

SIMATIC

S7-200可编程控制器 系统手册

前言, 目录	
产品概述	1
使用入门	2
S7-200的安装	3
PLC的基本概念	4
编程的概念、惯例及特点	5
S7-200指令集	6
网络通讯	7
硬件故障诊断指导和软件调试工具	8
S7-200开环运动控制	9
创建调制解调模块程序	10
使用USS协议库去控制一个 MicroMaster驱动	11
使用Modbus协议库	12
使用配方	13
使用数据归档	14
PID自整定和PID整定控制面板	15
附录	

索引

本手册的订货号:
6ES7298-8FA24-8FH0

版本**06/2004**
A5E00307992-01

安全指南

本手册包括了保证人身安全与保护本产品及连接的设备应遵守的注意事项。这些注意事项在手册中以警告三角形加以突出，并按照危险等级标明如下：



危险

表示如果不采取适当的预防措施，将导致死亡或者严重的人身伤害。



警告

表示如果不采取适当的预防措施，将有导致死亡或严重人身伤害的可能。



小心

表示如果不采取适当的预防措施将有导致轻微的人身伤害的可能。

小心

表示如果不采取适当的预防措施将有导致财产损失的可能。

注意

表示如果不采取适当的预防措施，有可能导致不希望的结果或状态。

合格人员

只有合格人员才允许安装和操作设备。合格人员是指被授权按照既定安全惯例和标准，对线路、设备和系统进行调试，接地和加标识的人员。

正确应用

注意如下：



警告

该设备及其部件只能用于产品目录或者技术说明中所描述的范畴，并且只能与Siemens公司认可或者推荐的第三方厂家出产的设备或部件一起使用。

只有正确地运输、保管、配置和安装，并且按照推荐的方式操作和维护，产品才能正常、安全地运行。

注册商标

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

本文档中的其它一些标志也是注册商标；如果它们因个人目的而被第三方厂家所使用，商标所有者的权力将受到侵害。

Siemens AG 2004版权所有

未经明确的书面许可，不得复制、传抄或者使用本资料的内容。违者应对造成的损失承担责任。保留实用模块或设计的专利许可及注册中提供的所有权力。

拒负责任的声明

我们已核对过本手册的内容与所描述的硬件和软件相符。由于差错难以完全避免，我们不能保证完全一致。我们会经常对手册中的数据进行检查，并在后续的编辑中进行必要的更正。欢迎您提出宝贵意见。

前言

本手册的用途

S7-200系列小型PLC (Micro PLC) 可应用于各种自动化系统。紧凑的结构、低廉的成本以及功能强大的指令集使得S7-200 PLC成为各种小型控制任务理想的解决方案。S7-200产品的多样化以及基于Windows的编程工具, 使您能够更加灵活地完成自动化任务。

本手册由具有一定PLC背景知识的工程人员、编程人员、安装人员及电气人员编写, 其内容涵盖了S7-200系列小型PLC的安装和编程信息。

阅读本手册所需的基本知识

如果具备了一定的自动化知识和PLC知识, 那么您将能更好地理解本手册的内容。

适用范围

本手册适用于STEP 7-Micro/WIN 4.0版和S7-200 CPU产品系列。关于本手册涉及的全部S7-200系列产品的清单和订货号, 可参见附录E。

修订的内容

本手册的修订内容包括:

- S7-200 CPU包括CPU 221、CPU 222、CPU 224、CPU 224XP和CPU 226等型号, 它们有如下新特性:

新CPU硬件支持: 通过关闭在运行模式下编辑程序的可选功能来获取更多的程序存储区。CPU 224XP支持集成的模拟量I/O和两个通讯端口。CPU 226带有附加的输入滤波器和脉冲捕获功能。

- 新型存储卡支持: S7-200资源浏览器、存储卡的程序传递、比较以及编程选择

- STEP 7-Micro/WIN 4.0版是用于S7-200的32位编程软件包, 它包括:

支持最新型CPU增强功能的新软件工具和改进过的软件工具: PID整定控制面板、PLC内置位控向导、数据归档向导和配方向导。

新的诊断工具: 可组态诊断LED

新指令: 夏令时 (READ_RTCX和SET_RTCX)、间隔定时器 (BITIM、CITIM)、清除中断事件 (CLR_EVNT) 以及诊断LED (DIAG_LED)。

POU和库的增强: 新型字符串常量和添加的间接寻址支持更多的存储类型, 增强了使用USS库函数对西门子变频器读写参数功能的支持。

改进的数据块: 数据块页、数据块自动增量

更佳的STEP 7-Micro/WIN可用性

认证标准

SIMATIC S7-200系列产品符合以下标准:

- Underwriters Laboratories, Inc. UL 508 Listed (工业控制设备)
注册号 E75310
- Canadian Standards Association: CSA C22.2 Number 142 (过程控制设备)
- Factory Mutual Research: Class Number 3600, Class Number 3611, FM Class I, Division 2, Groups A, B, C, & D Hazardous Locations, T4A and Class I, Zone 2, IIC, T4



提示

SIMATIC S7-200系列产品符合CSA标准。

标志cULus表示该S7-200已通过了Underwriters Laboratories (UL) to standards UL 508和CSA 22.2 No. 142标准的测试和认证。

CE标签

SIMATIC S7-200符合下列EU directive的要求和保护条例:

- EC Directive 73/23/EEC “Low-voltage directive”
- EC Directive 89/336/EEC “EMC directive”

关于所用产品是否符合European Community ATEX Directive 94/9/EC标准, 可向当地Siemens代表处询问。

C标记

SIMATIC S7-200产品符合AS/NZS 2064 (澳大利亚) 标准。

标准:

SIMATIC S7-200系列产品符合IEC 61131-2, 可编程控制器 - 设备要求的标准。

更多信息请参见附录A。

本文档的相关资料信息

产品系列	文档	订货号
S7-200	TP070触摸屏用户手册 (英语)	6AV6 591-1DC01-0AB0
	TP170 micro操作手册 (英语)	6AV6 691-1DB01-0AB0
	S7-200点到点接口通讯手册 (英语/德语)	6ES7 298-8GA00-8XH0
	CP 243-2 SIMATIC NET AS接口主站手册 (英语)	6GK7 243-2AX00-8BA0
	CP 243-1 IT因特网模块 (CD上有电子文档)	6GK7 243-1GX00-0XE0
	CP 243-1以太网模块 (CD上有电子文档)	6GK7 243-1GX00-0XE0
	S7-200可编程控制器系统手册 (中文)	6ES7 298-8FA24-8FH0

如何使用本手册

如果您是初次使用S7-200产品，那么您需要通读*S7-200 可编程控制器系统手册*。如果您是一位有经验的用户，则可以通过目录和索引查找相应信息。

*S7-200 可编程控制器系统手册*按照以下主题组织编排：

- ❑ 第1章（产品概述）对S7-200系列PLC产品的特点作简单的描述。
- ❑ 第2章（使用入门）教您如何创建并下载一个简单的控制程序。
- ❑ 第3章（S7-200的安装）提供S7-200 CPU模块和可扩展I/O模块的安装尺寸和基本安装指南。
- ❑ 第4章（PLC的基本概念）提供S7-200的操作信息。
- ❑ 第5章（编程的概念、惯例及特点）描述STEP 7-Micro/WIN软件的特点、程序编辑器、指令集的种类（IEC 1131或者SIMATIC）、S7-200的数据类型和创建程序的步骤。
- ❑ 第6章（S7-200指令集）编程指令的描述及示例。
- ❑ 第7章（网络通讯）介绍S7-200支持的各种网络配置。
- ❑ 第8章（硬件故障处理方法及软件调试工具）介绍S7-200硬件故障的处理方法以及STEP 7-Micro/WIN软件中为您提供的调试工具。
- ❑ 第9章（S7-200开环运行控制）介绍开环运动控制的三种方式：脉宽调制、脉冲串输出和EM 253位控模块。
- ❑ 第10章（为Modem模块创建程序）介绍如何使用指令和向导为EM 241模块创建程序。
- ❑ 第11章（用USS协议控制MicroMaster变频器）不仅介绍了如何用指令来创建MicroMaster变频器控制程序，还介绍了如何配置MicroMaster系列第三代和第四代变频器。
- ❑ 第12章（使用Modbus协议库）介绍如何使用指令创建一个通过Modbus进行通讯的程序。
- ❑ 第13章（使用配方）介绍怎样将自动化配方程序组织并装载到存储卡中。
- ❑ 第14章（使用数据归档）提供关于将过程测量数据存储到存储卡中的信息。
- ❑ 第15章（PID自整定和PID整定控制面板）介绍如何通过这些功能来大幅度增强系统的效用，以及如何轻松使用S7-200提供的PID功能。
- ❑ 附录A（技术规范）提供S7-200硬件的技术信息和数据清单。

其它附录提供了更多的相关参考信息，例如对于错误代码、特殊存储器（SM）标志位，S7-200设备的订货号和STL指令执行时间等信息的描述。

除了本手册之外，STEP 7-Micro/WIN还提供了关于S7-200编程入门的在线帮助。如果您购买STEP 7-Micro/WIN软件，将免费得到一张资料光盘。光盘的内容包括应用示例、电子版的系统手册和其它信息。

在线帮助

帮助信息距您仅“一键之遥”！只要按下F1键您就可以在STEP 7-Micro/WIN软件中得到在线帮助信息。在线帮助不仅能使您在S7-200的编程过程中得到帮助，而且包括其它一些主题。

电子手册

在资料光盘中有电子版的S7-200系统手册。您可以将它安装在计算机上，以便在需要时随时使用。

应用示例

资料光盘中包括了一些应用示例的程序。这些示例程序在您编制自己的应用程序时是值得借鉴的。在S7-200的因特网网站上，您可以找到这些应用示例的最新版本。

回收和废弃

请联系专业公司帮助您处理电气垃圾，以避免在回收和废弃设备时对环境造成危害。

其它支持

当地Siemens销售处或经销商

若需技术支持或S7-200培训，或需订购S7-200产品，可联系当地的Siemens经销商或销售处。我们的销售代表受过专业的培训，具备一定的工业过程知识背景和各种Siemens产品方面的知识，他们能以最快捷和最高效的方式来解决您碰到的问题。

在线服务与支持

除了文档支持以外，我们还将通过因特网尽我们之所能，网址为：

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

您将在该网站上找到：

- 关于S7-200系列产品的信息www.siemens.com/S7-200
S7-200因特网站点，其内容包括常见问题解答（FAQ）、应用示例（应用示例和程序示例）、新发布产品的信息，以及产品的更新或下载。
- 时事通讯，通过它您能了解到所用产品的最新消息。
- 通过Service & Support的搜索功能获得最佳资料。
- 论坛，通过它您将能和全球各地的用户和专业人员交流经验。
- Automation & Drives的当地代表机构。
- 在“Services”栏下的，关于现场服务、维修、备件的信息及其它信息。

技术服务

在S7-200技术支持中心，经过高级技术培训的人员同样可以为您解决您有可能遇到的问题。您可以随时与他们联系。

A&D技术支持中心

全球服务、全日制服务:



全球 (Nuernberg) 技术支持中心 每天24小时, 全年365天为您服务 电话: +49 (180) 505022 传真: +49 (180) 505023 Email: adsupport@siemens.com GMT: +1: 00	美国 (Johnson City) 技术支持和授权中心 当地时间: 周一至周五 8: 00 AM到5: 00 PM 电话: +1 (423) 262 2522 +1 (800) 333-7421 (仅限美国) 传真: +1 (423) 262 2289 Email: simatic.hotline@sea.siemens.com GMT: -5: 00	亚洲/澳大利亚 (北京) 技术支持和授权中心 当地时间: 周一至周五 8: 00 AM到5: 00 PM 电话: +86 10 64 75 75 75 传真: +86 10 64 74 74 74 Email: adsupportv.asia@siemens.com GMT: +8: 00
欧洲/非洲 (Nuernberg) 授权中心 当地时间: 周一至周五 8: 00 AM到5: 00 PM 电话: +49 (180) 5050-222 传真: +49 (180) 505023 Email: adsupport@siemens.com GMT: +1: 00		
通常情况下, SIMATIC热线和授权热线的语言为德语和英语。		

目录

1	产品概述	1
	新内容	2
	S7-200 CPU	2
	S7-200扩展模块	3
	STEP 7-Micro/WIN编程软件	4
	通讯方式选择	4
	显示面板	5
2	使用入门	7
	连接S7-200 CPU	8
	创建一个例子程序	10
	下载例子程序	14
	将S7-200转入运行模式	14
3	S7-200的安装	15
	S7-200设备的安装指南	16
	S7-200模块的安装和拆卸	17
	接地和接线指南	20
4	PLC的基本概念	23
	理解S7-200如何执行您的控制逻辑	24
	S7-200数据的存取	26
	理解S7-200如何保存和存储数据	36
	为S7-200 CPU选择操作模式	41
	使用S7-200资源管理器	41
	S7-200的特性	42
5	编程的概念、惯例及特点	51
	设计一个微型PLC系统的指导原则	52
	程序的基本组件	53
	用STEP 7-Micro/WIN创建用户程序	55
	SIMATIC和IEC 1131-3指令集的选择	57
	理解程序编辑器中使用的惯例	58
	使用向导帮您创建控制程序	60
	S7-200中的出错处理	60
	在数据块中指定地址和初始值	62
	用符号表来定义变量的符号地址	62
	使用局部变量	63
	用状态图来监视用户程序	63
	创建一个指令库	64
	应用程序的调试	64

6	S7-200指令集	65
	用于描述指令的习惯用语	67
	S7-200存储器范围及特性	68
	位逻辑指令	70
	触点	70
	线圈	73
	逻辑堆栈指令	75
	RS触发器指令	77
	时钟指令	78
	通讯指令	81
	网络读写指令	81
	发送和接收指令	86
	获取口地址和设定口地址指令	95
	比较指令	96
	数值比较	96
	字符串比较	98
	转换指令	99
	标准转换指令	99
	ASCII码转换指令	103
	字符串转换指令	107
	编码和解码指令	112
	计数器指令	113
	SIMATIC计数器指令	113
	IEC计数器指令	116
	高速计数器指令	118
	脉冲输出指令	133
	数字运算指令	140
	加、减、乘、除指令	140
	整数乘法产生双整数和带余数的整数除法	142
	数学功能指令	143
	递增和递减指令	144
	比例/积分/微分 (PID) 回路控制指令	145
	中断指令	153
	逻辑操作指令	161
	取反指令	161
	与、或和异或指令	162
	传送指令	164
	字节、字、双字或者实数传送	164
	字节立即传送 (读和写)	165
	块传送指令	166
	程序控制指令	167
	条件结束	167
	停止	167
	看门狗复位	167
	For-Next循环指令	169
	跳转指令	171
	顺控继电器 (SCR) 指令	172
	诊断LED指令	178

移位和循环指令	179
右移和左移指令	179
循环右移和循环左移指令	179
移位寄存器指令	181
字节交换指令	183
字符串指令	184
表指令	189
填表	189
先进先出和后进先出	190
内存填充	192
查表	193
定时器指令	196
SIMATIC定时器指令	196
IEC定时器指令	201
时间间隔定时器	203
子程序指令	204
7 通过网络进行通讯	209
理解S7-200网络通讯的基本概念	210
为网络选择通讯协议	214
通讯接口的安装和删除	220
网络的建立	221
用自由口模式创建用户定义的协议	225
在网络中使用Modem和STEP 7-Micro/WIN	228
高级议题	233
使用RS-232/PPI多主站电缆进行远端操作	239
8 硬件故障诊断指导和软件调试工具	243
调试应用程序	244
显示程序状态	246
使用状态图来显示和修改S7-200中的数据	247
强制指定值	248
指定程序执行的扫描周期数	248
硬件故障诊断指导	249
9 S7-200开环运动控制	251
概述	252
使用PWM（脉宽调制）输出	253
开环位控用于步进电机或伺服电机的基本信息	255
通过位控向导创建指令	260
PTO指令的错误代码	264
位控模块的特性	265
组态位控模块	267
通过位控向导为位控模块创建指令	273
位控模块的示例程序	285

	使用EM253控制面板监控位控模块	290
	位控模块和位控指令的错误代码	292
	高级议题	294
	理解位控模块所支持的RP寻找模式	303
10	创建调制解调模块程序	307
	EM241调制解调模块特点	308
	利用调制解调扩展向导组态EM241调制解调模块	314
	Modem指令和限定概述	318
	EM241 Modem模块指令	319
	EM241 Modem模块示例	323
	支持智能模块的CPU	323
	Modem模块的特殊存储区	323
	高级议题	325
	信息电话号码格式	327
	文本信息格式	328
	CPU数据传送信息格式	329
11	使用USS协议库去控制一个MicroMaster驱动	331
	使用USS协议的要求	332
	计算与驱动通讯的时间要求	332
	使用USS指令	333
	USS协议指令	334
	USS协议示例程序	341
	USS执行错误代码	342
	连接并设置3系列MicroMaster驱动	342
	连接和设置4系列MicroMaster驱动	345
12	使用Modbus协议库	347
	使用Modbus协议的要求	348
	Modbus协议的初始化和执行时间	348
	Modbus地址	349
	使用Modbus从站协议指令	350
	Modbus从站协议指令	351
13	使用配方	355
	概述	356
	配方的定义和术语	357
	使用配方向导	357
	由配方向导创建的指令	361

14	使用数据归档	363
	概述	364
	使用数据归档向导	365
	由数据归档向导创建的指令	369
15	PID自整定和PID整定控制面板	371
	理解PID自整定	372
	扩展回路表	372
	先决条件	375
	自滞后和自偏移	375
	自整定序列	376
	例外情况	377
	关于过程变量超限的说明（结果代码3）	377
	PID整定控制面板	378
附录 A	技术规范	381
	通用技术规范	382
	CPU规范	385
	数字扩展模块规范	394
	模拟量扩展模块规范	400
	热电偶和RTD（热电阻）扩展模块规范	411
	EM 277 PROFIBUS-DP模板规范	423
	EM 241 Modem模板规范	435
	EM 253位控模块规范	437
	(CP 243-1) 以太网模块规范	443
	(CP 243-1 IT)因特网模块规范	445
	(CP 243-2) AS-I接口模块规范	448
	可选卡件	450
	扩展电缆	451
	RS-232/PPI多主站电缆和USB/PPI多主站电缆	452
	输入仿真器	456
附录 B	电源设计	457
附录 C	错误代码	461
	致命错误代码和信息	462
	运行程序错误	463
	编译规则错误	464

附录 D 特殊存储器 (SM) 标志位	465
SMB0: 状态位	466
SMB1: 状态位	466
SMB2: 自由口接收字符	467
SMB3: 自由口奇偶校验错误	467
SMB4: 队列溢出	467
SMB5: I/O状态	468
SMB6: CPU识别 (ID) 寄存器	468
SMB7: 保留	468
SMB8到SMB21: I/O模块识别和错误寄存器	469
SMW22到SMW26: 扫描时间	470
SMB28和SMB29: 模拟电位器	470
SMB30和SMB130: 自由端口控制寄存器	470
SMB31和SMW32: 永久存储器 (EEPROM) 写控制	471
SMB34和SMB35: 定时中断的时间间隔寄存器	471
SMB36到SMB65: HSC0、HSC1和HSC2寄存器	471
SMB66到SMB85: PTO/PWM寄存器	473
SMB86到SMB94和SMB186到SMB194: 接收信息控制	474
SMW98: 扩展I/O总线错误	475
SMB130: 自由口控制寄存器 (见SMB30)	475
SMB131到SMB165: HSC3、HSC4和HSC5寄存器	475
SMB166到SMB185: PTO0, PTO1包络定义表	476
SMB186到SMB194: 接收信息控制 (见SMB86-SMB94)	476
SMB200到SMB549: 智能模块状态	477
附录 E S7-200订货号	479
附录 F STL指令的执行时间	483
附录 G S7-200快速参考信息	489



产品概述

S7-200系列是一种可编程序逻辑控制器（Micro PLCs）。它能够控制各种设备以满足自动化控制需求。

S7-200的用户程序中包括了位逻辑、计数器、定时器、复杂数学运算以及与其它智能模块通讯等指令内容，从而使它能够监视输入状态，改变输出状态以达到控制目的。紧凑的结构、灵活的配置和强大的指令集使S7-200成为各种控制应用的理想解决方案。

本章内容:

新内容	2
S7-200 CPU	2
S7-200扩展模块	3
STEP 7-Micro/WIN编程软件	4
通讯方式选择	4
显示面板	5

新内容

SIMATIC S7-200的新特性包括下列内容。表1-1列出了支持这些新特性的S7-200 CPU。

- ❑ S7-200 CPU包括CPU 221、CPU 222、CPU 224、CPU 224XP和CPU 226等型号，它们有如下特性：

新CPU硬件支持：通过关闭在运行模式下编辑程序的可选功能来获取更多的程序存储区。CPU 224XP支持集成的模拟量I/O和两个通讯端口。CPU 226带有附加的输入滤波器和脉冲捕获功能。

- ❑ 新型存储卡支持；S7-200资源浏览器的使用、存储卡的转换、比较以及编程选择

- ❑ STEP 7-Micro/WIN 4.0版是用于S7-200的32位编程软件包，它包括：

支持最新CPU增强功能的新软件工具和改进过的软件工具：PID整定控制面板、PLC内置位控向导、数据归档向导和配方向导。

新的诊断工具：可组态诊断LED

新指令：夏令时（READ_RTCX和SET_RTCX）、间隔定时器（BITIM, CITIM）、清除中断事件（CLR_EVNT）以及诊断LED（DIAG_LED）。

POU和库的增强功能：新型字符串常量和添加的间接寻址支持更多存储类型，增强了使用USS库函数对西门子变频器读写参数功能的支持。

改进的数据块：数据块页、数据块自动增量

改进的STEP 7-Micro/WIN可用性

表1-1 S7-200 CPU

S7-200 CPU	订货号
CPU 221 DC/DC/DC 6输入/4输出	6ES7 211-0AA23-0XB0
CPU 221 AC/DC/继电器6输入/4继电器输出	6ES7 211-0BA23-0XB0
CPU 222 DC/DC/DC 8输入/6输出	6ES7 212-1AB23-0XB0
CPU 222 AC/DC/继电器8输入/6继电器输出	6ES7 212-1BB23-0XB0
CPU 224 DC/DC/DC 14输入/10输出	6ES7 214-1AD23-0XB0
CPU 224 AC/DC/继电器14输入/10继电器输出	6ES7 214-1BD23-0XB0
CPU 224XP DC/DC/DC 14输入/10输出	6ES7 214-2AD23-0XB0
CPU 224XP AC/DC/继电器14输入/10继电器输出	6ES7 214-2BD23-0XB0
CPU 226 DC/DC/DC 24输入/16输出	6ES7 216-2AD23-0XB0
CPU 226 AC/DC/继电器24输入/16继电器输出	6ES7 216-2BD23-0XB0

S7-200 CPU

S7-200 CPU将一个微处理器、一个集成电源和数字量I/O点集成在一个紧凑的封装中，从而形成了一个功能强大的微型PLC，参见图1-1。在下载了程序之后，S7-200将保留所需的逻辑，用于监控应用程序中的输入输出设备。

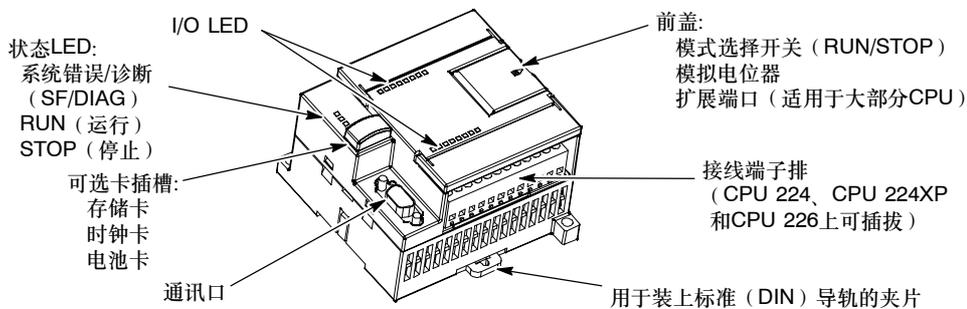


图1-1 S7-200微型PLC

Siemens公司提供多种类型的CPU以适应各种应用，表1-2中对各种CPU的特性作一简单比较，详细信息参见附录A。

表1-2 S7-200的技术指标

特性	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
外形尺寸 (mm)	90 x 80 x 62	90 x 80 x 62	120.5 x 80 x 62	140 x 80 x 62	190 x 80 x 62
程序存储器: 可在运行模式下编辑 不可在运行模式下编辑	4096字节 4096字节	4096字节 4096字节	8192字节 12288字节	12288字节 16384字节	16384字节 24576字节
数据存储区	2048字节	2048字节	8192字节	10240字节	10240字节
掉电保持时间	50小时	50小时	100小时	100小时	100小时
本机I/O 数字量 模拟量	6入/4出 -	8入/6出 -	14入/10出 -	14入/10出 2入/1出	24入/16出 -
扩展模块数量	0个模块	2个模块 ¹	7个模块 ¹	7个模块 ¹	7个模块 ¹
高速计数器 单相 双相	4路30KHz 2路20KHz	4路30KHz 2路20KHz	6路30KHz 4路20KHz	4路 30 kHz 2路200 kHz 3路20 kHz 1路100 kHz	6路30KHz 4路20KHz
脉冲输出 (DC)	2路20KHz	2路20KHz	2路20KHz	2路100 kHz	2路20KHz
模拟电位器	1	1	2	2	2
实时时钟	配时钟卡	配时钟卡	内置	内置	内置
通讯口	1 RS-485	1 RS-485	1 RS-485	2 RS-485	2 RS-485
浮点数运算	有				
I/O映象区	256 (128入/128出)				
布尔指令执行速度	0.22μs /指令				

¹ 您必须对电源消耗作出预算，从而确定S7-200 CPU能为您的配置提供多少功率（或电流）。如果超过CPU电源预算值，那么，您可能无法将全部模块都连接上去。关于CPU和扩展模块要求，可参看附录A，关于电源消耗预算，可参见附录B。

S7-200扩展模块

为了更好地满足应用要求，S7-200系列为您提供多种类型的扩展模块。您可以利用这些扩展模块完善CPU的功能。表1-3列出了现有的扩展模块，关于特定模块的详细信息，可参看附录A。

表1-3 S7-200扩展模块

扩展模块数量	型号												
数字量模块	<table border="0"> <tr> <td>输入</td> <td>8 x DC输入</td> <td>8 x AC输入</td> <td>16 x DC输入</td> </tr> <tr> <td>输出</td> <td>4 x DC输出 8 x DC输出</td> <td>4 x 继电器输出 8 x AC输出</td> <td>8 x 继电器输出</td> </tr> <tr> <td>混合</td> <td>4 x DC输入/4 x DC输出 4 x DC输入/4 x 继电器输出</td> <td>8 x DC输入/8 x DC输出 8 x DC输入/8 x 继电器输出</td> <td>16 x DC输入/16 x DC输出 16 x DC输入/16 x 继电器输出</td> </tr> </table>	输入	8 x DC输入	8 x AC输入	16 x DC输入	输出	4 x DC输出 8 x DC输出	4 x 继电器输出 8 x AC输出	8 x 继电器输出	混合	4 x DC输入/4 x DC输出 4 x DC输入/4 x 继电器输出	8 x DC输入/8 x DC输出 8 x DC输入/8 x 继电器输出	16 x DC输入/16 x DC输出 16 x DC输入/16 x 继电器输出
输入	8 x DC输入	8 x AC输入	16 x DC输入										
输出	4 x DC输出 8 x DC输出	4 x 继电器输出 8 x AC输出	8 x 继电器输出										
混合	4 x DC输入/4 x DC输出 4 x DC输入/4 x 继电器输出	8 x DC输入/8 x DC输出 8 x DC输入/8 x 继电器输出	16 x DC输入/16 x DC输出 16 x DC输入/16 x 继电器输出										
模拟量模块	<table border="0"> <tr> <td>输入</td> <td>4输入</td> <td>4热电偶输入</td> <td>2热电阻输入</td> </tr> <tr> <td>输出</td> <td>2输出</td> <td></td> <td></td> </tr> <tr> <td>混合</td> <td>4输入/1输出</td> <td></td> <td></td> </tr> </table>	输入	4输入	4热电偶输入	2热电阻输入	输出	2输出			混合	4输入/1输出		
输入	4输入	4热电偶输入	2热电阻输入										
输出	2输出												
混合	4输入/1输出												
智能模块	<table border="0"> <tr> <td>定位</td> <td>调制解调器</td> <td>PROFIBUS-DP</td> </tr> <tr> <td>以太网</td> <td>互联网</td> <td></td> </tr> </table>	定位	调制解调器	PROFIBUS-DP	以太网	互联网							
定位	调制解调器	PROFIBUS-DP											
以太网	互联网												
其它模块	ASI												

STEP 7-Micro/WIN编程软件

STEP 7-Micro/WIN编程软件为用户开发、编辑和监控自己的应用程序提供了良好的编程环境。为了能够快速高效地开发您的应用程序，STEP 7-Micro/WIN软件为您提供了三种程序编辑器。为了便于您找到所需的信息，STEP 7-Micro/WIN提供了一个详尽的在线帮助以及一个文档光盘，该光盘含有本手册的电子版、应用示例和其它有用的信息。

计算机配置要求

STEP 7-Micro/WIN既可以在PC机上运行，也可以在Siemens公司的编程器上运行。PC机或编程器的最小配置如下：

- 操作系统
Windows 2000, Windows XP
(专业版或家庭版)
- 至少100M硬盘空间
- 鼠标(推荐)

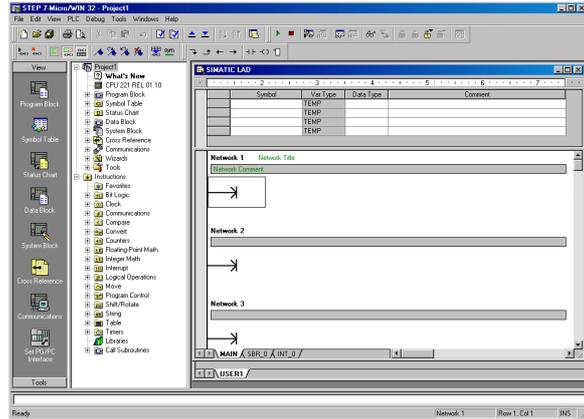


图1-2 STEP 7-Micro/WIN

安装STEP 7-Micro/WIN

将STEP 7-Micro/WIN的安装光盘插入您的CD-ROM，安装向导程序将自动启动并引导您完成整个安装过程。关于安装STEP 7-Micro/WIN的更多信息，可以参考Readme文件。



提示

若要将STEP 7-Micro/WIN装在Windows 2000或Windows XP（专业版或家庭版）上，您必须以管理员权限登录操作系统。

通讯方式选择

您可以有两种方式连接S7-200和您的编程设备：通过PPI多主站电缆直接连接，或者通过带有MPI电缆的通讯处理器（CP）卡连接。

要将计算机连接至S7-200，使用PPI多主站编程电缆是最常用和最经济的方式。它将S7-200的编程口与计算机的RS-232相连。PPI多主站编程电缆也能用来将其它通讯设备连接至S7-200。

显示面板

文本显示器 (TD 200和TD 200C)

TD 200和TD 200C是20字符双行显示器，可以连接在S7-200上。通过TD 200向导，您可以轻松地在S7-200上实现文本消息和其它应用程序数据的显示。

TD 200和TD 200C可以为您提供价格低廉的人机界面，通过它们您将能够查看、监控和改变应用程序的过程变量。

关于文本显示器的更多信息，请参考STEP 7-Micro/WIN文档光盘中的SIMATIC文本显示 (TD) 用户手册。

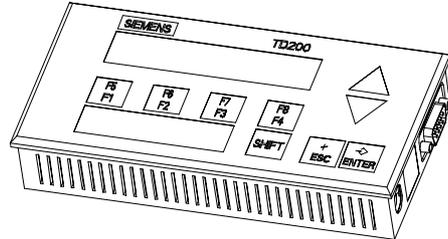


图1-3 文本显示器 (TD 200和TD 200C)



TD 200

在STEP 7-MicroWIN的TD 200组态向导的辅助下，您将可以轻松快捷地组态TD 200消息。要启动TD 200向导，可点选**Tools > TD 200 Wizard**菜单命令。

TP070和TP170触摸屏

TP070和TP170触摸屏可以连接至S7-200。通过该触摸屏，您可以自定义操作接口。

通过用户友好触摸屏，这些设备可以显示自定义图形、滚动条、应用程序变量、用户自定义按钮等等。

关于TP070和TP170触摸屏的可选编程软件，请参考附录E。

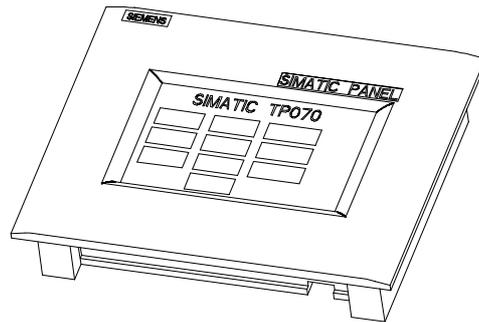


图1-4 触摸屏

2

使用入门

STEP 7-Micro/WIN软件使您能够很容易地对S7-200进行编程。通过一个简单例子程序的几个简短步骤，您将学会如何在S7-200中连接、编程和运行程序。

为了完成这个例子程序，您需要PPI多主站电缆、S7-200 CPU和能运行STEP 7-Micro/WIN软件的编程设备。

本章内容:

连接S7-200 CPU	8
创建一个例子程序	10
下载例子程序	14
将S7-200转入运行模式	14

连接S7-200 CPU

连接S7-200十分容易。在本例中，您只需要给S7-200 CPU供电，然后在编程设备与S7-200 CPU之间连上通讯电缆即可。

给S7-200 CPU供电

第一个步骤就是要给S7-200的CPU供电。图2-1给出了直流供电和交流供电两种CPU模块的接线方式。

在安装和拆除任何电气设备之前，必须确认该设备的电源已断开。在安装或拆除S7-200之前，必须遵守相应的安全防护规范，并务必将其电源断开。



警告

在带电情况下对S7-200及相关设备进行安装或接线有可能造成电击或者操作设备误动作。在安装或拆卸过程中，如果没有切断S7-200及相关设备的供电，有可能导致死亡或者严重的人身伤害和设备损坏。

必须遵循适当的安全防护规范，并确认S7-200的电源已断开。

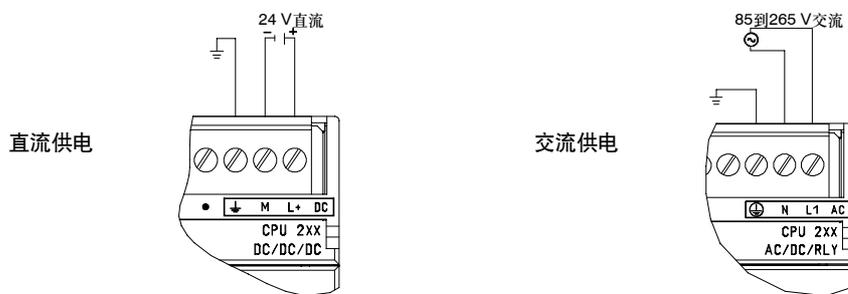


图2-1 给S7-200 CPU供电

连接RS-232/PPI多主站电缆

图2-2所示为连接S7-200与编程设备的RS-232/PPI多主站电缆。连接电缆:

1. 连接RS-232/PPI多主站电缆的RS-232端（标识为“PC”）到编程设备的通讯口上。（本例中为COM1）。
2. 连接RS-232/PPI多主站电缆的RS-485端（标识为“PPI”）到S7-200的端口0或端口1。
3. 如图2-2所示，设置RS-232/PPI多主站电缆的DIP开关。

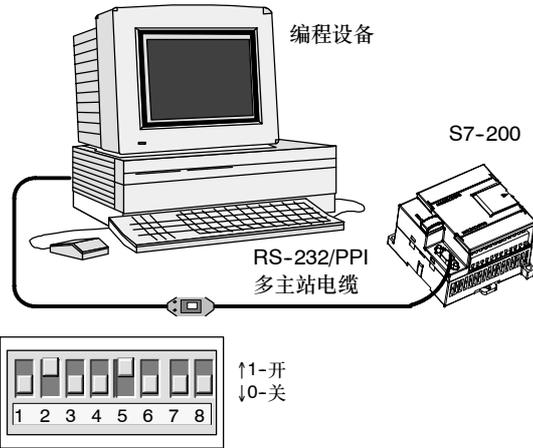


图2-2 连接RS-232/PPI多主站电缆



提示

本手册中所有的示例都使用的是RS-232/PPI多主站电缆。用RS-232/PPI多主站电缆来替代以前的PC/PPI电缆。USB/PPI多主站电缆也有供应。订货号参见附录C。

打开STEP 7-Micro/WIN

点击STEP 7-Micro/WIN的图标，打开一个新的项目，图2-3所示为一个新项目。

注意左侧的操作栏。您可以用操作栏中的图标，打开STEP 7-Micro/WIN项目中的组件。

点击操作栏中的通讯图标进入通讯对话框。您可以用这个对话框为STEP 7-Micro/WIN设置通讯参数。

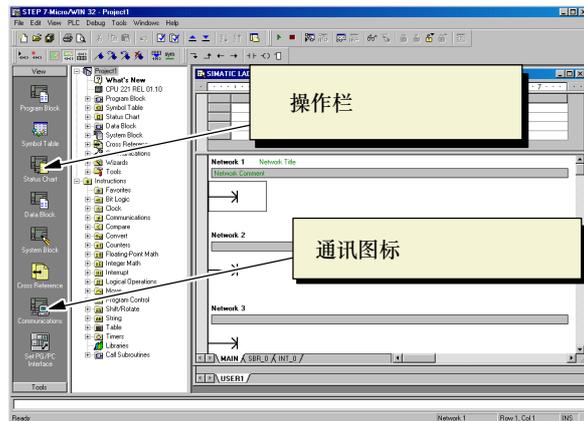


图2-3 新建STEP 7-Micro/WIN项目

为STEP 7-Micro/WIN设置通讯参数

在示例项目中使用的是STEP 7-Micro/WIN和RS-232/PPI多主站电缆的缺省设置。检查下列设置:

1. PC/PPI电缆的通讯地址设为 0。
2. 接口使用COM1。
3. 传输波特率用9.6Kbps。

如果您需要改变通讯设置，请参考第7章。

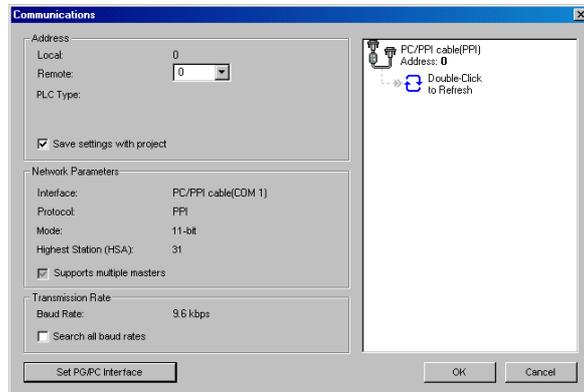


图2-4 设置通讯参数

与S7-200建立通讯

用通讯对话框与S7-200建立通讯:

1. 在通讯对话框中双击刷新图标。
STEP 7-Micro/WIN搜寻并显示所连接的S7-200站的CPU图标。
2. 选择S7-200站并点击OK。

如果STEP 7-Micro/WIN未能找到您的S7-200 CPU，请核对您的通讯参数设置。

建立与S7-200的通讯之后，您就可以创建并下载示例程序。

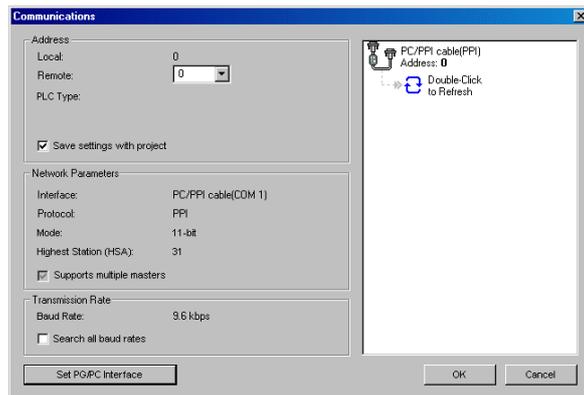


图2-5 与S7-200建立通讯

创建一个例子程序

创建这个例子程序将使您体会到使用STEP 7-Micro/WIN编程有多简单。这个例子程序在三个程序段中用6条指令，完成了一个定时器自启动、自复位的简单功能。

在本例中，您用梯形图编辑器来录入程序。下面给出了完整的梯形图和语句表程序。语句表中的注释，解释了程序的逻辑关系。时序图显示了程序的运行状态。

示例: STEP 7-Micro/WIN使用入门例子程序

Network 1

Network 2

Network 3

```

Network 1 //10 ms 定时器T33在
           //( 100 x 10 ms=1 s )后输出。
           //M0.0脉冲过窄不能由状态视图监视。
LDN      M0.0
TON      T33, +100
Network 2 //比较结果为真
           //可由状态视图监视。
           //在 ( 40 x 10 ms=0.4 s )之后,
           //Q0.0输出
           // 40%低电平、60%高电平的波形。
LDW>=    T33, +40
=        Q0.0
Network 3 //T33 (位) 脉冲太快
           //以致不能用状态视图监视。在 ( 100 x 10 ms=1 s )
           //之后,
           //通过M0.0复位定时器。
LD       T33
=        M0.0
    
```

时序图

打开程序编辑器

点击程序块图标，打开程序编辑器，见图2-6。

注意指令树和程序编辑器。您可以用拖拽的方式将梯形图指令插入到程序编辑器中。

在工具栏图标中有一些命令的快捷方式。

在输入和保存程序之后，您可以下载程序到S7-200中。

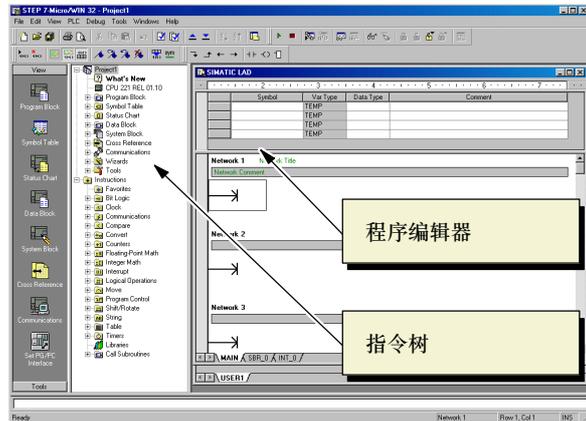


图2-6 STEP 7-Micro/WIN窗口

输入程序段1：启动定时器

当M0.0的状态为0时，常闭触点接通启动定时器。输入M0.0的触点：

1. 双击位逻辑图标或者单击其左侧的加号可以显示出全部位逻辑指令。
2. 选择常闭触点。
3. 按住鼠标左键将触点拖到第一个程序段中。
4. 单击触点上的“???”，并输入地址：M0.0。
5. 按回车键确认。

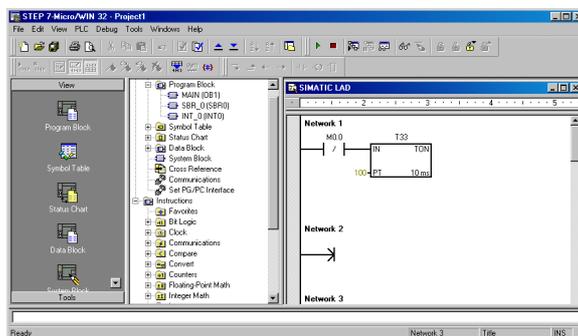


图2-7 Network 1

定时器指令T33的输入步骤如下：

1. 双击定时器图标，显示定时器指令。
2. 选择延时接通定时器TON。
3. 按住鼠标左键将定时器拖到第一个程序段中。
4. 单击定时器上方的“???”，输入定时器号：T33。
5. 按回车键确认后，光标会自动移动到预置时间值（PT）参数。
6. 输入预置时间值：100
7. 按回车键确认。

输入程序段2：使输出点闭合

当定时器T33的定时值大于等于40时（大于等于0.4秒），S7-200的输出点Q0.0会闭合。输入比较指令的步骤如下：

1. 双击比较指令图标，显示所有的比较指令。选择“>=I”指令。
2. 按住鼠标左键将比较指令拖到第二个程序段中。
3. 单击触点上方的“???”，输入定时器号：T33。
4. 按回车键确认后，光标会自动移动到比较指令下方的比较值参数。
5. 在该处输入比较值40
6. 按回车键确认。

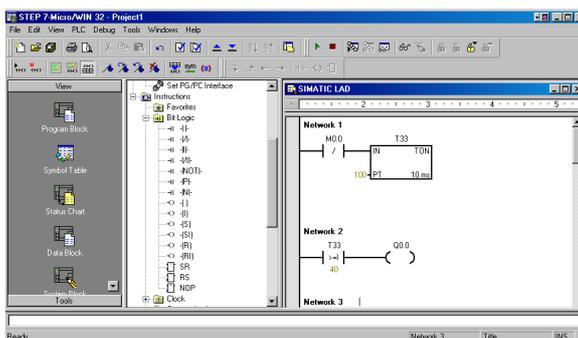


图2-8 Network 2

输出指令的输入步骤如下：

1. 双击位逻辑图标，显示位逻辑指令并选择输出线圈。
2. 按住鼠标左键将输出线圈拖到第二个程序段中。
3. 单击线圈上方的“???”，输入地址：Q0.0。
4. 按回车键确认。

输入程序段3：定时器复位

当计时值到达预置时间值（100）时，定时器触点会闭合。T33闭合会使M0.0置位。由于定时器是靠M0.0的常闭触点启动的，M0.0的状态由0变1会使定时器复位。

输入触点T33的步骤如下:

1. 在位逻辑指令中选择常开触点。
2. 按住鼠标左键将触点拖到第三个程序段中。
3. 单击触点上方的“???”，输入地址：T33。
4. 按回车键确认。

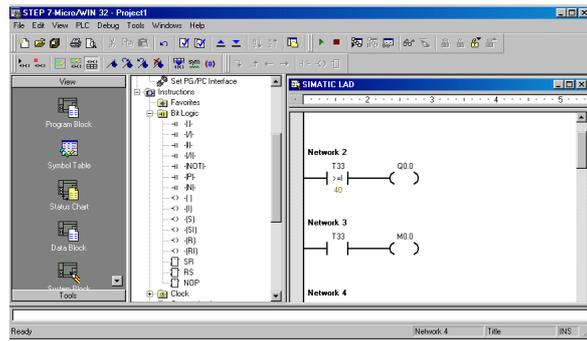


图2-9 Network 3

输入线圈M0.0的步骤如下:

1. 在位逻辑指令中选择输出线圈。
2. 按住鼠标左键将输出线圈拖到第三个程序段中。
3. 双击线圈上方的“???”，输入地址：M0.0。
4. 按回车键确认。

存储例子程序

在输入完以上三个程序段后，您已经完成了整个例子程序。当您存储程序时，您也创建了一个包括S7-200 CPU类型及其它参数在内的一个项目。保存项目:

1. 在菜单条中选择菜单命令 **File > Save As**。
2. 在Save As对话框中输入项目名。
3. 点击Save存储项目。

项目存储之后，您可以下载程序到S7-200。

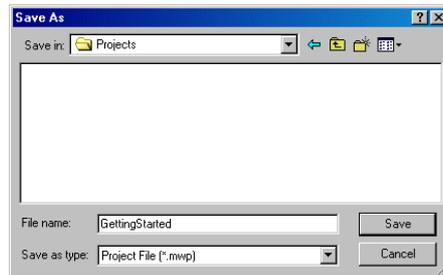


图2-10 保存例子程序

下载例子程序



提示

每一个STEP 7-Micro/WIN项目都会有一个CPU类型（CPU221、CPU222、CPU224、CPU224XP或CPU226）。如果您在项目中选择的CPU类型，与您实际连接的CPU类型不匹配，STEP 7-Micro/WIN会提示您并要您作出选择。如果您在本例中遇到这种情况，可以选择“继续下载”。

1. 您可以点击工具条中的下载图标或者在命令菜单中选择**File > Download**来下载程序。参见图2-11。
2. 点击OK下载程序到S7-200。

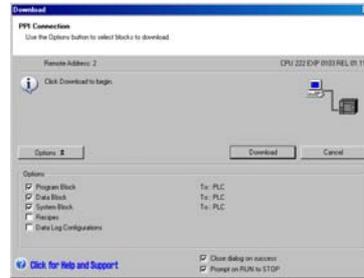


图2-11 下载程序

如果您的S7-200处于运行模式，将有一个对话提示您CPU将进入停止模式。单击Yes将S7-200置于STOP模式。

将S7-200转入运行模式

如果想通过STEP 7-Micro/WIN软件将S7-200转入运行模式，S7-200的模式开关必须设置为TERM或者RUN。当S7-200处于RUN模式时，执行程序：

1. 单击工具条中的运行图标或者在命令菜单中选择**PLC > RUN**。
2. 点击Yes切换模式。

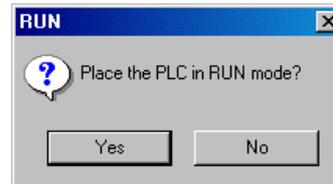


图2-12 将S7-200转入运行模式

当S7-200转入运行模式后，CPU将执行程序使Q0.0的LED指示灯时亮时灭。

恭喜您！您已经完成了您的第一个S7-200程序。

您可以通过选择**Debug > Program Status**来监控程序。STEP 7-Micro/WIN显示执行结果。要想终止程序，可以单击STOP图标或选择菜单命令**PLC > STOP**将S7-200置于STOP模式。

3

S7-200的安装

S7-200的设计使其便于安装。可以利用安装孔把模块固定在控制柜的背板上，或者利用设备上的DIN夹子，把模块固定在一个标准（DIN）的导轨上。体积小巧的S7-200可以使您更为有效地安排空间。

本章提供S7-200系统的安装和接线的指导。

本章内容:

S7-200设备的安装指南	16
S7-200模块的安装和拆卸	17
接地和接线指南	20

S7-200设备的安装指南

S7-200既可以安装在控制柜背板上，也可以安装在标准导轨上；既可以水平安装，也可以垂直安装。

将S7-200与加热装置、高电压和电子噪声隔离开

按照惯例，在安装元器件时，总是把产生高电压和高电子噪声的设备与诸如S7-200这样的低压、逻辑型的设备分隔开。

在控制柜背板上安排S7-200时，应考虑把产热器件和电子器件安排在控制柜中温度较低的区域。电子器件在高温环境下工作会缩短其无故障时间。

要考虑控制柜背板的布线，避免将低压信号线和通讯电缆与交流供电线和高能量、开关频率很高的直流信号线布置在一个线槽中。

为接线和散热留出适当的空间

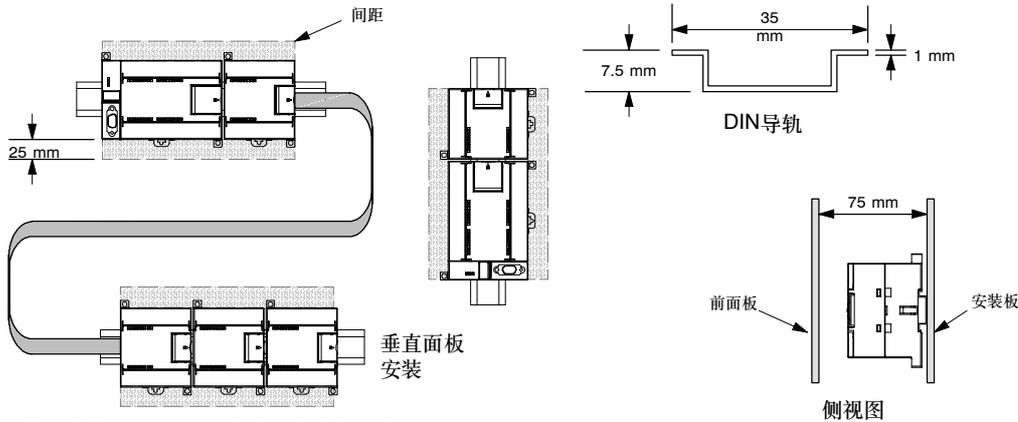
S7-200设备的设计采用自然对流散热方式，在器件的上方和下方都必须留有至少25mm的空间，以便于正常的散热。前面板与背板的板间距离也应保持至少75mm。



提示

在垂直安装时，其允许的最高环境温度要比水平安装时低10°C，而且CPU应安装在所有扩展模块的下方。

在安排S7-200设备时，应留出接线和连接通讯电缆的足够空间。当配置S7-200系统时，可以灵活地使用I/O扩展电缆。



DIN导轨水平安装（一个S7-200系统只允许使用一个可选的扩充电缆）

图3-1 安装方式、方向和间距

电源预算

所有的S7-200 CPU都有一个内部电源，为CPU自身、扩展模块和其它用电设备提供24V直流电源。

S7-200为系统中的所有扩展模块提供5V直流逻辑电源。必须格外注意您的系统配置，要确保CPU所提供的5V电源，能够满足您所选择的所有扩展模块的需要。如果您的配置要求超出了CPU的供电能力，您只有去掉一些模块或者选择一个供电能力更强的CPU。在附录A中，您会得到有关S7-200 CPU 5V直流逻辑电源的供电能力以及扩展模块对5V直流电源需求的信息。附录B给出了CPU所能为系统提供功率（电流）大小的计算方法。

S7-200的所有CPU也提供24V直流传感器供电，此24VDC可以为输入点、扩展模块上的继电器线圈或者其它设备供电。如果设备用电量超过了传感器供电预算，必须为系统另配一个外部24VDC供电电源。对于特定的S7-200 CPU，可以在附录A中查到其24VDC传感器供电电源预算。

如果您使用了外部24VDC供电电源，要确保该电源没有与S7-200 CPU上的传感器电源并联使用。为了加强电子噪音保护，建议将不同电源的公共端（M）连在一起。



警告

将外部24VDC电源与S7-200的24VDC传感器供电电源并联，会造成两路供电之间的冲突，每一路电源都试图建立自己的电位输出。

这种冲突的结果会缩短电源寿命，或者一路或二路电源立即损坏，这样会使PLC系统产生一系列不确定的操作。这种不确定的操作会造成死亡或者严重的人身伤害和设备损坏。

S7-200的24VDC传感器供电和任何外部供电应该分别给不同的点提供电源。

S7-200模块的安装和拆卸

S7-200可以很容易地安装在一个标准DIN导轨或控制柜背板上。

先决条件

在安装和拆卸电子器件之前，要确保该设备的供电已被切断。同样，也要确保与该设备相关联的设备的供电已被切断。



警告

试图在带电情况下安装或拆卸S7-200及其相关设备有可能导致电击或者设备误动作。

在安装和拆卸S7-200及其相关设备时，如果未能切断所有电源，有可能造成死亡或严重的人身伤害和设备损坏。

在安装和拆卸S7-200及其相关设备时，必须预先采取适当的安全措施并且确认S7-200的供电被切断。

在更换或安装S7-200器件时，要确保使用了正确的模块或等同的模块。



警告

如果您安装了不正确的模块，S7-200的程序可能会产生错误的功能。

如果未能使用相同的模块按照相同的方向和顺序替换S7-200的器件，有可能导致死亡或者严重的人身伤害和设备损坏。

在更换S7-200的器件时，除了要使用相同的模块外，还要确保安装的方向和位置是正确的。

安装尺寸

S7-200的CPU和扩展模块都有安装孔，可以很方便地安装在背板上。表3-1所示为安装尺寸。

表3-1 安装尺寸

S7-200模块		宽度A	宽度B
CPU221和CPU222		90 mm	82 mm
CPU 224		120.5 mm	112.5 mm
CPU224XP		140 mm	132 mm
CPU 226		196 mm	188 mm
扩展模块:	4点、8点直流和继电器I/O (8I、4Q、8Q、4I/4Q) 及模拟量输出 (2 AQ)	46 mm	38 mm
扩展模块:	16点数字量I/O (16I、8I/8Q)、模拟量I/O (4AI、4AI/1AQ)、RTD、热电偶、PROFIBUS、以太网、因特网、AS-涌流点交流 (8I和8Q)、定位模块和Modem模块	71.2 mm	63.2 mm
扩展模块:	32点数字量I/O (16I/16Q)	137.3 mm	129.3 mm

CPU和扩展模块的安装

按照以下步骤，可以非常容易地安装S7-200！

面板安装

1. 按照表3-1所示的尺寸进行定位、钻安装孔（用M4或美国标准8号螺钉）。
2. 用合适的螺钉将模块固定在背板上。
3. 如果您使用了扩展模块，将扩展模块的扁平电缆连到前盖下面的扩展口。

DIN导轨安装

1. 保持导轨到安装面板的距离为75 mm。
2. 打开模块底部的DIN夹子，将模块背部卡在DIN导轨上。
3. 如果您使用了扩展模块，将扩展模块的扁平电缆连到前盖下面的扩展口。
4. 旋转模块贴近DIN导轨，合上DIN夹子。仔细检查模块上DIN夹子与DIN导轨是否紧密固定好。为避免模块损坏，不要直接按压模块正面，而要按压安装孔的部分。



提示

当S7-200的使用环境振动比较大或者采用垂直安装方式时，应该使用DIN导轨挡块。

如果系统处于高震动环境中，使用背板安装方式可以得到较高的震动保护等级。

拆卸CPU或者扩展模块

按照以下步骤拆卸S7-200 CPU或扩展模块:

1. 拆除S7-200的电源。
2. 拆除模块上的所有连线和电缆。大多数的CPU和扩展模块有可拆卸的端子排，使这项工作变得简单。
3. 如果有其它扩展模块连接在您所拆卸的模块上，请打开前盖，拔掉相邻模块的扩展扁平电缆。
4. 拆掉安装螺钉或者打开DIN夹子。
5. 拆下模块。

拆卸和安装端子排

为了安装和替换模块方便，大多数的S7-200模块都有可拆卸的端子排。附录A中给出了哪些S7-200模块有可拆卸的端子排。您可以为不带端子排的模块订购，订货号参见附录C。

端子排的拆卸

1. 打开端子排安装位置的上盖，以便可以接近端子排。
2. 把螺丝刀插入端子块中央的槽口中。
3. 所示用力下压并撬出端子排。参见图3-2。

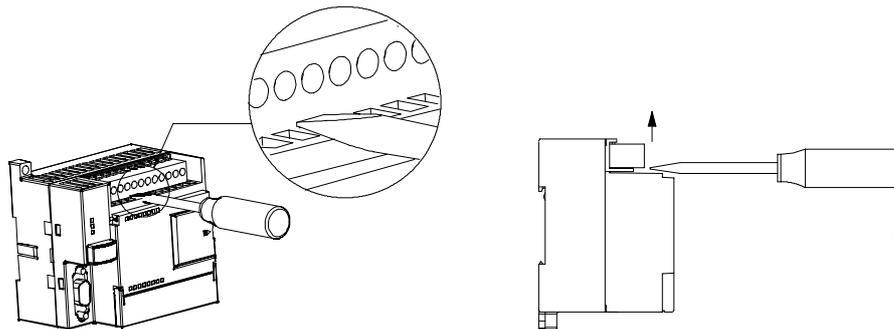


图3-2 拆卸端子排

端子排的安装

1. 打开端子排的上盖。
2. 确保模块上的插针与端子排边缘的小孔对正。
3. 将端子排向下压入模块。确保端子块对准了位置并锁住。

接地和接线指南

对所有电器设备进行合理的接地和接线是非常重要的，它能够确保您的系统具备最优的操作特性，同时能够为您的应用和S7-200提供更好的电子噪声保护。

先决条件

在接地和接线之前，必须先确保设备的电源已被切断。同样，也要确保与该设备相关联的设备的供电已被切断。

在对S7-200及其相关设备接线时，必须确保遵从所有有效的电气编码规则。安装和操作所有设备要符合所有有效的国家或地区标准。同地区的权威保持联系，以确定哪些标准符合您的特殊需要。



警告

试图在带电情况下安装或拆卸S7-200及其相关设备有可能导致电击或者设备误动作。在安装和拆卸S7-200及其相关设备时，如果未能切断所有电源，有可能造成死亡或严重的人身伤害和设备损坏。

在安装和拆卸S7-200及其相关设备时，必须采取适当的安全措施并且确认S7-200的供电被切断。

在设计S7-200系统的接地和接线时必须考虑安全因素。象S7-200这样的控制设备有可能造成它所监控的设备的误动作。因此，您应该执行所有的安全规定以避免人员伤亡和设备损坏。



警告

控制设备有可能造成它所控制的设备的误操作。这种误操作有可能导致死亡或者严重的人身伤害和设备损坏。

使用独立于S7-200的急停功能、机电互锁或者其它冗余的安全措施。

隔离指南

S7-200的交流供电和I/O点与交流电路之间的隔离为1500VAC。这些隔离被检验和证明可以作为交流线和低压电路之间的安全隔离。

所有与S7-200相连的低压电路，例如24V供电，必须与交流线和其它高压之间有安全隔离，符合各种安全标准。这些电源具有国际电气安全标准里所定义的双层绝缘保护，其输出符合SELV、PELV、等级2标准或者根据各种标准的限定电源输出。



警告

使用与交流电路不隔离或者单隔离的电源给低压电路供电，会在安全电路，例如通讯电路或者低压传感器电路中产生不安全电压。

这种高电压会导致死亡或者严重的人身伤害和设备损坏。

只有使用高电压到低电压的变换器，才能保证电路安全。

S7-200接地指南

对于您的应用，最佳的接地方案应该确保S7-200及其相关设备的所有接地点在一点接地。这个单独的接地点应该直接连接到您的系统地上。

为了提高抗电子噪声保护特性，建议将所有直流电源的公共点连接到同一个单一接地点上。同样建议将24VDC传感器供电的公共点（M）接地。

所有的接地线应该尽量短并且用较粗的线径（2mm²或者14AWG）。

当选择接地点时，应当考虑安全接地要求和对隔离器件的适当保护。

S7-200接线指南

在设计S7-200的接线时，应该提供一个单独的开关，能够同时切断S7-200 CPU、输入电路和输出电路的所有供电。提供熔断器或断路器等过流保护装置来限制供电线路中的电流。您也可以为每一输出电路都提供熔断器或其它限流设备作为额外的保护。

在有可能遭受雷击浪涌的线路上安装浪涌抑制器件。

避免将低压信号线和通讯电缆与交流线和高能量快速开关的直流线设计在同一个走线槽中。使用双绞线并且用中性线或者公共线与能量线或者信号线相配对。

导线尽量短并且保证线粗能够满足电流要求。端子排适合的线粗为2mm²到0.3mm²（14AWG到22AWG）。使用屏蔽电缆可以得到最佳的抗电子噪声特性。通常将屏蔽层接地可以得到最佳效果。

当输入电路由一个外部电源供电时，要在电路中添加过流保护器件。如果使用S7-200 CPU上的24VDC传感器供电电源，则无需额外添加过流保护器件，因为此电源已经有限流保护。

大多数的S7-200模块有可拆卸的端子排。（附录A中标明了哪些模块有端子排）。为了防止连接松动，要确保端子排插接牢固，同时也要确保导线牢固地连接在端子排上。为了避免损坏端子排，螺钉不要拧得太紧。螺钉连接的最大扭矩为0.56N·m（5 inch-pounds）。

为了避免意想不到的电流流入系统，S7-200在合适的部分提供电气隔离。当您设计系统走线时，应考虑这些隔离。附录A中给出了电路中包含哪些隔离及它们的隔离级别。级别低于1500VAC的隔离不能作为安全隔离。



提示

在通讯网络中，如果不使用中继器，通讯电缆的最大长度为50m。S7-200的通讯口是不隔离的。详细内容参见第7章。

感性负载设计指南

在使用感性负载时，要加入抑制电路来限制输出关断时电压的升高。抑制电路可以保护输出点不至于因为高感抗开关电流而过早的损坏。另外，抑制电路还可以限制感性负载开关时产生的电子噪声。



提示

抑制电路的有效性取决于应用，您应该调整其参数以适应您的特殊应用。要确保所有器件参数与实际应用相符合。

晶体管输出和控制直流负载的继电器输出

晶体管输出有内部保护，可以适应多种应用。由于继电器型输出既可以连接直流负载，又可以连接交流负载，因而没有内部保护。

图3-3给出了直流负载抑制电路的一个实例。在大多数的应用中，用附加的二极管A即可，但如果您的应用中要求更快的关断速度，则推荐您加上齐纳二极管B。确保齐纳二极管能够满足输出电路的电流要求。

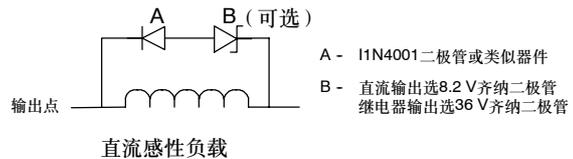


图3-3 直流负载的抑制电路

交流输出和控制交流负载的继电器输出

交流输出有内部保护，可以适应多种应用。由于继电器型输出既可以连接直流负载，又可以连接交流负载，因而没有内部保护。

图3-4给出了交流负载抑制电路的一个实例。当您采用继电器或交流输出来切换 115 V/230 V 交流负载时，交流负载电路中请采用该图所示的电阻/电容网络。您也可以使用金属氧化物可变电阻器（MOV）来限制峰值电压。确保MOV的工作电压比正常的线电压至少高出20%。

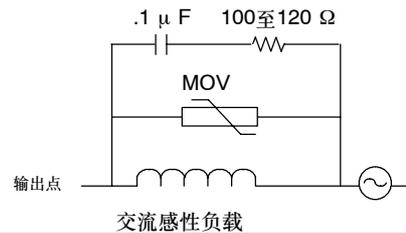


图3-4 交流负载的抑制电路

注释

当采用继电器扩展模块切换 230 V 交流感性负载时，必须在交流负载电路中采用图3-4所示的外部电阻器/电容器噪声抑制电路。

灯负载设计指南

灯负载会因高的接通浪涌电流而造成对继电器触点的损坏。对于一个钨丝灯，其浪涌电流实际上将是其稳态电流大小的10到15倍。对于使用期内高切换次数的灯负载，建议使用可替换的插入式继电器或加入浪涌限制器。

4

PLC的基本概念

S7-200的基本功能是监视现场的输入，根据您的控制逻辑去控制现场输出设备的接通和关断。本章为您解释有关程序执行、变量类型以及存储区掉电保持等方面的一些概念。

本章内容:

理解S7-200如何执行您的控制逻辑	24
S7-200数据的存取	26
理解S7-200如何保存和存储数据	36
为S7-200 CPU选择操作模式	41
使用S7-200资源管理器	41
S7-200的特性	42

理解S7-200如何执行您的控制逻辑

S7-200周而复始地执行程序中的控制逻辑和读写数据。

S7-200将您的程序和物理输入输出点联系起来。

S7-200的基本操作非常简单：

- CPU读输入状态
- CPU中存储的程序利用输入执行控制逻辑。当程序运行时，CPU刷新有关数据。
- CPU将数据写到输出。

图4-1给出了一个简图，说明一个电磁继电器如何连到S7-200。在本例中，电机启动开关的状态和其它输入点的状态结合在一起。它们计算的结果，最终决定了控制执行机构启动电机的输出点状态。

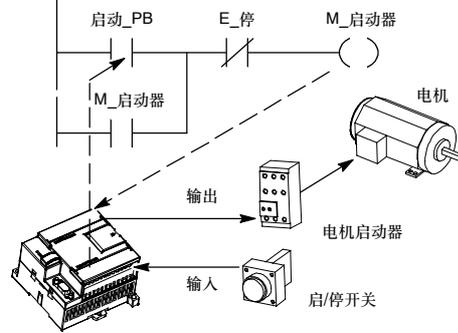


图4-1 输入和输出的控制

S7-200在扫描循环中完成它的任务

S7-200周而复始地执行一系列任务。任务循环执行一次称为一个扫描周期。如图4-2所示，在一个扫描周期中，S7-200将执行部分或全部下列操作：

- 读输入：S7-200将物理输入点上的状态复制到输入过程映像寄存器中。
- 执行逻辑控制程序：S7-200执行程序指令并将数据存储在变量存储区中。
- 处理通讯请求：S7-200执行通讯任务。
- 执行CPU自诊断：S7-200检查固件、程序存储器和扩展模块是否工作正常。
- 写输出：在输出过程映像寄存器中存储的数据被复制到物理输出点。

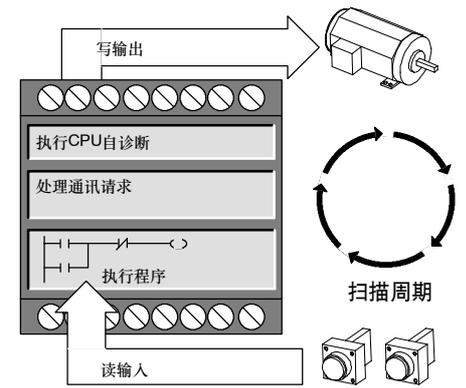


图4-2 S7-200扫描周期

用户程序的执行取决于S7-200是处于停止模式还是运行模式。当S7-200处于运行模式时，CPU执行程序；当S7-200处于停止模式时，CPU不执行程序。

读输入

数字量输入: 在每个扫描周期的开始, CPU会读取数字量输入, 并将这些值写入过程映像输入寄存器。

模拟量输入: 除非使能模拟量滤波, 否则S7-200在扫描周期中不会刷新模拟量输入值。模拟量滤波会使您得到较稳定的信号。您可以使能每个模拟量输入通道的滤波功能。

当您使能了模拟量输入滤波功能后, S7-200会在每一个扫描周期刷新模拟量、执行滤波功能并且在内部存储滤波值。当程序中访问模拟量输入时使用滤波值。

如果没有使能模拟量输入滤波, 则当程序访问模拟量输入时, S7-200都会直接从扩展模块读取模拟值。

在每次扫描期间, CPU224XP的AIW0和AIW2模拟量输入都会读取模-数转换器生成的最新值, 从而完成刷新。该转换器求取的是均值 (σ), 因此通常无需软件滤波。



提示

模拟量滤波会使您得到较稳定的信号。在模拟量输入信号随时间变化缓慢时使用模拟量输入滤波。如果信号变化很快, 不应该选用模拟量滤波。

不要对在模拟量字中传递数字信息或者报警指示的模块使用模拟量输入滤波。对于RTD、TC和ASI主站模块, 不能使用模拟量输入滤波。

执行程序

在扫描周期的执行程序阶段, CPU从头至尾执行应用程序。在程序或中断服务中, 直接I/O指令允许您对I/O点直接进行存取。

如果在程序中使用了中断, 与中断事件相关的中断服务程序作为程序的一部分被存储。中断程序并不作为正常扫描周期的一部分来执行, 而是当中断事件发生时才执行 (可能在扫描周期的任意点)。

处理通讯请求

在扫描周期的处理阶段, CPU处理从通讯端口或者智能I/O模块接收到的任何信息。

执行CPU自诊断测试

在扫描周期的处理阶段, S7-200检测CPU的操作和扩展模块的状态是否正常。

写数字输出

在每个扫描周期的结尾, CPU把存储在输出映像寄存器中的数据写到数字输出点。(模拟量输出直接刷新, 与扫描周期无关)

S7-200数据的存取

S7-200将信息存于不同的存储器单元，每个单元都有唯一的地址。您可以明确指出要存取的存储器地址。这就允许用户程序直接存取这个信息。表 4-1列出了不同长度的数据所能表示的数值范围。

表4-1 不同长度的数据表示的十进制和十六进制数范围

数制	字节 (B)	字 (W)	双字 (D)
无符号整数	0到255 0到FF	0到65,535 0到FFFF	0到4,294,967,295 0到FFFF FFFF
符号整数	-128到+127 80到7F	-32,768到+32,767 8000到7FFF	-2,147,483,648到+2,147,483,647 8000 0000到7FFF FFFF
实数IEEE 32 位浮点数	不用	不用	+1.175495E-38到+3.402823E+38 (正数) -1.175495E-38到-3.402823E+38 (负数)

若要存取存储区的某一位，则必须指定地址，包括存储器标识符、字节地址和位号。图4-3是一个位寻址的例子（也称为“字节.位寻址”）。在这个例子中，存储器区、字节地址（I代表输入，3代表字节3）和位地址（第4位）之间用点号（“.”）相隔开。

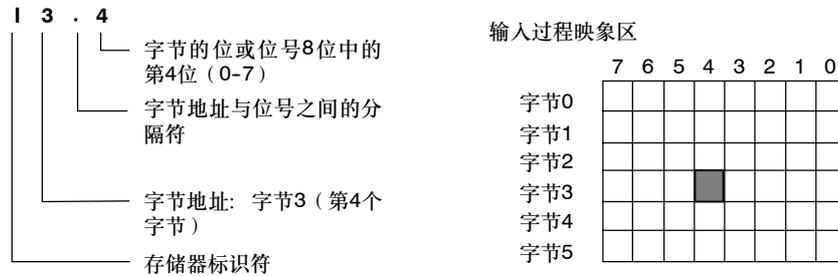


图4-3 字节.位寻址

使用这种字节寻址方式，可以按照字节、字或双字来存取许多存储区（V、I、Q、M、S、L及SM）中的数据。若要存取CPU中的一个字节、字或双字数据，则必须以类似位寻址的方式给出地址，包括存储器标识符、数据大小以及该字节、字或双字的起始字节地址，如图4-4所示。

其它CPU存储区（如T，C，HC和累加器）中存取数据使用的地址格式包括区域标识符和设备号。

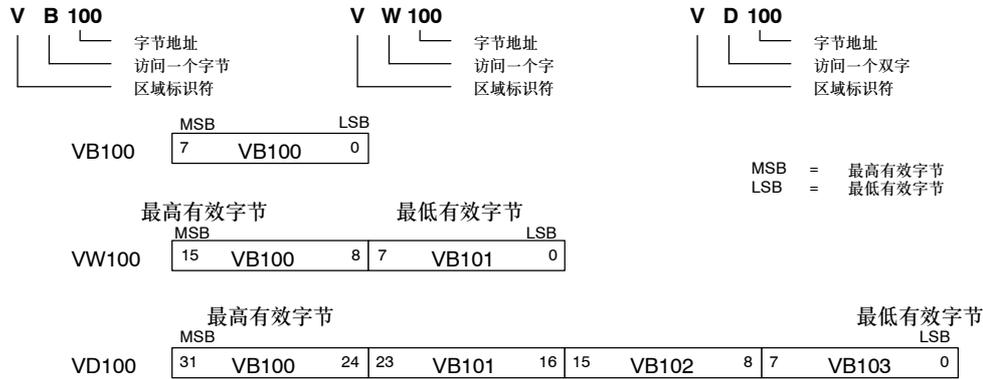


图4-4 对同一地址进行字节，字和双字存取操作的比较。

存储区数据的存取

输入过程映象寄存器: I

在每次扫描周期的开始，CPU对物理输入点进行采样，并将采样值写入输入过程映象寄存器中。可以按位、字节、字或双字来存取输入过程映象寄存器中的数据:

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

输出过程映象寄存器: Q

在每次扫描周期的结尾，CPU将输出过程映象寄存器中的数值复制到物理输出点上。可以按位、字节、字或双字来存取输出过程映象寄存器:

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

变量存储区: V

您可以用V存储器存储程序执行过程中控制逻辑操作的中间结果，也可以用它来保存与工序或任务相关的其它数据。并且可以按位、字节、字或双字来存取V存储区中的数据:

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

位存储区: M

可以用位存储区作为控制继电器来存储中间操作状态和控制信息。并且可以按位、字节、字或双字来存取位存储区:

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

定时器存储区: T

S7-200 CPU中, 定时器可用于时间累计, 其分辨率(时基增量)分为1ms、10ms和100ms三种。定时器有两种形式:

- ❑ 当前值: 16位有符号整数, 存储定时器所累计的时间。
- ❑ 定时器位: 按照当前值和预置值的比较结果置位或者复位。预置值是定时器指令的一部分。

可以用定时器地址(T+定时器号)来存取这两种形式的定时器数据。究竟使用哪种形式取决于所使用的指令: 如果使用位操作指令则是存取定时器位; 如果使用字操作指令, 则是存取定时器当前值。如图4-5中所示, 常开触点指令是存取定时器位; 而字移动指令则是存取定时器的当前值。

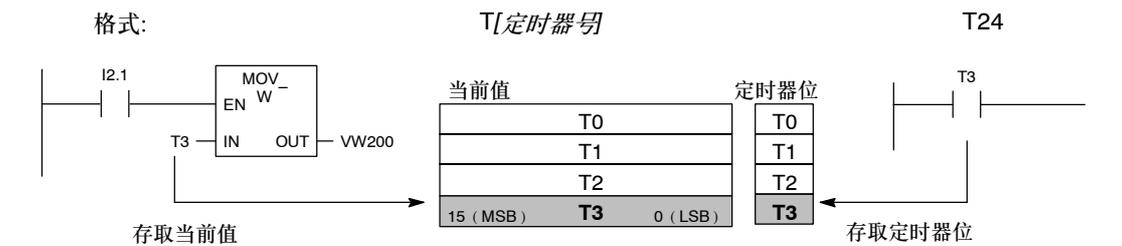


图4-5 存取定时器位或者定时器的当前值

计数器存储区: C

在S7-200 CPU中, 计数器可以用于累计其输入端脉冲电平由低到高的次数。CPU提供了三种类型的计数器: 一种只能增计数; 一种只能减计数; 另外一种既可以增计数, 又可以减计数。计数器有两种形式:

- ❑ 当前值: 16位有符号整数, 存储累计值。
- ❑ 计数器位: 按照当前值和预置值的比较结果置位或者复位。预置值是计数器指令的一部分。

可以用计数器地址(C+计数器号)来存取这两种形式的计数器数据。究竟使用哪种形式取决于所使用的指令: 如果使用位操作指令则是存取计数器位; 如果使用字操作指令, 则是存取计数器当前值。如图4-6中所示, 常开触点指令是存取计数器位; 而字移动指令则是存取计数器的当前值。

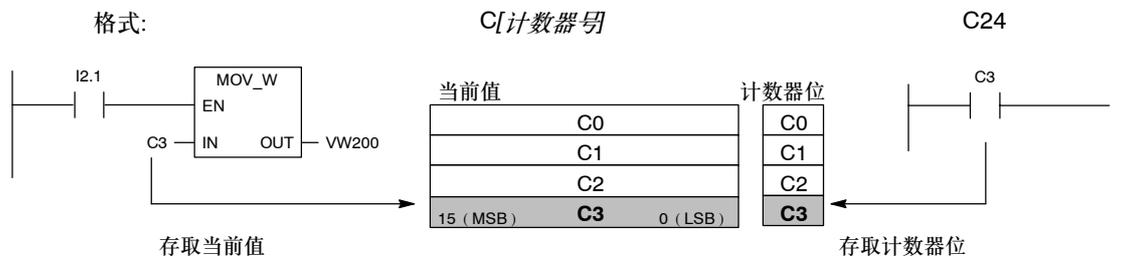


图4-6 存取计数器位或者计数器的当前值

高速计数器: HC

高速计数器对高速事件计数，它独立于CPU的扫描周期。高速计数器有一个32位的有符号整数计数值（或当前值）。若要存取高速计数器中的值，则应给出高速计数器的地址，即存储器类型（HC）加上计数器号（如HC0）。高速计数器的当前值是只读数据，仅可以作为双字（32位）来寻址。

格式: $HC[高速计数器号] \quad HC1$

累加器: AC

累加器是可以象存储器一样使用的读写设备。例如，可用它来向子程序传递参数，也可从子程序返回参数，以及用来存储计算的中间结果。S7-200提供4个32位累加器（AC0，AC1，AC2和AC3）。并且您可以按字节、字或双字的形式来存取累加器中的数值。

被访问的数据长度取决于存取累加器时所使用的指令。如图4-7所示，当以字节或者字的形式存取累加器时，使用的是数值的低8位或低16位。当以双字的形式存取累加器时，使用全部32位。

关于如何在中断服务程序中使用累加器的相关信息，参见第6章中的中断指令部分。

格式: $AC[累加器号] \quad AC0$

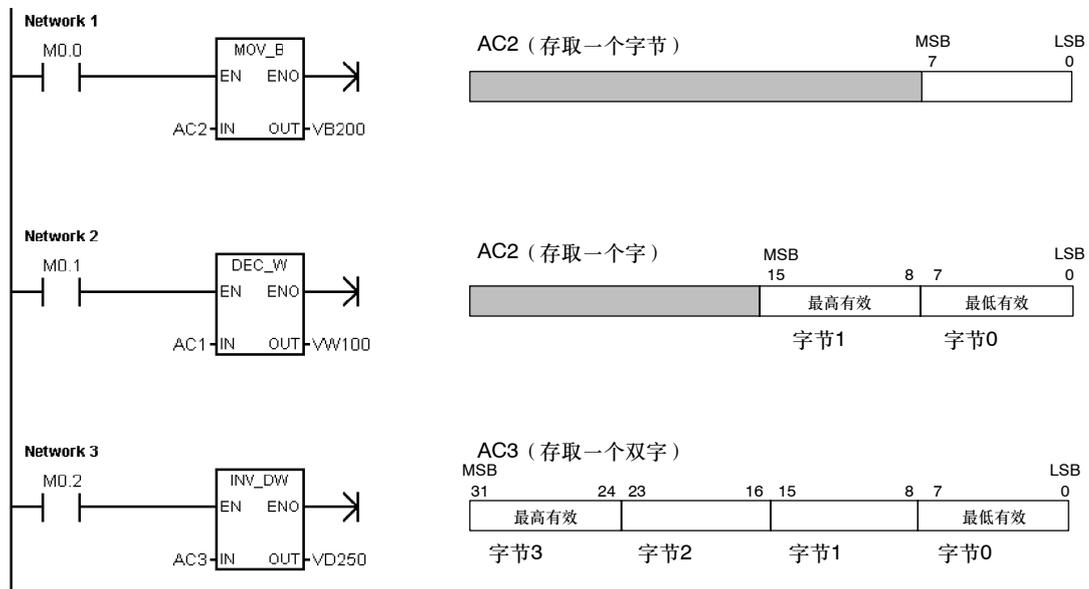


图4-7 存取累加器

特殊存储器: SM

SM位为CPU与用户程序之间传递信息提供了一种手段。可以用这些位选择和控制S7-200 CPU的一些特殊功能。例如: 首次扫描标志位、按照固定频率开关的标志位或者显示数学运算或操作指令状态的标志位。(有关SM位的详细信息参见附录B)。并且可以按位、字节、字或双字来存取SM位:

位:	SM[字节地址].[位地址]	SM0.1
字节、字或者双字:	SM[长度][起始字节地址]	SMB86

局部存储器: L

S7-200有64个字节的局部存储器, 其中60个可以用作临时存储器或者给予程序传递参数。

**提示**

如果用梯形图或功能块图编程, STEP 7-Micro/WIN保留这些局部存储器的最后四个字节。

局部存储器和变量存储器很相似, 但只有一处区别。变量存储器是全局有效的, 而局部存储器只在局部有效。全局是指同一个存储器可以被任何程序存取(包括主程序、子程序和中断服务程序)。局部是指存储器区和特定的程序相关联。S7-200给主程序分配64个局部存储器; 给每一级子程序嵌套分配64个字节局部存储器; 同样给中断服务程序分配64个字节局部存储器。

子程序或者中断服务程序不能访问分配给主程序的局部存储器。子程序不能访问分配给主程序、中断服务程序或者其它子程序的局部存储器。同样的, 中断服务程序也不能访问分配给主程序或子程序的局部存储器。

S7-200 PLC根据需要分配局部存储器。也就是说, 当主程序执行时, 分配给子程序或中断服务程序的局部存储器是不存在的。当发生中断或者调用一个子程序时, 需要分配局部存储器。新的局部存储器在分配时可以重新使用分配给另一个子程序或中断服务程序的局部存储器。

局部存储器在分配时PLC不进行初始化, 初值可能是任意的。当在子程序调用中传递参数时, 在被调用子程序的局部存储器中, 由CPU替换其被传递的参数的值。局部存储器在参数传递过程中不传递值, 在分配时不被初始化, 也没有任何数值。

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

模拟量输入: AI

S7-200将模拟量值(如温度或电压)转换成1个字长(16位)的数字量。可以用区域标识符(AI)、数据长度(W)及字节的起始地址来存取这些值。因为模拟输入量为1个字长, 且从偶数位字节(如0, 2, 4)开始, 所以必须用偶数字节地址(如AIW0, AIW2, AIW4)来存取这些值。模拟量输入值为只读数据。

格式:	AIW[起始字节地址]	AIW4
-----	-------------	------

模拟量输出: AQ

S7-200把1个字长(16位)数字值按比例转换为电流或电压。可以用区域标识符(AQ)、数据长度(W)及字节的起始地址来改变这些值。因为模拟量为一个字长,且从偶数字节(如0, 2, 4)开始,所以必须用偶数字节地址(如AQW0, AQW2, AQW4)来改变这些值。模拟量输出值是只写数据。

格式: $AQW[\text{起始字节地址}] \quad AQW4$

顺控继电器存储器: S

顺控继电器位(S)用于组织机器操作或者进入等效程序段的步骤。SCR提供控制程序的逻辑分段。可以按位、字节、字或双字来存取S位。

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

实数的格式

实数(浮点数)由32位单精度数表示,其格式按照ANSI/IEEE 754-1985标准中所描述的形式,参见图4-8。实数按照双字长度来存取。

对于S7-200来说,浮点数精确到小数点后第六位。因而当您使用一个浮点数常数时,您最多可以指定到小数点后第六位。

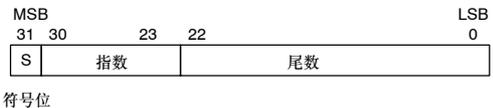


图4-8 实数的格式

实数运算的精度

在计算中涉及到非常大和非常小的数,则有可能导致计算结果不精确。例如 10^x 中当 $x > 6$ 时会出现以下结果。

例如: $100\ 000\ 000+1=100\ 000\ 000$

字符串的格式

字符串指的是一系列字符,每个字符以字节的形式存储。字符串的第一个字节定义了字符串的长度,也就是字符的个数。4-9给出了一个字符串的格式。一个字符串的长度可以是0到254个字符,再加上长度字节,一个字符串的最大长度为255个字节。而一个字符串常量的最大长度为126字节。



图4-9 字符串的格式

在S7-200指令中输入常数值

在S7-200的许多指令中，都可以使用常数值。常数可以是字节、字或者双字。S7-200以二进制数的形式存储常数，可以分别表示十进制数、十六进制数、ASCII码或者实数（浮点数）。见表4-2。

表4-2 常数表示法

数制	格式	举例
十进制	[十进制值]	20047
十六进制	16 # [十六进制值]	16 # 4E4F
二进制	2 # [二进制数]	2#1010_0101_1010_0101
ASCII码	'[ASCII码文本]'	'ABCD'
实数	ANSI/IEEE 754-1985	+1.175495E-38（正数）-1.175495E-38（负数）
字符串	"[字符串文本]"	"ABCDE"



提示

S7-200CPU不支持数据类型检测（例如指定常数存储为一个整数、有符号整数或者双整数）。例如：可以在加法指令中使用VW100中的值作为有符号整数，同时也可以异或指令中将VW100中的数据当作无符号的二进制数。

本地I/O和扩展I/O的寻址

CPU提供的本地I/O具有固定的I/O地址。您可以将扩展模块连接到CPU的右侧来增加I/O点，形成I/O链。对于同种类型的输入输出模块而言，模块的I/O地址取决于I/O类型和模块在I/O链中的位置。举例来说，输出模块不会影响输入模块上的点地址，反之亦然。类似的，模拟量模块不会影响数字量模块的寻址，反之亦然。



提示

数字量模块总是保留以8位（1个字节）递增的过程映象寄存器空间。如果模块没有给保留字节中每一位提供相应的物理点，那些未用位不能分配给I/O链中的后续模块。对于输入模块，这些保留字节中未使用的位会在每个输入刷新周期中被清零。

模拟量I/O点总是以两点递增的方式来分配空间。如果模块没有给每个点分配相应的物理点，则这些I/O点会消失并且不能够分配给I/O链中的后续模块。

图4-10中是一个特定的硬件配置中的I/O地址。地址间隙（用灰色斜体文字表示）无法在程序中使用。

CPU224XP	4入/4出	8入	4模拟量入 1模拟量出	8出	4模拟量入 1模拟量出
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 AIW0 AQW0 AIW2 AQW2 局部I/O	模块0 I2.0 Q2.0 I2.1 Q2.1 I2.2 Q2.2 I2.3 Q2.3 I2.4 Q2.4 I2.5 Q2.5 I2.6 Q2.6 I2.7 Q2.7	模块1 I3.0 I3.1 I3.2 I3.3 I3.4 I3.5 I3.6 I3.7	模块2 AIW4 AQW4 AIW6 AQW6 AIW8 AIW10	模块3 Q3.0 Q3.1 Q3.2 Q3.3 Q3.4 Q3.5 Q3.6 Q3.7	模块4 AIW12 AQW8 AIW14 AQW10 AIW16 AIW18
扩展I/O					

图4-10 CPU224XP的本地和扩展I/O地址举例

用指针对S7-200存储区间接寻址

间接寻址是指用指针来访问存储区数据。指针以双字的形式存储其它存储区的地址。只能用V存储器、L存储器或者累加器寄存器（AC1、AC2、AC3）作为指针。要建立一个指针，必须以双字的形式，将需要间接寻址的存储器地址移动到指针中。指针也可以为子程序传递参数。

S7-200允许指针访问以下存储区：I、Q、V、M、S、AI、AQ、SMT（仅限于当前值）和C（仅限于当前值）。您无法用间接寻址的方式访问位地址，也不能访问HC或者L存储区。

要使用间接寻址，您应该用“&”符号加上要访问的存储区地址来建立一个指针。指令的输入操作数应该以“&”符号开头来表明是存储区的地址，而不是其内容将移动到指令的输出操作数（指针）中。

当指令中的操作数是指针时，应该在操作数前面加上“*”号。如图4-11所示，输入*AC1指定AC1是一个指针，MOVW指令决定了指针指向的是一个字长的数据。在本例中，存储在VB200和VB201中的数值被移动到累加器AC0中。

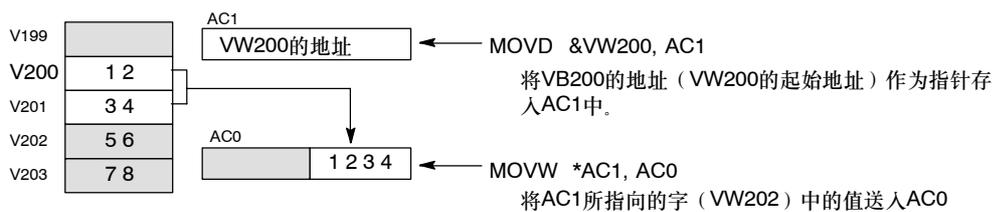


图4-11 创建和使用指针

如图4-12所示，您可以改变一个指针的数值。由于指针是一个32位的数据，要用双字指令来改变指针的数值。简单的数学运算，如加法指令或者递增指令，可用于改变指针的数值。

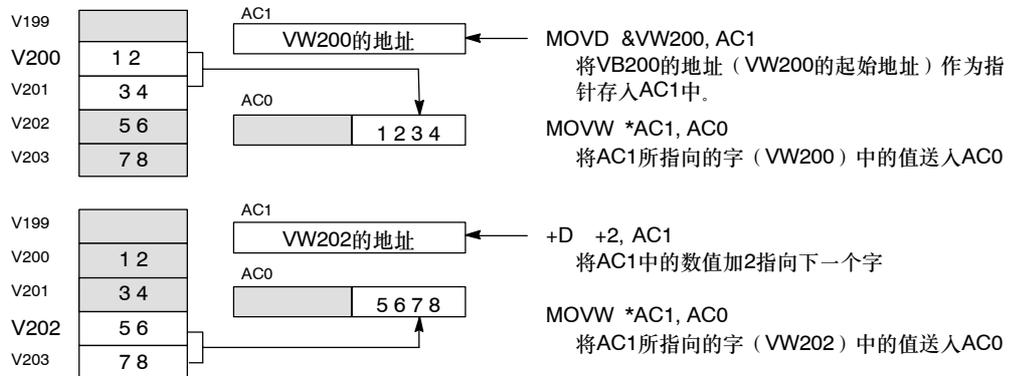


图4-12 改变指针

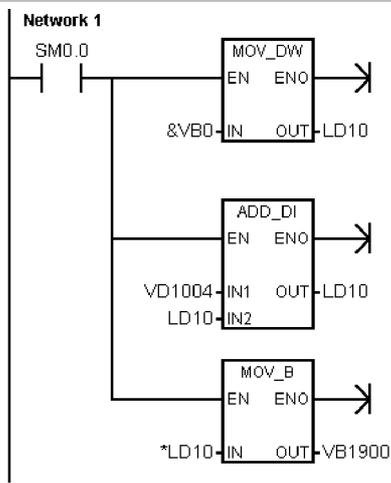


提示

按照所访问的数据长度使用不同的指令：当访问字节时，使用递增指令使指针值加1；当访问字或者计数器、定时器的当前值时，用加法或者递增指令使指针值加2；当访问双字时，使用加法或者递增指令使指针值加4。

用地址偏移量来访问V存储区数据的例子程序

本例中用LD10作为VB0的地址指针。然后可以利用VD1004中存储的地址偏移量来改变指针值。经过改变后，LD10指向V区中的另外一个地址（VB0 + 偏移量）。然后将LD10指向的V区地址中存储的数值复制到VB1900中。通过改变VD1004中的数值，您可以访问V存储器中的任意单元。



```

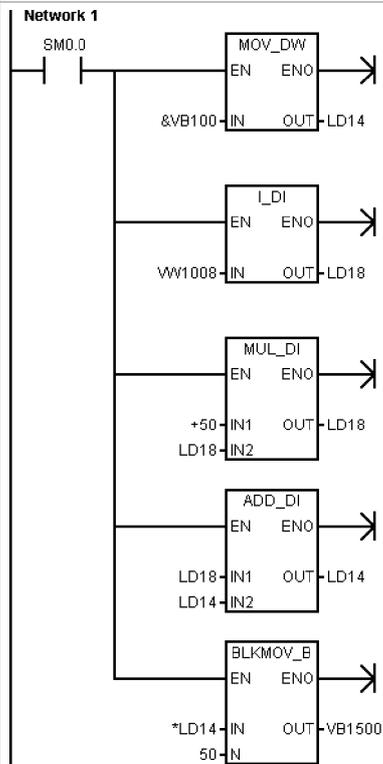
Network1 //如何利用偏移量访问
//V存储器中的任意单元:

//1. 将V区的起始地址装载到指针中。
//2. 将偏移量加到指针上。
//3. 将V存储器中的数值复制到VB1900中。
//

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

用指针访问数据表的例子程序

本例中用LD14作为指向一个配方表的指针，配方表的起始地址为VB100。在本例中VW1008用来存储一个指定的配方在表中的索引号。如果每条配方的长度为50个字节，则用这个索引号乘以50就可以得到这条配方起始地址的偏移量。用指针加上偏移量，您就可以访问表中的每一条配方。在本例中，配方被复制到从VB1500开始的50个字节中。



Network1 //如何从配方表中传递一条配方:
//每条配方的长度为50个字节。
// - 索引值 (VW1008) 指定装载哪一条配方。

- //1. 建立一个指针指向配方表的起始地址。
- //2. 将配方的索引值转换为双字。
- //3. 乘以每条配方的长度算出偏移量。
- //4. 将偏移量加到指针上。
- //5. 将配方传递至VB1500到VB1549中。

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

理解S7-200如何保存和存储数据

S7-200提供了多种安全措施来确保用户程序、程序数据和组态数据不丢失。

- 保持数据存储区 - 由用户选定的数据存储区，在一次上电周期中，只要超级电容和可选电池卡不放电，该存储器的数据就不会改变。在所有存储区中，只有V、M、定时器和计数器存储区能被组态为保持存储区。
- 永久存储器 - 不可变存储器，用来储存程序块、数据块、系统块、强制值、组态为掉电保存的M存储器和在用户程序的控制下写入的指定值。
- 存储卡 - 可拆卸的不可变存储器，用来储存程序块、数据块、系统块、配方、数据归档和强制值。

通过S7-200资源管理器，您可以将文档文件（doc、text、pdf等）储存在存储卡内，也可以将普通文件保留在存储卡中（复制、删除、创建目录和放置文件）。

要安装存储卡，应先从S7-200 CPU上取下塑料盖，然后将存储卡插入槽中。正确安装存储卡至关重要。

小心

静电放电会损坏存储卡或CPU接口。

当您拿存储卡时，您应使用接地导电垫或者戴接地手套，应当把存储卡存放在导电容器中。

下载和上载用户程序

用户程序包括以下几个部分：

- 程序块
- 数据块（可选）
- 系统块（可选）
- 配方（可选）
- 数据归档组态（可选）

当您下载程序时，出于安全考虑，程序块、数据块和系统块将储存在永久存储器中。而配方和数据归档组态将储存在存储卡中，并更新原有的配方和数据归档。那些不涉及下载操作的程序部分也将保留在永久存储器和存储卡中，保持不变。

如果程序下载涉及到配方或数据归档组态，则存储卡就必须一直装在S7-200上，否则程序可能无法正常运行。

将用户程序下载至S7-200 CPU：

1. 选中**File > Download** 菜单命令。
2. 点击需下载的程序部分。
3. 点击Download按钮。

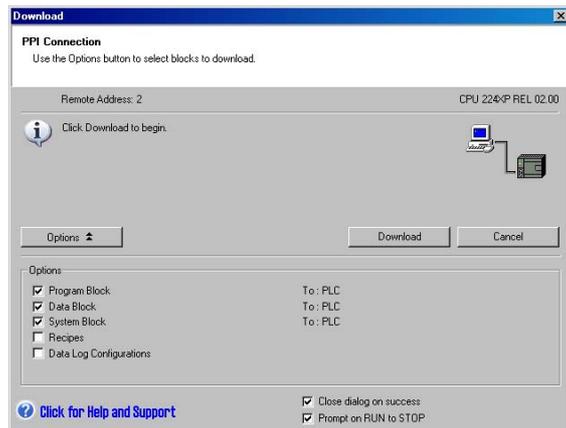


图4-13 下载程序到S7-200 CPU

当您通过STEP 7-Micro/WIN上传程序时，S7-200将从永久存储器中上传程序块、数据块和系统块。同时从存储卡中上传配方和数据归档组态。数据归档中的数据不会通过STEP 7-Micro/WIN上传至计算机，而是通过S7-200资源管理器上传（见14章）。

从S7-200 CPU上传用户程序：

1. 点击**File > Upload**菜单命令。
2. 点击选中需上传的程序部分。
3. 点击Upload按钮。

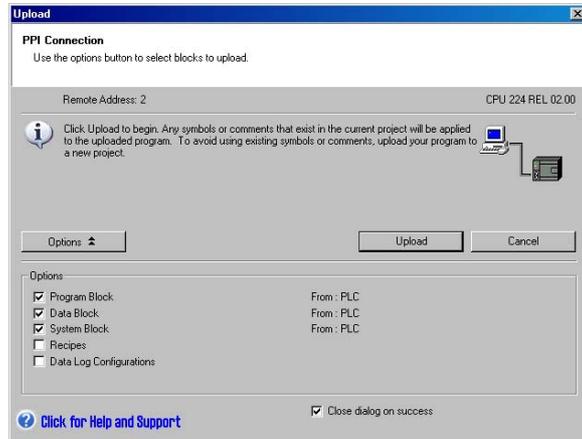


图4-14 从S7-200上传程序

将应用程序存储到存储卡上

使用S7-200时，您可以用存储卡将用户程序拷贝至其它计算机，还可以共享块更新，这些块包括：程序块、系统块和数据块。

在将程序部分复制到存储卡之前，STEP 7-Micro/WIN将删除所有程序部分（包括配方和数据归档），但存储卡中的用户文件不会删除。如果文件太大，程序没有足够的存储空间，则您可以通过两种方法来整理储存空间：点击**PLC > Erase Memory Cartridge**菜单命令清空存储卡；或者打开S7-200资源管理器，移除不需要的文件。

对存储卡进行编程，PLC必须进入STOP模式。

将程序存储到存储卡中：

1. 选中**PLC > Program Memory Cartridge**菜单命令
2. 点击选中需复制到存储卡的程序部分（缺省情况下，用户程序中的所有程序部分都会选中）。如果选中了系统块，则强制值也会被复制。
3. 点击Program按钮。

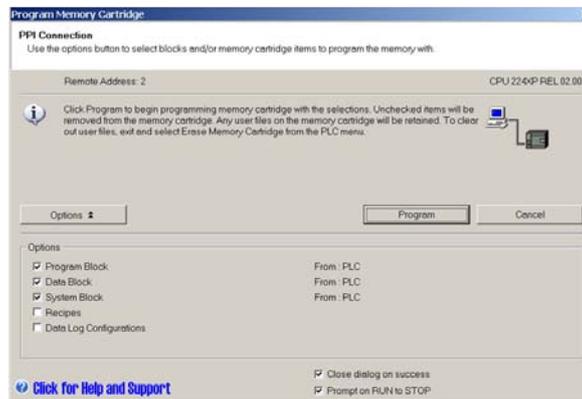


图4-15 将应用程序存储到存储卡上

程序块、系统块和强制值将从S7-200的永久存储器复制到存储卡中。配方和数据归档组态将从STEP 7-Micro/WIN复制到存储卡。

从存储卡中恢复程序

要将应用程序从存储卡上传送至S7-200，您必须先插上存储卡，然后给S7-200上电。只要存储卡中有块或强制值与S7-200中的块和强制值不同，则存储卡中的所有块都会复制给S7-200。

- 如果S7-200从存储卡中复制了程序块，则永久存储器中的程序块就会被替换掉。
- 如果从存储卡中复制的是数据块，则永久存储器中的数据块会被替换掉，而V存储器会清空，然后写入数据块的内容。
- 如果从存储卡中复制的是系统块，则永久存储器中的系统块和强制值会被替换掉，且所有的保持存储器都会清空。

当程序复制到永久存储器之后，就可以取下存储卡了。然而，如果存储卡里有配方和数据归档，则就必须将存储卡一直装在S7-200上。但如果S7-200装有存储卡，则在上电后，它进入运行模式的时间会延长。

注释

如果S7-200 CPU装有一个存储卡，而该卡是由其它型号的S7-200 CPU编程的，那么该S7-200在上电后可能会报错。高型号CPU可以读取低型号CPU编程的存储卡，反之则不行。比如，CPU224可以读取CPU221或CPU222编程的存储卡，但CPU221或CPU222却不能读取CPU224编程的存储卡。

关于存储卡使用限制的完整列表，请见附录A中的选件卡（存储卡）。

掉电后保存M存储区

如果位存储器（MB0到MB13）前14字节中的任何一个被指定为保持，则当S7-200掉电时，这些字节会被保存到永久存储器中。缺省情况下，M存储器的前14位是不保持的。

开机后数据的恢复

上电之后，S7-200将从永久存储器中恢复程序块和系统块。然后S7-200将检查超级电容和可选电池卡是否安装。如果是，则它将确认数据是否已成功地存储到RAM存储器。如果数据保存成功，则用户数据存储器的保持区将保持不变。而V存储器的非保持部分将根据永久存储器中的数据块内容来恢复。其它存储区的非保持部分则被清空。

如果RAM的内容没有保留（比如在扩展电源故障之后），则S7-200将清除所有用户数据区，将保持数据丢失存储器位（SM0.2）置位，并读取永久存储器的数据块内容来恢复V存储器，如果M存储器的前14位已预设保持，则S7-200还将读取永久存储器恢复这些位的内容。

通过编程方式将V存储器保存至永久存储器

可以将存储在V存储器中的数据（字节、字或双字）存储到永久存储器中。一般来说，一个写永久存储器的操作会使扫描周期最多增加5ms。通过保存操作所写入的数据会覆盖先前永久存储器中V存储区的数据。

保存至永久存储器的操作并不更新存储卡中的数据。



提示

由于保存至永久存储器（EEPROM）的操作次数是有限的（最少10万次，典型值为100万次），所以请注意只在必要时才进行保存操作。否则，EEPROM可能会失效，从而引起CPU故障。一般来说，当特定事件发生时，才执行存储操作，而特定事件是不很频繁发生的。

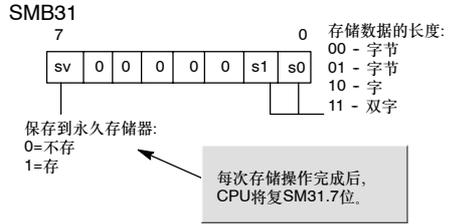
例如，如果S7-200扫描周期为50ms，每次扫描存一个数据，则EEPROM最短只能工作5,000秒，还不到一个半小时。另一方面，如果每小时存一个数据，则EEPROM至少可以工作11年。

将V存储器复制至永久存储器

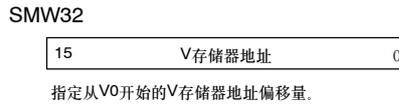
特殊存储器字节31（SMB31）命令S7-200将V存储区中的某个值复制到永久存储器的V存储区。特殊存储器字32（SMW32）中存储所要复制数据的地址。图4-16给出了SMB31和SMW32的格式。

采用下列步骤来保存或者写入V存储区中的一个特定数值：

1. 将要保存的V存储器的地址装载到SMW32中。
2. 将数据长度装载入SM31.0和SM31.1，如图4-16所示。
3. 将SM31.7置为1。



在每次扫描的末尾，CPU自动检查SM31.7，如果SM31.7为1，则将指定值保存到永久存储器。当CPU将SM31.7清零时，操作结束。



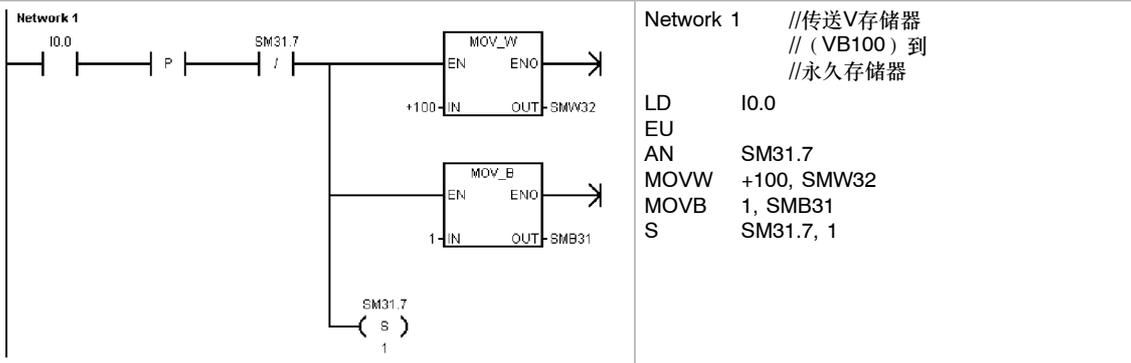
在保存操作完成之前，不要改变V存储器中的数值。

图4-16 SMB31和SMW32

例子程序将V存储器复制至永久存储器

在本例中，VB100将存入永久存储器。当I0.0有上升沿信号产生，并且没有其它写EEPROM操作发生时，将V存储器的地址传送到SMW32。接着选择所要传输数据的长度（1代表字节，2代表字，3代表双字或者实数）。然后将SM31.7置位，S7-200会在扫描末尾传送数据。

当传送完成后，S7-200会自动复位SM31.7。



为S7-200 CPU选择操作模式

S7-200有两种操作模式：停止模式和运行模式。CPU前面板上的LED状态显示了当前的操作模式。在停止模式下，S7-200不执行程序，您可以下载程序和CPU组态。在运行模式下，S7-200将运行程序。

- S7-200提供一个模式开关来改变操作模式。您可以用模式开关（位于S7-200前盖下面）手动选择操作模式：可以将模式开关打在停止模式，停止程序的执行；可以将模式开关打在运行模式，启动程序的执行；也可以将模式开关打在TERM（终端）模式，不改变当前操作模式。

如果模式开关打在STOP或者TERM模式，且电源状态发生变化，则当电源恢复时，CPU会自动进入STOP模式。如果模式开关打在RUN模式，且电源状态发生变化，则当电源恢复时，CPU会进入RUN模式。

- STEP7-Micro/WIN允许您改变与之相连的S7-200的操作模式。如果希望用软件来改变操作模式，CPU上的模式开关必须打在RUN或者TERM上。您可以用菜单命令中的**PLC > STOP**和**PLC > RUN**或者工具栏中的相关按钮来改变操作模式。
- 您可以在应用程序中插入STOP指令来将S7-200置为停止模式。它可以使逻辑程序停止运行。有关STOP指令的更多信息参见第6章。

使用S7-200资源管理器

S7-200资源管理器是Windows资源管理器应用的扩展，能用来访问S7-200 PLC并可以浏览PLC相关资源。能对不同的块作出判断，指明它们是存放在PLC中还是存储卡中。另外，它还能指出每个块的属性。

由于S7-200资源管理器是Windows资源管理器应用的扩展，因此它能支持标准的Windows浏览及其行为特性。

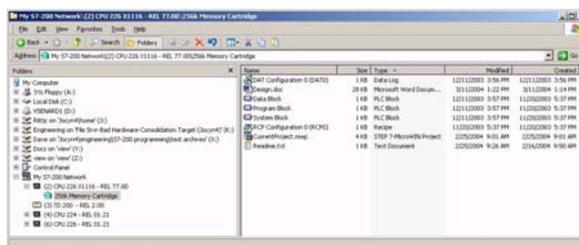


图4-17 S7-200资源管理器

S7-200资源管理器可用来读取存储卡中的数据归档。关于数据归档的更多信息，可参见第14章。

S7-200资源管理器还可以用于读取存储卡中的用户文件，或将用户文件写入存储卡。这些用户文件的类型包括：Word文档、位图文件、JPEG文件或STEP 7-Micro/WIN项目。

S7-200的特性

S7-200提供了几条特殊的性能帮助您更好地运用S7-200的功能，完成应用程序。

S7-200允许您在程序中立即读写I/O

在S7-200的指令集中提供了立即读写物理I/O点的指令。尽管通常情况下我们使用映象寄存器作为源地址和目的地址来访问I/O，但这些立即I/O指令却允许我们直接访问真正的输入、输出点。

当使用立即指令访问一个输入点时，相应的过程映象输入寄存器不会发生改变。而当您使用立即指令访问一个输出点时，相应的过程映象输出寄存器会被同步刷新。



提示

除非您使能了模拟量输入滤波功能，否则S7-200将直接从模块上读取模拟量输入值。当您将一个数值写到模拟量输出时，输出值会立即更新。

通常认为在执行应用程序时，用过程映象寄存器会比使用直接访问输入、输出具有优越性。之所以这样有以下三个原因：

- 所有输入点的采样是在扫描周期的一开始同步进行的。在整个扫描周期的程序执行过程中输入值被冻结。而输出点按照映象寄存器中的值刷新是在程序执行完成之后。这样会使系统更加稳定。
- 访问映象寄存器的速度比直接访问I/O点要快，有利于程序快速运行。
- I/O点是位实体，只能按位或者字节来访问，而您可以按位、字节、字或者双字的形式来访问映象寄存器。通过这种方式，映象寄存器将为您提供额外的灵活性。

S7-200允许在程序扫描周期中使用中断

如果您使用了中断，与中断事件相关的中断服务程序作为程序的一部分被保存。中断程序并不作为正常扫描周期的一部分来执行，而是当中断事件发生时才执行（可能在扫描周期的任意点）。

在中断优先级相同的情况下，S7-200遵循先来先服务的原则来执行中断服务程序。关于中断指令的更多信息，请参见第6章。

S7-200允许您为运行模式编辑和执行状态分配处理时间

您可以设定一个扫描周期的百分比用来处理运行模式编辑或执行状态相关的通讯请求。（运行模式编辑和执行状态是STEP 7-Micro/WIN提供的备选功能，能使您更轻松地进行调试程序。）在您增加用于通讯请求处理时间百分比的同时，扫描时间也会随之增加，从而会导致控制过程运行速度变慢。

在默认设置下，用于处理运行模式编辑和执行状态通讯请求的时间百分比为10%。这个默认设置为在对控制过程影响最小的前提下处理编译和状态操作，提供了一个合理的时间。您可以在5%到50%之间调节这个值。要想设置背景通讯的扫描周期时间片，按以下步骤：

1. 点选**View > Component > System Block**菜单命令并选中Background Time。
2. 在Background标签下，通过下拉选框选择通讯背景时间。
3. 点击OK保存您的选择。
4. 将改变后的系统块下载到S7-200中。

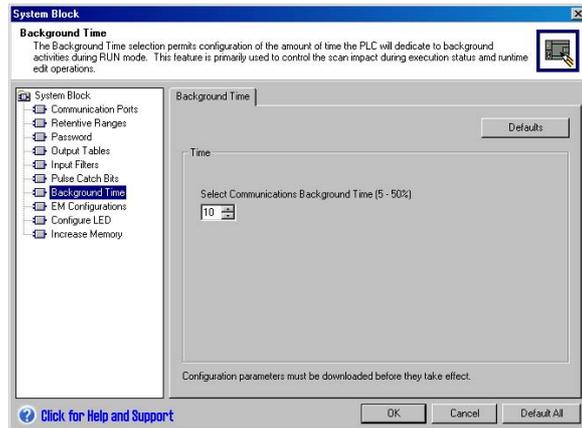


图4-18 通讯背景时间

S7-200允许您设置停止模式下的数字量输出状态

S7-200的输出表允许您选择STOP模式下的输出状态，是将已知值传送到数字量输出点，还是使输出保持STOP模式之前的状态。输出表是系统块的一部分，它被下载并储存在S7-200中。

1. 点选**View > Component > System Block**菜单命令并选中输出表。点击Digital标签。
2. 如果要冻结上一个状态的输出，选择Freeze Outputs复选框。
3. 如果要将输出表中的值复制到输出点上，则要填写输出表。在您希望从运行到停止模式转换后置1的相应位置上点击。输出表的缺省设置全部为0。
4. 点击OK保存您的选择。
5. 将改变后的系统块下载到S7-200中。

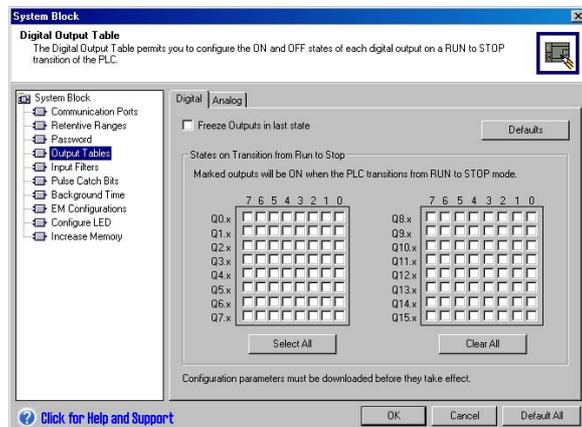


图4-19 数字量输出表

S7-200允许您设置模拟量输出值

模拟量输出表可以用来设置模拟量输出点，指明在从运行模式进入停止模式后，是将已知值传送至模拟量输出点，还是使输出保持停止模式之前的状态。模拟量输出表是下载并储存在S7-200中的系统块的一部分。

1. 點選 **View > Component > System Block** 菜单命令并选中 **Output Time**。点击 **Analog** 标签。
2. 如果要冻结上一个状态的输出，选择 **Freeze Outputs** 复选框。
3. 在从运行模式进入停止模式时，通过冻结值表，您可以将模拟量输出点设为一个已知值（-32768到37262）。
4. 点击 **OK** 保存您的选择。
5. 将改变后的系统块下载到S7-200中。

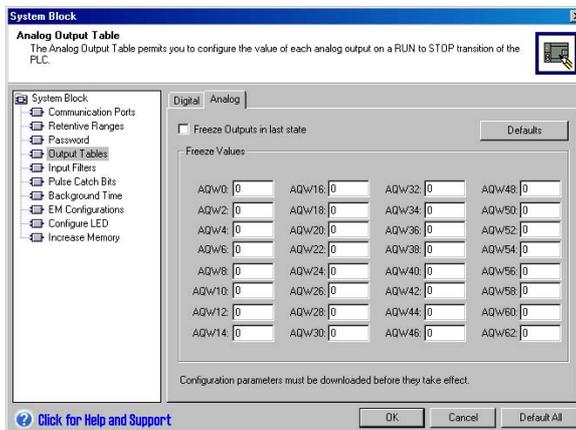


图4-20 模拟量输出表

S7-200允许您定义掉电保持存储区

如果您希望在掉电后仍然保持存储区中的数据，您可以定义最多六个掉电保持区的地址范围。在掉电保持区中您可以使用以下存储区的地址范围：V、M、C和T。只有保持型定时器（TONR）可以设为掉电保持的。在缺省情况下，M存储器的前14个字节是非保持的。

对于定时器和计数器来说，只有当前值可以保持，定时器位和计数器位是不能保持的。



提示

如果将地址范围MB0至MB13定义为保持，则在掉电时，这个区域里的值就会自动保存到永久存储器中。

按照以下步骤设置掉电保持区：

1. 點選 **View > Component > System Block** 菜单命令并选中 **Retentive Ranges**。
2. 设置掉电保持区的范围并单击 **OK**。
3. 将改变后的系统块下载到S7-200中。

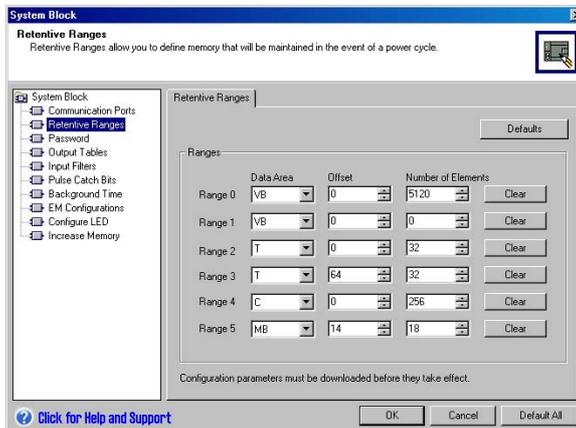


图4-21 保持存储器

S7-200允许您对数字量输入加滤波器

S7-200允许您为某些或者全部本机数字量输入点选择输入滤波器，并为滤波器定义延迟时间（从0.2ms到12.8ms可选）。这一延迟时间有助于滤除输入杂波，从而减小了输入状态发生意外改变的可能。

输入滤波器是系统块的一部分，它被下载并存储在CPU中。滤波器延迟时间的缺省值为6.4ms。如图4-22所示，一组输入点共用一个延迟时间。

按照以下步骤设置输入滤波器延迟时间：

1. 点选**View > Component > System Block**菜单命令并选中**Input Filters**。点击**Digital**标签。
2. 为每一组输入指定延迟时间。
3. 将改变后的系统块下载到S7-200中。

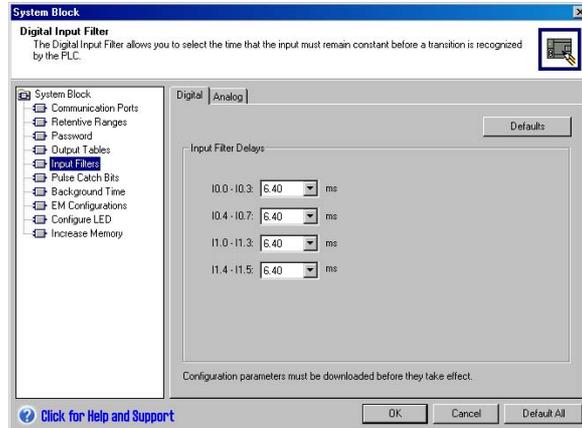


图4-22 数字输入滤波器



提示

数字量输入滤波器会对读输入指令、输入中断和脉冲捕捉产生影响。根据您的选择，应用程序有可能丢掉一个中断事件或者脉冲捕捉。高速计数器不受此影响。

S7-200允许您对模拟量输入加滤波器

S7-200允许您对每一路模拟量输入选择软件滤波器。滤波值是多个模拟量输入采样值的平均值。滤波器参数（采样次数和死区）对于允许滤波的所有模拟量输入是相同的。

滤波器具有快速响应的特点，可以反映信号的快速变化。当输入与平均值的差超过设定的变化时，滤波器对最近的模拟量输入值产生一个阶跃函数。这个差称为死区，并用模拟量输入的数字信号设定。

缺省配置将对所有的模拟量输入进行滤波，但CPU224XP上的AIW0和AIW2除外。

1. 点选**View > Component > System Block**菜单命令并选中**Input Filters**。点击**Analog**标签。
2. 选择需要滤波的模拟量输入、采样个数和死区。
3. 单击**OK**。
4. 将改变后的系统块下载到S7-200中。

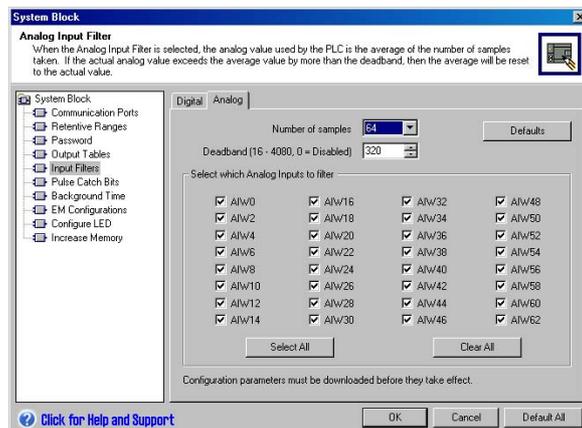


图4-23 模拟量输入滤波器

**提示**

不要对在模拟量字中传递数字信息或者报警指示的模块使用模拟量输入滤波。对于RTD、TC和ASI主站模块，不能使用模拟量输入滤波。

**提示**

CPU224XP上的AIW0和AIW2由模数转换器来滤波，且通常无需另外的软件滤波。

S7-200允许您捕捉窄脉冲

S7-200为每个本机数字量输入提供脉冲捕捉功能。脉冲捕捉功能允许PLC捕捉到持续时间很短的高电平脉冲或者低电平脉冲。而在扫描周期的开始，这些脉冲不是总能被CPU读到。当一个输入设置了脉冲捕捉功能时，输入端的状态变化被锁存并一直保持到下一个扫描循环刷新。这就确保了一个持续时间很短的脉冲能被捕捉到，并一直保持到S7-200读取输入点。

可以分别使能每一个本机数字量输入点的脉冲捕捉功能。

按照以下步骤设置脉冲捕捉：

1. 点选 **View > Component > System Block** 菜单命令并选中 **Pulse Catch Bits**。
2. 点击相应的复选框并点击OK。
3. 将改变后的系统块下载到S7-200中。

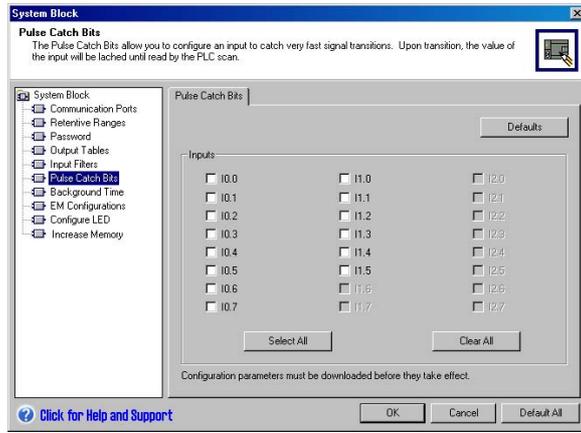


图4-24 脉冲捕捉使能

在使能或禁止脉冲捕捉的情况下，S7-200可进行的基本操作如图4-25所示。



图4-25 带有和不带有脉冲捕捉功能的S7-200操作

由于脉冲是在通过了输入滤波器之后，才能够被捕捉到，因而要调整输入滤波时间，确保脉冲不被滤掉。图4-26为数字量输入电路的方框图。



图4-26 数字量输入电路

对于不同的输入条件，脉冲捕捉功能的响应如图4-27所示。如果在给定的扫描周期中有不止一个脉冲，则只有第一个脉冲被读到。这种情况下，您应该使用上升/下降沿中断事件。（表6-46中给出了中断事件列表。）

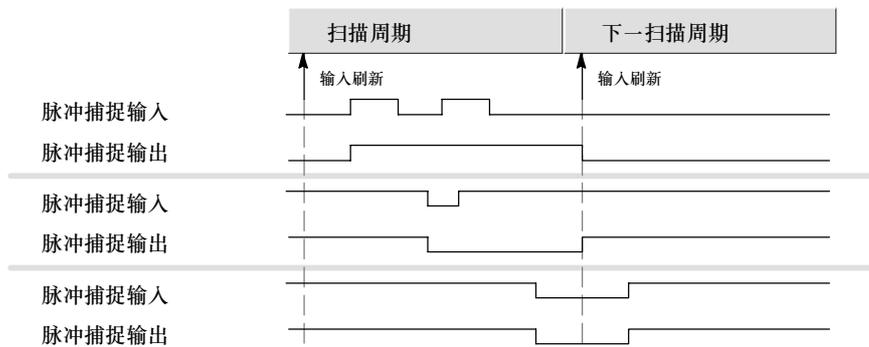


图4-27 不同输入条件下的脉冲捕捉响应

S7-200的用户控制LED

S7-200提供了一个可以发红光（系统故障LED）或黄光（诊断LED）的LED（SF/DIAG）。在用户程序的控制下，或在某些特定情况下，诊断LED都能点亮，这些情况包括：I/O点或数据值被强制，或模块发生I/O错误。

设置诊断LED自动点亮:

1. 点选 **View > Component > System Block** 菜单命令并选中 **Configure LED**。
2. 点击各选项，以指定在I/O点或数据被强制，或者模块发生I/O错误时，LED是否点亮。
3. 将改变后的系统块下载到S7-200中。

若要通过用户程序控制诊断LED的状态，可参见第6章所述的诊断LED指令。

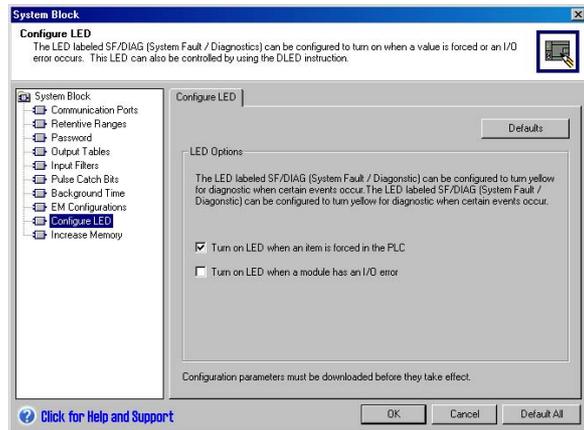


图4-28 诊断LED

S7-200保留一份关于主要CPU事件的历史归档

S7-200会保留一份关于主要CPU事件的历史归档，该归档带有时间标记，所归档的内容包括：何时上电、CPU何时进入运行模式，以及何时出现致命错误。在设置了实时时钟之后，归档条目就会带有正确的时间和日期。

要查看事件历史归档，可点击**PLC > Information**菜单命令并选择Event History。

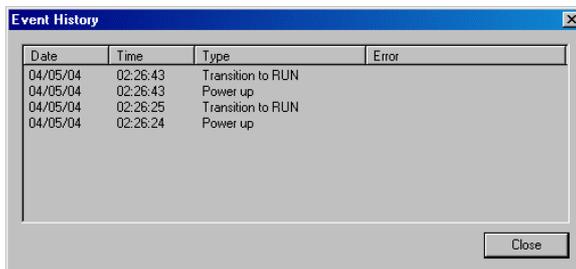


图4-29 查看事件历史归档

S7-200允许您增大可用用户程序存储器

在S7-200的CPU224、CPU 224XP和CPU226上，您可以禁用运行模式编辑功能，从而增大可用用户程序存储器的容量。关于各型CPU的程序存储器容量，可参见表1-2。

要禁用运行模式编辑功能，可执行以下步骤：

1. 点击**View Component > System Block**菜单命令并选中**Increase Program Memory**。
2. 点击**Increase Memory**选项，禁用运行模式编辑。
3. 将改变后的系统块下载到S7-200中。

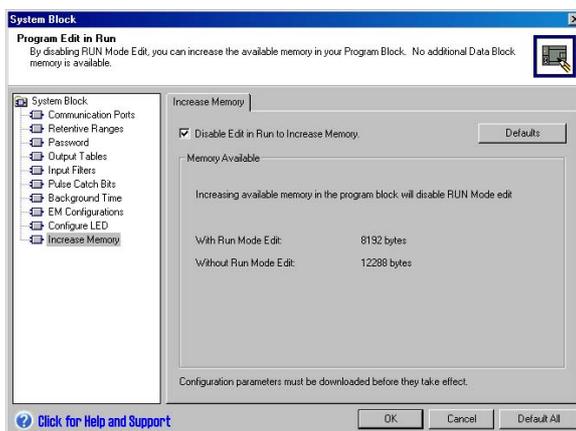


图4-30 禁用运行模式编辑

S7-200提供密码保护功能

S7-200的所有型号都提供密码保护功能，用以限制对特殊功能的访问。

对CPU功能及存储器的访问权限是通过密码来实现的。不设定密码保护，对S7-200的访问没有限制。设置了密码保护，根据安装密码时的设置，CPU禁止所有的受限操作。

密码不区分大小写。

如表4-3所示，S7-200 CPU提供了限制CPU访问功能的三个等级。每个等级允许特定的无需密码的访问功能。若要进行全部三个等级的访问，需输入正确的密码，只有这样您才能访问全部功能。S7-200的缺省设置为等级1（没有限制）。

在网络中输入密码不会对CPU的密码保护有所危害。

表4-3 限制对S7-200的访问

CPU功能	1级	2级	3级
读写用户数据	不限制	不限制	不限制
启动、停止重启CPU			
读写时钟			
上载程序、数据和配置	不限制	不限制	要密码
下载到CPU	不限制	要密码	
监视执行状态			
强制数据或执行程序			
强制数据或执行单/多次扫描			
复制到存储卡			
在STOP模式下写输出			

CPU只允许一个用户使用访问权限，禁止其它用户使用这些功能。同一时刻，只允许一个用户不受限制地访问CPU。



提示

当输入密码后，在编程设备同CPU断开的一分钟之内，该授权等级仍然有效。在断开电缆之前，一定要先退出STEP7-Micro/WIN，以避免其它用户利用编程设备访问CPU。

为S7-200配置密码

如图4-31所示的系统块对话框允许您为S7-200配置密码。S7-200的缺省设置是Full（1级），没有任何限制。

1. 选择**View > Component > System Block**菜单命令，弹出System Block对话框并选择Password。
2. 为S7-200选择合适的访问级别。
3. 输入并确认Partial（2级）或Minimum（3级）密码。
4. 单击OK。
5. 将改变后的系统块下载到S7-200中。

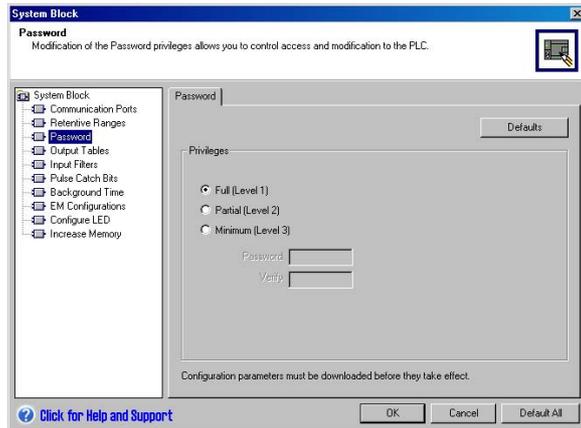


图4-31 创建密码

密码忘记后如何恢复

如果您忘记了密码，您必须清除存储器，重新下载应用程序。清除存储器会使S7-200处于停止模式，并且将S7-200中，除了网络地址、波特率和时钟以外的其它参数恢复到出厂设置。清除S7-200中的程序：

1. 在命令菜单中选择**PLC > Clear**来显示清除对话框。
2. 选择所有的块并点击OK确认。
3. 如果配置了密码，STEP7-Micro/WIN会显示密码授权对话框。要清除密码，在密码授权对话框中输入“CLEARPLC”，就可以继续执行全部清除的操作。（“CLEARPLC”不区分大小写。）

全部清除操作不会去掉存储卡中的程序。由于密码和程序一同保存在存储卡中，因而必须重新写存储卡，才能从程序中去掉密码。



警告

清除S7-200存储器会导致输出关闭（对于模拟量输出来说，会被冻结在某个特定值）。

当清除S7-200存储器时，如果S7-200与其它设备相连，则输出状态的变化会传递到该设备。如果输出的“安全状态”与厂家设定不同，输出的变化可能会引起设备产生不可预料的动作，也可能导致死亡或严重的人身伤害和设备损坏。

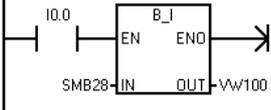
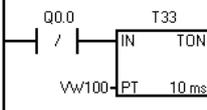
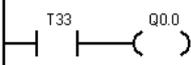
在清除S7-200存储器之前，必须有合适的安全预防措施，确保您的操作是安全的。

S7-200提供模拟电位器

模拟电位器位于模块前盖下面。您可以调节这些电位器来增加或者减小存于特殊存储器中的值（SMB）。这些只读值在程序中可用作很多功能，如更新定时器或计数器的当前值，输入或修改预置值、限定值等。可以用一个小螺丝刀来进行调节：将电位器顺时针（向右）旋转来使数值增大；逆时针（向左）旋转来使数值减小。

SMB28中的数值代表模拟电位器0的位置。SMB29中的数值代表模拟电位器1的位置。模拟电位器的标定范围为0到255，重复度为±2

用模拟电位器输入数值的例子程序

<p>Network 1</p> 	<p>Network1 //读模拟电位器0（SMB28）。 //将数值保存到VW100中。</p> <pre>LD I0.0 BTI SMB28, VW100</pre>
<p>Network 2</p> 	<p>Network2 //用VW100中的整数值作为定时器的预置值。</p> <pre>LDN Q0.0 TON T33, VW100</pre>
<p>Network 3</p> 	<p>Network3 //当T33达到预置值时接通Q0.0。</p> <pre>LD T33 = Q0.0</pre>

S7-200提供高速I/O

高速计数器

S7-200具有集成的高速计数功能，它能够对外部高速事件计数而不影响S7-200的性能。不同CPU支持的计数速率，参见附录A。每个计数器都有专用的输入点作为时钟、方向控制、复位端、启动端等功能输入。在不同的模式下有不同的计数速率。关于高速计数器的更多信息，请见第6章。

高速脉冲输出

S7-200支持高速脉冲输出功能，其输出点Q0.0和Q0.1可形成高速脉冲串输出（PTO）或脉宽调制（PWM）。

PTO输出方波（占空比50%），并可指定所输出的脉冲数量（从1到4,294,967,295）和周期（以微秒或毫秒为单位）。脉冲序列输出（PTO）功能可以编程为产生一系列脉冲或产生由多个脉冲序列组成的脉冲包络。在脉冲包络操作方式中，PTO功能被编程为控制一个步进电机运行一个简单的斜坡上升、运行和斜坡下降操作序列或更复杂的操作序列。

PWM功能提供具有可变占空比的固定周期的输出脉冲，周期和脉宽既可以用微秒又可以用毫秒为单位。当脉宽等于周期时，占空比为100%，输出恒定为1；当脉宽等于0时，占空比为0，输出恒定为0。

关于高速脉冲输出的详细信息，参见第6章。关于在开环运动控制中使用PTO的更多信息，请参见第9章。

5

编程的概念、惯例及特点

S7-200周而复始地执行应用程序，控制一个任务或过程。利用STEP 7-Micro/WIN可以创建一个用户程序并将它下载到S7-200中。STEP 7-Micro/WIN软件中提供了多种工具和特性用于完成和调试应用程序。

本章内容:

设计一个微型PLC系统的指导原则	52
程序的基本组件	53
用STEP 7-Micro/WIN创建用户程序	55
选择SIMATIC和IEC 1131-3指令集	57
理解程序编辑器中使用的惯例	58
使用向导帮您创建控制程序	60
S7-200中的出错处理	60
在数据块中指定地址和初始值	62
用符号表来定义变量的符号地址	62
使用局部变量	63
用状态图来监视用户程序	63
创建一个指令库	64
应用程序的调试	64

设计一个微型PLC系统的指导原则

设计一个微型PLC系统有许多设计方法。以下这些通用的指导原则适用于许多设计项目。当然，您所在公司的规程和您在培训中接受的实践经验是必须遵循的。

分解控制过程或者机器

将您的控制过程或者机器分解成相互独立的部分。分解决定了控制器之间的界限，并将影响功能描述和资源的分配。

创建功能说明

写出过程或者机器每一部分的操作描述。它包括以下内容：I/O点、操作的功能描述、每个执行机构（例如线圈、电机和驱动器等）在动作之前需要满足的状态、操作员接口的描述以及过程或机器与其他部分的接口。

安全电路的设计

识别要求设计硬件安全线路的设备。控制设备在不安全的条件下出现故障，会造成不可预料的启动或者机器操作的变化。在不可预料或者不正确的机器操作会造成人身伤害或严重的财产损失的情况，应该考虑采用独立于S7-200的机电冗余来防止不安全的操作。在设计安全电路时，应考虑以下几点：

- 识别有可能不合适或者不可预料操作有可能会造成危害的执行机构。
- 识别确保操作不发生危害的条件，并决定如何独立于CPU来检测这些条件。
- 识别上电或断电时，CPU和I/O对过程有何影响，识别错误何时被检测出来。这个信息只能用于常规的和可以预料的异常操作，不能用于保障安全的目的。
- 设计独立于CPU的手动或机电冗余来阻止危险的操作。
- 向CPU提供独立电路的状态信息，便于程序和操作员界面得到需要的信息。
- 识别其它与过程安全操作相关的安全要求。

指定操作员站

根据功能描述的要求建立操作员站的配置图。包括如下内容：

- 与过程或者机器有关的每个操作员站的位置总图。
- 操作员站
- 与CPU或扩展模块有关的电气图

创建配置图

根据功能描述的要求建立控制设备的配置图。包括如下内容:

- 和过程或者机器有关的每个CPU的位置图。
- CPU和扩展I/O模块的机械布局图（包括控制柜和其它设备）。
- 每个CPU和扩展模块的电气图（包括设备型号、通讯地址和I/O地址）。

建立符号名表（可选）

如果选择了符号名寻址，需要对绝对地址建立一个符号名表。符号名表不仅包括物理输入/输出信号，还包括程序中用到的其它元件。

程序的基本组件

一个程序块由可执行代码和注释组成。可执行代码由主程序和若干子程序或者中断服务程序组成。可执行代码被编译并下载到S7-200中，而程序注释不会被下载。您可以使用组织组件（主程序、子程序和中断服务程序）来结构化您的控制程序。

以下例子程序包括一个子程序和一个中断服务程序。该例子程序使用一个定时中断，每100 ms读一次模拟量的输入值。

实例: 一个程序的基本组件		
M A I N	Network 1 	Network 1 //在第一个扫描周期，调用子程序。 LD SM0.1 CALL SBR_0
S B R O	Network 1 	Network 1 //设置时间间隔为100 ms //用于定时中断。 //使能定时中断0。 LD SM0.0 MOVB 100, SMB34 ATCH INT_0, 10 ENI
I N T O	Network 1 	Network 1 //采样模拟量输入4 LD SM0.0 MOVW AIW4, VW100

主程序

主程序中包括控制应用的指令。S7-200在每一个扫描周期中顺序执行这些指令。主程序也被表示为OB1。

子程序

子程序是应用程序中的可选组件。只有被主程序、中断服务程序或者其它子程序调用时，子程序才会执行。当您希望重复执行某项功能时，子程序是非常有用的。与其在主程序中的不同位置多次使用相同的程序代码，不如将这段程序逻辑写在子程序中，然后在主程序中需要的地方调用。调用子程序有如下优点：

- ❑ 用子程序可以减小程序的长度
- ❑ 由于将代码从主程序中移出，因而用子程序可以缩短程序扫描周期。S7-200在每个扫描周期中处理主程序中的代码，不管代码是否执行。而子程序只有在被调用时，S7-200才会处理其代码。在不调用子程序时，S7-200不会处理其代码。
- ❑ 用子程序创建的程序代码是可传递的。您可以在一个子程序中完成一个独立的功能，然后将它复制到另一个应用程序中而无需作重复工作。



提示

在子程序中使用V存储器地址会限制它的可移植性。因为一个程序对于V存储器地址的分配有可能与另一个程序对其分配有冲突。相比之下，在子程序中的所有变量地址都使用局部变量（L存储器），会使子程序有极高的可移植性。因为当子程序使用局部变量时，子程序与程序的其它部分之间不会有地址冲突。

中断服务程序

中断服务程序是应用程序中的可选组件。当特定的中断事件发生时，中断服务程序执行。您可以为一个预先定义好的中断事件设计一个中断服务程序。当特定的事件发生时，S7-200会执行中断服务程序。

中断服务程序不会被主程序调用。只有当中断服务程序与一个中断事件相关联，且在该中断事件发生时，S7-200才会执行中断服务程序。



提示

因为无法预测何时会产生中断，所以应考虑尽量限制中断服务程序和程序中其它部分所使用的变量个数。

使用中断服务程序中的局部变量，可以保证中断服务程序只使用临时存储器，并且不会覆盖程序中其它部分使用的数据。

为了保证主程序与中断服务程序正确地共享数据，您可以使用许多编程技巧。关于这些技巧的描述在第6章的中断指令部分有详细说明。

程序中的其它组件

其它块中也包含了S7-200的信息。当您下载程序时，您可以选择同时下载这些块。



系统块

系统块

系统块允许您为S7-200配置不同的硬件参数。



数据块

数据块

数据块存储应用程序中所使用的不同变量值（V存储器）。您可以用数据块输入数据的初始值。

用STEP 7-Micro/WIN创建用户程序

要打开STEP 7-Micro/WIN，可以双击STEP 7-Micro/WIN图标，也可以在命令菜单中选择开始 > SIMATIC > STEP 7 MicroWIN 32 V4.0。如图5-1所示，STEP 7-Micro/WIN项目窗口为创建您的控制程序提供了一个便利的工作环境。

工具栏提供常用的菜单命令的快捷按钮。您可以显示或者隐藏任意工具栏。

操作栏为访问STEP 7-Micro/WIN中不同的程序组件提供了一组图标。

指令树显示了所有的项目对象和创建您的控制程序所需要的指令。您可以将指令从指令树中拖到您的应用程序中，也可用双击指令的方法将该指令插入到程序编辑器中的当前光标所在地。

程序编辑器中包括程序逻辑和局部变量表。您可以在局部变量表中为临时的局部变量定义符号名。在程序编辑器的底部有子程序和中断服务程序的标签。点击这些标签，您可以在主程序、子程序和中断服务程序之间切换。

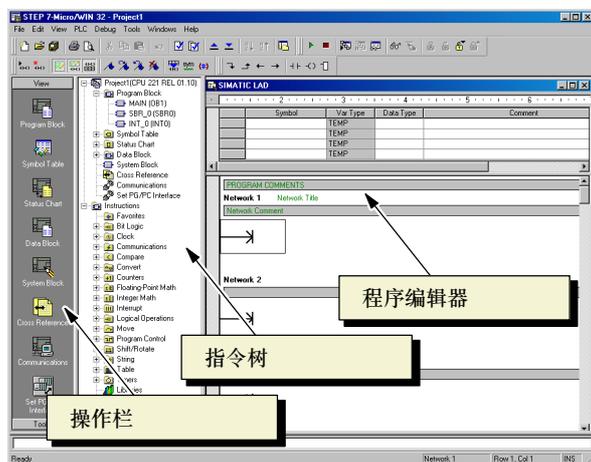


图5-1 STEP 7-Micro/WIN



程序编辑器

STEP 7-Micro/WIN提供三种编辑器来创建您的程序：梯形图（LAD）、语句表（STL）和功能块图（FBD）。尽管有一定限制，但是用任何一种程序编辑器编写的程序都可以用另外一种程序编辑器来浏览和编辑。

STL编辑器的特点

STL编辑器按照文本语言的形式显示程序。STL编辑器允许您输入指令助记符来创建您的控制程序。语句表也允许您创建用LAD和FBD编辑器无法创建的程序。这是因为您在使用S7-200的本族语言进行编程，而在图形编辑器中，为了正确地画出图形，必须遵守一些规则。如图5-2所示，文本方式与汇编语言的编程方式十分相象。

S7-200从上到下按照程序的次序执行每一条指令，然后回到程序的开始重新执行。

STL使用一个逻辑堆栈来分析控制逻辑。您插入STL指令来处理堆栈操作。

LD	I0.0	//读入一个输入
A	I0.1	//和另一个输入进行“与”
=	Q1.0	//向输出1写入值

图5-2 STL程序示例

当您选择STL编辑器时，考虑以下要点

- STL最适合于有经验的程序员。
- STL有时让您能够解决用LAD或者FBD不容易解决的问题。
- 当您使用STL编辑器时，只能使用SIMATIC指令集。
- 虽然您可以用STL编辑器查看或者编辑用LAD或者FBD编辑器编写的程序，但是反之不一定成立。LAD或者FBD编辑器不一定总能显示所有利用STL编辑器编写的程序。

LAD编辑器的特点

LAD编辑器以图形方式显示程序，与电气接线图类似。梯形图程序允许程序仿真来自电源的电流通过一系列的逻辑输入条件，决定是否使能逻辑输出。一个LAD程序包括左侧提供能流的能量线。闭合的触点允许能量通过它们流到下一个元素，而打开的触点阻止能量的流动。

逻辑控制是分段的，程序在同一时间执行一段，从左到右，从上到下。图5-3给出了LAD程序的一个例子。不同的指令用不同的图形符号表示。它包括三种基本形式。

触点代表逻辑输入条件，例如：开关、按钮或者内部条件等。

线圈通常表示逻辑输出结果，例如：灯负载、电机启动器、中间继电器或者内部输出条件。

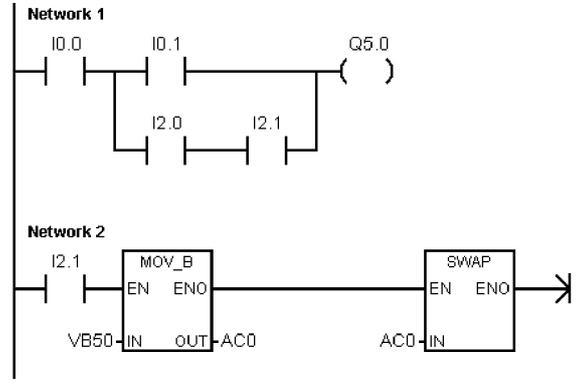


图5-3 LAD程序示例

盒表示其它一些指令，例如：定时器、计数器或者数学运算指令。

当您选择LAD编辑器时，考虑以下要点：

- 梯形图逻辑易于初学者使用。
- 图形表示法易于理解而且全世界通用。
- LAD编辑器能够使用SIMATIC和IEC 1131-3指令集。
- 可以使用STL编辑器显示所有用SIMATIC LAD编辑器编写的程序。

FBD编辑器的特点

FBD编辑器以图形方式显示程序，由通用逻辑门图形组成。在LAD编辑器中看不到触点和线圈，但是有等价的、以盒指令形式出现的指令。

图5-4中给出了FBD程序的一个例子。

FBD不使用左右能量线，因此“能流”这个术语用于表示通过FBD逻辑块控制流这样一个类似的概念。

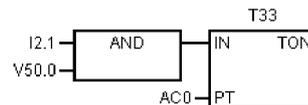


图5-4 FBD程序示例

逻辑“1”通过FBD元素称为能流。能流的原始输入和最终的输出可以直接分配给操作数。

程序逻辑由这些盒指令之间的连接决定。也就是说，一条指令（例如AND盒）的输出可以用来允许另一条指令（例如定时器），这样可以建立所需要的控制逻辑。这样的连接概念使您可以解决各种各样的逻辑问题。

当您选择FBD编辑器时，考虑以下要点：

- 图形逻辑门的表示形式有利于程序流的跟踪。
- FBD编辑器能够使用SIMATIC和IEC 1131-3指令集。
- 可以使用STL编辑器显示所有用SIMATIC FBD编辑器编写的程序。

SIMATIC和IEC 1131-3指令集的选择

大部分PLC产品提供相似的基本指令，但是不同厂商的PLC产品在它们的表示和操作上常常有小的差别。近年来，国际电工委员会（IEC）推出了一个有关PLC编程各个方面的一个全球标准。这个标准鼓励不同的PLC厂商向用户提供与IEC指令集的表示和操作一致的指令。

S7-200提供两种指令集用于完成各种自动化任务。IEC指令集符合PLC编程的IEC 1131-3标准，而SIMATIC指令集是专门为S7-200设计的。



提示

当在STEP 7-Micro/WIN中选择IEC模式时，指令树中不符合IEC1131-3标准的指令旁边会显示一个红色的菱形（◆）。

在SIMATIC指令集和IEC1131-3指令集之间有一些主要区别。

- ❑ IEC 1131-3 指令集是不同PLC厂商的指令标准。SIMATIC指令集中的一些指令并不是IEC1131-3规范中的标准指令。这些是仍在使用的非标准指令，但是如果使用它们，程序就不再严格的与IEC1131-3兼容。
- ❑ 一些指令可以接受多种数据格式，这个概念通常指多重功能。例如，数学指令盒中不区分ADD_I（整数加法）和ADD_R（实数加法），而是在加法指令中检查被加数的格式，并自动选择正确的CPU指令。这样可以节省宝贵的程序设计时间。
- ❑ 当使用IEC1131-3指令时，自动检查指令参数并选择合适的数据格式。数据格式检查不需要用户介入。例如，如果您给一个位操作指令输入一个整数值，就会出现一个错误。这样，有助于减少编程的语法错误。

在选择SIMATIC或IEC指令集时，应考虑以下因素：

- ❑ SIMATIC指令通常执行时间最短。一些IEC指令的执行时间较长。
- ❑ 一些IEC指令与SIMATIC指令操作数不同，例如定时器指令、计数器指令、乘法指令和除法指令等。
- ❑ 您可以在全部的三种程序编辑器（LAD、STL、FBD）中使用SIMATIC指令集，但只能在LAD和FBD编辑器中使用IEC指令。
- ❑ 对于不同品牌的PLC，IEC指令的操作是标准的，因而创建IEC程序的知识与PLC操作平台无关。
- ❑ 因为IEC标准中定义的指令少于SIMATIC指令集，因而可以用SIMATIC指令完成更多功能。
- ❑ IEC1131-3规定变量必须使用类型声明，而且支持系统数据类型检查。

理解程序编辑器中使用的惯例

STEP 7-Micro/WIN在所有程序编辑器中使用以下惯例:

- 在符号名前加# (# Var1) 表示该符号为局部变量。
- 在IEC指令中 % 表示直接地址。
- 操作数符号“?.?”或“????”表示需要一个操作数组态。

LAD程序被分为程序段。一个程序段是按照顺序安排的以一个完整电路的形式连接在一起的触点、线圈和盒，不能短路或者开路，也不能有能流倒流的现象存在。STEP 7-Micro/WIN允许您为LAD程序中的每一个程序段加注释。FBD编程使用网络的概念对程序进行分段和注释。

STL程序不用分段，但是您可以用关键词NETWORK将程序分段。

LAD编辑器中使用的惯例

在LAD编辑器中，您可以使用F4、F6和F9来快速输入触点、盒和线圈指令。LAD编辑器使用下列惯例:

- 符号“--->>”表示开路或者需要能流连接。
- 符号“→”表示指令输出能流，可以级连或串联。
- 符号“>>”表示您可以使用能流。

FBD编辑器中使用的惯例

在FBD编辑器中，您可以使用F4、F6和F9来快速输入AND、OR和盒指令。FBD编辑器使用下列惯例:

- 在EN操作数上的符号“--->>”表示能流或者操作数指示器。它也可用于表示开路或者需要能流连接。
- 符号“→”表示指令输出能流，可以级连或串联。
- 符号“<<”和“>>”表示您可以使用数值或能流。
- 反向圈：操作数或者能流的负逻辑或者反向输入表示为在输入端加一个小圆圈。在图5-5中，Q0.0等于I0.0的非和I0.1与的结果。反向圈仅用于能够作为参数或能流的布尔信号。

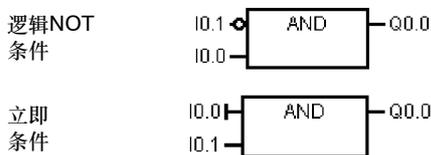


图5-5 FBD惯例

- 立即输入：如图5-5中所示，在FBD编辑器中，用在FBD指令输入端加一条垂直线的方法来表示布尔操作数的立即输入。立即输入直接从物理输入点上读取数据。立即操作数只能用物理输入点。
- 没有输入或者输出的盒：一个盒没有输入意味着这条指令与能流无关。



提示

AND和OR指令的操作数的个数可以扩展到最多32个。要增加或者减少操作数的个数，用键盘上的“+”或者“-”。

S7-200编程的通用惯例

EN/ENO的定义

EN（使能输入）是LAD和FBD中盒的布尔输入。要使盒指令执行，必须使能流到达这个输入。在STL中，指令没有EN输入，但是要想使STL指令执行，堆栈顶部的逻辑值必须是“1”。

ENO（使能输出）是LAD和FBD中盒的布尔输出。如果盒的EN输入有能流并且指令正确执行，则ENO输出会将能流传递给下一元素。如果指令的执行出错，则能流在出错的盒指令处被中断。

在STL中没有使能输出，但是STL指令象相关的有ENO输出的LAD和FBD指令一样，置位一个特殊的ENO位。这个位可以用AND ENO（AENO）指令访问，并且可以产生与盒的ENO位相同的作用。



提示

EN/ENO操作数的数据类型并没有在每条指令中的操作数表中给以说明，因为这一操作数在所有LAD和FBD指令中都是一样的。表5-1列出了这些LAD和FBD中的操作数和数据类型。这些操作数对本手册中介绍的所有LAD和FBD指令均适用。

表5-1 LAD和FBD中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	能流	BOOL
FBD	EN、ENO	I、Q、V、M、SM、S、T、C、L	BOOL

条件输入/无条件输入

在LAD和FBD中，如果一个盒或者线圈的左侧没有任何元素，则它与能流有关。如果一个盒或者线圈的左侧直接连接到能量线上，则它与能流无关。表5-2展示了一个既有条件输入又有无条件输入的实例。

表5-2 条件输入和无条件输入表示方法

能流	LAD	FBD
与能流有关的指令（条件输入）		
与能流无关的指令（无条件输入）		

没有输出的指令

无法级连的盒指令被表示为没有布尔输出。这些包括子程序调用、跳转和条件返回指令。梯形线圈也只能放在能量线之后。这些指令包括标签、装载SCR、SCR条件结束和SCR结束指令。它们在FBD中以盒指令的形式表示，并以无标签的能量输入和无输出来辨别。

比较指令

无论是否有能流，比较指令都会被执行。如果无能流则输出0。如果有能流，输出值取决于比较结果。虽然是作为一个触点来执行操作，但是SIMATIC FBD、IEC梯形图和IEC FBD比较指令都是以盒的形式表示的。

使用向导帮您创建控制程序

STEP 7-Micro/WIN提供向导使您的编程变得更自动更容易。在第章中，具有相关向导的指令会有以下指令向导图标：



指令向导

S7-200中的出错处理

S7-200将错误分为致命错误和非致命错误。您可以通过选择**PLC > Information**菜单命令，来查看因错误而产生的错误代码。

图5-6给出了PLC信息对话框，其中包括错误代码和错误描述。

Last Fatal 区显示S7-200发生的前一致命错误代码。如果RAM区是掉电保持的，这个次数也会保持。当S7-200全清或者RAM区掉电保持失败时，该区也被清除。

Total Fatal区是前一次CPU清除所有存储区后产生致命错误的次数。如果RAM区是掉电保持的，这个次数也会保持。当S7-200全清或者RAM区掉电保持失败时，该区也被清除。

附录A列出了S7-200的错误代码，而附录B描述了可以用于监视错误的特殊存储区（SM）位。

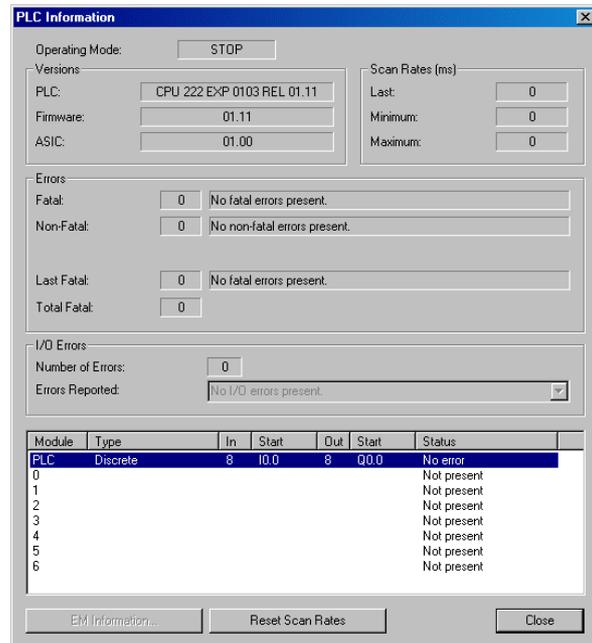


图5-6 PLC信息对话框

非致命错误

非致命错误是指用户程序结构问题、用户程序指令执行问题和扩展I/O模块问题。您可以用STEP 7-Micro/WIN来得到所产生错误的错误代码。非致命错误有三个基本的分类。

程序编译错误

当下载程序时，S7-200会编译程序。如果S7-200发现程序违反了编译规则，会停止下载并产生一个错误代码。（已经下载到S7-200中的程序将仍然在永久存储区中存在，并不会丢失。）可以在修正错误后再次下载程序。对于编译规则错误的列表，请参考附录A。

I/O错误

启动时，S7-200从每个模块读取I/O配置。正常运行过程中，S7-200周期性的检测每个模块的状态与启动时得到的配置相比较。如果S7-200检测到差别，它会将模块错误寄存器中的配置错误标志位置位。除非此模块的组态再次和启动时获得的组态相匹配，否则S7-200不会从此模块中读输入数据或者写输出数据到此模块。

模块的启动信息存储在特殊存储器（SM）标志位中。应用程序可以监视这些标志位。关于I/O错误报告的SM标志的更多信息，请参见附录B。SM5.0是全局I/O错误位，当扩展模块上存在一个错误条件时，它将保持置位。

程序执行错误

在程序执行过程中有可能产生错误。这类错误有可能来自使用了不正确的指令或者在过程中产生了非法数据。例如：一个编译正确的间接寻址指针，在程序执行过程中，可能会改为指向一个非法地址。这是一个实时程序问题的例子。当实时程序问题发生时，SM4.3会在CPU处于RUN模式期间置位。（实时程序问题的列表参见附录A）。程序执行错误信息存储在特殊寄存器（SM）标志位中。应用程序可以监视这些标志位。对于用于报告程序执行错误的SM位的更多信息，请参考附录B。

当S7-200发生非致命错误时，S7-200并不切换到STOP模式。它仅仅是把事件记录到SM存储器中并继续执行应用程序。但是如果用户希望在发生非致命错误时，将CPU切换到STOP模式，可以通过编程实现。下列例子程序用于监视两个非致命错误标志位。当两个标志中任意一个置位，S7-200将切换到STOP模式。

程序示例：用于检测一个非致命错误条件的逻辑	
	<p>Network 1 //当有I/O错误或实时运行程序错误发生时，将CPU //切换到STOP模式。</p> <pre>LD SM5.0 O SM4.3 STOP</pre>

致命错误

致命错误会导致S7-200停止程序执行。按照致命错误的严重程度，S7-200使其部分或全部功能无法执行。处理致命错误的目的是把CPU引向安全状态，CPU可以对存在的错误条件作出响应。当检测到一个致命错误时，S7-200将切换到STOP模式，打开SF/DIAG (Red) 和STOP LED，忽略输出表，并关闭输出。除非致命错误条件被修正，否则S7-200将保持这种状态不变。

一旦消除了致命错误条件，必须重新启动CPU。可以用以下方法重新启动CPU：

- 重新启动电源
- 将模式开关由RUN或者TERM变为STOP
- 在STEP 7-Micro/WIN命令菜单中选择**PLC > Power-Up Reset**，可以强制CPU启动并清除所有致命错误。

重启CPU会清除致命错误，并执行上电诊断测试来确认已改正错误。如果发现其它致命错误，CPU会重新点亮错误LED指示灯，表示仍存在错误。否则CPU会开始正常工作。

有些错误可能会使CPU无法进行通讯。这种情况下您无法看到来自CPU的错误代码。这种错误表示硬件故障，CPU模块需要修理，而修改程序或清除CPU内存是无法清除这些错误的。

在数据块中指定地址和初始值



数据块

数据块编辑器只用于为V存储器（变量存储器）指定初始值。您可以以字节、字或者双字的形式来分配V存储器，注释是可选的。

数据块编辑器是一个自由格式的文本编辑器，也就是说，没有特定的区域被定义用于特定类型的信息。当您完成一行的输入并按回车键确认后，数据块编辑器将该行格式化（将地址、数据和注释分别列对齐，V存储器地址大写）并重新显示。按下CTRL-ENTER，设置行完成后，地址将自动增加到下一个可用的地址处。

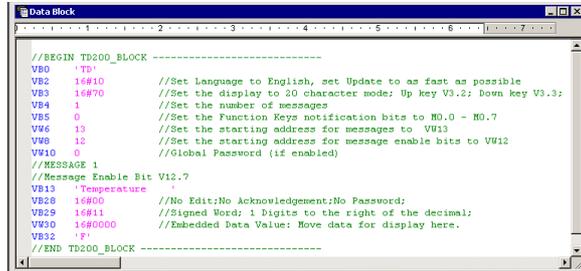


图5-7 数据块编辑器

数据块编辑器根据您所定义变量的地址和长度（字节、字或者双字）为V存储器分配空间。

数据块的第一行必须有一个明确的地址分配。接下来的行中可以是明确的地址，也可以使用隐含地址。隐含地址是由编辑器分配的。当您在一个地址后面输入多个数据或者在一行中只输入数据时，您使用的是隐含地址。

数据块编辑器接受大小写字母，并且用逗号、制表符或者空格作为地址与数据之间的分隔符。

用符号表来定义变量的符号地址



符号表

符号表允许您定义和编辑符号名，使您能在程序用符号地址访问变量。您可以创建多个符号表。您也可以使用系统定义的符号表。符号表还可以作为全局变量表进行参考。

您可以使用绝对地址或者符号地址来输入指令操作数。绝对地址用存储区加上位或字节地址来标识地址。符号地址则用一串字母组合来标识地址。

在SIMATIC程序中，您可以使用符号表中定义的全局符号。在IEC程序中，您可以使用全局变量表中定义的全局符号。

为地址定义符号按如下步骤：

	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 PRMs [mteger]
6			

图5-8 符号表

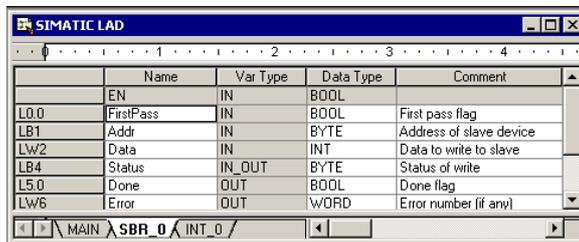
1. 在操作栏中单击符号表图标打开符号表。
2. 在Symbol Name列中输入一个符号名（例如：Input1），符号名的最大长度为23个字符。
3. 在Address列中输入地址（例如：I0.0）。
4. 对于IEC的全局变量表，在数据类型列中输入一个值或者从列表选择一个。

您可以创建多个符号表，但无论是在同一个符号表中还是在不同的符号表中，都不能多次使用同一个字符串作为全局符号。

使用局部变量

您可以使用程序编辑器中的局部变量表为子程序和中断服务程序分别指定变量，如图5-9所示。

局部变量可用于子程序传递参数，它增强了子程序的可移植性和再利用性。



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

用状态图来监视用户程序



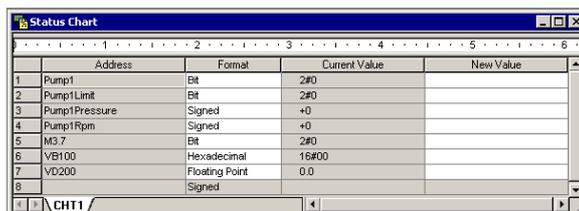
状态图

状态图允许您在控制程序运行的过程中对过程变量的值进行监视和修改。您可以跟踪程序的输入、输出或者变量，显示它们的当前值。状态表还允许您去强制或改变过程变量的值。

为了监控应用程序中不同部分的元素，您可以创建多个状态图。

在命令菜单中选择 **View > Component > Status Chart** 或者在操作过程中单击 Status Chart 图标来访问状态图。

当创建状态图时，您应该输入要监控的过程变量的地址。您无法监视常数、累加器和局部变量的状态。您可以按位或者字两种形式来显示定时器和计数器的值。以位形式显示的是定时器和计数器的状态位，而以字形式则显示定时器和计数器的当前值。



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

建立一个状态图来监视变量，按以下步骤：

1. 在地址区输入需要的地址。
2. 在格式列中选择数据类型。
3. 在命令菜单中选择 **Debug > Chart Status** 来监视 S7-200 中过程变量的状态。
4. 要连续采样数值或者单次读取状态，可以点击工具栏中相应的按钮。状态图也允许您强制或者修改过程变量的值。

在命令菜单中选择 **Edit > Insert > Row** 可以在状态图中插入一行。



提示

您可以按逻辑分组为变量创建多个状态图，使每个状态图更短，便于分别监视。

创建一个指令库

STEP 7-Micro/WIN允许您创建自己的指令库，也允许您使用其它人已建好的库。参见图5-11。

创建指令库的同时，您也在STEP 7-Micro/WIN中建立了子程序和中断服务程序。您可以在这些程序中隐藏代码来防止意外的改变或者防止机密技术（know-how）泄露。

要创建一个指令库，您需要完成以下任务：

1. 创建一个标准的STEP 7-Micro/WIN项目，并且把指令库中包括的功能写入一个子程序或者中断服务程序中。
2. 确保在子程序或中断服务程序中使用的V存储器都定义了符号名。使用连续的V存储区来最小化其要求的数量。
3. 将子程序或者中断服务程序更名为您希望在指令库中显示的名称。
4. 在命令菜单中选择**File > Create Library**来编译新的指令库。

在STEP 7-Micro/WIN的在线帮助中，您可以得到有关指令库的更多信息。

用下列步骤访问指令库中的指令：

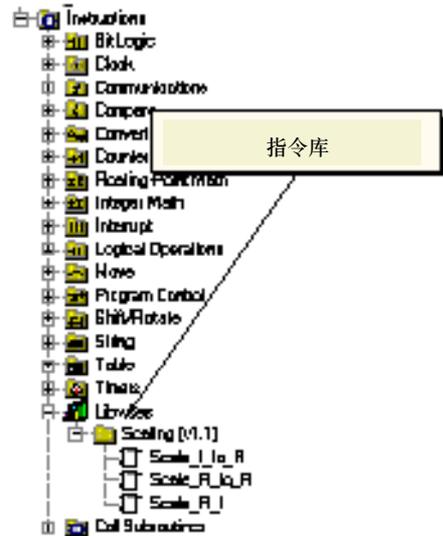


图5-11 带库的指令树

1. 在命令菜单中选择**File > Add Libraries**来在指令树中增加一个库。
2. 选择特定的指令插入您的程序。（就象使用任何标准指令一样。）

如果库程序要求V存储器，STEP 7-Micro/WIN会提示您分配一个存储器块。用库存储器对话框来分配存储器块。

应用程序的调试

STEP 7-Micro/WIN具备以下特点帮助您调试应用程序：

- 书签允许您在很长的程序中方便地来回移动。
- 交叉参考表允许您检查程序的使用参考信息。
- RUN模式下编辑允许您在小规模修改程序的过程中，对过程控制产生最小的影响。当在RUN模式下编辑程序时，您同样可以下载程序块。

关于调试应用程序的更多信息参见第8章。

6

S7-200指令集

本章描述用于对S7-200微型PLC编程的SIMATIC和IEC1131指令集。

本章内容:

用于描述指令的习惯用语	67
S7-200存储器范围及特性	68
位逻辑指令	70
触点	70
线圈	73
逻辑堆栈指令	75
RS触发器指令	77
时钟指令	78
通讯指令	81
网络读写指令	81
发送和接收指令	86
获取口地址和设定口地址指令	95
比较指令	96
数值比较	96
字符串比较	98
转换指令	99
标准转换指令	99
ASCII码转换指令	103
字符串转换指令	107
编码和解码指令	112
计数器指令	113
SIMATIC计数器指令	113
IEC计数器指令	116
高速计数器指令	118
脉冲输出指令	133
数字运算指令	140
加、减、乘、除指令	140
整数乘法产生双整数和带余数的整数除法	142
数学功能指令	143
递增和递减指令	144
比例/积分/微分 (PID) 回路控制指令	145
中断指令	153
逻辑操作指令	161
取反指令	161
与、或和异或指令	162
传送指令	164
字节、字、双字或者实数传送	164
字节立即传送 (读和写)	165
块传送指令	166

程序控制指令	167
条件结束	167
停止	167
看门狗复位	167
For - Next循环指令	169
跳转指令	171
顺控继电器 (SCR) 指令	172
诊断LED指令	178
移位和循环指令	179
右移和左移指令	179
循环右移和循环左移指令	179
移位寄存器指令	181
字节交换指令	183
字符串指令	184
表指令	189
填表	189
先进先出和后进先出	190
内存填充	192
查表	193
定时器指令	196
SIMATIC定时器指令	196
IEC定时器指令	201
时间间隔定时器	203
子程序指令	204

用于描述指令的习惯用语

图6-1给出了对一条指令的典型描述，并指出了用于描述指令及其操作的不同区域。指令说明包括LAD、FBD和STL三种格式。操作数表列出了指令的操作数，并给出有效数据类型、存储器区和每个操作数的大小。

EN/ENO操作数和数据类型没有在指令操作数表中列出，因为这些操作数对于所有的LAD和FBD指令来说都是一样的。

- ❑ 对于LAD：EN和ENO是能流，为布尔数据类型。
- ❑ 对于FBD：EN和ENO是I、Q、V、M、SM、S、T、C、L或者能流，为布尔数据类型。

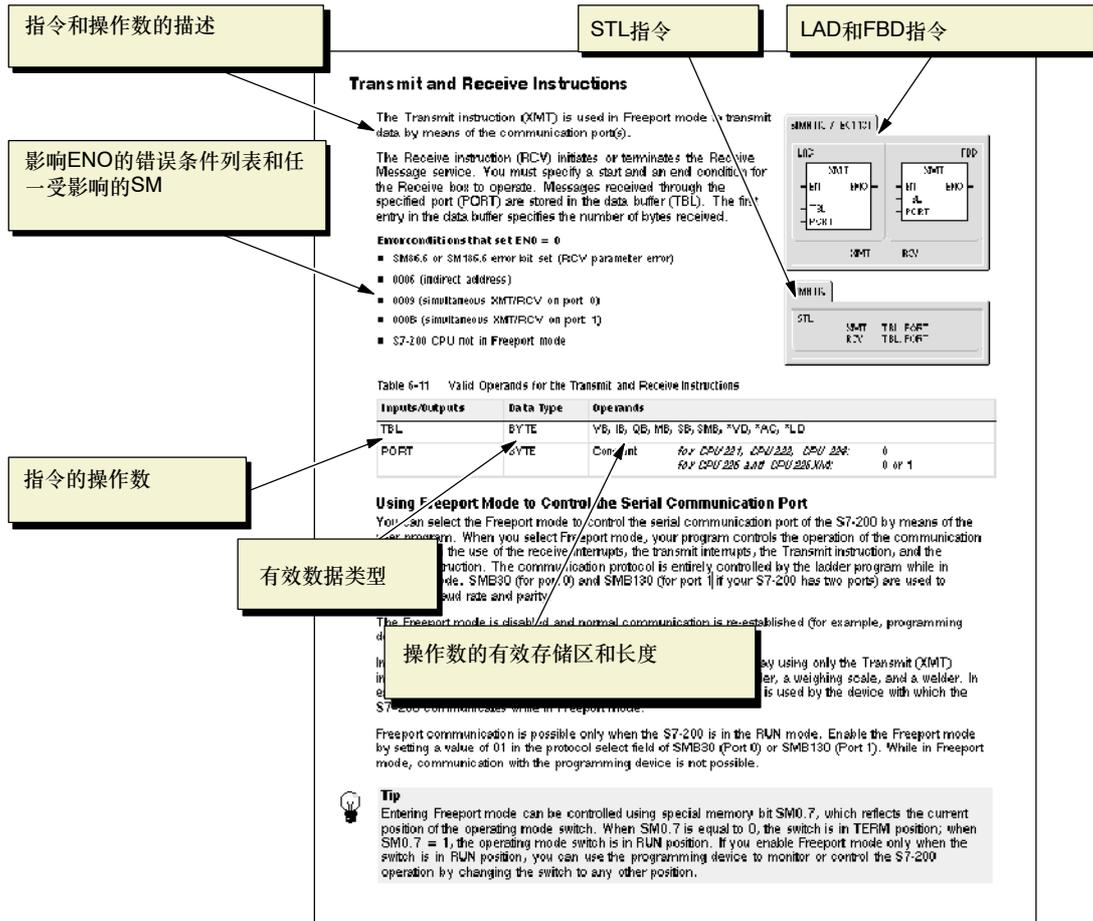


图6-1 指令描述

S7-200存储器范围及特性

表6-1 S7-200 CPU存储器范围及特性

中断描述	CPU221	CPU222	CPU224	CPU224XP	CPU226
用户程序大小 带运行模式下编辑 不带运行模式下编辑	4096字节 4096字节	4096字节 4096字节	8192字节 12288字节	12288字节 16384字节	16384字节 24576字节
用户数据大小	2048字节	2048字节	8192字节	10240字节	10240字节
输入映像寄存器	I0.0 - I15.7	I0.0 - I15.7	I0.0 - I15.7	I0.0 - I15.7	I0.0 - I15.7
输出映像寄存器	Q0.0 - Q15.7	Q0.0 - Q15.7	Q0.0 - Q15.7	Q0.0 - Q15.7	Q0.0 - Q15.7
模拟量输入（只读）	AIW0 - AIW30	AIW0 - AIW30	AIW0 - AIW62	AIW0 - AIW62	AIW0 - AIW62
模拟量输出（只写）	AQW0 - AQW30	AQW0 - AQW30	AQW0 - AQW62	AQW0 - AQW62	AQW0 - AQW62
变量存储器（V）	VB0 - VB2047	VB0 - VB2047	VB0 - VB8191	VB0 - VB10239	VB0 - VB10239
局部存储器（L） ¹	LB0 - LB63	LB0 - LB63	LB0 - LB63	LB0 - LB63	LB0 - LB63
位存储器（M）	M0.0 - M31.7	M0.0 - M31.7	M0.0 - M31.7	M0.0 - M31.7	M0.0 - M31.7
特殊存储器（SM） 只读	SM0.0 - SM179.7 SM0.0 - SM29.7	SM0.0 - SM299.7 SM0.0 - SM29.7	SM0.0 - SM549.7 SM0.0 - SM29.7	SM0.0 - SM549.7 SM0.0 - SM29.7	SM0.0 - SM549.7 SM0.0 - SM29.7
定时器	256 (T0 - T255)	256 (T0 - T255)	256 (T0 - T255)	256 (T0 - T255)	256 (T0 - T255)
有记忆接通延迟	1ms 10ms 100ms	T0,T64 T1 - T4, T65 - T68 T5 - T31, T69 - T95	T0,T64 T1 - T4, T65 - T68 T5 - T31, T69 - T95	T0,T64 T1 - T4, T65 - T68 T5 - T31, T69 - T95	T0,T64 T1 - T4, T65 - T68 T5 - T31, T69 - T95
接通/关断延迟	1ms 10ms 100ms	T32,T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255	T32,T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255	T32,T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255	T32,T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255
计数器	C0 - C255	C0 - C255	C0 - C255	C0 - C255	C0 - C255
高速计数器	HC0 - HC5	HC0 - HC5	HC0 - HC5	HC0 - HC5	HC0 - HC5
顺序控制继电器（S）	S0.0 - S31.7	S0.0 - S31.7	S0.0 - S31.7	S0.0 - S31.7	S0.0 - S31.7
累加寄存器	AC0 - AC3	AC0 - AC3	AC0 - AC3	AC0 - AC3	AC0 - AC3
跳转/标号	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255
调用/子程序	0 - 63	0 - 63	0 - 63	0 - 63	0 - 127
中断程序	0 - 127	0 - 127	0 - 127	0 - 127	0 - 127
正/负跳变	256	256	256	256	256
PID回路	0 - 7	0 - 7	0 - 7	0 - 7	0 - 7
端口	端口0	端口0	端口0	端口0, 1	端口0, 1

¹ LB60-LB63为STEP 7-Micro/WIN32的3.0版本或以后的版本软件保留

表6-2 S7-200 CPU的操作数范围

存取方式		CPU221	CPU222	CPU224	CPU224XP	CPU226
位存取 (字节.位)		0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7
	Q	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7	0.0 - 15.7
	V	0.0 - 2047.7	0.0 - 2047.7	0.0 - 8191.7	0.0 - 10239.7	0.0 - 10239.7
	M	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7
	SM	0.0 - 165.7	0.0 - 299.7	0.0 - 549.7	0.0 - 549.7	0.0 - 549.7
	S	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7
	T	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255
	C	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255
	L	0.0 - 63.7	0.0 - 63.7	0.0 - 63.7	0.0 - 63.7	0.0 - 63.7
字节存取	IB	0 - 15	0 - 15	0 - 15	0 - 15	0 - 15
	QB	0 - 15	0 - 15	0 - 15	0 - 15	0 - 15
	VB	0 - 2047	0 - 2047	0 - 8191	0 - 10239	0 - 10239
	MB	0 - 31	0 - 31	0 - 31	0 - 31	0 - 31
	SMB	0 - 165	0 - 299	0 - 549	0 - 549	0 - 549
	SB	0 - 31	0 - 31	0 - 31	0 - 31	0 - 31
	LB	0 - 63	0 - 63	0 - 63	0 - 63	0 - 63
	AC	0 - 3	0 - 3	0 - 3	0 - 255	0 - 255
	KB (常数)	KB (常数)	KB (常数)	KB (常数)	KB (常数)	KB (常数)
字存取	IW	0 - 14	0 - 14	0 - 14	0 - 14	0 - 14
	QW	0 - 14	0 - 14	0 - 14	0 - 14	0 - 14
	VW	0 - 2046	0 - 2046	0 - 8190	0 - 10238	0 - 10238
	MW	0 - 30	0 - 30	0 - 30	0 - 30	0 - 30
	SMW	0 - 164	0 - 298	0 - 548	0 - 548	0 - 548
	SW	0 - 30	0 - 30	0 - 30	0 - 30	0 - 30
	T	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255
	C	0 - 255	0 - 255	0 - 255	0 - 255	0 - 255
	LW	0 - 62	0 - 62	0 - 62	0 - 62	0 - 62
	AC	0 - 3	0 - 3	0 - 3	0 - 3	0 - 3
	AIW	0 - 30	0 - 30	0 - 62	0 - 62	0 - 62
	AQW	0 - 30	0 - 30	0 - 62	0 - 62	0 - 62
	KB (常数)	KB (常数)	KB (常数)	KB (常数)	KB (常数)	KB (常数)
双字存取	ID	0 - 12	0 - 12	0 - 12	0 - 12	0 - 12
	QD	0 - 12	0 - 12	0 - 12	0 - 12	0 - 12
	VD	0 - 2044	0 - 2044	0 - 8188	0 - 10236	0 - 10236
	MD	0 - 28	0 - 28	0 - 28	0 - 28	0 - 28
	SMD	0 - 162	0 - 296	0 - 546	0 - 546	0 - 546
	SD	0 - 28	0 - 28	0 - 28	0 - 28	0 - 28
	LD	0 - 60	0 - 60	0 - 60	0 - 60	0 - 60
	AC	0 - 3	0 - 3	0 - 3	0 - 3	0 - 3
	HC	0 - 5	0 - 5	0 - 5	0 - 5	0 - 5
	KD (常数)	KD (常数)	KD (常数)	KD (常数)	KD (常数)	KD (常数)

位逻辑指令

触点

标准触点

常开触点指令（LD、A和O）与常闭触点指令（LDN、AN和ON）从存储器或者过程映像寄存器中得到参考值。标准触点指令从存储器中得到参考值。（如果数据类型是I或Q，则从过程映像寄存器中得到参考值。）

当位值为1时，常开触点闭合；当位值为0时，常闭触点闭合。在FBD中，与和或操作的输入可以最多扩展到32个。在STL中，常开指令LD、AND或OR将相应地址位的位值存入栈顶；而常闭指令 LD、AND或OR则将相应地址位的位值取反，再存入栈顶。

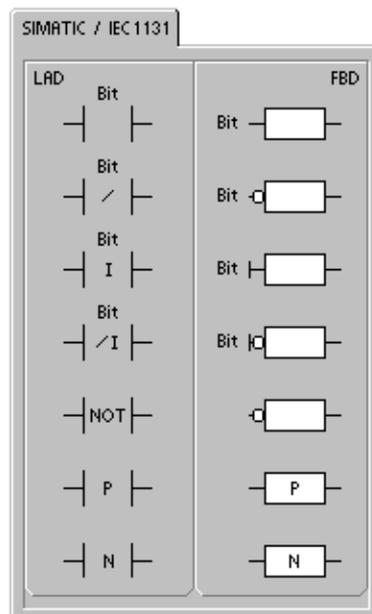
立即触点

立即触点并不依赖于S7-200的扫描周期刷新，它会立即刷新。常开立即触点指令（LDI、AI和OI）和常闭立即触点指令（LDNI、ANI和ONI）在指令执行时得到物理输入值，但过程映像寄存器并不刷新。

当物理输入点状态为1时，常开立即触点闭合；当物理输入点状态为0时，常闭立即触点闭合。常开立即指令LDI、AI或OI将物理输入值存入栈顶，而常闭立即指令LDNI、ANI或ONI将物理输入的值取反，再存入栈顶。

取反指令

取反指令（NOT）改变能流输入的状态（也就是说，它将栈顶值由0变为1，由1变为0）。



SIMATIC			
STL			
LD	Bit	LDI	Bit
A	Bit	AI	Bit
O	Bit	OI	Bit
LDN	Bit	LDNI	Bit
AN	Bit	ANI	Bit
ON	Bit	ONI	Bit
NOT			
EU			
ED			

正、负跳变指令

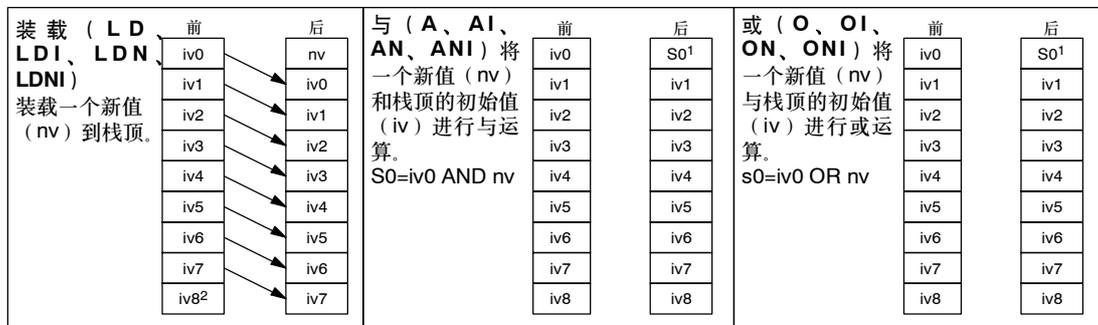
正跳变触点指令（EU）检测到每一次正跳变（由0到1），让能流接通一个扫描周期。负跳变触点指令（ED）检测到每一次负跳变（由1到0），让能流接通一个扫描周期。对于正跳变指令，一旦发现有正跳变发生（由0到1），该栈顶值被置为1，否则置0。对于负跳变指令，一旦发现有负跳变发生（由1到0），该栈顶值被置为1，否则置0。

对于运行模式下编辑（在RUN模式下编辑应用程序），您必须为正跳变指令和负跳变指令输入参数。关于在RUN模式下编辑程序的更多信息参见第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、能流
位（立即）	BOOL	I

如图6-2中所示，S7-200用逻辑堆栈来决定控制逻辑。在本例中，“iv0”到“iv7”表示逻辑堆栈的初始值，“nv”表示指令提供的一个新值，S0表示逻辑堆栈中存储的计算值。



- ¹ S0表示存储在逻辑栈中的计算值。
- ² 在装载指令执行之后，值iv8丢失。

图6-2 触点指令的操作。



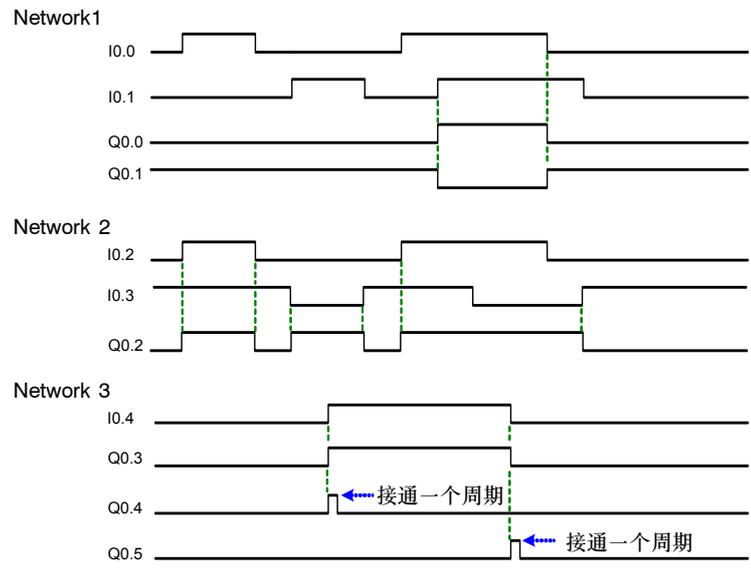
提示

由于正跳变指令和负跳变指令要求由1到0或者由0到1的变化，您不能在第一个扫描周期中检测到上升沿或者下降沿的变化。在第一个扫描周期，S7-200利用这些指令储存指定位的状态。在接下来的扫描周期中，这些指令能够检测到指定位的变化。

示例: 触点指令

<p>Network 1</p> <p>Network 2</p> <p>Network 3</p>	<p>Network 1 //要想激活Q0.0, 常开触点I0.0 //和I0.1必须为接通(闭合)。NOT //指令作为一个反向器使用。在RUN //模式下, Q0.0和Q0.1具有相反的逻辑状态。</p> <pre>LD I0.0 A I0.1 = Q0.0 NOT = Q0.1</pre> <p>Network 2 //要想激活Q0.2, 常开触点I0.2必须为on或者常闭 //触点I0.3必须为off。 //要想激活输出, 并行LAD分支 //(或逻辑输入)中应该有一个或 //多个逻辑值为真。</p> <pre>LD I0.2 ON I0.3 = Q0.2</pre> <p>Network 3 //在P触点的一个上升沿或者在N触点的一个下降 //沿出现时, 一个扫描周期内输出一个脉冲。 //在RUN模式, Q0.4和Q0.5的脉冲状态变化太快 //以至于在程序中无法用状态图监视。 //置位和复位指令将 //Q0.3的状态变化锁存, //使程序可以监视。</p> <pre>LD I0.4 LPS EU S Q0.3, 1 = Q0.4 LPP ED R Q0.3, 1 = Q0.5</pre>
---	--

时序图



线圈

输出

输出指令(=)将新值写入输出点的过程映像寄存器。当输出指令执行时，S7-200将输出过程映像寄存器中的位接通或者断开。在LAD和FBD中，指定点的值等于能流。在STL中，栈顶的值复制到指定位。

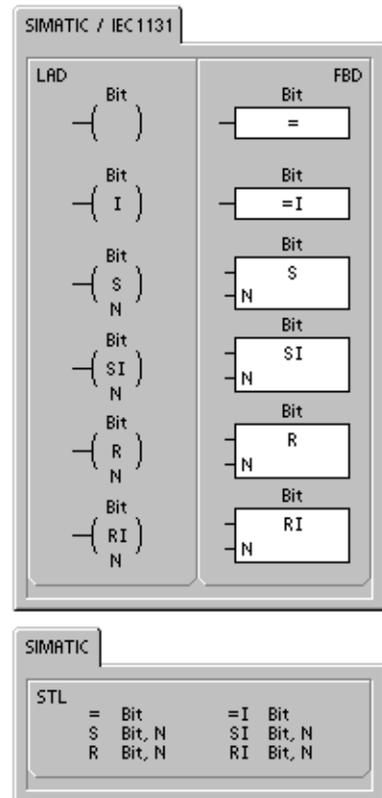
立即输出

当指令执行时，立即输出指令(=I)将新值同时写到物理输出点和相应的过程映像寄存器中。

当立即输出指令执行时，物理输出点立即被置为能流值。在STL中，立即指令将栈顶的值立即复制到物理输出点的指定位上。“I”表示立即，当指令执行时，新值会同时被写到物理输出和相应的过程映像寄存器。这一点不同于非立即指令，只把新值写入过程映像寄存器。

置位和复位

置位(S)和复位(R)指令将从指定地址开始的N个点置位或者复位。您可以一次置位或者复位1-255个点。



如果复位指令指定的是一个定时器位(T)或计数器位(C)，指令不但复位定时器或计数器位，而且清除定时器或计数器的当前值。

使ENO=0的错误条件:

- 0006 (间接寻址)
- 0091 (操作数超出范围)

立即置位和立即复位

立即置位和立即复位指令将从指定地址开始的N个点立即置位或者立即复位。您可以一次置位或复位1到128个点。

“I”表示立即，当指令执行时，新值会同时被写到物理输出和相应的过程映像寄存器。这一点不同于非立即指令，只把新值写入过程映像寄存器。

使ENO=0的错误条件:

- 0006 (间接寻址)
- 0091 (操作数超出范围)

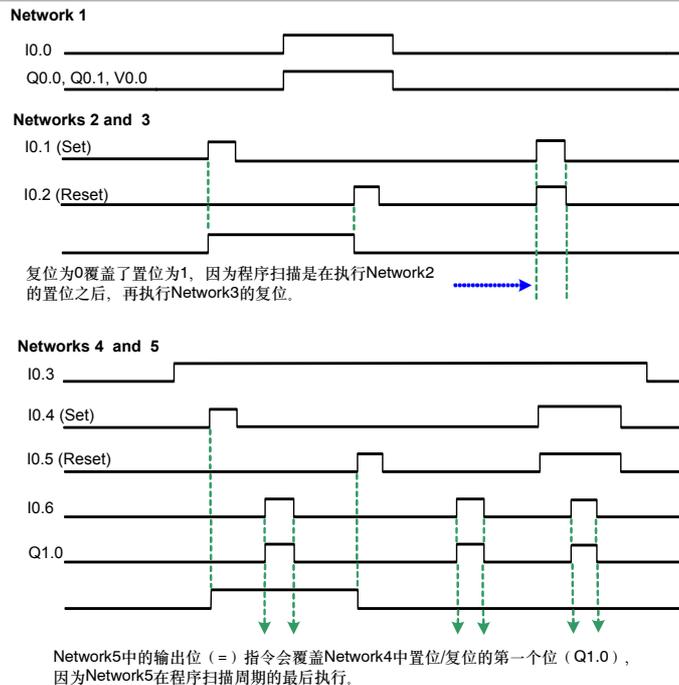
表6-4 位逻辑输出指令的有效操作数

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、常数

示例: 线圈指令

<p>Network 1</p> <p>Network 2</p> <p>Network 3</p> <p>Network 4</p> <p>Network 5</p>	<p>Network1 //输出指令为外部I/O (I、Q) 和内部存储 //器 (M、SM、T、C、V、S、L) 指定位值。</p> <pre>LD I0.0 = Q0.0 = Q0.1 = V0.0</pre> <p>Network2 //连续将一组6位置为1。 //指定起始地址和置位的个数。当第一位 // (Q0.2) 的值为1时, 置位指令 //的程序状态指示器为ON。</p> <pre>LD I0.1 S Q0.2, 6</pre> <p>Network3 //连续将一组6位置为0。 //指定起始地址和复位的个数。 //当第一位 (Q0.2) 的值为0时, 复位指 //令的程序状态指示器为ON。</p> <pre>LD I0.2 R Q0.2, 6</pre> <p>Network4 //置位和复位一组8个输出位 (Q1.0~Q1.7)。</p> <pre>LD I0.3 LPS A I0.4 S Q1.0, 8 LPP A I0.5 R Q1.0, 8</pre> <p>Network5 //置位和复位指令实现锁存器功能。 //完成置位/复位功能, 必须确保这些 //位没有在其他指令中被改写。在本例中, //Network4置位和复位一组 //8个输出位 (Q1.0~Q1.7)。在RUN模式 //下Network5会覆盖Q1.0的值, 从而 //控制Network4中的程序状态显示器。</p> <pre>LD I0.6 = Q1.0</pre>
---	--

时序图



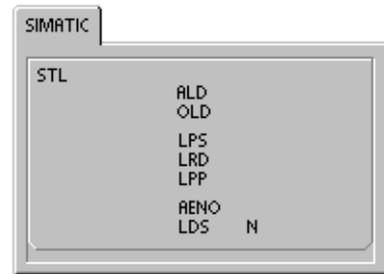
逻辑堆栈指令

栈装载与

栈装载与指令对堆栈中第一层和第二层的值进行逻辑与操作，结果放入栈顶。执行完栈装载与指令之后，栈深度减1。

栈装载或

栈装载或指令对堆栈中第一层和第二层的值进行逻辑或操作，结果放入栈顶。执行完栈装载或指令之后，栈深度减1。



逻辑推入栈

逻辑推入栈指令复制栈顶的值，并将这个值推入栈。栈底的值被推出并消失。

逻辑读栈

逻辑读栈指令复制堆栈中的第二个值到栈顶。堆栈没有推入栈或者弹出栈操作，但旧的栈顶值被新的复制值取代。

逻辑弹出栈

逻辑弹出栈指令弹出栈顶的值，堆栈的第二个栈值成为新的栈顶值。

ENO与

ENO与指令对ENO位和栈顶的值进行逻辑与操作，其产生的效果与LAD或者FBD中盒指令的ENO位相同。与操作结果成为新的栈顶。

ENO是LAD和FBD中盒指令的布尔输出。如果盒指令的EN输入有能流并且执行没有错误，则ENO将能流传递给下一元素。您可以把ENO作为指令成功完成的使能标志位。ENO位被用作栈顶，影响能流和后续指令的执行。STL中没有EN输入。条件指令要想执行，栈顶值必须为逻辑1。在STL中也没有ENO输出。但是在STL中，那些与LAD和FBD中具有ENO输出的指令相应的指令，存在一个特殊的ENO位。它可以被AENO指令访问。

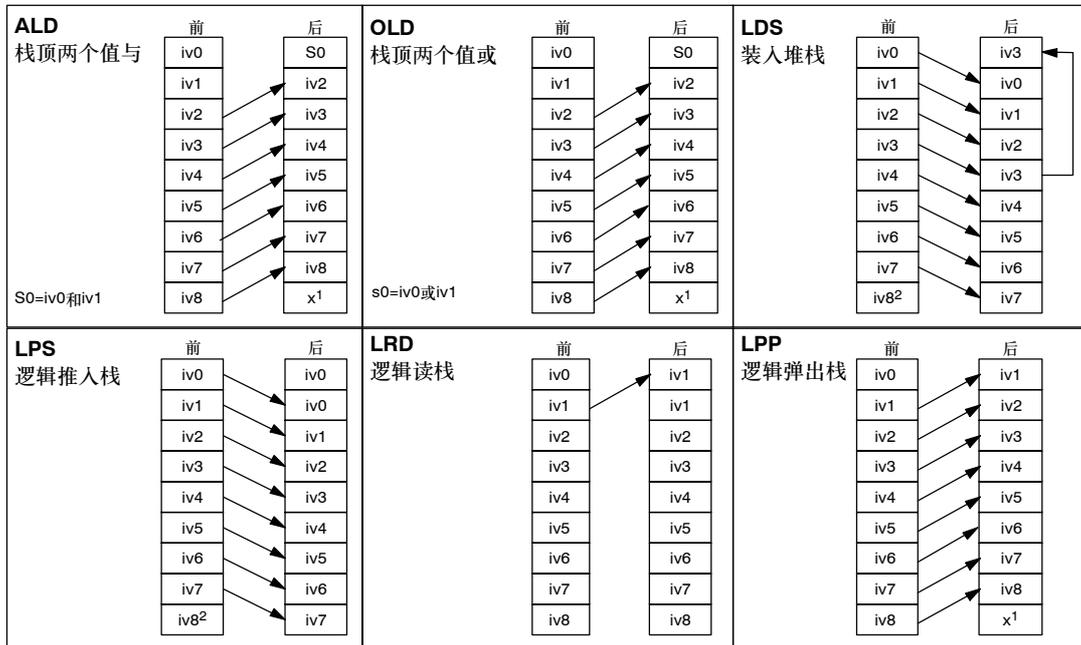
装入堆栈

装入堆栈指令复制堆栈中的第N个值到栈顶。栈底的值被推出并消失。

表6-5 装入堆栈指令的有效操作数

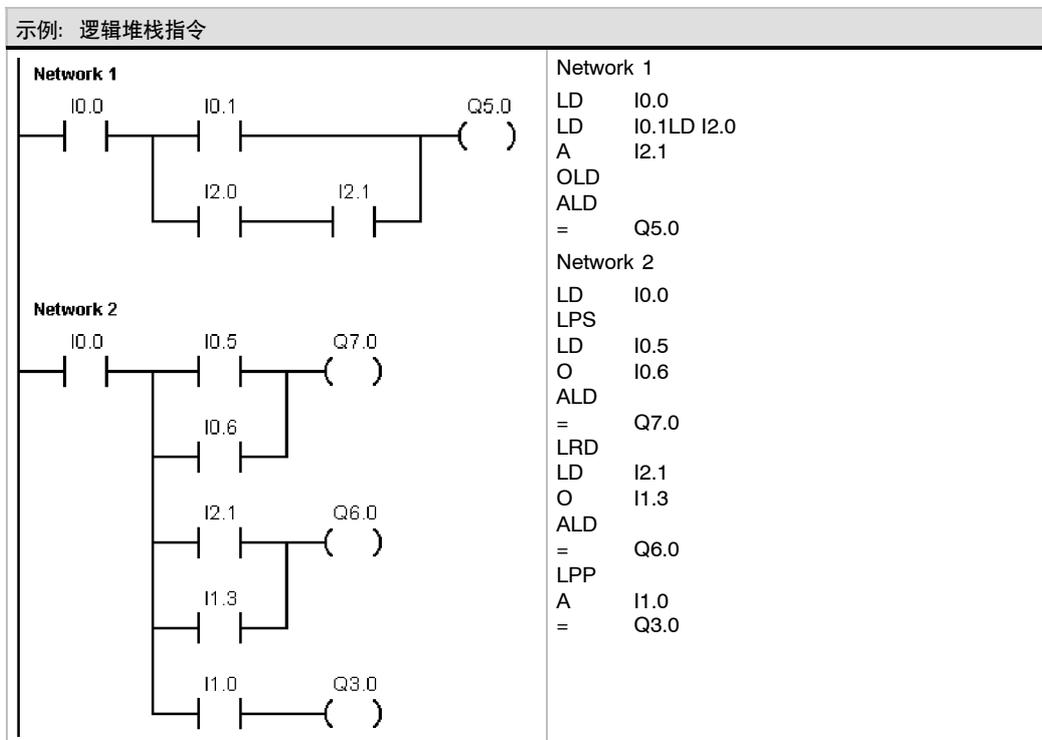
输入/输出	数据类型	操作数
N	BYTE	常数(0到8)

如图6-3中所示，S7-200用逻辑堆栈来决定控制逻辑。在本例中，“iv0”到“iv7”表示逻辑堆栈的初始值，“nv”表示指令提供的一个新值，而“S0”表示逻辑堆栈中存储的计算值。



- 1 数值是不确定的（可以是0，也可以是1）
- 2 在逻辑入栈或者装入堆栈指令执行后，iv8的值丢失。

图6-3 逻辑堆栈指令的操作



RS触发器指令

置位优先触发器是一个置位优先的锁存器。当置位信号(S1)和复位信号(R)都为真时，输出为真。

复位优先触发器是一个复位优先的锁存器。当置位信号(S)和复位信号(R1)都为真时，输出为假。

Bit参数用于指定被置位或者复位的布尔参数。可选的输出反映Bit参数的信号状态。

表6-7中给出了例子程序的真值表。

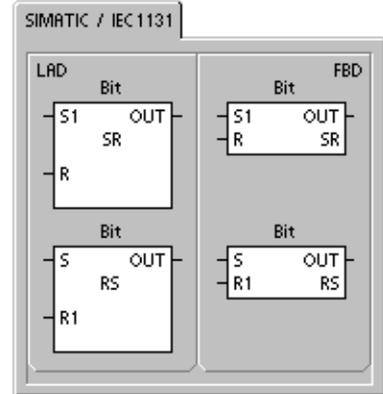


表6-6 RS触发器指令的有效操作数

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

示例: RS触发器指令

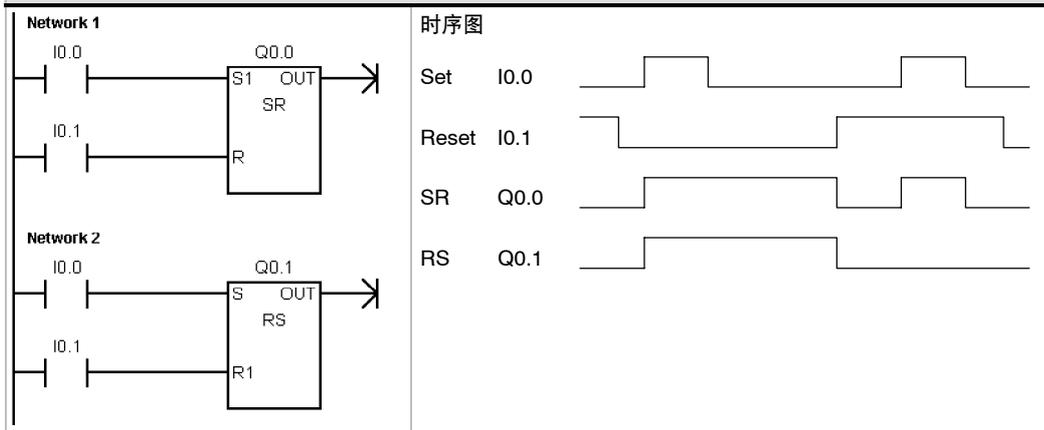


表6-7 RS触发器指令真值表

指令	S1	R	输出 (Bit)
置位优先触发器指令 (SR)	0	0	保持前一状态
	0	1	0
	1	0	1
	1	1	1
指令	S	R1	输出 (Bit)
复位优先触发器指令 (RS)	0	0	保持前一状态
	0	1	0
	1	0	1
	1	1	0

时钟指令

读实时时钟和写实时时钟

读实时时钟（TODR）指令从硬件时钟中读当前时间和日期，并把它装载到一个8字节，起始地址为T的时间缓冲区中。写实时时钟（TODW）指令将当前时间和日期写入硬件时钟，当前时钟存储在以地址T开始的8字节时间缓冲区中。

您必须按照BCD码的格式编码所有的日期和时间值（例如：用16 # 97表示1997年）。图6-4给出了时间缓冲区（T）的格式。

时间日期（TOD）时钟在电源掉电或内存丢失后，初始化为下列日期和时间：

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

使ENO=0的错误条件:

- 0006（间接寻址）
- 0007（TOD数据错误），只对写实时时钟指令有效。
- 000C（时钟模块不存在）

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、*VD、*LD、*AC

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

*T+7 1=星期日, 7=星期六

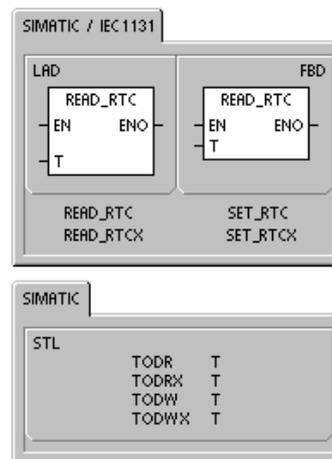


图6-4 8字节时间缓冲区的格式

**提示**

S7-200 CPU不会检查和核实日期与星期是否合理。无效日期February 30（2月30日）可能被接受。故必须确保输入的数据是正确的。

不要同时在主程序和中断程序中使用TODR/TODW指令。如果这样做，而在执行TOD指令时出现了执行TOD指令的中断，则中断程序中的TOD指令不会被执行。SM4.3指示了试图对时钟进行两个同时的访问（非致命错误0007）。

S7-200 PLC只使用年信息的后两位，不会受到世纪跨越的影响。但是，用到年份进行计算或比较的用户程序必须考虑两位的表示方法和世纪的变化。

在2096年之前可以进行闰年的正确处理。

扩展读实时时钟

扩展读实时时钟（TODRX）指令从PLC中读取当前时间、日期和夏令时组态，并装载到从由T指定的地址开始的19字节缓冲区内。

使ENO=0的错误条件:

- 0006（间接寻址）
- 000C（时钟卡不存在）
- 0091（操作数超出范围）

扩展写实时时钟

扩展写实时时钟（TODWX）指令写当前时间、日期和夏令时组态到PLC中由T指定的地址开始的19字节缓冲区内。

使ENO=0的错误条件:

- 0006（间接寻址）
- 0007（TOD数据错误）
- 000C（时钟卡不存在）
- 0091（操作数超出范围）

您必须按照BCD码的格式编码所有的日期和时间值（例如：用16 # 02表示2002年）。表6-9给出了19字节时间缓冲区（T）的格式。

时间日期时钟在电源掉电或内存丢失后，初始化下列日期和时间:

日期:	01-Jan-90
时间:	00:00:00
星期:	星期日

表6-9 19字节时间缓冲区 (T1) 的格式

T字节	中断描述	字节数据
0	年 (0 - 99)	当前年份 (BCD值)
1	月份 (1 - 12)	当前月份 (BCD值)
2	日期 (1 - 31)	当前日期 (BCD值)
3	小时 (0 - 23)	当前小时 (BCD值)
4	分钟 (0 - 59)	当前分钟 (BCD值)
5	秒 (0 - 59)	当前秒 (BCD值)
6	00	保留 - 一直为00
7	星期 (1 - 7)	当前是星期几, 1=Sunday (BCD值)
8	模式 (00H - 03H, 08H, 10H - 13H, FFH)	修正模式: 00H=禁止修改 01H=EU (与UTC的时差=0小时) ¹ 02H=EU (与UTC的时差=+1小时) ¹ 03H=EU (与UTC的时差=+2小时) ¹ 04H - 07H=保留 08H=EU (与UTC的时差=-1小时) ¹ 09H - 0FH=保留 10H=US ² 11H=澳大利亚 ³ 12H=澳大利亚 (塔斯马尼亚岛) ⁴ 13H=新西兰 ⁵ 14H - FEH=保留 FFH=用户指定 (使用字节9 - 18中的值)
9	小时修正 (0 - 23)	修正量, 小时 (BCD值)
10	分钟修正 (0 - 59)	修正量, 分钟 (BCD值)
11	开始月份 (1 - 12)	夏令时的开始月份 (BCD值)
12	开始日期 (1 - 31)	夏令时的开始日期 (BCD值)
13	开始小时 (0 - 23)	夏令时的开始小时 (BCD值)
14	开始分钟 (0 - 59)	夏令时的开始分钟 (BCD值)
15	结束月份 (1 - 12)	夏令时的结束月份 (BCD值)
16	结束日期 (1 - 31)	夏令时的结束日期 (BCD值)
17	结束小时 (0 - 23)	夏令时的结束小时 (BCD值)
18	结束分钟 (0 - 59)	夏令时的结束分钟 (BCD值)

- ¹ 欧洲惯例: 在UTC时间三月份的最后一个星期日的上午1:00向前调整时间一个小时。在UTC时间十月份的最后一个星期日的上午2:00向后调整时间一个小时。(当进行修正时, 当地时间依据于与UTC的时差。)
- ² 美洲惯例: 在当地时间四月份的第一个星期日的上午2:00向前调整时间一个小时。在当地时间十月份的最后一个星期日的上午2:00向后调整时间一个小时。
- ³ 澳洲惯例: 在当地时间十月份的最后一个星期日的上午2:00向前调整时间一个小时。在当地时间三月份的最后一个星期日的上午3:00向后调整时间一个小时。
- ⁴ 澳洲 (塔斯马尼亚岛) 惯例: 在当地时间十月份的第一个星期日的上午2:00向前调整时间一个小时。在当地时间三月份的最后一个星期日的上午3:00向后调整时间一个小时。
- ⁵ 新西兰惯例: 在当地时间十月份的第一个星期日的上午2:00向前调整时间一个小时。在当地时间三月份的第一个星期日或三月15号以后的上午3:00向后调整时间一个小时。

通讯指令

网络读写指令

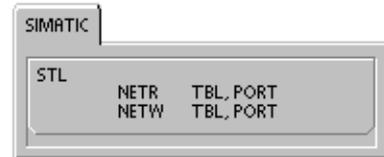
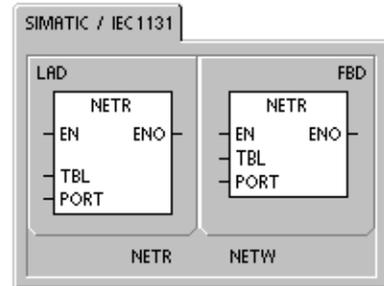
网络读指令（NETR）初始化一个通讯操作，通过指定端口（PORT）从远程设备上采集数据并形成表（TBL）。网络写指令（NETW）初始化一个通讯操作，通过指定端口（PORT）向远程设备写表（TBL）中的数据。

使ENO=0的错误条件:

- 0006（间接寻址）
- 如果功能返回出错信息，会置位表状态字节中的E。（见图6-5）

网络读指令可以从远程站点读取最多16个字节的信息，网络写指令可以向远程站点写最多16个字节的信息。

在程序中，您可以使用任意条网络读写指令，但是在同一时间，最多只能有8条网络读写指令被激活。例如，在所给的S7-200 CPU中，可以有4条网络读指令和4条网络写指令，或者2条网络读指令和6条网络写指令在同一时间被激活。



指令向导

您可以使用网络读写向导程序。要启动网络读写向导程序，在命令菜单中选择**Tools > Instruction Wizard**，并且在指令向导窗口中选择网络读写。

表6-10 网络读写指令的有效操作数

输入/输出	数据类型	操作数
TBL	BYTE	VB、MB、*VD、*LD、*AC
PORT	BYTE	常数 对于CPU221、CPU222、CPU224: 0 对于CPU224XP和CPU226: 0或1

图6-5中给出了TBL参数参照表，表6-11列出了错误代码。

字节 偏移量	7	0			错误代码	
0	D	A	E	0		
1	远程站地址				远程站地址:	被访问的PLC的地址
2	远程站的					
3	数据区				远程站的数据区指针:	被访问数据的间接指针
4	指针					
5	(I、Q、M、或V)				数据长度:	远程站上被访问数据的字节数
6	数据长度					
7	数据字节0				接收和发送数据区: 如下描述的保存数据的1到16个字节 对NETR, 执行NETR指令后, 从远程站读到的数据放在这个数据区 对NETW, 执行NETW指令前, 要发送到远程站的数据放在这个数据区	
8	数据字节1					
	;					
	;					
22	数据字节15					

图6-5 网络读写指令的TBL参数

表6-11 TBL参数的错误代码

错误代码	定义
0	无错误
1	时间溢出错误, 远程站点不响应
2	接收错: 奇偶校验错, 响应时帧或校验和出错
3	离线错: 相同的站地址或无效的硬件引发冲突
4	队列溢出错误: 激活了超过8个NETR/NETW方框
5	违反通信协议: 没有在SMB30中允许PPI, 就试图执行NETR/NETW指令
6	非法参数: NETR/NETW表中包含非法或无效的值
7	没有资源: 远程站点正在忙中(上装或下装程序在处理中)
8	第7层错误: 违反应用协议
9	信息错误: 错误的数据地址或不正确的数据长度
A-F	未用: (为将来的使用保留)

图6-6给出了一个实例来解释网络读写指令的使用。本例中, 考虑一条生产线正在灌装黄油桶并将其送到四台包装机(打包机)中的一台上。打包机把8个黄油桶包装到一个纸板箱中。一个分流机控制着黄油桶流向各个打包机。4个CPU221模块用于控制打包机, 一个CPU222模块安装了TD200操作器接口, 被用来控制分流机。

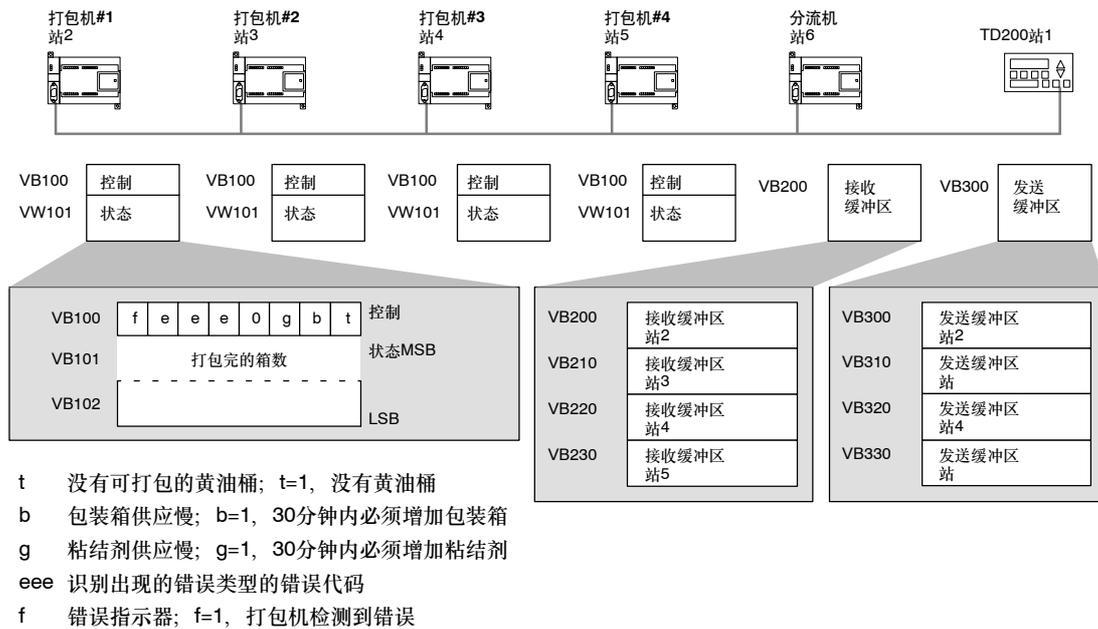


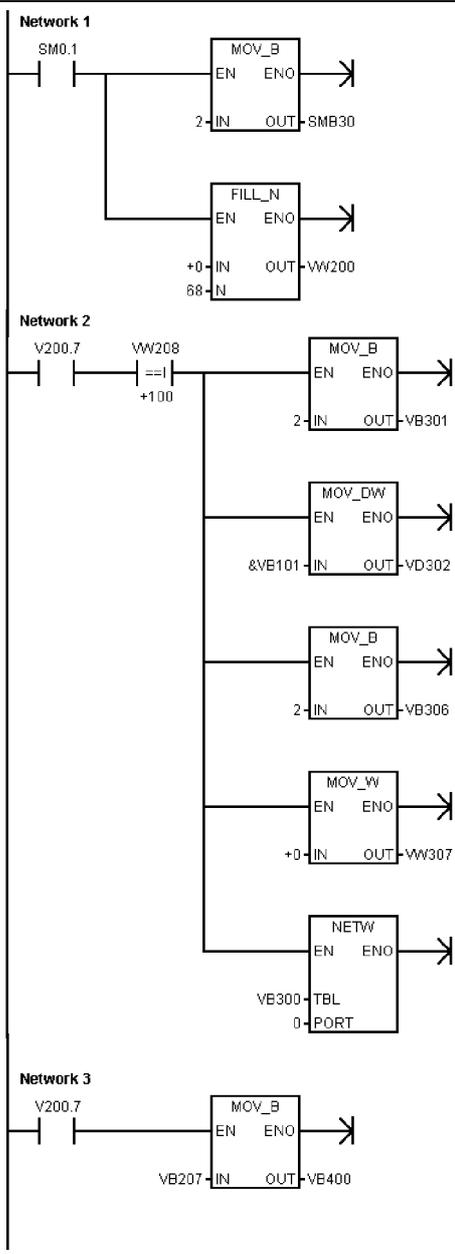
图6-6 网络读写指令举例

图6-7中给出了2号站中接收缓冲区（VB200）和发送缓冲区（VB300）中的数据。S7-200使用网络读指令不断地读取每个打包机的控制和状态信息。每次某个打包机包装完100箱，分流机会注意到，并用网络写指令发送一条信息清除状态字。



图6-7 网络读写指令中TBL数据举例

示例: 网络读写指令



Network1 //在第一个扫描周期,
//使能PPI主站模式, 并且
//清除所有接收和发送缓冲区。

```
LD SM0.1
MOVB 2, SMB30
FILL +0, VW200, 68
```

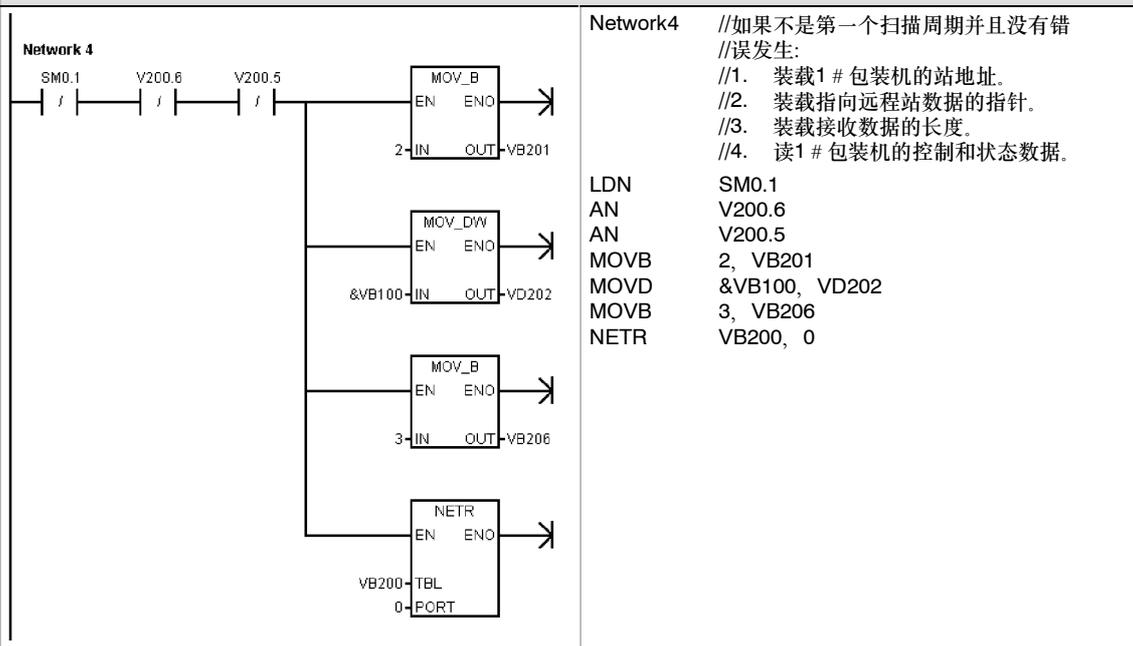
Network2 //当NETR完成标志位 (V200.7)
//置位并且包装完100
//箱时:
//1. 装载1# 包装机的站地址。
//2. 装载指向远程站数据的指针。
//3. 装载发送数据的长度。
//4. 装载发送数据。
//5. 复位1# 包装机包装的箱数。

```
LD V200.7
AW= VW208, +100
MOVB 2, VB301
MOVD &VB101, VD302
MOVB 2, VB306
MOVW +0, VW307
NETW VB300, 0
```

Network3 //当NETR完成标志位置位时,
//保存来自1# 包
//装机的控制数据。

```
LD V200.7
MOVB VB207, VB400
```

示例: 网络读写指令



发送和接收指令

发送指令（XMT）用于在自由口模式下依靠通讯口发送数据。

接收指令（RCV）启动或者终止接收信息功能。您必须为接收操作指定开始和结束条件。从指定的通讯口接收到的信息被存储在数据缓冲区（TBL）中。数据缓冲区的第一个数据指明了接收到的字节数。

使ENO=0的错误条件:

- 0006（间接寻址）
- 0009（在Port0同时发送和接收）
- 000B（在Port1同时发送和接收）
- RCV参数错误，置位SM86.6或者SM186.6
- S7-200 CPU没有处于自由口模式。

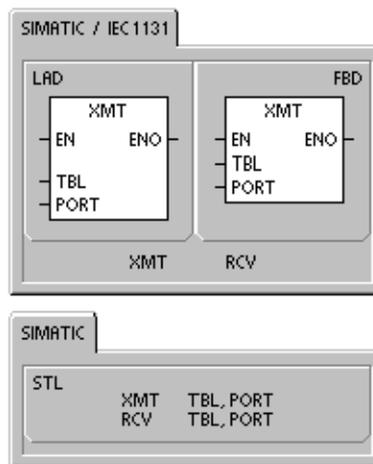


表6-12 发送和接收指令的有效操作数

输入/输出	数据类型	操作数
TBL	BYTE	IB、QB、VB、MB、SMB、SB、*VD、*LD、*AC
PORT	BYTE	常数 对于CPU221、CPU222、CPU224: 0 对于CPU224XP和CPU226: 0或1

关于使用自由口模式的更多信息，见第7章中用自由口模式创建用户自定义协议部分。

使用自由口模式控制串行通讯口

通过编程，您可以选择自由口模式来控制S7-200的串行通讯口。当选择了自由口模式，用户程序通过使用接收中断、发送中断、发送指令和接收指令来控制通讯口的操作。当处于自由口模式时，通讯协议完全由梯形图程序控制。SMB30（对于端口0）和SMB130（对于端口1，如果您的S7-200有两个端口的话）被用于选择波特率和校验类型。

当S7-200处于STOP模式时，自由口模式被禁止，重新建立正常的通讯（例如：编程设备的访问）。

在最简单的情况下，可以只用发送指令（XMT）向打印机或者显示器发送信息。其它例子包括与条码阅读器、称重计和焊机的连接。在每种情况下，您都必须编写程序，来支持在自由口模式下与S7-200通讯的设备所使用的协议。

只有当S7-200处于RUN模式时，才能进行自由口通讯。要使能自由口模式，应该在SMB30（端口0）或者SMB130（端口1）的协议选择区中设置01。处于自由口通讯模式时，不能与编程设备通讯。



提示

可以使用特殊寄存器位SM0.7来控制自由口模式。SM0.7反映的是操作模式开关的当前位置。当SM0.7=0时，开关处于TERM位置；当SM0.7=1时，开关处于RUN位置。如果只有模式开关处于RUN位置时，才允许自由口模式，您可以将开关改变到其它位置上，使用编程设备监控S7-200的运行。

将PPI通讯转变为自由口模式

SMB30和SMB130分别配置通讯口0和通讯口1，并且为自由口操作提供波特率、校验和数据位数的选择。自由口的控制字节如图6-8所示。每一个配置都产生一个停止位。

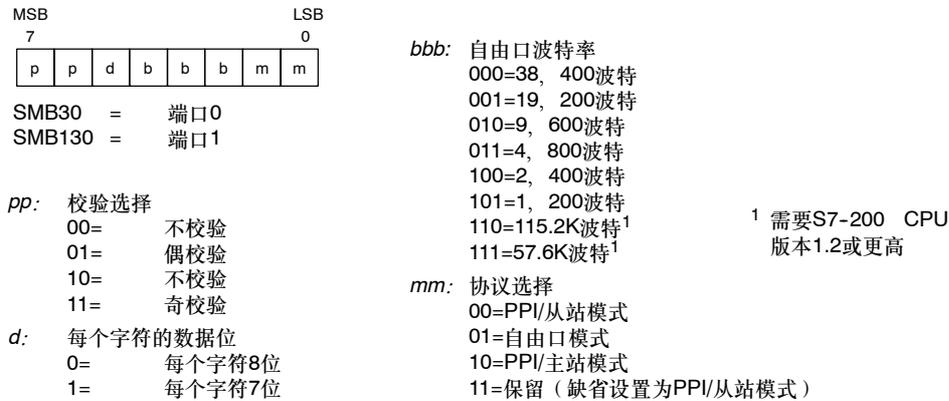


图6-8 用于自由口模式的SM控制字节 (SMB30或SMB130)

发送数据

发送指令使您能够发送一个字节或多个字节的缓冲区，最多为255个。

图6-9给出了发送缓冲区的格式。

如果有一个中断服务程序连接到发送结束事件上，在发送完缓冲区中的最后一个字符时，则会产生一个中断 (对端口0为中断事件9，对端口1为中断事件26)。



图6-9 发送缓冲区的格式

您可以不使用中断来执行发送指令 (例如：向打印机发送信息)。通过监视SM4.5或者SM4.6信号，判断发送是否完成。

把字符数设置为0并执行XMT指令，可以产生一个BREAK状态。这样产生的BREAK状态，在线上会持续以当前波特率传输16位数据所需要的时间。发送BREAK的操作和发送其它任何信息的操作是一样的。当BREAK完成时，产生一个发送中断并且SM4.5或者SM4.6反应发送操作的当前状态。

接收数据

接收指令使您能够接收一个字节或多个字节的缓冲区，最多为255个。

图6-10给出了接收缓冲区的格式。

如果有一个中断服务程序连接到接收信息完成事件上，在接收完缓冲区中的最后一个字符时，S7-200会产生一个中断 (对端口0为中断事件23，对端口1为中断事件24。)

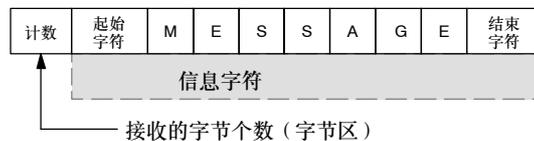


图6-10 接收缓冲区的格式

您可以不使用中断，通过监视SMB86 (端口0) 或者SMB186 (端口1) 来接收信息。当接收指令未被激活或者已经被中止时，这一字节不为0；当接收正在进行时，这一字节为0。

如表6-13中所示，接收指令允许您选择信息的起始和结束条件。使用SMB86至SMB94对端口0进行设置，SMB186至SMB194对端口1进行设置。



提示

当超限或有校验错误时，接收信息功能会自动终止。必须为接收信息功能操作定义一个起始条件和一个结束条件（最大字符数）。

表6-13 接收缓冲区字节（SMB86到SMB94和SMB186到SMB194）

端口0	端口1	中断描述																			
SMB86	SMB186	接收信息状态字节 <table style="margin-left: 20px;"> <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: right;">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: left;">0</td> </tr> </table> <p>n: 1= 接收信息功能终止: 用户发送禁止命令。</p> <p>r: 1= 接收信息功能终止: 输入参数错误或无起始或结束条件</p> <p>e: 1= 收到结束字符</p> <p>t: 1= 接收信息功能终止: 超时</p> <p>c: 1= 接收信息功能终止: 超出最大字符数</p> <p>p: 1= 接收信息功能终止: 奇偶校验错误。</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	接收信息控制字节 <table style="margin-left: 20px;"> <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: right;">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: left;">0</td> </tr> </table> <p>en: 0=禁止接收信息功能。 1=允许接收信息功能。 每次执行RCV指令时检查允许/禁止接收信息位。</p> <p>sc: 0=忽略SMB88或SMB188。 1=使用SMB88或SMB188的值检测起始信息。</p> <p>ec: 0=忽略SMB89或SMB189。 1=使用SMB89或SMB189的值检测结束信息。</p> <p>il: 0=忽略SMW90或SMW190。 1=使用SMW90或SMW190的值检测空闲状态。</p> <p>c/m: 0= 定时器是内部字符定时器。 1=定时器是信息定时器。</p> <p>tmr: 0=忽略SMW92或SMW192。 1=当SMW92或SMW192中的定时时间超出时终止接收。</p> <p>bk: 0=忽略中断条件。 1=用中断条件作为信息检测的开始。</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	信息字符的开始																			
SMB89	SMB189	信息字符的结束																			
SMW90	SMW190	空闲线时间段按毫秒设定。空闲线时间溢出后接收的第一个字符是新的信息的开始字符。SMB90（或SMB190）是最高有效字节，SMB91（或SMB191）是最低有效字节。																			
SMW92	SMW192	中间字符/信息定时器溢出值按毫秒设定。如果超过这个时间段，则终止接收信息。SMB92（或SMB192）是最高有效字节，SMB93（或SMB193）是最低有效字节。																			
SMB94	SMB194	要接收的最大字符数（1到255字节）。注：这个范围必须设置到所希望的最大缓冲区大小，即使信息的字符数始终达不到。																			

接收指令的启动和结束条件

接收指令使用接收信息控制字节（SMB87或SMB187）中的位来定义信息起始和结束条件。



提示

当接收指令执行时，在接收口上有来自其它器件的信号，接收信息功能有可能从一个字符的中间开始接收字符，从而导致校验错误和接收信息功能的中止。如果校验没有被使能，接收到的信息有可能包含错误字符。当起始条件被指定为一个特定的起始字符或任意字符时，这种情况有可能发生，正象下面第2条和第6条中所描述的那样。

接收指令支持几种信息起始条件。指定包含一个停顿或者一个空闲线检测的起始条件，通过在将字符放到信息缓冲区之前，用一个字符的起始来强制接收信息功能和信息的起始相同步，来避免以上问题。

接收指令支持几种起始条件:

1. **空闲线检测:** 空闲线条件是指在传输线上一段安静或者空闲的时间。在SMW90或者SMW190中指定其毫秒数。当接收指令在程序中执行时，接收信息功能对空闲线条件进行检测。如果在空闲线时间到之前接收到任何字符，接收信息功能会忽略那些字符并且按照SMW90或者SMW190中给定的时间值重新启动空闲线定时器。参见图6-11。在空闲线时间到之后，接收信息功能将所有接收到的字符存入信息缓冲区。

空闲线时间应该总是大于在指定波特率下传输一个字符（包括起始位、数据位、校验位和停止位）的时间。空闲线时间的典型值为在指定波特率下传输三个字符的时间。

对于二进制协议、没有特定起始字符的协议或者指定了信息之间最小时间间隔的协议，您可以使用空闲线检测作为起始条件。

设置: il=1, sc=0, bk=0, SMW90/SMW190=空闲线超时时间, 单位为毫秒



图6-11 用空闲时间检测来启动接收指令

2. **起始字符检测:** 起始字符可以是用于作为一条信息首字符的任意一个字符。当接收到SMB88或者SMB188中指定的起始字符后，一条信息开始。接收信息功能将起始字符作为信息的第一个字符存入接收缓冲区。接收信息功能忽略所有在起始字符之前接收到的字符。起始字符和起始字符之后接收到的所有字符一起存入信息缓冲区。

通常，对于所有信息都使用同一字符作为起始的ASCII码协议，您可以使用起使字符检测。

设置: il=0, sc=1, bk=0, SMW90/SM190被忽略, SMB88/SMB188=起始字符

3. **空闲线和起始字符:** 接收指令可以用空闲线和起始字符的组合来启动一条消息。当接收指令执行时,接收信息功能检测空闲线条件。在空闲线条件满足后,接收信息功能搜寻指定的起始字符。如果接收到的字符不是起始字符,接收信息功能重新检测空闲线条件。所有在空闲线条件满足和接收到起始字符之前接收到的字符被忽略掉。起始字符与字符串一起存入信息缓冲区。

空闲线时间应该总是大于在指定波特率下传输一个字符(包括起始位、数据位、校验位和停止位)的时间。空闲线时间的典型值为在指定波特率下传输三个字符的时间。

通常,对于指定信息之间最小时间间隔并且信息的首字符是特定设备的站号或其它信息的协议,您可以使用这种类型的起始条件。这种方式尤其适用于在通讯连接上有多个设备的情况。在这种情况下,只有当接收到的信息的起始字符为特定的站号或者设备时,接收指令才会触发一个中断。

设置: il=1, sc=1, bk=0, SMW90/SM190>0, SMB88/SMB188=起始字符

4. **断点检测:** 断点是指在大于一个完整字符传输时间的一段时间内,接收数据一直为0。一个完整字符传输时间定义为传输起始位、数据位、校验位和停止位的时间总和。如果接收指令被配置为用接收一个断点作为信息的起始,则任何在断点之后接收到的字符都会存入信息缓冲区。任何在断点之前接收到的字符都被忽略。

通常,只有当通讯协议需要时,才使用断点检测作为起始条件。

设置: il=0, sc=0, bk=1, SMW90/SM190被忽略, SMB88/SMB188被忽略

5. **断点和起始字符:** 接收指令可以被配置为接收到断点条件和一个指定的起始字符之后,启动接收。在断点条件满足之后,接收信息功能寻找特定的起始字符。如果收到了除起始字符以外的任意字符,接收信息功能重新启动寻找新的断点。所有在断点条件满足和接收到起始字符之前接收到的字符都会被忽略。起始字符与字符串一起存入信息缓冲区。

设置: il=0, sc=1, bk=1, SMW90/SM190被忽略,
SMB88/SMB188=起始字符

6. **任意字符:** 接收指令可以被配置为立即接收任意字符并把全部接收到的字符存入信息缓冲区。这是空闲线检测的一种特殊情况。在这种情况下,空闲线时间(SMW90或者SMW190)被设置为0。这使得接收指令一经执行,就立即开始接收字符。

设置: il=1, sc=0, bk=0, SMW90/SMW190=0, SMB88/SMB188被忽略

用任意字符开始一条信息允许使用信息定时器,来监控信息接收是否超时。这对于自由口协议的主站是非常有用的,并且当在指定时间内,没有来自从站的任何响应的情况,也需要采取超时处理。由于空闲线时间被设置为0,当接收指令执行时,信息定时器启动。如果没有其它终止条件满足,信息定时器超时会结束接收信息功能。

设置: il=1, sc=0, bk=0, SMW90/SMW190=0, SMB88/SMB188被忽略
c/m=1, tmr=1, SMW92/SMW192=信息超时时间,单位为毫秒。

接收指令支持几种结束信息的方式。结束信息的方式可以是以下一种或者几种的组合：

1. **结束字符检测**：结束字符是用于表示信息结束的任意字符。在找到起始条件之后，接收指令检查每一个接收到的字符，并且判断它是否与结束字符匹配。如果接收到了结束字符，将其存入信息缓冲区，接收结束。

通常，对于所有信息都使用同一字符作为结束的ASCII码协议，您可以使用结束字符检测。您可以使用结束字符检测与字符间隔定时器、信息定时器或者最大字符计数相结合来结束一条信息。

设置： ec=1, SMB89/SMB189=结束字符

2. **字符间隔定时器**：字符间隔时间是指从一个字符的结尾（停止位）到下一个字符的结尾（停止位）之间的时间。如果两个字符之间的时间间隔（包括第二个字符）超过了SMW92或者SMW192中指定的毫秒数，接收信息功能结束。接收到字符后，字符间隔定时器重新启动，见图6-12。

当协议没有特定的信息结束字符时，您可以用字符间隔定时器来结束一条信息。由于定时器总是包含接收一个完整字符（包括起始位、数据位、校验位和停止位）的时间，因而该时间值应设置为大于在指定波特率下传输一个字符的时间。

您可以使用字符间隔定时器与结束字符检测或者最大字符计数相结合，来结束一条信息。

设置： c/m=0, tmr=1, SMW92/SMW192=信息超时时间，单位为毫秒。



图6-12 使用字符间隔定时器来结束接收指令

3. **信息定时器**：从信息的开始算起，在经过一段指定的时间之后，信息定时器结束一条信息。接收信息功能的启动条件一满足，信息定时器就启动。当经过的时间超出SMW92或者SMW192中指定的毫秒数时，信息定时器时间到，见图6-13。

通常，当通讯设备不能保障字符中间没有时间间隔或者使用Modem通讯时，您可以使用信息定时器。对于Modem方式，您可以用信息定时器指定一个从信息开始算起，接收信息允许的最大时间。信息定时器的典型值是在当前波特率下，接收到最长信息所需时间值的大约1.5倍。

您可以使用信息定时器与结束字符检测或者最大字符计数相结合，来结束一条信息。

设置： c/m=1, tmr=1, SMW92/SMW192=信息超时时间，单位为毫秒。

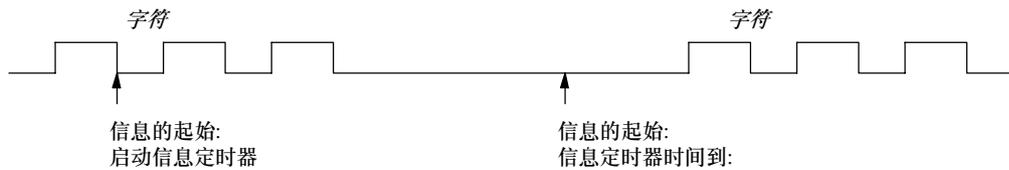


图6-13 使用信息定时器来结束接收指令

4. **最大字符计数:** 必须告诉接收指令接收字符的最大个数 (SMB94或者SMB194)。当达到或者超出这个值, 接收信息功能结束。即使不会被用作结束条件, 接收指令要求用户指定一个最大字符个数。这是因为接收指令需要知道接收信息的最大长度, 这样才能保证信息缓冲区之后的用户数据不会被覆盖。

对于信息的长度已知并且恒定的协议, 可以使用最大字符计数来结束信息。最大字符计数总是与结束字符检测、字符间隔定时器或者信息定时器结合在一起使用。

5. **校验错误:** 当接收字符的同时出现硬件信号校验错误时, 接收指令会自动结束。只有在SMB30或者SMB130中使能了校验位, 才有可能出现校验错误。没有办法禁止此功能。
6. **用户结束:** 用户可以通过程序来结束接收信息功能, 先将SMB87或SMB187中的使能位置为0, 再次执行接收指令即可。这样可以立即终止接收信息功能。

使用字符中断控制接收数据

为了完全适应对各种协议的支持, 您也可以使用字符中断控制的方式接收数据。接收每个字符时都会产生中断。在执行与接收字符事件相连的中断服务程序之前, 接收到的字符存入SMB2中, 校验状态 (如果使能的话) 存入SM3.0。SMB2是自由口接收字符缓冲区。在自由口模式下, 每一个接收到的字符都会存放到这一位置, 便于用户程序访问。SMB3用于自由口模式。它包含一个校验错误标志位。当接收字符的同时检测到校验错误时, 该位被置位。该字节的其它位被保留。利用校验位去丢弃信息或向该信息发送否定应答。

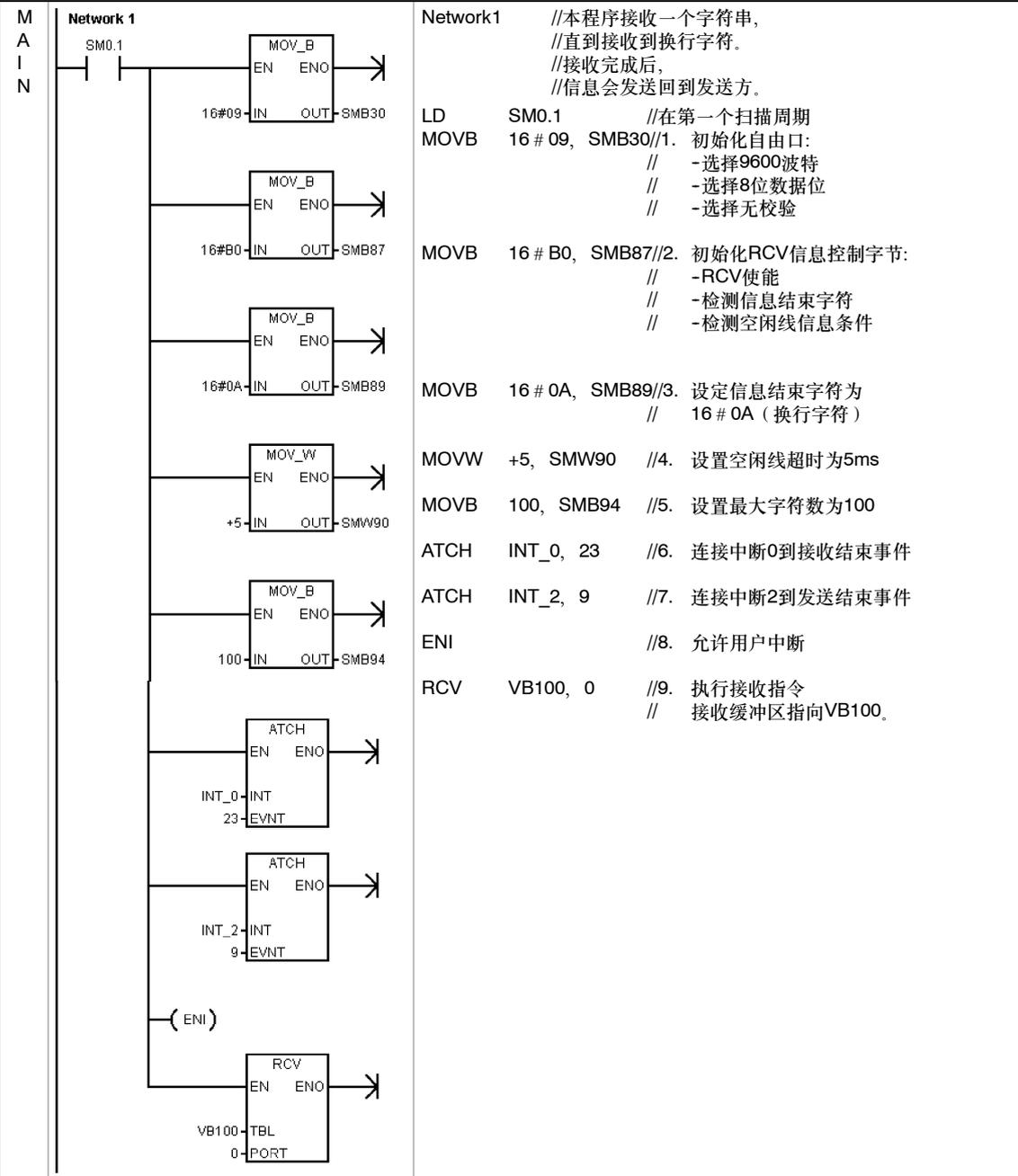
在较高的波特率下 (38.4K到115.2K) 使用字符中断时, 中断之间的时间间隔会非常短。例如: 在38.4时为260毫秒; 在57.6K时为173毫秒; 在115.2K时为86毫秒。确保您的中断服务程序足够短, 不会丢失字符或者使用接收指令。



提示

SMB2和SMB3是端口0和端口1共用的。当字符的接收来自端口0时, 执行与事件 (中断事件8) 相连的中断服务程序, SMB2中存储从端口0接收到的字符, SMB3中存储该字符的校验状态。当字符的接收来自端口1时, 执行与事件 (中断事件25) 相连的中断服务程序, SMB2中存储从端口1接收到的字符, SMB3中存储该字符的校验状态。

示例: 发送和接收指令



示例: 发送和接收指令		
<p>INT 0</p>	<p>Network 1</p>	<p>Network1 //接收完成中断 //1. 如果接收状态显示接收到结束字符, // 连接一个 // 10ms定时器触发发送, 然后返回。 //2. 如果由于任何其它原因接收完成, // 启动一个新的接收。</p> <pre> LDB= SMB86, 16#20 MOVB 10, SMB34 ATCH INT_1, 10 CRET NOT RCV VB100, 0 </pre>
<p>INT 1</p>	<p>Network 1</p>	<p>Network1 //10ms定时器中断 //1. 断开定时器中断 //2. 在端口0向用户回送信息</p> <pre> LD SM0.0 DTCH 10 XMT VB100, 0 </pre>
<p>INT 2</p>	<p>Network 1</p>	<p>Network1 //发送完成中断 //允许另一个接收</p> <pre> LD SM0.0 RCV VB100, 0 </pre>

获取口地址和设定口地址指令

获取口地址指令（GPA）读取PORT指定的CPU口的站地址，并将数值放入ADDR指定的地址中。

设定口地址指令（SPA）将口的站地址（PORT）设置为ADDR指定的数值。新地址不能永久保存。重新上电后，口地址将返回到原来的地址值（用系统块下载的地址）。

使ENO=0的错误条件:

- 0006（间接寻址）
- 0004（试图在中断服务程序中执行设定口地址指令）

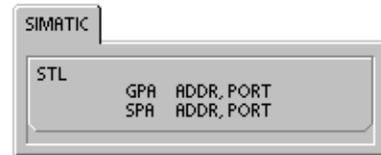
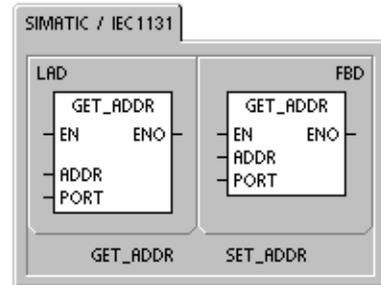


表6-14 获取口地址和设定口地址指令的有效操作数

输入/输出	数据类型	操作数
ADDR	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数 (常数值仅用于设定口地址指令。)
PORT	BYTE	常数 对于CPU221、CPU222、CPU224: 0 对于CPU224XP和CPU226: 0或1

比较指令

数值比较

比较指令用于比较两个数值:

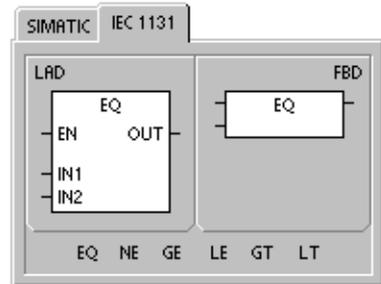
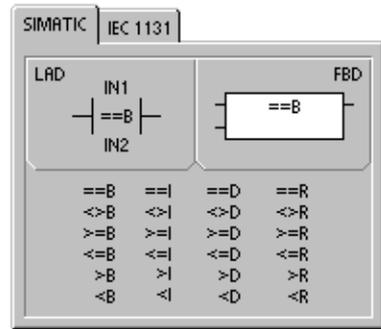
IN1 = IN2 IN1 >= IN2 IN1 <= IN2
 IN1 > IN2 IN1 < IN2 IN1 <> IN2

字节比较操作是无符号的。
 整数比较操作是有符号的。
 双字比较操作是有符号的。
 实数比较操作是有符号的。

对于LAD和FBD: 当比较结果为真时, 比较指令使能点闭合 (LAD) 或者输出接通 (FBD)。

对于STL: 当比较结果为真时, 将栈顶值置1。

当您使用IEC比较指令时, 您可以使用各种数据类型作为输入。但是, 两个输入的数据类型必须一致。



注释

下列情况是致命错误, 并且会导致S7-200立即停止执行用户程序:

- 非法的间接地址 (任意比较指令)
- 非法的实数 (例如: NAN), (实数比较指令)

为了避免这些情况的发生, 在执行比较指令之前, 要确保合理使用了指针和存储实数的数值单元。

不管能流的状态如何, 比较指令都会被执行。

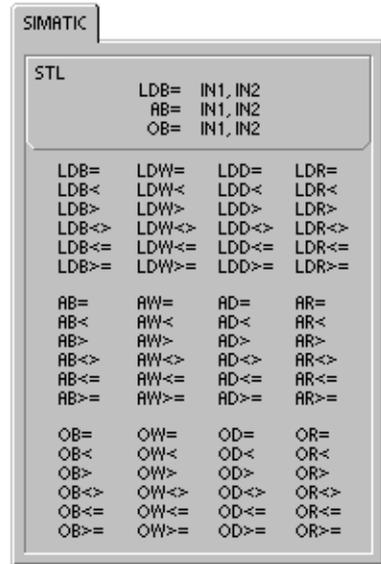
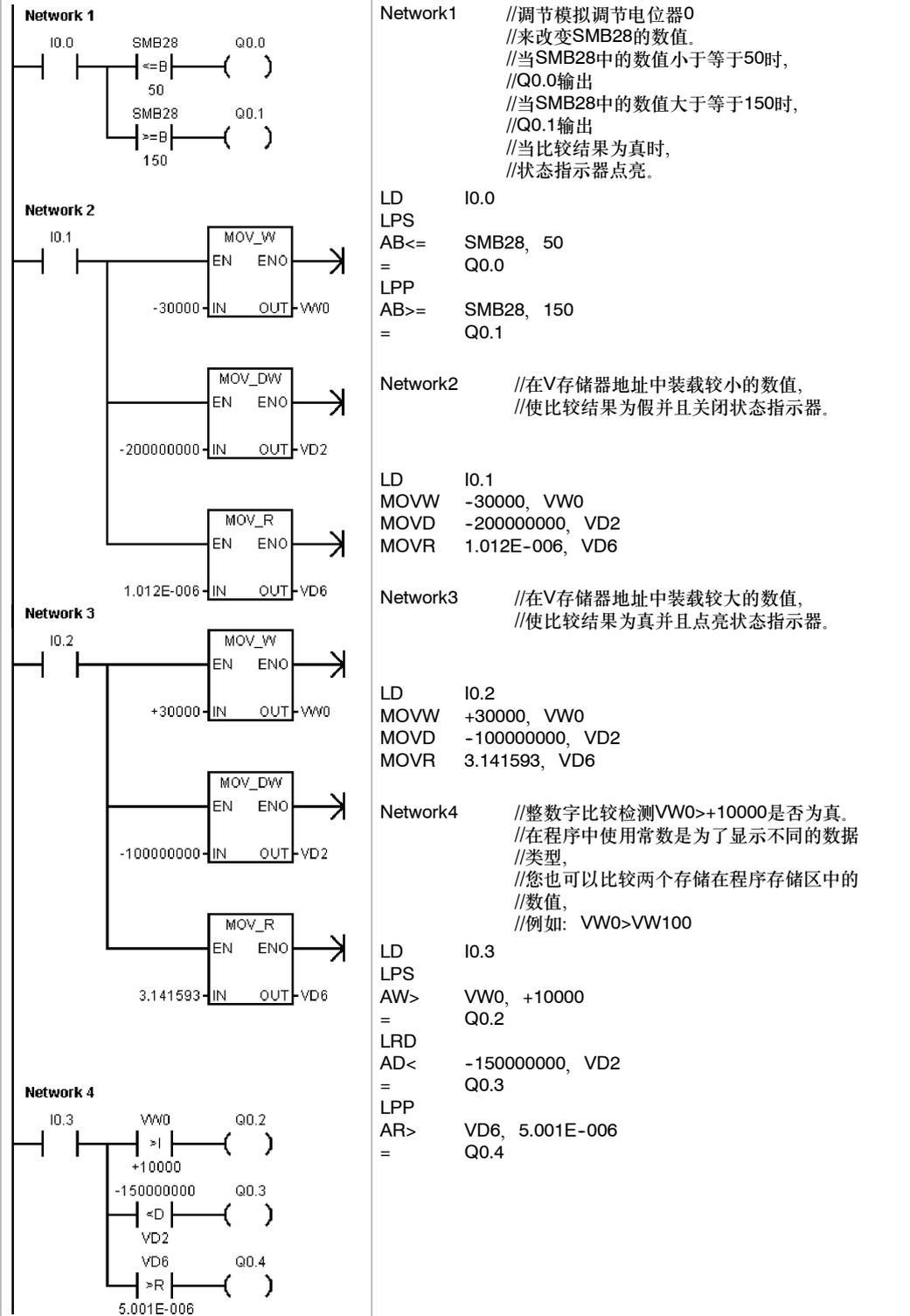


表6-15 比较指令的有效操作数

输入/输出	定时器类型	操作数
IN1, IN2	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常数
OUT	BOOL	I、Q、V、M、SM、S、T、C、L、能流

示例: 比较指令



字符串比较

字符串比较指令比较两个字符串的ASCII码字符:

IN1=IN2 IN1<>IN2

当比较结果为真时，比较指令使触点闭合（LAD）或者输出接通（FBD），或者将栈顶位置1（STL）。

注释

下列情况是致命错误，并且会导致S7-200立即停止执行用户程序:

- 非法的间接地址（任意比较指令）
- 字符串的长度超过254个字符（字符串比较指令）
- 一个字符串的起始地址和长度使它不适合所指定的存储区（字符串比较指令）

为了避免这些情况的发生，在执行比较指令之前，要确保合理使用了指针和保存ASCII码字符串的存储区。确保一个保存ASCII码字符串的缓冲区能够在指定的存储区完整的保留。

不管能流的状态如何，比较指令都会被执行。

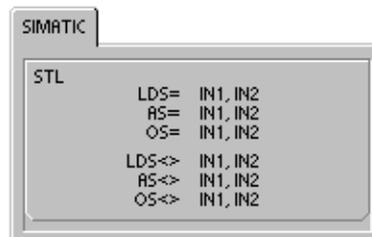
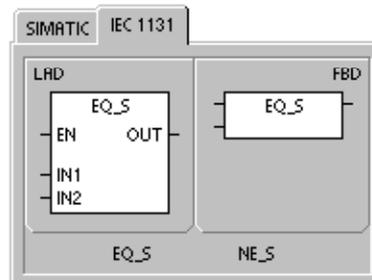
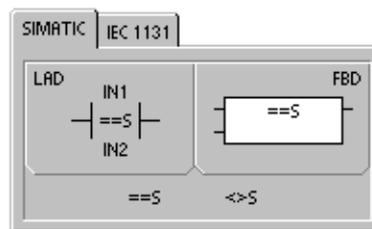


表6-16 字符串比较指令的有效操作数

输入/输出	定时器类型	操作数
IN1	STRING	VB, LB, *VD, *LD, *AC, 常数
IN2	STRING	VB, LB, *VD, *LD, *AC
OUT	BOOL	I、Q、V、M、SM、S、T、C、L、能流

转换指令

标准转换指令

数字转换

字节转为整数 (BTI)、整数转为字节 (ITB)、整数转为双整数 (ITD)、双整数转为整数 (DTI)、双整数转为实数 (DTR)、BCD码转为整数 (BCDI) 和整数转为BCD码 (IBCD)。以上指令将输入值IN转换为指定的格式并存储到由OUT指定的输出值存储区中。例如：您可以将双整数值转为实数值；您也可以将整数和BCD码格式之间相互转换。

四舍五入和取整

四舍五入指令 (ROUND) 将一个实数转为一个双整数值，并将四舍五入的结果存入OUT指定的变量中。

取整指令 (TRUNC) 将一个实数转为一个双整数值，并将实数的整数部分作为结果存入OUT指定的变量中。

包络段数

段码指令 (SEG) 允许您产生一个点阵，用于点亮七段码显示器的各个段。

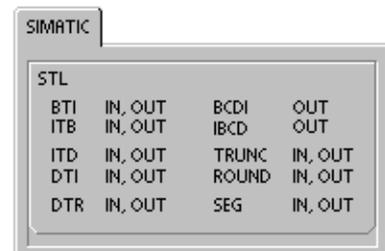
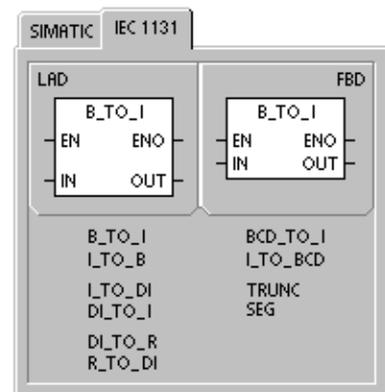
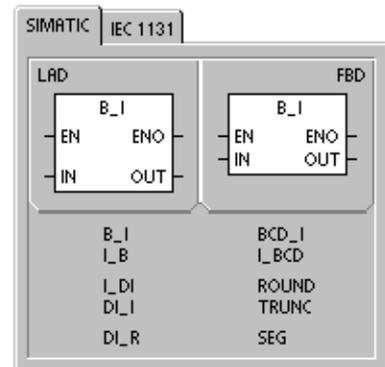


表6-17 标准转换指令的有效操作数

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

BCD码转为整数和整数转为BCD码的操作

BCD码转整数指令（BCDI）将一个BCD码IN的值转换成整数值，并且将结果存入OUT指定的变量中。IN的有效范围是0到9999的BCD码。

使ENO=0的错误条件:

- SM1.6（无效的BCD码）
- 0006（间接寻址）

整数转BCD码指令（IBCD）将输入的整数值IN转换成BCD码，并且将结果存入OUT指定的变量中。IN的有效范围是0到9999的整数。

受影响的SM标志位:

- SM1.6（无效的BCD码）

双整数转为实数指令的操作

双整数转实数指令（DTR）将一个32位，有符号整数值IN转换成一个32位实数，并将结果存入OUT指定的变量中。

使ENO=0的错误条件:

- 0006（间接寻址）

双整数转为整数指令的操作

双整数转整数指令（DTI）将一个双整数值IN转换成一个整数值，并将结果存入OUT指定的变量中。

使ENO=0的错误条件:

- SM1.1（溢出）
- 0006（间接寻址）

如果所转换的数值太大以致于无法在输出中表示，则溢出标志位置位并且输出不会改变。

受影响的SM标志位:

- SM1.1（溢出）

整数转为双整数指令的操作

整数转双整数指令（ITD）将整数值IN转换成双整数值，并且存入OUT指定的变量中。符号位扩展到高字节中。

使ENO=0的错误条件:

- 0006（间接寻址）

字节转为整数指令的操作

字节转整数指令（BTI）将字节值IN转换成整数值，并且存入OUT指定的变量中。字节是无符号的，因而没有符号位扩展。

使ENO=0的错误条件:

- 0006（间接寻址）

整数转为字节指令的操作

整数转字节指令（ITB）将一个字的值IN转换成一个字节值，并且存入OUT指定的变量中。只有0到255中的值被转换，所有其他值会产生溢出并且输出不会改变。

使ENO=0的错误条件:

- SM1.1（溢出）
- 0006（间接寻址）

受影响的SM标志位:

- SM1.1（溢出）



提示

如果想将一个整数转换成实数，先用整数转双整数指令，再用双整数转实数指令。

四舍五入取整和取整指令的操作

四舍五入取整指令 (ROUND) 将实数值IN转换成双整数值, 并且存入OUT指定的变量中。如果小数部分大于等于0.5, 则数字向上取整。

取整指令 (TRUNC) 将一个实数值IN转换成一个双整数, 并且存入OUT指定的变量中。只有实数的整数部分被转换, 小数部分舍去。

如果所转换的不是一个有效的实数, 或者其数值太大以致于无法在输出中表示, 则溢出标志位位置并且输出不会改变。

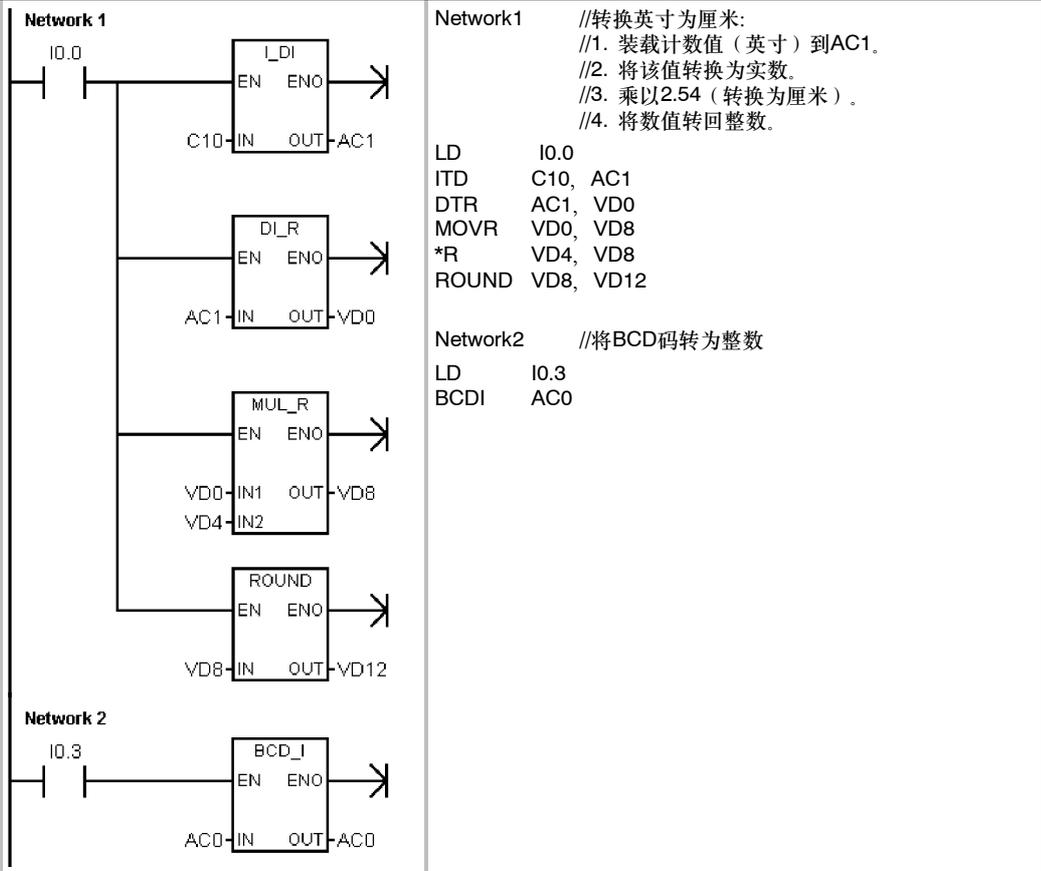
使ENO=0的错误条件:

- SM1.1 (溢出)
- 0006 (间接寻址)

受影响的SM标志位:

- SM1.1 (溢出)

示例: 标准转换指令



双字整数转实数和四舍五入取整

C10	101	计数值=101英寸
VD0	101.0	计数值 (实数形式)
VD4	2.54	2.54常数 (英寸到厘米)
VD8	256.54	256.54厘米数 (实数形式)
VD12	257	257厘米数 (双整数形式)

BCD转整数

AC0	1234
BCDI	
AC0	04D2

段码指令的操作

要点亮七段码显示器中的段，可以使用段码指令。段码指令将IN中指定的字符（字节）转换生成一个点阵并存入OUT指定的变量中。

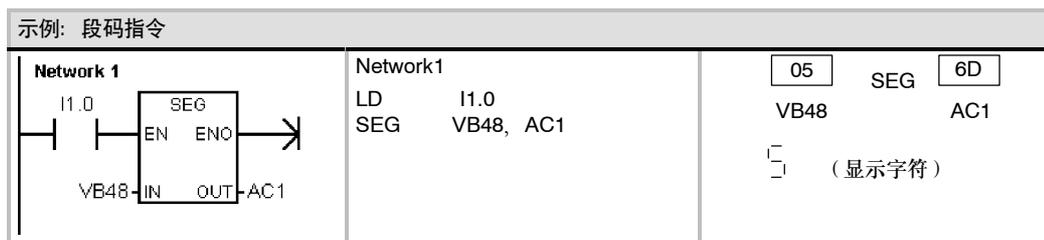
点亮的段表示的是输入字节中低4位所代表的字符。图6-14给出了段码指令使用的七段码显示器的编码。

使ENO=0的错误条件:
 ■ 0006 (间接寻址)

输入 LSD	七段码 显示器	输出			
		- 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		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		

输入 LSD	七段码 显示器	输出			
		- 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 七段码显示器的编码



ASCII码转换指令

有效的ASCII码字符为十六进制的30到39和41到46。

在ASCII码和十六进制数之间相互转换

ASCII码转十六进制数指令（ATH）将一个长度为LEN从IN开始的ASCII码字符串转换成从OUT开始的十六进制数。十六进制数转ASCII码指令（HTA）将从输入字节IN开始的十六进制数，转换成从OUT开始的ASCII码字符串。被转换的十六进制数的位数由长度LEN给出。

能够被转换的ASCII码字符串或者十六进制数的最大数量为255。有效ASCII码输入。

有效的ASCII码输入字符是0到9的十六进制数代码值30到39，和大写字母A到F的十六进制数代码值41到46这些字母数字字符。

使ENO=0的错误条件:

- SM1.7（非法的ASCII码）只对ATH有效
- 0006（间接寻址）
- 0091（操作数超出范围）

受影响的SM标志位:

- SM1.7（非法的ASCII码）

将数值转为ASCII码

整数转ASCII码（ITA）、双整数转ASCII码（DTA）和实数转ASCII码（RTA）指令，分别将整数、双整数或实数值转换成ASCII码字符。

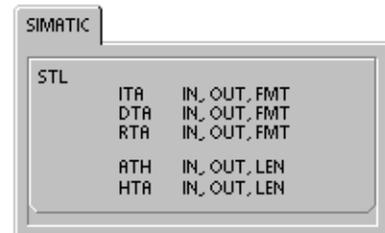
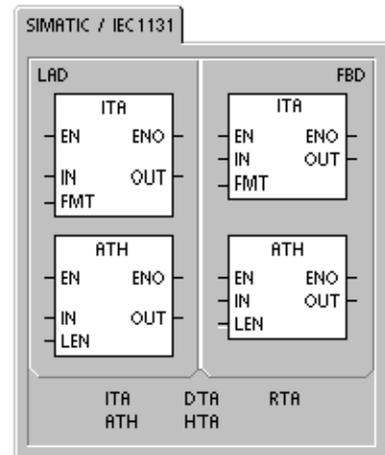


表6-18 ASCII码转换指令的有效操作数

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

整数转ASCII码指令的操作数

整数转ASCII码（ITA）指令将一个整数IN转换成一个ASCII码字符串。格式FMT指定小数点右侧的转换精度和小数点是使用逗号还是点号。转换结果放在OUT指定的连续8个字节中。

使ENO=0的错误条件:

- 0006（间接寻址）
- 非法的格式
- $nnn > 5$

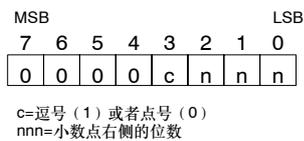
ASCII码字符串始终是8个字节。

整数转ASCII码指令的格式操作数如图6-15所示。输出缓冲区的大小始终是8个字节，*nnn*表示输出缓冲区中小数点右侧的数字位数。*nnn*的合理范围是0到5。将小数点右侧的位数定为0，使得所显示的数值没有小数点。对于*nnn*大于5的情况，输出缓冲区会被空格键的ASCII码填冲。*c*指定是用逗号(*c*=1)或者点号(*c*=0)作为整数和小数的分隔符。高4位必须为0。

图6-15中给出了一个数值的例子，其格式为使用点号(*c*=0)，小数点右侧有三位小数(*nnn*=011)。输出缓冲区的格式符合以下规则：

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号(-)开头。
- 小数点左侧的开头的0(除去靠近小数点的那个之外)被隐藏。
- 数值在输出缓冲区中是右对齐的。

FMT



| 输出 |
|----|----|----|----|----|----|----|----|
| +1 | +2 | +3 | +4 | +5 | +6 | +7 | |
| | | | 0 | . | 0 | 1 | 2 |
| | | - | 0 | . | 1 | 2 | 3 |
| | | | 1 | . | 2 | 3 | 4 |
| | - | 1 | 2 | . | 3 | 4 | 5 |

输入=12
输入=-123
输入=1234
输入=-12345

图6-15 整数转ASCII码(ITA)指令的FMT操作数

双整数转ASCII码指令操作

双整数转ASCII码(DTA)指令将一个双字IN转换成一个ASCII码字符串。格式操作数FMT指定小数点右侧的转换精度。转换结果存储在从OUT开始的连续12个字节中。

- 使ENO=0的错误条件:
- 0006(间接寻址)
 - 非法的格式
 - *nnn*>5

输出缓冲区的大小总是12个字节。

图6-16描述了双整数转ASCII码指令的格式操作数。*nnn*表示输出缓冲区中小数点右侧的数字位数。*nnn*的合理范围是0到5。将小数点右侧的位数定为0，使得所显示的数值没有小数点。对于*nnn*大于5的情况，输出缓冲区会被空格键的ASCII码填冲。*c*指定是用逗号(*c*=1)或者点号(*c*=0)作为整数和小数的分隔符。高4位必须为0。

图6-16中给出了一个数值的例子，其格式为使用点号(*c*=0)，小数点右侧有四位小数(*nnn*=100)。输出缓冲区的格式符合以下规则：

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号(-)开头。
- 小数点左侧的开头的0(除去靠近小数点的那个之外)被隐藏。
- 数值在输出缓冲区中是右对齐的。

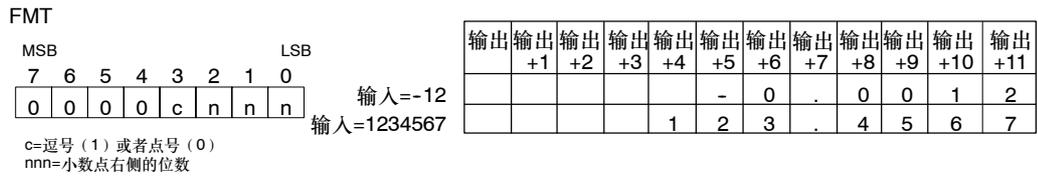


图6-16 双整数转ASCII码 (DTA) 指令的FMT操作数

实数转ASCII码指令操作

实数转ASCII码指令 (RTA) 将一个实数值IN转为ASCII码字符串。格式操作数FMT指定小数点右侧的转换精度，小数点是用逗号还是用点号表示和输出缓冲区的大小。

转换结果存储在从OUT开始的输出缓冲区中。

- 使ENO=0的错误条件:
- 0006 (间接寻址)
 - nnn>5
 - ssss<3
 - ssss<OUT中的字符个数

结果ASCII码字符的位数 (或长度) 就是输出缓冲区的大小，它的值可以在3到15字节或字符之间。

S7-200的实数格式支持最多7位小数。试图显示7位以上的小数会产生一个四舍五入错误。

图6-17是对RTA指令中格式操作数FMT的描述。ssss表示输出缓冲区的大小，0、1或者2个字节的大小是不合理的。nnn表示输出缓冲区中小数点右侧的数字位数，nnn的合理范围为0到5。将小数点右侧的位数定为0，使得所显示的数值没有小数点。对于nnn大于5或者指定的输出缓冲区太小以致于无法存储转换值的情况，输出缓冲区会被空格键的ASCII码填冲。c指定是用逗号 (c=1) 或者点号 (c=0) 作为整数和小数的分隔符。

图6-17中给出了一个数值的例子，其格式为：使用点号 (c=0)、小数点右侧有1位小数 (nnn=001) 和6个字节的缓冲区大小 (ssss=0110)。输出缓冲区的格式符合以下规则：

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号 (-) 开头。
- 小数点左侧的开头的0 (除去靠近小数点的那个之外) 被隐藏。
- 小数点右侧的数值按照指定的小数点右侧的数字位数被四舍五入。
- 输出缓冲区的大小应至少比小数点右侧的数字位数多三个字节。
- 数值在输出缓冲区中是右对齐的。

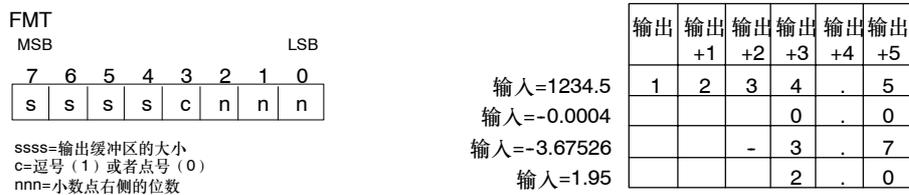


图6-17 实数转ASCII码 (RTA) 指令的FMT操作数

示例: ASCII码转十六进制数指令

<p>Network 1</p>	<p>Network1</p> <pre>LD I3.2 ATH VB30, VB40, 3</pre>																		
<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">'3'</td> <td style="text-align: center;">'E'</td> <td style="text-align: center;">'A'</td> <td style="text-align: center;">ATH</td> <td style="text-align: center;">'3E'</td> <td style="text-align: center;">'Ax'</td> </tr> <tr> <td style="text-align: center;">33</td> <td style="text-align: center;">45</td> <td style="text-align: center;">41</td> <td></td> <td style="text-align: center;">3E</td> <td style="text-align: center;">Ax</td> </tr> <tr> <td style="text-align: center;">VB30</td> <td></td> <td></td> <td></td> <td style="text-align: center;">VB40</td> <td></td> </tr> </table>	'3'	'E'	'A'	ATH	'3E'	'Ax'	33	45	41		3E	Ax	VB30				VB40		<p>注意: X表示 “半字节” (一个字节的一半) 未发生变化。</p>
'3'	'E'	'A'	ATH	'3E'	'Ax'														
33	45	41		3E	Ax														
VB30				VB40															

示例: 整数转ASCII码指令

<p>Network 1</p>	<p>Network 1</p> <pre>//将VW2中的整数值转换为从VB10 //开始的8个ASCII码字符, //使用16#0B的格式, //用逗号作小数点, 保留3位小数。 LD I2.3 ITA VW2, VB10, 16#0B</pre>																														
<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">12345</td> <td style="text-align: center;">ITA</td> <td style="text-align: center;">' '</td> <td style="text-align: center;">' '</td> <td style="text-align: center;">'1'</td> <td style="text-align: center;">'2'</td> <td style="text-align: center;">','</td> <td style="text-align: center;">'3'</td> <td style="text-align: center;">'4'</td> <td style="text-align: center;">'5'</td> </tr> <tr> <td style="text-align: center;">20</td> <td></td> <td style="text-align: center;">20</td> <td style="text-align: center;">31</td> <td style="text-align: center;">32</td> <td style="text-align: center;">2C</td> <td style="text-align: center;">33</td> <td style="text-align: center;">34</td> <td style="text-align: center;">35</td> <td></td> </tr> <tr> <td style="text-align: center;">VW2</td> <td></td> <td style="text-align: center;">VB10</td> <td style="text-align: center;">VB11</td> <td style="text-align: center;">...</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	12345	ITA	' '	' '	'1'	'2'	','	'3'	'4'	'5'	20		20	31	32	2C	33	34	35		VW2		VB10	VB11	...						
12345	ITA	' '	' '	'1'	'2'	','	'3'	'4'	'5'																						
20		20	31	32	2C	33	34	35																							
VW2		VB10	VB11	...																											

示例: 实数转ASCII码指令

<p>Network 1</p>	<p>Network 1</p> <pre>//将VD2中的实数值转换成 //从VB10开始的10个ASCII码字符, //使用16#A3的格式 //（用点号作小数点, //后面跟3位小数）。 LD I2.3 RTA VD2, VB10, 16#A3</pre>																																				
<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">123.45</td> <td style="text-align: center;">RTA</td> <td style="text-align: center;">' '</td> <td style="text-align: center;">' '</td> <td style="text-align: center;">' '</td> <td style="text-align: center;">'1'</td> <td style="text-align: center;">'2'</td> <td style="text-align: center;">'3'</td> <td style="text-align: center;">'.'</td> <td style="text-align: center;">'4'</td> <td style="text-align: center;">'5'</td> <td style="text-align: center;">'0'</td> </tr> <tr> <td style="text-align: center;">20</td> <td></td> <td style="text-align: center;">20</td> <td style="text-align: center;">20</td> <td style="text-align: center;">31</td> <td style="text-align: center;">32</td> <td style="text-align: center;">33</td> <td style="text-align: center;">2E</td> <td style="text-align: center;">34</td> <td style="text-align: center;">35</td> <td style="text-align: center;">30</td> <td></td> </tr> <tr> <td style="text-align: center;">VD2</td> <td></td> <td style="text-align: center;">VB10</td> <td style="text-align: center;">VB11</td> <td style="text-align: center;">...</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table>	123.45	RTA	' '	' '	' '	'1'	'2'	'3'	'.'	'4'	'5'	'0'	20		20	20	31	32	33	2E	34	35	30		VD2		VB10	VB11	...								
123.45	RTA	' '	' '	' '	'1'	'2'	'3'	'.'	'4'	'5'	'0'																										
20		20	20	31	32	33	2E	34	35	30																											
VD2		VB10	VB11	...																																	

字符串转换指令

将数值转换为字符串

整数转字符串（ITS）、双整数转字符串（DTS）和实数转字符串（RTS）指令，将整数、双整数或实数值（IN）转换成ASCII码字符串（OUT）。

整数转字符串的操作

整数转字符串指令（ITS）将一个整数IN转换为8个字符长的ASCII码字符串。格式操作数FMT指定小数点右侧的转换精度和使用逗号还是点号作为小数点。结果字符串被写入从OUT开始的9个连续字节中。要得到更多信息，请参见第章字符串的格式一节。

使ENO=0的错误条件:

- 0006（间接寻址）
- 0091（操作数超出范围）
- 非法格式（nnn>5）

图6-18是对整数转字符串指令中格式操作数的描述。输出字符串的长度总是8个字符。nnn表示输出缓冲区中小数点右侧的数字位数。nnn的合理范围为0到5。将小数点右侧的位数定为0，使得所显示的数值没有小数点。如果nnn的值大于5，输出是由8个空格键的ASCII码组成的字符串。c指定是用逗号（c=1）或者点号（c=0）作为整数和小数的分隔符。格式操作数的高4位必须为0。

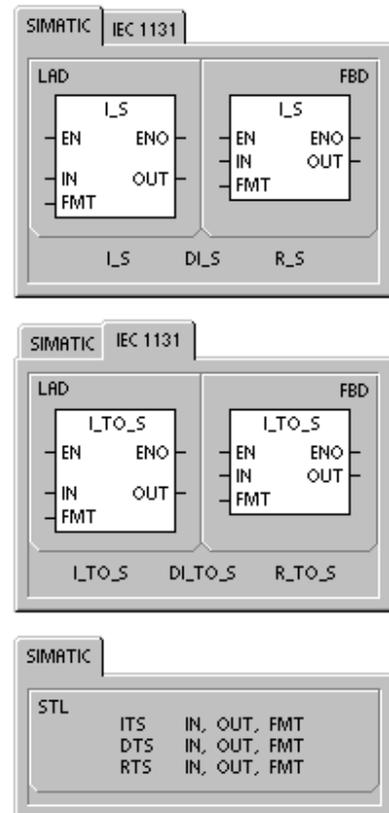


图6-18中给出了一个数值的例子，其格式为：使用点号（c=0）并且小数点后保留3位小数。OUT的值为字符串的长度。

输出缓冲区的格式符合以下规则:

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号（-）开头。
- 小数点左侧的开头的0（除去靠近小数点的那个之外）被隐藏。
- 数值在输出缓冲区中是右对齐的。

表6-19 数值转字符串指令的有效操作数

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

FMT

MSB							LSB
7	6	5	4	3	2	1	0
0	0	0	0	c	n	n	n

c=逗号(1) 或者点号(0)
nnn=小数点右侧的位数

输出	输出+1	输出+2	输出+3	输出+4	输出+5	输出+6	输出+7	输出+8
8				0	.	0	1	2
8				0	.	1	2	3
8				1	.	2	3	4
8		-	1	2	.	3	4	5

输入=12
输入=-123
输入=1234
输入=-12345

图6-18 整数转字符串指令的FMT操作数

双整数转字符串指令操作

双整数转字符串指令(DTS)将一个双整数IN转换为一个长度为12个字符的ASCII码字符串。格式操作数FMT指定小数点右侧的转换精度和使用逗号还是点号作为小数点。结果字符串被写入从OUT开始的连续13个字节。要得到更多信息,请参见第4章字符串的格式一节。

使ENO=0的错误条件:

- 0006(间接寻址)
- 0091(操作数超出范围)
- 非法格式(nnn>5)

图6-19是对整数转字符串指令中格式操作数的描述。输出字符串的长度总是8个字符。nnn表示输出缓冲区中小数点右侧的数字位数。nnn的合理范围为0到5。将小数点右侧的位数定为0,使得所显示的数值没有小数点。如果nnn的值大于5,输出是由12个空格键的ASCII码组成的字符串。c指定是用逗号(c=1)或者点号(c=0)作为整数和小数的分隔符。格式操作数的高4位必须为0。

图6-19中给出一个数值的例子,其格式为:使用点号(c=0)并且小数点后保留4位小数。OUT的值为字符串的长度。输出缓冲区的格式符合以下规则:

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号(-)开头。
- 小数点左侧的开头的0(除去靠近小数点的那个之外)被隐藏。
- 数值在输出缓冲区中是右对齐的。

FMT

MSB							LSB
7	6	5	4	3	2	1	0
0	0	0	0	c	n	n	n

c=逗号(1) 或者点号(0)
nnn=小数点右侧的位数

输出	输出+1	输出+2	输出+3	输出+4	输出+5	输出+6	输出+7	输出+8	输出+9	输出+10	输出+11	输出+12
12					.	-	0	.	0	0	1	2
12					1	2	3	.	4	5	6	7

输入=12
输入=-1234567

图6-19 整数转字符串指令的FMT操作数

实数转字符串指令操作

实数转字符串指令（RTS）将一个实数值IN转换为一个ASCII码字符串。格式操作数FMT指定小数点右侧的转换精度和使用逗号还是点号作为小数点。

- 使ENO=0的错误条件:
- 0006（间接寻址）
 - 0091（操作数超出范围）
 - 非法的格式
 - nnn > 5
 - ssss < 3
 - ssss < 要求的字符数

转换结果放在从OUT开始的一个字符串中。结果字符串的长度由格式操作数给出，它可以是3到15个字符。要得到更多信息，请参见第4章字符串的格式一节。

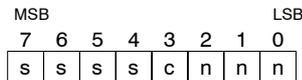
S7-200的实数格式支持最多7位小数。试图显示7位以上的小数会产生一个四舍五入错误。

图6-20是对实数转字符串指令中格式操作数的描述。ssss表示输出字符串的长度。0、1或者2个字节的大小是不合理的。nnn表示输出缓冲区中小数点右侧的数字位数。nnn的合理范围为0到5。将小数点右侧的位数定为0，使得所显示的数值没有小数点。对于nnn大于5或者指定的输出缓冲区太小以致于无法存储转换值的情况，输出缓冲区会被空格键的ASCII码填冲。c指定是用逗号（c=1）或者点号（c=0）作为整数和小数的分隔符。

图6-20中给出了一个数值的例子，其格式为：使用点号（c=0），小数点右侧有1位小数（nnn=001）和个字节的缓冲区大小（ssss=0110）。OUT的值为字符串的长度。输出缓冲区的格式符合以下规则：

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号（-）开头。
- 小数点左侧的开头的0（除去靠近小数点的那个之外）被隐藏。
- 小数点右侧的数值按照指定的小数点右侧的数字位数被四舍五入。
- 输出缓冲区的大小应至少比小数点右侧的数字位数多三个字节。
- 数值在输出缓冲区中是右对齐的。

FMT



ssss=输出字符串长度
c=逗号（1）或者点号（0）
nnn=小数点右侧的位数

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

图6-20 实数转字符串指令的FMT操作数

将子字符串转换为数字值

子字符串转整数（STI）、子字符串转双整数（STD）和子字符串转实数（STR）指令，将从偏移量INDX开始的字符串值IN转换成整数/双整数或实数值OUT。

使ENO=0的错误条件:

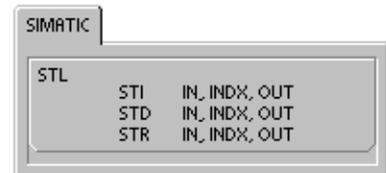
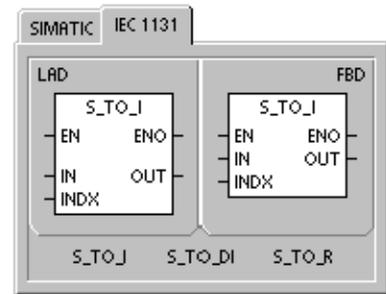
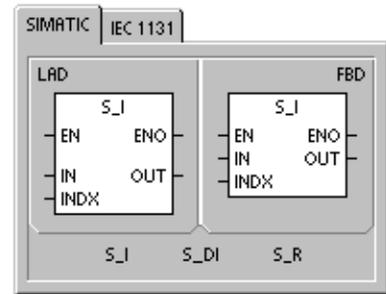
- 0006（间接寻址）
- 0091（操作数超出范围）
- 009B（偏移量=0）
- SM1.1（溢出）

子字符串转整数和子字符串转双整数指令将字符串转换为以下格式：[空格][+或-]数字0 - 9]

子字符串转实数指令将字符串转换为以下格式：
[空格][+或-]数字0 - 9]. 或， [数字0 - 9]

INDX值通常设置为1，从字符串的第一个字符开始转换。INDX可以被设置为其它值，从字符串的不同位置进行转换。这可以被用于字符串中包含非数值字符的情况。例如：输入字符串为“Temperature: 77.8”，您可以将INDX设为13，这样就可以跳过字符串开头的“Temperature: ”。

子字符串转实数指令不能用于转换为科学计数法或者指数形式表示实数的字符串。指令不会产生溢出错误（SM1.1），但是它会先将字符串转换到指数之前，然后停止转换。例如：字符串“1.234E6”转换为实数值1.234，并且没有错误提示。



当到达字符串的结尾或者遇到第一个非法字符时，转换指令结束。非法字符是指任意非数字(0 - 9)字符。

当转换产生的整数值过大以致输出值无法表示时，溢出标志（SM1.1）会置位。例如：当输入字符串产生的数值大于32767或者小于-32768时，子字符串转整数指令会置位溢出标志。

当输入字符串中并不包含可以转换的合法数值时，溢出标志（SM1.1）也会置位。例如：如果输入字符串的“A123”，转换指令会置位SM1.1（溢出）并且输出值保持不变。

表6-20 子字符串转换为数值指令的有效操作数

输入/输出	数据类型	操作数
IN	STRING	IB、QB、VB、MB、SMB、SB、LB、*VD、*LD、*AC、常数
INDX	BYTE	VB、IB、QB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
OUT	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

对于整数和双整数
合法的输入字符串

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

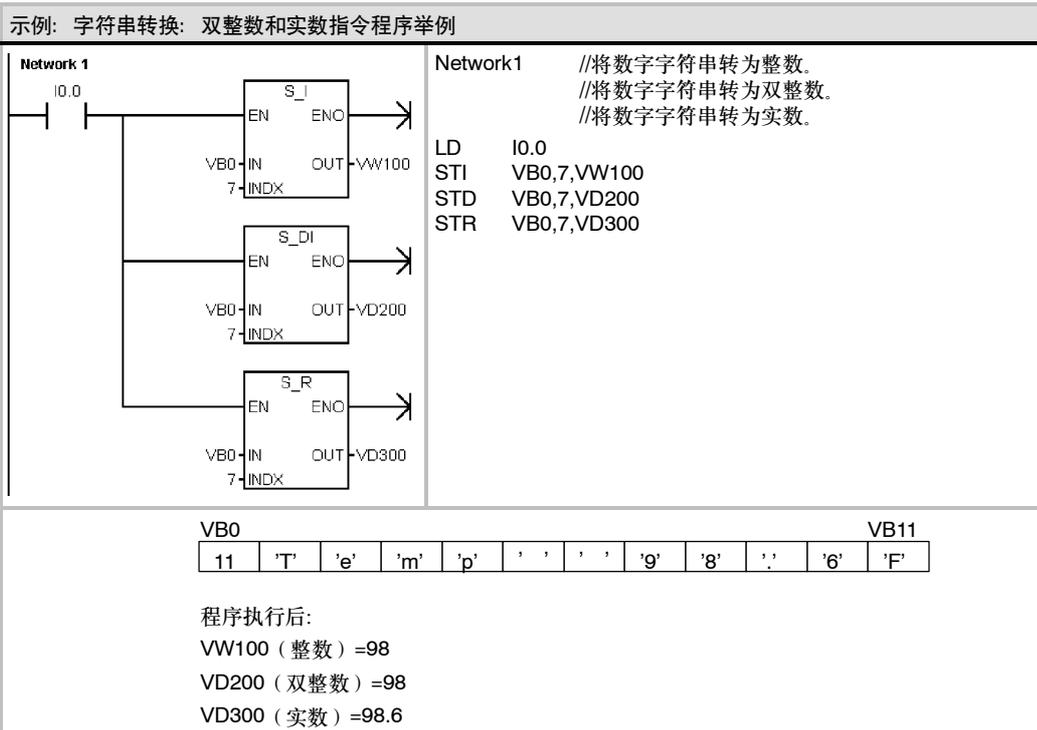
对于实数
合法的输入字符串

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

非法的输入字符串

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

图6-21 合法和非法的输入字符串的实例



编码和解码指令

编码

编码指令（ENCO）将输入字IN的最低有效位的位号写入输出字节OUT的最低有效“半字节”（4位）中。

译码

译码指令（DECO）根据输入字节（IN）的低四位所表示的位号置输出字（OUT）的相应位为1，输出字的所有其他位都清0。

SM标志位和ENO

对于编码和译码指令，下列条件影响ENO。

使ENO=0的错误条件:

- 0006（间接寻址）

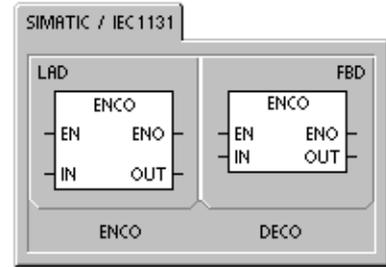


表6-21 编码和解码指令的有效操作数

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、常数 IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
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

示例: 解码和编码指令

Network 1

Network1 //AC2中包含错误检测位
//1. 译码指令将VW40中相应位置位。
//2. 编码指令将最低有效位转换为
// 错误代码，存入VB50中。

```
LD      I3.1
DECO   AC2, VW40
ENCO   AC3, VB50
```

AC2	3	AC3	<div style="display: flex; justify-content: space-between; font-size: 8px;"> 1590 </div> 1000 0010 0000 0000
VW40	<div style="display: flex; justify-content: space-between; font-size: 8px;"> 1530 </div> 0000 0000 0000 1000	VB50	<div style="display: flex; justify-content: space-between; font-size: 8px;"> 1590 </div> 9

计数器指令

SIMATIC计数器指令

增计数器

增计数指令（CTU）从当前计数值开始，在每一个（CU）输入状态从低到高时递增计数。当CXX的当前值大于等于预置值PV时，计数器位CXX置位。当复位端（R）接通或者执行复位指令后，计数器被复位。当它达到最大值（32, 767）后，计数器停止计数。

STL操作:

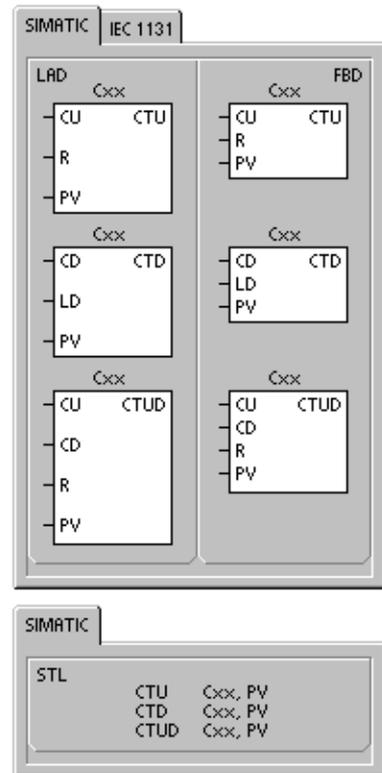
- 复位输入: 栈顶
- 计数输入: 其值被装载在第二个堆栈中。

减计数器

减计数指令（CTD）从当前计数值开始，在每一个（CD）输入状态的低到高时递减计数。当CXX的当前值等于0时，计数器位CXX置位。当装载输入端（LD）接通时，计数器位被复位，并将计数器的当前值设为预置值PV。当计数值到0时，计数器停止计数，计数器位CXX接通。

STL操作:

- 装载输入: 栈顶
- 计数输入: 其值被装载在第二个堆栈中。



增/减计数器

增/减计数指令（CTUD），在每一个增计数输入（CU）的低到高时增计数，在每一个减计数输入（CD）的低到高时减计数。计数器的当前值CXX保存当前计数值。在每一次计数器执行时，预置值PV与当前值作比较。

当达到最大值(32767)时，在增计数输入处的下一个上升沿导致当前计数值变为最小值(-32768)。当达到最小值(-32768)时，在减计数输入端的下一个上升沿导致当前计数值变为最大值(32767)。

当CXX的当前值大于等于预置值PV时，计数器位CXX置位。否则，计数器位关断。当复位端（R）接通或者执行复位指令后，计数器被复位。当达到预置值PV时，CTUD计数器停止计数。

STL操作:

- 复位输入: 栈顶
- 计数输入: 其值被装载在第二个堆栈中。
- 计数输入: 其值被装载在第二个堆栈中。

表6-22 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	常数 (C0到C255)
CU、CD、LD、R	BOOL	I、Q、V、M、SM、S、T、C、L、能流
PV	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数



提示

由于每一个计数器只有一个当前值，所以不要多次定义同一个计数器。（具有相同标号的增计数器、增/减计数器、减计数器访问相同的当前值。）

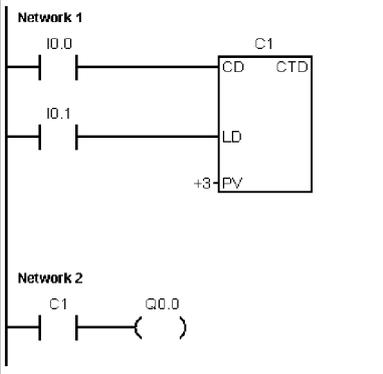
当使用复位指令复位计数器时，计数器位复位并且计数器当前值被清零。计数器标号既可以用来表示当前值，又可以用来表示计数器位。

表6-23 计数器指令的操作

定时器类型	操作	计数器位	上电周期/首次扫描
CTU	CU使当前值递增， 当前值持续递增直至32767。	当前值 >= 预设值时， 计数器位接通。	计数器位关断。 当前值可以保留。 ¹
CTUD	CU使当前值递增 CD使当前值递减 当前值持续递增或递减除非计数器被复位	当前值 >= 预设值时， 计数器位接通。	计数器位关断。 当前值可以保留。 ¹
CTD	CD使当前值递减直至当前值为0。	当前值 >= 预设值时， 当前值=0	计数器位关断。 当前值可以保留。 ¹

¹ 您可以选择计数器的当前值是否掉电保持。有关S7-200 CPU有记忆存储器的详细内容参阅第4章。

示例: SIMATIC减计数器指令

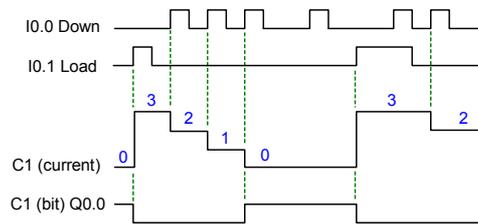


```

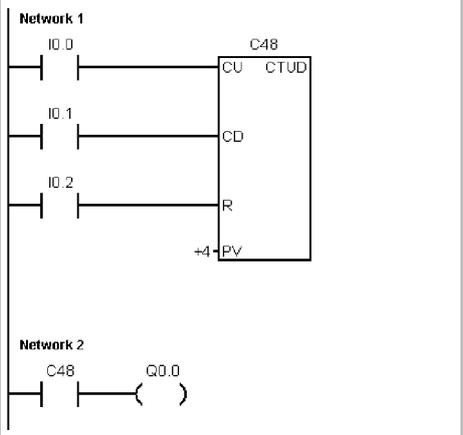
Network1 //当I0.1断开时,
//减计数器C1的当前
//值从3变到0, I0.0的上升沿使C1的
//当前值递减, I0.1接通时装载预置值3,
LD I0.0
LD I0.1
CTD C1, +3

Network 2 //当计数器C1的当前值=0时, C1接通。
LD C1
= Q0.0
    
```

时序图



示例: SIMATIC增/减计数器指令程序举例

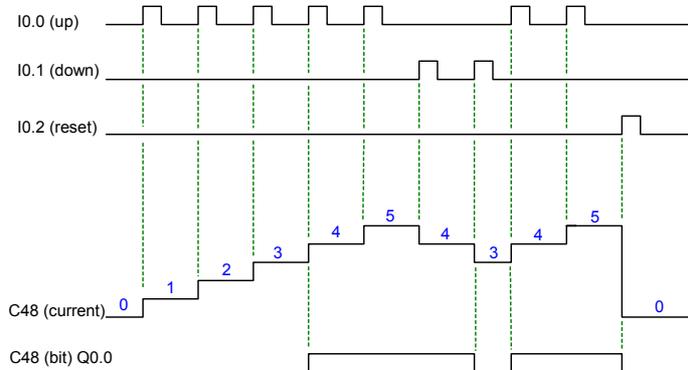


```

Network 1 // I0.0增计数
// I0.1减计数
// I0.2将当前值复位为0
LD I0.0
LD I0.1
LD I0.2
CTUD C48, +4

Network2 //当当前值=4时,
//将增/减计数器C48接通。
LD C48
= Q0.0
    
```

时序图



IEC计数器指令

增计数器

增计数指令（CTU）在每一个（CU）输入的上升沿从当前值开始增计数，直至预置值（PV）。当当前值（CV）大于等于预置值时，计数器输出位（Q）接通。当复位端（R）使能时，计数器复位。当计数到达预置值时，增计数器停止。

减计数器

减计数器指令（CTD）从预置值开始，在每一个（CD）输入的上升沿减计数。当当前值（CV）等于0时，计数器输出位（Q）接通。当装载输入（LD）使能时，计数器复位并且将计数器的当前值设为预置值PV。当计数值到0时，减计数器停止。

增/减计数器

增/减计数器指令（CTUD），在每一个增计数输入（CU）从低到高时增计数；在每一个减计数输入（CD）从低到高时减计数。当当前值等于预置值时，增计数输出（QU）接通。当当前值等于0时，减计数输出（QD）接通。当装载输入（LD）使能时，计数器将当前值设为预置值（PV）。类似的，当复位端（R）使能时，计数器复位并且当前值清0。当计数值达到预置值或者0时，计数器停止。

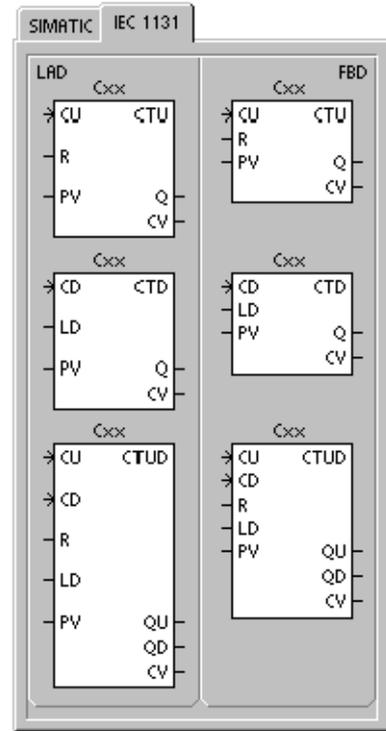


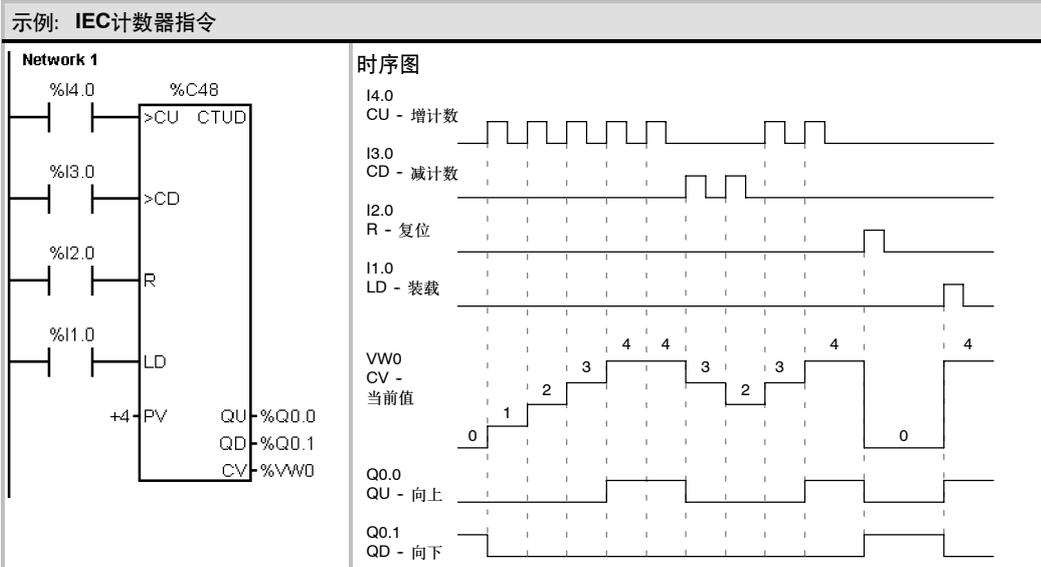
表6-24 IEC计数器指令的有效操作数

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

由于每一个计数器只有一个当前值，所以不要多次定义同一个计数器。（具有相同标号的增计数器、增/减计数器和减计数器访问相同的当前值。）



高速计数器指令

定义高速计数器

定义高速计数器指令（HDEF）为指定的高速计数器(HSCx)选择操作模式。模块的选择决定了高速计数器的时钟、方向、启动和复位功能。

对于每一个高速计数器使用一条定义高速计数器指令。

使**ENO=0**的错误条件:

- 0003（输入点冲突）
- 0004（中断中的非法指令）
- 000A（HSC重复定义）

高速计数器

高速计数器指令（HSC）在HSC特殊存储器位状态的基础上，配置和控制高速计数器。参数N指定高速计数器的标号。

高速计数器可以被配置为12种模式中的任意一种，参见表6-26。

每一个计数器都有时钟、方向控制、复位、启动的特定输入。对于双相计数器，两个时钟都可以运行在最高频率。在正交模式下，您可以选择一倍速（1x）或者四倍速（4x）计数速率。所有计数器都可以运行在最高频率下而互不影响。

使**ENO=0**的错误条件:

- 0001（在HDEF指令之前执行HSC指令）
- 0005（同时执行HSC/PLS）

表6-25 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	常数
N	WORD	常数



编程提示

可以参考资料光盘上应用程序中使用高速计数器的程序。参见应用示例4和应用示例29。

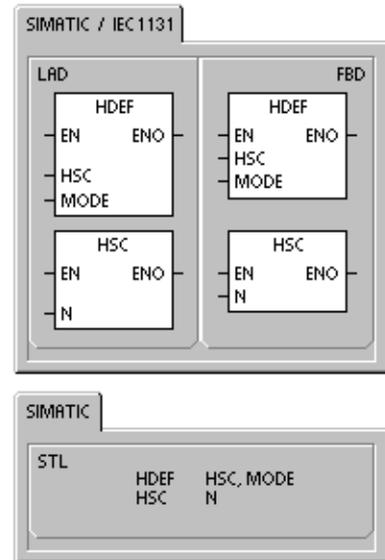
高速计数器用于对S7-200扫描速率无法控制的高速事件进行计数。高速计数器的最高计数频率取决于您的CPU类型。更多信息请参见附录A。



提示

CPU221和CPU222支持HSC0、HSC3、HSC4和HSC5。不支持HSC1和HSC2。

CPU224、CPU224XP和CPU 226全部支持六个高速计数器：HSC0到HSC5。



一般来说，高速计数器被用作驱动鼓式计时器，该设备有一个安装了增量轴式编码器的轴，以恒定的速度转动。轴式编码器每圈提供一个确定的计数值和一个复位脉冲。来自轴式编码器的时钟和复位脉冲作为高速计数器的输入。

高速计数器装入一组预置值中的第一个值，当前计数值小于当前预置值时，希望的输出有效。计数器设置成在当前值等于预置值和有复位时产生中断。

随着每次当前计数值等于预置值的中断事件的出现，一个新的预置值被装入，并重新设置下一个输出状态。当出现复位中断事件时，设置第一个预置值和第一个输出状态，这个循环又重新开始。

由于中断事件产生的速率远低于高速计数器的计数速率，用高速计数器可实现精确控制，而与PLC整个扫描周期的关系不大。采用中断的方法允许在简单的状态控制中用独立的中断程序装入一个新的预置值。（同样的，也可以在一个中断服务程序中，处理所有的中断事件。）

理解不同的高速计数器

对于操作模式相同的计数器，其计数功能是相同的。计数器共有四种基本类型：带有内部方向控制的单相计数器，带有外部方向控制的单相计数器，带有两个时钟输入的双相计数器和A/B相正交计数器。注意，并不是所有计数器都能使用每一种模式。您可以使用以下类型：无复位或启动输入，有复位无启动输入或既有启动又有复位输入。

- 当激活复位输入端时，计数器清除当前值并一直保持到复位端失效。
- 当激活启动输入端时，它允许计数器计数。当启动端失效时，计数器的当前值保持为常数，并且忽略时钟事件。
- 如果在启动输入端无效的同时，复位信号被激活，则忽略复位信号，当前值保持不变。如果在复位信号被激活的同时，启动输入端被激活，当前值被清除。

在使用高速计数器之前，应该用HDEF（高速计数器定义）指令为计数器选择一种计数模式。使用初次扫描存储器位SM0.1（该位仅在第一次扫描周期接通，之后断开）来调用一个包含HDEF指令的子程序。

高速计数器编程



指令向导

您可以使用指令向导来配置计数器。向导程序使用下列信息：计数器的类型和模式、计数器的预置值、计数器的初始值和计数的初始方向。要启动HSC指令向导，可以在命令菜单窗口中选择**Tools > Instruction Wizard**，然后在向导窗口中选择HSC指令。

对高速计数器编程，您必须完成下列基本操作：

- 定义计数器和模式
- 设置控制字节
- 设置初始值
- 设置预置值
- 指定并使能中断服务程序
- 激活高速计数器

定义计数器的模式和输入

使用高速计数器定义指令来定义计数器的模式和输入。

表6-26中给出了与高速计数器相关的时钟、方向控制、复位和启动输入点。同一个输入点不能用于两个不同的功能，但是任何一个没有被高速计数器的当前模式使用的输入点，都可以被用作其它用途。例如，如果HSC0正被用于模式1，它占用I0.0和I0.2，则I0.1可以被边缘中断或者HSC3占用。



提示

注意：HSC0的所有模式（模式12除外）总是使用I0.0，HSC4的所有模式总是使用I0.3，因此在使用这些计数器时，相应的输入点不能用于其它功能。

表6-26 高速计数器的输入点

模式	中断描述	输入点			
	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	带有内部方向控制的单相计数器	时钟			
1		时钟		复位	
2		时钟		复位	启动
3	带有外部方向控制的单相计数器	时钟	方向		
4		时钟	方向	复位	
5		时钟	方向	复位	启动
6	带有增减计数时钟的双相计数器	增时钟	减时钟		
7		增时钟	减时钟	复位	
8		增时钟	减时钟	复位	启动
9	A/B相正交计数器	时钟A	时钟B		
10		时钟A	时钟B	复位	
11		时钟A	时钟B	复位	启动
12	只有HSC0和HSC3支持模式12。 HSC0计数Q0.0输出的脉冲数。 HSC3计数Q0.1输出的脉冲数。				

HSC模式举例

图6-22到图6-26中给出了每种模式下计数器功能的时序图。

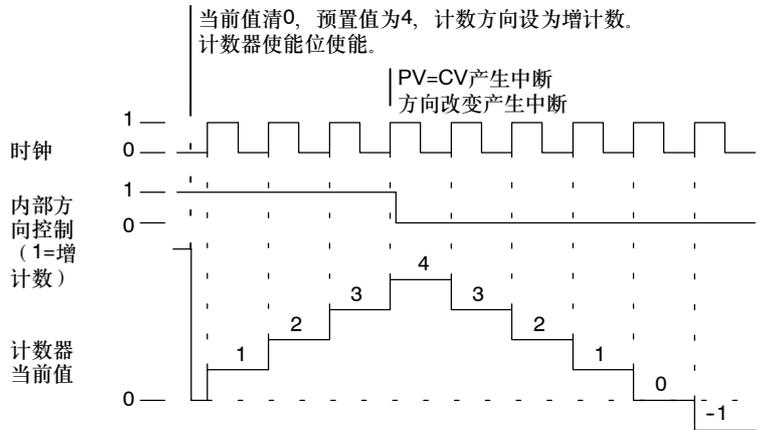


图6-22 模式0、1或2操作实例

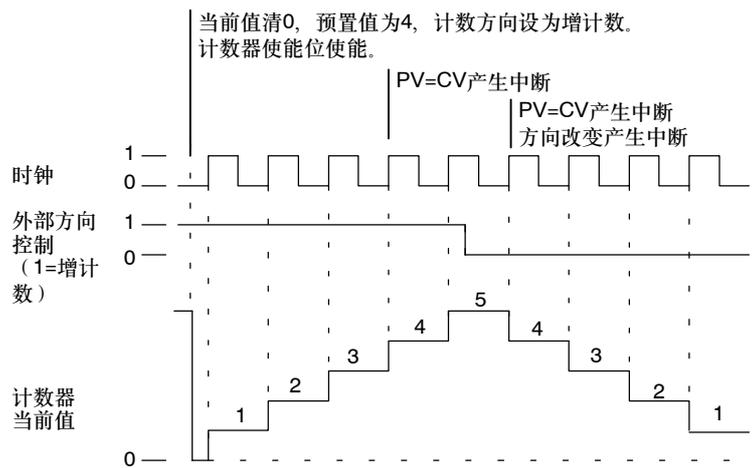


图6-23 模式3、4或5操作实例

当您使用模式6、7或者8时，如果增时钟输入的上升沿与减时钟输入的上升沿之间的时间间隔小于0.3毫秒，高速计数器会把这些事件看作是同时发生的。如果这种情况发生，当前值不变，计数方向指示不变。只要增时钟输入的上升沿与减时钟输入的上升沿之间的时间间隔大于0.3毫秒，高速计数器分别捕捉每个事件。在以上两种情况下，都不会有错误产生，计数器保持正确的当前值。

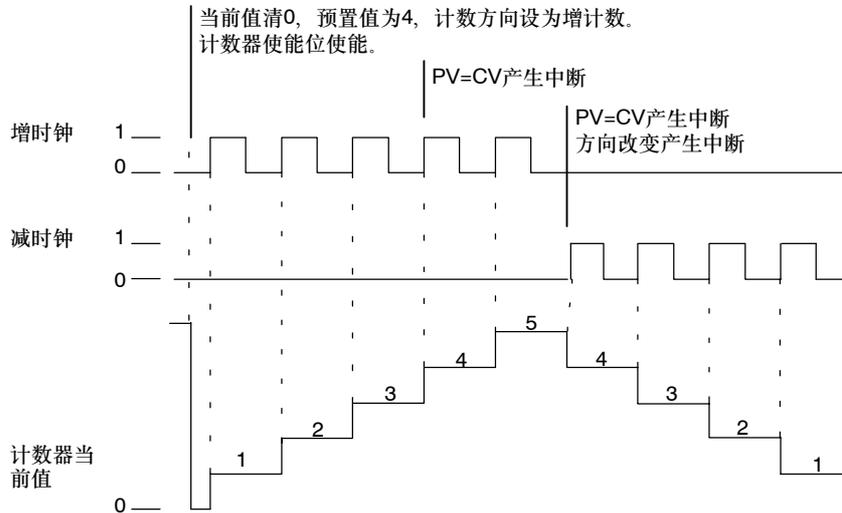


图6-24 模式6、7或8操作实例

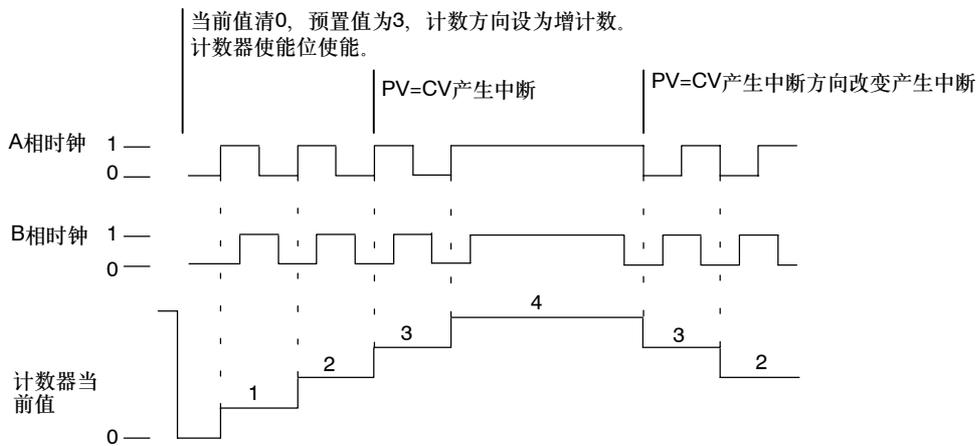


图6-25 模式9、10或者11操作实例（一倍速正交模式）

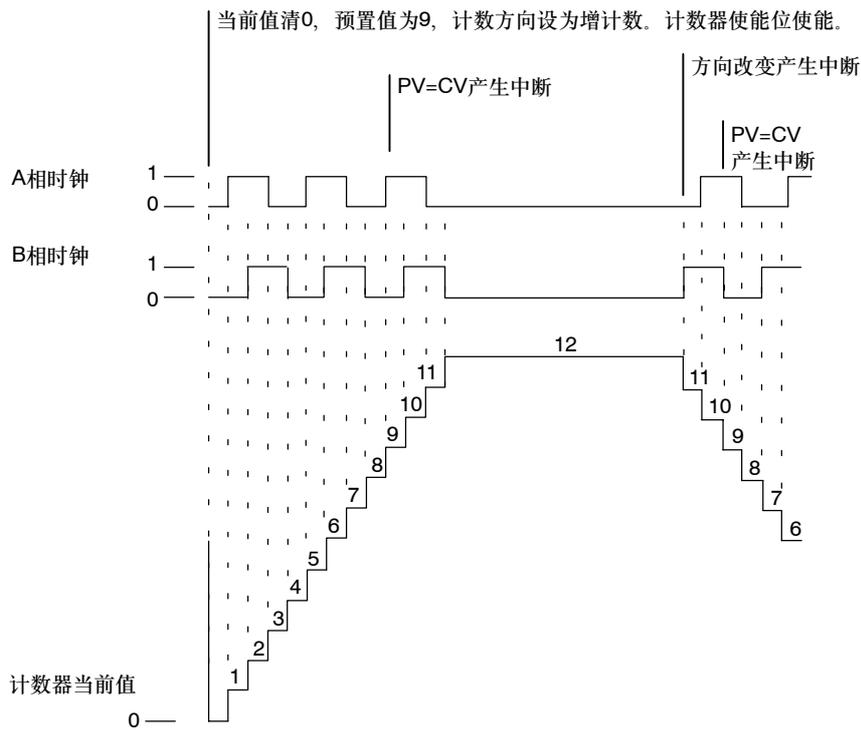


图6-26 模式9、10或者11操作实例（四倍速正交模式）

复位和启动操作

如图6-27中所示的复位和启动操作适用于使用复位和启动输入的所有模式。在复位和启动输入图中，复位输入和启动输入都被编程为高电平有效。

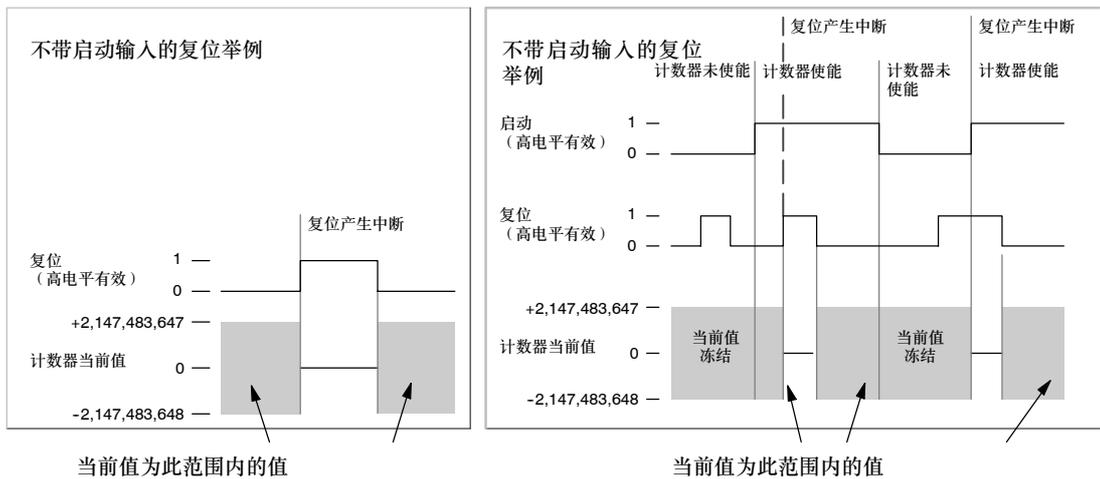


图6-27 带有或者不带启动输入的复位操作举例

对于高速计数器，有三个控制位用于配置复位和启动信号的有效状态以及选择一倍速或者四倍速计数模式（仅用于正交计数器）。这些位位于各个计数器的控制字节中并且只有在HDEF指令执行时使用。在表6-27中给出了这些位的定义。



提示

在执行HDEF指令前，必须把这些控制位设定到希望的状态。否则，计数器对计数模式的选择取缺省设置。

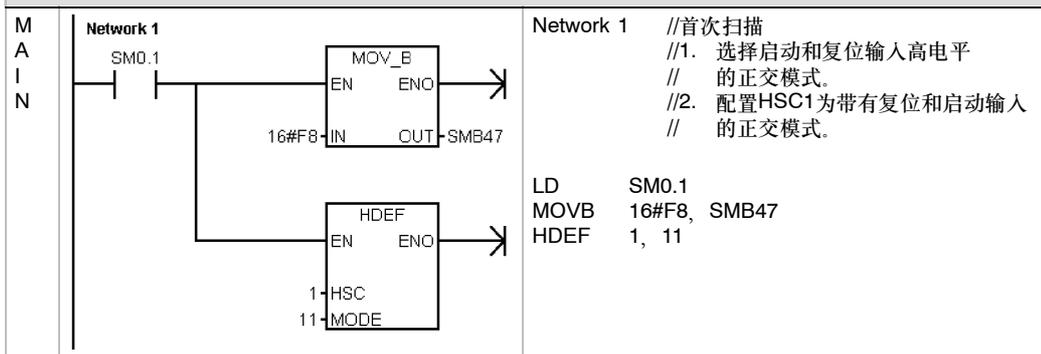
一旦HDEF指令被执行，您就不能再更改计数器的设置，除非先进入STOP模式。

表6-27 复位和启动输入的有效电平以及1x/4x控制位

HSC0	HSC1	HSC2	HSC4	描述（仅当HDEF执行时使用）
SM37.0	SM47.0	SM57.0	SM147.0	复位的有效控制位 ¹ 0=复位高电平有效； 1=复位低电平有效
-	SM47.1	SM57.1	-	启动有效电平控制位 ¹ ： 0=启动高电平有效； 1=启动低电平有效
SM37.2	SM47.2	SM57.2	SM147.2	正交计数器计数速率选择： 0=4X计数率； 1=1X计数率

¹ 缺省设置为：复位输入和启动输入高电平有效，正交计数率为四倍速（四倍输入时钟频率）。

示例：高速计数器定义指令



设置控制字节

只有定义了计数器和计数器模式，才能对计数器的动态参数进行编程。每个高速计数器都有一个控制字节，包括以下内容：

- 使能或者禁止计数器
- 控制计数方向（只对模式0、1和2有效）或者对所有其它模式定义初始化计数方向。
- 装载初始值
- 装载预置值

在执行HSC指令时，要检验控制字节和相关的初始值和预置值。表6-28中对这些控制位逐一做了说明。

表6-28 HSC0到HSC5的控制位

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	中断描述
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	计数方向控制位: 0=减计数; 1=增计数
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	向HSC中写入计数方向: 0=不更新; 1=更新 计数方向
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	向HSC中写入预置值: 0=不更新; 1=更新预置值
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	向HSC中写入新的初始值: 0=不更新; 1=更新初始值
SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	HSC允许: 0=禁止HSC; 1=允许HSC

设置初始值和预置值

每个高速计数器都有一个32位的初始值和一个32位的预置值。初始值和预置值都是符号整数。为了向高速计数器装入新的初始值和预置值，必须先设置控制字节，并且把初始值和预置值存入特殊存储器中，然后执行HSC指令，从而将新的值传送到高速计数器。表6-29中对保存新的初始值和预置值的特殊存储器作了说明。

除去控制字节和新的初始值与预置值保存字节外，每个高速计数器的当前值只能使用数据类型HC（高速计数器当前值）后面跟表6-29中列出的计数器号（0、1、2、3、4或5）的格式进行读取。可用读操作直接访问的当前值，但是写操作只能用HSC指令来实现。

表6-29 HSC0、HSC1、HSC2、HSC3、HSC4和HSC5的新当前值和新预设值

要装入的值	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
新初始值	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
新预置值	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

表6-30 HSC0、HSC1、HSC2、HSC3、HSC4和HSC5的当前值

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

高速计数器寻址 (HC)

如果要指定高速计数器的地址，访问高速计数器的计数值，要使用存储器类型HC和计数器号（例如HC0）。高速计数器的当前值是只读值，只能以双字（32位）分配地址，如图6-28所示。

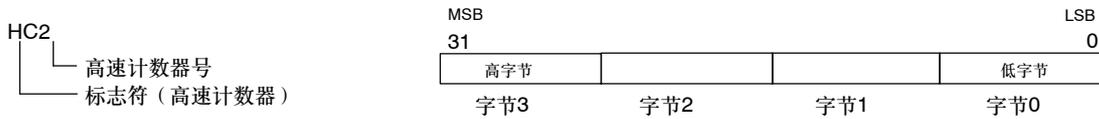


图6-28 访问高速计数器的当前值

指定中断

所有计数器模式都支持在HSC的当前值等于预设值时产生一个中断事件。使用外部复位端的计数模式支持外部复位中断。除去模式0、1和2之外，所有计数器模式支持计数方向改变中断。每种中断条件都可以分别使能或者禁止。要得到关于使用中断的更多信息，参见通讯和中断指令一节。

注释

当使用外部复位中断时，不要写入初始值，或者是在该中断服务程序中禁止再允许高速计数器，否则会产生一个致命错误。

状态字节

每个高速计数器都有一个状态字节，其中的状态存储位指出了当前计数方向，当前值是否大于或者等于预置值。表6-31给出了每个高速计数器状态位的定义。



提示

只有在执行中断服务程序时，状态位才有效。监视高速计数器状态的目的是使其它事件能够产生中断以完成更重要的操作。

表6-31 HSC0到HSC5的状态位

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	中断描述
SM36.0	SM46.0	SM56.0	SM136.0	SM46.0	SM156.0	不用
SM36.1	SM46.1	SM56.1	SM136.1	SM46.1	SM156.1	不用
SM36.2	SM46.2	SM36.2	SM136.2	SM46.2	SM156.2	不用
SM36.3	SM46.3	SM56.3	SM136.3	SM46.3	SM156.3	不用
SM36.4	SM46.4	SM56.4	SM136.4	SM46.4	SM156.4	不用
SM36.5	SM46.5	SM56.5	SM136.5	SM46.5	SM156.5	当前计数方向状态位: 0=减计数 1=增计数
SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	当前值等于预置值状态位: 0=不等; 1=相等
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	当前值大于预置值状态位: 0=小于等于; 1=大于

高速计数器的初始化步骤举例

以下以HSC1为例，对初始化和操作的步骤进行描述。在初始化描述中，假定S7-200已经置成RUN模式。因此，首次扫描标志位为真。如果不是这种情况，请记住在进入RUN模式之后，对每一个高速计数器的HDEF指令只能执行一次。对一个高速计数器第二次执行HDEF指令会引起运行错误，而且不能改变第一次执行HDEF指令时对计数器的设置。



提示

虽然下列步骤描述了如何分别改变计数方向、初始值和预置值，但您完全可以在同一操作步骤中对全部或者任意参数组合进行设置，只要设置正确的SMB47然后执行HSC指令即可。

初始化模式0、1或2

HSC1为内部方向控制的单相增/减计数器（模式0、1或2），初始化步骤如下：

1. 用初次扫描存储器位（SM0.1=1）调用执行初始化操作的子程序。由于采用了这样的子程序调用，后续扫描不会再调用这个子程序，从而减少了扫描时间，也提供了一个结构优化的程序。
2. 初始化子程序中，根据所希望的控制操作对SMB47置数。例如：
SMB47=16#F8 *产生如下的结果：*
 允许计数
 写入新的初始值
 写入新的预置值
 置计数方向为增
 置启动和复位输入为高电平有效
3. 执行HDEF指令时，HSC输入置1，MODE输入置0（无外部复位或启动）或置1（有外部复位和无启动）或置2（有外部复位和启动）。
4. 向SMD48（双字）写入所希望的初始值（若写入0，则清除）。
5. 向SMD52（双字）写入所希望的预置值。
6. 为了捕获当前值（CV）等于预置值（PV）中断事件，编写中断子程序，并指定CV=PV中断事件（事件号13）调用该中断子程序。参看本章中断一节，以了解中断处理的细节。
7. 为了捕获外部复位事件，编写中断子程序，并指定外部复位中断事件（事件号15）调用该中断子程序。
8. 执行全局中断允许指令（ENI）来允许HSC1中断。
9. 执行HSC指令，使S7-200对HSC1编程。
10. 退出子程序。

初始化模式3、4或5

HSC1为外部方向控制的单相增/减计数器（模式3、4或5），初始化步骤如下：

1. 用初次扫描存储器位（SM0.1=1）调用执行初始化操作的子程序。由于采用了这样的子程序调用，后续扫描不会再调用这个子程序，从而减少了扫描时间，也提供了一个结构优化的程序。
2. 初始化子程序中，根据所希望的控制操作对SMB47置数。例如：
SMB47=16#F8 *产生如下的结果：*
 允许计数
 写入新的初始值
 写入新的预置值
 置HSC的初始计数方向为增
 置启动和复位输入为高电平有效
3. 执行HDEF指令时，HSC输入置1，MODE输入置3（无外部复位或启动）或置4（有外部复位和无启动）或置5（有外部复位和启动）。
4. 向SMD48（双字）写入所希望的初始值（若写入0，则清除）。
5. 向SMD52（双字）写入所希望的预置值。
6. 为了捕获当前值（CV）等于预置值（PV）中断事件，编写中断子程序，并指定CV=PV中断事件（事件号13）调用该中断子程序。参看本章中断一节，以了解中断处理的细节。
7. 为了捕获计数方向改变中断事件，编写中断子程序，并指定计数方向改变中断事件（事件号14）调用该中断子程序。
8. 为了捕获外部复位事件，编写中断子程序，并指定外部复位中断事件（事件号15）调用该中断子程序。
9. 执行全局中断允许指令（ENI）来允许HSC1中断。
10. 执行HSC指令，使S7-200对HSC1编程。
11. 退出子程序。

初始化模式6、7或8

HSC1为具有增/减两种时钟的双相增/减计数器（模式6、7或8），初始化步骤如下：

1. 用初次扫描存储器位（SM0.1=1）调用执行初始化操作的子程序。由于采用了这样的子程序调用，后续扫描不会再调用这个子程序，从而减少了扫描时间，也提供了一个结构优化的程序。
2. 初始化子程序中，根据所希望的控制操作对SMB47置数。例如：
SMB47=16#F8 *产生如下的结果：*
 允许计数
 写入新的初始值
 写入新的预置值
 置HSC的初始计数方向为增
 置启动和复位输入为高电平有效
3. 执行HDEF指令时，HSC输入置1，MODE输入置6（无外部复位或启动）或置7（有外部复位和无启动）或置8（有外部复位和启动）。
4. 向SMD48（双字）写入所希望的初始值（若写入0，则清除）。
5. 向SMD52（双字）写入所希望的预置值。

6. 为了捕获当前值 (CV) 等于预置值 (PV) 中断事件, 编写中断子程序, 并指定CV=PV中断事件 (事件号13) 调用该中断子程序。参看本章中断一节, 以了解中断处理的细节。
7. 为了捕获计数方向改变中断事件, 编写中断子程序, 并指定计数方向改变中断事件 (事件号14) 调用该中断子程序。
8. 为了捕获外部复位事件, 编写中断子程序, 并指定外部复位中断事件 (事件号15) 调用该中断子程序。
9. 执行全局中断允许指令 (ENI) 来允许HSC1中断。
10. 执行HSC指令, 使S7-200对HSC1编程。
11. 退出子程序。

初始化模式9、10或11

HSC1为A/B相正交计数器 (模式9、10或11), 初始化步骤如下:

1. 用初次扫描存储器位 (SM0.1=1) 调用执行初始化操作的子程序。由于采用了这样的子程序调用, 后续扫描不会再调用这个子程序, 从而减少了扫描时间, 也提供了一个结构优化的程序。
2. 初始化子程序中, 根据所希望的控制操作对SMB47置数。

例如 (1X计数方式):
SMB47=16#FC

产生如下的结果:
允许计数
写入新的初始值
写入新的预置值
置HSC的初始计数方向为增
置启动和复位输入为高电平有效

例如 (4X计数方式):
SMB47=16#F8

产生如下的结果:
允许计数
写入新的初始值
写入新的预置值
置HSC的初始计数方向为增
置启动和复位输入为高电平有效

3. 执行HDEF指令时, HSC输入置1, MODE输入置9 (无外部复位或启动) 或置10 (有外部复位和无启动) 或置11 (有外部复位和启动)。
4. 向SMD48 (双字) 写入所希望的初始值 (若写入0, 则清除)。
5. 向SMD52 (双字) 写入所希望的预置值。
6. 为了捕获当前值 (CV) 等于预置值 (PV) 中断事件, 编写中断子程序, 并指定CV=PV中断事件 (事件号13) 调用该中断子程序。参见本章中断一节, 以了解中断处理的细节。
7. 为了捕获计数方向改变中断事件, 编写中断子程序, 并指定计数方向改变中断事件 (事件号14) 调用该中断子程序。
8. 为了捕获外部复位事件, 编写中断子程序, 并指定外部复位中断事件 (事件号15) 调用该中断子程序。
9. 执行全局中断允许指令 (ENI) 来允许HSC1中断。
10. 执行HSC指令, 使S7-200对HSC1编程。
11. 退出子程序。

初始化模式12

HSC0为PTO0产生的脉冲计数（模式12），初始化步骤如下：

1. 用初次扫描存储器位（SM0.1=1）调用执行初始化操作的子程序。由于采用了这样的子程序调用，后续扫描不会再调用这个子程序，从而减少了扫描时间，也提供了一个结构优化的程序。
2. 初始化子程序中，根据所希望的控制操作对SMB37置数。例如：

SMB37=16#F8	产生如下的结果：
	允许计数
	写入新的初始值
	写入新的预置值
	置计数方向为增
	置启动和复位输入为高电平有效
3. 执行HDEF指令时，HSC输入置0，MODE输入置12。
4. 向SMD38（双字）写入所希望的初始值（若写入0，则清除）。
5. 向SMD42（双字）写入所希望的预置值。
6. 为了捕获当前值（CV）等于预置值（PV）中断事件，编写中断子程序，并指定CV=PV中断事件（事件号13）调用该中断子程序。参看本章中断一节，以了解中断处理的细节。
7. 执行全局中断允许指令（ENI）来允许HSC1中断。
8. 执行HSC指令，使S7-200对HSC0编程。
9. 退出子程序。

改变模式0、1、2或12的计数方向

对具有内部方向（控制模式0、1、2或12）的单相计数器HSC1，改变其计数方向的步骤如下：

1. 向SMB47写入所需的计数方向：

SMB47=16#90	允许计数 置HSC计数方向为减。
SMB47=16#98	允许计数 置HSC计数方向为增
2. 执行HSC指令，使S7-200对HSC1编程。

写入新的初始值（任何模式下）

在改变初始值时，迫使计数器处于非工作状态，当计数器被禁止时，它既不及数也不产生中断。

以下步骤描述了如何改变HSC1的初始值（任何模式下）：

1. 向SMB47写入新的初始值的控制位：

SMB47=16#C0	允许计数
	写入新的初始值
2. 向SMD48（双字）写入所希望的初始值（若写入0，则清除）。
3. 执行HSC指令，使S7-200对HSC1编程。

写入新的预置值（任何模式下）

以下步骤描述了如何改变HSC1的预设值（任何模式）：

1. 向SMB47写入允许写入新的预置值的控制位：

SMB47=16#A0	允许计数
	写入新的预置值
2. 向SMD52（双字）写入所希望的预置值。
3. 执行HSC指令，使S7-200对HSC1编程。

禁止HSC（任何模式下）

以下步骤描述了如何禁止HSC1高速计数器（任何模式）：

1. 写入SMB47以禁止计数：

SMB47=16#00	禁止计数
-------------	------
2. 执行HSC指令，以禁止计数。

示例: 高速计数器指令		
M A I N	<p>Network 1</p>	<p>Network1 //在首次扫描, 调用SBR0</p> <pre>LD SM0.1 CALL SBR0</pre>
S B R O	<p>Network 1</p>	<p>Network 1 //在首次扫描, 配置HSC1: //1. 使能计数器。 // -写初始值。 // -写预置值。 // -设初始方向为增计数。 // -选择启动和复位输入高电平有效。 // -选择4倍速模式。 //2. 配置HSC1为带启动和复位输入的正交模式。 //3. 清除HSC1的初始值。 //4. 置HSC1的预置值为50。 //5. 当HSC1的当前值=预置值时, 执行INT_0。 //6. 全局中断允许。 //7. 执行HSC1</p> <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>
I N T O	<p>Network 1</p>	<p>Network1 //执行HSC1 //1. 清除HSC1的初始值。 //2. 选择写入新的初始值和HSC1使能。</p> <pre>LD SM0.0 MOVD +0, SMD48 MOVB 16#C0, SMB47 HSC 1</pre>

脉冲输出指令



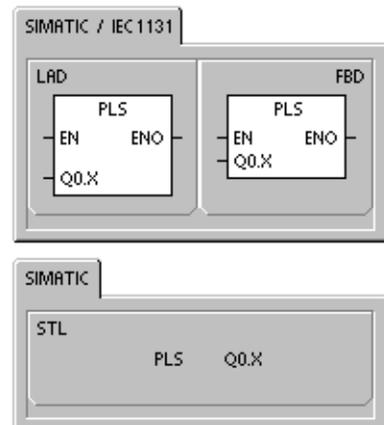
运动控制

脉冲输出指令（PLS）用于在高速输出（Q0.0和Q0.1）上控制脉冲串输出（PTO）和脉宽调制（PWM）功能。

改进的位控向导可以创建为您的应用程序定制的指令，这可以简化您的编程任务并充分利用S7-200 CPU的特有特性。关于位控向导的更多信息，参见9章。

可以继续使用旧的PLS指令创建您自己的运动应用，但是只有改进的位控向导创建的指令才支持PTO上的线性斜坡。

PTO可以输出一串脉冲（占空比50%），用户可以控制脉冲的周期和个数。



PWM可以输出连续的、占空比可调的脉冲串，用户可以控制脉冲的周期和脉宽。

S7-200有两个PTO/PWM发生器，它们可以产生一个高速脉冲串或者一个脉宽调制波形。一个发生器是数字输出点Q0.0，另一个发生器是数字输出点Q0.1。一个指定的特殊寄存器（SM）位置为每个发生器存储下列数据：一个控制字节（8位），一个计数值（32位无符号数）和一个周期或脉宽值（16位无符号数）。

PTO/PWM发生器与过程映像寄存器共用Q0.0和Q0.1。当在Q0.0或Q0.1上激活PTO或PWM功能时，PTO/PWM发生器对输出拥有控制权，同时普通输出点功能被禁止。输出波形不受过程映象区状态、输出点强制值或者立即输出指令执行的影响。当不使用PTO/PWM发生器功能时，对输出点的控制权交回到过程映象寄存器。过程映象寄存器决定输出波形的起始和结束状态，以高低电平产生波形的启动和结束。

表6-32 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	常数: 0 (=Q0.0) 或者 1 (=Q0.1)



提示

在使能PTO或者PWM操作之前，将Q0.0和Q0.1过程映像寄存器清0。

所有控制位、周期、脉宽和脉冲计数值的缺省值均为0。

PTO/PWM的输出负载至少为10%的额定负载，才能提供陡直的上升沿和下降沿。



编程提示

请参阅资料光盘上应用示例中使用PLS指令进行PTO/PWM操作的程序。见应用示例7、22、23、30和50。

脉冲串操作 (PTO)

PTO按照给定的脉冲个数和周期输出一串方波(占空比50%)。(见图6-29)PTO可以产生单段脉冲串或者多段脉冲串(使用脉冲包络)。可以指定脉冲数和周期(以微秒或毫秒为增加量):

- 脉冲个数: 1到4, 294, 967, 295
- 周期: 10 μ s到65,535 μ s或者2ms到65,535ms。

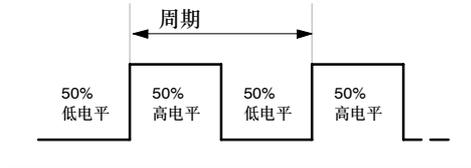


图6-29 脉冲串输出 (PTO)

如果为周期指定一个奇的微秒数或毫秒数(例如75ms),将会引起占空比失真。

表6-33中是对脉冲计数和周期的限定。

表6-33 PTO功能的脉冲个数及周期

脉冲个数/周期	结果
周期<2个时间单位	将周期缺省地设定为2个时间单位
脉冲个数=0	将脉冲个数缺省地设定为1个脉冲

PTO功能允许脉冲串“链接”或者“排队”。当当前脉冲串输出完成时,会立即开始输出一个新的脉冲串。这保证了多个输出脉冲串之间的连续性。

使用位控向导

位控向导自动处理PTO脉冲的单段管线和多段管线、脉宽调制、SM位置配置和创建包络表。这里是可供您参考的信息。建议您使用位控向导。关于位控向导的更多信息,参见第9章。

PTO脉冲串的单段管线

在单段管线模式,需要为下一个脉冲串更新特殊寄存器。一旦启动了起始PTO段,就必须按照第二个波形的要求改变特殊寄存器,并再次执行PLS指令。第二个脉冲串的属性在管线中一直保持到第一个脉冲串发送完成。在管线中一次只能存储一段脉冲串的属性。当第一个脉冲串发送完成时,接着输出第二个波形,此时管线可以用于下一个新的脉冲串。重复这个过程可以再次设定下一个脉冲串的特性。

除去以下两种情况之外,脉冲串之间可以作到平滑转换:时间基准发生了变化或者在利用PLS指令捕捉到新脉冲之前,启动的脉冲串已经完成。

PTO脉冲串的多段管线

在多段管线模式，CPU自动从V存储器区的包络表中读出每个脉冲串的特性。在该模式下，仅使用特殊存储器区的控制字节和状态字节。选择多段操作，必须装入包络表在V存储器中的起始地址偏移量（SMW168或SMW178）。时间基准可以选择微秒或者毫秒，但是，在包络表中的所有周期值必须使用同一个时间基准，而且在包络正在运行时不能改变。执行PLS指令来启动多段操作。

每段记录的长度为8个字节，由16位周期值、16位周期增量值和32位脉冲个数组成。表6-34中给出了包络表的格式。您可以通过编程的方式使脉冲的周期自动增减。在周期增量处输入一个正值将增加周期；输入一个负值将减少周期；输入0将不改变周期。

当PTO包络执行时，当前启动的段的编号保存在SMB166（或SMB176）。

表6-34 多段PTO操作的包络表格式

字节偏移量	包络段数	描述
0		段数1到255 ¹
1	#1	初始周期（2到65535时间基准单位）
3		每个脉冲的周期增量（有符号值）（-32768到32767时间基准单位）
5		脉冲数（1到4294967295）
9	#2	初始周期（2到65535时间基准单位）
11		每个脉冲的周期增量（有符号值）（-32768到32767时间基准单位）
13		脉冲数（1到4294967295）
（连续）	#3	（连续）

1 输入0作为脉冲串的段数会产生一个非致命错误，将不产生PTO输出。

脉宽调制（PWM）

PWM产生一个占空比变化周期固定的脉冲输出，（见图6-30）您可以以微秒或者毫秒为单位指定其周期和脉冲宽度：

- 周期: 10 μ s到65,535 μ s或者2ms到65,535ms。
- 脉宽: 0 μ s到65,535 μ s或者0ms到65,535ms。

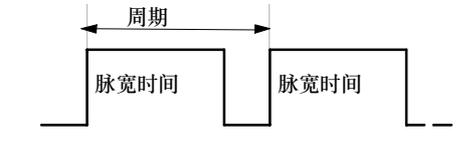


图6-30 脉宽调制（PWM）

如表6-35中所示，设定脉宽等于周期（使占空比为100%），输出连续接通。设定脉宽等于0（使占空比为0%），输出断开。

表6-35 脉宽、周期和PWM功能的执行结果。

脉宽/周期	结果
脉宽 ≥ 周期值	占空比为100%：输出连续接通。
脉宽=0	占空比为0%：输出断开。
周期 < 2个时间单位	将周期缺省地设定为2个时间单位。

有两个方法改变PWM波形的特性:

- 同步更新: 如果不需要改变时间基准, 就可以进行同步更新。利用同步更新, 波形特性的变化发生在周期边沿, 提供平滑转换。
- 异步更新: PWM的典型操作是当周期时间保持常数时变化脉冲宽度。所以, 不需要改变时间基准。但是, 如果需要改变PTO/PWM发生器的时间基准, 就要使用异步更新。异步更新会造成PTO/PWM功能被瞬时禁止, 和PWM波形不同步。这会引起被控设备的振动。由于这个原因, 建议采用PWM同步更新。选择一个适合于所有周期时间的时间基准。



提示

控制字节中的PWM更新方式位 (SM67.4或SM77.4) 用于指定更新方式。当PLS指令执行时变化生效。

如果改变了时间基准, 会产生一个异步更新, 而与PWM更新方式位的状态无关。

使用SM来配置和控制PTO/PWM操作

PLS指令会从特殊存储器SM中读取数据, 使程序按照其存储值控制PTO/PWM发生器。SMB67控制PTO0或者PWM0, SMB77控制PTO1或者PWM1。表6-36对用于控制PTO/PWM操作的存储器给出了描述。您可以使用表6-37作为一个快速参考, 用其中的数值作为PTO/PWM控制寄存器的值来实现需要的操作。

您可以通过修改SM存储区 (包括控制字节), 然后执行PLS指令来改变PTO或PWM波形的特性。您可以在任意时刻禁止PTO或者PWM波形, 方法为: 首先将控制字节中的使能位 (SM67.7或者SM77.7) 清0, 然后执行PLS指令。

PTO状态字节中的空闲位 (SM66.7或者SM76.7) 标志着脉冲串输出完成。另外, 在脉冲串输出完成时, 您可以执行一段中断服务程序。(参考中断指令和通讯指令中的描述)。如果您使用多段操作, 可以在整个包络表完成之后执行中断服务程序。

下列条件使SM66.4 (或SM76.4) 或SM66.5 (或SM76.5) 置位:

- 如果周期增量使PTO在许多脉冲后产生非法周期值, 会产生一个算术溢出错误, 这会终止PTO功能并在状态字节中将增量计算错误位 (SM66.4或者SM76.4) 置1, PLC的输出变为由映象寄存器控制。
- 如果要手动终止一个正在进行中的PTO包络, 要把状态字节中的用户终止位 (SM66.5或SM76.5) 置1。
- 当管线满时, 如果试图装载管线, 状态存储器中的PTO溢出位 (SM66.6或者SM76.6) 置1。如果想用该位检测序列的溢出, 必须在检测到溢出后手动清除该位。当CPU切换至RUN模式时, 该位被初始化为0。



提示

如果要装入新的脉冲数 (SMD72或SMD82)、脉冲宽度 (SMW70或SMW80) 或周期 (SMW68或SMW78), 应该在执行PLS指令前装入这些值和控制寄存器。如果要使用多段脉冲串操作, 在使用PLS指令前也需要装入包络表的起始偏移量 (SMW168或SMW178) 和包络表的值。

表6-36 PTO/PWM控制寄存器的SM标志

Q0.0	Q0.1	状态字节	
SM66.4	SM76.4	PTO包络由于增量计算错误而终止	0=无错误; 1=终止
SM66.5	SM76.5	PTO包络由于用户命令而终止	0=无错误; 1=终止
SM66.6	SM76.6	PTO管线上溢/下溢	0=无溢出; 1=上溢/下溢
SM66.7	SM76.7	PTO空闲	0=执行中; 1=PTO空闲
Q0.0	Q0.1	控制字节	
SM67.0	SM77.0	PTO/PWM更新周期值	0=不更新; 1=更新周期值
SM67.1	SM77.1	PWM更新脉冲宽度值	0=不更新; 1=脉冲宽度值
SM67.2	SM77.2	PTO更新脉冲数	0=不更新; 1=更新脉冲数
SM67.3	SM77.3	PTO/PWM时间基准选择	0=1μs/格, 1=1ms/格
SM67.4	SM77.4	PWM更新方法:	0=异步更新; 1=同步更新
SM67.5	SM77.5	PTO操作:	0=单段操作; 1=多段操作
SM67.6	SM77.6	PTO/PWM模式选择	0=选择PTO; 1=选择PWM
SM67.7	SM77.7	PTO/PWM允许	0=禁止; 1=允许
Q0.0	Q0.1	其它PTO/PWM寄存器	
SMW68	SMW78	PTO/PWM周期值 (范围: 2到65535)	
SMW70	SMW80	PWM脉冲宽度值 (范围: 0到65535)	
SMD72	SMD82	PTO脉冲计数值 (范围: 1到4, 294, 967, 295)	
SMB166	SMB176	进行中的段数 (仅用在多段PTO操作中)	
SMW168	SMW178	包络表的起始位置, 用从V0开始的字节偏移表示 (仅用在多段PTO操作中)	
SMB170	SMB180	线性包络状态字节	
SMB171	SMB181	线性包络结果寄存器	
SMD172	SMD182	手动模式频率寄存器	

表6-37 PTO/PWM控制字节参考

控制寄存器 (16进制)	执行PLS指令的结果							
	允许	模式选择	PTO段操作	PWM更新方法	时基	脉冲数	脉冲宽度	周期
16#81	Yes	PTO	单段		1μs/周期			装入
16#84	Yes	PTO	单段		1μs/周期	装入		
16#85	Yes	PTO	单段		1μs/周期	装入		装入
16#89	Yes	PTO	单段		1ms/周期			装入
16#8C	Yes	PTO	单段		1ms/周期	装入		
16#8D	Yes	PTO	单段		1ms/周期	装入		装入
16#A0	Yes	PTO	多段		1μs/周期			
16#A8	Yes	PTO	多段		1ms/周期			
16#D1	Yes	PWM		同步	1μs/周期			装入
16#D2	Yes	PWM		同步	1μs/周期		装入	
16#D3	Yes	PWM		同步	1μs/周期		装入	装入
16#D9	Yes	PWM		同步	1ms/周期			装入
16#DA	Yes	PWM		同步	1ms/周期		装入	
16#DB	Yes	PWM		同步	1ms/周期		装入	装入

计算包络表的值

PTO/PWM发生器的多段管线功能在许多应用中非常有用，尤其在步进电机控制中。

例如：您可以用带有脉冲包络的PTO来控制一台步进电机，来实现一个简单的加速、匀速和减速过程或者一个由最多255段脉冲包络组成的复杂过程，而其中每一段包络都是加速、匀速或者减速操作。

图6-31中的示例给出的包络表值要求产生一个输出波形包括三段：步进电机加速（第一段）；步进电机匀速（第二段）和步进电机减速（第三段）。

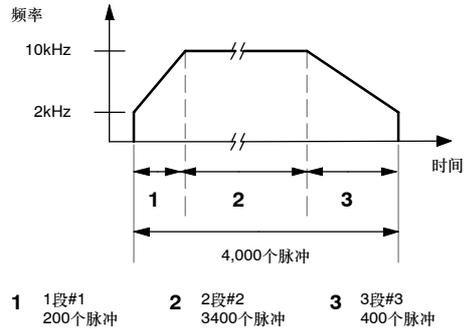


图6-31 频率/时间图

对该例，假定需要4000个脉冲达到要求的电机转动数，启动和结束频率是2kHz，最大脉冲频率是10kHz。由于包络表中的值是用周期表示的，而不是用频率，需要把给定的频率值转换成周期值。所以，启动和结束的脉冲周期为500 μs，最高频率的对应周期为100 μs。在输出包络的加速部分，要求在200个脉冲左右达到最大脉冲频率。也假定包络的减速部分，在400个脉冲完成。

在该例中，使用一个简单公式计算PTO/PWM发生器用来调整每个脉冲周期所使用的周期增量值：

$$De \text{ 给定段的周期增量} = | ECT - ICT | / Q$$

其中： ECT=该段结束周期时间
ICT=该段初始化周期时间
Q=该段的脉冲数量

利用这个公式，

加速部分（第1段）
的周期增量是-2。

由于第2段是恒速控制，因此，
该段的周期增量是0。

相似地，减速部分（第3段）
的周期增量是1。

假定包络表存放在从VB500开始的V存储器区，表6-38给出了产生所要求波形的值。该表的值可以在用户程序中用指令放在V存储器中。一种方法是在数据块中定义包络表的值。

表6-38 包络表值

V存储器地址	值	中断描述	
VB500	3	总段数	
VW501	500	初始周期	段#1
VW503	-2	周期增量	
VD505	200	脉冲数	
VW509	100	初始周期	段#2
VW511	0	周期增量	
VD513	3400	脉冲数	
VW517	100	初始周期	段#3
VW519	1	周期增量	
VD521	400	脉冲数	

段的最后一个脉冲的周期在包络中不直接指定，但必须计算出来（除非周期增量是0）。如果在段之间需要平滑转换，知道段的最后一个脉冲的周期是有用的。计算段的最后一个脉冲周期的公式是：

段的最后一个脉冲的周期时间= $ICT + (DEL * (Q - 1))$

其中： ICT=该段的初始化周期时间

DEL=该段的增量周期时间

Q=该段的脉冲数量

作为介绍，上面的简例是有用的，实际应用可能需要更复杂的波形包络。记住：周期增量只能以微秒数或毫秒数指定，周期的修改在每个脉冲上进行

这两项的影响使对于一个段的周期增量的计算可能需要叠代方法。对于结束周期值或给定段的脉冲个数，可能需要作调整。

在确定校正包络表值的过程中，包络段的持续时间很有用。按照下面的公式可以计算完成一个包络段的时间长短：

包络段的持续时间= $Q * (ICT + ((DEL/2) * (Q - 1)))$

其中： Q=该段的脉冲数量

ICT=该段的初始化周期时间

DEL=该段的增量周期时间

数字运算指令

加、减、乘、除指令

加法	减法	
IN1+IN2=OUT	IN1-IN2=OUT	LAD 和
FBD		
IN1+OUT=OUT	OUT-IN1=OUT	STL

整数加法 (+I) 或者整数减法 (-I) 指令, 将两个16位整数相加或者相减, 产生一个16位结果。双整数加法 (+D) 或者双整数减法 (-D) 指令, 将两个32位整数相加或者相减, 产生一个32位结果。实数加法 (+R) 和实数减法 (-R) 指令, 将两个32位实数相加或相减, 产生一个32位实数结果。

乘法	除法	
IN1*IN2=OUT	IN1/IN2=OUT	LAD 和
FBD		
IN1*OUT=OUT	OUT/IN1=OUT	STL

整数乘法 (*I) 或者整数除法 (/I) 指令, 将两个16位整数相乘或者相除, 产生一个16位结果。(对于除法, 余数不被保留。) 双整数乘法 (*D) 或者双整数除法 (/D) 指令, 将两个32位整数相乘或者相除, 产生一个32位结果。(对于除法, 余数不被保留。) 实数乘法 (*R) 或实数除法 (/R) 指令, 将两个32位实数相乘或相除, 产生一个32位实数结果。

SM标志位和ENO

SM1.1表示溢出错误和非法值。如果SM1.1置位, SM1.0和SM1.2的状态不再有效而且原始输入操作数不会发生变化。如果SM1.1和SM1.3没有置位, 那么数字运算产生一个有效的结果, 同时SM1.0和SM1.2有效。在除法运算中, 如果SM1.3置位, 其它数学运算标志位不会发生变化。

使ENO=0的错误条件:

- SM1.1 (溢出)
- SM1.3 (被0除)
- 0006 (间接寻址)

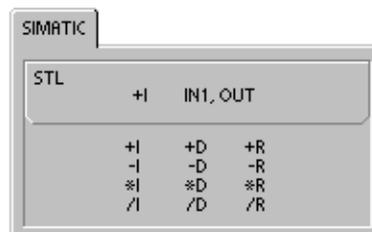
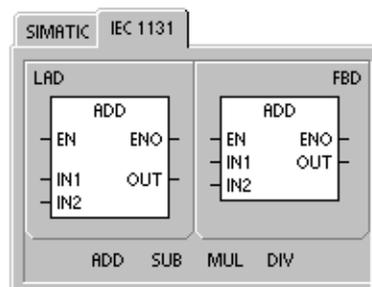
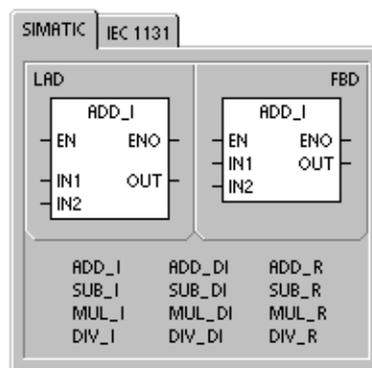
受影响的特殊存储器位:

- SM1.0 (结果为0)
- SM1.1 (溢出, 运算过程中产生非法数值或者输入参数非法)
- SM1.2 (结果为负)
- SM1.3 (被0除)

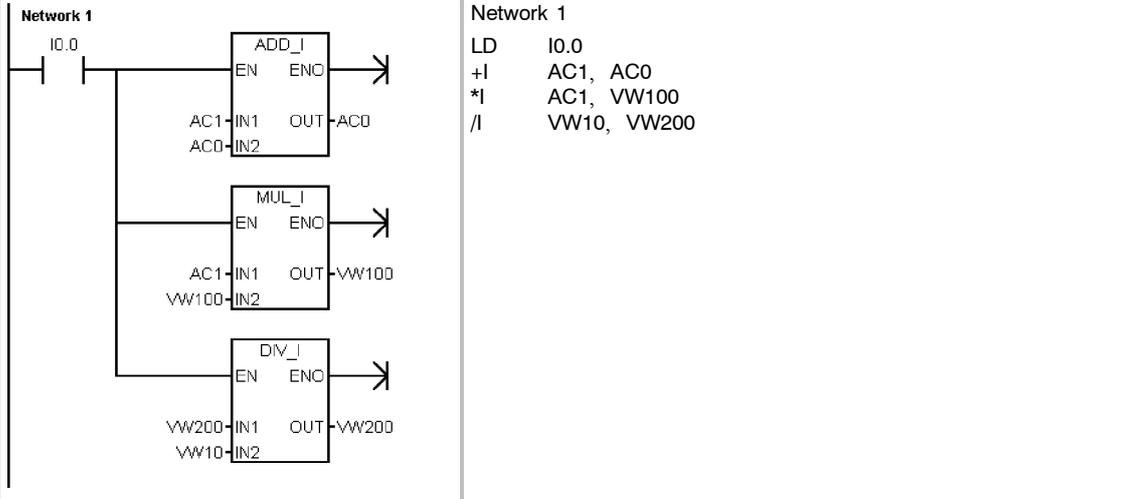
表6-39 加、减、乘、除指令的有效操作数

输入/输出	数据类型	操作数
IN1, IN2	INT DINT REAL	IIW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*AC、*LD、常数 ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数 ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常数
OUT	INT DINT、REAL	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、*VD、*AC、*LD ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

实数 (或者浮点数) 的表示格式采用ANSI/IEEE 754-1985标准 (单精度)。要得到更多信息请参考该标准。

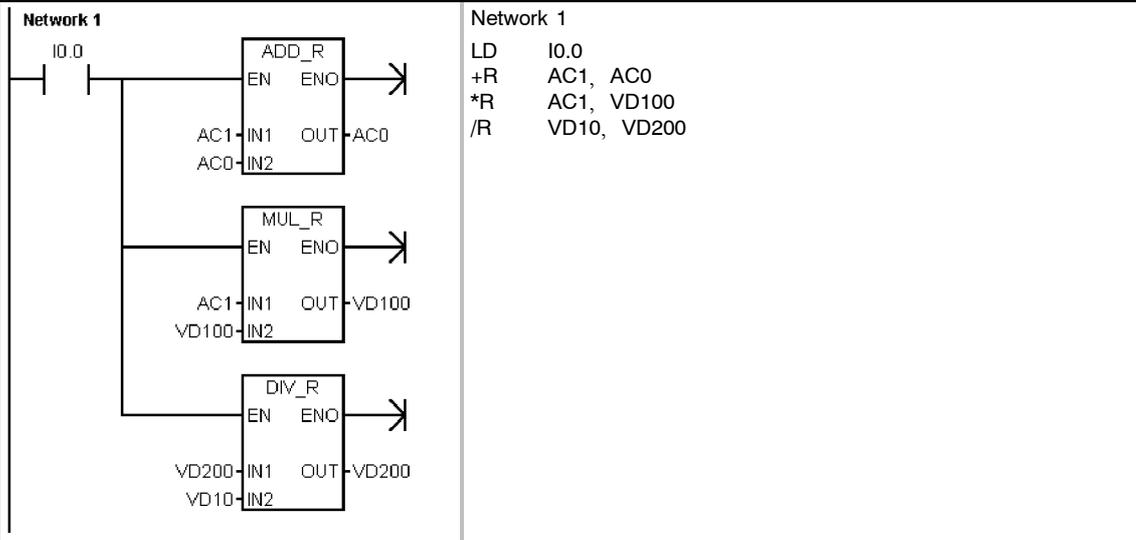


示例: 整数运算指令



<p>加法</p> <table style="margin: auto; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">40</td> <td style="padding: 0 5px;">+</td> <td style="border: 1px solid black; padding: 2px 10px;">60</td> <td style="padding: 0 5px;">=</td> <td style="border: 1px solid black; padding: 2px 10px;">100</td> </tr> <tr> <td style="text-align: center; font-size: small;">AC1</td> <td></td> <td style="text-align: center; font-size: small;">AC0</td> <td></td> <td style="text-align: center; font-size: small;">AC0</td> </tr> </table>	40	+	60	=	100	AC1		AC0		AC0	<p>乘法</p> <table style="margin: auto; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">40</td> <td style="padding: 0 5px;">*</td> <td style="border: 1px solid black; padding: 2px 10px;">20</td> <td style="padding: 0 5px;">=</td> <td style="border: 1px solid black; padding: 2px 10px;">800</td> </tr> <tr> <td style="text-align: center; font-size: small;">AC1</td> <td></td> <td style="text-align: center; font-size: small;">VW100</td> <td></td> <td style="text-align: center; font-size: small;">VW100</td> </tr> </table>	40	*	20	=	800	AC1		VW100		VW100	<p>除法</p> <table style="margin: auto; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">4000</td> <td style="padding: 0 5px;">/</td> <td style="border: 1px solid black; padding: 2px 10px;">40</td> <td style="padding: 0 5px;">=</td> <td style="border: 1px solid black; padding: 2px 10px;">100</td> </tr> <tr> <td style="text-align: center; font-size: small;">VW200</td> <td></td> <td style="text-align: center; font-size: small;">VW10</td> <td></td> <td style="text-align: center; font-size: small;">VW200</td> </tr> </table>	4000	/	40	=	100	VW200		VW10		VW200
40	+	60	=	100																												
AC1		AC0		AC0																												
40	*	20	=	800																												
AC1		VW100		VW100																												
4000	/	40	=	100																												
VW200		VW10		VW200																												

示例: 实数运算指令



<p>加法</p> <table style="margin: auto; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">4000.0</td> <td style="padding: 0 5px;">+</td> <td style="border: 1px solid black; padding: 2px 10px;">6000.0</td> <td style="padding: 0 5px;">=</td> <td style="border: 1px solid black; padding: 2px 10px;">10000.0</td> </tr> <tr> <td style="text-align: center; font-size: small;">AC1</td> <td></td> <td style="text-align: center; font-size: small;">AC0</td> <td></td> <td style="text-align: center; font-size: small;">AC0</td> </tr> </table>	4000.0	+	6000.0	=	10000.0	AC1		AC0		AC0	<p>乘法</p> <table style="margin: auto; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">400.0</td> <td style="padding: 0 5px;">*</td> <td style="border: 1px solid black; padding: 2px 10px;">200.0</td> <td style="padding: 0 5px;">=</td> <td style="border: 1px solid black; padding: 2px 10px;">80000.0</td> </tr> <tr> <td style="text-align: center; font-size: small;">AC1</td> <td></td> <td style="text-align: center; font-size: small;">VD100</td> <td></td> <td style="text-align: center; font-size: small;">VD100</td> </tr> </table>	400.0	*	200.0	=	80000.0	AC1		VD100		VD100	<p>除法</p> <table style="margin: auto; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px 10px;">4000.0</td> <td style="padding: 0 5px;">/</td> <td style="border: 1px solid black; padding: 2px 10px;">41.0</td> <td style="padding: 0 5px;">=</td> <td style="border: 1px solid black; padding: 2px 10px;">97.5609</td> </tr> <tr> <td style="text-align: center; font-size: small;">VD200</td> <td></td> <td style="text-align: center; font-size: small;">VD10</td> <td></td> <td style="text-align: center; font-size: small;">VD200</td> </tr> </table>	4000.0	/	41.0	=	97.5609	VD200		VD10		VD200
4000.0	+	6000.0	=	10000.0																												
AC1		AC0		AC0																												
400.0	*	200.0	=	80000.0																												
AC1		VD100		VD100																												
4000.0	/	41.0	=	97.5609																												
VD200		VD10		VD200																												

整数乘法产生双整数和带余数的整数除法

整数乘法产生双整数

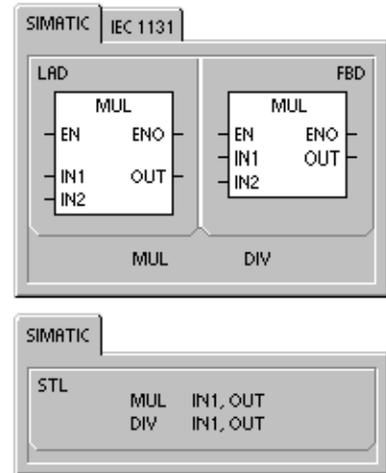
IN1 * IN2 = OUT LAD 和FBD
 IN1 * OUT = OUT STL

整数乘法产生双整数指令 (MUL)，将两个16位整数相乘，得到32位结果。在STL的MUL指令中，OUT的低16位被用作一个乘数。

带余数的整数除法

IN1 / IN2 = OUT LAD 和FBD
 OUT / IN1 = OUT STL

带余数的整数除法指令 (DIV)，将两个16位整数相除，得到32位结果。其中16位为余数（高16位字中），另外16位为商（低16位字中）。



在STL的DIV指令中，OUT的低16位被用作除数。

SM标志位和ENO

对于在本页中介绍的两条指令，特殊存储器 (SM) 标志位表示错误和非法值。如果在除法指令执行时，SM1.3置位，其它数字运算标志位不会发生变化。否则，当数字运算完成时，所有支持的数字运算状态位都包含有效状态。

使ENO=0的错误条件:

- SM1.1 (溢出)
- SM1.3 (被0除)
- 0006 (间接寻址)

受影响的特殊存储器位:

- SM1.0 (结果为0)
- SM1.1 (溢出)
- SM1.2 (结果为负)
- SM1.3 (被0除)

表6-40 整数乘法产生双整数和带余数的整数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、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
OUT	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

示例: 整数乘法产生双整数和带余数的整数除法指令程序举例

<p>Network 1</p>	<p>Network 1</p> <pre>LD I0.0 MUL AC1, VD100 DIV VW10, VD200</pre> <p>整数乘法产生双整数 400 * 200 = 80000 AC1 VW102 VD100</p> <p>带余数的整数除法 4000 / 41 = 23 97 VW202 VW10 余数 商 VW200 VW202 VD200</p>
-------------------------	--

注意: VD100包含: VW100和VW102, 而VD200包含: VW200和VW202。

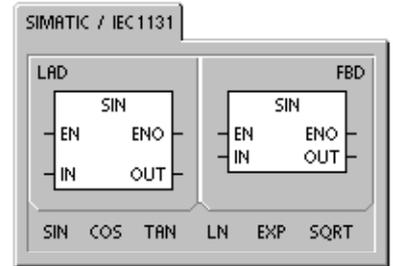
数学功能指令

正弦、余弦和正切

正弦 (SIN)、余弦 (COS) 和正切 (TAN) 指令计算角度值IN的三角函数值，并将结果存放在OUT中。输入角度值是弧度值。

$$\text{SIN}(\text{IN}) = \text{OUT} \quad \text{COS}(\text{IN}) = \text{OUT} \quad \text{TAN}(\text{IN}) = \text{OUT}$$

要将角度从度数变为弧度，可以使用MUL_R (*R) 指令，将度数乘以 $1.745329\text{E}-2$ (接近 $\pi/180$) 即可。

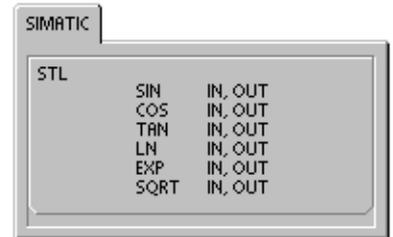


自然对数和自然指数

自然对数指令 (LN) 计算输入值IN的自然对数，并将结果存放到OUT中。

自然指数指令 (EXP) 计算输入值IN的自然指数值，并将结果存放到OUT中。

$$\text{LN}(\text{IN}) = \text{OUT} \quad \text{EXP}(\text{IN}) = \text{OUT}$$



要从自然对数计算出以10为底的对数值，可以使用除法指令，将自然对数值除以2.302585 (接近10的自然对数) 即可。

要计算任意实数的任意实数次方，包括分数形式的指数，需要将自然对数指令和自然指数指令结合在一起使用。例如：要计算X的Y次方，使用以下公式： $\text{EXP}(Y * \text{LN}(X))$ 。

平方根

平方根指令 (SQRT) 计算实数 (IN) 的平方根，并将结果存放到OUT中。

$$\text{SQRT}(\text{IN}) = \text{OUT}$$

如果要求其它次数的方根值：

$$5 \text{ 的立方} = 5^3 = \text{EXP}(3 * \text{LN}(5)) = 125$$

$$125 \text{ 的立方根} = 125^{(1/3)} = \text{EXP}((1/3) * \text{LN}(125)) = 5$$

$$5 \text{ 的立方的平方根} = 5^{(3/2)} = \text{EXP}(3/2 * \text{LN}(5)) = 11.18034$$

数学功能指令的SM位和ENO

对于本页中描述的所有指令，SM1.1用来表示溢出错误或者非法的数值。如果SM1.1置位，SM1.0和SM1.2的状态不再有效而且原始输入操作数不会发生变化。如果SM1.1没有置位，那么数字运算产生一个有效的结果，同时SM1.0和SM1.2状态有效。

使ENO=0的错误条件：

- SM1.1 (溢出)
- 0006 (间接寻址)

受影响的特殊存储器位：

- SM1.0 (结果为0)
- SM1.1 (溢出)
- SM1.2 (结果为负)

表6-41 数学功能指令的有效操作数

输入/输出	数据类型	操作数
IN	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常数
OUT	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

实数 (或者浮点数) 的表示格式采用ANSI/IEEE 754-1985标准 (单精度)。要得到更多信息请参考该标准。

递增和递减指令

递增

IN+1=OUT *LAD 和 FBD*
 OUT+1=OUT *STL*

递减

IN - 1=OUT *LAD 和 FBD*
 OUT - 1=OUT *STL*

递增或者递减指令将输入IN加1或者减1，并将结果存放在OUT中。

字节递增（INCB）和字节递减（DECB）操作是无符号的。

字递增（INCW）和字递减（DECW）操作是有符号的。

双字递增（INCD）和双字递减（DECD）操作是有符号的。

使ENO=0的错误条件:

- SM1.1（溢出）
- 0006（间接寻址）

受影响的特殊存储器位:

- SM1.0（结果为0）
- SM1.1（溢出）
- SM1.2（结果为负）对于字和双字操作有效

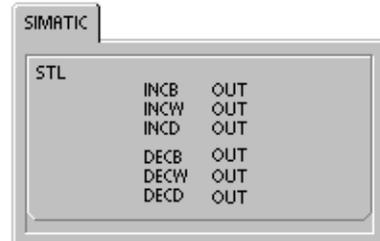
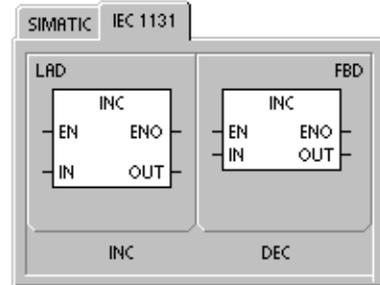
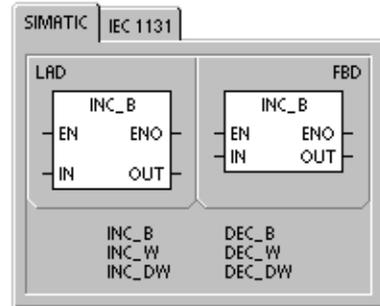


表6-42 递增和递减指令的有效操作数

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

示例：递增和递减指令



比例/积分/微分 (PID) 回路控制指令

PID回路控制指令 (PID) 利用表 (TBL) 中的输入和配置信息, 在被参考的LOOP上执行PID回路计算。

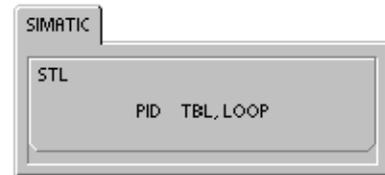
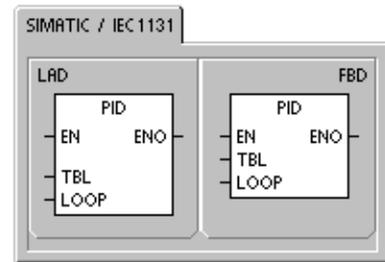
使ENO=0的错误条件:

- SM1.1 (溢出)
- 0006 (间接寻址)

受影响的特殊存储器位:

- SM1.1 (溢出)

PID回路指令 (包含比例、积分、微分回路) 可以用来进行PID运算。但是, 可以进行这种PID运算的前提条件是逻辑堆栈栈顶 (TOS) 值必须为1。该指令有两个操作数: TBL和LOOP。其中TBL是回路表的起始地址; LOOP是回路号, 可以是0到7的整数。



在程序中最多可以用8条PID指令。如果两个或两个以上的PID指令用了同一个回路号, 那么即使这些指令的回路表不同, 这些PID运算之间也会相互干涉, 产生不可预料的结果。

回路表包含9个参数, 用来控制和监视PID运算。这些参数分别是过程变量当前值 (PVn), 过程变量前值 (PVn-1), 给定值 (SPn), 输出值 (Mn), 增益 (Kc), 采样时间 (Ts), 积分时间 (TI), 微分时间 (TD) 和积分项前值 (MX)。

为了让PID运算以预想的采样频率工作, PID指令必须用在定时发生的中断程序中, 或者用在主程序中被定时器所控制以一定频率执行。采样时间必须通过回路表输入到PID运算中。

自整定功能已经集成到PID指令中。对于自整定的详细描述, 参考第15章。PID整定控制面板只能用于由PID向导创建的PID回路。

表6-43 PID回路控制指令的有效操作数

输入/输出	数据类型	操作数
TBL	BYTE	VB
LOOP	BYTE	常数 (0到7)



指令向导



STEP 7-Micro/WIN提供了PID指令向导, 指导您定义一个闭环控制过程的PID算法。在命令菜单中选择**Tools > Instruction Wizard**, 然后在指令向导窗口中选择**PID**指令。

提示

下限设置点和上限设置点要和过程变量的下限和上限相对应。

理解PID算法

PID控制器调节输出，保证偏差（ e ）为零，使系统达到稳定状态，偏差（ e ）是给定值（SP）和过程变量（PV）的差。PID控制的原理基于下面的算式；输出 $M(t)$ 是比例项、积分项和微分项的函数。

输出	=	比例项	+	积分项	+	微分项
$M(t)$	=	$K_C * e$	+	$K_I \int_0^t e dt + M_{initial}$	+	$K_D * de/dt$
其中:	$M(t)$	PID回路的输出，是时间的函数				
	K_C	PID回路的增益				
	e	PID回路的偏差（给定值与过程变量之差）				
	$M_{initial}$	PID回路输出的初始值				

为了能让数字计算机处理这个控制算式，连续算式必须离散化为周期采样偏差算式，才能用来计算输出值。数字计算机处理的算式如下：

M_n	=	比例项	+	积分项	+	微分项
M_n	=	$K_C * e_n$	+	$K_I * \sum_1^n e_x + M_{initial}$	+	$K_D * (e_n - e_{n-1})$
输出	=	比例项	+	积分项	+	微分项
其中:	M_n	在采样时刻 n ，PID回路输出的计算值				
	K_C	PID回路增益				
	e_n	采样时刻 n 的回路偏差值				
	e_{n-1}	回路偏差的前一个值（在采样时刻 $n-1$ ）				
	e_x	采样时刻 x 的回路偏差值				
	K_I	积分项的比例常数				
	$M_{initial}$	回路输出的初始值				
	K_D	微分项的比例常数				

从这个公式可以看出，积分项是从第1个采样周期到当前采样周期所有误差项的函数，微分项是当前采样和前一次采样的函数，比例项仅是当前采样的函数。在数字计算机中，不保存所有的误差项，实际上也不必要。

由于计算机从第一次采样开始，每有一个偏差采样值必须计算一次输出值，只需要保存偏差前值和积分项前值。作为数字计算机解决的重复性的结果，可以得到在任何采样时刻必须计算的方程的一个简化算式。简化算式是：

M_n	=	比例项	+	积分项	+	微分项
M_n	=	$K_C * e_n$	+	$K_I * e_n + MX$	+	$K_D * (e_n - e_{n-1})$
输出	=	比例项	+	积分项	+	微分项
其中:	M_n	在第 n 采样时刻，PID回路输出的计算值				
	K_C	PID回路增益				
	e_n	在第 n 采样时刻的偏差值				
	e_{n-1}	在第 $n-1$ 采样时刻的偏差值（偏差前项）				
	K_I	积分项的比例常数				
	MX	积分项前值				
	K_D	微分项的比例常数				

CPU实际使用以上简化算式的改进形式计算PID输出。这个改进型算式是:

M_n	=	MP_n	+	MI_n	+	MD_n
输出	=	比例项	+	积分项	+	微分项
其中:	M _n	第n采样时刻的计算值				
	MP _n	第n采样时刻的比例项值				
	MI _n	第n采样时刻的积分项值				
	MD _n	第n采样时刻的微分项值				

理解PID方程的比例项

比例项MP是增益(K_C)和偏差(e)的乘积。其中K_C决定输出对偏差的灵敏度, 偏差(e)是给定值(SP)与过程变量值(PV)之差。S7-200解决的求比例项的算式是:

MP_n	=	K_C	*	(SP_n - PV_n)	
其中:	MP _n	第n采样时刻比例项的值			
	K _C	增益			
	SP _n	第n采样时刻的给定值			
	PV _n	第n采样时刻的过程变量值			

理解PID方程的积分项

积分项值MI与偏差和成正比。S7-200解决的求积分项的算式是:

MI_n	=	K_C	*	T_S	/	T_I	*	(SP_n - PV_n)	+	MX
其中:	MI _n	第n采样时刻的积分项值								
	K _C	增益								
	T _S	采样时间间隔								
	T _I	积分时间								
	SP _n	第n采样时刻的给定值								
	PV _n	第n采样时刻的过程变量值								
	MX	第n - 1采样时刻的积分项(积分项前值) (也称积分和或偏置)								

积分和(MX)是所有积分项前值之和。在每次计算出MI_n之后, 都要用MI_n去更新MX。其中MI_n可以被调整或限定(详见“变量和范围”一节)。MX的初值通常在第一次计算输出以前被设置为M_{initial}(初值)。积分项还包括其他几个常数: 增益(K_C)、采样时间间隔(T_S)和积分时间(T_I)。其中采样时间是重新计算输出的时间间隔, 而积分时间控制积分项在整个输出结果中影响的大小。

理解PID方程的微分项

微分项值MD与偏差的变化成正比。S7-200使用下列算式来求解微分项:

$$MD_n = K_C * T_D / T_S * ((SP_n - PV_n) - (SP_{n-1} - PV_{n-1}))$$

为了避免给定值变化的微分作用而引起的跳变, 假定给定值不变 ($SP_n = SP_{n-1}$)。这样, 可以用过程变量的变化替代偏差的变化, 计算算式可改进为:

$$MD_n = K_C * T_D / T_S * (SP_n - PV_n - SP_n + PV_{n-1})$$

或

$$MD_n = K_C * T_D / T_S * (PV_{n-1} - PV_n)$$

其中:

MD_n	第n采样时刻的微分项值
K_C	回路增益
T_S	回路采样时间
T_D	微分时间
SP_n	第n采样时刻的给定值
SP_{n-1}	第n-1采样时刻的给定值
PV_n	第n采样时刻的过程变量值
PV_{n-1}	第n-1采样时刻的过程变量值

为了下一次计算微分项值, 必须保存过程变量, 而不是偏差。在第一采样时刻, 初始化为 $PV_{n-1} = PV_n$ 。

回路控制类型的选择

在许多控制系统中, 只需要一种或两种回路控制类型。例如只需要比例回路或者比例积分回路。通过设置常量参数, 可以选择需要的回路控制类型。

如果不要积分动作 (PID计算中没有“I”), 可以把积分时间 (复位) 置为无穷大“INF”。即使没有积分作用, 积分项还是不为零, 因为有初值MX。

如果不要微分回路, 可以把微分时间置为零。

如果不要比例回路, 但需要积分或积分微分回路, 可以把增益设为0.0, 系统会在计算积分项和微分项时, 把增益当作1.0看待。

回路输入的转换和标准化

每个PID回路有两个输入量, 给定值 (SP) 和过程变量 (PV)。给定值通常是一个固定的值, 比如设定的汽车速度。过程变量是与PID回路输出有关, 可以衡量输出对控制系统作用的大小。在汽车速度控制系统的实例中, 过程变量应该是测量轮胎转速的测速计输入。

给定值和过程变量都可能是现实世界的值，它们的大小、范围和工程单位都可能不一样。在PID指令对这些现实世界的值进行运算之前，必须把它们转换成标准的浮点型表达形式。

转换的第一步是把16位整数转换成浮点型实数值。下面的指令序列提供了实现这种转换的方法：

ITD	AIW0, AC0	//将输入值转换为双整数。
DTR	AC0, AC0	//将32位双整数转换为实数。

下一步是将现实世界的值的实数值表达形式转换成0.0~1.0之间的标准化值。下面的算式可以用于标准化给定值或过程变量值：

R_{Norm}	=	((R_{Raw} / Span) + Offset)
其中:	R _{Norm}	标准化的实数值
	R _{Raw}	没有标准化的实数值或原值
	Offset	单极性为0.0, 双极性为0.5
	Span	值域大小, 可能的最大值减去可能的最小值 单极性为32,000 (典型值) 双极性为64,000 (典型值)

下面的指令把双极性实数标准化为0.0~1.0之间的实数。通常用在第一步转换之后：

/R	64000.0, AC0	//累加器中的标准化值
+R	0.5, AC0	//加上偏置, 使其在0.0~1.0之间
MOVR	AC0, VD100	//标准化的值存入回路表

回路输出值转换成刻度整数

回路输出值一般是控制变量，比如，在汽车速度控制中，可以是油阀开度的设置。回路输出是0.0和1.0之间的一个标准化了的实数值。在回路输出可以用于驱动模拟输出之前，回路输出必须转换成一个16位的标定整数。这一过程，是给定值或过程变量的标准化转换的逆过程。第一步是使用下面给出的公式，将回路输出转换成一个标定的实数值：

R_{Scal}	=	(M_n - Offset) * Span
其中:	R _{Scal}	回路输出的刻度实数值
	M _n	回路输出的标准化实数值
	Offset	单极性为0.0, 双极性为0.5
	Span	值域大小, 可能的最大值减去可能的最小值 单极性为32,000 (典型值) 双极性为64,000 (典型值)

这一过程可以用下面的指令序列完成:

MOVR	VD108, AC0	//把回路输出值移入累加器
-R	0.5, AC0	//仅双极性有此句
*R	64000.0, AC0	//在累加器中得到刻度值

下一步是把回路输出的刻度转换成16位整数, 可通过下面的指令序列来完成:

ROUND	AC0, AC0	//把实数转换为32位整数
DTI	AC0, LW0	//把32位整数转换为16位整数
MOVW	LW0, AQW0	//把16位整数写入模拟输出寄存器

正作用或反作用回路

如果增益为正, 那么该回路为正作用回路。如果增益为负, 那么是反作用回路。(对于增益值为0.0的I或ID控制, 如果指定积分时间、微分时间为正, 就是正作用回路; 如果指定为负值, 就是反作用回路。)

变量和范围

过程变量和给定值是PID运算的输入值, 因此回路表中的这些变量只能被PID指令读而不能被改写。

输出变量是由PID运算产生的, 所以在每一次PID运算完成之后, 需更新回路表中的输出值, 输出值被限定在0.0~1.0之间。当PID指令从手动方式转变到自动方式时, 回路表中的输出值可以用来初始化输出值。(有关PID指令的方式详见下面的“控制方式”一节)。

如果使用积分控制, 积分项前值要根据PID运算结果更新。这个更新了的值用作下一次PID运算的输入, 当输出值超过范围(大于1.0或小于0.0), 那么积分项前值必须根据下列公式进行调整:

MX	= 1.0	- (MP_n + MD_n)	<i>当计算输出 M_n > 1.0</i>
or			
MX	= - (MP_n + MD_n)		<i>当计算输出 M_n < 0.0</i>
其中:	MX	经过调整了的积分和(积分项前值)	
	MP _n	第n采样时刻的比例项值	
	MD _n	第n采样时刻的微分项值	
	M _n	第n采样时刻的输出值	

这样调整积分前值, 一旦输出回到范围后, 可以提高系统的响应性能。而且积分项前值也要限制在0.0~0.1之间, 然后在每次PID运算结束之后, 把积分项前值写入回路表, 以备在下次PID运算中使用。

用户可以在执行PID指令以前修改回路表中积分项前值。在实际运用中, 这样做的目的是找到由于积分项前值引起的问题。手工调整积分项前值时, 必须小心谨慎, 还应保证写入的值在0.0~1.0之间。

回路表中的给定值与过程变量的差值(e)是用于PID运算中的差分运算, 用户最好不要去修改此值。

控制方式

S7-200的PID回路没有设置控制方式，只要PID块有效，就可以执行PID运算。在这种意义上说，PID运算存在一种“自动”运行方式。当PID运算不被执行时，我们称之为“手动”模式。

同计数器指令相似，PID指令有一个使能位。当该使能位检测到一个信号的正跳变（从0到1），PID指令执行一系列的动作，使PID指令从手动方式无扰动地切换到自动方式。为了达到无扰动切换，在转变到自动控制前，必须用手动方式把当前输出值填入回路表中的 M_n 栏。PID指令对回路表中的值进行下列动作，以保证当使能位正跳变出现时，从手动方式无扰动切换到自动方式：

- 置给定值 (SP_n) = 过程变量 (PV_n)
- 置过程变量前值 (PV_{n-1}) = 过程变量现值 (PV_n)
- 置积分项前值 (MX) = 输出值 (M_n)

PID使能位的默认值是1，在CPU启动或从STOP方式转到RUN方式时建立。CPU进入RUN方式后首次使PID块有效，没有检测到使能位正跳变，那么就没有无扰动切换的动作。

报警与特殊操作

PID指令是执行PID运算的简单而功能强大的指令。如果需要其它处理，如报警检查或回路变量的特殊计算等，则这些处理必须使用S7-200支持的基本指令来实现。

出错条件

如果指令指定的回路表起始地址或PID回路号操作数超出范围，那么在编译期间，CPU将产生编译错误（范围错误），从而编译失败。

PID指令不检查回路表中的值是否在范围之内，所以必须小心操作以保证过程变量和设定值不超界。PID指令不检查回路表中的值是否超界，您必须保证过程变量和设定值（以及偏置和前一次过程变量）必须在0.0到1.0之间。

如果PID计算的算术运算发生错误，那么特殊存储器标志位SM1.1（溢出或非法值）会被置1，并且中止PID指令的执行。（要想消除这种错误，单靠改变回路表中的输出值是不够的，正确的方法是在下一次执行PID运算之前，改变引起算术运算错误的输入值，而不是更新输出值）。

回路表

回路表有80字节长，它的格式如表6-44所示。

表6-44 回路表

偏移地址	域	格式	定时器类型	中断描述
0	过程变量 (PV_n)	REAL	输入	过程变量，必须在0.0~1.0之间
4	设定值 (SP_n)	REAL	输入	给定值，必须在0.0~1.0之间
8	输出值 (M_n)	REAL	输入/输出	输出值，必须在0.0~1.0之间
12	增益 (K_C)	REAL	输入	增益是比例常数，可正可负
16	采样时间 (T_S)	REAL	输入	单位为秒，必须是正数
20	积分时间 (T_I)	REAL	输入	单位为分钟，必须是正数
24	微分时间 (T_D)	REAL	输入	单位为分钟，必须是正数
28	积分项前项 (MX)	REAL	输入/输出	积分项前项，必须在0.0~1.0之间
32	过程变量前值 (PV_{n-1})	REAL	输入/输出	包含最后一次执行PID指令时存储的过程变量值。
36 - 79	保留给自整定变量。对于详细信息，参考表15-1。			

中断指令

中断允许和中断禁止

中断允许指令（ENI）全局地允许所有被连接的中断事件。中断禁止指令（DISI）全局地禁止处理所有中断事件。

当进入RUN模式时，中断被禁止。在RUN模式，您可以执行全局中断允许指令（ENI）允许所有中断。全局中断禁止指令（DISI）不允许处理中断服务程序，但中断事件仍然会排队等候。

使ENO=0的错误条件:

- 0004 (试图在中断服务程序中执行ENI、DISI或者HDEF指令。)

中断条件返回

中断条件返回指令（CRETI）用于根据前面的逻辑操作的条件，从中断服务程序中返回。

中断连接

中断连接指令（ATCH）将中断事件EVNT与中断服务程序号INT相关联，并使能该中断事件。

使ENO=0的错误条件:

- 0002 (与HSC的输入分配相冲突)

中断分离

中断分离指令（DTCH）将中断事件EVNT与中断服务程序之间的关联切断，并禁止该中断事件。

清除中断事件

清除中断事指令从中断队列中清除所有EVNT类型的中断事件。使用此指令从中断队列中清除不需要的中断事件。如果此指令用于清除假的中断事件，在从队列中清除事件之前要首先分离事件。否则，在执行清除事件指令之后，新的事件将被增加到队列中。

实例说明了处于正交模式的高速计数器如何使用CLR_EVNT指令清除中断事件。如果光电传感器正好处在从明亮过渡到黑暗的边界位置，那么在新的PV值装载之前，小的机械振动将生成实际并不需要的中断。

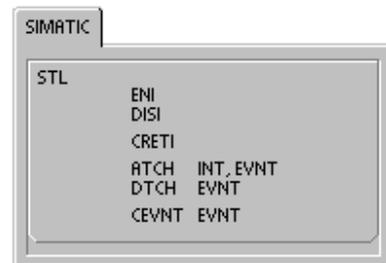
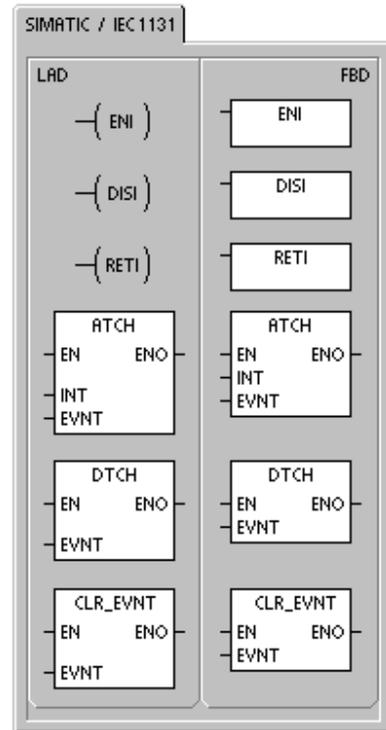


表6-45 中断指令的有效操作数

输入/输出	数据类型	操作数
INT	BYTE	常数 (0到127)
EVNT	BYTE	常数 CPU221 和 CPU222: 0到12, 19到23和27到33 CPU 224: 0到23和27到33 CPU224XP 和 CPU226: 0到33

对中断连接和中断分离指令的理解

在激活一个中断程序前，必须在中断事件和该事件发生时希望执行的那段程序间建立一种联系。中断连接指令（ATCH）指定某中断事件（由中断事件号指定）所要调用的程序段（由中断程序号指定）。多个中断事件可调用同一个中断程序，但一个中断事件不能同时指定调用多个中断程序。

当把中断事件和中断程序连接时，自动允许中断。如果采用禁止全局中断指令不响应所有中断，每个中断事件进行排队，直到采用允许全局中断指令重新允许中断。

可以用中断分离指令（DTCH）截断中断事件和中断程序之间的联系，以单独禁止中断事件。中断分离指令（DTCH）使中断回到不激活或无效状态表6-46列出了不同类型的中断事件。

表6-46 中断事件

事件号	中断描述	CPU221 CPU222	CPU224	CPU224XP CPU226
0	上升沿, I0.0	Y	Y	Y
1	下降沿, I0.0	Y	Y	Y
2	上升沿, I0.1	Y	Y	Y
3	下降沿, I0.1	Y	Y	Y
4	上升沿, I0.2	Y	Y	Y
5	下降沿, I0.2	Y	Y	Y
6	上升沿, I0.3	Y	Y	Y
7	下降沿, I0.3	Y	Y	Y
8	端口0: 接收字符	Y	Y	Y
9	端口0: 发送完成	Y	Y	Y
10	定时中断0	Y	Y	Y
11	定时中断1	SMB35	Y	Y
12	HSC0	CV=PV (当前值=预置值)	Y	Y
13	HSC1	CV=PV (当前值=预置值)	Y	Y
14	HSC1输入方向改变		Y	Y
15	HSC1外部复位		Y	Y
16	HSC2	CV=PV (当前值=预置值)	Y	Y
17	HSC2输入方向改变		Y	Y
18	HSC2外部复位		Y	Y
19	PTO 0完成中断	Y	Y	Y
20	PTO 1完成中断	Y	Y	Y
21	定时器T32 CT=PT中断	Y	Y	Y

表6-46 中断事件

事件号	中断描述	CPU221 CPU222	CPU224	CPU224XP CPU226
22	定时器T96 CT=PT中断	Y	Y	Y
23	端口0: 接收信息完成	Y	Y	Y
24	端口1: 接收信息完成			Y
25	端口1: 接收字符			Y
26	端口1: 发送完成			Y
27	HSC0输入方向改变	Y	Y	Y
28	HSC0外部复位	Y	Y	Y
29	HSC4 CV=P (当前值=预置值)	Y	Y	Y
30	HSC4输入方向改变	Y	Y	Y
31	HSC4外部复位	Y	Y	Y
32	HSC3 CV=P (当前值=预置值)	Y	Y	Y
33	HSC5 CV=P (当前值=预置值)	Y	Y	Y

理解S7-200对中断服务程序的处理

执行中断服务程序用于响应与其相关的内部或者外部事件。一旦执行完中断服务程序的最后一条指令，控制权会回到主程序。您可以执行中断条件返回指令（CRETI）退出中断服务程序。表6-47对于在应用程序中使用中断服务程序给出了一些指导和限定。

表6-47 使用中断服务程序的指导和限定

指导
中断处理提供了对特殊的内部或外部事件的响应。用户应当优化中断程序以执行一个特殊的任务，然后把控制返回主程序。 应当使中断程序短小而简单，执行时对其他处理也不要延时过长。如果做不到这些，意外的条件可能会引起由主程序控制的设备操作异常。对中断而言，其格言是“越短越好”。
限定
在中断程序中不能使用DISI、ENI、HDEF、LSCR和END指令。

系统对中断的支持

由于中断指令影响触点、线圈和累加器逻辑，所以系统保存和恢复逻辑堆栈、累加寄存器以及指示累加器和指令操作状态的特殊存储器标志位（SM）。这避免了进入中断程序或从中断程序返回对主用户程序造成破坏。

在主程序和中断程序间共享数据

您可以在主程序和一个或多个中断程序间共享数据。例如，用户主程序的某个地方可以为某个中断程序提供要用到的数据，反之亦然。如果用户程序共享数据，必须考虑中断事件异步特性的影响，这是因为中断事件会在用户主程序执行的任何地方出现。共享数据一致性问题的解决要依赖于主程序被中断事件中断时中断程序的操作。使用中断服务程序的局部变量表，这样可以保证中断服务程序只使用临时内存，而不会覆盖程序的其它地方使用的数据。

这里有几种可以确保在用户主程序和中断程序间正确共享数据的编程技巧。这些技巧或限制共享存储器单元的访问方式，或让使用共享存储器单元的指令序列不会被中断。

- STL程序共享单个变量：如果共享数据是单个字节、字、双字变量，而且用户程序用STL编写，那么通过把对共享数据操作得到的中间值只存储到非共享的存储器单元或累加器中，可以保证正确的共享访问。
- LAD程序共享单个变量：如果共享数据是单个字节、字或双字变量，而且用户程序用梯形图编写，那么通过建立只用Move指令（MOVB、MOVW、MOVD、MOVR）访问共享存储器单元的约定，可以保证正确的共享访问。这些Move指令由执行时不受中断事件影响的单条STL指令组成，而其它许多梯形图指令是由可被中断的STL指令序列组成的。
- STL或LAD程序共享多个变量：如果共享数据由一些相关的字节、字或双字组成，那么可以用中断禁止/允许指令（DISI和ENI）来控制中断程序的执行。在用户程序开始对共享存储器单元操作的地方禁止中断，一旦所有影响共享存储器单元的操作完成后，再允许中断。在访问共享存储器单元期间，中断被禁止，中断程序不能执行，因而也无法访问共享存储器单元，但这种方法导致了中断事件响应的延迟。

在中断服务程序中调用子程序

您可以在一个中断服务程序中调用一个子程序。中断服务程序与被调用的子程序共享累加器和逻辑堆栈。

S7-200支持的中断类型

S7-200支持下列类型的中断服务程序：

- 通讯口中断：S7-200生成使您的程序可以控制通讯口的事件。
- I/O中断：S7-200对I/O点状态的各种变化产生中断事件。这些事件使您可以对高速计数器、脉冲输出或输入的上升或下降状态做出响应。
- 时基中断：S7-200产生使您的程序在指定的间隔上起作用的事件。

通讯口中断

PLC的串行通讯口可由LAD或STL程序来控制。通讯口的这种操作模式称为自由端口模式。在自由端口模式下，用户可用程序定义波特率、每个字符位数、奇偶校验和通讯协议。利用接收和发送中断可简化程序对通讯的控制。对于更多信息，参考发送和接收指令。

I/O中断

I/O中断包含了上升沿或下降沿中断、高速计数器中断和脉冲串输出（PTO）中断。S7-200 CPU可用输入I0.0至I0.3的上升沿或下降沿产生中断。表9-21给出了允许中断的输入，上升沿事件和下降沿事件可被这些输入点捕获。这些上升沿/下降沿事件可被用于指示当某个事件发生时必须引起注意的条件。

高速计数器中断允许响应诸如当前值等于预置值、相应于轴转动方向变化的计数方向改变和计数器外部复位等事件而产生的中断。每种高速计数器可对高速事件实时响应，而PLC扫描速率对这些高速事件是不能控制的。

脉冲串输出中断给出了已完成指定脉冲数输出的指示。脉冲串输出的一个典型应用是步进电机。

可以通过将一个中断程序连接到相应的I/O事件上来允许上述的每一个中断。

时基中断

时基中断包括定时中断和定时器T32/T96中断。CPU可以支持定时中断。可以用定时中断指定一个周期性的活动。周期以1ms为增量单位，周期时间可从1ms到255ms。对定时中断0，必须把周期时间写入SMB34；对定时中断1，必须把周期时间写入SMB35。

每当定时器溢出时，定时中断事件把控制权交给相应的中断程序。通常可用定时中断以固定的时间间隔去控制模拟量输入的采样或者执行一个PID回路。

当把某个中断程序连接到一个定时中断事件上，如果该定时中断被允许，那就开始计时。在连接期间，系统捕捉周期时间值，因而后来的变化不会影响周期。为改变周期时间，首先必须修改周期时间值，然后重新把中断程序连接到定时中断事件上。当重新连接时，定时中断功能清除前一次连接时的任何累计值，并用新值重新开始计时。

一旦允许，定时中断就连续地运行，指定时间间隔的每次溢出时执行被连接的中断程序。如果退出RUN模式或分离定时中断，则定时中断被禁止。如果执行了全局中断禁止指令，定时中断事件会继续出现，每个出现的定时中断事件将进入中断队列（直到中断允许或队列满）。请参见定时中断的例子程序。

定时器T32/T96中断允许及时地响应一个给定的时间间隔。这些中断只支持1ms分辨率的延时接通定时器（TON）和延时断开定时器（TOF）T32和T96。T32和T96定时器在其它方面工作正常。一旦中断允许，当有效定时器的当前值等于预置值时，在CPU的正常1ms定时刷新中，执行被连接的中断程序。首先把一个中断程序连接到T32/T96中断事件上，然后允许该中断。

中断优先级和中断队列

在各个指定的优先级之内，CPU按先来先服务的原则处理中断。任何时间点上，只有一个用户中断程序正在执行。一旦中断程序开始执行，它要一直执行到结束。而且不会被别的中断程序，甚至是更高优先级的中断程序所打断。当另一个中断正在处理中，新出现的中断需要排队，等待处理。

表6-48给出了3个中断队列以及它们能够存储的中断个数。

表6-48 每个中断队列的最大数目

队列	CPU211、CPU222、CPU224	CPU224XP和CPU226
通讯中断队列	4	8
I/O中断队列	16	16
定时中断队列	8	8

有时，可能有多于队列所能保存数目的中断出现，因而，由系统维护的队列溢出存储器位表明丢失的中断事件的类型。中断队列溢出位如表6-49所示。您应当只在中断程序中使用这些位，因为在队列变空或控制返回到主程序时，这些位会被复位。

表6-50给出了所有中断事件的优先级和事件号。

表6-49 中断队列溢出标志位

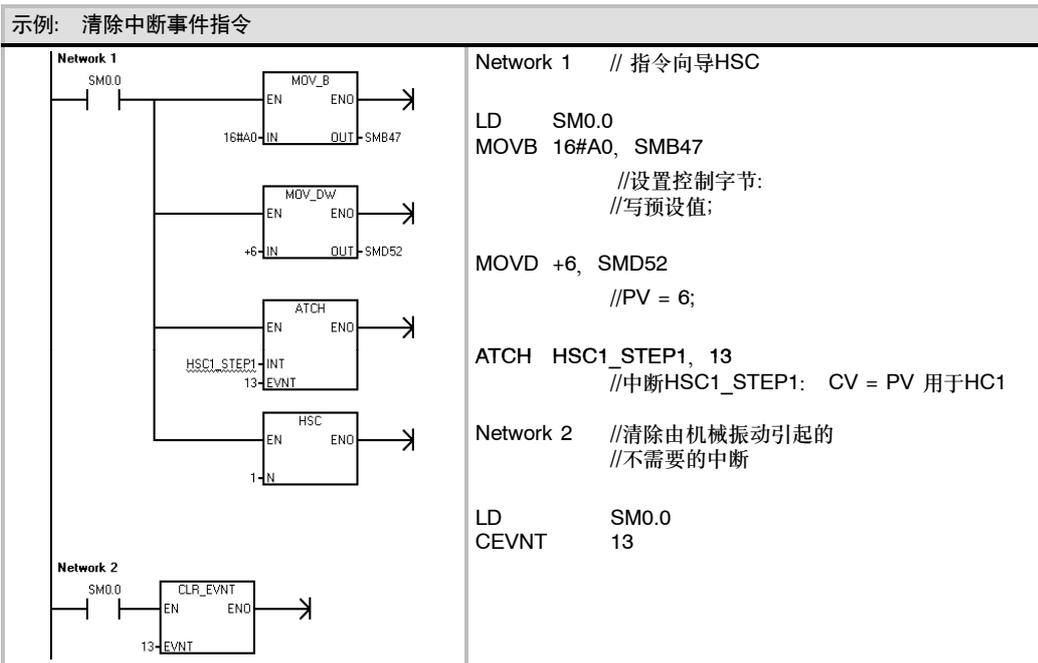
描述 (0=不溢出, 1=溢出)	SM位
通讯中断队列	SM4.0
I/O中断队列	SM4.1
定时中断队列	SM4.2

表6-50 中断事件的优先级顺序

事件号	中断描述	优先级	优先组中的优先级
8	端口0: 接收字符	通讯 (最高)	0
9	端口0: 发送完成		0
23	端口0: 接收信息完成		0
24	端口1: 接收信息完成		1
25	端口1: 接收字符		1
26	端口1: 发送完成		1
19	PTO 0完成中断		I/O (中等)
20	PTO 1完成中断	1	
0	上升沿, I0.0	2	
2	上升沿, I0.1	3	
4	上升沿, I0.2	4	
6	上升沿, I0.3	5	
1	下降沿, I0.0	6	
3	下降沿, I0.1	7	
5	下降沿, I0.2	8	
7	下降沿, I0.3	9	
12	HSC0 CV=PV (当前值=预置值)	10	
27	HSC0输入方向改变	11	
28	HSC0外部复位	12	
13	HSC1 CV=PV (当前值=预置值)	13	
14	HSC1输入方向改变	14	
15	HSC1外部复位	15	
16	HSC2 CV=PV (当前值=预置值)	16	
17	HSC2输入方向改变	17	
18	HSC2外部复位	18	
32	HSC3 CV=PV (当前值=预置值)	19	
29	HSC4 CV=PV (当前值=预置值)	20	
30	HSC4输入方向改变	21	
31	HSC4外部复位	22	
33	HSC5 CV=PV (当前值=预置值)	23	
10	定时中断0	定时 (最低)	0
11	定时中断1 SMB35		1
21	定时器T32 CT=PT中断		2
22	定时器T96 CT=PT中断		3

示例: 中断指令		
<p>M A I N</p>	<p>Network 1</p> <pre> //首次扫描 //1. 定义I0.0 // 的下降沿中断服务程序为INT_0 //2. 全局中断允许。 LD SM0.1 ATCH INT_0, 1 ENI Network 2 //如果检测到I/O错误, //禁止I0.0的下降沿中断。 //该程序段是可选的。 LD SM5.0 DTCH 1 Network 3 //当M5.0接通时, //禁止所有中断。 LD M5.0 DISI </pre>	<p>Network 1 //首次扫描 //1. 定义I0.0 // 的下降沿中断服务程序为INT_0 //2. 全局中断允许。 LD SM0.1 ATCH INT_0, 1 ENI</p> <p>Network 2 //如果检测到I/O错误, //禁止I0.0的下降沿中断。 //该程序段是可选的。 LD SM5.0 DTCH 1</p> <p>Network 3 //当M5.0接通时, 禁止所有中断。 LD M5.0 DISI</p>
<p>I N T O</p>	<p>Network 1</p> <pre> LD SM5.0 CRETI </pre>	<p>Network 1 //I0.0的下降沿中断服务程序: //当有I/O错误时返回。 LD SM5.0 CRETI</p>

示例: 用定时中断读取模拟量的数值程序举例		
<p>M A I N</p>	<p>Network 1</p> <pre> LD SM0.1 CALL SBR_0 ENI </pre>	<p>Network 1 //首次扫描, 调用子程序0。 LD SM0.1 CALL SBR_0 ENI</p>
<p>S B R O</p>	<p>Network 1</p> <pre> LD SM0.0 MOV_B 100, SMB34 ATCH INT_0, 10 ENI </pre>	<p>Network 1 //1. 设置定时中断的时间间隔为100ms。 //2. 连接INT_0到定时中断0 (事件10)。 //3. 全局中断允许。 LD SM0.0 MOVB 100, SMB34 ATCH INT_0, 10 ENI</p>
<p>I N T O</p>	<p>Network 1</p> <pre> LD SM0.0 MOV_W AIW4, VW100 ENI </pre>	<p>Network 1 //每100ms读AIW4的值。 LD SM0.0 MOVW AIW4, VW100 ENI</p>



逻辑操作指令

取反指令

字节、字和双字取反

字节取反 (INVB)、字取反 (INW) 和双字取反 (INVD) 指令将输入IN取反的结果存入OUT中。

使ENO=0的错误条件:

- 0006 (间接寻址)

受影响的SM标志位:

- SM1.0 (结果为0)

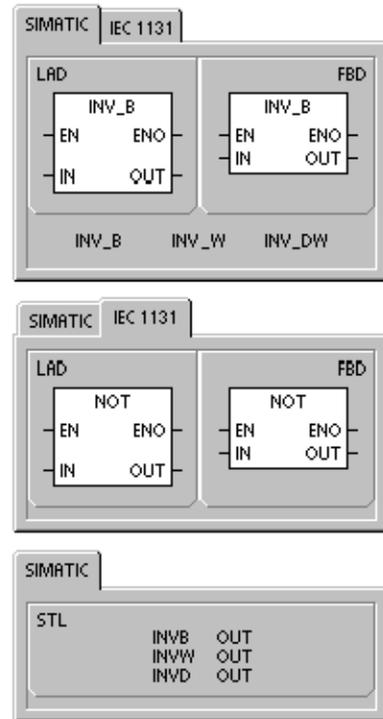


表6-51 取反指令的有效操作数

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



与、或和异或指令

字节与、字与和双字与

字节与（ANDB）、字与（ANDW）和双字与（ANDD）指令将输入值IN1和IN2的相应位进行与操作，将结果存入OUT中。

字节或、字或和双字或

字节或（ORB）、字或指令（ORW）和双字或（ORD）指令将两个输入值IN1和IN2的相应位进行或操作，将结果存入OUT中。

字节异或、字节或和双字异或

字节异或（XORB）、异或（XORW）和双字异或（XORD）指令将两个输入值IN1和IN2的相应位进行异或操作，将结果存入OUT中。

SM标志位和ENO

对于本页中描述的所有指令，下列情况影响SM位和ENO。

使ENO=0的错误条件:

- 0006（间接寻址）

受影响的SM标志位:

- SM1.0（结果为0）

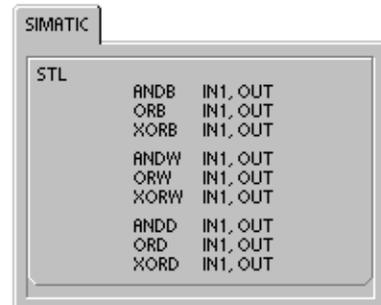
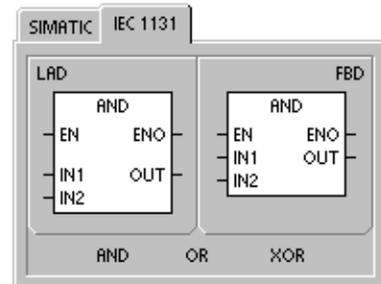
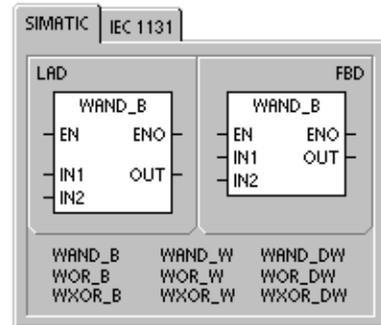


表6-52 与、或和异或指令的有效操作数

输入/输出	数据类型	操作数
IN1、IN2	BYTE WORD DWORD	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数 IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数 ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数
OUT	BYTE WORD DWORD	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*AC、*LD ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD

示例: 与、或和异或指令

Network 1

Network1

```

LD      I4.0
ANDW   AC1, AC0
ORW    AC1, VW100
XORW   AC1, AC0
        
```

字与

AC1	0001 1111 0110 1101	字或	AC1	0001 1111 0110 1101
	AND			OR
AC0	1101 0011 1110 0110		VW100	1101 0011 1010 0000
	等于			等于
AC0	0001 0011 0110 0100		VW100	1101 1111 1110 1101

字异或

AC1	0001 1111 0110 1101
	XOR
AC0	0001 0011 0110 0100
	等于
AC0	0000 1100 0000 1001

传送指令

字节、字、双字或者实数传送

字节传送 (MOVB)、字传送 (MOVW)、双字传送 (MOVD) 和实数传送指令在不改变原值的情况下将IN中的值传送到OUT。

使用双字传送指令可以创建一个指针。要得到更多信息，请参考第4章中指针和间接寻址一节。

对于IEC传送指令，输入和输出的数据类型可以不同，但数据长度必须相同。

使ENO=0的错误条件:

- 0006 (间接寻址)

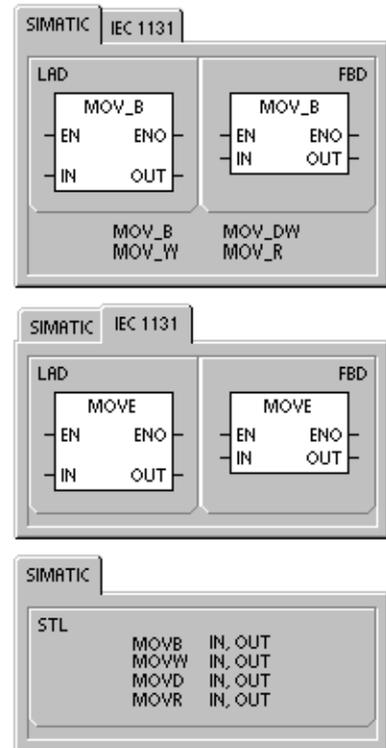


表6-53 传送指令的有效操作数

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

字节立即传送（读和写）

字节立即传送指令允许您在物理I/O和存储器之间立即传送一个字节数据。

字节立即读（BIR）指令读物理输入（IN），并将结果存入内存地址（OUT），但过程映像寄存器并不刷新。

字节立即写指令（BIW）从内存地址（IN）中读取数据，写入物理输出（OUT），同时刷新相应的过程映像区。

使ENO=0的错误条件:

- 0006（间接寻址）
- 不能访问扩展模块

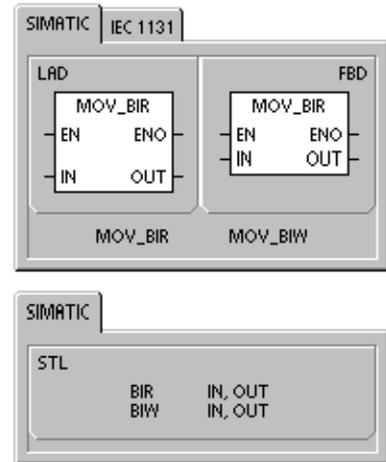


表6-54 字节立即读指令的有效操作数

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6-55 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、常数
OUT	BYTE	QB、*VD、*LD、*AC

块传送指令

字节、字、双字的块传送

字节块传送（BMB）、字块传送（BMW）和双字块传送（BMD）指令传送指定数量的数据到一个新的存储区，数据的起始地址IN，数据长度为N个字节、字或者双字，新块的起始地址为OUT。

N的范围从1到255。

使ENO=0的错误条件:

- 0006（间接寻址）
- 0091（操作数超出范围）

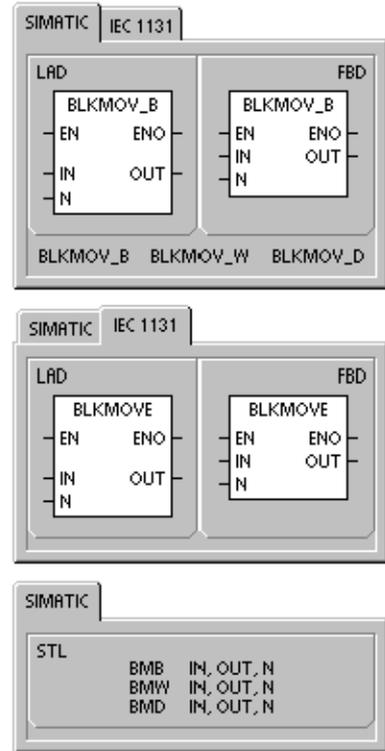
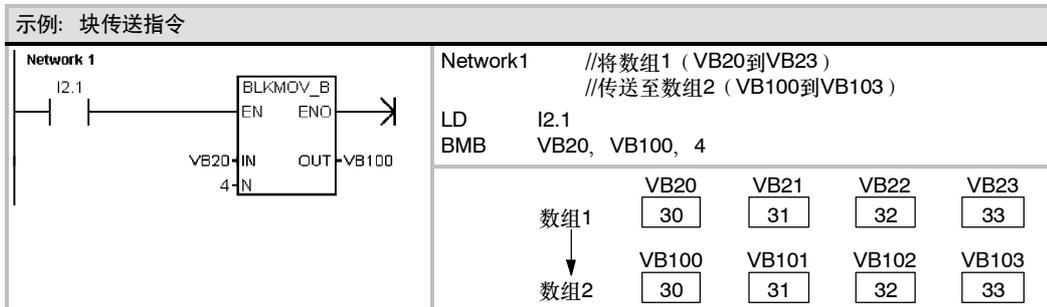


表6-56 传送指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE WORD、INT DWORD、DINT	IB、QB、VB、MB、SMB、SB、LB、*VD、*LD、*AC IW、QW、VW、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、*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、常数、*VD、*LD、*AC



程序控制指令

条件结束

条件结束指令（END）根据前面的逻辑关系终止当前扫描周期。可以在主程序中使用条件结束指令，但不能在子程序或中断服务程序中使用该命令。

停止

停止指令（STOP）导致CPU从RUN到STOP模式从而可以立即终止程序的执行。

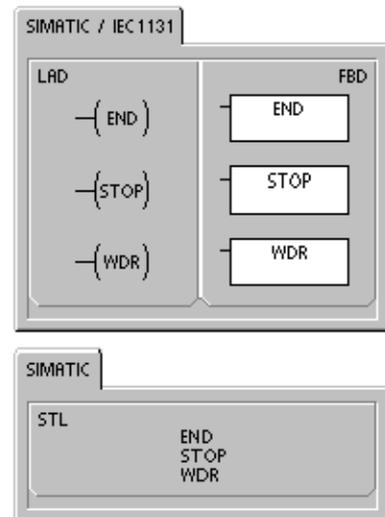
如果STOP指令在中断程序中执行，那么该中断立即终止，并且忽略所有挂起的中断，继续扫描程序的剩余部分。完成当前周期的剩余动作，包括主用户程序的执行，并在当前扫描的最后，完成从RUN到STOP模式的转变。

看门狗复位

看门狗复位指令（WDR）允许S7-200 CPU的系统看门狗定时器被重新触发，这样可以在不引起看门狗错误的情况下，增加此扫描所允许的时间。

使用WDR指令时要小心，因为如果您用循环指令去阻止扫描完成或过度的延迟扫描完成的时间，那么在终止本次扫描之前，下列操作过程将被禁止：

- 通讯（自由端口方式除外）
- I/O更新（立即I/O除外）
- 强制更新
- SM位更新（SM0，SM5～SM29不能被更新）
- 运行时间诊断
- 由于扫描时间超过25秒，10ms和100ms定时器将不会正确累计时间。
- 在中断程序中的STOP指令
- 带数字量输出的扩展模块也包含一个看门狗定时器，如果模块没有被S7-200写，则此看门狗定时器将关断输出。在扩展的扫描时间内，对每个带数字量输出的扩展模块进行立即写操作，以保持正确的输出。请参考这段描述之后的实例。



**提示**

如果希望程序的扫描周期超过500ms，或者在中断事件发生时有可能使程序的扫描周期超过500ms时，您应该使用看门狗复位指令来重新触发看门狗定时器。

每次使用看门狗复位指令，您应该对每个扩展模块的某一个输出字节使用一个立即写指令来复位每个扩展模块的看门狗。

如果您使用了看门狗复位指令允许程序的执行有一个很长的扫描时间，此时将S7-200的模式开关切换到STOP位置，则在1.4秒内，CPU转到STOP方式。

示例：停止、结束和看门狗复位指令程序举例

<p>Network 1</p> <p>Network 2</p> <p>Network 3</p>	<p>Network 1 //当检测到I/O错误时，强制切换到 //STOP模式。</p> <p>LD SM5.0 STOP</p> <p>Network2 //当M5.6接通时，允许扫描周期扩展： //1. 重新触发S7-200 CPU的看门狗。 //2. 重新触发第一个输出模块的看门狗。</p> <p>LD M5.6 WDR BIW QB2, QB2</p> <p>Network 3 //当I0.0接通时，终止当前扫描周期。</p> <p>LD I0.0 END</p>
---	--

For-Next循环指令

FOR和NEXT指令可以描述需重复进行一定次数的循环体。每条FOR指令必须对应一条NEXT指令。For-Next循环嵌套（一个For-Next循环在另一个For-Next循环之内）深度可达8层。

FOR-NEXT指令执行FOR指令和NEXT指令之间的指令。必须指定计数值或者当前循环次数INDX、初始值（INIT）和终止值（FINAL）。

NEXT指令标志着FOR循环的结束。

使ENO=0的错误条件:

- 0006（间接寻址）

如果允许FOR/NEXT循环，除非在循环内部修改了终值，循环体就一直循环执行直到循环结束。当FOR/NEXT循环执行的过程中可以修改这些值。当循环再次允许时，它把初始值拷贝到INDX中（当前循环次数）。

当下一次允许时，FOR/NEXT指令复位它自己。

例如，给定初值（INIT）为1，终值（FINAL）为10，那么随着当前计数值（INDX）从1增加到10，FOR与NEXT之间的指令被执行10次。

1, 2, 3, ...10.

如果初值大于终值，那么循环体不被执行。每执行一次循环体，当前计数值增加1，并且将其结果同终值作比较，如果大于终值，那么终止循环。

如果程序进入FOR-NEXT循环时，栈顶值为1，则当程序退出FOR-NEXT循环时，栈顶值也将为1。

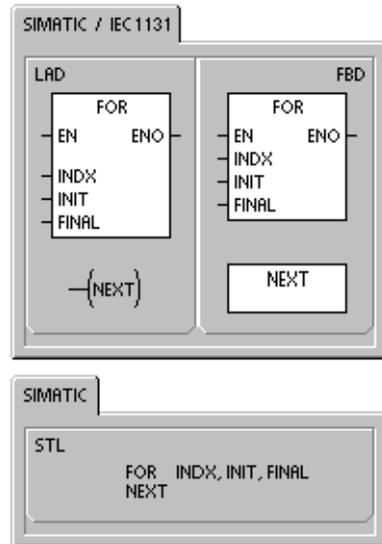
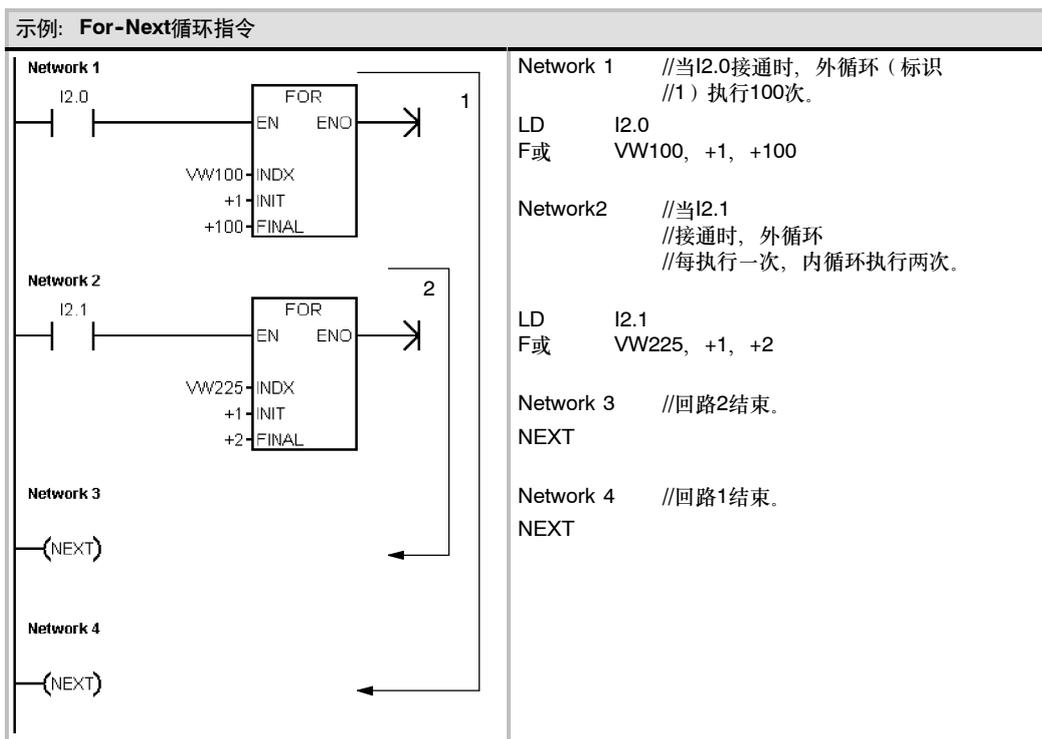


表6-57 FOR-NEXT指令的有效操作数

输入/输出	数据类型	操作数
INDX	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*LD、*AC
INIT、FINAL	INT	VW、IW、QW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*AC、常数



跳转指令

跳转到标号指令（JMP）执行程序内标号N指定的程序分支。

标号指令标记跳转目的地的位置N。

您可以在主程序、子程序或者中断服务程序中，使用跳转指令。跳转和与之相应的标号指令必须位于同一段程序代码（无论是主程序、子程序还是中断服务程序）。

不能从主程序跳到子程序或中断程序，同样不能从子程序或中断程序跳出。

可以在SCR程序段中使用跳转指令，但相应的标号指令也必须也在同一个SCR段中。

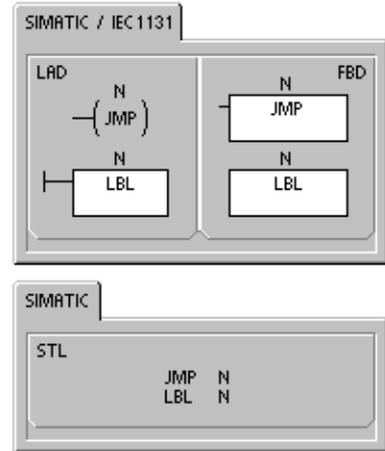


表6-58 跳转指令的有效操作数

输入/输出	数据类型	操作数
N	WORD	常数（0到255）

示例：跳转到标号指令	
<p>Network 1</p> <p>Network 2</p>	<p>Network 1 //如果掉电保持的数据没有丢失，跳 //转到LBL4。</p> <p>LDN SM0.2 JMP 4</p> <p>Network 2 LBL 4</p>

顺控继电器（SCR）指令

SCR指令使您能够按照自然工艺段在LAD、FBD或STL中编制状态控制程序。

无论如何，由一系列操作组成的应用程序都会反复执行，而SCR可以使程序更加结构化，以至于直接针对应用。这样可以使得编程和调试更加快速和简单。

装载SCR指令（LSCR）将S位的值装载到SCR和逻辑堆栈中。

SCR堆栈的结果值决定是否执行SCR程序段。SCR堆栈的值会被复制到逻辑堆栈中，因此可以直接将盒或者输出线圈连接到左侧的能流线上而不经中间触点。

限定

当使用SCR时，请注意下面的限定：

- 不能把同一个S位用于不同程序中。例如：如果在主程序中用了S0.1，在子程序中就不能再使用它。
- 在SCR段之间不能使用JMP和LBL指令，就是说不允许跳入、跳出。可以在SCR段附近使用跳转和标号指令或者在段内跳转。
- 在SCR段中不能使用END指令。

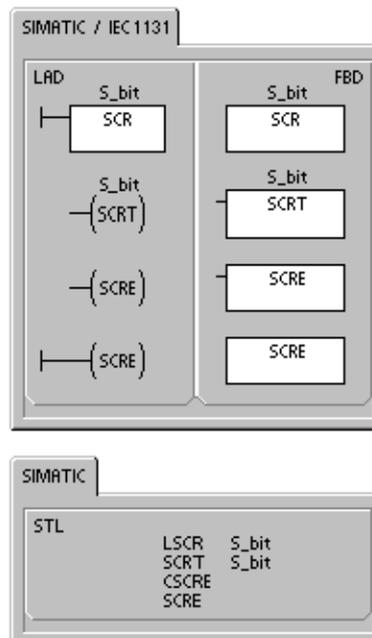


表6-59 顺控继电器指令的有效操作数

输入/输出	数据类型	操作数
S_bit	BOOL	S

图6-32给出了S堆栈和逻辑堆栈以及执行LSCR指令产生的影响。以下是对顺控继电器指令的正确理解:

- ❑ 装载SCR指令 (LSCR) 标志着SCR段的开始, SCR结束指令则标志着SCR段的结束。在装载SCR指令与SCR结束指令之间的所有逻辑操作的执行取决于S堆栈的值。而在SCR结束指令和下一条装载SCR指令之间的逻辑操作则不依赖于S堆栈的值。
- ❑ SCR传输指令 (SCRT) 将程序控制权从一个激活的SCR段传递到另一个SCR段。

执行SCRT指令可以使当前激活的程序段的S位复位, 同时使下一个将要执行的程序段的S位置位。在SCRT指令指行时, 复位当前激活的程序段的S位并不会影响S堆栈。SCR段会一直保持能流直到退出。

装载Sx.y的值到SCR和逻辑堆栈。

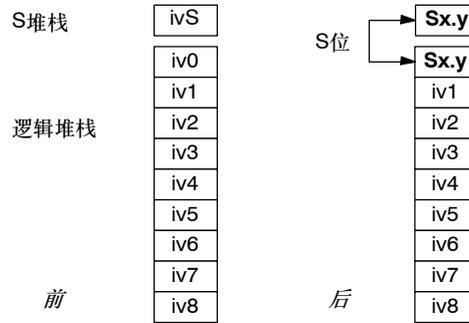
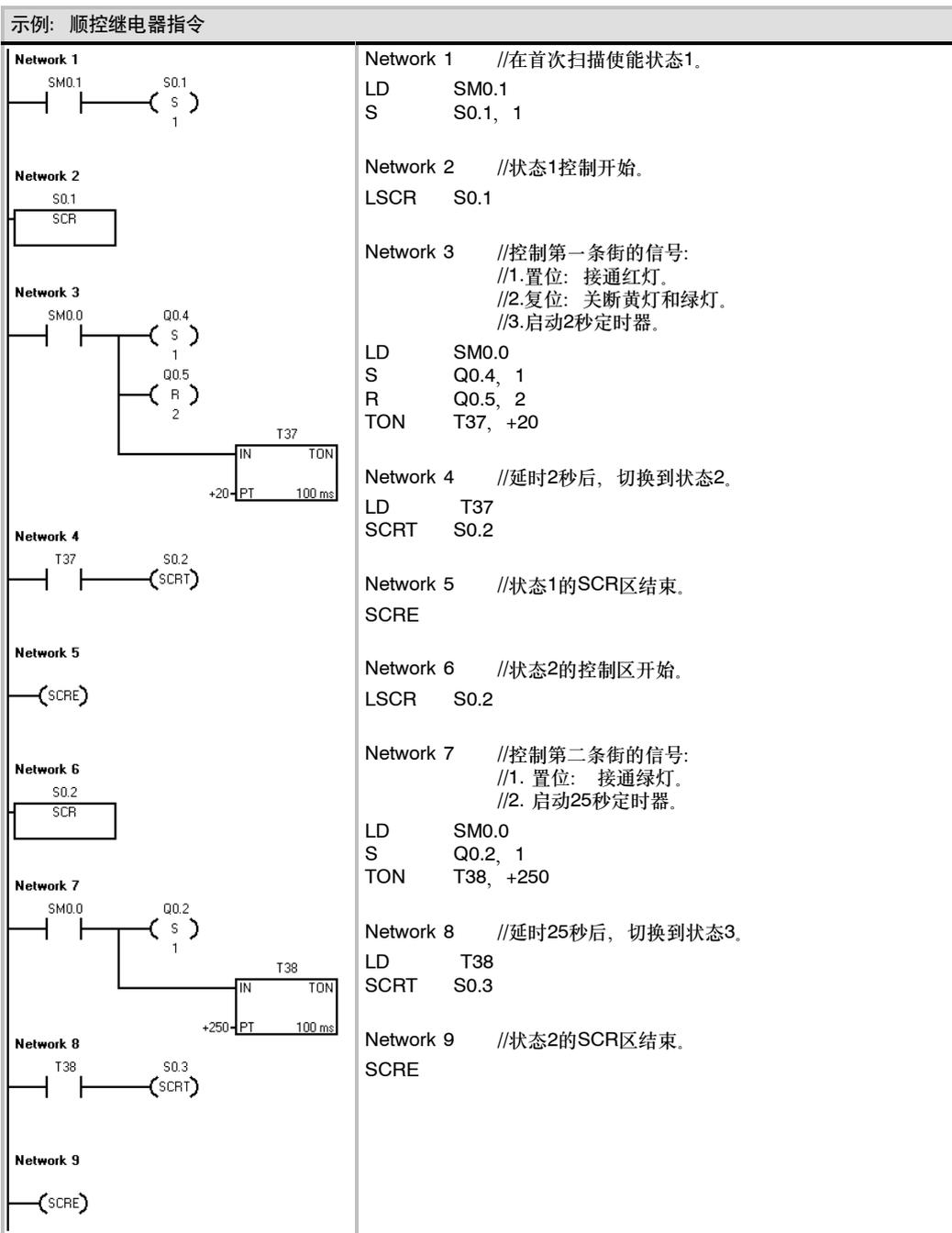


图6-32 LSCR对逻辑堆栈的影响

- ❑ SCR条件结束指令 (CSCRE) 可以使程序退出一个激活的程序段而不执行CSCRE与SCRE之间的指令。CSCRE指令不影响任何S位, 也不影响S堆栈。

在以下实例中, 首次扫描位SM0.1置位S0.1, 从而在首次扫描中, 激活状态1。延时2秒后, T37导致切换到状态2。切换使状态1停止, 激活状态2。



分支控制

在许多实例中，一个顺序控制状态流必须分成两个或多个不同分支控制状态流。当一个控制状态流分离成多个分支时，所有的分支控制状态流必须同时激活，如图 6-33所示。

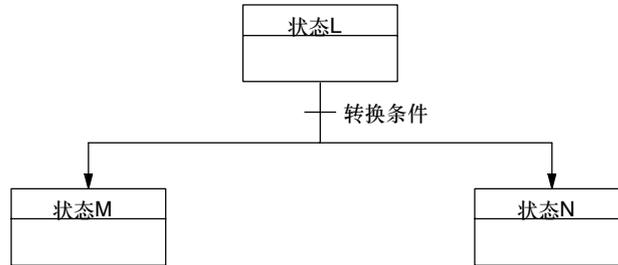
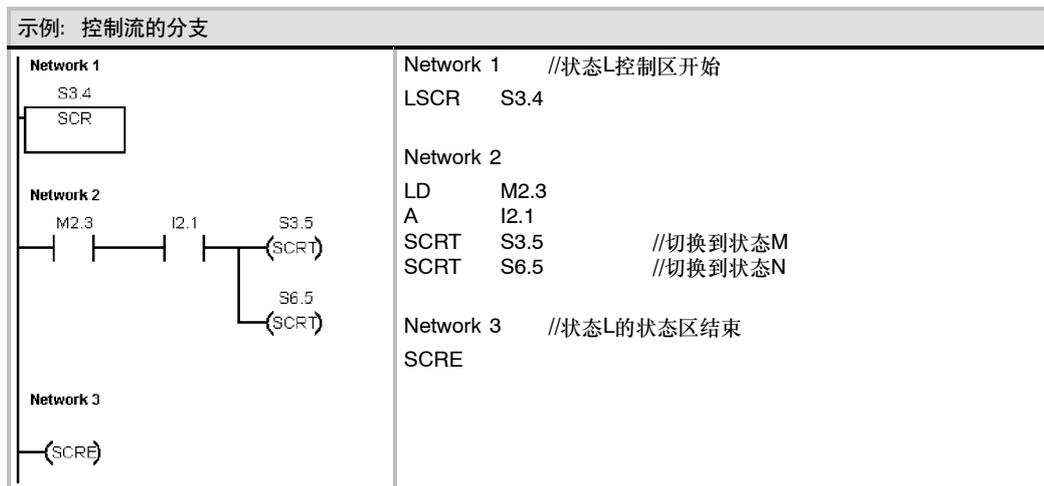


图6-33 控制流的分支

使用多条由相同转移条件激活的SCRT指令，可以在一段SCR程序中实现控制流的分支，如下面的实例所示。



合并控制

与分支控制的情况类似，两个或者多个分支状态流必须合并为一个状态流。当多个状态流汇集成一个时，我们称之为合并。当控制流合并时，所有的控制流必须都完成，才能执行下一个状态。图 6-34给出了两个控制流合并的示意图。

在SCR程序中，通过从状态L转到状态L，以及从状态M转到状态M的方法实现控制流的合并。当状态L、M的SCR使能位为真时，即可激活状态N，如下例所示。

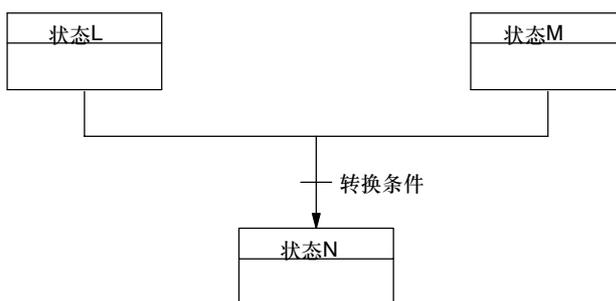
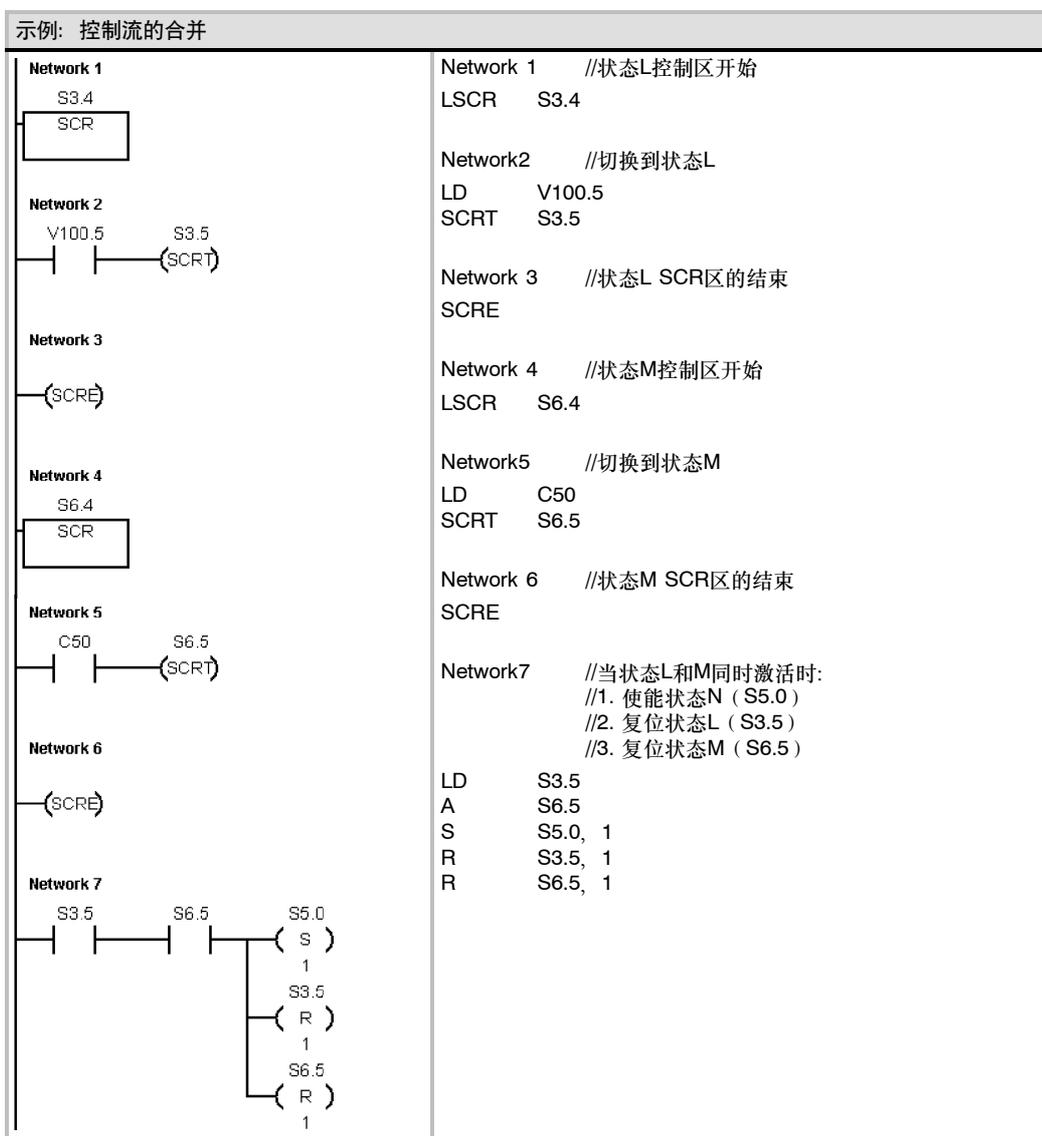


图6-34 控制流的合并



在有些情况下，一个控制流可能转入多个可能的控制流中的某一个。到底进入哪一个，取决于控制流前面的转移条件，哪一个首先为真，如图6-35所示。

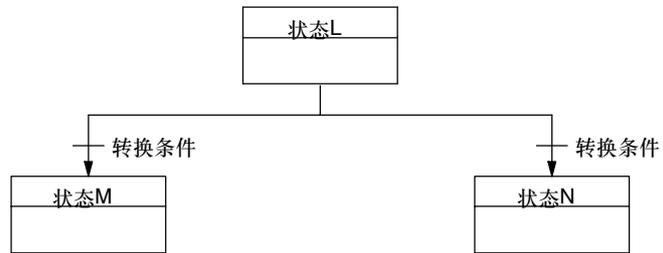


图6-35 条件转换控制流分支

示例: 条件转换	
<p>Network 1</p> <p>S3.4</p> <p>SCR</p> <p>Network 2</p> <p>M2.3 S3.5</p> <p>├──┤ ───┤ (SCRT)</p> <p>Network 3</p> <p>I3.3 S6.5</p> <p>├──┤ ───┤ (SCRT)</p> <p>Network 4</p> <p>(SCRE)</p>	<p>Network 1 //状态L控制区开始</p> <p>LSCR S3.4</p> <p>Network 2 //切换到M状态</p> <p>LD M2.3</p> <p>SCRT S3.5</p> <p>Network 3 //切换到N状态</p> <p>LD I3.3</p> <p>SCRT S6.5</p> <p>Network 4 //状态L的SCR区结束</p> <p>SCRE</p>

诊断LED指令

如果输入参数IN的值为零，就将诊断LED置为OFF。如果输入参数IN的值大于零，就将诊断LED置为ON（黄色）。

当系统块中指定的条件为真或者用非零IN参数执行DIAG_LED指令时，CPU发光二极管（LED）标注的SF/DIAG可以被配置用于显示黄色。

系统块（配置LED）复选框选项：

- 当有一项在CPU内被强制时，SF/DIAGLED为ON（黄色）
- 当模块有I/O错误时，SF/DIAGLED为ON（黄色）

两个配置LED选项都不选中，将使SF/DIAG黄光只受DIAG_LED指令控制。CPU系统故障（SF）用红光指示。

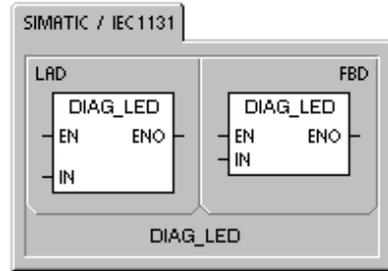
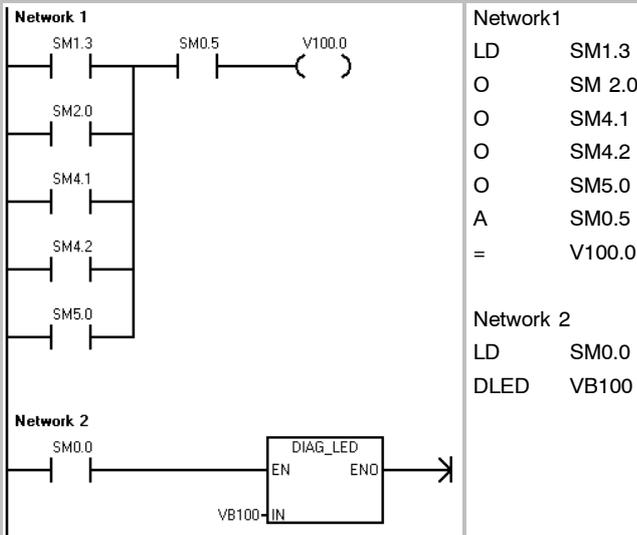


表6-60 诊断LED指令的有效操作数

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

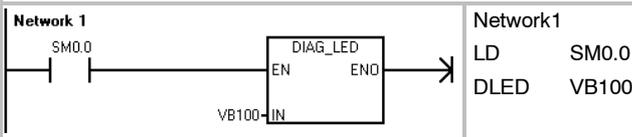
实例1诊断LED指令

当检测到错误时，诊断LED闪烁。
只要检测到5个错误条件中的一个，诊断LED就闪烁。



实例2诊断LED指令

当错误消失时，接通诊断LED。
当有错误代码在VB100中报告时，接通诊断LED



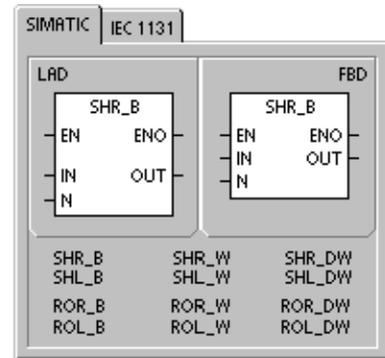
移位和循环指令

右移和左移指令

移位指令将输入值IN右移或左移N位，并将结果装载到输出OUT中。

移位指令对移出的位自动补零。如果位数N大于或等于最大允许值（对于字节操作为8，对于字操作为16，对于双字操作为32），那么移位操作的次数为最大允许值。如果移位次数大于0，溢出标志位（SM1.1）上就是最近移出的位值。如果移位操作的结果为0，零存储器位（SM1.0）置位。

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，符号位也被移动。

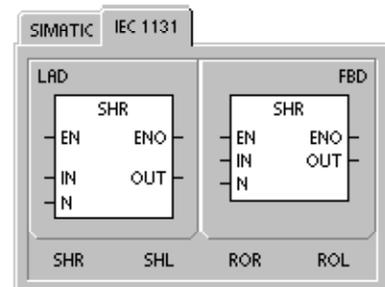


使ENO=0的错误条件:

- 0006 (间接寻址)

受影响的SM标志位:

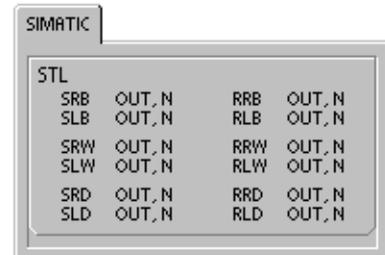
- SM1.0 (结果为0)
- SM1.1 (溢出)



循环右移和循环左移指令

循环移位指令将输入值IN循环右移或者循环左移N位，并将输出结果装载到OUT中。循环移位是圆形的。

如果位数N大于或者等于最大允许值（对于字节操作为8，对于字操作为16，对于双字操作为32），S7-200在执行循环移位之前，会执行取模操作，得到一个有效的移位次数。移位位数的取模操作的结果，对于字节操作是0到7，对于字操作是0到15，而对于双字操作是0到31。



如果移位次数为0，循环移位指令不执行。如果循环移位指令执行，最后一位的值会复制到溢出标志位（SM1.1）。

如果移位次数不是8（对于字节操作）、16（对于字操作）和32（对于双字操作）的整数倍，最后被移出的位会被复制到溢出标志位（SM1.1）。当要被循环移位的值是零时，零标志位（SM1.0）被置位。

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，符号位也被移位。

使ENO=0的错误条件:

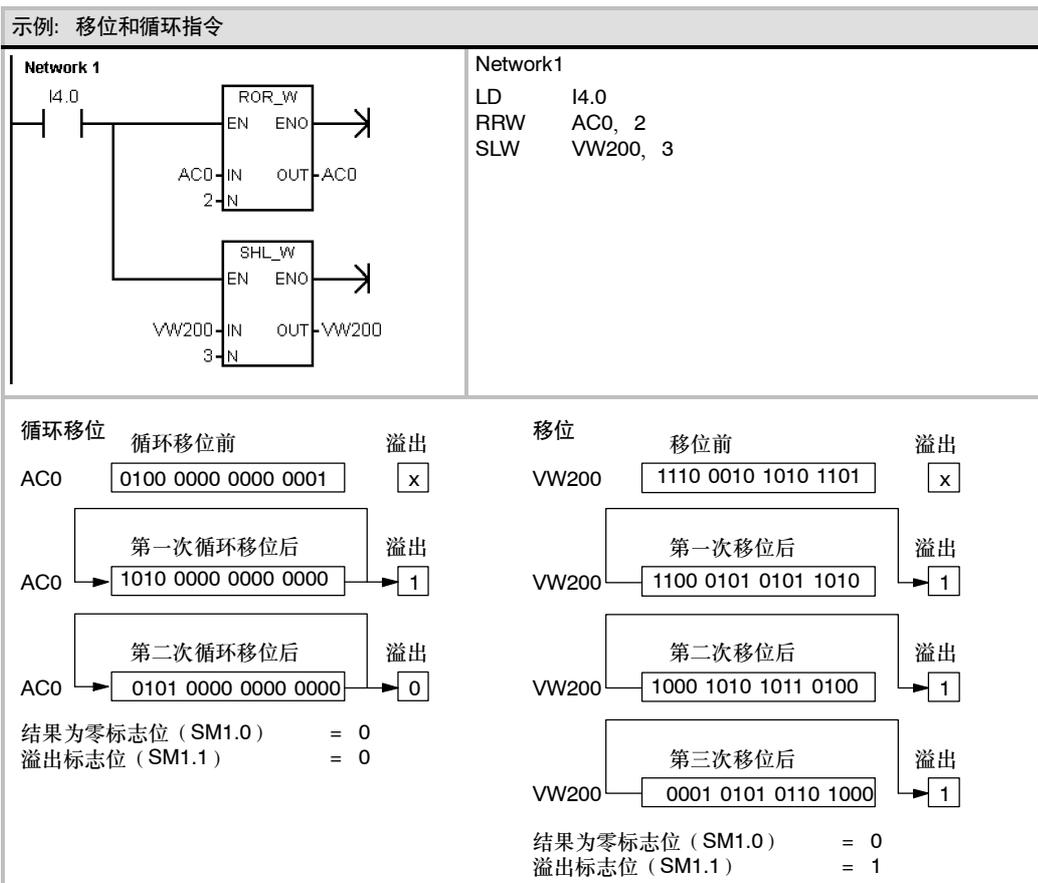
- 0006 (间接寻址)

受影响的SM标志位:

- SM1.0 (结果为0)
- SM1.1 (溢出)

表6-61 移位和循环移位指令的有效操作数

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



移位寄存器指令

移位寄存器指令将一个数值移入移位寄存器中。移位寄存器指令提供了一种排列和控制产品流或者数据的简单方法。使用该指令，每个扫描周期，整个移位寄存器移动一位。

移位寄存器指令把输入的DATA数值移入移位寄存器。其中，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方向（正向移位=N，反向移位=-N）。

SHRB指令移出的每一位都被放入溢出标志位（SM1.1）。

这条指令的执行取决于最低有效位（S_BIT）和由长度（N）指定的位数。

使ENO=0的错误条件:

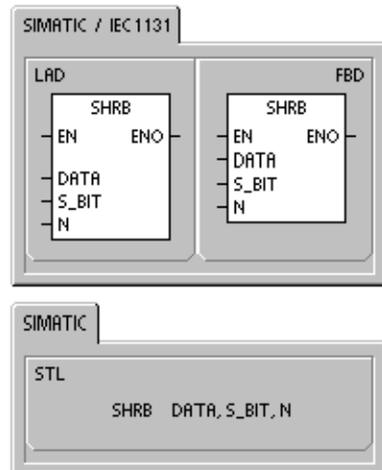
- 0006（间接寻址）
- 0091（操作数超出范围）
- 0092（计数区错误）

受影响的SM标志位:

- SM1.1（溢出）

表6-62 移位寄存器指令的有效操作数

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、常数



位移位寄存器的最高位 (MSB.b) 可通过下面公式计算求得:

$$MSB.b = [(S_BIT \text{的字节号}) + ([N] - 1 + (S_BIT \text{的位号})) / 8] \text{[除8的余数]}$$

例如: 如果S-BIT是V33.4, N是14, 那么MSB.b是V35.1, 或:

$$\begin{aligned} MSB.b &= V33 + ([14] - 1 + 4) / 8 \\ &= V33 + 17 / 8 \\ &= V33 + 2 \text{ (余数为1)} \\ &= V35.1 \end{aligned}$$

当反向移动时, N为负值, 输入数据从最高位移入, 最低位 (S_BIT) 移出。移出的数据放在溢出标志位 (SM1.1) 中。

当正向移动时, N为正值, 输入数据从最低位(S_BIT) 移入, 最高位移出。移出的数据放在溢出标志位 (SM1.1) 中。

移位寄存器的最大长度为64位, 可正可负。图6-36中给出了N为正和负两种情况下的移位过程。

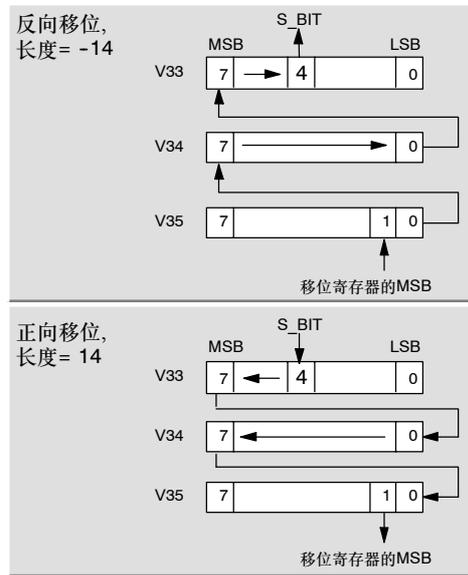
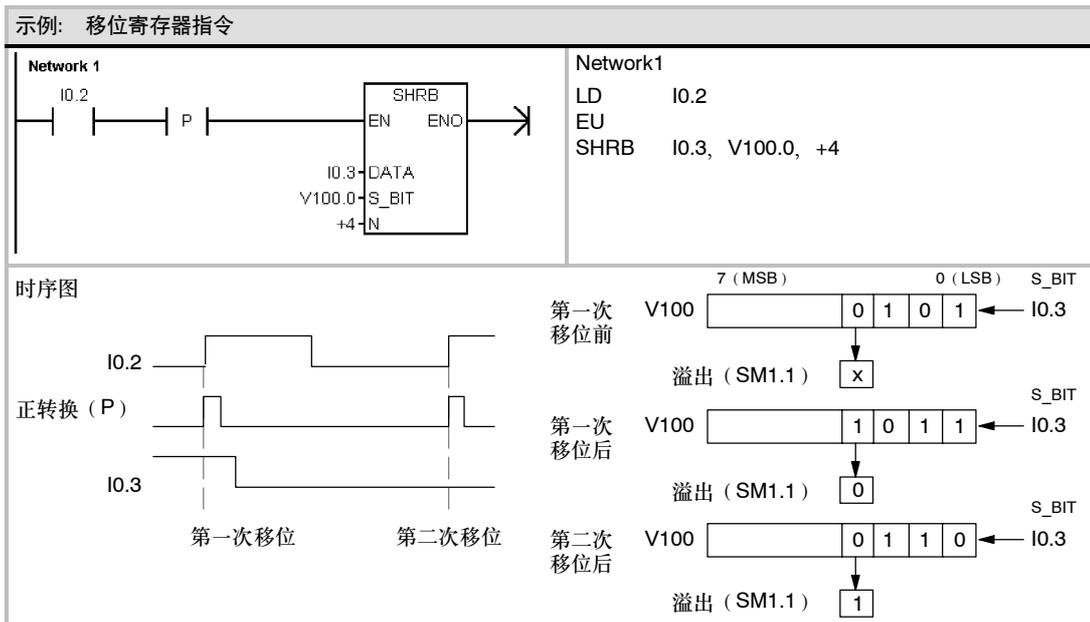


图6-36 移位寄存器的入口和出口



字节交换指令

字节交换指令用来交换输入字IN的高字节和低字节。

使ENO=0的错误条件:

- 0006 (间接寻址)

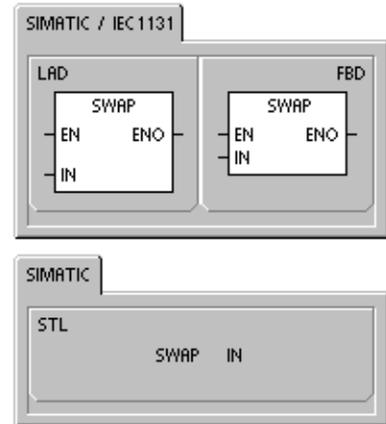
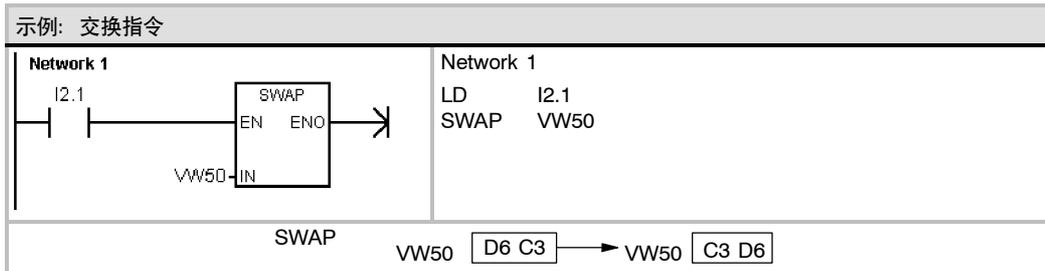


表6-63 字节交换指令的有效操作数

输入/输出	数据类型	操作数
IN	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、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中指定的字符串的长度值。

字符串复制

字符串复制指令（SCPY）将IN中指定的字符串复制到OUT中。

字符串连接

字符串连接指令（SCAT）将IN中指定的字符串连接到OUT中指定字符串的后面。

SM标志位和ENO

对于字符串长度、字符串复制和字符串连接指令，下列条件影响ENO。

使ENO=0的错误条件:

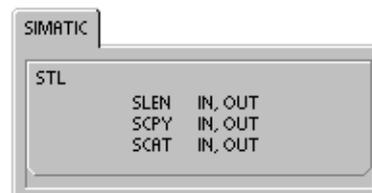
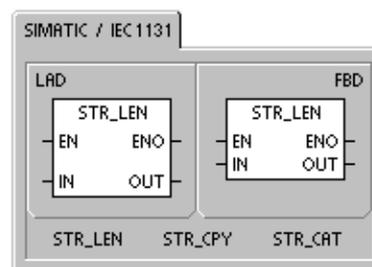
- 0006（间接寻址）
- 0091（操作数超出范围）

表6-64 字符串长度指令的有效操作数

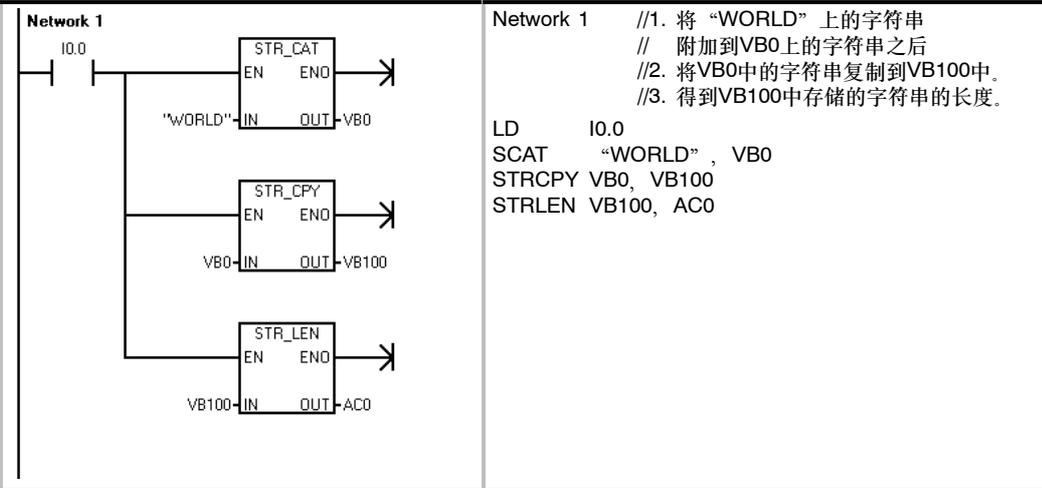
输入/输出	数据类型	操作数
IN	STRING	VB、LB、*VD、*LD、*AC、字符串常数
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC

表6-65 字符串复制和字符串连接

输入/输出	数据类型	操作数
IN	STRING	VB、LB、*VD、*LD、*AC、字符串常数
OUT	STRING	VB、LB、*VD、*AC、*LD



示例: 字符串连接、字符串复制和字符串长度指令程序举例



程序执行前

VB0						VB6	
6	'H'	'E'	'L'	'L'	'O'	' '	' '

程序执行后

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'	

AC0
11

从字符串中复制子字符串

从字符串中复制子字符串指令（SSCPY）从INDX指定的字符号开始，将IN中存储的字符串中的N个字符复制到OUT中。

使ENO=0的错误条件:

- 0006（间接寻址）
- 0091（操作数超出范围）
- 009B（INDX=0）

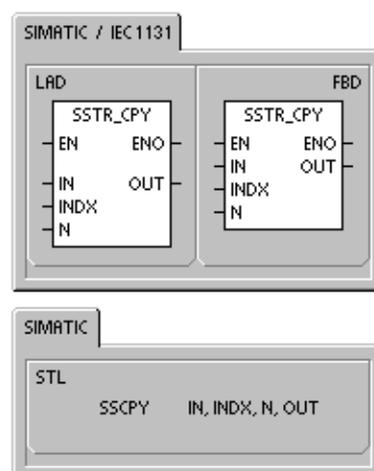


表6-66 从字符串中复制子字符串指令

输入/输出	数据类型	操作数
IN	STRING	VB、LB、*VD、*LD、*AC、字符串常数
OUT	STRING	VB、LB、*VD、*LD、*AC
INDX、N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数

示例: 复制子字符串指令

<p>Network 1</p>	<p>Network1 //从VB0中字符串的第7个字符开始， //复制5个字符到VB20开始的新字符串。</p> <pre>LD I0.0 SSCPY VB0, 7, 5, VB20</pre>															
<p>程序执行前</p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <td style="width: 10%;">VB0</td> <td style="width: 10%;">'1'</td> <td style="width: 10%;">'1'</td> <td style="width: 10%;">'H'</td> <td style="width: 10%;">'E'</td> <td style="width: 10%;">'L'</td> <td style="width: 10%;">'L'</td> <td style="width: 10%;">'O'</td> <td style="width: 10%;">' '</td> <td style="width: 10%;">'W'</td> <td style="width: 10%;">'O'</td> <td style="width: 10%;">'R'</td> <td style="width: 10%;">'L'</td> <td style="width: 10%;">'D'</td> <td style="width: 10%;">VB11</td> </tr> </table>		VB0	'1'	'1'	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'	VB11
VB0	'1'	'1'	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'	VB11		
<p>程序执行后</p> <table border="1" style="width: 50%; text-align: center; border-collapse: collapse;"> <tr> <td style="width: 10%;">VB20</td> <td style="width: 10%;">5</td> <td style="width: 10%;">'W'</td> <td style="width: 10%;">'O'</td> <td style="width: 10%;">'R'</td> <td style="width: 10%;">'L'</td> <td style="width: 10%;">'D'</td> <td style="width: 10%;">VB25</td> </tr> </table>		VB20	5	'W'	'O'	'R'	'L'	'D'	VB25							
VB20	5	'W'	'O'	'R'	'L'	'D'	VB25									

字符串搜索

字符串搜索指令 (SFND) 在 IN1 字符串中寻找 IN2 字符串。由 OUT 指定搜索的起始位置。如果在 IN1 中找到了与 IN2 中字符串相匹配的一段字符，则 OUT 中会存入这段字符串中首个字符的位置。如果没有找到，OUT 被清 0。

使 ENO=0 的错误条件:

- 0006 (间接寻址)
- 0091 (操作数超出范围)
- 009B (INDX=0)

字符搜索

字符搜索指令 (CFND) 在 IN1 字符串中寻找 IN2 字符串中的任意字符。由 OUT 指定搜索的起始位置。如果找到了匹配的字符，字符的位置被写入 OUT 中。如果没有找到，OUT 被清 0。

使 ENO=0 的错误条件:

- 0006 (间接寻址)
- 0091 (操作数超出范围)
- 009B (INDX=0)

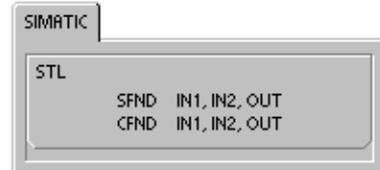
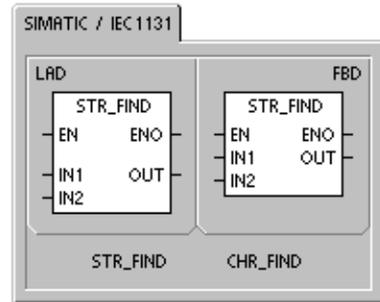
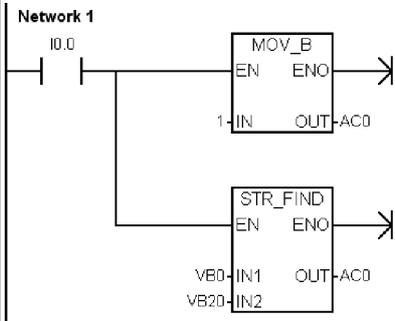


表6-67 字符串搜索和字符搜索指令的有效操作数

输入/输出	数据类型	操作数
IN1、IN2	STRING	VB、LB、*VD、*LD、*AC、字符串常数
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC

示例: 字符串搜索

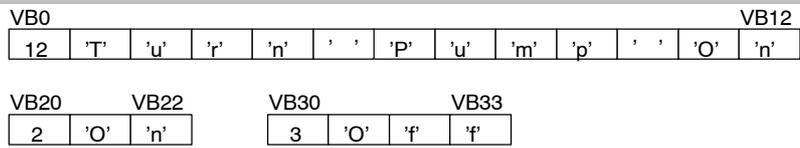
在以下例子中, 用存储在VB0中的字符串作为泵的启/停命令。字符串 'On' 存储在VB20中, 字符串 'Off' 存储在VB30中。搜索结果在AC0中 (OUT参数)。如果结果不是0, 就说明在命令字符串中找到了字符串 'On' (VB12)。



```

Network 1 //1. 置AC0为1。
           // (AC0用作OUT参数。)
           //2. 在VB0字符串中搜索VB20
           // 字符串 ( 'On' ),
           // 从首个字符开始
           // (AC0=1)。

LD      I0.0
MOVB   1, AC0
SFND   VB0, VB20, AC0
    
```

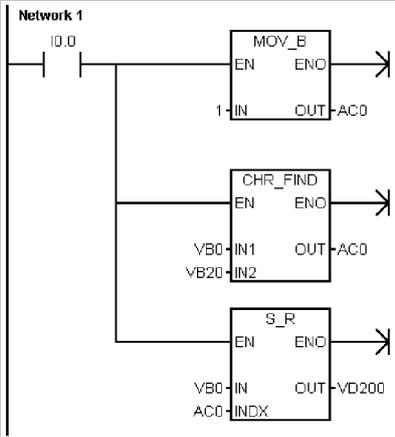


如果找到VB20中的字符串: AC0 11

如果没有找到VB20中的字符串: AC0 0

示例: 字符串搜索指令

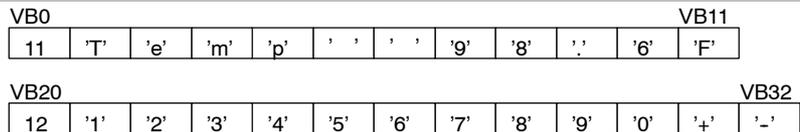
在以下例子中, 存储在VB0的字符串包含温度值。存储在VB20中的字符串包括所有的数字 (包括+和-), 用于识别字符串中的温度值。该例子程序在字符串中找到数字的起始位置, 并将其转换为实数, 温度值存放在VD200中。



```

Network 1 //1. 置AC0为1。
           // (AC0用作OUT参数并指向字符串的首个
           // 字符。)
           //2. 在VB0字符串中寻找数字字符。
           //3. 将字符串转换为实数。

LD      I0.0
MOVB   1, AC0
CFND   VB0, VB20, AC0
STR    VB0, AC0, VD200
    
```



VB0中存储的温度的起始地址: AC0 7

温度的实数值: VD200 98.6

表指令

填表

ATT指令向表 (TBL) 中增加一个数值 (DATA)。表中第一个数是最大填表数 (TL)，第二个数是实际填表数 (EC)，指出已填入表的数据个数。新的数据填加在表中上一个数据的后面。每向表中填加一个新的数据，EC会自动加1。

一个表最多可以有100条数据。

使ENO=0的错误条件:

- SM1.4 (表溢出)
- 0006 (间接寻址)
- 0091 (操作数超出范围)

受影响的SM标志位:

- 如果表出现溢出，SM1.4会置1。

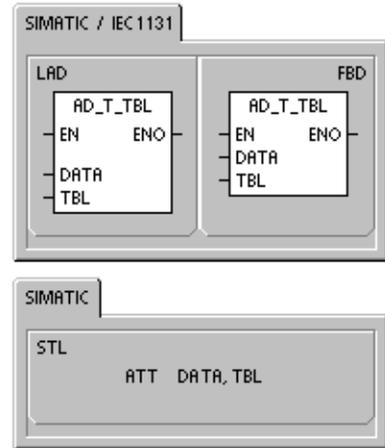
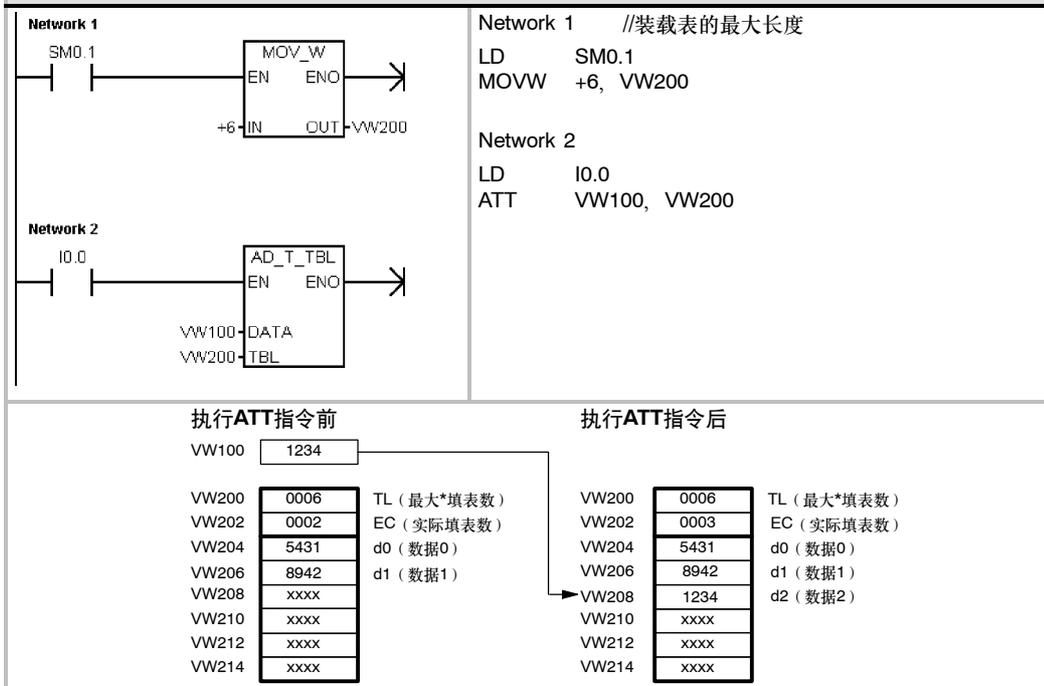


表6-68 表指令的有效操作数

输入/输出	数据类型	操作数
DATA	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
TBL	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、*VD、*LD、*AC

示例: 填表指令



先进先出和后进先出

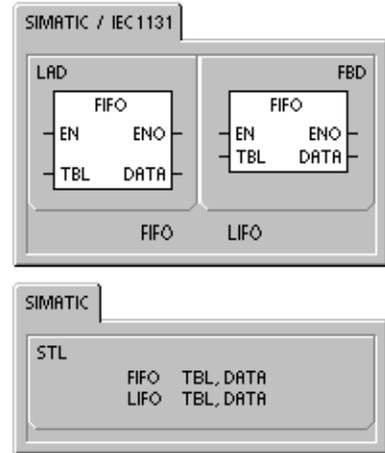
一个表可以有最多100条数据。

先进先出

先进先出（FIFO）指令从表（TBL）中移走第一个数据，并将此数输出到DATA。剩余数据依次上移一个位置。每执行一条本指令，表中的数据数减1。

后进先出

后进先出（LIFO）指令从表（TBL）中移走最后一个数据，并将此数输出到DATA。每执行一条本指令，表中的数据数减1。



使ENO=0的错误条件:

- SM1.5（表空）
- 0006（间接寻址）
- 0091（操作数超出范围）

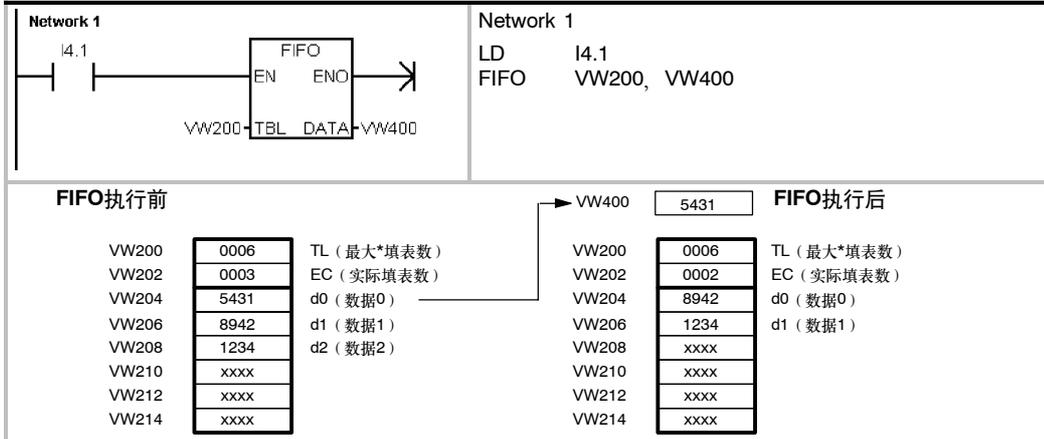
受影响的SM标志位:

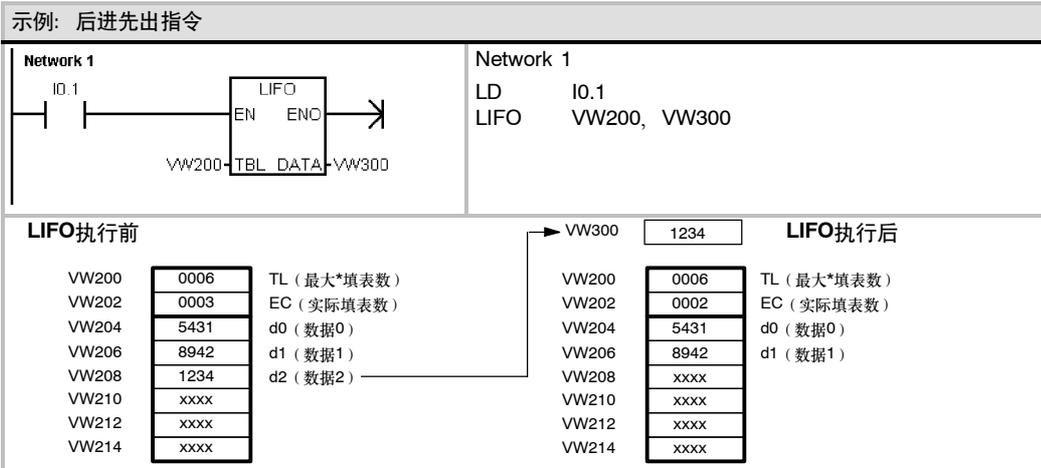
- 当您试图从一个空表中删除一条数据时，SM1.5会置1。

表6-69 先进先出和先进后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
DATA	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC

示例: 先进先出指令





内存填充

存储器填充指令 (FILL) 用输入值 (IN) 填充从输出 (OUT) 开始的N个字的内容。

N的范围从1到255。

使ENO=0的错误条件:

- 0006 (间接寻址)
- 0091 (操作数超出范围)

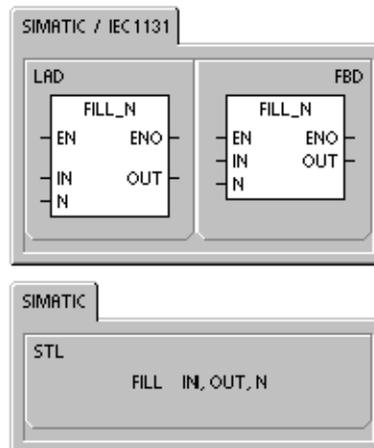
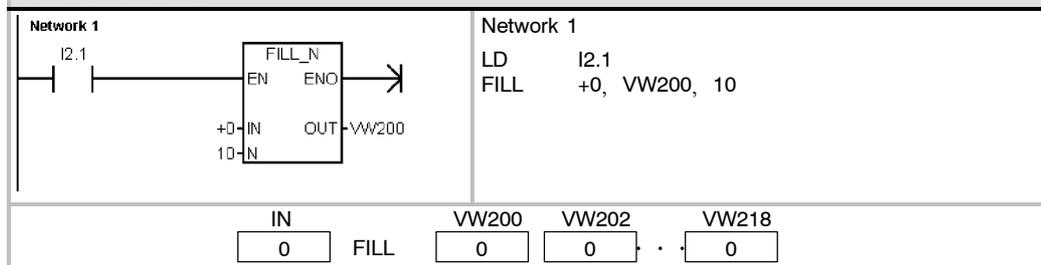


表6-70 内存填充指令的有效操作数

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

示例: 内存填充指令



查表

查表指令 (FND) 搜索表, 以查找符合一定规则的数据。查表指令从INDX开始搜索表 (TBL), 寻找符合PTN和条件 (=、<>、<或>) 的数据。命令参数CMD是一个1~4的数值, 分别代表=、<>、<和>。

如果发现了一个符合条件的数据, 那么INDX指向表中该数的位置。为了查找下一个符合条件的数据, 在激活查表指令前, 必须先对INDX加1。如果没有发现符合条件的数据, 那么INDX等于EC。

一个表可以有最多100条数据。数据条标号从0到99。

使ENO=0的错误条件:

- 0006 (间接寻址)
- 0091 (操作数超出范围)

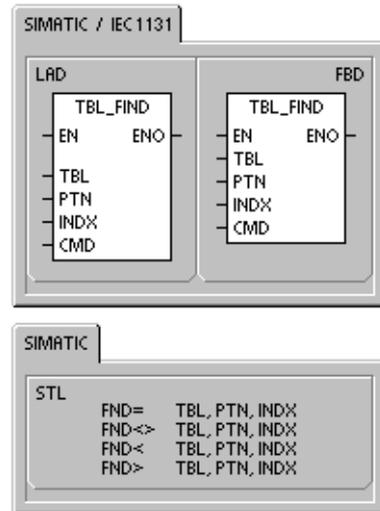


表6-71 查表指令的有效操作数

输入/输出	数据类型	操作数
TBL	WORD	IW、QW、VW、MW、SMW、T、C、LW、*VD、*LC、*AC
PTN	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
INDX	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*LD、*AC
CMD	BYTE	(常数) 1: 等于 (=), 2: 不等于 (<>), 3: 小于 (<), 4: 大于 (>)



提示

当您用FND指令查找由指令ATT、LIFO和FIFO生成的表时, 实际填表数 (EC) 和输入数据相符, 直接对应。最大填表数 (TL) 对ATT、LIFO和FIFO指令是必需的, 但FND指令并不需要它。参见图6-37。

因此, FND指令的操作数SRC是一个字地址 (指向EC), 比相应的ATT、LIFO或FIFO指令的操作数TABLE要高2个字节。

ATT、LIFO和FIFO指令的表格式

VW200	0006	TL (最大*填表数)
VW202	0006	EC (实际填表数)
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)

FND查表指令的表格式

VW202	0006	EC (实际填表数)
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 FND指令与ATT、LIFO和FIFO指令所使用的表格式上的差异

示例: 查表指令

Network 1

Network 1

```
LD I2.1
FND= VW202, 16#3130, AC1
```

当I2.1接通时，搜索表，寻找和3130 HEX相等的值。

VW202	0006	EC (实际填表数)
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)

如果表是用ATT、LIFO和FIFO指令创建的，VW200包含了允许的最大填表数，而Find指令不需要它。

执行查表 AC1 从表头开始查找，AC1必须置为0。

执行查表 AC1 AC1中保存了第1个符合查表条件的数据编号 (d2)。

执行查表 AC1 查表中剩余数据前，INDX加1。

执行查表 AC1 AC1中保存了第2个符合查表条件的数据编号 (d4)。

执行查表 AC1 查表中剩余数据前，INDX加1。

执行查表 AC1 AC1中保存了已填表数。整个表已经查完，没有发现另外的匹配数据。

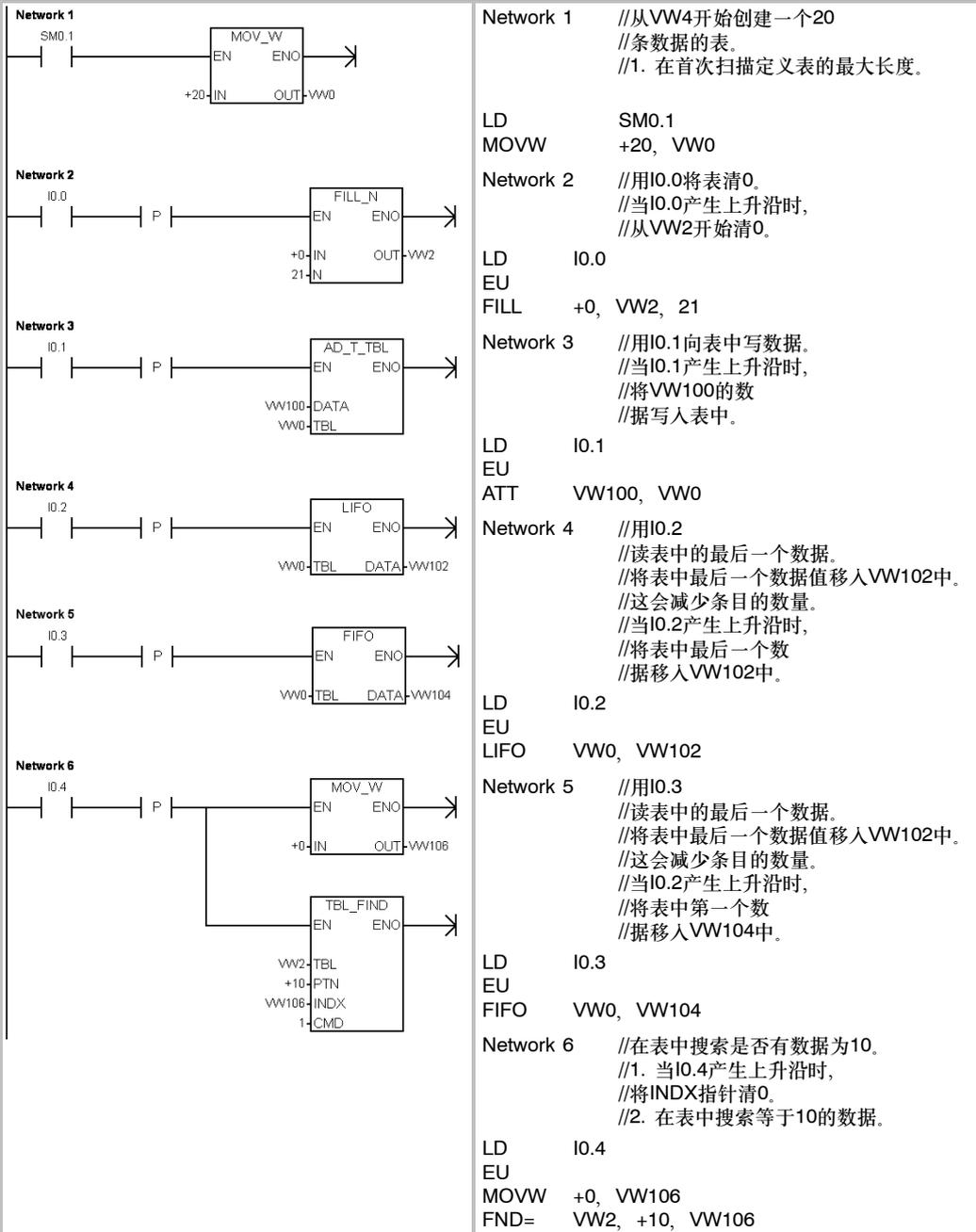
再次查表前，INDX的值必须复位到0。

示例: 创建一个表

下列程序创建一个包含20条数据的表。存储区中的第一个数据为表的长度（在本例中为20）。存储区中的第二个数据为表中数据的实际个数。其它存储区单元为数据。一个表可以有最多100条数据。其中不包括定义表长度和实际数据个数的两个单元（在本例中为VW0和VW2）。当CPU执行第一条指令时，表中的实际数据个数（VW2）会自动增或者减。

在使用表之前，必须为表指定数据的最多个数。否则您将无法在表中插入数据。同时，要确保使用边沿触发来激活读写指令。

在查表之前，INDX（VW106）必须清0。如果找到匹配的数据，INDX中会存入表中的条目号；如果没有找到，INDX中为实际数据个数（VW2）。



定时器指令

SIMATIC定时器指令

接通延时定时器 有记忆的接通延时定时器

接通延时定时器（TON）和有记忆的接通延时定时器在使能输入接通时计时。定时器号（Txx）决定了定时器的分辨率，并且分辨率现在已经在指令盒上标出了。

断开延时定时器

断开延时定时器用于在输入断开后延时一段时间断开输出。定时器号（Txx）决定了定时器的分辨率，并且分辨率现在已经在指令盒上标出了。

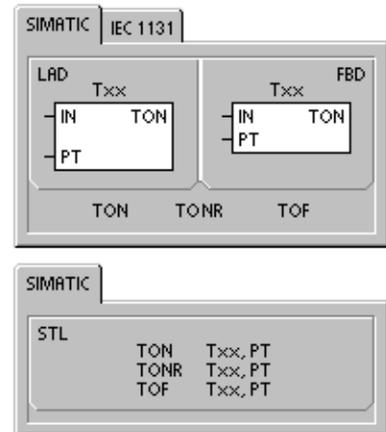


表6-72 SIMATIC定时器指令的有效操作数

输入/输出	数据类型	操作数
TXX	WORD	常数 (T0到T255)
IN	BOOL	I、Q、V、M、SM、S、T、C、L、能流
PT	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数



提示

不能将同一个定时器号同时用作TOF和TON。例如，不能够既有TON T32又有TOF T32。

如表6-73所示，三类定时器用于执行不同类型的定时任务：

- 接通延时定时器（TON）用于单一间隔的定时
- 有记忆接通延时定时器（TONR）用于累计许多时间间隔
- 断开延时定时器（TOF）用于关断或者故障事件后的延时（例如：在电机停后，需要冷却电机）

表6-73 定时器指令的操作数

定时器类型	当前值>=预设值	使能输入（IN）的状态	上电周期/首次扫描
TON	定时器位ON， 当前连续计数到32767	ON: 当前值计数时间 OFF: 定时器位OFF，当前值=0	定时器位OFF， 当前值=0
TONR	定时器位ON， 当前连续计数到32767	ON: 当前值计数时间 OFF: 定时器位和当前值保持最后状态	定时器位OFF， 当前值保持 ¹
TOF	定时器位OFF， 当前值=预设值，停止计数	ON: 定时器位ON，当前值=0 OFF: 发生ON到OFF的跳变之后，定时器计数	定时器位OFF， 当前值=0

¹ 有记忆定时器的当前值通过电源扫描选择有记忆，有关S7-200 CPU有记忆存储器的详细内容参阅第4章。



编程提示

请参考资料光盘上应用示例中使用接通延时定时器 (TON) 的示例程序。见Tip31。

当使能输入接通时，接通延时定时器和有记忆接通延时定时器开始计时，当前值 (Txxx) 大于等于预设值时，该定时器位被置位。

- 当使能输入断开时，清除接通延时定时器的当前值，而对于有记忆接通延时定时器，其当前值保持不变。
- 可以用有记忆接通延时定时器累计输入信号的接通时间，利用复位指令 (R) 清除其当前值。
- 当达到预设时间后，接通延时定时器和有记忆接通延时定时器继续计时，一直计到最大值 32767。

断开延时定时器 (TOF) 用来在输入断开后延时一段时间断开输出。当使能输入接通时，定时器位立即接通，并把当前值设为0。当输入断开时，定时器开始定时，直到达到预设的时间。

- 当达到预设时间时，定时器位断开，并且停止计时当前值。当输入断开的时间短于预设时间时，定时器位保持接通。
- TOF指令必须用输入信号的接通到断开的跳变启动计时。
- 如果TOF定时器在顺控 (SCR) 区，而且顺控区没有启动，TOF定时器的当前值设置为0，定时器位设置为断开，当前值不计时。



提示

可以只使用复位 (R) 指令来复位TONR。还可以使用复位指令去复位TON或TOF中的任何一个。复位指令执行如下的操作:

- 定时器位=OFF
- 定时器当前位置=0

TONR定时器只能通过复位指令进行复位操作。复位后，为了再启动，TOF定时器需要使能输入有一个从ON到OFF的跳变。

为定时器选择分辨率

定时器对时间间隔记数。定时器的分辨率 (时基) 决定了每个时间间隔的时间长短。例如：一个以10ms为时基的延时接通定时器，在使能位接通后，以10ms的时间间隔计数，10ms的定时器计数值为50代表500ms。SIMATIC定时器有三种分辨率：1ms、10ms和100ms。如表6-74所示，定时器号决定了定时器的分辨率。

**提示**

为确保时间间隔的最小值，预置值必须比它大1。例如：为确保最小时间间隔2100ms，要将100ms定时器的预置值PV设为22。

表6-74 定时器号和分辨率

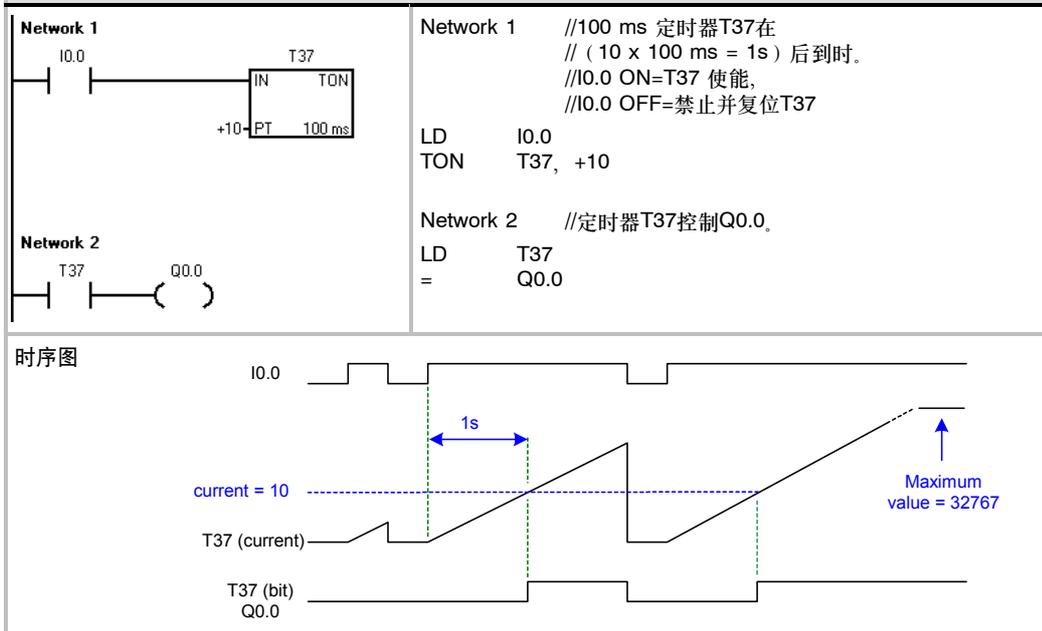
定时器类型	用毫秒 (ms) 表示的分辨率	用秒 (s) 表示的最大值	定时器号
TONR	1ms	32.767s	T0, T64
	10ms	327.67s	T1 - T4, T65 - T68
	100ms	3276.7s	T5 - T31, T69 - T95
TON、TOF	1ms	32.767s	T32, T96
	10ms	327.67s	T33 - T36, T97 - T100
	100ms	3276.7s	T37 - T63, T101 - T255

分辨率对定时器的影响

对于1ms分辨率的定时器来说，定时器位和当前值的更新不与扫描周期同步。对于大于1 ms的程序扫描周期，定时器位和当前值在一次扫描内刷新多次。

对于10ms分辨率的定时器来说，定时器位和当前值在每个程序扫描周期的开始刷新。定时器位和当前值在整个扫描周期过程中为常数。在每个扫描周期的开始会将一个扫描累计的时间间隔加到定时器当前值上。

对于100ms分辨率的定时器来说，定时器位和当前值在指令执行时刷新。因此，为了使定时器保持正确的定时值，要确保在一个程序扫描周期中，只执行一次100ms定时器指令。

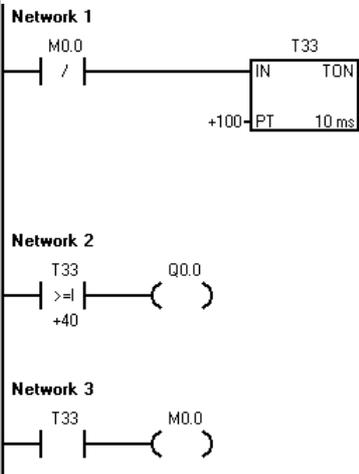
示例：SIMATIC接通延时定时器



提示

为了确保在每一次定时器达到预设值时，自复位定时器的输出都能接通一个程序扫描周期，用一个常闭触点来代替定时器位作为定时器的使能输入。

示例: SIMATIC自复位接通延时定时器



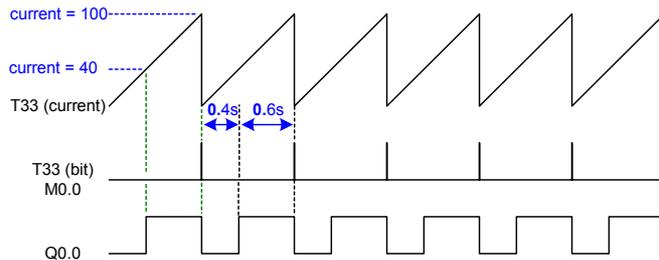
```

Network 1 //10 ms 定时器T33在
           //( 100 x 10 ms = 1s ) 后到时。
           //M0.0脉冲过快，以致在状态视图中
           //无法监视
LDN      M0.0
TON      T33, +100

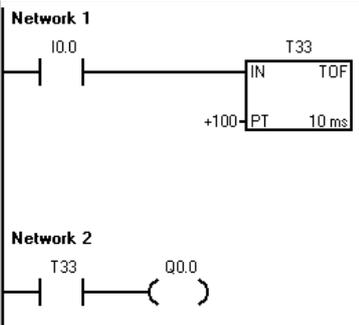
Network 2 //比较指令为真的时间较长，
           //可以在状态表中
           //监视，Q0.0的占空比为40%。
LDW>=   T33, +40
=        Q0.0

Network 3 //T33 (位) 的脉冲过窄，
           //在状态表中无法监视。
           //在1秒后复位M0.0。
LD       T33
=        M0.0
    
```

时序图



示例: SIMATIC断开延时定时器

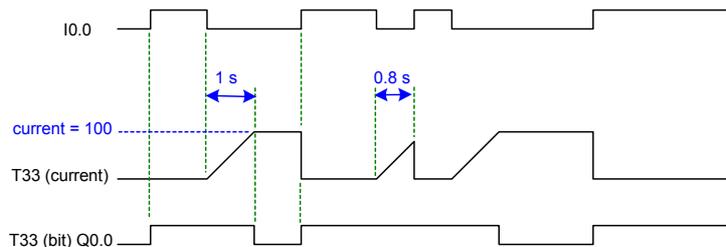


```

Network 1 //10ms定时器T33在1秒后到时。
           //I0.0关断使能T33。
           //I0.0接通T33复位。
LD       I0.0
TOF      T33, +100

Network 2 //定时器T33用其输出位控制Q0.0。
LD       T33
=        Q0.0
    
```

时序图



示例: SIMATIC有记忆的接通延时定时器

Network 1

Network 2

Network 3

Network 1 //10 ms TONR定时器T1在
//PT= (100 x 10 ms=1s) 后到时。

```
LD I0.0
TONR T1, +100
```

Network 2 //T1位控制Q0.0。
//1秒后T1使Q0.0接通。

```
LD T1
= Q0.0
```

Network 3 //TONR定时器必须用复位
//指令才能复位。
//当I0.1接通时，复位T1。

```
LD I0.1
R T1, 1
```

时序图

IEC定时器指令

接通延时定时器

当使能输入接通时，接通延时定时器（TON）指令对时间进行计数。

断开延时定时器

断开延时定时器（TOF）用于在输入断开后，延时一段时间后断开输出。

脉冲定时器

脉冲定时器（TP）以指定的周期产生脉冲。

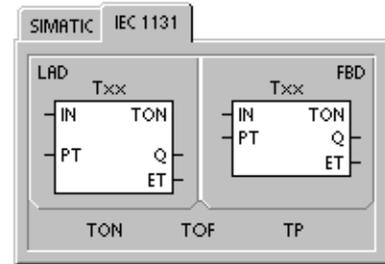


表6-75 IEC定时器指令的有效操作数

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、能流
PT	INT	IW、QW、VW、MW、SMW、SW、LW、AC、AIW、*VD、*LD、*AC、常数
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

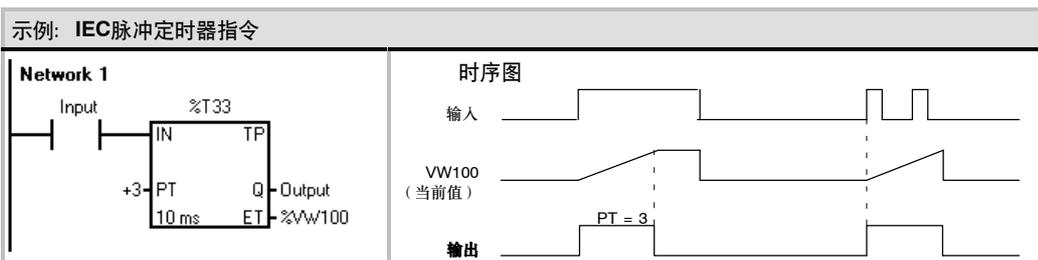
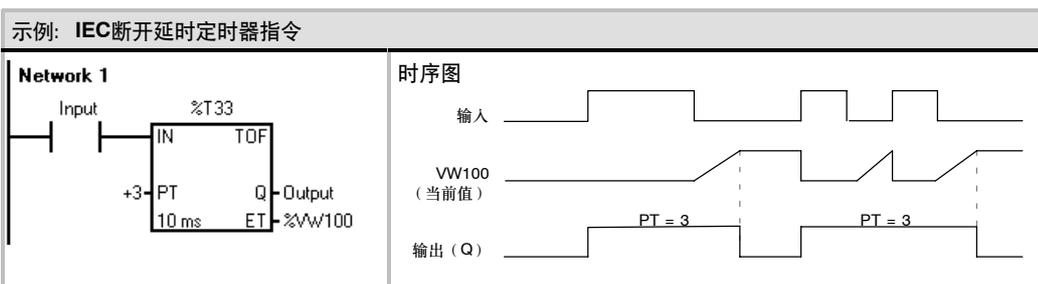
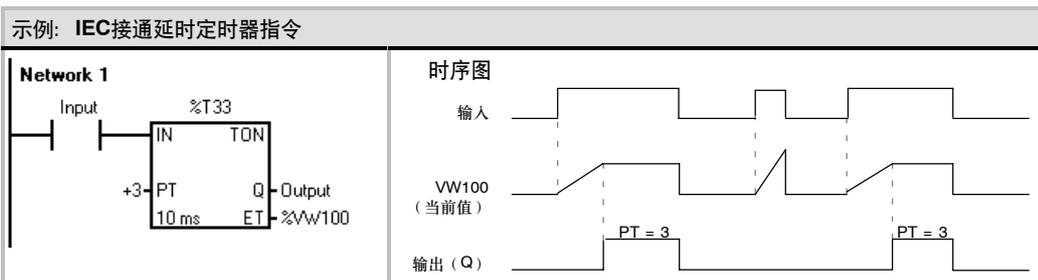
一个定时器号不能同时用于TOF、TP和TON，例如，不能够既有TON T32又有TOF T32。

- 当使能输入端（IN）为1时，接通延时定时器功能块开始启动定时，一直到预置值。当经过时间（ET）大于等于预置值（PT）时，定时器输出位（Q）变为1。当使能输入端（IN）为0时，定时器输出复位。当预置时间（PT）到达时，定时停止并且定时器不工作。
- 当输入断开时，断开延时定时器功能块把输出断开的延迟时间延迟一个固定的时间。当使能输入（IN）变为0时，定时器的值又变为预置值。当经过时间（ET）大于等于预置时间（PT）时，定时器输出位（Q）接通。一旦达到预置值，定时器的输出位变为0，一直保持到使能输入（IN）再变为1。如果使能输入（IN）变为0的持续时间小于预置值（PT），定时器的输出位一直保持接通。
- 脉冲定时器用于产生一个指定宽度的脉冲。当使能输入（IN）变为1时，输出位（Q）接通，在预置时间内输出位保持接通。一旦经过时间（ET）达到预置时间（PT），输出位变为0。经过的时间会被保存到使能输入断开。当输出接通时，它会一直保持到脉冲周期结束。

当前值的每次计数为时基的倍数。例如：以10ms为时基的定时器，计数值为50，代表500ms。IEC定时器（TON、TOF和TP）有三种分辨率。分辨率由定时器号决定，如表6-76所示。

表6-76 IEC定时器的分辨率

用毫秒（ms）表示的分辨率	用秒（s）表示的最大值	定时器号
1ms	32.767 s （0.546 分钟）	T32,T96
10ms	327.67s	T33 - T36, T97 - T100
100ms	3276.7s	T37 - T63, T101 - T255



时间间隔定时器

触发时间间隔

触发时间间隔（BITIM）指令读内置的1毫秒计数器的当前值，并将此值存储到OUT中。双字毫秒值的最大定时间隔是2的32次幂或49.7天。

计算时间间隔

计算时间间隔（CITIM）指令计算当前时间和IN提供的值之间的时间差。时间差被存储在OUT中。双字毫秒值的最大定时间隔是2的32次幂或49.7天。依据于BITIM指令执行的时间，CITIM自动处理在最大间隔内发生的1毫秒定时器翻转。

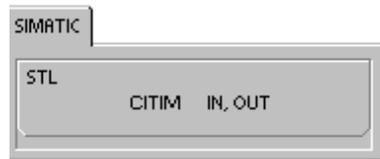
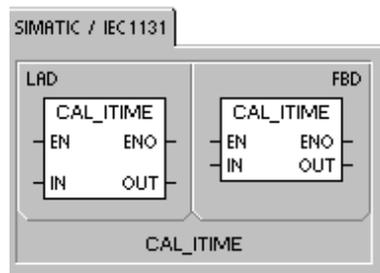
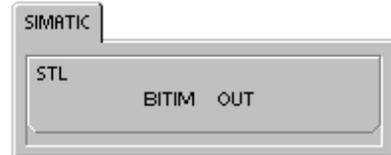
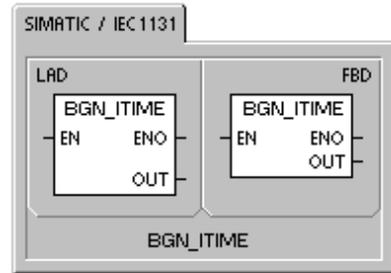


表6-77 间隔定时器指令的有效操作数

输入/输出	数据类型	操作数
IN	DWORD	VD、ID、QD、MD、SMD、SD、LD、HC、AC、*VD、*LD、*AC
OUT	DWORD	VD、ID、QD、MD、SMD、SD、LD、AC、*VD、*AC、*LD

示例: SIMATIC触发时间间隔和计算时间间隔

<p>Network 1</p>	<p>Network 1 //捕获Q0.0接通的时刻。</p> <pre>LD Q0.0 EU BITIM VD0</pre>
<p>Network 2</p>	<p>Network 2 // 计算Q0.0已经接通的时间。</p> <pre>LD Q0.0 CITIM VD0, VD4</pre>

子程序指令

子程序调用指令（CALL）将程序控制权交给子程序SBR_N。调用子程序时可以带参数也可以不带参数。子程序执行完成后，控制权返回到调用子程序的指令的下一条指令。

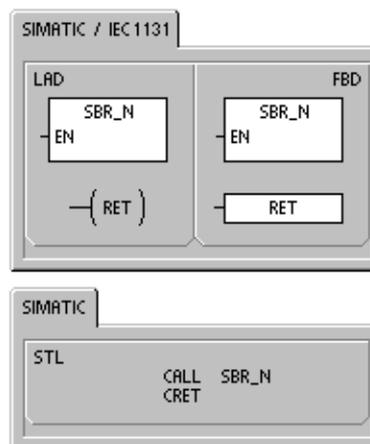
子程序条件返回指令（CRET）根据它前面的逻辑决定是否终止子程序。

要添加一个子程序可以在命令菜单中选择：**Edit > Insert > Subroutine**。

使ENO=0的错误条件：

- 0008（超过子程序嵌套最大限制）
- 0006（间接寻址）

在主程序中，可以嵌套调用子程序(在子程序中调用子程序)，最多嵌套8层。在中断服务程序中，不能嵌套调用子程序。



在被中断服务程序调用的子程序中不能再出现子程序调用。不禁止递归调用（子程序调用自己），但是当使用带子程序的递归调用时应慎重。

表6-78 子程序指令的有效操作数

输入/输出	数据类型	操作数
SBR_N	WORD	常数 对于CPU221, CPU222, CPU224: 对于CPU224XP和CPU226
IN	BOOL BYTE WORD、INT D W O R D、 DINT STRING	V、I、Q、M、SM、S、T、C、L、能流 VB、IB、QB、MB、SMB、SB、LB、AC、*VD、*LD、*AC ¹ 、常数 VW、T、C、IW、QW、MW、SMW、SW、LW、AC、AIW、*VD、*LD、*AC ¹ 、常数 VD、ID、QD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC ¹ 、&VB、&IB、&QB、&MB、&T、&C、&SB、&AI、&AQ、&SMB、常数 *VD、*LD、*AC、常数
输入/输出	BOOL BYTE WORD、INT D W O R D、 DINT	V、I、Q、M、SM ² 、S、T、C、L VB、IB、QB、MB、SMB ² 、SB、LB、AC、*VD、*LD、*AC ¹ VW、T、C、IW、QW、MW、SMW ² 、SW、LW、AC、*VD、*LD、*AC ¹ VD、ID、QD、MD、SMD ² 、SD、LD、AC、*VD、*LD、*AC ¹
OUT	BOOL BYTE WORD、INT D W O R D、 DINT	V、I、Q、M、SM ² 、S、T、C、L VB、IB、QB、MB、SMB ² 、SB、LB、AC、*VD、*LD、*AC ¹ VW、T、C、IW、QW、MW、SMW ² 、SW、LW、AC、AQW、*VD、*LD、*AC ¹ VD、ID、QD、MD、SMD ² 、SD、LD、AC、*VD、*LD、*AC ¹

¹ 必须偏移1个或1个以上的单位

² 必须偏移30个或30个以上的单位



提示

STEP 7-Micro/WIN为每个子程序自动加入返回指令。

当有一个子程序被调用时，系统会保存当前的逻辑堆栈，置栈顶值为1，堆栈的其他值为零，把控制权交给被调用的子程序。当子程序完成之后，恢复逻辑堆栈，把控制权交还给调用程序。

因为累加器可在主程序和子程序之间自由传递，所以在子程序调用时，累加器的值既不保存也不恢复。

当子程序在同一个周期内被多次调用时，不能使用上升沿、下降沿、定时器和计数器指令。

带参数调用子程序

子程序可以包含要传递的参数。参数在子程序的局部变量表中定义。参数必须有变量名（最多23个字符）、变量类型和数据类型。一个子程序最多可以传递16个参数。

局部变量表中的变量类型区定义变量是传入子程序（IN）、传入和传出子程序（IN_OUT）或者传出子程序（OUT）。表6-79中描述了一个子程序中的参数类型。要加入一个参数，把光标放到要加入的变量类型区（IN、IN_OUT、OUT）。点击鼠标右键可以得到一个菜单选择。选择插入选项，然后选择下一行选项。这样就出现了另一个所选类型的参数项。

表6-79 子程序的参数类型

参数	中断描述
IN	参数传入子程序。如果参数是直接寻址（如：VB10），指定位置的值被传递到子程序。如果参数是间接寻址（如：*AC1），指针指定位置的值被传入子程序；如果参数是常数（如：16 # 1234），或者一个地址（如：&VB100），常数或地址的值被传入子程序。
IN_OUT	指定参数位置的值被传到子程序，从子程序的结果值被返回到同样地址。常数（如：16 # 1234）和地址（如：&VB100）不允许作为输入/输出参数。
OUT	从子程序来的结果值被返回到指定参数位置。常数（如：16 # 1234）和地址（如：&VB100）不允许作为输出参数。由于输出参数并不保留子程序最后一次执行时分配给它的数值，所以必须在每次调用子程序时将数值分配给输出参数。注意：在电源上电时，SET和RESET指令只影响布尔量操作数的值。
TEMP	任何局部存储器都不能用来传递参数，只能在子程序内部暂时存储数据。

如图6-38中所示，局部变量表中的数据类型区定义了参数的大小和格式。参数类型如下所示：

- 布尔：该数据类型用于单独位的输入和输出。下例中的IN3是布尔输入。
- 字节、字和双字：这些数据类型分别指明一个1、2或者4个字节的无符号输入或输出参数。
- 整数、双整数：这些数据类型分别指明1、2或者4个字节的有符号输入或输出参数。

Name	Var Type	Data Type	Comment
EN	IN	BOOL	
L0.0	FirstPass	IN	First pass flag
LB1	Addr	IN	Address of slave device
Lw2	Data	IN	Data to write to slave
LB4	Status	IN_OUT	Status of write
LS.0	Done	OUT	Done flag
Lw6	Error	OUT	Error number (if any)

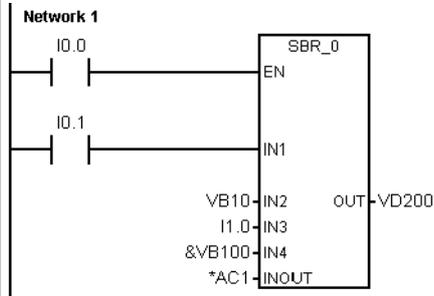
图6-38 本地变量表

- 实数：该数据类型指明一个（4字节）IEEE浮点值。
- STRING：该数据类型用作一个指向字符串的四字节指针。
- 能流：布尔能流仅允许对位输入操作。该变量声明告诉STEP 7-Micro/WIN 32此输入参数是位逻辑指令组合的能流结果。在局部变量表中布尔能流输入必须出现在其它类型的前面。只有输入参数可以这样使用。下例中的使能输入（EN）和IN1输入使用布尔逻辑。

示例: 子程序调用

以下有两个STL程序。第一个程序只能在STL编辑器中以STL的形式显示，因为用作能流输入的BOOL参数没有存储在L存储区中。

第二个程序能够在LAD和FBD编辑器中显示，因为使用了L存储器来存储用作能流输入的BOOL输入参数。



只能显示STL:

```
Network1
LD    I0.0
CALL  SBR0, I0.1, VB10, I1.0, &VB100, *AC1,
VD200
```

可以在LAD和FBD中正确显示:

```
Network1
LD    I0.0
=     L60.0
LD    I0.1
=     L63.7
LD    L60.0
CALL  SBR0, L63.7, VB10, I1.0, &VB100, *AC1,
VD200
```

地址参数（如IN4处的&VB100）以一个双字（无符号）的值传送到子程序。在带常数调用程序时必须指明常数类型。例如，把值为12345的无符号双字作为参数进行传递，常数参数必须用DW # 12345指明。如果参数中缺少了常数描述符，常数可能被当作不同的类型。

输入或输出参数上没有自动数据类型转换功能。例如，如果局部变量表明一个参数具有实型，而在调用时使用一个双字，子程序中的值就是双字。

当给子程序传递值时，它们放在子程序的局部存储器中。局部变量表的最左列是每个被传递参数的局部存储器地址。当子程序调用时，输入参数值被拷贝到子程序的局部存储器。当子程序完成时，从局部存储器区拷贝输出参数值到指定的输出参数地址。

数据单元的大小和类型用参数的代码表示。在子程序中局部存储器的参数值的分配如下所示:

- 按照子程序指令的调用顺序，参数值分别给局部存储器，起始地址是L0。
- 1到8连续位参数值分配一个字节，从Lx.0到Lx.7。
- 字节、字和双字值按照字节顺序分配在局部存储器中（LBx、LWx或LDx）。

在带参数调用子程序指令中，参数必须按照一定顺序排列，输入参数在最前面，其次是输入/输出参数，然后是输出参数。

如果用语句表编程，CALL指令的格式是:

```
CALL子程序号, 参数1, 参数2, ..., 参数
```

示例: 子程序和子程序返回指令程序举例		
M A I N	<p>Network 1</p>	<p>Network 1 //在首次扫描, 调用初始化子程序0。</p> <pre>LD SM0.1 CALL SBR_0</pre>
S B R 0	<p>Network 1</p> <p>Network 2</p>	<p>Network 1 //你可以使用条件返回指令在子程序 //结束之前返回。</p> <pre>LD M14.3 CRET</pre> <p>Network 2 //如果M14.3接通, 本段程序会被跳过。</p> <pre>LD SM0.0 MOVB 10, VBO</pre>

示例: 带字符串的子程序调用		
<p>该实例依据给定的输入, 复制不同的字符串文字到单独的地址。字符串的单独地址被保存。然后通过使用直接地址, 将字符串地址传递给子程序。子程序输入参数的数据类型是字符串。然后子程序移动字符串到不同位置。</p> <p>字符串文字也可以被传递给子程序。子程序内的字符串引用一直是相同的。</p>		
M A I N	<p>Network 1</p> <p>Network 2</p> <p>Network 3</p> <p>Network 4</p>	<p>Network 1 //</p> <pre>LD I0.0 SSCPY "string1", VB100 AENO MOVD &VB100, VDO</pre> <p>Network2 //</p> <pre>LD I0.1 SSCPY "string2", VB200 AENO MOVD &VB200, VDO</pre> <p>Network3 //</p> <pre>LD I0.2 CALL SBR_0, *VDO</pre>
S B R 0	<p>Network 1</p>	<p>Network 1 //</p> <pre>LD SM0.0 SSCPY *LD0, VB300</pre>

7

通过网络进行通讯

S7-200可以满足您的通讯和网络需求，它不仅支持简单的网络，而且支持比较复杂的网络。S7-200提供了通讯手段，使您可以用它与那些使用自己的通讯协议的设备，例如：打印机和称重天平等进行通讯。

STEP 7-Micro/WIN使得您建立和配置网络简便快捷。

本章内容:

理解S7-200网络通讯的基本概念	210
为网络选择通讯协议	214
通讯接口的安装和删除	220
网络的建立	221
用自由口模式创建用户定义的协议	225
在网络中使用Modem和STEP 7-Micro/WIN	228
高级议题	233
配置RS-232/PPI多主站电缆实现远程操作	239

理解S7-200网络通讯的基本概念

为网络选择通讯接口

S7-200可以支持各种类型的通讯网络。在SET PG/PC Interface属性对话框中进行网络选择。一个选定的网络将被作为一个接口来使用。能够访问这些通讯网络各类接口包括：

- 多主站PPI电缆
- CP通讯卡
- 以太网通讯卡

通过下列步骤，您可以为STEP 7-Micro/WIN选择通讯接口。参见图7-1。

1. 在通讯设置窗口中双击图标。
2. 为STEP 7-Micro/WIN选择接口参数。

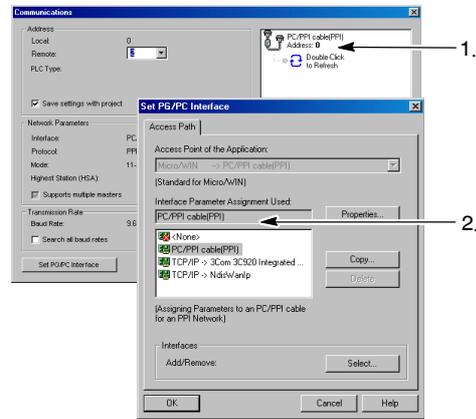


图7-1 STEP 7-Micro/WIN通讯接口

多主站PPI电缆

S7-200可以通过两种不同类型的PPI多主站电缆进行通讯。这些电缆允许通过RS-232或USB接口进行通讯。

如图7-2所示，选择PPI多主站电缆的方法很简单。只需执行以下步骤即可：

1. 在Set PG/PC Interface属性页中，点击属性按钮。
2. 在属性页中，点击本地连接标签。
3. 选中USB或所需的COM端口。

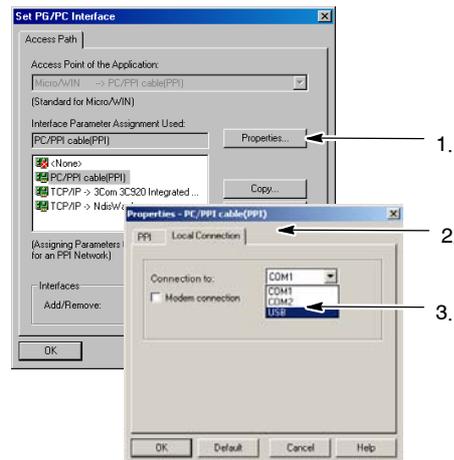


图7-2 PPI多主站电缆选择



提示

请注意，一次只能使用一个USB接口。



提示

本手册的实例使用的是RS-232/PPI多主站电缆。原先的PC/PPI电缆已被RS-232/PPI多主站电缆替代。另外，USB/PPI电缆也适用于这些实例。关于这些电缆的订货号，可参见附录C。

在PROFIBUS网络上使用主站和从站器件。

S7-200支持主-从网络，并能在PROFIBUS网络中充当主站或从站，而STEP 7-Micro/WIN只能作为主站。

主站

网络上的主站器件可以向网络上的其它器件发出要求。主站也可以对网络上其它主站的要求作出响应。典型的主站器件包括：STEP 7-Micro/WIN、TD200等HMI产品和S7-300或S7-400 PLC。在向其它S7-200发出请求以获取信息（点对点通讯）时，S7-200是作为主站的。



提示

如果网络上有其它主站，TP070将无法工作。

从站

配置为从站的器件只能对其它主站的要求作出响应，自己不能发出要求。对于多数情况，S7-200被配置为从站。作为从站，S7-200响应主站的要求。作为从站时，S7-200将负责响应来自某网络主站器件（如操作员面板或STEP 7-Micro/WIN）的请求。

设置波特率和站地址

数据通过网络传输的速度是波特率，其单位通常为Kbaud或者Mbaud。波特率用于量度在给定时间内传输数据的多少。比如，19.2k的波特率即表示传输速率为每秒19200比特。

在同一个网络中通讯的器件必须被配置成相同的波特率。因此，网络的最高波特率取决于连接在该网络上的波特率最低的设备。

表7-1中列出了S7-200支持的波特率。

在网络中要为每个设备指定唯一的站地址。唯一的站地址可以确保数据发送到正确的设备或者来自正确的设备。S7-200支持的网络地址为从0到126。如果某S7-200带有两个端口，那么每个端口都会有一个网络地址。表7-2列出了S7-200设备的缺省（工厂）设置。

表7-1 S7-200支持的波特率

网络	波特率
标准网络	9.6K到187.5K
使用EM277	9.6K到12M
自由口模式	1200到115.2K

表7-2 S7-200设备的缺省站地址

S7-200设备	缺省地址
STEP 7-Micro/WIN	0
HMI (TD200、TP或OP)	1
S7-200 CPU	2

为STEP 7-Micro/WIN设置波特率和站地址

您必须为STEP 7-Micro/WIN配置波特率和站地址。其波特率必须与网络上其它设备的波特率一致，而且站地址必须唯一。

通常，您不需要改变STEP 7-Micro/WIN的缺省站地址0。如果网络上还含有其它编程工具包，那么您可能需要改动STEP 7-Micro/WIN的站地址。

如图7-3所示，为STEP 7-Micro/WIN配置波特率和站地址非常简单。在操作栏中点击通讯图标，然后执行以下步骤：

1. 在通讯设置窗口中双击图标。
2. 在Set PG/PC Interface对话框中点击属性按钮。
3. 为STEP 7-Micro/WIN选择站地址。
4. 为STEP 7-Micro/WIN选择波特率。

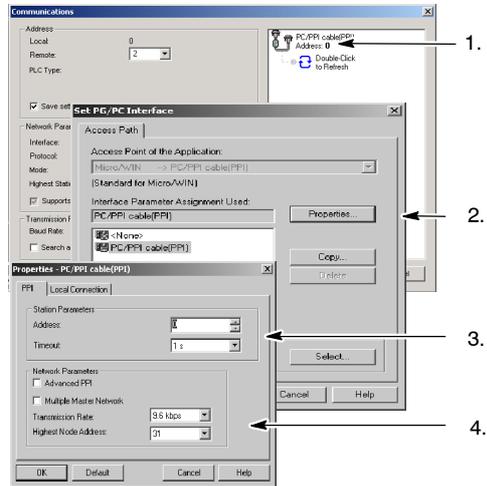


图7-3 配置STEP 7-Micro/WIN

为S7-200设置波特率和站地址

您也必须为S7-200配置波特率和站地址。S7-200的波特率和站地址存储在系统块中。在为S7-200设置了参数之后，您必须将系统块下载至S7-200中。

每一个S7-200通讯口的波特率缺省设置为9.6k，站地址的缺省设置为2。

如图7-4所示，使用STEP 7-Micro/WIN为S7-200设置波特率和站地址。您可以在操作栏中点击系统块图标或者在命令菜单中选择**View > Component > System Block**，然后执行以下步骤：

1. 为S7-200选择站地址。
2. 为S7-200选择波特率。
3. 下载系统块到S7-200。

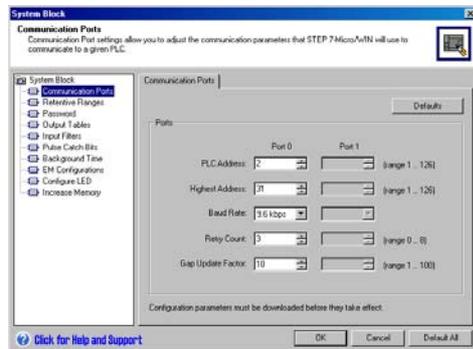


图7-4 配置S7-200 CPU



提示

可以选择各种波特率。在下载系统块期间，STEP 7-Micro/WIN将会验证所选的波特率。如果选定的波特率可能会妨碍STEP 7-Micro/WIN与其它S7-200进行通讯，那么它将被不下载。

设置远端地址

在将新设置下载到S7-200之前，您必须为STEP 7-Micro/WIN（本地）的通讯（COM）口和S7-200（远端）的地址作配置，使它与远端的S7-200的当前设置相匹配。如图7-5所示。

在下载了新设置后，您可能需要重新配置PG/PC接口波特率设置（如果新设置与远端S7-200的设置不同）。关于波特率的配置，可参考图7-3。

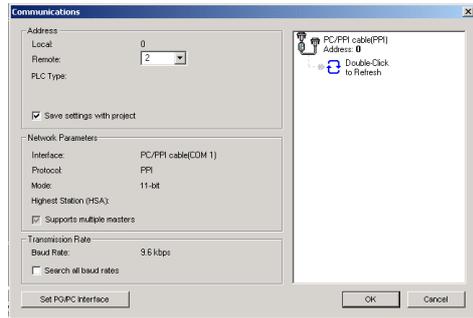


图7-5 配置STEP 7-Micro/WIN

在网络上寻找S7-200 CPU

您可以寻找并且识别连接在网络上的S7-200。在搜索S7-200时，您也可以寻找特定波特率上的网络或所有波特率上的网络。

只有在使用PPI多主站电缆时，才能实现全波特率搜索。若在使用CP卡进行通讯的情况下，该功能将无法实现。搜寻从当前选择的波特率开始。

1. 打开通讯对话框并双击刷新图标开始搜寻。
2. 要使用所有波特率搜寻，选中在所有波特率下搜寻复选框。

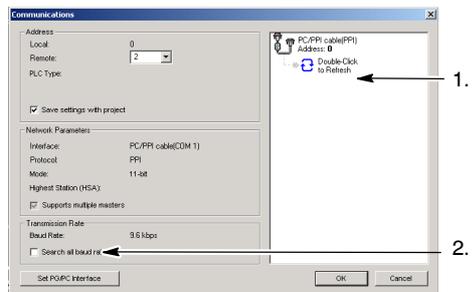


图7-6 搜索网络上的CPU

为网络选择通讯协议

下面是S7-200 CPU所支持的协议的总览。

- 点对点接口 (PPI)
- 多点接口 (MPI)
- PROFIBUS

在开放系统互联 (OSI) 七层模式通讯结构的基础上, 这些通讯协议在一个令牌环网络上实现。令牌环网络符合欧洲标准EN 50170中定义的PROFIBUS标准。这些协议是非同步的字符协议, 有1位起始位、8位数据位、偶校验位和1位停止位。通讯结构依赖于特定的起始字符和停止字符、源和目的地站地址, 报文长度和数据校验和。在波特率一致的情况下, 这些协议可以同时在一个网络上运行, 并且互不干扰。

如果带有扩展模块CP243-1和CP243-1 IT, 那么S7-200也能运行在以太网上。

PPI协议

PPI是一种主-从协议: 主站器件发送要求到从站器件, 从站器件响应, 参见图7-7。从站器件不发信息, 只是等待主站的要求并对要求作出响应。

主站靠一个PPI协议管理的共享连接来与从站通讯。PPI并不限制与任意一个从站通讯的主站数量, 但是在一个网络中, 主站的个数不能超过32。

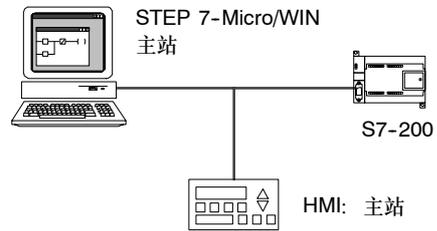


图7-7 PPI网络

如果在用户程序中启用PPI主站模式, S7-200 CPU在运行模式下可以作主站。(参见附录B中SMB 30的描述) 在启用PPI主站模式之后, 可以使用网络读写指令来读写另外一个S7-200。当S7-200作PPI主站时, 它仍然可以作为从站响应其它主站的请求。

PPI高级允许网络设备建立一个设备与设备之间的逻辑连接。对于PPI高级, 每个设备的连接个数是有限制的。S7-200支持的连接个数如表7-3所示。

所有的S7-200 CPU都支持PPI和PPI高级协议, 而EM277模块仅仅支持PPI高级协议。

表7-3 S7-200 CPU和EM277模块的连接个数

模块	波特率	连接数
S7-200 CPU	通讯口0	9.6k、19.2k或187.5k
	通讯口1	9.6k、19.2k或187.5k
EM277	9.6k到12M	6 (每个模块)

MPI协议

MPI允许主-主通讯和主-从通讯，参见图7-8。与一个S7-200 CPU通讯，STEP 7-Micro/WIN建立主-从连接。MPI协议不能与作为主站的S7-200 CPU通讯。

网络设备通过任意两个设备之间的连接通讯（由MPI协议管理）。设备之间通讯连接的个数受S7-200 CPU或者EM277模块所支持的连接个数的限制。S7-200支持连接个数如表7-3所示。

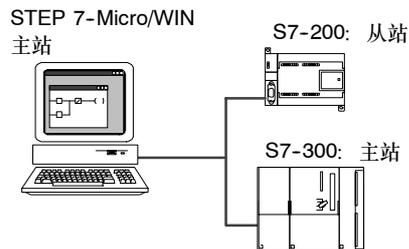


图7-8 MPI网络

对于MPI协议，S7-300和S7-400 PLC可以用XGET和XPUT指令来读写S7-200的数据。要得到更多关于这些指令的信息，参见S7-300或者S7-400的编程手册。

PROFIBUS协议

PROFIBUS协议通常用于实现与分布式I/O（远程I/O）的高速通讯。可以使用不同厂家的PROFIBUS设备。这些设备包括简单的输入或输出模块、电机控制器和PLC。

PROFIBUS网络通常有一个主站和若干个I/O从站，参见图7-9。主站器件通过配置可以知道I/O从站的类型和站号。主站初始化网络使网络上的从站器件与配置相匹配。主站不断地读写从站的数据。

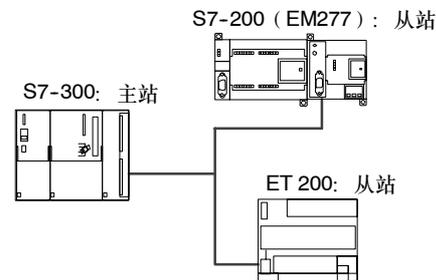


图7-9 PROFIBUS网络

当一个DP主站成功配置了一个DP从站之后，它就拥有了这个从站器件。如果在网上有第二个主站器件，那么它对第一个主站的从站的访问将会受到限制。

TCP/IP协议

通过以太网扩展模块（CP243-1）或互联网扩展模块（CP243-1 IT），S7-200将能支持TCP/IP以太网通讯。表7-4列出了这些模块所支持的波特率和连接数。

表7-4 以太网模块（CP243-1）和互联网模块（CP243-1 IT）的连接数

模块	波特率	连接数
以太网（CP243-1）模块	10到100兆	8个普通连接
互联网（CP243-1 IT）模块		1个STEP 7-Micro/WIN连接

若需更多信息，可参考SIMATIC NET CP243-1 工业以太网通讯处理器手册或 SIMATICNET CP243-1 IT 工业以太网及信息技术通讯处理器手册。

仅仅使用S7-200设备的网络配置实例

单主站PPI网络

对于简单的单主站网络来说，编程站可以通过PPI多主站电缆或编程站上的通讯处理器（CP）卡与S7-200 CPU进行通讯。

在图7-10上面的网络实例中，编程站（STEP 7-Micro/WIN）是网络的主站。在图7-10下面的网络实例中，人机界面（HMI）设备（例如：TD200、TP或者OP）是网络的主站。

在两个网络中，S7-200 CPU都是从站响应来自主站的要求。

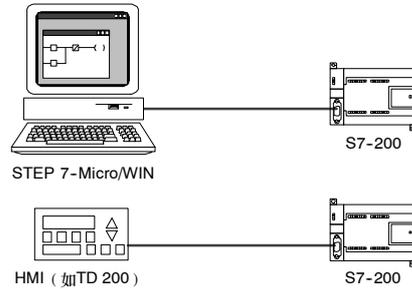


图7-10 单主站PPI网络

对于单主站PPI网络，需要配置STEP 7-Micro/WIN使用PPI协议。如果可能的话，请不要选择多主站网络，也不要选中PPI高级选框。

多主站PPI网络

图7-11中给出了有一个从站的多主站网络示例。编程站（STEP 7-Micro/WIN）可以选用CP卡或PPI多主站电缆。STEP 7-Micro/WIN和HMI共享网络。

STEP 7-Micro/WIN和HMI设备都是网络的主站，它们必须有不同的网络地址。如果使用PPI多主站电缆，那么该电缆将作为主站，并且使用STEP 7-Micro/WIN提供给它的网络地址。S7-200 CPU将作为从站。

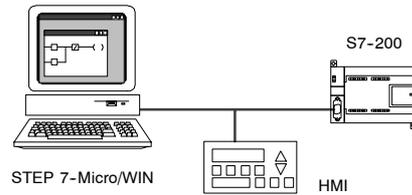


图7-11 只带一个从站的多主站

图7-12中给出了多个主站和多个从站进行通讯的PPI网络实例。在例子中，STEP 7-Micro/WIN和HMI可以对任意S7-200 CPU从站读写数据。STEP 7-Micro/WIN和HMI共享网络。

所有设备（主站和从站）有不同的网络地址。如果使用PPI多主站电缆，那么该电缆将作为主站，并且使用STEP 7-Micro/WIN提供给它的网络地址。S7-200 CPU将作为从站。

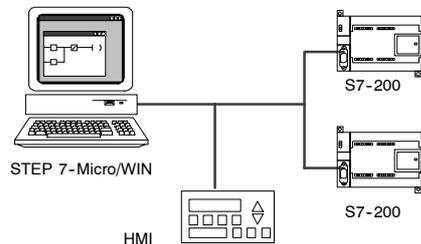


图7-12 多个主站和多个从站

对于带多个主站和一个或多个从站的网络，需配置STEP 7-Micro/WIN使用PPI协议，如果可能，还应使能多主网络并选中PPI高级选框。如果您使用的电缆是PPI多主站电缆，那么多主网络和PPI高级选框便可以忽略。

复杂的PPI网络

图7-13给出了一个带点对点通讯的多主网络。

STEP 7-Micro/WIN和HMI通过网络读写S7-200 CPU，同时S7-200 CPU之间使用网络读写指令相互读写数据（点对点通讯）。

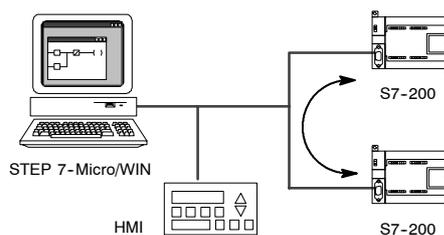


图7-13 点对点通讯

图7-14中给出了另外一个带点对点通讯的多主网络的复杂PPI网络实例。在本例中，每个HMI监控一个S7-200 CPU。

S7-200 CPU使用NETR和NETW指令相互读写数据（点对点通讯）。

对于复杂的PPI网络，配置STEP 7-Micro/WIN使用PPI协议时，最好使能多主站，并选中PPI高级选框。如果您使用的电缆是PPI多主站电缆，那么多主网络和PPI高级选框便可以忽略。

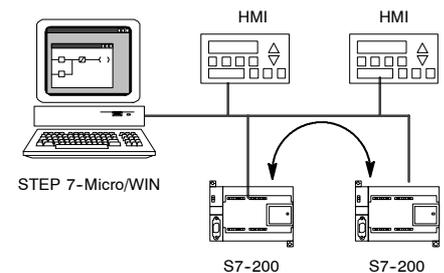


图7-14 HMI设备及点对点通讯

使用S7-200、S7-300和S7-400设备的网络配置实例

网络波特率可以达到187.5k

在图7-15所示的网络实例中，S7-200用XGET和XPUT指令与S7-300CPU通讯。如果S7-200处于主站模式，那么S7-200将无法与之通讯。

若要与S7 CPU通讯，则最好在配置STEP 7-Micro/WIN使用PPI协议时，使能多主站，并选中PPI高级选框。如果您使用的电缆是PPI多主站电缆，那么多主网络和PPI高级选框便可以忽略。

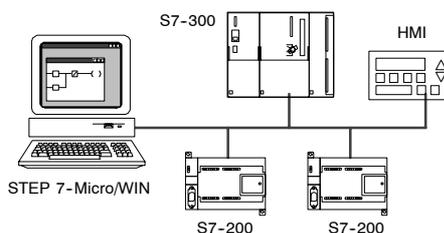


图7-15 波特率可以达到187.5k

网络波特率高于187.5k

对于波特率高于187.5k的情况，S7-200 CPU必须使用EM277模块连接网络，参见图7-16。STEP 7-Micro/WIN必须通过通讯处理器（CP）卡与网络连接。

在这个配置中，S7-300可以用XGET和XPUT指令与S7-200通讯，并且HMI可以监控S7-200或者S7-300。

EM277只能作从站。

通过S7-200附带的EM277，STEP 7-Micro/WIN将能够编程或监视S7-200。若要以高于187.5k的波特率与EM277通讯，则需配置STEP 7-Micro/WIN通过CP卡使用MPI协议。因为PPI多主站电缆的最高波特率为187.5k。

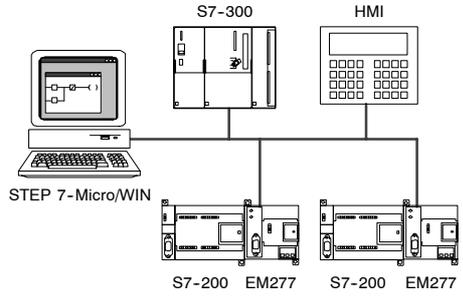


图7-16 波特率高于187.5k

PROFIBUS网络配置实例

S7-315-2DP作PROFIBUS主站，EM277作PROFIBUS从站的网络

图7-17中给出了用S7-315-2DP作PROFIBUS主站的PROFIBUS网络示例。EM 277模块是PROFIBUS从站。

S7-315-2DP可以发送数据到EM277，也可以从EM277读取数据。通讯的数据量为1到128个字节。S7-315-2DP读写S7-200的V存储器。

网络支持9.6K到12M的波特率。

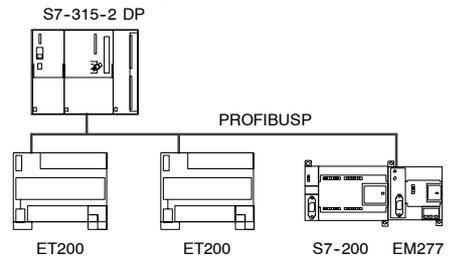


图7-17 S7-315-2 DP网络

有STEP 7-Micro/WIN和HMI的网络

图7-18中给出了用S7-315-2DP作PROFIBUS主站，EM277作PROFIBUS从站的网络示例。在这个配置中，HMI通过EM277监控S7-200。STEP 7-Micro/WIN通过EM277对S7-200进行编程。

网络支持9.6K到12M的波特率。当波特率高于19.2K时，STEP 7-Micro/WIN要用CP卡。

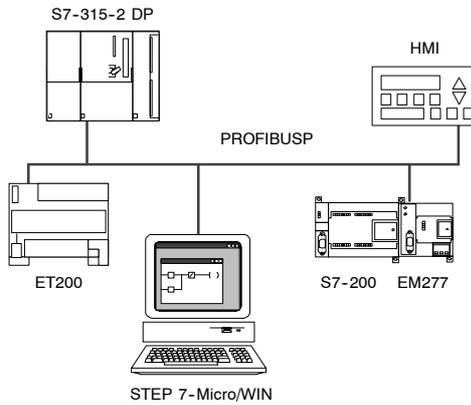


图7-18 PROFIBUS网络

若要使用CP卡，需配置STEP 7-Micro/WIN使用PROFIBUS协议。如果网络上只有DP设备，那么可以选择DP协议或标准协议。如果网络上有非DP设备（比如TD200），则可为所有的主站器件选择通用（DP/FMS）协议。网络上所有的主站都必须使用同样的PROFIBUS网络协议（DP、标准或通用）。

只有在所有主站器件都使用通用（DP/FMS）协议，并且网络的波特率小于187.5k时，PPI多主站电缆才能发挥其功能。

以太网和/或互联网设备的网络配置示例

在图7-19所示的配置中，STEP 7-Micro/WIN通过以太网连接与两个S7-200通讯，而这两个S7-200分别带有以太网（CP 243-1）模块和互联网（CP 243-1 IT）模块。S7-200 CPU可以通过以太网连接交换数据。安装了STEP 7-Micro/WIN之后，PC上会有一个标准浏览器，我们可以用它来访问互联网（CP 243-1 IT）模块的主页。

若要使用以太网连接，需配置STEP 7-Micro/WIN使用TCP/IP协议。

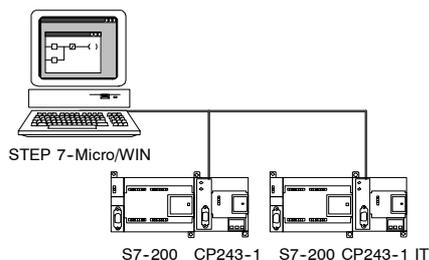


图7-19 10/100兆以太网



提示

在Set PG/PC Interface对话框中，至少有两个TCP/IP选项。S7-200没有TCP/IP -> NdisWanIp选项。

- Set PG/PC Interface对话框中的选项数取决于PC上的以太网接口类型。选择将您的计算机连接到以太网的接口类型，在这个以太网中连有CP243-1或CP243-1 IT模块。
- 在通讯对话框中，您必须为每个希望用它们进行通讯的以太网/互联网模块指定远端IP地址（一个或多个）。

通讯接口的安装和删除

在Set PG/PC Interface对话框中，您可以使用安装/删除接口对话框来安装或者删除计算机上的通讯接口。

1. 在Set PG/PC Interface对话框中，点击访问安装/删除接口对话框。
选择框中列出了可以使用的接口，安装框中显示计算机上已经安装了的接口。
2. 要添加一个接口：选择计算机上已经安装了的通讯硬件并点击安装。当关闭安装/删除接口对话框后，Set PG/PC Interface对话框中会在Interface Parameter Assignment Used框中显示接口。
3. 要删除一个接口：选择要删除的接口并点击删除。当关闭安装/删除接口对话框后，Set PG/PC Interface对话框中会在Interface Parameter Assignment Used框中删除该接口。

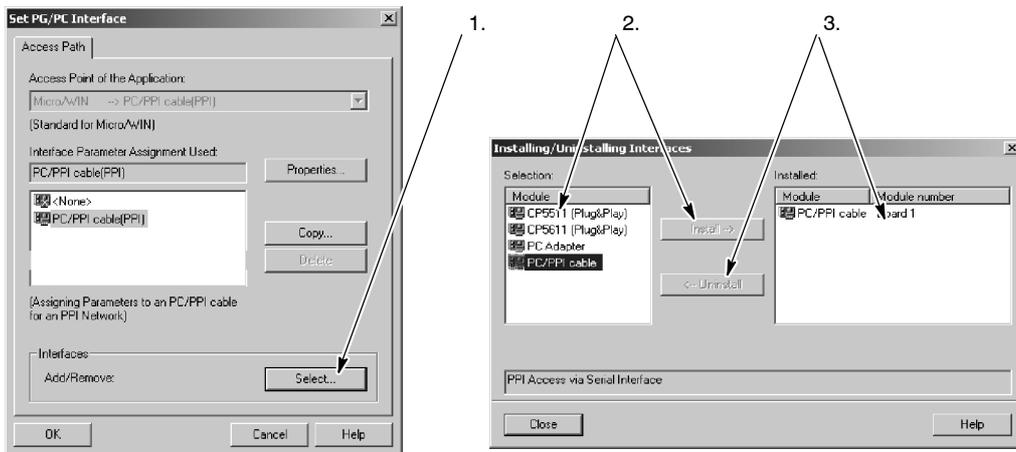


图7-20 Set PG/PC Interface和安装/删除接口对话框

在PPI多主模式下改变计算机的端口设置

如果在PPI模式下使用USB/PPI多主站电缆或RS-232/PPI多主站电缆，那么就无需调整计算机端口设置了，并且可以通过Windows NT操作系统在多主网络中进行操作。

如果您需要在某个支持PPI多主站配置（Windows NT不支持PPI多主站）的操作系统下，为了建立S7-200和STEP 7-Micro/WIN之间的通讯，而在PPI/自由口模式下使用RS-232/PPI多主站电缆，那么，您可能要调整计算机设置：

1. 在桌面上用右键单击“我的电脑”图标并在命令菜单中选择属性。
2. 选择设备管理标签。对于Windows 2000，选择第一个硬件标签然后按设备管理按钮。
3. 双击端口（COM和LPT）。
4. 选择当前使用的通讯端口（例如：COM1）。
5. 在端口设置页，点击高级按钮。
6. 将接收缓冲区和发送缓冲区调整到最低值（1）。
7. 点击OK使改变生效，关闭所有窗口并且重新启动计算机，使新的设置激活。

网络的建立

基本原则

导线必须安装合适的浪涌抑制器，这样可以避免雷击浪涌。

应避免将低压信号线和通讯电缆与交流导线和高能量、快速开关的直流导线布置在同一线槽中。要成对使用导线，用中性线或公共线与能量线或信号线配对。

S7-200 CPU的端口是不隔离的。如果想使网络隔离，应考虑使用RS-485中继器或者EM277。

小心

具有不同参考电位的互联设备有可能导致不希望的电流流过连接电缆。

这种不希望的电流有可能导致通讯错误或者设备损坏。

要确保用通讯电缆连接在一起的所有设备具有相同的参考电位，或者彼此隔离，来避免产生这种不希望的电流。关于接地、选择电路参考点和使用隔离电路的相关信息请参见第3章。

为网络确定通讯距离、通讯速率和电缆类型

如表7-5所示，网段的最大长度取决于两个因素：隔离（使用RS-485中继器）和波特率。

当您连接具有不同地电位的设备时需要隔离。当接地点之间的距离很远时，有可能具有不同的地电位。即使距离较近，大型机械的负载电流也能导致地电位不同。

表7-5 网络电缆的最大长度

波特率	非隔离CPU端口 ¹	有中继器的CPU端口或者EM277
9.6K到187.5K	50m	1,000m
500k	不支持	400m
1M到1.5M	不支持	200m
3M到12M	不支持	100m

¹ 如果不使用隔离端口或者中继器，允许的最长距离为50m。测量该距离时，从网段的第一个节点开始，到网段的最后一个节点。

在网络中使用中继器

RS-485中继器为网段提供偏压电阻和终端电阻。中继器有以下用途:

- ❑ **增加网络的长度:** 在网络中使用一个中继器可以使网络的通讯距离扩展50米。如图7-21所示,如果在已连接的两个中继器之间没有其它节点,那么网络的长度将能达到波特率允许的最大值。在一个串联网络中,您最多可以使用9个中继器,但是网络的总长度不能超过9600米。
- ❑ **为网络增加设备:** 在9600的波特率下,50米距离之内,一个网段最多可以连接32个设备。使用一个中继器允许您在网络上再增加32个设备。
- ❑ **实现不同网段的电气隔离:** 如果不同的网段具有不同的地电位,将它们隔离会提高网络的通讯质量。

一个中继器在网络中被算作网段的一个节点,尽管如此,它没有被指定站地址。



图7-21 带中继器的网络举例

选择网络电缆

S7-200网络使用RS-485标准,使用双绞线电缆。表7-6中列出了网络电缆的技术指标。每个网段中最多只能连接32个设备。

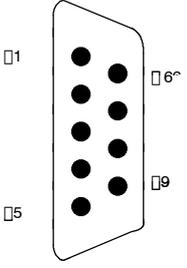
表7-6 网络电缆的通用技术指标

技术指标	描述
电缆类型	屏蔽双绞线
回路阻抗	$\leq 115 \Omega/\text{km}$
有效电容	30pF/m
标称阻抗	大约135 Ω ~ 160 Ω (频率=3 MHz ~ 20 MHz)
衰减	0.9dB/100m (频率=200KHz)
导线截面积	0.3mm ² ~0.5mm ²
电缆直径	8mm \pm 0.5mm

引脚分配

S7-200 CPU上的通讯端口是符合欧洲标准EN 50170中PROFIBUS标准的RS-485兼容9针D型连接器。表7-7列出了为通讯端口提供物理连接的设备，并描述了通讯端口的针脚分配。

表7-7 S7-200通讯端口的引脚分配

连接器	针	PROFIBUS名称	端口0/端口1
	1	屏蔽	机壳接地
	2	24V返回	逻辑地
	3	RS-485信号B	RS-485信号B
	4	发送申请	RTS (TTL)
	5	5V返回	逻辑地
	6	+ 5V	+ V, 100Ω串联电阻
	7	+ 24V	+ 24V
	8	RS-485信号A	RS-485信号A
	9	不用	10位协议选择 (输入)
连接器外壳	屏蔽	机壳接地	

网络电缆的偏置电阻和终端电阻

为了能够把多个设备很容易地连接到网络中，西门子公司提供两种网络连接器：一种标准网络连接器（引脚分配如表7-7所示）和一种带编程接口的连接器，后者允许您在不影响现有网络连接的情况下，再连接一个编程站或者一个HMI设备到网络中。带编程接口的连接器将S7-200的所有信号（包括电源引脚）传到编程接口。这种连接器对于那些从S7-200取电源的设备（例如TD200）尤为有用。

两种连接器都有两组螺钉连接端子，可以用来连接输入连接电缆和输出连接电缆。两种连接器也都有网络偏置和终端匹配的选择开关。典型的网络连接器偏置和终端如图7-22所示。

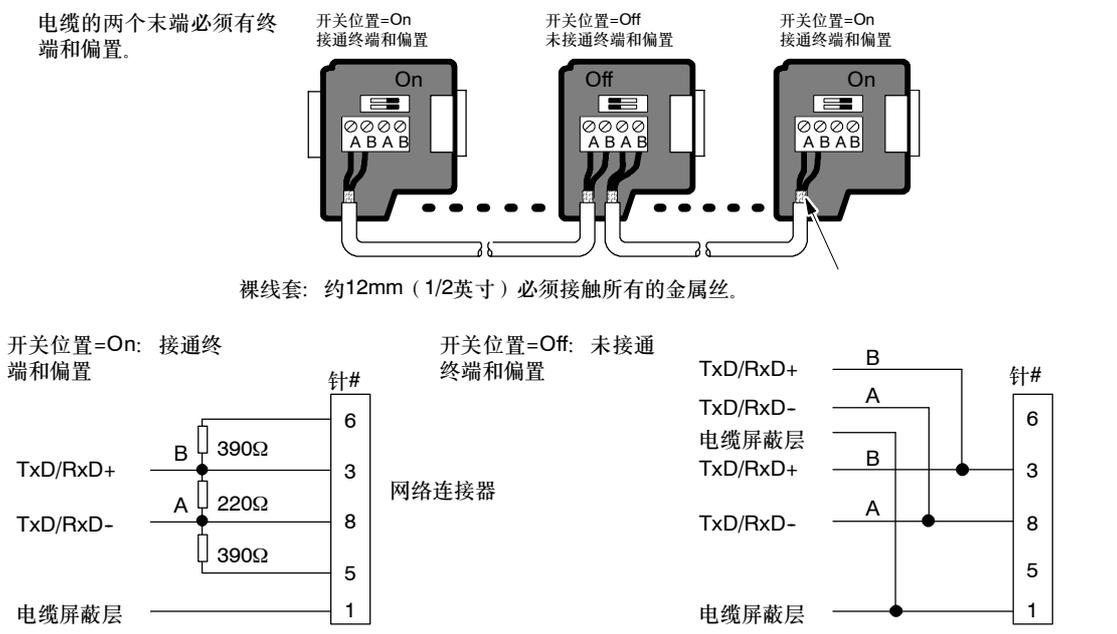


图7-22 网络电缆的偏置和终端

为您的网络选择PPI多主站电缆或CP卡

如表7-8所示，STEP 7-Micro/WIN支持多种CP卡以及RS-232/PPI多主站电缆和USB/PPI多主站电缆，并允许编程站（计算机或SIMATIC编程器）作为网络的主站。

当波特率小于等于187.5k时，PPI多主站电缆能以最简单和经济的方式将STEP 7-Micro/WIN连接到S7-200 CPU或S7-200网络。PPI多主站电缆有两种类型，它们都能将STEP 7-Micro/WIN连接到S7-200网络。

USB/PPI多主站电缆是一种即插即用设备，适用于支持USB1.1版的PC。当在187.5k的波特率下进行PPI通讯时，它能将PC和S7-200网络隔离。此时，无需设置任何开关，只需连上电缆，将PPI电缆设为接口并选用PPI协议，然后在PC接连标签下设置好USB端口即可。但在使用STEP 7-Micro/WIN时，不能同时将多根USB/PPI多主站电缆连接到PC上。

RS-232/PPI多主站电缆带有8个DIP开关：其中2个是用来配置电缆，使之可以用于STEP 7-Micro/WIN的。

- 如果需要将电缆连到PC上，则需选择PPI模式（开关5=1）和本地操作（开关6=0）。
- 如果需要将电缆连在调制解调器上，则需选用PPI模式（开关5=1）和远程操作（开关6=1）。

该电缆能将PC和S7-200网络隔离。要实现此功能，需将PPI电缆设为接口，并在PC接连标签下设置好RS-232端口。然后在PPI标签下，选定站地址和网络波特率。这时，协议将根据RS-232/PPI多主站电缆自动调整，因此您无需再做更多的设置。

USB/PPI多主站电缆和RS-232/PPI多主站电缆都带有LED，用来指示PC或网络是否在进行通讯。

- Tx LED用来指示电缆是否在将信息传送给PC。
- Rx LED用来指示电缆是否在接收PC传来的信息。
- 而PPI LED则用来指示电缆是否在网络上传输信息。由于多主站电缆是令牌持有方，因此，当STEP 7-Micro/WIN发起通讯时，PPI LED会保持点亮。而当与STEP 7-Micro/WIN的连接断开时，PPI LED会关闭。在等待加入网络时，PPI LED也会闪烁，其频率为1Hz。

CP卡为编程站管理多主网络提供了硬件，并且支持多种波特率下的不同协议。

每一块CP卡为网络连接提供了一个单独的RS-485接口。CP 5511 PCMCIA卡有一个提供9针D型接口的适配器。您可以将通讯电缆的一端接到CP卡的RS-485接口上，另一端接入网络。

如果您通过CP卡建立PPI通讯，那么，STEP 7-Micro/WIN将无法支持在同一块CP卡上同时运行两个应用。在通过CP卡将STEP 7-Micro/WIN连接到网络之前，您必须关掉另外一种应用。如果您使用的是MPI或PROFIBUS通讯，那么，将允许多个STEP 7-Micro/WIN应用在网上同时进行通讯。

小心

使用非隔离的RS-485到RS-232转换电缆会损坏计算机的RS-232端口。

Siemens RS-232/PPI和USB/PPI多主站电缆（订货号分别为6ES7 901-3CB30-0XA0或6ES7 901-3DB30-0XA0）提供S7-200 CPU的RS-485端口与连接到您的计算机上的RS-232或USB端口之间的电气隔离。如果您使用的不是Siemens多主站电缆，那么就必须另行为您的计算机的RS-232端口提供隔离。

表7-8 STEP 7-Micro/WIN支持的CP卡和协议

配置	波特率	协议
RS-232/PPI多主站或USB/PPI多主站电缆 ¹ 连接到编程站的一个端口	9.6k到187.5K	PPI
CP5511类型II, PCMCIA卡 (适用于笔记本电脑)	9.6k到12M	PPI、MPI和PROFIBUS
CP 5512 类型II, PCMCIA卡 (适用于笔记本电脑)	9.6k到12M	PPI、MPI和PROFIBUS
CP5611 (版本3以上) PCI卡	9.6k到12M	PPI、MPI和PROFIBUS
CP1613、S7613 PCI卡	10M或100M	TCP/IP
CP1612,SoftNet7 PCI卡	10M或100M	TCP/IP
CP1512,SoftNet7 PCMCIA卡 (适用于笔记本电脑)	10M或100M	TCP/IP

¹ 多主站电缆提供RS-485端口 (在S7-200上) 和计算机端口之间的电气隔离。若使用无隔离的RS-485至RS-232转换器, 则可能导致计算机的RS-232端口损坏。

在网络中使用HMI设备

S7-200 CPU支持西门子的多种HMI设备, 同时也支持其它厂家的产品。有些HMI设备 (如TD 200或TP070) 不允许您选择此设备所使用的通讯协议, 而另一些 (如OP7和TP170) 则允许您选择它们所用的通讯协议。

如果HMI设备允许您选择通讯协议, 应考虑以下原则:

- 对于直接连接在S7-200 CPU通讯端口上的HMI设备, 如果网络上没有其它设备, 您既可以选择PPI协议, 又可以选择MPI协议。
- 对于连接在EM277模块上的HMI设备, 您可以选择MPI或PROFIBUS。
 - 如果网络中有S7-300或S7-400 PLC, 为HMI设备选择MPI协议。
 - 如果HMI设备连接在一个PROFIBUS网络中, 为HMI设备选择PROFIBUS协议与其它主站相兼容。
- 如果HMI设备所连接的S7-200 CPU已经被配置为主站, 为HMI设备选择PPI。最好使用PPI高级。MPI和PROFIBUS协议不支持S7-200 CPU作主站。

用自由口模式创建用户定义的协议

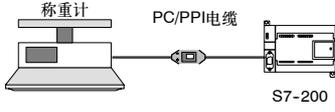
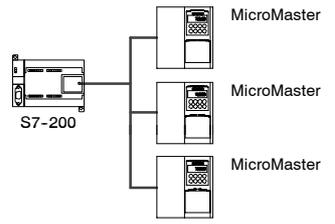
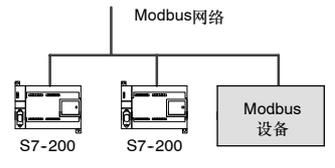
自由口模式允许应用程序控制S7-200 CPU的通讯端口。您可以在自由口模式下使用用户定义的通讯协议来实现与多种类型的智能设备的通讯。自由口模式支持ASCII和二进制协议。

要使能自由口模式，您需要使用特殊存储器字节SMB30（端口0）和SMB130（端口1）。应用程序中使用以下步骤控制通讯端口的操作：

- 发送指令（XMT）和发送中断：发送指令允许S7-200的通讯口上发送最多255个字节。发送中断通知程序发送完成。
- 接收字符中断：接收字符中断通知程序通讯口上接收到了一个字符。应用程序可以按字符执行操作。
- 接收指令（RCV）：接收指令从通讯口接收整条信息，当接收完成后产生中断通知应用程序。您需要在SM存储器中定义条件来控制接收指令开始和停止接收信息。接收指令可以根据特定的字符或时间间隔来启动和停止接收信息。接收指令可以实现多数通讯协议。

自由口模式只有在S7-200处于RUN模式时才能被激活。如果将S7-200设置为STOP模式，那么所有的自由口通讯都将中断，而且通讯端口会按照S7-200系统块中的配置转换到PPI协议。

表7-9 使用自由口模式

网络配置	描述	描述
<p>通过RS-232连接使用自由口</p> 	<p>举例：使用S7-200与带RS-232接口的电子天平通讯。</p> <ul style="list-style-type: none"> • RS-232/PPI多主站电缆连接在天平的RS-232端口与S7-200 CPU的RS-485端口之间。（将电缆设置为PPI/自由口模式，开关5=0。） • S7-200 CPU使用自由口与天平通讯。 • 波特率可以是1200~115.2k。 • 用户程序定义通讯协议。 	
<p>使用USS协议</p> 	<p>举例：使用S7-200与SIMODRIVE Micro Master驱动设备通讯。</p> <ul style="list-style-type: none"> • STEP 7-Micro/WIN提供USS库。 • S7-200 CPU是主站，驱动是从站。 <p> 关于USS程序的示例，可参阅资料光盘中的应用示例。见Tip28。</p> <p>应用示例</p>	
<p>创建用户程序来模仿另外一种网络上的从站器件</p> 	<p>举例：把S7-200 CPU连到Modbus网络。</p> <ul style="list-style-type: none"> • S7-200中的用户程序模仿一个Modbus从站。 • STEP 7-Micro/WIN提供Modbus库。 <p> 关于Modbus程序的实例，可参阅资料光盘中的应用示例。见Tip41。</p> <p>应用示例</p>	

使用RS-232/PPI多主站电缆和自由口模式连接RS-232设备

使用RS-232/PPI多主站电缆和自由口通讯功能，可以将S7-200 CPU连接到多种兼容RS-232标准的设备上。但电缆必须必须设为PPI/自由口模式（开关5=0）才能进行自由口通讯。开关6用于选择本地模式（DCE）（开关6=0）或远端模式（开关6=1）。

当数据从RS-232端口传输到RS-485端口时，RS-232/PPI多主站电缆将处于发送模式。当空闲或者数据从RS-485接口传输到RS-232接口时，电缆则处于接收模式。当电缆检测到RS-232传送线上的字符时，会马上由接收模式转入发送模式。

RS-232/PPI多主站电缆支持1200b至115.2k的波特率。通过RS-232/PPI多主站电缆护套上的DIP开关，可以配置合适的波特率。表7-10列出了波特率和开关位置的对应关系。

当RS-232传输线从空闲状态切换到接收模式时，需要一个时间周期，这个时间周期被定义为电缆的转换时间。如表7-10中所示，电缆的转换时间取决于所选择的波特率。

如果在应用自由口通讯的系统中使用RS-232/PPI多主站电缆，那么在以下情况下，必须考虑转换时间：

波特率	转换时间	设置 (1=上)
115200	0.15ms	110
57600	0.3ms	111
38400	0.5ms	000
19200	1.0ms	001
9600	2.0ms	010
4800	4.0ms	011
2400	7.0ms	100
1200	14.0ms	101

- S7-200响应RS-232设备发送的信息。

在S7-200接收到RS-232设备发送的要求信息之后，S7-200必须延时一段时间才能发送数据。延时时间应该大于或者等于电缆的转换时间。

- RS-232响应S7-200发送的信息。

在S7-200接收到RS-232设备的应答信息之后，S7-200必须延时一段时间才能发送下一条信息。延时时间应该大于或者等于电缆的转换时间。

在以上两种情况中，延时会使RS-232/PPI多主站电缆有足够的时间从发送模式切换到接收模式，从而使数据能从RS-485端口传送到RS-232端口。

在网络中使用Modem和STEP 7-Micro/WIN

STEP 7-Micro/WIN 3.2版或其后的版本使用标准的窗口电话和Modem选项来选择和配置电话线Modem。电话与Modem菜单在Windows的控制面板中。使用Windows设置菜单来设置Modem使您能够;

- 使用Windows支持的多数内置和外置Modem。
- 使用Windows支持的多数Modem的标准配置。
- 对于选择区域、国家和区域码; 选择脉冲或者语音拨号; 是否支持电话卡使用标准的Windows拨号规则。
- 当与EM241 Modem模块通讯时, 使用更高的波特率。

使用Windows控制面板可以显示, Modem属性对话框。这个对话框允许您配置本地Modem。您可以在Windows支持的Modem列表中选择您的Modem。如果您的Modem类型没有在Windows的Modem对话框中列出, 您可以选择一个最相似的型号或者与销售商联系。

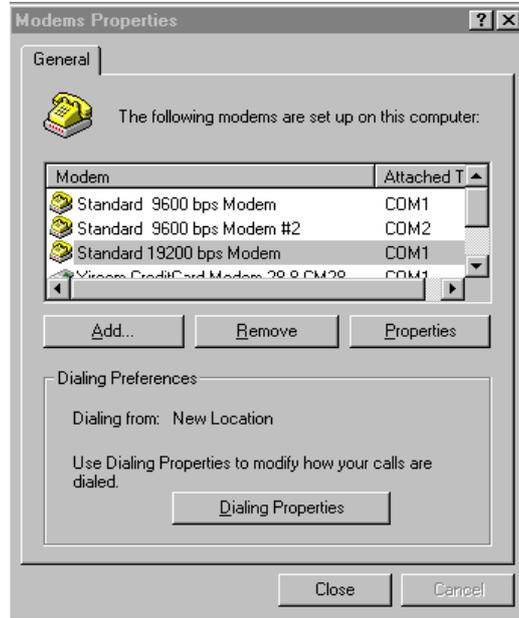


图7-23 配置本地Modem

STEP 7-Micro/WIN也支持电台或者移动Modem。这些Modem型号不会出现在Windows的Modem属性对话框中, 但是在STEP 7-Micro/WIN中配置之后可以使用。

配置一个Modem连接

一个连接有一个标识名与其物理属性相关联。对于一个电话Modem来说, 这些属性包括: Modem的类型、选择10位或11位协议和超时时间。对于移动Modem来说, 连接允许您设置引脚和其它参数。电台Modem属性包括波特率的选择、校验、数据流控制和其它参数。



连接向导

添加一个连接

使用连接向导可以添加一个新的连接，也可以删除或者编辑一个连接，如图7-24所示。

1. 在通讯设置窗口中双击图标。
2. 双击PC/PPI电缆打开PG/PC接口。选择PC/PPI电缆并点击属性按钮。在本地连接标签页中，选中Modem连接复选框。
3. 在通讯设置窗口中双击Modem连接图标。
4. 点击设置按钮，显示Modem连接设置对话框。
5. 点击添加按钮，启动添加Modem连接向导。
6. 按照向导配置连接。

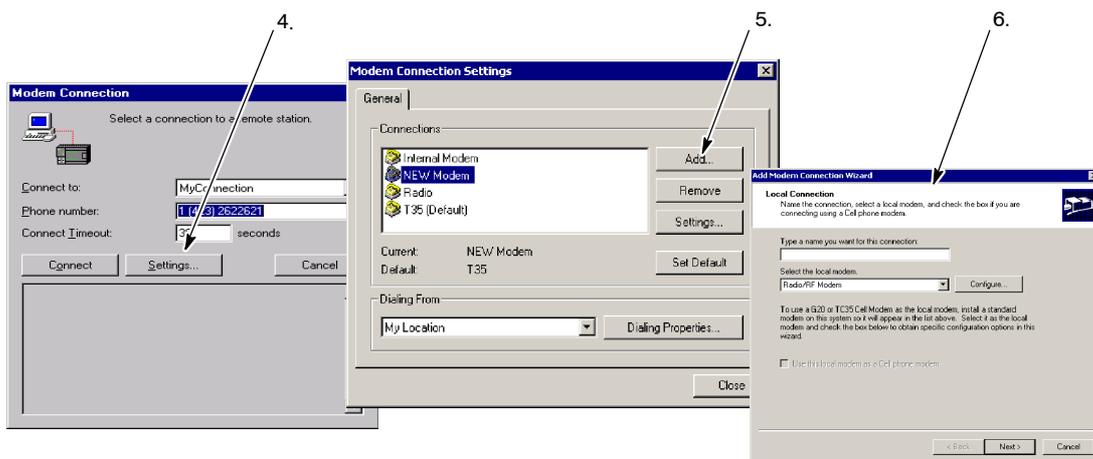


图7-24 添加一个Modem连接

通过Modem连接S7-200

在添加了一个Modem连接之后，您可以连接一个S7-200 CPU。

1. 打开通讯对话框并双击连接图标显示Modem连接对话框。
2. 在Modem连接对话框中，点击连接按钮对Modem拨号。

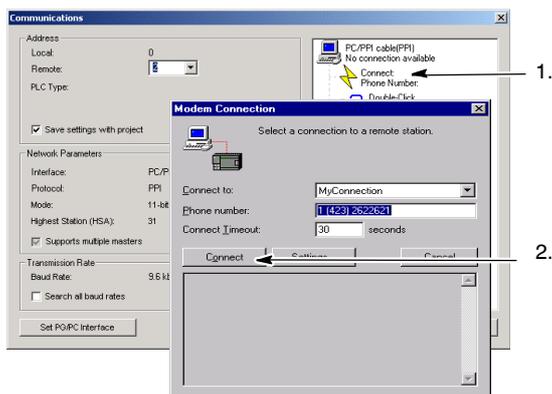


图7-25 连接S7-200



Modem扩展向导

配置远端Modem

远端Modem是指连接S7-200的Modem。如果远端Modem是EM241 Modem模块，则无需配置。如果您连接的是一个独立的Modem或移动Modem，那么就必须配置连接。

Modem扩展向导使配置连接变得容易。只有在进行了专门的Modem设置之后，我们才能通过RS-485半双工端口与S7-200 CPU建立正常通讯。只要简单地选择Modem类型并按照向导提示输入信息即可。要得到更多信息，参考在线帮助。

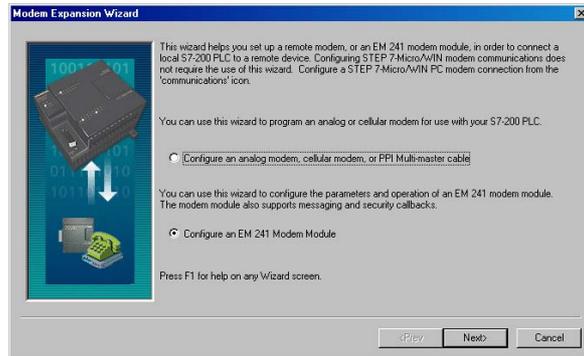


图7-26 Modem扩展向导

配置PPI多主站电缆连接远端Modem

RS-232 PPI多主站电缆能在电缆通电时向Modem发送AT命令串。请注意，如果必须改变Modem的缺省设置，则只需改变该配置就可以了。见图7-27。

在General命令栏中，您可以指定Modem命令。缺省设置只有一个：自动应答命令。

在Cell Phone Authorization区域中，可以指定移动电话授权命令和PIN号，比如：+CPIN=1234。

各个命令字符串将被分别发送给Modem。每个字符串的前面都会带有AT Modem申明命令。

如果点击了Program/Test按钮，那么这些命令就会在电缆中被初始化。

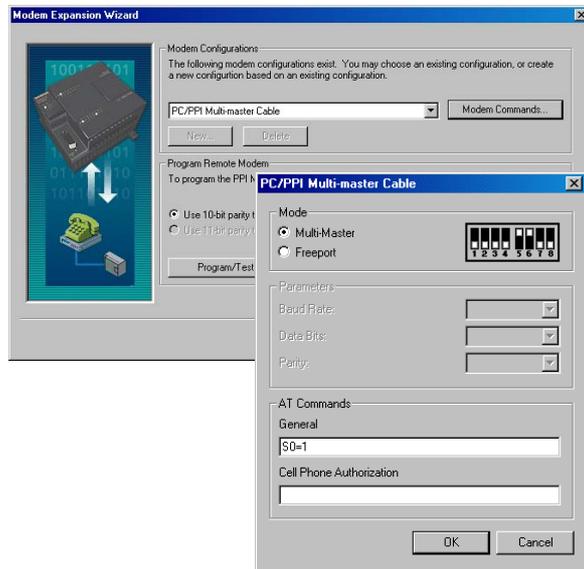


图7-27 Modem扩展向导 - 发送Modem命令

请注意，根据选定的参数，会出现一个供参考的开关设置位图。

在用STEP 7-Micro/WIN配置RS-232/PPI多主站电缆时，您必须将RS-485转换器连接在S7-200 CPU上。该转换器带有电缆工作必需的24V电源。因此，一定要确保S7-200 CPU的供电。

在STEP 7-Micro/WIN下完成RS-232/PPI多主站电缆配置之后，需要将电缆与PC断开，并将其连接到Modem上。对Modem和电缆周期性地通电。此时，您就可以在PPI多主网络中通过电缆进行远端操作了。



提示

如果要使用PPI多主站电缆，那么Modem必须处于工厂缺省设置。

配置PPI多主站电缆连接自由口

RS-232 PPI电缆也可以通过连接自由口的电缆发送Modem AT命令字符串。请注意，在改变Modem的缺省设置时，只需改变该配置就可以了。

但是，该电缆的配置必须与S7-200端口的波特率、奇偶校验和数据位数相匹配。这样，S7-200应用程序就能对这些参数的配置进行控制了。

波特率可在1.2k至115.2k之间选择。

数据位可以是7或8。

奇偶校验可以为偶、奇或无。

请注意，根据选定的参数，会出现一个推荐的开关设置位图。

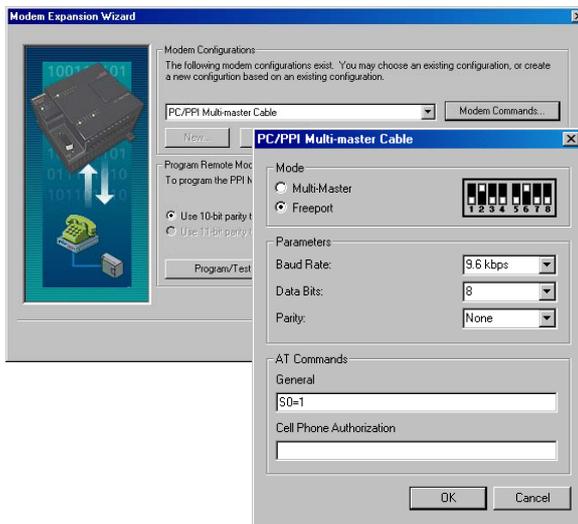


图7-28 Modem扩展向导 - 在自由口模式下发送Modem命令。

在用STEP 7-Micro/WIN配置RS-232/PPI多主站电缆时，您必须将RS-485转换器连接在S7-200 CPU上。该转换器带有电缆工作必需的24V电源。因此，一定要确保S7-200 CPU的供电。

在STEP 7-Micro/WIN下完成RS-232/PPI多主站电缆配置之后，需要将电缆与PC断开，并连接到Modem上。Modem和电缆将通电。此时，您就可以在PPI多主网络中通过电缆进行远端操作了。



提示

如果要使用PPI多主站电缆，那么Modem必须处于缺省的工厂设置。

用RS-232/PPI多主站电缆连接电话Modem

可以使用RS-232/PPI多主站电缆将Modem的RS-232通讯端口和S7-200 CPU连接起来。见图7-29。

- 开关1、2、和3用来设置波特率。
- 开关5用来选择PPI或PPI/自由口模式。
- 开关6选择本地（相当于数据通讯设备 - DCE）或远端（相当于数据终端设备 - DTE）模式。
- 开关7选择使用10位或11位PPI协议。

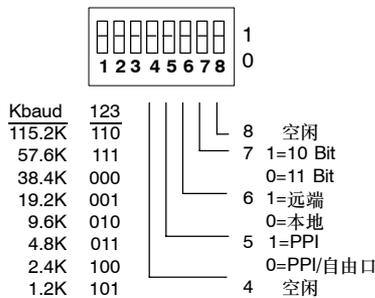


图7-29 设置RS-232/PPI多主站电缆

开关5用来切换PPI模式或PPI/自由口模式。如果要通过Modem来实现STEP 7-Micro/WIN和S7-200之间的通讯，那么选择PPI模式（开关5=1）。否则，需将电缆设置为PPI/自由口模式（开关5=0。）

RS-232/PPI多主站电缆的开关7用于选定PPI/自由口模式的位数是10位还是11位。仅在通过PPI/自由口模式的Modem连接S7-200和STEP 7-Micro/WIN的情况下，才需使用该开关。否则，只需将开关设置为11位模式，就能确保与其它设备的正常通讯了。

通过RS-232/PPI多主站电缆的开关6您可以将电缆的RS-232端口设置为本地（DCE）或远端（DTE）模式。

- ❑ 如果您在STEP 7-Micro/WIN下使用RS-232/PPI多主站电缆，或者已将RS-232/PPI多主站电缆连接在某计算机上，则需将RS-232/PPI多主站电缆设为本地（DCE）模式。
- ❑ 如果您将RS-232/PPI多主站电缆连接在Modem上（该Modem是DCE设备），则需将RS-232/PPI多主站电缆设为远端（DTE）模式。

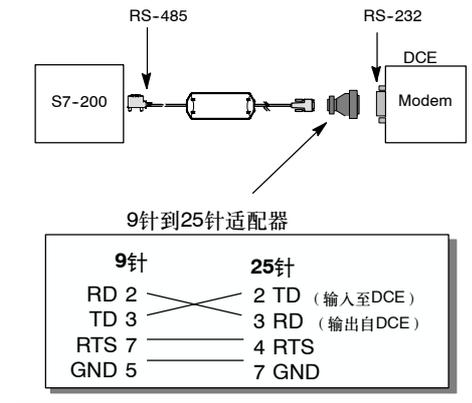


图7-30 适配器引脚分配

进行了该操作之后，就无需在RS-232/PPI多主站电缆和Modem之间安装空Modem适配器了。但如果Modem的连接口为25针，您还是需要一个9针转25针适配器。

图7-30给出了通用Modem适配器的引脚分配。

关于RS-232/PPI多主站电缆的更多信息，请参考附录A。本地（DCE）模式下RS-232/PPI多主站电缆的RS-485和RS-232端口的针脚号和功能已列在表A-66中。而远端（DTE）模式下RS-232/PPI多主站电缆的RS-485和RS-232端口的针脚号和功能则列在A-67中。RS-232/PPI多主站电缆仅在远端（DTE）模式下才支持RTS。

用RS-232/PPI多主站电缆连接无线Modem

RS-232/PPI多主站电缆可以将无线Modem的RS-232通讯端口连接到200 CPU上。然而，无线Modem的操作与电话线Modem的操作是不同的。

PPI模式

一般情况下，如果RS-232/PPI多主站电缆已设为PPI模式（开关5=1），且要与某个Modem连接，那么您就要将该电缆设为远端模式（开关6=1）。然而，如果电缆被设为远端模式，那么每次通电时，它都会向Modem发送字符串“AT”，并等待回应。通常，电话Modem会通过这一步骤来确定波特率，但无线Modem则不会接受AT命令。

因此，如果要将电缆连接至无线Modem，就必须将电缆设为本地模式（开关6=0），并在电缆的RS-232连接器和无线Modem的RS-232端口之间安装空Modem适配器。无论是9针至9针，还是9针至25针配置，都可以使用空Modem适配器。

配置无线Modem工作在9.6k、19.2k、38.4k、57.6k或115.2k速率下。当无线Modem发送出第一个字符时，RS-232/PPI多主站电缆会自动将波特率调整为前述波特率中的一个。

PPI/自由口

如果RS-232/PPI多主站电缆已设为PPI/自由口模式（开关5=0），且需连接至无线Modem，那么，就要将它设为远端模式（开关6=1）。配置完成后，电缆将不会发送AT命令并用该命令来设置Modem。

RS-232/PPI多主站电缆的开关1、2和3用于选择波特率。如图7-29所示，可根据PLC和无线Modem的波特率设置波特率。

高级议题

优化网络性能

影响网络性能的因素有以下几个（波特率和主站数的影响最大）：

- 波特率：如果网络能在所有设备都支持的最高波特率下运行，那么其效率是最大的。
- 网络中的主站个数：减少网络中的主站数目可以提高网络性能。网络中的每个主站都会增加网络的负载要求，主站少可以减轻网络负载。
- 主站和从站地址的选择：所有主站的地址应该不带地址间隙，顺序地进行设定。当主站间存在地址间隙时，主站连续检查间隙内的地址，确定是否有其它主站等待进入连接。这个检查需要时间，这样会增加网络的负载。如果主站之间没有地址间隙，就不需要进行检查，这样网络的负载最小。只要从站不位于主站之间，从站地址设置成任何值不会影响网络性能。位于主站之间的从站会造成主站之间的地址间隙，因而会增加网络的负载。
- 间隙刷新因子（GUF）：只有在S7-200 CPU作为PPI主站时才使用间隙刷新因子，它告诉S7-200检查其它主站地址间隙的时间间隔。使用STEP 7-Micro/WIN在CPU配置中为CPU通讯端口设置GUF。这个配置使S7-200周期性地检测地址间隔。如果GUF=1，S7-200每次占有令牌时都会检查地址间隔；如果GUF=2，S7-200每两次占有令牌时，才会检查一次地址间隔。如果主站之间有间隙，设置高的GUF可以降低网络负载。如果主站之间没有间隙，GUF不影响网络性能。由于不频繁检查地址，设置大的GUF会造成大的延时。缺省的GUF设置是10。
- 最高站地址（HSA）：只有在S7-200 CPU作为PPI主站时才使用最高站地址，它定义了一个主站寻找其它主站的最高地址。使用STEP 7-Micro/WIN在CPU配置中为CPU通讯端口设置HSA。设置HSA限制了最后一个主站（最高地址）必须检查的地址间隙。限制地址间隙的长度可以最小化寻找和连接另一个主站所需要的时间。最高站地址对于从站地址没有影响。主站仍然可以与地址大于HSA的从站通讯。总的规则是应该在所有的主站上设置相同的最高站地址。这个地址应该大于或等于系统中的最高主站地址。HSA的缺省值是31。

为网络计算令牌循环时间

在令牌传送网络中，只有拥有令牌的站有初始化通讯的权限。令牌循环时间可以体现出网络性能的高低（逻辑环中主站循环传送令牌的时间）。

图7-31为计算一个多主网络的令牌循环时间给出了一个网络实例。在这个例子中，TD200（3号站）与CPU222（2号站）通讯；TD200（5号站）与CPU222（4号站）通讯，以此类推。两个CPU224使用网络读写指令从其它S7-200采集数据：CPU224（6号站）向2号站、4号站和8号站发送数据；同时CPU224（8号站）向2号站、4号站和6号站发送数据。在该网络中，有6个主站（4个TD200和两个CPU224）和两个从站（两个CPU222）。



编程提示

关于令牌循环, 请参见资料光盘里的应用示例。见Tip42。

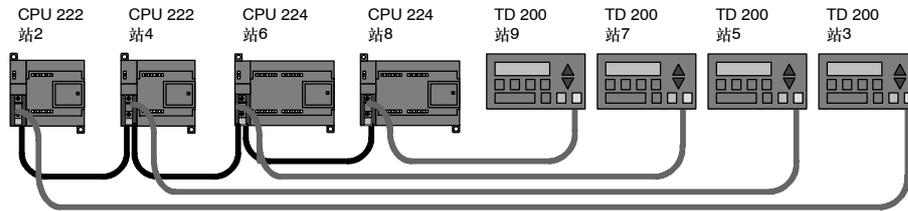


图7-31 令牌传送网络举例

主机要发送信息, 必须持有令牌。例如: 当站3持有令牌时, 它初始化到站2的请求, 然后把令牌传给站5。站5才能初始化到站4的请求信息, 然后把令牌传给站6。站6再初始化到站2、4或8的请求信息, 然后把令牌传给站7。这个初始化信息和传送令牌的过程会在逻辑环中持续进行, 从站3到站5, 又到站6、7、8、9, 最后又返回站3。主机要能够发出请求信息, 这个令牌必须在逻辑环中完整循环。对于一个6个站的逻辑环, 如果每个令牌持有者发送一个请求信息, 为一双字值(4个字节), 则令牌循环时间在9600波特下为900ms。如果信息访问的数据字节数增加, 或者站的数目增加, 那么令牌循环时间也会增加。

令牌循环时间是由各站占有令牌的时间决定的。对于多主网络, 令牌循环时间可以由各主站占有令牌时间相加得出。如果允许PPI主站模式(在网络中使用PPI协议), S7-200可以使用网络读写指令向其它S7-200发送信息。如果采用这些指令发送信息, 并且下述假设成立的话, 令牌循环时间可以由公式近似得出: 各站在每次占有令牌时发送一个请求; 该请求为数据连续的读或写请求; CPU的通讯缓冲区使用没有冲突; CPU的扫描时间不超过10ms。

令牌占有时间 (T_{hold}) = (128额外 + n 数据) 字符 \times 11位/字符 \times 1/波特率	
令牌循环时间 (T_{rot}) = 主站1的 T_{hold} + 主站2的 T_{hold} + ... + 主站 m 的 T_{hold}	
其中,	n 是数据的字符(字节)数 m 是主站数

图7-31中实例的令牌循环时间计算等式如下(1“位时间”等于1个信号的持续时间):

$$T(\text{令牌占用时间}) = (128 + 4) \text{ 字符} \times 11 \text{ 位/字符} \times 1 / 9600 \text{ 位} \\ = 151.25 \text{ ms/主机}$$

$$T(\text{令牌循环时间}) = 151.25 \text{ ms/主机} \leq 6 \text{ 主机} \\ = 907.5 \text{ ms} \text{ (一位时间等于一个信号的持续时间)}$$



提示

SIMATIC NET COM PROFIBUS软件提供对网络性能的分析器软件。

令牌循环时间比较

表7-11中给出了在不同通讯站个数、数据量以及波特率下的令牌循环时间比较。当您使用S7-200 CPU或其它主站器件进行网络读（ Network Read）或网络写（ Network Write）时，就需要计算时间。

表7-11 令牌循环时间（单位：秒）

波特率	传输字节数	主站的个数								
		2	3	4	5	6	7	8	9	10
9.6k	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
19.2k	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
187.5k	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

理解网络设备的链接

网络设备通过连接来实现通讯，连接是主站与从站之间的单独链接。如图7-32所示，连接的实现方式不同，通讯协议也会不同：

- PPI协议中所有网络设备共享一个连接。
- PPI高级、MPI和PROFIBUS协议中，一个主站和一个从站已经建立起来的连接不会被第二个主站干扰。

在使用PPI高级、MPI或PROFIBUS时，已经建立连接的主站与从站之间不能再加入第二个主站。S7-200 CPU和EM277总是为STEP 7-Micro/WIN和HMI设备各保留一个连接。其它主站器件不能使用这些被保留的连接。这就保证了当正在使用诸如PPI高级这样的协议时，在连接其它主站的同时，至少可以连接一个编程站或HMI设备到S7-200 CPU或EM277。

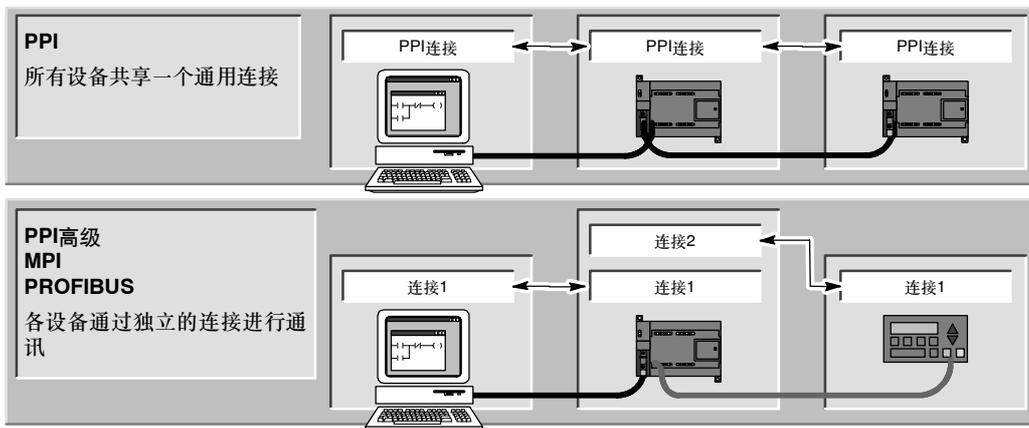


图7-32 管理通讯连接

如表7-12所示，S7-200 CPU或EM277提供一定数量的连接。S7-200 CPU的每一个端口（通讯口0和通讯口1）支持4个独立的连接。（这意味着S7-200 CPU最多允许8个连接）这去除了共享的PPI连接。一个EM277支持6个连接。

表7-12 S7-200 CPU和EM277模块的连接个数

连接点	波特率	连接数	STEP 7-Micro/WIN协议选择
S7-200 CPU 通讯口0	9.6k、19.2k或187.5k	4	PPI、PPI高级、MPI和PROFIBUS ¹
EM277	9.6K到12M	6（每个模块）	PPI高级、MPI和PROFIBUS

¹ 如果使用CP卡通过端口0或端口1连接STEP 7-Micro/WIN和S7-200 CPU，那么您只能在该S7-200设备作为从站时，才可以选用MPI或PROFIBUS协议中的一个。

配置复杂网络

对于S7-200来说，典型的复杂网络有多个S7-200主站，并且在PPI网络上用网络读写指令与其它设备通讯。复杂网络往往会出现一些特殊的问题，导致主站和从站之间的通讯阻塞。

如果网络运行在较低的波特率下（9.6k或19.2k），每个主站在传送令牌之前完成操作（读或写）。在187.5k波特率下，主站对从站提出要求然后传送令牌。这就对从站提出了较高要求。

图7-33中给出了一个有潜在的网络冲突的实例。在网络中，1号站、2号站和3号站是主站，使用网络读写指令与4号站通讯。网络读写指令使用PPI协议，因此所有S7-200共享4号站中的一个连接。

在本例中，1号站对4号站提出请求。对于高于19.2k的波特率，1号站将令牌传送给2号站。如果2号站也试图向4号站提出请求，其请求会被拒绝，因为4号站正在处理1号站的请求。在4号站完成对1号站的响应之前，所有请求都会被拒绝。只有在响应完成之后，4号站才能接受其它主站的请求。

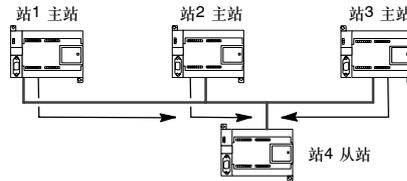


图7-33 通讯冲突

为了避免4号站通讯口上的通讯冲突，应考虑使4号站成为网络上唯一的主站，如图7-34所示。4号站可以向其它S7-200提出读写请求。

这样的配置不仅能够确保没有通讯冲突，而且减少了多主站导致网络负担，使网络更高效的运行。

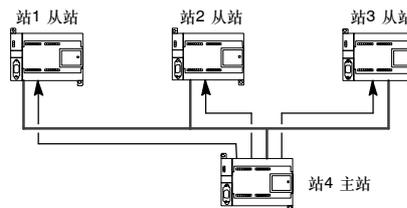


图7-34 避免冲突

对于某些应用来说，无法减少网络上的主站数量。当网上有多个主站时，您必须对令牌循环时间进行管理，并确保网络的令牌循环时间不超过目标值。（令牌循环时间是指一个主站传送令牌到再次得到令牌的时间间隔。）

表7-13 HSA和令牌循环时间目标值

HSA	9.6k	19.2k	187.5k
HSA=15	0.613s	0.307s	31ms
HSA=31	1.040s	0.520s	53ms
HSA=63	1.890s	0.950s	97ms
HSA=126	3.570s	1.790s	183ms

如果令牌回到主站的时间长于令牌循环时间目标值，该主站不能提出请求。只有当令牌循环时间小于目标值时，主站才能提出请求。

最高站地址（HSA）和S7-200的波特率设置决定了令牌循环时间。表7-13给出了令牌循环时间目标值列表。

对于较低的波特率，如：9.6k或者19.2k，主站会在传送令牌前等待应答。由于请求/应答的过程周期在扫描时间中占相当长的时间，因而很有可能每个网络上的主站在占有令牌时都作好了请求发送数据的准备。实际令牌循环时间增加并且有些主站可能不能发送任何请求。在某些情况下，某个主站只能偶尔处理其请求。

*例如：*考虑一个有10个主站的网络，其HSA值为15，只能在9.6k的波特率下传输1个字节。在该例中，每个主站总是会1个信息等着发送。如表7-13所示，令牌循环时间应该是0.613秒。然而，根据表7-11列出的网络性能数据，该网络的实际令牌循环时间是1.48秒。由于实际令牌循环时间大于目标令牌循环时间，因此，除非后来的一些令牌循环到其中的一些主站，否则它们将不能传送信息。

当实际令牌循环时间大于目标令牌循环时间时，有两种基本改进方法：

- 通过减少网络上的主站数减少实际令牌循环时间。其可行性要取决于您的应用。
- 您也可以增加网络上所有主站器件的HSA值，从而增加目标令牌循环时间。

增大HSA的值会导致其它问题，它会延长S7-200切换到主站模式进入网络的时间。如果您使用一个定时器来确保网络读写指令在指定的时间内完成，初始化主站模式和S7-200作为主站进入网络时间的延时，会导致指令超时。您可以用减小间隙刷新因子（GUF）的方法来使主站进入网络的时间最小化。

由于在187.5k波特率时，请求传送并驻留在从站中需要一定时间，因而在选择令牌循环时间目标值时应留有余量。在187.5k的波特率下，实际令牌循环时间应大约为目标值的一半。

要计算令牌循环时间，使用表7-11中完成网络读写指令所需要的时间数据。要计算HMI设备（例如TD200）的时间需求，使用传送16字节的数据。将网上每个设备的时间需求加在一起，就可以计算出令牌循环时间。将所有时间需求加在一起，意味着网上所有设备在同一个令牌循环里都有请求这种最坏的情况。这样计算出的时间是网络令牌循环时间的最大值。

例如: 考虑一个网络运行在9.6k波特率下, 有4个TD200和4个S7-200, 每个S7-200每秒钟向另一个S7-200写10个字节。用表7-11计算网络的传输时间为:

4个TD200传输16个字节数据 =	0.66s
4个S7-200传输10个字节数据 =	0.63s
0.63s总的令牌循环时间 =	<u>1.29s</u>

为了允许在一个令牌循环中有足够的时间处理所有请求, 将HSA设为63。(见表7-13)。选择的令牌循环目标值为1.89s, 大于最大令牌循环时间1.29s, 因而确保了每个设备在每一个令牌循环中都可以传输数据。

为了提高多主网络的可靠性, 您还必须考虑以下措施:

- 增大HMI设备的刷新时间。例如: 将TD200的刷新速率由“尽快”改为“每秒一次”。
- 减少网络读写指令的请求数量(减少处理请求的网络负担)。例如: 将两次读4个字节的网络读指令操作, 合并为一次读8个字节的网络读指令操作。两次读4个字节操作需要的时间会远远大于一次读8个字节操作。
- 改变S7-200主站的刷新时间, 使其不要试图使刷新时间小于令牌循环时间。

使用RS-232/PPI多主站电缆进行远端操作

超级终端作为配置工具

如果无法使用STEP 7-Micro/WIN来配置RS-232/PPI多主站电缆，将之用于远端操作，那么您也可以使用超级终端或其它通讯软件来配置它们。在为远端操作配置电缆时，RS-232/PPI多主站电缆的内置菜单会给您提供向导。

在用超级终端配置RS-232/PPI多主站电缆时，S7-200 CPU上必须装有RS-485连接器。该连接器将为电缆提供其运行所必须的24V电源。必须确保S7-200 CPU获得供电。

要在PC上调用超级终端，可点击开始 > 程序 > 附件 > 通讯 > 超级终端。

超级终端程序启动之后，会先显示连接说明画面。您必须为该连接设定一个名称（比如：Multi-master）。您也可以为新连接选择一个图标，或者使用默认的连接图标。参见图7-35。

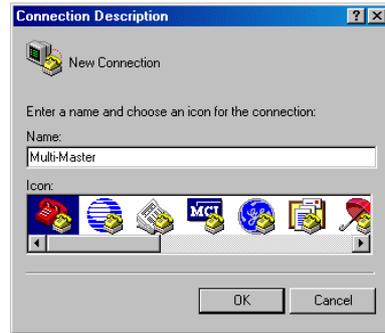


图7-35 超级终端连接说明

Connect To画面弹出。选择需要的通讯端口，点击OK。然后出现的是COMx Properties画面。接受缺省设置并点击OK。参见图7-36。

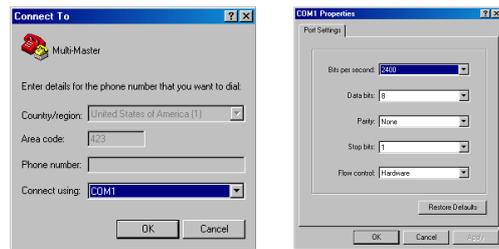


图7-36 超级终端的Connect To画面和COMx Properties画面

如图7-37所示，在点击了OK之后，光标将进入超级终端的编辑窗口。请注意，此时超级终端按钮处的状态栏将指出连接已建立，同时计数器开始计时，指出连接的持续时间。

在菜单中，选择Call > Disconnect。此时状态栏将指示连接已断开。

选择View > Font。选择Courier New，点击OK。

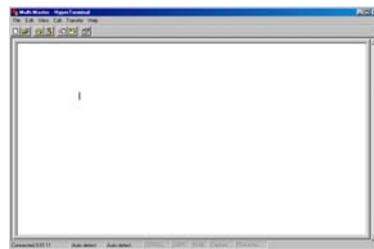


图7-37 多主站超级终端编辑窗口

选择**File > Properties**。在Connect To标签下，点击**Configure ...**按钮，通讯端口属性就会显示出来。参见图7-38。

在COMx Properties对话框中，通过下拉菜单选定波特率，其单位是Bit每秒。所选的波特率必须在9600至15200比特每秒之间（通常为9600）。然后在相应的下拉菜单中，选择8个数据位，无奇偶校验，一个停止位及无数据流控制。

点击OK，返回至Connect To标签下。

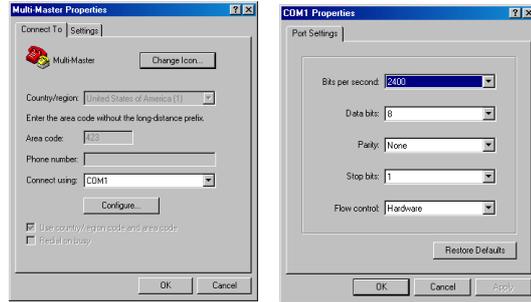


图7-38 多主站属性和COMx属性

选择Settings标签。在Emulation下拉菜单中，选择ANSI并点击OK。这时您将回到超级终端画面的编辑窗口中。状态栏将指示：

“Disconnected ANSI 9600 8-N-1”，如图7-39。

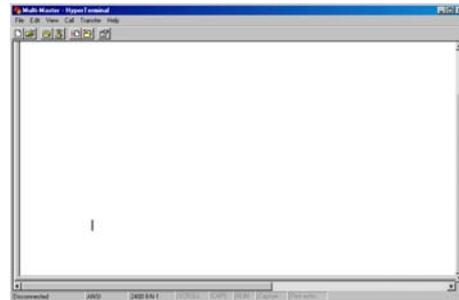


图7-39 超级终端 - 断开ANSI

要通过RS-232/PPI多主站电缆发起通讯，需键入“hhh”。电缆上的Rx LED将在键入“hhh”之后的几秒钟内开始闪烁。在选择语言时，电缆上的TX LED将短暂地变亮。

输入语言选择代号（使用backspace键可清除缺省设置），然后按下ENTER键。图7-40给出了语言选择画面和RS-232/PPI Cable Setup for Remote Operation选择画面。

电缆的硬件版本信息也会显示在该画面中。

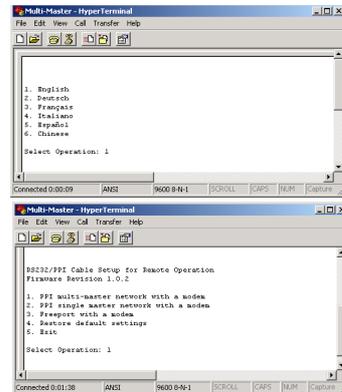


图7-40 超级终端语言选择和RS-232/PPI电缆设置。

RS-232/PPI Cable Setup for Remote Operation画面将一步步地指导您配置电缆，使之适用于所需要的远端操作类型。

- 如果您的STEP 7-Micro/WIN是早期版本，可选择选项2“PPI single master network with a modem”。
- 如果您需通过自由口与Modem连接，可选择选项3。

假如您使用的是STEP 7-Micro/WIN 3.2 Service Pack 4或其后续版本，并且希望通过PPI多主站网络将它与某Modem连接，那么就需要选择选项1。

如图7-41所示，超级终端画面将显示出电缆所需的开关设置。开关设置可以使STEP 7-Micro/WIN通过Modem加入到有一个或多个主站和一个或多个S7-200 PLC的远端网络中。这样的网络如图7-41所示。

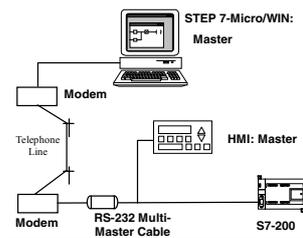
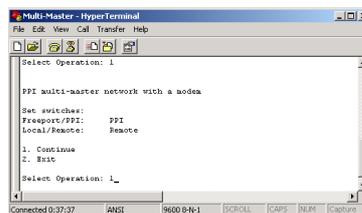


图7-41 超级终端 - RS-232/PPI电缆设置。

在开关设置提示出现后，选择continue。然后就会出现图7-42中的超级终端画面。

远端Modem（与RS-232/PPI多主站电缆连接的Modem）的设置应为工厂缺省设置。远端Modem设置完成后，输入需要的AT字符串，使Modem在连有RS-232/PPI多主站电缆的情况下能正常工作。通常只需输入字符串ATS0=1即可，该字符串会使Modem在第一次通讯中自动应答呼叫。

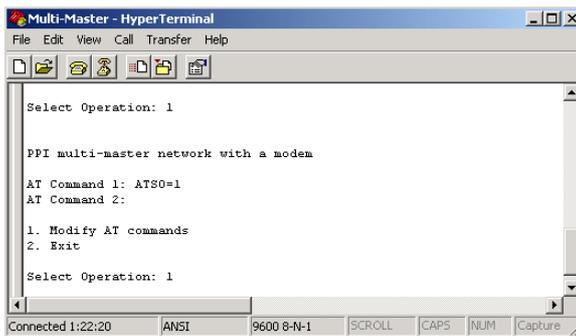


图7-42 超级终端 - 远端Modem

如果您使用的是移动Modem，需要PIN，那么可以输入第二种AT命令提供PIN（关于所用Modem支持的AT命令，可参见Modem手册）。如果要修改AT命令，只需根据提示，进行选择并输入需要的命令即可。提示包含了AT命令字符串的示例，它将在命令格式方面对您有所帮助。

每次电缆通电时，RS-232/PPI多主站电缆都会将这些AT字符串发送给Modem，Modem必须在电缆通电之前，或者几乎是在电缆通电的同时通电。同时，如果Modem周期性的通电，那么电缆也要相应地周期性通电。这样，电缆就能正确地配置Modem，并在最高的可用波特率下工作了。

关于如何输入AT命令，可参考图7-43中的超级终端画面。如果您不需要在提示符处输入第二个AT命令，可按下ENTER键。此时您就能回到修改AT命令或退出的选择画面了。完成AT命令的输入后，可选择Exit退出。

在退出超级终端的RS-232/PPI多主站电缆配置之后，需将电缆与PC断开，并将其连接至Modem上。电源将循环通过Modem和电缆。此时，您就可以在PPI多主网络中用电线进行远端操作了。

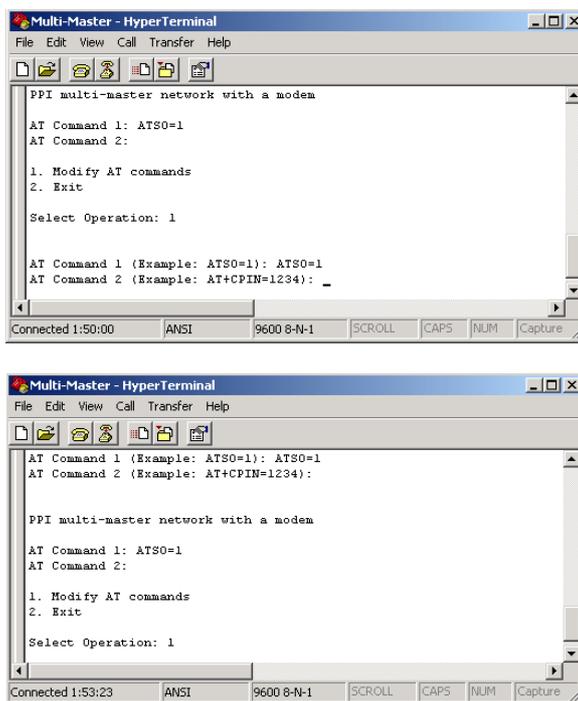


图7-43 超级终端 - AT命令

用超级终端进行自由口操作

用超级终端配置RS-232/PPI多主站电缆的步骤与前述配置实例十分相似。只需根据需要按提示配置电缆即可。

硬件故障诊断指导和软件调试工具

8

STEP 7-Micro/WIN提供软件工具帮助您调试和测试您的程序。这些特征包括：监视S7-200正在执行的用户程序状态，为S7-200指定运行程序的扫描次数，强制变量值等。

使用表8-1作为S7-200硬件故障诊断和找到解决方案的指导。

本章内容:

调试应用程序	244
显示程序状态	246
使用状态图来显示和修改S7-200中的数据	247
强制指定值	248
指定程序执行的扫描周期数	248
硬件故障诊断指导	249

调试应用程序

STEP 7-Micro/WIN 为帮助用户调试程序提供了多种手段：书签，交叉参考表，运行模式下编辑。

使用书签使编程更方便

您在程序中可以使用书签，它可以使您在一个很长的程序中，很方便的在编辑行之间前后移动。您可以移动到程序的下一个标签行或前一个标签行。

使用交叉参考表来检查应用程序



交叉参考

交叉参考表中能够显示应用程序中的交叉参考和元件使用信息。

交叉参考表能够识别程序中使用的所有操作数；程序块、程序段或者程序行的位置以及每一块使用该操作数的相关指令。

您可以在符号地址和绝对地址之间切换来改变所有操作数的表现形式。

	Element	Block	Location	Context
1	I0.0	MAIN (OB1)	Network 1	-I/-
2	SMW32	MAIN (OB1)	Network 1	MOV_W
3	SMB31	MAIN (OB1)	Network 1	MOV_B
4	SM31.7	MAIN (OB1)	Network 1	-I/-
5	SM31.7	MAIN (OB1)	Network 1	-(S)

图8-1 交叉参考表



提示

在交叉参考表中双击某一行可以切换到程序中的相应位置。

在RUN模式下编辑应用程序

S7-200 CPU Rel. 2.0及CPU Rel. 2.0以上的CPU模块支持RUN模式下编辑的功能。RUN模式下编辑功能可以在对控制过程影响较小的情况下，对用户程序进行少量修改。该功能也能使您对程序进行大量的改动，但这样做对程序的执行影响较大，甚至是危险的。



警告

当在RUN模式下向S7-200下载修改过的程序时，修改的程序将立即影响过程操作。在RUN模式下修改程序会导致不可预见的系统操作，可能会导致严重的人身伤害和财产损失。

只有了解RUN模式下修改程序对系统运行会造成何种影响的被授权人员，才可以执行在RUN模式下编辑程序。

要在RUN模式下编辑应用程序，在线的S7-200 CPU必须支持RUN模式下编辑，并且该CPU必须处于RUN状态。

1. 在命令菜单中选择**Debug > Program Edit in RUN**。
2. 如果您打开的项目与S7-200中的程序不同，将提示您存盘。RUN模式下编辑功能只能编辑CPU中的程序。
3. STEP 7-Micro/WIN对于您将在RUN模式下编辑程序提出警告，提示您是继续下一步还是取消操作。如果您选择继续，STEP 7-Micro/WIN会在S7-200中上载程序。现在您可以在RUN模式下编辑程序了。编辑中没有严格的限定。

**提示**

上升沿（EU）和下降沿（ED）指令带一个操作数。若需要查看有关边沿指令的信息，在屏幕上的示窗部分选择交叉参考图标，边沿指令使用标签页中列出了程序中使用的边沿指令的号码。在编辑应用程序时，请注意不要使用重复的号码。

在RUN模式下下载程序

RUN模式编辑功能允许在S7-200处于RUN模式时下载程序块。在下载程序块之前，考虑到RUN模式下编辑对S7-200操作的影响，请注意以下情况：

- 如果在RUN模式编辑状态下取消一个输出控制逻辑，则输出在下次CPU上电之前或CPU转换到STOP模式前将保持上一个状态。
- 如果在RUN模式编辑状态下取消一个正在运行的HSC或PTO/PWM功能，则这些功能在下次CPU上电或CPU转换到STOP模式前将保持运行状态。
- 如果在RUN模式编辑状态下取消ATCH指令，但没有删除中断程序，则在下次CPU上电或CPU转换到STOP模式之前将继续执行中断。同样，如果删除DTCH指令，在下次CPU上电之前或CPU转换到STOP模式前中断将不会停止。
- 如果在RUN模式编辑状态下加入ATCH指令，并且满足第一次扫描标志的条件，则在下次CPU上电或CPU从STOP转换到RUN模式前不会执行这些指令。
- 如果在RUN模式编辑状态下取消ENI指令，则在下次CPU上电之前或CPU从RUN转换到STOP模式前将继续执行中断。
- 如果在RUN模式编辑状态下修改接收指令的地址表，并且在旧程序向新程序转换时接收指令处于激活状态，则所接收的数据写入旧地址表。NETR和NETW指令同样如此。
- 由于RUN模式编辑不影响第一次扫描标志，因此在下次CPU上电之前或CPU从STOP转换到RUN模式前第一次扫描标志的逻辑条件不执行。

**提示**

在RUN模式下下载应用程序，S7-200必须支持RUN模式下编辑；程序编译必须没有错误；STEP 7-Micro/WIN与S7-200之间的通讯必须畅通。

您只能下载程序块。

在RUN模式下下载程序，点击下载按钮或在命令菜单中选择**File > Download**即可。若程序编译正确，STEP 7-Micro/WIN将程序块下载到S7-200上。

退出RUN模式编辑

要退出RUN模式编辑，在命令菜单中选择**Debug > Program Edit in RUN**，然后点击取消复选标志即可。如果修改完后没有存盘，STEP 7-Micro/WIN会有三种提示：继续编辑、下载并退出RUN模式编辑或者不下载退出。

显示程序状态

STEP 7-Micro/WIN允许您在程序执行时监视其状态。当您监视程序状态时，程序编辑器会显示指令操作数的值。

要显示程序状态，点击Program Status按钮或选择命令菜单**Debug > Program Status**即可。

显示LAD和FBD程序的状态

对于显示LAD和FBD程序的状态，STEP 7-Micro/WIN提供了两种选择。

- 扫描结束的状态：STEP 7-Micro/WIN在经过多个扫描周期得到显示状态值之后，刷新屏幕显示状态。状态显示并不反映程序执行时每个元素的实际状态。扫描结束状态不显示L存储器或者累加器的状态。

对于扫描结束状态显示，状态值在所有CPU操作模式下都刷新。

- 执行状态：STEP 7-Micro/WIN在S7-200程序执行过程中，显示程序段中的状态值。要显示执行状态，需在命令菜单中选择**Debug > Use Execution Status**。

对于执行状态，状态值只有在CPU处在RUN模式时才刷新。



提示

STEP 7-Micro/WIN提供一种简单的方法来改变变量的状态。只要选择变量并且用右键单击就会显示选择菜单。

对LAD和FBD程序中的状态显示进行配置

STEP 7-Micro/WIN为在程序中显示状态提供了多种选择。

要为程序状态显示作配置，在命令菜单中选择**Tools > Options**，然后选择程序编辑器并点击程序编辑器标签页即可，如图8-2所示。

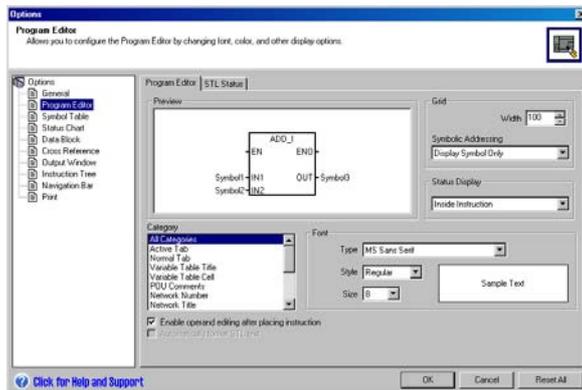


图8-2 状态显示的选项

显示STL程序的状态

您可以监视逐条指令编程的STL程序的执行状态。对于一个STL程序，STEP 7-Micro/WIN在屏幕上显示指令的状态。

从编辑器窗口顶部的第一条STL语句开始，STEP 7-Micro/WIN的S7-200采集状态信息。当您滚动编辑器窗口的屏幕时，新的信息从S7-200采集上来。

STEP 7-Micro/WIN不断地刷新屏幕上的数值。要使屏幕刷新暂停，选择触发暂停按钮。当前值会保持在屏幕上，直到触发暂停按钮失效。

为STL程序中显示的参数作配置

为STL程序中显示的参数作配置STEP 7-Micro/WIN允许您在STL指令中显示多种参数状态。在命令菜单中选择**Tools > Options**，然后选择程序编辑器并点击STL状态标签页。参见图8-3。

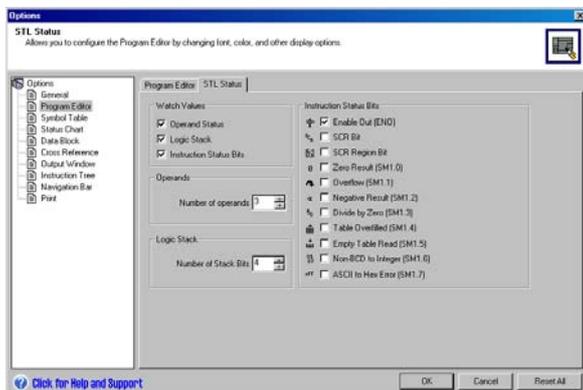


图8-3 STL状态显示选项

使用状态图来显示和修改S7-200中的数据

状态图允许您在S7-200运行程序时，读、写、强制和监视变量数据。在命令菜单中选择**View > Component > Status Chart**来创建一个状态图。图8-4中给出了一个状态图的例子。

您可以创建多个状态图。

STEP 7-Micro/WIN提供工具条图标来对状态图进行操作：升序排列、降序排列、单次读、全部写、强制、全部强制、和读所有强制值。

要为单元选择格式，选择该单元并单击鼠标右键会显示关联菜单。

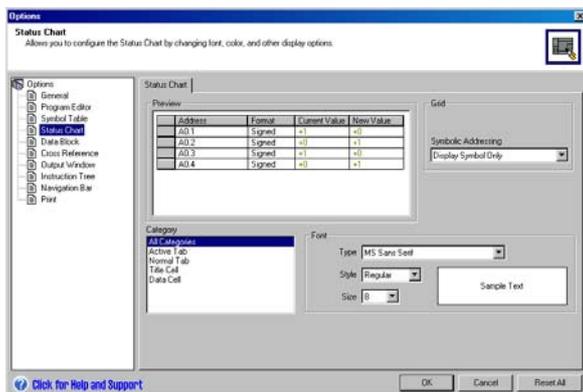


图8-4 状态图

强制指定值

S7-200 CPU允许您用指定值来强制赋给一个或所有的I/O点（I和Q位）。另外您也可以强制改变最多16个内部存储器数据（V或M）或模拟I/O量（AI或AQ）。V和M存储器变量可以按字节、字或双字来改变。模拟量只能以字方式改变，以偶字节开始（如AIW6或AQW14）。所有强制指定值存储在S7-200的永久存储器中。

因为在扫描周期的不同阶段（执行程序、或I/O更新、或通讯处理阶段）可能会改变强制数据。所以在扫描周期的不同时间，CPU又使用了这些强制变量。

- 读取输入：**当读取输入值时，S7-200使用强制指定值作为输入值。
- 在程序中执行控制逻辑：**S7-200使用强制值作为所有立即I/O值。程序执行后，强制数据最多可用于16个存储器。
- 处理通讯请求：**强制数据用于所有读/写通讯。
- 写输出：**S7-200使用强制值作为输出。

您可以使用状态表来强制变量，要强制一个新值，将其输入到状态表的新值列中，然后按工具条上的强制按钮。要将一个已经存在的当前值变为强制值，在当前值列中选中该变量，然后按强制按钮。

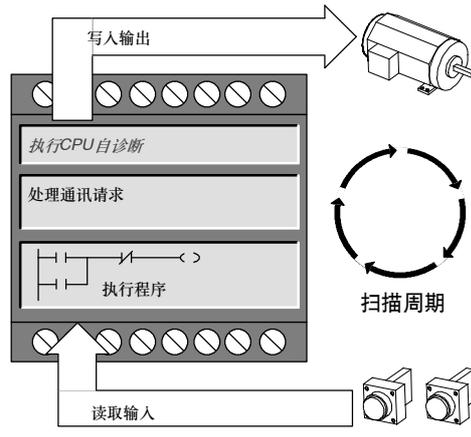


图8-5 S7-200扫描周期



提示

强制功能优于立即I/O指令。强制功能同样优于在STOP模式下使用的输出表。如果S7-200进入STOP模式，输出点上为强制值而不是输出表中配置的值。

指定程序执行的扫描周期数

为了帮助您调试程序，STEP 7-Micro/WIN允许您指定程序的扫描周期数。

您可以使让S7-200在首次扫描时执行一次程序。这使您能够监视首次扫描之后，S7-200中的数据。在命令菜单中选择**Debug > First Scan**即可。

您可以指定程序执行的扫描次数（从1到65,535）。这使您能在变量变化时监视程序。在命令菜单中选择**Debug > Multiple Scans**，然后指定扫描次数即可。

硬件故障诊断指导

表8-1 S7-200硬件故障诊断指导

问题	可能原因	解决方法
输出不工作	<ul style="list-style-type: none"> 被控制的设备产生了损坏 程序错误 接线松动或不正确 输出过载 输出被强制 	<ul style="list-style-type: none"> 当接到感性负载时，（例如电机或继电器），需要使用一个抑制电路，参考第3章。 修改程序。 检查接线，如果不正确，要改正。 检查输出的负载功率。 检查CPU是否有被强制的I/O。
S7-200上SF（系统故障）灯亮（红）	下面给出了可能的原因： <ul style="list-style-type: none"> 用户程序错误 <ul style="list-style-type: none"> - 0003 看门狗错误 - 0011 间接寻址 - 0012 非法的浮点数 - 0014 范围错误 电气干扰（0001到0009） 元件损坏（0001到0010） 	读出致命错误代码号后，其错误类型可参考附录A。 <ul style="list-style-type: none"> 对于编程错误，检查FOR，NEXT，JMP，LBL和比较指令的用法。 对于电气干扰： <ul style="list-style-type: none"> - 参考第3章的接线指南。控制面板良好接地和高电压与低电压不并行引线是很重要的。 - 把24VDC传感器电源的M端子接到地。
LED灯全部不亮	<ul style="list-style-type: none"> 保险丝烧断 24V供电线接反 不正确的供电电压 	把电源分析器连接到系统，检查过电压尖峰的幅值和持续时间。根据检查结果，给系统加一个合适的抑制设备。 有关现场接线的安装信息，请参考第3章。
电气干扰问题	<ul style="list-style-type: none"> 不合适的接地 在控制柜内交叉配线 对快速信号配置了输入滤波器 	参考第3章的接线指南。 控制面板良好接地和高电压与低电压不并行引线是很重要的。 把24VDC传感器电源的M端子接到地。 增加系统数据块中的输入滤波器的延迟时间。
当连接一个外部设备时通讯网络损坏。 （计算机接口、PLC的接口或PC/PPI电缆损坏）	如果所有的非隔离设备（例如PLC、计算机或其它设备）连到一个网络，而该网络没有共同的参考点，通讯电缆提供了一个不期望的电流通路。 这些不期望的电流可以造成通讯错误或损坏电路。	<ul style="list-style-type: none"> 参考第3章的接线指南和第7章的网络指南。 购买隔离型PC/PPI电缆。 当连接没有共同电气参考点的机器时，购买隔离型RS-485到RS-485中继器。 关于S7-200设备的订货号信息，请参考附录C。
STEP 7-Micro/WIN通讯问题	有关网络通讯的信息请参考第7章。	
错误处理	有关错误代码的信息请参考附录A。	

S7-200开环运动控制

S7-200提供有开环运动控制的三种方式:

- 脉宽调制 (PWM) -内置于S7-200, 用于速度、位置或占空比控制。
- 脉冲串输出 (PTO) -内置于S7-200, 用于速度和位置控制。
- EM253位控模块-用于速度和位置控制的附加模块。

为了简化您应用程序中位控功能的使用, STEP7-Micro/WIN提供的位控向导可以帮助您在几分钟内全部完成PWM、PTO或位控模块的组态。该向导可以生成位控指令, 您可以用这些指令在您的应用程序中对速度和位置进行动态控制。对于位控模块, STEP 7-Micro/WIN还提供了一个控制面板, 可以控制、监视和测试您的运动操作。

本章内容:

概述	252
使用PWM (脉宽调制) 输出	253
开环位控用于步进电机或伺服电机的基本信息	255
通过位控向导创建指令	260
PTO指令的错误代码	264
位控模块的特性	265
组态位控模块	267
通过位控向导为位控模块创建指令	273
位控模块的示例程序	285
使用EM253控制面板监视位控模块	290
位控模块和位控指令的错误代码	292
高级议题	294
理解位控模块所支持的参考点寻找模式	303

概述

S7-200提供了三种方式的开环运动控制:

- 脉宽调制 (PWM) -内置于S7-200, 用于速度、位置或占空比控制。
- 脉冲串输出 (PTO) -内置于S7-200, 用于速度和位置控制。
- EM253位控模块-用于速度和位置控制的附加模块。



位置
控制

S7-200提供了两个数字输出 (Q0.0和Q0.1), 该数字输出可以通过位控向导组态为PWM或PTO的输出。位控向导还可以用于组态EM253位控模块。

当组态一个输出为PWM操作时, 输出周期固定, 脉宽或脉冲占空比通过您的程序进行控制。脉冲宽度的变化在您程序中可以控制速度或位置。

当组态一个输出为PTO操作时, 生成一个50%占空比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。内置PTO功能仅提供了脉冲串输出。您的应用程序必须通过PLC内置I/O或扩展模块提供方向和限位控制。

EM253位控模块提供了带有方向控制、禁止和清除输出的单脉冲输出。另外, 专用输入允许将模块组态为包括自动参考点搜索在内的几种操作模式。模块为步进电机或伺服电机的速度和位置开环控制提供了统一的解决方案。

为了简化您的应用程序中位控功能的使用, STEP7-Micro/WIN提供的位控向导可以帮助您在几分钟内全部完成PWM, PTO或位控模块的组态。向导可以生成位置指令, 您可以用这些指令在您的应用程序中为速度和位置提供动态控制。对于位控模块, STEP 7-Micro/WIN还提供了一个控制面板, 可以控制, 监视和测试您的位控操作。

使用PWM（脉宽调制）输出

PWM可输出周期固定但占空比可变的脉冲。以指定频率（周期）启动后，PWM持续输出。脉冲宽度根据所需的控制要求进行变化。占空比可以表示为周期的一个百分比或对应于脉冲宽度的一个时间值。脉冲宽度可以从0%（无脉冲，一直为低电平）变化到100%（无脉冲，一直为高电平）。如图9-1。

由于PWM输出可以从0%变化到100%，在很多情况下，它可以提供一个类似于模拟量输出的数字量输出。例如，PWM输出可以用于电机从停止到最大速度的控制，或用于阀从关到全开的位置值控制。

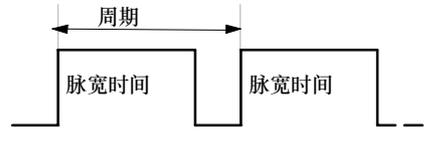


图9-1 脉宽调制（PWM）

组态PWM输出

使用STEP 7-Micro/WIN位控向导，为PWM控制组态一个内置输出。启动位控向导，可以点击浏览条中的工具图标，然后双击位控向导图标，或者从命令菜单中选择**Tools > Position Control Wizard**。如图9-2

1. 为S7-200 PLC选择选项组态板载PTO/PWM操作。
2. 选择Q0.0或Q0.1输出组态为PWM输出。
3. 然后，从下拉对话框中选择脉宽调制（PWM），选择微秒或毫秒作为时基并指定周期。
4. 选择完成结束向导。

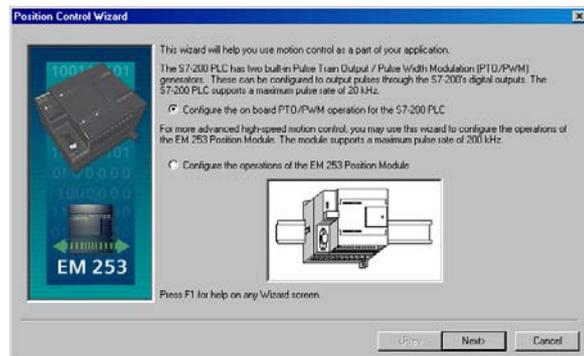


图9-2 组态PWM输出

向导将为您生成一个指令以控制PWM输出的占空比。

PWMx_RUN指令

PWMx_RUN指令允许您通过改变脉冲宽度从0到一个周期的宽度来控制输出占空比。

周期输入是一个为PWM输出定义周期的字值。允许的变化范围是2到65535个在向导中指定的时基单元（微秒或毫秒）。

占空比输入是一个为PWM输出定义脉宽的字值。值允许的变化范围是0到65535个在向导中指定的时间基准单元（微秒或毫秒）。

Error是一个由PWMx_RUN返回的字节值，指示执行的结果。参看可能的错误代码描述表。

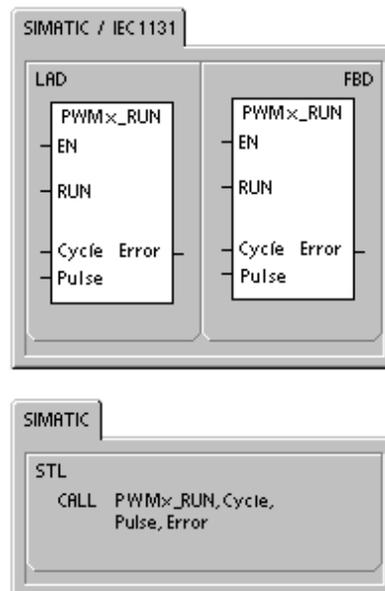


表9-1 PWMx_RUN指令参数

输入/输出	数据类型	操作数
Cycle, Duty_Cycle	WORD	IW、QW、MW、SW、SMW、LW、T、C、AC、AIW、*VD、*AC、*LD、常数
Error	BYTE	IB、QB、VB、MBV、SMB、LB、AC、*VD、*AC、*LD、常数

表9-2 PWMx_RUN指令错误代码

错误代码	描述
0	无错误，正常完成
1	在运行中立即发出STOP指令。STOP命令成功完成

开环位控用于步进电机或伺服电机的基本信息

内置于S7-200 PLC的PTO和EM253位控模块都使用一个脉冲串输出用于步进电机或伺服电机的速度和位置控制。

使用PTO或模块用于开环位置控制需要运动控制领域的专业技术。本章内容并不用于培训。而是，提供基础信息以帮助您使用STEP 7-Micro/WIN位控向导为您的应用程序组态PTO或模块。

最大速度和启动/停止速度

向导将提示您应用程序的最大速度（MAX_SPEED）和启动/停止速度（SS_SPEED）。如图9-3。

- ❑ MAX_SPEED: 该数值是您的应用中操作速度的最大值，它应在电机力矩能力的范围内。驱动负载所需的力矩由摩擦力、惯性以及加速/减速时间决定。
- ❑ 位控向导根据指定的MAX_SPEED，计算并显示位控模块所能控制的最小速度。
- ❑ 对于PTO输出，您必须指定期望的启动/停止速度。由于启动/停止速度在每次运动指令执行时至少会产生一次，所以启动/停止速度的周期应小于加速/减速时间。
- ❑ SS_SPEED: 输入该数值满足您的电机在低速时驱动负载的能力，如果SS_SPEED的数值过低，电机和负载在运动的开始和结束时可能会摇摆或颤动。如果SS_SPEED的数值过高，电机会在启动时丢失脉冲，并且负载在试图停止时会使电机过载。

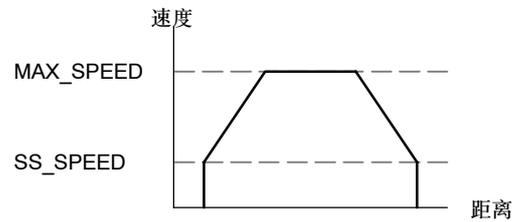


图9-3 最大速度和启动/停止速度

在电机的数据单中，对于电机和给定负载，有不同的方式定义启动/停止（或拉入/拉出）速度。通常，SS_SPEED值是MAX_SPEED值的5%至15%。请参考电机的数据单，为您的应用选择正确的速度。图9-4所示为典型的电机电矩/速度曲线。

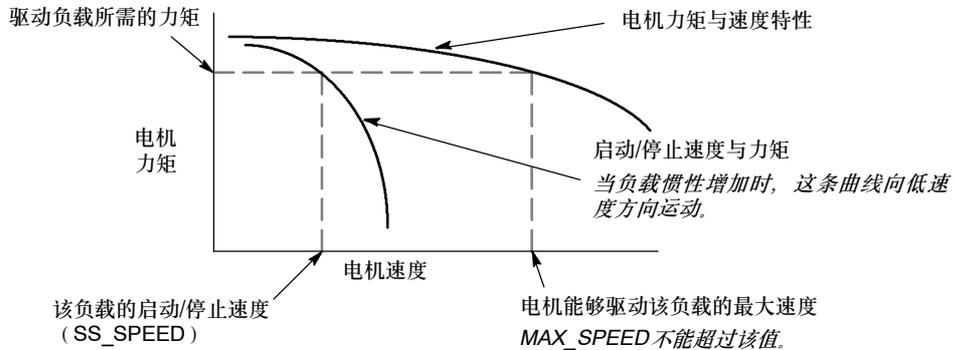


图9-4 典型电机电矩-速度曲线

输入加速和减速时间

作为组态内容的一部分，要设置加速和减速时间。加速时间和减速时间的缺省设置都是1秒。通常，电机可在小于1秒的时间内工作。参见图9-5。您要以毫秒为单位进行时间设定：

- ACCEL_TIME: 电机从SS_SPEED速度加速到MAX_SPEED速度所需的时间。缺省值=1000ms。
- DECEL_TIME: 电机从MAX_SPEED速度减速到SS_SPEED速度所需要的时间。缺省值=1000ms。

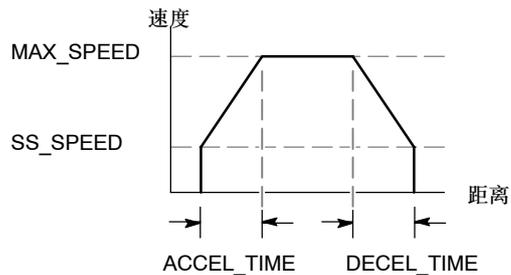


图9-5 加速和减速时间



提示

电机的加速和减速时间要经过测试来确定。开始时，您应输入一个较大的值。逐渐减少这个时间值直至电机开始停止，从而优化您应用中的这些设置。

组态移动包络

一个包络是一个预先定义的移动描述，它包括一个或多个速度，影响着从起点到终点的移动。即使不定义包络也可以使用PTO或模块，位控向导为您提供了指令以用于控制移动而无需运行一个包络。

一个包络由多段组成，每段包含一个达到目标速度的加速/减速过程和以目标速度匀速运行的一串固定数量的脉冲。如果是单段运动控制或者是多段运动控制中的最后一段，还应该包括一个由目标速度到停止的减速过程。

PTO支持最大100个包络，而模块支持最大25个包络。

定义移动包络

位控向导提供移动包络定义，在这里，您可以为您的应用程序定义每一个移动包络。对每一个包络，您可以选择操作模式并为每个包络的各步定义指标。位控向导中可以为每个移动包络定义一个符号名，其做法是您在定义包络时输入一个符号名即可。

选择包络的操作模式

您要按照操作模式组态包络。PTO支持相对位置和单一速度的连续转动。而位控模块支持绝对位置、相对位置、单一速度连续转动和以两种速度连续转动。图9-6所示为不同的操作模式。

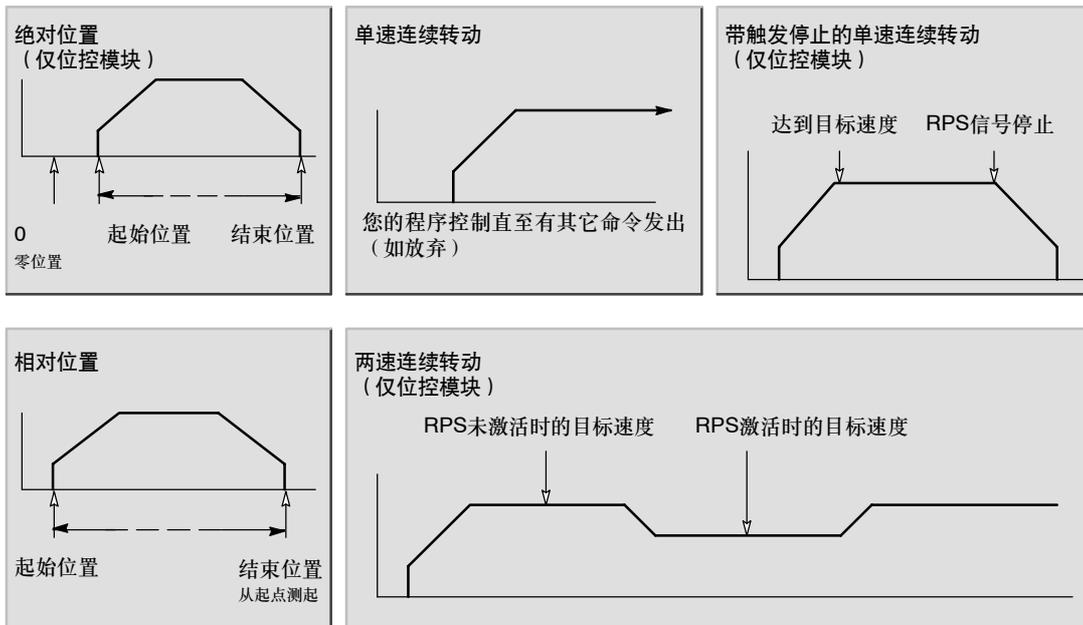


图9-6 位控模块的模式选择

创建包络中的步

一个步是工件运动的一个固定距离，包括加速和减速时间内的距离。PTO每一包络最大允许29个步，而模块的每一包络最大允许4个步。

您要为每一步指定目标速度和结束位置或脉冲数目，且每次输入一步。图9-7所示为一步、两步、三步和四步包络。

注意一步包络只有一个常速段，两步包络有两个常速段，依次类推。步的数目与包络中常速段的数目一致。

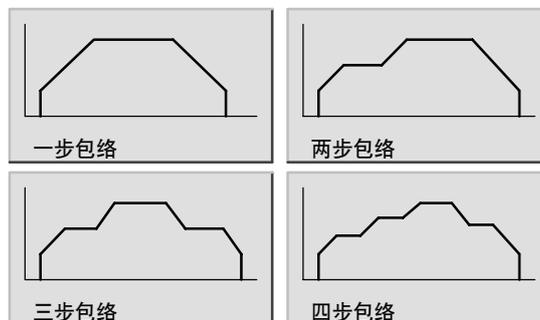


图9-7 移动包络示例

使用PTO输出

PTO提供一个指定脉冲数目的方波输出（50%占空比）每一脉冲的频率或周期随着加速和减速时的频率线性变化，而在移动的常频率段部分保持不变。一旦产生完指定数目的脉冲，PTO输出变为低电平，并且直到装载一个新的指定值时才产生脉冲。参见图9-8。

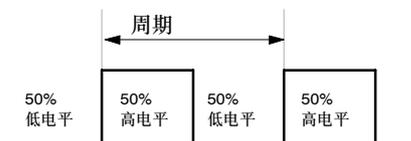


图9-8 脉冲串输出（PTO）

组态PTO输出

使用STEP 7-Micro/WIN位控向导，为PTO操作组态一个内置输出。启动位控向导，可以点击浏览条中的工具图标，然后双击位控向导图标，或者选择菜单命令**Tools > Position Control Wizard**。

1. 为S7-200 PLC选择选项组态板载PTO/PWM操作。
2. 选择Q0.0或Q0.1，组态作为PTO的输出。
3. 从下拉对话框中选择线性脉冲串输出（PTO）。
4. 若您想监视PTO产生的脉冲数目，点击复选框选择使用高速计数器。
5. 在对应的编辑框中输入MAX_SPEED和SS_SPEED速度值。
6. 在对应的编辑框中输入加速和减速时间。
7. 在移动包络定义界面，点击新包络按钮允许定义包络。选择所需的操作模式。

对于相对位置包络:

输入目标速度和脉冲数。然后，您可以点击绘制步按钮，查看移动的图形描述。

若需要多个步，点击新建步按钮并按要求输入步信息。

对于单速连续转动:

在编辑框中输入单速值。

若您想终止单速连续转动，点击子程序编程复选框，并输入停止事件后的移动脉冲数。

8. 根据移动需要，您可以定义多个包络和多个步。
9. 选择完成结束向导。

通过位控向导创建指令

通过创建四个唯一的指令子程序，位控向导使得控制内置PTO更加容易。每一个位控指令都有一个“PTOx_”前缀，其中x为模块位置。

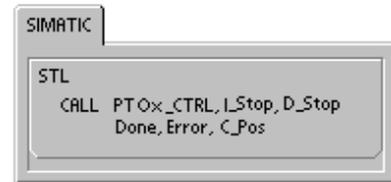
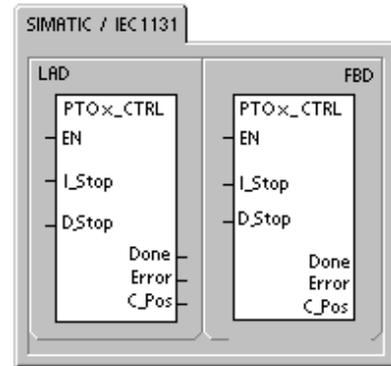
USS_CTRL子程序

PTOx_CTRL子程序（控制）使能和初始化用于步进电机或伺服电机的PTO输出。在您的程序中仅能使用该子程序一次，并保证每个扫描周期该子程序都被执行。一直使用SM0.0作为EN输入的输入。

I_STOP（立即STOP）输入量为一个布尔量输入。当输入为低电平时，PTO功能正常操作。当输入变为高电平时，PTO立即终止脉冲输出。

D_STOP（减速STOP）输入量为一个布尔量输入。当输入为低电平时，PTO功能正常操作。当输入变为高电平时，PTO产生一个脉冲串将电机减速到停止。

Done输出是一个布尔量输出。当Done位为高电平时，表明CPU已经执行完子程序。



当Done位为高电平时，Error字节以一个无错误代码或错误代码来报告正常完成。对于错误代码的定义，见表9-7。

若在向导中已使能HSC，则C_Pos参数包含以脉冲数表示的模块当前位置。否则，当前位置将一直为0。

表9-3 POSx_CTRL指令的参数

输入/输出	数据类型	操作数
I_STOP	BOOL	I、Q、V、M、SM、S、T、C、L、能流
D_STOP	BOOL	I、Q、V、M、SM、S、T、C、L、能流
Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
C_Pos	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD

PTOx_RUN子程序

PTOx_RUN子程序（运行包络）命令PLC在一个指定的包络中执行运动操作，此包络存储在组态/包络表中。

接通EN位，使能该子程序。确保EN位保持接通，直至Done位指示该子程序已完成。

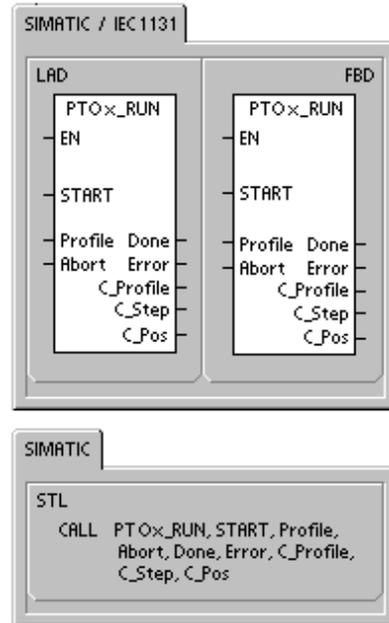
接通START参数以初始化包络的执行。对于每次扫描，当START参数接通且PTO当前未激活时，指令激活PTO。要保证该命令只发一次，使用边沿检测指令以脉冲触发START参数接通。

包络参数包含该移动包络的号码或符号名。

接通参数Abort，命令位控模块停止当前的包络并减速直至电机停下。

模块完成该指令时，参数Done接通。

参数Error包含指令的执行结果。如表9-7所示为错误代码定义。



参数C_Profile包含位控模块当前正在执行的包络。

参数C_Step包含当前正在执行的包络的步。

若在向导中已允许HSC，则C_Pos参数包含以脉冲数作为模块的当前位置。否则，当前位置将一直为0。

表9-4 PTOx_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、能流
Profile	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD、常数
Abort、Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
Error、C_Profile、C_Step	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
C_Pos	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD

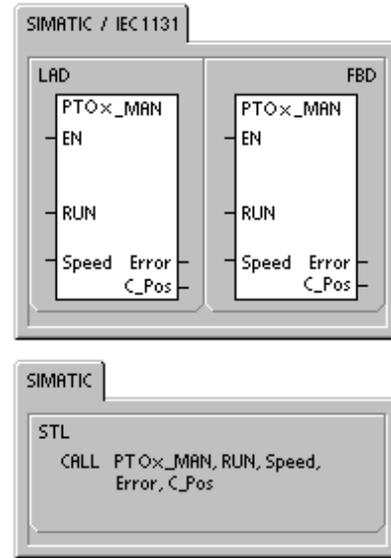
PTOx_MAN子程序

PTOx_MAN子程序（手动模式）使PTO输出置为手动模式。该子程序允许电机以不同的速度启动、停止和运行。当使能PTOx_MAN子程序时，不允许执行其它PTO子程序。

使能RUN（Run/Stop）参数，命令PTO加速到指定速度（速度参数）。即使在电机运行时，您也可以改变速度参数的值。禁止参数RUN则命令PTO减速，直至电机停止。

参数Speed决定RUN使能时的速度。速度是一个每秒多少个脉冲的双整型（DINT）值。电机运行时可以修改该速度参数。

参数Error包含指令的执行结果。如表9-7所示为错误代码定义。



若在向导中已允许HSC，则C_Pos参数包含以脉冲数作为模块的当前位置。否则，当前位置将一直为0。

表9-5 PTOx_MAN指令的参数

输入/输出	数据类型	操作数
RUN	BOOL	I、Q、V、M、SM、S、T、C、L、能流
SPEED	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD、常数
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
C_Pos	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD



提示

PTO也许不会响应速度参数小的变化，特别是如果组态的加速或减速时间较短并且最大速度和启动/停止速度相差较大的情况。

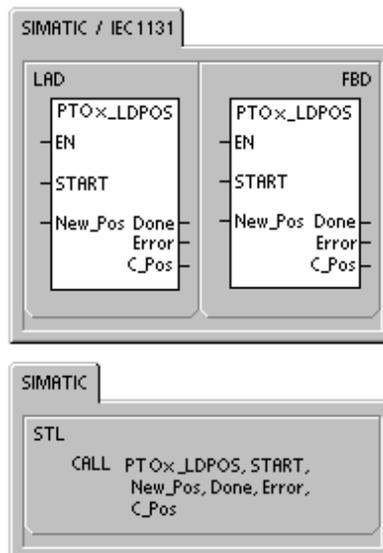
PTOx_LDPOS指令

PTOx_LDPOS指令（装载位置）改变PTO脉冲计数器的当前位置值为一个新值。您可以使用该指令为任何一个运动命令建立一个新的零位置。

接通EN位使能该指令。确保EN位保持接通直至Done位指示该指令完成。

接通START参数，以装载一个新的位置值到PTO脉冲计数器。每一循环周期，只要START参数接通且PTO当前不忙，该指令装载一个新的位置给PTO脉冲计数器。要保证该命令只发一次，使用边沿检测指令以脉冲触发START参数接通。

New_Pos参数提供一个新的值替代报告的当前位置值。位置值用脉冲数表示。



模块完成该指令时，参数Done接通。

参数Error包含指令的执行结果。如表9-7所示为错误代码定义。

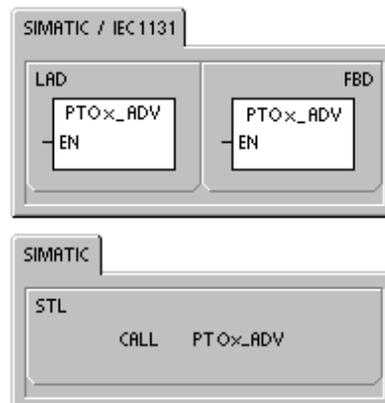
若在向导中已允许HSC，则C_Pos参数包含以脉冲数作为模块的当前位置。否则，当前位置将一直为0。

表9-6 PTOx_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、能流
New_Pos、C_Pos	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD
Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

PTOx_ADV子程序

PTOx_ADV子程序停止当前的连续运动包络，并增加向导包络定义中指定的脉冲数。若您已经通过在位控向导中使能的PTOx_ADV选项指定了至少一个单速连续转动，则该子程序被创建。



PTO指令的错误代码

表9-7 PTO指令错误代码

错误代码	描述
0	无错误，正常完成
1	在运行中执行立即STOP命令。STOP命令成功完成
2	在运行中执行减速STOP命令。STOP命令成功完成
3	在脉冲发生器中或PTO表的格式化中检测到的执行错误
127	HSC、PLS或PTO指令在执行过程中产生一个ENO错误
128	请求无法处理。CPU要么正在忙于处理另一个请求，要么该请求无START脉冲。
129	立即STOP和减速STOP命令都在同一时间使能
130	PTO指令当前正被命令为STOP
132	没有为指定包络所组态的包络块

位控模块的特性

位控模块可为您提供单轴、开环位置控制所需要的功能和性能。

- 提供高速控制，速度从每秒20个脉冲到每秒200,000个脉冲
- 支持急停（S曲线）或线性的加速、减速功能
- 提供可组态的测量系统，既可以使用工程单位（如英寸或厘米）也可以使用脉冲数
- 提供可组态的啮合间隙补偿
- 支持绝对、相对和手动的位控方式
- 提供连续操作
- 提供多达25组的移动包络，每组最多可有4种速度
- 提供4种不同的参考点寻找模式，每种模式都可对起始的寻找方向和最终的接近方向进行选择
- 提供可拆卸的现场接线端子便于安装和拆卸。

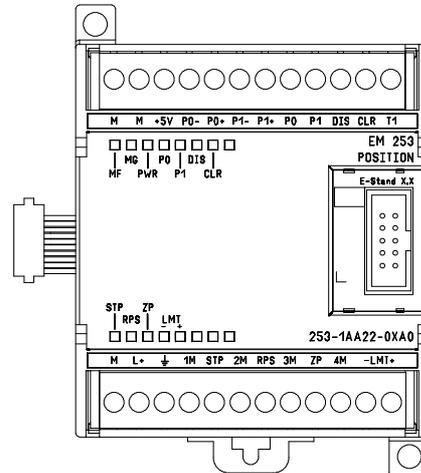


图9-9 EM253位控模块

使用STEP 7-Micro/WIN可生成位控模块所使用的所有组态和移动包络信息。这些信息和您的程序块一起下载到S7-200中。由于位控模块所需要的全部信息都存储在S7-200中，当您更换位控模块时不必重新编程或组态。

S7-200在输出的过程映像区中（Q区）保留8位作为位控模块的接口。S7-200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输出都不相连。

位控模块提供5个数字输入和4个数字输出与您的运动控制应用相连，见表9-1。参见表9-8。这些输入输出位于位控模块上。附录A给出了位控模块的详细规范，另外还提供了位控模块与一些常用的电机驱动/放大单元的接线图。

表9-8 位控模块的输入和输出

信号	描述
STP	STP输入可让模块停止脉冲输出。在位控向导中可选择您所需要的STP操作。
RPS	RPS（参考点切换）输入可为绝对运动操作建立参考点或零点位置。
ZP	ZP（零脉冲）输入可帮助建立参考点或零点位置。通常，电机驱动器/放大器每周产生一个ZP脉冲
LMT+ LMT-	LMT+和LMT-是运动位置的最大限制。位控向导中可以组态LMT+和LMT-输入。
P0 P1 P0+, P0- P1+, P1-	P0和P1是漏型晶体管输出，用以控制电机的运动和方向。P0+、P0-以及P1+、P1-是差分脉冲输出，与P0和P1的功能一样，但所提供的信号质量更好。漏型输出和差分输出同时有效。您可以根据电机驱动器/放大器的接口要求来选择使用哪种输出。
DIS	DIS是一个漏型输出，用来禁止或使能电机驱动器/放大器。
CLR	CLR是一个漏型输出，用来清除伺服脉冲计数器。

位控模块编程

STEP 7-Micro/WIN为位控模块的组态和编程提供便捷的工具。遵循以下步骤即可：

1. 组态位控模块。STEP 7-Micro/WIN提供一个位控向导，可生成组态/包络表和位控指令。有关组态位控模块的信息，请参看270页上的组态位控模块。
2. 测试位控模块的操作。STEP 7-Micro/WIN提供一个EM253控制面板，用以测试位控模块的输入、输出接线组态以及运动路径的执行。有关EM 253控制面板的信息，请参看第334页。
3. 创建S7-200的执行程序。位控向导自动生成位控指令。您可以将这些指令插入您的程序中。
 - 要启用位控模块，请插入一个POSx_CTRL指令。用SM0.0（始终接通）以确保这条指令在每一个循环周期中都得到执行。
 - 要将电机移动到一个指定位置，使用一条POSx_GOTO指令或一条POSx_RUN指令。POSx_GOTO指令使电机运动到您在程序中输入的指定位置。POSx_RUN指令则使电机按照您在位控向导中所组态的路线运动。
 - 要使用绝对坐标进行运动，您必须为您的应用建立零位置。使用一条POSx_RSEEK或一条POSx_LDPOS指令建立零位置。
 - 位控向导生成的其它指令为典型应用提供需要的功能，对于您的特定应用，这些指令是可选的。
4. 编译您的程序并将系统块、数据块和程序块下载到S7-200中。



提示

关于连接位控模块到几个通用步进电机控制器的信息，请参照附录A。



提示

要想与位控向导中的缺省设置相匹配，则将步进电机控制器上的DIP开关设置至每转10,000个脉冲。

组态位控模块



位置
控制

要进行位移控制必须为位控模块创建组态/包络表。位控向导引导您一步一步完成整个组态过程，非常便捷。关于组态/包络表的详细信息请参看高级议题。

使用位控向导可离线创建组态/包络表。您可以在不连接S7-200 CPU及位控模块的情况下进行组态。

要运行位控向导，必须对项目进行编译并选择符号寻址方式。

启动位控向导，可以点击浏览条中的工具图标，然后双击位控向导图标，或者选择菜单命令 **Tools > Position Control Wizard**。

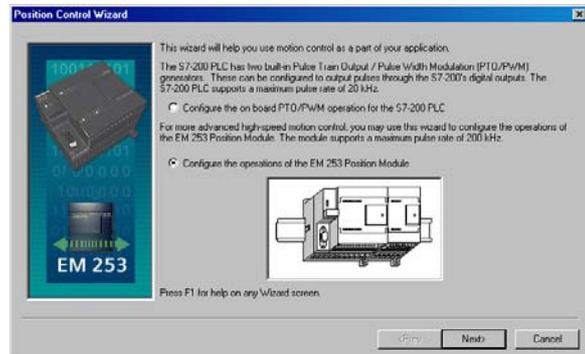


图9-10 位控向导

使用STEP 7-Micro/WIN位控向导组态位控模块。选择选项以组态EM253位控模块。

输入模块的位置

指定模块插槽位置（模块0到模块6）。若STEP 7-Micro/WIN被连接到PLC，您仅需点击Read Modules按钮。对于一个硬件版本是1.2之前的S7-200 CPU，模块必须安装在紧邻CPU的位置。

选择测量类型

选择测量系统。您可以选择使用工程单位或脉冲。若您选择脉冲，则无需其它的信息。而若您选择工程单位，则需要电机转一周产生的脉冲数（参考电机或驱动的数据表单）、测量基准单元（如英寸、英尺、毫米或厘米）和电机转一周运动的距离。

- STEP 7-Micro/WIN提供一个EM253控制面板，对已组态的位控模块，通过该面板可修改每周的单位数。
- 如果您在以后改变了测量系统，必须删除整个组态，包括位控向导生成的所有指令。您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组态

要编辑或查看集成输入/输出的缺省组态，选择Advanced Options按钮。

- 使用Input Active Levels标签页设置激活等级（高或低）。等级设为高时，当输入有电流时，读到逻辑1。等级设为低时，当输入无电流时，读到逻辑1。逻辑1总是解释为条件激活。不论激活等级是怎样的，输入有电流时LED灯亮。（缺省=高）
- 使用Input Filter Times标签页为STP、RPS、LMT+和LMT-设置滤波时间常数（0.20毫秒到12.80毫秒）。增加滤波时间可以去除更多噪音，但将降低对一个信号状态改变的响应时间。（缺省=6.4ms）
- 使用Pulse and Directional Outputs标签页可用来选择输出极性和方向控制方式。对于极性和方向控制方式设置的作用，见图9-11和9-12。

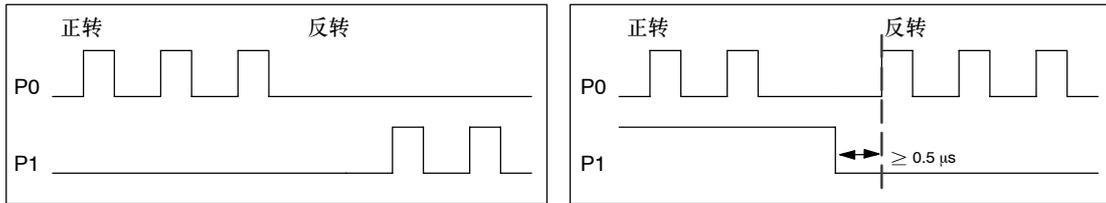


图9-11 正极性的转向选项

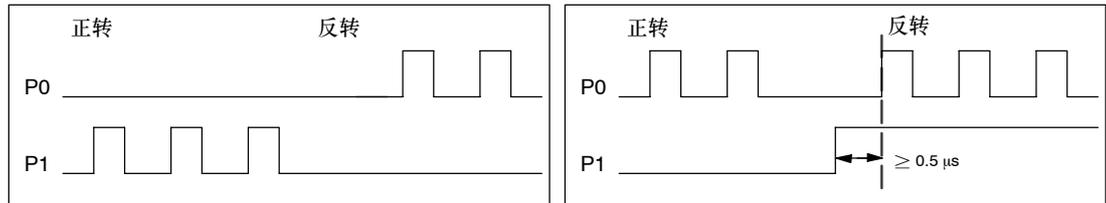


图9-12 负极性的转向选项



警告

控制设备可能在不安全条件下出现故障，并可能导致被控设备不可预知的操作。这些不可预知的操作可能导致人员的伤亡以及/或设备的损坏。

位控模块的限位和停止功能是电逻辑实现的，不能够提供机电控制所能提供的保护等级。请考虑使用独立于S7-200 CPU和位控模块的急停功能，机电互锁或冗余的机电保护。

组态模块对物理输入的响应

接下来，为LMT+、LMT-、和STP输入选择模块响应。使用下拉框进行选择：无动作（忽略输入条件）、减速至停止（缺省）或立即停止。

输入最大启动和停止速度

为您的应用输入最大速度（MAX_SPEED）和启动/停止速度（SS_SPEED）。

输入点动参数

接下来，输入JOG_SPEED和JOG_INCREMENT速度值。

- JOG_SPEED: JOG_SPEED（电机的拖动速度）是点动命令有效时能够得到的最大速度。
- JOG_INCREMENT: 是瞬间的点动命令能够将工件运动的距离。

图9-13所示为点动命令的操作。当位控模块收到一个点动命令后，它启动一个定时器。如果点动命令在0.5秒到时之前结束，位控模块则以定义的SS_SPEED速度将工件运动JOG_INCREMENT数值指定的距离。当0.5秒到时，点动命令仍然是激活的，位控模块加速至JOG_SPEED速度。继续运动直至点动命令结束。位控模块随后减速停止。您可以在EM253控制面板中使能点动命令，或者使用位控指令。

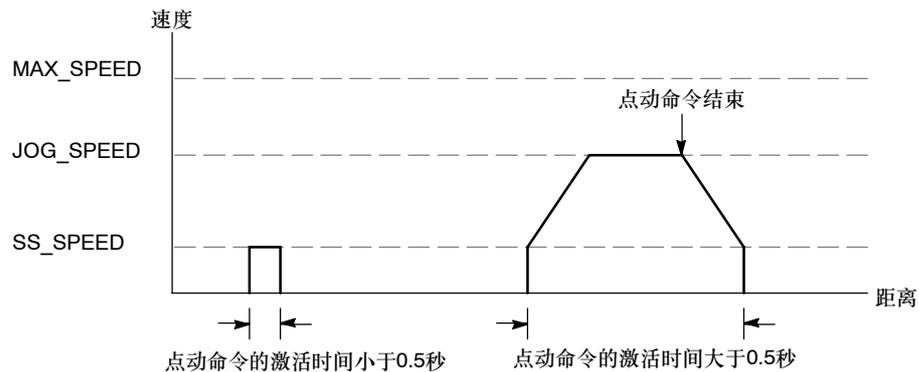


图9-13 JOG操作的表达

输入加速时间

在编辑框中输入加速和减速时间

输入急停时间

对于单步运动，输入急停时间补偿。通过减小运动包络的加速和减速部分的急停（改变速率）来提供更为平滑的位置控制。参见图9-14。

急停时间补偿也被称为“S曲线包络”。这种补偿同样地作用于加速曲线和减速曲线的开始和结束部分。急停补偿不能够应用在介于零速和SS_SPEED速度之间的初始段和结束段中。

您可以输入一个时间值 (JERK_TIME) 来指定急停补偿。这一时间是加速从零到达到最大加速度所需要的时间。和减少 ACCEL_TIME 和 DECEL_TIME 相比, 一个较长的急停时间由于能够使整个循环时间只有一个较小的增加, 从而可以产生更为平滑的操作。零值表示没有应用任何补偿。

(缺省=0ms)

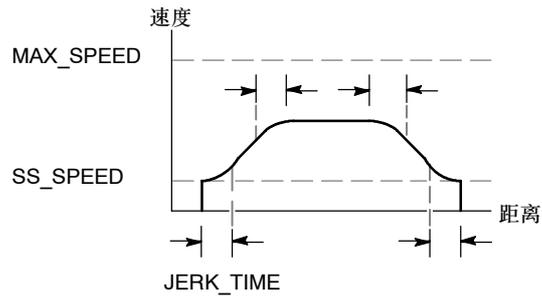


图9-14 急停补偿



提示

对于 JERK_TIME 来说, 一个好的初始值是 ACCEL_TIME 的 40%。

组态参考点和寻找参数

为您的应用选择使用参考点或不使用参考点。

- 若您的应用需要从一个绝对位置处开始运动或以绝对位置作为参考, 您必须建立一个参考点 (RP) 或零点位置, 该点将位置测量固定到物理系统的一个已知点上。
- 若使用一个参考点, 您将需要定义自动定位参考点的方法。自动定位参考点的过程称为参考点寻找 (RPS)。在向导中定义参考点寻找过程需要两步。

输入参考点寻找速度 (快速寻找速度和慢速寻找速度)。定义初始寻找方向和最终参考点接近方向。使用高级 RP Options 按钮输入 Reference Point Offset 和螺距补偿值。

RP_FAST 是模块执行 RP 寻找命令的初始速度。通常 RP_FAST 是 MAX_SPEED 的 2/3 左右。

RP_SLOW 是接近 RP 的最终速度。通常使用一个较慢的速度去接近 RP 以免错过。RP_SLOW 的典型值为 SS_SPEED。

RP_SEEK_DIR 是 RP 寻找操作的初始方向。通常, 这个方向是从工作区到 RP 附近。限位开关在确定 RP 的寻找区域时扮演重要角色。当执行 RP 寻找操作时, 遇到限位开关会引起方向反转, 使寻找能够继续下去。(缺省=反向)。

RP_APPR_DIR 是最终接近 RP 的方向。为了减小啮合间隙和提供更高的精度, 应该按照从 RP 移动到工作区所使用的方向来接近参考点。(缺省=正向)。

- 位控向导提供高级参考点选项，可以指定一个RP偏移量 (RP_OFFSET)，这个偏移量是指从RP到零位置的距离，参见图9-15。

RP_OFFSET: 在物理的测量系统中RP到零位置之间的距离。(缺省=0)

啮合间隙补偿: 当方向发生变化时，为消除系统中的滞慢(啮合间隙)，电机必须运动的距离。啮合间隙补偿总是正值。(缺省=0)



图9-15 RP与零位置的关系

选择参考点搜索顺序。

- 位控模块提供了一个参考点开关 (RPS) 输入，在搜索RP的过程中使用。以RPS为参考确定一个准确的位置作为RP。位控模块提供一个外部参考点开关 (RPS) 传感器，用于寻找RP。

您可以为位控模块组态参考点寻找的顺序。图9-16所示为一个简化了的缺省RP寻找顺序图。您可以为RP搜寻顺序作以下选择：

RP寻找模式0: 不执行RP寻找顺序

RP寻找模式1: RP位于RPS输入有效区接近工作区的一边开始有效的位置上。(缺省)

RP寻找模式2: RP位于RPS输入有效区的中央。

RP寻找模式3: RP位于RPS输入有效区之外。RP_Z_CNT指定了在RPS失效之后应接收多少个ZP(零脉冲)输入。

RP寻找模式4: RP通常位于RPS输入的有效区内。RP_Z_CNT指定在RPS激活后应接收多少个ZP(零脉冲)输入。

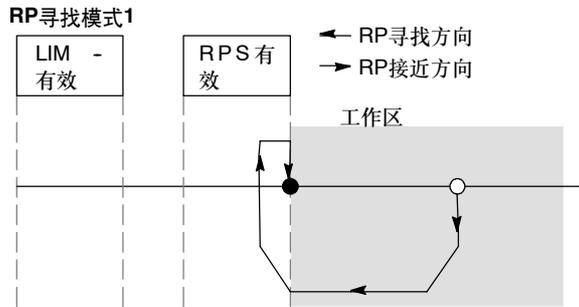


图9-16 缺省的RP搜寻顺序(简化的)



提示
RPS激活区 (RPS输入保持激活的距离) 必须大于从RP_FAST减速到RP_SLOW速度所需的距离。若距离太短，位控模块将产生一个错误。

命令字节

接下来为命令字节输入Q字节地址。命令字节是一个8位数字量输出的地址，该地址保留在用于访问位控模块的过程影像寄存器中。对于I/O计数的描述，参看第4章图4-10。

定义移动包络

在移动包络定义界面中，点击新建包络按钮，允许定义包络。选择所需的操作模式。

- 对于绝对位置包络:
输入目标速度和终点位置。然后，您可以点击打印步按钮，观察运动的图形描述。
若需要多个步，点击新建步按钮并按要求输入步信息。
- 对于相对位置包络:
输入目标速度和终点位置。然后，您可以点击打印步按钮，观察运动的图形描述。
若需要多个步，点击新建步按钮并按要求输入步信息。
- 对于单速连续转动:
在编辑框中输入单速度值。
选择转动方向
若您想用RPS输入终止单速连续转动运动，点击复选框。
- 对于两速连续转动:
在编辑框中输入RPS为高时的目标速度值。
在编辑框中输入RPS为低时的目标速度值。
选择转动方向

为了完成需要的运动，您可以按照定义任意多个包络和步。

完成组态

当您完成对位控模块的组态时，只需点击完成（Finish），然后位控向导会执行以下任务：

- 将模块的组态和包络表插入到您的S7-200程序的数据块中。
- 为位控参数生成一个全局符号表
- 在项目的程序块中增加位控指令子程序，您可在应用中使用这些指令

要修改任何组态或包络信息，您可以再次运行位控向导。



提示

由于位控向导修改了程序块、数据块和系统块，要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操作所需的所有程序组件。

通过位控向导为位控模块创建指令

位控向导能够根据位控模块的位置和您对模块所作的组态生成唯一的指令子程序，从而使位控模块的控制变得非常容易。每条位控指令都有一个前缀“POSx_”，这里x是模块位置。由于每个位控指令是一个子程序，11条位控指令使用11个子程序。



提示

位控指令使用户程序对存储空间的需求最多增加至1700字节。您可以删除不用的位控指令以减小对存储空间的需求。要恢复已删除的位控指令，只需再次运行位控向导即可。

位控指令使用指南

您必须确保每次仅有一个位控指令是激活的。

您可以在一个中断程序中执行POSx_RUN和POSx_GOTO。但是，当模块正忙于处理其它命令时，千万不要试图在中断程序中启动指令。如果您在一个中断程序中启动一条指令，您可以使用POSx_CTRL指令的输出来监控位控模块是何时完成运动的。

位控向导按照您所选的测量系统自动组态速度参数（Speed和C_Speed）和位置参数（Pos或C_Pos）的数值。对于脉冲，这些参数是双整数，对于工程单位，这些参数是您所选的单位的实数值。例如：选厘米（cm）为单位，位控参数则存为以厘米为单位的一个实数值，速度参数则选一个每秒若干厘米的实数值（cm/sec.）。

以下是特定的运动控制任务所需的位控指令：

- 在您的用户程序中插入POSx_CTRL，并以SM0.0为条件使之每个循环都执行。
- 要指定运动到一个绝对位置，您必须首先使用POSx_RSEEK或POSx_LDPOS指令建立零位置。
- 要运动到某个特定位置，根据您的程序中的输入，使用POSx_GOTO指令。
- 要运行您在位控向导中所组态的移动包络，使用POSx_RUN指令。

其它位控指令是可选的。

POSx_CTRL指令

POSx_CTRL指令在S7-200每次转换为RUN模式时自动向位控模块发出命令，装载组态/包络表，从而实现有位控模块的使能和初始化。

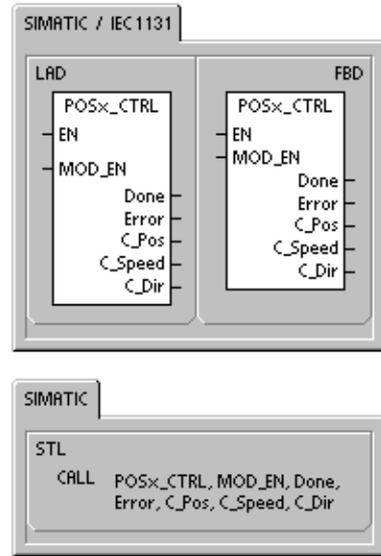
这条指令在您的项目中只使用一次，并且要确保您的用户程序在每一循环中调用该指令。使用SM0.0（常通）作为EN参数的输入。

MOD_EN参数必须为接通状态以确保其它位控指令发送命令给位控模块。如果MOD_EN参数为断开状态，位控模块放弃所有正在进行当中的模块。

POSx_CTRL指令的输出参数提供位控模块当前的状态。

当位控模块完成所有指令后，参数Done接通。

参数Error包含指令的执行结果。表9-20所示为错误代码定义。



参数C-Pos是模块的当前位置。基于测量的单位，该值可以是一个脉冲数（双整数）或者工程单位数（实数）。

参数C-Speed提供模块的当前速度。如果您组态模块的测量系统是脉冲，C-Speed是一个每秒脉冲数的长整数。如果您组态测量系统工程单位，C_Speed是一个每秒若干个所选工程单位数的实数。

参数C_Dir指示电机的当前方向。

表9-9 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、能流
Done、C_Dir	BOOL	I、Q、V、M、SM、S、T、C、L、常数
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

位控模块只在上电时或接到装载组态的命令时读取组态/包络表。

- 当使用位控向导修改组态时，POSx_CTRL指令自动命令位控模块在S7-200 CPU转为RUN模式时装载组态/包络表。
- 如果使用EM253控制面板修改组态，点击刷新组态按钮命令位控模块装载新的组态/包络表。
- 如果您用其它方式修改了组态，那么您必须向位控模块发出一条重新装载组态的命令使它装载组态/包络表。否则，位控模块继续使用旧的组态/包络表。

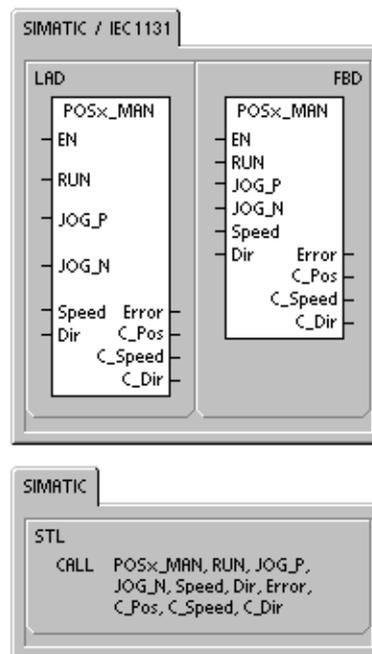
POSx_MAN指令

POSx_MAN指令（手动模式）将位控模块置于手动模式。这种模式下，电机可以以不同的速度沿正向或反向点动。当POSx_MAN指令使能时，只能运行POSx_CTRL和POSx_DIS指令。

RUN、JOG_P或JOG_N的输入您只能同时使能一个。

使能RUN（RUN/Stop）参数则命令位控模块按指定方向（参数Dir）加速到指定速度（参数Speed）。您可以在电机运行时改变速度值，但参数Dir必须保持恒定。禁止参数RUN则命令位控模块减速至电机停止。

使能参数JOG_P（点动正转）或JOG_N（点动反转）命令位控模块沿正向或反向点动。如果JOG_P或JOG_N有效的时间短于0.5秒，位控模块则输出脉冲运动到JOG_INCREMENT所指定的距离。如果JOG_P或JOG_N的有效时间等于或长于0.5秒，位控模块则开始加速到JOG_SPEED所指定的速度。



参数Speed决定RUN使能时的速度。如果位控模块的测量系统组态为脉冲，该速度是一个每秒若干脉冲数的数值（双整数）。若位控模块的测量系统组态为工程单位，该速度是一个每秒若干单位的实数值。电机运行时可以修改该速度参数。



提示

位控模块也许不会响应速度参数小的变化，特别是对于组态的加速和减速时间较短且组态的最大速度和启动/停止速度相差较大的情况。

参数Dir决定RUN使能时的运动方向。当RUN使能时，不能修改该方向参数。

参数Error包含指令的执行结果。表9-20所示为错误代码的定义。

参数C-Pos包含模块的当前位置。基于所选的测量单位，该值可以是一个脉冲数（双整数）或工程单位数（实数）。

参数C_Speed包含模块的当前速度。基于所选的测量单位，该值可以是每秒脉冲数（双整数）或是每秒工程单位（实数）。

参数C_Dir指示电机的当前方向。

表9-10 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、能流
Speed	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD、常数
Dir、C_Dir	BOOL	I、Q、V、M、SM、S、T、C、L、常数
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

指令POSx_GOTO命令位控模块走到指定位置。

接通EN位使能该指令。确保EN位始终保持接通直到Done位指示指令完成。

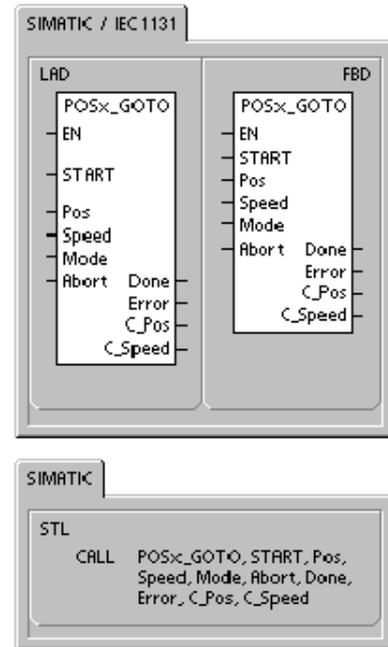
接通参数START向位控模块发送一个GOTO命令。当参数START接通且位控模块不忙时，每一循环都会向位控模块发送一条GOTO命令。要确保只发送一条GOTO命令，使用边沿检测来触发START参数。

参数Pos包含一个表示运动位置（对于绝对运动）或运动距离（对于相对运动）的值。基于所选的测量单位，该值可以是一个脉冲数（双整数）或工程单位数（实数）。

参数Speed决定了运动的最大速度。基于测量单位，该值可以是一个每秒脉冲数（双整数）或是每秒工程单位（实数）。

参数Mode可选择运动类型：

- 0 - 绝对位置
- 1 - 相对位置
- 2 - 单速连续正向转动
- 3 - 单速连续负向转动



当位控模块完成该指令时，参数Done接通。

接通参数Abort，命令位控模块停止当前的包络并减速直至电机停下。

参数Error包含指令的执行结果。参看表9-20所示为错误代码定义。

参数C_POS包含模块的当前位置。基于测量的单位，该值可以是一个脉冲数（双整数）或者工程单位数（实数）。

参数C_Speed包含模块的当前速度。基于测量的单位，该值可以是一个每秒脉冲数（双整数）或是每秒工程单位（实数）。

表9-11 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、能流
Pos、Speed	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD、常数
Mode	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD、常数
Abort、Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
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（运行包络）命令位控模块执行存储在组态/包络表中的某个包络的运动操作。

接通EN位使能该指令。确保EN位保持接通直至Done位指示该指令完成。

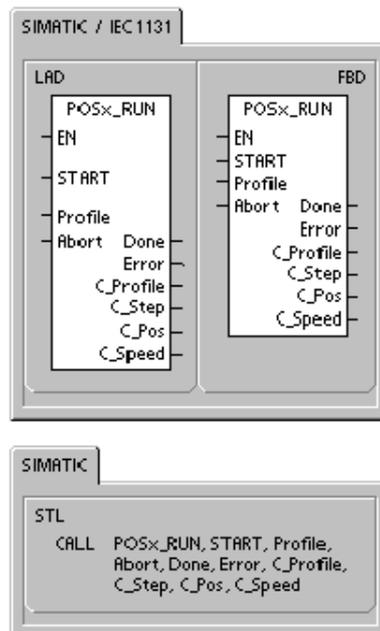
接通参数START发送一个RUN命令给位控模块。每一循环周期，只要START参数接通且位控模块不忙，该指令发送一个RUN命令给位控模块，要保证该命令只发一次，使用边沿检测指令以脉冲触发START参数接通。

包络参数包含该移动包络的号码或符号名。您也可以选择高级运动命令。关于运动命令的信息，请参看表9-26。

接通参数Abort，命令位控模块停止当前的包络并减速直至电机停下。

模块完成该指令时，参数Done接通。

参数Error包含指令的执行结果。参看表9-20所示为错误代码定义。



参数C_Profile包含位控模块当前正在执行的包络。

参数C_Step包含当前正在执行的包络的步。

参数C-POS包含模块的当前位置。基于测量的单位，该值可以是一个脉冲数（双整数）或者工程单位数（实数）。

参数C_Speed包含模块的当前速度。基于测量的单位，该值可以是一个每秒脉冲数（双整数）或是每秒工程单位（实数）。

表9-12 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、能流
Profile	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD、常数
Abort、Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
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

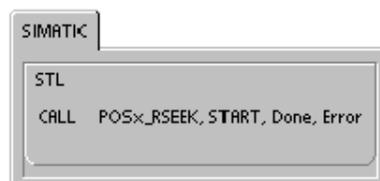
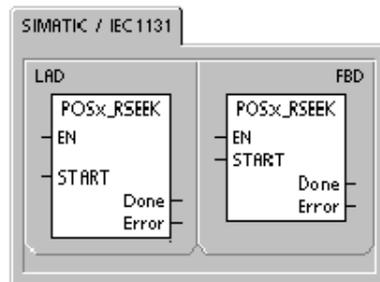
POSx_RSEEK指令

POSx_RSEEK指令（寻找参考点位置）触发一个参考点寻找操作，使用组态/包络表中的搜寻方式。当位控模块锁定参考点并且运动停止后，位控模块装载参数RP_OFFSET的值作为当前位置。

RP_OFFSET的缺省值是0。使用位控向导EM253控制面板或POSx_LDOFF（装载偏移量）指令可以改变RP_OFFSET值。

接通EN位使能该指令。确保EN位保持接通直至Done位指示该指令完成。

接通参数START则向位控模块发送一条RSEEK命令。每一循环周期，当参数START接通且模块不忙，该指令向位控模块发送一条RSEEK指令，要确保该指令只发送一次，使用边沿检测以脉冲触发参数START接通。要保证该命令只发一次，使用边沿检测指令以脉冲触发START参数接通。



模块完成该指令时，参数Done接通。

参数Error包含指令的执行结果。参看表9-20所示为错误代码定义。

表9-13 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、能流
Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

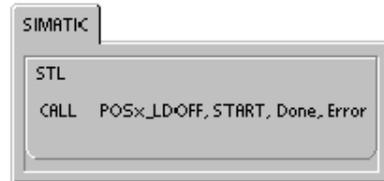
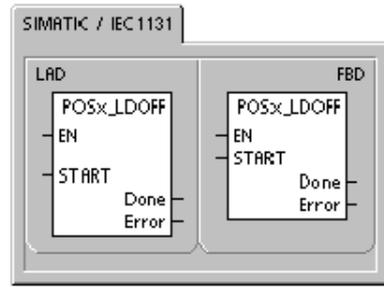
POSx_LDOFF指令

POSx_LDOFF指令（装载参考点偏移量）建立一个新的零位置，它与参考点位置不在同一处。

执行这条指令之前，必须首先决定参考点位置，还要把机器运动到起始位置，当该指令发送LDOFF命令时，位控模块计算起始位置（当前位置）与参考点之间的偏移量。然后，模块将计算所得的偏移量存为参数RP_OFFSET的值并将当前位置设为0。这样就将起始位置设置为零位置。

如果出现故障，电机找不到它的位置了（如，掉电或电机被手动重新定位），可以使用POSx_RSEEK指令自动地重建零位置。

接通EN位使能该指令。确保EN位保持接通直至Done位指示该指令完成。



接通参数START则向位控模块发送一条LDOFF命令。每一循环周期，只要参数START接通且位控模块不忙，该指令向位控模块发送一条LDOFF命令。要保证该命令只发一次，使用边沿检测指令以脉冲触发START参数接通。

模块完成该指令时，参数Done接通。

参数Error包含指令的执行结果。表9-20所示为错误代码定义。

表9-14 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、能流
Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

POSx_LDPOS指令

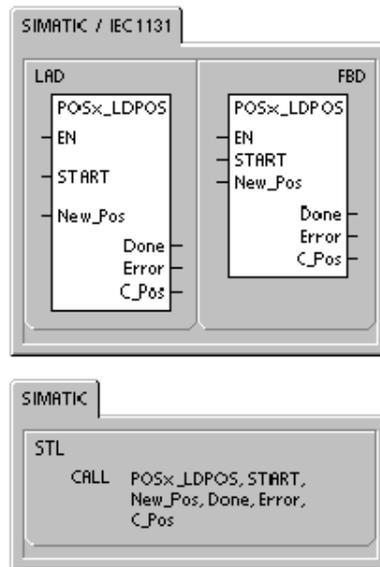
POSx_LDPOS指令（装载位置）改变位控模块的当前位置值。您也可以使用这条指令为绝对运动命令建立一个新的零位置。

接通EN位使能该指令。确保EN位保持接通直至Done位指示该指令完成。

接通参数START则向位控模块发送一条LDPOS命令。每一循环周期，参数START接通且位控模块不忙，该指令向位控模块发送一条LDPOS命令。要保证该命令只发一次，使用边沿检测指令以脉冲触发START参数接通。

参数New_Pos提供一个新值替换位控模块在绝对运动中报告并使用的当前位置值。基于测量单位，该值可以是一个脉冲数（双整数）或是工程单位数（实数）。

模块完成该指令时，参数Done接通。



参数Error包含指令的执行结果。表9-20所示为错误代码定义。

参数C-POS包含模块的当前位置。基于测量的单位，该值可以是一个脉冲数（双整数）或者工程单位数（实数）。

表9-15 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、能流
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、常数
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）命令位控模块改变加速、减速和急停时间。

接通EN位使能该指令。确保EN位保持接通直至Done位指示该指令完成。

接通参数START则将新的时间值拷贝到组态/包络表并向位控模块发送一条SRATE命令。每一循环周期，当START参数接通并且模块不忙时，该指令发送一条SRATE命令到位控模块。要保证该命令只发一次，使用边沿检测指令以脉冲触发START参数接通。

参数ACCEL_Time、DECEL_Time和JERK_Time决定新的加速时间、减速时间和急停时间，单位为毫秒（ms）。

模块完成该指令时，参数Done接通。

参数Error包含指令的执行结果。参看表9-20所示为错误代码定义。

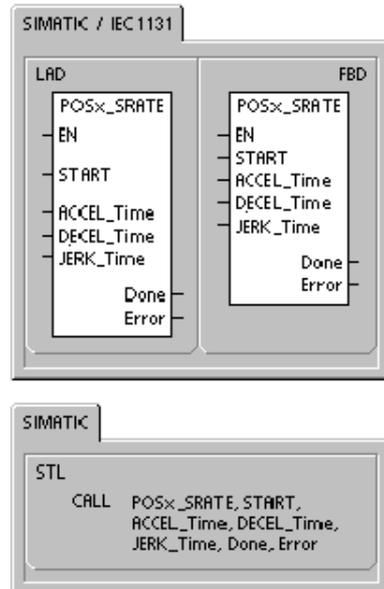


表9-16 POSx_SRATE指令的参数。

输入/输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、常数
ACCEL_Time、 DECEL_Time、JERK_Time	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD、常数
Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

POSx_DIS指令

指令POSx_DIS可接通或断开位控模块的DIS输出。您可以使用DIS输出来使能或禁止电机控制器。如果您要使用位控模块上的DIS输出，那么，这条指令可以在每一循环周期中调用，或者只在您需要改变DIS输出时调用。

EN位接通时使能该指令，参数DIS_ON控制位控模块的DIS输出。关于DIS输出的更多信息，请参看表9-8或参看附录A中的位控模块特性。

参数Error包含指令的执行结果。参看表9-20所示为错误代码定义。

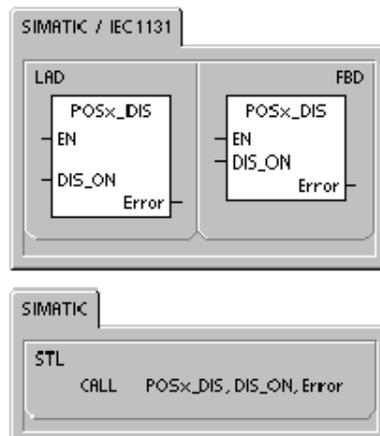


表9-17 POSx_DIS指令的参数。

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、常数
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

POSx_CLR指令

POSx_CLR指令（触发CLR输出）命令位控模块在CLR输出上生成一个50ms的脉冲。

接通EN位使能该指令。确保EN位保持接通直至Done位指示该指令完成。

接通参数START则向位控模块发送一条CLR命令。每一循环周期，当参数START接通并且模块不忙时，该指令向位控模块发送一条CLR命令。要保证该命令只发一次，使用边沿检测指令以脉冲触发START参数接通。

模块完成该指令时，参数Done接通。

参数Error包含指令的执行结果。参看表9-20所示为错误代码定义。

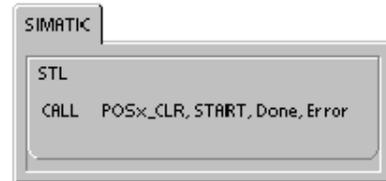
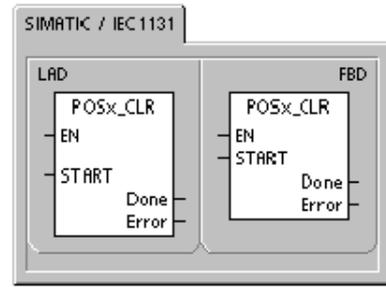


表9-18 POSx_CLR指令的参数。

输入/输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、能流
Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

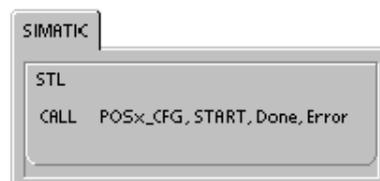
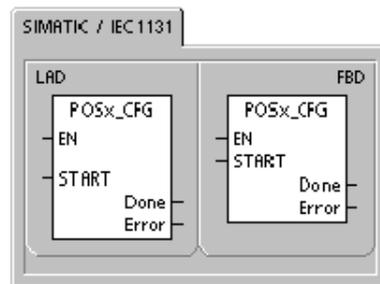
POSx_CFG指令

POSx_CFG指令（重新装载组态）命令位控模块从组态/包络表指针所指定的地方读取组态块。位控模块将新的组态与现有的组态进行比较并执行所有需要的设置改变或重新计算。

接通EN位使能该指令。确保EN位保持接通直至Done位指示该指令完成。

接通参数START使位控模块发送一条CFG命令。每一循环周期，当参数START接通且模块不忙时，该指令都会向位控模块发送一条CFG命令。要保证该命令只发一次，使用边沿检测指令以脉冲触发START参数接通。

模块完成该指令时，参数Done接通。



参数Error包含指令的执行结果。参看表9-20所示为错误代码定义。

表9-19 POSx_CFG指令的参数。

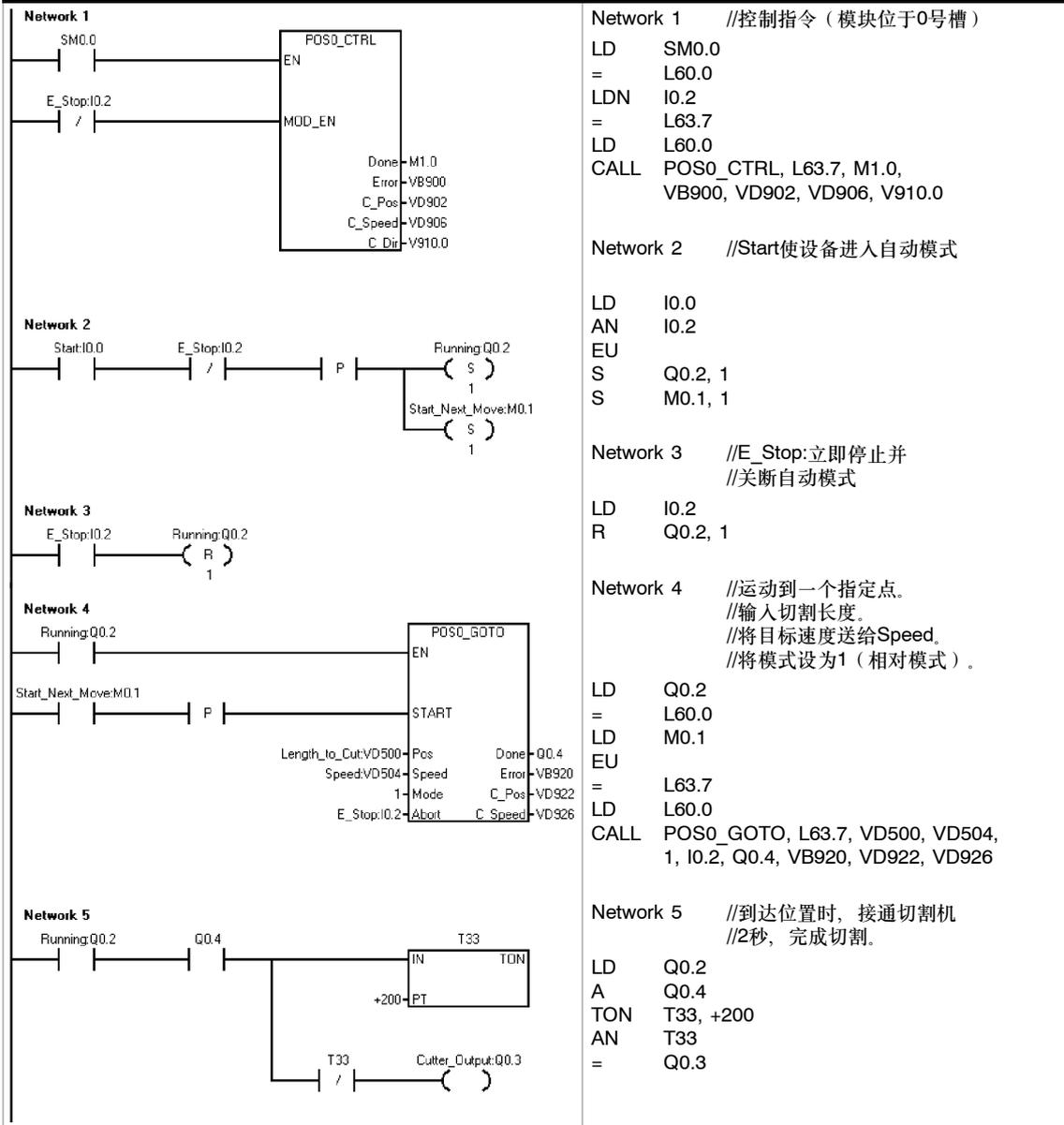
输入/输出	数据类型	操作数
START	BOOL	I、Q、V、M、SM、S、T、C、L、能流
Done	BOOL	I、Q、V、M、SM、S、T、C、L、常数
Error	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD

位控模块的示例程序

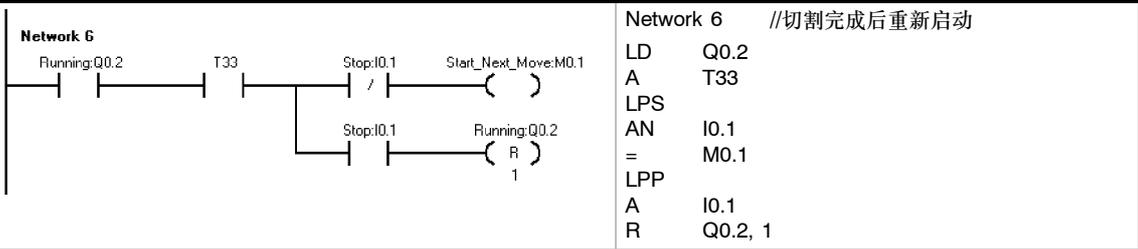
第一个示例程序是一个相对运动的示例，使用了POSx_CTRL和POSx_GOTO指令，完成一个切割长度的操作。该程序不需要RP寻找模式或移动包络，长度可以是脉冲数或工程单位。输入长度(VD500)和目标速度(VD504)，当I0.0(Start)接通时，设备启动。当I0.1(Stop)接通时，设备完成当前操作则停止。当I0.2(E_Stop)接通时，设备终止任何运动并立即停止。

第二个示例程序提供了一个使用POSx_CTRL、POSx_RUN、POSx_RSEEK和POSx_MAN指令的示例。您必须组态RP寻找模式和一个移动包络。

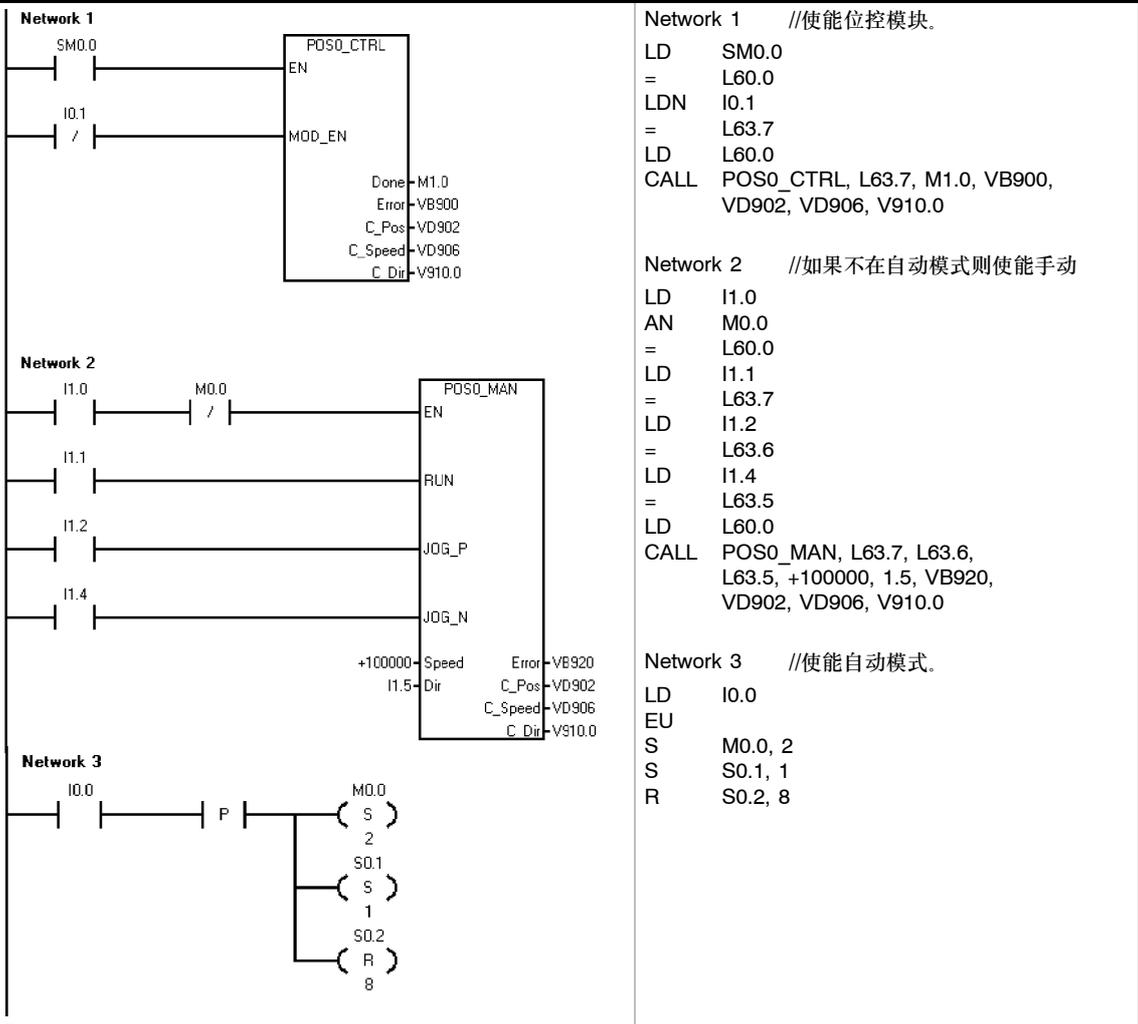
示例程序1: 简单相对运动 (长度切割应用)



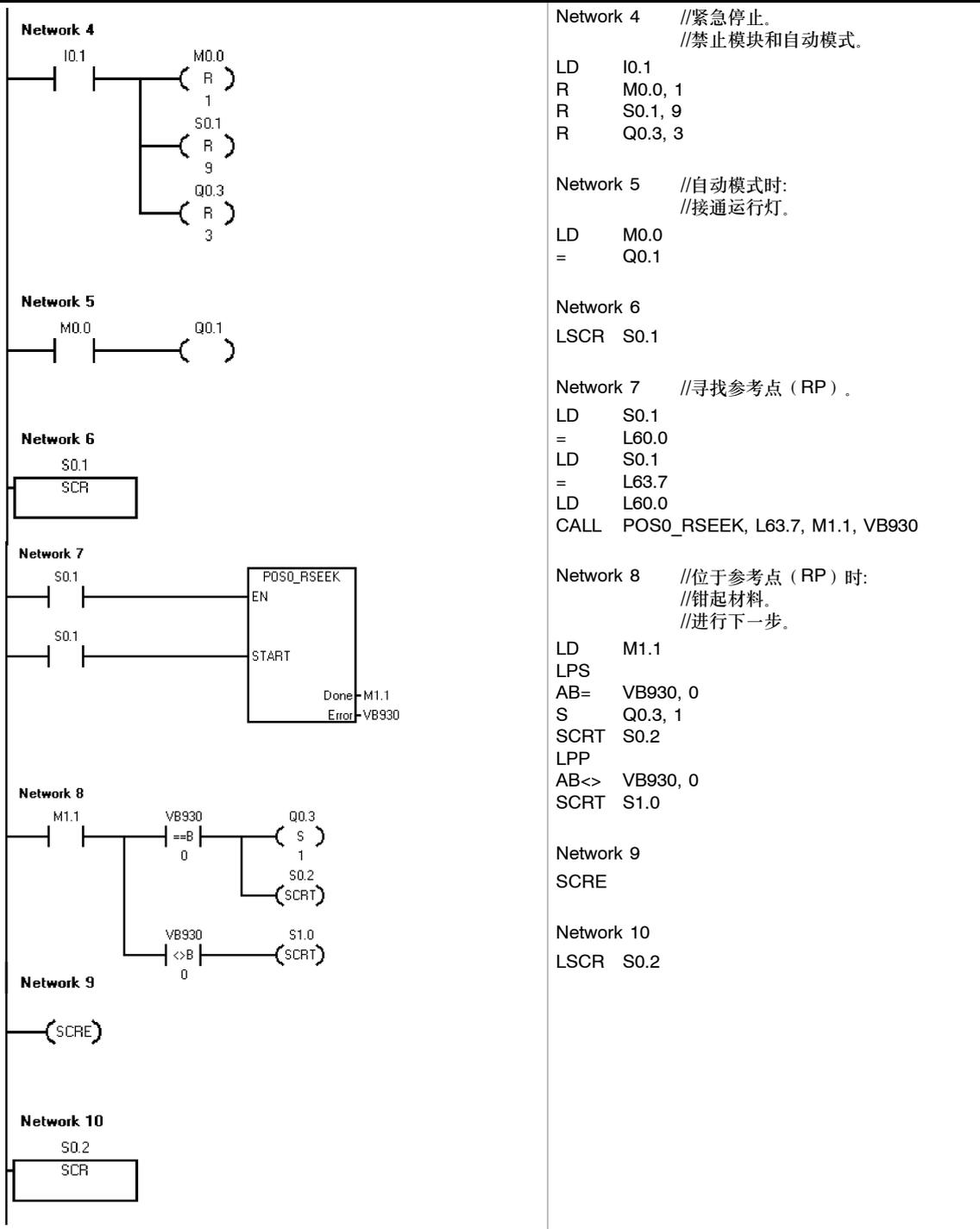
示例程序1: 简单相对运动 (长度切割应用)



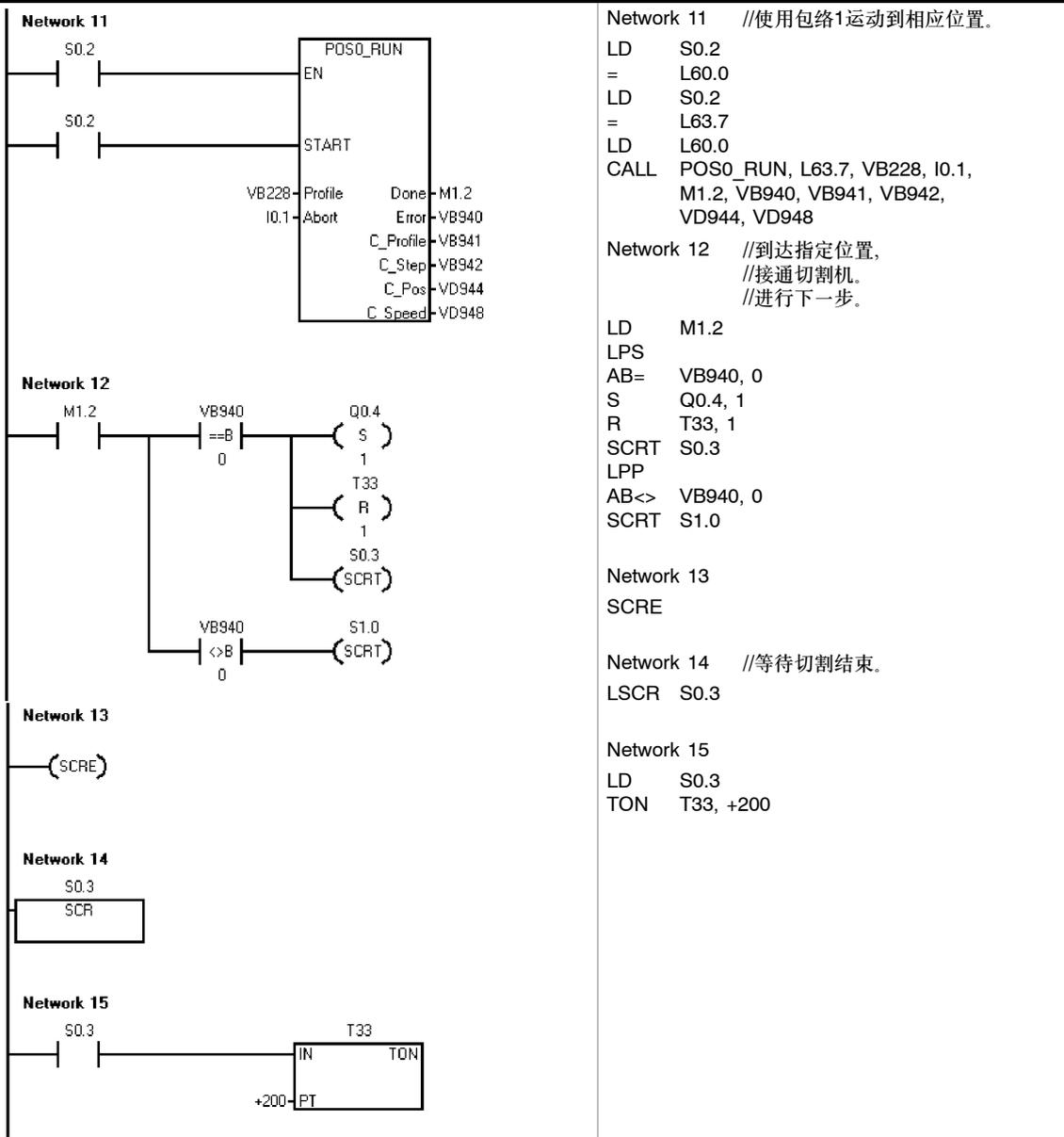
示例程序2: 使用POSx_CTRL、POSx_RUN、POSx_SEEK和POSx_MAN编程



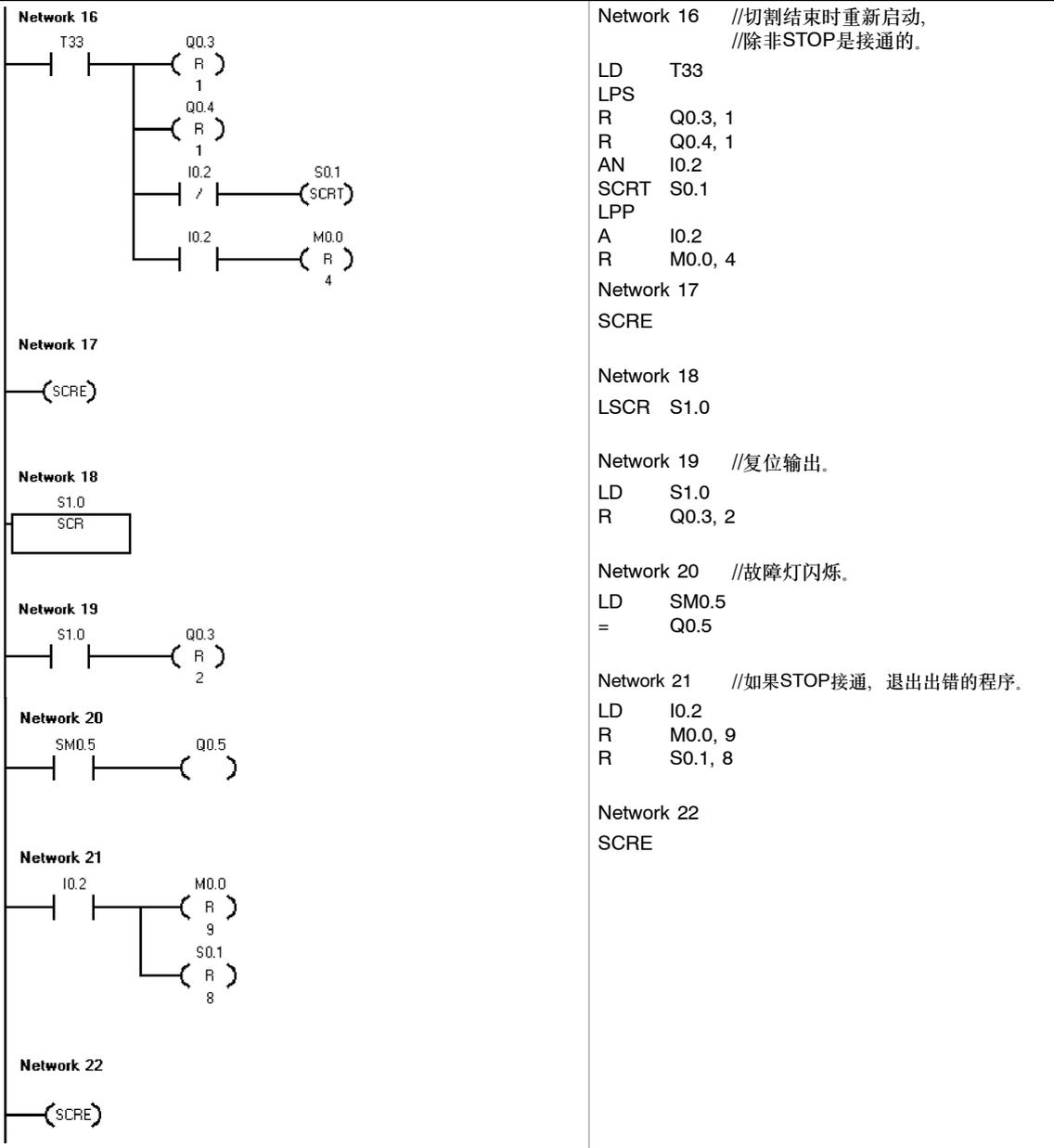
示例程序2: 使用POSx_CTRL、POSx_RUN、POSx_SEEK和POSx_MAN编程



示例程序2: 使用POSx_CTRL、POSx_RUN、POSx_SEEK和POSx_MAN编程



示例程序2: 使用POSx_CTRL、POSx_RUN、POSx_SEEK和POSx_MAN编程



使用EM253控制面板监控位控模块

为了帮助您开发您的运动控制方案，STEP7-Micro/WIN提供EM253控制面板。其中的操作、组态和诊断标签可以帮助您在开发过程的启动和测试阶段轻松监控位控模块的操作。

使用EM253控制面板可以验证位控模块是否正确地接线，可以调整组态数据并测试每个移动包络。

显示并控制位控模块的操作

控制面板的操作标签可以让您干涉位控模块的操作。控制面板显示位控模块的当前速度、当前位置和当前方向，您还可以看到输入和输出LED（脉冲LED除外）的状态。

控制面板可以让您干涉位控模块、改变速度和方向、停止和启动运动、拖动工件（如果运动停止了）。

您还可以生成以下运动命令：

- 使能手动操作。您可以使用该命令手动定位工件。
- 运行移动包络。您可以使用这个命令选择执行某个包络。控制面板显示位控模块正在执行的包络的状态。

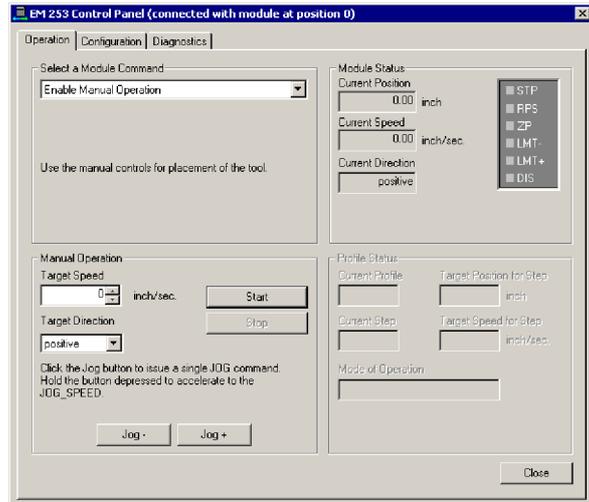


图9-17 EM253控制面板的操作标签

- 寻找参考点。该命令使用所组态的模式寻找参考点。
- 装载参考点偏移量。当您使用手动控制将工件拖动到一个新的零点后，您可以装载参考点偏移量。
- 重新装载当前位置。该命令可刷新当前位置值并建立一个新的零位置。
- 激活和禁止DIS输出。这些命令接通或断开位控模块上的DIS输出。
- 以脉冲触发CLR输出。该命令在位控模块的CLR输出产生一个50ms的脉冲。
- 教授一个移动包络。使用该命令您可以为一个移动包络存储目标位置和速度并记录您手动定位工件时的步。控制面板显示位控模块正在执行的包络的状态。
- 装载模块组态。该命令通过命令位控模块读取S7-200V区的组态块，可以装载一个新的组态。

- 运动到一个绝对位置。使用该命令，您可以以一个目标速度运动到指定位置。在使用该命令前，零位置必须已经建立。
- 以一个相对量运动。使用该命令，您可以以目标速度从当前位置运动到指定距离。您可以输入一个正的或负的距离。
- 复位命令接口。该命令清除位控模块的命令字节并置位Done位。当位控模块对命令没有响应时可以使用该命令。

显示并修改位控模块的组态

使用控制面板的组态标签页，您可以查看和修改存储在S7-200数据块中的位控模块的组态设置。

修改过组态设置后，只需点击一个按钮即可同时刷新STEP7-Micro/Win项目和S7-200数据块中的设置

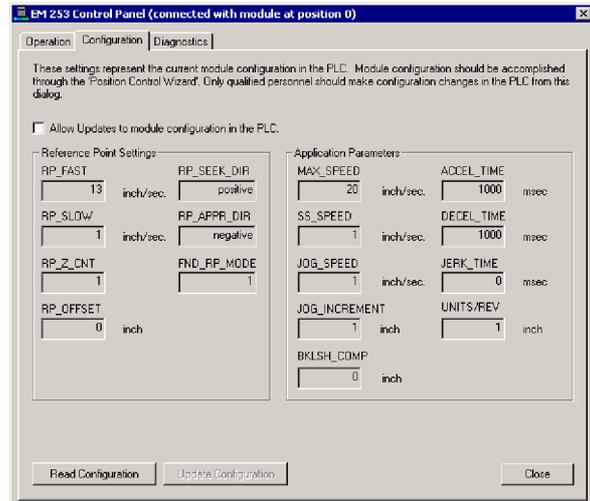


图9-18 EM253控制面板的组态标签

显示位控模块的诊断信息

使用控制面板的诊断标签，您可以查看位控模块的诊断信息。

您可以查看位控模块的特定信息，例如，模块在I/O总线上的位置，模块类型和硬件版本，以及用作该模块的命令字节的输出字节。

控制面板显示所命令的操作引起的错误条件。对于指令错误条件请参看表9-20。

您还可以查看位控模块所报告的所有错误条件。对于位控模块错误条件请参看表9-21。

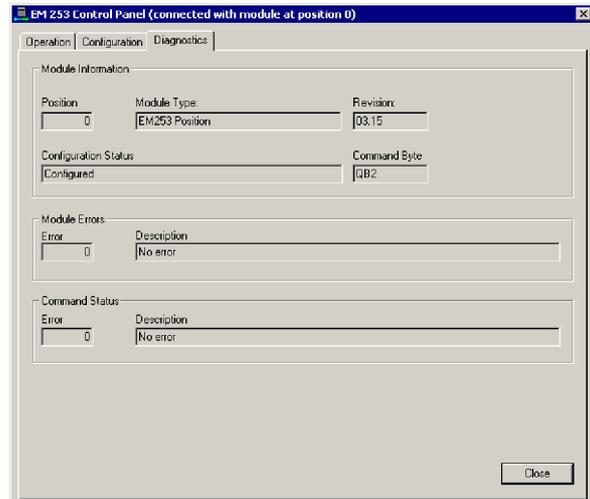


图9-19 EM253控制面板的诊断标签

位控模块和位控指令的错误代码

表9-20 指令错误代码

错误代码	描述
0	无错
1	用户放弃
2	组态错误 使用EM253控制面板的诊断标签查看错误代码
3	非法命令
4	由于没有有效的组态而放弃 使用EM253控制面板的诊断标签查看错误代码
5	由于没有用户电源而放弃
6	由于没有定义的参考点而放弃
7	由于STP输入激活而放弃
8	由于LMT-输入激活而放弃
9	由于LMT+输入激活而放弃
10	由于运动执行的问题而放弃
11	没有为指定包络所组态的包络块
12	非法操作模式
13	该命令不支持的操作模式
14	包络块中非法的步号
15	非法的方向改变
16	非法的距离
17	RPS触发在达到目标速度前出现
18	在RP附近时速度不是RP_Slow
19	速度超出范围
20	没有足够的距离执行所希望的速度改变
21	非法位置
22	零位置未知
23 to 127	保留
128	位控模块不能处理该指令：位控模块可能正在忙于处理另一个指令或该指令无启动脉冲。
129	位控模块错误模块ID不正确或模块已注销。其它错误条件请参看SMB8到SMB21（I/O模块ID和错误寄存器）。
130	位控模块未使能
131	位控模块不能使用由于模块故障或未使能 （参见POSx_CTRL状态）
132	由位控向导组态的Q内存地址与位控模块在该位置处模块的内存地址不一致。

表9-21 模块错误代码

错误代码	描述
0	无错
1	无用户电源
2	没有组态块
3	组态块指针错误
4	组态块的大小超过了可用的V区
5	非法的组态块格式
6	定义了太多的包络
7	非法的STP_RSP定义
8	非法LIM-定义
9	非法LIM+定义
10	非法的FILTER_TIME定义
11	非法的MEAS_SYS定义
12	非法的RP_CFG定义
13	非法的PLS/REV值
14	非法的UNITS/REV值
15	非法的RP_ZP_CNT值
16	非法的JOG_INCREMENT值
17	非法的MAX_SPEED值
18	非法的SS_SPD值
19	非法的RP_FAST值
20	非法的RP_SLOW值
21	非法的JOG_SPEED值
22	非法的ACCEL_TIME值
23	非法的DECEL_TIME值
24	非法的JERK_TIME值
25	非法的BKLSH_COMP值

高级议题

理解组态/包络表

位控向导基于您关于您的运动控制系统所给出的回答可以自动地生成组态和包络信息，帮助您轻松地完成运动程序。组态/包络表信息提供给那些想要创建他们自己的位控程序的高级用户。

组态/包络表位于S7-200的V区。正如表9-22所示，组态设置存储为信息的下列形式：

- 组态块包含用来作模块设置以备执行运动命令所需的信息。
- 这个交互作用的块支持通过用户程序对运动参数作直接设置。
- 每个包络块描述一个将由位控模块执行的预先定义的运动操作。最多可组态25个包络块。



提示

要生成多于25个的移动包络，您可以通过改变存储在组态/包络表指针中的数值变换组态/包络表。

表9-22 组态/包络表

偏移量	名称	功能描述	类型																														
组态块																																	
0	MOD_ID	模块识别域	--																														
5	CB_LEN	以字节为单位的组态块的长度（1字节）	--																														
6	IB_LEN	以字节为单位的交互块的长度（1字节）	--																														
7	PF_LEN	以字节为单位的单个包络的长度（1字节）	--																														
8	STP_LEN	以字节为单位的单步的长度（1字节）	--																														
9	STEPS	每个包络允许的步数（1字节）	--																														
10	PROFILES	从0到25的包络号（1字节）	--																														
11	保留	设为0x0000	--																														
13	IN_OUT-CFG	指定模块输入和输出的使用（1字节） <table border="1" style="margin-left: 20px; margin-top: 10px;"> <thead> <tr> <th colspan="2">MSB</th> <th colspan="6"></th> <th colspan="2">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> <th></th> <th></th> </tr> </thead> <tbody> <tr> <td>P/D</td> <td>POL</td> <td>0</td> <td>0</td> <td>STP</td> <td>RPS</td> <td>LMT-</td> <td>LMT+</td> <td></td> <td></td> </tr> </tbody> </table> P/D 该位指定P0和P1的使用。 正极性（POL=0）： 0 - P0脉冲正转 P1脉冲反转 1 - P0脉冲转动 P1控制转动方向（0-正向，1-反向） 负极性（POL=1）： 0 - P0脉冲正转 P1脉冲反转 1 - P0脉冲转动 P1控制转动方向（0-正向，1-反向） POL 该位为P0和P1选择极性转换。 （0-正极性，1-负极性） STP 该位控制STOP输入的有效等级。 RPS 该位控制RPS输入的有效等级。 LMT- 该位控制反向移动限位输入的有效等级。 LMT+ 该位控制正向移动限位输入的有效等级。 0-有效等级高 1-有效等级低	MSB								LSB		7	6	5	4	3	2	1	0			P/D	POL	0	0	STP	RPS	LMT-	LMT+			--
MSB								LSB																									
7	6	5	4	3	2	1	0																										
P/D	POL	0	0	STP	RPS	LMT-	LMT+																										

表9-22 组态/包络表

偏移量	名称	功能描述	类型																				
14	STP_RSP	定义驱动对STP输入的响应（1字节） 0 无响应，忽略输入条件 1 减速至停止并指示到达限位 2 终止脉冲并指示STP输入 3-255 保留（指定该数值则出错）	--																				
15	LMT-_RSP	定义驱动对反向限位输入的响应（1字节） 0 无响应，忽略输入条件 1 减速至停止并指示到达限位 2 终止脉冲并指示STP输入 3-255 保留（指定该数值则出错）	--																				
16	LMT+_RSP	定义驱动对正向限位输入的响应（1字节） 0 无响应，忽略输入条件 1 减速至停止并指示到达限位 2 终止脉冲并指示STP输入 3-255 保留（指定该数值则出错）	--																				
17	FILTER_TIME	为STP、LMT-、LMT+和RPS输入（1字节）指定滤波时间 <table border="1" style="margin-left: 20px;"> <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> '0000' 200微秒 '0101' 3200微秒 '0001' 400微秒 '0110' 6400微秒 '0010' 800微秒 '0111' 12800微秒 '0011' 1600微秒 '1000' 无滤波 '0100' 1600微秒 '1001'到'1111' 保留（指定该数值则出错）	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	定义测量系统（1字节） 0 脉冲（速度为每秒脉冲数，位置值为脉冲数）。数值存为DINT。 1 工程单位（速度为每秒单位数，位置值为单位数）。数值存为REAL。 2-255 保留（指定该数值则出错）	--																				
19	--	保留（设为0）	--																				
20	PLS/REV	定义电机每转的脉冲数（4字节） 只有当MEAS_SYS设为1时才有意义	DINT																				
24	UNITS/REV	定义电机每转的工程单位数（4字节） 只有当MEAS_SYS设为1时才有意义	REAL																				
28	UNITS	保留给STEP7-Micro/WIN存储一个定制单位的字符串（4字节）	--																				
32	RP_CFG	指定参考点寻找组态（1字节） <table border="1" style="margin-left: 20px;"> <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="4" style="text-align: center;">RP_ADDR_DIR</td> <td colspan="4" style="text-align: center;">RP_SEEK_DIR</td> <td colspan="2" style="text-align: center;">模式 (MODE)</td> </tr> </table> RP_SEEK_DIR 该位指定参考点寻找的起始方向。（0-正向，1-反向） RP_APPR_DIR 该位指定终止参考点寻找的接近方向。（0-正向，1-反向） MODE 指定参考点寻找模式。 '0000' 禁止参考点寻找。 '0001' 参考点在RPS输入开始有效的点上。 '0010' 参考点在RPS输入有效区域中央 '0011' 参考点在RPS输入有效区域以外。 '0100' 参考点在RPS输入有效区域内。 '0101'到'1111' 保留（若选择该数则出错）	MSB	7	6	5	4	3	2	1	LSB	0	RP_ADDR_DIR				RP_SEEK_DIR				模式 (MODE)		--
MSB	7	6	5	4	3	2	1	LSB	0														
RP_ADDR_DIR				RP_SEEK_DIR				模式 (MODE)															
33	--	保留（设为0）	--																				
34	RP_Z_CNT	用来定义参考点的ZP输入脉冲数（4字节）	DINT																				
38	RP_FAST	RP寻找操作的高速：小于等于MAX_SPD（4字节）	DINT REAL																				

表9-22 组态/包络表

偏移量	名称	功能描述	类型
42	RP_SLOW	RP寻找操作的低速: 小于等于电机能够在瞬间停止的最大速度 (4字节)	DINT REAL
46	SS_SPEED	启动/停止速度 (4字节) 启动速度是电机能够瞬间从停止状态启动以及从运行状态瞬间停下的最大速度。允许低于该速度的操作, 但加速和减速时间除外。	DINT REAL
50	MAX_SPEED	电机的最大操作速度 (4字节)	DINT REAL
54	JOG_SPEED	拖动速度。小于等于MAX_SPEED (4字节)	
58	JOG_INCREMENT	该拖动增量是相应于一个拖动脉冲应运动的距离 (或脉冲数) (4字节)	DINT REAL
62	ACCEL_TIME	从最小速度加速到最大速度所需时间, 单位为毫秒 (4字节)	DINT
66	DECEL_TIME	从最大速度减速至最小速度所需时间, 单位为毫秒 (4字节)	DINT
70	BKLSH_COMP	啮合间隙补偿: 方向转换时用于补偿系统啮合间隙的距离 (4字节)	DINT REAL
74	JERK_TIME	在加速/减速曲线 (S曲线) 的起始和结束两端进行急停补偿的时间。定义为零值则禁止急停补偿。急停时间以毫秒为单位 (4字节)	DINT
交互作用的块			
78	MOVE_CMD	选择操作模式 (1字节) 0 绝对位置 1 相对位置 2 单速连续正向转动 3 单速连续反向转动 4 手动速度控制, 正转 5 手动速度控制, 反转 6 带有触发停止的单速连续正向转动 (RPS输入指示停止) 7 带有触发停止的单速连续反向转动 (RPS输入指示停止) 8-255 保留 (如果指定为该数值则出错)	--
79	--	保留, 设为0	--
80	TARGET_POS	该运动的目标位置 (4字节)	DINT REAL
84	TARGET_SPEED	该运动的目标速度 (4字节)	DINT REAL
88	RP_OFFSET	参考点的绝对位置 (4字节)	DINT REAL
包络块0			
92 (+0)	STEPS	该运动序列中的步数 (1字节)	--
93 (+1)	模式 (MODE)	选择该包络块的操作模式 (1字节) 0 绝对位置 1 相对位置 2 单速连续正转 3 单速连续反转 4 保留 (定义的错误) 5 保留 (定义的错误) 6 带有触发停止的单速连续正转 (RPS选择速度) 7 带有触发停止的单速连续反转 (RPS输入指示停止) 8 两速, 连续正向转动 (RPS选择速度) 9 两速, 连续反向转动 (RPS选择速度) 10-255 保留 (如果指定为该数值则出错)	--

表9-22 组态/包络表

偏移量	名称		功能描述	类型
94 (+2)	0	POS	运动步0要去的位置(4字节)	DINT REAL
98 (+6)		SPEED	运动步0的目标速度(4字节)	DINT REAL
102 (+10)	1	POS	运动步1要去的位置(4字节)	DINT REAL
106 (+14)		SPEED	运动步1的目标速度(4字节)	DINT REAL
110 (+18)	2	POS	运动步2要去的位置(4字节)	DINT REAL
114 (+22)		SPEED	运动步2的目标速度(4字节)	DINT REAL
118 (+26)	3	POS	运动步3要去的位置(4字节)	DINT REAL
122 (+30)		SPEED	运动步3的目标速度(4字节)	DINT REAL
包络块1				
126 (+34)	STEPS		该运动序列中的步数(1字节)	--
127 (+35)	模式(MODE)		为该包络块选择操作模式(1字节)	--
128 (+36)	0	POS	运动步0要去的位置(4字节)	DINT REAL
132 (+40)		SPEED	运动步0的目标速度(4字节)	DINT REAL
...

控模块的特殊存储区分配

S7-200按照位控模块在I/O系统中的物理位置给每个智能模块分配50个字节的特殊存储区（SM）参见表9-23。当模块检测到错误条件或数据状态的变化，该模块会刷新这些SM的区域。当要求报告错误条件时，第一个模块刷新SMB200到SMB249，第二个模块刷新SMB250到SMB299，以此类推。

表9-23 特殊存储字节SMB200至SMB549

一个智能模块在以下各槽的SM字节:						
槽0	槽1	槽2	槽3	槽4	槽5	槽6
SMB200至 SMB249	SMB250至 SMB299	SMB300至 SMB349	SMB350至 SMB399	SMB400至 SMB449	SMB450至 SMB499	SMB500至 SMB549

表9-24所示是一个智能模块SM数据区分配的结构。这个定义是按照智能模块位于I/O系统的0号槽给出的。

表9-24 用于EM253位控模块的特殊存储区定义

SM地址	描述																											
SMB200至SMB215	模块名称（16个ASCII字符）SMB200是第一个字符：“EM253位置”																											
SMB216至SMB219	软件版本号（4个ASCII字符）。SMB216是第一个字符。																											
SMW220	模块错误代码。表9-21所示为错误代码定义。																											
SMB222	<p>输入/输出状态。反映模块的输入和输出状态。</p> <table border="1" style="margin-left: 20px;"> <tr> <td colspan="7" style="text-align: left;">MSB</td> <td colspan="2" style="text-align: right;">LSB</td> </tr> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td> </tr> <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><td></td> </tr> </table> <p>DIS 禁止输出 0=无电流 1=有电流 STP 停止输入 0=无电流 1=有电流 LMT- 反向限位输入 0=无电流 1=有电流 LMT+ 正向限位输入 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>瞬间模块状态。反映模块的组态状态和转向状态。</p> <table border="1" style="margin-left: 20px;"> <tr> <td colspan="7" style="text-align: left;">MSB</td> <td colspan="2" style="text-align: right;">LSB</td> </tr> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td> </tr> <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><td></td> </tr> </table> <p>OR 目标速度超范围 0=在范围内 1=超出范围 R 转动方向 0=正向 1=反向 CFG 组态的模块 0=未组态 1=已组态</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 是一个字节，它指示当前正在执行的包络																											
SMB225	CUR_STP 是一个字节，它指示包络中当前正在执行的步																											
SMD226	CUR_POS 是一个双字，该值指示模块的当前位置																											
SMD230	CUR_SPD 是双字，该值指示模块的当前速度																											
SMB234	<p>指令的结果。表9-20所示为错误代码的描述。大于127的错误条件由向导创建的指令子程序生成。</p> <table border="1" style="margin-left: 20px;"> <tr> <td colspan="2" style="text-align: left;">MSB</td> <td colspan="2" style="text-align: right;">LSB</td> </tr> <tr> <td>7</td><td>6</td><td colspan="2"></td> </tr> <tr> <td>D</td><td colspan="3">Error</td><td>0</td> </tr> </table> <p>D Done位 0=操作在进行中 1=操作完成（在初始化过程中由模块设置）</p>	MSB		LSB		7	6			D	Error			0														
MSB		LSB																										
7	6																											
D	Error			0																								
SMB235至SMB244	保留																											
SMB245	与该模块用作命令接口的第一个Q字节之间的偏移量。该偏移量由S7-200自动提供以方便用户，而非模板所需。																											
SMD246	指向组态/包络表V区地址的指针。指向V区以外区域的指针值无效。位控模块会一直监视该指针所指的区域直至它收到一个有效的指针值。																											

理解位控模块的命令字节

位控模块提供一个字节的实际输出作为命令字节。图9-20所示为命令字节的定义。表9-20为命令字节定义。

当有数据向命令字节写入使得R由0变1时，模块认为有新的命令写入。

当一个命令正处于激活状态，模块检测到有向闲置状态的转换时（R位状态变为0），则放弃正在进行的操作，并且，如果有运动在进行中，则执行减速停止。

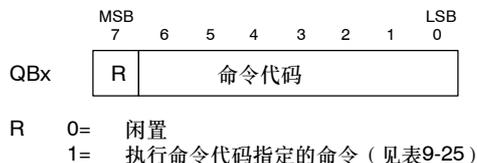


图9-20 命令字节的定义

一个操作完成后，模块必须看到一个向闲置的转换，然后才能接受新的命令。如果某个操作被放弃了，模块必须先完成所有的减速才能接受一个新命令。当有命令激活时，命令代码值的任何改变都忽略不计。

当S7-200出现操作模式的转换或故障条件时，位控模块的响应由S7-200实际输出的结果控制，而S7-200的实际输出的状态在S7-200的功能中已有定义：

- 如果S7-200从STOP变为RUN：S7-200中的程序控制位控模块的操作。
- 如果S7-200从RUN变为STOP：您可以选择这些实际输出在S7-200转为STOP时的状态，或者这些输出保持其最后的状态。
 - 如果变为STOP时R位被关断：位控模块减速所有进行中的运动至停止。
 - 如果变为STOP时R位是接通的：位控模块完成所有进行中的运动。如果没有正在进行的运动，则位控模块执行ID位所指定的包络。
 - 如果R位保持其最后的状态：位控模块完成所有进行中的运动。
- 如果S7-200检测到一个致命错误并关断所有离散输出：位控模块减速所有进行中的运动至停止。

位控模块有了看门狗定时器，如果发现与S7-200的通讯中断，则关断所有输出。如果看门狗时间到，位控模块减速所有进行中的运动至停止。

如果检测到模块的硬件或版本的致命错误，位控模块将输出P0、P1、DIS和CLR设为非激活状态。

表9-25 命令代码定义

命令代码	命令
000 0000到 000 1111	0~24 执行包络0~24中指定的运动操作
100 0000到 111 0101	25到117 保留（指定该数值则出错）
111 0110	118 激活DIS输出
111 0111	119 禁止DIS输出
111 1000	120 触发CLR脉冲输出
111 1001	121 重新装载当前位置
111 1010	122 执行交互块中指定的运动
111 1011	123 获取参考点偏移量
111 1100	124 正向拖动
111 1101	125 反向拖动
111 1110	126 寻找参考点位置
111 1111	127 重新装载组态

表9-26 运动命令

命令	描述
命令0-24: 执行包络块0到24中定义的运动	<p>该命令执行时，位控模块执行包络块中MODE域指定的运动操作，该包络块由命令的命令代码部分指示。</p> <ul style="list-style-type: none"> 在模式0（绝对位置）下，移动包络定义1至4步，每一步都包含对这个运动段进行描述的位置（POS）和速度（SPEED）参数。POS表述的是一个基于参考点位置的绝对位置。运动方向则由当前位置与包络中第一步的位置的相互关系决定。在多步运动中禁止反向运动，并且报告反向运动造成的错误条件。 在模式1（相对位置）下，移动包络1至4步，每一步都包含对这个运动段进行描述的位置（POS）和速度（SPEED）参数。位置值（POS）的符号位决定运动的方向。在多步运动中，禁止反向运动，并且报告反向运动造成的错误条件。 在模式2和3下（单速，连续速度模式），忽略位置（POS）参数，模块加速到第一步中指定的速度SPEED。模式2用于正转，模式3用于反转。当命令字节变为闲置时，运动停止。 在模式6和7下（带有触发停止的单速、连续速度模式），模块加速到第一步中指定的速度SPEED。如果一旦RPS输入激活，运动在完成第一步中POS指定的距离后停止。（在POS域中指定的距离必须包括减速距离。）若当RPS输入有效时，POS域为零，则位控模块减速到停止。模式6用于正转，模式7用于反转。 在模式8和9下，RPS输入的二进制值将选择包络块中前两步所定义的两个连续速度中的一个作为速度值。 <ul style="list-style-type: none"> 如果RPS未激活：步0控制驱动的速度。 如果RPS激活：步1控制驱动的速度。 <p>模式8用于正转，模式9用于反转。SPEED值控制运动速度POS值在该模式下忽略不计。</p>
命令118 激活DIS输出	该命令执行时，位控模块激活DIS输出。
命令119 禁止DIS输出	该命令执行时，位控模块禁止DIS输出。
命令120 触发CLR脉冲输出	该命令执行时，位控模块在CLR输出上产生一个50ms的脉冲。
命令121 重新装载当前位置	该命令执行时，位控模块将交互块的TARGET_POS域中的值置为当前位置值。

表9-26 运动命令

命令	描述
命令122 <i>执行交互块中指定的运动</i>	<p>该命令执行时，位控模块执行交互块MOVE_CMD域中指定的运动操作。</p> <ul style="list-style-type: none"> 在模式0和1（绝对和相对运动模式）下，执行单步运动，其目标速度和位置信息由交互块的TARGET_SPEED和TARGET_POS域提供。 在模式2和3下（单速，连续速度模式），忽略位置参数，位控模块加速到交互块的TARGET_SPEED域指定的速度。当命令字节变为闲置时，运动停止。 在模式4和5（手动速度控制模式）下，忽略位置参数，您的程序将变化的速度值装载到交互块的TARGET_SPEED域中。位控模块会持续地监视该参数区域并在速度值变化时进行合适地响应。
命令123 <i>获取参考点偏移量</i>	<p>该命令执行时，位控模块建立一个不同于参考点位置的零位置。</p> <p>在发出该命令前，您必须确定参考点位置并把工件拖到工作起始位置。接到该命令后，位控模块计算工作起始位置（当前位置）与参考点位置之间的偏移量，并且将计算得到的偏移量写入交互块的RP_OFFSET区域。随后将当前位置设为0，即将工作起始位置设为零位置。</p> <p>当步进电机无法跟踪其当前位置（如掉电或步进电机手动重新定位）时，可以发出寻找参考点命令来自动重建零位置。</p>
命令124 <i>正向拖动</i>	<p>使用该命令可以手动发出使步进电机正向运动的脉冲。</p> <p>若该命令有效时间小于0.5秒，位控模块发出脉冲，移动在JOG_INCREMENT中所指定的距离。</p> <p>若该命令保持0.5秒或更长，运动模块则开始加速到JOG_SPEED所指定的速度值。当检测到转换成闲置状态时，位控模块减速到停止。</p>
命令125 <i>反向拖动</i>	<p>使用该命令可以手动发出使步进电机反向运动的脉冲。</p> <p>若该命令有效时间小于0.5秒，位控模块发出脉冲，运动JOG_INCREMENT所指定的距离。</p> <p>若该命令保持0.5秒或更长，位控模块则开始加速到JOG_SPEED所指定的速度值。当检测到闲置状态的转换时，位控模块减速到停止。</p>
命令126 <i>寻找参考点位置</i>	<p>该命令执行时，位控模块按照指定的寻找模式发出一个参考点寻找操作。当参考点已经被找到并且运动停止时，位控模块将从交互块的RP_OFFSET域中读取的数据装载到当前位置，并在CLR输出产生50微秒的脉冲。</p>
命令127 <i>重新装载组态</i>	<p>该命令执行时，位控模块从SM区的适当区域读取组态/包络表指针，并在组态/包络表指针指定的位置读取组态块。位控模块将刚刚得到的组态数据与现有的模块组态进行比较，并执行任何所需的设置改变或重新计算。放弃所有缓存的包络。</p>

理解位控模块的包络缓存

位控模块在缓存区最多可以存储4个包络的执行数据。当接收一个执行包络的指令时，位控模块检查请求的包络是否存储在缓存区中。若包络的执行数据驻留在缓存中，位控模块立即执行包络。若包络的执行数据没有驻留在缓存中，位控模块从S7-200的组态/包络表中读取包络块信息，并在执行包络前计算包络的执行数据。

命令122（执行交互块中指定的运动）不使用缓存区去存储执行数据，而是一直从S7-200的组态/包络表中读取交互块，并计算运动的执行数据。

重新组态位控模块将删除所有存储在缓存区中的执行数据。

创建您自己的位控指令

位控向导为位控模块的控制操作创建位控指令，但是，您也可以创建您自己的指令。以下STL代码段提供了一个实例，即如何为位控模块创建您自己的控制指令。

该示例使用S7-200的CPU224，位控模块位于插槽0。位控模块在上电时被组态。CMD_STAT是SMB234的符号，CMD是QB2的符号，NEW_CMD是该包络的符号。

示例程序：控制位控模块

```

Network 1    //新的运动命令状态
LSCR      State_0

Network 2    //CMD_STAT是SMB234的一个符号
           //CMD是QB2的一个符号
           //NEW_CMD是包络的符号。
           //
           //1. 清除位控模块的Done位。
           //2. 清除位控模块的命令字节。
           //3. 发布新命令。
           //4. 等待命令被执行。

LD        SM0.0
MOVB     0, CMD_STAT
BIW      0, CMD
BIW      NEW_CMD, CMD
SCRT     State_1

Network 3
SCRE

Network 4    //等待命令完成。
LSCR      State_1

Network 5    //若命令完成后无错误，进入闲置状态。
LDB=     CMD_STAT, 16#80
SCRT     Idle_State

Network 6    //若命令完成后有错误，进入错误处理状态。
LDB>     CMD_STAT, 16#80
SCRT     Error_State

Network 7
SCRE

```

理解位控模块所支持的RP寻找模式

下列各图是每种RP寻找模式的不同选项示意图。

- ❑ 图9-21所示是RP寻找模式1的选项。这种模式将RP定位在RPS输入开始激活，并且靠近工作区的这端。
- ❑ 图9-22所示是RP寻找模式2的选项。这种模式将RP定位在RPS输入有效区域的中央。
- ❑ 图9-23所示是RP寻找模式3的选项。这种模式将RP定位在RPS输入有效区域之外一个指定数量的零脉冲（ZP）处。
- ❑ 图9-24所示是RP寻找模式4的选项。这种模式将RP定位在RPS输入有效区域内一个指定数量的零脉冲（ZP）处。

对于每一种模式都有RP寻找方向和RP接近方向的四种组合。（仅给出了两种组合的示意图。）这些组合决定了RP寻找操作的模式。对每一种组合，又有四种不同的起始点：

每个图中的工作区域都已确定，以便使从参考点到工作区的运动与RP接近方向一样。以这种方式选择工作区域，机械齿轮系统中的所有啮合间隙都能够在参考点找到后，向工作区的第一次运动中去除。

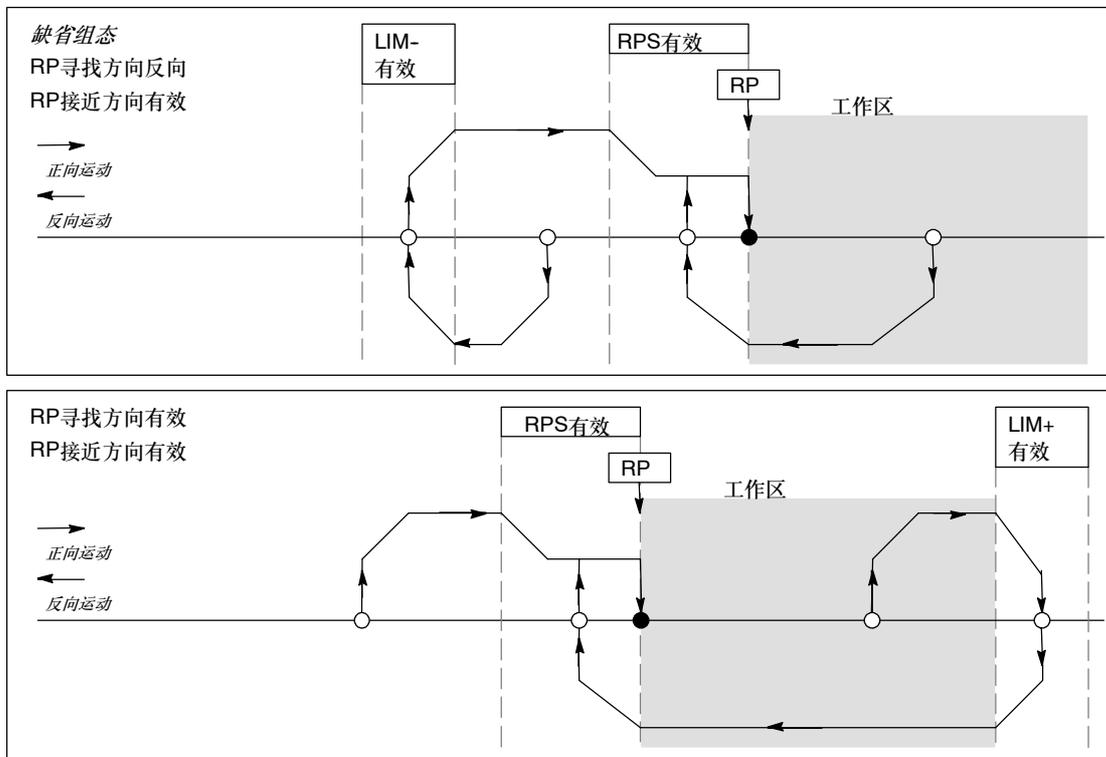


图9-21 RP寻找模式1

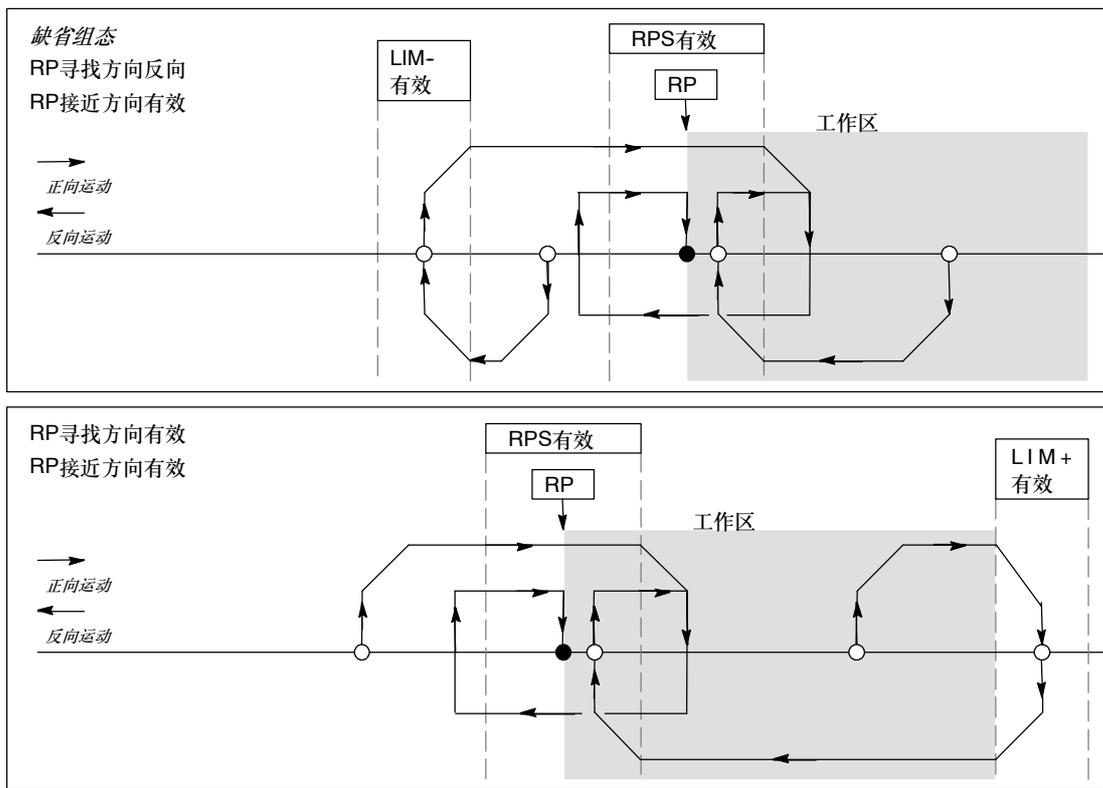


图9-22 RP寻找: 模式2

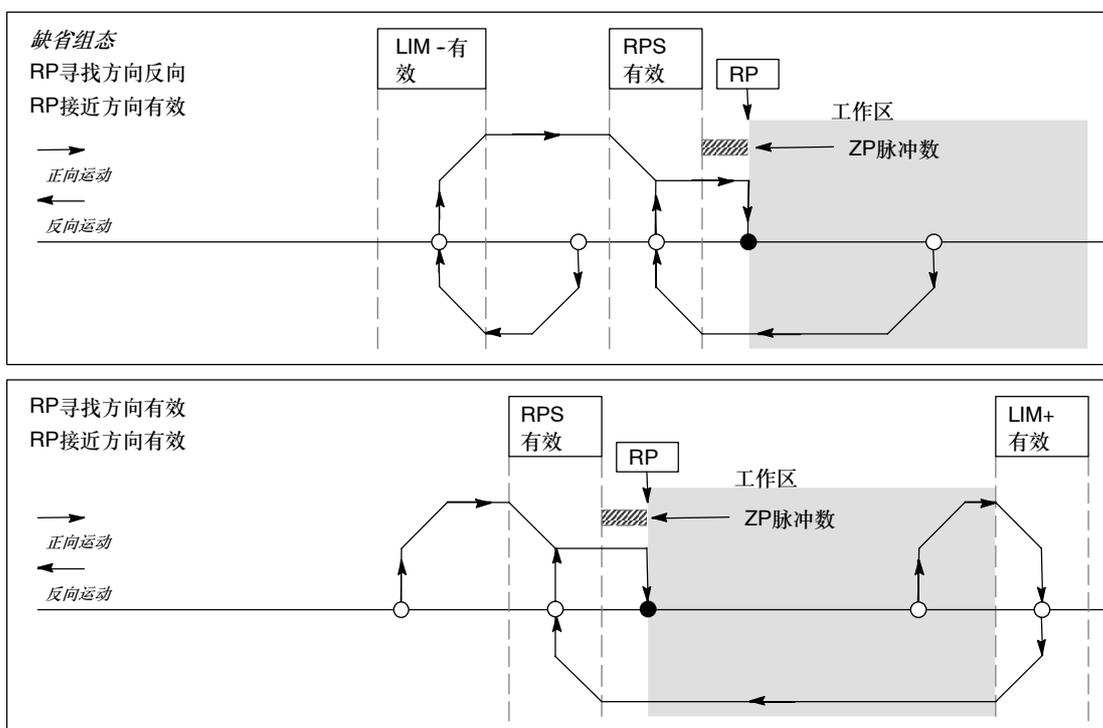


图9-23 RP寻找: 模式3

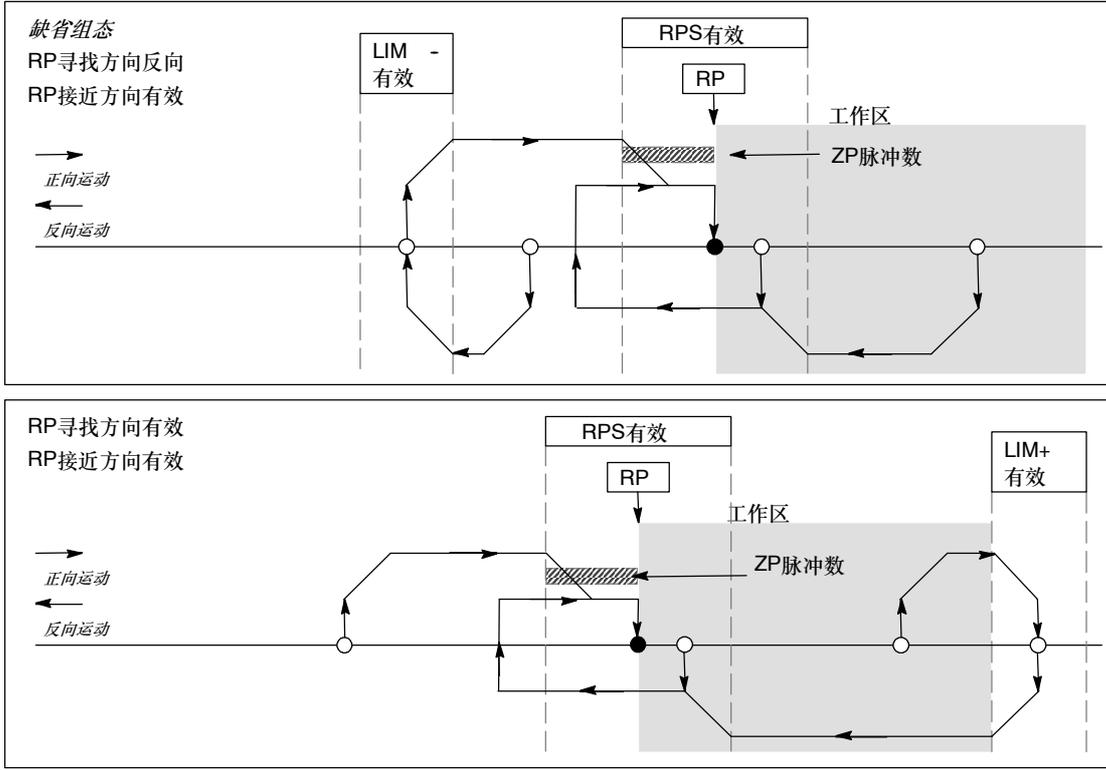


图9-24 RP寻找: 模式4

选择工作区位置消除啮合间隙

图9-25所示为工作区与参考点（RP）、RPS有效区以及限位开关（LIM+和LIM-）之间在接近方向上能够消除啮合间隙的关系。图中的第二部分工作区的位置不能够消除啮合间隙。图9-25所示为RP寻找模式3。其它RP寻找模式的每个搜索顺序也可以有类似的工作区位置，但不推荐。

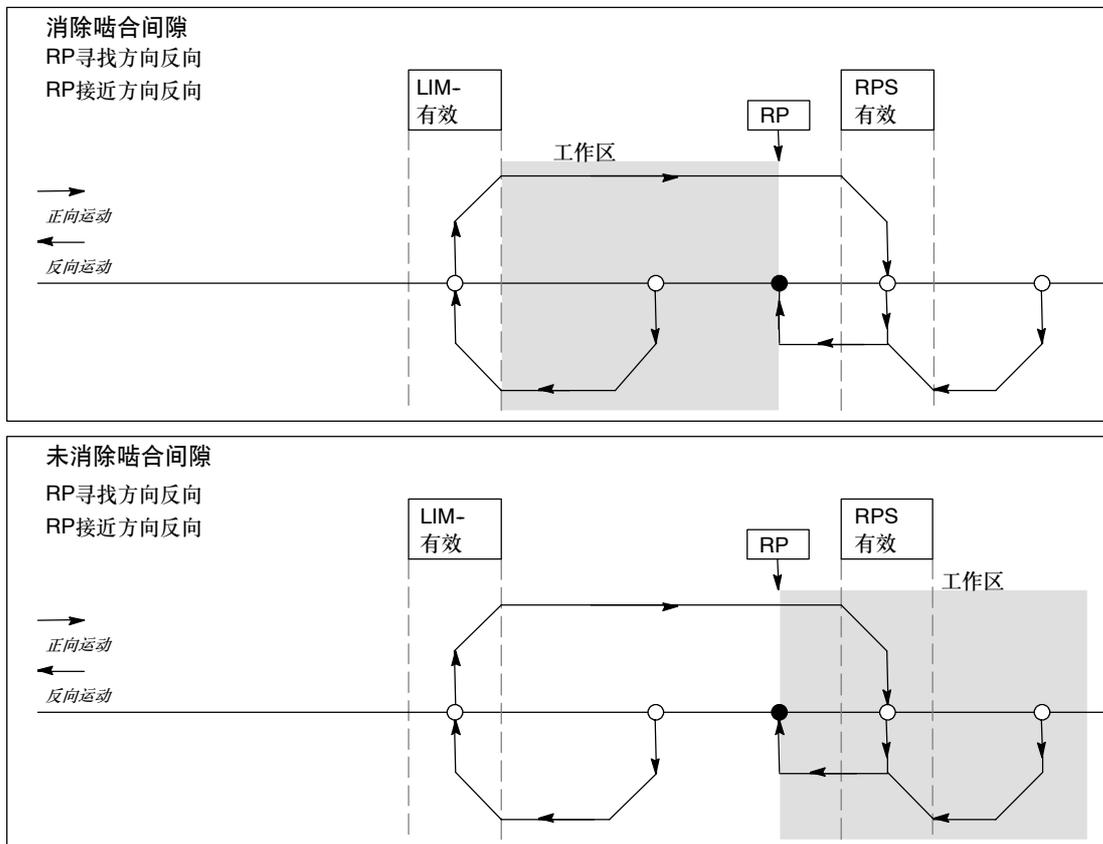


图9-25 消除啮合间隙和未消除啮合间隙的工作区位置

10

创建调制解调模块程序

使用EM241 调制解调模块可以将S7-200直接连到一个模拟电话线上，并且支持S7-200与STEP 7-Micro/WIN的通讯。该调制解调模块还支持Modbus从站RTU协议，该模块与S7-200之间的通讯通过扩展I/O总线实现。

STEP 7-Micro/WIN提供一个调制解调扩展向导，它可以帮助您设置一个远端的调制解调器，或者设置将S7-200连向远端设备的调制解调模块。

本章内容:

EM241 调制解调模块特点	308
利用调制解调扩展向导组态EM241 调制解调模块	314
Modem指令和限定概述	318
EM241 Modem模块指令	319
EM241 Modem模块示例	323
支持智能模块的CPU	323
EM241 MODEM模块的特殊存储区	323
高级议题	325
信息电话号码格式	327
文本信息格式	328
CPU数据传送信息格式	329

EM241调制解调模块特点

使用调制解调模块可将S7-200直接连到模拟电话线上，并且还提供以下特性：

- 提供国际电话线接口
- 提供与STEP 7-Micro/WIN的调制解调接口，可进行编程和诊断(teleservice)
- 支持Modbus RTU协议
- 支持数字和文本的寻呼
- 支持SMS短信息
- 允许CPU到CPU或CPU到Modbus的数据传送
- 密码保护
- 提供安全回拨功能

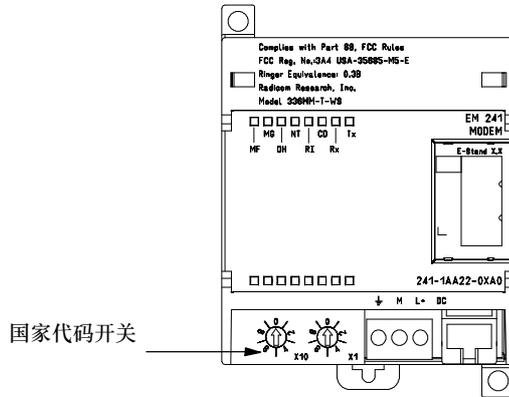


图10-1 EM241调制解调模块

- 调制解调模块的组态存储在CPU中

您可以使用STEP 7-Micro/WIN调制解调扩展向导来组态调制解调模块。对于调制解调模块的规范，请参考附录A。

国际电话线接口

调制解调模块是一个标准的V.34（33.6K波特）、10位调制解调器，它与大多数内置或外置PC调制解调器相兼容。该调制解调模块不能够与11位调制解调器通讯。



图10-2 RJ11插座的视图

可以通过安装在模块前面的6位4线RJ11连接器，将调制解调模块连接到电话线上。见图10-2。

您可以通过调制解调模块前端的六位四线RJ11接口(见图10-2)将模块连到电话线上，在不同的国家可能需要适配器将RJ11接口转换连接到标准电话线上。更多的信息请参见您的适配器的说明书。

调制解调器和电话线接口需要外供直流24V。可以使用CPU传感器电源或外部电源。将调制解调模块的接地端连到系统的地线。

当调制解调模块上电时，模块自动按特定国家的操作组态电话接口。模块前面的两个旋钮开关用来选择国家代码。您必须在模块上电前选好所要的国家。所支持的国家，请参见表10-1。

表10-1 EM241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
06	中国
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

您可以使用调制解调模块通过电话线与STEP 7-Micro/WIN通讯 (teleservice)。当使用STEP 7-Micro/WIN并将调制解调模块用作远程调制解调器时，不需要对S7-200 CPU进行组态或编程。

在STEP 7-Micro/WIN中按照以下步骤使用调制解调模块:

1. S7-200断电，将调制解调模块连至I/O扩展总线上。不要在S7-200 CPU上电时连接任何I/O模块。
2. 将电话线连接到调制解调模块上。如需要可使用适配器。
3. 给调制解调模块连接24V直流供电。
4. 将调制解调模块上的地与系统的地相连接。
5. 设置国家代码开关。
6. 给S7-200 CPU和调制解调模块上电。
7. 组态STEP 7-Micro/WIN与10位调制解调器通讯。

Modbus RTU协议

您可以将调制解调模块组态作为Modbus RTU从站。调制解调模块通过调制解调接口接收Modbus请求，解释这些请求与CPU交换数据。然后调制解调模块生成一个Modbus响应，并通过调制解调接口将此响应发送出去。



提示

如果调制解调模块组态为Modbus RTU从站，STEP 7-Micro/WIN不能通过电话线与调制解调模块通讯。

调制解调模块所支持的Modbus功能如表10-2所示。

Modbus功能4和功能16允许在一个请求中最多读或写125个保持寄存器（250字节的V区）。功能5和功能15可写CPU的映像寄存器。这些值可被用户程序覆盖。

Modbus地址通常写作包括数据类型和偏移量在内的5和6个字符。前一个或二个字符决定数据类型，最后的4个字符在数据类型的范围内选择适当的值。Modbus主设备会将这些地址映射到正确Modbus功能中。

表10-2 调制解调模块支持的Modbus功能

功能	描述
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	预设单个寄存器
功能15	写多个线圈（输出）
功能16	预设多个寄存器

表10-3所示是调制解调模块支持的Modbus地址，以及Modbus地址和S7-200 CPU地址之间的映射关系。

使用调制解调扩展向导可为调制解调模块生成一个支持Modbus RTU协议的组态块。在使用Modbus协议之前，必须将调制解调模块组态块下载到CPU的数据块中。

表10-3 映射Modbus地址至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)

寻呼和SMS短信息

调制解调模块支持向移动电话发送数字和文本的寻呼信息以及SMS（短信息服务）信息（在那些移动电话服务提供商所支持的地方），这些信息和电话号码存储在调制解调模块组态块中，这个组态块必须下载到S7-200 CPU的数据块中。您可以使用调制解调扩展向导为调制解调组态块生成信息和电话号码。调制解调扩展向导还创建允许您的程序去启动信息发送的程序代码。

数字寻呼

数字寻呼使用语音电话的语音功能向寻呼机发送数字值。调制解调模块拨出所请求的寻呼服务，等待语音信息后完成，然后，发送数字寻呼信息。在寻呼信息中可以使用0到9，星号（*），A、B、C和D。由寻呼机显示的星号和字符A、B、C和D的实际字符是不标准的，并且是由寻呼机和寻呼服务提供商所决定的。

文本寻呼

文本寻呼可以将字母信息传送给寻呼提供商，然后再传到寻呼机上，文本寻呼提供商通常使用一条与调制解调器相连的电话线来接收文本寻呼。调制解调模块使用本地电信文字数字协议（TAP）向服务提供商发送文本信息。很多文本寻呼提供商都使用此协议接收信息。

短信息服务（SMS）

一般来说，一些移动电话服务支持那些与GSM相兼容的短信息服务（SMS）信息。SMS允许调制解调模块通过一条模拟电话线向SMS提供商发送信息。SMS提供商再将信息传送到移动电话上，信息则以文本的形式显示在电话上，调制解调模块使用本地电信文字数字协议（TAP）和通用计算机协议（UCP）向SMS提供商发送信息，您可以在一条调制解调线路上，将SMS信息只发送给支持这些协议SMS提供商。

文本和SMS信息中的嵌入变量

调制解调模块能够在文本信息中嵌入来自CPU的数据并且基于在信息中的定义格式化该数据。您可以定义小数点左边和右边数字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模块发送一个文本信息时，调制解调模块从CPU中找到此信息，并确定在这个信息中需要哪些CPU值，从CPU中找到那些值，然后在把信息发送给服务提供商之前将这些值格式化并放到文本信息里面

信息提供商的电话号码、信息以及信息中的嵌入数据是通过多个CPU循环周期从CPU中读取的。当信息发送时，您的程序不能修改电话号码或信息，信息中的嵌入数据值则可以在信息发送过程中连续地被刷新，如果信息中包含多个变量，那些变量是通过多个CPU循环周期读取的，如果您要信息中所有的嵌入数据保持一致，那么，在发送信息时，不要修改任何嵌入变量。

数据传送

调制解调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，传送的数据和电话号码可通过调制解调扩展向导组态，存储在调制解调模块的组态块中。该组态块随后要下载到S7-200 CPU的数据块中。调制解调扩展向导还创建允许您的程序启动数据传送的程序代码。

数据传送可以从远端设备中读数据，也可以是向远端设备写数据，数据传送可以读或写1至100个数据字。数据传送从或者向所连接的CPU中读取或写入数据。

您在调制解调扩展向导中创建的一个数据传送可以包括一个单一的从远端设备的读操作，一个单一的向远端设备的写操作，或者与远端设备间的既读又写的操作。

数据传送使用调制解调模块的组态协议，如果该模块被组态为支持PPI协议（当它响应STEP 7-Micro/WIN时），模块使用PPI协议传送数据。如果该模块被组态为支持Modbus RTU协议，数据传送则使用Modbus协议。

远端设备的电话号码，数据传送请求以及被传送的数据是通过多个CPU循环周期从CPU中读取的。当信息发送时，您的程序不能修改电话号码或信息，而且，在信息传送过程中也不能够修改传送的数据。

如果远端设备也是一个调制解调模块，数据传送中可以使用密码功能，只需在电话号码的组态中输入远端调制解调模块的密码即可。数据传送不能使用回拨功能。

密码保护

密码安全是调制解调模块的可选功能，可在调制解调扩展向导中使能，调制解调模块使用的密码与CPU密码不同。调制解调模块的密码是另外的一个8字符的密码，拨号者必须向调制解调模块提供该密码，否则不允许访问调制解调模块所连接的CPU，密码作为调制解调模块组态块的一部分存在CPU的V区内。必须将调制解调模块组态块下载到连接的CPU的数据块中。

如果CPU的密码安全在系统块中被使能了，拨号者必须提供CPU密码以获取对任何密码保护功能的访问权。

安全回拨

安全回拨是调制解调模块的可选功能，在调制解调扩展向导中组态，回拨功能只允许那些预先设定的电话号码访问CPU，从而为CPU提供了额外的安全保护。回拨功能使能时，调制解调模块应答所有进入的拨号，验证拨号者，然后断开链接。如果呼叫者是有授权的，则调制解调模块就会为呼叫者拨一个预定义的电话号码，并允许呼叫者访问CPU。

调制解调模块支持三种回拨模式：

- 只预先定义一个号码，对此号码回拨。
- 预先设定多个号码，对这些号码皆可回拨。
- 对任何号码都可回拨。

在调制解调扩展向导中选择适当的选项确定回拨模式，然后定义回拨的电话号码。回拨电话号码存储在调制解调模块的组态块中，该组态块存储在所连接的CPU的数据块中。

最简单的回拨形式是一个单个的预先定义的电话号码。如果在调制解调模块的组态块中存储了一个回拨号码，每当调制解调模块应答进入的拨号时，它通知拨号者回拨功能已使能，断开拨号者，然后回拨在组态块中指定的号码。

调制解调模块还支持对多个预设电话号码的回拨功能。在该模式下，向拨号者查询电话号码，如果拨号者提供的号码与模块组态块中预先设定的号码中的某一个相匹配，调制模块则断开拨号者，然后使用与组态块中的号码相匹配的号码进行回拨。用户可最多组态250个回拨号码。

当存在多个预先设定的回拨号码时，向所连接的调制解调模块提供的回拨号码，除了前两个数字外，要与模块的组态块中的号码完全匹配。例如，如果组态的回拨号码是91 (123) 4569999，由于可能需要拨外线 (9) 和长途 (1)，所提供的回拨号码可以是以下号码中的任何一个：

- 91 (123) 4569999
- 1 (123) 4569999
- (123) 4569999

以上所有号码都可以认作是一个匹配的回拨号码。在进行回拨时，调制解调模块使用其组态块中的回拨电话号码，在本例中是91 (123) 4569999。在组态多个回拨号码时，确保所有电话号码除前两位数字以外，是唯一的。进行回拨号码比较时，在一个电话号码中只使用数字字符。逗号或括号这样的字符在回拨号码比较中忽略不计。

调制解调扩展向导中，在回拨组态中选择“Enable callbacks to any phone number”这一选项即可设置成对任意电话号码进行回拨，如果选择这一选项，调制解调模块应答到来的拨号并请求一个回拨号码，在拨号者提供电话号码后，模块断开链接并拨打该电话号码。这种回拨模式只提供一种向调制解调模块一方的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。若使用这种回拨模式，应使用调制解调模块的密码功能提供安全保护。

调制解调模块的密码和回拨功能可同时使能。模块在处理回拨功能之前，要求拨号者提供正确的密码。

EM241组态表

所有的文本信息，电话号码，数据传输信息，回拨号码以及其它选项都存储在调制解调模块的组态表中，该组态表必须下载到CPU的V区。调制解调扩展向导引导您完成整个调制解调模块组态表的生成。然后，STEP 7-Micro/WIN将调制解调模块组态表放到要下载到S7-200 CPU的数据块中。

调制解调模块在CPU运动时以及CPU的任何一次从STOP到RUN转换的5秒钟内读取该组态表。只要调制解调模块与STEP 7-Micro/WIN连接，模块就不会从CPU读取新的组态表。如果调制解调模块在线时有新的组态表下载，该模块会在任务结束时读取新的组态表。

如果调制解调模块检测到组态表中的错误，模块前部指示模块好的LED（MG）会闪烁。查看STEP 7-Micro/WIN中的PLC Information（PLC信息），或读取SMW220（由0号槽的模块占用）中关于组态错误信息的数值。调制解调模块的组态错误如表10-4所列。如果您使用调制解调扩展向导来创建调制解调模块的组态表，STEP 7-Micro/WIN在生成组态表前会进行数据检查。

表10-4 EM241组态错误（十六进制）

错误代码	描述
0000	没有错误
0001	无24V直流外供电电源
0002	模块故障
0003	无组态块ID - 位于组态表起始处的EM241标识对于该模块无效。
0004	组态块超范围 - 组态表指针未指向V区，或者表的一部分超出了所连的CPU的V区范围
0005	组态错误 - 回拨功能使能并且回拨的电话号码数等于0或大于255。信息数大于250。信息发送的电话号码数大于250，或者信息发送的电话号码的长度大于120字节。
0006	国家代码选择错误 - 不支持国家选择旋钮开关上设定的数值
0007	电话号码太大 - 回拨功能使能并且回拨号码长度大于上限
0008-00FF	保留
01xx	回拨号码xx有错 - 在回拨电话号码xx中有非法字符。xx值为1意味着第一个回拨号码，xx值为2是指第二个回拨号码，以此类推。
02xx	电话号码xx有错 - 在信息电话号码xx或数据传送的电话号码xx的某个区域中包含一个非法值。数值xx为1表示是第一个电话号码，2则指第二个号码等等。
03xx	信息xx有错 - 信息或数据传输的数量xx超过允许的最大长度，数值xx为1代表第一条信息，2指第二条信息等等
0400-FFFF	保留

EM241状态LED

位控模块的前面板有8个指示状态的LED。表10-5描述了状态LED。

表10-5 EM241状态LED

LED	描述
MF	模块故障 - 当模块检测到以下故障条件时，该LED灯亮： <ul style="list-style-type: none"> • 无24V直流外供电电源 • I/O看门狗超时 • 模块故障 • 与本地CPU通讯出错
MG	模块正常 - 当没有模块故障条件时该LED灯亮。如果组态表中有错或者用户为电话线接口设置了非法的国家代码设置，模块正常的LED会闪烁。可查看STEP 7-Micro/WIN的PLC Information (PLC信息)画面，或者读取SMW220（用于位于0号槽的模块）的数值以获得组态错误的信息
OH	Off Hook - 当EM241正在使用电话线时该LED亮
NT	无拨号音 - 当EM241接到发送信息的命令而电话线上无拨号音时，该LED灯亮指示有错误条件。只有当EM241被组态为拨号前检查拨号音的时候才会出现这个错误条件。当一次拨号尝试失败后该LED接通并保持5秒左右。
RI	振铃指示 - 该LED指示EM241正在接收一个拨入电话。
CD	载波检测 - 该LED指示与远程调制解调器的连接已建立。
RX	接收数据 - 当调制解调器接收数据时该LED闪烁。
TX	传送数据 - 当调制解调器进行数据传送时，该LED闪烁。

利用调制解调扩展向导组态EM241调制解调模块



Modem扩展

从STEP 7-Micro/WIN工具菜单或浏览器的工具部分启动调制解调扩展向导。

要使用该向导，必须对项目进行编译并设为符号寻址模式。如果您还未编译您的程序，请现在编译。

1. 在Modem扩展向导的第一个画面中，选择组态EM241 Modem模块并点击Next >。
2. 调制解调扩展向导要求调制解调模块相对于S7-200 CPU的位置信息以便生成正确的程序代码。点击读模块按钮可自动读取与CPU相连的智能模块的位置。扩展模块的号码从零开始顺序排列。双击您要组态的Modem模块，或在模块位置域中设置Modem模块的位置，点击Next>。
在硬件版本号1.2之前的S7-200 CPU中，智能模块必须位于紧靠CPU的位置，否则Modem扩展向导无法组态该模块。
3. 在密码保护画面中可以使能Modem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。该密码独立于S7-200 CPU的密码，当该模块设为密码保护后，任何人试图通过该Modem模块来连接S7-200 CPU时都必须输入一个正确的密码，如果需要选择密码功能，并输入一个密码，点击Next >。

4. Modem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。选择什么样的通讯协议取决于远端的通讯对象。该设置控制Modem模块在应答拨入, 以及进行CPU数据传送时所使用的通讯协议。选择适当的协议并点击Next >。
5. 您可以组态该模块向呼机发送数字和文本信息, 或向手机发送短信息(SMS)。点击信息使能多选框并点击组态信息…按钮, 定义信息及接收方的电话号码。
6. 当设置向手机或呼机发送信息时, 您必须定义信息和电话号码。选择组态信息画面的信息标签并点击新信息按钮。输入信息的文字内容并指定插入到信息中的CPU数据。要在信息中插入CPU数据, 将光标放在数据的位置上然后点击插入数据…按钮。指定CPU数据的地址(如: VW100), 显示格式(如: 有符号整数)以及小数点左右的位数。您还可以指定小数点是点号还是逗号。
 - 数据寻呼仅限于数字0-9, 字母A, B, C, D以及星号(*)。数字寻呼信息的最大长度取决于服务提供商。
 - 文本信息最长可有119个字符并且可以是所有的字母数字字符。
 - 文本信息可包含任意数目的嵌入变量。
 - 嵌入变量可以是所连CPU中的V、M、SM、I、Q、S、T、C或AI存储区。
 - 十六进制数的显示以16#开头。数值中的位数取决于变量的大小。例如, VW100显示为16#0123。
 - 小数点左边的位数必须足够大以显示所期望的数值范围, 如果是有符号整数或浮点数, 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, 该整数则显示为标定的整数。例如, 如果VW100 = 1234, 如果小数点右边有两位, 该数据则显示为'12.34'。
 - 如果数值太大无法在指定的域中显示, Modem模块则以#替代所有的字符位置显示该数据。
7. 在组态信息画面选择电话号码标签可组态电话号码。点击新电话号码…按钮, 并添加一个新的电话号码。一旦组态了一个电话号码, 则必须将它添加到项目中。选中已有的电话号码栏中的电话号码并点击向右的箭头即可将这些电话号码添加到当前的项目中。一旦您将这些电话号码添加到当前的项目中, 就可以选择这些电话号码并为这些号码添加符号名以用于您的程序。

基于用户所选择的信息的类型, 电话号码可包括几个不同的域。

- 信息协议的选择告诉Modem模块使用何种协议向信息服务提供商发送信息。数字呼机只支持数字协议。文字寻呼通常要求TAP (Telelocator Alphanumeric Protocol)。SMS信息提供商支持TAP或UCP (通用计算机协议)。通常有三种UCP服务用于SMS信息。多数提供商支持命令1或51。向SMS提供商进行查询以确定他们所要求的协议和命令。
- 在描述域中, 您可以为电话号码添加文字描述。

- 在电话号码域中输入信息服务提供商的电话号码。对于文本信息，这个电话号码是服务提供商用于接收文本信息的Modem线的号码。对于数字寻呼，这个电话号码是呼机本身的号码。Modem模块允许电话号码域中最多40个字符。以下字符可用于电话号码，由Modem模块拨出：

0-9	电话按键
A, B, C, D, *, #	DTMF 数字 (语音拨号)
,	停止拨号2秒钟
!	命令Modem产生一个挂机闪烁
@	等待5秒静音
W	等待拨号音
()	忽略 (可用于格式化电话号码)

- 特定的呼机ID域或手机号码域，这里您要输入接收信息的呼机号码或手机号码。这个号码除了数字0到9不应包含任何字符。最多可输入20个数字。
 - 密码域是TAP信息的可选项。有些提供商要求一个密码，但通常可以空着。Modem模块允许该密码最多可有15个字符。
 - Originating电话号码域使Modem模块在SMS信息中能够被识别。使用UCP命令的服务提供商要求有这样一个域输入。有些服务提供商对这个域可能还有最少字符数的要求。Modem模块最多允许15个字符。
 - Modem标准域的提供是用于Modem模块与服务提供商的modem之间无法进行modem标准沟通的情形。缺省设置是V.34 (33.6k波特)。
 - 数据格式域可用于向服务提供商传送信息时进行数据位和校验的调整。TAP通常使用7位数据位和偶校验，但是有些服务提供商使用8位数据位和无校验。UCP总是8位数据位，无校验。查询服务提供商以决定使用何种设置。
8. 您可以组态Modem模块向另一个S7-200 CPU(如果选择了PPI协议)或一个Modbus设备(如果选择了Modbus协议)传送数据。选中使能CPU数据传送的多选框并点击组态CPU至...按钮，定义要传送的数据以及远端设备的电话号码。
9. 当设置CPU至CPU或CPU至Modbus的数据传送时，您必须定义要传送的数据以及远端设备的电话号码。在组态数据传送画面中选择数据传送标签并点击新传送按钮。数据传送包括从远端设备读数据，向远端设备写数据，或者与远端设备之间的即读又写的操作。如果选择了即读又写，先执行读后执行写。

在每一个读或写操作中最多可写100个字。数据传送必须是对CPU的V存储区。向导在对远端设备的存储区进行描述时总是将远端设备当作S7-200 CPU。如果远端设备是一个Modbus设备，数据传送到Modbus设备 (地址04xxxx) 的保持寄存器。按如下结果确定相应的Modbus地址 (xxxx)：

$$\begin{aligned} \text{Modbus地址} &= 1 + (\text{V区地址}/2) \\ \text{V区地址} &= (\text{Modbus地址} - 1) * 2 \end{aligned}$$

10. 在组态CPU数据传送画面的电话号码标签中，您可以定义用于CPU到CPU或CUP到Modbus数据传送的电话号码。点击新电话号码…按钮，并添加一个新的电话号码。一旦组态了一个电话号码，则必须将它添加到项目中。选中已有的电话号码栏中的电话号码并点击向右的箭头即可将这些电话号码添加到当前的项目中。一旦您将号码加入到项目中，就可以为它增加在程序中使用的符号名。

对描述域和电话号码域的描述与前面信息功能相同。如果远程设备是一个Modem模块并且使能了密码保护功能则要求填写密码域。本地Modem模块的该密码域中必须填写远程Modem模块的密码。本地Modem模块在远程Modem模块有要求时，提供这个密码。

11. 回拨功能使得Modem模块在收到一个来自远程STEP 7-Micro/WIN的拨号后自动断开连接并拨出一个预定的电话号码。选择“使能回拨”复选框然后点击“组态回拨…”按钮组态回拨的电话号码。点击Next >。
12. 在“组态回拨…”画面中，您可以电话号码，Modem模块在应答拨入时会使用这些号码。如果回拨号码是预先确定的，可选择“使能回拨到指定的电话号码”。如果Modem模块接受拨号者提供的任何回拨号码（为反向计费），选择“使能对任意电话号码的回拨”。

如果只对特定的电话号码回拨，点击“新电话号码”按钮添加回拨电话号码。回拨属性画面中可以输入预先定义的回拨号码以及相应的描述。这里输入的电话号码是Modem模块在执行回拨时使用的号码。这个电话号码应包括所需要的所有数字，如连接到一个外线，等待外线时暂停，接通长途等。

在输入一个新的回拨号码后，必须将它加入到项目中。选中可选回拨电话号码，点击向右的箭头将这个号码加入到当前项目。

13. 您可以为Modem模块的信息传送或数据传送功能设置试拨次数。当所有拨号或信息发送的尝试都失败以后，Modem模块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，当Modem模块命令发信息执行回拨时，若没有拨号音，模块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，可选择“使能无拨号音拨号”的选项。

14. Modem扩展向导为Modem模块创建组态块并要求用户输入一个起始存储区地址用来存储Modem模块的组态数据。Modem模块的组态块存储在CPU的V存储区。STEP 7-Micro/WIN将该组态块写入项目的数据块中，组态块的大小则基于所组态的信息和电话号码。您可以选择您所希望的组态块存储的V区地址；或者点击“建议地址”按钮，让向导建议一个大小合适的未使用的V区。点击Next >。
15. 组态Modem模块的最后一步是为Modem模块指定命令字节的Q区地址。您可以计算S7-200上Modem模块之前的所有模块占用的实际的输出字节，从而决定该Q存储区地址。点击Next>。
16. 现在，Modem扩展向导为所选的组态生成项目组件（程序块和数据块），您可以在程序中使用这些程序代码。最后的向导屏幕显示您所要求的组态项目组件。您必须将这个Modem模块组态块（数据块）以及程序块下载到S7-200 CPU。

Modem指令和限定概述

Modem扩展向导可以基于模块位置和您所作的组态选项生成唯一的指令子程序，从而使Modem模块的控制变得非常简单。每个指令都加一个前缀“MODx_”，这里x是模块的位置。

使用EM241 Modem模块指令的要求

在使用Modem模块指令时请考虑以下要求：

- Modem模块指令使用三个子程序。
- Modem模块指令会增加您的程序对存储空间的需求，最多可达370字节。如果您删掉了一个无用的指令子程序，您可以在需要时重新运行Modem扩展向导再次生成这个指令。
- 必须确保在同一时间只有一条指令是激活的。
- 这些指令不能用在中断程序中。
- Modem模块在它第一次上电以及从STOP到RUN时读取组态表信息。除非有模式转换或再次上电，否则模块无法察觉程序中对组态表所作的任何改变。

使用EM241 Modem模块指令

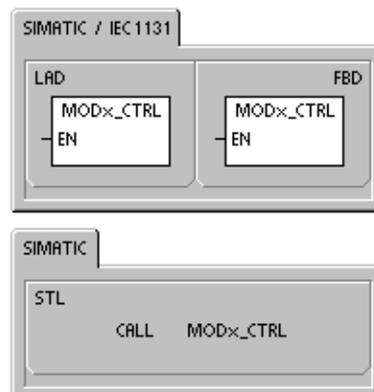
在您的S7-200程序中使用Modem模块指令请遵循以下步骤：

1. 使用Modem扩展向导生成Modem模块组态表。
2. 在程序中插入MODx_CTRL指令并以SM0.0为条件使之每一循环都执行。
3. 为每一个需要发送的信息插入一个MODx_MSG指令。
4. 为每一个数据传送插入一个MODx_XFR指令。

EM241 Modem模块指令

MODx_CTRL指令

MODx_CTRL（控制）指令用于使能和初始化Modem模块，此指令应该每个扫描周期调用一次，并且在项目中只使用一次。



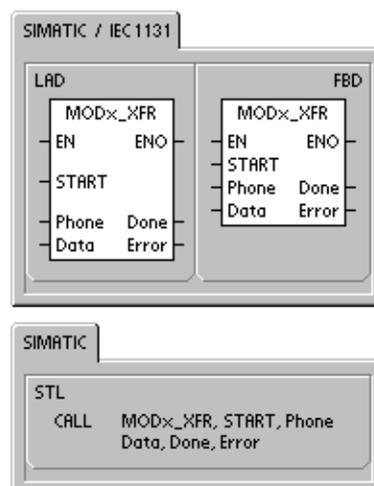
MODx_XFR指令

MODx_XFR（数据传送）指令用于命令Modem模块读写另一个S7-200 CPU或Modbus设备的数据。从START输入被触发的时刻开始到Done位被置位，此指令需要20到30秒的时间。

EN位必须接通向模板发出命令，并且要保持接通直至Done位置位，即标志整个过程完成。当START输入接通且模块不忙时，每一循环向模块发送一个XFR命令。START输入可以边沿检测脉冲触发，这样命令只发送一次。

Phone是数据传送电话号码中的一个。您可以使用您在Modem扩展向导中为这些数据传送电话号码定义的符号名。

Data是一个定义了的数据传送的号码。您可以使用在Modem扩展向导中为这个数据传送请求定义的符号名。



Done是一个位，当Modem模块完成数据传送时接通。

Error是一个字节，包含数据传送的结果。表10-4定义了这条指令的执行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、能流
Phone、Data	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、常数、*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

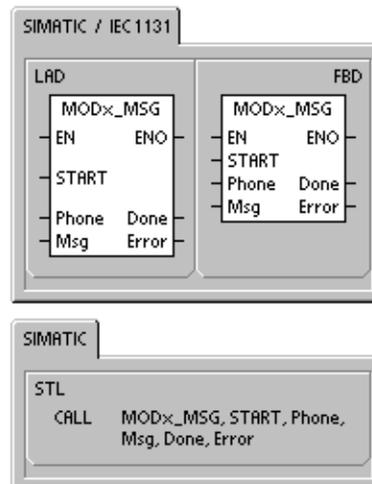
MODx_MSG指令

MODx_MSG（发送信息）指令用于从Modem模块发送寻呼或SMS短信息。从START输入被触发的时刻开始到Done位被置位，此指令需要20到30秒的时间。

EN位必须接通向模板发出命令，并且要保持接通直至Done位置位，即标志整个过程完成。当START输入接通且模块不忙时，每一循环向模块发送一个MSG命令。START输入可以边沿检测脉冲触发，这样命令只发送一次。

Phone是信息电话号码中的一个，您可以使用在Modem扩展向导中定义号码时您为每个信息电话号码分配的符号名。

Msg是一个已定义的信息的号码。您可以使用在Modem扩展向导中为每个信息分配的符号名。



Done是一个位，当Modem模块完成向服务提供商的信息发送后该位置1。

Error是一个字节，包含这个请求执行的结果。表10-8定义了这条指令的执行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、能流
Phone、Data	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、常数、*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

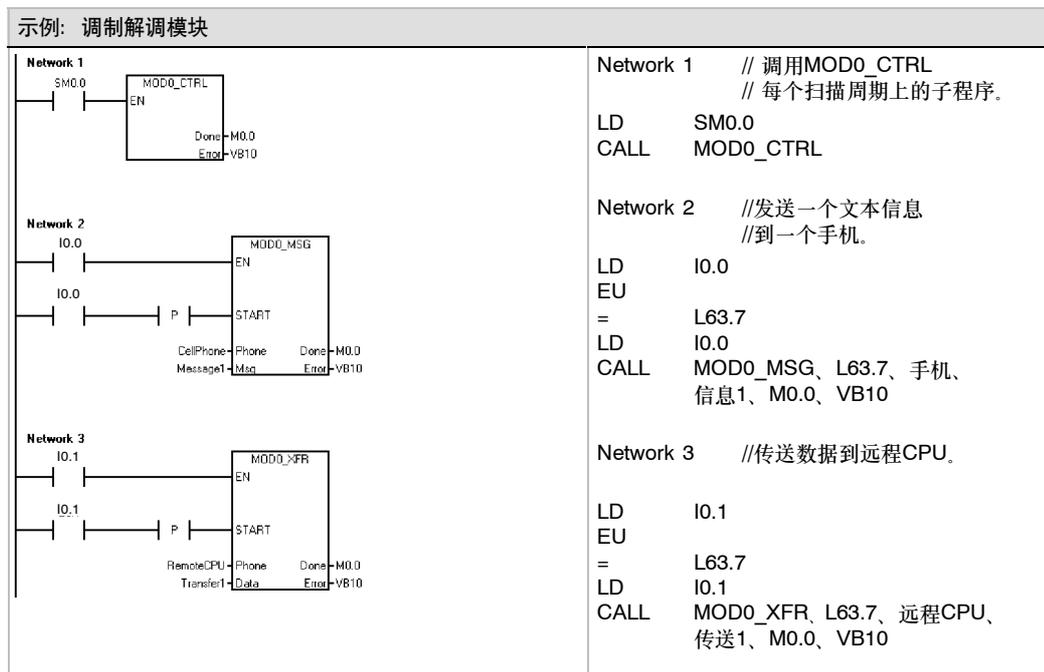
表10-8 MODx_MSG和MODx_XFR指令返回的错误代码

错误代码	描述
0	没有错误
电话线路错误	
1	无拨号音
2	线路忙
3	拨号错误
4	无应答
5	连接超时（1分钟内未有连接）
6	连接放弃或一个未知响应
命令错误	
7	数字寻呼信息包含非法数字
8	电话号码（号码输入）超范围
9	信息或数据传送（Msg或数据输入）超范围
10	文本信息或数据传送信息错误
11	信息或数据传送电话号码错误
12	不允许的操作（如：拨号尝试设为0）
服务提供商错误	
13	信息服务无响应（超时）
14	信息服务由于未知原因断开
15	用户放弃信息（禁止命令位）
TAP - 服务提供商返回的文本寻呼错误和SMS信息错误	
16	接收到远程断开（服务提供商放弃）
17	登录未被信息服务接收（不正确的密码）
18	块未被信息服务接受（校验或传送错误）
19	块未被信息服务接受（未知原因）
UCP - SMS服务提供商返回的SMS信息错误	
20	未知错误
21	检验错误
22	同步错误
23	系统不支持的操作（非法命令）
24	该操作此时不允许
25	拨号排除激活（黑名单）
26	拨号者地址无效
27	验证失败
28	合法代码失败
29	GA无效
30	重复不允许
31	重复合法代码失败
32	优先拨号不允许
33	优先拨号合法代码失败
34	紧急信息不允许
35	紧急信息合法代码失败
36	反向收费不允许
37	反向收费合法代码失败

表10-8 MODx_MSG和MODx_XFR指令返回的错误代码

错误代码	描述
UCP - 服务提供商返回的SMS信息错误 (续)	
38	延期发送不允许
39	新AC无效
40	新的合法代码不允许
41	标准文本无效
42	时间段无效
43	系统不支持的信息类型
44	信息太长
45	请求的标准文本无效
46	信息类型对呼机类型无效
47	SMSC中未发现信息
48	保留
49	保留
50	子机挂机
51	不支持传真组
52	不支持传真信息类型
数据传送错误	
53	信息超时 (远端设备无响应)
54	远端CPU正忙于上载或下载
55	访问错误 (存储器超范围, 非法数据类型)
56	通讯错误 (未知响应)
57	响应的校验错误或CRC错误
58	远端EM241设为回拨方式 (不允许)
59	远端EM241拒绝所提供的密码
60至127	保留
指令使用错误	
128	请求无法处理。模块可能正在处理另一个请求或该请求无启动脉冲。
129	调制解调模块错误: <ul style="list-style-type: none"> 用调制解调扩展向导组态的调制解调模块的位置或Q存储区地址和实际的位置或存储区地址不匹配。 参考SMB8到SMB21 (I/O模块ID和错误寄存器)

EM241 Modem模块示例



支持智能模块的CPU

调制解调模块是被设计用于和表10-9中列出的S7-200 CPU一起使用的智能扩展模块。

表10-9 和S7-200 CPU兼容的EM241模块

CPU	描述
CPU222 Rel.1.10或更高	CPU222 DC/DC/DC和CPU222 AC/DC/继电器
CPU224 Rel.1.10或更高	CPU224 DC/DC/DC和CPU224 AC/DC/继电器
CPU224XP Rel. 2.00或更高	CPU224XP DC/DC/DC和CPU224XP AC/DC/继电器
CPU226Rel.1.00或更高	CPU226 DC/DC/DC和CPU226 AC/DC/继电器

Modem模块的特殊存储区

智能模块按其I/O扩展总线上的物理位置分配有50个字节的特殊存储区（SM）。如果出错或检测到状态变化，模块会刷新与该模块位置相对应的SM区域。如果是第一个模块在需要报告状态和错误信息的时候应刷新SMB200到SMB249。第二个模块则刷新SMB250至SMB299等等，见表10-10。

表10-10 特殊存储字节SMB200至SMB549

特殊存储字节SMB200至SMB549						
智能模块 0号槽	智能模块 1号槽	智能模块 2号槽	智能模块 3号槽	智能模块 4号槽	智能模块 5号槽	智能模块 6号槽
SMB200至 SMB249	SMB250至 SMB299	SMB300至 SMB349	SMB350至 SMB399	SMB400至 SMB449	SMB450至 SMB499	SMB500至 SMB549

表10-11所示为Modem模块的特殊存储区数据区域的分配。该区域是按照智能模块位于I/O系统的0号槽定义的。

表10-11 用于EM241调制解调模块的SM位置

SM地址	描述																																													
SMB200至SMB215	模块名（16个ASCII字符）SMB200是第一个字符。 “EM241调制解调模块”																																													
SMB216至SMB219	S/W修订号码（4个ASCII字符）SMB216是第一个字符。																																													
SMW220	错误代码（SMB220）是错误代码的最高字节 0000 - 没有错误 0001 - 无用户电源 0002 - Modem故障 0003 - 无组态块ID 0004 - 组态块超范围 0005 - 组态错误 0006 - 国家代码选择错误 0007 - 电话号码太大 0008 - 信息太大 0009至00FF - 保留 01xx - 回拨号码XX出错 02xx - 呼机号码XX出错 03xx - 信息号码X 0400至FFFF - 保留																																													
SMB222	模块状态-反映LED状态 <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="text-align: center;">MSB</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">LSB</td> </tr> <tr> <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;">0</td> <td style="text-align: center;">0</td> </tr> <tr> <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> </div> <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- 错误</td> </tr> <tr> <td>G - EM_GOOD</td> <td>0 - 不好</td> <td>1- 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 - 语音拨号</td> <td>1 - 无拨号音</td> </tr> <tr> <td>R - RING</td> <td>0 - 无振铃</td> <td>1 - 电话振铃</td> </tr> <tr> <td>C - CONNECT</td> <td>0 - 未连接</td> <td>1 - 连通</td> </tr> </table>	MSB								LSB	7	6	5	4	3	2	1	0	0	F	G	H	T	R	C	0	0		F - EM_FAULT	0 - 无错误	1- 错误	G - EM_GOOD	0 - 不好	1- 好	H - OFF_HOOK	0 - 挂机,	1 - 摘机	T - NO DIALTONE	0 - 语音拨号	1 - 无拨号音	R - RING	0 - 无振铃	1 - 电话振铃	C - CONNECT	0 - 未连接	1 - 连通
MSB								LSB																																						
7	6	5	4	3	2	1	0	0																																						
F	G	H	T	R	C	0	0																																							
F - EM_FAULT	0 - 无错误	1- 错误																																												
G - EM_GOOD	0 - 不好	1- 好																																												
H - OFF_HOOK	0 - 挂机,	1 - 摘机																																												
T - NO DIALTONE	0 - 语音拨号	1 - 无拨号音																																												
R - RING	0 - 无振铃	1 - 电话振铃																																												
C - CONNECT	0 - 未连接	1 - 连通																																												
SMB223	由开关设置的国家代码（十进制值）																																													
SMW224	建立连接所使用的波特率（无符号十进制值）。																																													
SMB226	用户命令的结果 <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="text-align: center;">MSB</td> <td></td> <td></td> <td></td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td></td> <td style="text-align: center;">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;">错误代码</td> <td></td> </tr> </table> </div> D - Done位; 0 - 操作进行中 1 - 操作完成 错误: 错误代码描述, 见表10-8	MSB				LSB	7	6	5		0	D	0	错误代码																																
MSB				LSB																																										
7	6	5		0																																										
D	0	错误代码																																												
SMB227	电话号码选择 - 该字节指定发送信息时使用的信息电话号码有效值为1至250。																																													
SMB228	信息选择 - 该字节指定发送哪条信息。有效值为1至250。																																													
SMB229至SMB244	保留																																													
SMB245	与该模块用作命令接口的第一个Q字节之间的偏移量。该偏移量由CPU提供, 是为方便用户而非模块所需。																																													
SMD246	在V存储区中指向EM241Modem模块组态表的指针。SMB246是最高字节, SMB249是最低字节。指向V区以外其它区域的指针是不能被接受的, 模块会持续地检查该区域, 等待有效的指针值。																																													

表10-12 EM241组态表

回拨电话号码块 (可选)	
字节偏移量	描述
24	回拨电话号码1 - 一个字符串, 代表第一个授权以EM241Modem模块回拨访问的电话号码。按照回拨电话号码长度域中指定的长度每个回拨号码分配有相同的空间。(组态块中偏移量为6)
24 + 回拨号码	回拨电话号码2
:	:
:	回拨电话号码n
信息电话号码块 (可选)	
字节偏移量	描述
M	信息电话号码1 - 一个字符串, 代表信息电话号码, 包括协议和拨号选项。按照信息电话号码长度域中指定的长度。每个电话号码配有相同的空间。(组态块中的偏移量是7) 对于信息电话号码格式的描述在下面。
M+信息号码长度	信息电话号码2
:	:
:	信息电话号码n
信息块 (可选)	
字节偏移量	描述
N	第一个信息的V区偏移量 (相对于VBO) (2字节)
N+2	信息1的长度
N+3	信息2的长度
:	:
:	信息n的长度
P	信息1 - A一个字符串, 最大 (120字节), 代表第一个信息。该字符串包括文本和嵌入数据的规范, 或者指定一个CPU的数据传送。 请参见下面有关文本信息格式和CPU数据传送格式的描述。
P+信息1的长度	信息2
:	:
:	信息n

Modem模块在以下事件出现时会重新读取组态表:

- S7-200CPU从STOP转为RUN的5秒钟之内 (除非modem在线)
- 每5秒钟一次除非找到一个有效的组态 (除非modem在线)
- 每次modem从在线转为离线。

信息电话号码格式

信息电话号码的结构中包含Modem模块发送信息所需要的信息。信息电话号码是一个字符串，第一个字节是长度，其后是ASCII字符。信息电话号码的最大长度是120个字节（包括长度字节）。

信息电话号码包含由斜杠 (/) 隔开的6个区域。双斜杠之间指示空区域，在Modem模块中空区域改为缺省值。

格式: <电话号码>/<标识>/<密码>/<协议>/<标准>/<格式>

电话号码域是Modem模块发送信息时拨打的电话号码。如果要发送的信息是一个文本或SMS短信息，这个号码是服务提供商的电话号码。如果信息是数字寻呼，该区域是寻呼机的电话号码。如果信息是一个CPU的数据传送，该号码是远端设备的电话号码，该区域的最大字符数是40。

标识: 是寻呼机号或手机号。这个区域只能是数字0至9。如果协议是CPU数据传送，该区域用来提供远端设备的地址。该区域最多可有20个字符。

密码区域: 通过TAP发送信息时，如果服务提供商要求密码，则该区域提供密码。对于通过UCP发送信息，该区域用作原始地址或电话号码。如果信息是向另一个Modem模块的CPU数据传送，该区域可用作提供远端Modem模块的密码。该密码最大长度为15个字符。

协议域: 包含一个ASCII字符，指示Modem模块该如何格式化和传送信息。它可以是以下各值:

- 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 - UCP数据传送

标准域: 强制Modem模块使用一个特定的Modem标准。该标准域是一个ASCII字符。它可以是以下各值:

- 1-Bell103
- 2-Bell212
- 3-V.21
- 4-V.22
- 5-V.22bit
- 6-V.23c
- 7-V.32
- 8-V.32bit
- 9-V.34 (缺省)

格式域: 是三个ASCII字符，指定信息传送时使用的数据位和校验。如果协议设为数字寻呼，该区域无效。该区域只允许以下两种设置:

- 8N1-8位数据位，无校验，一位停止位 (缺省)
- 7E1-7位数据位，偶校验，一位停止位

文本信息格式

文本信息格式定义了文本寻呼或SMS短信息的格式。这些信息格式包含文本和嵌入变量。文本信息是一个字符串，以长度字节开始，其后为ASCII字符。文本信息的最大长度是120个字节（包括长度字节）。

格式: <文本><变量><文本><变量>…

文本域包含ASCII字符。

变量域定义了一个Modem模块从本地CPU中读取的嵌入数据，其格式以及它在信息中的位置。百分号（%）用来标识变量域的开始和结束。地址和左边区域用冒号隔开。左、右区域之间的小数点可以是逗号也可以是点号。变量域格式如下：

%地址: 左.右格式%

地址域指定该嵌入数据的地址、数据类型和长度（如：VD100、VW50、MB20或T10）。可以是以下数据类型：I、Q、M、SM、V、T（只有字）、C（只有字）、和AI（只有字）。长度可以是字节、字和双字。

左区域指定小数点左边所显示的位数。该数值应足够大以显示包括负号在内的完整的嵌入数据。如果左区域为零，则该值前面显示零。左区域的有效值为0到10。

右区域定义小数点右边显示的位数。小数点右边的零总能够被显示，如果右区域的值为零，则该数值的显示不带小数点。右区域的有效值为0至10。

格式域指定嵌入数据的显示格式。可以是以下字符：

i - 有符号整数
u - 无符号整数
h - 十六进制数
f - 浮点数/实数

示例: “Temperature=%VW100:3.1f% Pressure=%VD200:4.3f%”。

CPU数据传送信息格式

无论是CPU到CPU或是CPU到Modbus的数据传送，其数据传送的格式都是由CPU数据传送信息格式来定义。CPU数据传送信息是一个ASCII字符串。可以定义两台设备之间的任意数量的数据传送，其定义的最大数量不能超过信息长度的上限120字节（119字节的字符加上一个长度字节）。ASCII码的空格可用来分隔不同的数据传送，但不是必须的。所有的数据传送都在一个链接中执行。数据传送按照在信息中定义的顺序进行。如果在数据传送中检测到错误，到远程设备的连接将被断开，并且后面的传送将不被处理。

如果是读操作，从远端设备的“远端地址”开始读“数量”所指定的字符数，然后把这些数据写到本地CPU从“本地_地址”开始的V区中。

如果是写操作，从本地CPU的“本地_地址”开始读取“数量”所指定的字符数，然后将这些数据写到远端设备从“远端_地址”开始的区域。

格式: <操作>=<数量>, <本地_地址>,<远端_地址>

操作域中是一个ASCII字符，指定传送类型。

R - 从远端设备读数据

W - 向远端设备写数据

数量域指定要传送的字符数。该区域的有效值为1到100字。

本地_地址指定本地CPU中数据传送的V区地址（如：VW100）。

远端_地址指定远端设备中的数据传送地址（如：VW500）。即使是与Modbus设备传送数据，这个地址总是指定为V区。如果远端设备是一个Modbus设备，则V区地址与Modbus地址之间的对应关系如下：

$$\text{Modbus地址} = 1 + (\text{V区地址}/2)$$

$$\text{V区地址} = (\text{Modbus地址} - 1) * 2$$

示例: R=20, VW100, VW200 W=50, VW500, VW1000 R1=100, VW1000, VW2000

使用USS协议库去控制一个 MicroMaster驱动

STEP 7-Micro/WIN指令库，该指令库包括预先组态好的子程序和中断程序，这些子程序和中断程序都是专门为通过USS协议与驱动通讯而设计的。通过USS指令，您可以控制这个物理驱动，并读/写驱动参数。

您可以在STEP 7-Micro/WIN指令树的库文件夹中找到这些指令。当您选择一个USS指令时，系统会自动增加一个或多个相关的子程序（USS1到USS7）。

西门子库在一张单独的CD上出售，STEP 7-Micro/WIN附加件：指令库，订货号是6ES7 830-2BC00-0YX0。在订购和安装了1.1版本的西门子库后，任何后续STEP 7-Micro/WIN V3.2x和V4.0升级都会在不需要附加费用的情况下自动升级您的库（当增加或修改库时）。

本章内容:

使用USS协议的要求	332
计算与驱动通讯的时间要求	332
使用USS指令	333
USS协议指令	334
USS协议示例程序	341
USS执行错误代码	342
连接并设置3系列MicroMaster驱动	342
连接和设置4系列MicroMaster驱动	345

使用USS协议的要求

STEP 7-Micro/WIN指令库提供14个子程序、3个中断程序和8条指令支持USS协议。USS指令使用S7-200中的下列资源:

- 初始化USS协议将Port0专用作USS通讯
使用USS_INIT指令为Port0选择USS或PPI（USS是指SIMOTIONMicroMaster驱动USS协议）。在选择使用USS协议与驱动通讯后，Port0不能够再用作其它目的，包括与STEP 7-Micro/WIN通讯。
在使用USS协议为应用进行程序开发的过程中，您应该使用CPU 224XP, CPU 226, 或EM277 PROFIBUS-DP模块连接到您的计算机中的PROFIBUS CP卡上。STEP 7-Micro/WIN可以利用第二个通讯口在USS协议运行时监视程序。
- USS指令影响所有的与Port 0自由口通讯相关的SM区。
- USS指令使用14个子程序和3个中断程序。
- USS指令使得您的用户程序对存储空间的需求最多可增加3600字节。根据所使用的特定的USS指令，这些指令所支持的路径使控制程序对存储空间的分摊增加至少2300字节，最多3600字节。
- USS指令的变量需要400字节的V存储区。该区域的起始地址由用户指定并保留给USS变量。
- 有一些USS指令还要求16字节的通讯缓存区。作为一个指令的参数，您要为该缓存区提供一个V区的起始地址。建议为每一例USS指令指定一个单独的缓存区。
- 在执行计算时，USS指令使用累加器AC0至AC3。您仍然可以在您的程序中使用这些累加器，只是累加器中的数值会被USS指令改变。
- USS指令不能用在中断程序中。



提示

要将Port0恢复为PPI使之与STEP 7-Micro/WIN通讯，您可以使用另外一条USS_INIT指令重新设定Port 0。

您还可以将S7-200的模式开关设为STOP，这样就复位Port 0的参数。请注意，停止与驱动的通讯也就停止了驱动。

计算与驱动通讯的时间要求

S7-200的循环扫描和驱动的通讯是异步的。S7-200在完成一个驱动的通讯传送之前通常要完成若干个循环扫描。以下各因素可帮助您确定所需要的时间：S7-200的当前驱动数、波特率和扫描时间。

有一些驱动在使用参数访问指令时要求更长的延时。参数访问对时间的需求量取决于驱动的类型和要访问的参数。

在使用USS_INIT指令将Port 0指定为USS协议后，S7-200会以表11-1所示的时间间隔轮询所有激活的驱动。您必须为每个驱动设置超时（time-out）参数以完成该任务。

表11-1 通讯时间

波特率	对激活的驱动进行轮询的时间间隔 (无参数访问指令激活)
1200	240毫秒（最大）乘以驱动的数量
2400	130毫秒（最大）乘以驱动的数量
4800	75毫秒（最大）乘以驱动的数量
9600	50毫秒（最大）乘以驱动的数量
19200	35毫秒（最大）乘以驱动的数量
38400	30毫秒（最大）乘以驱动的数量
57600	25毫秒（最大）乘以驱动的数量
115200	25毫秒（最大）乘以驱动的数量



提示

同时只能有一个USS_RPM_x或USS_WPM_x指令激活。要等到每个指令的Done位输出指示完成，才能通过用户逻辑触发一个新的指令。
每个驱动只能使用一个USS_CTRL指令。

使用USS指令

为了在您的S7-200程序中使用USS协议指令，遵循下列步骤：

1. 在您的程序中插入USS_INIT指令并且该指令只在一个循环周期内执行一次，您可以使用USS_INIT指令启动或改变USS通讯参数。

当您插入USS_INIT指令时，若干个隐藏的子程序和中断服务程序会自动地加入到您的程序中。

2. 在您的程序中为每个激活的驱动只使用一个USS_CTRL指令。

您可以按需求尽可能多地使用USS_RPM_x和USS_WPMx指令，但是，在同一时刻，这些指令中只能有一条是激活的。

3. 在指令树中选中原程序块图标（Program Block）点击右键（显示菜单）为这些库指令分配V区。

选择库存储区选项，显示库存储区分配对话框。

4. 组态驱动参数使之与程序中所用的波特率和站地址相匹配。

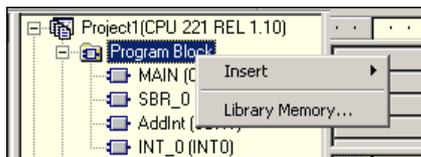


图11-1 为库指令分配V区

5. 连接S7-200和驱动之间的通讯电缆。

确保象S7-200这样的所有连接驱动的控制设备，通过一条短而粗的电缆连接到同一个地或象驱动一样星形连接。

小心

具有不同参考电位的设备相互连接时会在连接电缆中形成电流。这些电流会导致通讯错误或设备损坏。

确保所有通过通讯电缆连接在一起的设备共享一个公共参考点，或者彼此隔离以避免产生电流。

屏蔽层必须接到底盘地或9针接头的针1。建议您将MicroMaster驱动上的接线端2-0V接到外壳地上。

USS协议指令

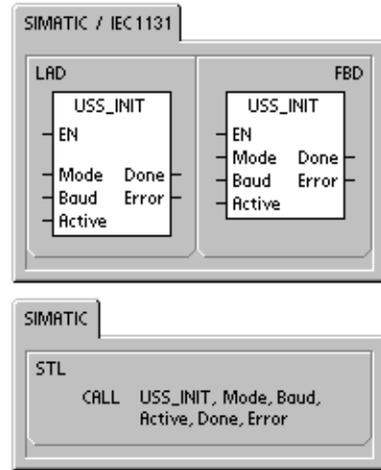
USS_INIT指令

USS_INIT指令用来使能、初始化或禁止MicroMaster驱动通讯。USS_INIT指令必须无错误地执行，才能够执行其它的USS指令。指令完成后，在继续进行下一个指令之前，Done位立即被置位。

当EN输入接通时，每一循环都执行该指令。

在每一次通讯状态改变时只执行一次USS_INIT指令。使用边缘检测指令脉冲触发EN输入接通。要改变初始化参数，需执行一个新的USS_INIT指令。

通过USS输入值可选择不同的通讯协议：输入值为1指定Port 0为USS协议并使能该协议，输入值为0指定Port 0为PPI并且禁止USS协议。



Baud设置波特率为1200、2400、4800、9600、19200、38400、57600或115200。S7-200 CPU版本1.2或之后的版本支持波特率57600和115200。

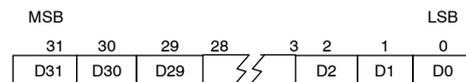
Active指示哪个驱动激活，有些驱动只支持地址0到30。

表11-2 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、常数、*VD、*AC、*LD
Baud、Active	DWORD	VD、ID、QD、MD、SD、SMD、LD、常数、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所示为对激活的驱动输入的描述和格式。所有标为Active（激活）的驱动都会在后台被自动地轮询，控制驱动搜索状态，防止驱动的串行链接超时。

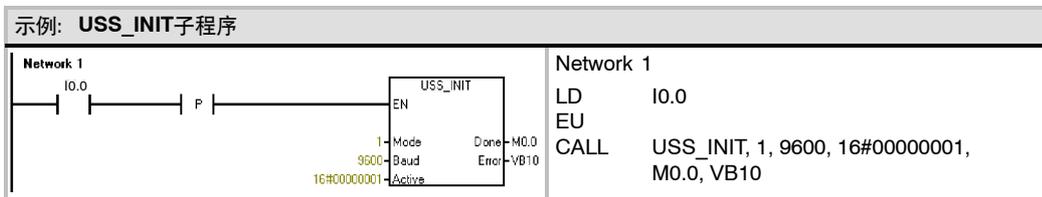
参见表11-1计算各状态间轮询的时间。



D0 驱动0激活位；0 - 驱动未激活，1 - 驱动激活
D1 驱动1激活位；0 - 驱动未激活，1 - 驱动激活
...

图11-2 Active参数的格式

当USS_INIT指令完成时，Done位输出接通。Error输出字节中包含该指令的执行结果。表11-6定义了该指令执行可能引起的错误条件。



USS_CTRL指令

USS_CTRL指令用于控制激活的MicroMaster驱动。

USS_CTRL指令将选择的命令放到通讯缓冲区内；然后，如果已经在USS_INIT指令的激活参数中选择了驱动，则此命令将被发送到寻址驱动（驱动参数）中。

对于每一个驱动只能使用一个USS_CTRL指令。

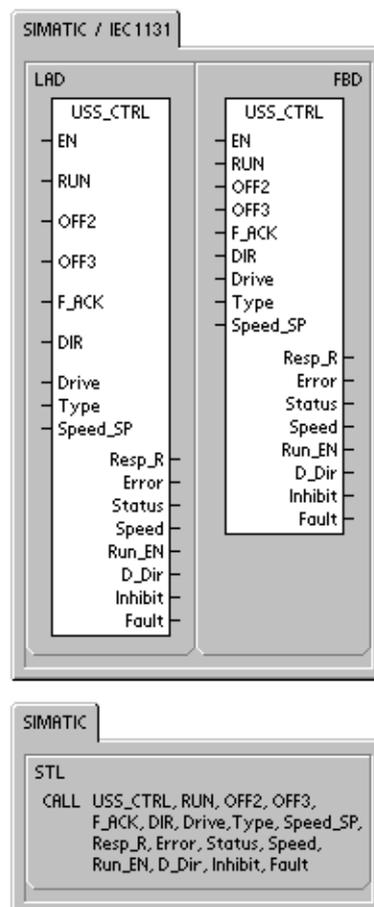
有些驱动只以正值的形式报告速度。如果速度为负，驱动报告一个正的速度值同时反向D_Dir位（方向位）。

EN位必须接通使能USS_CTRL指令。该指令要始终保持使能。

RUN（RUN/STOP）指示驱动是否接通（1）或断开（0）。当RUN位接通时，MicroMaster驱动接收命令，以指定的速度和方向运行。为使驱动运行，必须满足以下条件：

- 该驱动必须在USS_INIT中激活。
- OFF2和OFF3必须设为0。
- Fault和Inhibit位必须为0。

当RUN断开时，命令MicroMaster驱动斜坡减速直至电机停止。OFF2位用来允许MicroMaster驱动斜坡减至停止，OFF3位用来命令MicroMaster驱动快速停止。



Resp_R（响应收到）位应答来自驱动响应，轮询所有激活的驱动以获得最新的驱动的状态信息。每次S7-200接收到来自驱动响应时，Resp_R位在一个循环周期内接通并且刷新以下各值。

F_ACK（故障应答）位用于应答驱动故障。当F_ACK从0变1时，驱动清除该故障（Fault）。

DIR（方向）位指示驱动应向哪个方向运动。

表11-3 USS_CTRL指令的参数。

输入/输出	数据类型	操作数
RUN、OFF 2、OFF 3、F_ACK、DIR	BOOL	I、Q、M、S、SM、T、C、V、L、能流
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、常数
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、常数
Speed	REAL	VD、ID、QD、MD、SD、SMD、LD、AC、*VD、*AC、*LD

Drive（驱动地址）是MicroMaster驱动的地址，USS_CTRL命令发送到该地址。有效地址为0到31。

Type（驱动类型）选择驱动的类型。对于3系列的（或更早的）MicroMaster驱动，类型为0，对于4系列的MicroMaster驱动，类型为1。

Speed_SP（速度设定值）是驱动的速度，是满速度的百分比。Speed_SP的负值使驱动反向旋转。范围是：-200.0%至200.0%。

Error是错误字节，包含最近一次向驱动发出的通讯请求的执行结果。表11-6定义了该指令执行可能引起的错误条件。

Status是驱动返回的状态字的原始值。图11-3所示为标准状态字的状态位及主反馈。

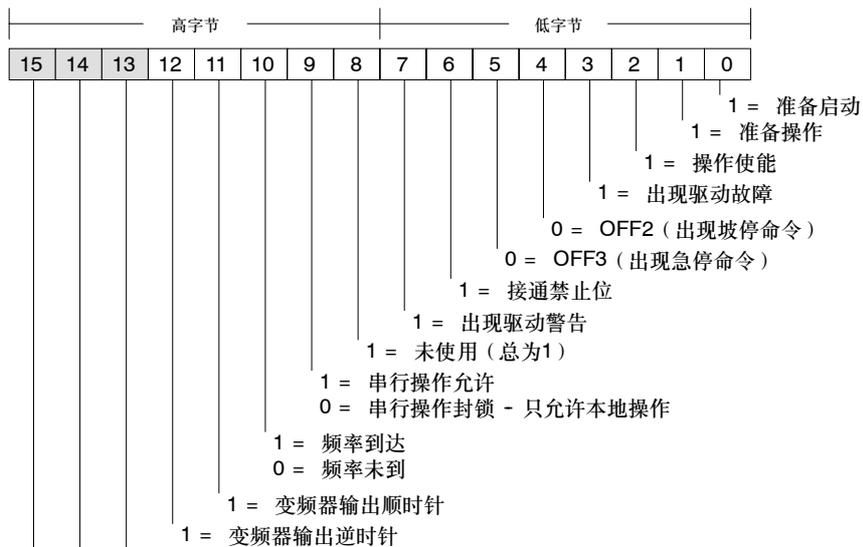
Speed是驱动速度，是满速度的百分比，范围是：-200.0%至200.0%。

Run_EN（RUN使能）指示驱动是运行（1）还是停止（0）。

D_Dir指示驱动转动的方向。

Inhibit指示驱动上禁止位的状态（0 - 未禁止，1 - 禁止）。要清除禁止位，Fault（故障）位必须为零，而且RUN、OFF2和OFF3输入必须断开。

Fault指示故障位的状态（0 - 无故障，1 - 有故障）。驱动显示故障代码。（请参考您的驱动手册）。要清除Fault，必须排除故障并接通F_ACK位。



留作将来之用：这些位不一定总是0。

图11-3 3系列小变频标准状态字的状态位和主反馈

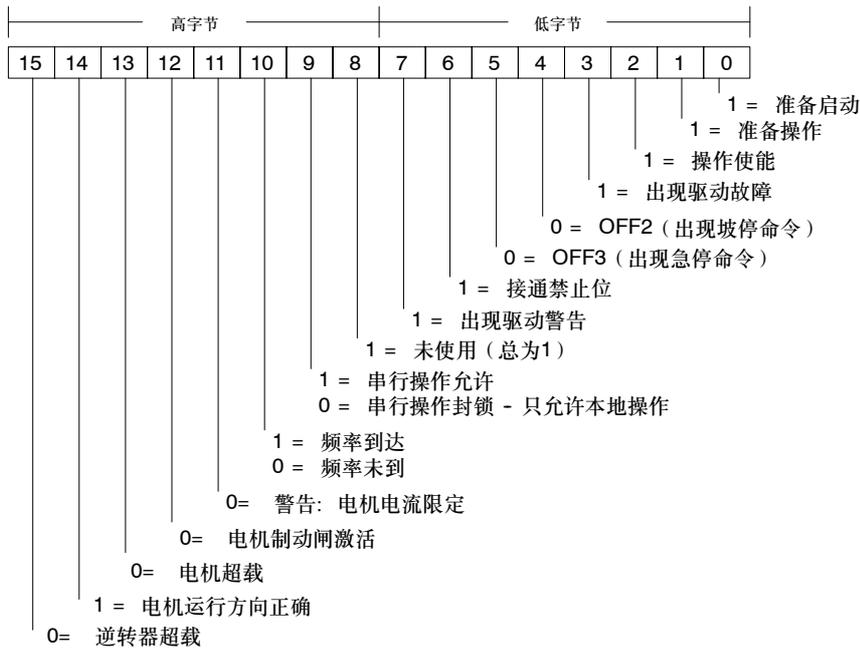


图11-4 4系列小变频标准状态字的状态位和主反馈

示例: USS_CTRL子程序

Network 1

SM0.0	EN
I 0.0	RUN
I 0.1	OFF2
I 0.2	OFF3
I 0.3	F_ACK
I 0.4	DIR

0	Drive	Resp_R	M0.0
1	Type	Error	VB2
100.0	Speed_SP	Status	VW4
		Speed	VD6
		Run_EN	AQ.0
		D_Dir	AQ.1
		Inhibit	AQ.2
		Fault	AQ.3

只显示为STL:

```

Network 1 //0号驱动的控制
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
    
```

显示为LAD或FBD:

```

Network 1 //0号驱动的控制
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

有三个用于USS协议的读指令：

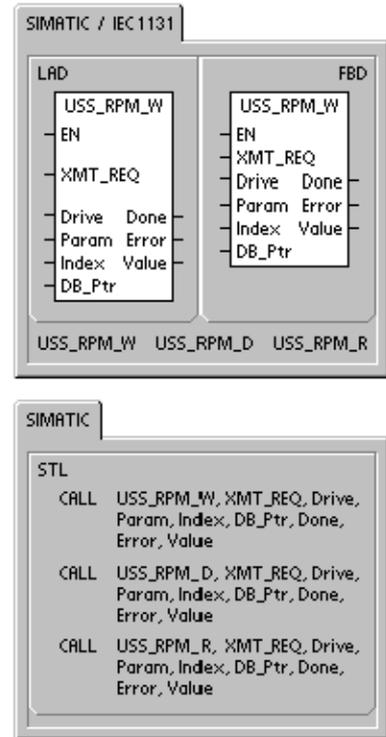
- USS_RPM_W指令读取一个无符号字类型的参数
- USS_RPM_D指令读取一个无符号双字类型的参数
- USS_RPM_R指令读取一个浮点数类型的参数

同时只能有一个读（USS_RPM_x）或写（USS_WPM_x）指令激活。

当MicroMaster驱动对接收的命令应答或有报错时，USS_RPM_x指令的处理结束。逻辑扫描继续执行。

要能使对一个请求的传送，EN位必须接通并且保持为1直至Done位置1，即意味着过程结束。例如，当XMT_REQ输入接通时，每一循环扫描向MicroMaster驱动传送一个USS_RPM_x请求。因此，应使用脉冲边沿检测作为XMT_REQ的输入，这样，每当EN输入有一个正的改变时，只发送一个请求。

Drive是向其发送USS_RPM_x命令的MicroMaster驱动的地址。每个驱动的有效地址为0到31。



Param是参数号码，Index是要读的参数的索引值。Value是返回的参数数值。您必须为DB_Ptr输入提供一个16字节缓存区的地址。该缓存区用于存储向MicroMaster驱动发送的命令的执行结果。

当USS_RPM_x指令结束时，Done输出接通，Error输出字节和Value输出包含该指令的执行结果。表11-6定义了该指令执行可能引起的错误条件。只有Done位输出接通时Error和Value输出才有效。

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、能流，上升沿有效
Drive	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD、常数
Param、Index	WORD	VW、IW、QW、MW、SW、SMW、LW、T、C、AC、AIW、*VD、*AC、*LD、常数
DB_Ptr	DWORD	&VB
Value	WORD D W O R D 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

有三个用于USS协议的写指令:

- USS_WPM_W指令写一个无符号字类型的参数
- USS_WPM_D指令写一个无符号双字类型的参数
- USS_WPM_R指令写一个浮点数类型的参数

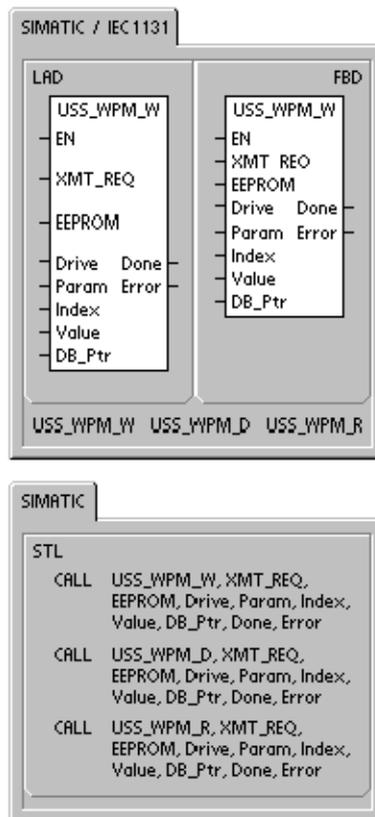
同时只能有一个读 (USS_RPM_x) 或写 (USS_WPM_x) 指令激活。

当MicroMaster驱动对接收的命令应答或有报错时, USS_WPM_x指令处理结束, 在这一过程等待应答时, 在这一过程等待应答时, 逻辑扫描继续执行。

要使能对一个请求的传送, EN位必须接通并且保持为1直至Done位置1, 即意味着过程结束。例如, 当XMT_REQ输入接通时, 每循环扫描向MicroMaster驱动传递一个USS_WPM_x请求。因此, 应使用脉冲边沿检测作为XMT_REQ的输入, 这样, 每当EN输入有一个正的改变时, 只发送一个请求。

Drive是向其发送USS_WPM_x命令的MicroMaster驱动的地址。每个驱动的有效地址为0到31。

Param是参数号码, Index是要写的参数的索引值。Value是要写到驱动上的RAM中的参数值。对于3系列的MicroMaster驱动, 您还可以将该值写到驱动上的EEPROM中, 这将基于您对P971的组态 (EEPROM存储控制)。



您必须为DB_Ptr输入提供一个16字节缓存区的地址。该缓存区由USS_WPM_x指令使用, 存储向MicroMaster驱动发送的命令的执行结果。

当USS_WPM_x指令结束时, Done输出接通, Error输出字节包含该指令的执行结果。表11-6定义了该指令执行可能引起的错误条件。

当EEPROM输入接通后, 指令对驱动的RAM和EEPROM都进行写操作。当此输入断开后, 指令只对驱动的RAM进行写操作。由于MicroMaster 3驱动并不支持此功能, 所以您必须确保输入为断开, 以便能对一个MicroMaster 3驱动使用此指令。

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、能流, 上升沿有效
EEPROM	BOOL	I、Q、M、S、SM、T、C、V、L、能流
Drive	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、*VD、*AC、*LD、常数
Param、Index	WORD	VW、IW、QW、MW、SW、SMW、LW、T、C、AC、AIW、*VD、*AC、*LD、常数
DB_Ptr	DWORD	&VB
Value	WORD D W O R D 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

小心
 当您使用USS_WPM_x指令刷新存储在驱动的EEPROM中的参数设置时，必须确保不超过对EEPROM写周期的最大次数的限定（大约50,000次）。
 写周期超限将引起存储数据的崩溃和数据丢失，读周期的次数没有限定，
 如果需要频繁地向驱动写参数，您首先要将驱动中的EEPROM存储控制参数设为零。

示例: USS_RPM_x和USS_WPM_x

<p>Network 1</p> <p>Network 2</p>	<p>Network 1 //这两个接触器必须有相同地址</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, &VB100, M0.0, VB10, VW200 </pre> <p>Network 2 //这两个接触器必须有相同地址</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, &VB120, M0.1, VB11 </pre>
---	---

USS协议示例程序

示例: 能够以STL方式正确显示的USS指令示例程序

<p>Network 1</p> <p>Network 2</p> <p>Network 3</p> <p>Network 4</p>	<pre> Network 1 //初始化USS协议: //在首次扫描中使能USS协议 //Port0的波特率为19200, //地址为“0” //的驱动激活。 LD SM0.1 CALL USS_INIT, 1, 19200, 16#00000001, Q0.0, VB1 Network 2 //控制0号驱动的参数 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, Q0.1, Q0.2, Q0.3, Q0.4 Network 3 //从0号驱动读一个字参数 //读索引为0的参数5。 //1. 将I0.5 // 存入临时变量以便该段 // 程序可显示为LAD //2. 将I0.5的上升沿存入临时变量 // 以便将它传递给子程序 // LD I0.5 = L60.0 LD I0.5 EU = L63.7 LD L60.0 CALL USS_RPM_W, L63.7, 0, 5, 0, &VB20, M0.1, VB10, VW12 Network 4 //向0号驱动写一个字参数 //写索引为0的参数2000。 LD I0.6 = L60.0 LD I0.6 EU = L63.7 LDN SM0.0 = L63.6 LD L60.0 CALL USS_WPM_R, L63.7, L63.6, 0, 2000, 0, 50.0, &VB40, M0.2, VB14 </pre> <p>注意: 该STL指令并不与LAD或FBD兼容</p>
---	--

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	没有错误
1	驱动没响应
2	来自驱动响应中检测到校验和错误
3	来自驱动响应中检测到奇偶校验错误
4	由来自用户程序的干扰引起的错误
5	尝试非法命令
6	提供非法驱动地址
7	通讯口未设为USS协议
8	通讯口正忙于处理某条指令
9	驱动速度输入超限
10	驱动响应的长度不正确
11	驱动响应的第一个字符不正确
12	驱动响应的长度字符不被USS指令所支持
13	错误的驱动响应
14	提供的DB_Ptr地址不正确
15	提供的参数号码不正确
16	所选协议无效
17	USS激活; 不允许改变
18	指定的波特率非法
19	没有通讯: 该驱动未激活
20	驱动响应的参数或数值不正确或包含错误代码
21	请求一个字类型的数值却返回一个双字类型值。
22	请求一个双字类型的数值却返回了一个字类型值。

连接并设置3系列MicroMaster驱动

连接3系列MicroMaster驱动

您可以使用标准的PROFIBUS电缆和接头来连接S7-200和3系列的MicroMaster驱动（MM3），合适的互连电缆的电缆偏置和终端，见图11-5。

小心

具有不同参考电位的设备相互连接时会在连接电缆中形成电流。

这些电流会导致通讯错误或设备损坏。

确保您要用通讯电缆连接在一起的设备要么共享一个公共参考点，要么彼此隔离以防止未预期电流的形成。

屏蔽层必须接到底盘地或9针接头的针1。建议您将MicroMaster驱动上的接线端2-0V接到外壳地上。

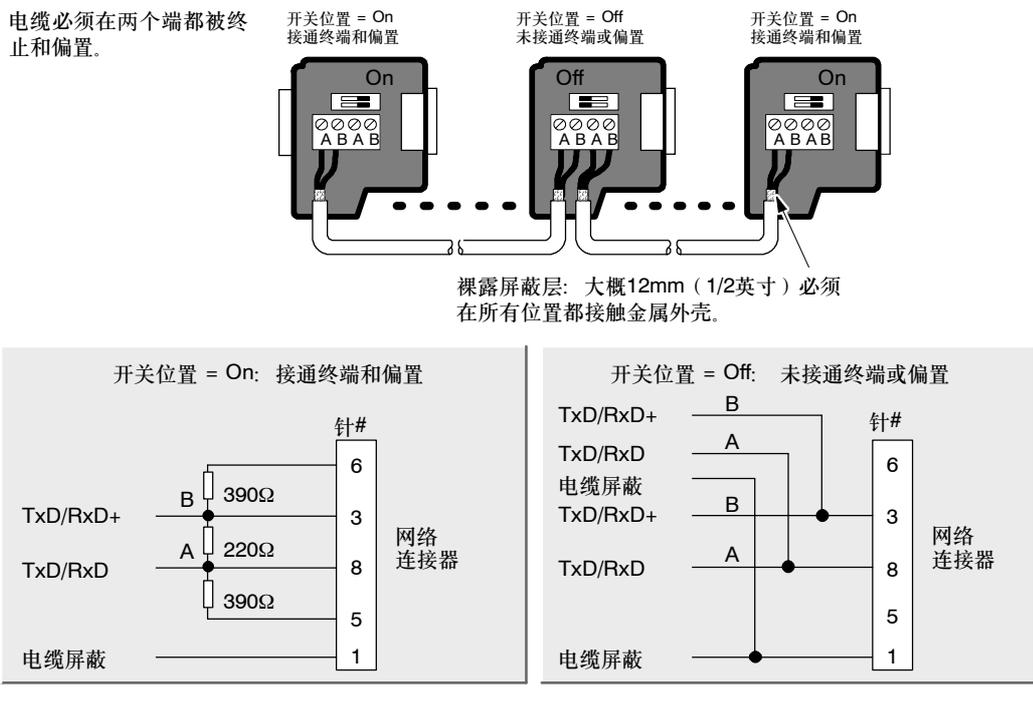


图11-5 网络电缆的偏置和终端

设置3系列MicroMaster驱动

在将驱动连至S7-200之前，您必须确保驱动具有以下系统参数。使用驱动上的按键设置参数：

1. 将驱动恢复为出厂设置（可选）。按P键：显示P000，按向上或向下的键头直至显示P944，按P输入这些参数。
P944=1
2. 使能对所有参数的读/写访问。按P键。按向上或向下的键头直至显示P009。按P输入这些参数。
P009=3
3. 检查您的驱动的电机设置。这些设置因使用的电机而不同。按P键。按上、下键头直至显示您的驱动的电机设置。按P输入这些参数。
P081=电机的标识频率（HZ）
P082=电机的标识速度（RPM）
P083=电机的标识电流（A）
P084=电机的标识电压（V）
P085=电机的标识功率（kW/HP）
4. 设置本地/远程控制模式。按P键。按上、下键头直至显示P910。按P输入这些参数。
P910=1远控模式

5. 设置RS-485串口的波特率。按P键。按上、下键直至显示P092。按P输入这些参数。按上、下键直至显示RS-485串口波特率的数值。按P输入。

P092	3	(1200波特)
	4	(2400波特)
	5	(5800波特)
	6	(9600波特 - 缺省)
	7	(19200波特)
6. 输入从站地址。每个驱动（最多31）都可通过总线操作。按P键。按上、下键头直至显示P091。按P输入这些参数。按上、下键头直至显示您想要的从站地址，按P输入。
P091=0至31
7. 斜坡上升时间（可选）。这是一个以秒为单位的时间，在这个时间内，电机加速至最高频率。按P键。按上、下键头直至P002显示。按P输入这些参数。按上、下键头直至显示您想要的斜坡加速时间。按P输入。
P002=0-650.00
8. 斜坡下降时间（可选）。这是一个以秒为单位的时间，在这个时间内，电机减速至完全停止。按P键。按上、下键头直至P003显示。按P输入这些参数。按上、下键直至显示您想要的减速时间。按P输入。
P003=0-650.00
9. 串行链接超时。这是到来的两个数据报文之间最大的间隔时间。该特性可用来在通讯失败时关断变频器。

当收到一个有效的数据报文后，计时启动。如果在指定的时间内未收到下一个数据报文，变频器将触发并显示故障代码F008，该值设为零则关断该控制。使用表11-1计算对驱动状态轮询的时间。

按P键。按上、下键头直至显示P093。按P输入这些参数。按上、下键直至显示您想要的串行链接超时。按P输入。
P093=0-240（0为缺省；时间单位为秒）
10. 串行链接标识系统设定值。该值可能不一样，但典型值为50Hz或60Hz，它定义了PV或SP的100%值。按P键。按上、下键直至显示P094。按P输入这些参数。按上、下键直至显示您想要的串行链接标识系统设定。按P输入。
P094=0-400.00
11. USS兼容性（可选）。按P键。按上、下键直至显示P095。按P输入这些参数。按上、下键直至显示与您想要的USS兼容性相对应的号码。按P输入。

P095 =	0	0.1Hz分辨（缺省）
	1	0.01Hz分辨
12. EEPROM存储控制（可选）。按P键。按上、下键直至显示P971。按P输入这些参数。按上、下键直至显示与EEPROM存储控制相对应的号码。按P输入。

P971 =	0	掉电时参数设置的改变（包括P971）丢失
	1	（缺省）掉电时，参数设置的改变保留。
13. 操作显示。按P退出参数模式。

连接和设置4系列MicroMaster驱动

连接4系列MicroMaster驱动

连接4系列MicroMaster驱动，将485电缆的两端插入为USS操作提供的两个卡式接线端。标准的PROFIBUS电缆和连接器可以用于连接S7-200。

小心

具有不同参考电位的设备相互连接时会在连接电缆中形成电流。

这些电流会导致通讯错误或设备损坏。

确保您要用通讯电缆连接在一起的设备要么共享一个公共参考点，要么彼此隔离以防止未预期电流的形成。

屏蔽层必须接到底盘地或9针接头的针1。建议您将MicroMaster驱动上的接线端2-0V接到外壳地上。

如图11-6所示，RS-485另外一端的两根接线必须插在MM4驱动的终端。在做MM4驱动的电连接时，取下驱动的前盖板露出接线终端。关于如何取下您的驱动的前盖板，请参见MM4用户手册。

接线终端的连接以数字标识。在S7-200端使用PROFIBUS连接器，将A端连至驱动端的15（对MM420而言）或30（对MM440而言），将B端连接到接线端14（MM420）或29（MM440）。

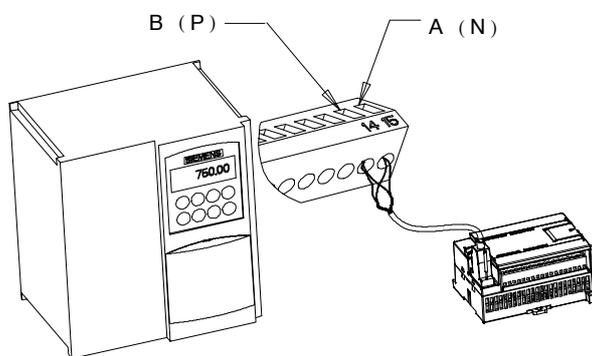


图11-6 连接到MM420的接线终端

如果S7-200是网络中的端点，或者，如果是点到点的连接，则必须使用连接器的端子A1和B1（而非A2和B2），因为这样可以接通终端电阻（例如，使用DP接头6ES7 972-0BA40-0XA0）。

小心

确保通电前驱动的前盖板正确地安置回原位。

如果驱动在网络中组态为端点站，那么终端和偏置电阻必须正确地连接至连接终端上。例如，图11-7是对MM4，SE6420驱动必须做的终端和偏置连接。

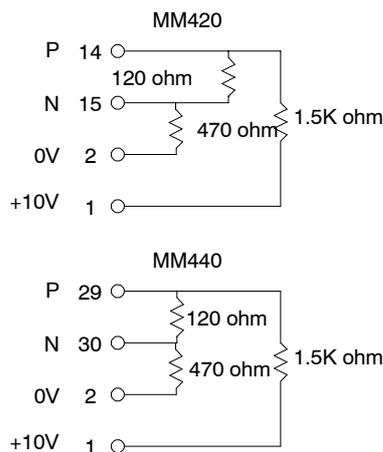


图11-7 终端和偏置示例

设置MM4驱动

在将驱动连至S7-200之前，您必须确保驱动具有以下系统参数。使用驱动上的按键设置参数：

1. 将驱动恢复为出厂设置（可选）：
P0010=30
P0970=1

如果您忽略该步骤，确保以下参数的设置：

USS PZD长度: P2012 Index 0=2
USS PKW长度: P2013 Index 0=127

2. 使能对所有参数的读/写访问（专家模式）：
P0003=3
3. 检查您的驱动的电机设置：
P0304=额定电机电压（V）
P0305=额定电机电流（A）
P0307=额定功率（W）
P0310=额定电机频率（Hz）
P0311=额定电机速度（RPM）

这些设置因使用的电机而不同。

要设置参数P304, P305, P307, P310和P311, 您必须先将参数P010设为1（快速调试模式）。当您完成参数设置后，将参数P010再设为0。参数P304, P305, P307, P310和P311只能在快速调试模式下修改。

4. 设置本地/远程控制模式：
P0700 Index0=5
5. 在COM链接中设置到USS的频率设定值：
P1000 Index0=5
6. 斜坡上升时间（可选）：
P1120=0至650.00
这是一个以秒为单位的时间，在这个时间内，电机加速至最高频率。
7. 斜坡下降时间（可选）：
P1121=0至650.00
这是一个以秒为单位的时间，在这个时间内，电机减速至完全停止。
8. 设置串行链接参考频率：
P2000=1至650Hz
9. 设置USS标准化：
P2009 Index0=0
10. 设置RS-485串口波特率：
P2010 Index0= 4 （2400波特）
5 （4800波特）
6 （9600波特）
7 （19200波特）
8 （38400波特）
9 （57600波特）
12 （115200波特）
11. 输入从站地址：
P2011 Index 0=0至31
每个驱动（最多31）都可通过总线操作。
12. 设置串行链接超时：
P2014 Index 0=0至65,535毫秒
（0=超时禁止）

这是到来的两个数据报文之间最大的间隔时间。该特性可用在通讯失败时关断变频器。当收到一个有效的数据报文后，计时启动。如果在指定时间内未收到下一个数据报文，变频器触发并显示故障代码F0070。该值设为零则关断该控制。使用表11-1计算对驱动状态轮询的时间。

13. 从RAM向EEPROM传送数据：
P0971=1（启动传送）将参数设置的改变存入EEPROM。

12

使用Modbus协议库

STEP 7-Micro/WIN指令库包含有专门为Modbus通讯设计的预先定义子程序和中断服务程序，使得与Modbus主站的通讯简单易行。使用Modbus从站协议指令，您可以将S7-200组态作为Modbus RTU从站，与Modbus主站通讯。

您可以在STEP 7-Micro/WIN指令树中找到这些指令。通过这些新指令，可以将S7-200作为Modbus从站。当选择一个Modbus从站指令时，会有一个或多个相关的子程序自动添加到您的项目中。

西门子库在一张单独的光盘上出售，STEP 7-Micro/WIN附加件：指令库，订货号为6ES7830-2BC00-0YX0。在订购和安装了1.1版本的西门子库后，任何后续的STEP 7-Micro/WIN V3.2x和V4.0升级都会在不需附加费用的情况下自动升级您的库（当增加或修改库时）。

本章内容:

使用Modbus协议的要求	348
Modbus协议的初始化和执行时间	348
Modbus地址	349
使用Modbus从站协议指令	350
Modbus从站协议指令	351

使用Modbus协议的要求

Modbus从站协议指令占用S7-200的以下资源:

- 初始化Modbus从站协议占用Port0作为Modbus从站协议通讯。
当Port0用作Modbus从站协议通讯时，它不能再用作任何其它目的，包括与STEP 7-Micro/WIN通讯。MBUS_INIT指令控制Port0的设定是Modbus从站协议还是PPI。
- Modbus从站协议指令响应与Port0自由口通讯相关的所有SM。
- Modbus从站协议指令使用3个子程序和2个中断服务程序。
- Modbus从站协议指令的两个Modbus从站指令及其支持子程序需占用1857字节的程序空间。
- Modbus从站协议指令的变量要求779字节的V区块。该块的起始地址由用户指定，保留给Modbus变量。



提示

使用MBUS_INIT指令重新设置Port0则可将Port0改为PPI操作，这样就可以和STEP 7-Micro/WIN通讯了。

您还可以将S7-200的模式开关设为STOP，这样就复位Port 0的参数。

Modbus协议的初始化和执行时间

Modbus通讯使用CRC（循环冗余检验）以确保通讯信息的完整性。Modbus从站协议使用一个预计算值的表以减少信息处理所需的时间。CRC表的初始化需要大约425毫秒。该初始化在MBUS_INIT内部完成，而且通常是在进入RUN模式的第一个用户程序周期完成。如果MBUS_INIT子程序和任何其它用户初始化所需的时间超过500毫秒的循环时间监控，您需要复位时间看门狗并保持输出使能（如果扩展模块要求）。输出模块时间看门狗可通过写模板输出复位。参见第6章中的看门狗复位指令。

当MBUS_SLAVE子程序进行请求服务时循环时间增加。由于大部分时间消耗在计算Modbus CRC上，所以对于每一字节的请求和响应，循环时间增加650微秒。最大的请求/响应（读或写120字）可增加循环时间大约165毫秒。

Modbus地址

Modbus地址通常是包含数据类型和偏移量的5个或6个字符值。第一个或前两个字符决定数据类型，最后的四个字符是符合数据类型的一个适当的值。Modbus主站则将这个地址对应到正确的功能上。Modbus从站指令支持以下地址：

- 000001至000128是实际输出，对应于Q0.0-Q15.7
- 010001至010128是实际输入，对应于 I0.0-I15.7
- 030001至030032是模拟输入寄存器，对应于AIW0至AIW62
- 040001至04XXXX是保持寄存器，对应于V区。

所有Modbus地址都是一个基础。表12-1所示为Modbus地址与S7-200地址的对应关系。

Modbus从站协议允许您对Modbus主站可访问的输入、输出、模拟输入和保持寄存器（V区）的数量进行限定。

MBUS_INIT指令的参数MaxIQ指定Modbus主站允许访问的实际输入或输出（I或Q）的最大数量。

MBUS_INIT指令的MaxAI参数指定Modbus主站允许访问的输入寄存器（AIW）的最大数量。

MBUS_INIT指令的MaxHold参数指定Modbus主站允许访问的保持寄存器（V存储区字）的最大数量。

请查看MBUS_INIT指令的描述，了解更多的关于为Modbus从站设置存储区的限制信息。

表12-1 映射Modbus地址至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)

组态符号表

当您为第一个符号输入一个地址后，符号表会自动计算并分配符号表中其余的符号地址。

您要为这个占用779字节的表分配一个起始的V区地址。确保对Modbus从站符号表的V区地址分配与MBUS_INIT指令中HoldStart和MaxHold参数为Modbus保持寄存器分配的V区地址不重叠。如果有存储区域的地址重叠，MBUS_INIT指令会返回一个错误。

使用Modbus从站协议指令

在S7-200程序中使用Modbus从站协议指令请遵循以下步骤:

1. 在您的程序中插入MBUS_INIT指令并且只在一个循环周期中执行该指令, MBUS_INIT指令可用于对Modbus通讯参数的初始化或修改。
当您插入MBUS_INIT指令时, 几个隐藏的子程序和中断服务程序会自动地添加到您的程序中。
2. 为Modbus从站协议指令所要求的V存储区设定一个起始地址, 该区域是一个779字节的连续区域。
3. 在您的程序中只使用一个MBUS_SLAVE指令。该指令在每个循环周期中执行, 为接收到的所有请求提供服务。
4. 用通讯电缆将S7-200 Port0和Modbus主站连接起来。

小心

具有不同参考电位的设备连接在一起可能会在连接电缆上引起未预期的电流。这种电流会引起通讯错误或设备损坏。

确保所有由通讯电缆连接在一起的设备要么共享一个公共参考点, 要么彼此隔离以防止产生未预期的电流。

累加器 (AC0、AC1、AC2、AC3) 由Modbus从站指令使用并显示在交叉参考列表中。在执行前, Modbus从站指令在累加器中的数值被存储并在Modbus从站指令完成前恢复到累加器中, 确保在执行Modbus从站指令时, 所有在累加器中的用户数据都得到保护。

Modbus从站协议指令支持Modbus RTU协议。这些指令使用S7-200的自由口功能, 支持大部分常用Modbus功能。以下是所支持的Modbus功能:

表12-2 支持的Modbus从站协议功能

功能	描述
1	读单个/多个线圈 (实际输出) 状态, 功能1返回任意数量输出点的接通/断开状态 (Q)
2	读单个/多个接触器 (实际输入) 状态。功能2返回任意数量的输入点的接通/断开状态 (I)
3	读单个/多个保持寄存器。功能3返回V存储器的内容。保持寄存器在Modbus下是字类型, 在一个请求中最多可读120个字。
4	读单个/多个输入寄存器。功能4 返回模拟输入值。
5	写单个线圈 (实际输出)。功能5将实际输出点设置为指定值。该输出点不是被强制, 用户程序可以应Modbus的请求重写该值。
6	写单个保持寄存器。功能6写一个单个保持寄存器的值到S7-200的V存储区。
15	写多个线圈 (实际输出)。功能15写多个实际输出值到S7-200的Q映象区。起始输出点必须是一个字节的开始 (如Q0.0或Q2.0), 并且要写的输出的数量是8的倍数。这是Modbus从站协议指令的限定。这些点不是被强制, 用户程序可以应Modbus的请求重写这些点的值。
16	写多个保持寄存器。功能16写多个保持寄存器到S7-200的V区。在一个请求中最多可写120字。

Modbus从站协议指令

MBUS_INIT指令

MBUS_INIT指令用于使能和初始化或禁止Modbus通讯。MBUS_INIT指令必须无错误的执行，然后才能够使用MBUS_SLAVE指令。在继续执行下一条指令前，MBUS_INIT指令必须执行完并且Done位被立即置位。

当EN输入接通时，该指令在每一循环周期内执行。

MBUS_INIT指令应该在每次通讯状态改变时只执行一次。因此，EN输入端应使用边沿检测元素以脉冲触发，或者只在第一个循环周期内执行一次。

Mode值选择通讯协议：输入1值将Port0定义为Modbus协议并使能该协议，输入值为0将Port0定义为PPI并禁止Modbus协议。

参数Baud设置波特率为1200、2400、4800、9600、19200、38400、57600或115200。S7-200CPU版本1.2或其以后版本的CPU支持57600和115200波特率。

参数Addr设置地址，其数值在1到247之间。

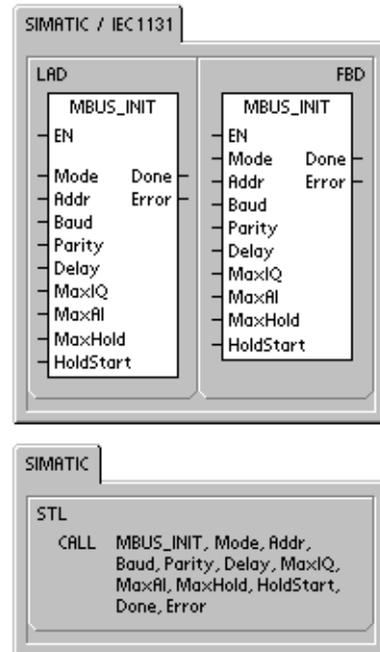


表12-3 MBUS_INIT指令的参数。

输入/输出	数据类型	操作数
Mode、Addr、Parity	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、常数、*VD、*AC、*LD
Baud、HoldStart	DWORD	VD、ID、QD、MD、SD、SMD、LD、AC、常数、*VD、*AC、*LD
Delay、MaxIQ、MaxAI、MaxHold	WORD	VW、IW、QW、MW、SW、SMW、LW、AC、常数、*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

参数Parity用于设置校验使之与Modbus主站相匹配。所有设置使用一个停止位。可接受值为:

- 0-无校验
- 1-奇校验
- 2-偶校验

参数Delay通过为标准Modbus信息超时增加指定数量的毫秒, 扩展标准Modbus信息结束超时条件。当在一个连接的网络上操作时, 该参数的典型值为0。如果您使用具有纠错功能的modem时, 将延迟时间设为50至100毫秒。如果您使用宽频电台, 设置该延迟值为10至100毫秒。Delay的数值可以是0到32767毫秒。

参数MaxIQ设置可使用的I和Q点数, 相应于Modbus地址00xxxx和01xxxx。其数值可为0到128。数值为0则禁止对输入和输出的读写。建议MaxIQ的取值为128, 即允许访问S7-200的所有I点和Q点。

参数MaxHold设置可使用的输入寄存器(AI)的字数, 相应于Modbus地址03xxx, 其数值为0到32。数值为0则禁止读模拟输入。要允许访问所有的S7-200模拟输入, MaxAI的建议值如下:

- CPU221为0
- CPU222为16
- CPU224、CPU224XP和CPU226为32。

参数MaxHold设置可以使用的V区的保持寄存器的字数, 相应于Modbus地址04xxx。例如, 要允许主站访问2000字节的V存储区, 则设置MaxHold为1000字(保持寄存器)。

参数HoldStart是V存储区的保持寄存器的起始地址。通常设为VB0, 所以参数HoldStart设为&VB0(VB0的地址)。也可以将其它的V区地址指定为保持寄存器的起始地址, 以便使VB0可以在项目中用作其它目的。Modbus主站可以访问起始地址为HoldStart, 字数为MaxHold的V存储区。

当MBUS_INIT指令完成时, Done输出接通。Error输出字节包含该指令的执行结果。表12-5定义了该指令执行可能引起的错误条件。

MBUS_SLAVE指令

MBUS_SLAVE指令用于服务来自Modbus主站的请求，必须在每个循环周期都执行，以便检查和响应Modbus请求。

当EN输入接通时，该指令在每一循环周期内执行。

MBUS_SLAVE指令无输入参数。

当MBUS_SLAVE指令响应Modbus请求时Done输出接通。如果没有服务的请求，Done输出会断开。

Error输出包含该指令的执行结果。该输出只有Done接通时才有效。如果Done断开，错误代码不会改变。表12-5定义了执行该指令可能引起的错误条件。

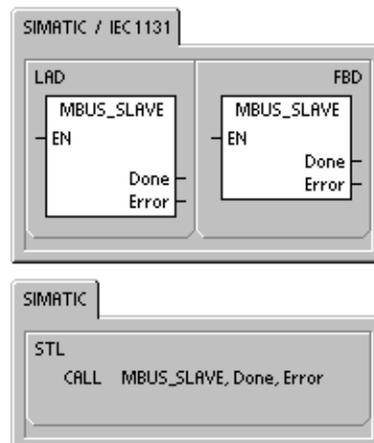
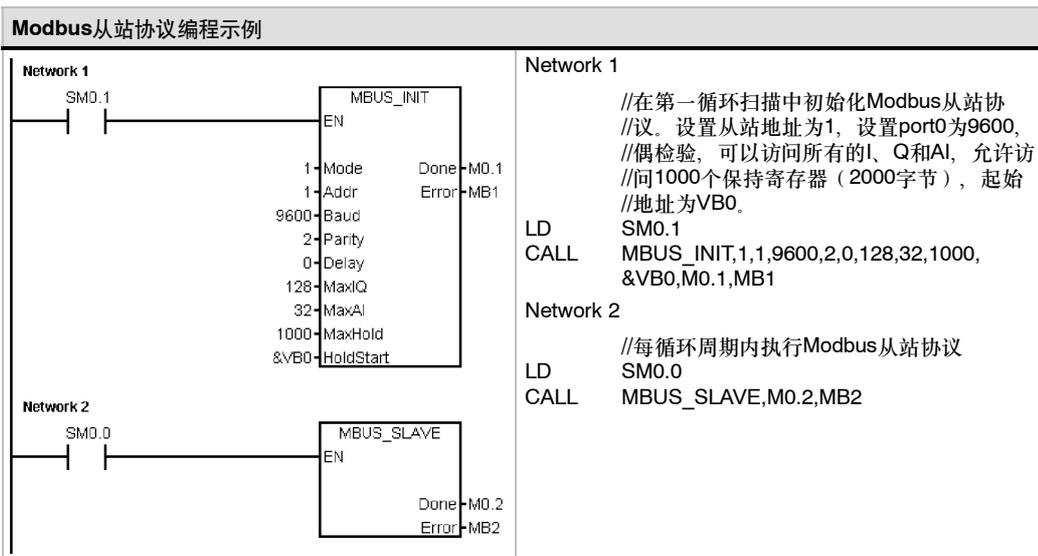


表12-4 MBUS_SLAVE指令的参数。

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从站协议执行错误代码

错误代码	描述
0	无错误
1	存储区范围错误
2	非法波特率或校验
3	非法从站地址
4	Modbus参数的非法值
5	保持寄存器与Modbus从站符号地址重叠
6	接收校验错误
7	接收CRC错误
8	非法功能请求/不支持的功能
9	请求中有非法存储区地址
10	从站功能未始能。



13

使用配方

STEP 7-Micro/Win软件中提供了配方向导程序来帮助您组织配方和定义配方。配方存在存储卡中，而不是PLC中。

本章内容:

概述	356
配方的定义和术语	357
使用配方向导	357
由配方向导创建的指令	361

概述



配方

STEP 7-Micro/WIN软件和S7-200 PLC已经支持配方功能。STEP 7-Micro/Win软件中提供了配方向导程序来帮助您组织配方和定义配方。

所有配方存在存储卡中。因此，为了使用配方功能，必须要在PLC中插入一块64K或者256K的存储卡。要查阅关于存储卡的更多信息，参见附录A。

所有配方存在存储卡中。但是，当用户程序处理一条配方时，该条配方被读入PLC的存储区。例如：如果您生产饼干的话，您会有很多种饼干的配方，巧克力夹心饼干、甜饼干和麦片饼干。但在同一时间只能生产一种饼干，因而必须选择合适的配方读入PLC的存储区。

图13-1阐述了一个使用配方来生产多种饼干的处理过程。每一种饼干的配方存在存储卡中。操作员使用TD 200C文本显示器来选择所要生产饼干的种类，用户程序将配方读入PLC的存储区。

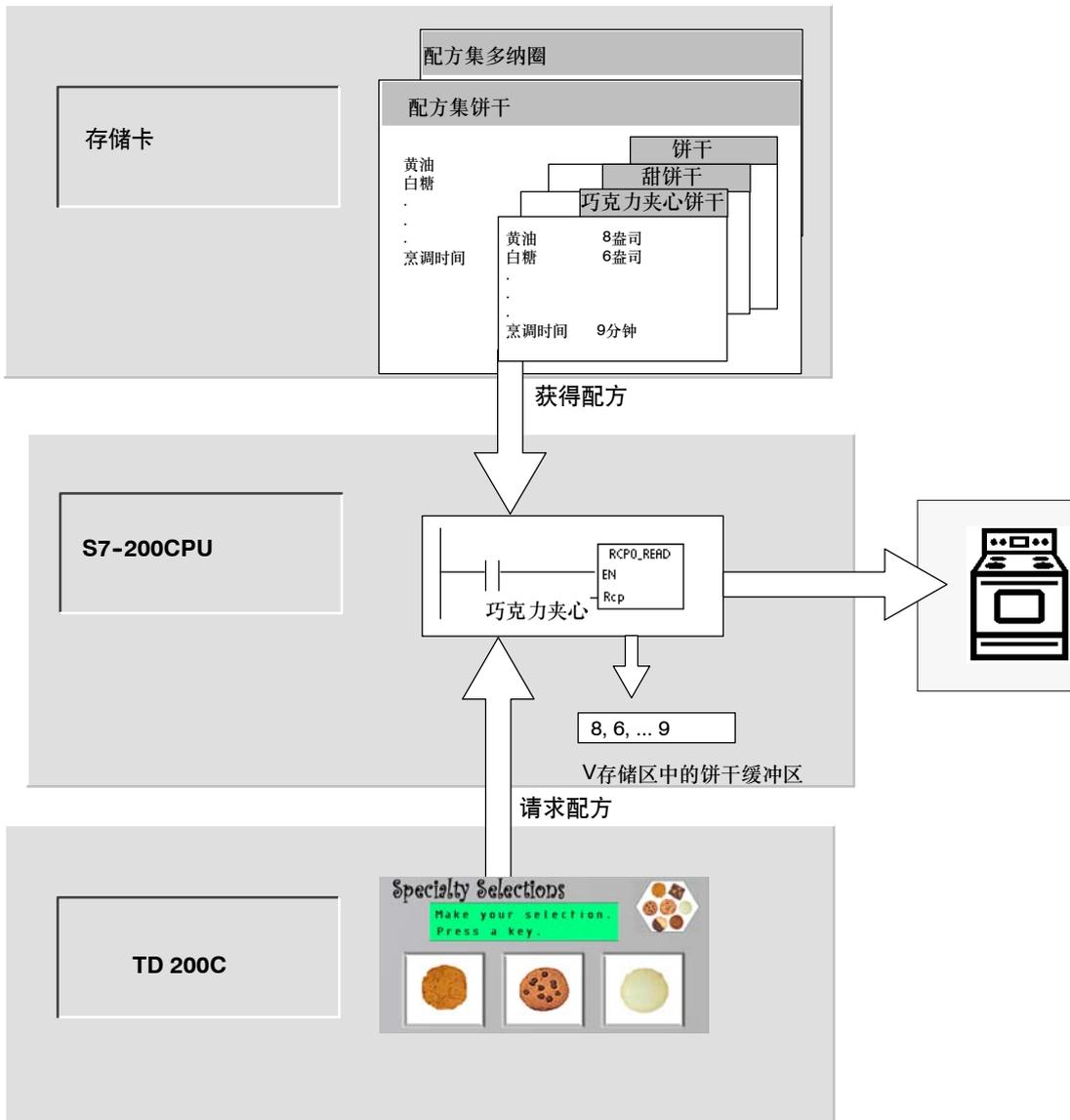


图13-1 配方应用举例

配方的定义和术语

为了帮助您理解配方向导，首先来解释以下定义和术语。

- 配方结构是由配方向导生成的一套组件。这些组件包括指令子程序。
- 配方集是指一个配方的集合，它们拥有相同的参数集合。但是依赖于配方，参数的数值各不相同。
- 一条配方是一组参数值，它提供了生产一种产品和控制一个过程所需要的信息。

例如：生产多纳圈和饼干的配方就分别属于不同的配方集。而饼干配方集中又包含多种不同的配方，例如：巧克力夹心饼干和甜饼干。表13-1中列出了例子中的域和值。

表13-1 配方应用举例-饼干

域名	数据类型	巧克力夹心 (配方0)	甜饼干 (配方1)	注释
黄油	字节	8	8	盎司
白糖	字节	6	12	盎司
红糖	字节	6	0	盎司
鸡蛋	字节	2	1	个
香草	字节	1	1	茶匙
面粉	字节	18	32	盎司
碳酸氢钠	实数	1.0	0.5	茶匙
发酵粉	实数	0	1.0	茶匙
盐	实数	1.0	0.5	茶匙
巧克力夹心	实数	16	0.0	盎司
柠檬皮	实数	0.0	1.0	大汤匙
烹调时间	实数	9.0	10.0	分钟

使用配方向导

使用配方向导来创建配方和配方集。配方是存在存储卡中的。使用配方向导可以直接输入配方和配方集。将来可以再次运行配方向导来改变配方，或者在用户程序中调用RCPx_WRITE指令子程序。

用配方向导来创建配方结构包含以下步骤：

- 为每个配方集建立一个符号表。每张表中都包含着与配方域名相同的符号名。这些符号定义了访问当前载入PLC存储区的配方值的V存储区地址。每张表还包含一个用于标识每个配方的符号常数。
- 为每个配方集建立一个数据块标签。这个标签定义了符号表中所描述的V存储区的起始地址。
- 生成一个RCPx_READ指令子程序。该指令用于将指定的配方从存储卡中读取到V存储区中。
- 生成一个RCPx_WRITE指令子程序。该指令用于将V存储区中的配方值写入存储卡中。

定义配方

要用配方向向导创建一个配方，可在命令菜单中选择**Tools > Recipe Wizard**。这时屏幕上将出现配方向向导基本操作的一个简单介绍。点击**Next**，开始配制您的配方。

要创建一个配方集，执行以下步骤。参见图13-2。

1. 为配方集指定域名。如同预先定义的那样，每一个名字都将成为项目中的一个符号名。
2. 在下拉列表中选择数据类型。
3. 为每个名字输入缺省值和注释。在该配方集中的所有新配方将使用这些缺省值作为初始值。
4. 点击**Next**，编辑配方集中的每条配方。

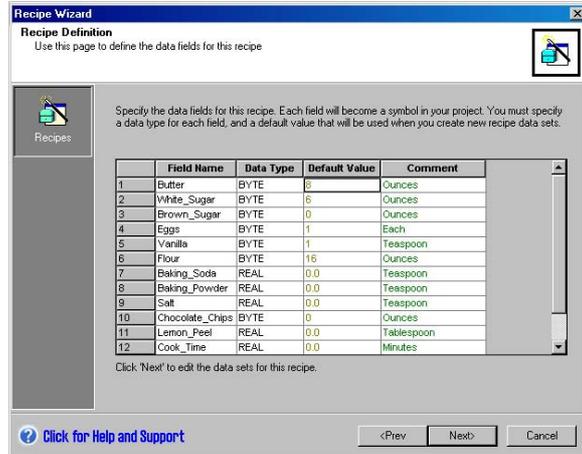


图13-2 定义配方

在配方表中为每条配方定义所有的数据域。您最多可以有4个配方集。而每个配方集中的配方个数，只受存储卡容量的限制。

创建和编辑配方

创建和编辑配方界面允许您创建单条配方并为这些配方分配数值。每一个可编辑的列都表示一个独立的配方。

可以按**New**按钮来创建配方。每个配方会将创建配方集时所指定的缺省值作为初始值。

也可以用另外一种方式创建配方。单击鼠标右键使用关联菜单，您可以复制和粘贴已有的配方。新的列将被插入在当前光标位置的左侧。

根据配方集和配方号，每条新的配方将被给予一个缺省的名字。该缺省名的格式为：DEFx_RCPy。

要创建并编辑配方，执行以下步骤。参见图13-3。

1. 点击**Next**，进入创建和编辑配方窗口。
2. 点击**New**按钮，插入一条配方。
3. 将配方的缺省名改为一个合适的名字。
4. 根据需求改变每个配方数据的值。
5. 点击**OK**。

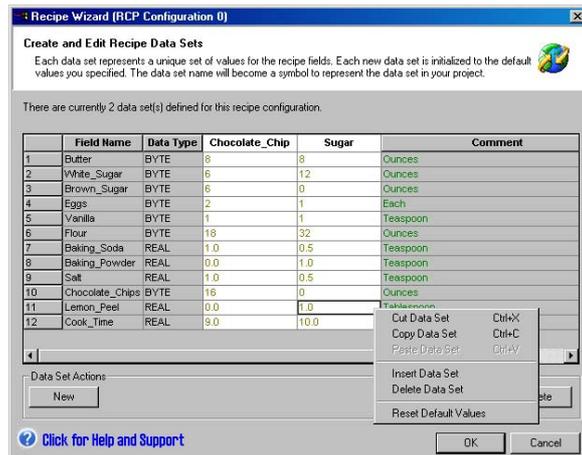


图13-3 创建和编辑配方

分配存储区

分配存储区画面用于指定V存储区的起始地址，从这一起始地址开始存储从存储卡中读取的配方。您可以自己选择V存储区地址，也可以使用配方向导建议的地址，配方向导会推荐您使用正确长度的尚未使用的V存储区。

要为配方分配存储区，执行以下步骤。参见图13-4。

1. 在窗口中输入地址值作为您希望储存配方的V存储区起始地址。
2. 您也可以通过点击Suggest Address按钮，让配方向导为您选择一个正确长度的尚未使用的V存储区。
3. 点击Next。

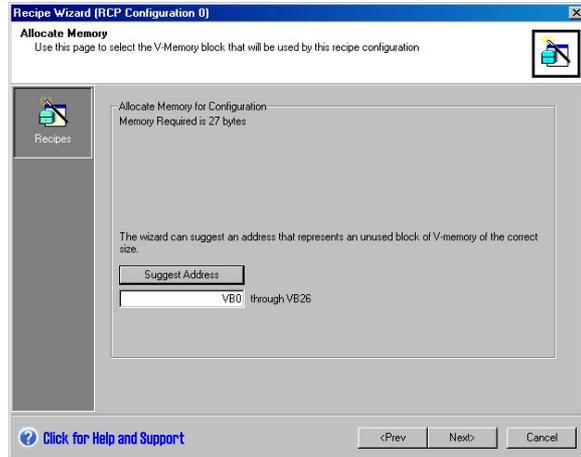


图13-4 分配存储区

项目组件

项目组件画面列出了将要被添加到您项目中的不同组件。参见图13-5。

点击Finish来完成配方向导并将这些组件添加到项目中。

每个配方结构拥有唯一的名字。这些名字会显示在项目树中。配方集名（RCPx）被附加在名字尾部。

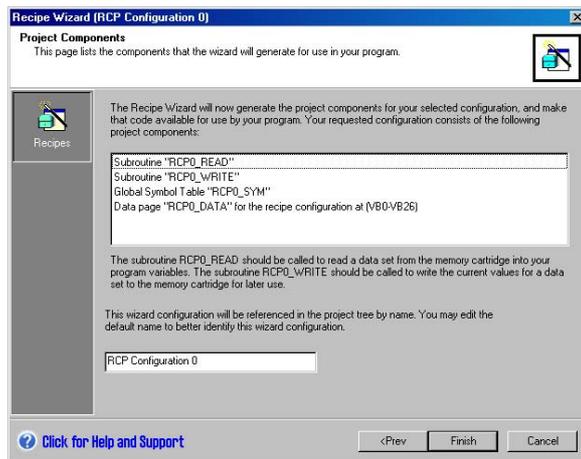


图13-5 项目组件

使用符号表

为每一个配方集创建一个符号表。每张表定义一些常用数值来表示每条配方。可以在RCPx_READ和RCPx_WRITE指令中使用这些符号来表示想要的配方。参见图13-6。

每张表中也为配方中的每个域创建符号名。您可以使用这些符号来访问V存储区中的配方值。

	Symbol	Address	Comment
1	Sugar	1	
2	Chocolate_Chip	0	
3	Cook_Time	VD23	Minutes
4	Lemon_Peel	VD19	Tablespoon
5	Chocolate_Chips	VB18	Ounces
6	Salt	VD14	Teaspoon
7	Baking_Powder	VD10	Teaspoon
8	Baking_Soda	VD6	Teaspoon
9	Flour	VB5	Ounces
10	Vanilla	VB4	Teaspoon
11	Eggs	VB3	Each
12	Brown_Sugar	VB2	Ounces
13	White_Sugar	VB1	Ounces
14	Butter	VB0	Ounces

图13-6 符号表

下载一个带有配方的项目

要下载一个带有配方的项目，执行以下步骤。参见图13-7。

1. 选择**File > Download**。
2. 在对话框中，确保程序块、数据块和配方均被选中。
3. 点击**Download**按钮。

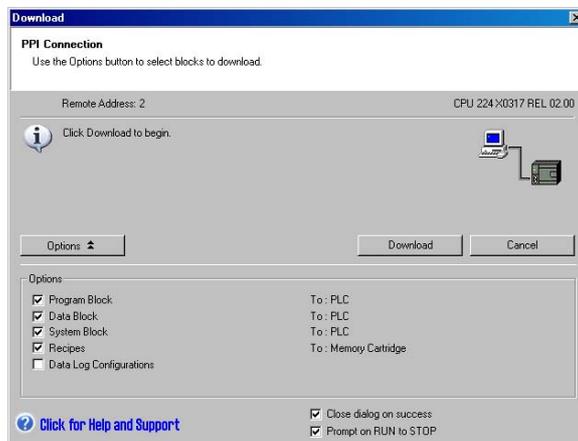


图13-7 下载一个带有配方的项目

编辑已有的配方

要编辑已有的配方，执行以下步骤。参见图13-8。

1. 点击下拉列表选择一个已有的配方。
2. 可以通过**Delete Configuration**按钮来删除一个已有的配方。

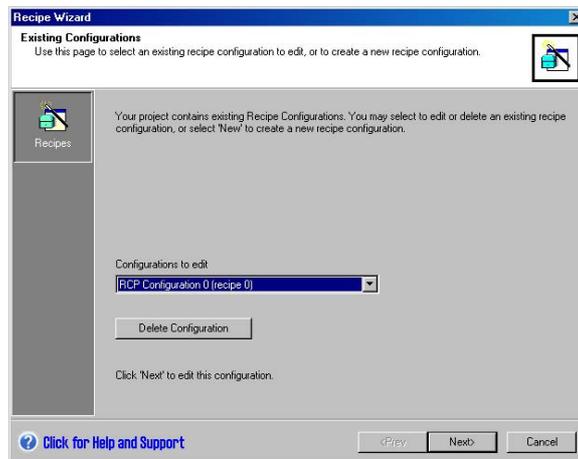


图13-8 编辑已有的配方

由配方向导创建的指令

RCPx_Read子程序

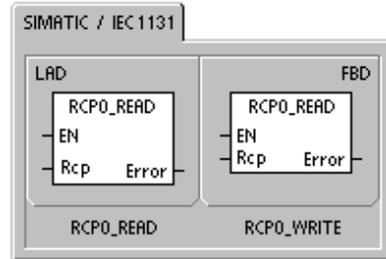
子程序RCPx_READ是由配方向导创建的，它用于将配方从存储卡中读取到V存储区中。

RCPx_READ指令中的x是指包含您想读取的配方的配方集编号。

当EN输入为高电平时，允许指令执行。

Rcp输入端决定了从存储卡中读取哪条配方。

Error输出端返回该指令的执行结果。表13-3中给出了错误代码表。



RCPx_Write子程序

子程序RCPx_WRITE是由配方向导创建的。调用该指令可以使V存储区中的配方内容替代存储卡中的配方。

RCPx_WRITE指令中的x是指包含您想替代的配方的配方集编号。

当EN输入为高电平时，允许指令执行。

Rcp输入决定了替代存储卡中的哪条配方。

Error输出端返回该指令的执行结果。表13-3中给出了错误代码表。

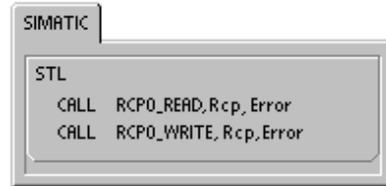


表13-2 配方子程序的有效操作数

输入/输出	数据类型	操作数
Rcp	字	VW, IW, QW, MW, SW, SMW, LW, AC, *VD, *AC, *LD, 常数
Error	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

表13-3 配方子程序的错误代码

错误代码	描述
0	无错误
132	存储卡访问失败



提示

EEPROM存储卡的写操作是有次数限制的，典型值为一百万次。一旦超出限制，EEPROM将失效。

因而请务必确认不是在每个程序周期中都执行RCPx_WRITE指令。否则在很短的时间内，存储卡就会被损坏。

14

使用数据归档

STEP 7-Micro/Win提供数据归档向导，将过程测量数据存入存储卡中。将过程数据移入存储卡可以节省V存储区的地址空间，否则这些数据将储存在V存储区中。

本章内容:

概述	364
使用数据归档向导	365
由数据归档向导创建的指令	369

概述

STEP 7-Micro/WIN软件和S7-200 PLC已经支持数据归档功能。使用这一功能，您可以在程序控制下永久地保存过程数据记录。这些记录可以包含时间日期标签。您最多可以配置4个独立的数据归档。可以用新的数据归档向导来定义数据归档的记录格式。

所有数据归档都存在存储卡中。因此，为了使用数据归档功能，必须要在PLC中插入一块64K或者256K的存储卡。要查阅关于存储卡的更多信息，参见附录A。

您必须使用S7-200资源管理器将数据归档中的内容上载到计算机。

图14-1中显示了数据归档应用的一个实例。

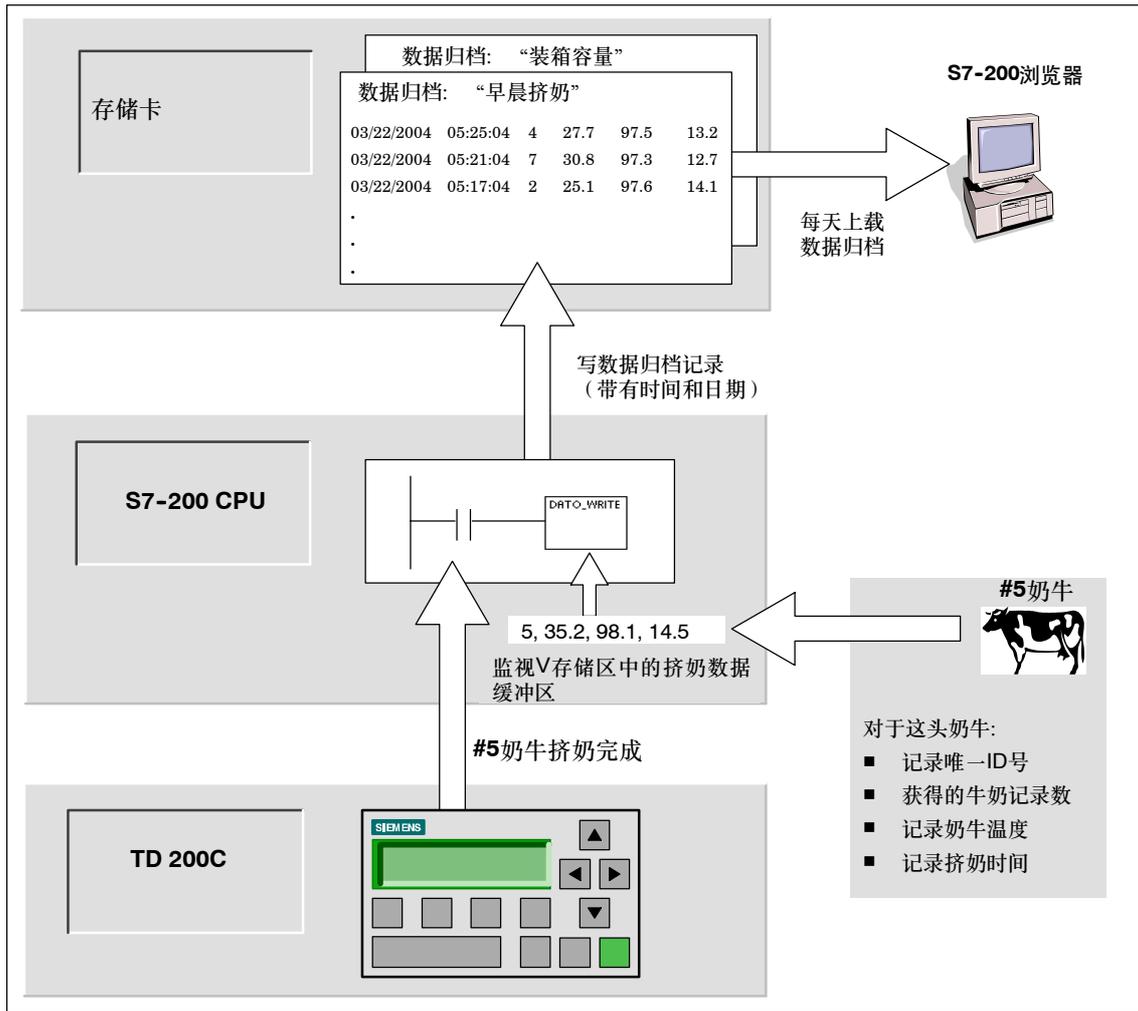


图14-1 数据归档应用实例

数据归档的定义和术语

为了帮助您理解数据归档向导，首先来解释以下定义和术语。

- 数据归档是指通常按照日期时间排序的一组记录。每条记录代表着一些过程事件，过程事件中记录了一套过程数据。由数据归档向导来定义数据的组织结构。
- 数据归档记录是指写入数据归档中的单独的数据行。

使用数据归档向导



数据归档

使用数据归档向导最多可以配置4个数据归档。数据归档向导用于：

- 定义数据归档记录的格式。
- 选择数据归档的可选项，例如：时间标签、日期标签和有上载时清除数据归档等。
- 指定数据归档中储存记录的最大数目。
- 创建用于在数据归档中储存记录的项目代码。

用数据归档向导创建数据归档包含以下步骤：

- 为每个数据归档创建一个符号表。每张表中都包含着与数据归档或域名相同的符号名。这些符号定义了储存当前数据归档所需的V存储区的地址。每张表还包含一个用于标识每个数据归档的符号常数。
- 为每条数据归档记录建立一个数据块标签，从而为每一个数据归档域分配了V存储器地址。用户程序使用这些V存储区地址来采集当前归档数据。
- 生成一个DATx_WRITE子程序。这条指令将指定的数据归档记录从V存储区复制到存储卡中。每执行一次DATx_WRITE指令，将会在存储卡的数据归档中添加一条新的数据记录。

数据归档可选项

您可以为您的数据归档配置如下可选项。参见图14-2。

时间标签

您可以在每条数据归档记录中包含时间标签。如果您选中该项，当用户程序写入一条数据归档记录时，CPU将自动在记录中加入时间标签。

日期标签

您可以在每条数据归档记录中包含日期标签。如果您选中该项，当用户程序写入一条数据归档记录时，CPU将自动在记录中加入日期标签。

清除归档数据

当数据被上载后，您可以清除数据归档中的所有记录。如果您选中该项，每次上载之后，数据归档将被清除。

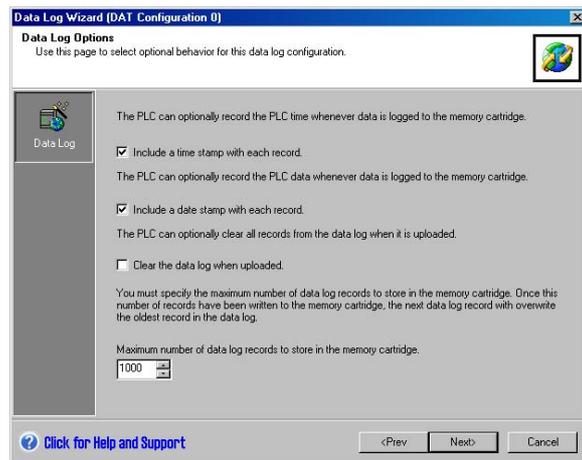


图14-2 数据归档可选项

数据归档是一个环形队列（当归档满时，一条新的记录将代替第一条记录）。您必须指定数据归档中储存记录的最大数目。最大数目的最大值为65535，缺省值为1000。

定义数据归档

您为数据归档指定域，每个域都成为项目中的一个符号。您必须为每个域指定数据类型。一条数据归档记录可以包含4到203个字节的数据。要在数据归档中定义数据域，执行以下步骤。参见图14-3。

1. 点击Field Name单元格来输入域名。名称变为用户程序引用的符号。
2. 点击Data Type单元格，从下拉列表中选择数据类型。
3. 点击Comment单元格来输入注释。
4. 根据需要使用多行来定义一条记录。
5. 点击OK。

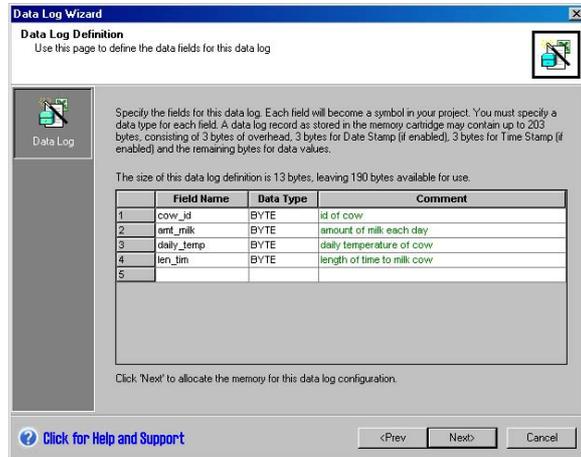


图14-3 定义数据归档记录

编辑已有的数据归档

要编辑已有的数据归档，执行以下步骤：

1. 如图14-4所示，点击下拉列表，选择一个已有的数据归档。
2. 可以通过Delete Configuration按钮来删除一个已有的数据归档。

您可以拥有最多4个不同的数据归档。

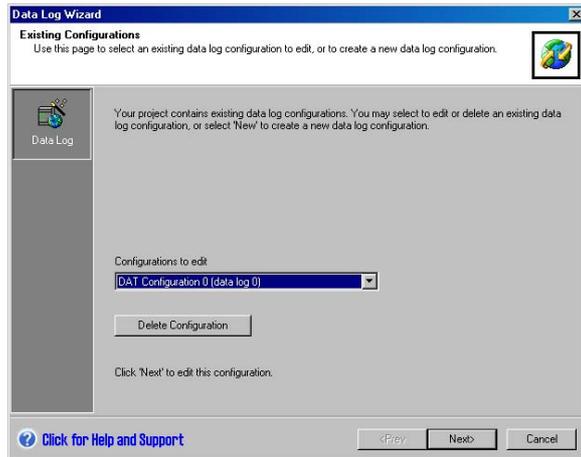


图14-4 编辑已有的数据归档配置

分配存储区

数据归档向导在PLC的V存储区中创建一个块。该块是一个存储区地址，一条数据归档记录在被写入存储卡之前，存储在这个存储区地址中。您可以为您要放置配置的V存储区指定起始地址。您可以自己选择V存储区地址，也可以使用数据归档向导建议的地址。数据归档向导会推荐您使用正确长度的尚未使用的V存储区。块的长度根据您在数据归档向导中的不同选择而有所不同。参见图14-5。

要为数据归档分配存储区，执行以下步骤：

1. 在窗口中输入地址值作为您创建数据归档记录的V存储区地址。
2. 您也可以点击Suggest Address按钮，让数据归档向导为您选择一个正确长度的尚未使用的V存储区。
3. 点击Next。

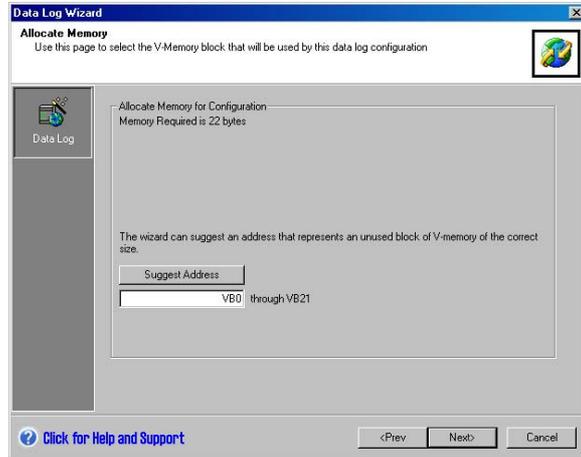


图14-5 分配存储区

项目组件

项目组件画面列出了将要被添加到您项目中的不同组件。参见14-6。

点击Finish来完成数据归档向导，并将这些组件添加到项目中。

每个数据归档结构拥有唯一的名字。这些名字会显示在项目树中。数据归档集名 (DATx) 被附加在名字尾部。

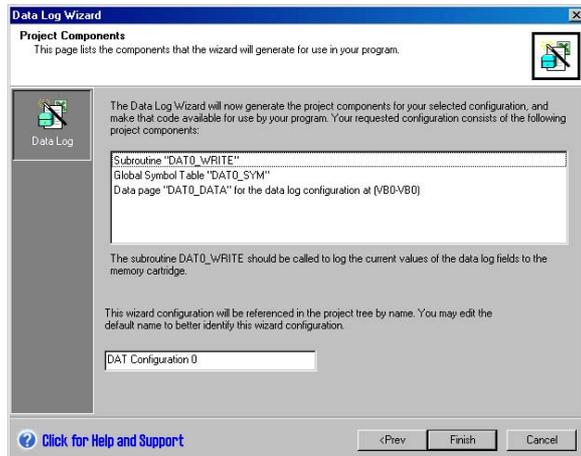


图14-6 项目组件

使用符号表

为每一个数据归档创建一个符号表。每张表定义一些常数来表示每个数据归档。可以在DATx_WRITE指令中使用这些符号。

每张表中也为数据归档中的每个域创建符号名。您可以使用这些符号来访问V存储区中的数据归档数值。

	Symbol	Address	Comment
1	len_time	VB3	length of time to milk cow
2	daily_temp	VB2	cow's daily temperature
3	amt_milk	VB1	amount of milk each day
4	cow_id	VB0	id of cow

图14-7 符号表

下载一个带有数据归档的项目

在使用数据归档之前，您必须将带有数据归档的项目下载到S7-200的CPU中。如果一个项目中带有数据归档，那么在下载窗口中，作为缺省值，Data Log Configuration选项将被选中。



提示

当您下载一个带有数据归档的项目时，当前存在存储卡中的所有数据归档记录将丢失。

要下载一个带有数据归档的项目，执行以下步骤。参见图14-8。

1. 选择**File > Download**。
2. 在对话框中，确保数据归档项被选中。
3. 点击Download按钮。

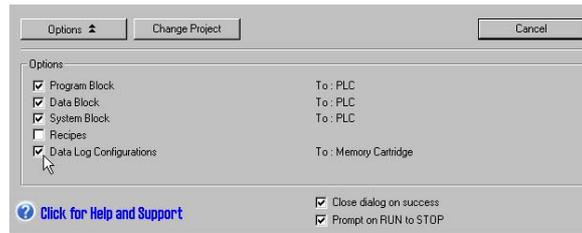


图14-8 下载一个带有数据归档的项目

使用S7-200资源管理器

S7-200资源管理器用于从存储卡中读取数据归档，并将数据归档存储在CSV文件中。

每次读取数据归档，都创建一个新的文件。这个文件存在数据归档目录中。文件名格式为：PLC地址，数据归档名，日期和时间。

您可以选择当数据归档被成功读取时，是否自动启动与CSV扩展名相关联的应用程序。您可以用鼠标右键单击数据归档文件，在关联菜单中作出选择。

数据归档目录将在安装过程中被指定。如果没有安装STET 7，缺省的安装目录为c:\program files\siemens\Microsystems。如果安装了STEP 7，缺省的安装目录为c:\siemens\Microsystems。

要读取数据归档，执行以下步骤：

1. 打开Windows资源管理器，将看到My S7-200 Network文件夹。
2. 选择My S7-200 Network文件夹。
3. 选择正确的S7-200 PLC文件夹。

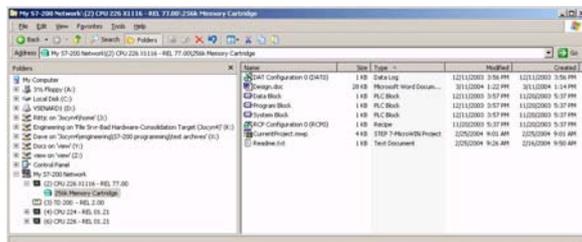


图14-9 使用S7-200资源管理器

4. 选择存储卡文件夹。
5. 找到正确的数据归档文件。这些文件的名称为DAT Configuration x (DATx)。
6. 用鼠标右键调出上下文关联菜单，选择Upload。

由数据归档向导创建的指令

数据归档向导会在您的项目中添加一条指令子程序。

DATx_WRITE子程序

子程序DATx_WRITE用于将数据归档域的当前值写入存储卡。DATxWRITE将归档数据中的一条记录写入存储卡。子程序的调用如图所示。

当指令访问存储卡失败时，会返回错误代码132。

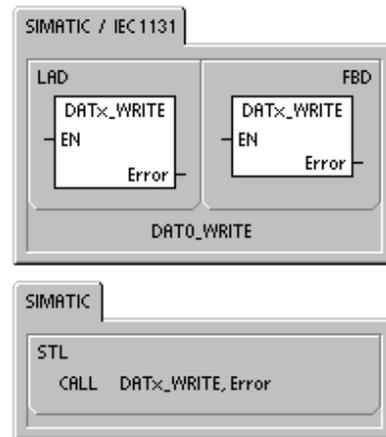


表14-1 DATx_WRITE子程序的参数

输入/输出	数据类型	操作数
Error	字节	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD



提示

EEPROM存储卡的写操作是有次数限制的，典型值为一百万次。一旦超出限制，EEPROM将失效。

因而请务必确认不是在每个程序周期中都执行DATx_WRITE指令。否则在很短的时间内，存储卡就会被损坏。

15

PID自整定和PID整定控制面板

S7-200PLC已经支持PID自整定功能，STEP 7-Micro/WIN中也添加了PID整定控制面板。这就大大增强了S7-200PLC的功能，并且使这一功能的使用变得更加容易。

您可以使用操作员面板中的用户程序或者PID整定控制面板来启动自整定功能。在同一时间，不仅仅只有一个PID回路可以进行自整定，如果需要的话，所有8个PID回路可以同时进行自整定。PID自整定算法向您推荐增益值、积分时间值和微分时间值。您也可以为您的调节回路选择快速响应、中速响应、慢速响应或者极慢速响应等调节类型。

用PID整定控制面板，您可以启动自整定过程，取消自整定过程和图表中监视结果。控制面板会显示所有可能发生的错误和警告信息。它也允许您将自整定后得到的增益值、积分时间值和微分时间值应用到实际控制中去。

本章内容:

理解PID的自整定	372
扩展回路表	372
先决条件	375
自滞后和自偏移	375
自整定序列	376
例外情况	377
关于过程变量超限的说明（结果代码3）	377
PID整定控制面板	378

理解PID自整定

引言

S7-200中使用的自整定算法是基于K.J.Åström和T.Hägglund在1984年提出的延时反馈算法。经过这二十年，延时反馈算法已被应用于工业控制的各个领域。

延时反馈的概念是指在一个稳定的控制过程中产生一个微小但持续的扰动。过程变量中扰动的周期和振幅，将最终决定控制过程的频率和增益。然后利用最终的增益和频率值，PID自整定向您推荐增益值、积分时间值和微分时间值。

所推荐的值与您为您的控制过程所选择的响应速度相关。您可以选择快速响应、中速响应、慢速响应或者极慢速响应。根据您的控制过程，一个快速响应会产生超调，它符合不完全衰减整定条件。一个中速响应会使控制过程濒临超调的边缘，它符合临界衰减整定条件。一个慢速响应不会导致超调，它符合强衰减整定条件。一个极慢速响应不会导致超调，它符合超强衰减整定条件。

PID自整定除了推荐指定值以外，还可以自动确定滞后值和过程变量峰值偏移。这些参数用于减少当PID自整定设置持续振荡限幅时过程噪声所产生的影响。

PID自整定适用于双向调节、反向调节、P调节、PI调节、PD调节和PID调节等各种调节回路。

PID自整定的目的在于为您的过程控制回路提供一套最优化的整定参数。使用这些整定值可以使您达到极佳的控制效果，真正优化您的控制过程。

扩展回路表

S7-200中的PID指令涉及到一张回路表，回路表中包含了回路参数。该表原来只有36个字节，现在有了PID自整定功能，表的长度增加到80个字节。表15-1和表15-2展示了扩展后的回路表。

如果您使用PID整定控制面板，您可以通过控制面板操纵PID回路表中的所有参数。如果您需要一个操作员面板来提供自整定功能，您必须通过编程的方式来实现操作员和PID回路表之间的信息交换，从而启动和监视自整定过程，最终将整定值应用到实际控制中去。

表15-1 回路表

偏移地址	参数名	格式	类型	描述
0	过程变量 (PV_n)	实数	输入	过程变量, 必须在0.0~1.0之间
4	设定值 (SP_n)	实数	输入	设定值, 必须在0.0~1.0之间
8	输出值 (M_n)	实数	输入 / 输出	输出值, 必须在0.0~1.0之间
12	增益 (K_C)	实数	输入	增益是比例常数, 可正可负
16	采样时间 (T_S)	实数	输入	单位是秒, 必须是正数
20	积分时间 (T_I)	实数	输入	单位是分钟, 必须是正数
24	微分时间 (T_D)	实数	输入	单位是分钟, 必须是正数
28	积分项前项 (MX)	实数	输入 / 输出	积分项前项, 必须在0.0~1.0之间
32	过程变量前值 (PV_{n-1})	实数	输入 / 输出	最近一次PID运算的过程变量值
36	PID回路表ID	ASCII码	常数	'PIDA' (PID扩展表, 版本A) ASCII码常数
40	AT控制 (ACNTL)	字节	输入	见表15-2
41	AT状态 (ASTAT)	字节	输出	见表15-2
42	AT结果 (ARES)	字节	输入 / 输出	见表15-2
43	AT配置 (ACNFG)	字节	输入	见表15-2
44	偏移 (DEV)	实数	输入	归一化以后的过程变量振幅最大值 (范围: 0.025~0.25)
48	滞后 (HYS)	实数	输入	归一化以后的过程变量滞后值, 用于确定零相交 (范围: 0.005 ~0.1) 当DEV与HYS的比率小于4时, 自整定过程中会发出警告。
52	初始输出阶跃幅度 (STEP)	实数	输入	归一化以后的输出值阶跃变化幅度, 用于减小过程变量的振动。 (范围: 0.05~0.4)
56	看门狗时间 (WDOG)	实数	输入	两次零相交之间允许的最大时间间隔, 单位是秒。 (范围: 60~7200)
60	推荐增益 (AT_ K_C)	实数	输出	自整定过程推荐的增益值
64	推荐积分时间 (AT_ T_I)	实数	输出	自整定过程推荐的积分时间值
68	推荐微分时间 (AT_ T_D)	实数	输出	自整定过程推荐的微分时间值
72	实际输出阶跃幅度 (ASTEP)	实数	输出	自整定过程确定的归一化以后的输出阶跃幅度。
76	实际滞后 (AHYS)	实数	输出	自整定过程确定的归一化以后的过程变量滞后值。

表15-2 控制参数和状态参数的具体描述

参数名	描述																								
AT控制 (ACNTL) 输入-字节	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">MSB</td> <td colspan="6"></td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td colspan="6"></td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">EN</td> </tr> </table> <p>EN位置1启动自整定; EN位清0取消自整定。</p>	MSB							LSB	7							0	0	0	0	0	0	0	0	EN
MSB							LSB																		
7							0																		
0	0	0	0	0	0	0	EN																		
AT状态 (ASTAT) 输出-字节	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">MSB</td> <td colspan="6"></td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td colspan="6"></td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">W0</td> <td style="text-align: center;">W1</td> <td style="text-align: center;">W2</td> <td style="text-align: center;">0</td> <td style="text-align: center;">AH</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">IP</td> </tr> </table> <p>W0 - 警告: 偏移设定没有超过滞后设定的4倍。 W1 - 警告: 不协调的过程偏移可能导致输出阶跃值的不正确调节。 W2 - 警告: 实际平均偏移没有超过滞后设定的4倍。 AH - 计算自滞后: 0 - 计算停止 1 - 计算运行 IP - 自整定运行 0 - 自整定停止 1 - 自整定运行</p> <p>每次自整定功能启动, PLC都清除警告位, 置位自整定运行位。直到自整定完成, PLC清除自整定运行位。</p>	MSB							LSB	7							0	W0	W1	W2	0	AH	0	0	IP
MSB							LSB																		
7							0																		
W0	W1	W2	0	AH	0	0	IP																		
AT结果 (ARES) 输入/输出 - 字节	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">MSB</td> <td colspan="6"></td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td colspan="6"></td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">D</td> <td colspan="6" style="text-align: center;">结果代码:</td> <td></td> </tr> </table> <p>D - 完成位: 0 - 自整定未完成 1 - 自整定完成 在自整定启动之前, 该位必须清0。</p> <p>结果代码: 00 - 正常完成 (推荐的整定值可以使用) 01 - 用户取消 02 - 取消, 零相交看门狗超时 03 - 取消, 过程变量超范围 04 - 取消, 超出最大滞后值 05 - 取消, 检测到非法配置值 06 - 取消, 检测到数字错误 07 - 取消, PID指令未使能 (回路处于手动模式) 08 - 取消, 自整定只适用于P调节, PI调节, PD调节或PID调节 09 到7F - 保留</p>	MSB							LSB	7							0	D	结果代码:						
MSB							LSB																		
7							0																		
D	结果代码:																								
AT配置 (ACNFG) 输入 - 字节	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">MSB</td> <td colspan="6"></td> <td style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">7</td> <td colspan="6"></td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">R1</td> <td style="text-align: center;">R0</td> <td style="text-align: center;">DS</td> <td style="text-align: center;">HS</td> </tr> </table> <p>R1 R0 动态响应 0 0 快速响应 0 1 中速响应 1 0 慢速响应 1 1 极慢速响应</p> <p>DS - 偏移设定: 0 - 使用回路表中的偏移值 1 - 自动计算偏移值</p> <p>HS - 滞后设定: 0 - 使用回路表中的滞后值 1 - 自动计算滞后值</p>	MSB							LSB	7							0	0	0	0	0	R1	R0	DS	HS
MSB							LSB																		
7							0																		
0	0	0	0	R1	R0	DS	HS																		

先决条件

您要进行自整定的回路必须处于自动模式。回路的输出必须由PID指令来控制。如果回路处于手动模式，自整定会失败。

在启动自整定之前，您的控制过程应该达到一种稳定状态。这种稳定状态是指过程变量已经达到设定值（或者对于P调节来说，过程变量与设定值之间的差值恒定）并且输出不会不规则地变化。

理想状态下，当自整定启动时，回路的输出值应该在控制范围中心附近。自整定过程在回路的输出中加入一些小的阶跃变化，使得控制过程产生振荡。如果回路的输出值没有处于控制范围中心附近，自整定的这种阶跃变化会导致输出超限。

如果这种情况发生，会使自整定发生错误，当然也会使推荐值并非最优化。

自滞后和自偏移

滞后参数给出一个相对于设定值的正负偏移量，过程变量在此偏移量范围内时，不会导致控制器改变输出值。这个值用于减小过程变量中噪声的影响，从而更精确地计算出过程自然振动频率。

如果您选用自动计算滞后值，PID自整定会生成一个滞后运算队列。该队列包含一段时间内的过程变量采样值，然后根据采样结果计算出标准偏移。

为了得到具有统计意义的采样数据，至少要有100个采样值。如果回路的采样周期为200 ms，100个采样值就需要20秒时间。回路采样周期更长会需要更多的时间。即使您使用的回路采样周期小于20ms，从而使得采样100次用不了20秒时间，滞后运算队列仍然需要至少20秒采样时间。

当得到足够的采样值以后，就可以算出标准偏移。决定滞后值需要两次标准偏移的计算值。计算后得到的滞后值被写入回路表中的实际滞后（AHYS）域中。



提示

在自滞后计算过程中，正常的PID运算会停止。因此，在启动自整定之前，控制过程应处于稳定状态。这样可以使滞后值的计算收到好的效果，同时也可以保证在自滞后运算过程中，控制过程不会失控。

偏移参数是指希望得到的过程变量相对于设定值的峰-峰值幅度。如果您选择自动计算该值，它将是滞后值的4.5倍。在自整定过程中，会适当地调节输出，使控制过程中的振动在这一范围内。

自整定序列

自整定序列在得到滞后值和偏移值之后开始执行。当初始输出阶跃实际应用到回路的输出时，整定过程就开始了。

输出值的这一变化会导致过程变量值产生相应的变化。当输出的变化使过程变量远离设定值以致于超出滞后区范围时，自整定将检测到一个零相交事件。在每次零相交事件发生时，自整定将反方向改变输出。

自整定继续采样过程变量值，等待下一次零相交事件。要完成整个序列，需要12次零相交事件。过程变量的峰-峰值和零相交事件的产生速度都与控制过程的动态特性直接相关。

在自整定过程一开始，会适当地调节输出阶跃值，促使过程变量的峰-峰值更接近想要得到的偏移值。一旦有调节产生，新的输出阶跃值将被写入回路表的实际输出阶跃幅度（ASTEP）域中。

如果两次零相交时间的间隔超过了零相交看门狗的间隔时间，自整定序列将被终止。零相交看门狗的间隔时间缺省值为两小时。

图15-1给出了在自整定序列过程中输出和过程变量的变化情况。PID整定控制面板用于启动和监视整定序列。

注意自整定是如何改变输出，来使控制过程（用过程变量值表示）经受小幅振动的。过程变量的振动幅度和频率代表着控制过程增益和自然频率。

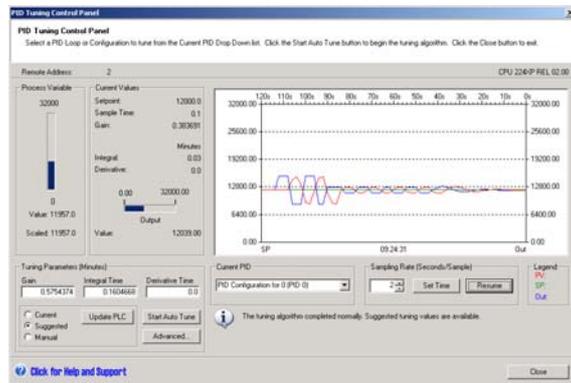


图15-1 自整定序列

根据在自整定过程中采集到的关于控制过程频率和增益的相关信息，能够计算出最终的增益和频率值。根据这些值又可以进一步计算出推荐的增益值、积分时间值和微分时间值。



提示

您的回路类型决定了自整定计算出的整定值。例如一个PI调节回路，自整定会计算出增益值和积分时间值，而推荐的微分时间值为0.0。

一旦自整定序列完成，回路的输出会恢复到初始值。在下一周期，正常的PID运算将被执行。

例外情况

在自整定执行过程中会产生三种警告。在回路表的ASTAT域中有三位用于表示这三种警告，并且一旦被置位，将会一直保持到下一次自整定序列启动。

- 当偏移设定没有超过滞后设定的4倍时产生警告0。该项检测在自滞后已经计算出实际滞后值之后执行。
- 在自整定过程最开始的2.5个循环周期内，如果两次峰值误差超出8倍，产生警告1。
- 如果测量到的平均峰值误差没有超过滞后值的4倍，产生警告2。

除此之外，还有几种可能性。表15-3中列出了可能导致每种错误的情况和描述。

表15-3 自整定执行过程中的错误情况。

结果代码（在ARES中）	情况
01 用户取消	在自整定执行过程中，EN被复位
02 因零相交看门狗超时而取消	超过零相交看门狗时间间隔半个周期
03 因过程变量超范围而取消	过程变量超范围 <ul style="list-style-type: none"> • 在自滞后序列期间 • 在4次零相交之内出现两次超范围 • 在4次零相交之后
04 因滞后超过最大值而取消	用户定义的或者自动计算的滞后值超过最大值
05 因非法的配置值而取消	在以下范围内检测错误: <ul style="list-style-type: none"> • 初始回路输出值 < 0.0 或者 > 1.0 • 用户定义的偏移值 < = 滞后值 或者 最大值 • 初始输出阶跃 < = 0.0 或者 > 最大值 • 零相交看门狗时间 < 最小值 • 回路表中的采样时间值为负。
06 因数字错误而取消	非法浮点数或者除以0。
07 因PID指令未使能（回路处于手动模式）而取消	当自整定正在执行或者被请求执行时，PID指令未使能。
08 自整定只适用于P调节、PI调节、PD调节、或者PID调节	回路类型不是P调节、PI调节、PD调节或者PID调节中的任何一种。

关于过程变量超限的说明（结果代码3）

如果过程变量值在0.0到1.0之间，我们认为它在范围内。

如果在自滞后序列过程中检测到过程变量超限，自整定会立即被取消并报错。

如果在第4次零相交之前检测到过程变量超限，输出阶跃值会被减半，自整定过程重新启动。此时，如果在第一次零相交之后再次出现过程变量超限，自整定会被取消并报错。

如果在第4次零相交之后检测到过程变量超限，自整定会立即被取消并报错。

PID整定控制面板

STEP 7-Micro/WIN软件中包含了一个PID整定控制面板，它使您能够以图形的方式来监视您的PID回路。另外，控制面板还可用于启动自整定序列，取消自整定序列，还可以将推荐整定值或者您设定的整定值应用到实际控制中去。

要使用控制面板，您必须在线连接一个S7-200PLC，并且该PLC中已经存在至少一个PID回路。为了显示控制面板对PID回路的操作，PLC必须处于运行状态。图15-2给出了控制面板的缺省画面。

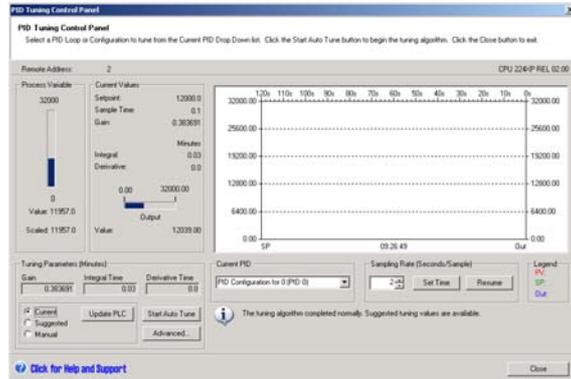


图15-2 PID整定控制面板

控制面板在屏幕左上角的位置上显示所连接的PLC站地址。在屏幕的右上角显示PLC的类型和版本号。在PLC站地址的下方是表示过程变量值的棒图，棒图下面是过程变量的标定值和非标定值。过程变量的右侧是当前值区域。

在当前值区域里，显示了设定值、采样时间、增益、积分时间和微分时间。输出值用一横向的棒图来表示，其数值显示在棒图的下方。当前值区域的右侧是图形显示区。

图形显示区中用不同的颜色显示了过程变量、设定值和输出值相对于时间的函数。过程变量和设定值共同使用左侧的纵轴，输出值使用右侧的纵轴。

屏幕的左下方是整定参数区。在这一区域中显示增益、积分时间和微分时间。单选按钮表示出这些参数是当前值、推荐值还是手动值。您可以通过点击单选按钮作出选择。要想改变整定参数，选择手动值。

您可以通过点击Update PLC按钮来将增益、积分时间和微分时间值传入被监视的PID回路中。您也可以使用Start Auto Tune按钮启动自整定序列。一旦自整定序列启动，Start Auto Tune按钮会变为Stop Auto Tune按钮。

在图形显示区下方是当前PID回路选择区，您可以在下拉菜单中选择您希望在控制面板中监视的PID回路。

在采样速率区域中，您可以在1到480秒之间选择图形显示的采样时间间隔。您可以编辑采样速率，用Set Time按钮使设定生效。图形显示区的时间坐标会随您的设置自动改变到最佳显示状态。

您可以点击Pause按钮来冻结画面，也可以用Resume按钮来重新启动数据采样。在图形区域内点鼠标右键选择Clear，可以清除图形。

在图形选项区的右侧是一个图例，标明了过程变量值、设定值和输出值的不同颜色。

在当前PID回路选择区域正下方的区域内，显示与当前操作相关的信息。

在整定参数区中的Advanced ...按钮，使您可以配置更多的自整定参数。图15-3中显示了高级配置画面。

在高级配置画面中，您可以选中复选框，让自整定来自动计算滞后值和偏移值。为了最小化在自整定过程中对控制过程的干扰，您也可以自己输入这些值。

在Other Options区域中，您可以指定初始输出阶跃幅值和零相交看门狗超时时间。

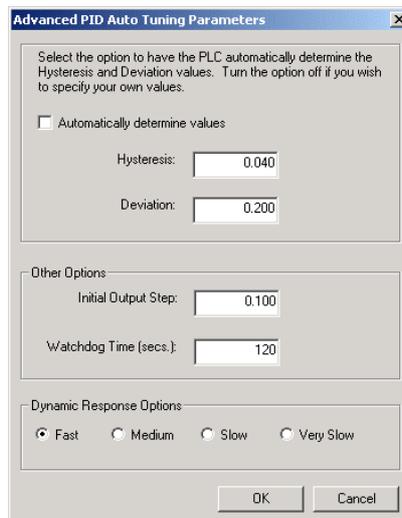


图15-3 高级参数

在动态响应选择区域中，点击单选按钮，您可以为您的控制回路选择响应类型。根据您的控制过程，一个快速响应会产生超调，它符合不完全衰减整定条件。一个中速响应会使控制过程濒临超调的边缘，它符合临界衰减整定条件。一个慢速响应不会导致超调，它符合强衰减整定条件。一个极慢速响应不会导致超调，它符合超强衰减整定条件。

一旦您完成了选择，可以点击OK按钮返回PID整定控制面板的主画面。

在完成了自整定序列并将推荐值传入PLC之后，您可以使用控制面板来监视当设定值有一个阶跃变化时，控制回路的响应如何。图15-4显示出当设定值有一个阶跃变化时，（从12000变为14000）时，回路的响应情况。其中整定参数为原始值（自整定以前的值）。

请注意在使用原始整定参数时，控制过程的超调和长时间振荡的现象。

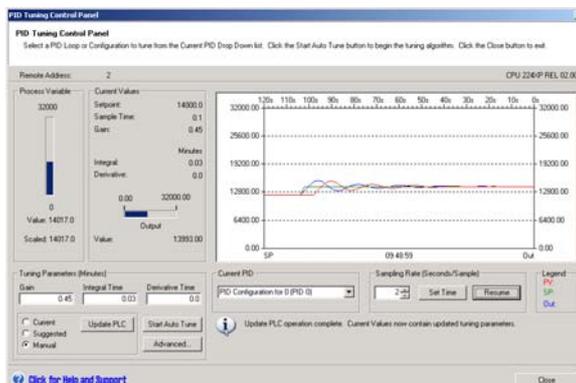


图15-4 对设定值变化的响应

图15-5中显示的回路响应也是设定值发生相同的变化（从12000变为14000）时产生的。此时，一个快速响应的自整定过程已经完成，整定值也已应用到实际控制中。请注意此时没有超调现象，只有一点点小的振荡。如果您希望牺牲一部分响应速度来消除这些振荡，您可以选择中速响应或者慢速响应类型，然后重新运行自整定过程。

一旦您有了一个好的起点，您就可以用控制面板来进一步优化您的参数。您可以改变设定值来观察回路响应。用这种方法，您可以微调您的控制参数，使您的应用达到最佳效果。

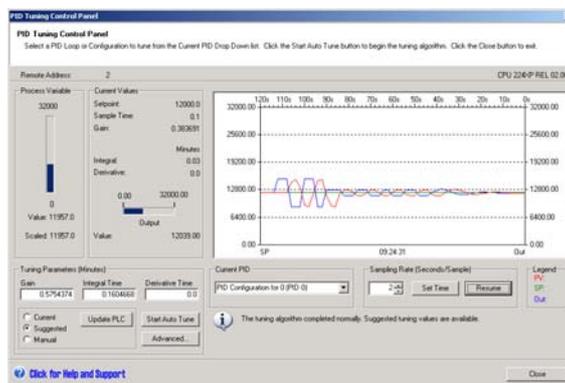
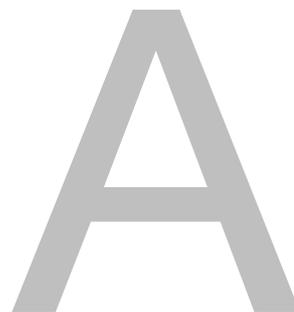


图15-5 自整定过程后的响应

技术规范



本章概述

通用技术规范	382
CPU规范	385
数字扩展模块规范	394
模拟量扩展模块规范	400
热电偶和RTD（热电阻）扩展模块规范	411
EM277 PROFIBUS-DP模板规范	423
EM241MODEM模块规范	435
EM253位控模块规范	437
（CP243-1）以太网模块规范	443
（CP243-1IT）因特网模块规范	445
（CP243-2）AS接口模块规范	448
可选卡件	450
扩展电缆	451
RS-232/PPI多主站电缆和USB/PPI多主站电缆	452
输入仿真器	456

通用技术规范

认证标准

下面列出的国内和国际标准用于确定合适的性能规范和S7-200系列产品的测试。表A-1列出了符合这些标准的规范。

- European Community (CE) Low Voltage Directive 73/23/EEC
EN61131-2: 可编程控制器 - 设备要求
- European Community (CE) EMC Directive 89/336/EEC
电磁辐射标准
EN61000-6-3: 民用、商务和轻工业
EN61000-6-4: 工业环境
防电磁辐射标准
EN61000-6-2: 工业环境
- Underwriters Laboratories, Inc: UL 508 Listed (工业控制设置), 注册号E75310
- Canadian Standards Association: CSA C22.2 Number 142 (过程控制设备)
- Factory Mutual Research: Class Number 3600, Class Number 3611, FM Class I, Division 2, Groups A, B, C, & D Hazardous Locations, T4A和Class I, Zone 2, IIC, T4.
- European Community (ATEX) Atmospheres Explosibles Directive 94/9/EC
EN60079-0常规要求
EN50020原有安全度 ‘i’
EN60079-15保护类型 ‘n’

ATEX Directive 94/9/EC证书在本手册出版时还没有完成。对于更新的信息, 请联系当地的西门子代理商。



提示

SIMATIC S7-200系列符合CSA标准。

cULus标志表明S7-200已经由Underwriters Laboratories (UL) 检验和证明符合标准UL 508和CSA 22.2 No. 142。

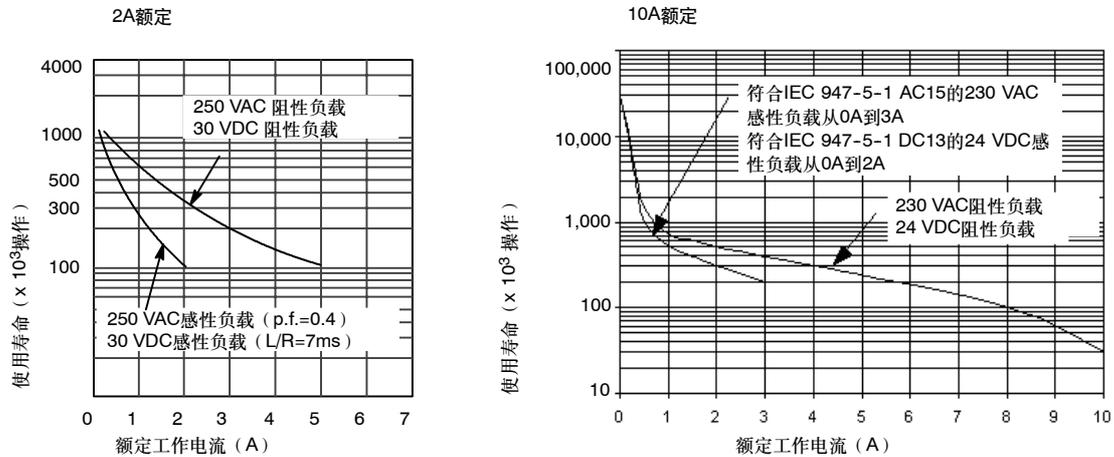
船用许可

S7-200产品定期提交给和指定市场和应用相关的特殊代理许可。此表指出了S7-200产品已经通过的代理和证书号。此手册中的大部分S7-200产品已经通过了这些特殊的代理许可。如果您需要与准确的许可证部件号码的最新列表相关的附加信息, 请联系当地的西门子代理商。

代理	证书号
Lloyds Register of Shipping (LRS)	99 / 20018 (E1)
American Bureau of Shipping (ABS)	01-HG20020-PDA
Germanischer Lloyd (GL)	12 045 - 98 HH
Det Norske Veritas (DNV)	A-8862
Bureau Veritas (BV)	09051 / B0BV
Nippon Kaiji Kyokai (NK)	A-534
Polski Rejestr	TE/1246/883241/99

继电器电气使用寿命

继电器供应商提供的典型性能数据如图A-1所示。实际的性能可能随特定应用而有所变化。适合负载的外部保护电路将增长触点的使用寿命。



图A-1 继电器电气使用寿命

技术规范

所有的S7-200 CPU和扩展模块符合表A-1中列出的电气规范。

注释

当一个机械触点接通S7-200 CPU或任意扩展模块的供电时，它发送一个大约50毫秒的信号到数字输出，您需要考虑这一点，尤其是您使用能够响应短脉冲的设备时。

表A-1 技术规范

环境条件—运输和存贮	
EN 60068-2-2, Test Bb, 干热 EN 60068-2-1, Test Ab, 低温	-40°C ~ +70°C
EN 60068-2-30, Test Db, 湿热	25°C ~ 55°C, 95%湿度
EN 60068-2-14, Test Na, 温度震动	-40°C ~ +70°C 停留时间3个小时, 2个周期
EN 60068-2-31, 倒下	100mm, 4次倒下, 未包装
EN 60068-2-32, 自由落下	1m, 5次, 运输包装
环境条件—工作	
控制柜温度范围 (单元下部25mm进入的空气)	0°C ~ 55°C水平安放0°C ~ 45°C垂直安放 95%非冷凝湿度
大气压力	1080 ~ 795hPa (对应海拔高度-1000 ~ 2000m)
污染程度	S0 ₂ : <0.5ppm; H ₂ S: <0.1ppm; RH<60%无冷藏
EN 60068-2-14, Test Nb, 温度改变	5°C ~ 55°C, 3°C/分钟
EN 60068-2-27机械震动	15G, 11ms脉冲, 每轴向(3轴)震动6次
EN 60068-2-6正弦波振动	面板安装: 峰-峰值0.30mm, 频率10~57Hz; 2G频率57~150Hz 导轨安装: 0.15mm, 频率10~57Hz; 1G 57 Hz~150Hz; 每轴向 10次振动, 1倍频程/分。
EN 60529, IP20机械保护	防止高压指状物接触设备。需要外部保护, 以防止灰尘、污物、水和直径小于12.5mm的异物造成破坏。

表A-1 技术规范, continued

电磁兼容性 - 抗干扰性能符合EN61000-6-2 ¹	
EN 61000-4-2静电放电	对所有的面和通讯接口8kV空气放电 对暴露的传导面4kV接触放电
EN 61000-4-3辐射电磁场	10 V/m, 80-1000 MHz和1.4 ~ 2.0 GHz, 80% AM, 1kHz
EN 61000-4-4瞬间冲击	2 kV, 5 kHz, 带AC和DC系统电源耦合网络 2 kV, 5 kHz, 带I/O耦合接线夹 1 kV, 5 kHz, 带通讯耦合接线夹
EN 61000-4-5浪涌防护	电源: 2kV非对称, 1kV对称 I/O 1 kV对称 (24 VDC电路要求外部浪涌保护)
EN 61000-4-6传导干扰	0.15 ~ 80 MHz, 10 V RMS, 80% AM, 1kHz
EN 61000-4-11电压波动, 短暂干扰和电压变化	对8.3ms, 83ms, 833ms和4167ms, >95%减少
VDE 0160非周期过电压	对85 VAC线, 90°相角, 允许峰值390V, 1.3ms脉冲 对180VAC线, 90°相角, 允许峰值750V, 1.3ms脉冲
电磁兼容性 - 感应和辐射 按照EN 61000-6-3 ² 和EN 61000-6-4	
EN 55011, Class A, Group 1, 传导 ¹ 0.15 MHz~0.5 MHz 0.5 MHz~5 MHz 5 MHz~30 MHz	< 79 dB (μV) 准峰值; < 66 dB (μV) 均值 < 73 dB (μV) 准峰值; < 60 dB (μV) 均值 < 73 dB (μV) 准峰值; < 60 dB (μV) 均值
EN 55011, Class A, Group 1, 辐射 ¹ 30 MHz ~ 230 MHz 230 MHz ~ 1 GHz	40 dB (μV/m) 准峰值; 10米测量 47 dB (μV/m) 准峰值; 10米测量
EN 55011, Class B, Group 1, 传导 ² 0.15~0.5 MHz 0.5 MHz~5 MHz 5 MHz~30 MHz	< 66 dB (μV) 准峰值按对数频率减少到56 dB (μV); < 56 dB (μV) 均值按对数频率减少到46dB (μV) < 56 dB (μV) 准峰值; < 46 dB (μV) 均值 < 60 dB (μV) 准峰值; < 50 dB (μV) 均值
EN 55011, Class B, Group 1, 辐射 ² 30 MHz ~ 230 kHz 230 MHz ~ 1 GHz	30 dB (μV/m) 准峰值; 10米测量 37 dB (μV/m) 准峰值; 10米测量
高压绝缘测试	
24 V/5 V 标称电路 115/230 V电路接地 115/230 V电路接到115/230 V 电路 230 V 电路接到24 V/5 V 电路 115 V 电路接到24 V/5 V 电路	500 VAC (可选隔离带) 1,500 VAC 1,500 VAC 1,500 VAC 1,500 VAC

¹ S7-200的接地直接连接到安装金属架上, 而单元必须安装在接地金属架上。电缆沿金属架布线。

² 设备必须安装在接地的金属壳中。AC输入电源必须接有一个EPCOS B84115-E-A30滤波器或等效设备。滤波器和S7-200间的导线不能超过25cm。24VDC供电线和传感器供电线必须屏蔽。

CPU规范

表A-2 CPU订货号

订货号	CPU模板	CPU供电 (常量)	数字量输入	数字量输出	通讯口	模拟量输入	模拟量输出	可拆卸连接
6ES7 211-0AA23-0XB0	CPU221	24 VDC	6 x 24 VDC	4 x 24 VDC	1	否	否	否
6ES7 211-0BA23-0XB0	CPU221	120至240 VAC	6 x 24 VDC	4 x 继电器	1	否	否	否
6ES7 212-1AB23-0XB0	CPU222	24 VDC	8 x 24 VDC	6 x 24 VDC	1	否	否	否
6ES7 212-1BB23-0XB0	CPU222	120至240 VAC	8 x 24 VDC	6 x 继电器	1	否	否	否
6ES7 214-1AD23-0XB0	CPU224	24 VDC	14 x 24 VDC	10 x 24 VDC	1	否	否	是
6ES7 214-1BD23-0XB0	CPU224	120至240 VAC	14 x 24 VDC	10 x 继电器	1	否	否	是
6ES7 214-2AD23-0XB0	CPU224XP	24 VDC	14 x 24 VDC	10 x 24 VDC	2	2	1	是
6ES7 214-2BD23-0XB0	CPU224XP	120至240 VAC	14 x 24 VDC	10 x 继电器	2	2	1	是
6ES7 216-2AD23-0XB0	CPU226	24 VDC	24 x 24 VDC	16 x 24 VDC	2	否	否	是
6ES7 216-2BD23-0XB0	CPU226	120至240 VAC	24 x 24 VDC	16 x 继电器	2	否	否	是

表A-3 CPU常规规范

订货号	模块名称及描述	尺寸 (毫米) (W x H x D)	重量	功耗	电流供应	
					+5 VDC	+24 VDC ¹
6ES7 211-0AA23-0XB0	CPU221 DC/DC/DC 6输入/4输出	90 x 80 x 62	270 g	3 W	0 mA	180 mA
6ES7 211-0BA23-0XB0	CPU221 AC/DC/ Relay 6输入/4继电器	90 x 80 x 62	310 g	6 W	0 mA	180 mA
6ES7 212-1AB23-0XB0	CPU222 DC/DC/DC 8输入/6输出	90 x 80 x 62	270 g	5 W	340 mA	180 mA
6ES7 212-1BB23-0XB0	CPU222 AC/DC/ Relay 8输入/6继电器	90 x 80 x 62	310 g	7 W	340 mA	180 mA
6ES7 214-1AD23-0XB0	CPU224 DC/DC/DC 14输入/10输出	120.5 x 80 x 62	360 g	7 W	660 mA	280 mA
6ES7 214-1BD23-0XB0	CPU224 AC/DC/ Relay 14输入/10继电器	120.5 x 80 x 62	410 g	10 W	660 mA	280 mA
6ES7 214-2AD23-0XB0	CPU224 XP DC/DC/DC 14输入/10输出	140 x 80 x 62	390 g	8 W	660 mA	280 mA
6ES7 214-2BD23-0XB0	CPU224 XP AC/DC/继电器14输入/10继电器	140 x 80 x 62	440 g	11 W	660 mA	280 mA
6ES7 216-2AD23-0XB0	CPU226 DC/DC/DC 24输入/16输出	196 x 80 x 62	550 g	11 W	1000 mA	400 mA
6ES7 216-2BD23-0XB0	CPU226 AC/DC/ 继电器 24输入/16继电器	196 x 80 x 62	660 g	17 W	1000 mA	400 mA

¹ 这是内部继电器线圈电源和24 VDC通讯口电源需求解决以后，可以使用的24 VDC传感器电源。

表A-4 CPU规范

	CPU221	CPU222	CPU224	CPU224XP	CPU226
存储器					
用户程序大小 运行模式下编辑 非运行模式下编辑	4096字节 4096字节		8192字节 12288字节	12288字节 16384字节	16384字节 24576字节
用户数据	2048字节		8192字节	10240字节	10240字节
装备（超级电容） （可选电池）	50小时/典型值（40°C时最少8小时）		100小时/典型值（40°C时最少70小时）	100小时/典型值（40°C时最少70小时）	
I/O					
数字量I/O	6输入/4输出	8输入/6输出	14输入/10输出	14输入/10输出	24输入/16输出
模拟量I/O	无			2输入/1输出	无
数字I/O映像区	256（128入/128出）				
模拟I/O映像区	无	32（16入/16出）	64（32入/32出）		
允许最大的扩展模块	无	2个模块 ¹	7个模块 ¹		
允许的最大的智能模块	无	2个模块 ¹	7个模块 ¹		
脉冲捕捉输入	6	8	14		24
高速计数 单相 两相	4个计数器 4个30kHz 2个20kHz		6个计数器 6个30kHz 4个20kHz	总共6个计数器 4个30kHz 2个200kHz 3个20kHz 1个100kHz	6个计数器 6个30kHz 4个20kHz
脉冲输出	2个20kHz（仅限于DC输出）			2个100kHz （仅限于DC输出）	2个20kHz （仅限于DC输出）
常规					
定时器	256定时器；4个定时器（1ms）；16定时器（10ms）；236定时器（100ms）				
计数器	256（由超级电容或电池备份）				
内部存储器位 掉电保存	256（由超级电容或电池备份） 112（存储在EEPROM）				
时间中断	2个1ms分辨率				
边沿中断	4个上升沿和/或4个下降沿				
模拟电位器	1个8位分辨率		2个8位分辨率		
布尔量运算执行速度	0.22μs每条指令				
实时时钟	可选卡件		内置		
卡件选项	存储器、电池和实时时钟		存储卡和电池卡		
集成的通讯功能					
端口（受限电源）	一个RS-485口			两个RS-485口	
PPI，DP/T波特率	9.6、19.2、187.5K波特				
自由口波特率	1.2K-15.2K波特				
每段最大电缆长度	使用隔离的中继器：187.5K波特可达1000米，38.4K波特可达1200米 未使用隔离中继器：50米				
最大站点数	每段32个站，每个网络126个站				
最大主站数	32				
点到点（PPI主站模式）	是（NETR/NETW）				
MPI连接	共4个，2个保留（1个给PG，1个给OP）				

¹ 您必须对电源消耗作出预算，从而确定S7-200 CPU能为您的配置提供多少功率（或电流）。如果超过CPU电源预算值，那么，您可能无法将全部模块都连接上去。关于CPU和扩展模块要求，可参看附录A，关于电源消耗估算，可参见附录B。

表A-5 CPU电源规范

DC			AC	
输入电源				
输入电压	20.4-28.8VDC		85-264VAC (47-63 Hz)	
输入电流	仅CPU, 24 VDC	最大负载24 VDC	仅CPU	最大负载
CPU221	80 mA	450 mA	30/15 mA 120/240 VAC	120/240 VAC时120/60 mA
CPU222	85 mA	500 mA	40/20 mA 120/240 VAC	120/240 VAC时140/70 mA
CPU224	110 mA	700 mA	60/30 mA 120/240 VAC	120/240 VAC时200/100 mA
CPU224XP	120mA	900 mA	70/35 mA 120/240 VAC	120/240 VAC时220/100 mA
CPU226	150 mA	1050 mA	80/40 mA 120/240 VAC	120/240 VAC时320/160 mA
冲击电流	28.8 VDC时12 A		264 VAC时20 A	
隔离 (现场与逻辑)	非隔离		1500 VAC	
保持时间 (掉电)	10 ms, 24 VDC		20/80 ms, 120/240 VAC	
保险 (不可替换)	3A, 250 V慢速熔断		2 A, 250 V慢速熔断	
24 VDC传感器电源				
传感器电压 (受限电源)	L+减5V		20.4-28.8 VDC	
电流限定	1.5 A 峰值, 热量限制无破坏性 (对于额定负载见表A-3)。			
纹波噪声	来自输入电源		小于1V峰-峰值	
隔离 (传感器与逻辑)	非隔离			

表A-6 CPU数字量输入规范

常规	24 VDC 输入 (CPU221、CPU222、CPU224、CPU226)		24 VDC 输入 (CPU224XP)	
类型	漏型/源型 (IEC类型1漏型)		漏型/源型 (IEC类型1漏型, I0.3 ~ I0.5除外)	
额定电压	24VDC, 4mA典型值		24VDC, 4mA典型值	
最大持续允许电压	30 VDC			
浪涌电压	35 VDC, 0.5s			
逻辑1 (最小)	15 VDC, 2.5mA		15 VDC, 2.5 mA (I0.0 ~ I0.2和I0.6 ~ I1.5) 4VDC, 8 mA (I0.3 ~ I0.5)	
逻辑0 (最大)	5 VDC 1 mA		5 VDC, 1 mA (I0.0 ~ I0.2和I0.6 ~ I1.5) 1VDC, 1 mA (I0.3 ~ I0.5)	
输入延迟	可选择的 (0.2 ~ 12.8ms)			
连接2线接近开关传感器 (Bero) 允许的漏电流 (最大)	1mA			
隔离 (现场与逻辑) 光电隔离 隔离组	是 500 VAC, 1分钟 见接线图			
高速计数器 (HSC) 输入速率 HSC输入 所有HSC 所有HSC HC4和HC5只在CPU224XP上	逻辑1电平 15 ~ 30 VDC 15 ~ 26 VDC > 4 VDC	单相 20 kHz 30 kHz 200 kHz	两相 10 kHz 20 kHz 100 kHz	
同时接通的输入	所有		所有 只有CPU224XPAC/DC/继电器: 所有的都是55°C, 带最大26VDC的DC输入。 所有的都是50°C, 带最大30VDC的DC输入。	
电缆长度 (最大) 屏蔽 未屏蔽	普通输入500米, HSC输入50米 ¹ 普通输入300米			

¹ 对于HSC输入, 推荐使用屏蔽双绞线。

表A-7 CPU数字量输出规范

常规	24VDC输出 (CPU221、CPU 222、CPU224、CPU226)	24VDC输出 (CPU224XP)	继电器输出
类型	固态MOSFET ¹ (信号源)		干触点
额定电压	24 VDC	24 VDC	24 VDC或250 VAC
电压范围	20.4-28.8 VDC	5 - 28.8 VDC (Q0.0 - Q0.4) 20.4 - 28.8VDC (Q0.5 - Q1.1)	5至30 VDC或5至250 VAC
浪涌电流 (最大)	8A, 100ms		5A, 4s@10%占空比
逻辑1 (最小)	20VDC, 最大电流	最大电流时, L+减0.4V	-
逻辑0 (最大)	0.1VDC, 10KΩ负载		-
每点额定电流 (最大)	0.75A		2.0A
每个公共端的额定电流 (最大)	6A	3.75A	10A
漏电流 (最大)	10μA		-
灯负载 (最大)	5W		30WDC; 200WAC ^{3, 4}
感性嵌位电压	L+ 减48VDC, 1W功耗		-
接通电阻 (触点)	0.3Ω典型 (最大0.6Ω,)		0.2Ω (新的时候的最大值)
隔离	光电隔离 (现场到逻辑) - 逻辑到触点 - 电阻 (逻辑到触点) - 隔离组 见接线图		- 1500VAC, 1分钟 100MΩ 见接线图
延时 (最大) 从关断到接通 (μs) 从接通到关断 (μs) 切换	2μs (Q0.0和Q0.1), 15μs (其它) 10μs (Q0.0和Q0.1), 130μs (其它) -	0.5μs (Q0.0和Q0.1), 15μs (其它) 1.5μs (Q0.0和Q0.1), 130μs (其它) -	- - 10ms
脉冲频率 (最大)	20kHz ² (Q0.0和Q0.1)		1Hz
机械寿命周期	-		10,000,000 (无负载)
触点寿命	-		100,000 (额定负载)
同时接通的输出	所有的都在55°C (水平), 所有的都在45°C (垂直)		
两个输出并联	是的, 只有输出在同一个组内		否
电缆长度 (最大) 屏蔽 非屏蔽	500m 150m		

¹ 当一个机械触点接通S7-200 CPU或任意扩展模块的供电时, 它发送一个大约50毫秒的“信号到数字输出, 您需要考虑这一点, 尤其是您使用触响应短脉冲的设备时。

² 依据于您的脉冲接收器和电缆, 附加的外部负载电阻 (至少是额定电流的10%) 可以改善脉冲信号的质量并提高噪音防护能力。

³ 带灯负载的继电器使用寿命将降低75%, 除非采取措施将接通浪涌降低到输出的浪涌电流额定值以下。

⁴ 灯负载的瓦特额定值是用于额定电压的。依据正被切换的电压, 按比例降低瓦特额定值 (例如120VAC-100W)。

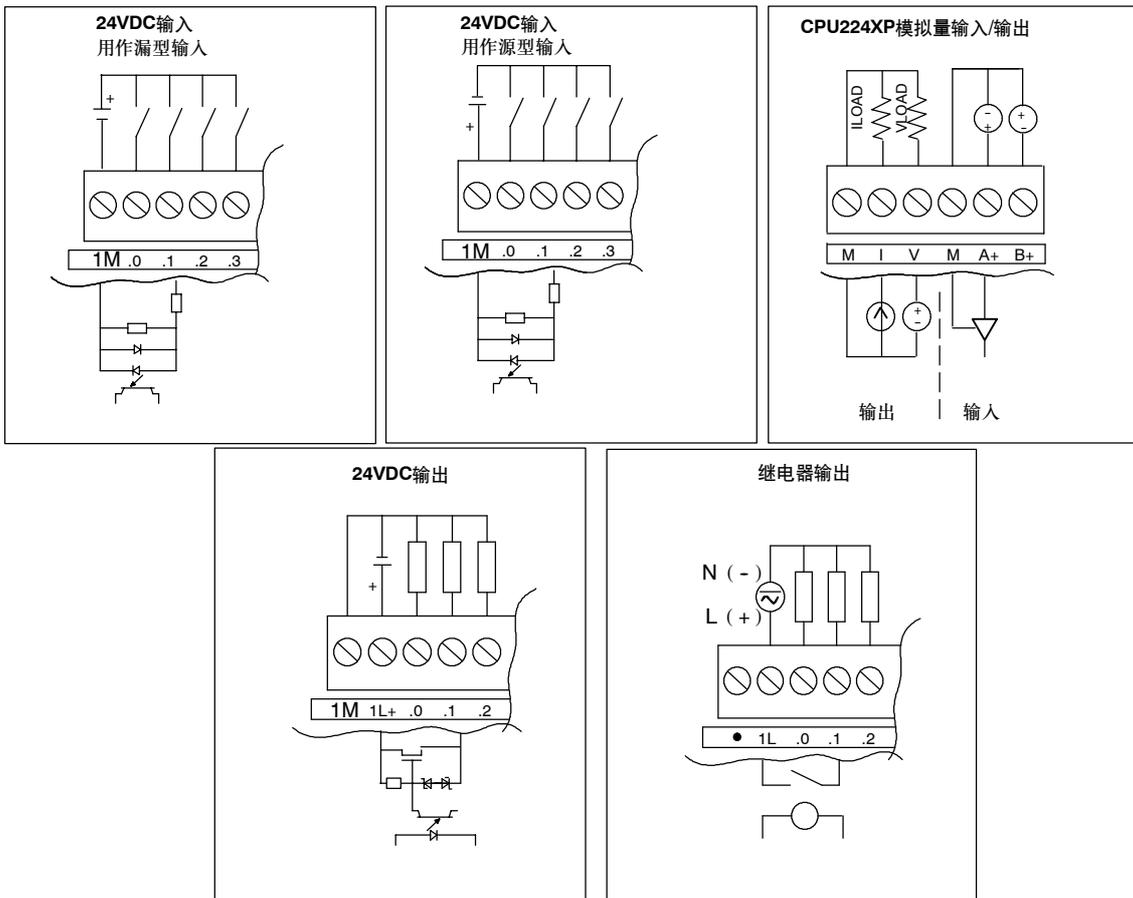
表A-8 CPU224XP模拟量输入规范

常规	模拟量输入 (CPU224XP)
输入数量	2点
模拟量输入字节	单端
电压范围	±10V
数据字格式, 满量程范围	-32,000至+32,000
DC输入阻抗	>100KΩ
最大输入电压	30VDC
分辨率	11位, 加1符号位
LSB值	4.88mV
隔离	无
精度 最差情况, 0°至55°C 典型, 25°C	±2.5%满量程 ±1.0%满量程
重复性	±0.05%满量程
模拟到数字转换时间	125msec
转换字节	SigmaDelta
步响应	最大250ms
噪音抑制	典型为-20dB@50Hz

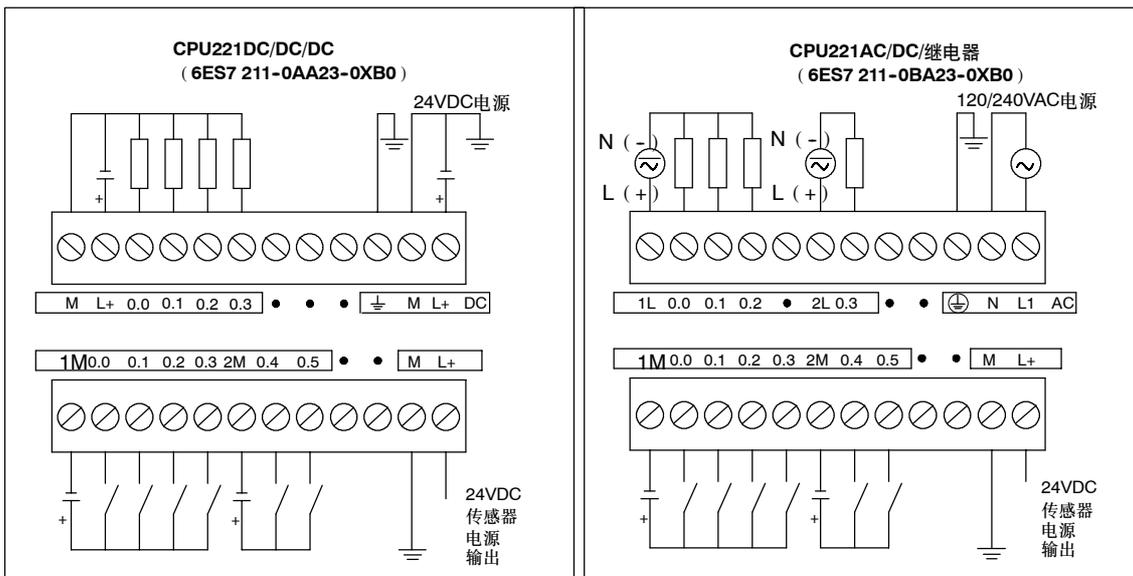
表A-9 CPU224XP模拟量输出规范

常规	模拟量输出 (CPU224XP)
输出数量	1点
信号范围 电压 电流	0 ~ 10V (有限电源) 0 ~ 20mA (有限电源)
数据字格式, 满量程范围	0至+32767
数据字格式, 满量程范围	0至+32000
分辨率, 满量程	12位
LSB值 电压 电流	2.44mV 4.88μA
隔离	无
精度 最差情况, 0°至55°C 电压输出 电流输出 典型, 25°C 电压输出 电流输出	±2%满量程 ±3%满量程 ±1%满量程 ±1%满量程
建立时间 电压输出 电流输出	<50μS <100μS
最大输出驱动 电压输出 电流输出	≥5000Ω最小 ≤500Ω最大

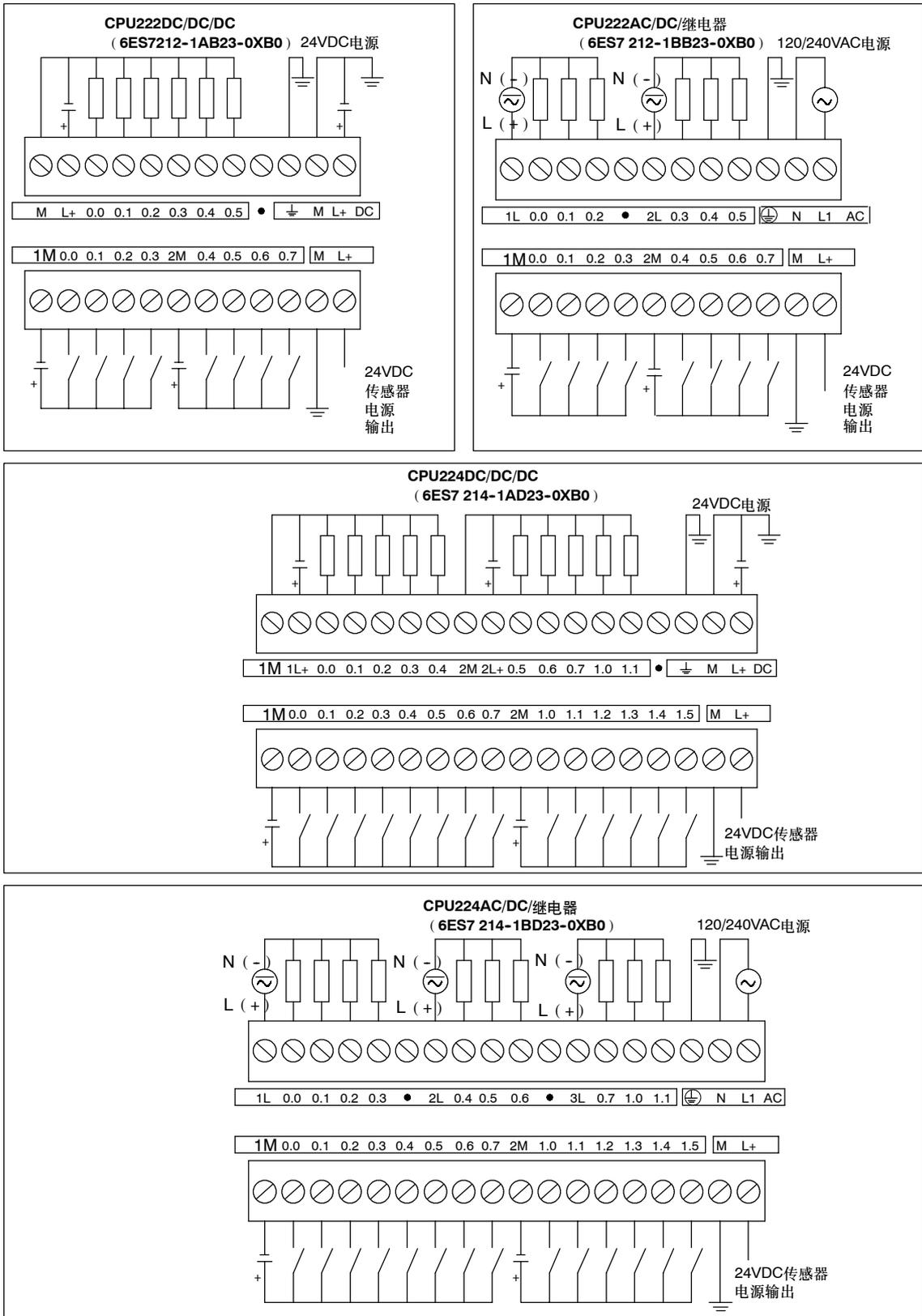
接线图



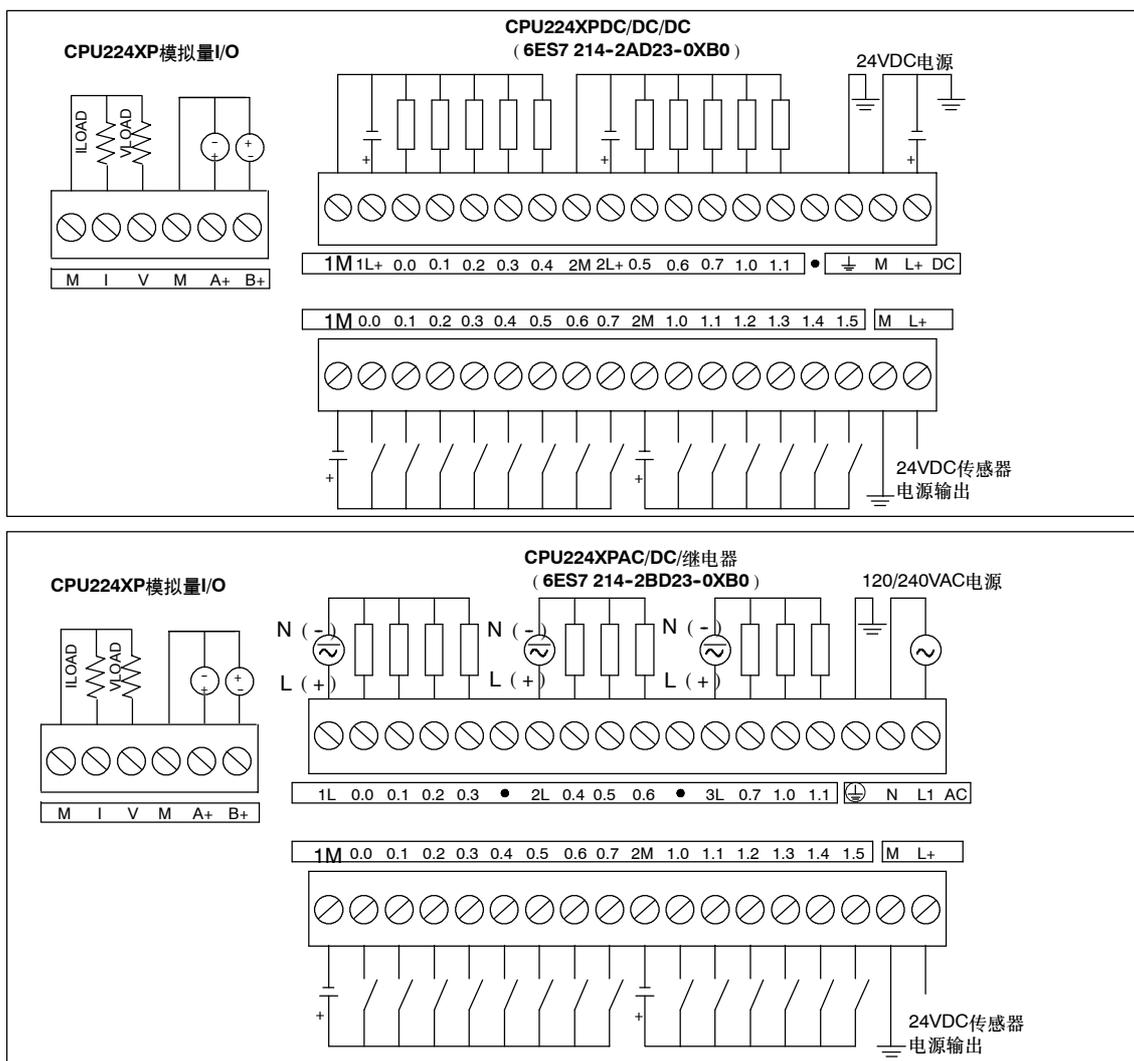
图A-2 CPU输入和输出



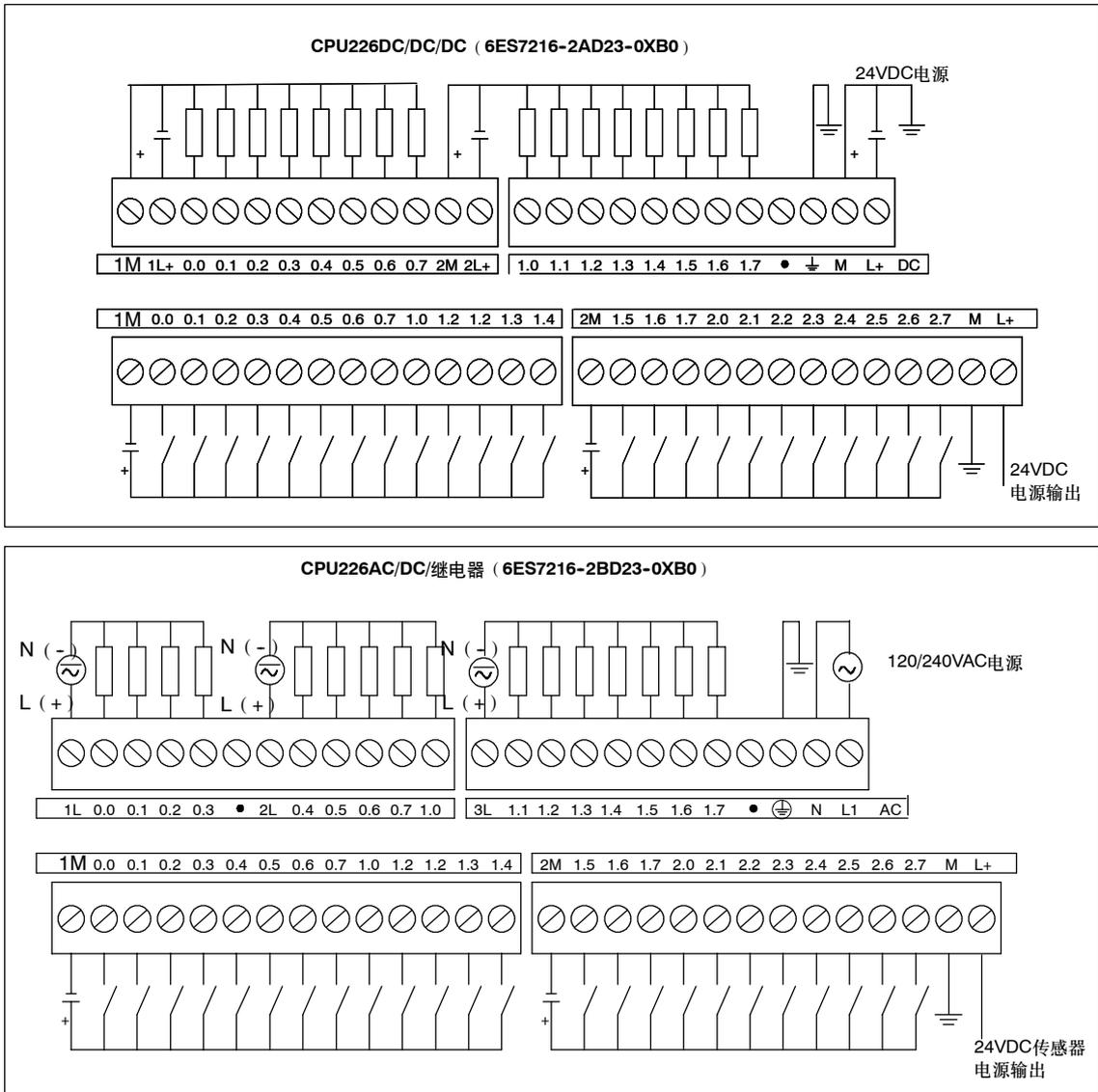
图A-3 CPU221接线图



图A-4 CPU 222和CPU 224接线图



图A-5 CPU 224XP接线图



图A-6 CPU 226接线图

表A-10 S7-200通讯口的引脚分配 (有限电源)

连接器	针号	信号	口0/口1
	1	屏蔽	机壳地
	2	24V返回	逻辑地
	3	RS-485信号B	RS-485信号B
	4	请求发送	RTS (TTL)
	5	5V返回	逻辑地
	6	+5 V	+ 5V, 100Ω串联电阻
	7	+24V	+24V
	8	RS-485信号A	RS-485信号A
	9	未用	选择10位协议 (输入)
	连接器外壳	屏蔽	机壳地

数字扩展模块规范

表A-11 数字量扩展模块订货号

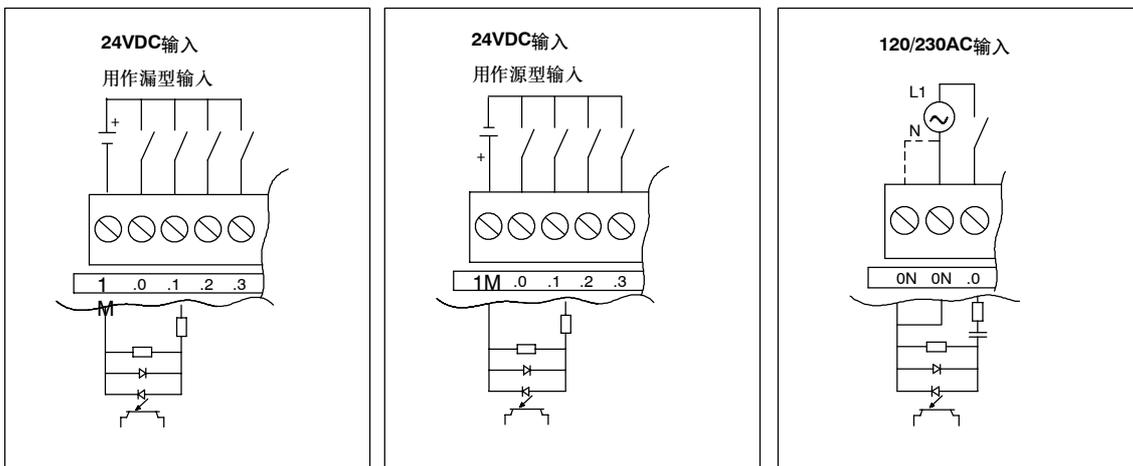
订货号	扩展模块	数字量输入	数字量输出	可拆卸连接
6ES7 221-1BF22-0XA0	EM221 数字输入 8x24VDC	8x24VDC	-	是
6ES7 221-1EF22-0XA0	EM221 数字输入 8x120/230VAC	8x120/230VAC	-	是
6ES7 221-1BH22-0XA0	EM221 数字输入 16x24VDC	16x24VDC	-	是
6ES7 222-1BD22-0XA0	EM222 数字输出 4x24VDC-5A	-	4x24VDC-5A	是
6ES7 222-1HD22-0XA0	EM222 数字输出 4x继电器-10A	-	4x继电器-10A	是
6ES7 222-1BF22-0XA0	EM222 数字输出 8x24VDC	-	8x24VDC-0.75A	是
6ES7 222-1HF22-0XA0	EM222 数字输出 8x继电器	-	8x继电器-10A	是
6ES7 222-1EF22-0XA0	EM222 数字输出 8x120/230VAC	-	8x120/230VAC	是
6ES7 223-1BF22-0XA0	EM223 24VDC 数字组合 4输入/4输出	4x24VDC	4x24VDC-0.75A	是
6ES7 223-1HF22-0XA0	EM223 24VDC 数字组合 4输入/4继电器输出	4x24VDC	4x继电器-2A	是
6ES7 223-1BH22-0AX0	EM223 24VDC 数字组合 8输入/8输出	8x24VDC	8x24VDC-0.75A	是
6ES7 223-1PH22-0XA0	EM223 24VDC 数字组合 8输入/8继电器输出	8x24VDC	8x继电器-2A	是
6ES7 223-1BL22-0XA0	EM223 24VDC 数字组合 16输入/16输出	16x24VDC	16x24VDC-0.75A	是
6ES7 223-1PL22-0XA0	EM223 24VDC 数字组合 16输入/16继电器输出	16x24VDC	16x继电器-2A	是

表A-12 数字量扩展模块常规范

订货号	模块名称及描述	尺寸 (mm) (W x H x D)	重量	功耗	VDC要求	
					+5VDC	+24VDC
6ES7 221-1BF22-0XA0	EM221 DI 8x24VDC	46x80x62	150g	2W	30mA	接通: 4mA/输入
6ES7 221-1EF22-0XA0	EM221 DI 8x120/230VAC	71.2x80x62	160g	3W	30mA	-
6ES7 221-1BH22-0XA0	EM221 DI 16x24VDC	71.2x80x62	160g	3W	70mA	接通: 4mA/输入
6ES7 222-1BD22-0XA0	EM222 DO 4x24VDC-5A	46x80x62	120g	3W	40mA	-
6ES7 222-1HD22-0XA0	EM222 DO 4x继电器-10A	46x80x62	150g	4W	30mA	接通: 20mA/输出
6ES7 222-1BF22-0XA0	EM222 DO 8x24VDC	46x80x62	150g	2W	50mA	-
6ES7 222-1HF22-0XA0	EM222 DO 8x继电器	46x80x62	170g	2W	40mA	接通: 9mA/输出
6ES7 222-1EF22-0XA0	EM222 DO 8x120/230VAC	71.2x80x62	165g	4W	110mA	-
6ES7 223-1BF22-0XA0	EM223 24VDC 4入/4出	46x80x62	160g	2W	40mA	接通: 4mA/输入
6ES7 223-1HF22-0XA0	EM223 24VDC 4入/4继电器	46x80x62	170g	2W	40mA	接通: 9mA/输出, 4mA/输入
6ES7 223-1BH22-0AX0	EM223 24VDC 8入/8出	71.2x80x62	200g	3W	80mA	-
6ES7 223-1PH22-0XA0	EM223 24VDC 8入/8继电器	71.2x80x62	300g	3W	80mA	接通: 9mA/输出, 4mA/输入
6ES7 223-1BL22-0XA0	EM223 24VDC 16入/16出	137.3x80x620	360g	6W	160mA	-
6ES7 223-1PL22-0XA0	EM223 24VDC 16入/16继电器	137.3x80x620	400g	6W	150mA	接通: 9mA/输出, 4mA/输入

表A-13 数字量扩展模块输入规范

常规	24VDC输入	120/230VAC输入 (47至63Hz)
类型	漏型/源型 (IEC类型1漏型)	IEC类型1
额定电压	24VDC, 4mA	120VAC, 6mA或230VAC, 9mA (通常)
最大持续允许电压	30VDC	264VAC
浪涌电压 (最大)	35VDC, 0.5s	-
逻辑1 (最小)	15VDC, 2.5mA	79VAC, 2.5mA
逻辑0 (最大)	5VDC, 1mA	20VAC或1mAAC
输入延时 (最大)	4.5ms	15ms
连接2线接近开关传感器 (Bero) 允许的漏电流 (最大)	1mA	1mAAC
隔离 光电隔离 (现场到逻辑) 隔离组	500VAC, 1分钟 见接线图	1500VAC1分钟 1点
同时接通的输入	所有水平安装时低于55°C, 垂直安装时低于45°C	
电缆长度 (最大) 屏蔽 非屏蔽	500m 300m	500m 300m

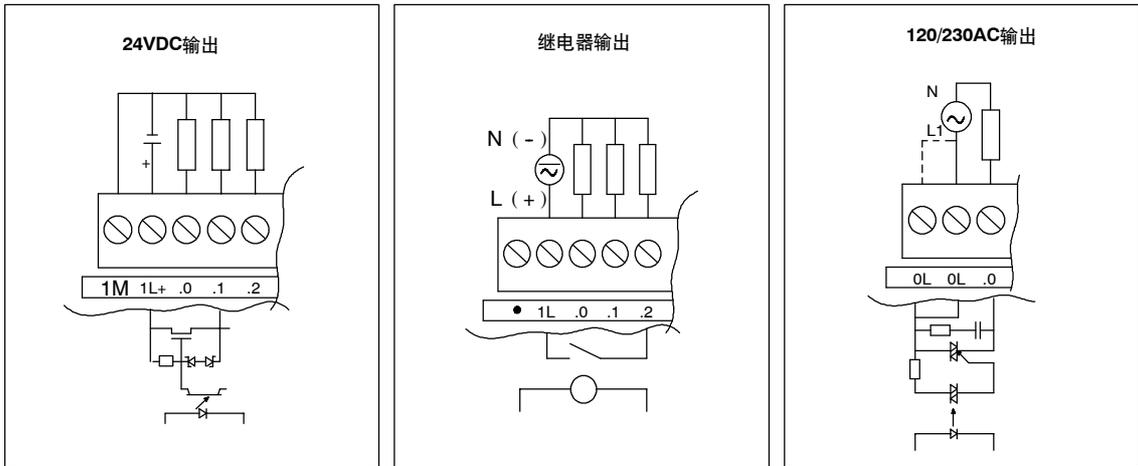


图A-7 S7-200数字量扩展模块输入

表A-14 数字量扩展模块输出规范

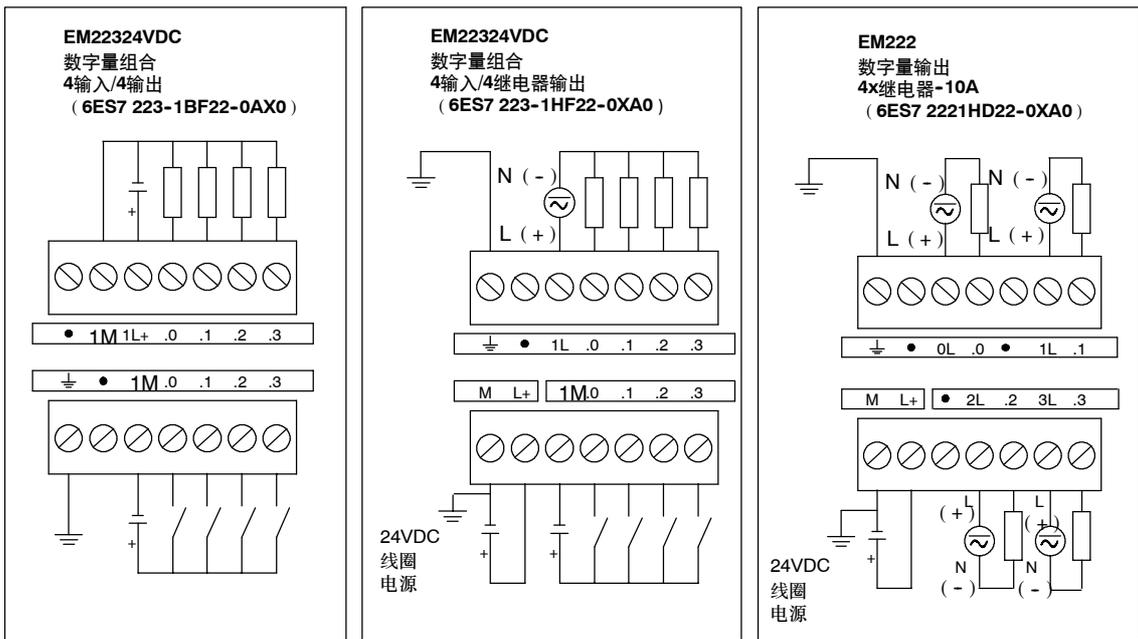
常规	24VDC输出		继电器输出		120/230VAC输出
	0.75A	5 μ A	2A	10A	
类型	固态-MOSFET ¹ (信号源)		干触点		直通
额定电压	24VDC		24VDC或250VAC		120/230VAC
电压范围	20.4-28.8VDC		5至30VDC或 5至250VAC	12至30VDC或 12至250VAC	40至264VAC (47至63Hz)
24VDC线圈电源电压范围	-		20.4-28.8VDC		-
浪涌电流 (最大)	8A, 100ms	30A	5A,4s@10%占空比	15A,4s@10%占空比	5Arms, 2AC周期
逻辑1 (最小)	20VDC		-		L1 (-0.9Vrms)
逻辑0 (最大)	0.1VDC, 10K Ω 负载	0.2VDC, 5K Ω 负载	-		-
每点额定电流 (最大)	0.75A	5 μ A	2.00A	10A阻性; 2ADC感性; 3AAC感性	0.5AAC
每个公共端的额定电流 (最大)	6A	5 μ A	8A	10A	0.5AAC
漏电流 (最大)	10 μ A	30 μ A	-		132VAC是1.1mArms, 264VAC是1.8mArms
灯负载 (最大)	5W	50W	30WDC/ 200WAC ^{6, 7}	100WDC/ 1000WAC	60W
感性嵌位电压	L+减48V	L+减47V	-		-
接通状态电阻 (触点)	0.3 Ω 典型 (0.6 Ω 最大)	0.05 Ω 最小	0.2 Ω 最小, 新的时候	0.1 Ω 最小, 新的时候	410 Ω 最大, 当负载电流小于 0.05A时
隔离 光电隔离 (现场到逻辑) 线圈到逻辑 线圈到触点 电阻 (线圈到触点) 隔离组	500VAC, 1分钟 - - - 见接线图		- 无 1500VAC, 1分钟 100M Ω 最小, 新的时候 见接线图		1500VAC, 1分钟 - - - 1点
延时断开到接通/接通到断开 切换 (最大)	50 μ s/200 μ s -	500 μ s -	- 10ms	- 15ms	0.2ms + 1/2AC周期 -
切换频率 (最大)	-		1Hz		10Hz
机械寿命周期	-		10,000,000 (无负载)	30,000,000 (无负载)	-
触点寿命	-		100,000 (额定负载)	30,000 (额定负载)	-
同时接通的输出	所有水平安装时低于55°C, 垂直安装时低于45°C		所有的都在55°C (水平), 最大 20A模块电流 所有的都在45°C (水平), 最大 20A模块电流 ⁵ 所有的都在40°C (垂直), 每个点 10A		所有水平安装时低于 55°C, 垂直安装时低于 45°C
两个输出并联	是的, 只有输出在同一个组内		否		否
电缆长度 (最大) 屏蔽 非屏蔽	500m 150m		500m 150m		500m 150m

- 1 当一个机械触点接通S7-200 CPU或任意扩展模块的供电时, 它发送一个大约50毫秒的“信号到数字输出, 您需要考虑这一点, 尤其是您使用触响响应短脉冲的设备时。
- 2 当一个机械触点接通AC扩展模块的输出电源时, 它向AC输出发出一个宽度为大约1/2AC周期的“1”信号。您必须考虑这一点。
- 3 由于是直通电路, 负载电流必须是完整的AC波形而非半波。最小负载电流是0.05AAC。当负载电流在5mA和50mAAC之间时, 该电流是可控的, 但是, 由于410 Ω 串行电阻的存在会有额外的压降。
- 4 如果因为过多的感性开关或不正常的条件而引起输出过热, 输出点可能关断或被损坏。如果输出在关断一个感性负载时遭受大于0.7J的能量, 那么输出将可能过热或被损坏。为了消除这个限制, 可以将第3章中描述的抑制电路和负载并联在一起。对于给定的应用, 这些部件的尺寸要合适。
- 5 EM222DO4x继电器的FM额定值和其它S7-200不同。此模块具有符合FM Class I, Division Groups A, B, C&D Hazardous Locations的T4额定值, 而不是的T4A。
- 6 继电器使用寿命将降低75%, 除非采取措施将接通浪涌降低到输出的浪涌电流额定值以下。
- 7 灯负载的瓦特额定值是用于额定电压的。依据正被切换的电压, 按比例降低瓦特额定值 (例如120VAC - 100 W)。

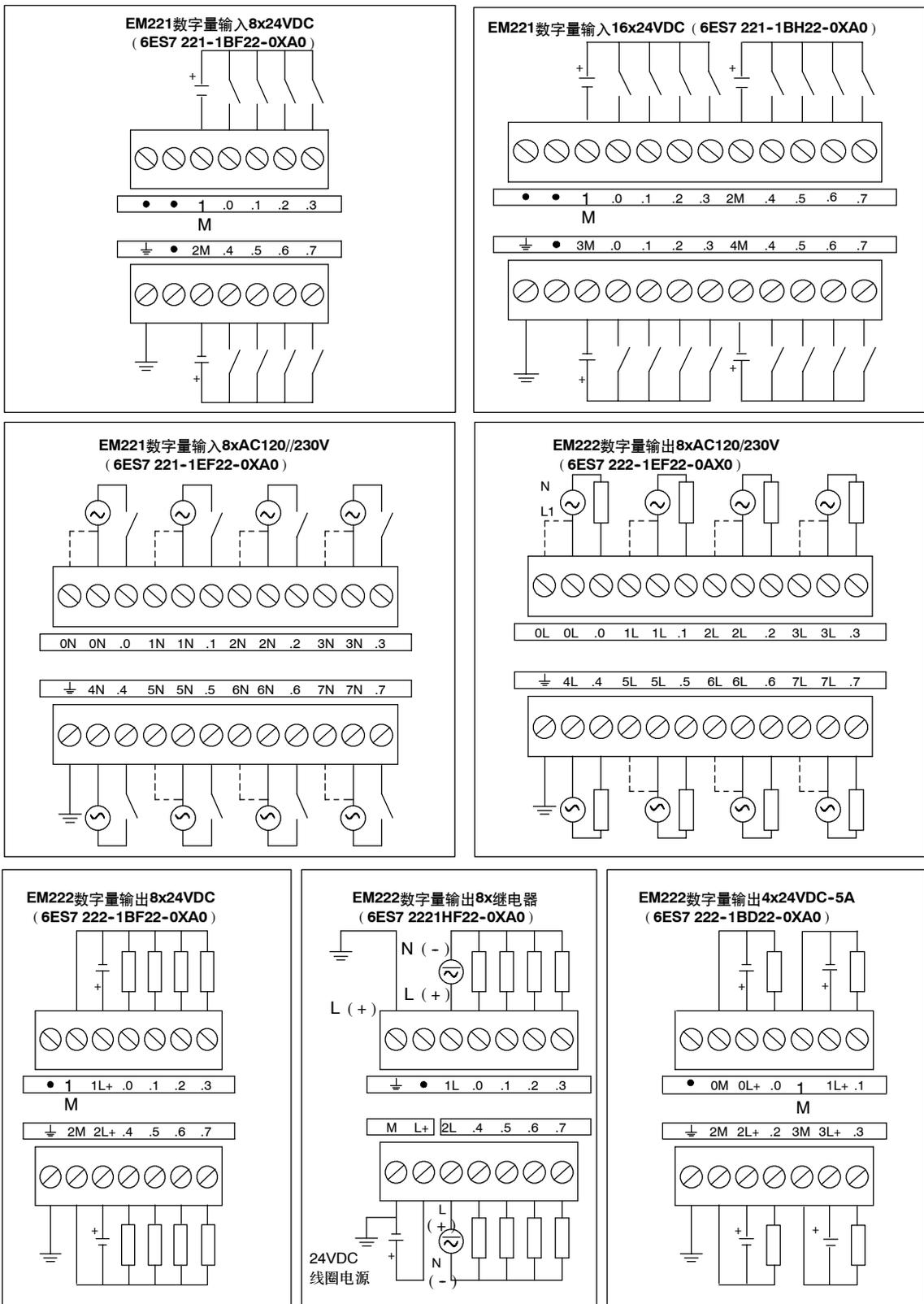


图A-8 S7-200数字量扩展模块输出

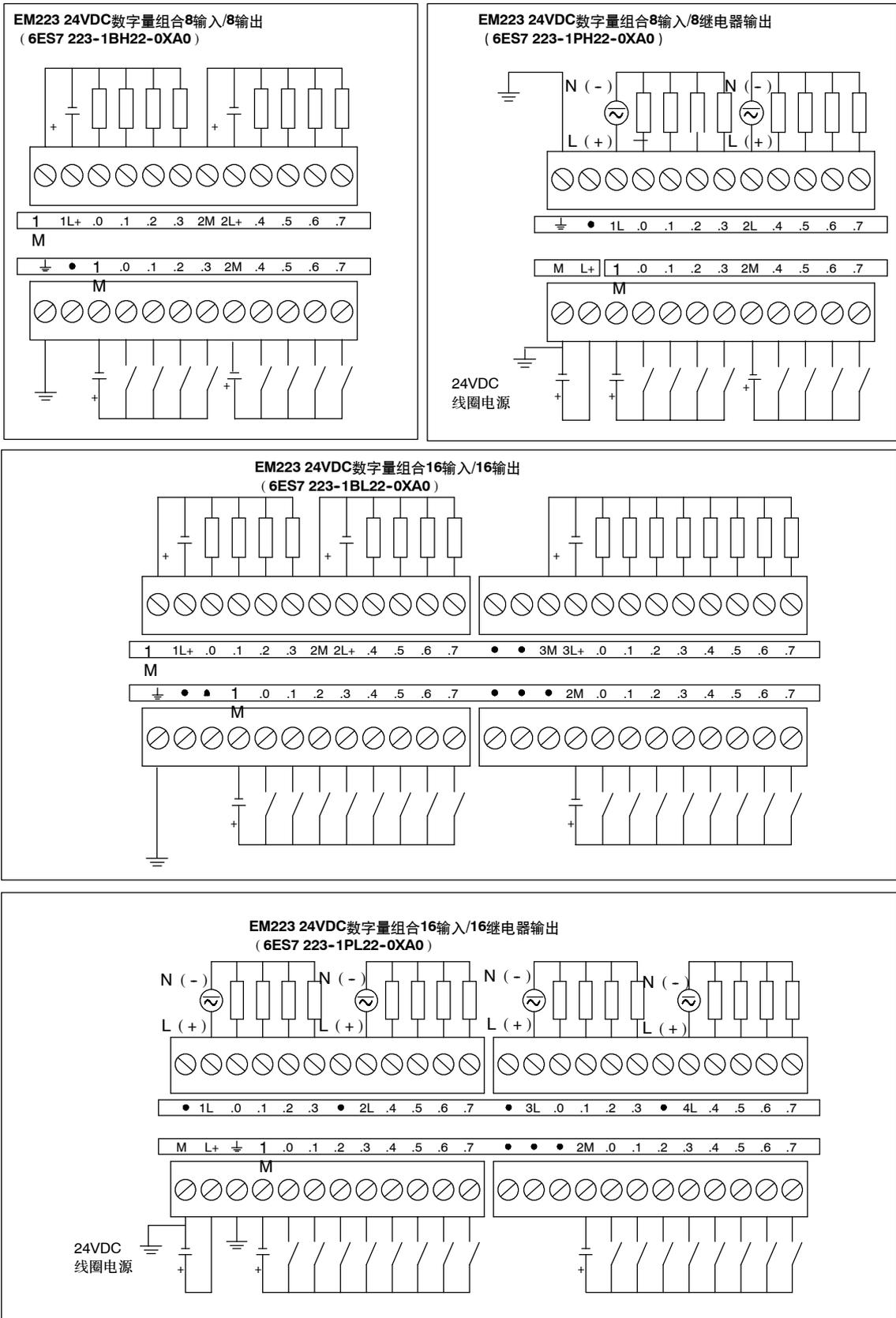
接线图



图A-9 EM222和EM223扩展模块的接线图



图A-10 EM221和EM222扩展模块的接线图



图A-11 EM 223扩展模块的接线图

模拟量扩展模块规范

表A-15 模拟量扩展模块订货号

订货号	扩展模块	输入	输出	可拆卸连接
6ES7 231-0HC22-0XA0	EM231模拟量输入, 4输入	4	-	否
6ES7 232-0HB22-0XA0	EM232模拟量输出, 2输出	-	2	否
6ES7 235-0KD22-0XA0	EM235模拟量混合模块4输入/1输出	4	1 ¹	否

1 CPU为该模块保留2个模拟量输出通道

表A-16 模拟量扩展模块常规规范

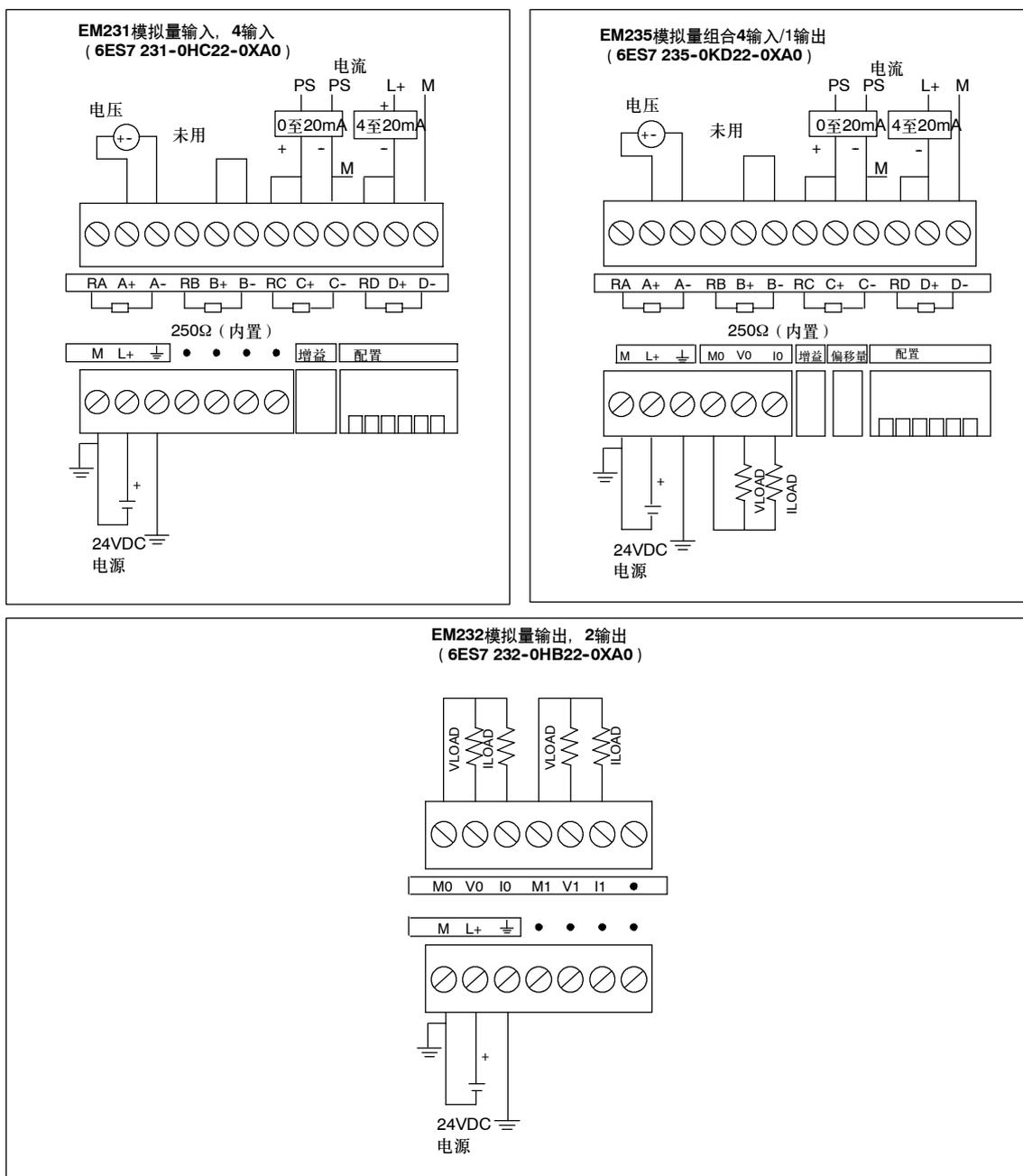
订货号	模块名称及描述	尺寸 (mm) (W x H x D)	重量	功耗	+5VDC	VDC要求 +24VDC
6ES7 231-0HC22-0XA0	EM231模拟量输入, 4输入	71.2x80x62	183g	2W	20mA	60mA
6ES7 232-0HB22-0XA0	EM232模拟量输出, 2输出	46x80x62	148g	2W	20mA	70mA (两个输出都是20mA)
6ES7 235-0KD22-0XA0	EM235模拟量混合模块4输入/1输出	71.2x80x62	186g	2W	30mA	60mA (输出为20mA)

表A-17 模拟量扩展模块输入规范

常规	6ES7 231-0HC22-0XA0	6ES7 235-0KD22-0XA0
数据字格式 双极性, 满量程 单极性, 满量程	(见图A-14) -32000至+32000 0至32000	(见图A-14) -32000至+32000 0至32000
DC输入阻抗	≥10MΩ电压输入 250Ω电流输入	≥10MΩ电压输入 250Ω电流输入
输入滤波衰减	-3db, 3.1KHz	-3db, 3.1KHz
最大输入电压	30VDC	30VDC
最大输入电流	32mA	32mA
精度 双极性 单极性	11位, 加1符号位 12位	
隔离 (现场与逻辑)	无	无
输入类型	差分	差分
输入范围 电压 电流	可选择的, 对于可用的范围, 见表A-20 0至20mA	可选择的, 对于可用的范围, 见表A-21 0至20mA
输入分辨率	参见表A-20	参见表A-21
模拟到数字转换时间	<250μs	<250μs
模拟输入阶跃响应	1.5ms到95%	1.5ms到95%
共模抑制	40dB, DC到60Hz	40dB, DC到60Hz
共模电压	信号电压加共模电压必须为≤±12V	信号电压加共模电压必须为≤±12V
24VDC电压范围	20.4 - 28.8VDC (等级2, 有限电源, 或来自PLC的传感器电源)	

表A-18 模拟量扩展模块输出规范

常规	6ES7 232-0HB22-0XA0	6ES7 235-0KD22-0XA0
隔离（现场与逻辑）	无	无
信号范围		
电压输出	±10V	±10V
电流输出	0至20mA	0至20mA
分辨率, 满量程		
电压	12位, 加符号位	11位, 加符号位
电流	11位	11位
数据字格式		
电压	-32000至+32000	-32000至+32000
电流	0至+32000	0至+32000
精度		
最差情况, 0°至55°C		
电压输出	±2%满量程	±2%满量程
电流输出	±2%满量程	±2%满量程
典型, 25°C		
电压输出	±0.5%满量程	±0.5%满量程
电流输出	±0.5%满量程	±0.5%满量程
建立时间		
电压输出	100μs	100μs
电流输出	2mS	2mS
最大驱动		
电压输出	5000Ω最小	5000Ω最小
电流输出	500Ω最大	500Ω最大
24VDC电压范围	20.4 - 28.8VDC (等级2, 有限电源, 或来自PLC的传感器电源)	



图A-12 模拟量扩展模块的接线图

模拟量模块的LED指示灯，如表A-17所示。

模拟量模块的LED指示灯，如表A-19所示。

表A-19 模拟量模块的LED指示灯，如表A-17所示。

LED指示灯	ON	OFF
24VDC电源良好	无故障	没有24VDC电源



提示

用户电源的状态也在特殊存储器（SM）位中报告。对于更多信息，参见附录D，SMB8到SMB21/O模块ID和错误寄存器。

输入校准

校准调节影响模拟量多路转换器运算的放大器（见EM231输入方框图图A-15和EM235的输入方框图图A-16）。因此，校准影响到所有的用户输入通道。即使在校准以后，如果模拟量多路转换器之前的输入电路的部件值发生变化，那么，从不同通道读入同一个输入信号，其信号值也会有微小的不同。

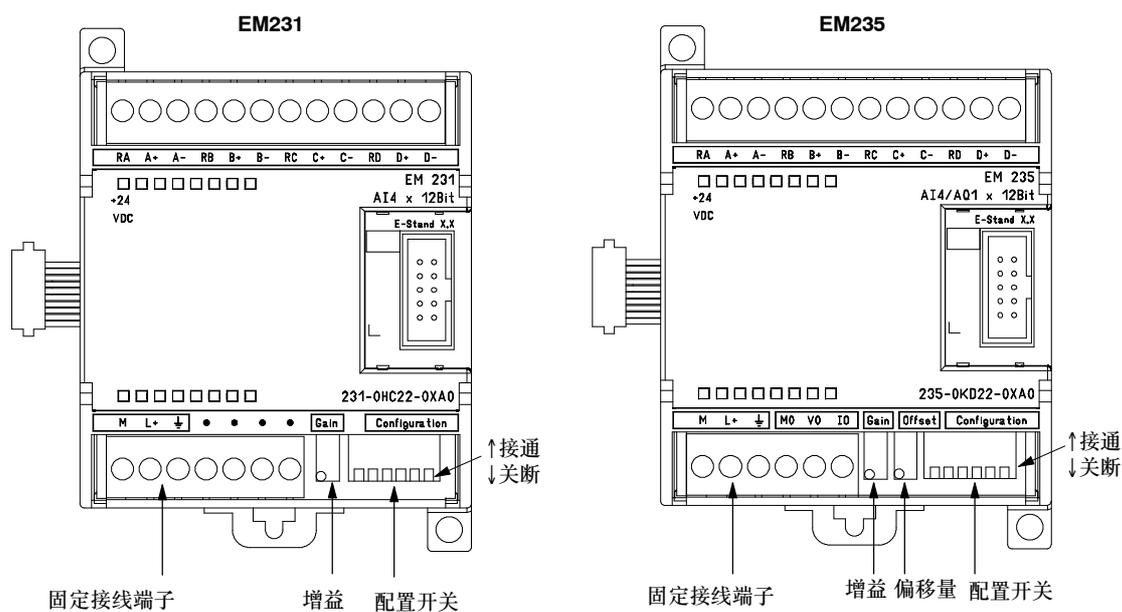
为了达到表中所列的技术参数，应启动用于模块所有输入的模拟输入滤波器。计算平均值时，选择64次或更多的采样次数。

校准输入时，其步骤如下：

1. 切断模块电源。选择需要的输入范围。
2. 接通CPU和模块电源。使模块稳定15分钟。
3. 用一个变送器、一个电压源或一个电流源，将零值信号加到一个输入端。
4. 读取适当的输入通道在CPU中的测量值。
5. 调节OFFSET（偏置）电位计，直到读数为零，或所需要的数字数据值。
6. 将一个满刻度值信号接到输入端子中的一个。读出送到CPU的值。
7. 调节GAIN（增益）电位计，直到读数为32000，或所需要的数字数据值。
8. 必要时，重复偏置和增益校准过程。

EM231和EM235的校准和配置位置

图A-13给出了在模块底部的接线端子上的校准电位计和设置配置DIP开关。



图A-13 EM231和EM235校准电位计和配置DIP开关位置

配置EM231

表A-20所示为如何使用配置DIP开关配置EM231模块。开关1、2和3选择模拟量输出范围。所有输入设置为相同的模拟量输入量程。该表中，ON是闭合，OFF是断开。只在电源接通时读取开关设置。

表A-20 用于选择模拟量输入量程的EM231配置开关表

单极性			满量程输入	分辨率
SW1	SW2	SW3		
ON	OFF	ON	0至10V	2.5mV
	ON	OFF	0至5V	1.25mV
			0至20mA	5μA
双极性			满量程输入	分辨率
SW1	SW2	SW3		
OFF	OFF	ON	±5V	2.5mV
	ON	OFF	±2.5V	1.25mV

配置EM235

表A-21所示为如何使用配置DIP开关配置EM235模块。开关1至6可选择输入量程和分辨率。所有的输入都设置为相同的模拟量输入量程和格式。表A-21所示为如何选择单极性/双极性（开关6）、增益（开关4和5）以及衰减（开关1，2和3），在该表中，ON是闭合，OFF是断开。只在电源接通时读取开关设置。

表A-21 用于选择模拟量量程和精度的EM235配置开关表

单极性						满量程输入	分辨率
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	ON	0至50mV	12.5 μ V
OFF	ON	OFF	ON	OFF	ON	0至100mV	25 μ V
ON	OFF	OFF	OFF	ON	ON	0至500mV	125 μ V
OFF	ON	OFF	OFF	ON	ON	0至1V	250 μ V
ON	OFF	OFF	OFF	OFF	ON	0至5V	1.25mV
ON	OFF	OFF	OFF	OFF	ON	0至20mA	5 μ A
OFF	ON	OFF	OFF	OFF	ON	0至10V	2.5mV
双极性						满量程输入	分辨率
SW1	SW2	SW3	SW4	SW5	SW6		
ON	OFF	OFF	ON	OFF	OFF	\pm 25mV	12.5 μ V
OFF	ON	OFF	ON	OFF	OFF	\pm 50mV	25 μ V
OFF	OFF	ON	ON	OFF	OFF	\pm 100mV	50 μ V
ON	OFF	OFF	OFF	ON	OFF	\pm 250mV	125 μ V
OFF	ON	OFF	OFF	ON	OFF	\pm 500mV	250 μ V
OFF	OFF	ON	OFF	ON	OFF	\pm 1V	500 μ V
ON	OFF	OFF	OFF	OFF	OFF	\pm 2.5V	1.25mV
OFF	ON	OFF	OFF	OFF	OFF	\pm 5V	2.5mV
OFF	OFF	ON	OFF	OFF	OFF	\pm 10V	5mV

EM231和EM235输入数据字格式

图A-14给出了12位数据值在CPU的模拟量输入字中的位置。



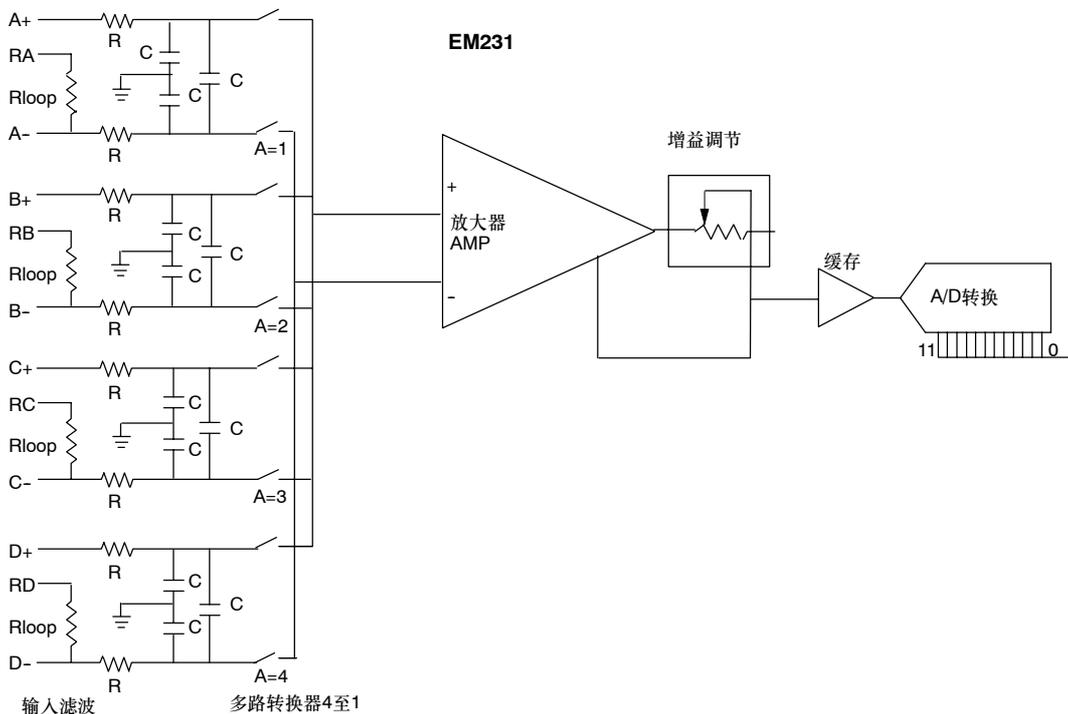
图A-14 EM231和EM235输入数据字格式



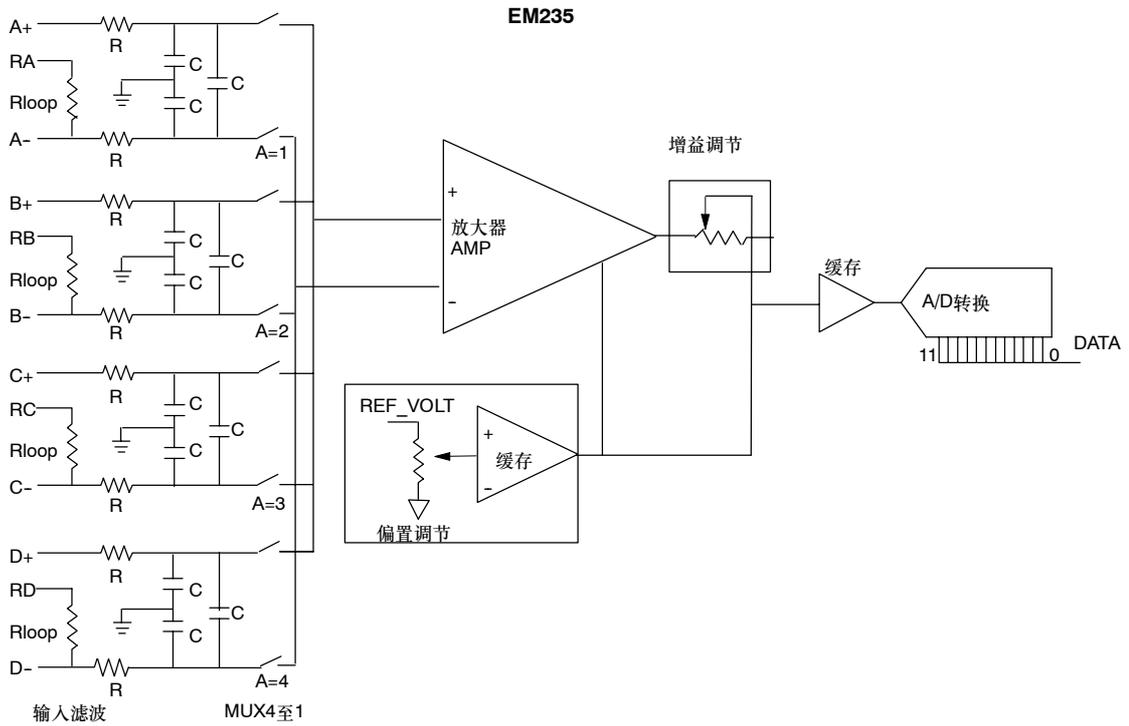
提示

模拟量到数字量转换器的12位读数是左对齐的。最高有效位是符号位：0表示是正值。
 在单极性格式中，3个连续的0使得ADC计数值每变化1个单位，数据字中则以8为单位变化。
 在双极性格式中，4个连续的0使得ADC计数值每变化1个单位，数据字中则以16为单位变化。

EM231和EM235输入方框图



图A-15 EM231的输入方框图



图A-16 EM235的输入方框图

EM232和EM235输出数据字格式

图A-17给出了12位数据在CPU的模拟量输出字中的位置。



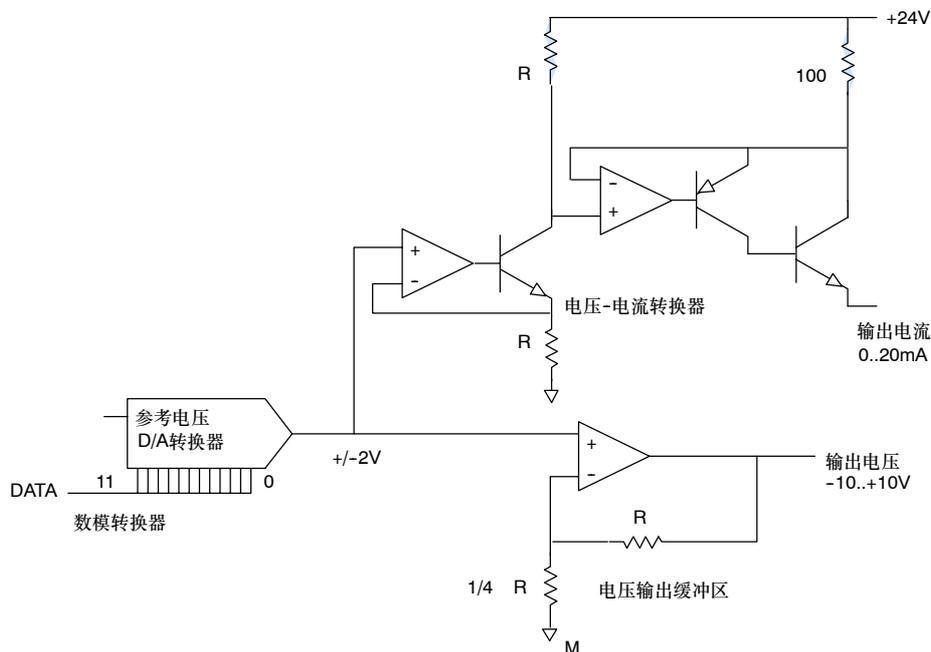
图A-17 EM232和EM235输出数据字格式



提示

数字量到模拟量转换器 (DAC) 的12位读数在其输出数据格式中是左端对齐的。最高有效位是符号位: 0表示是正值。数据在装载到DAC寄存器之前, 4个连续的0是被截断的, 这些位不影响输出信号值。

EM232和EM235输出方框图



图A-18 EM232和EM235的输出方框图

安装指南

使用下面的指南以确保安装正确、可靠：

- 确保24VDC传感器电源无噪声、稳定。
- 传感器线尽可能短。
- 传感器线使用屏蔽的双绞线。
- 仅在传感器侧将屏蔽接终端。
- 未用通道应短接，见图A-18。
- 避免将导线弯成锐角。
- 使用电缆槽进行敷线。
- 避免将信号线与高能量线平行布置。若两条线必须交叉，应以直角度相交。
- 通过隔离输入信号或输入信号参考于模拟量模块外部24V电源的公共端，从而确保输入信号范围在技术规范所规定的共模电压之内。



提示

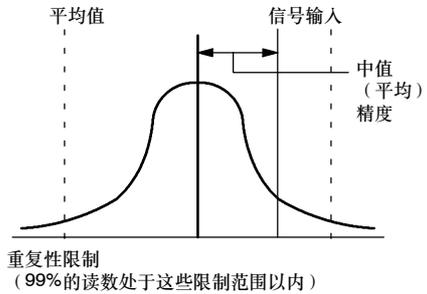
建议EM231和EM235扩展模块不用于热电耦。

理解模拟量输入模块：精度和重复性

EM231和EM235模拟量输入模块是价格适中、高速12位模拟量输入模块。此模块可以在149 μs内将模拟信号输入转换成其对应的数字量值。每当程序访问模拟量点时，模拟信号输入就会被转换。这些转换时间必须加到用于访问模拟量输入的指令的基本执行时间上。

EM231和EM235提供一个未经处理的数字值（未经线性化或滤波），它对应于模拟量输入端处出现的模拟量电压或电流。由于这种模块是高速模块，它们可以跟踪模拟量信号中的快速变化（包括内部和外部噪声）。

对一个恒定或缓慢变化的模拟量输入，由噪声引起的信号读数之间的差异，可通过对读数取平均值的方法使其影响为最小。但由于计算平均值而增加读取信号的次数（即采样次数），会相应地降低对外部输入信号的响应速度。



图A-19 精度定义

图A-19为99%的重复性限定，各个读入值的平均值，以及平均精度的图形示意。

重复性的技术规范描述不改变输入信号时，模块每次读数之间的差异。重复性技术规范规定限制范围，要求99%的读数处于这限制范围以内。重复性在图A-16中用钟形曲线描述。

平均精度的技术规范描述误差的平均值（各个读数的平均值和实际模拟量输入信号精确值之间的差异）。

表A-22给出重复性技术规范和与每个可配置范围有关的平均精度。

模拟量规范的定义

- 精度: 在某一给定点上与期望值的偏差。
- 分辨率: 输出中所反映的最低位 (LSB) 的变化。

表A-22 EM231和EM235规范

满量程输入	重复性 ¹		平均精度 ^{1,2,3,4}	
	%满量程	数值	%满量程	数值
EM231规范				
0至5V	±0.075%	±24	±0.1%	±32
0 ~ 20mA				
0 ~ 10V		±48	±0.05%	
±2.5V				
±5V				
EM235规范				
0 ~ 50mV	±0.075%	±24	±0.25%	±80
0 ~ 100mV			±0.2%	±64
0 ~ 500mV			±0.05%	±16
0 ~ 1V				
0 ~ 5V				
0 ~ 20mA				
0 ~ 10V			±0.075%	±48
±25mV	±0.2%	±128		
±50mV	±0.1%	±64		
±100mV	±0.05%	±32		
±250mV				
±500mV				
±1V				
±2.5V				
±5V				
±10V				

¹ 在所选择的输入量程标定后测量

² 接近零模拟量输入信号的偏置误差不予纠正, 也不包括在精度规范内。

³ 由于模拟量多路转换器设定时间的限制, 因此会有通道到通道的传递转换误差。最大的传递误差是通道间差分值的0.1%。

⁴ 平均精度包括非线性和0到55°C 漂移的影响。

热电偶和RTD（热电阻）扩展模块规范

表A-23 热电偶和RTD模块订货号

订货号	扩展模块	输入	输出	可拆卸连接
6ES7 231-7PD22-0XA0	EM231模拟输入热电偶, 4输入	4热电偶	-	否
6ES7 231-7PB22-0XA0	EM231模拟输入RTD, 2输入	2RTD	-	否

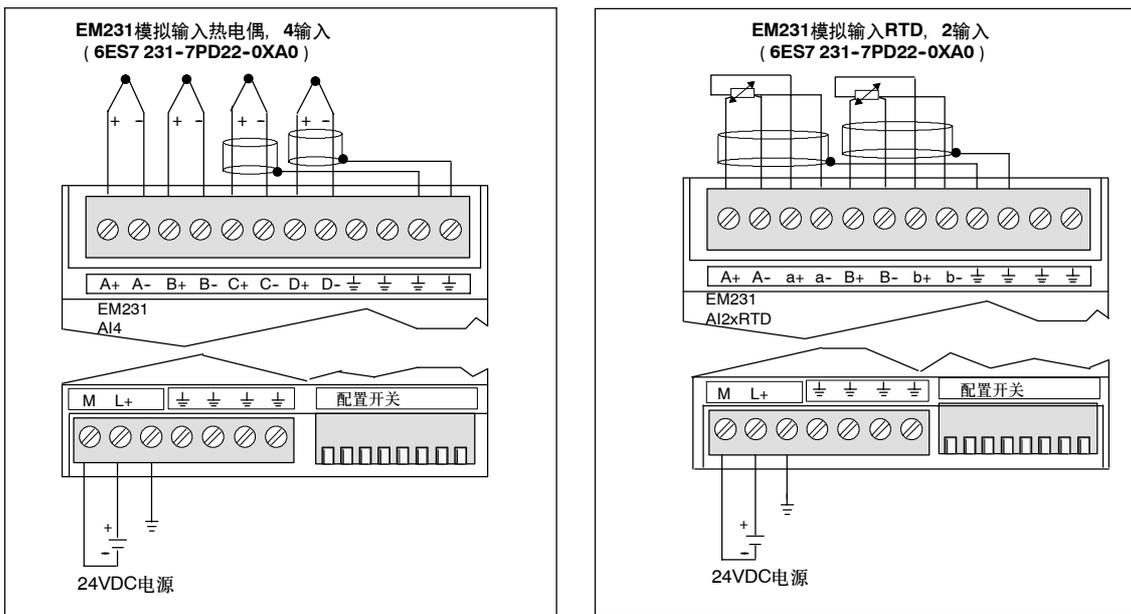
表A-24 热电偶和RTD模块常规规范

订货号	模块名称及描述	尺寸 (mm) (W x H x D)	重量	功耗	VDC要求	
					+5VDC	+24VDC
6ES7 231-7PD22-0XA0	EM231模拟输入热电偶, 4输入	71.2x80x62	210g	1.8W	87mA	60mA
6ES7 231-7PB22-0XA0	EM231模拟输入RTD, 2输入	71.2x80x62	210g	1.8W	87mA	60mA

表A-25 热电偶和RTD模块规范

常规	6ES7 231-7PD22-0XA0 热电偶	6ES7 231-7PB22-0XA0 RTD
隔离 现场侧到逻辑地 现场侧到24VDC 24VDC到逻辑地	500VAC 500VAC 500VAC	500VAC 500VAC 500VAC
共模输入范围 (输入通道到输入通道)	120VAC	0
共模抑制	>120dB@120VAC	>120dB@120VAC
输入类型	悬浮型热电偶	模块参考接地的热电阻
输入范围 ¹	TC类型 (选择一种) S, T, R, E, N, K, J 电压范围: +/-80mV	热电阻类型 (选择一种) 铂 (Pt)、铜 (Cu)、镍 (Ni) 或电阻 对于可用的RTD类型, 见表A-30。
输入分辨率 温度 电压 电阻	0.1°C/0.1°F 15位加符号位 -	0.1°C/0.1°F - 15位加符号位
测量原理	Sigma-delta	Sigma-delta
模块更新时间: 所有通道	405mS	405mS (Pt10000为700ms)
导线长度	到传感器最长为100m	到传感器最长为100m
导线回路电阻	100Ω最小	20Ω, 2.7Ω, (Cumax)
干扰抑制	85dB在50Hz/60Hz/400Hz时	85dB在50Hz/60Hz/400Hz时
数据字格式	电压: -27648到+27648	电阻: 0至+27648
传感器最大散热	-	1mW
输入阻抗	≥1MΩ	≥10MΩ
最大输入电压	30VDC	30VDC (检测), 5VDC (源)
输入滤波衰减	-3dbat21kHz	-3dbat3.6kHz
基本误差	0.1%FS (电压)	0.1%FS (电阻)
重复性	0.05%FS	0.05%FS
冷端误差	±1.5°C	-
24VDC电压范围	20.4-28.8VDC	20.4-28.8VDC

¹ 选择的输入范围 (温度、基于阻抗的电压) 将作用于模块的所有通道。



图A-20 EM 231热电偶和EM 231RTD模块的接线端子标识

兼容性

RTD和热电偶模块设计用于CPU222、CPU224、CPU224XP和CPU226。



提示

RTD和热电偶模块安装在一个稳定的温度环境内时，具有最佳的性能。

例如，EM231热电偶模块有专门的冷端补偿电路。该电路在模块连接器处测量温度，并对测量值作出必要的修正，以补偿基准温度和模块处温度之间的温度差。如果EM231热电偶模块安装环境的温度变化很剧烈，则会引起附加的误差。

为了达到最大的精度和重复性，西门子公司建议，S7-200RTD和热电偶模块要安装在环境温度稳定的地方。

噪声抑制

使用屏蔽线是最好的噪声抑制方法。如果热电偶的输入未使用，短接未使用的通道，或将它们并行连接到其它通道上。

EM231热电偶模块

EM231热电偶模块为S7-200系列产品提供了连接7种类型热电偶的使用方便、带隔离的接口：J、K、E、N、S、T和R。它可以使S7-200能连接低电平模拟信号，测量范围为 $\pm 80\text{mV}$ 。所有连接到该模块的热电偶都必须是同一类型的。

热电偶的基本知识

任何两种金属，其连接处都会形成热电偶。热电偶产生的电压与连接点温度成正比。这个电压很低，1微伏可能代表若干度。测量来自热电偶的电压，进行冷端补偿，然后线性化结果，这是使用热电偶进行温度测量的基本步骤。

当您将一个热电偶连接到EM231热电偶模块时，两根不同的金属导线连接到模块的输入信号接线端子。两根不同金属导线彼此连接处形成传感器的热电偶。

在两根不同金属导线连接到输入信号接线端子的地方形成其它两个热电偶。接线端子处的温度产生一个电压，加到从传感器热电偶来的电压上。如果这个电压不校正，那末，所测量的温度会偏离传感器的温度。

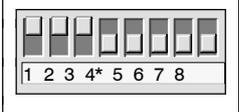
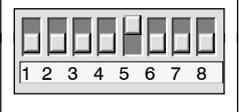
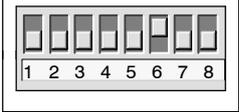
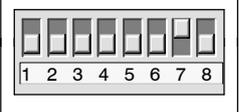
冷端点补偿用来补偿接线端子处的热电偶。热电偶表基于基准连接点温度，通常是摄氏0度。模块冷端补偿将接线端子处的温度补偿到摄氏0度。冷端补偿补偿了由于接线端子热电偶电压所引起的电压增加。模块温度是在内部测量的。这个温度转换成一个值，它加到传感器的转换值上。然后，用电偶表线性化被修正后的传感器转换值。

组态EM231热电偶模块

配置DIP开关位于模块的底部，可以选择热电偶模块的类型、断线检测、温度范围和冷端补偿。要使DIP开关设置起作用，需要循环地给PLC和/或用户的24V电源上电。

DIP开关4为以后的应用保留，将DIP开关4设定为0位置（向下），其他DIP开关的设定请参阅表A-26。

表A-26 组态热电偶模块DIP开关

开关1, 2, 3	热电偶类型	设置	描述
<p>SW1,2,3</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p> <p>*将DIP开关4 设定为0 (向下) 位置。</p>	J (缺省)	000	开关1至3为模块上的所有通道选择热电偶类型 (或mV操作)。例如, 选E类型, 热电偶开关SW1=0, SW2=1, SW3=1
	K	001	
	T	010	
	E	011	
	R	100	
	S	101	
	N	110	
	+/-80mV	111	
开关5	断线检测方向	设置	描述
<p>SW5</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p>	正向标定 (+3276.7度)	0	0指示正向断线检测 1指示负向断线检测
	负向标定 (-3276.8度)	1	
开关6	断线检测使能	设置	描述
<p>SW6</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p>	使能	0	通过加上25μA电流到输入端进行断线检测。断线检测使能开关可以使能或禁止检测电流。断线检测始终在进行, 即使关闭了检测电流。如果输入信号超过 ±200mV, 则EM231热电偶模块开始断线检测。如检测到断线, 测量读数被设定成由断线检测所选定的值。
	禁止	1	
开关7	温度范围	设置	描述
<p>SW7</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p>	摄氏度 (°C)	0	EM231热电偶模块能够报告摄氏温度和华氏温度。摄氏温度与华氏温度的转换在内部进行。
	华氏度 (°F)	1	
开关8	冷端补偿	设置	描述
<p>SW8</p>  <p>配置 ↑1 - 接通 ↓0 - 断开</p>	冷端补偿使能	0	使用热电偶必须进行冷端补偿, 如果没有使能冷端补偿, 模块的转换则会出现错误。因为热电偶导线连接到模块连接器时会产生电压, 当选择±80mV的范围时, 冷端补偿会自动禁止。
	冷端补偿禁止	1	

**提示**

- 断线检测电流源可能干扰某些低电平信号，例如热电偶模拟器。
- 输入电压超过±200mV将触发断线检测，即使断线检测电流被禁用。

**提示**

- 当环境温度变化时，模块误差有可能超过技术规范中的数据。
- 超过模块温度范围的规范时，有可能导致模块冷端补偿出错。

使用热电偶：状态指示器

热电偶模块提供PLC测量温度或出错类型的数据字。状态位指示输入范围错误和用户电源/模块故障。LED指示模块状态，用户程序也可检测出相应错误状态并采取相应的措施。表A-27给出了EM231热电偶状态指示器。

表A-27 EM231热电偶状态指示器

出错类型	通道数据	SF 指示灯 红色	24V指示灯 绿色	范围状态位1	24VDC用户 电源故障 ²
没有出错	转换数据	OFF	ON	0	0
24V丢失	32766	OFF	OFF	0	1
使能断线检测和检测电流源	-32768/32767	闪烁	ON	1	0
超出输入范围	-32768/32767	闪烁	ON	1	0
诊断出错	0000	ON	OFF	0	注释3

¹ 范围状态位是模块出错寄存器字节中的位3（SMB9用于模块1，SMB11用于模块2等等）

² 不良状态位是模块出错寄存器字节中的位2（SMB9，SMB11等，请参阅附录D）

³ 诊断出错引起模块组态错误。在模块组态错误之前，用户电源故障状态位可能设置或没有设置。

**提示**

通道数据格式是2的补码，16位字，表示温度的单位为0.1度例如测量的温度为100.2度，则报告数据是1002，电压数据标定到27648。例如，-60.0mV则报告为20736（=-60mV/80mV*27648）

如PLC已读取到数据，则每405ms更新所有4个通道的数据，如在一个更新时间内，PLC没有读数据，则模块报告原有的数据一直到PLC读数据后的下一次模块更新，为了保持通道数据总是为当前值，建议PLC读数据的频度至少和模块更新频率相同。

**提示**

如正在使用热电偶模块，应该禁止在PLC中使用模拟量滤波。模拟量滤波会妨碍出错条件的检测。

表A-28 温度范围 (°C) 和各类热电偶的精度

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	16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
:	:								
27649	6C01							80.0029mV	
27648	6C00							80mV	
:	:								
17680	4510		↑			↑			
:	:					1768.0°C			
13720	3598		↑						NR
:	:		超出范围						
13000	32C8	↑	1300.0°C				↑	1300.0°C	
:	:								
12000	2EE0	1200.0°C							
:	:								
10000	2710			↑					
:	:			400.0°C					
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							
:	:								
-2400	F6A0								
:	:		低于范围						
-2550	F60A								
:	:								
-2700	F574	↓	-270.0°C	-270.0°C	-270.0°C				NR
:	:								
-27648	9400		↓	↓	↓			-80mV	
-27649	93FF							-80.0029mV	
:	:								
-32512	8100							-94.071mV	UR
↓	↓							↓	↓
-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%	±0.3%	±0.6%	±0.4%	±0.1%	
精度 (无冷端的正常范围)		±1.5°C	±1.7°C	±1.4°C	±1.3°C	±3.7°C	±1.6°C	±0.10%	
冷端误差		±1.5°C	±1.5°C	±1.5°C	±1.5°C	±1.5°C	±1.5°C	N/A	
*OF=下溢OR=超出范围NR=正常范围; UR=低于范围UF=下溢									
↑表示所有大于该值但小于断线阈值的模拟量都报告为上溢出值, 32767 (0x7FFF)。 ↓表示所有小于该值但大于断线阈值的模拟量都报告为下溢出值, -32768 (0x8000)。									

表A-29 温度范围(°)用于热电偶类型。

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	16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					2764.8°F		80mV	NR
27648	6C00		↑						
:	:								
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	FDBC					-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						
-4000	F060				-400.0°F				
:	:				低于范围				
-4270	EF52			-427.0°F					
:	:			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	
↓	↓							↓	
-3268	8000	<-346.0°F	<-454.0°F	<-454.0°F	<-454.0°F	<-58.0°F	<-454.0°F	<-94.07mV	UF

*OF=下溢OR=超出范围NR=正常范围; UR=低于范围UF=下溢
 ↑表示所有大于该值但小于断线阈值的模拟量都报告为上溢出值, 32767 (0x7FFF)。
 ↓表示所有小于该值但大于断线阈值的模拟量都报告为下溢出值, -32768 (0x8000)。

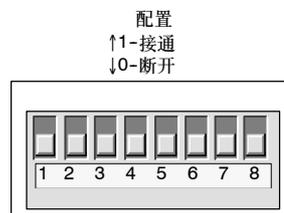
EM231热电阻模块

EM231热电阻模块为S7-200连接各种型号的热电阻提供了方便的接口。它也允许S7-200测量三个不同的电阻范围。连接到模块的热电阻必须是相同的类型。

组态EM231RTD（热电阻）模块

使用DIP开关可以选择热电阻的类型，接线方式，温度测量单位和传感器熔断方向。DIP配置开关位于模块的底部，如图A-21所示，要使DIP开关设置起作用，需要循环地给PLC和/或用户的24V电源上电。

可以通过设定DIP开关1、2、3、4和5来选择RTD的类型，如表A-30所示。对于DIP开关的设置，参见表A-31。



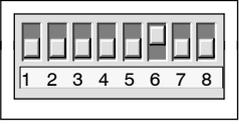
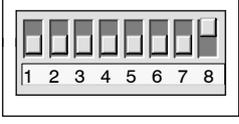
图A-21 EM231RTD模块的DIP开关

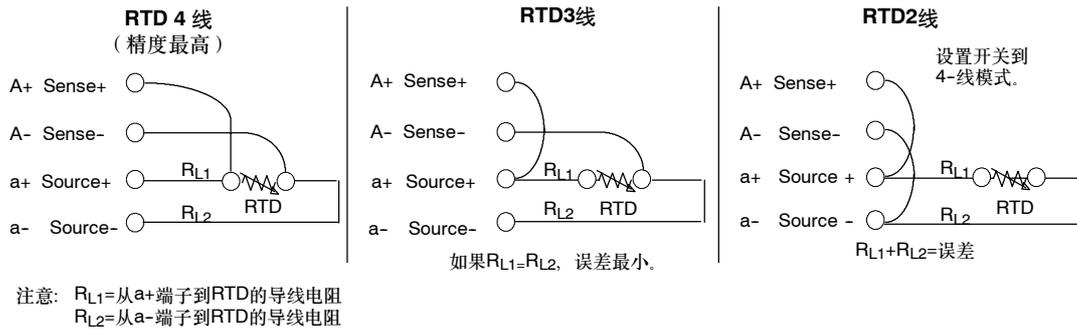
表A-30 选择RTD类型：DIP开关1-5。

RTD类型 ¹	SW1	SW2	SW3	SW4	SW5	RTD类型 ¹	SW1	SW2	SW3	SW4	SW5
100ΩPt0.003850 (缺省)	0	0	0	0	0	100ΩPt0.003902	1	0	0	0	0
200ΩPt0.003850	0	0	0	0	1	200ΩPt0.003902	1	0	0	0	1
500ΩPt0.003850	0	0	0	1	0	500ΩPt0.003902	1	0	0	1	0
1000ΩPt0.003850	0	0	0	1	1	1000ΩPt0.003902	1	0	0	1	1
100ΩPt0.003920	0	0	1	0	0	SPARE	1	0	1	0	0
200ΩPt0.003920	0	0	1	0	1	100ΩNi0.00672	1	0	1	0	1
500ΩPt0.003920	0	0	1	1	0	120ΩNi0.00672	1	0	1	1	0
1000ΩPt0.003920	0	0	1	1	1	1000ΩNi0.00672	1	0	1	1	1
100ΩPt0.0038505	0	1	0	0	0	100ΩNi0.006178	1	1	0	0	0
200ΩPt0.0038505	0	1	0	0	1	120ΩNi0.006178	1	1	0	0	1
500ΩPt0.0038505	0	1	0	1	0	1000ΩNi0.006178	1	1	0	1	0
1000ΩPt0.00385055	0	1	0	1	1	10000ΩPt0.003850	1	1	0	1	1
100ΩPt0.003916	0	1	1	0	0	10ΩCu0.004270	1	1	1	0	0
200ΩPt0.003916	0	1	1	0	1	150ΩFS电阻	1	1	1	0	1
500ΩPt0.003916	0	1	1	1	0	300ΩFS电阻	1	1	1	1	0
1000ΩPt0.003916	0	1	1	1	1	600ΩFS电阻	1	1	1	1	1

¹ 除Cu10ohm以外，当各RTD为表中对应的电阻值时，其表示的温度为0°C。Cu10ohm在10ohm时，表示温度为25°C，在9.035ohm时表示温度为0°C。

表A-31 设置RTD DIP 开关

开关6	断线检测/ 超出范围	设置	描述
SW6  配置 ↑1 - 接通 ↓0 - 断开	正向标定 (+3276.7度)	0	指示断线或超出范围的正极
	负向标定 (-3276.8度)	1	指示断线或超出范围的负极
开关7	温度范围	设置	描述
SW7  配置 ↑1 - 接通 ↓0 - 断开	摄氏度 (°C)	0	RTD模块可报告摄氏温度或华氏温度，摄氏温度与华氏温度的转换在内部进行。
	华氏度 (°F)	1	
开关8	接线方式	设置	描述
SW8  配置 ↑1 - 接通 ↓0 - 断开	3线	0	RTD模块与传感器的接线有3种方式（如图所示）。精度最高的是4线连接。2线连接精度最低，推荐只用于可忽略接线误差的应用场合。
	2线或4线	1	



图A-22 RTD到传感器的接线，4线、3线和2线

EM231RTD状态指示器

RTD模块提供PLC温度或出错类型的数据字、状态位指示输入范围错误和用户电源/模块故障。LED指示模块状态，用户程序也可检测出相应错误状态并采取相应的措施。热电偶状态指示器见表A-26。表A-32给出了EM231RTD模块提供的状态指示器。



提示

通道的数据格式是2的补码，16位字，表示温度的单位为0.1度（例如测量的温度为100.2度，则报告数据为1002），电阻数据标度到27648。例如，全量程电阻范围的75%报告为20736。

($225\Omega/300\Omega \times 27648 = 20736$)

表A-32 EM231RTD状态指示器

出错类型	通道数据	SF 指示灯 红色	24V指示灯 绿色	范围状态位1	24VDC用户电源故障 2
没有出错	转换数据	OFF	ON	0	0
24V丢失	32766	OFF	OFF	0	1
SW断线检测	-32768/32767	闪烁	ON	1	0
超出输入范围	-32768/32767	闪烁	ON	1	0
诊断出错3	0000	ON	OFF	0	注释3

1 范围状态位是模块出错寄存器字节中的位3（SMB9用于模块1，SMB11用于模块2……）

2 用户电源故障状态位是出错寄存器字节中的位2（SMB9，SMB11等。参阅附录D）

3 诊断出错引起模块组态错误。在模块组态错误之前，用户电源故障状态位可能设置或没有设置。

如PLC已读到数据，则需405ms更新所有通道的数据，如在一个更新时间内，PLC没有读数据，则模块报告原有的数据一直到PLC读数据后的下一次模块更新，为了保持通道数据总是为当前值，建议PLC读数据的频度至少和模块更新频率相同。



提示

如正在使用热电阻模块，应禁止在PLC中使用模拟量滤波，在以定时方式进行检查时，模拟量滤波会妨碍出错条件的检测。

断线检测是由软件在RTD模块内部完成的。设置了SMB范围状态位，并通过开关设置设定了通道数据的正负向标定之后，超限输入和检测到的断线情况就能通过信号指示出来了。断线检测至少需要三个模块扫描周期或更长时间，这通常取决于具体的断线类型。Source+和Source-的断线检测通常需要的时间最小，而Sense+或Sense-则需要5秒钟或更长的时间来检测。在电气噪声严重的环境中，间歇地检测到断线时，开路的Sense线上（测量线）也会有随机的有效数据出现。电气噪声会延长断线检测的时间。为此建议，在程序收到有效数据后，还应在应用程序中对断线检测/超输入范围的状态指示进行监控及锁定。



提示

如果有未用的通道，可以在此通道上接一个电阻来代替RTD，以防止因为断线检测引起的SF LED闪烁。电阻值必须和RTD的标称值相同。例如，PT100RTD需使用100Ω的电阻值。

EM231RTD模块的范围

EM231RTD的温度范围和每个类型的RTD模块的精度在表A-33和A-34中给出。

表A-33 各种热电阻的温度范围(°C)和精度

系统字 (1个数字=0.1°C)		Pt10000	Pt100, Pt200, Pt500, Pt1000	Ni100, Ni120, Ni1000 ¹	Cu10	0 - 150Ω	0 - 300Ω	0 - 600Ω	
10进制	16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			↑	312.0°C				
2950	0B86			↑	295.0°C				
2600	0A28				260.0°C				
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	(负值是不可能的)			
-600	FDA8					↓	↓	↓	
-1050	FBE6								N
-2000	F830	-200.0°C	-200.0°C						R
-2400	F6A0								
-2430	F682	-243.0°C	-243.0°C						
-5000	EC78								
-6000	E890								
-10500	D6FC								UR
-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(正常范围)		±4°C	±1°C	±0.6°C	±2.8°C	+0.15W	+0.3W	+0.6W	

*OF=下溢OR=超出范围NR=正常范围; UR=低于范围UF=下溢

↑或↓表示所有超限的模拟量值报告所选定的超出范围的值, 32767(0x7FF)或-32768(0x8000)。

¹ alpha值为0.006178的1000ΩNi的正常范围的下限是0摄氏度, 并且没有低于范围的值。alpha值为0.00672的1000ΩNi在此表中列出。

表A-34 RTD类型的温度范围 (°F)

系统字 (1个数字=0.1°F)		PT1000	Pt100, Pt200, Pt500, Pt1000	Ni100, Ni120, Ni1000 ¹	Cu10	
10进制	16进制					
32767	7FF.					
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	
						额定范围
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		-328.0°F	
-4000	F060				-400.0°F	
-4054	F02A	-405.4°F	-405.4°F			
		↓	↓			
-5000	EC78					
-6000	E890					↓ 低于范围
-10500	D6FC					
-32767	8001					
-32768	8000					

↑或↓表示所有超限的模拟量值报告选择的超出范围的值, 32767 (0x7FFF) 或-32768 (0x8000)。

¹ alpha值为0.006178的1000ΩNi的正常范围的下限是32华氏度, 并且没有低于范围的值。alpha值为0.00672的1000ΩNi在此表中列出。

EM277 PROFIBUS-DP模板规范

表A-35 EM277 PROFIBUS-DP模块订货号

订货号	扩展模块	输入	输出	可拆卸连接
6ES7277-0AA22-0XA0	EM277 PROFIBUS-DP	-	-	否

表A-36 EM277 PROFIBUS-DP模块常规规范

订货号	模块名称及描述	尺寸 (mm) (W x H x D)	重量	功耗	+5 VDC	VDC要求 +24 VDC
6ES7277-0AA22-0XA0	EM277 PROFIBUS-DP	71x80x62	175g	2.5W	150mA	见下文

表A-37 EM277 PROFIBUS-DP模板规范

常规	6ES7277-0AA22-0XA0
端口数 (有限电源)	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500K波特; 1M、1.5M、3M、6M和12M波特
协议	PROFIBUS-DP从站和MPI从站
电缆长度	
低于93.75波特	1200m
187.5波特	1000m
500波特	400m
1M到1.5M波特	200m
3M到12M波特	100m
连网能力	
站地址设置	0-99 (由旋钮开关设定)
每段最大站数	32
每个网络最大站数	126, 最多99个EM277站
MPI连接	一共6个, 2个保留 (1个给PG, 1个给OP)
输入电源要求	
电压范围	20.4 ~ 28.8VDC (等级2, 开关电源, 或来自PLC的传感器电源)
最大电流	
仅当模块端口激活时	30 mA
加90mA, 5V端口负载	60 mA
加120mA, 24V端口负载	180 mA
纹波噪声 (<10 MHz)	<1V峰-峰值 (最大)
隔离 (现场到逻辑) ¹	500 VAC, 1分钟
通讯口的5VDC电源	
每个口的最大电流	90mA
隔离 (24VDC到逻辑)	500 VAC, 1分钟
通讯口的24VDC电源	
电压范围	20.4-28.8 VDC
每个口的最大电流	120mA
电流限定	0.7-2.4A
隔离	无隔离, 与输入24VDC电路相同

¹ 24VDC电源不对模块逻辑供电, 用于通讯口的24 VDC电源。

支持智能模块的CPU

EM277 PROFIBUS-DP从站模块是一种智能扩展模块，可与表A-38中的S7-200 CPU连接。

表A-38 EM277 PROFIBUS-DP模块对S7-200 PLC的兼容性

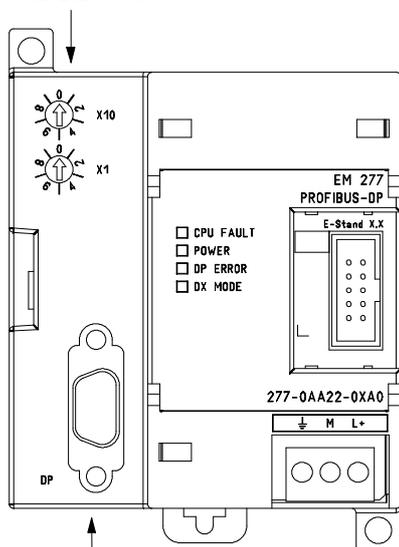
CPU	描述
CPU222版本1.10或更高	CPU222DC/DC/DC和CPU222AC/DC/继电器
CPU224版本1.10或更高	CPU224DC/DC/DC和CPU224AC/DC/继电器
CPU224XP 版本2.0或更高	CPU224XP DC/DC/DC和CPU224XP AC/DC/继电器
CPU226版本1.00或更高	CPU226 DC/DC/DC和CPU226 AC/DC/继电器

地址开关和LED

地址开关和状态LED位于模块的正面，见图A-23。DP从站的接口插针输出也如下图所示。对于状态LED的描述，见表A-42。

EM277 PROFIBUS-DP的前视图

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

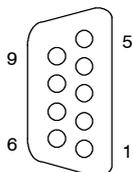


DP从站端口连接器

9针Sub D连接器的引脚

针脚描述

9针D型插座



- 1 外壳地，连接到连接器的外壳上
- 2 24V返回（同接线端子排上的M）
- 3 隔离的信号B（RxD/TxD+）
- 4 隔离的发送请求（TTL电平）
- 5 隔离的+5V返回
- 6 隔离的+5V（最大90mA）
- 7 +24V（最大120mA，带反向电压保护二极管）
- 8 隔离信号A（RxD/TxD-）
- 9 无连接

注意：隔离意味着对数字量逻辑和24V输入电源有500V的隔离。

图A-23 EM277 PROFIBUS-DP

分布式（DP）外围设备的标准通信

PROFIBUS-DP（或DP标准）是由欧洲标准EN 50170定义的一种远程I/O通信协议。遵守这种标准的设备，即使是由不同公司制造的，也是兼容的。DP表示分布式外围设备，亦即远程I/O。用于过程现场总线的PROFIBUS标准。

在下列通讯协议标准中，EM277 PROFIBUS-DP模块将作为从站器件来实现DP标准协议：

- EN 50170（PROFIBUS）描述总线访问和传送协议，并规定数据传送介质的性能。
- EN 50170（DP标准）描述DP主站和DP从站之间的高速循环交换数据。这个标准规定组态和参数赋值过程，解释具有分布式I/O功能的循环数据如何进行交换，并列出了支持的诊断选择。

一个DP主站组态应包含地址，从站类型以及从站所需要的任何参数赋值信息。还应告诉主站由从站（输入）读入的数据应放置何处，以及从何处获得写入从站（输出）的数据。DP主站建立网络，然后初始化其DP从站。主站将参数赋值信息和I/O组态写入到从站。然后，主站从从站那里读出诊断信息，并验证DP从站已接受参数和I/O组态。然后，主站开始与从站交换I/O数据。每次对从站的数据交换为写输出和读输入。这种数据交换方式无限期地继续下去。如果有意外情况发生，从站器件可以通知主站，而主站就会读取来自从站的诊断信息。

一旦DP主站已将参数和I/O组态写入到DP从站，而且从站已从主站那里接收到参数和组态，则主站就拥有那个从站。从站只能接收来自其主站的写请求。网络上的其它主站可以读取该从站的输入和输出，但是它们不能向该从站写入任何信息。

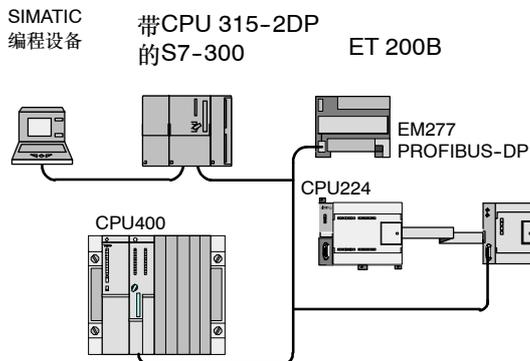
使用EM277将S7-200 CPU作为DP从站连接到网络

通过EM277 PROFIBUS-DP扩展从站模块，可将S7-200 CPU连接到PROFIBUS-DP网络。EM277经过串行I/O总线连接到S7-200 CPU。PROFIBUS网络经过其DP通信端口，连接到EM277 PROFIBUS-DP模块。这个端口可运行于9600波特和12M波特之间的任何PROFIBUS波特率。关于EM277 PROFIBUS-DP模块支持的波特率，可参见该模块的规范。

作为DP从站，EM277模块接受从主站来的多种不同的I/O组态，向主站发送和接收不同数量的数据。这种特性使用户能修改所传输的数据量，以满足实际应用的需要。与许多DP站不同的是，EM277模块不仅仅是传输I/O数据。EM277能读写S7-200 CPU中定义的变量数据块。这样，使用户能与主站交换任何类型的数据。首先将数据移到S7-200 CPU中的变量存储器，就可将输入、计数值、定时器值或其它计算值送到主站。类似地，从主站来的数据存储在S7-200 CPU中的变量存储器内，也可移到其它数据区。

EM277 PROFIBUS-DP模块的DP端口可连接到网络上的一个DP主站上，但仍能作为一个MPI从站与同一网络上如SIMATIC编程器或S7-300/S7-400CPU等其它主站进行通信。图A-24给出了一个带CPU 224和EM277 PROFIBUS-DP模块的PROFIBUS网络。

- ❑ CPU315-2是DP主站，并用带有STEP7编程软件的SIMATIC编程设备进行了组态。
- ❑ CPU-315-2是DP主站，并且已通过一个带有STEP7编程软件的SIMATIC编程器进行组态。CPU224是CP315-2所拥有的一个DP从站，ET200I/O模块也是CPU315-2的从站。
- ❑ S7-400CPU连接到PROFIBUS网络，并且借助于S7-400CPU用户程序中的XGET指令，可从CPU224读取数据。

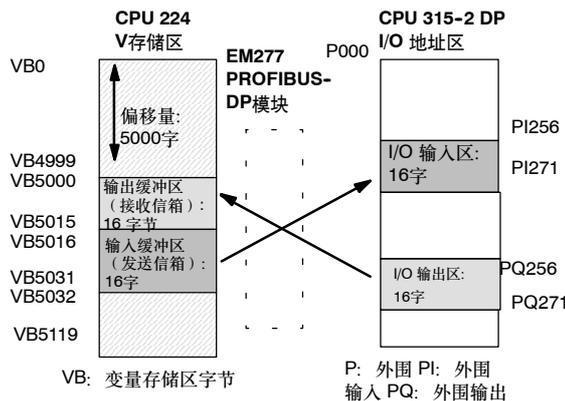


图A-24 PROFIBUS网络中EM277 PROFIBUS-DP模块和CPU224

组态

为了将EM277作为一个DP从站使用，用户必须设定与主站组态中的地址相匹配的DP端口地址。从站地址是使用EM277模块上的旋转开关设定的。在为新的从站地址按照顺序进行了开关改变以后，若要使改变生效，您必须对CPU循环上电。

主站通过将其输出区来的信息发送给从站的输出缓冲区（称为“接收信箱”），与其每个从站交换数据。从站将其输入缓冲区（称为“发送信箱”）的数据返回给主站的输入区，以响应从主站来的信息。



图A-25 V存储器和I/O地址区域

图A-25为PROFIBUS-DP主站的V存储器及I/O地址区域模型。

EM277可用DP主站组态，以接收从主站来的输出数据，并将输入数据返回给主站。输出和输入数据缓冲区驻留在S7-200 CPU的变量存储器（V存储器）内。当用户组态DP主站时，应定义V存储器内的字节位置。从这个位置开始为输出数据缓冲区，它应作为EM277的参数赋值信息的一个部分。用户也要定义I/O组态，它是写入到S7-200 CPU的输出数据总量和从S7-200 CPU返回的输入数据总量。EM277从I/O组态确定输入和输出缓冲区的大小。DP主站将参数赋值和I/O组态信息写入到EM277 PROFIBUS-DP模块，然后，EM277将V存储器地址和输入及输出数据长度传送给S7-200 CPU。

图A-25表示CPU224中的V存储器的一个存储器模型，以及一个DP主站CPU的I/O地址区。在这个例子中，DP主站已定义了16输出字节和16输入字节的一种I/O组态，以及V存储器偏移为5000。CPU224中的输出缓冲区和输入缓冲区长度（由I/O组态确定）都是16字节。输出数据缓冲区从V5000开始；输入缓冲区紧紧跟随输出缓冲区，并在V5016处开始。输出数据（从主站来）放置在V存储器中的V5000。输入数据（传送到主站）取自V存储器的V5016。

**提示**

如果处理的数据单位（一致性数据）是3字节，或数据单位（一致性数据）大于4字节，则必须使用SFC14，以便读出DP从站的输入，并使用SFC-15，以便对DP从站的输出进行编址。详细情况见S7-300和S7-400系统的系统软件和标准功能参考手册。

表A-39列出由EM277 PROFIBUS-DP模块支持的组态。EM277模块的默认组态是输入两个字，输出两个字。

表A-39 EM277组态选项

组态	输入到主站	从主站输出	数据的一致性
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字	2字	字节一致性
14	8字	8字	
15	32字	32字	
16	64字	64字	
17	4字	4字	缓存区一致性
18	8字	8字	
19	12字	12字	
20	16字	16字	

输入和输出缓存区的地址可以配置在S7-200 CPU V存储器中的任何位置。输入和输出缓冲器的缺省值地址为VB0。输入和输出缓冲地址是主站写入S7-200 CPU赋值参数信息的一部分，用户必须组态主站以识别所有的从站以及将需要的参数和I/O组态写入每一个从站。

使用以下工具以组态DP主站：

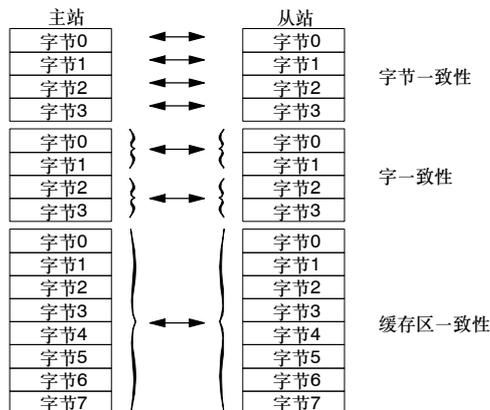
- 对于SIMATIC S5主站，使用COM PROFIBUS Windows软件
- 对于SIMATIC S7主站，使用STEP 7编程软件
- 对于SIMATIC 505主站，使COM PROFIBUS和TISOFT2或Softshop两种软件之一

关于使用这些组态和编程软件的详细信息，请参阅这些软件工具的使用手册，关于PROFIBUS网络和其部件的详细信息，请参阅ET 200分布式I/O系统使用手册。

数据的一致性

PROFIBUS支持三类数据一致性:

- 字节一致性保证了字节作为整个单元进行传送。
- 字一致性保证字的传送不会被CPU中的其它处理所中断。就是说，组成字的二个字节总是一起移动，不会被拆散。如果正被传送的数据值是整数，则使用字一致性。



图A-26 字节、字和缓冲区数据的一致性。

- 缓冲区一致性保证数据的整个缓冲区是作为一个单独单元传送的，不会被CPU的任何其它处理过程中断。如果数据值是双字或者浮点数，或者当一组值和一个计算或条目相关时，就应该使用缓冲区一致性。

用户将数据一致性设置成主站中的I/O组态部分。数据一致性选择作为从站的初始化部分写入DP从站。DP主站和DP从站都利用数据一致性选择，以便保证数据值（字节、字或缓冲区）在主站和从站内的传送是不中断的。图A-26表示不同类型的一致性。

用户程序需考虑的事项

一旦EM277 PROFIBUS-DP模块已用一个DP主站成功地进行了组态，EM277和DP主站就进入数据交换模式。在数据交换模式中，主站将输出数据写入到EM277 PROFIBUS-DP模块，然后，EM277模块响应最新的S7-200 CPU输入数据。EM277模块不断地更新其从S7-200 CPU来的输入，以便向DP主站提供最新的输入数据。然后，该模块将输出数据传送给S7-200 CPU。从主站来的输出数据放在V存储器中（输出缓冲区）由某地址开始的区域内，而该地址是在初始化期间，由DP主站所提供的。到主站的输入数据在输出数据之后立刻从V存储区（输入缓冲区）中提取出来。

从主站来的输出数据必须通过S7-200 CPU中的用户程序，从输出缓冲区转移到其它所用的数据区。类似地，传送到主站的输入数据也必须通过用户程序从各种数据区转移到输入缓冲区，进而发送到DP主站。

从DP主站来的输出数据，在执行程序扫描后立即放置在V存储区。输入数据（传送到主站）从V存储器复制到EM277中，以便同时传送到主站。

当主站提供有新的数据时，则从主站来的输出数据才写入到V存储器内。

在下次与主站交换数据时，将送到主站的输入数据发送到主站。

在建立S7-200 CPU用户程序时，必须知道V存储器中的数据缓冲区的开始地址和缓冲区大小。

状态信息

基于其物理位置，每个智能模块都分配有50个字节的专用存储区（SM）。模块按照它与CPU的相对位置，刷新SM区域。如果它是第一个智能模块，则刷新SMB200到SMB249。如果是第二个智能模块，那么，它刷新SMB250到SMB299，等等。参见表A-40。

表A-40 特殊存储字节SMB200-SMB549

特殊存储字节SMB200-SMB549						
智能模块在0号槽	智能模块位于槽1	智能模块位于槽2	智能模块位于槽3	智能模块槽4	智能模块位于槽5	智能模块在6号槽
SMB200-SMB249	SMB250-SMB299	SMB300-SMB349	SMB350-SMB399	SMB400-SMB449	SMB450-SMB499	SMB500-SMB549

如果DP尚未建立与主站的通信，那么，这些SM存储单元显示缺省值。当主站已将参数和I/O组态写入到EM277 PROFIBUS-DP模块后，这些SM存储单元显示DP主站的组态设置。表A-41列出的是SM区域中的信息，在使用这些信息或V存储器缓冲区中的数据之前，您应该检查协议状态字节（例如用于插槽0的SMB224），以确保EM277正处于和主站的数据交换模式。



提示

用户不能通过写入SM存储单元来组态EM277 PROFIBUS-DP I/O缓冲区的大小，或缓冲区的位置。只有DP主站才可以组态运行于DP方式下的EM277 PROFIBUS-DP模块。

表A-41 EM277 PROFIBUS-DP特殊存储字节

智能模块在0号槽	...	智能模块在6号槽	描述																															
SMB 200至SMB 215	...	SMB 500至SMB 515	模块名（16ASCII字符） “EM277 Profibus DP”																															
SMB 216至SMB 219	...	SMB 516至SMB 519	S/W版本号（4ASCII字符）																															
SMW 220	...	SMW 520	错误代码 16 # 0000 无错误 16 # 0001 无用户电源 16 # 0002至16 # FFFF 保留																															
SMB 222	...	SMB 522	DP从模块的站地址，由地址开关（0-99十进制）设定																															
SMB 223	...	SMB 523	保留																															
SMB 224	...	SMB 524	DP 标准协议状态字节 <table border="1" style="margin-left: 20px;"> <tr> <td colspan="6" style="text-align: center;">MSB</td> <td colspan="2" style="text-align: center;">LSB</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">S1</td> <td style="text-align: center;">S0</td> </tr> </table> <table border="1" style="margin-left: 20px;"> <tr> <td style="text-align: center;">S1</td> <td style="text-align: center;">S0</td> <td>DP标准状态字节描述</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>上电后未启动DP通讯</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>检测到的组态/参数分配错误</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>当前处于数据交换模式</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>退出数据交换模式</td> </tr> </table>	MSB						LSB		0	0	0	0	0	0	S1	S0	S1	S0	DP标准状态字节描述	0	0	上电后未启动DP通讯	0	1	检测到的组态/参数分配错误	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标准状态字节描述																																
0	0	上电后未启动DP通讯																																
0	1	检测到的组态/参数分配错误																																
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 - 从站的主站地址（0至126）																															
SMW226	...	SMW526	DP标准协议 - 输出缓冲区的V存储器地址，作为从VB0开始的输出缓冲区的偏移量																															
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-电源接通时清除																															

注意：每当DP从站模块接收组态/参数化信息时更新SM区域。即使探测出一个组态/参数化错误时，这些存储单元也要更新。每次电源接通时，这些存储单元被清除。

EM277 PROFIBUS-DP模块LED指示灯

EM277 PROFIBUS-DP模块在上面的面板上有四个状态LED，用来指示DP端口的运行状态：

- S7-200上电后，DX MODE灯一直熄灭直到DP通讯开始。
- 当DP的通讯成功地初始化后（EM277 PROFIBUS-DP模块进入和主站交换数据的状态时），DX MODE灯变绿直到数据交换状态结束。
- 如果DP通讯中断，强迫EM277模块退出数据交换模式，此时，DX MODE灯熄灭而DP ERROR灯变红。此状态一直保持到S7-200 CPU断电或数据交换重新开始。
- 如果主站写入EM277模块的I/O组态或参数信息错误，则DP ERROR的红灯将闪烁。
- 如果没有24VDC供电，POWER（电源）灯将熄灭。

表A-42总结了EM277状态指示器的各种状态。

表A-42 EM277位控模块状态LED

LED	OFF	红色	红灯闪烁	绿色
CPU 故障	模块良好	内部模块故障	-	-
POWER	没有24VDC用户电源	-	-	24VDC用户电源良好
DP ERROR	没有错误	脱离数据交换模式	参数化/组态错误	-
DX MODE	不在数据交换模式	-	-	在数据交换模式

注意：当EM277 PROFIBUS-DP模块专用作MPI从站时，仅绿色电源LED接通。

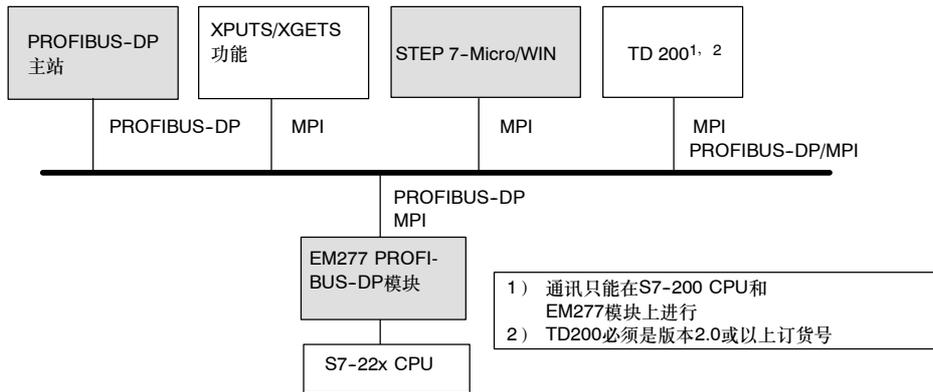
附加的组态特性

EM277 PROFIBUS-DP模块可作为连接到其它MPI主站的通信接口，而不论该模块是否用作PROFIBUS-DP从站。该模块利用S7-300/400的XGET/XPUT功能，提供从S7-300/400到S7-200的连接。STEP 7-Micro/WIN以及使用MPI或PROFIBUS参数集的网卡（例如CP5611），还有OP设备或者TD 200（发行版本2.0或更高，订货号6ES7 272-0AA20-0YA0）都可以通过EM277 PROFIBUS-DP模块和S7-200进行通讯。

除DP主站外，最多可以有6个连接（6个设备）与EM277 PROFIBUS-DP模块相连接。一个连接是为编程器（PG）而保留的，一个连接是为操作员面板（OP）而保留的。其它4个连接可被任何一个MPI主站使用。为了使EM277 PROFIBUS-DP模块与多个主站进行通信，所有主站都必须在相同的波特率下运行。见图A-27是一个可能的网络配置。

当EM277 PROFIBUS-DP模块用于MPI通信时，MPI主站必须使用EM277模块的站址向S7-200 CPU发送信息。发送给EM277 PROFIBUS-DP模块的MPI信息将通过EM277传送给S7-200 CPU。

EM277 PROFIBUS-DP模块是一种从站模块，不能用来通过NETR和NETW语句进行不同的S7-200 PLC之间的通讯。EM277 PROFIBUS-DP模块不能用于自由端口的通信，尽管S7-200本机上的通信端口都具有这种通讯功能。



图A-27 PROFIBUS-DP/MPI网络

设备数据库文件: GSD

不同的PROFIBUS设备有不同的性能特性。这些特性就功能（例如，I/O信号的数量和诊断信息）或总线参数（例如，传输速度和时间监视）而言是不同的。这些参数对每个设备类型和供应商来说都是不同的，而且通常汇编在技术手册内。为了帮助用户完成PROFIBUS的简单组态，通常把包含特定设备性能参数的电子表格称为设备数据库文件，即GSD文件。基于GSD文件的组态工具允许将不同供应商的设备简单地集成到一个单一网络中。

设备数据库文件以精确的格式提供对设备特性的全面描述。这些GSD文件是供应商为每种类型设备而准备并提供给PROFIBUS用户的。GSD文件能使组态系统读入PROFIBUS设备的特性，并在组态系统时利用这个信息。

COM PROFIBUS或STEP 7软件的最新版本包括EM277 PROFIBUS-DP模块的组态文件。如果您的软件版本不包括用于EM277的组态文件，您可在网址www.profibus.com下载最新的GSD文件(SIEM089D.GSD)。

如果您正在使用一个非西门子的主站，可参考由制造商提供的文件，了解如何用GSD文件组态主站。

```

;=====
;GSD File for the EM 277 PROFIBUS-DP with a DPC31
;MLFB:6ES7 277-0AA2.-0XA0
;DATE:26-March-2001
;=====
#Profibus_DP
;General parameters
GSD_Revision      = 1
Vendor_Name       = "Siemens"
Model_Name        = "EM277 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

;Slave-Specification:
OrderNumber="6ES7 277-0AA2.-0XA0"
Periphery="SIMATIC S5"
Slave_Family=10@Tdf@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-Definition
ExtUserPrmData=1 "I/O Offset in the V-memory"
Unsigned16 0 0-10239
EndExtUserPrmData
; UserPrmData[]Length and Preset:
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
;=====
; Continuation of GSD File
;=====
; Module Definition List
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

```

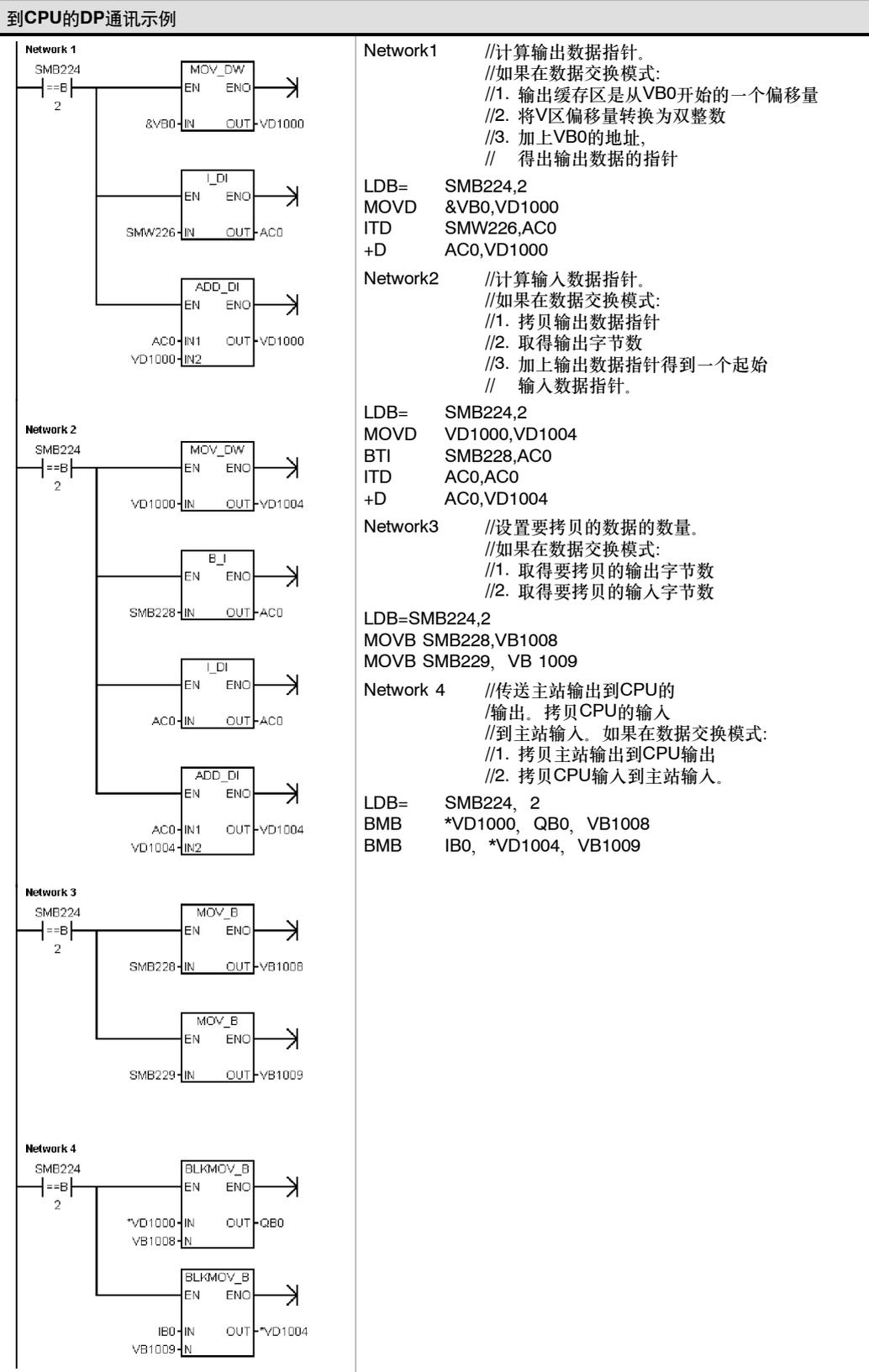
图A-28 EM277 PROFIBUS模块的GSD文件列表

CPU的DP通信的示例程序

以下是一个用语句表生成的CPU的例子程序，PROFIBUS-DP模块位于其0号槽，它使用如下所示的SM存储器中的DP端口信息。这个程序由SMW 226确定DP缓冲区的地址，由SMB 228和SMB 229确定了DP缓冲区的大小。程序使用这些信息以复制DP输出缓冲器中的数据到CPU224的过程映像输出寄存器。类似地，CPU的过程映像输入寄存器中的数据也被复制到V存储器输入缓冲区中。

以下示例程序中，DP模块位于0号槽，SM中的DP组态信息提供了DP从站的组态。程序使用以下数据：

```
//SMW220  DP模块出错状态
//SMB224  DP状态
//SMB225  主站地址
//SMW226  V存储器中输出的偏移
//SMB228  输出数据的字节数
//SMB229  输入数据的字节数
//VD1000  输出数据的指针
//VD1004  输入数据的指针
```



EM241 Modem模块规范

表A-43 EM241 Modem模块订货号

订货号	扩展模块	输入	输出	可拆卸连接
6ES7241-1AA22-0XA0	EM241 Modem模块	-	8 ¹	否

¹ 八个Q输出用作Modem功能的逻辑控制，不直接控制任何外部信号。

表A-44 EM241 Modem模块常规规范

订货号	模块名称及描述	尺寸 (mm) (W x H x D)	重量	功耗	+5 VDC	VDC要求 +24 VDC
6ES7 241-1AA22-0XA0	EM241 Modem模块	71.2 x 80 x 62	190g	2.1W	80 mA	70mA

表A-45 EM241 Modem模块规范

常规	6ES7 241-1AA22-0XA0
电话连接	
隔离 (电话线到逻辑和现场)	1500 VAC (电的)
物理连接	RJ11 (6位4线)
Modem标准	Bell103, Bell212, V.21, V.22bis, V.23c, V.32, V.32bis, V.24 (缺省)
安全特性	密码 回拨
拨号	脉冲或语音
信息协议	数字 TAP (字母及数字) UCP命令1,30,51
工业协议	Modbus PPI
输入电源要求	
电压范围	20.4-28.8 VDC
隔离 (现场电源到逻辑)	500 VAC, 1分钟

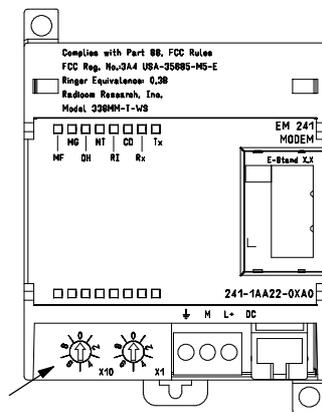
EM241 Modem模块替代连于CPU通讯口的外部Modem功能。与一个连有EM 241的S7-200系统进行通讯，您只需在远端的个人计算机上连接一个外置Modem并安装STEP7-Micro/WIN。

对于组态的信息，参见第7章“通过网络进行通讯”。对于编程和模块的高级特性，参见第10章“创建调制解调模块程序”。



Modem扩展

可以使用STEP 7-Micro/WIN Modem扩展向导去组态EM241 Modem模块。对于Modem扩展向导的更多信息，参见第10章。



国家代码开关

图A-29

EM241 Modem模块接线端子图

支持智能模块的S7-200 CPU

EM241 Modem模块是一种与S7-200 CPU一起工作的智能扩展模块，其适用的CPU如表A-46所示。

表A-46 EM241 Modem 模块对S7-200 CPU的兼容性

CPU	描述
CPU 222版本1.10或更高	CPU222 DC/DC/DC和CPU222 AC/DC/继电器
CPU 224版本1.10或更高	CPU224 DC/DC/DC和CPU224 AC/DC/继电器
CPU224XP 版本2.0或更高	CPU224XP DC/DC/DC和CPU224XP DC/DC/继电器
CPU 226版本1.00或更高	CPU226 DC/DC/DC和CPU226 AC/DC/继电器

安装EM 241

按以下步骤安装EM 241:

1. 将EM 241安装在DIN导轨上并插上扁平电缆。
2. 从CPU传感器电源或外部电源连接24VDC，装接地端连到系统的地。
3. 将电话线连至RJ11插座。
4. 按照表A-47设置国家代码。为了能够读取正确的国家代码，在CPU上电之前，必须设置这些开关。
5. CPU上电。绿色的MG（模板好）灯应接通。

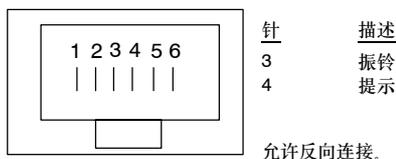
现在EM 241已为通讯作好准备。

表A-47 国家代码被EM 241支持

代码	国家	电话标准
00	澳大利亚	ACATS-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	PTC 200
27	挪威	CTR21
30	波兰	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 插座

图A-30所示为RJ11插座的详细图示。可以通过适配器与其它标准的电话接口相连。对于更多信息，请参考您的适配器连接器文档。



图A-30 RJ11插座的视图

小心

雷击或其它不可预期的高压作用于电话线上会损坏EM241 Modem模块。

使用经济实用的电话线冲击保护装置，比如常见的用于个人计算机Modem的保护装置。冲击保护装置为保护EM241 Modem模块，可能会损坏。选择一个能够正面指示它在工作的冲击保护装置。

定期检查冲击保护装置以确保EM241 Modem模块能够得到持续的保护。

EM253位控模块规范

表A-48 EM253 位控模块订货号

订货号	扩展模块	输入	输出	可拆卸连接
6ES7 253-1AA22-0XA0	EM253位控模块	-	8 ¹	是

¹ 八位Q输出用作运动功能的逻辑控制，不直接控制任何外部信号。

表A-49 EM253 位控模块常规规范

订货号	模块名称及描述	尺寸 (mm) (W x H x D)	重量	功耗	+5 VDC	VDC要求 +24 VDC
6ES7 253-1AA22-0XA0	EM253位控模块	71.2 x 80 x 62	0.190kg	2.5W	190mA	见下文

表A-50 EM253位控模块规范

常规	6ES7 253-1AA22-0XA0
输入特性	
输入数量	5点
输入类型	漏型/源型 (IEC类型1漏型, 除ZP外)
输入电压	
允许的最大持续电压 STP,RPS,LMT+,LMT-ZP	30 VDC 30 VDC, 20 mA, 最大
浪涌 (所有输入) 额定值 STP,RPS,LMT+,LMT-ZP	35 VDC, 0.5秒 24 VDC, 4mA, 正常 24 VDC, 15mA, 正常
逻辑“1”信号 (最小) STP,RPS,LMT+,LMT-ZP	15 VDC, 2.5mA, 最小 3 VDC, 8.0mA, 最小
逻辑“0”信号 (最大) STP,RPS,LMT+,LMT-ZP	5 VDC, 1mA, 最大 1 VDC, 1mA, 最大
隔离 (现场与逻辑) 光电隔离 组隔离	500 VAC, 1分钟 1点用于STP, RPS和ZP 2点用于LMT+和LMT-
输入延迟时间 STP,RPS,LMT+,LMT-ZP (可计脉冲宽度)	0.2ms-12.8ms, 用户选择 2 μsec, 最小
连接2线接近开关传感器 (Bero) 允许的源电流	1mA, 最大
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
同时接通的输入数	所有 水平安装时低于55°C, 垂直安装时低于45°C

表A-50 EM253位控模块规范

常规	6ES7 253-1AA22-0XA0	
常规		
集成的输出数 输出字节 P0+, P0-, P1+, P1- P0, P1, DIS, CLR	6点 (4个信号) 驱动 漏极输出	
输出电压 P0,P1,RS-422驱动, 差分输出电压* 断路 接入带有200Ω串行电阻的光耦二极管 100Ω负载 54Ω负载 P0, P1, DIS, CLR漏型 建议电压, 开路 允许电压, 开路 漏电流 接通状态电阻 断开状态下漏电流, 30VDC 上拉电阻, 到T1的漏型输出	3.5V典型 2.8V最小 1.5V最小 1.0V最小 5VDC, 来自模块 30VDC ¹ 50mA最大 15Ω最大 10μA最大 3.3KΩ ²	
输出电流 输出组数 同时接通的输出 每点漏电流 P0, P1, DIS, CLR 过载保护	1 所有 水平安装时低于55°C, 垂直安装时低于45°C。 10μA最大 否	
隔离 (现场与逻辑) 光电隔离	500 VAC, 1分钟	
输出时延 DIS, CLR: 断开到接通/接通到断开	30μs, 最大	
脉冲畸变 P0, P1, 输出, RS-422驱动, 100 Ω外部 负载 P0, P1输出, 漏型, 5V/470Ω外部 负载	75ns最大 300ns最大	
切换频率 P0+, P0-, P1+, P1-, P0和P1	200kHz	
电缆长度 未屏蔽 屏蔽	不推荐 10米	
电源		
L+提供电压 逻辑提供输出 L+ 供电电流对应5 VDC 负载 负载电流 0mA (无负载) 200mA (额定负载)	11 ~ 30 VDC (等级2, 开关电源, 或来自PLC的传感器电源) +5 VDC+/-10%, 200mA最大	
	<u>12VDC输入</u> 120mA 300mA	<u>24VDC输入</u> 70mA 130mA
隔离 L+电源到逻辑 L+电源到输入 L+电源到输出	500 VAC, 1分钟 500 VAC, 1分钟 无	
反向极性	L+输入和+5V输出有二极管保护。在M端接入正向电压, 就输出点的连接而言, 可能导致损害性的电流产生。	

¹ 高于5VDC的漏型输出可能会增加射频干扰使之超过允许的限定。您的系统或接线需要射频干扰抑制措施。

² 根据您的脉冲接收器和电缆, 一个额外的外部上拉电阻可能会改善脉冲信号的质量和噪声抑制功能。

支持智能模块的CPU

EM253 位控模块是用来与表A-51中列出的S7-200 CPU一起工作的智能扩展模块。

表A-51 EM253位控模块和S7-200 CPU的兼容性

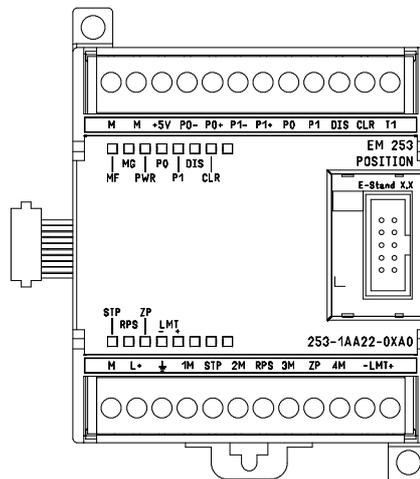
CPU	描述
CPU222版本1.10或更高	CPU222 DC/DC/DC和CPU222 AC/DC/继电器
CPU224版本1.10或更高	CPU224 DC/DC/DC和CPU224 AC/DC/继电器
CPU224XP 版本2.0或更高	CPU224XP DC/DC/DC和CPU224XP DC/DC/继电器
CPU226版本1.00或更高	CPU226 DC/DC/DC和CPU226 AC/DC/继电器

EM253位控模块状态LED

位控模块的状态LED如表A-52所示。

表A-52 位控模块状态LED

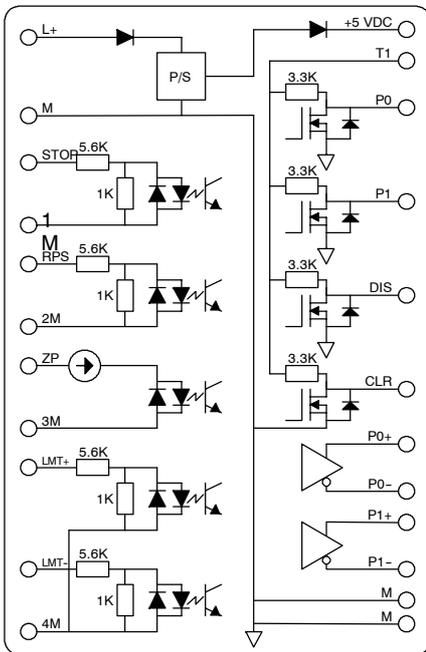
本地I/O	LED	颜色	功能描述
-	MF	红色	模板检测到一个致命故障时接通
-	MG	绿色	无故障时接通, 检测到组态错误时以1Hz频率闪烁
-	PWR	绿色	当L+和M端有24VDC供电时接通
输入	STOP	绿色	输入stop接通时亮
输入	RPS	绿色	参考点切换输入接通时亮
输入	ZP	绿色	零脉冲输入接通时亮
输入	LMT-	绿色	负向限位输入接通时亮
输入	LMT+	绿色	正向限位输入接通时亮
输出	P0	绿色	P0输出触发时亮
输出	P1	绿色	P1输出触发或该输出指示正向运动时亮
输出	DIS	绿色	DIS输出激活时亮
输出	CLR	绿色	当清除偏差计数器输出激活时亮



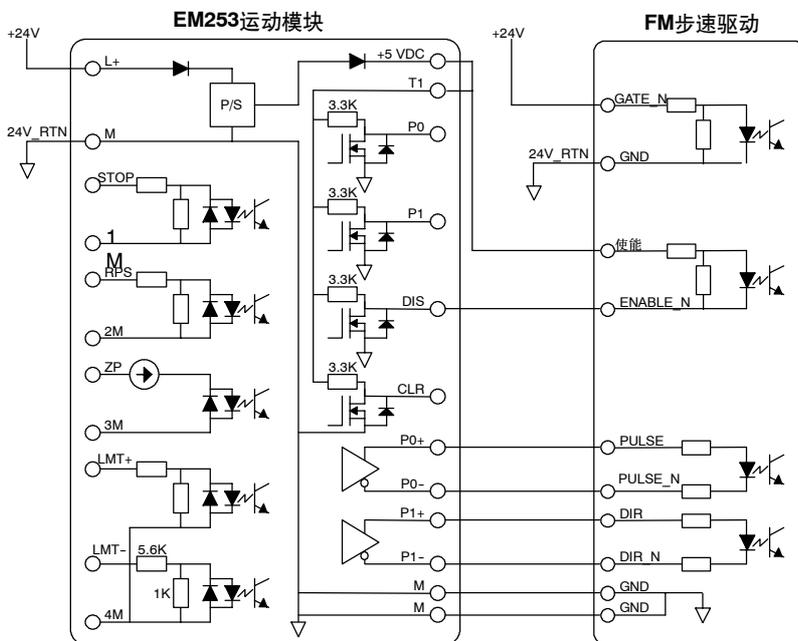
图A-31 EM253位控模块

接线图

下图中各端子没有按序排列。端子的排列请参见图A-31。

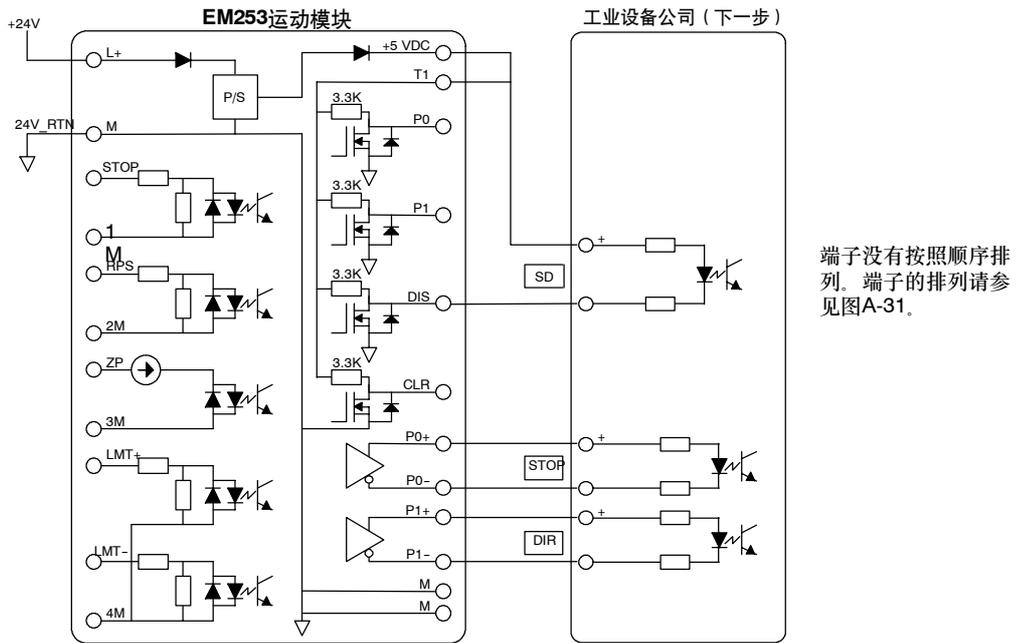


图A-32 EM253位控模块输入和输出内部示意图

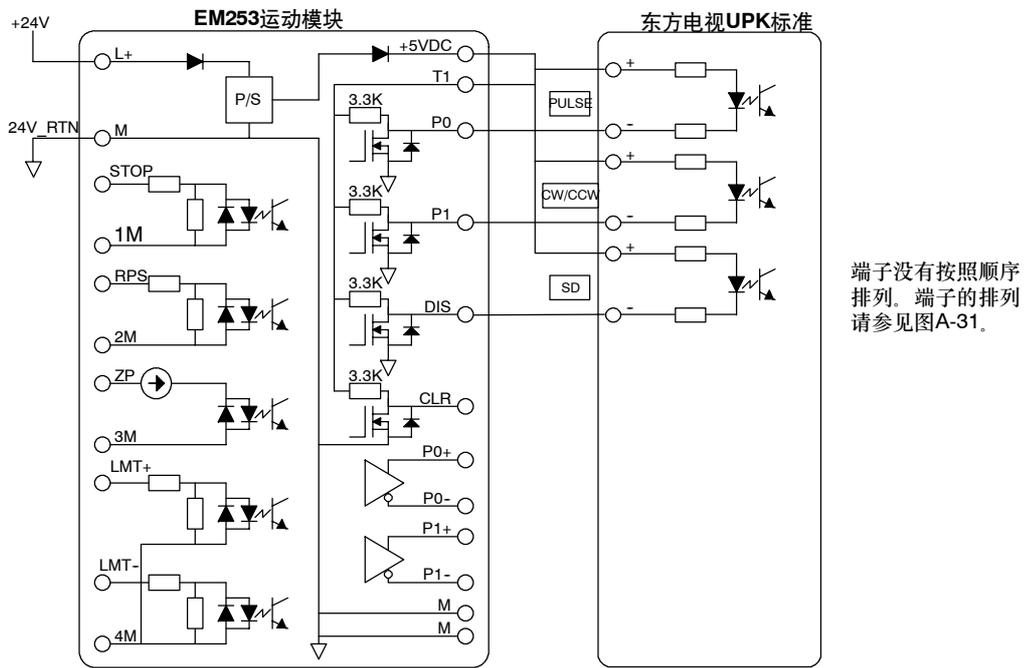


端子没有按照顺序排列。端子的排列请参见图A-31。

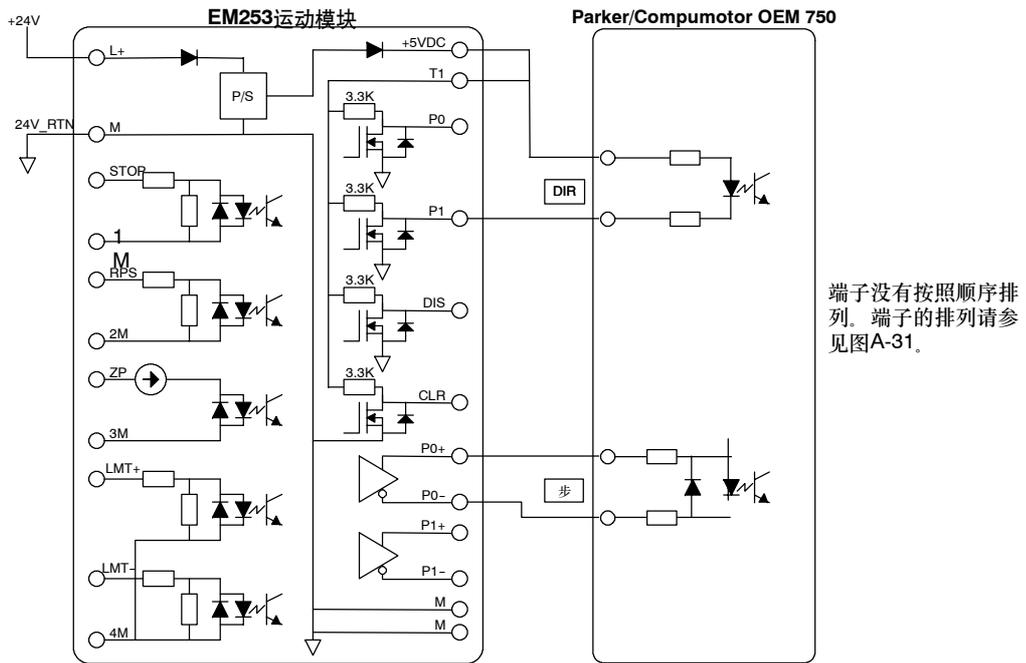
图A-33 连接EM253位控模块和FM步进驱动



图A-34 连接EM253位控模块到工业设备。（下一步）



图A-35 连接EM253位控模块和一个东方电机UPK标准



端子没有按照顺序排列。端子的排列请参见图A-31。

图A-36 连接EM253位控模块和Parker/Compumotor OEM 750

(CP243-1) 以太网模块规范

表A-53 (CP243-1) 以太网模块订货号

订货号	扩展模块	输入	输出	可拆卸连接
6GK7243-1EX00-OXE0	(CP243-1) 以太网模块	-	8 ¹	否

¹ 8位Q输出用作以太网功能的逻辑控制，并不直接控制任何外部信号。

表A-54 (CP243-1) 以太网模块常规规范

订货号	模块名称及描述	尺寸 (mm) (W x H x D)	重量	功耗	+5 VDC	VDC要求 +24 VDC
6GK7 243-1EX00-OXE0	(CP 243-1) 以太网模块	71.2 x 80 x 62	150g左右	1.75 W	55 mA	60mA

表A-55 (CP 243-1) 以太网模块规范

常规	6GK7 243-1EX00-OXE0
传输速率	10 Mbits/s 和100 Mbits/s
闪存大小	1M字节
SDRAM存储器大小	8M字节
连接到工业以太网的接口 (10/100Mbit/s)	8针RJ45插座
输入电压	20.4-28.8 VDC
最大连接数	每个 (CP243-1) 以太网模块最多连接8个S7 (XPUT/XGET和READ/WRITE)，另外，还可以建立1个到STEP7-Micro/WIN的连接 ² 。
启动时间或复位后的重新启动时间	大概10秒
用户数据质量	作为客户端: 对于XPUT/XGET, 最多212个字节 作为服务器: 对于XGET或READ, 最多222个字节 对于XPUT或WRITE, 最多212个字节

²每个S7-200 CPU只能连接一个 (CP243-1) 以太网模块。

(CP243-1) 以太网模块是用于将S7-200系统连接到工业以太网 (IE) 的通讯处理器。通过以太网，我们可以使用STEP 7 Micro/WIN来对S7-200进行远程组态、编程和诊断。S7-200可以通过以太网和其它S7-200、S7-300和S7-400控制器进行通讯，还可以与OPC服务器进行通讯。

工业以太网是为工业应用设计的。它可以与无噪声工业双绞线 (ITP) 技术，或者工业标准双绞线 (TP) 技术结合起来。工业以太网的用途非常之广，可用来实现多种特殊的应用，比如交换、高速冗余、快速连接和冗余网络。通过使用 (CP243-1) 以太网模块，S7-200PLC可以和现存的范围很广的各种产品相兼容。

支持智能模块的CPU

(CP243-1) 以太网模块是一种连接在S7-200上的智能扩展模块, 适用的CPU如表A-46所示。

表A-56 (CP243-1) 以太网模块和S7-200 CPU的兼容性

CPU	描述
CPU222版本1.10或更高	CPU222 DC/DC/DC和CPU222 AC/DC/继电器
CPU224版本1.10或更高	CPU224 DC/DC/DC和CPU224 AC/DC/继电器
CPU224XP 版本2.00或更高	CPU224XP DC/DC/DC和CPU224XP AC/DC/继电器
CPU226版本1.00或更高	CPU226 DC/DC/DC和CPU226 AC/DC/继电器

(CP243-1) 以太网模块提供了一个预设的、全球范围内唯一的MAC地址, 此地址不能被改变。

功能

(CP243-1) 以太网模块独立地处理在工业以太网上传输的数据。

- 通讯是基于TCP/IP协议的
- 可以作为通讯的客户端或服务, 从而使S7-200 CPU可以通过以太网和其它S7控制系统或PC之间进行通讯。最多可以建立八个连接。
- 集成S7-OPC服务器之后, 可以实现PC应用
- (CP243-1) 以太网模块可以使得S7-200编程软件, STEP 7-Micro/WIN通过以太网直接访问S7-200



Ethernet

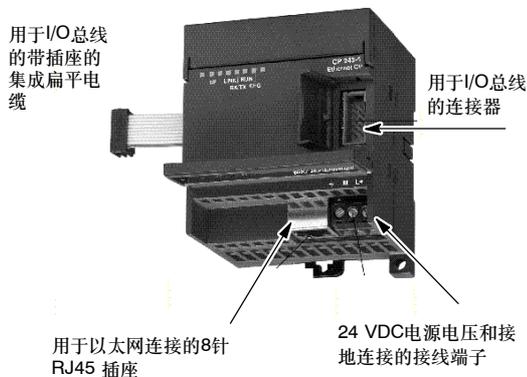
组态

可以使用STEP 7-Micro/WIN以太网向导来组态(CP243-1) 以太网模块, 以将S7-200PLC连接到以太网网络。以太网向导可以帮助您定义(CP243-1) 以太网模块的参数, 并把组态指令放到您的项目指令文件夹中。要启动以太网向导, 选择**Tools > Ethernet Wizard**菜单命令。向导需要下列信息: IP地址、子网掩码、网关地址和通讯连接类型。

连接

(CP243-1) 以太网模块设有一些连接点。连接点在前盖的盖子下面。包括:

- 用于连接24 VDC电源和接地的接线端子排
- 用于以太网连接的8针RJ45 插座
- I/O总线的插入连接器
- 用于I/O总线的带插座的集成扁平电缆



图A-37 连接(CP 243-1) 以太网模块

其它信息

对于(CP 243-1) 以太网模块的更多信息, 请参考SIMATIC NET CP 243-1 工业以太网通讯处理器技术手册。

(CP 243-1 IT) 因特网模块规范

表A-57 (CP 243-1 IT) 因特网模块订货号

订货号	扩展模块	输入	输出	可拆卸连接
6GK7 243-1GX00-OXE0	(CP 243-1 IT) 因特网模块	-	8 ¹	否

¹ 八个Q 输出用于IT功能的逻辑控制, 而不直接控制任何外部信号。

表A-58 (CP 243-1 IT) 因特网模块常规规范

订货号	模块名称及描述	尺寸 (mm) (W x H x D)	重量	功耗	+5 VDC	VDC要求 +24 VDC
6GK7 243-1GX00-OXE0	(CP 243-1 IT) 因特网模块	71.2 x 80 x 62	150g左右	1.75 W	55 mA	60mA

表A-59 (CP 243-1 IT) 因特网模块规范

常规	6GK7 243-1GX00-OXE0
传输速度	10 Mbit/s 和100 Mbits/s
闪存大小	8 M字节作为ROM用于 (CP 243-1 IT) 因特网模块的硬件, 8 M字节作为RAM用于文件系统
SDRAM存储器大小	16M字节
用于文件系统的闪存的使用寿命	1百万次写或删除操作
连接到工业以太网的接口 (10/100Mbit/s)	8- 针RJ45插座
输入电压	20.4-28.8VDC
最大连接数	每个 (CP 243-1 IT) 因特网模块最多8个S7连接 (XPUT/XGET和READ/WRITE) 加1个到STEP7-Micro/WIN的连接 ¹ 。
IT连接的最大数量	1个用于FTP服务器 1个用于FTP客户端 1个用于电子邮件客户端 4个用于HTTP连接
启动时间或复位后的重新启动时间	大概10秒
用户数据质量	客户端: 最多212个字节用于XPUT/XGET 服务器: 最多222个用于XGET或READ 最多212个字节用于XPUT或WRITE
电子邮件大小, 最大	1024个字符
文件系统: 包括文件大小和驱动名称的路径长度 文件名长度 目录嵌套深度	最大254个字符 最大99个字符 最大49个字符
可用的服务器端口: HTTP FTP命令通道 用于FTP服务器的FTP数据通道 S7连接建立 S7服务器	80 21 3100至3199 102 3000至3008

¹每个S7-200 CPU只能连接一个 (CP243-1IT) 因特网模块。

(CP 243-1 IT) 因特网模块是用于连接S7-200系统到工业以太网 (IE) 的通讯处理器。可以使用STEP 7 Micro/WIN, 通过以太网对S7-200进行远程组态、编程和诊断。S7-200可以通过以太网和其他S7-200、S7-300和S7-400控制器进行通讯。它还可以和OPC服务器进行通讯。

(CP243-1 IT) 因特网模块的IT功能构成了监视和如果需要的话, 通过联网的PC用WEB浏览器操作自动系统的基础。诊断信息可以通过电子邮件从系统中发送出去。通过使用IT功能, 可以很容易地和其他计算机和控制器系统交换所有文件。

工业以太网是用于SIMATIC NET开放通讯系统地过程控制级和单元级的网络。物理上, 工业以太网是一个基于屏蔽的、同轴双绞线的电气网络和光纤光学导线的光网络。工业以太网是由国际标准IEEE 802.3定义的。

支持智能模块的CPU

(CP 243-1 IT) 因特网模块是设计用于和表A-46中列出的S7-200 CPU一起工作的智能扩展模块。

表A-60 (CP 243-1 IT) 因特网模块和S7-200 CPU的兼容性

CPU	描述
CPU222版本1.10或更高	CPU222 DC/DC/DC和CPU222 AC/DC/继电器
CPU224版本1.10或更高	CPU224 DC/DC/DC和CPU224 AC/DC/继电器
CPU224XP版本2.00或更高	CPU224XP DC/DC/DC和CPU224XP AC/DC/继电器
CPU226版本1.00或更高	CPU226 DC/DC/DC和CPU226 AC/DC/继电器

(CP 243-1 IT) 因特网模块有下列特性:

- (CPU 243-1 IT) 因特网模块全面兼容 (CP 243-1) 以太网模块。为 (CP 243-1) 以太网模块写的用户程序, 可以在 (CP 243-1 IT) 因特网模块上运行。

(CP 243-1 IT) 因特网模块提供了一个预设的、全球范围内唯一的MAC地址, 此地址不能被改变。



提示

每个S7-200 CPU只能连接一个 (CP 243-1 IT) 因特网模块。如果连接了多于一个的 (CP 243-1 IT) 因特网模块, 则S7-200 CPU不能正确运行。

功能

(CP 243-1 IT) 因特网模块提供下列功能:

- S7通讯是基于TCP/IP的
- IT通讯
- 组态
- 看门狗定时器
- 预设的MAC地址 (48-位值) 可以被寻址的能力



Internet

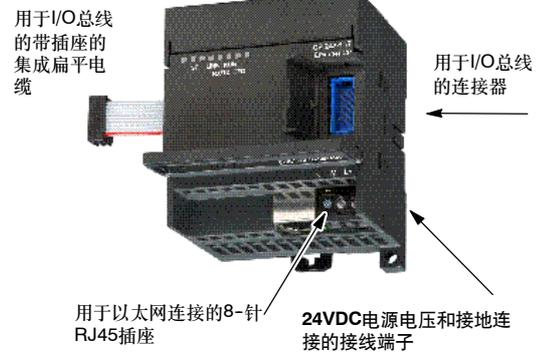
组态

可以使用STEP 7-Micro/WIN因特网向导来组态（CP 243-1 IT）因特网模块，以将S7-200 PLC连接到以太网/因特网网络上。（CP 243-1 IT）因特网模块有附加的网页服务器功能，可以使用因特网向导来组态此功能。启动因特网向导，选择**Tools > Internet Wizard**菜单命令。

连接

（CP 243-1 IT）因特网模块有下列一些连接。连接在前盖的盖子下面。

- 24 VDC电源电压和接地连接的接线端子
- 用于以太网连接的8-针RJ45插座
- I/O总线的插入连接器
- 用于I/O总线的带插座的集成扁平电缆



图A-38 连接（CP 243-1 IT）因特网模块

其它信息

对于（CP 243-1 IT）因特网模块的更多信息，请参考*SIMATIC NET CP 243-1 IT工业以太网通讯处理器和信息科学技术手册*。

(CP 243-2) AS-i接口模块规范

表A-61 (CP 243-2) AS-i接口模块定货号

订货号	扩展模块	输入	输出	可拆卸连接
6GK7 243-2AX01-0XA0	(CP 243-2) ASI-接口模块	8数字量8模拟量	8数字量8模拟量	是

表A-62 (CP 243-2) AS接口模块通用规范

订货号	模块名称及描述	尺寸 (mm) (W x H x D)	重量	功耗	+5 VDC	VDC要求 来自AS接口
6GK7 243-2AX01-0XA0	(CP 243-2) ASI-接口模块	71 x 80 x 62	250g左右	3.7W	220mA	100mA

表A-63 (CP 243-2) AS-i接口模块规范

常规	6GK7 243-2AX01-0XA0
循环时间	5ms, 31个从站 10ms, 62个AS-i从站使用扩展地址模式
组态	使用前面板上的按钮进行组态, 或使用完全组态命令 (参考CP 243-2 AS-i接口主站手册中关于AS-i命令的说明)
支持的AS-i主站协议	M1e
AS-i电缆附件	通过S7-200端子块。端子1到3或从端子2到4的允许电流负载为3A。
地址范围	一个带有8入/8出的数字量模块。 一个带有8通道模拟输入和8通道模拟输出的模拟量模块

特性

在一个S7-200上同时可操作多达两个AS-i接口模块, 增加了可使用的数字和模拟输入/输出 (每个CP AS-i接口最多124数字输入/124数字输出)。由于使用按钮组态从而节省了设置时间。LED通过显示CP和所连接的从站的状态以及监控AS-i接口主电压, 减少了停车时间。

AS接口模块有下列特性:

- 支持模拟量模块
- 支持所有的主站功能并能够最多连接62个AS-i接口从站
- 前面板上的LED显示所连从站的运行状态和就绪状态
- 前面板上的LED显示错误 (包括AS-i接口电压错误, 组态错误)
- 两个端子可直接连接AS-i接口电缆
- 两个按钮显示从站的状态信息, 切换运行模式, 并可将现有的组态作为设置组态



AS-i

可以使用STEP 7-Micro/WIN AS-i向导组态 (CP 243-2) AS接口模块。AS接口向导可以帮助您在您的组态中使用来自AS-i网络的数据。要启动AS-i向导, 选择**Tools > AS-i Wizard**菜单命令。

运行

在S7-200的内部映像区中，CP243-2占用一个数字输入字节（状态字节），一个数字输出字节（控制字节），以及8个模拟量输入和8个模拟量输出字。因此，CP243-2占用两个逻辑模块位置。通过用户程序，状态和控制字节可用于设置CP243-2的运行模式。AS-i接口有两种模式：存储AS-i接口从站的I/O数据和诊断值，或者使能S7-200的模拟量地址区中的主站调用（例如，改变从站的地址）。

所有连接的AS-i接口从站都可通过按压一个按钮进行配置。CP的进一步配置是不必要的。

小心

当使用CP243-2模块时，必须禁止PLC中的模拟量滤波。

如果不禁止PLC中的模拟量滤波，则会破坏数字点数据，而且出错条件也不会模拟量控制字中位的状态来返回。

应确实保证，PLC中的模拟量滤波已被禁止。

功能

CP243-2是符合MI主站行规的AS-i接口主站，这就是说，它支持所有规定的功能。因而它可以借助于双地址赋值（A-B），使AS-i接口上能运行最多31个数字从站。CP243-2可以设置为两种不同的模式：

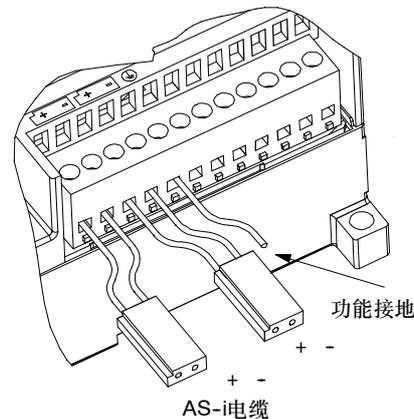
- 标准模式：存取AS-i接口从站的I/O数据
- 扩展模式：主站调用（例如写参数）或诊断值请求

连接

AS-i接口模块也可以进行下列连接：

- 两个连至AS-i接口模块电缆的连接（内置）
- 一个功能地的连接

如图A-39示，端子位于前面板盖下。



图A-39 CP243-2通讯处理器接线

小心

AS-i接口模块触点的负载能力大是3A。如果AS-i接口模块电缆上的电流超过该值，该AS-i接口则不能连接到AS-i电缆上，而应由一个分开的电缆来连接（这种情况下，只使用AS-i接口模块的一对端子）。该AS-i接口必须通过接地端子接到接地导体上。



提示

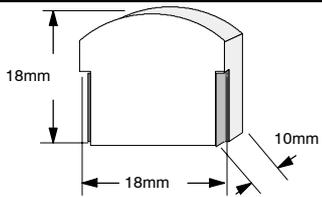
AS-i接口模块有功能地的连接。该接口应以尽可能少的电阻连至PE导体。

其它信息

关于CP243-2 AS-i接口主站更多的信息，请参考SIMATIC NET CP 243-2 AS-i接口主站手册。

可选卡件

卡件	描述	订货号
存储卡	存储卡, 32K (用户程序)	6ES7 291-8GE20-0XA0
存储卡	存储卡, 64K (用户程序, 配方和数据记录)	6ES7 291-8GF23-0XA0
存储卡	存储卡, 256K (用户程序, 配方和数据记录)	6ES7 291-8GH23-0XA0
带电池的实时时钟卡	时钟卡精度: 2分钟/月, 25°C, 7分钟/月, 0°C至55°C	6ES7 297-1AA23-0XA0
电池卡	电池卡 (数据保持时间): 200天, 典型	6ES7 291-8BA20-0XA0

常规特性		尺寸
电池 尺寸 类型	3V, 30mA小时Renata CR 1025 9.9mmx2.5mm 锂电池<0.6g	

存储卡

在不同型号的CPU之间使用存储卡有一些限制。由特定型号CPU编程的存储卡只能被相同或更高类型号的CPU读取, 见表A-64:

表A-64 存储卡读取限制

在...中编程的存储卡	可以被...读取
CPU221	CPU221、CPU222、CPU224、CPU224XP和CPU226
CPU222	CPU222、CPU224、CPU224XP和CPU226
CPU224	CPU224、CPU224XP和CPU226
CPU224XP	CPU224XP和CPU226
CPU226	CPU226

64K和256K存储卡只能用于某些新CPU, 这些CPU的订货号如表所示: 6ES721x-xx23-0XB0。每个“x”表示此位是不用考虑的位。

建议不要在“23”型CPU中使用32K存储卡(6ES7 291-8GE20-0XA0), 因为32K存储卡不支持新的CPU功能。当您使用某“23”型CPU在32K存储卡上存储程序时, 编程的存储卡将把该CPU作为早期型号的CPU来兼容。因此新CPU的所有高级功能将不能存储在32K存储卡上。

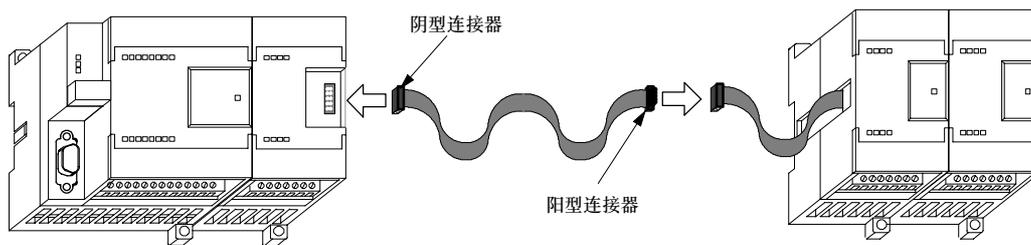
旧型CPU(“20”、“21”或“22”型)编辑的用户程序可以存储在32K存储卡上。这些存储卡可以被新CPU读取, 这些新CPU的型号如表A-64所示。

实时时钟卡

实时时钟卡(6ES7 297-1AA23-0XA0)是设计用于只和“23”CPU一起工作的。实时时钟卡(6ES7 297-1AA20-0XA0)的早期版本和“23”型CPU在物理或电气上不兼容。

扩展电缆

常规特性 (6ES7 290-6AA20-0XA0)	
电缆长度	0.8m
重量	25g
接口类型	10针扁平电缆



图A-40 I/O扩展电缆的典型安装



提示

在CPU/扩展模块的连接中只能使用一条扩展电缆。

RS-232/PPI多主站电缆和USB/PPI多主站电缆

表A-65 RS-232/PPI多主站电缆和USB/PPI多主站电缆规范

描述 订货号	S7-200 RS-232/PPI多主站电缆 6ES7 901-3CB30-0XA0	S7-200 USB/PPI 多主站电缆 6ES7-901-3DB30-0XA0
总体特性		
电源电压	14.4-28.8VDC	14.4-28.8VDC
24V电源的电流	最大60mA RMS	最大50mA RMS
方向改变延时: 接收的RS232停止位边沿到发送的RS485停止位边沿	-	-
隔离	RS-485到RS-232: 500 VDC	RS-485到USB: 500 VDC
RS-485一侧的电气特性		
共模电压范围	-7V到+12V,1秒,3V RMS	-7V到+12V,1秒,3V RMS
接收器输入阻抗	最小5.4 K Ω , 包括终端	最小5.4 K Ω , 包括终端
终端/偏置	B上10K Ω 对+5 V, PROFIBUS 针3 A上10K Ω 对GND, PROFIBUS 针 8	B上10K Ω 对+5 V, PROFIBUS 针3 A上10K Ω 对GND, PROFIBUS 针 8
接收器阈值/灵敏度	+/-0.2V, 60mV 典型滞后	+/-0.2V, 60mV 典型滞后
发送器差分输出电压	最小2 V, 在 $R_L=100 \Omega$ 处, 最小1.5 V, 在 $R_L=54 \Omega$ 处	最小2 V, 在 $R_L=100 \Omega$ 处, 最小1.5 V, 在 $R_L=54 \Omega$ 处
RS232一侧的电气特性		
接收器输入阻抗	最小3K Ω	-
接收器阈值/灵敏度	最低0.8 V, 最高2.4 V 0.5 V 典型滞后	-
发送器输出电压	最小+/-5V, 在 $R_L=3K\Omega$ 处	-
USB一侧的电气特性		
最大速率 (12MB/s), 人机界面设备 (HID)		
5V的电源电流	-	最大50mA
掉电电流	-	最大400uA

特性

为了取得最佳性能, S7-200 RS-232/PPI多主站电缆作了工厂设置, 该设置可在STEP 7-Micro/WIN 3.2 Service Pack 4 (或更高) 编程包中找到。RS-232/PPI多主站电缆的出厂设置和PC/PPI电缆的不同。若需根据具体的应用来配置电缆, 请参考图1。

S7-200 RS-232/PPI多主站电缆在经过配置之后, 可以和PC/PPI电缆一样操作, 另外, 在将设定开关5设置为PPI/Freeport, 并选择了需要的波特率之后, 该电缆可以和任意版本的STEP 7-Micro/WIN编程包相兼容。

若要使用USB电缆, 需要STEP 7-Micro/WIN 3.2 Service Pack 4 (或更高版本) 编程包。



提示

对于PC/PPI电缆的信息, 请参考S7-200可编程控制器手册的版本3, 订货号 (6ES7 298-8FA24-8FH0)。

S7-200 RS-232/PPI多主站电缆

表A-66 S7-200 RS-232/PPI多主站电缆 - RS-485到RS-232本地模式连接器的引脚分配

RS-485		RS-232本地连接器引脚分配	
针号	信号描述	针号	信号描述
1	未连接	1	数据载波检测 (DCD) (不用)
2	24V返回 (RS-485逻辑地)	2	接收数据 (RD) (从PC/PPI电缆输出)
3	信号B (RxD/TxD+)	3	发送数据 (TD) (输入到PC/PPI)
4	RTS (TTL电平)	4	数据终端就绪 (DTR) ¹
5	未连接	5	地 (RS-232逻辑地)
6	未连接	6	数据设置就绪 (DSR) ¹
7	24V电源	7	申请发送 (RTS) (不用)
8	信号A (RxD/TxD-)	8	清除发送 (CTS) (不用)
9	协议选择	9	振铃指示灯 (RI) (不用)

¹针4和6在内部连接在一起。

表A-67 S7-200 RS-232/PPI多主站电缆 - RS-485到RS-232远端模式连接器的引脚分配

RS-485		RS-232远端连接器引脚分配 ¹	
针号	RS-485连接器引针	针号	RS-485连接器引针
1	未连接	1	数据载波检测 (DCD) (不用)
2	24V返回 (RS-485逻辑地)	2	接收数据 (RD) (从PC/PPI电缆输出)
3	信号B (RxD/TxD+)	3	发送数据 (TD) (输入到PC/PPI)
4	RTS (TTL电平)	4	数据终端就绪 (DTR) ²
5	未连接	5	地 (RS-232逻辑地)
6	未连接	6	数据设置就绪 (DSR) ²
7	24V电源	7	请求发送 (RTS) (来自PC/PPI电缆的输出)
8	信号A (RxD/TxD-)	8	清除发送 (CTS) (不用)
9	协议选择	9	振铃指示灯 (RI) (不用)

¹Modem需要一个阴-阳型9到25针的转换器

²针4和6在内部连接在一起。

通过STEP 7-Micro/WIN, 用S7-200 RS-232/PPI多主站电缆替代PC/PPI电缆, 或用于自由口操作。

用于直接连接到您的个人计算机:

- 设置PPI/Freeport模式 (开关5=0)
- 设置波特率 (开关1、2和3)
- 设置本地模式 (开关6=0)。本地设置和将PC/PPI电缆设置到DCE一样。
- 设置11位 (开关7=0)

用于连接到Modem:

- 设置PPI/Freeport模式 (开关5=0)
- 设置波特率 (开关1、2和3)
- 设置远端模式 (开关6=1)。远端设置和将PC/PPI电缆设置到DTE一样。
- 设置10位或11位 (开关7), 和您的Modem的每个字符设置的位数相匹配。

通过STEP 7-Micro/WIN 3.2 Service Pack 4 (或更高版本) 来使用S7-200 RS-232/PPI多主站电缆

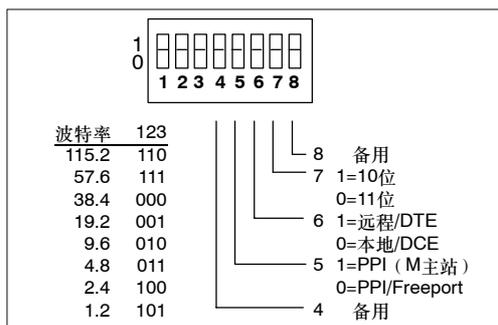
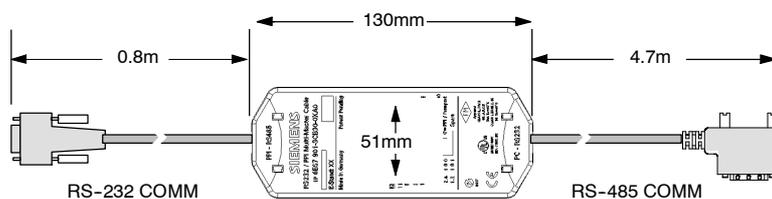
直接连接到个人计算机:

- 设置PPI模式 (开关5=1)
- 设置本地模式 (开关6=0)

用于连接到Modem:

- 设置PPI模式 (开关5=1)
- 设置远端模式 (开关6=1)

图A-41列出了S7-200 RS-232/PPI多主站电缆的尺寸、标签和LED。



LED	颜色	描述
Tx	绿色	RS-232发送指示灯
Rx	绿色	RS-232接收指示灯
PPI	绿色	RS-485发送指示灯

图A-41 S7-200 RS-232/PPI多主站电缆的尺寸、标签和LED

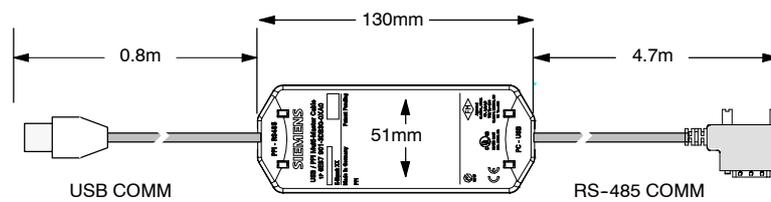
S7-200 USB/PPI多主站电缆

要使用USB电缆，必须安装STEP 7-Micro/WIN 3.2 Service Pack 4（或更高版本），且最好配合S7-200 CPU22x或更新型的CPU来使用。USB电缆不支持自由口通讯，也不能用来将TP编辑器下载到TP070。

表A-68 S7-200 USB/PPI多主站电缆 - RS-485到USB系列“A”连接器的引脚分配

RS-485		USB连接器引脚分配	
针号	RS-485连接器引脚	针号	RS-485连接器引脚
1	未连接	1	USB - DataP
2	24V返回（RS-485逻辑地）	2	USB - DataM
3	信号B（Rx/D/TxD+）	3	USB 5V
4	RTS（TTL电平）	4	USB 逻辑地
5	未连接		
6	未连接		
7	24V电源		
8	信号A（Rx/D/TxD-）		
9	协议选择（低=10位）		

图A-42列出了S7-200 USB/PPI多主站电缆的尺寸和LED。

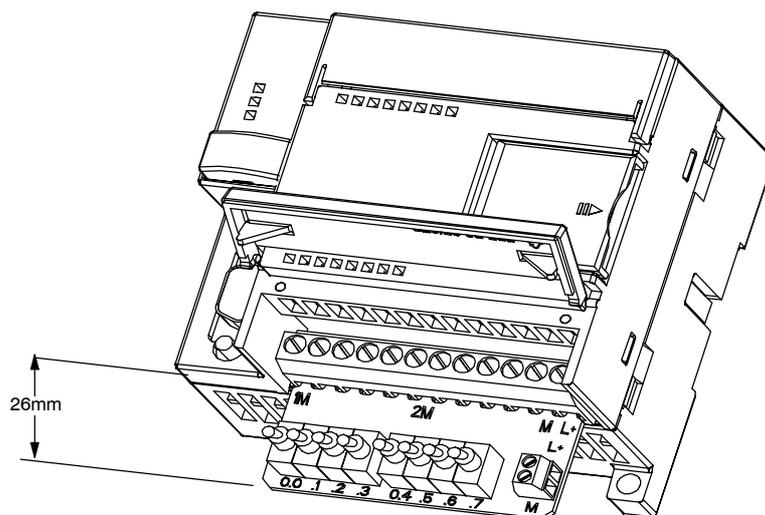


LED	颜色	描述
Tx	绿色	USB发送指示灯
Rx	绿色	USB接收指示灯
PPI	绿色	RS-485发送指示灯

图A-42 S7-200 USB/PPI多主站电缆的尺寸和LED

输入仿真器

订货号	8输入仿真器 6ES7 274-1XF00-0XA0	14输入仿真器 6ES7 274-1XH00-0XA0	24输入仿真器 6ES7 274-1XK00-0XA
尺寸 (L x W x D)	61 x 33.5 x 22 mm	91.5 x 35.5 x 22 mm	148.3 x 35.5 x 22 mm
重量	0.02 Kg	0.03 Kg	0.04 Kg
点数	8	14	24



图A-43 输入仿真器的安装



警告

这些输入仿真器未被证明能够用于Class 1 DIV 2或Class 1 Zone 2危险环境。这些开关可能会引起火花。
这些输入仿真器不能用于Class 1 DIV 2或Class 1 Zone 2危险环境。

电源设计



S7-200本机单元有一个内部电源，它为本机单元、扩展模块以及24VDC用户供电。利用下面提供的信息作为指导，计算S7-200 CPU能够为您的组态提供多大的功率（或电流）。

电源要求

每一个S7-200 CPU模块提供5VDC和24VDC电源：

- 每一个CPU模块都有一个24VDC传感器电源，它为本机输入点和扩展模块继电器线圈提供24VDC。如果电源要求超出了CPU模块24VDC电源的定额，您可以增加一个外部24VDC电源来供给扩展模块的24VDC。
- 当有扩展模块连接时CPU模块也为其提供5V电源。如果扩展模块的5V电源需求超出了CPU模块的电源定额，您必须卸下扩展模块，直到需求在电源预定值之内才行。

附录A数据表提供了有关CPU模块电源预算的信息，以及扩展模块所需电源要求的信息。



提示

若电源要求超出CPU模块预算，您将不能连接CPU允许的最大数目的模块。



警告

将S7-200 DC传感器电源与外部24VDC电源采用并联连接时，将会导致两个电源的竞争而影响它们各自的输出。

这种竞争的结果要缩短设备的寿命，或者使得一个电源或两者同时失效，并且使PLC系统产生不正确的操作。

S7-200 DC传感器电源和外部电源应该在不同的点上提供电源，两者之间只能有一个连接点。

计算举例

表B-1所示的是一个S7-200MicroPLC电源需求量计算的例子，它包括以下模块：

- CPU224 AC/DC/继电器
- 3个EM223数字输入8/继电器输出8
- 1个EM221数字输入8

该配置共有46个输入和34个输出。



提示

CPU已经为内部继电器线圈的驱动分配了所需要的电源。您在计算电源定额时无需包括内部继电器线圈的电源要求。

在本例中，CPU模块为扩展模块提供了足够的5V电源，但是它没有给所有的输入和输出线圈提供足够的24VDC电源。本例中I/O要求400mA的24VDC电源，但是CPU只能提供280mA。本配置需要提供额外的120mA电流。

表B-1 针对一个配置实例的电源计算

CPU电源预算	5VDC	24VDC
CPU224 AC/DC/继电器	660mA	280mA

减

系统要求	5VDC	24VDC
CPU224,14输入		14 * 4mA= 56mA
3 EM223, 5V电源需求	3*80mA= 240mA	
1 EM221, 5V电源需求	1*30mA= 30mA	
3 EM223, 每个8输入		3*8*4mA= 96mA
3 EM223, 每个8继电器线圈		3*8*9mA= 216mA
1 EM221, 每个8输入		8*4mA= 32mA
总需求	270mA	400mA

等于

电流平衡	5VDC	24VDC
总电流平衡	剩390mA	缺120mA

计算您的电能需求

利用下表计算CPU可以为您的配置提供多少电源（或电流）。请参考附录A给出的CPU电源预算和扩展模块的电源要求。

电源预算	5VDC	24VDC

减

系统要求	5VDC	24VDC
总需求		

等于

电流平衡	5VDC	24VDC
总电流平衡		

错误代码



有关错误代码的信息有助于查找S7-200 CPU的问题。

本章概述

致命错误代码和信息	462
运行程序错误	463
编译规则错误	464

致命错误代码和信息

致命错误会导致CPU停止执行用户程序。依据错误的严重性，一个致命错误会导致CPU无法执行某个或所有功能。处理致命错误的目标是，使CPU进入安全状态，可以对当前存在的错误状况进行询问并响应。

当一个致命错误发生时，CPU执行以下任务：

- 进入STOP（停止）方式
- 点亮SF/DIAG（红）LED指示灯和停止LED指示灯
- 断开输出

这种状态将会持续到错误清除之后。在主菜单中使用菜单命令**PLC > Information**可查看错误代码。表C-1列出了从S7-200上可读到的致命错误代码及其描述。

表C-1 从CPU读出的致命错误代码及其描述

错误代码	描述
0000	无致命错误
0001	用户程序校验和错误
0002	编译后的梯形图程序校验和错误
0003	扫描看门狗超时错误
0004	永久存储器失效
0005	永久存储器上用户程序校验和错误
0006	永久存储器上配置参数（SDB0）校验和错误
0007	永久存储器上强制数据校验和错误
0008	永久存储器上缺省输出表值校验和错误
0009	永久存储器上用户数据DB1校验和错误
000A	存储器卡失灵
000B	存储器卡上用户程序校验和错误
000C	存储器卡配置参数（SDB0）校验和错误
000D	存储器卡强制数据校验和错误
000E	存储器卡缺省输出表值校验和错误
000F	存储器卡用户数据DB1校验和错误
0010	内部软件错误
0011 ¹	比较接点间接寻址错误
0012 ¹	比较接点非法值错误
0013	程序不能被该S7-200理解
0014 ¹	比较接点范围错误

¹ 比较接点错误是唯一的一种即能产生致命错误又能产生非致命错误的错误。产生非致命错误是因为存储错误的程序地址。

运行程序错误

在程序的正常运行中，可能会产生非致命错误（如寻址错误）。在这种情况下，CPU产生一个非致命运行时刻错误代码。表C-2列出了这些非致命错误代码及其描述。

表C-2 运行程序错误

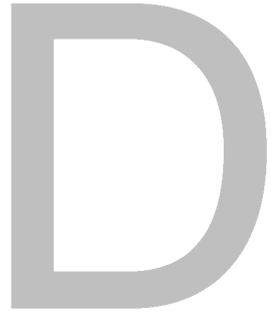
错误代码	描述
0000	无错误
0001	执行HDEF之前，HSC未允许
0002	输入中断分配冲突，已分配给HSC
0003	到HSC的输入分配冲突，已分配给输入中断
0004	试图执行在中断子程序中不允许的指令
0005	第一个HSC/PLS未执行完之前，又企图执行同编号的第二个HSC/PLS（中断程序中的HSC同主程序中的HSC/PLS冲突。）
0006	间接寻址错误
0007	TODW 写实时时钟）或TODR（读实时时钟）数据错误
0008	用户子程序嵌套层数超过规定
0009	在程序执行XMT或RCV时，通讯口0又执行另一条XMT/RCV指令
000A	在同一HSC执行时，又企图用HDEF指令再定义该HSC
000B	在通讯口1上同时执行XMT/RCV指令
000C	时钟存储卡不存在
000D	重新定义已经使用的脉冲输出
000E	PTO个数设为0
000F	比较触点指令中的非法数字值
0010	在当前PTO操作模式中，命令未允许
0011	非法PTO命令代码
0012	非法PTO包络表
0013	非法PID回路参数表
0091	范围错误（带地址信息）：检查操作数范围
0092	某条指令的计数域错误（带计数信息）：确认最大计数范围
0094	范围错误（带地址信息）：写无效存储器
009A	用户中断程序试图转换成自由口模式
009B	非法指针（字符串操作中起始位置值指定为0）
009F	无存储卡或存储卡无响应

编译规则错误

当你下装一个程序时，CPU将编译该程序。如果CPU发现程序违反编译规则（如非法指令），那么CPU就会停止下装程序，并生成一个非致命编译规则错误代码。表C-3列出了违反编译规则所生成的这些错误代码及其描述。

表C-3 编译规则错误

错误代码	编译错误（非致命）
0080	程序太大无法编译：你必须缩短程序
0081	堆栈溢出：把一个程序段分成多个
0082	非法指令：检查指令助记符
0083	无MEND或主程序中有不允许的指令：加条MEND或删除不正确的指令
0084	保留
0085	无FOR指令：加条FOR指令或删除NEXT指令
0086	无NEXT：加条NEXT指令，或删除FOR指令
0087	无标号（LBL、INT、SBR）：加上合适标号
0088	无RET或子程序中有不允许的指令：加条RET或删除不正确指令
0089	无RETI或中断程序中有不允许的指令：加条RETI或删除不正确指令
008A	保留
008B	从/向一个SCR段的非法跳转
008C	标号重复（LBL、INT、SBR）：重新命名标号
008D	非法标号（LBL、INT、SBR）：确保标号数在允许范围内
0090	非法参数：确认指令所允许的参数
0091	范围错误（带地址信息）：检查操作数范围
0092	指令计数域错误（带计数信息）：确认最大计数范围
0093	FOR/NEXT嵌套层数超出范围
0095	无LSCR指令（装载SCR）
0096	无SCRE指令（SCR结束）或SCRE前面有不允许的指令
0097	用户程序包含非数字编码的和数字编码的EV/ED指令
0098	在运行模式进行非法编辑（试图编辑非数字编码的EV/ED指令）
0099	隐含程序段太多（HIDE指令）
009B	非法指针（字符串操作中起始位置值指定为0）
009C	超出最大指令长度
009D	SDB0中检测到非法参数
009E	PCALL字符串太多
009F-00FF	保留



特殊存储器（SM）标志位

特殊存储器标志位提供大量的状态和控制功能，并能起到在CPU和用户程序之间交换信息的作用。特殊存储器标志位能以位、字节、字或双字使用。

本章概述

SMB0: 状态位	466
SMB1: 状态位	466
SMB2: 自由口接收字符	467
SMB3: 自由口奇偶校验错误	467
SMB4: 队列溢出	467
SMB5: I/O状态	468
SMB6: CPU识别（ID）寄存器	468
SMB7: 保留	468
SMB8到SMB21: I/O模块识别和错误寄存器	469
SMW22到SMW26: 扫描时间	470
SMB28和SMB29: 模拟电位器	470
SMB30和SMB130: 自由口控制寄存器	470
SMB31和SMW32: 永久存储器（EEPROM）写控制	471
SMB34和SMB35: 定时中断时间间隔寄存器	471
SMB36到SMB65: HSC0、HSC1和HSC2寄存器	471
SMB66到SMB85: PTO/PWM寄存器	473
SMB86到SMB94和SMB186到SMB194: 接收信息控制	474
SMW98: 扩展I/O总线错误	475
SMB130: 自由口控制寄存器（见SMB30）	475
SMB131到SMB165: HSC3、HSC4和HSC5寄存器	475
SMB166到SMB185: PTO0、PTO1包络定义表	476
SMB186到SMB194: 接收信息控制（见SMB86-SMB94）	476
SMB200到SMB549: 智能模块状态	477

SMB0: 状态位

如表D-1所示，SMB0有8个状态位，在每个扫描周期的末尾，由S7-200更新这些位。

表D-1 特殊存储器字节SMB0 (SM0.0至SM0.7)

SM位	描述 (只读)
SM0.0	该位始终为1
SM0.1	该位在首次扫描时为1，用途之一是调用初始化子程序
SM0.2	若保持数据丢失，则该位在一个扫描周期中为1。该位可用作错误存储器位，或用来调用特殊启动顺序功能。
SM0.3	开机后进入RUN方式，该位将ON一个扫描周期，该位可用作在启动操作之前给设备提供一个预热时间
SM0.4	该位提供了一个时钟脉冲，30秒为1，30秒为0，周期为一分钟，它提供了一个简单易用的延时或1分钟的时钟脉冲
SM0.5	该位提供了一个时钟脉冲，0.5秒为1，0.5秒为0，周期为1秒钟。它提供了一个简单易用的延时或1秒钟的时钟脉冲
SM0.6	该位为扫描时钟，本次扫描时置1，下次扫描时置0。可用作扫描计数器的输入
SM0.7	该位指示CPU工作方式开关的位置（0为TERM位置，1为RUN位置）。当开关在RUN位置时，用该位可使自由端口通信方式有效，那么当切换至TERM位置时，同编程设备的正常通讯也会有效。

SMB1: 状态位

如表D-2所示，SMB1包含了各种潜在的错误提示。这些位可由指令在执行时进行置位或复位。

表D-2 特殊存储器字节SMB1 (SM1.0至SM1.7)

SM位	描述 (只读)
SM1.0	当执行某些指令，其结果为0时，将该位置1。
SM1.1	当执行某些指令，其结果溢出或查出非法数值时，将该位置1。
SM1.2	当执行数学运算，其结果为负数时，将该位置1。
SM1.3	试图除以零时，将该位置1。
SM1.4	当执行ATT (Add to Table) 指令时，试图超出表范围时，将该位置1。
SM1.5	当执行LIFO或FIFO指令，试图从空表中读数时，将该位置1。
SM1.6	当试图把一个非BCD数转换为二进制数时，将该位置1。
SM1.7	当ASCII码不能转换为有效的十六进制数时，将该位置1。

SMB2: 自由口接收字符

SMB2为自由端口接收字符缓冲区。如表D-3所示，在自由端口通信方式下，接收到的每个字符都放在这里，便于梯形图程序存取。



提示

SMB2和SMB3由0口和1口共用。当0口接收到字符并使得与该事件（中断事件8）相连的中断程序执行时，SMB2包含0口接收到的字符，而SMB3包含该字符的校验状态。当1口接收到字符并使得与该事件（中断事件25）相连的中断程序执行时，SMB2包含1口接收到的字符，而SMB3包含该字符的校验状态。

表D-3 特殊存储器字节SMB2

SM位	描述（只读）
SMB2	在自由口通讯方式下，该字符存储从口0或口1接收到的每一个字符

SMB3: 自由口奇偶校验错误

SMB3用于自由口方式，当接收到的字符发现有奇偶校验错误时，将SM3.0置1。如表D-4所示，当检测到校验错误时，SM3.0接通。根据该位来废弃错误消息。

表D-4 特殊存储器字节SMB3（SM3.0至SM3.7）

SM位	描述（只读）
SM3.0	口0或口1的奇偶校验错（0=无错；1=有错）
SM3.1至SM3.7	保留

SMB4: 队列溢出

如表D-5所示，SMB4包含中断队列溢出位，中断是否允许标志位及发送空闲位。队列溢出表明要么是中断发生的频率高于CPU，要么是中断已经被全局中断禁止指令所禁止。

表D-5 特殊存储器字节SMB4（SM4.0至SM4.7）

SM位	描述（只读）
SM4.0 ¹	当通信中断队列溢出时，将该位置1。
SM4.1 ¹	当输入中断队列溢出时，将该位置1。
SM4.2 ¹	当定时中断队列溢出时，将该位置1。
SM4.3	在运行时刻，发现编程问题时，将该位置1。
SM4.4	该位指示全局中断允许位，当允许中断时，将该位置1。
SM4.5	当（口0）发送空闲时，将该位置1。
SM4.6	当（口1）发送空闲时，将该位置1。
SM4.7	当发生强置时，将该位置1

¹ 只有在中断程序里，才使用状态位SM4.0、SM4.1和SM4.2。当队列为空时，将这些状态位复位（置0），并返回主程序。

SMB5: I/O状态

如表D-6所示，SMB5包含I/O系统里发现的错误状态位。这些位提供了所发现的I/O错误的概况。

表D-6 特殊存储器字节SMB5 (SM5.0至SM5.7)

SM位	描述 (只读)
SM5.0	当有I/O错误时，将该位置1
SM5.1	当I/O总线上连接了过多的数字量I/O点时，将该位置1。
SM5.2	当I/O总线上连接了过多的模拟量I/O点时，将该位置1。
SM5.3	当I/O总线上连接了过多的智能I/O模块时，将该位置1。
SM5.4至SM5.7	保留

SMB6: CPU识别 (ID) 寄存器

如表D-7所示，SMB6为CPU识别 (ID) 寄存器。SM6.4到SM6.7识别CPU的类型，SM6.0到SM6.3保留，以备将来使用。

表D-7 特殊存储器字节SMB6

SM位	描述 (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> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">x</td> <td style="width: 20px;">x</td> <td style="width: 20px;">x</td> <td style="width: 20px;">x</td> <td style="width: 20px;">r</td> <td style="width: 20px;">r</td> <td style="width: 20px;">r</td> <td style="width: 20px;">r</td> </tr> </table> <div style="text-align: center; margin-left: 10px;">LSB 0</div> </div> <p style="text-align: right; margin-right: 20px;">CPU识别 (ID) 寄存器</p>	x	x	x	x	r	r	r	r				
x	x	x	x	r	r	r	r						
SM6.0至SM6.3	保留												
SM6.4至SM6.7	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">xxxx=</td> <td style="width: 25%;">0000=</td> <td style="width: 60%;">CPU 222</td> </tr> <tr> <td></td> <td>0010=</td> <td>CPU 224</td> </tr> <tr> <td></td> <td>0110=</td> <td>CPU 221</td> </tr> <tr> <td></td> <td>1001=</td> <td>CPU 226/CPU 226XM</td> </tr> </table>	xxxx=	0000=	CPU 222		0010=	CPU 224		0110=	CPU 221		1001=	CPU 226/CPU 226XM
xxxx=	0000=	CPU 222											
	0010=	CPU 224											
	0110=	CPU 221											
	1001=	CPU 226/CPU 226XM											

SMB7: 保留

SMB7为将来使用而保留。

SMW22到SMW26: 扫描时间

如表D-9所示, SMW22、SMW24和SMW26提供扫描时间信息: 以毫秒计的最短扫描时间、最长扫描时间及上次扫描时间。

表D-9 特殊存储器字SMW22到SMW26

SM字	描述 (只读)
SMW22	上次扫描时间
SMW24	进入RUN方式后, 所记录的最短扫描时间
SMW26	进入RUN方式后, 所记录的最长扫描时间

SMB28和SMB29: 模拟电位器

如表D-10所示, SMB28包含代表模拟调节器0位置的数字值。SMB29包含代表模拟调节器1位置的数字值。

表D-10 特殊存储器字节SMB28和SMB29

SM位	描述 (只读)
SMB28	存储模拟调节器0的输入值。在STOP/RUN方式下, 每次扫描时更新该值。
SMB29	存储模拟调节器1的输入值。在STOP/RUN方式下, 每次扫描时更新该值。

SMB30和SMB130: 自由端口控制寄存器

SMB30控制自由端口0的通讯方式, SMB130控制自由端口1的通讯方式。您可以对SMB30和SMB130进行写和读。如表D-11所示, 这些字节设置自由端口通讯的操作方式, 并提供自由端口或者系统所支持的协议之间的选择。

表D-11 特殊存储器字节SMB30

口0	口1	描述																		
SMB30的格式	SMB130的格式	自由口模式控制字节 <div style="text-align: center;"> <table style="border-collapse: collapse; margin: auto;"> <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: right;">7</td> <td style="border: 1px solid black; padding: 2px;">p</td> <td style="border: 1px solid black; padding: 2px;">p</td> <td style="border: 1px solid black; padding: 2px;">d</td> <td style="border: 1px solid black; padding: 2px;">b</td> <td style="border: 1px solid black; padding: 2px;">b</td> <td style="border: 1px solid black; padding: 2px;">b</td> <td style="border: 1px solid black; padding: 2px;">m</td> <td style="border: 1px solid black; padding: 2px;">m</td> <td style="text-align: left;">0</td> </tr> </table> </div>	MSB							LSB	7	p	p	d	b	b	b	m	m	0
MSB							LSB													
7	p	p	d	b	b	b	m	m	0											
SM30.0和SM30.1	SM130.0和SM130.1	mm: 协议选择 00=点到点接口协议 (PPI/从站模式) 01=自由口协议 10=PPI/主站模式 11=保留 (缺省是PPI/从站模式) 注意: 当选择mm=10 (PPI主站), PLC将成为网络的一个主站, 可以执行NETR和NETW指令。在PPI模式下忽略2到7位。																		
SM30.2到SM30.4	SM130.2到SM130.4	bbb: 自由口波特率 000=38,400波特 100=2,400波特 001=19,200波特 101=1,200波特 010=9,600波特 110=115,200波特 011=4,800波特 111=57,600波特																		
SM30.5	SM130.5	d: 每个字符的数据位 0=8位/字符 1=7位/字符																		
SM30.6和SM30.7	SM130.6和SM130.7	pp: 校验选择 00=不校验 10=不校验 01=偶校验 11=奇校验																		

表D-14 特殊存储器字节SMB36-SMD62

SM位	描述
SM36.0-SM36.4	保留
SM36.5	HSC0当前计数方向位: 1=增计数
SM36.6	HSC0当前值等于预置值位: 1=等于
SM36.7	HSC0当前值大于预置值位: 1=大于
SM37.0	复位的有效控制位: 0=高电平复位有效, 1=低电平复位有效
SM37.1	保留
SM37.2	正交计数器的计数速率选择: 0=4x计数速率; 1=1x速率
SM37.3	HSC0方向控制位: 1=增计数
SM37.4	HSC0更新方向: 1=更新方向
SM37.5	HSC0更新预置值: 1=向HSC0写新的预置值
SM37.6	HSC0更新当前值: 1=向HSC0写新的初始值
SM37.7	HSC0有效位: 1=有效
SMD38	HSC0新的初始值
SMD42	HSC0新的预置值
SM46.0-SM46.4	保留
SM46.5	HSC1当前计数方向: 1=增计数
SM46.6	HSC1当前值等于预置值位: 1=等于
SM46.7	HSC1当前值大于预置值位: 1=大于
SM47.0	HSC1复位有效电平控制位: 0=高电平, 1=低电平
SM47.1	HSC1启动有效电平控制位: 0=高电平, 1=低电平
SM47.2	HSC1正交计数器速率选择: 0=4x速率, 1=1x速率
SM47.3	HSC1方向控制位: 1=增计数
SM47.4	HSC1更新方向: 1=更新方向
SM47.5	HSC1更新预置值: 1=向HSC1写新的预置值
SM47.6	HSC1更新当前值: 1=向HSC1写新的初始值
SM47.7	HSC1有效位: 1=有效
SMD48	HSC1新的初始值
SMD52	HSC1新的预置值
SM56.0-SM56.4	保留
SM56.5	HSC2当前计数方向: 1=增计数
SM56.6	HSC2当前值等于预置值位: 1=等于
SM56.7	HSC2当前值大于预置值位: 1=大于
SM57.0	HSC2复位有效电平控制位: 0=高电平, 1=低电平
SM57.1	HSC2启动有效电平控制位: 0=高电平, 1=低电平
SM57.2	HSC2正交计数器速率选择: 0=4X速率, 1=1X速率
SM57.3	HSC2方向控制位: 1=增计数
SM57.4	HSC2更新方向: 1=更新方向
SM57.5	HSC2更新预置值: 1=向HSC2写新的预置值
SM57.6	HSC2更新当前值: 1=向HSC2写新的初始值
SM57.7	HSC2有效位: 1=有效
SMD58	HSC2新的初始值
SMD62	HSC2新的预置值

SMB66到SMB85: PTO/PWM寄存器

如表D-15所示, SMB66到SMB85用于监视和控制脉冲串输出 (PTO) 和脉宽调制 (PWM) 功能。对于这些位的完整描述见第6章的脉冲高速输出指令的信息。

表D-15 特殊存储字节SMB66-SMB85

SM位	描述
SM66.0-SM66.3	保留
SM66.4	PTO0包络终止: 0=无错, 1=由于增量计算错误而终止
SM66.5	PTO0包络终止: 0=不由用户命令终止; 1=由用户命令终止
SM66.6	PTO0管道溢出 (当使用外部包络时由系统清除, 否则由用户程序清除): 0=无溢出, 1=有溢出
SM66.7	PTO0空闲位: 0=PTO忙, 1=PTO空闲
SM67.0	PTO0/PWM0更新周期: 1=写新的周期值
SM67.1	PWM0更新脉冲宽度值: 1=写新的脉冲宽度
SM67.2	PTO0更新脉冲量: 1=写新的脉冲量
SM67.3	PTO0/PWM0基准时间单元: 0=1 μs/格, 1=1 ms/格
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 (周期和脉冲数存在SM存储器中), 1=多段操作 (包络表存在V存储器区)
SM67.6	PTO0/PWM0模式选择: 0=PTO, 1=PWM
SM67.7	PTO0/PWM0有效位: 1=有效
SMW68	PTO0/PWM0周期 (2~65,535个时间基准)
SMW70	PWM0脉冲宽度值 (0~65,535个时间基准)
SMD72	PTO0脉冲计数值 (1~2 ³² -1)
SM76.0-SM76.3	保留
SM76.4	PTO1包络终止; 0=无错, 1=由于增量计算错误终止
SM76.5	PTO1包络终止; 0=不由用户命令终止; 1=由用户命令终止
SM76.6	PTO1管道溢出 (当使用外部包络时由系统清除, 否则由用户程序清除): 0=无溢出, 1=有溢出
SM76.7	PTO1空闲位: 0=PTO忙, 1=PTO空闲
SM77.0	PTO1/PWM1更新周期值: 1=写新的周期值
SM77.1	PWM1更新脉冲宽度值: 1=写新的脉冲宽度
SM77.2	PTO1更新脉冲计数值: 1=写新的脉冲量
SM77.3	PTO1/PWM1时间基准: 0=1 μs/点, 1=1 ms/点
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 (周期和脉冲数存在SM存储器中), 1=多段操作 (包络表存在V存储器区)
SM77.6	PTO1/PWM1模式选择: 0=PTO, 1=PWM
SM77.7	PTO1/PWM1有效位: 1=有效
SMW78	PTO1/PWM1周期值 (2到65,535个时间基准)
SMW80	PWM1脉冲宽度值 (0到65,535个时间基准)
SMD82	PTO1脉冲计数值 (1到2 ³² -1)

SMB86到SMB94和SMB186到SMB194: 接收信息控制

如表D-16所示, SMB86到SMB94和SMB186到SMB194用于控制和读出接收信息指令的状态。

表D-16 特殊存储器字节SMB86到SMB94, SMB186到SMB194

□0	□1	描述								
SMB86	SMB186	接收信息状态字节 <div style="text-align: center;"> $\begin{array}{cccccccc} & & & & & & & \text{LSB} \\ & & & & & & & 0 \\ \text{MSB} & & & & & & & \\ 7 & & & & & & & \end{array}$ <table border="1" style="margin: auto;"> <tr> <td>n</td> <td>r</td> <td>e</td> <td>0</td> <td>0</td> <td>t</td> <td>c</td> <td>p</td> </tr> </table> </div> <p>1=接收用户的禁止命令终止接收信息 1=接收信息终止: 输入参数错误或无起始或结束条件 1=收到结束字符 1=接收信息终止: 超时 1=接收信息终止: 超出最大字符数 1=接收信息终止: 奇偶校验错误</p>	n	r	e	0	0	t	c	p
n	r	e	0	0	t	c	p			
SMB87	SMB187	接收信息控制字节 <div style="text-align: center;"> $\begin{array}{cccccccc} & & & & & & & \text{LSB} \\ & & & & & & & 0 \\ \text{MSB} & & & & & & & \\ 7 & & & & & & & \end{array}$ <table border="1" style="margin: auto;"> <tr> <td>en</td> <td>sc</td> <td>ec</td> <td>il</td> <td>c/m</td> <td>tmr</td> <td>bk</td> <td>0</td> </tr> </table> </div> <p>en: 0=禁止接收信息功能。 1=允许接收信息功能。 每次执行RCV指令时检查允许/禁止接收信息位。</p> <p>sc: 0=忽略SMB88或SMB188。 1=使用SMB88或SMB188的值检测起始信息。</p> <p>ec: 0=忽略SMB89或SMB189。 1=使用SMB89或SMB189的值检测结束信息。</p> <p>il: 0=忽略SMW90或SMW190。 1=使用SMW90或SMW190的值检测空闲状态。</p> <p>c/m: 0=定时器是内部字符定时器。 1=定时器是信息定时器。</p> <p>tmr: 0=忽略SMW92或SMW192。 1=当SMW92或SMW192中的定时时间超出时终止接收。</p> <p>bk: 0=忽略中断条件。 1=用中断条件作为信息检测的开始。</p>	en	sc	ec	il	c/m	tmr	bk	0
en	sc	ec	il	c/m	tmr	bk	0			
SMB88	SMB188	信息字符的开始								
SMB89	SMB189	信息字符的结束								
SMW90	SMW190	空闲行时间间隔用毫秒给出。在空闲行时间结束后接收的第一个字符是新信息的开始。								
SMW92	SMW192	字符间/信息间定时器超时值(用毫秒表示)。如果超过时间,就停止接收信息。								
SMB94	SMB194	接收字符的最大数(1到255字节)。 注意: 这个区一定要设为希望的最大缓冲区,即使不使用字符计数信息终止。								

SMW98: 扩展I/O总线错误

如表D-17所示, SMW98给出有关扩展I/O总线的错误数的信息。

表D-17 特殊存储器字节SMW98

SM位	描述
SMW98	当扩展总线出现校验错误时, 该处每次增加1。当系统得电时或用户程序写入零, 可以进行清零。

SMB130: 自由口控制寄存器 (见SMB30)

参考表D-11。

SMB131到SMB165: HSC3、HSC4和HSC5寄存器

如表D-18所示, SMB131到SMB165用于监视和控制高速计数器HSC3、HSC4和HSC5的操作。

表D-18 特殊存储器字节SMB131-SMB165

SM位	描述
SMB131-SMB135	保留
SM136.0-SM136.4	保留
SM136.5	HSC3当前计数方向状态位: 1=增计数
SM136.6	HSC3当前值等于预置值状态位: 1=等于
SM136.7	HSC3当前值大于预置值状态位: 1=大于
SM137.0-SM137.2	保留
SM137.3	HSC3方向控制位: 1=增计数
SM137.4	HSC3更新方向: 1=更新方向
SM137.5	HSC3更新设定值: 1=向HSC3写入新预置值
SM137.6	HSC3更新当前值: 1=向HSC3写新的初始值
SM137.7	HSC3有效位: 1=有效
SMD138	HSC3新初始值
SMD142	HSC3新预置值
SM146.0-SM146.4	保留
SM146.5	HSC4当前计数方向状态位: 1=增计数
SM146.6	HSC4当前值等于预置值状态位: 1=等于
SM146.7	HSC4当前值大于预置值状态位: 1=大于
SM147.0	复位的有效控制位: 0=高电平有效, 1=低电平有效
SM147.1	保留
SM147.2	正交计数器的计数速率选择: 0=4x计数速率, 1=1x计数速率
SM147.3	HSC4方向控制位: 1=增计数
SM147.4	HSC4更新方向: 1=更新方向
SM147.5	HSC4更新预置值: 1=向HSC4写新的预置值
SM147.6	HSC4更新当前值: 1=向HSC4写新的初始值
SM147.7	HSC4有效位: 1=有效
SMD148	HSC4新初始值
SMD152	HSC4预置值
SM156.0-SM156.4	保留
SM156.5	HSC5当前计数方向状态位: 1=增计数
SM156.6	HSC5当前值等于预置值状态位: 1=等于

表D-18 特殊存储字节SMB131-SMB165

SM位	描述
SM156.7	HSC5当前值大于预置值状态位: 1=大于
SM157.0-SM157.2	保留
SM157.3	HSC5方向控制位: 1=增计数
SM157.4	HSC5更新方向: 1=更新方向
SM157.5	HSC5更新预置值: 1=向HSC5写新的预置值
SM157.6	HSC5更新当前值: 1=向HSC5写新的初始值
SM157.7	HSC5有效位: 1=有效
SMD158	HSC5新初始值
SMD162	HSC5预置值

SMB166到SMB185: PTO0, PTO1包络定义表

如表D-19所示, SMB166到SMB194用来显示包络步的数量和包络表的地址和V存储器区中表的地址。

表D-19 特殊存储字节SMB166-SMB185

SM位	描述
SMB166	PTO0的包络步当前计数值
SMB167	保留
SMW168	PTO0的包络表V存储器地址(从V0开始的偏移量)
SMB170	线性PTO0状态字节
SMB171	线性PTO0结果字节
SMD172	指定线性PTO0发生器工作在手动模式时产生的频率。频率是一个以Hz为单位的双整型值。SMB172是MSB, 而SMB175是LSB
SMB176	PTO1的包络步当前计数值
SMB177	保留
SMW178	PTO1的包络表V存储器地址(从V0开始的偏移量)
SMB180	线性PTO1状态字节
SMB181	线性PTO1结果字节
SMD182	指定线性PTO1发生器工作在手动模式时产生的频率。频率是一个以Hz为单位的双整型值。SMB182是MSB, 而SMB178是LSB

SMB186到SMB194: 接收信息控制 (见SMB86-SMB94)

参考表D-16。

SMB200到SMB549: 智能模块状态

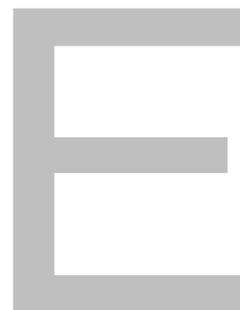
如表D-20所示, SMB200到SMB549预留存储智能扩展模块的信息。如EM277PROFIBUS-DP模块。参见附录A可得到您的模块如何使用SMB200到SMB549的信息以及您的指定模块的规范。

如果您使用版本2.2之前的CPU, 您要将所有的智能模块放置在所有非智能模块之前紧邻CPU的位置, 以确保其兼容性。

表D-20 特殊存储字节SMB200-SMB549

特殊存储字节SMB200-SMB549							
智能模块0	智能模块1	智能模块2	智能模块3	智能模块4	智能模块5	智能模块6	描述
SMB200-SM B215	SMB250-SM B265	SMB300-SM B315	SMB350-SM B365	SMB400-SM B415	SMB450-SM B465	SMB500-SM B515	模块名称 (16个 ASCII字符)
SMB216-SM B219	SMB266-SM B269	SMB316-SM B319	SMB366-SM B369	SMB416-SM B419	SMB466-SM B469	SMB516-SM B519	S/W修订号 (4个 ASCII字符xxxx)
SMW220	SMW270	SMW320	SMW370	SMW420	SMW470	SMW520	错误代码
SMB222-SM B249	SMB272-SM B299	SMB322-SM B349	SMB372-SM B399	SMB422-SM B449	SMB472-SM B499	SMB522-SM B549	与特定模块类型相关的 信息

S7-200订货号



CPU	订货号
CPU221 DC/DC/DC 6输入/4输出	6ES7 211-0AA23-0XB0
CPU221 AC/DC/继电器 6输入/4继电器输出	6ES7 211-0BA23-0XB0
CPU222 DC/DC/DC 8输入/6输出	6ES7 212-1AB23-0XB0
CPU222 AC/DC/继电器 8输入/6继电器输出	6ES7 212-1BB23-0XB0
CPU224 DC/DC/DC 14输入/10输出	6ES7 214-1AD23-0XB0
CPU224 AC/DC/继电器 14输入/10继电器输出	6ES7 214-1BD23-0XB0
CPU224XP DC/DC/DC 14输入/10输出	6ES7 214-2AD23-0XB0
CPU224XP AC/DC/继电器 14输入/10继电器输出	6ES7 214-2BD23-0XB0
CPU226 DC/DC/DC 24输入/16输出	6ES7 216-2AD23-0XB0
CPU226 AC/DC/继电器 24输入/16继电器输出	6ES7 216-2BD23-0XB0
扩展模块	订货号
EM221 数字量输入 8 × 24 VDC	6ES7 221-1BF22-0XA0
EM221 数字量输入 8 × 120/230 VAC	6ES7 221-1EF22-0XA0
EM221 数字量输入 16 × 24 VDC	6ES7 221-1BH22-0XA0
EM222 数字量输入 8 × 24 VDC	6ES7 222-1BF22-0XA0
EM222 8路继电器输出	6ES7 222-1HF22-0XA0
EM222 数字量输出 8 × 120/230 VDC	6ES7 222-1EF22-0XA0
EM222 数字量输出 4 × 24 VDC-5A	6ES7 222-1BD22-0XA0
EM222 数字量输出 4 × 继电器-10A	6ES7 222-1HD22-0XA0
EM223 24 VDC 数字组合4输入/4输出	6ES7 223-1BF22-0XA0
EM223 24 VDC 数字组合4输入/4继电器输出	6ES7 223-1HF22-0XA0
EM223 24 VDC 数字组合8输入/8输出	6ES7 223-1BH22-0XA0
EM223 24 VDC 数字组合8输入/8继电器输出	6ES7 223-1PH22-0XA0
EM223 24 VDC 数字组合16输入/16输出	6ES7 223-1BL22-0XA0
EM223 24 VDC 数字组合16输入/16继电器输出	6ES7 223-1PL22-0XA0
EM231 模拟量输入, 4输入	6ES7 231-0HC22-0XA0
EM231 模拟量输入热电阻, 2输入	6ES7 231-7PB22-0XA0
EM231 模拟量输入热电偶, 4输入	6ES7 231-7PD22-0XA0
EM231 模拟量输出, 2输出	6ES7 232-0HB22-0XA0
EM235 模拟量组合, 4输入/1输出	6ES7 235-0KD22-0XA0
EM241 Modem模块	6ES7 241-1AA22-0XA0
EM253 位控模块	6ES7 253-1AA22-0XA0
EM277 PROFIBUS-DP	6ES7 277-0AA22-0XA0
(CP243-2) AS-i接口模块	6GK7 243-2AX01-0XA0
(CP243-1 IT) 因特网模块 (CD上有电子文档)	6GK7 243-1GX00-0XE0
(CP243-1) 以太网模块 (CD上有电子文档)	6GK7 243-1EX00-0XE0

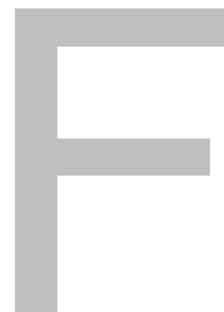
卡和电缆	订货号
存储卡, 32K (用户程序)	6ES7 291-8GE20-0XA0
存储卡, 64K (用户程序, 配方和数据记录)	6ES7 291-8GF23-0XA0
存储卡, 256K (用户程序, 配方和数据记录)	6ES7 291-8GH23-0XA0
带电池卡的实时时钟	6ES7 297-1AA23-0XA0
BC 293, CPU 22x电池卡	6ES7 291-8BA20-0XA0
电缆, I/O扩展, 0.8米, CPU22x/EM	6ES7 290-6AA20-0XA0
电缆, RS32/PPI多主站	6ES7 901-3CB30-0XA0
电缆, USB/PPI多主站	6ES7 901-3DB30-0XA0
软件	订货号
STEP 7-Micro/WIN 32 (V4.0) 单用户授权 (CD-ROM)	6ES7 810-2CC03-0YX0
STEP 7-Micro/WIN 32 (V4.0) 升级授权 (CD-ROM)	6ES7 810-2CC03-0YX3
S7-200工具箱: TP070的TP设计器, 版本1.0 (CD-ROM)	6ES7 850-2BC00-0YX0
STEP 7-Micro/WIN附加件: STEP 7-Micro/WIN 32指令库, V1.1 (CD-ROM)	6ES7 830-2BC00-0YX0
S7-200 PC Access V1.0 (OPC Server) 单一许可证	6ES7 840-2CC01-0YX0
S7-200 PC Access V1.0 (OPC Server) 多许可证	6ES7 840-2CC01-0YX1
WinCC Flexible Micro组态软件	6AV6 610-0AA01-0AA0

通讯卡	订货号
CP 5411: 短AT ISA	6GK1 541-1AA00
CP 5511: PCMCIA, Type II	6GK1 551-1AA00
CP 5611: PCI卡 (3.0版或更高版)	6GK1 561-1AA00
手册	订货号
S7-200可编程控制器系统手册 (德语)	6ES7 298-8FA24-8AH0
S7-200可编程控制器系统手册 (英语)	6ES7 298-8FA24-8BH0
S7-200可编程控制器系统手册 (法语)	6ES7 298-8FA24-8CH0
S7-200可编程控制器系统手册 (西班牙语)	6ES7 298-8FA24-8DH0
S7-200可编程控制器系统手册 (意大利语)	6ES7 298-8FA24-8EH0
S7-200点到点接口通讯手册 (英语/德语)	6ES7 298-8GA00-8XH0
TP070 触摸屏用户手册 (英语)	6AV6 591-1DC01-0AB0
TP170 micro操作手册 (英语)	6AV6 691-1DB01-0AB0
CP 243-2 SIMATIC NET AS-i接口主站手册 (英语)	6GK7 243-2AX00-8BA0
WinCC Flexible Micro用户手册 (英语)	6AV6 691-1AA01-0AB0

电缆, 网络连接器和中继器	订货号
MPI电缆	6ES7 901-0BF00-0AA0
PROFIBUS网络电缆	6XV1 830-0AH10
带编程接口的网络总线连接器, 垂直电缆出口	6ES7 972-0BB11-0XA0
网络总线连接器, (不带编程接口), 垂直电缆出口	6ES7 972-0BA11-0XA0
RS485总线连接器, 35°电缆出口 (不带编程接口)	6ES7 972-0BA40-0XA0
RS485总线连接器, 35°电缆出口 (带编程接口)	6ES7 972-0BB40-0XA0
CPU22x/EM连接器块, 7端子, 可移动式	6ES7 292-1AD20-0AA0
CPU22x/EM连接器块, 12端子, 可移动式	6ES7 292-1AE20-0AA0
CPU22x/EM连接器块, 14端子, 可移动式	6ES7 292-1AF20-0AA0
CPU22x/EM连接器块, 18端子, 可移动式	6ES7 292-1AG20-0AA0
RS-485 IP 20中继器, 隔离型	6ES7 972-0AA00-0XA0
操作员界面	订货号
文本显示TD200	6ES7 272-0AA30-0YA0
文本显示TD200C (自定义面板) ¹	6ES7 272-1AA10-0YA0
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触摸屏	6AV6 545-0AA15-2AX0
TP170 micro触摸屏	6AV6 640-0CA01-0AX0
附件	订货号
DIN导轨固定端子	6ES5 728-8MA11
12位扇出连接器 (CPU221, CPU222) 10包	6ES7 290-2AA00-0XA0
备用盖板套件, 包括以下每种 4 个: 7、12、14、18、2x12、2x14端子块盖板, CPU盖板, EM盖板	6ES7 291-3AX20-0XA0
8位模拟器	6ES7 274 1XF00-0XA0
14位模拟器	6ES7 274 1XH00-0XA0
24位模拟器	6ES7 274 1XK00-0XA0
TD200C面板薄膜层 (空白薄膜层包)	6ES7 272-1AF00-7AA0

¹ 包括一个用于自定义的空白薄膜层。对于附加的空白薄膜层, 请订购TD200C面板薄膜层。

STL指令的执行时间



指令执行时间对于对时间要求比较苛刻的应用来说非常重要。指令的执行时间如表F-2所示。



提示

当您使用表F-2的执行时间时，要考虑使能位，间接寻址以及使用某些存储区域对指令的执行时间的影响。这些因素可以直接地影响所列的执行时间。

使能位的影响

表F-2所列的执行时间是当使能位接通（顶端堆栈=1或接通）时该指令逻辑或功能所需的时间。

当使能位未接通时，指令的执行时间为1 μ s。

间接寻址的影响

表F-2所列为使用操作数或常数的直接寻址执行该指令的逻辑或功能所需要的时间。

当使用间接寻址的操作数时，指令中使用的每一个间接寻址操作数会使指令执行时间增加14 μ s。

访问某些存储区域的影响

访问某些存储区域，如AI、AQ、L和累加器需要额外的执行时间。

表F-1所示为在操作数中使用这些存储区需在指令执行时间上增加的时间。

表F-1 访问存储区的时间增加值

存储区	执行时间增加值
本机模拟量输入 (AI)	
未使能滤波	9.4 μ s
使能滤波	8.4 μ s
扩展模拟量输入 (AI)	
未使能滤波	134 μ s
使能滤波	8.4 μ s
本机模拟量输出 (AQ)	92 μ s
扩展模拟量输出 (AQ)	48 μ s
局域存储器 (L)	2.8 μ s
累加器 (AC)	2.8 μ s

表F-2 指令执行时间

指令	μs	指令	μs
= 使用: I SM、T、C、V、S、Q、M L	0.24 1.3 10.5	BITIM	16
+D	29	BIR 使用: 本机输入 扩展输入	23 30
-D	29	BIW 使用: 本机输出 扩展输出	24 32
*D	47	BMB 总时间=基本时间+(长度*LM) 基本时间(固定长度) 基本时间(变长度) 长度系数(LM)	10 28 5.7
/D	250	BMD 总时间=基本时间+(长度*LM) 基本时间(固定长度) 基本时间(变长度) 长度系数(LM)	11 29 10.6
+I	25	BMW 总时间=基本时间+(长度*LM) 基本时间(固定长度) 基本时间(变长度) 长度系数(LM)	10 28 8.6
-I	25	BTI	16
*I	37	CALL 无参数 有参数 总时间=基本时间+ Σ (操作数处理时间) 基本时间 操作数处理时间 位(输入, 输出) 字节(输入, 输出) 字(输入, 输出) 双字(输入, 输出)	9 14 10、11 8、7 10、9 12、10
/I	64	注意: 对输出操作数的处理在子程序返回时	
=I 使用: 本机输出 扩展输出	16 24	CEVNT	24
+R	71 典型值 99 最大值	CFND 最大时间= 基本时间+N1*(LM1*N2)+LM2) 基本时间 长度系数1(LM1) 长度系数2(LM2) N1是源字符串的长度 N2是字符串的长度	35 8.6 9.5
-R	72 典型值 100 最大值	CITIM	23
*R	56 典型值 166 最大值	COS	900 典型值 1070 最大值
/R	177 典型值 230 最大值	CRET 使能位接通 使能位未接通	16 0.8
A 使用: I SM、T、C、V、S、Q、M L	0.22 0.72 6.1	CRETI 使能位未接通	0.2
AB <=、=、>=、>、<、<>	18	CSCRE	3.1
AD <=、=、>=、>、<、<>	27	CTD 计数输入跳变 否则	27 19
AENO	0.4	CTU 计数输入跳变 否则	31 19
AI 使用: 本机输入 扩展输入	15 21	CTUD 计数输入跳变 否则	37 24
ALD	0.22	DECB	16
AN 使用: I SM、T、C、V、S、Q、M L	0.22 0.72 6.1	DECD	22
ANDB	19	DECO	19
ANDD	30	DECW	20
ANDW	25	DISI	9
ANI 使用: 本机输入 扩展输入	15 21	DIV	67
AR <=、=、>=、>、<、<>	29		
AS=、<>总时间 = 基本时间+(LM*N) 基本时间 长度系数(LM) N 比较的字符数	33 6.3		
ATCH	12		
ATH 总时间=基本时间+(长度*LM) 基本时间(固定长度) 基本时间(变长度) 长度系数(LM)	23 31 10.2		
ATT	36		
AW <=、=、>=、>、<、<>	23		
BCDI	35		

指令	μs	指令	μs
DLED	14	LD 使用: I SM、T、C、V、S、Q、M L	0.22 0.8 6
DTA	302	LDB<=、=、>=、>、<、<>	18
DTI	21	LDD<=、=、>=、>、<、<>	27
DTCH	12	LDI 使用: 本机输入 扩展输入	15 21
DTR	35 典型值 40最大值	LDN 使用: I SM、T、C、V、S、Q、M L	0.3 0.9 6.1
DTS	305	LDNI 使用: 本机输入 扩展输入	15 21
ED	8	LDR<=、=、>=、>、<、<>	29
ENCO	24最大值	LDS	0.22
END 使能位未接通	0.2	LDSAS=、<> 总时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N 比较的字符数	33 6.3
ENI	11	LDW<=、=、>=、>、<、<>	24
EU	8	LIFO	37
EXP	720典型值 860最大值	LN	680典型值 820最大值
FIFO 总时间=基本时间+ (长度 * LM) 基本时间 长度系数 (LM)	30 7	LPP	0.22
FILL 总时间=基本时间+ (长度 * LM) 基本时间 (固定长度) 基本时间 (变长度) 长度系数 (LM)	15 29 3.2	LPS	0.24
FND <、=、>、<> 时间=基本时间+ (长度 * LM) 基本时间 长度系数 (LM)	39 6.5	LRD	0.22
FOR 时间=基本时间+ (环数 * LM) 基本时间 循环系数 (LM)	35 28	LSCR	7.3
GPA	16	MOVb	15
HDEF	18	MOVd	20
HSC	30	MOVr	20
HTA 总时间=基本时间+ (长度 * LM) 基本时间 (固定长度) 基本时间 (变长度) 长度系数 (LM)	20 28 5.2	MOVw	18
IBCD	52	MUL	37
INCB	15	NETR	99
INCD	22	NETW 时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N为发送的字节数	95 4
INCW	20	NEXT	0
INT 1个中断的典型值	24	NOP	0.22
INVB	16	NOT	0.22
INVD	22	O 使用: I SM、T、C、V、S、Q、M L	0.22 0.72 6.4
INVW	20	OB <=、=、>=、>、<、<>	18
ITA	136	OD <=、=、>=、>、<、<>	26
ITB	17	OI 使用: 本机输入 扩展输入	15 21
ITD	20	OLD	0.22
ITS	139	ON 使用: I SM、T、C、V、S、Q、M L	0.22 0.72 6.4
JMP	1.8		
LBL	0.22		

指令	μS	指令	μS
ONI 使用: 本机输入 扩展输入	15 21	RRW 时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N是移位数	26 1.2
OR<=、=、>=、>、<、<>	29	RTA 总时间=基本时间+ (LM*N) * 基本时间 (结果中的第一个数) 长度系数 (LM) N结果中的额外的数字的数量	149 96
ORB	19	RTS 时间=基本时间+ (LM * N) 基本时间 (x) 长度系数 (LM) N为结果中额外数字的数量	154 96
ORD	29	S 长度=1指定为一个常数 否则: 总时间=基本时间+ (长度 * LM) 基本时间 长度系数 (LM) 如果长度存为变量, 加入基本时间中	2.9 14 0.5 17
ORW	25	SCAT 总时间=基本时间+ (LM*N) * 基本时间 长度系数 (LM) N是附加的字符的数量	30 5.3
OS=、<>总时间 = 基本时间+ (LM*N *) 基本时间 长度系数 (LM) N比较的字符数	33 6.3	SCPY 总时间=基本时间+ (LM*N) * 基本时间 长度系数 (LM) N复制的字符的数量	27 4.6
OW <=、=、>=、>、<、<>	24	SCRE	0.24
PID 典型 手动到自动转换 系数重算 自整定	400 800最大值 770最大值 650最大值	SCRT	10
PLS: 使用: PWM PTO单数 PTO复数	31 36 50	SEG	15
R 长度=1定义为常数 计数器基本时间 (C) 定时器基本时间 (T) 其它操作的基本时间 否则: 时间 = 基本时间+ (长度 * LM) 计数器基本时间 定时器基本时间 (T) 其它操作的基本时间 操作数C的长度系数 (LM) 操作数T的长度系数 (LM) 其它操作数的长度系数 (LM) 如果长度作为变量存储, 加入基本时间中	9.3 16 2.9 8.6 8.3 14 5.1 9.9 0.5 17	SFND 最大时间= 基本时间+ ((N1-N2) * LM2) + (N2 * LM1) 基本时间 长度系数1 (LM1) 长度系数2 (LM2) N1是源字符串的长度 N2是字符串的长度	39 7.6 6.8
RCV	51	SHRB 时间=基本时间+ (长度 * LM1) + ((长度/8) * LM2) 基本时间 (固定长度) 基本时间 (可变长度) 长度系数1 (LM1) 长度系数2 (LM2)	48 52 1.0 1.5
RET	16	SI 时间=基本时间+ (长度 * LM) 基本时间 LM使用本机输出 LM使用扩展输出 如果长度存为变量, 加入基本时间中	8.9 13 21 17
RI 时间 = 基本时间+ (length * LM) 基本时间 长度系数 (LM) 使用本机输出 长度系数 (LM) 使用扩展输出 输入 如果长度作为变量存储, 加入基本时间中	8.9 13 21 17	SIN	900典型值 1070最大值
RLB 时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N是移位数	23 0.2	SLB 时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N是移位数	23 0.2
RLD 时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N是移位数	28 1.4	SLD 时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N是移位数	29 1.1
RLW 时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N是移位数	27 0.9		
ROUND	56典型值 110最大值		
RRB 时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N是移位数	22 0.5		
RRD 时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N是移位数	28 1.7		

指令	μs	指令	μs
SLEN	21	STOP 使能位未接通	4
SLW 时间=基本时间+ (LM + N) 基本时间 长度系数 (LM)) N是移位数	27 0.6	STR 总时间=基本时间+ (LM * N) 基本时间 (第一个源字符的) 长度系数 (LM) N额外的源字符的数量	51 81
SPA	371	SWAP	17
SQRT	460典型值 550最大值	TAN	1080典型值 1300最大值
SRB 时间=基本时间+ (LM + N) 基本时间 长度系数 (LM)) N是移位数	22 0.6	TODR	331
SRD 时间=基本时间+ (LM + N) 基本时间 长度系数 (LM)) N是移位数	28 1.5	TODRX 夏令时纠正	391 典型值 783典型值
SRW 时间=基本时间+ (LM + N) 基本时间 长度系数 (LM)) N是移位数	27 1	TODW	436
SSCPY 总时间=基本时间+ (LM * N) 基本时间 长度系数 (LM) N复制的字符的数量	42 5.3	TODWX	554
STD 总时间=基本时间+ (LM * N) 基本时间 (第一个源字符的) 长度系数 (LM) N额外的源字符的数量	69 27	TOF	36
STI 总时间=基本时间+ (LM * N) 基本时间 (第一个源字符的) 长度系数 (LM) N额外的源字符的数量	58 27	TON	33
		TONR	32
		TRUNC	53 典型值 106最大值
		WDR	7
		XMT	42
		XORB	19
		XORD	29
		XORW	25



S7-200快速参考信息

为了便于您查找信息，本章节汇总了以下信息：

- 特殊存储器位
- 中断事件描述
- S7-200CPU存储器范围和特性汇总
- 高速计数器HSC0、HSC1、HSC2、HSC3、HSC4、HSC5
- S7-200指令

表G-1 特殊存储器位

特殊存储器位			
SM0.0	该位始终为1	SM1.0	操作结果=0
SM0.1	首次扫描时为1	SM1.1	结果溢出或非法数值
SM0.2	保持数据丢失时为1	SM1.2	结果为负数
SM0.3	开机上电进入RUN时为1一个扫描周期	SM1.3	被0除
SM0.4	时钟脉冲：30s闭合/30s断开	SM1.4	超出表范围
SM0.5	时钟脉冲：0.5s闭合/0.5s断开	SM1.5	空表
SM0.6	时钟脉冲：闭合1个扫描周期/断开1个扫描周期	SM1.6	BCD到二进制转换出错
SM0.7	开关放置在RUN位置时为1	SM1.7	ASCII到十六进制转换出错

表G-2 中断事件的优先级顺序

中断号	中断描述	优先级分组	按组排列的优先级	
8	通讯口0: 接收字符	通讯 (最高)	0	
9	通讯口0: 发送完成		0	
23	通讯口0: 接收信息完成		0	
24	通讯口1: 接收信息完成		1	
25	通讯口1: 接收字符		1	
26	通讯口1: 发送完成		1	
19	PTO 0完成中断	开关量 (中等)	0	
20	PTO 1完成中断		1	
0	I0.0的上升沿		2	
2	I0.1的上升沿		3	
4	I0.2的上升沿		4	
6	I0.3的上升沿		5	
1	I0.0的下降沿		6	
3	I0.1的下降沿		7	
5	I0.2的下降沿		8	
7	I0.3的下降沿		9	
12	HSC0 CV=PV (当前值=预设值)		10	
27	HSC0方向改变		11	
28	HSC0外部复位		12	
13	HSC1 CV=PV (当前值=预设值)		13	
14	HSC1方向改变		14	
15	HSC1外部复位		15	
16	HSC2 CV=PV (当前值=预设值)		16	
17	HSC2方向改变		17	
18	HSC2外部复位		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外部复位		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	定时器T32 CT=PT中断			2
22	定时器T96 CT=PT中断			3

表G-3 S7-200 CPU存储器范围及特性

描述	CPU221	CPU222	CPU224	CPU224XP	CPU226
用户程序长度 运行模式下编辑 非运行模式下编辑	4096字节 4096字节	4096字节 4096字节	8192字节 12288字节	12288字节 16384字节	16384字节 24576字节
用户数据区	2048字节	2048字节	8192字节	10240字节	10240字节
输入映像寄存器	I0.0-I15.7	I0.0-I15.7	I0.0-I15.7	I0.0-I15.7	I0.0-I15.7
输出映像寄存器	Q0.0-Q15.7	Q0.0-Q15.7	Q0.0-Q15.7	Q0.0-Q15.7	Q0.0-Q15.7
模拟输入（只读）	AIW0-AIW30	AIW0-AIW30	AIW0-AIW62	AIW0-AIW62	AIW0-AIW62
模拟输出（只写）	AQW0-AQW30	AQW0-AQW30	AIW0-AQW62	AIW0-AQW62	AIW0-AQW62
变量存储器（V） ¹	VB0-VB2047	VB0-VB2047	VB0-VB8191	VB0-VB10239	VB0-VB10239
局部存储器（L） ¹	LB0-LB63	LB0-LB63	LB0-LB63	LB0-LB63	LB0-LB63
位存储器（SM）	M0.0-M31.7	M0.0-M31.7	M0.0-M31.7	M0.0-M31.7	M0.0-M31.7
特殊存储器（SM） 只读	SM0.0-SM179.7 SM0.0-SM29.7	SM0.0-SM299.7 SM0.0-SM29.7	SM0.0-SM549.7 SM0.0-SM29.7	SM0.0-SM549.7 SM0.0-SM29.7	SM0.0-SM549.7 SM0.0-SM29.7
定时器	256（T0-T255）	256（T0-T255）	256（T0-T255）	256（T0-T255）	256（T0-T255）
保持接通延时	1ms T0、T64	T0、T64	T0、T64	T0、T64	T0、T64
接通/断开延时	10ms T1-T4、 T65-T68	T1-T4、 T65-T68	T1-T4、 T65-T68	T1-T4、 T65-T68	T1-T4、 T65-T68
接通/断开延时	100ms T5-T31、 T69-T95	T5-T31、 T69-T95	T5-T31、 T69-T95	T5-T31、 T69-T95	T5-T31、 T69-T95
接通/断开延时	1ms T32、T96	T32、T96	T32、T96	T32、T96	T32、T96
接通/断开延时	10ms T33-T36、 T97-T100	T33-T36、 T97-T100	T33-T36、 T97-T100	T33-T36、 T97-T100	T33-T36、 T97-T100
接通/断开延时	100ms T37-T63、 T101-T255	T37-T63、 T101-T255	T37-T63、 T101-T255	T37-T63、 T101-T255	T37-T63、 T101-T255
计数器	C0-C255	C0-C255	C0-C255	C0-C255	C0-C255
高速计数器	HC0-HC5	HC0-HC5	HC0-HC5	HC0-HC5	HC0-HC5
顺控继电器（S）	S0.0-S31.7	S0.0-S31.7	S0.0-S31.7	S0.0-S31.7	S0.0-S31.7
累加器	AC0-AC3	AC0-AC3	AC0-AC3	AC0-AC3	AC0-AC3
跳转/标号	0-255	0-255	0-255	0-255	0-255
调用/子程序	0-63	0-63	0-63	0-63	0-127
中断程序	0-127	0-127	0-127	0-127	0-127
正/负跳变	256	256	256	256	256
PID回路	0-7	0-7	0-7	0-7	0-7
通讯口	0	0	0	0/1	0/1

¹ LB60-LB63为STEP 7-Micro/WIN32的3.0版本或以后的版本软件保留。

表G-4 高速计数器HSC0, HSC3, HSC4, HSC5

模式	HSC0			HSC3	HSC4			HSC5
	计数	方向	复位	计数	计数	方向	复位	计数
0	I0.0			I0.1	I0.3			I0.4
1	I0.0		I0.2		I0.3		I0.5	
2								
3	I0.0	I0.1			I0.3	I0.4		
4	I0.0	I0.1	I0.2		I0.3	I0.4	I0.5	
5								
模式	HSC0			HSC4				
	增计数	减计数	复位	增计数	减计数	复位		
6	I0.0	I0.1		I0.3	I0.4			
7	I0.0	I0.1	I0.2	I0.3	I0.4	I0.5		
8								
模式	HSC0			HSC4				
	A相	B相	复位	A相	B相	复位		
9	I0.0	I0.1		I0.3	I0.4			
10	I0.0	I0.1	I0.2	I0.3	I0.4	I0.5		
11								
模式	HSC0			HSC3				
	计数			计数				
12	Q0.0			Q0.1				

表G-5 高速计数器HSC1和HSC2

模式	HSC1				HSC2			
	计数	减计数	复位	启动	计数	方向	复位	启动
0	I0.6				I1.2			
1	I0.6		I1.0		I1.2		I1.4	
2	I0.6		I1.0	I1.1	I1.2		I1.4	I1.5
3	I0.6	I0.7			I1.2	I1.3		
4	I0.6	I0.7	I1.0		I1.2	I1.3	I1.4	
5	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5
模式	HSC1				HSC2			
	增计数	减计数	复位	启动	增计数	减计数	复位	启动
6	I0.6	I0.7	I1.0		I1.2	I1.3		
7	I0.6	I0.7	I1.0		I1.2	I1.3	I1.4	
8	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5
模式	A相	B相	复位	启动	A相	B相	复位	启动
9	I0.6	I0.7			I1.2	I1.3		
10	I0.6	I0.7	I1.0		I1.2	I1.3	I1.4	
11	I0.6	I0.7	I1.0	I1.1	I1.2	I1.3	I1.4	I1.5

布尔指令			数学、增减指令		
LD	N	装载	+I	IN1, OUT	整数、双整数或实数加法 IN1+OUT=OUT
LDI	N	立即装载	+D	IN1, OUT	
LDN	N	取反后装载	+R	IN1, OUT	
LDNI	N	取反后立即装载	-I	IN1, OUT	整数、双整数或实数减法 OUT-IN1=OUT
A	N	与	-D	IN1, OUT	
AI	N	立即与	-R	IN1, OUT	
AN	N	取反后与	MUL	IN1, OUT	整数乘法 (16*16->32)
ANI	N	取反后立即与	*I	IN1, OUT	整数、双整数或实数乘法 IN1 * OUT = OUT
O	N	或	*D	IN1, OUT	
OI	N	立即或	*R	IN1, IN2	
ON	N	取反后或	DIV	IN1, OUT	整数除法 (16/16->32)
ONI	N	取反后立即或	/I	IN1, OUT	整数、双整数或实数除法 OUT / IN1 = OUT
LDBx	N1, N2	装载字节比较的结果 N1(x:<,<=,>=,>,<>) N2	/D,	IN1, OUT	
ABx	N1, N2	与字节比较的结果 N1(x:<,<=,>=,>,<>) N2	/R,	IN1, OUT	
OBx	N1, N2	或字节比较的结果 N1(x:<,<=,>=,>,<>)N2	SQRT	IN, OUT	平方根
LDWx	N1, N2	装载字比较的结果 N1(x:<,<=,>=,>,<>)N2	LN	IN, OUT	自然对数
AWx	N1, N2	与字比较的结果 N1(x:<,<=,>=,>,<>)N2	EXP	IN, OUT	自然指数
OWx	N1, N2	或字比较的结果 N1(x:<,<=,>=,>,<>)N2	SIN	IN, OUT	正弦
LDDx	N1, N2	装载双字比较的结果 N1(x:<,<=,>=,>,<>)N2	COS	IN, OUT	余弦
ADx	N1, N2	与双字比较的结果 N1(x:<,<=,>=,>,<>)N2	TAN	IN, OUT	正切
ODx	N1, N2	或双字比较的结果 N1(x:<,<=,>=,>,<>)N2	INCB	OUT	字节、字和双字增1
LDRx	N1, N2	装载实数比较的结果 N1(x:<,<=,>=,>,<>)N2	INCW	OUT	
ARx	N1, N2	与实数比较的结果 N1(x:<,<=,>=,>,<>)N2	INCD	OUT	
ORx	N1, N2	或实数比较的结果 N1(x:<,<=,>=,>,<>)N2	DECB	OUT	字节、字和双字减1
NOT		堆栈取反	DECW	OUT	
EU		检测上升沿	DECD	OUT	
ED		检测下降沿	PID	Table, Loop	PID回路
=	Bit	赋值	定时器和计数器指令		
=I	Bit	立即赋值	TON	Txxx, PT	接通延时定时器
S	Bit,N	置位一个区域	TOF	Txxx, PT	关断延时定时器
R	Bit,N	复位一个区域	TONR	Txxx, PT	带记忆的接通延时定时器
SI	Bit,N	立即置位一个区域	BITIM	OUT	启动间隔定时器
RI	Bit,N	立即复位一个区域	CITIM	IN, OUT	计算间隔定时器
LDSx	IN1,IN2	装载字符串比较结果 IN1 (x: =, <>) IN2	CTU	Cxxx, PV	增计数
ASx	IN1,IN2	与字符串比较结果 IN1 (x: =, <>) IN2	CTD	Cxxx, PV	减计数
OSXI	IN1,IN2	或字符串比较结果 IN1 (x: =, <>) IN2	CTUD	Cxxx, PV	增/减计数
ALD		与装载	实时时钟指令		
OLD		或装载	TODR	T	读实时时钟
LPS		逻辑压栈 (堆栈控制)	TODW	T	写实时时钟
LRD		逻辑读 (堆栈控制)	TODRX	T	扩展读实时时钟
LPP		逻辑弹出 (堆栈控制)	TODWX	T	扩展写实时时钟
LDS	N	装载堆栈 (堆栈控制)	程序控制指令		
AENO		与ENO	END		程序的条件结束
			STOP		切换到STOP模式
			WDR		看门狗复位 (300ms)
			JMP	N	跳到定义的标号
			IBL	N	定义一个跳转的标号
			CALL	N[N1, ...]	调用子程序[N1, ...可以有16个可选参数]
			CRET		从SBR条件返回
			FOR	INDX,INIT,FINAL	For/Next循环
			NEXT		
			LSCR	N	顺控继电器段的启动、转换、条件结束和结束
			SCRT	N	
			CSCRE		
			SCRE		
			DLED	IN	诊断LED

传送、移位、循环和填充指令		表、查找和转换指令	
MOVB IN, OUT MOVW IN, OUT MOVD IN, OUT MOVR IN, OUT	字节、字、双字和实数传送	ATT TABLE, DATA	把数据加到表中
BIR IN, OUT BIW IN, OUT	立即读取传送字节 立即写入传送字节	LIFO TABLE, DATA FIFO TABLE, DATA	从表中取数据
BMB IN, OUT, N BMW IN, OUT, N BMD IN, OUT, N	字节、字和双字块传送	FND= TBL, PTN, INDX FND<> TBL, PTN, INDX FND< TBL, PTN, INDX FND> TBL, PTN, INDX	根据比较条件在表中查找数据
SWAP IN	交换字节	FILL IN, OUT, N	用给定值占满存储器空间
SHRB DATA, S_BIT, N	寄存器移位	BCDI OUT IBCD OUT	把BCD码转换成整数 把整数转换成BCD码
SRB OUT, N SRW OUT, N SRD OUT, N	字节、字和双字右移	BTI IN, OUT ITB IN, OUT ITD IN, OUT DTI IN, OUT	把字节转换成整数 把整数转换成字节 把整数转换成双整数 把双整数转换成整数
SLB OUT, N SLW OUT, N SLD OUT, N	字节、字和双字左移	DTR IN, OUT TRUNC IN, OUT ROUND IN, OUT	把双字转换成实数 把实数转换成双字 把实数转换成双字
RRB OUT, N RRW OUT, N RRD OUT, N	字节、字和双字循环右移	ATH IN, OUT, LEN HTA IN, OUT, LEN ITA IN, OUT, FMT DTA IN, OUT, FM RTA IN, OUT, FM	把ASCII码转换成16进制格式 把16进制格式转换成ASCII码 把整数转换成ASCII码 把双整数转换成ASCII码 把实数转换成ASCII码
RLB OUT, N RLW OUT, N RLD OUT, N	字节、字和双字循环左移	DECO IN, OUT ENCO IN, OUT	解码 编码
逻辑操作		SEG IN, OUT	产生7段格式
ANDB IN1, OUT ANDW IN1, OUT ANDD IN1, OUT	对字节、字和双字取逻辑与	ITS IN, FMT, OUT DTS IN, FMT, OUT RTS IN, FMT, OUT	把整数转为字符串 把双整数转换成字符串 把实数转换成字符串
ORB IN1, OUT ORW IN1, OUT ORD IN1, OUT	对字节、字和双字取逻辑或	STI STR, INDX, OUT STD STR, INDX, OUT STR STR, INDX, OUT	把子字符串转换成整数 把子字符串转换成双整数 把子字符串转换成实数
XORB IN1, OUT XORW IN1, OUT XORD IN1, OUT	对字节、字和双字取逻辑异或	中断	
INVB OUT INWV OUT INVD OUT	对字节、字和双字取反 (1的补码)	CRETI	从中断条件返回
字符串指令		ENI	允许中断
SLEN IN, OUT SCAT IN, OUT SCPY IN, OUT SSCPY IN, INDX, N, OUT CFND IN1, IN2, OUT SFND IN1, IN2, OUT	字符串长度 连接字符串 复制字符串 复制子字符串 字符串中查找第一个字符 在字符串中查找字符串	DISI	禁止中断
		ATCH INT, EVENT DTCH EVENT	给事件分配中断程序 解除事件
		通讯	
		XMT TABLE, PORT RCV TABLE, PORT	自由口传送 自由口接受信息
		TODR TABLE, PORT TODW TABLE, PORT	网络读 网络写
		GPA ADDR, PORT SPA ADDR, PORT	获取口地址 设置口地址
		高速指令	
		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

To

SIEMENS ENERGY & AUTOMATION INC
ATTN: TECHNICAL COMMUNICATIONS
ONE INTERNET PLAZA
PO BOX 4991
JOHNSON CITY TN USA 37602-4991

From

Name: -----
Job Title: -----
Company Name: -----
Street: -----
City and State: -----
Country: -----
Telephone: -----

Please check any industry that applies to you:

- | | |
|--|---|
| <input type="checkbox"/> Automotive | <input type="checkbox"/> Pharmaceutical |
| <input type="checkbox"/> Chemical | <input type="checkbox"/> Plastic |
| <input type="checkbox"/> Electrical Machinery | <input type="checkbox"/> Pulp and Paper |
| <input type="checkbox"/> Food | <input type="checkbox"/> Textiles |
| <input type="checkbox"/> Instrument and Control | <input type="checkbox"/> Transportation |
| <input type="checkbox"/> Nonelectrical Machinery | <input type="checkbox"/> Other _____ |
| <input type="checkbox"/> Petrochemical | |



S7-200存储器范围及特性

中断描述	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
用户程序长度 在运行模式下编辑 不在运行模式下编辑	4096字节 4096字节	4096字节 4096字节	8192字节 12288字节	12288字节 16384字节	16384字节 24576字节
用户数据大小	2048字节	2048字节	8192字节	10240字节	10240字节
输入映像寄存器	I0.0-I15.7	I0.0-I15.7	I0.0-I15.7	I0.0-I15.7	I0.0-I15.7
输出映像寄存器	Q0.0-Q15.7	Q0.0-Q15.7	Q0.0-Q15.7	Q0.0-Q15.7	Q0.0-Q15.7
模拟量输入（只读）	AIW0-AIW30	AIW0-AIW30	AIW0-AIW62	AIW0-AIW62	AIW0-AIW62
模拟量输出（只写）	AQW0-AQW30	AQW0-AQW30	AQW0-AQW62	AQW0-AQW62	AQW0-AQW62
变量存储器（V）	VB0-VB2047	VB0-VB2047	VB0-VB8191	VB0-VB10239	VB0-VB10239
局部存储器（L） ¹	LB0-LB63	LB0-LB63	LB0-LB63	LB0-LB63	LB0-LB63
位存储器（M）	M0.0-M31.7	M0.0-M31.7	M0.0-M31.7	M0.0-M31.7	M0.0-M31.7
特殊存储器（SM） 只读	SM0.0-SM179.7 SM0.0-SM29.7	SM0.0-SM299.7 SM0.0-SM29.7	SM0.0-SM549.7 SM0.0-SM29.7	SM0.0-SM549.7 SM0.0-SM29.7	SM0.0-SM549.7 SM0.0-SM29.7
定时器 有记忆接通延迟	256(T0-T255) T0,T64 10ms T1-T4,T65-T68 100ms T5-T31,T69-T95	256(T0-T255) T0,T64 T1-T4,T65-T68 T5-T31,T69-T95	256(T0-T255) T0,T64 T1-T4,T65-T68 T5-T31,T69-T95	256(T0-T255) T0,T64 T1-T4,T65-T68 T5-T31,T69-T95	256(T0-T255) T0,T64 T1-T4,T65-T68 T5-T31,T69-T95
接通/关断延迟	1ms T32,T96 10ms T33-T36, T97-T100 100ms T37-T63, T101-T255	T32,T96 T33-T36, T97-T100 T37-T63, T101-T255	T32,T96 T33-T36, T97-T100 T37-T63, T101-T255	T32,T96 T33-T36, T97-T100 T37-T63, T101-T255	T32,T96 T33-T36, T97-T100 T37-T63, T101-T255
计数器	C0-C255	C0-C255	C0-C255	C0-C255	C0-C255
高速计数器	HC0-HC5	HC0-HC5	HC0-HC5	HC0-HC5	HC0-HC5
顺序控制继电器（S）	S0.0-S31.7	S0.0-S31.7	S0.0-S31.7	S0.0-S31.7	S0.0-S31.7
累加寄存器	AC0-AC3	AC0-AC3	AC0-AC3	AC0-AC3	AC0-AC3
跳转/标号	0-255	0-255	0-255	0-255	0-255
调用/子程序	0-63	0-63	0-63	0-63	0-127
中断程序	0-127	0-127	0-127	0-127	0-127
正/负跳变	256	256	256	256	256
PID回路	0-7	0-7	0-7	0-7	0-7
端口	端口0	端口0	端口0	端口0, 1	端口0, 1

¹ LB60~LB63为STEP 7-Micro/WIN32的3.0版本或以后的版本软件保留

STL	页	STL	页	STL	页	STL	页	STL	页
=	73	AW > =	140	IBCD	99	MOVB	164	RLW	179
+D	140	AW <>	140	INCB	144	MOVD	164	ROUND	99
-D	140	BCDI	99	INCD	144	MOVR	164	RRB	179
*D	140	BIR	165	INCW	144	MOVW	164	RRD	179
/D	140	BITIM	196	INVB	161	MUL	142	RRW	179
+I	140	BIW	165	INVW	161	NEXT	169	RTA	103
-I	140	BMB	166	ITB	99	NETR	125	RTS	107
=I	73	BMD	166	ITD	99	NETW	125	S	73
*I	140	BMW	166	ITS	107	NOT	70	SCAT	184
/I	140	BTI	99	JMP	171	O	70	SCPY	184
+R	140	CALL	204	LBL	171	OB =	140	SCRE	172
-R	140	CEVNT	153	LD	70	OB > =	140	SCRT	172
*R	140	CFND	187	LDB <=	140	OB >	140	SEG	99
/R	140	CITIM	196	LDB =	140	OB <	140	SFND	187
A	70	COS	143	LDB >=	140	OB < =	140	SHRB	181
AB < =	140	CRET	204	LDB >	140	OB <>	140	SI	73
AB =	140	CRETI	153	LDB <	140	OD <	140	SIN	143
AB >	140	CSCRE	172	LDB <>	140	OD < =	140	SLB	179
AB<	140	CTD	113	LDD >=	140	OD =	140	SLD	179
AB > =	140	CTU	113	LDD <	140	OD >	140	SLEN	184
AB <>	140	CTUD	113	LDD <=	140	OD > =	140	SLW	179
AD <	140	DECB	144	LDD =	140	OD <>	140	SPA	139
AD < =	140	DECD	144	LDD >	140	OI	70	SQRT	143
AD =	140	DECO	112	LDD <>	140	OLD	75	SRB	179
AD >	140	DECW	144	LDI	70	ON	70	SRD	179
AD > =	140	DISI	153	LDN	70	ONI	70	SRW	179
AD <>	140	DIV	142	LDNI	70	OR=	140	SSCPY	186
AENO	75	DLED	178	LDR=	140	OR <	140	STD	110
AI	70	DTA	103	LDR <	140	OR <=	140	STI	110
ALD	75	DTCH	153	LDR <=	140	OR >	140	STOP	167
AN	70	DTI	99	LDR >	140	OR >=	140	STR	110
ANDB	162	DTR	99	LDR >=	140	OR <>	140	SWAP	183
ANDD	162	DTS	107	LDR <>	140	ORB	162	TAN	143
ANDW	162	ED	70	LDS	75	ORD	162	TODR	78
ANI	70	ENCO	112	LDS=	98	ORW	162	TODRX	78
AR=	140	END	167	LDS<>	98	OS=	98	TODW	78
AR <	140	ENI	153	LDW <=	140	OS<>	98	TODWX	78
AR<=	140	EU	70	LDW <	140	OW <	140	TOF	196
AR >	140	EXP	143	LDW =	140	OW < =	140	TON	196
AR>=	140	FIFO	190	LDW >	140	OW =	140	TONR	196
AR <>	140	FILL	192	LDW >=	140	OW >	140	TRUNC	99
AS=	98	FND <	193	LDW <>	140	OW > =	140	WDR	167
AS<>	98	FND <>	193	LIFO	190	OW <>	140	XMT	130
ATCH	153	FND =	193	LN	143	PID	145	XORB	162
ATH	103	FND >	193	LPP	75	PLS	133	XORD	162
ATT	189	FOR	169	LPS	75	R	73	XORW	162
AW <	140	GPA	139	LRD	75	RCV	130		
AW < =	140	HDEF	118	LSCR	172	RI	73		
AW=	140	HSC	118			RLB	179		
AW >	140	HTA	103			RLD	179		