
6. 单击“Certificate”对话框中的“Install Certificate”按钮。
7. 按照“Certificate Import Wizard”对话框的提示导入证书，并使操作系统自动选择证书存储位置。
8. 如果出现“Security Warning”对话框，请单击“Yes”确认安装证书。

### 其它浏览器

导入和安装 Siemens 证书的过程与 Web 浏览器相同。

在 Web 浏览器的 Internet 选项中安装了 Siemens 安全证书“S7-Controller Family”之后，访问 [https:// ww.xx.yy.zz](https://ww.xx.yy.zz) 形式的 Web 服务器时不会再出现安全验证提示。

### 11.4.4 将 CSV 格式的数据日志导入非 USA/UK 版本的 Microsoft Excel 中

数据日志文件采用逗号分隔值 (CSV) 文件格式。如果系统正在运行 USA 或 UK 版本的 Excel，则可从“数据日志”(Data Logs) 页面直接用 Excel 打开这些文件。但是，在其它国家/地区，由于逗号经常出现在数字记数法中，因此并未广泛使用这种格式。



要打开已保存的数据日志文件，请对非 USA/UK 版本的 Excel 执行以下步骤：

1. 打开 Excel 并创建空白工作簿。
2. 从“数据 > 导入外部数据”(Data > Import External Data) 菜单中选择“导入数据”(Import Data) 命令。
3. 导航到要打开的数据日志文件并将其选中。将启动“文本导入向导”(Text Import Wizard)。
4. 在“文本导入向导”(Text Import Wizard) 中，将“原始数据类型”(Original data type) 的默认选项从“固定宽度”(Fixed width) 改为“带分隔符”(Delimited)。
5. 单击“下一步”(Next) 按钮。
6. 在“步骤 2”(Step 2) 对话框中，选中“逗号”(Comma) 复选框，以将分隔符类型从“制表符”(Tab) 改为“逗号”(Comma)。
7. 单击“下一步”(Next) 按钮。
8. 在“步骤 3”(Step 3) 对话框中，可根据需要将日期格式从“MDY (月/日/年)”(MDY (month/day/year)) 更改为其它格式。
9. 完成“文本导入向导”(Text Import Wizard) 的其余步骤，以导入文件。



## 通信处理器

### 12.1 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 909)
- CM 1241 RS422/485 (页 910)
- CB 1241 RS485 (页 906)

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。有关模块安装和拆卸的详细说明, 请参见 S7-1200 系统手册中的“安装”一章 (页 55)。

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
- 通过扩展指令和库功能进行组态和编程
- 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 895) 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。

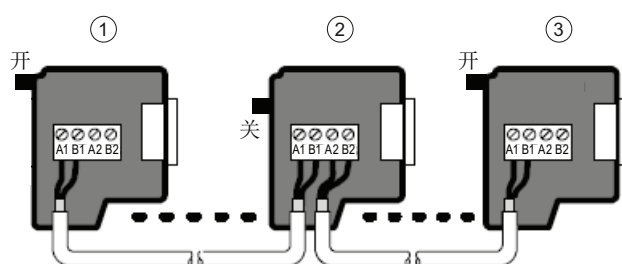
通信板具有发送 LED (TxD) 和接收 LED (RxD)。但没有诊断 LED。

## 12.2 偏置和端接 RS485 网络连接器

Siemens 提供了一个 RS485 网络连接器 (页 923), 用来方便地将多台设备连接到 RS485 网络。该连接器带有两组端子, 分别用于连接输入和输出网络电缆。连接器还包括用于选择性地偏置和端接网络的开关。

### 说明

但只能端接和偏置 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。  
 无电缆屏蔽: 所有位置的金属导线必须接触大约 12 mm (1/2 in)。



- ① 开关位置 = 开 (On): 端接且偏置
- ② 开关位置 = 关 (Off): 无端接或偏置
- ③ 开关位置 = 开 (On): 端接且偏置

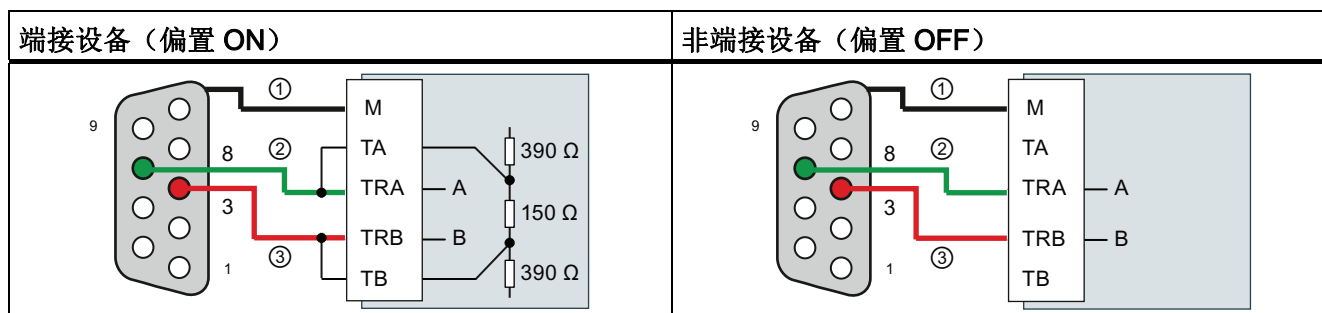
表格 12-1 RS485 连接器的端接和偏置

端接设备 (偏置 ON)	非端接设备 (偏置 OFF)

- ① 引脚编号
- ② 网络连接器
- ③ 电缆屏蔽

CB 1241 提供了用于端接和偏置网络的内部电阻。要终止或偏置连接, 应将 TRA 连接到 TA, 将 TRB 连接到 TB, 以便将内部电阻接到电路中。CB 1241 没有 9 针连接器。下表列出了与通信伙伴上的 9 针连接器之间的连接。

表格 12-2 CB 1241 的端接和偏置

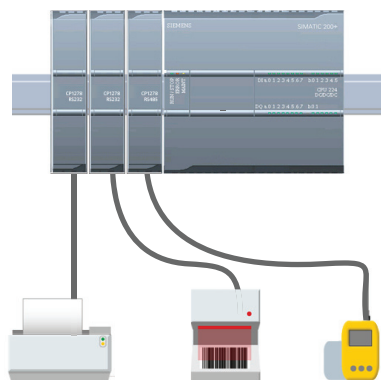


- ① 将 M 连接到电缆屏蔽
- ② A = TxD/RxD - (绿色线/针 8)
- ③ B = TxD/RxD + (红色线/针 3)

## 12.3 点对点 (PtP) 通信

CPU 支持下列有关基于字符的串行协议的点对点通信 (PtP)。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 (页 609)
- USS (页 653)
- Modbus (页 670)



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)

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 通信模块及 RS485 通信板提供了用于执行 PtP 通信的电气接口。

## 12.3 点对点 (PtP) 通信

## 12.3.1 点对点指令

## 12.3.1.1 点对点指令的公共参数

表格 12-3 PTP 指令的常见输入参数

参数	说明
REQ	<p>许多 PtP 指令使用 REQ 输入在由低电平向高电平切换时启动操作。REQ 输入在指令执行一次的时间内必须为高电平 (TRUE)，不过 REQ 输入可以在所需时间内一直保持为 TRUE。在 REQ 输入为 FALSE 时调用指令以便能复位 REQ 输入的历史状态之前，指令不会启动其它操作。只有这样，指令才能检测低电平到高电平的跳变以启动下一个操作。</p> <p>将 PtP 指令放入程序时，STEP 7 会提示用户指定背景数据块。对每个 PtP 指令调用使用一个唯一的背景数据块。这样可确保每个指令都能正确地处理诸如 REQ 等输入。</p>
PORT	<p>在通信设备组态过程中分配端口地址。组态后，可以从参数帮助下拉列表中选择默认端口的符号名称。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“常量”(Constants) 选项卡中分配。</p>
位时间精度	<p>有几个参数以位时间（通过组态的波特率确定）为单位指定的。以位时间为单位指定参数可以使参数与波特率无关。所有以位时间为单位的参数都可以被指定为最多 65535 个位。但 CM 或 CB 可测量的最长时间是 8 秒。</p>

PtP 指令的输出参数 DONE、NDR、ERROR 和 STATUS 可提供 PtP 操作的执行完成状态。

表格 12-4 DONE、NDR、ERROR 和 STATUS 输出参数

参数	数据类型	默认值	说明
DONE	Bool	FALSE	<p>设置为 TRUE 并持续执行一次所需的时间，以表明上一请求已经完成且没有出现错误；否则为 FALSE。</p>
NDR	Bool	FALSE	<p>设置为 TRUE 并持续执行一次所需的时间，以表明请求的动作已经完成且没有出现错误并已接收新的数据；否则为 FALSE。</p>

参数	数据类型	默认值	说明
ERROR	Bool	FALSE	设置为 TRUE 并持续执行一次所需的时间，以表明上一请求已经完成但出现了错误，相应的错误代码在 STATUS 中；否则为 FALSE。
STATUS	Word	0	<p>结果状态：</p> <ul style="list-style-type: none"> <li>• 如果设置了 DONE 或 NDR 位，则 STATUS 被设置为 0 或信息代码。</li> <li>• 如果设置了 ERROR 位，则 STATUS 被设置为一个错误代码。</li> <li>• 如果没有设置以上任何一位，则指令会返回说明功能当前状态的状态结果。</li> </ul> <p>STATUS 在该功能执行期间一直保持其值。</p>

#### 说明

DONE、NDR 和 ERROR 参数仅置位一个执行周期的时间。程序逻辑必须将临时输出状态值保存在数据锁存器中，以便能检测到后续程序扫描中的状态变化。

表格 12-5 公共条件代码

STATUS (W#16#...)	说明
0000	无错误
7000	功能不忙
7001	功能忙于处理第一个调用。
7002	功能忙于处理后续调用（第一个调用后的轮询）。
8x3A	参数 x 中的指针非法
8070	所有内部实例存储器都被占用，正在执行的并发指令过多
8080	端口号非法。
8081	超时、模块错误或其它内部错误
8082	由于正在后台进行参数化，参数化失败。
8083	<p>缓冲区溢出：</p> <p>CM 或 CB 返回一条接收到的消息，该消息的长度大于长度参数所允许的值。</p>

## 12.3 点对点 (PtP) 通信

STATUS (W#16#....)	说明
8090	内部错误：错误的消息长度、错误的子模块或非法消息 请联系客户支持。
8091	内部错误：参数化消息中的版本错误 请联系客户支持。
8092	内部错误：参数化消息中的记录长度错误 请联系客户支持。

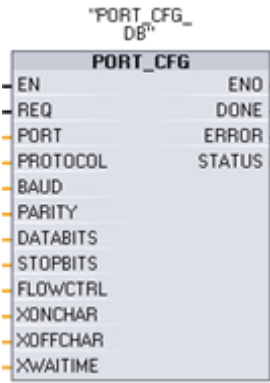
表格 12-6 常见的错误类别

类别说明	错误类别	说明
端口组态	80Ax	用于定义常见端口组态错误
传送组态	80Bx	用于定义常见传送组态错误
接收组态	80Cx	用于定义常见接收组态错误
传送运行时	80Dx	用于定义常见传送运行时错误
接收运行时	80Ex	用于定义常见接收运行时错误
信号处理	80Fx	用于定义与所有信号处理相关的常见错误



## 12.3.1.2 PORT\_CFG 指令

表格 12-7 PORT\_CFG (端口组态) 指令

LAD/FBD	SCL	说明
	<pre>"PORT_CFG_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   PROTOCOL:=_uint_in_,   BAUD:=_uint_in_,   PARITY:=_uint_in_,   DATABITS:=_uint_in_,   STOPBITS:=_uint_in_,   FLOWCTRL:=_uint_in_,   XONCHAR:=_char_in_,   XOFFCHAR:=_char_in_,   WAITTIME:=_uint_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	<p>PORT_CFG 可用于通过用户程序更改端口参数，如波特率等参数。</p> <p>可以在设备配置属性中设置端口的初始静态组态，或者仅使用默认值。可以在用户程序中执行 PORT_CFG 指令来更改组态。</p>

1 STEP 7 会在插入指令时自动创建 DB。

PORT\_CFG 组态更改不会永久存储在 CPU 中。CPU 从 RUN 模式切换到 STOP 模式和循环上电后将恢复设备配置中组态的参数。更多信息，请参见 组态通信端口 (页 630) 和管理流控制 (页 632)。

表格 12-8 参数的数据类型

参数和类型	数据类型	说明	
REQ	IN	Bool	在该输入的上升沿激活组态更改。(默认值: False)
PORT	IN	PORT	安装并组态 CM 或 CB 通信设备之后，端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。(默认值: 0)
PROTOCOL	IN	UInt	0 - 点对点通信协议 (默认值) 1..n - 用于在将来定义特定的协议
BAUD	IN	UInt	端口波特率 (默认值: 0) : 1 = 300 波特、2 = 600 波特、3 = 1200 波特、4 = 2400 波特、5 = 4800 波特、6 = 9600 波特、7 = 19200 波特、8 = 38400 波特、9 = 57600 波特、10 = 76800 波特、11 = 115200 波特

## 12.3 点对点 (PtP) 通信


参数和类型		数据类型	说明
PARITY	IN	UInt	端口奇偶校验（默认值：0）： 1 = 无奇偶校验、2 = 偶校验、3 = 奇校验、4 = 传号校验、5 = 空号校验
DATABITS	IN	UInt	每个字符的位数（默认值：）： 1 = 8 个数据位、2 = 7 个数据位
STOPBITS	IN	UInt	停止位（默认值：0）： 1 = 1 个停止位、2 = 2 个停止位
FLOWCTRL	IN	UInt	流控制（默认值：0）： 1 = 无流控制、2 = XON/XOFF、3 = 硬件 RTS 始终激活、4 = 硬件 RTS 切换
XONCHAR	IN	Char	指定用作 XON 字符的字符。这通常是 DC1 字符 (11H)。只有启用流控制时，才会评估该参数。（默认值：0）
XOFFCHAR	IN	Char	指定用作 XOFF 字符的字符。这通常是 DC3 字符 (13H)。只有启用流控制时，才会评估该参数。（默认值：0）
XWAITIME	IN	UInt	指定在接收 XOFF 字符后等待 XON 字符的时间，或者指定在启用 RTS 后等待 CTS 信号的时间（0 到 65535 ms）。只有启用流控制时，才会评估该参数。（默认值：2000）
DONE	OUT	Bool	上一请求已完成且没有出错后，保持为 TRUE 一个执行周期时间
ERROR	OUT	Bool	上一请求已完成但出现错误后，保持为 TRUE 一个执行周期时间
STATUS	OUT	Word	执行条件代码（默认值：0）

表格 12-9 条件代码

STATUS (W#16#....)	说明
80A0	特定协议不存在。
80A1	特定波特率不存在。
80A2	特定奇偶校验选项不存在。
80A3	特定数据位数不存在。
80A4	特定停止位数不存在。
80A5	特定流控制类型不存在。
80A6	等待时间为 0 且流控制启用
80A7	XON 和 XOFF 是非法值（例如，同一个值）

## 12.3.1.3 SEND\_CFG 指令

表格 12- 10 SEND\_CFG (发送组态) 指令

LAD/FBD	SCL	说明
	<pre>"SEND_CFG_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   RTSONDLY:=_uint_in_,   RTSOFFDLY:=_uint_in_,   BREAK:=_uint_in_,   IDLELINE:=_uint_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	SEND_CFG 可用于动态组态 PtP 通信端口的串行传输参数。执行 SEND_CFG 时，将放弃 CM 或 CB 内所有排队的消息。

1 STEP 7 会在插入指令时自动创建 DB。

可以在设备配置属性中设置端口的初始静态组态，或者仅使用默认值。可以在用户程序中执行 SEND\_CFG 指令来更改组态。

SEND\_CFG 组态更改不会永久存储在 CPU 中。CPU 从 RUN 模式切换到 STOP 模式和循环上电后将恢复设备配置中组态的参数。请参见 组态传送 (发送) 参数 (页 634)。

表格 12- 11 参数的数据类型

参数和类型	数据类型	说明	
REQ	IN	Bool	在该输入的上升沿激活组态更改。(默认值: False)
PORT	IN	PORT	安装并组态 CM 或 CB 通信设备之后，端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。(默认值: 0)
RTSONDLY	IN	UInt	启用 RTS 后执行任何 Tx 数据传输前要等待的毫秒数。只有启用硬件流控制时，该参数才有效。有效范围是 0 - 65535 ms。值 0 表示禁用此功能。(默认值: 0)
RTSOFFDLY	IN	UInt	执行 Tx 数据传输后禁用 RTS 前要等待的毫秒数：只有启用硬件流控制时，该参数才有效。有效范围是 0 - 65535 ms。值 0 表示禁用此功能。(默认值: 0)
BREAK	IN	UInt	该参数指定在各消息开始时将发送指定时间的中断。最大值是 65535 个位时间，最多为 8 秒。值 0 表示禁用该功能。(默认值: 12)

12.3 点对点 (PtP) 通信

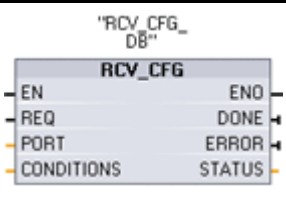
参数和类型		数据类型	说明
IDLELINE	IN	UInt	该参数指定在各消息开始前线路将保持空闲指定的位时间。最大值是 65535 个位时间，最多为 8 秒。值 0 表示禁用该功能。 (默认值: 12)
DONE	OUT	Bool	上一请求已完成且没有出错后，保持为 TRUE 一个执行周期时间
ERROR	OUT	Bool	上一请求已完成但出现错误后，保持为 TRUE 一个执行周期时间
STATUS	OUT	Word	执行条件代码 (默认值: 0)

表格 12- 12 条件代码

STATUS (W#16#....)	说明
80B0	不允许传送中断组态。请联系客户支持。
80B1	中断时间大于允许的最大值。
80B2	空闲时间大于允许的最大值。

12.3.1.4 RCV\_CFG 指令

表格 12- 13 RCV\_CFG (接收组态) 指令

LAD/FBD	SCL	说明
	<pre>"RCV_CFG_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   CONDITIONS:=_struct_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	<p>RCV_CFG 可用于动态组态 PtP 通信端口的串行接收方参数。该指令可组态表示接收消息开始和结束的条件。执行 RCV_CFG 时，将放弃 CM 或 CB 内所有排队的消息。</p>

1 STEP 7 会在插入指令时自动创建 DB。

可以在设备配置属性中设置通信端口的初始静态组态，或者索性使用默认值。可以在用户程序中执行 RCV\_CFG 指令来更改组态。

RCV\_CFG 组态更改不会永久存储在 CPU 中。CPU 从 RUN 模式切换到 STOP 模式和循环上电后将恢复设备配置中组态的参数。有关详细信息，请参见 组态接收参数 (页 634)。

表格 12-14 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	在该输入的上升沿激活组态更改。（默认值：False）
PORT	IN	PORT	安装并组态 CM 或 CB 通信设备之后，端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。（默认值：0）
CONDITIONS	IN	CONDITIONS	如下文所述，条件数据结构指定消息开始和结束条件。
DONE	OUT	Bool	上一请求已完成且没有出错后，保持为 TRUE 一个扫描周期时间
ERROR	OUT	Bool	上一请求已完成但出现错误后，保持为 TRUE 一个扫描周期时间
STATUS	OUT	Word	执行条件代码（默认值：0）

### RCV\_PTP 指令的开始条件

RCV\_PTP 指令使用 RCV\_CFG 指令指定的组态来确定点对点通信消息的开始和结束。消息开始由开始条件确定。消息开始可以由一个开始条件或开始条件的组合来确定。如果指定多个开始条件，则只有满足所有条件后才能使消息开始。

有关消息开始条件的说明，请参见主题“组态接收参数 (页 635)”。

## 参数 CONDITIONS 数据类型结构的第 1 部分（开始条件）

表格 12- 15 开始条件的 CONDITIONS 结构

参数和类型	数据类型	说明
STARTCOND	IN UInt	指定开始条件（默认值：1） <ul style="list-style-type: none"> <li>• 01H - 开始字符</li> <li>• 02H - 任意字符</li> <li>• 04H - 线路中断</li> <li>• 08H - 线路空闲</li> <li>• 10H - 序列 1</li> <li>• 20H - 序列 2</li> <li>• 40H - 序列 3</li> <li>• 80H - 序列 4</li> </ul>
IDLETIME	IN UInt	线路空闲超时所需的位时间数。（默认值：40）。仅与线路空闲条件一起使用。0 到 65535
STARTCHAR	IN Byte	用于开始字符条件的开始字符。（默认值：B#16#2）
STRSEQ1CTL	IN Byte	针对每个字符执行的序列 1 忽略/比较控制：（默认值：B#16#0） 它们是为开始序列中各字符启用的位。 <ul style="list-style-type: none"> <li>• 01H - 字符 1</li> <li>• 02H - 字符 2</li> <li>• 04H - 字符 3</li> <li>• 08H - 字符 4</li> <li>• 10H - 字符 5</li> </ul> 禁用与某个字符关联的位意味着该序列位置中的任意字符均符合条件。
STRSEQ1	IN Char[5]	序列 1 开始字符（5 个字符）。默认值：0
STRSEQ2CTL	IN Byte	针对每个字符执行的序列 2 忽略/比较控制。（默认值：B#16#0）
STRSEQ2	IN Char[5]	序列 2 开始字符（5 个字符）。默认值：0
STRSEQ3CTL	IN Byte	针对每个字符执行的序列 3 忽略/比较控制。默认值：B#16#0
STRSEQ3	IN Char[5]	序列 3 开始字符（5 个字符）。默认值：0

参数和类型		数据类型	说明
STRSEQ4CTL	IN	Byte	针对每个字符执行的序列 4 忽略/比较控制。默认值： B#16#0
STRSEQ4	IN	Char[5]	序列 4 开始字符（5 个字符），默认值：0

### 示例

请注意以下所接收的十六进制编码消息：“68 10 aa 68 bb 10 aa 16”以及下表中列出的已组态开始序列。在成功接收到第一个 68H 字符时，开始评估开始序列。在成功接收到第四个字符（第二个 68H）时，开始条件 1 得到满足。只要满足了开始条件，就会开始评估结束条件。

开始序列处理会因各种奇偶校验、成帧或字符间时间错误而终止。由于不再满足开始条件，因而这些错误将导致不会有接收消息。

表格 12- 16 开始条件

开始条件	第一个字符	第一个字符 +1	第一个字符 +2	第一个字符 +3	第一个字符 +4
1	68H	xx	xx	68H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

### RCV\_PTP 指令的结束条件

消息结束由指定的结束条件确定。消息结束由第一次出现的一个或多个已组态结束条件来确定。主题“组态接收参数 (页 635)”中“消息结束条件”部分介绍了可以在 RCV\_CFG 指令中组态的结束条件。

可以在设备配置的通信接口的属性中组态结束条件，或者通过 RCV\_CFG 指令组态结束条件。只要 CPU 从 STOP 模式切换到 RUN 模式，接收参数（开始条件和结束条件）就将恢复为设备配置设置。如果 STEP 7 用户程序执行 RCV\_CFG，则这些设置将更改为 RCV\_CFG 的条件。

## 参数 CONDITIONS 数据类型结构的第 2 部分（结束条件）

表格 12- 17 结束条件的 CONDITIONS 结构

参数	参数类型	数据类型	说明
ENDCOND	IN	UInt 0	该参数指定消息结束条件： <ul style="list-style-type: none"> <li>• 01H - 响应时间</li> <li>• 02H - 消息时间</li> <li>• 04H - 字符间隙</li> <li>• 08H - 最大长度</li> <li>• 10H - N + LEN + M</li> <li>• 20H - 序列</li> </ul>
MAXLEN	IN	UInt 1	最大消息长度：仅当选择最大长度结束条件时使用。1 到 1024 个字节
N	IN	UInt 0	长度域在消息中的字节位置。仅与 N + LEN + M 结束条件一起使用。1 到 1022 个字节
LENGTHSIZE	IN	UInt 0	字节域的大小（1、2 或 4 个字节）。仅与 N + LEN + M 结束条件一起使用。
LENGTHM	IN	UInt 0	指定跟在长度域后、不包含在长度域值内的字符数。该参数仅与 N + LEN + M 结束条件一起使用。0 到 255 个字节
RCVTIME	IN	UInt 200	指定接收第一个字符所需的等待时间。如果在指定时间内没有成功接收到字符，接收操作将被终止且包含错误。该参数仅与响应时间条件一起使用。（0 到 65535 个位时间，最多 8 秒）  此参数不是消息结束条件，因为在接收到第一个响应字符时评估即终止。由于在预期有响应时却接收不到响应，因此仅就其能够终止接收方操作而言，它又是一个结束条件。必须选择一个单独的结束条件。
MSGTIME	IN	UInt 200	指定在接收到第一个字符后完成接收整条消息所需的等待时间。只有选择了消息超时条件时，才会使用该参数。（0 到 65535 毫秒）



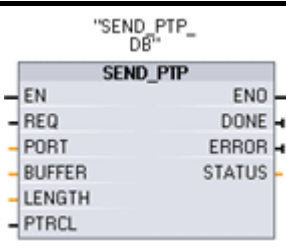
参数	参数类型	数据类型	说明
CHARGAP	IN	UInt 12	指定字符间的位时间数。如果字符间的位时间数超出指定值，则结束条件得到满足。该参数仅与字符间隙条件一起使用。（0 到 65535 个位时间，最多 8 秒）
ENDSEQ1CTL	IN	Byte B#16#0	针对每个字符执行的序列 1 忽略/比较控制：它们是为结束序列中各字符启用的位。字符 1 是位 0，字符 2 是位 1，依此类推，字符 5 是位 4。禁用与某个字符关联的位意味着该序列位置中的任意字符均符合条件。
ENDSEQ1	IN	Char[5] 0	序列 1 开始字符（5 个字符）

表格 12- 18 条件代码

STATUS (W#16#....)	说明
80C0	所选开始条件非法
80C1	所选结束条件非法；未选择结束条件
80C2	启用了接收中断，但不允许此操作。
80C3	启用了最大长度结束条件，最大长度是 0 或大于 1024。
80C4	启用了计算长度，但 $N \geq 1023$ 。
80C5	启用了计算长度，但长度不是 1、2 或 4。
80C6	启用了计算长度，但 M 值大于 255。
80C7	启用了计算长度，但计算长度大于 1024。
80C8	启用了响应超时，但响应超时为零。
80C9	启用了字符间隙超时，但该字符间隙超时为零。
80CA	启用了线路空闲超时，但该线路空闲超时为零。
80CB	启用了结束序列，但所有字符均“不相关”。
80CC	启用了开始序列（4 个中的任何一个），但所有字符均“不相关”。

## 12.3.1.5 SEND\_PTP 指令

表格 12- 19 SEND\_PTP (发送点对点数据) 指令

LAD/FBD	SCL	说明
	<pre>"SEND_PTP_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   BUFFER:=_variant_in_,   LENGTH:=_uint_in_,   PTRCL:=_bool_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	<p>SEND_PTP 用于启动数据传输，并将分配的缓冲区传送到通信接口。在 CM 或 CB 块以指定波特率发送数据的同时，CPU 程序会继续执行。仅一个发送操作可以在某一给定时间处于未决状态。如果在 CM 或 CB 已经开始传送消息时执行第二个 SEND_PTP，CM 或 CB 将返回错误。</p>

1 STEP 7 会在插入指令时自动创建 DB。

表格 12- 20 参数的数据类型

参数和类型	数据类型	说明	
REQ	IN	Bool	在该传送使能输入的上升沿激活所请求的传送。这会启动将缓冲区数据传送到点对点通信接口。（默认值：False）
PORT	IN	PORT	安装并组态 CM 或 CB 通信设备之后，端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。（默认值：0）
BUFFER	IN	Variant	该参数指向传送缓冲区的起始位置。（默认值：0） 注：不支持布尔数据或布尔数组。
LENGTH	IN	UInt	传输的帧长度（字节）（默认值：0） 传输复杂结构时，始终使用长度 0。
PTRCL	IN	Bool	该参数选择普通点对点协议或 Siemens 提供的特定协议所在的缓冲区，这些协议在所连接的 CM 或 CB 中实施。（默认值：False） FALSE = 用户程序控制的点对点操作。（仅限有效选项）
DONE	OUT	Bool	上一请求已完成且没有出错后，保持为 TRUE 一个扫描周期时间
ERROR	OUT	Bool	上一请求已完成但出现错误后，保持为 TRUE 一个扫描周期时间
STATUS	OUT	Word	执行条件代码（默认值：0）

传送操作进行期间，DONE 和 ERROR 输出均为 FALSE。传送操作完成后，DONE 或 ERROR 输出将被设置为 TRUE 以显示传送操作的状态。当 DONE 或 ERROR 为 TRUE 时，STATUS 输出有效。

如果通信接口接受传送数据，则该指令将返回状态值 16#7001。如果 CM 或 CB 仍在忙于传送，则后续的 SEND\_PTP 执行将返回 16#7002。传送操作完成后，CM 或 CB 将返回传送操作状态 16#0000（如果未出错）。后续执行 REQ 为低电平的 SEND\_PTP 时，将返回状态 16#7000（不忙）。

下图显示了输出值与 REQ 的关系。假设定期调用该指令以检查传送过程的状态。在下图中，假设每次扫描都调用该指令（用 STATUS 值表示）。

REQ							
DONE							
ERROR							
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H

下图显示通过 REQ 线路脉冲（持续一个扫描周期）启动传送操作时，DONE 和 STATUS 参数是如何仅在一个扫描周期内有效。

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

下图显示了出错时 DONE、ERROR 和 STATUS 参数之间的关系。

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	80D1H	7000H	7000H

只有 SEND\_PTP 再次使用相同的背景 DB 执行后，DONE、ERROR 和 STATUS 值才有效。

表格 12-21 条件代码

STATUS (W#16#...)	说明
80D0	传送方激活期间发出新请求
80D1	由于在等待时间内没有 CTS 信号，传送中止

STATUS (W#16#....)	说明
80D2	由于没有来自 DCE 设备的 DSR，传送中止
80D3	由于队列溢出（传送 1024 个字节以上），传送中止
80D5	反向偏置信号（断线检测）
833A	BUFFER 参数的 DB 不存在。

### SEND\_PTP 的 LENGTH 和 BUFFER 参数的交互作用

SEND\_PTP 指令可以传送的最小数据单位是一个字节。BUFFER 参数决定要传送的数据的大小。BUFFER 参数不接受 Bool 数据类型或 Bool 数组。

可以将 LENGTH 参数始终设置为 0，从而确保 SEND\_PTP 发送 BUFFER 参数表示的整个数据结构。如果仅要传送 BUFFER 参数中的部分数据结构，则可对 LENGTH 进行以下设置：

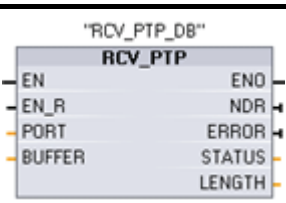
表格 12- 22 LENGTH 和 BUFFER 参数

LENGTH	BUFFER	说明
= 0	未使用	发送 BUFFER 参数中定义的全部数据。当 LENGTH = 0 时，用户无须指定传送字节数。
> 0	基本数据类型	LENGTH 值必须包含此数据类型的字节计数。例如，对于 Word 值，LENGTH 值必须为二。对于 Dword 或 Real，LENGTH 值必须为四。否则，不会传送任何数据并返回错误 8088H。
	结构	LENGTH 值可以包含小于结构完整字节长度的字节数，在这种情况下，将仅发送 BUFFER 中的结构的前 LENGTH 个字节。由于结构的内部字节组织不总是确定不变的，所以可能得到无法预料的结果。在这种情况下，使用值为 0 的 LENGTH 来发送整个结构。
	数组	LENGTH 值必须包含小于数组完整字节长度的字节数，而且还必须为数据元素字节计数的倍数。例如，对于 Word 数组，LENGTH 参数值必须为二的倍数；对于 Real 数组，必须为四的倍数。指定 LENGTH 后，将传送 LENGTH 字节中包含的相应数量的数组元素。例如，如果 BUFFER 包含由 15 个 Dword 构成的数组（总共 60 个字节），LENGTH 指定为 20，则将传送数组中的前五个 Dword。  LENGTH 值必须为数据元素字节数的倍数。否则，STATUS = 8088H，ERROR = 1，且不进行任何传送。

LENGTH	BUFFER	说明
	String	参数 LENGTH 包含要传送的字符数。只传送 String 中相应数量的字符。而不会传送 String 的最大长度和实际长度的字节数。

### 12.3.1.6 RCV\_PTP 指令

表格 12-23 RCV\_PTP（接收点对点）指令

LAD/FBD	SCL	说明
	<pre>"RCV_PTP_DB" (   EN_R:=_bool_in_,   PORT:=_uint_in_,   BUFFER:=_variant_in_,   NDR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   LENGTH=&gt;_uint_out_);</pre>	RCV_PTP 用于检查 CM 或 CB 中已接收的消息。如果有消息，则会将其从 CM 或 CB 传送到 CPU。如果发生错误，则会返回相应的 STATUS 值。

- STEP 7 会在插入指令时自动创建 DB。

表格 12-24 参数的数据类型

参数和类型	数据类型	说明	
EN_R	IN	Bool	该输入为 TRUE 并且有消息时，会将消息从 CM 或 CB 传送到 BUFFER。EN_R 为 FALSE 时，将检查 CM 或 CB 是否收到消息并更新 NDR、ERROR 和 STATUS 输出，但不会将消息传送到 BUFFER。（默认值：0）
PORT	IN	PORT	安装并组态 CM 或 CB 通信设备之后，端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。（默认值：0）
BUFFER	IN	VARIANT	该参数指向接收缓冲区的起始位置。该缓冲区应该足够大，可以接收最大长度消息。 不支持布尔数据或布尔数组。（默认值：0）
NDR	OUT	Bool	新数据就绪且操作无错完成后，保持为 TRUE 一个执行周期时间。
ERROR	OUT	Bool	操作已完成但出现错误后，保持为 TRUE 一个执行周期时间。
STATUS	OUT	Word	执行条件代码（默认值：0）
LENGTH	OUT	UInt	返回消息的长度（字节）（默认值：0）

## 12.3 点对点 (PtP) 通信

NDR 或 ERROR 为 TRUE 时，STATUS 值有效。STATUS 值提供 CM 或 CB 中的接收操作终止的原因。它通常是正值，表示接收操作成功且接收过程正常终止。如果 STATUS 值为负数（十六进制值的最高有效位置位），则表示接收操作因错误条件终止，例如，奇偶校验、组帧或超限错误。

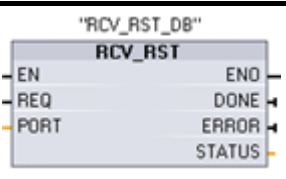
每个 PtP 通信接口最多可缓冲 1024 字节。这可以是一个大消息或几个较小的消息。如果 CM 或 CB 中存在多个消息，则 RCV\_PTP 指令将返回最早的可用消息。随后执行 RCV\_PTP 指令将返回下一个最早的可用消息。

表格 12- 25 条件代码

STATUS (W#16#...)	说明
0000	没有提供缓冲区
80E0	因接收缓冲区已满，消息被终止
80E1	因出现奇偶校验错误，消息被终止
80E2	因组帧错误，消息被终止
80E3	因出现超限错误，消息被终止
80E4	因计算长度超出缓冲区大小，消息被终止
80E5	反向偏置信号（断线检测）
0094	因接收到最大字符长度，消息被终止
0095	因消息超时，消息被终止
0096	消息因字符间超时而终止
0097	消息因响应超时而终止
0098	因已满足“N+LEN+M”长度条件，消息被终止
0099	因已满足结束序列，消息被终止
833A	BUFFER 参数的 DB 不存在。

## 12.3.1.7 RCV\_RST 指令

表格 12-26 RCV\_RST (接收方复位) 指令

LAD/FBD	SCL	说明
	<pre>"RCV_RST_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	RCV_RST 可清空 CM 或 CB 中的接收缓冲区。

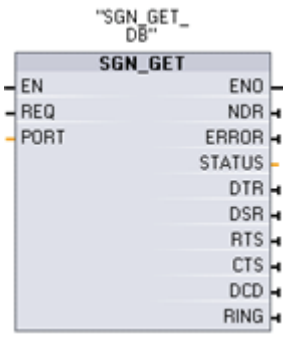
1 STEP 7 会在插入指令时自动创建 DB。

表格 12-27 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	该使能输入出现上升沿时将激活接收方复位 (默认值: False)
PORT	IN	PORT	安装并组态 CM 或 CB 通信设备之后, 端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。(默认值: 0)
DONE	OUT	Bool	在一个扫描周期内为 TRUE 时, 表示上一个请求已完成且没有错误。
ERROR	OUT	Bool	为 TRUE 时, 表示上一个请求已完成但有错误。此外, 该输出为 TRUE 时, STATUS 输出还会包含相关错误代码。
STATUS	OUT	Word	错误代码 (默认值: 0)

## 12.3.1.8 SGN\_GET 指令

表格 12- 28 SGN\_GET (获取 RS232 信号) 指令

LAD/FBD	SCL	说明
	<pre>"SGN_GET_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   NDR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   DTR=&gt;_bool_out_,   DSR=&gt;_bool_out_,   RTS=&gt;_bool_out_,   CTS=&gt;_bool_out_,   DCD=&gt;_bool_out_,   RING=&gt;_bool_out_);</pre>	<p>SGN_GET 用于读取 RS232 通信信号的当前状态。</p> <p>该功能仅对 RS232 CM 有效。</p>

1 STEP 7 会在插入指令时自动创建 DB。

表格 12- 29 参数的数据类型

参数和类型	数据类型	说明
REQ IN	Bool	在该输入的上升沿获取 RS232 信号状态值 (默认值: False)
PORT IN	PORT	安装并组态 CM 或 CB 通信设备之后, 端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。
NDR OUT	Bool	新数据就绪且操作无错误地完成时, 在一个扫描周期内为 TRUE
ERROR OUT	Bool	操作已完成但出现错误后, 保持为 TRUE 一个扫描周期时间
STATUS OUT	Word	执行条件代码 (默认值: 0)
DTR OUT	Bool	数据终端就绪, 模块就绪 (输出)。默认值: False
DSR OUT	Bool	数据设备就绪, 通信伙伴就绪 (输入)。默认值: False
RTS OUT	Bool	请求发送, 模块已做好发送准备 (输出)。默认值: False
CTS OUT	Bool	允许发送, 通信伙伴可以接收数据 (输入)。默认值: False
DCD OUT	Bool	数据载波检测, 接收信号电平 (始终为 False, 不支持)
RING OUT	Bool	响铃指示器, 来电指示 (始终为 False, 不支持)




表格 12-30 条件代码

STATUS (W#16#....)	说明
80F0	CM 或 CB 是 RS485 且没有信号可用

### 12.3.1.9 SGN\_SET 指令

表格 12-31 SGN\_SET (设置 RS232 信号) 指令

LAD/FBD	SCL	说明
	<pre>"SGN_SET_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   SIGNAL:=_byte_in_,   RTS:=_bool_in_,   DTR:=_bool_in_,   DSR:=_bool_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_);</pre>	<p>SGN_SET 用于设置 RS232 通信信号的状态。 该功能仅对 RS232 CM 有效。</p>

1 STEP 7 会在插入指令时自动创建 DB。

表格 12-32 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	在该输入的上升沿启动设置 RS232 信号的操作 (默认值: False)
PORT	IN	PORT	安装并组态 CM 或 CB 通信设备之后, 端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。(默认值: 0)
SIGNAL	IN	Byte	选择要设置的信号: (允许多个)。默认值: 0 <ul style="list-style-type: none"> <li>• 01H = 设置 RTS</li> <li>• 02H = 设置 DTR</li> <li>• 04H = 设置 DSR</li> </ul>
RTS	IN	Bool	请求发送, 模块准备好将值发送到设备 (真或假), 默认值: False

## 12.3 点对点 (PtP) 通信

参数和类型		数据类型	说明
DTR	IN	Bool	数据终端就绪，模块准备好将值发送到设备（真或假）。默认值：False
DSR	IN	Bool	数据设备就绪（仅适用于 DCE 型接口），不使用。
DONE	OUT	Bool	上一请求已完成且没有出错后，保持为 TRUE 一个执行周期时间
ERROR	OUT	Bool	上一请求已完成但出现错误后，保持为 TRUE 一个执行周期时间
STATUS	OUT	Word	执行条件代码（默认值：0）

表格 12- 33 条件代码

STATUS (W#16#....)	说明
80F0	CM 或 CB 是 RS485 且无法设置任何信号
80F1	因硬件流控制的原因而无法设置信号
80F2	因模块是 DTE 而无法设置 DSR
80F3	因模块是 DCE 而无法设置 DTR

## 12.3.2 组态通信端口

可以通过以下两种方式组态通信接口：

- 使用 STEP 7 中的设备组态组态端口参数（波特率和奇偶校验）、发送参数和接收参数。设备组态设置存储在 CPU 中。在循环上电和从 RUN 模式切换到 STOP 模式后会应用这些设置。
- 使用 PORT\_CFG (页 613)、SEND\_CFG (页 615) 和 RCV\_CFG (页 616) 指令设置参数。这些指令设置的端口设置在 CPU 处于 RUN 模式期间有效。在切换到 STOP 模式或循环上电后，这些端口设置会恢复为设备组态设置。

组态硬件设备 (页 121) 之后，通过选择机架上的某个 CM 或 CB（如果已组态）来组态通信接口的参数。



巡视窗口中的“属性”(Properties) 选项卡显示所选 CM 或 CB 的参数。选择“端口组态”(Port configuration) 以编辑以下参数：

- 波特率
- 奇偶校验
- 每个字符的数据位数
- 停止位的数目
- 流控制（仅限 RS232）
- 等待时间

对于 CM 1241 RS232 和 RS485 以及 CB RS485（除仅 CM 1241 RS232 支持的流控制外），无论是组态 RS232 或 RS485 通信模块还是 RS485 通信板，端口组态参数都是相同的。但是，参数值可以不同。

对于 CM 1241 RS422/485，将按不同的方式执行端口组态，如下图所示。CM 1241 RS422/485 模块的 422 模式还支持软件流控制。



选择“端口组态”(Port configuration) 以编辑以下 RS422/485 参数：

- “工作模式”(Operating mode):
  - 全双工 (RS422) 四线制模式（点对点连接）
  - 全双工 (RS422) 四线制模式（多点主站）
  - 全双工 (RS422) 四线制模式（多点从站）
  - 半双工 (RS485) 两线制模式
- “接收线路初始状态”(Receive line initial state):
  - 无
  - 正向偏置（信号 R(A) 0V、信号 R(B) 5V）

STEP 7 用户程序还可通过 PORT\_CFG 指令 (页 613) 组态端口或更改现有组态。

### 说明

在用户程序中通过 PORT\_CFG 指令设置的参数值会覆盖设备组态中设定的端口组态设置。请注意，发生掉电时，S7-1200 不会保留通过 PORT\_CFG 指令设置的参数。

参数	定义
波特率	波特率的默认值为 9.6 Kbps。有效选项有：300 波特、600 波特、1.2 Kb、2.4 Kb、4.8 Kb、9.6 Kb、19.2 Kb、38.4 Kb、57.6 Kb、76.8 Kb 和 115.2 Kb。
奇偶校验	奇偶校验的默认值是无奇偶校验。有效选项有：无奇偶校验、偶校验、奇校验、传号（奇偶校验位始终设为 1）和空号（奇偶校验位始终设为 0）。
每个字符的数据位数	字符中的数据位数。有效选择为 7 或 8。
停止位的数目	停止位的数目可以是 1 或 2。默认值是 1。
流控制	对于 RS232 通信模块，可以选择硬件或软件流控制，如“管理流控制 (页 632)”部分所述。如果选择硬件流控制，则可以选择是 RTS 信号始终激活还是切换 RTS。如果选择软件流控制，则可以定义 XON 和 XOFF 字符。 RS485 通信接口不支持流控制。CM 1241 RS422/485 模块的 422 模式支持软件流控制。
等待时间	等待时间是指 CM 或 CB 在断言 RTS 后等待接收 CTS 的时间，或者在接收 XOFF 后等待接收 XON 的时间，具体取决于流控制类型。如果在通信接口收到预期的 CTS 或 XON 之前超过了等待时间，CM 或 CB 将中止传送操作并向用户程序返回错误。指定等待时间，以毫秒表示。范围是 0 到 65535 毫秒。
工作模式	选择工作模式 RS422 或 RS485 以及网络组态。
接收线路初始状态	选择偏置选项。有效值为无、正向偏置和反向偏置。反向偏置用于检测电缆断线。

### 12.3.2.1 管理流控制

流控制是指为了不丢失数据而用来平衡数据发送和接收的一种机制。流控制可确保传送设备发送的信息量不会超出接收设备所能处理的信息量。流控制可以通过硬件或软件来实现。RS232 CM 支持硬件及软件流控制。RS485 CM 和 CB 不支持流控制。CM 1241 RS422/485 模块的 422 模式支持软件流控制。可在组态端口 (页 630) 时或使用 PORT\_CFG 指令 (页 613) 指定流控制类型。

硬件流控制通过请求发送 (RTS, Request To Send) 和允许发送 (CTS, Clear To Send) 通信信号来实现。对于 RS232 CM, RTS 信号从引脚 7 输出, 而 CTS 信号通过引脚 8 接收。RS232 CM 是 DTE (Data Terminal Equipment, 数据终端设备) 设备, 其将 RTS 断言为输出并将 CTS 作为输入来监视。

### 硬件流控制：RTS 切换

如果为 RS232 CM 启用 RTS 切换的硬件流控制，则模块会将 RTS 信号设置为激活状态以发送数据。它还会监视 CTS 信号以确定接收设备是否能接收数据。CTS 信号激活后，只要 CTS 信号保持激活状态，模块便可发送数据。如果 CTS 信号变为非激活状态，则传送必须停止。

CTS 信号变为激活状态时，传送会继续执行。如果 CTS 信号在组态的等待时间内未激活，则模块会中止传送并向用户程序返回错误。在端口组态 (页 630) 中指定等待时间。

对于需要“传送已激活”信号的设备，适合使用 RTS 切换流控制。例如，无线调制解调器使用 RTS 作为“键”信号来激励无线发送器。RTS 切换流控制对于标准电话调制解调器不起作用。对电话调制解调器使用“RTS 始终激活”选项。

### 硬件流控制：RTS 始终激活

在“RTS 始终激活”模式下，CM 1241 默认情况下将 RTS 设置为激活状态。设备（如电话调制解调器等）监视来自 CM 的 RTS 信号，并将该信号用作允许发送信号。调制解调器仅在 RTS 处于激活状态时才向 CM 传送数据，即，电话调制解调器在见到激活的 CTS 信号后发送数据。如果 RTS 处于非激活状态，电话调制解调器不向 CM 传送数据。

要使调制解调器随时都能向 CM 发送数据，请组态“RTS 始终激活”硬件流控制。CM 因此会将 RTS 信号设置为始终激活。即使模块无法接受字符，CM 也不会将 RTS 设置为非激活状态。传送设备必须确保不会使 CM 的接收缓冲区超负荷运行。

### 利用数据终端就绪 (DTR) 和数据设备就绪 (DSR) 信号

对于这两种硬件流控制类型的任何一种，CM 都会将 DTR 设置为激活状态。只有当 DSR 信号变为激活状态时，模块才会进行传送。仅在发送操作开始时评估 DSR 的状态。如果 DSR 在传送操作开始后变为非激活状态，将不能暂停传送操作。

### 软件流控制

软件流控制使用消息中的特殊字符来实现流控制。将组态表示 XON 和 XOFF 的十六进制字符。

XOFF 指示传送必须停止。XON 指示传送可以继续。XOFF 和 XON 不得是相同的字符。

传送设备从接收设备收到 XOFF 字符时，将停止传送。传送设备收到 XON 字符时，传送又继续进行。如果 CM 在通过端口组态 (页 630) 指定的等待时间内没有收到 XON 字符，它将中止传送并向用户程序返回错误。

12.3 点对点 (PtP) 通信

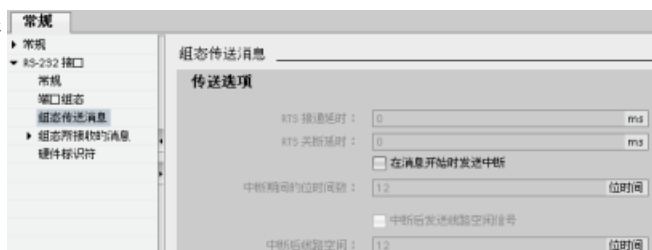
软件流控制需要全双工通信，因为在传送过程中接收伙伴必须能够将 XOFF 发送到传送伙伴。软件流控制只能用于仅包含 ASCII 字符的消息。二进制协议无法使用软件流控制。

12.3.3 组态传送（发送）和接收参数

在 CPU 可进行 PtP 通信前，必须组态传送（或发送）消息和接收消息的参数。这些参数决定了在向目标设备传送消息或从目标设备接收消息时的通信工作方式。

12.3.3.1 组态传送（发送）参数

在设备组态中，可以通过指定所选接口的“已传送消息的组态”(Configuration of transmitted message) 属性，来组态通信接口传送数据的方式。



还可以使用 SEND\_CFG (页 615) 指令，通过用户程序动态组态或更改传送消息参数。

说明

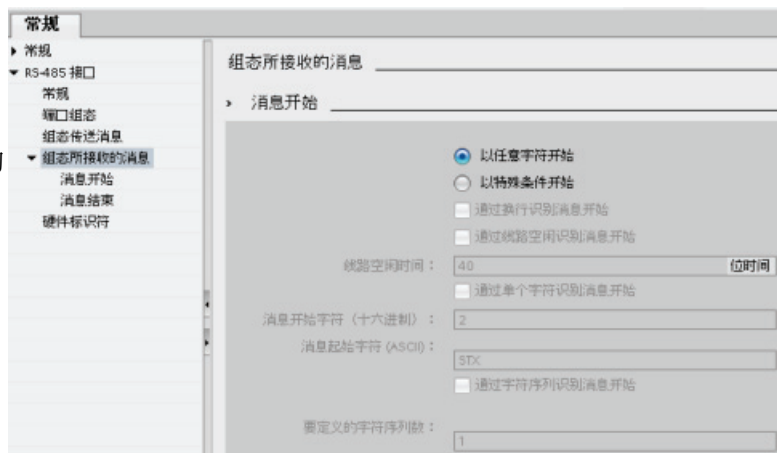
在用户程序中通过 SEND\_CFG 指令设置的参数值会覆盖该端口组态设置。请注意，发生掉电时，CPU 不会保留通过 SEND\_CFG 指令设置的参数。

参数	定义
RTS 接通延时 (RTS On delay)	指定在 RTS 激活后传送启动前需等待的时间。范围是 0 到 65535 ms，默认值为 0。仅当端口组态 (页 630) 指定的是硬件流控制时，该参数才有效。在经过 RTS 接通延迟时间后才会评估 CTS。 该参数仅适用于 RS232 模块。
RTS 关断延时 (RTS Off delay)	指定传送完成后 RTS 禁用前需等待的时间。范围是 0 到 65535 ms，默认值为 0。仅当端口组态 (页 630) 指定的是硬件流控制时，该参数才有效。 该参数仅适用于 RS232 模块。

参数	定义
在消息开始时发送中断 (Send break at message start) 中断期间的位时间数 (Number of bit times in a break)	指定在每条消息开始时，在 RTS 接通延时（如果已组态）已到且 CTS 已激活的情况下先发送中断。 用户指定多少个位的时间构成一个中断，线路在中断期间保持空号状态。默认值为 12，最大值为 65535，即最长 8 秒的限制。
发送中断后线路空闲信号 (Send idle line after a break) 中断后线路空闲 (Idle line after a break)	指定在消息开始前发送线路空闲信号。如果组态了中断，则将在中断后发送。“中断后线路空闲”(Idle line after a break) 参数指定多少个位时间构成一次线路空闲，线路在空闲期间保持传号状态。默认值为 12，最大值为 65535，即最长 8 秒的限制。

### 12.3.3.2 组态接收参数

通过设备组态，可以组态通信接口接收数据以及识别消息开始和结束的方式。在所选接口的接收消息组态中指定以上参数。



还可以在用户程序中使用 RCV\_CFG 指令 (页 616) 来动态组态或更改接收消息参数。

#### 说明

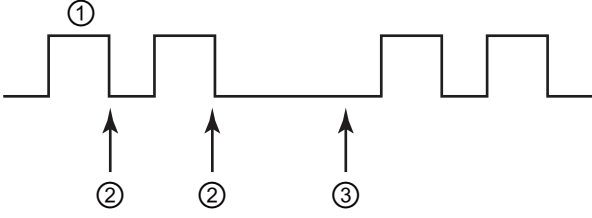
在用户程序中通过 RCV\_CFG 指令设置的参数值会覆盖该端口组态设置。请注意，发生掉电时，CPU 不会保留通过 RCV\_CFG 指令设置的参数。

### 消息开始条件

用户可以决定通信接口识别消息开始的方式。在满足所组态的结束条件之前，开始字符以及组成消息的字符会一直进入接收缓冲区。

12.3 点对点 (PtP) 通信

可以指定多个开始条件。如果指定多个开始条件，则只有在满足所有开始条件后才认为消息开始。例如，如果用户组态了线路空闲时间和特定开始字符，CM 或 CB 将首先查找要满足的线路空闲时间要求，然后 CM 将查找指定的开始字符。如果收到其它某个字符而不是指定的开始字符，CM 或 CB 将通过再次查找线路空闲时间来重新启动消息开始条件搜索。

参数	定义
以任意字符开始	“任意字符”条件指定，成功接收任何字符都将表示消息开始。该字符是消息中的第一个字符。
线路中断	“线路中断”条件指定在接收中断字符后开始消息接收操作。
线路空闲	<p>“线路空闲”条件指定在接收线路空闲或平静了指定时间后开始消息接收操作。一旦出现该条件，即启动消息接收。</p>  <p>① 字符 ② 重启线路空闲定时器 ③ 检测到线路空闲并启动消息接收操作</p>
特殊条件： 通过单个字符识别消息开始	指定通过特殊字符指示消息开始。然后，该字符便成为消息中的第一个字符。在该特定字符前接收到的任何字符都将被丢弃。默认字符是 STX。
特殊条件： 通过字符序列识别消息开始 (Recognize message start with a character sequence)	<p>指定通过最多四个组态序列中的一个特殊字符序列来指示消息开始。可以为每个序列最多指定 5 个字符。对于每个字符位置，可以指定一个特定的十六进制字符，或者指定在序列匹配时忽略该字符（通配符字符）。字符序列中最后一个特定字符终止该开始条件序列。</p> <p>程序根据组态的开始条件对进入序列进行评估，直到满足开始条件为止。只要满足了开始序列，就会开始评估结束条件。</p> <p>最多可组态四个特定字符序列。如果几个不同的字符序列都指示消息开始，则使用多序列开始条件。如果与其中一个字符序列相匹配，消息就会开始。</p>

检查开始条件的顺序是：



- 线路空闲
- 线路中断
- 字符或字符序列

检查多个开始条件时，如果有一个条件没有满足，则 **CM** 或 **CB** 将从第一个所需的条件开始重新启动检查。**CM** 或 **CB** 确定已满足启动条件后，将开始评估结束条件。

### 组态示例 - 消息在两个字符序列出现一个时开始

请注意以下消息开始条件组态：

通过字符序列识别消息开始

定义的字符序列数： 2

#### 5 字符消息的起始序列

消息开始序列 1

检查字符 1

字符值 (十六进制)： 6A

字符值 (ASCII)： J

检查字符 2

字符值 (十六进制)： 0

字符值 (ASCII)： ANY

检查字符 3

字符值 (十六进制)： 0

字符值 (ASCII)： ANY

检查字符 4

字符值 (十六进制)： 0

字符值 (ASCII)： ANY

检查字符 5

字符值 (十六进制)： 1C

字符值 (ASCII)： FS

## 12.3 点对点 (PtP) 通信

消息开始序列 2

检查字符 1

字符值 (十六进制) : : 0

字符值 (ASCII) : : ANY

检查字符 2

字符值 (十六进制) : : 6A

字符值 (ASCII) : :

检查字符 3

字符值 (十六进制) : : 6A

字符值 (ASCII) : :

检查字符 4

字符值 (十六进制) : : 0

字符值 (ASCII) : : ANY

检查字符 5

字符值 (十六进制) : : 0

字符值 (ASCII) : : ANY

对于该组态，只要出现其中一个序列，即会满足开始条件：

- 接到一个由五个字符构成的序列，且其第一个字符是 **0x6A** 而第五个字符是 **0x1C** 时。对于该组态，位置 **2、3 和 4** 的字符可以是任意字符。在接到第五个字符后，将开始评估结束条件。
- 接到两个连续的 **0x6A** 字符（前面为任意字符）时。在这种情况下，会在接到第二个 **0x6A** 后开始评估结束条件（**3** 个字符）。第一个 **0x6A** 前面的字符包含在开始条件中。

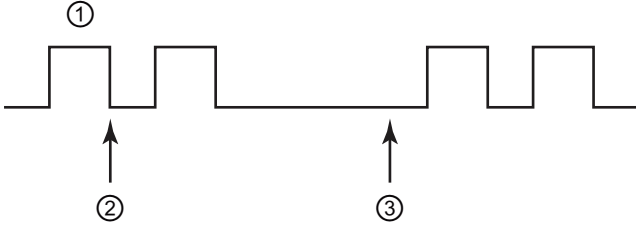
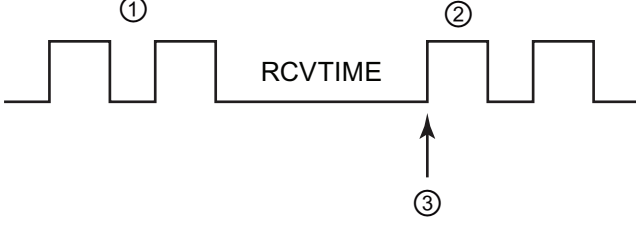
满足该开始条件的实例序列有：

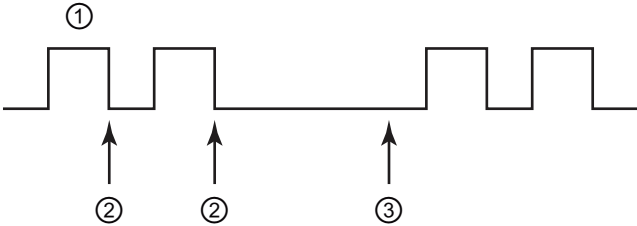
- <任意字符> **6A 6A**
- **6A 12 14 18 1C**
- **6A 44 A5 D2 1C**

## 消息结束条件

用户还可以组态通信接口识别消息结束的方式。可以组态多个消息结束条件。如果出现组态条件中的任何一个，消息就会结束。

例如，可以采用消息超时 **300 ms**、字符间超时 **40** 个位的时间以及最大长度 **50** 个字节作为消息结束的结束条件。如果接收消息的时间超过 **300 ms**、任意两个字符间的间隔超过 **40** 个位的时间或接收到 **50** 个字节，消息即会结束。

参数	定义
通过消息超时识别消息结束 (Recognize message end by message timeout)	<p>经过了组态的消息结束等待时间后，视为消息结束。消息超时时间从满足开始条件开始计算。默认值是 200 ms，有效范围是 0 到 65535 ms。</p>  <p>① 接收的字符            ② 满足消息开始条件：消息定时器启动            ③ 消息定时器时间已到并终止消息</p>
通过响应超时识别消息结束 (Recognize message end by response timeout)	<p>如果在接收到有效的开始序列之前超过了组态的响应等待时间，视为消息结束。响应超时时间从传送结束和 CM 或 CB 开始接收操作时开始计算。默认响应超时时间为 200 ms，取值范围是 0 到 65535 ms。如果在响应时间段 RCVTIME 内没有接收到字符，将向相应的 RCV_PTP 指令返回错误。响应超时不定义具体结束条件。它仅指定必须在指定时间内成功接收字符。用户必须另组态一个结束条件来指示实际的消息结束。</p>  <p>① 传送的字符            ② 接收的字符            ③ 必须在该时间之前成功接收到第一个字符。</p>

参数	定义
通过字符间隙识别消息结束 (Recognize message end by inter-character gap)	<p>经过了组态的消息中两个连续字符间的最大超时后，视为消息结束。字符间隙的默认值是 12 个位的时间，最大值是 65535 个位的时间，即最长 8 秒。</p>  <p>① 接收的字符                      ② 重启字符间定时器                      ③ 字符间定时器时间已到并终止消息。</p>
通过最大长度识别消息结束 (Recognize message end by max length)	<p>在接收到组态的最大字符数后，视为消息结束。最大长度的有效范围是 1 到 1023。</p> <p>使用该条件可以防止消息缓冲区超负荷运行错误。如果将该结束条件与超时结束条件结合使用，在出现超时条件时，即使未达到最大长度也会提供所有有效的已接收字符。仅当最大长度已知时，该条件才支持长度可变的协议。</p>
从消息读取消息长度 (Read message length from message)	<p>消息本身指定消息长度。在接收到指定长度的消息后，视为消息结束。以下说明了用于指定和解释消息长度的方法。</p>
通过字符识别消息结束 (Recognize message end with a character)	<p>在接收到指定的字符后，视为消息结束。</p>
通过字符序列识别消息结束 (Recognize message end with a character sequence)	<p>在接收到指定的字符序列后，视为消息结束。可以指定最多由 5 个字符组成的序列。对于每个字符位置，可以指定一个具体的十六进制字符，或者指定在序列匹配时忽略该字符。</p> <p>结束条件不包括被忽略的前导字符。结束条件包括被忽略的尾随字符。</p>

### 组态示例 - 出现字符序列时结束消息

请注意以下消息结束条件组态：

在这种情况下，在接收到两个连续的 **0x7A** 字符（后跟任意两个字符）时，即满足结束条件。**0x7A 0x7A** 序列前面的字符不是结束字符序列的组成部分。终止结束字符序列时需要跟在 **0x7A 0x7A** 序列后面的两个字符。字符位置 **4** 和 **5** 中接收的值不相关，但必须接收它们才能满足结束条件。

### 在消息中指定消息长度

选择在消息中包括消息长度这一特殊条件时，必须提供三个用于定义消息长度相关信息的参数。

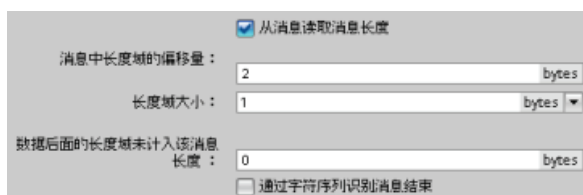
实际消息结构会因所用的协议而变化。三个参数如下所示：

- **n**：消息中出现长度说明符的字符位置（从 **1** 开始）
- **长度大小**：长度说明符的字节数（**1**、**2** 或 **4**）
- **长度 m**：跟在长度说明符后、不包括在长度计数范围内的字符数

结束字符可不连续。“长度 **m**”值可用于指定大小不包含在长度字段中的校验和字段的长度。

12.3 点对点 (PtP) 通信

这些字段位于设备属性的接收消息组态中：



**实例 1：** 假设某条消息是根据以下协议构造的：

STX	Len (n)	长度计数包括字符 3 到 14											
		ADR	PKE		INDEX		PWD		STW		HSW		BCC
1	2	3	4	5	6	7	8	9	10	11	12	13	14
STX	0x0C	xx	xxxx		xxxx		xxxx		xxxx		xxxx		xx

请按以下说明组态该消息的接收消息长度参数：

- n = 2 (消息长度从字节 2 开始。)
- 长度大小 = 1 (消息长度在一个字节中定义。)
- 长度 m = 0 (长度说明符后没有不包括在长度计数中的字符。长度说明符后有 12 个字符。)

在本例中，从 3 到 14 (包括 3 和 14) 的字符都是 Len (n) 计数的字符。

**实例 2：** 假设另一条消息是根据以下协议构造的：

SD1	Len (n)	Len (n)	SD2	长度计数包括字符 5 到 10						FCS	ED
				DA	SA	FA	数据单元 = 3 个字节				
1	2	3	4	5	6	7	8	9	10	11	12
xx	0x06	0x06	xx	xx	xx	xx	xx	xx	xx	xx	xx

请按以下说明组态该消息的接收消息长度参数：

- n = 3 (消息长度从字节 3 开始。)
- 长度大小 = 1 (消息长度在一个字节中定义。)
- 长度 m = 3 (长度说明符后有 3 个字符不包括在长度计数中。在本实例的协议中，字符 SD2、FCS 和 ED 不包括在长度计数中。其它 6 个字符均包括在长度计数中；因此，长度说明符后总共有 9 个字符。)

在本例中，从 5 到 10 (包括 5 和 10) 的字符都是 Len (n) 计数的字符。

### 12.3.4 设计 PtP 通信

STEP 7 提供了一些扩展指令，使得用户程序能够使用程序中设计和指定的协议来执行点对点通信。这些指令可以分为以下两类：

- 组态指令
- 通信指令

#### 组态指令

必须先组态通信接口端口以及用于发送数据和接收数据的参数，然后才能通过用户程序执行 PtP 通信。

可以通过设备配置或用户程序中的如下指令，对各个 CM 或 CB 执行端口组态和消息组态：

- PORT\_CFG (页 613)
- SEND\_CFG (页 615)
- RCV\_CFG (页 616)

#### 通信指令

PtP 通信指令使用户程序能够与通信接口交换消息。有关使用这些指令传送数据的信息，请参见 数据一致性 (页 154) 部分。

所有 PtP 功能都是异步运行的。用户程序可以使用轮询架构来确定传送和接收的状态。SEND\_PTP 和 RCV\_PTP 可以同时执行。通信模块和通信板根据需要对传送和接收消息进行缓冲，最大缓冲区大小为 1024 字节。

CM 和 CB 与实际的点对点设备交换消息。消息协议位于一个缓冲区中，该缓冲区与特定通信端口交换信息。缓冲区和端口是发送和接收指令的参数：

- SEND\_PTP (页 622)
- RCV\_PTP (页 625)

以下指令可用于复位接收缓冲区，以及获取和设置特定的 RS232 信号：

- RCV\_RST (页 627)
- SGN\_GET (页 628)
- SGN\_SET (页 629)

### 12.3.4.1 轮询架构

必须循环/周期性调用 S7-1200 点对点指令以检查收到的消息。发送轮训可在发送结束时刻即告知用户程序。

#### 轮询架构：主站

主站的典型轮询顺序如下：

1. SEND\_PTP 指令启动到 CM 或 CB 的传送。
2. 后续扫描期间会执行 SEND\_PTP 指令以轮询传送完成状态。
3. 当 SEND\_PTP 指令指示传送完成时，用户代码可以准备接收响应。
4. RCV\_PTP 指令反复执行以检查响应。在 CM 或 CB 收到响应消息后，RCV\_PTP 指令将响应复制到 CPU 并指示已接收到新数据。
5. 用户程序随即可处理响应。
6. 转到第 1 步并重复该循环。

#### 轮询架构：从站

从站的典型轮询顺序如下：

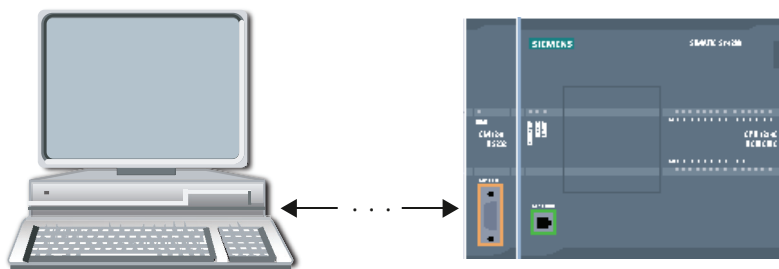
1. 每次扫描用户程序都会执行 RCV\_PTP 指令。
2. CM 或 CB 收到请求后，RCV\_PTP 指令将指示新数据准备就绪并将请求复制到 CPU 中。
3. 用户程序随即处理请求并生成响应。
4. 使用 SEND\_PTP 指令将该响应往回发送给主站。
5. 反复执行 SEND\_PTP 以确保执行传送。
6. 转到第 1 步并重复该循环。

从站在等待响应期间，必须尽量频繁地调用 RCV\_PTP，以便能够在主站超时之前接到来自主站的传送。要完成该任务，用户程序可以从循环 OB 调用 RCV\_PTP，且循环时间应足够大，以便能在超时时间用完之前接到来自主站的传送。如果将 OB 循环时间设置为可在主站的超时时间内执行该指令两次，则用户程序便可接到主站的传送，而不会错过任何传送。



### 12.3.5 示例：点对点通信

在此示例中，S7-1200 CPU 通过 CM 1241 RS232 模块与装有终端仿真器的 PC 通信。此示例中的点对点组态和 STEP 7 程序说明了 CPU 如何从 PC 接收消息并将该消息回送到 PC。



必须将 CM 1241 RS232 模块的通信接口连接到 PC 的 RS232 接口（通常为 COM1）。由于这两个端口都是数据终端设备 (DTE)，所以在连接这两个端口时必须交换接收和发送引脚（引脚 2 和 3），可通过以下任何一种方法实现交换：

- 使用 NULL 调制解调器适配器和标准 RS232 电缆交换引脚 2 和 3。
- 使用已交换引脚 2 和 3 的 NULL 调制解调器电缆。通常可以将电缆两端是否带有两个 9 针 D 型母头连接器作为识别 NULL 调制解调器电缆的依据。

## 12.3.5.1 组态通信模块

可通过 STEP 7 中的设备组态或通过用户程序指令来组态 CM 1241。此示例使用设备组态方法。

- 端口组态：在“设备组态”(Device configuration) 中单击 CM 模块的通信端口，然后如下所示组态该端口：

## 说明

“操作模式”和“接收线路初始状态”的组态设置，只适用于 CM 1241 (RS422/RS485) 模块。其它 CM 1241 模块没有这些端口组态设置。请参见组态 RS422 和 RS485 (页 648)

- 传送消息组态：接受传送消息组态的默认值。在消息开始时将不发送中断信号。
- 接收消息开始组态：将 CM 1241 组态为在通信线路处于非激活状态至少 50 个位时间（在 9600 波特时约为 5 毫秒 =  $50 * 1/9600$ ）时开始接收消息：

消息开始

以任意字符开始  
 以特殊条件开始  
 通过换行识别消息开始  
 通过线路空闲识别消息开始  
 通过单个字符识别消息开始

线路空闲时间： 50 位时间

消息开始字符 (十六进制)： 2

消息起始字符 (ASCII)： STX  
 通过字符序列识别消息开始

要定义的字符序列数： 1

- 接收消息结束组态：将 CM 1241 组态为在最多接收到 100 个字节或换行字符（十进制数 10 或十六进制数 A）时结束消息。结束序列最多允许序列中具有五个结束字符。该序列中的第五个字符是换行字符。前面四个结束序列字符均是“不相关”字符或不选择的字符。CM 1241 不评估“不相关”字符，但会在零或更多“不相关”字符后面寻找指示消息结束的换行字符。

消息结束

定义消息结束条件

通过消息超时识别消息结束  
 通过响应超时识别消息结束  
 通过字符间超时识别消息结束  
 通过最大长度识别消息结束

消息超时： 200 毫秒

响应超时： 200 毫秒

字符间超时： 12 位时间

最大消息长度： 100 bytes  
 从消息读取消息长度

消息中长度域的偏移量： 1 bytes

长度域大小： 1 bytes

数据后面的长度域来计入该消息长度： 0 bytes  
 通过字符序列识别消息结束

5 字符消息结束序列

检查字符 1  
 字符值 (十六进制)： 0  
 字符值 (ASCII)： ANY

检查字符 2  
 字符值 (十六进制)： 0  
 字符值 (ASCII)： ANY

检查字符 3  
 字符值 (十六进制)： 0  
 字符值 (ASCII)： ANY

检查字符 4  
 字符值 (十六进制)： 0  
 字符值 (ASCII)： ANY

检查字符 5  
 字符值 (十六进制)： A  
 字符值 (ASCII)： LF

### 12.3.5.2 组态 RS422 和 RS485

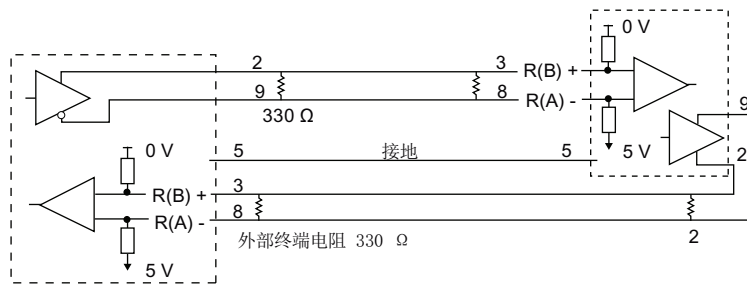
#### 组态 RS422

对于 RS422 模式，有三种工作模式，具体取决于网络组态。根据网络中的设备选择其中一种工作模式。接收线路初始状态的不同选择参考了如下所示的详细情况。

- 全双工 (RS422) 四线制模式（点对点连接）：在网络中有两台设备时选择此选项。在接收线路初始状态中：
  - 在提供偏置和终端时（第 3 种情况），选择无。
  - 选择正向偏置以使用内部偏置和终端（第 2 种情况）。
  - 选择反向偏置以使用内部偏置和终端，并为两台设备启用电缆断线检测（第 1 种情况）。
- 全双工 (RS422) 四线制模式（多点主站）：当网络具有一个主站和多个从站时，为主站选择此选项。在接收线路初始状态中：
  - 在提供偏置和终端时（第 3 种情况），选择无。
  - 选择正向偏置以使用内部偏置和终端（第 2 种情况）。
  - 在此模式下，不能进行电缆断线检测。
- 全双工 (RS422) 四线制模式（多点从站）：当网络具有一个主站和多个从站时，为所有从站选择此选项。在接收线路初始状态中：
  - 在提供偏置和终端时（第 3 种情况），选择无。
  - 选择正向偏置以使用内部偏置和终端（第 2 种情况）。
  - 选择反向偏置以使用内部偏置和终端，并为从站启用电缆断线检测（第 1 种情况）。

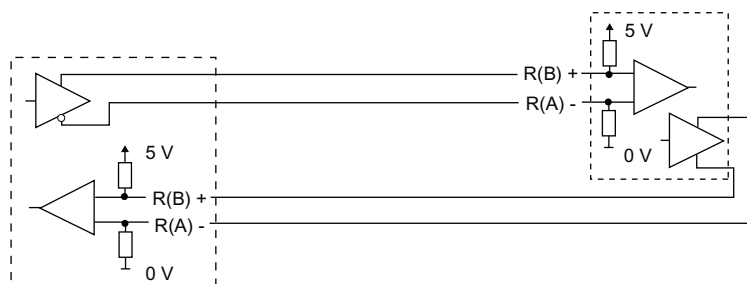
#### 第 1 种情况：RS422，带电缆断线检测

- 工作模式：RS422
- 接收线路初始状态：反向偏置（有偏置， $R(A) > R(B) > 0V$ ）
- 电缆断线：启用电缆断线检测（发送器始终处于激活状态）



### 第 2 种情况：RS422，不带电缆断线检测，正向偏置

- 工作模式：RS422
- 接收线路初始状态：正向偏置（有偏置， $R(B) > R(A) > 0V$ ）
- 电缆断线：无电缆断线检测（发送器仅在发送时才启用）

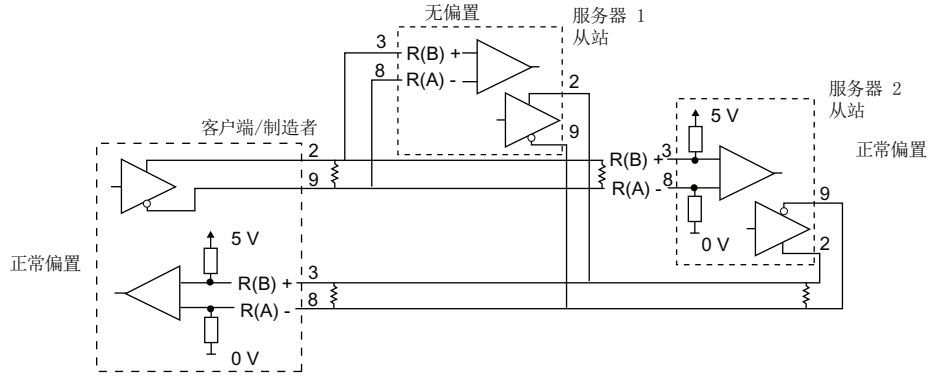


### 第 3 种情况：RS422：不带电缆断线检测，无偏置

- 工作模式：RS422
- 接收线路初始状态：无偏置
- 电缆断线：无电缆断线检测（发送器仅在发送时才启用）

12.3 点对点 (PtP) 通信

偏置和终端由用户在网络末端节点处添加。



组态 RS485

对于 RS485 模式，只有一种工作模式。接收线路初始状态的不同选择参考了如下所示的详细情况。

- 半双工 (RS485) 两线制模式。在接收线路初始状态中：
  - 在提供偏置和终端时（第 5 种情况），选择无。
  - 选择正向偏置以使用内部偏置和终端（第 4 种情况）。

第 4 种情况：RS485：正向偏置

- 工作模式：RS485
- 接收线路初始状态：正向偏置（有偏置， $R(B) > R(A) > 0V$ ）



第 5 种情况：RS485：无偏置（外部偏置）

- 工作模式：RS485
- 接收线路初始状态：无偏置（需要外部偏置）

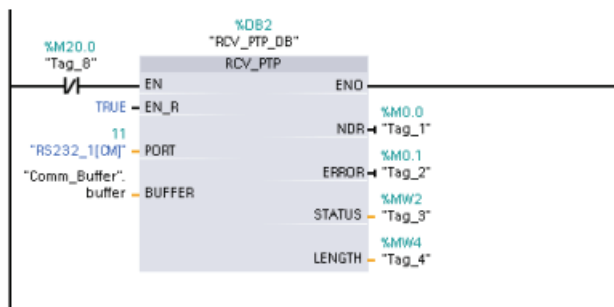


### 12.3.5.3 编写 STEP 7 程序

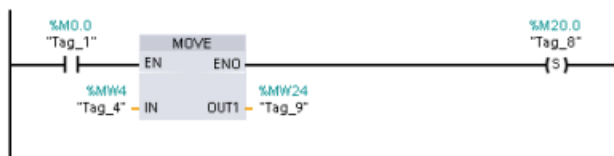
此示例程序使用全局数据块作为通信缓冲区，使用 RCV\_PTP 指令 (页 625) 从终端仿真器接收数据，使用 SEND\_PTP 指令 (页 622) 向终端仿真器回送缓冲数据。要对该示例编程，需要添加数据块组态和程序 OB1，如下所述。

**全局数据块“Comm\_Buffer”：** 创建一个全局数据块 (DB) 并将其命名为“Comm\_Buffer”。在该数据块中创建一个名为“buffer”，数据类型为“字节数组 [0 .. 99]”的值。

**程序段 1：** 只要 SEND\_PTP 未激活，就启用 RCV\_PTP 指令。在程序段 4 中，MW20.0 中的 Tag\_8 在发送操作完成时进行指示，因此是在通信模块相应地准备好接收消息时进行指示。



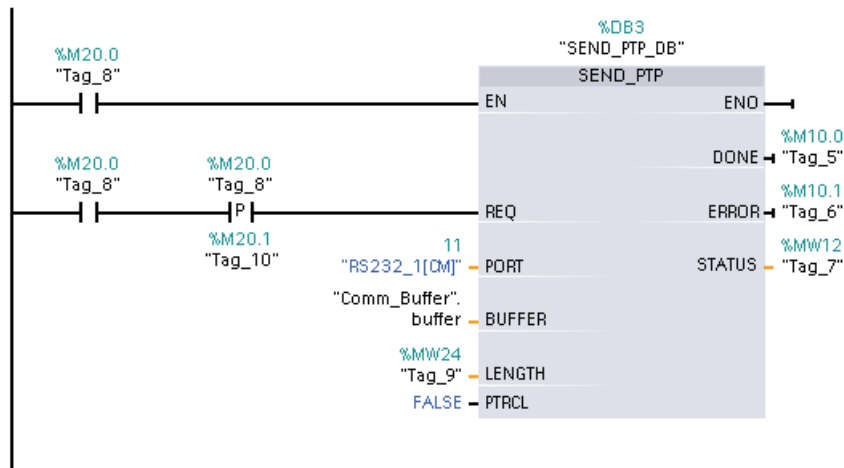
**程序段 2：** 使用由 RCV\_PTP 指令设置的 NDR 值 (M0.0 中的 Tag\_1) 来复制接收到的字节数，并使一个标记 (M20.0 中的 Tag\_8) 置位以触发 SEND\_PTP 指令。



**程序段 3：** M20.0 标记置位时启用 SEND\_PTP 指令。同时还使用此标记将 REQ 输入设置为 TRUE 一个扫描周期时间。REQ 输入会通知 SEND\_PTP 指令要传送新请求。REQ 输入必须仅在 SEND\_PTP 的一个执行周期内设置为 TRUE。每个扫描周期都会执行 SEND\_PTP 指令，直到传送操作完成。CM 1241 传送完消息的最后一个字节时，传

12.3 点对点 (PtP) 通信

送操作完成。传送操作完成后，DONE 输出（M10.0 中的 Tag\_5）将被置位为 TRUE 并持续 SEND\_PTP 的一个执行周期。



程序段 4： 监视 SEND\_PTP 的 DONE 输出并在传送操作完成时复位传送标记（M20.0 中的 Tag\_8）。传送标记复位后，程序段 1 中的 RCV\_PTP 指令可以接收下一条消息。



12.3.5.4 组态终端仿真器

必须设置终端仿真器以支持此示例程序。几乎可以在 PC 上使用任何终端仿真器，例如，超级终端。确定终端仿真器处于断开模式后，如下所述编辑各设置：

1. 将终端仿真器设置为使用 PC 上的 RS232 端口（通常为 COM1）。
2. 将端口组态为 9600 波特、8 个数据位、无奇偶校验（无）、1 个停止位和无流控制。
3. 更改终端仿真器设置使其仿真 ANSI 终端。
4. 组态终端仿真器 ASCII 设置，使其在每行后（用户按下 Enter 键后）发送换行信号。
5. 本地回送字符，以便终端仿真器显示输入的内容。

12.3.5.5 运行示例程序

要运行示例程序，请执行以下步骤：

1. 将 STEP 7 程序下载到 CPU 并确保其处于 RUN 模式。



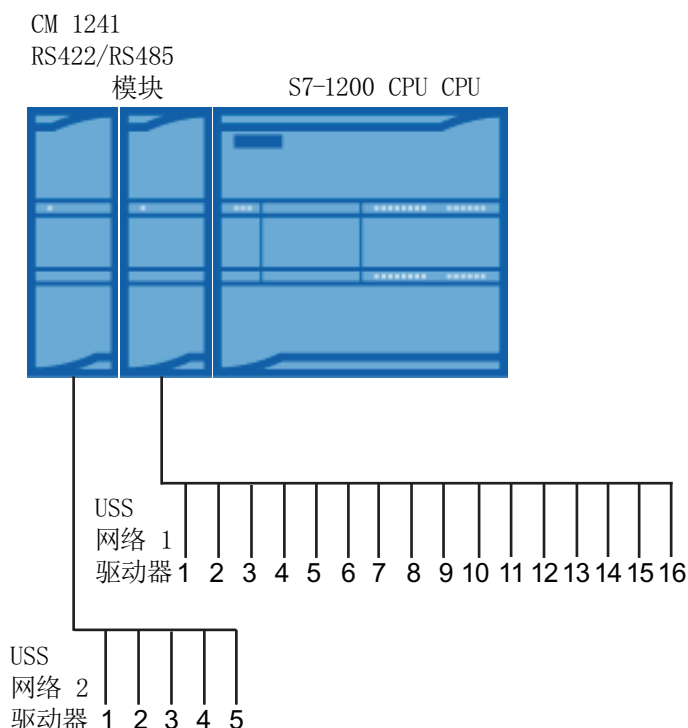
2. 单击终端仿真器上的“连接”(connect) 按钮以应用组态更改并启动与 CM 1241 的终端会话。
3. 在 PC 中键入字符并按 Enter 键。

终端仿真器会将输入的字符发送到 CM 1241 和 CPU。然后，CPU 程序将这些字符回送到终端仿真器。

## 12.4 通用串行接口 (USS) 通信

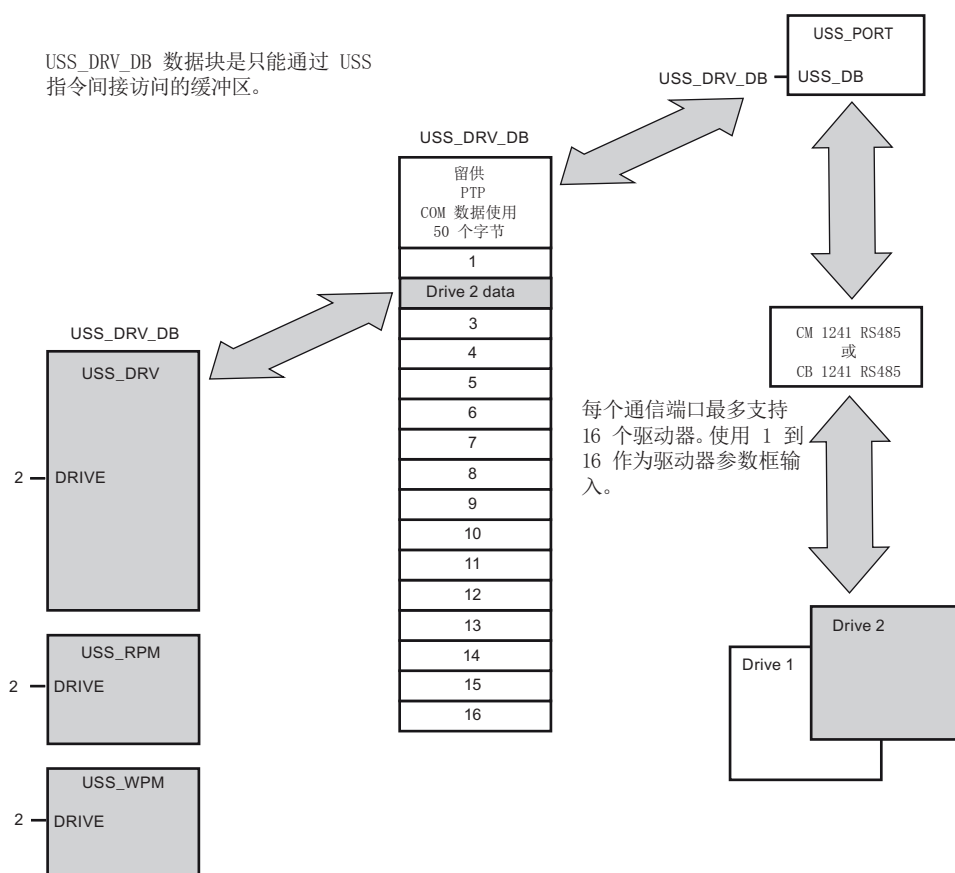
USS 指令可控制支持通用串行接口 (USS) 的电机驱动器的运行。可以使用 USS 指令通过与 CM 1241 RS485 通信模块或 CB 1241 RS485 通信板的 RS485 连接与多个驱动器通信。一个 S7-1200 CPU 中最多可安装三个 CM 1241 RS422/RS485 模块和一个 CB 1241 RS485 板。每个 RS485 端口最多操作十六台驱动器。

USS 协议使用主从网络通过串行总线进行通信。主站使用地址参数向所选从站发送消息。如果未收到传送请求，从站本身不会执行传送操作。各从站之间无法进行直接消息传送。USS 通信以半双工模式执行。以下 USS 图示显示了一个驱动器应用示例的网络图。



### 12.4.1 使用 USS 协议的要求

四条 USS 指令使用 1 个 FB 和 3 个 FC 来支持 USS 协议。一个 USS 网络使用一个 USS\_PORT 背景数据块 (DB)。USS\_PORT 背景数据块包含供该 USS 网络中所有驱动器使用的临时存储区和缓冲区。各 USS 指令共享此数据块中的信息。



连接到一个 RS485 端口的所有驱动器（最多 16 个）是同一 USS 网络的一部分。连接到另一 RS485 端口的所有驱动器是另一 USS 网络的一部分。各 USS 网络通过单独的数据块进行管理。与各 USS 网络相关的所有指令必须共享该数据块。这包括用于控制各 USS 网络上的所有控制器的所有 USS\_DRV、USS\_PORT、USS\_RPM 和 USS\_WPM 指令。

USS\_DRV 指令是函数块 (FB)。在程序编辑器中放置 USS\_DRV 指令时，系统将通过“调用选项”(Call options) 对话框提示您为该 FB 分配 DB。如果对于该 USS 网络而言，它是该程序中的第一条 USS\_DRV 指令，则可以接受默认的 DB 分配（或根据需要更改名称），将相应地创建一个新 DB。但是，如果对于该通道它不是第一条 USS\_DRV 指令，则必须使用“调用选项”(Call options) 对话框中的下拉列表选择先前为该 USS 网络分配的 DB 名称。

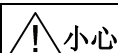
指令 `USS_PORT`、`USS_RPM` 和 `USS_WPM` 全部都是函数 (FC)。在编辑器中放置这些 FC 时不分配 DB。而您必须给这些指令的“`USS_DB`”输入分配合适的 DB 引用。双击该参数字段，然后单击参数助手图标可查看可用的 DB 名称。

`USS_PORT` 函数通过点对点 (PtP) RS485 通信端口处理 CPU 和驱动器之间的实际通信。每次调用此功能可处理与一个驱动器的一次通信。用户程序必须尽快调用此功能以防止与驱动器通信超时。可在主程序循环 OB 或任何中断 OB 中调用此函数。

通常，应在循环中断 OB 中调用 `USS_PORT` 函数。该循环中断 OB 的循环时间应设置为最小调用间隔的一半左右（例如，1200 波特的通信应使用 350 ms 或更短的循环时间）。

用户程序通过 `USS_DRV` 函数块可访问 USS 网络上指定的驱动器。其输入和输出是驱动器的状态和控制。如果网络上有 16 个驱动器，则用户程序必须具有至少 16 个 `USS_DRV` 调用，每个驱动器一个调用。应该以控制驱动器工作所需的速率调用这些块。

只能在主程序循环 OB 中调用 `USS_DRV` 函数块。



小心

只能在主程序循环 OB 中调用 `USS_DRV`、`USS_RPM` 和 `USS_WPM`。可在任何 OB 中调用 `USS_PORT` 函数，通常是在循环中断 OB 中调用。

不要在优先级比 `USS_PORT` 指令所在 OB 的优先级高的 OB 中使用指令 `USS_DRV`、`USS_RPM` 或 `USS_WPM`。例如，不要将 `USS_PORT` 放置在主程序循环 OB 中，而将 `USS_RPM` 放置在循环中断 OB 中。如果未能防止 `USS_PORT` 执行的中断，则会产生意外错误。

`USS_RPM` 和 `USS_WPM` 功能可读取和写入远程驱动器工作参数。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。用户程序可包含尽可能多的这些功能，但在任何特定时刻，每个驱动器只能激活一个读或写请求。只能在主程序循环 OB 中调用 `USS_RPM` 和 `USS_WPM` 函数。

### 计算与驱动器通信所需的时间

与驱动器进行的通信与 S7-1200 扫描周期不同步。在完成一个驱动器通信事务之前，S7-1200 通常完成了多个扫描。

`USS_PORT` 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` 时间间隔。比 `USS_PORT` 间隔更频繁地调用 `USS_PORT` 功能不会增加事务数。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。



## 12.4 通用串行接口 (USS) 通信

表格 12- 34 计算时间要求

波特率	计算的最小 USS_PORT 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

## 12.4.2 USS\_DRV 指令

表格 12- 35 USS\_DRV 指令

LAD/FBD	SCL	说明
<p>默认视图</p>  <p>展开后的视图</p> 	<pre>"USS_DRV_DB" (   RUN:=_bool_in_,   OFF2:=_bool_in_,   OFF3:=_bool_in_,   F_ACK:=_bool_in_,   DIR:=_bool_in_,   DRIVE:=_usint_in_,   PZD_LEN:=_usint_in_,   SPEED_SP:=_real_in_,   CTRL3:=_word_in_,   CTRL4:=_word_in_,   CTRL5:=_word_in_,   CTRL6:=_word_in_,   CTRL7:=_word_in_,   CTRL8:=_word_in_,   NDR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   RUN_EN=&gt;_bool_out_,   D_DIR=&gt;_bool_out_,   INHIBIT=&gt;_bool_out_,   FAULT=&gt;_bool_out_,   SPEED=&gt;_real_out_,   STATUS1=&gt;_word_out_,   STATUS3=&gt;_word_out_,   STATUS4=&gt;_word_out_,   STATUS5=&gt;_word_out_,   STATUS6=&gt;_word_out_,   STATUS7=&gt;_word_out_,   STATUS8=&gt;_word_out_);</pre>	<p>USS_DRV 指令通过创建请求消息和解释驱动器响应消息与驱动器交换数据。每个驱动器应使用一个单独的函数块，但与一个 USS 网络和 PtP 通信端口相关的所有 USS 函数必须使用同一个背景数据块。必须在放置第一个 USS_DRV 指令时创建 DB 名称，然后引用初次指令使用时创建的 DB。</p> <p>STEP 7 会在插入指令时自动创建该 DB。</p>

- <sup>1</sup> LAD 和 FBD: 通过单击功能框的底部, 可展开该功能框, 以显示所有参数。灰显的参数引脚可选, 不需要进行参数分配。

表格 12- 36 参数的数据类型

参数和类型		数据类型	说明
RUN	IN	Bool	驱动器起始位：该输入为真时，将使驱动器以预设速度运行。如果在驱动器运行时 RUN 变为假，电机将减速直至停止。这种行为不同于切断电源 (OFF2) 或对电机进行制动 (OFF3)。
OFF2	IN	Bool	电气停止位：该位为假时，将使驱动器在不经制动情况下逐渐自然停止。
OFF3	IN	Bool	快速停止位：该位为假时，将通过制动的方式使驱动器快速停止，而不只是使驱动器逐渐自然停止。
F_ACK	IN	Bool	故障确认位：设置该位以复位驱动器上的故障位。清除故障后会设置该位，以告知驱动器不再需要指示前一个故障。
DIR	IN	Bool	驱动器方向控制：设置该位以指示方向为向前（对于正 SPEED_SP）。
DRIVE	IN	USInt	驱动器地址：该输入是 USS 驱动器的地址。有效范围是驱动器 1 到驱动器 16。
PZD_LEN	IN	USInt	字长度：这是 PZD 数据的字数。有效值为 2、4、6 或 8 个字。默认值为 2。
SPEED_SP	IN	Real	速度设定值：这是以组态频率的百分比表示的驱动器速度。正值表示方向向前（DIR 为真时）。有效范围是 200.00 到 -200.00。
CTRL3	IN	Word	控制字 3：写入驱动器上用户可组态参数的值。必须在驱动器上组态该参数。（可选参数）
CTRL4	IN	Word	控制字 4：写入驱动器上用户可组态参数的值。必须在驱动器上组态该参数。（可选参数）
CTRL5	IN	Word	控制字 5：写入驱动器上用户可组态参数的值。必须在驱动器上组态该参数。（可选参数）
CTRL6	IN	Word	控制字 6：写入驱动器上用户可组态参数的值。必须在驱动器上组态该参数。（可选参数）
CTRL7	IN	Word	控制字 7：写入驱动器上用户可组态参数的值。必须在驱动器上组态该参数。（可选参数）
CTRL8	IN	Word	控制字 8：写入驱动器上用户可组态参数的值。必须在驱动器上组态该参数。（可选参数）
NDR	OUT	Bool	新数据就绪：该位为真时，表示输出包含新通信请求数据。

参数和类型		数据类型	说明
ERROR	OUT	Bool	出现错误：此参数为真时，表示发生错误，STATUS 输出有效。其它所有输出在出错时均设置为零。仅在 USS_PORT 指令的 ERROR 和 STATUS 输出中报告通信错误。
STATUS	OUT	Word	请求的状态值指示扫描的结果。这不是从驱动器返回的状态字。
RUN_EN	OUT	Bool	运行已启用：该位指示驱动器是否在运行。
D_DIR	OUT	Bool	驱动器方向：该位指示驱动器是否正在向前运行。
INHIBIT	OUT	Bool	驱动器已禁止：该位指示驱动器上禁止位的状态。
FAULT	OUT	Bool	驱动器故障：该位指示驱动器已注册故障。用户必须解决问题，并且在该位被置位时，设置 F_ACK 位以清除此位。
SPEED	OUT	Real	驱动器当前速度（驱动器状态字 2 的标定值）：以组态速度百分数形式表示的驱动器速度值。
STATUS1	OUT	Word	驱动器状态字 1：该值包含驱动器的固定状态位。
STATUS3	OUT	Word	驱动器状态字 3：该值包含驱动器上用户可组态的状态字。
STATUS4	OUT	Word	驱动器状态字 4：该值包含驱动器上用户可组态的状态字。
STATUS5	OUT	Word	驱动器状态字 5：该值包含驱动器上用户可组态的状态字。
STATUS6	OUT	Word	驱动器状态字 6：该值包含驱动器上用户可组态的状态字。
STATUS7	OUT	Word	驱动器状态字 7：该值包含驱动器上用户可组态的状态字。
STATUS8	OUT	Word	驱动器状态字 8：该值包含驱动器上用户可组态的状态字。

首次执行 USS\_DRV 时，将在背景数据块中初始化由 USS 地址（参数 DRIVE）指示的驱动器。完成初始化后，随后执行 USS\_PORT 即可开始与具有此驱动器编号的驱动器通信。

更改驱动器编号操作将要求 CPU 从 STOP 模式切换到 RUN 模式以初始化相应的背景数据块。将输入参数组态到 USS TX 消息缓冲区中，并从“前一个”有效响应缓冲区（如果存在）读取输出。USS\_DRV 执行期间不进行数据传送。驱动器在 USS\_PORT 执行时通信。USS\_DRV 仅组态要发送的消息并解释已从前一个请求中接收的数据。

用户可以使用 DIR 输入 (Bool) 或使用符号（正或负）和 SPEED\_SP 输入 (Real) 控制驱动器旋转方向。下表假定电机按正向旋转接线，说明这些输入如何一起决定驱动器旋转方向。

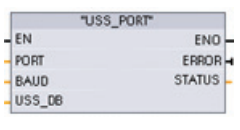
## 12.4 通用串行接口 (USS) 通信

表格 12- 37 SPEED\_SP 和 DIR 参数的交互作用

SPEED_SP	DIR	驱动器旋转方向
数值 > 0	0	反转
数值 > 0	1	正转
数值 < 0	0	正转
数值 < 0	1	反转

## 12.4.3 USS\_PORT 指令

表格 12- 38 USS\_PORT 指令

LAD/FBD	SCL	说明
	<pre>USS_PORT (   PORT:= _uint_in_,   BAUD:= _dint_in_,   ERROR=&gt; _bool_out_,   STATUS=&gt; _word_out_,   USS_DB:= fbtref_inout );</pre>	USS_PORT 指令用于处理 USS 网络上的通信。

表格 12- 39 参数的数据类型

参数和类型		数据类型	说明
PORT	IN	Port	安装并组态 CM 或 CB 通信设备之后，端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。
BAUD	IN	DInt	用于 USS 通信的波特率。
USS_DB	INOUT	USS_BASE	将 USS_DRV 指令放入程序时创建并初始化的背景数据块的名称。
ERROR	OUT	Bool	该输出为真时，表示发生错误，且 STATUS 输出有效。
STATUS	OUT	Word	请求的状态值指示扫描或初始化的结果。对于有些状态代码，还在“USS_Extended_Error”变量中提供了更多信息。

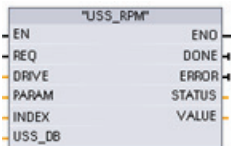
通常，程序中每个 PtP 通信端口只一个 USS\_PORT 指令，且每次调用该功能都会处理与单个驱动器的通信。与同一个 USS 网络和 PtP 通信端口相关的所有 USS 功能都必须使用同一个背景数据块。



用户程序执行 USS\_PORT 指令的次数必须足够多，以防止驱动器超时。通常从循环中断 OB 调用 USS\_PORT 以防止驱动器超时并确保 USS\_DRV 调用可使用最新的 USS 数据更新内容。

#### 12.4.4 USS\_RPM 指令

表格 12- 40 USS\_RPM 指令

LAD/FBD	SCL	说明
	<pre>USS_RPM(REQ:=_bool_in_,         DRIVE:=_usint_in_,         PARAM:=_uint_in_,         INDEX:=_uint_in_,         DONE=&gt;_bool_out_,         ERROR=&gt;_bool_out_,         STATUS=&gt;_word_out_,         VALUE=&gt;_variant_out_,         USS_DB:=_fbtref_inout_);</pre>	<p>USS_RPM 指令用于从驱动器读取参数。与同一个 USS 网络和 PtP 通信端口相关的所有 USS 功能都必须使用同一个数据块。必须从主程序循环 OB 调用 USS_RPM。</p>

表格 12- 41 参数的数据类型

参数类型		数据类型	说明
REQ	IN	Bool	发送请求：REQ 为真时，表示需要新的读请求。如果该参数的请求已处于待决状态，将忽略新请求。
DRIVE	IN	USInt	驱动器地址：DRIVE 是 USS 驱动器的地址。有效范围是驱动器 1 到驱动器 16。
PARAM	IN	UInt	参数编号：PARAM 指示要写入的驱动器参数。该参数的范围是 0 到 2047。在某些驱动器上，最高有效字节可以访问大于 2047 的 PARAM 值。有关如何访问扩展范围的详细信息，请参见驱动器手册。
INDEX	IN	UInt	参数索引：INDEX 指示要写入的驱动器参数索引。索引为一个 16 位值，其中最低有效字节是实际索引值，其范围是 0 到 255。最高有效字节也可供驱动器使用，且取决于具体的驱动器。有关详细信息，请参见驱动器手册。
USS_DB	INOUT	USS_BASE	将 USS_DRV 指令放入程序时创建并初始化的背景数据块的名称。

参数类型		数据类型	说明
VALUE	IN	Word, Int, UInt, DWord, DInt, UInt, Real	这是已读取的参数的值，仅当 DONE 位为真时才有效。
DONE <sup>1</sup>	OUT	Bool	该参数为真时，表示 VALUE 输出包含先前请求的读取参数值。USS_DRV 发现来自驱动器的读响应数据时会设置该位。满足以下条件之一时复位该位：用户通过另一个 USS_WPM 轮询请求响应数据，或在执行接下来两个 USS_DRV 调用的第二个时请求
ERROR	OUT	Bool	出现错误：ERROR 为真时，表示发生错误，并且 STATUS 输出有效。其它所有输出在出错时均设置为零。仅在 USS_PORT 指令的 ERROR 和 STATUS 输出中报告通信错误。
STATUS	OUT	Word	STATUS 表示读请求的结果。对于有些状态代码，还在“USS_Extended_Error”变量中提供了更多信息。

- <sup>1</sup> DONE 位表示已从参考电机驱动器读取有效数据并已将其传送给 CPU。它不表示 USS 库能够立即读取另一参数。必须将空的 PKW 请求发送到电机驱动器并由指令确认，才能使用特定驱动器的参数通道。立即调用指定电机驱动器的 USS\_RPM 或 USS\_WPM FC 将导致 0x818A 错误。

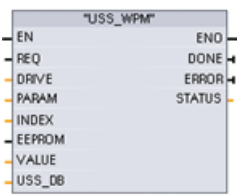
### 12.4.5 USS\_WPM 指令

#### 说明

#### EEPROM 写操作（用于 USS 驱动器内部的 EEPROM）

请勿过多使用 EEPROM 永久写操作。请尽可能减少 EEPROM 写操作次数以延长 EEPROM 的寿命。

表格 12-42 USS\_WPM 指令

LAD/FBD	SCL	说明
	<pre>USS_WPM(REQ:=_bool_in_,         DRIVE:=_usint_in_,         PARAM:=_uint_in_,         INDEX:=_uint_in_,         EEPROM:=_bool_in_,         VALUE:=_variant_in_,         DONE=&gt;_bool_out_,         ERROR=&gt;_bool_out_,         STATUS=&gt;_word_out_,         USS_DB:=_fbtref_inout_);</pre>	<p>USS_WPM 指令用于修改驱动器中的参数。与同一个 USS 网络和 PtP 通信端口相关的所有 USS 功能都必须使用同一个数据块。</p> <p>必须从主程序循环 OB 中调用 USS_WPM。</p>

表格 12-43 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	发送请求：REQ 为真时，表示需要新的写请求。如果该参数的请求已处于待决状态，将忽略新请求。
DRIVE	IN	USInt	驱动器地址：DRIVE 是 USS 驱动器的地址。有效范围是驱动器 1 到驱动器 16。
PARAM	IN	UInt	参数编号：PARAM 指示要写入的驱动器参数。该参数的范围是 0 到 2047。在某些驱动器上，最高有效字节可以访问大于 2047 的 PARAM 值。有关如何访问扩展范围的详细信息，请参见驱动器手册。
INDEX	IN	UInt	参数索引：INDEX 指示要写入的驱动器参数索引。索引为一个 16 位值，其中最低有效字节是实际索引值，其范围是 0 到 255。最高有效字节也可供驱动器使用，且取决于具体的驱动器。有关详细信息，请参见驱动器手册。
EEPROM	IN	Bool	存储到驱动器 EEPROM：该参数为真时，写驱动器参数事务将存储在驱动器 EEPROM 中。如果为假，则写操作是临时的，在驱动器循环上电后不会保留。
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	要写入的参数值。它必须在 REQ 切换时有效。
USS_DB	INOUT	USS_BASE	将 USS_DRV 指令放入程序时创建并初始化的背景数据块的名称。

参数和类型		数据类型	说明
DONE <sup>1</sup>	OUT	Bool	DONE 为真时，表示输入 VALUE 已写入驱动器。USS_DRV 发现来自驱动器的写响应数据时会设置该位。如果用户通过另一个 USS_WPM 轮询请求响应数据，或在执行接下来两个 USS_DRV 调用的第二个时请求响应数据，则复位该位。
ERROR	OUT	Bool	ERROR 为真时，表示发生错误，并且 STATUS 输出有效。其它所有输出在出错时均设置为零。仅在 USS_PORT 指令的 ERROR 和 STATUS 输出中报告通信错误。
STATUS	OUT	Word	STATUS 表示写请求的结果。对于有些状态代码，还在“USS_Extended_Error”变量中提供了更多信息。

- <sup>1</sup> DONE 位表示已从参考电机驱动器读取有效数据并已将其传送给 CPU。它不表示 USS 库能够立即读取另一参数。必须将空的 PKW 请求发送到电机驱动器并由指令确认，才能使用特定驱动器的参数通道。立即调用指定电机驱动器的 USS\_RPM 或 USS\_WPM FC 将导致 0x818A 错误。

### 12.4.6 USS 状态代码

在 USS 功能的 STATUS 输出端返回 USS 指令状态代码。

表格 12-44 STATUS 代码<sup>1</sup>

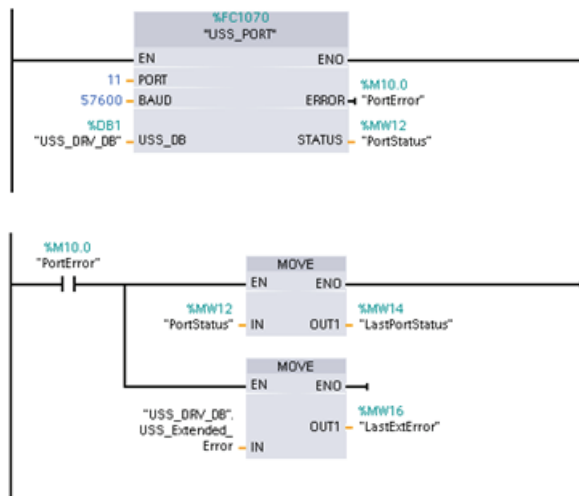
STATUS (W#16#...)	说明
0000	无错误
8180	驱动器响应的长度与从驱动器收到的字符数不匹配。出错的驱动器编号在“USS_Extended_Error”变量中返回。请参见本表格下方的扩展错误描述。
8181	VALUE 参数不是 Word、Real 或 DWord 数据类型。
8182	用户提供了 Word 参数值，但从驱动器响应中收到 DWord 或 Real 值。
8183	用户提供了 DWord 或 Real 参数值，但从驱动器响应中收到 Word 值。
8184	驱动器响应报文的校验和有错误。出错的驱动器编号在“USS_Extended_Error”变量中返回。请参见本表格下方的扩展错误描述。
8185	非法的驱动器地址（有效驱动器地址范围：1 到 16）
8186	速度设定值超出有效范围（有效速度 SP 范围：-200% 到 200%）。
8187	对已发送的请求响应了错误的驱动器编号。出错的驱动器编号在“USS_Extended_Error”变量中返回。请参见本表格下方的扩展错误描述。

STATUS (W#16#....)	说明
8188	指定的 PZD 字长度非法 (有效范围 = 2、4、6 或 8 个字)
8189	指定了非法的波特率。
818A	参数请求通道正在由该驱动器的另一个请求使用。
818B	驱动器尚未对请求和重试做出响应。出错的驱动器编号在“USS_Extended_Error”变量中返回。请参见本表格下方的扩展错误描述。
818C	驱动器返回了有关参数请求操作的扩展错误。请参见本表格下方的扩展错误描述。
818D	驱动器返回了有关参数请求操作的非法访问错误。有关可能限制参数访问的原因信息，请参见驱动器手册。
818E	驱动器尚未初始化。若从未调用过该驱动器的 USS_DRV，则该错误代码将返回到 USS_RPM 或 USS_WPM。这会防止首次扫描 USS_DRV 的初始化过程覆盖未决的参数读/写请求，因为它会将驱动器初始化为新条目。要修复该错误，请针对此驱动器编号调用 USS_DRV。
80Ax-80Fx	从 USS 库调用的 PtP 通信 FB 返回的特定错误 - 这些错误代码值不会被 USS 库修改且在 PtP 指令说明中定义。

<sup>1</sup> 除了上述列出的 USS 指令错误，还可能返回底层 PtP 通信指令 (页 610) 的错误信息。

对于一些 STATUS 代码，在 USS\_DRV 背景数据块的“USS\_Extended\_Error”变量中提供更多信息。对于 STATUS 代码 8180、8184、8187 和 818B (十六进制)，USS\_Extended\_Error 包含出现通信错误的驱动器编号。对于 STATUS 代码 818C (十六进制)，USS\_Extended\_Error 包含使用 USS\_RPM 或 USS\_WPM 指令时从驱动器返回的驱动器错误代码。

仅报告有关 USS\_PORT 指令 (而非 USS\_DRV 指令) 的通信错误 (STATUS = 16#818B)。例如，如果没有正确地终止网络，则驱动器可能切换到 RUN 模式，但 USS\_DRV 指令将为相关输出参数全部显示 0。在这种情况下，只能检测有关 USS\_PORT 指令的通信错误。由于该错误仅在一个扫描周期内可见，所以需要添加一些捕获逻辑，如下面的示例所示。在本例中，当 USS\_PORT 指令的错误位为 TRUE 时，STATUS 和 USS\_Extended\_Error 值将保存到 M 存储器中。当 STATUS 代码值是十六进制的 8180、8184、8187 或 818B 时，驱动器编号将放在 USS\_Extended\_Error 变量中。



网络 1 "PortStatus" 端口状态和 "USS\_DRV\_DB".USS\_Extended\_Error 扩展错误代码值仅在一个程序扫描周期内有效。必须捕获这些值以便后期处理。

网络 2 "PortError" 触点触发将 "PortStatus" 值存储在 "LastPortStatus" 中以及将 "USS\_DRV\_DB".USS\_Extended\_Error 值存储在 "LastExtError" 中。

USS 驱动器支持对驱动器的内部参数进行读写访问。通过该功能可进行驱动器的远程控制和组态。由于发生类似值超出范围或驱动器当前模式的请求非法等错误，驱动器参数访问操作可能会失败。驱动器会生成在 "USS\_Extended\_Error" 变量中返回的错误代码值。该错误代码值仅对 USS\_RPM 或 USS\_WPM 指令的最后一次执行有效。当 STATUS code 值为十六进制的 818C 时，驱动器错误代码将放入 USS\_Extended\_Error 变量中。"USS\_Extended\_Error" 的错误代码值取决于驱动器型号。有关读写参数操作的扩展错误代码的描述，请参见驱动器手册。

## 12.4.7 常规驱动器设置信息

### 常规驱动器设置要求

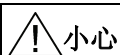
- 驱动器必须设置为使用 4 个 PKW 字。
- 驱动器可组态为使用 2、4、6 或 8 个 PZD 字。
- 驱动器中 PZD 字的数量必须与该驱动器的 USS\_DRV 指令的 PZD\_LEN 输入相匹配。
- 所有驱动器的波特率必须与 USS\_PORT 指令的 BAUD 输入相匹配。
- 驱动器必须设置为可进行远程控制。
- 驱动器必须设置为使用适合通信链路上 USS 的频率设定值。
- 驱动器地址必须设置为 1 到 16，并且与 USS\_DRV 块上对应该驱动器的 DRIVE 输入相匹配。

- 驱动器的方向控制必须设置为使用驱动器设定值的极性。
- 必须正确终止 RS485 网络。

### 连接 MicroMaster 驱动器

本部分以 SIEMENS MicroMaster 驱动器为例提供相关信息。对于其它驱动器，请参见驱动器手册查看相关设置说明。

要建立与 MicroMaster 系列 4 (MM4) 驱动器的连接，请将 RS485 电缆的两端插入两个用于 USS 操作的笼式夹持无螺丝端子中。可使用标准 PROFIBUS 电缆和连接器连接 S7-1200。

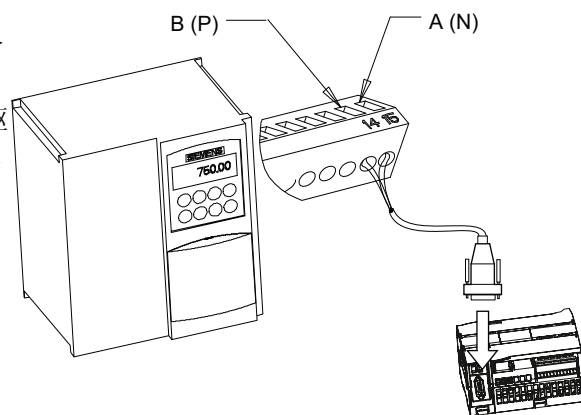


小心

#### 参考电位不同的互连设备可导致互连电缆中有不想要的电流流过


这些不想要的电流可引发通信错误或损坏设备。确保要使用通信电缆连接的所有设备都共用一个公共电路参考点，或者进行隔离以防止出现不想要的电流。屏蔽层必须与外壳地或 9 针连接器的引脚 1 连接。建议将 MicroMaster 驱动器上的接线端子 2--0V 与外壳地连接。

RS485 电缆另一端的两根线必须插入 MM4 驱动器的接线板中。要实现 MM4 驱动器上的电缆连接，请去掉驱动器的盖，露出接线板。有关如何去掉特定驱动器的盖的详细信息，请参见 MM4 用户手册。

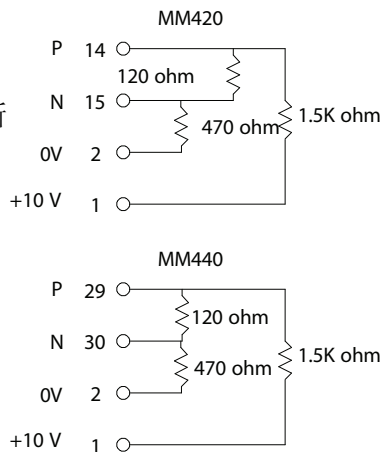


接线板连接标有数字。使用 S7-1200 侧的 PROFIBUS 连接器，将电缆的 A 端子连接到驱动器的端子 15（对于 MM420）或端子 30 (MM440)。将 B (P) A (N) 电缆连接器的 B 端子连接到端子 14 (MM420) 或端子 29 (MM440)。

如果 S7-1200 是网络中的终止节点，或者如果是点对点连接，则需要使用连接器的端子 A1 和 B1（不是 A2 和 B2），这是因为通过这两个端子可进行终端设置（例如，使用 6ES7 972--0BA40--0X40 型号的 DP 连接器）。

 **小心**  
 确保给设备通电之前已正确装上了驱动器盖。

如果驱动器被组态为网络中的终止节点，则还必须将终端电阻和偏压电阻连接到适当的终端连接。此图显示了终止和偏压所需的 MM4 驱动器连接示例。



### 设置 MM4 驱动器

将驱动器与 S7-1200 连接之前，必须确保驱动器具有下列系统参数。使用驱动器上的小键盘设置参数：

1. 恢复驱动器的出厂设置（可选）。	P0010=30 P0970=1
如果跳过步骤 1，则请确保这些参数被设置成所指示的值。	USS PZD 长度 = P2012 索引 0 = (2、4、6 或 8) USS PKW 长度 = P2013 索引 0 = 4
2. 启用对所有参数的读/写访问（专家模式）。	P0003=3
3. 检查驱动器的电机设置。所使用的电机不同，设置也会不同。要设置参数 P304、P305、P307、P310 和 P311，首先必须将参数 P010 设置为 1（快速调试模式）。这些参数设置完毕后，将参数 P010 设置为 0。参数 P304、P305、P307、P310 和 P311 只能在快速调试模式下更改。	P0304 = 电机额定电压 (V) P0305 = 电机额定电流 (A) P0307 = 电机额定功率 (W) P03010 = 电机额定频率 (Hz) P0311 = 电机额定速度
4. 设置本地/远程控制模式。	P0700 索引 0=5
5. 根据通信链路的 USS 选择频率设定值。	P1000 索引 0=5



6. 加速时间（可选） 这是电机加速到最大频率所需的时间（秒）。	P1120 =（0 到 650.00）
7. 减速时间（可选） 这是电机减速到完全停止所需的时间（秒）。	P1121 =（0 到 650.00）
8. 设置串行链路的基准频率：	P2000 =（1 到 650 Hz）
9. 设置 USS 标准化：	P2009 索引 0=0
10. 设置 RS485 串行接口的波特率：	P2010 索引 0= 4（2400 波特） 5（4800 波特） 6（9600 波特） 7（19200 波特） 8（38400 波特） 9（57600 波特） 12（115200 波特）
11. 输入从站地址。 可通过总线操作每台驱动器（最多 31 台）。	P2011 索引 0 =（0 到 31）
12. 设置串行链路超时。 这是两份传入数据报文之间所允许的最长时间段。此功能用于在出现通信故障时关闭反相器。收到有效的数据报文后开始计时。如果在指定的时间段内未收到其它数据报文，则反相器脱扣并显示故障代码 F0070。将该值设置为零将关闭控制。	P2014 索引 0 =（0 到 65,535 ms） （0 = 禁用超时）
13. 将数据从 RAM 传送到 EEPROM：	P0971=1（启动传送）将参数设置更改内容保存到 EEPROM

## 12.5 Modbus 通信

### 12.5.1 Modbus RTU 和 TCP 通信概述

#### Modbus 功能代码

- CPU 作为 Modbus RTU 主站（或 Modbus TCP 客户端）运行时，可在远程 Modbus RTU 从站（或 Modbus TCP 服务器）中读/写数据和 I/O 状态。可以在用户程序中读取和处理远程数据。
- CPU 作为 Modbus RTU 从站（或 Modbus TCP 服务器）运行时，允许监控设置在远程 CPU 中读/写数据和 I/O 状态。监控设备可在远程 CPU 储存器中写入可在用户程序中处理的新值。

表格 12- 45 读取数据功能： 读取远程 I/O 及程序数据

Modbus 功能代码	读取从站（服务器）功能 - 标准寻址
01	读取输出位： 每个请求 1 到 2000 个位
02	读取输入位： 每个请求 1 到 2000 个位
03	读取保持寄存器： 每个请求 1 到 125 个字
04	读取输入字： 每个请求 1 到 125 个字

表格 12- 46 写入数据功能： 写入远程 I/O 及修改程序数据

Modbus 功能代码	写入从站（服务器）功能 - 标准寻址
05	写入一个输出位： 每个请求 1 个位
06	写入一个保持寄存器： 每个请求 1 个字
15	写入一个或多个输出位： 每个请求 1 到 1968 个位
16	写入一个或多个保持寄存器： 每个请求 1 到 123 个字

- Modbus 功能代码 08 和 11 提供从站设备通信诊断信息。
- Modbus 功能代码 0 将消息广播到所有从站（无从站响应）。广播功能不能用于 Modbus TCP，因为通信是以连接为基础的。

表格 12- 47 Modbus 网络站地址

站		地址
RTU 站	标准站地址	1 到 247
	扩展站地址	1 到 65535
TCP 站	站地址	IP 地址和端口号

### Modbus 存储区地址

实际可用的 Modbus 存储区地址数取决于 CPU 型号、存在多少工作存储器以及其他程序数据占用多少 CPU 存储区。下表给出地址范围的额定值。

表格 12- 48 Modbus 存储区地址

站		地址范围
RTU 站	标准存储区地址	10K
	扩展存储区地址	64K
TCP 站	标准存储区地址	10K

### Modbus RTU 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 通信

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 使用开放式用户通信 (OUC, Open User Communication) 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 型号所允许的最大连接数 (页 458)。

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）。

---

**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。

---

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。

**程序中的 Modbus RTU 指令**

- **MB\_COMM\_LOAD:** 通过执行一次 MB\_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 MB\_MASTER MB\_SLAVE 指令使用。
- **MB\_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。
- **MB\_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。

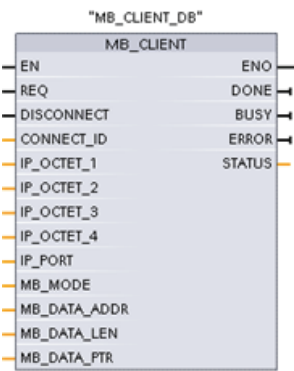
**程序中的 Modbus TCP 指令**

- **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 消息及发送响应

## 12.5.2 Modbus TCP

### 12.5.2.1 MB\_CLIENT (Modbus TCP)

表格 12- 49 MB\_CLIENT 指令

LAD/FBD	SCL	说明
	<pre>"MB_CLIENT_DB" (   REQ:=_bool_in_,   DISCONNECT:=_bool_in_,   CONNECT_ID:=_uint_in_,   IP_OCTET_1:=_byte_in_,   IP_OCTET_2:=_byte_in_,   IP_OCTET_3:=_byte_in_,   IP_OCTET_4:=_byte_in_,   IP_PORT:=_uint_in_,   MB_MODE:=_usint_in_,   MB_DATA_ADDR:=_udint_in_,   MB_DATA_LEN:=_uint_in_,   DONE=&gt;_bool_out_,   BUSY=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   MB_DATA_PTR:=_variant_inout_);</pre>	<p>MB_CLIENT 作为 Modbus TCP 客户端，通过 S7-1200 CPU 上的 PROFINET 连接器进行通信。不需要额外的通信硬件模块。</p> <p>MB_CLIENT 可进行客户端-服务器连接、发送 Modbus 功能请求、接收响应，以及控制 Modbus TCP 服务器的断开。</p>

表格 12- 50 参数的数据类型

参数和类型	数据类型	说明
REQ	In	Bool FALSE = 无 Modbus 通信请求 TRUE = 请求与 Modbus TCP 服务器通信
DISCONNECT	IN	Bool DISCONNECT 参数允许程序控制与 Modbus 服务器设备的连接和断开。 如果 DISCONNECT = 0 且不存在连接，则 MB_CLIENT 尝试连接到分配的 IP 地址和端口号。 如果 DISCONNECT = 1 且存在连接，则尝试断开连接操作。每当启用此输入时，无法尝试其它操作。
CONNECT_ID	IN	UInt CONNECT_ID 参数必须唯一标识 PLC 中的每个连接。 MB_CLIENT 或 MB_SERVER 指令的各唯一实例必须含有一个唯一的 CONNECT_ID 参数。

参数和类型		数据类型	说明
IP_OCTET_1	IN	USInt	Modbus TCP 服务器 IP 地址： 八位位组 1 Modbus TCP 服务器（客户端将通过 Modbus TCP 协议与其进行连接及通信）的 32 位 IPv4 IP 地址中的 8 位部分。
IP_OCTET_2	IN	USInt	Modbus TCP 服务器 IP 地址： 八位位组 2
IP_OCTET_3	IN	USInt	Modbus TCP 服务器 IP 地址： 八位位组 3
IP_OCTET_4	IN	USInt	Modbus TCP 服务器 IP 地址： 八位位组 4
IP_PORT	IN	UInt	默认值 = 502： 服务器（客户端尝试通过 TCP/IP 协议与其连接并最终通信）的 IP 端口号。
MB_MODE	IN	USInt	模式选择： 分配请求类型（读、写或诊断）。请参见下面的 Modbus 功能表了解详细信息。
MB_DATA_ADDR	IN	UDInt	Modbus 起始地址： 分配 MB_CLIENT 访问的数据的起始地址。请参见下面的 Modbus 功能表了解有效地址信息。
MB_DATA_LEN	IN	UInt	Modbus 数据长度： 分配此请求中要访问的位数或字数。请参见下面的 Modbus 功能表了解有效长度信息。
MB_DATA_PTR	IN_OUT	Variant	指向 Modbus 数据寄存器的指针： 寄存器缓冲数据进入 Modbus 服务器或来自 Modbus 服务器。该指针必须分配一个标准全局 DB 或一个 M 存储器地址。
DONE	OUT	Bool	上一请求已完成且没有出错后，DONE 位将保持为 TRUE 一个扫描周期时间。
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 - 无 MB_CLIENT 操作正在进行</li> <li>• 1 - MB_CLIENT 操作正在进行</li> </ul>
ERROR	OUT	Bool	MB_CLIENT 执行因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个循环周期内有效。
STATUS	OUT	Word	执行条件代码

## REQ 参数

FALSE = 无 Modbus 通信请求

TRUE = 请求与 Modbus TCP 服务器通信

如果 MB\_CLIENT 的实例没有激活且参数 DISCONNECT=0，当 REQ=1 时，将启动一个新的 Modbus 请求。如果尚未建立连接，则建立一个新的连接。

如果在当前请求完成前 DISCONNECT=0 且 REQ=1，从而再次执行 MB\_CLIENT 的同一个实例，则不会进行后续 Modbus 传送。但是，一旦完成当前请求，如果通过 REQ=1 执行 MB\_CLIENT，可处理新的请求。

完成当前 MB\_CLIENT 通信请求后，DONE 位将在一个周期内保持为 TRUE。DONE 位可用作定时门，对多个 MB\_CLIENT 请求进行排序。

#### 说明

##### MB\_CLIENT 处理期间输入数据的一致性

Modbus 客户端启动 Modbus 操作后，将在内部保存所有输入状态，然后在每次后续调用时进行比较。比较用于确定此特定调用是否是活动客户端请求的发起者。可使用一个公用背景数据块执行多个 MB\_CLIENT 调用。

因此，在主动处理 MB\_CLIENT 操作期间应不改变输入，这一点很重要。若不遵循此规则，MB\_CLIENT 无法确定其为活动实例。

#### MB\_MODE 和 MB\_DATA\_ADDR 参数用于选择 Modbus 通信功能

MB\_DATA\_ADDR 分配要访问数据的起始 Modbus 地址。MB\_CLIENT 指令使用 MB\_MODE 输入而非功能代码输入。

MB\_MODE 和 MB\_DATA\_ADDR 值一起确定实际 Modbus 消息中使用的功能代码。下表列出了 MB\_MODE 参数、Modbus 功能和 Modbus 地址范围之间的对应关系。

表格 12- 51 Modbus 功能

MB_MODE	Modbus 功能	数据长度	操作和数据	MB_DATA_ADDR
0	01	1 到 2000	读取输出位： 每个请求 1 到 2000 个位	1 到 9999
0	02	1 到 2000	读取输入位： 每个请求 1 到 2000 个位	10001 到 19999
0	03	1 到 125	读取保持寄存器： 每个请求 1 到 125 个字	40001 到 49999 或 40001 到 465535
0	04	1 到 125	读取输入字： 每个请求 1 到 125 个字	30001 到 39999
1	05	1	写入一个输出位： 每个请求 1 个位	1 到 9999

## 12.5 Modbus 通信

MB_MOD E	Modbus 功能	数据长度	操作和数据	MB_DATA_ADDR
1	06	1	写入一个保持寄存器： 每个请求 1 个字	40001 到 49999 或 400001 到 465535
1	15	2 到 1968	写入多个输出位： 每个请求 2 到 1968 个位	1 到 9999
1	16	2 到 123	写入多个保持寄存器： 每个请求 2 到 123 个字	40001 到 49999 或 400001 到 465535
2	15	1 到 1968	写入一个或多个输出位： 每个请求 1 到 1968 个位	1 到 9999
2	16	1 到 123	写入一个或多个保持寄存器： 每个请求 1 到 123 个字	40001 到 49999 或 400001 到 465535
11	11	0	读取服务器通信状态字和事件计数器。状态字指示忙闲情况（0 - 不忙，0xFFFF - 忙）。每成功完成一条消息，事件计数器的计数值递增。 对于该功能，MB_CLIENT 的 MB_DATA_ADDR 和 MB_DATA_LEN 参数都将被忽略。	
80	08	1	利用数据诊断代码 0x0000 检查服务器状态（回送测试 - 服务器回送请求） 每个请求 1 个字	
81	08	1	利用数据诊断代码 0x000A 重新设置服务器事件计数器 每个请求 1 个字	
3 到 10、 12 到 79、 82 到 255			保留	



---

**说明****MB\_DATA\_PTR 分配一个缓冲区来存储从 Modbus TCP 服务器读取或写入到该服务器的数据**

数据缓冲区可位于标准的全局 DB 或 M 存储器地址中。

对于 M 存储器中的缓冲区，使用标准的 Any 指针格式。具体格式为 P#“位地址”“数据类型”“长度”，例如 P#M1000.0 WORD 500。

---

**MB\_DATA\_PTR 分配一个通信缓冲区**

- MB\_CLIENT 通信功能：
  - 从 Modbus 服务器地址（00001 到 09999）读写 1 位数据
  - 从 Modbus 服务器地址（10001 到 19999）读取 1 位数据
  - 从 Modbus 服务器地址（30001 到 39999）和（40001 到 49999）读取 16 位字数据
  - 向 Modbus 服务器地址（40001 到 49999）写入 16 位字数据
- 向/从 MB\_DATA\_PTR 分配的 DB 或 M 存储器缓冲区传输字或位大小的数据。
- 如果通过 MB\_DATA\_PTR 分配 DB 为缓冲区，必须为所有 DB 数据元素分配数据类型。
  - 1 位 Bool 数据类型代表一个 Modbus 位地址
  - 16 位单字数据类型（如 WORD、UInt 和 Int）代表一个 Modbus 字地址
  - 32 位双字数据类型（如 DWORD、DInt 和 Real）代表两个 Modbus 字地址
- 可以通过 MB\_DATA\_PTR 分配复杂的 DB 元素，例如
  - 标准数组
  - 指定的结构，其中每个元素都是唯一的。
  - 指定的复杂结构，其中每个元素都具有唯一的名称以及 16 或 32 位数据类型。
- 不要求 MB\_DATA\_PTR 数据区位于同一个全局数据块（或 M 存储器区）中。可分配一个数据块供 Modbus 读取，分配另一个数据块供 Modbus 写入，或分配一个数据块用于各个 MB\_CLIENT 站。

## 多个客户端连接

Modbus TCP 客户端支持的并发连接数最多为 PLC 允许的开放式用户通信最大连接数。PLC 的连接总数（包括 Modbus TCP 客户端和服务端）不得超过支持的开放式用户通信最大连接数 (页 458)。可以在客户端和/或服务端类型的连接间共享 Modbus TCP 连接。

单独的客户端连接必须遵循以下规则：

- 每个 MB\_CLIENT 连接必须使用一个不同的背景数据块
- 每个 MB\_CLIENT 连接必须指定一个唯一的服务器 IP 地址
- 每个 MB\_CLIENT 连接必须指定一个唯一的连接 ID
- 是否需要唯一的 IP 端口号取决于服务器组态

连接 ID 对于每个单独的连接必须是唯一的。这意味着单个的唯一连接 ID 只能与每个单独的背景数据块配合使用。总之，背景数据块和连接 ID 成对使用，且对每个连接必须是唯一的。

表格 12- 52 MB\_CLIENT 实例数据块用户可访问静态变量

变量	数据类型	默认值	说明
Blocked_Proc_Timeout	Real	3.0	在 Modbus 客户端实例受阻后，移除该激活的实例前需等待的时间（秒）。例如，当已发出客户端请求，但应用程序在彻底完成该请求前停止执行该客户端功能时，就会出现这种情况。最大 S7-1200 限值是 55 秒。
MB_Unit_ID	Word	255	Modbus 设备标识符 Modbus TCP 服务器通过其 IP 地址寻址。因此 MB_UNIT_ID 参数不用于 Modbus TCP 寻址。 MB_UNIT_ID 参数与 Modbus RTU 协议中的从站地址相对应。如果 Modbus TCP 服务器用于采用 Modbus RTU 协议的网关，MB_UNIT_ID 可用于标识在串行网络上连接的从站设备。MB_UNIT_ID 将用于将请求转发给正确的 Modbus RTU 从站地址。 请注意，某些 Modbus TCP 设备可能需要在受限的值范围内初始化 MB_UNIT_ID 参数。
RCV_TIMEOUT	Real	2.0	MB_CLIENT 等待服务器响应请求的时间（秒）。
已连接	Bool	0	指示与所分配服务器的连接是已接通还是已断开： 1 = 接通，0 = 断开

表格 12- 53 MB\_CLIENT 协议错误

STATUS (W#16#)	发送到 Modbus 客户端的响应代码 (B#16#)	Modbus 协议错误
8381	01	不支持此功能代码
8382	03	数据长度错误
8383	02	数据地址错误或访问的数据超出 MB_HOLD_REG 地址区的界限
8384	03	数据值错误
8385	03	不支持此数据诊断代码值（功能代码 08）

表格 12- 54 MB\_CLIENT 执行条件代码<sup>1</sup>

STATUS (W#16#)	MB_CLIENT 参数错误
7001	MB_CLIENT 正在等待 Modbus 服务器响应指定 TCP 端口处的连接或断开连接请求。仅在第一次执行连接或断开操作时才报告此代码。
7002	MB_CLIENT 正在等待 Modbus 服务器响应指定 TCP 端口处的连接或断开连接请求。等待连接或断开操作完成时，将针对任何后续执行报告此代码。
7003	断开操作已成功完成（仅在一个 PLC 扫描周期内有效）。
80C8	服务器在分配的时间内无响应。MB_CLIENT 必须在分配的时间内使用最初传送的事务 ID 接收响应，否则将返回此错误。检查与 Modbus 服务器设备的连接。尝试过任何组态的重试操作（若适用）后，才报告此错误。
8188	模式值无效
8189	数据地址值无效
818A	数据长度值无效
818B	指向 DATA_PTR 区的指针无效。可以是 MB_DATA_ADDRESS 与 MB_DATA_LEN 的组合。
818C	指向优化的 DATA_PTR 区（必须是标准 DB 区或 M 储存器区）的指针
8200	端口正忙于处理现有的 Modbus 请求。
8380	接收到的 Modbus 帧有缺陷或接收到的字节太少。

12.5 Modbus 通信

<b>STATUS (W#16#)</b>	<b>MB_CLIENT 参数错误</b>
8387	分配的连接 ID 参数和用于先前请求的 ID 不同。只能有一个单个连接 ID 与每个 MB_CLIENT 背景数据块配合使用。 如果从一个服务器接收到的 Modbus TCP 协议 ID 不是 0，也可作为内部错误使用。
8388	Modbus 服务器返回一些和请求内容不同的数据。这只适用于 Modbus 功能 15 或 16。

1 除了上面列出的 MB\_CLIENT 错误外，也可以从底层传输块通信指令（TCON、TDISCON、TSEND 和 TRCV）返回错误。

参见

TCON、TDISCON、TSEND 和 TRCV (页 473)

12.5.2.2 MB\_SERVER (Modbus TCP)

表格 12- 55 MB\_SERVER 指令

LAD/FBD	SCL	说明
	<pre>"MB_SERVER_DB" (   DISCONNECT:=_bool_in_,   CONNECT_ID:=_uint_in_,   IP_PORT:=_uint_in_,   NDR=&gt;_bool_out_,   DR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   MB_HOLD_REG:=_variant_inout_);</pre>	<p>MB_SERVER 作为 Modbus TCP 服务器，通过 S7-1200 CPU 上的 PROFINET 连接器进行通信。不需要额外的通信硬件模块。</p> <p>MB_SERVER 可接收与 Modbus TCP 客户端连接请求、接收 Modbus 功能请求，以及发送响应消息。</p>

表格 12- 56 参数的数据类型

参数和类型		数据类型	说明
DISCONNECT	IN	Bool	MB_SERVER 尝试与伙伴设备进行“被动”连接。也就是说，服务器被动地侦听来自任何请求 IP 地址的 TCP 连接请求。 如果 DISCONNECT = 0 且连接不存在，则可以启动被动连接。 如果 DISCONNECT = 1 且连接存在，则启动断开操作。这允许程序控制何时接受连接。每当启用此输入时，无法尝试其他操作。
CONNECT_ID	IN	UInt	CONNECT_ID 唯一标识 PLC 中的每个连接。MB_CLIENT 或 MB_SERVER 指令的各唯一实例必须含有一个唯一的 CONNECT_ID 参数。
IP_PORT	IN	UInt	默认值 = 502：用来标识 IP 端口的 IP 端口号，将监视该端口是否有来自 Modbus 客户端的连接请求。 以下 TCP 端口号不允许用于 MB_SERVER 被动连接：20、21、25、80、102、123、5001、34962、34963 和 34964。
MB_HOLD_REG	IN_OUT	Variant	指向 MB_SERVER Modbus 保持寄存器的指针：保持寄存器必须是一个标准全局 DB 或 M 存储区地址。储存区用于保存值，允许 Modbus 客户端使用 Modbus 寄存器功能 3（读）、6（写）和 16（写）访问这些值。
NDR	OUT	Bool	新数据就绪：0 = 没有新数据，1 = 表示 Modbus 客户端已写入新数据
DR	OUT	Bool	数据读取：0 = 没有读取数据，1 = 表示 Modbus 客户端已读取该数据。
ERROR	OUT	Bool	MB_SERVER 执行因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个循环周期内有效。
STATUS	OUT	Word	执行条件代码

MB\_SERVER 允许进入的 Modbus 功能代码（1、2、4、5 和 15）在 S7-1200 CPU 的输入过程映像及输出过程映像中直接读或写位和字。对于数据传输功能代码（3、6 和 16），MB\_HOLD\_REG 参数必须定义为大于一个字节的类型。下表显示了 Modbus 地址到 CPU 中过程映像的映射。

表格 12- 57 Modbus 地址到过程映像的映射

Modbus 功能					S7-1200		
代码	功能	数据区	地址范围		数据区	CPU 地址	
01	读位	输出	1	到	8192	输出过程映像	Q0.0 到 Q1023.7
02	读位	输入	10001	到	18192	输入过程映像	I0.0 到 I1023.7
04	读字	输入	30001	到	30512	输入过程映像	IW0 到 IW1022
05	写位	输出	1	到	8192	输出过程映像	Q0.0 到 Q1023.7
15	写位	输出	1	到	8192	输出过程映像	Q0.0 到 Q1023.7

进入的 Modbus 消息功能代码（3、6 和 16）在 Modbus 保持寄存器中读取或写入字，该寄存器可以是 M 存储区地址范围或数据块。保持寄存器的类型由 MB\_HOLD\_REG 参数指定。

#### 说明

##### MB\_HOLD\_REG 参数分配

Modbus 保持寄存器可以位于标准全局 DB 或 M 存储区地址中。

对于 M 存储区地址中的 Modbus 保持寄存器，使用标准的 Any 指针格式。其格式为 P#“位地址”“数据类型”“长度”。例如 P#M1000.0 WORD 500

下表给出了 Modbus 地址到保持寄存器的映射示例，这种映射用于 Modbus 功能代码 03（读取字）、06（写入字）和 16（写入字）。DB 地址的实际上限取决于每种 CPU 型号的最大工作存储器限值和 M 存储器限值。

表格 12- 58 Modbus 地址到 CPU 存储器地址的映射示例

Modbus 地址	MB_HOLD_REG 参数示例		
	P#M100.0 Word 5	P#DB10.DBx0.0 Word 5	"Recipe".ingredient
40001	MW100	DB10.DBW0	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	"Recipe".ingredient[5]

## 多个服务器连接

可以创建多个服务器连接。这允许单个 PLC 建立与多个 Modbus TCP 客户端的并发连接。

Modbus TCP 服务器支持的并发连接数最多为 PLC 允许的开放式用户通信最大连接数。PLC 的连接总数（包括 Modbus TCP 客户端和服务端）不得超过支持的开放式用户通信最大连接数（页 458）。可以在客户端和/或服务端类型的连接间共享 Modbus TCP 连接。

单独的服务器连接必须遵循以下规则：

- 每个 MB\_SERVER 连接必须使用一个不同的背景数据块。
- 必须通过一个唯一的 IP 端口号建立每个 MB\_SERVER 连接。每个端口只能用于 1 个连接。
- 每个 MB\_SERVER 连接必须使用一个唯一的连接 ID。
- 必须为每个连接（带有各自的背景数据块）单独调用 MB\_SERVER。

连接 ID 对于每个单独的连接必须是唯一的。这意味着单个的唯一连接 ID 只能与每个单独的背景数据块配合使用。总之，背景数据块和连接 ID 成对使用，且对每个连接必须是唯一的。

表格 12- 59 Modbus 诊断功能代码

MB_SERVER Modbus 诊断功能		
代码	子功能	说明
08	0x0000	返回查询数据回送测试：MB_SERVER 将向 Modbus 客户端回送接收到的数据字。
08	0x000A	清除通信事件计数器：MB_SERVER 将清除用于 Modbus 功能 11 的通信事件计数器。
11		获取通信事件计数器：MB_SERVER 使用内部通信事件计数器来记录发送到 Modbus 服务器的 Modbus 成功读取和写入请求次数。该计数器不会因功能 8 或功能 11 请求而增加。同时也不会因导致通信错误的任何请求而增加。 广播功能不能用于 Modbus TCP，因为在任何时刻仅存在一个客户端-服务器连接。

**MB\_SERVER 变量**

下表给出了存储在 MB\_SERVER 背景数据块中的公共静态变量（可在用户程序中使用）。

表格 12- 60 MB\_SERVER 公共静态变量

变量	数据类型	默认值	说明
HR_Start_Offset	Word	0	指定 Modbus 保持寄存器的起始地址
Request_Count	Word	0	该服务器接收到的所有请求的数量。
Server_Message_Count	Word	0	该特定服务器接收到的请求的数量。
Xmt_Rcv_Count	Word	0	出现错误的传输或接收的数量。此外，如果接收到一条无效的 Modbus 消息，该值加 1。
Exception_Count	Word	0	需要返回例外的 Modbus 特定错误数
Success_Count	Word	0	该特定服务器接收到的没有协议错误的请求数量。
已连接	Bool	0	指示与所分配客户端的连接是已接通还是已断开： 1 = 接通，0 = 断开

用户程序可以将值写入 HR\_Start\_Offset，以控制 Modbus 服务器操作。可读取其它变量以监视 Modbus 的状态。

**HR\_Start\_Offset**

Modbus 保持寄存器地址从 40001 开始。这些地址与保持寄存器的 PLC 存储器起始地址对应。不过，可以组态“HR\_Start\_Offset”变量，将 Modbus 保持寄存器的起始地址定义为除 40001 之外的其它值。

例如，如果保持寄存器被组态为起始于 MW100 并且长度为 100 个字。偏移量 20 可指定保持寄存器的起始地址为 40021 而不是 40001。低于 40021 和高于 40119 的任何地址都将导致寻址错误。

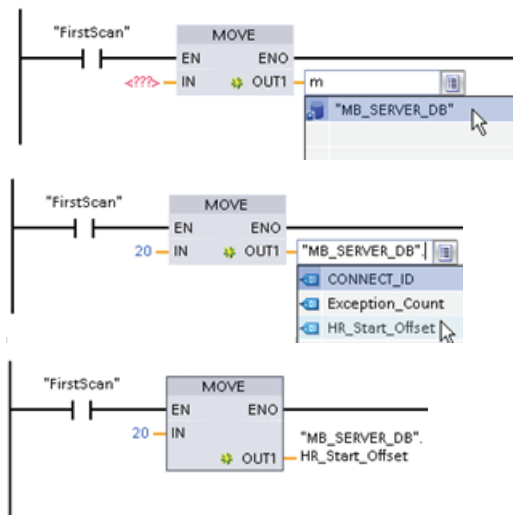


表格 12- 61 Modbus 保持寄存器寻址示例

HR_Start_Offset	地址	最小值	最大值
0	Modbus 地址 (字)	40001	40099
	S7-1200 地址	MW100	MW298
20	Modbus 地址 (字)	40021	40119
	S7-1200 地址	MW100	MW298

HR\_Start\_Offset 是一个字值，用于指定 Modbus 保持寄存器的起始地址，存储在 MB\_SERVER 背景数据块中。将 MB\_SERVER 放入程序后，可利用参数助手下拉列表设置该公共静态变量值。

例如，将 MB\_SERVER 放入 LAD 网络后，可以切换到先前的网络，并分配 HR\_Start\_Offset 值。该值必须在执行 MB\_SERVER 前分配。



使用默认 DB 名称

输入 Modbus 服务器变量：

1. 将光标放在参数字段中，然后输入 m 字符。
2. 从 DB 名称的下拉列表中选择“MB\_SERVER\_DB”。
3. 从 DB 变量的下拉列表中选择“MB\_SERVER\_DB.HR\_Start\_Offset”。

表格 12- 62 MB\_SERVER 执行条件代码<sup>1</sup>

STATUS (W#16#)	发送到 Modbus 服务器的响应代码 (B#16#)	Modbus 协议错误
7001		MB_SERVER 正在等待 Modbus 客户端连接到指定的 TCP 端口。仅在第一次执行连接或断开操作时才报告此代码。
7002		MB_SERVER 正在等待 Modbus 客户端连接到指定的 TCP 端口。等待连接或断开操作完成时，将针对任何后续执行报告此代码。
7003		断开操作已成功完成（仅在一个 PLC 扫描周期内有效）。
8187		指向 MB_HOLD_REG 的指针无效：区域太小
818C		指向优化的 MB_HOLD_REG 区（必须是标准 DB 区或 M 储存器区）的指针或受阻的过程超时超过 55 秒的限值。（仅适用于 S7-1200）
8381	01	不支持此功能代码
8382	03	数据长度错误
8383	02	数据地址错误或访问的数据超出 MB_HOLD_REG 地址区的界限
8384	03	数据值错误
8385	03	不支持此数据诊断代码值（功能代码 08）

<sup>1</sup>除了上面列出的 MB\_SERVER 错误外，也可以从底层传输块通信指令（TCON、TDISCON、TSEND 和 TRCV）返回错误。

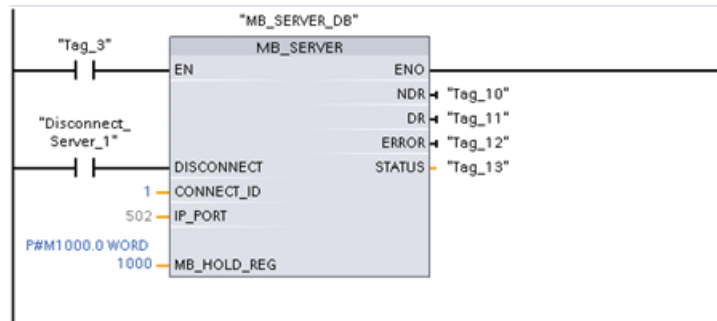
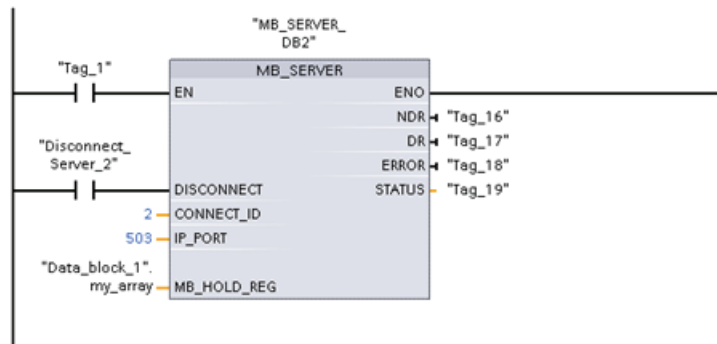
## 参见

TCON、TDISCON、TSEND 和 TRCV (页 473)

### 12.5.2.3 MB\_SERVER 示例：多个 TCP 连接

可以拥有多个 Modbus TCP 服务器连接。为此，必须为每个连接单独执行 MB\_SERVER。每个连接必须使用单独的背景数据块、连接 ID 和 IP 端口。S7-1200 仅允许每个 IP 端口进行一个连接。

为达到最佳性能，每个连接的每次循环都必须执行 MB\_SERVER。

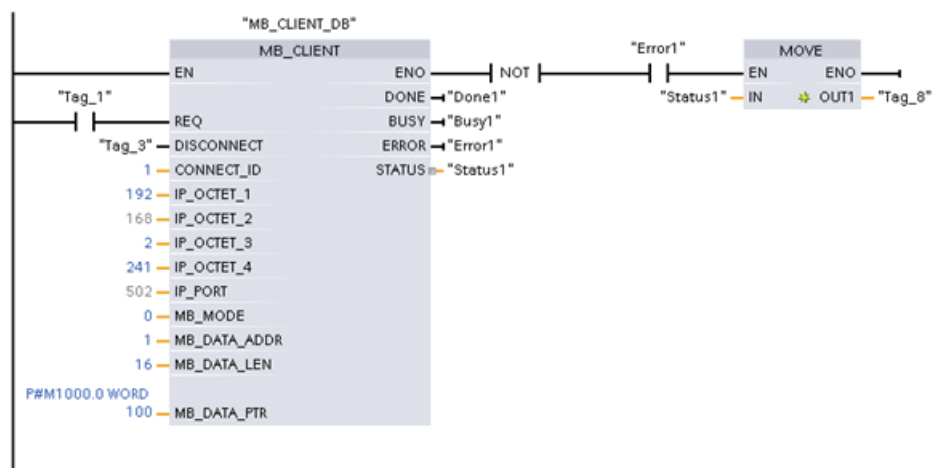
**程序段 1:** 带有独立 IP\_PORT、连接 ID 和背景数据块的 1 号连接**程序段 2:** 带有独立 IP\_PORT、连接 ID 和背景数据块的 2 号连接**12.5.2.4 MB\_CLIENT 示例 1: 通过公共 TCP 连接发送多个请求**

多个 Modbus 客户端请求可通过同一连接发送。为此，必须使用相同的背景数据块、连接 ID 和端口号。

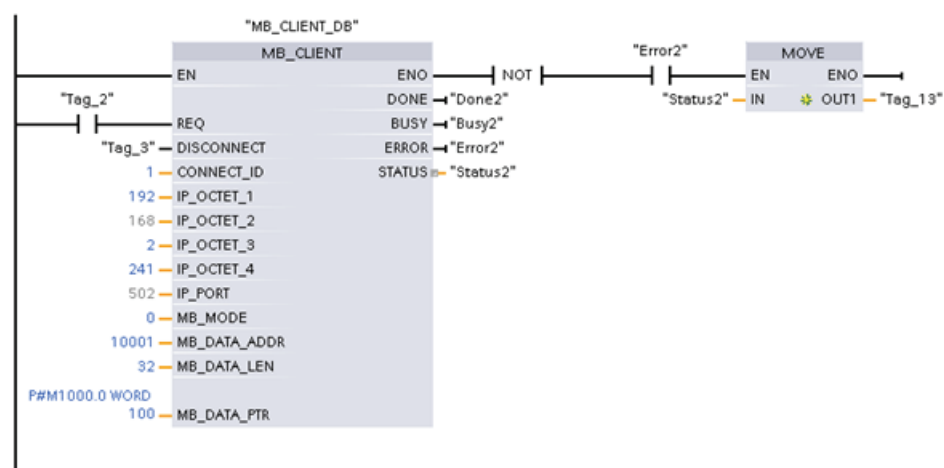
在任意给定时间，只能有一个客户端处于激活状态。在一个客户端完成执行后，下一个客户端再开始执行。执行顺序由您的程序负责指定。

以下示例显示的是对同一存储区执行写操作的两个客户端。此外，还捕获了返回的错误，这是可选的。

程序段 1: Modbus 功能 1 - 读取 16 个输出映像位



程序段 2: Modbus 功能 2 - 读取 32 个输入映像位



### 12.5.2.5 MB\_CLIENT 示例 2: 通过不同的 TCP 连接发送多个请求

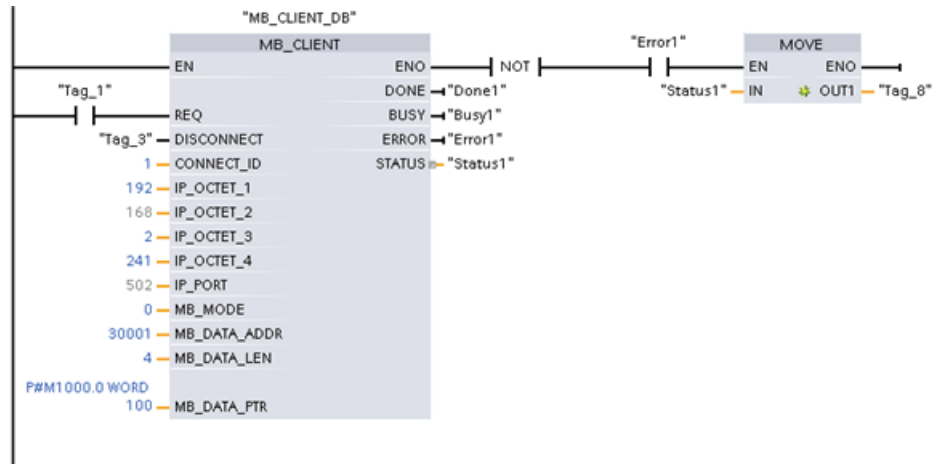
Modbus 客户端请求可通过不同连接来发送。为此，必须使用不同的背景数据块、IP 地址和连接 ID。

如要与同一 Modbus 服务器建立连接，端口号必须不同。如果与不同的服务器建立连接，则端口号方面没有限制。

以下示例显示的是对同一存储区执行写操作的两个客户端。此外，还捕获了返回的错误，这是可选的。

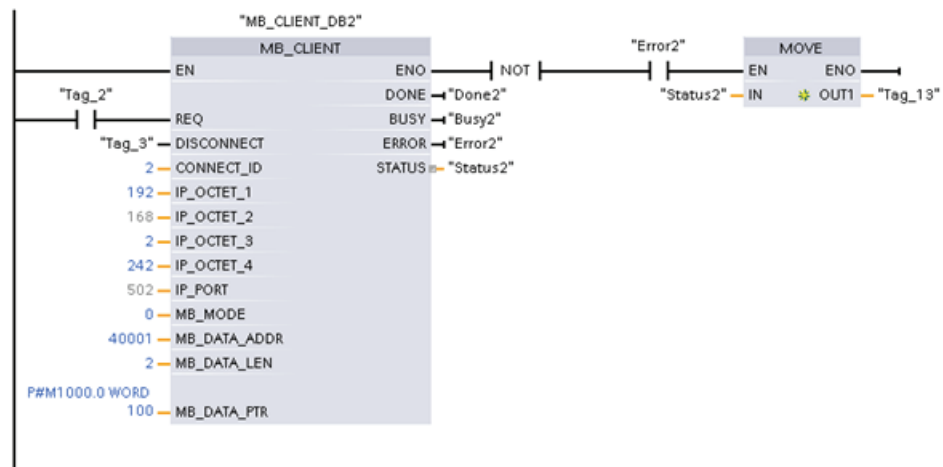
## 程序段 1:

Modbus 功能 4 - 读取 (S7-1200 存储器中的) 输入字



## 程序段 2:

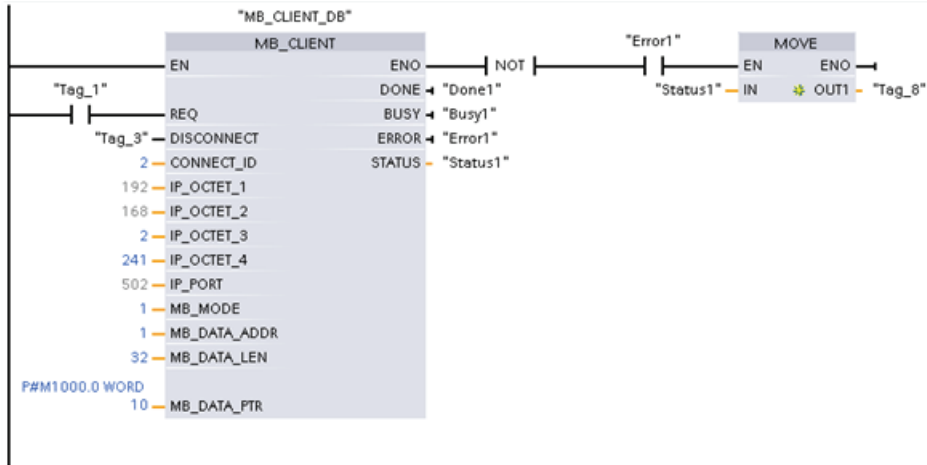
Modbus 功能 3 - 读取 (S7-1200 存储器中的) 保持寄存器字



12.5.2.6 MB\_CLIENT 示例 3: 输出映像写入请求

此示例显示了 Modbus 客户端请求写入 S7-1200 输出映像。

程序段 1: Modbus 功能 15 - 写入 S7-1200 输出映像位

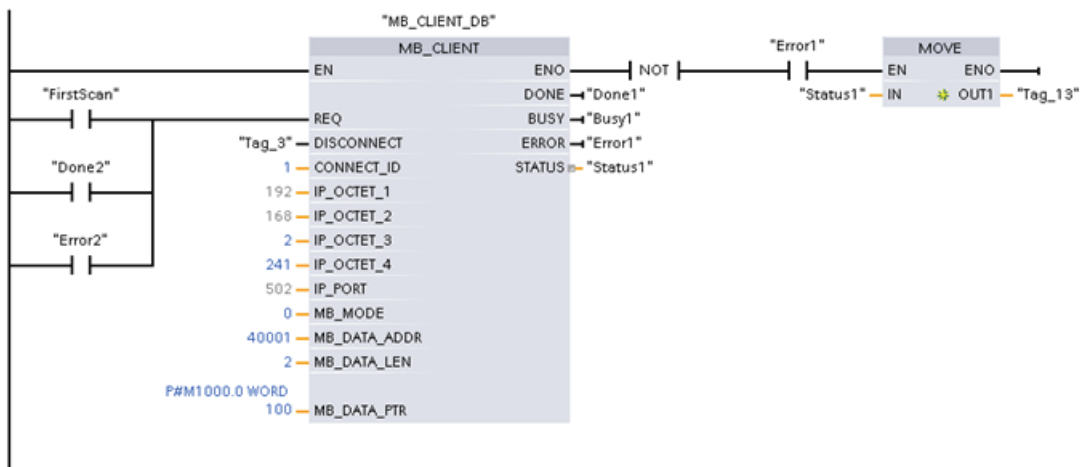


12.5.2.7 MB\_CLIENT 示例 4: 协调多个请求

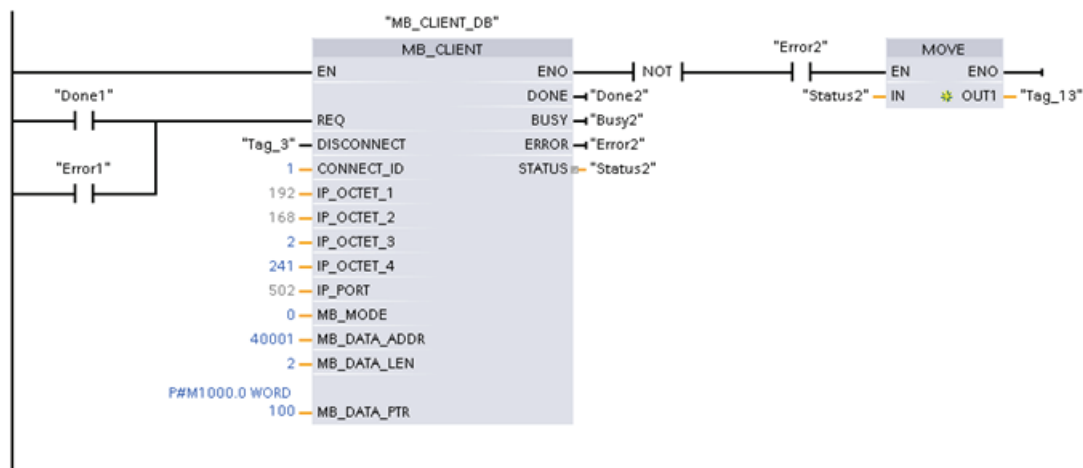
必须确保各个 Modbus TCP 请求都完成执行。此协调必须由程序提供。下面示例显示了首个和第二个客户端请求的输出如何用于协调执行。

以下示例显示的是对同一存储区执行写操作的两个客户端。此外，还捕获了返回的错误，这是可选的。

程序段 1: Modbus 功能 3 - 读取保持寄存器字



## 程序段 2: Modbus 功能 3 - 读取保持寄存器字



## 12.5.3

## Modbus RTU

在 STEP 7 中可使用两个版本的 Modbus RTU 指令:

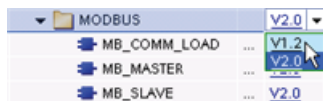
- 版本 1 最初在 STEP 7 Basic V10.5 中提供。
- 版本 2 在 STEP 7 Basic/Professional V11 中提供。版本 2 将参数 REQ 和 DONE 添加到 MB\_COMM\_LOAD。而且, MB\_MASTER 和 MB\_SLAVE 的 MB\_ADDR 参数现在还允许一个 UInt 值以进行扩展寻址。

可以从兼容性和移植便利性方面考虑, 选择将相应的指令版本插入用户程序中。

不要在同一 CPU 程序中同时使用 1.x 和 2.y 指令版本。用户程序的 Modbus 指令必须具有相同的主版本号 (1.x、2.y 或 V.z)。主版本组内的各个指令可具有不同的次版本号 (1.x)。



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 Modbus 指令的版本, 请从下拉列表中选择相应版本。可以选择一组指令或分别选择各个指令。

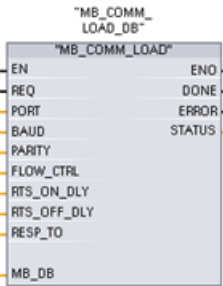
使用指令树将 Modbus 指令放入程序时, 将在项目树中创建新的 FB 实例。在项目树的“PLC\_x > 程序块 > 系统块 > 程序资源”(PLC\_x > Program blocks > System blocks > Program resources) 下可看到新的 FB 实例。

12.5 Modbus 通信

要确认程序中 Modbus 指令的版本，必须检查项目树的属性而不是程序编辑器中显示的框的属性。选择项目树的 Modbus FB 实例，单击右键，选择“属性”(Properties)，然后选择“信息”(Information) 页查看 Modbus 指令的版本号。

12.5.3.1 MB\_COMM\_LOAD

表格 12- 63 MB\_COMM\_LOAD 指令

LAD/FBD	SCL	说明
	<pre>"MB_COMM_LOAD_DB" (   REQ:=_bool_in_,   PORT:=_uint_in_,   BAUD:=_udint_in_,   PARITY:=_uint_in_,   FLOW_CTRL:=_uint_in_,   RTS_ON_DLY:=_uint_in_,   RTS_OFF_DLY:=_uint_in_,   RESP_TO:=_uint_in_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   MB_DB:=_fbtref_inout_);</pre>	<p>MB_COMM_LOAD 指令可组态用于 Modbus RTU 协议通信的 PtP 端口。Modbus 端口硬件选项：最多安装三个 CM (RS485 或 RS232)，及一个 CB (R4845)。将 MB_COMM_LOAD 指令放入程序时自动分配背景数据块。</p>

表格 12- 64 参数的数据类型

参数和类型	数据类型	说明
REQ IN	Bool	通过由低到高的（上升沿）信号启动操作。 (仅版本 2.0)
PORT IN	Port	安装并组态 CM 或 CB 通信设备之后，端口标识符将出现在 PORT 功能框连接的参数助手下拉列表中。分配的 CM 或 CB 端口值为设备配置属性“硬件标识符”。端口符号名称在 PLC 变量表的“系统常量”(System constants) 选项卡中分配。
BAUD IN	UDInt	波特率选择： 300、600、1200、2400、4800、9600、19200、38400、57600、76800、115200，其它所有值均无效
PARITY IN	UInt	奇偶校验选择： <ul style="list-style-type: none"> <li>• 0 – 无</li> <li>• 1 – 奇校验</li> <li>• 2 – 偶校验</li> </ul>



参数和类型		数据类型	说明
FLOW_CTRL	IN	UInt	流控制选择： <ul style="list-style-type: none"> <li>• 0 –（默认）无流控制</li> <li>• 1 – RTS 始终为 ON 的硬件流控制（不适用于 RS485 端口）</li> <li>• 2 - 带 RTS 切换的硬件流控制</li> </ul>
RTS_ON_DLY	IN	UInt	RTS 接通延时选择： <ul style="list-style-type: none"> <li>• 0 –（默认）从 RTS 激活一直到传送消息的第一个字符之前无延时</li> <li>• 1 到 65535 – 从 RTS 激活一直到传送消息的第一个字符之前以毫秒表示的延时（不适用于 RS485 端口）。不管 FLOW_CTRL 选择为何，都将应用 RTS 延时。</li> </ul>
RTS_OFF_DLY	IN	UInt	RTS 关断延时选择： <ul style="list-style-type: none"> <li>• 0 –（默认）从传送最后一个字符一直到 RTS 转入非活动状态之前无延时</li> <li>• 1 到 65535 – 从传送最后一个字符一直到 RTS 转入非活动状态之前以毫秒表示的延时（不适用于 RS485 端口）。不管 FLOW_CTRL 选择为何，都将应用 RTS 延时。</li> </ul>
RESP_TO	IN	UInt	响应超时： <p>MB_MASTER 允许用于从站响应的的时间（以毫秒为单位）。如果从站在此时间段内未响应，MB_MASTER 将重试请求，或者在发送指定次数的重试请求后终止请求并提示错误。</p> <p>5 ms 到 65535 ms（默认值 = 1000 ms）。</p>
MB_DB	IN	Variant	对 MB_MASTER 或 MB_SLAVE 指令所使用的背景数据块的引用。在用户的程序中放置 MB_SLAVE 或 MB_MASTER 后，该 DB 标识符将出现在 MB_DB 功能框连接的参数助手下拉列表中。
DONE	OUT	Bool	上一请求已完成且没有出错后，DONE 位将保持为 TRUE 一个扫描周期时间。（仅版本 2.0）
ERROR	OUT	Bool	上一请求因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个扫描周期内有效。
STATUS	OUT	Word	执行条件代码

可执行 MB\_COMM\_LOAD 来组态端口以使用 Modbus RTU 协议。为使用 Modbus RTU 协议组态端口后，该端口只能由 MB\_MASTER 或 MB\_SLAVE 指令使用。

对用于 Modbus 通信的每个通信端口，都必须执行一次 MB\_COMM\_LOAD 来组态。为要使用的每个端口分配一个唯一的 MB\_COMM\_LOAD 背景数据块。最多可在 CPU 中安装三个通信模块（RS232 或 RS485）和一个通信板（RS485）。从启动 OB 调用 MB\_COMM\_LOAD 并执行它一次，或使用第一个扫描系统标记（页 85）发起调用以执行它一次。只有在必须更改波特率或奇偶校验等通信参数时，才再次执行 MB\_COMM\_LOAD。

将 MB\_MASTER 或 MB\_SLAVE 指令放入用户程序中时，将为其分配背景数据块。指定 MB\_COMM\_LOAD 指令的 MB\_DB 参数时将引用该背景数据块。

### MB\_COMM\_LOAD 数据块变量

下表给出存储在 MB\_COMM\_LOAD 的背景数据块中的公共静态变量（可在用户程序中使用）。

表格 12-65 背景数据块中的静态变量

变量	数据类型	说明
ICHAR_GAP	Word	字符间隙延时。此参数以毫秒为单位指定，用于增加预期的接收字符间的时间量。与此参数对应的位时间个数加到 Modbus 默认的 35 个位时间（3.5 个字符时间）。
RETRIES	Word	主站在返回无响应错误代码 0x80C8 之前的重试次数。

表格 12-66 MB\_COMM\_LOAD 执行条件代码<sup>1</sup>

STATUS (W#16#)	说明
0000	无错误
8180	端口 ID 值无效（通信模块的端口/硬件标识符错误）
8181	波特率值无效
8182	奇偶校验值无效
8183	流控制值无效
8184	响应超时值无效（响应超时小于最小值 5 ms）
8185	MB_DB 参数不是 MB_MASTER 或 MB_SLAVE 指令的背景数据块。

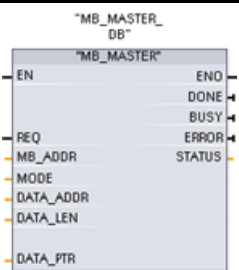
<sup>1</sup> 除了上述列出的 MB\_COMM\_LOAD 错误，还可能返回底层 PtP 通信指令的错误。

参见

点对点指令 (页 610)

## 12.5.3.2 MB\_MASTER

表格 12- 67 MB\_MASTER 指令

LAD/FBD	SCL	说明
	<pre>"MB_MASTER_DB" (   REQ:=_bool_in_,   MB_ADDR:=_uint_in_,   MODE:=_usint_in_,   DATA_ADDR:=_udint_in_,   DATA_LEN:=_uint_in_,   DONE=&gt;_bool_out_,   BUSY=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   DATA_PTR:=_variant_inout_);</pre>	<p>MB_MASTER 指令作为 Modbus 主站利用之前执行 MB_COMM_LOAD 指令组态的端口进行通信。将 MB_MASTER 指令放入程序时自动分配背景数据块。指定 MB_COMM_LOAD 指令的 MB_DB 参数时将使用该 MB_MASTER 背景数据块。</p>

表格 12- 68 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	0 = 无请求 1 = 请求将数据传送到 Modbus 从站
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Modbus RTU 站地址： 标准寻址范围（1 到 247） 扩展寻址范围（1 到 65535） 值 0 被保留用于将消息广播到所有 Modbus 从站。只有 Modbus 功能代码 05、06、15 和 16 是可用于广播的功能代码。
MODE	IN	USInt	模式选择：指定请求类型（读、写或诊断）。请参见下面的 Modbus 功能表了解详细信息。
DATA_ADDR	IN	UDInt	从站中的起始地址：指定要在 Modbus 从站中访问的数据的起始地址。请参见下面的 Modbus 功能表了解有效地址信息。
DATA_LEN	IN	UInt	数据长度：指定此请求中要访问的位数或字数。请参见下面的 Modbus 功能表了解有效长度信息。

参数和类型		数据类型	说明
DATA_PTR	IN	Variant	数据指针：指向要写入或读取的数据的 M 或 DB 地址（标准 DB 类型）。
DONE	OUT	Bool	上一请求已完成且没有出错后，DONE 位将保持为 TRUE 一个扫描周期时间。
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>• 0 – 无正在进行的 MB_MASTER 操作</li> <li>• 1 – MB_MASTER 操作正在进行</li> </ul>
ERROR	OUT	Bool	上一请求因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。STATUS 参数中的错误代码值仅在 ERROR = TRUE 的一个扫描周期内有效。
STATUS	OUT	Word	执行条件代码

### Modbus 主站通信规则

- 必须先执行 MB\_COMM\_LOAD 组态端口，然后 MB\_MASTER 指令才能与该端口通信。
- 如果要将某个端口用于初始化 Modbus 主站的请求，则 MB\_SLAVE 不应使用该端口。MB\_MASTER 执行的一个或多个实例可使用该端口，但是对于该端口，所有 MB\_MASTER 执行都必须使用同一个 MB\_MASTER 背景数据块。
- Modbus 指令不使用通信中断事件来控制通信过程。用户程序必须轮询 MB\_MASTER 指令以了解传送和接收的完成情况。
- 建议对于给定的端口，从程序循环 OB 中调用所有 MB\_MASTER 执行。Modbus 主站指令只能在一个程序循环或循环/延时执行等级执行。它们不能同时在两种执行优先级中执行。如果一个 Modbus 主站指令被另一个执行优先级更高的 Modbus 主站取代，将导致不正确的操作。Modbus 主站指令不能在启动、诊断或时间错误执行优先级执行。
- 主站指令启动传输后，必须连续执行已启用 EN 输入的该实例直到返回状态 DONE=1 或状态 ERROR=1 为止。在这两个事件其中之一发生前，一个特殊的 MB\_MASTER 实例被视为已激活。原始实例激活后，调用已启用 REQ 输入的其它任何实例都将导致错误。如果原始实例的连续执行过程停止，则请求状态保持激活一段时间，该时间由静态变量 Blocked\_Proc\_Timeout 指定。一旦超出该时间段，则下一个使用激活的 REQ 输入调用的主站指令成为激活实例。这可以防止单个 Modbus 主站指令独占或锁定对端口的访问。如果在由静态变量“Blocked\_Proc\_Timeout”指定的时间段内没有启用原始激活的实例，则下次执行此实例（未设置 REQ）时将清除激活状态。如果设置了 REQ，则此次执行将启动新的主站请求，如同其它实例未曾激活一样。

**REQ 参数**

0 = 无请求；1 = 请求将数据传送到 Modbus 从站

可使用电平或边沿触发的触点控制此输入。只要此输入启用，状态机便会启动，以确保在当前请求完成前不允许使用同一背景数据块的任何其它 MB\_MASTER 发出请求。在当前请求执行期间，将捕获所有其它输入状态并内部保存，直到接收到响应或检测到错误。

如果在当前请求完成前 REQ 输入 = 1，从而再次执行 MB\_MASTER 的同一实例，则不会进行任何后续传送。但是，如果当前请求已完成，因为 REQ 输入 = 1 而再次执行 MB\_MASTER 时，便会发出新请求。

**DATA\_ADDR 和 MODE 参数用于选择 Modbus 功能类型**

DATA\_ADDR（从站中的 Modbus 起始地址）：指定要在 Modbus 从站中访问的数据的起始地址。

MB\_MASTER 指令使用 MODE 输入而非功能代码输入。MODE 和 Modbus 地址一起确定实际 Modbus 消息中使用的功能代码。下表列出了 MODE 参数、Modbus 功能代码和 Modbus 地址范围之间的对应关系。

表格 12- 69 Modbus 功能

MODE	Modbus 功能	数据长度	操作和数据	Modbus 地址
0	01	1 到 2000 1 到 1992 <sup>1</sup>	读取输出位： 每个请求 1 到 1992 或 2000 个位	1 到 9999
0	02	1 到 2000 1 到 1992 <sup>1</sup>	读取输入位： 每个请求 1 到 1992 或 2000 个位	10001 到 19999
0	03	1 到 125 1 到 124 <sup>1</sup>	读取保持寄存器： 每个请求 1 到 124 或 125 个字	40001 到 49999 或 400001 到 465535
0	04	1 到 125 1 到 124 <sup>1</sup>	读取输入字： 每个请求 1 到 124 或 125 个字	30001 到 39999
1	05	1	写入一个输出位： 每个请求 1 个位	1 到 9999
1	06	1	写入一个保持寄存器： 每个请求 1 个字	40001 到 49999 或 400001 到 465535
1	15	2 到 1968 2 到 1960 <sup>1</sup>	写入多个输出位： 每个请求 2 到 1960 或 1968 个位	1 到 9999

MODE	Modbus 功能	数据长度	操作和数据	Modbus 地址
1	16	2 到 123 2 到 122 <sup>1</sup>	写入多个保持寄存器： 每个请求 2 到 122 或 123 个字	40001 到 49999 或 400001 到 465535
2	15	1 到 1968 2 到 1960 <sup>1</sup>	写入一个或多个输出位： 每个请求 1 到 1960 或 1968 个位	1 到 9999
2	16	1 到 123 1 到 122 <sup>1</sup>	写入一个或多个保持寄存器： 每个请求 1 到 122 或 123 个字	40001 到 49999 或 400001 到 465535
11	11	0	读取从站通信状态字和事件计数器。状态字指示忙闲情况（0 – 不忙，0xFFFF - 忙）。每成功完成一条消息，事件计数器的计数值递增。  对于该功能，MB_MASTER 的 DATA_ADDR 和 DATA_LEN 操作数都将被忽略。	
80	08	1	利用数据诊断代码 0x0000 检查从站状态（回送测试 - 从站回送请求） 每个请求 1 个字	
81	08	1	利用数据诊断代码 0x000A 重新设置从站事件计数器 每个请求 1 个字	
3 到 10、 12 到 79、 82 到 255			保留	

<sup>1</sup> 对于“扩展寻址”模式，根据功能所使用的数据类型，数据的最大长度将减小 1 个字节或 1 个字。

### DATA\_PTR 参数

DATA\_PTR 参数指向要写入或读取的 DB 或 M 地址。如果使用数据块，则必须创建一个全局数据块为读写 Modbus 从站提供数据存储位置。

#### 说明

#### DATA\_PTR 数据块类型必须允许直接寻址

该数据块必须允许直接（绝对）寻址和符号寻址。创建该数据块时，必须选择“标准”(Standard) 访问属性。

### DATA\_PTR 参数的数据块结构

- 这些数据类型对 Modbus 地址 30001 到 39999、40001 到 49999 和 400001 到 465536 的字读取有效，对 Modbus 地址 40001 到 49999 和 400001 到 465536 的字写入也有效。
  - WORD、UINT 或 INT 数据类型的标准数组
  - 指定的 WORD、UINT 或 INT 结构，其中每个元素都具有唯一的名称和 16 位数据类型。
  - 指定的复杂结构，其中每个元素都具有唯一的名称以及 16 或 32 位数据类型。
- 用于 Modbus 地址 00001 到 09999 的位读取和写入和 10001 到 19999 的位读取。
  - 布尔数据类型的标准数组。
  - 具有唯一名称的布尔变量的指定布尔结构。
- 尽管不是必需的，但还是建议每个 MB\_MASTER 指令都具有各自的单独存储区。此建议的原因在于，如果多个 MB\_MASTER 指令读取和写入同一个存储区，发生数据损坏的可能性会更大。
- 不要求 DATA\_PTR 数据区位于同一个全局数据块中。可创建一个具有多个区域的数据块供 Modbus 读取、一个数据块供 Modbus 写入或一个数据块用于各个从站。

### Modbus 主站数据块变量

下表给出存储在 MB\_MASTER 的背景数据块中的公共静态变量（可在用户程序中使用）。

表格 12-70 背景数据块中的静态变量

变量	数据类型	初始值	说明
Blocked_Proc_Timeout	Real	3.0	在 Modbus 主站实例受阻后，移除该激活的实例前需等待的时间（秒）。例如，当已发出主站请求，但程序在彻底完成该请求前停止调用该主站功能时，就会出现这种情况。时间值必须大于 0 且小于 55 秒，否则发生错误。默认值为 .5 秒。
Extended Addressing	Bool	False	组态单字节或双字节从站寻址。默认值 = 0。 (0= 单字节地址、1= 双字节地址)

用户程序可以将值写入 `Blocked_Proc_Timeout` 和 `Extended_Addressing` 变量，以控制 Modbus 主站操作。有关如何在程序编辑器中使用这些变量的示例以及有关 Modbus 扩展寻址的详细信息，请参见 `HR_Start_Offset` 和 `Extended_Addressing` 的 `MB_SLAVE` 主题说明 (页 701)。

## 条件代码

表格 12- 71 MB\_MASTER 执行条件代码（通信和组态错误）<sup>1</sup>

STATUS (W#16#)	说明
0000	无错误
80C8	从站超时。检查波特率、奇偶校验和从站的接线。
80D1	接收方发出了暂停主动传输的流控制请求并且在指定的等待时间内未重新激活该传输。 在硬件流控制期间，如果接收方在指定的等待时间内没有声明 CTS，也会产生该错误。
80D2	传送请求中止，因为没有从 DCE 收到任何 DSR 信号。
80E0	因接收缓冲区已满，消息被终止。
80E1	因出现奇偶校验错误，消息被终止。
80E2	因组帧错误，消息被终止。
80E3	因出现超限错误，消息被终止。
80E4	因指定长度超出总缓冲区大小，消息被终止。
8180	无效端口 ID 值或 <code>MB_COMM_LOAD</code> 指令出错
8186	Modbus 站地址无效
8188	指定给广播请求的模式无效
8189	数据地址值无效
818A	数据长度值无效
818B	指向本地数据源/目标的指针无效：大小不正确
818C	<code>DATA_PTR</code> 的指针无效或 <code>Blocked_Proc_Timeout</code> 无效：数据区必须是 DB（允许符号访问和直接访问）或 M 存储器。
8200	端口正忙于处理传送请求。



表格 12- 72 MB\_MASTER 执行条件代码 (Modbus 协议错误) <sup>1</sup>

STATUS (W#16#)	从站的响应代码	Modbus 协议错误
8380	-	CRC 错误
8381	01	不支持此功能代码
8382	03	数据长度错误
8383	02	数据地址错误或地址超出 DATA_PTR 区的有效范围
8384	大于 03	数据值错误
8385	03	不支持此数据诊断代码值 (功能代码 08)
8386	-	响应中的功能代码与请求中的代码不匹配。
8387	-	响应的从站错误
8388	-	从站对写请求的响应不正确。从站返回的写请求与主站实际发送的写请求不匹配。

<sup>1</sup> 除了上述列出的 MB\_MASTER 错误, 还可能返回底层 PtP 通信指令的错误。

参见

点对点指令 (页 610)

### 12.5.3.3 MB\_SLAVE

表格 12- 73 MB\_SLAVE 指令

LAD/FBD	SCL	说明
	<pre>"MB_SLAVE_DB" (   MB_ADDR:=_uint_in_,   NDR=&gt;_bool_out_,   DR=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,   MB_HOLD_REG:=_variant_inout_);</pre>	<p>MB_SLAVE 指令允许用户程序作为 Modbus 从站通过 CM (RS485 或 RS232) 和 CB (RS485) 上的 PtP 端口进行通信。远程 Modbus RTU 主站发出请求时, 用户程序会通过执行 MB_SLAVE 进行响应。STEP 7 在插入指令时自动创建背景数据块。在为 MB_COMM_LOAD 指令指定 MB_DB 参数时使用此 MB_SLAVE_DB 名称。</p>

## 12.5 Modbus 通信

表格 12- 74 参数的数据类型

参数和类型		数据类型	说明
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Modbus 从站的站地址： 标准寻址范围（1 到 247） 扩展寻址范围（0 到 65535）
MB_HOLD_REG	IN	Variant	指向 Modbus 保持寄存器 DB 的指针： Modbus 保持寄存器可以是 M 存储器或数据块。
NDR	OUT	Bool	新数据就绪： <ul style="list-style-type: none"> <li>• 0 – 无新数据</li> <li>• 1 – 表示 Modbus 主站已写入新数据</li> </ul>
DR	OUT	Bool	数据读取： <ul style="list-style-type: none"> <li>• 0 – 无数据读取</li> <li>• 1 – 表示 Modbus 主站已读取数据</li> </ul>
ERROR	OUT	Bool	上一请求因错误而终止后，ERROR 位将保持为 TRUE 一个扫描周期时间。如果执行因错误而终止，则 STATUS 参数的错误代码值仅在 ERROR = TRUE 的一个扫描周期内有效。
STATUS	OUT	Word	执行错误代码

Modbus 通信功能代码（1、2、4、5 和 15）可以在 CPU 的输入过程映像及输出过程映像中直接读写位和字。对于这些功能代码，MB\_HOLD\_REG 参数必须定义为大于一个字节的 datatype。下表给出了 Modbus 地址与 CPU 过程映像的映射示例。

表格 12- 75 Modbus 地址到过程映像的映射

Modbus 功能						S7-1200	
代码	功能	数据区	地址范围			数据区	CPU 地址
01	读位	输出	1	到	8192	输出过程映像	Q0.0 到 Q1023.7
02	读位	输入	10001	到	18192	输入过程映像	I0.0 到 I1023.7
04	读字	输入	30001	到	30512	输入过程映像	IW0 到 IW1022
05	写位	输出	1	到	8192	输出过程映像	Q0.0 到 Q1023.7
15	写位	输出	1	到	8192	输出过程映像	Q0.0 到 Q1023.7

Modbus 通信功能代码（3、6、16）使用 Modbus 保持寄存器，该寄存器可以是 M 存储器地址范围或数据块。保持寄存器的类型由 MB\_SLAVE 指令的 MB\_HOLD\_REG 参数指定。

#### 说明

#### MB\_HOLD\_REG 数据块类型

Modbus 保持寄存器数据块必须允许直接（绝对）寻址和符号寻址。创建该数据块时，必须选择“标准”(Standard) 访问属性。

下表给出了 Modbus 地址到保持寄存器的映射示例，这种映射用于 Modbus 功能代码 03（读取字）、06（写入字）和 16（读取字）。DB 地址的实际上限取决于每种 CPU 型号的最大工作存储器限值和 M 存储器限值。

表格 12- 76 Modbus 地址到 CPU 存储器的映射

Modbus 主站地址	MB_HOLD_REG 参数示例				
	MW100	DB10.DBw0	MW120	DB10.DBW50	"Recipe".ingredient
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Recipe".ingredient[5]

表格 12- 77 诊断功能

S7-1200 MB_SLAVE Modbus 诊断功能		
代码	子功能	说明
08	0000H	返回查询数据回送测试： MB_SLAVE 将向 Modbus 主站回送接收到的数据字。
08	000AH	清除通信事件计数器： MB_SLAVE 将清除用于 Modbus 功能 11 的通信事件计数器。
11		获取通信事件计数器： MB_SLAVE 使用内部通信事件计数器来记录发送到 Modbus 从站的 Modbus 成功读取和写入请求次数。该计数器不会因功能 8、功能 11 或广播请求而增加。同样也不会因任何导致通信错误（例如，奇偶校验错误或 CRC 错误）的请求而增加。

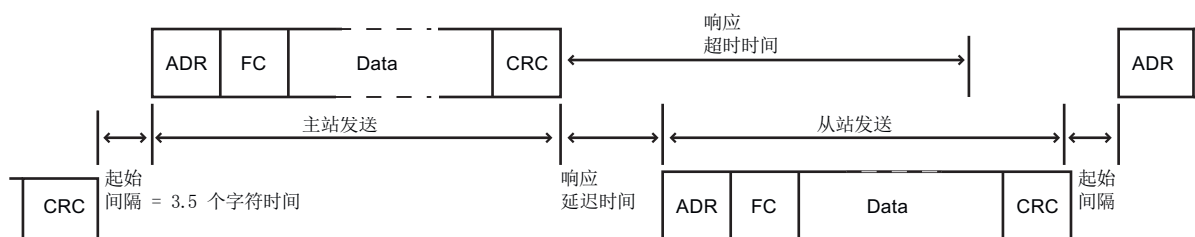
MB\_SLAVE 指令支持来自任何 Modbus 主站的广播写请求，只要该请求是用于访问有效地址的请求即可。对于广播不支持的功能代码，MB\_SLAVE 将生成错误代码 0x8188。

### Modbus 从站通信规则

- 必须先执行 MB\_COMM\_LOAD 组态端口，然后 MB\_SLAVE 指令才能通过该端口通信。
- 如果某个端口作为从站响应 Modbus 主站，则请勿使用 MB\_MASTER 指令对该端口进行编程。
- 对于给定端口，只能使用一个 MB\_SLAVE 实例，否则将出现不确定的行为。
- Modbus 指令不使用通信中断事件来控制通信过程。用户程序必须通过轮询 MB\_SLAVE 指令以了解传送和接收的完成情况来控制通信过程。
- MB\_SLAVE 指令必须以一定的速率定期执行，以便能够及时响应来自 Modbus 主站的进入请求。建议每次扫描时都从程序循环 OB 执行 MB\_SLAVE。也可以从循环中断 OB 执行 MB\_SLAVE，但并不建议这么做，因为中断例程的延时过长可能会暂时阻止其它中断例程的执行。

### Modbus 定时信号

必须周期性执行 MB\_SLAVE，才能接收来自 Modbus 主站的每个请求并随之按要求响应。MB\_SLAVE 的执行频率取决于 Modbus 主站的响应超时时间。下图对此进行了说明。



响应超时时间 RESP\_TO 是 Modbus 主站等待 Modbus 从站开始响应的的时间。该时间段不是由 Modbus 协议定义的，而是属于每个 Modbus 主站的一个参数。必须基于用户 Modbus 主站的具体参数确定 MB\_SLAVE 的执行频率（相邻两次执行之间的时间）。在 Modbus 主站的响应超时时间内至少应执行两次 MB\_SLAVE。

## Modbus 从站变量

下表给出了存储在 MB\_SLAVE 背景数据块（可在用户程序中使用）中的公共静态变量。

表格 12- 78 Modbus 从站变量

变量	数据类型	说明
HR_Start_Offset	Word	指定 Modbus 保持寄存器的起始地址（默认值 = 0）
Extended_Addressing	Bool	组态单字节或双字节从站寻址 (0= 单字节地址、1= 双字节地址、默认 = 0)
Request_Count	Word	该从站接收到的所有请求的数量
Slave_Message_Count	Word	该特定从站接收到的请求的数量
Bad_CRC_Count	Word	接收到的具有 CRC 错误的请求的数量
Broadcast_Count	Word	接收到的广播请求的数量
Exception_Count	Word	需要返回例外的 Modbus 特定错误数
Success_Count	Word	该特定从站接收到的没有协议错误的请求数量

程序可以将值写入 HR\_Start\_Offset 和 Extended\_Addressing 变量以控制 Modbus 从站操作。可读取其它变量以监视 Modbus 的状态。

## HR\_Start\_Offset

Modbus 保持寄存器的地址以 40001 或 400001 开始。这些地址与保持寄存器的 PLC 存储器起始地址对应。不过，可以组态“HR\_Start\_Offset”变量，将 Modbus 保持寄存器的起始地址定义为除 40001 或 400001 之外的其它值。

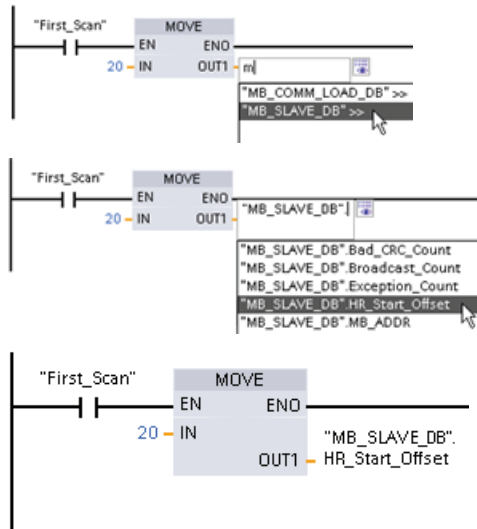
例如，如果保持寄存器被组态为起始于 MW100 并且长度为 100 个字。偏移量 20 可指定保持寄存器的起始地址为 40021 而不是 40001。低于 40021 和高于 400119 的任何地址都将导致寻址错误。

表格 12- 79 Modbus 保持寄存器寻址示例

HR_Start_Offset	地址	最小值	最大值
0	Modbus 地址（字）	40001	40099
	S7-1200 地址	MW100	MW298
20	Modbus 地址（字）	40021	40119
	S7-1200 地址	MW100	MW298

HR\_Start\_Offset 是一个字值，用于指定 Modbus 保持寄存器的起始地址，存储在 MB\_SLAVE 背景数据块中。将 MB\_SLAVE 放入程序后，可利用参数助手下拉列表设置该公共静态变量值。

例如，将 MB\_SLAVE 放入 LAD 程序段后，可以切换到先前的程序段，分配 HR\_Start\_Offset 值。该值必须在执行 MB\_SLAVE 前分配。



使用默认 DB 名称输入 Modbus 从站变量：

1. 将光标放在参数字段中，然后输入 m 字符。
2. 从下拉列表中选择“MB\_SLAVE\_DB”。
3. 将光标放在 DB 名称的右侧（引号字符的后面），然后输入句点字符。
4. 从下拉列表中选择 “MB\_SLAVE\_DB.HR\_Start\_Offset”。

### Extended Addressing

Extended Addressing 变量的访问方式与上述的 HR\_Start\_Offset 参考相似，只是 Extended Addressing 变量是布尔值。布尔值必须通过输出线圈（不是移动框）写入。

Modbus 从站寻址可组态为单字节（Modbus 标准方式）或双字节。扩展寻址用于对单一网络内超过 247 台设备进行寻址。选择扩展寻址后，最多可以对 64000 个地址进行寻址。下面以 Modbus 功能 1 的帧为例进行显示。

表格 12- 80 单字节从站地址（字节 0）

功能 1	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	
请求	从站地址	F 代码	起始地址		线圈长度		
有效响应	从站地址	F 代码	长度	线圈数据			
错误响应	从站地址	0x81	E 代码				

表格 12- 81 双字节从站地址（字节 0 和字节 1）

	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6
请求	从站地址		F 代码	起始地址		线圈长度	
有效响应	从站地址		F 代码	长度	线圈数据		
错误响应	从站地址		0x81	E 代码			

### 条件代码

表格 12- 82 MB\_SLAVE 执行条件代码（通信和组态错误）<sup>1</sup>

STATUS (W#16#)	说明
80D1	接收方发出了暂停主动传输的流控制请求并且在指定的等待时间内未重新激活该传输。 在硬件流控制期间，如果接收方在指定的等待时间内没有声明 CTS，也会产生该错误。
80D2	传送请求中止，因为没有从 DCE 收到任何 DSR 信号。
80E0	因接收缓冲区已满，消息被终止。
80E1	因出现奇偶校验错误，消息被终止。
80E2	因组帧错误，消息被终止。
80E3	因出现超限错误，消息被终止。
80E4	因指定长度超出总缓冲区大小，消息被终止。
8180	无效端口 ID 值或 MB_COMM_LOAD 指令出错
8186	Modbus 站地址无效
8187	指向 MB_HOLD_REG DB 的指针无效：区域太小
818C	指向 M 存储器或 DB（DB 区域必须允许符号地址和直接地址）的 MB_HOLD_REG 指针无效

表格 12- 83 MB\_SLAVE 执行条件代码（Modbus 协议错误）<sup>1</sup>

STATUS (W#16#)	从站的响应代码	Modbus 协议错误
8380	无响应	CRC 错误
8381	01	不支持功能代码或在广播内不支持

12.5 Modbus 通信

STATUS (W#16#)	从站的响应代码	Modbus 协议错误
8382	03	数据长度错误
8383	02	数据地址错误或地址超出 DATA_PTR 区的有效范围
8384	03	数据值错误
8385	03	不支持此数据诊断代码值（功能代码 08）

<sup>1</sup> 除了上述列出的 MB\_SLAVE 错误，还可能返回底层 PtP 通信指令的错误。

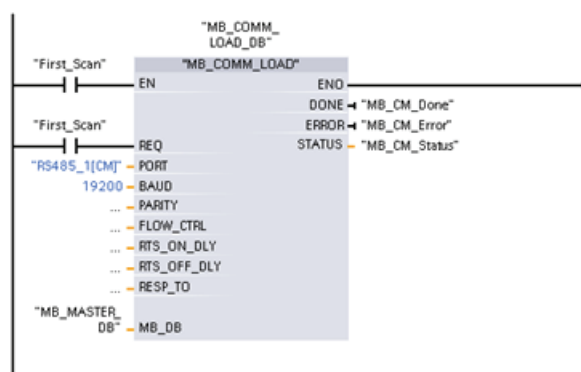
参见

点对点指令 (页 610)

12.5.3.4 Modbus RTU 主站示例程序

启动期间通过第一个扫描标志启用 MB\_COMM\_LOAD。通过此方式执行 MB\_COMM\_LOAD 时，必须保证串口组态在运行时不会更改。

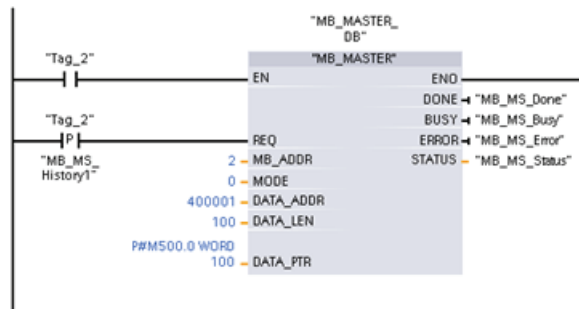
程序段 1 仅在第一次扫描期间初始化一次 RS485 模块参数。



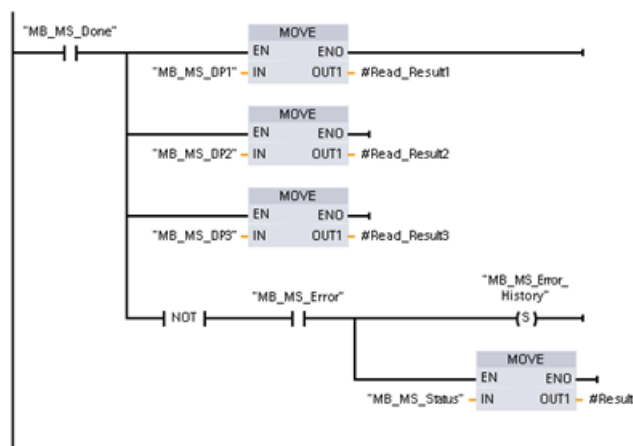
在程序循环 OB 中使用一个 MB\_MASTER 指令，以与单个从站进行通信。要与其它从站通信，可在程序循环 OB 中使用另外的 MB\_MASTER 指令，也可以重新使用一个 MB\_MASTER FB。

网络 2 从从站保持寄存器读取 100 个字。

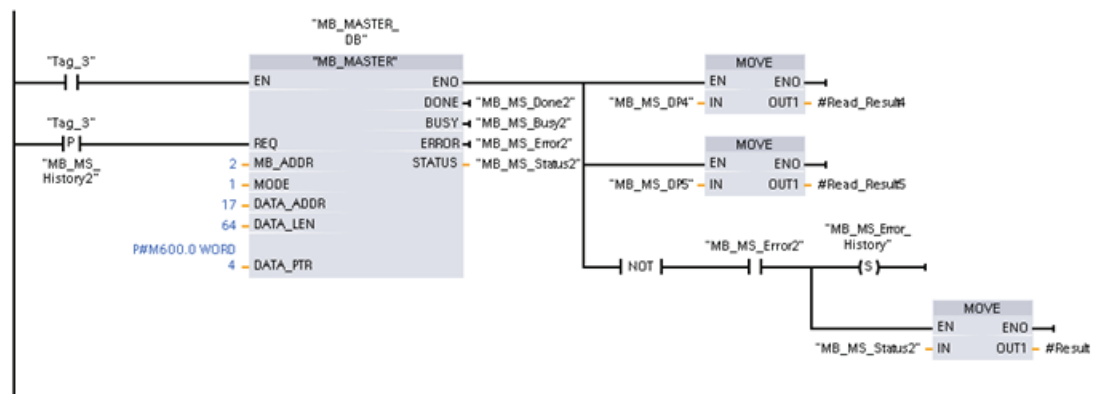




网络 3 这是一个可选网络，仅显示读操作完成后前 3 个字的值。



网络 4 将 64 个位写入起始于从站地址 Q2.0 的输出映像寄存器。



### 12.5.3.5

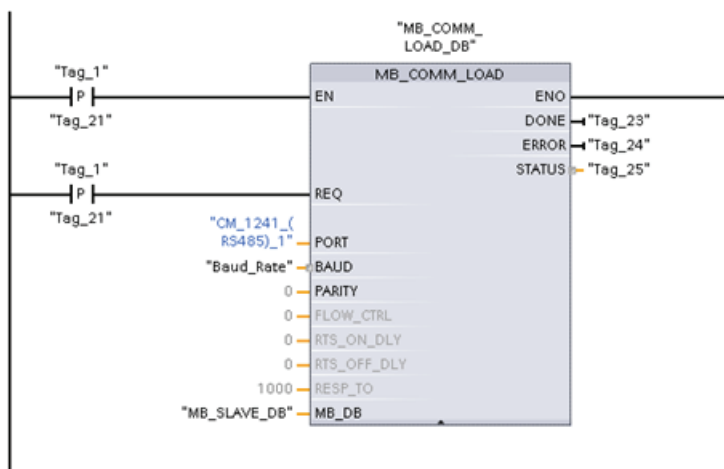
#### Modbus RTU 从站示例程序

每次启用“Tag\_1”启用时，初始化下面显示的 MB\_COMM\_LOAD。

通过此方式执行 MB\_COMM\_LOAD 时，必须保证串口组态在运行时会根据 HMI 配置进行更改。

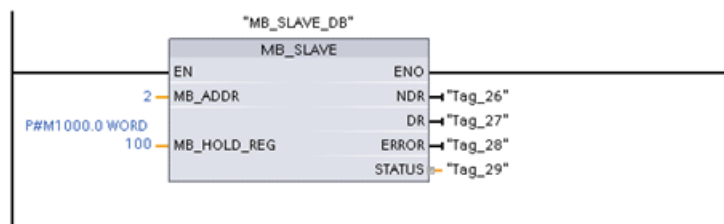
程序段 1 每次 HMI 设备更改 RS485 模块参数时，都会初始化该参数。

12.6 采用 CP 1242-7 的遥控和远程服务



下面显示的 MB\_SLAVE 置于每 10 ms 执行一次的循环 OB 中。尽管这样不会使从站的绝对响应速度达到最快，但却可使短消息（在请求中占 20 字节或更低）达到 9600 波特的良好性能。

程序段 2 每次扫描期间检查 Modbus 主站请求。Modbus 保持寄存器被组态为 100 个字（从 MW1000 开始）。



## 12.6 采用 CP 1242-7 的遥控和远程服务

### 12.6.1 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 适用于世界各地的工业行业，支持以下频段：

- 850 MHz
- 900 MHz
- 1800 MHz
- 1900 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 网络供应商必须给 CP 分配固定 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

### 12.6.2 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。要使用 GPRS 进行数据传送，需在主站中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 服务器功能与上层中央控制系统通信。

- S7-1200 站之间通过 GSM 网络进行通信

可以用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

在该组态情况下，互相通信的 S7-1200 站与遥控服务器间的永久安全连接在主站中建立。各个站通过遥控服务器通信。CP 1242-7 在“Telecontrol”模式下运行。

- 站间直接通信

若要各个站直接进行通信而不经由主站迂回通信，可以使用具有固定 IP 地址的 SIM 卡使得站可以直接互相寻址。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 直连”模式下运行。

## 通过 GPRS 实现 TeleService

可以通过 GSM 网络和 Internet 在装有 STEP 7 的工程师站与具有 CP 1242-7 的远程 S7-1200 站间建立 TeleService 连接。连接从工程师站通过用于中转帧和建立授权的遥控服务器或 TeleService 网关向外延伸。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

### 12.6.3 CP 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，由遥控服务器传送日时钟。CP 使用该时间来设置其自身的时间。
- 在“GPRS 直接”模式下，CP 可通过 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
- 由于可以选择连接到备用遥控服务器而增强的可用性
- 优化的数据量（临时连接）

除与远程控制服务器建立永久连接的方式外，还可以在 STEP 7 中使用与远程控制服务器建立的临时连接来组态 CP。这种情况下，仅在需要时建立与遥控服务器的连接。

- 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。

#### 组态和模块替换

要组态该模块，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，请在相应站的用户程序中使用遥控指令。

CP 1242-7 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 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。

## 12.6 采用 CP 1242-7 的遥控和远程服务

## 电气连接

- CP 1242-7 的电源  
CP 具有一个用来外接 24 VDC 电源的独立连接器。
- 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。

## 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：

42330276 (<http://support.automation.siemens.com/WW/view/zh/42330276>)

## 12.6.4 附件

## ANT794-4MR GSM/GPRS 天线

以下天线可以在 GSM/GPRS 网络中使用，并且可以安装在室内和室外：

- 四频天线 ANT794-4MR



图 12-1 ANT794-4MR GSM/GPRS 天线

简称	订货号	说明
ANT794-4MR	6NH9 860-1AA00	四频天线（900、1800/1900 MHz，UMTS）；防风雨，可在室内和室外使用；天线永久连接 5 m 连接电缆；SMA 连接器（包括安装支架、螺钉、墙用插头）

- 平面天线 ANT794-3M

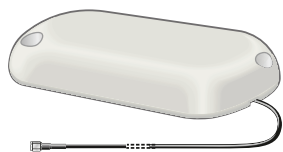


图 12-2 平面天线 ANT794-3M

简称	订货号	说明
ANT794-3M	6NH9 870-1AA00	平面天线（900、1800/1900 MHz）；防风雨，可在室内和室外使用；天线永久连接 1.2 m 连接电缆；SMA 连接器（包括粘板），可使用螺钉安装

天线必须单独订购。

### 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>)

### 12.6.5 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

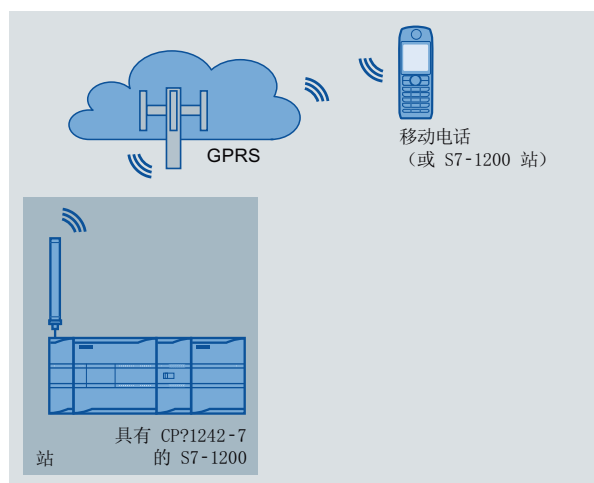


图 12-3 通过 SMS 从 S7-1200 站发送消息

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

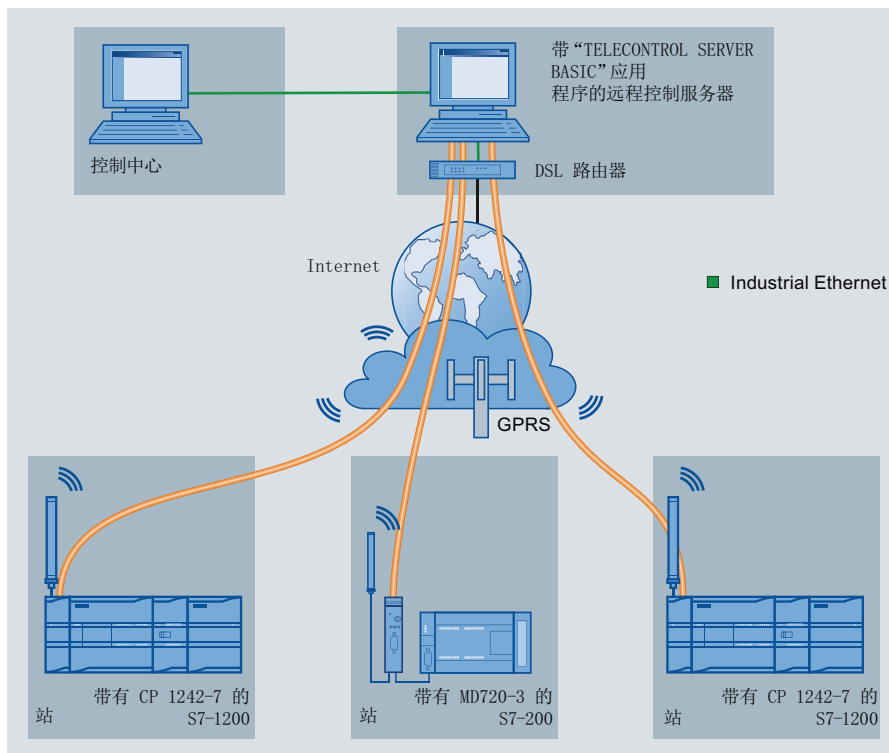


图 12-4 S7-1200 站与控制中心之间的通信

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。主站中的遥控服务器已安装“TELECONTROL SERVER BASIC”(TCSB) 应用程序。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，工作站通过 GSM 网络和 Internet 将现场数据发送到主站中的遥控服务器。遥控服务器用于监控远程站。

- 工作站和装有 OPC 客户机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。通过其集成 OPC 服务器，遥控服务器会与控制室的 OPC 客户机交换数据。

OPC 客户机和遥控服务器可位于单台计算机上，例如当 TCSB 安装在装有 WinCC 的控制中心计算机上时。

- 通过控制中心进行站间通信

对于配有 CP 1242-7 的 S7 站，可以进行站间通信。

遥控服务器会将发送站的消息转发给接收站，从而实现站间通信。

## 站间直接通信

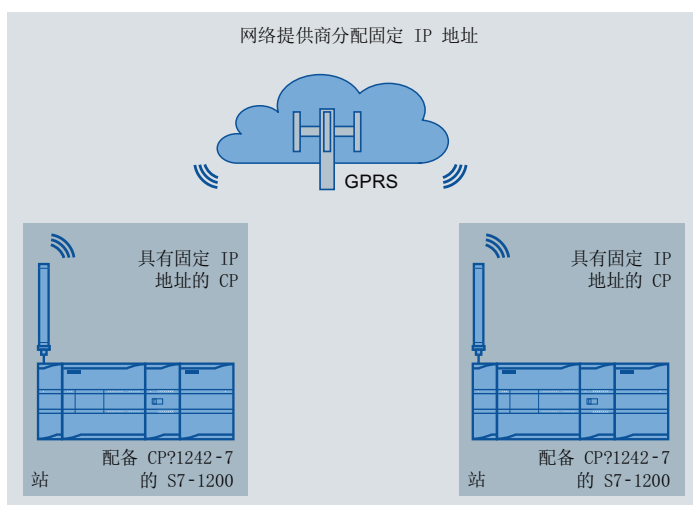


图 12-5 两个 S7-1200 站间直接通信

在该组态情况下，两个 SIMATIC S7-1200 站使用 CP 1242-7 通过 GSM 网络彼此直接通信。各 CP 1242-7 具有固定 IP 地址。GSM 网络供应商的相关服务必须支持该操作。

## GPRS TeleService

在 GPRS TeleService 中，安装了 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。

## 12.6 采用 CP 1242-7 的遥控和远程服务

由于防火墙因外部连接请求通常处于关闭状态，因此需要在远程站与工程师站之间设交换站。交换站可以是遥控服务器或 **TeleService** 网关（如果组态中没有遥控服务器）。

### 带遥控服务器的 **TeleService**

通过遥控服务器进行连接。

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

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

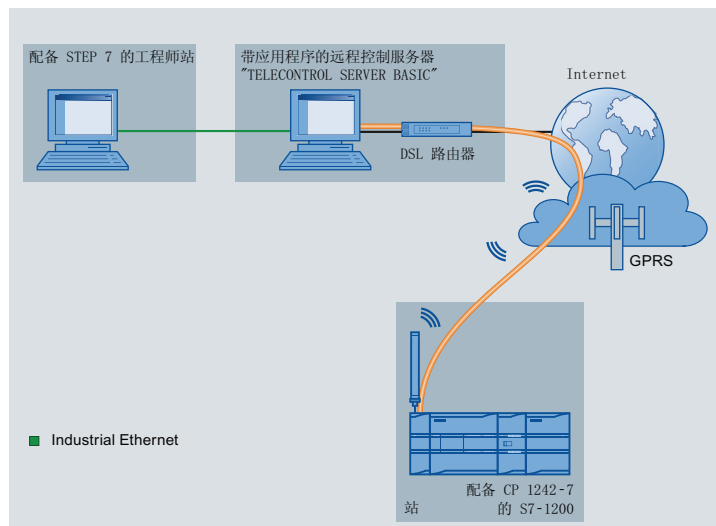


图 12-6 带遥控服务器的组态中的 GPRS TeleService

### 不带遥控服务器的 TeleService

通过 TeleService 网关进行连接。

工程师站与 TeleService 网关之间的连接可通过 LAN 在本地实现，也可以通过 Internet 实现。

12.6 采用 CP 1242-7 的遥控和远程服务

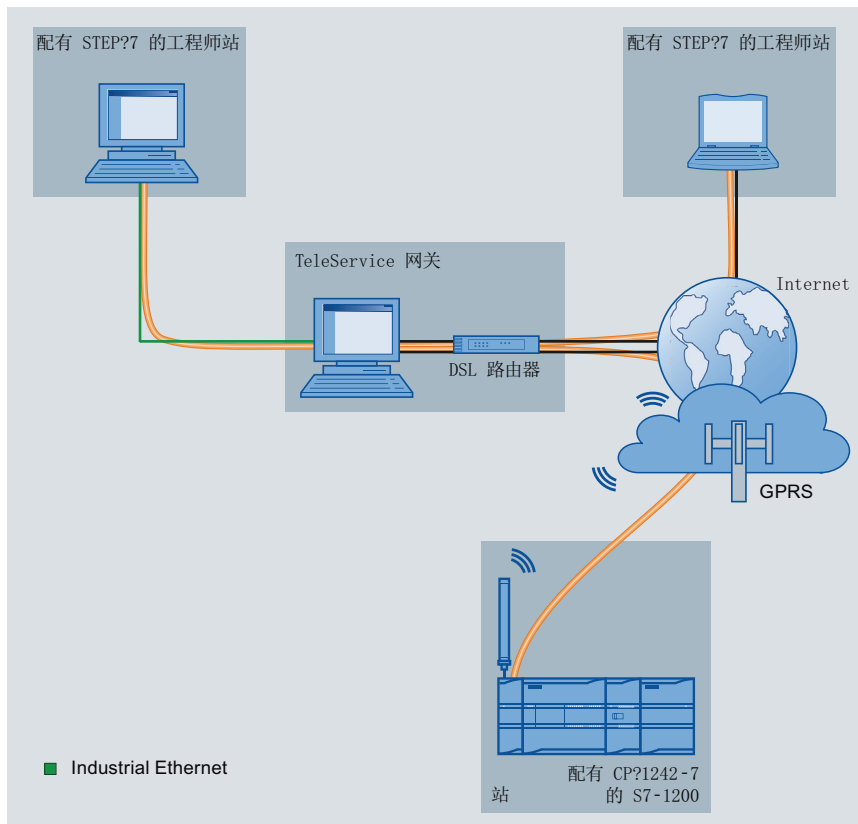


图 12-7 带 TeleService 网关的组态中的 GPRS TeleService

## 远程服务通信（SMTP 电子邮件）

## 13.1 TM\_Mail 发送电子邮件指令

表格 13-1 TM\_MAIL 指令

LAD/FBD	SCL	说明
	<pre> "TM_MAIL_DB" (   REQ:=_bool_in_,   ID:=_int_in_,   TO_S:=_string_in_,   CC:=_string_in_,   SUBJECT:=_string_in_,   TEXT:=_string_in_,   ATTACHMENT:=_variant_in_,   BUSY=&gt;_bool_out_,   DONE=&gt;_bool_out_,   ERROR=&gt;_bool_out_,   STATUS=&gt;_word_out_,); </pre>	<p>TM_MAIL 指令通过 CPU 工业以太网连接使用 TCP/IP 上的 SMTP（Simple Mail Transfer Protocol, 简单邮件传输协议）发送电子邮件消息。其中基于以太网的 Internet 连接性不可用，可选的远程服务适配器可用于与电话陆线的连接。TM_MAIL 会异步执行，并且该作业会持续多次 TM_MAIL 调用。调用 TM_MAIL 时，必须分配背景数据块。绝不可设置该背景数据块的保持性属性。如此可以确保背景数据块在 CPU 由 STOP 模式切换到 RUN 模式时初始化，以及可以触发新的 TM_MAIL 操作。</p>

1 STEP 7 会在插入指令时自动创建背景 DB。

当输入参数 REQ 出现上升沿（从 0 变为 1）时，开始发送电子邮件。下表给出了 BUSY、DONE 和 ERROR 之间的关系。可在连续调用期间评估这些参数来监视 TM\_MAIL 执行的进度和检查完成情况。

输出参数 BUSY 的状态从 1 变为 0 时，输出参数 DONE、ERROR、STATUS 和 SFC\_STATUS 仅在一个周期内有效。程序逻辑必须保存临时输出状态值，以便能检测到后续程序执行周期中的状态变化。

表格 13-2 Done、Busy 和 Error 参数之间的交互作用

DONE	BUSY	ERROR	说明
不相关	1	不相关	正在处理作业。
1	0	0	作业已成功完成。

13.1 TM\_Mail 发送电子邮件指令

DONE	BUSY	ERROR	说明
0	0	1	作业因出错而终止。有关错误原因的信息，请参见 STATUS 参数。
0	0	0	没有作业正在处理

如果 CPU 在 TM\_MAIL 激活期间切换到 STOP 模式，则将终止与电子邮件服务器之间的通信连接。如果通过工业以太网总线进行 CPU 通信时出现问题，那么将丢失与电子邮件服务器的通信连接。发生这些情况时，将暂停发送过程，同时接收方也收不到电子邮件。

**小心**

**修改用户程序**

仅在下列情况下更改用户程序中直接影响 TM\_MAIL 调用的部分：

- CPU 处于 STOP 模式
- 未发送任何电子邮件 (REQ 和 BUSY = 0)

具体而言是程序块的删除和替换、对 TM\_MAIL 的调用或者对 TM\_MAIL 背景数据块的调用。

如果未能保持已链接的程序块，则 TPC/IP 通信功能将进入不确定状态。 传送修改后的程序块之后，必须执行 CPU 重启（热启动）或冷启动。

**数据一致性**

在启动操作时会读取输入参数 ADDR\_MAIL\_SERVER。 只有在当前操作完成并且启动新的 TM\_MAIL 操作后，新值才会生效。

相反，参数 WATCH\_DOG\_TIME、TO\_S、CC、FROM、SUBJECT、TEXT、ATTACHMENT、USERNAME 和 PASSWORD 将在执行 TM\_MAIL 时被读取，并且仅在完成作业 (BUSY = 0) 后才可更改。

**拨号连接：组态 TS 适配器的 IE 参数。**

必须组态离开调用的远程服务适配器 IE 参数，以便与 Internet 服务提供商的拨号服务器相连接。如果设置了调用的“按需”属性，则仅在发送电子邮件时建立连接。对于模拟调制解调器连接，连接过程需要更多的时间（大约多出一分钟）。必须将额外的时间包括到 WATCH\_DOG\_TIME 值中。

表格 13-3 参数的数据类型

参数和类型		数据类型	说明
REQ	IN	Bool	通过由低到高的（上升沿）信号启动操作。
ID	IN	Int	连接标识符：请参见指令 TCON、TDISCON、TSEND 和 TRCV 的 ID 参数。 必须使用未在用户程序中用于该指令的任何其它实例的编号。
TO_S	IN	String	收件人地址：最大长度为 240 个字符的 STRING 数据
CC	IN	String	抄送收件人地址（可选）：最大长度为 240 个字符的 STRING 数据
SUBJECT	IN	String	电子邮件的主题名：最大长度为 240 个字符的 STRING 数据。
TEXT	IN	String	电子邮件的文本消息（可选）：最大长度为 240 个字符的 STRING 数据。 如果此参数是空字符串，则发送的电子邮件将不含任何消息文本。
ATTACHMENT	IN	Variant	指向电子邮件附件数据的指针：最大长度为 65534 字节的字节、字或双字数据。 如果未分配任何值，则发送的电子邮件不含附件。
DONE	OUT	Bool	<ul style="list-style-type: none"> <li>0 - 作业尚未启动或仍在执行。</li> <li>1 - 作业已执行，未出现错误。</li> </ul>
BUSY	OUT	Bool	<ul style="list-style-type: none"> <li>0 - 无操作正在进行</li> <li>1 - 操作正在进行</li> </ul>
ERROR	OUT	Bool	上一请求因错误而终止后，ERROR 位将在一个扫描周期的时间内保持为 1。STATUS 输出中的错误代码值仅在 ERROR = 1 的一个扫描周期内有效。
STATUS	OUT	Word	TM_MAIL 指令的返回值或错误信息。

13.1 TM\_Mail 发送电子邮件指令

参数和类型		数据类型	说明
ADDR_MAIL_SERVER	1 Static	DWord	<p>邮件服务器的 IP 地址： 必须将每个 IP 地址片段分配为两个 4 位十六进制字符组成的 8 位位组。 如果 IP 地址片段 = 等于十六进制值 A 的十进制值 10， 则必须为该 8 位位组输入“0A”。</p> <p>例如： IP 地址 = 192.168.0.10</p> <p>ADDR_MAIL_SERVER = DW#16#C0A8000A， 其中：</p> <ul style="list-style-type: none"> <li>• 192 = 16#C0,</li> <li>• 168 = 16#A8</li> <li>• 0 = 16#00</li> <li>• 10 = 16#0A</li> </ul>
WATCH_DOG_TIME	1 Static	Time	<p>通过 TM_MAIL 建立服务器连接所允许的最长时间。 如果超出该时间， TM_MAIL 结束执行并报告错误。</p> <p>在 TM_MAIL 结束并报告错误之前的实际时间延时可能超过 WATCH_DOG_TIME， 这是因为断开操作需要更多的时间。</p> <p>开始时， 应设置 2 分钟时间。 对于 ISDN 电话连接， 该时间可以短很多。</p>
USERNAME	1 Static	String	邮件帐户的用户名： 最大长度为 180 个字符的 STRING 数据。
PASSWORD	1 Static	String	邮件服务器密码： 最大长度为 180 个字符的 STRING 数据。
FROM	1 Static	String	发送方地址： 最大长度为 240 个字符的 STRING
SFC_STATUS	1 Static	Word	被调用通信块的执行条件代码

1 每次调用 TM\_MAIL 时都不会修改这些参数的值。 值分配在 TM\_MAIL 实例数据块中， 并且在首次调用 TM\_MAIL 时， 并不只被引用一次，

SMTP 验证

TM\_MAIL 支持 SMTP AUTH LOGIN 验证方法。 有关该验证方法的信息， 请参见邮件服务器手册或 Internet 服务提供商的网站。

AUTH LOGIN 验证方法使用 TM\_MAIL、 USERNAME 和 PASSWORD 参数连接邮件服务器。 以前必须在邮件服务器上设置电子邮件帐户的用户名和密码。



如果没有为 **USERNAME** 参数分配任何值，则不会使用 **AUTH LOGIN** 验证方法，并且电子邮件将在没有验证的情况下发送。

### TO\_S:、CC: 和 FROM:参数

参数 **TO\_S:**、**CC:** 和 **FROM:** 是字符串，如下面的示例所示：

**TO\_S:** <wenna@mydomain.com>, <ruby@mydomain.com>,

**CC:** <admin@mydomain.com>, <judy@mydomain.com>,

**FROM:** <admin@mydomain.com>

输入这些字符串时必须遵守以下规则：

- 必须输入“**TO\_S:**”、“**CC:**”和“**FROM:**”字符（包括冒号）。
- 在每个地址前必须输入空格字符和起始尖括号“<”。例如，在“**TO\_S:**”和 <电子邮件地址 > 之间必须有空格字符。
- 在每个地址后必须输入结束尖括号“>”。
- 在 **TO\_S:** 和 **CC:** 地址中的每个电子邮件地址后必须输入逗号字符“,”。例如，单个电子邮件地址后的逗号在“**TO\_S:** <email address>,”中是必填项。
- **FROM:** 条目只能使用一个电子邮件地址，并且末尾不能有逗号。

考虑到运行模式和存储器的使用，不会对 **TM\_MAIL** 的 **TO\_S:**、**CC:** 和 **FROM:** 数据执行语法检查。如果未严格遵照上述格式规则。**SMTP** 电子邮件服务器事务将会失败。

### STATUS 和 SFC\_STATUS 参数

**TM\_MAIL** 返回的执行条件代码可分为以下几类：

- **W#16#0000:** **TM\_MAIL** 操作已成功完成
- **W#16#7xxx:** **TM\_MAIL** 操作的状态
- **W#16#8xxx:** 内部调用通信设备或邮件服务器时出错

下表显示了 **TM\_MAIL** 的执行条件代码，但不包括内部调用通信模块时生成的错误代码。

13.1 TM\_Mail 发送电子邮件指令

**说明**

**电子邮件服务器要求**

TM\_MAIL 只能通过端口 25 与使用 SMTP 的电子邮件服务器通信。分配的端口号不能更改。

大多数 IT 部门和外部电子邮件服务器现在都禁用了端口 25 以防止 PC 受病毒感染而变为欺诈电子邮件生成器。

您可通过 SMTP 连接内部邮件服务器，并让内部服务器管理当前安全强化，该安全强化是通过 Internet 将电子邮件转发到外部邮件服务器所必需的功能。

**Internal 电子邮件服务器配置示例**

如果将 Microsoft Exchange 用作内部邮件服务器，则可以配置服务器以使 SMTP 通过分配了 S7-1200 PLC 的 IP 地址访问。配置交换管理控制台：“服务器组态”(Server configuration) >“集线器传输”(Hub transport) >“接收连接器”(Receive connectors) >“IP 转发”(IP relay)。在“网络”(Network) 选项卡上，有名为“从具有这些 IP 地址的远程服务器接收邮件”(Receive mail from remote servers that have these IP addresses) 的框。您可在此处输入执行 TM\_MAIL 指令的 PLC 的 IP 地址。该类使用内部 Microsoft Exchange 服务器的连接无需验证。

**电子邮件服务器配置**

TM\_MAIL 只能使用允许端口 25 通信、SMTP 和 AUTH LOGIN 验证（可选）的电子邮件服务器。

配置兼容的电子邮件服务器帐户以接受远程 SMTP 登录。然后编辑 TM\_MAIL 的背景数据块以输入 TM\_MAIL USERNAME 和 PASSWORD 字符串，这些字符串用于验证与您的电子邮件帐户的连接。

表格 13-4 条件代码

STATUS (W#16#...):	SFC_STATUS (W#16#...):	说明
0000	-	TM_MAIL 操作已完成，且未发生错误。这个零 STATUS 代码不能保证电子邮件确实已发送（请参见此表后的第一条注释）。
7001	-	TM_MAIL 处于激活状态 (BUSY = 1)。
7002	7002	TM_MAIL 处于激活状态 (BUSY = 1)。

STATUS (W#16#...):	SFC_STATUS (W#16#...):	说明
8xxx	xxxx	TM_MAIL 操作已完成, 但内部调用通信指令时出错。有关 SFC_STATUS 参数的详细信息, 请参见底层 PROFINET 开放式用户通信指令的 STATUS 参数说明。
8010	xxxx	连接失败: 有关 SFC_STATUS 参数的详细信息, 请参见 TCON 指令的 STATUS 参数说明。
8011	xxxx	发送数据时出错: 有关 SFC_STATUS 参数的详细信息, 请参见 TSEND 指令的 STATUS 参数说明。
8012	xxxx	接收数据时出错: 有关 SFC_STATUS 参数的详细信息, 请参见 TRCV 指令的 STATUS 参数说明。
8013	xxxx	连接失败: 有关评估 SFC_STATUS 参数的详细信息, 请参见 TCON 和 TDISCON 指令的 STATUS 参数说明。
8014	-	连接失败: 可能输入了错误的邮件服务器 IP 地址 (ADDR_MAIL_SERVER) 或过短的连接时间 (WATCH_DOG_TIME)。也可能是 CPU 未与网络连接或 CPU 组态不正确。
82xx, 84xx, 85xx	-	错误消息来自邮件服务器且对应于 SMTP 协议的错误编号“8”。请参见此表后的第二条注释。
8450	-	操作未执行: 邮箱不可用; 请稍后重试。
8451	-	操作已中止: 处理过程中出现本地错误, 请稍后重试
8500	-	命令语法错误: 原因可能是电子邮件服务器不支持 LOGIN 验证过程。请检查 TM_MAIL 的参数。尝试发送无需验证的电子邮件。尝试用空字符串替换参数 USERNAME。
8501	-	语法错误: 参数不正确; 可能在 TO_S 或 CC 参数中输入了错误地址。
8502	-	未知的命令或命令未执行: 请检查输入的内容, 尤其是参数 FROM。可能是输入不完整, 漏掉了“@”或“.”字符。
8535	-	SMTP 验证不完整。输入的用户名或密码可能不正确。
8550	-	无法访问邮件服务器, 或您没有访问权限。输入的用户名或密码可能不正确, 或者邮件服务器不支持登录访问。该错误的另一个原因可能是在 TO_S 或 CC 参数中字符“@”后面输入的域名不正确。
8552	-	操作已中止: 超出分配的存储器大小; 请稍后重试。
8554	-	传输失败: 请稍后重试。

### 13.1 TM\_Mail 发送电子邮件指令

---

#### 说明

##### 可能未报告的电子邮件传输错误

- 收件人地址输入不正确不会令 TM\_MAIL 产生 STATUS 错误。在这种情况下，无法保证其他具有正确电子邮件地址的收件人能收到电子邮件。
  - 有关 SMTP 错误代码的详细信息，请访问 Internet 或参见邮件服务器的错误文档。也可以从邮件服务器读取最后一条错误消息。该错误消息存储在 TM\_MAIL 背景数据块的参数 buffer1 中。
-

## 在线和诊断工具

### 14.1 状态 LED

CPU 和 I/O 模块使用 LED 提供有关模块或 I/O 的运行状态的信息。

#### CPU 上的状态 LED

CPU 提供以下状态指示灯：

- **STOP/RUN**
  - 黄色常亮指示 STOP 模式
  - 纯绿色指示 RUN 模式
  - 闪烁（绿色和黄色交替）指示 CPU 处于 STARTUP 模式
- **ERROR**
  - 红色闪烁指示有错误，例如，CPU 内部错误，存储卡错误或组态错误（模块不匹配）
  - 纯红色指示硬件出现故障
- **MAINT（维护）**在每次插入存储卡时闪烁。然后 CPU 切换到 STOP 模式。在 CPU 切换到 STOP 模式后，执行以下操作之一以启动存储卡评估：
  - 将 CPU 切换到 RUN 模式
  - 执行存储器复位 (MRES)
  - CPU 循环上电

也可使用 LED 指令 (页 319)来确定 LED 的状态。

表格 14-1 CPU 上的状态 LED

说明	STOP/RUN 黄色/绿色	ERROR 红色	MAINT 黄色
断电	灭	灭	灭
启动、自检或固件更新	闪烁 (黄色和绿色交替)	-	灭
停止模式	亮 (黄色)	-	-

## 14.1 状态 LED

说明	STOP/RUN 黄色/绿色	ERROR 红色	MAINT 黄色
运行模式	亮（绿色）	-	-
取出存储卡	亮（黄色）	-	闪烁
出错	亮（黄色或绿色）	闪烁	-
请求维护	亮（黄色或绿色）	-	亮
硬件出现故障	亮（黄色）	亮	灭
LED 测试或 CPU 固件出现故障	闪烁 (黄色和绿色交替)	闪烁	闪烁

CPU 还提供了两个可指示 PROFINET 通信状态的 LED。打开底部端子块的盖子可以看到 PROFINET LED。

- Link（绿色）点亮指示连接成功
- Rx/Tx（黄色）点亮指示传输活动

CPU 和各数字量信号模块 (SM) 为每个数字量输入和输出提供了 I/O Channel LED。I/O Channel（绿色）通过点亮或熄灭来指示各输入或输出的状态。

## SM 上的状态 LED

此外，各数字量 SM 还提供了指示模块状态的 DIAG LED：

- 绿色指示模块处于运行状态
- 红色指示模块有故障或处于非运行状态

各模拟量 SM 为各路模拟量输入和输出提供了 I/O Channel LED。

- 绿色指示通道已组态且处于激活状态
- 红色指示个别模拟量输入或输出处于错误状态

此外，各模拟量 SM 还提供有指示模块状态的 DIAG LED：

- 绿色指示模块处于运行状态
- 红色指示模块有故障或处于非运行状态

SM 可检测模块的通断电情况（必要时，还可检测现场侧电源）。

表格 14-2 信号模块 (SM) 上的状态 LED

说明	DIAG (红色/绿色)	I/O Channel (红色/绿色)
现场侧电源关闭	呈红色闪烁	呈红色闪烁
没有组态或更新正在进行	呈绿色闪烁	灭
模块已组态且没有错误	亮 (绿色)	亮 (绿色)
错误状态	呈红色闪烁	-
I/O 错误 (启用诊断时)	-	呈红色闪烁
I/O 错误 (禁用诊断时)	-	亮 (绿色)


## 14.2 转到在线并连接到 CPU

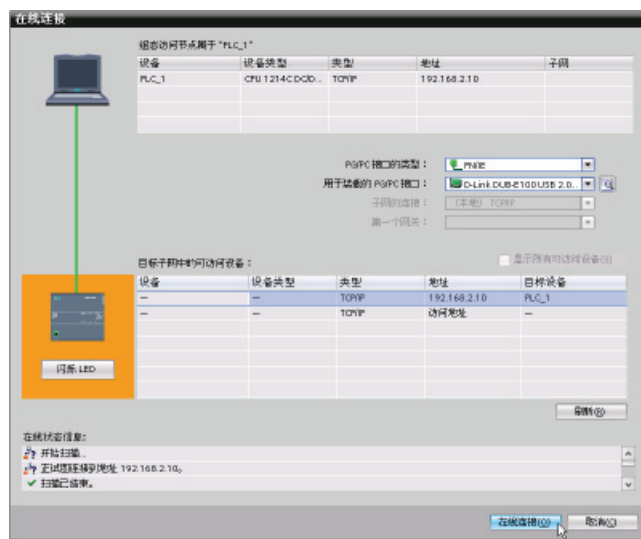
装载程序和项目工程数据以及执行下列操作时，编程设备和 CPU 之间必须存在在线连接：

- 测试用户程序
- 显示和改变 CPU 的工作模式 (页 735)
- 显示和 设置 CPU 的日期和日时钟 (页 734)
- 显示模块信息
- 比较和同步 (页 737)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 736)
- 通过使用 监视表格 (页 741)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 744)

## 14.3 在线为 PROFINET IO 设备分配名称

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：

 转到在线



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。

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。

## 14.3 在线为 PROFINET IO 设备分配名称

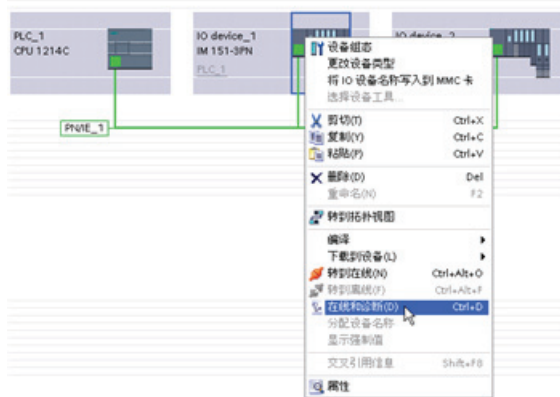
PROFINET 网络中的设备在分配名称后才可与 CPU 连接。如果 PROFINET 设备尚未分配名称，或要更改该设备的名称，则可使用“设备和网络”(Devices & networks) 编辑器为该设备分配名称。

对于各 PROFINET IO 设备，必须在 STEP 7 项目（使用“在线和诊断”(Online & diagnostics) 工具）和 PROFINET IO 设备组态存储器（例如 ET200 S 接口模块组态存储器）中为该设备分配相同的名称。如果名称缺失或两个位置中的名称不匹配，则 PROFINET IO 数据交换模式将不会运行。



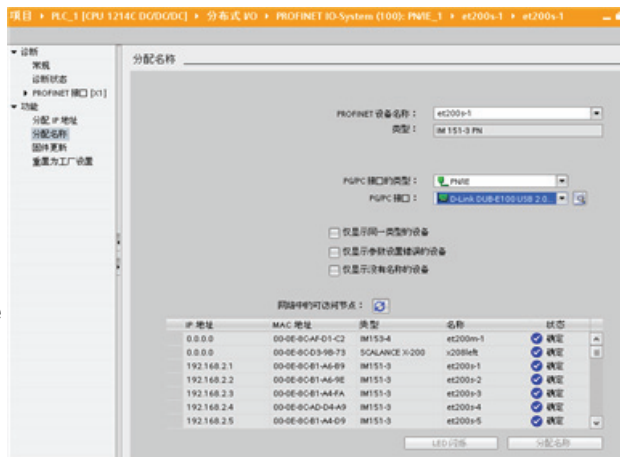
14.3 在线为 PROFINET IO 设备分配名称

1. 在“设备和网络”(Devices & networks) 编辑器中, 右键单击所需的 PROFINET IO 设备, 并选择“在线和诊断”(Online & diagnostics)。



2. 在“在线和诊断”(Online & diagnostics) 对话框中, 选择以下菜单项:

- “功能”(Functions)
  - “分配名称”(Assign name)
- 单击“网络中的可访问设备”(Accessible devices in the network) 图标以显示网络中的所有 PROFINET IO 设备。



3. 在显示的列表中, 单击所需的 PROFINET IO 设备, 然后单击“分配名称”(Assign name) 按钮将该名称写入 PROFINET IO 设备组态存储器。



## 14.4 设置 IP 地址和日时钟

可以设置在线 CPU 中的 IP 地址 (页 138)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。

## 14.5 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 中未插入存储卡。
- 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。

### 操作步骤

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. 如要保留 IP 地址，请选中“保留 IP 地址”(Keep IP address) 复选框；如果要删除 IP 地址，则选中“复位 IP 地址”(Reset 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，并被复位为出厂设置：

- 工作存储器、内部装载存储器以及所有操作数区域都将被清空。
- 所有参数都复位为其默认值。
- 诊断缓冲区被清空。
- 时间被复位。
- IP 地址将根据用户设置被保留或删除。（MAC 地址是固定的，始终不变。）

## 14.6 在线 CPU 的 CPU 操作员面板



“CPU 操作员面板”(CPU operator panel) 显示在线 CPU 的工作模式（STOP 或 RUN）。该面板还显示 CPU 是否有错误或值是否处于强制状态。

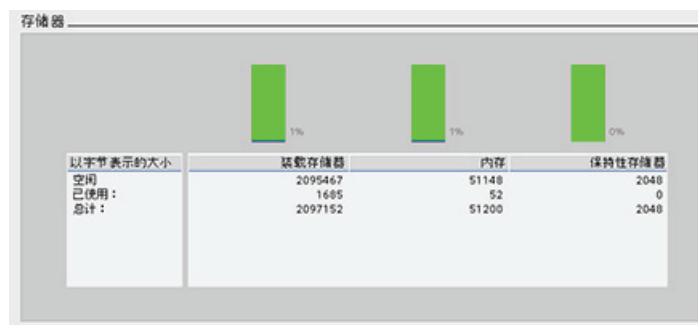
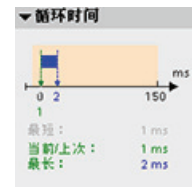
使用“在线工具”(Online Tools) 任务卡的 CPU 操作面板可更改在线 CPU 的工作模式。只要 CPU 处于在线模式，便可访问“在线工具”(Online Tools) 任务卡。

## 14.7 监视循环时间和存储器使用情况

可以监视在线 CPU 的循环时间和存储器使用情况。

连接到在线 CPU 后，打开“在线工具”(Online tools) 任务卡查看以下测量值：

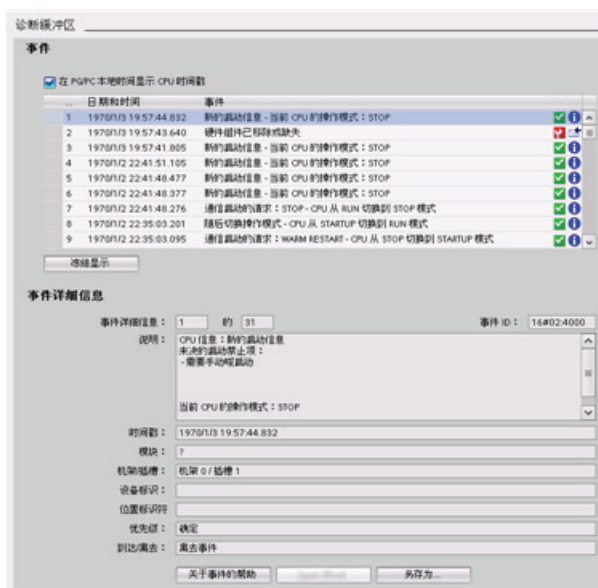
- 循环时间
- 存储器使用情况



## 14.8 显示 CPU 中的诊断事件

使用诊断缓冲区可以查看 CPU 的近期活动。可通过“在线和诊断”(Online & Diagnostics) 访问项目树中在线 CPU 的诊断缓冲区。它包含以下条目：

- 诊断事件
- CPU 工作模式改变（切换到 STOP 或 RUN 模式）



第一个条目包含最新的事件。诊断缓冲区中的各条目均包含记录事件的日期和时间以及一段说明。

最大条目数由 CPU 决定。最多支持 50 个条目。

仅永久存储诊断缓冲区中 10 个最新的事件。将 CPU 复位为工厂设置会通过删除条目的方式复位诊断缓冲区。

还可以使用 GET\_DIAG 指令 (页 324)来采集诊断信息。

## 14.9 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。

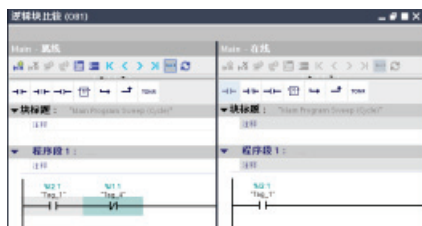
使用“比较离线/在线”(Compare Offline/online) 命令打开“比较”(Compare) 编辑器。（通过“工具”(Tools) 菜单或右键单击项目中 CPU 的方式访问该命令。）



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。

14.10 监视和修改 CPU 中的值



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。

## 14.10 监视和修改 CPU 中的值

STEP 7 提供了用于监视 CPU 的在线工具：

- 您可以显示或监视变量的当前值。监视功能不会改变程序顺序。它为用户提供有关 CPU 中程序的顺序以及数据信息。
- 还可以使用其它功能控制用户程序的顺序和数据：
  - 可以修改在线 CPU 中变量的值，了解用户程序如何响应。
  - 可以将外围设备输出（如 Q0.1:P 或“Start”:P）强制为特定值。
  - 可以在 STOP 模式下启用输出。

### 说明

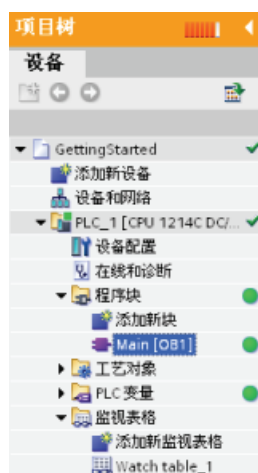
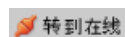
使用控制功能时必须始终小心谨慎。这些功能可能会严重影响用户/系统程序的执行。

表格 14-3 STEP 7 编辑器的在线功能

编辑器	监视	修改	强制
监视表格	有	有	无
强制表格	有	无	有
程序编辑器	有	有	无
变量表	有	无	无
DB 编辑器	有	无	无

### 14.10.1 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。监视表格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
1	"On"	%I 0.0	布尔型		
2	"Off"	%I 0.1	布尔型		
3	"Run"	%Q 0.0	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
1	"On"	%I 0.0	布尔型	<input type="checkbox"/> FALSE	
2	"Off"	%I 0.1	布尔型	<input type="checkbox"/> FALSE	
3	"Run"	%Q 0.0	布尔型	<input type="checkbox"/> FALSE	

“监视值”(Monitor value) 字段中将显示每个变量的值。

### 14.10.2 显示程序编辑器中的状态

还可以在 LAD 和 FBD 程序编辑器中监视各变量的状态。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。

### 14.10 监视和修改 CPU 中的值

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。

#### 14.10.3 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。

---

#### 14.10.4 使用监视表格来监视和修改 CPU 中的值

通过监视表格可以在 CPU 执行用户程序时对数据点执行监视和控制功能。根据监视或控制功能的不同，这些数据点可以是过程映像 (I 或 Q)、M、DB 或物理输入 (I: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 (Q:P)。

监视功能不会改变程序顺序。它为用户提供有关 CPU 中程序的顺序以及数据信息。

控制功能允许用户控制程序的顺序和数据。使用控制功能时必须小心谨慎。这些功能可能会严重影响用户/系统程序的执行。三种控制功能是修改、强制和在 STOP 模式下启用输出。

使用监视表格可以执行以下在线功能：

- 监视变量的状态
- 修改个别变量的值

选择监视或修改变量的时间：

- 扫描循环开始时：在该扫描循环开始时读取或写入值
- 扫描循环结束时：在该扫描循环结束时读取或写入值
- 切换到停止



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。

可使用以下选项监视变量：

- “监视全部”(Monitor all)： 该命令用于启动对激活的监视表格中的可见变量进行监视。
- “立即监视”(Monitor now)： 该命令用于启动对激活的监视表格中的可见变量进行监视。 监视表格仅立即监视变量一次。

可使用以下选项修改变量：

- “修改为 0”(Modify to 0) 将所选地址的值设置为“0”。
- “修改为 1”(Modify to 1) 将所选地址的值设置为“1”。
- “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，则使用“立即修改”(Modify now) 功能。

- “启用外围设备输出”(Enable peripheral outputs) 禁用输出禁用命令并且仅在 CPU 处于 STOP 模式时可用。

要监视变量，必须在线连接到 CPU。

名称	地址	显示格式	监视值	使用触发器监视	使用触发器进	修改变
1 "Start"	%I0.0	布尔型		永久	永久	<input type="checkbox"/>
2 "Stop"	%I0.1	布尔型		永久	永久	<input type="checkbox"/>
3 "Running"	%M0.0	布尔型		永久	永久	<input type="checkbox"/>

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该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。

#### 14.10.4.1 监视或修改 PLC 变量时使用触发器

触发决定将在扫描周期中的哪个点监视或修改所选地址。

表格 14-4 触发器类型

触发器	说明
永久	连续采集数据
扫描周期开始时	永久： CPU 读取输入后，在扫描周期开始时连续采集数据
	一次： CPU 读取输入后，在扫描周期开始时采集一次数据
扫描周期结束时	永久： CPU 写入输出前，在扫描周期结束时连续采集数据
	一次： CPU 写入输出前，在扫描周期结束时采集一次数据
切换到 STOP 时	永久： CPU 切换到 STOP 时连续采集数据
	一次： CPU 切换到 STOP 后采集一次数据

要在给定触发点修改 PLC 变量，请选择周期开始或结束。

- 修改输出： 触发修改输出事件的最佳时机是在扫描周期结束且 CPU 马上要写入输出之前的时间。

在扫描周期开始时监视输出的值以确定写入到物理输出中的值。此外，在 CPU 将值写入到物理输出前监视输出以检查程序逻辑并与实际 I/O 行为进行比较。


- 修改输入： 触发修改输入事件的最佳时机是在周期开始、CPU 刚读取输入且用户程序要使用输入值之前的时间。

如果在扫描周期开始时修改输入，则还应在扫描周期结束时监视输入值，以确保扫描周期结束时的输入值自扫描周期开始起未改变。如果值不同，则用户程序可能会错误地写入到输入。

要诊断 CPU 转到 STOP 的可能原因，请使用“切换到 STOP”(Transition to STOP) 触发器捕捉上一个过程值。

#### 14.10.4.2 在 STOP 模式下启用输出

监视表格允许用户在 CPU 处于 STOP 模式时写入输出。通过该功能可以检查输出的接线并检验连接到输出引脚的电线是将高电平信号还是低电平信号引入与其相连的过程设备端子。

 <b>警告</b>
即使在 CPU 处于 STOP 模式时，启用物理输出也可激活相连的过程点。

输出启用时，可以在 STOP 模式下修改输出的状态。如果输出禁用，则无法在 STOP 模式下修改输出。

- 要启用在 STOP 模式下修改输出，请选择“在线”(Online) 菜单中的“修改”(Modify) 命令的“启用外围设备输出”(Enable peripheral outputs) 选项或右键单击监视表格行。

如果已组态分布式 I/O，则无法在 STOP 模式下启用输出。若尝试启动，将返回错误。

- 将 CPU 设置为 RUN 模式会禁用“启用外围设备输出”(Enable peripheral outputs) 选项。
- 如果任何输入或输出被强制，则处于 STOP 模式时不允许 CPU 启用输出。必须先取消强制功能。

## 14.10.5 CPU 中的强制值

### 14.10.5.1 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）。



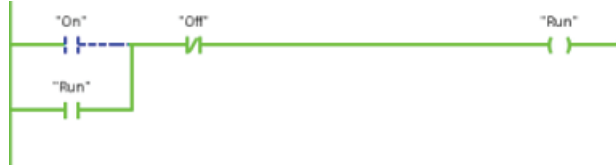
	i	名称	地址	显示格式	监视值	强制值	F
1		"On":P	%I0.0:P	布尔型		TRUE	<input checked="" type="checkbox"/>
2		"Off":P	%I0.1:P	布尔型			<input type="checkbox"/>
3		"Run":P	%Q0.1:P	布尔型			<input type="checkbox"/>

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。



#### 注意

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。

### 14.10.5.2

#### 强制功能的操作

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。

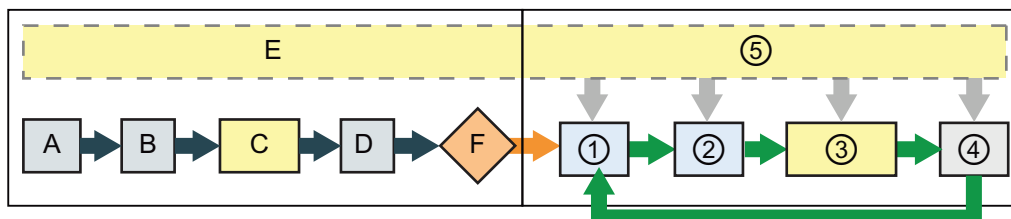
在强制表格中强制输入或输出时，强制操作将变成用户程序的一部分。即使编程软件已关闭，强制选项在运行的 CPU 程序中仍保持激活，直到在线连接到编程软件并停止强制功能将其清除为止。含有通过存储卡装载到另一个 CPU 的强制点的程序将继续强制程序中所选择的点。

如果 CPU 正在执行写保护存储卡上的用户程序，则无法通过监视表格初始化或更改对 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，无法通过监视表格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。

## 14.11 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。
- 创建和删除数据块 (DB)；但是，不会覆盖 DB 的结构更改。只能覆盖 DB 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 模式下最多可同时下载 10 个块。如果要下载 10 个以上的块，那么必须将 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，那么 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。

---

### 14.11.1 “在 RUN 模式下下载”的先决条件

要向 RUN 模式下的 CPU 中下载程序更改，必须满足以下先决条件：

- 程序必须编译成功。
- 必须已在运行 STEP 7 的编程设备和 CPU 之间成功建立通信。
- 在 V3.0 或更高版本中，目标 CPU 的固件必须支持“在 RUN 模式下下载”功能。

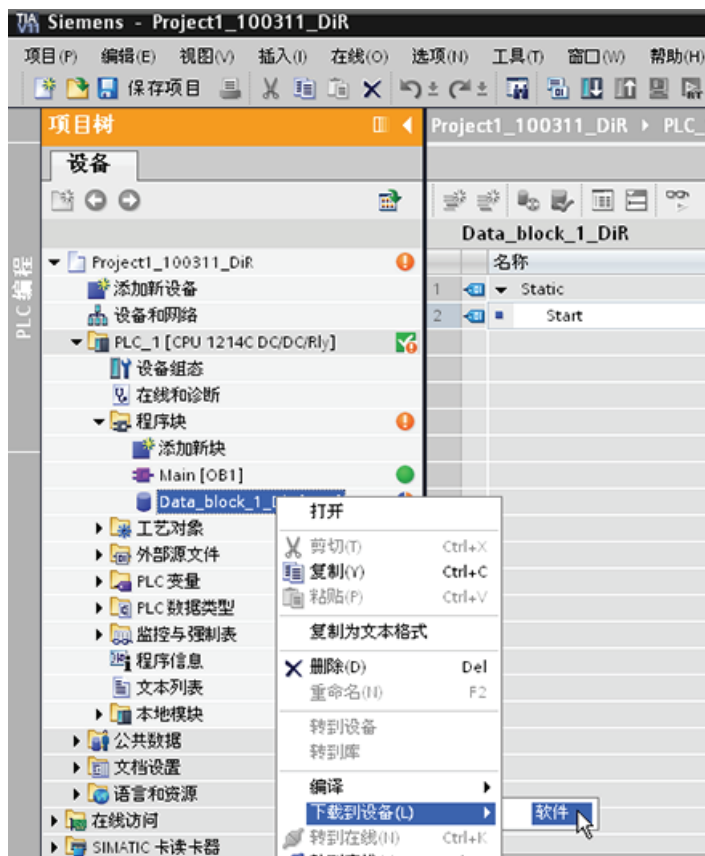
### 14.11.2 在 RUN 模式下更改程序

要在 RUN 模式下更改程序，必须首先确保 CPU 支持“在 RUN 模式下下载”，并且 CPU 已处于 RUN 模式：

1. 如果要在 RUN 模式下下载程序，请选择以下某种方法：

14.11 在 RUN 模式下下载

- 从“在线”(Online) 菜单中选择“下载到设备”(Download to device) 命令
- 按下工具栏中的“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。



2. 如果程序已编译成功，STEP 7 会将该程序下载到 CPU 中。
3. STEP 7 将提示加载程序或取消操作。
4. 如果单击“加载”(Load)，则 STEP 7 将把程序下载到 CPU。



### 14.11.3 下载所选块

光标放在“程序块”(Program blocks) 文件夹上，选择多个块或单个块。

1. 如果在块编辑器中选择下载单个块，那么“操作”(Action) 列中将只显示“统一下载”(Consistent download) 一个选项。可以展开类别行，以确保选择要加载的块。在本例中，仅对离线块进行少量更改，无需加载其它块。

2. 在本例中，需要下载多个块。



#### 说明

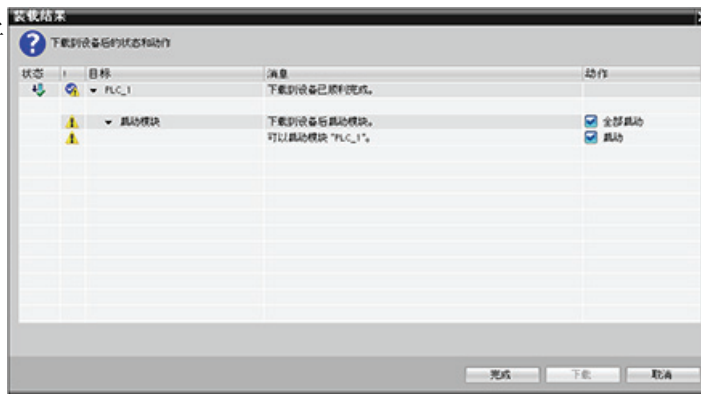
在 RUN 模式下，同时可最多下载 10 个块。如果要下载 10 个以上的块，那么必须将 CPU 置于 STOP 模式。

14.11 在 RUN 模式下下载

3. 如果尝试在 RUN 模式下下载，但系统在实际下载前检测出无法执行该操作，则该对话框中将显示“停止模块”(Stop modules) 类别行。



4. 单击“加载”(Load) 按钮，将显示“加载结果”(Load results) 对话框。单击“完成”(Finish) 按钮完成下载。



14.11.4

其它块中存在编译错误时下载选定的单个块

如果当其它块中存在编译错误时尝试执行统一下载，则该对话框中将显示一个错误信息，并禁用加载按钮。



用户必须更正其它块中的编译错误。之后，才会激活“加载”(Load) 按钮。



### 14.11.5 下载失败时的系统响应

执行“在 RUN 模式下下载”的过程中, 如果出现网络连接故障, 那么将显示“加载预览”(Load preview) 对话框, 如下图所示。



### 14.11.6 在 RUN 模式下下载程序

在 RUN 模式下下载程序之前, 如果发生以下情况, 则需考虑 RUN 模式下进行修改对 CPU 运行的影响:

- 如果删除一个输出的控制逻辑, 则在下一次上电循环或切换到 STOP 模式之前, CPU 将始终保持该输出的最终状态。
- 如果删除了正在运行的高速计数器或脉冲输出函数, 则该高速计数器或脉冲输出将继续运行, 直至下一次上电循环或切换到 STOP 模式。

14.11 在 RUN 模式下下载

- 在下一次上电循环或者从 STOP 切换到 RUN 模式之前，任何以首次扫描位状态为条件的逻辑都不会执行。首次扫描位只会因切换到 RUN 模式而置位，不受 RUN 模式下下载的影响。
- 不会覆盖数据块 (DB) 的当前值和/或变量。

**说明**

CPU 必须支持在 RUN 模式下进行更改，程序的编译必须没有错误，CPU 必须能与 STEP 7 通信，并且 CPU 必须无错误，这样才能在 RUN 模式下下载程序。

可在 RUN 模式下对程序块和变量进行以下更改，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。
- 创建和删除数据块 (DB)；但是，不会覆盖 DB 的结构更改。只能覆盖 DB 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，同时可最多下载 10 个块。如果要下载 10 个以上的块，那么必须将 CPU 置于 STOP 模式。

下载一旦启动，在其完成前 STEP 7 将无法执行其它任务。

由于“在 RUN 模式下下载”，可能导致出错的指令

CPU 中激活了“在 RUN 模式下下载”后，以下指令可能会发生临时错误。如果 CPU 正准备激活已下载的更改，那么初始化指令时将出现错误。在此过程中，CPU 将暂停用户程序访问加载存储器的初始化过程，同时完成正在进行的用户程序对加载存储器的访问。完成后，将统一激活所下载的更改。

指令	暂停激活时的响应
DataLogCreate	STATUS = W#16#80C0, ERROR = TRUE
DataLogOpen	STATUS = W#16#80C0, ERROR = TRUE
DataLogWrite	STATUS = W#16#80C0, ERROR = TRUE
DataLogClose	STATUS = W#16#80C0, ERROR = TRUE
DataLogNewFile	STATUS = W#16#80C0, ERROR = TRUE
READ_DBL	RET_VAL = W#16#82C0
WRIT_DBL	RET_VAL = W#16#82C0
RTM	RET_VAL = 0x80C0

---

无论何种情况，只要发生错误，指令的 RLO 输出都将失败。该错误是临时错误。如果出现错误，则需稍后重试该指令。

---

**说明**

而不能在执行 OB 的过程中重试该操作。

---

14.11 在 RUN 模式下下载

## 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: 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 60079-15:2005: 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  
IA AS RD ST PLC Amberg  
Werner-von-Siemens-Str. 50  
D92224 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）

#### 注意

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，替换组件可能会影响其安全性。  
只能由得到授权的 Siemens 维修中心维修设备。

### 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 60079-0:2006：爆炸性环境 - 一般要求

EN 60079-15:2005：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 II T3

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。



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，并且考虑设备将来使用的环境条件。

在额定条件下，如果电缆入口点温度超出 70° C 或者导线分支点超出 80° C，则所选电缆的温度规范应符合实际测量温度。

应采取措施防止额定电压受暂态干扰而超出 40% 以上。

### C-Tick 认证



S7-1200 自动化系统满足 AS/NZS 2064 (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。

### 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)

## 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	噪声辐射要求	抗噪声要求
工业	EN 61000-6-4:2007	EN 61000-6-2:2005

## 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 AC 和 DC 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	AC 系统 - 2 kV 共模, 1kV 差模 DC 系统 - 2 kV 共模, 1kV 差模 对于 DC 系统 (I/O 信号、DC 电源系统), 需要外部保护。
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	AC 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

表格 A-3 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

## 环境条件

表格 A-4 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, 停留时间 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-5 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 <sub>2</sub> : < 0.5 ppm; H <sub>2</sub> S: < 0.1 ppm; RH < 60%, 不结露
EN 60068-2-14, 测试 Nb, 温度变化	5° C 至 55° C, 3° C/分钟

A.1 常规技术规范

环境条件 - 运行	
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

注意
对于必须在 -20° C 到 0° C 范围内启动的系统, 在启动后用户程序应推迟通电输出 10 秒。

表格 A-6 高电位绝缘测试

高电位绝缘测试	
24 V/5 V 标称电路间	520 VDC (光隔离边界的型式测试)
115/230 V 电路对地	1500 VAC
115/230 V 电路对 115/230 V 电路	1500 VAC
115 V/230V 电路对 24 V/5 V 电路	1500 VAC (3000 VAC/4242 VDC 型式测试)
24V/5V 电路和接地电路的以太网端口 <sup>1</sup>	1500 VAC (仅限型式测试)

<sup>1</sup> 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-7 额定电压

额定电压	容错
24 VDC	20.4 VDC 到 28.8 VDC 22.0 VDC 到 28.8 VDC (环境温度 -20° C 到 0° C)
120/230 VAC	85 VAC 到 264 VAC, 47 到 63 Hz

### 注意

机械触点接通 S7-1200 CPU 的输出电源或任何数字量信号模块时，会发送信号“1”到数字量输出，时间约 50 微秒。这可能引发意外的机械或过程操作，从而导致死亡、重伤和/或设备损坏。必须考虑这一点，尤其是使用响应短脉冲的设备时。

## 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、SM 的继电器线圈的电源输入或非隔离模拟输入的电源。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

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。

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。

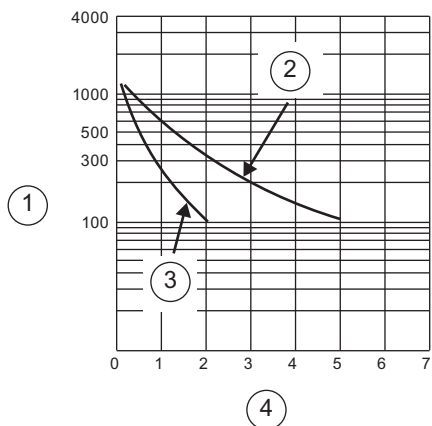
务必确保 S7-1200 系统中的所有非隔离 M 端子都连接到同一个参考电位。

## DC 输出

短路保护电路不适用于 CPU 上的 DC 输出、信号模块 (SM) 和信号板 (SB)。

### 继电器电气使用寿命

继电器供应商提供的典型性能数据如下。根据具体应用，实际性能可能不同。使用适合于负载的外部保护电路可增强触点的使用寿命。



- ① 使用寿命 (x 10<sup>3</sup> 次动作)
- ② 250 VAC 阻性负载  
30 VDC 阻性负载
- ③ 250 VAC 感性负载 (p.f=0.4)  
30 VDC 感性负载 (L/R=7ms)
- ④ 额定工作电流 (A)

## A.2 CPU 1211C

### A.2.1 常规规范和特性

表格 A-8 常规规范

技术数据	CPU 1211C AC/DC/继电器	CPU 1211C DC/DC/继电器	CPU 1211C DC/DC/DC
订货号	6ES7 211-1BE31-0XB0	6ES7 211-1HE31-0XB0	6ES7 211-1AE31-0XB0
尺寸 W x H x D (mm)	90 x 100 x 75	90 x 100 x 75	90 x 100 x 75
装运重量	420 g	380 g	370 g
功耗	10 W	8 W	8 W
可用电流 (CM 总线)	最大 750 mA (5 VDC)	最大 750 mA (5 VDC)	最大 750 mA (5 VDC)
可用电流 (24 VDC)	最大 300 mA (传感器电源)	最大 300 mA (传感器电源)	最大 300 mA (传感器电源)
数字量输入电流消耗 (24VDC)	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

表格 A-9 CPU 特征

技术数据	说明	
用户存储器 <sup>1</sup>	工作	30 KB
	负载	内置 1 MB，可用 SD 卡扩展，具体视卡容量而定
	保持性	10 KB
板载数字 I/O	6 点输入/4 点输出	
板载模拟 I/O	2 路输入	
过程映像大小	1024 字节输入 (I)/1024 字节输出 (Q)	
位存储器 (M)	4096 个字节	
临时（局部）存储器	<ul style="list-style-type: none"> <li>• 16 KB 用于启动和程序循环（包括相关的 FB 和 FC）</li> <li>• 4 KB 用于标准中断事件（包括 FB 和 FC）</li> <li>• 4 KB 用于错误中断事件（包括 FB 和 FC）</li> </ul>	
信号模块扩展	否	
SB、CB、BB 扩展	最多 1 个	
通信模块扩展	最多 3 个通信模块	
高速计数器	3 个内置 I/O，信号板为 5 个，参见表格 CPU 1211C 的 HSC 输入分配 (页 365) <ul style="list-style-type: none"> <li>• 单相：3 个，100 kHz，SB：2 个，30 kHz</li> <li>• 正交相位：3 个，80 kHz，SB：2 个，20 kHz</li> </ul>	
脉冲输出 <sup>2</sup>	4	
脉冲捕捉输入	6	
延时中断/循环中断	共 4 个，精度为 1 ms	
沿中断	6 个上升沿和 6 个下降沿（使用可选信号板时，各为 10 个）	
存储卡	SIMATIC 存储卡（选件）	
实时时钟精度	+/- 60 秒/月	
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）	

<sup>1</sup> 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的限制。对支持的 OB、FC、FB 和 DB 块数目或特定块的大小没有特殊限制；唯一的限制是存储器的总大小。

<sup>2</sup> 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- 10 性能

指令类型	执行速度
布尔运算	0.08 μs/指令
移动字	1.7 μs/指令
实数数学运算	2.3 μs/指令

### A.2.2 CPU 1211C 支持的定时器、计数器和代码块

表格 A- 11 CPU 1211C 支持的块、定时器和计数器

元素	说明	
块	类型	OB、FB、FC、DB
	大小	30 KB
	数量	一共多达 1024 个块 (OB + FB + FC + DB)
	FB、FC 和 DB 的地址范围	1 到 65535 (例如 FB 1 到 FB 65535)
	嵌套深度	16 (从程序循环或启动 OB 开始) ; 4 (从延时中断、日时钟中断、循环中断、硬件中断、时间错误中断或诊断错误中断 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: OB 1、OB 200 到 OB 65535
	启动	多个: OB 100、OB 200 到 OB 65535
	延时中断和循环中断	4 <sup>1</sup> (每个事件 1 个) : OB 200 到 OB 65535
	硬件中断 (沿和 HSC)	50 (每个事件 1 个) : OB 200 到 OB 65535
	时间错误中断	1: OB 80
	诊断错误中断	1: OB 82
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"> <li>• SInt 和 USInt: 3 个字节</li> <li>• Int 和 UInt: 6 个字节</li> <li>• DInt 和 UDInt: 12 个字节</li> </ul>

<sup>1</sup> 延时中断和循环中断在 CPU 中使用相同的资源。延时中断和循环中断的总和只能为 4 个。不能有 4 个延时中断和 4 个循环中断。

表格 A- 12 通信

技术数据	说明
端口数	1
类型	以太网
HMI 设备 <sup>1</sup>	3
编程设备 (PG)	1
连接	<ul style="list-style-type: none"> <li>• 8 个用于开放式用户通信（主动或被动）：TSEND_C、TRCV_C、TCON、TDISCON、TSEND 和 TRCV</li> <li>• 3 个用于服务器 GET/PUT（CPU 间）S7 通信</li> <li>• 8 个用于客户端 GET/PUT（CPU 间）S7 通信</li> </ul>
数据传输率	10/100 Mb/s
隔离（外部信号与 PLC 逻辑侧）	变压器隔离，1500 VAC（仅针对短期事件安全）
电缆类型	CAT5e 屏蔽电缆

<sup>1</sup> CPU 提供专用的 HMI 连接，以支持最多 3 个 HMI 设备。（最多可以有 2 个 SIMATIC 精智面板。）支持的 HMI 总数受组态中 HMI 面板类型的影响。例如，可以将最多 3 个 SIMATIC 基本面板连接到 CPU，或者最多可以连接两个 SIMATIC 精智面板与一个附加基本面板。

表格 A- 13 电源

技术数据	CPU 1211C AC/DC/继电器	CPU 1211C DC/DC/继电器	CPU 1211C DC/DC/DC
电压范围	85 到 264 VAC	20.4 VDC 到 28.8 VDC 22.0 VDC 到 28.8 VDC（环境温度 -20° C 到 0° C）	
电源频率	47 到 63 Hz	--	--

A.2 CPU 1211C

技术数据	CPU 1211C AC/DC/继电器	CPU 1211C DC/DC/继电器	CPU 1211C DC/DC/DC
输入电流 最大负载时仅包括 CPU	120 VAC 时 60 mA 240 VAC 时 30 mA	24 VDC 时 300 mA	24 VDC 时 300 mA
	最大负载时包括 CPU 和 所有扩展附件	120 VAC 时 180 mA 240 VAC 时 90 mA	24 VDC 时 900 mA 24 VDC 时 900 mA
浪涌电流 (最大)	264 VAC 时 20 A	28.8 VDC 时 12 A	28.8 VDC 时 12 A
隔离 (输入电源与逻辑侧)	1500 VAC	未隔离	未隔离
漏地电流, AC 线路对功能地	最大 0.5 mA	--	--
保持时间 (掉电)	120 VAC 时 20 ms 240 VAC 时 80 ms	24 VDC 时 10 ms	24 VDC 时 10 ms
内部保险丝, 用户不可更换	3 A, 250 V, 慢速熔 断	3 A, 250 V, 慢速 熔断	3 A, 250 V, 慢速熔 断

表格 A- 14 传感器电源

技术数据	CPU 1211C AC/DC/继电器	CPU 1211C DC/DC/继电器	CPU 1211C DC/DC/DC
电压范围	20.4 到 28.8 VDC	L+ - 4 VDC (最小) L+ - 5 VDC (最小) (环境温度 -20° C 到 0° C)	
额定输出电流 (最大)	300 mA (短路保护)	300 mA (短路保护)	300 mA (短路保 护)
最大波纹噪声 (<10 MHz)	< 1 V 峰峰值	与输入线路相同	与输入线路相同
隔离 (CPU 逻辑侧与传感器电源)	未隔离	未隔离	未隔离

### A.2.3 数字量输入和输出

表格 A- 15 数字量输入

技术数据	CPU 1211C AC/DC/继电器、CPU 1211C DC/DC/继电器和 CPU 1211C DC/DC/DC
输入点数	6
类型	漏型/源型 (IEC 1 类漏型)
额定电压	4 mA 时 24 VDC, 额定值
允许的连续电压	最大 30 VDC
浪涌电压	35 VDC, 持续 0.5 s
逻辑 1 信号 (最小)	2.5 mA 时 15 VDC
逻辑 0 信号 (最大)	1 mA 时 5 VDC
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 分钟
隔离组	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)
HSC 时钟输入频率 (最大) (逻辑 1 电平 = 15 到 26 VDC)	单相: 100 KHz 正交相位: 80 KHz
同时接通的输入数	6, 最高 60° C (水平) 或 50° C (垂直) 时
电缆长度 (米)	500 m (屏蔽); 300 m (非屏蔽); 50 m (屏蔽, HSC 输入)

表格 A- 16 数字量输出

技术数据	CPU 1211C AC/DC/继电器和 CPU 1211C DC/DC/继电器	CPU 1211C DC/DC/DC
输出点数	4	4
类型	继电器, 干触点	固态 - MOSFET (源型)
电压范围	5 到 30 VDC 或 5 到 250 VAC	20.4 到 28.8 VDC
最大电流时的逻辑 1 信号	--	最小 20 VDC
具有 10 K $\Omega$ 负载时的逻辑 0 信号	--	最大 0.1 VDC
电流 (最大)	2.0 A	0.5 A
灯负载	30 W DC/200 W AC	5 W

技术数据	CPU 1211C AC/DC/继电器和 CPU 1211C DC/DC/继电器	CPU 1211C DC/DC/DC
通态电阻	新设备最大为 0.2 Ω	最大 0.6 Ω
每点的漏泄电流	--	最大 10 μA
浪涌电流	触点闭合时为 7 A	8 A, 最长持续 100 ms
过载保护	无	无
隔离（现场侧与逻辑侧）	1500 VAC, 持续 1 分钟 （线圈与触点） 无（线圈与逻辑侧）	500 VAC, 持续 1 分钟
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	--
隔离组	1	1
电感钳位电压	--	L+ - 48 VDC, 1 W 损耗
继电器最大开关频率	1 Hz	--
开关延迟（Qa.0 到 Qa.3）	最长 10 ms	断开到接通最长为 1.0 μs 接通到断开最长为 3.0 μs
脉冲串输出频率 （Qa.0 和 Qa.2）	不推荐 <sup>1</sup>	最大 100 KHz, 最小 2 Hz <sup>2</sup>
机械寿命（无负载）	10,000,000 个断开/闭合周期	--
额定负载下的触点寿命	100,000 个断开/闭合周期	--
RUN 到 STOP 时的行为	上一个值或替换值 （默认值为 0）	上一个值或替换值 （默认值为 0）
同时接通的输出数	4, 最高 60° C（水平）或 50° C（垂直）时	
电缆长度（米）	500 m（屏蔽）， 150 m（非屏蔽）	500 m（屏蔽）， 150 m（非屏蔽）

<sup>1</sup> 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。

<sup>2</sup> 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。

## A.2.4 模拟量输入

表格 A- 17 模拟量输入

技术数据	说明
输入路数	2
类型	电压（单侧）
满量程范围	0 到 10 V
满量程范围（数据字）	0 到 27648
过冲范围	10.001 到 11.759 V
过冲范围（数据字）	27,649 到 32,511
上溢范围	11.760 到 11.852 V
上溢范围（数据字）	32,512 到 32,767
分辨率	10 位
最大耐压	35 VDC
平滑化	无、弱、中或强 请参见 CPU 模拟量输入的阶跃响应 (ms) (页 769)表格。
噪声抑制	10、50 或 60 Hz
阻抗	≥100 KΩ
隔离（现场侧与逻辑侧）	无
精度（25°C/-20 到 60°C）	满量程的 3.0%/3.5%
电缆长度（米）	100 m，屏蔽双绞线

### A.2.4.1 CPU 内置模拟量输入的阶跃响应

表格 A- 18 阶跃响应 (ms)，0V 到 10V（在 95% 处测得）

平滑化选项（采样平均）	抑制频率（积分时间）		
	60 Hz	50 Hz	10 Hz
无（1 个周期）：不求平均值	50 ms	50 ms	100 ms
弱（4 个周期）：4 次采样	60 ms	70 ms	200 ms
中（16 个周期）：16 次采样	200 ms	240 ms	1150 ms

平滑化选项（采样平均）	抑制频率（积分时间）		
	60 Hz	50 Hz	10 Hz
强（32 个周期）： 32 次采样	400 ms	480 ms	2300 ms
采样时间	4.17 ms	5 ms	25 ms

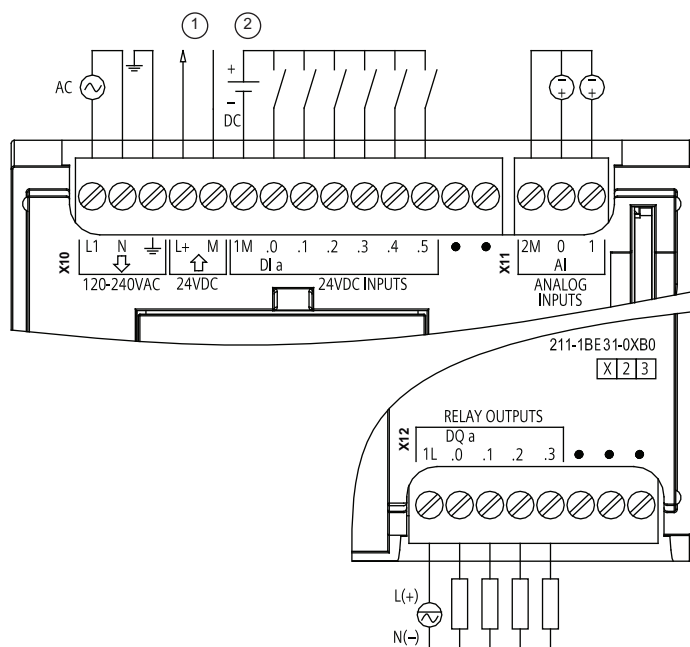
### A.2.4.2 CPU 内置模拟端口的采样时间

表格 A- 19 CPU 内置模拟量输入的采样时间

抑制频率（积分时间选项）	采样时间
60 Hz (16.6 ms)	4.17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

### A.2.5 CPU 1211C 的接线图

表格 A- 20 CPU 1211C AC/DC/继电器 (6ES7 211-1BE31-0XB0)

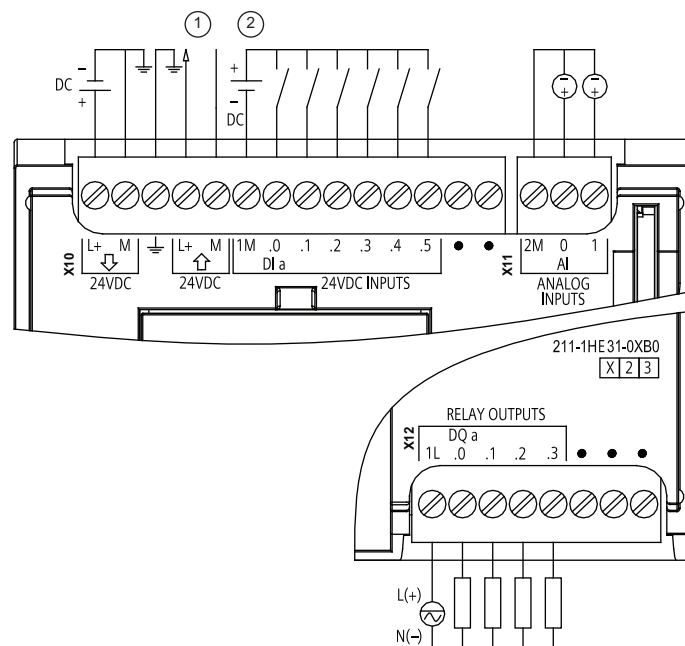


- ① 24 VDC 传感器电源输出  
要获得更好的抗噪声效果，即使未使用传感器电源，也可将“M”连接到机壳接地。
  - ② 对于漏型输入，将“-”连接到“M”（如图所示）。  
对于源型输入，将“+”连接到“M”。
- 注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A-21 CPU 1211C AC/DC/继电器 (6ES7 211-1BE31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L1/120-240 VAC	2 M	1L
2	N/120-240 VAC	AI 0	DQ a.0
3	功能性接地	AI 1	DQ a.1
4	L+/24VDC 传感器输出	--	DQ a.2
5	M/24VDC 传感器输出	--	DQ a.3
6	1M	--	无连接
7	DI a.0	--	无连接
8	DI a.1	--	无连接
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	无连接	--	--
14	无连接	--	--

表格 A-22 CPU 1211C DC/DC/继电器 (6ES7 211-1HE31-0XB0)



- 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。
- 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”。

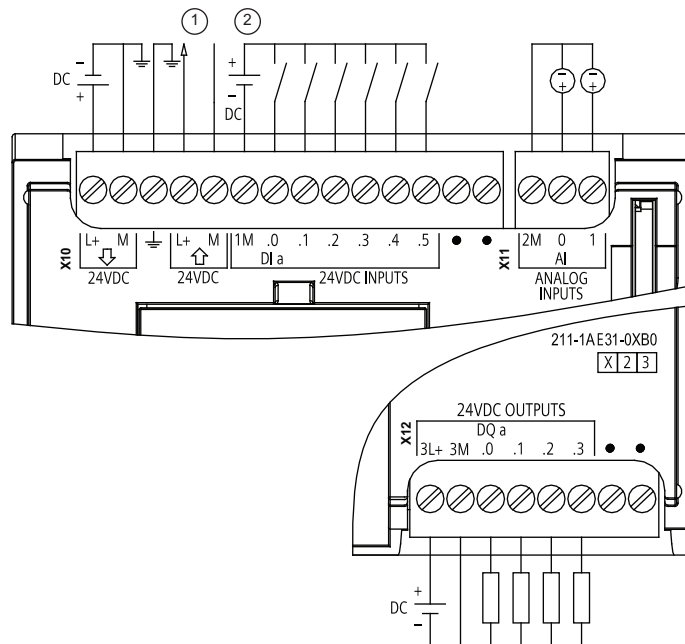
注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A- 23 CPU 1211C DC/DC/继电器 (6ES7 211-1BE31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L+/24VDC	2 M	1L
2	M/24VDC	AI 0	DQ a.0
3	功能性接地	AI 1	DQ a.1
4	L+/24VDC 传感器输出	--	DQ a.2
5	M/24VDC 传感器输出	--	DQ a.3
6	1M	--	无连接
7	DI a.0	--	无连接
8	DI a.1	--	无连接
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	无连接	--	--
14	无连接	--	--



表格 A-24 CPU 1211C DC/DC/DC (6ES7 211-1AE31-0XB0)



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。

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”。

注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A-25 CPU 1211C DC/DC/DC (6ES7 211-1AE31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L+/24VDC	2 M	3L+
2	M/24VDC	AI 0	3M
3	功能性接地	AI 1	DQ a.0
4	L+/24VDC 传感器输出	--	DQ a.1
5	M/24VDC 传感器输出	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	无连接
8	DI a.1	--	无连接
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	无连接	--	--
14	无连接	--	--

A.3 CPU 1212C

说明

应将未使用的模拟量输入短路。

A.3 CPU 1212C

A.3.1 常规规范和特性

表格 A-26 常规

技术数据	CPU 1212C AC/DC/继电器	CPU 1212C DC/DC/继电器	CPU 1212C DC/DC/DC
订货号	6ES7 212-1BE31-0XB0	6ES7 212-1HE31-0XB0	6ES7 212-1AE31-0XB0
尺寸 W x H x D (mm)	90 x 100 x 75	90 x 100 x 75	90 x 100 x 75
装运重量	425 g	385 g	370 g
功耗	11 W	9 W	9 W
可用电流 (SM 和 CM 总线)	最大 1000 mA (5 VDC)	最大 1000 mA (5 VDC)	最大 1000 mA (5 VDC)
可用电流 (24 VDC)	最大 300 mA (传感器电源)	最大 300 mA (传感器电源)	最大 300 mA (传感器电源)
数字输入电流消耗 (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

表格 A-27 CPU 特征

技术数据	说明	
用户存储器 <sup>1</sup>	工作	50 KB
	负载	内置 1 MB, 可用 SD 卡扩展, 具体视卡容量而定
	保持性	10 KB
板载数字 I/O	8 点输入/6 点输出	
板载模拟 I/O	2 路输入	

技术数据	说明
过程映像大小	1024 字节输入 (I)/1024 字节输出 (Q)
位存储器 (M)	4096 个字节
临时 (局部) 存储器	<ul style="list-style-type: none"> <li>• 16 KB 用于启动和程序循环 (包括相关的 FB 和 FC)</li> <li>• 4 KB 用于标准中断事件 (包括 FB 和 FC)</li> <li>• 4 KB 用于错误中断事件 (包括 FB 和 FC)</li> </ul>
信号模块扩展	最多 2 个信号模块
SB、CB、BB 扩展	最多 1 个
通信模块扩展	最多 3 个通信模块
高速计数器	5 个内置 I/O, 信号板为 6 个, 参见表格 CPU 1212C 的 HSC 输入分配 (页 365) <ul style="list-style-type: none"> <li>• 单相: 3 个 100 kHz 以及 1 个 30 kHz 的时钟频率, SB: 2 个, 30 kHz</li> <li>• 正交相位: 3 个 80 kHz 以及 1 个 20 kHz 的时钟频率, SB: 2 个, 20 kHz</li> </ul>
脉冲输出 <sup>2</sup>	4
脉冲捕捉输入	8
延时中断/循环中断	共 4 个, 精度为 1 ms
沿中断	8 个上升沿和 8 个下降沿 (使用可选信号板时, 各为 12 个)
存储卡	SIMATIC 存储卡 (选件)
实时时钟精度	+/- 60 秒/月
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)

<sup>1</sup> 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的限制。对支持的 OB、FC、FB 和 DB 块数目或特定块的大小没有特殊限制; 唯一的限制是存储器的总大小。

<sup>2</sup> 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- 28 性能

指令类型	执行速度
布尔运算	0.08 μs/指令
移动字	1.7 μs/指令
实数数学运算	2.3 μs/指令

### A.3.2 CPU 1212C 支持的定时器、计数器和代码块

表格 A-29 CPU 1212C 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
	数量	一共多达 1024 个块 (OB + FB + FC + DB)
	FB、FC 和 DB 的地址范围	1 到 65535 (例如 FB 1 到 FB 65535)
	嵌套深度	16 (从程序循环或启动 OB 开始) ; 4 (从延时中断、日时钟中断、循环中断、硬件中断、时间错误中断或诊断错误中断 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: OB 1、OB 200 到 OB 65535
	启动	多个: OB 100、OB 200 到 OB 65535
	延时中断和循环中断	4 <sup>1</sup> (每个事件 1 个) : OB 200 到 OB 65535
	硬件中断 (沿和 HSC)	50 (每个事件 1 个) : OB 200 到 OB 65535
	时间错误中断	1: OB 80
	诊断错误中断	1: OB 82
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
	存储	DB 结构, 大小取决于计数类型 <ul style="list-style-type: none"> <li>• SInt 和 USInt: 3 个字节</li> <li>• Int 和 UInt: 6 个字节</li> <li>• DInt 和 UDInt: 12 个字节</li> </ul>

<sup>1</sup> 延时中断和循环中断在 CPU 中使用相同的资源。延时中断和循环中断的总和只能为 4 个。不能有 4 个延时中断和 4 个循环中断。

表格 A- 30 通信

技术数据	说明
端口数	1
类型	以太网
HMI 设备 <sup>1</sup>	3
编程设备 (PG)	1
连接	<ul style="list-style-type: none"> <li>• 8 个用于开放式用户通信（主动或被动）：TSEND_C、TRCV_C、TCON、TDISCON、TSEND 和 TRCV</li> <li>• 3 个用于服务器 GET/PUT（CPU 间）S7 通信</li> <li>• 8 个用于客户端 GET/PUT（CPU 间）S7 通信</li> </ul>
数据传输率	10/100 Mb/s
隔离（外部信号与 PLC 逻辑侧）	变压器隔离，1500 VAC（仅针对短期事件安全）
电缆类型	CAT5e 屏蔽电缆

<sup>1</sup> CPU 提供专用的 HMI 连接，以支持最多 3 个 HMI 设备。（最多可以有 2 个 SIMATIC 精智面板。）支持的 HMI 总数受组态中 HMI 面板类型的影响。例如，可以将最多 3 个 SIMATIC 基本面板连接到 CPU，或者最多可以连接两个 SIMATIC 精智面板与一个附加基本面板。

表格 A- 31 电源

技术数据	CPU 1212C AC/DC/继电器	CPU 1212C DC/DC/继电器	CPU 1212C DC/DC/DC
电压范围	85 到 264 VAC	20.4 VDC 到 28.8 VDC 22.0 VDC 到 28.8 VDC（环境温度 -20° 到 0° C）	
电源频率	47 到 63 Hz	--	--
输入电流（最大负载时）	仅 CPU 120 VAC 时 80 mA 240 VAC 时 40 mA	24 VDC 时 400 mA	24 VDC 时 400 mA
	具有所有扩展附件的 CPU 120 VAC 时 240 mA 240 VAC 时 120 mA	24 VDC 时 1200 mA	24 VDC 时 1200 mA
浪涌电流（最大）	264 VAC 时 20 A	28.8 VDC 时 12 A	28.8 VDC 时 12 A
隔离（输入电源与逻辑侧）	1500 VAC	未隔离	未隔离
漏地电流，AC 线路对功能地	最大 0.5 mA	--	--

A.3 CPU 1212C

技术数据	CPU 1212C AC/DC/继电器	CPU 1212C DC/DC/继电器	CPU 1212C DC/DC/DC
保持时间（掉电）	120 VAC 时 20 ms 240 VAC 时 80 ms	24 VDC 时 10 ms	24 VDC 时 10 ms
内部保险丝，用户不可更换	3 A, 250 V, 慢速熔断	3 A, 250 V, 慢速熔断	3 A, 250 V, 慢速熔断

表格 A- 32 传感器电源

技术数据	CPU 1212C AC/DC/继电器	CPU 1212C DC/DC/继电器	CPU 1212C DC/DC/DC
电压范围	20.4 到 28.8 VDC	L+ - 4 VDC（最小） L+ - 5 VDC（最小）（环境温度 -20° C 到 0° C）	
额定输出电流（最大）	300 mA（短路保护）	300 mA（短路保护）	300 mA（短路保护）
最大波纹噪声 (<10 MHz)	< 1 V 峰峰值	与输入线路相同	与输入线路相同
隔离（CPU 逻辑侧与传感器电源）	未隔离	未隔离	未隔离

### A.3.3 数字量输入和输出

表格 A- 33 数字量输入

技术数据	CPU 1212C AC/DC/继电器、DC/DC/继电器和 DC/DC/DC
输入点数	8
类型	漏型/源型 (IEC 1 类漏型)
额定电压	4 mA 时 24 VDC, 额定值
允许的连续电压	最大 30 VDC
浪涌电压	35 VDC, 持续 0.5 s
逻辑 1 信号 (最小)	2.5 mA 时 15 VDC
逻辑 0 信号 (最大)	1 mA 时 5 VDC
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 分钟
隔离组	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)
HSC 时钟输入频率 (最大) (逻辑 1 电平 = 15 到 26 VDC)	单相: 100 KHz (Ia.0 到 Ia.5) 和 30 KHz (Ia.6 到 Ia.7) 正交相位: 80 KHz (Ia.0 到 Ia.5) 和 20 KHz (Ia.6 到 Ia.7)
同时接通的输入数	8, 60° C (水平) 或 50° C (垂直) 时
电缆长度 (米)	500 m (屏蔽); 300 m (非屏蔽); 50 m (屏蔽, HSC 输入)

表格 A- 34 数字量输出

技术数据	CPU 1212C AC/DC/继电器 和 DC/DC/继电器	CPU 1212C DC/DC/DC
输出点数	6	6
类型	继电器, 干触点	固态 - MOSFET (源型)
电压范围	5 到 30 VDC 或 5 到 250 VAC	20.4 到 28.8 VDC
最大电流时的逻辑 1 信号	--	最小 20 VDC
具有 10 K $\Omega$ 负载时的逻辑 0 信号	--	最大 0.1 VDC
电流 (最大)	2.0 A	0.5 A

A.3 CPU 1212C

技术数据	CPU 1212C AC/DC/继电器 和 DC/DC/继电器	CPU 1212C DC/DC/DC
灯负载	30 W DC/200 W AC	5 W
通态电阻	新设备最大为 0.2 Ω	最大 0.6 Ω
每点的漏泄电流	--	最大 10 μA
浪涌电流	触点闭合时为 7 A	8 A, 最长持续 100 ms
过载保护	无	无
隔离（现场侧与逻辑侧）	1500 VAC, 持续 1 分钟（线圈与触点） 无（线圈与逻辑侧）	500 VAC, 持续 1 分钟
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	--
隔离组	2	1
电感钳位电压	--	L+ - 48 VDC, 1 W 损耗
开关延迟（Qa.0 到 Qa.3）	最长 10 ms	断开到接通最长为 1.0 μs 接通到断开最长为 3.0 μs
开关延迟（Qa.4 到 Qa.5）	最长 10 ms	断开到接通最长为 50 μs 接通到断开最长为 200 μs
继电器最大开关频率	1 Hz	--
脉冲串输出频率 （Qa.0 和 Qa.2）	不推荐 <sup>1</sup>	最大 100 KHz, 最小 2 Hz <sup>2</sup>
机械寿命（无负载）	10,000,000 个断开/闭合周期	--
额定负载下的触点寿命	100,000 个断开/闭合周期	--
RUN 到 STOP 时的行为	上一个值或替换值 （默认值为 0）	上一个值或替换值（默认值为 0）
同时接通的输出数	6, 最高 60°C（水平）或 50°C（垂直）时	
电缆长度（米）	500 m（屏蔽）， 150 m（非屏蔽）	500 m（屏蔽）， 150 m（非屏蔽）

<sup>1</sup> 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。

<sup>2</sup> 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。



### A.3.4 模拟量输入

表格 A- 35 模拟量输入

技术数据	说明
输入路数	2
类型	电压（单侧）
满量程范围	0 到 10 V
满量程范围（数据字）	0 到 27648
过冲范围	10.001 到 11.759 V
过冲范围（数据字）	27,649 到 32,511
上溢范围	11.760 到 11.852 V
上溢范围（数据字）	32,512 到 32,767
分辨率	10 位
最大耐压	35 VDC
平滑化	无、弱、中或强 请参见 CPU 模拟量输入的阶跃响应 (ms) (页 782)表格。
噪声抑制	10、50 或 60 Hz
阻抗	≥100 KΩ
隔离（现场侧与逻辑侧）	无
精度（25°C/-20 到 60°C）	满量程的 3.0%/3.5%
电缆长度（米）	100 m，屏蔽双绞线

**A.3.4.1 CPU 内置模拟量输入的阶跃响应**

表格 A- 36 阶跃响应 (ms), 0V 到 10V (在 95% 处测得)

平滑化选项 (采样平均)	抑制频率 (积分时间)		
	60 Hz	50 Hz	10 Hz
无 (1 个周期): 不求平均值	50 ms	50 ms	100 ms
弱 (4 个周期): 4 次采样	60 ms	70 ms	200 ms
中 (16 个周期): 16 次采样	200 ms	240 ms	1150 ms
强 (32 个周期): 32 次采样	400 ms	480 ms	2300 ms
采样时间	<b>4.17 ms</b>	<b>5 ms</b>	<b>25 ms</b>

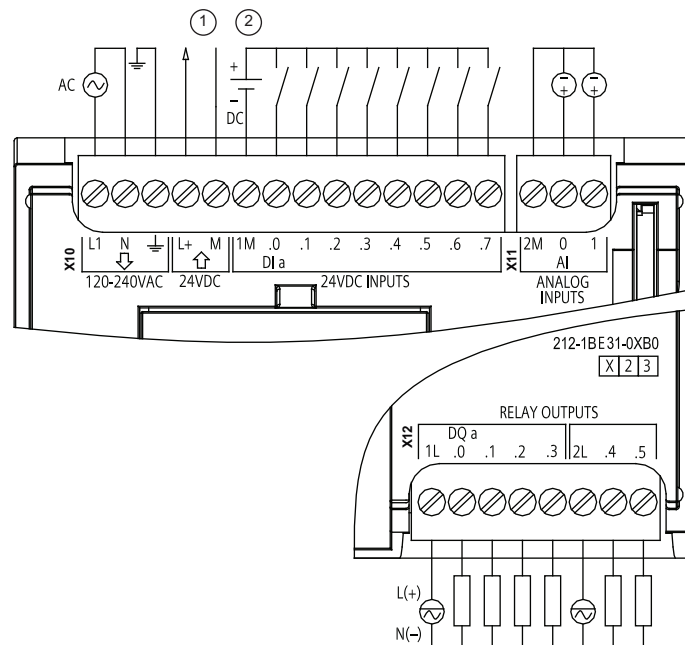
**A.3.4.2 CPU 内置模拟端口的采样时间**

表格 A- 37 CPU 内置模拟量输入的采样时间

抑制频率 (积分时间选项)	采样时间
60 Hz (16.6 ms)	4.17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

## A.3.5 CPU 1212C 的接线图

表格 A-38 CPU 1212C AC/DC/继电器 (6ES7 212-1BE31-0XB0)



- 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。
- 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”。

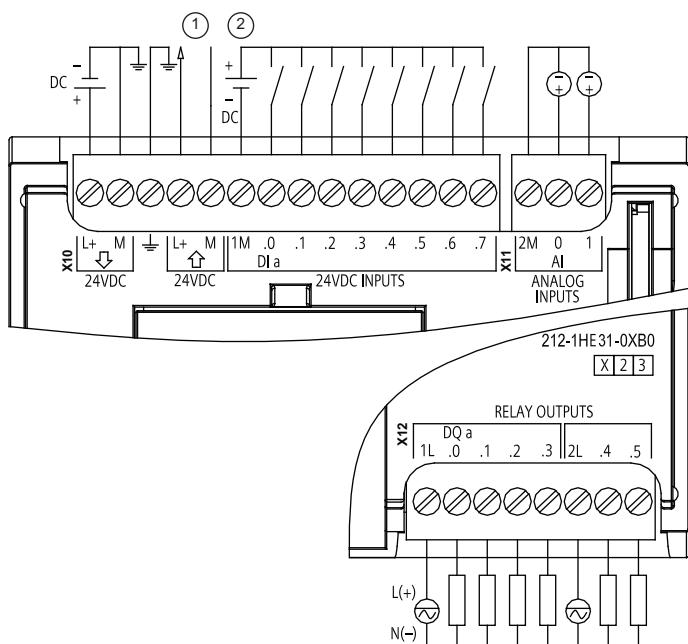
注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A-39 CPU 1212C AC/DC/继电器 (6ES7 212-1BE31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L1/120-240 VAC	2 M	1L
2	N/120-240 VAC	AI 0	DQ a.0
3	功能性接地	AI 1	DQ a.1
4	L+/24VDC 传感器输出	--	DQ a.2
5	M/24VDC 传感器输出	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--

引脚	X10	X11 (镀金)	X12
13	DI a.6	--	--
14	DI a.7	--	--

表格 A- 40 CPU 1212C DC/DC/继电器 (6ES7 212-1HE31-0XB0)



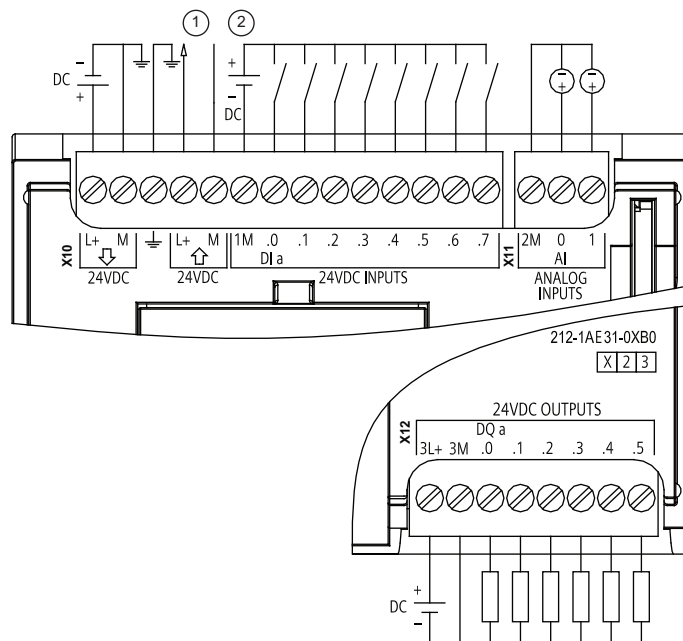
- ① 24 VDC 传感器电源输出  
要获得更好的抗噪声效果，即使未使用传感器电源，也可将“M”连接到机壳接地。
  - ② 对于漏型输入，将“-”连接到“M”（如图所示）。  
对于源型输入，将“+”连接到“M”。
- 注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A- 41 CPU 1212C DC/DC/继电器 (6ES7 212-1BE31-0XB0) 的连接器引脚位置

引脚	X10	X11 (镀金)	X12
1	L+/24VDC	2 M	1L
2	M/24VDC	AI 0	DQ a.0
3	功能性接地	AI 1	DQ a.1
4	L+/24VDC 传感器输出	--	DQ a.2
5	M/24VDC 传感器输出	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--

针脚	X10	X11 (镀金)	X12
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

表格 A- 42 CPU 1212C DC/DC/DC (6ES7 -212-1AE31-0XB0)



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。

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”。

注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A- 43 CPU 1212C DC/DC/DC (6ES7 212-1AE31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L+/24VDC	2 M	3L+
2	M/24VDC	AI 0	3M
3	功能性接地	AI 1	DQ a.0
4	L+/24VDC 传感器输出	--	DQ a.1
5	M/24VDC 传感器输出	--	DQ a.2

A.4 CPU 1214C

引脚	X10	X11 (镀金)	X12
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

说明

应将未使用的模拟量输入短路。

## A.4 CPU 1214C

### A.4.1 常规规范和特性

表格 A-44 常规

技术数据	CPU 1214C AC/DC/继电器	CPU 1214C DC/DC/继电器	CPU 1214C DC/DC/DC
订货号	6ES7 214-1BG31-0XB0	6ES7 214-1HG31-0XB0	6ES7 214-1AG31-0XB0
尺寸 W x H x D (mm)	110 x 100 x 75	110 x 100 x 75	110 x 100 x 75
装运重量	475 g	435 g	415 g
功耗	14 W	12 W	12 W
可用电流 (SM 和 CM 总线)	最大 1600 mA (5 VDC)	最大 1600 mA (5 VDC)	最大 1600 mA (5 VDC)

技术数据	CPU 1214C AC/DC/继电器	CPU 1214C DC/DC/继电器	CPU 1214C DC/DC/DC
可用电流 (24 VDC)	最大 400 mA (传感器电源)	最大 400 mA (传感器电源)	最大 400 mA (传感器电源)
数字量输入电流消耗 (24VDC)	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

表格 A- 45 CPU 特征

技术数据	说明
用户存储器 <sup>1</sup>	75 KB
工作	
负载	内置 4 MB, 可用 SD 卡扩展, 具体视卡容量而定
保持性	10 KB
板载数字 I/O	14 点输入/10 点输出
板载模拟 I/O	2 路输入
过程映像大小	1024 字节输入 (I)/1024 字节输出 (Q)
位存储器 (M)	8192 个字节
临时 (局部) 存储器	<ul style="list-style-type: none"> <li>• 16 KB 用于启动和程序循环 (包括相关的 FB 和 FC)</li> <li>• 4 KB 用于标准中断事件 (包括 FB 和 FC)</li> <li>• 4 KB 用于错误中断事件 (包括 FB 和 FC)</li> </ul>
信号模块扩展	最多 8 个信号模块
SB、CB、BB 扩展	最多 1 个
通信模块扩展	最多 3 个通信模块
高速计数器	总共 6 个, 参见表格 高速计数器的使用方法 (页 365) <ul style="list-style-type: none"> <li>• 单相: 3 个 100 kHz 以及 3 个 30 kHz 的时钟频率</li> <li>• 正交相位: 3 个 80 kHz 以及 3 个 20 kHz 的时钟频率</li> </ul>
脉冲输出 <sup>2</sup>	4
脉冲捕捉输入	14
延时中断/循环中断	共 4 个, 精度为 1 ms
沿中断	12 个上升沿和 12 个下降沿 (使用可选信号板时, 各为 14 个)
存储卡	SIMATIC 存储卡 (选件)

A.4 CPU 1214C

技术数据	说明
实时时钟精度	+/- 60 秒/月
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）

- 1 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的限制。对支持的 OB、FC、FB 和 DB 块数目或特定块的大小没有特殊限制；唯一的限制是存储器的总大小。
- 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- 46 性能

指令类型	执行速度
布尔运算	0.08 μs/指令
移动字	1.7 μs/指令
实数数学运算	2.3 μs/指令

A.4.2 CPU 1214C 支持的定时器、计数器和代码块

表格 A- 47 CPU 1214C 支持的块、定时器和计数器

元素	说明	
块	类型	OB、FB、FC、DB
	大小	64 KB
	数量	一共多达 1024 个块 (OB + FB + FC + DB)
	FB、FC 和 DB 的地址范围	1 到 65535（例如 FB 1 到 FB 65535）
	嵌套深度	16（从程序循环或启动 OB 开始）；4（从延时中断、日时钟中断、循环中断、硬件中断、时间错误中断或诊断错误中断 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：OB 1、OB 200 到 OB 65535
	启动	多个：OB 100、OB 200 到 OB 65535
	延时中断和循环中断	4 <sup>1</sup> （每个事件 1 个）：OB 200 到 OB 65535
	硬件中断（沿和 HSC）	50（每个事件 1 个）：OB 200 到 OB 65535



元素	说明
	时间错误中断
	1: OB 80
	诊断错误中断
	1: OB 82
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
	存储
	DB 结构，大小取决于计数类型
	<ul style="list-style-type: none"> <li>• SInt 和 USInt: 3 个字节</li> <li>• Int 和 UInt: 6 个字节</li> <li>• DInt 和 UDInt: 12 个字节</li> </ul>

<sup>1</sup> 延时中断和循环中断在 CPU 中使用相同的资源。延时中断和循环中断的总和只能为 4 个。不能有 4 个延时中断和 4 个循环中断。

表格 A- 48 通信

技术数据	说明
端口数	1
类型	以太网
HMI 设备 <sup>1</sup>	3
编程设备 (PG)	1
连接	<ul style="list-style-type: none"> <li>• 8 个用于开放式用户通信（主动或被动）：TSEND_C、TRCV_C、TCON、TDISCON、TSEND 和 TRCV</li> <li>• 3 个用于服务器 GET/PUT（CPU 间）S7 通信</li> <li>• 8 个用于客户端 GET/PUT（CPU 间）S7 通信</li> </ul>
数据传输率	10/100 Mb/s
隔离（外部信号与 PLC 逻辑侧）	变压器隔离，1500 VAC（仅针对短期事件安全）
电缆类型	CAT5e 屏蔽电缆

<sup>1</sup> CPU 提供专用的 HMI 连接，以支持最多 3 个 HMI 设备。（最多可以有 2 个 SIMATIC 精智面板。）支持的 HMI 总数受组态中 HMI 面板类型的影响。例如，可以将最多 3 个 SIMATIC 基本面板连接到 CPU，或者最多可以连接两个 SIMATIC 精智面板与一个附加基本面板。

A.4 CPU 1214C

表格 A- 49 电源

技术数据	CPU 1214C AC/DC/继电器	CPU 1214C DC/DC/继电器	CPU 1214C DC/DC/DC
电压范围	85 到 264 VAC	20.4 VDC 到 28.8 VDC 22.0 VDC 到 28.8 VDC (环境温度 -20° C 到 0° C)	
电源频率	47 到 63 Hz	--	
输入电流 (最大负载时)	仅 CPU 具有所有扩展附件的 CPU	120 VAC 时 100 mA 240 VAC 时 50 mA 120 VAC 时 300 mA 240 VAC 时 150 mA	24 VDC 时 500 mA 24 VDC 时 1500 mA
浪涌电流 (最大)	264 VAC 时 20 A	28.8 VDC 时 12 A	
隔离 (输入电源与逻辑侧)	1500 VAC	未隔离	
漏地电流, AC 线路对功能地	最大 0.5 mA	-	
保持时间 (掉电)	120 VAC 时 20 ms 240 VAC 时 80 ms	24 VDC 时 10 ms	
内部保险丝, 用户不可更换	3 A, 250 V, 慢速熔断		

表格 A- 50 传感器电源

技术数据	CPU 1214C AC/DC/继电器	CPU 1214C DC/DC/继电器	CPU 1214C DC/DC/DC
电压范围	20.4 到 28.8 VDC	L+ - 4 VDC (最小) L+ - 5 VDC (最小) (对于环境温度 - 20° C 到 0° C)	
额定输出电流 (最大)	400 mA (短路保护)		
最大波纹噪声 (<10 MHz)	< 1 V 峰峰值	与输入线路相同	
隔离 (CPU 逻辑侧与传感器电源)	未隔离		

### A.4.3 数字量输入和输出

表格 A- 51 数字量输入

技术数据	CPU 1214C AC/DC/继电器	CPU 1214C DC/DC/继电器	CPU 1214C DC/DC/DC
输入点数	14		
类型	漏型/源型 (IEC 1 类漏型)		
额定电压	4 mA 时 24 VDC, 额定值		
允许的连续电压	最大 30 VDC		
浪涌电压	35 VDC, 持续 0.5 s		
逻辑 1 信号 (最小)	2.5 mA 时 15 VDC		
逻辑 0 信号 (最大)	1 mA 时 5 VDC		
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 分钟		
隔离组	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)		
HSC 时钟输入频率 (最大) (逻辑 1 电平 = 15 到 26 VDC)	单相: 100 KHz (Ia.0 到 Ia.5) 和 30 KHz (Ia.6 到 Ib.5) 正交相位: 80 KHz (Ia.0 到 Ia.5) 和 20 KHz (Ia.6 到 Ib.5)		
同时接通的输入数	<ul style="list-style-type: none"> <li>• 7 (无相邻点), 60° C (水平) 或 50° C (垂直) 时</li> <li>• 14, 55° C (水平) 或 45° C (垂直) 时</li> </ul>		
电缆长度 (米)	500 m (屏蔽); 300 m (非屏蔽); 50 m (屏蔽, HSC 输入)		

表格 A- 52 数字量输出

技术数据	CPU 1214C AC/DC/继电器 和 DC/DC/继电器	CPU 1214C DC/DC/DC
输出点数	10	10
类型	继电器, 干触点	固态 - MOSFET (源型)
电压范围	5 到 30 VDC 或 5 到 250 VAC	20.4 到 28.8 VDC
最大电流时的逻辑 1 信号	--	最小 20 VDC
具有 10 K $\Omega$ 负载时的逻辑 0 信号	--	最大 0.1 VDC

A.4 CPU 1214C

技术数据	CPU 1214C AC/DC/继电器 和 DC/DC/继电器	CPU 1214C DC/DC/DC
电流（最大）	2.0 A	0.5 A
灯负载	30 W DC/200 W AC	5 W
通态电阻	新设备最大为 0.2 Ω	最大 0.6 Ω
每点的漏泄电流	--	最大 10 μA
浪涌电流	触点闭合时为 7 A	8 A, 最长持续 100 ms
过载保护	无	无
隔离（现场侧与逻辑侧）	1500 VAC, 持续 1 分钟（线圈与触点） 无（线圈与逻辑侧）	500 VAC, 持续 1 分钟
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	--
隔离组	2	1
电感钳位电压	--	L+ 减去 48 VDC, 1 W 损耗
开关延迟（Qa.0 到 Qa.3）	最长 10 ms	断开到接通最长为 1.0 μs 接通到断开最长为 3.0 μs
开关延迟（Qa.4 到 Qb.1）	最长 10 ms	断开到接通最长为 50 μs 接通到断开最长为 200 μs
继电器最大开关频率	1 Hz	--
脉冲串输出频率 （Qa.0 和 Qa.2）	不推荐 <sup>1</sup>	最大 100 KHz, 最小 2 Hz <sup>2</sup>
机械寿命（无负载）	10,000,000 个断开/闭合周期	--
额定负载下的触点寿命	100,000 个断开/闭合周期	--
RUN 到 STOP 时的行为	上一个值或替换值（默认值为 0）	

技术数据	CPU 1214C AC/DC/继电器 和 DC/DC/继电器	CPU 1214C DC/DC/DC
同时接通的输出数	<ul style="list-style-type: none"> <li>• 5（无相邻点），60° C（水平）或 50° C（垂直）时</li> <li>• 10，55° C（水平）或 45° C（垂直）时</li> </ul>	
电缆长度（米）	500 m（屏蔽）； 150 m（非屏蔽）	

- 1 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。
- 2 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。

#### A.4.4 模拟量输入

表格 A- 53 模拟量输入

技术数据	说明
输入路数	2
类型	电压（单侧）
满量程范围	0 到 10 V
满量程范围（数据字）	0 到 27648
过冲范围	10.001 到 11.759 V
过冲范围（数据字）	27,649 到 32,511
上溢范围	11.760 到 11.852 V
上溢范围（数据字）	32,512 到 32,767
分辨率	10 位
最大耐压	35 VDC
平滑化	无、弱、中或强 请参见 CPU 模拟量输入的阶跃响应 (ms) (页 794)表格。
噪声抑制	10、50 或 60 Hz
阻抗	≥100 KΩ
隔离（现场侧与逻辑侧）	无
精度（25°C/-20 到 60°C）	满量程的 3.0%/3.5%
电缆长度（米）	100 m，屏蔽双绞线

**A.4.4.1 CPU 内置模拟量输入的阶跃响应**

表格 A- 54 阶跃响应 (ms), 0V 到 10V (在 95% 处测得)

平滑化选项 (采样平均)	抑制频率 (积分时间)		
	60 Hz	50 Hz	10 Hz
无 (1 个周期): 不求平均值	50 ms	50 ms	100 ms
弱 (4 个周期): 4 次采样	60 ms	70 ms	200 ms
中 (16 个周期): 16 次采样	200 ms	240 ms	1150 ms
强 (32 个周期): 32 次采样	400 ms	480 ms	2300 ms
采样时间	<b>4.17 ms</b>	<b>5 ms</b>	<b>25 ms</b>

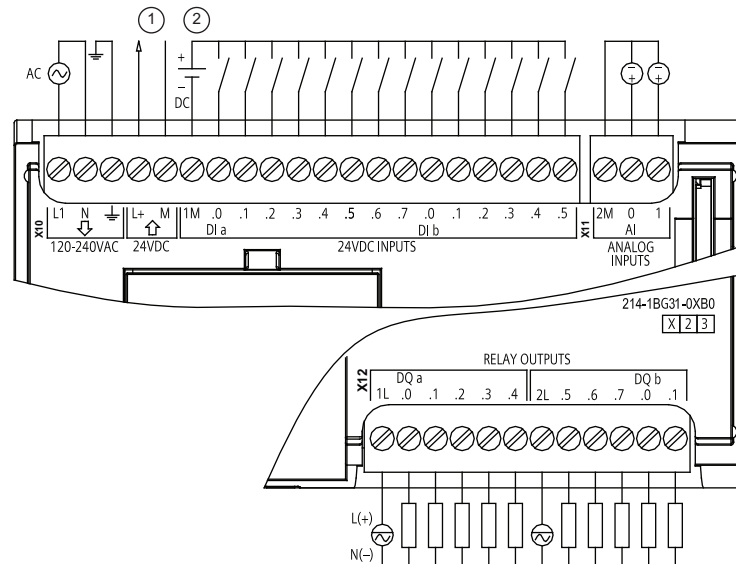
**A.4.4.2 CPU 内置模拟端口的采样时间**

表格 A- 55 CPU 内置模拟量输入的采样时间

抑制频率 (积分时间选项)	采样时间
60 Hz (16.6 ms)	4.17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

## A.4.5 CPU 1214C 的接线图

表格 A-56 CPU 1214C AC/DC/继电器 (6ES7 214-1BG31-0XB0)



## 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。

## 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”。

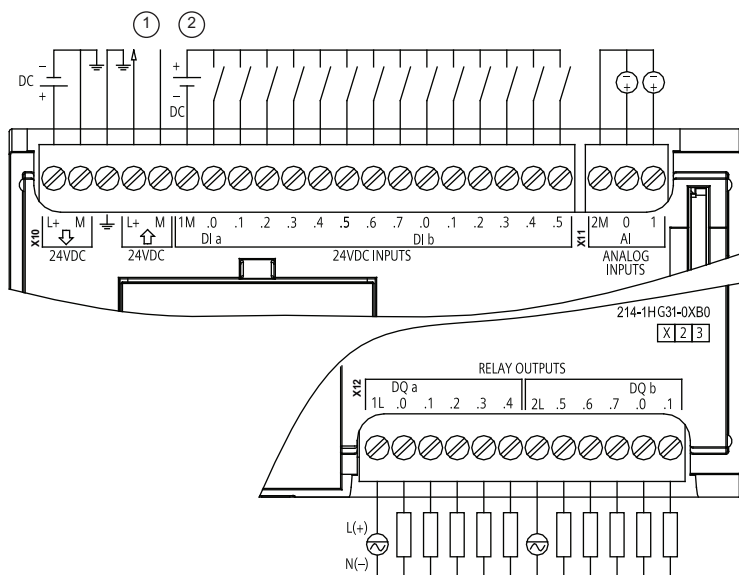
注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A-57 CPU 1214C AC/DC/继电器 (6ES7 214-1BG31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L1/120-240 VAC	2 M	1L
2	N/120-240 VAC	AI 0	DQ a.0
3	功能性接地	AI 1	DQ a.1
4	L+/24VDC 传感器输出	--	DQ a.2
5	M/24VDC 传感器输出	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1

针脚	X10	X11 (镀金)	X12
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

表格 A- 58 CPU 1214C DC/DC/继电器 (6ES7 214-1HG31-0XB0)



- 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。
- 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”。

注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

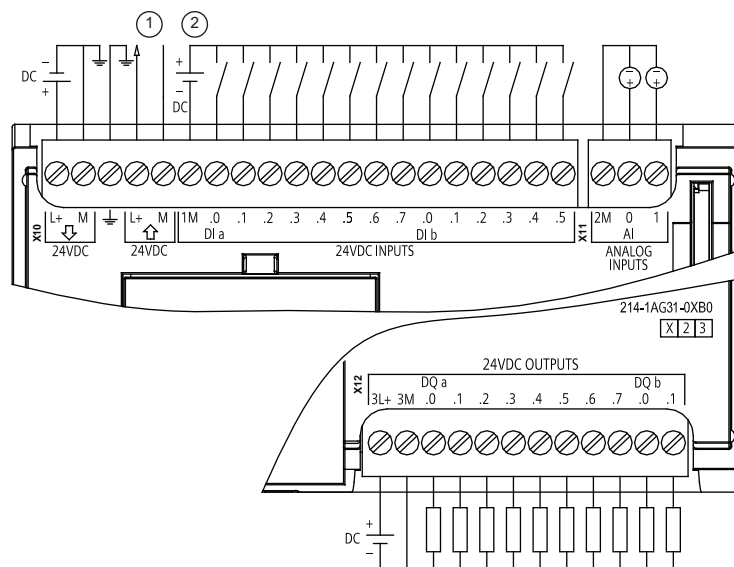
表格 A- 59 CPU 1214C DC/DC/继电器 (6ES7 214-1HG31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L+/24VDC	2 M	1L
2	M/24VDC	AI 0	DQ a.0
3	功能性接地	AI 1	DQ a.1



引脚	X10	X11 (镀金)	X12
4	L+/24VDC 传感器输出	--	DQ a.2
5	M/24VDC 传感器输出	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

表格 A- 60 CPU 1214C DC/DC/DC (6ES7 214-1AG31-0XB0)



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。

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”。

注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A- 61 CPU 1214C DC/DC/DC (6ES7 214-1AG31-0XB0) 的连接器引脚位置

引脚	X10	X11 (镀金)	X12
1	L+/24VDC	2 M	3L+
2	M/24VDC	AI 0	3M
3	功能性接地	AI 1	DQ a.0
4	L+/24VDC 传感器输出	--	DQ a.1
5	M/24VDC 传感器输出	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	-
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

**说明**

应将未使用的模拟量输入短路。

## A.5 CPU 1215C

### A.5.1 常规规范和特性

表格 A- 62 常规

技术数据	CPU 1215C AC/DC/继电器	CPU 1215C DC/DC/继电器	CPU 1215C DC/DC/DC
订货号	6ES7 215-1BG31-0XB0	6ES7 215-1HG31-0XB0	6ES7 215-1AG31-0XB0
尺寸 W x H x D (mm)	130 x 100 x 75	130 x 100 x 75	130 x 100 x 75
装运重量	550 g	585 g	520 g
功耗	14 W	12 W	12 W
可用电流 (SM 和 CM 总线)	最大 1600 mA (5 VDC)	最大 1600 mA (5 VDC)	最大 1600 mA (5 VDC)
可用电流 (24 VDC)	最大 400 mA (传感器电源)	最大 400 mA (传感器电源)	最大 400 mA (传感器电源)
数字量输入电流消耗 (24VDC)	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

表格 A- 63 CPU 特征

技术数据	说明
用户存储器 <sup>1</sup>	工作 100 KB
	负载 内置 4 MB, 可用 SD 卡扩展, 具体视卡容量而定
	保持性 10 KB
板载数字 I/O	14 点输入/10 点输出
板载模拟 I/O	2 点输入/2 点输出
过程映像大小	1024 字节输入 (I)/1024 字节输出 (Q)
位存储器 (M)	8192 个字节
临时 (局部) 存储器	<ul style="list-style-type: none"> <li>• 16 KB 用于启动和程序循环 (包括相关的 FB 和 FC)</li> <li>• 4 KB 用于标准中断事件 (包括 FB 和 FC)</li> <li>• 4 KB 用于错误中断事件 (包括 FB 和 FC)</li> </ul>

技术数据	说明
信号模块扩展	最多 8 个信号模块
SB、CB、BB 扩展	最多 1 个
通信模块扩展	最多 3 个通信模块
高速计数器	总共 6 个，请参见表格 CPU 1215C 的 HSC 输入分配 <ul style="list-style-type: none"> <li>• 单相：3 个 100 kHz 以及 3 个 30 kHz 的时钟频率</li> <li>• 正交相位：3 个 80 kHz 以及 3 个 20 kHz 的时钟频率</li> </ul>
脉冲输出 <sup>2</sup>	4
脉冲捕捉输入	14
延时中断/循环中断	共 4 个，精度为 1 ms
沿中断	12 个上升沿和 12 个下降沿（使用可选信号板时，各为 14 个）
存储卡	SIMATIC 存储卡（选件）
实时时钟精度	+/- 60 秒/月
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）

1 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的限制。对支持的 OB、FC、FB 和 DB 块数目或特定块的大小没有特殊限制；唯一的限制是存储器的总大小。

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-64 性能

指令类型	执行速度
布尔运算	0.08 μs/指令
移动字	1.7 μs/指令
实数数学运算	2.3 μs/指令

## A.5.2 CPU 1215C 支持的定时器、计数器和代码块

表格 A- 65 CPU 1215C 支持的块、定时器和计数器

元素	说明	
块	类型	OB、FB、FC、DB
	大小	64 KB
	数量	一共多达 1024 个块 (OB + FB + FC + DB)
	FB、FC 和 DB 的地址范围	1 到 65535 (例如 FB 1 到 FB 65535)
	嵌套深度	16 (从程序循环或启动 OB 开始)；4 (从延时中断、日时钟中断、循环中断、硬件中断、时间错误中断或诊断错误中断 OB 开始)
	监视	可以同时监视 2 个代码块的状态
	OB	程序循环
启动		多个：OB 100、OB 200 到 OB 65535
延时中断和循环中断		4 <sup>1</sup> (每个事件 1 个)：OB 200 到 OB 65535
硬件中断 (沿和 HSC)		50 (每个事件 1 个)：OB 200 到 OB 65535
时间错误中断		1: OB 80
诊断错误中断		1: OB 82
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
	存储	DB 结构，大小取决于计数类型 <ul style="list-style-type: none"> <li>• SInt 和 USInt: 3 个字节</li> <li>• Int 和 UInt: 6 个字节</li> <li>• DInt 和 UDInt: 12 个字节</li> </ul>

<sup>1</sup> 延时中断和循环中断在 CPU 中使用相同的资源。延时中断和循环中断的总和只能为 4 个。不能有 4 个延时中断和 4 个循环中断。

A.5 CPU 1215C

表格 A- 66 通信

技术数据	说明
端口数	2
类型	以太网
HMI 设备 <sup>1</sup>	3
编程设备 (PG)	1
连接	<ul style="list-style-type: none"> <li>• 8 个用于开放式用户通信（主动或被动）： TSEND_C、TRCV_C、TCON、TDISCON、TSEND 和 TRCV</li> <li>• 3 个用于服务器 GET/PUT（CPU 间）S7 通信</li> <li>• 8 个用于客户端 GET/PUT（CPU 间）S7 通信</li> </ul>
数据传输率	10/100 Mb/s
隔离（外部信号与 PLC 逻辑侧）	变压器隔离，1500 VAC（仅针对短期事件安全）
电缆类型	CAT5e 屏蔽电缆

<sup>1</sup> CPU 提供专用的 HMI 连接，以支持最多 3 个 HMI 设备。（最多可以有 2 个 SIMATIC 精智面板。）支持的 HMI 总数受组态中 HMI 面板类型的影响。例如，可以将最多 3 个 SIMATIC 基本面板连接到 CPU，或者最多可以连接两个 SIMATIC 精智面板与一个附加基本面板。

表格 A- 67 电源

技术数据	CPU 1215C AC/DC/继电器	CPU 1215C DC/DC/继电器	CPU 1215C DC/DC/DC
电压范围	85 到 264 VAC	20.4 VDC 到 28.8 VDC 22.0 VDC 到 28.8 VDC（环境温度 -20° C 到 0° C）	
电源频率	47 到 63 Hz	--	
输入电流 （最大负载 时）	仅 CPU 具有所有扩展附件的 CPU	120 VAC 时 100 mA 240 VAC 时 50 mA 120 VAC 时 300 mA 240 VAC 时 150 mA	24 VDC 时 500 mA 24 VDC 时 1500 mA
浪涌电流（最大）	264 VAC 时 20 A	28.8 VDC 时 12 A	
隔离（输入电源与逻辑侧）	1500 VAC	未隔离	
漏地电流，AC 线路对功能地	最大 0.5 mA	-	

技术数据	CPU 1215C AC/DC/继电器	CPU 1215C DC/DC/继电器	CPU 1215C DC/DC/DC
保持时间（掉电）	120 VAC 时 20 ms 240 VAC 时 80 ms	24 VDC 时 10 ms	
内部保险丝，用户不可更换	3 A, 250 V, 慢速熔断		

表格 A- 68 传感器电源

技术数据	CPU 1215C AC/DC/继电器	CPU 1215C DC/DC/继电器	CPU 1215C DC/DC/DC
电压范围	20.4 到 28.8 VDC	L+ - 4 VDC（最小） L+ - 5 VDC（最小）（对于环境温度 - 20° C 到 0° C）	
额定输出电流（最大）	400 mA（短路保护）		
最大波纹噪声 (<10 MHz)	< 1 V 峰峰值	与输入线路相同	
隔离（CPU 逻辑侧与传感器电源）	未隔离		

### A.5.3 数字量输入和输出

表格 A- 69 数字量输入

技术数据	CPU 1215C AC/DC/继电器	CPU 1215C DC/DC/继电器	CPU 1215C DC/DC/DC
输入点数	14		
类型	漏型/源型（IEC 1 类漏型）		
额定电压	4 mA 时 24 VDC, 额定值		
允许的连续电压	最大 30 VDC		
浪涌电压	35 VDC, 持续 0.5 s		
逻辑 1 信号（最小）	2.5 mA 时 15 VDC		
逻辑 0 信号（最大）	1 mA 时 5 VDC		
隔离（现场侧与逻辑侧）	500 VAC, 持续 1 分钟		
隔离组	1		

A.5 CPU 1215C

技术数据	CPU 1215C AC/DC/继电器	CPU 1215C DC/DC/继电器	CPU 1215C DC/DC/DC
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）		
HSC 时钟输入频率（最大） （逻辑 1 电平 = 15 到 26 VDC）	单相：100 KHz（Ia.0 到 Ia.5）和 30 KHz（Ia.6 到 Ib.5） 正交相位：80 KHz（Ia.0 到 Ia.5）和 20 KHz（Ia.6 到 Ib.5）		
同时接通的输入数	<ul style="list-style-type: none"> <li>• 7（无相邻点），60° C（水平）或 50° C（垂直）时</li> <li>• 14，55° C（水平）或 45° C（垂直）时</li> </ul>		
电缆长度（米）	500 m（屏蔽）；300 m（非屏蔽）；50 m（屏蔽，HSC 输入）		

表格 A-70 数字量输出

技术数据	CPU 1212C AC/DC/继电器 和 CPU 1215C DC/DC/继电器	CPU 1215C DC/DC/DC
输出点数	10	10
类型	继电器，干触点	固态 - MOSFET（源型）
电压范围	5 到 30 VDC 或 5 到 250 VAC	20.4 到 28.8 VDC
最大电流时的逻辑 1 信号	--	最小 20 VDC
具有 10 K $\Omega$ 负载时的逻辑 0 信号	--	最大 0.1 VDC
电流（最大）	2.0 A	0.5 A
灯负载	30 W DC/200 W AC	5 W
通态电阻	新设备最大为 0.2 $\Omega$	最大 0.6 $\Omega$
每点的漏泄电流	--	最大 10 $\mu$ A
浪涌电流	触点闭合时为 7 A	8 A，最长持续 100 ms
过载保护	无	无
隔离（现场侧与逻辑侧）	1500 VAC，持续 1 分钟（线圈与触点） 无（线圈与逻辑侧）	500 VAC，持续 1 分钟
隔离电阻	新设备最小为 100 M $\Omega$	--
断开触点间的绝缘	750 VAC，持续 1 分钟	--
隔离组	2	1



技术数据	CPU 1212C AC/DC/继电器 和 CPU 1215C DC/DC/继电器	CPU 1215C DC/DC/DC
电感钳位电压	--	L+ 减去 48 VDC, 1 W 损耗
开关延迟 (Qa.0 到 Qa.3)	最长 10 ms	断开到接通最长为 1.0 $\mu$ s 接通到断开最长为 3.0 $\mu$ s
开关延迟 (Qa.4 到 Qb.1)	最长 10 ms	断开到接通最长为 50 $\mu$ s 接通到断开最长为 200 $\mu$ s
继电器最大开关频率	1 Hz	--
脉冲串输出频率 (Qa.0 和 Qa.2)	不推荐 <sup>1</sup>	最大 100 KHz, 最小 2 Hz <sup>2</sup>
机械寿命 (无负载)	10,000,000 个断开/闭合周期	--
额定负载下的触点寿命	100,000 个断开/闭合周期	--
RUN 到 STOP 时的行为	上一个值或替换值 (默认值为 0)	
同时接通的输出数	<ul style="list-style-type: none"> <li>• 5 (无相邻点), 60° C (水平) 或 50° C (垂直) 时</li> <li>• 10, 55° C (水平) 或 45° C (垂直) 时</li> </ul>	
电缆长度 (米)	500 m (屏蔽); 150 m (非屏蔽)	

<sup>1</sup> 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。

<sup>2</sup> 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。

## A.5.4 模拟量输入和输出

### A.5.4.1 模拟量输入规范

表格 A- 71 模拟量输入

技术数据	说明
输入路数	2
类型	电压 (单侧)
满量程范围	0 到 10 V
满量程范围 (数据字)	0 到 27648

A.5 CPU 1215C

技术数据	说明
过冲范围	10.001 到 11.759 V
过冲范围 (数据字)	27,649 到 32,511
上溢范围	11.760 到 11.852 V
上溢范围 (数据字)	32,512 到 32,767
分辨率	10 位
最大耐压	35 VDC
平滑化	无、弱、中或强 请参见 CPU 模拟量输入的阶跃响应 (ms) 表格。
噪声抑制	10、50 或 60 Hz
阻抗	≥100 KΩ
隔离 (现场侧与逻辑侧)	无
精度 (25°C/-20 到 60°C)	满量程的 3.0%/3.5%
电缆长度 (米)	100 m, 屏蔽双绞线

A.5.4.2 CPU 内置模拟量输入的阶跃响应

表格 A-72 阶跃响应 (ms), 0V 到 10V (在 95% 处测得)

平滑化选项 (采样平均)	抑制频率 (积分时间)		
	60 Hz	50 Hz	10 Hz
无 (1 个周期): 不求平均值	50 ms	50 ms	100 ms
弱 (4 个周期): 4 次采样	60 ms	70 ms	200 ms
中 (16 个周期): 16 次采样	200 ms	240 ms	1150 ms
强 (32 个周期): 32 次采样	400 ms	480 ms	2300 ms
采样时间	4.17 ms	5 ms	25 ms

**A.5.4.3 CPU 内置模拟端口的采样时间**

表格 A-73 CPU 内置模拟量输入的采样时间

抑制频率（积分时间选项）	采样时间
60 Hz (16.6 ms)	4.17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

**A.5.4.4 模拟量输出规格****模拟量输出**

表格 A-74 模拟量输出

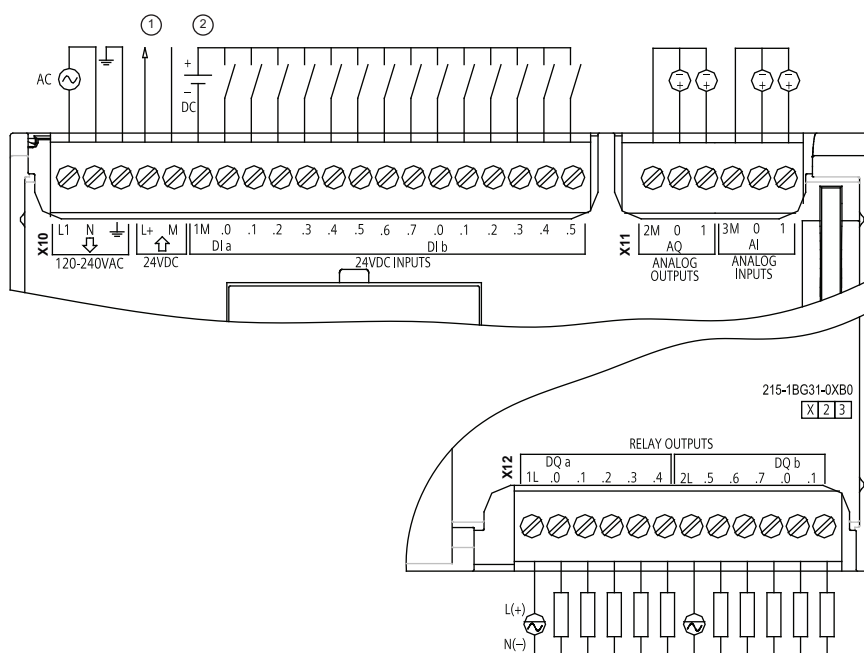
技术数据	说明
输出点数	2
类型	电流
满量程范围	0 到 20 mA
满量程范围（数据字）	0 到 27648
过冲范围	20.01 到 23.52 mA <sup>1</sup>
过冲范围（数据字）	27,649 到 32,511
上溢范围	参见脚注 <sup>2</sup>
上溢范围数据字	32,512 到 32,767
分辨率	10 位
输出驱动阻抗	最大 500 Ω
隔离（现场侧与逻辑侧）	无
精度（25°C/-20 到 60°C）	满量程的 3.0%/3.5%

技术数据	说明
稳定时间	2 ms
电缆长度 (米)	100 m, 屏蔽双绞线

- 1 对于使用 DC 电源的 CPU 1215C: 电源电压为 20.4 VDC 时, 过冲范围中支持最高 400 Ω 的输出驱动阻抗。
- 2 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.5.5 CPU 1215C 的接线图

表格 A- 75 CPU 1215C AC/DC/继电器 (6ES7 215-1BG31-0XB0)



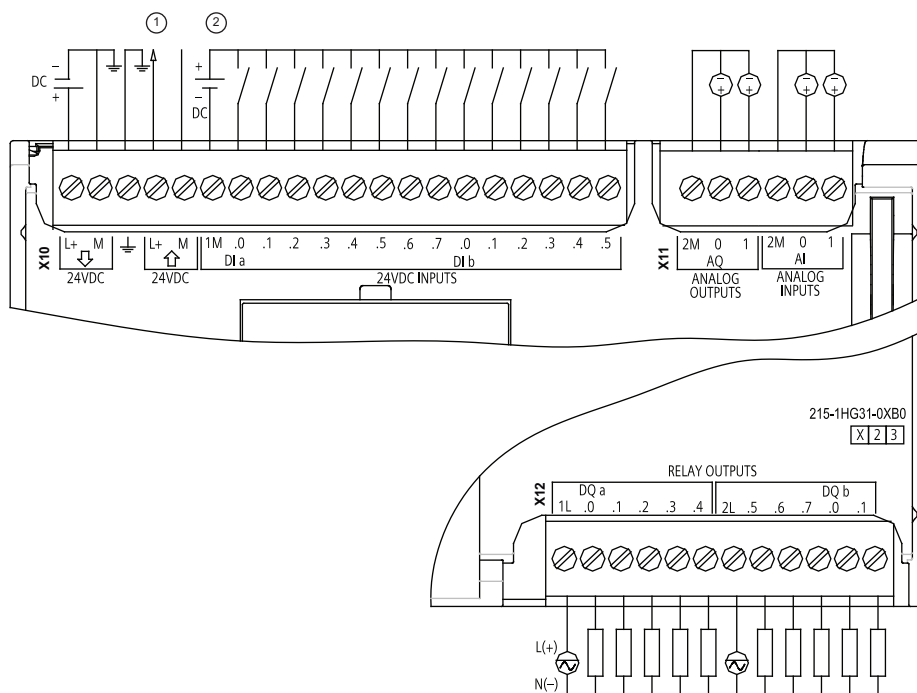
- ① 24 VDC 传感器电源输出  
要获得更好的抗噪声效果, 即使未使用传感器电源, 也可将“M”连接到机壳接地。
- ② 对于漏型输入, 将“-”连接到“M”(如图所示)。  
对于源型输入, 将“+”连接到“M”。

注: X11 连接器必须镀金。有关订货号, 请参见附录 C“备件”。

表格 A- 76 CPU 1215C AC/DC/继电器 (6ES7 215-1BG31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L1/120-240 VAC	2 M	1L
2	N/120-240 VAC	AQ 0	DQ a.0
3	功能性接地	AQ 1	DQ a.1
4	L+/24VDC 传感器输出	3M	DQ a.2
5	M/24VDC 传感器输出	AI 0	DQ a.3
6	1M	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

表格 A-77 CPU 1215C DC/DC/继电器 (6ES7 215-1HG31-0XB0)



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。

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”。

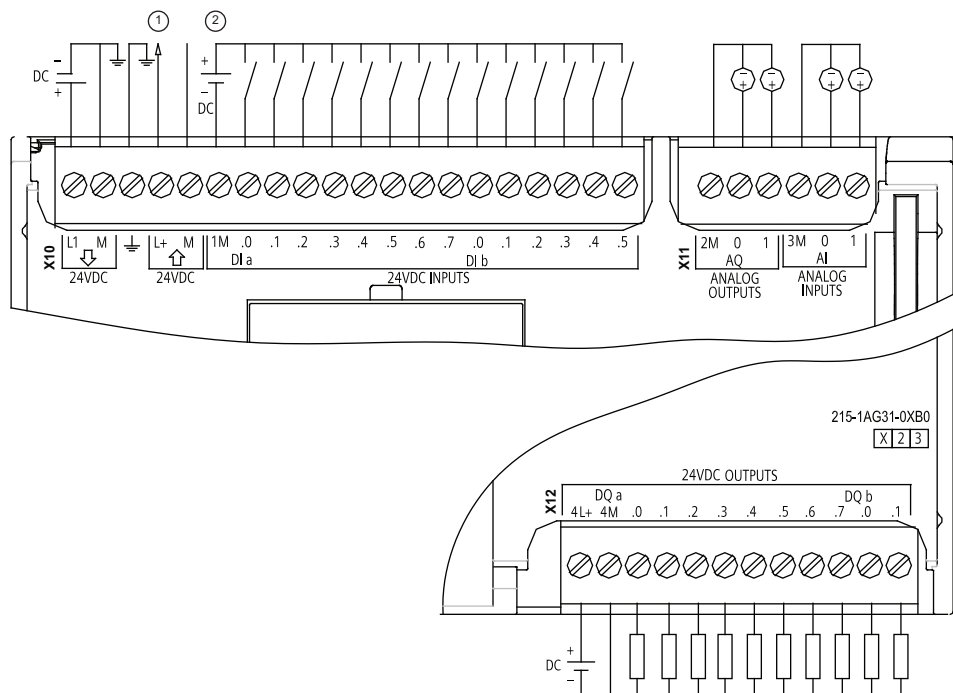
注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A-78 CPU 1215C DC/DC/继电器 (6ES7 215-1HG31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L+/24VDC	2 M	1L
2	M/24VDC	AQ 0	DQ a.0
3	功能性接地	AQ 1	DQ a.1
4	L+/24VDC 传感器输出	2M	DQ a.2
5	M/24VDC 传感器输出	AI 0	DQ a.3
6	1M	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7

针脚	X10	X11 (镀金)	X12
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

表格 A- 79 CPU 1215C DC/DC/DC (6ES7 215-1AG31-0XB0)



## 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。

## 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”。

注：X11 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A- 80 CPU 1215C DC/DC/DC (6ES7 215-1AG31-0XB0) 的连接器针脚位置

针脚	X10	X11 (镀金)	X12
1	L1/24VDC	2 M	4L+
2	M/24VDC	AQ 0	4M
3	功能性接地	AQ 1	DQ a.0
4	L+/24VDC 传感器输出	3M	DQ a.1
5	M/24VDC 传感器输出	AI 0	DQ a.2
6	1M	AI 1	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

**说明**

应将未使用的模拟量输入短路。



## A.6 数字信号模块 (SM)

### A.6.1 SM 1221 数字量输入规范

表格 A- 81 常规规范

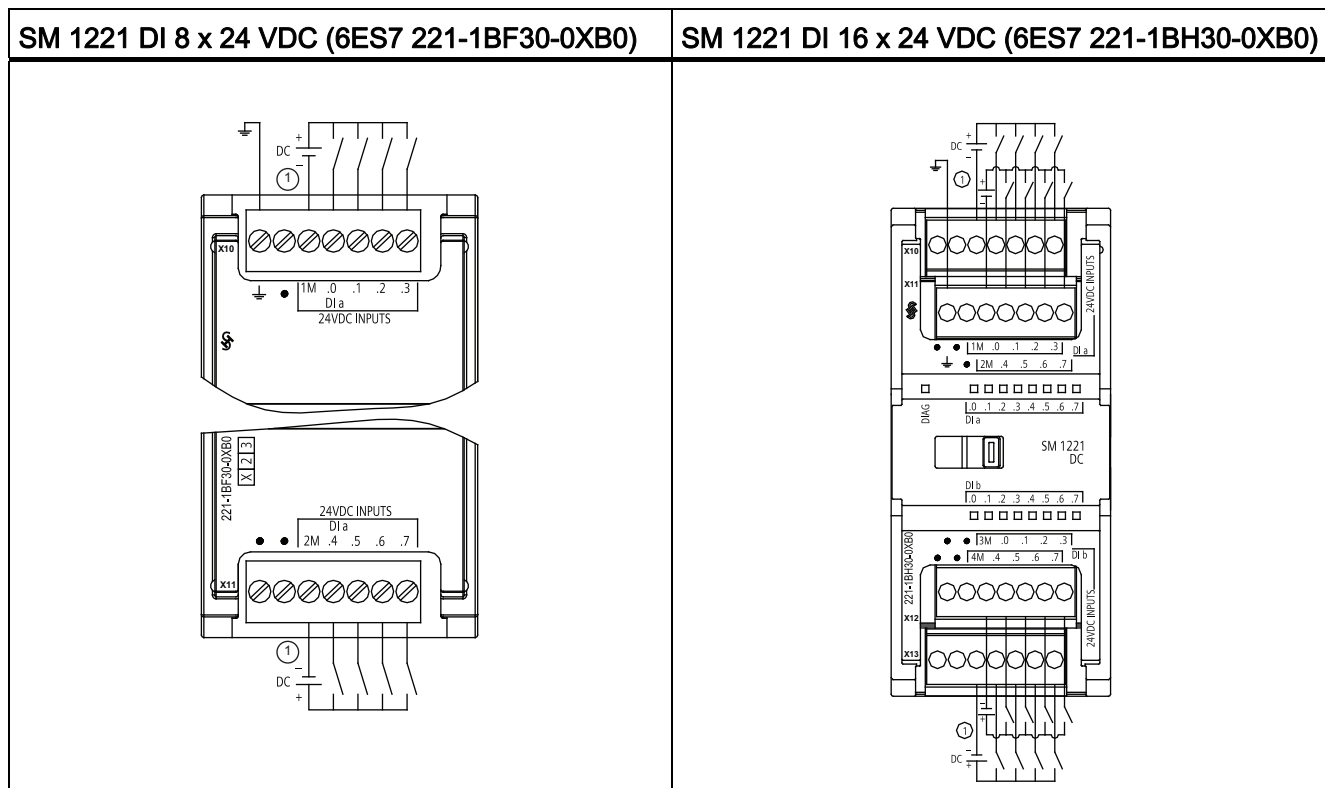
型号	SM 1221 DI 8 x 24 VDC	SM 1221 DI 16 x 24 VDC
订货号	6ES7 221-1BF30-0XB0	6ES7 221-1BH30-0XB0
尺寸 W x H x D (mm)	45 x 100 x 75	45 x 100 x 75
重量	170 g	210 g
功耗	1.5 W	2.5 W
电流消耗 (SM 总线)	105 mA	130 mA
电流消耗 (24 VDC)	所用的每点输入 4 mA	所用的每点输入 4 mA

表格 A- 82 数字量输入

型号	SM 1221 DI 8 x 24 VDC	SM 1221 DI 16 x 24 VDC
输入点数	8	16
类型	漏型/源型 (IEC 1 类漏型)	漏型/源型 (IEC 1 类漏型)
额定电压	4 mA 时 24 VDC, 额定值	4 mA 时 24 VDC, 额定值
允许的连续电压	最大 30 VDC	最大 30 VDC
浪涌电压	35 VDC, 持续 0.5 s	35 VDC, 持续 0.5 s
逻辑 1 信号 (最小)	2.5 mA 时 15 VDC	2.5 mA 时 15 VDC
逻辑 0 信号 (最大)	1 mA 时 5 VDC	1 mA 时 5 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	2	4
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)
同时接通的输入数	8	16
电缆长度 (米)	500 m (屏蔽); 300 m (非屏蔽)	500 m (屏蔽); 300 m (非屏蔽)

A.6 数字信号模块 (SM)

表格 A- 83 数字量输入 SM 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”。

表格 A- 84 SM 1221 DI 8 x 24 VDC (6ES7 221-1BF30-0XB0) 的连接器的引脚位置

引脚	X10	X11
1	GND	无连接
2	无连接	无连接
3	1M	2M
4	DI a.0	DI a.4
5	DI a.1	DI a.5
6	DI a.2	DI a.6
7	DI a.3	DI a.7

表格 A- 85 SM 1221 DI 16 x 24 VDC (6ES7 221-1BH30-0XB0) 的连接器引脚位置

引脚	X10	X11	X12	X13
1	GND	无连接	无连接	无连接
2	无连接	无连接	无连接	无连接
3	1M	2M	3 M	4 M
4	DI a.0	DI a.4	DI b.0	DI b.4
5	DI a.1	DI a.5	DI b.1	DI b.5
6	DI a.2	DI a.6	DI b.2	DI b.6
7	DI a.3	DI a.7	DI b.3	DI b.7

## A.6.2 SM 1222 8 点数字量输出技术规格

表格 A- 86 常规规范

型号	SM 1222 DQ 8 x 继电器	SM 1222 DQ8 RLY (切换)	SM 1222 DQ 8 x 24 VDC
订货号	6ES7 222-1HF30-0XB0	6ES7 222-1XF30-0XB0	6ES7 222-1BF30-0XB0
尺寸 W x H x D (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75
重量	190 g	310 g	180 g
功耗	4.5 W	5 W	1.5 W
电流消耗 (SM 总线)	120 mA	140 mA	120 mA
电流消耗 (24 VDC)	所用的每个继电器线圈 11 mA	所用的每个继电器线圈 16.7 mA	所用的每个继电器线圈 11 mA

A.6 数字信号模块 (SM)

表格 A- 87 数字量输出

型号	SM 1222 DQ 8 x 继电器	SM 1222 DQ8 RLY (切换)	SM 1222 DQ 8 x 24 VDC
输出路数	8	8	8
类型	继电器, 干触点	继电器切换触点	固态 - MOSFET (源型)
电压范围	5 到 30 VDC 或 5 到 250 VAC	5 到 30 VDC 或 5 到 250 VAC	20.4 到 28.8 VDC
最大电流时的逻辑 1 信号	--	--	最小 20 VDC
具有 10 K $\Omega$ 负载时的逻辑 0 信号	--	--	最大 0.1 VDC
电流 (最大)	2.0 A	2.0 A	0.5 A
灯负载	30 W DC/200 W AC	30 W DC/200 W AC	5W
通态触点电阻	新设备最大为 0.2 $\Omega$	新设备最大为 0.2 $\Omega$	最大 0.6 $\Omega$
每点的漏泄电流	--	--	最大 10 $\mu$ A
浪涌电流	触点闭合时为 7 A	触点闭合时为 7 A	8 A, 最长持续 100 ms
过载保护	否	否	否
隔离 (现场侧与逻辑侧)	1500 VAC, 持续 1 分钟 (线圈与触点) 无 (线圈与逻辑侧)	1500 VAC, 持续 1 分钟 (线圈与触点)	500 VAC, 持续 1 分钟
隔离电阻	新设备最小为 100 M $\Omega$	新设备最小为 100 M $\Omega$	--
断开触点间的绝缘	750 VAC, 持续 1 分钟	750 VAC, 持续 1 分钟	--
隔离组	2	8	1
每个公共端的电流 (最大)	10 A	2 A	4 A
电感钳位电压	--	--	L+ - 48 V, 1 W 损耗
开关延迟	最长 10 ms	最长 10 ms	断开到接通最长为 50 $\mu$ s 接通到断开最长为 200 $\mu$ s
继电器最大开关频率	1 Hz	1 Hz	--

型号	SM 1222 DQ 8 x 继电器	SM 1222 DQ8 RLY (切换)	SM 1222 DQ 8 x 24 VDC
机械寿命 (无负载)	10,000,000 个断开/闭合周期	10,000,000 个断开/闭合周期	--
额定负载下的触点寿命	100,000 个断开/闭合周期	100,000 个断开/闭合周期	--
RUN 到 STOP 时的行为	上一个值或替换值 (默认值为 0)	上一个值或替换值 (默认值为 0)	上一个值或替换值 (默认值为 0)
同时接通的输出数	8	<ul style="list-style-type: none"> <li>• 4 (无相邻点), 60° C (水平) 或 50° C (垂直) 时</li> <li>• 8, 55° C (水平) 或 45° C (垂直) 时</li> </ul>	8
电缆长度 (米)	500 m (屏蔽); 150 m (非屏蔽)	500 m (屏蔽); 150 m (非屏蔽)	500 m (屏蔽); 150 m (非屏蔽)

### A.6.3 SM 1222 16 点数字量输出技术规格

表格 A- 88 常规规范

型号	SM 1222 DQ 16 x 继电器	SM 1222 DQ 16 x 24 VDC
订货号	6ES7 222-1HH30-0XB0	6ES7 222-1BH30-0XB0
尺寸 W x H x D (mm)	45 x 100 x 75	45 x 100 x 75
重量	260 g	220 g
功耗	8.5 W	2.5 W
电流消耗 (SM 总线)	135 mA	140 mA
电流消耗 (24 VDC)	所用的每个继电器线圈 11 mA	-

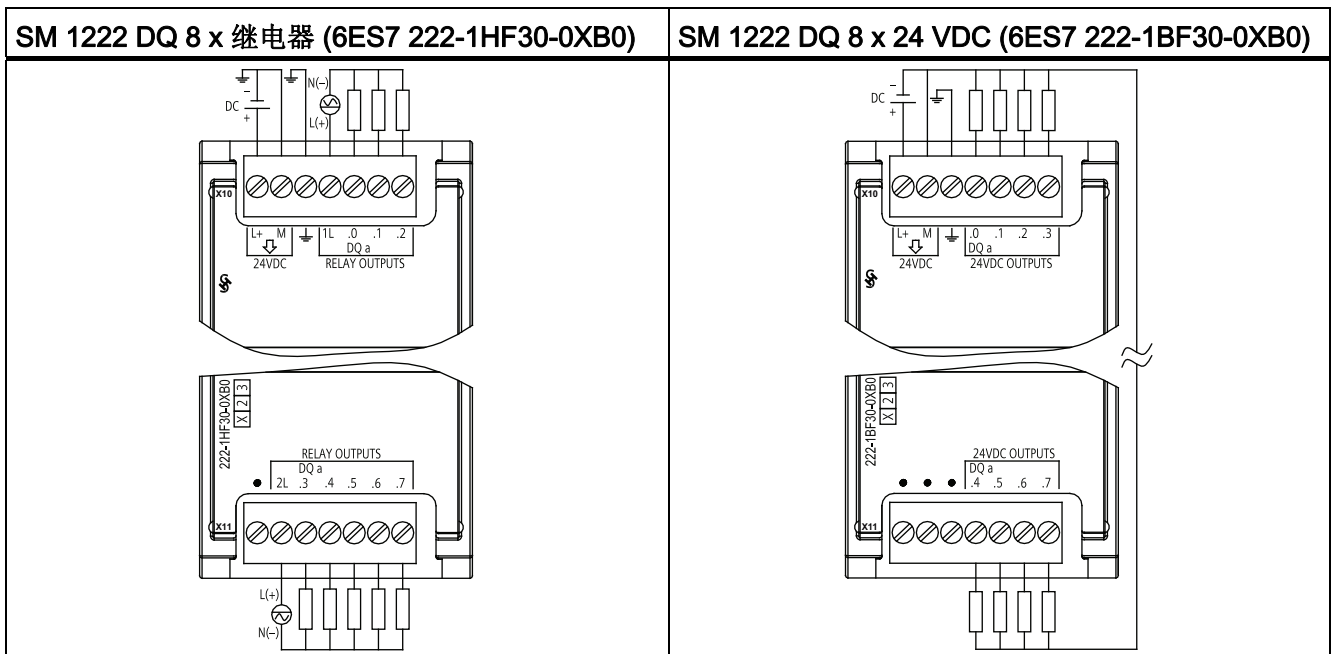
A.6 数字信号模块 (SM)

表格 A- 89 数字量输出

型号	SM1222 DQ 16 x 继电器	SM1222 DQ 16 x 24 VDC
输出路数	16	16
类型	继电器, 干触点	固态 - MOSFET (源型)
电压范围	5 到 30 VDC 或 5 到 250 VAC	20.4 到 28.8 VDC
最大电流时的逻辑 1 信号	-	最小 20 VDC
具有 10 K $\Omega$ 负载时的逻辑 0 信号	-	最大 0.1 VDC
电流 (最大)	2.0 A	0.5 A
灯负载	30 W DC/200 W AC	5W
通态触点电阻	新设备最大为 0.2 $\Omega$	最大 0.6 $\Omega$
每点的漏泄电流	--	最大 10 $\mu$ A
浪涌电流	触点闭合时为 7 A	8 A, 最长持续 100 ms
过载保护	否	否
隔离 (现场侧与逻辑侧)	1500 VAC, 持续 1 分钟 (线圈与触点) 无 (线圈与逻辑侧)	500 VAC, 持续 1 分钟
隔离电阻	新设备最小为 100 M $\Omega$	-
断开触点间的绝缘	750 VAC, 持续 1 分钟	-
隔离组	4	1
每个公共端的电流 (最大)	10 A	8 A
电感钳位电压	-	L+ - 48 V, 1 W 损耗
开关延迟	最长 10 ms	断开到接通最长为 50 $\mu$ s 接通到断开最长为 200 $\mu$ s
继电器最大开关频率	1 Hz	-
机械寿命 (无负载)	10,000,000 个断开/闭合周期	-
额定负载下的触点寿命	100,000 个断开/闭合周期	-
RUN 到 STOP 时的行为	上一个值或替换值 (默认值为 0)	上一个值或替换值 (默认值为 0)

型号	SM1222 DQ 16 x 继电器	SM1222 DQ 16 x 24 VDC
同时接通的输出数	<ul style="list-style-type: none"> <li>8 (无相邻点), 60° C (水平) 或 50° C (垂直) 时</li> <li>16, 55° C (水平) 或 45° C (垂直) 时</li> </ul>	16
电缆长度 (米)	500 m (屏蔽); 150 m (非屏蔽)	500 m (屏蔽); 150 m (非屏蔽)

表格 A-90 8 点数字量输出 SM 的接线图



A.6 数字信号模块 (SM)

表格 A- 91 SM 1222 DQ 8 x 继电器 (6ES7 222-1HF30-0XB0) 的连接器引脚位置

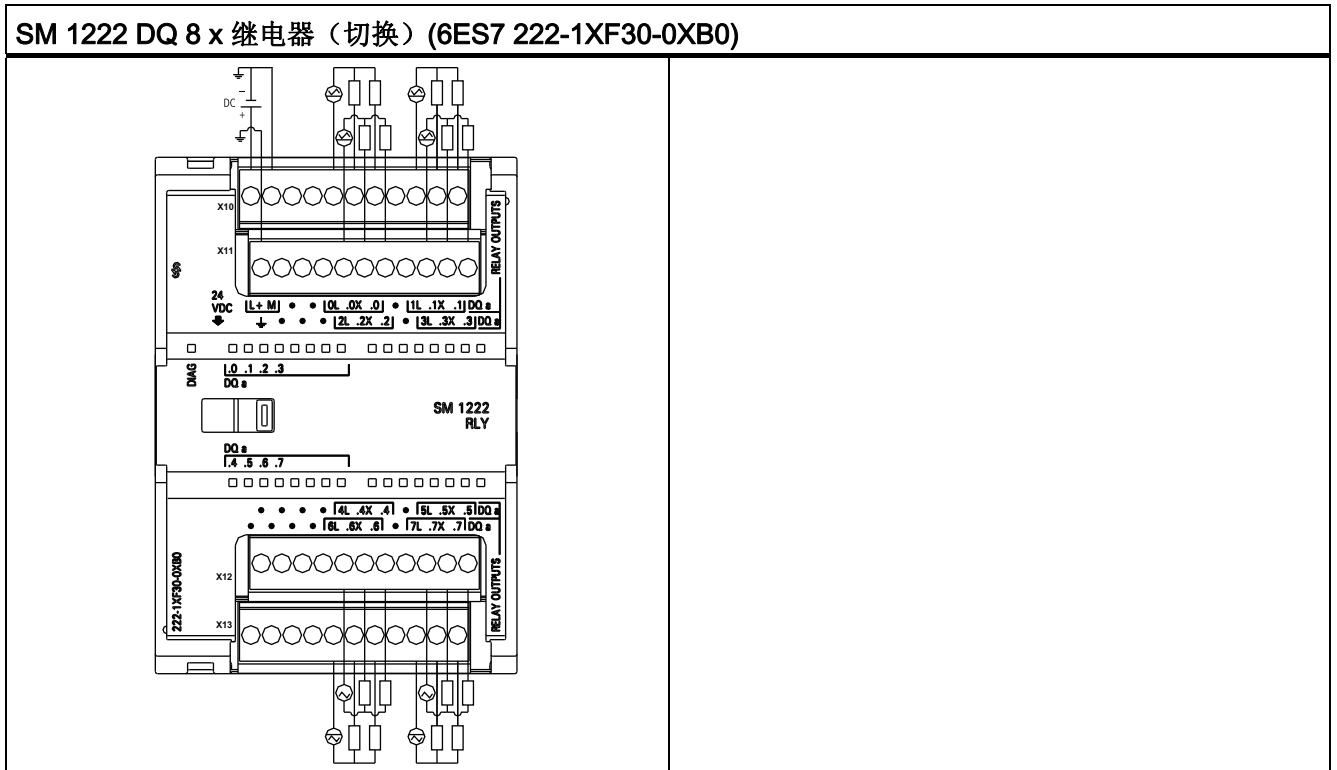
引脚	X10	X11
1	L+/24VDC	无连接
2	M/24VDC	2L
3	功能性接地	DQ a.3
4	1L	DQ a.4
5	DQ a.0	DQ a.5
6	DQ a.1	DQ a.6
7	DQ a.2	DQ a.7

表格 A- 92 SM 1222 DQ 8 x 24 VDC (6ES7 222-1BF30-0XB0) 的连接器引脚位置

引脚	X10	X11
1	L+/24VDC	无连接
2	M/24VDC	无连接
3	功能性接地	无连接
4	DQ a.0	DQ a.4
5	DQ a.1	DQ a.5
6	DQ a.2	DQ a.6
7	DQ a.2	DQ a.7



表格 A- 93 8 点数字量输出继电器 (切换) SM 的接线图

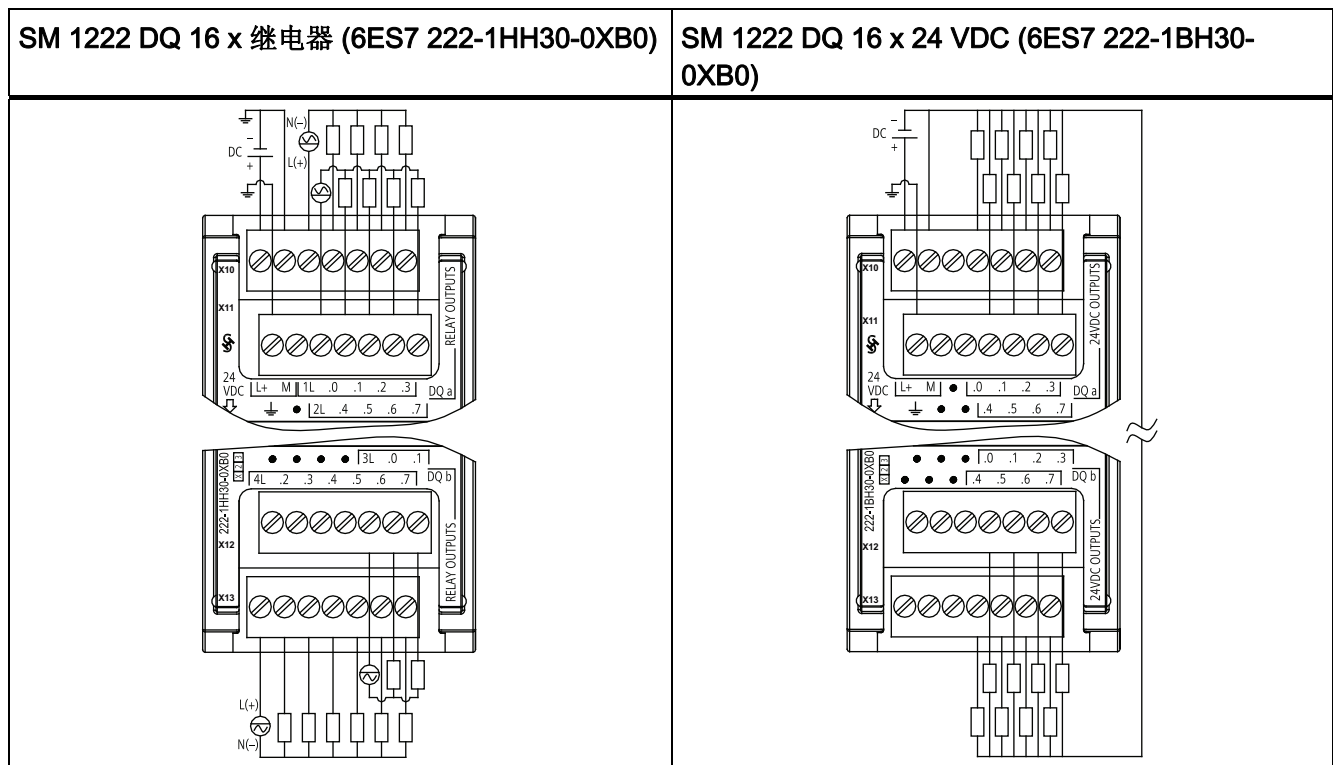


表格 A- 94 SM 1222 DQ 8 x 继电器切换 (6ES7 222-1XF30-0XB0) 的连接器针脚位置

针脚	X10	X11	X12	X13
1	L+/24VDC	功能性接地	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	无连接	无连接	无连接	无连接
4	无连接	无连接	无连接	无连接
5	0L	2L	4L	6L
6	DQ a.0X	DQ a.2X	DQ a.4X	DQ a.6X
7	DQ a.0	DQ a.2	DQ a.4	DQ a.6
8	无连接	无连接	无连接	无连接
9	1L	3L	5L	7L
10	DQ a.1X	DQ a.3X	DQ a.5X	DQ a.7X
11	DQ a.1	DQ a.3	DQ a.5	DQ a.7

A.6 数字信号模块 (SM)

表格 A-95 16 点数字量输出 SM 的接线图



表格 A-96 SM 1222 DQ 16 x 继电器 (6ES7 222-1HH30-0XB0) 的连接器针脚位置

针脚	X10	X11	X12	X13
1	L+/24VDC	功能性接地	无连接	4L
2	M/24VDC	无连接	无连接	DQ b.2
3	1L	2L	无连接	DQ b.3
4	DQ a.0	DQ a.4	无连接	DQ b.4
5	DQ a.1	DQ a.5	3L	DQ b.5
6	DQ a.2	DQ a.6	DQ b.0	DQ b.6
7	DQ a.3	DQ a.7	DQ b.1	DQ b.7

表格 A- 97 SM 1222 DQ 16 x 24 VDC (6ES7 222-1BH30-0XB0) 的连接器引脚位置

引脚	X10	X11	X12	X13
1	L+/24VDC	功能性接地	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	无连接	无连接	无连接	无连接
4	DQ a.0	DQ a.4	DQ b.0	DQ b.4
5	DQ a.1	DQ a.5	DQ b.1	DQ b.5
6	DQ a.2	DQ a.6	DQ b.2	DQ b.6
7	DQ a.3	DQ a.7	DQ b.3	DQ b.7

#### A.6.4 SM 1223 数字量输入/输出 VDC 规范

表格 A- 98 常规规范

型号	SM 1223 DI 8 x 24 VDC, DQ 8 x 继电器	SM 1223 DI 16 x 24 VDC, DQ 16 x 继电器	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
订货号	6ES7 223- 1PH30-0XB0	6ES7 223- 1PL30-0XB0	6ES7 223- 1BH30-0XB0	6ES7 223- 1BL30-0XB0
尺寸 W x H x D (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75	70 x 100 x 75
重量	230 g	350 g	210 g	310 g
功耗	5.5 W	10 W	2.5 W	4.5 W
电流消耗 (SM 总线)	145 mA	180 mA	145 mA	185 mA
电流消耗 (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	

A.6 数字信号模块 (SM)

表格 A- 99 数字量输入

型号	SM 1223 DI 8 x 24 VDC, DQ 8 x 继电器	SM 1223 DI 16 x 24 VDC, DQ 16 x 继电器	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
输入点数	8	16	8	16
类型	漏型/源型 (IEC 1 类漏型)			
额定电压	4 mA 时 24 VDC, 额定值			
允许的连续电压	最大 30 VDC			
浪涌电压	35 VDC, 持续 0.5 s			
逻辑 1 信号 (最小)	2.5 mA 时 15 VDC			
逻辑 0 信号 (最大)	1 mA 时 5 VDC			
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 分钟			
隔离组	2	2	2	2
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)			
同时接通的输入数	8	<ul style="list-style-type: none"> <li>• 8 (无相邻点), 60° C (水平) 或 50° C (垂直) 时</li> <li>• 16, 55° C (水平) 或 45° C (垂直) 时</li> </ul>	8	16
电缆长度 (米)	500 m (屏蔽); 300 m (非屏蔽)			

表格 A- 100 数字量输出

型号	SM 1223 DI 8 x 24 VDC, DQ 8 x 继电器	SM 1223 DI 16 x 24 VDC, DQ 16 x 继电器	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
输出点数	8	16	8	16
类型	继电器, 干触点		固态 - MOSFET (源型)	

型号	SM 1223 DI 8 x 24 VDC, DQ 8 x 继电器	SM 1223 DI 16 x 24 VDC, DQ 16 x 继电器	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
电压范围	5 到 30 VDC 或 5 到 250 VAC		20.4 到 28.8 VDC	
最大电流时的逻辑 1 信号	--		最小 20 VDC	
具有 10 K $\Omega$ 负载时的逻辑 0 信号	--		最大 0.1 VDC	
电流 (最大)	2.0 A		0.5 A	
灯负载	30 W DC/200 W AC		5 W	
通态触点电阻	新设备最大为 0.2 $\Omega$		最大 0.6 $\Omega$	
每点的漏泄电流	--		最大 10 $\mu$ A	
浪涌电流	触点闭合时为 7 A		8 A, 最长持续 100 ms	
过载保护	无			
隔离 (现场侧与逻辑侧)	1500 VAC, 持续 1 分钟 (线圈与触点) 无 (线圈与逻辑侧)		500 VAC, 持续 1 分钟	
隔离电阻	新设备最小为 100 M $\Omega$		--	
断开触点间的绝缘	750 VAC, 持续 1 分钟		--	
隔离组	2	4	1	1
每个公共端的电流	10A	8 A	4 A	8 A
电感钳位电压	--		L+ - 48 V, 1 W 损耗	
开关延迟	最长 10 ms		断开到接通最长为 50 $\mu$ s 接通到断开最长为 200 $\mu$ s	
继电器最大开关频率	1 Hz		--	
机械寿命 (无负载)	10,000,000 个断开/闭合周期		--	
额定负载下的触点寿命	100,000 个断开/闭合周期		--	
RUN 到 STOP 时的行为	上一个值或替换值 (默认值为 0)			

A.6 数字信号模块 (SM)

型号	SM 1223 DI 8 x 24 VDC, DQ 8 x 继电器	SM 1223 DI 16 x 24 VDC, DQ 16 x 继电器	SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC
同时接通的输出数	8	<ul style="list-style-type: none"> <li>8 (无相邻点), 60° C (水平) 或 50° C (垂直) 时</li> <li>16, 55° C (水平) 或 45° C (垂直) 时</li> </ul>	8	16
电缆长度 (米)	500 m (屏蔽); 150 m (非屏蔽)			

表格 A- 101 数字量输入 VDC/输出继电器 SM 的接线图

SM 1223 DI 8 x 24 VDC, DQ 8 x 继电器 (6ES7 223-1PH30-0XB0)	SM 1223 DI 16 x 24 VDC, DQ 16 x 继电器 (6ES7 223-1PL30-0XB0)	备注
		<p>① 对于漏型输入, 将“-”连接到“M”(如图所示)。</p> <p>对于源型输入, 将“+”连接到“M”。</p>

表格 A- 102 SM 1223 DI 8 x 24 VDC, DQ 8 x 继电器 (6ES7 223-1PH30-0XB0) 的连接器针脚位置

针脚	X10	X11	X12	X13
1	L+/24VDC	GND	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	1M	2M	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

表格 A- 103 SM 1223 DI 16 x 24 VDC, DQ 16 x 继电器 (6ES7 223-1PL30-0XB0) 的连接器针脚位置

针脚	X10	X11	X12	X13
1	L+/24VDC	GND	1L	3L
2	M/24VDC	无连接	DQ a.0	DQ b.0
3	1M	2M	DQ a.1	DQ b.1
4	DI a.0	DI b.0	DQ a.2	DQ b.2
5	DI a.1	DI b.1	DQ a.3	DQ b.3
6	DI a.2	DI b.2	无连接	无连接
7	DI a.3	DI b.3	2L	4L
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

A.6 数字信号模块 (SM)

表格 A- 104 数字量输入 VDC/输出 SM 的接线图

SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC (6ES7 223-1BH30-0XB0)	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC (6ES7 223-1BL30-0XB0)	备注
		<p>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”。</p>

表格 A- 105 SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC (6ES7 223-1BH30-0XB0) 的连接器针脚位置

针脚	X10	X11	X12	X13
1	L+/24VDC	GND	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	1M	2M	无连接	无连接
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7



表格 A- 106 SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC (6ES7 223-1BL30-0XB0) 的连接器  
针脚位置

针脚	X10	X11	X12	X13
1	L+/24VDC	GND	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	1M	2M	无连接	无连接
4	DI a.0	DI b.0	DQ a.0	DQ b.0
5	DI a.1	DI b.1	DQ a.1	DQ b.1
6	DI a.2	DI b.2	DQ a.2	DQ b.2
7	DI a.3	DI b.3	DQ a.3	DQ b.3
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

### A.6.5 SM 1223 数字量输入/输出 AC 规范

表格 A- 107 常规规范

<b>型号</b>	<b>SM 1223 DI 8 x120/230 VAC/DQ 8 x 继电器</b>
订货号	6ES7 223-1QH30-0XB0
尺寸 W x H x D (mm)	45 x 100 x 75 mm
重量	190 g
功耗	7.5 W
电流消耗 (SM 总线)	120 mA
电流消耗 (24 VDC)	接通时每点输出为 11 mA

A.6 数字信号模块 (SM)

表格 A- 108 数字量输入

型号	SM 1223 DI 8 x 120/230 VAC/DQ 8 x 继电器
输入点数	8
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 <sup>1</sup>	4
输入延迟时间	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
电缆长度	非屏蔽: 300 m 屏蔽: 500 m
同时接通的输入数	8

<sup>1</sup> 组中通道的相位必须相同。

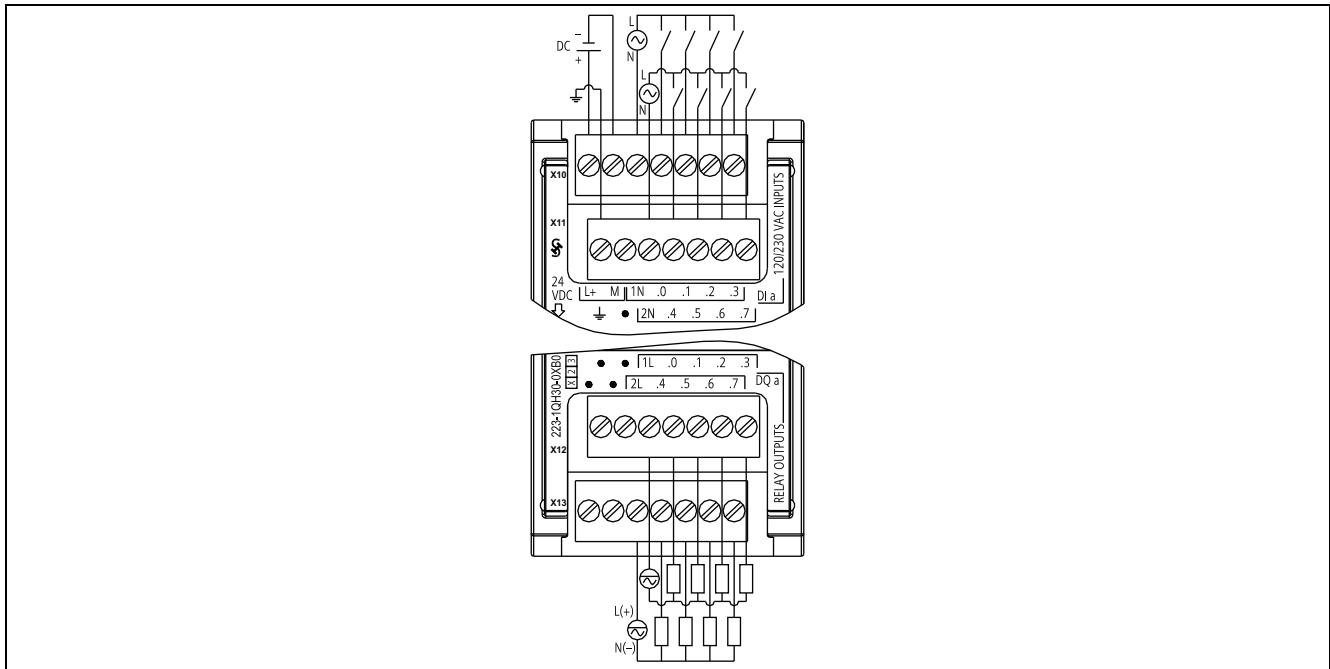
表格 A- 109 数字量输出

型号	SM 1223 DI 8 x 120/230 VAC/DQ 8 x 继电器
输出点数	8
类型	继电器, 干触点
电压范围	5 到 30 VDC 或 5 到 250 VAC
最大电流时的逻辑 1 信号	--
具有 10 K $\Omega$ 负载时的逻辑 0 信号	--
电流 (最大)	2.0 A
灯负载	30 W DC/200 W AC

型号	SM 1223 DI 8 x 120/230 VAC/DQ 8 x 继电器
通态触点电阻	新设备最大为 0.2 $\Omega$
每点的漏泄电流	--
浪涌电流	触点闭合时为 7 A
过载保护	无
隔离（现场侧与逻辑侧）	1500 VAC，持续 1 分钟（线圈与触点） 无（线圈与逻辑侧）
隔离电阻	新设备最小为 100 M $\Omega$
断开触点间的绝缘	750 VAC，持续 1 分钟
隔离组	2
每个公共端的电流（最大）	10 A
电感钳位电压	--
切换延迟（最大值）	10 ms
继电器最大开关频率	1 Hz
机械寿命（无负载）	10,000,000 个断开/闭合周期
额定负载下的触点寿命	1000,000 个断开/闭合周期
RUN 到 STOP 时的行为	上一个值或替换值（默认值为 0）
同时接通的输出数	<ul style="list-style-type: none"> <li>• 4（无相邻点），60° C（水平）或 50° C（垂直）时</li> <li>• 8，55° C（水平）或 45° C（垂直）时</li> </ul>
电缆长度（米）	500 m（屏蔽）；150 m（非屏蔽）

A.6 数字信号模块 (SM)

表格 A- 110 SM 1223 DI 8 x 120/230 VAC, DQ 8 x 继电器 (6ES7 223-1QH30-0XB0)



表格 A- 111 SM 1223 DI 8 x 120/240 VAC, DQ 8 x 继电器 (6ES7 223-1QH30-0XB0) 的连接  
器针脚位置

针脚	X10	X11	X12	X13
1	L+/24VDC	GND	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	1N	2N	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

## A.7 模拟信号模块 (SM)

### A.7.1 SM 1231 模拟量输入模块规范

表格 A- 112 常规规范

型号	SM 1231 AI 4 x 13 位	SM 1231 AI 8 x 13 位	SM 1231 AI 4 x 16 位
订货号	6ES7 231-4HD30-0XB0	6ES7 231-4HF30-0XB0	6ES7 231-5ND30-0XB0
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	2.2 W	2.3 W	2.0 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

表格 A- 113 模拟量输入

型号	SM 1231 AI 4 x 13 位	SM 1231 AI 8 x 13 位	SM 1231 AI 4 x 16 位
输入路数	4	8	4
类型	电压或电流 (差动): 可 2 个选为一组		电压或电流 (差动)
范围	$\pm 10\text{ V}$ 、 $\pm 5\text{ V}$ 、 $\pm 2.5\text{ V}$ 或 0 到 20 mA		$\pm 10\text{ V}$ 、 $\pm 5\text{ V}$ 、 $\pm 2.5\text{ V}$ 、 $\pm 1.25\text{ V}$ 、0 到 20 mA 或 4 mA 到 20 mA
满量程范围 (数据字)	-27,648 到 27,648 电压/0 到 27,648 电流		
过冲/下冲范围 (数据字) 请参见 电压和电流 (页 845)的 模拟量输入范围部分。	电压: 32,511 到 27,649/-27,649 到 -32,512 电流: 32,511 到 27,649/0 到 -4864		电压: 32,511 到 27,649/ -27,649 到 -32,512 电流: (0-20 mA): 32,511 到 27,649/ 0 到 -4864; 4-20 mA: 32511 到 27,649/ -1 到 -4,864

A.7 模拟信号模块 (SM)

型号	SM 1231 AI 4 x 13 位	SM 1231 AI 8 x 13 位	SM 1231 AI 4 x 16 位
上溢/下溢（数据字） 请参见 电压和电流 (页 845)的 输入范围部分。	电压： 32,767 到 32,512/-32,513 到 -32,768 电流： 32,767 到 32,512/-4865 到 -32,768		电压： 32,767 到 32,512/-32,513 到 - 32,768 电流： 0 到 20 mA 32,767 到 32,512/- 4865 到 -32,768 4-20 mA 32,767 到 32,512/-4,865 到 - 32,768
Resolution <sup>1</sup>	12 位 + 符号位		15 位 + 符号位
最大耐压/耐流	±35 V/±40 mA		
平滑化	无、弱、中或强 请参见 阶跃响应时间 (页 844)部分。		
噪声抑制	400、60、50 或 10 Hz 请参见 采样率 (页 844)部分。		
输入阻抗	≥ 9 MΩ（电压）/280 Ω（电流）		≥ 1 MΩ（电压）/ <315 Ω, >280 Ω（电 流）
隔离	否		
现场侧与逻辑侧			500 VAC
逻辑侧与 24 VDC			500 VAC
现场侧与 24 VDC			500 VAC
通道与通道			无
精度（25°C/-20 到 60°C）	满量程的 ±0.1%/±0.2%		满量程的 ±0.1%/±0.3%
测量原理	实际值转换		
共模抑制	40 dB, DC 到 60 Hz		
工作信号范围 <sup>1</sup>	信号加共模电压必须小于 +12 V 且大于 -12 V		
电缆长度（米）	100 m, 屏蔽双绞线		

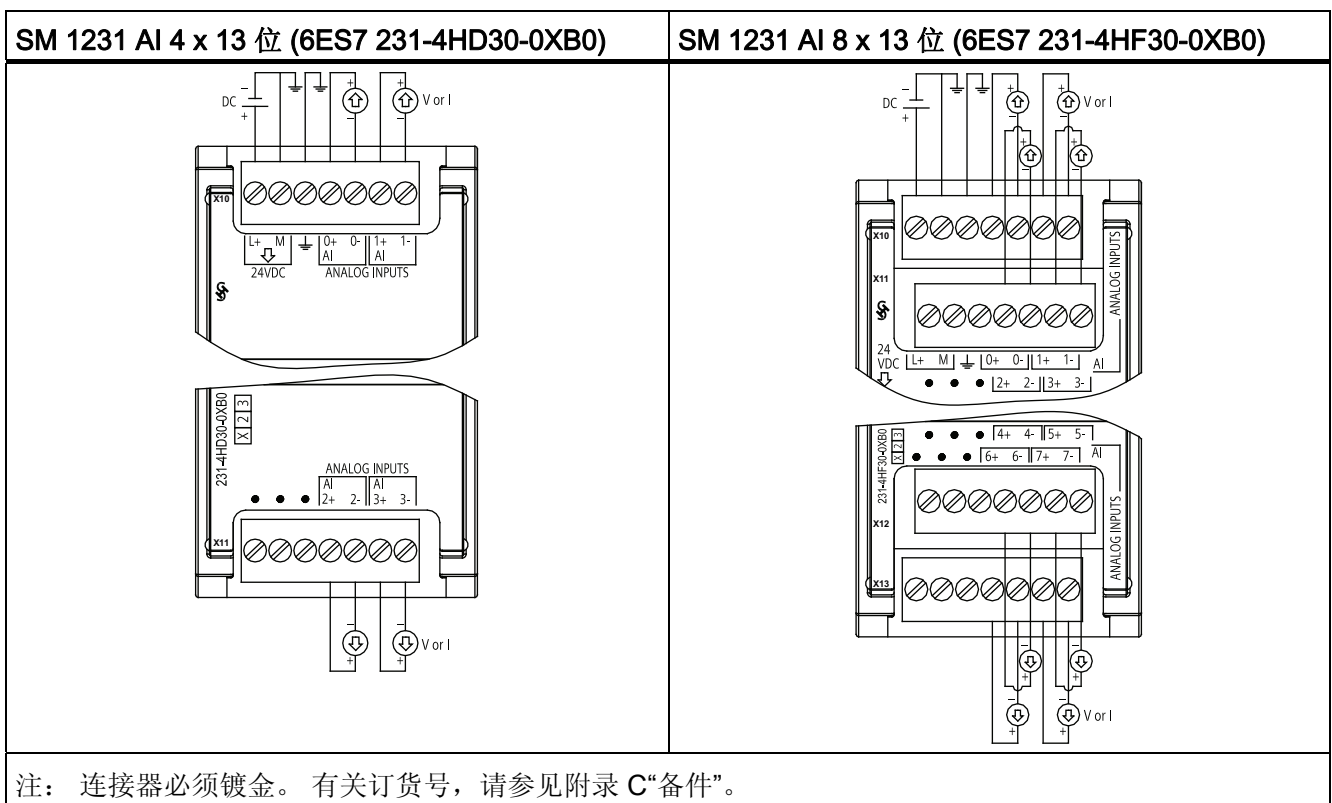
<sup>1</sup> 施加至某一通道的电压超出工作范围可能导致对其它通道造成干扰。

表格 A- 114 诊断

型号	SM 1231 AI 4 x 13 位	SM 1231 AI 8 x 13 位	SM 1231 AI 4 x 16 位
上溢/下溢	是 <sup>1</sup>	是	是
24 VDC 低压	是	是	是
开路	--	--	仅限 4 到 20 mA 范围 (如果输入低于 - 4,164; 1.0 mA)

<sup>1</sup> 对于 SM 1231 AI 4 x 13 位: 如果对输入端施加大于 +30 VDC 或小于 -15 VDC 的电压, 则结果值将是未知的, 因此相应的上溢或下溢不会激活。

表格 A- 115 模拟量输入 SM 的接线图



A.7 模拟信号模块 (SM)

表格 A- 116 SM 1231 AI 4 x 13 位 (6ES7 231-4HD30-0XB0) 的连接器和脚位置

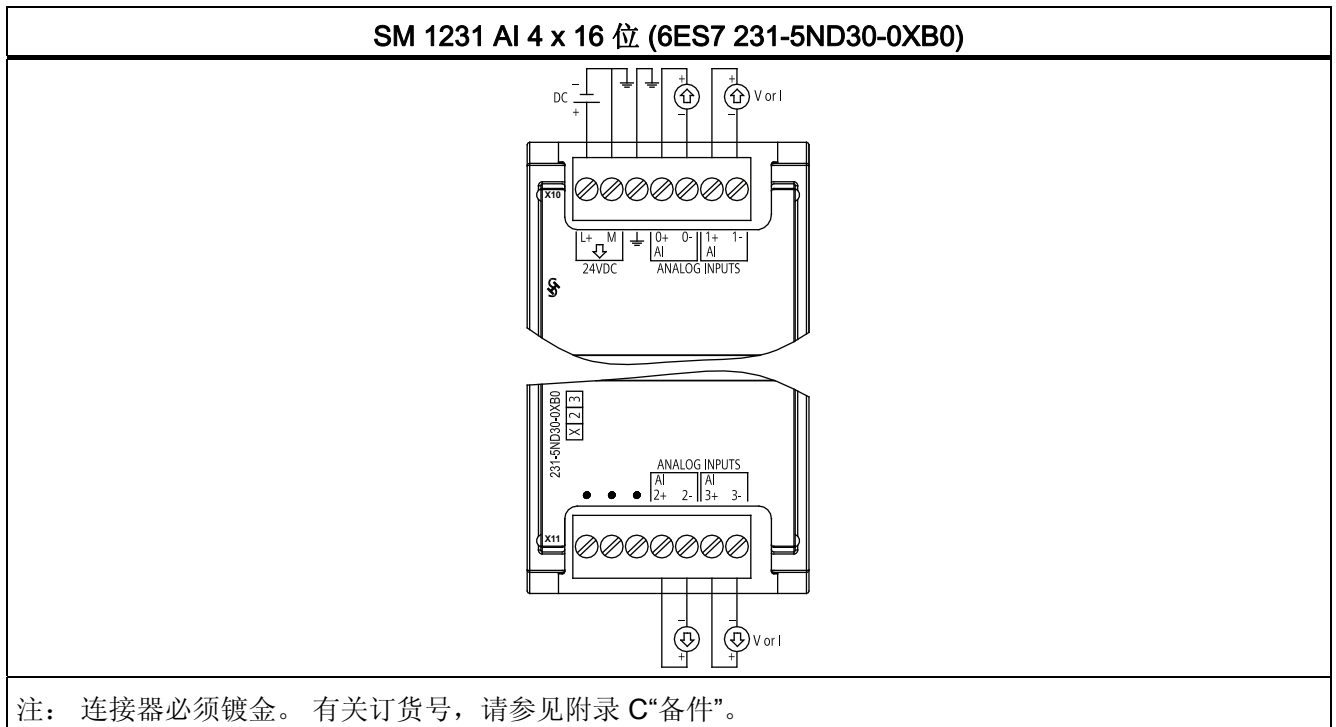
引脚	X10 (镀金)	X11 (镀金)
1	L+/24VDC	无连接
2	M/24VDC	无连接
3	GND	无连接
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

表格 A- 117 SM 1231 AI 8 x 13 位 (6ES7 231-4HF30-0XB0) 的连接器和脚位置

引脚	X10 (镀金)	X11 (镀金)	X12 (镀金)	X13 (镀金)
1	L+/24VDC	无连接	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	GND	无连接	无连接	无连接
4	AI 0+	AI 2+	AI 4+	AI 6+
5	AI 0-	AI 2-	AI 4-	AI 6-
6	AI 1+	AI 3+	AI 5+	AI 7+
7	AI 1-	AI 3-	AI 5-	AI 7-



表格 A- 118 模拟量输入 SM 的接线图



表格 A- 119 SM 1231 AI 4 x 16 位 (6ES7 231-5ND30-0XB0) 的连接器针脚位置

针脚	X10 (镀金)	X11 (镀金)
1	L+/24VDC	无连接
2	M/24VDC	无连接
3	GND	无连接
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

**说明**

应将未使用的模拟量输入短路。

当针对“电流”模式组态输入时，除非对模块施加外部电源，否则没有电流流过输入。

A.7 模拟信号模块 (SM)

A.7.2 SM 1232 模拟量输出模块规范

表格 A- 120 一般规格

技术数据	SM 1232 AQ 2 x 14 位	SM 1232 AQ 4 x 14 位
订货号	6ES7 232-4HB30-0XB0	6ES7 232-4HD30-0XB0
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	1.8 W	2.0 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)

表格 A- 121 模拟量输出

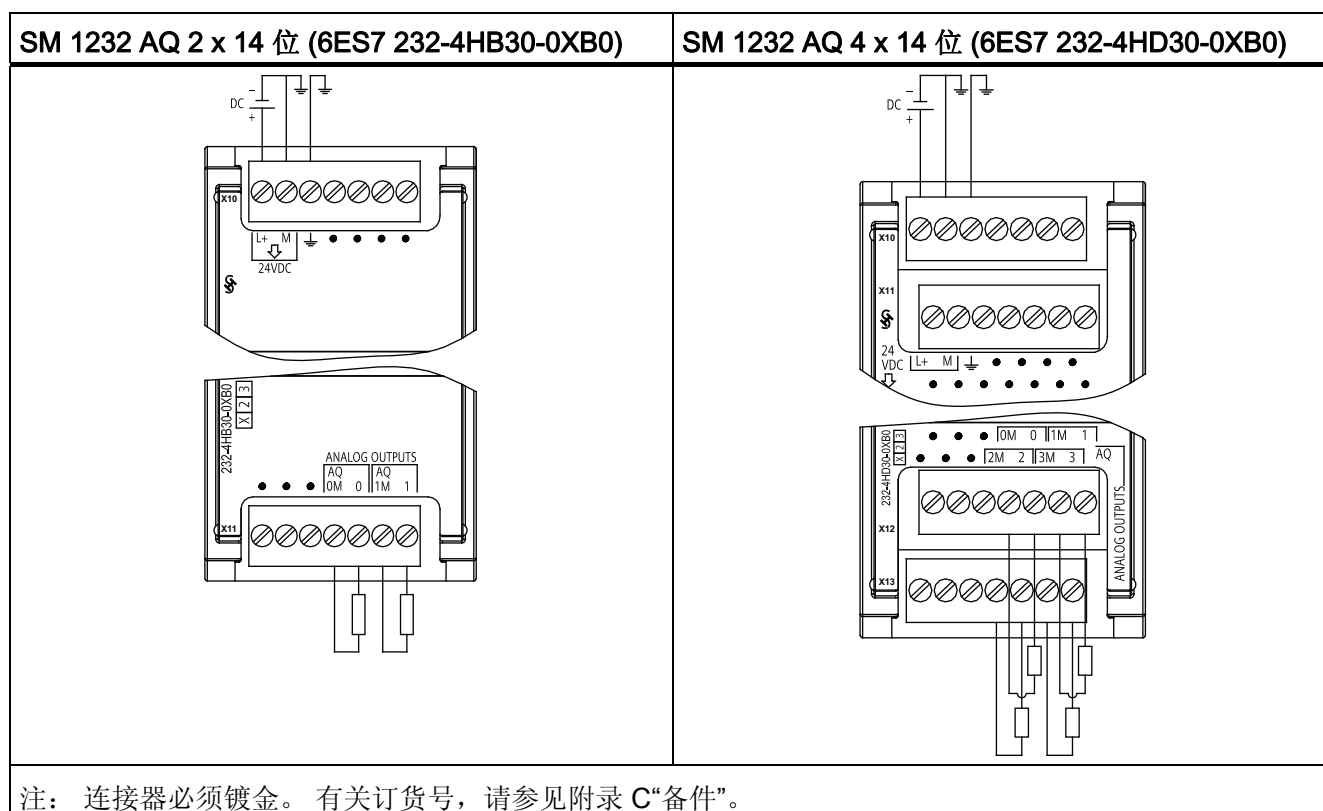
技术数据	SM 1232 AQ 2 x 14 位	SM 1232 AQ 4 x 14 位
输出路数	2	4
类型	电压或电流	电压或电流
范围	±10 V 或 0 到 20 mA	±10 V 或 0 到 20 mA
分辨率	电压: 14 位 电流: 13 位	电压: 14 位 电流: 13 位
满量程范围 (数据字)	电压: -27,648 到 27,648; 电流: 0 到 27,648 请参见 电压和电流 (页 846) 的输出范围。	
精度 (25°C/-20 到 60°C)	满量程的 ±0.3%/±0.6%	
稳定时间 (新值的 95%)	电压: 300 μS (R), 750 μS (1 uF) 电流: 600 μS (1 mH), 2 ms (10 mH)	
负载阻抗	电压: ≥ 1000 Ω 电流: ≤ 600 Ω	
RUN 到 STOP 时的行为	上一个值或替换值 (默认值为 0)	

技术数据	SM 1232 AQ 2 x 14 位	SM 1232 AQ 4 x 14 位
隔离（现场侧与逻辑侧）	无	
电缆长度（米）	100 m 屏蔽双绞线	

表格 A- 122 诊断

技术数据	SM 1232 AQ 2 x 14 位	SM 1232 AQ 4 x 14 位
上溢/下溢	有	有
对地短路（仅限电压模式）	有	有
断路（仅限电流模式）	有	有
24 VDC 低压	有	有

表格 A- 123 模拟量输出 SM 的接线图



A.7 模拟信号模块 (SM)

表格 A- 124 SM 1232 AQ 2 x 14 位 (6ES7 232-4HB30-0XB0) 的连接器和针脚位置

针脚	X10 (镀金)	X11 (镀金)
1	L+/24VDC	无连接
2	M/24VDC	无连接
3	GND	无连接
4	无连接	AQ 0M
5	无连接	AQ 0
6	无连接	AQ 1M
7	无连接	AQ 1

表格 A- 125 SM 1232 AQ 4 x 14 位 (6ES7 232-4HD30-0XB0) 的连接器和针脚位置

针脚	X10 (镀金)	X11 (镀金)	X12 (镀金)	X13 (镀金)
1	L+/24VDC	无连接	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	GND	无连接	无连接	无连接
4	无连接	无连接	AQ 0M	AQ 2M
5	无连接	无连接	AQ 0	AQ 2
6	无连接	无连接	AQ 1M	AQ 3M
7	无连接	无连接	AQ 1	AIQ 3

A.7.3 SM 1234 模拟量输入/输出模块规范

表格 A- 126 常规规范

技术数据	SM 1234 AI 4 x 13 位/AQ 2 x 14 位
订货号	6ES7 234-4HE30-0XB0
尺寸 W x H x D (mm)	45 x 100 x 75
重量	220 g
功耗	2.4 W

<b>技术数据</b>	<b>SM 1234 AI 4 x 13 位/AQ 2 x 14 位</b>
电流消耗 (SM 总线)	80 mA
电流消耗 (24 VDC)	60 mA (无负载)

表格 A- 127 模拟量输入

<b>型号</b>	<b>SM 1234 AI 4 x 13 位/AQ 2 x 14 位</b>
输入路数	4
类型	电压或电流 (差动): 可 2 个选为一组
范围	$\pm 10\text{ V}$ 、 $\pm 5\text{ V}$ 、 $\pm 2.5\text{ V}$ 或 0 到 20 mA
满量程范围 (数据字)	-27,648 到 27,648
过冲/下冲范围 (数据字)	电压: 32,511 到 27,649/-27,649 到 -32,512 电流: 32,511 到 27,649/0 到 -4864 请参见 电压和电流 (页 845)的输入范围部分。
上溢/下溢 (数据字)	电压: 32,767 到 32,512/-32,513 到 -32,768 电流: 32,767 到 32,512/-4865 到 -32,768 请参见 电压和电流 (页 845)的输入范围部分。
分辨率	12 位 + 符号位
最大耐压/耐流	$\pm 35\text{ V}/\pm 40\text{ mA}$
平滑化	无、弱、中或强 请参见 阶跃响应时间 (页 844)部分。
噪声抑制	400、60、50 或 10 Hz 请参见 采样率 (页 844)部分。
输入阻抗	$\geq 9\text{ M}\Omega$ (电压) /280 $\Omega$ (电流)
隔离 (现场侧与逻辑侧)	否
精度 (25 °C/-20 到 60°C)	满量程的 $\pm 0.1\%/ \pm 0.2\%$
模数转换时间	625 $\mu\text{s}$ (400 Hz 抑制)
共模抑制	40 dB, DC 到 60 Hz
工作信号范围 <sup>1</sup>	信号加共模电压必须小于 +12 V 且大于 -12 V
电缆长度 (米)	100 m, 屏蔽双绞线

<sup>1</sup> 施加至某一通道的电压超出工作范围可能导致对其它通道造成干扰。

A.7 模拟信号模块 (SM)

表格 A- 128 模拟量输出

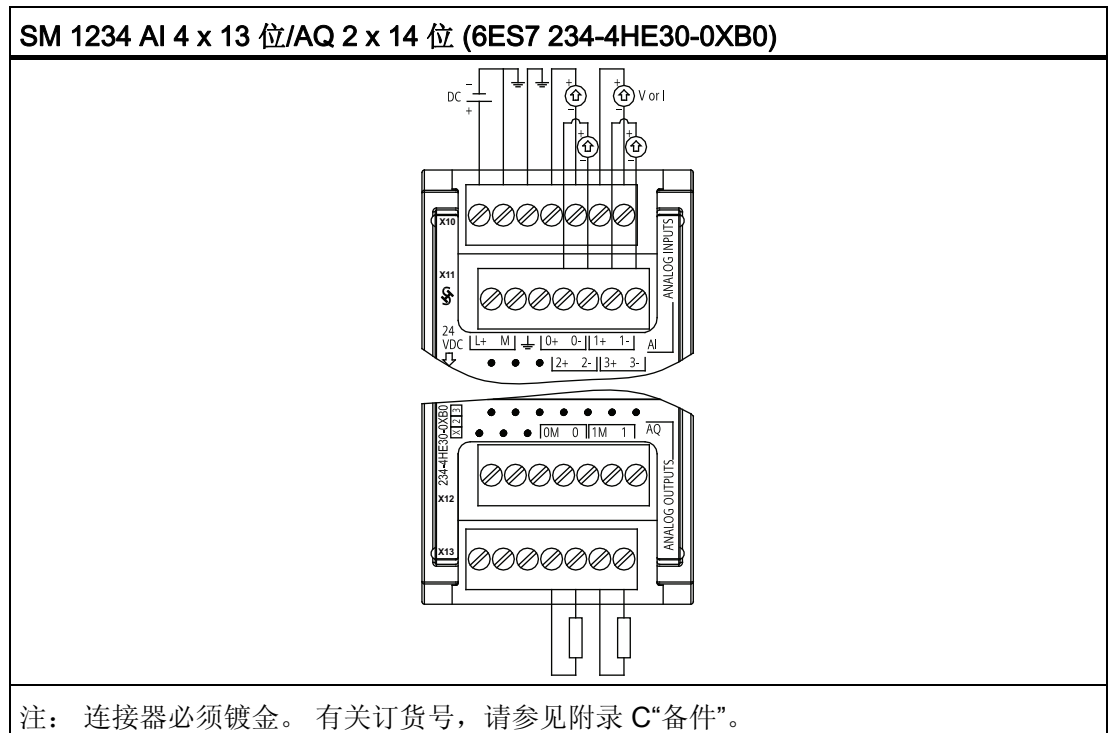
技术数据	SM 1234 AI 4 x 13 位/AQ 2 x 14 位
输出路数	2
类型	电压或电流
范围	±10 V 或 0 到 20 mA
分辨率	电压： 14 位； 电流： 13 位
满量程范围（数据字）	电压： -27,648 到 27,648； 电流： 0 到 27,648 请参见 电压和电流 (页 846)的输出范围部分。
精度（25°C/-20 到 60°C）	满量程的 ±0.3%/±0.6%
稳定时间（新值的 95%）	电压： 300 μS (R), 750 μS (1 uF) 电流： 600 μS (1 mH), 2 ms (10 mH)
负载阻抗	电压： ≥ 1000 Ω 电流： ≤ 600 Ω
RUN 到 STOP 时的行为	上一个值或替换值（默认值为 0）
隔离（现场侧与逻辑侧）	否
电缆长度（米）	100 m 屏蔽双绞线

表格 A- 129 诊断

型号	SM 1234 AI 4 x 13 位/AQ 2 x 14 位
上溢/下溢	是 <sup>1</sup>
对地短路（仅限电压模式）	输出端有
断路（仅限电流模式）	输出端有
24 VDC 低压	是

<sup>1</sup> 如果对输入端施加大于 +30 VDC 或小于 -15 VDC 的电压，则结果值将是未知的，因此相应的上溢或下溢可能不会激活。

表格 A- 130 模拟量输入/输出 SM 的接线图



表格 A- 131 SM 1234 AI 4 x 13 位/AQ 2 x 14 位 (6ES7 234-4HE30-0XB0) 的连接器针脚位置

针脚	X10 (镀金)	X11 (镀金)	X12 (镀金)	X13 (镀金)
1	L+/24VDC	无连接	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	GND	无连接	无连接	无连接
4	AI 0+	AI 2+	无连接	AQ 0M
5	AI 0-	AI 2-	无连接	AQ 0
6	AI 1+	AI 3+	无连接	AQ 1M
7	AI 1-	AI 3-	无连接	AIQ 1

**说明**

应将未使用的模拟量输入短路。

当针对“电流”模式组态输入时，除非对模块施加外部电源，否则没有电流流过输入。

A.7 模拟信号模块 (SM)

A.7.4 模拟量输入的阶跃响应

表格 A- 132 阶跃响应 (ms)，0 到满量程（在 95% 处测得）

平滑化选项（采样平均）	噪声消减/抑制频率（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	4 ms	18 ms	22 ms	100 ms
弱（4 个周期）： 4 次采样	9 ms	52 ms	63 ms	320 ms
中（16 个周期）： 16 次采样	32 ms	203 ms	241 ms	1200 ms
强（32 个周期）： 32 次采样	61 ms	400 ms	483 ms	2410 ms
采样时间 • 4 AI x 13 位 • 8 AI x 13 位 • 4 AI4 x 16 位	• 0.625 ms • 1.25 ms • 0.417 ms	• 4.17 ms • 4.17 ms • 0.397 ms	• 5 ms • 5 ms • 0.400 ms	• 25 ms • 25 ms • 0.400 ms

A.7.5 模拟量输入的采样时间和更新时间

表格 A- 133 采样时间和更新时间

抑制频率（积分时间）	采样时间	所有通道的模块更新时间	
		4 通道 SM	8 通道 SM
400 Hz (2.5 ms)	• 4 通道 SM: 0.625 ms • 8 通道 SM: 1.250 ms	0.625 ms	1.250 ms
60 Hz (16.6 ms)	4.170 ms	4.17 ms	4.17 ms
50 Hz (20 ms)	5.000 ms	5 ms	5 ms
10 Hz (100 ms)	25.000 ms	25 ms	25 ms



## A.7.6 模拟量输入的电压测量范围

表格 A- 134 模拟量输入的电压表示法

系统		电压测量范围						
十进制	十六进制	±10 V	±5 V	±2.5 V	±1.25V		0 到 10 V	
32767	7FFF	11.851 V	5.926 V	2.963 V	1.481 V	上溢	11.851 V	上溢
32512	7F00							
32511	7EFF	11.759 V	5.879 V	2.940 V	1.470 V	过冲范围	11.759 V	过冲范围
27649	6C01							
27648	6C00	10 V	5 V	2.5 V	1.250 V	额定范围	10 V	额定范围
20736	5100	7.5 V	3.75 V	1.875 V	0.938 V		7.5 V	
1	1	361.7 μV	180.8 μV	90.4 μV	45.2 μV		361.7 μV	
0	0	0 V	0 V	0 V	0 V		0 V	
-1	FFFF						不支持负值	
-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			
-32513	80FF					下溢		
-32768	8000	-11.851 V	-5.926 V	-2.963 V	-1.481 V			

A.7 模拟信号模块 (SM)

A.7.7 模拟量输入的电流测量范围

表格 A- 135 模拟量输入的电流表示法

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			

A.7.8 输出 (AQ) 的电压和电流测量范围 (SB 和 SM)

表格 A- 136 模拟量输出的电压表示法

系统		电压输出范围	
十进制	十六进制	$\pm 10 V$	
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	

系统		电压输出范围	
十进制	十六进制	$\pm 10 \text{ V}$	
1	1	361.7 $\mu\text{V}$	
0	0	0 V	
-1	FFFF	-361.7 $\mu\text{V}$	
-20736	AF00	-7.5 V	
-27648	9400	-10 V	
-27649	93FF		
-32512	8100	-11.76 V	下溢
-32513	80FF	请参见注 1	
-32768	8000	请参见注 1	

1 在上溢或下溢情况下，模拟量输出将根据为模拟量信号模块设置的设备配置属性动作。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。

表格 A- 137 模拟量输出的电流表示法

系统		电流输出范围	
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		
27648	6C00	20 mA	额定范围
20736	5100	15 mA	
1	1	723.4 nA	
0	0	0 mA	

1 在上溢或下溢情况下，模拟量输出将根据为模拟量信号模块设置的设备配置属性动作。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 热电偶和 RTD 信号模块 (SM)

A.8 热电偶和 RTD 信号模块 (SM)

A.8.1 SM 1231 热电偶

表格 A- 138 一般规格

型号	SM 1231 AI 4 x 16 位 TC	SM 1231 AI 8 x 16 位 TC
订货号	6ES7 231-5QD30-0XB0	6ES7 231-5QF30-0XB0
尺寸 W x H x D (mm)	45 x 100 x 75	45 x 100 x 75
重量	180 g	190 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) <sup>1</sup>	40 mA	40 mA

<sup>1</sup> 20.4 到 28.8 VDC (2 类受限制电源, 或 PLC 提供的传感器电源)

表格 A- 139 模拟量输入

型号	SM 1231 AI 4 x 16 位 TC	SM 1231 AI 8 x 16 位 TC
输入路数	4	8
范围	请参见 热电偶选型表 (页 852)。	请参见 热电偶选型表 (页 852)。
额定范围 (数据字)		
过量程/欠量程 (数据字)		
上溢/下溢 (数据字)		
分辨率		
温度	0.1° C/0.1° F	0.1° C/0.1° F
电压	15 位 + 符号	15 位 + 符号
最大耐压	±35 V	±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	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	120 VAC 时大于 120 dB
阻抗	≥ 10 MΩ	≥ 10 MΩ

型号	SM 1231 AI 4 x 16 位 TC	SM 1231 AI 8 x 16 位 TC	
隔离	现场侧与逻辑侧	500 VAC	500 VAC
	现场侧与 24 VDC	500 VAC	500 VAC
	24 VDC 与逻辑侧	500 VAC	500 VAC
通道间	120 VAC	120 VAC	
精度	请参见 热电偶选型表 (页 852)。	请参见 热电偶选型表 (页 852)。	
可重复性	±0.05% FS	±0.05% FS	
测量原理	积分型	积分型	
模块更新时间	请参见 噪声消减选项表 (页 852)。	请参见 噪声消减选项表 (页 852)。	
冷端误差	±1.5°C	±1.5°C	
电缆长度 (米)	到传感器最长为 100 米	到传感器最长为 100 米	
导线电阻	最大 100 Ω	最大 100 Ω	

表格 A- 140 诊断

型号	SM 1231 AI 4 x 16 位 TC	SM 1231 AI 8 x 16 位 TC
上溢/下溢 <sup>1</sup>	有	有
断线 (仅限电流模式) <sup>2</sup>	有	有
24 VDC 低压 <sup>1</sup>	有	有

<sup>1</sup> 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。

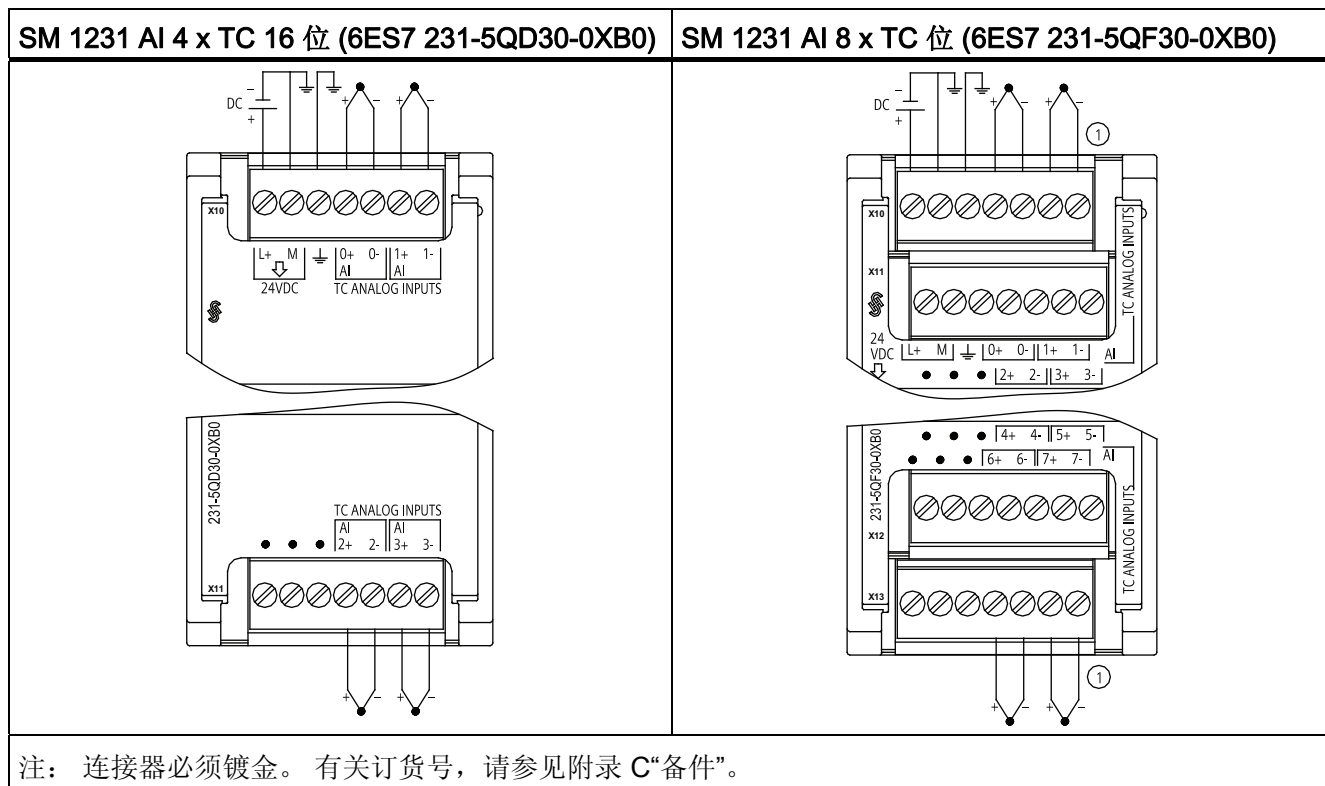
<sup>2</sup> 如果断线报警已禁用，但传感器接线存在开路情况，则模块可能会报告随机值。

SM 1231 热电偶 (TC) 模拟量信号模块可测量连接到模块输入的电压值。温度测量类型可以是“热电偶”或“电压”类型。

- “热电偶”：将度数乘 10 得到该值（例如，25.3 度将报告为十进制数 253）。
- “电压”：额定范围的满量程值将是十进制数 27648。

A.8 热电偶和 RTD 信号模块 (SM)

表格 A- 141 热电偶 SM 的接线图



① 为清晰起见，未显示 TC 2、3、4 和 5 的连接。

表格 A- 142 SM 1231 AI 4 x TC 16 位 (6ES7 231-5QD30-0XB0) 的连接器针脚位置

针脚	X10 (镀金)	X11 (镀金)
1	L+/24VDC	无连接
2	M/24VDC	无连接
3	GND	无连接
4	AI 0+/TC	AI 2+/TC
5	AI 0-/TC	AI 2-/TC
6	AI 1+/TC	AI 3+/TC
7	AI 1-/TC	AI 3-/TC

表格 A- 143 SM 1231 AI 8 x TC 位 (6ES7 231-5QF30-0XB0) 的连接器和针脚位置

针脚	X10 (镀金)	X11 (镀金)	X12 (镀金)	X13 (镀金)
1	L+/24VDC	无连接	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	GND	无连接	无连接	无连接
4	AI 0+/TC	AI 2+/TC	AI 4 I-/TC	AI 6 I-/TC
5	AI 0-/TC	AI 2-/TC	AI 4 I+/TC	AI 6 I+/TC
6	AI 1+/TC	AI 3+/TC	AI 5 M-/TC	AI 7 M-/TC
7	AI 1-/TC	AI 3-/TC	AI 5 M+/TC	AI 7 M+/TC

**说明**

应将未使用的模拟量输入短路。

可以取消激活热电偶的未使用通道。如果取消激活未使用的通道，不会出现任何错误。

**A.8.1.1 热电偶的基本操作**

两种不同的金属彼此之间存在电气连接时，便会形成热电偶。热电偶产生的电压与结点温度成正比。电压很小；一微伏能表示很多度。测量热电偶产生的电压，对额外的结点进行补偿，然后将测量结果线性化，这些是使用热电偶测量温度的基础。

将热电偶连接到 SM 1231 热电偶模块时，两条不同的金属线需连接到模块的信号连接器上。这两条不同的金属线互相连接的位置即形成了传感器热电偶。

在这两条不同的金属线与信号连接器相连的位置，构成了另外二个热电偶。连接器温度会引起一定的电压，该电压将添加到传感器热电偶产生的电压中。如果不对该电压进行修正，结果报告的温度将偏离传感器温度。

冷端补偿便是用于对连接器热电偶进行补偿。热电偶表是基于参比端温度（通常是零摄氏度）得来的。冷端补偿用于将连接器温度修正为零摄氏度。冷端补偿可消除连接器热电偶增加的电压。模块的温度在内部测量，然后转换为数值并添加到传感器换算中。之后是使用热电偶表对修正后的传感器换算值进行线性化。

为使冷端补偿取得最佳效果，必须将热电偶模块安装在温度稳定的环境中。符合模块规范的模块环境温度的缓慢变化（低于 0.1° C/min）能够被正确补偿。穿过模块的空气流动也会引起冷端补偿误差。

A.8 热电偶和 RTD 信号模块 (SM)

如果需要更佳冷端误差补偿效果，则可使用外部 iso 热端子块。热电偶模块可以使用 0° C 基准值或 50° C 基准值端子块。

A.8.1.2 SM 1231 热电偶选型表

下表给出了 SM 1231 热电偶信号模块支持的不同热电偶类型对应的测量范围和精度。

表格 A- 144 SM 1231 热电偶选型表

类型	低于范围最小值 <sup>1</sup>	额定范围下限	额定范围上限	超出范围最大值 <sup>2</sup>	25°C 时的额定范围 <sup>3、4</sup> 精度	-20°C 到 60°C 时的额定范围 <sup>1、2</sup> 精度
J	-210.0°C	-150.0°C	1200.0°C	1450.0°C	±0.3°C	±0.6°C
K	-270.0°C	-200.0°C	1372.0°C	1622.0°C	±0.4°C	±1.0°C
T	-270.0°C	-200.0°C	400.0°C	540.0°C	±0.5°C	±1.0°C
E	-270.0°C	-200.0°C	1000.0°C	1200.0°C	±0.3°C	±0.6°C
R & S	-50.0°C	100.0°C	1768.0°C	2019.0°C	±1.0°C	±2.5°C
N	-270.0°C	-200.0°C	1300.0°C	1550.0°C	±1.0°C	±1.6°C
C	0.0°C	100.0°C	2315.0°C	2500.0°C	±0.7°C	±2.7°C
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 -80mV	27648 80mV	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。
- 若是暴露在 970 MHz 到 990 MHz 的无线电辐射频率下，SM 1231 AI 4 x 16 位 TC 的精度可能会有所下降。

表格 A- 145 SM 1231 热电偶的噪声消减和更新时间

抑制频率选择	积分时间	4 通道模块更新时间 (秒)	8 通道模块更新时间 (秒)
400 Hz (2.5 ms)	10 ms <sup>1</sup>	0.143	0.285
60 Hz (16.6 ms)	16.67 ms	0.223	0.445



抑制频率选择	积分时间	4 通道模块更新时间 (秒)	8 通道模块更新时间 (秒)
50 Hz (20 ms)	20 ms	0.263	0.525
10 Hz (100 ms)	100 ms	1.225	2.450

- <sup>1</sup> 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。

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。

### 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。

## J 型热电偶模拟值的表示

J 型热电偶模拟值的表示如下表所示。

表格 A- 146 J 型热电偶模拟值的表示

用 °C 表示 的 J 型	功能单元		用 °F 表示 的 J 型	功能单元		范围
	十进制	十六进制		十进制	十六进制	
> 1450.0	32767	7FFF	> 2642.0	32767	7FFF	溢出
1450.0	14500	38A4	2642.0	26420	6734	超出上限
:	:	:	:	:	:	
1200.1	12001	2EE1	2192.2	21922	55A2	额定范围
1200.0	12000	2EE0	2192.0	21920	55A0	
:	:	:	:	:	:	下溢 <sup>1</sup>
-150.0	-1500	FA24	-238.0	-2380	F6B4	
< -150.0	-32768	8000	< -238.0	-32768	8000	

- <sup>1</sup> 如果发生接线错误（例如极性接反或输入开路），或者传感器在负测量范围内出现故障（例如，热电偶类型错误），可能会导致热电偶模块信号超出下限。

A.8 热电偶和 RTD 信号模块 (SM)

A.8.2 SM 1231 RTD

SM 1231 RTD 规范

表格 A- 147 一般规格

技术数据	SM 1231 AI 4 x RTD x 16 位	SM 1231 AI 8 x RTD x 16 位
订货号	6ES7 231-5PD30-0XB0	6ES7 231-5PF30-0XB0
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 (24 VDC) <sup>1</sup>	40 mA	40 mA

<sup>1</sup> 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)

表格 A- 148 模拟量输入

技术数据	SM 1231 AI 4 x RTD x 16 位	SM 1231 AI 8 x RTD x 16 位
输入路数	4	8
类型	模块参考 RTD 和 $\Omega$	模块参考 RTD 和 $\Omega$
范围	请参见 RTD 传感器选型表 (页 857)。	请参见 RTD 传感器选型表 (页 857)。
额定范围 (数据字)		
过冲/下冲范围 (数据字)		
上溢/下溢 (数据字)		
分辨率	0.1° C/0.1° F	0.1° C/0.1° F
温度		
电阻	15 位 + 符号	15 位 + 符号
最大耐压	±35 V	±35 V
噪声抑制	对于所选噪声消减为 85 dB (10 Hz、50 Hz、60 Hz 或 400 Hz)	对于所选噪声消减为 85 dB (10 Hz、50 Hz、60 Hz 或 400 Hz)

技术数据	SM 1231 AI 4 x RTD x 16 位	SM 1231 AI 8 x RTD x 16 位
共模抑制	> 120dB	> 120dB
阻抗	≥ 10 MΩ	≥ 10 MΩ
隔离	现场侧与逻辑侧	500 VAC
	现场侧与 24 VDC	500 VAC
	24 VDC 与逻辑侧	500 VAC
通道间隔离	无	无
精度	请参见 RTD 传感器选型表 (页 857)。	请参见 RTD 传感器选型表 (页 857)。
可重复性	±0.05% FS	±0.05% FS
最大传感器功耗	0.5m W	0.5m W
测量原理	积分型	积分型
模块更新时间	请参见 噪声消减选项表 (页 857)。	请参见 噪声消减选项表 (页 857)。
电缆长度 (米)	到传感器最长为 100 米	到传感器最长为 100 米
导线电阻	20 Ω, 对于 10 Ω RTD, 最大为 2.7 Ω	20 Ω, 对于 10 Ω RTD, 最大为 2.7 Ω

表格 A- 149 诊断

技术数据	SM 1231 AI 4 x RTD x 16 位	SM 1231 AI 8 x RTD x 16 位
上溢/下溢 <sup>1,2</sup>	有	有
断线 <sup>3</sup>	有	有
24 VDC 低压 <sup>1</sup>	有	有

1 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。

2 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。

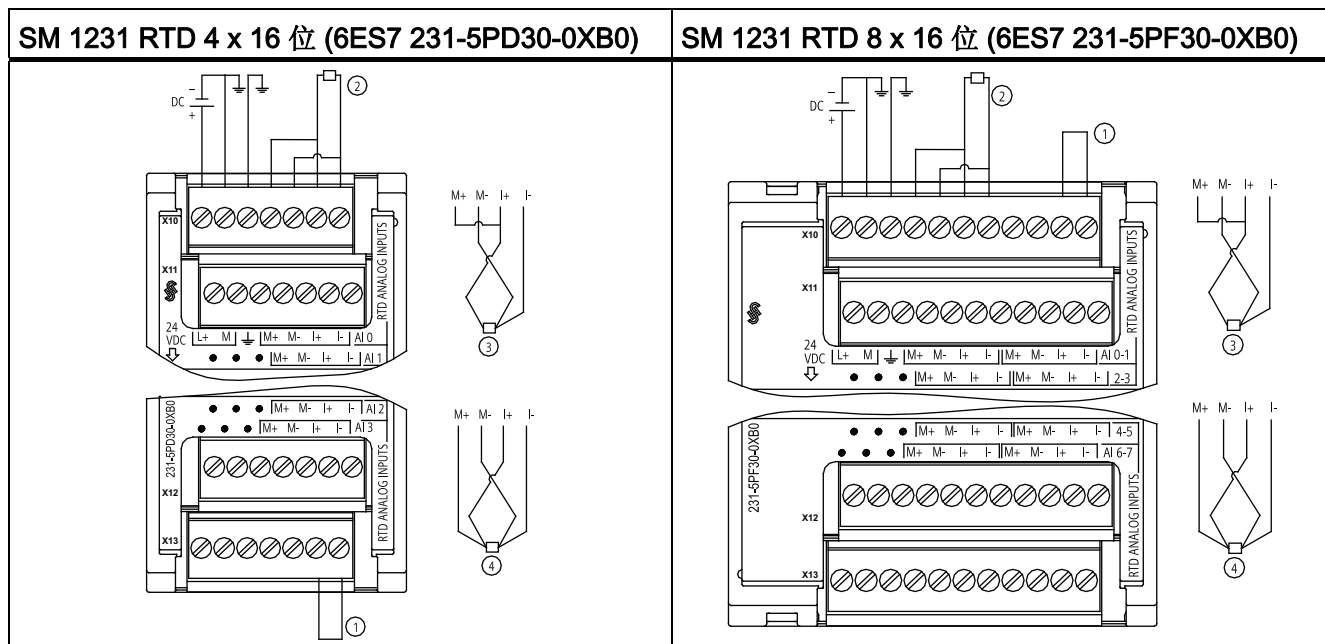
SM 1231 RTD 模拟量信号模块可测量连接到模块输入的电阻值。测量类型可选为“电阻”型或“热电阻”型。

- “电阻”： 额定范围的满量程值将是十进制数 27648。
- “热电阻”： 将度数乘 10 得到该值（例如，25.3 度将报告为十进制数 253）。

SM 1231 RTD 模块支持采用 2 线、3 线和 4 线制方式连接到传感器电阻进行测量。

A.8 热电偶和 RTD 信号模块 (SM)

表格 A- 150 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：注： 连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A- 151 SM 1231 RTD 4 x 16 位 (6ES7 231-5PD30-0XB0) 的连接器引脚位置

引脚	X10 (镀金)	X11 (镀金)	X12 (镀金)	X13 (镀金)
1	L+/24VDC	无连接	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	GND	无连接	无连接	无连接
4	AI 0 M+/RTD	AI 1 M+/RTD	AI 2 M+/RTD	AI 3 M+/RTD
5	AI 0 M-/RTD	AI 1 M-/RTD	AI 2 M-/RTD	AI 3 M-/RTD
6	AI 0 I+/RTD	AI 1 I+/RTD	AI 2 I+/RTD	AI 3 I+/RTD
7	AI 0 I-/RTD	AI 1 I-/RTD	AI 2 I-/RTD	AI 3 I-/RTD

表格 A- 152 SM 1231 RTD 8 x 16 位 (6ES7 231-5PF30-0XB0) 的连接器和针脚位置

针脚	X10 (镀金)	X11 (镀金)	X12 (镀金)	X13 (镀金)
1	L+/24VDC	无连接	无连接	无连接
2	M/24VDC	无连接	无连接	无连接
3	GND	无连接	无连接	无连接
4	AI 0 M+/RTD	AI 2 M+/RTD	AI 4 M+/RTD	AI 6 M+/RTD
5	AI 0 M-/RTD	AI 2 M-/RTD	AI 4 M-/RTD	AI 6 M-/RTD
6	AI 0 I+/RTD	AI 2 I+/RTD	AI 4 I+/RTD	AI 6 I+/RTD
7	AI 0 I-/RTD	AI 2 I-/RTD	AI 4 I-/RTD	AI 6 I-/RTD
8	AI 1 M+/RTD	AI 3 M+/RTD	AI 5 M+/RTD	AI 7 M+/RTD
9	AI 1 M-/RTD	AI 3 M-/RTD	AI 5 M-/RTD	AI 7 M-/RTD
10	AI 1 I+/RTD	AI 3 I+/RTD	AI 5 I+/RTD	AI 7 I+/RTD
11	AI 1 I-/RTD	AI 3 I-/RTD	AI 5 I-/RTD	AI 7 I-/RTD

**说明**

可以取消激活 RTD 的未使用通道。如果取消激活未使用的通道，不会出现任何错误。

RTD 模块需要使电流环不中断，以消除自动添加到未使用通道（未激活）的额外稳定时间。为保持一致性，RTD 模块应连接一个电阻（如 2 线制 RTD 连接）。

**A.8.2.1 SM 1231 RTD 选型表**

表格 A- 153 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 <sup>1</sup>	额定范围下限	额定范围上限	超出范围最大值 <sup>2</sup>	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 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						

A.8 热电偶和 RTD 信号模块 (SM)

温度系数	RTD 类型	低于范围最小值 <sup>1</sup>	额定范围下限	额定范围上限	超出范围最大值 <sup>2</sup>	25°C 时的额定范围精度	-20°C 到 60°C 时的额定范围精度
Pt 0.003902 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						
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- 154 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 <sup>1</sup>	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%

<sup>1</sup> 超出范围最大值以上的电阻值报告为 +32767。

### 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。

对于其它值较低的电阻使用 500 Ω 和 1000 Ω RTD 范围时，误差可能增加到指定误差的两倍。

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。

表格 A- 155 RTD 模块的噪声消减和更新时间

抑制频率选择	积分时间	更新时间 (秒)	
		4 通道模块	8 通道模块
400 Hz (2.5 ms)	10 ms <sup>1</sup>	4/2 线制: 0.142 3 线制: 0.285	4/2 线制: 0.285 3 线制: 0.525
60 Hz (16.6 ms)	16.67 ms	4/2 线制: 0.222 3 线制: 0.445	4/2 线制: 0.445 3 线制: 0.845
50 Hz (20 ms)	20 ms	4/2 线制: 0.262 3 线制: .505	4/2 线制: 0.524 3 线制: 1.015
10 Hz (100 ms)	100 ms	4/2 线制: 1.222 3 线制: 2.445	4/2 线制: 2.425 3 线制: 4.845

<sup>1</sup> 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。

A.8 热电偶和 RTD 信号模块 (SM)

**注意**

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。

**RTD 模拟值的表示**

RTD 标准温度范围传感器数字化测量值的表示如下表所示。

表格 A- 156 电阻温度计 PT 100、200、500、1000 和 PT 10、50、100、500 GOST (0.003850) 标准型的模拟值表示

Pt x00 标准型, 单位为 °C (1 位数字 = 0.1° C)	功能单元		Pt x00 标准型, 单位为 °F (1 位数字 = 0.1 F)	功能单元		范围
	十进制	十六进制		十进制	十六进制	
> 1000.0	32767	7FFF	> 1832.0	32767	7FFF	溢出
1000.0	10000	2710	1832.0	18320	4790	超出上限
:	:	:	:	:	:	
850.1	8501	2135	1562.1	15621	3D05	额定范围
850.0	8500	2134	1562.0	15620	3D04	
:	:	:	:	:	:	超出下限
-200.0	-2000	F830	-328.0	-3280	F330	
-200.1	-2001	F82F	-328.1	-3281	F32F	下溢
:	:	:	:	:	:	
-243.0	-2430	F682	-405.4	-4054	F02A	
< -243.0	-32768	8000	< -405.4	-32768	8000	



## A.9 数字信号板 (SB)

### A.9.1 SB 1221 200 kHz 数字量输入规范

表格 A- 157 常规规范

技术数据	SB 1221 DI 4 x 24 VDC, 200 kHz	SB 1221 DI 4 x 5 VDC, 200 kHz
订货号	6ES7 221-3BD30-0XB0	6ES7 221-3AD30-0XB0
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	1.5 W	1.0 W
电流消耗 (SM 总线)	40 mA	40 mA
电流消耗 (24 VDC)	7 mA/输入 + 20 mA	15 mA/输入 + 15 mA

表格 A- 158 数字量输入

技术数据	SB 1221 DI 4 x 24 VDC, 200 kHz	SB 1221 DI 4 x 5 VDC, 200 kHz
输入点数	4	4
类型	来源	来源
额定电压	7 mA 时 24 VDC, 额定值	15 mA 时 5 VDC, 额定值
允许的连续电压	28.8 VDC	6 VDC
浪涌电压	35 VDC, 持续 0.5 s	6 V
逻辑 1 信号 (最小)	2.9 mA 时 L+ - 10 VDC	5.1 mA 时 L+ - 2.0 VDC
逻辑 0 信号 (最大)	1.4 mA 时 L+ - 5 VDC	2.2 mA 时 L+ - 1.0 VDC
HSC 时钟输入频率 (最大)	单相: 200 kHz 正交相位: 160 kHz	单相: 200 kHz 正交相位: 160 kHz
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 分钟	500 VAC, 持续 1 分钟
隔离组	1	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

A.9 数字信号板 (SB)

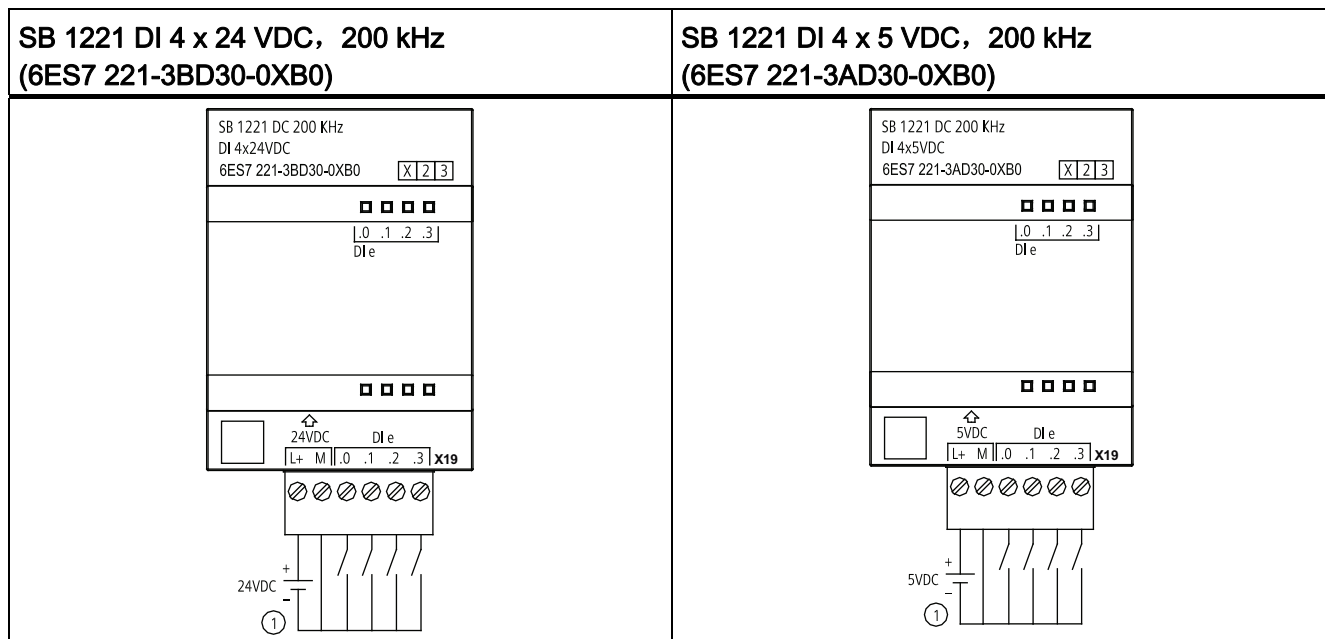
技术数据	SB 1221 DI 4 x 24 VDC, 200 kHz	SB 1221 DI 4 x 5 VDC, 200 kHz
同时接通的输入数	<ul style="list-style-type: none"> <li>• 2 (无相邻点), 60° C (水平) 或 50° C (垂直) 时</li> <li>• 4, 55° C (水平) 或 45° C (垂直) 时</li> </ul>	4
电缆长度 (米)	50 m 屏蔽双绞线	50 m 屏蔽双绞线

**注意**

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/组件电压从 24 V 降为 5 V
- 在输入端连接外部负载

表格 A- 159 200 kHz 数字量输入 SB 的接线图



① 仅支持源型输入

表格 A- 160 SB 1221 DI 4 x 24 VDC, 200 kHz (6ES7 221-3BD30-0XB0) 的连接器针脚位置

针脚	X19
1	L+/24VDC
2	M/24VDC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

表格 A- 161 SB 1221 DI 4 x 5 VDC, 200 kHz (6ES7 221-3AD30-0XB0) 的连接器针脚位置

针脚	X19
1	L+/5VDC
2	M/5VDC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

## A.9.2 SB 1222 200 kHz 数字量输出规范

表格 A- 162 常规规范

技术数据	SB 1222 DQ 4 x 24 VDC, 200 kHz	SB 1222 DQ 4 x 5 VDC, 200 kHz
订货号	6ES7 222-1BD30-0XB0	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	0.5 W	0.5 W
电流消耗 (SM 总线)	35 mA	35 mA
电流消耗 (24 VDC)	15 mA	15 mA

A.9 数字信号板 (SB)

表格 A- 163 数字量输出

技术数据	SB 1222 DQ 4 x 24 VDC, 200 kHz	SB 1222 DQ 4 x 5 VDC, 200 kHz
输出点数	4	4
输出类型	固态 - MOSFET 漏型和源型 <sup>1</sup>	固态 - MOSFET 漏型和源型 <sup>1</sup>
电压范围	20.4 到 28.8 VDC	4.25 到 6.0 VDC
最大电流时的逻辑 1 信号	L+ - 1.5 V	L+ - 0.7 V
最大电流时的逻辑 0 信号	最大 1.0 VDC	最大 0.2 VDC
电流 (最大)	0.1 A	0.1 A
灯负载	--	--
通态触点电阻	最大 11 Ω	最大 7 Ω
断态电阻	最大 6 Ω	最大 0.2 Ω
每点的漏泄电流	--	--
脉冲串输出频率	最大 200 kHz, 最小 2 Hz	最大 200 kHz, 最小 2 Hz
浪涌电流	0.11 A	0.11 A
过载保护	无	无
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 分钟	500 VAC, 持续 1 分钟
隔离组	1	1
每个公共端的电流	0.4 A	0.4 A
电感钳位电压	无	无
开关延迟	上升沿 1.5 μs + 300 ns 下降沿 1.5 μs + 300 ns	上升沿 200 ns + 300 ns 下降沿 200 ns + 300 ns
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)
同时接通的输出数	<ul style="list-style-type: none"> <li>• 2 (无相邻点), 60° C (水平) 或 50° C (垂直) 时</li> <li>• 4, 55° C (水平) 或 45° C (垂直) 时</li> </ul>	4
电缆长度 (米)	50 m 屏蔽双绞线	50 m 屏蔽双绞线

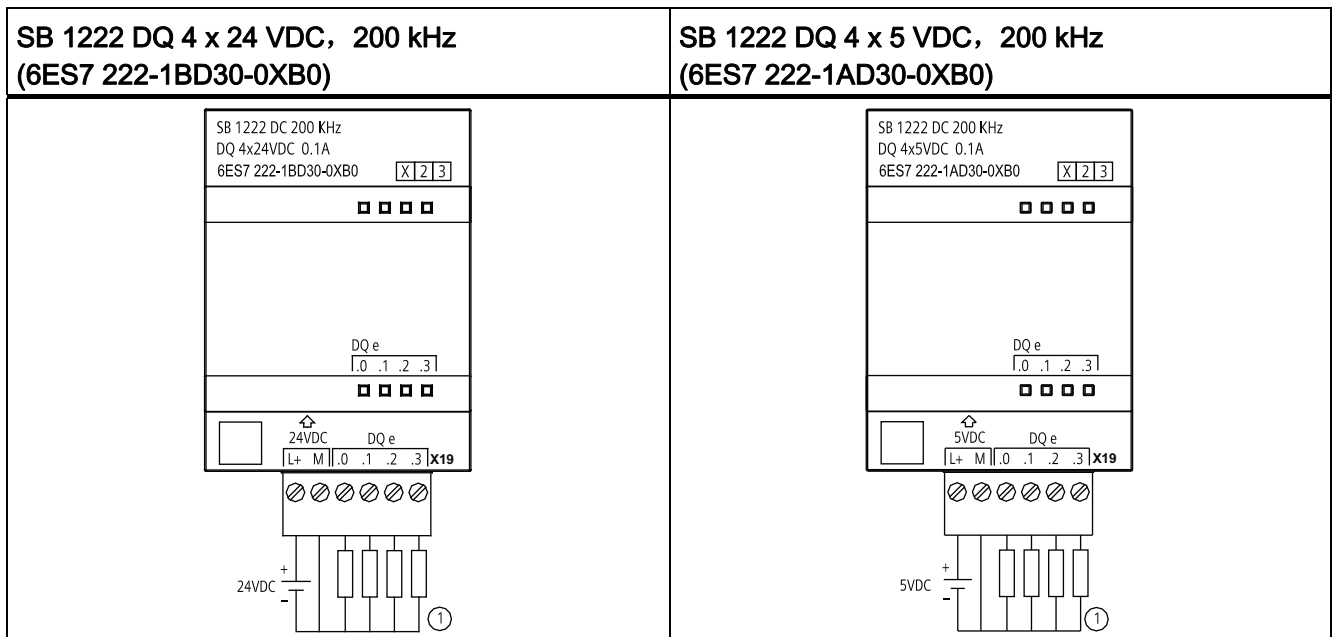
<sup>1</sup> 因为通过同一电路来支持漏型和源型配置, 所以源型负载的激活状态与漏型负载的相反。源型输出表现为正逻辑 (当负载有电流时, Q 位接通且 LED 亮起), 而漏型输出表现为负逻辑 (当负载有电流时, Q 位断开且 LED 熄灭)。如果插入模块且无用户程序, 则此模块的默认值是 0 V, 这意味着漏型负载将接通。

**注意**

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/组件电压从 24 V 降为 5 V
- 在输入端连接外部负载

表格 A- 164 200 kHz 数字量输出 SB 的接线图



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“+”。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.9 数字信号板 (SB)

表格 A- 165 SB 1222 DQ 4 x 24 VDC, 200 kHz (6ES7 222-1BD30-0XB0) 的连接器针脚位置

针脚	X19
1	L+/24VDC
2	M/24VDC
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

表格 A- 166 SB 1222 DQ 4 x 5 VDC, 200 kHz (6ES7 222-1AD30-0XB0) 的连接器针脚位置

针脚	X19
1	L+/5VDC
2	M/5VDC
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

A.9.3 SB 1223 200 kHz 数字量输入/输出规范

表格 A- 167 一般规格

技术数据	SB 1223 DI 2 x 24 VDC/ DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC/ DQ 2 x 5 VDC, 200 kHz
订货号	6ES7 223-3BD30-0XB0	6ES7 223-3AD30-0XB0
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	1.0 W	0.5 W
电流消耗 (SM 总线)	35 mA	35 mA
电流消耗 (24 VDC)	7 mA/输入 + 30 mA	15 mA/输入 + 15 mA

表格 A- 168 数字量输入

技术数据	SB 1223 DI 2 x 24 VDC/ DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC/ DQ 2 x 5 VDC, 200 kHz
输入路数	2	2
类型	来源	来源
额定电压	7 mA 时 24 VDC, 额定值	15 mA 时 5 VDC, 额定值
允许的连续电压	28.8 VDC	6 VDC
浪涌电压	35 VDC, 持续 0.5 s	6 V
逻辑 1 信号 (最小)	2.9 mA 时 L+ - 10 VDC	5.1 mA 时 L+ - 2.0 VDC
逻辑 0 信号 (最大)	1.4 mA 时 L+ - 5 VDC	2.2 mA 时 L+ - 1.0 VDC
HSC 时钟输入频率 (最大)	单相: 200 kHz 正交相位: 160 kHz	单相: 200 kHz 正交相位: 160 kHz
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 min	500 VAC, 持续 1 min
隔离组	1 (与输出无隔离)	1 (与输出无隔离)
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
同时接通的输入数	2	2
电缆长度 (米)	50 m 屏蔽双绞线	50 m 屏蔽双绞线

表格 A- 169 数字量输出

技术数据	SB 1223 DI 2 x 24 VDC/ DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC/ DQ 2 x 5 VDC, 200 kHz
输出路数	2	2
输出类型	固态 - MOSFET 漏型和源型 <sup>1</sup>	固态 - MOSFET 漏型和源型 <sup>1</sup>
电压范围	20.4 到 28.8 VDC	4.25 到 6.0 VDC
额定值	24 VDC	5 VDC
最大电流时的逻辑 1 信号	L+ - 1.5 V	L+ - 0.7 V
最大电流时的逻辑 0 信号	最大 1.0 VDC	最大 0.2 VDC

A.9 数字信号板 (SB)

技术数据	SB 1223 DI 2 x 24 VDC/ DQ 2 x 24 VDC, 200 kHz	SB 1223 DI 2 x 5 VDC/ DQ 2 x 5 VDC, 200 kHz
电流 (最大)	0.1 A	0.1 A
灯负载	--	--
通态触点电阻	最大 11 Ω	最大 7 Ω
断态电阻	最大 6 Ω	最大 0.2 Ω
每点的漏泄电流	--	--
脉冲串输出频率	最大 200 kHz, 最小 2 Hz	最大 200 kHz, 最小 2 Hz
浪涌电流	0.11 A	0.11 A
过载保护	否	否
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 min	500 VAC, 持续 1 min
隔离组	1 (与输入无隔离)	1 (与输入无隔离)
每个公共端的电流	0.2 A	0.2 A
电感钳位电压	否	否
开关延时	上升沿 1.5 μs + 300 ns 下降沿 1.5 μs + 300 ns	上升沿 200 ns + 300 ns 下降沿 200 ns + 300 ns
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)
同时接通的输出数	2	2
电缆长度 (米)	50 m 屏蔽双绞线	50 m 屏蔽双绞线

<sup>1</sup> 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。

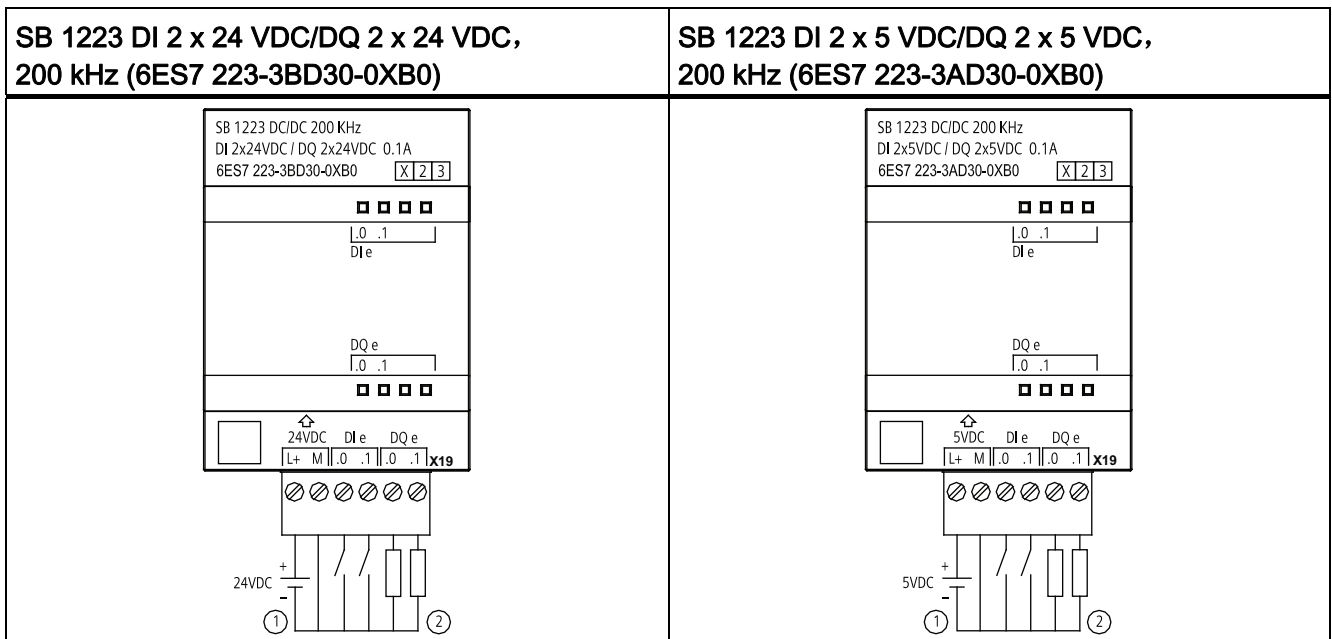


**注意**

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/组件电压从 24 V 降为 5 V
- 在输入端连接外部负载

表格 A- 170 200 kHz 数字量输入/输出 SB 的接线图



- ① 仅支持源型输入
- 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“+”。1 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.9 数字信号板 (SB)

表格 A- 171 SB 1223 DI 2 x 24 VDC/DQ 2 x 24 VDC, 200 kHz (6ES7 223-3BD30-0XB0) 的连接  
器针脚位置

针脚	X19
1	L+/24VDC
2	M/24VDC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

表格 A- 172 SB 1223 DI 2 x 5 VDC/DQ 2 x 5 VDC, 200 kHz (6ES7 223-3AD30-0XB0) 的连  
接器针脚位置

针脚	X19
1	L+/5VDC
2	M/5VDC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

**A.9.4 SB 1223 2 X 24 VDC 输入/2 X 24 VDC 输出规范**

表格 A- 173 一般规格

技术数据	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
订货号	6ES7 223-0BD30-0XB0
尺寸 W x H x D (mm)	38 x 62 x 21
重量	40 g
功耗	1.0 W

<b>技术数据</b>	<b>SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC</b>
电流消耗 (SM 总线)	50 mA
电流消耗 (24 VDC)	所用的每点输入 4 mA

表格 A- 174 数字量输入

<b>技术数据</b>	<b>SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC</b>
输入点数	2
类型	IEC 1 类漏型
额定电压	4 mA 时 24 VDC, 额定值
允许的连续电压	最大 30 VDC
浪涌电压	35 VDC, 持续 0.5 s
逻辑 1 信号 (最小)	2.5 mA 时 15 VDC
逻辑 0 信号 (最大)	1 mA 时 5 VDC
HSC 时钟输入频率 (最大)	20 kHz (15 到 30 VDC) 30 kHz (15 到 26 VDC)
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 min
隔离组	1
滤波时间	0.2、0.4、0.8、1.6、3.2、6.4 和 12.8 ms 可选择, 2 个为一组
同时接通的输入数	2
电缆长度 (米)	500 (屏蔽); 300 (非屏蔽)

表格 A- 175 数字量输出

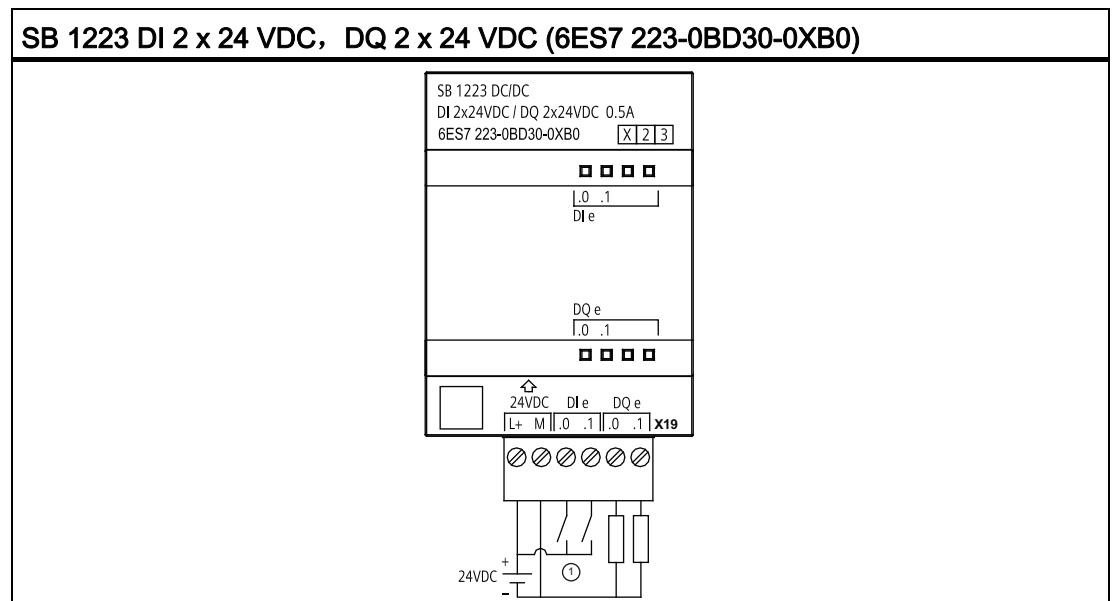
<b>技术数据</b>	<b>SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC</b>
输出点数	2
输出类型	固态 - MOSFET (源型)
电压范围	20.4 到 28.8 VDC
最大电流时的逻辑 1 信号	最小 20 VDC
具有 10 K $\Omega$ 负载时的逻辑 0 信号	最大 0.1 VDC

A.9 数字信号板 (SB)

技术数据	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
电流 (最大)	0.5 A
灯负载	5 W
通态触点电阻	最大 0.6 Ω
每点的漏泄电流	最大 10 μA
脉冲串输出 (PTO) 频率	最大 20 KHz, 最小 2 Hz <sup>1</sup>
浪涌电流	5 A, 最长持续 100 ms
过载保护	无
隔离 (现场侧与逻辑侧)	500 VAC, 持续 1 min
隔离组	1
每个公共端的电流	1 A
电感钳位电压	L+ - 48 V, 1 W 损耗
开关延迟	断开到接通最长为 2 μs 接通到断开最长为 10 μs
RUN 到 STOP 时的行为	上一个值或替换值 (默认值为 0)
同时接通的输出数	2
电缆长度 (米)	500 m (屏蔽); 150 m (非屏蔽)

<sup>1</sup> 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。

表格 A- 176 数字量输入/输出 SB 的接线图



① 支持漏型输入

表格 A- 177 SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC (6ES7 223-0BD30-0XB0) 的连接器针脚位置

针脚	X19
1	L+/24VDC
2	M/24VDC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

A.10 模拟信号板 (SB)

**A.10 模拟信号板 (SB)**

**A.10.1 SB 1231 1 路模拟量输入规范**

**说明**

要使用此 SB，CPU 固件必须为 V2.0 或更高版本。

表格 A- 178 常规规范

技术数据	SB 1231 AI 1 x 12 位
订货号	6ES7 231-4HA30-0XB0
尺寸 W x H x D (mm)	38 x 62 x 21
重量	35 g
功耗	0.4 W
电流消耗 (SM 总线)	55 mA
电流消耗 (24 VDC)	否

表格 A- 179 模拟量输入

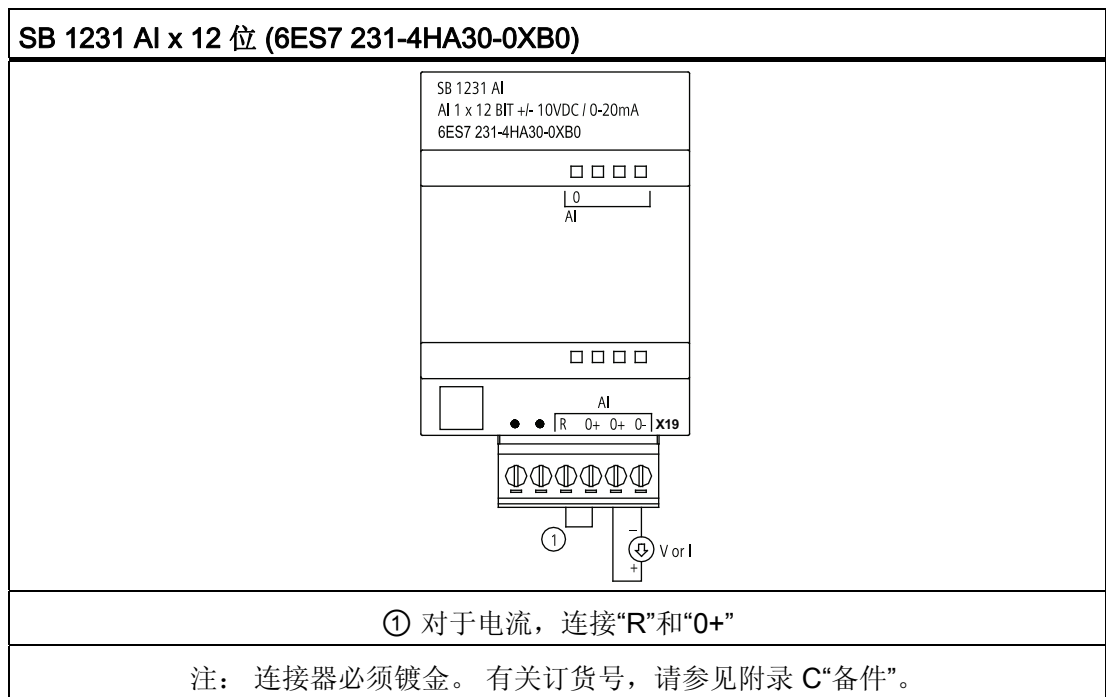
技术数据	SB 1231 AI 1 x 12 位
输入路数	1
类型	电压或电流 (差动)
范围	±10V、±5V、±2.5 或 0 到 20 mA
分辨率	11 位 + 符号位
满量程范围 (数据字)	-27,648 到 27,648
超出/低于范围 (数据字)	电压: 32,511 到 27,649/-27,649 到 -32,512 电流: 32,511 到 27,649/0 到 -4,864 (请参见模拟量输入的电压表示法以及模拟量输入的电流表示法。)

技术数据	SB 1231 AI 1 x 12 位
上溢/下溢 (数据字)	电压: 32,767 到 32,512/-32,513 到 -32,768 电流: 32,767 到 32,512/-4,865 到 -32,768 (请参见模拟量输入的电压表示法以及模拟量输入的电流表示法。)
最大耐压/耐流	±35V/±40 mA
平滑化	无、弱、中或强 (请参见模拟量输入的响应时间以了解阶跃响应时间。)
噪声抑制	400、60、50 或 10 Hz (请参考模拟输入的响应时间以了解采样速率。)
精度 (25°C/-20 到 60°C)	满量程的 ±0.3%/±0.6%
输入阻抗	
差模	电压: 220 kΩ; 电流: 250 Ω
共模	电压: 55 kΩ; 电流: 55 kΩ
RUN 到 STOP 时的行为	上一个值或替换值 (默认值为 0)
测量原理	实际值转换
共模抑制	400 dB, DC 到 60 Hz
工作信号范围	信号加共模电压必须小于 +35 V 且大于 -35 V
隔离 (现场侧与逻辑侧)	无
电缆长度 (米)	100 m, 屏蔽双绞线

表格 A- 180 诊断

技术数据	SB 1231 AI 1 x 12 位
上溢/下溢	是
24 VDC 低压	否

表格 A- 181 模拟量输入 SB 的接线图



表格 A- 182 SB 1231 AI x 12 位 (6ES7 231-4HA30-0XB0) 的连接器引脚位置

引脚	X19 (镀金)
1	无连接
2	无连接
3	AI R
4	AI 0+
5	AI 0+
6	AI 0-



## A.10.2 SB 1232 1 路模拟量输出规范

表格 A- 183 常规规范

技术数据	SB 1232 AQ 1 x 12 位
订货号	6ES7 232-4HA30-0XB0
尺寸 W x H x D (mm)	38 x 62 x 21
重量	40 g
功耗	1.5 W
电流消耗 (SM 总线)	15 mA
电流消耗 (24 VDC)	40 mA (无负载)

表格 A- 184 模拟量输出

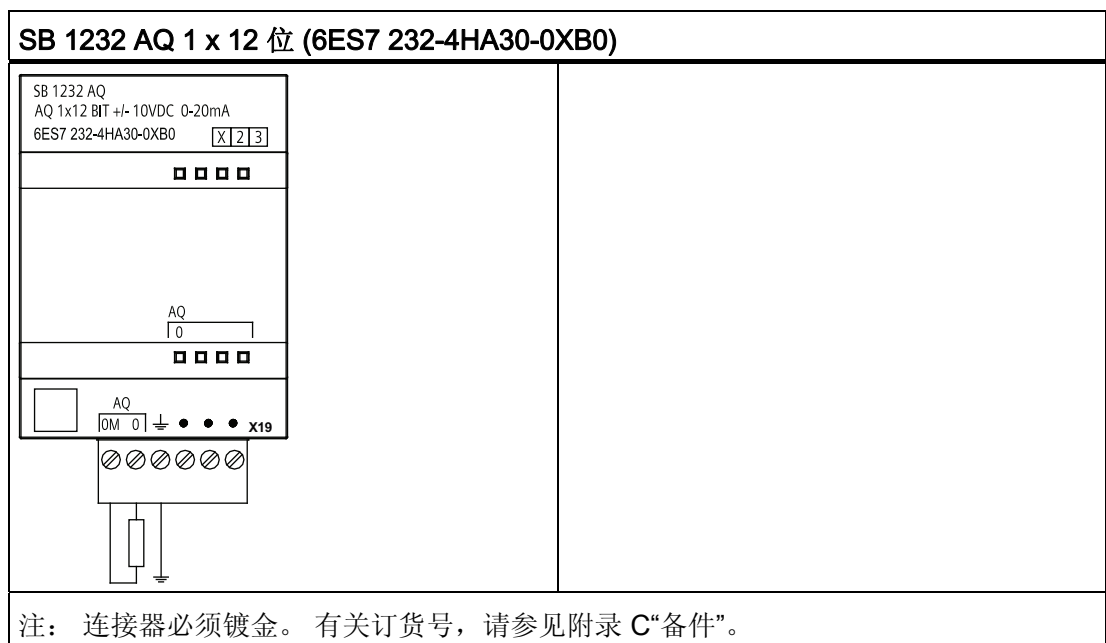
技术数据	SB 1232 AQ 1 x 12 位
输出路数	1
类型	电压或电流
范围	$\pm 10$ V 或 0 到 20 mA
分辨率	电压: 12 位 电流: 11 位
满量程范围 (数据字)	电压: -27,648 到 27,648
请参见 电压和电流 (页 881) 的输出范围。	电流: 0 到 27,648
精度 (25°C/-20 到 60°C)	满量程的 $\pm 0.5\%/\pm 1\%$
稳定时间 (新值的 95%)	电压: 300 $\mu$ S (R), 750 $\mu$ S (1 uF) 电流: 600 $\mu$ S (1 mH), 2 ms (10 mH)
负载阻抗	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)
隔离 (现场侧与逻辑侧)	无
电缆长度 (米)	100 m, 屏蔽双绞线

A.10 模拟信号板 (SB)

表格 A- 185 诊断

技术数据	SB 1232 AQ 1 x 12 位
上溢/下溢	有
对地短路（仅限电压模式）	有
断路（仅限电流模式）	有

表格 A- 186 SB 1232 AQ 1 x 12 位的接线图



表格 A- 187 SB 1232 AQ 1 x 12 位 (6ES7 232-4HA30-0XB0) 的连接器针脚位置

针脚	X19（镀金）
1	AQ 0M
2	AQ 0
3	GND
4	无连接
5	无连接
6	无连接

### A.10.3 模拟量输入和输出的测量范围

#### A.10.3.1 模拟量输入的阶跃响应

表格 A- 188 阶跃响应 (ms), 0V 到 10V (在 95% 处测得)

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	4.5 ms	18.7 ms	22.0 ms	102 ms
弱 (4 个周期): 4 次采样	10.6 ms	59.3 ms	70.8 ms	346 ms
中 (16 个周期): 16 次采样	33.0 ms	208 ms	250 ms	1240 ms
强 (32 个周期): 32 次采样	63.0 ms	408 ms	490 ms	2440 ms
采样时间	0.156 ms	1.042 ms	1.250 ms	6.250 ms

#### A.10.3.2 模拟量输入的采样时间和更新时间

表格 A- 189 采样时间和更新时间

选项	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.10 模拟信号板 (SB)

A.10.3.3 模拟量输入的电压测量范围

表格 A- 190 模拟量输入的电压表示法

系统		电压测量范围						
十进制	十六进制	±10 V	±5 V	±2.5 V	±1.25V		0 到 10 V	
32767	7FFF	11.851 V	5.926 V	2.963 V	1.481 V	上溢	11.851 V	上溢
32512	7F00							
32511	7EFF	11.759 V	5.879 V	2.940 V	1.470 V	过冲范围	11.759 V	过冲范围
27649	6C01							
27648	6C00	10 V	5 V	2.5 V	1.250 V	额定范围	10 V	额定范围
20736	5100	7.5 V	3.75 V	1.875 V	0.938 V		7.5 V	
1	1	361.7 μV	180.8 μV	90.4 μV	45.2 μV		361.7 μV	
0	0	0 V	0 V	0 V	0 V		0 V	
-1	FFFF						不支持负值	
-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			
-32513	80FF					下溢		
-32768	8000	-11.851 V	-5.926 V	-2.963 V	-1.481 V			

## A.10.3.4 模拟量输入的电流测量范围

表格 A- 191 模拟量输入的电流表示法

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			

## A.10.3.5 输出 (AQ) 的电压和电流测量范围 (SB 和 SM)

表格 A- 192 模拟量输出的电压表示法

系统		电压输出范围	
十进制	十六进制	$\pm 10$ 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 $\mu$ V	

A.10 模拟信号板 (SB)

系统		电压输出范围	
十进制	十六进制	±10 V	
0	0	0 V	
-1	FFFF	-361.7 μV	
-20736	AF00	-7.5 V	
-27648	9400	-10 V	
-27649	93FF		下冲范围
-32512	8100	-11.76 V	下溢
-32513	80FF	请参见注 1	
-32768	8000	请参见注 1	

1 在上溢或下溢情况下，模拟量输出将根据为模拟量信号模块设置的设备配置属性动作。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。

表格 A- 193 模拟量输出的电流表示法

系统		电流输出范围	
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		
27648	6C00	20 mA	额定范围
20736	5100	15 mA	
1	1	723.4 nA	
0	0	0 mA	

1 在上溢或下溢情况下，模拟量输出将根据为模拟量信号模块设置的设备配置属性动作。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.10.4 热电偶 SB

### A.10.4.1 SB 1231 1 路热电偶模拟量输入规范

#### 说明

要使用此 SB，CPU 固件必须为 V2.0 或更高版本。

表格 A- 194 常规规范

技术数据	SB 1231 AI 1 x 16 位热电偶
订货号	6ES7 231-5QA30-0XB0
尺寸 W x H x D (mm)	38 x 62 x 21
重量	35 g
功耗	0.5 W
电流消耗 (SM 总线)	5 mA
电流消耗 (24 VDC)	20 mA

表格 A- 195 模拟量输入

技术数据	SB 1231 AI 1x16 位热电偶
输入路数	1
类型	浮动 TC 和 mV
范围	请参见 热电偶滤波器选型表 (页 884)。
<ul style="list-style-type: none"> <li>• 额定范围 (数据字)</li> <li>• 过量程/欠量程 (数据字)</li> <li>• 上溢/下溢 (数据字)</li> </ul>	
分辨率	温度 0.1° C/0.1° F
	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、400 Hz) 为 85 dB

A.10 模拟信号板 (SB)

技术数据	SB 1231 AI 1x16 位热电偶
共模抑制	120 VAC 时大于 120 dB
阻抗	$\geq 10 \text{ M } \Omega$
精度	请参见 热电偶选型表 (页 884)。
可重复性	$\pm 0.05\% \text{ FS}$
测量原理	积分型
模块更新时间	请参见 热电偶滤波器选型表 (页 884)。
冷端误差	$\pm 1.5^\circ \text{ C}$
隔离 (现场侧与逻辑侧)	500 VAC
电缆长度 (米)	到传感器最长为 100 m
导线电阻	最大 100 $\Omega$

表格 A- 196 诊断

技术数据	SB 1231 AI 1 x 16 位热电偶
上溢/下溢 <sup>1</sup>	有
断线 <sup>2</sup>	有

- 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 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。

SM 1231 热电偶 (TC) 模拟量信号模块可测量连接到模块输入的电压值。

SB 1231 热电偶模拟信号板可测量连接到信号板输入的电压值。温度测量类型可以是“热电偶”或“电压”类型。

- “热电偶”：将度数乘 10 得到该值（例如，25.3 度将报告为十进制数 253）。
- “电压”：额定范围的满量程值将是十进制数 27648。

#### A.10.4.2 热电偶的基本操作

两种不同的金属彼此之间存在电气连接时，便会形成热电偶。热电偶产生的电压与结点温度成正比。电压很小；一微伏能表示很多度。测量热电偶产生的电压，对额外的结点进行补偿，然后将测量结果线性化，这些是使用热电偶测量温度的基础。

将热电偶连接到 SM 1231 热电偶模块时，两条不同的金属线需连接到模块的信号连接器上。这两条不同的金属线互相连接的位置即形成了传感器热电偶。



在这两条不同的金属线与信号连接器相连的位置，构成了另外二个热电偶。连接器温度会引起一定的电压，该电压将添加到传感器热电偶产生的电压中。如果不对该电压进行修正，结果报告的温度将偏离传感器温度。

冷端补偿便是用于对连接器热电偶进行补偿。热电偶表是基于参比端温度（通常是零摄氏度）得来的。冷端补偿用于将连接器温度修正为零摄氏度。冷端补偿可消除连接器热电偶增加的电压。模块的温度在内部测量，然后转换为数值并添加到传感器换算中。之后是使用热电偶表对修正后的传感器换算值进行线性化。

为使冷端补偿取得最佳效果，必须将热电偶模块安装在温度稳定的环境中。符合模块规范的模块环境温度的缓慢变化（低于  $0.1^{\circ}\text{C}/\text{min}$ ）能够被正确补偿。穿过模块的空气流动也会引起冷端补偿误差。

如果需要更佳的冷端误差补偿效果，则可使用外部 iso 热端子块。热电偶模块可以使用  $0^{\circ}\text{C}$  基准值或  $50^{\circ}\text{C}$  基准值端子块。

### SB 1231 热电偶选型表

下表给出了 SB 1231 热电偶信号板支持的不同热电偶类型对应的测量范围和精度。

表格 A- 197 SB 1231 热电偶选型表

热电偶类型	低于范围最小值 <sup>1</sup>	额定范围下限	额定范围上限	超出范围最大值 <sup>2</sup>	25°C 时的额定范围 <sup>3</sup> 精度	-20°C 到 60°C 时的额定范围 <sup>3</sup> 精度
J	-210.0°C	-150.0°C	1200.0°C	1450.0°C	±0.3°C	±0.6°C
K	-270.0°C	-200.0°C	1372.0°C	1622.0°C	±0.4°C	±1.0°C
T	-270.0°C	-200.0°C	400.0°C	540.0°C	±0.5°C	±1.0°C
E	-270.0°C	-200.0°C	1000.0°C	1200.0°C	±0.3°C	±0.6°C
R & S	-50.0°C	100.0°C	1768.0°C	2019.0°C	±1.0°C	±2.5°C
N	-270.0°C	0.0°C	1300.0°C	1550.0°C	±1.0°C	±1.6°C
C	0.0°C	100.0°C	2315.0°C	2500.0°C	±0.7°C	±2.7°C

A.10 模拟信号板 (SB)

热电偶类型	低于范围最小值 <sup>1</sup>	额定范围下限	额定范围上限	超出范围最大值 <sup>2</sup>	25°C 时的额定范围 <sup>3</sup> 精度	-20°C 到 60°C 时的额定范围 <sup>3</sup> 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	-32511	-27648 -80mV	27648 80mV	32511	±0.05%	±0.1%

- 1 “低于范围最小值”以下的热电偶值报告为 -32768。
- 2 “超出范围最大值”以上的热电偶值报告为 32767。
- 3 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。信号板需要至少 30 分钟的预热时间才能满足该规范。

表格 A- 198 SB 1231 热电偶的滤波器选型表

抑制频率 (Hz)	积分时间 (ms)	信号板更新时间 (秒)
10	100	0.306
50	20	0.066
60	16.67	0.056
400 <sup>1</sup>	10	0.036

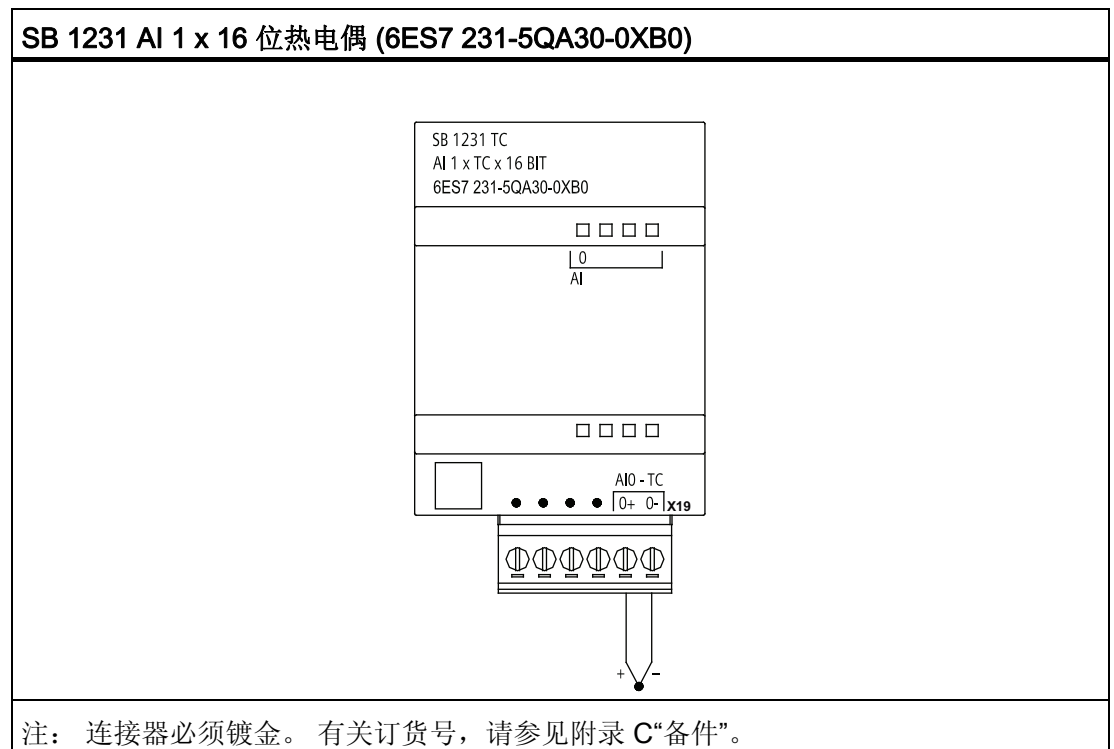
- 1 在选择 400 Hz 抑制频率时，要维持模块的分辨率和精度，积分时间应为 10 ms。该选择还可抑制 100 Hz 和 200 Hz 的噪声。

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。

**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。

表格 A- 199 模拟量输入热电偶 SB 的接线图



表格 A- 200 SB 1231 AI 1 x 16 位热电偶 (6ES7 231-5QA30-0XB0) 的连接器针脚位置

针脚	X19 (镀金)
1	无连接
2	无连接
3	无连接
4	无连接
5	AI 0-/TC
6	AI 0+/TC

## A.10.5 RTD SB

### A.10.5.1 SB 1231 1 路模拟量 RTD 输入的规范

#### 说明

要使用此 SB，CPU 固件必须为 V2.0 或更高版本。

A.10 模拟信号板 (SB)

表格 A- 201 常规规范

技术数据	SB 1231 AI 1 x 16 位 RTD
订货号	6ES7 231-5PA30-0XB0
尺寸 W x H x D (mm)	38 x 62 x 2
重量	35 g
功耗	0.7 W
电流消耗 (SM 总线)	5 mA
电流消耗 (24 VDC)	25 mA

表格 A- 202 模拟量输入

技术数据	SB 1231 AI 1 x 16 位 RTD				
输入路数	1				
类型	模块参考 RTD 和 $\Omega$				
范围	请参见 选型表 (页 891)。				
<ul style="list-style-type: none"> <li>• 额定范围 (数据字)</li> <li>• 过量程/欠量程 (数据字)</li> <li>• 上溢/下溢 (数据字)</li> </ul>					
分辨率	<table border="0" style="width: 100%;"> <tr> <td style="width: 30%; text-align: right;">温度</td> <td>0.1° C/0.1° F</td> </tr> <tr> <td style="text-align: right;">电压</td> <td>15 位 + 符号</td> </tr> </table>	温度	0.1° C/0.1° F	电压	15 位 + 符号
温度	0.1° C/0.1° F				
电压	15 位 + 符号				
最大耐压	±35 V				
噪声抑制	85 dB (10 Hz、50 Hz、60 Hz、400 Hz)				
共模抑制	> 120 dB				
阻抗	≥ 10 M $\Omega$				
精度	请参见 选型表 (页 891)。				
可重复性	±0.05% FS				
最大传感器功耗	0.5 m W				
测量原理	积分型				
模块更新时间	请参见 选型表 (页 891)。				
隔离 (现场侧与逻辑侧)	500 VAC				

技术数据	SB 1231 AI 1 x 16 位 RTD
电缆长度 (米)	到传感器最长为 100 m
导线电阻	20 $\Omega$ , 对于 10 $\Omega$ RTD, 最大为 2.7 $\Omega$

表格 A- 203 诊断

技术数据	SB 1231 AI 1 x 16 位 RTD
上溢/下溢 <sup>1, 2</sup>	有
断线 <sup>3</sup>	有

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 对于电阻范围, 始终会禁用下溢检测。

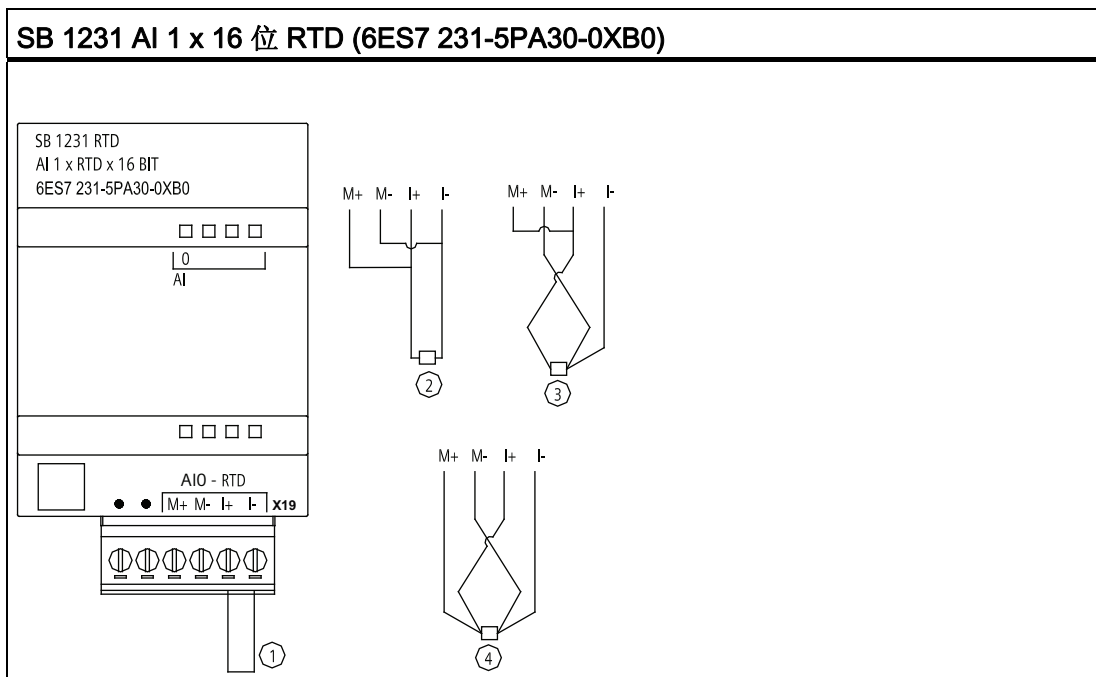
3 如果断线报警已禁用, 但传感器接线存在开路情况, 则模块可能会报告随机值。

SM 1231 RTD 模拟信号板可测量连接到信号板输入的电阻值。测量类型可选为“电阻”型或“热电阻”型。

- “电阻”: 额定范围的满量程值将是十进制数 27648。
- “热电阻”: 将度数乘 10 得到该值 (例如, 25.3 度将报告为十进制数 253)。

SB 1231 RTD 信号板支持采用 2 线、3 线和 4 线制方式连接到传感器电阻进行测量。

表格 A- 204 SB 1231 AI 1 x 16 位 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

注：连接器必须镀金。有关订货号，请参见附录 C“备件”。

表格 A- 205 SB 1231 AI 1 x 16 位 RTD (6ES7 231-5PA30-0XB0) 的连接器引脚位置

引脚	X19 (镀金)
1	无连接
2	无连接
3	AI 0 M+/RTD
4	AI 0 M-/RTD
5	AI 0 I+/RTD
6	AI 0 I-/RTD

## A.10.5.2 SB 1231 RTD 选型表

表格 A- 206 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 <sup>1</sup>	额定范围下限	额定范围上限	超出范围最大值 <sup>2</sup>	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 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						
	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						
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

A.10 模拟信号板 (SB)

温度系数	RTD 类型	低于范围最小值 <sup>1</sup>	额定范围下限	额定范围上限	超出范围最大值 <sup>2</sup>	25°C 时的额定范围精度	-20°C 到 60°C 时的额定范围精度
	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 +32768。

表格 A- 207 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 <sup>1</sup>	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。

**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。

对于其它值较低的电阻使用 500 Ω 和 1000 Ω RTD 范围时，误差可能增加到指定误差的两倍。

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。



表格 A- 208 RTD 模块的噪声消减和更新时间

抑制频率选择	积分时间	4/2 线制, 1 通道模块 更新时间 (秒)	3 线制, 1 通道模块 更新时间 (秒)
400 Hz (2.5 ms)	10 ms <sup>1</sup>	0.036	0.071
60 Hz (16.6 ms)	16.67 ms	0.056	0.111
50 Hz (20 ms)	20 ms	0.066	1.086
10 Hz (100 ms)	100 ms	0.306	0.611

- <sup>1</sup> 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。

**注意**

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.11 BB 1297 电池板

### BB 1297 电池板

S7-1200 BB 1297 电池板适用于实时时钟的长期备份。它可插入 S7-1200 CPU (固件版本 3.0 及更高版本) 的单个板插槽中。必须将 BB 1297 添加到设备组态并将硬件配置下载到 CPU 中, BB 才能正常工作。

电池 (型号 CR1025) 未随 BB 1297 一起提供, 必须由用户另行购买。

**说明**

BB 1297 在机械设计上适合固件版本为 3.0 及以上版本的 CPU。

不要将 BB 1297 与较早版本的 CPU 一起使用, 因为 BB 1297 连接器无法插入 CPU 中。

A.11 BB 1297 电池板

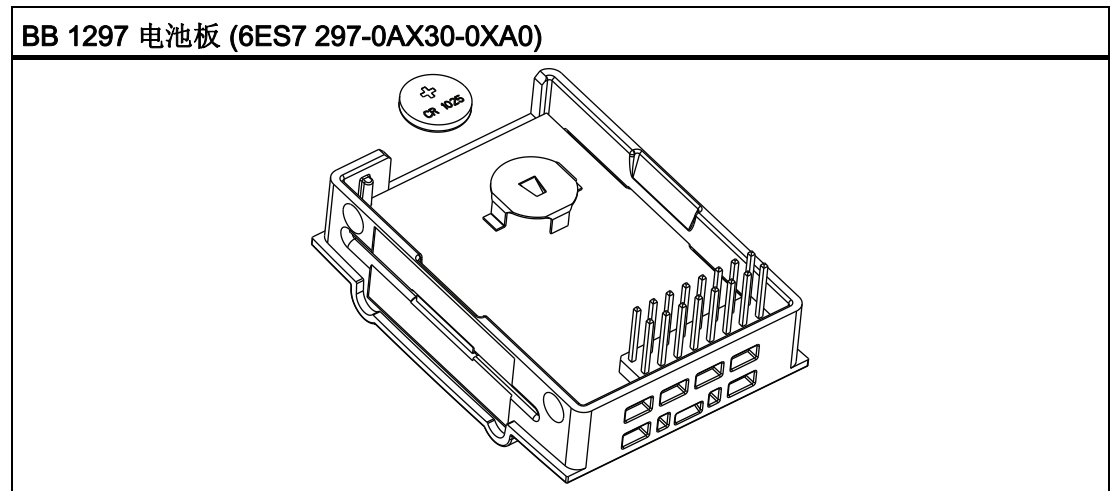
表格 A- 209 常规规范

技术数据	BB 1297 电池板
订货号	6ES7 297-0AX30-0XA0
尺寸 W x H x D (mm)	38 x 62 x 21
重量	28 g
功耗	0.5 W
电流消耗 (SM 总线)	11 mA
电流消耗 (24 VDC)	否

电池 (未包含)	BB 1297 电池板
保持时间	大约 1 年
电池类型	CR1025, 请参见 安装或更换 BB 1297 电池板中的电池 (页 51)
额定电压	3 V
额定容量	至少 30 mAh

诊断	BB 1297 电池板
临界电池电压	< 2.5 V
电池诊断	低压指示灯: <ul style="list-style-type: none"> <li>• 电池电压低会使 CPU MAINT LED 呈琥珀色常亮。</li> <li>• 诊断缓冲区事件: 16#06:2700“需要子模块维护: 至少一个电池已耗尽 (BATTF)”</li> </ul>
电池状态	提供的电池状态位 0 = 电池正常 1 = 电池电量低
电池状态更新	电池状态会在开机时更新, 之后在 CPU 处于 RUN 模式时, 每天更新一次。

表格 A- 210 BB 1297 电池板插件图



## A.12 通信接口

### A.12.1 PROFIBUS

#### A.12.1.1 CM 1242-5

表格 A- 211 CM 1242-5 的技术规范

技术规范	
订货号	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，光学网络组件）时 PROFIBUS 接口上的最大电流消耗	5 V 时为 15 mA（仅适用于总线终端*）
允许的环境条件	

---

**技术规范**

---

环境温度

- |                          |                  |
|--------------------------|------------------|
| • 存储期间                   | • -40 °C 到 70 °C |
| • 运输期间                   | • -40 °C 到 70 °C |
| • 垂直安装 (DIN 导轨水平放置) 运行期间 | • 0 °C 到 55 °C   |
| • 水平安装 (DIN 导轨垂直放置) 运行期间 | • 0 °C 到 45 °C   |
- 

在 25 °C 下运行而不会结露的最大相对湿度	95 %
-------------------------	------

---

防护等级	IP20
------	------

---

**电源、电流消耗和功耗**

---

电源类型	DC
------	----

---

来自背板总线的电力供应	5 V
-------------	-----

---

电流消耗 (典型值)	150 mA
------------	--------

---

有效功耗 (典型值)	0.75 W
------------	--------

---

**尺寸和重量**

---

- |      |          |
|------|----------|
| • 宽度 | • 30 mm  |
| • 高度 | • 100 mm |
| • 深度 | • 75 mm  |
- 

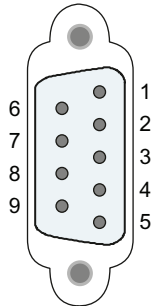
重量

- |          |         |
|----------|---------|
| • 净重     | • 115 g |
| • 含包装的重量 | • 152 g |
- 

\*) VP (针脚 6) 和 DGND (针脚 5) 间连接的外部用电器的电流负载不得超过总线终端允许的 15 mA 这一最大电流值 (防短路)。

## PROFIBUS 接口

表格 A- 212 D 型插座的引脚分配



引脚	说明	引脚	说明
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
4	RTS	9	- 未使用 -
5	M5V2: 数据参考电位 (接地 DGND)	外壳	接地线

## A.12.1.2 CM 1243-5

表格 A- 213 CM 1243-5 的技术规范

技术规范	
订货号	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, 光学网络组件) 时 PROFIBUS 接口上的最大电流消耗	5 V 时为 15 mA (仅适用于总线终端) *)
允许的环境条件	
环境温度	
<ul style="list-style-type: none"> <li>• 存储期间</li> <li>• 运输期间</li> <li>• 垂直安装 (DIN 导轨水平放置) 运行期间</li> <li>• 水平安装 (DIN 导轨垂直放置) 运行期间</li> </ul>	<ul style="list-style-type: none"> <li>• -40 °C 到 70 °C</li> <li>• -40 °C 到 70 °C</li> <li>• 0 °C 到 55 °C</li> <li>• 0 °C 到 45 °C</li> </ul>
在 25 °C 下运行而不会结露的最大相对湿度	95 %
防护等级	IP20
电源、电流消耗和功耗	
电源类型	DC

---

**技术规范**

---

电源/外部	24 V
• 最小值	• 19.2 V
• 最大值	• 28.8 V

---

电流消耗（典型值）	
• 通过 24 V DC	• 100 mA
• 通过 S7-1200 背板总线	• 0 mA

---

有效功耗（典型值）	
• 通过 24 V DC	• 2.4 W
• 通过 S7-1200 背板总线	• 0 W

---

24 VDC 电源/外部	
• 最小电缆横截面积	• 最小值: 0.14 mm <sup>2</sup> (AWG 25)
• 最大电缆横截面积	• 最大值: 1.5 mm <sup>2</sup> (AWG 15)
• 螺钉端子的拧紧扭矩	• 0.45 Nm (4 lb-in)

---

**尺寸和重量**

---

• 宽度	• 30 mm
• 高度	• 100 mm
• 深度	• 75 mm

---

重量	
• 净重	• 134 g
• 含包装的重量	• 171 g

---

\*) VP（针脚 6）和 DGND（针脚 5）间连接的外部用电器的电流负载不得超过总线终端允许的 15 mA 这一最大电流值（防短路）。

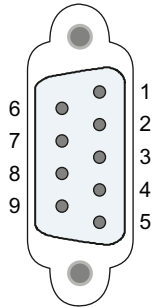
---

**说明**

CM 1243-5（PROFIBUS 主站模块）必须由 CPU 的 24 VDC 传感器电源供电。

---

## PROFIBUS 接口



表格 A- 214 D 型插座的引脚分配

引脚	说明	引脚	说明
1	- 未使用 -	6	VP: +5 V 电源, 仅用于总线终端电阻; 不用于为外部设备供电
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 接地	外壳	接地线

## PROFIBUS 电缆

### 注意

#### 连接 PROFIBUS 电缆的屏蔽层

必须连接 PROFIBUS 电缆的屏蔽层。

为此, 从 PROFIBUS 电缆末端剥去一段绝缘层, 将屏蔽层连接到功能地。

## A.12.2 GPRS

### 说明

#### GPRS CP 未经海事应用认证

以下模块未经海事认证:

- CP 1242-7 GPRS 模块

### 说明

要使用这些模块, CPU 固件必须为 V2.0 或更高版本。

### A.12.2.1 CP 1242-7

表格 A- 215 CP 1242-7 的技术规范

技术规范	
订货号	6GK7 242-7KX30-0XE0
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"> <li>• GSM 850, 类别 4: +33 dBm ±2dBm</li> <li>• GSM 900, 类别 4: +33 dBm ±2dBm</li> <li>• GSM 1800, 类别 1: +30 dBm ±2dBm</li> <li>• GSM 1900, 类别 1: +30 dBm ±2dBm</li> </ul>
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	
环境温度	<ul style="list-style-type: none"> <li>• 存储期间</li> <li>• 运输期间</li> <li>• 垂直安装 (DIN 导轨水平放置) 运行期间</li> <li>• 水平安装 (DIN 导轨垂直放置) 运行期间</li> </ul>
	<ul style="list-style-type: none"> <li>• -40 °C 到 70 °C</li> <li>• -40 °C 到 70 °C</li> <li>• 0 °C 到 55 °C</li> <li>• 0 °C 到 45 °C</li> </ul>
在 25 °C 下运行而不会结露的最大相对湿度	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"> <li>• 最小值</li> <li>• 最大值</li> </ul>	<ul style="list-style-type: none"> <li>• 19.2 V</li> <li>• 28.8 V</li> </ul>



### 技术规范

#### 电流消耗（典型值）

- |                   |          |
|-------------------|----------|
| • 通过 24 V DC      | • 100 mA |
| • 通过 S7-1200 背板总线 | • 0 mA   |

#### 有效功耗（典型值）

- |                   |         |
|-------------------|---------|
| • 通过 24 V DC      | • 2.4 W |
| • 通过 S7-1200 背板总线 | • 0 W   |

#### 24 V DC 电源

- |             |                                      |
|-------------|--------------------------------------|
| • 最小电缆横截面积  | • 最小值： 0.14 mm <sup>2</sup> (AWG 25) |
| • 最大电缆横截面积  | • 最大值： 1.5 mm <sup>2</sup> (AWG 15)  |
| • 螺钉端子的拧紧扭矩 | • 0.45 Nm (4 lb-in)                  |

#### 尺寸和重量

- |      |          |
|------|----------|
| • 宽度 | • 30 mm  |
| • 高度 | • 100 mm |
| • 深度 | • 75 mm  |

#### 重量

- |          |         |
|----------|---------|
| • 净重     | • 133 g |
| • 含包装的重量 | • 170 g |

### ANT794-4MR GSM/GPRS 天线的技术规范

#### ANT794-4MR

订货号	6NH9860-1AA00
移动无线网络	GSM/GPRS
频率范围	<ul style="list-style-type: none"> <li>• 824 到 960 MHz (GSM 850、900)</li> <li>• 1710 到 1880 MHz (GSM 1800)</li> <li>• 1900 到 2200 MHz (GSM/UMTS)</li> </ul>
特性	全向
天线增益	0 dB
阻抗	50 欧姆
驻波比 (SWR)	< 2,0

<b>ANT794-4MR</b>	
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"> <li>• 工作温度</li> <li>• 运输/存储温度</li> <li>• 相对湿度</li> </ul>	<ul style="list-style-type: none"> <li>• -40 °C 到 +70 °C</li> <li>• -40 °C 到 +70 °C</li> <li>• 100 %</li> </ul>
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
尺寸 (D x H) (mm)	25 x 193
重量	
<ul style="list-style-type: none"> <li>• 天线 (包括电缆)</li> <li>• 配件</li> </ul>	<ul style="list-style-type: none"> <li>• 310 g</li> <li>• 54 g</li> </ul>
安装	使用随附支架

平面天线 ANT794-3M 的技术规范

订货号	6NH9870-1AA00	
移动无线网络	<b>GSM 900</b>	<b>GSM 1800/1900</b>
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	具有 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
可燃性	UL 94 V2
外部材料	ABS Polylac PA-765, 浅灰色 (RAL 7035)
尺寸 (W x L x H) (mm)	70.5 x 146.5 x 20.5
重量	130 g

### A.12.3 CM 1243-2 AS-i 主站

#### A.12.3.1 AS-i 主站 CM 1243-2 的技术数据

表格 A- 216 AS-i 主站 CM 1243-2 的技术数据

<b>技术数据</b>	
订货号	3RK7243-2AA30-0XB0
<b>接口</b>	
最大电流消耗	
自 SIMATIC 背板总线	最大 250 mA, SIMATIC 背板总线电源电压为 5 V DC
自 AS-i 电缆	最大 100 mA
引脚分配	请参见 AS-i 主站 CM 1243-2 的电气连接 (页 905) 部分
导线横截面	0.2 mm <sup>2</sup> (AWG 24) ... 3.3 mm <sup>2</sup> (AWG 12)
ASI 连接器紧固扭矩	0.56 Nm
<b>允许的环境条件</b>	

技术数据	
环境温度	
存储期间	-40 °C ... 70 °C
运输期间	-40 °C ... 70 °C
垂直安装（水平标准安装轨）时的运行阶段	
水平安装（垂直标准安装轨）时的运行阶段	0 °C ... 55 °C
	0 °C ... 45 °C
在 25 °C 下运行阶段的最大相对湿度，无冷凝	95 %
防护等级	IP20
电源、电流消耗和功率损耗	
电源类型	DC
电流消耗（典型值）	
自 S7-1200 背板总线	200 mA
功率损耗（典型值）	2.4 W（自 AS-i）
自 S7-1200 背板总线	0.5 W
尺寸和重量	
宽度	30 mm
高度	100 mm
深度	75 mm
重量	
净重	122 g
含包装重量	159 g

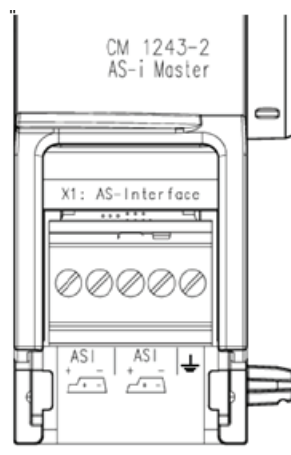
### A.12.3.2 AS-i 主站 CM 1243-2 的电气连接

#### AS-i 主站 CM 1243-2 的电源

AS-i 主站 CM 1243-2 通过 S7-1200 的通信总线供电。这意味着在 AS-i 电源发生故障后仍可向 S7-1200 发送诊断消息。通信总线的连接在 AS-i 主站 CM 1243-2 的右侧。

#### AS 接口端子

用于连接 AS-i 电缆的可拆卸端子位于 AS-i 主站 CM 1243-2 前面下盖的后面。



如果使用 AS-i 型电缆，则可以通过符号  识别正确的电缆极性。

有关如何拆卸和重新安装端子块的信息，请参见系统手册“SIMATIC S7-1200 可编程控制器”（订货号：6ES7298-8FA30-8AH0）

#### 注意

##### 端子触点的最大载流能力

连接触点的载流能力最大为 8 A。如果 AS-i 电缆上超过此值，AS-i 主站 CM 1243-2 不能直接连接到 AS-i 电缆，而必须通过分支电缆连接（AS-i 主站 CM 1243-2 上只分配了一个连接对）。

有关连接 AS-i 电缆的更多信息，请参见《SIMATIC S7-1200 的 AS-i 主站 CM 1243-2 和 AS-i 数据解耦单元 DCM 1271》手册的“模块安装、连接和调试”部分。

A.12 通信接口

接线端分配

标签	含义
ASI+	AS-i 连接 - 正极性
ASI-	AS-i 连接 - 负极性
	功能性接地

A.12.4 RS232、RS422 和 RS485

A.12.4.1 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-217 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-218 发送器和接收器

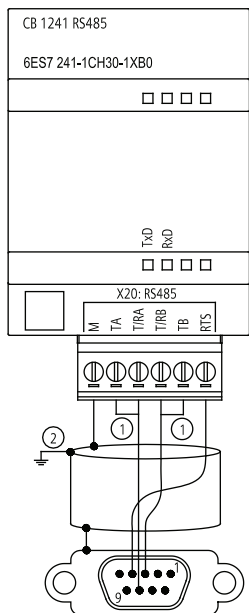
技术数据	CB 1241 RS485
类型	RS485 (2 线制半双工)
共模电压范围	-7 V 到 +12 V, 1 秒, 3 VRMS 连续
发送器差动输出电压	$R_L = 100 \Omega$ 时最小 2 V, $R_L = 54 \Omega$ 时最小 1.5 V
终端和偏置	B 上 10K 对 +5 V, RS485 针 3 A 上 10K 对 GND, RS485 针 4

技术数据	CB 1241 RS485
可选终端	短针 TB 对针 T/RB, 有效终端阻抗为 127 $\Omega$ , 连接至 RS485 针 3 短针 TA 对针 T/RA, 有效终端阻抗为 127 $\Omega$ , 连接至 RS485 针 4
接收器输入阻抗	最小 5.4K $\Omega$ , 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- 219 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

CB 1241 RS485 (6ES7 241-1CH30-1XB0)



- 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。）
- ② 使用屏蔽双绞线电缆，并将电缆屏蔽接地。

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 网络连接器”上的 S7-1200 系统手册部分。

表格 A- 220 CB 1241 RS485 (6ES7 241-1CH30-1XB0) 的连接器针脚位置

针脚	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	RS485/5V 电源	--
7	RS485/未使用	--
8	RS485/TxD-	4 - T/RA
9	RS485/未使用	--
Shell		7 - M



## 参见

偏置和端接 RS485 网络连接器 (页 608)

**A.12.4.2 CM 1241 RS232 规范**

表格 A- 221 常规规范

技术数据	CM 1241 RS232
订货号	6ES7 241-1AH30-0XB0
尺寸 (mm)	30 x 100 x 75
重量	150 g

表格 A- 222 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 $\Omega$
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
接收器输入电压	最大 +/- 30VDC
隔离	500 VAC, 1 分钟
RS 232 信号与机壳接地	
RS 232 信号与 CPU 逻辑公共端	
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

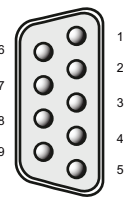
A.12 通信接口

<b>技术数据</b>	<b>CM 1241 RS232</b>
流控制	硬件, 软件
等待时间	0 到 65535 ms

表格 A- 223 电源

<b>技术数据</b>	<b>CM 1241 RS232</b>
功率损失 (损耗)	1.1 W
+5 VDC 电流	220 mA

表格 A- 224 RS232 连接器 (公)

引脚	说明	连接器 (公)	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.12.4.3 CM 1241 RS422/485 技术规格

CM 1241 RS422/485 技术规范

表格 A- 225 常规规范

<b>技术数据</b>	<b>CM 1241 RS422/485</b>
订货号	6ES7 241-1CH31-0XB0
尺寸 W x H x H (mm)	30 x 100 x 75
重量	155 g

表格 A- 226 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
终端和偏置	B 上, 10K $\Omega$ 对 +5 V, PROFIBUS 针脚 3 A 上 10K $\Omega$ 对 GND, PROFIBUS 针脚 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 (页 608) 和 S7-1200 系统手册中的 RS422 和 RS485 组态 (页 648)
接收器输入阻抗	最小 5.4K $\Omega$ , 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	500 VAC, 1 分钟
RS485 信号与机壳接地	
RS485 信号与 CPU 逻辑公共端	
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- 227 电源

技术数据	CM 1241 RS422/485
功率损失 (损耗)	1.2 W
+5 VDC 电流	240 mA

### A.13 远程服务 (TS 适配器和 TS 适配器模块)

表格 A- 228 RS485 或 RS422 连接器 (插孔式)

针脚	说明	连接器 (插孔式)	针脚	说明
1	逻辑接地或通信接地		6 PWR	+5V 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 <sup>2</sup>	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

<sup>1</sup> 针脚 2 和 9 仅适用于 RS422 的传送信号。

<sup>2</sup> 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.13 远程服务 (TS 适配器和 TS 适配器模块)

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


### A.14 SIMATIC 存储卡

订货号	容量
6ES7 954-8LF01-0AA0	24 MB
6ES7 954-8LE01-0AA0	12 MB
6ES7 954-8LB01-0AA0	2 MB

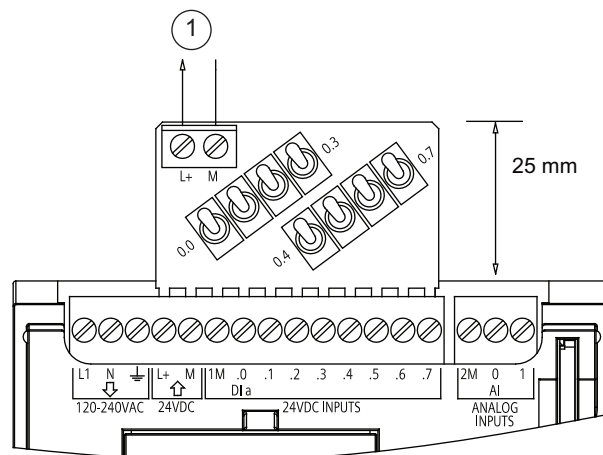
## A.15 输入仿真器

表格 A- 229 常规规范

技术数据	8 位置仿真器	14 位置仿真器
订货号	6ES7 274-1XF30-0XA0	6ES7 274-1XH30-0XA0
尺寸 W x H x D (mm)	43 x 35 x 23	67 x 35 x 23
重量	20 g	30 g
点数	8	14
配套使用的 CPU	CPU 1211C、CPU 1212C	CPU 1214C

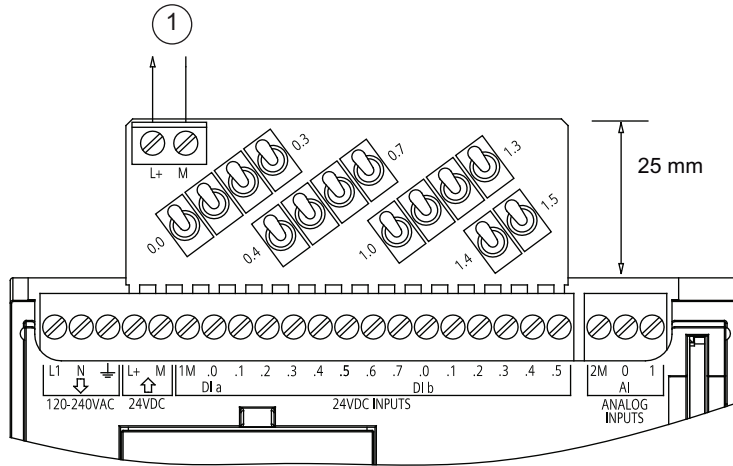
 <b>警告</b>
<p>这些输入仿真器未获准在 Class I DIV 2 或 Class I Zone 2 危险场所使用。如果在 Class I DIV 2 或 Class I Zone 2 场所使用，开关存在潜在的打火危险/爆炸危险。</p>

8 位置仿真器 (6ES7 274-1XF30-0XA0)



① 24 VDC 传感器电源输出

14 位置仿真器 (6ES7 274-1XF30-0XA0)



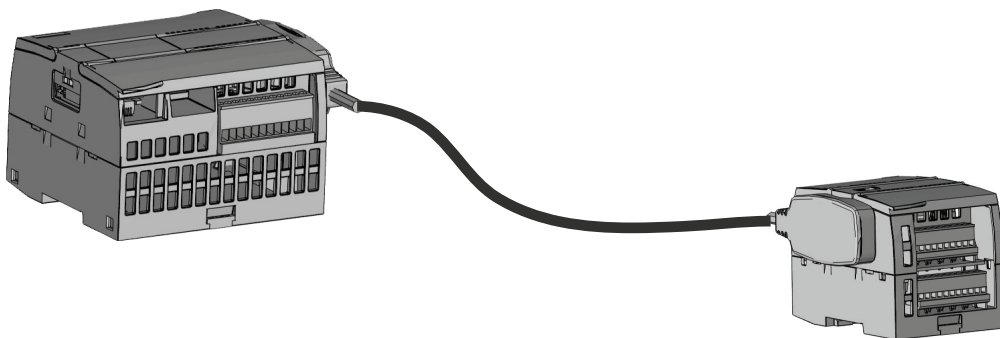
① 24 VDC 传感器电源输出

A.16 I/O 扩展电缆

技术数据

订货号	6ES7 290-6AA30-0XA0
电缆长度	2 m
重量	200 g

有关安装和拆卸 S7-1200 扩展电缆的信息，请参见 安装 (页 57)部分



## A.17 随附产品

### A.17.1 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

有关该产品以及产品文档的详细信息，请参见客户支持网站 (<http://www.siemens.com/automation/>)。

### A.17.2 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，请参见客户支持网站 (<http://www.siemens.com/automation/>)。





## 计算功率预算

CPU 有一个内部电源，用于为 CPU 本身和任何扩展模块供电以及满足其他 24 VDC 用户的功率要求。

有四种类型的扩展模块：

- 信号模块 (SM) 安装在 CPU 右侧。在不考虑功率预算的情况下，每个 CPU 可允许的最大信号模块数如下。
  - CPU 1214C 和 CPU 1215C 允许 8 个信号模块
  - CPU 1212C 允许 2 个信号模块
  - CPU 1211C 不允许任何信号模块
- 通信模块 (CM) 安装在 CPU 左侧。若不考虑功率预算，任何 CPU 都允许最多 3 个通信模块。
- 信号板 (SB)、通信板 (CB) 和电池板 (BB) 安装在 CPU 顶部。任何 CPU 最多允许使用 1 个信号板、通信板或电池板。

请使用以下信息作为指导，确定 CPU 可为您的组态提供多少电能（或电流）。

每个 CPU 都提供了 5 VDC 和 24 VDC 电源：

- 连接了扩展模块时，CPU 会为这些扩展模块提供 5 VDC 电源。如果扩展模块的 5 VDC 功率要求超出 CPU 的功率预算，则必须拆下一些扩展模块直到其功率要求在功率预算范围内。
- 每个 CPU 都有一个 24 VDC 传感器电源，该电源可以为本地输入点或扩展模块上的继电器线圈供给 24 VDC。如果 24 VDC 的功率要求超出 CPU 的功率预算，则可以增加外部 24 VDC 电源为扩展模块供应 24 VDC。必须将 24 VDC 电源手动连接到输入点或继电器线圈。



### 警告

将外部 24 VDC 电源与 DC 传感器电源并联会导致这两个电源之间有冲突，因为每个电源都试图建立自己首选的输出电压电平。

该冲突可能使其中一个电源或两个电源的寿命缩短或立即出现故障，从而导致 PLC 系统的运行不确定。运行不确定可能导致死亡、人员重伤和/或财产损失。

CPU 上的 DC 传感器电源和任何外部电源应分别给不同位置供电。允许将多个公共端连接到一个位置。

PLC 系统中的一些 24 V 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。在数据表中指定为非隔离时，CPU 的 24 VDC 电源输入、SM 继电器线圈电源输入以及非隔离模拟电源输入即是一些互连电路的实例。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**

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 和连接设备损坏或运行不确定。

这种损坏或不确定运行可能导致死亡、人员重伤和/或财产损失。

务必确保 PLC 系统中的所有非隔离 M 端子都连接到同一个参考电位。

有关 CPU 功率预算和信号模块功率要求的信息，请参见技术规范 (页 755)。

**说明**

若超出 CPU 功率预算，将导致无法连接 CPU 所允许的最大数量的模块。

**采样功率预算**

以下示例以包含一个 CPU 1214C AC/DC/继电器型、一个 SB 1223 2 x 24 VDC 输入/2 x 24 VDC 输出、一个 CM 1241、三个 SM 1223 8 DC 输入/8 路继电器输出以及一个 SM 1221 8 DC 输入的组态为例，给出了功率要求计算办法。该示例一共有 48 个输入和 36 个输出。

**说明**

该 CPU 已分配驱动内部继电器线圈所需的功率。功率预算计算中无需包括内部继电器线圈的功率要求。

在本例中的 CPU 为 SM 提供了足够的 5 VDC 电流，但没有通过传感器电源为所有输入和扩展继电器线圈提供足够的 24 VDC 电流。I/O 需要 456 mA，而 CPU 只提供 400 mA。该安装额外需要一个至少为 56 mA 的 24 VDC 电源以运行所有包括的 24 VDC 输入和输出。

表格 B-1 采样功率预算

CPU 功率预算	5 VDC	24 VDC
CPU 1214C AC/DC/继电器	1600 mA	400 mA
减		

CPU 功率预算	5 VDC	24 VDC
<b>系统要求</b>	<b>5 VDC</b>	<b>24 VDC</b>
CPU 1214C, 14 点输入	-	14 * 4 mA = 56 mA
1 个 SB 1223 2 x 24 VDC 输入/2 x 24 VDC 输出	50 mA	2 * 4 mA = 8 mA
1 个 CM 1241 RS422/485, 5 V 电源	220 mA	
3 个 SM 1223, 5 V 电源	3 * 145 mA = 435 mA	-
1 个 SM 1221, 5 V 电源	1 * 105 mA = 105 mA	-
3 个 SM 1223, 各 8 点输入	-	3 * 8 * 4 mA = 96 mA
3 个 SM 1223, 各 8 个继电器线圈	-	3 * 8 * 11 mA = 264 mA
1 个 SM 1221, 各 8 点输入	-	8 * 4 mA = 32 mA
<b>总要求</b>	<b>810 mA</b>	<b>456 mA</b>
等于		
<b>电流差额</b>	<b>5 VDC</b>	<b>24 VDC</b>
总电流差额	790 mA	(56 mA)

**功率预算计算表**

通过下表可以确定 S7-1200 CPU 可为您的组态提供多少电源（或电流）。有关用户 CPU 型号的功率预算和信号模块功率要求信息，请参见技术规范 (页 755)。

表格 B-2 功率预算计算

<b>CPU 功率预算</b>	<b>5 VDC</b>	<b>24 VDC</b>
<i>减</i>		
<b>系统要求</b>	<b>5 VDC</b>	<b>24 VDC</b>
<b>总要求</b>		
<i>等于</i>		
<b>电流差额</b>	<b>5 VDC</b>	<b>24 VDC</b>
<b>总电流差额</b>		

## 订货号

## C.1 CPU 模块

表格 C-1 S7-1200 CPU

CPU 型号		订货号
CPU 1211C	CPU 1211C DC/DC/DC	6ES7 211-1AE31-0XB0
	CPU 1211C AC/DC/继电器	6ES7 211-1BE31-0XB0
	CPU 1211C DC/DC/继电器	6ES7 211-1HE31-0XB0
CPU 1212C	CPU 1212C DC/DC/DC	6ES7 212-1AE31-0XB0
	CPU 1212C AC/DC/继电器	6ES7 212-1BE31-0XB0
	CPU 1212C DC/DC/继电器	6ES7 212-1HE31-0XB0
CPU 1214C	CPU 1214C DC/DC/DC	6ES7 214-1AG31-0XB0
	CPU 1214C AC/DC/继电器	6ES7 214-1BG31-0XB0
	CPU 1214C DC/DC/继电器	6ES7 214-1HG31-0XB0
CPU 1215C	CPU 1215C DC/DC/DC	6ES7 215-1AG31-0XB0
	CPU 1215C AC/DC/继电器	6ES7 215-1BG31-0XB0
	CPU 1215C DC/DC/继电器	6ES7 215-1HG31-0XB0

## C.2 信号模块 (SM)、信号板 (SB) 以及电池板 (BB)

表格 C-2 信号模块 (SM)

信号模块		订货号
数字量输入	SM 1221 8 x 24 VDC 输入 (漏型/源型)	6ES7 221-1BF30-0XB0
	SM 1221 16 x 24 VDC 输入 (漏型/源型)	6ES7 221-1BH30-0XB0
数字量输出	SM 1222 8 x 24 VDC 输出 (源型)	6ES7 222-1BF30-0XB0
	SM 1222 16 x 24 VDC 输出 (源型)	6ES7 222-1BH30-0XB0
	SM 1222 8 x 继电器输出	6ES7 222-1HF30-0XB0
	SM 1222 8 x 继电器输出 (切换)	6ES7 222-1XF30-0XB0

C.2 信号模块 (SM)、信号板 (SB) 以及电池板 (BB)

信号模块		订货号
	SM 1222 16 x 继电器输出	6ES7 222-1HH30-0XB0
数字量输入/输出	SB 1223 8 x 24 VDC 输入 (漏型/源型) /8 x 24 VDC 输出 (源型)	6ES7 223-1BH30-0XB0
	SB 1223 16 x 24 VDC 输入 (漏型/源型) /16 x 24 VDC 输出 (源型)	6ES7 223-1BL30-0XB0
	SM 1223 8 x 24 VDC 输入 (漏型/源型) /8 x 继电器输出	6ES7 223-1PH30-0XB0
	SM 1223 16 x 24 VDC 输入 (漏型/源型) /16 x 继电器输出	6ES7 223-1PL30-0XB0
	SM 1223 8 x 120/230 VAC 输入 (漏型/源型) /8 x 继电器输出	6ES7 223-1QH30-0XB0
模拟量输入	SM 1231 4 x 模拟量输入	6ES7 231-4HD30-0XB0
	SM 1231 8 x 模拟量输入	6ES7 231-4HF30-0XB0
	SM 1231 4 x 模拟量输入 x 16 位 (高性能型)	6ES7 231-5ND30-0XB0
模拟量输出	SM 1232 2 x 模拟量输出	6ES7 232-4HB30-0XB0
	SM 1232 4 x 模拟量输出	6ES7 232-4HD30-0XB0
模拟量输入/输出	SM 1234 4 x 模拟量输入 /2 x 模拟量输出	6ES7 234-4HE30-0XB0
RTD 和热电偶	SM 1231 TC 4 x 16 位	6ES7 231-5QD30-0XB0
	SM 1231 TC 8 x 16 位	6ES7 231-5QF30-0XB0
	SM 1231 RTD 4 x 16 位	6ES7 231-5PD30-0XB0
	SM 1231 RTD 8 x 16 位	6ES7 231-5PF30-0XB0

表格 C-3 信号板 (SB) 和电池板 (BB)

信号板和电池板		订货号
数字量输入	SB 1221 200 KHz 4 x 24 VDC 输入 (源型) ,	6ES7 221-3BD30-0XB0
	SB 1221 200 KHz 4 x 5 VDC 输入 (源型)	6ES7 221-3AD30-0XB0
数字量输出	SB 1222 200 KHz 4 x 24 VDC 输出 (漏型/源型)	6ES7 222-1BD30-0XB0
	SB 1222 200 KHz 4 x 5 VDC 输出 (漏型/源型)	6ES7 222-1AD30-0XB0
数字量输入/输出	SB 1223 2 x 24 VDC 输入 (漏型) /2 x 24 VDC 输出 (源型)	6ES7 223-0BD30-0XB0

信号板和电池板		订货号
	SB 1223 200 KHz 2 x 24 VDC 输入 (源型) /2 x 24 VDC 输出 (漏型/源型)	6ES7 223-3BD30-0XB0
	SB 1223 200 KHz 2 x 5 VDC 输入 (源型) /2 x 5 VDC 输出 (漏型/源型)	6ES7 223-3AD30-0XB0
模拟量	SB 1232 1 路模拟量输出	6ES7 232-4HA30-0XB0
	SB 1231 1 路模拟量输入	6ES7 231-4HA30-0XB0
	SB 1231 1 路模拟量输入热电偶	6ES7 231-5QA30-0XB0
	SB 1231 1 路模拟量输入 RTD	6ES7 231-5PA30-0XB0
电池	BB 1297 电池	6ES7 297-0AX30-0XA0

### C.3 通信

表格 C-4 通信模块 (CM)

通信模块 (CM)			订货号
RS232、RS422 和 RS485	CM 1241 RS232	RS232	6ES7 241-1AH30-0XB0
	CM 1241 RS422/485	RS422/485	6ES7 241-1CH31-0XB0
PROFIBUS	CM 1243-5	PROFIBUS 主站	6GK7 243-5DX30-0XE0
	CM 1242-5	PROFIBUS 从站	6GK7 242-5DX30-0XE0
AS-i 主站	CM 1243-2	AS-i 主站	3RK7 243-2AA30-0XB0

表格 C-5 通信板 (CB)

通信板 (CB)			订货号
RS485	CB 1241 RS485	RS485	6ES7 241-1CH30-1XB0

表格 C-6 通信处理器 (CP)

通信处理器 (CP)		订货号
CP 1242-7	GPRS	6GK7 242-7KX30-0XE0

C.4 其它模块

表格 C-7 TeleService

TS 适配器	订货号
TS Adapter IE Basic	6ES7 972-0EB00-0XA0
TS Module GSM	6GK7 972-0MG00-0XA0
TS Module RS232	6ES7 792-0MS00-0XA0
TS Module Modem	6ES7 972-0MM00-0XA0
TS Module ISDN	6ES7 972-0MD00-0XA0

表格 C-8 附件

附件			订货号
天线	ANT794-4MR	GSM/GPRS 天线	6NH9 860-1AA00
	ANT794-3M	平头天线	6NH9 870-1AA00

表格 C-9 连接器

连接器类型		订货号
RS485	35 度电缆输出，螺丝端子连接	6ES7 972-0BA42-0XA0
	35 度电缆输出，FastConnect 连接	6ES7 972-0BA60-0XA0

## C.4 其它模块

表格 C-10 随附产品

物品	订货号
电源模块	PM 1207 电源 6EP1 332-1SH71
以太网交换机	CSM 1277 以太网交换机 - 4 端口 6GK7 277-1AA10-0AA0



## C.5 存储卡

表格 C- 11 存储卡

SIMATIC 存储卡	订货号
SIMATIC MC 2 MB	6ES7 954-8LB01-0AA0
SIMATIC MC 12 MB	6ES7 954-8LE01-0AA0
SIMATIC MC 24 MB	6ES7 954-8LF01-0AA0

## C.6 Basic HMI 设备

表格 C- 12 HMI 设备

HMI 基本型面板	订货号
KTP400 Basic (单色, PN)	6AV6 647-0AA11-3AX0
KTP600 Basic (单色, PN)	6AV6 647-0AB11-3AX0
KTP600 Basic (彩色, PN)	6AV6 647-0AD11-3AX0
KTP1000 Basic (彩色, PN)	6AV6 647-0AF11-3AX0
TP1500 Basic (彩色, PN)	6AV6 647-0AG11-3AX0

## C.7 备件和其它硬件

表格 C- 13 扩展电缆、仿真器和连接板

物品	订货号	
I/O 扩展电缆	I/O 扩展电缆, 2 m 6ES7 290-6AA30-0XA0	
I/O 仿真器	仿真器 (1214C/1211C - 8 位置)	6ES7 274-1XF30-0XA0
	仿真器 (1214C - 14 位置)	6ES7 274-1XH30-0XA0
备用门配件	CPU 1211C/1212C	6ES7 291-1AA30-0XA0
	CPU 1214C	6ES7 291-1AB30-0XA0
	CPU 1215C	6ES7 291-1AC30-0XA0
	单个模块, 45 mm	6ES7 291-1BA30-0XA0

C.8 编程软件

物品		订货号	
	单个模块, 70 mm	6ES7 291-1BB30-0XA0	
	通信模块	6ES7 291-1CC30-0XA0	
连接板	镀锡	7 个端子, 4/pk	6ES7 292-1AG30-0XA0
		8 个端子, 4/pk	6ES7 292-1AH30-0XA0
		11 个端子, 4/pk	6ES7 292-1AL30-0XA0
		12 个端子, 4/pk	6ES7 292-1AM30-0XA0
		14 个端子, 4/pk	6ES7 292-1AP30-0XA0
		20 个端子, 4/pk	6ES7 292-1AV30-0XA0
	镀金	3 个端子, 4/pk (用于模拟 CPU)	6ES7 292-1BC30-0XA0
		6 个端子, 4/pk (用于数字板)	6ES7 292-1BF30-0XA0
		6 个端子, 4/pk (用于模拟 CPU)	6ES7 292-1BF30-0XB0
		7 个端子, 4/pk (用于模拟信号模块)	6ES7 292-1BG30-0XA0
11 个端子, 4/pk (用于模拟信号模块)		6ES7 292-1BL30-0XA0	

C.8 编程软件

表格 C- 14 编程软件

SIMATIC 软件		订货号
编程软件	STEP 7 Basic V11	6ES7 822-0AA01-0YA0
	STEP 7 Professional V11	6ES7 822-1AA01-0YA5
可视化软件	WinCC Basic V11	6AV2100-0AA01-0AA0
	WinCC Comfort V11	6AV2101-0AA01-0AA5
	WinCC Advanced V11	6AV2102-0AA01-0AA5
	WinCC Professional 512 PowerTags V11	6AV2103-0DA01-0AA5
	WinCC Professional 4096 PowerTags V11	6AV2103-0HA01-0AA5
	WinCC Professional max. PowerTags V11	6AV2103-0XA01-0AA5

## C.9 文档

表格 C- 15 S7-1200 文档

印刷文档	语言	订货号
S7-1200 可编程控制器系统手册	德语	6ES7 298-8FA30-8AH0
	英语	6ES7 298-8FA30-8BH0
	法语	6ES7 298-8FA30-8CH0
	西班牙语	6ES7 298-8FA30-8DH0
	意大利语	6ES7 298-8FA30-8EH0
	中文	6ES7 298-8FA30-8KH0
S7-1200 简明手册	德语	6ES7 298-8FA30-8AQ0
	英语	6ES7 298-8FA30-8BQ0
	法语	6ES7 298-8FA30-8CQ0
	西班牙语	6ES7 298-8FA30-8DQ0
	意大利语	6ES7 298-8FA30-8EQ0
	中文	6ES7 298-8FA30-8KQ0



# 索引

## A

ABS (绝对值), 207

### AC

电感负载, 66

绝缘准则, 64

接地, 65

接线准则, 63, 65

ACOS (反余弦), 210

ADD (加法), 204

AND, 248

### AS-i

AS-i 地址, 520

AS-i 地址属性, 520

RDREC, 291

WRREC, 291

不使用 STEP 7 组态从站,  
从站地址的系统分配, 521

分布式 I/O 指令, 290

传输数字值, 522

传输模拟值, 522

网络连接, 518

系统分配, 521

使用 STEP 7 组态从站,

添加 AS-i 从站, 518

添加 AS-i 主站 CM1243-2 模块, 518

AS-i 主站 CM 1243-2, 517

模块特征, 517

AS-i 地址, 520

组态, 520

ASIN (反正弦), 210

ATEX 认证, 756

ATH (ASCII 到十六进制), 279

ATTACH, 308

AWP 命令, 563

引用枚举类型, 572

写入变量, 565

写入特殊变量, 569

生成片段, 573

导入片段, 575

使用别名, 570

定义枚举类型, 571

组合定义, 575

读取特殊变量, 567

AWP\_Enum\_Def, 571

AWP\_Enum\_Ref, 572

AWP\_Import\_Fragment, 575

AWP\_In\_Variable, 565, 569

AWP\_Out\_Variable, 567

AWP\_Start\_Fragment, 573

## B

BUFFER 参数, SEND\_PTP, 624

## C

CALCULATE, 34, 203

标定模拟值, 35

CAN\_DINT (取消延时中断), 315

CB 1241 RS485 规范, 907

CE 认证, 755

CEIL (上取整), 225

Char (字符数据类型), 100

CM 1241

- RS232 规范, 910
- RS422/RS485 技术规范, 910
- Communication
  - 参数, 553
  - 统计数据, 553
- Communication 标准 Web 页面, 553
- CONCAT (连接), 282
- CONV (转换), 220
- Cookie 限制, 标准 Web 页面, 603
- Cookie, siemens\_automation\_language, 595
- COS (余弦), 210
- CPU
  - 1211C 的接线图, 773
  - 1211C 规范, 762
  - 1212C 的规范, 774
  - 1212C 的接线图, 785
  - 1214C 的规范, 786
  - 1215C 的接线图, 812
  - 1215C 规范, 799
  - AS-i, 519
  - AS-i 地址, 520
  - AS-i 端口, 519
  - HMI 设备, 29
  - HSC 组态, 374
  - IP 地址, 139
  - LED 指示灯, 729
  - MAC 地址, 139, 143
  - PROFIBUS, 515
  - PROFIBUS 地址, 515
  - PROFIBUS 端口, 515
  - PROFINET, 139
  - PROFINET IO, 503
  - RD\_LOC\_T (读取本地时间), 260
  - RD\_SYS\_T (读取系统时间), 260
  - RUN 和 STOP 模式, 735
  - RUN/STOP 按钮, 39
  - WR\_SYS\_T (写入系统时间), 260
  - 下载, 169
  - 下载到设备, 143
  - 工作模式, 71
  - 专有技术保护, 167
  - 从在线 CPU 复制块, 170
  - 比较并同步块, 737
  - 比较表, 22
  - 以太网端口, 139
  - 功率要求, 917
  - 功率预算, 44
  - 发热区, 44, 47
  - 处理 OB, 150
  - 未指定的 CPU, 123
  - 电感负载, 66
  - 丢失密码, 120
  - 丢失密码后恢复, 120
  - 传送卡, 111, 114
  - 创建传送卡, 114
  - 创建程序卡, 116
  - 在 STOP 模式下启用输出, 743
  - 在线, 734, 739
  - 存储卡, 111
  - 安全等级, 166
  - 安装, 48, 49
  - 扩展电缆, 57
  - 灯负载, 66
  - 网络连接, 128
  - 设备配置, 121
  - 访问保护, 166
  - 阶跃响应时间, 769, 782, 794, 806
  - 启动过程, 73
  - 启动参数, 113
  - 时间同步属性, 145
  - 运行时间计时器, 261
  - 空传送卡, 120

- 组态与 HMI 的通信, 497
- 组态多个, 499
- 组态参数, 125
- 组态脉冲通道, 337
- 组态模块, 127
- 转到在线, 731
- 信号板 (SB), 27
- 复位为出厂设置, 734
- 将 IP 地址分配给在线 CPU, 138
- 显示 MAC 地址和 IP 地址, 143
- 绝缘准则, 64
- 脉冲输出, 335
- 重设 DB 的起始值, 740
- 捕获 DB 的值, 740
- 监视, 739
- 监视表格, 741
- 通信负载, 83
- 通信连接数, 458
- 通信板 (CB), 27
- 通信类型, 457
- 密码保护, 166
- 接地, 65
- 接线准则, 63, 65
- 添加新设备, 122
- 添加模块, 124
- 强制, 744, 745
- 循环时间组态, 83
- 循环时间监视, 81
- 插入存储卡, 112
- 程序卡, 111, 116
- 程序执行, 69
- 概述, 21
- 模拟量输入的电压表示法, 845, 880
- 端子板连接器, 56
- 操作员面板, 39
- 操作面板 (在线 CPU), 735
- CPU 的 Identification 标准 Web 页面, 549
- CPU 通信, 460
- CPU 属性, 用户定义 Web 页面 (多语言), 597
- CPU 属性, 用户定义的 Web 页面, 577
- CTD (减计数), 193
- C-Tick 认证, 757
- CTRL\_PWM, 333
- CTS, 632
- CTU (加计数), 193
- CTUD (加计数和减计数), 193
- cULus 认证, 756
  
- D**
- Data Logs 标准 Web 页面, 556
- Date
  - Date 数据类型, 98
  - DTL (长格式日期和时间数据类型), 99
- DB (数据块), 69, 153
  - 有效 DB 号, 69
  - 重设起始值, 740
  - 捕获值, 740
- DC
  - 电感负载, 66
  - 绝缘准则, 64
  - 接地, 65
  - 接线准则, 63, 65
- DEC (递减), 207
- DECO (解码), 249
- DELETE (删除子串), 285
- DETACH, 308
- DeviceStates, 321
- Diagnostic 标准 Web 页面, 549
- DIN 导轨, 48
- DIS\_AIRT (禁用报警中断), 317
- DIV (除法), 204

DPNRM\_DG, 305

DPRD\_DAT, 302

DPWR\_DAT, 302

DTL 数据类型

系统时钟指令, 260

## E

EN 和 ENO (能流), 164

EN\_AIRT (启用报警中断), 317

ENCO (编码), 249

EXP (自然指数), 210

EXPT (一般指数), 210

## F

FB (功能块)

有效 FB 号, 69

概述, 69

FBD (功能块图), 157

FC (功能), 69, 152

有效 FC 号, 69

FieldRead, 215

FieldWrite, 215

FILL\_BLK, 217

FIND (查找子串), 289

FLOOR, 225

FM 认证, 756

FRAC (分数), 210

## G

GET, 526

组态连接, 130

GET\_DIAG, 324

GetError, 244

GetErrorID, 245

## H

HMI 设备

网络连接, 128

组态 PROFINET 通信, 497

概述, 29

HSC (高速计数器)

运行, 365

组态, 374

操作, 363

HTA (十六进制到 ASCII), 280

HTML 列表, 用户定义的 Web 页面示例, 587

HTML 页面, 用户定义的, 561

开发, 562

访问 S7-1200 数据, 563

页面位置, 577

刷新, 563

语言位置, 597

HTTP 连接, Web 服务器, 601

## I

I 存储器

HSC (高速计数器), 365

外围设备输入地址 (强制表格), 744

监视, 739

监视 LAD, 739

监视表格, 739

强制, 744

强制表格, 744

强制操作, 745

I/O

电感负载, 66

寻址, 93

访问错误, PROFINET, 507

阶跃响应时间 (CPU), 769, 782, 794, 806

阶跃响应时间 (SB), 879



信号模块 (SM) 的阶跃响应时间, 844  
 监视 LAD 中的状态, 739  
 通过监视表格监视, 741  
 强制, 744  
 强制操作, 745  
 数字量状态指示灯, 730  
 模拟量状态指示灯, 730  
 模拟量输入的电压表示法, 845, 880  
 模拟量输入的电流表示法, 846, 881  
 模拟量输出的电压表示法, 846, 881  
 模拟量输出的电流表示法, 847, 882  
 Identification 标准 Web 页面, 549  
 IN\_RANGE (在范围之内), 201  
 INC (递增), 207  
 INSERT (插入子串), 286  
 Intro 标准 Web 页面, 547  
 IP 地址, 140  
     MAC 地址, 139  
     分配, 135, 142  
     对在线 CPU 进行组态, 734  
     在线分配, 138  
     设备组态, 125  
     组态, 139  
 IP 路由器, 139  
 ISO on TCP  
     特殊模式, 464  
 ISO on TCP 协议, 463  
 ISO-on-TCP  
     连接 ID, 461  
     连接组态, 129  
     参数, 132  
  
**J**  
 JavaScript 限制, 标准 Web 页面, 602  
 JMP, 237

JMP\_LIST, 238  
 JMPN, 237

## L

LAD (梯形图)  
     状态, 739, 744  
     监视, 739  
     程序编辑器, 739  
     概述, 156  
 LED 指示灯  
     CPU 状态, 729  
     LED 指令, 319  
     通信接口, 607, 729  
 LED (获取 LED 状态), 319  
 LEFT (左侧子串), 284  
 LEN (长度), 282  
 LENGTH 参数, SEND\_PTP, 624  
 Limit, 209  
 LN (自然对数), 210

## M

MAC 地址, 139, 143  
 MAX (最大值), 208  
 MB\_CLIENT, 673  
 MB\_COMM\_LOAD, 692  
 MB\_MASTER, 695  
 MB\_SERVER, 680  
 MB\_SLAVE, 701  
 MC\_ChangeDynamic, 428  
 MC\_CommandTable, 425  
 MC\_Halt, 414  
 MC\_Home, 411  
 MC\_MoveAbsolute, 416  
 MC\_MoveJog, 423

MC\_MoveRelative, 418  
MC\_MoveVelocity, 420  
MC\_Power, 407  
MC\_Reset, 410  
MID (中间子串), 284  
MIN (最小值), 208  
MOD (求模), 205  
MODBUS  
    MB\_CLIENT, 673  
    MB\_COMM\_LOAD, 692  
    MB\_MASTER, 695  
    MB\_SERVER, 680  
    MB\_SLAVE, 701  
    Modbus 从站示例, 709  
    版本, 37, 691  
Module information 标准 Web 页面, 550  
ModuleStates, 322  
MOVE, 212  
MRES  
    操作员面板, 39  
MUL (乘法), 204  
MUX (多路复用), 251  
My Documentation Manager, 4

## N

N\_TRIG, 184  
NEG (取反), 206  
NORM\_X (标准化), 226  
NOT OK 指令, 202

## O

OK 指令, 202  
OPC, 组态, 717  
OR, 248

OUT\_RANGE (在范围之外), 201

## P

P\_TRIG, 184  
PID  
    PID\_3STEP, 384  
    PID\_3Step 算法, 375, 384  
    PID\_Compact, 378  
    PID\_Compact 算法, 375, 378  
    概述, 375

## PLC

CPU 1211C, 762  
CPU 1212C, 774  
CPU 1214C, 786  
CPU 1215C, 799  
CPU 概述, 21  
HSC 组态, 374  
RD\_LOC\_T (读取本地时间), 260  
RD\_SYS\_T (读取系统时间), 260  
WR\_SYS\_T (写入系统时间), 260  
下载, 169  
工作模式, 71  
专有技术保护, 167  
从在线 CPU 复制块, 170  
比较并同步, 737  
功率预算, 44  
存储卡, 111  
安装, 48, 49  
扩展电缆, 57  
设备配置, 121  
启动过程, 73  
时间同步属性, 145  
系统设计, 147  
运行时间计时器, 261  
使用块, 148

- 变量, 88
- 将 IP 地址分配给在线 CPU, 138
- 监视, 739
- 监视表格, 741
- 通信负载, 83
- 添加模块, 124
- 强制, 744
- 强制操作, 745
- 循环时间, 83
- 循环时间, 83
- 端子板连接器, 56
- Pointer
  - Any 数据类型, 105
  - Pointer 数据类型, 104
  - Variant 数据类型, 107
  - 指针概述, 103
- PORT\_CFG (端口组态), 613
- PROFIBUS
  - CM 1242-5 (DP 从站) 模块, 508
  - CM 1243-5 (DP 主站) 模块, 508
  - DPNRM\_DG, 305
  - DPRD\_DAT, 302
  - DPWR\_DAT, 302
  - GET, 526
  - PROFIBUS 地址, 515
  - PROFIBUS 地址属性, 516
  - PUT, 526
  - RALRM, 294
  - RDREC, 291
  - S7 连接, 530
  - WRREC, 291
  - 从站, 508
  - 分布式 I/O 指令, 290
  - 主站, 508
  - 网络连接, 128, 514
  - 添加 CM 1243-5 (DP 主站) 模块, 514
  - 添加 DP 从站, 514
- PROFIBUS 地址, 515, 516
  - 组态, 515
- PROFINET
  - CPU 与 CPU 通信, 499
  - DPRD\_DAT, 302
  - DPWR\_DAT, 302
  - GET, 526
  - IP 地址, 139
  - IP 地址分配, 146
  - MAC 地址, 139
  - PLC 与 PLC 通信, 499
  - PUT, 526
  - RALRM, 294
  - RDREC, 291
  - S7 连接, 530
  - T\_CONFIG, 486
  - TCON, 473
  - TDISCON, 473
  - TRCV, 473
  - TRCV\_C, 466
  - TSEND, 473
  - TSEND\_C, 466
  - TURCV, 481
  - TUSEND, 481
  - WRREC, 291
  - 分布式 I/O 指令, 290
  - 以太网地址属性, 140
  - 网络连接, 128, 498, 500, 504
  - 设备命名和寻址, 146
  - 时间同步, 125
  - 时间同步属性, 145
  - 系统启动时间, 146
  - 诊断, 507
  - 连接 ID, 461
  - 组态 CPU 与 HMI 设备之间的通信, 497

- 组态 IP 地址, 125
  - 测试网络, 142
  - 特殊模式, 464
  - 通信连接数, 458
  - 通信类型, 457
  - 概述, 463
  - PROFINET IO
    - 分配 CPU, 504
    - 分配设备名称, 505
    - 在线分配设备名称, 732
    - 在线设备名称, 732
    - 设备, 503
    - 设备名称, 505
    - 添加设备, 503
  - PROFINET RT, 463
  - PTO (脉冲串输出)
    - CTRL\_PWM, 333
    - 无法进行强制, 745
    - 运行, 335
    - 组态脉冲通道, 337
  - PtP 指令返回值, 610
  - PtP 通信, 609
    - 用于示例程序的终端仿真器, 652
    - 示例程序, 645
    - 示例程序, STEP 7 编程, 651
    - 示例程序, 运行, 652
    - 示例程序组态, 646
    - 组态参数, 634
    - 组态端口, 630
    - 编程, 643
  - PtP 错误类别, 612
  - PUT, 526
    - 组态连接, 130
  - PWM (脉冲宽度调制)
    - CTRL\_PWM, 333
    - 无法进行强制, 745
  - 运行, 335
  - 组态脉冲通道, 337
- ## Q
- Q 存储器
    - 组态脉冲通道, 337
    - 脉冲输出, 335
  - QRY\_CINT (查询循环中断), 313
- ## R
- RALRM, 294, 297
  - RCV\_CFG (接收组态), 616
  - RCV\_PTP (接收点对点), 625
  - RCV\_RST (接收方复位), 627
  - RD\_LOC\_T (读取本地时间), 260
  - RD\_SYS\_T (读取系统时间), 260
  - RDREC, 291, 297
  - RE\_TRIGR, 242
  - READ\_DBL, 358
  - REPLACE (替换子串), 287
  - RIGHT (右侧子串), 284
  - ROL 和 ROR (循环左移和循环右移), 255
  - ROUND, 224
  - RS232 和 RS485 通信模块, 607
  - RT (重置定时器), 185
  - RTS, 632
  - RTS 切换, 632
  - RTS 关断延时, 634
  - RTS 始终激活, 633
  - RTS 接通延时, 634
  - RUN 模式, 71, 74, 735
    - 工具栏按钮, 39
    - 强制操作, 745
    - 操作员面板, 39

RUN/STOP 按钮, 39

## S

S\_CONV (值到字符串的转换), 266

S\_MOV (字符串移动), 266

S7 通信

组态连接, 130

S7-1200

AS-i, 520

AS-i 地址, 520

AS-i 端口, 519

CPU 安装, 49

CPU 概述, 21

HMI 设备, 29

HSC 组态, 374

IP 地址, 139

MAC 地址, 139

PROFIBUS, 515

PROFIBUS 地址, 515

PROFIBUS 端口, 515

PROFINET, 139

RUN/STOP 按钮, 39

TS 适配器, 24

工作模式, 71

专有技术保护, 167

比较代码块, 737

气流, 44

以太网端口, 139

功率预算, 44

发热区, 44, 47

电感负载, 66

丢失密码, 120

丢失密码时使用的空传送卡, 120

传送卡, 114

各种 CPU 型号的比较表, 22

安装, 48

安装 CB, 51

安装 CM, 55

安装 SB, 51

安装 SM, 53

安装尺寸, 47

扩展电缆, 57

灯负载, 66

网络连接, 128

设备配置, 121

访问保护, 166

冷却, 44

启动过程, 73

启动参数, 113

运行, 741

空隙, 44

组态 CPU 参数, 125

组态模块, 127

信号板 (SB), 27

信号模块 (SM), 27

绝缘准则, 64

脉冲输出, 335

重设 DB 的起始值, 740

捕获 DB 的值, 740

监视, 739

通信处理器 (CP), 27

通信负载, 83

通信板 (CB), 27

通信模块 (CM), 27

密码保护, 166

接地, 65

接线准则, 63, 65

添加新设备, 122

添加模块, 124

强制, 744

强制操作, 745

- 循环时间, 83
- 程序卡, 116
- 模块, 24
- 端子板连接器, 56
- 操作员面板, 39
- SB 1222
  - SB 1222 DQ 4 x 24 VDC, 200 kHz 接线图, 865
- SB 1223
  - SB 1223 DI 2/DQ 2 接线图, 873
  - SB 1223 DI 2/DQ, 200 kHz 接线图, 869
- SB 1231 AI 1 x 16 位热电偶
  - 滤波器选型表, 886
- SB 1232
  - SB 1232 AQ 1 x 12 位的接线图, 878
- SCALE\_X (标定), 226
- SCL (结构化控制语言)
  - ABS (绝对值), 207
  - ACOS (反余弦), 210
  - AND, 248
  - ASIN (反正弦), 210
  - ATAN (反正切), 210
  - ATH (ASCII 到十六进制), 279
  - ATTACH 和 DETACH, 308
  - CAN\_DINT (取消延时中断), 315
  - CASE, 231
  - CEIL (上取整), 225
  - CONCAT (连接), 282
  - CONTINUE, 235
  - CONV (转换), 220
  - COS (余弦), 210
  - CTD (减计数), 193
  - CTU (加计数), 193
  - CTUD (加计数和减计数), 193
  - DataLogClose (关闭数据日志), 344
  - DataLogCreate (创建数据日志), 339
  - DataLogNewFile (根据现有数据日志创建数据日志), 347
  - DataLogOpen (打开数据日志), 342
  - DataLogWrite (写入数据日志), 346
  - DEC (递减), 207
  - DECO (解码), 249
  - DELETE (删除子串), 285
  - DeviceStates, 321
  - DIS\_AIRT (禁用报警中断), 317
  - EN 和 ENO (能流), 164
  - EN\_AIRT (启用报警中断), 317
  - ENCO (编码), 249
  - EXIT, 236
  - EXP (自然指数), 210
  - EXPT (一般指数), 210
  - FILL\_BLK, 217
  - FIND (查找子串), 289
  - FLOOR, 225
  - FOR, 232
  - FRAC (分数), 210
  - GET\_DIAG, 324
  - GOTO, 236
  - HTA (十六进制到 ASCII), 280
  - IF-THEN, 230
  - IN\_RANGE (在范围之内), 201
  - INC (递增), 207
  - INSERT (插入子串), 286
  - INV (取反), 249
  - JMP\_LIST, 238
  - LED 状态, 319
  - LEFT (左侧子串), 284
  - LEN (长度), 282
  - LIMIT, 209
  - LN (自然对数), 210
  - MAX (最大值), 208
  - MC\_ChangeDynamic, 428

- MC\_CommandTable, 425
- MC\_Halt, 414
- MC\_Home, 411
- MC\_MoveAbsolute, 416
- MC\_MoveJog, 423
- MC\_MoveRelative, 418
- MC\_MoveVelocity, 420
- MC\_Power, 407
- MC\_Reset, 410
- MID (中间子串), 284
- MIN (最小值), 208
- MOD (求模), 205
- ModuleStates, 322
- MUX (多路复用), 251
- N\_TRIG, 184
- NEG (取反), 206
- NORM\_X (标准化), 226
- OK, 202
- OR, 248
- OUT\_RANGE (在范围之外), 201
- P\_TRIG, 184
- PID 概述, 375
- PID\_3STEP, 384
- PID\_3Step 算法, 375, 384
- PID\_Compact, 378
- PID\_Compact 算法, 375, 378
- QRY\_CINT (查询循环中断), 313
- RD\_LOC\_T (读取本地时间), 260
- RD\_SYS\_T (读取系统时间), 260
- REPEAT, 234
- REPLACE (替换子串), 287
- RETURN, 237
- RIGHT (右侧子串), 284
- ROL 和 ROR (循环左移和循环右移), 255
- S\_CONV (值到字符串的转换), 266
- S\_MOV (移动字符串), 266
- SCALE\_X (标定), 226
- SEL (选择), 251
- SET\_CINT (设置循环中断), 311
- SET\_TIMEZONE (设置时区), 263
- SHL 和 SHR (左移和右移), 254
- SIN (正弦), 210
- SQR (平方), 210
- SQRT (平方根), 210
- SRT\_DINT (启动延时中断), 315
- STRG\_VAL (字符串到值), 266
- SWITCH, 239
- T\_ADD (加上时间), 258
- T\_COMBINE (组合时间), 259
- T\_CONV (转换时间), 257
- T\_DIFF (时差), 258
- T\_SUB (减去时间), 258
- TAN (正切), 210
- UFILL\_BLK (不中断填充), 217
- VAL\_STRG (值到字符串), 266
- Var 段, 157
- WHILE, 233
- WR\_SYS\_T (写入系统时间), 260
- XOR, 248
- 比较, 200
- 计数器, 193
- 交换, 218
- 寻址, 158
- 位逻辑, 177
- 条件, 158
- 运行时间计时器, 261
- 运算符, 158
- 运算符的优先级, 158
- 取整, 224
- 定时器, 185
- 定时器运行, 189
- 表达式, 158

- 转换指令, 221
- 浮点型算术运算, 210
- 调用 FB 或 FC, 158
- 调用块, 149
- 控制语句, 158, 229, 230, 231, 232, 233, 234, 235, 236, 237
- 移动, 212
- 程序控制, 229
- 程序编辑器, 157
- 数学, 204
- 数学运算 (浮点型), 210
- 概述, 157
- 置位和复位, 180
- 截取, 224
- SEL (选择), 251
- SEND\_CFG (发送组态), 615
- SEND\_PTP (发送点对点数据), 622
  - LENGH 和 BUFFER 参数, 624
- SET\_CINT (设置循环中断), 311
- SET\_TIMEZONE (设置时区), 263
- SGN\_GET (获取 RS232 信号), 628
- SGN\_SET (设置 RS232 信号), 629
- SHL 和 SHR (左移和右移), 254
- Siemens 安全证书, Web 页面, 547, 603
- siemens\_automation\_language cookie, 595
- SIN (正弦), 210
- SM 1231 RTD
  - 选型表, 859, 893
- SM 和 SB
  - 比较表, 24
  - 设备配置, 121
- SMS, 715
- SQR (平方), 210
- SQRT (平方根), 210
- SRT\_DINT (启动延时中断), 315
- Start 标准 Web 页面, 548
- STARTUP 模式
  - 强制操作, 745
- STEP 7
  - AS-i, 520
  - AS-i 端口, 519
  - FB 的初始值, 152
  - HSC 组态, 374
  - PROFIBUS, 515
  - PROFIBUS 端口, 515
  - PROFINET, 139
  - RD\_LOC\_T (读取本地时间), 260
  - RD\_SYS\_T (读取系统时间), 260
  - RUN/STOP 按钮, 39
  - WR\_SYS\_T (写入系统时间), 260
  - 下载, 169
  - 工作模式, 71
  - 门户视图, 32
  - 从在线 CPU 复制块, 170
  - 比较并同步, 737
  - 计数器, 193
  - 代码块类型, 69
  - 以太网端口, 139
  - 功能 (FC), 152
  - 功能块 (FB), 69, 152
  - 可扩展输入或输出, 37
  - 优先等级 (OB), 75
  - 传送卡, 111
  - 向 LAD 或 FBD 指令添加输入或输出, 36
  - 在用户程序内调用代码块, 149
  - 在编辑器之间拖放, 38
  - 存储卡, 111
  - 收藏夹, 34
  - 有效 FC、FB 和 DB 号, 69
  - 网络连接, 128
  - 设备配置, 121
  - 启动过程, 73



- 块调用, 69
- 时间同步属性 (PROFINET), 145
- 更改设置, 38
- 运行, 741
- 运行时间计时器, 261
- 拔出的模块, 42
- 线性和结构化程序, 148
- 组态 CPU, 125
- 组态模块, 127
- 将 IP 地址分配给在线 CPU, 138
- 背景数据块 (DB), 152
- 重设 DB 的起始值, 740
- 项目视图, 32
- 捕获 DB 的值, 740
- 监视, 739
- 通信负载, 83
- 密码保护, 167
- 添加 PROFINET IO 设备, 503
- 添加新设备, 122
- 添加模块, 124
- 强制, 744
- 强制操作, 745
- 循环时间, 83
- 循环时间, 83
- 插入指令, 33
- 程序卡, 111
- 数据块 (DB), 69
- 操作员面板, 39
- STEP 7 网页, 4
- STEP 7 编程
  - PtP 示例程序, 651
  - 用户定义的 Web 页面, 579
- STOP 模式, 71, 735
  - 工具栏按钮, 39
  - 在 STOP 模式下启用输出, 743
  - 强制操作, 745
- 操作员面板, 39
- STP (停止 PLC 扫描循环), 243
- STRG\_VAL (字符串到值), 266
- String
  - S\_MOVE (字符串移动), 266
  - String 数据类型, 100
  - 字符串数据概述, 265
  - 字符串操作概述, 281
- SUB (减法), 204
- SWAP, 218
- SWITCH, 239
  
- T
- T\_ADD (加上时间), 258
- T\_COMBINE (组合时间), 259
- T\_CONFIG, 486
- T\_CONV (转换时间), 257
- T\_DIFF (时差), 258
- T\_SUB (减去时间), 258
- TAN (正切), 210
- TCON, 473
  - 连接 ID, 461
  - 连接参数, 132
  - 组态, 129
- TCON\_Param, 132
- TCP
  - 协议, 463
  - 连接 ID, 461
  - 连接组态, 129
  - 参数, 132
  - 特殊模式, 464
- TCP/IP 通信, 463
- TDISCON, 473
- TIA Portal
  - 门户视图, 32

项目视图, 32

Time

- DTL (长格式日期和时间数据类型), 99
- Time 数据类型, 98
- TOD (日时钟数据类型), 99

TM\_MAIL, 721

TRCV, 473

- 连接 ID, 461
- 特殊模式, 464

TRCV\_C, 466, 502

- 连接 ID, 461
- 连接参数, 132
- 组态, 129
- 特殊模式, 464

TRCV\_C 指令组态, 503

TRUNC (截取), 224

TS 适配器, 24

- SIM 卡, 60
- 在 DIN 导轨上安装, 61
- 在墙上安装, 62
- 插入一个 TS 模块, 59

TSAP, 463

- 限制, 495

TSAP 和端口号限制, 495

TSAP (传输服务访问点), 131, 465, 500, 531

TSEND, 473

- 连接 ID, 461

TSEND\_C, 501

- 连接 ID, 461
- 连接参数, 132
- 组态, 129

TSEND\_C, 501

TSEND\_C 指令组态, 502

TURCV, 481

- 连接参数, 132
- 组态, 129

TUSEND, 481

- 参数, 132

- 组态, 129

## U

UDP

- 连接组态, 129

- 参数, 132

UDP 协议, 463

UFILL\_BLK (不中断填充), 217

USS 协议库

- USS\_DRV, 657

- USS\_PORT, 660

- USS\_RPM, 661

- USS\_WPM, 663

- 状态代码, 664

- 使用要求, 654

- 常规驱动器设置信息, 667

- 概述, 653

## V

VAL\_STRG (值到字符串), 266

Variable Status 标准 Web 页面, 554

## W

Web 服务器, 541

- HTTP 最大连接数, 601

- 引号约定, 575

- 启用, 542

- 更新速率, 542

- 限制, 601

- 标准 Web 页面, 543

Web 服务器, 用户定义的 Web 页面, 561

Web 服务器最大连接数, 601

WR\_SYS\_T (写入系统时间), 260

WRIT\_DBL, 358

WRREC, 291, 297

WWW (启用用户定义 Web 页面), 579

## X

XON/XOFF, 633

XOR, 248

## 一划

一致性检查, 174

## 三划

上升沿, 183

上电后启动, 71

启动过程, 73

上传

从在线 CPU 复制块, 170

用户程序, 170

下降沿, 183

下载

Siemens 的 PC 安全证书, 547, 603

用户定义 Web 页面 DB, 580

用户程序, 169

固件更新, 118

显示 MAC 地址和 IP 地址, 143

项目, 169

子网掩码, 140

工艺对象

HSC (高速计数器), 365

工作存储器, 22

CPU 1211C, 762

CPU 1212C, 774

CPU 1214C, 786

CPU 1215C, 799

工作模式, 39

CPU 的工作模式, 71

更改 STOP/RUN, 735

已传送消息的组态, 634

门户视图, 32

## 四划

不中断移动 (UMOVE\_BLK), 212

不重启, 71

## 三划

与 CPU 或存储卡捆绑, 168

## 四划

专有技术保护

密码保护, 167

中断, 635, 636

ATTACH 和 DETACH, 308

CAN\_DINT (取消延时中断), 315

SRT\_DINT (启动延时中断), 315

中断等待时间, 76

概述, 75

从 RUN 切换到 STOP, 88

从在线 CPU 复制块, 170

从站轮询架构, 644

分配枚举类型, 用户定义的 Web 页面, 572

切换语言, 用户定义 Web 页面, 594

开始条件, 636

开放式用户通信指令返回值, 494

引号约定, Web 服务器, 575

引用枚举类型, 用户定义的 Web 页面, 572

手册, 4

手动片段 DB 控制, 597

- 支持, 3
- 文件夹, 用户定义 Web 页面的语言, 594
- 文档, 4
- 日历, 257
- 日期
  - SET\_TIMEZONE (设置时区), 263
  - T\_ADD (加上时间), 258
  - T\_COMBINE (组合时间), 259
  - T\_CONV (转换时间), 257
  - T\_DIFF (时差), 258
  - T\_SUB (减去时间), 258
- 比较, 200
- 比较代码块, 737
- 比较并同步在线/离线 CPU, 737
- 比较表
  - CPU 型号, 22
  - HMI 设备, 29
  - 模块, 24
- 气流, 44
- 片段 DB (用户定义的 Web 页面)
  - 生成, 579
- 片段 (用户定义的 Web 页面)
  - 使用 AWP 命令导入, 575
  - 通过 AWP 命令创建, 573
- 计数器
  - HSC 运行, 365
  - HSC 组态, 374
  - HSC (高速计数器), 363
    - 大小, 24, 765, 776, 789, 801
    - 数量, 24, 764, 776, 789, 801
- 计算机要求, 31
- 订货号
  - CPU, 921
  - CSM 1277 以太网交换机, 924
  - HMI 基本型面板, 925
  - PM 1207 电源, 924
  - STEP 7, 926
  - WinCC, 926
- 文档, 927
- 可视化软件, 926
- 仿真器, 925
- 存储卡, 925
- 扩展电缆, 925
- 连接板, 925
- 连接器和终端连接, 924
- 信号板 (SB), 922
- 信号模块 (SM), 921
- 通信接口 (CM、CB 和 CP), 923, 924
- 编程软件, 926
- 认证
  - ATEX 认证, 756
  - CE 认证, 755
  - C-Tick 认证, 757
  - cULus 认证, 756
  - FM 认证, 756
  - 海事认证, 757
- 长度
  - 消息, 641
  - 长度 m, 641
  - 长度 n, 641
- 五划
  - 主动/被动连接, 459
  - 主动/被动通信
    - 连接 ID, 461
    - 参数, 132
    - 组态伙伴, 129, 532
  - 主站轮询架构, 644
  - 代码块, 149
    - DB (数据块), 69, 153
    - FB 的初始值, 152
    - FB (功能块), 69, 152

FC (功能), 69, 152  
 OB 数目, 23, 764, 776, 788, 801  
 与 CPU 或存储卡捆绑, 168  
 专有技术保护, 167  
 中断, 23, 764, 776, 788, 801  
 计数器 (数量和存储器要求), 24, 764, 776, 789, 801  
 代码块类型, 69  
 代码块数目, 23, 764, 776, 788, 801  
 用户程序的大小, 23, 764, 776, 788, 801  
 在用户程序内调用代码块, 149  
 有效 FC、FB 和 DB 号, 69  
 块调用, 69  
 定时器 (数量和存储器要求), 24, 764, 776, 789, 801  
 线性和结构化程序, 148  
 组织块 (OB), 23, 150, 764, 776, 788, 801  
 复制保护, 168  
 背景数据块 (DB), 152  
 监视, 23, 764, 776, 788, 801  
 嵌套深度, 23, 764, 776, 788, 801

## 四划

### 以太网

DPNRM\_DG, 305  
 DPRD\_DAT, 302  
 DPWR\_DAT, 302  
 GET, 526  
 IP 地址, 139  
 MAC 地址, 139  
 PUT, 526  
 RALRM, 294  
 RDREC, 291  
 T\_CONFIG, 486  
 TCON, 473

TDISCON, 473  
 TRCV, 473  
 TRCV\_C, 466  
 TSEND, 473  
 TSEND\_C, 466  
 TURCV, 481  
 TUSEND, 481  
 WRREC, 291  
 网络连接, 128  
 连接 ID, 461  
 特殊模式, 464  
 通信连接数, 458  
 通信类型, 457  
 概述, 463  
 以太网协议, 463  
 多节点连接, 530

## 五划

出厂设置复位, 734

### 功能 (FC)

专有技术保护, 167  
 在用户程序内调用代码块, 149  
 有效 FC 号, 69  
 线性和结构化程序, 148  
 概述, 69, 152

### 功能块 (FB)

专有技术保护, 167  
 有效 FB 号, 69  
 初始值, 152  
 单个 FB 使用多个背景数据块, 153  
 线性和结构化程序, 148  
 背景数据块, 152  
 概述, 69, 152  
 输出参数, 152

功率要求

- 计算功率预算, 918
- 功率预算, 917
  - 用于计算的表格, 920
- 功率预算, 44
  - 用于计算的表格, 920
  - 示例, 918
  - 概述, 917
- 发现上传在线 CPU, 123
- 发送参数组态, 129, 502, 532
- 发送消息组态, 634
- 发热区, 44, 47
- 可扩展指令, 37
- 可视化
  - HMI 设备, 29
- 外围设备访问报警, 507
- 布尔值或位值, 90
- 未指定的 CPU, 123
- 本地/伙伴连接, 459
- 生成用户定义的 Web 页面 DB, 579
- 用于 PtP 示例程序的终端仿真器, 652
- 用户定义 Web 页面的控制 DB
  - WWW 指令的参数, 579
  - 全局命令, 597
  - 请求命令和状态, 597
- 用户定义的 Web 页面, 541, 561
  - HTML 列表, 587
  - 下载相应 DB, 580
  - 手动片段 DB 控制, 597
  - 写入变量, 565
  - 写入特殊变量, 569
  - 处理特殊字符, 575
  - 生成程序块, 579
  - 用 WWW 指令启用, 579
  - 用于访问 S7-1200 数据的 AWP 命令, 563
  - 示例, 582
  - 创建片段, 573
  - 在 STEP 7 中进行编程, 579
  - 多语言, 594
  - 多语言组态, 597
  - 导入片段, 575
  - 删除程序块, 579
  - 使用 HTML 编辑器创建, 562
  - 刷新, 563
  - 组态, 577
  - 读取变量, 564
  - 读取特殊变量, 567
  - 通过 PC 访问, 581
  - 通过控制 DB 进行激活和取消激活, 597
  - 装载存储器限制, 581
- 用户定义的 Web 页面中的别名, 570
- 用户定义的 Web 页面中的枚举类型, 571, 572
- 用户定义的 Web 页面的数据块
  - 导入片段, 575
- 用户界面
  - 门户视图, 32
  - 项目视图, 33
- 用户程序
  - 下载, 169
  - 与 CPU 或存储卡捆绑, 168
  - 从在线 CPU 复制块, 170
  - 可扩展指令, 37
  - 传送卡, 111
  - 向 LAD 或 FBD 指令添加输入或输出, 36
  - 在用户程序内调用代码块, 149
  - 在编辑器之间拖放, 38
  - 存储卡, 111
  - 收藏夹, 34
  - 线性和结构化程序, 148
  - 组织块 (OB), 150
  - 密码保护, 167
  - 插入指令, 33
  - 程序卡, 111

- 电感负载, 66
- 电磁兼容性 (EMC), 758
- 目录, 用户定义 Web 页面的语言, 594
- 示例
  - Modbus 从站, 709
  - PtP 通信, 645
  - PtP 通信, STEP 7 编程, 651
  - PtP 通信, 运行, 652
  - PtP 通信, 组态, 646
  - PtP 通信, 终端仿真器, 652
  - 用户定义的 Web 页面, 582
  
- 六划**
- 丢失密码, 120
- 交叉引用, 173
  - 使用, 173
  - 简介, 173
- 任务卡
  - 列和标题, 37, 691
- 任务卡中的列和标题, 37
- 仿真器, 914
- 优先级
  - 处理优先级, 76
  - 优先等级, 75
- 优先等级, 75
- 传送卡, 114
  - 丢失密码, 120
  - 丢失密码时使用的空传送卡, 120
  - 运行, 111
  - 组态启动参数, 113
  - 插入 CPU 中, 112
  - 概述, 111
- 传送运行错误, 623
- 传送组态错误, 616
- 传送消息组态, 634
  - PtP 示例程序, 646
- 传输块 (T-block), 501
- 全局库
  - USS 协议概述, 653
- 全局数据块, 89, 153
- 关断延时 (TOF), 185
- 关断延迟 (TOF)
  - 运行, 189
- 创建用户定义的 Web 页面, 562
- 创建用户定义的 Web 页面 DB, 579
- 创建网络连接, 128
- 协议
  - ISO on TCP, 463
  - Modbus, 609
  - PROFINET RT, 463
  - TCP, 463
  - UDP, 463
  - USS, 609
  - 自由口, 609
  - 通信, 609
- 同步
  - 时间同步属性 (PROFINET), 145
- 向 LAD 或 FBD 指令添加输入或输出, 36
- 在用户程序内调用代码块, 149
- 在线
  - IP 地址, 734
  - RUN/STOP 按钮, 39
  - 工具, 738
  - 分配 IP 地址, 138
  - 比较并同步, 737
  - 存储器使用情况, 736
  - 时间, 734
  - 状态, 739
  - 诊断缓冲区, 736
  - 转到在线, 731
  - 重设 DB 的起始值, 740
  - 捕获 DB 的值, 740
  - 监视, 739

- 监视表格, 739, 741
- 强制, 744
- 强制操作, 745
- 循环时间, 736
- 操作员面板, 39
- 操作面板, 735
- 在线设备名称
  - PROFINET IO, 732
- 在线和诊断工具
  - 在 RUN 模式下下载, 746
- 在编辑器之间拖放, 38
- 多个 AWP 变量定义, 575
- 多节点连接
  - 以太网协议, 530
  - 连接类型, 530
- 字符位置
  - 消息长度, 641
- 字符序列
  - 消息开始, 636
  - 消息结束, 640
- 字符间隙, 640
- 存储区
  - 对布尔值或位值进行寻址, 90
  - 立即访问, 89
  - 过程映像, 89
- 存储卡
  - 丢失密码, 120
  - 丢失密码时使用的空传送卡, 120
  - 传送卡, 114
  - 运行, 111
  - 固件更新, 118
  - 组态启动参数, 113
  - 插入 CPU 中, 112
  - 程序卡, 116
  - 概述, 111
- 存储单元, 88, 90
- 存储器
  - I (过程映像输入), 90

- L (本地存储器), 89
- M (位存储器), 92
- Q (过程映像输出), 91
- 工作存储器, 83
- 外围设备输入地址 (强制表格), 744
- 时钟存储器, 85
- 系统存储器, 85
- 临时存储器, 92
- 保持性存储器, 83
- 监视存储器使用情况, 736
- 装载存储器, 83
- 安全性
  - CPU, 166
    - 与 CPU 或存储卡捆绑, 168
    - 代码块的专有技术保护, 167
    - 丢失密码, 120
    - 访问保护, 166
    - 复制保护, 168
- 安装
  - CPU, 49
  - DIN 导轨上的 TS 适配器, 61
  - TS 适配器 SIM 卡, 60
  - TS 适配器和 TS 模块, 59
  - 尺寸, 47
  - 气流, 44
  - 功率预算, 44
  - 发热区, 44, 47
  - 电感负载, 66
  - 安装尺寸, 47
  - 扩展电缆, 57
  - 灯负载, 66
  - 冷却, 44
  - 空隙, 44
  - 信号板 (SB), 51
  - 信号模块 (SM), 27, 53
  - 绝缘准则, 64



- 准则, 43
- 通信板 (CB), 51
- 通信模块 (CM), 55
- 接地, 65
- 接线准则, 63, 65
- 隔离, 64
- 概述, 43, 48
- 墙上的 TS 适配器, 62
- 端子板连接器, 56
- 安装要求, 31
- 寻址
  - 布尔值或位值, 90
  - 存储区, 89
  - 过程映像, 89
  - 单独输入 (I) 或输出 (Q), 90
- 导入 Siemens 安全证书, 603
- 延时中断, 315
- 当地时间
  - RD\_LOC\_T (读取本地时间), 260
- 扩展 S7-1200 的能力, 24
- 扩展电缆
  - 安装, 57
  - 卸下, 57
- 扫描周期
  - 强制, 744
  - 强制操作, 745
- 扫描周期时间
  - 概述, 81
- 收藏夹工具栏, 34
- 灯负载, 66
- 网页
  - STEP 7, 4
- 网络时间协议 (NTP), 144
- 网络连接
  - 多个 CPU, 498, 500, 504, 514, 518
  - 组态, 128
- 网络通信, 495
- 自由口协议, 609
- 西门子技术支持, 3
- 设计 PLC 系统, 147, 148
- 设备
  - PROFINET IO, 503
- 设备名称
  - PROFINET IO, 505
- 设备组态, 496
  - AS-i, 520
  - AS-i 端口, 520
  - 组态 CPU, 125
  - 添加新设备, 122
- 设备配置, 121
  - PROFIBUS, 515
  - PROFIBUS 端口, 515
  - PROFINET, 139
  - 下载, 169
  - 以太网端口, 139
  - 发现, 123
  - 网络连接, 128
  - 时间同步属性 (PROFINET), 145
  - 拔出的模块, 42
  - 组态模块, 127
  - 添加模块, 124
- 设置, 38
- 设置用户定义 Web 页面语言切换, 595
- 访问
  - PC 中的数据日志, 556
  - 用户定义的 Web 页面, 581
- 访问保护
  - CPU, 166
- 过程映像
  - 状态, 739, 744
  - 监视, 739
  - 强制, 744
  - 强制操作, 745

## 七划

串行通信, 609

位逻辑, 177

冷却, 44

冷端补偿

    热电偶, 852, 885

初始值

    捕获和重设 DB 的起始值, 740

启动 OB, 75

启动参数, 113

块

    FB 的初始值, 152

    OB 数目, 23, 76, 764, 776, 788, 801

    一致性检查, 174

    下载, 169

    中断, 23, 76, 764, 776, 788, 801

    从在线 CPU 复制块, 170

    计数器（数量和存储器要求）, 24, 764, 776, 789, 801

    代码块类型, 69

    代码块数目, 23, 764, 776, 788, 801

    功能 (FC), 69, 152

    功能块 (FB), 69, 152

    用户程序的大小, 23, 69, 764, 776, 788, 801

    有效 FC、FB 和 DB 号, 69

    启动 OB, 76

    块调用, 69

    事件, 76

    使用 SCL 调用 FB 或 FC, 158

    单个背景或多重背景数据块, 152

    定时器（数量和存储器要求）, 24, 764, 776, 789, 801

    线性和结构化程序, 148

    组织块 (OB), 23, 69, 75, 76, 764, 776, 788, 801

    背景数据块 (DB), 152

    监视, 23, 764, 776, 788, 801

    密码保护, 167

    嵌套深度, 23, 69, 764, 776, 788, 801

    数据块 (DB), 69

块移动 (MOVE\_BLK), 212

技术支持, 3

技术规范, 755

报警

    外围设备访问, 507

报警中断, 317

时间

    RD\_LOC\_T（读取本地时间）, 260

    RD\_SYS\_T（读取系统时间）, 260

    SET\_TIMEZONE（设置时区）, 263

    T\_ADD（加上时间）, 258

    T\_COMBINE（组合时间）, 259

    T\_CONV（转换时间）, 257

    T\_DIFF（时差）, 258

    T\_SUB（减去时间）, 258

    WR\_SYS\_T（写入系统时间）, 260

    对在线 CPU 进行组态, 734

时间同步属性, 145

时间错误中断 OB, 76

时钟

    RD\_LOC\_T（读取本地时间）, 260

    RD\_SYS\_T（读取系统时间）, 260

    WR\_SYS\_T（写入系统时间）, 260

    日时钟, 87

    存储器字节, 87

更改 STEP 7 的设置, 38

更换模块, 42

更新用户定义的 Web 页面, 563

状态

    LED 指令, 319

    LED 指示灯, 729

    LED 指示灯（通信接口）, 607

系统存储器字节, 86

系统时钟

- RD\_LOC\_T (读取本地时间), 260
- RD\_SYS\_T (读取系统时间), 260
- WR\_SYS\_T (写入系统时间), 260
- 系统要求, 31
- 证书导入向导, 603
- 诊断
  - DeviceStates, 321
  - GET\_DIAG, 324
  - LED 指令, 319
  - LED 指示灯, 729
  - ModuleStates, 322
  - 中断 OB, 507
  - 存储器使用情况, 736
  - 状态, 507
  - 状态指示, 86
  - 诊断缓冲区, 736
  - 监视表格, 741
  - 循环时间, 736
- 诊断缓冲区, 87, 736
- 诊断数据
  - 使用 GET\_DIAG 读出, 324
- 诊断错误中断 OB, 76
- 运动控制
  - MC\_ChangeDynamic, 428
  - MC\_CommandTable, 425
  - MC\_Halt, 414
  - MC\_Home, 411
  - MC\_MoveAbsolute, 416
  - MC\_MoveJog, 423
  - MC\_MoveRelative, 418
  - MC\_MoveVelocity, 420
  - MC\_Power, 407
  - MC\_Reset, 410
  - 回原点 (主动回原点的顺序), 438
  - 使组态参数回原点, 436
  - 使轴回原点, 435
- 组态轴, 401
- 硬件和软件限位开关, 432
- 概述, 397
- 运行时间计时器, 261
- 返回值
  - PtP 指令, 610
  - 开放式用户通信指令, 494
- 返回值 (RET), 241
- 远程服务通信
  - TM\_MAIL, 721
- 连接
  - S7 连接, 530
  - 以太网协议, 530
  - 伙伴, 129, 532
  - 连接 ID, 461
  - 连接数 (PROFINET), 458
  - 组态, 132
  - 类型, 多节点连接, 530
  - 通信类型, 457
- 连接, Web 服务器, 601
- 连接触点
  - 最大载流能力, 905
- 连接器
  - 安装和拆卸, 56
- 八划
- 事件执行, 76
- 其它 PtP 参数错误, 611
- 函数块 (FB)
  - 在用户程序内调用代码块, 149
- 刷新用户定义的 Web 页面, 563
- 参数分配, 152
- 参数组态
  - SEND\_PTP 的 LENGH 和 BUFFER, 624
  - 传送, 502
  - 接收, 503

- 取反 (INV), 249
- 固件更新, 118
- 奇偶校验, 632
- 定义枚举类型, 用户定义的 Web 页面, 571
- 定时器
  - RT (重置定时器), 185
  - TOF (关断延时定时器), 185
  - TON (接通延时定时器), 185
  - TONR (保持型接通延时) 定时器, 185
  - TP (脉冲延时定时器), 185
  - 大小, 24, 764, 776, 789, 801
  - 运行, 189
  - 数量, 24, 764, 776, 789, 801
- 拔出的模块, 42
- 服务与支持, 3
- 波特率, 632
- 环境
  - 工业环境, 758
  - 运行条件, 759
  - 运输和存储条件, 759
- 空隙
  - 气流和冷却, 44
- 线性编程, 148
- 线路空闲, 635, 636
- 组态
  - AS-i, 520
  - AS-i 地址, 520
  - AS-i 端口, 519
  - CPU 参数, 125
  - HSC (高速计数器), 374
  - IP 地址, 139
  - MAC 地址, 139
  - PLC 到 PLC 通信, 499
  - PROFIBUS, 515
  - PROFIBUS 地址, 515
  - PROFIBUS 端口, 515
  - PROFINET, 139
  - 下载, 169
  - 以太网端口, 139
  - 用户定义 Web 页面 (多语言), 597
  - 用户定义的 Web 页面, 577
  - 启动参数, 113
  - 时间同步属性 (PROFINET), 145
  - 通信负载, 83
  - 通信接口, 630
  - 接收消息, 635
  - 添加模块, 124
  - 循环时间, 81
  - 模块, 127
  - 端口, 630
- 组织块
  - 专有技术保护, 167
  - 功能, 75
  - 处理, 150
  - 优先等级, 75
  - 创建, 151
  - 在用户程序内调用代码块, 149
  - 多个循环, 151
  - 启动过程, 73
  - 线性 and 结构化编程, 148
  - 组态运行, 152
  - 期间, 75
  - 概述, 69
- 规范
  - ATEX 认证, 756
  - CB 1241 RS485, 907
  - CE 认证, 755
  - CM 1241 RS232, 910
  - CPU 1211C, 762
  - CPU 1212C, 774
  - CPU 1214C, 786
  - CPU 1215C, 799
  - C-Tick 认证, 757

- cULus 认证, 756
  - FM 认证, 756
  - SB 1221 DI 4, 200 kHz, 861
  - SB 1222 DQ 4, 200 kHz, 863
  - SB 1223 DI 2 x/DQ 2, 200 kHz, 866
  - SB 1223 DI 2/DQ 2, 870
  - SB 1231 AI 1 x 12 位, 874
  - SB 1231 AI 1 x 16 位 RTD, 888
  - SB 1231 AI 1 x 16 位 RTD 的接线图, 890
  - SB 1231 AI 1 x 16 位热电偶, 883
  - SB 1231 AI x 12 位的接线图, 876
  - SB 1232 AQ 1 x 12 位, 878
  - SM 1221 信号模块, 813
  - SM 1221 接线图, 814
  - SM 1222 DQ8 RLY (切换), 817
  - SM 1222 信号模块, 817, 819
  - SM 1222 接线图, 819
  - SM 1223 信号模块, 826, 829
  - SM 1223 接线图, 826, 829
  - SM 1231 AI 4 x 13 位, 833
  - SM 1231 AI 4 x 16 位 TC 信号模块, 848
  - SM 1231 AI 4 x 16 位 TC 接线图, 850
  - SM 1231 AI 4 x RTD x 16 位信号模块, 854
  - SM 1231 AI 8 x 16 位 TC 信号模块, 848
  - SM 1231 AI 8 x 16 位 TC 接线图, 850
  - SM 1231 AI 8 x RTD x 16 位信号模块, 854
  - SM 1231 RTD 4 x 16 位的接线图, 856
  - SM 1231 RTD 8 x 16 位的接线图, 856
  - SM 1231 模拟量输入的接线图, 837
  - SM 1232 AQ 2 x 14 位, 838
  - SM 1232 AQ 4 x 14 位, 838
  - SM 1232 模拟量输出的接线图, 839
  - SM 1234 AI 4 x 13 位/AQ 2 x 14 位, 840
  - SM 1234 模拟量输入/输出的接线图, 843
  - 工业环境, 758
  - 电磁兼容性 (EMC), 758
  - 阶跃响应时间 (CPU), 769, 782, 794, 806
  - 阶跃响应时间 (SB), 879
  - 阶跃响应时间 (SM), 844
  - 环境条件, 759
  - 保护, 760
  - 海事认证, 757
  - 继电器电气使用寿命, 762
  - 常规技术规范, 755
  - 输入仿真器, 914
  - 模拟量输入的电压表示法, 845, 880
  - 模拟量输入的电流表示法, 846, 881
  - 模拟量输出的电压表示法, 846, 881
  - 模拟量输出的电流表示法, 847, 882
  - 额定电压, 761
  - 转换 (SCL 指令), 221
  - 轮询架构, 644
  - 软件流控制, 633
  - 限制
    - Web 服务器, 601
    - 用户定义的 Web 页面, 581
- ## 九划
- 保护等级, 760
    - CPU, 166
    - 与 CPU 或存储卡捆绑, 168
  - 代码块, 167
  - 丢失密码, 120
  - 保持性存储器, 22, 83
    - CPU 1211C, 762
    - CPU 1212C, 774
    - CPU 1214C, 786
    - CPU 1215C, 799
  - 保持型接通延时 (TONR), 185
  - 保持型接通延迟 (TONR)
    - 运行, 189

信号处理错误, 629, 630

#### 信号板 (SB)

SB 1221 DI 4, 200 kHz, 861

SB 1222 DQ, 200 kHz, 863

SB 1223 DI 2/DQ 2, 870

SB 1223 DI 2/Q 2, 200 kHz, 866

SB 1231 AI 1 x 12 位, 874

SB 1231 AI 1 x 16 位 RTD, 888

SB 1231 AI 1 x 16 位热电偶, 883

SB 1232 AQ 1 x 12 位, 878

功率要求, 917

安装, 51

阶跃响应时间, 879

参数的组态, 127

卸下, 51

添加模块, 124

概述, 27

输入的电压表示法, 845, 880

输入的电流表示法, 846, 881

模拟量输出的电压表示法, 846, 881

模拟量输出的电流表示法, 847, 882

#### 信号模块 (SM)

SM 1221, 813

SM 1222, 817, 819

SM 1222 DQ8 RLY (切换), 817

SM 1223, 826

SM 1223 的规范, 829

SM 1231 AI 4 x 13 位, 833

SM 1231 AI 4 x 16 位 TC, 848

SM 1231 AI 4 x RTD x 16 位, 854

SM 1231 AI 8 x 16 位 TC, 848

SM 1231 AI 8 x RTD x 16 位, 854

SM 1232 AQ 2 x 14 位, 838

SM 1232 AQ 4 x 14 位, 838

SM 1234 AI 4 x 13 位/AQ 2 x 14 位, 840

功率要求, 917

安装, 53

扩展电缆, 57

阶跃响应时间, 844

参数的组态, 127

卸下, 54

添加模块, 124

概述, 27

模拟量输入的电压表示法, 845, 880

模拟量输入的电流表示法, 846, 881

模拟量输出的电压表示法, 846, 881

模拟量输出的电流表示法, 847, 882

信息资源, 4

#### 修改

监视表格, 741

程序编辑器状态, 739

## 八划

### 变量

状态, 739

监视, 739

通过 PC 进行监视和修改, 554

强制, 744

强制操作, 745

## 九划

复位, 180

复位为出厂设置, 734

### 复制保护

与 CPU 或存储卡捆绑, 168

客户支持, 3

将存储卡插入 CPU, 112

总线连接器, 27

### 指令

ABS (绝对值), 207

ACOS (反余弦), 210

- ADD (加法), 204  
 AND, 248  
 AS-i 分布式 I/O, 290  
 ASIN (反正弦), 210  
 ATAN (反正切), 210  
 ATH (ASCII 到十六进制), 279  
 ATTACH, 308  
 CALCULATE, 34, 203  
 CAN\_DINT (取消延时中断), 315  
 CASE (SCL), 231  
 CEIL (上取整), 225  
 CONCAT (连接), 282  
 CONTINUE (SCL), 235  
 CONV (转换), 220  
 COS (余弦), 210  
 CTD (减计数), 193  
 CTRL\_PWM), 333  
 CTU (加计数), 193  
 CTUD (加计数和减计数), 193  
 DataLogClose (关闭数据日志), 344  
 DataLogCreate (创建数据日志), 339  
 DataLogNewFile (根据现有数据日志创建数据日志), 347  
 DataLogOpen (打开数据日志), 342  
 DataLogWrite (写入数据日志), 346  
 DEC (递减), 207  
 DECO (解码), 249  
 DELETE (删除子串), 285  
 DETACH, 308  
 DeviceStates, 321  
 DIS\_AIRT (禁用报警中断), 317  
 DIV (除法), 204  
 DPNRM\_DG, 305  
 DPRD\_DAT, 302  
 DPWR\_DAT, 302  
 EN\_AIRT (启用报警中断), 317  
 ENCO (编码), 249  
 EXIT (SCL), 236  
 EXP (自然指数), 210  
 EXPT (一般指数), 210  
 FieldRead, 215  
 FieldWrite, 215  
 FILL\_BLK, 217  
 FIND (查找子串), 289  
 FLOOR, 225  
 FOR (SCL), 232  
 FRAC (分数), 210  
 GET, 526  
 GET\_DIAG, 324  
 GetError, 244  
 GetErrorID, 245  
 GOTO (SCL), 236  
 HSC (高速计数器), 363, 365  
 HTA (十六进制到 ASCII), 280  
 IF-THEN (SCL), 230  
 IN\_RANGE (在范围之内), 201  
 INC (递增), 207  
 INSERT (插入子串), 286  
 INV (取反), 249  
 JMP, 237  
 JMP\_LIST, 238  
 LED 状态, 319  
 LEFT (左侧子串), 284  
 LEN (长度), 282  
 LN (自然对数), 210  
 MAX (最大值), 208  
 MB\_CLIENT, 673  
 MC\_ChangeDynamic, 428  
 MC\_CommandTable, 425  
 MC\_Halt, 414  
 MC\_Home, 411  
 MC\_MoveAbsolute, 416

- MC\_MoveJog, 423  
MC\_MoveRelative, 418  
MC\_MoveVelocity, 420  
MC\_Power, 407  
MC\_Reset, 410  
MID (中间子串), 284  
MIN (最小值), 208  
MOD (求模), 205  
ModuleStates, 322  
MOVE, 212  
MUL (乘法), 204  
MUX (多路复用), 251  
N\_TRIG, 184  
NEG (取反), 206  
NORM\_X (标准化), 226  
NOT OK, 202  
OK, 202  
OR, 248  
OUT\_RANGE (在范围之外), 201  
P\_TRIG, 184  
PID\_Compact, 378  
PORT\_CFG (端口组态), 613  
PROFIBUS 分布式 I/O, 290  
PROFINET 分布式 I/O, 290  
PUT, 526  
QRY\_CINT (查询循环中断), 313  
RALRM, 294  
RCV\_CFG (接收组态), 616  
RCV\_PtP (接收点对点), 625  
RCV\_RST (接收方复位), 627  
RD\_LOC\_T (读取本地时间), 260  
RD\_SYS\_T (读取系统时间), 260  
RDREC, 291  
RE\_TRIGR, 81, 242  
REPEAT (SCL), 234  
REPLACE (替换子串), 287  
RETURN (SCL), 237  
RIGHT (右侧子串), 284  
ROL 和 ROR (循环左移和循环右移), 255  
ROUND, 224  
RT (重置定时器), 185  
S\_CONV (值到字符串的转换), 266  
S\_MOV (字符串移动), 266  
SCALE\_X (标定), 226  
SCL 转换指令, 221  
SEL (选择), 251  
SEND\_CFG (发送组态), 615  
SEND\_PTP (发送点对点数据), 622  
SET\_CINT (设置循环中断), 311  
SET\_TIMEZONE, 263  
SGN\_GET (获取 RS232 信号), 628  
SGN\_SET (设置 RS232 信号), 629  
SHL 和 SHR (左移和右移), 254  
SIN (正弦), 210  
SQR (平方), 210  
SQRT (平方根), 210  
SRT\_DINT (启动延时中断), 315  
STP (停止 PLC 扫描循环), 243  
STRG\_VAL (字符串到值), 266  
SUB (减法), 204  
SWAP, 218  
SWITCH, 239  
T\_ADD (加上时间), 258  
T\_COMBINE (组合时间), 259  
T\_CONFIG, 486  
T\_CONV (转换时间), 257  
T\_DIFF (时差), 258  
T\_SUB (减去时间), 258  
TAN (正切), 210  
TCON, 473  
TDISCON, 473  
TOF (关断延时定时器), 185



- TON (接通延时定时器), 185
- TONR (保持型接通延时定时器), 185
- TP (脉冲定时器), 185
- TRCV, 473
- TRCV\_C, 466, 502
- TRUNC (截取), 224
- TSEND, 473
- TSEND\_C, 466, 501
- TURCV, 481
- TUSEND, 481
- UFILL\_BLK (不中断填充), 217
- USS 状态代码, 664
- USS\_DRV, 657
- USS\_PORT, 660
- USS\_RPM, 661
- USS\_WPM, 663
- VAL\_STRG (值到字符串), 266
- WHILE (SCL), 233
- WR\_SYS\_T (写入系统时间), 260
- WRREC, 291
- WWW (启用用户定义 Web 页面), 579
- XOR, 248
- 上升沿, 183
- 下降沿, 183
- 不中断移动 (UMOVE\_BLK), 212
- 日历, 257
- 日期, 257
- 比较, 200
- 计数器, 193
- 可扩展指令, 37
- 列和标题, 37, 691
- 向 LAD 或 FBD 指令添加输入或输出, 36
- 在编辑器之间拖放, 38
- 收藏夹, 34
- 位逻辑, 177
- 块移动 (MOVE\_BLK), 212
- 时间, 257
- 时钟, 260
- 状态, 739
- 运行时间计时器, 261
- 返回值 (RET), 241
- 定时器, 185
- 定时器运行, 189
- 拖放, 33
- 限制, 209
- 复位, 180
- 指令版本, 37, 691
- 标定模拟值, 35
- 标签, 237
- 浮点型算术运算, 210
- 监视, 739
- 常见参数, 493
- 强制, 744
- 强制操作, 745
- 插入, 33
- 程序控制 (SCL), 229
- 数据块控制, 358
- 置位, 180
- 指令版本, 37, 691
- 故障排除
  - LED 指示灯, 729
  - 诊断缓冲区, 736
- 显示 MAC 地址和 IP 地址, 143
- 标定模拟值, 35, 227
- 标准 Web 页面, 541
  - communication, 553
  - cookie 限制, 603
  - Data Logs, 556
  - Diagnostic, 549
  - Identification, 549
  - Intro, 547
  - JavaScript 限制, 602

- Module information, 550
  - Start, 548
  - Variable Status, 554
  - 布局, 545
  - 安全访问, 544
  - 更改操作模式, 548
  - 通过 PC 访问, 543
  - 登录和注销, 546
  - 标准化模拟值, 227
  - 标签, 237
  - 测试程序, 172
  - 点对点通信, 609
  - 点对点编程, 643
  - 结束条件, 638
  - 结构化编程, 148, 149
    - 调用块, 149
  - 绝缘准则, 64
  - 背景数据块, 89
  - 脉冲延时 (TP), 185
  - 脉冲延迟 (TP)
    - 运行, 189
  - 脉冲输出, 335
  - 要求, 安装, 31
  - 语言, 用户定义 Web 页面, 594
  - 重设 DB 的起始值, 740
  - 重置定时器 (RT), 185
  - 面板 (HMI), 29
  - 项目
    - 下载, 169
    - 与 CPU 或存储卡捆绑, 168
    - 比较并同步, 737
    - 丢失密码, 120
    - 传送卡, 114
    - 访问保护, 166
    - 空传送卡, 120
    - 限制对 CPU 的访问, 166
    - 保护代码块, 167
    - 程序卡, 116
  - 项目视图, 32, 33
  - 首次扫描指示, 86
- ## 十划
- ### 准则
- CPU 安装, 49
  - 电感负载, 66
    - 安装, 43
    - 安装步骤, 48
  - 灯负载, 66
  - 接地, 65
  - 接线准则, 63, 65
  - 隔离, 64
  - 捕获在线 DB 的值, 740
  - 流控制, 632
    - 组态, 632
  - 浮点型算术运算, 210
  - 海事认证, 757
  - 消息开始, 636
  - 消息开始字符, 636
  - 消息长度, 640
  - 消息组态
    - 传送, 634
    - 指令, 643
    - 接收, 635
  - 消息结束, 638
  - 消息结束字符, 640
  - 热电偶
    - SB 1231 热电偶过滤器选型表, 885
    - SB 1231 热电偶选型表, 885
    - SM 1231 热电偶选型表, 852
    - SM 1231 热电偶滤波器选型表, 852
    - 冷端补偿, 852, 885
    - 基本操作, 852, 885

- 热线, 3
- 特殊字符
  - 用户定义的 Web 页面, 575
- 特殊模式
  - ISO on TCP, 464
  - TCP, 464
- 监视
  - LAD 状态, 739
  - LED 指令, 319
  - 存储器使用情况, 736
  - 重设 DB 的起始值, 740
  - 捕获 DB 的值, 740
  - 监视表格, 739, 741
  - 强制表格, 744
  - 强制操作, 745
  - 循环时间, 736
- 监视狗, 242
- 监视表格
  - 在 STOP 模式下启用输出, 743
  - 存储卡, 111
  - 运行, 741
  - 监视, 739
  - 强制, 172
  - 触发器值, 743
- 监视程序, 172
- 继电器电气使用寿命, 762
- 获取 LED 状态, 319
- 被动/主动通信
  - 连接 ID, 461
  - 参数, 132
  - 组态伙伴, 129, 532
- 读取 HTTP 变量, 567
- 调用结构, 174
- 调试
  - 在 RUN 模式下下载, 746, 751
- 通过 GPRS 实现 TeleService, 712
- 通过 PC 修改变量, 554
- 通过 PC 监视变量, 554
- 通过变量索引数组, 216
- 通信
  - AS-i 地址, 520
  - IP 地址, 139
  - MAC 地址, 139
  - PROFIBUS 地址, 515
  - PROFINET 和 PROFIBUS, 457
  - TCON\_Param, 132
  - 主动/被动, 129, 132, 532
  - 发送和接收参数, 634
  - 网络, 495
  - 网络连接, 128
  - 时间同步属性 (PROFINET), 145
  - 连接 ID, 461
  - 连接数 (PROFINET), 458
  - 参数, 132
  - 组态, 129, 132, 532
  - 轮询架构, 644
  - 流控制, 632
  - 通信负载, 83
  - 循环时间, 83
  - 硬件连接, 495
- 通信处理器 (CP)
  - 比较表, 24
  - 设备配置, 121
  - 参数的组态, 127
  - 添加模块, 124
  - 概述, 27
- 通信板 (CB)
  - CB 1241 RS485, 907
  - LED 指示灯, 607, 729
  - RS485, 607
  - 比较表, 24
  - 安装, 51
  - 设备配置, 121

- 参数的组态, 127
- 卸下, 51
- 添加模块, 124
- 编程, 643
- 概述, 27
- 通信接口
  - CB 1241 RS485, 907
  - CM 1241 RS232, 910
  - LED 指示灯, 729
  - RS232 和 RS485, 607
  - 设备配置, 121
  - 组态, 630
  - 添加模块, 124
  - 编程, 643
  - 模块比较表, 24
- 通信模块
  - CM 1241 RS422/RS485 技术规范, 910
- 通信模块 (CM)
  - CM 1241 RS232 规范, 910
  - LED 指示灯, 607, 729
  - PtP 示例程序组态, 646
  - RS232 和 RS485, 607
  - 比较表, 24
  - 功率要求, 917
  - 安装, 55
  - 设备配置, 121
  - 参数的组态, 127
  - 卸下, 55
  - 添加 AS-i 主站 CM1243-2 模块, 518
  - 添加 CM 1243-5 (DP 主站) 模块, 514
  - 添加模块, 124
  - 编程, 643
  - 数据接收, 625
  - 概述, 27
- 配置
  - 发现, 123
  - 网络连接, 128
- 高速计数器
  - HSC, 363
  - 运行, 365
  - 组态, 374
- 高速计数器 (HSC)
  - 无法进行强制, 745
- 十一划
  - 停止位, 632
  - 基本型面板 (HMI), 29
  - 密码保护
    - CPU, 166
    - 与 CPU 或存储卡捆绑, 168
    - 代码块, 167
    - 对 CPU 进行访问, 166
    - 丢失密码, 120
    - 空传送卡, 120
    - 复制保护, 168
  - 常见问题解答, 4
  - 常规技术规范, 755
  - 排队, 76
  - 接收运行返回值, 625
  - 接收参数组态, 503
  - 接收组态错误, 621
  - 接收消息组态, 635
    - PtP 示例程序, 646
  - 接线图
    - CPU 1211C, 773
    - CPU 1212C, 785
    - CPU 1215C, 812
    - SB 1222 DQ 4, 200 kHz, 865
    - SB 1223 DI 2/DQ 2, 873
    - SB 1223 DI 2/DQ 2, 200 kHz, 869
    - SB 1231 AI 1 x 16 位 RTD, 890
    - SB 1231 AI x 12 位, 876

- SB 1232 AQ 1 x 12 位, 878
- SM 1221 信号模块, 814
- SM 1222 信号模块, 819
- SM 1223 信号模块, 826, 829
- SM 1231 AI 4 x 16 位 TC, 850
- SM 1231 AI 8 x 16 位 TC, 850
- SM 1231 RTD 4 x 16 位, 856
- SM 1231 RTD 8 x 16 位, 856
- SM 1231 模拟量输入, 837
- SM 1232 模拟量输出, 839
- SM 1234 模拟量输入/输出, 843
- 接线准则, 65
  - 气流和冷却空隙, 44
  - 先决条件, 63
  - 接地, 65
- 接通延时 (TON), 185
- 接通延迟 (TON)
  - 运行, 189
- 添加新设备
  - CPU, 122
  - 未指定的 CPU, 123
  - 检测现有硬件, 123
- 十二划**
- 嵌套深度, 69
- 强制, 744, 745
  - I 存储器, 744, 745
  - 外围设备输入, 744, 745
  - 存储卡, 111
  - 扫描周期, 745
  - 监视表格, 741
  - 输入和输出, 745
- 强制表格
  - 寻址外围设备输入, 744
  - 强制, 744
  - 强制操作, 745
- 循环中断 OB, 75
- 循环时间
  - 组态, 83
  - 监视, 736
  - 概述, 81
- 插入设备
  - 未指定的 CPU, 123
- 插入指令
  - 在编辑器之间拖放, 38
  - 收藏夹, 34
  - 拖放, 33
- 最大消息长度, 640
- 登录/注销
  - 标准 Web 页面, 546
- 硬件中断 OB, 75
- 硬件组态
  - AS-i, 520
  - AS-i 端口, 520
- 硬件流控制, 632
- 硬件配置, 121
  - PROFIBUS, 515
  - PROFIBUS 端口, 515
  - PROFINET, 139
  - 下载, 169
  - 以太网端口, 139
  - 发现, 123
  - 网络连接, 128
  - 组态 CPU, 125
  - 组态模块, 127
  - 添加新设备, 122
  - 添加模块, 124
- 程序
  - 下载, 169
  - 与 CPU 或存储卡捆绑, 168
  - 从在线 CPU 复制块, 170
  - 优先等级, 75

- 在用户程序内调用代码块, 149
- 存储卡, 111
- 线性和结构化程序, 148
- 组织块 (OB), 150
- 重设 DB 的起始值, 740
- 捕获 DB 的值, 740
- 密码保护, 167
- 程序卡
  - 创建, 116
  - 运行, 111
  - 组态启动参数, 113
  - 插入 CPU 中, 112
  - 概述, 111
- 程序执行, 69
- 程序信息
  - 在调用结构中, 174
- 程序结构, 149
  - 调用代码块, 149
- 程序控制 (SCL), 229
  - CASE, 231
  - CONTINUE, 235
  - EXIT, 236
  - FOR, 232
  - GO TO, 236
  - IF-THEN, 230
  - REPEAT, 234
  - RETURN, 237
  - WHILE, 233
- 程序编辑器
  - 状态, 739
  - 重设 DB 的起始值, 740
  - 捕获 DB 的值, 740
  - 监视, 739
- 等待时间, 76, 632
- 编程
  - CPU 的工作模式, 71
  - FB 的初始值, 152
  - FBD (功能块图), 157
  - LAD (梯形图), 156
  - PID 概述, 375
  - PID\_3STEP, 384
  - PID\_3Step 算法, 375, 384
  - PID\_Compact, 378
  - PID\_Compact 算法, 375, 378
  - PtP 指令, 643
  - RD\_LOC\_T (读取本地时间), 260
  - RD\_SYS\_T (读取系统时间), 260
  - SCL (结构化控制语言), 157, 158
  - WR\_SYS\_T (写入系统时间), 260
  - 与 CPU 或存储卡捆绑, 168
  - 比较并同步代码块, 737
  - 计数器, 193
  - 代码块类型, 69
  - 功能 (FC), 152
  - 功能块 (FB), 69, 152
  - 可扩展指令, 37
  - 未指定的 CPU, 123
  - 优先等级, 75
  - 向 LAD 或 FBD 指令添加输入或输出, 36
  - 在用户程序内调用代码块, 149
  - 在编辑器之间拖放, 38
  - 收藏夹, 34
  - 有效 FC、FB 和 DB 号, 69
  - 块调用, 69
  - 运行时间计时器, 261
  - 拔出的模块, 42
  - 线性程序, 148
  - 组织块 (OB), 150
  - 结构化程序, 148
  - 背景数据块 (DB), 152
  - 能流 (EN 和 ENO), 164
  - 插入指令, 33
  - 数据块 (DB), 69

- 联系信息, 3
- 装载存储器, 22
  - CPU 1211C, 762
  - CPU 1212C, 774
  - CPU 1214C, 786
  - CPU 1215C, 799
  - 传送卡, 111
  - 存储卡, 111
  - 程序卡, 111
- 装载存储器, 用户定义 Web 页面, 581
  
- 十三划**
- 数字信号板 (SB) 规范
  - SB 1221 DI 4, 200 kHz, 861
  - SB 1222 DQ 4, 200 kHz, 863
  - SB 1223 DI 2/DQ 2, 870
  - SB 1223 DI 2/Q 2, 200 kHz, 866
- 数字信号模块 (SM)
  - SM 1221, 813
  - SM 1222, 817, 819
  - SM 1223, 826
- 数字量 I/O
  - 状态指示灯, 730
  - 组态, 127
- 数学, 34, 203, 204
- 数组
  - 通过变量访问成员, 216
- 数组的变量索引, 216
- 数值
  - 二进制, 96
  - 实数, 97
  - 整型, 97
- 数据日志
  - DataLogClose (关闭数据日志), 344
  - DataLogCreate (创建数据日志), 339
  - DataLogNewFile (根据现有数据日志创建数据日志), 347
  - DataLogOpen (打开数据日志), 342
  - DataLogWrite (写入数据日志), 346
  - 示例程序, 354
  - 查看数据日志, 350
  - 数据日志大小限制, 351
  - 数据日志概述, 338
  - 数据记录结构, 338
- 数据处理块 (DHB), 153
- 数据传输, 启动, 622
- 数据块
  - CONF\_DATA, 489
  - 全局数据块, 89, 153
  - 有效 DB 号, 69
  - 单个 FB 使用多个背景数据块, 153
  - 组织块 (OB), 150
  - 背景数据块, 89
  - 重置起始值, 740
  - 捕获值, 740
  - 概述, 69, 153
- 数据块控制, 358
- 数据类型, 94
  - Any (指针), 105
  - Bool、Byte、Word 和 DWord, 96
  - Char (字符) 和字符串, 100
  - PLC 数据类型编辑器, 103
  - Pointer (指针), 104
  - Real、LReal (浮点型实数), 97
  - Struc, 103
  - Time、Date、TOD (日时钟)、DTL (日期和时间长型), 98
  - USInt、SInt、UInt、Int、UDInt、Dint (整型), 97
  - Variant (指针), 107
  - 指针数据类型概述, 103
  - 数组, 101

- 暖启动, 71
- 置位, 180
- 触发监视表格中的值, 743
- 路由器 IP 地址, 140
- 输入仿真器, 914
- 输入和输出
  - 监视, 739
- 输出参数, 152
  - 组态脉冲通道, 337
  - 脉冲输出, 335
- 遥控, 712
- 错误
  - 时间错误, 79
  - 诊断错误, 80
- 错误代码
  - RALRM, 297
  - RDREC, 297
  - WRREC, 297
  - 扩展指令的常见错误, 361
- 频率、时钟位, 87
  
- 十四划**
- 模块
  - CB 1241 RS485, 907
  - CM 1241 RS232, 910
  - CPU 1211C 规范, 762
  - CPU 1212C 规范, 774
  - CPU 1214C 规范, 786
  - CPU 1215C 规范, 799
  - SB 1221 DI 4, 200 kHz, 861
  - SB 1222 DQ 4, 200 kHz, 200 kHz, 863
  - SB 1223 DI 2/DQ 2, 870
  - SB 1223 DI 2/Q 2, 200 kHz, 866
  - SB 1231 AI 1 x 12 位, 874
  - SB 1231 AI 1 x 16 位 RTD, 888
  - SB 1231 AI 1 x 16 位热电偶信号板, 883
  - SB 1232 AQ 1 x 12 位, 878
  - SM 1221, 813
  - SM 1222, 817, 819
  - SM 1222 DQ8 RLY (切换), 817
  - SM 1223, 826
  - SM 1231 AI 4 x 16 位 TC, 848
  - SM 1231 AI 4 x RTD x 16 位, 854
  - SM 1231 AI 8 x 13 位, 833
  - SM 1231 AI 8 x 16 位 TC 信号模块, 848
  - SM 1231 AI 8 x RTD x 16 位, 854
  - SM 1232 AQ 2 x 14 位, 838
  - SM 1232 AQ 4 x 14 位, 838
  - SM 1234 AI 4 x 13 位/AQ 2 x 14 位, 840
  - 比较表, 24
  - 发热区, 44, 47
  - 组态参数, 127
  - 信号板 (SB), 27
  - 信号模块 (SM), 27
  - 通信处理器 (CP), 27
  - 通信板 (CB), 27
  - 通信模块 (CM), 27
  - 模拟信号板 (SB)
    - SB 1231 AI 1 x 12 位, 874
    - SB 1231 AI 1 x 16 位 RTD, 888
    - SB 1231 AI 1 x 16 位热电偶, 883
    - SB 1232 AQ 1 x 12 位, 878
  - 模拟量 I/O
    - 阶跃响应时间 (CPU), 769, 782, 794, 806
    - 阶跃响应时间 (SB), 879
    - 阶跃响应时间 (SM), 844
    - 状态指示灯, 730
    - 组态, 127
    - 转换为工程单位, 35, 94, 227
    - 输入的电压表示法, 845, 880
    - 输入的电流表示法, 846, 881
    - 输出的电压表示法, 846, 881



输出的电流表示法, 847, 882

模拟量信号 (SM)

SM 1232 AQ 4 x 14 位, 838

模拟量信号模块 (SM)

SM 1231 AI 4 x RTD x 16 位, 854

SM 1231 AI 8 x 13 位, 833

SM 1231 AI 8 x 16 位 TC, 848

SM 1231 AI 8 x RTD x 16 位, 854

SM 1231 AI4 x 16 位 TC, 848

SM 1232 AQ 2 x 14 位, 838

SM 1234 AI 4 x 13 位/AQ 2 x 14 位, 840

端口号, 463

限制, 495

端口组态, 630

PtP 示例程序, 646

指令, 643

错误, 614

端子板连接器, 56

## 十五划

播客, 4

额定电压, 761

## 十六划

操作员面板, 29, 39

CPU 的工作模式, 71

