

## SIMATIC

### S7-200 可编程控制器 系统手册

6ES7298-8FA22-8FH0

第4版

前言、目录	
产品概述	1
使用入门	2
安装S7-200	3
PLC原理	4
编程原则、惯例和特征	5
S7-200指令集	6
通过网络进行通讯	7
硬件故障检测指南和软件调试工具	8
为定位模块创建程序	9
为调制解调器模块创建程序	10
使用USS协议库控制MicroMaster 驱动器	11
使用Modbus协议库	12
技术规范	A
计算电源预算	B
错误代码	C
特殊内存（SM）位	D
S7-200订购号	E
STL指令的执行时间	F
S7-200快速参考信息	G
索引	

## 安全指南

本手册包含应该引起注意的安全事项，以确保人身安全，并保护产品及与之连接的设备。这些注意事项在手册中由一个警告三角符号突出显示，并根据危险的程度不同作如下标记：



### 危险

表示极其危险，如果不避免，将导致死亡或严重的人身伤害。



### 警告

表示具有潜在危险，如果不避免，可能导致死亡或严重人身伤害。



### 当心

与安全警示标记一起使用，表示具有潜在危险，如果不避免，可能导致轻微或中等程度的人身伤害。

### 当心

不与安全警示标记一起使用，表示具有潜在危险，如果不避免，将可能导致财产损失。

### 注意

表示如果无法避免，将可能产生不希望发生的结果或状态。

## 合格人员

只有合格人员才允许对该设备进行安装和操作。合格人员的定义是：根据已建立的安全实践和标准，经过授权允许进行调试、接地以及连接电路、设备和系统的人员。

## 正确应用

注意：



### 警告

该设备及其组件只能用于目录或技术说明中描述的应用程序，只能与西门子允许或建议的其它厂商的设备或组件连接。

只有在正确运输、存储、设置和安装并且按照建议进行操作和维护的情况下，该产品才能正确而安全地运行。

## 注册商标

SIMATIC®、SIMATIC HMI®和SIMATIC NET®是SIEMENS AG的注册商标。

这些文档使用的一些其它标志也是注册商标；如果第三方出于自身利益擅自使用，则可能侵犯商标所有人的合法权益。

版权所有Siemens AG 2003保留所有权利

没有明确的书面授权，不允许对本文档或其内容进行复制、传播或使用。违反者将要对此进行赔偿。所有权利，包括专利权和申请模式或设计的专利权，都予以保留。

拒绝承担责任的声明

已经对本手册的内容与所描述的硬件与软件进行一致性检查。由于不可能完全避免偏差，所以不能保证完全一致。但是，本手册中的数据会定期进行检查，并且所有必要的更正都将包含于随后的版本中。欢迎提出改进建议。

Siemens AG  
自动化和驱动部门 (A&D)  
工业自动化系统 (AS)  
Postfach 4848, D-90327 Nürnberg

© Siemens AG 2003  
技术数据以修改后的数据为准。

Siemens Aktiengesellschaft

# 前言

S7-200系列是一组可以控制许多自动化应用程序的微型可编程逻辑控制器（微型PLC）。紧凑设计、低成本和功能强大的指令集使得S7-200成为控制小型应用程序的完美解决方案。S7-200的多种型号和基于Windows的编程工具提供了解决自动化问题所需的灵活性。

## 适用对象

本手册提供关于S7-200微型PLC安装和编程的信息，为工程师、程序员、安装人员和具有可编程逻辑控制器常识的电工而设计。

## 本手册范围

手册中包含的信息主要适用于下列产品：

- S7-200 CPU型号：CPU 221、CPU 222、CPU 224、CPU 226和CPU 226XM
- S7-200 EM 22x扩展模块
- STEP 7-Micro/WIN版本3.2，用于S7-200的32位编程软件包
- STEP 7-Micro/WIN指令库和用于TP070的TPesigner，版本1.0，为使用S7-200与其它组件（例如TP070 Touch Panel、Modbus 或MicroMaster驱动）的用户提供的一套软件工具

## 标准适应性

SIMATIC S7-200系列符合下列标准：

- 欧共体（CE）低电压指令73/23/EEC  
EN61131-2：可编程控制器 - 设备要求
- 欧共体（CE）EMC指令89/336/EEC  
电磁辐射标准  
EN 61000-6-3：住宅、商业和轻工业  
EN 61000-6-4：工业环境  
电磁抗扰标准  
EN 61000-6-2：工业环境
- 保险商实验室  
UL 508列出的（工业控制设备）登记号E75310
- 加拿大标准协会：CSA C22.2号142（过程控制设备）
- 工厂共同研究：FM等级I，部分2，组A、B、C和D危险位置，T4A和等级I，2区，IIC，T4

参考附录A以获取适应性信息。



### 提示

SIMATIC S7-200系列符合CSA标准。

cULus标志指示S7-200已经由保险商实验室（UL）检查并鉴定为标准UL 508和CSA 22.2 No. 142。

## 海事认证

S7-200产品定期提交特定机构进行关于特定市场和应用的认证。本表格标识授予S7-200产品认证的机构及其证书编号。并不是本手册中所有S7-200产品都得到这些特定机构的认证。如果需要关于按部件号准确认证的最新列表的附加信息，请咨询本地西门子代表。

代理	证书号
英国劳氏船级社 (LRS)	99 / 20018 (E1)
美国船级社 (ABS)	01-HG20020-PDA
德国船级社 (GL)	12 045 - 98 HH
挪威船级社 (DNV)	A-8071
法国船级社 (BV)	09051 / A2 BV
日本海事协会 (NK)	A-534

## 如何使用本手册

如果您第一次使用（新手）S7-200微型PLC，应该阅读整本S7-200 可编程控制器系统手册。如果您是经验丰富的用户，请参考目录表或索引来查找指定的信息。

S7-200 可编程控制器系统手册按照下列主题组织：

- 第1章（产品概述）提供了微型PLC产品S7-200系列的一些特性的概述。
- 第2章（使用入门）提供了创建和下载实例控制程序到S7-200的教程。
- 第3章（安装S7-200）提供了安装S7-200 CPU模块和扩展I/O模块的尺寸和基本准则。
- 第4章（PLC原则）提供了关于操作S7-200的信息。
- 第5章（编程原则、约定和特征）提供了关于STEP 7-Micro/WIN的特征、程序编辑器和指令类型（IEC 1131或SIMATIC）、S7-200数据类型和程序创建指南的信息。
- 第6章（S7-200指令集）提供了S7-200所支持的编程指令的说明和实例。
- 第7章（通过网络进行通讯）提供了建立S7-200所支持的不同网络配置的信息。
- 第8章（硬件故障检测指南和软件调试工具）提供了检测S7-200硬件故障和帮助用户调试程序的STEP 7-Micro/WIN特征的信息。
- 第9章（为定位模块创建程序）提供了用于为EM 253定位模块创建程序的说明和向导的信息。
- 第10章（为调制解调器模块创建程序）提供了用于为EM 241调制解调器模块创建程序的指令和向导的信息。
- 第11章（使用USS协议库来控制MicroMaster驱动器）提供了用于为MicroMaster驱动器创建控制程序的指令的信息。还提供了关于如何配置MicroMaster 3和MicroMaster 4驱动器的信息。
- 第12章（使用Modbus协议库）提供了用于创建使用Modbus协议进行通讯的程序的指令的信息。
- 附录A（技术规范）提供了关于S7-200硬件的技术信息和数据表。

其它附录提供附加的参考信息，例如错误代码说明、特殊内存（SM）区域说明、订购S7-200的部件号和语句表指令执行时间。

## 附加信息和帮助

### 关于S7-200和STEP 7-Micro/WIN的信息

除了本手册，STEP 7-Micro/WIN还提供了S7-200编程入门的详尽的联机帮助。购买STEP 7-Micro/WIN软件时包含了免费的文档光盘。该光盘中包含应用程序提示、本手册的电子版本和其它信息。

#### 联机帮助

只需按键即可获得帮助！按下F1访问STEP 7-Micro/WIN的详尽的联机帮助。联机帮助包含S7-200编程入门的有用信息，以及许多其它主题。

#### 电子手册

文档光盘上包含有本S7-200系统手册的电子版本。可以将电子手册安装到计算机上，这样，使用STEP 7-Micro/WIN软件时就可以方便地访问手册中的信息。

#### 编程提示

文档光盘包含编程提示和一些实例程序的应用实例。检查或修改这些实例可以为您自己的应用找到有效或创新的解决方案提供帮助。也可以在S7-200网站上找到编程提示的最新版本。

Internet: [www.siemens.com/S7-200](http://www.siemens.com/S7-200)

关于西门子产品和服务、技术支持、常见问题解答（FAQ）、产品更新或应用提示的附加信息，请参见下列Internet地址：

- [www.ad.siemens.de](http://www.ad.siemens.de) *常规西门子信息*

该西门子自动化和驱动部门网站包含关于SIMATIC产品线和其它西门子产品的信息。

- [www.siemens.com/S7-200](http://www.siemens.com/S7-200) *S7-200产品信息*

S7-200网站包含常见问题解答（FAQ）、编程提示（应用实例和实例程序）、关于新发布产品和产品更新或下载的信息。

## 技术支持和购买S7-200产品

### 本地西门子销售部或经销商

要获取技术支持、S7-200产品培训或订购S7-200产品，请联系西门子经销商或销售部。因为您的销售代表经过技术方面的培训，并且具有关于您的操作、处理和行业的最专业知识以及您所使用的单个西门子产品的知识，他们可以为您可能遇到的问题提供最快最有效的解答。

### 技术服务

S7-200技术服务中心经过严格培训的人员也可以帮助您解决可能遇到的问题。您可以随时电话联系：

- 对于美国国内的电话*  
当地时间：星期一至星期五东部时间0800至1900  
电话：+1 800 241-4453  
传真：+1 (0) 770 740-3699  
电子信箱：drives.support@sea.siemens.com
- 对于美国以外美洲的电话*  
当地时间：星期一至星期五东部时间0800至1900  
电话：+1 (0) 770 740-3505  
传真：+1 (0) 770 740-3699  
电子信箱：drives.support@sea.siemens.com
- 对于欧洲和非洲的电话*  
当地时间（纽伦堡）：星期一至星期五0700至1700  
电话：+49 (0) 180 5050-222  
传真：+49 (0) 180 5050-223  
电子信箱：techsupport@ ad.siemens.de
- 对于亚洲和澳大利亚的电话*  
当地时间（新加坡）：星期一至星期五0830至1730  
电话：+65 (0) 740-7000  
传真：+65 (0) 740-7001  
电子信箱：drives.support@sae.siemens.com.sg

# 目录

<b>1</b>	<b>产品概述</b> .....	<b>1</b>
	S7-200 CPU .....	2
	S7-200扩充模块 .....	3
	STEP 7-Micro/WIN程序包 .....	3
	通讯选项 .....	4
	显示面板 .....	4
<b>2</b>	<b>入门指南</b> .....	<b>5</b>
	连接S7-200 CPU .....	6
	创建采样程序 .....	9
	下载采样程序 .....	12
	将S7-200置于RUN（运行）模式 .....	12
<b>3</b>	<b>安装S7-200</b> .....	<b>13</b>
	安装S7-200设备指南 .....	14
	安装和删除S7-200模块 .....	15
	接地和布线指南 .....	18
<b>4</b>	<b>PLC原理</b> .....	<b>21</b>
	了解S7-200如何执行控制逻辑 .....	22
	存取S7-200的数据 .....	24
	了解S7-200如何保存和恢复数据 .....	34
	将程序存储到内存磁带上 .....	36
	选择S7-200 CPU的操作模式 .....	37
	使用程序将V内存保存到EEPROM .....	38
	S7-200的特征 .....	39
<b>5</b>	<b>编程原则、约定和特征</b> .....	<b>47</b>
	Micro PLC系统设计指南 .....	48
	程序的基本单元 .....	49
	使用STEP 7-Micro/WIN创建程序 .....	51
	在SIMATIC和IEC 1131-3指令集之间进行选择 .....	53
	了解程序编辑器所使用的约定 .....	54
	使用向导来帮助创建控制程序 .....	56
	处理S7-200中的错误 .....	56
	在数据块编辑程序中分配地址和初始值 .....	58
	使用符号表对变量进行符号编址 .....	58
	使用局部变量 .....	59
	使用状态图监控程序 .....	59
	创建指令库 .....	60
	程序调试的特性 .....	60

<b>6</b>	<b>S7-200指令集</b>	<b>61</b>
	用于描述指令的约定	63
	S7-200内存范围和特征	64
	位逻辑指令	66
	接点	66
	线圈	69
	逻辑堆栈指令	71
	设置和重设主要双稳态指令	73
	时钟指令	74
	通讯指令	75
	网络读取和网络写入指令	75
	传输和接收指令（空闲端口）	80
	获取端口地址和设置端口地址指令	89
	比较指令	90
	比较数字值	90
	比较字符串	92
	转换指令	93
	标准转换指令	93
	ASCII转换指令	97
	字符串转换指令	101
	编码和解码指令	106
	计数器指令	107
	SIMATIC计数器指令	107
	IEC计数器指令	110
	高速计数器指令	112
	脉冲输出指令	126
	运算指令	141
	加、减、乘、除指令	141
	乘以整数到双整数，除以整数带余数	143
	数字函数指令	144
	递增和递减指令	145
	比例 / 积分 / 微分（PID）环路指令	146
	中断指令	156
	逻辑操作指令	163
	反转指令	163
	AND（与）、OR（或）和Exclusive OR（异或）指令	164
	移动指令	166
	移动字节、字、双字或实数	166
	立即移动字节（读和写）	167
	成块移动指令	168
	程序控制指令	169
	有条件结束	169
	停止	169
	监视程序重设	169
	For-Next循环指令	171
	跳转指令	173
	顺序控制继电器（SCR）指令	174



移位和旋转指令 .....	180
向右移位和向左移位指令 .....	180
向右旋转和向左旋转指令 .....	180
移位寄存器指令 .....	182
交换字节指令 .....	184
字符串指令 .....	185
表格指令 .....	190
添加到表格 .....	190
先入先出和后人先出 .....	191
内存填充 .....	193
表格查找 .....	194
计时器指令 .....	197
SIMATIC计时器指令 .....	197
IEC计时器指令 .....	202
子例行程序指令 .....	204
<b>7 通过网络进行通讯 .....</b>	<b>209</b>
了解S7-200网络通讯基础知识 .....	210
选择网络通讯协议 .....	214
安装和删除通讯接口 .....	220
构建网络 .....	222
创建具有自由端口模式的自定义协议 .....	227
在网络中使用调制解调器和STEP 7-Micro/WIN .....	229
高级主题 .....	235
配置用于远程操作的RS-232/PPI多台主设备电缆 .....	241
<b>8 硬件故障诊断指南和软件调试工具 .....</b>	<b>245</b>
程序调试特征 .....	246
显示程序状态 .....	248
使用状态图监控和修改S7-200中的数据 .....	249
强制指定值 .....	250
以指定的扫描次数运行程序 .....	250
硬件故障诊断指南 .....	251
<b>9 创建位控模块程序 .....</b>	<b>253</b>
位控模块的特性 .....	254
配置位控模块 .....	256
位置控制向导创建的位控指令 .....	267
位控模块的范例程序 .....	279
使用EM 253控制面板监控位控模块 .....	284
位控模块和位控指令的出错代码 .....	286
高级主题 .....	288

<b>10</b>	<b>创建调制解调器模块程序</b> .....	<b>297</b>
	调制解调器模块的性能 .....	298
	使用调制解调器扩充向导来配置调制解调器模块 .....	304
	调制解调器指令和限制概述 .....	308
	调制解调器模块的指令 .....	309
	调制解调器模块的采样程序 .....	313
	支持智能模块的S7-200 CPU .....	313
	调制解调器模块的特殊内存位置 .....	314
	高级主题 .....	316
	讯息传送电话号码格式 .....	318
	文本讯息格式 .....	319
	CPU数据传送讯息格式 .....	320
<b>11</b>	<b>使用USS协议库控制MicroMaster驱动器</b> .....	<b>321</b>
	使用USS协议的要求 .....	322
	计算与驱动器通讯所需要的时间 .....	323
	使用USS指令 .....	324
	USS协议的指令 .....	325
	USS协议的采样程序 .....	332
	USS执行错误代码 .....	333
	连接和设置MicroMaster系列3驱动器 .....	334
	连接和设置MicroMaster系列4驱动器 .....	337
<b>12</b>	<b>使用Modbus协议库</b> .....	<b>339</b>
	使用Modbus协议的要求 .....	340
	Modbus协议的初始化和执行时间 .....	340
	Modbus编址 .....	341
	使用Modbus从属协议指令 .....	342
	Modbus从属协议指令 .....	343
<b>A</b>	<b>技术规范</b> .....	<b>347</b>
	常规技术规范 .....	348
	CPU规范 .....	351
	数字扩充模块规范 .....	357
	模拟扩充模块规范 .....	363
	热电偶和RTD扩充模块规范 .....	373
	EM 277 PROFIBUS-DP模块规范 .....	385
	EM 241调制解调器模块规范 .....	397
	EM 253定位模块规范 .....	399
	(CP 243-1) 以太网模块规范 .....	405
	(CP 243-1 IT) 互联网模块规范 .....	407
	(CP 243-2) AS接口模块规范 .....	410
	供选用磁带 .....	412
	I/O扩展电缆 .....	412
	RS-232/PPI多台主设备电缆和USB/PPI多台主设备电缆 .....	413
	输入模拟程序 .....	417

<b>B</b>	<b>计算功率分配</b> .....	<b>419</b>
<b>C</b>	<b>错误代码</b> .....	<b>423</b>
	严重错误代码和讯息 .....	424
	运行系统编程问题 .....	425
	违反编译规则 .....	426
<b>D</b>	<b>特殊内存 (SM) 位</b> .....	<b>427</b>
	SMB0: 状态位 .....	428
	SMB1: 状态位 .....	428
	SMB2: 自由端口接收字符 .....	429
	SMB3: 自由端口奇偶校验错误 .....	429
	SMB4: 队列溢出 .....	429
	SMB5: I/O状态 .....	430
	SMB6: CPU标识寄存器 .....	430
	SMB7: 保留 .....	430
	SMB8到SMB21: I/O模块标识号和出错寄存器 .....	431
	SMW22到SMW26: 扫描时间 .....	432
	SMB28和SMB29: 模拟调整 .....	432
	SMB30和SMB130: 自由端口控制寄存器 .....	432
	SMB31和SMW32: 永久性内存 (EEPROM) 写控制 .....	433
	SMB34和SMB35: 用于定时中断的时间间隔寄存器 .....	433
	SMB36到SMB65: HSC0、HSC1和HSC2寄存器 .....	433
	SMB66到SMB85: PTO/PWM寄存器 .....	435
	SMB86到SMB94和SMB186到SMB194: 接收讯息控制 .....	436
	SMW98: 扩展I/O总线出错 .....	437
	SMB130: 自由端口控制寄存器 (参见SMB30) .....	437
	SMB131到SMB165: HSC3、HSC4和HSC5寄存器 .....	437
	SMB166到SMB185: PTO0、PTO1配置文件定义表 .....	438
	SMB186到SMB194: 接收讯息控制 (参见SMB86到SMB94) .....	438
	SMB200到SMB549: 智能模块状态 .....	439
<b>E</b>	<b>S7-200订购号</b> .....	<b>441</b>
<b>F</b>	<b>STL指令的执行时间</b> .....	<b>445</b>
<b>G</b>	<b>S7-200快速参考手册信息</b> .....	<b>451</b>

索引



# 产品概述

微型可编程逻辑控制器（微型PLC）的S7-200系列可以控制多种设备以支持您的自动化需求。

S7-200如同由用户程序控制那样监控输入并改变输出，可以包含布尔型逻辑、计算、定时、复杂数学运算和与其它智能设备通讯。紧凑的设计、灵活的配置和功能强大的指令集组合在一起，使得S7-200成为控制多种应用程序的理想解决方案。

## 本章内容

S7-200 CPU .....	2
S7-200扩充模块 .....	3
STEP 7-Micro/WIN程序包 .....	3
通讯选项 .....	4
显示面板 .....	4

## S7-200 CPU

S7-200 CPU在紧凑的外壳中组合了微处理器、集成的电源、输入电路和输出电路，以创建功能强大的微型 PLC。参见图1-1。下载程序后，S7-200包含应用程序中需要用来监控和控制输入和输出设备的逻辑。

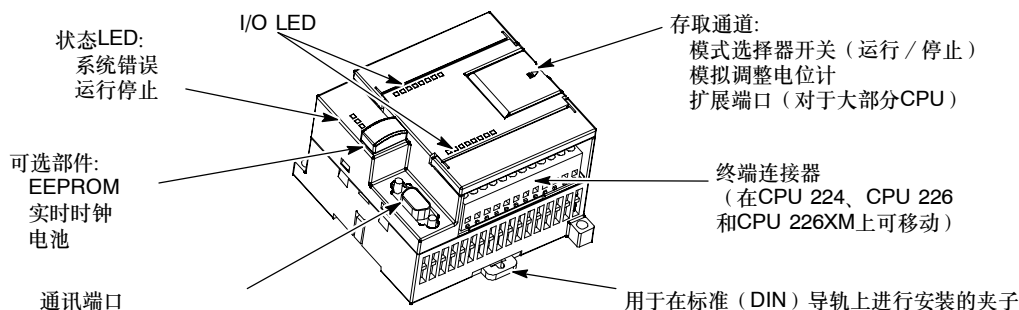


图1-1 S7-200 Micro PLC

西门子提供不同的S7-200 CPU型号，具有多种特征和性能，可以帮助您为不同的应用创建有效的解决方案。表1-1简要地比较了CPU的一些性能。要获取关于指定CPU的详细信息，参见附录A。

表1-1 S7-200 CPU型号的比较

特征	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
物理尺寸（毫米）	90 x 80 x 62	90 x 80 x 62	120.5 x 80 x 62	190 x 80 x 62	190 x 80 x 62
程序内存	4096字节	4096字节	8192字节	8192字节	16384字节
数据内存	2048字节	2048字节	5120字节	5120字节	10240字节
内存备份	50小时典型的	50小时典型的	190小时典型的	190小时典型的	190小时典型的
本地板载I/O	6输入 / 4输出	8输入 / 6输出	14输入 / 10输出	24输入 / 16输出	24输入 / 16输出
扩充模块	0个模块 <sup>1</sup>	2个模块 <sup>1</sup>	7个模块 <sup>1</sup>	7个模块 <sup>1</sup>	7个模块 <sup>1</sup>
高速计数器					
单相	30 kHz时为4	30 kHz时为4	30 kHz时为6	30 kHz时为6	30 kHz时为6
双相	20 kHz时为2	20 kHz时为2	20 kHz时为4	20 kHz时为4	20 kHz时为4
脉冲输出（DC）	20 kHz时为2	20 kHz时为2	20 kHz时为2	20 kHz时为2	20 kHz时为2
模拟调整	1	1	2	2	2
实时时钟	部件	部件	内置	内置	内置
通讯端口	1 RS-485	1 RS-485	1 RS-485	2 RS-485	2 RS-485
浮点数学	是				
数字I/O图像大小	256（128个输入，128个输出）				
布尔型执行速度	0.37微秒 / 指令				

<sup>1</sup> 必须对电源进行预算，以确定S7-200 CPU可以为配置提供多少电源（或电流）。如果超出了CPU电源预算，可能无法连接最大数目的模块。参见附录A以了解CPU和扩充模块的电源要求，参见附录B以计算电源预算。

## S7-200扩充模块

为了更好地解决您的应用需要，S7-200系列包含许多种扩充模块。可以使用这些扩充模块将功能添加到S7-200 CPU。表1-2提供了当前可用的扩充模块的列表。至于有关指定模块的详细信息，参见附录A。

表1-2 S7-200扩充模块

扩充模块	类型			
离散模块	输入	8 × DC输入	8 × AC输入	16 × DC输入
	输出	4 × DC 8 × DC输出	4 × 继电器 8 × AC输出	8 × 继电器
	组合	4 × DC输入 / 4 × DC输出 4 × DC输入 / 4 × 继电器	8 × DC输入 / 8 × DC输出 8 × DC输入 / 8 × 继电器	16 × DC输入 / 16 × DC输出 16 × DC输入 / 16 × 继电器
	模拟模块	4 × 模拟输入	4 × 热电偶输入	2 × RTD输入
智能模块	位置		调制解调器	PROFIBUSP
	以太网		互联网	
	其它模块	自动化系统接口		

## STEP 7-Micro/WIN程序包

STEP 7-Micro/WIN程序包提供用户友好的环境来开发、编辑和监控控制应用程序所需的逻辑。STEP 7-Micro/WIN提供了三个程序编辑器，以便方便有效地为应用开发控制程序。STEP 7-Micro/WIN提供了丰富的联机帮助系统和文档光盘（包含该手册的电子版本、应用提示和其它有用信息）来帮助您找到所需的信息。

### 计算机要求

STEP 7-Micro/WIN在个人计算机或西门子编程设备（例如PG 760）上运行。您的计算机或编程设备应该满足下列最低要求：

- 操作系统：  
Windows 95、Windows 98、Windows 2000、Windows Me（千禧版）、Windows NT 4.0（或更新版本）、Windows XP Professional
- 至少100MB空闲的硬盘空间
- 鼠标（建议）

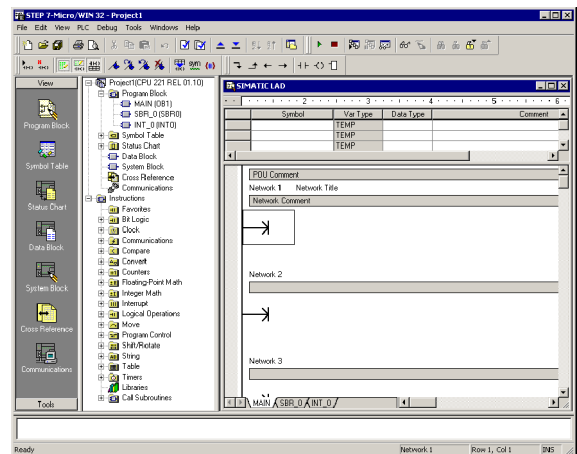


图1-2 STEP 7-Micro/WIN

## 安装STEP 7-Micro/WIN

将STEP 7-Micro/WIN光盘插入计算机的光盘驱动器。安装向导自动启动，在整个安装过程中提示用户。参考自述文件以获取关于安装STEP 7-Micro/WIN的更多信息。



### 提示

要在Windows NT、Windows 2000或Windows XP Professional操作系统上安装STEP7-Micro/WIN，必须以管理员权限登录。

## 通讯选项

西门子提供了两个编程选项用于将您的计算机连接到S7-200：使用PPI多主站电缆直接连接或使用MPI电缆的通讯处理器（CP）卡。

PPI多主站编程电缆是将计算机连接到S7-200的最常用最经济的方法。该电缆将S7-200的通讯端口连接到计算机的串行通讯端口。PPI多主站编程电缆还可以用于将其它通讯设备连接到S7-200。

## 显示面板

### TD 200文本显示单元

TD 200是可以连接到S7-200的2行、20字符的文本显示设备。使用TD 200向导，可以容易地对S7-200进行编程，以显示应用程序所带的文字讯息和其它数据。

通过允许用户查看、监控和改变应用程序中的过程变量，TD 200提供了与应用程序的低成本接口。

独立的手册说明了TD 200的详细功能和规范。

STEP 7-MicroWIN中的TD 200配置向导帮助快速方便地配置TD 200。要启动TD 200向导，选择工具（Tools）> TD 200向导（TD 200 Wizard）菜单命令。



TD 200

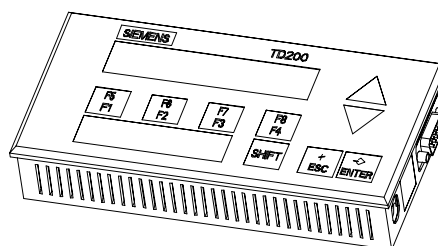


图1-3 TD 200文本显示单元

### TP070触摸面板显示

TP070是可以连接到S7-200的触摸面板显示设备。该触摸面板提供了自定义操作员界面的方法。

TP070可以通过用户友好的触摸面板显示自定义图形、滚动条、应用程序变量、自定义用户按钮等等。

用于TP070的可选的TP-Designer，版本1.0光盘提供了对TP070进行编程所需的TP Designer软件。

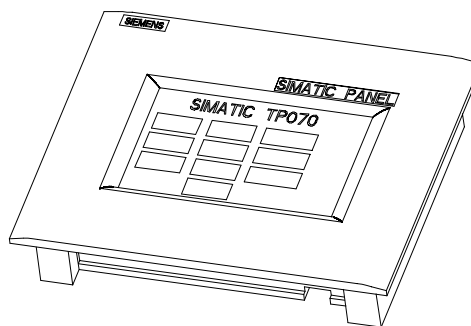


图1-4 TP070触摸面板单元



# 使用入门



STEP 7-Micro/WIN可以方便地对S7-200编程。使用简单的实例，只需几个简短的步骤，就可以学会如何连接、编程和运行S7-200。

本实例中，需要PPI多台主设备电缆、S7-200 CPU和运行STEP7-Micro/WIN编程软件的编程设备。

## 本章内容

连接S7-200 CPU .....	6
创建采样程序 .....	9
下载采样程序 .....	12
将S7-200置于RUN（运行）模式 .....	12

## 连接S7-200 CPU

连接S7-200十分简单。对于本实例，仅需将S7-200 CPU通电，然后在编程设备和S7-200 CPU之间连接通讯电缆。

### 将S7-200 CPU通电

第一步是将S7-200连接到电源。图2-1显示S7-200 CPU的DC或AC模块的导线连接。

安装或删除任何电气设备之前，确保已经断开该设备的电源。始终遵守合适的安全预防，尝试安装或删除S7-200之前，确保已断开S7-200的电源。



#### 警告

带电情况下试图对S7-200或相关设备进行安装或布线可能会导致电击或设备运行故障。在安装或拆除过程中，如果断开S7-200和相关设备的所有电源失败，可能会导致死亡或人员受伤，并且/或者损坏设备。

始终遵守合适的安全预防措施，尝试安装或删除S7-200或相关设备之前，请确保已断开S7-200的电源。

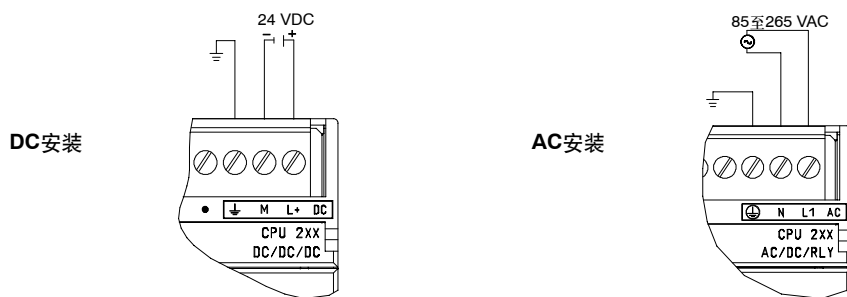


图2-1 将S7-200 CPU通电



#### 提示

本手册中的实例使用RS-232/PPI多台主设备电缆。RS-232/PPI多台主设备电缆更换以前的PC/PPI电缆。也可以使用USB/PPI多台主设备电缆。参考附录E获得订购号。

## 连接RS-232/PPI多台主设备电缆

图2-2显示将S7-200连接到编程设备的RS-232/PPI多台主设备电缆。要连接电缆:

1. 将RS-232/PPI多台主设备电缆的RS-232连接器（标记为“PC”）连接到编程设备的通讯端口。（对于本实例，连接到COM 1。）
2. 将RS-232/PPI多台主设备电缆的RS-485连接器（标记为“PPI”）连接到S7-200的端口0或端口1。
3. 确保RS-232/PPI多台主设备电缆的DIP开关如图2-2中显示的那样设置。

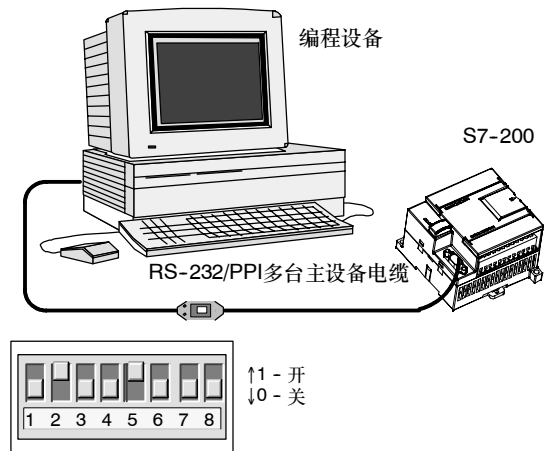


图2-2 连接RS-232/PPI多台主设备电缆

## 启动STEP 7-Micro/WIN

单击STEP 7-Micro/WIN图标以打开新项目。图2-3显示新项目。

注意浏览条。可以使用浏览条上的图标打开STEP 7-Micro/WIN项目的单元。

单击浏览条中的“通讯”图标以显示“通讯”对话框。使用该对话框为STEP 7-Micro/WIN建立通讯。

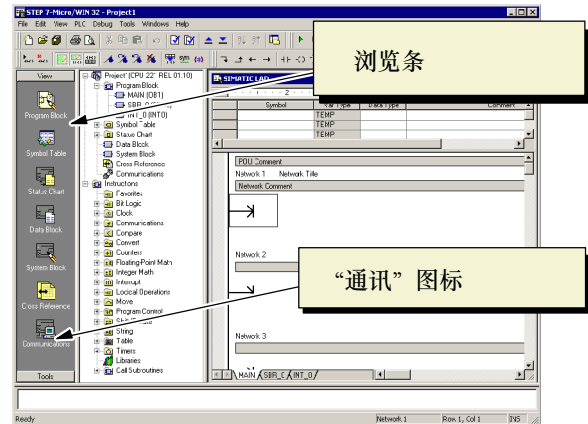


图2-3 新STEP 7-Micro/WIN项目

## 核实STEP 7-Micro/WIN的通讯参数

实例项目为STEP 7-Micro/WIN和RS-232/PPI多台主设备电缆使用缺省设置。要核实这些设置：

1. 核实“通讯”对话框中PC/PPI电缆地址设置为0。
2. 核实网络接口参数设置为PC/PPI电缆（COM1）。
3. 核实传输率设置为9.6 kbps。

如果需要改变通讯参数设置，参见第7章。

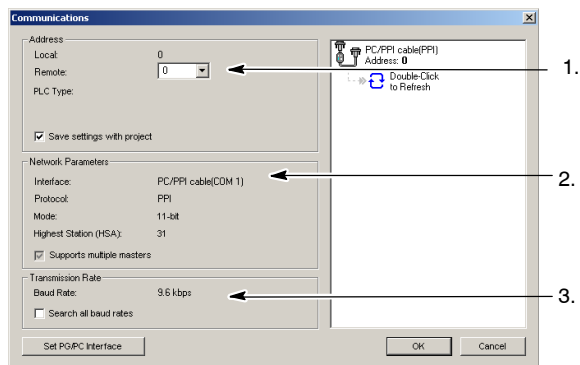


图2-4 核实通讯参数

## 建立与S7-200的通讯

使用“通讯”对话框来连接S7-200 CPU：

1. 双击“通讯”对话框中的“刷新”图标。STEP 7-Micro/WIN搜索S7-200站并为连接的S7-200站显示CPU图标。
2. 选择S7-200并单击“确定”。

如果STEP 7-Micro/WIN未找到S7-200 CPU，检查通讯参数设置并重复这些步骤。

建立与S7-200的通讯后，可以创建和下载实例程序。

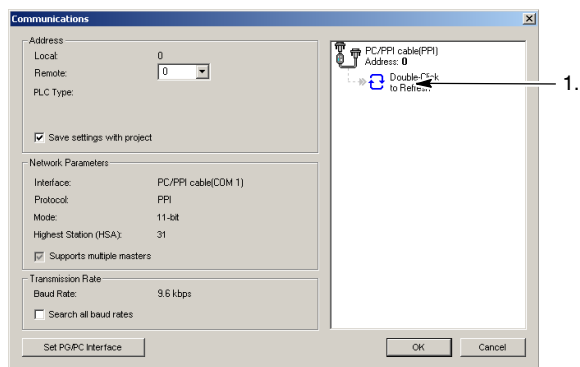


图2-5 建立与S7-200的通讯

## 创建范例程序

输入该控制程序实例有助于了解使用STEP 7-Micro/WIN十分简单。该程序在三个程序段中使用六个指令来创建十分简单且自己重设的自启动计时器。

对于本实例，使用梯形图（LAD）编辑器为程序输入指令。下列实例在LAD和语句表（STL）中显示了完整的程序。STL程序中的程序段注释说明每个程序段的逻辑。计时图显示程序的操作。

实例: STEP 7-Micro/WIN使用入门的范例程序

	<pre> Network 1 //10毫秒计时器T33在（100 x 10毫秒 = 1秒）后超时 //M0.0时钟脉冲太快，无法使用状态视图监控。 LDN M0.0 TON T33, +100  Network 2 //使用状态视图可见的比率时， //可以进行比较。在（40 x 10毫秒 = 0.4秒）后接通Q0.0， //产生40%关/60%开的波形。 LDW &gt;= T33, +40 = Q0.0  Network 3 //T33（位）时钟脉冲太快，无法监控状态视图。 //（100 x 10毫秒 = 1秒）后， //通过M0.0重设计时器。 LD T33 = M0.0 </pre>
--	---

计时图

## 打开程序编辑器

单击“程序块”图标打开程序编辑器。参见图2-6。

注意指令树和程序编辑器。通过将指令从指令树拖放到程序段，使用指令树将LAD指令插入程序编辑器的程序段中。

工具栏图标提供菜单命令的快捷键。

输入和保存程序后，可以将程序下载到S7-200。

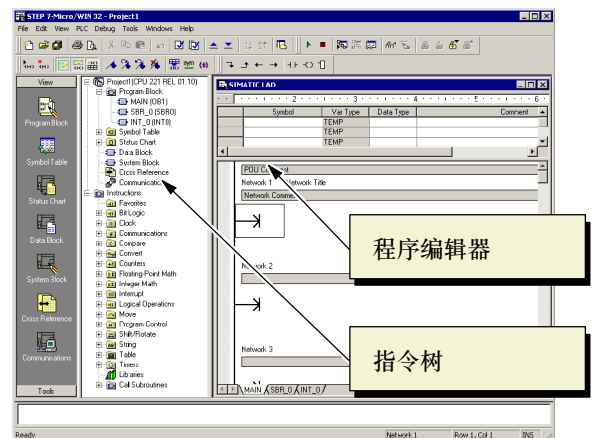


图2-6 STEP 7-Micro/WIN窗口

## 输入程序段1：启动计时器

M0.0断开时（0），该接点接通并提供功率流来启动计时器。要为M0.0输入接点：

1. 双击“位逻辑”图标或单击加号（+）显示位逻辑指令。
2. 选择“常闭”接点。
3. 按住鼠标左键并将接点拖动到第一个程序段。
4. 单击接点上方的“???”并输入下列地址：M0.0
5. 按下回车键，以输入接点的地址。

为T33输入计时器指令：

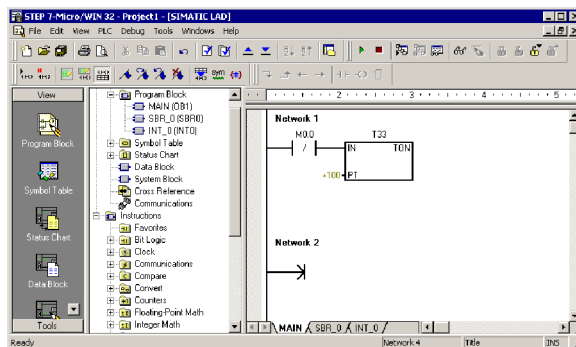


图2-7 段1

1. 双击计时器图标来显示计时器指令。
2. 选择TON（打开延迟计时器）。
3. 按住鼠标左键并将计时器拖动到第一个程序段。
4. 单击计时器方框上方的“???”并输入下列计时器编号：T33
5. 按下回车键以输入计时器编号并将焦点移动到预置时间（PT）参数。
6. 为预置时间输入下列值：100
7. 按下回车键以输入数值。

## 输入程序段2：接通输出

T33的计时器数值大于或等于40（40乘以10毫秒，也就是0.4秒）时，接点提供功率流来接通S7-200的输出Q0.0。要输入“比较”指令：

1. 双击“比较”图标以显示比较指令。选择>=I指令（大于或等于整数）。
2. 按住鼠标左键并将比较指令拖动到第二个程序段。
3. 单击接点上方的“???”并输入计时器数值的地址：T33
4. 按下回车键来输入计时器编号并将焦点移动到要与计时器数值比较的其它值。
5. 输入要与计时器数值比较的下列数值：40
6. 按下回车键以输入数值。

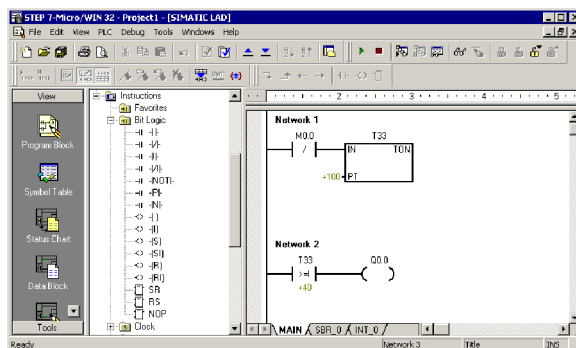


图2-8 程序段2

要输入指令以接通输出Q0.0:

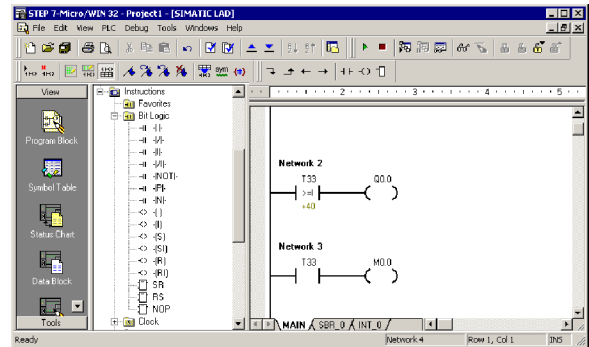
1. 双击“位逻辑”图标来显示位逻辑指令并选择输出线圈。
2. 按住鼠标左键并将线圈拖动到第二个程序段。
3. 单击线圈上方的“???”并输入下列地址：Q0.0
4. 按下回车键，以输入线圈的地址。

## 输入程序段3: 重设计时器

计时器达到预设值（100）并且接通计时器位时，T33接点接通。来自该接点的功率流接通到M0.0内存位置。因为计时器由M0.0的常闭接头启用，将M0.0的状态从关（0）改变为开（1）重设计时器。

要为T33计时器位输入接点:

1. 从位逻辑指令选择“常开”接点。
2. 按住鼠标左键并将接点拖动到第三个程序段。
3. 单击接点上方的“???”并输入计时器位的地址: T33
4. 按下回车键，以输入接点的地址。



要输入线圈以接通M0.0:

1. 从位逻辑指令选择输出线圈。
2. 按住鼠标左键并将输出线圈拖动到第三个程序段。
3. 双击线圈上方的“???”并输入下列地址: M0.0
4. 按下回车键，以输入线圈的地址。

## 保存实例项目

输入指令的三个程序段后，程序输入完成。保存程序时，创建包含S7-200 CPU类型和其它参数的项目。要保存项目:

1. 从菜单栏中选择文件（File）> 另存为（Save As）菜单命令。
2. 在“另存为”对话框中输入项目的名称。
3. 单击“确定”，保存项目。

保存项目后，可以将程序下载到S7-200。

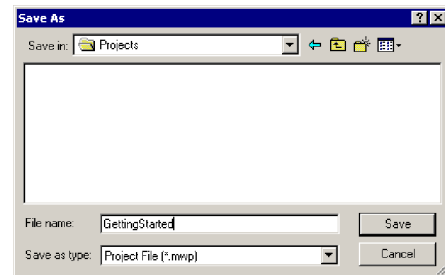


图2-9 保存实例程序

## 下载范例程序



### 提示

每个STEP 7-Micro/WIN项目都与CPU类型（CPU 221、CPU 222、CPU 224、CPU 226或CPU 226XM）相关。如果项目类型与所连接的CPU不匹配，STEP7-Micro/WIN指示不匹配并提示要采取措施。如果发生这种情况，在本实例中选择“继续下载”。

1. 单击工具栏上的“下载”图标或选择文件（File）> 下载（Download）菜单命令来下载程序。参见图2-10。
2. 单击“确定”将程序单元下载到S7-200。

如果S7-200在RUN（运行）模式中，出现对话框提示将S7-200置于STOP（停止）模式。单击“是”将S7-200置于STOP（停止）模式。

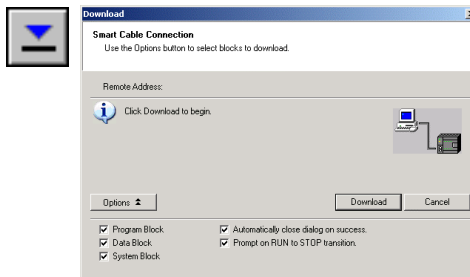


图2-10 下载程序

## 将S7-200置于RUN（运行）模式中

对于STEP 7-Micro/WIN，要将S7-200 CPU置于RUN（运行）模式，S7-200的模式开关必须设置为TERM或RUN。将S7-200置于RUN（运行）模式时，S7-200执行程序：

1. 单击工具栏上的“运行”图标或选择PLC > 运行（RUN）菜单命令。
2. 单击“确定”改变S7-200的操作模式。

S7-200转到RUN（运行）模式时，Q0.0的输出LED在S7-200执行程序时接通和断开。



图2-11 将S7-200置于RUN（运行）模式中

恭喜！您已经完成了第一个S7-200程序。

可以选择调试（Debug）> 程序状态（Program Status）菜单命令来监控程序。STEP 7-Micro/WIN显示指令的数值。要停止程序，通过单击“停止”图标或选择PLC > 停止（STOP）菜单命令将S7-200置于STOP（停止）模式。



## 安装S7-200

S7-200设备设计成易于安装。可以使用安装孔将模块装到面板上，或者可以使用内置的夹片将模块安装到标准（DIN）横杆上。S7-200尺寸较小，可以有效地利用空间。

本章提供了S7-200系统的安装和布线指南。

### 本章内容

安装S7-200设备指南 .....	14
安装和删除S7-200模块 .....	15
接地和布线指南 .....	18

## 安装S7-200设备指南

可以将S7-200安装在面板或标准横杆上，S7-200的方向可以水平或垂直。

### 避免S7-200设备受热、高电压和电噪声

布置系统设备的常规原则是，始终将产生高压和高电噪声的设备与低压、逻辑类型的设备（例如S7-200）分开。

在配置面板内部S7-200的布局时，要考虑到发热设备，将电气类型设备放在机柜的较冷区域中。在高温环境中操作任何电气设备都将缩短故障时间。

还需考虑面板中设备布线的路线。避免使用AC电力配线和高能量、快速转换的DC配线将低压导线和通讯电缆放在同一个托盘中。

### 为冷却和配线提供足够的空隙

S7-200设备设计成自然对流冷却。要正确冷却，在设备上方和下方必须提供至少为25毫米的间隙。而且要允许至少75毫米的深度。



#### 提示

对于垂直安装，允许的最高环境温度降低10°C。将S7-200 CPU安装在任何扩展模块下方。

规划S7-200系统的布局时，为配线和通讯电缆连接留出足够的间隙。为了配置S7-200系统布局时有附加的灵活性，使用I/O扩展电缆。

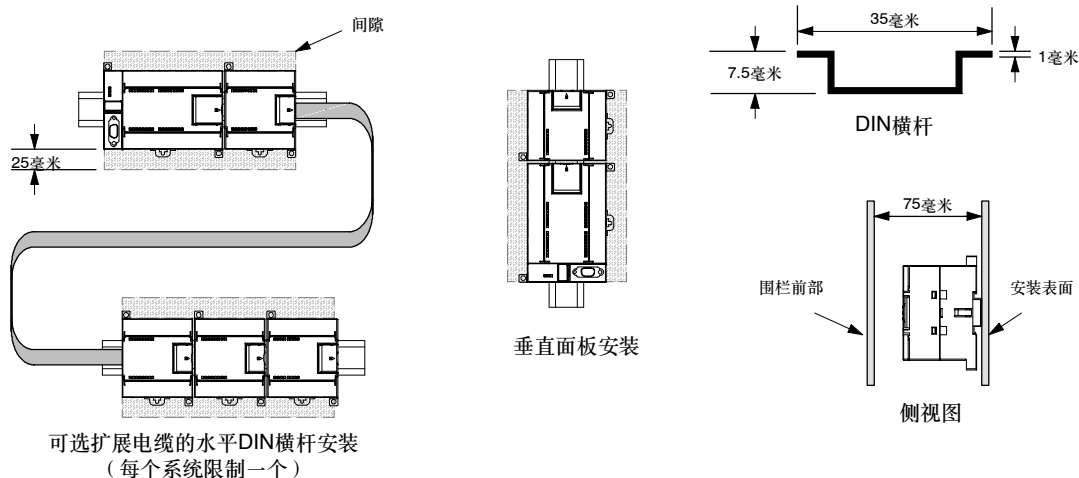


图3-1 安装方法、方向和间隙

## 电源预算

所有S7-200 CPU都有内部电源，为CPU、扩充模块和其它24 VDC用户电源供电。

S7-200 CPU提供5 VDC逻辑电源，用于系统中的任何扩展。密切注意系统配置，以确保CPU可以提供所选扩充模块所需的5V电源。如果配置需要的电源大于CPU可以提供的电源，必须删除某些模块或选择可以提供更大电源的CPU。参考附录A，了解关于S7-200 CPU所提供的5 VDC逻辑预算和扩充模块5 VDC电源要求的信息。使用附录B作为确定CPU为您的配置所能提供的电源（或电流）的指南。

所有S7-200 CPU还提供24 VDC传感器电源，可以为输入点、扩充模块上的继电器线圈电源或其它需要提供24 VDC。如果电源要求超出传感器电源预算，则必须在系统中添加外部24 VDC电源。参考附录A，了解特定S7-200 CPU的24 VDC传感器电源预算。

如果需要外部24 VDC电源，确保电源未并行连接到S7-200 CPU的传感器电源。为了提高电噪声保护，建议连接不同电源的公共端（M）。



### 警告

与S7-200 24 VDC传感器电源并行连接外部24 VDC电源会导致冲突，因为每个电源都寻求建立自己首选的输出电压电平。

这种冲突的结果会使一个或两个电源缩短使用寿命或立即故障，随后对PLC系统进行不可预知的操作。不可预知的操作可以导致人员死亡或严重伤害，并且/或者损坏设备。

S7-200 DC传感器电源和任何外部电源应该为不同点供电。

## 安装和删除S7-200模块

可以很方便地将S7-200安装在标准DIN横杆或面板上。

### 先决条件

安装或删除任何电气设备之前，确保已经断开该设备的电源。还要确保已经断开任何相关设备的电源。



### 警告

在带电情况下试图安装或删除S7-200或相关设备可能会导致电击或设备运行故障。

安装或删除过程期间，如果未能断开S7-200和相关设备的所有电源，会导致人员死亡或严重伤害，并且/或者损坏设备。

始终遵守正确的安全预防措施，安装或删除S7-200 CPU或相关设备之前，确保断开S7-200的电源。

始终确保更换或安装S7-200设备时使用正确的模块或等同的设备。



### 警告

如果安装错误的模块，S7-200中的程序会不可预测地运行。

未能使用相同型号、方向或顺序更换S7-200设备会导致人员死亡或严重伤害，并且/或者损坏设备。

使用相同的型号更换S7-200设备，确保方向和位置正确。

## 安装尺寸

S7-200 CPU和扩充模块包含安装孔，以帮助安装到面板上。参考表3-1了解安装尺寸。

表3-1 安装尺寸

S7-200模块		宽度A	宽度B
CPU 221和CPU 222		90毫米	82毫米
CPU 224		120.5毫米	112.5毫米
CPU 226和CPU 226XM		196毫米	188毫米
扩充模块:	4和8点DC和继电器I/O (8I、4Q、8Q、4I/4Q) 和模拟输出 (2 AQ)	46毫米	38毫米
扩充模块:	16点数字I/O (16I, 8I/8Q)、模拟I/O (4AI, 4AI/1AQ)、RTD、热电偶、PROFIBUS、以太网、互联网、自动化系统接口、8点AC (8I和8Q)、位置和调制解调器	71.2毫米	63.2毫米
扩充模块:	32点数字I/O (16I/16Q)	137.3毫米	129.3毫米

## 安装CPU或扩充模块

安装S7-200十分简单！只需按照这些步骤进行。

### 面板安装

1. 定位、钻孔和敲开安装孔（M4或美国标准号8），使用表3-1中的尺寸。
2. 使用合适的螺丝将模块固定到面板上。
3. 如果使用扩充模块，将扩充模块带状电缆连接到访问门下面的扩展端口连接器中。

### DIN横杆安装

1. 每隔75毫米将横杆固定在安装面板上。
2. 咔嚓一声打开DIN夹片（位于模块底部）并将模块背面挂到DIN横杆上。
3. 如果使用扩充模块，将扩充模块带状电缆连接到访问门下面的扩展端口连接器中。
4. 向下旋转模块至DIN横杆，合上夹片。仔细检查夹片是否将模块牢固地栓在横杆上。为了避免损坏模块，按在安装孔的突起上，而不要直接按在模块的前端。

**提示**

如果S7-200用于高振动电位环境中或如果垂直安装了S7-200，使用DIN横杆止块十分有用。  
如果系统在高振动环境中，则对S7-200进行面板安装将提供较高级别的振动保护。

**拆除CPU或扩充模块**

要拆除S7-200 CPU或扩充模块，按照下列步骤进行：

1. 从S7-200拆除电源。
2. 断开连接到模块的所有配线和电缆。大部分S7-200 CPU和扩充模块具有可移动的连接器，以使得此工作更容易。
3. 如果有扩充模块连接到正在删除的单元，打开访问盖门，将扩充模块带状电缆从邻近的模块断开。
4. 旋下安装螺丝或咔嚓一声打开DIN夹片。
5. 拆除模块。

**拆除和重新安装接线盒连接器**

大部分S7-200模块具有可移动的连接器的，以使得安装和更换模块容易。参考附录A以确定S7-200模块是否具有可移动的连接器的。可以为不具有可移动连接器的模块订购可选的外置式连接器。参见附录E获取订购号。

**要删除连接器**

1. 打开连接器门来访问连接器。
2. 在连接器中间的槽口中插入小的螺丝起子。
3. 从S7-200外壳撬开螺丝起子，删除终端连接器。参见图3-2。

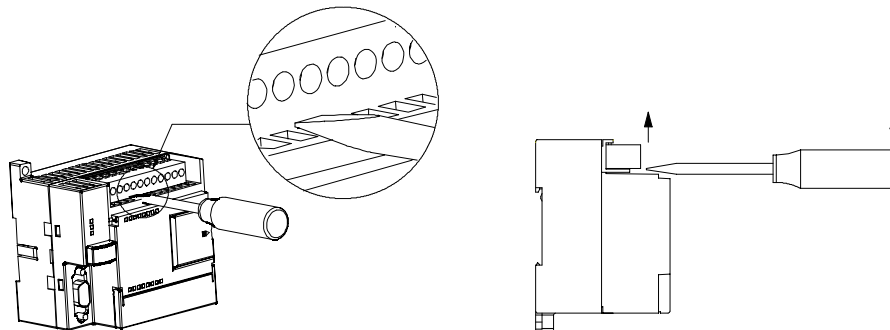


图3-2 删除连接器

**要重新安装连接器**

1. 打开连接器门。
2. 将连接器与单元上的插针对齐，在连接器底部边缘内对齐连接器的导线边缘。
3. 有力地按下以旋转连接器，直到其啮合到位置中。仔细检查以确保连接器正确对齐并完全啮合。

## 接地和布线指南

所有电气设备正确接地和布线可以帮助确保系统的最佳运行和为应用程序和S7-200提供附加的电噪声保护。

### 先决条件

为任何电气设备接地或安装导线之前，确保已经断开该设备的电源。还要确保已经断开任何相关设备的电源。

为S7-200和相关设备布线时，确保遵守所有适用的电气代码。根据所有适用的国家和本地标准安装和操作所有设备。联系本地管理部门，确定哪些代码和标准适用于特定的情况。



#### 警告

带电情况下试图对S7-200或相关设备进行安装或布线可能会导致电击或设备运行故障。安装或删除过程期间，如果未能断开S7-200和相关设备的所有电源，会导致人员死亡或严重伤害，并且/或者损坏设备。

始终遵守合适的安全预防措施，尝试安装或删除S7-200或相关设备之前，请确保已断开S7-200的电源。

设计S7-200系统的接地和布线时，始终要考虑安全问题。S7-200等电子控制设备可能会发生故障并导致受控制或监控的设备意外运行。因此，应该使用独立于S7-200的安全装置进行保护以防止可能的人身伤害或设备损坏。



#### 警告

控制设备在不安全的条件中可能发生故障，导致所控制设备的意外运行。这种意外运行可以导致人员死亡或严重伤害，并且/或者损坏设备。

使用紧急停止功能、机电装置或其它独立于S7-200的冗余安全装置。

### 绝缘指南

S7-200 AC电源边界和AC电路的I/O边界额定为1500 VAC。这些绝缘边界经检查证实为提供AC线路和低压电路之间的安全隔离。

所有连接到S7-200的低压电路（例如24V电源）都必须由经过许可的电源供电，这些电源提供与AC电路和其它高压电路的安全绝缘。这种电源包括国际电气安全标准中定义的双层绝缘，根据不同的标准，输出等级为SELV、PELV、Class 2或限制电压。



#### 警告

将不绝缘或单层绝缘物体用于提供来自AC线路的低压电路，可能会导致在认为可以安全接触的电路中产生危险的电压，例如通讯电路和低压传感器线路中。

这种意外的高压可以导致人员死亡或严重伤害，并且/或者损坏设备。

只能使用已经批准可以安全接触、限制电压电路的高压到低压的电源转换器。

## S7-200接地指南

对应用情况进行接地的最佳方法是确保S7-200和相关设备的所有公共和接地接头都接地到单个点。该单个点应该直接连接到系统的接地地面。

如果要改善电气噪声保护，建议将所有DC公共回路连接到相同的单个点接地地面。将24 VDC传感器供电共同端（M）连接到接地处。

所有地线都应该越短越好，并且应该使用大尺寸的导线，例如2平方毫米（14 AWG）。

确定接地位置时，务必考虑安全接地要求和保护性中断设备的正确操作。

## S7-200布线指南

为S7-200设计和布线时，提供单个断开开关，用来同时给S7-200 CPU电源、所有输入电路和所有输出电路断电。提供过流保护（例如保险丝或断路器）来限制供电电线上的故障电流。可能想要通过在每个输出电路中放置保险丝或其它电流限制器来提供附加的保护。

为任何可能遭受闪电电涌的线路安装合适的电涌抑制设备。

避免将低压信号线和通讯电缆放在与AC导线和高能量、快速转换的DC导线相同的线盒中。始终要成对布线，中性线或公共线与热线或信号线成对。

尽可能使用最短的导线，确保导线尺寸正确以输送需要的电流。连接器接受的导线尺寸从2平方毫米至0.3平方毫米（14 AWG至22 AWG）。使用屏蔽导线以最好地防止电气噪声。通常，将屏蔽在S7-200处接地效果最好。

为由外部电源供电的输入电路布线时，请在该电路中安装过流保护设备。对于由S7-200的24 VDC传感器供电的电路，外部保护并无必要，因为传感器电源已经经过电流限制。

大部分S7-200模块都具有可移动的连接器的，用于用户布线。（参考附录A以确定模块是否具有可移动的连接器的。）要防止连接器松开，确保连接器固定牢靠并且导线牢固地安装在连接器中。要避免损坏连接器，必须小心不要将螺丝旋得过紧。连接器螺丝的最大扭矩为0.56牛米（5英寸-磅）。

要帮助防止安装中不需要的电流，S7-200在某些点提供绝缘边界。为系统规划布线时，应该考虑这些绝缘边界。参考附录A了解提供的绝缘数量和绝缘边界位置。如果绝缘边界等级低于1500 VAC，则不能依赖其作为安全边界。



### 提示

对于通讯网络，在不使用中继器的情况下，通讯电缆的最大长度是50米。S7-200上的通讯端口不绝缘。参考第7章获取更多信息。

## 抑制电路指南

应该安装带抑制电路的感应负载，以便在控制输出断开时限制电压上升。抑制电路保护输出不会因为高感应开关电流而过早发生故障。此外，抑制电路在切换感应负载时限制产生的电气噪声。



### 提示

所提供的抑制电路的效果取决于应用情况，并且必须为特定用途进行检验。始终确保抑制电路中使用的所有组件都达到应用中使用的等级。

3

### DC输出和控制DC负载的继电器

DC输出具有对大部分应用都足够的内部保护。因为继电器可用于DC或AC负载，所以不提供内部保护。

图3-3显示用于DC负载的实例抑制电路。在大部分应用中，在感应负载上并联一个二极管（A）就够了，但是在应用中需要更快的断开时间，那么建议并联稳压二极管（B）。确保正确调整稳压二极管的大小，以适合输出电路中的电流。

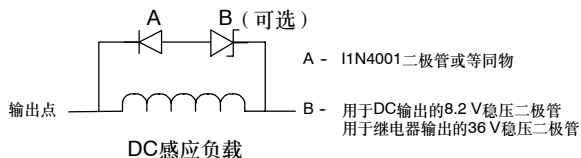


图3-3 用于DC负载的抑制电路

### AC输出和控制AC负载的继电器

AC输出具有对大部分应用都足够的内部保护。因为继电器可用于DC或AC负载，所以不提供内部保护。

图3-4显示用于AC负载的实例抑制电路。使用继电器或AC输出来转换115 V/230 VAC负载时，如图所示布置电阻器/电容器网络，与AC负载并联。也可以使用金属氧化物变阻器（MOV）来限制峰值电压。确保MOV的工作电压至少高出额定线路电压20%。

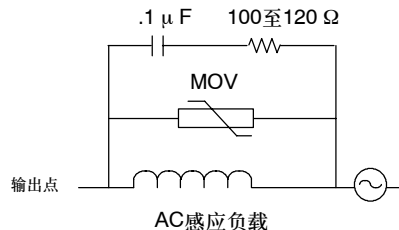


图3-4 用于AC负载的抑制电路

### 注意

继电器扩充模块用于转换230 VAC感应负载时，必须如图3-4中所示将外部电阻器/电容器噪声抑制电路与AC负载并联。



# PLC原则

# 4

S7-200的基本功能是监控域输入并且基于控制逻辑，接通或断开域输出设备。本章说明用于执行程序的原则，使用的各种类型的内存，以及内存如何保持。

## 本章内容

了解S7-200如何执行控制逻辑 .....	22
存取S7-200的数据 .....	24
了解S7-200如何保存和恢复数据 .....	34
存储程序到内存磁带 .....	36
选择S7-200 CPU的操作模式 .....	37
使用程序来保存V内存到EEPROM .....	38
S7-200的特征 .....	39

## 了解S7-200如何执行控制逻辑

S7-200持续循环通过控制逻辑到程序中，读取和写入数据。

### S7-200建立程序与实际输入和输出联系

S7-200的基本操作非常简单：

- ❑ S7-200读取输入状态。
- ❑ 存储在S7-200中的程序使用这些输入来判断控制逻辑。当程序运行时，S7-200更新数据。
- ❑ S7-200将数据写入输出。

图4-1显示一个关于电气继电器图联系到S7-200的简单图示。在该实例中，启动电机的开关状态与其他输入状态相结合。这些状态的计算确定输出状态，后者传送带启动电机的传动器。

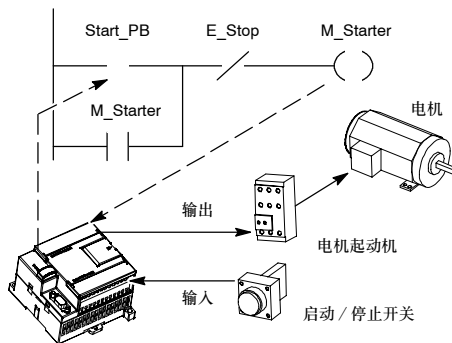


图4-1 控制输入和输出

### S7-200以扫描循环方式执行任务

S7-200反复执行一系列任务。此任务的循环执行称为扫描循环。如图4-2所示，S7-200在一个扫描循环期间完成大多数或全部下列任务：

- ❑ 读取输入：S7-200复制实际输入的状态到进程图像输入寄存器。
- ❑ 在程序中执行控制逻辑：S7-200执行程序指令并存储数值在各种内存区域中。
- ❑ 处理任何通讯请求：S7-200完成任何用于通讯请求的任务。
- ❑ 执行CPU自测诊断：S7-200确保固化程序、程序内存以及所有扩充模块工作正常。
- ❑ 写入输出：存储在进程图像输出寄存器的数值写入实际输出。

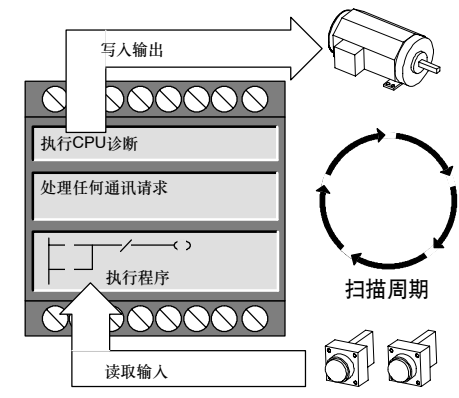


图4-2 S7-200扫描循环

扫描循环的执行取决于S7-200是为停止模式还是运行模式。在运行模式，执行程序；在停止模式，不执行程序。

## 读取输入

**数字输入:** 每个扫描循环由读取数字输入的当前值开始, 然后写入这些数值到进程图像输入寄存器。

**模拟输入:** S7-200不将模拟输入作为普通扫描循环部分更新, 除非模拟输入过滤启用。模拟过滤器能提供更稳定的信号。用户可以对每个模拟输入点启用模拟过滤器。

当对模拟输入启用模拟输入过滤, S7-200每经过一个扫描循环更新一次模拟输入, 执行过滤功能并且内部存储过滤数值。这样每次程序存取模拟输入都能提供已过滤的数值。

未启用模拟过滤时, 当每次程序访问模拟输入, S7-200从实际模块读取模拟输入的数值。



### 提示

模拟输入过滤能提供更稳定的模拟信号。若输入信号随着时间缓慢变化, 对应用程序使用模拟输入过滤器。如果信号是高速信号, 那么不应启用模拟过滤器。

不要对通过数字信息或以模拟字进行报警指示的模块使用模拟过滤器。对RTD、热电偶和AS接口主模块始终禁用模拟过滤。

## 执行程序

在扫描循环执行阶段, S7-200执行程序, 由第一条指令开始到最后一条结束。执行程序或中断例行程序期间, 直接输入/输出指令能直接访问输入和输出。

如果在程序中使用中断, 关联中断事件的中断例行程序存储为程序的一部分。中断例行程序不作为普通扫描循环的一部分来执行, 但是当中断事件产生时执行(可能在扫描循环中的任意点)。

## 处理任何通讯请求

扫描循环的讯息处理阶段, S7-200处理任何从通讯端口或智能输入/输出模块接收到的讯息。

## 执行CPU自测诊断

扫描循环期间, S7-200检查CPU操作正确与否、内存区域和任意扩充模块的状态。

## 写入数字输出

在每个扫描循环的结尾, S7-200写入存储在进程图像输出寄存器的数值到数字输出。(模拟输出立即更新, 独立于扫描循环。)

## 存取S7-200的数据

S7-200将信息存储在具有唯一地址的不同内存位置上。可以明确识别要存取的内存地址。这能允许程序直接存取信息。表4-1显示可以由不同大小数据表达的整数值范围。

表4-1 用于不同大小数据的十进制和十六进制范围

表达式	字节 (B)	字 (W)	双字 (D)
不带符号的整数	0至255 0至FF	0 ~ 65,535 0至FFFF	0至4,294,967,295 0至FFFF FFFF
带符号的整数	-128至+127 80至7F	-32,768到+32,767 8000至7FFF	-2,147,483,648到+2,147,483,647 8000 0000至7FFF FFFF
实数 IEEE 32位浮点数	不适用	不适用	+1.175495E-38至+3.402823E+38 (正数) -1.175495E-38至-3.402823E+38 (负数)

要存取内存区域的位，需要指定地址，其中包含内存区域标识符、字节地址和位号。图4-3显示存取位（也称为“字节.位”编址）的实例。在此实例中，内存区域和字节地址（I = 输入，而3 = 字节3）由句点（“.”）来分隔位地址（位4）。



图4-3 字节.位编址

可以通过使用字节地址格式按字节、字或双字在大多数内存区域 (V、I、Q、M、S、L和SM) 存取数据。要在内存中存取数据的字节、字或双字，必须指定地址，其方式类似于指定位地址。这包括区域标识符、数据大小标志和字节、字或双字数值的起始字节地址，如图4-4所示。

在其他内存区域中的数据 (诸如T、C、HC和累加器) 通过使用地址格式 (其中包含区域标识符和设备编号) 进行存取。

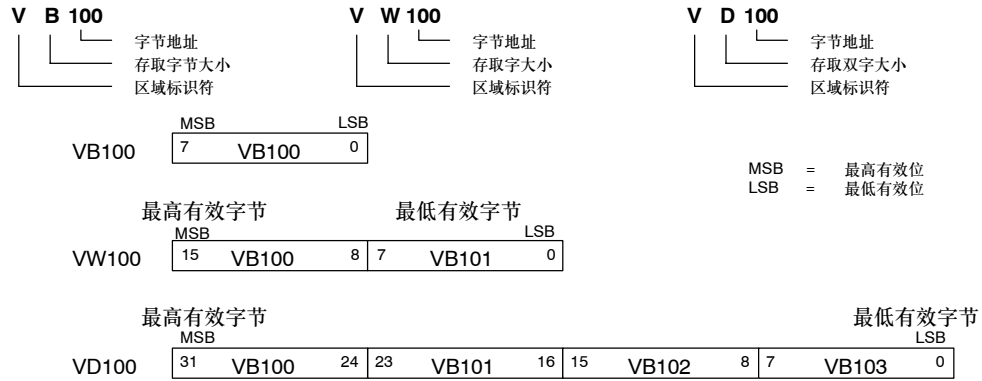


图4-4 比较字节、字、和双字存取相同地址

## 在内存区域中存取数据

### 进程图像输入寄存器: I

S7-200在每个扫描循环的开始处采样实际输入点并写入这些数值到进程图像输入寄存器。可以以位、字节、字或双字存取进程图像输入寄存器:

位:  $I[\text{字节地址}][\text{位地址}]$  I0.1  
字节、字、或双字:  $I[\text{大小}][\text{起始字节地址}]$  IB4

### 进程图像输出寄存器: Q

在每个扫描循环的结尾, S7-200复制存储在进程图像输出寄存器的数值到实际输出点。可以按位、字节、字或双字存取进程图像输出寄存器:

位:  $Q[\text{字节地址}][\text{位地址}]$  Q1.1  
字节、字或双字:  $Q[\text{大小}][\text{起始字节地址}]$  QB5

### 变量内存区域: V

可以使用V内存来存储通过程序中的控制逻辑执行的操作的中间结果。也可以使用V内存来存储其他与进程或任务有关的数据。可以按位、字节、字或双字形式存取V内存区域:

位:  $V[\text{字节地址}][\text{位地址}]$  V10.2  
字节、字、或双字:  $V[\text{大小}][\text{起始字节地址}]$  VW100

### 位存储器区域: M

可以使用位存储器区域(M内存)作为控制继电器来存储操作或其他控制信息的中间状态。可以按位、字节、字或双字形式存取位存储器区域:

位:  $M[\text{字节地址}][\text{位地址}]$  M26.7  
字节、字或双字:  $M[\text{大小}][\text{起始字节地址}]$  MD20

**计时器内存区域: T**

S7-200提供计时器以1毫秒、10毫秒、或100毫秒的分辨率（基于时间的增量）来计算时间增量。两个变量与计时器关联:

- 当前值: 该16位有符号整数存储由计时器计算的时间量。
- 计时器位: 设置或清除该位作为当前和预设值比较的结果。预置数值输入为计时器指令的一部分。

可以通过使用计时器地址（T + 计时器号码）存取这些变量。计时器位或当前值的存取取决于使用的指令: 位操作数指令存取计时器位, 字操作数指令存取当前值。如图4-5所示, 当移动字指令存取计时器的当前值, 常开接点指令存取计时器位。

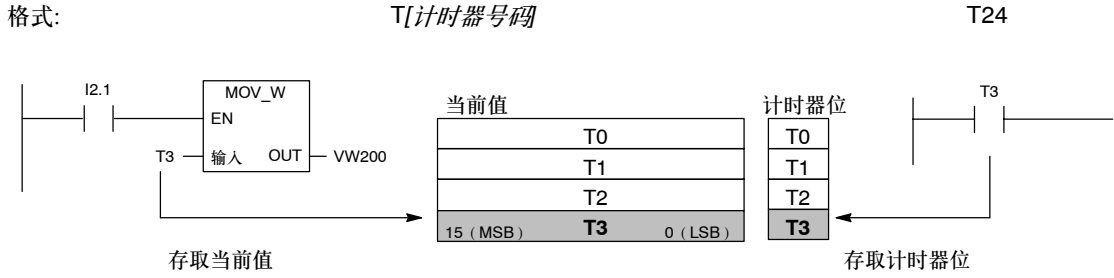


图4-5 存取计时器位或计时器的当前值

**计数器内存区域: C**

S7-200提供三种类型的计数器, 可以在计数器输入上计算每个从低到高转换事件: 一种类型只往上计算, 一种类型只往下计算, 还有一种类型既往上计算又往下计算。两个变量与计数器关联:

- 当前值: 该16位有符号整数存储累加数。
- 计数器位: 设置或清除该位作为当前和预设值比较的结果。预置数值输入为计数器指令的一部分。

可以通过使用计数器地址（C + 计数器号码）存取这些变量。计数器位或当前值的存取取决于使用的指令: 位操作数指令存取计数器位, 字操作数指令存取当前值。如图4-6所示, 当移动字指令存取计数器的当前值, 常开接点指令存取计数器位。

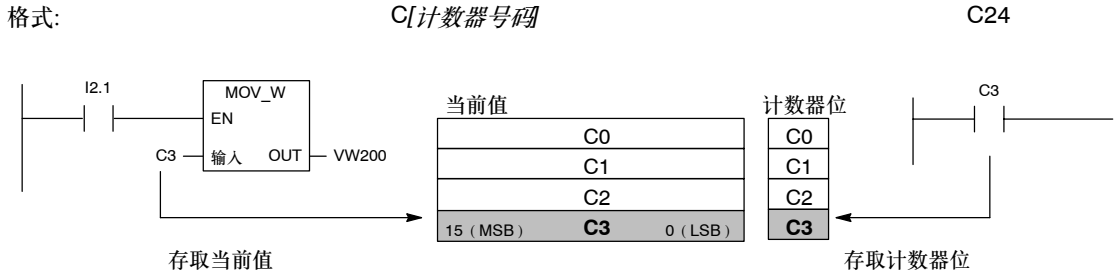


图4-6 存取计数器位或计数器的当前值

**高速计数器: HC**

高速计数器计算高速事件独立于CPU扫描。高速计数器具有有符号32位整数计算值（或当前值）。要对高速计数器存取计算数值，需要指定高速计数器的地址、使用的内存型号（HC）和计数器号码（诸如HC0）。高速计数器的当前值是一个只读数值并且仅能按双字（32位）编址。

格式: HC[高速计数器号码] HC1

**累加器: AC**

累加器是读取/写入设备，能象内存一样使用。例如，可以使用累加器对子例行程序进行参数的往复传递，并存储用于计算的中间数值。S7-200提供四个32位累加器（AC0、AC1、AC2和AC3）。可以按字节、字或双形式字存取累加器中的数据。

存取的数据大小取决于用于访问累加器的指令。如图4-7所示，使用存储在累加器中数值的最低有效8位或16位来按字节或字访问累加器。要按双字访问累加器，都使用32位。

关于如何使用中断子例行程序中的累加器的信息，请参考章6中的中断指令。

格式: AC[累加器号码] AC0

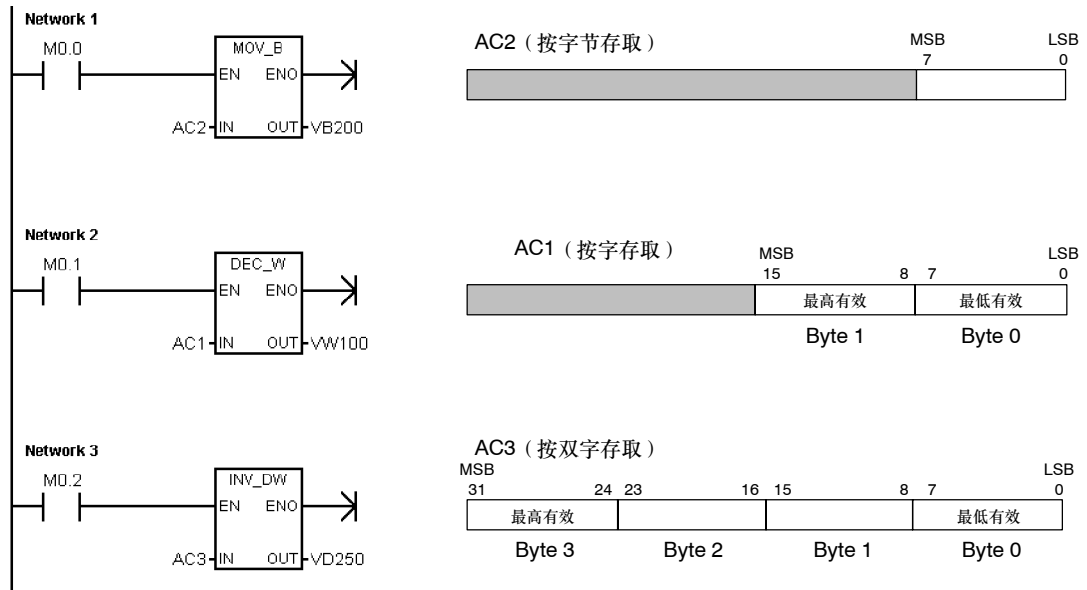


图4-7 存取累加器

**特殊内存: SM**

SM位提供了在CPU和程序之间进行信息通讯的一种手段。可以使用这些位来选择和控制S7-200 CPU的一些特殊功能，例如：用于第一次扫描循环的位、以固定比率切换的位、或显示数学或操作指令状态的位。（关于SM位的更多信息，请参见附录D。）可以按位、字节、字或双字形式存取SM位：

位:	SM[字节地址].[位地址]	SM0.1
字节、字、或双字:	SM[大小][起始字节地址]	SMB86

**局部内存区域: L**

S7-200提供64个字节的局部内存，其中的60字节可作为高速暂存存储器或用于传递形式参数到子例行程序。

**提示**

如果在LAD或FBD中编程，STEP 7-Micro/WIN保留局部内存的最后四个字节为自身使用。如果在STL中编程，L内存的所有64个字节都可存取，但是建议不要使用L内存的最后四个字节。

局部内存类似于V内存，但有一点主要区别。V内存为全局范围，而L内存是局部范围。术语全局范围的意思是同样的内存位置可以从任意程序实体（主程序、子例行程序或中断例行程序）存取。术语局部范围的意思是内存分配与特殊程序实体相联系。S7-200为主程序分配64个字节的L内存、64个字节用于每个子例行程序嵌套层，以及64个字节用于中断例行程序。

主程序的L内存分配不能从子例行程序或中断例行程序存取。子例行程序不能存取主程序、中断例行程序或其他子例行程序的L内存分配。同样，中断例行程序不能存取主程序或子例行程序的L内存分配。

根据需要由S7-200进行L内存分配。也就是说当程序的主体部分执行时，子例行程序和中断例行程序的L内存分配不存在。中断产生或调用子例行程序时，局部内存根据要求进行分配。L内存的新分配可能重新使用不同子例行程序或中断例行程序的相同L内存位置。

分配时L内存未由S7-200初始化且可能包含任意数值。当在子例行程序调用中传递形式参数，传递的参数数值由S7-200置于调用的子例行程序的相关L内存位置中。不接收数值作为形式参数传递步骤结果的L内存位置，在分配时将不会初始化并且可能包含任意数值。

位:	L[字节地址].[位地址]	L0.0
字节、字或双字:	L[大小][起始字节地址]	LB33



**模拟输入: AI**

S7-200转换模拟数值（例如温度或电压）为一个字长度的（16位）数字值。可以通过区域标识符（AI）、数据大小（W）和起始字节地址存取这些数值。因为模拟输入是字并且始终以偶数字节（例如0、2或4）开始，所以可以用偶数字节地址（例如AIW0、AIW2或AIW4）存取它们。模拟输入数值是只读数值。

格式:  $\text{AIW}[\text{起始字节地址}] \quad \text{AIW4}$

**模拟输出: AQ**

S7-200转换字长度（16位）的数字值到电流或电压，与数字值成比例（例如对于电流或电压）。可以通过区域标识符（AQ）、数据大小（W）和起始字节地址写入这些数值。因为模拟输出是字并且始终以偶数字节（例如0、2或4）开始，所以可以用偶数字节地址（例如AQW0、AQW2或AQW4）写入它们。模拟输出值是只写数值。

格式:  $\text{AQW}[\text{起始字节地址}] \quad \text{AQW4}$

**顺序控制继电器 (SCR) 内存区域: S**

SCR或S位用于在相同程序段中组织机械设备操作或步骤。SCR允许控制程序的逻辑段。可以按位、字节、字或双字存取S位。

位:  $\text{S}[\text{字节地址}][\text{位地址}] \quad \text{S3.1}$   
 字节、字或双字:  $\text{S}[\text{大小}][\text{起始字节地址}] \quad \text{SB4}$

**用于实数的格式**

实数（或浮点数）表示为32位单精度型数字，其格式在ANSI/IEEE 754-1985标准中有描述。参见图4-8。实数以双字长度存取。

对于S7-200，浮点数精确到小数位6位。因此，输入浮点常量时可以指定最多6个小数位。

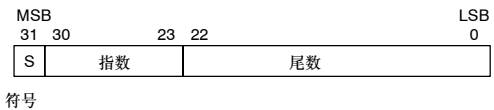


图4-8 实数格式

**计算实数时的精度**

计算涉及一长串数值，且其中包含有非常大和非常小的数字，可导致不精确的结果。如果数字之间相差为以10为底数的x次幂，若 $x > 6$ ，就会发生上述情况。

例如:  $100\ 000\ 000 + 1 = 100\ 000\ 000$

## 字符串格式

字符串是字符的序列，其中每个字符都存储为一个字节。字符串的第一个字节定义了字符串的长度，即字符数目。图4-9显示字符串格式。字符串的长度为0至254个字符加上长度字节，这样字符串的最大长度为255个字节。



图4-9 字符串格式

# 4

## 对S7-200指令指定常数

在许多S7-200指令中可以使用常数。常量可以是字节、字或双字。S7-200以二进制数字存储所有常量，这样就可以表示为十进制、十六进制、ASCII或实数（浮点数）格式。参见表4-2。

表4-2 常数的表达式

表达式	格式	实例
十进制	[十进制数值]	20047
十六进制	16#[十六进制数值]	16#4E4F
二进制	2#[二进制数字]	2#1010_0101_1010_0101
ASCII	'[ASCII文本]'	'文本放在单引号之间。'
实数	ANSI/IEEE 754-1985	+1.175495E-38 (正数)      -1.175495E-38 (负数)



### 提示

S7-200 CPU不支持“数据类型”或数据勾选（例如指定常量存储为整数、有符号整数、或双整数）。例如，“添加”指令可以将VW100中的数值作为有符号整数使用，而“异或”指令可以将VW100中的相同数值作为无符号二进制数值使用。

## 编址局部和扩展输入 / 输出

CPU提供的局部I/O提供I/O地址的固定设置。通过连接扩展I/O模块到CPU右侧，组成I/O链，可以添加I/O点到S7-200 CPU。模块点的地址取决于I/O类型和模块在链中的位置，与前述相同类型的输入或输出模块相关。例如，输出模块不影响输入模块上点的地址，反之亦然。同样，模拟模块不影响数字模块的编址，反之亦然。



### 提示

数字扩充模块始终以8位的增量（1个字节）保留进程图像寄存器空间。如果模块不为每个保留字节的每个位提供实际点，这些不使用的位不能分配到I/O链中的随后模块。对于输入模块，保留字节中不使用的位随着每个输入更新周期设置为零。

模拟扩充模块始终分配在两点的增量中。如果模块不为每个点提供实际I/O，这些I/O点将丢失并且不能分配到I/O链中的后续模块。

图4-10提供用于特殊硬件配置的I/O编号的实例。编址间隙（显示为灰色斜体文本）不能由程序使用。

CPU 224	4输入 / 4输出	8输入	4模拟输入 1模拟输出	8输出	4模拟输入 1模拟输出
I0.0    Q0.0 I0.1    Q0.1 I0.2    Q0.2 I0.3    Q0.3 I0.4    Q0.4 I0.5    Q0.5 I0.6    Q0.6 I0.7    Q0.7 I1.0    Q1.0 I1.1    Q1.1 I1.2    Q1.2 I1.3    Q1.3 I1.4    Q1.4 I1.5    Q1.5 I1.6    Q1.6 I1.7    Q1.7	模块0 I2.0    Q2.0 I2.1    Q2.1 I2.2    Q2.2 I2.3    Q2.3 <i>I2.4    Q2.4</i> <i>I2.5    Q2.5</i> <i>I2.6    Q2.6</i> <i>I2.7    Q2.7</i>	模块1 I3.0 I3.1 I3.2 I3.3 I3.4 I3.5 I3.6 I3.7	模块2 AIW0    AQW0 AIW2    AQW2 AIW4 AIW6	模块3 Q3.0 Q3.1 Q3.2 Q3.3 Q3.4 Q3.5 Q3.6 Q3.7	模块4 AIW8    AQW4 AIW10    AQW6 AIW12 AIW14
局部I/O	扩展I/O				

图4-10 为局部和扩展I/O（CPU224）采样I/O地址

## 使用指针对S7-200内存区域间接编址

间接编址使用指针来存取内存中的数据。指针是双字内存位置，其包含另一内存位置的地址。只能使用V内存位置、L内存位置或累加器寄存器（AC1、AC2、AC3）作为指针。要创建指针，必须使用“移动双字”指令来移动间接编址的内存位置的地址到指针位置。指针也可作为参数传递到子例行程序。

S7-200允许指针存取下列内存区域：I、Q、V、M、S、T（仅限当前值）和C（仅限当前值）。不能使用间接编址存取单个位，也不能存取AI、AQ、HC、SM或L内存区域。

要间接存取内存地址中的数据，可通过输入符号（&）和要编址的内存位置建立指针到该位置。指令的输入操作数前必须标记（&）符号以表示其是内存位置的地址，而不是其内容，将移动到指令输出操作数（指针）中的识别位置。

在指令的操作数前输入星号（\*）指定该操作数为指针。如图4-11所示，输入\*AC1指定AC1被“移动字”（MOVW）指令引用的字长度数值的指针。在该实例中，存储在VB200和VB201中的数值移动到累加器AC0。

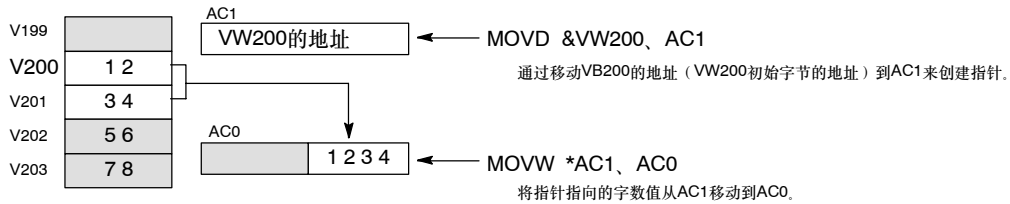


图4-11 创建和使用指针

如图4-12所示，可以改变指针的数值。由于指针是32位数值，所以使用双字指令来修改指针数值。简单的数学运算，诸如加法或增量，可以用来修改指针数值。

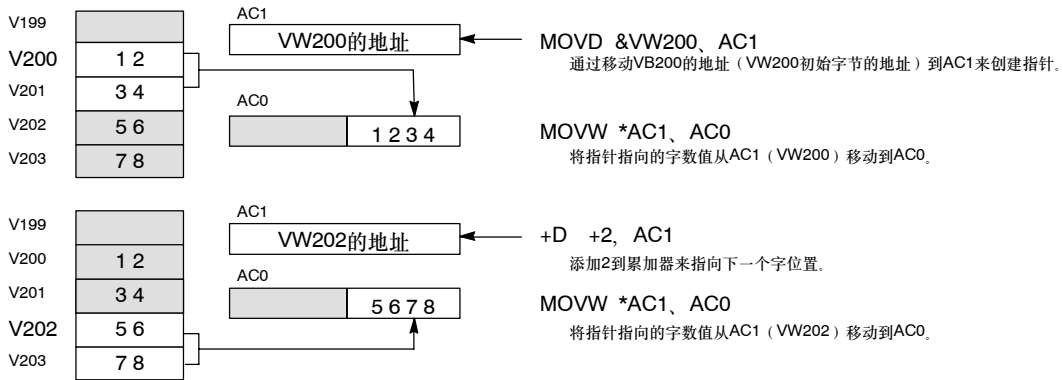


图4-12 修改指针

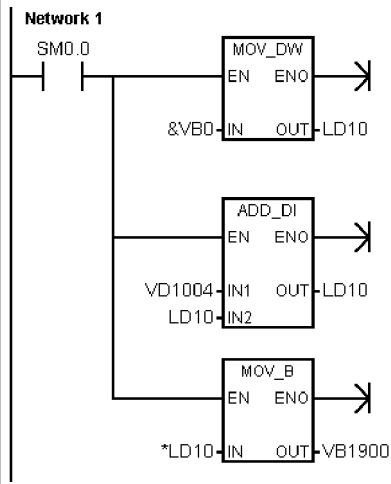


### 提示

记得调整存取的数据大小：要存取字节，指针数值以1递增；要为计时器或计数器存取字或当前值，指针数值增加或以2递增；而要存取双字，指针数值增加或以4递增。

## 在内存中使用偏移量存取数据的抽样程序

该实例使用LD10为地址VB0的指针。然后通过存储在VD1004的偏移量递增指针。LD10然后指向V内存中的另一地址（VB0 + 偏移量）。存储在V内存地址中的数值由LD10指向然后复制到VB1900。通过改变VD1004中的数值，可以存取任意V内存位置。



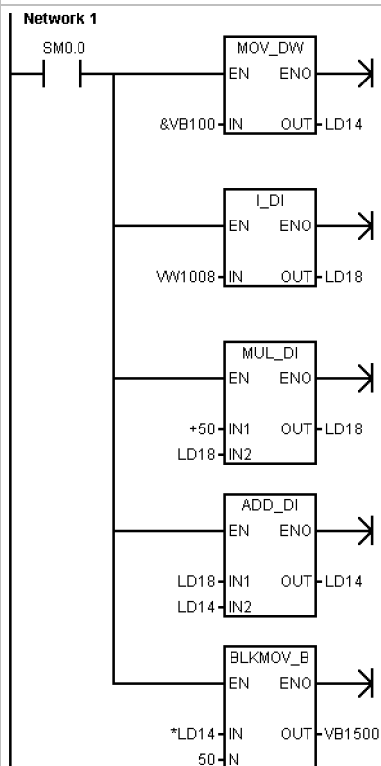
Network 1 //如何使用偏移量来读取任意VB位置的数值:  
//  
//1. 载入V内存的起始地址到指针。  
//2. 添加偏移量数值到指针。  
//3. 从V内存位置（偏移量）复制数值到VB1900。

```
LD SM0.0
MOVD &VB0, LD10
+D VD1004, LD10
MOVB *LD10, VB1900
```

4

## 使用指针在表中存取数据的抽样程序

该实例使用LD14为指针，其指向存储在从VB100开始的配方表中的配方。在该实例中，VW1008存储索引到表中指定的配方。如果表中的每个配方为50个字节长，将下标乘以50获得指定配方的起始地址的偏移量。通过添加偏移量到指针，可以存取表中的单个配方。在该实例中，配方复制到起始于VB1500的50个字节。



Network 1 //如何从配方表传送配方:  
// - 每个配方为50个字节长。  
// - 索引参数（VW1008）识别要载入的配方。  
//  
//1. 创建到配方表的起始地址的指针。  
//2. 转换配方索引到双字数值。  
//3. 乘以偏移量以适合每个配方大小。  
//4. 添加调整的偏移量到指针。  
//5. 传送所选择的配方通过VB1549到VB1500。

```
LD SM0.0
MOVD &VB100, LD14
ITD VW1008, LD18
*D +50, LD18
+D LD18, LD14
BMB *LD14, VB1500, 50
```

## 了解S7-200如何保存和恢复数据

S7-200提供了各种保护措施来确保S7-200的程序、程序数据和配置数据正确保留。

S7-200提供超级电容器能在断电后维持RAM的完整性。根据S7-200的模块，超级电容器可以维持RAM几天。

S7-200提供EEPROM来永久存储所有的程序、用户所选择的数据区和配置数据。

S7-200还支持选用电池盒，这样能延长在S7-200断电后RAM保持的时间。只有当超级电容器耗尽后电池盒才提供电源。

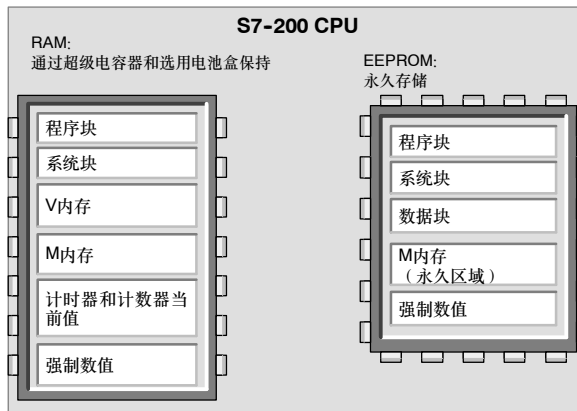


图4-13 S7-200 CPU存储区域

### 下载和上载项目元素

项目由三个元素组成：程序块、数据块（可选）和系统块（可选）。

图4-14显示项目如何下载到S7-200。

下载项目时，下载项目的元素存储在RAM区域中。S7-200也能自动复制用户程序、数据块和系统块到EEPROM以永久存储。

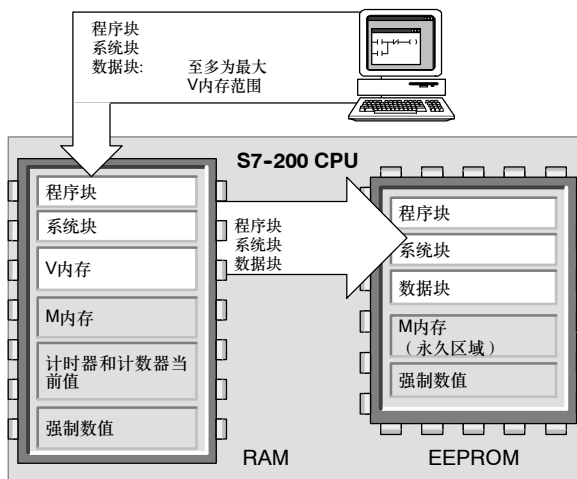


图4-14 下载项目到S7-200

图4-15显示项目如何从S7-200上载。

当上载项目到计算机，S7-200从RAM上载系统块并且从EEPROM上载程序块和数据块。

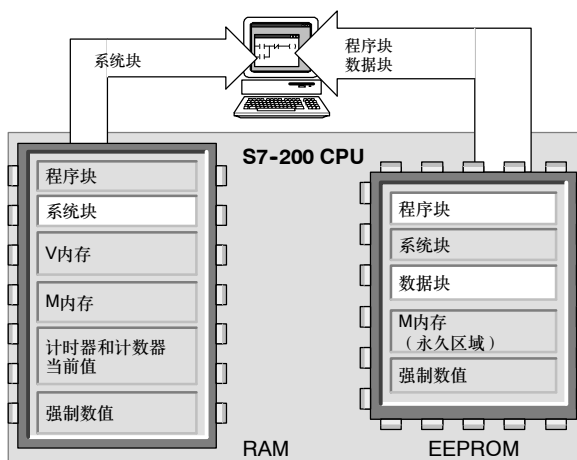


图4-15 从S7-200上载项目

### 断电时保存保留M内存区域

如果配置了位存储器首先的14个字节（MB0到MB13）来作为保留，在S7-200断电的情况下，这些字节永久保存到EEPROM。

如图4-16所示，S7-200移动这些M内存保留区到EEPROM。

M内存的最先14个字节的缺省设置为不保留。当断开S7-200电源时，默认禁用通常产生的保存。

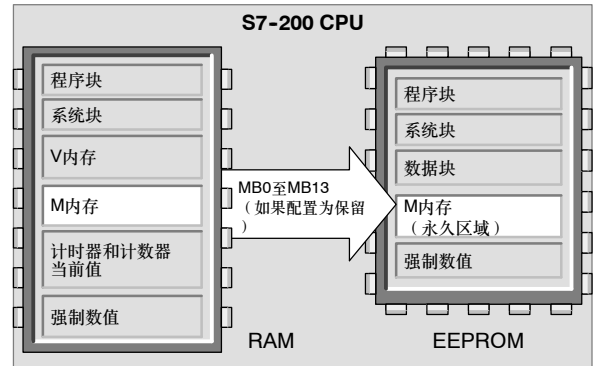


图4-16 断电时保存M内存

### 通电后恢复数据

通电时，S7-200从EEPROM内存恢复程序块和系统块，如图4-17所示。也在通电时，S7-200检查RAM检验超级电容器成功的保持了存储在RAM内存的数据。如果RAM成功保持，保留RAM区域不更改。

V内存的保留和非保留区域从EEPROM中的相关数据块恢复。如果RAM内容未保持（例如长期断电之后），S7-200清除RAM（包括保留和非保留范围）并为通电后首次扫描循环设置保留数据丢失内存位（SM0.2），然后将存储在EEPROM的数据复制到RAM。

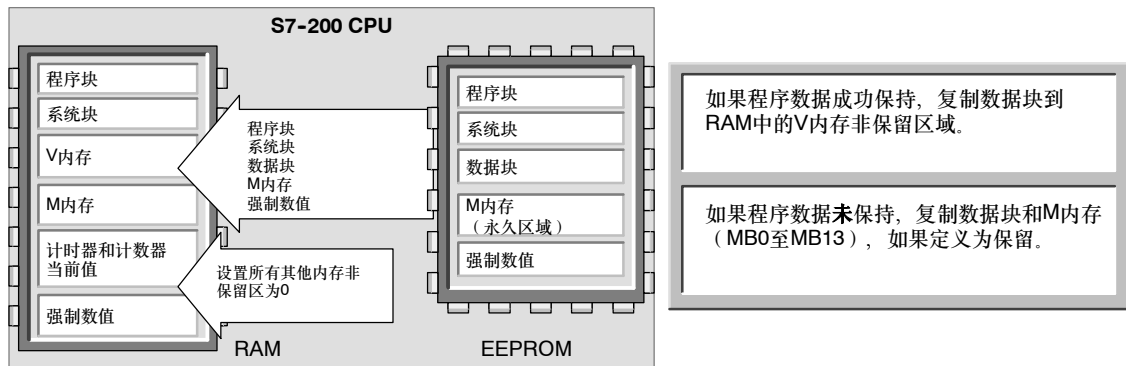


图4-17 通电后恢复数据

## 存储程序到内存磁带

S7-200支持选用内存磁带，其能为程序提供便携EEPROM存储器。S7-200在内存磁带中存储下列元素：程序块、数据块、系统块和强制数值。

仅当S7-200通电并且在STOP（停止）模式而内存磁带也已安装时，可以将程序从RAM复制到内存磁带。当S7-200通电时，用户可以安装或拆除内存磁带。

### 小心

静电释放可能会损坏内存磁带或S7-200 CPU上的插座。

连接接地导电板和 / 或在操作磁带时带上接地腕带。将磁带存储在导电容器中。

安装内存磁带，从S7-200 CPU拆除塑料插槽盖并插入内存磁带到插槽中。内存磁带是正确安装的关键。

### 复制程序到内存磁带

安装内存磁带之后，使用下列步骤复制程序：

1. 将S7-200 CPU置于STOP（停止）模式。
2. 如果程序未下载到S7-200，下载程序。
3. 选择PLC > 程序内存磁带（**Program Memory Cartridge**）菜单命令将程序复制到内存磁带。图4-18显示了存储在内存磁带上的CPU内存元素。
4. 供选用：拆除内存磁带并更换S7-200上的封面。

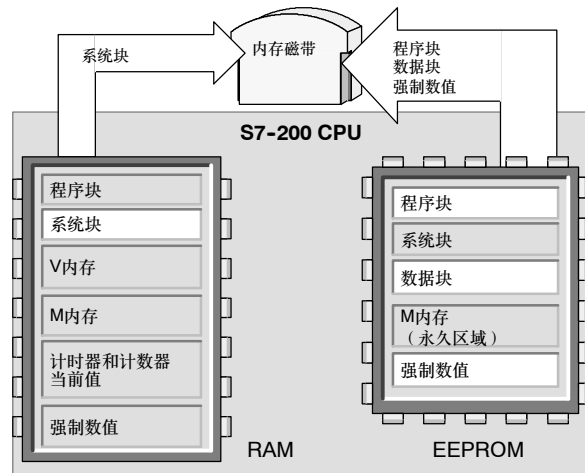


图4-18 复制到内存磁带

### 从内存磁带恢复程序

要将程序从内存磁带传送到S7-200，必须对装有内存磁带的S7-200执行电源循环。

### 注意

对装有空白内存磁带的S7-200 CPU通电或由不同型号的S7-200 CPU编程的内存磁带可能引起错误。由低数字型号CPU编程的内存磁带可由高数字型号CPU读取。然而，反之不可行。例如，由CPU 221或CPU 222编程的内存磁带可由CPU 224读取，但是内存磁带若由CPU 224编程，则会被CPU 221或CPU 222拒绝。

拆除内存磁带并接通S7-200电源。通电后，如需要，内存磁带可以插入并重新编程。



如图4-19所示，在用安装的内存磁带断开电源重新启动后，S7-200执行下列任务：

1. 如果内存磁带的内容与EEPROM的不同，S7-200清除RAM。
2. S7-200复制内存磁带的内容到RAM。
3. S7-200复制程序块、系统块和数据块到EEPROM。

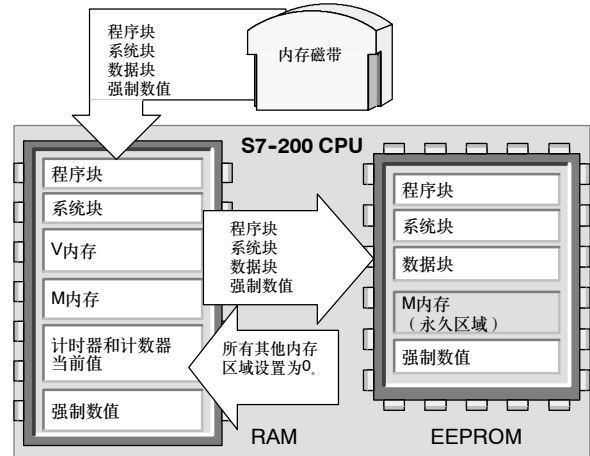


图4-19 从内存磁带恢复

## 选择S7-200 CPU的操作模式

S7-200有两种操作模式：STOP（停止）模式和RUN（运行）模式。CPU前端上的LED状态指示了当前操作模式。在STOP（停止）模式，S7-200不执行程序，而且可以下载程序或CPU配置。在RUN（运行）模式，S7-200 运行程序。

- S7-200提供模式转换来更换操作模式。可以使用模式转换（位于S7-200前存取门下方）来手动选择操作模式：设置模式转换到STOP（停止）模式停止程序的执行；设置模式转换到RUN（运行）模式启动程序的执行；以及设置模式转换到TERM（终端）模式不改变操作模式。

如果模式转换设置为STOP（停止）或TERM（终端）时电源循环产生，电源恢复后S7-200自动转为STOP（停止）模式。如果模式转换设置为RUN（运行）时电源循环产生，电源恢复后S7-200跳转到RUN（运行）模式。

- STEP7-Micro/WIN允许改变联机S7-200的操作模式。要启用软件来改变操作模式，必须手动设置S7-200上的模式转换为TERM或RUN。可以使用PLC > 停止（STOP）或PLC > 运行（RUN）菜单命令或工具栏上的关联按钮来改变操作模式。
- 可以在程序中插入STOP（停止）指令来改变S7-200为STOP（停止）模式。这允许暂停基于程序逻辑的程序执行。关于停止（STOP）指令的更多信息，请参见第6章。

## 使用程序保存V内存到EEPROM

可以保存存储在V内存区域任意位置的数值（字节、字或双字）到EEPROM。保存到EEPROM的操作一般增加扫描时间，最多为5毫秒。由“保存”操作所写入的数值覆盖所有存储在EEPROM的V内存区域中的先前数值。

保存到EEPROM操作不更新内存磁带中的数据。



### 提示

由于至EEPROM的“保存”操作次数是有限的（最小100,000，一般为1,000,000），所以应确保仅保存必要的数值。否则，EEPROM将耗尽且CPU会出故障。一般地，应该对极少产生的指定事件执行保存操作。

例如，若S7-200的扫描时间为50毫秒而每次扫描保存一个数值，那么EEPROM能持续至少5,000秒，少于一个半小时。另一方面，如果数值每小时保存一次，EEPROM将至少能维持11年。

## 4

### 复制V内存到EEPROM

特殊内存字节31（SMB31）命令S7-200复制V内存中的数值到EEPROM的V内存区域。特殊内存字32（SMW32）存储要复制数值的地址位置。图4-20显示SMB31和SMW32的格式。

使用下列步骤执行S7-200在V内存中保存或写入指定数值：

1. 载入保存在SMW32中数值的V内存地址。
2. 载入SM31.0和SM31.1中数据的大小，如图4-20所示。
3. 设置SM31.7为1。

在每个扫描循环的结尾，S7-200检查SM31.7；如果SM31.7等于1，指定的数值保存到EEPROM。当S7-200重新设定SM31.7为0，操作完成。

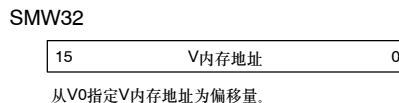
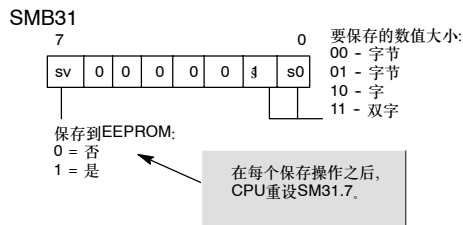
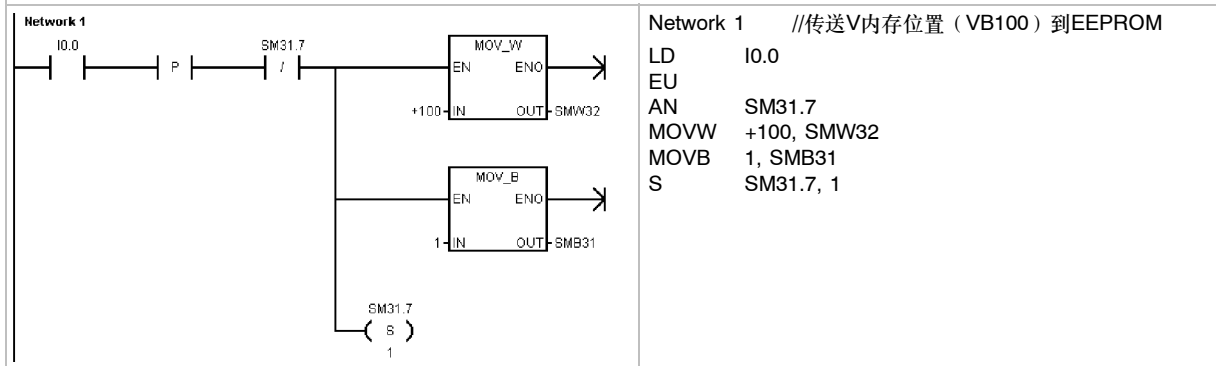


图4-20 SMB31和SMW32

不改变V内存中的数值直到保存操作完成。

#### 抽样程序：复制V内存到EEPROM

该实例传送VB100到EEPROM。在I0.0的上升边缘上，如果其他传送未在进行中，其载入要传送到SMW32的V内存位置的地址。其选择V内存的传送数量（1=字节；2=字；3=双字或实数）。然后设置SM31.7使S7-200在扫描结束时传送数据。传送完成后S7-200自动重设SM31.7。



## S7-200性能

S7-200提供多种允许用户自定义的特性以使S7-200功能更适合应用。

### S7-200允许程序对I/O立即进行读或写

S7-200指令集提供指令能立即从实际I/O进行读取和写入。这些直接I/O指令允许直接存取实际输入或输出点，即使图象寄存器通常可用于I/O存取源或目标地址。

当使用立即指令来存取输入点时，相应的进程图像输入寄存器位置未修改。当使用立即指令存取输出点时，相应的进程图像输出寄存器位置同步更新。



#### 提示

S7-200操作模拟输入的读入为立即数据，除非启用模拟输入过滤器。当写入数值到模拟输出，输出立即更新。

通常，在执行程序时使用进程图像寄存器胜于直接存取输入或输出。有三个原因使用图象寄存器：

- 所有扫描启动时输入的范例同步并冻结扫描循环程序执行阶段的输入数值。程序执行完成后输出从图象寄存器更新。可以在系统中提供稳定的效果。
- 比起存取I/O点，程序可以存取图象寄存器要快得多，允许程序更快的执行。
- I/O点是位实体而且必须按位或字节存取，但是可以按位、字节、字或双字存取图象寄存器。因此，图象寄存器提供其它的灵活性。

### S7-200允许程序中断扫描循环

如果使用中断，关联于每个中断事件的例行程序存储为程序的一部分。中断例行程序不作为普通扫描循环的一部分来执行，但是当中断事件产生时执行（可能在扫描循环中的任意点）。

S7-200的中断按各自的优先级分配以先来先服务原则服务。关于更多信息，请参见第6章中的中断指令

## S7-200允许用户为通讯任务分配进程时间

可以配置扫描循环的百分比专用于处理关于RUN（运行）模式编辑编译或执行状态的通讯请求。（RUN（运行）模式编辑和执行状态是STEP 7-Micro/WIN提供的选项，这样能更简单的调试程序。）当增加专用于处理通讯请求时间的百分比，就会增加扫描时间，将使控制进程运行更缓慢。

专门用于处理通讯请求的默认扫描时间百分比被设为10%。该设置为处理编译和状态操作同时尽量减小对控制进程的影响提供合理的调节。可以调整该设置，每次增量为5%，最大为50%。为后台通讯设置扫描循环时间段：

1. 选择视图（**View**）> 组件（**Component**）> 系统块（**System Block**）菜单命令并单击“背景时间”标签。
2. 编辑通讯背景时间属性并单击“确定”。
3. 下载修改的系统块到S7-200。



图4-21 通讯后台时间

## S7-200允许设置Stop（停止）模式数字输出的状态

S7-200的输出表允许确定是否在转换为STOP（停止）模式时将数字输出点状态设为已知的数值，或者在转换为STOP（停止）模式之前使输出保持在以前的状态。输出表是下载的系统块的一部分，被存储在S7-200中，仅应用于数字输出。

1. 选择视图（**View**）> 组件（**Component**）> 系统块（**System Block**）菜单命令并单击“输出表”标签。
2. 要在最后状态冻结输出，选择冻结输出复选框。
3. 要复制表数值到输出，在“运行至停止”转换之后，对每个想要设置为“接通”（1）的输出位通过单击复选框输入输出表数值。（表的所有默认值都是零。）
4. 单击“确定”保存选项。
5. 下载修改的系统块到S7-200。

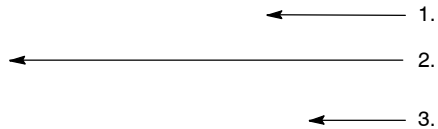


图4-22 配置输出表

## S7-200允许定义断电时保持内存

至多可以定义6个保留范围用来选择在电源循环时想要保留的的内存区域。可以在下列要保留的内存区中定义地址范围：V、M、C和T。对于计时器，只有保留计时器（TONR）可以保持。M内存的先前14个字节的缺省设置为非保留。

只能保持计时器和计数器的当前值：不保留计时器和计数器位。



### 提示

将范围MB0至MB13更改为保留范围会启用一种特殊功能，该功能会在断电时自动将这些位置保存在EEPROM中。

定义保留内存:

1. 选择视图 (**View**) > 组件 (**Component**) > 系统块 (**System Block**) 菜单命令并单击“保留范围”标签。 ← 1.
2. 选择在断电时要保持的内存范围并单击“确定”。 ← 2.
3. 下载修改的系统块到S7-200。

图4-23 保留内存

## S7-200允许过滤数字输入

对于一些或全部局部数字输入点，S7-200允许选择输入过滤器以定义延迟时间（选择范围从0.2毫秒至12.8毫秒）。该延迟有助于过滤输入布线上可能对输入状态造成不良改动的噪音。

输出过滤器是下载的系统块的一部分，存储在S7-200中。默认过滤器时间为6.4毫秒。如图4-24所示，每个延迟规定适用于输入点群组。 ← 1.

为输入过滤器配置延迟时间:

1. 选择视图 (**View**) > 组件 (**Component**) > 系统块 (**System Block**) 菜单命令并单击“输入过滤”标签。 ← 2.
2. 为每个输入组输入延迟量并单击“确定”。
3. 下载修改的系统块到S7-200。

图4-24 配置输入过滤器。



### 提示

当指令读取、输入中断和脉冲捕获时，数字输入过滤器影响输入值。根据过滤器选项，程序会错过中断事件或脉冲捕获。高速计数器记录未过滤输入中的事件数目。

## S7-200允许过滤模拟输入

S7-200允许选择单个模拟输入上的软件过滤。过滤数值是模拟输入的样本预选数的平均值。过滤器规格（样本和死区数目）对所有启用过滤功能的模拟输入均相同。

过滤器具有快速应答功能，允许过滤数值快速反映出较大的改动。当输入超出对平均值的指定改动时，过滤器对最新模拟输入值进行步骤功能改动。这一改动被称为死区，用模拟输入数字值计数表示。

默认配置对所有模拟输入启用过滤。

1. □ 选择视图 (**View**) > 组件 (**Component**) > 系统块 (**System Block**) 菜单命令并单击模拟输入过滤器标签。 ← 1.
2. 选择想要过滤的模拟输入、样本数和死区。 ← 2.
3. 单击“确定”。
4. 下载修改的系统块到S7-200。

图4-25 模拟输入过滤器



### 提示

不要对通过数字信息或以模拟字进行报警指示的模块使用模拟过滤器。始终禁用RTD模拟过滤、热电偶和AS接口主模块。

## S7-200允许捕捉短时脉冲

S7-200提供脉冲捕捉功能，它可用于一些或全部局部数字输入点。脉冲捕捉功能允许捕捉高位脉冲或低位脉冲，此类脉冲出现的时间极短，S7-200在扫描循环开始处读取数字输入时，可能无法一直看到此类脉冲。当为某一输入启用脉冲捕捉时，输入状态改动被锁定，并保持至下一次输入循环更新。这样可确保持续时间很短的脉冲被捕捉，并保持至S7-200读取输入。

对于每个局部数字输入，可以单个地启用脉冲捕捉操作。

要存取脉冲捕捉配置画面：

1. □ 选择视图 (**View**) > 组件 (**Component**) > 系统块 (**System Block**) 菜单命令并单击脉冲捕捉位标签。 ← 1.
2. 单击相应的复选框并单击“确定”。 ← 2.
3. 下载修改的系统块到S7-200。

图4-26 脉冲捕捉

图4-27显示S7-200启用和不启用脉冲捕获的基本操作。

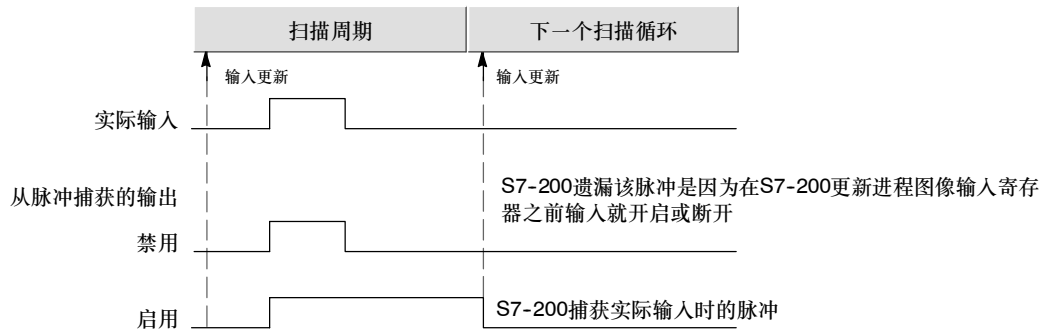


图4-27 在脉冲捕获功能启用或禁用下S7-200的运行

由于脉冲捕获功能在输入通过输入过滤器后运行，必须调整输入过滤器时间，以防过滤器移除脉冲。图4-28显示数字输入线路块图。

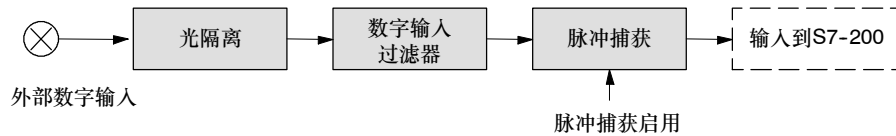


图4-28 数字输入线路

图4-29显示各种输入条件对启用脉冲捕获功能的响应。如果在某一给定扫描中存在一个以上脉冲，仅读取第一个脉冲。如果在某一给定扫描中有多个脉冲，则应当使用上升/下降边缘中断事件。（对于中断事件列表，请参见表6-45。）

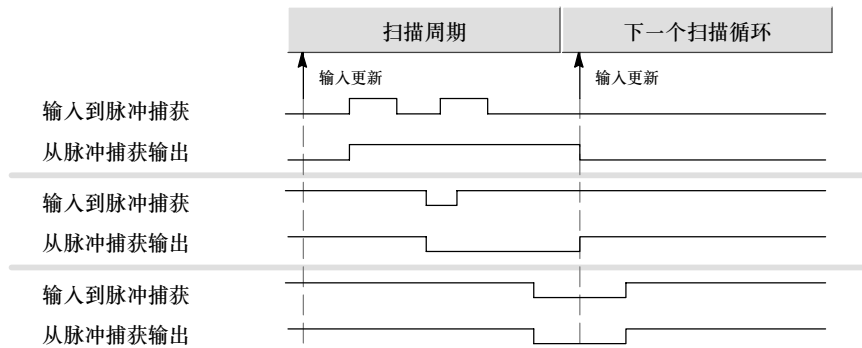


图4-29 脉冲捕获功能的响应到各种输入条件

## S7-200提供密码保护

S7-200的所有型号均提供密码保护，用于指定功能的限制存取。

密码授权存取功能和内存：不设密码的话，S7-200提供无限制的存取。当处于密码保护时，S7-200根据安装密码时的配置限制所有相关受限操作。

密码不区分大小写。

如表4-3所示，S7-200提供三种存取限制级别。每个级别均允许存取某些功能而不需密码。对于所有三种存取级别，输入正确的密码能提供所有功能的存取。S7-200的默认条件为级别1（无限制）。

通过网络输入密码不会泄露S7-200的密码保护。

表4-3 限制对S7-200的存取

CPU功能	等级1	等级2	等级3
读和写用户数据	存取允许	存取允许	存取允许
启动、停止和重新启动CPU			
读和写日历时钟			
上载用户程序、数据和CPU配置	存取允许	存取允许	需要密码
下载到CPU	存取允许	需要密码	
获得执行状态			
删除程序块、数据块或系统块			
强制数据或执行单重/多重扫描			
复制到内存磁带			
在STOP（停止）模式写入输出			

授权一个用户存取限制功能并不授权其他用户存取这些功能。同一时间只允许有一个用户无限制存取S7-200。



### 提示

输入密码后，当编程设备从S7-200上断开后，该密码的授权级别最多可保持一分钟有效时间。在断开电缆之前一定要退出STEP 7-Micro/WIN，以防另一位用户访问编程设备特权。

### 为S7-200配置密码

系统块对话框（图4-30）允许配置S7-200密码：

1. 选择视图（**View**）> 组件（**Component**）> 系统块（**System Block**）菜单命令来显示系统块对话框并单击密码标签。
2. 为S7-200选择合适的存取层。
3. 输入和校验密码。
4. 单击“确定”。
5. 下载修改的系统块到S7-200。

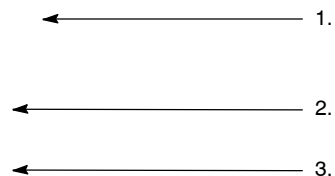


图4-30 创建口令



### 丢失密码情况下的恢复

如果忘记密码，必须清除S7-200内存并重新装载程序。要清除内存将S7-200置于STOP（停止）模式并复位S7-200到现场默认设置，除了网络地址、波特率和日历时钟。清除S7-200中的程序

1. 选择**PLC > 清除 (Clear)** 菜单命令显示清除对话框。
2. 选择全部三个块并通过单击“确定”确认操作。
3. 如果密码已配置，STEP 7-Micro/WIN显示密码授权对话框。欲清除密码，在密码授权对话框中输入CLEARPLC，继续执行“全部清除”操作。（CLEARPLC密码不区分大小写。）

“全部清除”操作不从内存磁盘中移除程序。因为内存磁盘中存储密码和程序，必须重新对内存磁带进行编程，才能移除丢失的密码。



#### 警告

清除S7-200内存将引起输出断开（或在模拟输出的情况下，在指定值处冻结）。

如果在清除内存时S7-200连接到设备，输出状态的改变可传输到设备。如果配置的输出的“安全状态”不同于工厂设置，输出的改变会引起设备不可知的运作，可能会导致死亡或严重的人身伤害，和/或损坏设备。

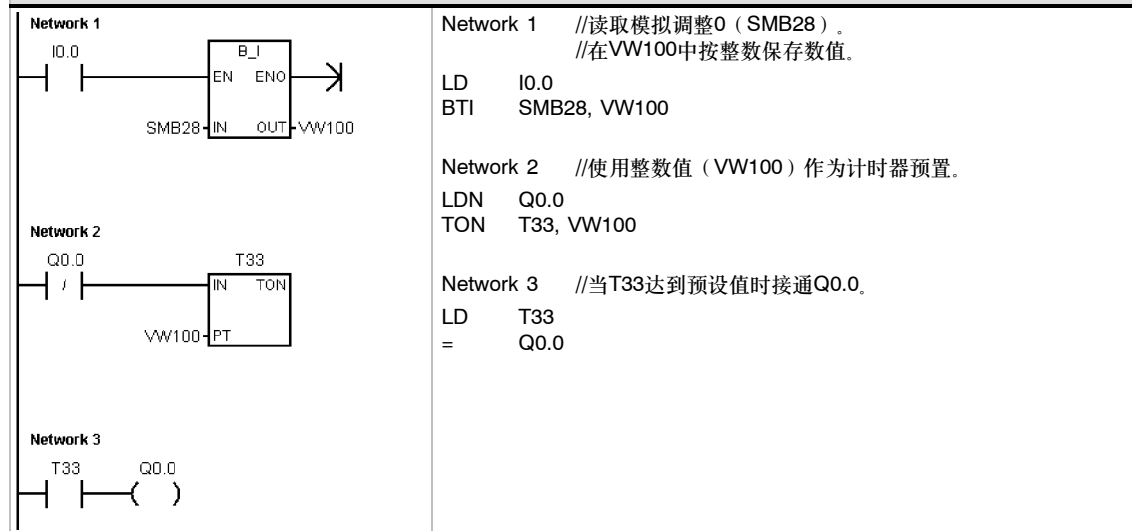
始终依照正确的安全预防措施并确保在清除S7-200内存之前进程处于安全状态。

### S7-200提供模拟调整电位器

模拟调整电位器位于模块前部存取盖下。可以调整这些电位器来增加或减少按特殊内存（SMB）的字节存储的数值。这些只读数值由程序用于各种功能，例如更新计时器或计数器的当前值，输入或更换预设值，或设置限制。使用小螺丝起子来调整：顺时针调节电位器（向右）来增加数值，而逆时针（向左）来减少数值。

SMB28保持表示模拟调整0位置的数字值。SMB29保持表示模拟调整1位置的数字值。模拟调整有一个0至255的额定范围，以及±2计数的重合性。

#### 使用模拟调整电位器引用输入数值的抽样程序



## S7-200提供高速I/O

### 高速计数器

S7-200提供集成高速计数器功能，能计数高速外部事件而不影响S7-200的性能。关于CPU型号所支持的比率，请参见附录 A。每台计数器对支持此类功能的时钟、方向控制、重设和起始均有专用输入。对于各种计数率，可以选择不同的求积模式。关于高速计数器的更多信息，请参见第6章。

### 高速脉冲输出

S7-200支持高速脉冲输出，以输出Q0.0和Q0.1生成高速脉冲链输出（PTO）或脉冲宽度调制（PWM）。

PTO功能为指定的脉冲数（从1至4,294,967,295脉冲）和指定的周期时间（以微秒或毫秒为增量，从50微秒至65,535微秒或从2毫秒至65,535毫秒）提供方波（50%工作循环）输出。用户可以执行PTO功能来制造单脉冲链或由多脉冲链组成的脉冲概要图。例如，通过简单的斜坡向上、运行和斜坡向下顺序或更复杂的顺序，可以使用脉冲概要图来控制步进电机。脉冲概要图最多可包含255个段，其中一个段与斜坡向上、运行或斜坡向下操作相对应。

PWM功能以变量工作循环输出，周期时间及以按微秒或毫秒增量指定的脉冲宽度提供一个固定的周期时间。周期时间从50秒至65,535秒或从2毫秒至65,535毫秒。脉冲宽度时间从0秒至65,535秒或从0毫秒至65,535毫秒。当脉冲宽度等于周期时间，工作循环为100%且输出持续接通。当脉冲宽度为零，工作循环则是0%且输出断开。

关于高速脉冲输出指令的更多信息，请参见第6章。

# 5

## 编程原理、约定和特点

S7-200将连续不断地执行程序，以控制任务或进程。可使用STEP 7-Micro/WIN创建程序，并将其下载到S7-200。STEP 7-Micro/WIN提供了各种各样的工具和手段，用于程序的设计、执行和调试。

### 本章内容

Micro PLC系统设计指南 .....	48
程序的基本单元 .....	49
使用STEP 7-Micro/WIN创建程序 .....	51
SIMATIC和IEC 1131-3指令集的选择 .....	53
了解程序编辑器所使用的约定 .....	54
使用向导来帮助创建控制程序 .....	56
处理S7-200中的错误 .....	56
数据块编辑器中的地址和初始值的分配 .....	58
使用符号表对变量进行符号编址 .....	58
使用局部变量 .....	59
使用状态图监控程序 .....	59
创建指令库 .....	60
用于程序调试的特性 .....	60

## Micro PLC系统设计指南

存在许多设计Micro PLC系统的方法。下列常用指南可用于许多设计项目。当然，必须遵守用户所在公司的操作规程以及用户自己的培训和特定场合下的公认准则。

### 划分进程或机械设备

将进程或机械设备划分为多个部分，它们具有相互独立的层次。这些分区确定了控制器之间的边界，并将影响资源的功能描述规范和分配。

### 创建功能规范

写入进程或机械设备的每个部分的操作描述。包括下列主题：I/O点、操作的功能描述、在允许每个执行机构（例如继电器、电机和驱动器）动作之前必须达到的状态、操作员接口的描述以及与进程或机械设备其它部分的任何接口。

### 设计安全电路

识别设备的安全性需要有物理连接的逻辑电路。控制设备在危险操作方式下可能出现故障，产生意外启动或切换机器设备的操作。此时，机器设备的意外或错误操作将可能导致人员身体伤害或重大财产损失，因此，应考虑使用独立于S7-200的机电式代用装置，以避免发生危险操作。所设计的安全电路应具有下列功能：

- 识别可能具有危险的执行机构误操作或意外操作。
- 识别可确保操作不具危险性的条件，并确定如何检测独立于S7-200的这些条件。
- 识别S7-200 CPU和I/O在电源接通和断开以及检测到错误时，将如何影响进程。这些信息应只用于设计正常和意料之中的异常操作，安全性不应用其来保证。
- 设计可阻断危险操作的独立于S7-200的手动或机电式安全装置。
- 提供独立电路的相应状态信息给S7-200，以便程序和任意操作员接口具有必需的信息。
- 识别用于进程安全操作的任何其它与安全有关的要求。

### 指定操作站

基于功能规范的要求，可创建操作站的简图。包括有下列条目：

- 显示每个操作站关于进程或机械设备位置的总览
- 用于操作站的设备的机械布局，例如显示器、开关和光源
- 具有S7-200 CPU相关I/O或扩充模块的电气图

## 创建配置图

以功能规范的要求为基础，创建控制设备的配置图。包括下列条目：

- 显示每个S7-200关于进程或机械设备位置的总览
- S7-200和扩展I/O模块的机械布局（包括机柜和其它设备）
- 每个S7-200和扩展I/O模块的电气图（包括设备型号、通讯地址和I/O地址）

## 创建符号名列表（可选）

如果选择使用符号名用于编址，则可创建一个用于绝对地址的符号名列表。它不仅包括物理I/O信号，也包括将在程序中使用的其它单元。

## 程序的基本单元

程序块由可执行代码和注释组成。可执行代码由主程序和任意子程序或中断程序组成。代码将被编译，并下载给S7-200；程序注释则不进行编译和下载。使用组织单元（主程序、子程序和中断程序）可构建控制程序。

下列实例将说明包含有子程序和中断程序的程序。该范例程序使用了一个定时中断，用于每隔100毫秒读取一个模拟输入的数值。

实例：程序的基本单元		
M A I N	<p>Network 1</p>	<p>Network 1 //一旦首先扫描，则调用子例行程序0。</p> <pre>LD SM0.1 CALL SBR_0</pre>
S B R 0	<p>Network 1</p>	<p>Network 1 //将周期设置为100毫秒用于定时中断。 //启用中断0。</p> <pre>LD SM0.0 MOVB 100, SMB34 ATCH INT_0, 10 ENI</pre>
I N T 0	<p>Network 1</p>	<p>Network 1 //对模拟输入4进行采样。</p> <pre>LD SM0.0 MOVW AIW4, VW100</pre>

## 主程序

程序的主体包含有控制应用的指令。S7-200将按顺序执行这些指令，每个扫描周期执行一次。主程序也可引用为OB1。

## 子程序

程序的这些可选单元只有在被下列程序调用时才执行：主程序、中断程序或另一个子程序。在希望重复执行某个函数时，可使用子程序。与其在希望使用函数的主程序中的各处重复编写函数的逻辑程序，不如在子程序中编写一次函数逻辑程序，然后根据主程序中的需要多次调用子例行程序。子程序具有多个好处：

- 使用子程序可减少整个程序的大小。
- 使用子程序可缩短扫描时间，因为已经将代码移出主程序。无论代码执行与否，S7-200都将在每个扫描周期内对代码进行计算，但只有在调用子例行程序时才对子程序中的代码进行计算，并且S7-200在不调用子例行程序的扫描期间，将不计算代码。
- 使用子程序可创建可移植代码。可将子程序中的函数代码隔离，然后几乎或完全不用作任何重复工作，就可将该子程序复制到其它程序中。



### 提示

使用V内存地址可限制子程序的可移植性，因为一个程序中的V内存地址分配有可能与另一个程序中的分配发生冲突。相反，使用用于所有地址分配的局部变量表（L内存）将具有更高的可移植性，因为在使用局部变量时完全不用担心子程序与程序的另一部分发生地址冲突。

## 中断程序

程序中的这些可选单元将对指定的中断事件作出反应。可设计一个中断程序，处理预定义的中断事件。只要指定的事件一发生，S7-200就将执行中断程序。

主程序将不调用中断程序。可将中断程序与中断事件关联起来，并且只有在每次发生中断事件时，S7-200才执行中断程序中的指令。



### 提示

因为不可能预知S7-200何时会产生中断，所以对中断程序和程序中的其他部分都要用到的变量的数目加以限制是有必要的。

使用中断程序的局部变量表，可确保中断程序只使用临时内存，而不会覆盖将在程序其它地方使用的数据。

存在可使用的多种编程技术，都可确保数据在主程序和中断程序之间正确共享。这些技术的有关描述参见第6章中的中断指令。

## 程序的其它单元

程序的其它块包括了用于S7-200的信息。当下载程序时可选择下载这些块。



系统块

### 系统块

系统块允许配置用于S7-200的不同硬件选项。



数据块

### 数据块

数据块将存储程序所使用的不同变量（V内存）的数值。可使用数据块来输入数据的初始值。

## 使用STEP 7-Micro/WIN创建程序

为接通STEP 7-Micro/WIN, 可双击STEP 7-Micro/WIN图标, 或选择开始 (Start) > SIMATIC > STEP 7 MicroWIN 3.2菜单命令。如图5-1所示, STEP 7-Micro/WIN项目窗口将提供用于创建控制程序的便利工作空间。

工具栏将提供快捷按钮, 用于经常使用的菜单命令。可显示或隐藏工具栏的任何按钮。

浏览条给出了多组图标, 用于访问STEP 7-Micro/WIN的不同编程特性。

指令树将显示用于创建控制程序的所有项目对象和指令。可将单个的指令从指令树拖放到程序中, 或双击某个指令, 以便将其插入到程序编辑器中光标的当前位置。

程序编辑器包括程序逻辑和局部变量表, 可在其中分配临时局部变量的符号名。子程序和中断程序在程序编辑器窗口的底部均按标签显示。单击标签可在子程序、中断程序和主程序之间来回变换。

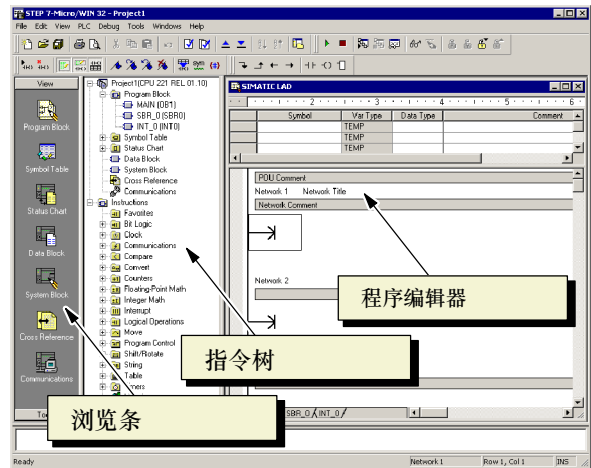


图5-1 STEP 7-Micro/WIN



程序编辑器

STEP 7-Micro/WIN提供了用于创建程序的三个编辑器: 梯形图 (LAD)、语句表 (STL) 和功能块图 (FBD)。尽管有某些限制, 在这些程序编辑器的任何一个中编写的程序均可用其它程序编辑器进行浏览和编辑。

### STL编辑器的特点

STL编辑器可将程序显示为基于文本的语言。STL编辑器允许通过输入指令助记符来创建控制程序。另一方面, STL编辑器还允许创建LAD或FBD编辑器所不能创建的程序。这是因为您是在S7-200的本土语言下进行编程, 而不是在图形编辑器中编程, 在图形编辑器中必须加以某些限制, 以便正确进行绘图。如图5-2所示, 这种基于文本的概念十分类似于汇编语言程序。

S7-200将按照程序中所规定的次序执行每条指令, 首先从上到下, 然后, 重新从上开始。

STL使用了逻辑堆栈以便处理控制逻辑。可插入处理堆栈操作的STL指令。

```
LD    I0.0      //读入一个输入
A     I0.1      //与另一个输入进行“与”运算
=     Q1.0      //将数值写入输出1
```

图5-2 范例STL程序

在选择STL编辑器时, 请考虑这些要点:

- STL最适合于有经验的程序员使用。
- 在某些情况下, STL允许处理较难使用LAD或FBD编辑器进行处理的问题。
- 可只使用具有SIMATIC指令集的STL编辑器。
- 虽然总可以使用STL编辑器来浏览或编辑用LAD或FBD编辑器所创建的程序, 但反之则不一定正确。并非始终都能使用LAD或FBD显示由STL编辑器所编写的程序。

## LAD编辑器的特点

LAD编辑器可将程序显示为类似于电气导线图的图形形式。使用梯形程序将能够通过一系列的逻辑输入条件（从而依次激活输出条件）来模拟电源电流的流向。梯形程序包括已通电的左电源导轨。已断开的触点将允许电流通过它们流向下一个单元，而已接通的触点将中断电流的流动。

可将逻辑电路分隔为多个网络。本程序将按照程序规定，从左到右，然后从上到下地一次执行一段程序。图5-3显示了梯形程序的实例。各种不同的指令均由图形符号来表示，并包含有三种基本形式。

触点代表逻辑输入条件，例如开关、按钮或内部条件。

线圈通常代表逻辑输出结果，例如指示灯、电动起动机、中介继电器或内部输出条件。

方框代表附加的指令，例如，定时器、计数器或数学指令。

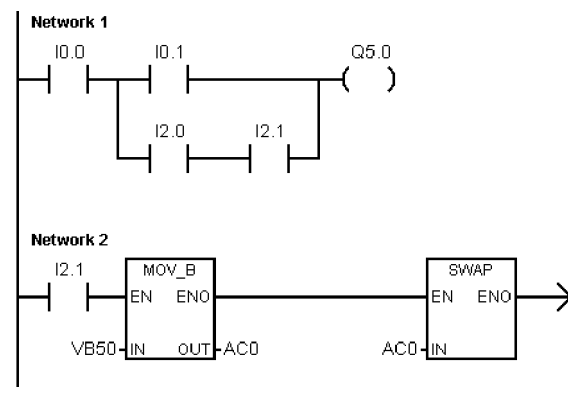


图5-3 梯形程序范例

在选择LAD编辑器时，请考虑这些要点：

- 梯形图适合于初级程序员使用。
- 采用图形表达方式便于理解和在全世界通用。
- SIMATIC和IEC 1131-3指令集二者均适用于LAD编辑器。
- 也可始终使用STL编辑器来显示SIMATIC LAD编辑器所创建的程序。

## FBD编辑器的特点

FBD编辑器可将程序显示为图形表达式，它类似于通用的逻辑门图。在LAD编辑器中不存在任何触点和线圈，但存在显示为方框指令的等效指令。

图5-4显示了FBD程序的实例。

FBD不使用左右电源导轨的概念；因此，术语“功率流”将用来表示通过FBD逻辑块的控制流的类似概念。

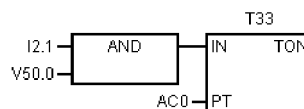


图5-4 范例FBD程序

经由FBD单元的逻辑“1”路径将称作功率流。功率流输入的起始点以及功率流输出的目标点均直接分配给操作数。

程序逻辑将源于这些方框指令之间的连接。也就是说，可使用某个指令（例如“与”方框）的输出来激活另一个指令（例如计时器），以便创建必需的控制逻辑。这种连接概念允许解决各种逻辑问题。

在选择FBD编辑器时，请考虑这些要点：

- 图形逻辑门风格的表达方式更适合下面的程序流程。
- SIMATIC和IEC 1131-3指令集二者均适用于FBD编辑器。
- 也可始终使用STL编辑器来显示SIMATIC FBD编辑器所创建的程序。



## SIMATIC和IEC 1131-3指令集的选择

大多数的PLC都提供了相似的基本指令，但是，不同厂家的产品之间在外观、操作等方面通常仍然存在一些差别。在过去的几年中，国际电工技术委员会（IEC）已经开发了一种针对PLC编程的众多方面的新兴全球标准。该标准鼓励不同的PLC生产商提供在外观和操作两方面都相同的指令。

S7-200提供了两个指令集，以解决各种自动化任务。IEC指令集符合PLC编程的IEC 1131-3标准，而SIMATIC指令集则专为S7-200设计。



### 提示

当将STEP 7-Micro/WIN设置为IEC模式时，它在未按IEC 1131-3标准定义的指令旁的指令树中，将显示为红色菱形(◆)。

在SIMATIC指令集与IEC指令集之间存在一些重要区别：

- ❑ IEC指令集仅限于那些属于PLC厂商之间标准的指令。通常，SIMATIC指令集中所包含的一些指令将不是IEC 1131-3规范中的标准指令。这些指令仍然可以用作非标准指令，只是一旦使用了这些指令，程序就不再严格与IEC 1131-3兼容。
- ❑ 某些IEC方框指令可接受多种数据格式。这种应用经常在超载时使用。例如，虽然没有独立的ADD\_I（添加整型）和ADD\_R（添加实型）、数学方框，但IEC ADD指令将检验所添加数据的格式，并自动选择S7-200中的正确指令。这可节省宝贵的程序设计时间。
- ❑ 在使用IEC指令时，将自动检查指令参数特有的数据格式，比如说有符号整型与无符号整型。例如，如果试图为期望是“位”数值（开/关）的指令输入一个整型数值，则将导致出现错误。该特性有助于最大限度地减少编程的语法错误。

在选择SIMATIC或IEC指令集时，请注意以下要点：

- ❑ SIMATIC指令通常具有最短的执行时间。某些IEC指令可能要花费较长的执行时间。
- ❑ 某些IEC指令的操作，例如定时器、计数器、乘法器以及除法器，将与其SIMATIC对应指令不同。
- ❑ 可使用具有SIMATIC指令集的三种程序编辑器（LAD、STL、FBD）。也可只使用用于IEC指令的LAD和FBD程序编辑器。
- ❑ IEC运算指令是各种不同品牌PLC的标准，且可通过PLC平台影响有关创建符合IEC的程序的经验。
- ❑ 当IEC标准定义的指令少于SIMATIC指令集中的可用指令时，可始终将SIMATIC指令包含在IEC程序中。
- ❑ IEC 1131-3指定必须为变量声明一个类型，并支持系统对数据类型进行检查。

## 了解程序编辑器所使用的约定

STEP 7-Micro/WIN在所有程序编辑器中使用了下列约定:

- 符号名 (#var1) 前的#表示符号的作用范围为局部。
- 对于IEC指令, %符号表示直接地址。
- 操作数符号 “?.?” 或 “????” 表示需要操作数配置。

LAD程序均被划分为称作程序段的多个段。程序段就是按一定次序排列的触点、线圈和方框, 将它们都连接起来, 就构成了一个完整的回路: 任何短路的回路、断开的回路以及反向的功率流条件都不存在。STEP 7-Micro/WIN允许创建基于程序段节点的LAD程序注释。FBD编程将使用用于对程序进行分段和注释的程序段概念。

STL程序不使用程序段; 然而, 可使用“程序段”关键字对程序进行分段。

### 面向LAD编辑器的约定

在LAD编辑器中, 可使用键盘上的F4、F6和F9键来访问触点、方框和线圈等指令。LAD编辑器使用了下列约定:

- 符号 “--->>” 表示一个断开的回路或所需要的功率流连接。
- 符号 “→” 表示输出是指令的一个可选的功率流, 它们可层叠或串联在一起。
- 符号 “>>” 表示可使用功率流。

### 面向FBD编辑器的约定

在FBD编辑器中, 可使用键盘上的F4、F6和F9键来访问“与”、“或”以及方框等指令。FBD编辑器使用了下列约定:

- EN操作数上的符号 “--->>” 是功率流或操作数指示符。也可将其描述为是断开的回路或所需要的功率流连接。
- 符号 “→” 表示该输出是指令的一种可选的功率流, 它们可层叠或串联在一起。
- 符号 “<<” 和 “>>” 表示既可以使用数值, 也可以使用功率流。
- “非”泡: 逻辑“非”条件或操作数与功率流的反向条件在输入中都用小圆圈来表示。在图5-5中, Q0.0等同於I0.0“与”I0.1的“非”。“非”泡仅适用于布尔型信号, 可将它们指定为参数或功率流。

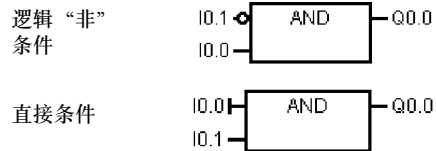


图5-5 FBD约定

- 直接指示符: 如图5-5所示, FBD编辑器显示布尔型操作数的直接条件, 它在FBD指令的输入上有一个垂直行。直接指示符将引起特定物理输入的直接读取。直接操作符仅适用于物理输入。
- 没有任何输入或输出的方框: 没有任何输入的方框表示指令与功率流无关。



#### 提示

操作数的数目最多可扩展到“与”和“或”指令的32个输入。为添加或减少操作数数量, 可使用键盘上的“+”和“-”键。

## S7-200编程的常规约定

### EN/ENO定义

EN（激活输入）是用于LAD和FBD中方框的布尔型输入。功率流必须显示在所要执行的方框指令的该输入中。在STL中，指令没有EN输入，但堆栈数值的最顶端必须为所要执行的相应STL指令的逻辑“1”。

ENO（激活输出）是用于LAD和FBD中方框的布尔型输出。如果方框在EN输入处具有功率流，且方框执行其功能时没有错误，则ENO输出将把功率流传递到下一个单元。如果在方框的执行中检测到错误，则在导致错误的方框处，功率流将中断。

在STL中，不存在任何ENO输出，但与具有ENO输出的LAD和FBD指令相对应的STL指令一定要设置一个特殊的ENO位。该位可用AND ENO（AENO）指令来存取，并可用于生成和方框的ENO位一样的效果。



#### 提示

EN/ENO操作数和数据类型均不在各个指令的有效操作数表中显示，因为操作数对于所有的LAD和FBD指令都是一样的。表5-1出了LAD和FBD的这些操作数和数据类型。这些操作数可应用于本手册所描述的所有LAD和FBD指令。

表5-1 LAD和FBD的EN/ENO操作数和数据类型

程序编辑器	输入 / 输出	操作数	数据类型
LAD	EN、ENO	功率流	布尔型
FBD	EN、ENO	I、Q、V、M、SM、S、T、C、L	布尔型

### 条件 / 无条件输入

在LAD和FBD中，与功率流有关的方框或线圈将用与左侧任意单元的连接来表示。与功率流无关的线圈或方框将用与左电源导轨的直接连接来表示。表5-2表示了条件和无条件输入的实例。

表5-2 条件和无条件输入的代表

功率流	LAD	FBD
与功率流有关的指令（有条件的）		
与功率流无关的指令（无条件的）		

### 无输出的指令

不能层叠的方框绘制时将不带任何布尔型输出。这些方框包含了子程序调用、跳转和条件返回指令。还存在只能放置在左电源导轨上的梯形图线圈。它们包括Label、Next、Load SCR、Conditional SCR End和SCR End等指令。它们在FBD中均显示为方框，并可用未经标记的电源输入进行区分，且没有任何输出。

### 比较指令

比较指令的执行与功率流的状态无关。如果功率流为假，则输出为假。如果功率流为真，则输出的设置将取决于比较的结果。SIMATIC FBD、IEC梯形图和IEC FBD等的比较指令均显示为方框，即使操作作为一个触点来完成。

## 使用向导以帮助创建控制程序

STEP 7-Micro/WIN提供了向导，以使编程的方式更简单、更轻松。在第6章中，具有相关向导的指令将用下列指令向导图标来标识：



指令向导

## 处理S7-200中的错误

S7-200将错误分类为严重错误或非严重错误。浏览因错误而生成的错误代码，可通过选择**PLC > 信息 (Information)** 菜单命令。

图5-6表示了显示错误代码和错误描述的PLC信息对话框。

“最后一个严重错误”域表示S7-200所生成的先前的严重错误代码。如果RAM的内容保留，则该数值在整个电源周期内保留。该单元的内容将清除，不管何时清除S7-200的所有内存，抑或在长时间停电之后丢失RAM的内容。

“严重错误的总数”域为S7-200所生成的严重错误的总数，因为上次S7-200的所有内存区域的内容均已清除。如果RAM的内容保留，则该数值在整个电源周期内保留。该单元的内容将清除，无论何时清除S7-200所有内存的内容，抑或在长时间停电之后丢失RAM的内容。

附录C列出了S7-200出错代码，而附录D描述了专用内存（SM）位，它们都可用于对错误进行监控。

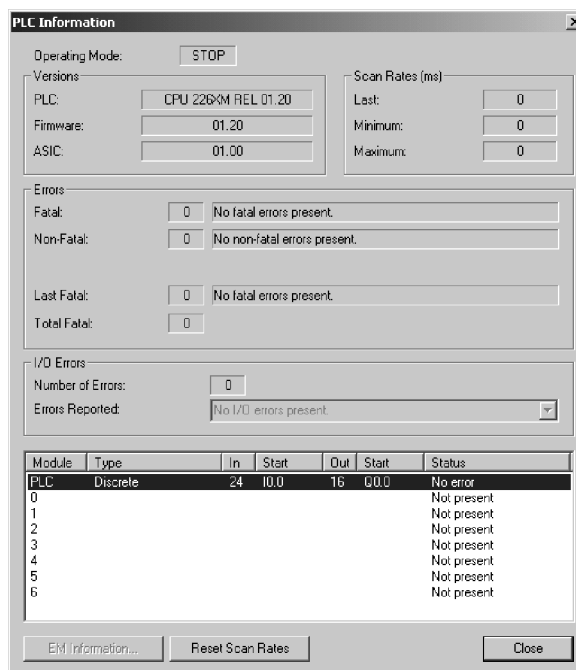


图5-6 PLC信息对话框

### 非严重错误

非严重错误就是那些在用户程序的构建、用户程序指令的执行以及扩展I/O模块中所出现的问题。可使用STEP 7-Micro/WIN来浏览由非严重错误所生成的错误代码。存在三种基本类别的非严重错误。

#### 程序编译错误

S7-200在下载程序时将程序进行编译。如果S7-200检测到程序与编译规则有冲突，则下载将中止，并生成出错代码。（已经下载到S7-200的程序将仍然存在于EEPROM中，不会丢失。）在更正程序之后，可重新开始下载。请参见关于编译规则冲突表的附录C。

## I/O错误

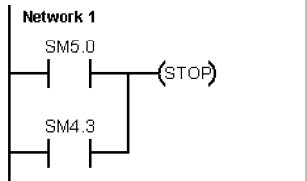
在启动时，S7-200将读取每个模块的I/O配置。在正常操作期间，S7-200将周期性地检查每个模块的状态，并将其与启动期间所获得的配置进行比较。如果S7-200检测到了差异，则S7-200将设置模块错误寄存器中的配置错误位。S7-200不从该模块中读取输入数据，也不输出数据写入到该模块，直到模块配置重新与启动时获得的配置相匹配。

模块状态信息将存储在专用内存（SM）位。程序可对这些位进行监控和计算。关于报告I/O错误所使用的SM位的更详细信息，请参见附录D。SM5.0是全局I/O错误位，且在扩充模块上存在错误条件时仍然保留。

## 程序执行错误

程序在执行时可产生错误条件。这些错误来源于指令的不恰当使用或指令对无效数据的处理。例如，在程序编译时有效的间接地址指针，在程序执行期间，有可能被修改到指向超出范围的地址。这就是运行系统编程问题的一个实例。一旦发生运行系统编程问题，就将设置SM4.3，且在S7-200处于RUN（运行）模式时仍然保留设置。（请参见关于运行系统编程问题表的附录C）。程序执行错误信息将存储在专用内存（SM）位。程序可对这些位进行监控和计算。关于报告程序执行错误的SM位的更详细信息，请参见附录D。

S7-200在检测到非严重错误时，将不切换到STOP（停止）模式。它只将事件记录在SM内存中，并继续执行程序。然而，可将程序设计为当检测到非严重错误时，强制S7-200切换到STOP（停止）模式。下列范例程序表示程序的一个段，该程序段对两个全局非严重错误位进行监控，且无论这两个位中的一个位何时接通，S7-200都将切换到STOP（停止）模式。

范例程序：用于检测非严重错误条件的逻辑	
	<p>Network 1 //当I/O错误或运行系统错误产生时，将跳转到STOP（停止）模式</p> <pre>LD    SM5.0 O     SM4.3 STOP</pre>

## 严重错误

严重错误将导致S7-200停止执行程序。根据严重错误的严重程度，它将致使S7-200失去执行某些或所有函数的能力。处理严重错误的目的是使S7-200恢复到安全状态，S7-200可据此对关于现有错误条件的询问作出反应。当检测到严重错误时，S7-200将切换到STOP（停止）模式，接通系统错误LED和STOP LED，清除输出表，并断开输出。S7-200此时将仍然保持原有状态，直到严重错误的条件得到更正。

一旦进行修改，更正了严重错误的条件，则可使用下列方法重新启动S7-200：

- 断开电源，然后重新接通。
- 将模式从RUN（运行）或TERM（终端）切换到STOP（停止）。
- 从STEP 7-Micro/WIN中选择PLC > 电源复位（Powerp Reset）菜单命令，以便重新启动S7-200。这将强制S7-200重新启动，并清除许多严重错误。

重新启动S7-200将清除严重错误的条件，并完成电源启动的诊断测试，以验证严重错误已经纠正。如果发现了另一个严重错误的条件，则S7-200又将设置故障LED，表示错误仍然存在。否则，S7-200将开始正常的操作。

某些错误条件将使S7-200失去通讯能力。此时，将不能浏览S7-200的错误代码。这些类型的错误将表示S7-200存在需要修复的硬件故障；它们将不能通过程序的切换或S7-200内存内容的清除来修理。

## 数据块编辑器中的地址和初始值的分配



数据块

数据块编辑器将允许只对V内存（变量内存）进行初始数据分配。可对V内存的字节、字或双字进行分配。注释则是可选的。

数据块编辑器是一种形式自由的文本编辑器；也就是说，没有为特定类型的信息定义任何具体的域。在完成一行的键入，并按下Enter键之后，数据块编辑器将对行进行格式化（对齐地址、数据、注释的列；将V内存地址大写），并重新对其进行显示。数据块编辑器将基于先前的地址分配和数据值的大小（字节、字或双字）来分配V内存的相应总量。

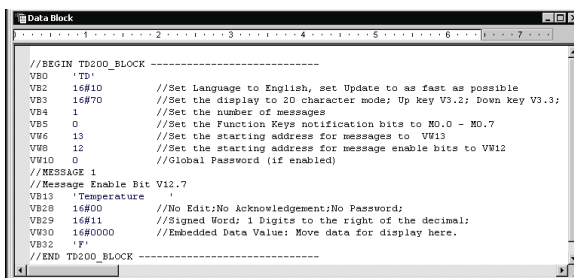


图5-7 数据块编辑器

数据块的第一行必须具有明确的地址分配。后续的行既可具有明确的地址分配，也可具有隐含的地址分配。当在单个地址分配之后键入多个数据值时，或键入只包含数据值的某一行时，编辑器将进行隐含地址分配。

数据块编辑器可接受大写或小写字母，并允许将逗号、制表符或空格用作地址和数据值之间的分隔符。

## 使用符号表对变量进行符号编址



符号表

符号表将允许定义和编辑那些可由程序任意位置处的符号名所访问的符号。可创建多种符号表。在用于系统定义符号的符号表中还存在一个可在程序中使用的标签。符号表也可引用为全局变量表。

可在程序中实际或用符号标识指令的操作数。实际引用可使用内存区以及位或字节单元来标识地址。符号引用可使用字母数字字符的组合来标识地址。

对于SIMATIC程序，通过采用符号表可进行全局符号分配。对于IEC程序，通过采用全局变量表可进行全局符号分配。

为了将符号分配给地址：

1. 单击浏览条中的符号表图标，以便打开符号表。
2. 在符号名称列中输入符号名称（例如Input1）。最大符号长度为23个字符。
3. 在地址列中输入地址（例如I0.0）。
4. 对于IEC全局变量表，在数据类型列中输入数值或从列表框中选择一个数值。

可创建多种符号表；然而，无论是在单个表内，还是在多个表中，都不能多次使用同样的字符串作为全局符号分配。

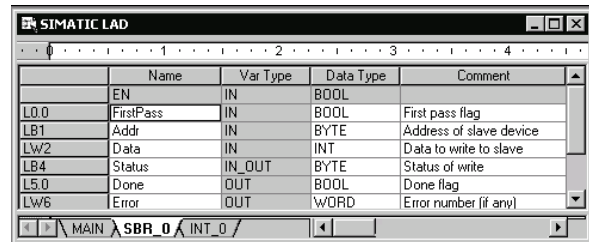
	Symbol	Address	Comment
1	AlwaysOn	SM0.0	Always on contact
2	Pump1	Q2.3	Pump 1 on/off
3	Pump1Limit	I1.1	Pump 1 pressure limit switch
4	Pump1Pressure	VD100	Pump 1 current pressure (real)
5	Pump1Rpm	VW200	Pump1 PRM: [integer]
6			

图5-8 符号表

## 使用局部变量

可使用程序编辑器的局部变量表来分配变量，这些变量对单个的子程序或中断程序都是唯一的。参见图5-9。

可将局部变量用作将要传递给子程序的参数，从而增加了子程序的可移植性或可重复使用性。



Name	Var Type	Data Type	Comment	
EN	IN	BOOL		
L0.0	FirstPass	IN	BOOL	First pass flag
LB1	Addr	IN	BYTE	Address of slave device
LW2	Data	IN	INT	Data to write to slave
LB4	Status	IN_OUT	BYTE	Status of write
L5.0	Done	OUT	BOOL	Done flag
LW6	Error	OUT	WORD	Error number (if any)

图5-9 局部变量表

## 使用状态图对程序进行监控



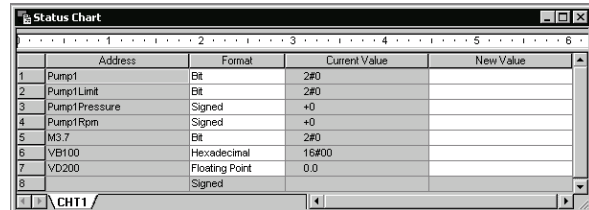
状态图

状态图允许在S7-200运行控制程序时能够监控或修改过程变量的数值。通过显示当前的数值，可对程序输入、输出或变量的状态进行跟踪。状态图还允许强制或修改过程变量的数值。

可创建多种状态图，以便对程序不同部分的单元进行浏览。

为了访问状态图，可选择视图（**View**）> 组件（**Component**）> 状态图（**Status Chart**）菜单命令或单击浏览条中的“状态图”图标。

在创建状态图时，可输入用于监控的过程变量的地址。不能浏览常数、累加器或局部变量的状态。计时器或计数器的数值可按位或字显示。将数值按位显示可表示计时器或计数器位的状态；将数值按字显示可表示计时器或计数器的数值。



Address	Format	Current Value	New Value
1 Pump1	Bit	2#0	
2 Pump1Limit	Bit	2#0	
3 Pump1Pressure	Signed	+0	
4 Pump1Rpm	Signed	+0	
5 M3.7	Bit	2#0	
6 VB100	Hexadecimal	16#00	
7 VD200	Floating Point	0.0	
8	Signed		

图5-10 状态图

为构造状态图和监控变量：

1. 在地址域中输入每个期望值的地址。
2. 在格式列中选择数据类型。
3. 为了在S7-200中浏览过程变量的状态，可选择调试（**Debug**）> 状态图（**Chart Status**）菜单命令。
4. 为连续采样数值，或完成状态的单个读操作，可单击工具栏上的按钮。状态图还允许修改或强制不同过程变量的数值。

通过选择编辑（**Edit**）> 插入（**Insert**）> 行（**Row**）菜单命令，可在状态图中插入附加的行。



### 提示

可创建多种状态图将变量划分为逻辑组，以便在更简短的独立状态图中浏览每个组。

## 创建指令库

STEP7-Micro/WIN既允许创建自定义的指令库，也允许使用由其他人创建的库。参见图5-11。

为了创建指令库，可创建标准的STEP 7-Micro/WIN子程序和中断程序，并将二者组合到一起。可隐藏这些子程序中的代码，以避免意外的修改或保护作者的技术（技术诀窍）。

为创建指令库，请完成下列任务：

1. 按标准STEP 7-Micro/WIN项目编写程序，并将库中所包含的函数放入子程序或中断程序中。
2. 请确保子程序或中断程序中的所有V内存单元都已经分配有符号名。为使库所需要的V内存的总量最小，可使用连续V内存单元。
3. 将子程序或中断程序重新命名为在指令库中所希望显示的名称。
4. 选择文件（**File**）> 创建库（**Create Library**）菜单命令来编译新的指令库。

有关库创建的详细信息，请参见STEP 7-Micro/WIN联机帮助。

请使用下列步骤来访问指令库中的指令：

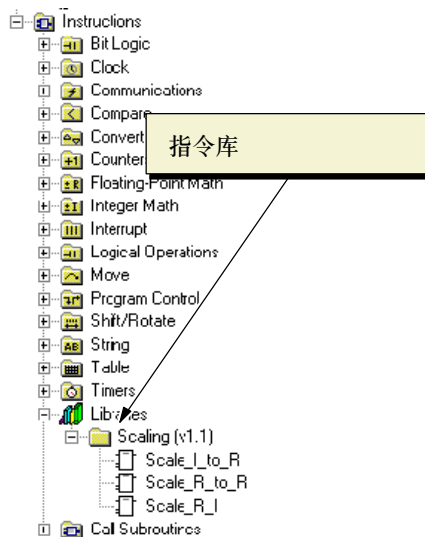


图5-11 具有库的指令树

1. 将库目录添加给指令树可通过选择文件（**File**）> 添加库（**Add Libraries**）菜单命令。
2. 选择指定的指令，并将其插入程序（当愿意将其用作任意标准指令时）。

如果库程序需要任意的V内存，则STEP 7-Micro/WIN在对项目进行编译时将提示分配一块内存。使用“库内存分配”对话框可分配多块内存。

## 调试程序的特点

STEP7-Micro/WIN提供了下列特点，用于帮助调试程序：

- 程序中的书签，将使长程序的行之间的来回移动十分方便。
- 交叉索引表，允许检查程序中所使用的引用。
- 运行模式编辑，允许以对程序所控制进程的最低限度的干扰来对程序进行少量修改。当在RUN（运行）模式下进行编辑时，也可下载程序块。

有关调试程序的更详细信息，请参见第8章。



# S7-200指令集

# 6

本章描述S7-200 Micro PLC的SIMATIC和IEC 1131指令集。

## 本章内容

用于描述指令的约定 .....	63
S7-200内存范围和特征 .....	64
位逻辑指令 .....	66
接点 .....	66
线圈 .....	69
逻辑堆栈指令 .....	71
设置和重设主要双稳态指令 .....	73
时钟指令 .....	74
通讯指令 .....	75
网络读取和网络写入指令 .....	75
发送和接收指令（空闲端口） .....	80
获得端口地址和设置端口地址指令 .....	89
比较指令 .....	90
比较数值 .....	90
比较字符串 .....	92
转换指令 .....	93
标准转换指令 .....	93
ASCII转换指令 .....	97
字符串转换指令 .....	101
编码和解码指令 .....	106
计数器指令 .....	107
SIMATIC计数器指令 .....	107
IEC计数器指令 .....	110
高速计数器指令 .....	112
脉冲输出指令 .....	126
运算指令 .....	141
加、减、乘和除指令 .....	141
乘以整数到双整数，除以整数带余数 .....	143
数字函数指令 .....	144
递增和递减指令 .....	145
比例 / 积分 / 微分（PID）环路指令 .....	146
中断指令 .....	156
逻辑操作指令 .....	163
反转指令 .....	163
AND（与）、OR（或）和Exclusive OR（异或）指令 .....	164
移动指令 .....	166
移动字节、字、双字或实数 .....	166
立即移动字节（读和写） .....	167
成块移动指令 .....	168

程序控制指令 .....	169
有条件结束 .....	169
停止 .....	169
监视程序重设 .....	169
For-Next循环指令 .....	171
跳转指令 .....	173
顺序控制继电器 (SCR) 指令 .....	174
移位与循环指令 .....	180
向右移位和向左移位指令 .....	180
向右循环和向左循环指令 .....	180
移位寄存器位指令 .....	182
交换字节指令 .....	184
字符串指令 .....	185
表指令 .....	190
添加到表格 .....	190
先入先出和后人先出 .....	191
内存填充 .....	193
表查找 .....	194
计时器指令 .....	197
SIMATIC计时器指令 .....	197
IEC计时器指令 .....	202
子例行程序指令 .....	204

## 用于描述指令的约定

图6-1显示典型的指令描述，并指向用于描述指令和其操作的不同区域。指令的图解以梯形图、功能块图表和语句表形式显示。操作数表列出指令的操作数并显示有效的数据类型、内存区域和每个操作数的大小。

EN/ENO操作数和数据类型没有显示在指令操作数表中，因为操作数对于所有梯形图和功能块图表指令是相同的。

- 对于梯形图: EN和ENO是功率流并且是布尔数据类型。
- 对于功能块图表: EN和ENO是I、Q、V、M、SM、S、T、C、L或功率流并且是布尔数据类型。

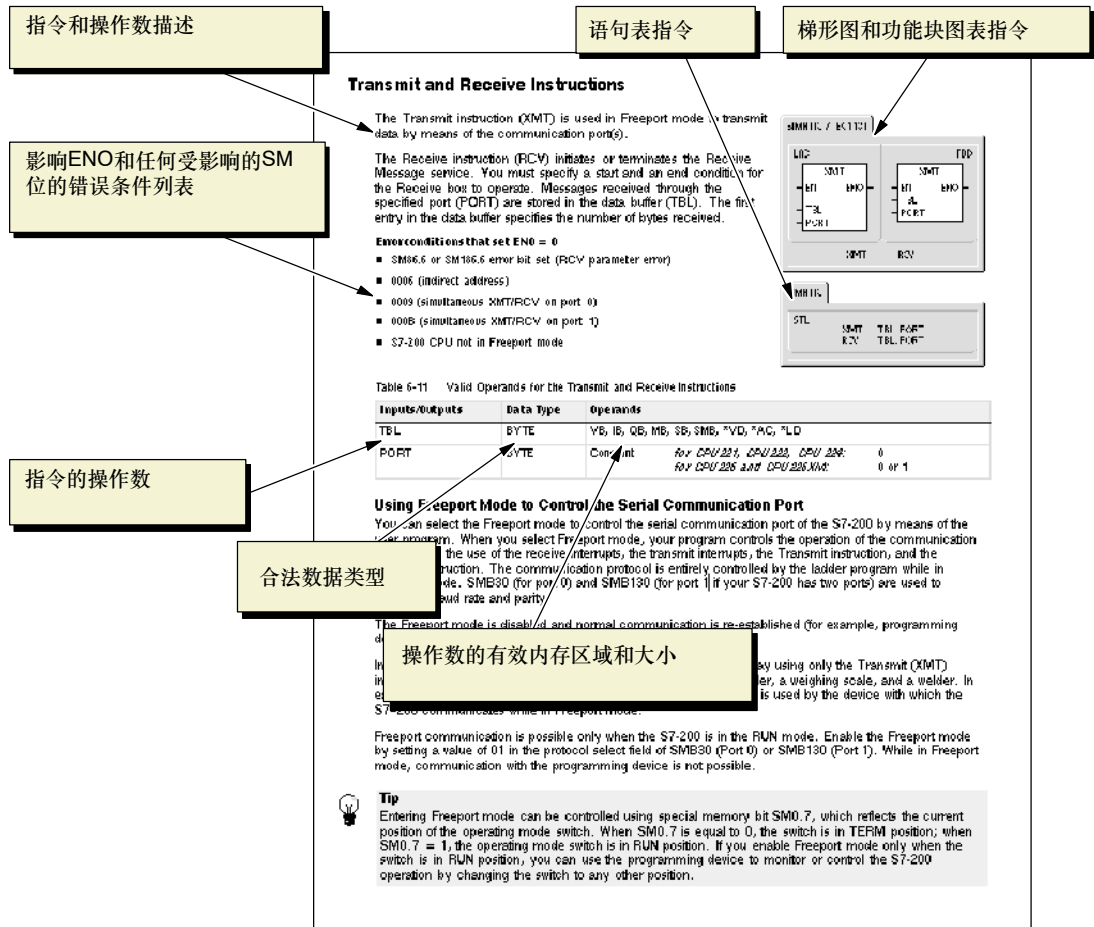


图6-1 指令描述

## S7-200内存范围和特征

表6-1 S7-200 CPU的内存范围和特征

描述	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
用户程序大小	4096个字节	4096个字节	8192个字节	8192个字节	16384个字节
用户数据大小	2048个字节	2048个字节	5120个字节	5120个字节	10240个字节
进程图像输入寄存器	I0.0到I15.7	I0.0到I15.7	I0.0到I15.7	I0.0到I15.7	I0.0到I15.7
进程图像输出寄存器	Q0.0到Q15.7	Q0.0到Q15.7	Q0.0到Q15.7	Q0.0到Q15.7	Q0.0到Q15.7
模拟输入（只读）	--	AIW0到AIW30	AIW0到AIW62	AIW0到AIW62	AIW0到AIW62
模拟输出（只写）	--	AQW0到AQW30	AQW0到AQW62	AQW0到AQW62	AQW0到AQW62
变量内存（V）	VB0到VB2047	VB0到VB2047	VB0到VB5119	VB0到VB5119	VB0到VB10239
局部内存（L） <sup>1</sup>	LB0到LB63	LB0到LB63	LB0到LB63	LB0到LB63	LB0到LB63
位存储器（M）	M0.0到M31.7	M0.0到M31.7	M0.0到M31.7	M0.0到M31.7	M0.0到M31.7
特殊内存（SM） 只读	SM0.0到SM179.7 SM0.0到SM29.7	SM0.0到SM299.7 SM0.0到SM29.7	SM0.0到SM549.7 SM0.0到SM29.7	SM0.0到SM549.7 SM0.0到SM29.7	SM0.0到SM549.7 SM0.0到SM29.7
计时器	256（T0到T255）	256（T0到T255）	256（T0到T255）	256（T0到T255）	256（T0到T255）
保持开延迟	1毫秒 T0、T64 10毫秒 T1到T4、和 T65到T68 100毫秒 T5到T31、和 T69到T95	1毫秒 T0、T64 10毫秒 T1到T4、和 T65到T68 100毫秒 T5到T31、和 T69到T95	1毫秒 T0、T64 10毫秒 T1到T4、和 T65到T68 100毫秒 T5到T31、和 T69到T95	1毫秒 T0、T64 10毫秒 T1到T4、和 T65到T68 100毫秒 T5到T31、和 T69到T95	1毫秒 T0、T64 10毫秒 T1到T4、和 T65到T68 100毫秒 T5到T31、和 T69到T95
开 / 关延迟	1毫秒 T32、T96 10毫秒 T33到T36、和 T97到T100 100毫秒 T37到T63、和 T101到T255	1毫秒 T32、T96 10毫秒 T33到T36、和 T97到T100 100毫秒 T37到T63、和 T101到T255	1毫秒 T32、T96 10毫秒 T33到T36、和 T97到T100 100毫秒 T37到T63、和 T101到T255	1毫秒 T32、T96 10毫秒 T33到T36、和 T97到T100 100毫秒 T37到T63、和 T101到T255	1毫秒 T32、T96 10毫秒 T33到T36、和 T97到T100 100毫秒 T37到T63、和 T101到T255
计数器	C0到C255	C0到C255	C0到C255	C0到C255	C0到C255
高速计数器	HC0、HC3、HC4 和HC5	HC0、HC3、HC4 和HC5	HC0到HC5	HC0到HC5	HC0到HC5
顺序控制继电器（S）	S0.0到S31.7	S0.0到S31.7	S0.0到S31.7	S0.0到S31.7	S0.0到S31.7
累加器寄存器	AC0到AC3	AC0到AC3	AC0到AC3	AC0到AC3	AC0到AC3
跳转 / 标签	0至255	0至255	0至255	0至255	0至255
调用 / 子例行程序	0至63	0至63	0至63	0至63	0至127
中断例行程序	0至127	0至127	0至127	0至127	0至127
正向 / 负向转换	256	256	256	256	256
PID循环	0至7	0至7	0至7	0至7	0至7
端口	端口0	端口0	端口0	端口0、端口1	端口0、端口1

<sup>1</sup> LB60到LB63由STEP7-Micro/WIN版本3.0或更新版本保留。

表6-2 S7-200 CPU的操作数范围

存取方式		CPU 221	CPU 222	CPU 224□CPU 226	CPU 226XM
位存取 (字节·位)	I	0.0至15.7	0.0至15.7	0.0至15.7	0.0至15.7
	Q	0.0至15.7	0.0至15.7	0.0至15.7	0.0至15.7
	V	0.0至2047.7	0.0至2047.7	0.0至5119.7	0.0至10239.7
	M	0.0至31.7	0.0至31.7	0.0至31.7	0.0至31.7
	SM	0.0至179.7	0.0至299.7	0.0至549.7	0.0至549.7
	S	0.0至31.7	0.0至31.7	0.0至31.7	0.0至31.7
	T	0至255	0至255	0至255	0至255
	C	0至255	0至255	0至255	0至255
	L	0.0至59.7	0.0至59.7	0.0至59.7	0.0至59.7
字节存取	IB	0到15	0到15	0到15	0到15
	QB	0到15	0到15	0到15	0到15
	VB	0至2047	0至2047	0至5119	0至10239
	MB	0到31	0到31	0到31	0到31
	SMB	0至179	0至299	0至549	0至549
	SB	0到31	0到31	0到31	0到31
	L	0至63	0至63	0至63	0至255
	AC	0至3	0至3	0至3	0至255
	字存取	IW	0至14	0至14	0至14
QW		0至14	0至14	0至14	0至14
VW		0至2046	0至2046	0至5118	0至10238
MW		0至30	0至30	0至30	0至30
SMW		0至178	0至298	0至548	0至548
SW		0至30	0至30	0至30	0至30
T		0至255	0至255	0至255	0至255
C		0至255	0至255	0至255	0至255
LW		0至58	0至58	0至58	0至58
AC		0至3	0至3	0至3	0至3
AIW		无	0至30	0至62	0至62
AQW		无	0至30	0至62	0至62
双字存取标识号		ID	0至12	0至12	0至12
	QD	0至12	0至12	0至12	0至12
	VD	0至2044	0至2044	0至5116	0至10236
	MD	0至28	0至28	0至28	0至28
	SMD	0至176	0至296	0至546	0至546
	SD	0至28	0至28	0至28	0至28
	LD	0至56	0至56	0至56	0至56
	AC	0至3	0至3	0至3	0至3
	HC	0、3、4、5	0、3、4、5	0至5	0至5

## 位逻辑指令

### 接点

#### 标准接点

常开接点指令（LD、A和O）和常闭接点指令（LDN、AN、ON）从内存或从进程图像寄存器获取参考数值。标准接点指令从内存（如果数据类型是I或Q就从进程图像寄存器）获取参考数值。

当位等于1时，常开接点闭合（接通）；当位等于0时，常闭接点闭合（接通）。在功能块图表中，到“与”和“或”方框的输入可以扩展到最大32个输入点。在语句表中，常开指令将地址位“载入”堆栈的顶部，或与之作“与”、“或”运算；常闭指令将地址位逻辑非地“载入”堆栈顶部位，或与堆栈顶部位的逻辑非作“与”、“或”运算。

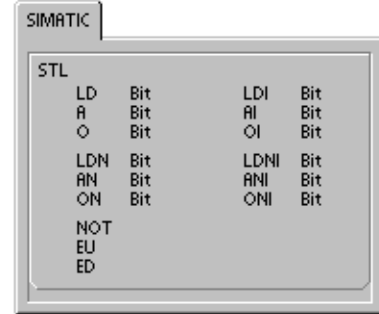
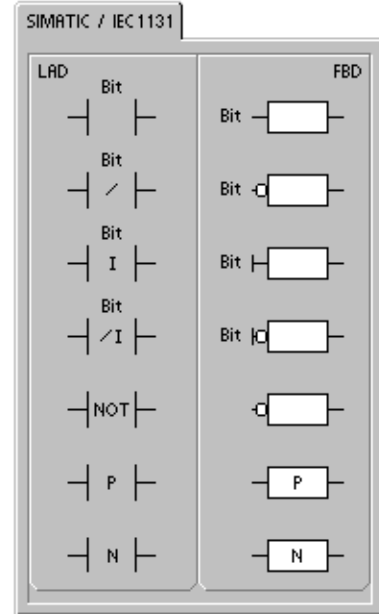
#### 立即接点

立即接点不依赖于S7-200扫描循环更新；它立即更新。立即常开接点指令（LDI、AI和OI）和立即常闭接点指令（LDNI、ANI和ONI）当执行指令时获得实际输入数值，但进程图像寄存器不更新。

当实际输入点（位）是1时，立即常开接点关闭（接通）；当实际输入点（位）是0时，立即常闭接点关闭（接通）。常开指令立即将实际输入数值“载入”堆栈的顶部，或与之作“与”或“或”运算。常闭指令立即将实际输入点的逻辑非“载入”堆栈的顶部，或将它与之作“与”或“或”运算。

#### NOT（非）指令

非指令（NOT）改变功率流输入的状态（即它将堆栈顶部的数值从0改为1或从1改为0）。



#### 正向和负向转换指令

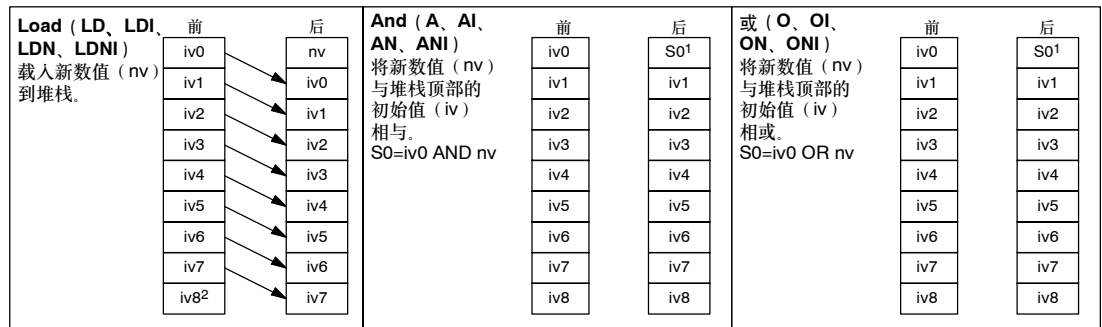
正向转换接点指令（EU）让功率流对每个断开至接通转换进行一次扫描。负向转换接点指令（ED）让功率流对每个接通至断开转换进行一次扫描。对于正向转换指令，检测到堆栈顶部的数值0到1的转换，就将堆栈顶部数值设为1；否则，设为0。对于负向转换指令，检测到堆栈顶部的数值1到0的转换，就将堆栈顶部数值设为1；否则，设为0。

对于运行时间编辑（当在运行模式下编辑程序时），必须为正向转换和负向转换指令输入一个参数。关于在运行模式下编辑的更多信息参见第5章。

表6-3 位逻辑输入指令的有效操作数

输入 / 输出	数据类型	操作数
位	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
位（立即）	BOOL	I

如图6-2所示，S7-200使用逻辑堆栈来解决控制逻辑问题。在这些实例中，“iv0”到“iv7”标识逻辑堆栈的初始值，“nv”标识指令提供的新数值，而“S0”标识存储在逻辑堆栈中的计算后数值。



- 1 S0标识存储在逻辑堆栈中的计算后的数值。
- 2 在执行载入之后，数值iv8丢失。

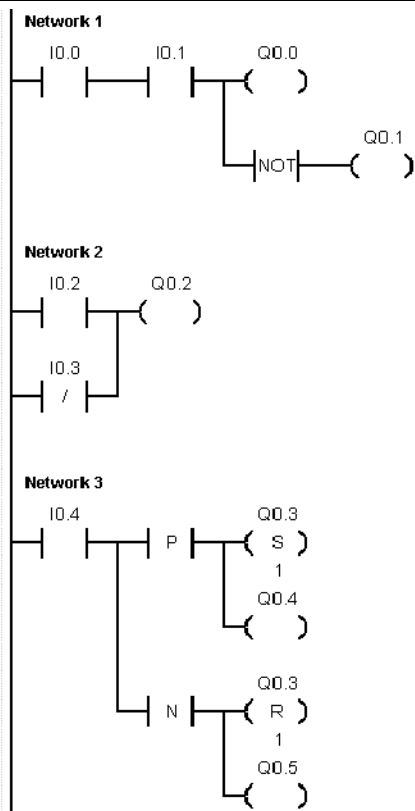
图6-2 接点指令的操作。



#### 提示

因为正向转换和负向转换指令需要接通至断开或断开至接通转换，所以在第一次扫描无法检测到边沿向上或边沿向下跳转。在第一次扫描期间，S7-200设置由这些指令指定的位的状态。在随后的扫描中，然后这些指令能检测指定位的跳转。

实例：接点指令



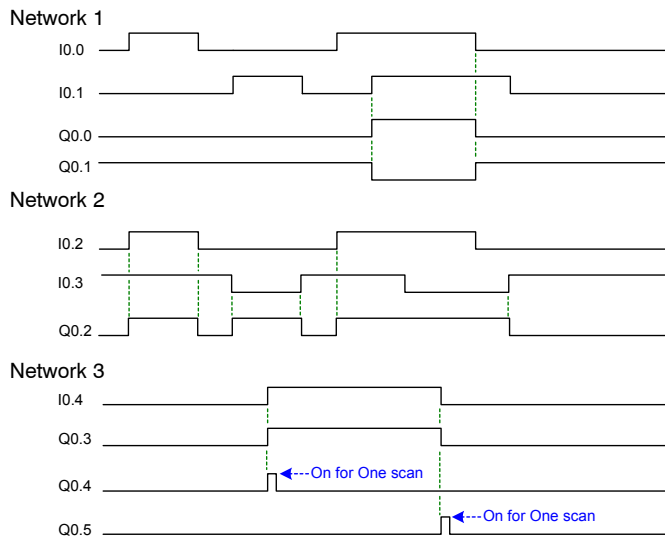
```

Network 1 //常开接点I0.0 “与” I0.1必须为接通（关闭）以激活Q0.0。NOT
           //(非)指令作为反相器。
           //在“运行”模式下，Q0.0和Q0.1具有相反的逻辑状态。
LD I0.0
A I0.1
= Q0.0
NOT
= Q0.1

Network 2 //常开接点I0.2必须为接通或者常闭接点I0.3必须为断开以激活
           //Q0.2。一个或多个平行LAD分支（“或”逻辑输入）必须为
           //真，以使输出激活。
LD I0.2
ON I0.3
= Q0.2

Network 3 //P接点上的上升沿输入或N接点上的下降沿输入在1个扫描循环持续
           //时间内输出一个脉冲。在“运行”模式下，Q0.4和Q0.5的脉冲
           //状态改变太快，以致在程序状态视图中无法看见。
           //设置和重设输出在Q0.3中锁存该脉冲，
           //使得状态改变在程序状态视图中可见。
LD I0.4
LPS
EU
S Q0.3, 1
= Q0.4
LPP
ED
R Q0.3, 1
= Q0.5
    
```

计时图





## 线圈

### 输出

输出指令(=)将输出位的新数值写到进程图像寄存器。当执行输出指令时，S7-200将进程图像寄存器中的输出位转为接通或断开。对于梯形图和功能块图表，指定的位设置为等于功率流。对于语句表，堆栈顶部的数值复制到指定位。

### 立即输出

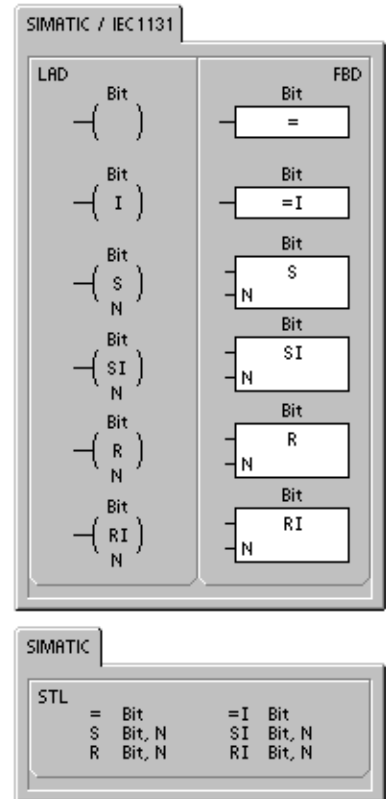
当执行指令时，立即输出指令(=I)将新数值写到实际输出和相应的进程图像寄存器位置。

当执行立即输出指令时，实际输出点(位)立即设置为等于功率流。对于语句表，该指令立即将堆栈顶部的数值复制到指定的实际输出位(语句表)。“I”说明立即引用；当执行该指令时，新数值被写入实际输出和相应的进程图像寄存器位置。这与非立即引用不同，后者只将新数值写入进程图像寄存器。

### 设置和重设

设置(S)和重设(R)指令设置(接通)或重设(断开)以指定地址(位)开始的指定数目的点(N)。可以设置或重设从1到255个点。

如果重设指令指定定时器位(T)或计数器位(C)，该指令重设计时器或计数器位并清除计时器或计数器的当前值。



### 设置ENO = 0的错误条件

- 0006 (间接地址)
- 0091 (操作数超出范围)

### 立即设置和立即重设

立即设置和立即重设指令立即设置(接通)或立即重设(断开)以指定地址(位)开始的数目点(N)。可以立即设置或重设从1到128个点。

“I”说明立即引用；当执行该指令时，新数值被写到实际输出点和相应的进程图像寄存器位置。这与非立即引用不同，后者只将新数值写到进程图像寄存器。

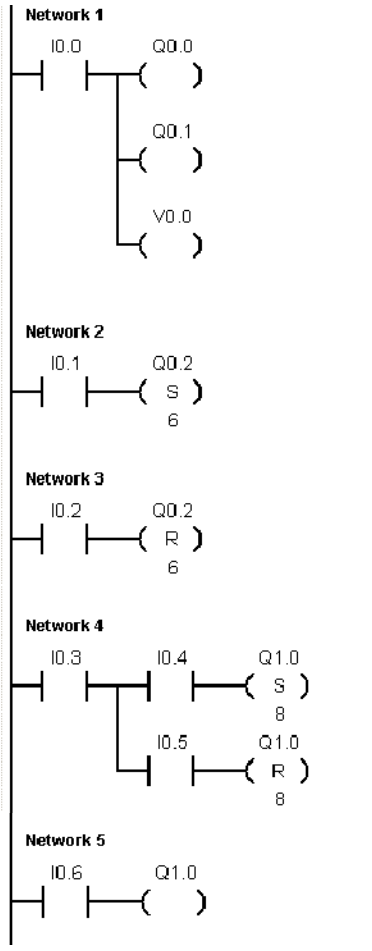
### 设置ENO = 0的错误条件

- 0006 (间接地址)
- 0091 (操作数超出范围)

表6-4 位逻辑输出指令的有效操作数

输入/输出	数据类型	操作数
位	BOOL	I、Q、V、M、SM、S、T、C、L
位(立即)	BOOL	Q
N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量

实例：线圈指令



```

Network 1 //输出指令将位数值分配给外部I/O (I, Q)
           //和内部存储器 (M、SM、T、C、V、S、L)。
LD I0.0
= Q0.0
= Q0.1
= V0.0

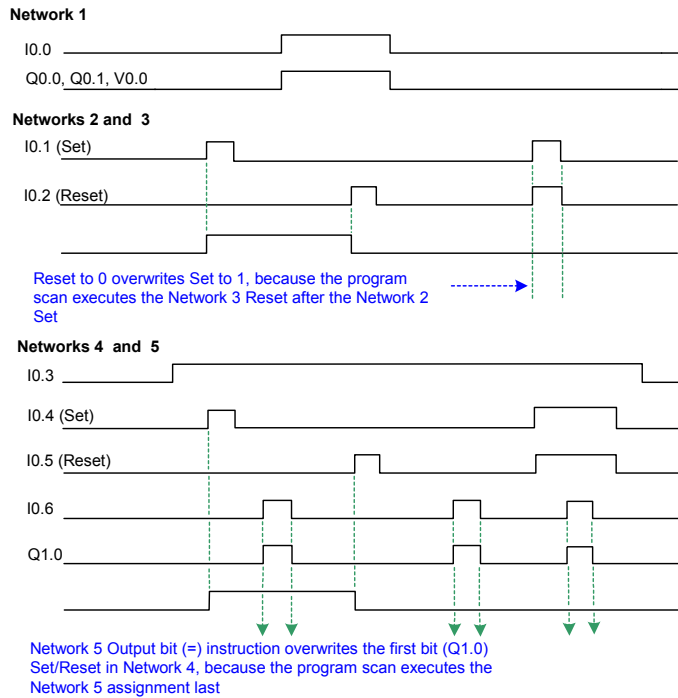
Network 2 //将6位连续的组设置为数值1。
           //指定起始位地址和要设置多少位。
           //当第一位 (Q0.2) 的数值为1时, 用于设置的程序状态指示器为ON。
LD I0.1
S Q0.2, 6

Network 3 //将6位连续的组重设为数值0。
           //指定起始位地址和要重设多少位。
           //当第一位 (Q0.2) 的数值为0时, 用于重设的程序状态指示器为ON。
LD I0.2
R Q0.2, 6

Network 4 //设置和重设8个输出位 (Q1.0到Q1.7) 作为一个群组。
LD I0.3
LPS
A I0.4
S Q1.0, 8
LPP
A I0.5
R Q1.0, 8

Network 5 //“设置和重设”指令完成锁存继电器的功能。
           //要隔离“设置/重设”位, 确保它们没有被其它赋值指令重写。
           //在此例中, Network 4设置和重设8个输出位 (Q1.0到Q1.7)
           //作为一个群组。
           //在“运行”模式下, Network 5可以重写Q1.0位数值,
           //并控制Network 4中的“设置/重设”程序状态指示器。
LD I0.6
= Q1.0
    
```

计时图



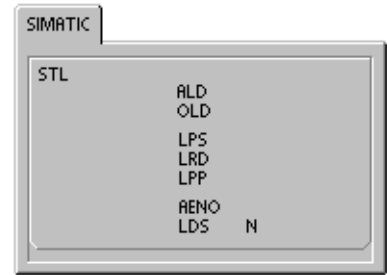
## 逻辑堆栈指令

### AND（与）载入

AND（与）载入指令（ALD）使用逻辑“与”操作将堆栈第一层和第二层的数值组合在一起。结果载入堆栈顶部。在执行ALD后，堆栈深度减少一层。

### OR（或）载入

OR（或）载入指令（OLD）使用逻辑“或”操作将堆栈第一层和第二层的数值组合在一起。结果载入堆栈顶部。在执行OLD后，堆栈深度减少一层。



### 逻辑进栈

逻辑进栈指令（LPS）复制堆栈顶部的数值，并将此数值推到堆栈中。堆栈底部被推出和丢失。

### 逻辑读取

“逻辑读取”指令（LRD）将第二个堆栈值复制到堆栈顶部。堆栈没有被推出或弹出，但堆栈顶部的旧数值被复制的数值破坏。

### 逻辑出栈

“逻辑出栈”指令（LPP）将堆栈的一个数值弹出堆栈。第二个堆栈数值成为堆栈数值的新顶部。

### AND ENO

AND ENO指令（AENO）完成ENO位和堆栈顶部的逻辑“与”运算，以在梯形图或功能块图表中产生与方框的ENO位同样的效果。“与”操作的结果是堆栈的新顶部。

ENO是梯形图和功能块图表中方框的布尔型输出。如果方框在EN输入处有功率流，并无错地执行，ENO输出将功率流传递到下一个单元。可以使用ENO作为启用位，指示指令的成功完成。ENO位和堆栈的顶部一起用于影响随后指令功率流的执行。语句表指令没有EN输入。堆栈的顶部必须为逻辑1，以用于条件指令的执行。在语句表中也没有ENO输出。然而，相应于梯形图和功能块图表指令带有ENO输出的语句表指令设置特殊ENO位。此位可用AENO指令存取。

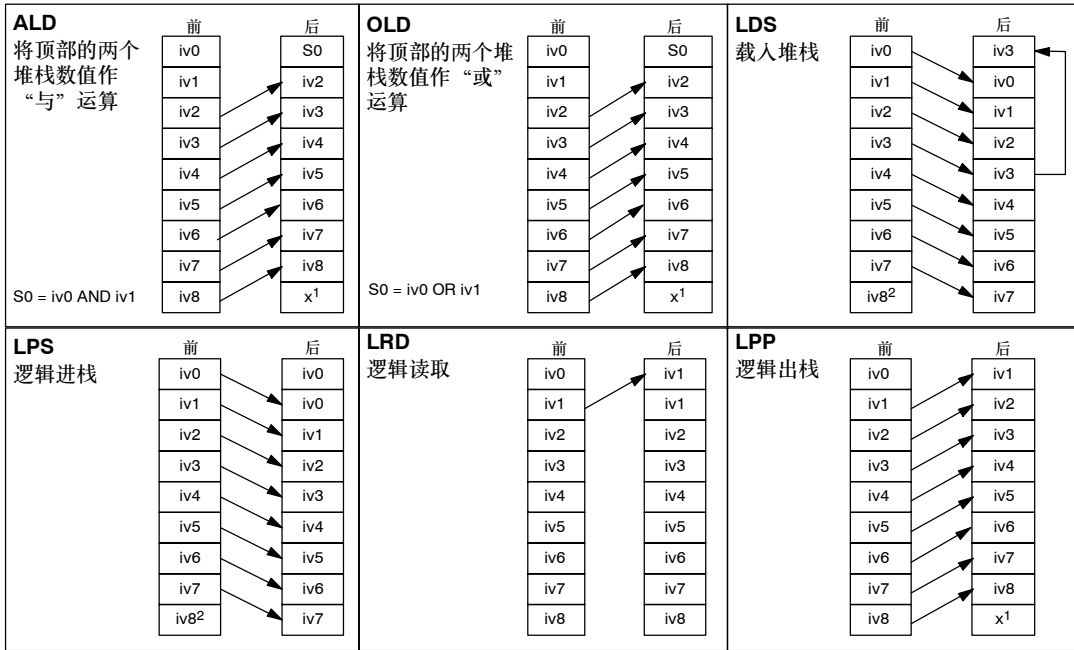
### 载入堆栈

载入堆栈指令（LDS）复制堆栈上的堆栈位（N）并将此数值放在堆栈的顶部。堆栈底部被推出和丢失。

表6-5 载入堆栈指令的有效操作数

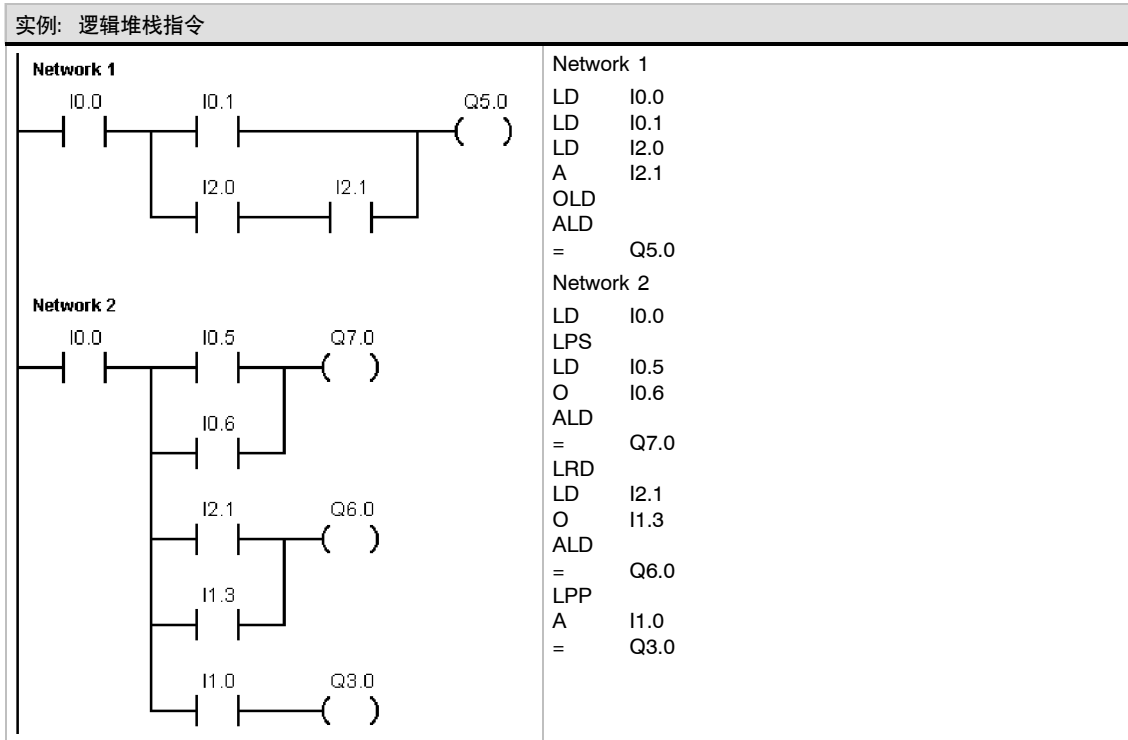
输入 / 输出	数据类型	操作数
N	BYTE	常量（0到8）

如图6-3所示，S7-200使用逻辑堆栈来解决控制逻辑问题。在这些实例中，“iv0”到“iv7”标识逻辑堆栈的初始值，“nv”标识指令提供的新数值，而“S0”标识存储在逻辑堆栈中的计算后数值。



- 1 数值未知（它不是0就是1）。
- 2 在执行逻辑进栈或载入堆栈指令后，数值iv8丢失。

图6-3 操作逻辑堆栈指令



## 设置和重设主要双稳态指令

设置主要双稳态是一个锁存器，在那里设置占优。如果设置（S1）和重设（R）信号都为真，输出（OUT）为真。

重设主要双稳态是一个锁存器，在那里重设占优。如果设置（S）和重设（R1）信号都是真，输出（OUT）为假。

位参数指定设置或重设的布尔型参数。可选输出反映位参数的信号状态。

表6-7显示范例程序的真值表。

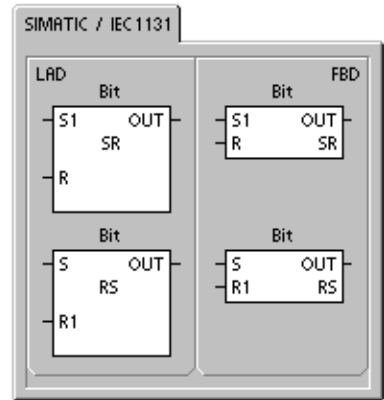


表6-6 设置主要双稳态和重设主要双稳态指令的有效操作数

输入 / 输出	数据类型	操作数
S1, R	BOOL	I、Q、V、M、SM、S、T、C、功率流
S、R1、OUT	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
位	BOOL	I、Q、V、M、S

### 实例：设置和重设主要双稳态指令

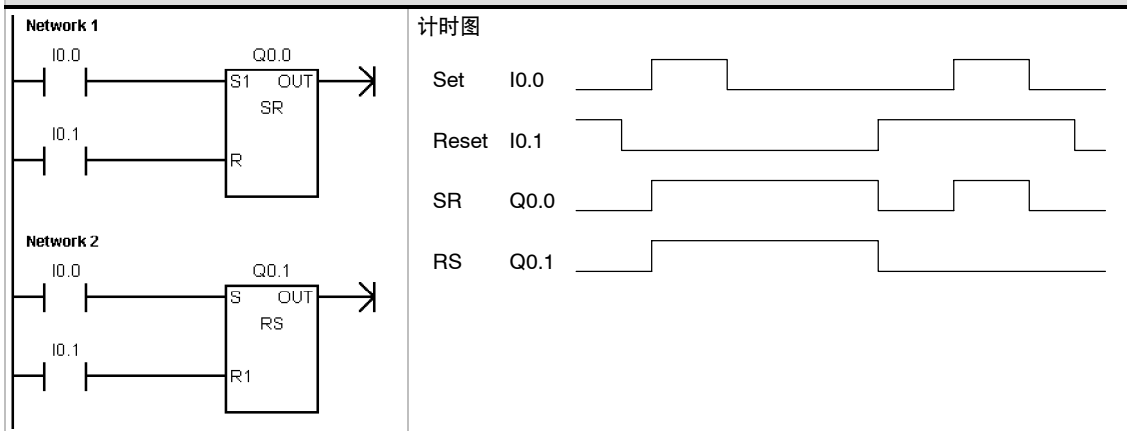


表6-7 设置和重设主要双稳态指令的真值表

指示	S1	R	Out (位)
设置主要双稳态指令 (SR)	0	0	前一个状态
	0	1	0
	1	0	1
	1	1	1
指示	S	R1	Out (位)
重设主要双稳态指令 (RS)	0	0	前一个状态
	0	1	0
	1	0	1
	1	1	0

## 时钟指令

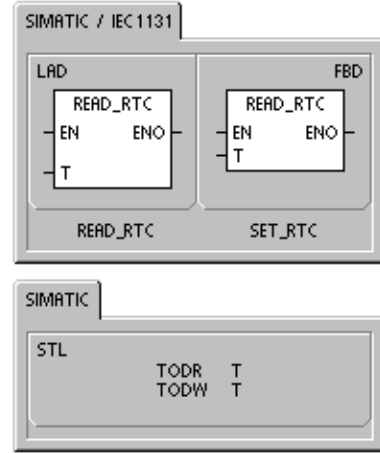
### 读取实时时钟和设置实时时钟

读取实时时钟（TODR）指令从硬件时钟读当前时间和日期，并将它载入以地址T开始的8个字节时间缓冲器。设置实时时钟（TODW）指令将当前时间和日期写入硬件时钟，硬件时钟以由T指定的8字节时间缓冲器地址开始。

必须将所有日期和时间数值以BCD码格式编码（例如，16#97表示1997年）。图6-4显示时间缓冲器（T）的格式。

在扩展电源停电后或当内存丢失时，日时（TOD）时钟初始化下列日期和时间：

日期: 01-01-90  
 时间: 00:00:00  
 星期: 星期日



设置ENO = 0的错误条件

- 0006（间接地址）
- 0007（TOD数据出错）只设置实时时钟
- 000C（时钟没有显示）

表6-8 时钟指令的有效操作数

输入 / 输出	数据类型	操作数
T	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC

T	T+1	T+2	T+3	T+4	T+5	T+6	T+7
年: 00至99	月: 01至12	日: 01至31	小时: 00至23	分钟: 00至59	秒: 00至59	0	星期: 0到7*

\*T+7 1=星期日、7=星期六  
0禁用星期。

图6-4 8字节时间缓冲器（T）的格式



#### 提示

S7-200 CPU不执行检查以校验该星期基于日期是否正确。无效的日期，如2月30日可能被接受。必须确保输入的日期是正确的。

不要在主程序和中断例行程序中都使用TODR/TODW指令。在中断例行程序中的TODR/TODW指令试图执行，而另一个在过程中的TODR/TODW指令无法执行。设置SM4.3指示尝试两个同时进行存取时钟（非致命错误0007）。

在S7-200中日时时钟只使用最低有效的两个数字表示年，所以对于2000年，表达为00。S7-200 PLC不以任何方式使用年信息。然而，使用年数值来作运算或比较的用户程序必须考虑到二位表达方式，把年变成四位数字。

闰年经正确处理到2096年。

## 通讯指令

### 网络读取和网络写入指令

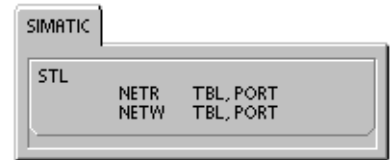
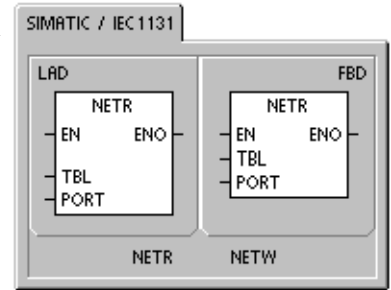
网络读取指令（NETR）启动通讯操作以通过指定的端口（PORT）如表（TBL）所定义从远程设备收集数据。网络写入指令（NETW）启动通讯操作以如表（TBL）所定义的通过指定的端口（PORT）将数据写到远程设备。

设置ENO = 0的错误条件

- 0006（间接地址）
- 如果函数返回错误，并设置表格状态字节的E位（参见图6-5）

网络读取指令可以从远程站读取至多16个字节的的信息，而网络写入指令可以从远程站写至多16个字节的的信息。

在程序中可以有任意数量的网络读取和网络写入指令，但在同一时间只有最多八个网络读取和网络写入指令可以激活。例如，在S7-200中同时可以有4个网络读取和4个网络写入指令，或2个网络读取和6个网络写入指令激活。



指令向导

可以使用网络读取/网络写入指令向导配置计数器。要启动网络读取/网络写入指令向导，选择工具（Tools）> 指令向导（Instruction Wizard）菜单命令，然后从指令向导窗口选择网络读取/网络写入。

表6-9 网络读取和网络写入指令的有效操作数

输入/输出	数据类型	操作数
TBL	BYTE	VB、MB、*VD、*LD、*AC
PORT	BYTE	常量 对于CPU 221、CPU 222、CPU 224: 0 对于CPU 226和CPU 226XM: 0或1

图6-5描述由TBL参数引用的表格，表6-10列出出错代码。

字节 偏移	7	0	D 完成（函数已完成）： A 激活（功能已排队）： E 出错（函数返回错误）：	0 = 没有完成 0 = 没有激活 0 = 没有出错	1 = 完成 1 = 激活 1 = 出错	
0	D	A	E	0	错误代码	
1	远程站地址					远程站地址：其数据被存取的PLC的地址。
2	指向数据的指针					在远程站中指向数据区的指针：指向要存取数据的间接指针。
3	区域在					
4	远程站					
5	(I、Q、M或V)					数据长度：要在远程站中存取的数据的字节数（1到16个字节）。
6	数据长度					
7	数据字节0					接收或传输数据区。1到16个字节为数据保留。 对于网络读取指令，存储当执行指令时从远程站读取的数值。 对于网络写入指令，存储当执行指令时要发送到远程站的数值。
8	数据字节1					
	⋮					
	⋮					
22	数据字节15					

图6-5 网络读取和网络写入指令的TBL参数

表6-10 TBL参数的出错代码

代码	定义
0	无错。
1	超时出错：远程站没有反应。
2	接收出错：在响应中奇偶校验、帧或检验和出错。
3	脱机出错：复制站地址或故障硬件引起的冲突。
4	队列溢出出错：多于8个网络读取或网络写入指令激活。
5	违反协议：尝试执行网络读取或网络写入指令而没有在SMB30或SMB130中启用PPI主设备模式。
6	非法参数：TBL参数包含非法的或无效的数值。
7	没有资源：远程站占线。（上载或下载序列在处理中。）
8	层面7出错：违反应用程序协议
9	讯息出错：错误数据地址或错误数据长度
A到F	未用。（保留）

图6-6显示实例说明网络读取和网络写入指令的效用。对于此实例，考虑桶装黄油被填充并送到四个装箱机（容器包装机）之一的生产线。容器包装机把八桶黄油装入单个纸板箱。分流机械设备控制黄油桶流向每个容器包装机。四个S7-200控制容器包装机，一个带TD 200操作员界面的S7-200控制分流机。



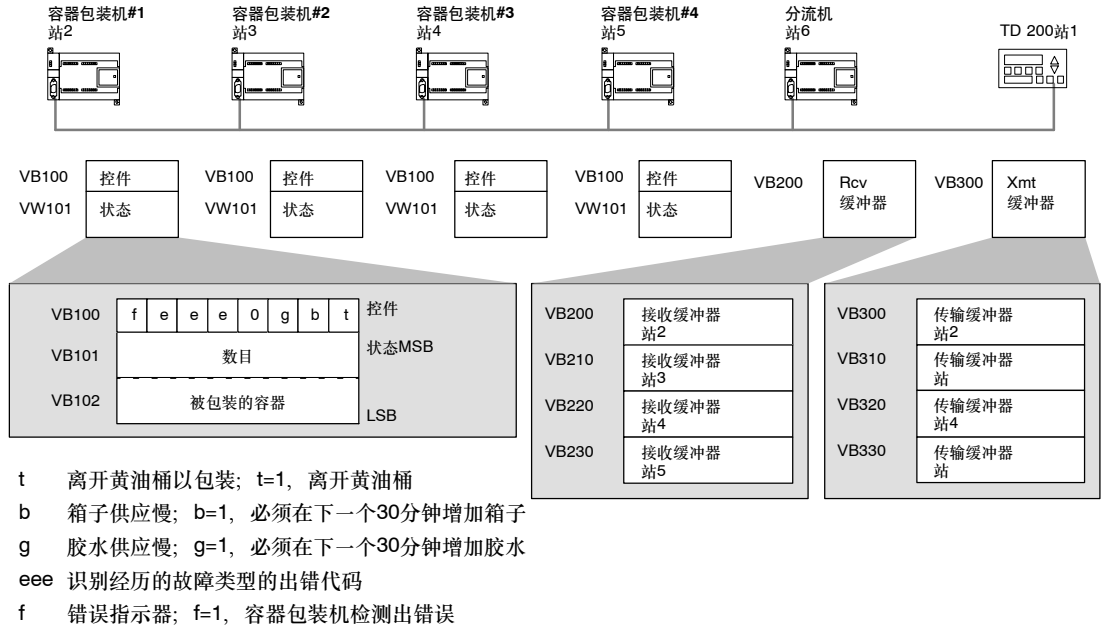


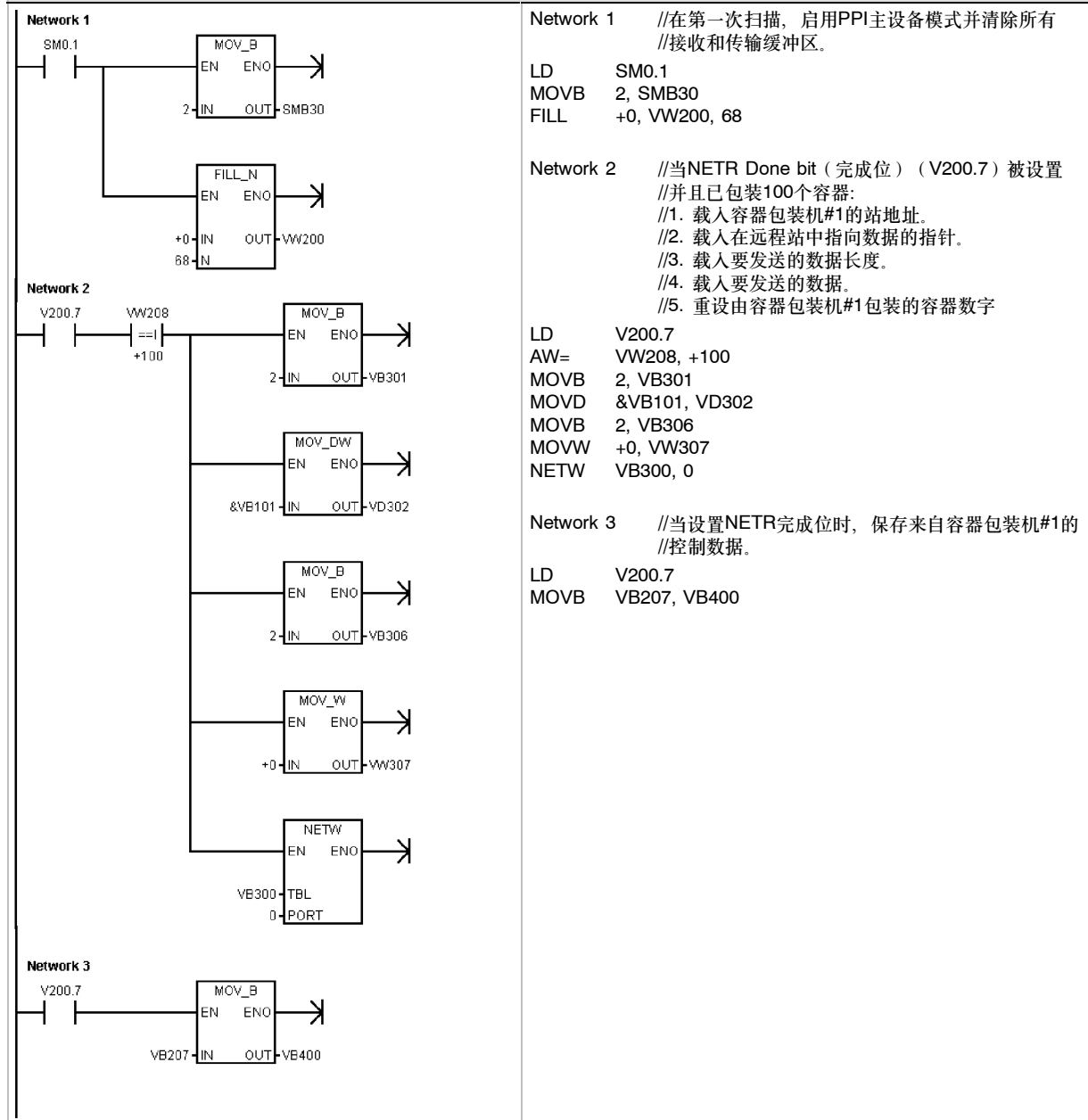
图6-6 网络读取和网络写入指令的实例

图6-7显示用于在站2中存取数据的接收缓冲器（VB200）和传输缓冲器。S7-200使用网络读取指令从每个容器包装机连续读取控制和状态信息。容器包装机每次包装100个容器，分流机记下此事，并使用网络写入指令发送讯息清除状态字。

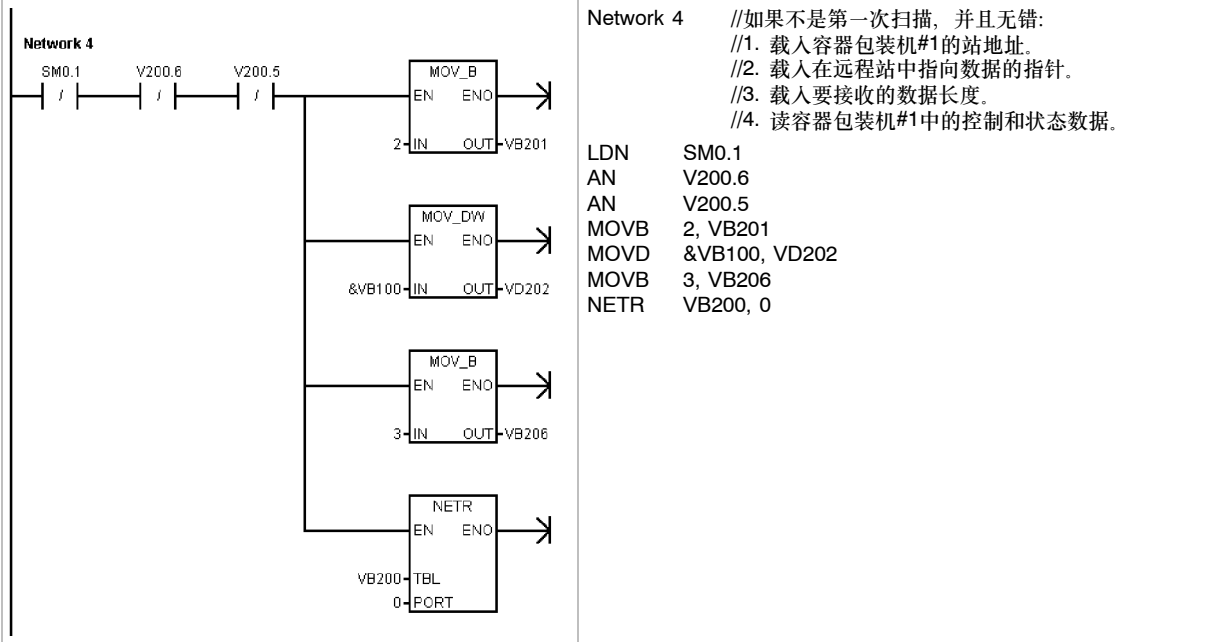
用于从容器包装机#1 读取的接收缓冲器		用于清除容器包装机#1 的计数的传输缓冲器	
	7		0
VB200	D	A	E 0 错误代码
VB201	远程站地址 = 2		
VB202	指针指向		
VB203	数据区		
VB204	在		
VB205	远程站 = (&VB100)		
VB206	数据长度 = 3个字节		
VB207	控件		
VB208	状态 (MSB)		
VB209	状态 (LSB)		
VB300	D	A	E 0 错误代码
VB301	远程站地址 = 2		
VB302	指针指向		
VB303	数据区		
VB304	在		
VB305	远程站 = (&VB101)		
VB306	数据长度 = 2个字节		
VB307	0		
VB308	0		

图6-7 网络读取 / 写入实例的范例TBL数据

## 实例：网络读取和网络写入指令



## 实例: 网络读取和网络写入指令 (续)



## 发送和接收指令（空闲端口）

“传输”指令（XMT）用于自由端口模式以通过通讯端口传输数据。

“接收”指令（RCV）启动或终止接收讯息功能。必须指定一个接收方框操作的开始和结束条件。通过指定的端口（PORT）接收的讯息存储在数据缓冲区（TBL）。数据缓冲区的第一个条目指定接收的字节数。

设置ENO = 0的错误条件

- 0006（间接地址）
- 0009（同时在端口0传输 / 接收）
- 000B（同时在端口1传输 / 接收）
- 接收参数错误设置SM86.6或SM186.6
- S7-200 CPU不在自由端口模式

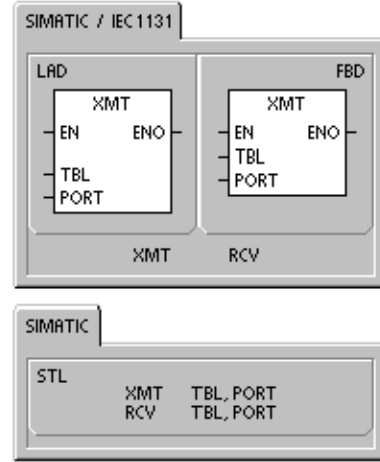


表6-11 传输和接收指令的有效操作数

输入 / 输出	数据类型	操作数
TBL	BYTE	IB、QB、VB、MB、SMB、SB、*VD、*LD、*AC
PORT	BYTE	常量 对于 CPU 221, CPU 222, CPU 224: 0 对于 CPU 226 和 CPU 226XM: 0或1

关于使用自由端口模式的更多信息，参见在第227页的第7章的“用自由端口模式创建自定义协议”部分。

### 使用自由端口模式控制串行通讯口

可以通过用户程序选择自由端口模式控制S7-200的串行通讯口。当选择自由端口模式时，用户程序通过使用接收中断、传输中断、“传输”指令和“接收”指令控制通讯端口的操作。在自由端口模式下，通讯协议完全由梯形程序控制。SMB30（对于端口0）和SMB130（对于端口1如果用户S7-200有两个端口）用于选择波特率和奇偶校验。

当S7-200处于“停止”模式下，自由端口模式禁用，正常通讯重新建立（例如，程序设计设备存取）。

在最简单的情况下，可以发送讯息到打印机或只使用“传输”（XMT）指令显示。其它实例包括到棒图代码阅读器、秤和焊接机的连接。在每个实例中，必须写用户程序以支持设备使用的协议，S7-200在自由端口模式下与该设备通讯。

自由端口通讯只有当S7-200在“运行”模式下才可能。通过在SMB30（端口0）或SMB130（端口1）的协议选择域设置01的数值启用自由端口模式。当处于自由端口模式下，与程序设计设备的通讯是不可能的。



#### 提示

自由端口模式可以使用特殊内存位SM0.7控制，SM0.7反映了操作模式开关的当前位置。当SM0.7等于0时，开关处于“终止”位置；当SM0.7 = 1时，操作模式开关处于“运行”位置。如果仅在开关处于“运行”位置时启用自由端口模式，可以使用程序设计设备通过将开关改变到其它位置监控或控制S7-200操作。

## 更换PPI通讯到自由端口模式

SMB30和SMB130为自由端口操作分别配置通讯端口0和1，并提供波特率、奇偶校验和数据位数的选择。图6-8描述自由端口控制字节。为所有配置产生一个停止位。

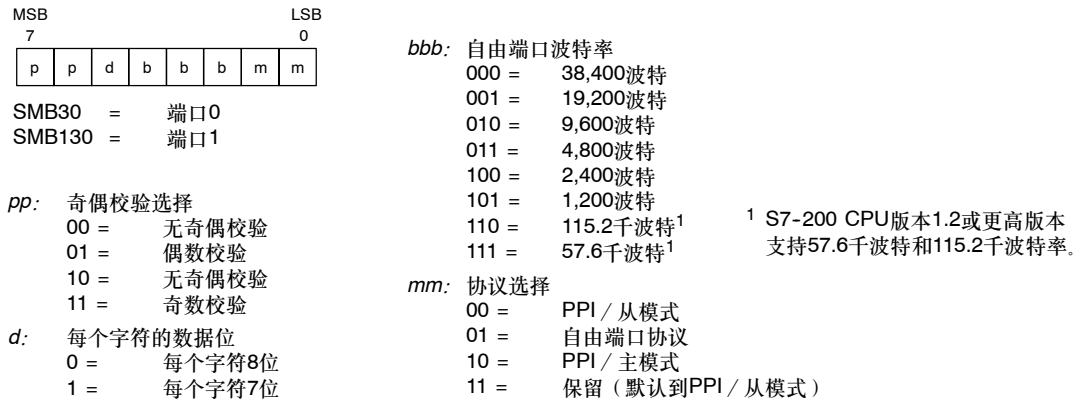


图6-8 自由端口模式的SM控制字节 (SMB30或SMB130)

## 传输数据

“传输”指令让用户发送一个或多个字符的缓冲区，至多为255个字符。

图6-9显示传输缓冲区的格式。

如果中断例行程序附加在传输完整事件上，在最后一个缓冲区的字符发送后，S7-200产生中断（对于端口0为中断事件9，而对于端口1为中断事件26）。

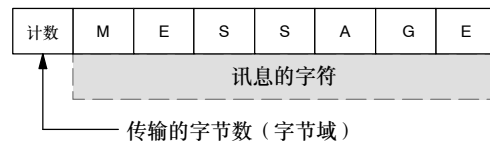


图6-9 传输缓冲区的格式

可以通过监控SM4.5或SM4.6进行传输而不需使用中断（例如，发送讯息到打印机）以用信号表示传输何时完成。

可以使用“传输”指令通过设置字符数为零产生“断开”条件，然后执行“传输”指令。这以当前的波特率对于16位次在线产生“断开”条件。传输“断开”与传输任何其它讯息一样处理，在那里当“断开”完成时产生传输中断，SM4.5或SM4.6以信号表示传输操作的当前状态。

## 接收数据

“接收”指令让用户接收一个或多个字符的缓冲区，至多255个字符。

图6-10显示接收缓冲器的格式。

如果中断例行程序附加在接收讯息完整事件上，在最后一个缓冲区的字符接收后，S7-200产生中断（对于端口0为中断事件23，而对于端口1为中断事件24）。

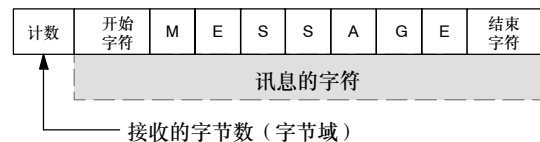


图6-10 接收缓冲器的格式

可以通过监控SMB86（端口0）或SMB186（端口1）接收讯息而不使用中断。当接收指令为非现用或已终止时，此字节为非零。当正在接收过程中时它为零。

如表6-12所示，使用SMB86通过SMB94的端口0和SMB186通过SMB194的端口1，接收指令允许选择讯息开始和讯息结束条件。



#### 提示

在超出或奇偶校验出错情况下，接收讯息功能自动终止。必须定义开始条件和结束条件（最大字符计数）用于操作接收讯息功能。

表6-12 接收缓冲器的字节（SMB86到SMB94，SM1B86到SMB194）

端口0	端口1	描述																		
SMB86	SMB186	接收讯息状态字节 <table style="margin-left: 20px;"> <tr> <td style="text-align: right;">MSB</td> <td colspan="6"></td> <td style="text-align: left;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td style="border: 1px solid black; padding: 2px;">n</td> <td style="border: 1px solid black; padding: 2px;">r</td> <td style="border: 1px solid black; padding: 2px;">e</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="border: 1px solid black; padding: 2px;">t</td> <td style="border: 1px solid black; padding: 2px;">c</td> <td style="border: 1px solid black; padding: 2px;">p</td> <td style="text-align: center;">0</td> </tr> </table> <p>n: 1 = 接收讯息功能终止：用户发出的禁用命令。            r: 1 = 接收讯息功能终止：输入参数中的错误或丢失开始或结束条件。            e: 1 = 结束字符接收。            t: 1 = 接收讯息功能终止：计时器时间到。            c: 1 = 接收讯息功能终止：最大字符计数完成。            p 1 = 接收讯息功能终止：奇偶校验出错。</p>	MSB							LSB	7	n	r	e	0	0	t	c	p	0
MSB							LSB													
7	n	r	e	0	0	t	c	p	0											
SMB87	SMB187	接收讯息控制字节 <table style="margin-left: 20px;"> <tr> <td style="text-align: right;">MSB</td> <td colspan="6"></td> <td style="text-align: left;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td style="border: 1px solid black; padding: 2px;">en</td> <td style="border: 1px solid black; padding: 2px;">sc</td> <td style="border: 1px solid black; padding: 2px;">ec</td> <td style="border: 1px solid black; padding: 2px;">il</td> <td style="border: 1px solid black; padding: 2px;">c/m</td> <td style="border: 1px solid black; padding: 2px;">tmr</td> <td style="border: 1px solid black; padding: 2px;">bk</td> <td style="border: 1px solid black; padding: 2px;">0</td> <td style="text-align: center;">0</td> </tr> </table> <p>en: 0 =接收讯息功能禁用。            1 =接收讯息功能启用。            启用 / 禁用接收讯息位在每次执行RCV指令时检查。            sc: 0 =忽略SMB88或SMB188。            1 =使用SMB88或SMB188的数值检测讯息的开始。            ec: 0 =忽略SMB89或SMB189。            1 =使用SMB89或SMB189的数值检测讯息的结束。            il: 0 =忽略SMW90或SMW190。            1 =使用SMW90或SMW190的数值检测空闲行条件。            c/m: 0 =计时器是字符间的计时器。            1 =计时器是讯息计时器。            tmr: 0 =忽略SMW92或SMW192。            1 =如果SMW92或SMW192中的时间间隔超出，终止接收。            bk: 0 =忽略断开条件。            1 =使用断开条件作为讯息检测的开始。</p>	MSB							LSB	7	en	sc	ec	il	c/m	tmr	bk	0	0
MSB							LSB													
7	en	sc	ec	il	c/m	tmr	bk	0	0											
SMB88	SMB188	讯息字符的开始。																		
SMB89	SMB189	讯息字符的结束。																		
SMW90	SMW190	以毫秒为单位的空闲行时间周期。在空闲行时间到期后接收的第一个字符是新讯息的开始。																		
SMW92	SMW192	以毫秒为单位的字符间 / 讯息计时器超时数值。如果时间周期超出，接收讯息功能终止。																		
SMB94	SMB194	接收的最大字符数（1到255个字节）。此范围必须设置到期望的最大缓冲区大小，即使不使用字符计数讯息终端。																		

## 接收指令的开始和结束条件

接收指令使用接收讯息控制字节的位（SMB87或SMB187）来定义讯息开始和结束条件。



### 提示

如果在执行接收指令时通讯端口上有来自其它设备的通信量显示，接收讯息功能可能在该字符的中间开始接收字符，这将导致可能奇偶校验出错和接收讯息功能终止。如果奇偶校验未启用，接收的讯息可能包含不正确的字符。如下面的条目2.和条目6.所述，当指定开始条件为指定的起始字符或任何字符，此状况可能产生。

接收指令支持多个讯息开始条件。指定包括断开或空闲行检测的开始条件，通过在把字符放入讯息缓冲区前强制接收讯息功能将讯息的开始与字符的开始同步来避免此问题。

接收指令支持多个开始条件:

1. **空闲行检测:** 空闲行条件被定义为传输行上的静态或空闲时间。当通讯行静止或空闲在SMW90或SMW190中指定的毫秒数后，接收开始。当执行程序中的接收指令时，接收讯息功能开始搜索空闲行条件。如果在空闲行时间结束前收到任何字符，接收讯息功能忽略那些字符，并且以来自SMW90或SMW190的时间重新启动空闲行计时器。参见图6-11。在空闲行时间结束前，接收讯息功能存储所有随后在讯息缓冲区接收的字符。

空闲行时间应始终大于在指定的波特率下传送一个字符（起始位、数据位、奇偶校验和停止位）的时间。空闲行时间的典型数值是在指定的波特率下传送三个字符的时间。

使用空闲行检测作为二进制协议的的开始条件，前提是协议中没有特别的起始字符，或当协议指定讯息之间的最小时间时。

设置:  $il = 1, sc = 0, bk = 0, SMW90/SMW190 = \text{空闲行超时 (毫秒)}$

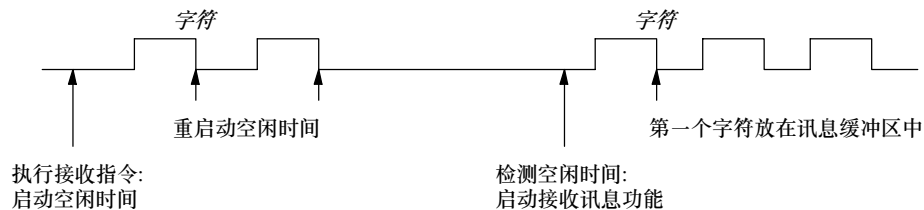


图6-11 使用空闲时间检测来启动接收指令

2. **起始字符检测:** 起始字符是用作讯息的第一个字符的任意字符。当接收到在SMB88或SMB188中指定的起始字符时，讯息功能启动。接收讯息功能将起始字符存储在接收缓冲器并作为讯息的第一个字符。接收讯息功能忽略任何在起始字符之前接收到的字符。起始字符和所有在起始字符后接收的字符都存储在讯息缓冲区。

典型地，将起始字符检测用于ASCII协议，在该协议中所有讯息以相同的字符开始。

设置:  $il = 0, sc = 1, bk = 0, SMW90/SMW190 = \text{无关}, SMB88/SMB188 = \text{起始字符}$

3. **空闲行和起始字符:** 接收指令可以以空闲行和起始字符的组合启动讯息。当执行接收指令时, 接收讯息功能搜索空闲行条件。在找到空闲行条件后, 接收讯息功能查找指定的起始字符。如果接收到除起始字符之外的任何字符, 接收讯息功能重新启动搜索空闲行条件。在空闲行条件前接收到的所有字符满足要求, 在起始字符之前接收的被忽略。起始字符与所有随后的字符一起放在讯息缓冲区中。

空闲行时间应始终大于在指定的波特率下传输一个字符(起始位、数据位、奇偶校验和停止位)的时间。空闲行时间的典型数值是在指定的波特率下传送三个字符的时间。

一般地, 当有协议指定讯息间的最小时间, 并且讯息的第一个字符是地址或指定特殊设备的東西时, 使用这种类型的启动条件。当在通讯链上有多种设备时, 这对于实现协议是很有用的。在这种情况下, 接收指令只在为指定地址或由起始字符指定的设备接收讯息时触发中断。

设置:  $il = 1, sc = 1, bk = 0, SMW90/SMW190 > 0, SMB88/SMB188 =$  起始字符

4. **断开检测:** 当接收到的数据保持为零值的时间大于完整的字符传输时间时, 指示断开。完整的字符传输时间定义为启动、数据、奇偶校验和停止位的总时间。如果接收指令配置为一旦接收到断开条件就启动讯息, 在断开条件后接收的任何字符被放入讯息缓冲区。在断开条件前接收的任何字符被忽略。

一般地, 只有当协议需要它时才能使用断开检测作为启动条件。

设置:  $il = 0, sc = 0, bk = 1, SMW90/SMW190 =$  无关,  $SMB88/SMB188 =$  无关

5. **断开和起始字符:** 接收指令可以在接收断开条件后配置来启动接收字符, 然后以此顺序启动指定起始字符。在断开条件后, 接收讯息功能查找指定的起始字符。如果接收到除起始字符之外的任何字符, 接收讯息功能重新启动搜索断开条件。在断开条件前接收的所有字符满足要求, 在起始字符之前接收的忽略。起始字符与所有随后的字符一起放在讯息缓冲区。

设置:  $il = 0, sc = 1, bk = 1, SMW90/SMW190 =$  无关,  
 $SMB88/SMB188 =$  起始字符

6. **任何字符:** 接收指令可以配置为立即启动接收任意和所有字符, 并将它们放入讯息缓冲区。这是空闲行检测的特殊情况。在这种情况下, 空闲行时间(SMW90或SMW190)设置为零。这强制接收指令在执行时立即开始接收字符。

设置:  $il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0, SMB88/SMB188 =$  无关

启动任何字符的讯息允许讯息计时器用于暂停接收讯息。该功能用于在自由端口用于实现协议的主设备或主机部分, 并且如果在指定的时间内没有接受到来自从属装置的响应而需要暂停的情况。因为空闲行时间设置为零, 当接收指令执行时讯息计时器启动。如果没有其它结束条件满足要求, 讯息计时器暂停, 并终止接收讯息功能。

设置:  $il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0, SMB88/SMB188 =$  无关  
 $c/m = 1, tmr = 1, SMW92 =$  讯息超时(毫秒)



接收指令支持多种方式终止讯息。讯息可以因为下列一个或多个条件终止:

1. **结束字符检测:** 结束字符是用作指定讯息结束的任何字符。在查找到启动条件后, 接收指令检查每个接收的字符是否匹配结束字符。当接收到结束字符时, 就放进讯息缓冲区, 然后终止接收。

一般地, 使用带有ASCII协议的结束字符检测, 在该协议中每个讯息以指定字符结束。可以使用结束字符检测, 以及字符间计时器、讯息计时器或最大字符计数以终止讯息。

设置:  $ec = 1$ ,  $SMB89/SMB189 = \text{结束字符}$

2. **字符间计时器:** 字符间时间是从一个字符的结束(停止位)到下一个字符的结束(停止位)的时间。如果字符之间的时间(包含第二个字符)超出在SMW92或SMW192中指定的毫秒数, 接收讯息功能终止。字符间计时器在接收到每个字符时重新启动。参见图6-12。

可以使用字符间计时器来终止协议的讯息, 协议没有指定的讯息结束字符。该计时器必须设置数值为大于以所选波特率传送一个字符的时间, 因为此计时器始终包含接收一个完整的字符(起始位、数据位、奇偶校验和停止位)的时间。

可以使用字符间计时器, 以及结束字符检测和最大字符计数以终止讯息。

设置:  $c/m = 0$ ,  $tmr = 1$ ,  $SMW92/SMW192 = \text{超时(毫秒)}$

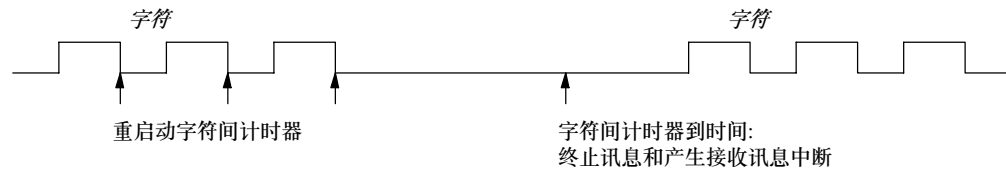


图6-12 使用字符间计时器以终止接收指令

3. **讯息计时器:** 讯息计时器在启动讯息后指定的时间终止讯息。一旦满足接收讯息功能的启动条件, 讯息计时器就启动。当在SMW92或SMW192中指定的毫秒数过去之后, 讯息计时器到时间。参见图6-13。

一般地, 当通讯设备不能保证字符之间有时间间隔, 或当使用调制解调器操作时, 就使用讯息计时器。对于调制解调器, 可以使用讯息计时器来指定允许在讯息启动后接收讯息的最大时间。讯息计时器所需的典型数值约为在选择的波特率下接收最长可能讯息所需时间的1.5倍。

可以使用讯息计时器, 以及结束字符检测和最大字符计数以终止讯息。

设置:  $c/m = 1$ ,  $tmr = 1$ ,  $SMW92/SMW192 = \text{超时(毫秒)}$

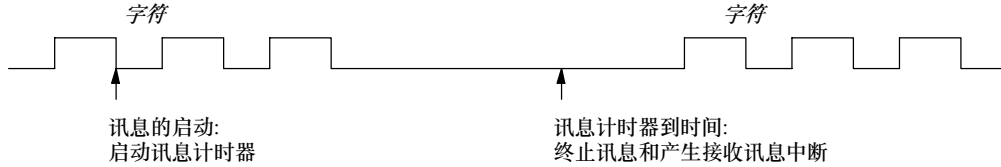


图6-13 使用讯息计时器以终止接收指令

4. **最大字符计数:** 接收指令必须被告之接收的最大字符数 (SMB94或SMB194)。当达到或超过此数值, 接收讯息功能终止。接收指令需要用户指定最大字符计数, 即使这不是特别用作终止条件。这是因为接收指令需要知道接收讯息的最大尺寸, 以便放在讯息缓冲区后的用户数据不被覆盖。

最大字符计数可以用于终止协议的讯息, 在该协议中知道讯息长度并且始终相同。最大字符计数始终与结束字符检测、字符间计时器或讯息计时器结合在一起使用。

5. **奇偶校验出错:** 当硬件发出信号在接收的字符上有奇偶校验出错时, 接收指令自动终止。仅当奇偶校验在SMB30或SMB130中启用时, 奇偶校验出错才可能。无法禁用此功能。
6. **用户终端:** 用户程序可以通过执行另一个在SMB87或SMB187中的启用位 (EN) 设置为零的接收指令来终止接收讯息功能。这立即终止接收讯息功能。

### 使用字符中断控制来接收数据

要允许在协议支持中的完全灵活性, 也可以使用字符中断控制接收数据。每个接收的字符产生中断。接收的字符放在SMB2中, 在刚执行附加在接收字符事件的中断例行程序之前, 奇偶校验状态 (如果启用) 放在SM3.0中。SMB2是自由端口接收字符缓冲区。在自由端口模式下接收的每个字符放在此位置中, 以从用户程序方便地存取。SMB3用于自由端口模式并包含奇偶校验出错位, 当在接收的字符上检测到奇偶校验出错时该位变就被置位。该字节所有其它位保留。使用奇偶校验位以放弃该讯息或产生对该讯息的否认。

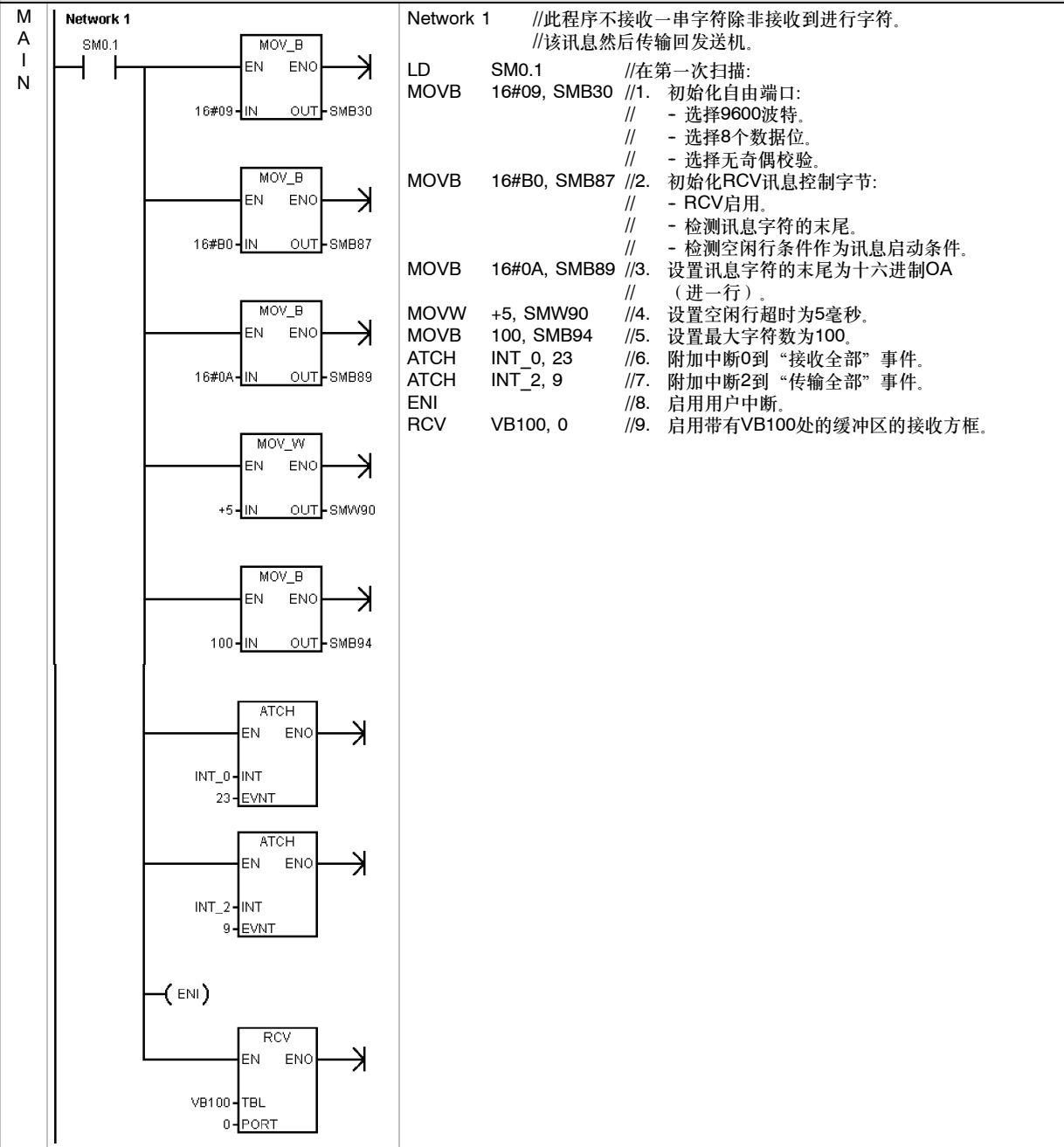
当字符中断用于高波特率时 (38.4千波特到115.2千波特), 中断之间的时间非常短。例如, 38.4千波特的字符中断是260微秒, 57.6千波特为173微秒, 对于115.2千波特则为86微秒。确保保持中断例行程序非常短以避免丢失字符, 否则使用接收指令。



#### 提示

SMB2和SMB3在端口0和端口1之间共享。当接收端口0上的字符导致执行附加在那个事件 (中断事件8) 的中断例行程序时, SMB2包含端口0上接收的字符, 而SMB3包含该字符的奇偶校验状态。当接收端口1上的字符导致执行附加在那个事件 (中断事件25) 的中断例行程序时, SMB2包含端口1上接收的字符, 而SMB3包含该字符的奇偶校验状态。

## 实例：传输和接收指令



实例: 传输和接收指令 (续)		
I N T 0	<p><b>Network 1</b></p>	<p><b>Network 1</b> //接收全部中断例行程序:                  //1. 如果接收状态显示接收结束字符,                  // 那么附加10毫秒计时器触发传输和回流。                  //2. 如果接收因为其它原因完成,                  // 那么启动新的接收。</p> <p>LDB= SMB86, 16#20                  MOVB 10, SMB34                  ATCH INT_1, 10                  CRET I                  NOT                  RCV VB100, 0</p>
I N T 1	<p><b>Network 1</b></p>	<p><b>Network 1</b> //10毫秒计时器中断:                  //1. 释放计时器中断。                  //2. 传输讯息回端口上的用户。</p> <p>LD SM0.0                  DTCH 10                  XMT VB100, 0</p>
I N T 2	<p><b>Network 1</b></p>	<p><b>Network 1</b> //“传输全部”中断:                  //启用其它接收。</p> <p>LD SM0.0                  RCV VB100, 0</p>

## 获得端口地址和设置端口地址指令

“获得端口地址”指令（GPA）读取在“端口”中指定的S7-200 CPU端口的站地址，并将此数值放在ADDR中指定的地址。

“设置端口地址”指令（SPA）设置端口站地址（端口）为在ADDR中指定的数值。新地址不是永久保存。在电源循环后，受影响的端口返回最后一个地址（用系统块下载的那个地址）。

设置ENO = 0的错误条件

- 0006（间接地址）
- 0004（尝试在中断例行程序中完成“设置端口地址”指令）

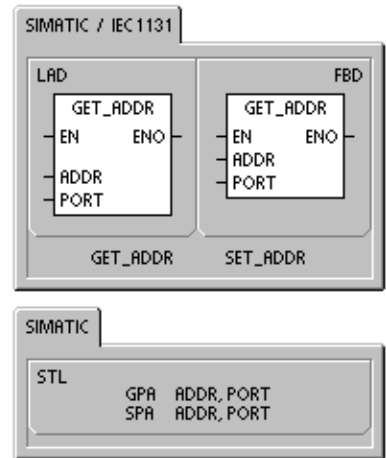


表6-13 “获得端口地址”和“设置端口地址”指令的有效操作数

输入 / 输出	数据类型	操作数
ADDR	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量 (常量数值只对“设置端口地址”指令有效。)
PORT	BYTE	常量 对于CPU 221、CPU 222、CPU 224: 0 对于CPU 226和CPU 226XM: 0或1

## 比较指令

### 比较数字值

比较指令用于比较两个数值：

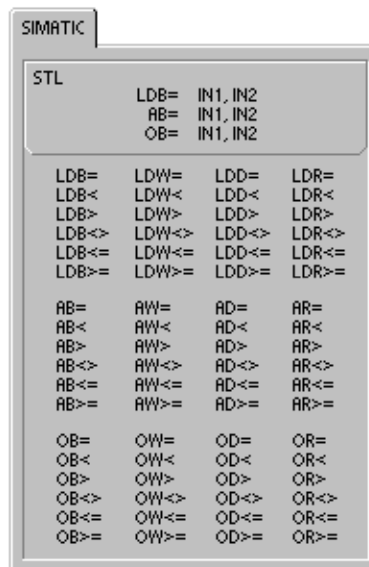
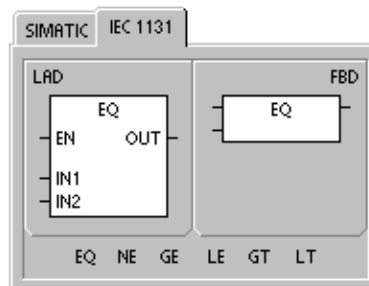
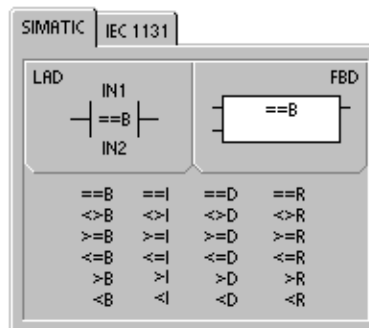
IN1 = IN2    IN1 >= IN2    IN1 <= IN2  
 IN1 > IN2    IN1 < IN2    IN1 <> IN2

比较字节操作是无符号的。比较整数操作是有符号的。  
 比较双字操作是有符号的。  
 比较实数操作是有符号的。

对于梯形图和功能块图表：当比较结果为真时，“比较”指令闭合接点（梯形图）或输出（功能块图表）。

对于语句表：当比较结果为真时，比较指令将1载入堆栈的顶部，再将1与堆栈顶部的值作“与”或者“或”运算（STL）。

当使用IEC比较指令时，可以使用各种数据类型用于输入。然而，输入数值都必须是一样的数据类型。



6

#### 注意

下列条件是致命错误，会引起S7-200立即停止执行程序：

- 遇到非法的间接地址（任何比较指令）
- 遇到非法的实数（例如NAN）（比较实数指令）

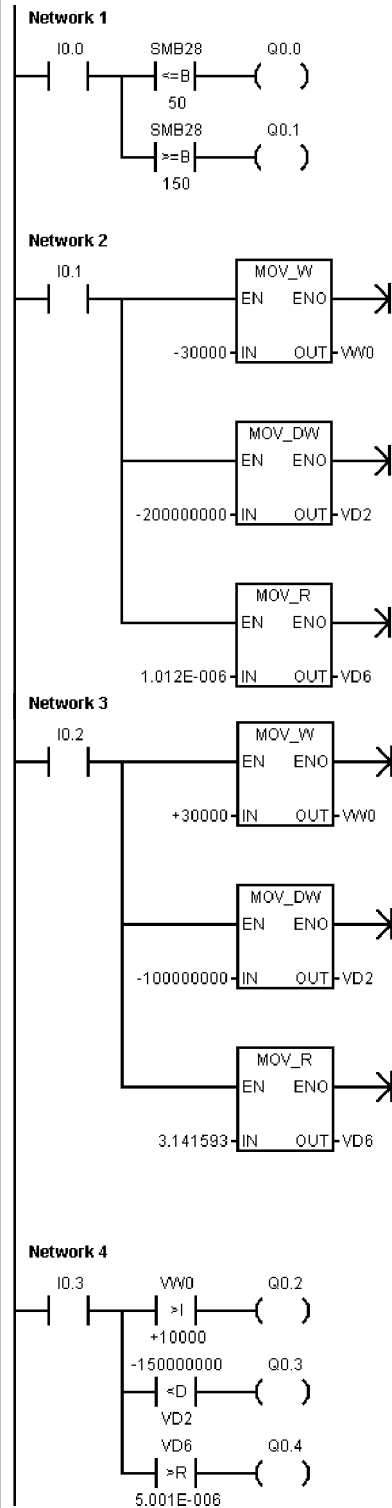
为了防止这些条件产生，在使用这些数值的执行比较指令前确保正确初始化指针和包含实数值。

比较指令执行与功率流的状态无关。

表6-14 比较指令的有效操作数

输入 / 输出	类型	操作数
IN1、IN2	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常量
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常量
Output (或OUT)	BOOL	I、Q、V、M、SM、S、T、C、L、功率流

## 实例: 比较指令



Network 1 //转动模拟调节电位器0以改变SMB28字节数值。  
//当SMB28数值小于等于50时, Q0.0激活。  
//当SMB28数值大于等于150时, Q0.1激活。  
//当比较结果为真时, 状态指示器为on。

```
LD I0.0
LPS
AB<= SMB28, 50
= Q0.0
LPP
AB>= SMB28, 150
= Q0.1
```

Network 2 //用低数值载入V内存地址使比较结果为假,  
//将状态指示器变为off。

```
LD I0.1
MOVW -30000, VW0
MOVD -200000000, VD2
MOVR 1.012E-006, VD6
```

Network 3 //用高数值载入V内存地址使比较结果为真,  
//将状态指示器变为on。

```
LD I0.2
MOVW +30000, VW0
MOVD -100000000, VD2
MOVR 3.141593, VD6
```

Network 4 //整数比较测试查找是否VW0 > +10000为真。  
//使用程序常量显示不同的数据类型。也可以比较存储  
//在可编程内存的两个数值, 象:  
//VW0 > VW100

```
LD I0.3
LPS
AW> VW0, +10000
= Q0.2
LRD
AD< -150000000, VD2
= Q0.3
LPP
AR> VD6, 5.001E-006
= Q0.4
```

## 比较字符串

“比较字符串”指令比较两个ASCII字符的字符串:

IN1 = IN2     IN1 <> IN2

当比较结果为真时,“比较”指令将接点(梯形图)或输出(功能块图表)接通,或比较指令将1载入堆栈顶部的数值,再将1与堆栈顶部的数值作“与”运算、或作“或”运算(语句表)。

### 注意

下列条件是致命错误,会引起S7-200立即停止执行程序:

- 遇到非法的间接地址(任何比较指令)
- 遇到长度大于254个字符的字符串(“比较字符串”指令)
- 字符串的起始地址和长度不适合指定的内存区域(“比较字符串”指令)

为了防止这些条件产生,在使用这些数值执行比较指令前,确保正确初始化指针和期望放置ASCII字符串的内存位置。确保为ASCII字符串保留的缓冲区可以完全驻留在指定的内存区域。

比较指令执行与功率流的状态无关

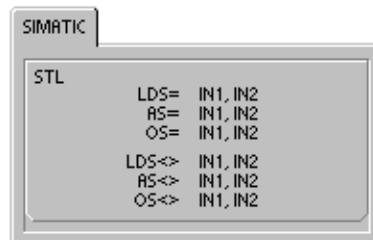
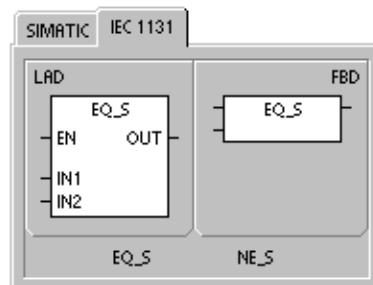
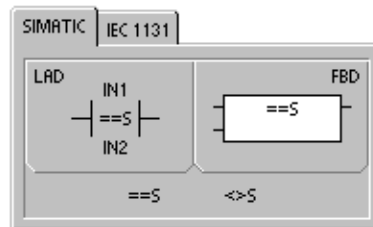


表6-15 比较字符串指令的有效操作数

输入/输出	类型	操作数
IN1、IN2	BYTE (字符串)	VB、LB、*VD、*LD、*AC
Output (OUT)	BOOL	I、Q、V、M、SM、S、T、C、L、功率流



## 转换指令

### 标准转换指令

#### 数字转换

字节至整数（BTI）、整数至字节（ITB）、整数至双整数（ITD）、双整数至整数（DTI）、双整数至实数（DTR）、BCD至整数（BCDI）和整数至BCD（IBCD）指令将输入数值IN转换到指定的格式，并将输出值存储在OUT指定的内存位置。例如，可以将双整数值转换到实数。也可以在整数和BCD码格式之间转换。

#### 进位和截尾

“进位”指令（ROUND）将实数IN转换为双整数值，并将四舍五入结果放入OUT指定的变量中。

“截尾”指令（TRUNC）将实数IN转换为双整数，并把结果的整数部分放到OUT指定的变量中。

#### 分段

段指令（SEG）允许产生位模式说明七段显示的段。

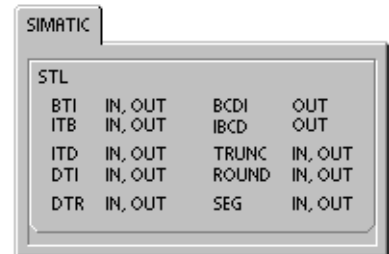
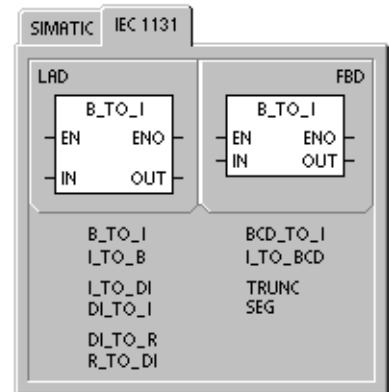
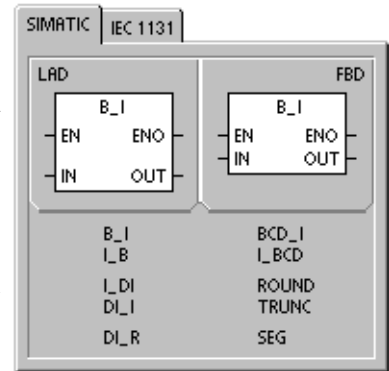


表6-16 标准转换指令的有效操作数

输入 / 输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
	WORD、INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、AC、*VD、*LD、*AC、常量
	DINT	ID、QD、VD、MD、SMD、SD、LD、HC、AC、*VD、*LD、*AC、常量
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常量
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC
	WORD、INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、AC、*VD、*LD、*AC
	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

### BCD至整数和整数至BCD的操作指令

BCD至整数指令（BCDI）将二进制编码的十进制数值IN转换为整数，并将结果载入OUT指定的变量。IN的有效范围是0到9999BCD码。

整数至BCD指令（IBCD）将输入的整数IN转换为二进制编码的十进制数值，并将结果载入OUT指定的变量。IN的有效范围是0到9999整数。

设置ENO = 0的错误条件

- SM1.6（无效的BCD码）
- 0006（间接地址）

受影响的SM位:

- SM1.6（无效的BCD码）

### 双整数至实数的操作指令

双整数至实数指令（DTR）将32位、有符号整数IN转换为32位实数，并将结果放在OUT指定的变量中。

设置ENO = 0的错误条件

- 0006（间接地址）

### 双整数至整数的操作指令

双整数至整数指令（DTI）将双整数值IN转换为整数值，并将结果放在OUT指定的变量中。

如果转换的数值太大不能在输出中表示，那么会设置溢出位，输出不受影响。

设置ENO = 0的错误条件

- SM1.1（溢出）
- 0006（间接地址）

受影响的SM位:

- SM1.1（溢出）

### 整数至双整数的操作指令

整数至双整数指令（ITD）将整数值IN转换为双整数值，并将结果放入OUT指定的变量中。扩展符号。

设置ENO = 0的错误条件

- 0006（间接地址）

### 字节至整数的操作指令

字节至整数指令（BTI）将字节数值IN转换为整数值，并将结果放入OUT指定的变量中。字节是无符号的，因此没有符号扩展。

设置ENO = 0的错误条件

- 0006（间接地址）

### 整数至字节的操作指令

整数至字节指令（ITB）将数值IN转换为字节数值，并将结果放入OUT指定的变量中。转换数值0到255。导致溢出和输出的所有其它数值不受影响。

设置ENO = 0的错误条件

- SM1.1（溢出）
- 0006（间接地址）

受影响的SM位:

- SM1.1（溢出）



#### 提示

要将整数转变为实数，使用整数至双整数指令，然后使用双整数至实数指令。

### 操作进位和截尾指令

“进位”指令（ROUND）将实数IN转换为双整数值，并将结果放入OUT指定的变量中。如果小数部分为0.5或以上，数字向上进位。

“截尾”指令（TRUNC）将实数IN转换为双整数，并将结果放入OUT指定的变量中。只有实数的整数部分被转换，小数被丢弃。

设置ENO = 0的错误条件

- SM1.1（溢出）
- 0006（间接地址）

受影响的SM位:

- SM1.1（溢出）

如果转换的数值不是有效的实数或太大不能在输出中表示，那么会设置溢出位，而输出不受影响。

**实例：标准转换指令**

```

Network 1 //转换英寸到厘米:
//1. 将计数器数值（英寸）载入AC1.
//2. 将数值转换为实数.
//3. 乘以2.54（转换为厘米）。
//4. 将数值转换回整数.
LD      I0.0
ITD     C10, AC1
DTR     AC1, VD0
MOVR    VD0, VD8
*R      VD4, VD8
ROUND   VD8, VD12

Network 2 //转换BCD码数值到整数
LD      I0.3
BCDI    AC0
        
```

**双整数到实数和进位**

C10	101	计数 = 101英寸
VD0	101.0	计数（作为实数）
VD4	2.54	2.54常量（英寸到厘米）
VD8	256.54	256.54厘米作为实数
VD12	257	257厘米作为双整数

**BCD码转换到整数**

AC0	1234
	BCDI
AC0	04D2

### 段操作指令

为说明七段显示的段，段指令（SEG）将转换IN指定的字符（字节）以在OUT指定的位置产生位模式（字节）。

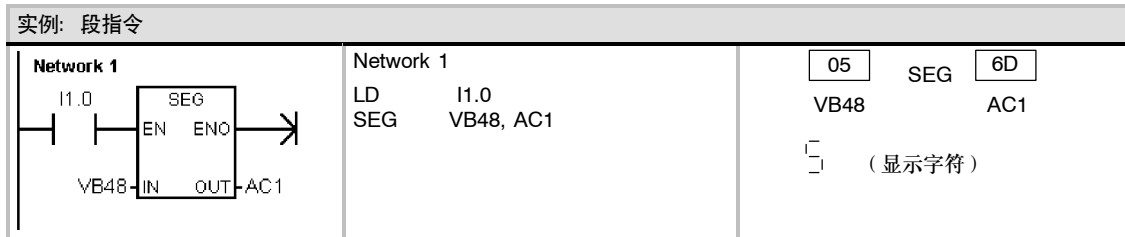
显示的段代表了输入字节的最低位中的字符。图6-14显示段指令使用的七段显示编码。

- 0006 (间接地址)

(IN) LSD	段 显示	(OUT) - g f e d c b a
0	□	0 0 1 1 1 1 1 1
1		0 0 0 0 0 1 1 0
2	□	0 1 0 1 1 0 1 1
3	□	0 1 0 0 1 1 1 1
4	4	0 1 1 0 0 1 1 0
5	□	0 1 1 0 1 1 0 1
6	□	0 1 1 1 1 1 0 1
7		0 0 0 0 0 1 1 1

(IN) LSD	段 显示	(OUT) - g f e d c b a
8	□	0 1 1 1 1 1 1 1
9	□	0 1 1 0 0 1 1 1
A	□	0 1 1 1 0 1 1 1
B	□	0 1 1 1 1 1 0 0
C	□	0 0 1 1 1 0 0 1
D	□	0 1 0 1 1 1 1 0
E	□	0 1 1 1 1 0 0 1
F	□	0 1 1 1 0 0 0 1

图6-14 七段显示的编码



## ASCII转换指令

有效的ASCII字符是十六进制数值30到39和41到46。

### 在ASCII和十六进制数值之间转换

“ASCII到十六进制”指令（ATH）将以IN开始的ASCII字符的数字LEN转换为以OUT开始的十六进制数字。“十六进制到ASCII”指令（HTA）将以输入字节IN开始的十六进制数字转换为以OUT开始的ASCII字符。要转换的十六进制数字的数目由长度LEN指定。

可转换的ASCII字符或十六进制数字的最大数目是255。

设置ENO = 0的错误条件

- SM1.7（非法的ASCII）仅ASCII到十六进制
- 0006（间接地址）
- 0091（操作数超出范围）

受影响的SM位:

- SM1.7（非法的ASCII）

### 转换数字值到ASCII

“整数到ASCII”（ITA）、“双整数到ASCII”（DTA）和“实数到ASCII”（RTA）指令用于转换整数、双整数或实数值到ASCII字符。

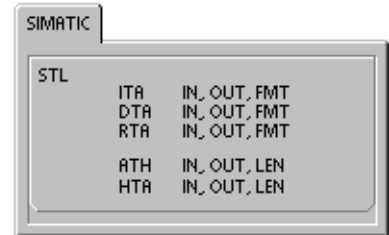
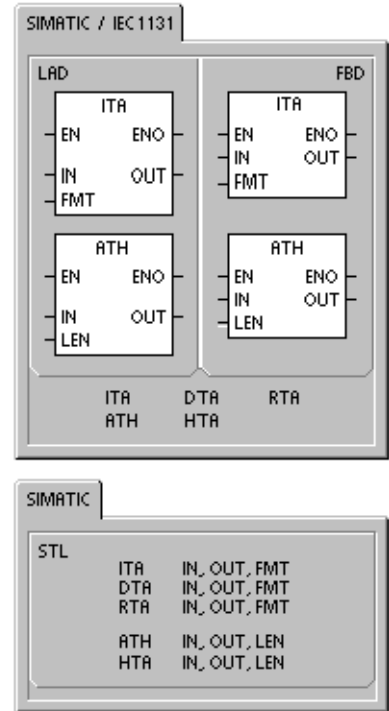


表6-17 ASCII转换指令的有效操作数

输入 / 输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC
	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常量
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常量
LEN、FMT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC

### 操作整数到ASCII指令

“整数到ASCII”指令（ITA）将整数IN转换为ASCII字符的数组。格式FMT指定小数点右边的转换精度，以及小数点以逗号还是以句号显示。转换结果放在以OUT开头的8个连续的字节中。

设置ENO = 0的错误条件

- 0006（间接地址）
- 非法的格式
- $nnn > 5$

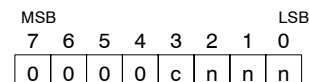
ASCII字符的数组始终是8个字符。

图6-15描述用于“整数到ASCII”指令的格式操作数。输出缓冲区的大小始终为8个字节。输出缓冲区中的小数点右边的位数由 $nnn$ 域指定。 $nnn$ 域的有效范围是0到5。将小数点右边的位数指定为0会导致显示的数值没有小数点。对于 $nnn$ 的数值大于5，输出缓冲区用ASCII空格填充。 $c$ 位指定是使用逗号（ $c=1$ ）还是小数点（ $c=0$ ）作为整数和小数之间的分隔符。上面4位必须为零。

图6-15显示格式化的数值的实例，使用小数点（c=0），小数点右边有三个数字（nnn=011）。输出缓冲区根据下列规则格式化：

- 正数值写入至输出缓冲区而没有符号。
- 负数值前面带负号（-）写入至输出缓冲区。
- 小数点左边的先行零（除邻近小数点的数字外）被排除。
- 输出缓冲区中的数值右对齐。

FMT



c = 逗号 (1) 或小数点 (0)  
 nnn = 小数点右边的位数

	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7
in=12			0	.	0	1	2
in=-123		-	0		1	2	3
in=1234			1	.	2	3	4
in = -12345	-	1	2	.	3	4	5

图6-15 用于“整数到ASCII”（ITA）指令的FMT操作数

6

双整数至ASCII的操作指令

“双整数到ASCII”（DTA）指令将双字IN转换到ASCII字符的数组。格式操作数FMT指定小数点右边的转换精度。转换结果放在以OUT开头的12个连续的字节中。

设置ENO = 0的错误条件

- 0006（间接地址）
- 非法的格式
- nnn > 5

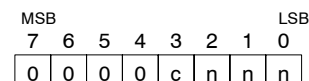
输出缓冲区的大小始终为12个字节。

图6-16描述用于双整数到ASCII指令的格式操作数。输出缓冲区中的小数点右面的位数由nnn域指定。nnn域的有效范围是0到5。将小数点右边的位数指定为0会导致显示的数值没有小数点。对于nnn的数值大于5，输出缓冲区用ASCII空格填充。c位指定是使用逗号（c=1）还是小数点（c=0）作为整数和小数之间的分隔符。上面的4位必须为零。

图6-16显示格式化的数值的实例，使用小数点（c=0），小数点右边有四个数字（nnn=100）。输出缓冲区根据下列规则格式化：

- 正数值不带符号写入输出缓冲区。
- 负数值前面带负号（-）写入输出缓冲区。
- 小数点左边的先行零（除邻近小数点的数字外）被排除。
- 输出缓冲区中的数值右对齐。

FMT



c = 逗号 (1) 或小数点 (0)  
 nnn = 小数点右边的位数

	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7	Out +8	Out +9	Out +10	Out +11
in=-12					-	0		0	0	1	2
in=1234567				1	2	3	.	4	5	6	7

图6-16 用于“双整数到ASCII”（DTA）指令的FMT操作数

### “实数到ASCII”的操作指令

“实数到ASCII”指令（RTA）将实数值IN转换为ASCII字符。格式FMT指定小数点右边的转换精度，小数点以逗号还是以句号显示，以及输出缓冲区大小。

设置ENO = 0的错误条件

- 0006（间接地址）
- nnn > 5
- ssss < 3
- ssss < OUT中的字符数

转换结果放在以OUT开头的输出缓冲区中。

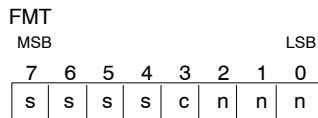
结果ASCII字符的数目（或长度）是输出缓冲区的大小，可以指定为从3个字节到15个字节或字符的大小范围。

S7-200使用的实数格式支持最大7个有效数字。尝试显示大于7个有效数字会产生进位错误。

图6-17描述RTA指令的格式操作数（FMT）。输出缓冲区的大小由ssss域指定。大小为0、1或2个字节无效。输出缓冲区中的小数点右面的位数由nnn域指定。nnn域的有效范围是0到5。将小数点右边的位数指定为0会导致显示的数值没有小数点。对于nnn数值大于5或当指定的输出缓冲区太小而不能存储转换的数值时，输出缓冲区用ASCII空格填充。c位指定是使用逗号（c=1）还是小数点（c=0）作为整数和小数之间的分隔符。

图6-17也显示了数值的实例，数值的格式使用小数点（c=0），小数点右面的数字是一位（nnn=001），缓冲区大小为六个字节（ssss=0110）。输出缓冲区根据下列规则格式化：

- 正数值不带符号写入至输出缓冲区。
- 负数值前面带负号（-）写入输出缓冲区。
- 小数点左边的先行零（除邻近小数点的数字外）被排除。
- 小数点右面的数值被进位了，以适合小数点右面的指定的位数。
- 输出缓冲区的大小必须最小大于小数点右面位数三个字节。
- 输出缓冲区中的数值右对齐。



ssss = 输出缓冲区的大小  
 c = 逗号（1）或小数点（0）  
 nnn = 小数点右面的位数

	Out	Out	Out	Out	Out	Out
	+1	+2	+3	+4	+5	
in = 1234.5	1	2	3	4		5
in = -0.0004				0	.	0
in = -3.67526			-	3	.	7
in = 1.95				2	.	0

图6-17 “实数到ASCII”（RTA）指令的FMT操作数

**实例: “ASCII到十六进制” 指令**

<p><b>Network 1</b></p>	<p><b>Network 1</b></p> <pre>LD I3.2 ATH VB30, VB40, 3</pre>																			
<table border="0"> <tr> <td>'3'</td><td>'E'</td><td>'A'</td><td></td><td></td><td></td></tr> <tr> <td>33</td><td>45</td><td>41</td><td>ATH</td><td>3E</td><td>Ax</td></tr> <tr> <td colspan="3">VB30</td><td></td><td colspan="3">VB40</td></tr> </table>	'3'	'E'	'A'				33	45	41	ATH	3E	Ax	VB30				VB40			<p>注意: X指示“半字节”(字节的一半)不变。</p>
'3'	'E'	'A'																		
33	45	41	ATH	3E	Ax															
VB30				VB40																

**实例: “整数到ASCII” 指令**

<p><b>Network 1</b></p>	<p><b>Network 1</b></p> <pre>//将VW2处的整数值转换为以VB10开始的 //8位ASCII字符, //使用16#0B格式(逗号作为小数点,后跟3位小数)。 LD I2.3 ITA VW2, VB10, 16#0B</pre>																														
<table border="0"> <tr> <td></td><td></td><td></td><td></td><td>'1'</td><td>'2'</td><td>'.'</td><td>'3'</td><td>'4'</td><td>'5'</td></tr> <tr> <td>12345</td><td>ITA</td><td>20</td><td>20</td><td>31</td><td>32</td><td>2C</td><td>33</td><td>34</td><td>35</td></tr> <tr> <td>VW2</td><td></td><td colspan="2">VB10</td><td>VB11</td><td>...</td><td></td><td></td><td></td><td></td></tr> </table>					'1'	'2'	'.'	'3'	'4'	'5'	12345	ITA	20	20	31	32	2C	33	34	35	VW2		VB10		VB11	...					
				'1'	'2'	'.'	'3'	'4'	'5'																						
12345	ITA	20	20	31	32	2C	33	34	35																						
VW2		VB10		VB11	...																										

**实例: “实数到ASCII” 指令**

<p><b>Network 1</b></p>	<p><b>Network 1</b></p> <pre>//将VD2处的实数值转换为以VB10开始的 //10位ASCII字符, //使用16#A3格式(句号作为小数点,后跟3位小数。) LD I2.3 RTA VD2, VB10, 16#A3</pre>																																			
<table border="0"> <tr> <td></td><td></td><td></td><td></td><td>'1'</td><td>'2'</td><td>'3'</td><td>'.'</td><td>'4'</td><td>'5'</td><td>'0'</td></tr> <tr> <td>123.45</td><td>RTA</td><td>20</td><td>20</td><td>20</td><td>31</td><td>32</td><td>33</td><td>2E</td><td>34</td><td>35</td><td>30</td></tr> <tr> <td>VD2</td><td></td><td colspan="2">VB10</td><td>VB11</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>					'1'	'2'	'3'	'.'	'4'	'5'	'0'	123.45	RTA	20	20	20	31	32	33	2E	34	35	30	VD2		VB10		VB11	...							
				'1'	'2'	'3'	'.'	'4'	'5'	'0'																										
123.45	RTA	20	20	20	31	32	33	2E	34	35	30																									
VD2		VB10		VB11	...																															



## 字符串转换指令

### 转换数字值到字符串

“整数到字符串”（ITS）、“双整数到字符串”（DTS）和“实数到字符串”（RTS）指令将整数、双整数或实数值（IN）转换为ASCII字符串（OUT）。

### 整数到字符串的操作

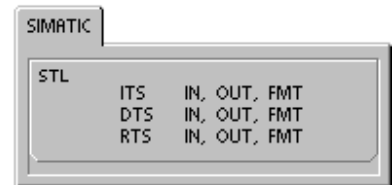
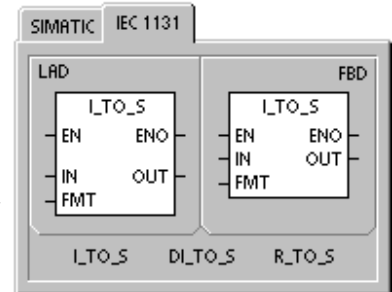
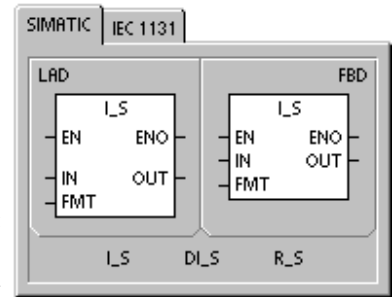
整数到字符串指令（ITS）将整数IN转换到带有8个字符长度的ASCII字符串。格式（FMT）指定小数点右边的转换精度，以及小数点以逗号还是以句号显示。结果字符串被写入以OUT开始的9个连续的字节。关于更多信息参见第4章中字符串格式部分。

#### 设置ENO = 0的错误条件

- 0006（间接地址）
- 0091（操作数超出范围）
- 非法的格式（nnn > 5）

图6-18描述用于“整数到字符串”指令的格式操作数。输出字符串的长度始终为8个字符。输出缓冲区中的小数点右面的位数由nnn域指定。nnn域的有效范围是0到5。将小数点右边的位数指定为0会导致显示的数值没有小数点。对于nnn大于5的数值，输出是8个ASCII空格字符的字符串。c位指定是使用逗号（c=1）还是小数点（c=0）作为整数和小数之间的分隔符。该格式的上4位必须为零。

图6-18也显示了数值的实例，数值的格式使用小数点（c=0），小数点右面的数字是三位（nnn=011）。位于OUT处的数值是字符串的长度。



输出字符串根据下列规则格式化:

- 正数值不带符号写入输出缓冲区。
- 负数值前面带负号（-）写入输出缓冲区。
- 小数点左边的先行零（除邻近小数点的数字外）被排除。
- 输出字符串中的数值右对齐。

表6-18 “转换数字值到字符串”指令的有效操作数

输入 / 输出	数据类型	操作数
IN	BYTE (字符串)	VB、LB、*VD、*LD、*AC
	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、*VD、*LD、*AC、常量
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常量
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常量
FMT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
OUT	BYTE (字符串)	VB、LB、*VD、*LD、*AC
	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC
	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

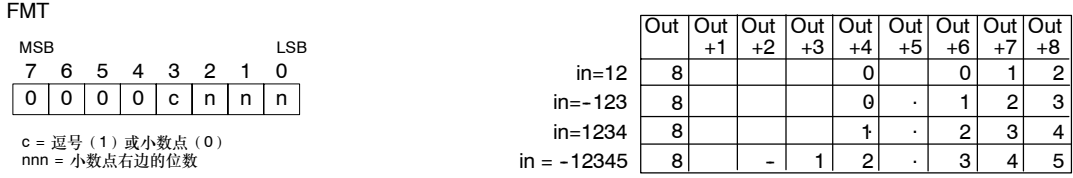


图6-18 整数到字符串指令的FMT操作数

### 双整数到字符串的操作

双整数到字符串指令 (DTS) 将双整数IN转换到带有12个字符长度的ASCII字符串。格式 (FMT) 指定小数点右边的转换精度，以及小数点以逗号还是以句号显示。结果字符串被写入以OUT开始的13个连续的字节。关于更多信息，参见第4章中描述字符串格式的部分。

设置ENO = 0的错误条件

- 0006 (间接地址)
- 0091 (操作数超出范围)
- 非法的格式 (nnn > 5)

图6-19描述用于“整数到字符串”指令的格式操作数。输出字符串的长度始终为8个字符。输出缓冲区中的小数点右面的位数由nnn域指定。nnn域的有效范围是0到5。将小数点右边的位数指定为0会导致显示的数值没有小数点。对于nnn大于5的数值，输出是12个ASCII空格字符的字符串。c位指定是使用逗号 (c=1) 还是小数点 (c=0) 作为整数和小数之间的分隔符。格式的上位4位必须为零。

图6-19也显示数值的实例，这些数值的格式使用小数点 (c=0)，小数点右边有四个数字 (nnn=100)。位于OUT处的数值是字符串的长度。输出字符串根据下列规则格式化：

- 正数值不带符号写入输出缓冲区。
- 负数值前面带负号 (-) 写入输出缓冲区。
- 小数点左边的先行零 (除邻近小数点的数字外) 被排除。
- 输出字符串中的数值右对齐。

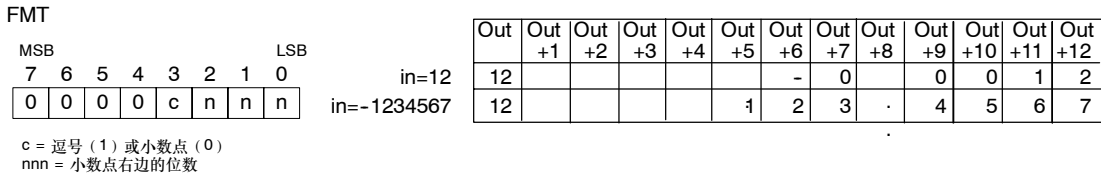


图6-19 双整数到字符串指令的FMT操作数

### 实数到字符串的操作

“实数到字符串”指令（RTS）将实数值IN转换为ASCII字符串。格式（FMT）指定小数点右边的转换精度，小数点以逗号还是以句号显示，以及输出字符串的长度。

转换结果放在以OUT开头的字符串中。结果字符串的长度在格式中指定，可以是3到15个字符。关于更多信息，参见第4章中描述字符串格式的部分。

- 设置ENO = 0的错误条件
- 0006（间接地址）
  - 0091（操作数超出范围）
  - 非法的格式
    - nnn > 5
    - ssss < 3
    - ssss < 所需字符数

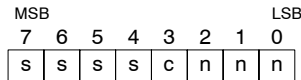
S7-200使用的实数格式支持最大7个有效数字。尝试显示大于7个有效数字会产生进位错误。

图6-20描述用于“实数到字符串”指令的格式操作数。输出字符串的长度由ssss域指定。大小为0、1或2个字节无效。输出缓冲区中的小数点右面的位数由nnn域指定。nnn域的有效范围是0到5。将小数点右边的位数指定为0会导致显示的数值没有小数点。当nnn大于5或当输出字符串的指定长度太小以致不能存储转换的数值时，输出字符串用ASCII空格字符填充。c位指定是使用逗号（c=1）还是小数点（c=0）作为整数和小数之间的分隔符。

图6-20也显示了数值的实例，数值的格式使用小数点（c=0），小数点右面的数字是一位（nnn=001），输出字符串长度为6个字符（ssss = 0110）。OUT处的数值是字符串的长度。输出字符串根据下列规则格式化：

- 正数值不带符号写入输出缓冲区。
- 负数值前面带负号（-）写入输出缓冲区。
- 小数点左边的先行零（除邻近小数点的数字外）被排除。
- 小数点右面的数值进位了，以适合小数点右面的指定的位数。
- 输出字符串的大小最小必须大于小数点右面位数三个字节。
- 输出字符串中的数值右对齐。

#### FMT



ssss = 输出字符串的长度  
 c = 逗号（1）或小数点（0）  
 nnn = 小数点右面的位数

	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6
in = 1234.5	6	1	2	3	4	5
in = -0.0004	6			0	.	0
in = -3.67526	6		-	3	.	7
in = 1.95	6			2	.	0

图6-20 “实数到字符串指令”的FMT操作数

## 转换子字符串到数字值

“子字符串到整数”（STI）、“子字符串到双整数”（STD）和“子字符串到实数”（STR）指令将以偏移量INDX开始的字符串数值IN转换为整数、双整数或实数值OUT。

设置ENO = 0的错误条件

- 0006（间接地址）
- 0091（操作数超出范围）
- 009B（下标 = 0）
- SM1.1（溢出）

“子字符串到整数”和“子字符串到双整数”用下列形式转换字符串：  
[空格][+或-][数字0 - 9]

“子字符串到实数”指令以下列形式转换字符串：  
[空格][+或-][数字0 - 9].[或,][数字0 - 9]

INDX数值常规设置为1，它用字符串的第一个字符开始转换。INDX数值可以设置为其它数值，以字符串中的不同点开始转换。当输入字符串包含不是要转换的数字部分的文本时，这可以使用。例如，如果输入字符串是“Temperature: 77.8”，设置INDX为数值13以跳过在字符串开始的字“Temperature:”。

“子字符串到实数”指令不转换使用科学计数制或指数形式的实数的字符串。该指令不产生溢出错误（SM1.1），但将字符串转换到实数的指数前，然后终止转换。例如，字符串‘1.234E6’无错地转换到实数值1.234。

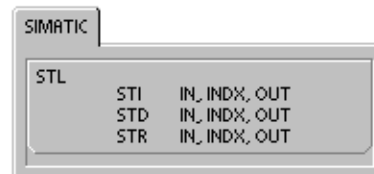
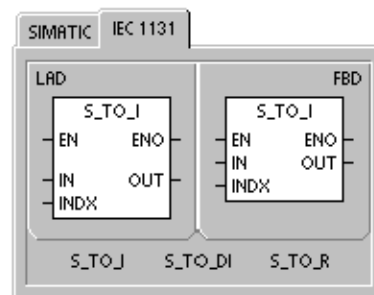
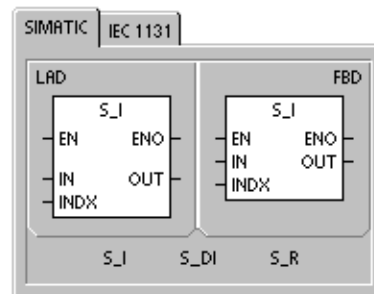
当字符串的末尾到达或发现第一个无效的字符时，转换终止。无效的字符是任何除数字（0 - 9）外的字符。

只要转换产生对于输出值来说太大的整数值，就要设置溢出错误（SM1.1）。例如，如果输入字符串产生大于32767或小于-32768的数值，“子字符串到整数”指令设置溢出错误。

如果输入字符串不包含有效的数值而不能转换，也设置溢出错误（SM1.1）。例如，如果输入字符串包含‘A123’，转换指令设置SM1.1（溢出），输出值保持不变。

表6-19 “转换子字符串到数字值”指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE（字符串）	IB、QB、VB、MB、SMB、SB、LB、*VD、*LD、*AC、常量
INDX	BYTE	VB、IB、QB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
OUT	BYTE（字符串）	VB、IB、QB、MB、SMB、SB、LB、*VD、*LD、*AC、常量
	INT DINT、REAL	VW、IW、QW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC VD、ID、QD、MD、SMD、SD、LD、AC、*VD、*LD、*AC



整数和双整数的有效输入字符串

输入字符串	输出整数
'123'	123
'-00456'	-456
'123.45'	123
'+2345'	2345
'000000123ABCD'	123

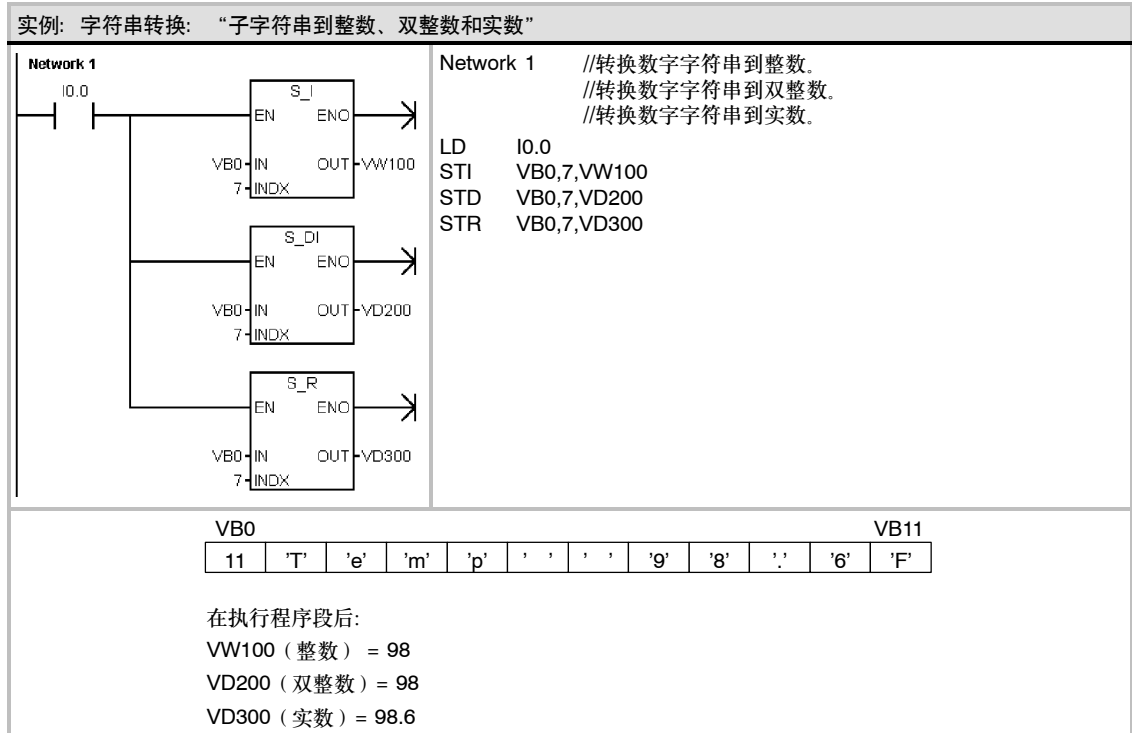
实数的有效输入字符串

输入字符串	输出实数
'123'	123.0
'-00456'	-456.0
'123.45'	123.45
'+2345'	2345.0
'00.000000123'	0.000000123

无效输入字符串

输入字符串
'A123'
' '
'++123'
'+-123'
'+ 123'

图6-21 有效和无效输入字符串实例



## 编码和解码指令

### 编码

“编码”指令（ENCO）将输入字IN的最低位设置的位号写入输出字节OUT的最低有效“半字节”（4位）。

### 解码

“解码”指令（DECO）设置输出字OUT中的位，该位相当于输入字节IN的最低有效“半字节”（4位）代表的位号。输出字的所有其它位设置为0。

### SM位和ENO

对于“编码”和“解码”指令，下列条件影响ENO。

设置ENO = 0的错误条件

- 0006（间接地址）

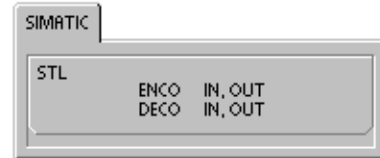
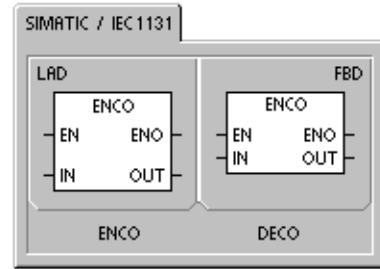
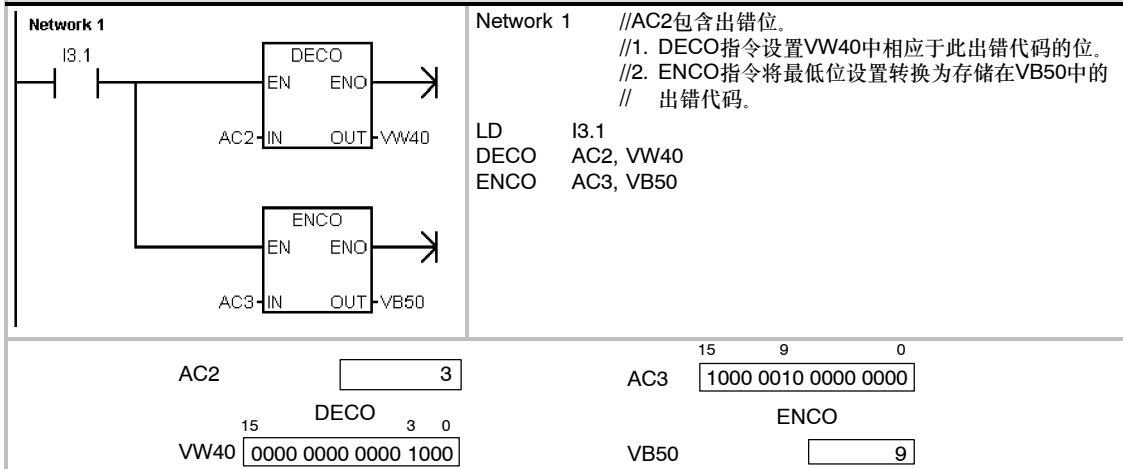


表6-20 “编码和解码指令”的有效操作数

输入 / 输出	数据类型	操作数
IN	BYTE WORD	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量 IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
OUT	BYTE WORD	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC

### 实例：“编码和解码指令”



## 计数器指令

### SIMATIC计数器指令

#### 向上计数计数器

“向上计数”指令（CTU）每当向上计数（CU）输入从断开变为接通时从当前值向上计数。当当前值Cxx大于等于预设值PV，计数器位Cxx接通。当“重设”（R）输入接通，或当执行“重设”指令时，计数器重设。当它达到最大值（32,767）时，计数器停止计数。

语句表操作：

- 重新设定输入：堆栈顶
- “向上计数”输入：在第二堆栈位置载入数值

#### “向下计数计数器”

“向下计数”指令（CTD）每当向下计数（CD）输入从断开变接通时从该计数器的当前值向下计数。当当前值Cxx等于0时，计数器位Cxx接通。当载入输入LD接通时，计数器重设计数器位Cxx并用预设值PV载入当前值。计数器在到达零时停止，并且计数器位Cxx接通。

语句表操作：

- 载入输入：堆栈顶
- “向下计数”输入：在第二堆栈位置载入的数值。

#### “向上 / 向下计数计数器”

“向上 / 向下计数”指令（CTUD）每当向上计数（CU）输入从断开变为接通时向上计数，每当向下计数（CD）输入从断开变为接通时向下计数。计数器的当前值Cxx保持当前计数。每次执行计数器指令时，将预设值PV与当前值比较。

当达到最大数值（32,767），下一个向上计数输入处的上升沿使当前计数绕回最小数值（-32,768）。当达到最小数值（-32,768），下一个向下计数输入处的上升沿使当前计数绕回最大数值（32,767）。

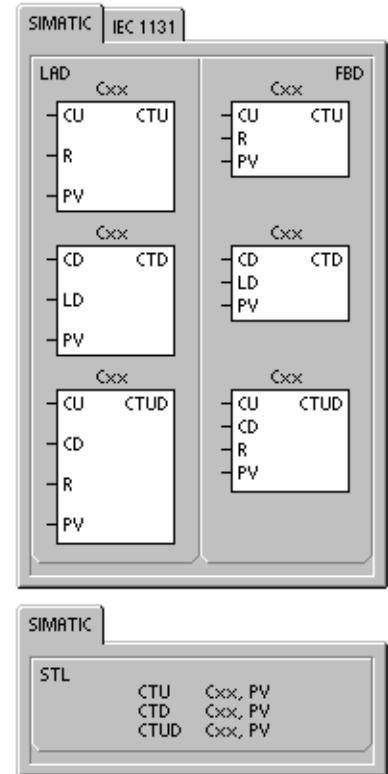
当当前值Cxx大于等于预设值PV，计数器位Cxx接通。否则，计数器位断开。当“重设”（R）输入接通，或当执行“重设”指令时，计数器重设。当CTUD计数器达到PV时，停止计数。

语句表操作：

- 重新设定输入：堆栈顶
- “向下计数”输入：在第二堆栈位置载入的数值
- “向上计数”输入：在第三堆栈位置载入的数值

表6-21 SIMATIC计数器指令的有效操作数

输入 / 输出	数据类型	操作数
Cxx	WORD	常量（C0到C255）
CU、CD、LD、R	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
PV	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常量





## 提示

因为每个计数器有一个当前值，不要分配同样的数字给一个以上计数器。（具有同样数字的“向上计数器”、“向上/向下计数器”和“向下计数器”存取同样的当前值。）

当使用“重设”指令重设计数器时，计数器位被重设，计数器当前值设置为零。使用计数器号码参考该计数器的当前值和计数器位。

表6-22 计数器的操作指令

类型	操作	计数器位	电源循环 / 第一次扫描
CTU	CU增加当前值。 当前值继续增加直到达到32,767。	计数器位接通当： 当前值 >= 预置	计数器位断开。 可以保持当前值。 <sup>1</sup>
CTUD	CU增加当前值。 CD减少当前值。 当前值继续增加或减少直到计数器重设。	计数器位接通当： 当前值 >= 预置	计数器位断开。 可以保持当前值。 <sup>1</sup>
CTD	CD减少当前值直到当前值达到0。	计数器位接通当： 当前值 = 0	计数器位断开。 可以保持当前值。 <sup>1</sup>

<sup>1</sup> 可以选择计数器要保留的当前值。关于S7-200 CPU的内存保留的信息参见第4章。

## 实例: SIMATIC向下计数计数器指令

```

Network 1 //向下计数计数器C1当前值从3计数到0, I0.1 断开,
//I0.0 断开-接通减少C1的当前值
//I0.1 接通载入向下计数预设值3
LD I0.0
LD I0.1
CTD C1, +3

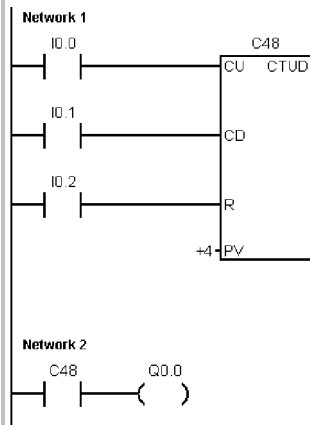
Network 2 //当计数器C1当前值 = 0时C1位接通
LD C1
= Q0.0

```

计时图



## 实例: SIMATIC向上 / 向下计数计数器指令



```

Network 1 //I0.0向上计数
           //I0.1向下计数
           //I0.2重设当前值为0
LD      I0.0
LD      I0.1
LD      I0.2
CTUD   C48, +4

```

```

Network 2 //向上 / 向下计数计数器C48接通C48位
           //当当前值>= 4时

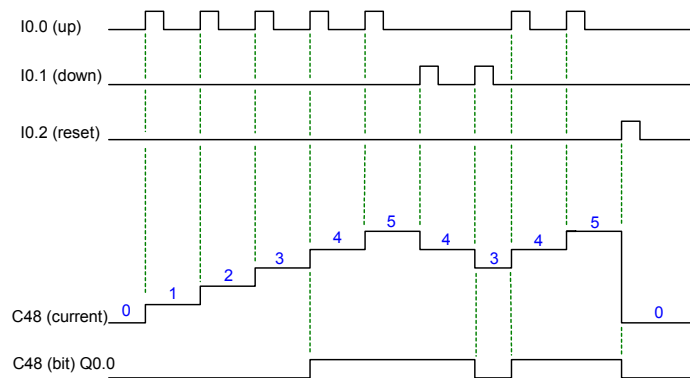
```

```

LD      C48
=       Q0.0

```

## 计时图



## IEC计数器指令

### 向上计数器

“向上计数”指令（CTU）在“向上计数”（CU）输入的上升沿时从当前值向上计数到预设值（PV）。当当前值（CV）大于等于预设值时，计数器输出位（Q）接通。当重设输入（R）启用时，计数器重设。当它达到预设值时，向上计数器停止计数。

### 向下计数器

“向下计数”指令（CTD）在向下计数（CD）输入的上升沿从预设值（PV）向下计数。当当前值（CV）等于零时，计数器输出位（Q）接通。当载入输入（LD）启用时，计数器重设并用预设值载入当前值。当向下计数器达到零时，停止计数。

### 向上 / 向下计数器

“向上 / 向下计数”指令（CTUD）在向上计数（CU）或向下计数（CD）输入的上升沿从当前值（CV）向上或向下计数。当当前值等于预置时，向上输出（QU）接通。在当前值等于零时，向下输出（QD）接通。当载入（LD）输入启用时，计数器用预设值（PV）载入当前值。相似地，当重设（R）启用时，计数器重设并用0载入当前值。当计数器达到预置或0时，停止计数。

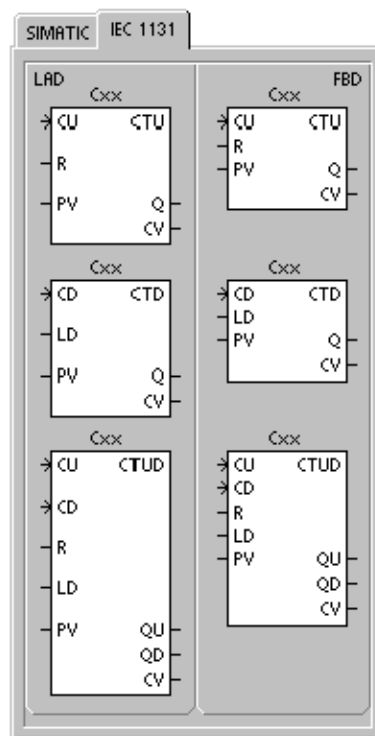


表6-23 IEC计数器指令的有效操作数

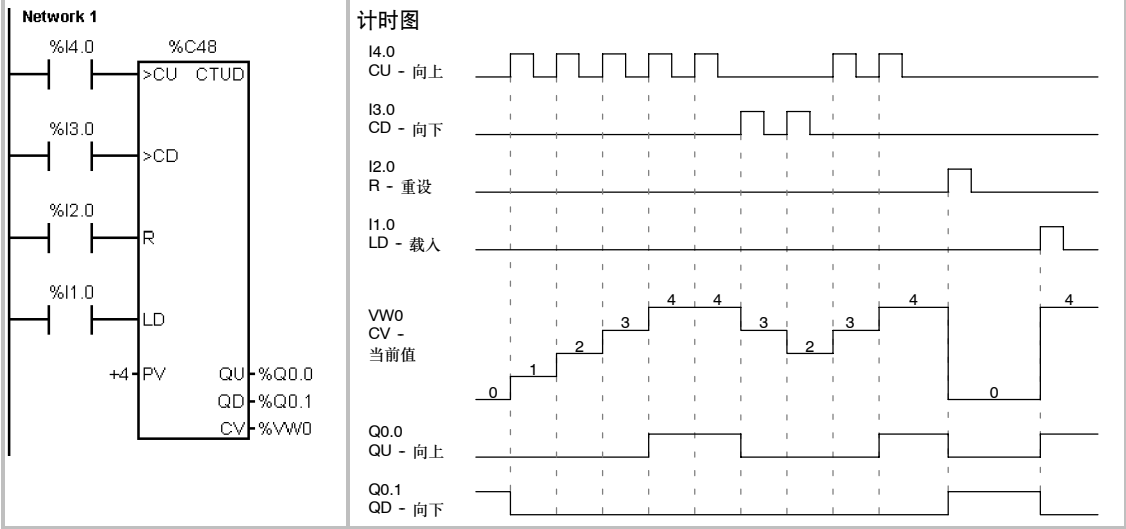
输入 / 输出	数据类型	操作数
Cxx	D	常量（C0到C255）
CU、CD、LD、R	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
PV	INT	IW、QW、VW、MW、SMW、SW、LW、AC、AIW、*VD、*LD、*AC、常量
Q、QU、QD	BOOL	I、Q、V、M、SM、S、L
CV	INT	IW、QW、VW、MW、SW、LW、AC、*VD、*LD、*AC



#### 提示

因为每个计数器有一个当前值，不要分配同样的数字给一个以上计数器。（“向上计数器”、“向下计数器”和“向上 / 向下计数器”存取同样的当前值。）

实例: IEC计数器指令



## 高速计数器指令

### 高速计数器指令定义

“高速计数器定义”指令（HDEF）选择指定高速计数器（HSCx）的操作模式。模式选择定义高速计数器的时钟、方向、开始和重设功能。

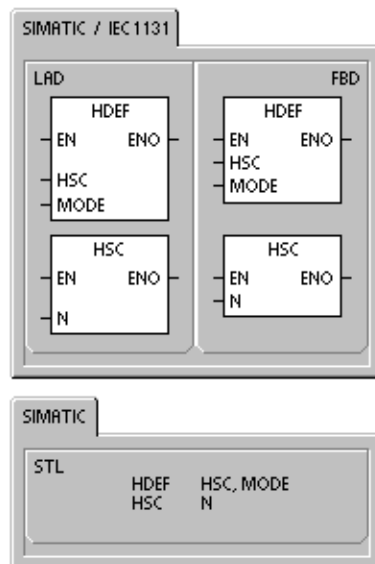
可以将一个“高速计数器定义”指令用于每个高速计数器。

设置ENO = 0的错误条件

- 0003（输入点冲突）
- 0004（中断中的非法的指令）
- 000A（HSC重新定义）

### 高速计数器指令

“高速计数器（HSC）”指令基于HSC特殊内存位的状态以配置和控制高速计数器。参数N指定高速计数器数目。



高速计数器可以配置为至多12种不同的操作模式。参见表6-25。

每个计数器有用于时钟、方向控制、重设和启动的专用输入，在那里支持这些功能。对于双相计数器，两个时钟都可以在最大计数率下运行。在求积模式下，可以选择一倍（1x）或四倍（4x）最大计数率。所有运行在最大计数率的计数器不会互相干扰。

设置ENO = 0的错误条件

- 0001（HSC在HDEF前）
- 0005（同时HSC/PLS）

表6-24 高速计数器指令的有效操作数

输入 / 输出	数据类型	操作数
HSC、MODE	BYTE	常量
N	WORD	常量



提示与技巧

关于使用高速计数器的程序，请参考文档光盘上的“程序提示”。参见提示4和提示29。

高速计数器计算S7-200扫描率无法控制的高速事件。高速计数器的最大计数频率取决于S7-200 CPU型号。关于更多的信息请参考附录A。



#### 提示

CPU 221和CPU 222支持四个高速计数器：HSC0、HSC3、HSC4和HSC5。这些CPU不支持HSC1和HSC2。

CPU 224、CPU 226和CPU226XM支持六个高速计数器：HSC0到HSC5。

一般地，高速计数器用作鼓式定时器的驱动器，在定时器中，以恒定速度旋转的轴安装有递增转轴编码器。轴编码器提供每转的指定计数数字，重设脉冲在每转产生一次。时钟和来自轴编码器的重设脉冲提供到高速计数器的输入。

高速计数器用第一个几次预置载入，期望的输出在当前计数小于当前预置的时间期间激活。安装计数器以在当前计数等于预置以及重设产生时提供中断。

当每个当前计数数值等于预置数值中断事件产生时，载入新的预置，设置下一个输出状态。当重设中断事件产生时，设置第一个预置和第一个输出状态，重复周期。

因为中断发生率远低于高速计数器的计数率，所以可以实现高速操作的精确控制，而对全部的PLC扫描循环影响相当小。附加中断的方法允许每次载入新预置在独立的中断例行程序中完成以便易于状态控制。（此外，所有中断事件也可以在单个中断例行程序中处理。）

### 了解不同的高速计数器

所有计数器功能对于同样的计数器操作模式是一样的。有四种基本计数器类型：单相计数器带有内部方向控制，单相计数器带有外部方向控制，双相计数器带有2个时钟输入和A/B相正交计数器。注意每种模式不能为每个计数器所支持。可以使用每种类型：没有重设或启动输入、有重设和没有启动、或启动和重设输入都有。

- 当激活重设输入时，它清除当前值并保持清除直到取消激活重设。
- 当激活启动输入时，就允许计数器计数。取消激活启动的同时，计数器的当前值保持为常量，时钟事件忽略。
- 如果重设激活而启动非现用，重设被忽略，而当前值不变。如果启动输入成为现用，而重设输入为现用，当前值被清除。

在使用高速计数器前，请使用HDEF指令（高速计数器定义）选择计数器模式。使用第一个扫描内存位SM0.1（此位在第一次扫描时接通，然后断开），来调用包含HDEF指令的子例行程序。

### 对高速计数器编程



指令向导

可以使用HSC指令向导配置计数器。向导使用下列信息：计数器的类型和模式、计数器预设值、计数器当前值和初始计数方向。要启动“HSC指令向导”，选择工具（Tools）> 指令向导（Instruction Wizard）菜单命令，然后从“指令向导”窗口选择HSC。

要对高速计数器编程，必须完成下列基本任务：

- 定义计数器和模式。
- 设置控制字节。
- 设置当前值（起始值）。
- 设置预设值（目标数值）。
- 分配和启用中断例行程序。
- 激活高速计数器。

## 定义计数器模式和输入

使用“高速计数器定义”指令定义计数器模式和输入。

表6-25显示用于与高速计数器有关的时钟、方向控制、重设和启动功能的输入。同样的输入不能用于两种不同的功能，但任何不被其高速计数器的显示模式使用的输入可以用于其它用途。例如，如果HSC0正被用于模式1，它使用I0.0和I0.2，I0.1可用于HSC3的边沿中断。



### 提示

注意HSC0的所有模式始终使用I0.0，而HSC4的所有模式始终使用I0.3，所以当这些计数器在使用中时，这些点永远不能用于其它用途。

表6-25 高速计数器的输入

模式	描述	输入				
		HSC0	I0.0	I0.1	I0.2	
		HSC1	I0.6	I0.7	I1.0	I1.1
		HSC2	I1.2	I1.3	I1.4	I1.5
		HSC3	I0.1			
		HSC4	I0.3	I0.4	I0.5	
		HSC5	I0.4			
0	具有内部方向控制的单相计数器	时钟				
1		时钟		重设		
2		时钟		重设	启动	
3	具有外部方向控制的单相计数器	时钟	方向			
4		时钟	方向	重设		
5		时钟	方向	重设	启动	
6	具有2个时钟输入的双相计数器	向上时钟	向下时钟			
7		向上时钟	向下时钟	重设		
8		向上时钟	向下时钟	重设	启动	
9	A/B相正交计数器	时钟A	时钟B			
10		时钟A	时钟B	重设		
11		时钟A	时钟B	重设	启动	

## HSC模式的实例

图6-22到图6-26的计时图显示每个计数器是如何根据模式工作的。

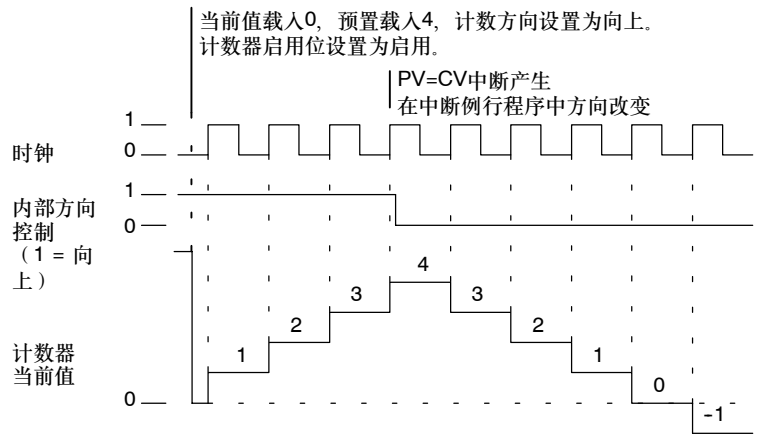


图6-22 模式0、1或2的操作实例

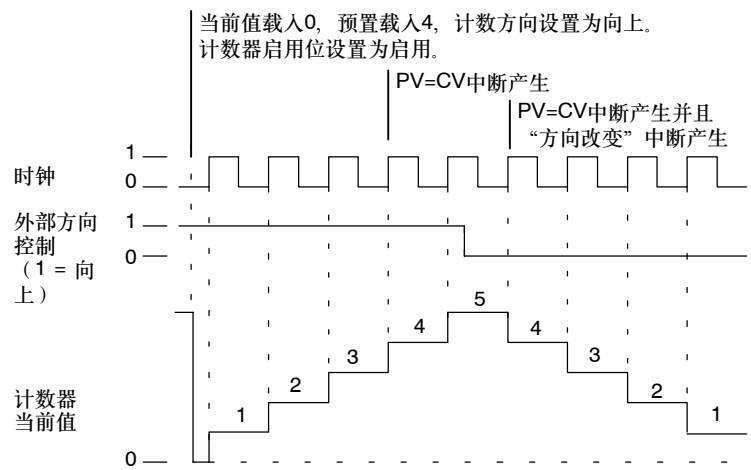


图6-23 模式3、4或5的操作实例

当使用计数模式6、7或8，并且向上时钟和向下时钟输入的上升沿互相都在0.3微秒中发生，高速计数器可能会将这些事件视作同时发生。如果这发生，当前值不变，指示计数方向没有改变。只要向上和向下时钟输入的上升沿之间的间隔大于此时间间隔，高速计数器分别捕获每个事件。在两种情况下，没有产生错误，计数器保持正确的计数数值。

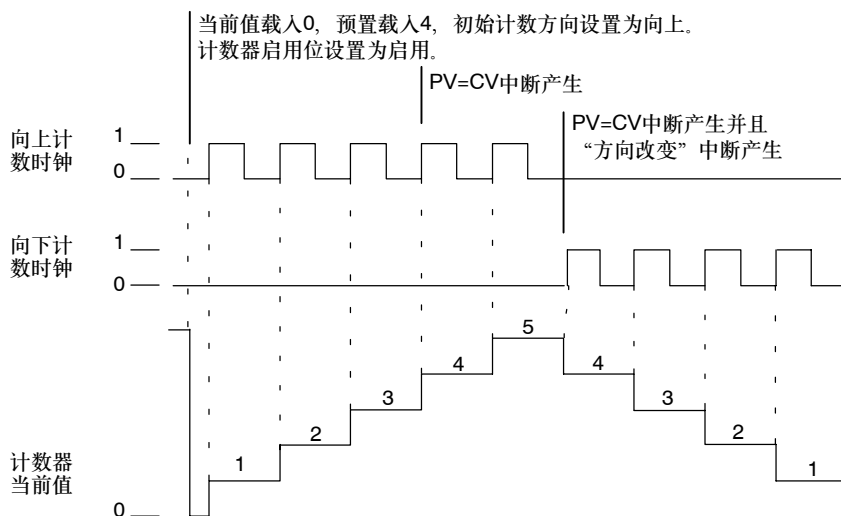


图6-24 模式6、7或8的操作实例

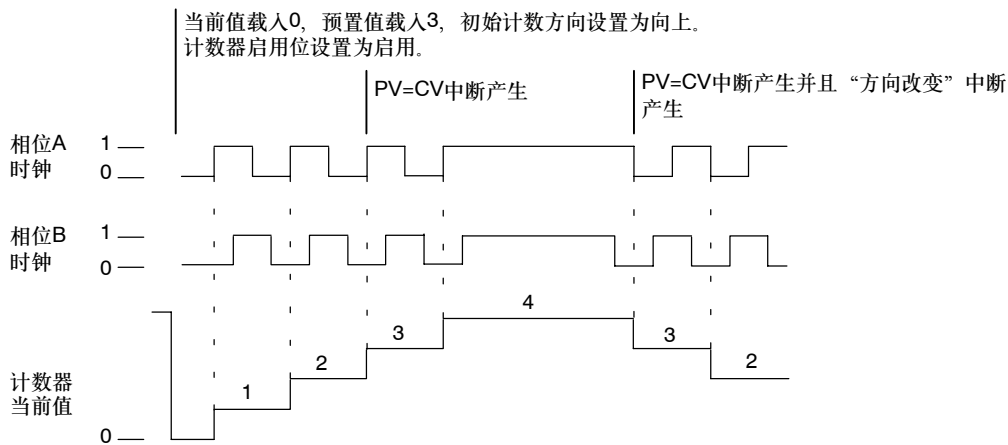


图6-25 模式9、10或11（正交1x模式）的操作实例



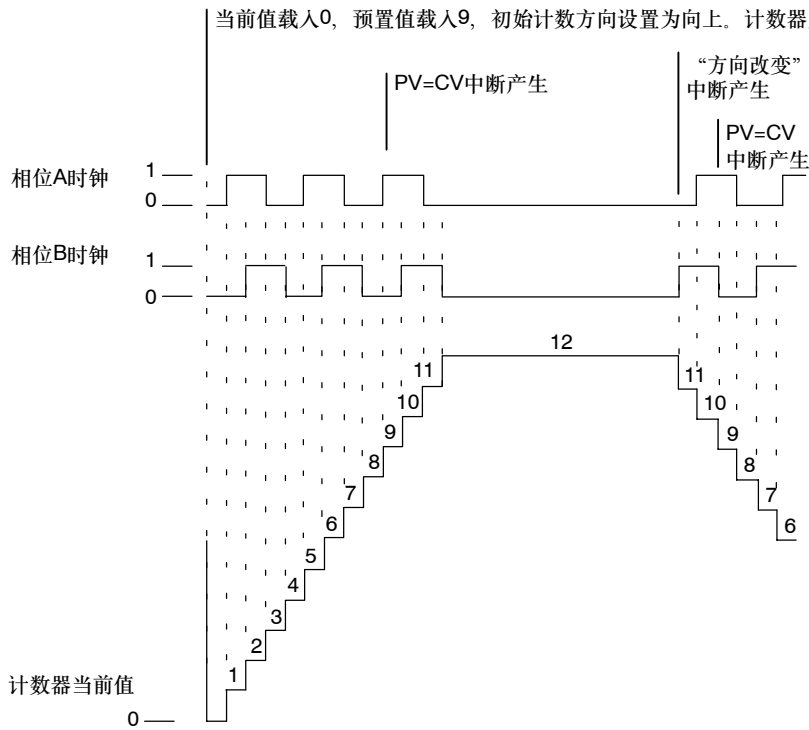


图6-26 模式9、10或11（求积4x模式）的操作实例

### 重设和启动操作

图6-27中显示的重设和启动输入的操作适用于所有使用重设和启动输入的模式。在重设和启动输入的图中，重设和启动都以激活状态设计成高位显示。

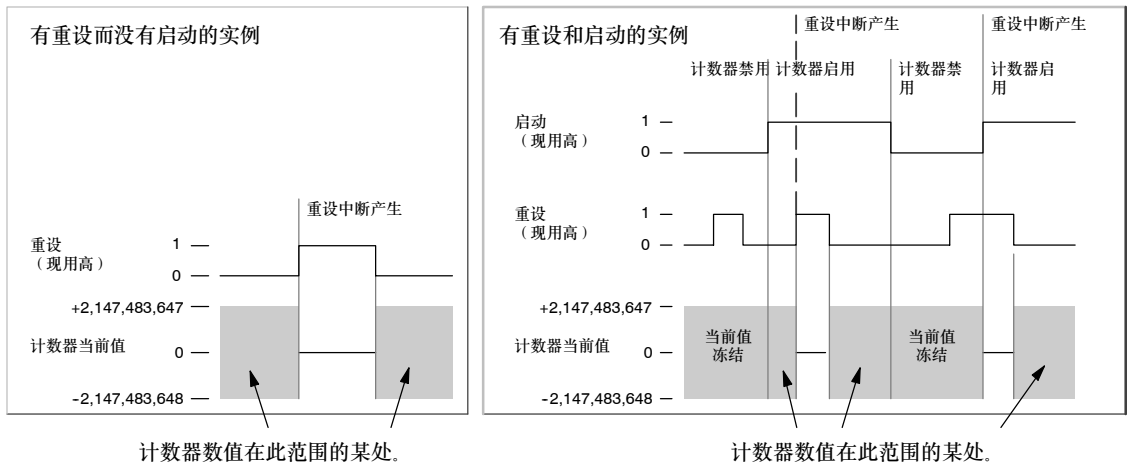


图6-27 使用带或不带启动进行重设的操作实例

四个计数器有三个控制位，控制位用于配置重设和启动输入的激活状态，以及用于选择1x或4x计数模式（仅用于求积计数器）。这些位位于各自计数器的控制位，只有当执行HDEF指令时才使用。这些位在表6-26中定义。



#### 提示

必须在执行HDEF指令前将这三个控制位设置为期望的状态。否则，计数器以默认配置作为所选的计数器模式。

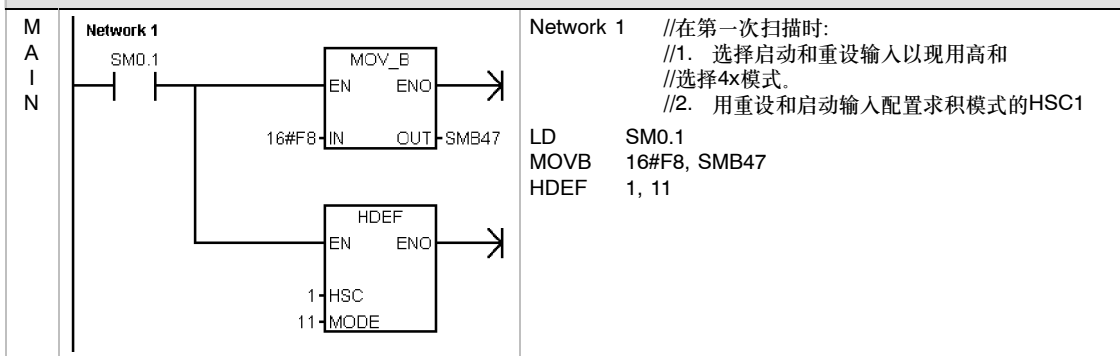
一旦执行了HDEF指令，就不能改变计数器设置，除非首先将S7-200置于STOP（停止）模式。

表6-26 重设、启动和1x/4x控制位的激活级别

HSC0	HSC1	HSC2	HSC4	描述（仅当执行HDEF时使用）
0	0	0	0	“重设”的激活级别控制位 <sup>1</sup> ： 0 = 重设为现用高      1 = 重设为现用低
---	1	1	---	“启动”的激活级别控制位 <sup>1</sup> ： 0 = 启动为现用高      1 = 启动为现用低
SM37.2	SM47.2	SM57.2	SM147.2	求积计数器的计数率选择： 0 = 4X 计数率      1 = 1X 计数率

1 重设输入和启动输入的缺省设置是现用高，而求积计数率是4x（或输入时钟频率的四倍）。

#### 实例：高速计数器指令定义指令



## 设置控制字节

在定义计数器和计数器模式后，可以设计计数器的动态参数。每个高速计数器有一个控制字节，允许下列操作：

- 启用或禁用计数器
- 控制方向（仅用于模式0、1和2），或所有其它模式的初始计数方向
- 载入当前值
- 载入预设值

控制字节的检查和相关的当前和预设值被HSC指令的执行调用。表6-27对这些控制位的每一个进行描述。

表6-27 HSC0、HSC1、HSC2、HSC3、HSC4和HSC5的控制位

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	描述
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	计数方向控制位： 0 = 向下计数      1 = 向上计数
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	将计数方向写入HSC： 0 = 无更新      1 = 更新方向
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	将新预设值写入HSC： 0 = 无更新      1 = 更新预置
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	将新当前值写入HSC： 0 = 无更新      1 = 更新当前值
SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	启用HSC： 0 = 禁用HSC      1 = 启用HSC

## 设置当前值和预设值

每个高速计数器有一个32位当前值和一个32位预设值。当前值和预设值都是有符号整数。要将新当前值或预设值载入高速计数器，必须设置保存当前和/或预设值的控制字节和特殊内存字节，也要执行HSC指令以使新数值传送到高速计数器。表6-28列出用于保存新当前值和预设值的特殊内存字节。

除控制字节和新预置和当前保存字节以外，每个高速计数器的当前值只能用数据类型HC（高速计数器趋势）后跟计数器的编号（0、1、2、3、4或5）来读取，如表6-28所示。当前值对于读操作是直接可存取的，但是只能用HSC指令写。

表6-28 HSC0、HSC1、HSC2、HSC3、HSC4和HSC5的新当前值和新预设值

要载入的数值	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
新当前值	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
新预置	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

表6-29 HSC0、HSC1、HSC2、HSC3、HSC4和HSC5的当前值

数值	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
当前	HC0	HC1	HC2	HC3	HC4	HC5

## 对高速计数器（HC）编址

要对高速计数器存取计数值，需要使用内存型号（HC）和计数器号码（诸如HC0）指定高速计数器的地址。高速计数器的当前值是一个只读数值，仅能以双字（32位）编址，如图6-28所示。

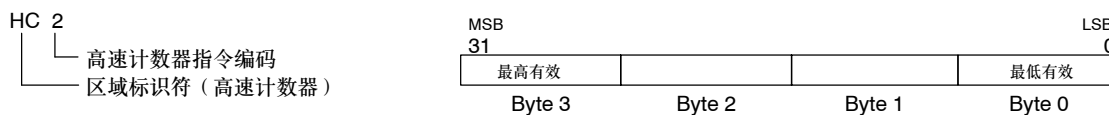


图6-28 存取高速计数器当前值

## 分配中断

所有计数器模式支持当前值等于预设值时的中断。使用外部重设输入的计数器模式支持激活外部重设时的中断。所有计数器模式除了模式0、1和2都支持计数方向改变时的中断。这些中断条件的每一个都可以分别启用或禁用。关于使用中断完全讨论，参见章节“通讯和中断”指令。

### 注意

如果从外部重设中断例行程序中尝试载入新当前值或禁用然后再启用高速计数器，可能会发生致命错误。

## 状态字节

每个高速计数器的状态字节提供状态内存位，它指示当前计数方向以及当前值是否大于或等于预设值。表6-30为每个高速计数器定义这些状态位。



### 提示

只有当高速计数器中断例行程序被执行时，状态位才有效。监控高速计数器状态的目的是启用对被执行的操作很重要事件的中断。

表6-30 HSC0、HSC1、HSC2、HSC3、HSC4和HSC5的状态位

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	描述
0	0	0	0	0	0	未用
1	1	1	1	1	1	未用
SM36.2	SM46.2	SM56.2	SM136.2	SM146.2	SM156.2	未用
SM36.3	SM46.3	SM56.3	SM136.3	SM146.3	SM156.3	未用
SM36.4	SM46.4	SM56.4	SM136.4	SM146.4	SM156.4	未用
SM36.5	SM46.5	SM56.5	SM136.5	SM146.5	SM156.5	当前计数方向状态位: 0 = 向下计数 1 = 向上计数
SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	当前值等于预设值状态位: 0 = 不等于 1 = 等于
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	当前值大于预设值状态位: 0 = 小于或等于 1 = 大于

## 高速计数器的初始化顺序范例

在下列初始化和操作顺序描述中，HSC1用作模型计数器。初始化描述假定S7-200刚放在RUN（运行）模式，因此，第一次扫描内存位为真。如果情况不是这样，记住在进入RUN（运行）模式后，对于每个高速计数器HDEF指令只能执行一次。对高速计数器第二次执行HDEF产生运行时错误，不会以第一次执行HDEF时为此计数器设置的方式改变计数器设置。



### 提示

尽管下列顺序显示如何单个地改变方向、当前值和预设值，但是可以通过适当地设置SMB47数值，然后执行HSC指令，以同样的顺序改变它们所有或任何的组合。

### 初始化模式0、1或2

下列步骤描述如何为具有内部方向（模式0、1或2）的单相向上/向下计数器初始化HSC1。

1. 使用第一次扫描内存位调用在其中执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序，这减少扫描执行时间和提供更多的结构程序。
2. 在初始化子例行程序中，根据期望的控制操作载入SMB47。例如：
 

SMB47 = 16#F8	产生下列结果：
	启用计数器
	写新当前值
	写新预设值
	设置方向为向上计数
	设置启动和重设输入到现用高
3. 执行HDEF指令，HSC输入设置为1，“模式”输入设置为下列之一：对于无外部重设或启动为0，对于外部重设和无启动为1，对于外部重设和启动为2。
4. 用期望的当前值（用0载入以清除它）载入SMD48（双字大小数值）。
5. 用期望的预设值载入SMD52（双字大小数值）。
6. 为了捕获当前值等于预置事件，通过将CV= PV中断事件（事件13）附加到中断例行程序编程中断。参见讨论中断过程完全细节的“中断指令”部分。
7. 为了捕获外部重设事件，通过将外部重设中断事件（事件15）附加到中断例行程序编程中断。
8. 执行全局中断启用指令（ENI）来启用中断。
9. 执行HSC指令来引起S7-200编程HSC1。
10. 退出子例行程序。

**初始化模式3、4或5**

下列步骤描述如何为具有外部方向（模式3、4或5）的单相向上 / 向下计数器初始化HSC1。

1. 使用第一次扫描内存位调用在其中执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序，这减少扫描执行时间和提供更多的结构程序。
2. 在初始化子例行程序中，根据期望的控制操作载入SMB47。例如：  
 SMB47 = 16#F8      *产生下列结果：*  
                          启用计数器  
                          写新当前值  
                          写新预设值  
                          设置HSC的初始方向为向上计数  
                          设置开始和重设输入为现用高
3. 执行HDEF指令，HSC输入设置为1，“模式”输入设置为下列之一：对于无外部重设或启动为3，对于外部重设和无启动为4，对于外部重设和启动为5。
4. 用期望的当前值（用0载入以清除它）载入SMD48（双字大小数值）。
5. 用期望的预设值载入SMD52（双字大小数值）。
6. 为了捕获当前值等于预置事件，通过将CV= PV中断事件（事件13）附加到中断例行程序编程中断。参见讨论中断过程完全细节的“中断指令”部分。
7. 为了捕获外部方向改变，通过将方向改变中断事件（事件14）附加到中断例行程序编程中断。
8. 为了捕获外部重设事件，通过将外部重设中断事件（事件15）附加到中断例行程序编程中断。
9. 执行全局中断启用指令（ENI）来启用中断。
10. 执行HSC指令以引发S7-200对HSC1编程。
11. 退出子例行程序。

**初始化模式6、7或8**

下列步骤描述如何为具有向上 / 向下时钟（模式6、7或8）的双相向上 / 向下计数器初始化HSC1：

1. 使用第一次扫描内存位调用在其中执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序，这减少扫描执行时间和提供更多的结构程序。
2. 在初始化子例行程序中，根据期望的控制操作载入SMB47。例如：  
 SMB47 = 16#F8      *产生下列结果：*  
                          启用计数器  
                          写新当前值  
                          写新预设值  
                          设置HSC的初始方向为向上计数  
                          设置开始和重设输入为现用高
3. 执行HDEF指令，HSC输入设置为1，“模式”设置为下列之一：对于无外部重设或启动为6，对于外部重设和无启动为7，对于外部重设和启动为8。
4. 用期望的当前值（用0载入以清除它）载入SMD48（双字大小数值）。
5. 用期望的预设值载入SMD52（双字大小数值）。
6. 为了捕获当前值等于预置事件，通过将CV= PV中断事件（事件13）附加到中断例行程序编程中断。参见关于中断的部分。

7. 为了捕获外部方向改变, 通过将方向改变中断事件(事件14)附加到中断例行程序编程中断。
8. 为了捕获外部重设事件, 通过将外部重设中断事件(事件15)附加到中断例行程序编程中断。
9. 执行全局中断启用指令(ENI)来启用中断。
10. 执行HSC指令以引发S7-200对HSC1编程。
11. 退出子例行程序。

### 初始化模式9、10或11

下列步骤描述如何为A/B相正交计数器(模式9、10或11)初始化HSC1。

1. 使用第一次扫描内存位调用在其中执行初始化操作的子例行程序。因为使用子例行程序调用, 随后的扫描不再调用子例行程序, 这减少扫描执行时间并提供更多的结构化程序。
2. 在初始化子例行程序中, 根据期望的控制操作载入SMB47。

实例(1x计数模式):

SMB47 = 16#FC

产生下列结果:

启用计数器  
写新当前值  
写新预设值  
设置HSC的初始方向为向上计数  
设置开始和重设输入为现用高

实例(4x计数模式):

SMB47 = 16#F8

产生下列结果:

启用计数器  
写新当前值  
写新预设值  
设置HSC的初始方向为向上计数  
设置开始和重设输入为现用高

3. 执行HDEF指令, HSC输入设置为1, “模式”输入设置为下列之一: 对于无外部重设或启动为9, 对于外部重设和无启动为10, 对于外部重设和启动为11。
4. 用期望的当前值(用0载入以清除它)载入SMD48(双字大小数值)。
5. 用期望的预设值载入SMD52(双字大小数值)。
6. 为了捕获当前值等于预置事件, 通过将CV= PV中断事件(事件13)附加到中断例行程序编程中断。关于中断处理的完全细节参见启用中断(ENI)部分。
7. 为了捕获方向改变, 通过将方向改变中断事件(事件14)附加到中断例行程序编程中断。
8. 为了捕获外部重设事件, 通过将外部重设中断事件(事件15)附加到中断例行程序编程中断。
9. 执行全局中断启用指令(ENI)来启用中断。
10. 执行HSC指令以引发S7-200对HSC1编程。
11. 退出子例行程序。

**在模式0、1或2中改变方向**

下列步骤描述如何为具有内部方向（模式0、1或2）单相计数器的改变方向配置HSC1:

1. 载入SMB47以写期望的方向:
 

SMB47 = 16#90	启用计数器 设置HSC的方向为向下计数
SMB47 = 16#98	启用计数器 设置HSC的方向为向上计数
2. 执行HSC指令以引发S7-200对HSC1编程。

**载入新当前值（任何模式）**

更换当前值强制当改变进行时计数器禁用。当计数器禁用时，它不计数或产生中断。

下列步骤描述如何改变HSC1的计数器当前值（任何模式）:

1. 载入SMB47以写期望的当前值:
 

SMB47 = 16#C0	启用计数器 写新当前值
---------------	----------------
2. 用期望的当前值（用0载入以清除它）载入SMD48（双字大小数值）。
3. 执行HSC指令以引发S7-200对HSC1编程。

**载入新预设值（任何模式）**

下列步骤描述如何改变HSC1的预设值（任何模式）:

1. 载入SMB47以写期望的预设值:
 

SMB47 = 16#A0	启用计数器 写新预设值
---------------	----------------
2. 用期望的预设值载入SMD52（双字大小数值）。
3. 执行HSC指令以引发S7-200对HSC1编程。

**禁用高速计数器（任何模式）**

下列步骤描述如何禁用HSC1高速计数器（任何模式）:

1. 将SMB47载入到禁用计数器:
 

SMB47 = 16#00	禁用计数器
---------------	-------
2. 执行HSC指令以禁用计数器。



实例: 高速计数器指令		
M A I N	<p>Network 1</p>	<p>Network 1 //在第一次扫描时, 调用SBR_0.</p> <pre>LD SM0.1 CALL SBR_0</pre>
S B R 0	<p>Network 1</p>	<p>Network 1 //在第一次扫描时, 配置HSC1:</p> <ul style="list-style-type: none"> <li>//1. 启用计数器。</li> <li>// - 写新当前值。</li> <li>// - 写新预设值。</li> <li>// - 设置初始方向为向上计数。</li> <li>// - 选择启动和重设输入为现用高。</li> <li>// - 选择4x模式。</li> <li>//2. 用重设和启动输入配置求积模式的HSC1</li> <li>//3. 清除HSC1的当前值。</li> <li>//4. 将HSC1预设值设置为50。</li> <li>//5. 当HSC1当前值 = 预设值,</li> <li>// 将事件13附加到中断例行程序INT_0。</li> <li>//6. 全局中断启用。</li> <li>//7. 编程HSC1。</li> </ul> <pre>LD SM0.1 MOVB 16#F8, SMB47 HDEF 1, 11 MOVD +0, SMD48 MOVD +50, SMD52 ATCH INT_0, 13 ENI HSC 1</pre>
整 数 0	<p>Network 1</p>	<p>Network 1 //编程HSC1:</p> <ul style="list-style-type: none"> <li>//1. 清除HSC1的当前值。</li> <li>//2. 选择只写新当前和让HSC1启用。</li> </ul> <pre>LD SM0.0 MOVD +0, SMD48 MOVB 16#C0, SMB47 HSC 1</pre>

## 脉冲输出指令



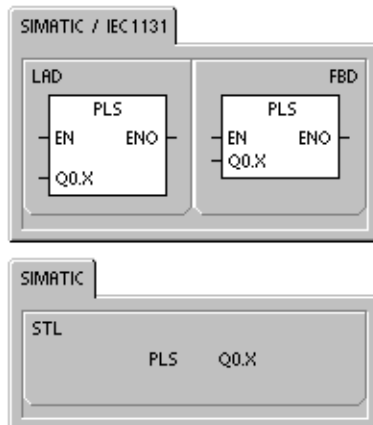
定位控制

“脉冲输出”指令（PLS）用于控制在高速输出（Q0.0和Q0.1）可用的“脉冲链输出”（PTO）和“脉冲宽度调制”（PWM）功能。可以使用“位置控制”向导配置脉冲输出。

PTO提供方波（50%工作循环）输出，用户控制周期时间和脉冲的数目。

PWM提供持续、可变的工作循环输出，用户控制周期时间和脉冲宽度。

S7-200有两个PTO/PWM生成器，它创建高速度脉冲链或脉冲宽度调制波形。一个生成器分配给数字输出点Q0.0，另一个生成器分配给数字输出点Q0.1。指定特殊内存（SM）位置为每个生成器存储下列数据：控制字节（8位数值），脉冲计数值（无符号32位数）以及周期时间和脉冲宽度数值（无符号16位数）。



## 6

PTO/PWM生成器和进程图像寄存器共享使用Q0.0和Q0.1。当PTO或PWM功能在Q0.0或Q0.1激活，PTO/PWM生成器控制输出，正常使用输出点禁止。输出波形不受进程图像寄存器的状态、点的强制数值或执行立即输出指令影响。当PTO/PWM生成器非现用，输出的控制回复到进程图像寄存器。进程图像寄存器确定输出波形的初始和最终状态，引起波形在高位或低位启动和结束。

表6-31 脉冲输出指令的有效操作数

输入 / 输出	数据类型	操作数
Q0.X	WORD	常量: 0 (= Q0.0) 或 1 (= Q0.1)



### 提示

在启用PTO或PWM操作之前，为Q0.0和Q0.1到0设置进程图像寄存器的数值。

所有控制位、周期时间、脉冲宽度和时钟脉冲计数数值的默认值是0。

**PTO/PWM输出必须具有至少额定负载的10%最小载入，以提供从断开到接通以及从接通到断开的明确跳转。**



提示与技巧

关于使用PLS指令为PTO/PWM操作的程序，请参考文档光盘上的编程提示。参见提示7、提示22、提示23、提示30和提示50。

## 脉冲链操作 (PTO)

对于指定数目的时钟脉冲和指定周期时间，PTO提供方波（50%工作循环）。（参见图6-29。）PTO可以产生单脉冲链或多脉冲链（使用脉冲概要图）。指定脉冲数和周期时间（以微秒或毫秒为增量）：

- 脉冲数: 1至4,294,967,295
- 周期时间: 50微秒到65,535微秒或  
2毫秒到65,535毫秒。

指定奇数数目的微秒或毫秒作为周期时间（例如75毫秒）会引起工作循环中的一些失真。

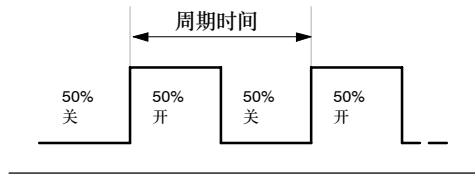


图6-29 脉冲链输出 (PTO)

参见表6-32用于脉冲计数和周期时间限制。

表6-32 在PTO功能中的脉冲计数和周期时间

脉冲计数 / 周期时间	反应
周期时间 < 2个时间单元	周期时间默认为2个时间单元。
脉冲计数 = 0	脉冲计数默认为1个脉冲。

PTO功能允许脉冲链的“链接”或“流水线操作”。当激活的脉冲链完成，新脉冲链的输出立即开始。这允许随后的输出脉冲链之间的连续性。

### PTO脉冲的单段流水线操作

在单段流水线操作中，用户负责为下一个时钟脉冲链更新SM位置。在初始PTO程序段启动后，必须如第二波形所需立即修改SM位置，并再次执行PLS指令。第二脉冲链的属性保持在流水线中，直到第一脉冲链完成。一次只有一个条目可以存储在流水线中。当第一脉冲链完成时，第二波形的输出开始，流水线对于新脉冲链规定可用。然后可以重复此过程以设置下一个脉冲链的特征。

脉冲链之间的平滑跳转产生，除非在时基中有改变或者在新脉冲链装置被PLS指令的执行捕获前激活的脉冲链完成。

### PTO脉冲的多段流水线操作

在多段流水线操作中，S7-200从位于V内存的概要表自动读每个脉冲链段的特征。用在此模式中的SM位置是控制字节、状态字节和概要表的启动V内存偏移量（SMW168或SMW178）。时基既可以是微秒也可以是毫秒，但所作的选择将应用于概要表中的所有周期时间值，并且当概要图正在运行时无法改变。执行PLS指令启动多段操作。

每个段条目有8个字节长，由16位周期时间值、16位周期时间德耳塔数值和32位脉冲计数数值组成。表6-33显示概要表的格式。可以通过为每个脉冲编程指定的数目自动增加或减少周期时间。在周期时间德耳塔域中的正数值增加周期时间，在周期时间德耳塔域中的负数值减少周期时间，0导致周期时间不变。

当操作PTO概要图时，当前激活段数在SMB166（或SMB176）中可用。

表6-33 多段PTO操作的概要表格式

字节偏移量	分段	表格条目的描述
0		段数: 1至255 <sup>1</sup>
1	#1	初始周期时间 (时基的2到65,535个单元)
3		每个脉冲的周期时间德耳塔 (有符号数值) (时基的-32,768到32,767个单元)
5		脉冲计数 (1到4,294,967,295)
9	2#	初始周期时间 (时基的2到65,535个单元)
11		每个脉冲的周期时间德耳塔 (有符号数值) (时基的-32,768到32,767个单元)
13		脉冲计数 (1到4,294,967,295)
(继续)	# 3	(继续)

1 为段数输入数值0产生非严重错误。没有PTO输出产生。

## 6

## 脉冲宽度调制 (PWM)

PWM提供固定的周期时间输出, 带有可变的工作循环。(参见图6-30。)可以指定周期时间和脉冲宽度以微秒或毫秒增加:

- 周期时间: 50微秒到65,535微秒或  
2毫秒到65,535毫秒
- 脉冲宽度时间: 0微秒到65,535微秒或  
0毫秒到65,535毫秒

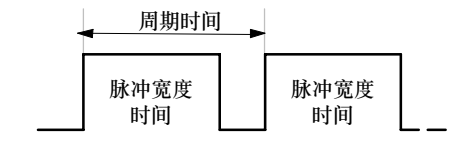


图6-30 脉冲宽度调制 (PWM)

如表6-34所示, 设置脉冲宽度等于周期时间 (使工作循环为100%) 将输出连续接通。设置脉冲宽度为0 (使工作循环为0%) 将输出断开。

表6-34 PWM功能中的脉冲宽度时间和周期时间以及反应

脉冲宽度时间 / 周期时间	反应
脉冲宽度时间 >= 周期时间数值	工作循环为100%: 输出连续接通。
脉冲宽度时间 = 0	工作循环为0%: 输出断开。
周期时间 < 2个时间单元	周期时间默认为两个时间单元。

有两种不同的方式改变PWM波形的特征:

- 同步更新: 如果不需要时基改变, 可以使用同步更新。使用同步更新, 波形特征中的改变以周期为界线发生, 假设是平滑转变。
- 异步更新: 对于PWM操作, 一般地是, 脉冲宽度在周期时间保持常量时变化, 所以不需要时基改变。然而, 如果PTO/PWM生成器的时基改变是所需要的, 则使用异步更新。异步更新引起PTO/PWM生成器即刻禁用, 与PWM波形异步。这会引引起控制设备中的发生意外的不稳定。因此, 推荐使用同步PWM更新。选择希望为所有预期的周期时间数值工作的时基。

**提示**

在控制字节中的“PWM更新方法”位（SM67.4或SM77.4）指定在执行PLS指令以调用改变时使用的更新类型。

如果时基改变，不管“PWM更新方法”位的状态是什么，异步更新产生。

**使用SM位置配置和控制PTO/PWM操作**

PLS指令读存储在指定SM内存位置的数据，并相应地编程PTO/PWM生成器。SMB67控制PTO 0或PWM 0，SMB77控制PTO 1或PWM 1。表6-35描述用于控制PTO/PWM操作的寄存器。可以使用表6-36作为快速参考确定放置在PTO/PWM控制寄存器中的数值以调用期望的操作。

可以通过修改SM区域中的位置（包含控制字节）改变PTO或PWM波形的特征，然后执行PLS指令。通过将0写入控制字节（SM67.7或SM77.7）的PTO/PWM启用位，可以在任何时间禁用PTO或PWM波形的产生，然后执行PLS指令。

在状态字节（SM66.7或SM76.7）中的PTO空闲位用来指示编程脉冲链的完成。此外，中断例行程序可以在完成脉冲链时调用。（参考“中断”指令和“通讯”指令的描述。）如果正在使用多段操作，则在完成概要表时将调用中断例行程序。

下列条件设置SM66.4（或SM76.4）和SM66.5（或SM76.5）：

- 在许多脉冲后，指定导致非法周期时间的周期时间德耳塔数值将产生运算溢出条件，该条件终止PTO功能并设置“德耳塔计算错误”位（SM66.4或SM76.4）为1。输出回复图象寄存器控制。
- 手动中止（禁用）处理中的PTO概要图将设置“用户中止”位（SM66.5或SM76.5）为1。
- 当流水线满了时尝试将其载入可以设置PTO溢出位（SM66.6或SM76.6）为1。如果要检测随后的溢出，在溢出被检测后必须手动清除此位。转变为RUN（运行）模式初始化此位为0。

**提示**

当载入新脉冲计数（SMD72或SMD82）、脉冲宽度（SMW70或SMW80）或周期时间（SMW68或SMW78）时，在执行PLS指令前，也设置在控制寄存器中合适的更新位。对于多段脉冲链操作，在执行PLS指令之前，还必须载入概要表的起始偏移量（SMW168或SMW178）和概要表数值。

表6-35 PTO/PWM控制寄存器的SM位置

Q0.0	Q0.1	状态位		
SM66.4	SM76.4	PTO概要图被中止 (delta计算出错):	0 = 没有出错	1 = 中止
SM66.5	SM76.5	PTO概要图由于用户命令而中止:	0 = 没有中止	1 = 中止
SM66.6	SM76.6	PTO流水线溢出 / 下溢:	0 = 无溢出	1 = 溢出 / 下溢
SM66.7	SM76.7	PTO闲置:	0 = 在过程中	1 = PTO闲置
Q0.0	Q0.1	控制位:		
SM67.0	SM77.0	PTO/PWM更新周期时间:	0 = 无更新	1 = 更新周期时间
SM67.1	SM77.1	PWM更新脉冲宽度时间:	0 = 无更新	1 = 更新脉冲宽度
SM67.2	SM77.2	PTO更新脉冲计数数值:	0 = 无更新	1 = 更新脉冲计数
SM67.3	SM77.3	PTO/PWM时基:	0 = 1微秒 / 刻度	1 = 1毫秒 / 刻度
SM67.4	SM77.4	PWM更新方法:	0 = 异步	1 = 同步
SM67.5	SM77.5	PTO单 / 多段操作:	0 = 单	1 = 多
SM67.6	SM77.6	PTO/PWM模式选择:	0 = PTO	1 = PWM
SM67.7	SM77.7	PTO/PWM启用:	0 = 禁用	1 = 启用
Q0.0	Q0.1	其它PTO/PWM寄存器		
SMW68	SMW78	PTO/PWM周期时间数值范围: 2至65,535		
SMW70	SMW80	PWM脉冲宽度数值范围: 0 ~ 65,535		
SMD72	SMD82	PTO脉冲计数数值范围: 1至4,294,967,295		
SMB166	SMB176	在处理中的段数	仅用于多段PTO操作	
SMW168	SMW178	概要表的起始位置 (与V0的字节偏移量)	仅用于多段PTO操作	

表6-36 PTO/PWM控制字节参考

控制寄存器 (十六进制数值)	执行PLS指令的结果							
	允许	选择模式	PTO段操作	PWM更新方法	时基	脉冲计数	脉冲宽度	周期时间
16#81	是	PTO	单精度型		1微秒 / 周期			装载
16#84	是	PTO	单精度型		1微秒 / 周期	装载		
16#85	是	PTO	单精度型		1微秒 / 周期	装载		装载
16#89	是	PTO	单精度型		1毫秒 / 周期			装载
16#8C	是	PTO	单精度型		1毫秒 / 周期	装载		
16#8D	是	PTO	单精度型		1毫秒 / 周期	装载		装载
16#A0	是	PTO	多种		1微秒 / 周期			
16#A8	是	PTO	多种		1毫秒 / 周期			
16#D1	是	PWM		同步	1微秒 / 周期			装载
16#D2	是	PWM		同步	1微秒 / 周期		装载	
16#D3	是	PWM		同步	1微秒 / 周期		装载	装载
16#D9	是	PWM		同步	1毫秒 / 周期			装载
16#DA	是	PWM		同步	1毫秒 / 周期		装载	
16#DB	是	PWM		同步	1毫秒 / 周期		装载	装载

## 计算概要表数值

PTO/PWM发生器的多段流水线操作能力在许多应用中是有用的，尤其在步进电机控制中。

例如，通过定义由至多255个段组成的脉冲概要图，可以使用带有脉冲概要图的PTO控制步进电机从简单的斜坡向上、运行和斜坡向下顺序或更复杂的顺序，每个程序段相当于斜坡向上、运行或斜坡向下操作。

图6-31说明需要生成加速步进电机（段1）、以恒定速度操作电机（段2）和减速电机（段3）的输出波形范例概要表数值。

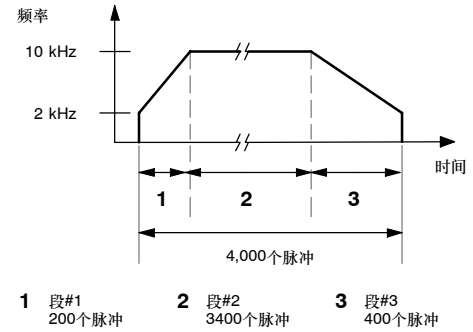


图6-31 频率/时间图

对于此例：启动和最终脉冲频率是2 kHz，最大脉冲频率是10 kHz，4000个脉冲需要达到期望的电机旋转数。因为概要表的数值是按照时间段（周期时间）表达的，而不是频率，所以必须将给定的频率数值转换为周期时间数值。因此，启动（初始）和最终（结束）周期时间是500微秒，相应于最大频率的周期时间是100微秒。在输出概要图的加速部分，最大脉冲频率应达到大约200个脉冲。概要图的减速部分应在大约400个脉冲中完成。

可以使用下列公式确定用于给定段的德耳塔周期时间数值，PTO/PWM生成器使用给定的段调整每个脉冲的周期时间：

$$\text{段的德耳塔周期时间} = | \text{End\_CT}_{\text{seg}} - \text{Init\_CT}_{\text{seg}} | / \text{Quantity}_{\text{seg}}$$

在此：  
 $\text{End\_CT}_{\text{seg}}$  = 此段的结束周期时间  
 $\text{Init\_CT}_{\text{seg}}$  = 此段的初始周期时间  
 $\text{Quantity}_{\text{seg}}$  = 此段中的脉冲数量

使用此公式计算应用实例的德耳塔周期时间数值：

段1（加速）：  
 德耳塔周期时间 = -2

段2（恒速）：  
 德耳塔周期时间 = 0

段3（减速）：  
 德耳塔周期时间 = 1

表6-37列出生成实例波形的数值（假设概要表位于V内存，开始于V500）。可以将指令包括在程序中以载入这些数值进入V内存，或者在数据块中定义概要表的数值。

表6-37 概要表数值

地址	数值	描述	
VB500	3	总段数:	
VW501	500	初始周期时间	段1
VW503	-2	初始德耳塔周期时间	
VD505	200	脉冲数	
VW509	100	初始周期时间	段2
VW511	0	德耳塔周期时间	
VD513	3400	脉冲数	
VW517	100	初始周期时间	段3
VW519	1	德耳塔周期时间	
VD521	400	脉冲数	

为了确定波形段之间的跳转是否可接受，需要确定段中最后一个脉冲的周期时间。除非delta周期时间是0，否则必须计算段的最后一个脉冲的周期时间，因为该数值没有在概要图中指定。使用下列公式计算最后一个脉冲的周期时间：

$$\text{段最后一个脉冲的周期时间} = \text{Init\_CT}_{\text{seg}} + (\text{Delta}_{\text{seg}} * (\text{Quantity}_{\text{seg}} - 1))$$

在此：  $\text{Init\_CT}_{\text{seg}}$  = 此段的初始周期时间

$\text{Delta}_{\text{seg}}$  = 此段的德耳塔周期时间

$\text{Quantity}_{\text{seg}}$  = 此段中的脉冲数量

上面简化的实例作为一个简介是很有用的，而真正的应用当中可能需要更复杂的波形概要图。记住德耳塔周期时间只能指定为微秒或毫秒的整数数字，周期时间的修改在每个脉冲上进行。

这两个条目的作用是计算给定段的德耳塔周期时间数值可能需要迭代法。在结束周期时间的数值或给定段的脉冲数中，可能需要一些适应性。

所提供的概要图段的持续时间可能在确定正确概要表数值的过程中是很有用的。使用下列公式来计算完成所提供概要图段的时间长度：

$$\text{段持续时间} = \text{Quantity}_{\text{seg}} * (\text{Init\_CT} + ((\text{Delta}_{\text{seg}}/2) * (\text{Quantity}_{\text{seg}} - 1)))$$

在此：  $\text{Quantity}_{\text{seg}}$  = 此段中的脉冲数量

$\text{Init\_CT}_{\text{seg}}$  = 此段的初始周期时间

$\text{Delta}_{\text{seg}}$  = 此段的德耳塔周期时间



## PWM输出的范例操作



### 提示

下列PWM初始化和操作顺序的描述推荐使用“首次扫描”位（SM0.1）以初始化脉冲输出。使用“首次扫描”位调用初始化子例行程序减少扫描时间，因为随后的扫描不调用此子例行程序。（“首次扫描”位只在转变为RUN（运行）模式后在首次扫描时设置。）然而，应用程序可能有其它限制需要初始化（或重新初始化）脉冲输出。在那种情况下，可以使用另一个条件调用初始化例行程序。

### 初始化PWM输出

一般地，使用子例行程序为脉冲输出初始化PWM。从主程序调用初始化子例行程序。使用首次扫描内存位（SM0.1）初始化被PWM使用的输出为0，并调用子例行程序完成初始化操作。当使用子例行程序调用时，后来的扫描不调用子例行程序，这减少扫描执行时间并提供更结构化的程序。

在从主程序创建对初始化子例行程序的调用后，使用下列步骤在初始化子例行程序中创建对配置脉冲输出Q0.0的控制逻辑：

1. 通过载入下列数值之一到SMB67配置控制字节：16#D3（选择微秒增量）或16#DB（选择毫秒增量）。  
这些数值都启用PTO/PWM功能，选择PWM操作，设置更新脉冲宽度和周期时间数值，选择时基（微秒或毫秒）。
2. 为周期时间将字大小数值载入SMW68。
3. 为脉冲宽度将字大小数值载入SMW70。
4. 执行PLS指令（以致S7-200对PTO/PWM生成器编程）。
5. 要为后来的脉冲宽度改变（供选用）预载新控制字节数值，将下列数值之一载入SMB67：16#D2（微秒）或16#DA（毫秒）。
6. 退出子例行程序。

### 更换PWM输出的脉冲宽度

如果用16#D2或16#DA预载SMB67（参见以上的步骤5.），可以使用改变脉冲输出（Q0.0）的脉冲宽度的子例行程序。在创建调用此子例行程序后，使用下列步骤创建改变脉冲宽度的控制逻辑：

1. 为新脉冲宽度将字大小数值载入SMW70。
2. 执行PLS指令（以致S7-200对PTO/PWM生成器编程）。
3. 退出子例行程序。

**实例: 脉冲宽度调制 (PWM)**

<p>M A I N</p>	<p><b>Network 1</b></p> <p><b>Network 2</b></p>	<p><b>Network 1</b> //在首次扫描时, //设置图像寄存器位为低位并调用SBR_0.</p> <pre>LD SM0.1 R Q0.1, 1 CALL SBR_0</pre> <p><b>Network 2</b> //在程序的其它地方设置M0.0以改变脉冲宽度到50%工作 //循环.</p> <pre>LD M0.0 EU CALL SBR_1</pre>
<p>S B R 0</p>	<p><b>Network 1</b></p>	<p><b>Network 1</b> //启动子例行程序0: //1. 设置控制字节。 // - 选择PWM操作。 // - 选择毫秒增量和同步更新。 // - 启用载入脉冲宽度和周期时间数值。 // - 启用PWM功能。 //2. 设置周期时间为10,000毫秒。 //3. 设置脉冲宽度为1,000毫秒。 //4. 调用PWM操作: PLS1=&gt;Q0.1。 //5. 为随后的脉冲宽度改变预载控制字节</p> <pre>LD SM0.0 MOVB 16#DB, SMB77 MOVW +10000, SMW78 MOVW +1000, SMW80 PLS 1 MOVB 16#DA, SMB77</pre>
<p>S B R 1</p>	<p><b>Network 1</b></p>	<p><b>Network 1</b> //启动子例行程序1: //设置脉冲宽度为5000毫秒。 //断言脉冲宽度改变。</p> <pre>LD SM0.0 MOVW +5000, SMW80 PLS 1</pre>
<p><b>计时图</b></p> <p>Q0.1</p> <p>10%工作循环   10%工作循环   50%工作循环   50%工作循环</p> <p>周期时间 = 10,000毫秒</p> <p>子例行程序1在此处执行</p>		

## PTO输出的范例操作



### 提示

下列PTO初始化和操作顺序的描述推荐使用“首次扫描”内存位（SM0.1）以初始化脉冲输出。使用“首次扫描”位调用初始化子例行程序减少扫描时间，因为随后的扫描不调用此子例行程序。（“首次扫描”位只在转变为RUN（运行）模式后在首次扫描时设置。）然而，在应用当中可能有其它限制需要初始化（或重新初始化）脉冲输出。在那种情况下，可以使用另一个条件调用初始化例行程序。

### 初始化单段操作的PTO输出

一般地，使用子例行程序为脉冲输出配置和初始化PTO。从主程序调用初始化子例行程序。使用首次扫描内存位（SM0.1）初始化被PTO使用的输出为0，并调用子例行程序完成初始化操作。当使用子例行程序调用时，随后的扫描不调用子例行程序，这减少了扫描执行时间并提供更结构化的程序。

在从主程序创建对初始化子例行程序的调用后，使用下列步骤在初始化子例行程序中创建对配置脉冲输出Q0.0的控制逻辑：

1. 通过载入下列数值之一到SMB67配置控制字节：16#85（选择微秒增量）或16#8D（选择毫秒增量）。  
这些数值都启用PTO/PWM功能，选择PTO操作，设置更新脉冲计数和周期时间数值，选择时基（微秒或毫秒）。
2. 为周期时间将字大小数值载入SMW68。
3. 在SMD72中，载入脉冲计数的双字大小数值。
4. （供选用）一旦脉冲链输出完成，要完成相应功能，可以通过将脉冲链完成事件（中断事件19）附加到中断子例行程序安排一个中断。使用ATCH指令和执行全局中断启用指令ENI。
5. 执行PLS指令（以致S7-200对PTO/PWM生成器编程）。
6. 退出子例行程序。

### 改变PTO周期时间（单段操作）

对于单段PTO操作，可以使用中断例行程序或子例行程序改变周期时间。当使用单段PTO操作时，要改变中断例行程序或子例行程序中的PTO周期时间，按照这些步骤：

1. 通过载入下列数值之一到SMB67，设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基和设置更新周期时间数值）：16#81（微秒）或16#89（毫秒）。
2. 为新周期时间将字大小数值载入SMW68。
3. 执行PLS指令（以致S7-200对PTO/PWM生成器编程）。在启动生成带有更新的周期时间的PTO波形前，S7-200完成任何在过程中的PTO。
4. 退出中断例行程序或子例行程序。

### 改变PTO脉冲计数（单段操作）

对于单段PTO操作，可以使用中断例行程序或子例行程序改变脉冲计数。当使用单段PTO操作时，要改变中断例行程序或子例行程序中的PTO脉冲计数，按照这些步骤：

1. 通过载入下列数值之一到SMB67，设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基和设置更新脉冲计数数值）：16#84（微秒）或16#8C（毫秒）。
2. 在SMD72中，载入新脉冲计数的双字大小数值。
3. 执行PLS指令（以致S7-200对PTO/PWM生成器编程）。在启动生成带有更新的脉冲计数的波形前，S7-200完成任何在过程中的PTO。
4. 退出中断例行程序或子例行程序。

### 改变PTO周期时间和脉冲计数（单段操作）

对于单段PTO操作，可以使用中断例行程序或子例行程序改变周期时间和脉冲计数。当使用单段PTO操作时，要改变中断例行程序或子例行程序中的PTO周期时间和脉冲计数，按照这些步骤：

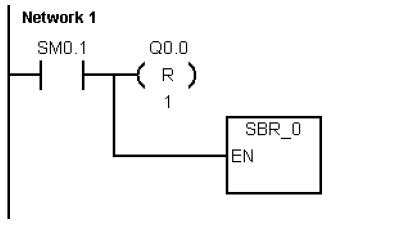
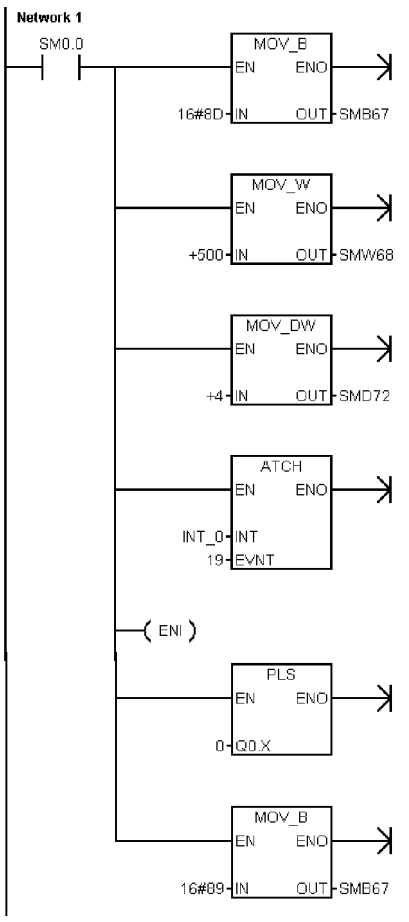
1. 通过载入下列数值之一到SMB67，设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基和设置更新周期时间和脉冲计数数值）：16#85（微秒）或16#8D（毫秒）。
2. 为新周期时间将字大小数值载入SMW68。
3. 在SMC72中，载入新脉冲计数的双字大小数值。
4. 执行PLS指令（以致S7-200对PTO/PWM生成器编程）。在启动生成带有更新的脉冲计数和周期时间的波形前，S7-200完成任何过程中的PTO。
5. 退出中断例行程序或子例行程序。

### 初始化多段操作的PTO输出

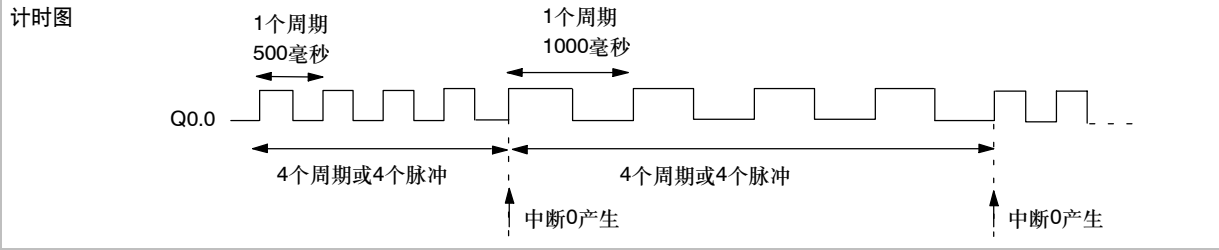
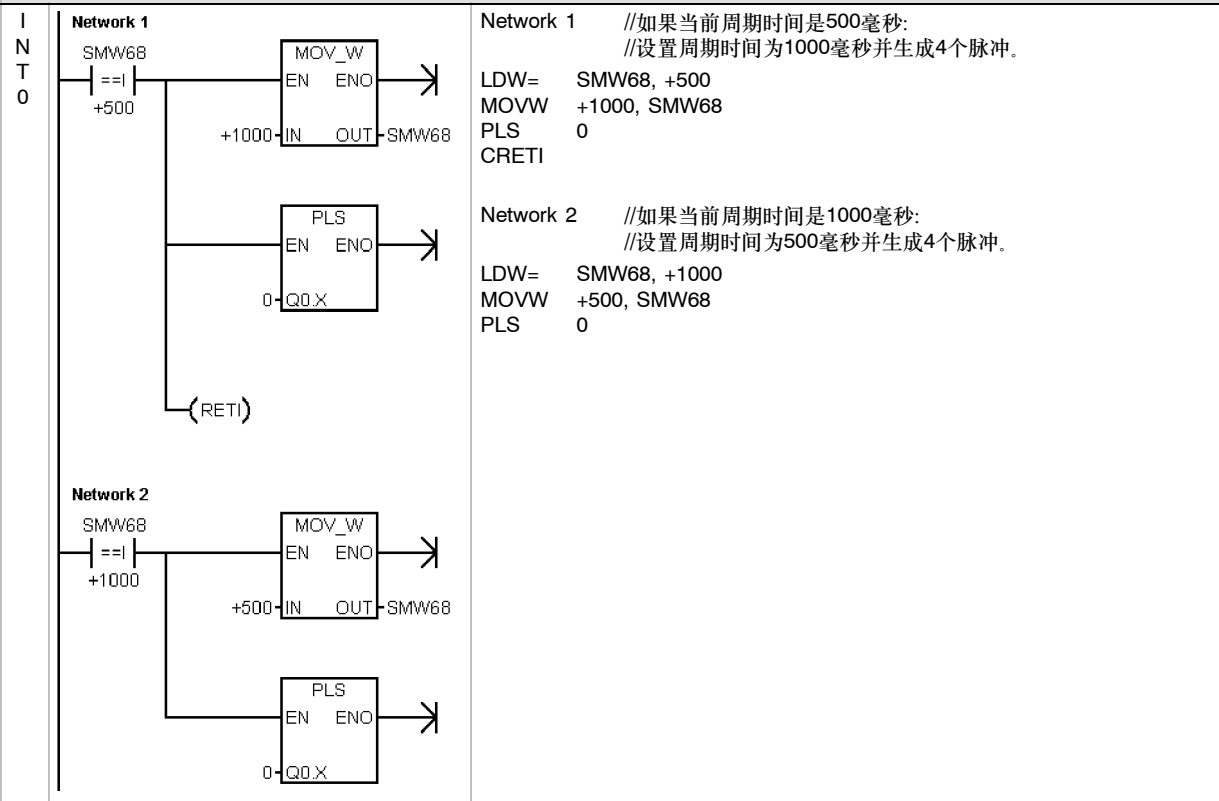
一般地，使用子例行程序为多段操作的脉冲输出配置和初始化PTO。从主程序调用初始化子例行程序。使用首次扫描内存位（SM0.1）初始化被PTO使用的输出为0，并调用子例行程序完成初始化操作。当使用“首次扫描”调用初始化子例行程序，随后的扫描不调用子例行程序，这减少了扫描执行时间。

在从主程序创建对初始化子例行程序的调用后，使用下列步骤在初始化子例行程序中创建对配置脉冲输出Q0.0的控制逻辑：

1. 通过载入下列数值之一到SMB67配置控制字节：16#A0（选择微秒增量）或16#A8（选择毫秒增量）。  
这些数值都启用PTO/PWM功能、选择PTO操作、选择多段操作和选择时基（微秒或毫秒）。
2. 在SMW168中，载入字大小数值用于启动概要表的V内存偏移量。
3. 使用V内存在概要表中设置段数值。确保段域数（表格的第一个字节）是正确的。
4. （供选用）如果在PTO概要图完成后要执行相应的功能，可以通过将脉冲链完成事件（中断事件19）附加到中断子例行程序来对一个中断进行编程。使用ATCH指令和执行全局中断启用指令ENI。
5. 执行PLS指令（以致S7-200对PTO/PWM生成器编程）。
6. 退出子例行程序。

实例: 单段脉冲链操作 (PTO)		
M A I N	<p>Network 1</p> 	<p>Network 1 //在首次扫描时, //设置图像寄存器位为低位并调用子例行程序0。</p> <pre>LD SM0.1 R Q0.0, 1 CALL SBR_0</pre>
S B R 0	<p>Network 1</p> 	<p>Network 1 //启动子例行程序0: 配置PTO</p> <p>//1. 设置控制字节: // - 选择PTO操作。 // - 选择单段操作。 // - 选择毫秒增量。 // - 启用载入脉冲计数和周期时间数值。 // - 启用PTO功能。</p> <p>//2. 设置周期时间为500毫秒。 //3. 设置脉冲计数为4个脉冲。 //4. 为处理PTO整个中断定义要中断的中断例行程序0 //5. 全局中断启用。 //6. 调用PTO操作, PLS0=&gt;Q0.0。 //7. 为随后的周期时间改变预载控制字节。</p> <pre>LD SM0.0 MOVB 16#8D, SMB67 MOVW +500, SMW68 MOVD +4, SMD72 ATCH INT_0, 19 ENI PLS 0 MOVB 16#89, SMB67</pre>

实例：单段脉冲链操作（PTO）（续）



实例: 多段脉冲链操作 (PTO)		
M A I N	<p><b>Network 1</b></p>	<p><b>Network 1</b> //在首次扫描时, //设置图像寄存器位为低位并调用子例行程序0</p> <pre>LD    SM0.1 R     Q0.0, 1 CALL  SBR_0</pre>
S B R 0	<p><b>Network 1</b></p>	<p><b>Network 1</b> //预载PTO概要表: //设置概要表段数为3。 //配置3个段的每一个。 // //1. 配置段1: // - 设置初始周期时间 = 500毫秒。 // - 设置德耳塔周期时间为-2毫秒。 // - 设置脉冲数为200。 //2. 配置段2: // - 设置初始周期时间为100毫秒。 // - 设置德耳塔周期时间为0毫秒。 // - 设置脉冲数为3400。 //3. 配置段3: // - 设置初始周期时间为100毫秒。 // - 设置德耳塔周期时间为1毫秒。 // - 设置脉冲数为400。</p> <pre>LD    SM0.0 MOVB  3, VB500 MOVW  +500, VW501 //段1 MOVW  -2, VW503 MOVD  +200, VD505 MOVW  +100, VW509 //段2 MOVW  +0, VW511 MOVD  +3400, VD513 MOVW  +100, VW517 //段3 MOVW  +1, VW519 MOVD  +400, VD521</pre>

实例: 多段脉冲链操作 (PTO) (续)		
S B R O c o n t i n u e d		<p>Network 2</p> <pre> //1. 设置控制字节: // - 选择PTO操作 // - 选择多段操作 // - 选择毫秒增量 // - 启用PTO功能 //2. 设置概要表的起始地址为V500. //3. 为处理PTO整个中断定义要中断的中断例行程序0 //4. 全局中断启用 //5. 调用PTO操作, PLS0 =&gt; Q0.0.  LD SM0.0 MOVB 16#A8, SMB67 MOVW +500, SMW168 ATCH INT_0, 19 ENI PLS 0                     </pre>
I N T O	<p>Network 1</p>	<p>Network 1</p> <pre> //当PTO输出概要图完成时, //将输出Q0.5接通  LD SM0.0 = Q0.5                     </pre>



## 运算指令

### 加、减、乘和除指令

“加” “减”  
 $IN1 + IN2 = OUT$   $IN1 - IN2 = OUT$  梯形图和功能块图  
 $IN1 + OUT = OUT$   $OUT - IN1 = OUT$  语句表

“加整数” (+I) 或 “减整数” (-I) 指令加或减两个16位整数以产生16位的结果。“加双整数” (+D) 或 “减双整数” (-D) 指令加或减两个32位整数以产生32位的结果。“加实数” (+R) 和 “减实数” (-R) 指令加或减两个32位实数产生一个32位实数结果。

“乘” “除”  
 $IN1 * IN2 = OUT$   $IN1 / IN2 = OUT$  梯形图和功能块图  
 $IN1 * OUT = OUT$   $OUT / IN1 = OUT$  语句表

“乘以整数” (\*I) 或 “除以整数” (/I) 指令乘以或除以两个16位整数产生16位结果。(对于除法, 不保留余数。) “乘以双整数” (\*D) 或 “除以双整数” (/D) 指令乘以或除以两个32位整数产生32位结果。(对于除法, 不保留余数。) “乘以实数” (\*R) 和 “除以实数” (/R) 指令乘以或除以两个32位实数产生一个32位实数结果。

### SM位和ENO

SM1.1指示溢出错误和非法数值。如果SM1.1被设置, 那么SM1.0和SM1.2的状态不是有效的, 原输入操作数不改变。如果SM1.1和SM1.3没有设置, 那么运算操作带有有效的结果完成, SM1.0和SM1.2包含有效的状态。如果在除法操作期间SM1.3被设置, 那么其它运算状态位保持不变。

#### 设置ENO = 0的错误条件

- SM1.1 (溢出)
- SM1.3 (被零除)
- 0006 (间接地址)

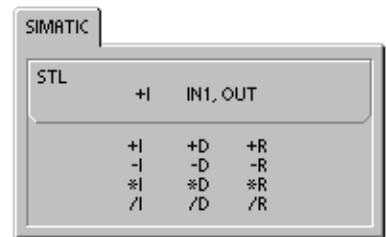
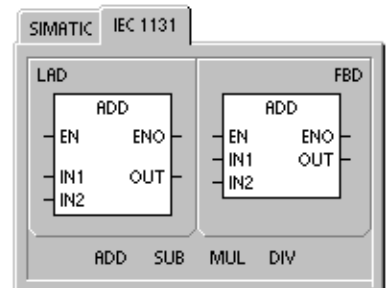
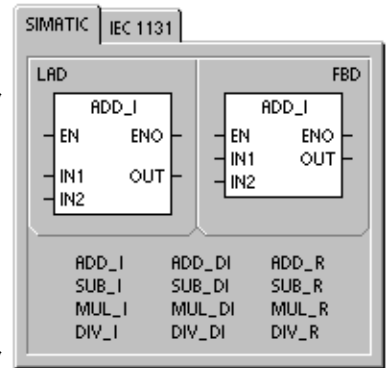
#### 受影响的特殊内存位

- SM1.0 (零)
- SM1.1 (溢出、在操作期间生成的非法数值或发现的非法输入参数)
- SM1.2 (负)
- SM1.3 (被零除)

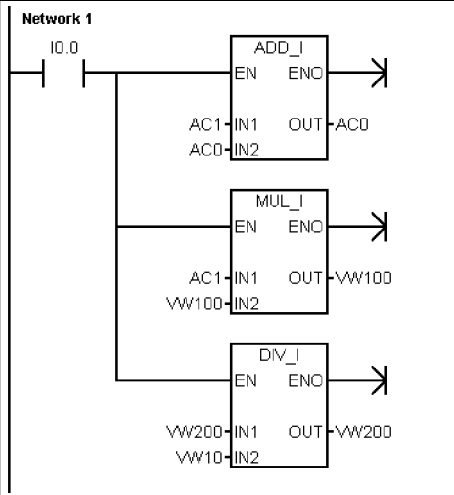
表6-38 加、减、乘和除法指令的有效操作数

输入 / 输出	数据类型	操作数
IN1, IN2	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*AC、*LD、常量
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常量
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常量
OUT	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、*VD、*AC、*LD
	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

实数 (或浮点型) 数字以在ANSI/IEE 754-1985标准 (单精度) 中描述的格式表达。参考该标准以获取更多信息。



实例: 整数运算指令



```

Network 1
LD   I0.0
+I   AC1, AC0
*I   AC1, VW100
/I   VW10, VW200
    
```

添加

40	+	60	=	100
AC1		AC0		AC0

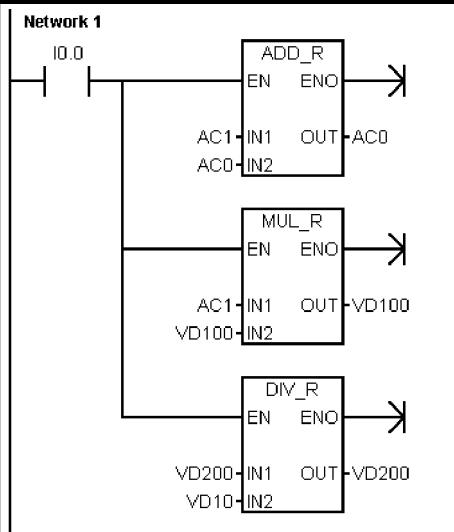
乘

40	*	20	=	800
AC1		VW100		VW100

除

4000	/	40	=	100
VW200		VW10		VW200

实例: 实数运算指令



```

Network 1
LD   I0.0
+R   AC1, AC0
*R   AC1, VD100
/R   VD10, VD200
    
```

添加

4000.0	+	6000.0	=	10000.0
AC1		AC0		AC0

乘

400.0	*	200.0	=	80000.0
AC1		VD100		VD100

除

4000.0	/	41.0	=	97.5609
VD200		VD10		VD200

## 乘以整数到双整数, 除以整数带余数

### 乘以整数到双整数

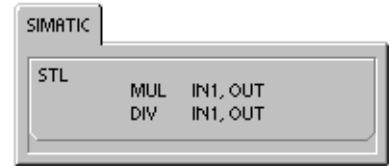
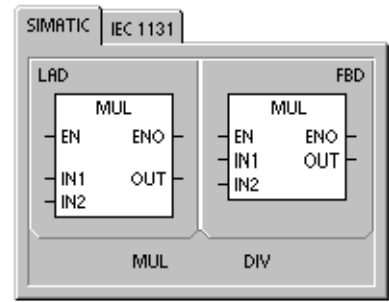
IN1 \* IN2 = OUT 梯形图和功能块图  
 IN1 \* OUT = OUT 语句表

“乘以整数到双整数”指令 (MUL) 将两个16位整数相乘, 并产生一个32位乘积。在语句表MUL指令中, 32位OUT的低位字 (16位) 被用作因子之一。

### 除以整数带余数

IN1 / IN2 = OUT 梯形图和功能块图  
 OUT / IN1 = OUT 语句表

“除以整数带余数”指令 (DIV) 将两个16位整数相除, 产生一个32位结果, 包括一个16位余数 (最高位字) 和一个16位商 (低位字)。



在语句表中, 32位OUT的低位字 (16位) 被用作被除数。

### SM位和ENO

对于本页上的两个指令, 特殊内存 (SM) 位指示错误和非法数值。如果在除法操作期间SM1.3 (被零除) 被设置, 那么其它运算状态位保持不变。否则, 一旦完成运算操作, 所有支持的运算状态位包含有效的状态。

#### 设置ENO = 0的错误条件

- SM1.1 (溢出)
- SM1.3 (被零除)
- 0006 (间接地址)

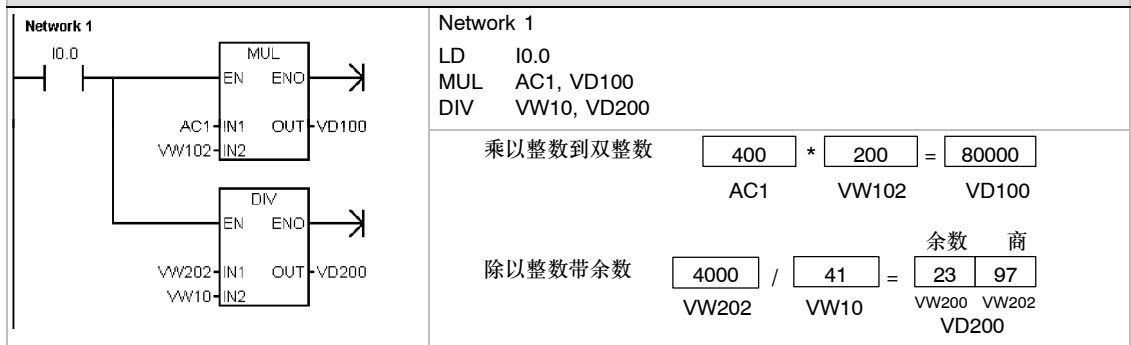
#### 受影响的特殊内存位

- SM1.0 (零)
- SM1.1 (溢出)
- SM1.2 (负)
- SM1.3 (被零除)

表6-39 乘以整数到长整数和除以整数带余数的有效操作数

输入 / 输出	数据类型	操作数
IN1, IN2	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
OUT	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

### 实例: 乘以整数到长整数指令和除以整数带余数指令



注意: VD100包含: VW100和VW102, VD200包含: VW200和VW202。

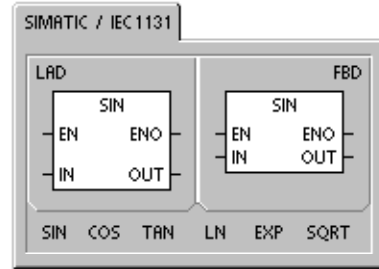
## 数字函数指令

### 正弦、余弦和正切

正弦 (SIN)、余弦 (COS) 和正切 (TAN) 指令计算角数值IN的三角函数，并将结果放在OUT中。输入角数值是以弧度为单位。

$$\text{SIN (IN)} = \text{OUT} \quad \text{COS (IN)} = \text{OUT} \quad \text{TAN (IN)} = \text{OUT}$$

将角度从度数转换为弧度: 使用MUL\_R (\*R)指令将以度数表示的角度乘以1.745329E-2 (大约乘以 $\pi/180$ )。

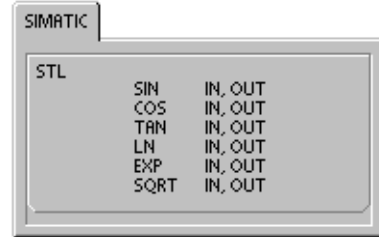


### 自然对数和自然指数

“自然对数”指令 (LN) 完成IN中数值的自然对数，然后把结果放入OUT。

“自然指数”指令 (EXP) 完成e的IN中数值次方的指数操作，并把结果放入OUT。

$$\text{LN (IN)} = \text{OUT} \quad \text{EXP (IN)} = \text{OUT}$$



从自然对数获得基数10的对数: 将自然对数除以2.302585 (大约10的自然对数)。

取任何实数的另一个实数次方, 包含小数指数: 将“自然指数”指令与“自然对数”指令组合在一起。例如, 取X的Y次方, 输入下列指令: EXP (Y \* LN (X))。

### 平方根

“平方根”指令 (SQRT) 取实数 (IN) 的平方根，并产生实数结果OUT。

$$\text{SQRT (IN)} = \text{OUT}$$

要获得其它根:

$$\begin{aligned} 5 \text{的立方} &= 5^3 = \text{EXP}(3 * \text{LN}(5)) = 125 \\ 125 \text{的立方根} &= 125^{(1/3)} = \text{EXP}((1/3) * \text{LN}(125)) = 5 \\ 5 \text{的平方根的三次方} &= 5^{(3/2)} = \text{EXP}(3/2 * \text{LN}(5)) = 11.18034 \end{aligned}$$

### 数字函数指令的SM位和ENO

对于在此页上描述的所有指令，SM1.1用于指示溢出错误和非法数值。如果SM1.1被设置，那么SM1.0和SM1.2的状态不是有效的，原输入操作数不改变。如果SM1.1没有设置，那么运算操作带有有效的结果完成，SM1.0和SM1.2包含有效的状态。

设置ENO = 0的错误条件

- SM1.1 (溢出)
- 0006 (间接地址)

受影响的特殊内存位

- SM1.0 (零)
- SM1.1 (溢出)
- SM1.2 (负)

表6-40 数字函数的有效操作数

输入 / 输出	数据类型	操作数
IN	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常量
OUT	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

实数 (或浮点型) 数字以在ANSI/IEE 754-1985标准 (单精度) 中描述的格式表达。参考该标准以获取更多信息。

## 递增和递减指令

### 增量

IN + 1 = OUT      梯形图和功能块图  
 OUT + 1 = OUT    语句表

### 递减

IN - 1 = OUT      梯形图和功能块图  
 OUT - 1 = OUT    语句表

“递增和递减”指令 添加或减1到或从输入IN，并把结果放入变量OUT。

“递增字节”（INCB）和“递减字节”（DECB）操作是无符号的。

“递增字”（INCW）和“递减字”（DECW）操作是有符号的。

“递增双字”（INCD）和“递减双字”（DECD）操作是有符号的。

设置ENO = 0的错误条件:

- SM1.1 (溢出)
- 0006 (间接地址)

受影响的特殊内存位:

- SM1.0 (零)
- SM1.1 (溢出)
- 用于“字和双字”操作的SM1.2 (负)

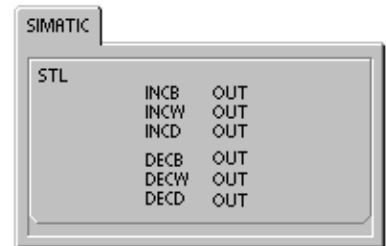
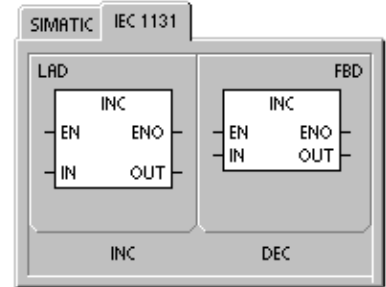
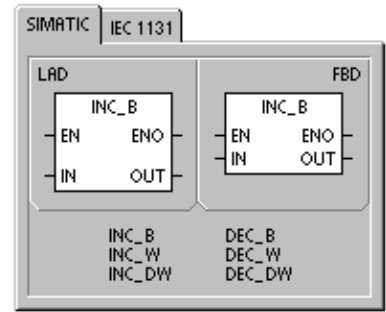
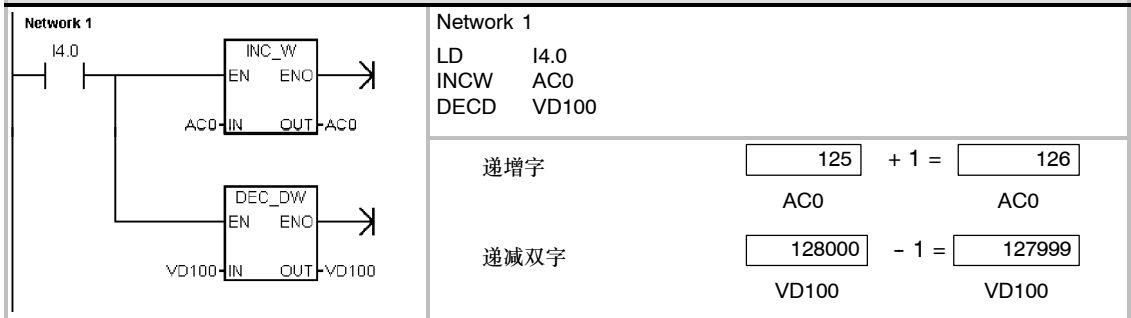


表6-41 “递增和递减指令”的有效操作数

输入 / 输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常量
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*LD、*AC
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

### 实例: 递增和递减指令



## 比例 / 积分 / 微分 (PID) 环路指令

PID环路指令 (PID) 基于表 (TBL) 中的输入和配置信息在参考的循环上执行PID环路计算。

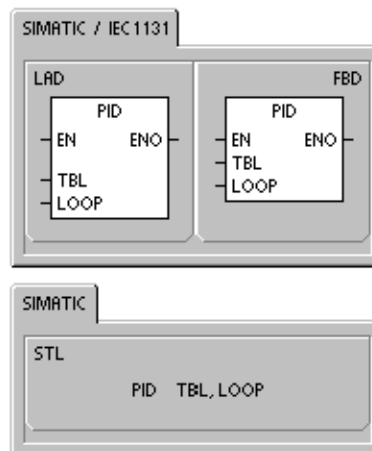
设置 **ENO = 0** 的错误条件:

- SM1.1 (溢出)
- 0006 (间接地址)

受影响的特殊内存位:

- SM1.1 (溢出)

PID环路指令 (比例、积分、微分环路) 提供用于进行PID计算。逻辑堆栈的顶部 (TOS) 必须为ON (功率流) 以启用PID计算。指令有两个操作数: 作为循环表起始地址的“表”地址和从0到7的常量的环路号码。



可以在程序中使用八个PID指令。如果两个或更多PID指令使用同样的环路号码 (即使它们有不同的表地址), PID计算将互相干扰, 输出将不可预知。

循环表存储九个用于控制和监控循环操作的参数, 并包含进程变量、设定值、输出、增益、采样时间、积分时间 (重设)、微分时间 (率) 和积分总和 (偏流) 的当前和前一个数值。

要以期望的采样率完成PID计算, PID指令必须从定时中断例行程序中或从主程序中由由计时器控制率执行。采样时间必须作为输入通过循环表提供给PID指令。

表6-42 PID循环指令的有效操作数

输入 / 输出	数据类型	操作数
TBL	BYTE	VB
LOOP	BYTE	常量 (0到7)



指令向导

STEP 7micro/WIN提供“PID向导”指导定义闭环控制进程的PID算法。选择工具 (Tools) > 指令向导 (Instruction Wizard) 菜单命令, 然后从“指令向导”窗口选择PID。

## 了解PID算法

在稳定的状态操作中，PID控制器调节输出数值以驱动错误（e）为零。错误的测量由设定值（SP）（期望的操作点）和进程变量（PV）（实际的操作点）之间的差给出。PID控制的原理基于下列表达输出的公式M(t)，作为比例项、积分项和微分项的函数：

输出	=	比例项	+	积分项	+	微分项
<b>M(t)</b>	=	<b>K<sub>C</sub> * e</b>	+	<b>K<sub>C</sub> ∫<sub>0</sub><sup>t</sup> e dt + M<sub>initial</sub></b>	+	<b>K<sub>C</sub> * de/dt</b>
在此:	M(t)	是作为时间函数的环路输出				
	K <sub>C</sub>	是环路增益				
	e	是环路错误（设定值和进程变量之间的差）				
	M <sub>initial</sub>	是环路输出的初始值				

为了在数字计算机上实现此控制函数，持续的函数必须被量化成具有随后输出计算的错误数值的周期性采样。作为数字计算机解决方案基本的相应的公式为：

<b>M<sub>n</sub></b>	=	<b>K<sub>C</sub> * e<sub>n</sub></b>	+	<b>K<sub>I</sub> * ∑<sub>1</sub><sup>n</sup></b>	+	<b>M<sub>initial</sub></b>	+	<b>K<sub>D</sub> * (e<sub>n</sub>-e<sub>n-1</sub>)</b>
output	=	比例项	+	积分项	+	微分项		
在此:	M <sub>n</sub>	在采样时间n时的环路输出的计算数值						
	K <sub>C</sub>	是环路增益						
	e <sub>n</sub>	在采样时间n时的环路错误值						
	e <sub>n-1</sub>	前一个环路错误值（在采样时间n-1）						
	K <sub>I</sub>	是积分项的比例常量						
	M <sub>initial</sub>	是环路输出的初始值						
	K <sub>D</sub>	是微分项的比例常量						

从此公式看出，积分项显示为所有出错项（从第一个采样到当前采样）的函数。微分项是当前采样和以前采样的函数，而比例项只是当前采样的函数。在数字计算机上，存储所有的出错项的采样是不实际的，也是不必要的。

因为数字计算机必须从第一个采样开始在每次出错被采样时计算输出值，只有必要存储前一个出错数值和前一个积分项的数值。因为数字计算机解决方案的重复特性，可以使用必须在任何采样时间都能解决的简化公式。简化的公式为：

<b>M<sub>n</sub></b>	=	<b>K<sub>C</sub> * e<sub>n</sub></b>	+	<b>K<sub>I</sub> * e<sub>n</sub> + MX</b>	+	<b>K<sub>D</sub> * (e<sub>n</sub>-e<sub>n-1</sub>)</b>
output	=	比例项	+	积分项	+	微分项
在此:	M <sub>n</sub>	在采样时间n时的环路输出的计算数值				
	K <sub>C</sub>	是环路增益				
	e <sub>n</sub>	在采样时间n时的环路出错值				
	e <sub>n-1</sub>	前一个环路出错值（在采样时间n-1）				
	K <sub>I</sub>	是积分项的比例常量MX是积分项的前一个数值（在采样时间n-1）				
	K <sub>D</sub>	是微分项的比例常量				

在计算环路输出值时，S7-200使用上面简化公式的修改形式。此修改的公式为：

<b>Mn</b>	=	<b>MPn</b>	+	<b>MI<sub>n</sub></b>	+	<b>MDn</b>
output	=	比例项	+	积分项	+	微分项
在此:	Mn	是在采样时间n时的环路输出的计算数值				
	MP <sub>n</sub>	是在采样时间n时环路输出比例项的数值				
	MI <sub>n</sub>	是在采样时间n时环路输出积分项的数值				
	MD <sub>n</sub>	是在采样时间n时环路输出微分项的数值				

### 了解PID公式的比例项

比例项MP是增益（K<sub>C</sub>）的乘积，它控制输出计算的灵敏度，和错误（e）。错误是设定值（SP）和进程变量（PV）在给定采样时间的差。作为S7-200解决方案的比例项公式是：

<b>MPn</b>	=	<b>K<sub>C</sub></b>	*	<b>(SP<sub>n</sub> - PV<sub>n</sub>)</b>	
在此:	MP <sub>n</sub>	是在采样时间n时的环路输出的比例项值			
	K <sub>C</sub>	是环路增益			
	SP <sub>n</sub>	是在采样时间n时的设定值的数值			
	PV <sub>n</sub>	是在采样时间n时进程变量的数值			

### 了解PID公式的积分项

积分项MI在一段时间内的错误总和成比例。作为S7-200解决方案的积分项公式是：

<b>MI<sub>n</sub></b>	=	<b>K<sub>C</sub></b>	*	<b>T<sub>S</sub></b>	/	<b>T<sub>I</sub></b>	*	<b>(SP<sub>n</sub> - PV<sub>n</sub>)</b>	+	<b>MX</b>
在此:	MI <sub>n</sub>	是在采样时间n时的环路输出积分项的数值								
	K <sub>C</sub>	是环路增益								
	T <sub>S</sub>	是环路采样时间								
	T <sub>I</sub>	是环路的积分时期（也称为积分时间或重设）								
	SP <sub>n</sub>	是在采样时间n时的设定点的值								
	PV <sub>n</sub>	是在采样时间n时的进程变量的值								
	MX	是在采样时间n - 1时的积分项的数值（也称为积分和或偏流）								

积分总和或偏流（MX）是积分项所有以前数值的运行总和。在每次计算MI<sub>n</sub>后，偏流被用MI<sub>n</sub>的数值更新，该数值可以被调整或钳位（关于详细资料参见章节“变量和范围”）。偏流的初始值一般地设置为输出值（M<sub>initial</sub>）刚在第一次环路输出计算之前。几个常量也是积分项的部分，增益（K<sub>C</sub>），采样时间（T<sub>S</sub>），它是PID环路重新计算输出值的周期时间，以及积分时间或重设（T<sub>I</sub>），它是在输出计算中用于控制积分项的影响的时间。



### 了解PID公式的微分项

微分项MD与错误中的改变成比例。S7-200使用下列公式作为微分项:

$$MD_n = K_C * T_D / T_S * ((SP_n - PV_n) - (SP_{n-1} - PV_{n-1}))$$

为避免由于在设定值改变的微分操作而引起的输出的阶跃改变或冲击，此公式被修改成假设设定值是常量 ( $SP_n = SP_{n-1}$ )。这导致进程变量中改变的计算，而不是所示错误改变的计算:

$$MD_n = K_C * T_D / T_S * (SP_n - PV_n - SP_{n-1} + PV_{n-1})$$

或只是:

$$MD_n = K_C * T_D / T_S * (PV_{n-1} - PV_n)$$

在此:	$MD_n$	是在采样时间n时环路输出微分项的数值
	$K_C$	是环路增益
	$T_S$	是环路采样时间
	$T_D$	是环路的微分时期 (也称为微分时间或率)
	$SP_n$	是在采样时间n时设定点的值
	$SP_{n-1}$	是在采样时间n-1时设定点的值
	$PV_n$	是在采样时间n时进程变量的值
	$PV_{n-1}$	是在采样时间n-1时进程变量的值

过程变量而不是错误必须保存，用于微分项的下一计算。在第一次采样时， $PV_{n-1}$ 的值初始化为等于 $PV_n$ 。

### 选择环路控制的类型

在许多控制系统中，有必要只应用一种或两种环路控制的方法。例如，可能只需要比例控制或比例积分控制。选择期望的的环路控制类型通过设置常量参数的数值来进行。

如果不希望积分操作 (在PID计算中没有“ $I$ ”)，那么应为积分时间 (重设) 指定数值无穷大“ $INF$ ”。即使没有积分操作，积分项的值可能不是零，因为有积分总和MX的初始值。

如果不希望微分操作 (在PID计算中没有“ $D$ ”)，那么数值0.0应为微分时间 (速率) 指定。

如果不希望比例操作 (在PID计算中没有“ $P$ ”)，并且希望I或ID控制，那么应为增益指定数值0.0。因为环路增益是公式中用于计算积分和微分项的一个因子，将环路增益设置为数值0.0将导致在计算积分和微分项时数值1.0用作环路增益。

## 转换和标准化环路输入

一个环路有两个输入变量，设定值和过程变量。设定值通常是一个固定的数值，诸如在汽车定速控制中的速度设置。过程变量是相关于环路输出的数值，因此权衡环路输出在控制系统上的结果。在定速控制的实例中，过程变量将是测量轮胎转速的转速计输入。

设定值和过程变量是现实世界的数值，其大小、范围和工程单位可能不同。在这些现实世界的数值可以被PID指令操作之前，这些数值必须转换成标准化的、浮点数表达式。

第一步是将现实世界的数值从16位整数转换为浮点或实数数值。提供下列指令序列显示如何从整数值转换为实数。

ITD	AIW0, AC0	//转换输入数值到双字
DTR	AC0, AC0	//转换32位整数到实数

下一步是将现实世界数值的实数值表达式转换为标准化的在0.0和1.0之间的数值。下列公式用于标准化设定值或过程变量值：

$$R_{Norm} = ((R_{Raw} / Span) + Offset)$$

在此：	$R_{Norm}$	是现实世界数值的标准化、实数值表达式
	$R_{Raw}$	是现实世界数值的未标准化或原始的、实数值表达式
	Offset (偏移量)	对单极数值是0.0, 对双极数值是0.5
	Span (扩展)	是最大可能的数值减去最小可能的数值: = 32,000对于单极数值 (典型的) = 64,000对于双极数值 (典型的)

作为前一指令序列的继续，下列指令序列显示如何标准化AC0中的双极数值（其扩展是64,000）：

/R	64000.0, AC0	//标准化累加器中的数值
+R	0.5, AC0	//偏移数值范围从0.0到1.0
MOVR	AC0, VD100	//在环路TABLE中存储标准化数值

## 将环路输出转换为成比例的整数值

环路输出是控制变量，诸如汽车上定速控制的节流阀设置。环路输出是标准化的、在0.0和1.0之间的实数值。在环路输出可以用来驱动模拟输出前，环路输出必须转换为16位、成比例的整数值。此进程是PV和SP到标准化数值的反向转换。第一步是使用下面给出的公式将环路输出转换到成比例的、实数值：

$$RScal = (Mn - Offset) * Span$$

在此：

RScal	是成比例的，环路输出的实数值
M <sub>n</sub>	是标准化的、环路输出的实数值
Offset (偏移量)	对单极数值是0.0, 对双极数值是0.5
Span (扩展)	是最大可能的数值减去最小可能的数值 = 32,000对于单极数值 (典型) = 64,000对于双极数值 (典型)

下列指令顺序显示如何标度环路输出：

```
MOVR VD108, AC0 //移动环路输出到累加器
-R 0.5, AC0 //只有当数值是双极的才包含此语句
*R 64000.0, AC0 //标度累加器中的数值
```

代表环路输出的下一个、标度的实数值必须转换为16位整数。下列指令顺序显示如何做此转换：

```
ROUND AC0, AC0 //转换实数到32位整数
DTI AC0, LW0 //转换数值到16位整数
MOVW LW0, AQW0 //将数值写到模拟输出
```

## 正向或反向作用循环

如果增益是正的，环路是正向作用的，如果增益为负，环路是反向作用的。（对于I或ID控制，当增益数值是0.0时，为积分和微分时间指定正数值将导致正向作用循环，指定负数值将导致反向作用循环。）

### 变量和范围

进程变量和设定值是到PID计算的输入。因此这些变量的循环表域被读出，但不被PID指令改变。

输出值由PID计算生成，所以循环表中的输出值域在每次PID计算完成时更新。输出值被箝位在0.0和1.0之间。当进行从手动控制到PID指令（自动）控制的输出转变时，输出值域可以被用户用作输入以指定初始输出值。（参见下面“模式”部分的讨论）。

如果正使用积分控制，然后偏差数值由PID计算更新，更新数值用作下一个PID计算的输入。当计算的输出值超出范围（输出小于0.0或大于1.0），偏差根据下列公式调整：

$$MX = 1.0 - (MP_n + MD_n) \quad \text{当计算输出 } M_n > 1.0 \text{ 时}$$

或

$$MX = - (MP_n + MD_n) \quad \text{当计算输出 } M_n < 0.0 \text{ 时,}$$

在此：

MX	是调整的偏差的数值
MP <sub>n</sub>	是在采样时间n时环路输出的比例项的数值
MD <sub>n</sub>	是在采样时间n时环路输出的微分项的数值
M <sub>n</sub>	是在采样时间n时环路输出的数值

通过根据描述调整偏差，一旦计算输出回到正确的范围，系统响应将获得改进。计算偏差被箝位在0.0和1.0之间，然后在每次完成PID计算时写到循环表的偏差域。存储在循环表中的数值用于下一个PID计算。

循环表中的偏差值可在执行PID指令前由用户修改，以便在某些应用状况下对偏差值进行编址。在手动调整偏差时必须当心，任何写入循环表的偏差数值必须在0.0和1.0之间的实数。

进程变量的比较数值在循环表中保持，以用于PID计算的微分操作部分。不应修改此数值。

## 模式

对于S7-200 PID循环，没有内置模式控制。只有当功率流到PID方框时，才能进行PID计算。因此，当PID计算周期性地进行时，存在“自动的”或“自动”模式。当PID计算不进行时，存在“手动”模式。

PID指令有一个功率流历史位，与计数器指令相似。指令使用此历史位来检测0到1功率流转变。当检测到功率流转变，将引起指令进行一系列操作以便无波动地从手动控制改变到自动控制。为了无波动地改变到自动模式控制，在转换到自动控制之前，由手动控制设置的输出值必须作为到PID指令的输入提供（对于 $M_n$ 写到循环表条目）。当检测到0到1的功率流转变时，PID指令对循环表中的数值进行下列操作，以确保从手动到自动控制的无波动改变。

- 设置定点 ( $SP_n$ ) = 进程变量 ( $PV_n$ )
- 设置旧进程变量 ( $PV_{n-1}$ ) = 进程变量 ( $PV_n$ )
- 设置偏差 ( $MX$ ) = 输出值 ( $M_n$ )

PID历史位的默认状态是“设置”，在启动和每次控制器停止到运行模式转变时建立该状态。如果在输入RUN（运行）模式后第一次执行功率流到PID方框，那么检测不到功率流转变，不进行无波动模式改变操作。

## 报警检查和特殊操作

PID指令是进行PID计算的简单但功能强大的指令。如果需要其它进程诸如报警检查或环路变量的特殊计算，这些必须使用S7-200支持的基本指令实现。

## 错误条件

当到了编译的时间，如果循环表起始地址或在指令中指定的PID循环数字操作数超出范围，CPU将生成编译错误（范围错误），编译失败。

某些循环表输入数值没有被PID指令进行范围检查。必须当心以确保进程变量和设定值（以及偏差和前一个进程变量，如果用作输入）是在0.0和1.0之间的实数。

如果在进行PID计算的运算操作时，遇到出错，那么SM1.1（溢出或非法数值）被设置，PID指令的执行被终止。（循环表中输出值的更新可能不完全，所以可以不管这些数值，在环路PID指令的下次执行之间更正引起运算出错的输入数值。）

## 循环表

循环表有36个字节长，具有在表6-43中显示的格式。

表6-43 循环表

偏移量	域	格式	类型	描述
0	进程变量 ( $PV_n$ )	双字 - 实数	输入	包含必须在0.0和1.0之间的进程变量。
4	设定值 ( $SP_n$ )	双字 - 实数	输入	包含必须在0.0和1.0之间的设定值。
8	输出 ( $M_n$ )	双字 - 实数	输入 / 输出	包含必须在0.0和1.0之间的计算输出。
12	增益 ( $K_C$ )	双字 - 实数	输入	包含是比例常量的增益。可以是正或负数。
16	采样时间 ( $T_S$ )	双字 - 实数	输入	包含采样时间，单位为秒。必须是正数。
20	积分时间或重设 ( $T_I$ )	双字 - 实数	输入	包含积分时间或重设，以分钟为单位。必须是正数。
24	微分时间或速率 ( $T_D$ )	双字 - 实数	输入	包含微分时间或速率，以分钟为单位。必须是正数。
28	偏差 ( $MX$ )	双字 - 实数	输入 / 输出	包含在0.0和1.0之间的偏差或积分总和值
32	前一个进程变量 ( $PV_{n-1}$ )	双字 - 实数	输入 / 输出	包含从最后一次执行PID指令中存储的进程变量数值的前一数值。

## PID程序实例

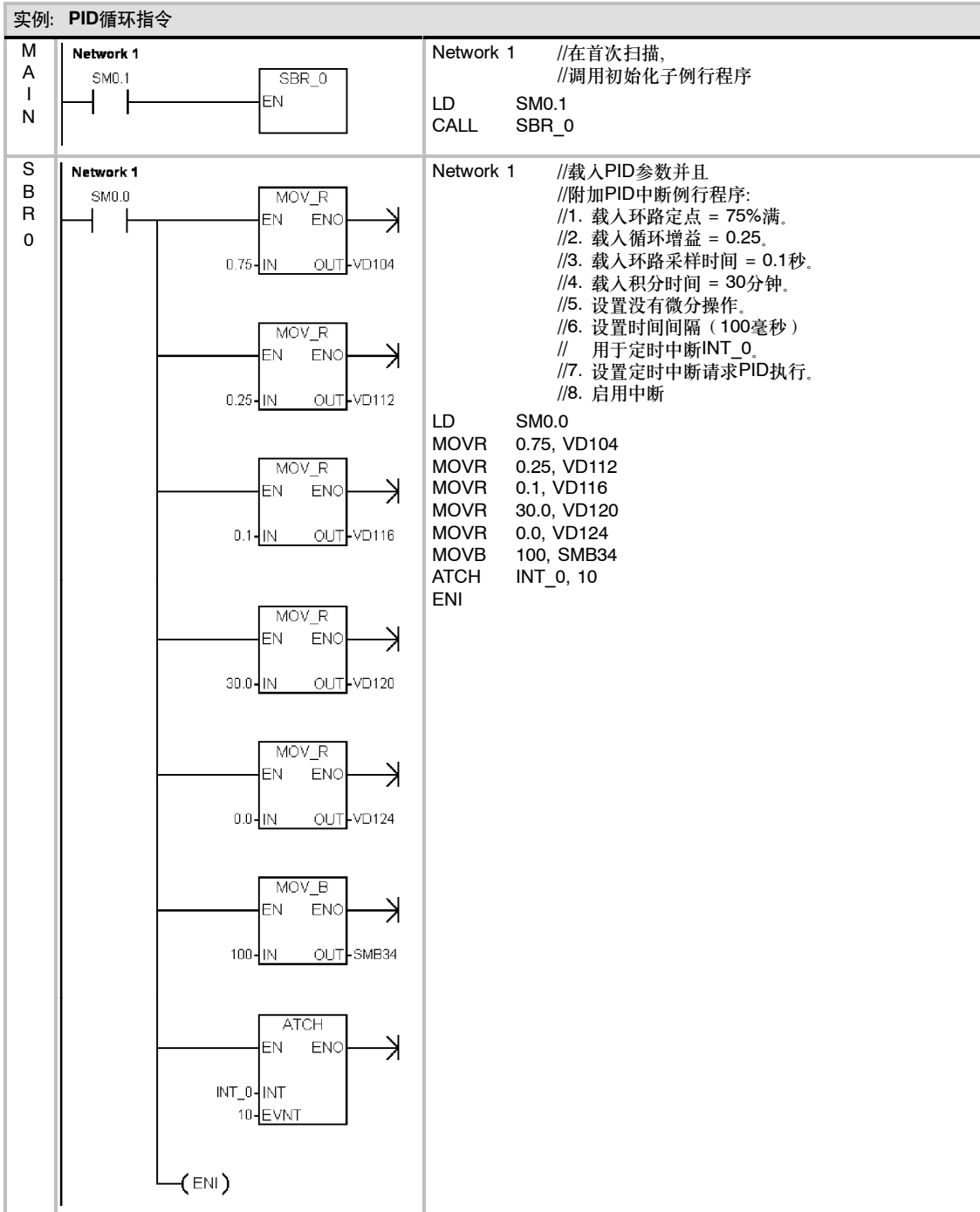
在此实例中，水储罐用于保持恒定水压。水以变化的速率不断地从水储罐取出。变速泵用于以保持充足水压的速率添加水到储罐，并且也防止储罐空。

此系统的设定值等于储罐达到充满75%水位的设置。进程变量由浮点型测量器提供，它提供储罐充满程度的相同读数，可以从0%（或空）到100%（或全部满）之间变化。输出是泵速的数值，允许泵从最大速度的0%到100%运行。

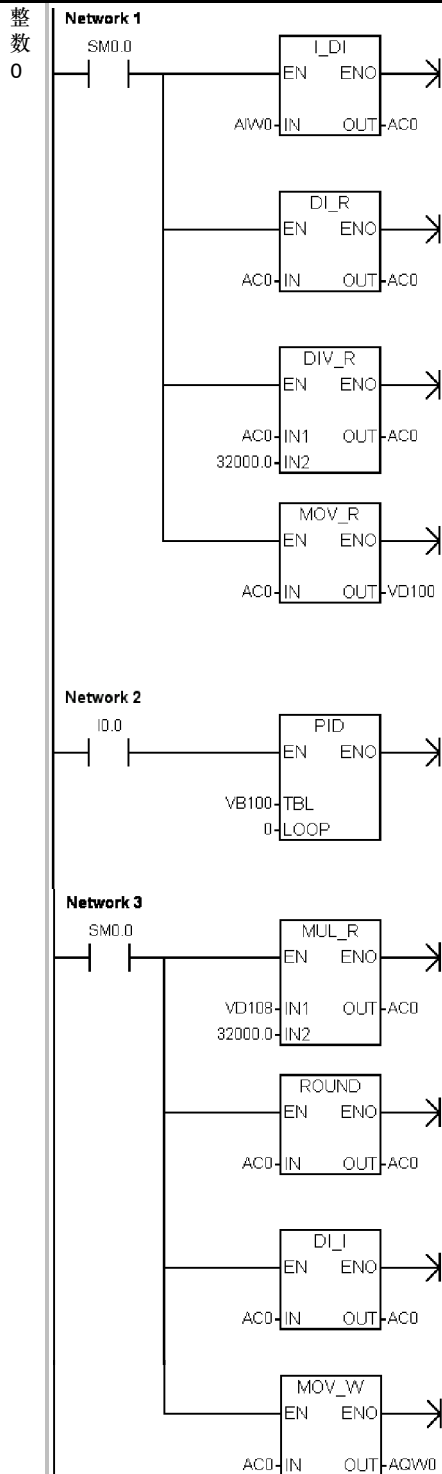
设定值是预先确定的，直接输入循环表。进程变量作为来自浮点型测量器的单极、模拟数值提供。环路输出写入用于控制泵速的单极、模拟输出。模拟输入和模拟输出的Span（扩展）都是32,000。

只有比例和积分控制可应用于此例。循环增益和时间常量从工程计算中确定，可以根据需要调整以获得最佳控制。时间常量的计算数值是： $K_C = 0.25$ ， $T_S = 0.1$ 秒， $T_I = 30$ 分钟。

泵速是手动控制的，直到水储罐为75%满，阀打开允许水从储罐排出。同时，泵从手动切换到自动控制模式。数字输入用于将控制从手动切换到自动。此输入（I0.0）手动/自动控制：0 = 手动，1 = 自动。当处于手动控制模式，泵速度由操作员按从0.0到1.0的实数值写到VD108。



## 实例: PID循环指令 (续)



**Network 1** //标度PV到标准化的实数:  
//1. 转换整数值到双整数。  
//2. 转换双整数到实数。  
//3. 标准化数值。  
//4. 在循环表中存储标准化的PV。

```
LD SM0.0
ITD AIW0, AC0
DTR AC0, AC0
/R 32000.0, AC0
MOVR AC0, VD100
```

**Network 2** //当置于自动模式时执行环路。

```
LD I0.0
PID VB100, 0
```

**Network 3** //标度输出Mn到整数。  
//Mn是单极数值, 不能为负。  
//1. 将环路输出移动到累加器。  
//2. 在累加器中标度数值。  
//3. 转换实数到双整数。  
//4. 转换双整数至整数。  
//5. 将数值写到模拟输出。

```
LD SM0.0
MOVR VD108, AC0
*R 32000.0, AC0
ROUND AC0, AC0
DTI AC0, AC0
MOVW AC0, AQW0
```

## 中断指令

### 启用中断和禁用中断

启用中断指令（ENI）全局启用所有附加中断事件的处理。禁用中断指令（DISI）全局禁用所有中断事件的处理。

当转变到RUN（运行）模式时，中断开始禁用。在RUN（运行）模式，可以通过执行“启用中断”指令启用中断进程。执行“禁用中断”指令禁止中断的处理；然而，现用中断事件将继续排队。

设置ENO = 0的错误条件

- 0004（在中断例行程序中尝试执行ENI、DISI或HDEF指令）

### 从中断指令有条件返回

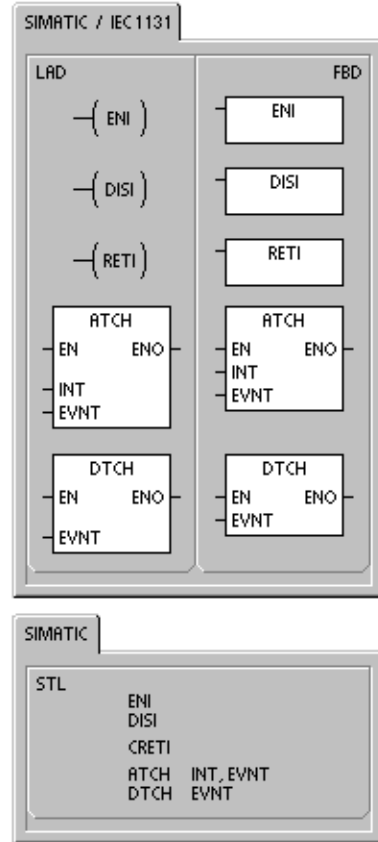
“从中断有条件返回”指令（CRETI）可以用于基于前面的逻辑条件从中断返回。

### 附加中断

附加中断指令（ATCH）将中断事件EVNT与中断例行程序数字INT联合起来，并启用中断事件。

设置ENO = 0的错误条件:

- 0002（输入HSC的冲突分配）



### 分离中断

分离中断指令（DTCH）将中断事件EVNT从所有中断例行程序分离，并禁用中断事件。

表6-44 附加中断和分离中断指令的有效操作数

输入 / 输出	数据类型	操作数
INT	BYTE	常量 (0到127)
EVNT	BYTE	常量 CPU 221 和CPU 222: 0到12、19到23和27到33 CPU 224: 0到23和27到33 CPU 226 和CPU 226XM: 0至33

### 附加中断和分离中断的操作指令

在中断例行程序可以被调用前，必须在中断事件和当事件产生时希望执行的程序段之间建立联系。使用“附加中断”指令联系中断事件（由中断事件号指定）和程序段（由中断例行程序号指定）。可以将多种中断事件附加到一个中断例行程序，但是一个事件不能同时被附加到多种中断例行程序。

当附加中断事件到中断例行程序时，该中断自动启用。如果使用全局禁用中断指令禁用所有中断，每个中断事件的发生排队直到使用全局启用中断指令中断被重新启用，或者中断排队溢出。



可以禁用单个中断事件通过使用“分离中断”指令断开中断事件和中断例行程序之间的联系。“分离中断”指令将中断返回非现用或忽略状态。表6-45列出不同类型的中断事件。

表6-45 中断事件

事件	描述	CPU 221 CPU 222	CPU 224	CPU 226 CPU 226XM
0	I0.0 上升边缘	Y	Y	Y
1	I0.0 下降边缘	Y	Y	Y
2	I0.1 上升边缘	Y	Y	Y
3	I0.1 下降边缘	Y	Y	Y
4	I0.2 上升边缘	Y	Y	Y
5	I0.2 下降边缘	Y	Y	Y
6	I0.3 上升边缘	Y	Y	Y
7	I0.3 下降边缘	Y	Y	Y
8	端口0 接收字符	Y	Y	Y
9	端口0 传输完成	Y	Y	Y
10	定时中断0 SMB34	Y	Y	Y
11	定时中断1 SMB35	Y	Y	Y
12	HSC0 CV=PV (当前值 = 预设值)	Y	Y	Y
13	HSC1 CV=PV (当前值 = 预设值)		Y	Y
14	HSC1 方向改变		Y	Y
15	HSC1 外部重设		Y	Y
16	HSC2 CV=PV (当前值 = 预设值)		Y	Y
17	HSC2 方向改变		Y	Y
18	HSC2 外部重设		Y	Y
19	PLS0 PTO脉冲计数完成中断	Y	Y	Y
20	PLS1 PTO脉冲计数完成中断	Y	Y	Y
21	计时器T32 CT=PT中断	Y	Y	Y
22	计时器T96 CT=PT中断	Y	Y	Y
23	端口0 接收讯息完成	Y	Y	Y
24	端口1 接收讯息完成			Y
25	端口1 接收字符			Y
26	端口1 传输完成			Y
27	HSC0 方向改变	Y	Y	Y
28	HSC0 外部重设	Y	Y	Y
29	HSC4 CV=PV (当前值 = 预设值)	Y	Y	Y
30	HSC4 方向改变	Y	Y	Y
31	HSC4 外部重设	Y	Y	Y
32	HSC3 CV=PV (当前值 = 预设值)	Y	Y	Y
33	HSC5 CV=PV (当前值 = 预设值)	Y	Y	Y

## 了解S7-200如何处理中断例行程序

中断例行程序响应相关的内部或外部事件而执行。一旦中断例行程序的最后一个指令已执行，控制返回到主程序。可以通过执行“从中断有条件返回”指令（CRETI）退出例行程序。表6-46强调关于在程序中使用中断例行程序的一些指南和限制。

表6-46 使用中断例行程序的指南和限制

参考标准
<p>中断处理提供到特殊内部或外部事件的快速反应。应该优化中断例行程序以完成指定任务，然后将控制返回主例行程序。</p> <p>通过保持中断例行程序简短扼要，使执行快速，其它进程不会延迟太长时间。如果不这样做，意外的条件可能引起受主程序控制设备的异常操作。对于中断，公理“越短越好”是相当正确的。</p>
限制
<p>不能在中断例行程序中使用“禁用中断”（DISI）、“启用中断”（ENI）、“高速计数器定义”（HDEF）和“结束”（END）指令。</p>

## 6

### 中断的系统支持

因为接点、线圈和累加器逻辑可能受中断影响，系统保存和重新装载逻辑堆栈、累加器寄存器和指示累加器和指令操作状态的特殊内存位（SM）。这避免由于跳转到和来自中断例行程序引起的对主用户程序的破坏。

### 在主程序和中断例行程序之间共享数据

可以在主程序和一个或更多的中断例行程序之间共享数据。因为不可能预知S7-200何时会产生中断，所以对中断程序和程序中的其他部分都要用到的变量的数目加以限制是有必要的。由于中断例行程序的操作，当执行主程序中的指令被中断事件中断时，可能引起共享数据的一致性问题。使用中断例行程序的局部变量来确保中断例行程序只使用临时内存和不重写在程序的其它地方使用的数据。

存在可使用的多种编程技术，都可确保数据在主程序和中断程序之间正确共享。这些技术或者限制存取共享内存位置的方式或者防止使用共享的内存位置中断指令序列。

- 对于语句表程序，那是共享单个变量：如果共享数据是单字节、字或双字变量，而程序用语句表写，那么正确共享存取可以通过只在非共享的内存位置或累加器中存储来自共享数据上操作的立即数值来确保。
- 对于梯形程序，那是共享单个变量：如果共享数据是单字节、字或双字变量，而程序用梯形图编写，那么正确的共享存取可以通过建立只使用“移动”指令（MOVB、MOVW、MOVD、MOVR）进行存取共享的内存位置的惯例来确保。而许多梯形图指令由可中断语句表指令序列组成，这些“移动”指令由单个语句表指令组成，其执行不会受中断事件影响。
- 对于语句表或梯形程序，共享多种变量：如果共享数据由许多相关字节、字或双字组成，那么中断禁用/启用指令（DISI和ENI）可以用于控制中断例行程序执行。在主程序中共享的内存位置的操作开始的地方，禁用中断。一旦所有影响共享位置的操作完成，重新启用中断。在中断禁用的时候，中断例行程序不能执行，因此不能存取共享的内存位置；然而，此方法可能导致对中断事件的延迟响应。

### 从中断例行程序调用子例行程序

可以从中断例行程序调用子例行程序的一个嵌套层。累加器和逻辑堆栈在中断例行程序和调用的子例行程序之间共享。

## S7-200支持的中断类型

S7-200支持下列中断例行程序类型:

- 通讯端口中断: S7-200生成允许用户程序控制通讯端口的事件。
- I/O中断: S7-200生成各种I/O状态不同改变的事件。这些事件允许用户程序响应高速计数器、脉冲输出或响应输入的上升或下降状态。
- 时基中断: S7-200生成允许程序反应指定间隔的事件。

### 通讯端口中断

S7-200串行通讯口可以由程序控制。这种操作通讯端口的模式称为“自由端口”模式。在“自由端口”模式中, 用户程序定义波特率、每个字符的位数、奇偶校验和协议。“接收”和“传输”中断可用于促进程序控制的通讯。关于更多的信息参考“传输和接收”指令。

### I/O中断

I/O中断包含上升/下降边缘中断、高速计数器中断和脉冲串输出中断。S7-200可以在输入的上升和/或下降边缘生成中断(I0.0、I0.1、I0.2或者I0.3)。上升边缘和下降边缘事件可以为这些输入点的每个捕获。这些上升/下降边缘事件可以用来表示当事件发生时必须接收立即注意的条件。

高速计数器中断允许响应诸如达到预设值的当前值、符合轴旋转方向反转的计数方向改变、或计数器外部重设的条件。这些高速计数器事件的每个允许实时进行操作, 以响应无法以可编程逻辑控制器扫描速度控制的高速事件。

脉冲串输出中断提供完成输出规定数目的时钟脉冲的立即通知。脉冲串输出的典型使用是步进电机控制。

可以通过将中断例行程序附加到相应I/O事件启用以上每一个中断。

### 时基中断

时基中断包含定时中断和计时器T32/T96中断。可以使用定时中断指定以周期为基础进行的操作。周期时间以1毫秒递增, 从1毫秒到255毫秒。必须为定时中断0在SMB34中写周期时间, 为定时中断1在SMB35中写周期时间。

定时中断事件每次计时器到期时传送控制到合适的中断例行程序。一般地, 使用定时中断控制模拟输入的采样或以定期间隔执行PID循环。

定时中断启用, 当将中断例行程序附加到定时中断事件时定时开始。在附加期间, 系统捕获周期时间数值, 所以随后对SMB34和SMB35的改变不影响周期时间。要改变周期时间, 必须修改周期时间数值, 然后再将中断例行程序重新附加到定时中断事件。当再附加产生时, 定时中断功能从以前的附加清除任何积累的时间, 以新数值开始计时。

在被启用后，定时中断连续运行，在每个指定时间间隔到期时执行附加中断例行程序。如果退出RUN（运行）模式或分离定时中断，定时中断禁用。如果全局禁用中断指令执行，定时中断继续产生。每次定时中断的发生排队（直到中断启用或排队满）。

计时器T32/T96中断允许及时地响应指定时间间隔的完成。这些中断只受1毫秒分辨率的接通延迟（TON）和断开延迟（TOF）计时器T32和T96支持。否则，T32和T96计时器正常运行。一旦中断启用，当现用计时器的当前值等于在标准1毫秒计时器更新完成期间在S7-200中预置时间数值时，附加中断例行程序执行。通过将中断例行程序附加到T32/T96中断事件启用这些中断。

## 中断优先级和排队

S7-200的中断按各自的优先级组以先来先服务原则服务。在任何一个时间点，只有一个用户中断例行程序被执行。一旦执行中断例行程序开始，该例行程序执行到完成。它不能被任何其它中断例行程序预清空，即使是具有更高优先级的例行程序。当另一个中断被处理时发生的中断被排队用于以后处理。

表6-47显示三个中断队列和它们可以存储的最大中断数。

表6-47 每个中断队列的最大条目数

队列	CPU 221、CPU 222、CPU 224	CPU 226和CPU 226XM
通讯队列	4	8
I/O中断队列	16	16
定时中断队列	8	8

潜在地，比队列能保存的更多的中断可能产生。因此，队列溢出内存位（识别已丢失的中断事件的类型）被系统保持。表6-48显示中断队列溢出位。应该只在中断例行程序中使用这些位，因为当队列清空时它们重置，控制返回到主程序。

表6-49显示所有中断事件，以及它们的优先级和分配的事件号。

表6-48 中断队列溢出位

说明（0 = 没有溢出，1 = 溢出）	SM位
通讯队列	SM4.0
I/O中断队列	SM4.1
定时中断队列	SM4.2

表6-49 中断事件的优先级顺序

事件	描述	优先级组	组中的优先级
8	端口0 接收字符	通讯 最高优先级	0
9	端口0 传输完成		0
23	端口0 接收讯息完成		0
24	端口1 接收讯息完成		1
25	端口1 接收字符		1
26	端口1 传输完成		1
19	PLS0 PTO脉冲计数完成中断	离散的 中优先级	0
20	PLS1 PTO脉冲计数完成中断		1
0	I0.0 上升边缘		2
2	I0.1 上升边缘		3
4	I0.2 上升边缘		4
6	I0.3 上升边缘		5
1	I0.0 下降边缘		6
3	I0.1 下降边缘		7
5	I0.2 下降边缘		8
7	I0.3 下降边缘		9
12	HSC0 CV=PV (当前值 = 预设值)		10
27	HSC0 方向改变		11
28	HSC0 外部重设		12
13	HSC1 CV=PV (当前值 = 预设值)		13
14	HSC1 方向改变		14
15	HSC1 外部重设		15
16	HSC2 CV=PV (当前值 = 预设值)		16
17	HSC2 方向改变		17
18	HSC2 外部重设		18
32	HSC3 CV=PV (当前值 = 预设值)		19
29	HSC4 CV=PV (当前值 = 预设值)		20
30	HSC4 方向改变		21
31	HSC4 外部重设		22
33	HSC5 CV=PV (当前值 = 预设值)	23	
10	定时中断0 SMB34	定时的 最低优先级	0
11	定时中断1 SMB35		1
21	计时器T32 CT=PT中断		2
22	计时器T96 CT=PT中断		3

实例: 中断指令		
<p>M A I N</p>	<p>Network 1 //在第一次扫描: //1. 定义中断例行程序INT_0为I0.0的下降边缘中断 //2. 全局启用中断。 LD SM0.1 ATCH INT_0, 1 ENI</p> <p>Network 2 //如果检测到I/O错误, //禁用I0.0的下降边缘中断。 //此程序段是可选的。 LD SM5.0 DTCH 1</p> <p>Network 3 //当M5.0为接通时, //禁用所有中断。 LD M5.0 DISI</p>	
<p>I N T 0</p>	<p>Network 1 //I0.0下降边缘中断例行程序: //基于I/O错误的有条件返回。 LD SM5.0 CRETI</p>	

实例: 读模拟输入数值的定时中断		
<p>M A I N</p>	<p>Network 1 //一旦首先扫描, 则调用子例行程序0。 LD SM0.1 CALL SBR_0</p>	
<p>S B R 0</p>	<p>Network 1 //1. 设置定时中断的间隔为0到100毫秒。 //2. 附加定时中断0 (事件10) 到INT_0。 //3. 全局中断启用。 LD SM0.0 MOVB 100, SMB34 ATCH INT_0, 10 ENI</p>	
<p>I N T 0</p>	<p>Network 1 //每隔100毫秒读AIW4的值 LD SM0.0 MOVW AIW4, VW100</p>	

## 逻辑操作指令

### 反转指令

#### 反转字节、字和双字

“反转字节”（INVB）、“反转字”（INW）和“反转双字”（INVD）指令形成输入IN的反码，并将结果载入内存位置OUT。

设置ENO = 0的错误条件

- 0006（间接地址）

受影响的SM位:

- SM1.0（零）

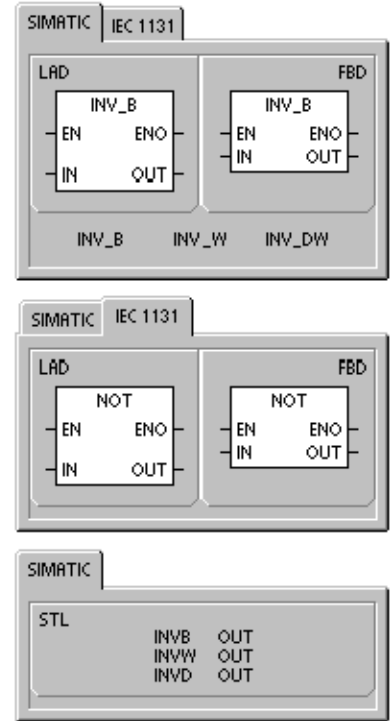


表6-50 反转指令的有效操作数

输入 / 输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常量
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC
	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、AC、*VD、*LD、*AC
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

#### 实例: 反转指令



## AND（与）、OR（或）和Exclusive OR（异或）指令

### AND（与）字节、AND（与）字和AND（与）双字

“AND（与）字节”（ANDB）、“AND（与）字”（ANDW）和“AND（与）双字”（ANDD）指令将两个输入数值IN1和IN2的相应位作AND（与）运算，并将结果载入内存位置OUT。

### OR（或）字节、OR（或）字和OR（或）双字

“OR（或）字节”（ORB）、“OR（或）字”指令（ORW）和“OR（或）双字”（ORD）指令将两个输入数值IN1和IN2的相应位作OR（或）运算，并将结果载入内存位置OUT。

### Exclusive OR（异或）字节、Exclusive OR（异或）字和Exclusive OR（异或）双字

“Exclusive OR（异或）字节”（XORB）、“Exclusive OR（异或）字”（XORW）和“Exclusive OR（异或）双字”（XORD）指令将两个输入数值IN1和IN2的相应位作XOR（异或）运算，并将结果载入内存位置OUT。

### SM位和ENO

对于在此页描述的所有指令，下列条件影响SM位和ENO。

设置ENO = 0的错误条件

- 0006（间接地址）

受影响的SM位:

- SM1.0（零）

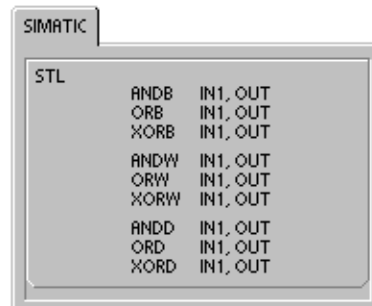
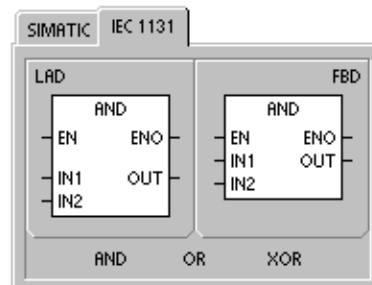
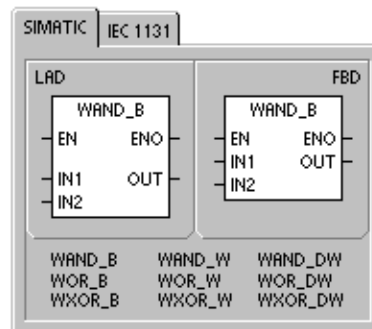
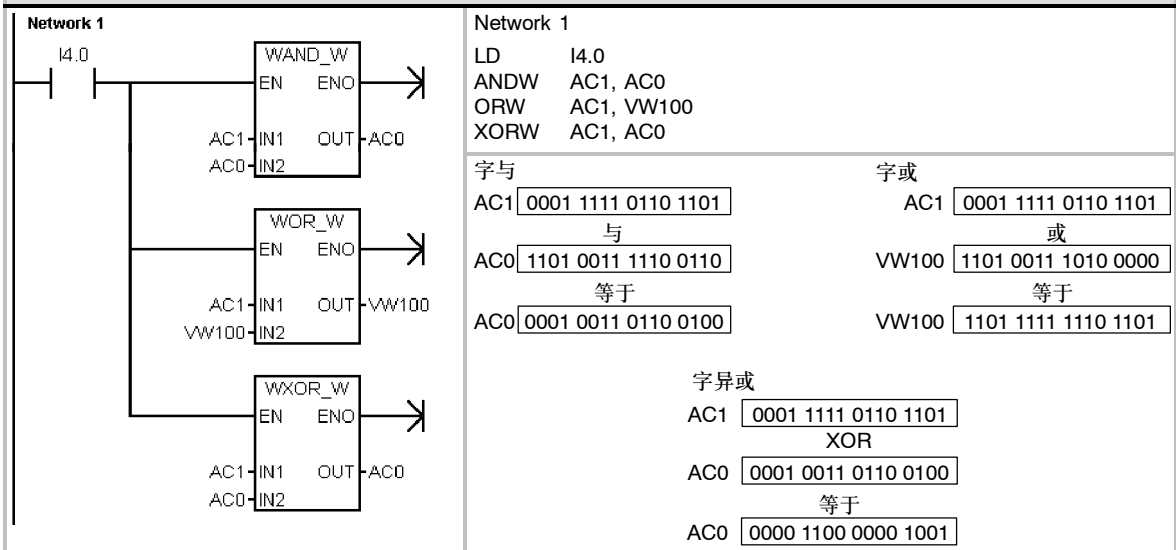


表6-51 AND（与）、OR（或）和Exclusive OR（异或）指令的有效操作数

输入 / 输出	数据类型	操作数
IN1、IN2	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常量
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*AC、*LD
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD



## 实例: AND (与)、OR (或) 和Exclusive OR (异或) 指令



## 移动指令

### 移动字节、字、双字或实数

“移动字节”（MOVB）、“移动字”（MOVW）、“移动双字”（MOVD）和“移动实数”（MOVR）指令将来自内存位置IN的数值移动到新内存位置OUT，而不改变原数值。

使用“移动双字”指令创建指针。关于更多的信息，请参考在第4章中的指针和间接地址部分。

对于IEC“移动”指令，输入和输出数据类型可以变化，但是必须是相同尺寸的。

设置ENO = 0的错误条件

- 0006（间接地址）

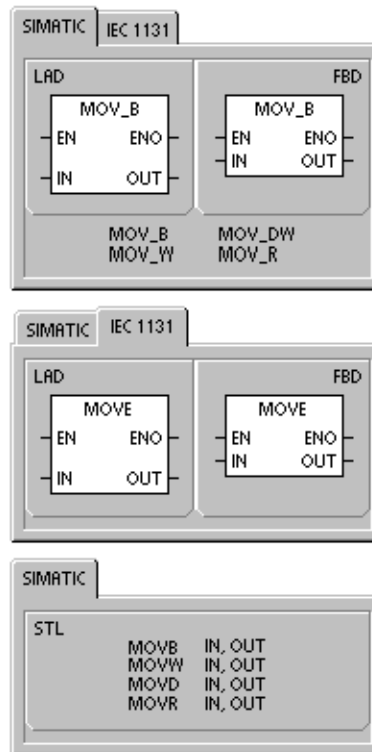


表6-52 移动指令的有效操作数

输入 / 输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
	WORD、INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*AC、*LD、常量
	DWORD、DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、&IB、&QB、&VB、&MB、&SB、&T、&C、*VD、*LD、*AC、常量
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常量
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC
	WORD、INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC
	DWORD、DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

## 立即移动字节（读和写）

“立即移动字节”指令允许在物理I/O和内存位置之间立即移动字节。

“移动字节立即读”（BIR）指令读物理输入（IN），并将结果写到内存地址（OUT），但是进程图像寄存器不更新。

“移动字节立即写”指令（BIW）从内存地址（IN）读数据，并写到物理输出（OUT）和相应的进程映像位置。

设置ENO = 0的错误条件

- 0006（间接地址）
- 不能存取扩充模块

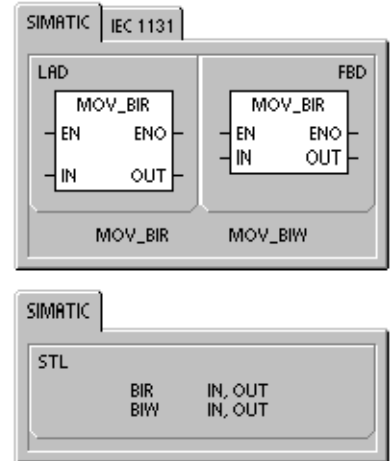


表6-53 移动字节立即读指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE	IB、*VD、*LD、*AC
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC

表6-54 移动字节立即写指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
OUT	BYTE	QB、*VD、*LD、*AC

## 成块移动指令

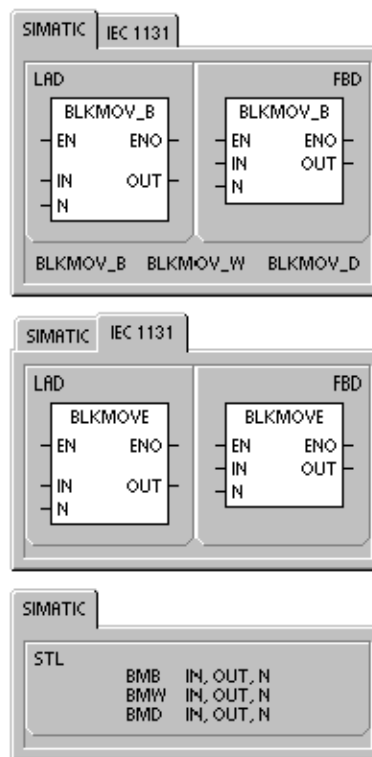
### 成块移动字节、字或双字

“成块移动字节”（BMB）、“成块移动字”（BMW）和“成块移动双字”（BMD）指令移动指定数目的数据到新内存位置，通过移动以输入地址IN开始的指定数目的字节、字或双字N到以输出地址OUT开始的新块。

N的范围是1到255。

设置ENO = 0的错误条件

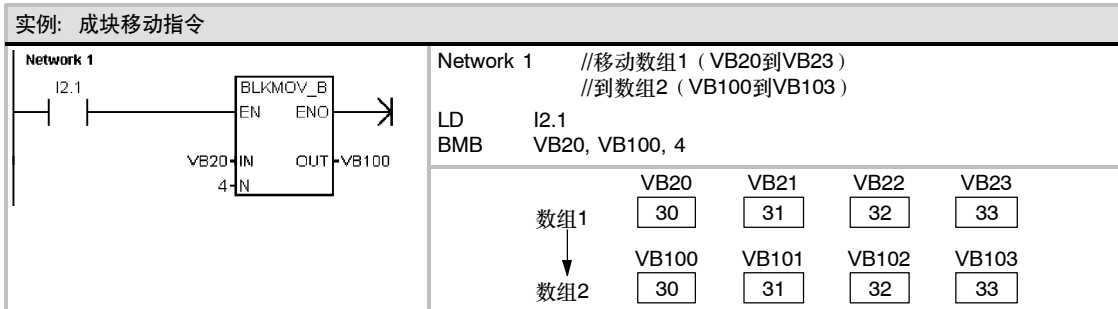
- 0006（间接地址）
- 0091（操作数超出范围）



6

表6-55 成块移动指令的有效操作数

输入 / 输出	数据类型	操作数
IN	BYTE WORD、INT DWORD、DINT	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、*VD、*LD、*AC ID、QD、VD、MD、SMD、SD、LD、*VD、*LD、*AC
OUT	BYTE WORD、INT DWORD、DINT	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC IW、QW、VW、MW、SMW、SW、T、C、LW、AQW、*VD、*LD、*AC ID、QD、VD、MD、SMD、SD、LD、*VD、*LD、*AC
N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、常量、*VD、*LD、*AC



## 程序控制指令

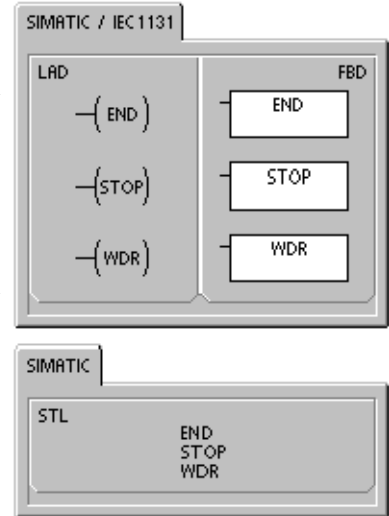
### 有条件结束

“有条件结束”指令（END）基于前面的逻辑条件终止当前的扫描。可以在主程序中使用“有条件结束”指令，但是不能将它用在子例行程序或中断例行程序中。

### 停止

“停止”指令（STOP）通过将S7-200 CPU从RUN（运行）模式转变到STOP（停止）模式终止程序的执行。

如果在中断例行程序中执行“停止”指令，中断例行程序立即终止，并且所有未决的中断被忽略。在当前扫描的结束时，在当前扫描循环中的保留动作都完成，包含执行主用户程序，并且进行从RUN（运行）到STOP（停止）模式的转变。



6

### 监视程序重设

“监视程序重设”指令（WDR）重触发S7-200 CPU的系统监视程序计时器以扩展扫描允许进行的时间而没有监视程序出错。

应仔细使用监视程序重设指令。如果使用循环指令以防止扫描完成或过分地延迟扫描的完成，下列进程禁止直到扫描循环完成：

- 通讯（除了自由端口模式）
- I/O更新（除了立即I/O）
- 强制更新
- SM位更新（SM0、SM5到SM29不更新）
- 运行时间诊断
- 10毫秒和100毫秒计时器对于超出25秒的扫描将不会完全积累时间。
- “停止”指令，当用于中断例行程序时
- 具有离散输出的扩充模块也包含监视程序计时器，如果模块不由S7-200写，它将输出断开。使用立即写入至具有离散输出的每个扩充模块，在扩展扫描时间期间以保持正确输出接通。参考此描述后的实例。



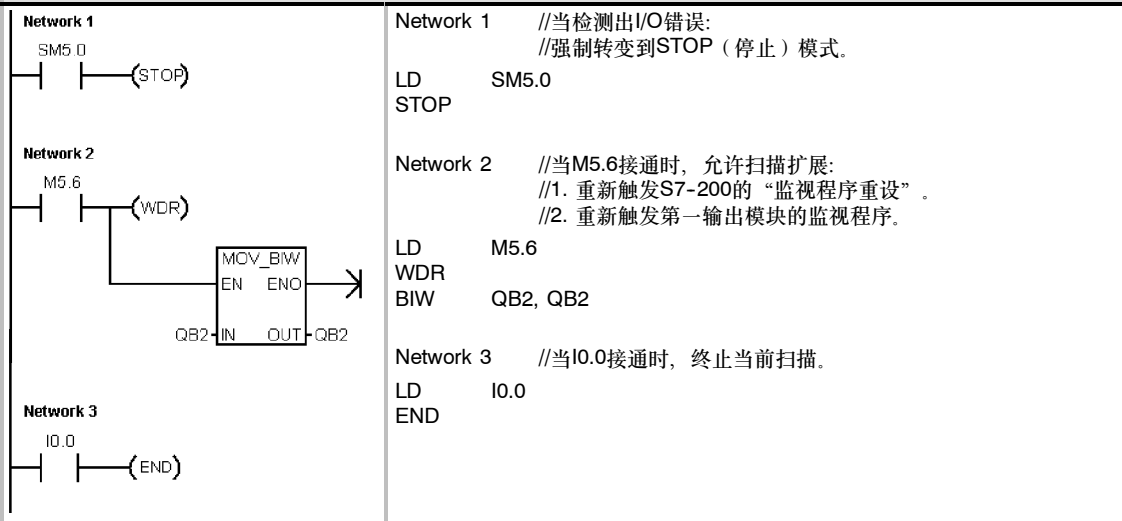
#### 提示

如果预期扫描时间超出500毫秒，或如果预期中断脉冲行动可能防止返回到主扫描大于500毫秒以上，应使用“监视程序重设”指令重新触发监视程序计时器。

每次使用监视程序重设指令时，也应该使用立即写入至在每个离散扩充模块中的一个输出字节（QB），以重新设定每个扩充模块监视程序。

如果使用“监视程序重设”指令允许执行需要长扫描时间的程序，改变模式开关到停止位置引起S7-200在1.4秒中转变到STOP（停止）模式。

## 实例：停止、结束和监视程序重设指令



## For-Next循环指令

使用For (FOR) 和Next (NEXT) 指令来描绘重复指定计数的循环。每个For指令需要一个Next指令。可以最多嵌套八个For-Next循环（将For-Next循环放在另一个For-Next循环中）。

For指令执行For和Next指令之间的指令。指定索引数值或当前循环计数INDX、起始值INIT和结束数值FINAL。

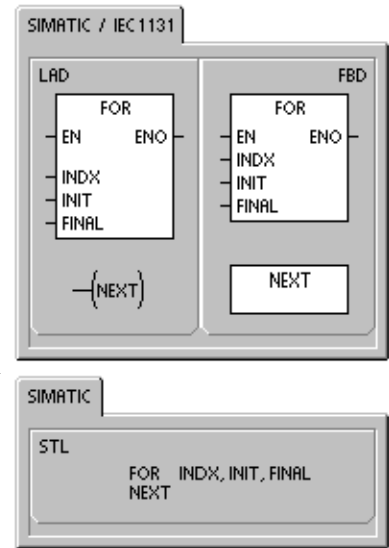
Next 指令标记FOR循环的结束。

设置ENO = 0的错误条件

- 0006 (间接地址)

如果启用For-Next循环，它继续循环过程直到它完成迭代操作，除非从循环自身内部改变最后数值。可以当For-Next循环在循环过程中时改变数值。当循环再次启用时，它将初始值复制到索引数值（当前环路号码）。

For-Next指令在下一启用时自己重新设定。



例如，给定INIT值为1和FINAL值为10，随着INDX数值增加，在For指令和Next指令之间的指令被执行10次：1、2、3、...10。

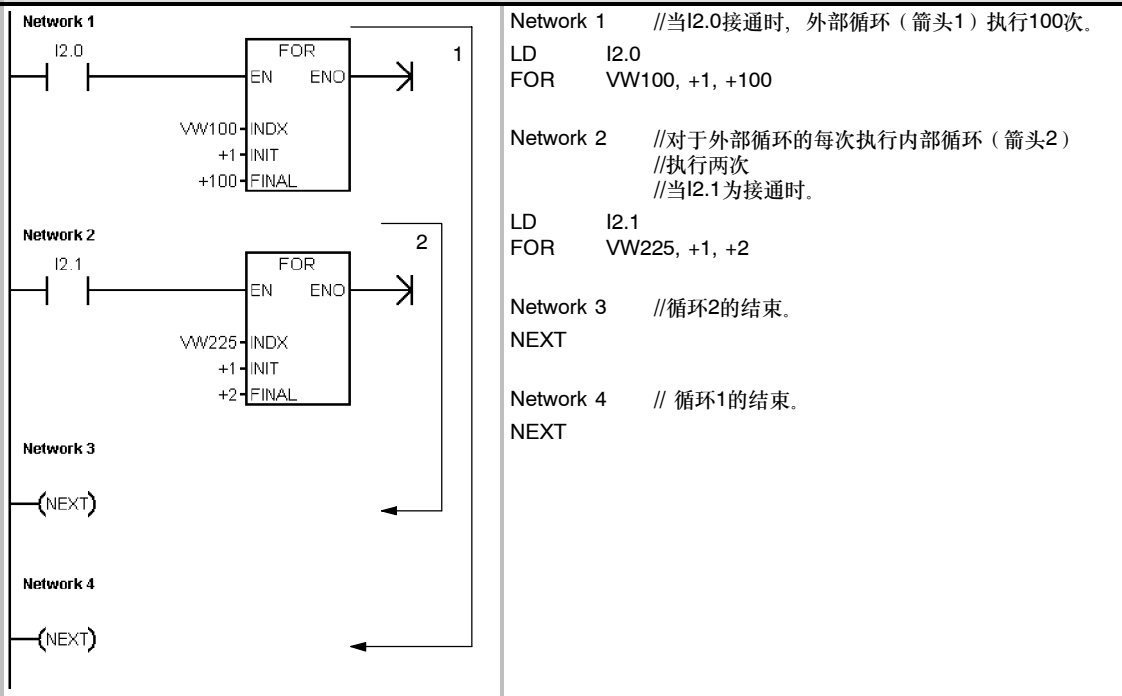
如果起始值大于最后数值，循环不执行。在每次执行For指令和Next指令之间的指令后，INDX数值增加，结果与最终数值比较。如果INDX大于最终数值，循环终止。

当程序进入For-Next循环时，如果堆栈顶部为1，那么当程序退出For-Next循环时，堆栈顶部将为1。

表6-56 For和Next指令的有效操作数

输入 / 输出	数据类型	操作数
INDX	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、AC、*VD、*LD、*AC
INIT、FINAL	INT	VW、IW、QW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量

## 实例: For-Next循环指令





## 跳转指令

“跳转至标签”指令（JMP）完成跳转到程序中指定的标签N。

“标签”指令（LBL）标记跳转目标地址N的位置。

可以在主程序、子程序或中断例行程序中使用“跳转”指令。“跳转”及其相应的“标签”指令必须始终位于同样的代码段中（或者是主程、子程序或者是中断例行程序）。

不能从主程序跳转到子程序或中断例行程序中的标签中。同样，不能从子程序或中断例行程序跳转到子程序或中断例行程序外的标签。

可以在SCR段中使用“跳转”指令，但是相应的“标签”指令必须位于同样的SCR段中。

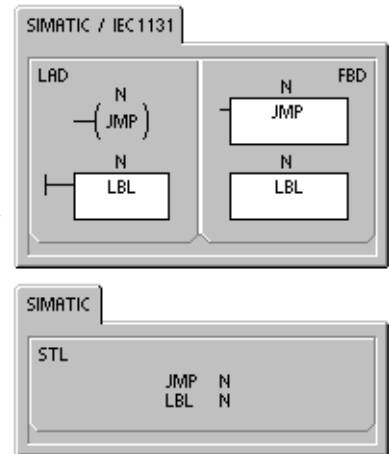


表6-57 跳转指令的有效操作数

输入 / 输出	数据类型	操作数
N	WORD	常量（0到255）

### 实例：跳转至标签指令

<p><b>Network 1</b></p> <p>SM0.2      4</p> <p>  /   — (JMP)</p> <p><b>Network 2</b></p> <p>4</p> <p>  — [ LBL ]</p>	<p>Network 1    //如果保留性数据没有丢失，“跳转”到LBL4</p> <p>LDN    SM0.2</p> <p>JMP    4</p> <p>Network 2</p> <p>LBL    4</p>
---	--

## 顺序控制继电器（SCR）指令

SCR指令提供简单但强大的，自然地适合梯形图、功能块图表或语句表程序的状态控制编程技术。

无论何时应用程序包括一系列必须重复完成操作，SCR可以用于构造用户程序以使它直接符合应用程序。作为结果，可以更快和更容易地编程和调试应用程序。

“载入SCR”指令（LSCR）用指令N引用的S位的值载入SCR和逻辑堆栈。

SCR段由SCR堆栈的结果数值通电或断电。SCR堆栈的数值被复制到逻辑堆栈的顶部，所以方框或输出线圈可以直接连接到左电源横杆而不要干涉触点。

### 限制

当使用SCR时，要知道下列限制：

- 不能在一个以上例行程序中使用同样的S位。例如，如果在主程序中使用S0.1，不要在子程序中使用它。
- 不能跳转入或跳转出SCR段；然而，可以使用“跳转”和“标签”指令在SCR段附近跳转或在SCR段中跳转。
- 不能在SCR段中使用“END”指令。

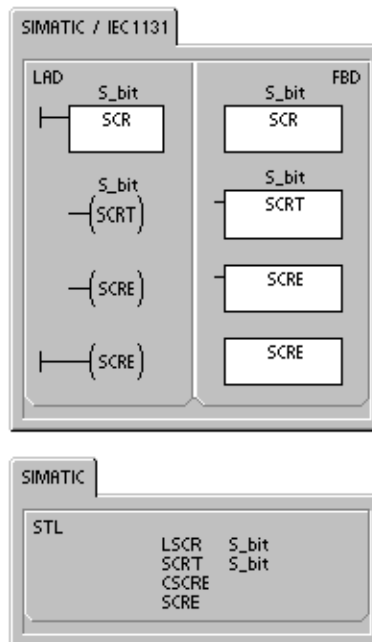


表6-58 顺序控制继电器指令的有效操作数

输入 / 输出	数据类型	操作数
S_bit	BOOL	S

图6-32显示S堆栈和逻辑堆栈和执行“载入SCR”指令的效果。下列描述对“顺序控制继电器”指令是正确的：

- “载入SCR”指令（LSCR）标记SCR段的开始，“SCR结束”指令（SCRE）标记SCR段的结束。“载入SCR”和“SCR结束”指令之间的所有逻辑取决于其执行后S堆栈的值。“SCR结束”和下一个“载入SCR”指令之间的逻辑不取决于S堆栈的数值。
- “SCR转变”指令（SCRT）提供将控制从现用的SCR段传送到另一个SCR段的方法。

当它有功率流时，执行“SCR转变”指令将重新设定当前现用的段的S位，并且将设置引用段的S位。在“SCR转变”指令执行时，重新设定现用程序段的S位不影响S堆栈。因此，SCR段保持通电直到它退出。

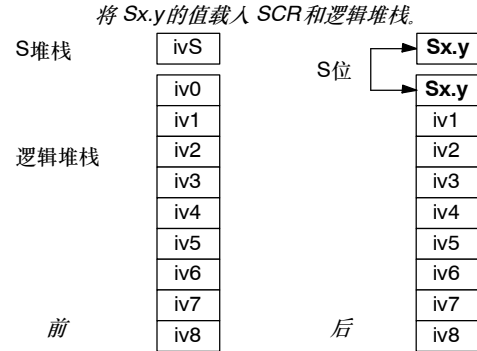
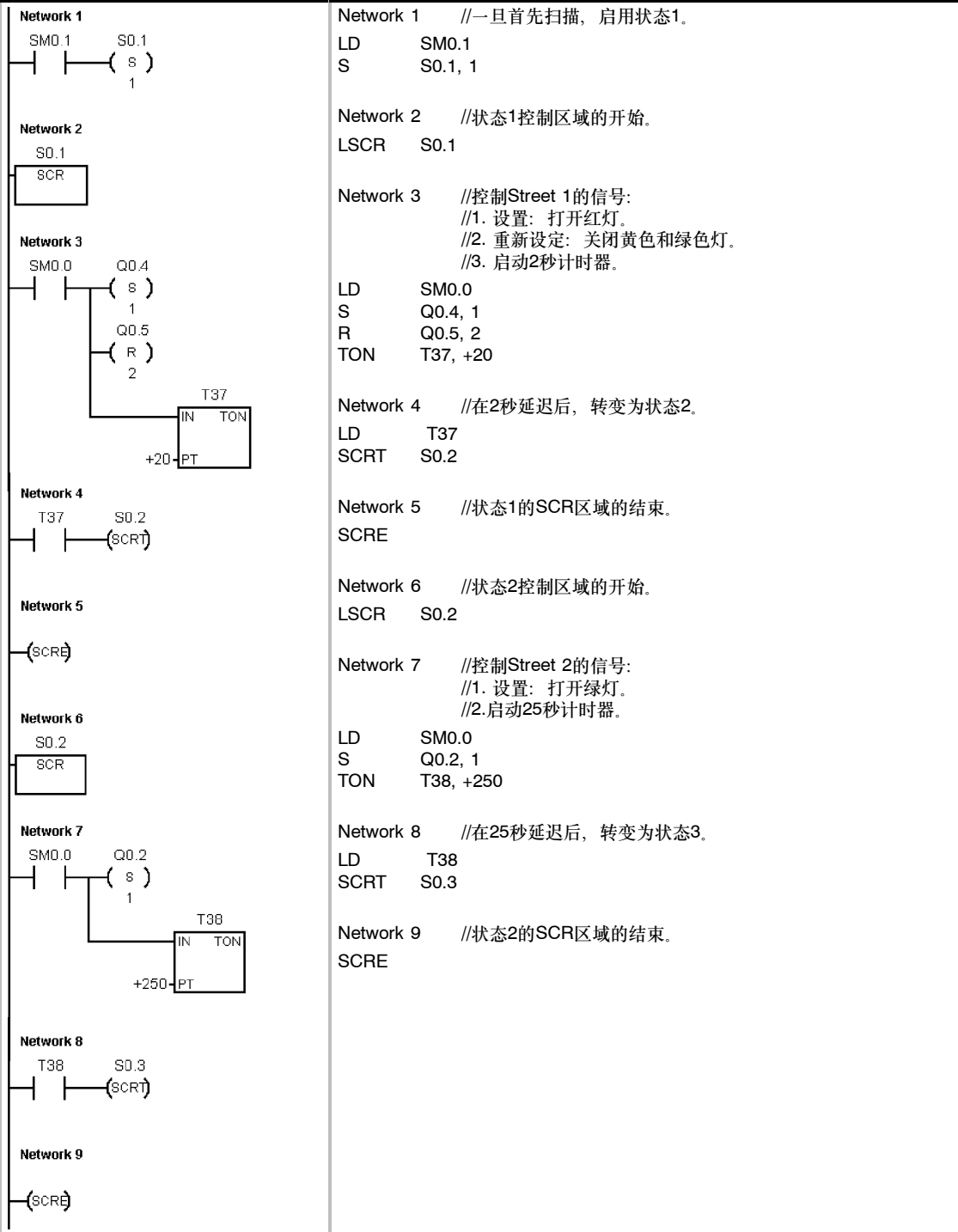


图6-32 逻辑堆栈上的LSCR结果

- “有条件SCR结束”指令（CSCRE）提供一种方法以退出现用SCR段而不用执行“有条件SCR结束”和“SCR结束”指令之间的指令。“有条件SCR结束”指令不影响任何S位，也不影响S堆栈。

在下列实例中，首次扫描位SM0.1设置S0.1，它在首次扫描时将成为激活状态1。在2秒延迟后，T37引起转变到状态2。此转变取消激活状态1 SCR（S0.1）段，并激活状态2 SCR（S0.2）段。

## 实例：顺序控制继电器指令





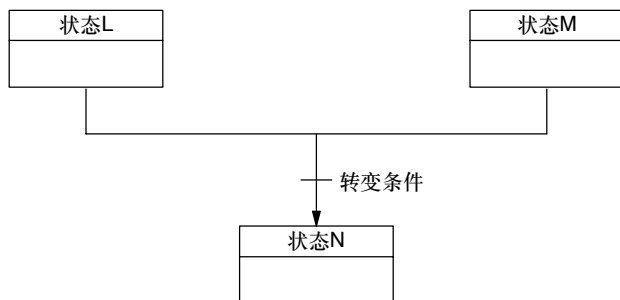


图6-34 控制流的汇合

6

**实例：控制流的汇合**

<p><b>Network 1</b> S3.4 SCR</p> <p><b>Network 2</b> V100.5      S3.5     (SCRT)</p> <p><b>Network 3</b> (SCRE)</p> <p><b>Network 4</b> S6.4 SCR</p> <p><b>Network 5</b> C50      S6.5     (SCRT)</p> <p><b>Network 6</b> (SCRE)</p> <p><b>Network 7</b> S3.5      S6.5      S5.0       ( S )       1       S3.5       ( R )       1       S6.5       ( R )       1</p>	<p>Network 1 //状态L控制区域的开始 LSCR S3.4</p> <p>Network 2 //转变到状态L' LD V100.5 SCRT S3.5</p> <p>Network 3 //状态L的SCR区域的结束 SCRE</p> <p>Network 4 //状态M控制区域的开始 LSCR S6.4</p> <p>Network 5 //转变到状态M' LD C50 SCRT S6.5</p> <p>Network 6 //状态M的SCR区域的结束 SCRE</p> <p>Network 7 //当状态L'和状态M'都激活时: //1. 启用状态N (S5.0) //2. 重新设定状态L' (S3.5) //3. 重新设定状态M' (S6.5)</p> <p>LD S3.5 A S6.5 S S5.0, 1 R S3.5, 1 R S6.5, 1</p>
---	---

在其它状况下，控制流可能指向几个可能的控制流之一，取决于哪一个转变条件首先变为真。这样的状况在图6-35中描述，该图显示等同SCR程序。

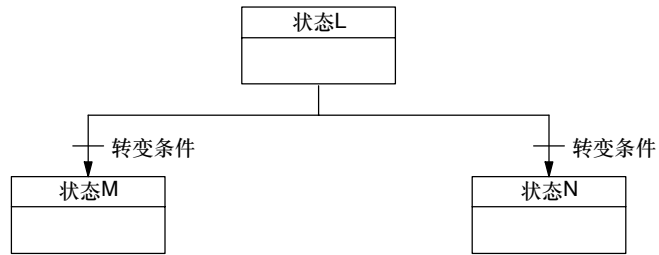
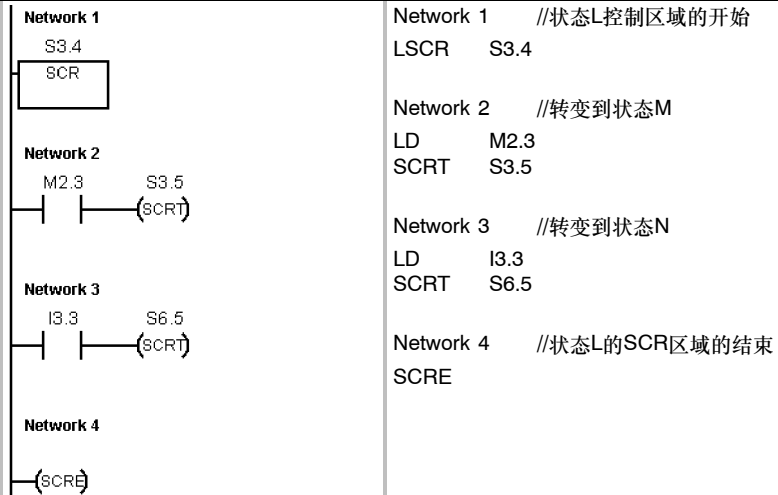


图6-35 控制流的扩散，根据转变条件

实例：有条件跳转



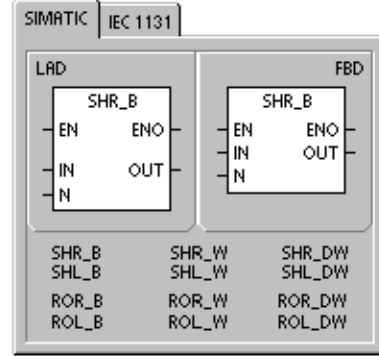
## 移位与循环指令

### 向右移位和向左移位指令

移位指令将输入数值IN向左或向右移位移位计数N个位，并将结果载入输出OUT。

当每个位都被移出，移位指令将用零填补每个位。如果移位计数(N)大于或等于允许的最大值(字节操作为8、字操作为16、双字操作为32)，数值移位操作的最大次数。如果移位计数大于0，溢出内存位(SM1.1)采用最后移出位的数值。如果移位操作的结果是零，零内存位(SM1.0)被设置。

字节操作是无符号的。对于字和双字操作，当使用有符号数据类型时符号位被移位。

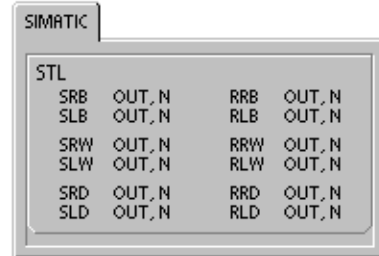
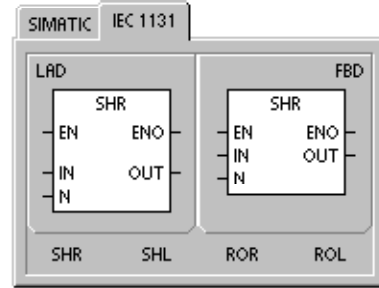


设置ENO = 0的错误条件

- 0006 (间接地址)

受影响的SM位:

- SM1.0 (零)
- SM1.1 (溢出)



### 向右循环和向左循环指令

“循环”指令将输入数值(IN)向右或向左循环移位计数(N)个位，并将结果载入内存位置(OUT)。循环是轮循的。

如果移位计数大于或等于运算的最大值(字节操作为8，字操作为16，双字操作为32)，在执行循环前，S7-200在移位计数上完成模操作以获得有效的移位计数。此结果经字节操作是0到7的移位计数，对字操作是0到15，对于双字操作是0到31。

如果移位计数是0，循环操作不进行。如果循环操作完成，最后循环位的数值复制到溢出位(SM1.1)。

如果移位计数不是8的整数倍(对于字节操作)、16的整数倍(对于字操作)或者32的整数倍(对于双字操作)，最后循环出的位被复制到溢出内存位(SM1.1)。当循环的数值为零时，零内存位(SM1.0)被设置。

字节操作是无符号的。对于字和双字操作，当使用有符号数据类型时，符号位被移位。

设置ENO = 0的错误条件

- 0006 (间接地址)

受影响的SM位:

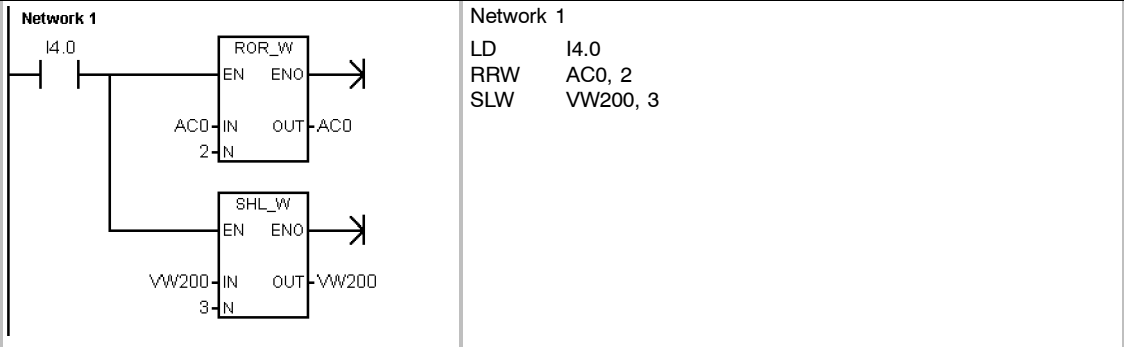
- SM1.0 (零)
- SM1.1 (溢出)

表6-59 移位和循环指令的有效操作数

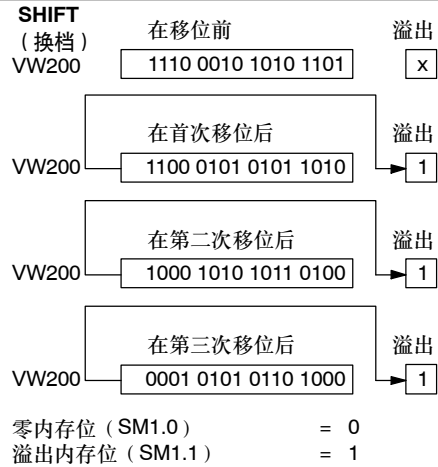
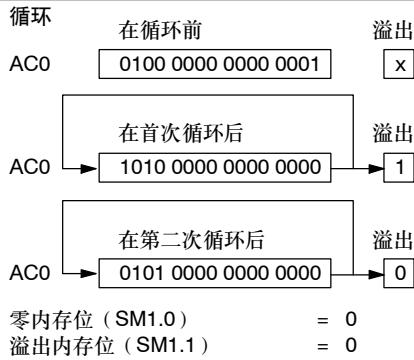
输入 / 输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常量
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC
	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、AC、*VD、*LD、*AC
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC
N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量



实例: 移位与循环指令



Network 1  
 LD I4.0  
 RORW AC0, 2  
 SLW VW200, 3



## 移位寄存器位指令

“移位寄存器位”指令将数值移入“移位寄存器”。此指令提供用于排序和控制产品流或数据的容易方法。使用此指令移位整个寄存器一位，每个扫描一次。

“移位寄存器位”指令将数据的数值移入“移位寄存器”。S\_BIT指定移位寄存器的最低位。N指定移位寄存器的长度和移位的方向（移位加 = N，移位减 = -N）。

每个由SHRB指令移出的位放入溢出内存位（SM1.1）。

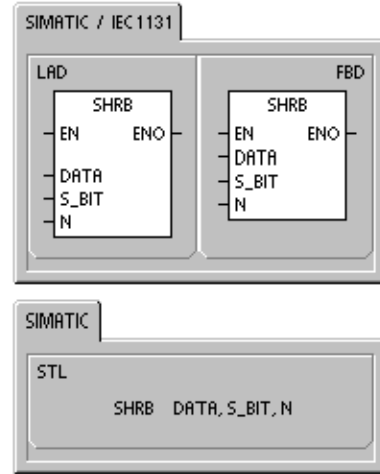
此指令由最低位（S\_BIT）和由长度（N）指定的位数定义。

设置ENO = 0的错误条件

- 0006（间接地址）
- 0091（操作数超出范围）
- 0092（计数域出错）

受影响的SM位:

- SM1.1（溢出）



6

表6-60 移位寄存器位指令的有效操作数

输入 / 输出	数据类型	操作数
DATA、S_Bit	BOOL	I、Q、V、M、SM、S、T、C、L
N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量

使用下列公式计算移位寄存器（MSB.b）最高位的地址:

$$MSB.b = [(S\_BIT \text{ 的字节}) + ((N) - 1 + (S\_BIT \text{ 的位}) / 8)] \text{ [除以 8 的余数]}$$

例如: 如果S\_BIT是V33.4, N是14, 下列计算显示MSB.b是V35.1。

$$\begin{aligned} MSB.b &= V33 + ((14) - 1 + 4) / 8 \\ &= V33 + 17 / 8 \\ &= V33 + 2 \text{ 带余数 } 1 \\ &= V35.1 \end{aligned}$$

在移位减时，由负的长度（N）数值指示，输入数据移入移位寄存器的最高位，并移出最低位（S\_BIT）。移出的数据然后放置在溢出内存位（SM1.1）。

在移位加时，由正的长度（N）数值指示，输入数据（数据）移入由S\_BIT指定的移位寄存器的最低位，并移出移位寄存器的最高位。移出的数据然后放入溢出内存位（SM1.1）。

移位寄存器的最大长度是64位正数或负数。图6-36显示正和负的N值的移位。

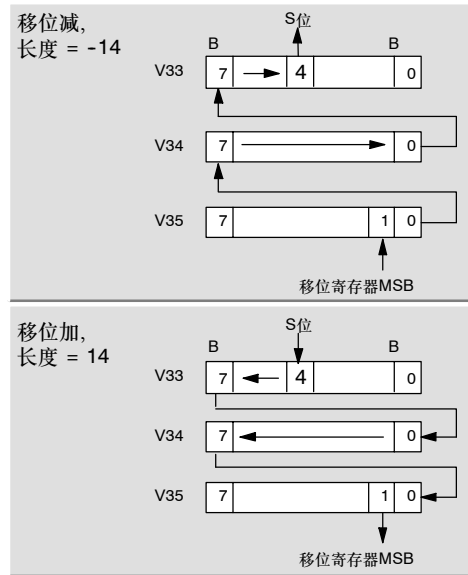
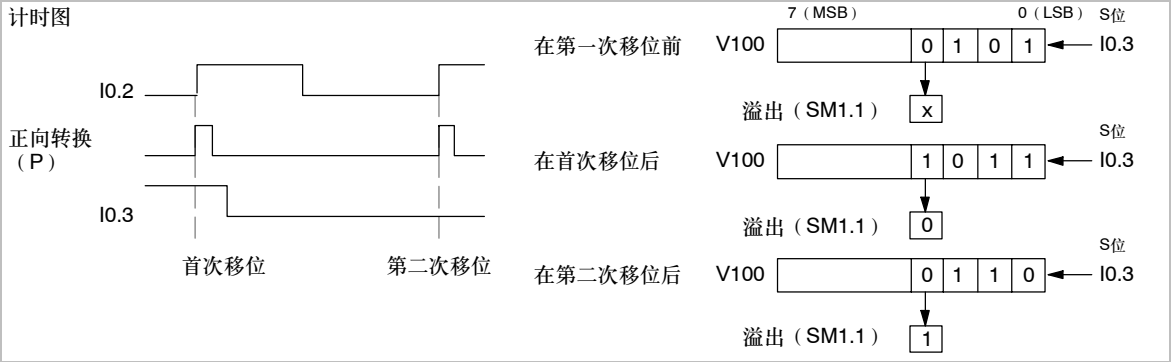
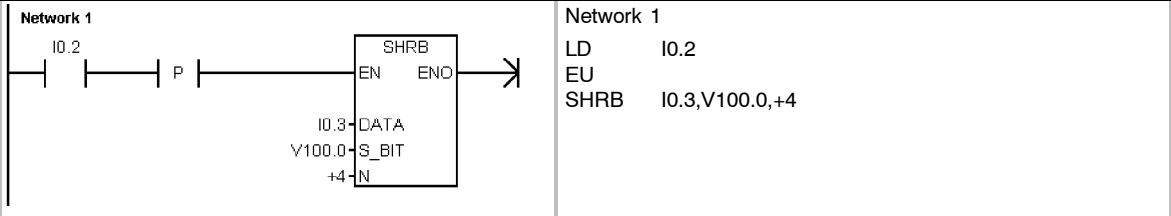


图6-36 移位寄存器进入和退出

实例: 移位寄存器位指令

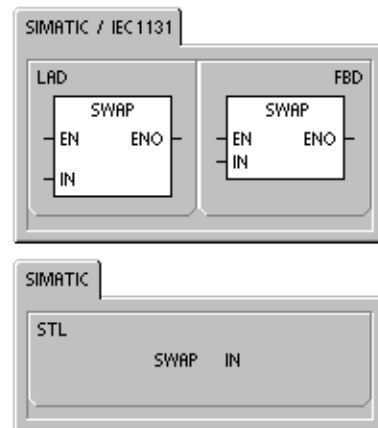


## 交换字节指令

交换字节指令用字IN的最低字节交换最高有效字节。

设置ENO=0的错误条件

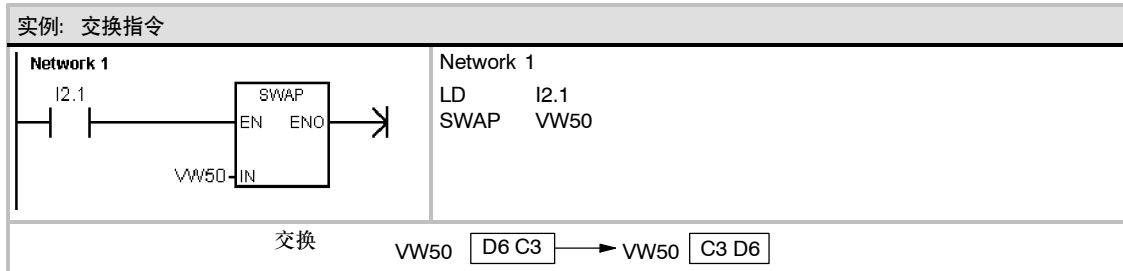
- 0006 (间接地址)



# 6

表6-61 交换字节指令的有效操作数

输入 / 输出	数据类型	操作数
IN	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、AC、*VD、*LD、*AC



## 字符串指令

### 字符串长度

“字符串长度指令”（SLEN）返回IN指定的字符串的长度。

### 复制字符串

“复制字符串”指令（SCPY）复制由IN指定的字符串到由OUT指定的字符串。

### 并置字符串

“并置字符串”指令（SCAT）将由IN指定的字符串附加到由OUT指定的字符串的结尾。

### SM位和ENO

对于“字符串长度”、“复制字符串”和“并置字符串”指令，下列条件影响ENO。

设置ENO=0的错误条件

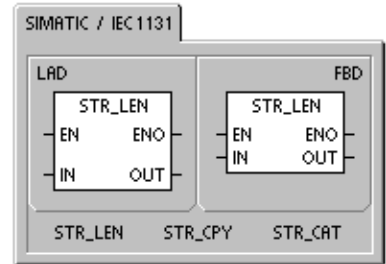
- 0006（间接地址）
- 0091（范围出错）

表6-62 字符串长度指令的有效操作数

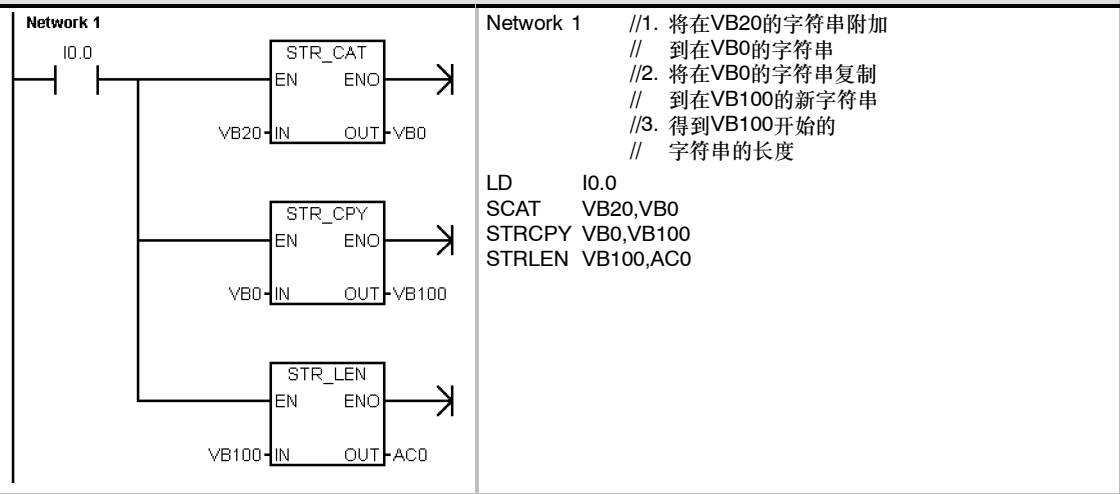
输入/输出	数据类型	操作数
IN	BYTE（字符串）	VB、LB、*VD、*LD、*AC
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC

表6-63 复制字符串和并置字符串指令的有效操作数

输入/输出	数据类型	操作数
IN、OUT	BYTE（字符串）	VB、LB、*VD、*LD、*AC



实例：并置字符串、复制字符串和字符串长度指令



```

Network 1 //1. 将在VB20的字符串附加
           // 到在VB0的字符串
           //2. 将在VB0的字符串复制
           // 到在VB100的新字符串
           //3. 得到VB100开始的
           // 字符串的长度

LD I0.0
SCAT VB20,VB0
STRCPY VB0,VB100
STRLEN VB100,ACO
    
```

在执行程序前

VB0						VB6					
6	'H'	'e'	'L'	'L'	'O'	' '					

VB20						VB25					
5	'W'	'O'	'R'	'L'	'D'						

在执行程序后

VB0											VB11	
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'	

VB100											VB111	
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'	

ACO	
11	

## 从字符串复制子字符串

“从字符串复制子字符串”指令（SSCPY）从以索引INDX开始的由IN指定的字符串将指定数目的字符N复制到由OUT指定的新字符串。

设置ENO=0的错误条件

- 0006（间接地址）
- 0091（范围出错）
- 009B（索引=0）

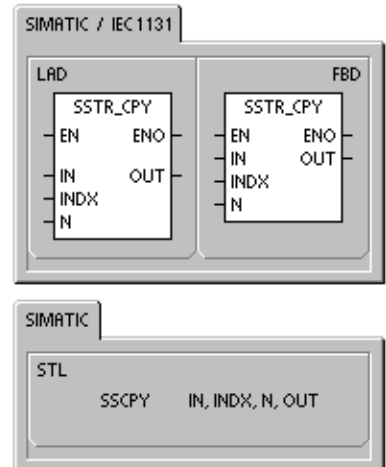
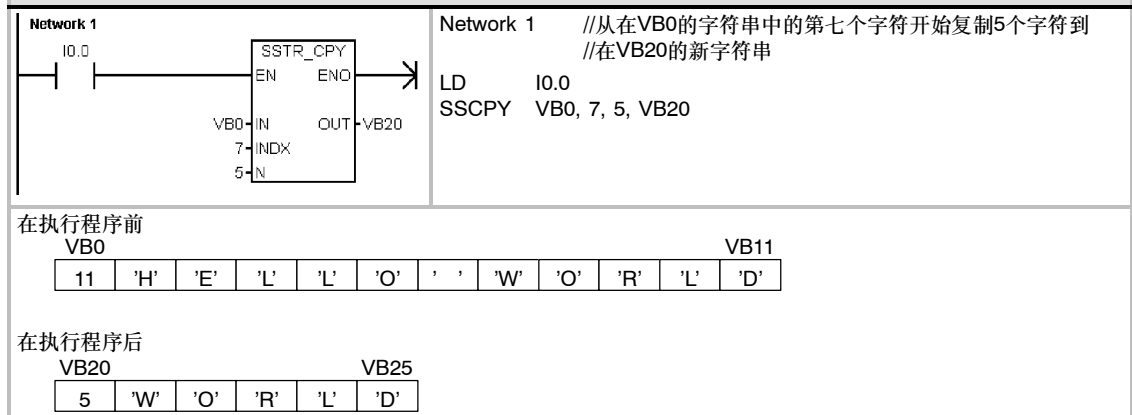


表6-64 从字符串复制子字符串指令的有效操作数

输入 / 输出	数据类型	操作数
IN、OUT	BYTE（字符串）	VB、LB、*VD、*LD、*AC
INDX、N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量

### 实例：复制子字符串指令



## 在字符串中查找字符串

“在字符串中查找字符串”指令（SFND）在字符串IN1中搜索字符串IN2的第一次出现。搜索从由OUT指定的起始位置开始。如果一序列字符被发现完全匹配字符串IN2，字符串序列中的第一个字符的位置写入OUT。如果字符串IN2没有在字符串IN1中找到，指令OUT设置为0。

设置ENO=0的错误条件

- 0006（间接地址）
- 0091（范围出错）
- 009B（索引=0）

## 在字符串中查找第一个字符

“在字符串中查找第一个字符”指令（CFND）在字符串IN1中搜索第一个出现的来自字符串IN2中描述的字符集的任何字符。搜索从起始位置OUT开始。如果匹配的字符找到，该字符的位置写入OUT。如果没有找到匹配的字符，OUT设置为0。

设置ENO=0的错误条件

- 0006（间接地址）
- 0091（范围出错）
- 009B（索引=0）

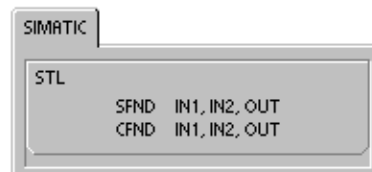
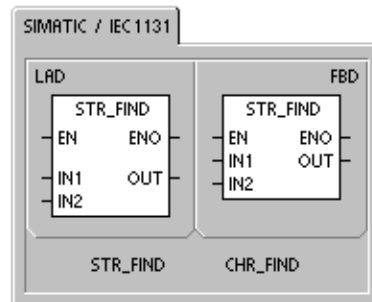


表6-65 在字符串中查找字符串和在字符串中查找第一个字符的指令的有效操作数

输入 / 输出	数据类型	操作数
IN1、IN2	BYTE（字符串）	VB、LB、*VD、*LD、*AC
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC



实例：在字符串中查找字符串指令

下列实例使用存储在VB0的字符串作为命令用于打开或关断泵。字符串“On”存储在VB20，字符串“Off”存储在VB30。“在字符串中查找字符串”指令的结果存储在AC0（OUT参数）。如果结果不是0，那么字符串“On”在命令字符串（VB12）中发现。

**Network 1**

**Network 1** //1. 设置AC0为1。（AC0用作OUT参数。）  
 //2. 从第一位置（AC0=1）开始，在VB0的字符串中查找  
 // 在VB20（“On”）的字符串。

```
LD I0.0
MOV B 1, AC0
SFND VB0, VB20, AC0
```

VB0	12	'T'	'u'	'r'	'n'	' '	'P'	'u'	'm'	'p'	' '	'O'	'n'	VB12
VB20	2	'O'	'n'											
VB22														
VB30	3	'O'	'f'	'f'										
VB33														

如果在VB20中的字符串被发现:

AC0

11

如果在VB20中的字符串没有被发现:

AC0

0

实例：在字符串中查找字符指令

在下列实例中，存储在VB0中的字符串包含温度。在VB20的字符串存储所有数字字符（以及+和-），这些数字字符可以识别在字符串中的温度。采样程序在该字符串中找到一个数字的起始位置，然后将这个数字字符转换为实数数字。VD200存储温度的实数数值。

**Network 1**

**Network 1** //1. 设置AC0为1。（AC0用作OUT参数并指向字符串的  
 // 第一位。）  
 //2. 在VB0的字符串中查找数字字符。  
 //3. 将字符串转换为实数。

```
LD I0.0
MOV B 1, AC0
CFND VB0, VB20, AC0
STR VB0, AC0, VD200
```

VB0	11	'T'	'e'	'm'	'p'	' '	' '	'g'	'8'	'.'	'6'	'F'	VB11	
VB20	12	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'0'	'+'	'-'	VB32

存储在VB0的温度的起始位置:

AC0

7

温度的实数值:

VD200

98.6

# 表格指令

## 添加到表格

“添加到表格”指令将数值（数据）添加到表格中（TBL）。表格的第一个数值是最大表格长度（TL）。第二个数值是输入计数（EC），它指定表格中输入的数目。新数据添加到表格中最后一个条目的后面。每次新数据添加到表格，输入计数增加。

一个表格可以有至多100个数据条目。

设置ENO=0的错误条件

- SM1.4（表格溢出）
- 0006（间接地址）
- 0091（操作数超出范围）

受影响的SM位:

- 如果试图把表格装得溢出，SM1.4设置为1。

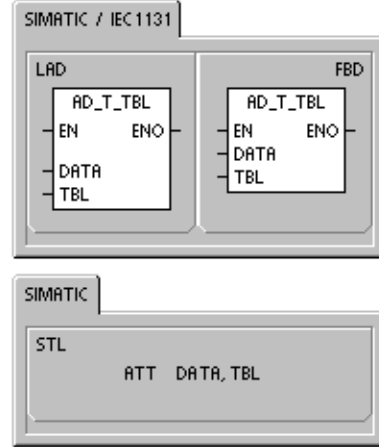


表6-66 表格指令的有效操作数

输入 / 输出	数据类型	操作数
数据	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
TBL	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、*VD、*LD、*AC

**实例：添加到表格指令**

**Network 1**

**Network 2**

**Network 1 //载入最大表格长度**

```
LD SM0.1
MOVW +6,VW200
```

**Network 2**

```
LD IO.0
ATT VW100,VW200
```

**在执行ATT之前**

VW100	1234
VW200	0006
VW202	0002
VW204	5431
VW206	8942
VW208	xxxx
VW210	xxxx
VW212	xxxx
VW214	xxxx

TL (条目的最大编号)  
EC (条目计数)  
d0 (数据0)  
d1 (数据1)

**在执行ATT之后**

VW200	0006
VW202	0003
VW204	5431
VW206	8942
VW208	1234
VW210	xxxx
VW212	xxxx
VW214	xxxx

TL (条目的最大编号)  
EC (条目计数)  
d0 (数据0)  
d1 (数据1)  
d2 (数据2)

## 先入先出和后入先出

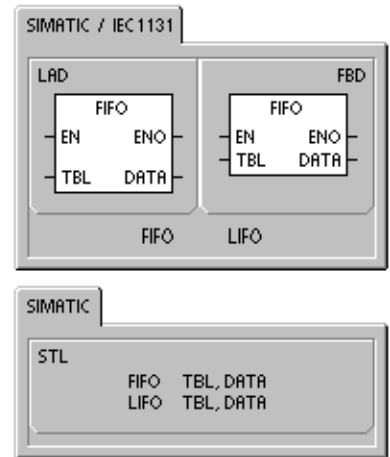
表格可以有至多100个数据条目。

### 先入先出

“先入先出”指令（FIFO）通过删除表格中第一个条目（TBL）将表格中最早的（或第一个）条目移动到输出内存地址，并且将该数值移动到数据指定的位置。表格的所有其它条目向上移位一个位置。指令每次执行，表格的条目计数减少。

### 后入先出

“后入先出”指令（LIFO）通过删除表格中最后一个条目（TBL）将表格中最新的（或最后一个）条目移动到输出内存地址，并且将该数值移动到数据指定的位置。每次执行指令，表格中的条目计数减少。



设置ENO=0的错误条件

- SM1.5（空表格）
- 0006（间接地址）
- 0091（操作数超出范围）

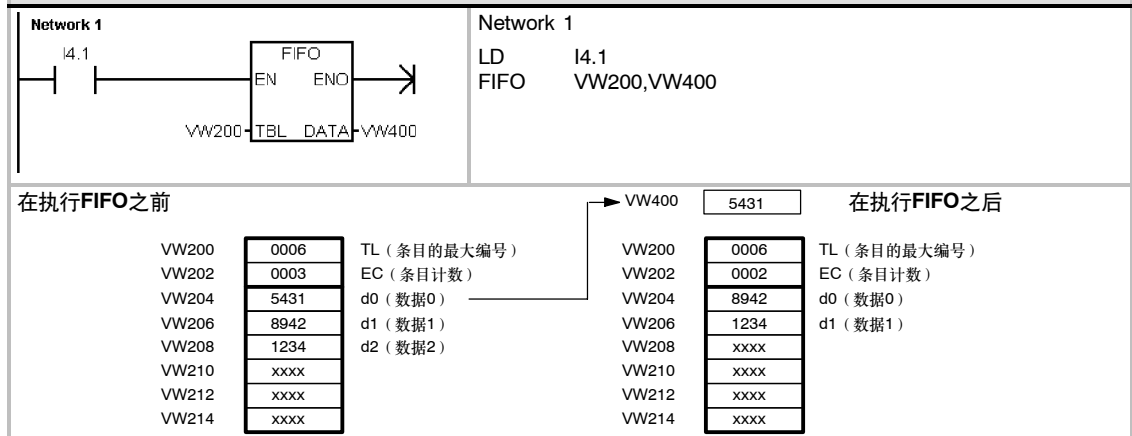
受影响的SM位:

- 如果试图从空表格中删除条目，SM1.5设置为1。

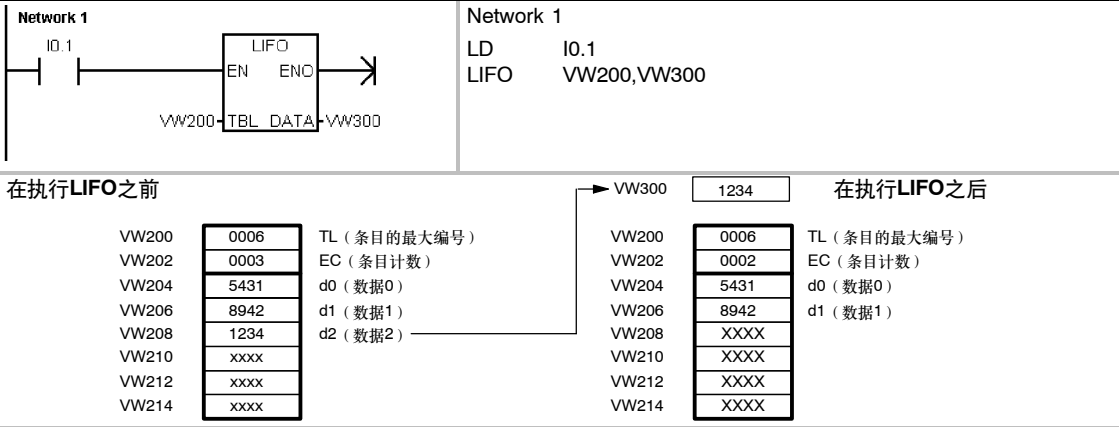
表6-67 先入先出和后入先出指令的有效操作数

输入 / 输出	数据类型	操作数
TBL	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、*VD、*LD、*AC
数据	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC

### 实例: 先入先出指令



实例: 后入先出指令



## 内存填充

内存填充指令（FILL）将在地址IN中包含的字数值写到以地址OUT开始的N个连续的字。

N的范围是1到255。

设置ENO=0的错误条件

- 0006（间接地址）
- 0091（操作数超出范围）

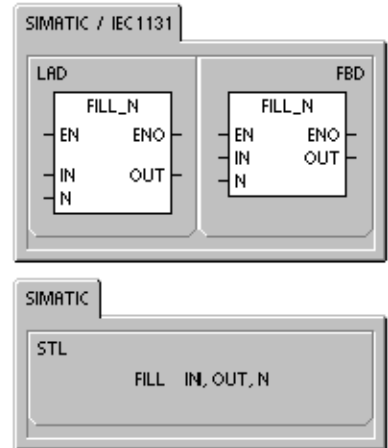
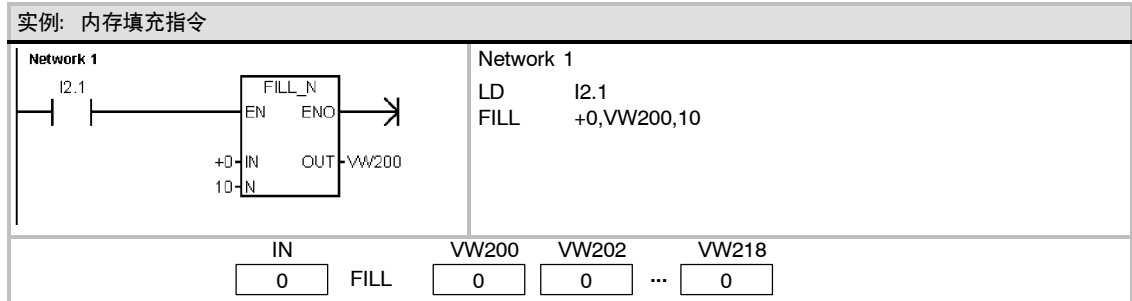


表6-68 内存填充指令的有效操作数

输入 / 输出	数据类型	操作数
IN	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常量
OUT	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AQW、*VD、*LD、*AC



## 表格查找

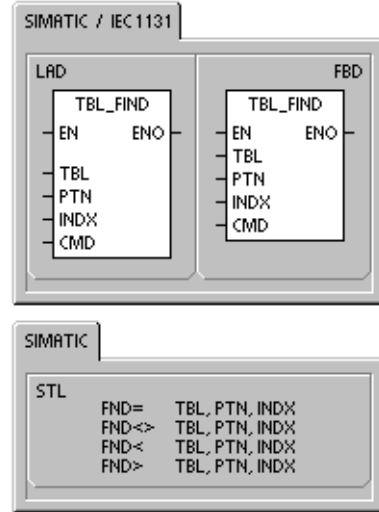
“表格查找”指令（FND）搜索表格以找到匹配某种标准的数据。“表格查找”指令搜索表格TBL，以表格条目INDX开始，查找匹配由CMD指定的搜索标准的数据数值或模式PTN。命令参数CMD给定一个1到4的数字值，分别相当于=、<>、<和>。

如果匹配找到，INDX指向表格中匹配的条目。要发现下一个匹配条目，在再次调用“表格查找”指令前INDX必须增加。如果匹配没有找到，INDX具有等于条目计数的数值。

表格可以有至多100个数据条目。数据条目（搜索的区域）从0计数到最大数值99。

设置ENO=0的错误条件

- 0006（间接地址）
- 0091（操作数超出范围）



6

表6-69 表格查找指令的有效操作数

输入 / 输出	数据类型	操作数
TBL	WORD	IW、QW、VW、MW、SMW、T、C、LW、*VD、*LD、*AC
PTN	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量
INDX	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、AC、*VD、*LD、*AC
CMD	BYTE	(常量) 1: 等于 (=), 2: 不等于 (<>), 3: 小于 (<), 4: 大于 (>)



### 提示

当使用表格查找指令，用添加到表格、后入先出和先入先出指令产生的表格时，条目计数和数据条目直接对应。“添加到表格”、“后入先出”或“先入先出”指令所需的最大条目数对表格查找指令不需要。参见图6-37。

因此，应设置“查找”指令的操作数TBL为一个字地址（两个字节），高于“添加到表格”、“后入先出”或“先入先出”指令的相应TBL操作数。

#### ATT、LIFO和FIFO的表格格式

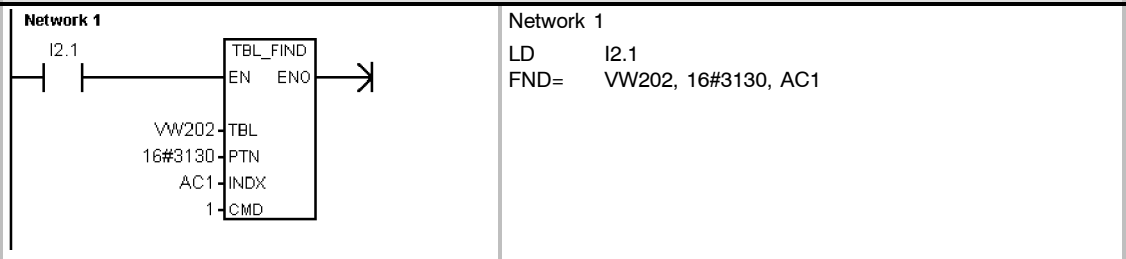
VW200	0006	TL (条目的最大编号)
VW202	0006	EC (条目计数)
VW204	xxxx	d0 (数据0)
VW206	xxxx	d1 (数据1)
VW208	xxxx	d2 (数据2)
VW210	xxxx	d3 (数据3)
VW212	xxxx	d4 (数据4)
VW214	xxxx	d5 (数据5)

#### TBL\_FIND的表格格式

VW202	0006	EC (条目计数)
VW204	xxxx	d0 (数据0)
VW206	xxxx	d1 (数据1)
VW208	xxxx	d2 (数据2)
VW210	xxxx	d3 (数据3)
VW212	xxxx	d4 (数据4)
VW214	xxxx	d5 (数据5)

图6-37 表查找指令和ATT、LIFO和FIFO指令之间的不同表格式

实例: 表格查找指令



当I2.1为接通时，在表格中搜索等于十六进制数3130的数值。

VW202	0006	EC (条目计数)
VW204	3133	d0 (数据0)
VW206	4142	d1 (数据1)
VW208	3130	d2 (数据2)
VW210	3030	d3 (数据3)
VW212	3130	d4 (数据4)
VW214	4541	d5 (数据5)

如果表格使用ATT、LIFO和FIFO指令创建，VW200包含最大允许的条目数，而不为“查找”指令所需。

AC1  AC1必须设置为0以从表格顶部搜索。

执行表格搜索

AC1  AC1包含相应于在表格（d2）中第一次匹配发现的数据条目数目。

AC1  在搜索剩余表格条目前，将INDX递增一。

执行表格搜索

AC1  AC1包含相应于在表格（d4）中第二次匹配发现的数据条目数目。

AC1  在搜索剩余表格条目前，将INDX递增一。

执行表格搜索

AC1  AC1包含等于条目计数的数值。整个表格已搜索过，没有发现另一个匹配。

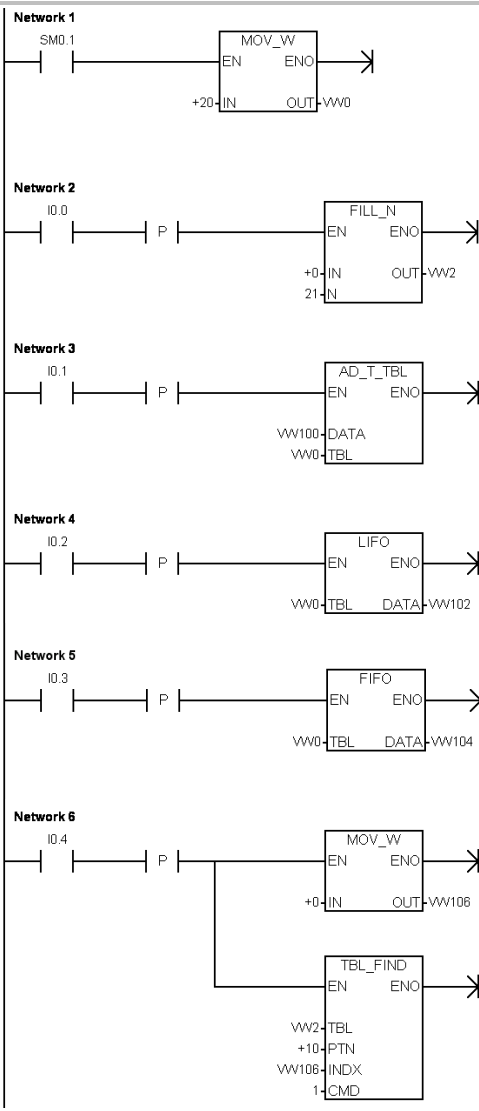
AC1  在表格能再次搜索前，INDX数值必须重新设定为0。

## 实例：创建表格

下列程序创建具有20个条目的表格。表格的第一个内存位置包含表格的长度（在此例中为20个条目）。第二内存位置显示当前表格条目数。其它位置包含条目。表格可以有至多100个条目。它不包含定义表格最大长度或真实条目数（此处为VW0和VW2）的参数。表格中的真实条目数（此处为VW2）随着每个命令自动由CPU增加或减少。

在使用表格前，分配表格条目的最大数目。否则，无法在表格中制作条目。也要确定所有的读和写命令在边沿激活。

要搜索表格，在进行查找前，索引（VW106）必须设置为0。如果匹配查找到，索引将具有表格条目号码，但如果没有找到匹配，索引将匹配表格（VW2）当前条目计数。



Network 1 //创建以内存位置4开始的具有20个条目的表格。  
//1. 在首次扫描，定义表格的最大长度。

```
LD SM0.1
MOVW +20,VW0
```

Network 2 //用输入I0.0重新设定表  
//在I0.0的上升边缘，  
//从VW2用“+0”填充内存位置。

```
LD I0.0
EU
FILL +0,VW2,21
```

Network 3 //用输入I0.1写数值到表格，  
//在I0.1的上升边缘，  
//将内存位置VW100的数值复制到表格。

```
LD I0.1
EU
ATT VW100,VW0
```

Network 4 //用输入I0.2读第一个表格数值  
//将最后一个表格数值移动到位置VW102。  
//这减少了条目数。  
//在I0.2的上升边缘，  
//移动最后一个表格数值到VW102

```
LD I0.2
EU
LIFO VW0,VW102
```

Network 5 //用输入I0.3读最后一个表格数值  
//将第一个表格数值移动到位置VW102。  
//这减少了条目数。  
//在I0.0的上升边缘，  
//移动第一个表格数值到VW104

```
LD I0.3
EU
FIFO VW0,VW104
```

Network 6 //搜索表查找具有数值10的第一个位置。  
//1. 在I0.4的上升边缘，重新设定索引指针。  
//2. 查找等于10的表格条目。

```
LD I0.4
EU
MOVW +0,VW106
FND= VW2,+10,VW106
```



## 计时器指令

### SIMATIC计时器指令

#### 打开延迟计时器保留性打开延迟计时器

“打开延迟计时器”（TON）和“保留性打开延迟计时器”（TONR）指令当允许输入接通时计时。计时器编号（Txx）确定计时器的分辨率。

#### 关闭延迟计时器

“关闭延迟计时器”（TOF）用于在输入断开后，在一段固定的时间内延迟将输出断开。计时器编号（Txx）确定计时器的分辨率。

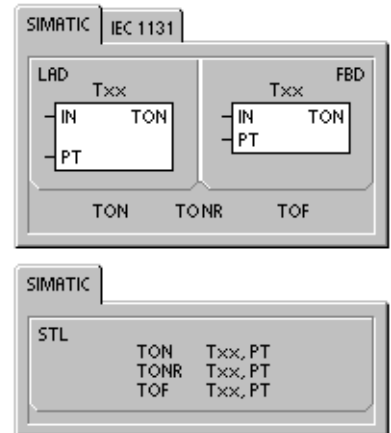


表6-70 SIMATIC计时器指令的有效操作数

输入 / 输出	数据类型	操作数
Txx	WORD	常量 (T0到T255)
IN	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
PT	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*LD、*AC、常量



#### 提示

对于关闭延迟计时器（TOF）和打开延迟计时器（TON），不能共享相同的计时器号（Txx）。例如，不能都有TON T32和TOF T32。

如表6-71所示，三种类型的计时器完成不同类型的计时任务：

- 可以将TON用于计时单个间隔。
- 可以将TONR用于积累许多时间间隔。
- 可以将TOF用于扩展时间经过断开（或假）条件，诸如用于在断开后冷却电机。

表6-71 计时器指令的操作

类型	当前 >= 预置	允许输入（IN）的状态	电源循环 / 第一次扫描
TON	当前继续计数到32,767时的计时器位	ON: 当前值计数时间 OFF: 计时器位断开, 当前值=0	计时器位断开 当前值 = 0
TONR	当前继续计数到32,767时的计时器位	ON: 当前值计数时间 OFF: 计时器位和当前值保持最后状态	计时器位断开 当前值可以保持 <sup>1</sup>
TOF	计时器位断开 当前 = 预置, 停止计数	ON: 计时器位接通, 当前值 = 0 OFF: 在接通至断开转变后计时器计数	计时器位断开 当前值 = 0

<sup>1</sup> 可以选择保持计时器当前值用于在电源循环中保持。关于S7-200 CPU的内存保留的信息参见第4章。



提示与技巧

参考文档CD上的“编程提示”的使用打开延迟计时器（TON）的采样程序。参见提示31

TON和TONR指令当允许输入接通时计时。在当前值等于或大于预置时间时，计时器位接通。

- 当允许输入为断开时，TON计时器的当前值被清除，而当输入断开时，TONR计时器的当前值保持。
- 当输入接通和断开时，可以使用TONR计时器累加时间。使用“重新设定”指令（R）清除TONR的当前值。
- 在预置到达后，TON和TONR计时器继续计数，在到达最大值32,767时停止计数。

TOF指令用于在输入断开后，在一段固定的时间内延迟将输出断开。当允许输入接通，计时器位立即接通，并且当前值设置为0。当输入变断开时，计时器仍计时，直到过去的时间达到预置时间。

- 当预置达到时，计时器位断开，当前值停止递增；然而，如果在TOF达到预设值之前输入再次接通，计时器位保持接通。
- 允许输入必须作接通至断开的转变，用于TOF开始计数时间间隔。
- 如果TOF计时器在SCR区域内部，并且SCR区域非现用，那么当前值设置为0，计时器位断开，当前值不递增。



#### 提示

只需通过使用复位（R）指令来复位TONR。也可以使用“重新设定”指令来重新设定任何TON或TOF。“重新设定”指令完成下列操作：

- 计时器位 = 断开
- 计时器当前值 = 0

在重新设定后，TOF计时器需要允许输入从接通变为断开，以为了计时器重新启动。

## 6

### 确定计时器的分辨率

计时器计数时间间隔。计时器的分辨率（或时基）确定每个间隔的时间数。例如，具有10毫秒分辨率的TON计算在TON允许后经过的10毫秒间隔数字：在10毫秒计时器上计数50代表500毫秒。SIMATIC计时器有三种分辨率可用：1毫秒、10毫秒和100毫秒。如表6-72所示，计时器号码确定计时器的分辨率。



#### 提示

为保证最小时间间隔，预设值（PV）增加1。例如：对于100毫秒计时器，为确保最小时间间隔至少2100毫秒，设置PV为22。

表6-72 计时器号码和分辨率

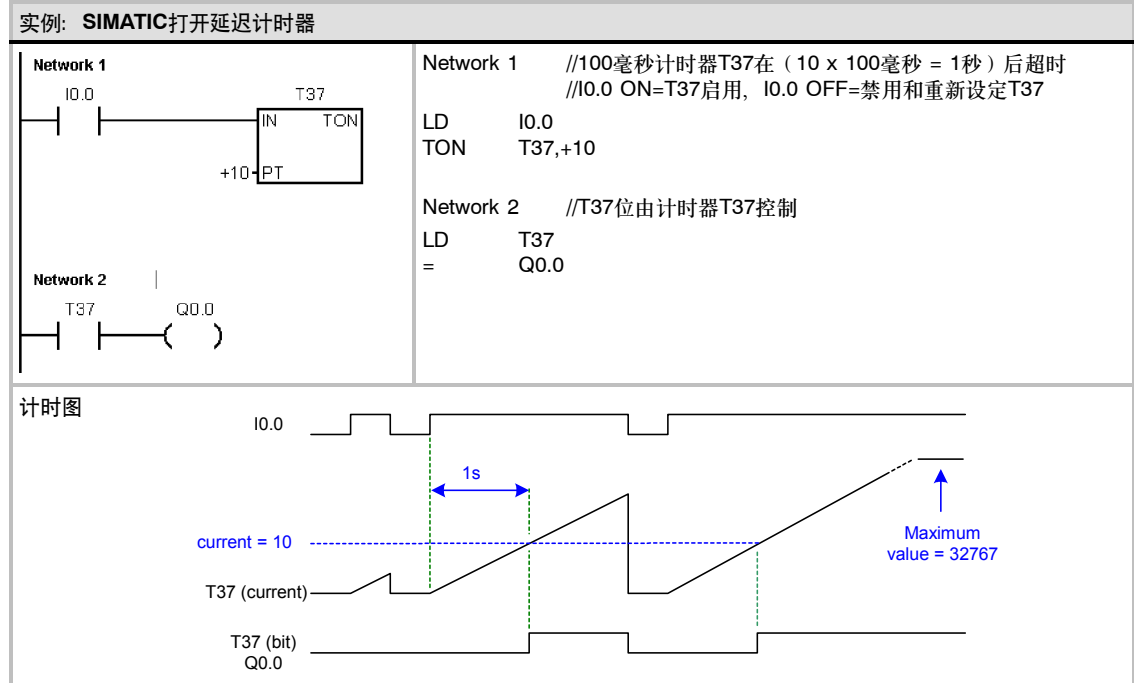
计时器类型	分辨率	最大值	计时器编号
TONR (保持的)	1毫秒	32.767秒 (0.546分钟)	T0、T64
	10毫秒	327.67秒 (5.46分钟)	T1到T4, T65到T68
	100毫秒	3276.7秒 (54.6分钟)	T5到T31, T69到T95
TON、TOF (非保持)	1毫秒	32.767秒 (0.546分钟)	T32、T96
	10毫秒	327.67秒 (5.46分钟)	T33到T36, T97到T100
	100毫秒	3276.7秒 (54.6分钟)	T37到T63, T101到T255

## 了解分辨率如何影响计时器操作

对于具有1毫秒分辨率的计时器，计时器位和当前值异步更新到扫描循环。对于扫描大于1毫秒，计时器位和当前值在整个扫描期间更新多次。

对于具有10毫秒分辨率的计时器，计时器位和当前值在每次扫描循环的开始更新。计时器位和当前值在整个扫描期间保持常量，在扫描期间积聚的时间间隔在每次扫描开始时添加到当前值。

对于具有100毫秒分辨率的计时器，计时器位和当前值在执行指令时更新。因此，确保程序每个扫描循环只为100毫秒计时器执行指令一次，以为了计时器保持正确的计时。



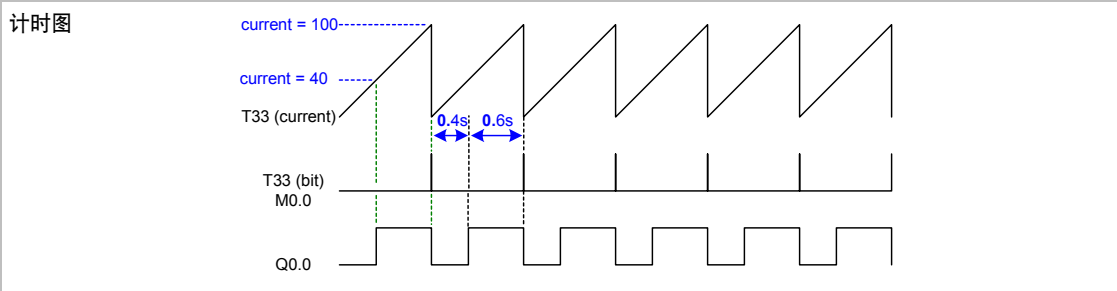
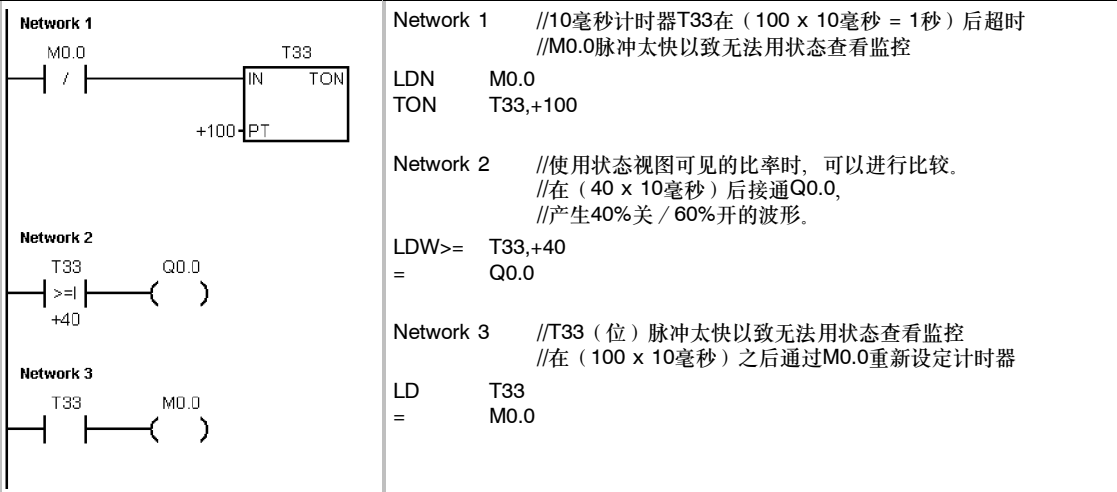


**提示**

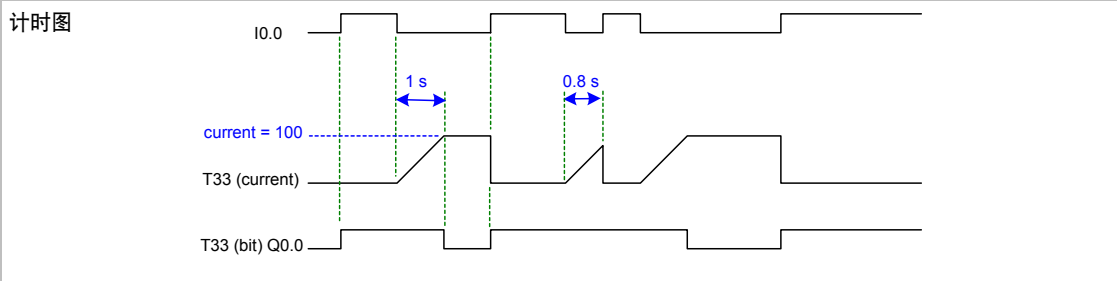
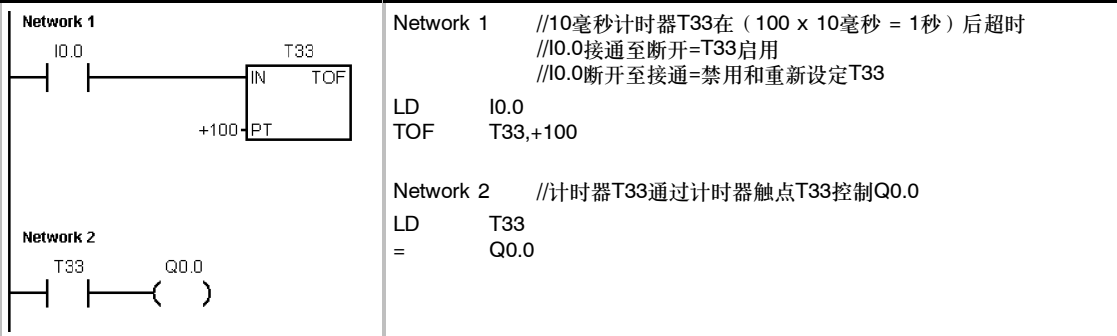
为保证每次计时器达到预设值时，自复位计时器的输出接通保持一个扫描，使用常闭触点而不是计时器位作为到计时器的允许输入。

6

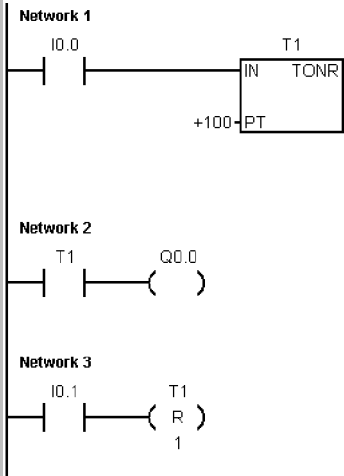
**实例: SIMATIC自重新设定打开延迟计时器**



**实例: SIMATIC关闭延迟计时器**



## 实例: SIMATIC保留性打开延迟计时器



Network 1 //10毫秒TONR计时器T1在PT= ( 100 x 10毫秒 = 1秒 ) 时  
//超时

```
LD I0.0
TONR T1,+100
```

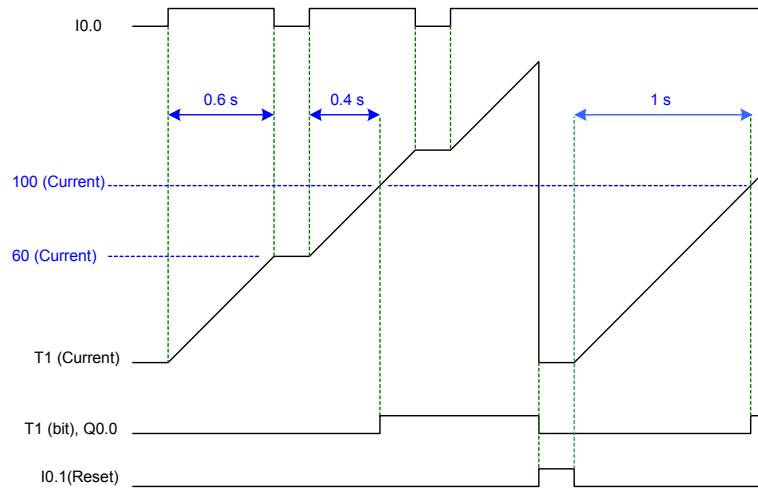
Network 2 //T1位由计时器T1控制  
//在计时器积累总计1秒后Q0.0接通

```
LD T1
= Q0.0
```

Network 3 //TONR计时器必须由带有T地址的“重新设定”指令  
//重新设定。  
//当I0.1接通时，复位计时器T1（趋势和位）。

```
LD I0.1
R T1,1
```

计时图



## IEC计时器指令

### 打开延迟计时器

打开延迟计时器（TON）指令当允许输入接通对时间计数。

### 关闭延迟计时器

“关闭延迟计时器”（TOF）在输入断开后，在一段固定的时间内延迟将输出断开。

### 脉冲计时器

脉冲计时器（TP）在指定持续时间内产生时钟脉冲。

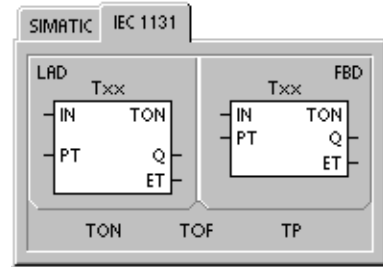


表6-73 IEC计时器指令的有效操作数

输入 / 输出	数据类型	操作数
Txx	TON、TOF、TP	常数（T32到T63、T96到T255）
IN	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
PT	INT	IW、QW、VW、MW、SMW、SW、LW、AC、AIW、*VD、*LD、*AC、常量
Q	BOOL	I、Q、V、M、SM、S、L
ET	INT	IW、QW、VW、MW、SMW、SW、LW、AC、AQW、*VD、*LD、*AC



#### 提示

对于TOF、TON和TP，不能共享同样的计时器号码。例如，不能都有TONT32和TOFT32。

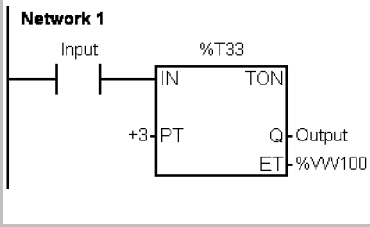
- 当允许输入（IN）变为真时，TON指令计数时间间隔直到预设值。当过去的时间（ET）等于“预置时间”（PT）时，计时器输出位（Q）接通。当允许输入断开时，输出位重新设定。当达到预置时，计时器停止，计时器禁用。
- TOF指令在输入断开后，在一段固定的时间内延迟将输出设定为断开。当允许输入（IN）断开时，它计时直到预设值。当过去的时间（ET）等于“预置时间”（PT）时，计时器输出位（Q）断开。当预置达到时，计时器输出位断开，过去的时间保持，直到允许输入接通。如果允许输入设置转变到的时间小于预置时间，输出位保持接通。
- TP指令产生指定持续时间的脉冲。当允许输入（IN）接通时，输出位（Q）也接通。对于指定的时钟脉冲在预置时间（PT）内，输出位保持接通。当过去的时间（ET）达到预置（PT）时，输出位断开。过去的时间保持直到允许输入断开。当输出位变接通时，它保持接通直到脉冲时间过去。

当前值的每个计数是时基的倍数。例如，在10毫秒计时器上计数50代表500毫秒。IEC计时器（TON、TOF和TP）有三种分辨率可用。分辨率由计时器号码确定，如表6-74所示。

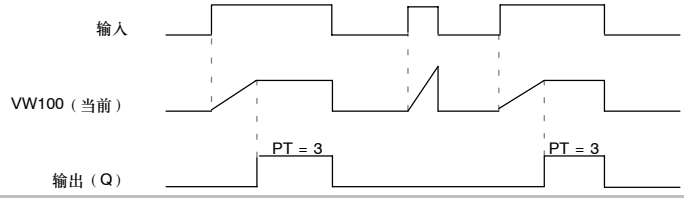
表6-74 IEC计时器的分辨率

分辨率	最大值	计时器编号
1毫秒	32.767秒 (0.546分钟)	T32、T96
10毫秒	327.67秒 (5.46分钟)	T33到T36, T97到T100
100毫秒	3276.7秒 (54.6分钟)	T37到T63, T101到T255

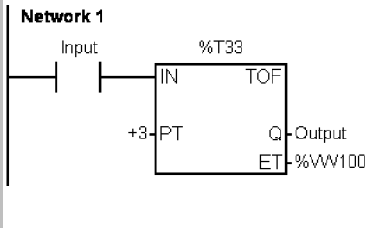
实例: IEC打开延迟计时器指令



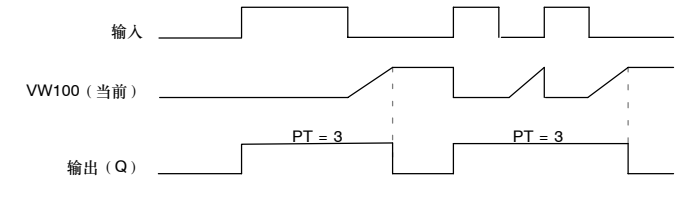
计时图



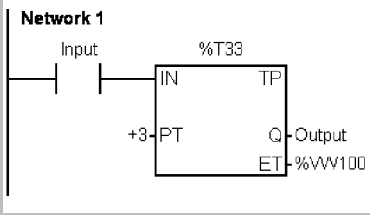
实例: IEC关闭延迟计时器指令



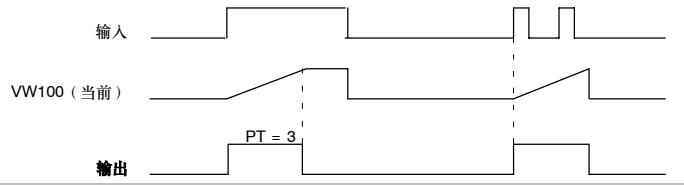
计时图



实例: IEC脉冲计时器指令



计时图



## 子例行程序指令

调用子例行程序指令（CALL）将控制传送到子例行程序SBR\_N。可以带或不带参数使用“调用子例行程序”指令。在子程序执行完成后，控制返回到“调用子例行程序”后的指令。

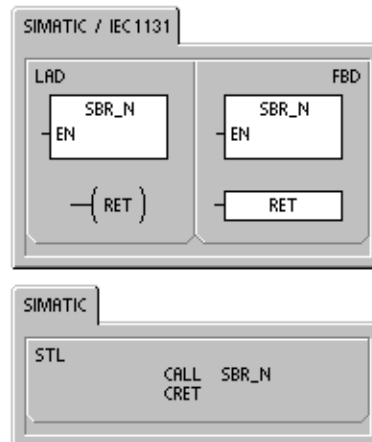
“从子例行程序有条件返回”指令（CRET）基于在前的逻辑终止子例行程序。

要添加子例行程序，选择编辑（Edit）>插入（Insert）>子例行程序（Subroutine）菜单命令。

设置ENO=0的错误条件

- 0008（最大子例行程序嵌套超出）
- 0006（间接地址）

从主程序，可以嵌套子例行程序（将子例行程序调用放在子例行程序中）最多达八层。无法从中断例行程序嵌套子例行程序。



# 6

子例行程序调用不能放在从中断例行程序调用的任何子例行程序中。递归（调用自己的子例行程序）不禁，但是当使用递归子例行程序时要当心。

表6-75 子例行程序指令的有效操作数

输入 / 输出	数据类型	操作数
SBR_N	WORD	常量 对于 CPU 221、CPU 222、CPU 224 和 CPU 226: 0到63 对于 CPU 226XM: 0至127
IN	BOOL BYTE WORD、INT DWORD、DINT	V、I、Q、M、SM、S、T、C、L、功率流 VB、IB、QB、MB、SMB、SB、LB、AC、*VD、*LD、*AC <sup>1</sup> 、常量 VW、T、C、IW、QW、MW、SMW、SW、LW、AC、AIW、*VD、*LD、*AC <sup>1</sup> 、常量 VD、ID、QD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC <sup>1</sup> 、&VB、&IB、&QB、&MB、&T、&C、&SB、常量
IN/OUT	BOOL BYTE WORD、INT DWORD、DINT	V、I、Q、M、SM <sup>2</sup> 、S、T、C、L VB、IB、QB、MB、SMB <sup>2</sup> 、SB、LB、AC、*VD、*LD、*AC <sup>1</sup> VW、T、C、IW、QW、MW、SMW <sup>2</sup> 、SW、LW、AC、*VD、*LD、*AC <sup>1</sup> VD、ID、QD、MD、SMD <sup>2</sup> 、SD、LD、AC、*VD、*LD、*AC <sup>1</sup>
OUT	BOOL BYTE WORD、INT DWORD、DINT	V、I、Q、M、SM <sup>2</sup> 、S、T、C、L VB、IB、QB、MB、SMB <sup>2</sup> 、SB、LB、AC、*VD、*LD、*AC <sup>1</sup> VW、T、C、IW、QW、MW、SMW <sup>2</sup> 、SW、LW、AC、AQW、*VD、*LD、*AC <sup>1</sup> VD、ID、QD、MD、SMD <sup>2</sup> 、SD、LD、AC、*VD、*LD、*AC <sup>1</sup>

<sup>1</sup> 必须为偏移量1或以上

<sup>2</sup> 必须为偏移量30或以上



### 提示

STEP 7-Micro/WIN自动从每个子例行程序添加无条件的回车。

当调用子例行程序时，整个逻辑堆栈保存，堆栈的顶部设置为一，所有其它堆栈位置设置为零，控制传送到调用的子例行程序。当此子例行程序完成，堆栈用调用点保存的值恢复，控制返回到调用的例行程序。

累加器对子例行程序和调用的例行程序是公用的。由于子例行程序使用，在累加器上没有进行保存或恢复操作。



## 调用带参数的子例行程序

子例行程序可以包含已被传递的参数。参数在子例行程序的局部变量表中定义。参数必须有符号名称（最大23个字符）、变量类型和数据类型。16个参数可以被传递或从子例行程序传出。

局部变量表中的变量类型域定义变量传入子例行程序（IN）、传入和传出子例行程序（IN\_OUT），还是传出子例行程序（OUT）。表6-76描述子例行程序的参数类型。要添加参数条目，将光标放在要添加的类型（IN、IN\_OUT或OUT）的变量类型域中。单击鼠标按钮得到选项菜单。选择Insert（插入）选项，然后选择Row Below（下面行）选项。另一个所选类型的参数条目在下面显示当前条目。

表6-76 子例行程序的参数类型

参数	描述
IN	参数被传递到子例行程序中。如果参数是直接地址（诸如VB10），指定位置的值被传递到子例行程序。如果参数是间接地址（诸如*AC1），指向位置的值被传递到子例行程序。如果参数是数据常量（16#1234）或地址（&VB100），常量或地址数值被传递到子例行程序。
IN_OUT	指定参数位置的值传入子例行程序，来自子例行程序的结果数值返回到同样的位置。常量（诸如16#1234）和地址（诸如&VB100）不允许用于输入/输出参数。
OUT	来自子例行程序的结果数值返回到指定的参数位置。常量（诸如16#1234）和地址（诸如&VB100）不允许作为输出参数。
TEMP	不用于已被传递的参数的任何局部内存可以在子例行程序中用作临时存储器。

如图6-38所示，局部变量表中的数据类型的定义参数的大小和格式。参数类型在下面列出：

- BOOL**: 此数据类型用于单个位输入和输出。在下列实例中IN3是布尔型输入。
- BYTE、WORD、DWORD**: 这些数据类型分别识别1、2或4个字节无符号输入或输出参数。
- 整数、长整数**: 这些数据类型分别识别2或4个字节有符号输入或输出参数。

Name	Var Type	Data Type	Comment
EN	IN	BOOL	
LB0	IN	BYTE	First pass flag
LB1	IN	BYTE	Address of slave device
LB2	IN	INT	Data to write to slave
LB4	IN_OUT	BYTE	Status of write
LB5	OUT	BOOL	Done flag
LB6	OUT	WORD	Error number [if any]

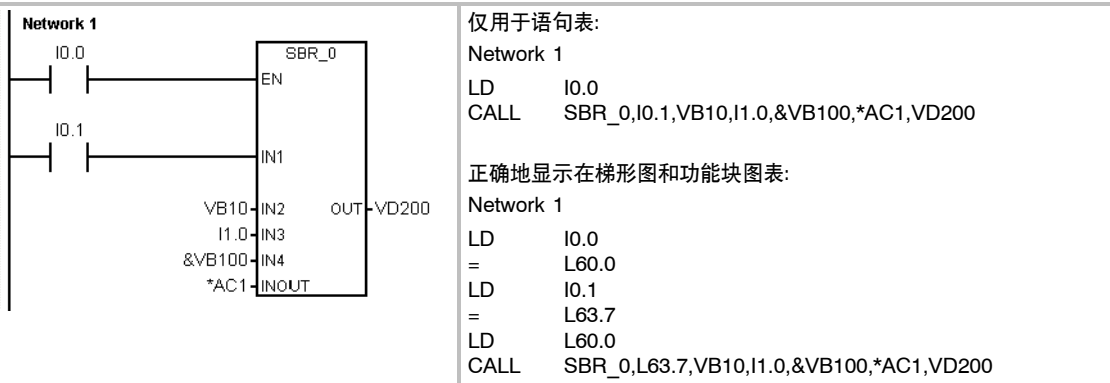
图6-38 局部变量表

- 实数**: 此数据类型识别单精度型（4字节）IEEE浮点数值。
- 功率流**: 布尔型功率流只允许位（布尔型）输入。这说明告诉STEP7micro/WIN此输入参数是基于位逻辑指令组合的功率流的结果。布尔型功率流输入在任何其它类型输入前必须首先显示在局部变量表中。只有输入参数允许以此方式使用。在下列实例中允许输入（EN）和IN1输入使用布尔型逻辑。

## 实例：子例行程序调用

提供两种语句表实例。第一集语句表指令只能在语句表编辑器中显示，因为用作功率流输入的布尔型参数没有保存到L内存。

第二集语句表指令也可以显示在梯形图和功能块图表编辑器，因为L内存用于保存作为功率流输入在梯形图和功能块图表中显示的布尔型输入参数的状态。



地址参数诸如IN4(&VB100)传递入子例行程序作为双字（无符号双字）数值。常量参数的类型必须为参数在调用的例行程序中用常量描述符在常量数值前指定。例如，要传递无符号双字常量数值12,345作为参数，常量参数必须被指定为DW#12345。如果常量描述符从参数中省略，常量可以假设为不同的类型。

没有对输入或输出参数的自动数据类型转换。例如，如果局部变量表指定参数具有数据类型实数，而在调用的例行程序中指定该参数双字（DWORD），子例行程序中的数值将会是双字。

当数值传递到子例行程序，它们被放在子例行程序的局部内存中。局部变量表最左边的列显示每个已被传递的参数的局部内存地址。当子例行程序被调用时，输入参数值被复制到子例行程序的局部内存。输出参数值当子例行程序执行完成时，被从子例行程序的局部内存复制到指定的输出参数地址。

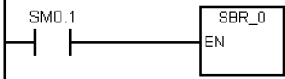
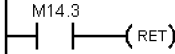
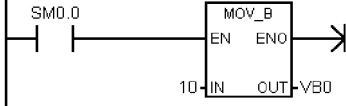
数据元素大小和类型以参数的代码代表。到子例行程序中局部内存的参数值分配如下：

- 参数值以调用子例行程序指令指定的顺序分配到局部内存，参数以L.0开始。
- 一至八个连续的位参数数值分配给单字节，以Lx.0开始，连续到Lx.7。
- 字节、字和双字数值以字节为界限（LBx、LWx或LDx）分配到局部内存。

在调用带参数的子例行程序指令中，参数必须按顺序排列，首先是输入参数，然后是输入 / 输出参数，最后是输出参数。

如果用语句表编程，调用指令的格式是：

调用子例行程序号码、参数1、参数2、...、参数

实例：子例行程序和从子例行程序指令返回		
M A I N	<p><b>Network 1</b></p> 	<p>Network 1 //一旦首先扫描，则为初始化调用子例行程序0。</p> <pre>LD SMO.1 CALL SBR_0</pre>
S B R 0	<p><b>Network 1</b></p>  <p><b>Network 2</b></p> 	<p>Network 1 //在最后的程序段之前，可以使用有条件返回 //离开子例行程序。</p> <pre>LD M14.3 CRET</pre> <p>Network 2 //如果M14.3接通，此程序段将跳过。</p> <pre>LD SMO.0 MOVB 10,VBO</pre>



# 7

## 通过网络进行通讯

S7-200旨在解决通讯与连网需求，它不仅支持最简单的网络，而且也支持较复杂的网络。S7-200还提供了能够与其它设备（例如使用其自身通讯协议的打印机和计量器）进行通讯的工具。

STEP 7-Micro/WIN使网络的组建和配置变得简单明了。

### 本章内容

了解S7-200网络通讯基础知识 .....	210
选择网络通讯协议 .....	214
安装和删除通讯接口 .....	220
构建网络 .....	222
创建具有自由端口模式的自定义协议 .....	227
在网络中使用调制解调器和STEP 7-Micro/WIN .....	229
高级主题 .....	235
配置用于远程操作的RS-232/PPI多台主设备电缆 .....	241

## 了解S7-200网络通讯基础知识

### 选择网络通讯接口

S7-200支持许多不同类型的通讯网络。网络选择可在“设置PG/PC接口”属性对话框中完成。所选网络将作为接口引用。有不同类型的接口可用于访问这些通讯网络：

- PPI多台主设备电缆
- CP通讯卡
- 以太网通讯卡

要为STEP 7-Micro/WIN选择通讯接口，可执行下列步骤，参见图7-1。

1. 双击通讯设置窗口中的图标。
2. 选择STEP 7-Micro/WIN的接口参数。

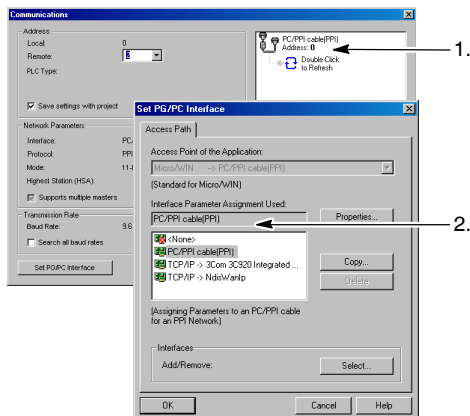


图7-1 STEP 7-Micro/WIN通讯接口

### PPI多台主设备电缆

S7-200支持通过两种不同类型的PPI多台主设备电缆进行通讯。这些电缆类型允许通过RS-232或USB接口进行通讯。

如图7-2所示，选择PPI多台主设备电缆类型很简单。请执行下列步骤：

1. 单击“设置PG/PC接口属性”页面上的“属性”按钮。
2. 单击“属性”页面上的“本地连接”标签。
3. 选择USB或期望的COM端口。

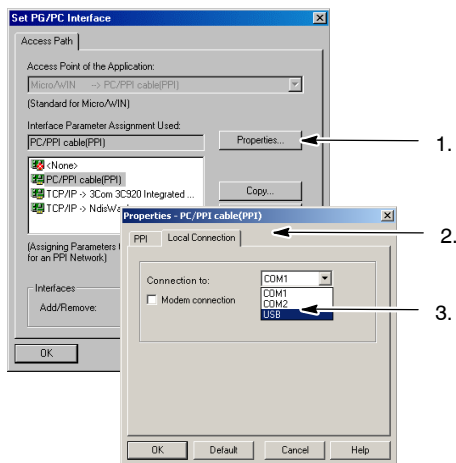


图7-2 PPI多台主设备电缆选择



#### 提示

请注意一次只能使用一根USB电缆。



#### 提示

本手册中的实例使用RS-232/PPI多台主设备电缆。RS-232/PPI多台主设备电缆替换以前的PC/PP1电缆。也可以使用USB/PPI多台主设备电缆。请参见附录中的订购号。

## 使用PROFIBUS网络上的主设备和从属装置

S7-200支持主从设备网络，在PROFIBUS网络中，既可以用作主设备，也可以用作从属装置，而STEP 7-Micro/WIN则始终作为主设备。

### 主设备

在网络上作为主设备的设备可以启动对网络上其它设备的请求。主设备也可以响应来自网络上其它主设备的请求。典型的主设备包括STEP 7-Micro/WIN、人机界面设备，例如TD 200、S7-300或S7-400PLC。S7-200从其它S7-200请求信息时用作主设备（对等通讯）。



### 提示

TP070不能与其它主设备在同一网络上工作。

### 从属装置

配置为从属装置的设备只能响应来自主设备的请求；从属装置永不发出请求。对于大多数网络，S7-200都用作从属装置。作为从属装置，S7-200将响应来自网络主设备的请求，例如操作员面板或STEP 7-Micro/WIN的请求。

## 设置波特率和网络地址

网络上的数据传输速度就是波特率，一般用千波特（kbaud）或兆波特（Mbaud）为单位进行测量。波特率测量给定时间范围内可传输的数据量。例如，波特率为19.2kbaud表示传输率为每秒19,200比特。

每个通过给定网络进行通讯的设备都必须配置为以相同的波特率传输数据。因此，网络的最大波特率将由连接网络的最慢设备来决定。

表7-1列出了S7-200所支持的各种波特率。

网络地址是分配给网络上每个设备的唯一编号。唯一的网络地址可确保数据传送到正确的设备或从正确的设备中返回。S7-200支持0到126的网络地址。对于具有两个端口的S7-200，每个端口都有一个网络地址。表7-2列出了S7-200设备的缺省（工厂）设置。

表7-1 S7-200所支持的波特率

网络	波特率
标准网络	9.6 kbaud到187.5 kbaud
使用EM277	9.6 kbaud到12 Mbaud
自由端口模式	1200波特到115.2 kbaud

表7-2 S7-200设备的缺省地址

S7-200设备	缺省地址
STEP 7-Micro/WIN	0
HMI（TD 200、TP或OP）	1
S7-200 CPU	2

### 设置STEP 7-Micro/WIN的波特率和网络地址

必须配置STEP 7-Micro/WIN的波特率和网络地址。波特率必须与网络其他设备的波特率相同，且网络地址必须唯一。

一般来讲，无需修改STEP 7-Micro/WIN的网络地址（0）。如果网络包括其它程序包，则或许需要修改STEP 7-Micro/WIN的网络地址。

如图7-3所示，配置STEP 7-Micro/WIN的波特率和网络地址十分简单。单击浏览条上的通讯图标后，请执行下列步骤：

1. 双击通讯设置窗口中的图标。
2. 单击“设置PG/PC接口”对话框上的“属性”按钮。
3. 选择STEP 7-Micro/WIN的网络地址。
4. 选择STEP 7-Micro/WIN的波特率。

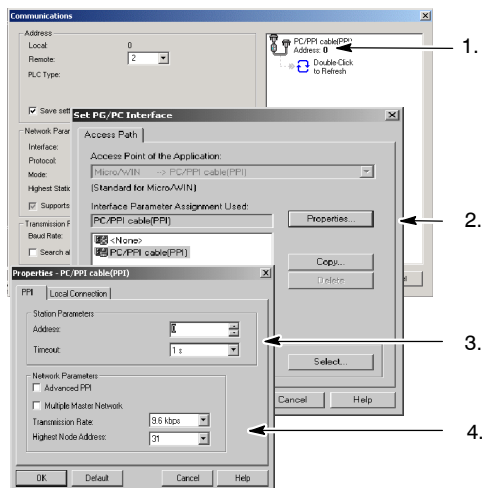


图7-3 配置STEP 7-Micro/WIN

### 设置S7-200的波特率和网络地址

还必须配置S7-200的波特率和网络地址。S7-200的系统块存储波特率和网络地址。选择S7-200的参数之后，必须将系统块下载到S7-200。

每个S7-200端口的缺省波特率为9.6 kbaud，缺省网络地址为2。

如图7-4所示，使用STEP 7-Micro/WIN来设置S7-200的波特率和网络地址。选择浏览条上的系统块图标或选择视图（View）> 组件（Component）> 系统块（System Block）菜单命令之后，请执行下列步骤：

1. 选择S7-200的网络地址。
2. 选择S7-200的波特率。
3. 将系统块下载到S7-200。

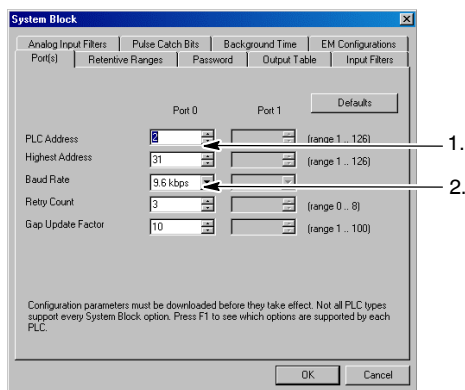


图7-4 配置S7-200 CPU



#### 提示

允许选择所有的波特率选项。STEP 7-Micro/WIN将在下载系统块期间对这种选择进行验证。可能使STEP 7-Micro/WIN无法与S7-200进行通讯的波特率选择将不会被下载。



### 设置远程地址

在将已更新的设置下载到S7-200之前，STEP 7-Micro/WIN的通讯（COM）端口（本地）和S7-200的地址（远程）二者都必须进行设置，以便与远程S7-200的当前设置相匹配。参见图7-5。

下载已更新的设置之后，可能需要重新配置PG/PC接口波特率设置（如果这些设置与下载到远程S7-200时所使用的设置不同）。参考图7-3来配置波特率。

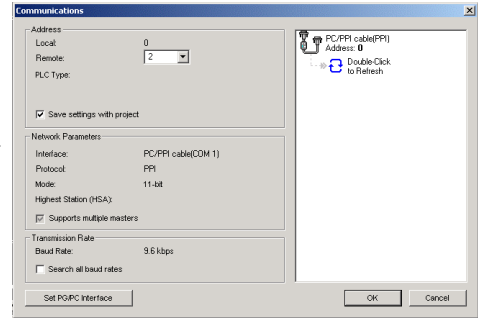


图7-5 配置STEP 7-Micro/WIN

### 搜索网络中的S7-200 CPU

可对连接到网络的S7-200 CPU进行搜索和标识。搜索S7-200时，也可按特定的波特率或所有波特率对网络进行搜索。

仅PPI多台主设备电缆允许搜索所有波特率。该特点在通过CP卡进行通讯时不适用。将以当前所选择的波特率启动搜索。

1. 打开“通讯”对话框，双击“刷新”图标启动搜索。
2. 为搜索所有的波特率，可选择“搜索所有波特率”复选框。

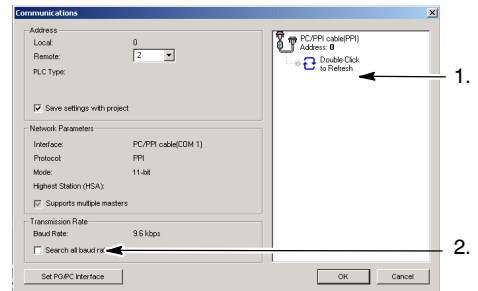


图7-6 搜索网络上的CPU

## 选择网络的通讯协议

下列信息是S7-200 CPU所支持协议的概述。

- 点对点接口 (PPI)
- 多点接口 (MPI)
- PROFIBUS

基于通讯结构的“开放系统互连”(OSI)七层模型, 这些协议用于令牌环网络, 该网络符合欧洲标准EN 50170所定义的PROFIBUS标准。这些协议是异步的、基于字符的协议, 具有一个起始位、八个数据位、一个偶数校验位和一个停止位。通讯帧将取决于特殊的启动与停止字符、源站地址与目标站地址、帧长度以及数据完整性的检验和。协议可同时在一个网络上运行, 而不会相互干扰, 只要每个协议的波特率相同。

以太网也可用于具有扩充模块CP243-1和CP243-1 IT的S7-200 CPU。

### PPI协议

PPI是一种主从设备协议: 主设备给从属装置发送请求, 从属装置进行响应。参见图7-7。从属装置不发出讯息, 而是一直等到主设备发送请求或轮询时才作出响应。

主设备与从属装置的通讯将通过按PPI协议进行管理的共享连接来进行。PPI不限制与任何一个从属装置进行通讯的主设备的数目; 然而, 网络上最多可安装32个主设备。

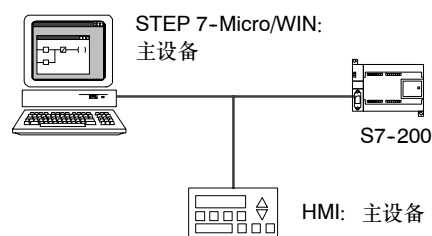


图7-7 PPI网络

如果在用户程序中激活PPI主设备模式, 则S7-200 CPU在处于RUN (运行) 模式时可用作主设备。(参见附录D中关于SMB30的描述。) 激活PPI主设备模式之后, 可使用“网络读取”或“网络写入”指令从其它S7-200读取数据或将数据写入其它S7-200。当S7-200用作PPI主设备时, 它将仍然作为从属装置对来自其他主设备的请求进行响应。

PPI高级协议允许网络设备建立设备之间的逻辑连接。对于PPI高级协议, 存在由每台设备所提供的有限数目的连接。参见表7-3来了解S7-200所支持连接的数目。

所有S7-200 CPU均支持PPI和PPI高级协议, 而PPI高级协议是EM 277模块所支持的唯一PPI协议。

表7-3 S7-200 CPU和EM 277模块的连接数目

模块	波特率	连接
S7-200 CPU 端口0	9.6 kbaud、19.2 kbaud或187.5kbaud	4
端口1	9.6 kbaud、19.2 kbaud或187.5kbaud	4
EM 277模块	9.6 kbaud到12 Mbaud	每个模块6个

## MPI协议

MPI允许进行主设备与主设备和主设备与从属装置之间的通讯。参见图7-8。为与S7-200 CPU进行通讯，STEP 7-Micro/WIN建立一个主设备与从属装置之间的连接。MPI协议不与用作主设备的S7-200 CPU进行通讯。

网络设备通过任意两台设备之间的独立连接（由MPI协议进行管理）进行通讯。设备之间的通讯将受限于S7-200 CPU或EM 277模块所支持的连接数目。参见表7-3以了解S7-200所支持连接数目。

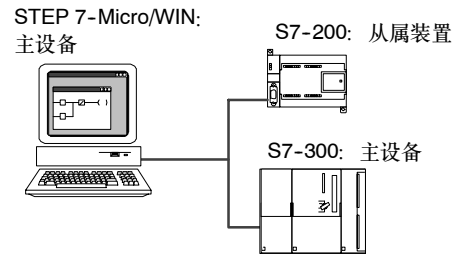


图7-8 MPI网络

对于MPI协议，S7-300和S7-400 PLC将使用XGET和XPUT指令从S7-200 CPU中读写数据。有关这些指令的信息，请参见S7-300或S7-400编程手册。

## PROFIBUS协议

PROFIBUS协议设计用于具有分布式I/O设备（远程I/O）的高速通讯。许多来自各个不同厂家的PROFIBUS设备均可使用。这些设备包括从简单的输入或输出模块到电动机控制器和PLC。

PROFIBUS网络的典型特点就是具有一个主设备和多个I/O从属装置。参见图7-9。将主设备配置为知道所连接的I/O从属装置的型号及地址。主设备将初始化网络，并验证网络上的从属装置是否与配置相符。主设备可将输出数据连续地写入从属装置，以及从中读出输入数据。

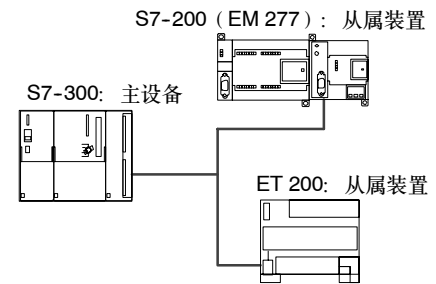


图7-9 PROFIBUS网络

当DP主设备成功地配置从属装置时，它就拥有了该从属装置。如果网络上存在第二个主设备，则它将只能十分有限地对属于第一个主设备的从属装置进行访问。

## TCP/IP协议

S7-200通过使用以太网（CP 243-1）或因特网（CP 243-1 IT）扩充模块可支持TCP/IP以太网通讯。表7-4显示了这些模块所支持连接数目和波特率。

表7-4 以太网（CP 243-1）和互联网（CP 243-1 IT）模块的连接数目

模块	波特率	连接
以太网（CP 243-1）模块	10到100 Mbaud	8个常规目的的连接
互联网（CP 243-1 IT）模块		1个STEP 7-Micro/WIN连接

请参见用于工业以太网的CP 243-1通讯处理器手册或用于工业以太网和信息技术的CP 243-1 IT通讯处理器手册以获取附加的信息。

## 仅使用S7-200设备的范例网络配置

### 单台主设备PPI网络

对于简单的单台主设备网络，编程站和S7-200 CPU既可以通过PPI多台主设备电缆连接，也可以通过安装在编程站中的通讯处理器（CP）卡连接。

在图7-10上部的范例网络中，编程站（STEP 7-Micro/WIN）是网络主设备。在图7-10下部的范例网络中，人机界面（HMI）设备（例如TD 200、TP或OP）是网络主设备。

在两个范例网络中，S7-200 CPU是对主设备的请求进行响应的从属装置。

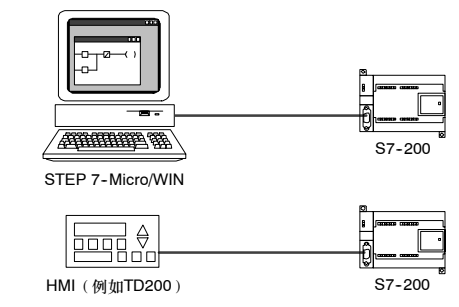


图7-10 单台主设备PPI网络

对于单台主设备PPI网络，配置STEP 7-Micro/WIN来使用PPI协议。如果可能，取消勾选“多台主设备网络”和“PPI高级协议”复选框。

### 多台主设备PPI网络

图7-11显示了一个具有一个从属装置的多台主设备范例网络。编程站（STEP 7-Micro/WIN）既可以使用CP卡，也可以使用PPI多台主设备电缆。STEP 7-Micro/WIN和HMI设备共享网络。

STEP 7-Micro/WIN和HMI设备是主设备，必须具有独立的网络地址。在使用PPI多台主设备电缆时，电缆是主设备，使用STEP 7-Micro/WIN所提供的网络地址。S7-200 CPU是从属装置。

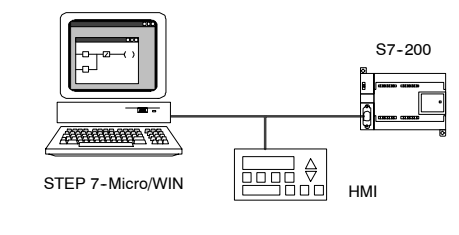


图7-11 具有一个从属装置的多台主设备

图7-12显示了一个PPI网络，它具有与多台从属装置进行通讯的多台主设备。在本实例中，STEP 7-Micro/WIN和HMI都将从任意S7-200 CPU从属装置请求数据。STEP 7-Micro/WIN和HMI设备共享网络。

所有设备（主设备和从属装置）均具有不同的网络地址。在使用PPI多台主设备电缆时，电缆是主设备，并使用STEP 7-Micro/WIN所提供的网络地址。S7-200 CPU为从属装置。

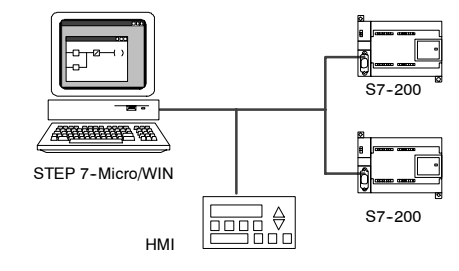


图7-12 多台主设备和从属装置

对于具有多台主设备和一台或多台从属装置的网络，配置STEP 7-Micro/WIN来使用PPI协议，如果可用，勾选“多台主设备网络”和“PPI高级协议”复选框。如果正在使用PPI多台主设备电缆，则“多台主设备网络”和“PPI高级协议”复选框均将忽略。

### 复杂PPI网络

图7-13显示了一个范例网络，该网络使用了具有对等通讯的多台主设备。

STEP 7-Micro/WIN和HMI设备通过网络对S7-200 CPU进行读写，S7-200 CPU使用“网络读取”和“网络写入”指令互相读写（对等通讯）。

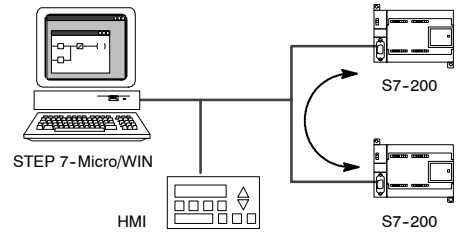


图7-13 对等通讯

图7-14显示了另一个复杂PPI网络的实例，该网络使用了具有对等通讯的多台主设备。在本实例中，每个HMI监控一个S7-200 CPU。

S7-200 CPU使用NETR和NETW指令互相读写（对等通讯）。

对于复杂PPI网络，配置STEP 7-Micro/WIN来使用PPI协议，如果可用，勾选“多台主设备网络”和“PPI高级协议”复选框。如果正在使用PPI多台主设备电缆，则“多台主设备网络”和“PPI高级协议”复选框均将忽略。

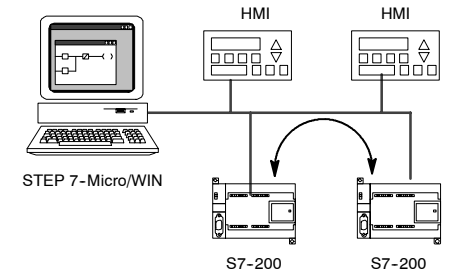


图7-14 HMI设备与对等通讯

## 使用S7-200、S7-300和S7-400设备的范例网络配置

### 波特率最高为187.5 kbaud的网络

在图7-15所示的范例网络中，S7-300使用XPUT和XGET指令与S7-200 CPU进行通讯。在主设备模式中，S7-300不能与S7-200 CPU进行通讯。

为了与S7 CPU进行通讯，配置STEP 7-Micro/WIN来使用PPI协议，如果可用，勾选“多台主设备网络”和“PPI高级协议”复选框。如果正在使用PPI多台主设备电缆，则“多台主设备网络”和“PPI高级协议”复选框均将忽略。

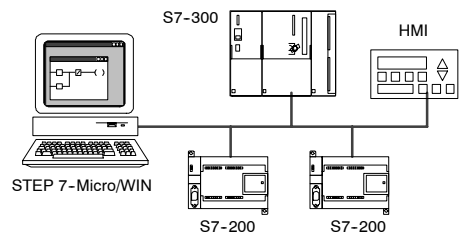


图7-15 波特率最大为187.5 Kbaud

### 波特率在187.5 kbaud以上的网络

对于187.5 kbaud以上的波特率，S7-200 CPU必须使用EM 277模块，以便连接到网络。参见图 7-16。STEP 7-Micro/WIN必须通过通讯处理器（CP）卡进行连接。

在该配置中，使用XPUT和XGET指令，S7-300可与S7-200进行通讯，而HMI既可以监控S7-200，也可以监控S7-300。

EM 277始终是从属装置。

STEP 7-Micro/WIN可通过所附EM 277对S7-200 CPU进行编程或监控。为了与187.5 kbaud以上的EM 277进行通讯，配置STEP 7-Micro/WIN，以使用具有CP卡的MPI协议。PPI多台主设备电缆的最大波特率为187.5 kbaud。

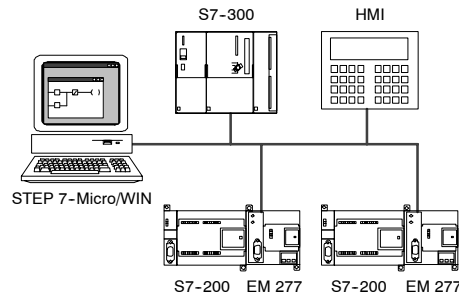


图7-16 187.5 Kbaud以上的波特率

## 范例PROFIBUS网络配置

以**S7-315-2 DP**作为**PROFIBUS**主设备、**EM 277**作为**PROFIBUS**从属装置的网络

图7-17表示了一个使用S7-315-2 DP作为PROFIBUS主设备的范例PROFIBUS网络。EM 277模块为PROFIBUS从属装置。

S7-315-2 DP可读取EM 277的数据或将数据写入EM 277，从1个字节到128个字节不等。S7-315-2 DP对S7-200中的V存储区单元进行读写。

该网络支持9600 baud到12 Mbaud的波特率。

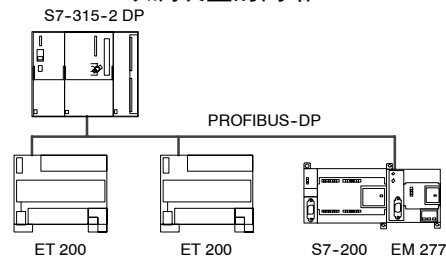


图7-17 具有S7-315-2 DP的网络

具有**STEP 7-Micro/WIN**和**HMI**的网络

图7-18显示了一个S7-315-2 DP作为PROFIBUS主设备、EM 277作为PROFIBUS从属装置的范例网络。在该配置中，HMI通过EM 277对S7-200进行编程。STEP 7-Micro/WIN通过EM 277对S7-200进行编程。

该网络支持9600 baud到12 Mbaud的波特率。STEP 7-Micro/WIN需要一个187.5 kbaud以上波特率的CP卡。

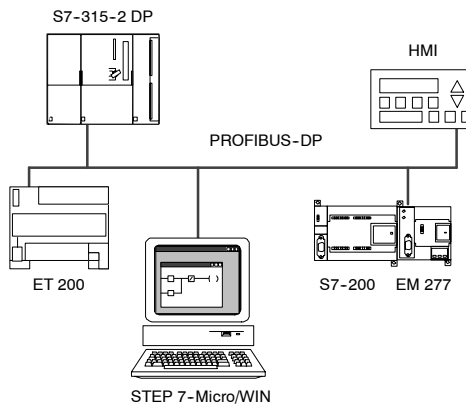


图7-18 PROFIBUS网络

配置STEP 7-Micro/WIN，以使用用于CP卡的PROFIBUS协议。如果只有DP设备出现在网络上，则选择DP或标准配置文件。如果网络上存在任意的非DP设备，例如TD200，则选择适用于所有主设备的“通用（DP/Fms）”配置文件。网络上的所有主设备都必须设置为使用相同的PROFIBUS配置文件（DP、标准或通用），用于网络的运行。

只有所有主设备均使用“通用（DP/Fms）”配置文件，PPI多台主设备电缆在网络上才能发挥出187.5 kbaud的作用。

## 使用以太网和/或互联网设备的范例网络配置

在如图7-19所示的配置中，以太网连接将用于使STEP 7-Micro/WIN能够和正在使用以太网（CP 243-1）模块或互联网（CP 243-1 IT）模块的任意S7-200 CPU进行通讯。S7-200 CPU可通过以太网连接交换数据。在具有STEP 7-Micro/WIN的PC上运行的标准浏览器程序，将用于访问互联网（CP 243-1 IT）模块的主页。

对于以太网网络，配置STEP 7-Micro/WIN，以使用TCP/IP协议。

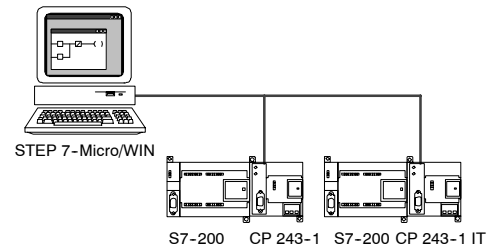


图7-19 10/100 Mbaud以太网网络



### 提示

在“设置PG/PC接口”对话框中，至少存在两种TCP/IP选择。S7-200不支持选择“TCP/IP -> NdisWanIp”。

- 在“设置PG/PC接口”对话框中，选项将取决于PC所提供的以太网接口的类型。选择一种类型将计算机连接到连接了CP 243-1或CP 243-1 IT模块的以太网网络。
- 在“通讯”对话框上，必须输入想要与其进行通讯的每个以太网/互联网模块的远程IP地址。

## 安装和删除通讯接口

从“设置PG/PC接口”对话框中，使用“安装/卸载接口”对话框来安装或删除计算机的通讯接口。

1. 在“设置PG/PC接口”对话框中，单击“选择”，以打开“安装/卸载接口”对话框。  
“选择”对话框将列出可供使用的接口，而“安装”对话框将显示计算机上已经安装的接口。
2. 为添加通讯接口：选择计算机上已安装的通讯硬件，并单击“安装”。关闭“安装/卸载接口”对话框时，“设置PG/PC接口”对话框将显示“已使用的接口参数分配”对话框中的接口。
3. 为删除通讯接口：选择将要删除的接口，并单击“卸载”。关闭“安装/卸载接口”对话框时，“设置PG/PC接口”对话框将从“已使用的接口参数分配”对话框中删除接口。

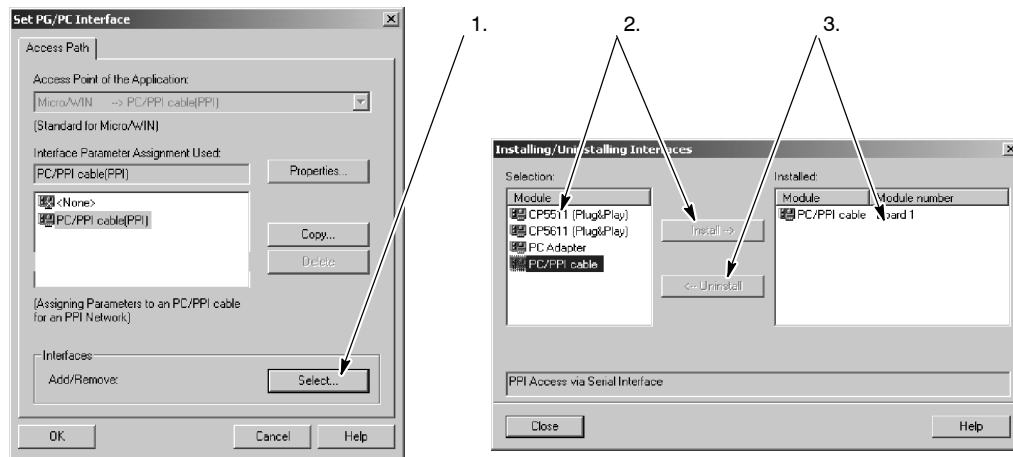


图7-20 “设置PG/PC接口”和“安装/取消安装接口”对话框



### 提示

#### 用于 Windows NT 用户的特殊硬件安装信息

在Windows NT操作系统下安装硬件模块与在Windows 95下安装硬件模块稍有不同。虽然硬件模块在任意操作系统下都是一样的，但在Windows NT下的安装将需要对希望安装的硬件具有更多的了解。Windows 95试图自动安装系统资源，但Windows NT则不会。Windows NT将只提供缺省值。这些数值可能或不可能与硬件配置相匹配。可方便地对这些参数进行修改，以便与所需要的系统设置相匹配。

安装完硬件后，从“已安装”列表框中选择硬件并单击“资源”按钮。显示“资源”对话框。“资源”对话框允许修改实际安装的硬件的系统设置。如果该按钮不能使用（按钮显示为灰色），则不需要再进行任何操作。

此时，可能需要查阅硬件手册，以便按照硬件设置，确定对话框中所列出的每个参数的设置。为了建立正确的通讯，或许需要尝试多种不同的中断。



### 调整用于PPI多台主设备的计算机端口设置

如果正在使用PPI模式下的USB/PPI多台主设备电缆或RS-232/PPI多台主设备电缆，则不需要调整计算机的端口设置，可以使用Windows NT操作系统进行多台主设备网络中的操作。

如果正在使用PPI/自由端口模式下的RS-232/PPI多台主设备电缆，用于S7-200 CPU与支持PPI多台主设备配置的操作系统（Windows NT不支持PPI多台主设备）上的STEP 7-Micro/WIN之间进行通讯，则可能需要调整计算机的端口设置：

1. 右击桌面上的“我的计算机”图标，选择“属性”菜单命令。
2. 选择“设备管理器”标签。对于Windows 2000，首先选择“硬件”标签，然后选择“设备管理器”按钮。
3. 双击“端口”（COM和LPT）。
4. 选择当前正在使用的通讯端口（例如COM1）。
5. 在“端口设置”标签上，单击“高级”按钮。
6. 将“接收缓冲区”和“传输缓冲区”控件设置为最低值（1）。
7. 单击“确定”按钮应用所作的修改，关闭所有窗口，然后重新启动计算机，以便使新的设置生效。

## 构建网络

### 常规准则

对于可能受快速电涌影响的任何配线，应始终安装合适的电涌抑制设备。

请避免将低压信号线和通讯电缆与AC导线和大功率的快速转换DC导线放在同一个导线盒中。应始终成对布线，导线采用中性导线或通用导线，并用热电阻线或信号线进行配对。

S7-200 CPU的通讯端口不绝缘。请考虑使用RS-485中继器或EM 277模块来提供网络的绝缘。

#### 当心

具有不同参考电位的互连设备将可能导致出现不必要的电流流过互连电缆。

这些不必要的电流可能导致通讯出错，甚至可能损坏设备。

请确保即将用通讯电缆连接的所有设备均具有公用电路参考电势，或对其进行绝缘，以避免产生不必要的电流。关于使用绝缘电路的接地和电流参考点等信息，请参见第3章。

### 确定网络的距离、传输率和电缆

如表7-5所示，网络段的最大长度取决于两个因素：绝缘（使用RS-485中继器）和波特率。

当连接不同接地电位的设备时，需要进行绝缘。如果接地点在物理上相隔较远的距离，就可能存在不同的接地电位。甚至在较短的距离内，重型机械的负荷电流也可能引起接地电位上的差异。

表7-5 网络电缆的最大长度

波特率	不绝缘的CPU端口 <sup>1</sup>	具有中继器或EM 277的CPU端口
9.6 kbaud到187.5 kbaud	50M	1,000M
500 kbaud	不支持	400M
1Mbaud到1.5Mbaud	不支持	200M
3Mbaud到12Mbaud	不支持	100M

<sup>1</sup> 不使用绝缘体或中继器时所允许的最大距离为50米。测量网络段中的第一个节点到最后一个节点之间的距离。

#### 在网上使用中继器

RS-485中继器提供了用于网络段的偏差和终端。使用中继器的目的如下：

- **为增加网络的长度：**为网络添加一个中继器，能够将网络再扩展50米。如果连接了相互之间没有任何其它节点的两个中继器（如图7-21所示），则可将网络扩展到波特率所允许的最大电缆长度。在一个网络中，最多可串联使用9个中继器，但网络的总长度不能超过9600米。
- **为在网络中添加设备：**每段网络最多可有传输率9600波特、最大连接长度50米的32个设备。使用中继器将允许给网络添加另一个网络段（32个设备）。
- **为使不同的网络段电气绝缘：**通过把可能具有不同接地电位的网络段隔开，网络绝缘将可改进传输的质量。

网络上的中继器将算作网络段上的节点之一，即使没有为其分配网络地址。

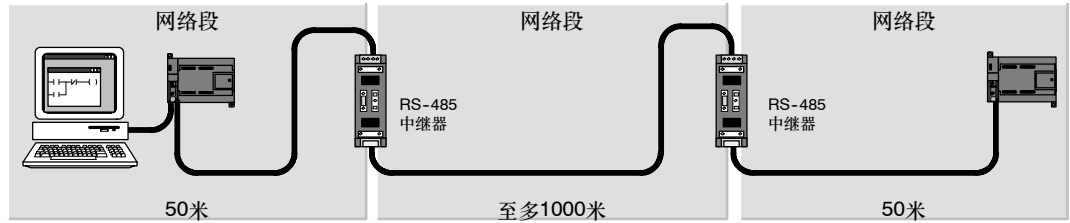


图7-21 具有中继器的范例网络

### 选择网络电缆

S7-200网络使用符合RS-485标准的双绞线电缆。表7-6列出了网络电缆的规格。在一个网络段上最多可连接32个设备。

表7-6 网络电缆的一般规格

规格	描述
电缆型号	屏蔽、双绞线
环路电阻	≤115 Ω/km
有效电容	30 pF/m
额定阻抗	大约135 Ω到160 Ω (频率 = 3 MHz 到 20 MHz)
衰减	0.9 dB/100 m (频率 = 200 kHz)
横断面核心区域	0.3 mm <sup>2</sup> 到0.5 mm <sup>2</sup>
电缆直径	8 mm±0.5 mm

### 接头插针分配

S7-200 CPU上的通讯端口为与RS-485兼容的九针微型D型连接器，它符合在欧洲标准EN 50170中所定义的PROFIBUS标准。表7-7显示了提供通讯端口物理连接的分连接器，并描述了通讯端口的插针分配。

表7-7 S7-200通讯端口的插针分配

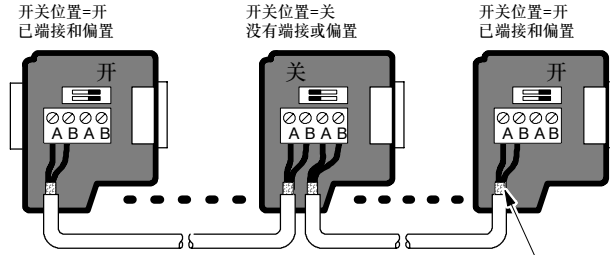
连接器	插针号	PROFIBUS信号	端口0/端口1
	1	屏蔽	机壳接地
	2	24 V回流	逻辑中性线
	3	RS-485信号B	RS-485信号B
	4	请求发送	RTS (TTL)
	5	5 V回流	逻辑中性线
	6	+5V	+5 V、100Ω串联电阻器
	7	+24V	+24V
	8	RS-485信号A	RS-485信号A
	9	不适用	10位协议选择 (输入)
连接器外壳	屏蔽	机壳接地	

## 偏置并端接网络电缆

西门子提供了两种类型的网络连接器，可用来方便地将多个设备连接到网络：标准网络连接器（参见表7-7的插针分配）和包含有编程端口的连接器。该连接器允许将编程站或HMI设备连接到网络，且对现有网络连接没有任何干扰。编程端口连接器将把所有信号（包括电源插针）从S7-200完全传递到编程端口，它特别适用于连接将从S7-200取电的设备（例如TD 200）。

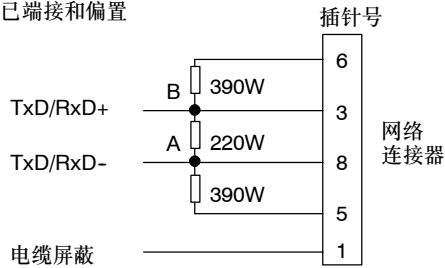
两种连接器都有两套终端螺丝，允许用来连接进入和出去的网络电缆。两种连接器还具有转换开关，以便有选择地偏置和端接网络。图7-22显示了用于电缆连接器的典型偏置和端接。

电缆在两端都必须端接和偏置。



裸防护层：大约12毫米（1/2英寸）必须接触所有位置的金属导轨。

开关位置=开：  
已端接和偏置



开关位置=关：  
没有端接或偏置

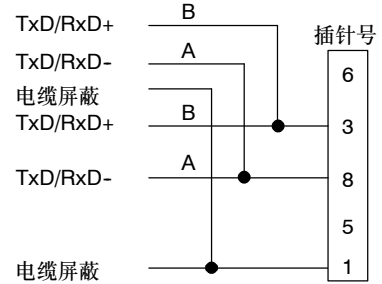


图7-22 网络电缆的偏置和端接

## 选择用于网络的PPI多台主设备电缆或CP卡

如表7-8所示，STEP 7-Micro/WIN支持RS-232/PPI多台主设备电缆和USB/PPI多台主设备电缆以及允许编程站（计算机或SIMATIC编程设备）作为网络主设备的多个CP卡。

对于至多187.5 kbaud的波特率，PPI多台主设备电缆提供STEP 7-Micro/WIN与S7-200 CPU或S7-200网络之间最简单、最经济有效的连接。这两个型号的PPI多台主设备电缆均可使用，并都可用于STEP 7-Micro/WIN与S7-200网络之间的本地连接。

USB/PPI多台主设备电缆是一种即插即用设备，可用于支持USB V1.1的PC。在支持至多以187.5 kbaud波特率进行通讯时，它将提供PC和S7-200网络之间的绝缘。没有任何转换装置要设置：只要连接电缆、选择PC/PPI电缆作为接口、选择PPI协议，并在“PC连接”标签中将端口设置为USB。STEP 7-Micro/WIN使用时，每次只能有一个USB/PPI多台主设备电缆连接到PC。

RS-232/PPI多台主设备电缆具有八个DIP开关：这些开关中有两个将用于配置运行STEP 7-Micro/WIN时的电缆。

- 如果将电缆连接到PC，请选择PPI模式（5号开关=1）和本地操作（6号开关=0）。
- 如果将电缆连接到调制解调器，请选择PPI模式（5号开关=1）和远程操作（6号开关=1）。

电缆将提供PC和S7-200网络之间的绝缘。选择PC/PPI电缆作为接口，在“PC连接”标签下选择希望使用的RS-232端口。在PPI标签下，选择站地址和网络波特率。不需要进行任何其它选择，因为RS-232/PPI多台主设备电缆下的协议选择是自动的。

USB/PPI和RS-232/PPI多台主设备电缆二者都具有LED，它将提供PC通讯活动以及网络通讯活动的指示。

- Tx LED指示电缆正在传输信息给PC。
- Rx LED指示电缆正在接收来自PC的信息。
- PPI LED指示电缆正在传输网络上的数据。因为多台主设备电缆是令牌的持有者，因此，一旦STEP 7-Micro/WIN启动通讯，PPI LED就将连续地闪烁。断开与STEP 7-Micro/WIN的连接时，PPI LED也将关闭。在等待加入网络时，PPI LED也将以1Hz的频率闪烁。

CP卡包含有帮助编程站管理多台主设备网络的专用硬件，并可支持多个波特率下的不同协议。

每个CP卡都提供有一个单独的RS-485端口，用于与网络的连接。CP 5511 PCMCIA卡具有一个适配器，可提供9插针D型端口。将电缆的一端连接到卡的RS-485端口，将电缆的另一端连接连接到网络上的编程端口连接线。

如果使用的是具有PPI通讯的CP卡，则STEP 7-Micro/WIN将不支持在同一CP卡上同时运行的两个不同的应用程序。在通过CP卡将STEP 7-Micro/WIN连接到网络之前，必须关闭其他应用程序。如果使用MPI或PROFIBUS通讯，则允许多个STEP 7-Micro/WIN应用程序通过网络同时进行通讯。

### 当心

使用不绝缘的RS-485到RS-232转换器有可能损坏计算机的RS-232端口。

西门子RS-232/PPI和USB/PPI多台主设备电缆（订购号分别为6ES7 901-3CB30-0XA0或6ES7 901-3DB30-0XA0）提供S7-200 CPU上的RS-485端口与连接到计算机的RS-232或USB端口之间的电气绝缘。如果不使用西门子多台主设备电缆，则必须提供对计算机RS-232端口的绝缘。

表7-8 STEP 7-Micro/WIN所支持的CP卡和协议

配置	波特率	协议
RS-232/PPI多台主设备或USB/PPI多台主设备电缆 <sup>1</sup> 连接到编程站的端口	9.6 kbaud到 187.5 kbaud	PPI
CP 5511 型号II、PCMCIA卡（用于笔记本电脑）	9.6 kbaud到 12 Mbaud	PPI、MPI和PROFIBUS
CP 5512 型号II、PCMCIA卡（用于笔记本电脑）	9.6 kbaud到 12 Mbaud	PPI、MPI和PROFIBUS
CP 5611（版本3或更新的版本） PCI卡	9.6 kbaud到 12 Mbaud	PPI、MPI和PROFIBUS
CP1613、S7613 PCI卡	10 Mbaud或 100 Mbaud	TCP/IP
CP1612、SoftNet7 PCI卡	10 Mbaud或 100 Mbaud	TCP/IP
CP1512、SoftNet7 PCMCIA卡（用于笔记本电脑）	10 Mbaud或 100 Mbaud	TCP/IP

<sup>1</sup> 多台主设备电缆提供了RS-485端口（S7-200 CPU上）和连接到计算机的端口之间的电气绝缘。使用不绝缘的RS-485到RS-232转换器有可能损坏计算机的RS-232端口。

## 7

## 在网络上使用HMI设备

S7-200 CPU支持来自西门子以及其他厂商的许多型号的HMI设备。当这些HMI设备中的某些设备（例如TD 200或TP 070）不允许选择设备所使用的通讯协议时，其他设备（例如OP7和TP170）将允许选择该设备的通讯协议。

如果HMI设备允许选择通讯协议，请考虑下列原则：

- 对于与S7-200 CPU通讯端口相连接的HMI设备，当网络上没有任何其他设备时，可以为HMI设备选择PPI协议或MPI协议。
- 对于与EM277 PROFIBUS模块相连接的HMI设备，即可选择MPI协议，也可选择PROFIBUS协议。
  - 如果具有HMI设备的网络包含有S7-300或S7-400 PLC，则为HMI设备选择MPI协议。
  - 如果具有HMI设备的网络是一个PROFIBUS网络，则为HMI设备选择PROFIBUS协议，并选择与PROFIBUS网络上其它主设备相容的配置文件。
- 对于与已经配置为主设备的S7-200 CPU通讯端口相连接的HMI设备，为HMI设备选择PPI协议。“PPI高级协议”是最适当的。MPI和PROFIBUS协议不支持S7-200 CPU作为主设备。

## 创建具有自由端口模式的自定义协议

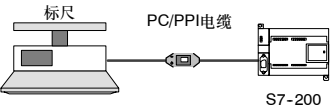
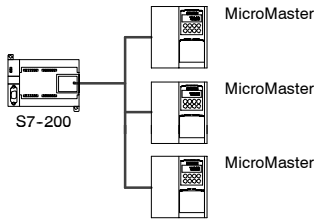

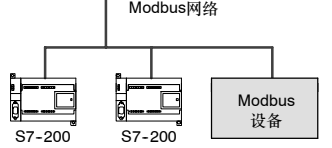

自由端口模式将使程序能够控制S7-200CPU的通讯端口。可使用自由端口模式来实现自定义通讯协议，以便与多种类型的智能设备进行通讯。自由端口模式支持ASCII协议和二进制协议。

为启用自由端口模式，可使用特殊内存字节SMB30（适用于端口0）和SMB130（适用于端口1）。程序将使用下列方法来控制通讯端口的操作：

- ❑ “传输”指令（XMT）和传输中断：“传输”指令允许S7-200从COM端口传输多达255个字符。传输完成后，传输中断将通知S7-200中的程序。
- ❑ 接收字符中断：接收字符中断将通知用户程序，COM端口上的字符已经接收完毕。程序于是基于使用的协议，对该字符做出反应。
- ❑ 接收指令（RCV）：接收指令接收COM端口的整条讯息，然后在完全接收到讯息后，生成程序中中断。可使用S7-200的SM存储器来配置接收指令，用于在已定义的环境下，启动和停止讯息的接收。接收指令将使程序能够启动或停止基于特定字符或时间周期的讯息。大多数协议均可通过接收指令来完成。

只有在S7-200处于RUN（运行）模式时，才能激活自由端口模式。将S7-200设置为STOP（停止）模式将暂停所有的“自由端口”通讯，而通讯端口随后将返回到具有S7-200系统块所配置设置的协议。

表7-9 使用自由端口模式

网络配置		描述
使用通过RS-232连接的自由端口		<p>实例：使用电子标尺带RS-232端口的S7-200。</p> <ul style="list-style-type: none"> <li>■ RS-232/PPI多台主设备电缆将标尺上的RS-232端口连接到S7-200 CPU上的RS-485端口。（将电缆设置为PPI/自由端口模式，5号开关=0。）</li> <li>■ S7-200 CPU使用自由端口与天平进行通讯。</li> <li>■ 波特率可为1200 baud到115.2 kbaud。</li> <li>■ 用户程序将对协议进行定义。</li> </ul>
使用USS协议		<p>实例：使用带有SIMODRIVE MicroMaster驱动器的S7-200。</p> <ul style="list-style-type: none"> <li>■ STEP 7-Micro/WIN提供了一个USS库。</li> <li>■ S7-200 CPU是主设备，而驱动器是从属装置。</li> </ul> <p> 请参见文档光盘范例 USS 程序的编程提示。参见提示 28。</p> <p>编程提示</p>
创建模拟另一网络上从属装置的用户程序		<p>实例：将S7-200 CPU连接到Modbus网络。</p> <ul style="list-style-type: none"> <li>■ S7-200中的用户程序模拟Modbus从属装置。</li> <li>■ STEP 7-Micro/WIN提供Modbus库。</li> </ul> <p> 请参见文档光盘范例 Modbus 程序的编程提示。参见提示 41。</p> <p>编程提示</p>

## 使用RS-232/PPI多台主设备电缆和具有RS-232设备的自由端口模式

可使用RS-232/PPI多台主设备电缆和自由端口通讯功能将S7-200 CPU连接到与RS-232标准兼容的许多设备。必须将电缆设置为用于自由端口操作的PPI/自由端口模式（5号开关=0）。6号开关既可选择为本地模式（DCE）（6号开关=0），也可选择为远程模式（DTE）（6号开关=1）。

数据从RS-232端口传输到RS-485端口时，RS-232/PPI多台主设备电缆处于“传输”模式。电缆在闲置或将数据从RS-485端口传输到RS-232端口时，处于“接收”模式。电缆检测到RS-232传输行上有字符时，立即从“接收”模式切换到“传输”模式。

RS-232/PPI多台主设备电缆支持1200 baud与115.2kbaud之间的波特率。使用RS-232/PPI多台主设备电缆外壳上的DIP开关，可配置恰当的电缆波特率。表7-10显示了波特率和开关位置。

当RS-232传输线在闲置状态下闲置一段定义为电缆周转时间的时间周期之后，电缆将重新切换到“接收”模式。电缆的波特率选择将确定周转时间，如表7-10所示。

如果在使用了自由端口通讯的系统中，正在使用RS-232/PPI多台主设备电缆，则S7-200中的程序必须包含下列情形下的周转时间：

表7-10 周转时间和设置

波特率	周转时间	设置（1=向上）
115200	0.15ms	110
57600	0.3ms	111
38400	0.5ms	000
19200	1.0ms	001
9600	2.0 ms	010
4800	4.0 ms	011
2400	7.0 ms	100
1200	14.0 ms	101

- S7-200将响应由RS-232设备所传输的讯息。

在S7-200接收到来自RS-232设备的请求讯息之后，S7-200必须将响应讯息的传输延迟一段时间，这段时间应大于或等于电缆的周转时间。

- RS-232设备将响应从S7-200传输的讯息。

在S7-200接收到来自RS-232设备的请求讯息之后，S7-200必须将下一个请求讯息的传输延迟一段时间，这段时间应大于或等于电缆的周转时间。

在上面两种情况中，延迟将使RS-232/PPI多台主设备电缆具有充足的时间从“传输”模式切换到“接收”模式，以便可将数据从RS-485端口传输到RS-232端口。



## 在网络中使用调制解调器和STEP 7-Micro/WIN

STEP 7-Micro/WIN V.2将使用标准的Windows电话和调制解调器选项，用于选择和配置电话调制解调器。电话和调制解调器选项均位于Windows的控制面板中。使用调制解调器的Windows安装选项，可以：

- 使用Windows所支持的大多数内置和外置调制解调器。
- 使用Windows所支持的大多数调制解调器的标准配置。
- 使用标准的Windows拨号规则，用于选择所在位置、国家和区域代码支持、脉冲或音频拨号以及呼叫卡支持。
- 在与EM 241调制解调器模块进行通讯时使用更高的波特率。

使用Windows控制面板来显示“调制解调器属性”对话框。该对话框将允许对本地调制解调器进行配置。可从Windows所支持的调制解调器列表中选择调制解调器。如果在Windows调制解调器对话框中没有您的调制解调器的型号，则可选择与您的调制解调器最相匹配的型号，或与您的调制解调器销售商联系，以获得调制解调器的Windows配置文件。

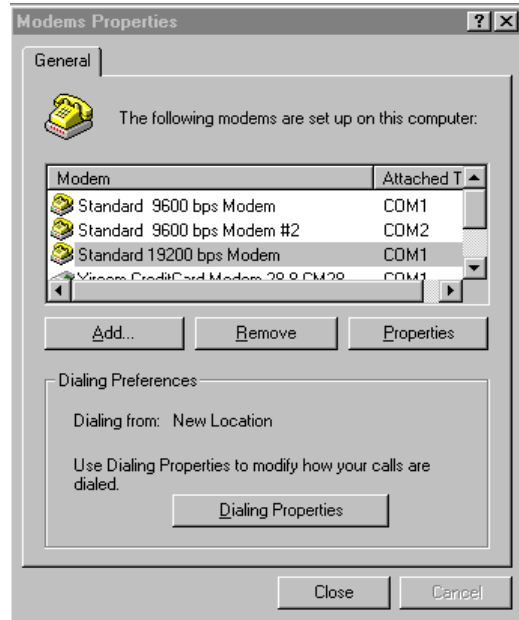


图7-23 配置本地调制解调器

STEP 7-Micro/WIN还允许使用无线电和手机调制解调器。这些类型的调制解调器将不显示在“Windows调制解调器属性”对话框中，但在配置STEP 7-Micro/WIN连接时可用。

## 配置调制解调器连接

连接将识别名称与连接的物理属性相关联。对于电话调制解调器，这些属性包括调制解调器型号、10或11位的协议选择以及超时。对于手机调制解调器，连接将允许对PIN和其他参数进行设置。无线电调制解调器属性包括波特率的选择、奇偶校验、流量控制和其他参数。



连接向导

### 添加连接

使用连接向导可如图7-24所示添加新连接、删除连接或编辑连接。

1. 双击“通讯设置”窗口中的图标。
2. 双击PC/PPI电缆以打开PG/PC接口。选择PPI电缆并单击“属性”按钮。在“本地连接”标签上，勾选“调制解调器连接”框。
3. 双击“通讯”对话框中的调制解调器连接图标。
4. 单击“设置”按钮，以显示“调制解调器连接设置”对话框。
5. 单击“添加”按钮，启动“添加调制解调器连接”向导。
6. 按照向导的提示配置连接。

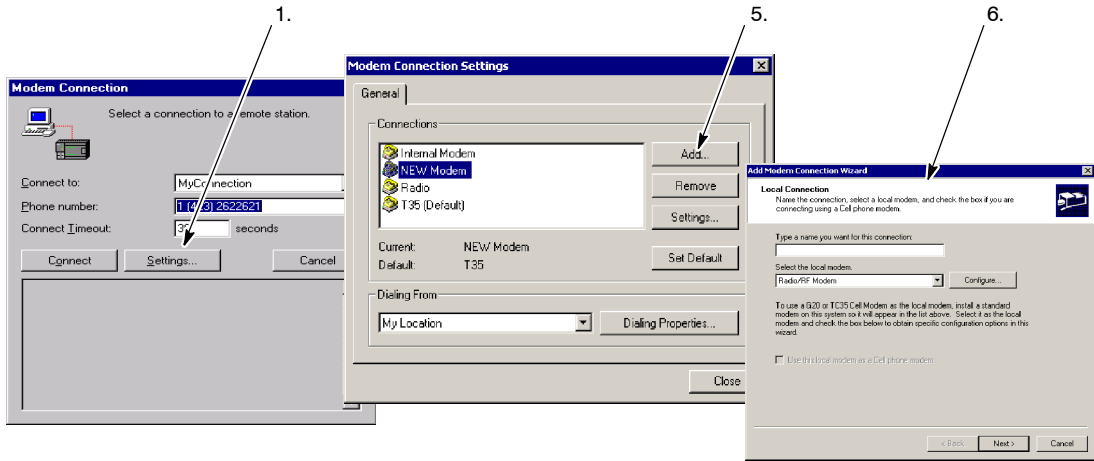


图7-24 添加调制解调器连接

## 通过调制解调器连接到S7-200

在添加调制解调器连接之后，可连接到S7-200 CPU。

1. 打开“通讯”对话框，双击“连接”图标来显示“调制解调器连接”对话框。
2. 在“调制解调器连接”对话框中，单击“连接”，使调制解调器开始拨号。

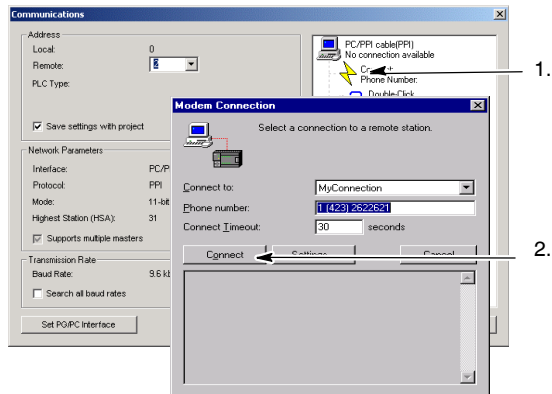


图7-25 连接到S7-200



### 配置远程调制解调器

远程调制解调器是与S7-200相连接的调制解调器。如果远程调制解调器为EM 241调制解调器模块，则不需要进行任何配置。如果连接的是单机调制解调器或手机调制解调器，则必须对连接进行配置。

调制解调器扩充向导将对连接到S7-200 CPU的远程调制解调器进行配置。为了与S7-200 CPU的RS-485半双工端口进行正确通讯，需要特殊的调制解调器配置。这只需要选择调制解调器的型号，并按照向导的提示输入信息。更详细情况，参见联机帮助。

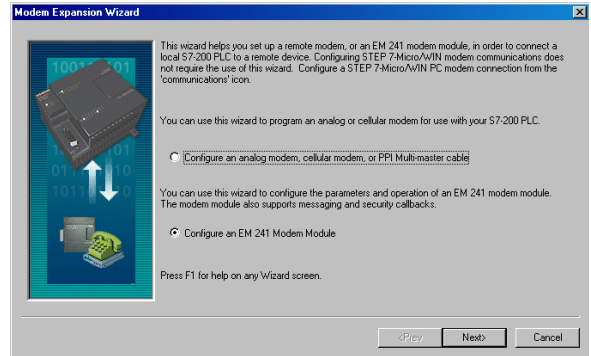


图7-26 调制解调器扩充向导

### 配置PPI多台主设备电缆以使用远程调制解调器

RS-232 PPI多台主设备电缆具有在电缆加电时发送调制解调器AT命令字符串的能力。请注意，该配置只有在必须修改默认的调制解调器设置时才需要。参见图7-27。

调制解调器命令可在常规命令中进行指定。自动响应命令将是唯一的缺省设置。

手机授权命令和PIN号码可在“手机授权”域中指定，例如+CPIN=1234。

每个命令串都将分别发送给调制解调器。每个串的前面均将放置AT调制解调器注意命令。

通过选择“程序/测试”按钮，可以在电缆中初始化这些命令。

请注意，位图将描绘所推荐的开关设置，这些设置取决于所选参数。

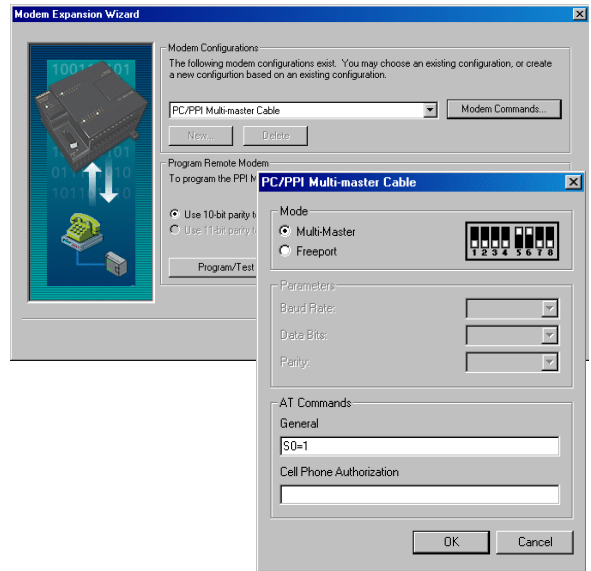


图7-27 调制解调器扩充向导 - 发送调制解调器命令

在使用STEP 7-Micro/WIN配置RS-232/PPI多台主设备电缆时，必须将RS-485连接器连接到S7-200 CPU。这是接通电缆所需24V电源的来源。请确保为S7-200 CPU提供电源。

在退出RS-232/PPI多台主设备电缆的STEP 7-Micro/WIN配置之后，请断开与PC的电缆，并将其连接到调制解调器。电源将循环接通调制解调器和电缆。现在准备在PPI多台主设备网络中使用电缆进行远程操作。



#### 提示

调制解调器必须为工厂缺省设置，以便使用PPI多台主设备电缆。

## 配置PPI多台主设备电缆以使用自由端口

RS-232PPI多台主设备电缆提供了发送调制解调器AT命令串的能力，该能力与为“自由端口”模式所配置电缆的能力相同。请注意，该配置只有在必须修改默认的调制解调器设置时才需要。

然而，还必须将电缆配置为与S7-200端口的波特率、奇偶校验以及数据位数等相匹配。这在S7-200应用程序将控制这些参数的配置时需要。

可选择1.2kbaud和115.2kbaud之间的波特率。

可选择七个或八个数据位。

可选择偶校验、奇校验或无奇偶校验。

请注意，位图将描绘所推荐的开关设置，这些设置将取决于所选择的参数。

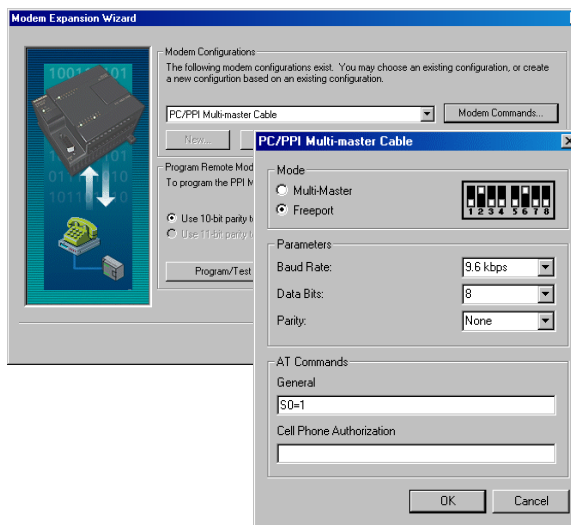


图7-28 调制解调器扩充向导-在自由端口模式下发送调制解调器命令

7

在使用STEP 7-Micro/WIN配置RS-232/PPI多台主设备电缆时，必须将RS-485连接器连接到S7-200 CPU。这是接通电缆所需24V电源的来源。请确保为S7-200 CPU提供电源。

在退出RS-232/PPI多台主设备电缆的STEP 7-Micro/WIN配置之后，请断开与PC的电缆，并将其连接到调制解调器。电源将循环接通调制解调器和电缆。此时即可准备使用PPI多台主设备网络中的远程操作电缆。



### 提示

调制解调器必须为工厂缺省设置，以便使用PPI多台主设备电缆。

### 使用具有RS-232/PPI多台主设备电缆的电话调制解调器

可使用RS-232/PPI多台主设备电缆将调制解调器的RS-232通讯端口连接到S7-200 CPU。参见图7-29。

- 1号开关、2号开关和3号开关用于设置波特率。
- 5号开关用于选择PPI或PPI/自由端口模式。
- 6号开关用于选择本地（等同于数据通讯设备-DCE）或远程（等同于数据终端装置-DTE）模式。
- 7号开关用于选择10位或11位PPI协议。

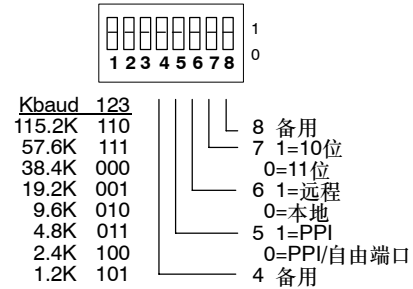


图7-29 用于RS-232/PPI多台主设备电缆的设置

5号开关用于选择运行于PPI模式，还是运行于PPI/自由端口模式。如果使用STEP 7-Micro/WIN通过调制解调器与S7-200进行通讯，则选择PPI模式（5号开关=1）。否则，选择PPI/自由端口模式（5号开关=0）。

RS-232/PPI多台主设备电缆的7号开关将为PPI/自由端口模式选择10位或11位模式。只有在通过处于PPI/自由端口模式的调制解调器将S7-200连接到STEP 7-Micro/WIN时，才使用7号开关。否则，将7号开关设置为11位模式，以确保其他设备的正确操作。

RS-232/PPI多台主设备电缆的6号开关将允许把电缆的RS-232端口设置为本地（DCE）模式或远程（DTE）模式。

- 如果使用的是具有STEP 7-Micro/WIN的RS-232/PPI多台主设备电缆，或RS-232/PPI多台主设备电缆已连接到计算机，则将RS-232/PPI多台主设备电缆设置为局部（DCE）模式。
- 如果使用的是具有调制解调器（它属于DCE设备）的RS-232/PPI多台主设备电缆，则将RS-232/PPI多台主设备电缆设置为远程（DTE）模式。

这就排除了在RS-232/PPI多台主设备电缆和调制解调器之间安装空转调制解调器适配器的需要。根据调制解调器上连接器的不同，可能仍然需要使用9针对25针的适配器。

图7-30表示了通用调制解调器适配器的插针分配。

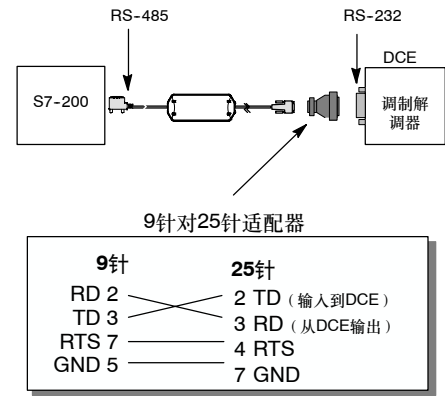


图7-30 适配器的插针分配

RS-232/PPI多台主设备电缆的更详细信息，请参见附录A。处于本地（DCE）模式的RS-232/PPI多台主设备电缆RS-485和RS-232端口的插针号及其功能均显示在表A-64中。表A-65显示了处于远程（DTE）模式的RS-232/PPI多台主设备电缆RS-485和RS-232端口的插针号及其功能。RS-232/PPI多台主设备电缆只有在处于远程（DTE）模式时才提供RTS。

## 使用具有RS-232/PPI多台主设备电缆的无线电调制解调器

可使用RS-232/PPI多台主设备电缆将无线电调制解调器的RS-232通讯端口连接到S7-200 CPU。然而，无线电调制解调器的操作不同于电话调制解调器。

### PPI模式

使用为PPI模式（5号开关=1）所设置的RS-232/PPI多台主设备电缆。一般选择远程模式（6号开关=1）用于调制解调器的操作。然而，选择远程模式将导致电缆发送字符串“AT”，并在每次加电时等待调制解调器响应。当电话调制解调器按照该顺序确定波特率时，无线电调制解调器通常不接受AT命令。

因此，为了操作无线电调制解调器，必须选择本地模式（6号开关=0），并在电缆的RS-232连接器与无线电调制解调器的RS-232端口之间使用一个空转的调制解调器适配器。空转调制解调器适配器在9针对9针或9针对25针配置下均可使用。

配置无线电调制解调器，以便在9.6、19.2、38.4、57.6或115.2 kbaud下运行。RS-232/PPI多台主设备电缆在无线电调制解调器传输第一个字符时，将自动调整到的这些波特率中的任一个。

### PPI/自由端口模式

使用为PPI/自由端口模式（5号开关=0）所设置的RS-232/PPI多台主设备电缆，可选择用于无线电调制解调器操作的远程模式（6号开关=1）。对电缆进行配置，以便其不发送任何AT命令来设置调制解调器。

RS-232/PPI多台主设备电缆上的开关1、2、3将对波特率进行设置。参见图7-29。选择与PLC和无线电调制解调器波特率相对应的波特率设置。

## 高级主题

### 优化网络性能

下列因素将影响网络性能（效果最佳的波特率和主设备数目）：

- 波特率：以所有设备都支持的最高波特率操作网络将具有最佳的效果。
- 网络上的主设备数：减少网络上的主设备数目也可改善网络的性能。网络上的每个主设备都将要求增加网络开销；减少主设备可减轻开销。
- 主设备和从属装置地址的选择：应对主设备的地址进行设置，以便所有主设备的地址都是连续的，地址之间没有任何间隙。无论主设备之间何时存在地址间隙，主设备都将不断检查地址间隙，以确定是否有另一个主设备即将联机。这种检查将需要时间，并增加网络的开销。如果主设备之间不存在任何地址间隙，则不进行任何检查，从而减小了开销。可将从属装置地址设置为不影响网络性能的任何值，只要从属装置没有位于主设备之间。主设备之间的从属装置与主设备一样，如果存在地址间隙，将增加网络开销。
- 间隙刷新因子（GUF）：只有在S7-200 CPU当作PPI主设备运行时才使用，GUF告诉S7-200隔多久检查其它主设备的地址间隙。可使用STEP 7-Micro/WIN来设置CPU配置中用于CPU端口的GUF。这将把S7-200配置为只定时检查地址间隙。对于GUF=1，S7-200将在每一次持有令牌时检查地址间隙；对于GUF=2，S7-200将在每两次持有令牌时检查地址间隙。如果主设备之间存在地址间隙，则设置更高的GUF将可减小网络开销。如果主设备之间不存在任何地址间隙，则GUF对性能将不产生任何影响。设置过高的GUF数值将导致主设备联机成功的时间产生较大的延迟，因为并不经常对地址进行检查。默认的GUF设置为10。
- 最高站址（HSA）：只有在S7-200 CPU当作PPI主设备运行时才使用，HSA定义了某个主设备寻找另一个主设备时的最高地址。可使用STEP 7-Micro/WIN来设置CPU配置中用于CPU端口的HSA。设置HSA将限制网络中最后一个主设备（具有最高地址）所必须检查的地址间隙。限制地址间隙的大小将减少对另一个主设备进行查找和联机时所需要的时间。最高站址对从属装置地址没有任何影响：主设备可仍然与地址大于HSA的从属装置进行通讯。通常，将所有主设备上的最高站址都设置为同一个数值。该地址应大于或等于最高主设备地址。HSA的默认值为31。

#### 计算网络的令牌轮转时间

在令牌传递网络中，可启动通讯的唯一一站就是持有令牌的站。令牌轮转时间（令牌轮转到逻辑环中的每个主设备所需要的时间）衡量网络的性能。

图7-31提供了一个范例网络，作为计算多台主设备网络的令牌轮转时间的实例。在该实例中，TD 200（站3）与CPU 222（站2）进行通讯，TD 200（站5）与CPU 222（站4）进行通讯，依此类推。两个CPU 224模块都使用“网络读取”和“网络写入”指令来收集来自其它S7-200的数据：CPU 224（站6）发送讯息给站2、4和8，CPU 224（站8）发送讯息给站2、4和6。在该网络中，存在有六个主设备站（四个TD 200单元和两个CPU 224模块）和两个从属装置站（两个CPU 222模块）。



提示与技巧

请参见文档光盘编程提示中关于令牌轮转的讨论。参见提示 42。

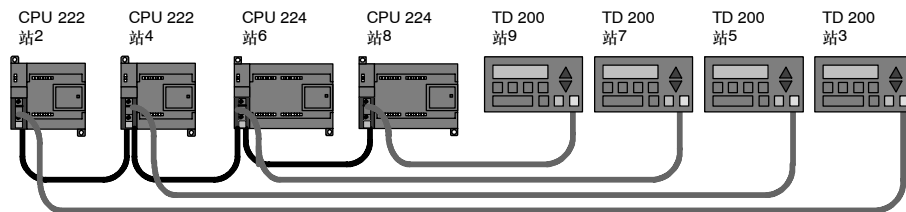


图7-31 令牌传递网络的实例

主设备为了发送讯息，必须持有令牌。例如：当站3具有令牌时，它将启动对站2的请求讯息，然后将令牌传递给站5。站5随后启动对站4的请求讯息，然后将令牌传递给站6。站6随后启动给站2、站4或站8的讯息，并将令牌传递给站7。启动讯息和传递令牌的这些过程将继续沿着逻辑环从属装置3到站5、站6、站7、站8、站9，最后回到站3。令牌必须完全沿着逻辑环进行轮转，以便主设备能够发送对信息的请求。对于具有六个站的逻辑环，如果每一持有的令牌发送一条请求讯息，以读或写一个双数值（四个字节的数据），则令牌轮转时间在9600波特下大约为900ms。增加每一讯息所访问的数据字节数或增加站数都将延长令牌轮转时间。

令牌轮转时间取决于各个站持有令牌的时间。将各个站持有令牌的时间相加，即可确定多台主设备网络的令牌轮转时间。如果PPI主设备模式已经启用（在网络上的PPI协议下），则通过使用S7-200的“网络读取”和“网络写入”指令，可将讯息发送给其它S7-200。如果使用这些指令发送讯息，则在下列假设的基础上，可使用下列公式来计算近似的令牌轮转时间：每个站每持有有一个令牌只发送一个请求；请求可以是连续数据单元的读或写请求；在使用S7-200中的某个通讯缓冲区时不存在任何冲突；不存在任何需要10ms以上扫描时间的S7-200。

令牌持有时间 ( $T_{\text{hold}}$ ) = (平均128+n数据字符) × 11位/字符 × 1/波特率
令牌轮转时间 ( $T_{\text{rot}}$ ) = 主设备1的 $T_{\text{hold}}$ + 主设备2的 $T_{\text{hold}}$ + ... + 主设备m的 $T_{\text{hold}}$
这里 $n$ 为数据字符的数目（字节数） $m$ 为主设备数

下列等式计算了轮转时间（一个“位时间”等于一个信号周期的持续时间），其实例如图7-31所示：

$$\begin{aligned}
 T(\text{令牌持有时间}) &= (128+4\text{字符}) \times 11\text{位/字符} \times 1/9600\text{位时间/s} \\
 &= 151.25\text{ ms/主设备} \\
 T(\text{令牌轮转时间}) &= 151.25\text{ ms/主设备} \times 6\text{主设备} \\
 &= 907.5\text{ ms}
 \end{aligned}$$



#### 提示

SIMATIC NET COM PROFIBUS软件提供了确定网络性能的分析仪。



## 比较令牌轮转时间

表7-11显示了令牌轮转时间与站数、数据量以及波特率之间的对比关系。当在S7-200 CPU或其他主设备下使用“网络读取”和“网络写入”指令时，可用图表来表示时间。

表7-11 令牌轮转时间（以秒为单位）

波特率	所传送的字节数	主设备数								
		2	3	4	5	6	7	8	9	10
9.6kbaud	1	0.30	0.44	0.59	0.74	0.89	1.03	1.18	1.33	1.48
	16	0.33	0.50	0.66	0.83	0.99	1.16	1.32	1.49	1.65
19.2kbaud	1	0.15	0.22	0.30	0.37	0.44	0.52	0.59	0.67	0.74
	16	0.17	0.25	0.33	0.41	0.50	0.58	0.66	0.74	0.83
187.5kbaud	1	0.009	0.013	0.017	0.022	0.026	0.030	0.035	0.039	0.043
	16	0.011	0.016	0.021	0.026	0.031	0.037	0.042	0.047	0.052

## 了解链接网络设备的连接

网络设备将通过单个的连接进行通讯，这些连接均是主设备和从属装置之间的专用连接。如图7-32所示，通讯协议在连接的处理方式上有所不同：

- PPI协议利用了一个所有网络设备的共享连接。
- PPI高级协议、MPI和PROFIBUS协议利用了相互进行通讯的任意两个设备之间的单独连接。

当使用PPI高级协议、MPI或PROFIBUS时，第二个主设备将不能干涉主设备与从属装置之间已经建立的连接。S7-200 CPU和EM277始终保留一个用于STEP 7-Micro/WIN的连接和一个用于HMI设备的连接。其他主设备不能使用这些保留的连接。这确保在主设备使用支持连接的协议（如PPI高级协议）时，始终可以将至少一个编程站和至少一个HMI设备连接到S7-200 CPU或EM277。

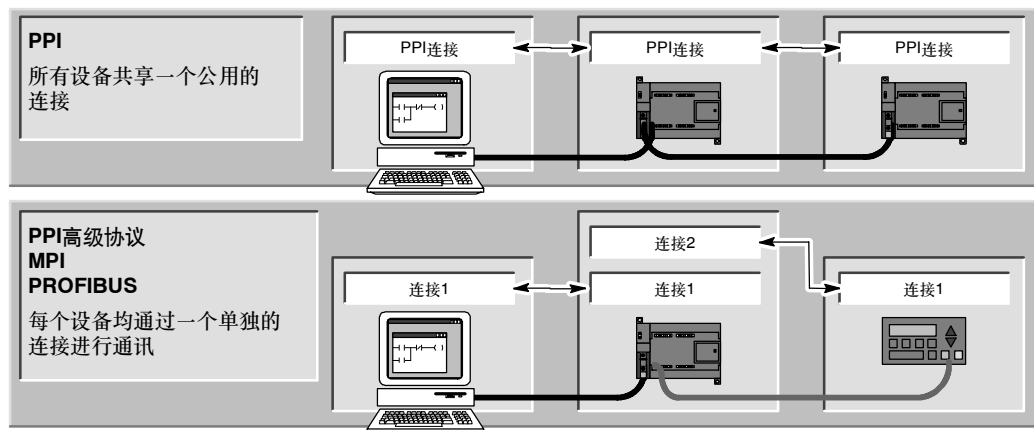


图7-32 管理通讯连接

如表7-12所示，S7-200 CPU或EM 277将提供特定数量的连接。S7-200的每个端口（端口0和端口1）最多可支持四个单独的连接。（这将允许S7-200 CPU最多具有八个连接。）这包含共享的PPI连接。EM 277支持六个连接。

表7-12 S7-200 CPU和EM 277模块的容量

连接点	波特率	连接	STEP 7-Micro/WIN协议配置文件选择	
S7-200 CPU	端口0	9.6 kbaud、 19.2 kbaud或 187.5 kbaud	4	PPI、PPI高级协议、MPI和PROFIBUS <sup>1</sup>
	端口1	9.6 kbaud、 19.2 kbaud或 187.5 kbaud	4	
EM 277模块	9.6kbaud到12 kbaud	每个模块6个	PPI高级协议、MPI和PROFIBUS	

<sup>1</sup> 如果使用CP卡将STEP 7-Micro/WIN通过端口0或端口1连接到S7-200 CPU，则只有在将S7-200配置为从属装置时，才可选择MPI或PROFIBUS配置文件。

## 7

## 使用复杂网络

对于S7-200，复杂网络的一个典型特点就是具有多个S7-200主设备，这些主设备将使用“网络读取（NETR）”和“网络写入（NETW）”指令与PPI网络中的其他设备进行通讯。特别地，复杂网络将存在一些特殊问题，这些问题可能使主设备中断与从属装置的通讯。

如果网络以较低的波特率运行（例如9.6 kbaud或19.2 kbaud），则在传递令牌之前，每个主设备将完成事务处理（读或写）。然而，如果波特率为187.5 kbaud，则主设备将对从属装置发出请求，然后传递令牌，它将使未完成的请求留在从属装置上。

图7-33表示了一个具有潜在通讯冲突的网络。在该网络中，“站1”、“站2”和“站3”均是主设备，它们将使用“网络读取”或“网络写入”指令与“站4”进行通讯。“网络读取”和“网络写入”指令使用PPI协议，这样，所有S7-200均将共享“站4”中的单个PPI连接。

在本实例中，“站1”将发出对“站4”的请求。对于19.2kbaud以上的波特率，“站1”将令牌传递给“站2”。如果“站2”试图发出对“站4”的请求，则“站2”的请求将被拒绝，因为“站1”的请求仍然存在。对“站4”的所有请求都将被拒绝，直到“站4”完成对“站1”的响应。只有在响应已经完成之后，另一个主设备才能发出对“站4”的请求。

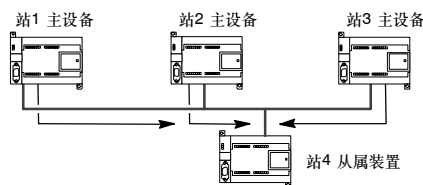


图7-33 通讯冲突

为避免“站4”通讯端口的该冲突，如图7-34所示，应考虑使“站4”成为网络上的唯一主设备。“站4”随后即可发出对其它S7-200的读/写请求。

这种配置不但可确保通讯中不存在任何冲突，而且也可减少由于具有多台主设备而导致的额外开销，并可使网络运行更为有效。

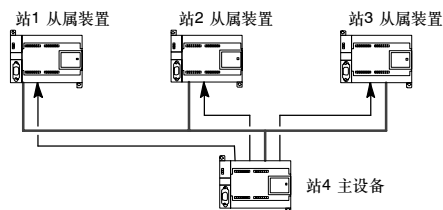


图7-34 避免冲突

然而，对于某些应用场合，不能随意选择减少网络中的主设备数量。当存在多个主设备时，必须对令牌轮转时间进行管理，确保网络不超出目标令牌轮转时间。（令牌轮转时间就是从主设备传递令牌开始到主设备又重新收到令牌为止所花费的总时间。）

表7-13 HSA和目标令牌轮转时间

HSA	9.6kbaud	19.2kbaud	187.5kbaud
HSA=15	0.613s	0.307s	31 ms
HSA=31	1.040s	0.520s	53 ms
HSA=63	1.890s	0.950s	97 ms
HSA=126	3.570s	1.790s	183 ms

如果令牌返回到主设备所需要的时间大于目标令牌轮转时间，则将不允许主设备发出请求。只有在实际令牌轮转时间低于目标令牌轮转时间时，主设备才可发出请求。

S7-200的最高站址（HSA）和波特率设置决定了目标令牌轮转时间。表7-13列出了目标轮转时间。

对于较低的波特率，例如9.6kbaud和19.2kbaud，主设备在传递令牌之前，将等待对其请求的响应。因为按照扫描时间，处理请求/响应轮转将要花费相对较长的时间，所以，每当网络上的每个主设备握有令牌时，它们极有可能已经具有已准备就绪的传送请求。这样，实际的令牌轮转时间将增加，且某些主设备将有可能不能处理任何请求。在某些情况下，或许完全不允许主设备对请求进行处理。

*例如：*一个具有10个主设备的网络，该网络以9.6 kbaud的波特率传输1个HSA配置为15的字节。对于本实例，每个主设备将始终具有准备发送的讯息。如表7-13所示，该网络的目标轮转时间为0.613秒。然而，基于表7-11所列出的性能数据，该网络所需要的实际令牌轮转时间将为1.48秒。因为实际的令牌轮转时间大于目标令牌轮转时间，所以，在稍后的令牌轮转之前将不允许某些主设备传输讯息。

改进实际令牌轮转时间大于目标令牌轮转时间这种状况，有两种基本的选择：

- 通过减少网络上的主设备数目，可以缩短实际令牌轮转时间。随着应用场合变化，这可能不是可行的解决方案。
- 通过增加网络上的所有主设备的HSA，可以增加目标令牌轮转时间。

增加HSA可能引起网络的其它问题，因为这影响S7-200切换到主设备模式并进入网络所占用的总时间。如果使用计时器来确保在指定时间内完成“网络读取”或“网络写入”指令的执行，则在启动主设备模式并将S7-200添加为网络中的主设备期间的延迟将可能导致提示出现超时。通过减小网络上所有主设备的间隙刷新因子（GUF），可最大限度减小添加主设备所产生的延迟。

由于以187.5kbaud将请求发送并保留在从属装置上所采取的方式，在选择目标令牌轮转时间时应留出多余的时间。对于187.5kbaud，实际的令牌轮转时间应大约为目标令牌轮转时间的一半。

为确定令牌轮转时间，请使用表7-11中的性能数据来确定完成“网络读取”和“网络写入”操作所需要的时间。为计算HMI设备（例如TD 200）所需要的时间，请使用用于传送16个字节的性能数据。通过将网络上每台设备的时间加在一起来计算令牌轮转时间。把描述最坏情况下的所有时间加在一起，在最坏情况下，所有设备在同一令牌轮转期间都希望处理一个请求。这定义网络所需要的最大令牌轮转时间。

例如: 假设具有四个TD 200和四个S7-200的网络以9.6kbaud运行, 每个S7-200每秒将10个字节的数据写入另一个S7-200。请使用表7-11来计算网络的特定传送时间:

4个TD 200设备传送16字节的数据=	0.66 s
4个S7-200传送10字节的数据=	0.63 s
总的令牌轮转时间=	1.29 s

为允许该网络有足够的时间来处理一个令牌轮转期间的所有请求, 可将HSA设置为63。(参见表7-13。)选择目标令牌轮转(1.89 s)大于最大令牌轮转时间(1.29 s), 确保每个设备在令牌的每个轮转中都可传送数据。

为帮助改进多台主设备网络的可靠性, 还应考虑下列操作:

- 改变HMI设备的更新速率, 以便允许两次更新之间有更多的时间。例如, 将TD 200的更新速率从“尽可能快”改变为“每秒一次”。
- 对“网络读取”操作或“网络写入”操作进行组合, 减少请求数量(以及处理请求时的网络开销)。例如, 不使用各自读取4字节的两个“网络读取”操作, 而使用一个读取8字节的“网络读取”操作。处理两个4字节的请求所需要的时间将远远大于处理一个8字节的请求所需要的时间。
- 改变S7-200主设备的更新速率, 以使其不试图更新地比令牌轮转时间快。

## 配置用于远程操作的RS-232/PPI多台主设备电缆

### 超级终端作为配置工具

如果没有STEP 7-Micro/WIN可用于配置远程操作的RS-232/PPI多台主设备电缆，则可使用超级终端或任何其他哑元终端数据包。RS-232/PPI多台主设备电缆提供了内置菜单，用于指导远程操作电缆的配置。

当使用超级终端对RS-232/PPI多台主设备电缆进行配置时，必须将RS-485连接器与S7-200 CPU相连接。这是接通电缆所需24V电源的来源。请确保为S7-200 CPU提供电源。

为激活PC上的超级终端，单击开始 (Start) > 程序 (Programs) > 附件 (Accessories) > 通讯 (Communications) > 超级终端 (HyperTerminal)。

“超级终端”应用程序将运行并提示一个“连接描述”。必须为连接提供一个名称（例如多台主设备）。单击“确定”。可选择图标或接受新建连接所提供的默认图标。参见图7-35。



图7-35 超级终端连接描述

显示“连接到”屏幕。选择将要使用的通讯端口，单击“确定”按钮。所显示的下一个屏幕为“COMx属性”。接受默认值，单击“确定”。参见图7-36。

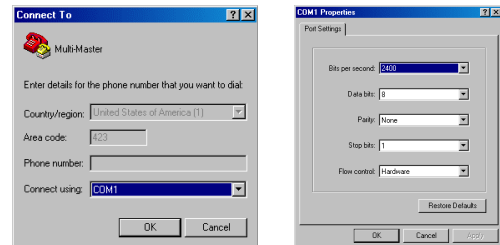


图7-36 “超级终端连接到”屏幕和“COMx属性”屏幕

单击“确定”之后，光标将放置在如图7-37所示的超级终端屏幕的编辑窗口中。请注意，超级终端窗口底部的状态栏表示已经连接，且正在运行的计时器表示连接的持续时间。

从菜单中选择调用 (Call) > 断开连接 (Disconnect)。状态栏此时将指示连接已经断开。

选择视图 (View) > 字体 (Font)。选择“Courier New”，单击“确定”。

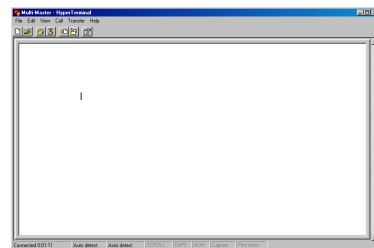


图7-37 多台主设备超级终端编辑窗口

选择文件 (File) > 属性 (Properties)。在“连接到”标签上, 单击配置... (Configure...) 按钮, 以显示通讯端口属性。参见图7-38。

在“COMx属性”对话框中, 选择下拉菜单中的波特率, 即每秒位数。必须选择每秒9600比特到115200比特之间的波特率 (通常选择9600)。使用相应的下拉菜单, 选择8个数据位、无奇偶校验、一个停止位和无流量控制。

单击“确定”, 返回到“连接到”标签。

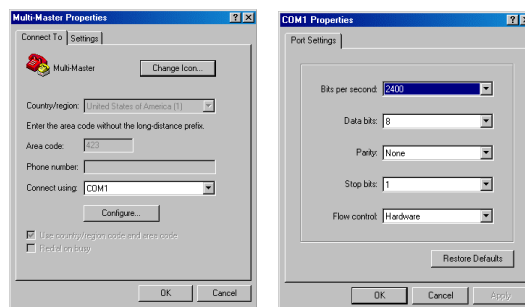


图7-38 多台主设备属性和COMx属性

选择“设置”标签。在“模拟”下拉菜单中选择ANSI, 单击“确定”。这将返回到超级终端屏幕的编辑窗口。屏幕底部的状态栏应显示:

“断开 ANSI 9600 8-N-1”  
如图7-39所示。

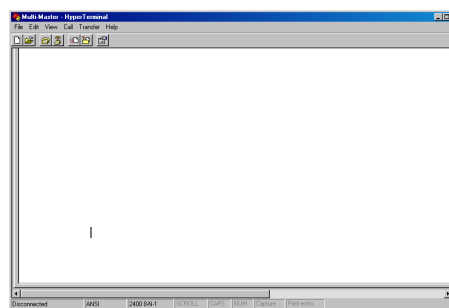


图7-39 超级终端编辑-断开连接ANSI

为启动与RS-232/PPI多台主设备电缆的通讯, 键入“hhh”。电缆上的Rx LED在键入“hhh”时应闪烁约一秒钟。电缆以语言选择进行响应时, TX LED短暂点亮。

输入对应于所选语言的编号 (使用退格键可取消默认选择), 按下回车键。图7-40显示语言选择显示以及“用于远程操作的RS-232/PPI电缆设置”选择显示。

该显示也显示了电缆的固件修订版。

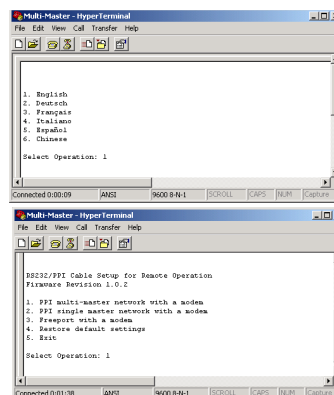


图7-40 超级终端语言选择和RS-232/PPI电缆设置

“用于远程操作的RS-232/PPI电缆设置”将指导按照所需要的步骤，给电缆配置期望的远程操作类型。

- 如果具有较早版本的STEP 7-Micro/WIN，选择第二个选项“带有调制解调器的PPI单台主设备网络”。
- 如果通过调制解调器使用自由端口通讯，选择第三个选项。

例如，使用STEP 7-Micro/WIN 3.2 Service Pack 4或更新的版本，为“带有调制解调器的PPI多台主设备网络”选择第一个选项。

如图7-41所示的超级终端显示将指出电缆上所需要设置的开关设置。这些开关设置将允许STEP 7-Micro/WIN通过调制解调器参与具有一个或多个主设备以及一个或多个S7-200 PLC的网络。类似的网络如图7-41所示。

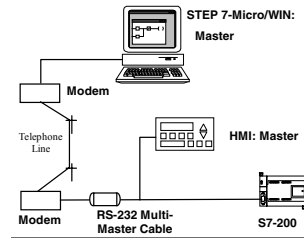
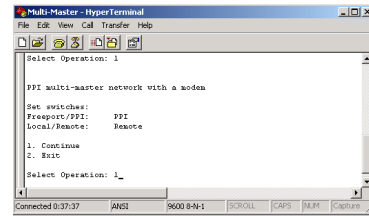


图7-41 超级终端-RS-232/PPI电缆设置

如图所示设置完开关后，选择继续。结果的超级终端显示如图7-42所示。

远程调制解调器（连接到RS-232/PPI多台主设备电缆的调制解调器）应设置为工厂的默认值。将远程调制解调器设置为工厂默认值之后，输入调制解调器编程所需要的AT串，用于RS-232/PPI多台主设备电缆的操作。典型地，需要发送的唯一串就是ATSO=1，它将把调制解调器配置为第一时间自动响应进入的调用。

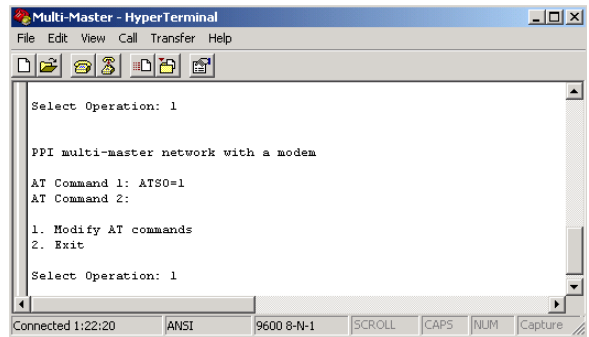


图7-42 超级终端-远程调制解调器

如果使用的是一个需要PIN的手机调制解调器，则使用第二个AT命令来提供PIN（参见调制解调器手册中调制解调器所支持的AT命令）。如果需要修改AT命令，可在出现这样的提示时进行选择，并输入所需要的命令。提示中包含有实例命令串，以便帮助进行命令的格式化。

RS-232/PPI多台主设备电缆在每次加电时，将把这些AT串发送给调制解调器。请确保在电缆加电之前或几乎在同一时间时，调制解调器也已加电。此外，如果调制解调器循环加电，则确保电缆也循环加电。这允许电缆正确配置调制解调器，并以可用的最高波特率运行。

图7-43中的超级终端显示表明了如何输入AT命令。如果在提示时不需要提供第二个AT命令，则可按下回车键。这样可返回去选择是修改AT命令还是退出。如果完成了AT命令的输入，则选择“退出”。

在退出RS-232/PPI多台主设备电缆的超级终端配置之后，请断开PC的电缆，并将其连接到调制解调器。电源将循环接通调制解调器和电缆。此时即可准备使用PPI多台主设备网络中的远程操作电缆。

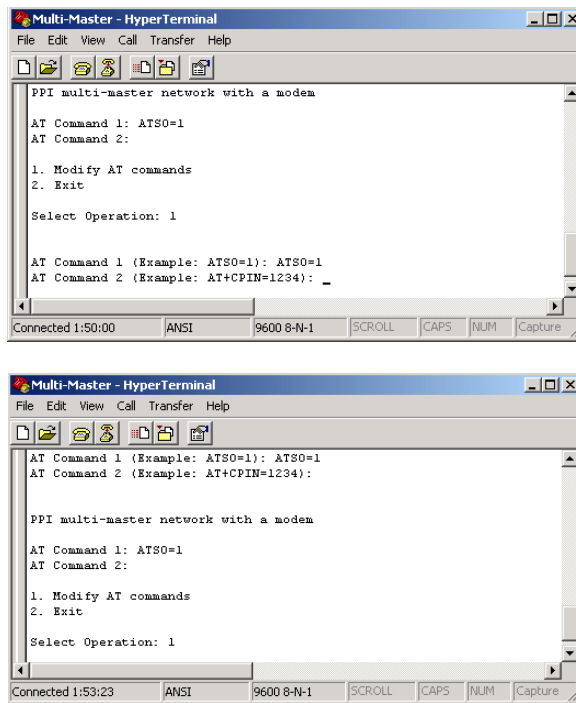


图7-43 超级终端-AT命令

#### 使用超级终端的自由端口操作

使用超级终端对用于自由端口操作的RS-232/PPI多台主设备电缆进行配置，将十分类似于上面所描述的实例配置。请按照提示根据需要来配置电缆。



# 8

## 硬件故障诊断指南和软件调试工具

STEP 7-Micro/WIN提供了软件工具用于帮助调试和测试程序。这些功能包括：在S7-200执行程序时查看程序状态、选择运行指定扫描数的S7-200以及强制数值。

当S7-200硬件在故障诊断方面有问题时，使用表8-1作为确定故障原因和可能的解决方案的指南。

### 本章内容

程序调试的功能 .....	246
显示程序状态 .....	248
使用状态图监控和修改S7-200中的数据 .....	249
强制指定值 .....	250
运行指定扫描数的程序 .....	250
硬件故障诊断指南 .....	251

## 调试程序的功能

STEP 7-Micro/WIN提供了一些帮助调试程序的功能：书签、交叉引用表和运行时间编辑。

### 使用书签简化程序访问

可以在程序中设置书签，从而能够很容易地在长程序的指定行（作了书签）之间前后移动。可以移动到程序中下一个或前一个作了书签的行。

### 使用交叉引用表检查程序引用



交叉引用

交叉引用表允许显示程序的交叉引用和元素用法信息。

交叉引用表识别在程序中使用的所有操作数，并识别程序块、程序段或行位置以及每次操作数使用时的关联指令。

可以在符号和绝对视图之间切换以改变所有操作数的表示法。

	Element	Block	Location	Context
1	I0.0	MAIN (OB1)	Network 1	- -
2	SMW32	MAIN (OB1)	Network 1	MOV_W
3	SMB31	MAIN (OB1)	Network 1	MOV_B
4	SM31.7	MAIN (OB1)	Network 1	- -
5	SM31.7	MAIN (OB1)	Network 1	-(S)

图8-1 交叉引用表



#### 提示

双击交叉引用表中的元素将转到用户程序或块的该部分。

## 8

### 在RUN（运行）模式中编辑程序

CPU 224 Rel. 1.10（以及更高版本）和CPU 226 Rel. 1.00（以及更高版本）型号支持RUN（运行）模式编辑。RUN（运行）模式编辑功能允许对用户程序进行小改动，从而实现对程序控制的进程干扰最小化。然而，执行此功能也允许大量的程序改变，这可能会引起中断或者甚至有危险。



#### 警告

当在RUN（运行）模式下下载改变到S7-200时，改变立即影响进程操作。在RUN（运行）模式中改变程序可能导致意外的系统操作，此操作可能会引起严重的人员伤亡或者设备损坏。

只有了解RUN（运行）模式编辑的系统操作的授权人员才能进行RUN（运行）模式编辑。

要在RUN（运行）模式下进行程序编辑，联机S7-200 CPU必须支持RUN（运行）模式编辑，并且必须处于RUN（运行）模式。

1. 选择调试（**Debug**）> 在运行模式中程序编辑（**Program Edit in RUN**）菜单命令。
2. 如果项目与S7-200中的程序不同，将提示用户进行保存。RUN（运行）模式编辑只能对S7-200的程序进行。
3. STEP 7-Micro/WIN警告用户在RUN（运行）模式下编辑程序，并提示继续或者取消操作。如果单击“继续”，STEP 7-Micro/WIN从S7-200上载程序。现在可以在RUN（运行）模式下编辑程序。对于编辑没有任何限制。



#### 提示

正向（EU）和负向（ED）转换指令与操作数一起显示。要查看关于边缘指令的信息，可在“视图”中选择“交叉引用”图标。“边缘应用”标签列出程序中边缘指令的数目。在编辑程序时，小心不要分配重复的边缘号码。

### 在RUN（运行）模式中下载程序

运行模式编辑允许当S7-200处于RUN（运行）模式时，仅下载程序块。在RUN（运行）模式中下载程序块前，考虑在下列状况下操作S7-200时运行模式更改的效果：

- 如果删除输出的控制逻辑，则S7-200保持输出的最后状态，直到下一个电源循环或转变到STOP（停止）模式。
- 如果删除正在运行的高速计数器或脉冲输出功能，则高速计数器或脉冲输出将继续运行，直到下一个电源循环或转变到STOP（停止）模式。
- 如果删除“附加中断”指令但没有删除中断例行程序，则S7-200将继续执行中断例行程序，直到电源循环或转变到STOP（停止）模式。同样，如果删除“分离中断”指令，则中断将不被关闭，直到下一个电源循环或转变到STOP（停止）模式。
- 如果添加以首次扫描位为条件的“附加中断”指令，则该事件将不被激活，直到下一个电源循环或从STOP（停止）转变到RUN（运行）模式。
- 如果删除“启用中断”指令，中断将继续操作，直到下一个电源循环或从RUN（运行）转变到STOP（停止）模式。
- 如果修改接收框的表格地址，且接收框在S7-200从旧程序切换到修改的程序时激活，则S7-200继续将接收的数据写到旧表格地址。“网络读取”和“网络写入”指令功能方式相同。
- 任何以首次扫描位状态为条件的逻辑将不执行，直到下一个电源循环或从STOP（停止）转变到RUN（运行）模式。首次扫描位仅通过转变到RUN（运行）模式设置，并且不受RUN（运行）模式编辑的影响。



#### 提示

在RUN（运行）模式中可以下载程序之前，S7-200必须支持RUN（运行）模式编辑，程序必须无错地编译，并且STEP 7-Micro/WIN和S7-200之间的通讯必须无错。

只能下载程序块。

要在RUN（运行）模式中下载程序，单击“下载”按钮或选择文件（**File**）> 下载（**Download**）菜单命令。如果程序成功编译，STEP 7-Micro/WIN下载程序块到S7-200。

### 退出RUN（运行）模式编辑

要退出RUN（运行）模式编辑，在“运行”菜单命令中选择调试（**Debug**）> 程序编辑（**Program Edit**），并且取消选择复选标记。如果没有保存所作的改变，STEP7-Micro/WIN将提示是否继续编辑、下载改变和退出运行模式编辑，或者不下载就退出。

## 显示程序状态

STEP 7-Micro/WIN允许在用户程序执行时监控其状态。当监控程序状态时，程序编辑器显示指令操作数数值的状态。

要显示状态，单击“程序状态”按钮或选择调试（Debug）> 程序状态（Program Status）菜单命令。

### 以LAD和FBD显示程序的状态

STEP 7-Micro/WIN提供两种用于显示LAD和FBD程序状态的选项:

- 扫描状态结束: STEP 7-Micro/WIN获得经过多个扫描循环的状态显示值，然后更新状态屏幕显示。状态显示不反应每个元素在执行时的真实状态。扫描结束状态不显示L内存或累加器的状态。  
对于扫描结束状态，状态值在所有CPU操作模式下更新。
- 执行状态: 当元素在S7-200中执行时，STEP 7-Micro/WIN显示程序段的值。要显示执行状态，选择调试（Debug）> 使用执行状态（Use Execution Status）菜单命令。  
对于执行状态，状态值只有当CPU处于RUN（运行）模式时才更新。



#### 提示

STEP 7-Micro/WIN提供简单的方法用于改变变量的状态。只需选择变量然后用鼠标右键单击，即可显示菜单选项。

### 配置如何在LAD和FBD程序中显示状态

STEP 7-Micro/WIN提供多种选项用于在程序中显示状态。

要为状态屏幕配置显示选项，选择工具（Tools）> 选项（Options）菜单命令，然后选择程序编辑器（Program Editor）标签，如图8-2所示。

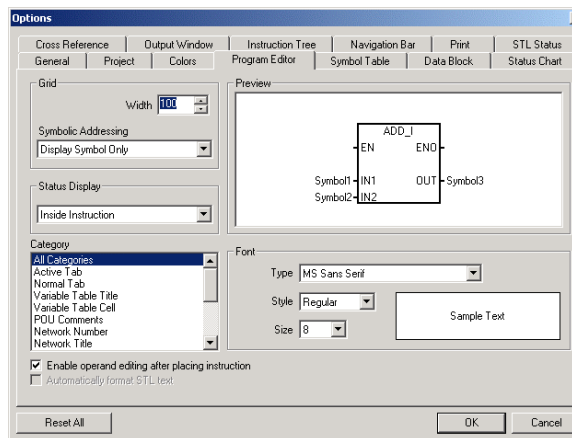


图8-2 状态显示的选项

## 以STL显示程序状态

可以在逐个指令的基础上监控STL程序的执行状态。对于STL程序，STEP 7-Micro/WIN显示被显示在屏幕上的指令的状态。

STEP 7-Micro/WIN从位于编辑器窗口顶部的第一个STL语句开始，收集来自S7-200的状态信息。当向下滚动编辑器窗口时，从S7-200收集新信息。

STEP 7-Micro/WIN持续更新屏幕上的数值。要暂停屏幕更新，选择“触发暂停”按钮。当前数据将保留在屏幕上，直到取消选定“触发暂停”按钮。

### 配置哪些参数在STL程序中显示

STEP 7-Micro/WIN允许显示STL指令的各种参数的状态。选择工具 (Tools) > 选项 (Options) 菜单命令，然后选择“STL状态”标签。参见图8-3。

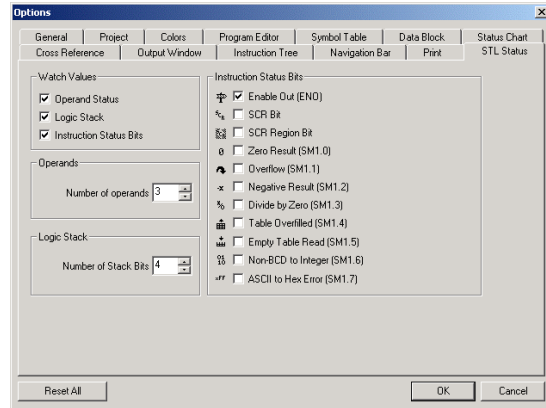


图8-3 显示STL状态的选项

## 使用状态图监控和修改S7-200中的数据

“状态图”允许当S7-200在执行程序时读、写、强制和监控变量。选择视图 (View) > 组件 (Component) > 状态图 (Status Chart) 菜单命令创建状态图。图8-4显示采样状态图。

可以创建多种状态图。

STEP 7-Micro/WIN提供了用于操作状态图的工具栏图标：递增、递减、单次读取、全部写入、强制、取消强制、全部取消强制和读取所有强制。

要选择单元格的格式，选择单元格然后单击鼠标右键显示关联菜单。

Address	Name	Format	Current Value	New Value
1	Start_1	Bit	2#1	
2	Start_2	Bit	2#0	
3	Stop_1	Bit	2#1	
4	Stop_2	Bit	2#0	
5	High_Level	Bit	2#0	
6	Low_Level	Bit	2#0	
7	Reset	Bit	2#0	
8		Signed		
9	Pump_1	Bit	2#1	
10	Pump_2	Bit	2#0	
11	Mixer_Motor	Bit	2#0	
12	Steam_Valve	Bit	2#0	
13	Drain_Valve	Bit	2#1	
14	Drain_Pump	Bit	2#1	
15		Signed		
16	High_Lev_Reached	Bit	2#1	
17	Mix_Timer	Signed	+32767	
18	Cycle_Counter	Signed	+0	

图8-4 状态图

## 强制指定值

S7-200允许强制任何或所有I/O点（I和Q位）。此外，也可以强制至多16个内存数值（V或M）或者模拟I/O数值（AI或AQ）。V内存或M内存数值可以以字节、字或双字强制。模拟值只能在偶数字节边界（诸如AIW6或AQW14）上强制为字。所有强制值都存储在S7-200的永久性EEPROM内存中。

由于在扫描循环期间所强制的数据可能被改变（被程序、I/O更新周期或通讯处理周期改变），S7-200在扫描循环的不同时间重新应用强制值。

- **读取输入：** S7-200在读取强制值时将其应用到输入。
- **在程序中执行控制逻辑：** S7-200将强制值应用到所有立即I/O存取。在程序执行后，强制值可申请至多16个内存数值。
- **处理任何通讯请求：** S7-200将强制值应用到所有读写通讯存取。
- **写入输出：** S7-200在写入强制值时将其应用到输出。

可以使用“状态图”强制数值。要强制新数值，在“状态图”的“新数值”列中输入数值，然后按下工具栏上的“强制”按钮。要强制已存在的数值，高亮显示“当前值”列中的数值，然后按下“强制”按钮。

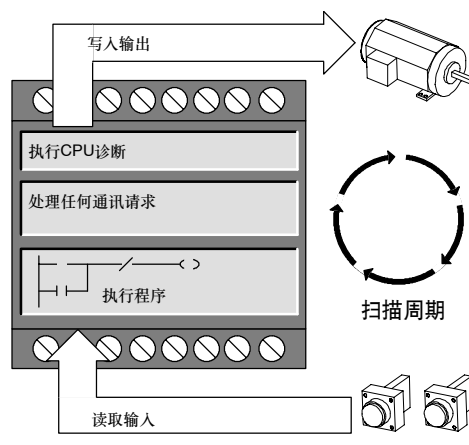


图8-5 S7-200扫描循环

8



### 提示

“强制”功能覆盖了“立即读或“立即写”指令。“强制”功能也覆盖了为转变到STOP（停止）模式而配置的输出表格。如果S7-200跳转到STOP（停止）模式，则输出将反映强制值，而不是在输出表中配置的数值。

## 运行指定扫描数的程序

为了帮助调试程序，STEP 7-Micro/WIN允许运行指定扫描数的程序。

可以让S7-200只执行首次扫描。这允许在首次扫描后监控在S7-200中的数据。选择调试（**Debug**）> 首次扫描（**First Scan**）菜单命令来运行首次扫描。

可以让S7-200执行有限扫描数的程序（从1次扫描到65,535次扫描）。这允许当程序改变变量时对其进行监控。选择调试（**Debug**）> 多次扫描（**Multiple Scans**）菜单命令来指定要执行的扫描数。

## 硬件故障诊断指南

表8-1 S7-200硬件的故障诊断指南

症状	可能的原因	可能的解决方案
输出停止工作	<ul style="list-style-type: none"> <li>■ 受控设备引起损坏输出的电涌。</li> <li>■ 用户程序出错</li> <li>■ 布线松动或不正确</li> <li>■ 过多的负载</li> <li>■ 输出点被强制</li> </ul>	<ul style="list-style-type: none"> <li>■ 当连接到电感负载（例如电机或继电器）时，应使用适当的抑制电路。参考第3章。</li> <li>■ 正确的用户程序</li> <li>■ 检查布线并更正</li> <li>■ 检查负载点的额定值</li> <li>■ 检查S7-200的强制I/O</li> </ul>
S7-200上的SF（系统故障）灯点亮	<p>下面的列表描述了最常规的出错代码和原因：</p> <ul style="list-style-type: none"> <li>■ 用户程序出错               <ul style="list-style-type: none"> <li>- 0003 监视程序出错</li> <li>- 0011 间接地址</li> <li>- 0012 非法的浮点数</li> <li>- 0014 范围出错</li> </ul> </li> <li>■ 电气噪声（0001到0009）</li> <li>■ 组件损坏（0001到0010）</li> </ul>	<p>阅读严重错误代码号，并参考附录C以获取有关出错类型的信息：</p> <ul style="list-style-type: none"> <li>■ 对于编程错误，请检查FOR、NEXT、JMP、LBL和Compare指令的使用。</li> <li>■ 对于电气噪声：               <ul style="list-style-type: none"> <li>- 参考第3章中的布线指南。将控制面板连接到良好的接地且高压布线不与低压布线并行，这是非常重要的。</li> <li>- 将24 VDC传感器电源的M端接地。</li> </ul> </li> </ul>
没有LED点亮	<ul style="list-style-type: none"> <li>■ 熔断丝</li> <li>■ 反向的24 V电源线</li> <li>■ 不正确电压</li> </ul>	<p>将线性分析仪与系统相连以检查过电压峰值的大小和持续时间。基于此信息，为系统添加合适类型的制动器装置。</p> <p>关于安装现场布线的信息请参考第3章中的布线指南。</p>
与高能设备相关的间歇操作	<ul style="list-style-type: none"> <li>■ 不正确接地</li> <li>■ 控制机柜中的布线路由</li> <li>■ 输入过滤的延迟时间太短</li> </ul>	<p>参考第3章中的布线指南。</p> <p>将控制面板连接到良好的接地且高压布线不与低压布线并行，这是非常重要的。</p> <p>将24 VDC传感器电源的M端接地。</p> <p>在系统数据块中增加输入过滤延迟</p>
当连接到外部设备时，通讯网络损坏。 计算机上的端口、S7-200上的端口或PC/PPI电缆损坏。	<p>如果连接到网络的所有非绝缘设备（例如PLC、计算机或其它设备）不共享同样的普通电路基准，则通讯电缆可以为不必要的电流提供路径。</p> <p>不必要的电流可能导致通讯出错，甚至可能损坏电路。</p>	<ul style="list-style-type: none"> <li>■ 参考第3章的布线指南和第7章中的网络指南。</li> <li>■ 购买绝缘PC/PPI电缆。</li> <li>■ 当连接不具有普通电气基准的机器时，请购买绝缘RS85oS85中继器期。</li> </ul> <p>请参考附录E以获取关于S7-200设备的订购号的信息。</p>
其它通讯问题 (STEP 7-Micro/WIN)	请参考第7章以获取关于网络通讯的信息。	
错误处理	请参考附录C以获取关于出错代码的信息。	





# 创建位控模块程序

EM 253位控模块是S7-200的特殊功能模块，它能够产生脉冲串，用于步进电机或伺服电机的速度和位置的开环控制。它与S7-200通过扩展的I/O总线进行通讯，并作为具有八个数字输出的智能模块出现在I/O配置中。

基于存储在S7-200的V内存中的配置信息，位控模块将生成移动控制所需要的脉冲串。

为了简化应用程序中位置控制功能的使用，STEP7-Micro/WIN提供了一个位置控制向导，该向导允许在几分钟之内完成对位控模块的配置。STEP7-Micro/WIN还提供了一个控制面板，允许对移动操作进行控制、监视和测试。

## 本章内容

位控模块的特性 .....	254
配置位控模块 .....	256
位置控制向导创建的位控指令 .....	267
位控模块的范例程序 .....	279
使用EM 253控制面板监控位控模块 .....	284
位控模块和位控指令的出错代码 .....	286
高级主题 .....	288

## 位控模块的特性

位控模块可提供单轴开环位置控制所需要的功能和性能:

- 提供高速控制，范围从每秒12个脉冲至每秒200,000个脉冲
- 支持跳动（S曲线）或线性加速减速
- 提供可配置的测量系统，该系统允许将数据输入为工程单位（例如英寸或厘米）或脉冲数
- 提供可配置的间隙补偿
- 支持绝对的、相对的和手动的定位控制方式
- 提供连续操作
- 提供多达25组运行的概要图，每一概要图最多可有4种速度变化
- 提供4种不同的参考点查找模式，每种模式都可对起始的查找方向和最终的接近方向进行选择
- 提供可拆卸的现场布线连接器，以便安装和拆卸

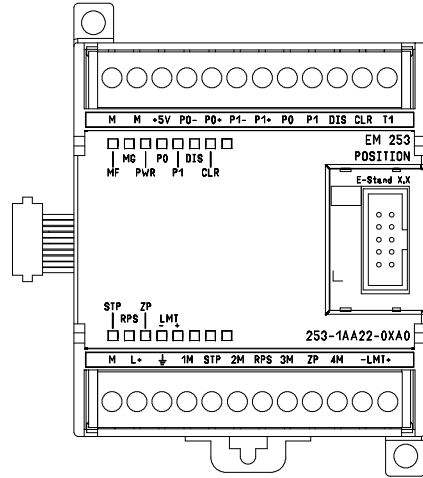


图9-1 EM 253位控模块

使用STEP 7-Micro/WIN可创建位控模块所使用的全部配置和概要图信息。这些信息将和程序块一起下载到S7-200中。由于定位控制所需要的全部信息都存储在S7-200中，因此，可更换位控模块而不必对模块重新编程或配置。

S7-200将在过程映像输出寄存器（Q内存）中保留8位作为位控模块的接口。S7-200中的应用程序将使用这些位来控制位控模块的操作。这8个输出位与位控模块上的任何物理场输出都不相连。

位控模块提供了5个数字输入和4个数字输出，为移动应用程序提供接口。参见表9-1。这些输入和输出均位于位控模块上。附录A提供了关于位控模块的详细规范，同时还包括位控模块与一些更通用的电机驱动器 / 放大器的接线图。

表9-1 位控模块的输入和输出

信号	描述
STP	STP输入将引起模块停止正在进行的移动。在位置控制向导中可选择所需要的STP操作。
RPS	RPS（参考点开关）输入为绝对移动操作建立参考点或原位。
ZP	ZP（零时钟脉冲）输入可帮助建立参考点或原位。典型地，电机驱动器 / 放大器在电机每旋转一周就产生一个ZP脉冲。
LMT+ LMT-	LMT+和LMT-输入将建立移动行程的最大限制。位置控制向导将允许配置LMT+和LMT-输入的操作。
P0 P1 P0+、P0- P1+、P1-	P0和P1为控制电机移动及移动方向的开路漏极晶体管脉冲输出。P0+、P0-和P1+、P1-为分别提供与P0和P1一样功能的差分脉冲输出，但它们所提供的信号质量更好。开路漏极输出和差分输出可同时全部激活。可以根据电机驱动器 / 放大器的接口要求来选择使用哪一种脉冲输出。
DIS	DIS是一种用来禁用或启用电机驱动器 / 放大器的开路漏极晶体管输出。
CLR	CLR是一种用来清除伺服脉冲计数寄存器的开路漏极晶体管输出。

## 位控模块编程

STEP 7-Micro/WIN为位控模块的配置和编程提供了便捷的工具。只须遵循以下步骤:

1. 配置位控模块。STEP 7- Micro/WIN提供一个位置控制向导, 用于创建配置 / 概要表和位控指令。参见页面256关于位控模块配置的信息。
2. 测试位控模块的运行。STEP 7-Micro/WIN提供了一个EM 253控制面板, 用于测试输入输出的布线、位控模块的配置以及对移动概要图的操作。参见页面284关于EM 253控制面板的信息。
3. 创建S7-200将要执行的程序。位置控制向导将自动创建位控指令, 可以将这些指令插入到程序中。参见页面267关于位控指令的信息。将下列指令插入到程序中:
  - 为启用位控模块, 插入POSx\_CTRL指令。请使用SM0.0 (始终接通), 以确保这条指令在每一个扫描周期中都得到执行。
  - 为将电机移动到指定位置, 请使用POSx\_GOTO或POSx\_RUN指令。POSx\_GOTO指令将移动到由程序输入所指定的位置。POSx\_RUN指令将执行在位置控制向导中所配置的概要图的移动。
  - 为使用移动的绝对坐标, 必须为应用建立零点位置。使用POSx\_RSEEK或POSx\_LDPOS指令来建立零位置。
  - 位置控制向导所创建的其它指令将提供用于典型应用的功能, 并且对于特殊应用情况下是可选的。
4. 编译程序并将系统块、数据块和程序块下载到S7-200中。



### 提示

请参见附录A中关于位控模块与多个公用步进电机控制器连接的信息。



### 提示

为与位置控制向导中的缺省设置相匹配, 可将步进电机控制器上的DIP开关设置为每转10,000个时钟脉冲。

## 配置位控模块



为了使模块能够控制移动，必须为位控模块创建配置 / 概要表。位置控制向导将引导一步一步地完成整个配置过程，从而使配置过程快速而方便。关于配置 / 概要表的更详细信息，请参见页面288上的高级主题。

定位控制

位置控制向导还允许脱机创建配置 / 概要表。可以在不连接S7-200 CPU的情况下，使用所安装的位控模块创建配置。

为运行位置控制向导，必须对项目进行编译，并选择符号编址模式。

为启动位置控制向导，既可以单击浏览条中的“工具”图标，然后双击“位置控制向导”图标，也可以选择菜单命令工具>位置控制向导。

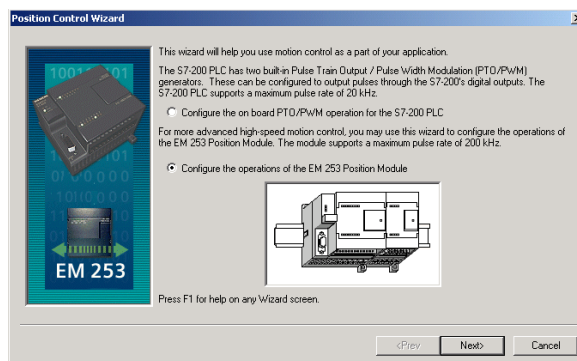


图9-2 位置控制向导

位置控制向导允许既能配置位控模块的操作，也能配置脉冲输出指令的PTO/PWM操作。在选择位控模块选项之后，单击“下一步”，向导将引导完成位控模块配置所需的步骤。

### 输入位控模块的位置

必须通过输入模块类型和位置来定义模块参数，并为应用设置移动概要图。位置控制向导可自动读取智能模块的位置，从而简化上述任务。只需单击“读取模块”按钮。

对于固化程序版本在V1.2之前的S7-200 CPU，智能模块必须安装在紧邻CPU的位置，以便使用位置控制向导对模块进行配置。

### 选择测量类型

必须选择要在整个配置中使用的测量系统。可选择使用工程单位或脉冲。如果选择脉冲，则不必再定义任何其它信息。如果选择工程单位，则必须输入以下数据：使电机旋转一周所需的脉冲数（参见电机或驱动器的数据表）、测量的基本单位（例如英寸、英尺、毫米或厘米）以及电机旋转一周所引起的位移量（或“单元”）。

STEP 7-Micro/WIN提供了一个EM 253控制面板，允许在配置位控模块后修改每次旋转的单元数。

如果以后改变测量系统，则必须删除整个配置，包括位置控制向导所生成的所有指令。之后，必须输入与新的测量系统一致的选项。

## 编辑缺省的输入和输出配置

位置控制向导提供了一个“高级”选项，可对位控模块的缺省输入和输出配置进行浏览和编辑：

- “输入现用级别”标签可改变现用级别设置。当将级别设置为“高”时，逻辑1将在输入中存在电流时读取。当将级别设置为“低”时，逻辑1将在输入中没有电流时读取。逻辑1级别始终意味着条件是现用的。不论现用级别如何，LED在输入中存在电流时将点亮。（缺省值=现用高）
- “输入过滤器时间”标签用来为过滤STP、RPS、LMT+和LMT-输入指定延迟时间（范围为0.20毫秒至12.80毫秒）。这种延迟有助于过滤输入布线上的噪声，这种噪声可能会导致不小心改变输入的状态。（缺省值=6.4毫秒）
- “脉冲与定向输出”标签可用来指定对方向进行控制的方式。首先，必须指定输出的极性。

### 选择正极

对于使用正极的应用，请选择下列方式之一（如图9-3所示），以适应驱动器及应用的方向：

- 位控模块从P0输出发出正向旋转脉冲，从P1输出发出负向旋转脉冲。
- 位控模块从P0输出发出脉冲。模块在正向旋转时将接通P1输出，在负向旋转时将关断P1输出。（此为缺省设置。）

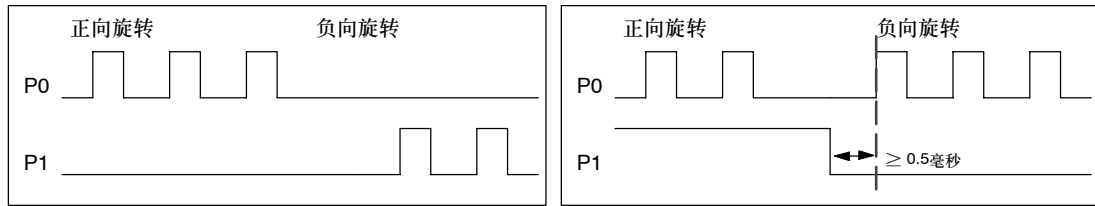


图9-3 正极的转向选项

### 选择负极

对于使用负极的应用，请选择下列方式之一（如图9-4所示），以适应驱动器及应用的方向：

- 位控模块从P0输出发出负向旋转脉冲，从P1输出发出正向旋转脉冲。
- 位控模块从P0输出发出脉冲。模块在正向旋转时断开P1输出，在负向旋转时接通P1输出。

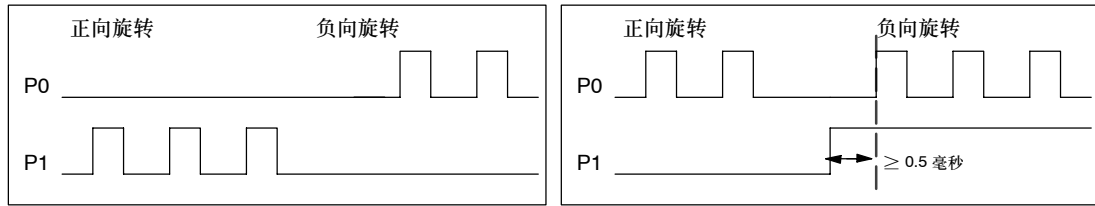


图9-4 负极的转向选项

### 方向改变

为改变方向，必须停止脉冲输出，而这可通过结束概要图或使用STOP命令和新方向所发出的新RUN命令来完成。方向信号将在新序列的第一个脉冲到达之前至少0.5微秒进行切换。

## 配置模块对物理输入的响应

必须指定位控模块对每个LMT+开关、LMT-开关以及STP输入的响应：无任何动作（忽略输入条件）、减速至停止（缺省）或立即停止。



### 警告

控制设备在不安全条件下可能出现故障，并可导致对被控设备的不可预知的操作。这些不可预知的操作可能导致人员的严重伤亡以及/或设备的损坏。

位控模块的限位和停止功能通过电子逻辑实现，没有提供机电控制设备所能提供的保护等级。请考虑使用不受S7-200 CPU和位控模块限制的急停功能、机电保险装置或冗余安全措施。

## 输入最大速度和起动 / 停止速度

必须为应用指定最大速度（MAX\_SPEED）和启动 / 停止速度（SS\_SPEED）：

- ❑ MAX\_SPEED: 在电机力矩能力范围内，输入应用中最佳操作速度的数值。驱动负载所需的力矩由摩擦力、惯性以及加速 / 减速时间所决定。位置控制向导将根据指定的MAX\_SPEED来计算并显示位控模块所能控制的最小速度。
- ❑ SS\_SPEED: 在电机能力范围内输入一个数值，以便以较低的速度驱动负载。如果SS\_SPEED值过低，则电机和负载在移动开始和结束时，将可能会摇摆或颤动。如果SS\_SPEED值过高，则电机在启动时丢失脉冲，而且负载在试图停止时会过度驱动电机。

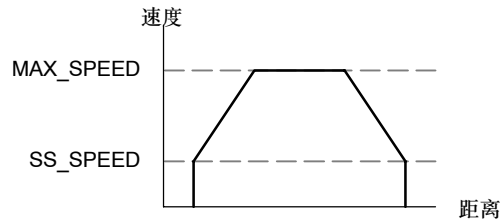


图9-5 最大速度和启动 / 停止速度

对于电机和给定的负载，电机数据单存在多种不同的方式来指定启动 / 停止（或拉入 / 拉出）速度。通常，有效的SS\_SPEED值为MAX\_SPEED值的5%至15%。SS\_SPEED值必须大于根据指定的MAX\_SPEED值而显示出来的最小速度。

为帮助所应用的正确速度，请参考电机的数据表。图9-6显示了一个典型的电机扭矩 / 速度曲线。

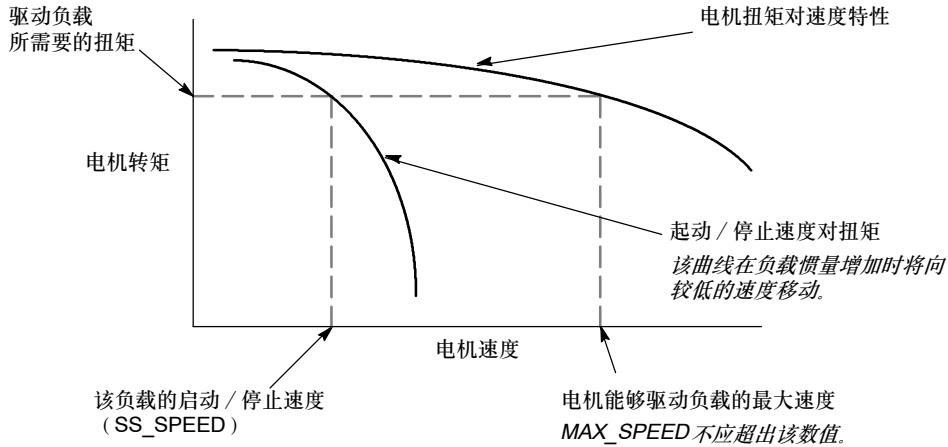


图9-6 电机的转矩-速度曲线

## 输入慢进参数

慢进命令用于将工具以手动方式移动到所期望的位置。使用位置控制向导，可指定以下慢进参数值：

- JOG\_SPEED: JOG\_SPEED（电机的慢进速度）是JOG命令仍然有效时能够得到的最大速度。
- JOG\_INCREMENT: 瞬间的JOG命令能够将工具移动的距离。

图9-7显示了慢进命令的操作。当位控模块接收到一个慢进命令时，它将启动一个计时器。如果在0.5秒内终止慢进命令，则位控模块将以SS\_SPEED所定义的速度，按JOG\_INCREMENT所指定的距离移动工具。如果0.5秒后慢进命令仍然激活，则位控模块将加速至JOG\_SPEED。移动将继续，直至慢进命令终止。位控模块随后将减速停止。激活慢进命令，既可在EM 253控制面板中进行，也可使用位控指令。

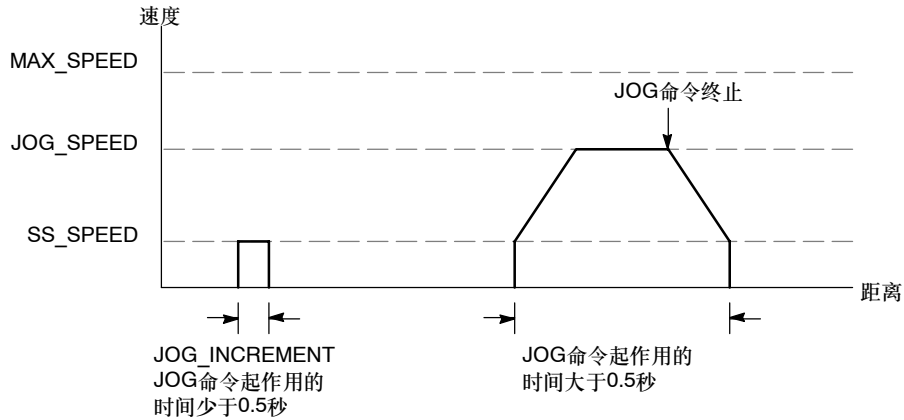


图9-7 JOG操作说明

## 输入加速和减速时间

作为位控模块配置的一部分，必须设置加速和减速时间。加速时间和减速时间二者的缺省设置都是1秒。典型地，电机可在小于1秒的时间内完成。以毫秒为单位指定下列时间：

- ACCEL\_TIME: 电机从SS\_SPEED加速到MAX\_SPEED所需要的时间。缺省值=1000毫秒
- DECEL\_TIME: 电机从MAX\_SPEED减速到SS\_SPEED所需要的时间。缺省值=1000毫秒

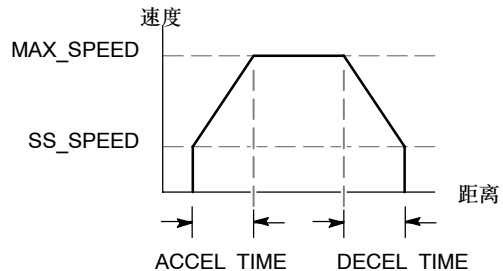


图9-8 加速和减速时间



### 提示

电机的加速和减速时间通过试验和错误确定。必须一开始就在位置控制向导中输入一个较大的值。进行应用测试时，可以根据需要，使用EM 253控制面板来调整数值。通过逐渐减少时间来优化应用中的这些设置，直至电机开始停转。

## 输入跳动时间

跳动补偿通过减小移动概要图加速和减速部分中的跳动（变化速率）来提供较平滑的位置控制。参见图9-9。减少跳动将可改善位置跟踪性能。跳动补偿也称为“S曲线包络图”。跳动补偿只能应用于简单的单工步概要图。这种补偿可同样地应用于加速曲线和减速曲线的开始和结束部分。跳动补偿不能够应用在介于零速和SS\_SPEED速度之间的初始工步和结束工步上。

通过输入时间值（JERK\_TIME）来指定跳动补偿。这一时间，是加速度从零改变到由MAX\_SPEED、SS\_SPEED和ACCEL\_TIME或与之相等的DECEL\_TIME参数所定义的最大加速度所需要的时间。与只是简单地增加ACCEL\_TIME和DECEL\_TIME相比，较长的跳动时间将由于能够对整个循环时间只有一个较小的增加，从而可产生平滑的操作。零值表示没有进行任何补偿。

（缺省值=0毫秒）

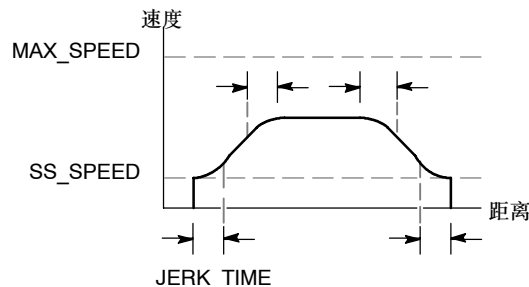


图9-9 跳动补偿



### 提示

JERK\_TIME的适当的初始值应为ACCEL\_TIME的40%。

## 配置参考点和搜索参数

如果应用指定从一个绝对位置开始移动，则必须建立一个零点位置，它将使位置测量与实际系统中的一个已知点相吻合。一种方法就是提供一个实际系统中的参考点（RP）。位控模块提供了一个外部参考点开关（RPS）输入，用于在搜索RP时使用。

可配置参考点查找（RP查找）参数，以控制移动应用将如何查找RP。RP可以位于RPS有效区的中央，也可以位于RPS有效区的边缘，甚至可以就位于RPS有效区边缘的特定数目的零脉冲（ZP）输入转换处。为配置RP，可输入下列信息：

- 指定电机的RP搜索速度：
  - RP\_FAST是执行RP搜索命令时模块所使用的初始速度。通常，RP\_FAST数值大约为MAX\_SPEED数值的2/3。
  - RP\_SLOW是最终接近RP的速度。将使用一个较慢的速度去接近RP，以免将其错过。通常，RP\_SLOW数值为SS\_SPEED数值。
- 为RP搜索指定初始的搜索方向（RP\_SEEK\_DIR）和最终的接近方向（RP\_APPR\_DIR）。这些方向指定为正方向或负方向。
  - RP\_SEEK\_DIR为RP搜索操作的初始方向。通常，这个方向为工作区到RP附近的方向。限位开关在定义RP搜索区域时扮演重要的角色。在执行RP搜索操作时，如果遇到限位开关，将可能导致方向反转，从而使搜索能够继续进行。（缺省值=负方向）
  - RP\_APPR\_DIR是最终接近RP的方向。为减少反冲并提供更高的准确性，RP\_APPR\_DIR将按照与工作循环相同的方向移动。（缺省值=正方向）



位置控制向导将提供高级参考点选项，该选项允许指定一个RP偏移量（RP\_OFFSET），这个偏移量是从RP到零位置的距离。参见图9-10。以RPS为参考确定一个准确的位置，采用这种方法，可标识RP。为配置RP偏移量，可输入下列数值：

- RP\_OFFSET: RP到实际测量系统零位置之间的距离。缺省值=0
- 间隙补偿: 在方向发生变化时，为消除系统中的停滞（反冲），电机必须移动的距离。间隙补偿始终为正值。缺省值=0

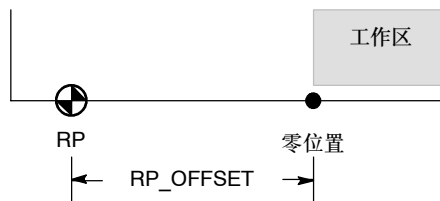


图9-10 RP和零位置之间的关系

### 配置RP搜索顺序

可对位控模块用来搜索参考点的顺序进行配置。图9-11显示了缺省RP搜索顺序的简化图。可为RP搜索顺序选择下列选项：

- RP查找模式0: 不执行RP搜索顺序
- RP查找模式1: RP就是RPS输入在接近工作区一侧开始激活的地方。（缺省）
- RP查找模式2: RP位于RPS输入有效区内的中心。
- RP查找模式3: RP位于RPS输入有效区的外面。RP\_Z\_CNT指定在RPS失效之后应接收多少个ZP（零脉冲）输入计数。
- RP查找模式4: RP通常位于RPS输入有效区内。RP\_Z\_CNT指定在RPS有效之后应接收多少个ZP（零脉冲）输入计数。

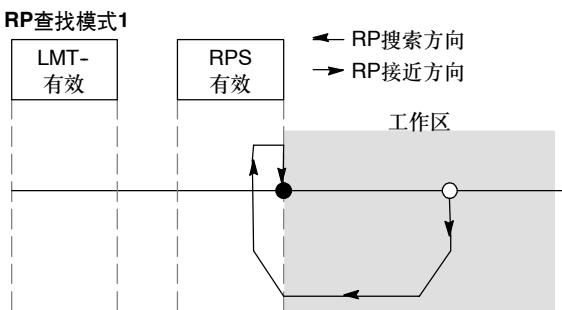


图9-11 缺省RP搜索顺序（已简化）



**提示**

RPS有效区（即RPS输入仍然有效的距离）必须大于从RP\_FAST速度减速到RP\_SLOW速度需要的距离。如果距离太短，则位控模块产生错误。

更多的关于位控模块不同RP搜索顺序的信息，请参见图9-14至9-17，这些图位于页面264至265上。

## 配置位控模块的移动概要图

概要图是预定义的移动描述，包括一个或多个移动速度，这些速度将影响从起点到终点的移动。即使不定义概要图，也可以使用模块。位置控制向导提供了一个子例行程序（POSx\_GOTO），用于对移动进行控制。

- 概要图数目：最多可选择25个概要图。
- 命令字节的地址：必须为位控模块输入命令字节的输出（Q）内存地址。参见图4-10中关于I/O编号的描述（页面31）。
- 配置 / 概要表的地址：必须为配置 / 概要表输入起始内存地址，该配置 / 概要表存储位控模块的配置数据和所有概要图的数据。位控模块的配置数据需要92字节的V内存，每个概要图需要34字节的V内存。例如，对于带有一个概要图的位控模块，其配置 / 概要表所需占用的内存是126字节的V内存。

位置控制向导可以建议一个大小合适的未经使用的V内存地址。

## 定义移动概要图

位置控制向导提供一个“移动概要图定义”，可以在该处为应用定义每个移动概要图。可为每个概要图选择操作模式，并定义概要图各个单工步的具体内容。“位置控制向导”还允许在定义概要图时，为每个概要图定义一个符号名。在完成概要图的配置后，可将其存储到配置，并为参数打印一个副本。

### 选择概要图的操作模式

按照操作模式配置概要图，选择绝对位置或相对位置、单速连续旋转或双速连续旋转。图9-12显示了各种不同的操作模式。

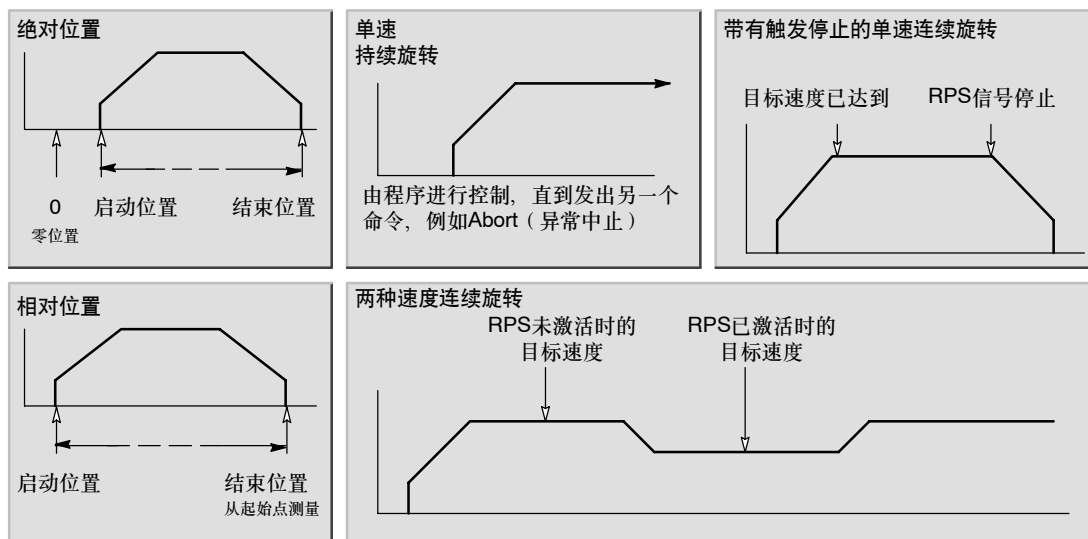


图9-12 位控模块的模式选择

### 创建概要图的步进

步进是工具移动的一个固定距离，包括加速和减速时间内所经过的距离。每个概要图至多可以有4个单独的步进。

为每个步进指定目标速度和结束位置。如果不只一个步进，则只需单击“新步进”按钮，为概要图的每个步进输入信息。图9-13显示了四种可能的概要图；然而，还存在其它可能的组合。

只需单击“绘制步进”按钮，即可看到由位置控制向导计算的、以图形方式表示的步进。这样，就能简单交互地浏览和编辑每个步进。

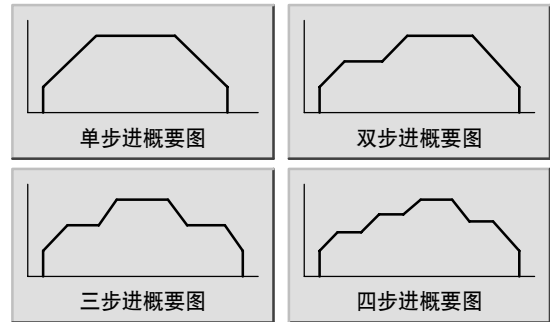


图9-13 范例移动概要图

### 完成位控模块的配置

完成对位控模块操作的配置之后，只需单击“完成”，位置控制向导即可执行以下任务：

- 将模块配置和概要表插入到S7-200程序的数据块中。
- 为移动参数创建一个全局符号表
- 在项目程序块中添加移动指令子例行程序，以便在应用中使用。

要修改任何配置或概要图信息，可以再次运行位置控制向导。



#### 提示

由于位置控制向导对程序块、数据块和系统块进行了修改，请确保这三种块都下载到S7-200 CPU中。否则，位控模块可能会无法得到正确操作所需要的所有程序组件。

### 了解位控模块所支持的RP查找模式

下列各图提供了每种RP查找模式不同选项的示意图。

- 图9-14显示RP查找模式1的两种选项。这种模式将RP定位在靠近工作区一侧的RPS输入开始激活的地方。
- 图9-15显示RP查找模式2的两种选项。这种模式将RP定位在RPS输入有效区域内的中央。
- 图9-16显示RP查找模式3的两种选项。这种模式将RP定位在RPS输入有效区域之外的指定数量的零脉冲（ZP）处。
- 图9-17显示RP查找模式4的两种选项。这种模式将RP定位在RPS输入有效区域内的指定数量的零脉冲（ZP）处。

对于每种模式，都存在有RP搜索方向和RP接近方向的四种组合。（只显示了两种组合。）这些组合确定RP搜索操作的模式。对每一种组合，又存在四种不同的起始点：

每个图中的工作区都已确定，以便使从参考点到工作区的移动按照与RP接近方向相同的方向进行。通过以这种方式选择工作区的位置，机械齿轮系统中的所有反冲，都能够在进行参考点搜索之后，在工作区的第一次移动中去除。

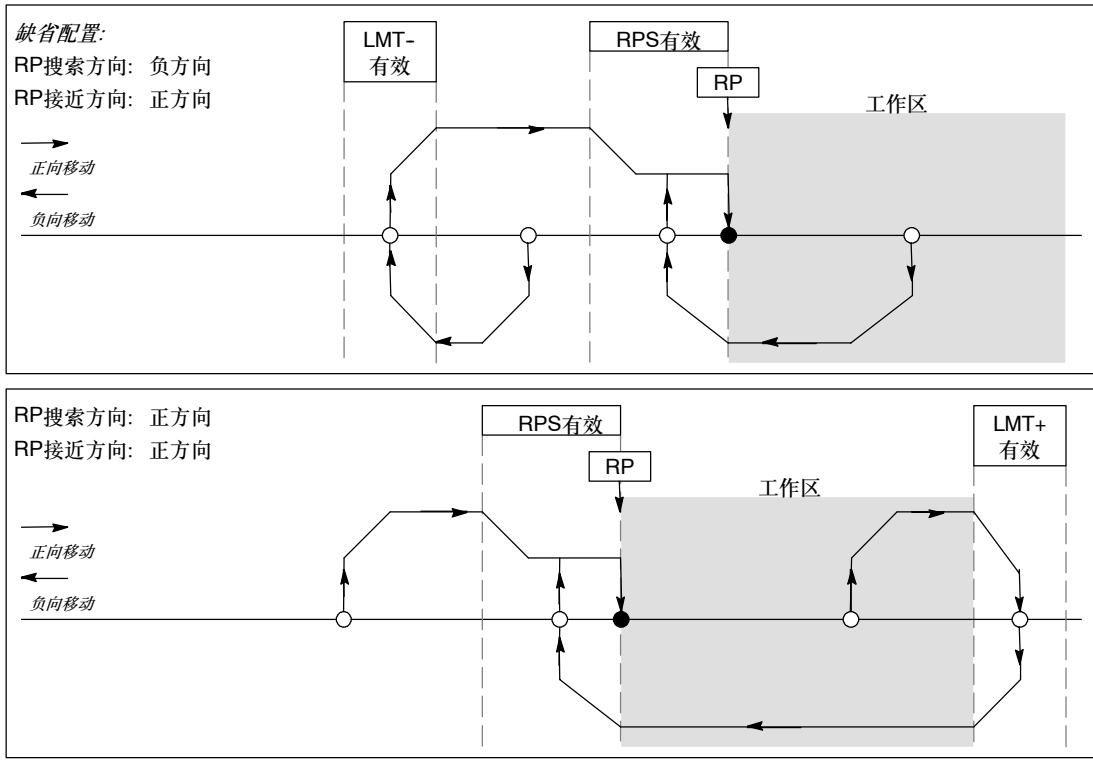


图9-14 RP查找模式1

9

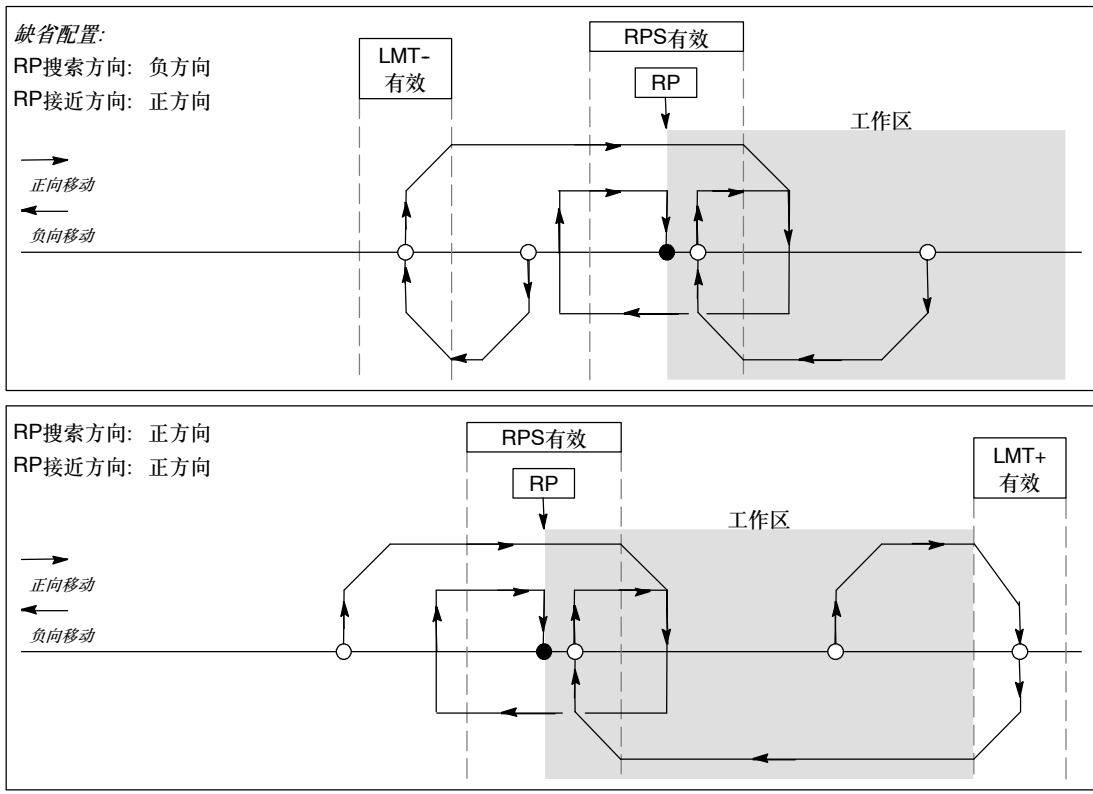


图9-15 RP搜索: 模式2

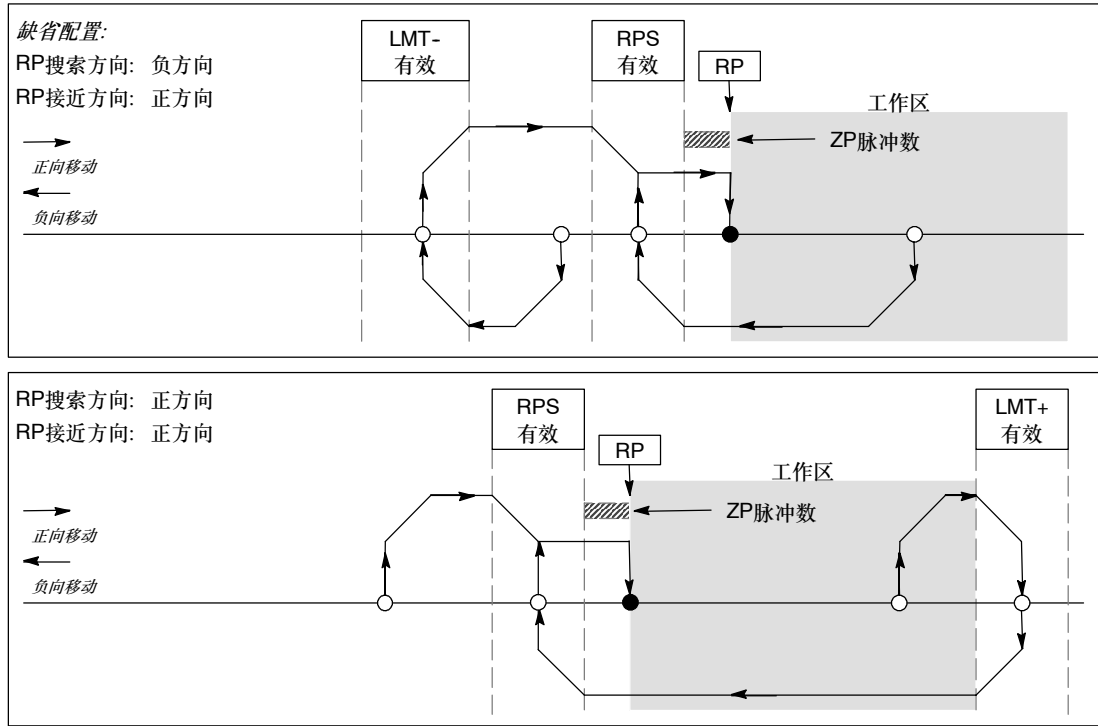


图9-16 RP搜索：模式3

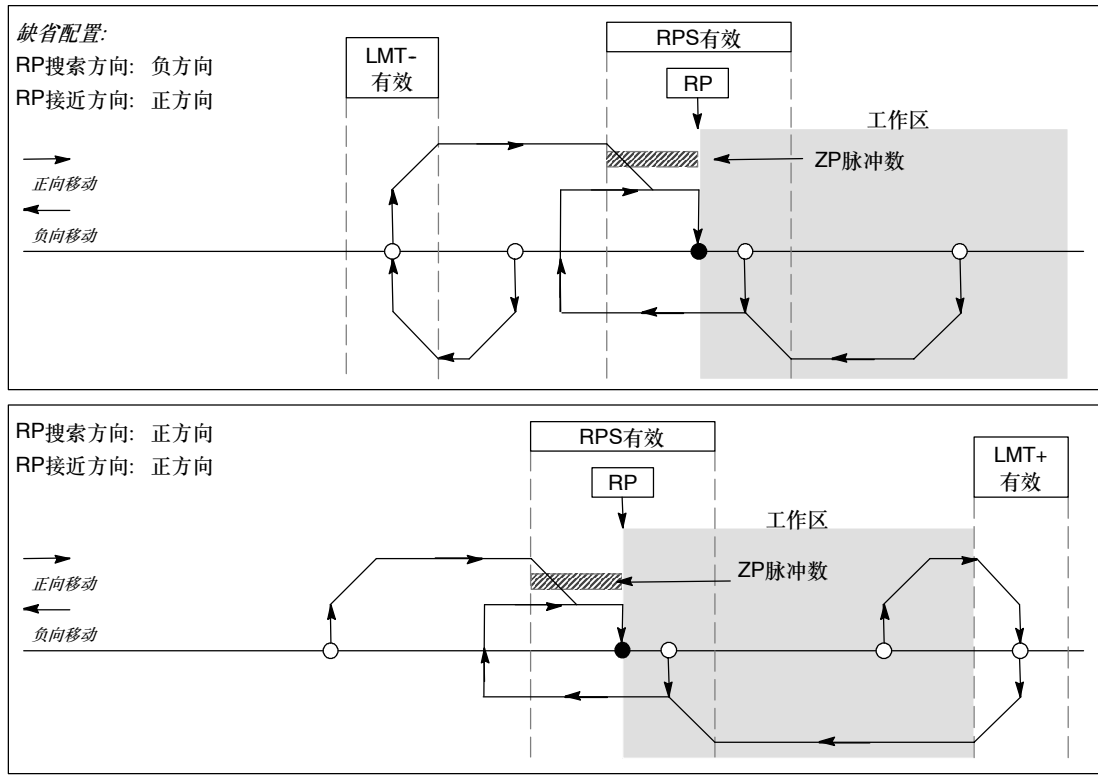


图9-17 RP搜索：模式4

### 选择可以消除反冲的工作区位置

图9-18显示了在接近方向上能够消除反冲的工作区及其相关的参考点（RP）、RPS有效区以及限位开关（LIM+和LIM-）。图示的第二部分放置了工作区，这样，反冲没有消除。图9-18显示了RP查找模式3。对于各个其它RP查找模式的每种搜寻顺序，也可以有类似的工作区位置，但没有推荐。

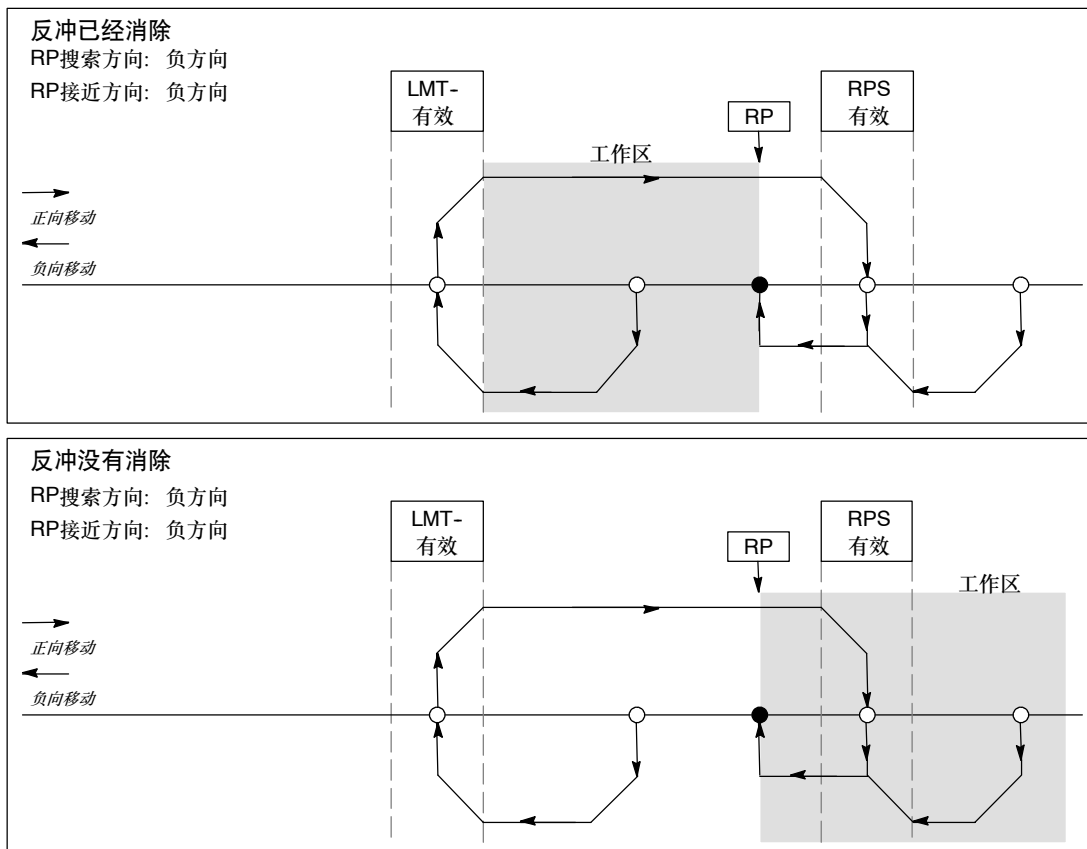


图9-18 已消除和尚未消除反冲的工作区位置

## 通过位置控制向导创建的位控指令

基于模块位置和所选的配置选项，通过创建唯一的子例行程序，位置控制向导使位控模块的控制变得比较容易。每条位控指令都有前缀“POS<sub>x</sub>”，这里<sub>x</sub>为模块位置。由于每条位控指令都是一个子例行程序，因此，11条位控指令将使用11个子例行程序。



### 提示

位控指令将使用户程序对存储空间的需求量最多增加至1700字节。可以删除不用的位控指令，以便减小对存储空间的需求量。要恢复已删除的位控指令，只需再次运行位置控制向导即可。

## 位控指令使用指南

必须确保在同一时刻只有一条位控指令激活。

可以从中断例行程序执行POS<sub>x</sub>\_RUN和POS<sub>x</sub>\_GOTO。然而，如果模块正忙于处理其它命令，请千万不要试图启动中断例行程序中的指令。如果启动中断例行程序中的指令，则随后可以使用POS<sub>x</sub>\_CTRL指令的输出来监控位控模块何时完成移动。

位置控制向导将按照所选择的测量系统，自动配置速度参数（Speed和C\_Speed）和位置参数（Pos或C\_Pos）的值。对于脉冲，这些参数就是DINT值。对于工程单元，这些参数就是所选单元类型的REAL值。例如：选择厘米（cm）为单位将位控参数存储为以厘米为单位的REAL值，将速度参数存储为以每秒厘米数（cm/sec）为单位的REAL值。

以下是特定的定位控制任务所需要的位控指令：

- 在程序中插入POS<sub>x</sub>\_CTRL指令，使用SM0.0触点，以便在每个循环中执行。
- 为指定到绝对位置的移动，必须首先使用POS<sub>x</sub>\_RSEEK或POS<sub>x</sub>\_LDPOS指令建立零位置。
- 为移动到某个特定位置，根据程序中的输入，可使用POS<sub>x</sub>\_GOTO指令。
- 为运行在位置控制向导中所配置的移动概要图，可使用POS<sub>x</sub>\_RUN指令。

其他位控指令是可选的。

## POSx\_CTRL指令

每当S7-200转换为RUN（运行）模式时，POSx\_CTRL指令（控制）将自动向位控模块发出装载配置 / 概要表的命令，从而激活并启动位控模块。

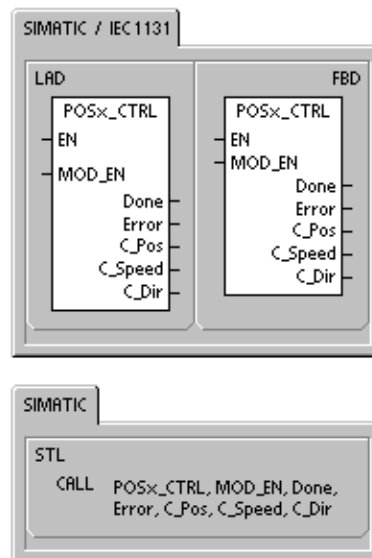
项目中只能使用一次该指令，请确保程序在每次扫描中都调用该指令。使用SM0.0（始终接通）作为MOD\_EN参数的输入。

MOD\_EN参数必须处于接通状态，以使其它位控指令能够发送命令给位控模块。如果MOD\_EN参数处于断开状态，则位控模块将放弃执行所有正在进行当中的命令。

POSx\_CTRL指令的输出参数提供位控模块的当前状态。

位控模块完成任何指令时，参数“Done”将接通。

参数Error包含该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。



参数“C-Pos”是模块的当前位置。基于测量单位，该值既可以是一系列的脉冲（DINT），也可以是一系列的工程单位（REAL）。

参数“C-Speed”提供了模块的当前速度。如果配置用于脉冲的位控模块测量系统，则C\_Speed是一个包含有每秒脉冲数的DINT数值。如果配置用于工程单位的测量系统，则C\_Speed是一个包含有每秒所选工程单位数的REAL数值。

参数“C\_Dir”表示电机的当前方向。

表9-2 用于POSx\_CTRL指令的参数

输入 / 输出	数据类型	操作数
MOD_EN	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
Done、C_Dir	BOOL	I、Q、V、M、SM、S、T、C、L
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
C_Pos、C_SPEED	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD



### 提示

位控模块只在上电或接到装载配置的命令时才读取配置 / 概要表。

- 如果使用位置控制向导来修改配置，则每当S7-200 CPU切换到RUN模式时，POSx\_CTRL指令就将自动命令位控模块装载配置 / 概要表。
- 如果使用EM 253控制面板来修改配置，则可单击“更新配置”按钮，命令位控模块装载新配置 / 概要表。
- 如果使用另一种方法修改配置，则还必须发出一条“重新载入配置”命令给位控模块，以便载入配置 / 概要表。否则，位控模块将继续使用旧的配置 / 概要表。



## POSx\_MAN指令

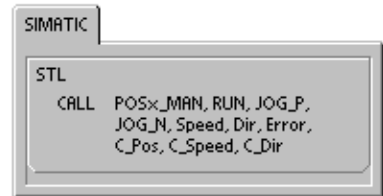
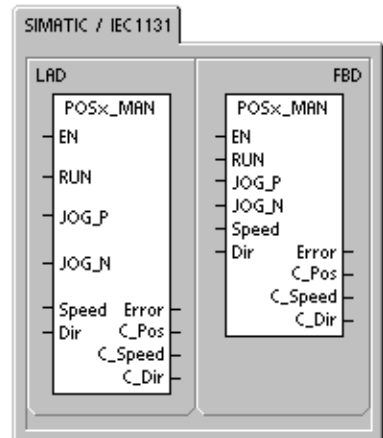
POSx\_MAN指令（手动模式）将位控模块置于手动模式。这将使电机能够以不同的速度运行或沿正向或负向方向慢进。激活POSx\_MAN指令时，将只允许运行POSx\_CTRL和POSx\_DIS指令。

每次只能激活RUN、JOG\_P或JOG\_N输入中的一个。

激活RUN（RUN/Stop）参数，命令位控模块加速到指定的速度（Speed参数）和方向（Dir参数）。你可以在电机运行时修改Speed参数的值，但参数Dir必须保持为常数。禁止RUN参数，命令位控模块减速至电机逐渐停止为止。

激活参数JOG\_P（慢进正向旋转）或JOG\_N（慢进负向旋转），命令位控模块沿正方向或反方向慢进。如果JOG\_P或JOG\_N参数在短于0.5秒的时间内仍然有效，则位控模块将发出脉冲，使其移动JOG\_INCREMENT所指定的距离。如果JOG\_P或JOG\_N参数在等于或长于0.5秒的时间内仍然有效，则移动模块将开始加速到指定的JOG\_SPEED。

Speed参数确定RUN激活时的速度。如果将位控模块的测量系统配置为脉冲，则该速度是一个每秒若干脉冲数的DINT数值。如果将位控模块的测量系统配置为工程单位，则该速度是一个每秒若干单位的REAL数值。可在电机运行时修改该参数。



### 提示

位控模块有可能不对Speed参数的微小变化作出反应，特别是所配置的加速或减速时间较短，而所配置的最大速度和起始/停止速度之间相差较大的时候。

参数Dir确定RUN激活时的移动方向。该参数值在RUN参数有效时不能进行修改。

参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。

参数C\_Pos包含模块的当前位置。基于所选测量单位，该值既可以是脉冲数（DINT），也可以是工程单位数（REAL）。

参数C\_SPEED包含模块的当前速度。基于所选测量单位，该值既可以是每秒脉冲数（DINT），也可以是每秒工程单位数（REAL）。

参数“C-Dir”表示电机的当前方向。

表9-3 POSx\_MAN指令的参数

输入 / 输出	数据类型	操作数
RUN、JOG_P、JOG_N	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
Speed	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD、常量
Dir、C_Dir	BOOL	I、Q、V、M、SM、S、T、C、L
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
C_Pos、C_SPEED	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD

## POSx\_GOTO指令

POSx\_GOTO指令将命令位控模块跳转到所期望的位置。

接通EN位将激活指令。请确保EN位始终处于接通，直到DONE位指示已经完成指令的执行。

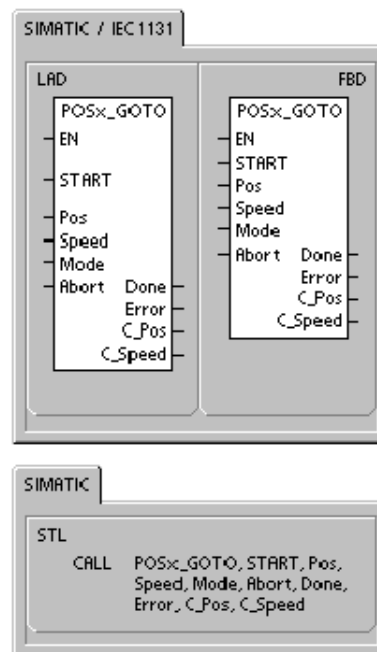
接通参数START将可向位控模块发送GOTO命令。对于每一循环，当参数START接通且位控模块当前不忙时，指令将给位控模块发送GOTO命令。为确保只发送一条GOTO命令，可使用边沿检测单元来以脉冲触发START参数的接通。

参数POS包含一个表示移动位置（对于绝对移动）或移动距离（对于相对移动）的值。基于所选测量单位，该值既可以是一个脉冲数（DINT），也可以是工程单位数（REAL）。

参数Speed确定该移动的最大速度。基于所选测量单位，该值既可以是每秒脉冲数（DINT），也可以是每秒工程单位数（REAL）。

参数Mode将确定移动的类型：

- 0 - 绝对位置
- 1 - 相对位置
- 2 - 单速、连续正向旋转
- 3 - 单速、连续负向旋转



参数Done在位控模块完成该指令时接通。

接通Abort参数将命令位控模块停止当前概要图并减速，直到电机逐渐停止。

参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。

参数C\_Pos包含模块的当前位置。基于测量单位，该值既可以是一系列的脉冲（DINT），也可以是一系列的工程单位（REAL）。

参数C\_SPEED包含模块的当前速度。基于所选测量单位，该值既可以是每秒脉冲数（DINT），也可以是每秒工程单位数（REAL）。

表9-4 POSx\_GOTO指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
Pos、Speed	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD、常量
Mode	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD、常量
Abort、Done	BOOL	I、Q、V、M、SM、S、T、C、L
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
C_Pos、C_SPEED	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD

## POSx\_RUN指令

POSx\_RUN指令（运行概要图）将命令位控模块执行已存储在配置/概要表某个特定概要图中的移动操作。

接通EN位将激活指令。请确保EN位始终处于接通，直到DONE位指示已经完成了指令的执行。

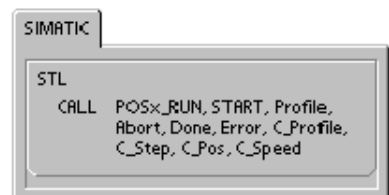
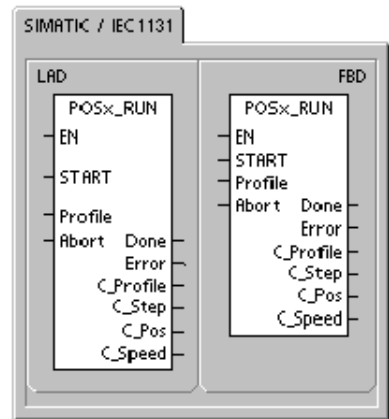
接通参数START将发送RUN命令给位控模块。对每一循环，当参数START接通且位控模块不忙时，该指令将发送一个RUN命令给位控模块。要确保该命令只发送一次，可使用边沿检测单元来以脉冲触发START参数的接通。

Profile参数包含该移动概要图的编号或符号名。也可选择高级移动命令（118至127）。对于移动命令的有关信息，参见表9-19（页面294）。

接通Abort参数将命令位控模块停止当前概要图并减速，直到电机逐渐停止。

参数Done将在模块完成该指令时接通。

参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。



C\_Profile参数包含了位控模块当前正在执行的概要图。

C\_Step参数包含了当前正在执行的概要图的步进。

参数C\_Pos包含了模块的当前位置。基于测量单位，该值既可以是一系列的脉冲（DINT），也可以是一系列的工程单位（REAL）。

参数C\_SPEED包含了模块的当前速度。基于所选测量单位，该值既可以是每秒脉冲数（DINT），也可以是每秒工程单位数（REAL）。

表9-5 POSx\_RUN指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
Profile	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD、常量
Abort、Done	BOOL	I、Q、V、M、SM、S、T、C、L
Error、C_Profile、C_Step	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
C_Pos、C_SPEED	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD

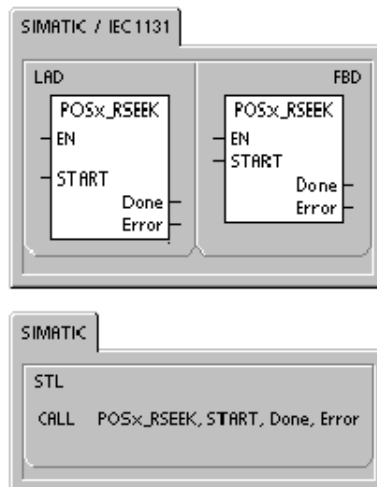
## POSx\_RSEEK指令

POSx\_RSEEK指令（查找参考点位置）将使用配置 / 概要表中的搜索方式，启动参考点搜索操作。当位控模块锁定参考点并且移动停止后，位控模块将把参数RP\_OFFSET的值装载到当前位置，并在CLR输出上生成一个50秒的脉冲。

RP\_OFFSET的缺省值为0。可使用位置控制向导、EM 253控制面板或POSx\_LDOFF（载入偏移量）指令来修改RP\_OFFSET值。

接通EN位将激活指令。请确保EN位始终处于接通，直到DONE位指示已经完成了指令的执行。

接通参数START将向位控模块发送一条RSEEK命令。对于每一循环，当参数START接通且位控模块当前不忙时，该指令将向位控模块发送一条RSEEK指令。要确保该命令只发送一次，可使用边沿检测单元来以脉冲触发START参数的接通。



参数Done将在模块完成该指令时接通。

参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。

表9-6 POSx\_RSEEK指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
Done	BOOL	I、Q、V、M、SM、S、T、C、L
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

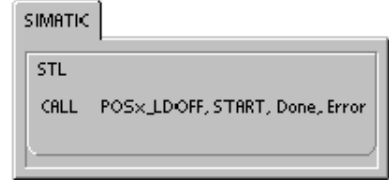
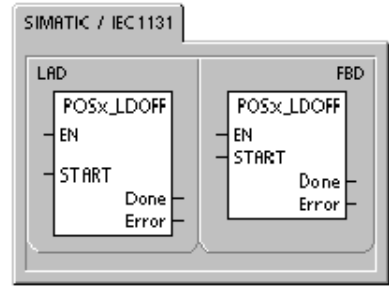
## POSx\_LDOFF指令

POSx\_LDOFF指令（载入参考点偏移）将建立一个新的零位置，该位置与参考点位置不在同一处。

在执行这条指令之前，必须首先确定参考点的位置。此外，还要把机器移动到起始位置。当该指令发送LDOFF命令时，位控模块将计算起始位置（当前位置）与参考点位置之间的偏移量。然后，位控模块将把计算所得的偏移量存储到参数RP\_OFFSET，并将当前位置设置为0。这样，就将起始位置设置成了零位置。

万一出现电机找不到其位置的情况（例如掉电或电机手动重新定位），可使用POSx\_RSEEK指令来自动重建零位置。

接通EN位将激活指令。请确保EN位始终处于接通，直到DONE位指示已经完成了指令的执行。



接通START参数将发送一个LDOFF命令给位控模块。对每一循环，当参数START接通且位控模块当前不忙时，该指令将发送一个LDOFF命令给位控模块。要确保该命令只发送一次，可使用边沿检测单元来以脉冲触发START参数的接通。

参数Done将在模块完成该指令时接通。

参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。

表9-7 POSx\_LDOFF指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
Done	BOOL	I、Q、V、M、SM、S、T、C、L
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

## POSx\_LDPOS指令

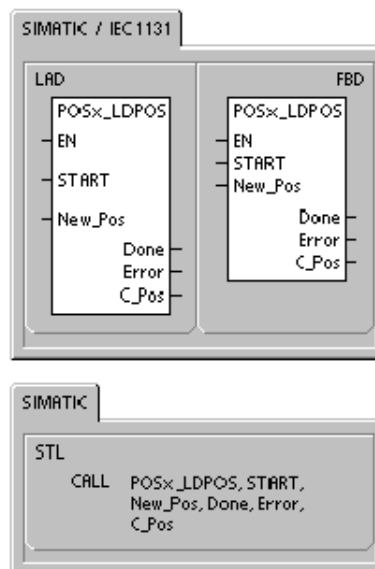
POSx\_LDPOS指令（载入位置）将把位控模块中的当前位置值修改为新的值。也可使用该指令来为任何绝对移动命令建立新的零位置。

接通EN位将激活指令。请确保EN位始终处于接通，直到DONE位指示已经完成了指令的执行。

接通参数START将发送一个LDPOS命令给位控模块。对每一循环，当参数START接通且位控模块当前不忙时，该指令将发送一个LDPOS命令给位控模块。要确保该命令只发送一次，可使用边沿检测单元来以脉冲触发START参数的接通。

New\_Pos参数将提供新的值，以更换位控模块在绝对移动中所报告和使用的当前位置值。基于测量单位，该值既可以是一个脉冲数（DINT），也可以是工程单位数（REAL）。

参数Done将在模块完成该指令时接通。



参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。

参数C\_Pos包含了模块的当前位置。基于测量单位，该值既可以是一系列的脉冲（DINT），也可以是一系列的工程单位（REAL）。

表9-8 POSx\_LDPOS指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
New_Pos、C_Pos	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD
Done	BOOL	I、Q、V、M、SM、S、T、C、L
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

## POSx\_SRATE指令

POSx\_SRATE指令（设置速率）将命令位控模块修改加速、减速和跳动时间。

接通EN位将激活指令。请确保EN位始终处于接通，直到DONE位指示已经完成了指令的执行。

如果接通参数START，则将新的时间值复制到配置 / 概要表，并向位控模块发送一条SRATE命令。对每一循环，当START参数接通并且位控模块当前不忙时，该指令将发送一条SRATE命令到位控模块。要确保该命令只发送一次，可使用边沿检测单元来以脉冲触发START参数的接通。

ACCEL\_TIME、DECEL\_TIME和JERK\_TIME参数将确定以毫秒（ms）为单位的新的加速时间、减速时间和跳动时间。

参数Done将在模块完成该指令时接通。

参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。

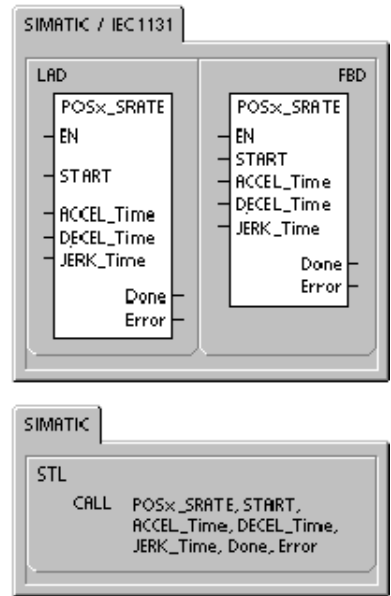


表9-9 POSx\_SRATE指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L
ACCEL_Time、 DECEL_Time、JERK_Time	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD、常量
Done	BOOL	I、Q、V、M、SM、S、T、C、L
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

## POSx\_DIS指令

POSx\_DIS指令可接通或断开位控模块的DIS输出。这允许使用DIS输出来激活或禁止电机控制器。如果使用位控模块上的DIS输出，则该指令可以在每一循环周期中调用，或者仅在需要改变DIS的输出值时调用。

当接通EN位以激活该指令时，参数DIS\_ON将控制位控模块的DIS输出。关于DIS输出的更详细信息，参见表9-1（页面254），或参见附录A中的位控模块的说明。

参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。

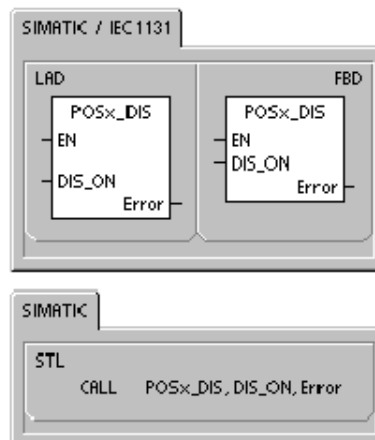


表9-10 POSx\_DIS指令的参数

输入/输出	数据类型	操作数
DIS_ON	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD、常量
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD



## POSx\_CLR指令

POSx\_CLR指令（脉冲CLR输出）将命令位控模块在CLR输出上生成50ms的脉冲。

接通EN位将激活指令。请确保EN位始终处于接通，直到DONE位指示已经完成了指令的执行。

如果接通参数START，则将向位控模块发送一条CLR命令。对于每一循环周期，当参数START接通并且位控模块当前不忙时，该指令将向位控模块发送一条CLR命令。要确保该命令只发送一次，可使用边沿检测单元来以脉冲触发START参数的接通。

参数Done将在模块完成该指令时接通。

参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。

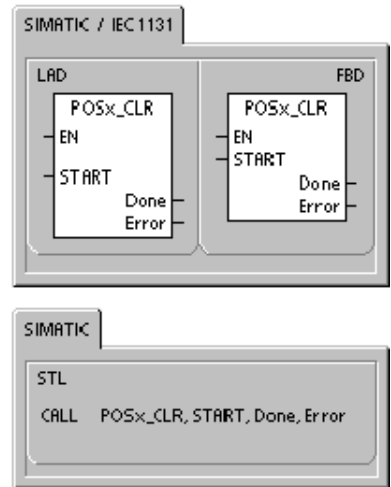


表9-11 POSx\_CLR指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
Done	BOOL	I、Q、V、M、SM、S、T、C、L
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

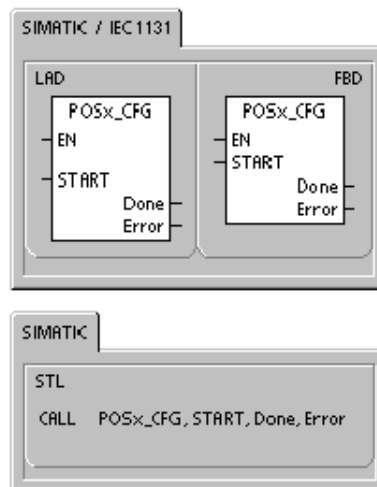
## POSx\_CFG指令

POSx\_CFG指令（重新装载配置）将命令位控模块读取由配置 / 概要表指针所指定位置中的配置块。位控模块随后将把新的配置与现有配置进行比较，并完成所需要的设置修改或重新计算。

接通EN位将激活指令。请确保EN位始终处于接通，直到DONE位指示已经完成了指令的执行。

如果接通参数START，则将向位控模块发送一条CFG命令。对每一循环，当参数START接通且位控模块当前不忙时，该指令将发送一个CFG命令给位控模块。要确保该命令只发送一次，可使用边沿检测单元来以脉冲触发START参数的接通。

参数Done将在模块完成该指令时接通。



参数Error包含了该指令的执行结果。参见表9-13关于出错代码的定义（页面286）。

表9-12 POSx\_CFG指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、功率流
Done	BOOL	I、Q、V、M、SM、S、T、C、L
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

## 位控模块的范例程序

第一个范例程序表示了一个使用POSx\_CTRL和POSx\_GOTO指令完成定长剪切操作的简单相对移动。该程序不需要RP查找模式或移动概要图，且长度测量的单位既可以采用脉冲数，也可以采用工程单位数。输入长度（VD500）和目标速度（VD504）。当I0.0（Start）接通时，机器将启动。当I0.1（Stop）接通时，机器将完成当前的操作后停止。当I0.2（E\_Stop）接通时，机器将中止所有移动，并立即停止。

第二个范例程序提供了POSx\_CTRL、POSx\_RUN、POSx\_RSEEK和POSx\_MAN指令的实例。必须配置RP查找模式和移动概要图。

范例程序1: 简单的相对移动 (定长剪切应用)	
	<pre> 程序段1 //控制指令（插槽0中的模块）。 LD SM0.0 = L60.0 LDN I0.2 = L63.7 LD L60.0 CALL POS0_CTRL, L63.7, M1.0, VB900, VD902, VD906, V910.0  程序段2 //开始将机器置于自动模式 LD I0.0 AN I0.2 EU S Q0.2, 1 S M0.1, 1  程序段3 //E_Stop: 立即停止，并 //断开自动模式。 LD I0.2 R Q0.2, 1  程序段4 //移动到特定点: //输入将要剪切的长度。 //将目标速度输入到参数Speed。 //将模式设置为1（相对模式）。 LD Q0.2 = L60.0 LD M0.1 EU = L63.7 LD L60.0 CALL POS0_GOTO, L63.7, VD500, VD504, 1, I0.2, Q0.4, VB920, VD922, VD926  程序段5 //当处于适当位置时，接通剪切刀具 //2秒钟，以便完成剪切。 LD Q0.2 A Q0.4 TON T33, +200 AN T33 = Q0.3                     </pre>

范例程序1: 简单的相对移动 (定长剪切应用) (续)

<p><b>Network 6</b></p>	<p>程序段6 //当剪切完成后重新启动 //除非Stop激活。</p> <pre> LD    Q0.2 A     T33 LPS AN    I0.1 =     M0.1 LPP A     I0.1 R     Q0.2, 1                     </pre>
-------------------------	---

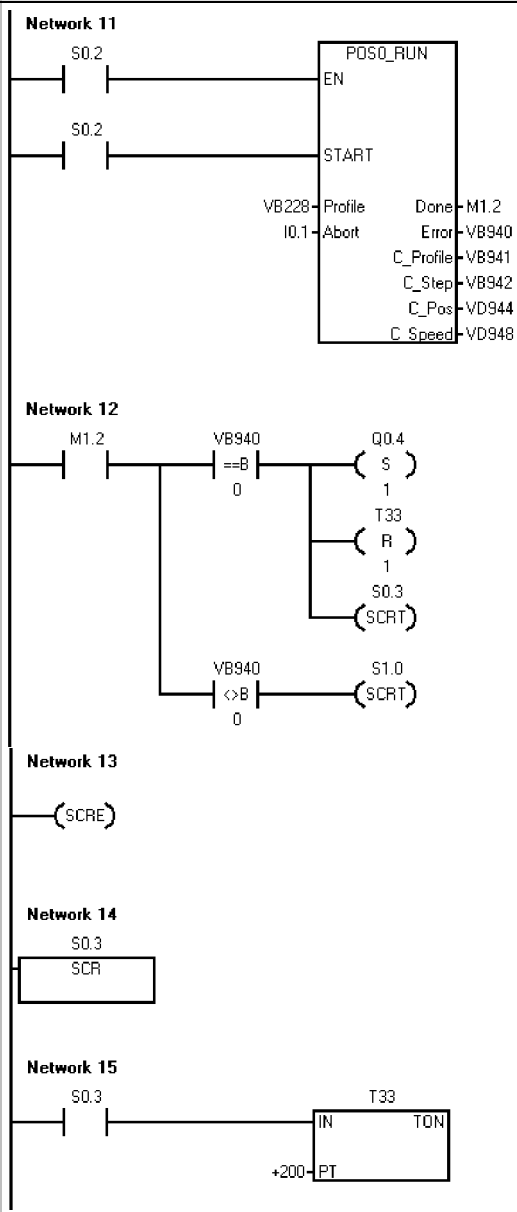
范例程序2: 使用POSx\_CTRL、POSx\_RUN、POSx\_SEEK和POSx\_MAN的程序

<p><b>Network 1</b></p> <p><b>Network 2</b></p> <p><b>Network 3</b></p>	<p>程序段1 //激活位控模块</p> <pre> LD    SM0.0 =     L60.0 LDN   I0.1 =     L63.7 LD    L60.0 CALL  POS0_CTRL, L63.7, M1.0, VB900, VD902,       VD906, V910.0                     </pre> <p>程序段2 //非自动模式时手动模式</p> <pre> LD    I1.0 AN    M0.0 =     L60.0 LD    I1.1 =     L63.7 LD    I1.2 =     L63.6 LD    I1.4 =     L63.5 LD    L60.0 CALL  POS0_MAN, L63.7, L63.6, L63.5, +100000,       I1.5, VB920, VD902, VD906, V910.0                     </pre> <p>程序段3 //激活自动模式</p> <pre> LD    I0.0 EU S     M0.0, 2 S     S0.1, 1 R     S0.2, 8                     </pre>
---	---

范例程序2: 使用POSx\_CTRL、POSx\_RUN、POSx\_SEEK和POSx\_MAN的程序 (续)

<p><b>Network 4</b></p>	<p>程序段4 //紧急停止 //禁用模块和自动模式</p> <pre>LD I0.1 R M0.0, 1 R S0.1, 9 R Q0.3, 3</pre>
<p><b>Network 5</b></p>	<p>程序段5 //处于自动模式时: //打开运行灯</p> <pre>LD M0.0 = Q0.1</pre>
<p><b>Network 6</b></p>	<p>程序段6</p> <pre>LSCR S0.1</pre> <p>程序段7 //查找参考点 (RP)</p> <pre>LD S0.1 = L60.0 LD S0.1 = L63.7 LD L60.0 CALL POS0_RSEEK, L63.7, M1.1, VB930</pre>
<p><b>Network 7</b></p>	<p>程序段8 //位于参考点 (RP) 时: //夹住材料, 并跳转到下一个步进。</p> <pre>LD M1.1 LPS AB= VB930, 0 S Q0.3, 1 SCRT S0.2 LPP AB&lt;&gt; VB930, 0 SCRT S1.0</pre>
<p><b>Network 8</b></p>	<p>程序段9</p> <pre>SCRE</pre> <p>程序段10</p> <pre>LSCR S0.2</pre>
<p><b>Network 9</b></p>	
<p><b>Network 10</b></p>	

范例程序2: 使用POSx\_CTRL、POSx\_RUN、POSx\_SEEK和POSx\_MAN的程序 (续)



```

程序段11 //使用概要图1移动到位置。
LD S0.2
= L60.0
LD S0.2
= L63.7
LD L60.0
CALL POS0_RUN, L63.7, VB228, IO.1, M1.2, VB940,
VB941, VB942, VD944, VD948
    
```

```

程序段12 //在定位后, 接通剪切工具
//并跳转到下一个步进。
    
```

```

LD M1.2
LPS
AB= VB940, 0
S Q0.4, 1
R T33, 1
SCRT S0.3
LPP
AB<> VB940, 0
SCRT S1.0
    
```

```

程序段13
SCRE
    
```

```

程序段14 //等待完成剪切
LSCR S0.3
    
```

```

程序段15
LD S0.3
TON T33, +200
    
```

范例程序2: 使用POSx\_CTRL、POSx\_RUN、POSx\_SEEK和POSx\_MAN的程序 (续)

<p><b>Network 16</b></p>	<p>程序段16 //除非STOP接通, 否则, 将在 //剪切完成后重新启动。</p> <pre>LD T33 LPS R Q0.3, 1 R Q0.4, 1 AN I0.2 SCRT S0.1 LPP A I0.2 R M0.0, 4</pre>
<p><b>Network 17</b></p>	<p>程序段17 SCRE</p>
<p><b>Network 18</b></p>	<p>程序段18 LSCR S1.0</p>
<p><b>Network 19</b></p>	<p>程序段19 //重新设置输出。</p> <pre>LD S1.0 R Q0.3, 2</pre>
<p><b>Network 20</b></p>	<p>程序段20 //闪烁错误灯。</p> <pre>LD SM0.5 = Q0.5</pre>
<p><b>Network 21</b></p>	<p>程序段21 //如果STOP接通, 则退出错误例行程序。</p> <pre>LD I0.2 R M0.0, 9 R S0.1, 8</pre>
<p><b>Network 22</b></p>	<p>程序段22 SCRE</p>

## 使用EM 253控制面板监控位控模块

为帮助开发位置控制解决方案，STEP 7-Micro/WIN提供了EM 253控制面板。“操作”、“配置”和“诊断”标签将使您在开发过程的启动和测试阶段，对位控模块的运行进行方便的监控和控制。

使用EM 253控制面板来检验位控模块的布线是否正确，以便调整配置数据，并测试每个移动概要图。

### 显示和控制位控模块的操作

控制面板的“操作”标签允许与位控模块的操作进行交互。控制面板显示位控模块的当前速度、当前位置和当前方向。也可看见输入和输出LED的状态（不包括脉冲LED）。

控制面板允许通过改变速度和方向、停止和启动移动以及慢进工具（如果移动停止）来与位控模块进行交互。

也可生成下列移动命令：

- 启用“手动运行”。该命令允许使用手动控件对工具进行定位。
- 运行移动概要图。该命令允许选择要执行的概要图。控制面板将显示位控模块正在执行的概要图的状态。
- 搜索参考点。该命令将使用所配置的查找模式对参考点进行检索。

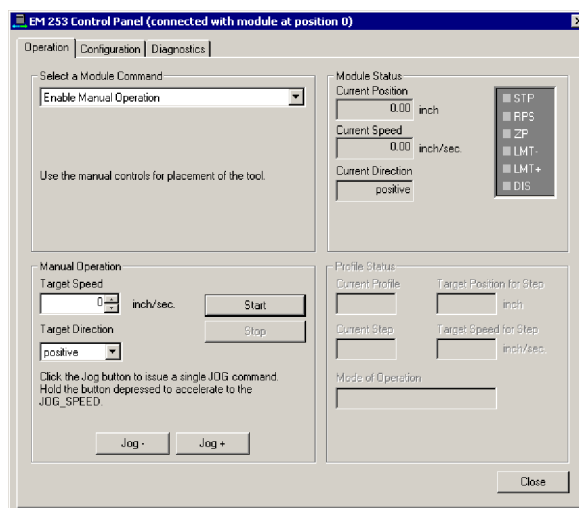


图9-19 EM253控制面板的运行标签

- 载入参考点偏移。在使用手动控件将工具慢进到新的零位置后，载入参考点偏移。
- 重新装载当前位置。该命令将更新当前位置值，并建立新的零位置。
- 激活DIS输出并取消激活DIS输出。这些命令接通和断开位控模块的DIS输出。
- 脉冲触发CLR输出。该命令将在位控模块的CLR输出上生成一个50ms的脉冲。
- 示教移动概要图。该命令允许在对工具进行手动定位时，保存移动概要图和步进的目标位置和速度。控制面板将显示位控模块正在执行的概要图的状态。
- 载入模块配置。该命令将通过命令位控模块读取S7-200 V内存中的配置块来载入新的配置。
- 移动到绝对位置。该命令将允许以目标速度移动到指定的位置。在使用该命令之前，必须已经建立了零位置。
- 移动相对量。该命令允许以目标速度从当前位置移动指定的距离。可输入正的或负的距离。
- 重设命令接口。该命令清除位控模块的命令字节，设置Done位。如果位控模块似乎对命令没有响应，可使用该命令。



### 显示和修改位控模块的配置

控制面板的“配置”标签允许查看和修改存储在S7-200数据块中的位控模块的配置设置。

在修改配置设置之后，只需单击按钮即可更新STEP 7-Micro/Win项目和S7-200数据块二者中的设置。

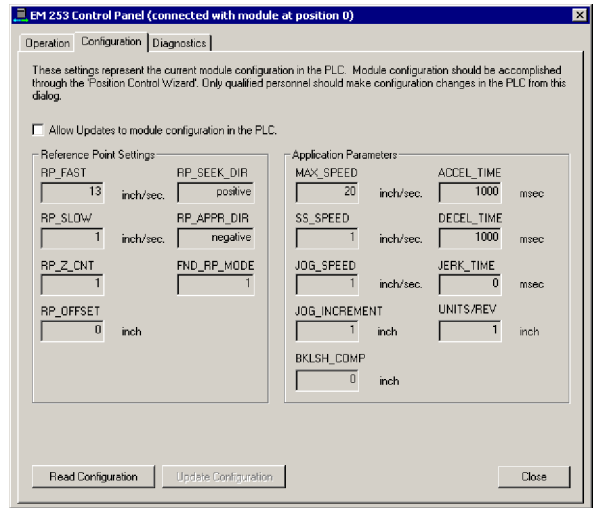


图9-20 EM253控制面板的配置标签

### 显示位控模块的诊断信息

控制面板的“诊断”标签允许查看位控模块的有关诊断信息。

可查看位控模块的特定信息，例如I/O链中的模块位置、模块型号和固化程序版本号以及作为模块命令字节使用的输出字节。

控制面板将显示所控制的操作导致的错误条件。关于指令错误条件，请参见表9-13（页面286）。

也可查看位控模块所报告的错误条件。关于模块错误条件，请参见表9-14（页面287）。

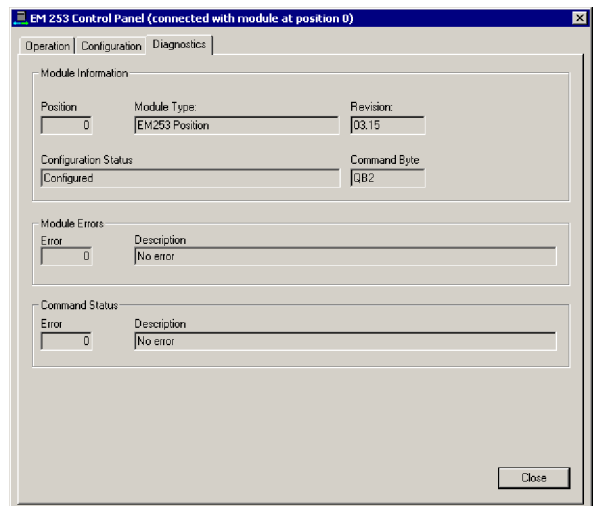


图9-21 EM253控制面板的诊断标签

## 位控模块和位控指令的出错代码

表9-13 指令出错代码

出错代码	描述
0	无错
1	被用户中止
2	配置出错 使用EM 253控制面板诊断标签浏览出错代码
3	非法的命令
4	由于没有任何有效配置而中止 使用EM 253控制面板诊断标签浏览出错代码
5	由于没有任何用户电源而中止
6	由于没有任何已定义的参考点而中止
7	由于STP输入有效而中止
8	由于LMT-输入有效而中止
9	由于LMT+输入有效而中止
10	由于执行移动时存在问题而中止
11	没有为指定的概要图配置任何概要块
12	非法的运行模式
13	该命令不支持的运行模式
14	概要块中的非法步进数
15	非法的方向改变
16	非法距离
17	在达到目标速度之前发生了RPS触发
18	没有足够的RPS有效区宽度
19	速度超出了范围
20	完成所期望速度改变的距离不够
21	非法位置
22	零位置未知
23至127	保留
128	位控模块不能处理该指令：位控模块正忙于执行另一个指令，或者该指令上不存在任何启动脉冲
129	位控模块出错：模块标识号不正确或模块已经注销。参见其他错误条件的SMB8至SMB21（I/O模块标识号和出错寄存器）。
130	位控模块未激活
131	由于模块出错或模块未激活而使位控模块不能使用 （参见POSx_CTRL状态）
132	位置控制向导配置的Q内存地址与该位置的模块内存地址不匹配。

表9-14 模块出错代码

出错代码	描述
0	无错
1	无任何用户电源
2	配置块未出现
3	配置块指针出错
4	配置块的大小超出了可用的V内存
5	非法的配置块格式
6	指定的概要图太多
7	非法的STP_RSP规定
8	非法的LMT-_RPS规定
9	非法的LMT+_RPS规定
10	非法的FILTER_TIME规定
11	非法的MEAS_SYS规定
12	非法的RP_CFG规定
13	非法的PLS/REV数值
14	非法的UNITS/REV数值
15	非法的RP_ZP_CNT数值
16	非法的JOG_INCREMENT数值
17	非法的MAX_SPEED数值
18	非法的SS_SPD数值
19	非法的RP_FAST数值
20	非法的RP_SLOW数值
21	非法的JOG_SPEED数值
22	非法的ACCEL_TIME数值
23	非法的DECEL_TIME数值
24	非法的JERK_TIME数值
25	非法的BKLSH_COMP数值

## 高级主题

### 了解配置 / 概要表

位置控制向导可在您对位置控制系统作出的应答的基础上自动生成配置和概要信息，从而使移动应用简单方便。配置 / 概要表信息主要提供给那些需要创建其自己的位置控制例行程序的高级用户。

配置 / 概要表位于S7-200的V内存中。如表9-15中所示，配置设置存储在下列类型的信息中：

- 配置块包含有用于对模块进行设置的信息，以便为执行移动命令作准备。
- 交互块支持用户程序直接设置移动参数。
- 各个概要块描述了位控模块将要完成的预定义移动操作。最多可配置25个概要块。



#### 提示

为创建多于25个移动概要图，通过修改存储在配置 / 概要表指针中的数值，可调换配置 / 概要表。

表9-15 配置 / 概要表

偏移量	名称	功能描述	类型
<b>配置块</b>			
0	MOD_ID	模块标识域	--
5	CB_LEN	以字节为单位的配置块长度（1个字节）	--
6	IB_LEN	以字节为单位的交互块长度（1个字节）	--
7	PF_LEN	以字节为单位的单个概要图长度（1个字节）	--
8	STP_LEN	以字节为单位的单步进长度（1个字节）	--
9	STEPS	每个概要图所允许的步进数（1个字节）	--
10	概要图	从0至25的概要图数（1个字节）	--
11	保留	设置为0x0000	--
13	IN_OUT_CFG	指定模块输入和输出的使用 (1个字节)	--

MSB 7	6	5	4	3	2	1	LSB 0
P/D	POL	0	0	STP	RPS	LMT-	LMT+

P/D 该位指定P0和P1的使用。  
 正极（POL=0）：  
 0- P0 用于正向旋转的脉冲  
 P1 用于负向旋转的脉冲  
 1- P0 用于旋转的脉冲  
 P1控制旋转方向（0-正向、1-负向）  
 负极（POL=1）：  
 0- P0 用于正向旋转的脉冲  
 P1 用于负向旋转的脉冲  
 1- P0 用于旋转的脉冲  
 P1控制旋转方向（0-正向、1-负向）

POL 该位将选择P0和P1的极性约定。  
 （0-正极、1-负极）

STP 该位控制停止输入的现用等级。

RPS 该位控制RPS输入的现用等级。

LMT- 该位控制负向移动限制输入的现用等级。

LMT+ 该位控制正向移动限制输入的现用等级。

0- 现用高  
 1- 现用低

表9-15 配置 / 概要表 (续)

偏移量	名称	功能描述	类型																				
14	STP_RSP	指定STP输入的驱动器响应 (1个字节) 0 没有任何动作。忽略输入条件。 1 减速直至停止, 并指示STP输入已激活。 2 终止脉冲, 并指示STP输入 3至255 保留 (如果指定, 则出错)	--																				
15	LMT-_RSP	指定负向限制输入的驱动器响应 (1个字节) 0 没有任何动作。忽略输入条件。 1 减速直至停止, 并指示已经达到限制。 2 终止脉冲, 并指示已经达到限制。 3至255 保留 (如果指定, 则出错)	--																				
16	LMT+_RSP	指定正向限制输入的驱动器响应 (1个字节) 0 没有任何动作。忽略输入条件。 1 减速直至停止, 并指示已经达到限制。 2 终止脉冲, 并指示已经达到限制。 3至255 保留 (如果指定, 则出错)	--																				
17	FILTER_TIME	指定STP、LMT-、LMT+和RPS输入的过滤时间 (1字节) <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">LSB</td> <td style="text-align: center;">0</td> </tr> <tr> <td colspan="5" style="text-align: center;">STP、LMT-、LMT+</td> <td colspan="5" style="text-align: center;">RPS</td> </tr> </table> </div> '0000' 200 μsec    '0101' 3200 μsec '0001' 400 μsec    '0110' 6400 μsec '0010' 800 μsec    '0111' 12800 μsec '0011' 1600 μsec    '1000' 无过滤 '0100' 1600 μsec    '1001' 至 '1111'保留 (如果指定, 则出错)	MSB	7	6	5	4	3	2	1	LSB	0	STP、LMT-、LMT+					RPS					--
MSB	7	6	5	4	3	2	1	LSB	0														
STP、LMT-、LMT+					RPS																		
18	MEAS_SYS	指定测量系统 (1个字节) 0 脉冲数 (按每秒的脉冲数来对速度进行测量, 按脉冲数对位置值进行测量)。数值均存储为DINT。 1 工程单位数 (按每秒的单位数来对速度进行测量, 按单位数对位置值进行测量)。数值均存储为单精度REAL。 2至255 保留 (如果指定, 则出错)	--																				
19	--	保留 (设置为0)	--																				
20	PLS/REV	指定电机每个旋转周期的脉冲数 (4个字节) 只有在MEAS_SYS设置为1时才能适用。	DINT																				
24	UNITS/REV	指定电机每个旋转周期的工程单位数 (4个字节) 只有在MEAS_SYS设置为1时才能适用。	REAL																				
28	单元	为STEP 7-Micro/WIN保留以存储自定义单位字符串 (4个字节)	--																				
32	RP_CFG	指定参考点搜索配置 (1个字节) <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">LSB</td> <td style="text-align: center;">0</td> </tr> <tr> <td colspan="2"></td> <td colspan="2" style="text-align: center;">0 0</td> <td colspan="6" style="text-align: center;">Mode</td> </tr> </table> </div> RP_SEEK_DIR 该位指定参考点搜索的启动方向。 (0-正方向、1-负方向) RP_APPR_DIR 该位指定终止参考点搜索时的接近方向。 (0-正方向、1-负方向) MODE 指定参考点搜索方法。 '0000' 参考点搜索禁用。 '0001' 参考点位于激活RPS输入的地方。 '0010' 参考点位于RPS输入激活区的中央。 '0011' 参考点位于RPS输入激活区之外。 '0100' 参考点位于RPS输入激活区之内。 '0101'至'1111'保留 (如果选择, 则出错)	MSB	7	6	5	4	3	2	1	LSB	0			0 0		Mode						--
MSB	7	6	5	4	3	2	1	LSB	0														
		0 0		Mode																			
33	--	保留 (设置为0)	--																				
34	RP_Z_CNT	用于定义参考点的ZP输入的脉冲数 (4个字节)	DINT																				

表9-15 配置 / 概要表 (续)

偏移量	名称	功能描述	类型
38	RP_FAST	用于RP搜索操作的高速度: MAX_SPD或以下 (4字节)	DINT REAL
42	RP_SLOW	用于RP搜索操作的低速度: 电机可立即跳转到停止的最快速度或以下 (4字节)	DINT REAL
46	SS_SPEED	启动 / 停止速度。(4字节) 启动速度是电机可立即从停止开始跳转的最快速度以及电机可立即跳转到停止的最快速度。允许低于该速度运行, 但不适用于加速和减速时间。	DINT REAL
50	MAX_SPEED	电机的最大运行速度 (4字节)	DINT REAL
54	JOG_SPEED	慢进速度。MAX_SPEED或以下 (4字节)	
58	JOG_INCREMENT	慢进递增数值是响应单个慢进脉冲所移动的距离 (或脉冲数)。(4字节)	DINT REAL
62	ACCEL_TIME	以毫秒为单位的从最小速度加速到最大速度所需要的时间 (4字节)	DINT
66	DECEL_TIME	以毫秒为单位的从最大速度减速到最小速度所需要的时间 (4字节)	DINT
70	BKLSH_COMP	间隙补偿: 用于在方向变化时对系统反冲进行补偿所使用的距离 (4字节)	DINT REAL
74	JERK_TIME	在对加速 / 减速曲线 (S曲线) 的开始和终端部分进行跳动补偿期间的的时间。指定数值0为禁用跳动补偿。跳动时间的单位为毫秒。(4字节)	DINT
<b>交互块</b>			
78	MOVE_CMD	选择运行模式 (1个字节) 0 绝对位置 1 相对位置 2 单速、连续运行、正向旋转 3 单速、连续运行、负向旋转 4 手动速度控制、正向旋转 5 手动速度控制、负向旋转 6 单速、连续运行、具有触发停止的正向旋转 (RPS输入信号停止) 7 单速、连续运行、具有触发停止的负向旋转 (RPS输入信号停止) 8至255 保留 (如果指定, 则出错)	--
79	--	保留。设置为0	--
80	TARGET_POS	该移动中要跳转到的目标位置 (4字节)	DINT REAL
84	TARGET_SPEED	该移动的目标速度 (4字节)	DINT REAL
88	RP_OFFSET	参考点的绝对位置 (4字节)	DINT REAL
<b>概要块0</b>			
92 (+0)	STEPS	该移动序列中的步进数 (1个字节)	--
93 (+1)	Mode	选择该概要块的操作模式 (1个字节) 0 绝对位置 1 相对位置 2 单速、连续运行、正向旋转 3 单速、连续运行、负向旋转 4 保留 (如果指定, 则出错) 5 保留 (如果指定, 则出错) 6 单速、连续运行、具有触发停止的正向旋转 (RPS选择速度) 7 单速、连续运行、具有触发停止的负向旋转 (RPS输入信号停止) 8 双速、连续运行、正向旋转 (RPS选择速度) 9 双速、连续运行、负向旋转 (RPS选择速度) 10至255 保留 (如果指定, 则出错)	--

表9-15 配置/概要表(续)

偏移量	名称		功能描述	类型
94 (+2)	0	POS	移动步进0中要跳转到的位置(4个字节)	DINT REAL
98 (+6)		SPEED	移动步进0的目标速度(4个字节)	DINT REAL
102 (+10)	1	POS	移动步进4中要跳转到的位置(4个字节)	DINT REAL
106 (+14)		SPEED	移动步进1的目标速度(4个字节)	DINT REAL
110 (+18)	2	POS	移动步进2中要跳转到的位置(4个字节)	DINT REAL
114 (+22)		SPEED	移动步进2的目标速度(4个字节)	DINT REAL
118 (+26)	3	POS	移动步进3中要跳转到的位置(4个字节)	DINT REAL
122 (+30)		SPEED	移动步进3的目标速度(4个字节)	DINT REAL
<b>概要块1</b>				
126 (+34)	STEPS		该移动序列中的步进数(1个字节)	--
127 (+35)	MODE		选择该概要块的操作模式(1个字节)	--
128 (+36)	0	POS	移动步进0中要跳转到的位置(4个字节)	DINT REAL
132 (+40)		SPEED	移动步进0的目标速度(4个字节)	DINT REAL
...	...	...	...	...

### 位控模块的特殊内存位置

基于I/O系统中模块的物理位置，S7-200将分配50个字节的特殊内存(SM)给每个智能模块。参见表9-16。当模块检测到错误条件或数据状态发生改变时，模块将更新这些SM位置。第一个模块根据需要更新SMB200至SMB249来报告错误条件，第二个模块更新SMB250至SMB299，依此类推。

**注意**

为智能模块分配SM位置的方式对V2.2及以后的版本将有所不同。

如果现在使用的是V2.2以前的CPU，则应将所有智能模块放置在邻近CPU的插槽中，同时确保所有非智能模块的兼容性。

表9-16 特殊内存字节SMB200至SMB549

位于下列插槽的智能模块的SM字节:						
插槽0	插槽1	插槽2	插槽3	插槽4	插槽5	插槽6
SMB200至 SMB249	SMB250至 SMB299	SMB300至 SMB349	SMB350至 SMB399	SMB400至 SMB449	SMB450至 SMB499	SMB500至 SMB549

表9-17显示了为智能模块所分配的SM数据区的结构。按照这种方式所给出的定义就好像智能模块就位于I/O系统的插槽0中。

表9-17 EM 253位控模块的特殊内存区定义

SM地址	描述																								
SMB200至SMB215	模块名称（16个ASCII字符）。SMB200是第一个字符：“EM 253位置”																								
SMB216至SMB219	软件修订号（4个ASCII字符）。SMB216是第一个字符。																								
SMW220	模块的出错代码。参见表9-14的出错代码描述。																								
SMB222	<p>输入 / 输出状态。反映了模块的输入和输出状态。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="4">MSB</th> <th colspan="4">LSB</th> </tr> <tr> <th>7</th> <th>6</th> <th>5</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>Dis</td> <td>0</td> <td>0</td> <td>STP</td> <td>LMT-</td> <td>LMT+</td> <td>RPS</td> <td>ZP</td> </tr> </tbody> </table> <p>DIS 禁用输出                      0=无电流                      1=有电流                      STP 停止输入                      0=无电流                      1=有电流                      LMT- 负向移动限制输入            0=无电流                      1=有电流                      LMT+ 正向移动限制输入            0=无电流                      1=有电流                      RPS 参考点开关输入                0=无电流                      1=有电流                      ZP 零脉冲输入                      0=无电流                      1=有电流</p>	MSB				LSB				7	6	5	4	3	2	1	0	Dis	0	0	STP	LMT-	LMT+	RPS	ZP
MSB				LSB																					
7	6	5	4	3	2	1	0																		
Dis	0	0	STP	LMT-	LMT+	RPS	ZP																		
SMB223	<p>瞬时模块状态。反映了模块配置状态以及旋转方向状态。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="4">MSB</th> <th colspan="4">LSB</th> </tr> <tr> <th>7</th> <th>6</th> <th>5</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>OR</td> <td>R</td> <td>CFG</td> </tr> </tbody> </table> <p>OR 目标速度超出范围                0=在范围内                      1=超出范围                      R 旋转方向                            0=正向旋转                      1=负向旋转                      CFG 模块已配置                        0=未配置                        1=已配置</p>	MSB				LSB				7	6	5	4	3	2	1	0	0	0	0	0	0	OR	R	CFG
MSB				LSB																					
7	6	5	4	3	2	1	0																		
0	0	0	0	0	OR	R	CFG																		
SMB224	CUR_PF为一个字节，用于指示当前正在执行的概要图。																								
SMB225	CUR_STP为一个字节，用于指示概要图中当前正在执行的步进。																								
SMD226	CUR_POS为一个双字数值，用于指示模块的当前位置。																								
SMD230	CUR_SPD为一个双字数值，用于指示模块的当前速度。																								
SMB234	<p>指令的结果。参见表9-13的出错代码描述。127以上的错误条件由向导所创建的指令子例行程序生成。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">MSB</th> <th colspan="2">LSB</th> </tr> <tr> <th>7</th> <th>6</th> <th colspan="2">0</th> </tr> </thead> <tbody> <tr> <td>D</td> <td colspan="3">错误</td> </tr> </tbody> </table> <p>D Done（完成）位                      0=操作正在进行                      1=操作完成（由模块在启动期间进行设置）</p>	MSB		LSB		7	6	0		D	错误														
MSB		LSB																							
7	6	0																							
D	错误																								
SMB235至SMB244	保留																								
SMB245	用作该模块命令接口的第一个Q字节的偏移量。为方便用户，偏移量由S7-200自动提供，且模块不需要偏移量。																								
SMD246	指向配置 / 概要表V内存位置的指针。指向V内存以外区域的指针值无效。位控模块将监控该位置，直到其接收到一个非零指针值。																								



## 了解位控模块的命令字节

位控模块将提供一个字节的离散输出，它将用作命令字节。图9-22显示命令字节定义。表9-18显示Command\_code的定义。

对R位从0改变为1的命令字节的写操作将被模块解释为一个新的命令。

如果模块检测到在命令有效时转换到闲置状态（R位状态改变为0），则正在进行的操作将被中止，如果正在移动，则进行减速停止。



图9-22 命令字节的定义

完成操作后，模块在接受新命令之前必须确认已转换到闲置状态。如果运行中止，则模块在接受新命令之前必须完成任何减速。在命令处于激活状态时，将忽略Command\_code数值上的任何变化。

位控模块对S7-200操作模式变化的响应或对错误条件的响应都取决于S7-200根据现有S7-200函数定义对离散输出所发挥的作用：

- 如果S7-200从STOP切换到RUN：S7-200中的程序将控制位控模块的操作。
- 如果S7-200从RUN切换到STOP：可选择离散输出在转换为STOP时将要达到的状态，或输出将要保留其前一个状态时所处的状态。
  - 如果R位在跳转到STOP时断开：位控模块将减小处于进行过程中的移动速度，直到停止。
  - 如果R位在跳转到STOP时接通：位控模块将完成正在执行中的任何命令。如果没有任何命令处于执行状态，则位控模块将执行Command\_code位所指定的命令。
  - 如果R位保留最近的状态：位控模块将完成处于进行过程中的任何移动。
- 如果S7-200检测到严重错误并断开所有离散输出：位控模块将减小处于进行过程中的移动速度，直到停止。

位控模块具有一个监视程序计时器，用于在与S7-200的通讯丢失时断开输出。如果输出监视程序计时器用完规定的时间，则位控模块将减小进行中的移动速度，直到停止。

如果检测到模块硬件或固件中严重错误，则位控模块将把P0、P1、DIS和CLR输出设置为非激活状态。

表9-18 Command\_code定义

Command_code	命令
000 0000至 000 1111	0至 24 执行概要块0至24中所指定的移动
100 0000至 111 0101	25至 117 保留 (如果指定，则出错)
1110110	118 激活DIS输出
1110111	119 取消激活DIS输出
1111000	120 脉冲触发CLR输出
1111001	121 重新装载当前位置
1111010	122 执行交互块中所指定的移动
1111011	123 捕获参考点偏移量
1111100	124 慢进正向旋转
1111101	125 慢进负向旋转
1111110	126 搜索参考点位置
1111111	127 重新装载配置

表9-19 移动命令

命令	描述
命令0至24: <i>执行概要块 0 至 24 中所指定的移动</i>	<p>执行该命令时，位控模块进行由命令的Command_code部分所指出的概要块MODE域中所指定的移动操作。</p> <ul style="list-style-type: none"> <li>在模式0（绝对位置）中，移动概要块将定义1到4个步进，每个步进包含有描述移动区间的位置（POS）和速度（SPEED）这两部分。POS规定代表了绝对位置，它以指定为参考点的位置为基础，移动方向取决于当前位置与概要图中第一个步进的位置之间的关系。在多步进移动中，将禁止负向移动，这将导致报告错误条件。</li> <li>在模式1（相对位置）中，移动概要块将定义1到4个步进，每个步进包含有描述移动区间的位置（POS）和速度（SPEED）这两部分。位置值（POS）的符号将确定移动的方向。在多步进移动中，将禁止负向移动，这将导致报告错误条件。</li> <li>在模式2和3（单速、连续运行模式）中，将忽略位置（POS），且模块将加速到第一个步进的SPEED域所指定的速度。模式2用于正向旋转，模式3用于负向旋转。命令字节转换为闲置状态时，移动停止。</li> <li>在模式6和7（具有已触发停止的单速、连续运行模式）中，模块将加速到第一个步进的SPEED域所指定的速度。如果RPS输入变为有效，则移动将在完成在第一个步进的POS域中所指定的距离之后停止。（POS域中所指定的距离必须包括减速距离。）如果在RPS输入变为有效时POS域为零，则位控模块将减速，直到停止。模式6用于正向旋转，模式7用于负向旋转。</li> <li>在模式8和9中，RPS输入的二进制数值将选择概要块中最前面两个步进所指定的两个速度值中的一个。 <ul style="list-style-type: none"> <li>如果RPS处于未激活状态：步进0将控制驱动器的速度。</li> <li>如果RPS处于激活状态：步进1将控制驱动器的速度。</li> </ul>           模式8用于正向旋转，模式9用于负向旋转。SPEED值将控制移动速度。POS值在该模式下将被忽略。 </li> </ul>
命令118 <i>激活 DIS 输出</i>	执行该命令时，位控模块将激活DIS输出。
命令119 <i>取消激活 DIS 输出</i>	当该命令执行时，位控模块将取消激活DIS输出。
命令120 <i>脉冲触发 CLR 输出</i>	当该命令执行时，位控模块将在CLR输出上生成一个50毫秒的脉冲。
命令121 <i>重新装载当前位置</i>	当该命令执行时，位控模块将把当前位置设置为交互块的TARGET_POS域中所发现的数值。
命令122 <i>执行交互块中所指定的移动</i>	<p>当该命令执行时，位控模块将执行交互块的MOVE_CMD域中所指定的移动操作。</p> <ul style="list-style-type: none"> <li>在模式0和1（绝对和相对移动模式）中，将以交互块的TARGET_SPEED和TARGET_POS域中所提供的目标速度和位置信息为基础，完成单步进移动。</li> <li>在模式2和3（单速、连续运行模式）中，将忽略位置规定，且位控模块将加速到交互块的TARGET_SPEED域所指定的速度。当命令字节转换为闲置状态时，移动将停止。</li> <li>在模式4和5（手动速度控制模式）中，忽略位置技术规定，并且程序将速度变化值载入到交互块的TARGET_SPEED域中。位控模块将连续监控该位置，并在速度值改变时适时作出响应。</li> </ul>

表9-19 移动命令 (续)

命令	描述
命令123 <i>捕获参考点偏移量</i>	<p>当该命令执行时，位控模块将建立一个零位置，该位置位于与参考点位置不同的位置。</p> <p>在发出该命令之前，必须已经确定了参考点的位置，并且机器已慢进到工作启动位置。在接收该命令之后，位控模块将计算工作启动位置（当前位置）和参考点位置之间的偏移量，并将计算得到的偏移量写入交互块的RP_OFFSET域中。随后，当前位置将被设置为0，以便将工作启动位置确定为零位置。</p> <p>万一步进电机找不到其位置（例如，如果电源断电或手动重新定位步进电机），则可发出“搜索参考点位置”命令，以便自动重新建立零位置。</p>
命令124 <i>慢进正向旋转</i>	<p>该命令允许手动发出正向移动步进电机的脉冲。</p> <p>如果该命令保持有效的时间少于0.5秒，则位控模块将发出脉冲，以按照JOG_INCREMENT中所指定的距离移动。</p> <p>如果命令保持有效的时间0.5秒或更长，则移动模块将开始加速到指定的JOG_SPEED。</p> <p>当检测到转换为闲置状态时，位控模块将减速，直到停止。</p>
命令125 <i>慢进负向旋转</i>	<p>该命令允许手动发出反向移动步进电机的脉冲。</p> <p>如果该命令保持有效的时间少于0.5秒，则位控模块将发出脉冲，以按照JOG_INCREMENT中所指定的距离移动。</p> <p>如果命令保持有效的时间0.5秒或更长，则位控模块将开始加速到指定的JOG_SPEED。</p> <p>当检测到转换为闲置状态时，位控模块将减速，直到停止。</p>
命令126 <i>搜索参考点位置</i>	<p>当该命令执行时，位控模块将使用所指定的搜索方法启动参考点搜索操作。当参考点的位置已经找到，且移动停止时，位控模块将把从交互块的RP_OFFSET域中所读出的数值载入当前位置，并使用脉冲在CLR输出上触发一个50毫秒的脉冲。</p>
命令127 <i>重新载入配置</i>	<p>当该命令执行时，位控模块将从SM内存中的相应位置读出配置/概要表指针，然后从配置/概要表指针所指定的位置中读出配置块。位控模块将把刚刚获得的配置数据与现有的模块配置进行对比，并完成所需要的任何设置修改或重新计算。任何高速缓存的概要图均将丢弃。</p>

## 了解位控模块的概要图高速缓存

位控模块将把多达4个概要图的执行数据存储于高速缓存中。当位控模块接收到执行概要图的命令时，将检查所请求的概要图是否存储在高速缓存中。如果概要图的执行数据已经驻留在高速缓存中，则位控模块将立即执行概要图。如果概要图的执行数据没有驻留在高速缓存中，则位控模块将从S7-200配置/概要表中读出概要图信息，并在执行概要图之前计算概要图的执行数据。

命令122（执行交互块中所指定的移动）不使用高速缓存来存储执行数据，而是始终从S7-200配置/概要表中读取交互块，并计算移动的执行数据。

重新配置位控模块将删除所有存储在高速缓存中的执行数据。

## 创建自己的定位控制指令

位置控制向导将创建控制位控模块操作的定位指令；然而，也可创建自己的指令。下列STL代码段提供了一个实例，即如何创建自己的位控模块控制指令。

该实例使用位控模块位于插槽0的S7-200 CPU 224。位控模块将在加电时进行配置。CMD\_STAT是用于SMB 234的符号，CMD是用于QB2的符号，而NEW\_CMD则是用于概要图的符号。

### 范例程序：控制位控模块

```

程序段1      //新的移动命令状态
LSCR      State_0

程序段2      //CMD_STAT是用于SMB 234的符号
            //CMD是用于QB2的符号
            //NEW_CMD是用于概要图的符号。
            //
            //1.清除位控模块的Done（完成）位。
            //2.清除位控模块的命令字节。
            //3.发出新的命令。
            //4.等待执行命令。

LD      SM0.0
MOVB   0, CMD_STAT
BIW    0, CMD
BIW    NEW_CMD, CMD
SCRT   State_1

程序段3
SCRE

程序段4      //等待命令完成。
LSCR      State_1

程序段5      //如果命令完成时没有出现错误，则跳转到闲置状态。
LDB=     CMD_STAT, 16#80
SCRT    Idle_State

程序段6      //如果命令完成时出现错误，则跳转到错误处理状态。
LDB>    CMD_STAT, 16#80
SCRT    Error_State

程序段7
SCRE

```

# 10

## 创建用于调制解调器模块的程序

使用EM 241调制解调器模块可以直接将S7-200连接到一条模拟电话线上，并且支持S7-200与STEP 7-Micro/WIN之间的通讯。该调制解调器模块还支持Modbus从站RTU协议。调制解调器模块与S7-200之间的通讯将通过扩展I/O总线实现。

STEP 7-Micro/WIN 提供了调制解调器扩充向导，以帮助设置一个远程调制解调器或者将本地S7-200连接到远程设备的调制解调器模块。

### 本章内容

调制解调器模块的特性 .....	298
使用调制解调器扩充向导配置调制解调器模块 .....	304
调制解调器指令和约束的概述 .....	308
用于调制解调器模块的指令 .....	309
用于调制解调器模块的采样程序 .....	313
支持智能模块的S7-200 CPU .....	313
用于调制解调器模块的特殊内存位置 .....	314
高级主题 .....	316
讯息传送电话号码格式 .....	318
文字讯息格式 .....	319
CPU数据传送讯息格式 .....	320

## 调制解调器模块的特性

使用调制解调器模块允许将S7-200直接连接到模拟电话线上，并提供以下特性：

- 提供国际电话线接口
- 提供与STEP 7-Micro/WIN的调制解调器接口，用于进行编程和故障诊断（远程服务）。
- 支持Modbus RTU协议
- 支持数字和文字寻呼
- 支持SMS讯息传送
- 允许CPU至CPU或CPU至Modbus的数据传送
- 提供口令保护
- 提供安全回叫
- 调制解调器模块配置存储在CPU中

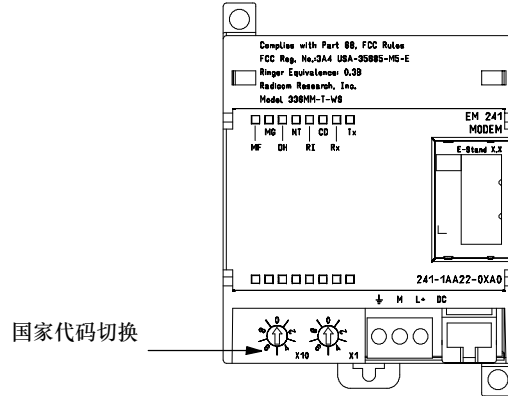
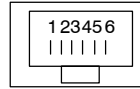


图10-1 EM 241调制解调器模块

可以使用STEP 7-Micro/WIN调制解调器扩充向导来配置调制解调器模块。请参见附录A以获取关于调制解调器模块的技术规范。

## 国际电话线接口

调制解调器模块是一个标准的V.34（33.6 kBaud）的10位调制解调器，并与大多数的内置和外置PC调制解调器相兼容。该调制解调器模块不能与11位调制解调器进行通讯。



- | 引脚 | 说明 |
|----|----|
| 3  | 环  |
| 4  | 触点 |
- 允许反向连接。

图10-2 RJ11插头的外形

通过安装在模块前端的六位四线RJ11连接器，可将调制解调器模块连接到电话线上。参见图10-2。

可能需要一个适配器来转换RJ11连接器，以便连接不同国家的标准电话线终端。更详细信息，请参见适配器连接器的有关文档。

调制解调器与电话线接口由外部24 VDC电源供电。可将其连接到CPU传感器电源或外部电源。将调制解调器模块上的接地端连接到系统的接地端。

在向调制解调器模块供电时，模块将自动配置用于特定国家操作的电话接口。模块前面的两个旋转开关可选择国家代码。在模块上电之前，必须将开关设置为所期望的国家选择。请参见表10-1以获取有关所支持国家的开关设置。

表10-1 EM 241所支持的国家

开关设置	国家
00	澳大利亚
01	奥地利
02	比利时
05	加拿大
08	丹麦
09	芬兰
10	法国
11	德国
12	希腊
16	爱尔兰
18	意大利
22	卢森堡
25	荷兰
26	新西兰
27	挪威
30	葡萄牙
34	西班牙
35	瑞典
36	瑞士
38	英国
39	美国

## STEP 7-Micro/WIN接口

调制解调器模块允许通过电话线与STEP 7-Micro/WIN进行通讯（远程服务）。在使用STEP 7-Micro/WIN时，无需对S7-200 CPU进行配置或编程即可将调制解调器模块用作远程调制解调器。

按照下列步骤使用具有STEP 7-Micro/WIN的调制解调器模块：

1. 取下S7-200 CPU的电源，然后将调制解调器模块连接到I/O扩充总线。在S7-200 CPU通电时，切勿连接任何I/O模块。
2. 将电话线连接到调制解调器模块。必要时使用适配器。
3. 将24V直流电源连接到调制解调器模块接线盒。
4. 将调制解调器模块接线盒的地线连接到系统地线。
5. 设置国家代码开关。
6. 接通S7-200 CPU和调制解调器模块的电源。
7. 配置STEP 7-Micro/WIN，以与10位调制解调器进行通讯。

## Modbus RTU协议

可以配置调制解调器模块作为Modbus RTU从属装置响应。调制解调器模块将通过调制解调器接口接收Modbus请求，对这些请求进行解释，并与CPU相互传送数据。然后，调制解调器模块将生成一个Modbus响应，并通过调制解调接口将其传出去。



### 提示

如果将调制解调器模块配置成作为Modbus RTU 从属装置响应，则STEP 7-Micro/WIN将不能通过电话线与调制解调器模块进行通讯。

调制解调器模块所支持的Modbus功能如表10-2所示。

Modbus功能4和功能16允许在一个请求中读取或写入多达125个的保持寄存器（250个字节的V内存）。功能5和功能15写入CPU的输出图象寄存器。这些值可被用户程序所覆盖。

通常，将Modbus地址写为包括数据类型和偏移量的5或6个字符值。第1或2个字符确定数据类型，后4个字符在数据类型范围内选择适当的值。Modbus主设备将地址映射到正确的Modbus功能中。

表10-2 调制解调器模块所支持的Modbus功能

功能	说明
功能01	读线圈（输出）状态
功能02	读输入状态
功能03	读保持寄存器
功能04	读输入（模拟输入）寄存器
功能05	写单个线圈（输出）
功能06	预置单个寄存器
功能15	写多个线圈（输出）
功能16	预置多个寄存器

表10-3显示了调制解调器模块所支持的Modbus地址，以及Modbus地址与S7-200 CPU地址的映射关系。

使用调制解调扩充向导可为调制解调器模块创建一个支持Modbus RTU协议的配置块。在使用Modbus协议之前，必须将调制解调器模块的配置块下载到CPU数据块。

表10-3 将Modbus地址映射到S7-200 CPU

Modbus地址	S7-200 CPU地址
000001	Q0.0
000002	Q0.1
000003	Q0.2
...	...
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
...	...
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
...	...
030032	AIW62
040001	VW0
040002	VW2
040003	VW4
...	...
04xxxx	VW 2* (xxxx-1)

## 寻呼与SMS讯息传送

调制解调器模块支持向移动电话（移动电话服务供应商支持的范围内）发送数字和文字寻呼讯息以及SMS（短讯息服务）讯息。这些讯息和电话号码均存储在调制解调器模块配置块中，必须下载到S7-200 CPU的数据块中。可使用调制解调器扩充向导来创建调制解调器配置块的讯息和电话号码。调制解调器扩充向导还将创建程序代码，以允许程序启动讯息的发送。

## 数字寻呼

数字寻呼使用按键式电话机的按键音向寻呼机发送数字值。调制解调器模块拨打所请求的寻呼服务，并等待将要完成的语音讯息，然后发送与数字寻呼讯息相对应的按键音。在寻呼讯息中，允许使用数字0到9、星号（\*）、A、B、C、D等。寻呼机对星号和A、B、C、D等字符所显示的实际字符均是非标准字符，并且由寻呼机和寻呼服务供应商决定。

## 文字寻呼

文字寻呼允许将字母数字讯息传送给寻呼服务供应商，然后再传到寻呼机上。文字寻呼供应商通常使用调制解调器线来接收文字寻呼。调制解调器模块使用远程定位器字母数字协议（TAP）向服务供应商发送文字讯息。许多文字寻呼供应商使用该协议接受讯息。

## 短讯息服务（SMS）

短讯息服务（SMS）由某些与GSM兼容的移动电话服务所支持。SMS允许调制解调器模块通过模拟电话线向SMS供应商发送讯息。然后SMS供应商将讯息传送到移动电话上，讯息即以文本形式显示在电话上。调制解调器模块使用远程定位器字母数字协议（TAP）和通用计算机协议（UCP）向SMS供应商发送讯息。只能将SMS讯息发送给那些在调制解调器线上支持这些协议的SMS供应商。



## 在文字和SMS讯息中嵌入变量

调制解调器模块可在文字讯息中嵌入来自CPU的数据值，并基于讯息中的规范对该数据值进行格式化。可以指定小数点左边和右边的位数，以及小数点是逗号还是点号。当用户程序命令调制解调器模块发送一条文字讯息时，调制解调器模块将从CPU检索讯息，判断该讯息中需要什么样的CPU值，从CPU中获取这些数值，然后对这些数值进行格式化并将其放置在文字讯息内，最后将讯息传送给服务供应商。

讯息传送供应商的电话号码、讯息以及讯息中所嵌入的变量通过多个CPU扫描循环从CPU中读取。在发送讯息时，用户程序不能修改电话号码或讯息。讯息中所嵌入的变量则可在讯息发送期间继续刷新。如果讯息包含多个变量，则那些变量将通过多个CPU扫描循环进行读取。如果希望讯息中的所有嵌入变量保持一致，则在发送讯息后不要修改任何嵌入的变量。

## 数据传送

调制解调器模块允许用户程序通过电话线向其它CPU或Modbus设备传送数据。数据传送和电话号码可通过调制解调器扩充向导进行配置，并存储在调制解调器模块的配置块中。该配置块随后将下载到S7-200 CPU的数据块中。调制解调器扩充向导还创建程序代码，以允许程序开始进行数据传送。

数据传送既可以是向远程设备读取数据的请求，也可以是向远程设备写入数据的请求。数据传送可以在1至100数据字之间进行读或写操作。数据传送可与所连CPU的V内存之间相互移动数据。

调制解调器扩充向导允许创建一个数据传送，它包括从远程设备的单次读取、向远程设备的单次写入、或者对远程设备的同时读写。

数据传送使用所配置的调制解调器模块的协议。如果调制解调器模块配置为支持PPI协议（它对STEP 7-Micro/WIN作出响应），则调制解调器模块将使用PPI协议传送数据。如果调制解调器模块配置为支持Modbus RTU协议，则将使用Modbus协议进行数据传送。

远程设备的电话号码、数据传送请求以及正在传送的数据均通过多个CPU扫描循环从CPU读取。在发送讯息时，用户程序不能修改电话号码或讯息。而且，在传送讯息时，也不能够修改正在传送的数据。

如果远程设备是另一个调制解调器模块，则通过在电话号码配置中输入远程调制解调器模块的密码，数据传送可使用密码功能。回叫功能不能用于数据传送。

## 口令保护

密码安全性是调制解调器模块的可选功能，并可使用调制解调器扩充向导来启用。调制解调器模块所使用的密码不同于CPU密码。调制解调器模块的密码是独立的8位字符密码，呼叫人必须在允许访问所连CPU前将其提供给调制解调器模块。密码将作为调制解调器模块配置块的一部分，存储在CPU的V内存内。调制解调器模块的配置块必须下载到所连CPU的数据块中。

如果在系统块中启用CPU的密码安全性，则呼叫人必须提供CPU密码，以便获得对任何密码保护功能的访问。

## 安全回叫

安全回叫功能是调制解调器模块的可选功能。使用调制解调器扩充向导可对其进行配置。回叫功能只允许从预先设定的电话号码访问CPU，从而为所连CPU提供了额外的安全保护。当启用回叫功能时，调制解调器模块应答拨入的所有呼叫，并对呼叫人进行验证，然后断开线路。如果呼叫人是经授权的，则调制解调器模块为该呼叫人拨打预先设定的电话号码，并允许其访问CPU。

调制解调器模块支持三种回叫模式：

- 回叫预先设定的单个电话号码
- 回叫预先设定的多个电话号码
- 回叫任意电话号码

通过在调制解调器扩充向导中勾选适当的选项，然后定义回叫的电话号码，可选择回叫模式。回叫电话号码存储在所连CPU的数据块的调制解调器模块配置块中。

最简单的回叫形式是呼叫一个已预先设定的电话号码。如果在调制解调器模块的配置块中只存储了一个回叫号码，则无论调制解调器模块何时应答正在进入的呼叫，它都将通知呼叫人回叫功能已启用，然后断开呼叫人，并拨打在配置块中指定的回叫号码。

调制解调器模块还支持对多个预设电话号码的回叫。在该模式下，将向呼叫人查询电话号码。如果提供的号码与调制解调器模块的配置块中预先设定的其中一个号码相匹配，则调制解调器模块将断开呼叫人，然后使用与配置块相匹配的号码进行回叫。用户可配置多达250个的回叫号码。

当存在多个预先设定的回叫号码时，连接调制解调器模块时所提供的回叫号码除了前两个数字外，必须与调制解调器模块配置块中的号码完全匹配。例如，如果由于可能需要拨打外线（9）和长途（1）而将回叫号码配置为91（123）4569999，则所提供的回叫号码可以是以下号码中的任何一个：

- 91（123）4569999
- 1（123）4569999
- （123）4569999

以上所有电话号码都认为是匹配的回叫号码。在进行回叫时，调制解调器模块将使用其配置块中的回叫电话号码，在本实例中为91（123）4569999。在配置多个回叫号码时，请确保所有电话号码除了前两位数字以外，都是唯一的。进行回叫号码比较时，在电话号码中只使用数字字符。类似逗号或括号这样的字符在进行回叫号码比较时将被忽略。

通过在回叫配置期间选择“启用任意电话号码的回叫”选项，可在调制解调器扩充向导中设置对任意电话号码的回叫。如果选择这一选项，则调制解调器模块将应答正在进入的呼叫，并请求一个回叫电话号码。在呼叫人提供电话号码之后，调制解调器模块将断开连接，并拨打该电话号码。这种回叫模式只提供一种向调制解调器模块的电话连接收取电话费用的方式，并不为S7-200 CPU提供任何安全保护。若使用该回叫模式，应使用调制解调器模块的密码功能来提供安全保护。

调制解调器模块的密码和回叫功能可同时启用。调制解调器模块在处理回叫前将要求呼叫人提供正确的密码。

## 调制解调器模块的配置表

所有的文字讯息、电话号码、数据传送讯息、回叫号码以及其它选项，均存储在调制解调器的配置表中，必须将其下载到S7-200 CPU的V内存中。调制解调器扩充向导将引导完成调制解调器模块配置表的创建。STEP 7-Micro/WIN随后将调制解调器模块配置表放置到要下载至S7-200 CPU的数据块中。

在启动时以及CPU的任何一次从STOP（停止）到RUN（运行）转换的5秒钟之内，调制解调器模块将从CPU读取该配置表。只要调制解调器模块与STEP 7-Micro/WIN联机，调制解调器模块就不会从CPU读取新的配置表。如果在调制解调器模块联机时下载新的配置表，则调制解调器模块将在联机对话结束时读取新的配置表。

如果调制解调器模块检测到配置表中的错误，则模块前部的“模块正常（MG）”LED将会不断闪烁。查看STEP 7-Micro/WIN中的“PLC信息”画面，或读取SMW220（用于模块的0号槽）中关于配置错误信息的数值。调制解调器模块的配置错误如表10-4所列。如果使用调制解调器扩充向导来创建调制解调器模块的配置表，则STEP 7-Micro/WIN在创建配置表前将对数据进行检查。

表10-4 EM 241 配置错误（十六进制）

错误	说明
0000	无错误
0001	无24 VDC外接电源
0002	调制解调器故障
0003	无配置块ID - 位于配置表起始处的EM 241标识对于该模块无效。
0004	配置块超出范围 - 配置表指针未指向V内存，或者表的某些部分超出了所连CPU的V内存范围。
0005	配置错误 - 回叫功能被启用，且回叫电话号码数等于0或大于250。讯息数大于250。讯息传送的电话号码数大于250，或者讯息传送的电话号码的长度大于120个字节。
0006	国家代码选择错误 - 两个旋转开关上的国家选择不是所支持的值。
0007	电话号码太长 - 回叫功能被启用，且回叫号码长度大于上限值
0008至00FF	保留
01xx	回叫号码xx错误 - 在回叫电话号码xx中存在非法字符。数值xx为1代表第一个回叫号码，为2代表第二个回叫号码，依此类推。
02xx	电话号码xx错误 - 在讯息电话号码xx或数据传送电话号码xx的某个区域中包含一个非法数值。数值xx为1代表第一个电话号码，为2代表第二个电话号码，依此类推。
03xx	讯息xx错误 - 讯息或数据传输数xx超出最大长度。数值xx为1代表第一条讯息，为2代表第二条讯息，依此类推。
0400至FFFF	保留

## 调制解调器模块的状态LED

前面板上有8个调制解调器模块的状态LED。表10-5描述了状态LED。

表10-5 EM 241状态LED

LED	说明
MF	模块故障 - 当模块检测到下列故障条件时该LED灯亮: <ul style="list-style-type: none"> <li>■ 无24 VDC外接电源</li> <li>■ I/O监视程序超时</li> <li>■ 调制解调器故障</li> <li>■ 与本地CPU的通讯出错</li> </ul>
MG	模块正常 - 当不存在模块故障条件时, 该LED点亮。如果配置表中存在错误或者用户为电话线接口选择了非法的国家设置, 则“模块正常”LED将闪烁。查看STEP 7-Micro/WIN中的“PLC信息”画面, 或读取SMW220(用于模块的0号槽)中关于配置错误信息的数值。
OH	摘机 - 当EM 241正在使用电话线时, 该LED点亮。
NT	无拨号音 - 当EM 241接到发送讯息命令而电话线上无拨号音时, 该LED点亮, 表示存在错误条件。只有当EM 241被配置为拨号前检查拨号音时, 才会出现这个错误条件。当尝试拨号失败后, 该LED将仍然点亮约5秒左右。
RI	振铃指示器 - 该LED指示EM 241正在接收一个拨入电话。
CD	载体检测 - 该LED指示与远程调制解调器的连接已经建立。
Rx	接收数据 - 调制解调器在接收数据时, 该LED闪烁。
Tx	发送数据 - 调制解调器在传送数据时, 该LED闪烁。

## 使用调制解调器扩充向导配置EM 241调制解调器模块



调制解调器  
扩充

10

从STEP 7-Micro/WIN工具菜单或从浏览条的工具部分启动调制解调器扩充向导。

要使用该向导, 必须对项目进行编译, 并将其设置为“符号编址模式”。如果尚未编译程序, 请现在编译。

1. 在调制解调器扩充向导的第一个画面中, 选择“配置EM 241调制解调器模块”, 然后单击“下一步>”。
2. 调制解调器扩充向导要求调制解调器模块的位置与S7-200 CPU相关, 以便生成正确的程序代码。单击“读取模块”按钮, 即可自动读取与CPU相连的智能模块的位置。扩展模块从零开始顺序编号。双击想要配置的调制解调器模块, 或将“调制解调器模块位置”域设置为调制解调器模块的位置。单击“下一步>”。

对于固化程序版本1.2之前的S7-200 CPU, 必须将智能模块安装在CPU旁, 以使调制解调器扩充向导能够配置该模块。

3. 密码保护画面允许启用调制解调器模块的密码保护功能, 并为该模块设置一个1至8个字符的密码。该密码与S7-200 CPU的密码无关。当模块受密码保护时, 如果任何人试图通过调制解调器模块连接S7-200 CPU, 则需要提供正确的密码。如果需要, 选择密码功能并输入密码。单击“下一步>”。
4. 调制解调器模块支持两种通讯协议: PPI协议(与STEP 7-Micro/WIN通讯)和Modbus RTU协议。协议选择取决于远程通讯伙伴正在使用的设备型号。当调制解调器模块应答呼叫并进行CPU数据传送时, 该设置将控制所使用的通讯协议。选择适当的协议, 然后单击“下一步>”。

5. 可配置该模块，以便向寻呼机发送数字和文字讯息，或向手机发送短讯息服务（SMS）讯息。选中“启用讯息传送”复选框，然后单击“配置讯息传送…”按钮，以便定义讯息及接收方的电话号码。
6. 当设置将要向寻呼机或手机发送的讯息时，必须定义讯息和电话号码。选择“配置讯息传送”画面中的“讯息”标签，并单击“新讯息”按钮。输入讯息的文字内容，并指定要插入到讯息中的CPU数据值。为在讯息中插入CPU数据值，将光标移动到数据位置，然后单击“插入数据…”按钮。指定CPU数据值的地址（例如VW100）、显示格式（例如有符号整数）以及小数点左右的位数。还可以指定小数点是点号还是逗号。
  - 数字寻呼讯息仅限于数字0-9，字母A、B、C、D以及星号（\*）。数字寻呼讯息所允许的最大长度随服务供应商而不同。
  - 文字讯息最长可有119个字符，并且可以是任意的字母数字字符。
  - 文字讯息可包含任意数目的嵌入变量。
  - 嵌入变量可以来自所连CPU的V、M、SM、I、Q、S、T、C或AI内存。
  - 显示十六进制数时将以“16#”开头。数值中的字符数取决于变量的大小。例如，VW100将显示为16# 0123。
  - 小数点左边的位数必须足够大，以便显示所期望的数值范围，如果数据值是带符号的整数或浮点数，则还应包括正负号。
  - 如果数据格式是整数，而且小数点右边的位数为零，则该整数将显示为标定的整数。例如，如果VW100 = 1234，且小数点右边有两位，则该数据将显示为“12.34”。
  - 如果数据值太大，以致无法在指定的域中显示，则调制解调器模块将以#来替代数据值的所有字符位置。
7. 通过选择“配置讯息传送”画面上的“电话号码”标签来配置电话号码。单击“新电话号码…”按钮，以便添加一个新的电话号码。一旦配置了电话号码，则必须将它添加到项目中。高亮显示“可供使用的电话号码”栏中的电话号码，并单击向右方向箭头，即可将这些电话号码添加到当前项目中。一旦将电话号码添加到当前项目中，就可以选择该电话号码并为其添加符号名，以在程序中使用。

根据用户所选择的讯息传送类型，电话号码由多个域组成。

- “讯息传送协议”选项告诉调制解调器模块在向讯息服务供应商发送讯息时要使用何种协议。数字寻呼机只支持数字协议。文字寻呼服务通常需要TAP（远程定位器字母数字协议）。SMS讯息传送供应商支持TAP或UCP（通用计算机协议）。通常，对于SMS讯息传送有三种不同的UCP服务。大多数供应商支持命令1或51。请向SMS供应商查询，以确定他们所要求的协议和命令。
- 在“说明”域中可以为电话号码添加文字说明。

- 在“电话号码”域中可输入讯息传送服务供应商的电话号码。对于文字讯息，这是服务供应商用于接收文字讯息的调制解调器线路的电话号码。对于数字寻呼，这是寻呼机本身的电话号码。调制解调器模块允许电话号码域中的字符最多为40个。在调制解调器模块向外拨号时允许下列字符出现在电话号码中：

0至9	电话机键盘上所允许
A、B、C、D、*、#	DTMF数字（仅适用于音频拨号）
,	暂停拨号2秒
!	命令调制解调器产生挂机闪烁
@	等待5秒没有声音
W	继续之前等待拨号音
( )	忽略（可用于格式化电话号码）

- 在“特殊寻呼机ID”或“手机号码”域中，输入接收讯息的寻呼机号码或手机号码。该号码除了数字0到9之外，不应包含任何字符。最多允许输入20个字符。
  - “密码”域是TAP讯息的可选项。有些供应商要求密码，但通常可以让该域空着。调制解调器模块允许该密码最多有15个字符。
  - “始发电话号码”域允许在SMS讯息中对调制解调器模块进行识别。有些使用UCP命令的服务供应商需要该域。有些服务供应商对于该域可能还有最低字符数的要求。调制解调器模块允许最多15个字符。
  - “调制解调器标准”域在调制解调器模块与服务供应商的调制解调器之间无法进行调制解调器标准协调时使用。缺省设置是V.34（33.6 kBaud）。
  - “数据格式”域允许在向服务供应商传送讯息时，对调制解调器使用的数据位和奇偶校验进行调整。TAP通常使用7位数据位和偶校验，但是有些服务供应商则使用8位数据位和无奇偶校验。UCP始终使用8位数据位，不带奇偶校验。请向服务供应商查询，以决定使用何种设置。
8. 可配置调制解调器模块以便向另一个S7-200 CPU传送数据（如果选择了PPI协议），或向Modbus设备传送数据（如果选择了Modbus协议）。选中“启用CPU数据传送”复选框，并单击“配置CPU至...”按钮，以便定义数据传送和远程设备的电话号码。
  9. 当设置CPU至CPU或CPU至Modbus的数据传送时，必须定义所要传送的数据以及远程设备的电话号码。选择“配置数据传送”画面中的“数据传送”标签，并单击“新传送”按钮。数据传送包括远程设备的数据读取、远程设备的数据写入，或者同时对远程设备的读写操作。如果同时选择进行读写，则先读后写。

在每次读或写操作中，最多可传送100个字。必须在本机CPU的V内存中进行数据传送。向导在对远程设备的内存位置进行描述时，始终将远程设备当作S7-200 CPU。如果远程设备是Modbus设备，则将在Modbus设备的保持寄存器中进行传送（地址04xxxx）。等效的Modbus地址（xxxx）确定如下：

$$\begin{aligned} \text{Modbus地址} &= 1 + (\text{V内存地址} / 2) \\ \text{V内存地址} &= (\text{Modbus地址} - 1) * 2 \end{aligned}$$

10. 在“配置CPU数据传送”画面的“电话号码”标签中，可以定义用于CPU到CPU或CPU到Modbus数据传送的电话号码。单击“新电话号码…”按钮以添加一个新的电话号码。一旦已经配置了一个电话号码，就必须将它添加到项目中。高亮显示“可供使用的电话号码”栏中的电话号码，并单击右方向箭头，即可将电话号码添加到当前项目中。一旦将电话号码添加到当前项目中，就可以选择这些电话号码，并为其添加符号名，以在程序中使用。

“说明”域和“电话号码”域与前面对讯息传送进行的说明相同。如果远程设备是一个调制解调器模块，并且启用了密码保护功能，则需要“密码”域。本地调制解调器模块中的“密码”域必须设置为远程调制解调器模块的密码。当远程调制解调器模块需要密码时由本地调制解调器模块提供。

11. 回叫功能使得调制解调器模块能够在接收到一个来自远程STEP7-Micro/WIN的呼叫后，自动断开连接，并拨打预先设定的电话号码。选择“启用回叫”复选框，然后单击“配置回叫…”按钮以配置回叫的电话号码。单击下一步>。
12. “配置回叫…”画面允许输入当调制解调器模块应答拨入的呼叫时所使用的电话号码。如果回叫号码是预先设定的，则选择“启用对指定电话号码的回叫”。如果调制解调器模块要接受呼叫人所提供的任何回叫号码（以进行反向连接计费），则选中“启用对任意电话号码的回叫”选项。

如果只允许对特定的电话号码进行回叫，则可单击“新电话号码”按钮以添加回叫电话号码。“回叫属性”画面允许输入预先设定的回叫电话号码以及回叫号码的说明。此处输入的回叫号码是调制解调器模块在进行回叫时所使用的电话号码。该电话号码应包括连接到外线、等待外线时的停顿、连接长途等所需要的所有数字。

在输入新的回叫电话号码后，必须将其添加到项目中。高亮显示“可供使用的回叫电话号码”栏中的电话号码，并单击右方向箭头，即可将电话号码添加到当前项目中。

13. 可以在发送讯息时或数据传送期间，对调制解调器模块的试拨次数进行设置。只有在拨号或发送讯息的所有尝试都失败时，调制解调器模块才会向用户程序报告错误。

有些电话线在话筒摘机时没有拨号音。通常，如果在命令调制解调器模块发送讯息或执行回叫时没有拨号音，则调制解调器模块将会向用户程序返回一个错误。要想在无拨号音的线路上拨出电话，可选中复选框“启用无拨号音的拨号”选项。

14. 调制解调器扩充向导将为调制解调器模块创建一个配置块，并要求用户输入起始内存地址，用来存储调制解调器模块的配置数据。调制解调器模块的配置块存储在CPU的V内存中。STEP7-Micro/WIN将该配置块写入项目数据块中。配置块的大小将根据所配置的讯息数和电话号码而不同。可以选择想要存储配置块的V内存地址，或者在希望向导建议一个大小合适的未使用的V内存块地址时，单击“建议地址”按钮。单击“下一步>”。
15. 配置调制解调器模块的最后一步就是为调制解调器模块指定命令字节的Q内存地址。通过计算安装在S7-200上调制解调器模块之前具有离散输出的任意模块所占用的输出字节，可确定Q内存地址。单击“下一步>”。
16. 现在，调制解调器扩充向导即可为所选配置生成项目组件（程序块和数据块），并且使这些程序代码可为程序所用。最后的向导画面将显示所要求的配置项目组件。必须将调制解调器模块配置块（数据块）和程序块下载到S7-200 CPU。

## 调制解调器指令和约束的概述

通过在模块位置和用户所选配置选项的基础上创建唯一的指令子例行程序，调制解调器扩充向导将使调制解调器模块的控制变得更容易。每条指令都有一个前缀“MODx\_”，这里，x为模块位置。

### 使用EM 241调制解调器模块指令的要求

在使用调制解调器模块指令时，请考虑这些要求：

- 调制解调器模块指令使用三个子例行程序。
- 调制解调器模块指令将程序所需的存储空间增加到370字节。如果删除一个无用的指令子例行程序，则可在需要时返回调制解调器扩充向导以重新创建该指令。
- 必须确保在同一时间只有一条指令激活。
- 这些指令不能用在中断例行程序中。
- 调制解调器模块在其首次通电以及从STOP（停止）转换到RUN（运行）模式时，将读取配置表信息。除非有模式切换或再次通电，否则模块将无法察觉程序中对配置表所作的任何改变。

### 使用EM 241调制解调器模块指令

为在S7-200程序中使用调制解调器模块指令，请遵循这些步骤：

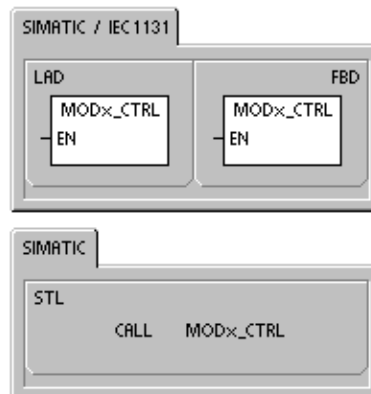
1. 使用调制解调器扩充向导来创建调制解调器模块配置表。
2. 在程序中插入MODx\_CTRL指令，并使用SM0.0接点在每一循环中对其执行。
3. 为每一条需要发送的讯息插入MODx\_MSG指令。
4. 为每次数据传送插入MODx\_XFR指令。



## 用于调制解调器模块的指令

### MODx\_CTRL指令

MODx\_CTRL (控制) 指令用于调制解调器模块的启用和启动。该指令应在每个扫描周期内调用，而且在项目中只能使用一次。



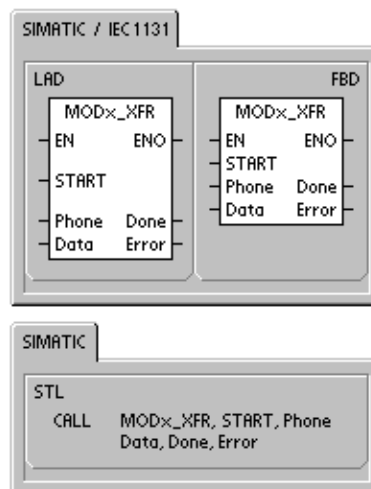
### MODx\_XFR指令

MODx\_XFR (数据传送) 指令用于命令调制解调器模块将数据读写到另一个S7-200 CPU或Modbus设备中。从触发START输入到设置Done位，该指令需要20到30秒的时间。

EN位必须接通以向模板发出命令，并应保持接通一直到Done位设置完毕，这标志了整个过程的完成。当接通START输入且模块当前不忙时，每次扫描周期都将向调制解调器模块发送XFR命令。START输入可以用边沿检测元素进行脉冲触发，这样命令将只发送一次。

Phone是其中一个数据传送电话号码。如果已使用调制解调器扩充向导对数据传送电话号码进行定义，则可以使用为每个号码分配的符号名。

Data是其中一个所定义的数据传送号码。当使用调制解调器扩充向导对请求进行定义时，可以使用为数据传送分配的符号名。



Done是一个在调制解调器模块完成数据传送时接通的位。

Error是一个包含了数据传送结果的字节。表10-4定义了在执行该指令时可能引起的错误条件。

表10-6 MODx\_XFR指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、M、S、SM、T、C、V、L、流量
Phone, Data	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、常量、*VD、*AC、*LD
Done	BOOL	I、Q、M、S、SM、T、C、V、L
Error	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD

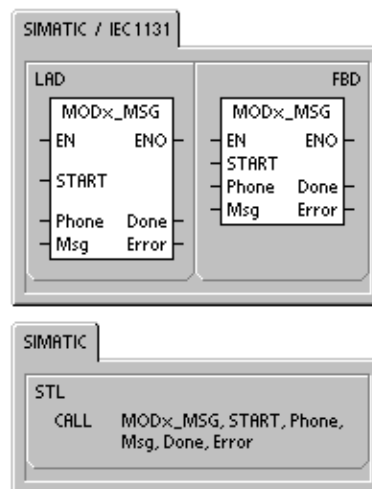
## MODx\_MSG指令

MODx\_MSG（发送讯息）指令用于从调制解调器模块发送寻呼或SMS短讯息。从触发START输入到设置Done位，该指令大约需要20到30秒的时间。

EN位必须接通以便向模块发出命令，并应保持接通一直到Done位设置完毕，这标志了整个过程的完成。当接通START输入而模块当前又不忙时，每次扫描周期都将向调制解调器模块发送一条MSG命令。START输入可以用边沿检测元素进行脉冲触发，这样命令将只发送一次。

Phone是其中一个讯息电话号码。如果已使用调制解调器扩充向导对讯息电话号码进行定义，则可以使用为每个号码分配的符号名。

Msg是所定义的其中一条讯息的号码。当使用调制解调器扩充向导对请求进行定义时，可以使用为讯息分配的符号名。



Done是一个在调制解调器模块完成向服务供应商的讯息发送后将接通的位。

Error是一个包含了模块请求结果的字节。表10-8定义了执行该请求后可能引起的错误条件。

表10-7 MODx\_MSG指令的参数

输入 / 输出	数据类型	操作数
START	BOOL	I、Q、M、S、SM、T、C、V、L、流量
Phone, Msg	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、常量、*VD、*AC、*LD
Done	BOOL	I、Q、M、S、SM、T、C、V、L
Error	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD

表10-8 MODx\_MSG和MODx\_XFR指令返回的错误值

错误	说明
0	无错误
<b>电话线出错</b>	
1	无拨号音出现
2	线路忙
3	拨号出错
4	无应答
5	连接超时（1分钟之内无任何连接）
6	连接中止或出现未知响应
<b>命令出错</b>	
7	数字寻呼讯息包含非法数字
8	电话号码（Phone输入）超出范围
9	讯息或数据传送（Msg或Data输入）超出范围
10	文字讯息或数据传送讯息出错
11	讯息传送或数据传送电话号码出错
12	不允许的操作数（例如，试图设置为零）
<b>服务供应商错误</b>	
13	讯息传送服务无响应（超时）
14	讯息服务因未知原因而断开
15	用户中止的讯息（禁用命令位）
<b>TAP - 服务供应商返回的文字寻呼和SMS讯息错误</b>	
16	接收到远程断开（服务供应商中止的对话）
17	讯息服务不接受登录（密码不正确）
18	讯息服务不接受块（检验和或传输出错）
19	讯息服务不接受块（未知原因）
<b>UCP - 服务供应商返回的SMS讯息错误</b>	
20	未知错误
21	检验和出错
22	句法出错
23	系统不支持的操作数（非法命令）
24	此时不允许的操作数
25	呼叫限制已激活（黑名单）
26	呼叫人地址无效
27	验证故障
28	合法代码故障
29	GA无效
30	不允许重复
31	重复的合法代码出现故障
32	不允许优先呼叫
33	优先呼叫的合法代码出现故障
34	不允许紧急讯息
35	紧急讯息的合法代码出现故障
36	不允许逆向收费
37	逆向收费的合法代码出现故障

表10-8 MODx\_MSG和MODx\_XFR指令返回的错误值（续）

错误	说明
UCP - 服务供应商返回的SMS讯息（继续）	
38	不允许延期传输
39	新的AC无效
40	不允许新的合法代码
41	标准文本无效
42	时间周期无效
43	系统不支持的讯息类型
44	讯息太长
45	所请求的标准文本无效
46	讯息类型对寻呼机类型不适用
47	讯息在SMSC中未找到
48	保留
49	保留
50	用户挂机
51	不支持传真组
52	不支持的传真讯息类型
数据传送错误	
53	讯息超时（远程设备无响应）
54	远程CPU正忙于上传或下载
55	访问出错（超出内存范围，非法数据类型）
56	通讯出错（未知响应）
57	响应中的检验和或CRC出错
58	用于回叫的远程EM 241设置（不允许）
59	远程EM 241拒绝所提供的密码
60至127	保留
指令使用出错	
128	不能处理该请求。调制解调器模块正忙于处理另一个请求，或者对该请求不存在任何START脉冲。
129	调制解调器模块出错： <ul style="list-style-type: none"> <li>■ 使用调制解调器扩充向导配置的调制解调器模块的位置或Q内存地址与实际的位置或内存地址不匹配。</li> <li>■ 参见SMB8至SMB21（I/O模块ID和出错寄存器）</li> </ul>

## 用于调制解调器模块的采样程序

**实例: 调制解调器模块**

```

Network 1 //在每次扫描周期调用MOD0_CTRL
           // 子例行程序。
LD      SM0.0
CALL   MOD0_CTRL

Network 2 //发送文字讯息给手机。
LD      I0.0
EU
=       L63.7
LD      I0.0
CALL   MOD0_MSG, L63.7, CellPhone, Message1,
      M0.0, VB10

Network 3 //将数据传送给远程CPU。
LD      I0.1
EU
=       L63.7
LD      I0.1
CALL   MOD0_XFR, L63.7, RemoteCPU, Transfer1,
      M0.0,
      VB10
    
```

## 支持智能模块的S7-200 CPU

调制解调器模块是专为与S7-200 CPU（如表10-9所示）一起工作而设计的智能扩展模块。

表10-9 EM 241调制解调器模块与S7-200 CPU的兼容性

CPU	说明
CPU 222版本1.10或以上版本	CPU 222 DC/DC/DC
	CPU 222 AC/DC/Relay
CPU 224版本1.10或以上版本	CPU 224 DC/DC/DC
	CPU 224 AC/DC/Relay
CPU 226版本1.00或以上版本	CPU 226 DC/DC/DC
	CPU 226 AC/DC/Relay
CPU 226XM版本1.00或以上版本	CPU 226XM DC/DC/DC
	CPU 226XM AC/DC/Relay

## 用于调制解调器模块的特殊内存位置

根据每个智能模块在I/O扩展总线上的物理位置，均为其分配有50个字节的特殊内存区（SM）。当检测到错误条件或状态改变时，模块将通过更新与该模块位置相对应的SM位置对此进行说明。如果它是第一个模块，则它将根据需要更新SMB200到SMB249，以报告状态和出错信息。如果它是第二个模块，则将更新SMB250至SMB299，依此类推。参见表10-10。

### 注意

智能模块SM位置的分配方式对于V2.2及以后的版本均有所不同。

如果使用的是V2.2以前的CPU，则应将所有的智能模块安装在邻近CPU和所有非智能模块之前的插槽中，以确保兼容性。

表10-10 特殊存储器字节SMB200至SMB549

特殊存储器字节SMB200至SMB549						
0号槽中的智能模块	1号槽中的智能模块	2号槽中的智能模块	3号槽中的智能模块	4号槽中的智能模块	5号槽中的智能模块	6号槽中的智能模块
SMB200至SMB249	SMB250至SMB299	SMB300至SMB349	SMB350至SMB399	SMB400至SMB449	SMB450至SMB499	SMB500至SMB549

表10-11表示为调制解调器模块所分配的特殊内存数据。该区域是按照智能模块位于I/O系统的0号槽进行定义的。

表10-11 EM 241调制解调器模块的SM位置

SM地址	说明																																						
SMB200至SMB215	模块名称（16个ASCII字符）SMB200为第一个字符。 “EM241调制解调器”																																						
SMB216至SMB219	S/W修订号（4个ASCII字符）SMB216是第一个字符。																																						
SMW220	出错代码 0000 - 无错误 0001 - 无用户电源 0002 - 调制解调器故障 0003 - 无配置块ID 0004 - 配置块超出范围 0005 - 配置出错 0006 - 国家代码选择出错 0007 - 电话号码太长 0008 - 讯息太长 000900FF - 保留 01xx - 回叫号码xx出错 02xx - 寻呼号码xx出错 03xx - 讯息号xx出错 0400FFFF - 保留																																						
SMB222	模块状态 - 反映LED状态  <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">MSB</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">LSB</td> <td style="text-align: center;">0</td> </tr> <tr> <td></td> <td style="text-align: center;">F</td> <td style="text-align: center;">G</td> <td style="text-align: center;">H</td> <td style="text-align: center;">T</td> <td style="text-align: center;">R</td> <td style="text-align: center;">C</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td></td> </tr> </table> <table style="width: 100%; border: none;"> <tr> <td style="width: 33%;">F - EM_FAULT</td> <td style="width: 33%;">0 - 无错误</td> <td style="width: 33%;">1 - 有错误</td> </tr> <tr> <td>G - EM_GOOD</td> <td>0 - 信号不好</td> <td>1 - 信号良好</td> </tr> <tr> <td>H - OFF_HOOK</td> <td>0 - 挂机</td> <td>1 - 摘机</td> </tr> <tr> <td>T - NO_DIALTONE</td> <td>0 - 有拨号音</td> <td>1 - 无拨号音</td> </tr> <tr> <td>R - RING</td> <td>0 - 无振铃</td> <td>1 - 有电话振铃</td> </tr> <tr> <td>C - CONNECT</td> <td>0 - 未连接</td> <td>1 - 已连接</td> </tr> </table>	MSB	7	6	5	4	3	2	1	LSB	0		F	G	H	T	R	C	0	0		F - EM_FAULT	0 - 无错误	1 - 有错误	G - EM_GOOD	0 - 信号不好	1 - 信号良好	H - OFF_HOOK	0 - 挂机	1 - 摘机	T - NO_DIALTONE	0 - 有拨号音	1 - 无拨号音	R - RING	0 - 无振铃	1 - 有电话振铃	C - CONNECT	0 - 未连接	1 - 已连接
MSB	7	6	5	4	3	2	1	LSB	0																														
	F	G	H	T	R	C	0	0																															
F - EM_FAULT	0 - 无错误	1 - 有错误																																					
G - EM_GOOD	0 - 信号不好	1 - 信号良好																																					
H - OFF_HOOK	0 - 挂机	1 - 摘机																																					
T - NO_DIALTONE	0 - 有拨号音	1 - 无拨号音																																					
R - RING	0 - 无振铃	1 - 有电话振铃																																					
C - CONNECT	0 - 未连接	1 - 已连接																																					
SMB223	通过开关设置国家代码（十进制数值）																																						

表10-11 EM 241调制解调器模块的SM位置 (续)

SM地址	说明								
SMW224	建立连接时所使用的波特率（无符号十进制数值）。								
SMB226	<p>用户命令的结果</p> <div style="text-align: center;"> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">MSB 7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">LSB 0</td> </tr> <tr> <td style="text-align: center;">D</td> <td style="text-align: center;">0</td> <td colspan="2" style="text-align: center;">ERROR</td> </tr> </table> </div> <p>D - Done位;                      0 - 正在执行操作                      1 - 操作完成</p> <p>ERROR: 出错代码说明, 参见表10-8</p>	MSB 7	6	5	LSB 0	D	0	ERROR	
MSB 7	6	5	LSB 0						
D	0	ERROR							
SMB227	电话号码选择器 - 该字节指定在发送讯息时, 将使用哪一个讯息传送电话号码。有效的数值为1至250。								
SMB228	讯息选择器 - 该字节指定将发送哪一条讯息。有效的数值为1至250。								
SMB229至SMB244	保留								
SMB245	第一个Q字节的偏移量用作该模块的命令接口。该偏移量将由CPU提供, 这主要是为了方便用户, 而非模块一定需要。								
SMD246	在V内存中指向调制解调器模块配置表的指针。指向V内存以外其它区域的指针将不被接受, 模块会继续检查该区域, 等待非零的指针值。								





表10-12 调制解调器模块的配置表（续）

回叫电话号码块（供选用）	
字节偏移量	说明
24	回叫电话号码1 - 表示从EM 241 调制解调器模块进行授权回叫访问的第一个电话号码的一个字符串。每个回叫电话号码必须分配有与回叫电话号码长度域中所指定的相同的空间（配置块中偏移量为6）。
24+ 回叫号码	回叫电话号码2
:	:
:	回叫电话号码n
讯息传送电话号码块（供选用）	
字节偏移量	说明
M	讯息传送电话号码1 - 表示包含有协议和拨号选项的讯息传送电话号码的一个字符串。每个回叫电话号码必须分配有与回叫电话号码长度域中所指定的相同的空间（配置块中偏移量为7）。 讯息传送电话号码的格式如下
M + 讯息号长度	讯息传送电话号码2
:	:
:	讯息传送电话号码n
讯息块（供选用）	
字节偏移量	说明
n	第一个讯息的V内存偏移量（相对于VB0）（2个字节）
N+2	讯息1的长度
N+3	讯息2的长度
:	
:	讯息n的长度
P	讯息1 - 代表第一个讯息的字符串（最多120个字节）。该字符串包括文本和嵌入变量说明，还可指定CPU的数据传送。 参见下面所描述的文字讯息格式和CPU数据传送格式。
P + 讯息1的长度	讯息2
:	:
:	讯息n

当这些事件发生时，调制解调器模块将重新读取配置表：

- 在S7-200 CPU从STOP（停止）转为RUN（运行）的5秒钟内（除非调制解调器当前联机）
- 每5秒钟一次，直到找到有效配置（除非调制解调器当前联机）
- 每次调制解调器从联机跳转到脱机状态时

## 讯息传送电话号码格式

讯息传送电话号码是一个包含有调制解调器模块发送讯息所需信息的结构。讯息传送电话号码是一个ASCII字符串，它以长度字节开始，后跟ASCII字符。讯息传送电话号码的最大长度为120个字节（包括长度字节）。

讯息传送电话号码最多可包含由斜杠 (/) 分隔的6个域。紧接着的斜杠表示一个空 (null) 域。空域在调制解调器模块中被设置为缺省值。

格式: <电话号码>/<ID>/<密码/<协议>/<标准>/<格式>

“电话号码”域是发送讯息时调制解调器模块所拨出的电话号码。如果所发送的讯息是文本或SMS讯息，则这就是服务供应商所提供的电话号码。如果讯息是一个数字寻呼，则该域就是寻呼机电话号码。如果讯息是一个CPU的数据传送，则该号码就是远程设备的电话号码。该域的最大字符数是40。

ID是寻呼机号或手机号。该域只能由数字0至9组成。如果协议是CPU数据传送，则该域将用来提供远程设备的地址。该域最多可有20个字符。

如果服务供应商需要密码，则“密码”域用于向通过TAP发送的讯息提供密码。对于通过UCP所发送的讯息，该域用作起始地址或电话号码。如果讯息是向另一个调制解调器模块的CPU数据传送，则该域可用于提供远程调制解调器模块的密码。密码最大长度可为15个字符。

“协议”域由一个ASCII字符组成，它将指示调制解调器模块该如何格式化和传送讯息。它可以是下列值:

- 1 - 数字寻呼协议 (缺省)
- 2 - TAP
- 3 - UCP命令1
- 4 - UCP命令30
- 5 - UCP命令51
- 6 - CPU 数据传送

“标准”域将强制调制解调器模块使用指定的调制解调器标准。标准域是一个ASCII字符。它可以是下列值:

- 1 - Bell 103
- 2 - Bell 212
- 3 - V.21
- 4 - V.22
- 5 - V.22位
- 6 - V.23c
- 7 - V.32
- 8 - V.32位
- 9 - V.34 (缺省)

“格式”域是三个ASCII字符，它指定了发送讯息时所要使用的数据位和奇偶校验的数目。该域不适用于协议为数字寻呼时的情况。只允许下面的两种设置:

- 8N1 - 8个数据位、无奇偶校验、一个停止位 (缺省)
- 7E1 - 7个数据位、偶校验、一个停止位

## 文字讯息格式

文字讯息格式定义了文字寻呼或SMS讯息的格式。这些讯息类型可以包含文字和嵌入的变量。文字讯息是一个以长度字节开始、后跟ASCII字符的ASCII字符串。文字讯息的最大长度为120个字节（包括长度字节）。

格式: <Text><Variable><Text><Variable>...

Text（文字）域由ASCII字符组成。

Variable（变量）域定义了调制解调器模块从本地CPU读取的嵌入的数据值、格式和讯息中的位置。百分号（%）字符用于标记变量域的起始和结束。Address（地址）和Left（左边）域用冒号进行分隔。Left（左边）和Right（右边）域之间的分隔符可以是点号或逗号，并用作格式变量中的小数点。变量域的语法是：

%Address:Left.Right Format%

Address（地址）域指定了嵌入的数据值的地址、数据类型和大小（例如，VD100、VW50、MB20或T10）。允许使用下列数据类型：I、Q、M、SM、V、T（只适用于字）、C（只适用于字）和AI（只适用于字）。字节、字和双字的大小均是允许的。

Left（左边）域定义了小数点左边将要显示的数字的位数。该数值应足够大，以便在需要时显示包括负号在内的嵌入变量的期望范围。如果Left（左边）域为零，则该值前面将显示零。Left（左边）域的有效值为0到10。

Right（右边）域定义了小数点右边将要显示的位数。小数点右边的零将始终显示。如果Right（右边）域的值为零，则显示该数值时将不带小数点。Right（右边）域的有效值为0到10。

Format（格式）域指定了所嵌入的数值的显示格式。下列字符对于格式域来说都是允许的：

- i - 带符号的整数
- u - 不带符号的整数
- h - 十六进制
- f - 浮点数/实数

实例: “Temperature = %VW100:3.1i% Pressure = %VD200:4.3f%”

## CPU数据传送讯息格式

CPU数据传送，无论是CPU到CPU还是CPU到Modbus的数据传送，都由CPU数据传送讯息格式指定。CPU数据传送讯息是一个ASCII字符串，它可以定义设备之间的数据传送的数量，最大不能超过120个字节的讯息长度（119个字符加上一个长度字节）。ASCII空格字符可用于分隔数据传送说明，但并非必须这么做。所有的数据传送说明都将在一个链接中执行。数据传送将按照在讯息中定义的顺序进行。如果在数据传送中检测到错误，则将断开与远程设备的连接，并且不再进行后续的事务处理。

如果操作被指定为读，则从远程设备的Remote\_address开始读取所计字数，然后将其写入本地CPU中从Local\_address开始的V内存中。

如果操作被指定为写，则从本地CPU的Local\_address开始读取所计字数，然后将其写入从Remote\_address开始的远程设备中。

格式: <Operation>=<Count>,<Local\_address>,<Remote\_address>

Operation（操作）域由一个ASCII字符组成，并定义了传送的类型。

R - 从远程设备读取数据  
W - 将数据写入远程设备

Count（计数）域指定了将要传送的字数。Count（计数）域的有效范围是1至100字。

Local\_address（本地地址）域指定了用于数据传送的本地CPU的V内存地址（例如，VW100）。

Remote\_address（远程地址）域指定了用于数据传送的远程设备的地址（例如，VW500）。该地址始终指定为V内存地址，即使是到Modbus设备的数据传送。如果远程设备为Modbus设备，则V内存地址和Modbus地址之间的转换如下：

$$\begin{aligned} \text{Modbus地址} &= 1 + (\text{V内存地址} / 2) \\ \text{V内存地址} &= (\text{Modbus地址} - 1) * 2 \end{aligned}$$

实例: R=20,VW100, VW200 W=50,VW500,VW1000 R=100,VW1000,VW2000

# 11

## 使用USS协议库控制MicroMaster驱动器

通过将预先配置的子程序和中断例行程序（专门为使用USS协议与驱动器进行通讯而设计）包含在STEP 7-Micro/WIN指令库中，将使MicroMaster驱动器的控制更轻松。使用USS指令，可对实际的驱动器和读/写驱动器的参数进行控制。

在STEP 7-Micro/WIN指令树的库文件夹中，可找到这些指令。当选择一条USS指令时，将自动添加一个或多个相关联的子程序（USS1至USS7）。

### 本章内容

USS协议的使用要求 .....	322
计算与驱动器进行通讯所需要的时间 .....	323
使用USS指令 .....	324
用于USS协议的指令 .....	325
USS协议的范例程序 .....	332
USS执行的错误代码 .....	333
连接和设置MicroMaster系列3的驱动器 .....	334
连接和安装MicroMaster系列4的驱动器 .....	337

## USS协议的使用要求

STEP 7-Micro/WIN指令库提供了支持USS协议的14个子程序、3个中断例行程序和8个指令。USS指令将使用S7-200中的下列资源:

- 启用USS协议将使端口0专门用于USS通讯。  
可使用USS\_INIT指令来选择用于端口0的USS或PPI。(关于USS协议,请参见SIMOTION MicroMaster驱动器的USS协议。)在选择使用与驱动器进行通讯的USS协议之后,不能再将端口0用于任何其它的用途,包括与STEP7-Micro/WIN的通讯。  
在使用USS协议开发应用程序期间,应使用计算机中与PROFIBUS CP卡相连接的CPU226、CPU 226XM或EM277 PROFIBUS-DP模块。第二个通讯端口将允许STEP7-Micro/WIN在运行USS协议时对应用程序进行监控。
- USS指令将影响与端口0上的自由端口通讯相关联的所有SM单元。
- USS指令使用了14个子程序和3个中断例行程序。
- USS指令将使应用程序所需要的存储空间最多达到3450个字节。根据所使用的特定USS指令,这些指令的支持程序将使控制程序的负荷至少增加2150个字节,最多可达到3450个字节。
- 用于USS指令的变量需要一个400字节的V存储器块。用户可分配该字节块的起始地址,并可为USS变量保留。
- 某些USS指令还可能需要一个16字节的通讯缓冲区。至于指令参数,可为该缓冲区提供一个V存储器中的起始地址。建议为USS指令的每个事例分配一个唯一的缓冲区。
- 当完成计算时,USS指令将使用AC0至AC3的累加器。也可在程序中使用累加器;然而,累加器中的值将由USS指令进行修改。
- 在中断例行程序中,不能使用USS指令。



### 提示

为将端口0的操作切换回PPI,以便能够与STEP7-Micro/WIN进行通讯,可使用另一个USS\_INIT指令来重新分配端口0。

也可将S7-200上的模式开关设置为STOP模式。这样可重新设置端口0的参数。须知:停止与驱动器的通讯,也将停止驱动器。

## 计算与驱动器进行通讯所需要的时间

驱动器的通讯与S7-200的扫描是不同步的。在完成一个驱动器通讯事务处理之前，S7-200通常将完成多次扫描。下列因素均有助于确定所需要的时间量：现有驱动器数目、波特率以及S7-200的扫描时间。

当使用参数访问指令时，某些驱动器将需要更长的延迟。参数访问所需要的时间量将取决于驱动器的类型以及正在访问的参数。

一旦USS\_INIT指令指定端口0使用USS协议之后，S7-200就将按表11-1所示的周期定期轮询所有有效的驱动器。必须设置每个驱动器的超时参数宜允许执行该任务。

表11-1 通讯时间

波特率	有效驱动器两次轮询之间的时间 (尚未激活任何有效的参数访问指令)
1200	240毫秒(最大值)乘以驱动器的数目
2400	130毫秒(最大值)乘以驱动器的数目
4800	75毫秒(最大值)乘以驱动器的数目
9600	50毫秒(最大值)乘以驱动器的数目
19200	35毫秒(最大值)乘以驱动器的数目
38400	30毫秒(最大值)乘以驱动器的数目
57600	25毫秒(最大值)乘以驱动器的数目
115200	25毫秒(最大值)乘以驱动器的数目



### 提示

每次只能有一个USS\_RPM\_x或USS\_WPM\_x指令是有效的。在用户逻辑启用新的指令之前，每个指令的Done输出应发出完成信号。

每个驱动器应只使用一个USS\_CTRL指令。

## 使用USS指令

为在S7-200控制器程序中使用USS协议指令，请按照下列步骤操作：

1. 在程序中插入USS\_INIT指令，并只在每次扫描时执行USS\_INIT指令。可使用USS\_INIT指令来启用或修改USS通讯参数。

当插入USS\_INIT指令时，多个隐藏的子程序和中断例行程序均将自动添加到程序中。

2. 在每个有效驱动器的程序中，只能放置一个USS\_CTRL指令。

可根据需要添加足够的USS\_RPM\_x和USS\_WPM\_x指令，但在这些指令中，每次只能有一个是有效的。

3. 通过右击指令树中的程序块节点（以便获得菜单），为库指令分配V内存。

选择库内存选项，以显示“库内存分配”对话框。

4. 配置驱动器参数，以便与程序中所使用的波特率和地址相匹配。

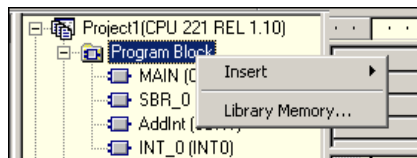


图11-1 分配指令库的V内存

5. 连接S7-200与驱动器之间的通讯电缆。

确保与驱动器连接的所有控制设备，例如S7-200，已经用一条短粗电缆连接到与驱动器相同的地线或星形点。

### 当心

具有不同参考电位的互连设备将可能导致出现意外的电流流过互连电缆。这些意外电流有可能导致通讯故障，甚至可能损坏设备。

请确保用通讯电缆连接的所有设备既可以共有一个共同的电流基准，也可以对其进行绝缘，以避免产生不必要的电流。

防护罩必须连接到机壳地线或9针连接器的1号针脚。建议将MicroMaster驱动器上的2-0V接线柱连接到机壳地线。



## 用于USS协议的指令

### USS\_INIT指令

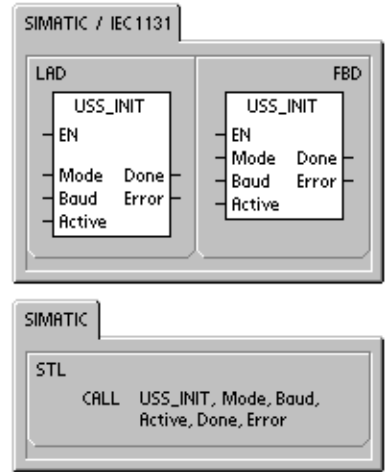
USS\_INIT指令用于激活和启用，或禁止MicroMaster驱动器通讯。在任何其它USS指令可使用之前，必须没有错误地执行USS\_INIT指令。指令完成后，将立即设置Done位，然后继续执行下一个指令。

当EN输入处于接通状态时，指令将在每次扫描时执行。

对于通讯状态的每次变化，可只执行一次USS\_INIT指令。可使用边沿检测指令脉冲将EN输入触发为接通状态。为改变初始化参数，可执行新的USS\_INIT指令。

Mode的值选择通讯协议：输入值为1时表示将端口0分配给USS协议，并激活协议，而输入值为0时表示将端口0分配给PPI，并禁止USS协议。

Baud将波特率设置为1200、2400、4800、9600、19200、38400、57600或115200。



Active将指示有效的驱动器是哪一个。某些驱动器只支持地址0至30。

表11-2 用于USS\_INIT指令的参数

输入/输出	数据类型	操作数
Mode	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、常量、*VD、*AC、*LD
Baud、Active	DWORD	VD、ID、QD、MD、SD、SMD、LD、常量、AC *VD、*AC、*LD
Done	BOOL	I、Q、M、S、SM、T、C、V、L
Error	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD

图11-2所示为关于有效驱动器输入的描述和格式。在后台中，将自动轮询标记为激活的任何驱动器，以便控制驱动器、收集状态和避免驱动器中的串行链接超时。

请参考表11-1来计算状态轮询之间的时间。

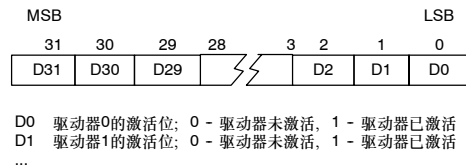
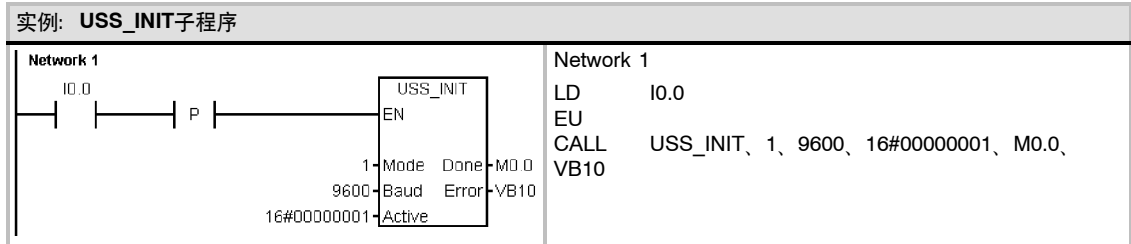


图11-2 已激活驱动器参数的格式

当USS\_INIT指令完成时，Done输出将变为接通。Error输出字节包含指令执行的结果。表11-6定义执行指令时可能导致的出错条件。



## USS\_CTRL指令

USS\_CTRL指令用于控制已激活的MicroMaster驱动器。USS\_CTRL指令将所选的命令放置在通讯缓冲区中，它们随后将被发送到已编址的驱动器（驱动器参数）中，前提是在USS\_INIT指令的激活参数中已经选择了该驱动器。

每个驱动器应只分配一个USS\_CTRL指令。

某些驱动器将只报告正数值的速度。如果速度是负的，驱动器将仍把速度报告为正的，只是要将D\_Dir（方向）位取反。

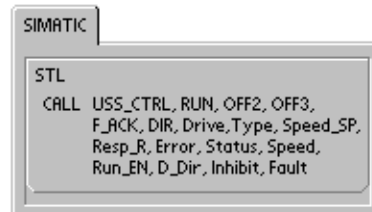
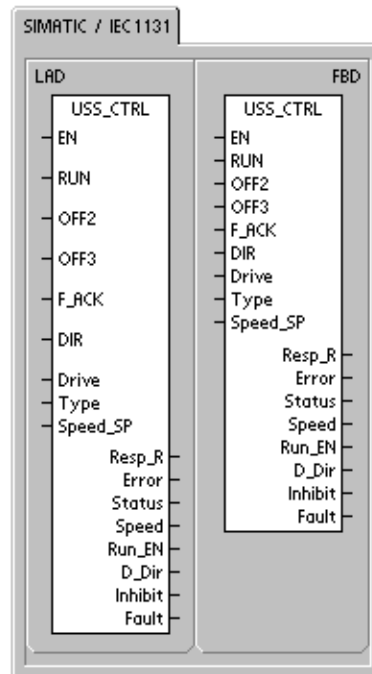
EN位必须处于接通状态，以激活USS\_CTRL指令。该指令应始终激活。

RUN（RUN/STOP）将指示驱动器是接通（1），还是断开（0）。当RUN位接通时，MicroMaster驱动器将接收到一条命令，以启动特定速度和方向下的运行。为了驱动器的运行，必须具备下列条件：

- 驱动器必须选择为USS\_INIT中的激活。
- OFF2和OFF3必须设置为0。
- Fault和Inhibit必须设置为0。

当RUN处于断开状态时，将发送一条命令给MicroMaster驱动器，以缓慢降低速度，直到电机完全停止。OFF2位用于使MicroMaster驱动器能够慢慢滑行到停止。OFF3位用于命令MicroMaster驱动器迅速停止。

Resp\_R（所接收的响应）位将确认驱动器的响应。将轮询所有激活的驱动器，以获得最新的驱动器状态信息。每当S7-200接收到驱动器的响应时，就将接通Resp\_R位以扫描一次，并更新以下数值。



F\_ACK（故障确认）位用于确认驱动器中的故障。当F\_ACK从0跳转到1时，驱动器将对故障（Fault）清零。

DIR（方向）位将指示驱动器应移动的方向。

Drive（驱动器地址）输入是MicroMaster驱动器的地址，USS\_CTRL命令将发送给该驱动器。有效的地址：0到31

Type（驱动器类型）输入可选择驱动器的类型。对于MicroMaster 3（或更早的）驱动器，将Type设置为0。对于MicroMaster 4驱动器，则将Type设置为1。

表11-3 USS\_CTRL指令的参数

输入输出	数据类型	操作数
RUN、OFF2、OFF3、F_ACK、DIR	BOOL	I、Q、M、S、SM、T、C、V、L、功率流
Resp_R、Run_EN、D_Dir、Inhibit、Fault	BOOL	I、Q、M、S、SM、T、C、V、L
Drive、Type	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD、常量
Error	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD
Status	WORD	VW、T、C、IW、QW、SW、MW、SMW、LW、AC、AQW、*VD、*AC、*LD
Speed_SP	REAL	VD、ID、QD、MD、SD、SMD、LD、AC、*VD、*AC、*LD、常量
Speed	REAL	VD、ID、QD、MD、SD、SMD、LD、AC、*VD、*AC、*LD

Speed\_SP (速度设定值) 是全速度百分比形式的驱动器速度。Speed\_SP 的值为负, 将使驱动器倒转其旋转方向。范围: -200.0%到200.0%

Error是包含有驱动器最新通讯请求结果的错误字节。表11-6定义了执行指令时可能导致的出错条件。

Status为驱动器所返回的状态字原始数值。图11-3显示了标准状态字和主反馈的状态位。

Speed是全速度百分比形式的驱动器速度。范围: -200.0%到200.0%

Run\_EN (RUN激活) 将指示驱动器是运行 (1), 还是停止 (0)。

D\_Dir将指示驱动器的旋转方向。

Inhibit将指示驱动器上的禁止位的状态 (0 - 未禁止、1 - 禁止)。为对Inhibit位清零, Fault位必须处于断开状态, 且RUN、OFF2和OFF3输入也必须断开。

Fault将指示故障位的状态 (0 - 没有故障、1 - 有故障)。驱动器将显示故障的代码。(参见驱动器手册)。为使故障位清零, 可纠正故障的原因, 并接通F\_ACK位。

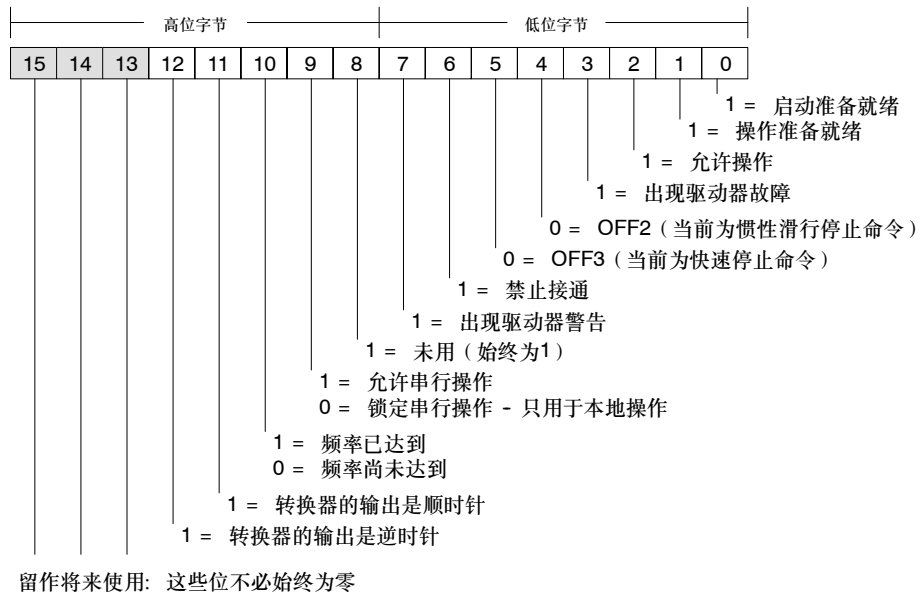


图11-3 MicroMaster 3标准状态字和主反馈的状态位

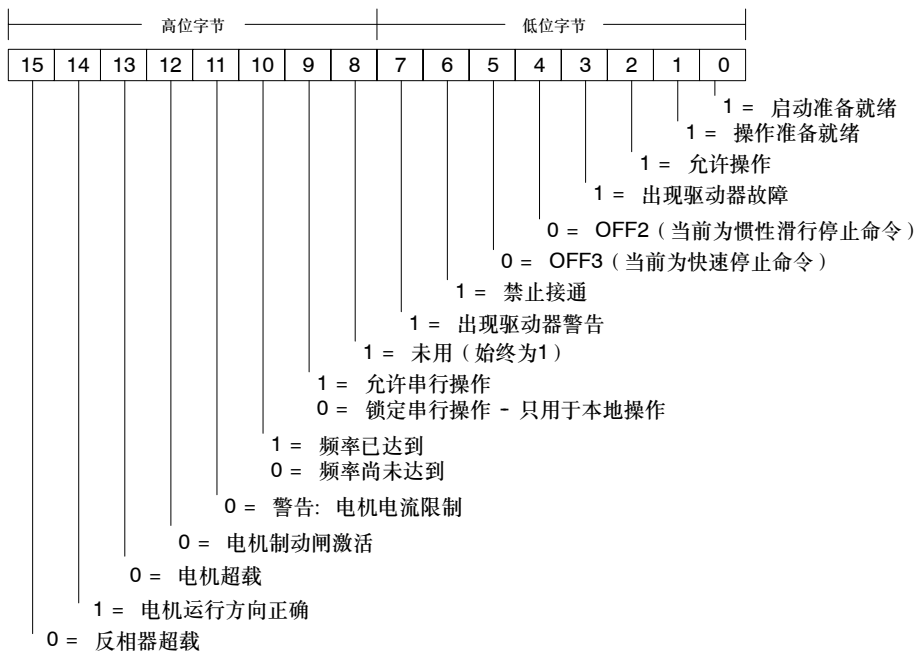


图11-4 MicroMaster 4标准状态字和主反馈的状态位

**实例: USS\_CTRL子程序**

**Network 1**

SM0.0	EN
I0.0	RUN
I0.1	OFF2
I0.2	OFF3
I0.3	F_ACK
I0.4	DIR

0	Drive	Resp_R	M0.0
1	Type	Error	VB2
100.0	Speed_SP	Status	VW4
		Speed	VD6
		Run_EN	Q0.0
		D_Dir	Q0.1
		Inhibit	Q0.2
		Fault	Q0.3

仅用于语句表的显示:

Network 1 //驱动器0的控制盒

```
LD SM0.0
CALL USS_CTRL, I0.0, I0.1, I0.2, I0.3, I0.4,
0, 1, 100.0, M0.0, VB2, VW4, VD6, Q0.0,
Q0.1, Q0.2, Q0.3
```

**LAD或FBD的显示:**

Network 1 //驱动器0的控制盒

```
LD SM0.0
= L60.0
LD I0.0
= L63.7
LD I0.1
= L63.6
LD I0.2
= L63.5
LD I0.3
= L63.4
LD I0.4
= L63.3
LD L60.0
CALL USS_CTRL, L63.7, L63.6, L63.5,
L63.4, L63.3, 0, 1, 100.0, M0.0, VB2, VW4,
VD6, Q0.0, Q0.1, Q0.2, Q0.3
```

## USS\_RPM\_x指令

存在USS协议的三种读操作指令：

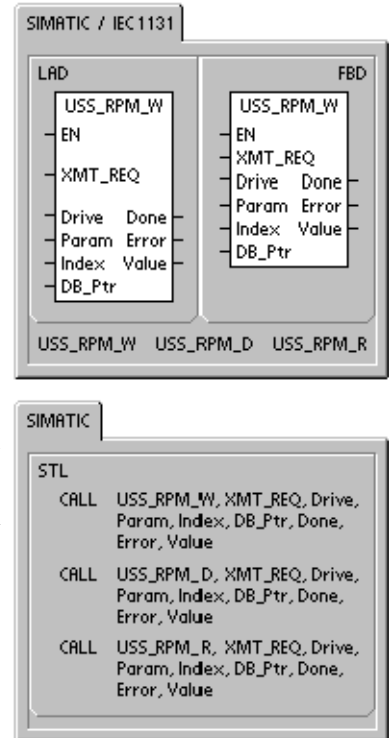
- USS\_RPM\_W指令将读取无符号字参数。
- USS\_RPM\_D指令将读取无符号双字参数。
- USS\_RPM\_R指令读取浮点型参数。

每次只能有一个读（USS\_RPM\_x）或写（USS\_WPM\_x）指令是激活的。

USS\_RPM\_x事务处理将在MicroMaster驱动器确认已经收到命令或错误条件已发送时完成。当该过程等待响应时，逻辑扫描将继续执行。

EN位必须接通，以便激活请求的传输，并应一直保持到设置Done位，发出该过程已经完成的信号。例如，当XMT\_REQ输入处于接通状态时，USS\_RPM\_x请求将在每次扫描时发送给MicroMaster驱动器。因此，XMT\_REQ输入应通过边沿探测元素脉冲进行触发，该边沿探测元素将在EN输入每次进行正向转换时发送一个请求。

Drive输入是MicroMaster驱动器的地址，USS\_RPM\_x命令将发送给该驱动器。各个驱动器的有效地址为0到31。



Param为参数的数目。下标是将要读取的参数的下标值。Value是所返回的参数值。16字节缓冲区的地址必须提供给DB\_Ptr输入。USS\_RPM\_x指令将使用该缓冲区来存储发送给MicroMaster驱动器的命令的执行结果。

当USS\_RPM\_x指令完成时，Done输出将接通，且Error输出字节和Value输出均包含指令的执行结果。表11-6定义了执行指令时可能导致的出错条件。Error和Value输出只有在Done输出接通时才是有效的。

表11-4 USS\_RPM\_x的有效操作数

输入/输出	数据类型	操作数
XMT_REQ	BOOL	I、Q、M、S、SM、T、C、V、L、由上升沿探测元素控制的功率流
Drive	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD、常量
Param、Index	WORD	VW、IW、QW、MW、SW、SMW、LW、T、C、AC、AIW、*VD、*AC、*LD、常量
DB_Ptr	DWORD	&VB
Value	WORD、DWORD、REAL	VW、IW、QW、MW、SW、SMW、LW、T、C、AC、AQW、*VD、*AC、*LD、VD、ID、QD、MD、SD、SMD、LD、*VD、*AC、*LD
Done	BOOL	I、Q、M、S、SM、T、C、V、L
Error	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD

## USS\_WPM\_x指令

存在USS协议的三种写操作指令：

- USS\_WPM\_W指令将写入无符号字参数。
- USS\_WPM\_D指令将写入无符号双字参数。
- USS\_WPM\_R指令将写入浮点型参数。

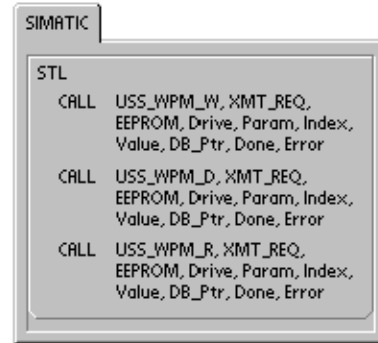
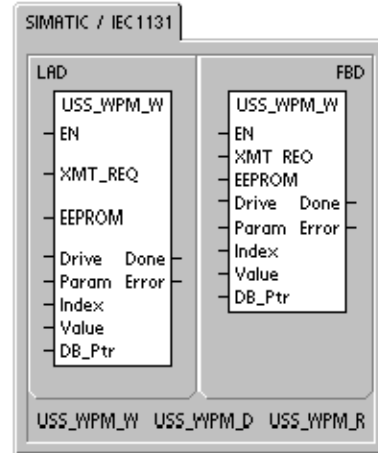
每次只能有一个读（USS\_RPM\_x）或写（USS\_WPM\_x）指令激活。

USS\_WPM\_x事务处理将在MicroMaster驱动器确认已经收到命令或错误条件已发送时完成。在该过程等待响应时，逻辑扫描将继续执行。

EN位必须接通，以便激活请求的传输，并应一直保持到设置Done位，发出该过程已经完成的信号。例如，当XMT\_REQ输入接通时，USS\_WPM\_x请求将在每次扫描时发送给MicroMaster驱动器。因此，XMT\_REQ输入应通过边沿探测元素脉冲进行触发，该元素将在EN输入每次进行正向转换时发送一个请求。

Drive输入是MicroMaster驱动器的地址，USS\_WPM\_x命令将发送给该驱动器。各个驱动器的有效地址为0到31。

Param为参数数目。下标是要写入的参数的下标值。Value是要写入到驱动器RAM中的参数值。对于MicroMaster 3驱动器，也可基于配置P971（EEPROM存储器控件）的方式，将该数值写入驱动器的EEPROM中。



16字节缓冲区的地址必须提供给DB\_Ptr输入。USS\_WPM\_x指令将使用该缓冲区来存储发送给MicroMaster驱动器的命令的执行结果。

当USS\_WPM\_x指令完成时，Done输出将接通，且Error输出字节将包含指令的执行结果。表11-6定义了执行指令时可能导致的出错条件。

当EEPROM输入接通时，指令将同时写入驱动器的RAM和EEPROM中。当输入断开时，指令将只写入驱动器的RAM中。因为MicroMaster 3驱动器不支持该功能，必须确保该输入断开，以便MicroMaster 3驱动器使用该指令。

表11-5 USS\_WPM\_x指令的有效操作数

输入/输出	数据类型	操作数
XMT_REQ	BOOL	I、Q、M、S、SM、T、C、V、L、由上升沿探测元素控制的功率流
EEPROM	BOOL	I、Q、M、S、SM、T、C、V、L、功率流
Drive	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD、常量
Param、Index	WORD	VW、IW、QW、MW、SW、SMW、LW、T、C、AC、AIW、*VD、*AC、*LD、常量
DB_Ptr	DWORD	&VB
数值	WORD DWORD、 REAL	VW、IW、QW、MW、SW、SMW、LW、T、C、AC、AQW、*VD、*AC、*LD VD、ID、QD、MD、SD、SMD、LD、*VD、*AC、*LD
Done	BOOL	I、Q、M、S、SM、T、C、V、L
Error	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD

**当心**

当使用USS\_WPM\_x指令来更新存储在驱动器EEPROM中的参数集时，必须确保不超过EEPROM写周期的最大值（大约为50,000次）。

超出写周期的最大值将导致所存储的数据发生崩溃，从而导致数据丢失。读周期的次数则没有限制。

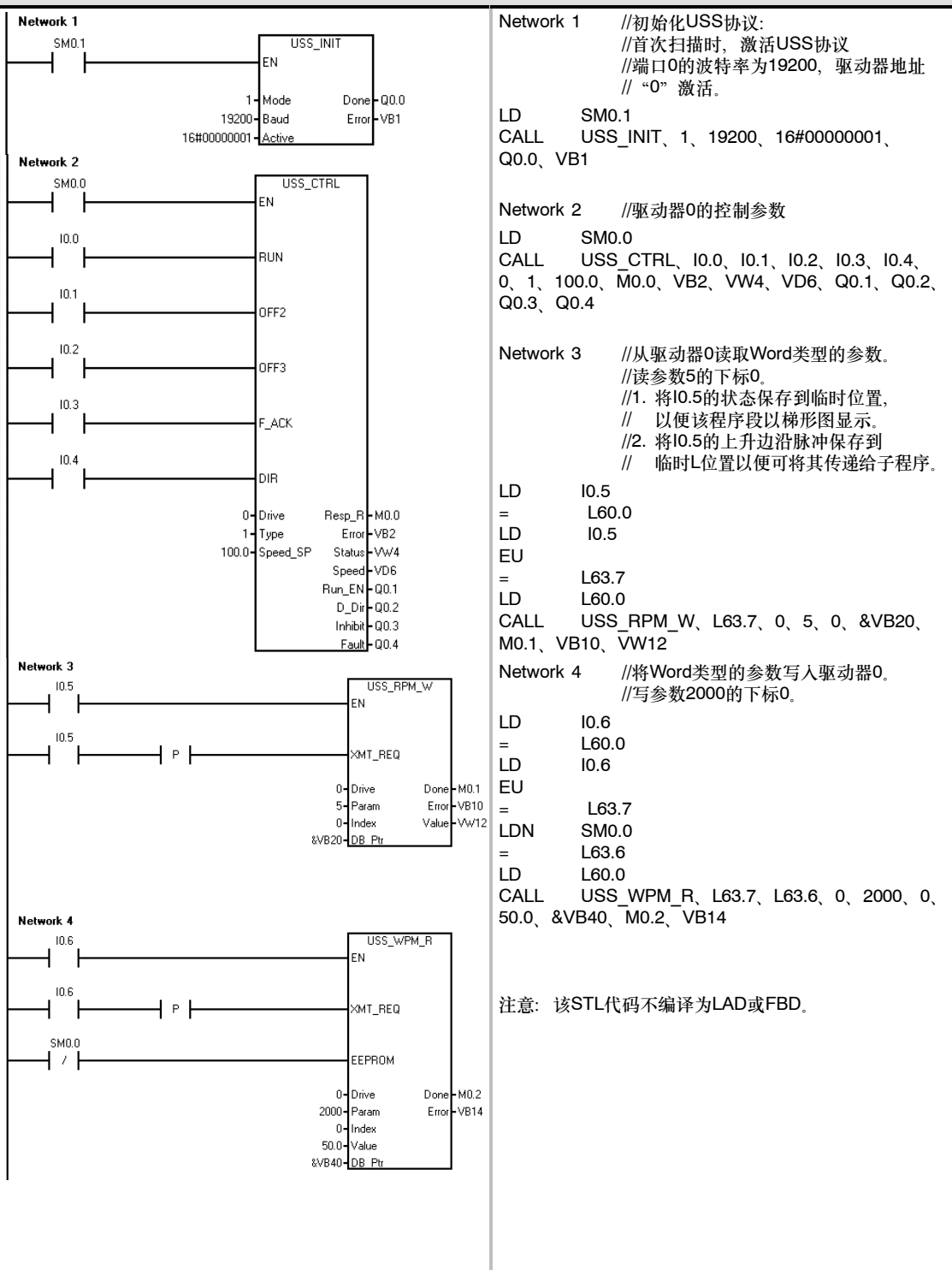
如果需要频繁写入驱动器参数，则首先应将驱动器中的EEPROM存储器控制参数设置为零（对于MicroMaster 3驱动器）和断开MicroMaster 4驱动器的EEPROM输入。

**实例: USS\_RPM\_x和USS\_WPM\_x**

<p><b>Network 1</b></p> <p>USS_RPM_W</p> <ul style="list-style-type: none"> <li>EN</li> <li>XMT_REQ</li> <li>0- Drive Done M0.0</li> <li>3- Param Error VB10</li> <li>0- Index Value VW200</li> <li>&amp;VB100- DB Ptr</li> </ul> <p><b>Network 2</b></p> <p>USS_WPM_W</p> <ul style="list-style-type: none"> <li>EN</li> <li>XMT_REQ</li> <li>EEPROM</li> <li>0- Drive Done M0.1</li> <li>971- Param Error VB11</li> <li>0- Index</li> <li>1- Value</li> <li>&amp;VB120- DB Ptr</li> </ul>	<p><b>Network 1</b> //两个触点必须具有同样的地址。</p> <pre> LD I0.0 = L60.0 LD I0.0 EU = L63.7 LD L60.0 CALL USS_RPM_W, L63.7, 0, 3, 0, &amp;VB100, M0.0, VB10, VW200     </pre> <p><b>Network 2</b> //两个触点必须具有相同的地址</p> <pre> LD I0.1 = L60.0 LD I0.1 EU = L63.7 LDN SM0.0 = L63.6 LD L60.0 CALL USS_WPM_W, L63.7, L63.6, 0, 971, 0, 1, &amp;VB120, M0.1, VB11     </pre>
---	---

## USS协议的采样程序

实例: USS指令 以STL形式正确显示的范例程序





## USS执行错误代码

表11-6 USS指令的执行错误代码

错误代码	描述
0	无错
1	驱动器没有响应
2	检测到驱动器的响应中出现检验和错误
3	检测到驱动器的响应中出现奇偶校验错误
4	由用户程序干扰所导致的错误
5	试图运行非法的命令
6	提供了非法的驱动器地址
7	未设置用于USS协议的通讯端口
8	通讯端口正在忙于处理指令
9	驱动器速度输入超出了输入范围
10	驱动器响应的长度不正确
11	驱动器响应的第一个字符不正确
12	驱动器响应中的长度字符, USS指令不支持
13	响应了错误的驱动器
14	所提供的DB_Ptr地址不正确
15	所提供的参数数目不正确
16	选择了无效的协议
17	USS已激活; 不允许进行改变
18	指定了非法的波特率
19	无任何通讯: 驱动器未设置为ACTIVE
20	驱动器响应中的参数或数值不正确或包含错误代码
21	用返回双字类型的数值代替所请求的字类型的数值
22	用返回的字类型的数值代替所请求的双字类型的数值

## 连接和安装MicroMaster系列3驱动器

### 连接MicroMaster 3驱动器

可使用标准PROFIBUS电缆和连接器将S7-200连接到MicroMaster系列3 (MM3) 驱动器。参见关于互连电缆正常电缆偏置和终端的图11-5。

#### 当心

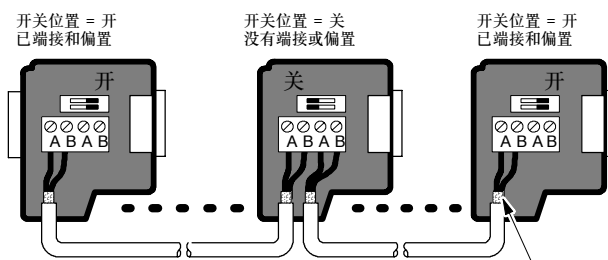
具有不同参考电位的互连设备将可能导致出现意外的电流流过互连电缆。

这些意外的电流有可能导致通讯出现故障，甚至可能损坏设备。

请确保即将用通讯电缆连接的所有设备均具有共同的电流基准，或对其进行绝缘，以避免产生意外的电流。

防护罩必须连接到机壳地线或9针连接器的1号针脚。建议将MicroMaster驱动器上的2-0V接线柱连接到机壳地线。

电缆在两端都必须端接和偏置。



裸防护层: 大约12毫米 (1/2英寸) 必须接触所有单元的金属导轨。



图11-5 网络电缆的偏置和端接

## 安装MicroMaster 3驱动器

在将驱动器连接到S7-200之前，必须确保驱动器具有下列系统参数。可使用驱动器上的键盘来设置参数：

- 重新设定驱动器为出厂设置（供选用）。按下P键：显示P000。按下向上或向下箭头键直到出现P944的显示。按下P键，输入参数。  
P944=1
- 允许对所有参数进行读/写访问。按下P键。按下向上或向下箭头键直到出现P009的显示。按下P键，输入参数。  
P009=3
- 检查驱动器的电机设置。设置将随所使用的电机的不同而不同。按下P键。按下向上或向下箭头键，直到出现驱动器电机设置的显示。按下P键，输入参数。  
P081=电机的额定频率（Hz）  
P082=电机的额定速度（RPM）  
P083=电机的额定电流（A）  
P084=电机的额定电压（V）  
P085=电机的额定功率（kW/HP）
- 设置本地/远程控制模式。按下P键。按下向上或向下箭头键直到出现P910的显示。按下P键，输入参数。  
P910=1 远程控制模式
- 设置RS-485串行通讯口的波特率。按下P键。按下向上或向下箭头键，直到显示P092。按下P键，输入参数。按下向上或向下箭头键，直到出现与RS-485串行通讯口的波特率相对应的数字的显示。按下P键进行输入。  
P092    3            (1200波特)  
          4            (2400波特)  
          5            (4800波特)  
          6            (9600波特 - 默认)  
          7            (19200波特)
- 输入从属装置地址。各个驱动器（最大31）可通过总线进行操作。按下P键。按下向上或向下箭头键，直到显示P091。按下P键，输入参数。按下向上或向下箭头键，直到出现想要的从属装置地址的显示。按下P键进行输入。  
P091=0至31。
- 加速时间（供选用）。这就是将电机加速到最大频率时所花费的以秒为单位的时间。按下P键。按下向上或向下箭头键，直到显示P002。按下P键，输入参数。按下向上或向下箭头键，直到出现想要的加速时间的显示。按下P键进行输入。  
P002=0-650.00
- 减速时间（供选用）。这就是将电机减速到完全停止时所花费的以秒为单位的时间。按下P键。按下向上或向下箭头键，直到显示P003。按下P键，输入参数。按下向上或向下箭头键，直到出现想要的减速时间的显示。按下P键进行输入。  
P003=0-650.00

9. 串行链接超时。这是两次传入数据报文之间所允许的最大周期。该特征用于在万一出现通讯故障时断开反相器。

在接收到有效的数据报文之后开始进行计时。如果在指定的时间周期之内没有接收到更多的数据报文，则释放反相器，并显示故障代码F008。将数值设置为零，以便断开控件。可使用表11-1来计算驱动器状态轮询之间的时间。

按下P键。按下向上或向下箭头键，直到显示P093。按下P键，输入参数。按下向上或向下箭头键，直到出现想要的串行链接超时的显示。按下P键进行输入。

P093=0-240 (0为默认值；时间单位为秒)

10. 串行链接额定系统设定值。该数值可改变，但通常等于50Hz或60Hz，该数值将定义PV或SP的相应100%的数值。按下P键。按下向上或向下箭头键，直到显示P094。按下P键，输入参数。按下向上或向下箭头键，直到出现想要的串行链接额定系统定点。按下P键进行输入。

P094=0-400.00

11. USS兼容性 (供选用)。按下P键。按下向上或向下箭头键，直到显示P095。按下P键，输入参数。按下向上或向下箭头键，直到出现与想要的USS兼容相对应的数字的显示。按下P键进行输入。

P095= 0 0.1Hz分辨率 (默认)  
1 0.01Hz分辨率

12. EEPROM存储器控件 (供选用)。按下P键。按下向上或向下箭头键，直到显示P971。按下P键，输入参数。按下向上或向下箭头键，直到出现与想要的EEPROM存储器控件相对应的数字的显示。按下P键进行输入。

P971 = 0 对参数设置 (包含P971) 的修改在电源断开时将丢失。  
1 (缺省) 当电源断开时，对参数设置的修改在周期内仍将保持。

13. 操作显示。按下P，退出参数模式。

## 连接和安装MicroMaster系列4的驱动器

### 连接MicroMaster 4驱动器

为了连接到MicroMaster系列4 (MM4) 驱动器, 可将RS85电缆的末端插入为USS操作所提供的两个具有壳体夹的无螺钉的终端中。标准PROFIBUS电缆和连接器均可用于S7-200的连接。

#### 当心

具有不同参考电位的互连设备将可能导致出现意外的电流流过互连电缆。

这些意外的电流有可能导致通讯出现故障, 甚至可能损坏设备。

请确保即将用通讯电缆连接的所有设备均具有共同的电流基准, 或对其进行绝缘, 以避免产生意外的电流。

防护罩必须连接到机壳地线或9针连接器的1号针脚。建议将MicroMaster驱动器上的2-0V接线柱连接到机壳地线。

如图11-6所示, RS85电缆另一端的两条导线必须插入MM4驱动器接线盒中。为连接MM4驱动器上的电缆, 可拆除驱动器机壳, 以便操作接线盒。关于如何拆除指定驱动器机壳的详细情况, 请参见MM4用户手册。

接线盒连接均用数字作了标签。使用靠近S7-200一侧上的PROFIBUS连接器, 可将电缆的A端连接到驱动器的端子15 (MM420) 或端子30 (MM440)。将电缆连接器的B端连接到端子14 (MM420) 或端子29 (MM440)。

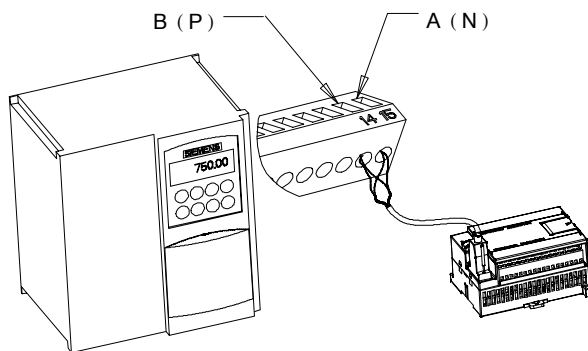


图11-6 连接到MM420接线盒

如果S7-200是网络中的端接节点, 或连接为点对点的连接, 则必须使用连接器的端子A1和B1 (不是A2和B2), 因为他们允许对终端设置进行定义 (例如, DP连接器类型6ES7 972-0BA40-0X40)。

#### 当心

请确保在给单元供电之前, 已经正确更换了驱动器壳体。

如果将驱动器配置为网络中的端接节点, 则终端和偏置电阻器也必须用导线连接到适当的端子连接。例如, 图11-7显示了MM4驱动器终端和偏置必需连接的实例。

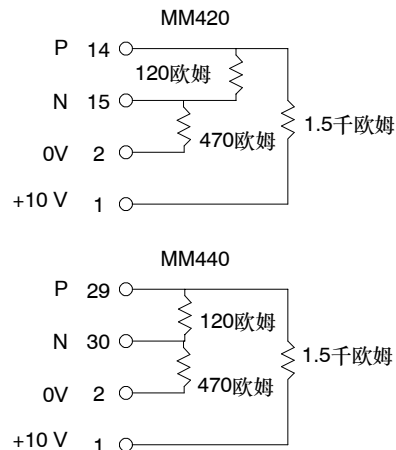


图11-7 终端和偏置的实例

## 安装MM4驱动器

在将驱动器连接到S7-200之前，必须确保驱动器具有下列系统参数。可使用驱动器上的键盘来设置参数：

1. 将驱动器重新设定为出厂设置（供选用）：P0010=30  
P0970=1  
  
如果跳过该步骤，请确保将下列参数设置为这些数值：  
USS PZD长度：P2012 下标0=2  
USS PKW长度：P2013 下标0=127
2. 激活所有参数的读/写访问（高级模式）：P0003=3
3. 检查驱动器的电机设置：P0304=电机额定电压（V）  
P0305=电机额定电流（A）  
P0307=电机额定功率（W）  
P0310=电机额定频率（Hz）  
P0311=电机额定速度（RPM）

设置将随所使用的电机的不同而不同。

为了设置参数P304、P305、P307、P310和P311，必须首先将参数P010设置为1（快速调试模式）。当完成了参数的设置时，可将参数P010设置为0。参数P304、P305、P307、P310和P311只能在快速调试模式下进行修改。

4. 设置本地/远程控制模式：P0700 下标0=5
5. 将频率设定值的选择设置为COM链接上的USS：P1000 下标0=5
6. 加速时间（供选用）：P1120=0到650.00  
这就是将电机加速到最大频率时所花费的以秒为单位的时间。
7. 减速时间（供选用）：P1121=0到650.00  
这就是将电机减速到完全停止时所花费的以秒为单位的时间。
8. 设置串行链接参考频率：P2000=1到650Hz
9. 设置USS标准：P2009 下标0=0
10. 设置RS-485串行通讯口的波特率：P2010 下标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. 输入从属装置地址：P2011 下标0=0到31  
各个驱动器（最大31）可通过总线进行操作。
12. 设置串行链接超时：P2014 下标0=0到65,535毫秒  
（0=禁止超时）

这是两次传入数据报文之间所允许的最大周期。该特征用于在万一出现通讯时断开反相器。在接收到有效的数据报文之后开始进行计时。如果在指定的时间周期之内没有接收到更多的数据报文，则释放反相器，并显示故障代码F0070。将数值设置为零，以便断开控件。可使用表11-1来计算驱动器状态轮询之间的时间。

13. 将数据从RAM传送到EEPROM：  
P0971=1（启动传送）将参数设置的修改保存到EEPROM

# 12

## 使用Modbus协议库

STEP 7-Micro/WIN指令库通过包含专门为Modbus通讯设计的预配置子例行程序和中断例行程序，使得与Modbus主设备进行通讯更容易。使用“Modbus从属协议指令”，可以配置S7-200以象Modbus RTU从属装置那样操作，并与Modbus主设备通讯。

可以在STEP 7-Micro/WIN指令树的“库”文件夹中找到这些指令。使用这些新指令，可以使S7-200用作Modbus从属装置。选择Modbus从属指令时，一个或多个相关子例行程序自动添加到项目。

### 本章内容

使用Modbus协议的要求 .....	340
Modbus协议的初始化和执行时间 .....	340
Modbus编址 .....	341
使用Modbus从属协议指令 .....	342
Modbus从属协议指令 .....	343

## 使用Modbus协议的要求

“Modbus从属协议”指令使用来自S7-200的下列资源:

- 初始化“Modbus从属协议”将端口0专门用作“Modbus从属协议”通讯。  
当端口用作“Modbus从属协议”通讯时，它不能用作任何其它用途，包括与STEP7-Micro/WIN的通讯。MBUS\_INIT指令控制端口0到Modbus从属协议或PPI的分配。
- “Modbus从属协议”指令影响端口0上所有与自由端口通讯相关的SM位置。
- “Modbus从属协议”指令使用3个子例行程序和2个中断。
- “Modbus从属协议”指令需要1857字节的程序空间用于两个“Modbus从属指令”和支持的例行程序。
- “Modbus从属协议”指令的变量需要779字节的V内存块。此块的起始地址由用户分配，并为Modbus变量保留。



### 提示

要将端口0的操作改变回PPI以便与STEP 7-Micro/WIN通讯，可使用另一个MBUS\_INIT指令重新分配端口0。

也可以将S7-200上的模式开关设置为STOP（停止）模式。这将为端口0重新设定参数。

## Modbus协议的初始化和执行时间

Modbus通讯利用CRC（循环冗余检查）保证通讯信息的完整性。“Modbus从属协议”使用预先计算好的数值表来减少处理讯息所需要的时间。CRC表格的初始化需要大约425毫秒。此初始化在MBUS\_INIT子例行程序中进行，通常在进入RUN（运行）模式后在用户程序的首次扫描中进行。如果MBUS\_INIT子例行程序和任何其它用户初始化所需的时间超出500毫秒扫描监视程序，用户负责重新设定监视程序定时器和保持输出允许（如果扩充模块需要）。输出模块监视程序计时器通过写入模块输出进行重新设定。参见第6章中的监视程序重设指令。

当MBUS\_SLAVE子例行程序服务请求时，扫描时间延长。因为大多数时间花费在计算Modbus CRC上，对于请求和响应中的每个字节，扫描时间延长大约650微秒。最大请求/响应（读或写120个字）延长扫描时间大约165毫秒。



## Modbus编址

通常，将Modbus地址写为5个或6个字符数值，包括数据类型和偏移量。前一个或前2个字符确定数据类型，最后4个字符选择数据类型范围内适当的值。然后Modbus主设备将把这些地址映射到正确的功能中。下列地址由Modbus从属指令支持：

- 000001到000128是映射到Q0.0 Q15.7的离散输出。
- 010001到010128是映射到I0.0 I15.7的离散输入。
- 030001到030032是映射到AIW0到AIW62的模拟输入寄存器。
- 040001到04xxxx是映射到V内存的保持寄存器。

所有Modbus地址都基于1。表12-1显示Modbus地址到S7-200地址的映射。

“Modbus从属协议”允许限制可访问Modbus主设备的输入、输出、模拟输入和保持寄存器（V内存）的数量。

MBUS\_INIT指令的MaxIQ参数指定Modbus主设备允许存取的离散输入或输出（Is或Qs）的最大数目。

MBUS\_INIT指令的MaxAI参数指定Modbus主设备允许存取的输入寄存器（AIWs）的最大数目。

MBUS\_INIT指令的MaxHold参数指定Modbus主设备允许存取的保持寄存器（V内存字）的最大数目。

关于设置Modbus从属装置的内存限制的更多信息参见MBUS\_INIT指令的说明。

表12-1 将Modbus地址映射到S7-200

Modbus地址	S7-200地址
000001	Q0.0
000002	Q0.1
000003	Q0.2
...	...
000127	Q15.6
000128	Q15.7
010001	I0.0
010002	I0.1
010003	I0.2
...	...
010127	I15.6
010128	I15.7
030001	AIW0
030002	AIW2
030003	AIW4
...	...
030032	AIW62
040001	HoldStart
040002	HoldStart+2
040003	HoldStart+4
...	...
04xxxx	HoldStart+2 x (xxxx-1)

### 配置符号表

在为第一个符号输入地址后，表格自动计算并在表格中分配符号的余数。

应该为表格分配开始V位置，占用779字节。确定Modbus从属装置符号的分配不要与分配给Modbus保持寄存器的V内存重叠。Modbus保持寄存器用MBUS\_INIT指令的HoldStart和MaxHold参数分配。如果内存区域有任何重叠，MBUS\_INIT指令返回出错。

## 使用Modbus从属协议指令

要在S7-200程序中使用Modbus从属协议指令，遵循下列步骤：

1. 将MBUS\_INIT指令插入程序，执行MBUS\_INIT指令只能用于一次扫描。可以使用MBUS\_INIT指令初始化或改变Modbus通讯参数。  
插入MBUS\_INIT指令时，几个隐藏的子例行程序和中断例行程序自动添加到程序中。
2. 为Modbus从属协议指令所需要的779个字节的连续V内存分配起始地址。
3. 只将一个MBUS\_SLAVE指令放入程序。此指令每次扫描都被调用以服务任何接收到的请求。
4. 连接S7-200上端口0和Modbus主设备之间的通讯电缆。

### 当心

具有不同参考电位的互连设备将可能导致出现不必要的电流流过互连电缆。这些不必要的电流可能导致通讯出错，甚至可能损坏设备。

请确保用通讯电缆连接的所有设备具有公用电路参考电势，或进行绝缘，以避免产生不必要的电流。

累加器（AC0、AC1、AC2、AC3）被Modbus从属指令利用，并显示在交叉引用列表中。在执行前，Modbus从属指令的累加器中值被保存，并恢复到Modbus从属指令完成之前的累加器，确保当执行Modbus从属指令时累加器中的所有用户数据被保存。

Modbus从属协议指令支持Modbus RTU协议。这些指令利用S7-200的自由端口实用程序支持最一般的Modbus功能。支持下列Modbus功能：

表12-2 支持的Modbus从属协议功能

功能	描述
1	读单/多线圈（离散输出）状态。功能1返回输出点（Qs）任何数目的开/关状态。
2	读单/多接点（离散输入）状态。功能2返回输入点（Is）任何数目的开/关状态。
3	读单/多保持寄存器。功能3返回V内存的内容。保持寄存器是Modbus下的字数值，允许在一个请求中读至多120个字。
4	读单/多输入寄存器。功能4返回“模拟输入”数值。
5	写单线圈（离散输出）。功能5将离散输出点设置为指定的数值。该点不是强制的，程序可以重写由Modbus请求写的数值。
6	写单保持寄存器。功能6将单保持寄存器数值写到S7-200的V内存。
15	写多线圈（离散输出）。功能15将多离散输出数值写到S7-200的Q图象寄存器。开始输出点必须以字节边界开始（例如Q0.0或Q2.0），被写的输出数字必须是八的倍数。这是对Modbus从属协议指令的限制。该点不是强制的，程序可以重写由Modbus请求写的数值。
16	写多保持寄存器。功能16将多保持寄存器写到S7-200的V内存。在一个请求中可以写至多120个字。

## Modbus从属协议指令

### MBUS\_INIT指令

MBUS\_INIT指令用于允许和初始化或禁止Modbus通讯。在MBUS\_SLAVE指令可以使用之前，MBUS\_INIT指令必须无错执行。在继续下一个指令之前，指令完成并且Done（完成）位立即被置位。

当EN输入为开时，指令在每次扫描时都执行。

对于每次通讯状态的改变，MBUS\_INIT指令应正好执行一次。因此，EN输入应通过边缘探测元素发出脉冲，或只在首次扫描时执行。

“模式”输入的数值选择通讯协议：输入数值1将端口0分配给Modbus协议并启用协议；输入数值0将端口0分配给PPI并禁用Modbus协议。

参数“波特”设置波特率为1200、2400、4800、9600、19200、38400、57600或115200。

参数Addr设置地址包括数值1到247之间。

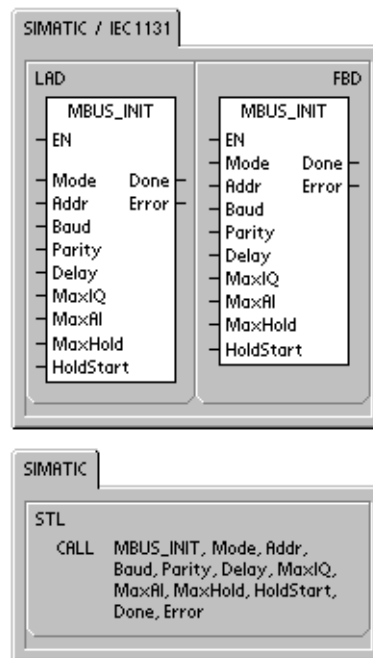


表12-3 MBUS\_INIT指令的参数

输入/输出	数据类型	操作数
模式、地址、奇偶校验	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、常量、*VD、*AC、*LD
波特、HoldStart	DWORD	VD、ID、QD、MD、SD、SMD、LD、AC、常量、*VD、*AC、*LD
延迟、MaxIQ、MaxAI、MaxHold	WORD	VW、IW、QW、MW、SW、SMW、LW、AC、常量、*VD、*AC、*LD
Done（完成）	BOOL	I、Q、M、S、SM、T、C、V、L
错误	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD

参数“奇偶校验”设置为匹配Modbus主站的奇偶校验。接受的数值是：

- 0-无奇偶校验
- 1-奇校验
- 2-偶校验

通过为标准Modbus讯息超时添加指定的毫秒数，参数Delay可延长标准Modbus讯息结束超时条件。当在布线网络上操作时，此参数的典型数值应为0。如果使用带有纠错功能的调制解调器，将Delay设置为数值50到100毫秒。如果使用展布频谱无线电，设置延迟为数值10到100毫秒。Delay值可以为0到32767毫秒。

参数MaxIQ设置可用于Modbus地址00xxx和01xxx的I和Q点的数目，数值为0到128。数值0禁用所有对输入和输出的读和写。MaxIQ的建议值为128，允许存取所有S7-200中的I和Q点。

参数MaxAI设置可用于Modbus地址03xxx的字输入（AI）寄存器的数目，数值为0到32。数值0禁止读模拟输入。MaxAI允许存取所有S7-200模拟输入的建议数值如下：

- CPU221为0
- CPU222为16
- CPU224、226和226XM为32

参数MaxHold设置可用于Modbus地址04xxx在V内存中的字保持寄存器的数目。例如，要允许主设备存取V内存的2000个字节，设置MaxHold为1000个字的数值（保持寄存器）。

参数HoldStart是V内存中保持寄存器的开始地址。此数值通常设置为VB0，所以参数HoldStart设置为&VB0（VB0的地址）。其它V内存地址可以指定为保持寄存器的起始地址以允许VB0在项目的其它地方使用。Modbus主设备已存取以HoldStart开始的MaxHold数目的V内存字。

当MBUS\_INIT指令完成时，Done（完成）输出接通。“出错”输出字节包含执行指令的结果。表12-5定义了执行指令可能导致的错误条件。

## MBUS\_SLAVE指令

MBUS\_SLAVE指令用于服务来自Modbus主设备的请求，并且必须每次扫描执行以允许它检查和响应Modbus请求。

当EN输入为开时，指令在每次扫描时执行。

MBUS\_SLAVE指令没有任何输入参数。

当MBUS\_SLAVE指令响应Modbus请求时，Done（完成）输出为开。如果没有对任何请求服务，Done（完成）输出断开。

“出错”输出包含执行指令的结果。此输出只有当Done（完成）是开时有效。如果Done（完成）为关，出错参数不变。表12-5定义执行指令可能导致的错误条件。

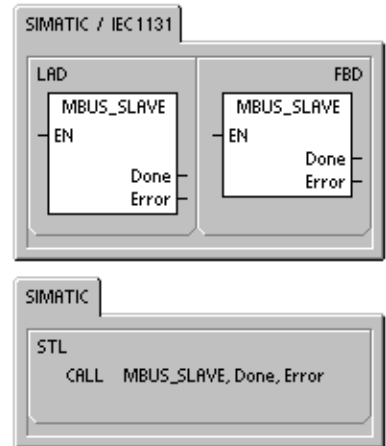


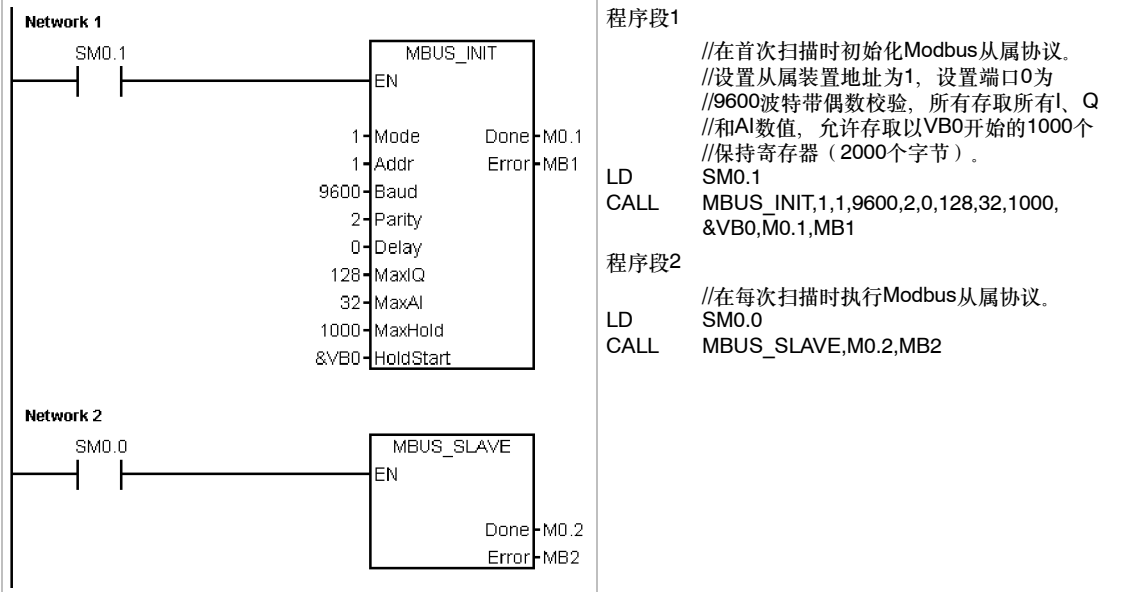
表12-4 MBUS\_SLAVE指令的参数

参数	数据类型	操作数
Done	BOOL	I、Q、M、S、SM、T、C、V、L
Error	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD

表12-5 Modbus从属协议执行出错代码

错误代码	描述
0	无错
1	内存范围出错
2	非法的波特率或奇偶校验
3	非法的从属装置地址
4	Modbus参数的非法数值
5	保持寄存器重叠Modbus从属装置符号
6	接收奇偶校验出错
7	接收CRC出错
8	非法的功能请求/功能不支持
9	请求中有非法内存地址
10	从属功能未启用

## 对Modbus从属协议编程的实例



# 技术规范



## 本章内容

通用技术规范 .....	348
CPU规范 .....	351
数字扩充模块规范 .....	357
模拟扩充模块规范 .....	363
热电偶和RTD扩充模块规范 .....	373
EM 277 PROFIBUS-DP模块规范 .....	385
EM 241调制解调器模块规范 .....	397
EM 253位控模块规范 .....	399
(CP 243-1) 以太网模块规范 .....	405
(CP 243-1 IT) 互联网模块规范 .....	407
(CP 243-2) AS接口模块规范 .....	410
供选用的盒式磁盘 .....	412
I/O扩展电缆 .....	412
RS-232/PPI多台主设备电缆和USB/PPI多台主设备电缆 .....	413
输入模拟程序 .....	417

## 通用技术规范

### 标准适用性

下面所列出的国家标准和国际标准将用于为S7-200系列产品确定合适的性能规范并进行测试。表A-1定义了遵守这些标准的具体细节。

- 欧共体 (CE) 低电压指令73/23/EEC  
EN 61131-2: 可编程控制器 - 设备要求
- 欧共体 (CE) EMC指令89/336/EEC  
电磁辐射标准  
EN 61000-6-3: 住宅业、商业和轻工业  
EN 61000-6-4: 工业环境  
电磁抗干扰标准  
EN 61000-6-2: 工业环境
- 保险商实验室, 包括: UL 508条款 (工业控制设备)  
注册号E75310
- 加拿大标准协会: CSA C22.2 142号 (过程控制设备)
- 工厂互济研究会: FM I类, 2分部, A、B、C和D组, 危险场所, T4A; I类, 2区, IIC, T4。



#### 提示

SIMATIC S7-200系列符合CSA标准。

cULus标志表示S7-200已经通过保险商实验室 (UL) 的审查和鉴定, 符合UL 508和CSA 22.2 142号标准。

### 海事认证

定期提交S7-200产品以获得与特定市场和应用有关的专业代理认证。该表标识S7-200产品已获得认证的代理和证书号。本手册中的S7-200产品不一定全部都获得了这些专业代理认证。如果您需要零件号确切认证的最新列表的附加信息, 请咨询您本地的西门子代理商。

代理	证书号
英国劳氏船级社 (LRS)	99 / 20018 (E1)
美国船级社 (ABS)	01-HG20020-PDA
德国船级社 (GL)	12 045 - 98 HH
挪威船级社 (DNV)	A-8071
法国船级社 (BV)	09051 / A2 BV
日本海事协会 (NK)	A-534



## 继电器电气使用寿命

继电器厂商提供的标准性能数据如图A-1所示。实际性能可能随具体应用场合的不同而有所不同。与负载相适应的外部保护电路将延长继电器接点的使用寿命。

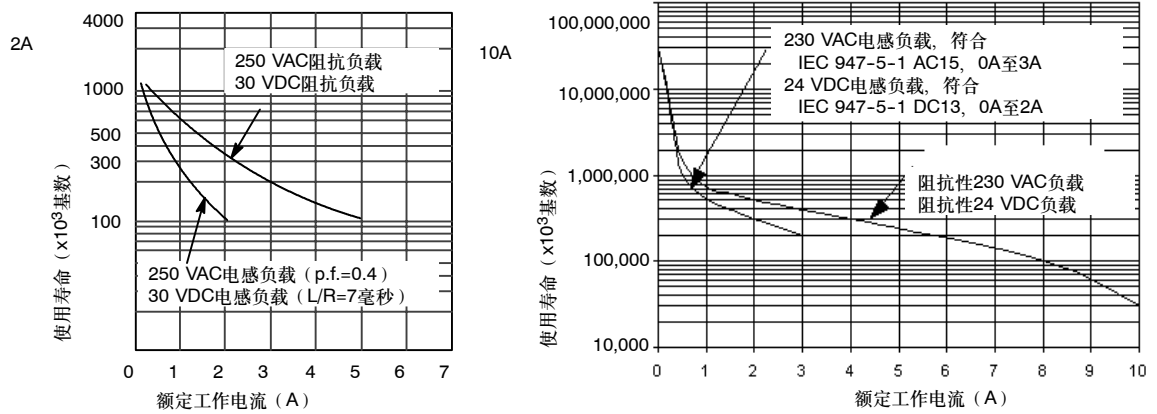


图 A-1 2A和10A情况下继电器的电气使用寿命

## 技术规范

所有S7-200 CPU及其扩展模块均符合表A-1中列出的技术规范。

### 注意

当机械接点接通S7-200 CPU或任意数字扩充模块的输出电源时，将给数字输出发送一个持续约50毫秒的“1”信号。对此，您必须进行规划，特别是在使用对短持续时间脉冲进行响应的设备时。

表 A-1 技术规范

环境条件 — 运输和存储	
EN 60068-2-2, Test Bb, 干热 EN60068-2-1, Test Ab, 严寒	-40° C至+70° C
EN 60068-2-30, Test Db, 湿热	25° C至55° C, 95%湿度
EN 60068-2-14, Test Na, 温度骤变	-40° C至+70° C持续时间3小时, 循环2次
EN 60068-2-31, 倾斜	100毫米, 4次落体, 未包装
EN 60068-2-32, 自由落体	1米, 5次, 运输包装
环境条件 — 操作	
环境温度范围 (入口空气单元以下25毫米)	0° C至55° C水平安装, 0° C至45° C垂直安装 95%未冷凝湿度
大气压力	1080至795 hPa (相当于-1000至2000米的海拔高度)
污染物浓度	SO <sub>2</sub> : <0.5ppm; H <sub>2</sub> S: <0.1ppm; RH<60% 未冷凝
EN 60068-2-14, Test Nb, 温度变化	5° C至55° C, 3° C/分钟
EN 60068-2-27机械震动	15 G, 11毫秒脉冲, 3轴中的每个轴震动6次
EN 60068-2-6正弦波振动	面板安装: 0.30毫米, 10至57赫兹; 2G, 57至150赫兹 DIN横杆安装: 0.15毫米, 10至57赫兹; 1G, 57至150赫兹 每个轴向扫描10次, 1倍频程/分钟
EN 60529, IP20机械保护	防止手指接触时带有可用标准探测器测试的高电压。需要采取外部保护措施, 以防止灰尘、污垢、水以及直径<12.5毫米的杂质。
电磁兼容性 — 抗干扰按照EN 61000-6-2 <sup>1</sup>	
EN 61000-4-2静电放电	对所有表面和通讯端口进行8 kV空气放电、 对暴露的可导电表面进行4 kV接点放电
EN 61000-4-3辐射电磁场	80 MHz至1G赫兹 10 V/m, 使用1 kHz信号进行80%调制
EN 61000-4-4快速瞬态短脉冲	2 kV, 5 kHz, 具有与AC、DC系统电源的连接网络 2 kV, 5 kHz, 具有与数字I/O的连接夹头 1 kV, 5 kHz, 具有与通讯装置的连接夹头
EN 61000-4-5电涌防护	电源: 2 kV不对称, 1 kV对称 I/O 1 kV对称 (24 VDC电路需要外部电涌保护)
EN 61000-4-6传导干扰	0.15至80 MHz 10 V RMS, 在1 kHz下进行80%振幅调制
EN 61000-4-11电压降、短中中断和电压变化	>95%衰减8.3毫秒、83毫秒、833毫秒、4167毫秒
VDE 0160非周期过电压	对于85 VAC线路, 90°相位角、应用峰值390 V、1.3毫秒脉冲 对于180 VAC线路, 90°相位角、应用峰值750 V、1.3毫秒脉冲
电磁兼容性 — 传导和辐射按照EN 61000-6-3 <sup>2</sup> 和EN 61000-6-4	
EN 55011, 等级A, 组1, 传导 <sup>1</sup> 0.15 MHz至0.5 MHz 0.5 MHz至5 MHz 5 MHz至30 MHz	< 79 dB (μV) 准峰值; < 66 dB (μV) 平均值 < 73 dB (μV) 准峰值; < 60 dB (μV) 平均值 < 73 dB (μV) 准峰值; < 60 dB (μV) 平均值
EN 55011, 等级A, 组1, 辐射 <sup>1</sup> 30 MHz至230 MHz 230 MHz至1 G赫兹	40 dB (μV/m) 准峰值; 10米处进行测量 47 dB (μV/m) 准峰值; 10米处进行测量
EN 55011, 等级B, 组1, 传导 <sup>2</sup> 0.15至0.5 MHz 0.5 MHz至5 MHz 5 MHz至30 MHz	<66 dB (μV) 准峰值按对数频率减少到56 dB (μV); <56 dB (μV) 平均值按对数频率减少到46 dB (μV) <56 dB (μV) 准峰值; < 46 dB (μV) 平均值 <60 dB (μV) 准峰值; < 50 dB (μV) 平均值
EN 55011, 等级B, 组1, 辐射 <sup>2</sup> 30 MHz至230 MHz 230 MHz至1 G赫兹	30 dB (μV/m) 准峰值; 10米处进行测量 37 dB (μV/m) 准峰值; 10米处进行测量
高电位绝缘测试	
24V/5V额定电路 115/230V接地电路 115/230V电路到115/230V电路 230V电路到24V/5V电路 115V电路到24V/5V电路	500 VAC (光绝缘边界) 1,500 VAC 1,500 VAC 1,500 VAC 1,500 VAC

<sup>1</sup> 必须将S7-200单元安装在接地的金属架上, 其地线接头直接连接到安装金属。电缆沿金属支架布线。

<sup>2</sup> 设备单元必须安装在接地的金属壳体中。AC输入电源线必须安装有EPCOS B84115-E-A30滤波器或等同设备, 且滤波器到S7-200之间的导线长度不能超过25厘米。24 VDC电源和传感器电源线必须进行屏蔽。

## CPU规范

表 A-2 CPU订购号

订购号	CPU型号	CPU电源 (额定)	CPU输入	CPU输出	可移动连接器
6ES7 211-0AA22-0XB0	CPU 221	24 VDC	6 x 24 VDC	4 x 24 VDC	否
6ES7 211-0BA22-0XB0	CPU 221	120至240 VAC	6 x 24 VDC	4 x 继电器	否
6ES7 212-1AB22-0XB0	CPU 222	24 VDC	8 x 24 VDC	6 x 24 VDC	否
6ES7 212-1BB22-0XB0	CPU 222	120至240 VAC	8 x 24 VDC	6 x 继电器	否
6ES7 214-1AD22-0XB0	CPU 224	24 VDC	14 x 24 VDC	10 x 24 VDC	是
6ES7 214-1BD22-0XB0	CPU 224	120至240 VAC	14 x 24 VDC	10 x 继电器	是
6ES7 216-2AD22-0XB0	CPU 226	24 VDC	24 x 24 VDC	16 x 24 VDC	是
6ES7 216-2BD22-0XB0	CPU 226	120至240 VAC	24 x 24 VDC	16 x 继电器	是
6ES7 216-2AF22-0XB0	CPU 226XM	24 VDC	24 x 24 VDC	16 x 24 VDC	是
6ES7 216-2BF22-0XB0	CPU 226XM	120至240 VAC	24 x 24 VDC	16 x 继电器	是

表 A-3 CPU通用技术规范

订购号	模块名称与说明	尺寸 (毫米) (W x H x D)	重量	损耗	可用的VDC	
					+5 VDC	+24 VDC
6ES7 211-0AA22-0XB0	CPU 221 DC/DC/DC 6输入/4输出	90 x 80 x 62	270 g	3 W	0毫安	180毫安
6ES7 211-0BA22-0XB0	CPU 221 AC/DC/继电器 6输入/4继电器	90 x 80 x 62	310 g	6 W	0毫安	180毫安
6ES7 212-1AB22-0XB0	CPU 222 DC/DC/DC 8输入/6输出	90 x 80 x 62	270 g	5 W	340毫安	180毫安
6ES7 212-1BB22-0XB0	CPU 222 AC/DC/继电器 8输入/6继电器	90 x 80 x 62	310 g	7 W	340毫安	180毫安
6ES7 214-1AD22-0XB0	CPU 224 DC/DC/DC 14输入/10输出	120.5 x 80 x 62	360 g	7 W	660毫安	280毫安
6ES7 214-1BD22-0XB0	CPU 224 AC/DC/继电器 14输入/10继电器	120.5 x 80 x 62	410 g	10 W	660毫安	280毫安
6ES7 216-2AD22-0XB0	CPU 226 DC/DC/DC 24输入/16输出	196 x 80 x 62	550 g	11 W	1000毫安	400毫安
6ES7 216-2BD22-0XB0	CPU 226 AC/DC/继电器 24输入/16继电器	196 x 80 x 62	660 g	17 W	1000毫安	400毫安
6ES7 216-2AF22-0XB0	CPU 226XM DC/DC/DC 24 输入/16 输出	196 x 80 x 62	550 g	11 W	1000毫安	400毫安
6ES7 216-2BF22-0XB0	CPU 226XM AC/DC/继电器 24 输入/16 继电器	196 x 80 x 62	660 g	17 W	1000毫安	400毫安

表 A-4 CPU规范

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
<b>内存</b>					
用户程序大小 (EEPROM)	4096个字节		8192个字节	8192个字节	16384个字节
用户数据 (EEPROM)	2048个字节 (永久存储)		5120个字节 (永久存储)	5120个字节 (永久存储)	10240个字节 (永久存储)
备份 (超级电容) (供选用电池)	50小时典型 (最少8小时, 40°C) 200日典型		100小时典型 (最少70小时, 40°C) 200日典型		190小时典型 (最少120小时, 40°C) 200日典型
<b>I/O</b>					
固定数字输入/输出	6个输入/4个输出	8个输入/6个输出	14个输入/10个输出	24个输入/16个输出	
数字I/O图像大小	256 (128进/128出)				
模拟I/O图像大小	无	32 (16进/16出)	64 (32进/32出)		
允许的最多扩充模块数	无	2个模块 <sup>1</sup>	7个模块 <sup>1</sup>		
允许的最多智能模块数	无	2个模块 <sup>1</sup>	7个模块 <sup>1</sup>		
脉冲捕获输入	6	8	14		
高速计数器 单相 双相	总共4个计数器 4个以30 kHz工作 2个以20 kHz工作		总共6个计数器 6个以30 kHz工作 4个以20 kHz工作		
脉冲输出	2个以20 kHz工作 (只适用于DC输出)				

A

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
<b>常规</b>					
定时器	总共256个定时器；4个定时器（1毫秒）；16个定时器（10毫秒）；236个定时器（100毫秒）；				
计数器	256（由超级电容器或电池支持）				
内部内存位 断电时进行存储	256（由超级电容器或电池支持） 112（存储到EEPROM）				
定时中断	2个，1毫秒分辨率				
边沿中断	4个上升沿和/或4个下降沿				
模拟调整	1个，8位分辨率		2个，8位分辨率		
布尔型执行速度	0.37微秒/指令				
日历时钟	供选用盒式磁盘		固定		
盒式磁盘选项	内存、电池和日历时钟		内存和电池		
<b>通讯内置</b>					
端口	1 RS-485端口			2 RS-485端口	
PPI, DP/T波特率	9.6、19.2、187.5 KB				
自由端口波特率	1.2 KB到115.2 KB				
每段最大电缆长度	带有绝缘中继器：对1000米的长度，可达到187.5 KB；对1200米的长度，可达到38.4 KB 不带绝缘中继器：50米				
最大站数	每段32个，每网络126个				
最大主设备数	32				
对等网络（PPI主设备模式）	是（NETR/NETW）				
MPI连接	总共4个，2个保留（1个用于PG，1个用于OP）				

1 您必须计算功率分配，以便确定S7-200 CPU能为您的配置提供多少功率（或电流）。如果超出了CPU功率分配，您将不能连接最大数目的模块。参考附录A中CPU和扩充模块的功率要求，参考附录B来计算功率分配。

表 A-5 CPU电源规范

	DC		AC	
<b>输入功率</b>				
输入电压	20.4到28.8 VDC		85到264 VAC（47到63赫兹）	
输入电流	仅24VDC下的CPU	24VDC下的最大负载	仅CPU	最大负载
CPU 221	80毫安	450 mA	在120/240 VAC下为30/15毫安	在120/240 VAC下为120/60毫安
CPU 222	85 mA	500 mA	在120/240 VAC下为40/20毫安	在120/240 VAC下为140/70毫安
CPU 224	110 mA	700 mA	在120/240 VAC下为60/30毫安	在120/240 VAC下为200/100毫安
CPU 226/CPU 226XM	150 mA	1050 mA	在120/240 VAC下为80/40毫安	在120/240 VAC下为320/160毫安
突入电流	在28.8 VDC下为10 A		在264 VAC下为20 A	
绝缘（现场到逻辑电路）	不绝缘		1500 VAC	
保持时间（电源损耗）	在24 VDC下为10毫秒		在120/240 VAC下为20/80毫秒	
保险丝（不可更换）	3 A, 250 V缓慢熔断		2 A, 250 V缓慢熔断	
<b>24 VDC传感器电源</b>				
传感器电压	L+ 减去5 V		20.4到28.8 VDC	
电流限制	峰值电流1.5 A，无破坏性的发热限制（参见表A-3的额定负载）			
电源脉动噪声	源于输入电源		峰间值低于1 V	
绝缘（传感器到逻辑电路）	不绝缘			

表 A-6 CPU输入规范

常规	24 VDC输入
类型	汇点/源点（IEC 类型1 汇点）
额定电压	典型地，4毫安时为24 VDC
允许的最大连续电压	30 VDC
电涌电压	35 VDC持续0.5 s
逻辑1（最小）	在2.5毫安时为15 VDC

表A-6 CPU输入规范 (续)

常规	24 VDC输入	
逻辑0 (最大)	在1毫安时为5 VDC	
输入延迟	可选择 (0.2到12.8毫秒) CPU 226, CPU 226XM: 点I1.6到I2.7具有固定的延迟 (4.5毫秒)	
邻近传感器的两条导线的连接 (Bero) 允许的泄漏电流 (最大值)	1毫安	
绝缘 (现场到逻辑电路) 光学 (电流的) 绝缘群组	是 500 VAC, 1分钟 参见布线图	
高速输入率 (最大) Logic1=15至30 VDC Logic1=15至26 VDC	单相 20 kHz 30 kHz	双相 10 kHz 20 kHz
同时接通输入	全部为55°C	
电缆长度 (最大) 屏蔽 未屏蔽	500米标准输入, 50米HSC输入 300米标准输入	

表 A-7 CPU输出规范

常规	24VDC输出	继电器输出
类型	固态OSFET <sup>1</sup>	干触点
额定电压	24 VDC	24 VDC或250 VAC
电压范围	20.4到28.8 VDC	5至30 VDC或5至250 VAC
电涌电流 (最大值)	8 A, 100毫秒	7 A, 接点闭合
逻辑1 (最小)	20 VDC, 最大电流	-
逻辑0 (最大)	0.1 VDC, 10 K $\Omega$ 负载	-
每点额定电流 (最大值)	0.75 A	2.0 A
每个公共端的额定电流 (最大值)	6 A	10 A
泄漏电流 (最大值)	10 $\mu$ A	-
照明负载 (最大值)	5 W	30 W DC; 200 W AC
感应钳位电压	L+减去48 VDC, 1 W损耗	-
接通电阻 (接点)	0.3 $\Omega$ , 典型	0.2 $\Omega$ (新使用时最大)
绝缘 光学 (电化, 现场到逻辑电路) 逻辑电路到接点 电阻 (逻辑电路到接点) 绝缘群组	500 VAC, 1分钟 - - 参见布线图	- 1500 VAC, 1分钟 100米 $\Omega$ 参见布线图
延迟关到开 / 开到关 (最大值) 转换 (最大值)	2/10微秒 (Q0.0和Q0.1) 15/100微秒 (所有其他情况) -	- 10毫秒
脉冲频率 (最大值) Q0.0和Q0.1	20 kHz	1 赫兹
使用寿命机械周期	-	10,000,000 (无负载)
使用寿命接点	-	100,000 (额定负载)
同时打开输出	全部为55°C	全部为55°C
并行连接两个输出	是	否
电缆长度 (最大值) 屏蔽 未屏蔽	500米 150米	500米 150米

<sup>1</sup> 当机械接点接通S7-200 CPU或任意数字量扩充模块的输出电源时, 它将给数字量输出发送一个约持续50毫秒的“1”信号。对此, 您必须进行规划, 特别是在使用对短持续时间脉冲进行响应的设备时。

### 布线图

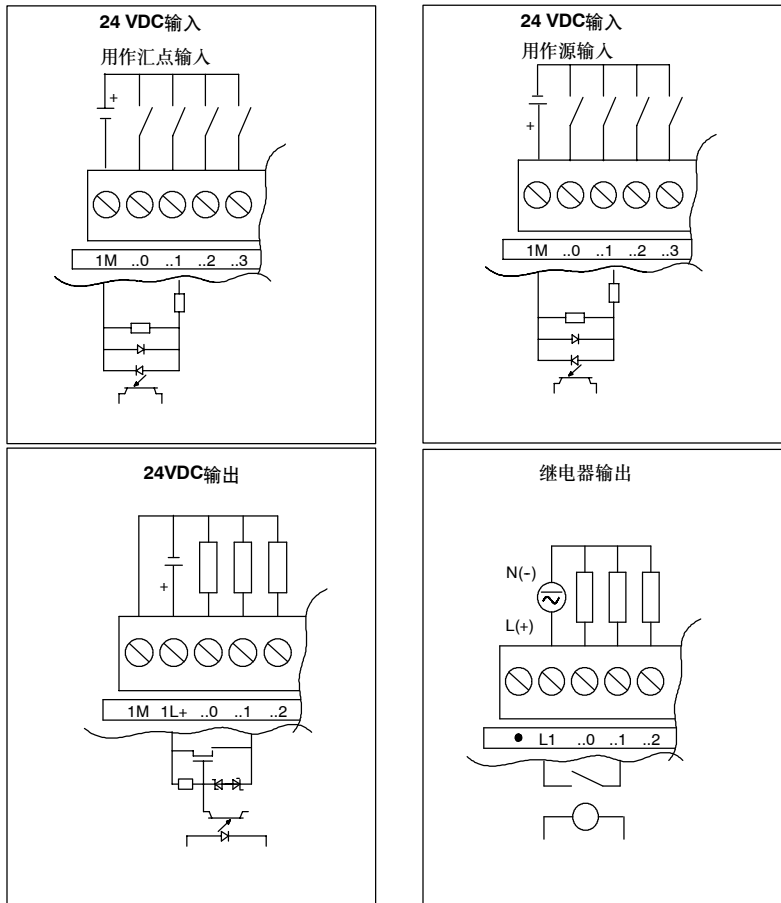


图 A-2 CPU输入和输出

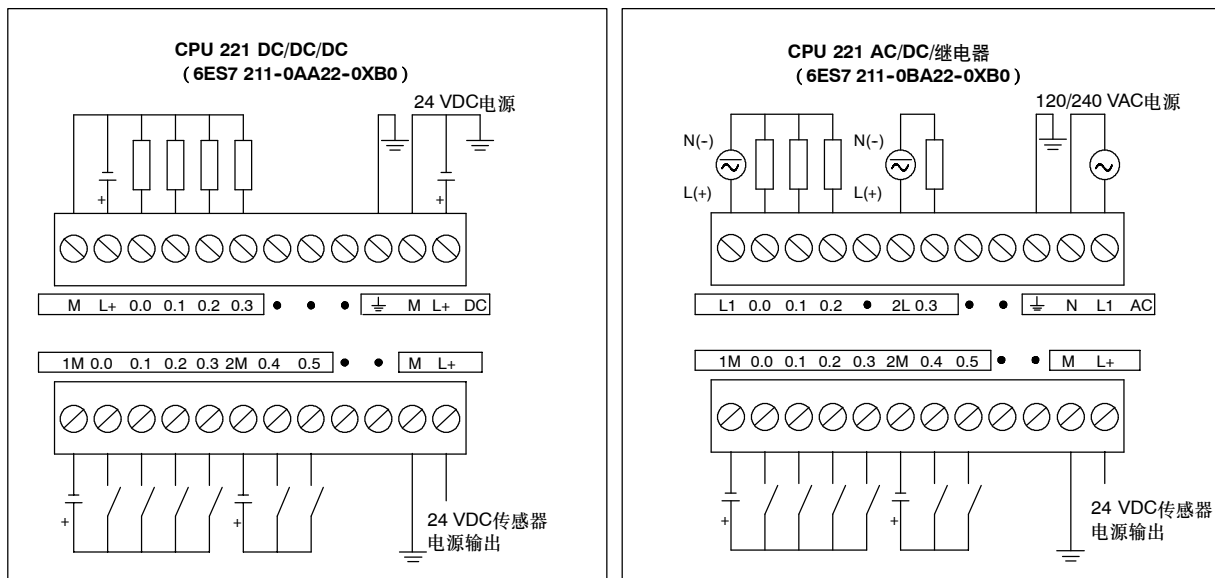
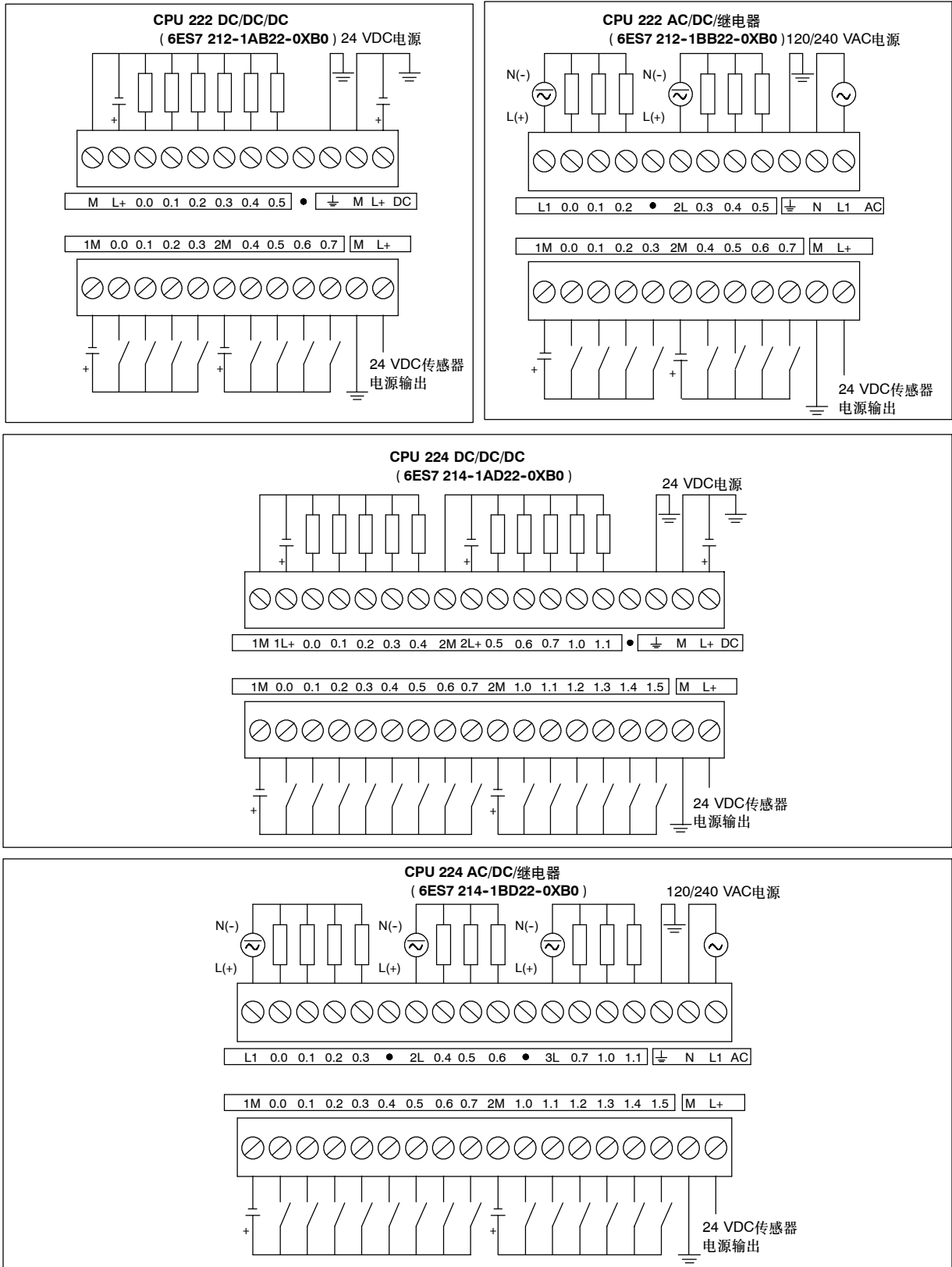


图 A-3 CPU 221布线图

A



A

图 A-4 CPU 222和CPU 224布线图

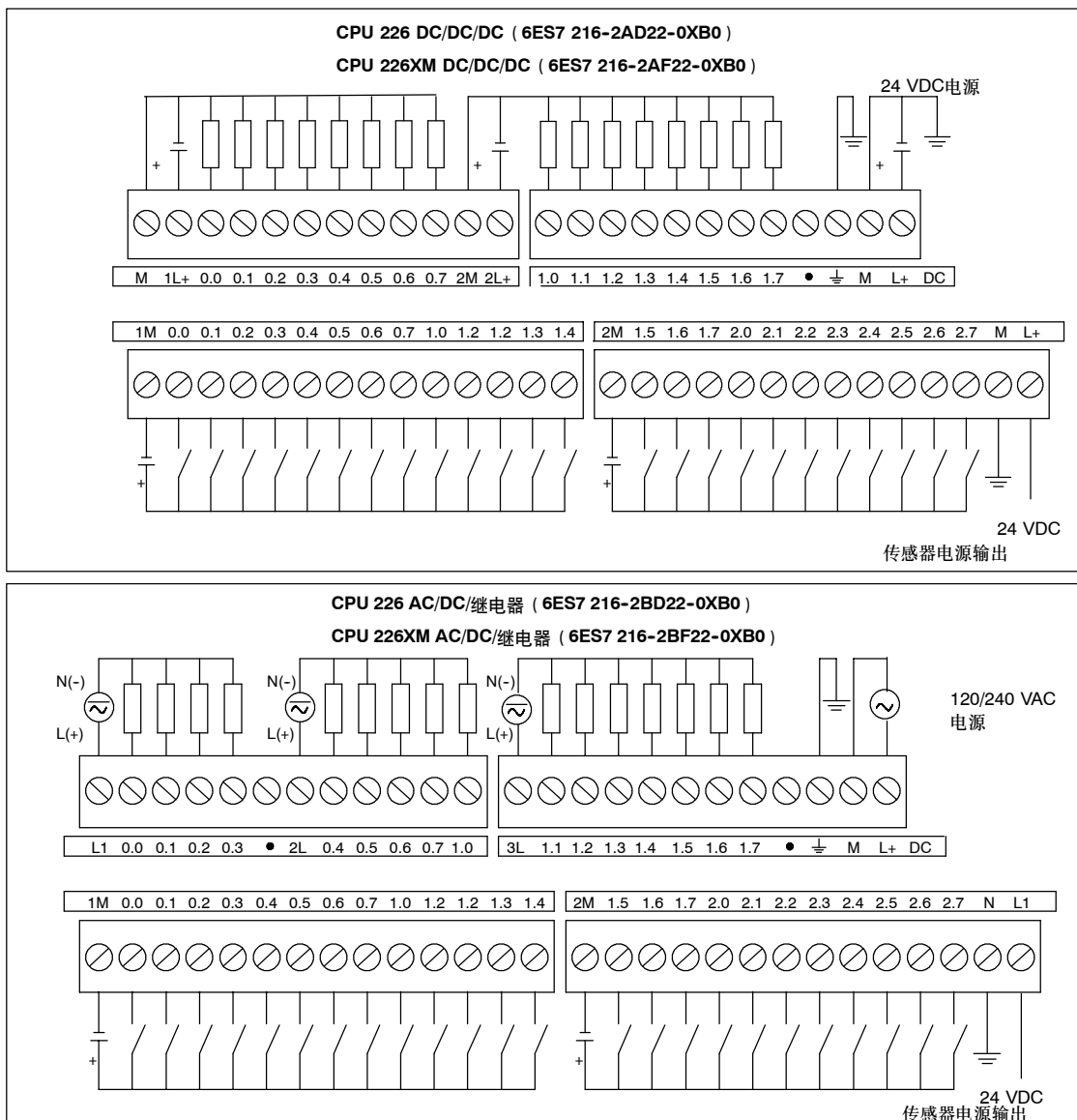


图 A-5 CPU 226和CPU 226XM布线图

表 A-8 S7-200通讯端口的插针分配

连接器	插针号	PROFIBUS信号	端口0/端口1
	1	屏蔽	机壳接地
	2	24 V回流	逻辑中性线
	3	RS-485信号B	RS-485信号B
	4	请求发送	RTS (TTL)
	5	5 V回流	逻辑中性线
	6	+5 V	+5 V、100Ω串联电阻器
	7	+24 V	+24 V
	8	RS-485信号A	RS-485信号A
	9	不适用	10位协议选择 (输入)
	连接器外壳		屏蔽

A



## 数字扩充模块规范

表 A-9 数字扩充模块订购号

订购号	扩展型号	EM输入	EM输出	可移动连接器
6ES7 221-1BF22-0XA0	EM 221 数字输入 8 x 24 VDC	8 x 24 VDC	-	是
6ES7 221-1EF22-0XA0	EM 221 数字输入 8 x AC 120/230 V	8 x AC 120/230V	-	是
6ES7 221-1BH22-0XA0	EM 221 数字输入 16 x 24 VDC	16 x 24 VDC	-	是
6ES7 222-1BD22-0XA0	EM 222 数字输出 4 x 24 VDC-5A	-	4 x 24 VDC-5A	是
6ES7 222-1BF22-0XA0	EM 222 数字输出 8 x 24 VDC	-	8 x 24 VDC-0.75A	是
6ES7 222-1HD22-0XA0	EM 222 数字输出 4 x 继电器-10A	-	4 x 继电器-10A	是
6ES7 222-1HF22-0XA0	EM 222 数字输出 8 x 继电器	-	8 x 继电器-10A	是
6ES7 222-1EF22-0XA0	EM 222 数字输出 8 x AC 120/230 V	-	8 x AC 120/230V	是
6ES7 223-1BF22-0XA0	EM 223 24 VDC 数字组合 4 输入/4 输出	4 x 24 VDC	4 x 24 VDC-0.75A	是
6ES7 223-1HF22-0XA0	EM 223 24 VDC 数字组合 4 输入/4 继电器输出	4 x 24 VDC	4 x 继电器-2A	是
6ES7 223-1BH22-0AX0	EM 223 24 VDC 数字组合 8 输入/8 输出	8 x 24 VDC	8 x 24 VDC-0.75A	是
6ES7 223-1PH22-0XA0	EM 223 24 VDC 数字组合 8 输入/8 继电器输出	8 x 24 VDC	8 x 继电器-2A	是
6ES7 223-1BL22-0XA0	EM 223 24 VDC 数字组合 16 输入/16 输出	16 x 24 VDC	16 x 24 VDC-0.75A	是
6ES7 223-1PL22-0XA0	EM 223 24 VDC 数字组合 16 输入/16 继电器输出	16 x 24 VDC	16 x 继电器-2A	是

表 A-10 数字扩充模块通用规范

订购号	模块名称与说明	尺寸 (毫米) (W x H x D)	重量	损耗	VDC要求	
					+5 VDC	+24 VDC
6ES7 221-1BF22-0XA0	EM 221 DI 8 x 24 VDC	46 x 80 x 62	150 g	2 W	30 mA	-
6ES7 221-1EF22-0XA0	EM 221 DI 8 x AC 120/230 V	71.2 x 80 x 62	160 g	3 W	30 mA	-
6ES7 221-1BH22-0XA0	EM 221 DI 16 x 24 VDC	71.2 x 80 x 62	160 g	3 W	70毫安	-
6ES7 222-1BD22-0XA0	EM 222 DO 4 x 24 VDC-5A	46 x 80 x 62	120 g	3 W	40 mA	-
6ES7 222-1BF22-0XA0	EM 222 DO 8 x 24 VDC	46 x 80 x 62	150 g	2 W	50 mA	-
6ES7 222-1HD22-0XA0	EM 222 DO 4 x 继电器-10A	46 x 80 x 62	150 g	4 W	30 mA	接通: 20毫安/输出, 20.4至28.8 VDC
6ES7 222-1HF22-0XA0	EM 222 DO 8 x 继电器	46 x 80 x 62	170 g	2 W	40 mA	接通: 9毫安/输出, 20.4至28.8 VDC
6ES7 222-1EF22-0XA0	EM 222 DO 8 x AC 120/230 V	71.2 x 80 x 62	165 g	4 W	110 mA	-
6ES7 223-1BF22-0XA0	EM 223 24 VDC 4 进/4 出	46 x 80 x 62	160 g	2 W	40 mA	-
6ES7 223-1HF22-0XA0	EM 223 24 VDC 4 进/4 继电器	46 x 80 x 62	170 g	2 W	40 mA	接通: 9毫安/输出, 20.4至28.8 VDC
6ES7 223-1BH22-0AX0	EM 223 24 VDC 8 进/8 出	71.2 x 80 x 62	200 g	3 W	80毫安	-
6ES7 223-1PH22-0XA0	EM 223 24 VDC 8 进/8 继电器	71.2 x 80 x 62	300 g	3 W	80毫安	接通: 9毫安/输出, 20.4至28.8 VDC
6ES7 223-1BL22-0XA0	EM 223 24 VDC 16 进/16 出	137.3 x 80 x 62	360 g	6 W	160 毫安	-
6ES7 223-1PL22-0XA0	EM 223 24 VDC 16 进/16 继电器	137.3 x 80 x 62	400 g	6 W	150 mA	接通: 9毫安/输出, 20.4至28.8 VDC

表 A-11 数字扩充模块输入规范

常规	24 VDC输入	120/230 VAC 输入 (47至63赫兹)
类型	汇点/源点 (IEC 类型1 汇点)	IEC类型I
额定电压	4 毫安时为24 VDC	6毫安时为120 VAC或额定为9 毫安时的230 VAC
允许的最大持续电压	30 VDC	264 VAC
电涌电压 (最大值)	35 VDC持续0.5 s	-
逻辑1 (最小)	在2.5毫安时为15 VDC	在2.5毫安下为79 VAC
逻辑0 (最大)	在1毫安时为5 VDC	20 VAC或1毫安AC
输入延迟 (最大值)	4.5毫秒	15 毫秒
邻近传感器的两条导线的连接 (Bero) 允许的泄漏电流 (最大值)	1毫安	1毫安AC
绝缘 光学 (电流的, 现场到逻辑 电路) 绝缘组	500 VAC, 1 分钟 参见布线图	1500 VAC, 1 分钟 1点
同时接通输入	全部为55°C	全部为55°C
电缆长度 (最大值) 屏蔽 未屏蔽	500米 300米	500米 300米

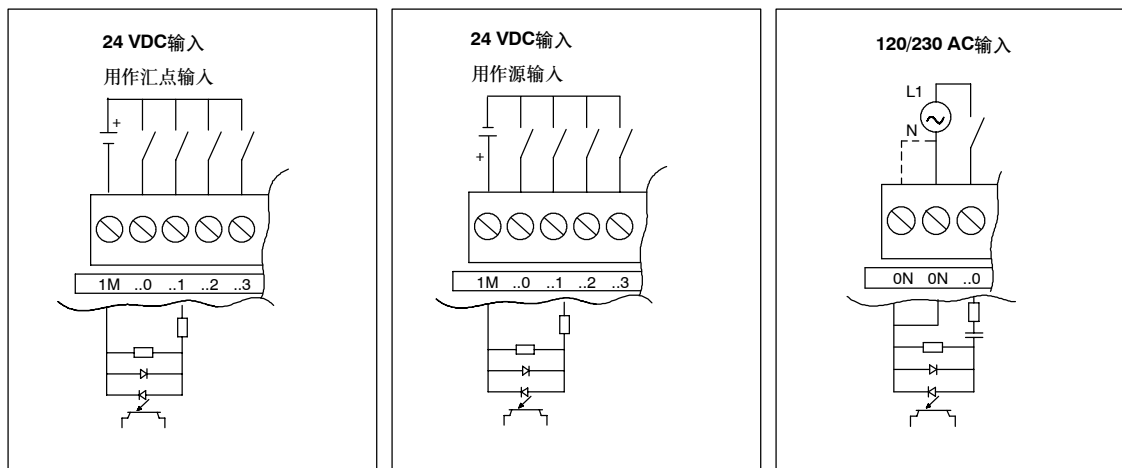


图 A-6 S7-200数字扩充模块输入

表 A-12 数字扩充模块输出规范

常规	24VDC输出		继电器输出		120/230 VAC 输出
	0.75 A	5A	2A	10 A	
类型	固态OSFET <sup>1</sup>		干触点		三端双向可控硅开关, 零交叉接通 <sup>2</sup>
额定电压	24 VDC		24 VDC或250 VAC		120/230 VAC
电压范围	20.4到28.8 VDC		5至30 VDC或 5至250 VAC	12至30 VDC或 12至250 VAC	40到264 VAC (47到63 赫兹)
24 VDC线圈电源电压范围	-		20.4到28.8 VDC		-
电涌电流 (最大值)	8 A, 100毫秒	30 A	7A, 触点关闭	15A, 4s@10% 工作循环	5A, r 毫秒, 2个AC循环
逻辑1 (最小)	20 VDC		-		L1 (-0.9 V RMS)
逻辑0 (最大)	0.1 VDC	0.2 VDC	-		-
每点额定电流 (最大值)	0.75 A	5A	2.00 A	10 A 阻抗性; 2 A DC 感性性; 3 A AC 感性性	0.5 A AC <sup>3</sup>
每个公共端的额定电流 (最大值)	6 A	5A	8A	10 A	0.5 A AC
泄漏电流 (最大值)	10 $\mu$ A	30 $\mu$ A	-		132 VAC时为1.1毫安 rms以及264 VAC时为 1.8 毫安rms
照明负载 (最大值)	5 W	50 W	30 W DC/ 200 W AC	100 W DC/ 1000 W AC	60 W
感应错位电压	L+ - 48 V	L+ - 47 V <sup>4</sup>	-		-
接通电阻 (触点)	0.3 $\Omega$ , 典型地	0.05 $\Omega$ 最大	0.2 $\Omega$ 新的时最大值	0.1 $\Omega$ 新的时 最大值	410 $\Omega$ 负荷电流低于 0.05 A时的最大值
绝缘					
光学 (电化, 现场到逻辑电路)	500 VAC, 1分钟		-		1500 VAC, 1分钟
线圈到逻辑电路	-		无		-
线圈到触点	-		1500 VAC, 1分钟		-
电阻 (线圈到触点)	-		100米 $\Omega$ 新的时最小值		-
绝缘群组	参见布线图		参见布线图		1点
延迟 Off到On/On到Off (最大值)	50 $\mu$ s / 200 $\mu$ s	500 $\mu$ s	-	-	0.2 毫秒 + 1/2 AC周期
转换 (最大值)	-	-	10毫秒	15 毫秒	-
转换频率 (最大值)	-		1 赫兹		10 赫兹
使用寿命机械周期	-		10,000,000 (无负载)	30,000,000 (无负载)	-
使用寿命触点	-		100,000 (额定负载)	30,000 (额定负载)	-
同时输出on	全部在55°C		全部在55°C	全部在55 °C, 最大值20A, 模块电流 <sup>5</sup> 全部在40 °C, 10 A / 点	全部在55°C
并行连接两个输出	是		否		否
电缆长度 (最大)					
屏蔽	500米		500米		500米
未屏蔽	150米		150米		150米

- 当机械触点接通S7-200 CPU或任意数字量扩充模块的输出电源时, 它将给数字量输出发送一个约持续50毫秒的“1”信号。对此, 您必须进行规划, 特别是在使用对短持续时间脉冲进行响应的设备时。
- 当机械触点接通AC扩充模块的输出电源时, 它将给AC输出发送一个约1/2 AC周期的“1”信号。对此, 您必须加以周到考虑。
- 由于零交叉电路系统的缘故, 负荷电流必须为全波AC, 而不能为半波AC。最小负荷电流为0.05A AC。负荷电流在5毫安和50毫安AC之间, 电流可以控制, 但由于有410欧姆的一系列电阻, 将存在一个额外的电压降。
- 如果由于感应转换次数过多或出现异常情况而使输出过热, 则输出点可能关闭或被损坏。如果在切断感应负载时, 输出端产生大于0.7J的能量, 则输出端可能过热或被损坏。为消除该限制, 可将如第3章所述的抑制电路与负载一起并行加入。这些元件需要根据特定应用适当地按大小排列。
- EM 222 DO 4 x 继电器具有不同于S7-200其余部分的FM额定值。该模块具有一个T4额定值, 以代替FM类I, A、B、C和D区组危险位置的T4A。

A

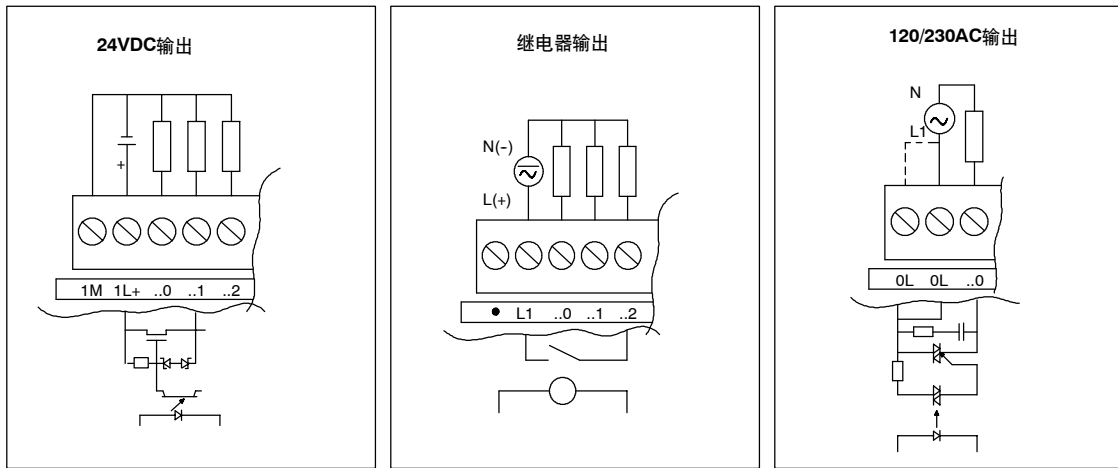


图 A-7 S7-200数字扩充模块输出

布线图

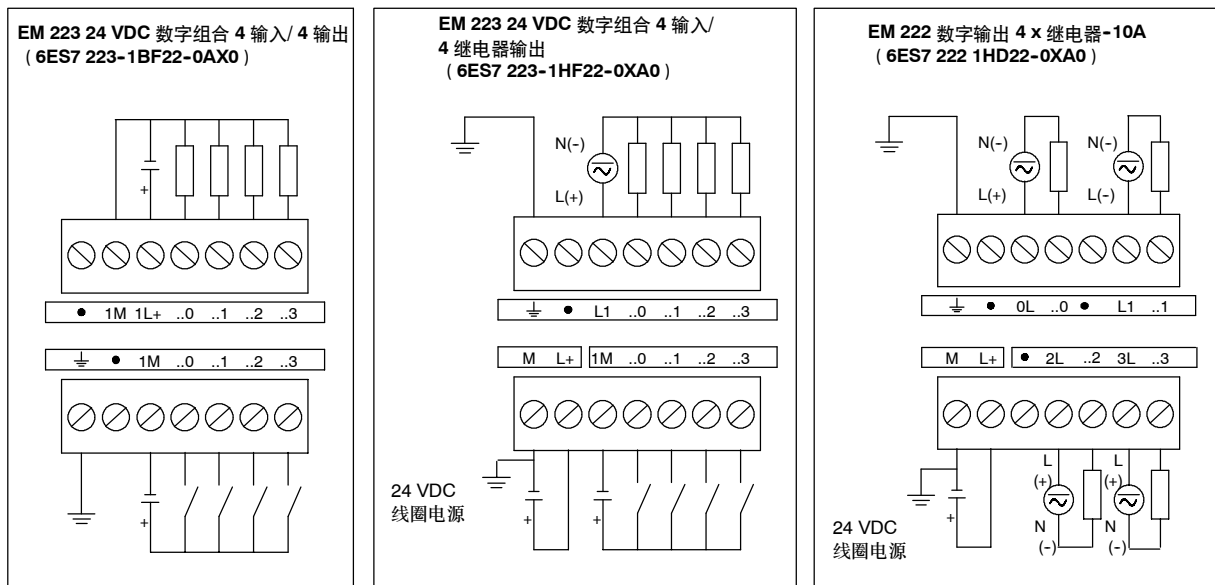


图 A-8 EM 222和EM223扩充模块的布线图

A

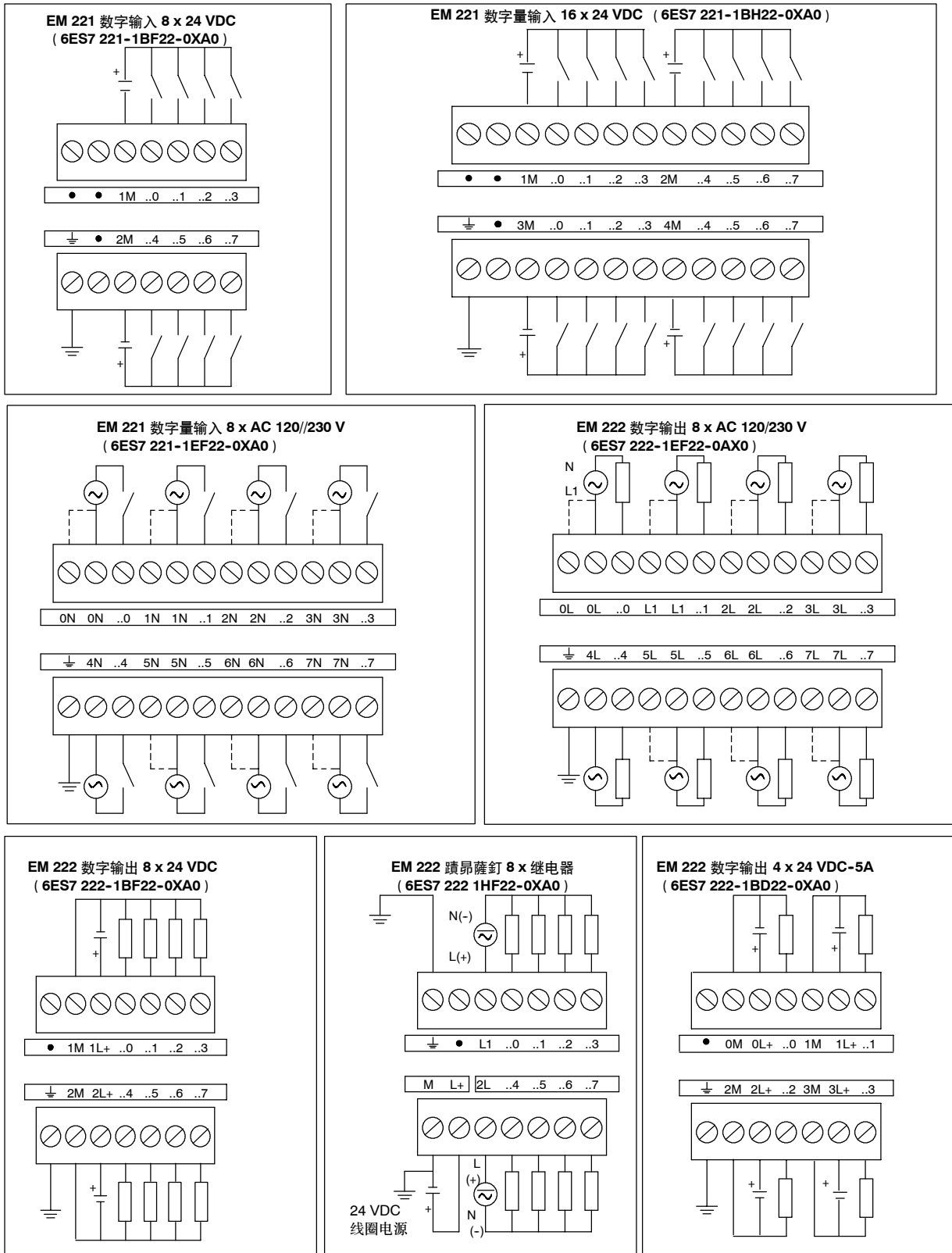


图 A-9 EM 221和EM 222扩充模块的布线图

A

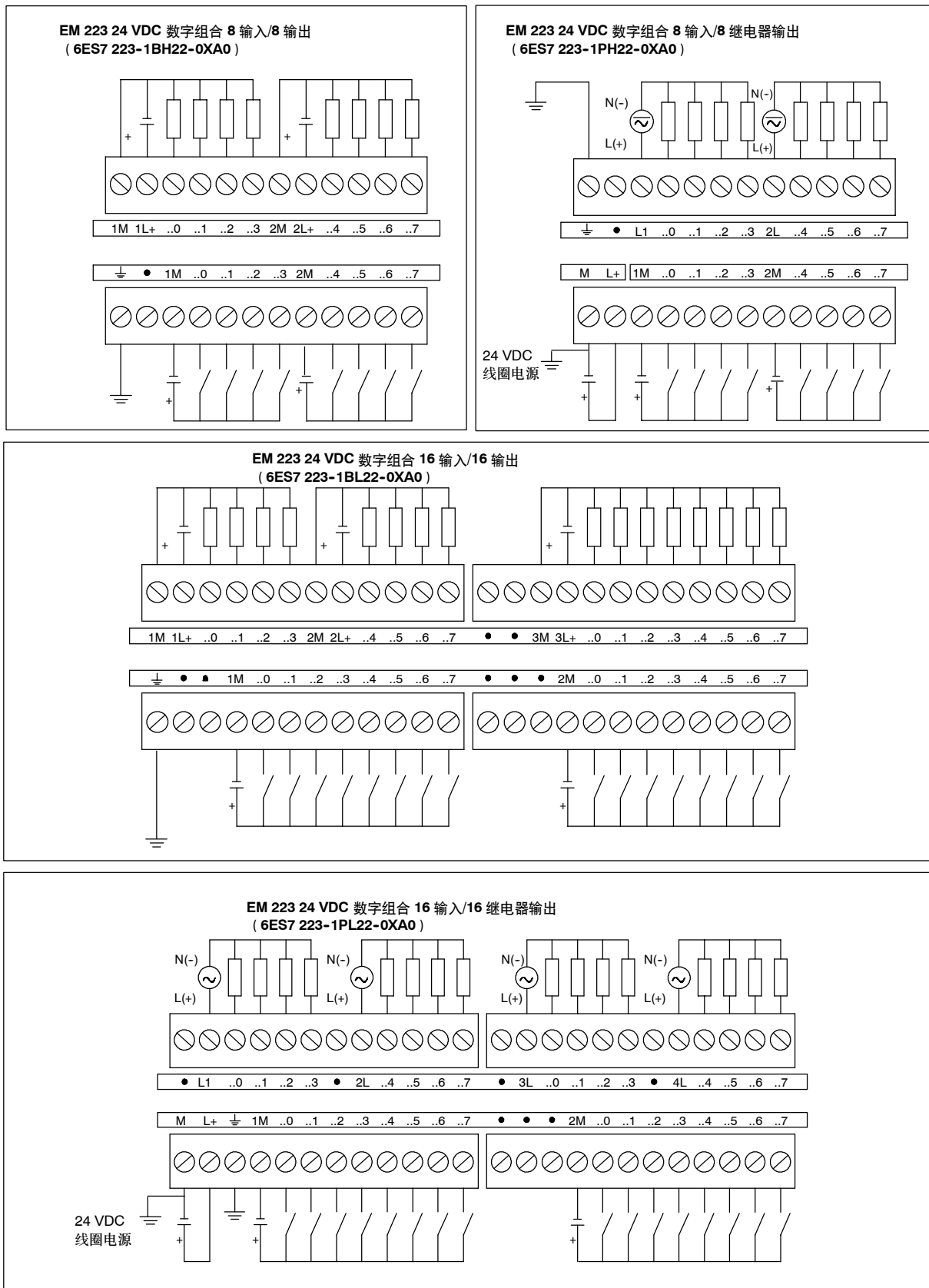


图 A-10 EM223扩充模块的布线图

A

## 模拟扩充模块规范

表 A-13 模拟扩充模块订购号

订购号	扩展型号	EM输入	EM输出	可移动连接器
6ES7 231-0HC22-0XA0	EM 231 模拟输入, 4输入	4	-	否
6ES7 232-0HB22-0XA0	EM 232 模拟输出, 2 输出	-	2	否
6ES7 235-0KD22-0XA0	EM 235 模拟组合 4 输入/1 输出	4	1 <sup>1</sup>	否

1 CPU将为该模块保留个2个模拟输出点。

表 A-14 模拟扩充模块通用规范

订购号	模块名称与说明	尺寸 (毫米) (W x H x D)	重量	损耗	VDC要求	
					+5 VDC	+24 VDC
6ES7 231-0HC22-0XA0	EM 231模拟量输入, 4输入	71.2 x 80 x 62	183 g	2 W	20 毫安	60毫安
6ES7 232-0HB22-0XA0	EM 232 模拟输出, 2输出	46 x 80 x 62	148 g	2 W	20 毫安	70毫安 (具有20毫安下的两个输出)
6ES7 235-0KD22-0XA0	EM 235模拟组合 4输入/1输出	71.2 x 80 x 62	186 g	2 W	30 mA	60 毫安 (具有20毫安下的输出)

表 A-15 模拟扩充模块输入规范

常规	6ES7 231-0HC22-0XA0	6ES7 235-0KD22-0XA0
数据字格式	(参见图A-13)	(参见图A-13)
双极、全量程范围	-32000至+32000	-32000至+32000
单极、全量程范围	0至32000	0至32000
DC输入阻抗	≥10 MΩ 电压输入, 250 Ω 电流输入	≥ 10 MΩ 电压输入, 250 Ω 电流输入
输入滤波器衰减	在3.1 kHz下为-3db	在3.1 kHz下为-3db
最大输入电压	30 VDC	30 VDC
最大输入电流	32 毫安	32 毫安
分辨率	12位A/D转换器	12位A/D转换器
绝缘 (现场到逻辑电路)	无	无
输入类型	微分	微分
输入范围		
电压 (单极)	0至10 V, 0至5 V	0至10 V, 0至5 V 0至1 V, 0至500 mV, 0至100 mV, 0至50 mV
电压 (双极)	±5 V, ±2.5 V	±10 V, ±5V, ±2.5 V, ±1 V, ±500 mV, ±250 mV, ±100 mV, ±50 mV, ±25 mV
电流	0至20 毫安	0至20 毫安
输入分辨率	参见表A-18	参见表A-19
电压 (单极)		
电压 (双极)		
电流		
模拟到数字的转换时间	< 250 μs	< 250 μs
模拟输入阶跃响应	1.5 毫秒至95%	1.5 毫秒至95%
共模抑制	40 dB, DC至60 赫兹	40 dB, DC至60 赫兹
共模电压	信号电压加上共模电压必须为 ≤ ±12V	信号电压加上共模电压必须为 ≤ ±12V
24 VDC电源电压范围	20.4至28.8	20.4至28.8

表 A-16 模拟扩充模块输出规范

常规	6ES7 232-0HB22-0XA0	6ES7 235-0KD22-0XA0
绝缘（现场到逻辑电路）	无	无
信号范围 电压输出 电流输出	$\pm 10\text{ V}$ 0至20 毫安	$\pm 10\text{ V}$ 0至20 毫安
分辨率, 全量程 电压 电流	12 位 11 位	12 位 11 位
数据字格式 电压 电流	-32000至+32000 0至+32000	-32000至+32000 0至+32000
精度 最坏情况, 0°至55° C 电压输出 电流输出 典型地, 25° C 电压输出 电流输出	$\pm$ 满量程的2% $\pm$ 满量程的2% $\pm$ 满量程的0.5% $\pm$ 满量程的0.5%	$\pm$ 满量程的2% $\pm$ 满量程的2% $\pm$ 满量程的0.5% $\pm$ 满量程的0.5%
设置时间 电压输出 电流输出	100 $\mu\text{S}$ 2 毫秒	100 $\mu\text{S}$ 2 毫秒
最大驱动 电压输出 电流输出	5000 $\Omega$ 最小 500 $\Omega$ 最大	5000 $\Omega$ 最小 500 $\Omega$ 最大

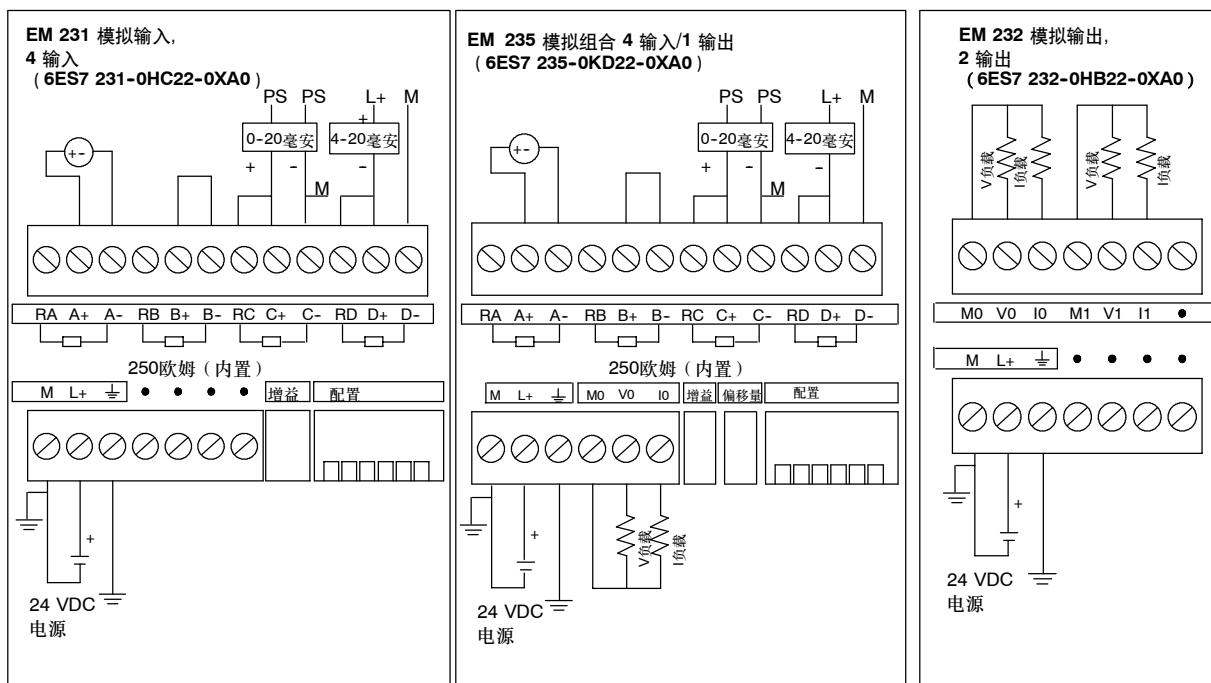


图 A-11 模拟扩充模块的布线图

A



## 模拟LED指示器

模拟模块的LED指示器均显示在表A-17中。

表 A-17 模拟LED指示器

LED指示器	ON	OFF
24VDC电源正常	无故障	无24VDC电源



### 提示

用户电源的状态也将用特殊内存（SM）位进行报告。更详细信息，请参见附录D中关于SMB8至SMB21 I/O模块ID和错误寄存器的说明。

## 输入校准

校准调整将影响接在模拟多路转接器后面的仪器的放大级（参见图A-14中EM 231的输入方框图和图A-15中的EM 235）。因此，校准将影响所有的用户输入通道。即使在校准完成之后，在模拟多路转接器前面的每个输入电路中，由于各个元件值的偏差，也将使连接到同一输入信号的各个通道的读数产生细小的差别。

为符合上述的各种规范，您应启用用于模块所有输入的模拟输入过滤器。请选择64个或更多的范例来计算平均值。

为校准输入，请使用下列步骤。

1. 关闭模块电源。选择所期望的输入范围。
2. 接通CPU和模块的电源。让模块稳定15分钟。
3. 使用一个发送器、一个电压源或一个电流源，将零值信号加到某个输入端子上。
4. 将所报告的数值通过适当的输入通道读到CPU中。
5. 调整偏移量电位计，直到读数为零或为所期望的数字式数据值。
6. 将满量程数值信号连接到某个输入端子上。将所报告的数值读到CPU。
7. 调整增益电位计，直到读数为32000或为所期望的数字式数据值。
8. 如果需要，可重复上述的偏移量和增益校准过程。

## EM 231和EM 235的校准和配置位置

图A-12显示了位于模块底部接线盒右侧的校准电位计和配置DIP开关。

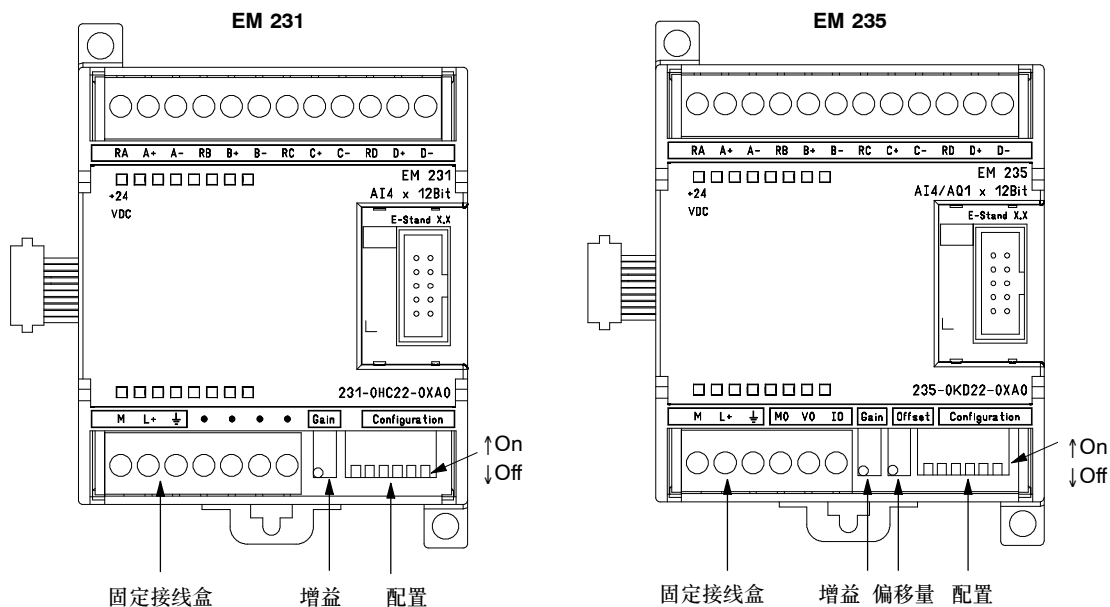


图 A-12 EM 231和EM 235的校准电位计与配置DIP开关的位置

### EM 231的配置

表A-18显示了如何使用配置DIP开关对EM 231进行配置。使用开关1、2、3可选择模拟输入范围。所有输入均可设置为同样的模拟量输入范围。在本表中，ON表示闭合，OFF表示打开。开关设置只有在电源接通时才能读取。

表 A-18 用于选择模拟输入范围的EM 231配置开关表

单极			满量程输入	分辨率
SW1	SW2	SW3		
ON	OFF	ON	0至10 V	2.5 mV
	ON	OFF	0至5V	1.25 mV
			0至20毫安	5 μA
双极			满量程输入	分辨率
SW1	SW2	SW3		
OFF	OFF	ON	±5 V	2.5 mV
	ON	OFF	± 2.5 V	1.25 mV

## EM 235的配置

表A-19显示了如何使用配置DIP开关对EM 235进行配置。使用开关1至6可选择模拟输入范围和分辨率。所有输入均可设置为同样的模拟输入范围和格式。表A-20显示了如何选择单极/双极（开关6）、增益（开关4和5）、衰减（开关1、2、3）。在这些表中，ON表示闭合，OFF表示打开。这些开关设置只有在电源接通时才能读取。

表 A-19 用于选择模拟范围和分辨率的EM 235配置开关表

单极						满量程输入	分辨率
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	ON	0至50 mV	12.5 $\mu$ V
OFF	ON	OFF	ON	OFF	ON	0至100 mV	25 $\mu$ V
ON	OFF	OFF	OFF	ON	ON	0至500 mV	125 $\mu$ V
OFF	ON	OFF	OFF	ON	ON	0至1 V	250 $\mu$ V
ON	OFF	OFF	OFF	OFF	ON	0至5V	1.25 mV
ON	OFF	OFF	OFF	OFF	ON	0至20 毫安	5 $\mu$ A
OFF	ON	OFF	OFF	OFF	ON	0至10 V	2.5 mV
双极						满量程输入	分辨率
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	OFF	+25 mV	12.5 $\mu$ V
OFF	ON	OFF	ON	OFF	OFF	+50 mV	25 $\mu$ V
OFF	OFF	ON	ON	OFF	OFF	+100 mV	50 $\mu$ V
ON	OFF	OFF	OFF	ON	OFF	+250 mV	125 $\mu$ V
OFF	ON	OFF	OFF	ON	OFF	+500 mV	250 $\mu$ V
OFF	OFF	ON	OFF	ON	OFF	+1 V	500 $\mu$ V
ON	OFF	OFF	OFF	OFF	OFF	+2.5 V	1.25 mV
OFF	ON	OFF	OFF	OFF	OFF	+5 V	2.5 mV
OFF	OFF	ON	OFF	OFF	OFF	+10 V	5 mV

表 A-20 用于选择单极/双极、增益、衰减的EM 235配置开关表

EM 235配置开关						单极/双极选择	增益选择	衰减选择
SW1	SW2	SW3	SW4	SW5	SW6			
					ON	单极		
					OFF	双极		
			OFF	OFF			x1	
			OFF	ON			x10	
			ON	OFF			x100	
			ON	ON			无效的	
ON	OFF	OFF						0.8
OFF	ON	OFF						0.4
OFF	OFF	ON						0.2

### EM 231和EM 235的输入数据字格式

图A-13显示了12位数据值在CPU模拟输入字中的存放位置。

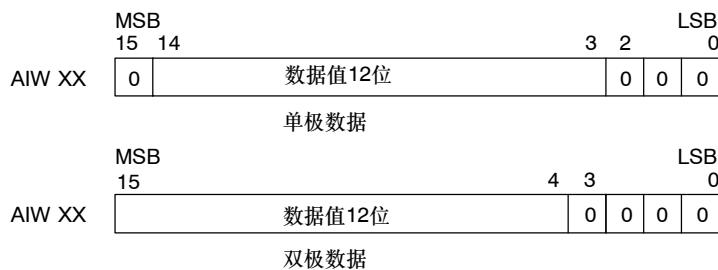


图 A-13 EM 231和EM 235的输入数据字格式



**提示**

模拟到数字转换器（ADC）读数的12个位以数据字格式左对齐。MSB是符号位：零表示正的数据字数值。在单极的格式中，拖后的三个零将使数据字在ADC数值的计数每变化一次时，改变八次。在双极的格式中，拖后的四个零将使数据字的值在ADC数值的计数每变化一次时，改变16次。

### EM 231和EM 235的输入方框图

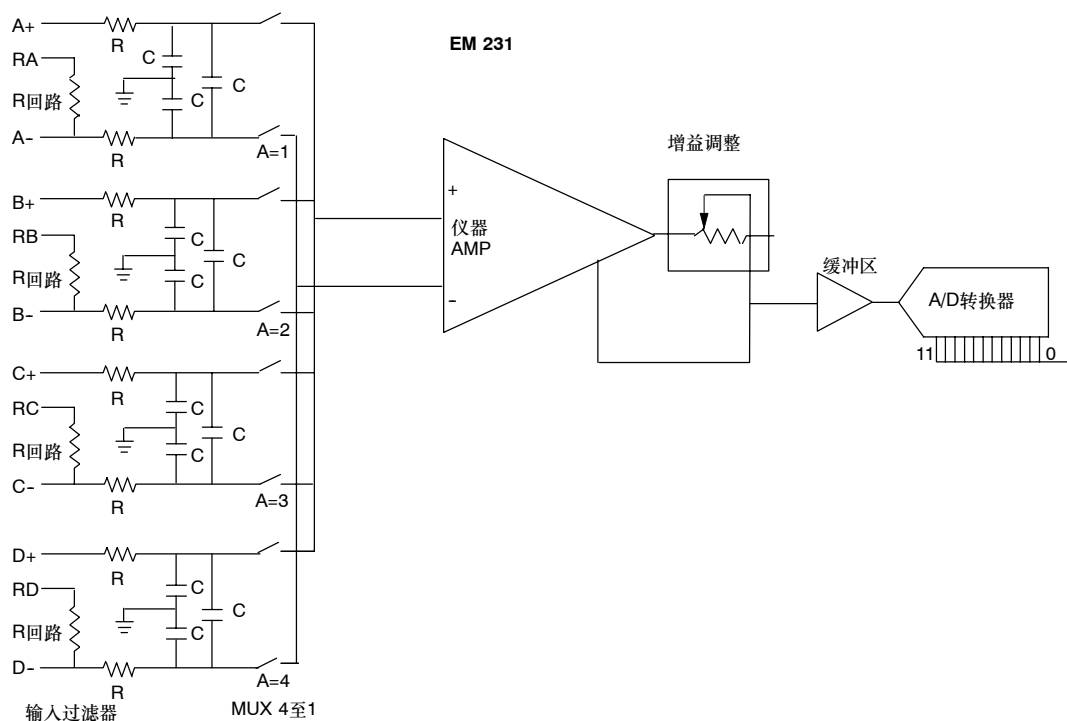


图 A-14 EM 231的输入方框图

A

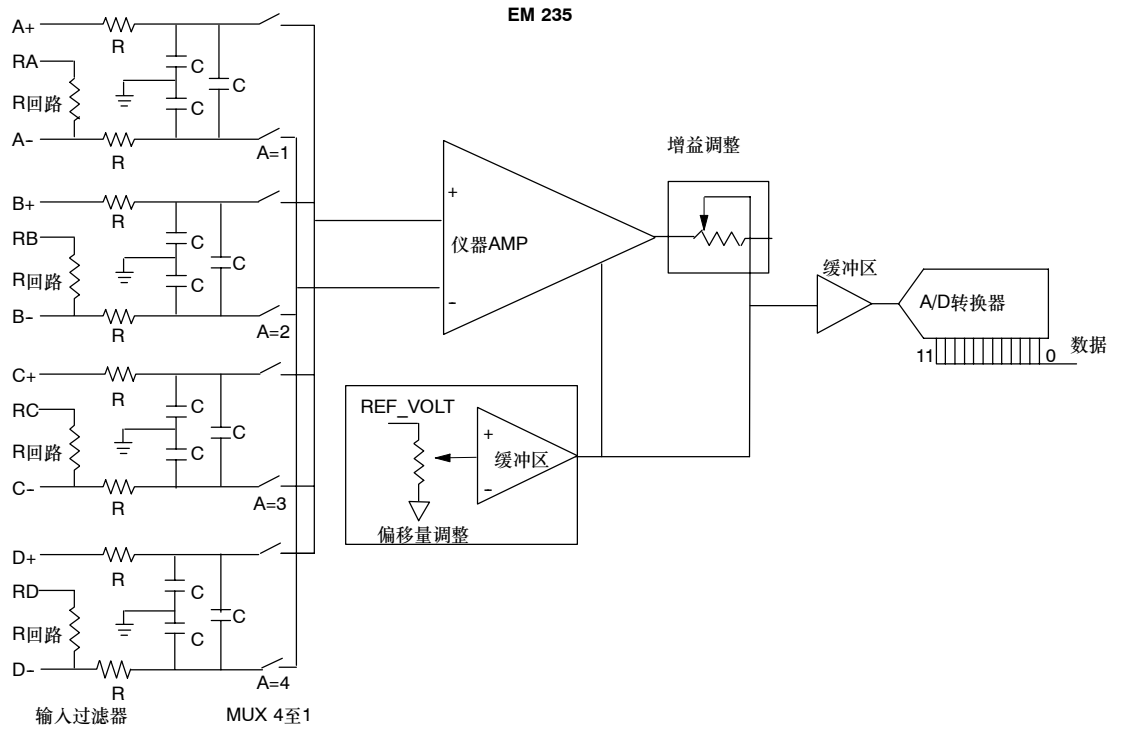


图 A-15 EM 235的输入方框图

**EM 232和EM 235的输出数据字格式**

图A-16显示了12位数据数值在CPU模拟输出字内的存放位置。

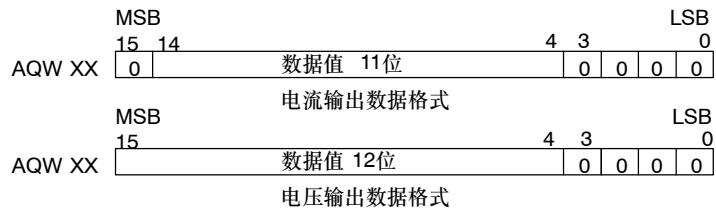


图 A-16 EM 232和EM 235的输出数据字格式



**提示**

数字到模拟转换器 (DAC) 读数的12个位以输出数据字格式左对齐。MSB是符号位：零表示正的数据数值。在将数据装载到DAC寄存器之前，四个拖后的零将被截去。这些位对输出信号数值没有任何影响。



## EM 232和EM 235的输出方框图

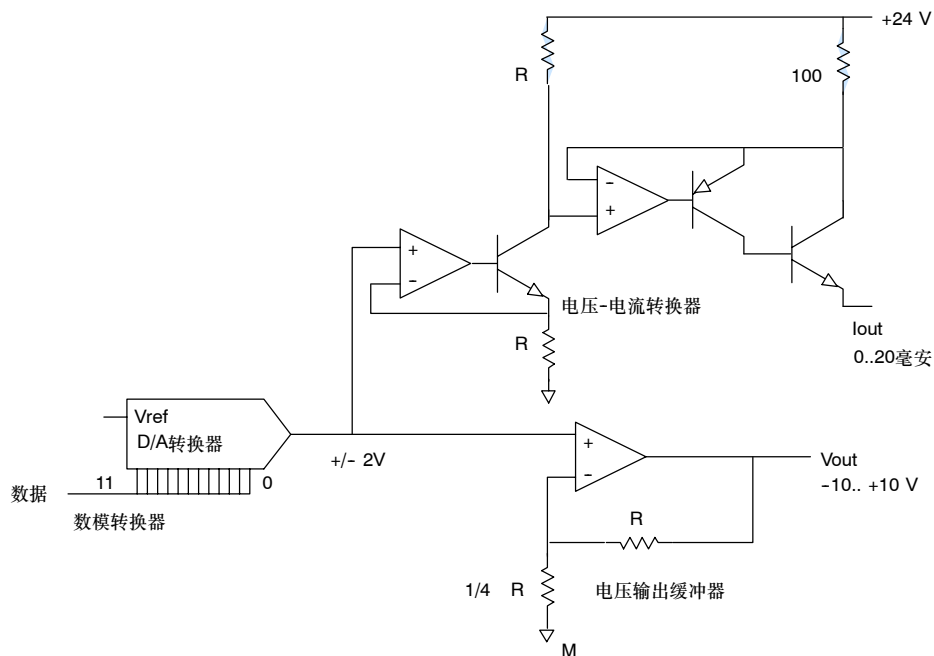


图 A-17 EM 232和EM 235的输出方框图

## 安装指南

使用下列指南可确保安装的准确性和可重复性:

- 确保24DC传感器电源工作稳定、没有噪声。
- 尽可能使用最短的传感器导线。
- 传感器导线应使用带屏蔽的双绞线。
- 屏蔽只应端接在传感器处。
- 未使用的通道的输入应短路，见图A-11所示。
- 应避免将导线弯曲成锐角。
- 布线时请使用电缆槽。
- 应避免将信号线与高能导线平行布置。如果两种导线必须相交，则相交的角度必须正确。
- 应通过绝缘输入信号或将其连接到模拟模块的外部24V公共端，确保输入信号位于共模电压规范之内。



## 提示

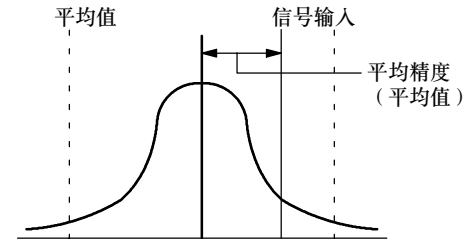
建议不要将EM 231和EM 235扩充模块和热电偶一起使用。

### 理解模拟输入模块：精度与可重复性

EM 231和EM 235模拟输入模块均是低成本的高速12位模拟输入模块。这些模块可在149微秒之内将模拟信号输入转换为其相应的数字值。每当用户程序存取模拟点时，模拟信号输入都将进行转换。必须将这些转换时间加到用于存取模拟量输入的指令基本执行时间中。

EM 231和EM 235将提供一个未经处理的数字值（未线性化或过滤），该数值对应于模块输入端子所显示的模拟电压或电流。由于这些模块均为高速模块，因此，它们均可跟随模拟输入信号的快速变化（包括内部和外部的噪声）。

对于恒定不变或缓慢变化的模拟输入信号，由噪声所引起的读数之间差异，通过对这一系列的读数值取平均值，可使这些差异最小。请注意，增加在计算平均值中使用的读数次数，将导致相对降低对输入信号变化的响应速度。



可重复性的限制  
(所有读数的99%均应落在这些限制之内)

图 A-18 精度定义

图A-18以图表的形式显示了99%可重复性限制值、各个读数的平均数或平均值以及平均精度。

用于可重复性的规范描述了输入信号保持不变时的模块读数之间的差异。可重复性规范定义了99%的读数将处于的限制范围。可重复性如本图中的钟形曲线所示。

平均精度规范描述了误差（各个读数的平均值与实际模拟输入信号的精确值之间的差异）的平均值。

表A-21给出了与各个可配置范围有关的可重复性规范与平均精度。

## 模拟规范的定义

- 精度: 与给定点上的期望值之间的偏差
- 分辨率: 反映到输出上的LSB变化的影响。

表 A-21 EM 231和EM 235规范

满量程输入范围	可重复性 <sup>1</sup>		平均(平均值)精度 <sup>1, 2, 3, 4</sup>	
	满量程的 %	计数	满量程的 %	计数
<b>EM 231规范</b>				
0至5V	± 0.075%	±24	± 0.1%	±32
0至20 毫安				
0至10 V		± 48	± 0.05%	
± 2.5 V				
± 5 V				
<b>EM 235规范</b>				
0至50 mV	± 0.075%	± 24	± 0.25%	± 80
0至100 mV			± 0.2%	± 64
0至500 mV			± 0.05%	± 16
0至1 V				
0至5V				
0至20 毫安				
0至10 V	± 0.075%	± 48	± 0.25%	± 160
± 25 mV			± 0.2%	± 128
± 50 mV			± 0.1%	± 64
± 100 mV			± 0.05%	± 32
± 250 mV				
± 500 mV				
± 1 V				
± 2.5 V				
± 5 V				
± 10 V				

<sup>1</sup> 在所选输入范围校准后进行测量。

<sup>2</sup> 不更正接近零模拟输入的信号偏移误差, 该误差不包含在精度规范内。

<sup>3</sup> 由于有限的模拟多路转换器的设置时间, 通道与通道之间将存在传递转换误差。最大传递误差是通道之间差值的0.1%。

<sup>4</sup> 平均精度包含了非线性影响以及温度从0漂移到55摄氏度的影响。



## 热电偶与RTD扩充模块规范

表 A-22 热电偶与RTD模块订购号

订购号	扩展型号	EM输入	EM输出	可移动连接器
6ES7 231-7PD22-0XA0	EM 231模拟量输入热电偶, 4输入	4热电偶	-	否
6ES7 231-7PB22-0XA0	EM 231模拟输入RTD, 2输入	2 RTD	-	否

表 A-23 热电偶与RTD模块通用规范

订购号	模块名称与说明	尺寸 (毫米) (W x H x D)	重量	损耗	VDC要求	
					+5 VDC	+24 VDC
6ES7 231-7PD22-0XA0	EM 231模拟量输入热电偶, 4输入	71.2 x 80 x 62	210 g	1.8 W	87毫安	60毫安
6ES7 231-7PB22-0XA0	EM 231模拟输入RTD, 2输入	71.2 x 80 x 62	210 g	1.8 W	87毫安	60毫安

表 A-24 热电偶和RTD模块规范

常规	6ES7 231-7PD22-0XA0 热电偶	6ES7 231-7PB22-0XA0 RTD
绝缘 现场到逻辑电路 现场到24 VDC 24 VDC到逻辑电路	500 VAC 500 VAC 500 VAC	500 VAC 500 VAC 500 VAC
共模输入范围 (输入通道到输入通道)	120 VAC	0
共模抑制	> 120 dB, 120 VAC时	> 120 dB, 120 VAC时
输入类型	未接地TC	模块接地基准RTD
输入范围 <sup>1</sup>	TC类型 (每个模块选择一种) S、T、R、E、N、K、J 电压范围: +/- 80 mV	RTD类型 (每个模块选择一种): Pt -100Ω, 200Ω, 500Ω, 1000Ω (α = 3850 ppm, 3920 ppm, 3850.55 ppm, 3916 ppm, 3902 ppm)  Pt -10000Ω (α = 3850 ppm) Cu -9.035Ω (α = 4720 ppm) Ni -10Ω, 120Ω, 1000Ω (α = 6720 ppm, 6178 ppm) R -150Ω, 300Ω, 600Ω FS
输入分辨率 温度 电压 电阻	0.1° C / 0.1° F 15位加符号 -	0.1° C / 0.1° F - 15位加符号
测量原理	西格马 德耳塔	西格马 德耳塔
模块更新时间: 所有通道	405毫秒	405 毫秒 (Pt10000时为700 毫秒)
导线长度	到传感器的最大长度为100米	到传感器的最大长度为100米
导线环路电阻	最大100Ω	20Ω, 2.7Ω, 对于Cu的最大值
干扰的抑制	85 dB, 50 赫兹/60 赫兹/ 400 赫兹时	85 dB, 50 赫兹/60 赫兹/ 400 赫兹时
数据字格式	电压: -27648至+ 27648	电阻: -27648至+27648
最大传感器损耗	-	1m W
输入阻抗	≥1MΩ	≥ 10MΩ
最大输入电压	30 VDC	30 VDC (传感器), 5 VDC (源)
输入过滤器衰减	-3 db, 21 kHz	-3 db, 3.6 kHz
基本误差	0.1% FS (电压)	0.1% FS (电阻)
可重复性	0.05% FS	0.05% FS
冷接点误差	±1.5 ° C	-
24 VDC电源电压范围	20.4到28.8 VDC	20.4到28.8 VDC

<sup>1</sup> 输入范围选项 (温度、电阻上的压降) 适用于模块上的所有通道。

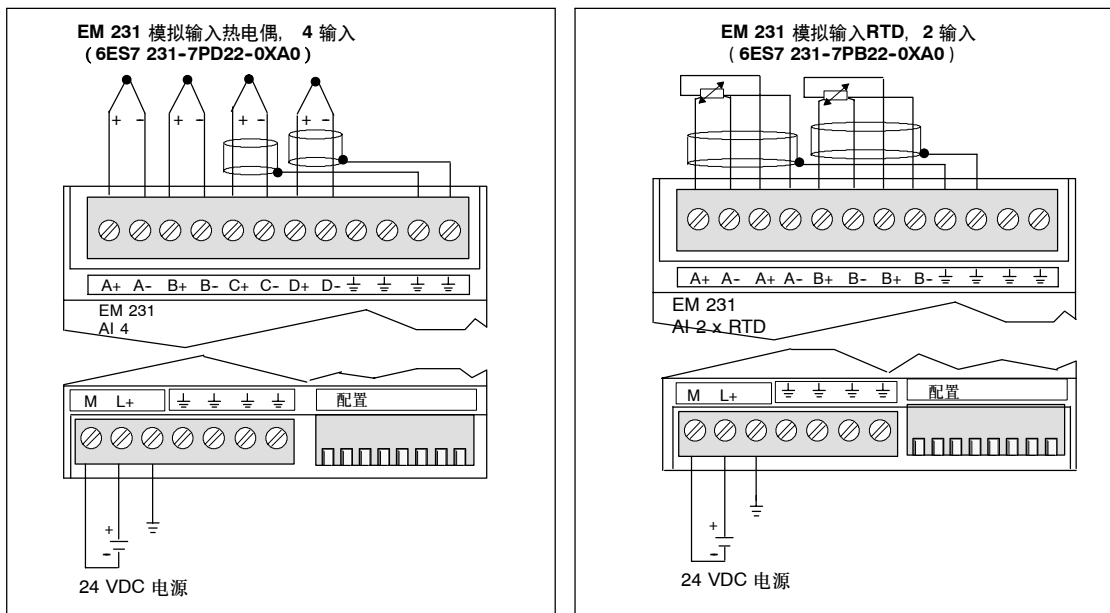


图 A-19 EM 231 热电偶与 EM 231 RTD 模块的连接器端子标识

### 兼容性

RTD与热电偶模块设计成与CPU 222、CPU 224、CPU 226和CPU 226XM一起工作。



### 提示

RTD与热电偶模块均设计成安装在稳定温度环境下时具有最佳的性能。

例如，EM 231热电偶模块具有专门的冷接点补偿电路，该电路用于对模块连接器处的温度进行测量，并对测量值进行必要的修改，以补偿基准温度与模块温度之间的温度差。如果安装有EM 231热电偶模块的区域，其周围温度变化很快，则将引起附加的误差。

为达到最大的精度和可重复性，西门子建议将S7-200 RTD与热电偶模块安装在具有稳定环境温度的位置上。

### 抗扰性

使用带有屏蔽的导线，以获得最佳的抗扰性。如果不使用热电偶输入通道，则应将未使用的通道输入短路，或将其并联到另一个通道。

A

## EM 231热电偶模块

EM 231热电偶模块为S7-200系列提供了一种方便的、单独的接口，用于连接七种热电偶类型：J、K、E、N、S、T和R。它允许S7-200连接到低电位的模拟信号，其范围为 $\pm 80$  mV。该模块所附带的所有热电偶必须具有同样的类型。

### 热电偶基础

只要将两种不同的金属在电气上相互连接到一起，就将形成热电偶。与连接处温度成正比的电压将产生。这种电压很低；1微伏可能代表了若干度的温度。测量热电偶的电压，对额外连接点进行补偿，然后对结果进行线性化，这些就构成了使用热电偶对温度进行测量的基础。

当将热电偶连接到EM 231热电偶模块时，两根不同金属的导线将连接到模块的模块信号连接器上。在两根不同导线相互连接的位置处将形成传感器热电偶。

在两根不同导线与信号连接器的连接处将形成其它的两个热电偶。连接器的温度将产生一个电压，该电压将叠加到传感器热电偶的电压上。如果没有对该电压进行修正，那么，所报告的温度将偏离传感器温度。

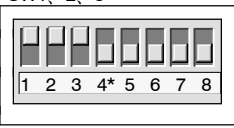
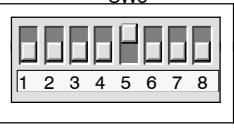
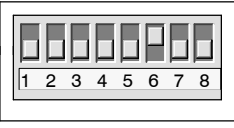

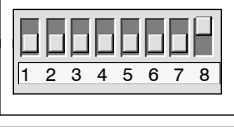
冷结点补偿用于补偿连接器热电偶。热电偶表将以基准接点温度为基础，通常为0摄氏度。冷结点补偿将用于将连接器补偿到0摄氏度。冷结点补偿可恢复连接器热电偶所增加的电压。先从内部测量模块的温度，然后将其转换为一个数值，该数值将添加到传感器转换值上。更正后的传感器转换值将通过热电偶表进行线性化。

### 配置EM 231热电偶模块

位于模块底部的配置用DIP开关，使您能够选择热电偶类型、明线检测、温度量程、冷结点补偿。为使DIP开关的设置生效，需要给PLC和/或给用户提24V电源重新上电。

DIP开关4保留为以后使用。将DIP开关4设置为0（向下或断开）位置。表A-25显示了其它DIP开关的设置情况。

表 A-25 配置热电偶模块DIP开关

开关1、2、3	热电偶类型	设置	说明
<p>SW1、2、3</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p> <p>*将DIP开关4设置为0 (向下) 位置。</p>	J (默认设置)	000	开关1到3为模块上所有通道选择热电偶类型 (或mV操作)。例如, 对于E类型, 热电偶 SW1 = 0、SW2 = 1、SW3 = 1。
	K	001	
	T	010	
	E	011	
	R	100	
	S	101	
	N	110	
	+/-80mV	111	
开关5	明线检测方向	设置	说明
<p>SW5</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p>	高标度端 (+3276.7度)	0	0指示明线的正方向 1指示明线的负方向
	低标度端 (-3276.8度)	1	
开关6	启用明线检测	设置	说明
<p>SW6</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p>	启用	0	将25 $\mu$ A电流注入输入端子, 可完成明线检测。明线启用开关可启用或禁用电流源。任何时候都要进行明线范围检查, 即使禁用电流源时也是如此。如果输入信号超出大约 $\pm 200$ mV, EM 231热电偶模块将检测明线。当检测到明线时, 模块读数将设置为由“明线检测”所选择的数值。
	禁用	1	
开关7	温度标尺	设置	说明
<p>SW7</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p>	摄氏度 ( $^{\circ}$ C)	0	EM 231热电偶模块可以以摄氏温度或华氏温度为单位报告温度。摄氏温度到华氏温度的转换在模块内部完成。
	华氏温度 ( $^{\circ}$ F)	1	
开关8	冷结点	设置	说明
<p>SW8</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p>	冷结点补偿启用	0	使用热电偶时, 必须启用冷结点补偿。如果未启用冷结点补偿, 则模块转换将出现错误, 因为热电偶导线连接到模块连接器时将产生电压。选择 $\pm 80$ mV范围时, 将自动禁用冷结点补偿。
	冷结点补偿禁用	1	

**提示**

- 明线电流源可能对某些低电平源（例如热电偶模拟程序）的信号产生干扰。
- 输入电压超过约 $\pm 200\text{mV}$ 时，将触发明线检测，即使此时禁用明线电流源。

**提示**

- 当环境温度变化时，模块错误可能超过规范。
- 超出模块环境温度范围规范，将可能引起模块冷结点补偿出错。

**使用热电偶：状态指示器**

EM 231热电偶模块为PLC提供指示温度或错误条件的数据字。状态位指示范围错误和用户电源/模块故障。LED指示模块的状态。您的程序应设计有逻辑语句以检测错误条件，并对应用作出适当的响应。表A-26显示了EM 231热电偶状态指示器。

表 A-26 EM 231热电偶状态指示器

错误条件	通道数据	SF LED 红色	24 V LED 绿色	范围状态位 <sup>1</sup>	24 VDC用户 电源故障 <sup>2</sup>
无错	转换数据	OFF	ON	0	0
24 V丢失	32766	OFF	OFF	0	1
明线和电流源启用	-32768/32767	闪烁	ON	1	0
超出输入范围	-32768/32767	闪烁	ON	1	0
诊断错误 <sup>3</sup>	0000	ON	OFF	0	注释 <sup>3</sup>

<sup>1</sup> 范围状态位是模块错误寄存器字节中的位3（对于模块1，是SMB9；对于模块2，是SMB11，依此类推）

<sup>2</sup> 用户电源故障状态位是模块错误寄存器字节中的位2（SMB9、SMB11等，参见附录D）

<sup>3</sup> 诊断错误将引起模块配置错误。模块配置错误出现之前，“用户电源故障”状态位可能设置，也可能未设置。

**提示**

通道数据格式是2的补码，16位字。温度以0.1度为单位显示。例如，如果测得的温度为100.2度，则报告的数据为1002。电压数据将标定到27648。例如， $-60.0\text{mV}$ 将报告为 $-20736$ （ $= -60\text{mV}/80\text{mV} * 27648$ ）。

如果PLC已经读取数据，将每隔400毫秒对所有四个通道进行更新。如果在一个更新周期内，PLC没有读取数据，则模块将继续报告旧的数据，直到PLC读取数据之后的下一个模块更新周期。为了使通道数据总是为当前数据，建议PLC读取数据的频率至少和模块更新速率相同。

**提示**

如果正在使用EM 231热电偶模块，应该禁用PLC中的模拟过滤。模拟过滤可能会妨碍及时检测到错误条件。

表 A-27 各种热电偶类型的温度范围 (°C) 和精度

数据字 (1个数字位=0.1°C)		类型J	类型K	类型T	类型E	类型R、S	类型N	±80mV	
十进制	十六进制								
32767	7FFF	>1200.0 °C	>1372.0 °C	>400.0 °C	>1000.0°C	>1768.0°C	>1300.0°C	>94.071mV	OF
↑	↑							↑	↑
32511	7EFF							94.071mV	OR
:	:							80.0029mV	
27649	6C01							80mV	NR
27648	6C00								
:	:								
17680	4510		↑						
:	:								
13720	3598		1372.0°C						
:	:		超出范围						
13000	32C8	↑	1300.0°C					1300.0°C	
:	:								
12000	2EE0	1200.0°C							
:	:								
10000	2710			↑					
:	:								
4000	0FA0			400.0°C		400.0°C			
:	:								
1	0001	0.1°C	0.1°C	0.1°C	0.1°C	0.1°C	0.1°C	0.0029mV	
0	0000	0.0°C	0.0°C	0.0°C	0.0°C	0.0°C	0.0°C	0.0mV	
-1	FFFF	-0.1°C	-0.1°C	-0.1°C	-0.1°C	-0.1°C	-0.1°C	-0.0029mV	
:	:								
-500	FE0C								
-1500	FA24	-150.0°C							
:	:								
-2000	F830	低于范围	-200.0°C						
:	:								
-2100	F7CC	-210.0°C							
:	:								
-2550	F60A		低于范围						
:	:								
-2700	F574	↓	-270.0°C	-255.0°C	-255.0°C				
:	:			低于范围	低于范围				
-27648	9400								
-27649	93FF		↓	↓	↓				
:	:								
-32512	8100								
↓	↓								
-32768	8000	<<-210.0°C	<<-270.0°C	<<-270.0°C	<<-270.0°C	<<-50.0°C	<<-270.0°C	<<-94.071mV	UF
全量程范围的精度		±0.1%	±0.3%	±0.6%	±0.1%	±0.6%	±0.1%	±0.1%	
精度 (没有冷结点补偿的额定范围)		±1.5°C	±1.7°C	±1.4°C	±1.3°C	±3.7°C	±1.6°C	±0.10%	
冷接点误差		±1.5°C	±1.5°C	±1.5°C	±1.5°C	±1.5°C	±1.5°C	N/A	

\*OF = 溢出; OR = 超出范围; NR = 额定范围; UR = 低于范围; UF = 下溢

↑ 表示大于该值以及小于明线阈值的模拟值均报告为溢出数据值32767 (0x7FFF)。  
 ↓ 表示小于该值以及大于明线阈值的模拟值均报告为下溢数据值-32768 (0x8000)。

A

表 A-28 各种热电偶类型的温度范围 (°F)

数据字 (1个数字位=0.1°F)		类型J	类型K	类型T	类型E	类型R、S	类型N	±80mV	
十进制	十六进制								
32767	7FFF	>2192.0 °F	>2502.0 °F	>752.0 °F	>1832.0 °F	>3214.0 °F	>2372.0 °F	>94.071mV	OF
↑	↑					↑		↑	↑
32511	7EFF							94.071mV	OR
32140	7D90					3214.0 °F		80.0029mV	
27649	6C01							80mV	NR
27648	6C00					2764.8 °F			
:	:								
25020	61B8		2502.0 °F						
:	:		超出范围						
23720	5CA8	↑	2372.0 °F				2372.0 °F		
:	:								
21920	55A0	2192.0 °F							
:	:								
18320	4790				1832.0 °F				
:	:								
7520	1D60			752.0 °F		752.0 °F			
:	:								
320	0140					低于范围	32.0 °F		
:	:								
1	0001	0.1 °F	0.1 °F	0.1 °F	0.1 °F	0.1 °F	0.1 °F	0.0029mV	
0	0000	0.0 °F	0.0 °F	0.0 °F	0.0 °F	0.0 °F	0.0 °F	0.0mV	
-1	FFFF	-0.1 °F	-0.1 °F	-0.1 °F	-0.1 °F	-0.1 °F	-0.1 °F	-0.0029mV	
:	:								
-580	FD8C					-58.0 °F			
:	:								
-2380	F6B4	-238.0 °F							
:	:								
-3280	F330	低于范围	-328.0 °F				低于范围		
:	:								
-3460	F27C	-346.0 °F	低于范围						
:	:								
-4270	EF52			-427.0 °F	-427.0 °F				
:	:			低于范围	低于范围				
-4540	EE44	↓	-454.0 °F	-454.0 °F	-454.0 °F		-454.0 °F		
:	:								
-27648	9400							-80mV	
-27649	93FF							-80.0029mV	
:	:								
-32512	8100							-94.071mV	OR
↓	↓							↓	↓
-3268	8000	<-346.0 °F	<-454.0 °F	<-454.0 °F	<-454.0 °F	<-58.0 °F	<-454.0 °F	<-94.07 mV	UF

\*OF = 溢出; OR = 超出范围; NR = 正常范围; UR = 低于范围; UF = 下溢  
 ↑ 表示大于该值以及小于明线阈值的模拟值均报告为溢出数据值32767 (0x7FFF)。  
 ↓ 表示小于该值以及大于明线阈值的模拟值均报告为下溢数据值-32768 (0x8000)。

A

## EM 231 RTD模块

EM 231 RTD模块为S7-200系列与几个不同RTD的连接提供了方便的接口。它还使S7-200能够测量三个不同的阻抗范围。连接到该模块的两个RTD必须具有同样的类型。

### 配置EM 231 RTD模块

DIP开关使您能够选择RTD类型、布线配置、温度标尺以及传感器熔断方向。DIP开关位于模块底部，如图A-20所示。为使DIP开关设置生效，需要给PLC和/或用户24V电源重新上电。

根据表A-29所示，将DIP开关1、2、3、4和5设置成对应于RTD，可选择RTD类型。其它DIP开关的设置，请参见表A-30。

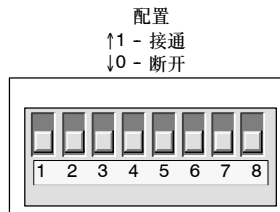


图 A-20 EM 231 RTD模块的DIP开关

表 A-29 选择RTD类型: DIP开关1到5

RTD类型和Alpha	SW1	SW2	SW3	SW4	SW5	RTD类型和Alpha	SW1	SW2	SW3	SW4	SW5
100Ω Pt 0.003850 (默认值)	0	0	0	0	0	100Ω Pt 0.00302	1	0	0	0	0
200Ω Pt 0.003850	0	0	0	0	1	200Ω Pt 0.003902	1	0	0	0	1
500Ω Pt 0.003850	0	0	0	1	0	500Ω Pt 0.003902	1	0	0	1	0
1000Ω Pt 0.003850	0	0	0	1	1	1000Ω Pt 0.003902	1	0	0	1	1
100Ω Pt 0.003920	0	0	1	0	0	备用	1	0	1	0	0
200Ω Pt 0.003920	0	0	1	0	1	100Ω Ni 0.00672	1	0	1	0	1
500Ω Pt 0.003920	0	0	1	1	0	120Ω Ni 0.00672	1	0	1	1	0
1000Ω Pt 0.003920	0	0	1	1	1	1000Ω Ni 0.00672	1	0	1	1	1
100Ω Pt 0.00385055	0	1	0	0	0	100Ω Ni 0.006178	1	1	0	0	0
200Ω Pt 0.00385055	0	1	0	0	1	120Ω Ni 0.006178	1	1	0	0	1
500Ω Pt 0.00385055	0	1	0	1	0	1000Ω Ni 0.006178	1	1	0	1	0
1000Ω Pt 0.00385055	0	1	0	1	1	10000Ω Pt 0.003850	1	1	0	1	1
100Ω Pt 0.003916	0	1	1	0	0	10Ω Cu 0.004270	1	1	1	0	0
200Ω Pt 0.003916	0	1	1	0	1	150Ω FS 电阻	1	1	1	0	1
500Ω Pt 0.003916	0	1	1	1	0	300Ω FS 电阻	1	1	1	1	0
1000Ω Pt 0.003916	0	1	1	1	1	600Ω FS 电阻	1	1	1	1	1



表 A-30 设置RTD DIP开关

开关6	明线检测	设置	说明
 <p>配置 ↑ 1 - 接通 ↓ 0 - 断开</p>	高标度端 (+3276.7度)	0	指示明线的正方向
	低标度端 (-3276.8度)	1	指示明线的负方向
开关7	温度标尺	设置	说明
 <p>配置 ↑ 1 - 接通 ↓ 0 - 断开</p>	摄氏度 (°C)	0	RTD模块可以以摄氏温度或华氏温度为单位报告温度。摄氏温度到华氏温度的转换在模块内部完成。
	华氏温度 (°F)	1	
开关8	布线方案	设置	说明
 <p>配置 ↑ 1 - 接通 ↓ 0 - 断开</p>	3线	0	有三种方式可将RTD模块用导线连接到传感器（如图所示）。其中精度最高的是4线连接。精度最低的是2线连接，建议只将其用于可忽略布线误差的应用场合。
	2线或4线	1	

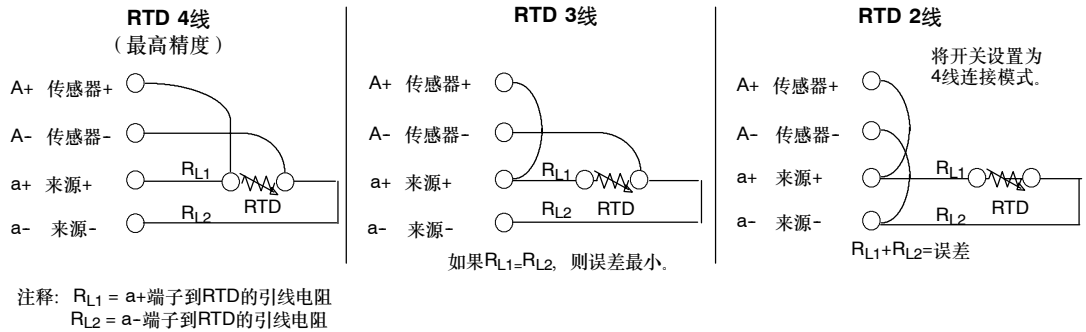


图 A-21 通过4、3、2线连接方式将RTD用导线连接到传感器

## EM 231 RTD状态指示器

RTD模块为PLC提供指示温度或错误条件的数据字。状态位指示范围错误和用户电源/模块故障。LED指示模块的状态。您的程序应设计有逻辑语句以检测错误条件，并对应用作出适当的响应。表A-31所示为EM 231 RTD模块提供的状态指示器。



### 提示

通道数据格式为2的补码，16位字。温度以0.1度为单位显示。（例如，如果测得的温度为100.2度，则报告的数据为1002。）电阻数据标定到27648。例如，满量程电阻的75%将报告为20736。  
( $225\Omega / 300\Omega * 27648 = 20736$ )

表 A-31 EM 231 RTD状态指示器

错误条件	通道数据	SF LED 红色	24 V LED 绿色	范围状态位 <sup>1</sup>	24 VDC用户电源故障 <sup>2</sup>
无错	转换数据	OFF	ON	0	0
24 V丢失	32766	OFF	OFF	0	1
SW检测明线	-32768/32767	闪烁	ON	1	0
超出输入范围	-32768/32767	闪烁	ON	1	0
诊断错误 <sup>3</sup>	0000	ON	OFF	0	注 <sup>3</sup>

<sup>1</sup> 范围状态位是模块错误寄存器字节中的位3（对模块1，是SMB9；对模块2，是SMB11，依此类推）

<sup>2</sup> “用户电源故障”状态位是模块错误寄存器字节中的位2（例如SMB 9、SMB 11，请参见附录D。）

<sup>3</sup> 诊断错误将导致模块配置错误。模块配置错误出现之前，“用户电源故障”状态位可能设置，也可能未设置。

如果PLC已经读取数据，将每隔405毫秒对通道数据进行更新。如果在一个更新周期内，PLC没有读取数据，则模块将继续报告旧的数据，直到PLC读取数据之后的下一个模块更新周期。为了使通道数据总是为当前数据，建议PLC读取数据的频率至少和模块更新速率相同。



### 提示

如果正在使用RTD模块，请务必禁用PLC中的模拟过滤。模拟过滤可能妨碍及时检测错误条件。

明线检测由RTD模块内部软件完成。声明超出输入范围，明线数据报告为熔断数据。明线检测最少需要三个模块扫描周期，也有可能需要更长的周期，这取决于哪一条导线开路。检测开路的Source+和/或Source-导线所花费的时间最少。检测开路的Sense+和/或Sense-导线则可能需要5秒或更多的时间。在断续检测明线时，尤其是在电气噪声严重的环境中，开路的sense线上可能随机出现有效的数据。电气噪声还可能延长检测明线条件的时间。为此，建议在有效数据报告完毕之后，将明线/超出范围的指示锁定在应用程序中。

### EM 231 RTD模块范围

各种类型RTD模块的EM 231 RTD温度范围和精度均显示在表A-32和A-33中。

表 A-32 各种RTD类型的温度范围 (°C) 和精度

系统字 (1个数字位=0.1°C)		Pt10000	Pt100、 Pt200、 Pt500、 Pt1000	Ni100、 Ni120、Ni1000	Cu9.035	0-150Ω	0-300Ω	0-600Ω	
十进制	十六进制								
32767	7FF.								
32766	7FFE					↑	↑	↑	
32511	7EFF					176.383Ω	352.767Ω	705.534Ω	
29649	6C01					150.005Ω	300.011Ω	600.022Ω	
27648	6C00					150.000Ω	300.000Ω	600.000Ω	
25000	61A8								↑
18000	4650								OR
15000	3A98								
13000	32C8								
10000	2710	↑ 1000.0°C	↑ 1000.0°C						
8500	2134		850.0°C						
6000	1770	600.0°C							
3120	0C30			↑					
2950	0B86			295.0°C					
2600	0A28								
2500	09C4			250.0°C					
1	0001	0.1°C	0.1°C	0.1°C	0.1°C	0.005Ω	0.011Ω	0.022Ω	
0	0000	0.0°C	0.0°C	0.0°C	0.0°C	0.000Ω	0.000Ω	0.000Ω	
-1	FFFF	-0.1°C	-0.1°C	-0.1°C	-0.1°C				
						(不可能为负值)			
-600	FDA8			-60.0°C		↓	↓	↓	N R
-1050	FBE6			-105.0°C					
-2000	F830	C	-200.0°						
-2400	F6A0								
-2430	F682	C	-243.0°C						
		↓	↓						
-5000	EC78								
-6000	E890								UR
-10500	D6FC								↓
-12000	D120								
-20000	4E20								
-32767	8001								
-32768	8000								
全量程范围的精度		±0.4%	±0.1%	±0.2%	±0.5%	±0.1%	±0.1%	±0.1%	
精度 (额定范围)		±4°C	±1°C	±0.6°C	±2.8°C	±0.15Ω	±0.3Ω	±0.6Ω	
*OF = 溢出; OR = 超出范围; KNURL = 额定范围; OUR = 低于范围; OUI = 下溢									
↑或↓表示超过限制值的所有模拟数值都将报告所选择的熔断数值32767 (0x7FF) 或-32768 (0x8000)。									

A

表 A-33 各种RTD类型的温度范围 (°F)

系统字 (1个数字位=0.1°F)		PT1000	Pt100、Pt200、Pt500、Pt1000	Ni100、Ni120、Ni1000	Cu9.035	
十进制	十六进制					
32767	7FFF					
32766	7PHAGE					↑
						超出范围
		↑	↑			
18320	4790	1832.0°F	1832.0°F			
15620	3D04		1562.0°F			
11120	2B70	1112.0°F				
5936	1730			↑	593.6°F	
5630	15FE			↑	563.0°F	
5000	1388				500.0°F	
4820	12D4				482.0°F	
						正常范围
1	0001	0.1°F	0.1°F	0.1°F	0.1°F	
0	0000	0.0°F	0.0°F	0.0°F	0.0°F	
-1	FFFF	-0.1°F	-0.1°F	-0.1°F	-0.1°F	
-760	FD08				-76.0°F	
-1570	F9DE				-157.0°F	
-3280	F330	-328.0°F	-328.0°F			
-4000	F060					
-4054	F02A	-405.4°F	-405.4°F			
		↓	↓			
-5000	EC78					
-6000	E890					
-10500	D6FC					
-32767	8001					
-32768	8000					
↑或↓表示超过限制值的所有模拟数值都将报告所选择的熔断数值32767 (0x7FFF) 或-32768 (0x8000)。						

A

## EM 277 PROFIBUS-DP模块规范

表 A-34 EM 277 PROFIBUS-DP模块订购号

订购号	扩展型号	EM输入	EM输出	可移动连接器
6ES7 277-0AA22-0XA0	EM 277 PROFIBUS-DP	-	-	否

表 A-35 EM 277 PROFIBUS-DP模块通用规范

订购号	模块名称与说明	尺寸 (毫米) (W x H x D)	重量	损耗	+5 VDC	VDC要求 +24 VDC
6ES7 277-0AA22-0XA0	EM 277 PROFIBUS-DP	71 x 80 x 62	175 g	2.5 W	150毫安	参见下文

表 A-36 EM 277 PROFIBUS-DP模块规范

常规	6ES7 277-0AA22-0XA0
端口数	1
电气接口	RS-485
PROFIBUS-DP/MPI波特率 (自动设置)	9.6、19.2、45.45、93.75、187.5和500千波特；1、1.5、3、6和12兆波特
协议	PROFIBUS-DP从属装置和MPI从属装置
电缆长度	
至多93.75千波特	1200米
187.5千波特	1000米
500千波特	400米
1至1.5兆波特	200米
3至12兆波特	100米
网络性能	
站地址设置	0至99 (由旋转开关进行设置)
每段最大站数	32
每个网络最大站数	126, 至多99个EM 277站
MPI接头	总共6个, 2个保留 (1个用于PG, 1个用于OP)
24 VDC输入电源要求	
电压范围	20.4至28.8 VDC (2类或PLC的传感器电源)
最大电流	
仅端口激活的模块	30毫安
增加90毫安、5V的端口负载	60毫安
增加120毫安、24V的端口负载	180毫安
纹波噪声 (<10 MHz)	<1 V 峰间值 (最大值)
绝缘 (现场到逻辑电路) <sup>1</sup>	500 VAC, 1分钟
通讯端口上为5 VDC电源	
每个端口的最大电流	90毫安
绝缘 (24 VDC到逻辑电路)	500 VAC, 1分钟
通讯端口上为24 VDC电源	
电压范围	20.4到28.8 VDC
每个端口的最大电流	120毫安
电流限制	0.7至2.4 A
绝缘	未绝缘, 与输入24 VDC相同的电路

<sup>1</sup> 24 VDC电源不给模块逻辑电路提供任何电源。24 VDC为通讯端口提供电源。

## 支持智能模块的S7-200CPU

EM 277 PROFIBUS-DP从属装置模块设计为与表A-37所示S7-200 CPU一起工作的智能扩充模块。

表 A-37 EM 277 PROFIBUS-DP模块与S7-200 CPU的兼容性

CPU	说明
CPU 222版本1.10或更高版本	CPU 222 DC/DC/DC
	CPU 222 AC/DC/继电器
CPU 224版本1.10或更高版本	CPU 224 DC/DC/DC
	CPU 224 AC/DC/继电器
CPU 226版本1.10或更高版本	CPU 226 DC/DC/DC
	CPU 226 AC/DC/继电器
CPU 226XM版本1.10或更高版本	CPU 226XM DC/DC/DC
	CPU 226XM AC/DC/继电器

## 地址开关和LED

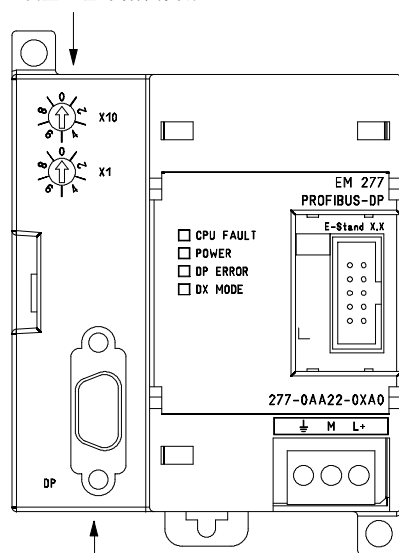
地址开关和状态LED位于模块正面，如图A-22所示。此外，还显示了DP从属装置端口连接器的外接插针。

EM 277 PROFIBUS-DP的正视图

地址开关:

x10=设置地址的最高有效位

x1= 设置地址的最低有效位



9针D型连接器外接插针

9针D型 内孔接头	插针#	说明
1	1	机壳接地，连接到连接器外壳
2	2	24V回流（与接线盒上的M相同）
3	3	绝缘信号B（RxD/TxD+）
4	4	要发送的绝缘请求（TTL电平）
5	5	绝缘的+5V回流
6	6	绝缘的+5V（最大90毫安）
7	7	+24V（最大120毫安，带反向电压保护二极管）
8	8	绝缘信号A（RxD/TxD-）
9	9	无任何连接

注意：绝缘意味着与数字逻辑电路和24V输入电源的500V绝缘。

图 A-22 EM 277 PROFIBUS-DP

## 分布式外围设备（DP）标准通讯

PROFIBUS-DP（或DP标准）是由欧洲标准EN 50170定义的一种远程I/O通讯协议。对于遵守该标准的设备，即使由各个不同厂商生产，也互相兼容。DP表示分布式外围设备，也就是远程I/O。PROFIBUS表示过程现场总线。

EM 277 PROFIBUS-DP模块使用DP标准协议，该协议在下列通讯协议标准中为从属装置定义：

- EN 50 170 (PROFIBUS) 说明总线访问与传输协议，并指定数据传输介质的属性。
- EN 50 170 (DP标准) 说明DP主设备与DP从属装置之间数据的高速周期性交换。该标准定义配置与参数分配的步骤，解释如何与分布式I/O函数进行周期性数据交换，并列出了所支持的诊断选项。

DP主设备的配置包括地址、从属装置类型以及从属装置所需要的任何参数分配信息。此外，还告诉主设备把从从属装置中读取的数据（输入）放置在何处，以及从哪里获取将要写入从属装置的数据（输出）。DP主设备将建立网络，然后启动其DP从属装置。主设备把参数分配信息和I/O配置写入从属装置。然后，主设备将读取从属装置中的诊断信息，以便验证DP从属装置接受了参数和I/O配置。然后，主设备开始与从属装置交换I/O数据。与从属装置进行的每次事务处理都是写入输出、读取输入。这种数据交换方式将无限期地继续下去。如果存在例外情况，从属装置将通知主设备，主设备随后将读出从属装置中的诊断信息。

一旦DP主设备将参数和I/O配置写入DP从属装置，并且从属装置已接受来自主设备的参数和配置，主设备就拥有了该从属装置。从属装置只接受其所属主设备的写入请求。网络上的其它主设备可读取从属装置的输入和输出，但它们不能将任何内容写入从属装置。

### 使用EM 277将S7-200连接为DP从属装置

通过EM 277 PROFIBUS-DP扩展从属装置模块，可将S7-200 CPU连接到PROFIBUS-DP网络。通过串行I/O总线，可将EM 277连接到S7-200 CPU。PROFIBUS网络通过其DP通讯端口连接到EM 277 PROFIBUS-DP模块。该端口以9600波特到12兆波特之间的任意PROFIBUS波特率运行。参见EM 277 PROFIBUS-DP模块规范，获取支持的波特率。

作为DP从属装置，EM 277模块将接受来自主设备的多个不同I/O配置，这将使您能够对所要传送的数据进行处理，以满足实际应用的需要。与许多DP设备不同，EM 277模块不仅仅只是传送I/O数据。通过首先将数据移动到S7-200 CPU中的变量内存，可将输入、计数器数值、计时器数值或其它计算得到的数值传送到主设备。同样，主设备中的数据将存储在S7-200 CPU的变量内存中，并可移动到其它数据区。

EM 277 PROFIBUS-DP模块的DP端口将被连接到网络中的DP主设备上，且仍然作为MPI从属装置与同一网络中的其它主设备（例如SIMATIC编程设备或S7-300/S7-400 CPU）进行通讯。图A-23显示包含CPU 224和EM 277 PROFIBUS-DP模块的PROFIBUS网络。

- CPU 315-2为DP主设备，且已经通过具有STEP 7编程软件的SIMATIC编程设备进行了配置。
- CPU 224是属于CPU 315-2的DP从属装置。ET 200 I/O模块也是属于CPU 315-2的从属装置。
- S7-200 CPU连接到PROFIBUS网络，且可通过S7-200 CPU用户程序中的XGET指令来读取CPU 224的数据。

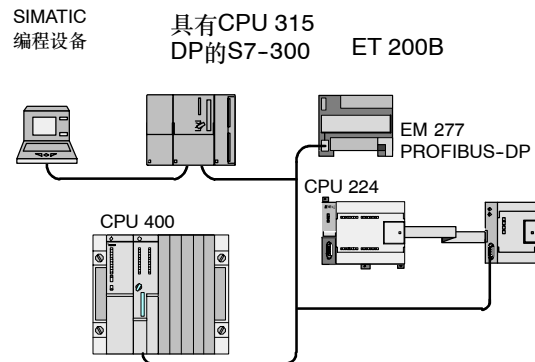


图 A-23 PROFIBUS网络中的EM 277 PROFIBUS-DP模块和CPU 224

## 配置

为将EM 277 PROFIBUS-DP用作DP从属装置，必须设置DP端口的站地址，以便与主设备配置中的地址相匹配。使用EM 277模块上的旋转开关可设置站地址。在切换开关之后，必须给CPU重新上电，以便使新的从属装置地址生效。

通过将其输出区中的信息发送到从属装置的输出缓冲区（称为“接收信箱”），主设备可与其各个从属装置进行数据交换。从属装置将通过返回一个输入缓冲区（称为“发送信箱”）来响应主设备中的信息，该输入缓冲区将被主设备存储在输入区中。

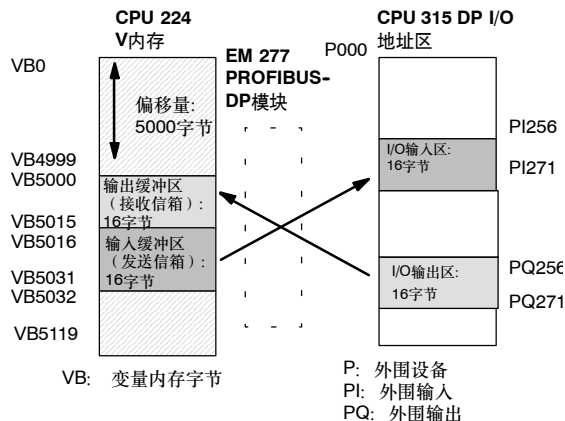


图 A-24 V内存和I/O地址区

图A-24表示PROFIBUS-DP主设备的V内存和I/O地址区的实例。

EM 277 PROFIBUS-DP可由DP主设备进行配置，以便接受主设备的输出数据，并将输入数据返回到主设备。输出和输入数据缓冲区位于S7-200 CPU的变量内存（V内存）中。在配置DP主设备时，将定义V内存中的字节位置，输出数据缓冲区将从这里开始作为EM 277的参数分配信息的一部分。您也应定义I/O配置，该配置将作为写入S7-200 CPU的输出数据量以及从S7-200 CPU返回的输入数据量。EM 277将确定I/O配置的输入和输出缓冲区的大小。DP主设备将把参数分配和I/O配置信息写入EM 277 PROFIBUS DP模块。随后，EM 277把V内存地址和输入与输出数据长度传送给S7-200 CPU。

图A-24显示了CPU 224中V内存的内存模型以及DP主设备CPU的I/O地址区。在本实例中，DP主设备已经定义了I/O配置，该配置具有16个输出字节、16个输入字节和V内存偏移量5000。CPU 224中的输出缓冲区与输入缓冲区长度（由I/O配置确定）都具有16个字节的长度。输出数据缓冲区的起始地址为V5000；输入缓冲区紧接在输出缓冲区的后面，从V5016开始。输出数据（来自主设备）放置在V内存的V5000处。输入数据（输入到主设备）则取自于V内存的V5016处。



### 提示

如果使用具有三个字节的数据单元（一致性数据）或大于四个字节的数据单元进行工作，则必须使用SFC14来读取DP从属装置的输入，使用SFC15对DP从属装置的输出进行编址。更多信息请参见S7-300和S7-400系统的系统软件与标准函数参考手册。



表A-38列出了EM 277 PROFIBUS-DP模块所支持的配置。EM 277模块的默认配置是两个字的输入和两个字的输出。

表 A-38 EM 277配置选项

配置	输入到主设备	从主设备输出	数据一致性
1	1字	1字	字一致性
2	2字	2字	
3	4字	4字	
4	8字	8字	
5	16字	16字	
6	32字	32字	
7	8字	2字	
8	16字	4字	
9	32字	8字	
10	2字	8字	
11	4字	16字	
12	8字	32字	
13	2个字节	2个字节	字节一致性
14	8字节	8字节	
15	32字节	32字节	
16	64字节	64字节	
17	4字节	4字节	缓冲区一致性
18	8字节	8字节	
19	12字节	12字节	
20	16字节	16字节	

您可将输入和输出缓冲区的位置配置在S7-200CPU的V内存中的任意位置。输入和输出缓冲区的缺省地址是VB0。输入和输出缓冲区的位置是主设备将要写入到S7-200 CPU的参数分配信息的一部分。对主设备进行配置，使其能够识别其从属装置，并将所需参数和I/O配置写入其每个从属装置中。

使用下列工具对DP主设备进行配置：

- 对于SIMATIC S5主设备，请使用COM PROFIBUS Windows软件
- 对于SIMATIC S7主设备，请使用STEP 7编程软件
- 对于SIMATIC 505主设备，请使用COM PROFIBUS以及TISOFT2或SoftShop

关于使用这些配置和编程软件程序包的详细信息，请参见这些设备的相关手册。关于PROFIBUS网络及其组件的详细信息，请参见 *ET 200 分布式 I/O 系统手册*。

## 数据一致性

PROFIBUS支持三种类型的数据一致性:

- 字节一致性可确保字节作为一个整体单元进行传送。
- 字一致性可确保字传送不被CPU的其它进程中中断（组成字的两个字节将始终一起移动，不能分开）。如果正在传送的数据值是整型数值，则应使用字一致性。
- 缓冲区一致性可确保数据的整个缓冲区作为一个独立的单元进行传送，而不被CPU中的任何其它进程中中断。如果数据值是双字或浮点值，或者当一组数值都与一个计算式或计算项目相关时，应使用缓冲区一致性。

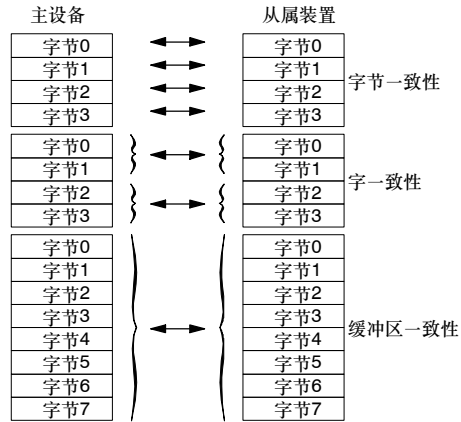


图 A-25 字节、字和缓冲区数据一致性

可将数据一致性设置为主设备I/O配置的一部分。数据一致性选项将写入到DP从属装置中，作为从属装置初始化的一部分。DP主设备和DP从属装置二者都将使用数据一致性选项，以便确保数据值（字节、字或缓冲区）能够在主设备与从属装置内进行不间断地传送。各种不同类型的一致性如图A-25所示。

## 用户程序注意事项

一旦DP主设备成功配置了EM 277 PROFIBUS-DP模块，则EM 277和DP主设备就将进入数据交换模式。在数据交换模式下，主设备将把输出数据写入EM 277 PROFIBUS-DP模块，随后，EM 277模块将响应最新的S7-200 CPU输入数据。EM 277模块然后将不断地更新S7-200 CPU的输入，以便将最新的输入数据提供给DP主设备。模块然后将输出数据传送给S7-200 CPU。主设备的输出数据将存放在从某地址开始的V内存（输出缓冲区）中，该地址由DP主设备在初始化期间提供。主设备的输入数据紧接着输出数据立即从V内存位置（输入缓冲区）获取。

主设备的输出数据必须由S7-200 CPU中的用户程序从输出缓冲区移动到它将要使用的数据区中。同样，主设备的输入数据必须从各种不同的数据区移动到输入缓冲区，然后再传送到主设备。

在循环的用户程序部分执行完毕之后，DP主设备的输出数据将立即存放到V内存。同时，输入数据（输入到主设备）将从V内存复制到EM 277中，以便传送到主设备。

当主设备中存在可供使用的新数据时，主设备的输出数据将只能写入到V内存。

主设备的输入数据将传送给与主设备进行下一个数据交换时的主设备。

在创建S7-200的用户程序时，V内存中数据缓冲区的起始地址和缓冲区大小必须已知。

## 状态信息

存在50个字节的特殊内存（SM），它们将分配给每个基于其物理位置的智能模块。模块将更新SM位置，对应于模块相对于CPU的位置（对其它模块而言）。如果它是第一个模块，则更新SMB200至SMB249。如果它是第二个模块，则更新SMB250至SMB299，依此类推。参见表A-39。

**注意**

智能模块SM区的分配方式对于V2.2及以后的版本有所不同。  
如果您使用的是V2.2以前的CPU，则应将所有的智能模块安装在邻近CPU和所有非智能模块之前的插槽中，以确保兼容性。

表 A-39 特殊内存字节SMB200至SMB549

特殊内存字节SMB200至SMB549						
插槽0中的智能模块	插槽1中的智能模块	插槽2中的智能模块	插槽3中的智能模块	插槽4中的智能模块 插槽4	插槽5中的智能模块	插槽6中的智能模块
SMB200至SMB249	SMB250至SMB299	SMB300至SMB349	SMB350至SMB399	SMB400至SMB449	SMB450至SMB499	SMB500至SMB549

这些SM单元在尚未建立与主设备的DP通讯时，将显示缺省值。在主设备已经将参数和I/O配置写入EM 277 PROFIBUS-DP模块之后，这些SM单元将显示DP主设备所设置的配置。应该检查协议状态字节（例如插槽0的SMB224），以便确保在使用表A-40所示的SM单元中的信息或V内存缓冲区中的数据之前，EM 277当前处于与主设备的数据交换模式。



**提示**

不能通过写入SM内存单元来配置EM 277 PROFIBUS-DP I/O缓冲区大小或缓冲区位置。只有DP主设备才能配置用于DP操作的EM 277 PROFIBUS-DP模块。

表 A-40 EM 277 PROFIBUS-DP的特殊内存字节

插槽0中的智能模块	...	插槽6中的智能模块	说明																															
SMB200至SMB215	...	SMB500至SMB515	模块名称（16个ASCII字符） “EM 277 ProfibusDP”																															
SMB216至SMB219	...	SMB516至SMB519	S/W修订号（4个ASCII字符） xxxx																															
SMW220	...	SMW520	错误代码 16#0000 无错 16#0001 无用户电源 16#0002至16#FFFF 保留																															
SMB222	...	SMB522	DP从属装置模块的站地址将通过地址开关进行设置（0-99十进制）																															
SMB223	...	SMB523	保留																															
SMB224	...	SMB524	DP标准协议状态字节 <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td colspan="6" style="text-align: left;">MSB</td> <td colspan="2" style="text-align: right;">LSB</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>S1</td><td>S0</td> </tr> </table>   <table style="margin-left: auto; margin-right: auto;"> <tr> <td>S1</td><td>S0</td><td>DP 标准状态字节说明</td> </tr> <tr> <td>0</td><td>0</td><td>DP 电源接通后没有启动通讯</td> </tr> <tr> <td>0</td><td>1</td><td>检测到配置/参数化错误</td> </tr> <tr> <td>1</td><td>0</td><td>当前处于数据交换模式</td> </tr> <tr> <td>1</td><td>1</td><td>退出数据交换模式</td> </tr> </table> </div>	MSB						LSB		0	0	0	0	0	0	S1	S0	S1	S0	DP 标准状态字节说明	0	0	DP 电源接通后没有启动通讯	0	1	检测到配置/参数化错误	1	0	当前处于数据交换模式	1	1	退出数据交换模式
MSB						LSB																												
0	0	0	0	0	0	S1	S0																											
S1	S0	DP 标准状态字节说明																																
0	0	DP 电源接通后没有启动通讯																																
0	1	检测到配置/参数化错误																																
1	0	当前处于数据交换模式																																
1	1	退出数据交换模式																																
SMB225	...	SMB525	DP标准协议-从属装置的主设备的地址（0至126）																															
SMW226	...	SMW526	DP标准协议-作为偏离VB0偏移量的输出缓冲区的V内存地址。																															
SMB228	...	SMB528	DP标准协议-输出数据的字节数																															
SMB229	...	SMB529	DP标准协议-输入数据的字节数																															
SMB230至SMB249	...	SMB530至SMB549	保留-上电时将清除																															

注意：每当DP从属装置模块接受配置/参数信息时，就将更新SM单元。即使检测到配置/参数错误，也将更新这些单元。每次上电时，这些单元都将清零。



## EM 277 PROFIBUS-DP的LED状态指示器

EM 277 PROFIBUS-DP模块在前面板上具有四个状态LED，用于指示DP端口的运行状态：

- 在接通S7-200 CPU之后，只要没有试图进行DP通讯，DX MODE LED就将仍然保持关闭。
- 一旦成功启动DP通讯（EM 277 PROFIBUS-DP模块已经处于与主设备进行数据交换的模式），DX MODE LED就将变为绿色，且保持点亮，直到退出数据交换模式。
- 如果DP通讯丢失，这将迫使EM 277模块退出数据交换模式，则DX MODE LED将熄灭，且DP ERROR LED将变为红色。这种情况将一直持续到S7-200 CPU的电源关闭或重新开始进行数据交换。
- 如果在DP主设备正在写入EM 277模块的I/O配置或参数信息中存在错误，则DP ERROR LED将以红色进行闪烁。
- 如果没有提供用户24 VDC电源，则POWER LED将熄灭。

表A-41概括了用EM 277状态LED表示的各种状态指示。

表 A-41 EM 277 PROFIBUS-DP模块状态LED

LED	OFF	红色	闪烁红色	绿色
CPU故障	模块正常	内部模块故障	--	--
POWER	无24 VDC用户电源	--	--	24 VDC用户电源正常
DP ERROR	无错	离开数据交换模式	参数化/配置错误	--
DX MODE	未处于数据交换模式	--	--	处于数据交换模式

注意：当EM 277 PROFIBUS-DP模块专门用作MPI从属装置时，只有绿色电源LED点亮。

### 附加的配置特性

EM 277 PROFIBUS-DP模块可用作其他MPI主设备的通讯接口，而不管它是否用作PROFIBUS-DP从属装置。模块可使用S7-300/400的XGET/XPUT功能提供S7-300/400到S7-200的连接。使用MPI或PROFIBUS参数设置的STEP 7-Micro/WIN和网卡（例如CP5611）、OP设备或TD 200（版本2.0或更高版本，订购号6ES7 272-0AA20-0YA0）均可通过EM 277 PROFIBUS-DP模块与S7-200进行通讯。

除DP主设备以外，最多可有六个连接（六个设备）与EM 277 PROFIBUS-DP模块相连。其中，一个连接为编程设备（PG）保留，一个为操作员面板（OP）保留。其它四个连接则可由任意MPI主设备使用。为了使EM 277 PROFIBUS-DP模块能够与多个主设备进行通讯，所有的主设备都必须在同样的波特率下运行。关于可能的网络配置，请参见图A-26。

当EM 277 PROFIBUS-DP模块用于MPI通讯时，MPI主设备的所有信息都必须使用模块的站地址进行发送，这些信息发送给模块所连接的S7-200。发送给EM 277 PROFIBUS-DP模块的MPI信息将传递给S7-200。

EM 277 PROFIBUS-DP模块是一个从属装置模块，不能使用NETR和NETW功能在S7-200 PLC之间进行通讯。EM 277 PROFIBUS-DP模块也不能用于自由端口的通讯。

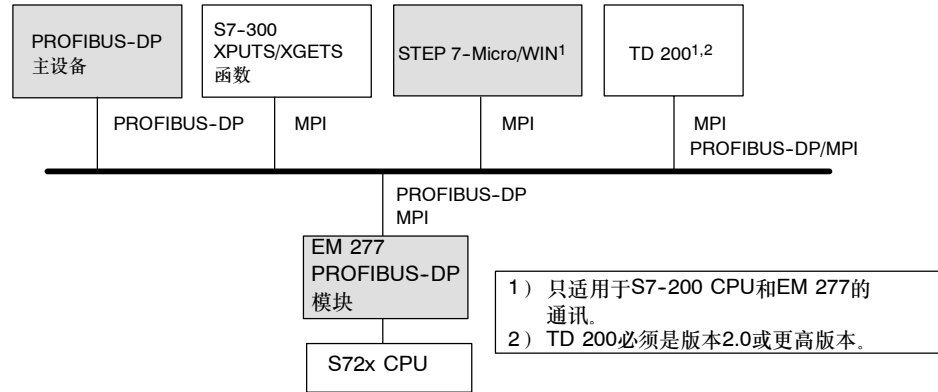


图 A-26 PROFIBUS-DP/MPI网络

### 设备数据库文件: GSD

不同的PROFIBUS设备具有不同的性能特征。这些特征在功能（例如I/O信号和诊断信息数量）或总线参数（例如传输速度和时间监控）方面各不相同。不同的设备类型和厂商，这些参数有所不同，通常在技术手册中说明。为帮助您进行简单的PROFIBUS配置，我们在一个称作设备数据库文件或GSD文件的电子数据表单中对特殊设备的性能特征进行了说明。基于GSD文件的这些配置工具允许将来自不同厂商的设备简单地集成到一个单独的网络中。

设备数据库文件以精确定义的格式提供设备特征的全面说明。这些GSD文件由各类设备的厂商准备，可供PROFIBUS用户使用。GSD文件允许配置系统读取PROFIBUS设备的特征，并在配置网络时使用该信息。

COM PROFIBUS或STEP 7软件的最新版本包括用于EM 277 PROFIBUS-DP模块的配置文件。如果您的软件版本不包括EM 277的配置文件，可以通过网站[www.profibus.com](http://www.profibus.com)访问最新的GSD文件（SIEM089D.GSD）。

如果您不是使用西门子主设备，请参考厂商提供的文档，了解如何使用GSD文件配置主设备。

```

;=====
;具有DPC31的EM277PROFIBUS-DP的GSD文件
;MLFB: 6ES7 277-0AA2-0XA0
;日期: 2001年3月26日
;=====
#Profibus_DP
;常规参数
GSD_Revision      = 1
Vendor_Name       = "Siemens"
Model_Name        = "EM 277 PROFIBUS-DP"
Revision          = "V1.02"
Ident_Number      = 0x089D
Protocol_Ident    = 0
Station_Type      = 0
FMS_supp         = 0
Hardware_Release  = "1.00"
Software_Release  = "1.02"
9.6_supp         = 1
19.2_supp        = 1
45.45_supp       = 1
93.75_supp       = 1
187.5_supp       = 1
500_supp         = 1
1.5M_supp        = 1
3M_supp          = 1
6M_supp          = 1
12M_supp         = 1
MaxTsdr_9.6      = 60
MaxTsdr_19.2     = 60
MaxTsdr_45.45    = 250
MaxTsdr_93.75    = 60
MaxTsdr_187.5    = 60
MaxTsdr_500      = 100
MaxTsdr_1.5M     = 150
MaxTsdr_3M       = 250
MaxTsdr_6M       = 450
MaxTsdr_12M      = 800
Redundancy        = 0
Repeater_Ctrl_Sig = 2
24V_Pins         = 2

;从属装置规范:
OrderNumber="6ES7 277-0AA2-0XA0"
Periphery="SIMATIC S5"
Slave_Family=10rdF@SIMATIC

Freeze_Mode_supp = 1
Sync_Mode_supp   = 1
Set_Slave_Add_Supp = 0
Auto_Baud_supp   = 1
Min_Slave_Intervall = 1
Fail_Safe        = 0
Max_Diag_Data_Len = 6
Modul_Offset     = 0
Modular_Station  = 1
Max_Module       = 1
Max_Input_len    = 128
Max_Output_len   = 128
Max_Data_len     = 256

; UserPrmData定义
ExtUserPrmData=1 "I/O Offset in the V-memory"
Unsigned16 0 0-10239
EndExtUserPrmData
; UserPrmData: 长度和预置:
User_Prm_Data_Len=3
User_Prm_Data= 0,0,0
Max_User_Prm_Data_Len=3
Ext_User_Prm_Data_Const ( 0 ) =0x00,0x00,0x00
Ext_User_Prm_Data_Ref ( 1 ) =1

```

```

;=====
; GSD文件, 续前
;=====
; 模块定义列表
Module = "2 Bytes Out/ 2 Bytes In"      -" 0x31
EndModule
Module = "8 Bytes Out/ 8 Bytes In"      -" 0x37
EndModule
Module = "32 Bytes Out/ 32 Bytes In"    -"
0xC0,0x1F,0x1F
EndModule
Module = "64 Bytes Out/ 64 Bytes In"    -"
0xC0,0x3F,0x3F
EndModule
Module = "1 Word Out/ 1 Word In"        -" 0x70
EndModule
Module = "2 Word Out/ 2 Word In"        -" 0x71
EndModule
Module = "4 Word Out/ 4 Word In"        -" 0x73
EndModule
Module = "8 Word Out/ 8 Word In"        -" 0x77
EndModule
Module = "16 Word Out/ 16 Word In"      -" 0x7F
EndModule
Module = "32 Word Out/ 32 Word In"      -"
0xC0,0x5F,0x5F
EndModule
Module = "2 Word Out/ 8 Word In"        -"
0xC0,0x41,0x47
EndModule
Module = "4 Word Out/ 16 Word In"       -"
0xC0,0x43,0x4F
EndModule
Module = "8 Word Out/ 32 Word In"       -"
0xC0,0x47,0x5F
EndModule
Module = "8 Word Out/ 2 Word In"        -"
0xC0,0x47,0x41
EndModule
Module = "16 Word Out/ 4 Word In"       -"
0xC0,0x4F,0x43
EndModule
Module = "32 Word Out/ 8 Word In"       -"
0xC0,0x5F,0x47
EndModule
Module = "4 Byte buffer I/O"            -" 0xB3
EndModule
Module = "8 Byte buffer I/O"            -" 0xB7
EndModule
Module = "12 Byte buffer I/O"           -" 0xBB
EndModule
Module = "16 Byte buffer I/O"           -" 0xBF
EndModule

```

图 A-27 EM 277 PROFIBUS模块的GSD文件列表

## 与CPU进行DP通讯的范例程序

下面所示的是用于插槽0中PROFIBUS-DP模块的CPU语句表的范例程序，它们将使用SM内存中的DP端口信息。该程序可确定从SMW226开始的DP缓冲区位置以及SMB228至SMB229的缓冲区大小。该信息用于将DP输出缓冲区中的数据复制到CPU过程图像输出寄存器。类似地，可将CPU过程图像输入寄存器中的数据复制到V内存输入缓冲区。

### 注意

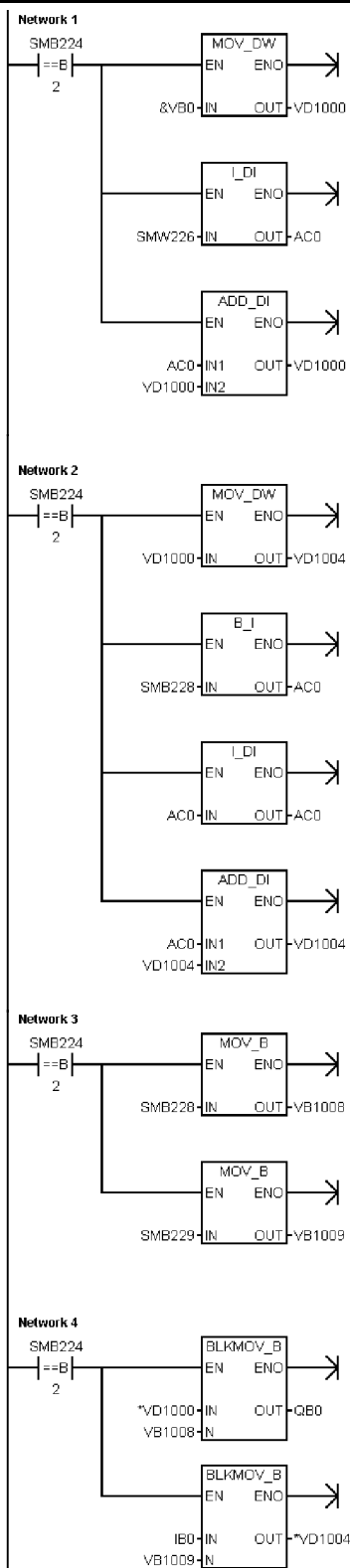
智能模块SM区的分配方式对于V2.2及以后的版本有所不同。

如果您使用的是V2.2以前的CPU，则应将所有的智能模块安装在邻近CPU和所有非智能模块之前的插槽中，以确保兼容性。

在位置0中的DP模块的下列范例程序中，SM存储区中的DP配置数据将提供DP从属装置的配置。程序使用下列数据：

SMW220	DP模块错误状态
SMB224	DP状态
SMB225	主设备地址
SMW226	V内存的输出偏移量
SMB228	输出数据的字节数
SMB229	输入数据的字节数
VD1000	输出数据指针
VD1004	输入数据指针

## 与CPU进行DP通讯的实例



程序段1 //计算输出数据指针。如果处于数据交换模式：  
//1. 输出缓冲区是偏移VB0的偏移量  
//2. 将Vmem偏移量转换为双整数  
//3. 添加到VB0地址，得到输出数据指针。

```
LDB= SMB224, 2
MOV_D &VB0, VD1000
ITD SMW226, AC0
+D AC0, VD1000
```

程序段2 //计算输入数据指针。如果处于数据交换  
//模式：  
//1. 复制输出数据指针  
//2. 获取输出字节数  
//3. 添加到输出数据指针，获取起始输入数据  
// 指针。

```
LDB= SMB224, 2
MOV_D VD1000, VD1004
BTI SMB228, AC0
ITD AC0, AC0
+D AC0, VD1004
```

程序段3 //设置要复制的数据量。如果处于数据交换模式：  
//1. 获取要复制的输出字节数  
//2. 获取要复制的输入字节数

```
LDB= SMB224, 2
MOV_B SMB228, VB1008
MOV_B SMB229, VB1009
```

程序段4 //将主设备输出传送到CPU输出。将CPU输入  
//复制到主设备输入。如果处于数据交换模式：  
//1. 将主设备输出复制到CPU输出  
//2. 将CPU输入复制到主设备输入

```
LDB= SMB224, 2
BMB *VD1000, QB0, VB1008
BMB IB0, *VD1004, VB1009
```



## EM 241调制解调器模块规范

表 A-42 EM 241调制解调器模块订购号

订购号	扩展型号	EM输入	EM输出	可移动连接器
6ES7 241-1AA22-0XA0	EM 241调制解调器模块	-	8 <sup>1</sup>	否

1 8个Q输出用作调制解调器功能的逻辑控制，不能直接控制任何外部信号。

表 A-43 EM 241调制解调器模块通用规范

订购号	模块名称与说明	尺寸 (毫米) (W x H x D)	重量	损耗	VDC要求	
					+5 VDC	+24 VDC
6ES7 241-1AA22-0XA0	EM 241调制解调器模块	71.2 x 80 x 62	190 g	2.1 W	80毫安	70毫安

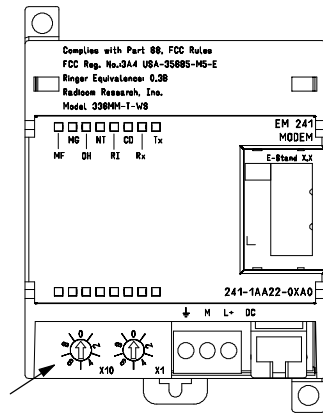
表 A-44 EM 241调制解调器模块规范

常规	6ES7241-1AA22-0XA0
<b>电话连接</b>	
绝缘 (电话线到逻辑电路和现场电源)	1500 VAC (电化)
物理连接	RJ11 (6位置, 4线)
调制解调器标准	Bell 103, Bell 212, V.21, V.22, V.22 bis, V.23c, V.32, V.32 bis, V.34 (默认)
安全性能	密码 回叫
拨号	脉冲或音调
讯息传送协议	数字 TAP (字母数字) UCP命令1、30、51
工业协议	Modbus PPI
<b>24 VDC输入电源要求</b>	
电压范围	20.4到28.8 VDC
绝缘 (现场电源到逻辑电路)	500 VAC, 1分钟

EM 241调制解调器模块将取代连接到CPU通讯端口的外部调制解调器的功能。利用安装在S7-200系统中的EM 241，与CPU进行远程通讯所需要的全部设备就是一台带有外部调制解调器和STEP7-Micro/WIN的个人计算机。

有关的配置信息，请参见第7章（通过网络进行通讯）。参见第10章（为调制解调器模块创建程序），以了解关于模块编程和高级性能的信息。

您可使用STEP7-Micro/WIN调制解调器扩充向导来配置EM 241调制解调器模块。关于调制解调器扩充向导的更详细信息，请参见第10章。



国家代码开关

图 A-28 EM 241调制解调器模块接线盒示意图

A

## 支持智能模块的S7-200CPU

EM 241调制解调器模块是一个智能扩充模块，设计成与表A-45所示的S7-200 CPU一起进行工作。

表 A-45 EM 241调制解调器模块与S7-200 CPU的兼容性

CPU	说明
CPU 222版本1.10或更高版本	CPU 222 DC/DC/DC和CPU 222 AC/DC/继电器
CPU 224版本1.10或更高版本	CPU 224 DC/DC/DC和CPU 224 AC/DC/继电器
CPU 226版本1.10或更高版本	CPU 226 DC/DC/DC和CPU 226 AC/DC/继电器
CPU 226XM版本1.10或更高版本	CPU 226XM DC/DC/DC和CPU 226XM AC/DC/继电器

## 安装EM 241

请按照下列步骤安装EM 241:

1. 将EM 241压进DIN导轨中，并插入带状电缆。
2. 连接来自CPU传感器电源或外部电源的24 VDC，并将接地端子连接到系统地线。
3. 将电话线插入到RJ11插孔中。
4. 按照表A-46设置国家代码开关。在接通CPU的电源之前，必须对该开关进行设置，以便读取正确的国家代码。
5. 接通CPU的电源。绿色MG（模块正常）灯应点亮。

EM 241现在即可进行通讯。

表 A-46 EM 241所支持的国家代码

代码	国家	电信标准
00	澳大利亚	ACA TS-002
01	奥地利	CTR21
02	比利时	CTR21
05	加拿大	IC CS03
06	中国	GB3482
08	丹麦	CTR21
09	芬兰	CTR21
10	法国	CTR21
11	德国	CTR21
12	希腊	CTR21
16	爱尔兰	CTR21
18	意大利	CTR21
22	卢森堡	CTR21
25	荷兰	CTR21
26	新西兰	
27	挪威	CTR21
30	葡萄牙	CTR21
34	西班牙	CTR21
35	瑞典	CTR21
36	瑞士	CTR21
38	英国	CTR21
39	美国	FCC Part 68

## RJ11插口

图A-29显示了RJ11插口的具体细节。对其它标准的电话连接器，您可使用相应的适配器。详细信息请参见适配器连接器文档。

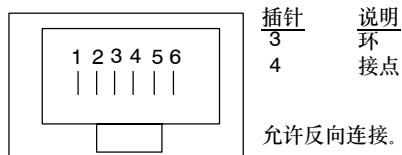


图 A-29 RJ11插头的外观

### 当心

电话线上的雷电浪涌或其它无法预料的高电压都有可能损坏EM 241调制解调器模块。

请使用商用电话线电涌保护器，例如通常销售的用于个人计算机调制解调器防护的产品。电涌保护器在保护EM 241调制解调器模块时，自己也可能受损坏。请选择具有直接指示器的电涌保护器，该指示器可显示其正常工作。

请定期检查电涌保护器，以确保继续保护EM 241调制解调器模块。

## EM 253位控模块规范

表 A-47 EM 253位控模块订购号

订购号	扩展型号	EM输入	EM输出	可移动连接器
6ES7 253-1AA22-0XA0	EM 253位控模块	-	8 <sup>1</sup>	是

<sup>1</sup> 8个Q输出用作移动功能的逻辑控制，但不能直接控制任何外部信号。

表 A-48 EM 253位控模块通用规范

订购号	模块名称与说明	尺寸 (毫米) (W x H x D)	重量	损耗	+5 VDC	VDC要求 +24 VDC
6ES7 253-1AA22-0XA0	EM253位控模块	71.2 x 80 x 62	0.190 kg	2.5 W	190毫安	参见下文

表 A-49 EM 253位控模块规范

常规	6ES7 253-1AA22-0XA0
输入性能	
输入数量	5点
输入类型	汇点/源点 (IEC 类型1汇点, 不包括ZP)
输入电压	
允许的最大持续 STP、RPS、LMT+、LMT- ZP	30 VDC 30 VDC, 20 毫安, 最大值
电涌 (所有输入)	35 VDC, 0.5秒
额定值 STP、RPS、LMT+、LMT- ZP	24 VDC, 4 毫安, 额定 24 VDC, 15 毫安, 额定
逻辑“1”信号 (最小值) STP、RPS、LMT+、LMT- ZP	15 VDC, 2.5 毫安, 最小值 3 VDC, 8.0 毫安, 最小值
逻辑“0”信号 (最大值) STP、RPS、LMT+、LMT- ZP	5 VDC, 1 毫安, 最大值 1 VDC, 1 毫安, 最大值
绝缘 (现场到逻辑电路)	
光学绝缘 (电化)	500 VAC, 1分钟
绝缘组	1点, STP、RPS和ZP 2点, LMT+和LMT-
输入延迟时间	
STP、RPS、LMT+、LMT- ZP (可数脉冲宽度)	0.2毫秒至12.8毫秒, 用户可选 2微秒 最小值
邻近传感器的两条导线的连接 (Bero)	
允许的泄漏电流	1毫安, 最大值
电缆长度	
未屏蔽 STP、RPS、LMT+、LMT- ZP	30米 不建议使用
屏蔽 STP、RPS、LMT+、LMT- ZP	100米 10米
可同时输入的数量	
55摄氏度	5

表 A-49 EM 253位控模块规范

常规	6ES7 253-1AA22-0XA0	
<b>输出性能</b>		
集成输出的数量	6点（4信号）	
输出类型 P0+、P0-、P1+、P1- P0、P1、DIS、CLR	RS422/485驱动程序 漏极开路	
输出电压 P0、P1、RS-422驱动程序，差分输出电压 开路 光耦合器二极管，具有200Ω 系列电阻 100Ω 负载 54Ω 负载 P0、P1、DIS、CLR漏极开路 建议电压，开路 允许电压，开路 反向电流 On 状态电阻 Off 状态泄漏电流，30 VDC T1的内部上拉电阻、漏极输出	3.5 V，典型 2.8V，最小值 1.5V，最小值 1.0V，最小值 5VDC，供模块使用 30VDC <sup>1</sup> 50毫安，最大值 15Ω 最大值 10 μA，最大值 3.3K Ω <sup>2</sup>	
输出电流 输出组的数量 输出ON的数量（最大值） 每点的泄漏电流 P0、P1、DIS、CLR 过载保护	1 6 10 μA，最大值 否	
绝缘（现场到逻辑电路） 光学绝缘（电化）	500 VAC，1分钟	
输出延迟 DIS、CLR: Off到On/On到Off	30微秒，最大值	
脉冲失真 P0、P1、输出、RS-422驱动程序、100Ω 外部 负载 P0、P1输出、漏极开路、5 V/470Ω 外部 负载	75毫微秒，最大值 300毫微秒，最大值	
切换频率 P0+、P0-、P1+、P1-、P0和P1	200千赫兹	
电缆长度 未屏蔽 屏蔽	不建议使用 10米	
<b>电源</b>		
L+ 电源电压 逻辑电源输出 L+电源电流对 5 VDC负载 负载电流 0毫安（无负载） 200毫安（额定负载）	11至30 VDC +5 VDC +/- 10%，200 毫安，最大值	
	<u>12 VDC输入</u> 120毫安 300毫安	<u>24 VDC输入</u> 70毫安 130毫安
绝缘 L+电源到逻辑电路 L+电源到输入 L+电源到输出	500 VAC，1分钟 500 VAC，1分钟 无	
反极性	L+输入和+5V输出均通过二极管进行保护。 将正电压接入到与输出点连接有关的任何M端子上都将导致潜在的破坏性电流。	

<sup>1</sup> 运行5VDC以上的漏极开路输出，将使射频辐射超过允许的限制。可能需要对系统或布线采取射频辐射防护措施。

<sup>2</sup> 根据脉冲接收装置和电缆，加装一个外部上拉电阻，可改善脉冲信号的质量和抗扰性。

## 支持智能模块的S7-200CPU

EM 253位控模块是一个智能扩充模块，设计成与表A-50所示的S7-200 CPU一起工作。

表 A-50 EM 253位控模块与S7-200 CPU的兼容性

CPU	说明
CPU 222版本1.10或更高版本	CPU 222 DC/DC/DC和CPU 222 AC/DC/继电器
CPU 224版本1.10或更高版本	CPU 224 DC/DC/DC和CPU 224 AC/DC/继电器
CPU 226版本1.10或更高版本	CPU 226 DC/DC/DC和CPU 226 AC/DC/继电器
CPU 226XM版本1.10或更高版本	CPU 226XM DC/DC/DC和CPU 226XM AC/DC/继电器

## EM 253位控模块状态LED

位控模块的状态LED显示在表A-51中。

表 A-51 位控模块状态LED

本地I/O	LED	颜色	功能说明
-	MF	红色	当模块检测到严重错误时照亮。
-	MG	绿色	当没有任何模块故障时照亮，检测到配置错误时以1赫兹的频率闪烁
-	PWR	绿色	当模块L+和M端子加上24 VDC电源时照亮
输入	STP	绿色	当“停止”输入打开时照亮
输入	RPS	绿色	当参考点开关输入打开时照亮
输入	ZP	绿色	当零脉冲输入打开时照亮
输入	LMT-	绿色	当负限制输入打开时照亮
输入	LMT+	绿色	当正限制输入打开时照亮
输出	P0	绿色	当P0输出发出脉冲时照亮
输出	P1	绿色	当P1输出发出脉冲时或当该输出指示正向运动时照亮
输出	DIS	绿色	当DIS输出激活时照亮
输出	CLR	绿色	当清除漂移计数器输出激活时照亮

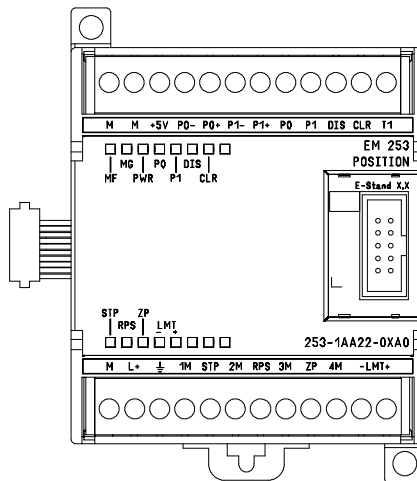


图 A-30 EM 253位控模块

A

## 布线图

在下列示意图中，端子并没有都按次序排列。关于端子排列，请参见图A-30。

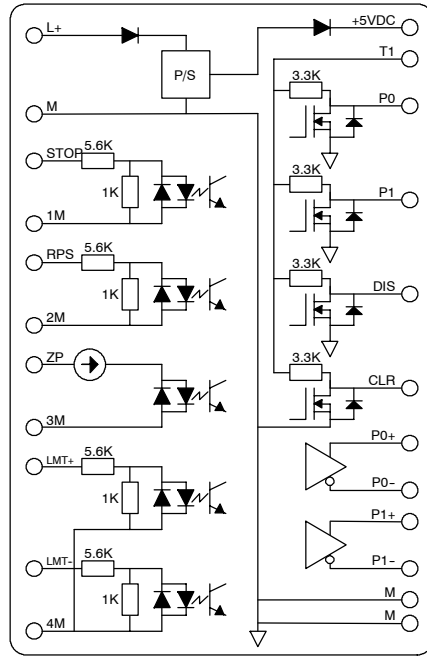
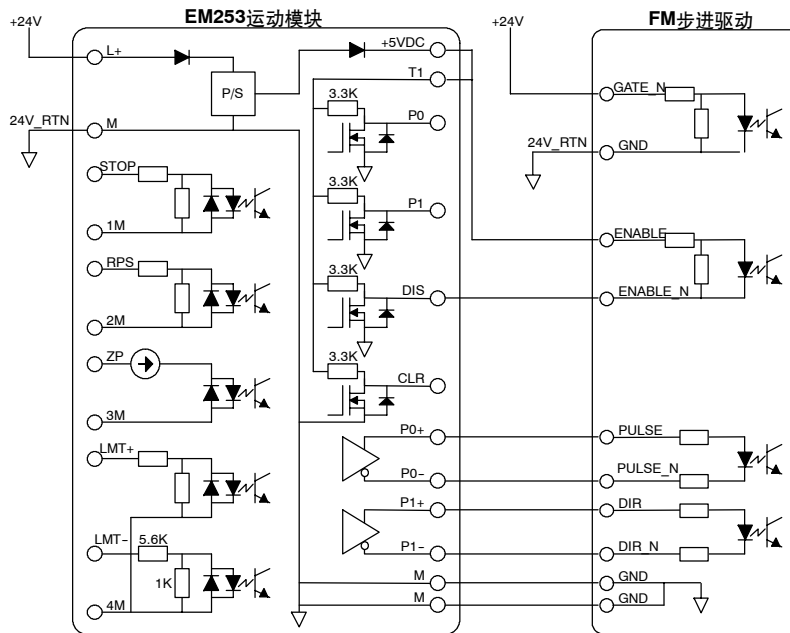


图 A-31 EM 253位控模块输入和输出的内部示意图



端子未按次序排列。  
关于端子排列，请参见图A-30。

图 A-32 EM 253位控模块与SIMATIC FM步进驱动的连接

A

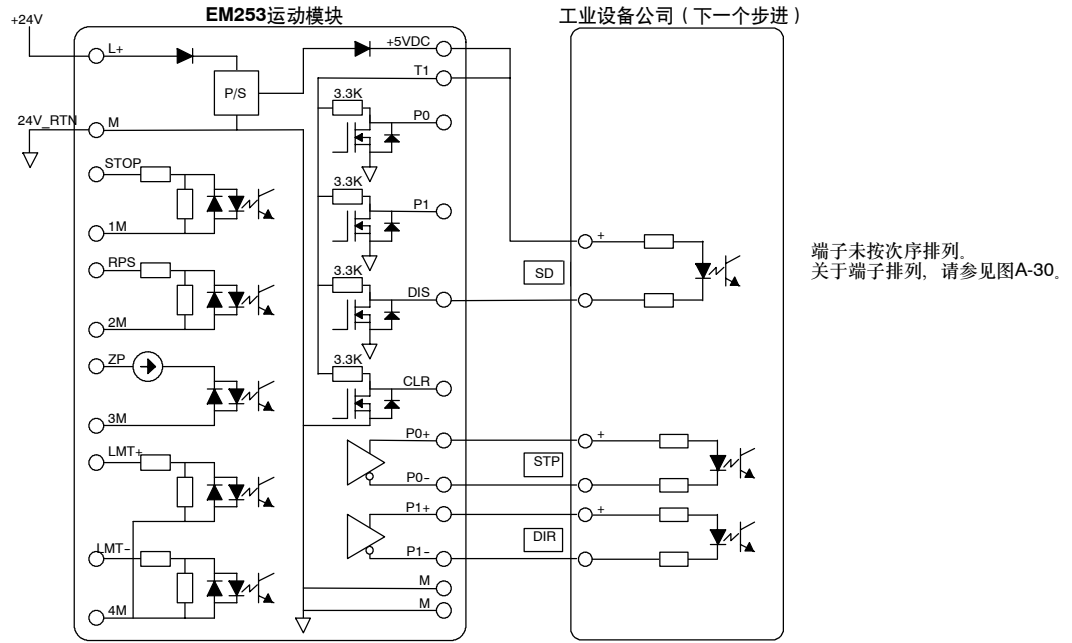


图 A-33 EM 253位控模块与工业设备公司（下一个步进）的连接

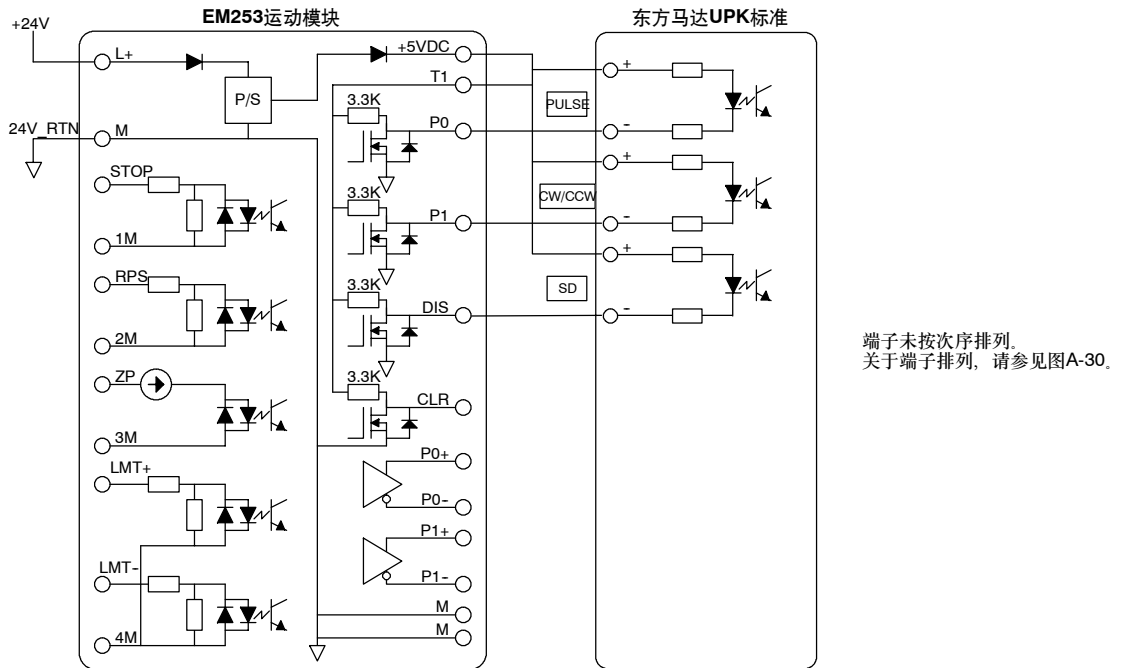
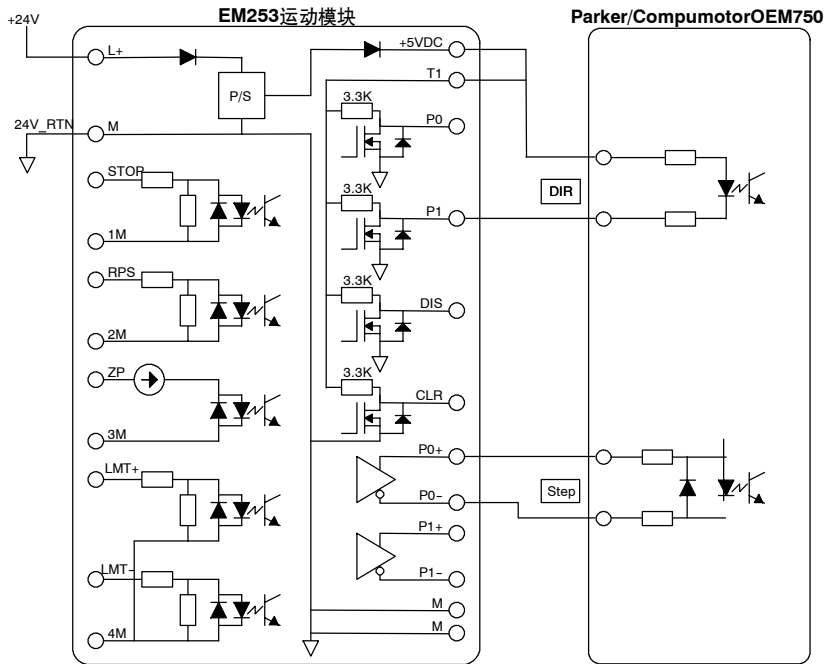


图 A-34 EM 253位控模块与定向马达UPK标准的连接

A



端子未按次序排列。  
关于端子排列，请参见图A-30。

图 A-35 EM 253位控模块与Parker/Compumotor OEM 750的连接



## ( CP 243-1 ) 以太网模块规范

表 A-52 ( CP 243-1 ) 以太网模块订购号

订购号	扩充模块	EM输入	EM输出	可移动连接器
6GK7 243-1EX00-OXE0	( CP 243-1 ) 以太网模块	-	8 <sup>1</sup>	否

<sup>1</sup> 8个Q输出用作以太网功能的逻辑控制，但不能直接控制任何外部信号。

表 A-53 ( CP 243-1 ) 以太网模块通用规范

订购号	模块名称与说明	尺寸 (毫米) ( W x H x D )	重量	损耗	+5 VDC	VDC要求 来自AS接口
6GK7 243-1EX00-OXE0	( CP 243-1 ) 以太网模块	71.2 x 80 x 62	大约 150g	1.75W	55毫安	60毫安

表 A-54 ( CP 243-1 ) 以太网模块规范

常规	6GK7 243-1EX00-OXE0
传输率	10兆位/秒和100兆位/秒
闪存大小	1MB
SDRAM内存大小	8MB
接口 连接工业以太网 ( 10/100MB/秒 )	8插针RJ45插口
输入电压	20.4到28.8 VDC
最大连接数	每个 ( CP 243-1 ) 以太网模块最多8个S7连接 ( XPUT/XGET和READ/WRITE ) 加上1个STEP7-Micro/WIN连接 <sup>2</sup>
启动时间或复位之后重新启动的时间	大约10秒
用户数据数量	对于客户机: 至多212个字节, 用于XPUT/XGET 对于服务器: 至多222字节用于XGET或READ 至多212字节用于XPUT或WRITE

<sup>2</sup> 每个S7-200 CPU只应连接一个 ( CP 243-1 ) 以太网模块。

( CP 243-1 ) 以太网模块是一个用于将S7-200系统与工业以太网 ( IE ) 相连接的通讯处理器。使用STEP 7 Micro/WIN, 可通过以太网对S7-200进行远程配置、编程或诊断。S7-200也可通过以太网与其他S7-200、S7-300或S7-400控制器进行通讯。它还可与OPC服务器进行通讯。

工业以太网专用于工业领域。它既可与防噪声的工业双绞线 ( ITP ) 技术一起使用, 也可与工业标准的双绞线 ( TP ) 技术一起使用。工业以太网用来实现范围极广的应用场合的具体使用, 例如转换、高速冗余、快速连接和冗余网络。使用 ( CP 243-1 ) 以太网模块, 可使S7-200 PLC与支持以太网的众多现有产品相兼容。

### 支持智能模块的S7-200 CPU

( CP 243-1 ) 以太网模块是一个智能扩充模块, 设计成与表A-45所示的S7-200 CPU一起工作。

表 A-55 ( CP 243-1 ) 以太网模块与S7-200 CPU的兼容性

CPU	说明
CPU 222版本1.10或更高版本	CPU 222 DC/DC/DC和CPU 222 AC/DC/继电器
CPU 224版本1.10或更高版本	CPU 224 DC/DC/DC和CPU 224 AC/DC/继电器
CPU 226版本1.10或更高版本	CPU 226 DC/DC/DC和CPU 226 AC/DC/继电器
CPU 226XM版本1.10或更高版本	CPU 226XM DC/DC/DC和CPU 226XM AC/DC/继电器

A

(CP 243-1) 以太网模块使用固定不变的预置、全世界唯一的MAC地址进行交付。

## 功能

(CP 243-1) 以太网模块可独立处理工业以太网上的数据通讯量。

- 通讯基于TCP/IP进行
- 对于S7-200 CPU与其它S7控制系统或PC之间通过以太网进行的通讯，客户机和服务器均可使用通讯服务。至多可运行8个连接。
- 通过集成S7PC服务器，可以实现许多PC的应用。
- (CP 243-1) 以太网模块允许S7-200编程软件、STEP 7-Micro/WIN通过因特网直接访问S7-200。



以太网

## 配置

您可使用STEP7-Micro/WIN以太网向导来配置(CP 243-1)以太网模块，以便将S7-200 PLC连接到以太网网络。以太网向导将帮助您定义(CP 243-1)以太网模块的参数，并随后将配置指令放置在项目指令文件夹中。为启动以太网向导，可选择工具>以太网向导菜单命令。向导使用下列信息：IP地址、子网掩模、网关地址和通讯连接类型。

## 连接

(CP 243-1) 以太网模块具有下列接头。接头位于前门的外壳下。

- 用于24 VDC电源电压和地线连接的接线盒
- 用于以太网连接的8插针RJ45插口
- 用于I/O总线的插头连接器
- 用于I/O总线的具有插口的集成带状电缆

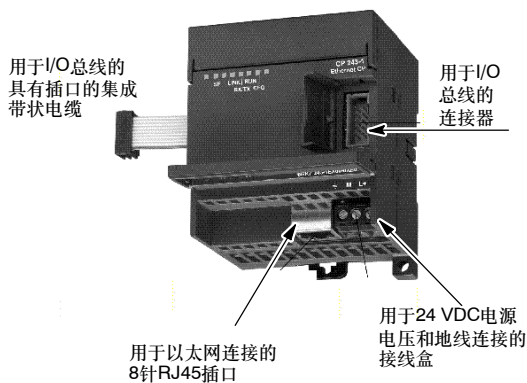


图 A-36 连接(CP 243-1)以太网模块

## 附加信息

对于有关(CP 243-1)以太网模块的更详细信息，请参见用于工业以太网的SIMATICNET CP 243-1通讯处理器技术手册。

## ( CP 243-1 IT ) 互联网模块规范

表 A-56 ( CP 243-1 IT ) 互联网模块订购号

订购号	扩充模块	EM输入	EM输出	可移动连接器
6GK7 243-1GX00-0XE0	( CP 243-1 IT ) 互联网模块	-	8 <sup>1</sup>	否

<sup>1</sup> 8个Q输出用作IT功能的逻辑控制，但不能直接控制任何外部信号。

表 A-57 ( CP 243-1 IT ) 互联网模块通用规范

订购号	模块名称与说明	尺寸 (毫米) ( W x H x D )	重量	损耗	+5 VDC	VDC要求 来自AS接口
6GK7 243-1GX00-0XE0	( CP 243-1 IT ) 互联网模块	71.2 x 80 x 62	大约 150g	1.75W	55毫安	60毫安

表 A-58 ( CP 243-1 IT ) 互联网模块规范

常规	6GK7 243-1GX00-0XE0
传输速度	10MB/秒和100MB/秒
闪存大小	8MB用于 ( CP 243-1 IT ) 互联网模块固化程序的ROM, 8MB用于文件系统的RAM
SDRAM内存大小	16MB
文件系统闪存的保修寿命	1百万次写或删除操作
接口 连接工业以太网 ( 10/100MB/秒 )	8插针RJ45插口
输入电压	20.4到28.8 VDC
最大连接数	每个 ( CP 243-1 IT ) 以太网模块最多8个S7连接 ( XPUT/XGET和READ/WRITE ) 加上1个STEP7-Micro/WIN连接 <sup>2</sup>
IT连接的最大数目	1个用于FTP服务器 1个用于FTP客户机 1个用于电子邮件客户机 4个用于HTTP连接
启动时间或复位之后重新启动的时间	大约10秒
用户数据数量	客户机: 至多212个字节, 用于XPUT/XGET 服务器: 至多222字节用于XGET或READ 至多212字节用于XPUT或WRITE
电子信箱大小, 最大值	1024个字符
文件系统: 包含文件大小和驱动器名称的路径长度 文件名长度 目录嵌套深度	最多254个字符 最多99个字符 最多49个
可用服务器端口: HTTP FTP命令通道 FTP数据通道, 用于FTP服务器 S7连接建立 S7服务器	80 21 3100至3199 102 3000至3008

<sup>2</sup> 每个S7-200 CPU只应连接一个 ( CP 243-1 IT ) 互联网模块。

(CP 243-1 IT) 以太网模块是一个用于将S7-200系统与工业以太网 (IE) 相连接的通讯处理器。使用STEP 7 Micro/WIN, 可通过以太网对S7-200进行远程配置、编程或诊断。S7-200也可通过以太网与其他S7-200、S7-300或S7-400控制器进行通讯。它还可与OPC服务器进行通讯。

(CP 243-1 IT) 互联网模块的IT功能将构成监控的基础, 必要时, 也是通过联网PC的WEB浏览器操作自动化系统的基础。系统的诊断消息可通过电子邮件进行发送。使用IT功能, 可与其它计算机和控制系统方便地交换整个文件。

工业以太网属于过程控制层的网络, 也是SIMATIC NET开放式通讯系统的单元层。物理上, 工业以太网是基于屏蔽、同轴线、双绞线电缆的一种电气网络, 也是一种光纤电缆的光网络。工业以太网符合国际标准IEEE 802.3。

## 支持智能模块的S7-200 CPU

(CP 243-1 IT) 互联网模块是一种智能扩充模块, 设计成与如表A-45所示的S7-200 CPU一起工作。

表 A-59 (CP 243-1 IT) 以太网模块与S7-200 CPU的兼容性

CPU	说明
CPU 222版本1.10或更高版本	CPU 222 DC/DC/DC和CPU 222 AC/DC/继电器
CPU 224版本1.10或更高版本	CPU 224 DC/DC/DC和CPU 224 AC/DC/继电器
CPU 226版本1.10或更高版本	CPU 226 DC/DC/DC和CPU 226 AC/DC/继电器
CPU 226XM版本1.10或更高版本	CPU 226XM DC/DC/DC和CPU 226XM AC/DC/继电器

(CP 243-1 IT) 互联网模块具有下列特性:

- (CPU 243-1 IT) 互联网模块与 (CP 243-1) 以太网模块完全兼容。为 (CP 243-1) 以太网模块编写的用户程序也可在 (CP 243-1 IT) 互联网模块上运行。

(CP 243-1 IT) 以太网模块使用固定不变的预置、全世界唯一的MAC地址进行交付。



### 提示

每个S7-200 CPU只应连接一个 (CP 243-1 IT) 互联网模块。如果连接的 (CP 243-1 IT) 互联网模块多于一个, 则S7-200 CPU有可能不能正常运行。

### 功能

(CP 243-1 IT) 互联网模块提供了下列功能:

- 基于TCP/IP进行S7通讯
- IT通讯
- 配置
- 监视程序计时器
- 对要编址的MAC地址 (48位数值) 进行预置的能力



互联网

## 配置

您可使用STEP 7-Micro/WIN互联网向导来配置 (CP 243-1 IT) 互联网模块, 以便将S7-200 PLC连接到以太网/互联网网络。(CP 243-1 IT) 互联网模块具有附加的web服务器功能, 它可使用互联网向导进行配置。为启动互联网向导, 选择工具>互联网向导菜单命令。

## 连接

(CP 243-1 IT) 互联网模块具有下列接头。接头位于正门的外壳下。

- 用于24 VDC电源电压和地线连接的接线盒
- 用于以太网连接的8插针RJ45插口
- 用于I/O总线的插头连接器
- 用于I/O总线的具有插口的集成带状电缆

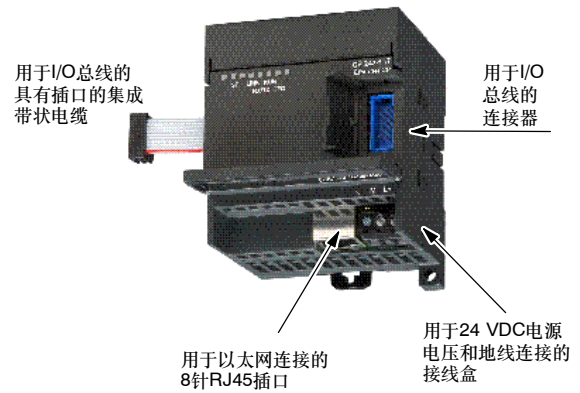


图 A-37 连接 (CP 243-1 IT) 互联网模块

## 附加信息

对于有关 (CP 243-1 IT) 互联网模块的更详细信息, 请参见关于工业以太网和信息技术的SIMATICNET CP 243-1 IT 通讯处理器技术手册。

## ( CP 243-2 ) AS接口模块规范

表 A-60 ( CP 243-2 ) AS接口模块订购号

订购号	扩展型号	EM输入	EM输出	可移动连接器
6GK7 243-2AX01-0XA0	( CP 243-2 ) AS接口模块	8个数字位和 8个模拟位	8个数字位和 8个模拟位	是

表 A-61 ( CP 243-2 ) AS接口模块通用规范

订购号	模块名称与说明	尺寸 (毫米) ( W x H x D )	重量	损耗	+5 VDC	VDC要求 来自AS接口
6GK7 243-2AX01-0XA0	( CP 243-2 ) AS接口模块	71 x 80 x 62	大约 250g	3.7W	220毫安	100毫安

表 A-62 ( CP 243-2 ) AS接口模块规范

常规	6GK7 243-2AX01-0XA0
周期时间	对于31个从属装置为5毫秒 对于使用扩展编址模式的62个AS-I从属装置为10毫秒
配置	前面板上的设置按钮, 或使用总的配置命令 (参见CP 243-2 AS-I接口主设备手册中的AS-I命令的说明)
所支持的AS-I主设备配置文件	M1e
AS-I电缆的附件	通过S7-200接线盒。允许电流从端子1流入到端子3或从端子2流入到端子4, 其最大电流为3A。
地址范围	具有8个数字输入和8个数字输出的一个数字模块, 以及 具有8个模拟输入和8个模拟输出的一个模拟模块

### 特性

您最多可同时在S7-200上运行两个AS接口模块, 这将明显增加可用的数字和模拟输入/输出的数量 (在每个CP上的AS接口中, 最多可有124个数字输入/124个数字输出)。由于具有通过触摸按钮进行配置的能力, 可减少设置时间。通过CP和所有已连接从属装置的状态显示, 以及对AS接口的主电压进行监控, 在万一出现故障时, LED可减少停机时间。

AS接口模块具有下列特性:

- 支持模拟模块
- 支持所有主设备功能, 并允许与最多62个AS接口从属装置进行连接
- 前面板上的LED将对运行状态和所连接从属装置的可用性进行显示。
- 前面板上的LED将对错误进行显示 (包含AS接口电压错误、配置错误)
- 两个端子将允许直接连接AS接口电缆。
- 两个按钮将显示从属装置的状态信息、开关的操作模式, 并选用现有配置作为SET配置。

您可使用STEP 7-Micro/WIN AS-i向导来配置 ( CP 243-2 ) AS接口模块。AS接口向导将帮助您在配置中使用来自AS接口网络中的数据。为启动AS-i向导, 可选择工具>AS-i向导菜单命令。



AS

## 操作

在S7-200的过程图像中，AS接口模块将占据一个数字输入字节（状态字节）、一个数字输出字节（控制字节）、8个模拟输入和8个模拟输出字。AS接口模块使用了两个逻辑模块位置。您可使用状态和控制字节来设置使用用户程序的AS接口模块的模式。根据其模式的不同，AS接口既可存储AS接口模块的I/O数据、诊断数值，也可以激活S7-200模拟地址区中的主设备调用（例如，改变从属装置地址）。

所有已连接的AS接口从属装置都必须在触摸按钮时进行配置。不需要对CP再作进一步的配置。



### 当心

当使用AS接口模块时，必须禁用CPU中的模拟过滤。

如果没有禁用CPU中的模拟过滤，则数字点的数据将被破坏，且错误条件将不会作为模拟字中的位数值进行返回。

请确保禁用CPU中的模拟过滤。

## 功能

CP 243-2是M1主设备级的AS接口主设备，这意味着它将支持所有指定的功能。这样就有可能通过双精度型的地址分配在AS接口上运行多达31个的数字从属装置（A-B）。CP 243-2可设置为两种不同的模式：

- 标准模式：访问AS接口从属装置的I/O数据
- 扩展模式：主设备调用（例如，写参数）或诊断数值请求

## 连接

AS接口模块具有下列接头：

- 两个接头用于连接AS接口模块电缆（内部桥接）
- 一个接头用于功能性接地

端子均位于如图A-38所示的前面板上的盖子下。

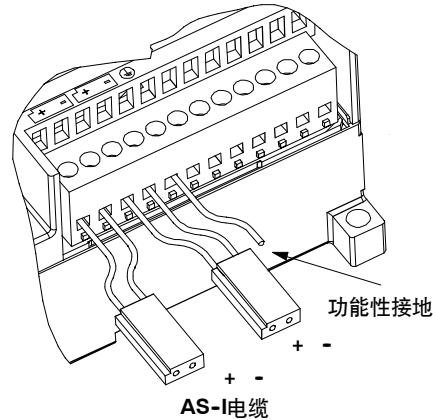


图 A-38 连接AS接口模块电缆

### 当心

AS接口模块接点的负载容量最大为3A。如果AS接口模块电缆上的电流超过该数值，AS接口不能与AS-I电缆构成回路，而必须用一根单独的电缆进行连接（此时，只应使用AS接口模块的一对端子）。AS接口必须通过地线端子连接到接地导体上。



### 提示

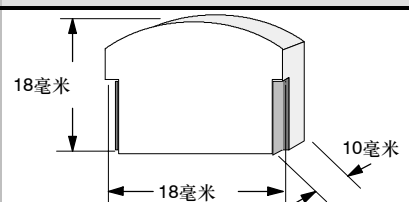
AS接口模块具有用于功能性接地的接头。该接头应以尽可能小的阻抗连接到PE导体上。

## 附加信息

关于CP 243-2 AS接口主设备的详细信息，请参见SIMATIC NET CP 243-2 AS接口主设备手册。

## 供选用的夹头

夹头	说明	订购号
内存磁带	内存磁带存储: 程序、数据和配置	6ES7 291-8GE20-0XA0
带有电池的实时时钟	时钟磁带精度: 2分钟/月, 在25°C时, 7分钟/月, 在0°C至55°C时	6ES7 297-1AA20-0XA0
电池夹头	电池夹头 (数据保持时间): 200日, 典型	6ES7 291-8BA20-0XA0

通用性能		尺寸
电池	3V, 30毫安小时, Renata CR 1025	
尺寸	9.9 毫米 x 2.5 毫米	
类型	锂<0.6g	

内存磁带将存储所有CPU的完整程序和数据块 (CPU 221、CPU 222、CPU 224、CPU 226和CPU 226XM)。

## I/O扩展电缆

通用性能 (6ES7 290-6AA20-0XA0)	
电缆长度	0.8米
重量	25g
连接器类型	10插针 带状

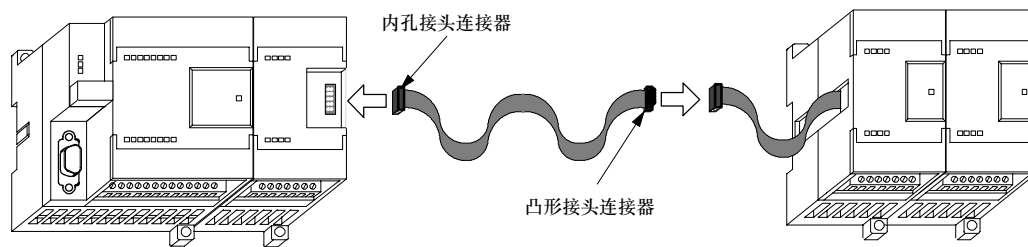


图 A-39 I/O扩展电缆的典型安装



### 提示

在CPU/扩充模块链中只允许有一根扩展电缆。



## RS-232/PPI多台主设备电缆和USB/PPI多台主设备电缆

表 A-63 RS-232/PPI多台主设备电缆和USB/PPI多台主设备电缆的规范

说明 订购号	S7-200 RS-232/PPI 多台主设备电缆 6ES7 901-3CB30-0XA0	S7-200 USB/PPI 多台主设备电缆 6ES7-901-3DB30-0XA0
<b>通用特征</b>		
电源电压	14.4至28.8 VDC	14.4至28.8 VDC
24V额定电源下的供电电流	60毫安 RMS, 最大值	50毫安 RMS, 最大值
方向改变延迟: RS-232停止位边沿接收RS-485 传输禁用	-	-
绝缘	RS-485至RS-232: 500 VDC	RS-485至USB: 500 VDC
<b>RS-485侧电气特征</b>		
共模电压范围	-7 V至+12 V, 1秒钟, 3 V RMS 连续	-7 V至+12 V, 1秒钟, 3 V RMS 连续
接收端输入阻抗	5.4 K $\Omega$ 最小值, 包含终端	5.4 K $\Omega$ 最小值, 包含终端
终端/偏差	10K $\Omega$ 至+5 V, 在B点、PROFIBUS 插针3 10K $\Omega$ 至GND, 在A点、PROFIBUS 插针8	10K $\Omega$ 至+5 V, 在B点、PROFIBUS 插针3 10K $\Omega$ 至GND, 在A点、PROFIBUS 插针8
接收端阈值/敏感度	+/-0.2 V, 60 mV 典型滞后	+/-0.2 V, 60 mV 典型滞后
传输端差分输出电压	2 V 最小值, 当 $R_L=100 \Omega$ 时, 1.5 V 最小值, 当 $R_L=54 \Omega$ 时	2 V 最小值, 当 $R_L=100 \Omega$ 时, 1.5 V 最小值, 当 $R_L=54 \Omega$ 时
<b>RS-232侧电气特征</b>		
接收端输入阻抗	3K $\Omega$ 最小值	-
接收端阈值/敏感度	0.8 V 最低, 2.4 V 最高 0.5 V 典型滞后	-
传输端输出电压	+/- 5 V 最小值, 当 $R_L=3K \Omega$ 时	-
<b>USB侧电气特征</b>		
全速 (12MB/s)、操作接口设备 (HID)		
5V下的供电电流	-	50毫安 最大值
掉电电流	-	400 $\mu$ A 最大值

### 特性

在S7-200 RS-232/PPI多台主设备电缆的工厂设置中具有使用STEP 7-Micro/WIN 3.2 Service Pack 4 (或稍后版本) 编程软件包进行性能优化的设置。该电缆的工厂设置不同于PC/PPI电缆。请参考图1来配置您实际应用中的电缆。

通过将开关5设置为PPI/自由端口设置, 然后选择所需要的波特率, 您可配置S7-200 RS-232/PPI多台主设备电缆, 以便与PC/PPI电缆进行同样的操作, 并与任意版本的STEP 7-Micro/WIN编程软件包兼容。

USB电缆需要使用STEP 7-Micro/WIN 3.2 Service Pack 4 (或稍后版本) 编程软件包进行操作。



#### 提示

关于PC/PPI电缆的信息, 请参阅第三版的S7-200 PLC手册订购号 (6ES7 298-8FA22-8BH0)。

# A

## S7-200 RS-232/PPI多台主设备电缆

表 A-64 S7-200 RS-232/PPI多台主设备电缆-RS-485到RS-232本地模式连接器的外接插针

RS-485连接器的外接插针		RS-232本地连接器的外接插针	
插针号	信号说明	插针号	信号说明
1	无连接	1	数据载波检测 (DCD) (未用)
2	24V回流 (RS-485逻辑地线)	2	接收数据 (RD) (从PC/PPI电缆输出)
3	信号B (Rx/D/TxD+)	3	传输数据 (TD) (输入PC/PPI电缆)
4	RTS (TTL电平)	4	数据端子准备就绪 (DTR) <sup>1</sup>
5	无连接	5	地线 (RS-232逻辑地线)
6	无连接	6	数据设置准备就绪 (DSR) <sup>1</sup>
7	24V电源	7	请求发送 (RTS) (未用)
8	信号A (Rx/D/TxD-)	8	清除发送 (CTS) (未用)
9	协议选择	9	铃声指示器 (RI) (未用)

<sup>1</sup> 插针4和6在内部连接。

表A-65 S7-200 RS-232/PPI多台主设备电缆-RS-485到RS-232远程模式连接器的外接插针

RS-485连接器的外接插针		RS-232远程连接器的外接插针 <sup>1</sup>	
插针号	信号说明	插针号	信号说明
1	无连接	1	数据载波检测 (DCD) (未用)
2	24V回流 (RS-485逻辑地线)	2	接收数据 (RD) (输入到PC/PPI电缆)
3	信号B (Rx/D/TxD+)	3	传输数据 (TD) (从PC/PPI电缆输出)
4	RTS (TTL电平)	4	数据端子准备就绪 (DTR) <sup>2</sup>
5	无连接	5	地线 (RS-232逻辑地线)
6	无连接	6	数据设置准备就绪 (DSR) <sup>2</sup>
7	24V电源	7	请求发送 (RTS) (从PC/PPI电缆输出)
8	信号A (Rx/D/TxD-)	8	清除发送 (CTS) (未用)
9	协议选择	9	铃声指示器 (RI) (未用)

<sup>1</sup> 调制解调器需要进行内孔接头与凸形接头的转换和9针插针到25针插针的转换。

<sup>2</sup> 插针4和6在内部连接。

### 使用具有STEP 7-Micro/WIN的S7-200 RS-232/PPI多台主设备电缆替换PC/PPI电缆或自由端口操作

为直接连接到个人计算机:

- 可设置PPI/自由端口模式 (开关5=0)
- 设置波特率 (开关1、2、3)
- 设置本地模式 (开关6=0)。本地设置与将PC/PPI电缆设置为DCE的方式相同。
- 设置第11位 (开关7=0)

为连接调制解调器:

- 可设置PPI/自由端口模式 (开关5=0)
- 设置波特率 (开关1、2、3)
- 设置远程模式 (开关6=1)。远程设置与将PC/PPI电缆设置为DTE的方式相同。
- 设置第10位或11位 (开关7)，以便调制解调器的每个字符的位数相匹配。

使用具有STEP 7-Micro/WIN3.2 Service Pack 4（或稍后版本）的S7-200 RS-232/PPI多台主设备电缆

为直接连接到个人计算机:

- 设置PPI模式（开关5=1）
- 设置本地模式（开关6=0）

为连接调制解调器:

- 设置PPI模式（开关5=1）
- 设置远程模式（开关6=1）

图A-40显示了S7-200 RS-232/PPI多台主设备电缆的尺寸、标签和LED。

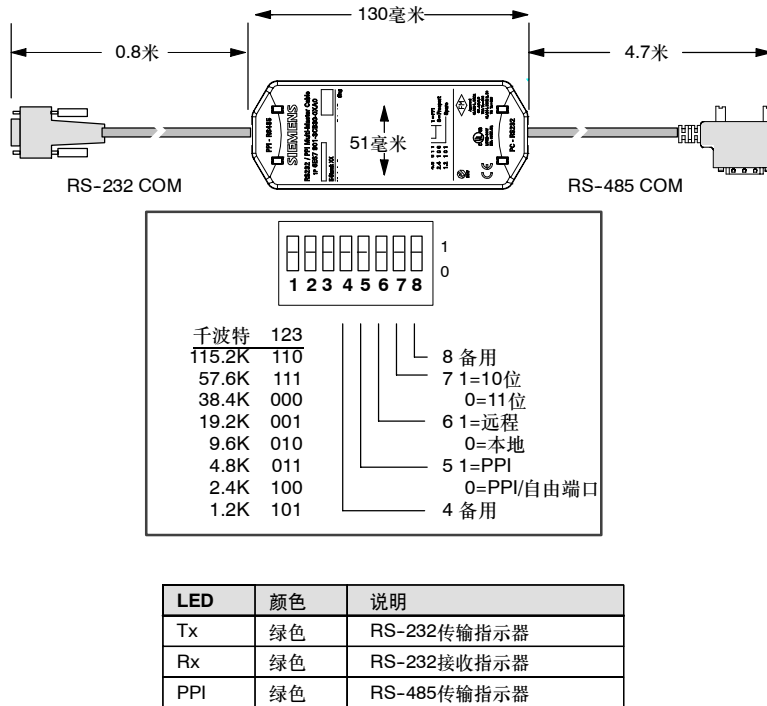


图 A-40 S7-200 RS-232/PPI多台主设备电缆的尺寸、标签和LED



## S7-200 USB/PPI多台主设备电缆

为使用USB电缆，必须已安装STEP 7-Micro/WIN3.2 Service Pack 4（或更新版本）。USB电缆不支持自由端口通讯。

表 A-66 S7-200 USB/PPI多台主设备电缆-RS-485到USB系列“A”连接器的外接插针

RS-485连接器的外接插针		USB连接器的外接插针	
插针号	信号说明	插针号	信号说明
1	无连接	1	USB-DataP
2	24V回流（RS-485逻辑地线）	2	USB-DataM
3	信号B（Rx/D/TxD+）	3	USB 5V
4	RTS（TTL电平）	4	USB逻辑地线
5	无连接		
6	无连接		
7	24V电源		
8	信号A（Rx/D/TxD-）		
9	协议选择（低=10位）		

图A-41显示了S7-200 USB/PPI多台主设备电缆的尺寸和LED。

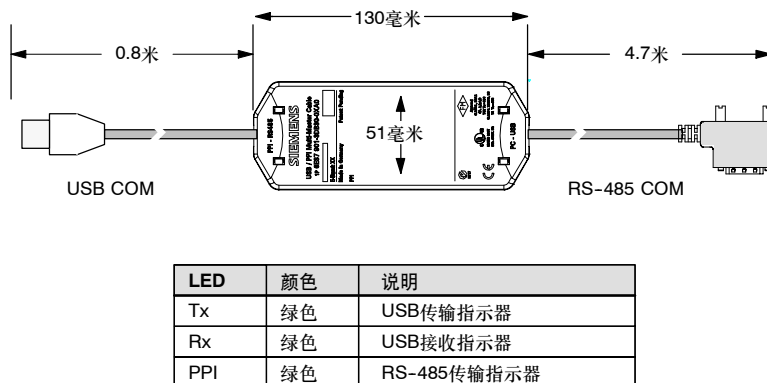


图 A-41 S7-200 USB/PPI多台主设备电缆的尺寸和LED

## 输入模拟程序

订购号	8位置模拟程序 6ES7 274-1XF00-0XA0	14位置模拟程序 6ES7 274-1XH00-0XA0	24位置模拟程序 6ES7 274-1XK00-0XA0
尺寸 (L x W x D)	61 x 36 x 22 毫米	91 x 36 x 22 毫米	147 x 36 x 25 毫米
重量	0.02kg	0.03kg	0.04kg
点	8	14	24

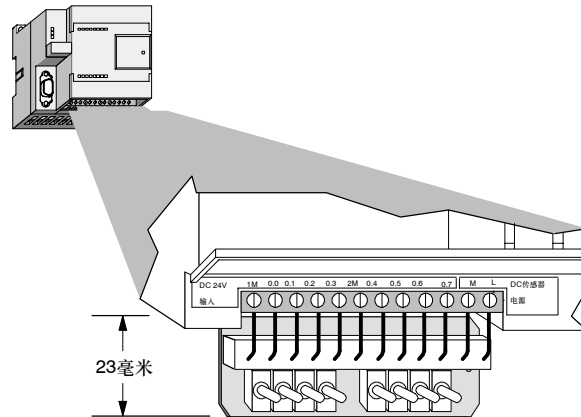


图 A-42 输入模拟程序的安装



### 当心

这些输入模拟程序均不允许在Class I DIV 2或Class I Zone 2危险场所下使用。开关有潜在出现火花的危险。切勿在Class I DIV 2或Class I Zone 2危险场所下使用输入模拟程序。

**A**

# 计算电源预算



S7-200 CPU有内部电源，为CPU自己、任何扩充模块和其它24 VDC用户电源供电。使用下列信息作为指南以确定S7-200 CPU可以为用户配置提供多少功率（或电流）。

## 电源要求

每个S7-200 CPU都提供5 VDC和24 VDC电源:

- 每个CPU有24 VDC传感器供电，可以为本地输入点或为扩充模块上的继电器线圈提供24 VDC。如果对于24 VDC的电源要求超出CPU的电源预算，可以添加外部24 VDC电源以为扩充模块提供24 VDC。必须手动连接24 VDC电源到输入点或继电器线圈。
- 当扩充模块连接时，CPU也为扩充模块提供5 VDC电源。如果扩充模块的5 VDC电源要求超出CPU的电源预算，必须拆除扩充模块直到需求在电源预算之内。

附录A中的规定提供关于CPU电源预算和扩充模块电源要求的信息。



### 提示

如果超出了CPU电源预算，可能无法连接CPU允许的最大数目的模块。



### 危险

与S7-200 DC传感器电源并行连接外部24 VDC电源会导致冲突，因为每个电源都寻求建立自己首选的输出电压电平。

这种冲突的结果会使一个或两个电源缩短使用寿命或立即故障，随后对PLC系统进行不可预知的操作。不可预知的操作可以导致人员死亡或严重伤害，并且 / 或者损坏设备。

S7-200 DC传感器电源和任何外部电源应该为不同点供电。公用电源的单独连接是允许的。

## 计算电源要求实例

表B-1显示计算电源要求的实例，其S7-200包含下列内容：

- S7-200 CPU 224交流/直流/继电器
- 每个有3个EM 223，8个直流输入/8个继电器输出
- 每个有1个EM 221，8个直流输入

该装置共有46个输入和34个输出。

在此例中S7-200 CPU为扩充模块提供足够的5 VDC电流，但没有为所有输入和扩充继电器线圈从传感器提供足够的24 VDC电流。I/O需要400毫安，而S7-200 CPU只提供280毫安。此装置需要至少120毫安的24 VDC附加的电源以操作所有包含的24 VDC输入和输出。

表 B-1 实例配置的电源预算计算

CPU电源预算	5 VDC	24 VDC
CPU 224交流/直流/继电器	660毫安	280毫安
减		
系统要求	5 VDC	24 VDC
CPU 224, 14个输入		14 * 4毫安 = 56毫安
3 EM 223, 需要5 V电源	3 * 80毫安 = 240毫安	
1 EM 221, 需要5V电源	1 * 30毫安 = 30毫安	
3 EM 223, 每个8个输入		3 * 8 * 4毫安 = 96毫安
3 EM 223, 每个8个继电器线圈		3 * 8 * 9毫安 = 216毫安
1 EM 221, 每个8个输入		8 * 4毫安 = 32毫安
总要求	270毫安	400毫安
等于		
电流平衡	5 VDC	24 VDC
总电流平衡	390毫安	[120毫安]



### 计算电源需要

使用下表确定S7-200 CPU可以为用户配置提供多少功率（或电流）。关于用户CPU型号的电源预算和扩充模块的电源要求请参考附录A。

电源预算	5 VDC	24 VDC

减

系统要求	5 VDC	24 VDC
总要求		

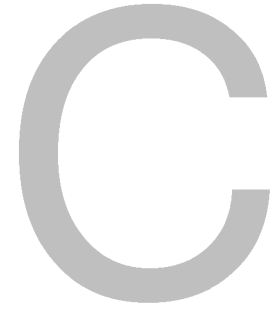
等于

电流平衡	5 VDC	24 VDC
总电流平衡		



**B**

# 错误代码



提供关于错误代码的信息帮助用户识别S7-200 CPU的问题。

## 本章内容

严重错误代码和讯息 .....	424
运行时间程序问题 .....	425
编译规则违反 .....	426

## 严重错误代码和讯息

严重错误将导致S7-200停止执行程序。根据错误的严重程度，严重错误将致使S7-200失去执行某些或所有函数的能力。处理严重错误的目的是使S7-200恢复到安全状态，S7-200可据此对关于现有错误条件的询问作出反应。

当检测到严重错误时，S7-200完成下列任务：

- 切换到STOP（停止）模式
- 点亮“系统故障LED”和“停止LED”
- 断开输出

S7-200保持此条件，直到严重错误被更正。要查看错误代码，从主菜单栏选择**PLC > Information（信息）**菜单命令。表C-1提供列表描述可以从S7-200读取的严重错误代码。

表 C-1 从S7-200读取的严重错误代码和讯息

错误代码	描述
0000	没有严重错误显示
0001	用户程序检验和出错
0002	编译的梯形程序检验和出错
0003	扫描监视程序超时出错
0004	内部EEPROM故障
0005	用户程序上的内部EEPROM检验和出错
0006	配置（SDB0）参数内部EEPROM检验和出错
0007	强制数据的内部EEPROM检验和出错
0008	默认输出表数值内部EEPROM检验和出错
0009	用户数据，DB1的内部EEPROM检验和出错
000A	内存磁带故障
000B	用户程序内存磁带检验和出错。
000C	配置（SDB0）参数内存磁带检验和出错
000D	强制数据的内存磁带检验和出错
000E	默认输出表数值内存磁带检验和出错
000F	用户数据，DB1的内存磁带检验和出错
0010	内部软件出错
0011 <sup>1</sup>	比较接点间接地址出错
0012 <sup>1</sup>	比较接点非法浮点数数值
0013	内存磁带空白，或者S7-200不理解程序
0014 <sup>1</sup>	比较接点范围出错

<sup>1</sup> 比较触点出错是产生严重和非严重错误条件的仅有的错误。产生非严重错误条件的原因是保存出错的程序地址。

## 运行系统程序问题

在正常执行程序期间，用户程序可以创建非严重错误条件（诸如编址错误）。在这种情况下，S7-200产生非严重运行系统错误代码。表C-2列出非严重错误代码的描述。

表 C-2 运行系统程序问题

错误代码	描述
0000	没有严重错误显示；无错
0001	在执行HDEF框之前，HSC框启用
0002	输入中断的冲突分配到已分配给HSC的点
0003	输入的冲突分配给已分配给输入中断或其它HSC的HSC
0004	在中断例行程序中尝试执行ENI、DISI、SPA或HDEF指令
0005	在完成第一次之前，尝试执行具有相同编号的第二次HSC/PLS（中断例行程序中的HSC/PLS与在主程序中的HSC/PLS冲突）
0006	间接地址出错
0007	TODW（目的时间写）或TODR（目的时间读）数据出错
0008	最大用户子例行程序嵌套层超出
0009	在端口0同时执行XMT/RCV指令
000A	通过执行另一个HDEF指令尝试为相同的HSC重定义HSC
000B	在端口1同时执行XMT/RCV指令
000C	时钟磁带不为TODR、TODW或通讯的存取显示
000D	尝试当它为激活时重定义脉冲输出
000E	PTO图形段的数目设置为0
000F	比较接点指令中的非法数字值
0091	范围出错（带有地址信息）：检查操作数范围
0092	指令的计数域中出错（带有计数信息）：校验最大计数大小
0094	写入至带有地址信息的非易失内存的范围出错
009A	当在用户中断中时尝试切换到“自由端口”模式
009B	非法的索引（指定起始位置数值0的字符串操作）

## 编译规则违反

当下载程序时，S7-200编译程序。如果S7-200检测到程序违反编译规则（诸如非法的指令），S7-200中止下载并产生非严重的编译规则错误代码。表C-3列出由违反编译规则产生的错误代码描述。

表 C-3 编译规则违反

错误代码	编译出错（非严重）
0080	程序太大而无法编译；减少程序大小
0081	堆栈下溢；将程序段拆分为多个程序段。
0082	非法的指令；检查指令助记符。
0083	丢失MEND或指令不允许在主程序中：添加MEND指令，或删除不正确指令。
0084	保留
0085	丢失FOR；添加FOR指令或删除NEXT指令。
0086	丢失NEXT；添加NEXT指令或删除FOR指令。
0087	丢失标签（LBL、INT、SBR）；添加合适的标签。
0088	丢失RET或指令不允许在子例行程序中：添加RET到子例行程序的结尾或删除不正确的指令。
0089	丢失RETI或指令不允许在中断例行程序中：添加RETI到中断例行程序的结尾或删除不正确的指令。
008A	保留
008B	非法JMP到或从SCR程序段
008C	复制标签（LBL、INT、SBR）；重命名标签之一。
008D	非法标签（LBL、INT、SBR）；确保允许的标签数没有超出。
0090	非法的参数；校验指令允许的参数。
0091	范围出错（带有地址信息）；检查操作数范围。
0092	指令的计数域中错误（带有计数信息）；校验最大计数大小。
0093	FOR/NEXT嵌套层超出。
0095	丢失LSCR指令（载入SCR）
0096	丢失SCRE指令（SCR结束）或在SCRE指令前不接受指令
0097	用户程序包含未编号的和编号的EV/ED指令
0098	在RUN（运行）模式中的非法编辑（尝试在程序用未编号的EV/ED指令编辑）
0099	太多的隐藏程序段（“隐藏”指令）
009B	非法的索引（指定起始位置数值0的字符串操作）
009C	最大指令长度超出



# 特殊内存（SM）位

特殊内存位提供各种状态和控制功能，也用作一种在S7-200和用户程序之间通讯信息的方式。特殊内存位可以被用作位、字节、字或双字。

## 本章内容

SMB0: 状态位 .....	428
SMB1: 状态位 .....	428
SMB2: 自由端口接收字符 .....	429
SMB3: 自由端口奇偶校验错误 .....	429
SMB4: 队列溢出 .....	429
SMB5: I/O状态 .....	430
SMB6: CPU标识寄存器 .....	430
SMB7: 保留 .....	430
SMB8到SMB21: I/O模块标识号和错误寄存器 .....	431
SMW22到SMW26: 扫描时间 .....	432
SMB28和SMB29: 模拟调整 .....	432
SMB30和SMB130: 自由端口控制寄存器 .....	432
SMB31和SMW32: 永久性内存（EEPROM）写控制 .....	433
SMB34和SMB35: 用于定时中断的时间间隔寄存器 .....	433
SMB36到SMB65: HSC0、HSC1和HSC2寄存器 .....	433
SMB66到SMB85: PTO/PWM寄存器 .....	435
SMB86到SMB94, SMB186到SMB194: 接收讯息控制 .....	436
SMW98: 扩展I/O总线出错 .....	437
SMB130: 自由端口控制寄存器（参见SMB30） .....	437
SMB131到SMB165: HSC3、HSC4和HSC5寄存器 .....	437
SMB166到SMB185: PTO0、PTO1配置文件定义表 .....	438
SMB186到SMB194: 接收讯息控制（参见SMB86到SMB94） .....	438
SMB200到SMB549: 智能模块状态 .....	439

## SMB0: 状态位

如表D-1所述，SMB0包含八个状态位，它们在每个扫描循环的结束由S7-200更新。

表 D-1 特殊内存字节SMB0 (SM0.0到SM0.7)

SM位	说明 (只读)
SM0.0	此位始终接通。
SM0.1	此位在首次扫描周期接通。一个用途是调用初始化子例行程序。
SM0.2	如果保留性数据丢失，此位在一个扫描循环内变为接通。此位可以用作错误内存位或用作调用特殊启动顺序的机制。
SM0.3	当从上电条件进入RUN (运行) 模式时，此位变为一个扫描循环而接通。此位可以用作在开始操作前提供机器预热时间。
SM0.4	此位提供时钟脉冲，对于1分钟的工作循环时间，30秒接通，30秒断开。它提供容易使用的延迟，或者1分钟时钟脉冲。
SM0.5	此位提供时钟脉冲，对于1秒的工作循环时间，0.5秒接通，0.5秒断开。它提供容易使用的延迟，或者1秒钟时钟脉冲。
SM0.6	此位是扫描循环时钟，在一个扫描循环接通，然后在下一个扫描循环断开。此位可以用作扫描计数器输入。
SM0.7	此位反映了模式开关的位置 (断开是TERM (终端) 位置，接通是RUN (运行) 位置)。如果当开关在RUN (运行) 位置时使用此位启用自由端口模式，与编程设备的正常通讯可以通过切换到TERM (终端) 位置来启用。

## SMB1: 状态位

如表D-2所述，SMB1包含各种电位出错指示器。这些位在执行时间由指令置位和重设。

表 D-2 特殊内存字节SMB1 (SM1.0到SM1.7)

SM位	说明 (只读)
SM1.0	当操作结果为零时，此位通过执行某些指令而接通。
SM1.1	当引起溢出或当检测到非法的数字值时，此位通过执行某些指令而接通。
SM1.2	当通过算术运算产生负结果时，此位接通。
SM1.3	当尝试除以零时，此位接通。
SM1.4	当“添加到表格”指令试图填满表格时，此位接通。
SM1.5	当LIFO或FIFO指令尝试从空表读时，此位接通。
SM1.6	当进行尝试转换非BCD码数值到二进制时，此位接通。
SM1.7	当ASCII数值无法转换为有效的十六进制数值时，此位接通。



## SMB2: 自由端口接收字符

SMB2是自由端口接收字符缓冲区。如表D-3中所述, 在自由端口模式下接收的每个字符放在此位置中, 以从梯形程序方便地存取。



### 提示

SMB2和SMB3在端口0和端口1之间共享。当接收端口0上的字符导致执行附加在那个事件(中断事件8)的中断例行程序时, SMB2包含端口0上接收的字符, 而SMB3包含该字符的奇偶校验状态。当接收端口1上的字符导致执行附加在那个事件(中断事件25)的中断例行程序时, SMB2包含端口1上接收的字符, 而SMB3包含该字符的奇偶校验状态。

表 D-3 特殊内存字节SMB2

SM字节	说明(只读)
SMB2	此字节包含在自由端口通讯期间从端口0或端口1接收的每个字符。

## SMB3: 自由端口奇偶校验错误

SMB3用于自由端口模式并包含奇偶校验错误位, 当在接收的字符上检测到奇偶校验出错时该位就被置位。如表D-4所示, 当检测到奇偶校验出错时, SM3.0接通。使用此位放弃讯息。

表 D-4 特殊内存字节SMB3 (SM3.0到SM3.7)

SM位	说明(只读)
SM3.0	来自端口0或端口1的奇偶校验错误 (0 = 无错; 1 = 检测到错误)
SM3.1到SM3.7	保留

## SMB4: 队列溢出

如表D-5所述, SMB4 包含中断队列溢出位, 一个状态指示器显示中断是启用还是禁用, 以及发送器闲置内存位。队列溢出位指示中断发生率大于可以被处理的率, 或中断用全局中断禁用指令禁用。

表 D-5 特殊内存字节SMB4 (SM4.0到SM4.7)

SM位	说明(只读)
SM4.0 <sup>1</sup>	当通讯中断队列溢出时, 此位接通。
SM4.1 <sup>1</sup>	当输入中断队列溢出时, 此位接通。
SM4.2 <sup>1</sup>	当定时中断队列溢出时, 此位接通。
SM4.3	当检测到运行系统程序问题时, 此位接通。
SM4.4	此位反映全局中断启用状态。当中断启用时, 它接通。
SM4.5	当发送器闲置时(端口0), 此位接通。
SM4.6	当发送器闲置时(端口1), 此位接通。
SM4.7	当有东西被强制时, 此位接通。

<sup>1</sup> 在中断例行程序中只使用状态位4.0、4.1和4.2。当队列被清空时, 这些状态位重置, 并且控制返回到主程序。

## SMB5: I/O状态

如表D-6所述，SMB5 包含关于在I/O系统中检测出的出错条件的状态位。这些位提供检测出的I/O错误总览。

表 D-6 特殊内存字节SMB5 (SM5.0到SM5.7)

SM位	说明 (只读)
SM5.0	如果显示任何I/O错误，此位接通。
SM5.1	如果太多的数字I/O点连接到I/O总线，此位接通。
SM5.2	如果太多的模拟I/O点连接到I/O总线，此位接通。
SM5.3	如果太多的智能I/O模块连接到I/O总线，此位接通。
SM5.4到 SM5.7	保留。

## SMB6: CPU标识寄存器

如表D-7所述，SMB6是S7-200 CPU的标识寄存器。SM6.4到SM6.7识别S7-200 CPU的型号。SM6.0到SM6.3保留作为将来使用。

表 D-7 特殊内存字节SMB6

SM位	说明 (只读)								
格式	<div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center; margin-right: 10px;">           MSB 7         </div> <div style="text-align: center; margin-right: 10px;">           LSB 0         </div> <div style="border: 1px solid black; padding: 2px;"> <table style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 15px; height: 15px;">x</td> <td style="width: 15px; height: 15px;">x</td> <td style="width: 15px; height: 15px;">x</td> <td style="width: 15px; height: 15px;">x</td> <td style="width: 15px; height: 15px;">r</td> <td style="width: 15px; height: 15px;">r</td> <td style="width: 15px; height: 15px;">r</td> <td style="width: 15px; height: 15px;">r</td> </tr> </table> </div> <div style="margin-left: 20px;">CPU标识寄存器</div> </div>	x	x	x	x	r	r	r	r
x	x	x	x	r	r	r	r		
SM6.0到 SM6.3	保留								
SM6.4到 SM6.7	xxxx =   0000 =   CPU 222 0010 =   CPU 224 0110 =   CPU 221 1001 =   CPU 226/CPU 226XM								

## SMB7: 保留

SMB7保留作为将来使用。



## SMW22到SMW26: 扫描时间

如表D-9所述, SMW22、SMW24和SMW26提供扫描时间信息: 最小扫描时间、最大扫描时间和最后扫描时间(以毫秒为单位)。

表 D-9 特殊内存字SMW22到SMW26

SM字	说明(只读)
SMW22	最后扫描循环的扫描时间(以毫秒为单位)
SMW24	从进入RUN(运行)模式开始记录的最小扫描时间(以毫秒为单位)
SMW26	从进入RUN(运行)模式开始记录的最大扫描时间(以毫秒为单位)

## SMB28和SMB29: 模拟调整

如表D-10所述, SMB28保持表示模拟调整0位置的数字值。SMB29保持表示模拟调整1位置的数字值。

表 D-10 特殊内存字节SMB28和SMB29

SM字节	说明(只读)
SMB28	此字节存储以模拟调整0输入的数值。在每次停止/运行扫描中, 此数值更新一次。
SMB29	此字节存储以模拟调整1输入的数值。在每次停止/运行扫描中, 此数值更新一次。

## SMB30和SMB130: 自由端口控制寄存器

SMB30控制端口0的自由端口通讯; SMB130控制端口1的自由端口通讯。可以读和写入至SMB30和SMB130。如表D-11所述, 这些字节为自由端口操作配置各自的通讯端口, 并提供自由端口或系统协议支持的选择。

表 D-11 特殊内存字节SMB30

端口0	端口1	说明								
SMB30的格式	SMB130的格式	自由端口模式控制字节 <div style="text-align: center;">           MSB            7  <table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">p</td> <td style="padding: 2px;">p</td> <td style="padding: 2px;">d</td> <td style="padding: 2px;">b</td> <td style="padding: 2px;">b</td> <td style="padding: 2px;">b</td> <td style="padding: 2px;">m</td> <td style="padding: 2px;">m</td> </tr> </table>             LSB            0         </div>	p	p	d	b	b	b	m	m
p	p	d	b	b	b	m	m			
SM30.0和SM30.1	SM130.0和SM130.1	mm: 协议选项 00=点至点接口协议(PPI/从模式) 01=自由端口协议 10=PPI/主模式 11=保留(默认到PPI/从模式) 注意: 当选择代码mm=10(PPI主设备), S7-200将成为网络上的主设备, 允许NETR和NETW指令执行。在PPI模式中位2到7忽略。								
SM30.2到SM30.4	SM130.2到SM130.4	bbb: 自由端口波特率 000=38,400波特 001=19,200波特 010=9,600波特 011=4,800波特 100=2,400波特 101=1,200波特 110=115,200波特 111=57,600波特								
SM30.5	SM130.5	d: 每个字符的数据位 0=每个字符8位 1=每个字符7位								
SM30.6和SM30.7	SM130.6和SM130.7	pp: 奇偶校验选择 00=无奇偶校验 01=偶数校验 10=偶数校验 11=奇数校验								



表 D-14 特殊内存字节SMB36到SMD62

SM字节	说明
SM36.0到SM36.4	保留
SM36.5	HSC0当前计数方向状态位: 1=向上计数
SM36.6	HSC0当前值等于预设值状态位: 1=相等
SM36.7	HSC0当前值大于预设值状态位: 1=大于
SM37.0	“重设”的激活级别控制位: 0=重设为现用高, 1=重设为现用低
SM37.1	保留
SM37.2	求积计数器的计数率选择: 0 = 4x计数率; 1 = 1x计数率
SM37.3	HSC0方向控制位: 1 = 向上计数
SM37.4	HSC0更新方向: 1 = 更新方向
SM37.5	HSC0更新预设值: 1 = 写新预设值到HSC0预置
SM37.6	HSC0更新当前值: 1 = 写新当前值到HSC0当前
SM37.7	HSC0启用位: 1 = 启用
SMD38	HSC0新当前值
SMD42	HSC0新预设值
SM46.0到SM46.4	保留
SM46.5	HSC1当前计数方向状态位: 1 = 向上计数
SM46.6	HSC1当前值等于预设值状态位: 1 = 相等
SM46.7	HSC1当前值大于预设值状态位: 1 = 大于
SM47.0	HSC1重设的激活级别控制位: 0 = 现用高, 1 = 现用低
SM47.1	HSC1启动的激活级别控制位: 0 = 现用高, 1 = 现用低
SM47.2	HSC1求积计数率选择: 0 = 4x率, 1 = 1x率
SM47.3	HSC1方向控制位: 1 = 向上计数
SM47.4	HSC1更新方向: 1 = 更新方向
SM47.5	HSC1更新预设值: 1 = 写新预设值到HSC1预置
SM47.6	HSC1更新当前值: 1 = 写新当前值到HSC1当前
SM47.7	HSC1启用位: 1 = 启用
SMD48	HSC1新当前值
SMD52	HSC1新预设值
SM56.0到SM56.4	保留
SM56.5	HSC2当前计数方向状态位: 1 = 向上计数
SM56.6	HSC2当前值等于预设值状态位: 1 = 相等
SM56.7	HSC2当前值大于预设值状态位: 1 = 大于
SM57.0	HSC2重设的激活级别控制位: 0 = 现用高, 1 = 现用低
SM57.1	HSC2启动的激活级别控制位: 0 = 现用高, 1 = 现用低
SM57.2	HSC2求积计数率选择: 0 = 4x率, 1 = 1x率
SM57.3	HSC2方向控制位: 1 = 向上计数
SM57.4	HSC2更新方向: 1 = 更新方向
SM57.5	HSC2更新预设值: 1 = 写新预设值到HSC2预置
SM57.6	HSC2更新当前值: 1 = 写新当前值到HSC2当前
SM57.7	HSC2启用位: 1 = 启用
SMD58	HSC2新当前值
SMD62	HSC2新预设值

## SMB66到SMB85: PTO/PWM寄存器

如表D-15所述, SMB66到SMB85用于监视和控制脉冲串输出和脉冲宽度调制功能。关于这些位的完整说明参见第6章中的脉冲输出高速输出指令的信息。

表 D-15 特殊内存字节SMB66到SMB85

SM字节	说明
SM66.0到SM66.3	保留
SM66.4	PTO0配置文件中止: 0 = 无错, 1 = 由于德耳塔计算出错而中止
SM66.5	PTO0配置文件中止: 0 = 没有被用户命令中止, 1 = 被用户命令中止
SM66.6	PTO0管道溢出 (当使用外部配置文件时由系统清除, 否则必须由用户重设): 0 = 无溢出, 1 = 管道溢出
SM66.7	PTO0空闲位: 0 = PTO在进程中, 1 = PTO空闲
SM67.0	PTO0/PWM0更新周期时间数值: 1 = 写新周期时间
SM67.1	PWM0更新时钟脉冲宽度数值: 1 = 写新时钟脉冲宽度
SM67.2	PTO0更新脉冲计数数值: 1 = 写新脉冲计数
SM67.3	PTO0/PWM0时基: 0 = 1微秒 / 刻度, 1 = 1毫秒 / 刻度
SM67.4	同步更新PWM0: 0 = 异步更新, 1 = 同步更新
SM67.5	PTO0操作: 0 = 单段操作 (周期时间和脉冲计数存储在SM内存), 1 = 多段操作 (概要表存储在V内存中)
SM67.6	PTO0/PWM0模式选择: 0 = PTO, 1 = PWM
SM67.7	PTO0/PWM0启用位: 1 = 启用
SMW68	PTO0/PWM0周期时间数值 (时基的2到65,535个单元)
SMW70	PWM0脉冲宽度数值 (时基的0到65,535个单元)
SMD72	PTO0脉冲计数数值 (1到 $2^{32}-1$ );
SM76.0到SM76.3	保留
SM76.4	PTO1配置文件中止: 0 = 无错, 1 = 由于德耳塔计算出错而中止
SM76.5	PTO1配置文件中止: 0 = 没有被用户命令中止, 1 = 被用户命令中止
SM76.6	PTO1管道溢出 (当使用外部配置文件时由系统清除, 否则必须由用户重设): 0 = 无溢出, 1 = 管道溢出
SM76.7	PTO1空闲位: 0 = PTO在进程中, 1 = PTO空闲
SM77.0	PTO1/PWM1更新周期时间数值: 1 = 写新周期时间
SM77.1	PWM1更新时钟脉冲宽度数值: 1 = 写新时钟脉冲宽度
SM77.2	PTO1更新脉冲计数数值: 1 = 写新脉冲计数
SM77.3	PTO1/PWM1时基: 0 = 1微秒 / 刻度, 1 = 1毫秒 / 刻度
SM77.4	同步更新PWM1: 0 = 异步更新, 1 = 同步更新
SM77.5	PTO1操作: 0 = 单段操作 (周期时间和脉冲计数存储在SM内存), 1 = 多段操作 (概要表存储在V内存中)
SM77.6	PTO1/PWM1模式选择: 0 = PTO, 1 = PWM
SM77.7	PTO1/PWM1启用位: 1 = 启用
SMW78	PTO1/PWM1周期时间数值 (时基的2到65,535个单元)
SMW80	PWM1脉冲宽度数值 (时基的0到65,535个单元)
SMD82	PTO1脉冲计数数值 (1到 $2^{32}-1$ );

## SMB86到SMB94, SMB186到SMB194: 接收讯息控制

如表D-16所示, SMB86到SMB94和SMB186到SMB194用于控制和读“接收讯息”指令的状态。

表 D-16 特殊内存字节SMB86到SMB94, SMB186到SMB194

端口0	端口1	说明																											
SMB86	SMB186	接收讯息状态字节 <table style="margin-left: 20px; border-collapse: collapse;"> <tr> <td style="text-align: right;">MSB</td> <td colspan="7"></td> <td style="text-align: left;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="text-align: center;">0</td> </tr> <tr> <td></td> <td style="text-align: center;">n</td> <td style="text-align: center;">r</td> <td style="text-align: center;">e</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">t</td> <td style="text-align: center;">c</td> <td style="text-align: center;">p</td> </tr> </table> <p>n: 1 = 接收被用户禁用命令终止的讯息  r: 1 = 接收终止的讯息: 输入参数出错或丢失启动或结束条件  e: 1 = 结束字符已接收  t: 1 = 接收讯息已终止: 计时器时间到  c: 1 = 接收讯息已终止: 最大字符计数已完成  p 1= 因为奇偶校验出错接收讯息终止</p>	MSB								LSB	7								0		n	r	e	0	0	t	c	p
MSB								LSB																					
7								0																					
	n	r	e	0	0	t	c	p																					
SMB87	SMB187	接收讯息控制字节 <table style="margin-left: 20px; border-collapse: collapse;"> <tr> <td style="text-align: right;">MSB</td> <td colspan="7"></td> <td style="text-align: left;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="border: 1px solid black; width: 20px; height: 20px;"></td> <td style="text-align: center;">0</td> </tr> <tr> <td></td> <td style="text-align: center;">en</td> <td style="text-align: center;">sc</td> <td style="text-align: center;">ec</td> <td style="text-align: center;">il</td> <td style="text-align: center;">c/m</td> <td style="text-align: center;">tmr</td> <td style="text-align: center;">bk</td> <td></td> </tr> </table> <p>en: 0 =接收讯息功能禁用。  1 =接收讯息功能启用。  启用 / 禁用接收讯息位在每次执行RCV指令时检查。  sc: 0 =忽略SMB88或SMB188。  1 =使用SMB88或SMB188的数值检测讯息的开始。  ec: 0 =忽略SMB89或SMB189。  1 =使用SMB89或SMB189的数值检测讯息的结束。  il: 0 =忽略SMW90或SMW190。  1 =使用SMW90或SMW190的数值检测空闲行条件。  c/m: 0 =计时器是字符间的计时器。  1 =计时器是讯息计时器。  tmr: 0 =忽略SMW92或SMW192。  1 =如果SMW92或SMW192中的时间间隔超出, 终止接收。  bk: 0 =忽略断开条件。  1 =使用断开条件作为讯息检测的开始。</p>	MSB								LSB	7								0		en	sc	ec	il	c/m	tmr	bk	
MSB								LSB																					
7								0																					
	en	sc	ec	il	c/m	tmr	bk																						
SMB88	SMB188	讯息字符的开始																											
SMB89	SMB189	讯息字符的结束																											
SMW90	SMW190	以毫秒为单位的空闲行时间周期。在空闲行时间到期后接收的第一个字符是新讯息的开始。																											
SMW92	SMW192	以毫秒为单位的字符间 / 讯息计时器超时数值。如果时间周期超出, 接收讯息终止。																											
SMB94	SMB194	接收的最大字符数 (1到255个字节)。 注意: 此范围必须设置到期望的最大缓冲区大小, 即使不使用字符计数讯息终端。																											



## SMW98: 扩展I/O总线出错

如表D-17所述, SMW98给用户关于扩展I/O总线上出错数目的信息。

表 D-17 特殊内存字节SMW98

SM字节	说明
SMW98	每次在扩展I/O总线上检测到奇偶校验出错, 此位置增加。一旦上电它就已清除, 并且可以由用户清除。

## SMB130: 自由端口控制寄存器 (参见 SMB30)

参考表D-11。

## SMB131到SMB165: HSC3、HSC4和HSC5寄存器

如表D-18所述, SMB131到SMB165用于监视和控制高速计数器HSC3、HSC4和HSC5的运行。

表 D-18 特殊内存字节SMB131到SMB165

SM字节	说明
SMB131到SMB135	保留
SM136.0到SM136.4	保留
SM136.5	HSC3当前计数方向状态位: 1 = 向上计数
SM136.6	HSC3当前值等于预设值状态位: 1 = 相等
SM136.7	HSC3当前值大于预设值状态位: 1 = 大于
SM137.0到SM137.2	保留
SM137.3	HSC3方向控制位: 1 = 向上计数
SM137.4	HSC3更新方向: 1 = 更新方向
SM137.5	HSC3更新预设值: 1 = 写新预设值到HSC3预置
SM137.6	HSC3更新当前值: 1 = 写新当前值到HSC3当前
SM137.7	HSC3启用位: 1 = 启用
SMD138	HSC3新当前值
SMD142	HSC3新预设值
SM146.0到SM146.4	保留
SM146.5	HSC4当前计数方向状态位: 1 = 向上计数
SM146.6	HSC4当前值等于预设值状态位: 1 = 相等
SM146.7	HSC4当前值大于预设值状态位: 1 = 大于
SM147.0	“重设”的激活级别控制位: 0 = 重设为现用高, 1 = 重设为现用低
SM147.1	保留
SM147.2	求积计数器的计数率选择: 0 = 4x计数率, 1 = 1x计数率
SM147.3	HSC4方向控制位: 1 = 向上计数
SM147.4	HSC4更新方向: 1 = 更新方向
SM147.5	HSC4更新预设值: 1 = 写新预设值到HSC4预置
SM147.6	HSC4更新当前值: 1 = 写新当前值到HSC4当前
SM147.7	HSC4启用位: 1 = 启用
SMD148	HSC4新当前值
SMD152	HSC4新预设值

表 D-18 特殊内存字节SMB131到SMB165 (续)

SM字节	说明
SM156.0到SM156.4	保留
SM156.5	HSC5当前计数方向状态位: 1 = 向上计数
SM156.6	HSC5当前值等于预设值状态位: 1 = 相等
SM156.7	HSC5当前值大于预设值状态位: 1 = 大于
SM157.0到SM157.2	保留
SM157.3	HSC5方向控制位: 1 = 向上计数
SM157.4	HSC5更新方向: 1 = 更新方向
SM157.5	HSC5更新预设值: 1 = 写新预设值到HSC5预置
SM157.6	HSC5更新当前值: 1 = 写新当前值到HSC5当前
SM157.7	HSC5启用位: 1 = 启用
SMD158	HSC5新当前值
SMD162	HSC5新预设值

## SMB166到SMB185: PTO0、PTO1配置文件定义表

如表D-19所示, SMB166到SMB185用于显示现用配置文件步骤数和概要表在V内存中的地址。

表 D-19 特殊内存字节SMB166到SMB185

SM字节	说明
SMB166	PTO0现用配置文件步骤的当前条目编号
SMB167	保留
SMW168	作为从V0偏移量的PTO0概要表的V内存地址。
SMB170到SMB175	保留
SMB176	PTO1现用配置文件步骤的当前条目编号
SMB177	保留
SMW178	作为V0偏移量的PTO1概要表的V内存地址。
SMB180到SMB185	保留

## SMB186到SMB194: 接收讯息控制 (参见 SMB86 到 SMB94 )

参考表D-16。

## SMB200到SMB549: 智能模块状态

如表D-20所示, SMB200到SMB549为智能扩充模块(诸如EM 277 PROFIBUS-DP模块)提供的信息保留。关于模块如何使用SMB200到SMB549的信息,参考附录A指定模块的规定。

智能模块SM区的分配方式对于V2.2及以后的版本均有所不同。

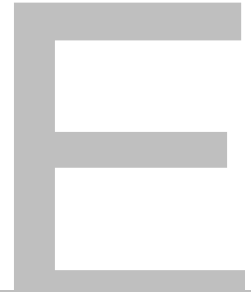
对于固化程序版本号1.2之前的S7-200 CPU,智能模块必须安装在紧靠CPU的位置,以确保兼容性。

表 D-20 特殊存储器字节SMB200至SMB549

特殊存储器字节SMB200至SMB549							
插槽0中的智能模块	插槽1中的智能模块	插槽2中的智能模块	插槽3中的智能模块	插槽4中的智能模块	插槽5中的智能模块	插槽6中的智能模块	说明
SMB200至SMB215	SMB250至SMB265	SMB300至SMB315	SMB350至SMB365	SMB400至SMB415	SMB450至SMB465	SMB500至SMB515	模块名 (16个ASCII字符)
SMB216至SMB219	SMB266至SMB269	SMB316至SMB319	SMB366至SMB369	SMB416至SMB419	SMB466至SMB469	SMB516至SMB519	S/W修订号 (4个ASCII字符)
SMW220	SMW270	SMW320	SMW370	SMW420	SMW470	SMW520	错误代码
SMB222至SMB249	SMB272至SMB299	SMB322至SMB349	SMB372至SMB399	SMB422至SMB449	SMB472至SMB499	SMB522至SMB549	专用于特殊模块类型的信息



# S7-200订购号



CPU	订购号
CPU 221 DC/DC/DC 6个输入/4个输出	6ES7 211-0AA22-0XB0
CPU 221 AC/DC/继电器6个输入/4个继电器	6ES7 211-0BA22-0XB0
CPU 222 DC/DC/DC 8个输入/6个输出	6ES7 212-1AB22-0XB0
CPU 222 AC/DC/继电器8个输入/6个继电器	6ES7 212-1BB22-0XB0
CPU 224 DC/DC/DC14个输入/10个输出	6ES7 214-1AD22-0XB0
CPU 224 AC/DC/继电器14个输入/10个继电器	6ES7 214-1BD22-0XB0
CPU226 DC/DC/DC24个输入/16个输出	6ES7 216-2AD22-0XB0
CPU226 AC/DC/继电器24个输入/16个继电器	6ES7 216-2BD22-0XB0
CPU 226XM DC/DC/DC24个输入/16个继电器	6ES7 216-2AF22-0XB0
CPU 226XM AC/DC/继电器24个输入/16个继电器	6ES7 216-2BF22-0XB0
扩充模块	订购号
EM 221 24 VDC 数字8个输入	6ES7 221-1BF22-0XA0
EM 221 数字8个AC输入 ( 8 x 120/230 VAC )	6ES7 221-1EF22-0XA0
EM 221 数字输入16 x 24 VDC	6ES7 221-1BH22-0XA0
EM222 24 VDC 数字8个输出	6ES7 222-1BF22-0XA0
EM 222 数字输出8 x 继电器	6ES7 222-1HF22-0XA0
EM 222 数字8个AC输出 ( 8 x 120/230 VAC )	6ES7 222-1EF22-0XA0
EM 222 数字输出4 x 24 VDC - 5A	6ES7 222-1BD22-0XA0
EM 222数字输出 4 x 继电器 -10A	6ES7 222-1HD22-0XA0
EM 223 24 VDC 数字组合4个输入/4个输出	6ES7 223-1BF22-0XA0
EM 223 24 VDC 数字组合4个输入/4个继电器输出	6ES7 223-1HF22-0XA0
EM 223 24 VDC 数字组合8个输入/8个输出	6ES7 223-1BH22-0XA0
EM 223 24 VDC 数字组合8个输入/8个继电器输出	6ES7 223-1PH22-0XA0
EM 223 24 VDC 数字组合16个输入/16个输出	6ES7 223-1BL22-0XA0
EM 223 24 VDC 数字组合16个输入/16个继电器输出	6ES7 223-1PL22-0XA0
EM 231模拟输入, 4个输入	6ES7 231-0HC22-0XA0
EM 231模拟输入RTD, 2个输入	6ES7 231-7PB22-0XA0
EM 231模拟输入热电偶, 4个输入	6ES7 231-7PD22-0XA0
EM 232模拟输出, 2个输出	6ES7 232-0HB22-0XA0
EM 235模拟组合4个输入/1个输出	6ES7 235-0KD22-0XA0
EM 241调制解调器模块	6ES7 241-1AA22-0XA0
EM253位控模块	6ES7 253-1AA22-0XA0
EM 277 PROFIBUS-DP	6ES7 277-0AA22-0XA0
( CP 243-2 ) AS接口模块	6GK7 243-2AX01-0XA0
( CP 243-1 ) IT互联网模块 ( 光盘上有电子文档 )	6GKY 243-1GX00-0XE0
( CP 243-1 ) 以太网模块 ( 光盘上有电子文档 )	6GKY 243-1EX00-0XE0

盒式磁带和电缆	订购号
MC 291, 32K x 8 EEPROM内存磁带	6ES7 291-8GE20-0XA0
CC 292, CPU 22x 实时时钟带有电池套管	6ES7 297-1AA20-0XA0
BC 293, CPU 22x 电池套管	6ES7 291-8BA20-0XA0
电缆、I/O扩展、8米、CPU 22x/EM	6ES7 290-6AA20-0XA0
电缆、RS-232/PPI多台主设备	6ES7 901-3CB30-0XA0
电缆、USB/PPI多台主设备	6ES7 901-3DB30-0XA0
编程软件	订购号
STEP 7-Micro/WIN 32 (V3.2) 单个许可证 (CDOM)	6ES7 810-2BC02-0YX0
STEP 7-Micro/WIN 32 (V3.2) 升级许可证 (CDOM)	6ES7 810-2BC02-0YX3
S7-200工具箱: 用于TP070的TP-Designer, 版本1.0 (CDOM) 更新	6ES7 850-2BC00-0YX0
STEP 7-Micro/WIN附加软件: STEP 7-Micro/WIN 32指令库, V1.1 (CDOM)	6ES7 830-2BC00-0YX0

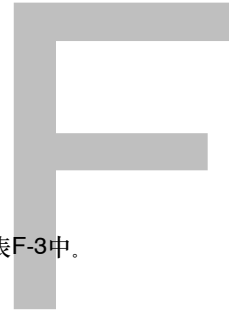
通讯卡	订购号
CP 5411: 短AT ISA	6GK 1 541-1AA00
CP 5511: PCMCIA、型号II	6GK 1 551-1AA00
CP5611: PCI卡 (版本3.0或更新的版本)	6GK 1 561-1AA00
手册	订购号
TD 200操作员界面用户手册	6ES7 272-0AA20-8BA0
TP 070触摸面板用户手册 (英语)	6AV6 591-1DC01-0AB0
S7-200点至点接口通讯手册 (英语/德语)	6ES7 298-8GA00-8XH0
CP 243-2 SIMATIC NET AS接口主设备手册 (英语)	6GK7 243-2AX00-8BA0
S7-200可编程控制器系统手册 (德语)	6ES7 298-8FA23-8AH0
S7-200可编程控制器系统手册 (英语)	6ES7 298-8FA23-8BH0
S7-200可编程控制器系统手册 (法语)	6ES7 298-8FA23-8CH0
S7-200可编程控制器系统手册 (西班牙语)	6ES7 298-8FA23-8DH0
S7-200可编程控制器系统手册 (意大利语)	6ES7 298-8FA23-8EH0
电缆、网络连接器和中继器	订购号
MPI电缆	6ES7 901-0BF00-0AA0
PROFIBUS网络电缆	6XVI 830-0AH10
网络总线连接器带编程口连接器、垂直电缆输出口	6ES7 972-0BB11-0XA0
网络总线连接器 (无编程口连接器)、垂直电缆输出口	6ES7 972-0BA11-0XA0
RS85总线连接器带有35°电缆输出口 (没有编程口连接器)	6ES7 972-0BA40-0XA0
RS85总线连接器带有35°电缆输出口 (带编程口连接器)	6ES7 972-0BB40-0XA0
CPU 22x/EM连接器块、7个端子、可移动	6ES7 292-1AD20-0AA0
CPU 22x/EM连接器块、12个可移动端子	6ES7 292-1AE20-0AA0
CPU 22x/EM连接器块、14个可移动端子	6ES7 292-1AF20-0AA0
CPU 22x/EM连接器块、18个可移动端子	6ES7 292-1AG20-0AA0
RS85 IP20中继器, 绝缘	6ES7 972-0AA00-0XA0

操作员界面	订购号
TD 200操作员界面	6ES7 272-0AA30-0YA0
OP3操作员界面	6AV3 503-1DB10T
OP7操作员界面	6AV3 607-1JC20-0AX1
OP17操作员界面	6AV3 617-1JC20-0AX1
TP070触摸面板	6AV6 545-0AA15-2AX0
TP170A触摸面板	6AV6 545-0BA15-2AX0
其它	订购号
DIN导轨停止	6ES5 728-8MA11
连接器（CPU 221、CPU 222）10个组件外12个位置风扇	6ES7 290-2AA00-0XA0
备用门套件，每个包含4套下列部件： 用于7、12、14、18、2x12、2x14端子的接线盒盖；CPU访问门，EM访问门	6ES7 291-3AX20-0XA0
8位置模拟程序	6ES7 274 1XF00-0XA0
14位置模拟程序	6ES7 274 1XH00-0XA0
24位置模拟程序	6ES7 274 1XK00-0XA0





# STL指令的执行时间



如果用户应用程序有时间临界的函数，指令执行次数十分重要。指令执行次数显示在表F-3中。



## 提示

当使用表格F-3中的执行次数时，应该考虑功率流对指令的影响、间接地址的影响以及在这些执行次数中使用某些内存区域。这些因子可以直接影响列出的执行次数。

## 功率流的影响

表F-3显示当功率流为该指令显示时（堆栈顶部 = 1或ON），执行指令的逻辑或函数所需的时间。

当功率流不显示时，执行该指令的时间为3  $\mu\text{s}$ 。

## 间接地址的影响

表F-3显示当使用操作数和常量的直接地址时，执行指令的逻辑或函数所需的时间。

当指令使用间接编址的操作数时，对于指令中使用的每个间接编址的操作数，指令的执行时间增加22微秒。

## 存取某些内存区的影响

存取某些内存区（例如AI、AQ、L和累加器）需要附加的执行时间。

表F-1显示在操作数中指定这些内存区后必须添加到指令执行时间的附加时间。

表 F-1 存取内存区的增加时间

内存区	增加的执行时间
模拟输入 (AI)	
没有启用模拟过滤:	149 $\mu\text{s}$
启用模拟过滤:	0 $\mu\text{s}$
模拟输出 (AQ)	73 $\mu\text{s}$
局部内存 (L)	5.4 $\mu\text{s}$
累加器 (AC)	4.4 $\mu\text{s}$

## 使用某些CPU 226XM指令的影响

在CPU 226XM上执行某些分支类型指令需要附加的执行时间。

表F-2为每个列出的指令提供添加到基本执行时间的因子。

表 F-2 CPU 226XM指令的增加时间

指示	增加的执行时间
ATCH	1.0 $\mu\text{s}$
CALL	4.3 $\mu\text{s}$
CSCRE	3.1 $\mu\text{s}$
FOR (对基时的增加时间)	3.1 $\mu\text{s}$
(对环路乘法器的增加时间)	3.1 $\mu\text{s}$
INT	1.7 $\mu\text{s}$
JMP	3.1 $\mu\text{s}$
RET	2.8 $\mu\text{s}$

表 F-3 指令执行次数

指令		$\mu\text{S}$	指令	$\mu\text{S}$
=	使用: I SM、T、C、V、S、Q、M L	0.37 1.8 19.2	ATT	70
+D		55	AW < =、=、>=、>、<、<>	45
-D		55	BCDI	66
*D		92	BIR 使用: 本地输入 扩展输入	45 53
/D		376	BIW 使用: 本地输出 扩展输出	46 56
+I		46	BMB 时间 = 基时 + (长度 * LM) 基时 (常量长度) 基时 (变量长度) 长度乘数 (LM)	21 51 11
-I		47	BBD 时间 = 基时 + (长度 * LM) 基时 (常量长度) 基时 (变量长度) 长度乘数 (LM)	21 51 20
*I		71	BMW 时间 = 基时 + (长度 * LM) 基时 (常量长度) 基时 (变量长度) 长度乘数 (LM)	21 51 16
/I		115	BTI	27
=I 使用: 本地输出 扩展输出		29 39	CALL 不使用任何参数: 使用参数: 时间 = 基时 + $\Sigma$ (操作数时间) 基时 操作数时间	15 32
+R		110 163最大	位 (输入、输出) 字节 (输入、输出) 字 (输入、输出) 双字 (输入、输出)	23、21 21、14 24、18 27、20
-R		113 166最大	注意: 从子例行程序返回期间, 进行输出操作数的 处理。	
*R		100 130最大	CFND 最大时间 = 基时 + $N1 * ((LM1 * N2) + LM2)$ 基时 长度乘数1 (LM1) 长度乘数2 (LM2) N1是源字符串的长度 N2是字符集字符串的长度	79 79 9.2 4.4
/R		300 360最大	COS	1525 1800最大
A 使用: I SM、T、C、V、S、Q、M L		0.37 1.1 10.8	CRET	13
AB < =、=、>=、>、<、<>		35	CRETI	23
AD < =、=、>=、>、<、<>		53	CSCRE	0.9
AENO		.6	CTD 在计数输入的转变时 否则	48 36
AI 使用: 本地输入 扩展输入		27 35	CTU 在计数输入的转变时 否则	53 35
ALD		0.37	CTUD 在计数输入的转变时 否则	64 45
AN 使用: I SM、T、C、V、S、Q、M L		0.37 1.1 10.8	DECB	30
ANDB		37	DECD	42
ANDD		55	DECO	36
ANDW		48		
ANI使用: 本地输入 扩展输入		27 35		
AR < =、=、>=、>、<、<>		54		
AS=、<> 时间 = 基时 + (LM * N) 基本 长度乘数 (LM) N是比较的字符数		51 9.2		
ATCH		20		
ATH 时间 = 基时 + (长度 * LM) 基时 (常量长度) 基时 (变量长度) 长度乘数 (LM)		41 55 20		

指令	$\mu\text{s}$	指令	$\mu\text{s}$
DECW	37	ITS	260
DISI	18	JMP	0.9
DIV	119	LBL	0.37
DTA	540	LD 使用: I、SM0.0 SM、T、C、V、S、Q、M L	0.37 1.1 10.9
DTI	36	LDB <=、=、>=、>、<、<>	35
DTCH	18	LDD <=、=、>=、>、<、<>	52
DTR	60 70最大	LDI 使用: 本地输入 扩展输入	26 34
DTS	540	LDN 使用: I SM、T、C、V、S、Q、M L	0.37 1.1 10.9
ED	15	LDNI 使用: 本地输入 扩展输入	26 34
ENCO	39 43最大	LDR <=、=、>=、>、<、<>	55
END	0.9	LDS	0.37
ENI	53	LDS=、<> 时间 = 基时 + (LM * N) 基本 长度乘数 (LM) N是比较的字符数	51 9.2
EU	15	LDW <=、=、>=、>、<、<>	42
EXP	1170 1375最大	LIFO	70
FIFO 时间 = 基时 + (长度 * LM) 基时 长度乘数 (LM)	70 14	LN	1130 1275最大
填充时间 = 基时 + (长度 * LM) 基时 (常量长度) 基时 (变量长度) 长度乘数 (LM)	29 50 7	LPP	0.37
FND <、=、>、<> 时间 = 基时 + (长度 LM) 基时 长度乘数 (LM)	85 12	LPS	0.37
FOR 时间 = 基时 + (循环数 * LM) 基时 循环乘数 (LM)	64 50	LRD	0.37
GPA	31	LSCR	12
HDEF	35	MEND	0.5
HSC	37	MOVB	29
HTA 时间 = 基时 + (长度 * LM) 基时 (常量长度) 基时 (变量长度) 长度乘数 (LM)	38 48 11	MOVD	38
IBCD	114	MOVW	38
INCB	29	MUL	34
INCD	42	NEXT	70
INCW	37	NETR	0
INT 典型带有1个中断	47	NETW 时间 = 基时 + (长度 * LM) 基时 长度乘数 (LM)	175 8
INVB	31	NOP	0.37
INVD	42	NOT	0.37
INVW	38	O 使用: I SM、T、C、V、S、Q、M L	0.37 1.1 10.8
ITA	260	OB <=、=、>=、>、<、<>	35
ITB	27	OD <=、=、>=、>、<、<>	53
ITD	36		

指令		$\mu\text{s}$	指令	$\mu\text{s}$
OI	使用: 本地输入 扩展输入	27 35	ROUND	108 183最大
OLD		0.37	RRB	总和 = 基时 + (长度LM) 基时 长度乘数 (LM)
ON	使用: I SM、T、C、V、S、Q、M L	0.37 1.1 10.8	RRD	总和 = 基时 + (长度LM) 基时 长度乘数 (LM)
ONI	使用: 本地输入 扩展输入	27 35	RRW	总和 = 基时 + (长度LM) 基时 长度乘数 (LM)
OR <=、=、>=、>、<、<>		55	RTA	时间 = 基时 + (LM * N) 基时 (用于结果的第一位) 长度乘数 (LM) N是结果中附加位的数目
ORB		37	RTS	时间 = 基时 + (LM * N) 基时 (用于结果的第一位) 长度乘数 (LM) N是结果中附加位的数目
ORD		55	S	用于长度 = 1 并指定为 常量 否则: 时间 = 基时 + (长度 * LM) 基时 长度乘数 (LM) 如果长度存储为变量, 添加到 时基
ORW		48	SBR	0
OS=、<>	时间 + 基时 + (LM * N) 基时 长度乘数 (LM) N是比较的字符数	51 9.2	SCAT	时间 = 基时 + (LM * N) 基时 长度乘数 (LM) N是附加的字符数
OW <=、=、>=、>、<、<>		45	SCPY	时间 = 基时 + (LM * N) 基时 长度乘数 (LM) N是复制的字符数
PID	基时 重算积分和微分比例常量的增加时间	750 1000	SCRE	0.37
PLS:	使用: PWM PTO单程序段 PTO多程序段	57 67 92	SCRT	17
R	长度=1并指定为常量 使用操作数 = C、T 使用所有其它操作数 其它, 时间=时基 + (长度 * LM) 操作数的时基 = C、T 所有其它操作数的时基 操作数的LM = C、T 所有其它操作数的LM 如果长度存储为变量, 添加到 时基	17, 24 5  19, 19 28 8.6, 16.5 0.9 29	SEG	30
RCV		80	SFND	最大时间 = 基时 + (N1-N2) * ((LM1*N2) + LM2) 基时 长度乘数1 (LM1) 长度乘数2 (LM2) N1是源字符串的长度 N2是搜索字符串的长度
RET		13	SHRB	总和 = 基时 + (长度 * LM1) + ((长度/8) * LM2) 基时 (常量长度) 基时 (变量长度) 长度乘数1 (LM1) 长度乘数2 (LM2)
RETI		23		76 84 1.6 4
RI	总和 = 时基 + (长度LM) 时基 LM 使用本地输出 LM 使用扩展输出 如果长度存储为变量, 添加到 时基	18 22 32 30		
RLB	总和 = 基时 + (长度+LM) 基时 长度乘数 (LM)	42 0.6		
RLD	总和 = 基时 + (长度LM) 基时 长度乘数 (LM)	52 2.5		
RLW	总和 = 基时 + (长度LM) 基时 长度乘数 (LM)	49 1.7		

指令	$\mu\text{s}$	指令	$\mu\text{s}$
SI 总和 = 基时 + (长度LM) 基时 LM使用本地输出 LM使用扩展输出 如果长度存储为变量, 添加到基时	18 22 32 30	STD 时间 = 基时 + (LM * N) 基时 (用于第1个源字符) 长度乘数 (LM) N是附加源字符的数目	84 59
SIN	1525 1800最大	STI 时间 = 基时 + (LM * N) 基时 (用于第1个源字符) 长度乘数 (LM) N是附加源字符的数目	84 59
SLB 总和 = 基时 + (长度LM) 基时 长度乘数 (LM)	43 0.7	STOP	16
SLD 总和 = 基时 + (长度LM) 基时 长度乘数 (LM)	53 2.6	STR 时间 = 基时 + (LM * N) 基时 (用于第1个源字符) 长度乘数 (LM) N是附加源字符的数目	100 120
SLEN	46	SWAP	32
SLW 总和 = 基时 + (长度LM) 基时 长度乘数 (LM)	51 1.3	TAN	1825 2100最大
SPA	243	TODR	2400
SQRT	725 830最大	TODW	1600
SRB 总和 = 基时 + (长度LM) 基时 长度乘数 (LM)	43 0.7	TOF	64
SRD 总和 = 基时 + (长度LM) 基时 长度乘数 (LM)	53 2.6	TON	64
SRW 总和 = 基时 + (长度LM) 基时 长度乘数 (LM)	51 1.3	TONR	56
SSCPY 时间 = 基时 + (LM * N) 基时 长度乘数 (LM) N是复制的字符数	82 8.8	TRUNC	103 178最大
		WDR	16
		XMT	78
		XORB	37
		XORD	55
		XORW	48



# S7-200快速参考信息



为帮助用户更容易地查找信息，此部分总结了下列信息：

- 特殊内存位
- 中断事件的说明
- S7-200 CPU内存范围和性能摘要
- 高速计数器HSC0、HSC1、HSC2、HSC3、HSC4、HSC5
- S7-200指令

表 G-1 特殊内存位

特殊内存位			
SM0.0	始终接通	SM1.0	操作结果 = 0
SM0.1	首次扫描	SM1.1	溢出或非法数值
SM0.2	保留性数据丢失	SM1.2	负结果
SM0.3	通电	SM1.3	除以0
SM0.4	30秒断开 / 30秒接通	SM1.4	表格满
SM0.5	0.5秒断开 / 0.5秒接通	SM1.5	表格空
SM0.6	断开1扫描 / 接通1扫描	SM1.6	BCD码到二进制转换出错
SM0.7	在“运行”位置的切换	SM1.7	ASCII到十六进制转换出错

表 G-2 以优先级顺序的中断事件

事件编号	中断说明	优先级组	组中的优先级
8	端口0: 接收字符	通讯 (最高)	0
9	端口0: 传输完成		0
23	端口0: 接收讯息完成		0
24	端口1: 接收讯息完成		1
25	端口1: 接收字符		1
26	端口1: 传输完成		1
19	PTO 0完成中断	离散 (中)	0
20	PTO 1完成中断		1
0	I0.0, 上升边缘		2
2	I0.1, 上升边缘		3
4	I0.2, 上升边缘		4
6	I0.3, 上升边缘		5
1	I0.0, 下降边缘		6
3	I0.1, 下降边缘		7
5	I0.2, 下降边缘		8
7	I0.3, 下降边缘		9
12	HSC0 CV=PV (当前值 = 预设值)		10
27	HSC0方向改变		11
28	HSC0外部重设		12
13	HSC1 CV=PV (当前值 = 预设值)		13
14	HSC1方向输入改变		14
15	HSC1外部重设		15
16	HSC2 CV=PV		16
17	HSC2方向改变		17
18	HSC2外部重设		18
32	HSC3 CV=PV (当前值 = 预设值)		19
29	HSC4 CV=PV (当前值 = 预设值)		20
30	HSC4 方向改变		21
31	HSC4 外部重设		22
33	HSC5 CV=PV (当前值 = 预设值)	23	
10	定时中断0	定时 (最低)	0
11	定时中断1		1
21	计时器T32 CT=PT 中断		2
22	计时器T96 CT=PT 中断		3



表 G-3 S7-200 CPU内存范围和性能摘要

说明	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
用户程序大小	4096个字节	4096个字节	8192个字节	8192个字节	16384个字节
用户数据大小	2048个字节	2048个字节	5120个字节	5120个字节	10240个字节
进程图像输入寄存器	I0.0到I15.7	I0.0到I15.7	I0.0到I15.7	I0.0到I15.7	I0.0到I15.7
进程图像输出寄存器	Q0.0到Q15.7	Q0.0到Q15.7	Q0.0到Q15.7	Q0.0到Q15.7	Q0.0到Q15.7
模拟输入（只读）	--	AIW0到AIW30	AIW0到AIW62	AIW0到AIW62	AIW0到AIW62
模拟输出（只写）	--	AQW0到AQW30	AQW0到AQW62	AQW0到AQW62	AQW0到AQW62
变量内存（V）	VB0到VB2047	VB0到VB2047	VB0到VB5119	VB0到VB5119	VB0到VB10239
局部内存（L） <sup>1</sup>	LB0到LB63	LB0到LB63	LB0到LB63	LB0到LB63	LB0到LB63
位存储器（M）	M0.0到M31.7	M0.0到M31.7	M0.0到M31.7	M0.0到M31.7	M0.0到M31.7
特殊内存（SM） 只读	SM0.0到SM179.7 SM0.0到SM29.7	SM0.0到SM299.7 SM0.0到SM29.7	SM0.0到SM549.7 SM0.0到SM29.7	SM0.0到SM549.7 SM0.0到SM29.7	SM0.0到SM549.7 SM0.0到SM29.7
计时器	256（T0到T255）	256（T0到T255）	256（T0到T255）	256（T0到T255）	256（T0到T255）
保持开延迟	1毫秒 10毫秒	T0、T64	T0、T64	T0、T64	T0、T64
100毫秒	T1到T4、和 T65到T68	T1到T4、和 T65到T68	T1到T4、和 T65到T68	T1到T4、和 T65到T68	T1到T4、和 T65到T68
开 / 关延迟	1毫秒 10毫秒 100毫秒	T5到T31、和 T69到T95	T5到T31、和 T69到T95	T5到T31、和 T69到T95	T5到T31、和 T69到T95
	T32、T96	T32、T96	T32、T96	T32、T96	T32、T96
	T33到T36、和 T97到T100	T33到T36、和 T97到T100	T33到T36、和 T97到T100	T33到T36、和 T97到T100	T33到T36、和 T97到T100
	T37到T63、和 T101到T255	T37到T63、和 T101到T255	T37到T63、和 T101到T255	T37到T63、和 T101到T255	T37到T63、和 T101到T255
计数器	C0到C255	C0到C255	C0到C255	C0到C255	C0到C255
高速计数器	HC0、HC3、HC4 和HC5	HC0、HC3、HC4 和HC5	HC0到HC5	HC0到HC5	HC0到HC5
顺序控制继电器（S）	S0.0到S31.7	S0.0到S31.7	S0.0到S31.7	S0.0到S31.7	S0.0到S31.7
累加器寄存器	AC0到AC3	AC0到AC3	AC0到AC3	AC0到AC3	AC0到AC3
跳转 / 标签	0至255	0至255	0至255	0至255	0至255
调用 / 子例行程序	0至63	0至63	0至63	0至63	0至127
中断例行程序	0至127	0至127	0至127	0至127	0至127
正向 / 负向转换	256	256	256	256	256
PID循环	0至7	0至7	0至7	0至7	0至7
端口	端口0	端口0	端口0	端口0、端口1	端口0、端口1

<sup>1</sup> LB60到LB63由STEP 7-Micro/WIN版本3.0或更新版本保留。

表 G-4 高速计数器HSC0、HSC3、HSC4和HSC5

模式	HSC0			HSC3	HSC4			HSC5
	I0.0	I0.1	I0.2	I0.1	I0.3	I0.4	I0.5	I0.4
0	Clk			Clk	Clk			Clk
1	Clk		重设		Clk		重设	
2								
3	Clk	方向			Clk	方向		
4	Clk	方向	重设		Clk	方向	重设	
5								
6	Clk向上	Clk向下			Clk向上	Clk向下		
7	Clk向上	Clk向下	重设		Clk向上	Clk向下	重设	
8								
9	相位A	相位B			相位A	相位B		
10	相位A	相位B	重设		相位A	相位B	重设	
11								

表 G-5 高速计数器HSC1和HSC2

模式	HSC1				HSC2			
	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5
0	Clk				Clk			
1	Clk		重设		Clk		重设	
2	Clk		重设	启动	Clk		重设	启动
3	Clk	方向			Clk	方向		
4	Clk	方向	重设		Clk	方向	重设	
5	Clk	方向	重设	启动	Clk	方向	重设	启动
6	Clk向上	Clk向下			Clk向上	Clk向下		
7	Clk向上	Clk向下	重设		Clk向上	Clk向下	重设	
8	Clk向上	Clk向下	重设	启动	Clk向上	Clk向下	重设	启动
9	相位A	相位B			相位A	相位B		
10	相位A	相位B	重设		相位A	相位B	重设	
11	相位A	相位B	重设	启动	相位A	相位B	重设	启动

布尔型指令		数学、递增和递减指令	
LD位	装载	+I IN1, OUT	加整数、双整数或实数 IN1+OUT=OUT
LDI位	立即载入	+D IN1, OUT	
LDN位	载入NOT (非)	+R IN1, OUT	
LDNI位	立即载入NOT (非)	-I IN1, OUT	减整数、双整数或实数 OUT-IN1=OUT
A位	与	-D IN1, OUT	
AI位	立即AND (与)	-R IN1, OUT	
AN位	与非	MUL IN1, OUT	乘以整数 (16*16->32)
ANI位	立即AND NOT (与非)	*I IN1, OUT	乘以整数、双整数或实数 IN1 * OUT = OUT
O位	OR (或)	*D IN1, OUT	
OI位	立即OR (或)	*R IN1, IN2	
ON位	OR NOT (或非)	DIV IN1, OUT	除以整数 (16/16->32)
ONI位	立即OR NOT (或非)	/I IN1, OUT	除以整数、双整数或实数 OUT / IN1 = OUT
LDBx IN1, IN2	载入“比较字节”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	/D IN1, OUT	
ABx IN1, IN2	AND (与) “字节比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	/R IN1, OUT	
OBx IN1, IN2	OR (或) “字节比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	SQRT IN, OUT	平方根
LDWx IN1, IN2	载入“字比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	LN IN, OUT	自然对数
AWx IN1, IN2	AND (与) “字比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	EXP IN, OUT	自然指数
OWx IN1, IN2	OR (或) “字比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	SIN IN, OUT	正弦
LDDx IN1, IN2	载入“双字比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	COS IN, OUT	余弦
ADx IN1, IN2	AND (与) “双字比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	TAN IN, OUT	正切
ODx IN1, IN2	OR (或) “双字比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	INCB OUT	
LDRx IN1, IN2	载入“实数比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	INCW OUT	递增字节、字或双字
ARx IN1, IN2	AND (与) “实数比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	INCD OUT	
ORx IN1, IN2	OR (或) “实数比较”的结果 IN1 (x: <, <=, =, >=, >, <>) IN2	DECB OUT	
NOT	堆栈非	DECW OUT	递减字节、字或双字
EU	检测上升边缘	DECD OUT	
ED	检测下降边缘	PID TBL, LOOP	PID循环
= 位	赋值	<b>计时器和计数器指令</b>	
=I 位	立即赋值	TON Txxx, PT	打开延迟计时器
S位、N	置位范围	TOF Txxx, PT	关闭延迟计时器
R位、N	重置位范围	TONR Txxx, PT	保留性打开延迟计时器
SI位、N	设置立即位范围	CTU Cxxx, PV	向上计数
RI位、N	重置立即位范围	CTD Cxxx, PV	向下计数
LDSx IN1, IN2	载入“字符串比较”的结果 IN1 (x: =, <>) IN2	CTUD Cxxx, PV	向上/向下计数
ASx IN1, IN2	AND (与) 运算“字符串比较”的结果 IN1 (x: =, <>) IN2	<b>实时时钟指令</b>	
OSx IN1, IN2	OR (或) “字符串比较”的结果 IN1 (x: =, <>) IN2	TODR T	读当日时钟时间
ALD	AND (与) 载入	TODW T	写当日时钟时间
OLD	OR (或) 载入	<b>程序控制指令</b>	
LPS	逻辑进栈 (堆栈控制)	END	程序有条件结束
LRD	逻辑读取 (堆栈控制)	STOP	切换到STOP (停止) 模式
LPP	逻辑出栈 (堆栈控制)	WDR	监视程序重置 (300毫秒)
LDS N	载入堆栈 (堆栈控制)	JMP N	跳转到定义标签
AENO	AND ENO	LBL N	定义要跳转的标签
		CALL N [N1,...]	调用子例行程序[N1, ... 至多16个供选用参数]
		CRET	从SBR有条件返回
		FOR INDX, INIT, FINAL	For/Next循环
		NEXT	
		LSCR N	
		SCRT N	
		CSCRE (有条件顺序控制继电器结束)	载入、转变、有条件结束和结束顺序控制继电器
		SCRE	

移动、移位和旋转指令		表格、查找和转换指令	
MOVB IN, OUT	移动字节、字、双字、实数	ATT DATA, TBL	添加数据到表格
MOVW IN, OUT		LIFO TBL, DATA	从表格取得数据
MOVD IN, OUT		FIFO TBL, DATA	
MOVR IN, OUT		FND= TBL, PTN, INDX	在表中查找匹配比较的数据值
BIR IN, OUT	FND< > TBL, PTN, INDX		
BIW IN, OUT	FND< TBL, PTN, INDX		
	FND> TBL, PTN, INDX		
BMB IN, OUT, N	块移动字节、字、双字	FILL IN, OUT, N	用模式填充内存空间
BMW IN, OUT, N		BCDI OUT	转换BCD至整数
BMD IN, OUT, N		IBCD OUT	转换整数至BCD
SWAP IN	交换字节	BTI IN, OUT	转换字节至整数
SHRB DATA, SBIT, N	移位寄存器位	ITB IN, OUT	转换整数至字节
SRB OUT, N	向右移位字节、字、双字	ITD IN, OUT	转换整数至双整数
SRW OUT, N		DTI IN, OUT	转换双整数至整数
SRD OUT, N		DTR IN, OUT	转换双字至实数
SLB OUT, N	向左移位字节、字、双字	TRUNC IN, OUT	转换实数至双整数
SLW OUT, N		ROUND IN, OUT	转换实数至双整数
SLD OUT, N		ATH IN, OUT, LEN	转换ASCII码至十六进制
RRB OUT, N	向右旋转字节、字、双字	HTA IN, OUT, LEN	转换十六进制至ASCII码
RRW OUT, N		ITA IN, OUT, FMT	转换整形至ASCII
RRD OUT, N		DTA IN, OUT, FM	转换双整数至ASCII
RLB OUT, N	向左旋转字节、字、双字	RTA IN, OUT, FM	转换实数至ASCII
RLW OUT, N		DECO IN, OUT	解码
RLD OUT, N		ENCO IN, OUT	编码
<b>逻辑指令</b>		SEG IN, OUT	生成7段模式
ANDB IN1, OUT	字节、字和双字的逻辑AND（与）	ITS IN, FMT, OUT	转换整数至字符串
ANDW IN1, OUT		DTS IN, FMT, OUT	转换双整数至字符串
ANDD IN1, OUT		RTS IN, FMT, OUT	转换实数至字符串
ORB IN1, OUT	字节、字和双字的逻辑OR（或）	STI STR, INDX, OUT	转换子字符串至整型
ORW IN1, OUT		STD STR, INDX, OUT	转换子字符串至双整数
ORD IN1, OUT		STR STR, INDX, OUT	转换子字符串至实数
XORB IN1, OUT	字节、字和双字的逻辑XOR（异或）	<b>中断指令</b>	
XORW IN1, OUT		CRETI	从中断指令有条件返回
XORD IN1, OUT		ENI	启用中断
INVB OUT	反转字节、字和双字 (1的反码)	DISI	禁用中断
INW OUT		ATCH INT, EVNT	附加中断例行程序至事件
INVD OUT		DTCH EVNT	分离事件
<b>字符串指令</b>		<b>通讯指令</b>	
SLEN IN, OUT	字符串长度	XMT TBL, PORT	自由端口传送
SCAT IN, OUT	并置字符串	RCV TBL, PORT	自由端口接收讯息
SCPY IN, OUT	复制字符串	NETR TBL, PORT	网络读取
SSCPY IN, INDX, N, OUT	从字符串复制子字符串	NETW TBL, PORT	网络写入
CFND IN1, IN2, OUT	在字符串中查找第一个字符	GPA ADDR, PORT	获取端口地址
SFND IN1, IN2, OUT	在字符串中查找字符串	SPA ADDR, PORT	设置端口地址
		<b>高速指令</b>	
		HDEF HSC, MODE	定义高速计数器模式
		HSC N	激活高速计数器
		PLS Q	脉冲输出

# 索引

## 符号

- &, 32
- \*, 32
- “传输”指令, 80

## 字母

### A

- AC安装指南, 19
- AC输出和继电器, 20
- ACCEL\_TIME (加速时间), EM 253位控模块, 259
- AENO指令, 71
- AI, 参见模拟输入 (AI)
- AND (或) 指令, 实例, 165
- AND (与) 载入指令, 71
- AND (与) 指令, 164
- AQ, 参见模拟输出 (AQ)
- AS-i向导, 410
- AS接口模块, 手册的订购号, 442
- ASCII到十六进制指令, 97
  - 实例, 100
- ASCII转换指令, 97

### B

- BCD至整数指令, 94

### C

- C内存, 26
- CP 243-1 IT互联网模块
  - 订购号, 408, 441
  - 功能, 408
  - 规范, 408
  - 互联网向导, 409
  - 接头, 409
  - 连接数目, 215
  - 配置, 409
  - 通讯处理器, 408
- CP 243-1以太网模块
  - 订购号, 405, 441
  - 功能, 406
  - 规范, 405
  - 连接, 406
  - 连接数目, 215
  - 配置, 406
  - 通讯处理器, 405
  - 向导, 406
- CP 243-2 AS-I接口, 规范, 410
- CP 243-2 AS接口模块
  - 订购号, 410, 441
  - 性能, 410

- CP 243-2通讯处理器
  - 操作, 411
  - 功能, 411
  - 接头, 411
  - 向导, 410
- CP卡
  - 订购号, 442
  - 选择, 225
  - 由STEP 7Micro/WIN支持, 226
- CP5411, 442
- CP5511, 442
- CP5611, 442
- CPU
  - 安装的螺丝尺寸, 16
  - 备份, 2
  - 标识寄存器 (SMB6), 430
  - 尺寸, 2
  - 错误处理, 56
  - 电源要求, 15, 419
  - 扩充模块, 2
  - 连接数目, 214
  - 密码保护, 44
  - 内存, 2
  - 扫描循环, 22
  - 实时时钟, 2
  - 输入模拟程序, 417
  - 数字输入/输出, 2
  - 通讯端口, 2
  - 网络上的搜索, 213
  - 支持智能模块, 313
  - 执行速度, 2
- CPU 224, 范例DP程序, 395

### D

- DC安装指南, 19
- DC晶体管, 保护, 20
- DC输出和继电器, 20
- DECEL\_TIME (减速时间), EM 253位控模块, 259
- DIN横杆安装, 16
- DIP开关
  - RS-232/PPI多台主设备电缆, 415
  - RTD, 380-381
  - 热电偶, 376
- DIP开关设置, PPI多主设备电缆, 7
- DIP开关设置, PPI多台主设备电缆, 228

### E

- EEPROM, 34
  - 保存变量内存 (V), 38, 433
  - 保存位存储器 (M), 35
  - 错误代码, 424
- EM 231 RTD模块
  - CPU兼容性, 374
  - 规范, 373
  - 连接器端子, 374
  - 配置, 380
  - 温度范围和精度, 383-384
  - 选择DIP开关, 380-381
  - 状态指示器, 382

- EM 231模拟输入模块
  - 安装, 370
  - 规范, 372
  - 精度与可重复性, 371
  - 输入方框图, 368
  - 输入数据字格式, 368
  - 校准, 365
- EM 231热电偶模块
  - CPU兼容性, 374
  - 规范, 373
  - 基础, 375
  - 连接器端子, 374
  - 配置, 375
  - 温度范围和精度, 378–379
  - 选择DIP开关, 376
  - 状态指示器, 377
- EM 232模拟量输出模块, 输出方框图, 370
- EM 232模拟输出模块, 输出数据字格式, 369
- EM 235模拟I/O模块
  - 安装, 370
  - 规范, 372
  - 精度与可重复性, 371
  - 配置, 367
  - 输出方框图, 370
  - 输出数据字格式, 369
  - 输入方框图, 369
  - 输入数据字格式, 368
  - 校准, 365
- EM 241 调制解调器模块, 实例, 313
- EM 241调制解调器模块
  - CPU兼容性, 398
  - CPU数据传送讯息格式, 320
  - Modbus RTU协议, 299
  - Modbus地址, 300
  - MODx\_CTRL指令, 309
  - MODx\_MSG指令, 310
  - MODx\_XFR指令, 309
  - RJ11插头, 298
  - STEP 7-Micro/WIN接口, 299
  - 安全回叫, 302
  - 安装, 398
  - 调制解调器扩充向导, 304
  - 短讯息服务 (SMS), 300
  - 方框图, 397
  - 规范, 397
  - 国际电话线接口, 298
  - 密码保护, 301
  - 配置表, 303, 316
  - 数据传送, 301
  - 数字寻呼, 300
  - 特殊内存, 314
  - 特性, 298
  - 文字寻呼, 300
  - 寻呼, 300
  - 讯息传送电话号码, 318
  - 支持的国家, 298
  - 指令, 308
  - 指令中的错误, 311
  - 状态LED, 304
- EM 253控制面板, 284–286
- EM 253位控模块
  - ACCEL\_TIME, 259
  - CPU兼容性, 401
  - DECEL\_TIME, 259
  - EM 253控制面板, 284–286
  - POSx\_CFG, 278
  - POSx\_CLR, 277
  - POSx\_CTRL, 268
  - POSx\_DIS, 276
  - POSx\_GOTO, 270
  - POSx\_LDOFF, 273
  - POSx\_LDPOS, 274
  - POSx\_MAN, 269
  - POSx\_RSEEK, 272
  - POSx\_RUN, 271
  - POSx\_SRATE, 275
  - RP查找模式, 263–267
  - RP搜索顺序, 261
  - SS\_SPEED, 258
  - 编程, 255
  - 布线图, 403–404
  - 参考点 (RP), 260
  - 参考点搜索 (RS搜索), 260
  - 测量类型, 选择, 256
  - 出错代码, 286
  - 创建指令, 296
  - 负极, 257
  - 概要图, 262
  - 概要图步进, 263
  - 概要图操作模式, 262
  - 控制模块的范例程序, 296
  - 脉冲与定向输出, 257
  - 慢进参数, 259
  - 命令字节, 293
  - 模块出错代码, 287
  - 配置, 256, 285
  - 配置 / 概要表, 288
  - 实例, 279–283
  - 输入过滤器时间, 257
  - 输入和输出, 254
  - 输入现用级别, 257
  - 特殊内存, 291
  - 特性, 254
  - 跳动时间, 260
  - 位置控制向导, 256
  - 显示和控制运行, 284
  - 响应物理输入, 258
  - 消除反冲, 266
  - 移动概要图, 定义, 262
  - 移动命令, 294
  - 诊断信息, 285
  - 正极, 257
  - 指令, 267
  - 指令出错代码, 286
  - 指令使用指南, 267
  - 最大速度和启动 / 停止速度, 258

- EM 277 PROFIBUS-DP模块
  - CPU兼容性, 386
  - DP协议, 387
  - LED状态指示器, 392
  - PROFIBUS网络上, 387
  - 地址开关, 386
  - 附加特性, 392
  - 规范, 385
  - 连接数目, 214
  - 配置, 388-389
  - 配置文件, 393-394
  - 配置选项, 389
  - 数据交换模式, 390
  - 特殊内存字节, 391
  - 状态LED, 386
  - 作为DP从属装置, 387
- EN, 55, 63
- ENO, 55, 63
- Exclusive OR (异或) 指令, 164
  - 实例, 165
  
- F**
- FBD编辑器
  - 描述, 52
  - 特点, 52
  - 约定, 54
- For-next循环指令
  - for, 171
  - next, 171
  - 实例, 172
  
- G**
- GSD文件, EM 277 PROFIBUS-DP, 393-394
  
- H**
- HMI设备, 226
- HSC, 46
  - 指令向导, 113
- HSC0, HSC1, HSC2寄存器 (SMB36到SMB65), 433
- HSC3, HSC4, HSC5寄存器 (SMB131到SMB165), 437
  
- I**
- I/O, 读和写, 39
- I/O编址, 31
- I/O错误, 57
- I/O扩展电缆, 安装, 412
- I/O模块标识和错误寄存器, 431
- I/O状态 (SMB5), 430
- I内存, 25
- IEC计时器指令, 202
  - 实例, 202
- IEC计数器指令
  - 实例, 110
  - 向上 / 向下计数器, 110
  - 向上计数器, 110
  - 向下计数器, 110
- IEC1131-3指令集, 53
  
- J**
- JOG\_INCREMENT, EM 253位控模块, 259
- JOG\_SPEED, EM 253位控模块, 259
  
- L**
- LAD编辑器
  - 描述, 52
  - 特点, 52
  - 约定, 54
- LED, EM 241调制解调器模块, 304
- L内存, 28
  
- M**
- MAX\_SPEED, EM 253位控模块, 258
- MBUS\_INIT指令, 343
- MBUS\_SLAVE指令, 345
- Micro PLC系统, 设计, 48
- Micro/WIN, 参考STEP 7-Micro/WIN
- MicroMaster驱动器
  - 读和写, 329, 330
  - 控制, 321
  - 连接, 334
  - 通讯, 322
- MM3驱动器
  - 安装, 335
  - 连接, 334
- MM4驱动器
  - 安装, 338
  - 连接, 337
- Modbus RTU协议, 342
  - EM 241调制解调器模块, 299
  - 调制解调器模块所支持的功能, 299
  - 映射地址, 300
- Modbus从属协议
  - CRC表格, 340
  - MBUS\_INIT, 343
  - MBUS\_SLAVE, 345
  - 编程实例, 346
  - 初始化, 340
  - 地址, 341
  - 配置符号表, 341
  - 使用的资源, 340
  - 特殊内存, 340
  - 映射地址到S7-200, 341
  - 支持的功能, 342
  - 执行出错代码, 345
  - 执行时间, 340
  - 指令, 342
- Modbus协议库, 339
- MODx\_CTRL指令, EM 241调制解调器模块, 309
- MODx\_MSG指令, EM 241调制解调器模块, 310

MODx\_XFR指令, EM 241调制解调器模块, 309

MPI网络

大于187.5 Kbaud, 217

低于187.5 Kbaud, 217

MPI协议, 215, 237

M内存, 25

## N

NETR, NETW, 指令向导, 75

NEXT指令, 实例, 172

Next指令, 171

NOT (非) 指令, 66

PPI协议, 214, 237

单台主设备网络, 216

多台主设备网络, 216

复杂网络, 217

PROFIBUS, 主设备和从属装置, 211

PROFIBUS-DP

标准通讯, 386

范例程序, 395

模块 (EM 277), 387

数据一致性, 390

PROFIBUS-DP网络

S7-315-2和EM 277, 218

STEP 7-Micro/WIN和HMI, 218

插针分配, 223

电缆规格, 222

中继器, 222

PROFIBUS协议, 215, 237

PTO/PWM功能, 寄存器 (WMB66到SMB85), 435

PTO0, PTO1配置文件定义表 (SMB166到SMB185), 438

## O

OP3、OP7、OP17, 订购号, 443

OR (或) 指令, 实例, 165

OR (或) 载入指令, 71

OR (或) 指令, 164

## P

PID环路指令, 146

比例项, 148

标准化环路输入, 150

环路控制, 149

环路控制类型, 149

积分项, 148

微分项, 149

转换环路输入, 150

PID循环指令

报警检查, 152

变量, 151

程序实例, 153

错误条件, 152

反向作用, 151

范围, 151

了解, 147

模式, 152

实例, 154

手动模式, 152

向导, 146

循环表, 153

正向作用, 151

转换环路输出到成比例的整数, 151

自动模式, 152

PLC信息对话框, 56

POSx\_CFG, 278

POSx\_CLR, 277

POSx\_CTRL, 268

POSx\_DIS, 276

POSx\_GOTO, 270

POSx\_LDOFF, 273

POSx\_LDPOS, 274

POSx\_MAN, 269

POSx\_RSEEK, 272

POSx\_RUN, 271

POSx\_SRATE, 275

PPI/自由端口模式, 无线电调制解调器, 234

PPI多台主设备电缆

RS-232标准, 228

波特率开关选择, 233

无线电调制解调器, 234

配置, 231

配置自由端口, 232

使用调制解调器, 233

使用无线电调制解调器, 234

选择, 210, 225

自由端口模式, 228

PPI多主站电缆, 4

PPI模式, 无线电调制解调器, 234

PPI通讯, 更换到自由端口模式, 81

## Q

Q内存, 25

## R

RJ11插口, EM 241调制解调器模块, 398

RJ11插头, EM 241调制解调器模块, 298

RP查找模式选项, 263267

EM 253位控模块, 263267

RS-232/PPI多台主设备电缆

DIP开关设置, 415

端口设置, 221

规范, 413

配置超级终端, 241485

RS-232标准

PPI多台主设备电缆, 228

自由端口模式, 228

RS-485标准, 223

RTD模块 (EM 231), 380

RUN (运行) 模式, 12, 37

编辑程序, 246



## S

- S7-200
  - C内存, 26
  - CPU模块, 2
  - I内存, 25
  - L内存, 28
  - M内存, 25
  - Q内存, 25
  - RS-232标准, 228
  - RUN (运行) 模式, 12, 37
  - S内存, 29
  - SM内存, 28
  - STOP (停止) 模式, 12, 37
  - T内存, 26
  - V内存, 25
  - 安装, 15
  - 安装指南, 14
  - 保存数据, 34
  - 保留内存, 41
  - 编程约定, 55
  - 编址, 24
  - 变量内存区域 (V), 25
  - 波特率, 211, 212–214
  - 布线图, 354–356
  - 布线指南, 19
  - 常数, 30
  - 尺寸, 2
  - 存取数据, 24
  - 错误处理, 56
  - 错误代码, 424
  - 电磁兼容性, 350
  - 电源, 6
  - 调制解调器, 233
  - 读取和写入数据, 22
  - 范例网络配置, 216, 217, 218, 219
  - 高速计数器, 27
  - 故障诊断, 251
  - 环境条件, 350
  - 技术规范, 350
  - 计时器内存区域 (T), 26
  - 计数器内存区域 (C), 26
  - 接地指南, 19
  - 进程图像寄存器, 39
  - 进程图像输出寄存器 (Q), 25
  - 进程图像输入寄存器 (I), 25
  - 局部内存区域 (L), 28
  - 扩充模块, 3
  - 累加器, 27
  - 连接STEP 7-Micro/WIN, 8
  - 脉冲捕获功能, 42
  - 密码保护, 44
  - 模拟输出 (AQ), 29
  - 模拟输入 (AI), 29
  - 内存, 24
  - 内存磁带, 36
  - 内存范围, 64, 482
  - 扫描循环, 22, 39
  - 上载, 34
  - 顺序控制继电器内存区域 (S), 29
  - 特殊内存区域 (SM), 28
  - 特性, 39
  - 通讯进程, 40
  - 网络地址, 212–214
  - 位存储器区域 (M), 25
  - 无线电调制解调器, 234
  - 系统组件, 2
    - 下载, 34
  - 响应严重错误, 424
  - 硬件故障诊断, 251
  - 支持的中断例行程序, 159
  - 执行控制逻辑, 22
  - 中断例行程序, 158
  - 状态图, 249
    - 作为从属装置, 211, 387
- S7-200系统手册, 订购号, 442
- S7-300, 范例网络配置, 217
- S7-400, 范例网络配置, 217
- SIMATIC计时器指令, 197
  - 实例, 199, 200, 201
- SIMATIC计数器指令
  - 实例, 109
  - 向上 / 向下计数, 107
  - 向上计数, 107
  - 向下计数, 107
- SIMATIC指令集, 53
- SM内存, 28
  - Modbus从属协议, 340
  - PTO/PWM操作, 129
- SM0.2保留数据丢失内存位, 35
- SMB0: 状态位, 428
- SMB1: 状态位, 428
- SMB130: 自由端口控制寄存器, 432
- SMB131到SMB165: HSC3, HSC4, HSC5寄存器, 437
- SMB166到SMB185: PTO0, PTO1配置文件定义表, 438
- SMB186到SMB194: 接收讯息控制, 436
- SMB2: 自由端口接收字符, 429
- SMB200到SMB549: 智能模块状态, 439
- SMB28, SMB29模拟调整, 45, 432
- SMB3: 自由端口奇偶校验错误, 429
- SMB30和SMB130: 自由端口控制寄存器, 432
- SMB31和SMW32: EEPROM写控制, 433
- SMB34和SMB35: 定时中断寄存器, 433
- SMB36到SMB65: HSC0, HSC1, HSC2寄存器, 433
- SMB4: 队列溢出, 429
- SMB5: I/O状态, 430
- SMB6: CPU标识寄存器, 430
- SMB66到SMB85: PTO/PWM寄存器, 435
- SMB7: 保留, 430
- SMB8到SMB21: I/O模块标识和错误寄存器, 431
- SMB86到SMB94, SMB186到SMB194: 接收讯息控制, 436
- SMS, 调制解调器模块, 300
- SMW22到SMW26: 扫描时间, 432
- SMW98: 扩展I/O总线出错, 437
- SS\_SPEED, EM 253位控模块, 258
- STEP 7-Micro/WIN
  - CP卡, 226
  - EM 241的接口, 299
  - PPI多台主设备电缆, 226
  - 安装, 4
  - 波特率, 212–214
  - 程序包, 3
  - 调试工具, 245

订购号, 442  
 范例网络配置, 216–219  
 计算机要求, 3  
 连接S7-200, 8  
 启动, 7  
 设备要求, 3  
 通讯设置, 8  
 网络地址, 212–214  
 作为主设备, 211  
 STEP 7-Micro/WIN 32指令库, 442  
 STEP7-Micro/WIN  
   程序编辑器, 51  
   创建程序, 51  
   打开, 51  
   指令集  
     IEC1131-3, 53  
     SIMATIC, 53  
     选择, 53  
 STL编辑器  
   描述, 51  
   特点, 51  
 STOP (停止) 模式, 12, 37  
 S内存, 29  
  
**T**  
 TC/IP, 通讯协议, 215  
 TCP/IP, 协议, 215  
 TD 200, 向导, 4  
 TD 200文本显示单元, 4  
   订购号, 443  
   手册的订购号, 442  
 TOD时钟, 74  
 TP070触摸面板单元, 4  
   订购号, 443  
   手册的订购号, 442  
 TP070的TP-Designer, 版本1.0, 442  
 T内存, 26  
  
**U**  
 USB/PPI多台主设备电缆  
   订购号, 413  
   端口设置, 221  
   选择, 210  
 USS协议, 要求, 322  
 USS协议库, 控制MicroMaster驱动器, 321  
 USS协议指令  
   USS4\_DRV\_CTRL, 326  
   USS4\_INIT, 325  
   USS4\_RPM\_x和USS4\_WPM\_x, 329, 330  
   范例程序, 332  
   使用指南, 324  
   执行错误代码, 333  
 USS4\_DRV\_CTRL指令, 326  
 USS4\_INIT指令, 325  
 USS4\_RPM\_x指令, 329, 330  
 USS4\_WPM\_x指令, 329, 330

## V

V内存, 25  
   保存到EEPROM, 38  
   从EEPROM恢复, 35  
   分配地址, 58

## A

安全电路, 设计, 48  
 安全回叫, EM 241调制解调器模块, 302  
 安全性, 密码, 44  
 安装  
   CPU模块, 16  
   DIN横杆, 16  
   EM 231, 370  
   EM 235, 370  
   I/O扩展电缆, 412  
   MM3驱动器, 335  
   MM4驱动器, 338  
   S7-200, 15  
   STEP 7-Micro/WIN, 4  
   安装要求, 16  
   尺寸, 16  
   电源, 15  
   电噪声, 14  
   发热设备, 14  
   高压设备, 14  
   间隙要求, 14, 16  
   扩充模块, 16  
   面板, 16  
   内存磁带, 36  
   指南, 14

## B

保存  
   变量内存 (V) 到EEPROM, 38  
   程序, 11  
   数值到EEPROM, 433  
   位存储器 (M) 到EEPROM, 35  
   S7-200程序数据, 34  
 保护电路, 349  
 保留内存, 41  
 保留位存储器, 35  
 保留性打开延迟计时器指令 (TONR), 197  
   实例, 201  
 报警检查, PID循环, 152  
 比较, 令牌轮转时间, 237  
 比较实数指令, 90  
 比较双字指令, 90  
 比较整数指令, 90  
 比较指令, 55  
   比较实数, 90  
   比较双字, 90  
   比较整数, 90  
   比较字符串, 92  
   比较字节, 90  
   实例, 90  
 比较字符串指令, 92

比较字节指令, 90  
 比例项, PID算法, 148  
 避免, 网络冲突, 238  
 编程  
   EM 253位控模块, 255  
   高速计数器, 113  
 编程电缆, 4  
 编程软件, 订购号, 442  
 编辑, 在RUN (运行) 模式中, 246  
 编辑器  
   功能块图 (FBD), 52  
   梯形图 (LAD), 52  
   语句表 (STL), 51  
 编码指令, 106  
   实例, 106  
 编译错误, 56  
 编译规则违反, 426  
 编址  
   S7-200内存, 24  
   变量内存, 25  
   高速计数器, 27  
   计时器内存, 26  
   计数器内存, 26  
   间接 (指针), 32  
   进程图像输出寄存器, 25  
   进程图像输入寄存器, 25  
   局部内存, 28  
   局部I/O, 31  
   扩展I/O, 31  
   累加器, 27  
   模拟输出, 29  
   模拟输入, 29  
   内存区域, 25-28  
   顺序控制继电器 (SCR) 内存, 29  
   特殊内存 (SM) 位, 28  
   位存储器, 25  
   直接, 24  
   字节:位, 24  
 变量  
   PID循环, 151  
   符号编址, 58  
   监控, 59  
   用状态图监控, 249  
 变量内存区域 (V), 25  
 标签指令, 173  
 标准, 国家和国际, 348  
 标准化, 环路输入, 150  
 标准接点指令, 66  
 标准转换指令, 93  
 标准DIN横杆, 15  
 表查找指令, 194  
   实例, 196  
 表格指令  
   后入先出, 191  
   内存填充, 193  
   添加到表格, 190  
   先入先出, 191  
 表指令, 表查找, 194  
 并置字符串指令, 185

波特率  
 开关选择  
   PPI多台主设备电缆, 213, 226, 228, 233, 234  
   多主设备电缆, 7  
   设置, 211  
   网络, 222  
   最优, 235  
 布尔型指令  
   接点, 66  
   逻辑堆栈, 71  
   设置/重设双稳态, 73  
   线圈, 69  
 布线, 18, 19  
 布线图  
   CPU模块, 354-356  
   CPU输入和输出, 354  
   EM 253位控模块, 403-404  
   分立扩充模块, 360-362  
   模拟扩充模块, 364  
 步进电机控制, PTO/PWM发生器, 131

## C

参考点 (RP), 260  
 RP\_OFFSET, 261  
 RP搜索  
   搜索方向, 260  
   RP\_APPR\_DIR, 260  
   RP\_FAST, 260  
   RP\_SEEK\_DIR, 260  
   RP\_SLOW, 260  
   RP搜索顺序, 模式, 261  
 参数  
   对于子例行程序的类型, 205  
   在子例行程序中, 205  
 操作模式, CPU, 状态位, 428  
 操作模式, CPU  
   更换, 37  
   自由端口协议, 227  
 操作数范围, 65  
 操作员界面, 订购号, 443  
 操作站, 指定, 48  
 测量类型, EM 253位控模块, 256  
 插针分配, 通讯端口, 223  
 查找指令, 194  
 拆除  
   CPU模块, 17  
   扩充模块, 17  
 常数, 30  
 超级终端, 配置RS-232/PPI多台主设备电缆, 241-485  
 成比例的整数, 转换环路输出, 151  
 成块移动双字指令, 168  
 成块移动指令, 实例, 168  
 成块移动字节指令, 168  
 成块移动字指令, 168  
 乘法指令, 141  
 乘以整数到双整数指令 (MUL), 143  
   实例, 143  
 乘指令, 141

## 程序

- PID实例, 153
- 保存, 11
- 编译错误, 56
- 创建, 9
- 从内存磁带恢复, 37
- 存储, 34-38
- 调试功能, 246
- 复制到内存磁带, 36
- 构建, 49
- 基本单元, 49
- 监控, 12
- 监控状态, 248
- 模拟输入, 22
- 上载, 34
- 使用STEP7-Micro/WIN进行创建, 51
- 下载, 12, 34
- 与中断例行程序共享数据, 158
- 运行, 12
- 在RUN (运行) 模式中编辑, 246
- 执行错误, 57
- 指定扫描数, 250
- 状态图, 59
- 子程序, 50
- 程序范例, 控制位控模块, 296
- 程序编辑器, 51
  - 打开, 9
  - 功能块图 (FBD), 51
  - 梯形图 (LAD), 51
  - 选择, 51
  - 语句表 (STL), 51
  - 约定, 54
- 程序控制指令
  - for-next循环, 171
  - 基本程序控制, 169
  - 监视程序重设, 169
  - 顺序控制继电器 (SCR), 174
  - 跳转指令, 173
  - 停止, 169
  - 有条件结束, 169
- 尺寸, CPU, 2
- 冲突, 避免网络, 238
- 初始化
  - Modbus协议, 340
  - PWM输出, 133
  - 高速计数器, 121
  - 用于单段操作的PTO, 135
  - 用于多段操作的PTO, 136
- 初始值, 分配, 58
- 出错
  - EM 241 调制解调器模块配置, 303
  - Modbus从属协议执行, 345
  - SMB1, 执行出错, 428
  - 网络读取和写入指令, 76
- 出错代码
  - EM 253位控模块, 286
  - EM 253位控模块的模块错误, 287
  - EM 253位控模块的指令, 286
- 除法指令, 141
- 除以整数带余数指令 (DIV), 143
  - 实例, 143

## 处理

- 错误, 56
  - 复杂通讯, 238
  - 通讯请求, 23
  - 传输率, 网络, 222
  - 传输指令
    - 传输数据, 81
    - 实例, 87
    - 自由端口模式, 80
  - 创建
    - Micro/WIN程序, 51
    - 程序, 9
    - 符号名列表, 49
    - 配置图, 49
    - 自定义协议, 227
  - 从输入读取数据, 22, 23
  - 从属装置, 211
    - EM 277 PROFIBUS-DP, 387
    - S7-200, 387
  - 从中断返回指令, 156
  - 从中断有条件返回指令, 156
  - 从子例行程序返回指令, 204
    - 实例, 207
  - 从子例行程序有条件返回指令, 204
  - 从字符串复制子字符串指令, 187
  - 存储, S7-200程序数据, 34
  - 存取
    - S7-200数据, 24, 32
    - 直接编址, 24
  - 存取限制, 44
  - 错误
    - EM 241 调制解调器模块指令, 311
    - I/O, 57
    - PID循环, 152
    - 程序编译, 56
    - 程序执行, 57
    - 处理, 56
    - 非严重, 56
    - 严重, 57
    - 运行系统, 57
    - 浏览, 56
  - 错误代码, 424
    - USS协议指令, 333
    - 编译规则违反, 426
    - 严重错误, 424
    - 运行系统程序问题, 425
- ## D
- 打开, 程序编辑器, 9
  - 打开延迟计时器指令 (TON), 197, 202
    - 实例, 199
  - 代理认证, 348
  - 单段操作
    - 初始化PTO, 135
    - 改变PTO脉冲计数, 135
    - 改变PTO周期时间, 135
    - 改变PTO周期时间和脉冲计数, 136
  - 单台主设备PPI网络, 216

- 当前值
    - 为HSC进行设置, 119
    - 在HSC中更换, 124
  - 德意志劳埃德 (GL) 海事代理, 348
  - 地址
    - Modbus, 341
      - 分配, 58
      - 符号, 58
      - 高速计数器, 120
      - 设置远程, 213
      - 网络, 211
  - 地址开关, EM 277 PROFIBUS-DP, 386
  - 递减指令, 145
    - 实例, 145
  - 递增指令, 145
    - 实例, 145
  - 电池盒, 34
  - 电池夹头, 412
  - 电池套管, 订购号, 442
  - 电磁
    - 辐射标准, 350
    - 兼容性, S7-200, 350
    - 抗干扰标准, 350
  - 电话线接口, 国际, EM 241调制解调器模块, 298
  - 电机, 扭矩速度曲线, 典型的, 258
  - 电缆
    - I/O扩展, 412
    - RS-232/PPI多台主设备, 413
    - USB/PPI多台主设备, 413
    - 订购号, 442
    - 端接, 224
    - 配置
      - 远程调制解调器, 231
      - 自由端口, 232
    - 偏置, 224
    - 网络, 222, 223
    - 选择, 210
  - 电气使用寿命, 349
  - 电位器, 模拟调整, 45
  - 电涌抑制, 20
  - 电源, 6, 15
  - 电源规范, CPU模块, 352
  - 电源要求, 15
    - CPU, 419
      - 计算, 419, 421
      - 计算表格, 421
      - 扩充模块, 419
      - 实例, 420
  - 调试
    - 多次扫描, 250
    - 功能, 246
    - 强制值, 250
    - 在RUN (运行) 模式中编辑, 246
  - 调用子例行程序指令, 204
    - 实例, 205
  - 调制解调器
    - 使用PPI多台主设备电缆, 233
    - 远程, PPI多台主设备电缆, 231
  - 调制解调器扩充向导, 231
    - EM 241调制解调器模块, 304
  - 调制解调器模块, 397
    - CPU数据传送讯息格式, 320
    - MODx\_CTRL指令, 309
    - MODx\_MSG指令, 310
    - MODx\_XFR指令, 309
    - RJ11插头, 298
    - SMS讯息传送, 300
    - 安全回叫, 302
    - 调制解调器扩充向导, 304
    - 短讯服务, 300
    - 国际电话线接口, 298
    - 密码保护, 301
    - 配置表, 303
    - 实例, 313
    - 数据传送, 301
    - 数字寻呼, 300
    - 特性, 298
    - 文字寻呼, 300
    - 文字讯息格式, 319
    - 寻呼, 300
    - 讯息传送电话号码, 318
    - 指令, 308
    - 指令中的错误, 311
    - 状态LED, 304
  - 定时中断, 时间间隔寄存器 (SMB34, SMB35), 433
  - 定时中断例行程序, 实例, 162
  - 定时中断排队, 160
  - 定位控制向导, 126
  - 订购号, 441-884
    - CP 243-2 AS接口模块, 410
    - CPU模块, 351
    - PPI多台主设备电缆, 413
    - 模拟扩充模块, 363, 373
    - 输入模拟程序, 417
    - 数字扩充模块, 357
  - 丢失密码, 45
  - 读取实时时钟指令, 74
  - 端接, 网络电缆, 224
  - 端口, 设置, PPI多台主设备电缆, 221
  - 短讯服务, EM 241调制解调器模块, 300
  - 段指令, 96
    - 实例, 96
  - 断电, 保留内存, 35, 41
  - 断开检测, 84
  - 队列, 中断例行程序, 160
  - 队列溢出 (SMB4), 429
  - 对等通讯, 217-218
  - 多段操作, 初始化PTO, 136
  - 多台主设备PPI网络, 216
  - 多主设备电缆, 7
- ## E
- 二极管抑制, 20

## F

法国船级社 (BV) 海事代理, 348  
反冲  
  补偿, 261  
  选择工作区, 266  
反向作用循环, 151  
反转双字指令, 163  
反转指令, 实例, 163  
反转字节指令, 163  
反转字指令, 163  
范例程序, 9  
范围, PID循环, 151  
方框图, EM 241 调制解调器, 397  
方向, 在HSC中改变, 124  
访问, 状态图, 59  
非严重错误, 56  
分辨率, 计时器, 198, 199  
分离中断指令, 156  
分配  
  HSC的中断, 120  
  初始值, 58  
  地址, 58  
  网络地址, 211  
分散控制  
  实例, 177  
  顺序控制继电器指令, 177  
符号编址, 58  
符号表, 58  
  编址, 58  
  配置Modbus, 341  
符号名, 创建列表, 49  
浮点数, 29  
浮点数数值, 150  
复位指令, 69  
复杂PPI网络, 217  
复制, 程序到内存磁带, 36  
复制字符串指令, 185  
负极, EM 253位控模块, 257  
负向转换指令, 66  
附加中断指令, 156

## G

改变  
  PTO脉冲计数, 135  
  PTO周期时间, 135  
  PTO周期时间和脉冲计数, 136  
  高速计数器方向, 124  
  脉冲宽度, 133  
概要表, EM 253位控模块, 288  
概要表数值, PTO/PWM生成器, 131  
概要图步进, EM 253位控模块, 263  
概要图操作模式, EM 253位控模块, 262  
高电位绝缘测试, 350  
高速度脉冲输出, SMB66-SMB85, 435

高速计数器, 46  
  SMB36-SMB65, 433  
  编程, 113  
  编址, 120  
  初始化顺序, 121  
  定义横式和输入, 114  
  分配中断, 120  
  改变方向, 124  
  更换当前值, 124  
  更换预设值, 124  
  计时图, 115-117  
  禁用, 124  
  控制字节, 113  
  了解, 113  
  模式, 115,454  
  内存区域,编址, 27  
  设置当前值和预设值, 119  
  设置控制字节, 119  
  选择激活状态, 118  
  中断, 114  
  重设和启动操作, 117  
  状态字节, 120  
高速计数器 (HSC) 指令, 112  
  实例, 125  
高速计数器定义 (HDEF) 指令, 112  
高速脉冲输出, 46  
  操作, 126  
  改变脉冲宽度, 133  
更换  
  HSC中的预设值, 124  
  HSG中的当前值, 124  
更新, PWM波形, 128  
工具栏, 51  
工作区位置, EM 253位控模块, 266  
功率流, 子例行程序参数, 205  
功能, Modbus, 342  
功能块图, 参考FBD编辑器  
构建  
  程序, 49  
  网络, 222  
故障检测, 错误代码, 424  
故障诊断  
  S7-200硬件, 251  
  非严重错误, 56  
  严重错误, 57  
  指南, 251  
关闭延迟计时器指令 (TOF), 197, 202  
  实例, 200  
规范  
  CP 243-1 IT互联网模块, 408  
  CP 243-1以太网模块, 405  
  CP 243-2 AS接口模块, 410  
  CPU模块, 351  
  EM 231 RTD, 373  
  EM 231热电偶, 373  
  EM 241调制解调器, 397  
  EM 277 PROFIBUS-DP, 385  
  Micro PLC系统, 48  
  RS-232/PPI多台主设备电缆, 413  
  分立扩充模块, 357  
  模拟扩充模块, 363

国际标准, 348  
 国家标准, 348  
 国家代码, EM 241支持, 298  
 过滤  
   模拟输入, 42  
   数字输入, 41

## H

海事代理认证, 348  
 横杆  
   安装, 16  
   尺寸, 16  
 后进先出指令, 实例, 192  
 后人先出指令, 191  
 互联网模块, 向导, 409  
 互联网设备, CP 243-1 IT, 219  
 环境规范  
   操作, 350  
   传输和存储, 350  
 环路控制  
   (PID) 指令, 146–157  
   程序实例, 153–155  
   错误条件, 152  
   调整偏差, 151  
   范围/变量, 151  
   模式, 152  
   选择类型, 149  
   正向/反向, 151  
   转换输出, 151  
   转换输入, 150  
 环路输出, 转换到成比例的整数, 151  
 环路输入  
   标准化, 150  
   转换, 150  
 缓冲区一致性, PROFIBUS, 390  
 恢复  
   EEPROM数据, 35  
   丢失密码, 45  
   内存磁带的程序, 37  
 回叫, EM 241调制解调器模块, 302  
 汇合控制  
   实例, 178  
   顺序控制继电器指令, 177  
 获得端口地址指令, 89  
 积分项, PID算法, 148

## J

激活输出 (ENO), 55  
 激活输入 (EN), 55  
 计时器分辨率, 198, 199  
 计时器内存区域 (T), 26  
 计时器指令  
   IEC  
     打开延迟计时器 (TON), 202  
     关闭延迟计时器 (TOF), 202  
     脉冲计时器 (TP), 202  
   SIMATIC  
     保留性打开延迟计时器 (TONR), 197

    打开延迟计时器 (TON), 197  
     关闭延迟计时器 (TOF), 197  
   中断, 159  
 计数器, 高速, 46  
 计数器内存区域 (C), 26  
 计数器指令  
   IEC  
     向上/向下计数器, 110  
     向上计数器, 110  
     向下计数器, 110  
   SIMATIC  
     向上/向下计数计数器, 107  
     向上计数计数器, 107  
     向下计数计数器, 107  
   高速计数器 (HSC), 112  
   高速计数器定义 (HDEF), 112  
 计算  
   电源要求, 419–421  
   令牌轮转时间, 235  
 继电器, 20  
   使用寿命, 349  
 夹头, 内存, 412  
 加指令, 141  
 监控, 12  
   程序状态, 248  
   过程变量, 59  
   状态图的变量, 249  
 监视程序重设指令, 169  
   实例, 170  
 间接编址, 32  
   &和\*, 32  
   修改指针, 32  
 间隙刷新因子 (GUF), 235  
 间隙要求, 14  
 兼容性  
   EM 231 RTD, 374  
   EM 231热电偶, 374  
   EM 241调制解调器, 398  
   EM 253位控, 401  
   EM 277 PROFIBUS-DP, 386  
 减指令, 141  
 交叉引用表, 246  
 交换字节指令, 184  
   实例, 184  
 接地, 18, 19  
 接点指令, 66  
   实例, 68  
 接口, 选择通讯, 210  
 接收数据, 86  
 接收指令, 80  
   SMB86到SMB94, SMB186到SMB194, 436  
   断开检测, 84  
   接收数据, 81  
   结束条件, 83  
   结束字符检测, 85  
   开始条件, 83  
   空闲行检测, 83  
   奇偶校验出错, 86  
   起始字符检测, 83  
   实例, 87  
   讯息计时器, 85  
   用户终端, 86

自由端口模式, 80  
 字符间计时器, 85  
 最大字符计数, 86

接头  
 CP 243-1 IT互联网模块, 409  
 CP 243-2, 411

接线盒连接器  
 删除, 17  
 重新安装, 17

截尾指令, 95  
 结束指令, 169  
 结束字符检测, 85  
 解码指令, 106  
 实例, 106

进程图像寄存器, 39  
 进程图像输出寄存器 (Q), 25  
 进程图像输入寄存器 (I), 25  
 进位指令, 95  
 禁用, 高速计数器, 124  
 禁用中断指令, 156  
 局部变量表, 51, 59  
 局部内存区域 (L), 28  
 局部I/O, 31

绝缘  
 布线指南, 18  
 网络, 222

## K

空闲行检测, 83  
 空转调制解调器适配器, 233  
 控制逻辑, 22  
 控制字节, 为HSC进行设置, 119  
 库, 指令, 60  
 快速参考手册信息, 451

扩充模块, 3  
 安装, 16  
 编址I/O点, 31  
 标识号和错误寄存器, 431  
 拆除, 17  
 尺寸, 16  
 电源要求, 15, 419  
 订购号, 441

分立  
 布线图, 360–362  
 输出规范, 359  
 输入规范, 358  
 通用规范, 357

模拟  
 布线图, 364  
 订购号, 363, 373  
 规范, 363  
 输出规范, 364  
 输入规范, 363

数字, 订购号, 357

扩展电缆, 412  
 扩展I/O, 31  
 总线出错 (SMW98), 437

## L

劳埃德船舶注册社 (LRS) 海事代理, 348

累加器, 27

离散模块, 3  
 立即复位指令, 69  
 立即接点指令, 66  
 立即输出指令, 69  
 立即置位指令, 69  
 立即重设指令, 69

连接  
 CP 243-1以太网模块, 406  
 MM3驱动器, 334  
 MM4驱动器, 337  
 S7-200, 8  
 到S7-200的调制解调器, 233  
 多主设备电缆, 7  
 网络设备, 237  
 无线电调制解调器到S7-200, 234

连接器, 订购号, 442  
 连接器插针, 通讯端口分配, 223  
 连接器端子  
 CPU 221 AC/DC/继电器, 354  
 CPU 221 DC/DC/DC, 354  
 CPU 222 AC/DC/继电器, 355  
 CPU 222 DC/DC/DC, 355  
 CPU 224 AC/DC/继电器, 355  
 CPU 224 DC/DC/DC, 355  
 CPU 226 AC/DC/继电器, 356  
 CPU 226 DC/DC/DC, 356  
 CPU 226XM AC/DC/继电器, 356  
 CPU 226XM DC/DC/DC, 356  
 EM 221 DI 8 x AC, 361  
 EM 221 DI 8x24 VDC, 361  
 EM 221DI 8x24 VDC, 361  
 EM 222 DO 8 x 继电器, 361  
 EM 222 DO 8x24 VDC, 361  
 EM 223 4x24 VDC 进/4x24 VDC出, 360  
 EM 223 4x24 VDC进/4x24 VDC出, 360  
 EM 223 DI 16/DO 16 x 24 VDC 继电器, 362  
 EM 223 DI 16/DO 16 x DC 24V, 362  
 EM 223 DI 4/DO 4 x DC 24V/继电器, 360, 361  
 EM 223 DI 8/DO 8 x 24 VDC, 362  
 EM 223 DI 8/DO 8 x 24 VDC/继电器, 362  
 EM 231 AI 4 x 12位, 364  
 EM 231 RTD, 374  
 EM 231热电偶, 374  
 EM 232 AQ 2 x 12 位, 364  
 EM 235 AI 4/AQ 1 x 12位, 364

令牌传递网络, 实例, 236  
 令牌轮转时间, 235  
 比较, 237

流水线操作, PTO脉冲, 127

逻辑, 控制, 22  
 逻辑操作指令  
 AND (与)、OR (或)、XOR (异或), 164  
 反转, 163

逻辑出栈指令, 71  
 逻辑读取指令, 71



- 逻辑堆栈指令
    - AENO, 71
    - AND (与) 载入, 71
    - OR (或) 载入, 71
    - 逻辑出栈, 71
    - 逻辑读取, 71
    - 逻辑进栈, 71
    - 实例, 72
    - 载入堆栈, 71
  - 逻辑进栈指令, 71
  - 逻辑连接
    - MPI, 215
    - PPI, 214
- ## M
- 脉冲捕获, 41
  - 脉冲捕获功能, 42
  - 脉冲计时器 (TP), 202
  - 脉冲宽度调制指令 (PWM), 更新方法, 128
  - 脉冲宽度调制指令 (PWM), 46, 126
    - 步进电机控制, 131
    - 初始化, 133
    - 改变脉冲宽度, 133
    - 概要表数值, 131
    - 了解, 128
    - 实例, 133
    - 位置控制向导, 126
    - 用SM内存配置, 129
    - 周期时间, 128
  - 脉冲链输出指令 (PTO), 定位控制向导, 126
  - 脉冲链输出指令 (PTO), 46, 126
    - 步进电机控制, 131
    - 初始化单段操作, 135
    - 初始化多段操作, 136
    - 单段流水线操作, 127
    - 多段流水线操作, 127
    - 改变时钟脉冲计数, 135
    - 改变周期时间, 135
    - 改变周期时间和脉冲计数, 136
    - 概要表数值, 131
    - 了解, 127
    - 实例, 137, 139
    - 用SM内存配置, 129
    - 周期时间, 127
  - 脉冲输出
    - 高速, 46
    - 脉冲宽度调制指令 (PWM), 126
    - 脉冲链输出指令 (PTO), 126
    - 脉冲输出指令 (PLS), 126
  - 脉冲输出指令 (PLS), 126
  - 脉冲与定向输出, EM 253位控模块, 257
  - 慢进参数
    - EM 253位控模块, 259
    - 慢进运行, 259
  - 美国船舶局 (ABS) 海事代理, 348
  - 密码
    - CPU功能, 44
    - 恢复丢失, 45
    - 配置, 44
    - 清除, 45
    - 限制存取, 45
    - 密码保护, EM 241调制解调器模块, 301
    - 面板安装, 16
    - 命令字节, EM 253位控模块, 293
    - 模块出错代码, EM 253位控模块, 287
    - 模拟程序, 订购号, 443
    - 模拟程序, 输入, 417
    - 模拟调整
      - SMB28和SMB29, 432
      - 电位器, 45
    - 模拟模块, 3
      - EM 231 RTD, 373
      - EM 231模拟输入, 366
      - EM 231热电偶, 373
      - EM 232模拟输出, 370
      - EM 235模拟I/O, 367
    - 模拟输出 (AQ), 编址, 29
    - 模拟输入 (AI)
      - 编址, 29
      - 过滤, 42
    - 模式
      - PID循环, 152
      - 高速计数器, 114
    - 模式转换, 37
- ## N
- 内存
    - CPU, 2
      - 保留, 41
      - 从EEPROM恢复, 35
      - 存取, 24
    - 内存保留, 34–37
      - EEPROM, 34–36
      - 电池盒 (供选用), 34
      - 范围, 41
      - 通电, 35–37
    - 内存磁带, 34, 36, 412
      - 安装, 36
      - 错误代码, 424
      - 订购号, 442
      - 复制程序, 36
      - 恢复程序, 37
    - 内存范围
      - CPU模块, 64, 482
      - 快速参考手册, 453
    - 内存功能
      - 成块移动指令, 168
      - 交换指令, 184
      - 循环指令, 180
      - 移动指令, 166
      - 移位寄存器位指令, 182
      - 移位指令, 180
    - 内存区, 清除, 45
    - 内存区域, 操作数范围, 65
    - 内存填充指令, 193
      - 实例, 193
    - 扭矩速度, 用于电机的通常情况, 258
    - 挪威船级社 (DNV) 海事代理, 348

**P****配置**

- CP 243-1 IT互联网模块, 409
- CP 243-1以太网模块, 406
- EM 231, 366
- EM 231 RTD, 380
- EM 231热电偶, 375
- EM 235, 367
- EM 253位控模块, 285
- EM 277 PROFIBUS-DP, 388-389
- FBD和LAD状态显示, 248
- Modbus符号表, 341
- PTO/PWM操作, 129
- STL状态显示, 249
- 超级终端, 241-485
- 密码, 44
- 内存的保留范围, 41
- 输出状态, 40
- 网络, 222
- 配置 / 概要表, EM 253位控模块, 288
- 配置表, EM 241调制解调器模块, 303, 316
- 配置图, 49
- 偏流**
  - PID环路, 148
  - PID循环, 146
- 偏置, 网络电缆, 224
- 平方根指令, 144

**Q**

- 七段显示, 96
- 奇偶校验出错
  - SMB3, 429
  - SMB30和SMB130, 86
- 起始字符检测, 83
- 启动**
  - STEP 7-Micro/WIN, 7
  - 高速计数器, 117
- 启用中断指令, 156
- 嵌入的变量, 在文字和SMS讯息中, 301
- 嵌套, 子例行程序, 204
- 强制数值, 249
- 强制值, 250
- 驱动器, 参见MicroMaster驱动器
- 驱动器通讯, 计算时间要求, 323
- 全局变量表, 58

**R**

- 热电偶模块 (EM 231)
  - 基础, 375
  - 配置, 375
  - 温度范围, 378-379
  - 状态指示器, 377
- 日本海事协会 (NK) 海事代理, 348
- 日期, 设置, 74
- 日时 (TOD) 时钟, 74
- 软件调试, 245

**S**

- 扫描时间: SMW22到SMW26, 432
- 扫描循环, 22
  - 计时器, 199
  - 指定数目, 250
- 删除**
  - 接线盒连接器, 17
  - 内存磁带, 36
- 上载, 程序, 34
- 设备数据库文件 (GSD), EM 277 PROFIBUS-DP, 393-394
- 设备要求, 3
- 设计**
  - Micro PLC系统, 48
  - 安全电路, 48
- 设置**
  - HSC的当前值和预设值, 119
  - S7-200的远程地址, 213
  - 波特率, 211
  - 控制字节 (HSC), 119
  - 日期, 74
  - 时间, 74
- 设置端口地址指令, 89
- 设置实时时钟指令, 74
- 设置主要双稳态指令, 73
- 十六进制到ASCII指令, 97
- 时基中断, 159
- 时间, 设置, 74
- 时钟**
  - 夹头, 412
  - 状态位, 428
- 时钟指令**
  - 读取实时时钟, 74
  - 设置实时时钟, 74
- 实例**
  - AND (与) 指令, 165
  - ASCII到十六进制指令, 100
  - DIV指令, 143
  - EM 241调制解调器模块, 313
  - for-next循环指令, 172
  - IEC计时器, 202
  - IEC计数器指令, 110
  - Modbus从属协议, 编程, 346
  - MUL指令, 143
  - OR (或) 指令, 165
  - PID程序, 153
  - PID循环指令, 154
  - SIMATIC 计时器, 200
  - SIMATIC计时器, 199, 201
  - SIMATIC计数器, 109
  - USS协议程序, 332
  - XOR (异或) 指令, 165
  - 向上 / 向下计数计数器指令, 109
  - 保留性打开延迟计时器指令, 201
  - 比较指令, 90
  - 编码指令, 106
  - 标准转换指令, 95
  - 表查找指令, 196
  - 成块移动指令, 168
  - 传输指令, 87
  - 从子例行程序返回指令, 207

- 打开延迟计时器指令, 199
- 单段PTO, 137
- 递减指令, 145
- 递增指令, 145
- 定时中断例行程序, 162
- 段指令, 96
- 多段PTO, 139
- 反转指令, 163
- 复位指令, 69
- 高速计数器模式, 115
- 高速计数器指令, 125
- 关闭延迟计时器指令, 200
- 后进先出指令, 192
- 计算电源要求, 419
- 监视程序重设指令, 170
- 交换指令, 184
- 接点指令, 68
- 接收指令, 87
- 解码指令, 106
- 控制流的汇合, 178
- 控制流的扩散, 177
- 令牌传递网络, 236
- 逻辑堆栈指令, 72
- 脉冲宽度调制 (PWM), 133
- 内存填充指令, 193
- 实数到ASCII指令, 100
- 实数运算指令, 142
- 顺序控制继电器指令, 174
- 添加到表格指令, 190
- 跳转至标签指令, 173
- 停止指令, 170
- 网络读取 / 写入指令, 78
- 位控模块, 279~283
- 先入先出指令, 191
- 向下计数计数器指令, 109
- 循环指令, 181
- 移动指令, 184
- 移位寄存器位指令, 183
- 移位指令, 181
- 有条件结束指令, 170
- 有条件跳转, 179
- 整数到ASCII指令, 100
- 整数运算指令, 142
- 置位指令, 69
- 中断程序, 49
- 中断指令, 162
- 子程序, 49
- 子例行程序调用, 205
- 子例行程序指令, 207
- 自由端口模式, 227
- 实时时钟指令, 74
- 实数到ASCII指令, 99
- 实例, 100
- 实数运算指令, 实例, 142
- 实数值, 24, 29
- 手册, 订购号, 442
- 手动模式, PID循环, 152
- 输出, 22
  - CPU模块, 354
  - EM 253位控模块, 254
  - 分立扩充模块, 360
  - 配置状态, 40
  - 指令, 55
- 输出方框图
  - EM 232, 370
  - EM 235, 370
- 输出规范
  - CPU模块, 353
  - 分立扩充模块, 359
  - 模拟扩充模块, 364
- 输出和继电器, 20
- 输出配置, 编辑位控模块中的缺省值, 257
- 输出数据字格式
  - EM 232, 369
  - EM 235, 369
- 输出图像寄存器, 22
- 输出指令, 69
- 输入, 22, 23
  - CPU模块, 354
  - EM 253位控模块, 254
  - 分立扩充模块, 358
  - 高速计数器, 114
  - 过滤模拟, 42
  - 过滤数字, 41
  - 启动和重设 (HSC), 117
  - 条件/无条件, 55
  - 校准, 365
  - 指令, 10
- 输入/输出中断, 159
- 输入/输出中断排队, 160
- 输入方框图
  - EM 231, 368
  - EM 235, 369
- 输入规范
  - CPU模块, 352
  - 分立扩充模块, 358
  - 模拟扩充模块, 363
- 输入过滤器时间, EM 253位控模块, 257
- 输入模拟程序, 417
- 输入配置, 编辑EM 253位控模块中的缺省值, 257
- 输入数据字格式
  - EM 231, 368
  - EM 235, 368
- 输入图像寄存器, 23
- 输入现用级别, EM 253位控模块, 257
- 书签, 246
- 数据
  - 保存和恢复, 34
  - 传输, 81
  - 接收, 81, 86
- 数据传送, EM 241调制解调器模块, 301
- 数据交换模式, EM 277 PROFIBUS-DP, 390
- 数据块, 50
- 数据块编辑器
  - 分配初始值, 58
  - 分配地址, 58
- 数据类型, 子例行程序参数, 205
- 数据一致性, PROFIBUS, 390
- 数字, 表达式, 30
- 数字, 表达式, 24, 29
- 数字扩充模块, 编址, 31
- 数字输出, 配置状态, 40
- 数字输入
  - 读, 39

过滤, 41  
 脉冲捕获, 41  
 数字寻呼, EM 241调制解调器模块, 300  
 数字指令  
   平方根, 144  
   余弦, 144  
   正切, 144  
   正弦, 144  
   自然对数, 144  
   自然指数, 144  
 双整数到ASCII指令, 98  
 双整数至实数指令, 94  
 双整数至整数指令, 94  
 顺序, 中断事件的, 161  
 顺序控制继电器结束指令, 174  
 顺序控制继电器内存区域 (S), 29  
 顺序控制继电器指令  
   分散控制, 177  
   汇合控制, 177  
   实例, 174  
   顺序控制继电器结束, 174  
   顺序控制继电器转换, 174  
   限制, 174  
   有条件顺序控制继电器结束, 174  
   载入顺序控制继电器, 174  
 顺序控制继电器转换指令, 174  
 搜索参数, EM 253位控模块, 260

## T

套管, 订购号, 442  
 特殊内存  
   EM 241调制解调器模块, 314  
   Modbus从属协议, 340  
 特殊内存区 (SM), 模拟调整电位器, 45  
 特殊内存区域 (SM), 28  
 特殊内存位, 428–436  
   快速参考手册, 451  
 特殊内存位置, EM 253位控模块, 291  
 特殊内存字节, EM 277 PROFIBUS-DP, 391  
 特性  
   CPU模块, 482  
   EM 241调制解调器模块, 298  
   EM 253位控模块, 254  
 特征, CPU模块, 64  
 梯形图逻辑, 参考LAD编辑器  
 添加到表格指令, 190  
   实例, 190  
 填充指令, 193  
 条件输入, 55  
 跳动时间, EM 253位控模块, 260  
 跳转到标签指令, 173  
 跳转指令  
   标签, 173  
   跳转至标签, 173  
 跳转至标签指令, 实例, 173  
 停止指令, 169  
   实例, 170  
 通电, 内存保留, 35–37  
 通讯, 210  
   Modbus从属协议, 340

S7-200, 8  
 波特率, 233, 234  
 冲突, 238  
 调制解调器, 233–234  
   接口, 选择, 210  
   所支持的协议, 214  
   无线电调制解调器, 234–235  
   与MicroMaster驱动器, 322  
 通讯处理器  
   CP 243-1 IT互联网模块, 408  
   CP 243-1以太网模块, 405  
   CP 243-2 AS接口, 410  
   参见 CP卡  
 通讯端口  
   连接器插针分配, 223  
   中断, 159  
   自由端口模式, 227  
 通讯进程, 40  
 通讯卡, 订购号, 442  
 通讯排队, 160  
 通讯请求, 处理, 23  
 通讯设置, STEP 7-Micro/WIN, 8  
 通讯协议  
   PROFIBUS, 215, 237  
   TCP/IP, 215  
   点对点接口 (PPI), 214  
   点至点接口 (PPI), 237  
   多点接口 (MPI), 215, 237  
   选择, 214  
   自定义, 227  
 通讯指令  
   传输, 80  
   获得端口地址, 89  
   接收, 80  
   设置端口地址, 89  
   网络读取, 75  
   网络写入, 75  
 同步更新, PWM指令, 128  
 图, 创建配置, 49

## W

网络  
   CP卡, 225  
   HMI设备, 226  
   MPI, 大于187.5 Kbaud, 217  
   MPI, 低于187.5 Kbaud, 217  
   PPI多台主设备电缆, 225  
   PROFIBUS, 211  
   PROFIBUS-DP, 218  
   波特率, 222  
   传输率, 222  
   从属装置, 211  
   单台主设备PPI, 216  
   地址, 211  
   电缆, 222, 223  
   调制解调器, 233  
   多台主设备PPI, 216  
   范例配置, 216, 217, 218  
   复杂, 238  
   复杂PPI, 217

- 构建, 222
  - 计算距离, 222
  - 间隙刷新因子 (GUF), 235
  - 绝缘, 222
  - 令牌轮转时间, 235
  - 配置准则, 222
  - 偏置电缆, 224
  - 设备地址, 214
  - 搜索CPU, 213
  - 通讯端口, 223
  - 通讯设置, 210–433
  - 无线电调制解调器, 234
  - 优化性能, 235
  - 中继器, 222
  - 终端电缆, 224
  - 主设备, 211
  - 组件, 222–225
  - 最高站地址 (HSA), 235
  - 网络读取指令, 75, 76
    - 出错代码, 76
    - 实例, 78
  - 网络写入指令, 75, 76
    - 出错代码, 76
    - 实例, 78
  - 微分项, PID算法, 149
  - 位存储器区域 (M), 25
    - 断电时保存, 35
  - 位控模块
    - ACCEL\_TIME, 259
    - DECEL\_TIME, 259
    - EM 253控制面板, 284–286
    - MAX\_SPEED, 258
    - POSx\_CFG, 278
    - POSx\_CLR, 277
    - POSx\_CTRL, 268
    - POSx\_DIS, 276
    - POSx\_GOTO, 270
    - POSx\_LD OFF, 273
    - POSx\_LD POS, 274
    - POSx\_MAN, 269
    - POSx\_RSEEK, 272
    - POSx\_RUN, 271
    - POSx\_SRATE, 275
    - RP查找模式, 263–267
    - RP搜索顺序, 261
    - SS\_SPEED, 258
    - 编程, 255
    - 参考点, 260
    - 参考点搜索, 260
    - 测量类型, 选择, 256
    - 出错代码, 286
    - 创建指令, 296
    - 负极, 257
    - 概要图, 262
    - 概要图步进, 263
    - 概要图操作模式, 262
    - 控制的范例程序, 296
    - 脉冲与定向输出, 257
    - 慢进参数, 259
    - 模块出错代码, 287
    - 配置, 256, 285
    - 配置 / 概要表, 288
    - 输入过滤器时间, 257
    - 输入和输出, 254
    - 输入和输出配置, 257
    - 输入现用级别, 257
    - 特殊内存, 291
    - 特性, 254
    - 跳动时间, 260
    - 位置控制向导, 256
    - 显示和控制运行, 284
    - 消除反冲, 266
    - 移动概要图, 定义, 262
    - 移动命令, 294
    - 诊断信息, 285
    - 正极, 257
    - 指令, 267
    - 指令出错代码, 286
    - 指令使用指南, 267
  - 位逻辑指令
    - 接点指令, 66
    - 逻辑堆栈指令, 71
    - 设置 / 重设双稳态, 73
    - 线圈指令, 69
    - 重设主要双稳态, 73
  - 位置控制向导, 256
  - 温度范围
    - EM 231 RTD, 383–384
    - EM 231热电偶, 378–379
  - 文字寻呼, EM 241调制解调器模块, 300
  - 文字讯息格式, EM 241调制解调器模块, 319
  - 无操作指令, 69
  - 无条件输入, 55
  - 无线电调制解调器
    - PPI/自由端口模式, 234
    - PPI模式, 234
    - 使用PPI多台主设备电缆, 234
  - 物理输入, EM 253位控模块的响应, 258
- ## X
- 系统块, 50
  - 系统设计, Micro PLC, 48
  - 系统支持, 用于中断例行程序, 158
  - 下载, 程序, 12, 34
  - 先入先出指令, 191
    - 实例, 191
  - 显示
    - 程序单元, 51
    - 程序状态, 248
  - 显示面板
    - TD 200文本显示, 4
    - TP070触摸面板, 4
  - 线圈指令
    - 复位, 69
    - 立即复位, 69
    - 立即输出, 69
    - 立即置位, 69
    - 输出, 69
    - 无操作, 69
    - 置位, 69

## 向导

- AS-i, 410
- PID, 146
- TD 200, 4
- 调制解调器扩充, 231, 304
- 定位控制, 256
- 互联网, 409
- 以太网, 406
- 指令, HSC, 113
- 指令, NETR, NETW, 75
- 向上 / 向下计数计数器指令, 107
- 实例, 109
- 向上 / 向下计数计数指令, 107
- 向上 / 向下计数器指令, 110
- 向上计数计数器指令, 107
- 向上计数器指令, 110
- 向下计数计数器指令, 107
- 实例, 109
- 向下计数器指令, 110
- 向右循环双字指令, 180
- 向右循环字节指令, 180
- 向右循环字指令, 180
- 向右移位双字指令, 180
- 向右移位字节指令, 180
- 向右移位字指令, 180
- 向左循环双字指令, 180
- 向左循环字节指令, 180
- 向左循环字指令, 180
- 向左移位双字指令, 180
- 向左移位字节指令, 180
- 校准
  - EM 231, 365
  - EM 235, 365
  - 输入, 365
- 协议
  - PROFIBUS-DP, 386
  - 由STEP 7-Micro/WIN支持, 226
- 协议.通讯
  - 选择, 214
  - 自定义, 227
- 性能, 优化网络, 235
- 修改, 指针, 32
- 选择
  - CP卡, 225
  - PPI多台主设备电缆, 225
  - RTD DIP开关, 380-381
  - S7-200操作模式, 37
  - 程序编辑器, 51
  - 热电偶DIP开关, 376
  - 通讯协议, 214
  - 指令集, 53
- 循环表, 153
- 循环指令, 180
  - 类型, 180
  - 实例, 181
- 寻呼, 调制解调器模块, 300
- 讯息, 令牌传递网络, 236
- 讯息传送电话号码格式, EM 241调制解调器模块, 318
- 讯息计时器, 85

## Y

- 哑元终端, 配置RS-232/PPI多台主设备电缆, 241-485
- 严重错误, 57
  - 查看, 424
- 要求, Modbus从属协议, 340
- 移动概要图, 用于EM 253位控模块的定义, 262
- 移动描述的概要图, EM 253位控模块, 262
- 移动命令, EM 253位控模块, 294
- 移动实数指令, 166
- 移动双字指令, 166
- 移动指令, 实例, 184
- 移动字节立即读指令, 167
- 移动字节立即写指令, 167
- 移动字节指令, 166
- 移动字指令, 166
- 移位寄存器位指令, 182
  - 实例, 183
- 移位指令
  - 类型, 180
  - 实例, 181
- 以太网, 214
  - TCP/IP, 215
- 以太网模块, 向导, 406
- 以太网网络, CP 243-1, 219
- 抑制电路, 20
- 异步更新, PWM指令, 128
- 硬件, 故障诊断, 251
- 映射地址到S7-200, Modbus, 341
- 永久程序存储, 38
- 用户库, 60
- 用户终端, 86
- 优化, 网络性能, 235
- 优先级
  - 中断例行程序, 160
  - 中断事件, 161
- 有条件结束指令, 169
  - 实例, 170
- 有条件顺序控制继电器结束指令, 174
- 有条件跳转, 实例, 179
- 余弦指令, 144
- 语句表, 参见 STL编辑器
- 语句表指令
  - 快速参考手册, 455
  - 执行次数, 445
- 预设值
  - 为HSC进行设置, 119
  - 在HSC中更换, 124
- 远程地址, S7-200的设置, 213
- 远程服务, 299
- 约定
  - 程序编辑器, 54
  - S7-200编程, 55
- 运算指令
  - 乘, 141
  - 乘以整数到双整数 (MUL), 143
  - 除, 141
  - 除以整数带余数 (DIV), 143
  - 递减, 145
  - 递增, 145
  - 加, 141
  - 减, 141

运行, 程序, 12  
 运行系统编程, 错误代码, 425  
 运行系统错误, 57

## Z

### 载入

HSC中的新当前值, 124  
 HSC中的新预设值, 124  
 载入堆栈指令, 71  
 载入顺序控制继电器指令, 174  
 在字符串中查找第一个字符指令, 188  
 在字符串中查找字符串指令, 188  
 噪声抑制, 输入过滤, 41  
 增量指针, 32  
 诊断, CPU自测, 23  
 诊断信息, EM 253位控模块, 285  
 整数到ASCII指令, 97  
   实例, 100  
 整数运算指令, 实例, 142  
 整数至双整数指令, 94  
 整数至字节指令, 94  
 整数至BCD指令, 94  
 正极, EM 253位控模块, 257  
 正切指令, 144  
 正弦指令, 144  
 正向转换指令, 66  
 正向作用循环, 151  
 执行  
   控制逻辑, 22  
   指令, 23  
 执行次数, 语句表指令, 445  
 执行错误, 57  
 执行错误代码, USS协议指令, 333  
 指令  
   AENO, 71  
   AND (与), 164  
   AND (与) 载入, 71  
   ASCII到十六进制, 97  
   BCD至整数, 94  
   EM 241调制解调器模块, 308  
   EM 253位控模块, 267  
   Exclusive OR (异或), 164  
   for, 171  
   MBUS\_INIT, 343  
   MBUS\_SLAVE, 345  
   Modbus从属协议, 342  
   MODx\_CTRL, 309  
   MODx\_MSG, 310  
   MODx\_XFR, 309  
   next, 171  
   NOT (非), 66  
   OR (或), 164  
   OR (或) 载入, 71  
   PID环路, 146  
   POSx\_CFG, 278  
   POSx\_CLR, 277  
   POSx\_CTRL, 268  
   POSx\_DIS, 276  
   POSx\_GOTO, 270  
   POSx\_LDOFF, 273

POSx\_LDPOS, 274  
 POSx\_MAN, 269  
 POSx\_RSEEK, 272  
 POSx\_RUN, 271  
 POSx\_SRATE, 275  
 USS协议, 324  
 保留性打开延迟计时器 (TONR), 197  
 比较, 55  
 编码, 106  
 标签, 173  
 标准接点, 66  
 表查找, 194  
 表格, 191, 196  
 并置字符串, 185  
 成块移动双字, 168  
 成块移动字, 168  
 成块移动字节, 168  
 乘, 141  
 乘以整数到双整数 (MUL), 143  
 除, 141  
 除以整数带余数 (DIV), 143  
 传输, 80  
 创建移动, 296  
 从中断返回, 156  
 从中断有条件返回, 156  
 从子例行程序返回, 204  
 从子例行程序有条件返回, 204  
 从字符串复制子字符串, 187  
 打开延迟计时器 (TON), 197, 202  
 到表格, 190  
 递减, 145  
 递增, 145  
 调用子例行程序, 204  
 读取实时时钟, 74  
 段, 96  
 反转双字, 163  
 反转字, 163  
 反转字节, 163  
 分离中断, 156  
 复位, 69  
 复制字符串, 185  
 负向转换, 66  
 附加中断, 156  
 高速计数器 (HSC), 112  
 高速计数器定义 (HDEF), 112  
 高速脉冲输出 (PLS), 126  
 关闭延迟计时器 (TOF), 197, 202  
 后入先出, 191  
 环路控制 (PID), 146  
 获得端口地址, 89  
 加, 141  
 监视程序重设, 169  
 减, 141  
 交换字节, 184  
 接收, 80  
 截尾, 95  
 结束, 169  
 解码, 106  
 进位, 95  
 禁用中断, 156  
 立即接点, 66  
 立即输出, 69

立即置位, 69  
 逻辑出栈, 71  
 逻辑读取, 71  
 逻辑进栈, 71  
 脉冲计时器 (TP), 202  
 脉冲宽度调制 (PWM), 126  
 脉冲链输出 (PTO), 126  
 脉冲输出 (PLS), 126  
 没有输出, 55  
 内存填充, 193  
 平方根, 144  
 启用中断, 156  
 设置端口地址, 89  
 设置实时时钟, 74  
 设置主要双稳态, 73  
 十六进制到ASCII, 97  
 实时时钟, 74  
 实数到ASCII, 99  
 输出, 69  
 输入, 10  
 双整数到ASCII, 98  
 双整数至实数, 94  
 双整数至整数, 94  
 顺序控制继电器结束, 174  
 顺序控制继电器转换, 174  
 跳转至标签, 173  
 停止, 169  
 网络读取, 75  
 网络写入, 75  
 位逻辑, 66  
 无操作, 69  
 先入先出, 191  
 向上 / 向下计数计数器, 107  
 向上 / 向下计数器, 110  
 向上计数计数器, 107  
 向上计数器, 110  
 向下计数计数器, 107  
 向下计数器, 110  
 向右循环双字, 180  
 向右循环字, 180  
 向右循环字节, 180  
 向右移位双字, 180  
 向右移位字, 180  
 向右移位字节, 180  
 向左循环双字, 180  
 向左循环字, 180  
 向左循环字节, 180  
 向左移位双字, 180  
 向左移位字, 180  
 向左移位字节, 180  
 移动实数, 166  
 移动双字, 166  
 移动字, 166  
 移动字节, 166  
 移动字节立即读, 167  
 移动字节立即写, 167  
 移位寄存器位, 182  
 有条件结束, 169  
 有条件顺序控制继电器结束, 174  
 余弦, 144  
 载入堆栈, 71  
 载入顺序控制继电器, 174

在字符串中查找第一个字符, 188  
 在字符串中查找字符串, 188  
 整数到ASCII, 97  
 整数至双整数, 94  
 整数至字节, 94  
 整数至BCD, 94  
 正切, 144  
 正弦, 144  
 正向转换, 66  
 执行, 23  
 置位, 69  
 中断, 156–162  
 重设复位, 69  
 重设主要双稳态, 73  
 转换 双整数到字符串, 104  
 转换实数到字符串, 101, 104  
 转换双整数到字符串, 101  
 转换整数到字符串, 101, 104  
 转换子字符串到实数, 101, 104  
 转换子字符串到双整数, 101, 104  
 转换子字符串到整数, 101, 104  
 自然对数, 144  
 自然指数, 144  
 字符串长度, 185  
 字节至整数, 94  
 指令, 快速参考手册指南, 455  
 指令出错代码, EM 253位控模块, 286  
 指令集  
   IEC1131-3, 53  
   SIMATIC, 53  
   选择, 53  
 指令库, 60  
 指令树, 9, 51  
 指令向导  
   HSC, 113  
   NETR, NETW, 75  
 指南  
   安装, 14  
   布线, 18  
   垂直安装, 17  
   高振动环境, 17  
   接地和布线, 19  
   接地和电路, 18  
   绝缘, 18  
   设计Micro PLC系统, 48  
   为间接编址修改指针, 32  
   抑制电路, 20  
   用于EM 253位控模块的指令, 267  
   中断程序, 50  
   中断例行程序, 158  
   子程序, 50  
 指针, 间接地址, 32  
 置位指令, 69  
   实例, 69  
 智能模块, 3  
   支持的CPU, 313  
   状态 (SMB200到SMB549), 439  
 中断  
   分配给HSC, 120  
   高速计数器, 114



- 中断程序
  - 实例, 49
  - 指南, 50
- 中断例行程序, 23, 39
  - S7-200支持的类型, 159
  - 调用子例行程序从, 159
  - 队列, 160
  - 了解, 158
  - 上升/下降边缘, 159
  - 时基, 159
  - 输入/输出, 159
  - 通讯端口, 159
  - 系统支持, 158
  - 优先级, 160
  - 与主程序共享数据, 158
  - 指南, 158
- 中断事件
  - 快速参考手册, 452
  - 类型, 157
  - 优先级, 161
- 中断指令
  - 从中断有条件返回, 156
  - 分离中断, 156
  - 附加中断, 156
  - 禁用中断, 156
  - 启用中断, 156
  - 实例, 162
- 中继器
  - 订购号, 442
  - 网络, 222
- 重置, 高速计数器, 117
- 重置指令, 实例, 69
- 重置主要双稳态指令, 73
- 重新安装, 接线盒连接器, 17
- 重新启动, 严重错误之后, 57
- 周期时间 (PTO功能), 126
- 主设备, 211
- 转换
  - 环路输出到成比例的整数, 151
  - 环路输入, 150
- 转换实数到字符串指令, 101, 104
- 转换双整数到字符串指令, 101, 104
- 转换整数到字符串指令, 101, 104
- 转换指令, 66
  - ASCII, 97
  - ASCII到十六进制, 97
  - 编码, 106
  - 标准, 93
  - 段, 96
  - 截尾, 95
  - 解码, 106
  - 进位, 95
  - 十六进制到ASCII, 97
  - 实例, 95
  - 字符串, 101, 104
- 转换子字符串到实数指令, 101, 104
- 转换子字符串到双整数指令, 101, 104
- 转换子字符串到整数指令, 101, 104
- 状态
  - 监控程序, 248
  - 扫描结束, 248
  - 以LAD和FBD显示, 248
  - 以STL显示, 249
  - 执行, 248
- 状态图, 59
  - 监控数值, 249
  - 强制值, 250
- 状态字节, 高速计数器, 120
- 状态LED
  - EM 231 RTD, 382
  - EM 231热电偶, 377
  - EM 253位控模块, 401
  - EM 277 PROFIBUS-DP, 386, 392
- 准则, 网络配置, 222
- 子程序
  - 实例, 49
  - 指南, 50
- 子例行程序
  - 参数类型, 205
  - 带参数, 205
  - 功率流参数, 205
  - 嵌套, 204
  - 数据类型, 205
  - 中断例行程序调用, 159
- 子例行程序指令
  - 从子例行程序有条件返回, 204
  - 调用子例行程序, 204
  - 实例, 207
- 自定义协议, 自由端口模式, 227
- 自动模式, PID循环, 152
- 自然对数指令, 144
- 自然指数指令, 144
- 自由端口, 配置, PPI多台主设备电缆, 232
- 自由端口控制寄存器 (SMB30和SMB130), 432
- 自由端口模式
  - 传输和接收指令, 80
  - 从PPI改变, 81
  - 定义, 159
  - 接收字符 (SMB2), 429
  - 奇偶校验出错 (SMB3), 429
  - 启用, 80
  - 实例, 227
  - 自定义协议, 227
  - 字符中断控制, 86
  - RS-232标准, 228
- 自由端口通讯, 429
  - SMB30和SMB130, 432
- 字存取, 24
- 字符串长度指令, 185
- 字符串指令
  - 并置字符串, 185
  - 从字符串复制子字符串, 187
  - 复制字符串, 185
  - 在字符串中查找第一个字符, 188
  - 在字符串中查找字符串, 188
  - 字符串长度, 185

字符间计时器, 85  
字符中断控制, 86  
字节地址格式, 24  
字节和整型范围, 24  
字节一致性, PROFIBUS, 390  
字节至整数指令, 94  
字一致性, PROFIBUS, 390  
最大字符计数, 86  
最高站地址 (HSA) , 235  
浏览, 错误, 56  
浏览条, 51

**To**

SIEMENS ENERGY & AUTOMATION INC  
ATTN: TECHNICAL COMMUNICATIONS  
ONE INTERNET PLAZA  
PO BOX 4991  
JOHNSON CITY TN USA 37602-4991

**From (来自)**

Name (姓名): \_\_\_\_\_  
Job Title (职务): \_\_\_\_\_  
Company Name (公司名称): \_\_\_\_\_  
Street (街道): \_\_\_\_\_  
City and State (城市或州): \_\_\_\_\_  
Country (国家): \_\_\_\_\_  
Telephone (电话): \_\_\_\_\_

请选择您从事的行业:

- |                                |                                   |
|--------------------------------|-----------------------------------|
| <input type="checkbox"/> 汽车    | <input type="checkbox"/> 制药       |
| <input type="checkbox"/> 化学    | <input type="checkbox"/> 塑料       |
| <input type="checkbox"/> 电气机械  | <input type="checkbox"/> 纸浆和纸张    |
| <input type="checkbox"/> 食品    | <input type="checkbox"/> 纺织品      |
| <input type="checkbox"/> 仪表和控制 | <input type="checkbox"/> 运输       |
| <input type="checkbox"/> 非电气机械 | <input type="checkbox"/> 其它 _____ |
| <input type="checkbox"/> 石化    |                                   |





## S7-200内存范围和特征

描述	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
用户程序大小	4096个字节	4096个字节	8192个字节	8192个字节	16384个字节
用户数据大小	2048个字节	2048个字节	5120个字节	5120个字节	10240个字节
进程图像输入寄存器	I0.0到I15.7	I0.0到I15.7	I0.0到I15.7	I0.0到I15.7	I0.0到I15.7
进程图像输出寄存器	Q0.0到Q15.7	Q0.0到Q15.7	Q0.0到Q15.7	Q0.0到Q15.7	Q0.0到Q15.7
模拟输入（只读）	--	AIW0到AIW30	AIW0到AIW62	AIW0到AIW62	AIW0到AIW62
模拟输出（只写）	--	AQW0到AQW30	AQW0到AQW62	AQW0到AQW62	AQW0到AQW62
变量内存（V）	VB0到VB2047	VB0到VB2047	VB0到VB5119	VB0到VB5119	VB0到VB10239
局部内存（L） <sup>1</sup>	LB0到LB63	LB0到LB63	LB0到LB63	LB0到LB63	LB0到LB63
位存储器（M）	M0.0到M31.7	M0.0到M31.7	M0.0到M31.7	M0.0到M31.7	M0.0到M31.7
特殊内存（SM） 只读	SM0.0到SM179.7 SM0.0到SM29.7	SM0.0到SM299.7 SM0.0到SM29.7	SM0.0到SM549.7 SM0.0到SM29.7	SM0.0到SM549.7 SM0.0到SM29.7	SM0.0到SM549.7 SM0.0到SM29.7
计时器	256（T0到T255）	256（T0到T255）	256（T0到T255）	256（T0到T255）	256（T0到T255）
保持开延迟	1毫秒 10毫秒 100毫秒	T0、T64 T1到T4，和 T65到T68 T5到T31，和 T69到T95	T0、T64 T1到T4，和 T65到T68 T5到T31，和 T69到T95	T0、T64 T1到T4，和 T65到T68 T5到T31，和 T69到T95	T0、T64 T1到T4，和 T65到T68 T5到T31，和 T69到T95
开 / 关延迟	1毫秒 10毫秒 100毫秒	T32、T96 T33到T36，和 T97到T100 T37到T63，和 T101到T255	T32、T96 T33到T36，和 T97到T100 T37到T63，和 T101到T255	T32、T96 T33到T36，和 T97到T100 T37到T63，和 T101到T255	T32、T96 T33到T36，和 T97到T100 T37到T63，和 T101到T255
计数器	C0到C255	C0到C255	C0到C255	C0到C255	C0到C255
高速计数器指令	HC0、HC3、HC4 和HC5	HC0、HC3、HC4 和HC5	HC0到HC5	HC0到HC5	HC0到HC5
顺序控制继电器（S）	S0.0到S31.7	S0.0到S31.7	S0.0到S31.7	S0.0到S31.7	S0.0到S31.7
累加器寄存器	AC0到AC3	AC0到AC3	AC0到AC3	AC0到AC3	AC0到AC3
跳转 / 标签	0至255	0至255	0至255	0至255	0至255
调用 / 子例行程序	0至63	0至63	0至63	0至63	0至127
中断例行程序	0至127	0至127	0至127	0至127	0至127
正向 / 负向转换	256	256	256	256	256
PID循环	0至7	0至7	0至7	0至7	0至7
端口	端口0	端口0	端口0	端口0、端口1	端口0、端口1

<sup>1</sup> LB60□LB63□STEP 7-Micro/WIN□3.0□□□□□□□□

STL	页面
=	69
+D	141
-D	141
*D	141
/D	141
+I	141
-I	141
=I	69
*I	141
/I	141
+R	141
-R	141
*R	141
/R	141
A	66
AB < =	90
AB =	90
AB >	90
AB<	90
AB > =	90
AB <>	90
AD <	90
AD < =	90
AD =	90
AD >	90
AD > =	90
AD <>	90
AENO	71
AI	66
ALD	71
AN	66
ANDB	164
ANDD	164
ANDW	164
ANI	66
AR=	90
AR <	90
AR<=	90
AR >	90
AR>=	90
AR <>	90
AS=	92
AS<>	92
ATCH	156
ATH	97
ATT	190
AW <	90
AW < =	90

STL	页面
AW=	90
AW >	90
AW > =	90
AW <>	90
BCDI	93
BIR	167
BIW	167
BMB	168
BMD	168
BMW	168
BTI	93
CALL	204
CFND	188
COS	144
CRET	204
CRETI	156
CSCRE	174
CTD	107
CTU	107
CTUD	107
DECB	145
DECD	145
DECO	106
DECW	145
DISI	156
DIV	143
DTA	97
DTCH	156
DTI	93
DTR	93
DTS	101
ED	66
ENCO	106
END	169
ENI	156
EU	66
EXP	144
FIFO	191
FILL	193
FND <	194
FND <>	194
FND =	194
FND >	194
FOR	171
GPA	89
HDEF	112
HSC	112
HTA	97

STL	页面
IBCD	93
INCB	145
INCD	145
INCW	145
INVB	163
INVD	163
INVW	163
ITA	97
ITB	93
ITD	93
ITS	101
JMP	173
LBL	173
LD	66
LDB <=	90
LDB =	90
LDB >=	90
LDB >	90
LDB <	90
LDB <>	90
LDD >=	90
LDD <	90
LDD <=	90
LDD =	90
LDD >	90
LDD <>	90
LDI	66
LDN	66
LDNI	66
LDR=	90
LDR <	90
LDR<=	90
LDR >	90
LDR>=	90
LDR <>	90
LDS	71
LDS=	92
LDS<>	92
LDW <=	90
LDW <	90
LDW =	90
LDW >	90
LDW >=	90
LDW <>	90
LIFO	191
LN	144
LPP	71
LPS	71

STL	页面
LRD	71
LSCR	174
MOVB	166
MOVD	166
MOVR	166
MOVW	166
MUL	143
NEXT	171
NETR	75
NETW	75
NOT	66
O	66
OB =	90
OB > =	90
OB >	90
OB <	90
OB < =	90
OB <>	90
OD <	90
OD < =	90
OD =	90
OD >	90
OD > =	90
OD <>	90
OI	66
OLD	71
ON	66
ONI	66
OR=	90
OR <	90
OR<=	90
OR >	90
OR >=	90
OR <>	90
ORB	164
ORD	164
ORW	164
OS=	92
OS<>	92
OW <	90
OW < =	90
OW =	90
OW >	90
OW > =	90
OW <>	90
PID	146
PLS	126
R	69

STL	页面
RCV	80
RI	69
RLB	180
RLD	180
RLW	180
ROUND	93
RRB	180
RRD	180
RRW	180
RTA	97
RTS	101
S	69
SCAT	185
SCPY	185
SCRE	174
SCRT	174
SEG	93
SFND	188
SHRB	182
SI	69
SIN	144
SLB	180
SLD	180
SLEN	185
SLW	180
SPA	89
SQRT	144
SRB	180
SRD	180
SRW	180
SSCPY	187
STD	104
STI	104
STOP	169
STR	104
SWAP	184
TAN	144
TODR	74
TODW	74
TOF	197
TON	197
TONR	197
TRUNC	93
WDR	169
XMT	80
XORB	164
XORD	164
XORW	164