



# Help & Manual<sup>®</sup> 3.0

## Standard print manual template

Siemens Energy & Automation, Inc.

获取更多资料

# Table of Contents

Foreword	1
<b>Part I 入门指南</b>	<b>4</b>
1 目录	4
2 第1章 有关STEP 7-Micro/WIN和S7-200 CPU的基本信息	6
2.1 步骤7-Micro/WIN Window元件	6
2.2 快捷键	7
2.3 如何使用联机帮助	11
2.4 如何定制STEP 7-Micro/WIN的外观	12
2.5 安装S7-200设备指南	12
2.6 安装和移除S7-200模块	13
2.7 接地和布线指南	15
3 第2章 编程概念	17
2.1 如何控制程序作业	17
2.2 编址概述	18
2.3 如何组织程序	19
2.4 项目元件及其工作原理	20
2.5 编辑器比较:LAD、FBD、STL	20
2.6 编程模式比较: SIMATIC、IEC	23
2.7 如何使用向导帮助您编程	24
4 第3章 如何输入阶梯逻辑程序	24
3.1 如何建立项目	24
3.2 阶梯逻辑元素及其工作原理	24
3.3 在LAD中建立简单、串行和并行网络的规则	25
3.4 如何在LAD中输入指令	26
3.5 如何在LAD中输入地址	29
3.6 如何在LAD中输入程序注解	32
3.7 如何在LAD程序中编辑元素	35
3.8 如何使用查找、替换和转入功能	38
3.9 程序编辑器如何在LAD中显示输入错误	40
3.10 如何在LAD中编译	40
3.11 如何保存作业	41
5 第4章 如何输入功能块图程序	41
4.1 如何建立项目	41
4.2 功能块图元素及其工作原理	42
4.3 在FBD中建立网络的规则	44
4.4 如何在FBD中输入指令	44
4.5 如何在FBD中输入地址	47
4.6 如何在FBD中输入程序注释	51
4.7 如何在FBD中编辑程序元素	54
4.8 如何使用查找、替换和转入功能	58
4.9 程序编辑器如何在FBD中显示输入错误	60
4.10 如何在FBD中编译	60
4.11 如何保存作业	60
6 第5章 如何输入语句列表程序	61
5.1 如何建立项目	61
5.2 如何在STL中输入语句	61

5.3	如何在STL中输入程序注解 .....	62
5.4	语句列表程序举例 .....	66
5.5	如何编辑STL程序 .....	67
5.6	如何使用查找、替换和转入功能 .....	69
5.7	程序编辑器如何在STL中显示输入错误 .....	70
5.8	如何在STL中编译 .....	71
5.9	如何保存作业 .....	71
<b>7</b>	<b>第6章 如何建立通讯和下载程序 .....</b>	<b>72</b>
6.1	通讯概述 .....	72
6.2	如何测试通讯网络 .....	72
6.3	如何下载程序 .....	73
6.4	如何纠正编译错误和下载错误 .....	73
<b>8</b>	<b>第7章 如何监控和调试程序 .....</b>	<b>74</b>
7.1	调试和监控特征概述 .....	74
7.2	如何在程序编辑器中显示状态 .....	83
7.3	如何在状态图中显示状态 .....	92
7.4	如何执行有限数目的扫描 .....	100
<b>9</b>	<b>第8章 如何管理项目 .....</b>	<b>101</b>
8.1	如何打印 .....	101
8.2	如何移动、复制、重新命名和邮寄项目 .....	102
8.3	如何从PLC上载 .....	102
8.4	如何复制程序段 .....	103
8.5	较早版本的STEP 7-Micro/WIN、Micro/DOS项目 .....	104
<b>Part II</b>	<b>应用程序用户参考手册 .....</b>	<b>108</b>
<b>1</b>	<b>窗口元件 .....</b>	<b>108</b>
<b>2</b>	<b>主菜单 .....</b>	<b>0</b>
<b>3</b>	<b>工具条 .....</b>	<b>0</b>
<b>4</b>	<b>快捷键 .....</b>	<b>112</b>
<b>5</b>	<b>编程向导 .....</b>	<b>116</b>
	HSC指令向导 .....	116
	NETR/NETW指令向导 .....	118
	PID指令向导 .....	120
	PID调谐控制板 .....	122
	TD 200 文字显示功能 .....	128
	TD 200向导 (TD 200 v2.1版或之前版本) .....	128
	TD 200向导 (TD 200 v3.0版或TD 200C v1.0版) .....	134
	EM241调制解调器模块性能 .....	144
	EM241调制解调器向导概述 .....	148
	EM253位置模块性能 .....	148
	EM253位置PTO/PWM向导概述 .....	149
	EM253控制板 .....	151
	CP243-1以太网模块性能 .....	154
	CP243-1以太网向导概述 .....	155
	CP243-1 IT互联网模块性能 .....	156
	CP243-1 IT互联网向导概述 .....	157
	CP243-2 AS-i模块性能 .....	162
	CP243-2 AS-i向导概述 .....	162
	菜谱向导 .....	163
	数据日志向导 .....	167

## Part III LAD、FBD和STL程序编辑器 172

<b>1 阶梯逻辑编程</b> .....	<b>172</b>
SIMATIC LAD指令 .....	172
IEC 1131-3 LAD指令 .....	172
功率流指示灯 .....	173
连接指令 .....	174
指令参数 .....	175
通用指令 .....	176
保留参数 .....	177
编辑指令参数 .....	178
使用符号表帮助组织程序 .....	182
为每个网络显示符号信息表 .....	189
SIMATIC和IEC 1131-3数据类型 .....	191
使用“查找”和“替换”进行修改 .....	0
理解S7-200计时器指令 .....	194
理解S7-200计数器指令 .....	199
理解高速计数器指令 .....	200
理解高速输出指令 .....	210
子例行程序编程 .....	224
中断例行程序编程 .....	229
局部变量表编程 .....	231
使用数据块在PLC V内存中存储数据 .....	235
编译项目 .....	239
检视PLC内存用法的交叉引用总结 .....	240
将程序下载至PLC .....	244
使用状态监控确认运行时间性能 .....	247
保存作业 .....	256
<b>2 功能块图编程</b> .....	<b>257</b>
SIMATIC FBD指令 .....	257
IEC 1131-3 FBD指令 .....	258
逻辑流指示灯 .....	259
连接指令 .....	259
指令参数 .....	261
通用指令 .....	262
保留参数 .....	263
编辑指令参数 .....	263
在AND（与）或OR（或）方框中增加输入行 .....	267
从AND（与）或OR（或）方框移除多余的输入行 .....	268
负（反转）逻辑连接 .....	268
为逻辑连接指定立即属性（直接I/O存取） .....	271
使用符号表帮助组织程序 .....	275
为每个网络显示符号信息表 .....	282
SIMATIC和IEC 1131-3数据类型 .....	284
使用“查找”和“替换”进行修改 .....	287
在程序中增加注解 .....	289
理解S7-200计时器指令 .....	293
理解S7-200计数器指令 .....	298
理解高速计数器指令 .....	299
理解高速输出指令 .....	309
子例行程序编程 .....	323
中断例行程序编程 .....	328
局部变量表编程 .....	330

使用数据块在PLC V内存中存储数据 .....	334
编译项目 .....	338
检视PLC内存用法的交叉引用总结 .....	339
将程序下载至PLC .....	343
使用状态监控确认运行时间性能 .....	346
保存作业 .....	355
<b>3 语句表编程 .....</b>	<b>356</b>
SIMATIC STL指令 .....	356
语句列表程序举例 .....	357
编辑指令参数 .....	358
使用符号表帮助组织程序 .....	359
SIMATIC和IEC 1131-3数据类型 .....	366
STL程序书签 .....	370
使用“查找”和“替换”进行修改 .....	370
为程序增加注解 .....	372
理解S7-200计时器指令 .....	376
理解S7-200计数器指令 .....	381
理解高速计数器指令 .....	382
理解高速输出指令 .....	392
子例行程序编程 .....	406
中断例行程序编程 .....	411
局部变量表编程 .....	413
使用数据块在PLC V内存中存储数据 .....	417
编译项目 .....	421
检视PLC内存用法的交叉引用总结 .....	422
将程序下载至PLC .....	426
使用状态监控确认运行时间性能 .....	429
保存作业 .....	438
<b>Part IV LAD、STL和FBD指令集 .....</b>	<b>442</b>
<b>1 ENO用法 .....</b>	<b>442</b>
<b>2 使用IEC超载和非标准指令 .....</b>	<b>442</b>
<b>3 受S7-200 CPU模型支持的指令 .....</b>	<b>442</b>
<b>4 SIMATIC / 国际助记符 .....</b>	<b>442</b>
<b>5 SIMATIC LAD指令 .....</b>	<b>443</b>
位逻辑 .....	443
时钟 .....	444
通讯 .....	444
比较 .....	444
转换 .....	446
计数器 .....	447
浮点数学 .....	447
整数数学 .....	448
中断 .....	449
逻辑操作 .....	449
移动 .....	450
程序控制 .....	450
移位 / 旋转 .....	451
字符串 .....	451
表 .....	452
计时器 .....	453

子例行程序 .....	453
<b>6 SIMATIC FBD指令 .....</b>	<b>455</b>
位逻辑 .....	455
时钟 .....	455
通讯 .....	455
比较 .....	456
转换 .....	457
计数器 .....	458
浮点数学 .....	459
整数数学 .....	459
中断 .....	460
逻辑操作 .....	460
移动 .....	461
程序控制 .....	461
移位 / 旋转 .....	462
字符串 .....	463
表 .....	464
计时器 .....	464
子例行程序 .....	464
<b>7 SIMATIC STL指令 .....</b>	<b>466</b>
位逻辑 .....	466
时钟 .....	468
通讯 .....	468
比较 .....	468
转换 .....	472
计数器 .....	473
浮点数学 .....	473
整数数学 .....	474
中断 .....	475
逻辑操作 .....	475
移动 .....	476
程序控制 .....	476
移位 / 旋转 .....	477
字符串 .....	478
表 .....	479
计时器 .....	479
子例行程序 .....	480
<b>8 IEC LAD指令 .....</b>	<b>482</b>
位逻辑 .....	482
时钟 .....	482
通讯 .....	483
比较 .....	483
转换 .....	484
计数器 .....	485
中断 .....	485
逻辑操作 .....	486
数学 .....	486
移动 .....	486
数字 .....	487
程序控制 .....	487
移位 / 旋转 .....	488
字符串 .....	488
表 .....	489

计时器 .....	490
子例行程序 .....	490
<b>9 IEC FBD指令 .....</b>	<b>492</b>
位逻辑 .....	492
时钟 .....	492
通讯 .....	493
比较 .....	493
转换 .....	494
计数器 .....	495
中断 .....	495
逻辑操作 .....	496
数学 .....	496
移动 .....	496
数字 .....	497
程序控制 .....	497
移位 / 旋转 .....	498
字符串 .....	498
表 .....	499
计时器 .....	499
子例行程序 .....	500
<b>Part V SM特殊内存赋值和功能 .....</b>	<b>504</b>
<b>1 SMB 0-29 (预定义S7-200只读内存) .....</b>	<b>504</b>
SMB0系统状态位 .....	504
SMB1指令执行状态位 .....	504
SMB2自由端口接收字符 .....	504
SMB3自由端口奇偶校验错误 .....	505
SMB4中断队列溢出、运行时间程序错误、中断启用、自由端口传输器闲置、数值强制 .....	505
SMB5 I/O错误状态位 .....	505
SMB6 CPU代码寄存器 .....	506
SMB8-SMB21 I/O模块代码和错误寄存器 .....	506
SMW22-SMW26扫描时间 .....	507
SMB28-SMB29模拟调整 .....	507
<b>2 SMB 30-194 (预定义S7-200读取 / 写入内存) .....</b>	<b>507</b>
SMB30和SMB130自由端口控制寄存器 .....	507
SMB31-SMW32永久内存 (EEPROM) 写入控制 .....	508
SMB34-SMB35定时中断的时间间隔寄存器 .....	508
SMB36-SMB65 HSC0、HSC1和HSC2高速计数寄存器 .....	509
SMB66-SMB85 PTO / PTM高速输出寄存器 .....	511
SMB86-SMB94和SMB186-SMB194接收讯息控制 .....	512
SMW98 I/O扩充总线 - 通讯错误 .....	513
SMB130-SMB165 HSC3、HSC4和HSC5高速计数寄存器 .....	514
用于PLC (脉冲) 指令的SMB166-SMB194 PTO概要表 .....	515
为由智能扩充模块提供的状态信息保留的SMB200-SMB549 .....	516
<b>Part VI PLC操作和选项 .....</b>	<b>518</b>
<b>1 停止和运行PLC操作模式 .....</b>	<b>518</b>
<b>2 在停止模式写入和强制输出 .....</b>	<b>519</b>
<b>3 控制程序元素 .....</b>	<b>520</b>
<b>4 系统块配置 .....</b>	<b>520</b>

5 如何使用状态工具确认程序逻辑 .....	521
<b>Part VII 设置通讯</b> .....	<b>532</b>
1 了解S7-200网络通讯的基本知识 .....	532
2 我的通讯选择是什么? .....	534
3 设置通讯 .....	537
4 仅使用S7-200设备的网络配置样本 .....	541
5 安装和移除通讯接口 .....	544
6 建立网络 .....	545
7 在网络中使用调制解调器和STEP 7-Micro/WIN .....	550
8 高级标题 .....	557
9 以太网通讯 .....	562
<b>Part VIII 如何使用PLC内存</b> .....	<b>568</b>
1 内存类型和属性 .....	568
2 直接和间接编址 .....	568
3 位存取 .....	570
4 字节、字或双字存取 .....	570
5 CPU内存地址范围 .....	571
6 SIMATIC和IEC 1131-3数据类型 .....	572
7 系统电源中断后的数据保留 .....	575
8 常量 .....	578
9 关键字 .....	580
10 ASCII (.awl) 文本文件的词法规则 .....	581
11 CPU内存中的指令大小 .....	583
<b>Part IX 如何执行常见任务</b> .....	<b>592</b>
1 设置工作区选项 .....	592
2 编辑程序 .....	593
STL (语句表) .....	593
LAD (阶梯逻辑图) .....	593
FBD (功能块图) .....	593
使用数据块在V内存中存储数据 .....	593
查找 / 替换 / 转入编辑操作 .....	597
使用交叉引用和元素用法 .....	597
成功编译 .....	601
3 使用全局符号和局部变量3 定义全局符号 .....	602
4 显示每个网络的符号信息表 .....	610
5 定义局部变量 .....	612
6 PLC设置 .....	615
读取PLC信息 .....	615
选择目标PLC类型 (型号和版本) .....	615



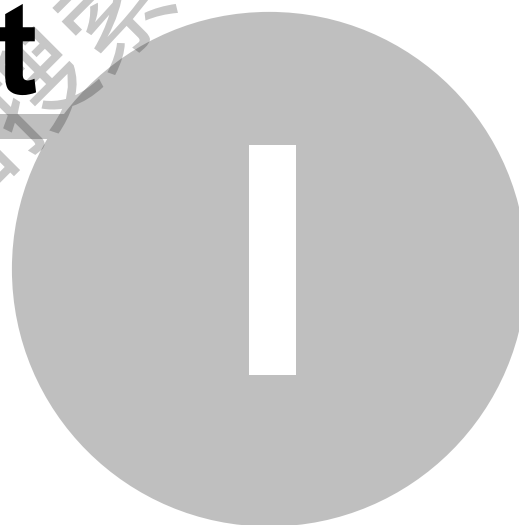
使用系统块配置PLC .....	617
设置当日时钟时间 .....	618
将PLC与PC内存中的打开项目进行比较 .....	618
为内存盒编程 .....	619
PLC电源开启重设 .....	620
<b>7 上载和下载程序 .....</b>	<b>620</b>
上载 .....	620
下载 .....	620
清除PLC内存 .....	623
解释错误讯息 .....	623
<b>8 使用向导 .....</b>	<b>624</b>
HSC指令向导 .....	624
NETR/NETW指令向导 .....	626
PID指令向导 .....	627
PID调谐控制板 .....	630
TD 200 文字显示功能 .....	636
TD 200向导 (TD 200 v2.1版或之前版本) .....	636
TD 200向导 (TD 200 v3.0版或TD 200C v1.0版) .....	642
EM241调制解调器模块性能 .....	652
EM241调制解调器模块和调制解调器向导概述 .....	656
EM253位置模块性能 .....	656
EM253位置和PTO/PWM向导概述 .....	657
EM253控制板 .....	659
CP243-1以太网模块性能 .....	662
CP243-1以太网向导概述 .....	663
CP243-1 IT互联网模块性能 .....	664
CP243-1 IT互联网向导概述 .....	665
CP243-2 AS-i模块性能 .....	670
CP243-2 AS-i向导概述 .....	670
菜谱向导 .....	671
数据日志向导 .....	675
<b>9 监控和调试程序 .....</b>	<b>678</b>
调试和监控特征概述 .....	678
停止和运行PLC操作模式 .....	688
执行一组固定数目的扫描 .....	690
如何存取读取、写入、强制和取消强制功能 .....	0
在运行模式中编辑程序 .....	691
使用状态图 .....	693
<b>10 打印 .....</b>	<b>700</b>
打印项目和文档 .....	700
<b>Part X 错误讯息 .....</b>	<b>708</b>
<b>1 STEP 7-Micro/WIN编译程序错误 .....</b>	<b>708</b>
<b>2 通讯错误 .....</b>	<b>709</b>
<b>3 LAD/FBD编辑程序错误 .....</b>	<b>710</b>
<b>4 项目文件I/O错误 .....</b>	<b>711</b>
<b>5 PLC严重错误 .....</b>	<b>711</b>
<b>6 PLC编译和运行时间错误 .....</b>	<b>712</b>
<b>7 PLC非严重错误内存位置 .....</b>	<b>714</b>

8	时间戳记不符 .....	714
9	EM241调制解调器模块错误代码 .....	715
10	EM253位置模块错误代码 .....	716
11	以太网 / 互联网模块和ETHx_CTRL指令的CP243-1错误代码 .....	717
12	信道和ETHx-XFR/ETHx_CFG的CP243-1错误代码 .....	718
13	ETHx_EMAIL指令的CP243-1 IT错误代码 .....	720
14	ETHx_FTPC指令的CP243-1 IT错误代码 .....	721
15	CP243-2 AS-i模块错误代码 .....	722
<b>Part XI</b>	<b>库</b> .....	<b>726</b>
1	一般库概述 .....	726
2	用户定义库概述 .....	726
3	用户定义库辅导 .....	729
4	USS协议 .....	733
	使用USS协议库 .....	733
	使用USS协议指令 .....	735
	USS_CTRL USS协议 .....	735
	USS_INIT USS协议 .....	739
	USS_RPM_x USS协议 .....	740
	USS_WPM_x USS协议 .....	742
	USS协议程序举例 .....	744
	USS指令执行错误 .....	748
5	Modbus协议库 .....	749
	使用Modbus协议库 .....	749
	使用Modbus从属协议指令 .....	750
	MBUS_INIT指令 .....	751
	MBUS_Slave指令 .....	753
	Modbus从属协议执行错误代码 .....	755
<b>Part XII</b>	<b>快速参考手册</b> .....	<b>758</b>
	<b>Index</b> .....	<b>763</b>

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part



获取更多资料 微信搜索 蓝领星球

# 1 入门指南

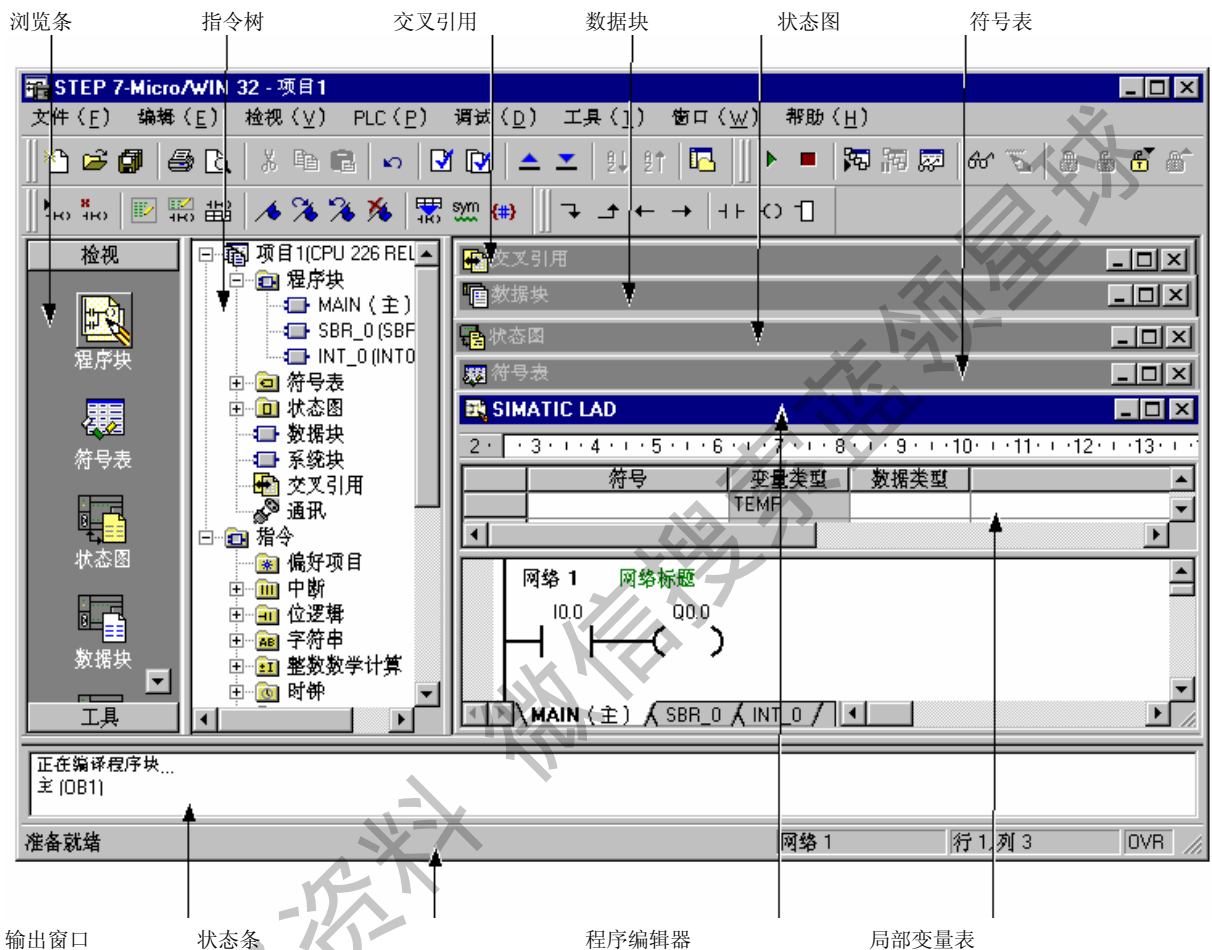
## 1.1 目录

<b>第1章</b>	<b>欢迎使用Micro/WIN</b>
<a href="#">1.1</a>	步骤7—Micro/WIN窗口元件
<a href="#">1.2</a>	捷径键
<a href="#">1.3</a>	如何使用联机帮助
<a href="#">1.4</a>	如何定制STEP 7-Micro/WIN的外观
<a href="#">1.5</a>	S7-200设备安装指南
<a href="#">1.6</a>	安装和移除S7-200模块
<a href="#">1.7</a>	接地和布线指南
<b>第2章</b>	<b>编程概念</b>
<a href="#">2.1</a>	如何控制程序作业
<a href="#">2.2</a>	编址概述
<a href="#">2.3</a>	如何组织程序
<a href="#">2.4</a>	项目元件及工作原理
<a href="#">2.5</a>	编辑器比较：LAD、FBD、STL
<a href="#">2.6</a>	编程模式比较：SIMATIC、IEC
<a href="#">2.7</a>	如何使用向导帮助编程
<b>第3章</b>	<b>如何输入梯形逻辑程序</b>
<a href="#">3.1</a>	如何建立项目
<a href="#">3.2</a>	梯形逻辑元素及其工作原理
<a href="#">3.3</a>	在LAD中建立简单、串行和并行网络
<a href="#">3.4</a>	如何在LAD中输入指令
<a href="#">3.5</a>	如何在LAD中输入地址
<a href="#">3.6</a>	如何在LAD中输入程序注解
<a href="#">3.7</a>	如何在LAD中编辑程序元素
<a href="#">3.8</a>	如何使用“查找”、“替换”和“转入”功能
<a href="#">3.9</a>	程序编辑器如何在LAD中显示输入错误
<a href="#">3.10</a>	如何在LAD中编译
<a href="#">3.11</a>	如何保存作业
<b>第4章</b>	<b>如何输入功能快图程序</b>
<a href="#">4.1</a>	如何建立项目
<a href="#">4.2</a>	功能快图元素及其工作原理
<a href="#">4.3</a>	在FBD中建立网络的规则
<a href="#">4.4</a>	如何在FBD中输入指令
<a href="#">4.5</a>	如何在FBD中输入地址
<a href="#">4.6</a>	如何在FBD中输入程序注解
<a href="#">4.7</a>	如何在FBD中编辑程序元素
<a href="#">4.8</a>	如何使用“查找”、“替换”和“转入”功能
<a href="#">4.9</a>	程序编辑器如何在FBD中显示输入错误
<a href="#">4.10</a>	如何在FBD中编译
<a href="#">4.11</a>	如何保存作业
<b>第5章</b>	<b>如何输入语句列表程序</b>
<a href="#">5.1</a>	如何建立项目
<a href="#">5.2</a>	如何在STL中输入语句

- [5.3](#) 如何在STL中输入程序注解
- [5.4](#) 语句列表程序举例
- [5.5](#) 如何编辑STL程序
- [5.6](#) 如何使用“查找”、“替换”和“转入”功能
- [5.7](#) 程序编辑器如何在STL中显示输入错误
- [5.8](#) 如何在STL中编译
- [5.9](#) 如何保存作业
- 第6章 如何建立通讯和下载程序**
- [6.1](#) 通讯概述
- [6.2](#) 如何测试通讯网络
- [6.3](#) 如何下载程序
- [6.4](#) 如何纠正编译错误和下载错误
- 第7章 如何监管和调试程序**
- [7.1](#) 调试和监管功能概述
- [7.2](#) 如何在程序编辑器中显示状态
- [7.3](#) 如何在状态图中显示状态
- [7.4](#) 如何执行有限数目的扫描
- 第8章 如何管理项目**
- [8.1](#) 如何打印
- [8.2](#) 如何移动、复制、重新命名和邮寄项目
- [8.3](#) 如何从PLC上载
- [8.4](#) 如何复制项目段
- [8.5](#) STEP 7-Micro/WIN的较早版本，Micro/DOS项目

## 1.2 第1章 有关STEP 7-Micro/WIN和S7-200 CPU的基本信息

### 1.2.1 1.1 步骤7—Micro/WIN Window元件



**浏览条** 显示编程特性的按钮控制群组：

“视图” — 选择该类别，为程序块、符号表，状态图，数据块，系统块，交叉引用及通讯显示按钮控制。

“工具” — 选择该类别，显示指令向导、TD200向导、位置控制向导、EM 253控制面板和调制解调器扩充向导的按钮控制。

**注释：** 当浏览条包含的对象因为当前窗口大小无法显示时，浏览条显示滚动按钮，使您能向上或向下移动至其他对象。

**指令树** 提供所有项目对象和为当前程序编辑器（LAD、FBD或STL）提供的所有指令的树型视图。您可以用鼠标右键单击树中“**程序块**” — 选择该类别，为程序块、符号表，状态图，数据块，系统块，交叉引用及通讯显示按钮控制。

**交叉引用** 允许您检视程序的交叉引用和组件使用信息。

**数据块** 允许您显示和编辑数据块内容。

**状态图窗口** 允许您将程序输入、输出或变量置入图表中，以便追踪其状态。您可以建立多个状态图，以便从程序的不同部分检视组件。每个状态图在状态图窗口中有自己的标记。

**符号表/全局变量表窗口** 允许您分配和编辑全局符号（即可在任何**POU**中使用的符号值，不只是建立符号的**POU**）。您可以建立多个符号表。可在项目中增加一个S7-200系统符号预定义表。

**输出窗口** 在您编译程序时提供讯息。当输出窗口列出程序错误时，可双击错误讯息，会在程序编辑器窗口中显示适当的网络。当您编译程序或指令库时，提供讯息。当输出窗口列出程序错误时，您可以双击错误讯息，会在程序编辑器窗口中显示适当的网络。

**状态条**提供您在STEP 7-Micro/WIN中操作时的操作状态信息。

**程序编辑器窗口**包含用于该项目的编辑器（**LAD**、**FBD**或**STL**）的局部变量表和程序视图。如果需要，您可以拖动分割条，扩展编辑区域。除了（**OB1**）之外，建立**子例行程序**或**中断例行程序**时，标记出现在程序编辑器窗口中。

**局部变量表**包含您对局部变量所作的赋值（即子例行程序和中断例行程序使用的变量）。在局部变量表中建立的变量使用暂时内存；地址赋值由系统处理；变量的使用仅限于建立此变量的**POU**。

**菜单条**允许您使用滑鼠或键击执行操作。您可以**定制**“工具”菜单，在该菜单中增加自己的工具。

**工具条**为最常用的STEP 7-Micro/WIN操作提供便利的滑鼠存取。您可以**定制**每个工具条的内容和外观。

## 1.2.2 1.2 快捷键

您可以在STEP 7-Micro/WIN中使用快捷键，更有效地完成多项任务。

[LAD](#)  
[FBD](#)  
[STL](#)  
[符号表 / 全局变量表](#)  
[局部变量表](#)  
[状态图](#)  
[程序状态](#)  
[在绝对编址模式和符号编址模式之间变换](#)



**提示：**如果列出多个键，所有的键均须一起按下，快捷键才会有效。

### LAD和FBD快捷键

#### 返回顶部

#### 1. 浏览键

##### 键排列顺序

##### 措施

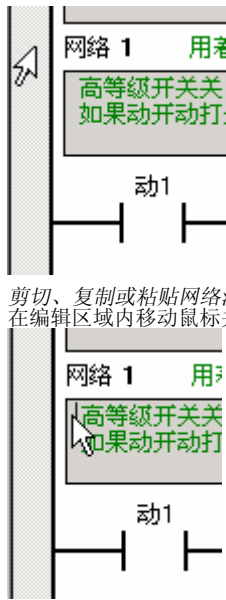
HOME	将光标移至同行的第一列
END	将光标移至同行的最后一列
PAGE UP	垂直向上移动一个屏幕
PAGE DOWN	垂直向下移动一个屏幕
LEFT ARROW	将光标向左移动一个单元格
RIGHT ARROW	将光标向右移动一个单元格
UP ARROW	将光标向上移动一个单元格
DOWN ARROW	将光标向下移动一个单元格
CTRL + HOME	将光标移至第一个网络的第一个单元格
CTRL + END	将光标移至最后一个网络的最后一个单元格
F1	当前单元格指令的帮助
CTRL + PAGE UP	显示下一个POU，向左移动浏览POU标记
CTRL + PAGE DOWN	显示下一个POU，向右移动浏览POU标记
TAB	选择带布尔（位值）参数指令时，TAB键移至该指令中的下一个布尔参数（仅限FBD）当您选择一个带布尔参数指令时，TAB键将选项移至该指令中的下一个布尔参数（仅限FBD）

#### 2. 编辑

##### 鼠标拖过或快捷键网络选择

剪切、复制或粘贴网络：

将鼠标移动到网络区域左边的装订线上，然后单击以设置网络上上下文。



剪切、复制或粘贴网络注解：  
在编辑区域内移动鼠标并单击，以设置注解上下文。

#### 键排列顺序

SHIFT + 上箭头  
SHIFT + 下箭头  
SHIFT + PAGE UP  
SHIFT + PAGE DOWN  
CONTROL + SHIFT + HOME  
CONTROL + SHIFT + END  
CONTROL + SHIFT + END

#### 措施

向上扩展选择  
向下扩展选择  
向上扩展网络选择  
向下扩展网络选择  
扩展选择至第一个网络  
扩展选择至最后一个网络  
选择全部网络

CTRL + X SHIFT + DELETE 选择网络系列时，剪切系列；当光标位于网络标题上时，剪切整个网络；当光标位于单元格上时，剪切单元格内容  
CTRL + C CTRL + INSERT 选择网络系列时，复制系列；当光标位于网络标题上时，复制整个网络；当光标位于单元格上时，复制单元格内容  
CTRL + V SHIFT + INSERT 选择网络系列时，粘贴系列；当光标位于网络标题上时，粘贴网络；当光标位于单元格上时，粘贴单元格。  
在光标当前位置的前方 / 上方粘贴所选内容

SPACEBAR 编辑当前单元格的助记符或操作数  
ENTER 编辑当前单元格的助记符或操作数  
CTRL + LEFT ARROW 放下一条水平线，将光标向左移动一个单元格  
CTRL + RIGHT ARROW 放下一条水平线，将光标向右移动一个单元格  
CTRL + UP ARROW 放下一条垂直线，将光标向上移动一个单元格  
CTRL + DOWN ARROW 放下一条垂直线，将光标向下移动一个单元格  
DELETE 删除当前单元格、网络或网络选项  
BACKSPACE 将光标向左移动一个单元格，删除当前单元格  
SHIFT + LEFT ARROW 选择当前网络，将光标向左移动一个单元格  
SHIFT + RIGHT ARROW 选择当前网络，将光标向右移动一个单元格  
SHIFT + HOME 选择当前网络，将光标置于当前行的第一个单元格处  
SHIFT + END 选择当前网络，将光标置于当前行的最后一个单元格处  
SHIFT + DOWN ARROW 从当前网络向下选择  
SHIFT + UP ARROW 从当前网络向上选择  
SHIFT + PAGE UP 从当前网络向上选择相邻网络的一个页面  
SHIFT + PAGE DOWN 从当前网络向下选择相邻网络的一个页面  
CTRL + SHIFT + HOME 从光标至POU顶部选择全部相邻网络



CTRL + SHIFT + END	从光标至POU底部选择全部相邻网络
CTRL + A	选择POU中的全部网络
F11	将输入更改为求反输入 / 切换复原 (仅限FBD、布尔值)
CTRL + F11	将输入或输出更改为立即类型 / 切换复原 (仅限FBD、布尔值)
PLUS SIGN	增加一个附加输入参数 (仅限FDB、AND/OR (与 / 或) 指令)
MINUS SIGN	删除一个附加输入参数 (仅限FDB、AND/OR (与 / 或) 指令)
CTRL + Y	在符号和绝对编址模式之间切换
CTRL + T	显示每个网络的符号信息表

### 3. 放下指令

键排列顺序	措施
F4	包含所有接点助记符类型的列表框 (仅限LAD)
F4	放下AND (与) 框 (仅限FBD)
F6	包含所有线圈助记符类型的列表框 (仅限LAD)
F6	放下OR (或) 框 (仅限FBD)
F9	包含所有方框助记符类型的列表框

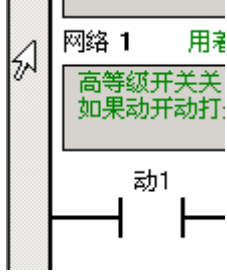
### STL快捷键

#### 返回顶端

#### 鼠标拖过或快捷键网络选择

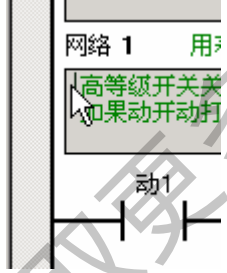
剪切、复制或粘贴网络:

将鼠标移动到网络区域左边的装订线上, 然后单击以设置网络上上下文。



剪切、复制或粘贴网络注解:

在编辑区域内移动鼠标并单击, 以设置注解上下文。



#### 键排列顺序

键排列顺序	措施
SHIFT + 上箭头	向上扩展选择
SHIFT + 下箭头	向下扩展选择
SHIFT + PAGE UP	向上扩展网络选择
SHIFT + PAGE DOWN	向下扩展网络选择
CONTROL + SHIFT + HOME	扩展选择至第一个网络
CONTROL + SHIFT + END	扩展选择至最后一个网络
CONTROL + SHIFT + END	选择全部网络
CTRL + A	全选
CTRL + C	复制选项
CTRL + F	查找字符串
CTRL + H	替换字符串

CTRL + V	粘贴选项
CTRL + X	剪切选项
CTRL + Z	撤销
DELETE	删除选项
SHIFT + DELETE	将选项剪切至剪贴板
F2	移至下一个书签的位置
CTRL + F2	在光标的当前位置设为书签
CTRL + F3	插入网络
CTRL + F4	删除网络
F3	查找下一个字符串
BACKSPACE	退格
SHIFT + BACKSPACE	退格
ALT + BACKSPACE	撤销
INSERT	切换盖写
CTRL + INSERT	复制选项
SHIFT + INSERT	粘贴选项
HOME	移至当前行开始
CTRL + HOME	移至文件顶部
SHIFT + HOME	从光标当前位置选择至同一行的开始
CTRL + SHIFT + HOME	从光标位置选择至文件顶部
END	移至当前行结尾
CTRL + END	移至文件底部
SHIFT + END	从光标当前位置选择至同一行的结尾
CTRL + SHIFT + END	从光标位置选择至文件底部
LEFT ARROW	左移光标, 每次移动一个字符
CTRL + LEFT ARROW	左移光标, 每次移动一个字
SHIFT + LEFT ARROW	向左选择, 每次选择一个字符
CTRL + SHIFT + LEFT ARROW	向左选择, 每次选择一个字
RIGHT ARROW	右移光标, 每次移动一个字符
CTRL + RIGHT ARROW	右移光标, 每次移动一个字
SHIFT + RIGHT ARROW	向右选择, 每次选择一个字符
CTRL + SHIFT + RIGHT ARROW	向右选择, 每次选择一个字
UP ARROW	将光标向上移动, 一次移动一行
SHIFT + UP ARROW	向上选择, 一次选择一行
DOWN ARROW	光标下移, 一次移动一行
SHIFT + DOWN ARROW	向下选择, 一次选择一行
PAGE DOWN	光标下移一个屏幕的数据
SHIFT + PAGE DOWN	从当前光标位置向下选择一个屏幕的数据
PAGEUP	光标上移一个屏幕的数据
SHIFT + PAGE UP	从当前光标位置向上选择一个屏幕的数据
TAB	向前跳格
SHIFT + TAB	向后跳格
CTRL + Y	在符号和绝对编址模式之间切换
CTRL + PAGE UP	显示下一个POU, 向左移动浏览POU标记
CTRL + PAGE DOWN	显示下一个POU, 向右移动浏览POU标记

[符号表 / 全局变量表、  
局部变量表、状态图](#)  
[返回顶端](#)

**键排列顺序**

CTRL + X  
CTRL + C

**措施**

剪切选择的文本  
复制选择的文本

CTRL + V	粘贴选择的文本
DOWN ARROW	光标下移（在表尾处增加新行）
UP ARROW	光标上移
LEFT ARROW	光标左移
RIGHT ARROW	光标右移
DELETE	删除选择的文本
F1	在当前窗口中提供帮助
F2	允许您编辑当前的单元格
CTRL + A	选择所有的单元格
TAB	光标从左至右、从上至下移动
SHIFT + TAB	光标从右至左、从下至上移动
ENTER	校验单元格并右移光标
HOME	将光标移至第一列
END	将光标移至最后一列
CTRL + PAGE UP	显示下一个表 / 图。向左移动浏览符号表 / 状态图标记
CTRL + PAGE DOWN	显示下一个表 / 图。向右移动浏览符号表 / 状态图标记
SHIFT + ARROWS	选择文本
CTRL + I	在当前行上方插入新行
CTRL + SHIFT + I	在当前行下方插入新行

#### 程序状态快捷键

##### 返回顶端

F1	允许您不使用滑鼠存取菜单项目
----	----------------

#### 在绝对和符号编址模式之间变换

##### 返回顶端

#### 键排列顺序

#### 措施

CTRL + Y	在符号和绝对编址模式之间切换
CTRL + T	显示LAD或FBD程序每个网络的符号信息表

**注释：**对于LAD/FBD程序，您可以选择菜单命令**工具 (Tools) > 选项 (Options)**，并单击“程序编辑”标记，仅检视符号编址，或在启用符号编址模式时，检视符号和绝对地址。

## 1.2.3 1.3 如何使用联机帮助

#### 上下文相关的帮助

对于您希望获得帮助的标题，选择菜单项目或打开对话框，按“F1”键存取该标题的上下文相关帮助。（在某些情形下，可按“Shift”和“F1”键存取帮助标题。）

#### 从菜单获得帮助

STEP 7-Micro/WIN中的“帮助”菜单提供下列选项：

- **目录和索引** 允许您借助目录浏览程序（显示每本书包含的标题）或可搜索索引浏览该帮助系统。
- **这是什么？** 提供接口元素定义。通过同时按Shift和F1键，您还能存取“这是什么？”帮助。光标变为一个问号；用它  
误 M 辣冒镗 南牟可系七鳌
- **网络上的S7-200**为技术支持和产品信息提供西门子（Siemens）互联网网站存取。
- **关于**列出STEP 7-Micro/WIN的产品和版权信息。

#### 目录浏览程序

从STEP 7-Micro/WIN，从菜单条选择**帮助 > 目录和索引**，调出“目录”浏览程序。

从帮助标题内，单击“帮助标题”按钮，调出“目录”浏览程序。

**索引**

从按字母顺序排列的标题关键字列表，选择“目录”浏览程序中的“索引”标记。

**如何打印帮助**

可从“目录”浏览程序或单独的标题内打印标题：

- 欲打印书中的所有标题，从“目录”浏览程序选择书标题，并使用位于浏览程序底部的“打印”按钮。
- 欲从“目录”浏览程序打印个别标题，选择标题，并使用位于浏览程序底部的“打印”按钮。
- 欲从单个标题内打印，使用位于标题上方的“打印”按钮。

**注释：**《入门指南》“帮助”一节的标题按顺序编号，类似于手册各节的编排，便于您使用打印出的材料。

**1.2.4 1.4 如何定制STEP 7-Micro/WIN的外观**

STEP 7-Micro/WIN提供多种存取和显示信息的方法。

为了简化程序设计，您可能希望不用浏览条和输出窗口。您可以将在程序设计时需要的窗口盖住或最小化，例如局部变量表和符号表，仅在必要时调出。这样可为以下主要项目腾出最大的空间：指令树（供LAD和FBD程序员使用）和程序编辑器窗口（供STL、LAD和FBD程序员使用）。

**注释：**编译时输出窗口自动恢复。

以下是一些安排STEP 7-Micro/WIN工作区不同组件的提示：

- **检视或隐藏各种窗口组件**  
从菜单条选择“检视”，并选择一个对象，将其标示符号在打开和关闭之间切换。带标示符号的对象是当前在STEP 7-Micro/WIN环境中打开的对象。
- **级联窗口** 从菜单条选择窗口 > 级联、窗口 > 垂直或窗口 > 水平。
- **最小化、恢复、最大化或关闭窗口**  
使用位于每个窗口标题条中的最小化、恢复、最大化和关闭按钮。请注意，当您最大化窗口时，按钮在STEP 7-Micro/WIN主窗口按钮下方的菜单条区内显示。当您最大化窗口时，窗口会盖住您已经打开的任何其他窗口显示，但最靠前的窗口仍保持可见。
- **使用标记检视窗口的不同组件**  
诸如程序编辑器、状态图、符号表和数据块的窗口可能有多个标记。例如，程序编辑器窗口包含的标记允许您在主程序（OB1）、子例行程序和中断例行程序之间浏览。

**注释：**当您显示几个窗口时，可能没有足够的空间显示每个窗口包含的所有标记。您可以用标记旁的一对滚动箭头，从第一个标记浏览到最后标记。（标记旁的一对滚动箭头的功能与可能提供的允许您沿水平方向滚动窗口内容的滚动箭头不同。）

- **更改尺寸或移除局部变量表**  
将光标放置在程序编辑器和局部变量表的分隔条上方，拖动光标，增加或缩小局部变量表的尺寸。如果程序不包含要求您定义任何局部变量的子例行程序或中断例行程序，则拖动程序编辑器，使之完全盖住局部变量表。（因为局部变量表是程序编辑器窗口的一部分，您无法取消局部变量表。）
- **移动或隐藏工具条** 根据默认值，文件、调试和程序工具条在STEP 7-Micro/WIN的菜单条下方显示。然而，您可以移动任何工具条，将光标放在工具条区域内，移动工具条。如果您将工具条拖至STEP 7-Micro/WIN中任何窗口的边框附近，工具条将停放在该窗口的边框处，否则工具条成为一个独立的、自由漂浮的工具条。当工具条独立时，单击工具条标题条中的“X”按钮，隐藏工具条。您可以选择**工具 > 定制**，并从“定制”对话框“工具条”标记选择适当的复选框（文件、调试、阶梯、FBD、STL）恢复工具条。

**另请参阅：**

[如何定制工具条](#)  
[选择颜色首选项](#)  
[定制菜单和增加工具](#)

**1.2.5 1.5 安装S7-200设备指南****S7-**

200装置的设计便于安装。您可以使用安装孔将模块安装在面板上，您也可以使用内装夹片将模块安装在标准（DIN）横杆上。S7-200体积小，允许您有效地利用空间。

您可以将S7-200安装在面板上或安装在标准横杆上，S7-200可以横放，也可以竖放。

**请勿让S7-200设备接触热、高压和电气噪音**

作为放置系统设备的一般规则，始终将生成高压和高电气噪音的设备与生成低压、逻辑型设备（例如S7-200）的装置分开放置。

在面板中配置S7-

200布局时，应当考虑生成热的设备，将电子型设备放置在机箱中温度较低的位置。在高温环境中操作任何电子设备都会增加发生故障的频率。

另请考虑面板中布线的路径。避免将低压信号线和通讯电缆与交流电线和高能快速切换直流电线放在同一个盒内。

### 提供适当的冷却和布线空间

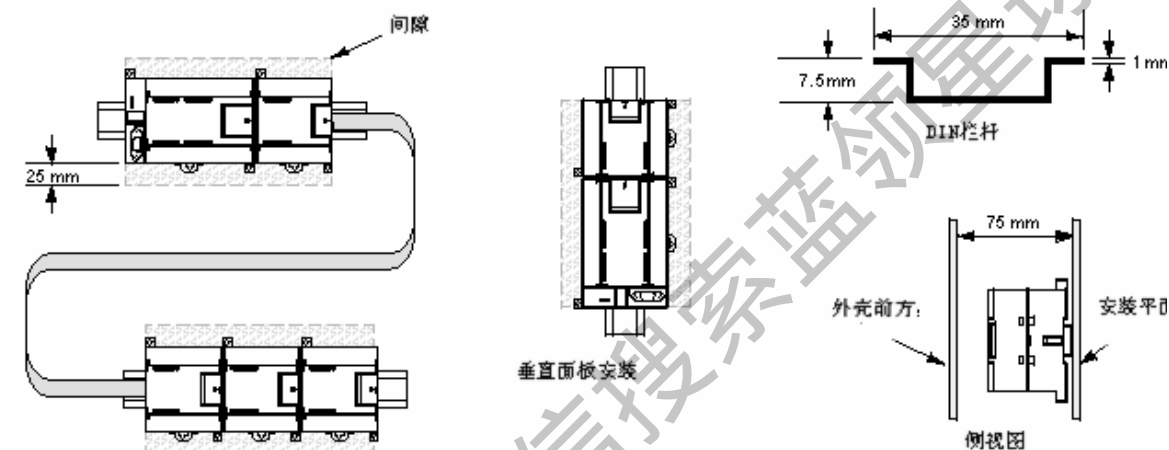
S7-

200设备的设计适用于自然对流冷却。为了达到适当的冷却目的，您必须在设备的上方和下方至少留出 25毫米的空间。此外，至少留出75毫米的深度。



**提示：** 对于纵向安装，允许的最大环境温度会降低摄氏 105度。将S7-200 CPU安装在任何扩充模块的下方。

在规划S7-200系统的布局时，请为布线和通讯电缆连接留出足够的空间。为了获得配置 S7-200系统布局的更大的灵活性，请使用 I/O扩充电缆。



带供选用扩充电缆的水平DIN栏杆安装（一个系统仅限一个）

### 电源预算

所有的S7-200 CPU都有一个为CPU、扩充模块和其他24 VDC用户电源要求提供电源的内装电源。

S7-200 CPU提供系统中任何扩充所需的5 VDC逻辑电源。请特别注意系统配置，确保CPU可提供所选扩充模块要求的5V电源。如果您的配置要求的电源超出CPU提供的电源范围，则必须移除一个模块，或选择一个功能更大的CPU。请参阅《S7-200可编程控制器系统手册》中的附录A，查阅有关S7-200 CPU提供的5 VDC逻辑预算以及扩充模块的5 VDC电源要求。请将《S7-200可编程控制器系统手册》附录B用作指南，确定CPU可为您的配置提供的功率（或电流）。

所有的S7-200 CPU还提供一个24 VDC传感器电源，可为输入点、扩充模块上的中继线圈电源或其他要求提供 24 VDC。如果您的电源要求超过传感器电源的预算，您必须在系统中增加一个外接 24 VDC电源。请参阅《S7-200可编程控制器系统手册》附录A，了解有关您的特定S7-200 CPU的24 VDC传感器电源预算的信息。

如果您需要安装一个外接24 VDC电源，核实电源不得与S7-200 CPU传感器电源并连。欲获得进一步电气噪声保护，建议将不同电源的公共导线（M）连接在一起。



**警告：** 将外接24 VDC电源与S7-200 24 VDC传感器电源并连会导致两种电源之间的冲突，因为每种电源都寻求建立自身需要的输出电压水平。此一冲突的结果会缩短一种或两种电源的使用寿命或立即发生故障，并随之导致 PLC系统不可预测的操作状况。不可预测的操作状况会导致死亡或严重人身伤害和 / 或装置和财产 损坏。S7-200 DC传感器电源和任何外接电源应当为不同的点提供电源。

## 1.2.6 1.6 安装和移除S7-200模块

S7-200可方便地安装在标准DIN横杆上或安装在面板上。

### 先决条件

在安装或移除任何电气设备之前，核实该装置的电源已经关闭。此外，核实任何相关装置的电源已经关闭。



**警告：** 在接通电源的情况下尝试安装或移除 S7-200或相关装置会造成电击或装置操作发生故障。在安装或移除程序中禁用S7-200及相关装置的所有电源会导致死亡或严重人身伤害和 /

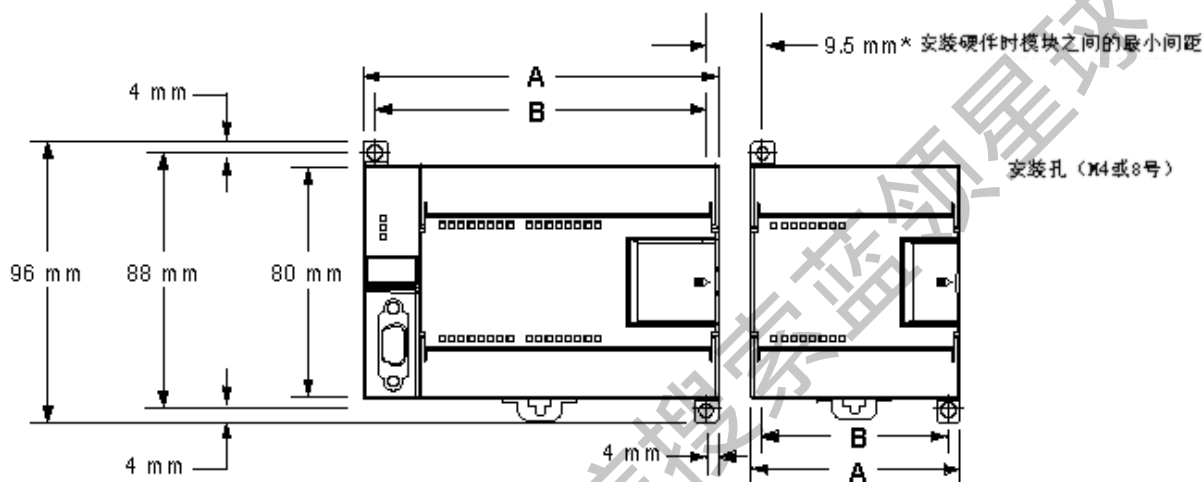
或装置损坏。始终遵循适当的安全措施，并确保在尝试安装或移除 S7-200 CPU 或相关装置之前已经禁用 S7-200 的电源。  
每当您更换或安装 S7-200 设备时，始终确保使用正确的模块或等同设备。



**警告：** 如果安装了不正确的模块，S7-200 中的程序会以不可预测的方式作业。  
不使用相同的型号、沿相同的方向或顺序替换 S7-200 设备会导致死亡或严重人身伤害和 / 或装置损坏。用相同的型号替换 S7-200 设备，确保方向和位置正确。

#### 安装尺寸

S7-200 CPU 和扩充模块包括用于在面板上安装的安装孔。请参阅下表中的安装尺寸。



S7-200 模块	宽度 A	宽度 B
CPU 221 和 CPU 222	90 毫米	82 毫米
CPU 224	120.5 毫米	112.5 毫米
CPU 226 和 CPU 226XM	196 毫米	188 毫米
扩充模块：8 针直流电和中继 I/O (8I/8Q 和 4I/4Q)	46 毫米	38 毫米
扩充模块：16 针数字 I/O (8I/8Q)、模拟 I/O (4AI、4AI/1AQ、2AQ)、RTD、热电偶、PROFIBUS、AS 接口、8 针交流电 (8I 和 8Q)、位置 and 调制解调器	71.2 毫米	63.2 毫米
扩充模块：32 点数字 I/O (16I/16Q)	137.3 毫米	129.3 毫米

#### 安装 CPU 或扩充模块

安装 S7-200 十分简便。只需遵循以下步骤。

##### 面板安装

1. 使用上表中的尺寸，定位、钻孔和轻敲安装孔（M4 或美国标准 8 号）。
2. 用适当的螺丝将模块固定在面板上。
3. 如果您使用的是扩充模块，将扩充模块带状电缆与存取门下方的扩充端口接头连接。

##### DIN 横杆安装

1. 每隔 75 毫米将横杆固定在安装面板上。
2. 咔嚓一声打开（位于模块底部的）DIN 夹片，并将模块背面扣在 DIN 横杆上。
3. 如果您使用的是扩充模块，将扩充模块带状电缆与存取门下方的扩充端口接头连接。
4. 将模块向下旋转至 DIN 横杆位置，咔嚓一声将夹片关闭。请仔细检查夹片是否夹紧，模块是否安全地固定在横杆上。为了避免损坏模块，按安装孔的标记，而不要直接按模块前侧。



**提示** 如果 S7-200 位于的环境带有高振荡电位或者 S7-200 沿垂直方向安装，使用 DIN 横杆阻块可能会有帮助。  
如果系统位于高振荡环境，则面板安装 S7-200 会提供较高的振荡保护水平。

#### 取出 CPU 或扩充模块

欲取出 S7-200 CPU 或扩充模块，请遵循以下步骤：

1. 断开 S7-200 电源。

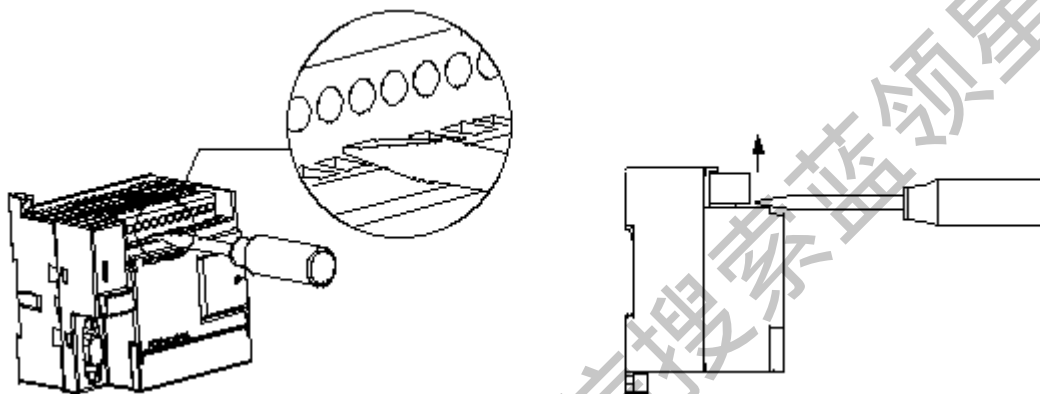
2. 断开所有与模块连接的线路和电缆。大多数 S7-200 CPU 和扩充模块配有可移动接头，便于完成此项工作。
3. 如果扩充模块与您正在移除的装置连接，打开存取盖门，并从相邻的模块上断开扩充模块带状电缆连接。
4. 松开安装螺丝或咔嚓一声松开 DIN 夹片。
5. 取出模块。

#### 取出和重新安装终端块接头

大多数 S7-200 模块配有可移动接头，便于安装和替换模块。请参阅《S7-200 可编程控制器系统手册》附录 A，确定您的 S7-200 模块是否配有可移动接头。您可以为未配备可移动接头的模块订购供选用扇出接头。订购号码请参阅《S7-200 可编程控制器系统手册》附录 E。

欲取出接头

1. 打开接头门，可接触接头。
2. 在接头中部的槽口中插入一把小螺丝刀。
3. 将螺丝刀朝与 S7-200 外壳相反的方向旋转，取出终端接头。请参阅下图。



欲重新安装接头

1. 打开接头门。
2. 将接头与装置上的针对齐，并将接头底座轮圈中的接头布线边缘对齐。
3. 用力向下按，旋转接头，直至咔嚓一声就位。请仔细检查，核实接头已适当对齐，并完全就位。

## 1.2.7 1.7 接地和布线指南

所有电气装置适当接地和布线对于帮助确保系统的最佳操作以及为应用程序和 S7-200 提供附加电气噪声保护十分重要。

#### 先决条件

在您为任何电气设备接地或安装布线之前，核实该装置的电源已经关闭。此外，核实通往任何相关装置的电源已经关闭。

在为 S7-

200 和相关装置布线时，请务必遵循所有适用的电气准则。根据所有适用的国家和地方标准安装和操作所有的装置。请与您所在地的管理部门联系，确定适用于您的具体情形的准则和标准。



**警告** 在接通电源的情况下尝试安装或布线 S7-200 或相关装置会造成电击或装置操作发生故障。

在安装或移除程序中禁用 S7-200 及相关装置的所有电源会导致死亡或严重人身伤害和 / 或装置损坏。始终遵循适当的安全措施，并确保在尝试安装或移除 S7-200 CPU 或相关装置之前已经禁用 S7-200 的电源。

在您设计 S7-200 系统的接地和布线时，始终考虑安全因素。电子控制设备（例如 S7-200）可能发生故障和引起控制或监管的设备出现意外操作状况。因此，您应当安装与 S7-200 分开的安全装置，以便预防可能造成的人身伤害或设备损坏。



**警告：** 控制设备在不安全的条件下可能出现故障，导致控制装置出现意外操作状况。此类意外操作状况可引发疲劳、头晕、眼花、肌肉抽搐、停止、窒息、盟、请考虑使用与 S7-200 分开安装的紧急情况停止功能、机电越权或其他冗余安全设施。

#### 绝缘指南

S7-200 AC 的交流电路电源界限和 I/O 界限为 1500

VAC。这些绝缘界限经过检查和核准，可在交流电路和低压线路之间提供双倍安全绝缘保障。

所有与 S7-

200 连接的低压线路（例如 24V 电源）均需由经核准提供安全绝缘的交流电路和其他高压线路供电。此类来源包括在国际电气安全标准中定义的双重绝缘，并配有根据不同标准评级为 SELV、PELV、二类或有限电压的输出。



**警告：**使用未绝缘或单绝缘的电源从交流电线路为低压线路供电可能导致触摸安全的线路中出现危险的电击。此类意外的高压可能导致严重人身伤害、死亡或重大财产损失。始终仅限使用经核准为触摸安全、有限电压线路的高压至低压电源转换器。

#### S7-200接地指南

为您的应用程序接地的最佳方法是确保 S7-200 及其相关装置的所有公共导线连接均接地至一个单点。该单点应当直接与您的系统的接地相连。

为了获得进一步电气噪声保护，建议所有的直流电公共回线与同一个单点接地相连。将 24 VDC 传感器电源公共导线 (M) 与接地连接。

所有的接地线路必须尽量缩短，应当使用较大型号的电线，例如 2 mm<sup>2</sup> (14 AWG)。

在为接地定位时，请记住考虑安全接地要求以及保护性中断设备的适当操作。

#### S7-200布线指南

在为 S7-200 设计布线时，提供一个同时从 S7-200

CPU 电源、所有的输入、输出线路切断电源的单个断路器。提供过电流保护，例如熔丝或断路器，以便限制供电线路上的电流。

为可能遭受雷击的所有线路安装适当的电涌抑制设备。

避免将低压信号线路和通讯电缆与交流电线路和高能、快速切换直流电线路安装在同一个线路盒内。始终成对布线，将零线

尽量使用最短的线路，确保线路型号适当，可负载要求的电流。接头可接受 2 mm<sup>2</sup>至 0.3 mm<sup>2</sup> (14 AWG 至 22

AWG) 型号的线路。使用屏蔽线路，获得防止电气噪声的最佳保护。通常，在 S7-200 位置将屏蔽线路接地可提供最佳结果。

在为由外接电源供电的输入线路布线时，请在该线路中包括一台过电流保护设备。由来自 S7-200 的 24

VDC 传感器电源供电的线路无须安装外接保护装置，因为传感器电源已经是限制电流电源。

大多数 S7-200 模块有供用户布线的可移动接头。(请参阅《S7-

200 可编程控制器系统手册》附录 A，确定您的模块是否配有可移动接头。) 欲防止接头松脱，核实接头已安装稳妥，线路已

为了帮助预防安装设备中出现不需要的电流，S7-

200 在某些点提供绝缘界限。当您为系统规划布线时，应当考虑这些绝缘界限。请参阅《S7-

200 可编程控制器系统手册》附录 A，了解所提供的绝缘数量和绝缘界限位置。评级低于 1500VAC 的绝缘界限不能被视作安全的界限。



**提示：**对于通讯网络，如果不使用中继器，通讯电缆的最大长度为 50 米。S7-200 上的通讯端口为非绝缘端口。详情请参阅 [建立网络](#)。

#### 抑制线路指南

您应当为电感负载配备抑制线路，以便在控制输出关闭时限制电压升高。抑制线路保护输出，防止因高电感转换电流过早出

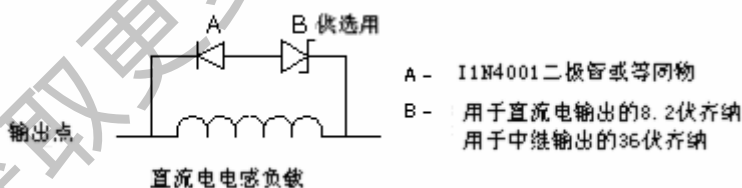


**提示：**特定抑制线路的有效性取决于使用的应用程序，您必须根据您的具体应用进一步核实。请始终确保

#### 控制直流电负载的直流电输出和继电器

直流电输出装有适合大多数应用程序的内装保护设备。由于继电器可用于直流电或交流电负载，因而不提供内装保护设备。

下图显示一个直流电负载抑制线路的范例。在大多数应用程序中，适合在电感负载中增加一个二极管 (A)，但是如果您的应用程序要求更快的关闭时间，则建议增加一个齐纳二极管 (B)。请务必选择适当大小的齐纳二极管，使之适合您的输出线路中的电流量。

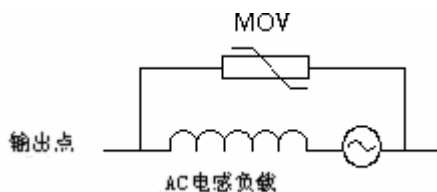


#### 控制交流电负载的交流电输出和继电器

交流电输出装有适合大多数应用程序的内装保护设备。由于继电器可用于直流电或交流电负载，因而不提供内装保护设备。

下图显示一个交流电负载抑制线路的范例。在大多数应用程序中，增加一个氧化变阻器 (MOV) 会限制峰值电压，并为内装 S7-200 线路提供保护。请核实 MOV 的工作电压至少应超过额定线路电压 20%。





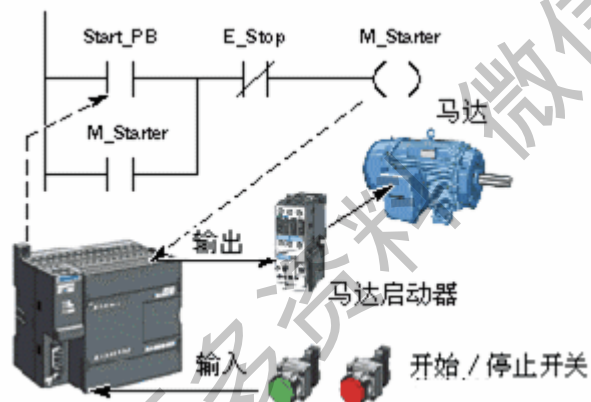
## 1.3 第2章 编程概念

### 1.3.1 2.1 如何控制程序作业

S7-200在程序的控制逻辑中不断循环，读取和写入数据。当您下载程序至PLC并将PLC放置在RUN（运行）模式时，PLC的中央处理器（CPU）按下列顺序执行程序：

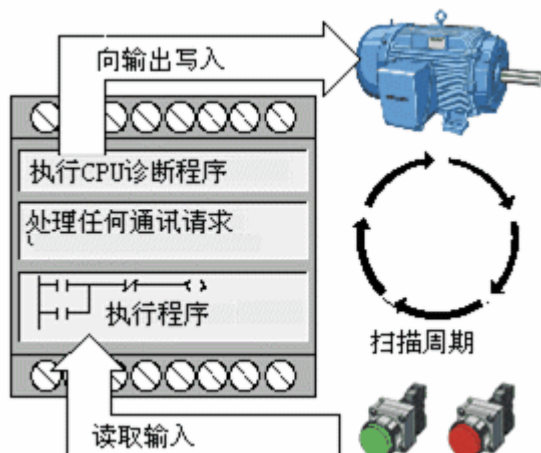
- S7-200读取输入状态。
- 存储在S7-200中的程序使用这些输入评估（或执行）控制逻辑。
- 当程序经过评估，S7-200将程序逻辑结果存储在称作进程图像输出寄存器的输出内存区中。
- 在程序结束时，S7-200将数据从进程图像输出寄存器写入至域输出。
- 重复任务循环。

以下是显示电中继图形与S7-200关系的简单图形。在该范例中，启动马达的开关状态与其他输入的状态相结合。因此，这些状态的计算决定进入启动马达的传动装置的输出状态。



S7-200反复执行一系列任务。该循环执行任务被称作扫描循环。如下所示，S7-200在扫描循环过程中执行大多数或全部下列任务：

- 读取输入：S7-200将实际输入状态复制至进程图像输入寄存器。
- 在程序中执行控制逻辑：S7-200执行程序的指令，并将数值存储在不同的内存区。
- 处理所有通讯请求：S7-200执行点至点或网络通讯要求的所有任务。
- 执行CPU自测试诊断程序：S7-200保证固件、程序内存和所有扩充模块均正常作业。
- 向输出写入：存储在进程图像输出寄存器中的数值被写入实际输出。

**注释:**

扫描循环的执行取决于S7-

200是位于STOP（停止）模式还是RUN（运行）模式。在RUN（运行）模式中，程序被执行；在STOP（停止）模式中，程序不被执行。



提示

**提示:** 在程序执行过程中，最好使用进程图像寄存器，而不要直接存取输入或输出。使用图像寄存器共有

- 扫描开始的所有输入取样会使扫描循环的程序执行阶段的输入数值同步化，并冻结这些数值。在程序执行过程中，程序存取图像寄存器的速度远远超过存取I/O点的速度，从而可以更快地执行程序。

另请参阅:

[PLC运行/停止模式](#)

## 1.3.2 2.2 编址概述

### 理解绝对和符号地址

您可以用绝对或符号方式识别程序中的指令操作数。绝对引用使用内存区和位或字节位置识别地址。符号引用使用字母数字字符组合识别某地址（符号地址）或数值（符号常量）。

SIMATIC程序员比IEC程序员更可能使用绝对地址；然而，STEP 7-Micro/WIN支持IEC编辑器的绝对编址。

### 程序编辑器如何显示地址举例

- I0.0 绝对地址由内存区和地址数目指定（SIMATIC程序编辑器）
- %I0.0 在IEC中，百分比符号位于绝对地址之前（IEC程序编辑器）
- #INPUT1 #符号位于局部变量之前（SIMATIC或IEC程序编辑器）
- INPUT1 全局符号名（SIMATIC或IEC程序编辑器）
- ?? or ??? 红色问号表示未定义的地址（必须在程序编译之前定义）

### 全局范围与局部范围

在符号表中指定的符号值（全局变量表）具有全局范围，在局部变量表中指定的符号值具有局部范围。

### 全局符号

全局符号可在SIMATIC和IEC程序编辑器中使用。

在SIMATIC程序中，您使用符号表进行全局符号赋值。在IEC程序中，您使用全局变量表进行全局符号赋值。在程序程序中使用

### 局部变量

局部变量可在SIMATIC和IEC程序编辑器中使用。

局部变量在各自的POU的局部变量表中赋值，其范围局限于建立局部变量的POU。每个POU都有单独的局部变量表。

举例:

您在称作SBR1的子例行程序局部变量表中定义一个称作INPUT1的变量。

当您从SBR1内引用INPUT1时, 程序编辑器将其识别为SBR1的一个局部变量。

然而, 如果您从程序中的其他位置(例如, 从OB1或第二个子例行程序)引用INPUT1, 程序编辑器不将其视作局部变量(与SBR1之外), 而将INPUT1视作未定义的全局符号。



**提示:**

如果您在局部和全局级别使用一个相同的地址名, 局部用法优先。即如果程序编辑器将PumpOn定义为全局符号。您还在SBR2中(而不是在SBR1中)将其定义为局部变量。当程序被编译时, 局部定义被用于SBR2中的PumpOn; 全局定义被用于SBR1中的PumpOn。局部变量使用临时PLC、L内存, 而不要求PLC程序内存空间。仅使用局部变量参数(或根本不使用参数)的子例行程序是可移动子例行程序, 可在一个以上程序中重复使用。并不要求您使用局部变量, 此为高级编程技术的一个选项。如果您想要在若干个POU中使用一个参数, 请迷洗 疟患 直淤勘碇薪 涸丁邈 址 牛 灰 涸丁邈 植勘淤浚 蛭 裨蚰 阨欠悦扛 POU的局部变量表分别赋值。因为局部变量使用临时内存, 每次POU被调用时, 请务必在POU中初始化局部变量。您无法保证局部变量编译准确 料乱准蔚 墓 讨斜 3质 奠怠 全局符号表支持全局符号常量(向数值常量直接分配符号名)。局部变量表不支持符号常量。

**局部变量表的硬件支持**

STEP 7-Micro/WIN的局部变量表功能要求硬件支持。您必须有一台CPU 221、CPU 222、CPU 224或CPU 226才能使用局部变量表。

另请参阅:

[直接与间接编址](#)

[子例行程序](#)

[中断例行程序](#)

[常量](#)

[选择助记符集](#)

[如何在LAD中输入地址 \(GS 3.5\)](#)

[如何在FBD中输入地址 \(GS 4.5\)](#)

[如何在STL中输入语句 \(GS 5.2\)](#)

## 1.3.3 2.3 如何组织程序

**控制程序的基本构成元素**

S7-200 CPU的控制程序由以下程序组织单位(POU)类型组成:

**主程序**

程序的主体(称为OB1), 是您放置控制应用程序指令的位置。主程序中的指令按顺序执行, 每次CPU扫描循环时执行一次。

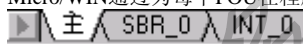
**子例行程序**

子例行程序是指令的一个选用集, 存放在单独的程序块中, 仅从主程序、中断例行程序或另一个子例行程序调用时被执行。

**中断例行程序** 中断例行程序是指令的一个选用集, 存放在单独的程序块中, 仅在中断事件发生时被执行。

STEP 7-

Micro/WIN通过为每个POU在程序编辑器窗口中提供单独的标记组织程序。主程序OB1总是第一个标记,

 其后才是您建立的子例行程序或中断例行程序。

**如何终止POU**

由于程序被划分(每个POU占有一个单独的标记), OB1或各种不同的子例行程序和中断例行程序的终止位置不成问题。编译程序 蝓 服导是 树鲈靡 桓鑫 尤登 END、MEND、RET或RETI终止每个POU。您不应该在程序中提供该代码; 如果您在程序中放入无登 END、MEND、RET或RETI, 编译程序会返回一则错误讯息。

**子例行程序**

子例行程序在您希望反复执行一个功能时特别有用; 您只需在子例行程序中写入一次逻辑, 然后在主程序过程中每次需要时调用子例行程序即可, 而无需在主程序中您希望执行功能的每个位置重写逻辑。这样做有若干优点:

- 您的总体代码尺寸减小。
- 因为您从主程序中移出代码, 扫描时间也会减少(在主程序中每次扫描循环时都会自动评估, 而无论是否执行)。可 杂刑 邕 氏 蓑米 永 谐 绦 颖 以 谏 韞 讨 肋 槐 坏 饕 饕 奉 欢 宰 永 谐 绦 蚪 衅 拦 馈
- 子例行程序容易移动; 您可以单独挑出一个功能, 并将其复制至其他程序中, 而无须或很少需要重复操作。

**注释:** V内存的使用限制子例行程序的可移动性, 因为一个程序的V内存地址赋值可能与另一个程序中的赋值发生冲突。相反 植勘淤勘碇糜 葜 械 邕 犯 持 档 淖 永 谐 绦 龄 春 菟 菟 淖 贫 蛭 槐 氏 P 幕 岢 鲛 直 喃 烦 逋 弧

**中断例行程序**

您可以写入中断例行程序, 处理某些预定义的中断事件; 中断例行程序不由主程序调用; 在中断事件发生时由PLC操作系统调用 R 蛭 蓑 瞻 菜 卮 尼 低 激 问 被 岬 襄 弥 卸 希 宰 罾 貌 灰 市 碇 卸 侠 谐 绦 蚰 慈 舛 贻 菜 沫 绦 蚱 绿 恢 襟 褂 玫 哪 哇 姪 D 梢 允 褂 镁 植 勘 淤 勘 碇 繁 V 卸 侠 谐 绦 蚪 葵 褂 味 偈 踟 洼 姝 切 闾 绦 蚱 绿 恢 致 氛 荨

**局部变量表的硬件支持**

STEP 7-Micro/WIN的局部变量表功能要求硬件支持。您必须有一台 CPU 221、CPU 222、CPU 224或CPU226才能使用局部变量表。

**另请参阅：**

[子例行程序](#)  
[中断例行程序](#)  
[局部变量表](#)

**1.3.4 2.4 项目元件及其工作原理**

在STEP 7-Micro/WIN中。项目为您提供程序与所有与PLC交流及下载指定程序至特定PLC所需信息之间联系的一种途径。项目包括下列基本元件：



程序块由可执行代码和注解组成。可执行代码包含一个主程序（OB1）和任意子例行程序或中断例行程序。代码被编译并下载至 PLC；程序注解不被编译和下载。



符号表是允许程序员使用符号编址的一种工具。符号有时对程序员更加方便，程序逻辑更容易遵循。下载至 PLC 的编译程序  
机械抛晃缘可罚 症碾畔 2.幌略可 PLC。



状态图允许您在执行程序时观察进程数值如果受到影响。状态图不下载至 PLC，而仅是监控 PLC（或模拟 PLC）活动的一种工具  
£



数据块由数据（初始内存值；常量值）和注解组成。数据被编译并下载至 PLC，注解则不被编译或下载。



系统块由配置信息组成，例如通讯参数、保留数据范围，模拟和数字输入过滤程序，用于 STOP（停止）转换的输出值和密码信  
i O 低晨樾畔 1.幌略可 PLC。



交叉引用窗口允许您检查表格，这些表格列举在程序中何处使用操作数以及哪些内存区已经被指定（位用法和字节用法）。在 RUN（运行）模式中进行程序编辑时，您还可以检查程序目前正在使用的边缘号码（EU、ED）。交叉引用及用法信息不下载至 PLC。



**提示：** • 您无法从 PLC 上载状态图、交叉引用或符号表信息，除非首先下载。  
 • 您在STEP 7-Micro/WIN项目中所作的修改在您将修改下载至 PLC 之后才会对程序产生影响。

**1.3.5 2.5 编辑器比较:LAD、FBD、STL**

SIMATIC S7-200CPU提供很多类型的指令，允许您执行各种类型的自动化任务。在 S7-200CPU中，有两种基本指令集：

- SIMATIC
- IEC 1131-3

STEP 7-

Micro/WIN提供不同的编辑器选项，允许您用这些指令建立控制程序。例如，您可能偏爱在一个图形环境中建立程序，而您所在  
镜牧砒恒漆丝瞻芷 嗒 鞞奈谋臼交惚嚶锯苑络瘳

建立程序时，您有两个基本的选择：

- 最适合您的应用程序的指令集类型（SIMATIC或IEC 1131-3）
- 最适合您的编程需求的编辑器类型（语句表、梯形逻辑或功能块图）

可采用下列指令集和编辑器组合：

- SIMATIC指令集和LAD、FBD或STL编辑器
- 配备LAD或FBD编辑器的IEC 1131-3指令集

本标题讨论下列三种程序编辑器：

[梯形逻辑](#)  
[功能块图](#)  
[语句表](#)

## 梯形逻辑编辑器

[返回顶端](#)

### STEP 7-

Micro/WIN梯形逻辑（LAD）编辑器允许您建立与电子线路图相似的程序。梯形编程是很多PLC程序员和维护人员选用的方法。您可用梯形逻辑范围建立的网络从简单到极为复杂。您可用中线输出建立网络；您甚至能连接一系列多个方框指令。系列连接方框指令带有“启用输出”（ENO）线条标记。如果方框在EN输入处有功率流，而且执行时无错误，则ENO输出向下一个元素传递功率流。ENO可用作启动位，表示指令成功完成。ENO位于堆栈顶端，影响用于后续指令执行的功率流。

下图显示一个梯形程序范例。



由图形符号代表的各种指令，包括三个基本形式。



接点—代表逻辑输入条件模拟开关、按钮、内部条件等。



线圈—通常代表逻辑输出结果模拟灯、马达启动器、干预中继、内部输出条件等。



方框—代表附加指令，例如计时器、计数器或数学指令。

您可用梯形逻辑范围建立的网络从简单到极为复杂。您可用中线输出建立网络；您甚至能连接一系列多个方框指令。系列连接方框指令带有“启用输出”（ENO）线条标记。如果方框在EN输入处有功率流，而且执行时无错误，则ENO输出向下一个元素传递功率流。ENO可用作启动位，表示指令成功完成。ENO位于堆栈顶端，影响用于后续指令执行的功率流。

#### 注释：

ENO功能仅限于STEP 7-Micro/WIN 3.0版（或更高版本）编程软件，与CPU221、CPU222、CPU224、CPU226或CPU 226XM配合使用。

选择LAD编辑器的要点是：

- 梯形逻辑便于新程序员使用。
- 图形显示通常很容易理解，在全世界通行。
- LAD编辑器可与SIMATIC和IEC1131-3指令集一起使用。
- 您始终可以使用STL编辑器显示用LAD编辑器建立的程序。

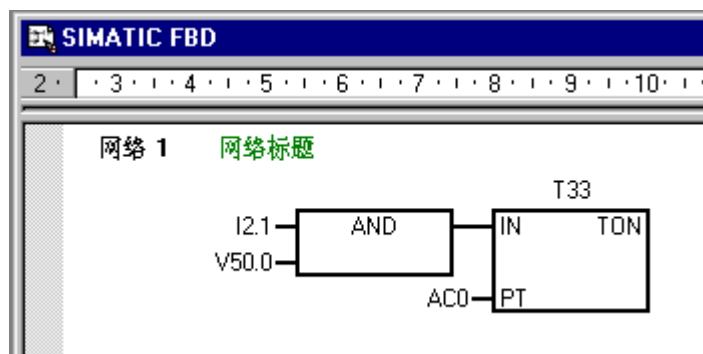
## 功能块图编辑器

[返回顶端](#)

### STEP 7-

Micro/WIN功能块图（FBD）编辑器允许您将指令作为与通用逻辑门图相似的逻辑方框检视。在LAD编辑器中无接点和线圈，但您可用逻辑门图建立的网络从简单到极为复杂。您可用中线输出建立网络；您甚至能连接一系列多个方框指令。系列连接方框指令带有“启用输出”（ENO）线条标记。如果方框在EN输入处有功率流，而且执行时无错误，则ENO输出向下一个元素传递功率流。ENO可用作启动位，表示指令成功完成。ENO位于堆栈顶端，影响用于后续指令执行的功率流。

下图显示一个用功能块图编辑器建立的程序范例。



如果方框在EN输入位置有功率流且执行无错误，则ENO输出向下一个元素传递功率流。ENO可用作启用位，表示指令成功完成。ENO位被用于堆栈顶端，影响用于后续指令执行的功率流。

#### 注释:

ENO功能仅限于STEP 7-Micro/WIN 3.0版（或更高版本）编程软件，与CPU221、CPU222、CPU224、CPU226或CPU 226XM配合使用。

选择FBD编辑器的要点是:

- 图形逻辑门显示式样对遵循程序流有益。
- FBD编辑器可用于SIMATIC和IEC 1131-3指令集。
- 您始终可以使用STL编辑器显示LAD编辑器建立的程序。
- 可扩充AND/OR（与 / 或）方框可简化绘制复杂的输入组合图。

#### 语句表编辑器

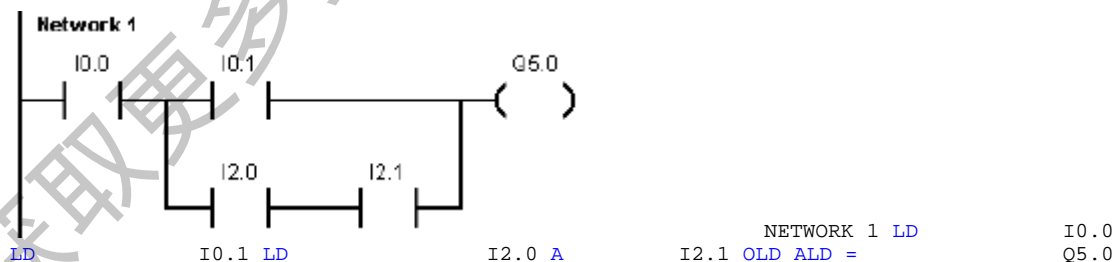
[返回顶端](#)

#### STEP 7-

Micro/WIN语句表（STL）编辑器允许您用输入指令助记符的方法建立控制程序。总体而言，STL编辑器对熟悉PLC和逻辑编程的经验丰富的程序员更合适。STL编辑器还允许您建立无法以其他方法用梯形逻辑或功能块图编辑器建立的程序。这是因为您是用CPU的本机语言在编程，而不是在图形编辑器中编程，后者有某些限制，以便正确绘图。下图显示一个语句表程序范例。

```
NETWORK 1 LD      I0.0 LD      I0.1 LD      I2.0 A      I2.1 OLD
ALD =           Q5.0
```

如图所示，这种基于文字的概念与汇编语言编程十分相似。CPU按照程序记录的顺序，从顶部至底部，然后再从头重新开始执行每条指令。STL和汇编语言在另一种意义上也很相似。S7-200CPU使用一种逻辑堆栈解决控制逻辑。LAD和FBD编辑器自动插入处理堆栈操作所需的指令。在STL中，您必须自己插入这些指令处理堆栈。下图列举一个LAD中的简单程序和STL中的对应程序。



下图显示堆栈中的情况:

堆栈	LD I0.0	LD I0.1	LD	指令
I2.0	A	OLD	ALD	
S0	I0.0	I0.1	I2.0	I2.0和I2.1 (I2.0和I2.1)
或I0.1	I0.0和[(I2.0和I2.1)或I0.1]			
S1		I0.0	I0.1	I0.1 I0.0
S2			I0.0	I0.0

S3

S4

S5

S6

S7

S8

选择STL编辑器的要点是：

- STL对经验丰富的程序员最适合。
- STL有时允许您解决无法用LAD或FBD编辑器方便地解决的问题。
- 您只能将SIMATIC指令集与STL编辑器一起使用。STL没有IEC指令集。
- 您始终可以用STL编辑器检视或编辑用SIMATIC LAD或FBD编辑器建立的程序，反之则并不一定正确。您无法始终使用SIMATIC LAD或FBD编辑器显示用STL编辑器写入的程序。

另请参阅：

[编程模式比较：SIMATIC、IEC \(GS 2.6\)](#)  
[ENO](#)

### 1.3.6 2.6 编程模式比较：SIMATIC、IEC

#### SIMATIC指令集

大多数PLC提供同类型的基本指令，但其外观和操作因供应商而异，通常有微小的差异。SIMATIC语言仅是西门子（Siemens）公司提供的指令集，您选择SIMATIC指令集时考虑的要点是：

- SIMATIC指令通常执行时间最短。
- 所有三种编辑器（LAD、FBD、STL）都可与SIMATIC指令集共同使用。

#### IEC 1131-3指令集

国际电子技术委员会是为电子技术所有领域制订全球标准的国际性组织。在最近几年里，他们制定了专门针对众多PLC编程领域男肉曜肌；旧希虎尸曜脊睦煌 PLC制造商提供在外观和操作上相似的指令。在SIMATIC指令集和IEC 1131-3指令集之间存在一些关键的差别：

- IEC 1131-3指令集中的指令较少。有些SIMATIC通常包含的指令在IEC 1131-3规格中并非标准指令。（与SIMATIC相当的指令以IEC 1131-1指令集的非标准扩展形式提供。在指令树中用红色标记（♦）表示。）
- IEC 1131-3指令集中的一些方框指令接受多种数据格式。这一概念通常被称为过载。例如，取代独立的ADD\_I（加整数）和ADD\_R（加实数）数学方框，IEC 1131-3 ADD（加）指令检查相加数据的格式，并自动在CPU中选择正确的指令。这样可节省宝贵的程序设计时间。
- 当您使用IEC 1131-3指令时，指令参数会自动检查适当的数据格式。例如，如果您尝试为指令输入一个整数值，而该指令期待一个位值（蚩毛眨蚩嵯允敬碧蟆U庖还δ苡兄浩×考跽俦暗逃侵u碧蟆

有关IEC编程的要点是：

- IEC 1131-3规定，在定义变量时必须说明变量类型，并支持数据类型的系统检查。
- 使用IEC 1131-3标准指令时，更容易学习如何为PLC的不同品牌建立程序。
- （由标准指定的）指令较少，但您能使用SIMATIC的许多指令。
- 某些IEC 1131-3指令与对应的SIMATIC指令（计时器、计数器、乘和除）操作方法不同。
- IEC 1131-3指令可能需要较长的执行时间。
- IEC 1131-3指令仅限在LAD和FBD编辑器中使用。

另请参阅：

[编辑器比较：LAD、FBD、STL \(GS 2.5\)](#)

### 1.3.7 2.7 如何使用向导帮助您编程

STEP 7-Micro/WIN提供下列“向导”（即配置工具）使您的编程更容易、自动化程度更高：

根据您对需要高速计数器之应用程序的进程需求，帮助您在不同的模式和中断事件之间进行选择。 [HSC指令向导](#)

帮助您在网络中多个S7-200 PLC之间配置通讯（信号交换信息）。 [NETR/NETW指令向导](#)

指导您为一个闭环控制进程定义PID（比例、积分、微分）算法。 [PID指令向导 PID调谐控制板](#)

帮助您配置供选用的TD 200操作员界面设备和建立TD200讯息。 [TD 200 SIMATIC文字显示功能 TD 200向导 \(TD 200](#)

[v2.1版或之前版本\)](#) [TD 200向导 \(TD 200 v3.0版或TD 200C v1.0版\)](#)

帮助您设置远程调制解调器或EM 241调制解调器模块，以便将本地S7-200 PLC与远程设备连接。 [EM241调制解调器模块性能](#)

[EM241调制解调器向导](#)

帮助您将位置控制用作应用程序的一部分。 [EM253位置模块性能 EM253位置向导 EM253控制板](#)

帮助您配置以太网模块，以便将S7-200 PLC与工业以太网网络连接。 [CP243-1以太网模块性能 CP243-1以太网向导](#)

配置CP243-1 IT互联网模块，将S7-200 PLC与以太网连接，并增加互联网电子邮件和FTP选项 [CP243-1 IT互联网模块性能 CP243-1 IT互联网向导](#)

帮助您建立在您创建的程序和AS-i主模块之间传输数据所需的代码。 [CP243-2 AS-i模块性能 CP243-2 AS-i向导](#)

在可移动非易失内存盒（64或256千字节）中存储菜单数据。您的程序能够读取和写入此菜单数据 [Recipe 向导](#)

帮助您在可移动非易失内存盒（64或256千字节）中记录工艺数据。数据日志可作为Windows文件得到提取和使用诸如MS Excel等Windows应用程序打开 [数据日志向导](#)

## 1.4 第3章 如何输入阶梯逻辑程序

### 1.4.1 3.1 如何建立项目

#### 打开新项目



双击STEP 7-Micro/WIN图标 ，或从“开始”菜单选择Simatic>STEP 7 Micro/WIN，启动应用程序。会打开一个新STEP 7-Micro/WIN项目。

#### 打开现有项目

从STEP 7-Micro/WIN中，使用文件菜单，选择下列选项之一：

- **打开**—允许您浏览至一个现有项目，并且打开该项目。
- **文件名称**—如果您最近在一项目中工作过，该项目在“文件”菜单下列出，可直接选择，不必使用“打开”对话框。


您也可以使用Windows Explorer浏览至适当的目录，无需将STEP 7-Micro/WIN作为一个单独的步骤启动即可打开您的项目。在STEP 7-Micro/WIN 3.0版或更高版本中，项目包含在带有.mwp扩展名的文件中。



提示：

一旦打开一个项目，您就可以开始写入程序。开始之前，您可能需要执行下列一项或多项任务： • **根据PLC类型进行范围检查** 为了使STEP 7-

Micro/WIN检查参数范围，您可以在写入程序前 **选择一个PLC类型**。（如果您已经为项目指定了一个PLC类型，指令树用红色标



示：  显示对PLC无效的任何指令。） • **设置通讯**

您可以 **现在设置通讯**，或等到准备好下载程序时再设置通讯。 • **定制工作区**


如有必要，您可以采用多种方法 **定制工作区**。

### 1.4.2 3.2 阶梯逻辑元素及其工作原理

阶梯逻辑（LAD）是一种与电气中继图相似的图形语言。当您在LAD中写入程序时，您使用图形组件，并将其排列成一个逻辑网络。下列元件类型在您建立程序时可供使用：

- **触点**  代表电源可通过的开关。电源仅在触点关闭时通过正常打开的触点（逻辑值一）；电源仅在触点打开时通过正常关闭或负值（非）触点（逻辑值零）。
- **线圈**  代表由功率流充电的中继或输出。



- **方框**  代表当功率流到达方框时执行的一项功能（例如，计时器、计数器或数学运算）。网络由以上元素组成并代表一个完整的线路。电源从左边的电源杆流过（在 LAD 编辑器中由窗口左边的一条垂直线代表）闭合触点，为线圈或方框充电。

### 1.4.3 3.3 在LAD中建立简单、串行和并行网络的规则

#### 放置触点的规则

每个网络必须以一个触点开始。  
网络不能以触点终止

#### 放置线圈的规则

网络不能以线圈开始；线圈用于终止逻辑网络。一个网络可有若干个线圈，只要线圈位于该特定网络的并行分支上。不能在网络分支

#### 放置方框的规则

如果方框有 ENO，功率流扩充至方框外；这意味着您可以在方框后放置更多的指令。在网络同级线路中，可以串联若干 ENO 的方框。如果方框没有 ENO，则不能在其后放置任何指令。

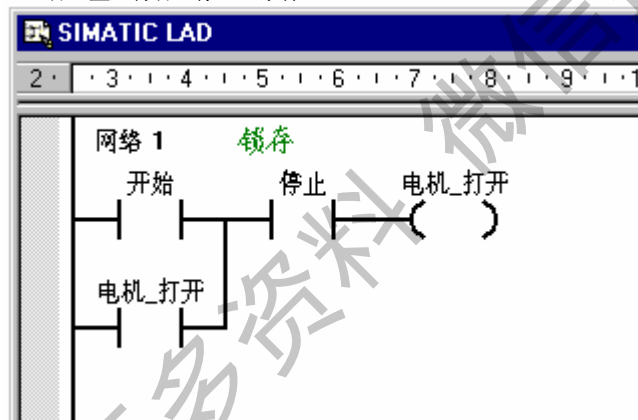
#### 网络尺寸限制

您可以将程序编辑器窗口视图划分为单元格的网格（单元格是可放置指令、为参数指定数值或绘制线段的区域）。在网格中，一个单独的网络最多能垂直扩充 32 个单元格或水平扩充 32 个单元。

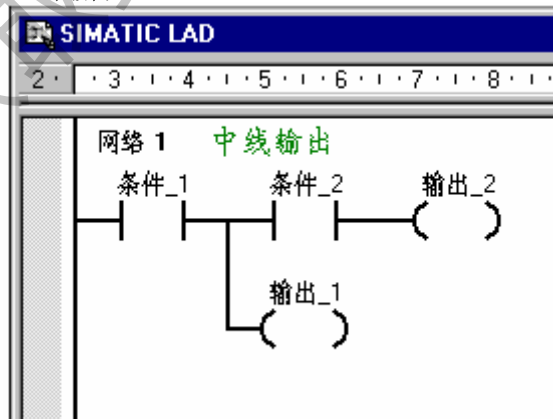
您可以用滑鼠右键在程序编辑器中单击，并选择“选项”菜单项目，改变网格大小。

下图列出一些 STEP 7-Micro/WIN LAD 编辑器中可能存在的逻辑结构。

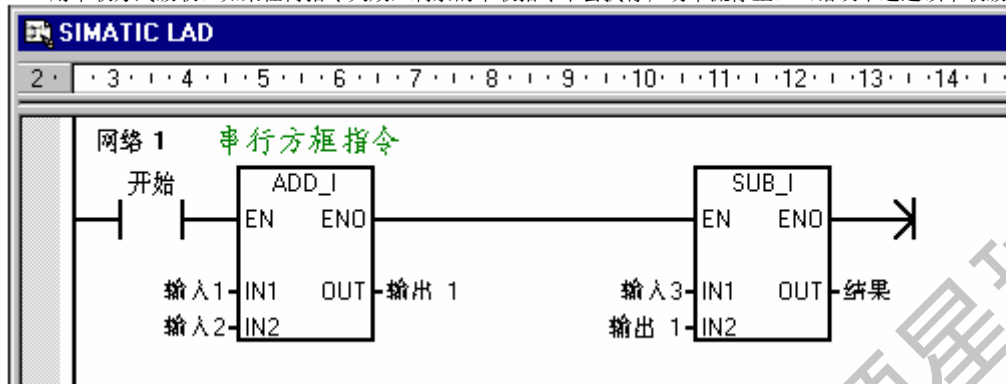
1. 该网络使用一个正常的触点（“开始”）和一个负（非）触点（“停止”）。一旦马达成功激活，则保持咬合，直至符合“停止”条件。



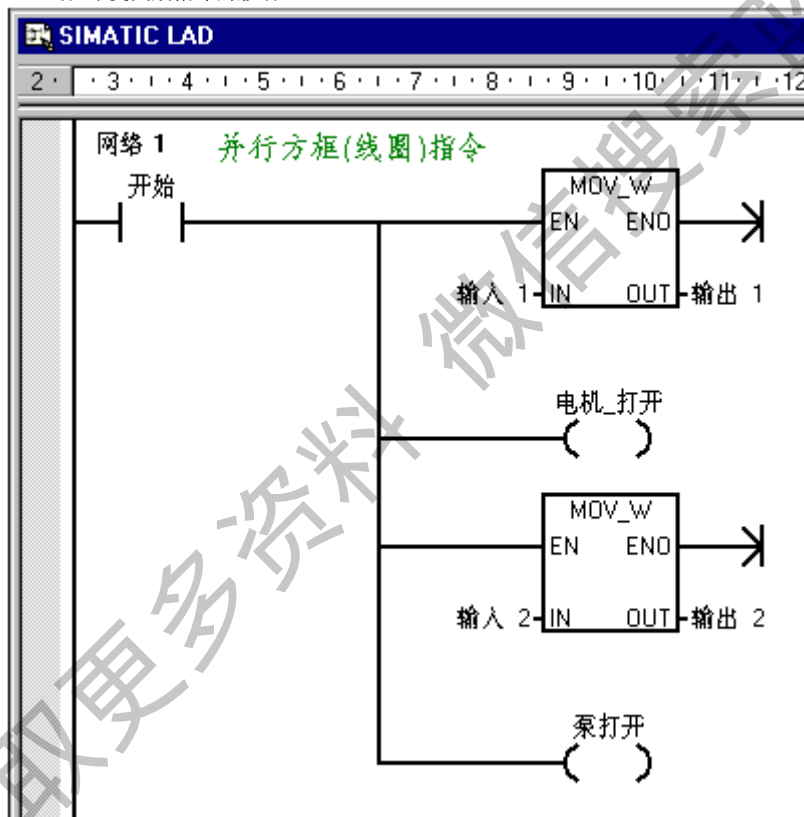
2. 请注意如果符合第一个条件，初步输出（输出 1）在第二个条件评估之前显示。您可以建立有中线输出的多个级档。



3. 如果第一个方框指令评估成功，电源顺网络流至第二个方框指令。您可以在网络的同一级上将多条 ENO指令用串联方式级联。如果任何指令失败，剩余的串联指令不会执行；功率流停止。（错误不通过该串联级联。）



4. 当符合起始条件时，所有的输出（方框和线圈）均被激活。如果一个输出未评估成功，电源仍然流至其他输出；不受失败指令的影响。



另请参阅：  
[Micro/WIN编译程序错误](#)

#### 1.4.4 3.4 如何在LAD中输入指令

有几种输入LAD指令的方法：

[指令树：拖放](#)  
[指令树：双击](#)  
[工具条按钮](#)  
[功能键](#)

[插入与覆盖方式](#)  
[如何划线](#)

### 插入与覆盖模式

(返回顶端)

STEP 7-Micro/WIN允许您在键盘上切换INSERT（插入）键，在两种编辑模式之间转换：

- 在插入模式（按INSERT（插入）键时选择）中，如果您将一条指令放在另一条指令上，程序编辑器将现有指令移开，为新指令让出位置。
- 在覆盖模式（INSERT（插入）键未按下时的默认值）中，如果您将一条指令放在另一条指令上，程序编辑器删除现有指令，并用新指令替换现有指令。

#### 覆盖模式中的参数保留

- 如果您用具有相同概况的另一个方框替换（覆盖）一条指令，对旧参数所作的任何赋值都会传输至新参数。（即如果第二条指令与第一条指令有同样数目的功率流输入、输入地址参数、功率流输出和输出地址参数，则当您用第二条指令覆盖第一条指令时参数赋值被保留。）

### 划线

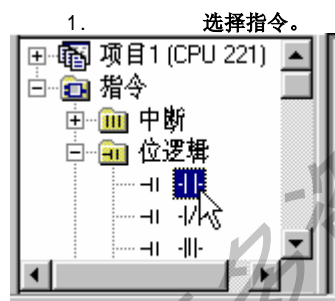
(返回顶端)

您可以从“程序”工具条使用水平和垂直线，或按住键盘上的CTRL键并按左、右、上或下箭头键，必要时在网络和左侧电源



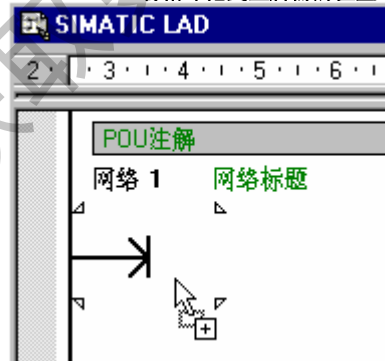
### 从指令树拖放

(返回顶端)

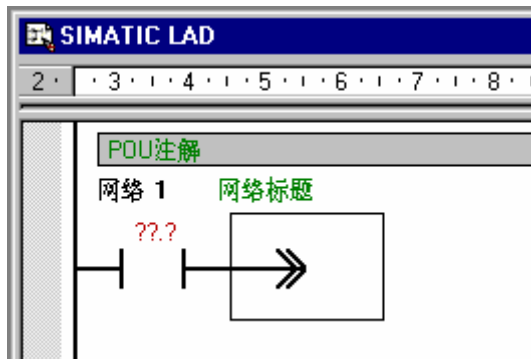


**注释：** 在IEC指令前的一个红点表示指令是标准IEC指令的非标准扩充，用于

2. 将指令拖曳至所需的位置。



3. 松开鼠标按钮，将指令放置在所需的位置。



放置在网络标题或另一条指令的参数上)。

注释：光标会自动阻止您将指令放置在非法位置（例如，

#### 从指令树双击 (返回顶端)

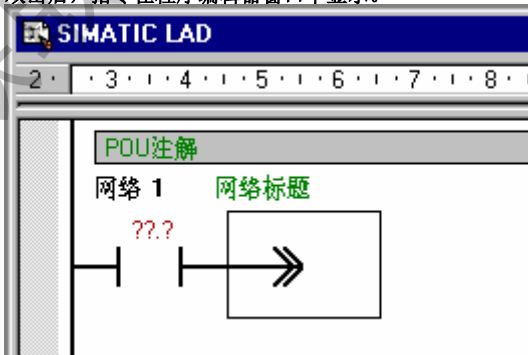
1. 在程序编辑器窗口中将光标放在所需的位置。一个选择方框在该位置周围出现。



2. 在指令树中，浏览至所需的指令并双击该指令。



双击后，指令在程序编辑器窗口中显示。

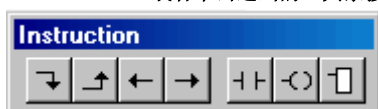


### 使用工具条按钮或功能键 (返回顶端)

1. 在程序编辑器窗口中将光标放在所需的位置。一个选择方框在位置周围出现。



2. 或者单击适当的工具条按钮 ...



或使用适当的功能键（F4=触点、F6=线圈、F9=方框）插入一个类属指令。

注释：该类属指令不能编址，并会阻止编译程序；您必须按步骤3所示选择一个具体指令类型。

3. 出现一个下拉列表。滚动或键入开头的几个字母，浏览至所需的指令。双击所需的指令或使用 ENTER 键  
迦敷弥嘴雌（如果此时您不选择具体的指令类型，则可返回网络，单击类属指令的助记符区域（该区域包含 ???，  
皇侵 欠 蚌眶=窈弥嘴贩 ENTER键，将列表调回。）



## 1.4.5 3.5 如何在LAD中输入地址

当您在LAD中输入一条指令时，参数开始用问号表示，例如（??）或（???）。

问号表示参数未赋值。您可以在输入元素时为该元素的参数指定一个常量或绝对值、符号或变量地址或者以后再赋值。如果有任何参数未赋值，程序将不能正确编译。

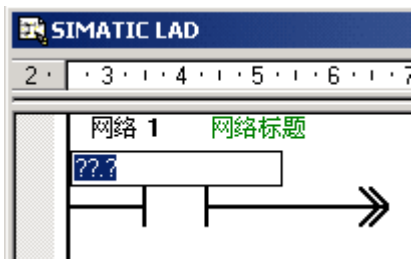


**提示：** 您可以选择工具（Tools）>

选项（Options）菜单命令，并选择**放置指令后启用操作数编辑**复选框，自动将光标放在您希望输入的参数上。

### 指定地址

欲指定一个常量数值（例如100）或一个绝对地址（例如I0.1），只需在指令地址区域中键入所需的数值。（用滑鼠或ENTER键选择键入的地址区域。）



欲指定一个符号地址（使用诸如INPUT1的全局符号或局部变量），您必须执行下列简单的步骤：

1. 在指令的地址区域中键入符号或变量名称。
2. 如果是全局符号，使用符号表 / 全局变量表为内存地址指定符号名。

**注释：**您不必预定义符号，则可在程序中使用，您可以以后再定义内存地址。

如果是局部变量，在程序编辑器窗口的顶端使用局部变量表。在“符号”列输入符号名。因为编译程序会自动指定L内存地址。

使用局部变量是一种高级编程技术。无经验的程序员应当考虑在符号表 / 全局变量表中将所有符号值指定为全局符号。



**提示：**在您输入这些地址数值后，程序编辑器会自动格式化地址数值。您无需键入格式化字符；在您完成输入后，程序编辑器会自动完成。

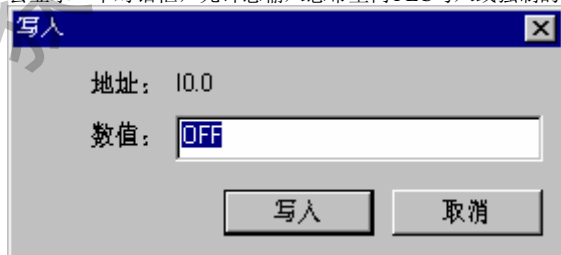
#### 写入和强制地址

**注释：**您必须在尝试写入或强制地址之前打开“程序状态”。

欲写入或强制地址，用鼠标右键单击操作数，并从鼠标右键菜单选择“写入”或“强制”。



这会显示一个对话框，允许您输入您希望向PLC写入或强制的数值。



#### 程序编辑器显示地址方法举例

I0.0 绝对地址由内存区和地址数目决定（SIMATIC程序编辑器）

%I0.0	在IEC中绝对地址前有一个百分号（IEC程序编辑器）
#INPUT1	局部变量前有一个#符号（SIMATIC或IEC程序编辑器）
INPUT1	全局符号名（SIMATIC或IEC程序编辑器）
???.? 或 ????.?	红色问号表示未定义的地址（必须在程序编译之前定义）

#### 匹配地址和定义符号

当您用鼠标右键单击指令的参数时，会弹出菜单，允许您在符号表中快速定义地址，或根据已经在地址区域中键入的数值最多

= 程雷佳贻芳钠ヅ洱



**提示：** • 使用ENTER键浏览网络的所有指令，每次一个参数，以便快速编辑所有必要的地址。 • 另外可用鼠标右键单击的方法选择个别参数，并定义符号，或使用弹出菜单为这些地址查找匹配。

#### 有效和无效符号名

符号名允许包含数字字母字符和下划线，也允许包含扩充字符（ASCII 128至ASCII 255）。第一个字符仅限使用字母和扩充字符。

**有效名称：**

a11  
a\_b\_1\_2

**非法名称：**

lloop  
l:kdl";ld

懒糖愚惚\_1284938

非法名称是以一个数字开始或包含不属于字母数目或扩充字符集的字符。

#### 另请参阅：

[程序编辑器如何在LAD中显示条目错误 \(GS 3.9\)](#)

[编址简介 \(GS 2.2\)](#)

[常量](#)

[选择助记符集](#)

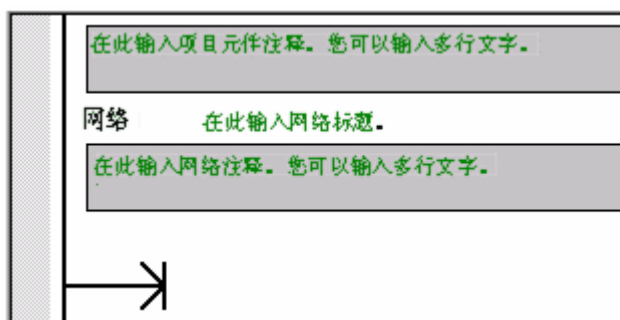
[子例行程序](#)

[中断](#)

### 1.4.6 3.6 如何在LAD中输入程序注解


LAD编辑器中共有四个注释级别：

- 项目组件注释
- 网络标题
- 网络注释
- 项目组件属性



#### 项目组件注释

在“网络1”上方的灰色方框中单击，输入POU注释。

您可以单击“切换POU注释”按钮  或选择和取消选择 **检视 (View) > POU注释 (POU Comments)** 选项，在POU注释“打开”（可视）或“关闭”（隐藏）之间切换。


每条POU注释所允许使用的最大字符数为4,098。POU注释是供选用项目，可视时，始终位于POU顶端，并在第一个网络之前显示。

#### 网络标题

将光标放在网络标题行的任何位置，输入一个识别该逻辑网络的标题。网络标题中可允许使用的最大字符数为 127。

#### 网络注释

在“网络1”下方的灰色方框中单击，输入网络注释。您可以输入识别该逻辑网络的注释，并输入有关网络内容的说明。

您可以单击“切换网络注释”按钮  或选择和取消选择 **检视 (View) > 网络注释 (Network Comments)** 选项，在网络注释“打开”（可视）和“关闭”（隐藏）之间切换。

网络注释中可允许使用的最大字符数为4,096。

#### 项目组件属性

您可以用以下两种方法中的一种存取“属性”标记。

用鼠标右键单击指令树中的 **POU**，并从鼠标右键菜单中选择“属性”。





用鼠标右键单击程序编辑器窗口中的任 何一个POU 标记，并从弹出菜单选择“属性”。



“属性”对话框中有两个标记：

- 一般
- 保护



#### “一般”标记

“一般”标记允许您为子例行程序、中断例行程序和主程序块（OB1）重新编号和重新命名，并为项目指定一个作者。

#### 注释：

您不能将默认名称（STEP 7-Micro/WIN指定的POU地址，例如，SBR1代表子例行程序或INT1代表中断例行程序）用作符号名，因为这样即构成撞锤持怠H 纒 彳 戳朔 琶 持档墓嬰颞 蹦 (6)员喊氩纒纒保 STEP 7-Micro/WIN会报告一则错误。

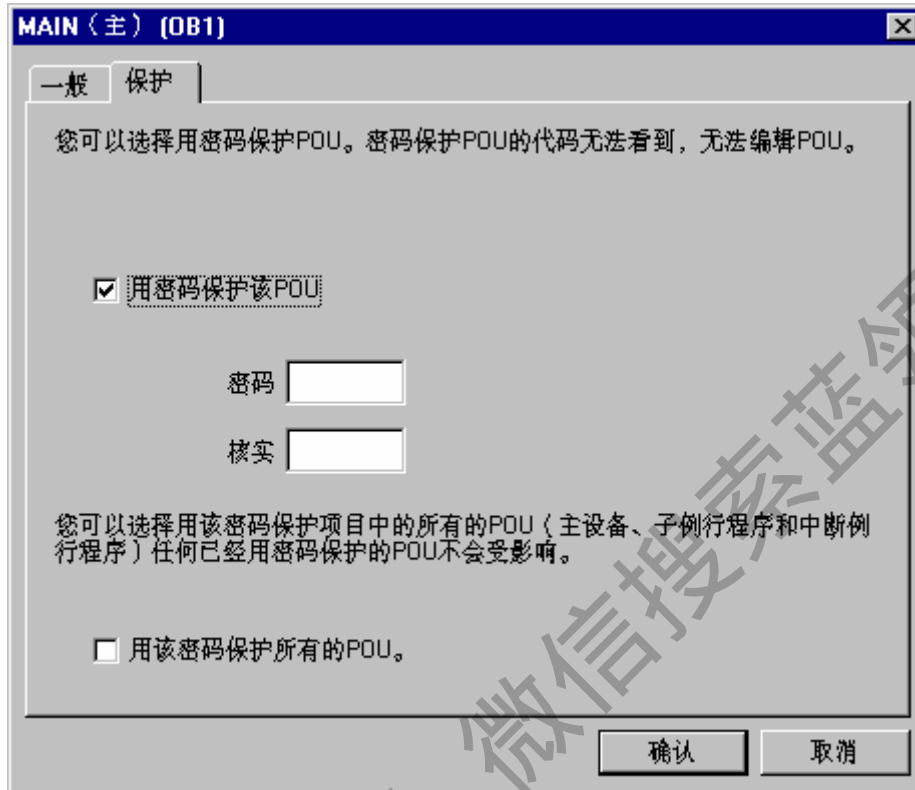
如果您在程序中为POU指定符号名，符号名会在程序代码中显示，即使您没有启用“符号编址”视图亦如此。符号表显示一个列出所有符号名赋值的标记（“POU符号”）。您只能检视该标记；但无法从符号表编辑条目。欲改变赋值，您必须编辑适当的POU的“属性”对话框。

#### “保护”标记

“保护”标记允许您选择一个密码保护POU，以便其他用户无法看到该POU，并在下载时加密。

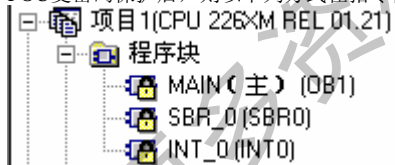
欲用密码保护POU:


1. 用滑鼠单击“主POU”，并选择“属性...”。
2. 选择“保护”标记。
3. 选择“用密码保护该POU”复选框。输入一个四个字符的密码并核实该密码。



您还可以选择“用密码保护所有POU”复选框，保护项目中的所有POU（主程序、子例行程序和中断例行程序）。

POU受密码保护后，则以下列方式在指令树中显示：



此外，在程序编辑器窗口的装订线中会显示一个灰色锁定图标 .

欲锁定POU:

1. 浏览至“保护”标记。
  2. 输入密码。
  3. 单击“授权”。
- 您必须为您选择用密码保护的每个POU重复该步骤。

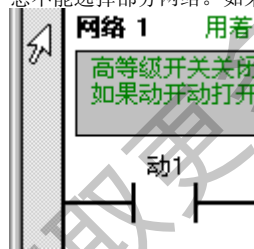


### 1.4.7 3.7 如何在LAD程序中编辑元素

#### 剪切、复制、粘贴或删除多个网络

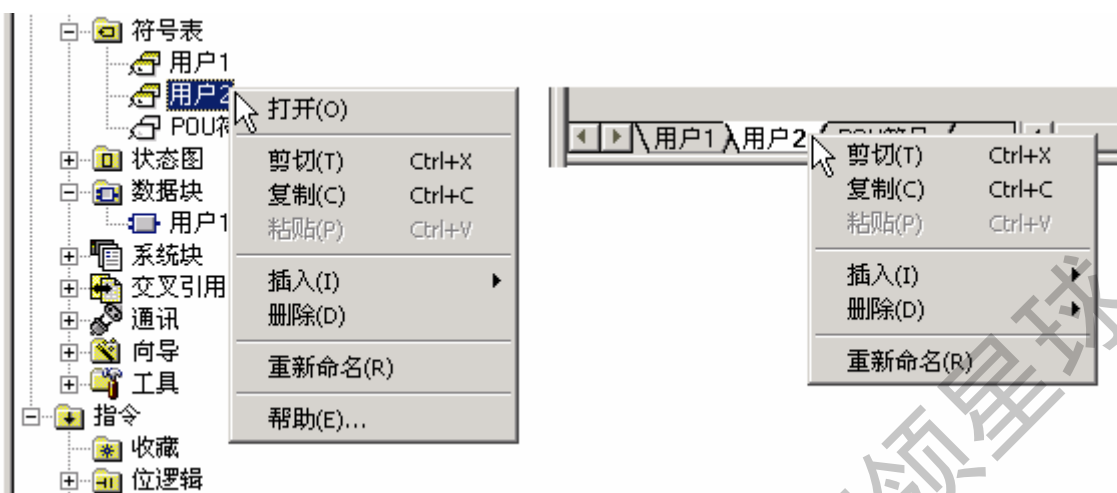
通过拖曳滑鼠或使用SHIFT键和UP（向上）、DOWN（向下）箭头键，您可以选择多个相邻的网络，用于剪切、复制、粘贴或删除选项。使用工具条按钮，从“编辑”菜单选择一条命令，或用滑鼠右键单击，调出编辑选项的弹出菜单。

如下图所示，将鼠标移到编辑器的左侧边缘（装订线区域），然后单击以开始对多个网络进行拖过选择。您不能选择部分网络。如果您尝试选择部分网络，会自动选择整个网络。



#### 剪切、复制、粘贴项目元件

如下图所示，将鼠标移到指令树或编辑器标记上，然后单击鼠标右键。由弹出菜单中选取“复制”命令，以复制整个项目元件。

**注释:**

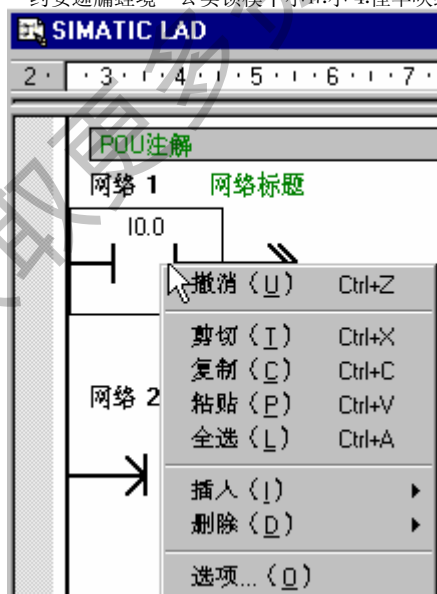
“复制”功能只在您选择了一个需要复制的项目后才能使用。您作出的选择确切描述了将复制到Windows剪贴板缓冲器的内容

您可以在项目中选择以下目标

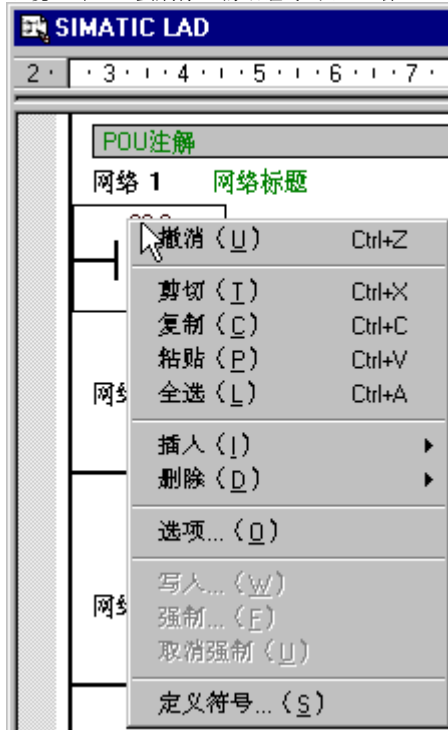
- 程序文字或数据域
- LAD、FBD、STL等编辑器中的指令
- 单个网络：将鼠标放在“网络”的左侧边缘上，然后单击
- 多个相邻网络：在选取第一个网络后，使用Shift+下箭头/上箭头或Shift+Page Up/Page Down来扩大或减少所包括的网络数目
- 某POU内的所有网络：请使用 **编辑** >
- **全选** 菜单命令，或用鼠标右键单击某指令树POU分支，或用鼠标右键单击某标记名
- 选定的数据块文字，或整个数据块标记页：  
当通过用鼠标右键单击编辑器标记名而复制整个数据块标记页时，随后的粘贴操作能够输出用制表符分隔的数据。举例来说，倘若将此数据块文字粘贴到MS Excel，则电子表格的多个单元行将得到填充。
- 选定的符号表行或列，或整个符号表标记页
- 选定的状态图行或列，或整个状态图标记页

**编辑单元格、指令、地址和网络**

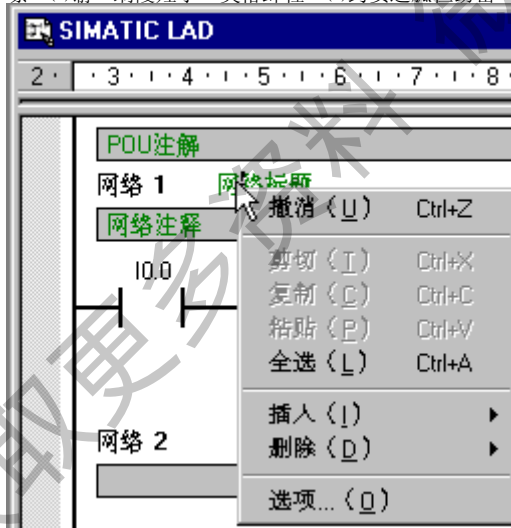
- 当您单击程序编辑器中的空单元格时，会出现一个方框，显示您已经选择的单元格。您可以使用弹出菜单在空单元格
- 单击指令时，会在指令周围出现一个方框，显示您选择的指令。您可以使用弹出菜单在该位置剪切、拷贝或粘贴指令



- 单击指令参数时，会在域周围出现一个方框，显示您选择的参数。您可以使用弹出菜单撤销键入、剪切、拷贝、粘贴



- 当您在网络标题行中单击时，您可以编辑项目的标题。您可以使用弹出菜单全选、取消键入的内容、对整个网络执行



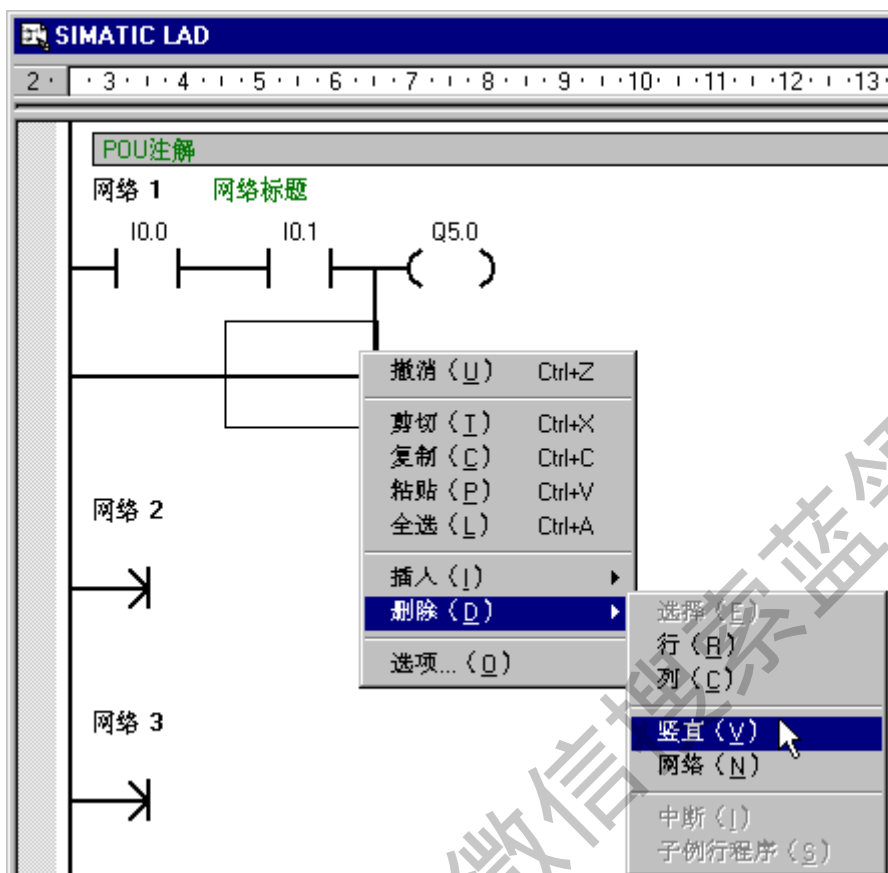
您也可以使用工具条按钮、标准窗口控制键和“编辑”菜单剪切、拷贝或粘贴选项。

#### 将元素移近

您可以剪切和粘贴元素和行，删除行或列；但是，程序编辑器要求元素之间有一定的间距。在某些情况下，您不能把一个元素

#### 删除元素

您可以用DELETE（删除）或BACKSPACE（退格）键删除个别单元格；您可以用“编辑”菜单或滑鼠右键单击的方法调出弹



**注释：** 为了正确选择一条需要删除的垂直线，始终将光标放在垂直线左边的单元格上。

另请参阅：  
[快捷键 \(GS 1.2\)](#)

## 1.4.8 3.8 如何使用查找、替换和转入功能

欲使用“查找 / 替换”或“转入”：

- 选择 **编辑 > 查找 (Edit>Find)**， **编辑 > 替换 (Edit>Replace)** 或 **编辑 > 转入 (Edit>GoTo)** 菜单命令
- 按 CTRL+F 执行查找， CTRL+H 执行替换， 或 CTRL+G 执行转入

**如何使用：**  
[查找](#)  
[替换](#)

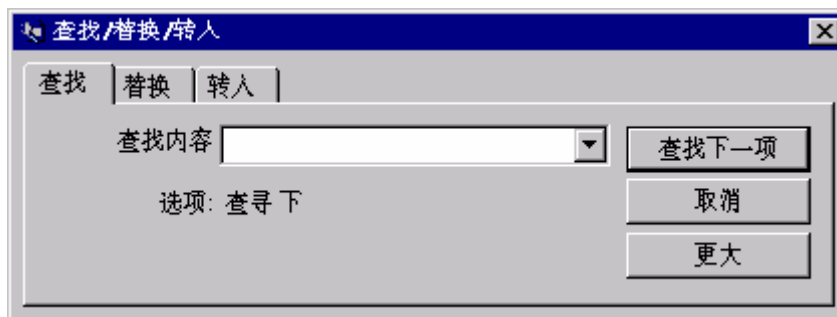
**在何处使用**

可以在程序编辑器窗口、局部变量表、符号表、状态图、交叉引用标签和数据块中使用“查找”、“替换”和“转入”。

**如何作业**

- “查找”功能允许您查找指定的字符串，例如操作数、网络标题或指令助记符。（“查找”不搜索网络注释，仅搜索峻晏俊！安擒摇辈凰阍 LAD和FBD中的网络符号信息表。）
- “替换”功能允许您替换指定的字符串。（“替换”对指令助记符不起作用。）
- “转入”功能允许您通过指定网络数目或您希望浏览的行的方式快速移至另一个位置。

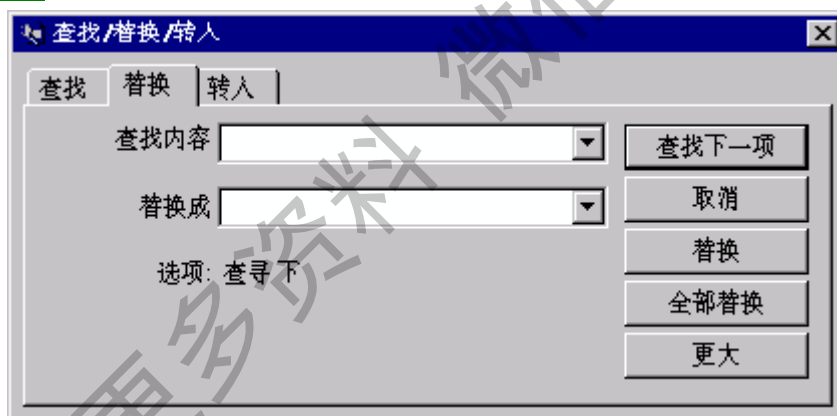
[查找功能](#)  
[返回顶端](#)



1. 在“查找内容”域中键入要搜索的字符串。
  2. 欲移至下一个搜索字符串，单击“查找下一项”按钮。
- 注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反映了操作数在代码中的存储方式。另外，如果您在 STL 中建立的程序包含在 LAD 或 FBD 编辑器中非法的网络，在 LAD 或 FBD 编辑器中检视程序时，“查找下一项”命令不对这些网络执行搜索操作。
- 欲进一步您可以使用“搜索”列表框选择搜索方向。
  - 您可以选择“大小写匹配”复选框，仅搜索与您“查找内容”中键入的字符串大小写数值相同的字符串。
  - 您可以选择“全字”复选框，除去包含作为较长字一部分的搜索短语的字符串。
  - 您可以选择适当的 SPAN（跨接）复选框，搜索所有的 POU（OB1、所有的子例行程序和中断例行程序）或局部变量表乃 惺道 7. 疔砣龋刺 肌
  - 您可以指定对一定的行范围进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认段 B 荒 部梢裁谕 缙屑 膈蚣 胃泻牛 魑 陶鞞目 己徒岷
  - 您可以指定是否通过选择适当的复选框搜索网络标题、POU 和网络注释和 / 或程序代码。

#### 替换功能

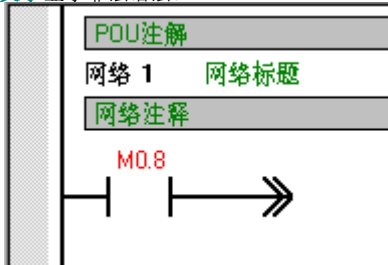
[返回顶端](#)



1. 在“查找内容”栏中键入要搜索的字符串。
  2. 在“替换内容”栏中键入您希望用作替换搜索字符串的字符串。
  3. 欲查找存在的搜索字符串，单击“查找下一项”按钮。
- 注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反映了操作数在代码中的存储方式。另外，如果您在 STL 中创建的程序包含在 LAD 或 FBD 编辑器中非法的网络，在 LAD 或 FBD 编辑器中查看该程序时，“查找下一项”命令不对这些网络进行搜索操作。
4. 如果您希望替换字符串，单击“替换”。如果您已经仔细地定义搜索字符串而且没有误改的可能，您可以单击“全部替换”，替换所有存在的字符串，而无须逐一检查每个字符串。
- 欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方法定制搜索：
- 您可以使用“搜索”列表框选择搜索的方向。
  - 您可以选择“大小写匹配”复选框，仅搜索与您“查找内容”中键入字符串大小写数值相同的字符串。
  - 您可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
  - 您可以选择适当的“SPAN”复选框，对 POU（OB1、所有的子程序和中断例行程序）或局部变量表、符号表或状态图 砣械乃 惺道 兴陶整
  - 您可以指定对一部分行进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认范围 荒 部梢约 驴 缙蚰泻牛 魑 陶鞞钠鸮 己徒岷
  - 您可以指定是否通过选择适当的复选框搜索网络标题、POU 和网络注解。

### 1.4.9 3.9 程序编辑器如何在LAD中显示输入错误

红色文字显示非法语法。

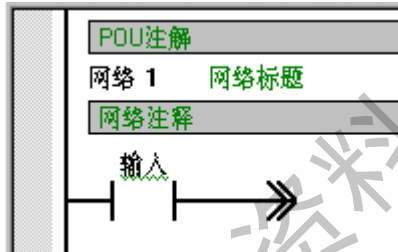


注释：当您用有效数值替换非法地址值或符号时，字体自动更改为默认字体颜色（黑色，除非您已定制窗口）。

一条红色波浪线位于数值下方，表示该数值或是超出范围或是不适用于此类指令。



一条绿色波浪线位于数值下方，表示正在使用的变量或符号尚未定义。STEP 7-Micro/WIN允许您在定义变量和符号之前写入程序。您可随时将数值增加至局部变量表或符号表中。



提示：• 用滑鼠右键单击，调出包含与全局符号最接近的五个匹配项的菜单。• 使  
谩岸丁宸 拧倍巴翱颞 气鑫炊丁宓姆 拧

### 1.4.10 3.10 如何在LAD中编译


如何编译

您可以用工具条按钮或PLC菜单进行编译。



- “编译”  允许您编译项目的单个元素。当您选择“编译”时，带有焦点的窗口（程序编辑器或数据块）是编译窗



- “全部编译”  对程序编辑器、系统块和数据块进行编译。当您使用“全部编译”命令时，哪一个窗口是焦点无关紧要。


#### 使用输出窗口解决错误

当您编译时，“输出窗口”列出发生的所有错误。错误根据位置（网络、行和列）以及错误类型识别。您可以双击错误线，调出程序编辑器中包含错误的代码网络。

另请参阅：

[Micro/WIN编译程序错误](#)

## 1.4.11 3.11 如何保存作业

您可以使用工具条上的“保存”按钮  保存您的作业，或从“文件”菜单选择“保存”和“另存为”选项保存您的作业。



- “保存”允许您在作业中快速保存所有改动。（然而，您初次保存一个项目时，会被提示核实或修改当前项目名称和柯嫉哪 涎之睡#
  - “另存为”允许您修改当前项目的名称和 / 或目录位置。
- 当您首次建立项目时，STEP 7-Micro/WIN提供默认值名称“Project1.mwp”。您可以接受或修改该名称；如果您接受该名称，下一个项目的默认名称将自动递增为“Project2.mwp”。
- STEP 7-Micro/WIN项目的默认目录位置是位于“Microwin”目录中的称作“项目”的文件夹，您可以不接受该默认位置。

## 1.5 第4章 如何输入功能块图程序

### 1.5.1 4.1 如何建立项目

#### 打开新项目

双击STEP 7-Micro/WIN图标 ，或从“开始”菜单选择Simatic >STEP 7-Micro/WIN，启动应用程序。打开新的STEP 7-Micro/WIN项目。

#### 打开现有项目

在STEP 7-Micro/WIN中，使用文件菜单，并选择下列选项之一：


- 打开—允许您浏览至现有项目并打开该项目。
- 文件名称—如果您最近在某个项目中作业，则该项目会在“文件”菜单中列出，您可以直接选择该项目，无需使用“

您还可以使用Windows Explorer浏览至适当的目录并打开项目，无需事先单独启动STEP 7-Micro/WIN。在STEP 7-Micro/WIN 3.0版或更高版本中，项目包含在单个文件中，带有扩展名.mwp。



**提示：**一旦打开一个项目，您可以开始写入程序。开始之前，您可能需要执行一项或多项下列任务：

**根据PLC类型进行范围检查** 为了让STEP 7-Micro/WIN检查参数范围，您可以在写入程序之前 [选择PLC类型](#)。（如果您已经为项目指定PLC类型，指令树用红色 标记 x:

 表示对您的PLC无效的所有指令。）

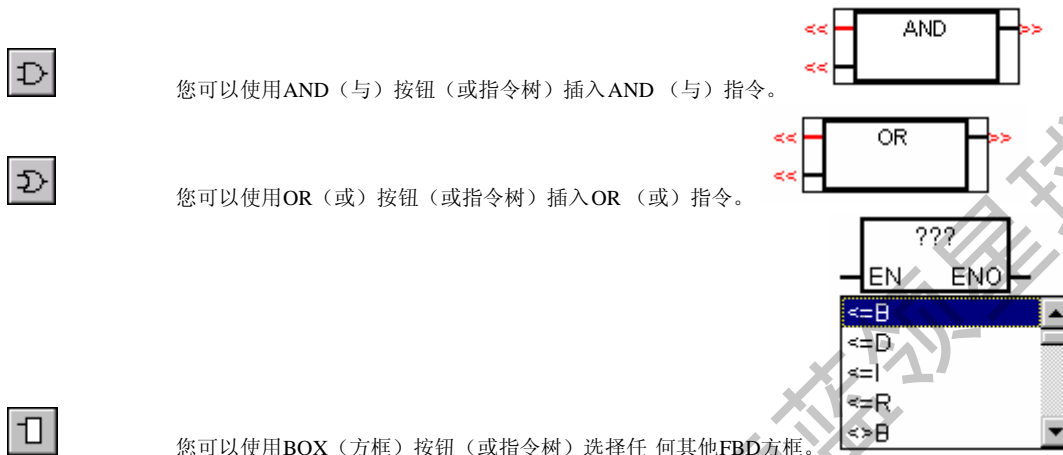
**设置通讯**

您可以 [现在设置通讯](#)，或等到准备下载程序再设置。

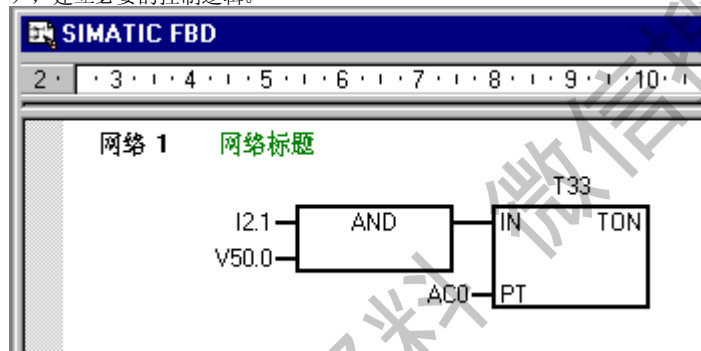
**定制工作区** 您可以多种不同的方法 [定制工作区](#)。

## 1.5.2 4.2 功能块图元素及其工作原理

STEP 7-Micro/WIN的功能块图（FBD）编辑器允许您与公用逻辑门图相似的逻辑框查看指令：



程序逻辑从这些方框指令之间的连接派生。即来自一条指令的输出（例如 AND（与）方框）能用于启用另一条指令（例如计时器），建立必要的控制逻辑。



本标题说明可用于修改FBD指令输入和输出的几种方法：

### 反转输入

您可以反转布尔（位值）输入、建立一个反转逻辑条件（输入成为通常关闭 / ON（打开），而不是通常打开 / OFF（关闭）。

### 将输入或输出设为立即模式

您可以将布尔（位值）输入或输出转换为立即输入 / 立即输出，使地址立即更新，而不是在程序定期扫描循环过程中按固定的间隔时间更新。

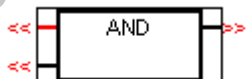
### 增加或移除附加输入

如果程序逻辑要求，您也可以将附加输入增加至 AND（与）或 OR（或）指令。如果您改变主意，也可以移除附加输入，并将指令

### 反转输入 (返回顶端)

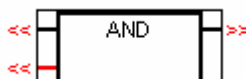
您可以反转布尔（位值）输入或切换至正常的输入：


1. 所需的指令（单击指令或按 ENTER 键，直至指令周围出现一个选项框）。



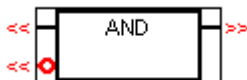
2. 使用 TAB 键，移至您希望更改的输入（或直接单击输入）。

注释：所选的输入行为红色；未选择的输入行为黑色。



3. 单击“切换反转”按钮，或按F11快捷键，根据需要在正常和反转之间切换输入。

注释：正常输入看起来像行；反转输入看起来像圆圈。

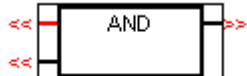


注释：反转和立即属性可以组合起来应用于参数；他们不互相排斥。

### 将输入或输出设为立即模式 (返回顶端)

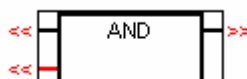
您可以将布尔（位值）输入或输出改变为立即I/O值，您也可以将立即值切换为缓冲值：


1. 选择所需的指令（单击指令或按ENTER键，直至在指令周围出现一个选项框）。

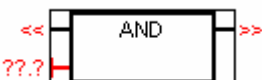


2. 使用TAB键，移至您希望更改的参数（或直接单击参数）。

注释：所选的输入或输出行为红色；未选择的输入行为黑色。



3. 单击“切换立即”按钮，或按CTRL + F11快捷键，根据需要参数在缓冲和立即之间切换。

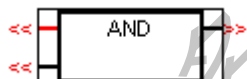



注释：反转和立即属性可以组合起来应用于参数；他们不互相排斥。

### 增加或移除附加输入 (返回顶端)


您可以对AND（与）和OR（或）指令增加和移除附加输入：

1. 选择所需的指令（单击指令或按ENTER键，直至在指令周围出现一个选项框）。



2. 单击“增加输入”按钮或按加号快捷键，增加附加输入。 -或-



3. 单击“移除输入”按钮或按减号快捷键，移除附加输入。



注释：

支持单输入AND（与）方框。单输入AND（与）方框是STL

LD指令或单LAD触点的FBD程序视图。欲在程序中放置单输入AND（与）方框，请使用双输入方框，然后利用删除输入按钮



删除单输入。

### 1.5.3 4.3 在FBD中建立网络的规则

FBD网络由一个或多个FBD方框指令组成。无须使用触点，您可以将一个或多个位值作为输入分配给FBD方框指令。指令输出可

- 如果适合您的程序的上下文，您可以建立仅包含一条FBD指令的网络。
- 每个网络没有固定的最大指令数目。您可以将FBD程序编辑器窗口视作一个划分为单元格的网格（单元格是您可以放置指令、将数值分配给参数或划线的区域）。在该网格中，单个网络最多可沿水平方向延伸32个单元格或沿垂直方向延伸32个单元格。您可以用滑鼠右键在程序编辑器单击，并选择“选项”菜单项目，编辑栅格尺寸。
- 如果方框指令有>>输出，您必须提供与另一个方框的连接或为输出参数指定赋值。但是，在>|输出标有ENO时，输出可以留白、与另一个方框连接或指定赋值。
- 您不能将几个指令输出一一起直接连接为一条水平线。如果您希望连接几个输出，必须将每个输出与AND（与）方框或OR（或）方框的输入参数连接（产生单一输出）。

### 1.5.4 4.4 如何在FBD中输入指令

本标题讨论插入FBD指令的几种方法：

[指令树：拖放](#)  
[指令树：双击](#)  
[工具条按钮](#)  
[功能键](#)

当您插入指令时，可能还需要下列与本题目有关的信息：

[纠正指令位置](#)  
[如何划线](#)  
[插入与覆盖模式](#)

**相关标题：**

[反转布尔输入](#)  
[将布尔输入或输出设为立即模式](#)  
[增加或移除附加输入（仅限AND/OR（与/或））](#)

**纠正指令位置：从左至右作业**

[返回顶端](#)

建立FBD网络时，始终从左至右作业，以便利用程序编辑器的特别功能：

- （当插入光标作为前一个方框的打开输出>>正确放置在同一行上时）方框自动连接
- （当插入光标作为前一个方框的打开输出>>正确放置在同一行上时）自动调整方框位置
- 自动调整方框尺寸（以便启用正在插入的方框输入，与前一个方框的打开输出>>对齐和连接）

当您从左至右作业，并适当放置指令时（将插入光标放在与前一条指令的输出>>相同的行中），程序编辑器会自动连接和排列指令。

**为参数留出位置**

将FBD程序编辑器视作由列和行组成的网格。

您不得将指令放在网络的最上一行；光标必须放在第二行或下方。另外，如果指令左边有输入，则不能在程序编辑器的第一



**提示：**当您在程序编辑器中工作时，留心FBD工具条；如果将光标放在程序编辑器的一个不允许

放置特定类型线条或指令的区域中，线条工具和指令按钮会被自动禁用。

**注释：**将一条指令与另一条指令连接时，不必在指令之间留出额外的空格。您可以在第一条指令的适当输出参数中（>>）直接放置第二条指令。

**将输出赋值更改为连接**

如果您将参数值指定给指令的输出>>，但是后来决定将指令与另一条指令连接，您可以直接在第一个输出上放置第二条指令。以前指定给该输出参数的数值（例如Q0.1）会被自动盖写，输出与替换指令连接。

**划线**

[返回顶端](#)

您可以使用程序工具条中的水平和垂直线，或按住CTRL键并按键盘上的左、右、上或下箭头键，必要时在网络元素之间划线（例如，建立分支）。



### 插入与覆盖模式

[返回顶端](#)

STEP 7-Micro/WIN允许您在键盘上按“插入”键，在两个编辑模式之间切换：

- 在插入模式（当您按“插入”键时选择）中，如果您在另一条指令上放置一条指令，程序编辑器移动现有指令，为新指令留出位置。
- 在覆盖模式（“插入”键未按下时的默认值）中，如果您在另一条指令上放置一条指令，程序编辑器删除现有指令，并用新指令替换现有指令。

### 覆盖模式中的参数保留

- 如果您用具有相同概况的另一个方框替换（覆盖）一条指令，您对旧参数所作的任何赋值都会转移至新参数。（即如果第二条指令具有与第一条指令相同的逻辑流输入数目、相同的输入地址参数数目、相同的逻辑流输出数目和相同的输出地址参数数目，则当您用第二条指令覆盖第一条指令时，参数赋值被保留。）
- 对于AND（与）方框和OR（或）方框，即使您修改了原方框以便增加附加输入，旧参数仍然保留。（例如，如果您<sup>a</sup>AND（与）方框分配了四个输入赋值，您用一个OR（或）方框覆盖AND（与）方框，OR（或）方框会自动继承四个钟惺凉癖#

### 从指令树拖放

[返回顶端](#)

#### 1. 选择指令。

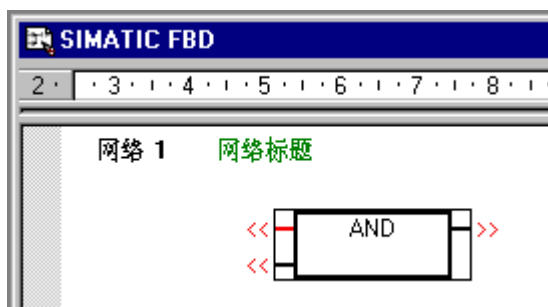


**注释：**位于IEC指令前的红点表示指令是标准IEC指令的非标准扩展，因为有一条相似的指令是SIMATIC指令集的一部分  
糜谢媛叔康攀

#### 2. 将指令拖至所需的位置。



#### 3. 松开鼠标按钮，在所需的位置放下指令。

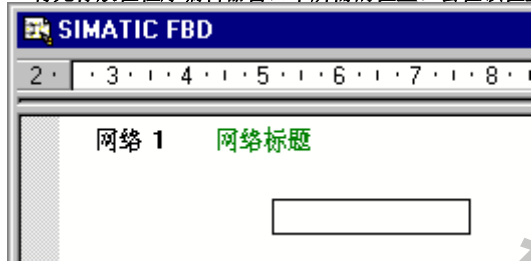


**注释：**光标会自动阻止您在非法的位置放下指令（例如首列、网络标题或另一条指令的参数）。

### 从指令树双击

[返回顶端](#)

1. 将光标放在程序编辑器窗口中所需的位置。会在该位置周围出现一个选项方框。



2. 在指令树中，浏览至所需的指令并双击该指令。



3. 双击指令后，指令出现在程序编辑器窗口中。



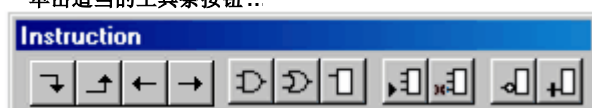
### 使用工具条按钮或功能键

[返回顶端](#)

1. 将光标放在程序编辑器窗口中所需的位置。会在该位置周围出现一个选择方框。



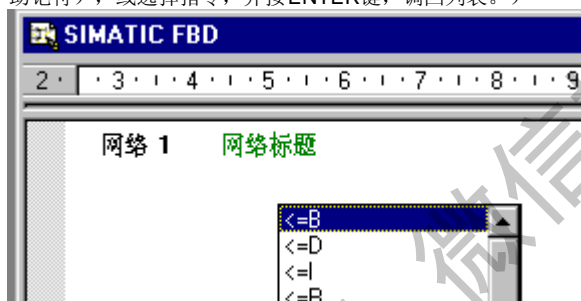
2. 单击适当的工具条按钮...



或使用适当的功能键（F4=AND（与）、F6=OR（或）、F9=通用方框）插入指令。

注释：不得将数值指定给类属指令框的参数；您必须如步骤3所示，选择一个具体的指令类型。

3. 会出现一个下拉列表。滚动或键入指令的前几个字母，浏览至所需的指令。双击所需的指令，或使用 ENTER 键，将其插入。（如果您这时不选择特定的指令种类，可以返回网络，单击类属指令的助记符区域（该区域包含 ???，而不是助记符），或选择指令，并按 ENTER 键，调回列表。）



## 1.5.5 4.5 如何在FBD中输入地址

指令参数的开始在FBD程序编辑器中用多种方法表示。不同的参数类型允许使用不同类型的赋值：

???

仅有位值赋值（例如，绝对或符号位址或布尔值，例如 OFF（关闭）或 ON（打开））对此类输入或输出参数有效。您可以反转此类输入，将此类输入和 / 或输出变成立即参数。

???

此类输入或输出参数不接受位值赋值；因为有效的数据类型根据指令不同而异，按 F1 参阅有效数据类型列表。

<<

您可以将该输入参数与前一条指令连接，或者为该输入参数指定一个位值赋值。您可以反转此类输入；您可以将此类输入转变

>>

您可以将该输出参数与后一条指令连接，或者为该输出参数指定一个位值赋值。您可以将此类输出转变成立即参数。

>|

您可以不为该 ENO 输出参数赋值，或将其与后一条指令连接，或者为该输出参数指定一个位值赋值。

当您第一次在程序中放置指令时，可以为指令参数指定数值 / 进行连接，或以后再完成该步骤。用红色表示的参数（<<、>>、??、?、????）必须分配一个数值或与另一条指令连接，才能使程序正确编译。（黑色 >| 输出参数（ENO 参数）可以保留不赋值状态。）

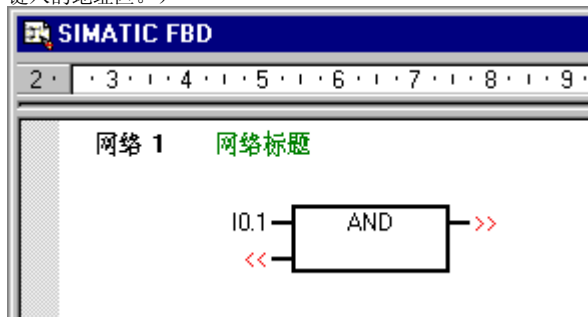
欲了解 FBD 程序编辑器如何显示非法 / 未定义参数值，请参阅 [程序编辑器如何在 FBD 中显示条目错误 \(GS 4.9\)](#)。



**提示：** 您可以选择工具（Tools）> 选项（Options）菜单命令，并选择放置指令后启用操作数编辑复选框，自动将光标放置在您需要输入的参数上。

**参数赋值**

欲指定一个常量值（例如100）或一个绝对地址（例如I0.1），只需在指令的地址区中键入所需的数值。（用滑鼠或ENTER键选择键入的地址区。）



欲指定一个符号地址（使用诸如INPUT1之类名称的全局符号或局部变量），必须执行下列简单的步骤：

1. 在指令的地址区中键入符号或变量名称。
2. **全局符号** 使用**符号表**将符号名分配给一个内存地址。

**注释：** 您不必预先定义符号，即可在程序中使用这些符号。您可以以后再定义内存地址。

**局部变量** 使用位于程序编辑器窗口顶端的**局部变量表**。在“符号”列输入符号名。因为编译程序自动分配L内存地址，您无需为植物淋渴凉氩 司貳D 梢允褂猛弦繁砀癖啾档姆椒ñ 植勒淋勒森某叽缙醯阶坏 使用局部变量是一种高级编程技术。无经验的程序员应考虑在符号表 / 全局变量表中将所有的符号值指定为全局符号。



嗟允尽

**提示：** 在输入地址值之后，程序编辑器自动将其格式化。您无需键入格式化字符；在完成键入后，这些字

**程序编辑器如何显示地址的范例**

I0.0	绝对地址由内存区和地址数目指定（SIMATIC程序编辑器）
%I0.0	在IEC中，百分号位于绝对地址之前（IEC程序编辑器）
#INPUT1	#符号位于局部变量之前（SIMATIC或IEC程序编辑器）
INPUT1	全局符号名（SIMATIC或程序编辑器）

**写入和强制地址**

在尝试写入或强制地址之前，必须打开程序状态。

欲写入或强制地址，用滑鼠右键单击操作数，并从滑鼠右键菜单选择“写入”或“强制”。





会显示一个对话框，允许您输入希望在PLC中写入或强制的数值。无论程序状态是否打开，这些选项均可供使用。



#### 匹配地址和定义符号

当您用滑鼠右键单击指令参数时，弹出菜单允许您在符号表中快速定义地址，或根据已经在地址区中键入的数值，最多选择五个可能的匹配项。



**提示：** • 使用ENTER键循环浏览所有的网络元素，每次查看一个参数，以便快速编辑所有必要的地址。  
• 另一种方法是用滑鼠右键单击个别参数，选择该参数，并定义符号或使用弹出菜单为这些地址查找匹配项。

#### 有效和无效符号名

符号名允许包含数字字母字符和下划线，也允许包含扩展字符（ASCII 128至ASCII 255）。首字符仅限于字母和扩展字符。

#### 有效名称：

a11  
a\_b\_1\_2

懒糖愚惚\_1284938

#### 非法名称：

lloop  
l:kdl";ld

非法名称是以一个数字开始或包含非数字字母或扩展字符集的字符。

#### 另请参阅：

[编址 \(GS 2.2\)](#)

[常量](#)

[选择助记符集](#)

[子例行程序](#)

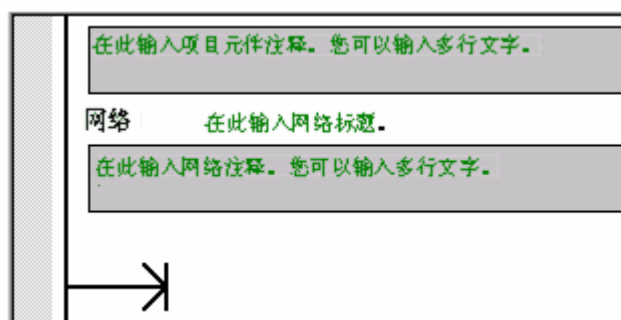
[中断](#)

[SIMATIC和IEC 1131-3数据类型](#)

## 1.5.6 4.6 如何在FBD中输入程序注释


FBD编辑器中共有四个注释级别：

- 项目元件注释
- 网络标题
- 网络注释
- 项目元件属性



### 项目元件注释

在网络1上方的灰色方框中单击，输入POU注释。


您可以单击“切换POU注释”按钮  或选择和取消选择 **检视 (View) > POU注释 (POU Comments)** 选项，在POU注释ON (打开) (可视) 或OFF (关闭) (隐藏) 之间切换。每条POU注释允许使用的最大字符数为4096。POU注释为供选用项目，可视时，始终位于POU的顶端，在第一个网络之前显示。

### 网络标题

将光标放在网络标题行中的任何位置，输入一个识别逻辑网络的标题。网络标题中允许使用的最大字符数为127。

### 网络注释

在网络1下方的灰色方框中单击，输入网络注释。您可以输入一条识别逻辑网络的注释，并输入一条有关网络内容的说明。

您可以单击“切换网络注释”按钮  或选择和取消选择 **检视 (View) > 网络注释 (Network Comments)** 选项，在网络注释ON (打开) (可视) 或OFF (关闭) (隐藏) 之间切换。网络注释中允许使用的最大字符数为4,096。

### 项目元件属性

您可以用以下任意一种方法存取“属性”标记：

用滑鼠右键单击指令树中的 **POU**，并从滑鼠右键菜单中选择“属性”。



用滑鼠右键单击程序编辑器窗口中的任何POU标记，并从弹出菜单选择“属性”。



在“属性”对话框中有两个标记：

- 一般
- 保护



#### 一般标记

“一般”标记允许您为子例行程序、中断例行程序和主程序块（OB1）重新编号和重新命名，并为项目指定一个作者。

#### 注释：

您不得将默认名称（由STEP 7-Micro/WIN指定的POU地址，例如SBR1用于子例行程序，或INT1用于中断例行程序）用作符号名，因为这样会构成重复赋值。如果您违反了符号名赋值的指南，STEP 7-Micro/WIN会在您尝试编译程序时报告一则错误。

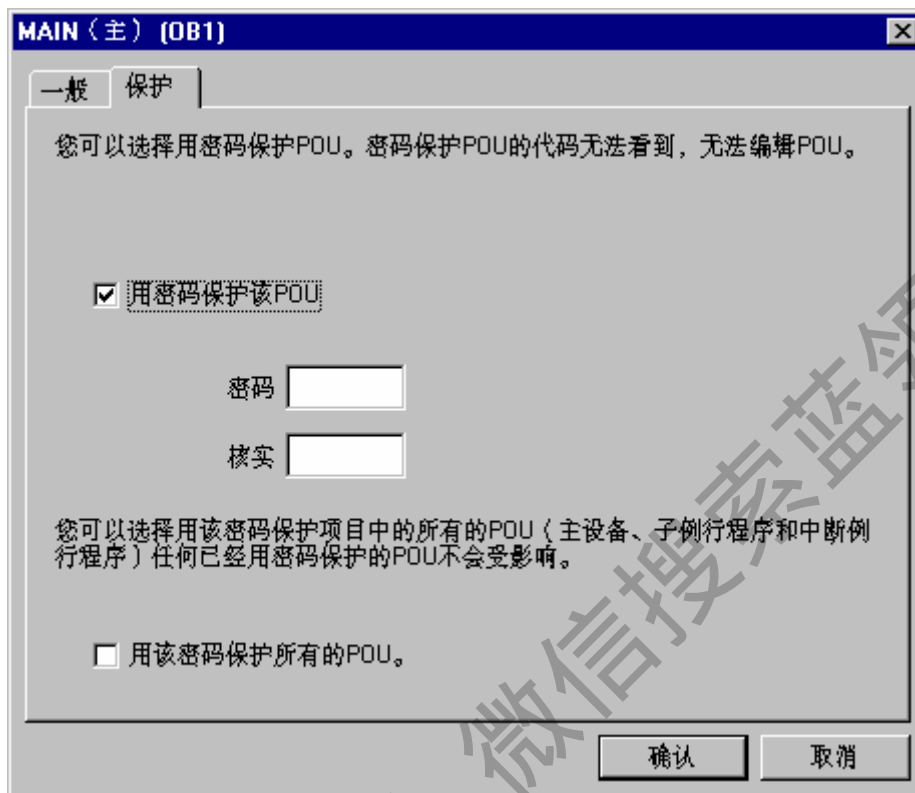
如果您在程序中为POU指定符号名，符号名会在程序代码中显示，即使“符号编址”视图未启用亦如此。符号表显示一个列出所有符号名赋值的特别标记（“POU符号”）。您只能检视该标记，无法编辑符号表中的条目。欲改动赋值，您必须编辑适当POU的“属性”对话框。

#### 保护标记

“保护”标记允许您选择用密码保护POU，使其他用户无法看到，并在下载时加密。

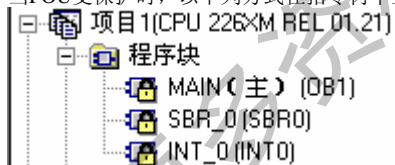
欲用密码保护POU:


1. 用滑鼠右键单击“主POU”，并选择“属性...”
2. 选择“保护”标记。
3. 选择“用密码保护该POU”复选框。输入并核实由四个字符组成的密码。



您还可以选择“用密码保护所有POU”复选框，保护项目中的所有POU（主程序、子例行程序和中断例行程序）。

当POU受保护时，以下列方式在指令树中显示:



此外，还会在程序编辑器窗口的装订线中显示一个灰色锁定图标 .

欲取消锁定POU:

1. 浏览至“保护”标记。
  2. 输入密码。
  3. 单击“授权”。
- 您必须为您选择保护的每个POU重复该步骤。

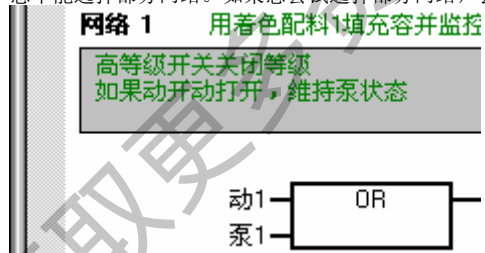


### 1.5.7 4.7 如何在FBD中编辑程序元素

#### 剪切、复制、粘贴或删除多个网络

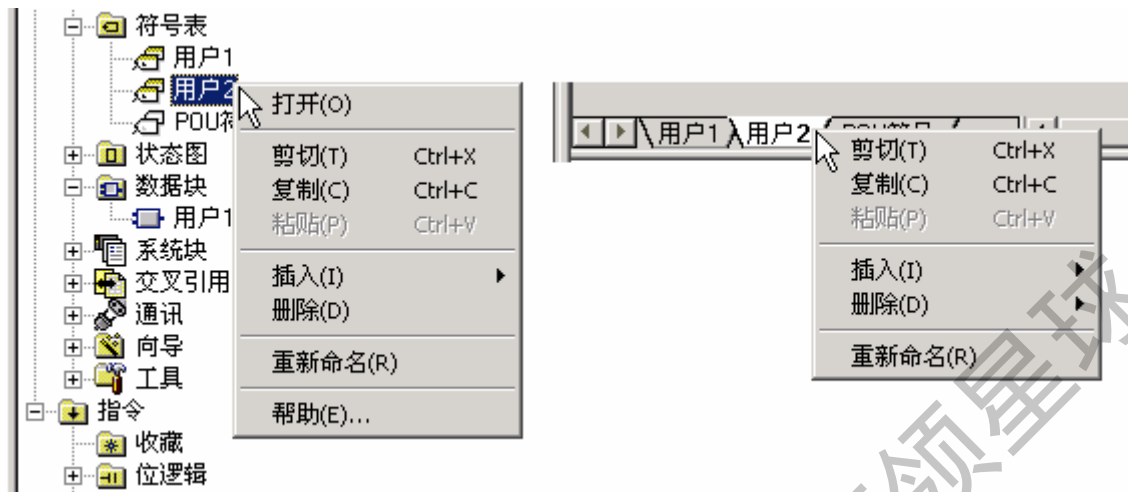
使用拖曳滑鼠或SHIFT键和上、下箭头键，可选择多个相邻的网络，然后剪切、复制、粘贴或删除选项。利用工具条按钮，从“

如下图所示，将鼠标移到编辑器的左侧边缘（灰条区域），然后单击以开始对多个网络进行拖过选择。您不能选取部分网络。您不能选择部分网络。如果您尝试选择部分网络，会自动选择整个网络。



#### 剪切、复制、粘贴项目元件

如下图所示，将鼠标移到指令树或编辑器标记上，然后单击鼠标右键。由弹出菜单中选取“复制”命令，以复制整个项目元件。

**注释:**

“复制”功能只在您选择了一个需要复制的项目后才能使用。您作出的选择确切描述了将复制到Windows剪贴板缓冲器的内容。

您可以在项目中选择以下目标

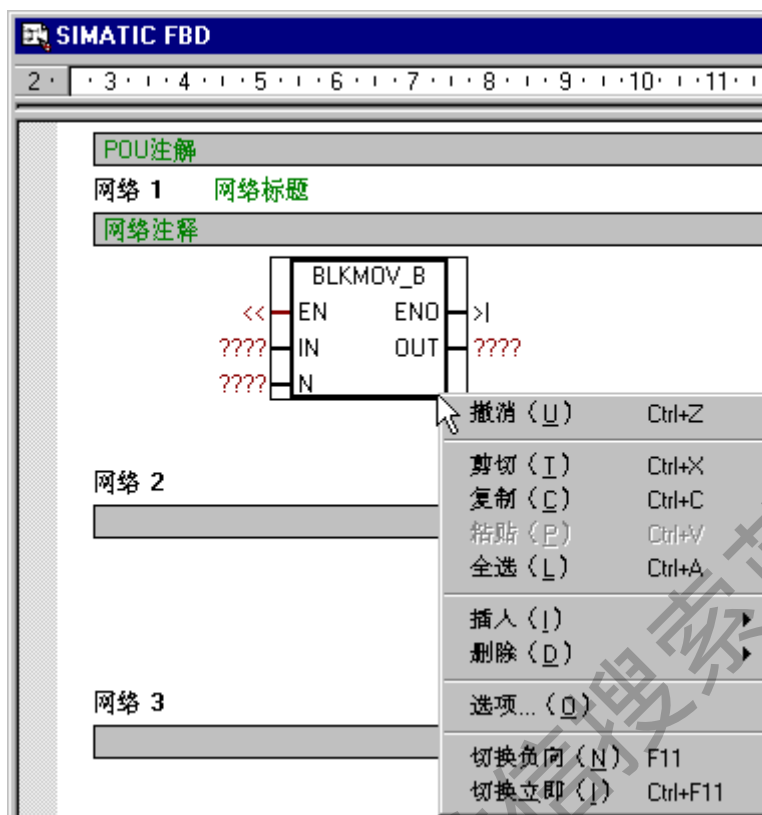
- 程序文字或数据域
- LAD、FBD、STL等编辑器中的指令
- 单个网络：将鼠标放在“网络”的左侧边缘上，然后单击
- 多个相邻网络：在选择第一个网络后，使用Shift+下箭头/上箭头或Shift+Page Up/Page Down来扩大或减少所包括的网络数目
- 某POU内的所有网络：请使用 **编辑 >**

**全选** 菜单命令，或用鼠标右键单击某指令树POU分支，或用鼠标右键单击某标记名

- 选定的数据块文字，或整个数据块标记页：  
当通过用鼠标右键单击编辑器标记名而复制整个数据块标记页时，随后的粘贴操作能够输出用制表符分隔的数据。举例来说，倘若将此数据块文字粘贴到MS Excel，则电子表格的多个单元行将得到填充。
- 选定的符号表行或列，或整个符号表标记页
- 选定的状态图行或列，或整个状态图标记页

**编辑单元格、指令、地址和网络**

- 当您单击程序编辑器中的空单元格时，会出现一个方框，显示您选择的单元格。您可以使用弹出菜单将选项粘贴进空单元格。
- 当您单击一条指令时，会在指令周围出现一个方框，显示您选择的指令。您可以使用弹出菜单在指令上全选、撤销键

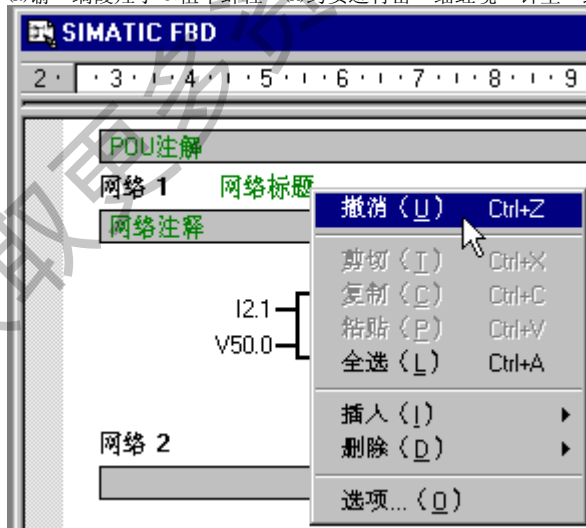


- 当您单击一个指令参数时，会在域周围出现一个方框，显示您选择的参数。您可以使用弹出菜单撤消键入、剪切、复制、粘贴、全选、插入、删除、选项...、切换负向、切换立即。





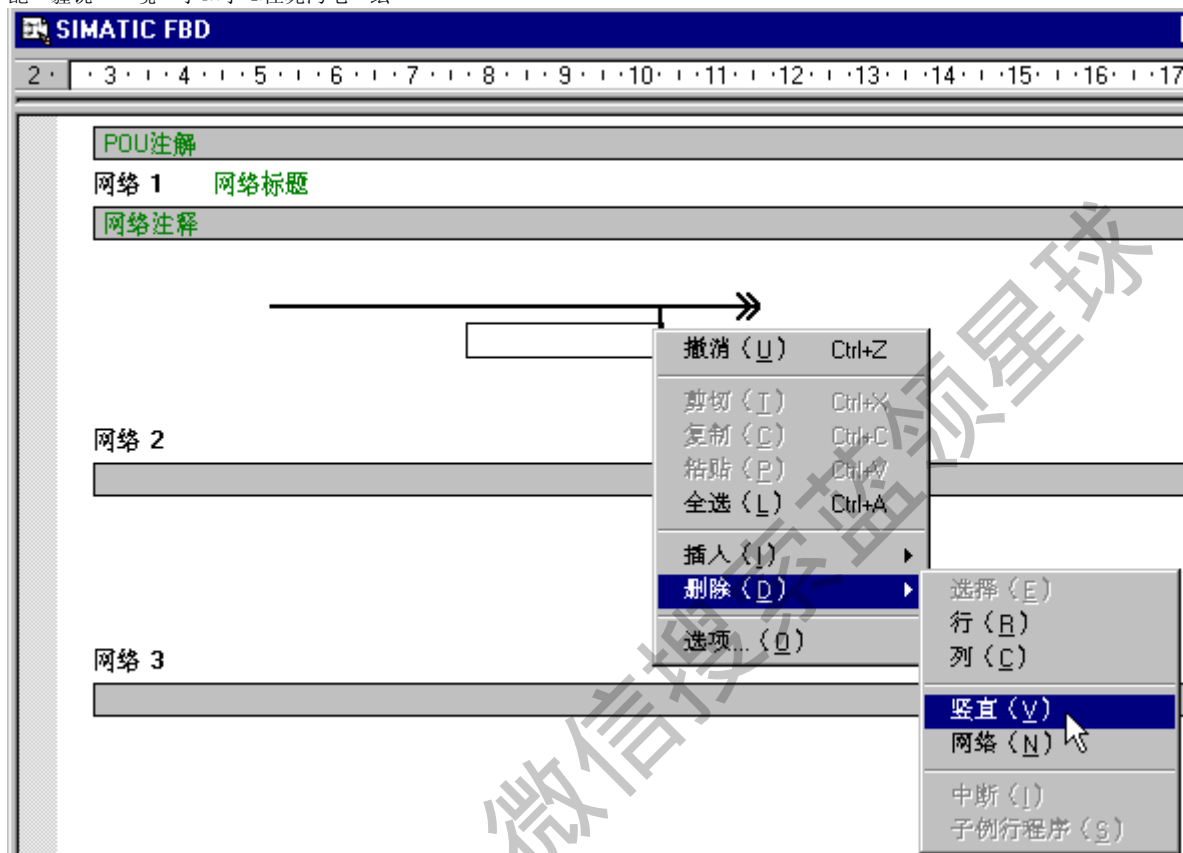
- 当您在网络标题行中单击时，您可以编辑项目的标题。您可以使用弹出菜单全选、撤销键入、在整个网络中执行编辑



您还可以使用工具条按钮、标准窗口控制键和“编辑”菜单剪切、复制或粘贴选项。

#### 删除元素

您可以使用DELETE（删除）或BACKSPACE（退格）键删除单个单元格；您可以使用“编辑”菜单或用鼠标右键单击的方法调整



**注释：** 为了正确选择用作删除的垂直行，始终将光标放在垂直行左边的单元格上。

另请参阅：  
[快捷键 \(GS 1.2\)](#)

## 1.5.8 4.8 如何使用查找、替换和转入功能

欲使用查找 / 替换或转入：

- 选择**编辑 (Edit)** > **查找 (Find)**、**编辑 (Edit)** > **替换 (Replace)** 或 **编辑 (Edit)** > **转入 (GoTo)** 菜单命令
- 按CTRL+F执行查找，按CTRL+H执行替换或按CTRL+G执行转入

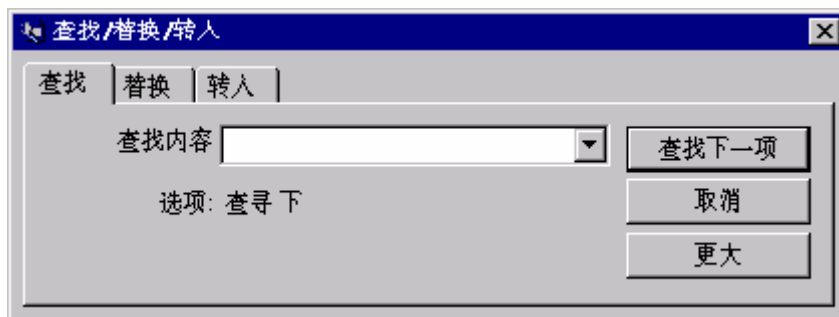
如何使用：  
[查找](#)  
[替换](#)

在何处使用  
可以在程序编辑器窗口、局部变量表、符号表、状态图、交叉引用标记和数据块中使用“查找”、“替换”和“转入”。

如何作业

- “查找”功能允许您查找指定的字符串，例如操作数、网络标题或指令助记符。（“查找”不搜索网络注解，仅搜索符号表！安椅摇翠凰阁影 LAD和FBD中提供的网络符号信息表。）
- “替换”功能允许您替换指定的字符串。（“替换”不对指令助记符操作。）
- “转入”功能允许您通过指定网络数目或您希望浏览的行迅速移至另一个位置。

[查找功能](#)  
[返回顶端](#)



1. 在“查找内容”域中键入要搜索的字符串。

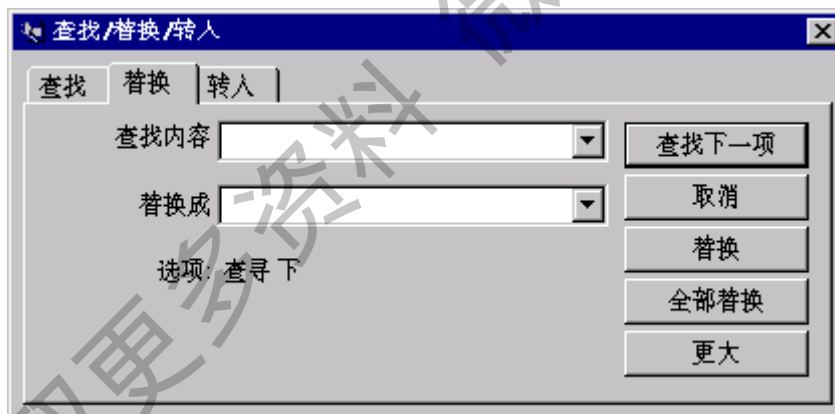
2. 欲移动下一个搜索字符串，单击“查找下一项”按钮。

**注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反映了操作数在代码中的存储方式。另外，如果您在 STL 中建立的程序包含在 LAD 或 FBD 编辑器中非法的网络，在 LAD 或 FBD 编辑器中查看该程序时，“查找下一项”命令不对这些网络执行搜索操作。

欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方式定制搜索：

- 您可以使用“搜索”列表框选择搜索方向。
- 您可以选择“大小写匹配”复选框，仅搜索与您在“查找内容”中键入字符串大小写数值相同的字符串。
- 您可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的 SPAN（扩展）复选框，对所有的 POU（OB1、所有的子例行程序和中断例行程序）或局部变量表的所有实例、符号表或状态图进行搜索。
- 您可以指定对一定的行范围进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认段 B 荒 部梢约 肚 缙蚰泻牛 魑 闹鞞钠鹦己徒峨
- 您可以指定是否通过选择适当的复选框搜索网络标题、POU 和网络注释和 / 或程序代码。

#### 替换功能 返回顶端



1. 在“查找内容”域中键入要搜索的字符串。

2. 在“替换内容”域中键入您希望用作替换搜索字符串的字符串。

3. 欲查找出现的搜索字符串，单击“查找下一项”按钮。

**注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反映了操作数在代码中的存储方式。另外，如果您在 STL 中建立的程序包含在 LAD 或 FBD 编辑器中非法的网络，在 LAD 或 FBD 编辑器中查看该程序时，“查找下一项”命令不对这些网络进行搜索操作。

4. 如果您希望替换字符串，单击“替换”。如果您已经仔细地定义搜索字符串而且没有误改的可能，您可以单击“全部替换”。

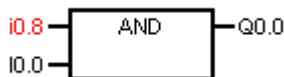
欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方法定制搜索：

- 您可以使用“搜索”列表框选择搜索方向。
- 您可以选择“大小写匹配”复选框，仅搜索与您在“查找内容”中键入字符串大小写数值相同的字符串。
- 您可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的 SPAN（扩展）复选框，对所有的 POU（OB1、所有的子例行程序和中断例行程序）或局部变量表的所有实例、符号表或状态图进行搜索。
- 您可以指定对一定的行范围进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认

- 您可以指定是否通过选择适当的复选框搜索网络标题、POU和网络注释和 / 或程序代码。

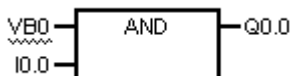
## 1.5.9 4.9 程序编辑器如何在FBD中显示输入错误

红色文字表示非法语法。

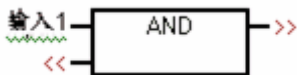


注释：当您用一个有效数值替换非法地址数值时，字体自动更改为默认字体颜色（黑色，除非您已定制窗口）。

一条红色波浪线位于数值下方，表示数值超出范围或对此类指令不正确。



一条绿色波浪线位于数值下方，表示正在使用的变量或符号尚未定义。STEP 7-Micro/WIN允许您在定义变量和符号之前写入程序。您可以在任何时间把数值增加至局部变量表或符号表中。



**提示：** • 用鼠标右键单击，调出与全局符号最接近的五个匹配菜单。 • 使用“定义符号”对话框，解决未定义的符号。

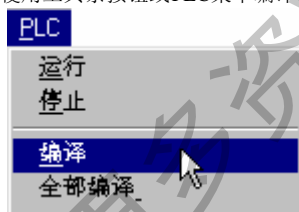
用鼠标右键单击，调出与全局符号最接近的五个匹配菜单。 •



使用“定义符号”对话框，解决未定义的符号。

## 1.5.10 4.10 如何在FBD中编译

如何编译

您可以使用工具条按钮或PLC菜单编译。



- “编译”  允许您编译项目的单个元素。在您选择“编译”时带有焦点的窗口（程序编辑器、系统块或数据块）是编译的目标。
- “全部编译”  对程序编辑器、系统块和数据块进行编译。当您使用“全部编译”命令时，哪一个窗口是焦点无关紧要。

使用输出窗口解决错误

编译时，“输出窗口”列出发生的所有错误。错误根据位置（网络、行和列）以及错误类型识别。您可以双击错误行，调出程序编辑器中包含错误的代码网络。

另请参阅：

[Micro/WIN编译程序错误](#)

## 1.5.11 4.11 如何保存作业

您可以使用工具条上的“保存”按钮  保存作业，或从“文件”菜单选择“保存”和“另存为”选项保存作业。



- “保存”允许您在作业中快速保存所有改动。（然而，您初次保存项目时，会被提示核实或修改当前项目名称和目录哪一位置。）
- “另存为”允许您修改当前项目的名称和 / 或目录位置。

当您首次建立项目时，STEP 7-Micro/WIN的默认值名称为“Project1.mwp”。您可以接受或修改该名称；如果您接受该名称，下一个项目的默认名称将自动递增为“Project2.mwp”。STEP 7-Micro/WIN项目的默认目录位置是位于“Microwin”目录中的称作“项目”的文件夹，您可以不接受该默认位置。

## 1.6 第5章 如何输入语句列表程序

### 1.6.1 5.1 如何建立项目

#### 打开新项目

双击STEP 7-Micro/WIN图标或从“开始”菜单选择Simatic>STEP 7-Micro/WIN，启动应用程序。打开新的STEP 7-Micro/WIN项目。

#### 打开现有项目

在STEP 7-Micro/WIN中，使用文件菜单并选择下列选项之一：

- **打开**—允许您浏览至现有项目并打开该项目。
- **文件名称**—如果您最近在某个项目中作业，则该项目会在“文件”菜单中列出，您可以直接选择该项目，无需使用“**最近**”列表。

您也可以使用Windows Explorer浏览至适当的目录并打开项目，无需事先单独启动STEP 7-Micro/WIN。在STEP 7-Micro/WIN 3.0版或更高版本中，项目包含在单个文件中，带有扩展名.mwp。



提示：一旦打开一个项目，您可以开始编写程序。开始之前，您可能需要执行一项或多项下列任务：

**范围—根据PLC类型检查** 为了让STEP 7-Micro/WIN检查参数的范围，您可以在编写程序之前**选择PLC类型**。（如果您已经为项目指定PLC类型，指令树用红色标记x表示对您的PLC无效的任何指令。）

您可以**现在设定通讯**，或等到下载程序就绪再设定。您可以多种不同的方法**定制工作区域**。

**设定通讯**

**定制工作区域**

### 1.6.2 5.2 如何在STL中输入语句

#### 步骤：

您可以在程序编辑器窗口中键入STL指令或用从指令树拖放的方式插入STL指令。

欲在STL编辑器中键入语句：

1. 将光标放在一行的开始。
2. 输入指令。（例如：LD）
3. 按SPACEBAR（空格键）或TAB（制表符）键。
4. 输入操作数，操作数可以是绝对值（例如：I0.0）、符号（例如：Input1）或常量（例如：100）。带指令和地址的整行范例：LD I0.0

**注释：**如果一行有多个操作数，您可以使用空格、制表符或逗号将其分开。

5. 如果无须注解，按ENTER键，移至下一行。

请勿使用MEND、RET、RETI、SBR或INT终止程序的POU（OB1、子程序、中断例行程序）。这些指令由编译程序自动插入

蝗继 POU中使用这些指令，编译程序返回一则错误。

欲从指令树拖放：

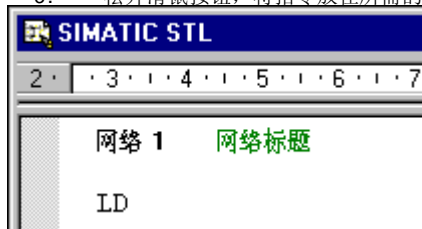
1. 选择指令。



2. 将指令拖拉至所需的位置。



3. 松开滑鼠按钮，将指令放在所需的位置。



**注释：**光标会自动阻止您将指令放在非法的位置（例如，放在网络  
空行 或另一条指令的参数上）。

如果您将指令放在一行上，该行会向下移动，将新指令放在该行原来的位置。

**指南：**

- 每行仅输入一个指令和 / 或注解。
- 按INSERT（插入）键，在键入时在“插入模式”和“覆盖模式”之间切换。“插入模式”是默认值。
- 欲分隔程序中的网络，您可以使用位于构成网络的指令开始前的一行的关键字 NETWORK（网络）。（请勿键入网络  
怕毫坏蹦 喊胧被嶙远 崑雍怕瘠）

举例：

```
Network 1 //该行介绍程序的第一个网络
LD      I0.0 //在I0.0载入数值
O       I0.1 //或带有I0.1数值
//此处输入更多指令：每个网络由多个指令组成
Network 2 //开始新的指令网络
```

分隔网络为供选用选项；该选项使查看配备 LAD或FBD程序编辑器的STL程序更加简便。

- 当程序超过一个屏蔽时，使用滚动条或PAGE UP/DOWN（上页 / 下页）键查看程序。

**另请参阅：**

[地址简介 \(GS 2.2\)](#)  
[如何在STL中输入程序注解 \(GS 5.3\)](#)  
[语行列表程序范例 \(GS 5.4\)](#)

### 1.6.3 5.3 如何在STL中输入程序注解

**程序注解**

所有STL中的注解前均须有双斜线：//

您可以将一整行作为注解；您可以把注解作为一行放在有效指令之后。您不能创建行中注解；双斜线之后的所有内容均被编译程序视作注解。

最大行长（包括注解）为255个字符。

举例：

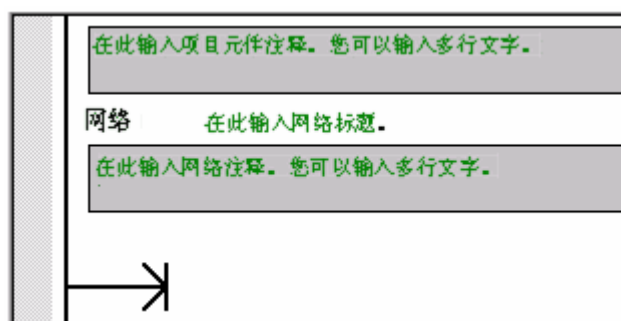
```
Network 1 //本行介绍程序的第一个网络
LD      I0.0 //在I0.0处载入数值
O       I0.1 //或带有I0.1数值
```

//此处输入更多指令：每个网络由多个指令组成

Network 2 //开始新的指令网络


在STL编辑器中共有四个注解级别：

- 项目组件注解
- 网络标题
- 网络注解
- 项目组件属性



#### 项目组件注解

在网络1上方的灰色方框中单击，输入POU注解。


您可以单击“切换POU注解”按钮或选择和取消选择**检视 (View) > POU注解 (POU Comments)**选项的方法，在POU注解“打开”（可见）或“关闭”（隐藏）之间切换。每条POU注解最多可使用4096个字符。POU注解是选用项目，显示时始终位于POU顶端，并在第一个网络之前显示。

#### 网络标题

将光标放在网络标题行中的任何位置，输入一个可识别该逻辑网络的标题。网络标题最多可使用 127个字符。

#### 网络注解

在网络1下方的灰色方框中单击，输入网络注解。您可以输入一条识别该逻辑网络的注解，并增加有关网络内容的说明。

您可以单击“切换网络注解”按钮或选择和取消选择**检视 (View) > 网络注解 (Network Comments)**选项的方法，在网络注解“打开”（可见）或“关闭”（隐藏）之间切换。网络注解中最多可使用4096个字符。

#### 项目组件属性

您可以以下任何一种方法存取“属性”制表符：

用鼠标右键单击指令树中的 **POU**，并从鼠标右键菜单中选择“属性”。

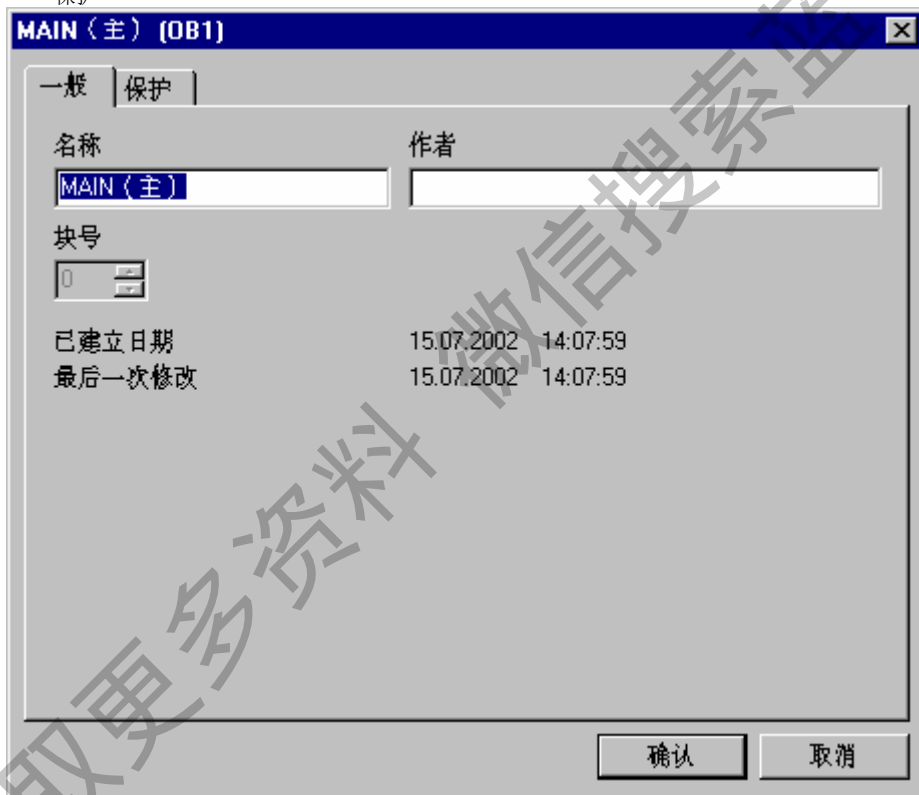


用鼠标右键单击程序编辑器窗口中的 **POU** 制表符，并从弹出菜单中选择“属性”。



“属性”对话框中有两个制表符：

- 一般
- 保护



#### “一般”制表符

“一般”制表符允许您为子例行程序、中断例行程序和主程序块（OB1）重新编号和重新命名，并为项目指定一个作者。

#### 注释：

您不能将默认名称（由STEP 7-Micro/WIN指定的POU地址，例如，SBR1代表子例行程序或INT1代表中断例行程序）用作符号名，因为这将构成重复。STEP 7-Micro/WIN会报告一则错误。

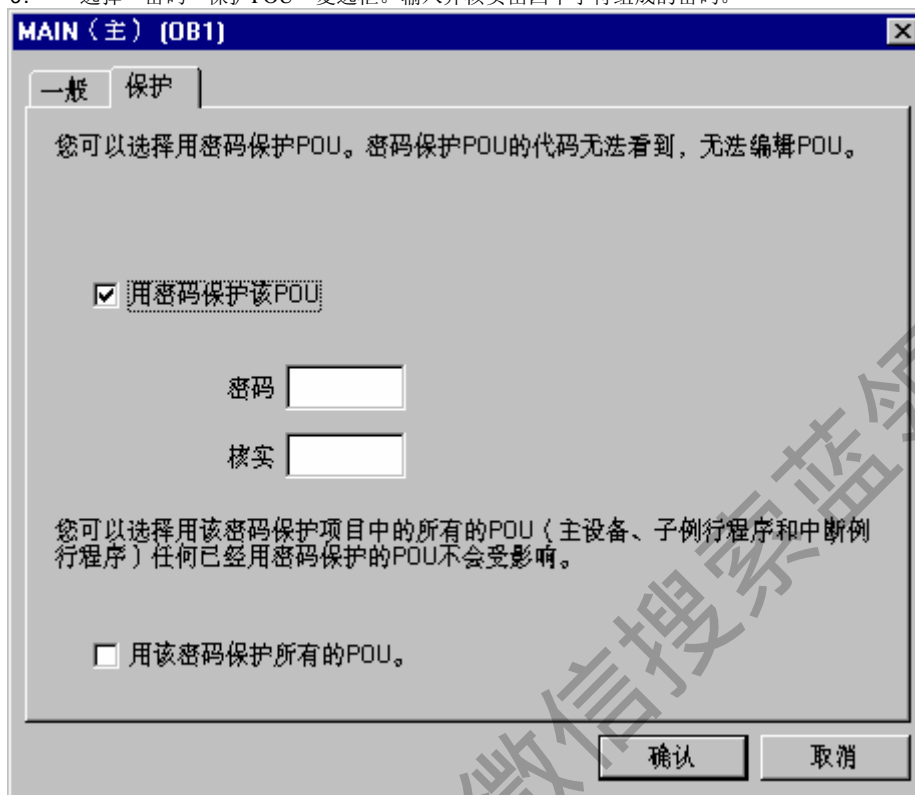
如果您为程序中的POU指定符号名，符号名会在程序代码中显示，即使您没有启用“符号编址”视图亦如此。“符号表”显示一个特殊的制表符（“POU符号”），该制表符列出所有符号名赋值。您只能检视该制表符，但无法从“符号表”编辑条目。欲更改赋值，您必须编辑适当POU的“属性”对话框。

#### “保护”制表符

“保护”制表符允许您选择用密码保护POU，以便使其他用户无法看到POU，并在下载时加密。  
欲用密码保护POU：



1. 用滑鼠右键单击“主POU”，并选择“属性...”。
2. 选择“保护制表符”。
3. 选择“密码—保护POU”复选框。输入并核实由四个字符组成的密码。



您还可以选择保护项目中的所有POU（主、子例行程序和中断例行程序），方法是选择“密码—保护所有POU”。

POU受到保护后，则会以以下方式在指令树中显示：



此外，会在程序编辑器窗口中显示一个灰色锁定图标 .

欲为POU解除锁定：

1. 浏览至“保护制表符”。
  2. 输入密码。
  3. 单击“授权”。
- 您必须为选择保护的每个POU重复该步骤。



### 1.6.4 5.4 语句列表程序举例

下面列举的程序显示根据 [如何在STL中输入语句](#) 中的指南在STL编辑器中制定格式。您可能希望以类似的方式设定程序，由网络注解对网络的功能作出一般概述，由语句注解识别语句的功能。请注意示范程序使用了符号地址。

如果您希望在STEP 7-Micro/WIN中看到该程序（用STL或LAD），选择**文件 > 打开**，浏览至Microwin\Samples\English目录，增亮示范程序，并单击“打开”按钮。

#### 油漆混合器示范程序

```

NETWORK 1 //用1号成分装满储罐并监视储罐水平。
//如果“起始”开关打开，保持泵原有状态。
LD Start_1 //载入Start_1 I0.0数值。
O Pump_1 //或带有Pump_1 Q0.0数值。
A Stop_1 //并且带有Stop_1 I0.2。
AN High_Level //并且不带有High_Level I0.4。
= Pump_1 //将结果指定给Pump_1 Q0.0。
NETWORK 2 //用油漆成分2装满储罐并监视储罐。
//位置。如果“起始”开关打开，保持泵原有状态。
LD Start_2 //载入Start_2 I0.1数值。
O Pump_2 //或带有Pump_2 Q0.1数值。
A Stop_2 //并且带有Stop_2 I0.3。
AN High_Level //并且不带有High_Level I0.4。
= Pump_2 //将结果指定给Pump_2 Q0.1。
NETWORK 3 //如果达到高位，设定内存位。
LD High_Level //载入High_Level I0.4数值。
S High_Lev_Reached, 1 //将High_Lev_Reached M0.1设定为1（打开）。
NETWORK 4 //如果达到高位，设定起始定时器。
LD High_Lev_Reached //载入High_Level Reached M0.1数值。
TON Mix_Timer, +100 //定时器37，预设 = 100 (@0.1 s)

```

```

NETWORK 5 //打开混合器马达。
LDN Mix_Timer //不载入Mix_Timer T37数值。
A High_Lev_Reached //并且带有High_Lev_Reached M0.1。
= Mixer_Motor //将结果指定给Mixer_Motor Q0.2。
= Steam_Valve //将结果指定给Steam_Valve Q0.3。
NETWORK 6 //排空油漆混合储罐。
LD Mix_Timer //载入Mix_Timer T37数值。
AN Low_Level //并且不带有Low_Level I0.5。
= Drain_Valve //将结果指定给Drain_Valve Q0.4。
= Drain_Pump //将结果指定给Drain_Pump Q0.5。
NETWORK 7 //为每次循环计数。
LD Low_Level //载入Low_Level I0.5数值。
A Mix_Timer //以及Mix_Timer T37数值。
LD Reset //载入Reset I0.7数值。
CTU Cycle_Counter, +12 //计数器30, 预设 = 12。
NETWORK 8 //如果达到“低位”, 重设内存位, 计时器超时。
LD Low_Level //载入Low_Level I0.5数值。
A Mix_Timer //以及Mix_Timer T37数值。
R High_Lev_Reached, 1 //将High_Lev_Reached M0.1数值重设为0。

```

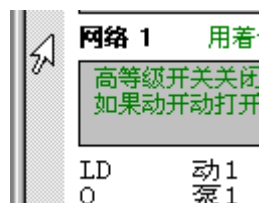
## 1.6.5 5.5 如何编辑STL程序

### 剪切、复制、粘贴或删除多个网络

您可以通过拖过鼠标或将SHIFT键与上下箭头键合用, 选取多个相邻网络。通过使用工具条按钮、选取“编辑”菜单中的命令或一

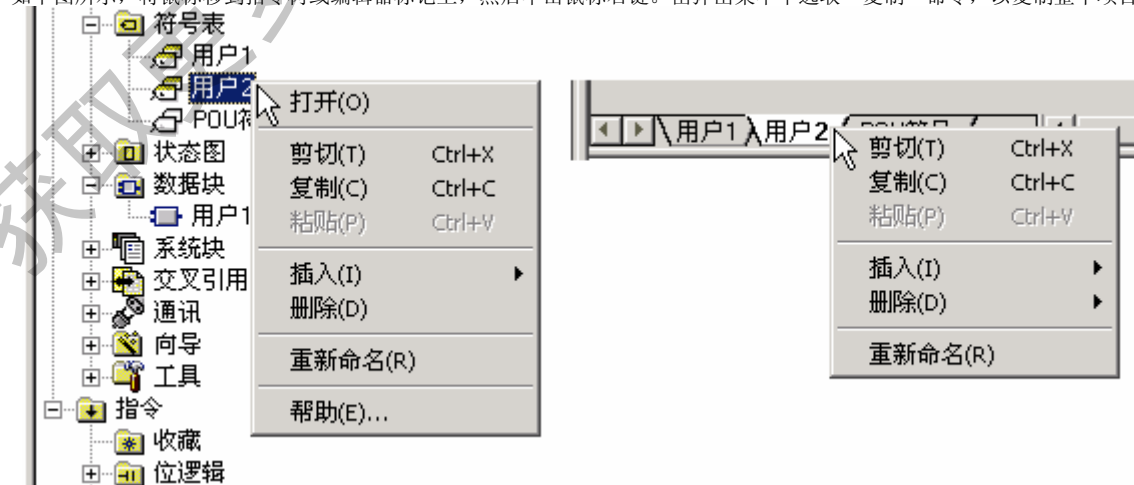
如下图所示, 将鼠标移到编辑器的左侧边缘

(灰条区域), 然后单击以开始对多个网络进行拖过选择。您不能选取部分网络。如果您试图选取部分网络, 整个网络会被自动选



### 剪切、复制、粘贴项目元件

如下图所示, 将鼠标移到指令树或编辑器标记上, 然后单击鼠标右键。由弹出菜单中选取“复制”命令, 以复制整个项目元件。



注释:

“复制”功能只在您选择了一个需要复制的项目后才能使用。您作出的选择确切描述了将复制到Windows剪贴板缓冲器的内容。

您可以在项目中选择以下目标

- 程序文字或数据域
- LAD、FBD、STL等编辑器中的指令
- 单个网络：将鼠标放在“网络”的左侧边缘上，然后单击
- 多个相邻网络：在选取第一个网络后，使用Shift+下箭头/上箭头或Shift+Page Up/Page Down来扩大或减少所包括的网络数目
- 某POU内的所有网络：请使用**编辑 > 全选**菜单命令，或用鼠标右键单击某指令树POU分支，或用鼠标右键单击某标记名
- 选定的数据块文字，或整个数据块标记页；  
当通过用鼠标右键单击编辑器标记名而复制整个数据块标记页时，随后的粘贴操作能够输出用制表符分隔的数据。举例来说，倘若将此数据块文字粘贴到MS Excel，则电子表格的多个单元行将得到填充。
- 选定的符号表行或列，或整个符号表标记页

语句表（STL）是一种基于文本的编程语言。您可以使用下列键 / 键组合更快速、更高效地编辑程序。



**提示：** • 如果列出多个键，所有的键必须同时按下，才能使快捷键工作。 • 您

#### 键序列

#### 措施

CTRL + A	全选
CTRL + C	拷贝选项
CTRL + F	查找字符串
CTRL + H	替换字符串
CTRL + V	粘贴选项
CTRL + X	剪切选项
CTRL + Z	撤销
SHIFT + Z	重复
DELETE	删除选项
SHIFT + DELETE	剪切选项至剪贴板
F2	转入下一个书签位置
CTRL + F2	在当前光标位置设定书签
CTRL + F3	插入网络
CTRL + F4	删除网络
F3	查找下一个字符串
BACKSPACE	退格
SHIFT + BACKSPACE	退格
ALT + BACKSPACE	撤销
INSERT	切换盖写
CTRL + INSERT	复制选项
SHIFT + INSERT	粘贴选项
HOME	转入当前行开始
CTRL + HOME	转入文件顶端
SHIFT + HOME	从当前光标位置选择至同一行开始
CTRL + SHIFT + HOME	从光标位置向上选择至文件顶端
END	转入当前行结尾
CTRL + END	转入文件底部
SHIFT + END	从当前光标位置选至同一行结尾
CTRL + SHIFT + END	从光标位置向下选至文件底部
LEFT ARROW	光标左移，一次移动一个字符
CTRL + LEFT ARROW	光标左移，一次移动一个字
SHIFT + LEFT ARROW	向左选择，一次选择一个字符
CTRL + SHIFT + LEFT ARROW	向左选择，一次选择一个字
RIGHT ARROW	光标右移，一次选择一个字符
CTRL + RIGHT ARROW	光标右移，一次选择一个字
SHIFT + RIGHT ARROW	向右选择，一次选择一个字符

CTRL + SHIFT + RIGHT ARROW	向右选择，一次选择一个字
UP ARROW	光标上移，一次移动一行
SHIFT + UP ARROW	向上选择，一次选择一行
DOWN ARROW	光标下移，一次移动一行
SHIFT + DOWN ARROW	向下选择，一次选择一行
PAGE DOWN	光标下移一个屏幕的数据
SHIFT + PAGE DOWN	从当前光标位置向下选择一个屏幕的数据
PAGEUP	光标上移一个屏幕的数据
SHIFT + PAGE UP	从当前光标位置向上选择一个屏幕的数据
TAB	用制表符向前跳格
SHIFT + TAB	用制表符向后跳格

## 1.6.6 5.6 如何使用查找、替换和转入功能

欲使用“查找”/“替换”或“转入”：

- 选择**编辑 > 查找 (Edit>Find)**、**编辑 > 替换 (Edit>Replace)** 或 **编辑 > 转入 (Edit>GoTo)** 菜单命令
- 按CTRL+F执行查找，CTRL+H执行替换或CTRL+G执行转入

**如何使用：**

[查找](#)  
[替换](#)

**在何处使用**

可以在程序编辑器窗口、局部变量表、符号表、状态图表、交叉引用标签和数据块中使用“查找”、“替换”和“转入”。

**如何使用**

- “查找”功能允许您查找指定的字符串，例如操作数、网络标题或指令助记符。（“查找”不搜索网络注解，仅搜索符号？）
- “替换”功能允许您替换指定的字符串。（“替换”不对指令助记符操作。）
- “转入”功能允许您通过指定网络号码或您希望浏览至的行快速移至另一个位置。

**查找功能**  
[返回顶端](#)



1. 在“查找内容”栏中键入要搜索的字符串。

2. 欲移至下一个搜索字符串，单击“查找下一项”按钮。

**注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反应了操作数在代码中的存储方式。另外，如果您在STL中创建的程序包含在LAD或FBD编辑器中非法的网络，在LAD或FBD编辑器中查看该程序时，“查找下一项”命令不对这些网络进行搜索操作。

欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方式定制搜索：

- 您可以使用“搜索”列表框选择搜索的方向。
- 您可以选择“大小写匹配”复选框，仅搜索与您键入的字符串大小写数值相同的字符串。
- 您可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的“SPAN”复选框，对所有的POU（OB1、所有的子程序和中断例行程序）或局部变量表的所有实例
- 7. 疵砵蜃刺 急斫 兴阂鳌
- 您可以指定对一部分行进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认范围

- 荒 部梢约 庐 缙蚰泻牛 魑 阉鞞钠鸫已徒岷
- 您可以指定是否通过选择适当的复选框搜索网络标题、POU和网络注解。

替换功能  
返回顶端



- 在“查找内容”栏中键入要搜索的字符串。
  - 在“替换内容”栏中键入您希望用作替换搜索字符串的字符串。
  - 欲查找存在的搜索字符串，单击“查找下一项”按钮。
- 注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反应了操作数在代码中的存储方式。另外，如果您在STL中创建的程序包含在LAD或FBD编辑器中非法的网络，在LAD或FBD编辑器中查看该程序时，“查找下一项”命令不对这些网络进行搜索操作。
- 如果您希望替换字符串，单击“替换”。如果您已经仔细地定义搜索字符串而且没有误改的可能，您可以单击“全部替换”。

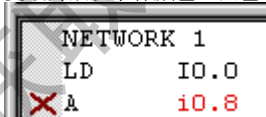
欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方法定制搜索：

- 您可以使用“搜索”列表框选择搜索的方向。
- 您可以选择“大小写匹配”复选框，仅搜索与您“查找内容”中键入字符串大小写数值相同的字符串。
- 您可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的“SPAN”复选框，对POU（OB1、所有的子程序和中断例行程序）或局部变量表、符号表或状态图
- 您可以指定对一部分行进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认范围
- 您可以指定是否通过选择适当的复选框搜索网络标题、POU和网络注解。

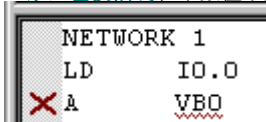
## 1.6.7 5.7 程序编辑器如何在STL中显示输入错误

一个X位于一行前，表示该行语法不完整或者不是有效行，不会以适当方式编译。

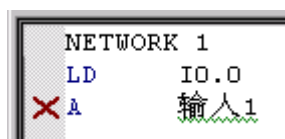
红色文字表示非法语法。（当您键入指令和地址时，自动变红，直至完成。）当您用一个有效数值替换非法数值时，字体自动更改为默认字体颜色（黑色，除非您已定制窗口）。



一条红色波浪线位于数值下方，表示数值超出范围或对此类指令不正确。



一条绿色波浪线位于数值下方，表示正在使用的变量或符号尚未定义。STEP 7-Micro/WIN允许您在定义变量和符号之前编写程序。您可以在任何时间把数值增加至局部变量表或符号表中。



**提示：** 用滑鼠右键单击，调出与全局符号最接近的五个匹配菜单。使用“定义符号”对话框，解决未定义的符号。

用滑鼠右键单击，调出与全局符号最接近的五个匹配菜单。

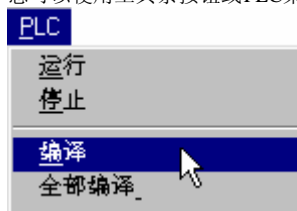
使用“定义符号”对话框，解决未定义的符号。

## 1.6.8 5.8 如何在STL中编译

在STL中，程序行前的X表示该行必须修改，否则不会对其编译。

### 如何编译

您可以使用工具条按钮或PLC菜单编译。



- “编译”  允许您编译项目的单个元素。在您选择“编译”时带有焦点的窗口（程序编辑器、系统块或数据块）是编译的目标。
- “全部编译”  对程序编辑器、系统块和数据块进行编译。当您使用“全部编译”命令时，哪一个窗口是焦点无关紧要。


### 使用输出窗口解决错误

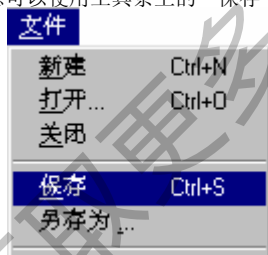
当您编译时，“输出窗口”列出发生的所有错误。错误根据位置（网络、行和列）以及错误类型识别。您可以单击错误行，调出错误的详细信息。

另请参阅：

[Micro/WIN编译错误](#)

## 1.6.9 5.9 如何保存作业

您可以使用工具条上的“保存”按钮  保存作业，或从“文件”菜单选择“保存”和“另存为”选项保存作业。



- “保存”允许您在作业中快速保存所有改动。（然而，您初次保存一个项目时，会被提示核实或修改当前项目名称和位置。）
- “另存为”允许您修改当前项目的名称和 / 或目录位置。

当您首次创建项目时，STEP 7-

Micro/WIN的默认值名称为 **Project1.mwp**。您可以接受或修改该名称；如果您接受该名称，下一个项目的默认名称将自动递增为 **Project2.mwp**。

STEP 7-Micro/WIN项目的默认目录位置是位于STEP-7-MicroWIN目录中的称作“项目”的文件夹，您可以不接受该默认位置。

## 1.7 第6章 如何建立通讯和下载程序

### 1.7.1 6.1 通讯概述

如何在运行STEP 7-


Micro/WIN的个人计算机和PLC之间建立通讯取决于您安装的硬件。如果您仅使用PC/PPI电缆连接计算机和PLC，您只需连接电缆，接受安装STEP 7-Micro/WIN软件时，在STEP 7-Micro/WIN中为个人计算机和PLC指定的默认参数即可。

您可以在任何时间建立通讯或编辑通信设置。

以下列出建立通讯通常要求的任务：

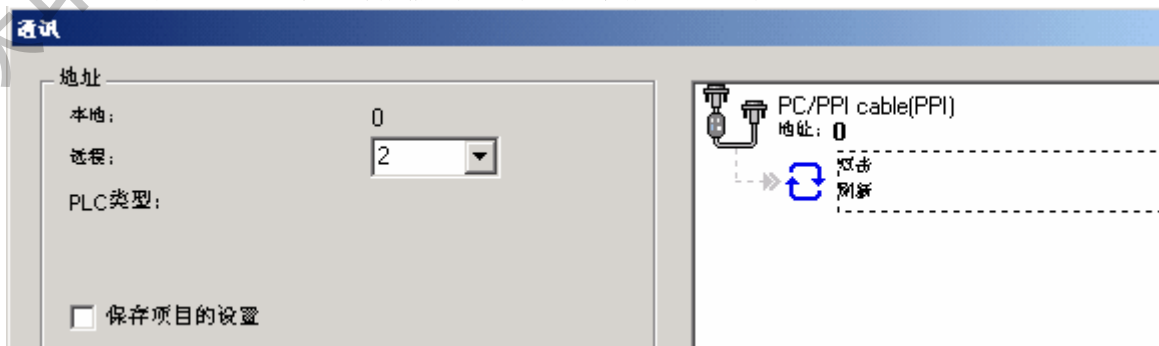
- 在PLC和运行STEP 7-Micro/WIN的个人计算机之间连接一条电缆。对于简单的PC/PPI连接，将速度设为9600波特、DCE、11位。如果您使用PC/PPI电缆，请确保电缆长度不超过100米。
- 供选用：核实STEP 7-Micro/WIN中的PLC类型选项与您的PLC实际类型相符。
- 如果您使用简单的PC/PPI连接，您可以接受安装STEP 7-Micro/WIN时在“设置PG/PC接口”对话框中提供的默认通讯协议。否则，从“设置PG/PC接口”对话框为个人计算机选择另一个通讯协议，并核实参数（站址、波特率等）。
- 供选用：核实系统块的端口标记中的PLC配置（站址、波特率等）。如有必要，修改和下载更改的系统块。

### 1.7.2 6.2 如何测试通讯网络

1. 在STEP 7-Micro/WIN中，单击浏览条中的“通讯”图标，或从菜单选择检视 > 元件 > 通讯。



2. 从“通讯”对话框的右侧窗格，单击显示“双击刷新”的蓝色文字。





如果您成功地在网络上的个人计算机与设备之间建立了通讯，会显示一个设备列表（及其模型类型和站址）。





STEP 7-Micro/WIN在同一时间仅与一个PLC通讯。会在PLC周围显示一个红色方框，说明该PLC目前正在与STEP 7-Micro/WIN通讯。您可以双击另一个PLC，更改为与该PLC通讯。

### 1.7.3 6.3 如何下载程序

如果您已经成功地在运行STEP 7-Micro/WIN的个人计算机和PLC之间建立通讯，您可以将程序下载至该PLC。请遵循下列步骤。

#### 注释：

从您的个人计算机将程序块、数据块或系统块下载至PLC时，从个人计算机下载的块内容盖写目前在PLC中的块内容（如PLC中有）。在您开始下载之前，核实您希望盖写PLC中的块。

1. 下载至PLC之前，您必须核实PLC位于“停止”模式。检查PLC上的模式指示灯。如果PLC未设为“停止”模式，单击工具条中的“停止”按钮，或选择**PLC > 停止**。
2. 单击工具条中的“下载”按钮，或选择**文件 > 下载**。出现“下载”对话框。
3. 根据默认值，在您初次发出下载命令时，“程序代码块”、“数据块”和“CPU配置”（系统块）复选框被选择。如果您不需要下载某一特定的块，清除该复选框。
4. 单击“确定”，开始下载程序。
5. 如果下载成功，一个确认框会显示以下讯息：下载成功。继续执行步骤12。
6. 如果STEP 7-Micro/WIN中用于您的PLC类型的数值与您实际使用的PLC不匹配，会显示以下警告讯息：“为项目所选的PLC类型与远程PLC类型不匹配。继续下载吗？”
7. 欲纠正PLC类型选项，选择“否”，终止下载程序。
8. 从菜单条选择**PLC > 类型**，调出“PLC类型”对话框。
9. 您可以从下拉列表方框选择纠正类型，或单击“读取PLC”按钮，由STEP 7-Micro/WIN自动读取正确的数值。
10. 单击“确定”，确认PLC类型，并清除对话框。
11. 单击工具条中的“下载”按钮，重新开始下载程序，或从菜单条选择**文件 > 下载**。
12. 一旦下载成功，在PLC中运行程序之前，您必须将PLC从STOP（停止）模式转换回RUN（运行）模式。单击工具条中的“运行”按钮，或选择**PLC > 运行**，转换回RUN（运行）模式。

#### 另请参阅：

[通讯概述 \(GS 6.1\)](#)

[如何测试通讯网络 \(GS 6.2\)](#)

[下载 \(文件 > 下载\)](#)

[在运行模式中进行程序编辑](#)

[STEP 7-Micro/WIN编译程序错误](#)

[如何纠正编译错误和下载错误 \(GS 6.4\)](#)

[如何从PLC上载 \(GS 8.3\)](#)

[上载 \(文件 > 上载\)](#)

### 1.7.4 6.4 如何纠正编译错误和下载错误

输出窗口在您编译程序或下载程序时随时自动显示编译程序信息和错误讯息。讯息通常包括发生错误的网络、列和行位置以及错误代码和说明。

```
正在编译程序块...
MAIN (OB1)
行 16, 错误 2700: 非法 STL 指令
块尺寸 = 0 (字节), 1 错误
```

双击错误讯息，在程序编辑器中显示包含错误的网络。

如果您已经关闭输出窗口，从菜单条选择**检视 > 帧 输出窗口**，重新显示输出窗口。

另请参阅：  
[错误讯息](#)

## 1.8 第7章 如何监控和调试程序

### 1.8.1 7.1 调试和监控特征概述

您成功地在运行的STEP 7-Micro/WIN编程设备和PLC之间建立通讯并向PLC下载程序后，则可利用“调试”工具条的诊断特征。可单击工具条按钮或从“调试”菜单列表选择项目，选择调试工具。

[调试工具条](#)



本标题讨论下列主题：

[什么是“状态”？](#)

[前提条件](#)

[收集状态的不同方法](#)

[以连续、瞬态图或触发暂停的方式收集状态](#)

[PLC模式](#)

[状态通讯与扫描循环](#)

[模拟程序条件（读取、写入、强制、取消强制）](#)

[检查交叉引用和元素用法](#)

[向程序下载改动](#)

[重设用于调试和编辑的帧设置](#)

## 什么是“状态”？

[返回顶端](#)

‘状态’这一术语指显示程序在PLC中执行时有关PLC数据当前值和功率流的信息。您可以使用状态图和程序状态窗口读取、写入和强制PLC数据值。

在控制程序的执行过程中，PLC数据的动态改变可用三种不同方式检视：

### 图状态

在一表格中显示状态数据：每行指定一个要监视的PLC数据值。您指定一个内存地址、格式、当前值及新值（如果使用写入命令）。

### 趋势显示

用随时间而变的PLC数据之绘图跟踪状态数据：

您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势数据亦可在趋势视图中直接赋值。

### 程序状态

在程序编辑器窗口中显示状态数据：当前PLC数据值会显示在引用该数据的STL语句或LAD/FBD图形旁边。LAD图形也显示功率流。

## 程序状态和图状态 (或趋势视图) 窗口可以同时运行：

在状态图窗口写入或强制PLC数据将PLC数据改动应用于程序状态窗口。

在程序状态窗口写入或强制PLC数据还会将新PLC数据改动应用于状态图窗口。

## 检视状态数据的前提条件

[返回顶端](#)

您必须执行下列任务，才能监控状态数据或调试程序：


- 成功地在STEP 7-Micro/WIN和PLC之间建立通讯。
- 成功地向PLC下载程序。
- 欲检视连续状态更新，PLC必须位于RUN（运行）模式。否则，您只能看到对I/O的改动（如果有）。由于PLC程序辉溢葱校 I/O的改动不会对“状态”显示中的程序逻辑产生预期的影响。
- 如果您位于程序不在执行的区域（例如，子例行程序、中断例行程序或由于JMP指令被绕过的区域），不会有状态显示，因为不在对代码执行扫描。

### 注释：

- STEP 7-Micro/WIN中项目的时间标记和PLC必须匹配，才能激活状态数据通讯。时间标记比较（在您尝试打开状态时自动执行）繁 STEP 7-Micro/WIN中的项目状态表示正确反映实际PLC程序。如果时间标记不匹配，您可以检查“时间标记不匹配”对话框中南吕 斜患 私铸男 繅斐刹黄 涸。
- 下载程序后，您还应将PLC转换回RUN（运行）模式。否则会报告STOP（停止）模式状态数据，但PLC无法执行程序，因此您将无法看到预期的逻辑操作。
- 

## 在程序编辑器窗口中收集状态的不同方法

[返回顶端](#)

单击“程序状态打开/关闭”按钮，或选择菜单命令**调试 (Debug) > 程序状态 (Program Status)**，在程序编辑器窗口中显示PLC数据状态。状态数据收集按以前选择的模式开始。请参阅以下说明。

LAD和FBD程序有两种不同的程序状态数据收集模式。选择**调试 (Debug) > 使用执行状态 (Use Execution Status)**菜单命令会在状态模式复选标记打开和关闭之间切换。必须在程序状态操作开始之前选择状态模式。

- **执行状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)**菜单项目被勾选后）



- **扫描结束状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)**菜单项目被取消勾选后）



状态图始终使用“扫描结束”状态数据。以前的 STEP 7 Micro/WIN 版本用“扫描结束”状态数据建立 LAD 和 FBD 程序状态视图。如果您使用的是第一代 PLC，执行状态不受支持。如果您尝试执行执行状态，“扫描结束”状态开始。

#### 执行状态色彩：

- 程序被扫描时，电源横杆会带有色彩。
- 图形中的功率流用色彩表示。
- 接点？接点打开时，指令会带有色彩。
- 线圈？输出打开时，指令会带有色彩。
- 方框和 SUBR 指令？指令接通电源并准确无误地成功执行时，方框和 SUBR 指令带有色彩。
- 绿色计时器和计数器表示计时器和计数器包含有效数据。
- 红色表示执行的指令有误。
- 跳接和标签指令现用时，显示为功率流色彩。如果为非现用，则显示为灰色。
- 灰色（灰色为默认赋值）表示无功率流、指令未扫描（跳过或未调用）或位于 STOP（停止）模式的 PLC。
- 布尔功率流位（仅限 FBD）。

LAD、FBD 和 STL 执行状态视图显示操作数数值，并表示每条指令在位于扫描循环执行程序阶段时的功率流。执行状态可以显示可能被执行后续程序指令盖写的中间数据值。所有显示的 PLC 数据值均从一个程序扫描循环收集。

#### 扫描结束状态

- 接通电源或逻辑真的接点和线圈显示为蓝色（您可以选择 **工具 (Tools) > 选项 (Options)** 菜单项目并单击“色彩”标记，指定自己选择的色彩）。

“扫描结束”状态显示在程序扫描结束时读取的状态结果。这些结果可能不会反映 PLC 数据地址的所有数值改动，因为随后某绿蜂鸣铃语筑绿蛭 杞喂 翱瞻芊慈牒椭恹漆慈胧 怠 S 捐诳炯 PLC 扫描循环和相对慢速的 PLC 状态数据通讯之间存语乃俛快败稿 吧 杞喂 弊刺 允炯父鑿 柁 方喂 辈杉 氛 奠怠

#### STL 程序中程序状态举例：

当您开启 STL 中的状态时，程序编辑器窗口被分为一个代码区（左侧）和一个状态区（右侧）。可以根据您希望监控的数值类型

在 STL 状态中共有三个可监控的数值类别：

- **操作数** 每条指令最多可监控三个操作数。
- **逻辑堆栈** 最多可监控四个来自逻辑堆栈的最新数值。
- **指令状态位** 最多可监控十二个状态位。

**工具 (Tools) >**

**选项 (Options)** 对话框的 STL 状态标记允许您选择或取消选择任何此类数值类别。如果您选择一个项目，该项目不会在“状态”显示中出现。

	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+16095	
5	T32	位	2#0	
6	T32	带符号	+0	

SIMATIC STL

2 · 3 · 4 · 5 · 6 · 7 · 8 · 9 · 10 · 11 · 12 · 13 · 14 · 15 · 16 ·

		操作数 1	操作数 2	操作数 3	0123	空
LD	I0.0	ON			1000	1
A	SM0.5	ON			1000	1
LD	I1.0	OFF			0100	0
A	I1.1	OFF			0100	0
OLD					1000	1
LD	I2.0	OFF			0100	0
A	SM0.5	ON			0100	1
OLD					1000	1
LD	I0.2	ON			1100	1
A	SM0.5	ON			1100	1
LD	I1.2	OFF			0110	0
A	I1.3	OFF			0110	0
OLD					1100	1
ALD					1000	1
LPS					1100	1
MOVW	VW0, VW2	+15919	+15919		1100	1
AENO					1100	1
+I	VW0, VW2	+15919	+31838		1100	1
AENO					1100	1
=	Q0.0	ON			1100	1
LRD					1100	1
TON	T32, +32000	+288	+32000		1100	1
LRD					1100	1
TNCW	VW0	+15920			1100	1

MAIN | SBR\_0 | INT\_0 /

**LAD图形中执行状态举例：**

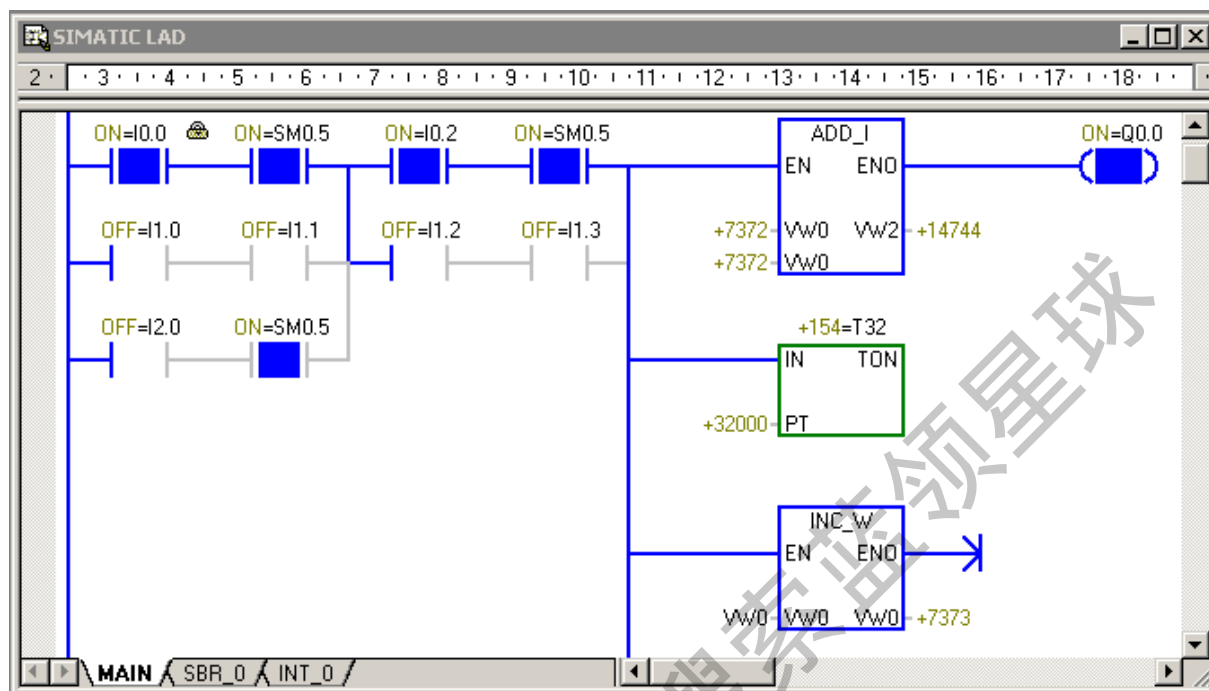
以下范例在LAD图形中使用LAD执行状态。状态图仅使用“扫描结束状态”数据收集。可在LAD图形和对应的状态图之间看出

状态图

3 · 4 · 5 · 6 · 7 · 8 · 9 · 10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20

	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+7652	
5	T32	位	2#0	
6	T32	带符号	+0	

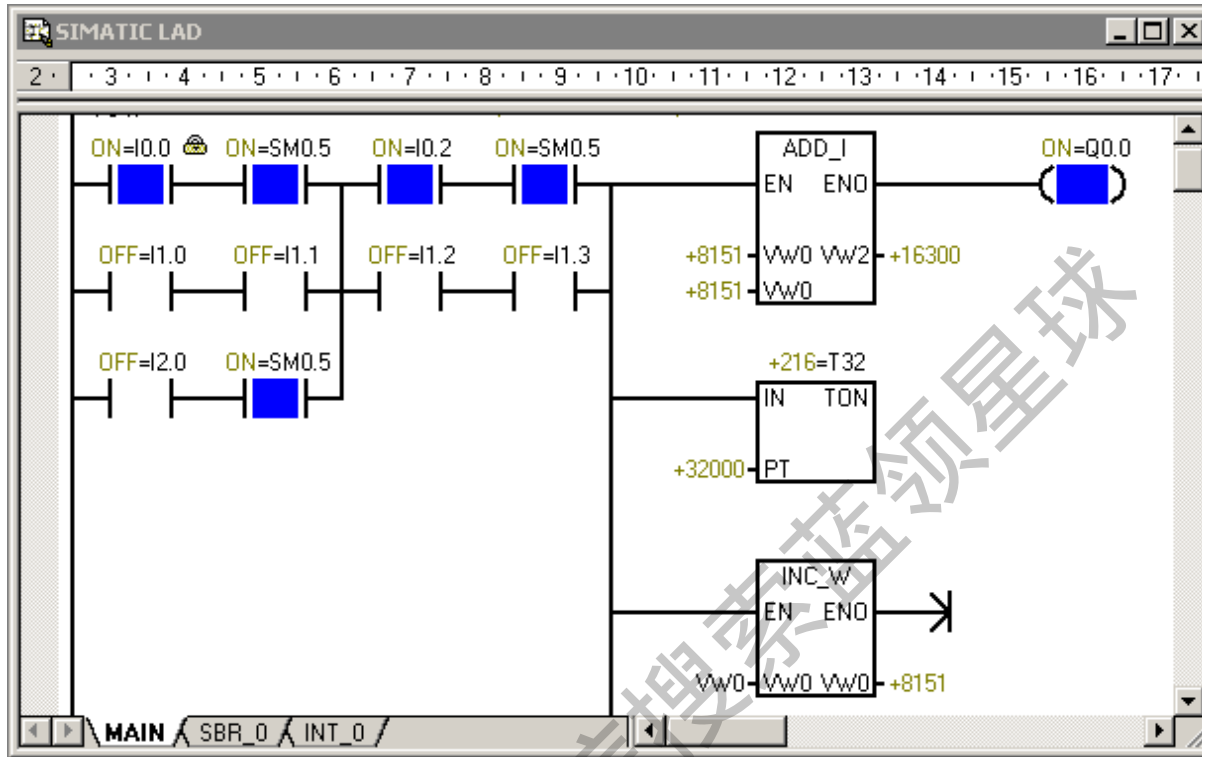
CHT1 /



#### LAD图形中扫描结束状态举例：

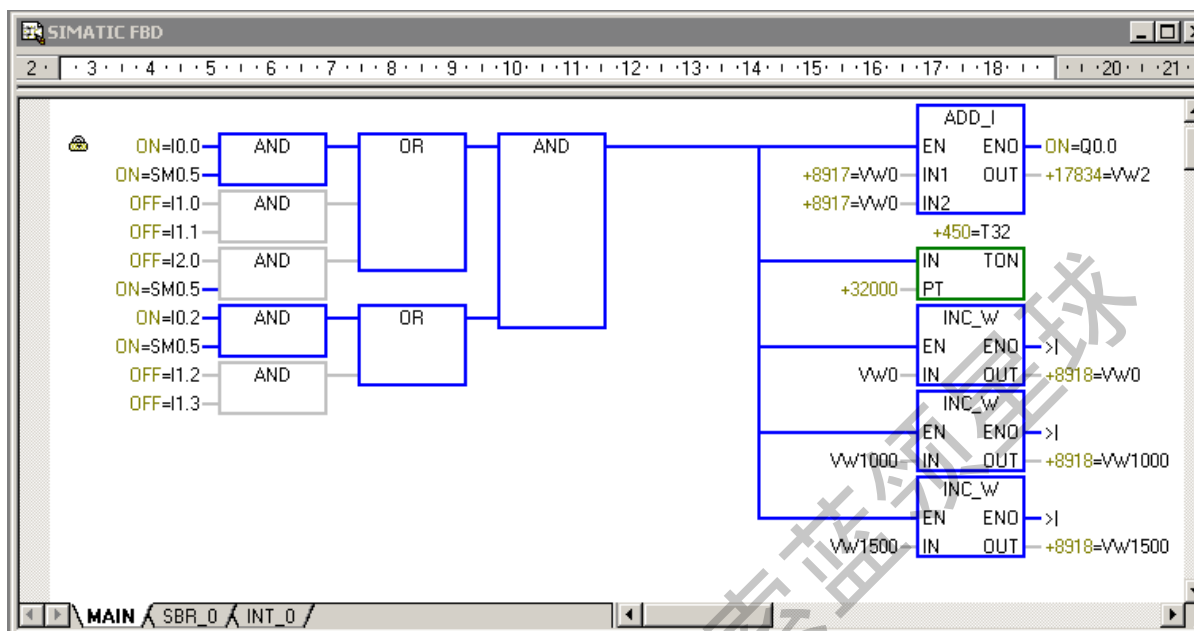
以下范例在LAD图形中使用“扫描结束状态”。由于两个窗口（程序状态和图状态）均使用相同的扫描 PLC结束数据，两个窗口允鞠嗤 PLC数据值。这种情形会引起误解，因为程序可以在收集最终“扫描结束”数值之前为相同的位置指定很多数值。中间临时数值不显示。在该范例中，ADD\_I方框未能正确执行，因为输入相加超出了带符号字整数的最大范围。与红色方框的执行状态范例不同，您必须读取状态值，才能看到Vw0 + Vw0 <> VW2。

地址	格式	当前值	新数值
1 I0.0	位	2#1	
2 I0.2	位	2#1	
3	带符号		
4 VW0	带符号	+8418	
5 T32	位	2#0	
6 T32	带符号	+485	



FBD图形中的执行状态举例:

地址	格式	当前值	新数值
1 I0.0	位	2#1	
2 I0.2	位	2#1	
3	带符号		
4 Vw0	带符号	+8958	
5 T32	位	2#0	
6 T32	带符号	+0	



按照连续、瞬态图或触发（暂停程序状态）的方式收集状态值

[返回顶端](#)

**注释：**

“连续”并非意味着实时；而是指编程设备不断地从 PLC 轮询状态信息，在屏幕上显示，并按照通讯允许的速度更新显示。可能

**连续**

- 打开程序编辑器窗口，并启动“程序状态”，当 PLC 位于 RUN（运行）模式时检视程序执行的连续状态更新。
- 打开状态图窗口，并启用“图状态”，当 PLC 位于 RUN（运行）模式时检视连续更新。


**瞬态图**


- 将 PLC 转换为 STOP（停止）模式，收集单个状态更新。当 PLC 位于 STOP（停止）模式时，您可以使用“多次扫描”功能检视一次或多次扫描。您还可以引用网络中的特殊内存 SM0.1 首次扫描位，读取该网络的首次扫描状态值。
- 关闭图状态，如果您希望收集单次状态图更新并不希望将 PLC 转换至 STOP（停止）模式，使用“单次读取”功能。如 PLC 转换为 STOP（停止）模式，并开启“图状态”，您也会收集单次状态更新。此外，您可以在检视状态图时使用“多次扫描”和“首次扫描”功能。

**执行状态模式（LAD / FBD / STL）中的程序状态触发暂停**

欲检视触发暂停 PLC 数据状态，您必须：

1. 将程序下载至 PLC，并将 PLC 设为 RUN（运行）模式。您还可以从 STOP（停止）模式开始触发暂停。如果您从 STOP（停止）模式开始，则可以捕获首次扫描。
2. 滚动程序编辑器窗口，显示您希望检视的程序网络。网络可以位于任何 POU（主程序、子例行程序或中断程序）中。
3. 使用 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单命令，将“使用执行状态”复选标记切换为打开。
4. 激活程序状态，开始状态数据通讯。

5. 单击“暂停程序状态” 工具条按钮，或用鼠标右键在程序编辑器中单击，并选择弹出菜单命令“暂停程序状态”。
6. 更新暂停之后，触发网络中的状态值在您检视时保持不变。当您准备移至另一个网络时，则单击“暂停程序状态”

 工具条按钮，将暂停切换为关闭，并再次开始连续更新。



一旦获得新的状态信息，只要程序状态暂停功能被启用，该信息就会保持在屏幕上。在您取消选择程序状态暂停功能之前，无论 PLC 中实际变化速度多快，均不会进一步执行状态更新。如果没有程序状态暂停功能，屏幕就会不断刷新，状态数据在您读取和解释之前就可能消失（或者数据根本没有显示）。

**PLC RUN / STOP（运行 / 停止）模式**

[返回顶端](#)



使用以下一种方法改变PLC操作模式：

- 单击“运行”按钮，执行RUN（运行）模式，或者单击“停止”按钮，执行STOP（停止）模式。
- 选择**PLC > 运行（RUN）**菜单命令，执行RUN（运行）模式，或选择**PLC > 停止（STOP）**菜单命令，执行STOP（停止）模式。
- 以手动方式改变位于PLC上的模式开关。
- 在程序中插入一条STOP（停止）指令。

#### 注释：

欲使用STEP 7-Micro/WIN软件控制RUN / STOP（运行 / 停止）模式，必须在STEP 7-Micro/WIN和PLC之间存在一条通讯路径。此外，必须将PLC硬件模式开关设为TERM（终端）或RUN（运行）。将模式开关设为TERM（终端）不会改变PLC操作模式，但却允许STEP 7-Micro/WIN改变PLC操作模式。位于PLC前方的状态LED表示当前操作模式。当程序状态或状态图操作正在进行时，在STEP 7-Micro/WIN窗口右下方处附近的状态条上会出现一个RUN / STOP（运行 / 停止）指示灯。

虽然程序在STOP（停止）模式中不执行，PLC操作系统继续监控PLC（收集PLC RAM和I/O状态），将状态数据传递给STEP 7-Micro/WIN，并执行所有的“强制”或“取消强制”命令。当PLC位于STOP（停止）模式中时，您可以执行以下操作：

- 使用图状态或程序状态检视操作数的当前值。（此一步骤具有与执行“单次读取”相同的效果，因为程序未执行。）
- 您可以使用图状态或程序状态强制数值。使用图状态写入数值。
- 写入或强制输出。
- 执行有限次数扫描，并通过状态图和 / 或项目状态检视效果。

当PLC位于RUN（运行）模式时，您不得使用“首次扫描”或“多次扫描”功能。您可以在状态图中写入和强制数值，或使用LAD或FBD程序编辑器强制数值，方法与您在STOP（停止）模式中强制数值相同。您还可以执行以下操作（不得从STOP（停止）

- 使用图状态收集不断变化的PLC数据值的连续更新信息。（如果您希望使用单次更新，图状态必须关闭，才能使用“单次读取”命令。）
- 使用程序状态收集不断变化的PLC数据值的连续更新信息。
- 使用RUN（运行）模式功能中的“程序编辑”编辑程序，并将改动下载至PLC。

#### 状态通讯与扫描循环

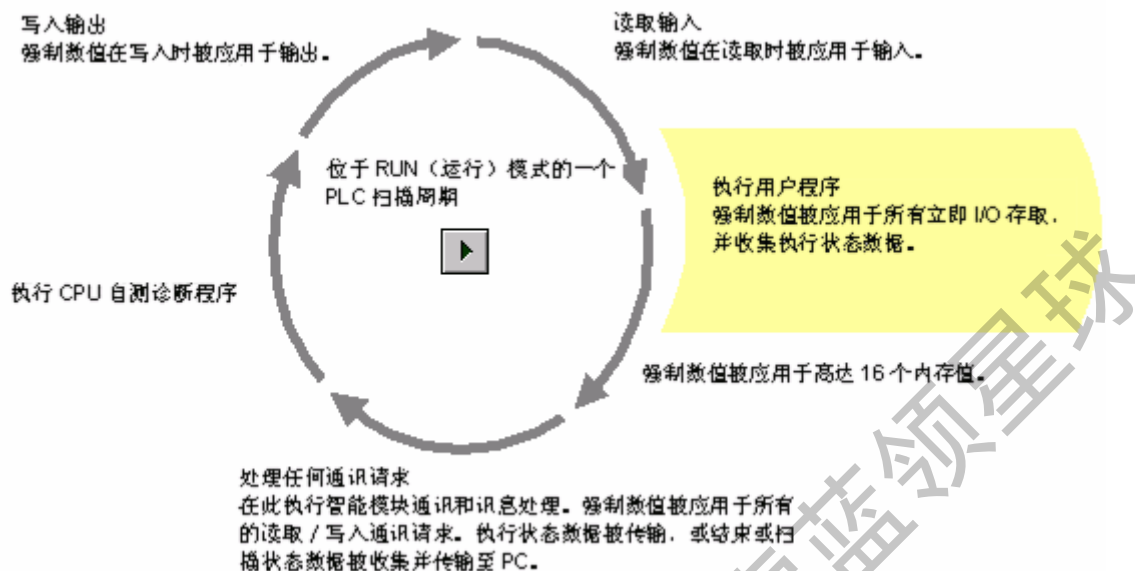
[返回顶端](#)

PLC在连续循环中读取输入、执行程序逻辑、写入输出和执行系统操作和通讯。该扫描循环速度极快，每秒执行多次。虽然STEP 7-Micro/WIN会快速发出状态请求，您应当认识到您检视的不是PLC中出现的每一个事件，这一点十分重要。

由于PLC和您观察的程序状态编程设备之间存在通讯时间滞后，您看到的显示的操作数数值总是在状态显示中改变之前即在PLC中改变，但没有在程序状态表示中改变。

如果您使用“扫描结束”状态模式检视程序状态（当**调试（Debug） > 使用执行状态（Use Execution Status）**菜单项目被取消勾选时），您在几个扫描循环中采集数据。

如果您使用“执行状态”模式检视程序状态（当**调试（Debug） > 使用执行状态（Use Execution Status）**菜单项目被勾选时），所有显示的程序状态值一定来自同一个扫描循环。



**注释:**

一次只有一个状态窗口被授权运行状态。如果您尝试在任何其他与同一个 PLC 连接的 STEP 7 Micro/WIN 应用程序中运行状态, 则会收到错误信息。

**模拟进程条件 (读取、写入、强制、取消强制)**

[返回顶端](#)

您可以模拟进程条件, 方法是在程序状态操作过程中从程序编辑器和从状态图向操作数写入或强制新数值。

使用“程序状态”和“图状态”按钮或“调试”菜单命令, 开始状态数据通讯, 并启用调试工具。您还可以使用硬件输入模拟器开关 (西门子 (Siemens) 可提供) 以手动方式在数字输入打开和关闭之间切换, 并观察 PLC 上的 LED 指示灯, 观察数字输出的条件。

从“调试”工具条或“调试”菜单列表存取以下功能。






**单次读取** (仅限状态图) 如果您希望获得一次“瞬态图” (对状态图中的所有数值一次更新), 使用“单次读取”。默认值为状态图连续轮询 PLC, 获取状态更新信息。当您单击“状态图”时, 状态图会切换为关闭, “单次读取”图标变为黄色。




**全部写入** (仅限状态图)

在您完成对几行的“新数值”列的改动后, 您可以使用“全部写入”, 将所有需要的改动发送至 PLC。



**强制** 您可以在操作程序状态时从程序编辑器和状态图强制地址。欲强制程序状态地址, 用鼠标右键单击一个参数, 并选择从菜单列表强制。欲强制状态图地址为某一数值, 您必须首先规定所需的数值, 方法是读取该数值 (如果您希望强制当前值) 或键入该数值 (如果您希望将地址强制为一个新数值)。您一旦使用了强制功能, 则在每次扫描时该数值均被重新应用于地址, 直至您取消强制地址。“强制”功能盖写立即读取或立即写入指令。“强制”功能还盖写被配置为在转换为 STOP (停止) 模式时进入一个指定数值的输出: 如果 CPU 进入 STOP (停止) 模式, 输出被设为强制数值, 而非非配置的数值。通过将 V 或 M 内存强制为字节、字或双字, 将 AI 或 AQ 内存在偶数字节边界上 (例如 AIW6 或 AIW14) 或 I/O 点上 (I 和 Q 位址) 强制为字, 模拟逻辑条件。您可以一次强制 16 个 (V、M、AI 或 AQ) 地址和所有的 I/O 位 (所有 I 和 Q 位址)。 该图标表示该地址被明确强制。该地址数值在地址被取消强制之前无法改变。 该图标表示该地址被隐含强制。如果地址是一个被明确强制的较大地址的一部分, 该地址则被认为是隐含强制。例如, 如果 VW0 被强制, 则 VB0 是隐含强制 (VB0 是 VW0 的第一个字节)。隐含强制数值无法自身取消强制。您必须取消强制较大的地址, 然后才能改变该地址数值。如果您强制 VD0 (该地址包含 VB0、VB1、VB2 和 VB3), 则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制, 则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。



**取消强制** 对于程序状态和状态图, 选择一个地址, 并使用“取消强制”按钮从该特定地址移除强制。



**取消强制** 对于程序状态和状态图, 选择一个地址, 并使用“取消强制”按钮从该特定地址移除

强制功能。您还可以选择（单击）一个参数，然后用鼠标右键单击该参数，查看“强制”和“取消强制”功能的弹出菜单。



#### 全部取消强制

如果您希望从所有的地址中移除强制功能，使用“全部取消强制”按钮。在应用“全部取消强制”之前，您无须选择单个地址。



#### 读取全部强制（仅限状态图）

当您使用“读取全部强制”功能时，状态图的“当前数值”列会为已经明确强制、隐含强制或部分隐含强制的所有地址显示一个图标。

### 检查交叉引用和元素用法

[返回顶端](#)

调试程序时，您可以决定是否增加、删除或编辑参数。

使用“交叉引用”窗口查看程序中当前指定参数的方法，这样可以防止您意外地重复赋值。

此外，如果您希望在RUN（运行）模式中执行程序编辑，并且某些网络使用正向或负向转换（EP和N接点或方框以及STL EU或ED指令），您一定要检查“交叉引用”信息中的EU/ED数目，以防重复指令号码。

### 向程序下载改动

[返回顶端](#)

一旦调试部分代码，如果PLC位于STOP（停止）模式，您可以编辑程序、下载程序、然后再次监控状态，查看改动是否发生作用。

您不必进入STOP（停止）模式就可以在RUN（运行）模式中执行程序编辑并向PLC下载较小的改动。



**警告** 在RUN（运行）模式中向PLC下载改动时，您的改动会立即影响程序操作。不得出错；程序编辑中的错误可能导致人员死亡或严重伤害和/或设备损坏。仅限合格人员在RUN（运行）模式中执行程序编辑。

### 重设用于调试和编辑的帧设置

[返回顶端](#)

欲重设将窗口和工具条的可视性、尺寸和位置设为默认值，选择**检视（View）> 帧（Frame）> 全部重设（Reset All）**菜单命令。

您有两套不同的工作空间窗口的便利，可在调试程序时获得较大的灵活性：窗口可视性和定位。这是比正常编辑空间更大的检视空间。当您进入程序状态时，您已经设置的任何窗口定位保持初始进入的状态；如果您关闭程序状态，窗口保持初始进入的状态。同样，任何在编辑程序时执行的窗口定位均在运行时与程序状态分开维护。

[返回顶端](#)

#### 另请参阅：

[如何在程序编辑器中显示状态（GS 7.2）](#)

[如何在状态图中显示状态（GS 7.3）](#)

[如何执行有限数目扫描（GS 7.4）](#)

[如何下载程序（GS 6.3）](#)

[时间标记不匹配错误](#)（确保编程设备中的项目与PLC中的项目相匹配）

[交叉引用和元素用法](#)（确保程序编辑不引起重复赋值）

[在运行模式中执行程序编辑](#)

[PLC运行/停止模式](#)

[在停止模式中写入和强制输出](#)

[安装和移除S7-200设备](#)







[接地和布线指南](#)

## 1.8.2 7.2 如何在程序编辑器中显示状态


当您成功地在您运行STEP 7-

Micro/WIN的编程设备和PLC之间建立通讯并将程序下载至PLC后，您可以用“程序状态”功能操作和测试程序网络。

使用以下一种方法，设置程序编辑器窗口，显示您希望测试的程序部分和网络：

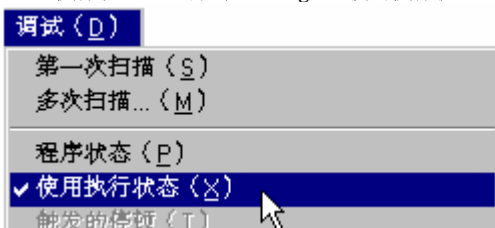
- 单击**浏览条**上的“程序块”按钮，则会打开主程序（OB1）[POU](#)。您可以单击子例行程序或中断例行程序标记
- 打开**指令树**“程序块”文件夹，方法是单击分支扩充  图标，或双击“程序块”文件夹  图标。然后双击主程序（OB1）  图标、一个子例行程序  图标或一个中断例行程序  图标，打开所需的[POU](#)。

在程序编辑器视图正确设置后，您必须启动程序状态，开始 PLC 状态数据通讯。通常，PLC 应当位于 RUN（运行）模式，监控改 PLC 数据值。

- 单击“程序状态打开 / 关闭”按钮  或选择菜单命令**调试（Debug） > 程序状态（Program Status）**，在程序编辑器窗口中显示 PLC 数据状态。状态数据通讯从以前选择的模式开始。请参阅以下说明。

共有两种不同的 LAD 和 FBD 程序状态数据收集模式。选择 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单命令使状态模式复选标记在打开和关闭之间切换。在开始程序状态操作之前，必须选择状态收集模式。

- 执行状态**（当 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单项目被勾选时）



- 扫描结束状态**（当 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单项目被取消勾选时）





状态图始终使用“扫描结束”状态数据。以前的 STEP 7 Micro/WIN 版本用“扫描结束”状态数据建立 LAD 和 FBD 程序状态视图。如果您使用的是第一代 PLC，执行状态不受支持。如果您尝试执行执行状态，“扫描结束”状态开始。

对于 LAD 和 FBD 程序状态，使用 **工具（Tools） > 选项（Options）** 程序编辑器标记，或调整图形和字体大小。您可以在状态运行时设置工具选项。LAD/FBD 程序编辑器和程

本题讨论下列主题：  
[理解程序状态显示](#)  
[以连续、瞬态图或触发暂停方式收集状态](#)  
[状态通讯和扫描循环](#)  
[模拟进程条件（写入、强制、取消强制、全部取消强制）](#)  
[无法预期的强制数值程序操作](#)  
[调节程序状态显示](#)

#### 理解程序状态显示

锁定图标  表示该数值已经被“明确”或直接强制为当前正在显示的数值。

灰色隐去锁定图标  表示该数值已经被“隐含”强制，即不对地址进行直接强制，但内存区落入另一个被明确强制的较大区域

半块图标  表示数值被“部分”强制。例如，如果您明确强制 VB1，则 VW0 被部分强制，因为其中的一个字节 VB1 被强制。

#### STL 程序中的执行状态：

程序编辑器窗口开始显示您已经选择的任何信息类别的状态，方法是使用“选项”对话框（可通过菜单命令 **工具（Tools） > 选项（Options）** 进入该对话框）的 STL 状态标记。

执行状态色彩（以下命名的色彩是默认色彩赋值）：

- 正在扫描程序时，电源横杆显示为蓝色。
- 图形中的功率流用蓝色表示。
- 接点 接点打开时，指令会显示为蓝色。
- 线圈？ 输出打开时，指令会显示为蓝色。
- 方框和SUBR指令？ 指令接通电源并准确无误地成功执行时，方框和SUBR指令显示为蓝色。
- 绿色计时器和计数器表示计时器和计数器包含有效数据。
- 红色表示执行的指令有误。
- 跳接和标签指令现用时，显示为功率流色彩。如果为非现用，则显示为灰色。
- 灰色表示无功率流、指令未扫描（跳过或未调用）或位于STOP（停止）模式的PLC。
- 布尔功率流位（仅限FBD）。
- 如果您尝试将地址视作指针，但该地址包含的数值不包含有效内存地址，则错误用红色显示。

LAD、FBD和STL执行状态视图显示操作数数值，并表示每条指令在位于扫描循环执行程序阶段时的功率流。执行状态可以显示可能被执行后续程序指令盖写的中间数据值。所有的执行状态数据值均从一个程序扫描循环收集。



**执行状态提示：**

- PLC位于RUN（运行）模式时，会显示状态值和元素色彩。
- PLC位于STOP（停止）模式时，元素色彩不会显示。
- 写入、强制和取消强制只有在PLC位于RUN（运行）模式并实际收集状态时才能使用。
- 当状态不在STOP（停止）模式中收集时，写入、强制和取消强制弹出菜单选项被禁用。
- 当状态被“暂停”时，写入、强制和取消强制弹出菜单选项被禁用。
- 如果输入是文字，写入、强制和取消强制弹出菜单选项被禁用。
- 强制和取消强制工具条按钮和弹出菜单选项仅在用于带有V、M、AI和AQ内存类型的字节、字和双字长度时才被启用。
- 强制和取消强制工具条按钮和弹出菜单选项仅在用于带有I和Q内存类型的位、字节、字和双字长度时才被启用。

#### LAD和FBD图形和所有状态图中的扫描结束状态：

- 接通电源或逻辑真的接点和线圈显示为蓝色（您可以选择**工具（Tools）> 选项（Options）**菜单项目并单击“色彩”标记，指定自己选择的色彩）。

“扫描结束状态”显示在程序扫描结束时读取的状态结果。这些结果可能不会反映PLC数据地址的所有数值改动，因为随后某些蜂鸣铃铛酱蒜绿蛙 杞喂 翱瞻芊慈牒楠甌深慈胧 怠S捎诟烟 PLC扫描循环和相对慢速的PLC状态数据通讯之间存话乃俣快收稿 吧 杞喂 弊刺 允炯父簪 枢 方喂 辈杉 氛 奠怠



**扫描结束状态提示：**

- 当PLC位于RUN（运行）或STOP（停止）模式并实际收集状态时，显示状态数值。
- 元素色彩只有在RUN（运行）模式中才会显示，帮助区别RUN（运行）和STOP（停止）模式。
- 锁定只有在PLC位于RUN（运行）或STOP（停止）模式并实际收集状态时才会显示。
- 写入、强制和取消强制只有在PLC位于RUN（运行）或STOP（停止）模式并实际收集状态时才能使用。但是，在STOP（停止）模式中时，写入输出（Q和AQ）选项只有在启用该功能的菜单项目被勾选时才能使用。
- 强制和取消强制工具条按钮和弹出菜单选项仅在用于带有V、M、AI和AQ内存类型的字节、字和双字长度时才被启用。
- 强制和取消强制工具条按钮和弹出菜单选项仅在用于带有I和Q内存类型的位、字节、字和双字长度时才被启用。

#### STL程序中程序状态举例：

当您开启STL中的状态时，程序编辑器窗口被分为一个代码区（左侧）和一个状态区（右侧）。可以根据您希望监控的数值类型

在STL状态中共有三个可监控的数值类别：

- **操作数** 每条指令最多可监控三个操作数。
- **逻辑堆栈** 最多可监控四个来自逻辑堆栈的最新数值。
- **指令状态位** 最多可监控十二个状态位。

**工具（Tools）>**

**选项（Options）** 对话框的STL状态标记允许您选择或取消选择任何此类数值类别。如果您选择一个项目，该项目不会在“状态”显示中出现。

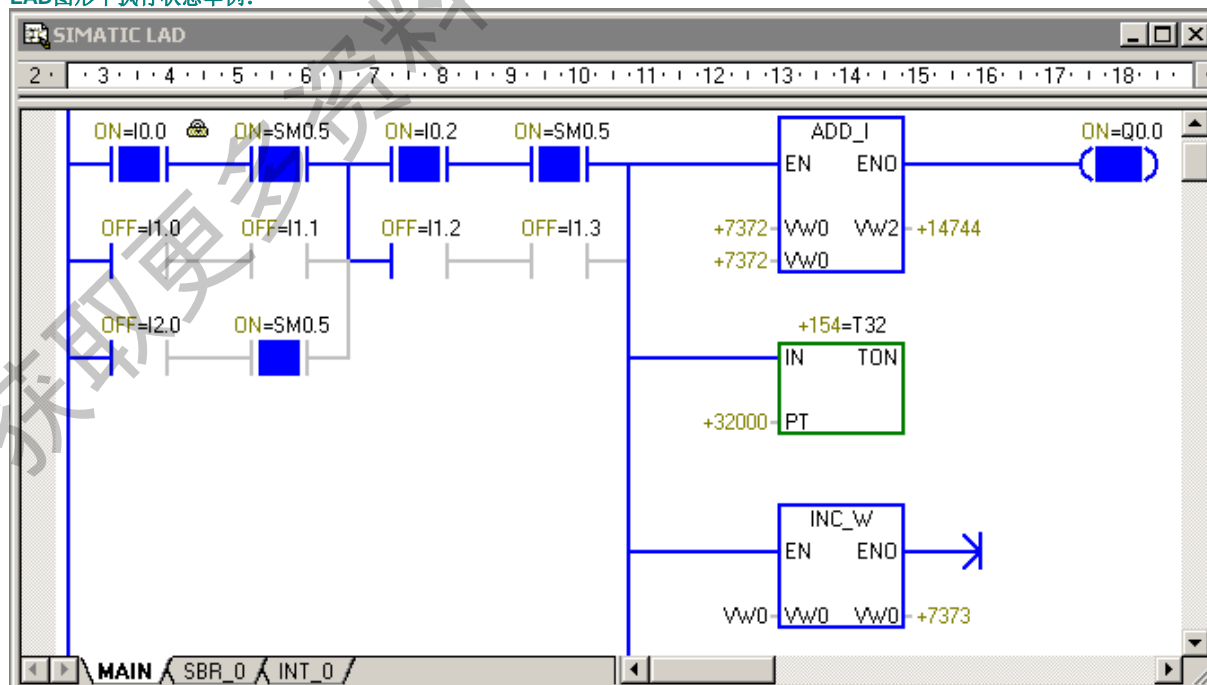
SIMATIC STL

2 · 3 · 4 · 5 · 6 · 7 · 8 · 9 · 10 · 11 · 12 · 13 · 14 · 15 · 16 ·

		操作数 1	操作数 2	操作数 3	0123	空
LD	I0.0	ON			1000	1
A	SM0.5	ON			1000	1
LD	I1.0	OFF			0100	0
A	I1.1	OFF			0100	0
OLD					1000	1
LD	I2.0	OFF			0100	0
A	SM0.5	ON			0100	1
OLD					1000	1
LD	I0.2	ON			1100	1
A	SM0.5	ON			1100	1
LD	I1.2	OFF			0110	0
A	I1.3	OFF			0110	0
OLD					1100	1
ALD					1000	1
LPS					1100	1
MOVW	VW0, VW2	+15919	+15919		1100	1
AENO					1100	1
+I	VW0, VW2	+15919	+31838		1100	1
AENO					1100	1
=	Q0.0	ON			1100	1
LRD					1100	1
TON	T32, +32000	+288	+32000		1100	1
LRD					1100	1
TNCW	VW0	+15920			1100	1

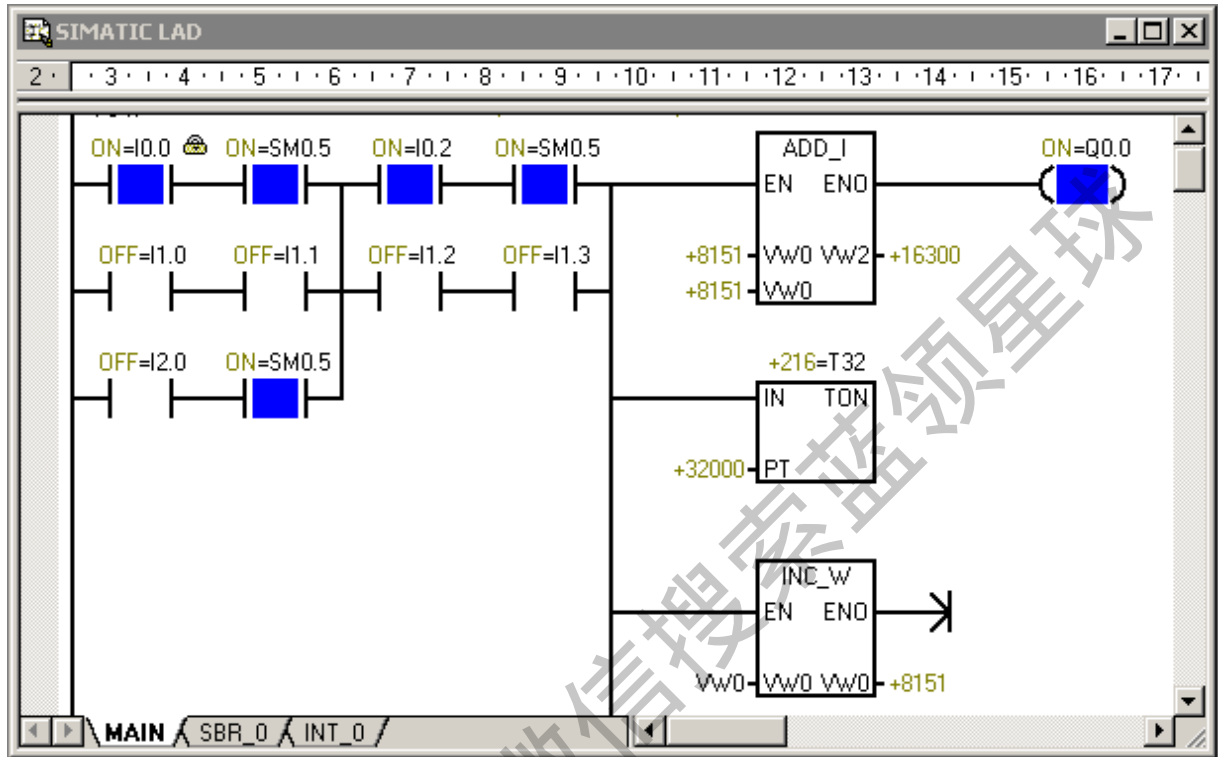
MAIN / SBR\_0 / INT\_0 /

LAD图形中执行状态举例:



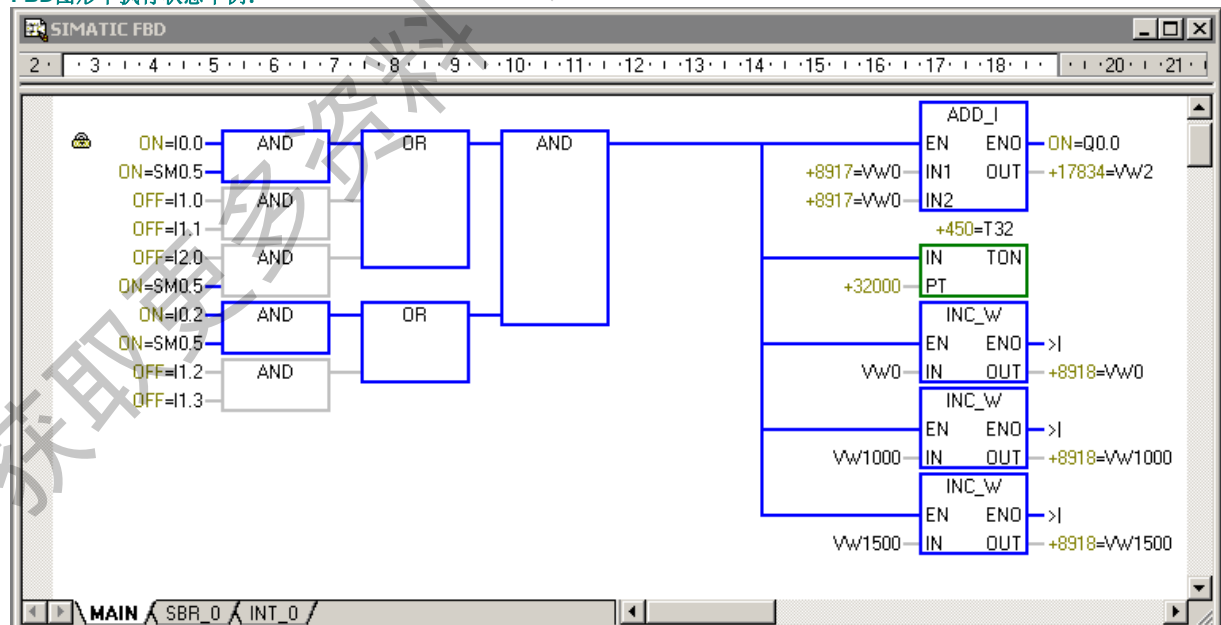
LAD图形中扫描结束状态举例:

程序可以在收集最终“扫描结束”数值之前为相同的位置指定很多数值。中间临时数值不显示。在该范例中，ADD\_I方框未能正确执行，因为输入相加超出了带符号字整数的最大范围。与红色方框的执行状态范例不同，您必须读取状态值，才能看到 Vw0 + Vw0 <> VW2。其后的网络可以将Q0.0写入为1或0，该功能防止将Q0.0状态值用作正确的ADD\_I方框执行指示灯。



在任意一种状态模式中，其他操作数的当前数据值在操作数旁（或在操作数位置）显示，从 PLC 读取改动时，显示被更新。

#### FBD 图形中执行状态举例：



**提示：** • 当您程序下载至PLC时，会被提示将PLC转换为STOP（停止）模式。如果您希望连续更新程序状态，请务必将PLC转换回RUN（运行）模式。您可以勾选STEP 7-Micro/WIN状态条，查看PLC的当前模式。该状态条每20-30秒更新一次。

- 当您启动程序状态时，很多其他STEP 7-Micro/WIN功能会被禁用。例如，除

非您再次关闭程序状态，否则不能编辑程序。其他功能（例如将一个程序编辑器的视图改变为另一个程序编辑器的视图）会钩踪唇刺 远 七侧日继 M 综 焯幼刺 虬虱脰髓卵=癍日甌麓蚩 十实锁淖刺 乙睢

如果您位于程序不在执行的区域（例如，子例行程序、中断例行程序或由于 JMP 指令被绕过的区域），不会有状态显示，因为不在对代码执行扫描，所有的指令操作数和功率流线均显示为灰色。

### 以连续、瞬态图或触发（暂停程序状态）的方式收集状态

[返回顶端](#)

#### 注释：

“连续”并非意味着实时；而是指编程设备不断地从 PLC 轮询状态信息，在屏幕中显示，并按照通讯允许的速度更新显示。可能藁ū痘衲承┘炯伊ū 氛 挡(0)谄聊恢邢允荆 蚌哒庀┘ 悼瞻鞑涕 坎 藁ū寥

#### 连续

- 打开程序编辑器窗口，并启动“程序状态”，当 PLC 位于 RUN（运行）模式时检视程序执行的连续状态更新。


#### 瞬态图

- 将 PLC 转换为 STOP（停止）模式，收集单个状态更新。当 PLC 位于 STOP（停止）模式时，您可以使用“多次扫描”功能检视一次或多次扫描。您还可以引用网络中的特殊内存 SM0.1 首次扫描位，读取该网络的首次扫描状态值。

### 执行状态模式（LAD / FBD / STL）中的程序状态触发暂停

欲检视触发暂停 PLC 数据状态，您必须：

1. 将程序下载至 PLC，并将 PLC 设为 RUN（运行）模式。您还可以从 STOP（停止）模式开始触发暂停。如果您从 STOP（停止）模式开始，则可以捕获首次扫描。
2. 滚动程序编辑器窗口，显示您希望检视的程序网络。网络可以位于任何 POU（主程序、子例行程序或中断程序）中
3. 使用 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单命令，将“使用执行状态”复选标记切换为打开。
4. 激活程序状态，开始状态数据通讯。

5. 单击“暂停程序状态”  工具条按钮，或用鼠标右键在程序编辑器中单击，并选择弹出菜单命令“暂停程序状态”
6. 更新暂停之后，触发网络中的状态值在您检视时保持不变。当您准备移至另一个网络时，则单击“暂停程序状态”



工具条按钮，将暂停切换为关闭，并再次开始连续更新。

一旦获得新的状态信息，只要程序状态暂停功能被启用，该信息就会保持在屏幕上。在您取消选择程序状态暂停功能之前，靡凶 菰 PLC 中实际变化速度多快，均不会进一步执行状态更新。如果没有程序状态暂停功能，屏幕则会不断刷新，状态数据在您读取和解释之前就可能消失（或者数据根本没有显示）。

### 状态通讯与扫描循环

[返回顶端](#)

PLC 在连续循环中读取输入、执行程序逻辑、写入输出和执行系统操作和通讯。该扫描循环速度极快，每秒执行多次。虽然 STEP 7-Micro/WIN 会快速发出状态请求，您应当认识到您检视的不是 PLC 中出现的每一个事件，这一点十分重要。

由于 PLC 和您观察的程序状态编程设备之间存在通讯时间滞后，您看到的显示的操作数数值总是在状态显示中改变之前即在 PLC 中改变，但没有在程序状态表示中改变。

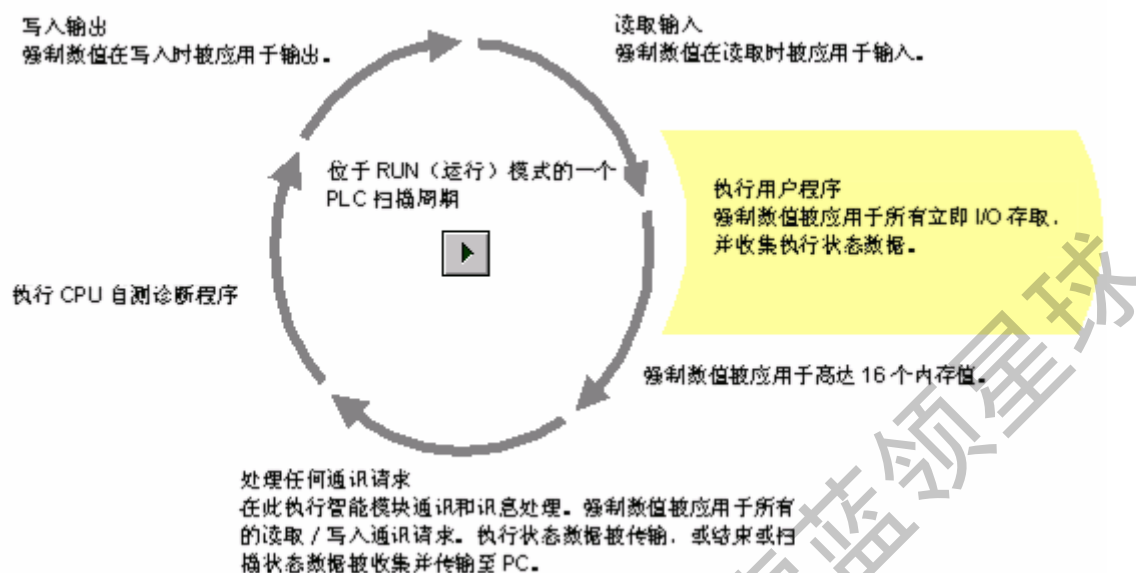
如果您使用“扫描结束”状态模式检视程序状态（当 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单项目被取消勾选时），您在几个扫描循环中采集数据。

如果您使用“执行状态”模式检视程序状态（当 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单项目被勾选时），所有显示的程序状态值一定来自同一个扫描循环。

#### 注释：

一次只有一个状态窗口被授权运行状态。如果您尝试在任何其他与同一个 PLC 连接的 STEP 7 Micro/WIN 应用程序中运行状态，则会收到错误讯息。





**提示：**

- 当您强制操作数时，始终注意状态显示，直至显示锁定图标。一旦显示锁定图标，则表示数值已经正确强制（否则您会看到被强制的数值，可以加以纠正）。
- 如果您希望强制一个数值起伏波动的操作数，请使用状态图。这样，您就无须键入来自不断变化的显示的数值，而直接将所需的数值键入“新数值”列，然后发出“强制”命令。

#### 用程序状态模拟进程条件（读取、强制、取消强制和全部取消强制）

[返回顶端](#)

您可以模拟进程条件，方法是在程序状态操作过程中从程序编辑器向操作数写入或强制新数值。

使用“程序状态”和“图状态”按钮或“调试”菜单命令，开始状态数据通讯，并启用调试工具。

欲写入操作数：

- 直接单击操作数（不要单击指令），然后用鼠标右键直接单击操作数，并从弹出菜单选择“写入”。

欲强制单个操作数：

- 直接单击操作数（不是指令），然后从“调试”工具条单击“强制”图标，或
- 直接用鼠标右键单击操作数（不是指令），并从弹出菜单选择“强制”。

欲取消强制单个操作数：

- 直接单击操作数（不是指令），然后从“调试”工具条单击“取消强制”图标，或
- 直接用鼠标右键单击操作数（不是指令），并从弹出菜单选择“取消强制”。


欲取消强制全部强制数值：

- 从“调试”工具条单击“全部取消强制”图标。

强制功能盖写立即读取或立即写入指令。强制功能还盖写向 STOP（停止）模式转换时配置进入某一指定数值的输出：如果 CPU 进入 STOP（停止）模式，输出被设为强制数值，而不是配置数值。



**警告** 在程序中强制数值时，在程序每次扫描时将操作数重设为该数值，与输入 / 输出条件或其他正常情况下对操作数值有影响的程序逻辑无关。强制可能导致程序操作无法预料，无法预料的程序操作可能导致人员死亡或严重伤害和 / 或设备损坏。强制功能是调试程序的辅助工具，切勿为了弥补处理装置的故障而执行强制。仅限合格人员使用强制功能。强制程序数值后，务必通知所有授权维修或调试程序的人员。

使用“程序状态”按钮或“调试”菜单命令，开始状态数据通讯，并启用调试工具。

您还可以使用硬件输入模拟器开关（西门子（Siemens）可提供）以手动方式在数字输入打开和关闭之间切换，并观察 PLC 上的 LED 指示灯，观察数字输出的条件。

从“调试”工具条或“调试”菜单列表存取以下程序状态功能。



**强制** 您可以在操作程序状态时从程序编辑器和状态图强制地址。欲强制程序状态地址，用鼠标右键单击一个参数，并选择从菜单列表强制。欲强制状态图地址为某一数值，您必须首先规定所需的数值，方法是读取该数值（如果您希望强制当前值）或键入该数值（如果您希望将地址强制为一个新数值）。您一旦使用了强制功能，则在每次扫描时该数值均被重新应用于地址，直至您取消强制地址。“强制”功能盖写立即读取或立即写入指令。“强制”功能还盖写被配置为在转换为 STOP（停止）模式时进入一个指定数值的输出：如果 CPU 进入 STOP（停止）模式，输出被设为强制数值，而非非配置的数值。通过将 V 或 M 内存强制为字节、字或双字，将 AI 或 AQ 内存存在偶数字节边界上（例如 AIW6 或 AIW14）或 I/O 点上（I 和 Q 位址）强制为字，模拟逻辑条件。您可以一次强制 16 个（V、M、AI 或 AQ）地址和所有的 I/O 位（所有 AI 和 Q 位址）。 该图标表示该地址被明确强制。该地址数值在地址被取消强制之前无法改变。 该图标表示该地址被隐含强制。如果地址是一个被明确强制的较大地址的一部分，该地址则被认为是隐含强制。例如，如果 VW0 被强制，则 VB0 是隐含强制（VB0 是 VW0 的第一个字节）。隐含强制数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制 VD0（该地址包含 VB0、VB1、VB2 和 VB3），则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制，则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。 该图标表示该地址的一部分被部分强制。例如，如果 VW0 被明确强制，则 VW1 的一部分被强制（VW1 的第一个字节是 VW0 的第二个字节）。被部分强制的数值无法自身取消强制。您必须取消强制的谄磺恐频牡 刁罚 玫 刁肥 挡拍 茆 谋 洄 H 卷 蹦 3. 航岸 寥：： 星恐 啤泵 钚币 陨先 璧急 晨 丛 诘 刁 放 缘 摹 暗 鼻 笛 怠 绷 兄 邢 允 荆 蚋 玫 刁 肺 幢 磺 恐 啤



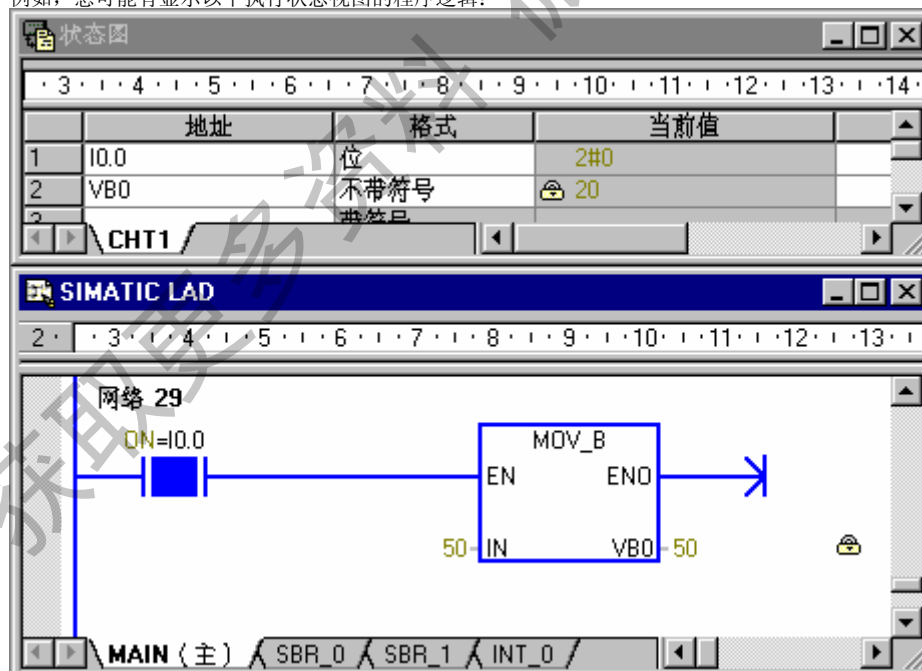
**取消强制** 对于程序状态和状态图，选择一个地址，并使用“取消强制”按钮从该特定地址移除强制功能。您还可以选择（单击）一个参数，然后用鼠标右键单击该参数，查看“强制”和“取消强制”功能的弹出菜单。

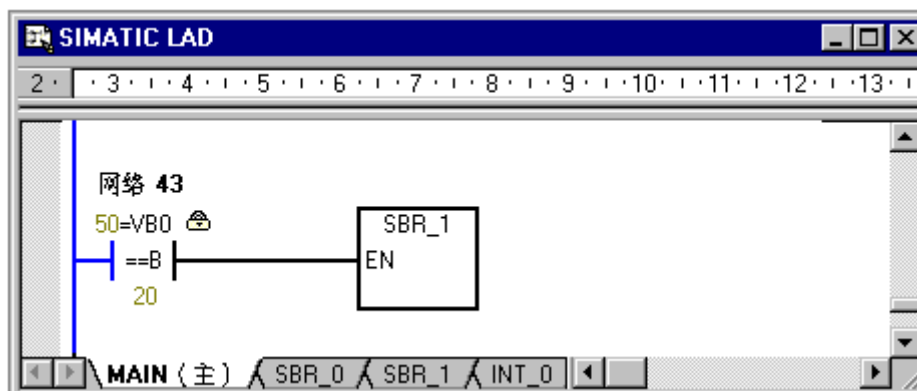


**全部取消强制** 如果您希望从所有的地址中移除强制功能，使用“全部取消强制”按钮。在应用“全部取消强制”之前，您无须选择单个地址。

#### 无法预测的强制数值程序操作

“强制”功能是一种循环软件强制，并非硬件强制。您需要理解如何对每种内存类型进行强制，才能有效地使用“强制”功能。以下范例显示 LAD 程序状态，但 STL 和 FBD 状态视图具有相同的行为方式。例如，您可能看到以下执行状态视图的程序逻辑：

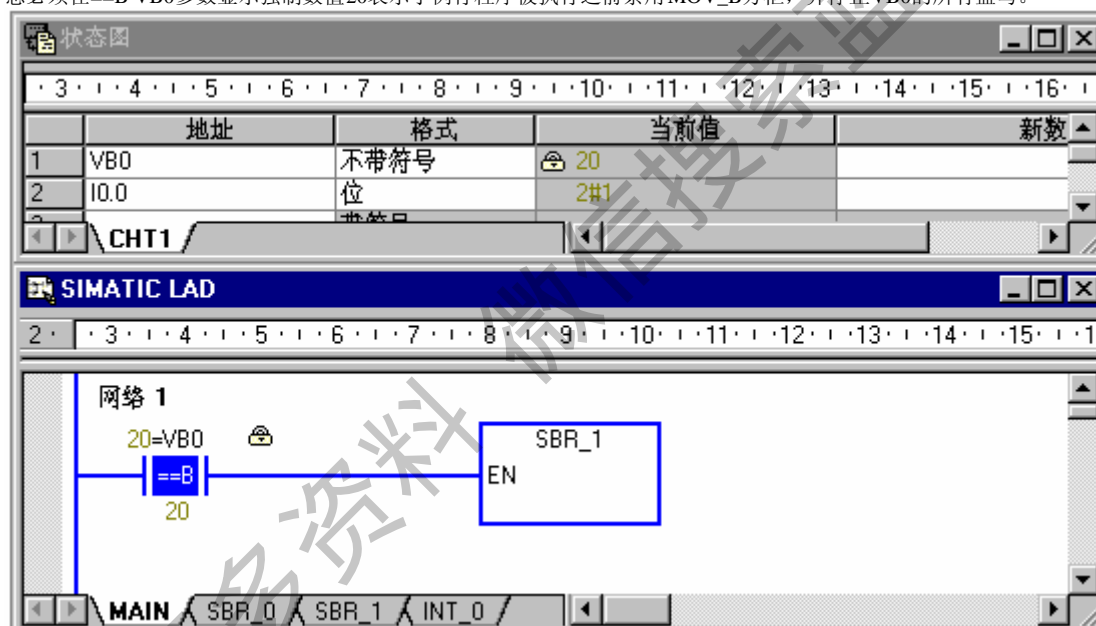




如果您正在查看该程序的网络43，而且不记得VB0数值是如何被网络29盖写，您可能认为为了激活SBR1可以将VB0强制为20。但请注意网络29用MOV\_B指令将VB0数值设为50。PLC执行网络43时，VB0包含数值50，所以不会激活SBR1。即使您建立状态图

VB0键入新数值20，并应用“强制”命令，只有在评估程序逻辑后，才能在程序扫描结尾将VB0数值设为20。VB0是V内存数担 谄拦莱缘蚱呢 虻哦 V内存地址应用强制命令。此种行为使程序状态显示在数值20已经被强制时显示一个VB0数值50。

您必须在==B VB0参数显示强制数值20表示子例行程序被执行之前禁用MOV\_B方框，并停止VB0的所有盖写。



#### 注释：

以上范例显示LAD执行状态结果。“扫描结束状态”模式具有相关的行为。如果“扫描结束”状态值被使用，程序状态在强制数值20的位置显示VB0。但当网络43被PLC实际执行时，VB0可能不包含强制数值；因此，强制VB0可能不会对SBR1产生所需的效

如果您在调试程序并发现带有强制操作数的网络没有按照您期待的方式执行，使用 **交叉引用 (Cross Reference)** 或 **编辑 (Edit) > 查找 (Find)** 命令，查找其他可能引用并改变操作数数值的网络。

#### 调节程序状态显示

[返回顶端](#)

#### LAD和FBD程序状态首选项

当LAD或FBD程序状态现用时，使用菜单命令 **工具 (Tools) >**

**选项 (Options)** 并选择“程序编辑器”标记，修改以下程序状态设置：

- 栅格宽度 (图形尺寸)
- 符号编址

“栅格宽度”和“符号编址”选项可以以不同的方式设置，用于程序状态和程序编辑器操作。其余的选项可使用程序状态打开或关闭功能设置。

- 状态显示
- 字体（类型、样式和大小）：在选择字体之前选择类别。请注意，并非所有的字体均支持所有的字体大小。选择“色彩”标记，修改用于显示状态信息的色彩。

### STL程序状态首选项

您可以按照用在“选项”对话框的“STL状态”标记中编辑设置的方法定制状态被报告的方式。选择菜单命令 **工具 (Tools) > 选项 (Options)** 并单击“STL状态”标记。

“观察状态”复选框允许您在程序状态显示中包括或从程序状态显示中移除操作数、堆栈数值和指令状态位（即旗标）。操作数、逻辑堆栈和指令状态位域允许您指定显示的操作数或逻辑堆栈数值数目或显示的状态位。

当您使用垂直滚动按钮时，程序代码行和程序状态显示中的行同时前进。滚动时，由于需要从 PLC 获取的数据量，程序状态显示会滞后于程序代码行。您可以通过调整“操作数”列尺寸，使之能够纳入较小的显示区域。您无法更改“逻辑堆栈”或“指令状态位”列，也无法更改“观察状态”复选框。您可以通过调整“观察状态”复选框，使之能够纳入较小的显示区域。您无法更改“逻辑堆栈”或“指令状态位”列，也无法更改“观察状态”复选框。

### 状态色彩

选择 **工具 (Tools) > 选项 (Options)** 色彩标记，并修改“状态”类别，改动默认色彩赋值。

此外，您可以重新排列 STEP 7-

Micro/WIN 中的其他窗口，以便为程序编辑器窗口安排更多空间（或以便与另一个窗口（例如状态图、符号表或交叉引用窗口）并排显示）。

### 返回顶端

#### 另请参阅：

[调试和监控特征概述 \(GS 7.1\)](#)

[如何在状态图中显示状态 \(GS 7.3\)](#)

[如何执行有限次数扫描 \(GS 7.4\)](#)

[如何下载程序 \(GS 6.3\)](#)

[时间标记不匹配错误](#)（确保编程设备中的项目与 PLC 中的项目相匹配）

[交叉引用和元素用法](#)（确保程序编辑不引起重复赋值）

[在运行模式中编辑程序](#)

[PLC 运行 / 停止模式](#)

[在停止模式中写入和强制输出](#)

## 1.8.3 7.3 如何在状态图中显示状态

您可以建立一个或多个状态图，在将程序下载至 PLC 之后监管和调试程序操作。打开一个图，检视或编辑图的内容。开启图，收集数据。

在控制程序的执行过程中，状态图数据的动态改变可用两种不同方式检视：

### 图状态


在一表格中显示状态数据：每行指定一个要监视的 PLC 数据值。您指定一个内存地址、格式、当前值及新值（如果使用写入命令）。

**趋势显示** 用随时间而变的 PLC 数据之绘图跟踪状态数据：

您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势数据亦可在趋势视图中直接赋值。

使用以下一种方法打开状态图：

- 单击 **浏览条** 上的“状态图”按钮。
- 选择 **检视 (View) > 元件 (Component) > 状态图 (Status Chart)** 菜单命令。
- 打开 **指令树** 中的“状态图”文件夹，然后双击“图”图标。
- 如果您在项目中有一个以上状态图，使用位于“状态图”窗口底部的“图”标记

 在图之间移动。


如果您已经打开一个空状态图，编辑图行，增加您希望监管的 PLC 数据地址。

使用以下一种方法启动在状态图中载入 PLC 数据的通讯：

- 欲连续收集状态图信息，开启状态图：使用菜单命令 **调试 (Debug) > 图状态 (Chart**

**Status)** 或使用“图状态” 工具条按钮。

- 欲获得单个数值“瞬态图”，使用“单次读取”功能：使用菜单命令 **调试 (Debug) > 单次读取 (Single**

**Read)** 或使用“单次读取” 工具条按钮。（但是，如果您已经开启图状态，“单次读取”功能被禁用。）



**提示：** • 打开图并不意味着检视状态。您必须开启图，才能收集状态信息。

- 如果图空置，开启图没有任何意义：您必须首先将程序数值（操作数）放在“地址”列中，并为“格式”列中的每一个数据选择数据类型，“建立”图。
- 您可以用于STOP（停止）模式的PLC收集状态，检查初始或最终条件。PLC必须位于RUN（运行）模式，才能从连续执行的程序收集数据。

如果PLC位于RUN（运行）模式，程序在连续扫描的状况下执行。您可以开启图状态，连续更新状态图数值。作为替代开启图状  
牧俄恢址椒ā 梢允褂漫暗ゴ味寥 愈δ埽 占 刺 际 档牡ゞ航八蔡 肌薄  
您在检视状态图时，还可以将PLC设为STOP（停止）模式，使用“首次扫描”或“多次扫描”功能，监管有限数目扫描的程序操作。



**图状态提示：** • 无论PLC位于何种操作模式，锁定始终会显示。

- 如果PLC位于RUN（运行）或STOP（停止）模式，可使用写入、强制和取消强制功能。但是，在STOP（停止）模式中，写入输出（Q和AQ）选项只有在**调试（Debug）> 在停止中写入-强制输出（Write-Force Outputs in Stop）**菜单项目被勾选时才能使用。
- 当PLC位于RUN（运行）或STOP（停止）模式时，状态数值被显示。
- 强制和取消强制工具条按钮选项仅在用于带有V、M、AI和AQ内存类型的字节、字和双字长度时才被启用
- 强制和取消强制工具条按钮和弹出菜单选项在用于带有I和Q内存类型的位、字节、字和双字长度时被启用。工具条按钮只有在所有选项可以被适当强制/取消强制（多行选项时如此）时才会变成现用。
- 强制和取消强制工具条按钮和弹出菜单选项在用于所有完全或部分隐含强制数值 时被禁用。
- 明确强制数值可以被强制为一个新数值，而无须取消强制后再强制新数值。

本标题讨论下列主题：

[状态图举例](#)  
[建立图](#)  
[使用多个状态图](#)  
[数据格式](#)  
[单次读取与连续图状态](#)  
[写入与强制数值](#)  
[在状态图中使用调试功能](#)  
[编辑捷径](#)  
[趋势显示详细说明](#)

### 状态图举例

[返回顶端](#)

打开状态图窗口与启动状态图不同。您可以打开状态图进行编辑或检查，但除非您（从“调试”菜单或工具条）发出“单次读取”命令或（从“调试”菜单或工具条）启动状态图，否则不会在“当前数值”列中显示状态信息。

利用“单次读取”特征（只有在关闭图状态时才能使用）检查状态图时，从PLC收集当前数值，并在“当前数值”列显示，但PLC执行程序时并不对其进行更新。

（从“调试”菜单或工具条）启动图状态时，在连续循环的基础上从PLC收集当前数值。从PLC接收到改变时，对“当前数值”  
薪 懈 隆

您可以利用“新数值”列指定（写入或强制）您确定的一个数值。如果“新数值”欲空置，并且在“当前数值”域中存在一个数值，则当前数值被强制。

	地址	格式	当前数值	新数值
1	起始_1	位	2#1	
2	起始_2	位	2#0	
3	停止_1	位	2#1	
4	停止_2	位	2#0	
5	高位	位	2#0	
6	低位	位	2#0	
7	重设	位	2#0	
8		带符号		
9	泵_1	位	2#1	
10	泵_2	位	2#0	
11	混合器_端	位	2#0	
12	蒸汽阀	位	2#0	
13	排放阀	位	2#1	
14	排放泵	位	2#1	
15		带符号		
16	高位_已达到	位	2#1	
17	混合_定时器	带符号	+7426	
18	循环_计数器	带符号	+0	

CHT1

### 建立图

(返回顶端)

在状态图中，您可以输入地址或定义符号名，从程序监管或修改数值。计时器或计数器数值可以显示为位或字。如果您将计时器或计数器数值显示为位，则会显示输出状态（输出打开或关闭）。如果您将计时器或计数器显示为字，则使用当前数值。欲建立状态图，遵循以下步骤。

1. 在“地址”域中为每个需要的数值输入地址（或符号名）。

PLC内存地址范围中列出的大多数内存类型有效，数据常量、累加器和高速计数器除外。

欲编辑地址单元格，使用箭头键或鼠标选择您希望编辑的单元格。

如果您开始键入，域被清除，输入新字符。

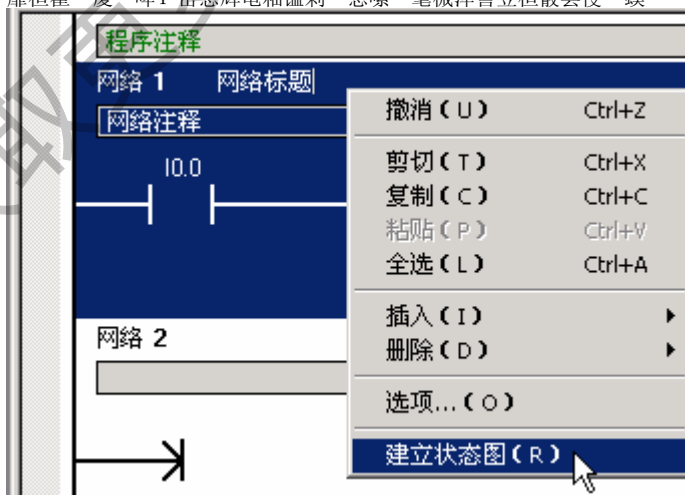
如果您单击鼠标或按“F2”键，域被增亮，您可以使用箭头键将编辑光标移至您希望编辑的位置。

地址	
1	I0.0

或

增亮程序代码的一部分，用鼠标右键调出弹出菜单，并选择命令 **建立状态图 (Create Status Chart)**。

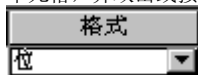
新图包含可以收集状态的选择区域中每个独特操作数的一个条目。条目在程序中按照其出现的顺序排列，图获



每次您选择 **建立状态图 (Create Status Chart)** 时，只能增加头150个地址。

**注释：**  
一个项目最多可存储32个状态图。

2. 如果元素是位（例如，I、Q或M），格式在“格式”列中被设为位。如果元素是字节、字或双字，选择“格式”列中的单元格，并双击或按SPACEBAR或ENTER键，在有效格式中循环，直至显示适当的格式。



欲插入附加行，使用“编辑”菜单或用鼠标右键单击状态图中的一个单元格，调出弹出菜单，并选择菜单命令 **插入 (Insert) > 行 (Row)**。新行被插入在状态图中光标当前位置的上方。您还可以将光标放在最后一行的任何一个单元格中，并按下箭头键，在状态图底部插入一行。



**提示：** 您可以从符号表选择地址，并将地址复制至状态图，更快地建立图。

- 您可以建立多个状态图。这样可允许您将元素分为逻辑群组，以便每个群组可以在一个不同的图中检视。则可不必在一个包括全部元素的长列表中滚动。


### 使用多个状态图

[\(返回顶端\)](#)

欲建立一个新状态图，核实“图状态”已经关闭，然后：

- 从指令树，用鼠标右键单击“状态图”文件夹，并选择弹出菜单命令 **插入 (Insert) > 图 (Chart)**。
- 打开状态图窗口，并使用“编辑”菜单或用鼠标右键单击，调出弹出菜单，选择 **插入 (Insert) > 图 (Chart)**。

**注释：**

- 成功地插入新状态图后，在状态图窗口底部会出现一个新标记 ，并在指令树
- 有时位于标记部分右侧的滚动按钮可能会遮住标记。如果看不到标记，拖曳标记区域和滚动按钮之间的分界线，以便看到更多的标记。

### 数据格式

[\(返回顶端\)](#)

您为数值指定的数据格式决定如何在状态图中显示该数值。在下例中，VD0（以及VB0和V0.0）包含数值1。

地址	格式	当前数值
1	位	2#1
2	二进制	2#0000_0001
3	不带符号	1
4	带符号	+1
5	十六位进制	16#01
6	ASCII	'01'
7	字符串	'\$01\$00\$00\$01\$00\$00\$00\$00\$00\$0'
8	浮点数	2.350989E-038





**提示：** 位和二进制数值前面都带有数字2和井号（#）。十六进制数值前面带有数字16和井号（#）。带符号和不带符号的数值均使用十进制数。

写入与读取相反。当您读取字符串格式的VB地址时，第一个字节首先读取，这代表字符串的长度。然后再读取其他字符串。鼻爸邓孀址 旧矜 隆H缙 粤俄恢指裕较允 VB地址，则会看到长度字节数值。字符串长度被写入第一个字节，其后是字符串本身。

### 单次读取与连续图状态

[\(返回顶端\)](#)

- 欲获得数值的单次“瞬态图”，使用“单次读取”功能：使用菜单命令 **调试 (Debug) > 单次读取 (Single**

- Read**) 或使用“单次读取”工具条按钮。(但是如果您已经启动图状态,“单次读取”功能则被禁用。)
- 欲连续收集状态图信息,启动图状态:使用菜单命令 **调试 (Debug) > 图状态 (Chart Status)** 或使用“图状态”工具条按钮。

### 写入与强制数值

[\(返回顶端\)](#)

“全部写入”功能允许您向程序写入一个或多个数值,模拟一种条件或一系列条件。然后您可以运行程序或使用状态图(以及程序状态(如果需要))监控运行状况。执行程序时,您用“全部写入”功能修改的数值可能被新数值盖写。

“强制”功能允许您模拟逻辑条件,方法是强制 V(字节、字或双字)、M(字节、字或双字)、AI(字)或AQ(字)或实际条件(通过强制配备 I 和 Q 位址的 I/O 点)。当您在状态图中使用“强制”功能强制一个或多个程序数值时,每次扫描循环时都会有效地重置数值。虽然在单次扫描过程中,程序可修改强制数值,但新扫描开始时,会重新应用强制数值。由于 PLC 执行速度很快,

PLC 向您的编程设备传输状态信息和在编程设备屏幕中显示信息所需的延迟时间很长,您可能无法观察到强制数值的任何改变。但是,在单个扫描过程中可以观察到此种改变。



**提示:** 如果您将一个数值写入 I/O 点,您可能永远不会看到该数值在状态图中作为当前值显示,因为在下一次扫描时程序可能写入一个不同的数值(与您状态图中写入的数值不同)。使用“强制”功能将数值指定给 I/O 更加有效。

- 图状态允许您在“新数值”列键入一个所需的数值。请记住,当 PLC 收到强制命令时,PLC 会将操作数强制为当前值。这是一个快速的程序,但不是瞬时程序。操作数数值有可能在您单击 STEP 7-Micro/WIN 中的状态显示时间和强制命令被 PLC 收到时间之间发生变化。一旦锁定图标在操作数旁显示,您可以看到在 PLC 中被强制的数值。如果数值未被正确强制,您可以使用状态图在“新数值”列中键入您希望使用的确切数值,并从图状态强制操作数。




**警告** 在程序中强制数值时,在程序每次扫描时将操作数重设为该数值,与输入/输出条件或其他正常情况下对操作数值有影响的程序逻辑无关。强制可能导致程序操作无法预料,无法预料的程序操作可能导致人员死亡或严重伤害和/或设备损坏。强制功能是调试程序的辅助工具,切勿为了弥补处理设备的故障而执行强制。仅限合格人员使用强制功能。强制程序数值后,务必通知所有授权维修或调试程序的人员。

### 在状态图中使用调试功能

[\(返回顶端\)](#)

您可以在程序状态操作过程中从程序编辑器和状态图将新数值写入或强制至操作数,用此种方法模拟进程条件。

使用“图状态”按钮或“调试”菜单命令,开始状态数据通讯。

您可以从“调试”菜单或“调试”工具条存取以下“调试”功能。





**单次读取** (仅限状态图) 欲获得“瞬态图”,即对全部数值程序状态的一次更新,使用“单次读取”功能。在默认情况下,图状态从 PLC 连续获取状态更新信息。单击状态图且图状态关闭时,则启用“单次读取”按钮。



**全部写入** (仅限状态图) 对状态图中的“新数值”列的改动完成后,您可以使用“全部写入”,将所需的改动传送到 PLC。



**强制** 欲将地址强制为某一数值,您必须首先规定所需的数值,可通过读取数值(如果您希望强制当前数值)或键入数值(如果您希望将地址强制为新数值)来完成。一旦您单击“强制”按钮,每次扫描都会将数值重新应用于该地址,直至您对该地址执行取消强制。“强制”功能盖写立即读取或立即写入指令。“强制”功能还盖写被配置为在转换为 STOP(停止)模式时进入一个指定数值的输出:如果 CPU 进入 STOP(停止)模式,输出被设为强制数值,而非非配置的数值。通过将 V 或 M 内存强制为字节、字或双字,将 AI 或 AQ 内存存在偶数字节边界上(例如 AIW6 或 AIW14)或 I/O 点上(I 和 Q 位址)强制为字,模拟逻辑条件。您可以一次强制 16 个(V、M、AI 或 AQ)地址和所有的 I/O 位(所有 AI 和 Q 位址)。 该图标表示该地址被明确强制。该图单元格数值在地址被取消强制之前无法改变。 该图标表示该地址被隐含强制。如果地址是一个被明确强制的较大地址的一部分,该地址则被认为是隐含强制。例如,如果 VW0 被强制,则 VB0 是隐含强制(VB0 是 VW0 的第一个字节)。隐含强制数值无法自身取消强制。您必须取消强制较大的地址,然后才能改变该地址数值。如果您强制 VD0(该地址包含 VB0、VB1、VB2 和 VB3),则被计数为



可以强制的16个内存数值之一。如果您将VB0、VB1、VB2和VB3作为分开的实体强制，则计

数为您可以强制的16个内存数值中的四个数值。所有被强制的数值均存储在CPU的永久性EEPROM内存中。

该图标表示该地址的一部分被部分强制。例如，如果VW0被明确强

制，则VW1的一部分被强制（VW1的第一个字节是VW0的第二个字节）。被部分强制的数值无法自身取消强制。您必须取消强制的该强制频率。您可以通过以下方式取消强制：星型泵、钼币、先、紧急、从、放、缘、摹、暗、鼻、色、总、绷、兄、那、允、荆、蚋、玫、肺、幢、磺、恐、啤



**取消强制** 对于程序状态和状态图，选择一个地址并使用“取消强制”从该特定地址移除强制。

您也可以选择（单击）一个参数，然后用鼠标右键单击该参数，调出“强制”和“取消强制”功能弹出菜单。



**全部取消强制**（仅限状态图）如果您希望从全部地址移除强制，使用“全部取消强制”功能。应用“全部取消强制”之前不必选择单个地址。



**读取全部强制**（仅限状态图）使用“读取全部强制”特征时，状态图的“当前数值”列为已经被明确强制、隐含强制或部分隐含强制的所有地址显示一个图标。

### 图编辑捷径


[\(返回顶端\)](#)

- 欲插入或附加一个具有下一个顺序地址和相同数据格式的新行，选择一个“地址”单元格，并按 **ENTER** 键。
- 欲在特定地址的所有可能数据格式中循环，选择“数据格式”单元格，并连续按 **ENTER** 键。欲检视所有可供使用的选项，按 **斜键**。
- 欲移至图的下一个单元格，按 **TAB** 键。
- 欲调整列宽，将鼠标指针放在列边缘，直至光标变成“调整大小”光标，然后将列边缘拉至所需的位置。
- 欲选择一整行（用于剪切或复制），单击行号。
- 欲插入一个新行，选择一个单元格或用鼠标右键单击。选择下拉菜单命令 **插入 (Insert) > 行 (Row)**。行被插入在光标位置；其后的行则向下移动一行。
- 欲在图当前最后一行下方插入一行，将光标放在最后一行的任何一个单元格中，并按下箭头键。
- 欲删除一个单元格或行，增亮单元格或行，并用鼠标右键单击。选择弹出菜单命令 **删除 (Delete) > 选项 (Selection)**。如果删除一行，其后的行（如果有）则向上移动一行。
- 欲选择整个状态图，在行号上方的左上角单击一次。

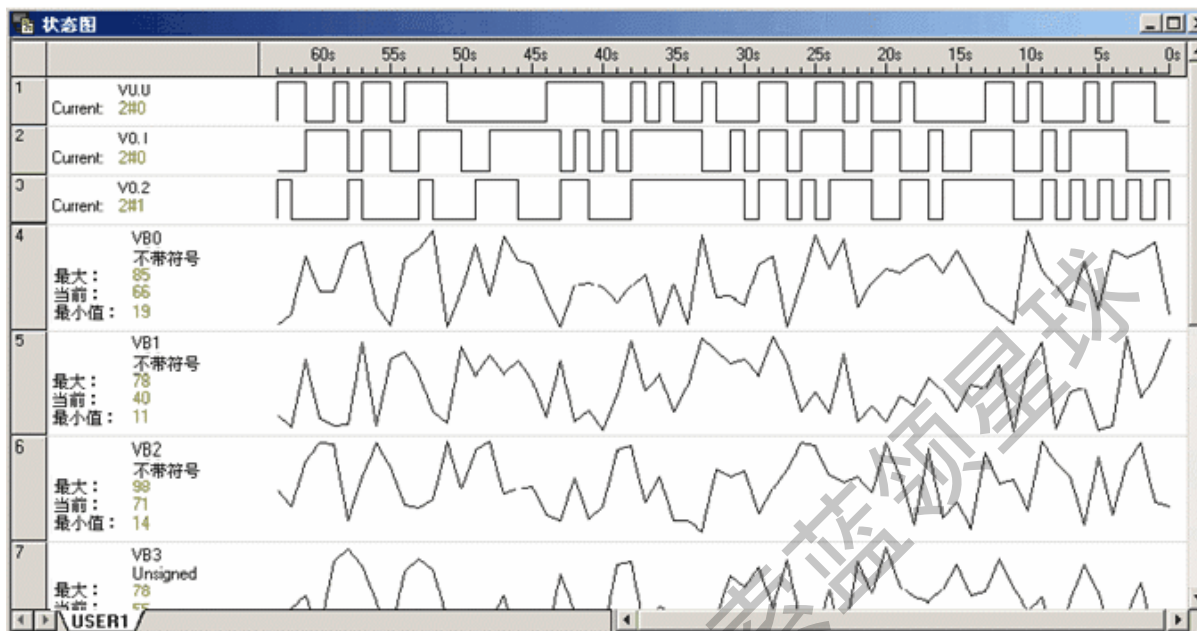
### 趋势显示详细说明




[\(返回顶端\)](#)

您可以使用下列方法在状态图的表格视图和趋势视图之间切换：

- 使用菜单命令 **检视 (V) > 检视趋势 (T)**。
- 用鼠标右键单击状态图，然后复选或取消复选 **作为趋势检视** 命令。
- 单击调试工具条的趋势视图按钮 

趋势图用随时间而变的PLC数据之图形绘图跟踪状态数据：您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势图



倘若在选取切换趋势按钮之前没有选取图状态按钮，供趋势功能使用的状态收集将不会开始。如要启动PLC数据收集，请单击趋势按钮。

您可以通过单击暂停趋势视图按钮或选取**调试(D) > 暂停趋势图(P)**命令，凝固趋势绘图。

#### 趋势视图词汇表

**趋势显示** 随时间而变的PLC数据之图形绘图，通过选取“切换趋势”按钮由预配置的状态图创建。

**趋势绘图**

就在状态图中已配置的每个地址而言，该地址的PLC数据之屏幕位置会以时间为自变量得到显示。您不能调整趋势绘图的宽度。

**数据点或趋势点**

数据点意指出自PLC中某地址的单个数据。数据点是在相应地址的趋势绘图中赋予最后一个像素的数值。

**Range**

趋势数据的绘图在最小值和最大值之间多个数据的一个跨度上做出。此范围通常定义了数值的某个固定跨度。

**自由范围**

如果没有给趋势绘图选取范围，则最小值和最大值之间的数据范围会动态变化。

**抽样速率或收集速率**

定义由PLC收集数据的速率。此速率将是固定的，并且将基于选定的趋势时间单位计算得出。

**趋势窗口**


用于描述对趋势数据进行绘图的时间窗口；换句话说，由趋势绘图屏幕宽度代表的时间跨度。

**实时趋势**

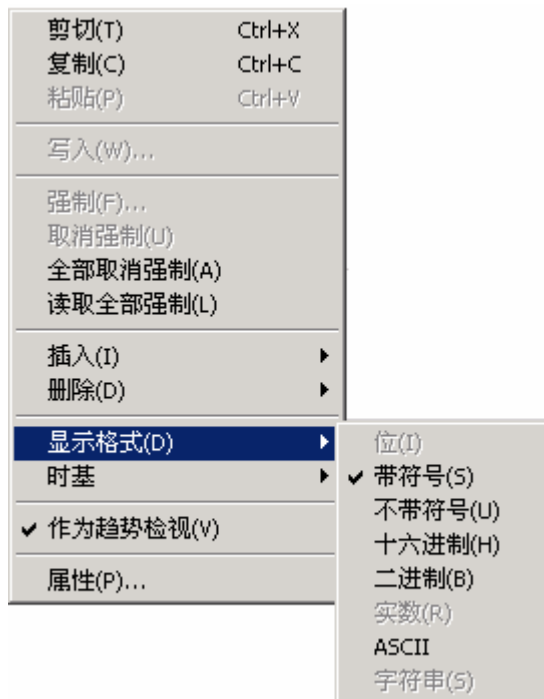
实时趋势意指以实时方式收集和在趋势窗口中显示数据。一旦此数据填满趋势窗口，最先的数据就会从趋势绘图上消失。

趋势图的单次读取和写入按钮与状态图上的相应按钮作用相同。‘强制’、‘取消强制’、‘全部取消强制’及‘读取全部强制’的作用与在状态图中相似。

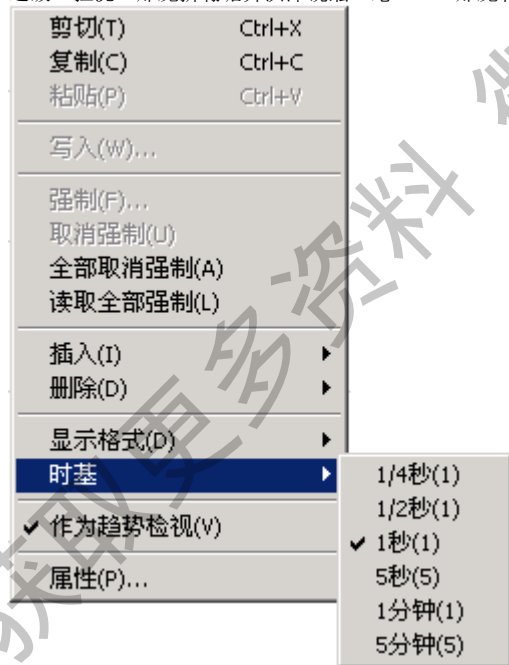
实时趋势功能不支持历史趋势。这意味着超出一个趋势窗口之时间跨度的趋势数据不会得到保留。

将光标放在分隔趋势行的横线上直至出现双箭头光标。在光标为双箭头时，向上拖动以减少或向下拖动以增加高度。

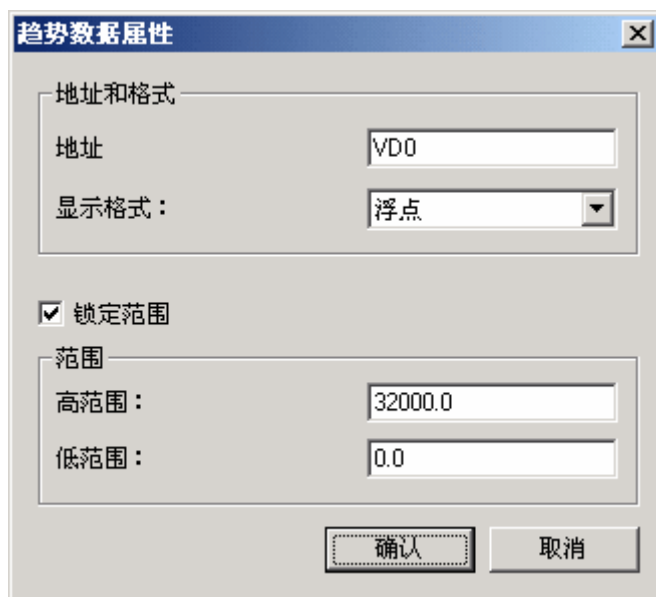
在单击并选定一趋势行后，用鼠标右键单击会显示一弹出菜单，由此可指定该行的显示格式。变更显示格式及添加或删除行也会改变状态图的表格视图。



您可以用秒或分为单位设置趋势绘图的时基，其默认值为1秒。配置的时基会在趋势图上用s（秒）或m（分）显示，其位置为送级ノ打浇 那魔拼翱谗奔淇潭饶淞H纒 那魔仆嫉氛被 鑿嫉氛 荻蓟岜磺究 (15)睐碌氛被 阨李允尽



在单击并选定一趋势行后，用鼠标右键单击会显示一弹出菜单，由此可指定此行的趋势数据属性。



由“趋势数据属性”对话框，您可以对趋势绘图做出下列变更：

- 编辑地址（例如将VW0更改为VW10）。
- 增加一个新地址表 如果选定的点以前没有得到配置。
- 如前面的上下文菜单讨论所述，更改显示格式类型。
- 基于选定的显示格式设置范围。

如果显示格式被修改，所有数据将会重新显示，并且可能因选定的新格式而改变。如果这些格式为数值兼容，例如带符号或不带符号，此趋势绘图将只会由可能的范围更改而改变。如果有些数值现在是在旧范围之外，并且范围已锁定，那么此绘图将在最大或最小范围值处截短。如果此绘图配置为自由范围，那么如有必要，这些数据将使用新的范围值重新显示。

按照默认规定，固定范围跨度不会启用；因此，这些趋势图绘图将为自由范围。如要能够针对给定趋势点选取范围值，请参见【段 A 备用】[额微钝 奋烁凶】](#) [螭旅姻挠蚩山邮丛涿舜](#)

由此对话框对地址或显示格式做出的任何更改将同时影响状态图和趋势图。

另请参阅：

[调试和监控特征概述 \(GS 7.1\)](#)

[如何在程序编辑器中显示状态 \(GS 7.2\)](#)

[如何执行有限次数扫描 \(GS 7.4\)](#)

[如何下载程序 \(GS 6.3\)](#)

[时间标记不匹配错误 \(确保编程设备中的项目与 PLC 中的项目相匹配\)](#)

[交叉引用和元素用法 \(确保程序编辑不引起重复赋值\)](#)

[在运行模式中进行程序编辑](#)

[PLC 运行 / 停止模式](#)

[在停止模式中写入和强制输出](#)

## 1.8.4 7.4 如何执行有限数目的扫描

您可以指定 PLC 对程序执行有限次数扫描（从 1 次扫描到 65,535 次扫描）。通过选择 PLC 运行的扫描次数，您可以在程序改变进程时指定扫描次数。第一次扫描时，SM0.1 数值为 1（打开）。

**执行单次扫描：**

1. PLC 必须位于 STOP（停止）模式。如果不是已经位于 STOP（停止）模式，[将 PLC 转换成停止模式](#)。
2. 从菜单条选择 **调试 (Debug) > 首次扫描 (First Scan)**。


**执行多次扫描：**

1. PLC 须位于 STOP（停止）模式。如果不是已经位于 STOP（停止）模式，[将 PLC 转换成停止模式](#)。
2. 欲执行多次扫描，从菜单条选择 **调试 (Debug) > 多次扫描 (Multiple Scans)**。出现“执行扫描”对话框。



3. 输入所需的扫描次数数值，单击“确定”，确认您的选择并取消对话框。

注释：

当您准备好恢复正常程序操作时，请务必将PLC转回RUN（运行）模式（单击工具条中的“运行”按钮或选择PLC > 运行（RUN））。

另请参阅：

[调试和监控特征概述 \(GS 7.1\)](#)  
[如何在编辑器窗口中显示状态 \(GS 7.2\)](#)  
[如何在状态图中显示状态 \(GS 7.3\)](#)

[如何下载程序 \(GS 6.3\)](#)  
[时间标记不匹配错误 \(确保编程设备内的项目与PLC内的项目相匹配\)](#)  
[交叉引用和元素用法 \(确保程序编辑不引起重复赋值\)](#)  
[在运行模式中进行程序编辑](#)  
[PLC运行 / 停止模式](#)  
[在停止模式中写入和强制输出](#)

## 1.9 第8章 如何管理项目

### 1.9.1 8.1 如何打印

预览

欲在纸张上实际打印之前预览项目打印页面，选择文件 (File) > 打印预览 (Print

Preview) 菜单命令。单击“打印预览”工具条按钮，或单击“打印”对话框中的“打印预览”按钮。

“打印预览”对话框提供下列选项：

- 一次检视相当于一页或并排两页的程序。
- 改变POU标记，检视不同的程序块。
- 前进至下一页；返回至前一页。
- 增加（放大）或减少（缩小）程序显示，用于预览目的。
- 使用浏览条在不同的编辑器之间转换。

主打印选项

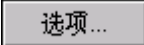
使用以下一种方法，打印程序和项目文档的复制件。

- 单击“打印”按钮。
- 选择文件 (File) > 打印 (Print) 菜单命令。
- 按Ctrl+P快捷键组合。

主要“打印”对话框提供下列选项：

- 选择打印机
- 彩色或黑白打印
- 打印单个项目元件中的一系列网络或行
- 打印多个项目元件
- 改变打印顺序

打印选项

从主“打印”对话框，选择“选项”按钮，存取“打印选项”对话框。单击一个标记，修改打印选项。选择或

“打印选项”对话框提供下列标记：

- 程序编辑器标记：打印程序组织单元的作者和时间戳记属性；打印所有选择的程序组织单元的局部变量表；在一个新页面上打印大的LAD和FBD网络。
- 数据块标记：打印数据块的属性信息。

#### 页面设置选项

选择菜单命令文件 (File) > 页面设置 (Page

Setup) 或从主“打印”对话框中单击“页面设置”  按钮，以便存取“页面设置”对话框。

“页面设置”对话框允许您配置以下选项：

- **页边距** — 调整打印件的左、右、上、下和页眉 / 页脚边距
- **页面方向** — 选择纸张尺寸和纸张来源。调整页面方向（纵向或横向）

“页面设置”对话框包括“页眉 / 页脚”按钮 ，该按钮允许您输入“页眉 / 页脚”文本。您还

另请参考：

[打印预览](#)  
[打印](#)

## 1.9.2 8.2 如何移动、复制、重新命名和邮寄项目

在STEP 7-Micro/WIN中，项目包含在带“.mwp”扩展名的单个文件内。这使您能够利用标准 Windows Explorer程序和电子邮件移动、复制、重新命名或邮寄项目。

您还可以使用项目文件中的 [另存为](#) 对话框更改项目名称或目录位置。

另请参阅：

[如何复制程序段](#)  
[导入文件](#)  
[导出文件](#)

## 1.9.3 8.3 如何从PLC上载

您可以使用工具条按钮或文件菜单，从PLC将程序上载至运行STEP 7-Micro/WIN的个人计算机中。

#### 上载单块或全部三个块

您可以上载程序块（OB1、子例行程序和中断例行程序）、系统块和数据块；另外，您也可以仅上载三个块之一。PLC不包含了解详细信息。）

#### 上载至新的空项目

这是捕获程序块、系统块和 / 或数据块信息的保险方法。由于项目空置，所以您无法反向损坏数据。如果您希望使用为该项目 Micro/WIN，并从另一个项目文件复制该信息。（详情请参阅 [如何复制程序段](#)。）


#### 上载至现有项目


如果您希望盖写自下载至PLC以来对程序进行的全部修改，这是一个好办法。


如果您需要保留下载至PLC之后对程序块、系统块和 / 或数据块所作的任何修改，则不应采用这种方法，因为上载会盖写这些

#### 步骤：

1. 打开STEP 7-Micro/WIN中的一个项目，容纳您将从PLC上载的块。

\* 如果您希望上载至一个空项目，选择文件 > 新，或使用“新项目”  工具条按钮。

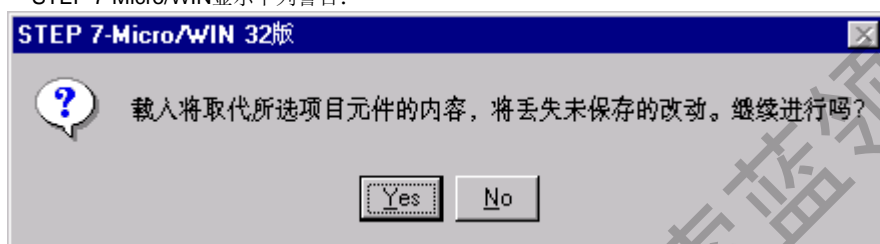
\* 如果您希望上载至现有项目，选择文件 > 打开，或使用“打开项目”  工具条按钮。

2. 选择文件 > 上载，或使用“上载”  工具条按钮，初始化上载程序。

3. “上载”方框显示程序块、数据块和系统块复选框。请核实已选择您希望上载的块复选框，并取消选择您不希望上载的任何块，然后单击“确认”。



4. STEP 7-Micro/WIN显示下列警告：



单击“是”。

从PLC成功地将块上载至个人计算机时，STEP 7-Micro/WIN显示一则讯息。

另请参阅：

[密码](#)

[上载（文件 > 上载）](#)

[通讯概述（GS 6.1）](#)

[如何测试通讯网络（GS 6.2）](#)

[如果下载程序（GS 6.3）](#)

[下载（文件 > 下载）](#)

[在运行模式中进行程序编辑](#)

## 1.9.4 8.4 如何复制程序段

使用“编辑”菜单命令和鼠标或标准Microsoft®键组合，例如CTRL + A（全选），CTRL + C（复制），CTRL + X（剪切）和CTRL + V（粘贴）。您还可以在网络中用鼠标右键单击，并选择“复制”。



**提示：** 粘贴材料时，将材料插入光标当前位置上方的行或格中。

本标题讨论下列主题：

[复制一段网络或一个POU](#)

[复制符号表或状态图信息](#)

[复制数据块信息](#)

[局部变量表](#)

[使用两个STEP 7-Micro/WIN实例，在项目之间复制](#)

[使用一个STEP 7-Micro/WIN实例，在项目之间复制](#)

### 复制一段网络或一个POU

[（返回顶端）](#)

- 您可以复制LAD、FBD或STL程序编辑器中的一段网络或整个POU。
- 只需转换位于程序编辑器窗口底部的标记，即可将选项粘贴至程序的不同POU中。
- 您也可以将选项粘贴至不同的项目中。



**提示：**

为每项剪切 / 复制 / 粘贴操作所选择的网络范围必须是连续范围：必须互相接触。如果您希望从POU的不同部分复制几个网络或网络集，则必须对每个单独的网络选项重复执行剪切 / 复制和粘贴操作。

**复制符号表或状态图信息**[\(返回顶端\)](#)

- 您可以从符号表复制符号名或地址信息，并粘贴至状态图的“地址”列。
- 您可以从符号表或状态图复制所有列中的信息，并将其粘贴至 Microsoft Excel®工作单，或从Excel将适当的列中的信息粘贴至符号表或状态图。

**复制数据块信息**[\(返回顶端\)](#)


- 您可以复制数据块信息，并将其粘贴至文本编辑器。
- 您可以从文本编辑器将信息粘贴至数据块。但是，如果信息违反语法规则，数据块编辑器会出示红色警告 X。

**局部变量表**[\(返回顶端\)](#)

您不能在局部变量表之间成块复制数据，因为每个表格的只读 L 内存赋值必须保持独特性。

**使用两个STEP 7-Micro/WIN实例，在项目之间复制**[\(返回顶端\)](#)

执行下列步骤：


1. 打开两个STEP 7-Micro/WIN实例。（双击STEP 7-Micro/WIN图标，或从“开始”菜单选择 **Simatic > STEP 7-Micro/WIN**。重复该步骤。）
  2. 在第一个STEP 7-Micro/WIN实例中，使用 **文件 > 打开**，打开您希望复制的项目。
  3. 选择材料，并使用 **CTRL + C** 或 **编辑 > 复制**，进行复制。
  4. 在第二个STEP 7-Micro/WIN实例中，使用 **文件 > 打开**，打开您希望粘贴材料的项目。
  5. 将光标放在适当的位置，使用 **CTRL + V** 或 **编辑 > 粘贴**，粘贴材料。
- 注释：** 请记住在关闭项目之前保存作业。

**使用一个STEP 7-Micro/WIN实例，在项目之间复制**[\(返回顶端\)](#)

执行下列步骤：

1. 使用 **文件 > 打开**，打开您希望复制的第一个项目。
  2. 选择材料，并使用 **CTRL + C** 或 **编辑 > 复制**，进行复制。
  3. 使用 **文件 > 关闭**，关闭第一个项目。
  4. 使用 **文件 > 打开**，打开第二个项目，即您希望将粘贴材料在其中的项目。
  5. 将光标放在适当的位置，使用 **CTRL + V** 或 **编辑 > 粘贴**，粘贴材料。
- 注释：** 请记住在关闭项目之前保存作业。

**1.9.5 8.5 较早版本的STEP 7-Micro/WIN、 Micro/DOS项目**

欲使用在STEP 7-Micro/WIN或STEP 7-Micro/DOS早期版本中建立的项目，单击“打开”，或选择 **文件 > 打开**，并选择需要的项目。

**注释：**

- 用STEP 7-Micro/WIN或STEP 7-Micro/DOS早期版本建立的项目可能包含一个或多个STEP 7-Micro/WIN 3.0版或更高版本不支持的逻辑结构。欲成功地打开项目，您必须使用建立该项目的较早版本程序包，并按照下列步骤
1. 将程序编辑器转换成STL。
  2. 关闭符号编址。



### 3. 保存项目文件。

完成上述步骤后，即可在3.0版或更高版本中成功地打开项目。

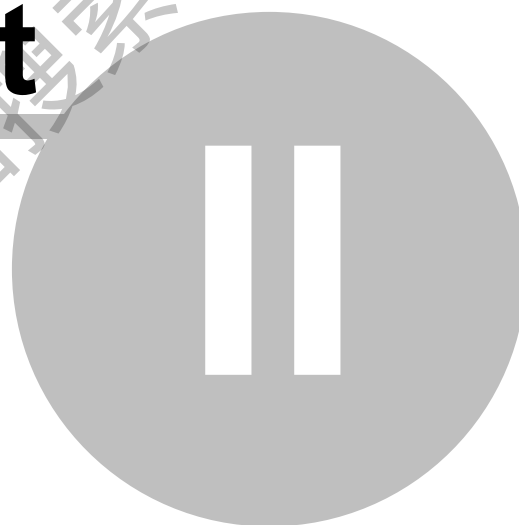
- 在STEP 7-Micro/WIN V3.1 SP1中建立、在FBD中使用单输入AND（与）指令并在FBD视图中保存的程序无法用STEP 7-Micro/WIN V3.1打开。为了能用STEP 7-Micro/WIN V3.1打开这些项目，必须首先将项目转换成STL视图并保存。
- 您不能使用“打开”命令打开位于PLC中的项目；项目文件必须位于您的个人计算机 / 编程设备中。
- 每个STEP-7 Micro/WIN实例只能打开一个项目。您必须运行两个STEP 7-Micro/WIN实例，才能同时打开两个项目。打开两个实例时，您可以在两个项目之间复制和粘贴 LAD/FBD程序元素和STL文本。

获取更多资料 微信搜索 蓝领星球

## Top Level Intro

This page is printed before a new  
top-level chapter starts

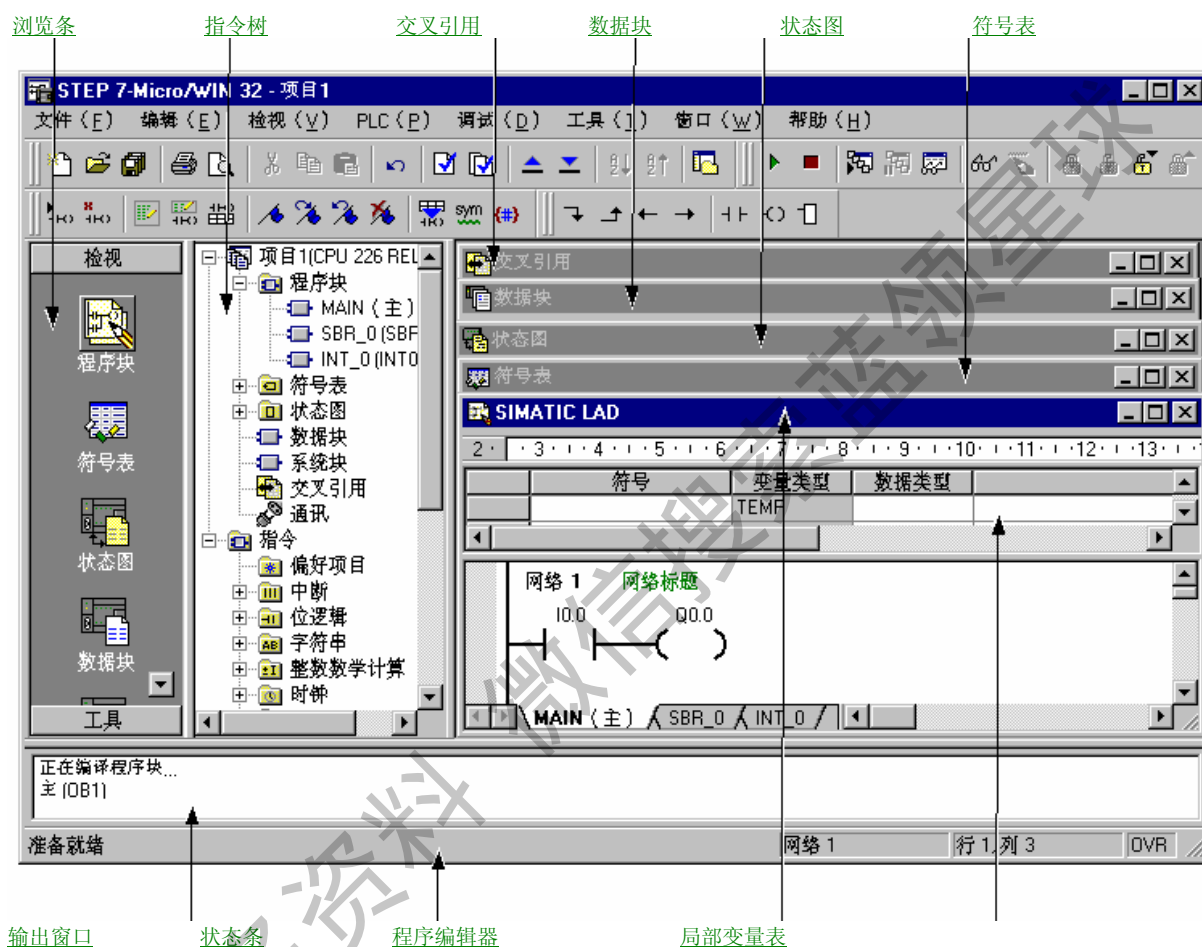
# Part



获取更多资料 微信搜索 蓝领星球

## 2 应用程序用户参考手册

### 2.1 窗口元件



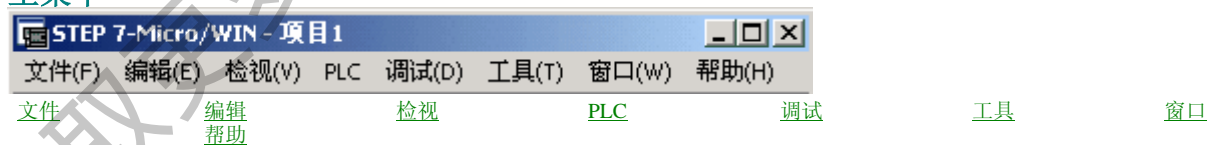
输出窗口

状态条

程序编辑器

局部变量表

### 主菜单



快捷键  
快捷键

### 工具条共同特征

在可以使用工具时，工具条按钮就会被启用

例如，从菜单选择LAD toolbar is active when 检视 (View) >

Ladder (梯形图) 时，LAD工具条变为现用。如果工具条按钮被禁用，则显示为灰色。

#### 移动工具条

如果您将鼠标指针放在工具条帧内的区域，但在任何按钮上，则可以单击和拖拉工具条。可在主窗口中移动工具条，亦可将工具条移至主窗口外。将工具条移至主窗口外时，工具条变成另一个窗口。

您可以选择检视 (View) > 工具条 (Toolbars) > 全部重设 (Reset

All) 菜单命令，在主窗口中将工具条恢复至原来的位置。

#### 工具条上下文帮助链接

1) 欲查看工具条按钮名称，将鼠标指针在工具条按钮上移过，然后停下。会显示按钮名称。

2) 欲了解有关工具功能的详情，按SHIFT+F1，将  鼠标指针放在一个工具条按钮上，然后单击。

标准工具条



调试工具条



获取更多资料 微信搜索 蓝领星球



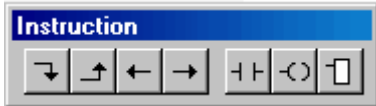
公用工具条



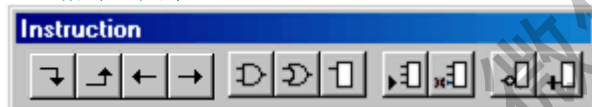
获取更多资料 微信搜索 蓝领星球



LAD指令工具条



FBD指令工具条



获取更多资料 微信搜索 蓝领星球

## 向导

[指令向导](#) [HSC指令向导](#) [NETR/NETW指令向导](#) [PID指令向导](#) [PID调谐控制面板](#) [TD-200向导](#) [TD 200 SIMATIC文字显示功能](#) [TD 200向导 \(TD 200 v2.1或之前版本\)](#) [TD 200向导 \(TD 200 v3.0或TD 200C v1.0\)](#) [调制解调器扩充向导](#) [EM241调制解调器模块性能](#) [EM241调制解调器向导](#) [位置控制向导](#) [EM253位置模块性能](#) [EM253位置向导](#) [EM253控制面板](#) [以太网向导](#) [CP243-1以太网模块性能](#) [CP243-1以太网向导](#) [互联网向导](#) [CP243-1 IT互联网模块性能](#) [CP243-1 IT互联网向导](#) [AS-i向导](#) [CP243-2 AS-i模块性能](#) [CP243-2 AS-i向导](#) [菜谱向导](#) [菜谱向导数据日志](#) [数据日志向导](#)

## 库

[USS协议库](#) [USS协议库](#) [Modbus协议库](#) [Modbus协议库](#) [用户定义库](#) [用户定义库](#)

## 错误讯息

[STEP 7-Micro/WIN编译程序错误](#) [通讯错误](#) [LAD/FBD编辑器错误](#) [项目文件I/O错误](#) [PLC严重错误](#) [PLC编译和运行时间错误](#) [PLC非严重错误](#) [内存位置](#) [时间标记不匹配](#) [EM241调制解调器模块错误](#) [EM253位置模块错误](#) [CP243-1以太网 / 互联网模块错误](#) [和ETHx\\_CTRL](#) [CP243-1信道和ETHx-XFR/ETHx\\_CFG错误](#) [CP243-1 IT](#) [ETHx EMAIL错误](#) [CP243-1 IT ETHx FTPC错误](#) [CP243-2 AS-i模块错误](#) [USS协议指令错误](#) [Modbus协议指令错误](#)

## 2.4 快捷键

您可以在STEP 7-Micro/WIN中使用快捷键，更有效地完成多项任务。

[LAD](#)  
[FBD](#)  
[STL](#)  
[符号表 / 全局变量表](#)  
[局部变量表](#)  
[状态图](#)  
[程序状态](#)  
[在绝对编址模式和符号编址模式之间变换](#)



**提示：** 如果列出多个键，所有的键均须一起按下，快捷键才会有效。

### LAD和FBD快捷键

#### 返回顶端

#### 1. 浏览键

##### 键排列顺序

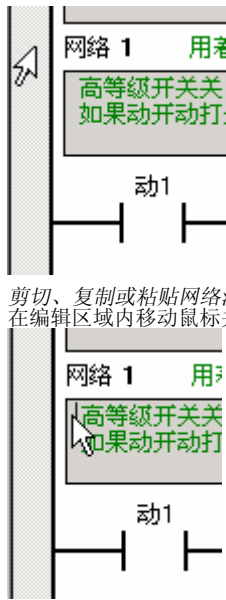
##### 措施

HOME	将光标移至同行的第一列
END	将光标移至同行的最后一列
PAGE UP	垂直向上移动一个屏幕
PAGE DOWN	垂直向下移动一个屏幕
LEFT ARROW	将光标向左移动一个单元格
RIGHT ARROW	将光标向右移动一个单元格
UP ARROW	将光标向上移动一个单元格
DOWN ARROW	将光标向下移动一个单元格
CTRL + HOME	将光标移至第一个网络的第一个单元格
CTRL + END	将光标移至最后一个网络的最后一个单元格
F1	当前单元格指令的帮助
CTRL + PAGE UP	显示下一个POU，向左移动浏览POU标记
CTRL + PAGE DOWN	显示下一个POU，向右移动浏览POU标记
TAB	选择带布尔（位值）参数指令时，TAB键移至该指令中的下一个布尔参数（仅限FBD）当您选择一个带布尔参数指令时，TAB键将选项移至该指令中的下一个布尔参数（仅限FBD）

#### 2. 编辑

##### 鼠标拖过或快捷键网络选择

**剪切、复制或粘贴网络：**  
 将鼠标移动到网络区域左边的装订线上，然后单击以设置网络上下文。



剪切、复制或粘贴网络注解：  
在编辑区域内移动鼠标并单击，以设置注解上下文。

#### 键排列顺序

SHIFT + 上箭头  
SHIFT + 下箭头  
SHIFT + PAGE UP  
SHIFT + PAGE DOWN  
CONTROL + SHIFT + HOME  
CONTROL + SHIFT + END  
CONTROL + SHIFT + END

#### 措施

向上扩展选择  
向下扩展选择  
向上扩展网络选择  
向下扩展网络选择  
扩展选择至第一个网络  
扩展选择至最后一个网络  
选择全部网络

CTRL + X SHIFT + DELETE 选择网络系列时，剪切系列；当光标位于网络标题上时，剪切整个网络；当光标位于单元格上时，剪切单元格内容  
CTRL + C CTRL + INSERT 选择网络系列时，复制系列；当光标位于网络标题上时，复制整个网络；当光标位于单元格上时，复制单元格内容  
CTRL + V SHIFT + INSERT 选择网络系列时，粘贴系列；当光标位于网络标题上时，粘贴网络；当光标位于单元格上时，粘贴单元格。  
在光标当前位置的前方 / 上方粘贴所选内容

SPACEBAR 编辑当前单元格的助记符或操作数  
ENTER 编辑当前单元格的助记符或操作数  
CTRL + LEFT ARROW 放下一条水平线，将光标向左移动一个单元格  
CTRL + RIGHT ARROW 放下一条水平线，将光标向右移动一个单元格  
CTRL + UP ARROW 放下一条垂直线，将光标向上移动一个单元格  
CTRL + DOWN ARROW 放下一条垂直线，将光标向下移动一个单元格  
DELETE 删除当前单元格、网络或网络选项  
BACKSPACE 将光标向左移动一个单元格，删除当前单元格  
SHIFT + LEFT ARROW 选择当前网络，将光标向左移动一个单元格  
SHIFT + RIGHT ARROW 选择当前网络，将光标向右移动一个单元格  
SHIFT + HOME 选择当前网络，将光标置于当前行的第一个单元格处  
SHIFT + END 选择当前网络，将光标置于当前行的最后一个单元格处  
SHIFT + DOWN ARROW 从当前网络向下选择  
SHIFT + UP ARROW 从当前网络向上选择  
SHIFT + PAGE UP 从当前网络向上选择相邻网络的一个页面  
SHIFT + PAGE DOWN 从当前网络向下选择相邻网络的一个页面  
CTRL + SHIFT + HOME 从光标至POU顶部选择全部相邻网络



CTRL + SHIFT + END	从光标至POU底部选择全部相邻网络
CTRL + A	选择POU中的全部网络
F11	将输入更改为求反输入 / 切换复原 (仅限FBD、布尔值)
CTRL + F11	将输入或输出更改为立即类型 / 切换复原 (仅限FBD、布尔值)
PLUS SIGN	增加一个附加输入参数 (仅限FDB、AND/OR (与 / 或) 指令)
MINUS SIGN	删除一个附加输入参数 (仅限FDB、AND/OR (与 / 或) 指令)
CTRL + Y	在符号和绝对编址模式之间切换
CTRL + T	显示每个网络的符号信息表

### 3. 放下指令

键排列顺序	措施
F4	包含所有接点助记符类型的列表框 (仅限LAD)
F4	放下AND (与) 框 (仅限FBD)
F6	包含所有线圈助记符类型的列表框 (仅限LAD)
F6	放下OR (或) 框 (仅限FBD)
F9	包含所有方框助记符类型的列表框

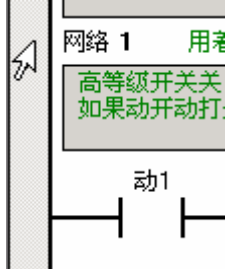
### STL快捷键

#### 返回顶端

#### 鼠标拖过或快捷键网络选择

剪切、复制或粘贴网络:

将鼠标移动到网络区域左边的装订线上, 然后单击以设置网络上上下文。



剪切、复制或粘贴网络注解:

在编辑区域内移动鼠标并单击, 以设置注解上下文。



#### 键排列顺序

SHIFT + 上箭头	向上扩展选择
SHIFT + 下箭头	向下扩展选择
SHIFT + PAGE UP	向上扩展网络选择
SHIFT + PAGE DOWN	向下扩展网络选择
CONTROL + SHIFT + HOME	扩展选择至第一个网络
CONTROL + SHIFT + END	扩展选择至最后一个网络
CONTROL + SHIFT + END	选择全部网络
CTRL + A	全选
CTRL + C	复制选项
CTRL + F	查找字符串
CTRL + H	替换字符串

CTRL + V	粘贴选项
CTRL + X	剪切选项
CTRL + Z	撤销
DELETE	删除选项
SHIFT + DELETE	将选项剪切至剪贴板
F2	移至下一个书签的位置
CTRL + F2	在光标的当前位置设为书签
CTRL + F3	插入网络
CTRL + F4	删除网络
F3	查找下一个字符串
BACKSPACE	退格
SHIFT + BACKSPACE	退格
ALT + BACKSPACE	撤销
INSERT	切换盖写
CTRL + INSERT	复制选项
SHIFT + INSERT	粘贴选项
HOME	移至当前行开始
CTRL + HOME	移至文件顶部
SHIFT + HOME	从光标当前位置选择至同一行的开始
CTRL + SHIFT + HOME	从光标位置选择至文件顶部
END	移至当前行结尾
CTRL + END	移至文件底部
SHIFT + END	从光标当前位置选择至同一行的结尾
CTRL + SHIFT + END	从光标位置选择至文件底部
LEFT ARROW	左移光标, 每次移动一个字符
CTRL + LEFT ARROW	左移光标, 每次移动一个字
SHIFT + LEFT ARROW	向左选择, 每次选择一个字符
CTRL + SHIFT + LEFT ARROW	向左选择, 每次选择一个字
RIGHT ARROW	右移光标, 每次移动一个字符
CTRL + RIGHT ARROW	右移光标, 每次移动一个字
SHIFT + RIGHT ARROW	向右选择, 每次选择一个字符
CTRL + SHIFT + RIGHT ARROW	向右选择, 每次选择一个字
UP ARROW	将光标向上移动, 一次移动一行
SHIFT + UP ARROW	向上选择, 一次选择一行
DOWN ARROW	光标下移, 一次移动一行
SHIFT + DOWN ARROW	向下选择, 一次选择一行
PAGE DOWN	光标下移一个屏幕的数据
SHIFT + PAGE DOWN	从当前光标位置向下选择一个屏幕的数据
PAGEUP	光标上移一个屏幕的数据
SHIFT + PAGE UP	从当前光标位置向上选择一个屏幕的数据
TAB	向前跳格
SHIFT + TAB	向后跳格
CTRL + Y	在符号和绝对编址模式之间切换
CTRL + PAGE UP	显示下一个POU, 向左移动浏览POU标记
CTRL + PAGE DOWN	显示下一个POU, 向右移动浏览POU标记

[符号表 / 全局变量表、  
局部变量表、状态图  
返回顶端](#)

#### 键排列顺序

CTRL + X  
CTRL + C

#### 措施

剪切选择的文本  
复制选择的文本

CTRL + V	粘贴选择的文本
DOWN ARROW	光标下移（在表尾处增加新行）
UP ARROW	光标上移
LEFT ARROW	光标左移
RIGHT ARROW	光标右移
DELETE	删除选择的文本
F1	在当前窗口中提供帮助
F2	允许您编辑当前的单元格
CTRL + A	选择所有的单元格
TAB	光标从左至右、从上至下移动
SHIFT + TAB	光标从右至左、从下至上移动
ENTER	校验单元格并右移光标
HOME	将光标移至第一列
END	将光标移至最后一列
CTRL + PAGE UP	显示下一个表 / 图。向左移动浏览符号表 / 状态图标记
CTRL + PAGE DOWN	显示下一个表 / 图。向右移动浏览符号表 / 状态图标记
SHIFT + ARROWS	选择文本
CTRL + I	在当前行上方插入新行
CTRL + SHIFT + I	在当前行下方插入新行

#### 程序状态快捷键

##### 返回顶端

F1	允许您不使用滑鼠存取菜单项目
----	----------------

#### 在绝对和符号编址模式之间变换

##### 返回顶端

#### 键排列顺序

#### 措施



CTRL + Y	在符号和绝对编址模式之间切换
CTRL + T	显示LAD或FBD程序每个网络的符号信息表

**注释：**对于LAD/FBD程序，您可以选择菜单命令 **工具 (Tools) > 选项 (Options)**，并单击“程序编辑”标记，仅检视符号编址，或在启用符号编址模式时，检视符号和绝对地址。

## 2.5 编程向导

### 2.5.1 HSC指令向导

欲使用HSC向导：  
选择菜单命令 **工具 (T) >**

**指令向导...**，然后选取HSC；或单击浏览条中的指令向导图标 ，然后选取HSC；或打开**指令树**中的“向导”文件夹并  
 蜻蟠蚩 HSC向导  **高速计数器**。

#### 步骤

- 1 **选择计数器类型和操作模式**
- 2 **指定初始参数**
- 3 **程序中中断事件 / 多个步骤**
- 4 **生成代码**

LAD或STL中的“高速计数器”指令（HSC）由S7-

200提供，控制速率超过PLC扫描速率的事件。HSC操作十分灵活，有多种确定计数器行为方式的初始化选项。欲以适当的方式配置HSC，您需要在SM内存中设置几个参数，为操作模式配置一个具体的HSC，将中断例行程序附加在具体的计数器事件上，最后启用计数器操作。

**注释：**在使用HSC向导之前，必须编译程序。如果您尚未编译程序，向导在HSC配置进程开始时提示您编译程序。

HSC向导向您查询初始化选项，然后为指定的配置生成程序代码和数据块代码。从第一个屏幕选择HSC后，向导向您提示下一步操作。

### 步骤 1 选择计数器类型和操作模式

从列表框（依赖PLC）选择须配置的计数器。

从列表框选择操作模式。您的计数器选择决定可用的模式。

### 步骤 2 指定初始参数

向导为子例行程序指定一个默认名称。您可以指定一个不同的名称，但请勿使用现有子例行程序名称。

为计数器CV和PV指定一个双字地址、全局符号或整数文字。

**注释：**如果您为PV或CV输入一个未定义的符号，则在定义符号之前无法执行步骤3。向导会提示您在“定义符号”对话框中定义符号。

当您为PV和CV选择地址或符号时，更新PV和CV的选项被禁用。您必须修改生成的代码，才能修改或传递数值。根据定义，这意味着代码是可重新载入。

指定初始计数方向。

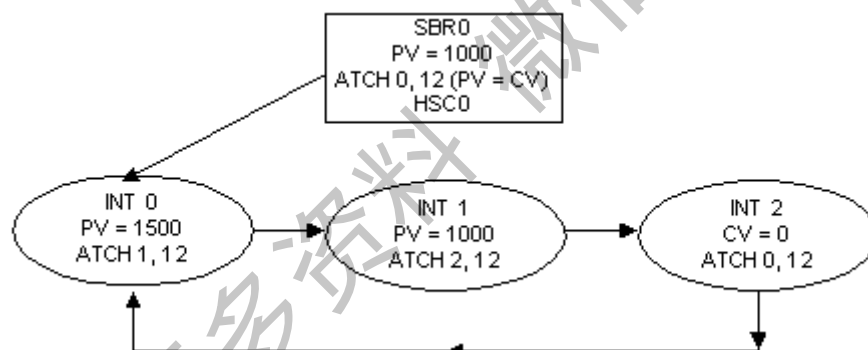
### 步骤 3 程序中断事件 / 程序多种步骤

您的计数器类型和操作模式选择决定可用的中断事件。

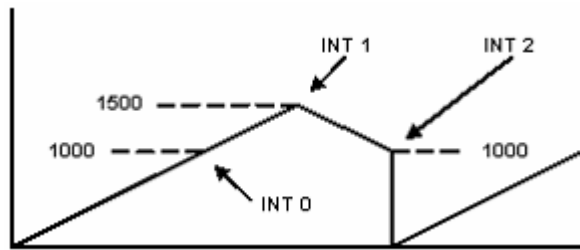
当您选择对当前数值等于预设值事件（CV = PV）进行编程时，向导允许您指定多种步骤计数器执行。

为了理解多种步骤HSC执行，请考虑下列内容：

在已编程的中断例行程序中，您可以选择将相同的中断事件重新连接至另一个中断例行程序。这将有效地在计数器操作性几个步骤，其中每个步骤改变HSC的一些动态参数，并将一个不同的中断例行程序连接至相同的中断事件。最后一个街码磨苯甬铝 釉 贾卸侠 谱缘颖 P 植苦仿罚 喂瞻懿皇侨编怨 O 吕 得饕恒龟蚘 暴(4) 霾街壁 HSC执行。以下是每个步骤的说明。



- SBR 0** 该子例行程序包含计数器初始化。计数器的当前值被指定为0（CV = 0）。计数器的预设值被指定为1000（PV = 1000）。计数方向为UP（向上）。事件12（HSC0 CV = PV）被连接至INT 0，计数器被启动。
- INT 0** 当计数器达到第一个预设值1000时，执行INT 0。计数器预设值被更改为1500，方向不变。事件12（HSC0 CV = PV）被重新连接至INT1，计数器被重新启动。
- INT 1** 当计数器再次达到预设值（1500）时，执行INT 1。此时，我们将预设值更改成1000（PV = 1000），将计数方向更改为DOWN（向下），将INT 1连接至事件12（HSC0 CV = PV），并重新启动计数器。
- INT 2** 当计数器向下计数至预设值1000时，执行INT 2。此时，我们将当前值设为0（CV = 0），并将计数方向更改为UP（向上）。事件12被重新连接至INT 0，至此则完成了计数器操作环路。下图显示计数器当前值随时间的变化。每个（CV = PV）中断事件均标有为该事件调用的INT例行程序。



时间 ->

#### 步骤 4 生成代码

完成HSC参数配置后，您可以检查计数器使用的子例行程序 / 中断例行程序列表。该列表仅供参考。

单击“完成”按钮，允许向导为HSC生成必要的程序代码。代码包括由用于高速计数器的初始化代码组成的子例行程序

注释：欲启用计数器操作，您必须从主程序中将调用放置在包含初始化代码的子例行程序中。使用SM0.1或边缘触发指

到繁 8 米永 谐绿蚪戳坏寝靡准巍

另请参阅：

[理解高速计数器指令](#)

[HSC指令](#)

[HDEF指令](#)

## 2.5.2 NETR/NETW指令向导

欲使用NETR/NETW向导：  
选择菜单命令工具 (T) >



指令向导...，然后选取NETR/NETW；或单击浏览条中的指令向导图标，然后选取NETR/NETW；或打开指令树中的“向

NETR/NETW

NET配置

起始地址 (VB332)

网络操作 (4已配置)

通讯端口 (0)

注释：本向导仅限于第二代CPU。如果您在一个3.1x项目中有一个向导配置，项目依然工作。但是，向导不会识别该项目，您

#### 步骤

1

指定您需要的网络操作数目

2

指定端口号码和子例行程序名称

- 3 指定网络操作  
4 分配V内存  
5 生成代码

为了帮助与网络连接的PLC之间的数据交换，S7-200支持网络读取 (NETR) 和网络写入 (NETW) 指令。NETR指令从远程PLC中的指定地址读取一定数量的配置数据。NETW指令向远程PLC中的指定地址写入一定数量的配置数据。NETR和NETW指令操作由数据表中的前七个字节控制。表格囊桓叁 凳嵌寥。慈胧 享と麟 梢允 1至16个字节。因此数据表总大小可达23个字节。

**注释：**在使用HSC向导之前，程序必须被编译程序，且位于符号编址模式。如果您尚未编译程序，向导在HSC配置进程开始时提示您编译程序。向导将向您查询初始化选项，然后为指定配置生成程序代码和数据块代码。从第一个屏幕选择NETR/NETW后，向导

### 步骤 1 指定您需要的网络操作数目

**注释：**如果项目包含使用STEP 7 Micro/WIN 3.2版建立的现有NETR/NETW配置，您必须在继续执行步骤1之前选择编辑其中一个现有配置或建立一个新配置。

可使用NETR/NETW指令向导，简化网络操作配置。向导向您查询初始化选项，并根据您的选择生成完整的配置。向导

### 步骤 2 指定端口号码和子例行程序名称

**注释：**项目可能包含一个NETR/NETW向导配置。所有以前建立的配置均被自动载入向导。向导会提示您完成以下两个

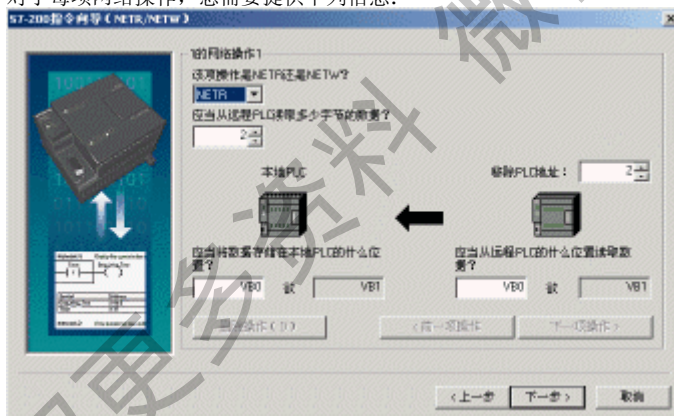
- 选择编辑现有配置，方法是单击“下一步>”。
- 选择从项目中删除现有配置，方法是选择“删除”复选框，并单击“完成”。

如果不存在以前的配置，向导会查询以下信息：

1. 必须被设为在PPI主模式中通讯。您指定操作将通过哪一个PLC端口通讯。
2. 向导建立一个用于执行具体网络操作的参数化子例行程序。向导还为子例行程序指定一个默认名称。您可以编辑默认名称。

### 步骤 3 指定网络操作

对于每项网络操作，您需要提供下列信息：



1. 指定操作是NETR还是NETW。
2. 指定从远程PLC (NETR) 读取的数据字节数或向远程PLC (NETW) 写入的数据字节数。
3. 指定您希望用于通讯的远程PLC网络地址。
4. 如果您在配置NETR，指定以下内容：
  - 数据存储在本地PLC中的位置。有效操作数：VB、IB、QB、MB、LB。
  - 从远程PLC读取数据的位置。有效操作数：VB、IB、QB、MB、LB。
5. 如果您在配置NETW，指定以下内容：
  - 数据存储在本地PLC中的位置。有效操作数：VB、IB、QB、MB、LB。
  - 向远程PLC写入数据的位置。有效操作数：VB、IB、QB、MB、LB。

### 步骤 4 分配V内存

对于您配置的每一项网络操作，要求有12个字节的V内存。您指定可放置配置的V内存起始地址。向导会自动为您建议一个地址，但您可以编辑该地址。请注意，如果您使用重叠地址，向导在显示一则警告后，允许您进入下一步。

**注释：**在NETR操作中，核实将存储数据的变量没有与分配给向导的内存地址重叠。

### 步骤 5 生成程序代码

回答这些查询后，单击“完成”，S7-

200指令向导将为您指定的网络操作生成代码。由向导建立的子例行程序成为项目的一部分。欲在程序中启用网络通讯，在主程序块中调用执行子例行程序（NET\_EXE）。每次扫描循环时，使用SM0.0调用该子例行程序。这样会启动配置网络操作执行。为每项网络操作建立的数据处理子例行程序会在适当时间被自动调用。

另请参阅：

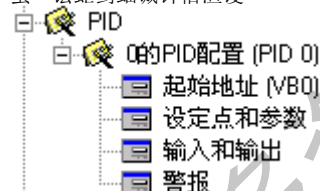
[NET\\_EXE指令](#)  
[标准NETR/NETW指令](#)

### 2.5.3 PID指令向导

欲使用PID向导：  
选择菜单命令**工具(T) >**



指令向导...，然后选取PID；或单击浏览条中的指令向导图标，然后选取PID；或打开[指令树](#)中的“向导”文件夹并随后



**注释：**

本向导仅适用于第二代CPU。如果您在一个3.1x项目中有一个向导配置，项目依然工作。但是，向导不会识别该项目，您无法编

**步骤**

- 1 指定环路号码
- 2 设置环路参数
- 3 环路输入和输出选项
- 4 环路警报选项
- 5 为计算指定内存区
- 6 指定初始化子例行程序和中断例行程序
- 7 生成代码

S7-

200指令向导的PID公式可用于简化PID操作配置。向导向您查询初始化选项，然后为指定配置生成程序代码和数据块

**注释：**在使用PID向导之前，程序必须被编译并位于符号编址模式。如果您尚未编译，向导会在PID配置进程开始提示您进行编译。

从第一个屏幕选择**PID**后，向导向您查询下列信息：

### 步骤 1 指定环路号码

**注释:** 如果项目包含使用STEP 7 Micro/WIN

3.2版建立的现有PID配置, 您必须在继续执行步骤1之前选择编辑其中一个现有配置或建立一个新配置。

您指定配置哪一个PID环路。

### 步骤 2 设置环路参数

**注释:** 参数表地址的符号名已经由向导指定。PID向导生成的代码使用参数表地址建立操作数, 作为参数表中的相对  
屏俊H继 问 森 卅方 7. 琶 缓蟾谋湮 梅 胖付ũ牡 卅罚 PID向导生成的代码则不再能够正确执行

环路定点是您为向导生成的子例行程序提供的一个参数。

1. 指定环路定点 (SP) 应当如何缩放。为“低范围”和“高范围”选择任何实数。默认值是 0.0和100.0之间的一个实数。
2. 指定下列环路参数:

- 增益时间
- 样本时间
- 积分时间
- 派生时间

欲了解环路参数如何影响PID计算性能, 请参阅[PID计算](#)。

### 步骤 3 设置环路输入和输出选项

环路进程变量 (PV) 是您为向导生成的子例行程序指定的一个参数。

向导向您查询以下环路输入选项:

1. 指定环路进程变量 (PV) 应当如何缩放。您可以选择:
  - 单极 (可编辑默认范围0至32000)
  - 双极 (可编辑默认范围-32000至32000)
  - 20% 偏移量 (设置范围6400至3200, 不可变更)

您还必须输入以下环路输出选项:

2. 指定环路输出应当如何缩放。您可以选择:
  - 输出类型 (模拟或数字)

**注释:** 如果您选择配置数字输出类型, 则必须以秒为单位输入“任务循环时间”。

- 缩放 (单极、双击或20%偏移量)

### 步骤 4 设置环路警报选项

该向导为各种环路条件提供输出。当达到警报条件时, 输出被设置。

指定您希望识别警报输入的哪些条件:

- 启用低警报 (PV) 并设置从0.0到高警报限制的正常化低警报限制
- 启用高警报 (PV) 并设置从低警报限制到1的正常化高警报限制
- 启用模拟输入模块错误, 并指定输入模块附加在 PLC 上的位置

### 步骤 5 为计算指定内存区

**PID指令**使用V内存中的一个36个字节的**参数表**, 存储用于控制环路操作的参数。PID计算还要求一个“暂存区”, 用于  
媧17.偈笨崩 D 枰 付ũ眉扑闹 嫉 V内存字节地址。

您还可以选择增加PID的手动控制。位于手动模式时, PID计算不执行, 环路输出不改变。

**注释:**

当PID位于手动模式时, 输出应当通过向“手动输出”参数写入一个正常化数值 (0.00至1.00) 的方法控制输出, 而不是用直接改变输出的方法控制输出。这样会在PID返回自动模式时提供无障碍式转换。

### 步骤 6 指定子例行程序和中断例行程序

**注释:**

如果项目包含一个现用PID配置, 已经建立的中断例行程序被列为唯读。因为项目中的所有配置共享一个公用中断例行  
缘颀 牟 恐性毘拥娜魏涡屣渲貌坏酶谋潞 弥卸侠 谐绦虻拿 啤

向导为初始化子例行程序和中断例行程序指定默认名称。您可以编辑默认名称。

### 步骤 7 生成PID代码

该屏幕显示PID向导生成的POU列表, 并对如何综合进程程序作出简要说明。

回答这些查询后, 单击“完成”, S7-

200指令向导将为您指定的配置生成程序代码和数据块代码。由向导建立的子例行程序和中断例行程序成为项目的一部分。欲在程序中启用该配置, 每次扫描循环时, 使用SM0.0从主程序块调用该子例行程序。该代码配置PID  
0. 该子例行程序初始化PID控制逻辑使用的变量, 并启动PID中断“PID\_EXE”例行程序。根据PID样本时间循环调用PID中断例行程序。

另请参阅:

[PIDx\\_INIT \(PID向导子例行程序\)](#)

[PID算法](#)

[PID环路表](#)




[PID调谐控制面板](#)**注释:**

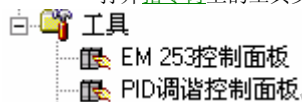
具有PID向导配置的项目不直接使用标准PID指令。如果您使用PID向导配置，您的程序必须使用CALL PIDxINIT来激活PID向导子例行程序。

[PID \(标准PID指令\)](#)

## 2.5.4 PID调谐控制板

使用以下一种方法打开PID调谐控制面板:

- 选取**工具(I) > PID调谐控制面板**菜单命令。
- 单击**浏览条**上的PID调谐控制面板  按钮。
- 打开**指令树**上的工具文件夹，然后打开PID调谐控制面板。

**PID调谐兼容性**

- PID环路自动调谐不能用于固件版本早于2.00发行版的控制器上的PID指令。
- 在早于STEP 7-Micro/WIN 4.0发行版的版本中使用PID向导创建或手工生成的PID指令，能够以手动输入调谐参数的校正方式进行调谐。这些PID指令无法使用自动调谐。
- 在早于STEP 7-Micro/WIN 4.0发行版的版本中使用PID向导创建的PID指令环路表不符合自动调谐格式。但是，如果您按照自动调谐格式修改环路表，则ID指令可有自动调谐算法在其上执行。

[理解自动调谐](#)[扩展的环路表](#)[先决条件](#)[自动滞后和自动偏差](#)[自动调谐序列](#)[例外条件](#)[关于PV超出范围的注释 \(结果代码3\)](#)[PID调谐控制面板](#)**理解自动调谐**

S7-200 PLC现已包含PID自动调谐能力，并且STEP 7-Micro/WIN也已添加一PID调谐控制面板。这两项功能相结合，就大大加强了S7-200 Micro PLC系列产品的用途和便利。

自动调谐能够由用户程序从操作员板发起，或由PID调谐控制面板发起。PID环路能够一次一个地得到自动调谐；如有必要，也可八个环路同时进行自动调谐。PID自动调谐会计算增益、积分时间（重设）、微分时间（速率）等调谐数值的建议（接近最优）数值，并且允许您选择调谐环路的快速、中速、慢速或很慢应答。

借助于PID调谐控制面板，您可以发起自动调谐进程、中止自动调谐进程、用图形方式监视其结果。此控制面板不仅显示可能产生的误差，还显示过程的动态特性。

S7-200使用的自动调谐算法，是基于K. J. 奥斯特和T.

Hägglund于1984年提出的一项称为继电器反馈的方法。在过去二十年里，继电器反馈已应用于范围广泛的行业。

继电器反馈的核心，是在一原本稳定的进程中产生一个小幅度但却持续的振荡。基于此振荡的周期和在进程变量中观察到的幅度改变，此进程的终极频率和终极增益可得到确定。然后，使用这些终极增益值和值，PID自动调谐器会就增益、重设、速率调谐提出建议。

所建议的数值取决于您选择的该进程环路的应答速度。您可以选取快速、中速、慢速或很慢等应答。取决于您的进程，快速应答可能过调且将对应于某欠阻尼调谐状态。中速应答可能濒临过调且将对应于某临界阻尼调谐状态。慢速应答可能没有任何过调且将对应于某过阻尼调谐状态。很慢应答可能没有过调且将对应于某严重过阻尼调谐状态。

自动调谐器除建议调谐值外，还能够自动确定滞后和高峰PV偏差的数值。这些参数用于在限制由自动调谐器建立的持续振荡之幅度的同时，减轻进程噪声的效应。

此PID自动调谐器能够为直接作用和反作用P、PI、PD、PID诸环路确定建议调谐值。

PID自动调谐器的目的，在于确定一组调谐参数，从而为您的环路之最优数值提供合理的近似。由建议调谐值开始，您将能够进行进一步的优化。

### 先决条件

您想要自动调谐的环路必须处于自动模式。环路输出必须由PID指令的执行来控制。如果环路处于手动模式，自动调谐将会失败。

在发起自动调谐操作前，您的进程必须已达到某种稳定态；也就是说，PV已经达到设定点（或就P型环路而言，PV与设定点的某误差在允许范围内波动）。

在开始自动调谐时，环路输出值最好能够在控制范围的中点附近。自动调谐过程会通过向环路输出中做出小步长改变，在进程中建立一个振荡。倘若环路输出很接近其控制范围的任何一端，由自动调谐过程引入的步进改变可能令输出值试图超出最小或最大范围极限。

如果发生此情况，就可能导致自动调谐错误状态的产生，并且肯定会得出较不理想的建议值。

### 自动滞后和自动偏差

滞后参数规定了PV（process variable、进程变量）相对于设定点的允许漂移（加或减）；此漂移不会令继电器控制器改变输出。此数值用于最小化PV信号中噪声的效应，从而更精确地确定此进程的固有振荡频率。

如果您选择自动确定滞后值，PID自动调谐器将会进入滞后确定序列。此序列包括对进程变量进行一段时间的抽样，然后计算抽样误差。

为了使抽样具有统计意义，样本集必须至少有100个样本。假设环路的抽样时间为200毫秒，那么取得100个样本需要20秒。抽样速率越高，取得100个样本所需的时间就越短。对于200毫秒的环路，尽管不需要20秒就可以取得100个样本，滞后确定序列总计需要20秒的样本。

在取得全部样本后，标准偏差将得到计算。滞后值被定义为标准偏差的两倍。此计算的滞后值将会写入环路表的实际滞后域（AHYS）。



### 提示：

在自动滞后序列工作过程中，正常的PID计算不会进行。因此，在发起自动调谐序列前，此进程处于稳定态是至关重要的。这样可避免不必要的过程扰动。

偏差参数规定了欲用的PV围绕设定点之峰到峰摆动。如果您选择自动确定此数值，欲用的PV偏差将为滞后值乘以4.5。输出将被限制在设定点±偏差范围内。

### 自动调谐序列

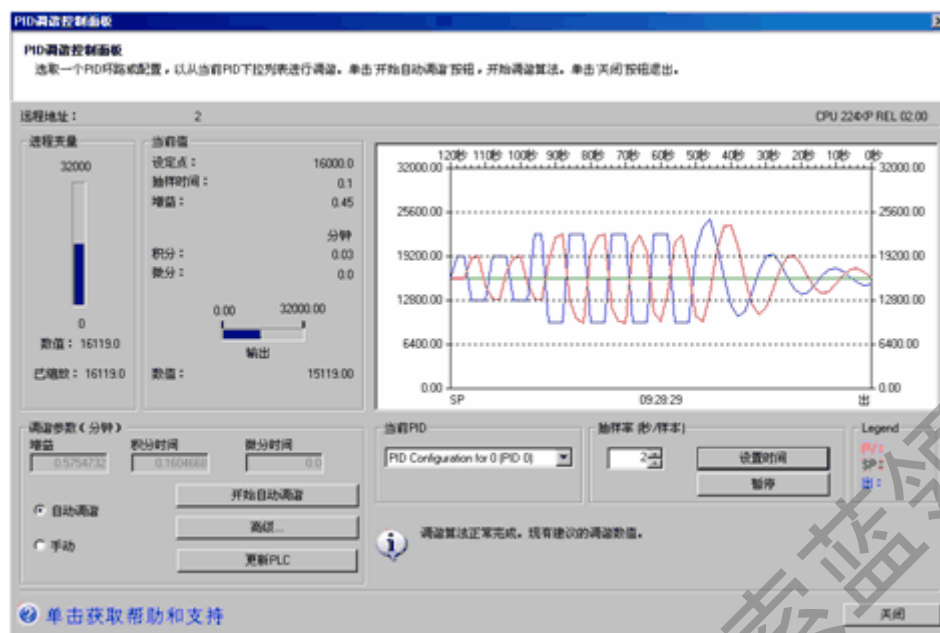
在确定了滞后值和偏差值后，自动调谐序列就会开始。此调谐过程开始于在环路输出上施加初始输出步长。

此输出值的改变应会导致进程变量值的相应改变。当输出变化将PV推离设定点足够远以至于超出滞后边界时，自动调谐器就会检测到一个过零事件。在每次发生过零事件时，自动调谐器将向相反方向推动输出。

调谐器会继续对PV进行抽样和等待下一个过零事件。此序列总共需要十二个过零才能完成。所观察到的峰到峰PV值幅度（峰值误差）和过零发生的速率与此进程的动态特性直接相关。

在自动调谐过程初期，输出步长值会按比例调整一次，从而令随后的PV峰到峰摆动更接近欲用的偏差值。一旦做出此调整后，新的输出步长值将写入环路表的“实际步长”域（ASTEP）。

如果两次过零之间的时间超出过零监视间隔时间，自动调谐序列将以错误告终。过零监视间隔时间的默认值为两小时。



上面的屏幕图象，显示了在某直接作用环路的自动调谐序列中，输出和进程变量的行为。PID调谐控制面板被用于发起和监视此过程。

请注意为了此过程经历小幅度振荡（如PV值所证实），自动调谐器是如何切换输出的。此PV振荡的频率和幅度，预示了进程增益和固有频率。

基于在自动调谐过程中收集的进程频率和增益信息，终极增益和终极频率将被计算得出，由此将计算增益（环路增益）、重设（积分时间）、速率（微分时间）的建议值。



**提示：**目前的环路参数确定自动调谐器将计算的调谐值。对于只有“增益”和“积分时间”的PI环路，远 饜称鹊 扑阙碌脑蚌婧突 质奔渲担 丁桐奈 7.质奔浣 0.0（无微分动作）。如果在“进程变量”值中有噪音，“微分时间”参数则无法正常作业。

#### 例外条件

调谐的执行过程可能产生三种警告状况。这些警告用环路表ASTAT域的三个位报告；这些位一经设置，在自动调谐序列被再次发

- 如果偏差值小于滞后值的4倍，警告0就会发出。这项检查在实际已知滞后值时进行，该数值取决于自动滞后设置。
- 如果在自动调谐过程的头2.5个循环中收集到的两个高峰误差值之差超过滞后值的8倍，警告1就会发出。
- 如果测量的平均峰值误差小于滞后值的4倍，警告2就会发出。

除了这些警告状况外，还可能有一些错误状况。下表列出了这些错误状况，以及对每项错误之原因的描述。

#### 调谐执行中的错误状况

结果代码（在ARES内）	条件
01	由用户中止 EN位在调谐进行中被清除
02	由于过零监视超时而中止 半循环持续时间超过过零监视间隔
03	由于进程超出范围而中止 PV超出范围：-在自动滞后序列过程中，或者 -在第四次过零前两次，或者 -在第四次过零后
04	由于滞后值超出最大值而中止 用户指定的滞后值或自动确定的滞后值 > 最大值
05	由于非法配置值而中止 下列范围检查错误：-初始环路输出值 < 0.0 或 > 1.0-用户指定的偏差值 ≤ 滞后值或 > 最大值 -初始输出步长 ≤ 0.0 或 > 最大偏差 -过零监视间隔时间 < 最小值-环路表中的抽样时间值为负数
06	由于数字错误而中止 遇到非法的浮点数或除数为零
07	执行PID指令时无功率流（手动模式） 当自动调谐在进行中或被请求时PID指令以无功率流的方式执行
08	自动调谐只可用于P、PI、PD或PID等环路 环路类型不是P、PI、PD或PID

**关于PV超出范围的注释（结果代码3）**

进程变量只要大于0.0且小于1.0，就会被自动调谐器视为在范围内。

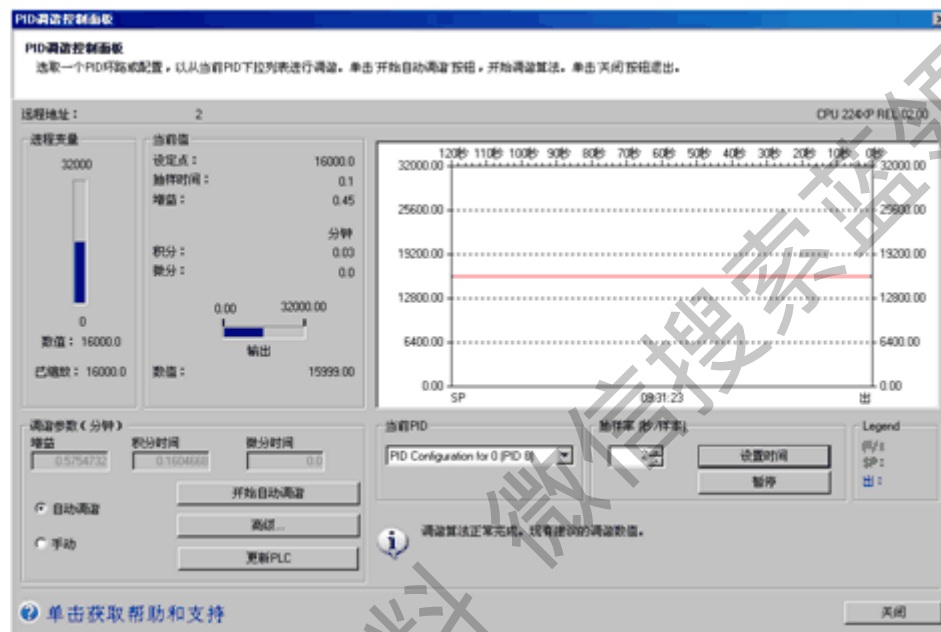
如果在自动滞后序列中发现PV超出范围，调谐将立即中止，并且会发出进程超出范围错误。

如果在调谐序列的起始点和第四个过零之间发现PV超出范围，输出步长值将被减半，调谐序列将从头开始。如果在重新启动后于第一个过零后检测到第二个PV超出范围事件，调谐就会中止，并且发出进程超出范围错误结果。

在第四个过零之后发生的任何PV超出范围事件会导致调谐的立即中止，并且会给出进程超出范围错误结果。

**PID调谐控制面板****STEP 7-**

Micro/WIN包括一个PID调谐控制面板，令您可用图形方式监视PID环路的行为。此外，这个控制面板也令您可发起自动调谐序列



欲使用此控制面板，您必须与一个S7-200

PLC通讯，并且此PLC内必须存在向导生成的PID环路配置。此PLC必须处于‘运行’模式，以允许控制面板显示某PID环路的操作

控制面板在此屏幕左侧顶部显示目标PLC的站地址（远程地址）。此屏幕右侧顶部显示了PLC类型和版本号。在远程地址域下面，有进程变量的条形图表示及已缩放数值和未缩放数值。PV条形图右侧紧接着当前值区域。

当前值区域显示了设定点、抽样时间、增益、积分时间、微分时间的数值。输出值用一水平条形图及其数值显示。当前值区域右边是一图形显示。

此图形显示用不同颜色绘出PV、SP及输出作为时间函数的曲线。PV和SP共用此图左侧的垂直尺度，输出的垂直尺度在此图右侧。

此屏幕底部左侧是“调谐参数（分钟）”区域。增益、积分时间、微分时间诸值会显示在此区域内。圆点按钮会表明所显示的是增益、积分时间、微分时间的“当前”、“建议”或“手动”数值。您可以单击此圆点按钮，显示三个来源之一的这些数值。如要修改调谐参数，请单击手动圆点按钮。

您可以使用“更新PLC”按钮，就正在监视的PID环路，将所显示的增益、积分时间、微分时间诸值传输给PLC。您可以使用“开

图形显示的正下方是“当前PID”选择区域，具有一个拉下菜单，供您选取想要用此控制面板监视的PID环路。

在“抽样速率”区域，您可以选取图形显示的抽样速率，其范围为每个样本1至480秒。您可以编辑抽样速率，然后使用“设置时间”来套用此更改。此图的时间尺度会得到自动调整，从而最佳显示新速率的数据。

您可以通过按“暂停”按钮冻结此图。按“继续”即可以选定的速率继续进行数据抽样。如要清除此图，请在图中用鼠标右键单击，然后选取“清除”。

图选项区域的右边是图例，指明PV、SP、输出等绘图所用的颜色。

当前PID选择区域正下方用于显示正在执行操作的相关信息。



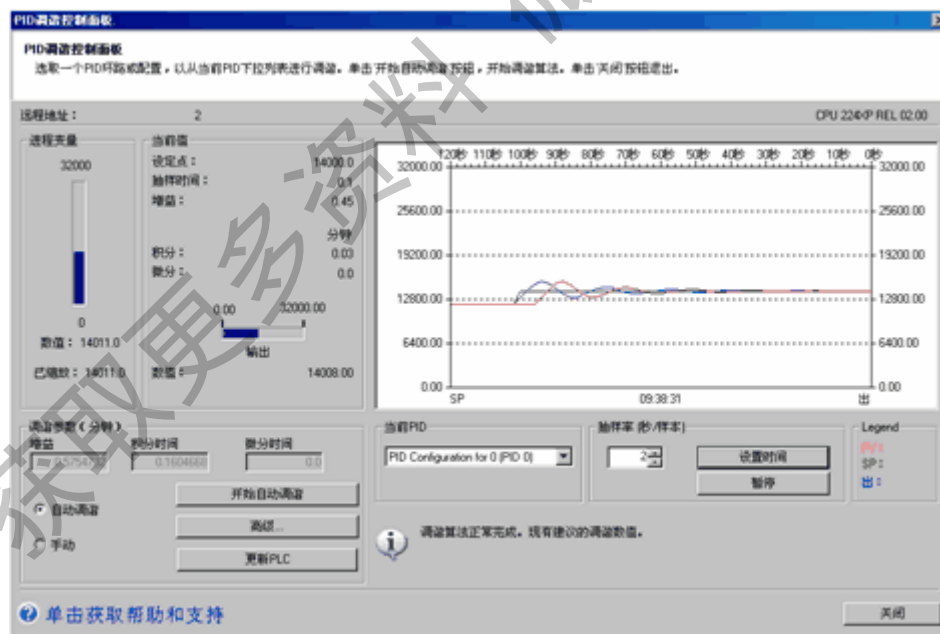
“调谐参数”区域内的“高级...”按钮令您可进一步配置自动调谐过程的参数。上面所示为高级屏幕图象。

由高级屏幕，您可以复选让自动调谐器自动确定滞后值和偏差值（默认设置）的方框；或在这些域输入特定数值，从而最小化自动调谐过程对进程的扰动。

在“其他选项”区域，您可以指定初始输出步长和过零监视超时期限。

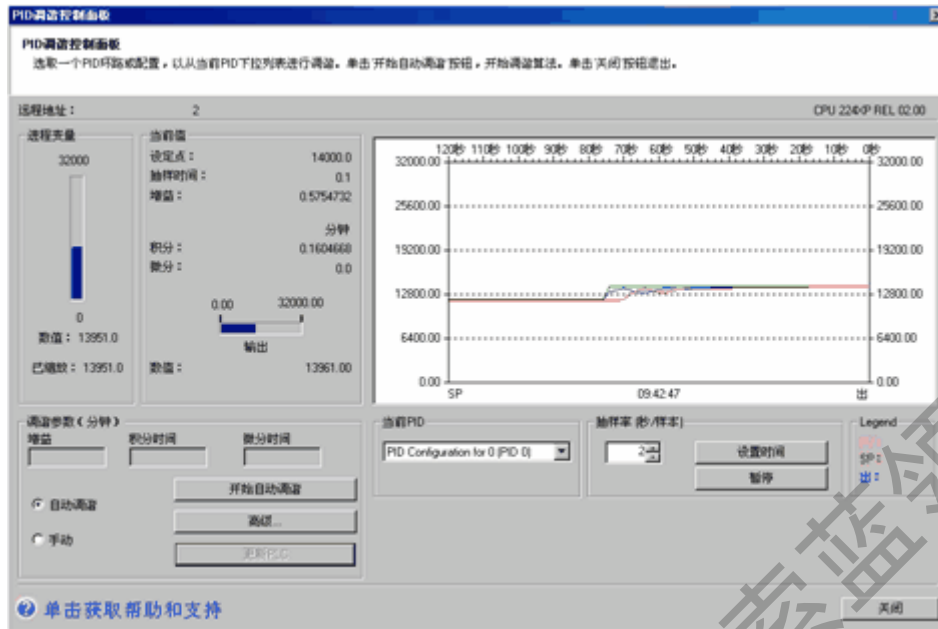
在“动态应答选项”区域，请针对您的进程，单击对应于您希望的环路应答类型的圆点按钮。“动态应答”指环路对进程变化的应答速度，而不是指进程的速度。取决于您的进程，快速应答可能过调且将对应于某欠阻尼调谐状态。中速应答可能濒临过调且将对应于某临界阻尼调谐状态。慢速应答可能没有任何过调且将对应于某过阻尼调谐状态。很慢应答可能没有过调且将对应于某严重过阻尼调谐状态。

一旦做出欲用的选择，请单击‘确定’返回至PID调谐控制面板的主屏幕。



在完成自动调谐序列且已将建议的调谐参数传输至PLC后，您可以使用控制面板监视环路对设定点步进变更的应答。上面的屏幕截图显示了对设定点（范围22000至14000）的应答。

请注意此进程使用原始调谐参数的过调和阻尼的长程振荡行为。



上面的屏幕图象，显示了在套用选取快速应答后由自动调谐过程确定的数值时，环路对设定点变更（由12000至14000）的应答。请注意就此进程而言已经没有过调，但仍有些微振荡。如欲以应答速度为代价消除此振荡，您需要选取中速或慢速应答，然后重新运行自动调谐过程。

一旦针对有关环路的良好起始点后，您可使用控制面板微调有关参数，并监视环路对设定点变更的应答。以此方式，您可以精调有关进程，以在您的应用中获得最优应答。

#### 另请参阅：

[PID算法](#)  
[PID环路表](#)

#### PID向导

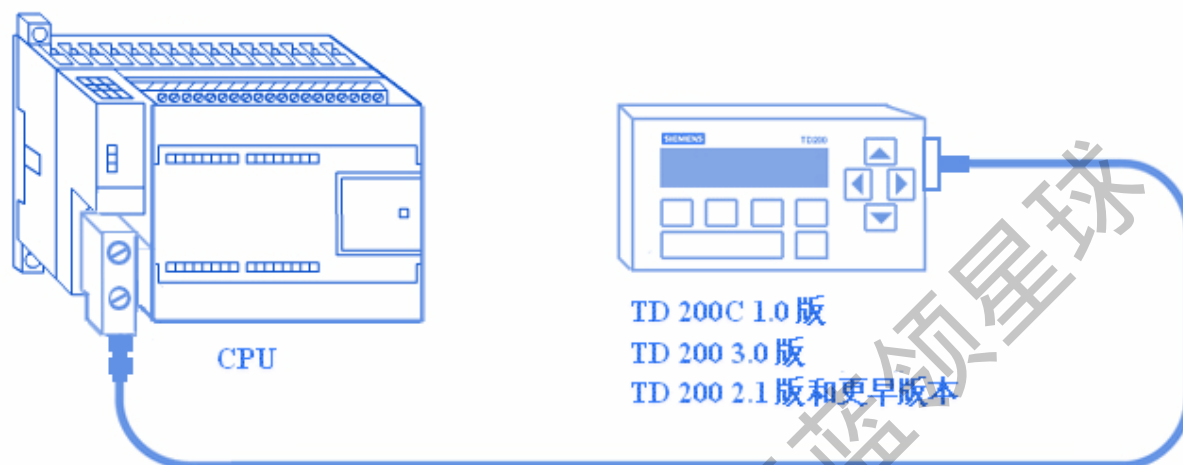
[PIDx\\_INIT \(PID向导子例行程序\)](#)

#### 注释：

具有PID向导配置的项目不直接使用标准PID指令。如果您使用PID向导配置，您的程序必须使用CALL PIDxINIT来激活PID向导子例行程序。

[PID \(标准PID指令\)](#)

## 2.5.5 TD 200 文字显示功能



请使用TD 200向导配置下列TD 200版本

- TD 200 v2.1或之前版本具有9个固定功能按钮
- TD 200 v3.0支持多个语言集、自定义用户菜单、用字符串显示的PLC数据，并且带有9个固定功能按钮（与TD 200第1.0发行版的项目向后兼容）
- TD 200C v1.0不仅具有TD 200 v3.0的全部功能，还有可配置的小键盘面板布局（使用TD小键盘设计员工具）、更多的诊断功能（载入内存盒、‘停止至运行’命令、编辑PLC内存），以及多达20个可配置功能按钮（与TD 200第1.0发行版的项目向后兼容）

TD 200向导会引导您逐步完成TD 200的配置，协助您设置TD 200功能的参数，以及输入用户讯息。如果您需要特定TD 200信息以完成配置，请参阅《SIMATIC文字显示（TD）用户手册》（SIMATIC Text Display (TD) User Manual）。在向导配置完成时，您在向导中指定的信息和地址将用于生成数据页、符号表及指令子例行程序。通过调用TD 200向导和为新的配置指定新的起始地址，您可以创建多个TD 200配置。

另请参阅：

[TD 200 v2.1或之前版本的TD 200向导步骤](#)  
[TD 200 v3.0或TD 200C v1.0的TD 200向导步骤](#)

## 2.5.6 TD 200向导 (TD 200 v2.1版或之前版本)



通过选取菜单命令 **工具(T) > TD 200向导? T** 或单击浏览条上的TD 200向导图表，启动TD 200向导单击“下一步”按钮前往TD 200向导的每个新页。

在完成配置时，单击“完成”按钮。

在向导配置完成时，一个数据页会得到创建，其中包含您在TD 200向导中指定的信息和地址。此信息存储于V内存中，并且每个位置均有注解。通过调用TD 200向导和为新的配置指定新的起始地址，您可以创建多个TD 200配置。

注释：

- 使用STEP 7-Micro/WIN V3.2或更高版本下载的TD 200配置可在上载到新项目中后加以编辑。
- 如果想要在完成后检视已配置的参数块和讯息，您可以打开数据块编辑器，然后选择查看适当的数据页。

### 步骤

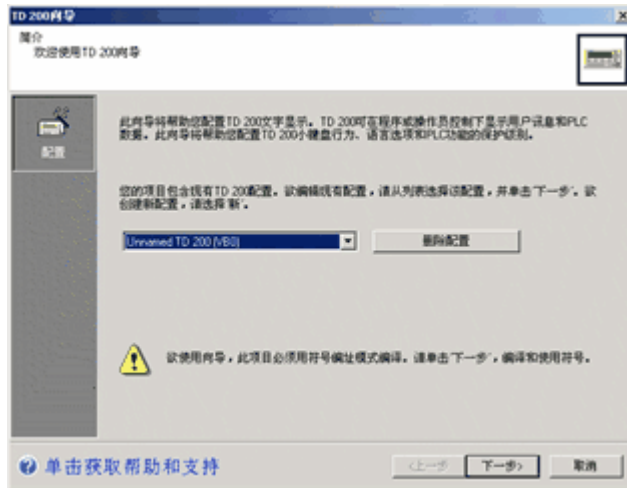
- 1 [编辑现有配置](#)
- 2 [选择TD型号和版本](#)
- 3 [本地化显示](#)
- 4 [标准菜单（每日时间、强制、密码）](#)
- 5 [功能键和更新速率](#)
- 6 [警报选项](#)
- 7 [分配内存](#)
- 8 [警报（输入讯息）](#)

## 9 生成项目元件

### 第1步 编辑现有配置 (TD 200 v2.1或之前版本)

[返回顶端](#)

[TD 200 v3.0或TD 200C v1.0的向导步骤](#)

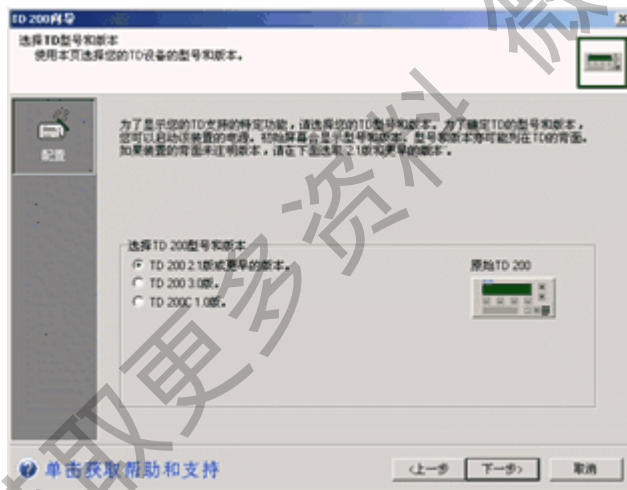


选取您想要编辑的配置。如果您的项目不存在现有配置，那么单击“下一步”前往第2步。

### 第2步 选择TD 200型号和版本 (TD 200 v2.1或之前版本)

[返回顶端](#)

[TD 200 v3.0或TD 200C v1.0的向导步骤](#)

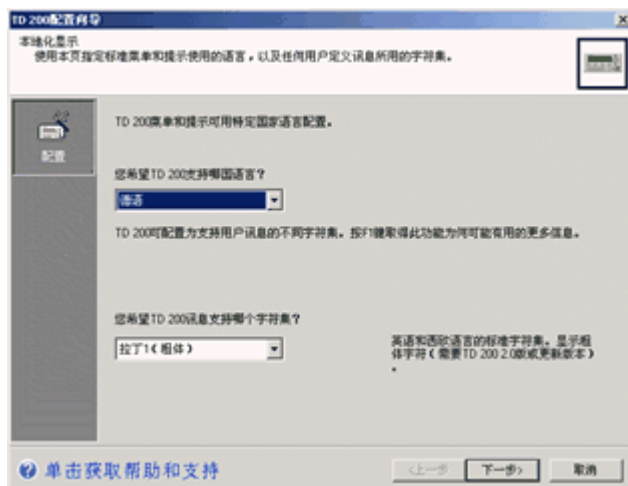


不同的TD 200型号和版本支持不同的功能。为了正确配置TD 200，您必须选取所用TD 200的型号和版本。TD 200 v3.0和TD 200C v1.0使用与TD 200 v2.1不同的向导步骤。

### 第3步 本地化显示

[返回顶端](#)





**国家语言：** 此设定会影响TD 200的预配置菜单和提示。请选取您希望TD 200用于预配置菜单和提示的语言。

**字符集：** 此设定决定了您在创建用户讯息时可以使用的字符类型。

- 每行讯息最多可显示20个单字节字符。

**注释：** 当您在TD 200向导中键入或由STEP 7-Micro/WIN的数据块编辑器检查讯息时，Latin 1 Bold（拉丁1粗体）字符不会显示为粗体；但这些字符会在TD 200显示单元上正确显示（即以粗体出现）。

- **注释：** 简体中文字符集中有一些拉丁1字符。取决于您的键盘和（或）仿真器，拉丁1字符既可能为单字节格式，也可能为双字节格式。例如：荒美拥丿纸讠址 蚩崙瘰

讯息编辑器会显示10个双字节字符位置。每个字符位置可容纳下列组合之一：

- \* 一个双字节字符
- \* 两个单字节字符
- \* 一个单字节字符和一个单字节空格

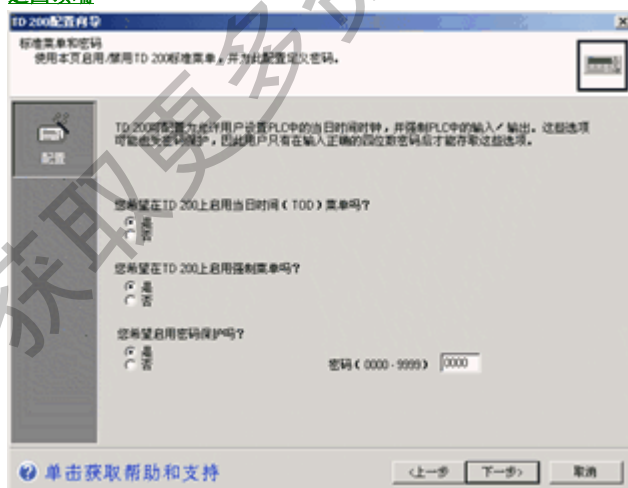
一个字符位置中不能容纳一个单字节字符加一个双字节字符。



**提示：** 如果您在使用2.0版之前的TD 200版本，请勿使用默认字符集（拉丁1）。请只使用原版TD 200字符集或条形图字符集。倘若选择了所用TD 200版本不支持的字符集，在下次使用新配置启动TD 200时，您将收到一个关于参数块的非具体错误讯息。如要纠正此错误，请更改字符集选择，然后将修正后的配置下载到TD 200上。如果您不知道所用TD 200的版本，请将它关闭后重新启动。版本号会在接通电源时显示。

#### 第4步 标准菜单（每日时间、强制、密码）

[返回顶端](#)

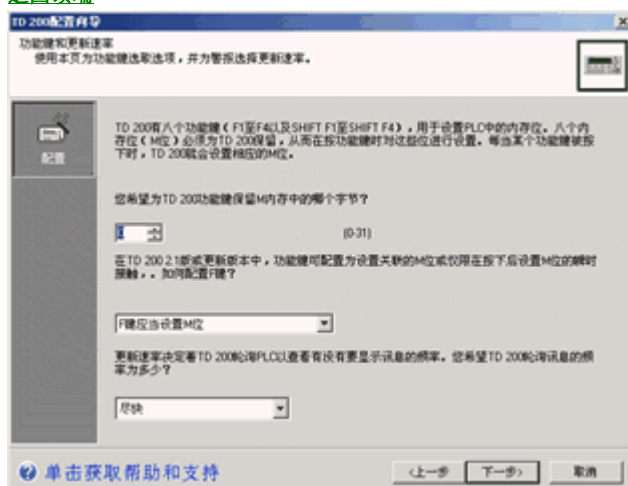


“每日时间（TOD）”和“强制”菜单选择令您可选择是否启用“TOD时钟”菜单和“强制”菜单。一旦启用某个选择，用户可在TD 200中使用相应菜单。若某个选择未启用，该菜单不会在TD 200的菜单模式中出现。

“密码保护”令您可启用一个四位数字密码（从0000到9999）。此密码控制着用户编辑嵌入讯息中的变量和使用菜单模式的能力。如果启用密码保护，向导中会出现一个域，供设置密码之用。此密码不是CPU密码。此密码存储于TD 200内，只影响对该TD 200内编辑功能的使用。

**提示:**

嵌入数据在向向导中后面出现的一个屏幕上加以配置，但此处的密码保护选项会影响到您在后面能否选择“数据的用户编辑”。

**第5步 功能键和更新速率**
[返回顶端](#)


TD 200具有8个用于设置内存位或作为瞬时接触的功能键（F1至F4和SHIFT F1至SHIFT F4）。为了读取TD 200功能键，内存的八个位（M位）必须为TD 200保留。每当某个功能键被按下时，TD 200就会设置相应的M位。

- 请为TD 200输入一个M内存字节地址。
- 选取TD 200的F键行为。这些F键能够将M位设置为如同锁存接触或瞬时接触那样操作。

**警告** 每当某个功能键被按下时，TD

200就会设置相应的M位。如果您不欲使用功能键，并且因此没有指定M字节地址，TD 200为这些功能键指定的默认值为字节M0。如果您的程序使用M0中的位，并且某用户按下任何功能键，TD 200就会设置M0中的相应位，从而重写已由程序赋予该位的数值。对M位的无意变更可能令您的程序有意外举动。不可预知的控制器操作可能造成人员死亡或严重伤害，以及设备损坏。即使您的程序不使用功能键，也请总是保留一个M区地址。

更新速率决定着TD 200轮询PLC以查看有没有要显示信息的频率。

- 请由列表框中选取一个更新速率。

**第6步 警报选项**
[返回顶端](#)


TD 200允许两种讯息大小。请选取欲用的大小：

- 20字符讯息模式 — 每次显示两条讯息
- 40字符讯息模式 — 每次显示一条讯息

TD 200允许您配置多达80条讯息。

- 请输入您想要创建的讯息数目。

### 第7步 分配内存

[返回顶端](#)

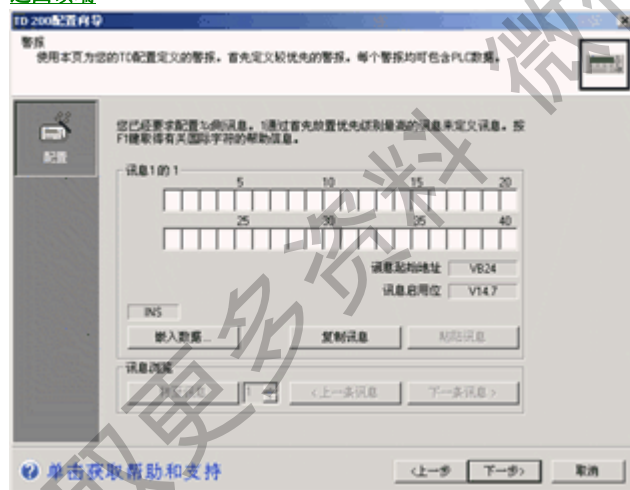


存储参数定义的位置必须得到指定，正常位置为V0。

- 请输入参数块的起始字节地址（最多可包含14个字节）。
- 请输入讯息启用标志的起始字节地址。
- 请输入讯息内容的起始字节地址。

### 第8步 警报（输入讯息）

[返回顶端](#)



在20字符域中输入您的讯息。

**注释：** 原版TD

200字符集和条形图字符集必须使用特别ALT键组合键入讯息字符。所有其他字符集可用普通键盘和（或）仿真器键入。

您能够复制当前可见的讯息并粘贴到下一条讯息。

在“讯息浏览”页中，您可以用下列两种方式之一转到另一条讯息：

- 单击“< 上一个讯息”和“下一个讯息 >”按钮，或者
- 从“转到讯息”框中选取适当讯息，然后单击“转到讯息”按钮，即可直接提取任何讯息。

#### 原版TD 200字符集的ALT键组合

TD 200支持对特定国际字符和特殊字符的显示，这些字符是在TD 200向导中使用ALT键组合加以输入。如要使用ALT键组合：

1. 确定键盘上的NUM LOCK已启用。
2. 按住ALT键。
3. 使用键盘上的数字小键盘，键入该字符的4位数码。
4. 放开ALT键。

请注意，您在TD 200向导中输入的某些字符，在数据块编辑器中有不同的表示。但是，这些字符在TD 200中的显示将与您在TD 200向导中的输入一样。

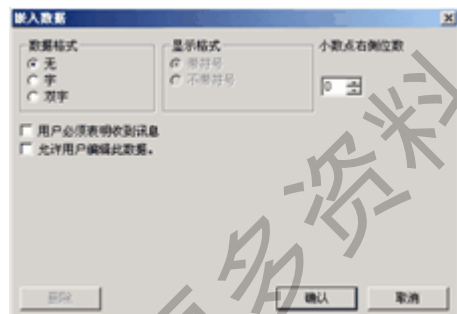
°	ALT + 0248	m	ALT + 0230	e	ALT +
0238					
å	ALT + 0134	ñ	ALT + 0164	b	ALT +
0225					
ä	ALT + 0132	ö	ALT + 0148	S	ALT +
0159					
æ	ALT + 0145	a	ALT + 0224	ü	ALT +
0129					
À	ALT + 0143	s	ALT + 0229		ALT +
0157					
Æ	ALT + 0146	W	ALT + 0234	Þ	ALT +
0195					
ç	ALT + 0155	P	ALT + 0227	Û	ALT +
0180					

### 条形图字符集的ALT键组合

如要在TD

200向导中输入条形图字符，您必须使用下列ALT键组合。请注意，这些字符在此向导（及数据块编辑器）中的表示，与在TD 200本身中不同，但会在TD 200中正确显示。

□	ALT + 0200
□□	ALT + 0201
□□□	ALT + 0202
□□□□	ALT + 0203
□□□□□	ALT + 0204
↑	ALT + 0194



### 嵌入数据

- 将光标放在内的正确位置，然后单击“嵌入数据”按钮。
- 选择数据格式和显示格式。
- 输入小数点右边要显示的位数。

当心：嵌入数据值右端的字符具有该数值在TD 200显示中的锚固点的作用。在TD

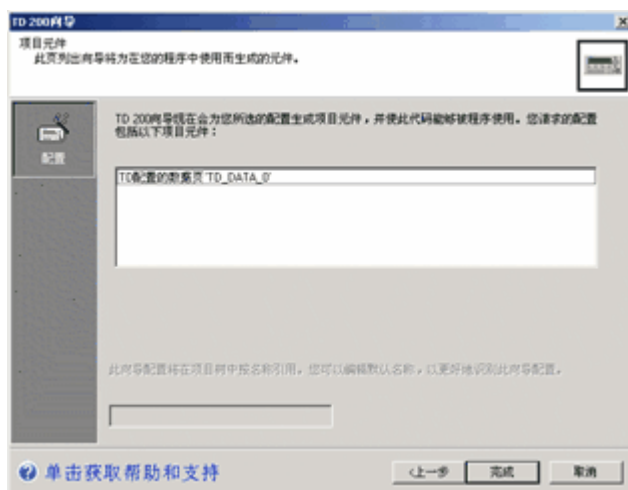
200显示上的讯息中，数据值始终以该锚固点为基准向右对齐。随着数据值的加大，它将在锚固点的左边使用更多的空间

- 选取用户是否必须确认收到该讯息。
- 选择是否允许用户编辑此数据。（如果您在第3步中未启用“密码保护”，则可使用“上一步”按钮返回至该屏幕，然后更改您的选择。使用“上一步”按钮不会丢失在之前的屏幕上已做出的选择。）
- 如果允许编辑，请选择“数据的用户编辑”是否应受密码保护。此外，请选择当讯息被启用后，该数据是否应变成可

嗦 尊

### 第9步 生成项目元件

[返回顶端](#)



TD 200向导会为您选定的配置（程序块和数据块）生成项目元件，并令此代码可供您的程序使用。您必须下载TD 200配置块（数据块）、系统块及程序块至S7-200 CPU，才能更新此配置。

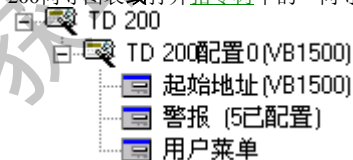
另请参阅：

[TD 200 SIMATIC文字显示功能](#)

## 2.5.7 TD 200向导 (TD 200 v3.0版或TD 200C v1.0版)



通过选取菜单命令 **工具(T) > TD 200向导? (W)** 或单击浏览条上的TD 200向导图标或打开 **指令树** 中的“向导”文件夹并随后打开此向导或某现有配置，启动TD 200向导。



单击“下一步”按钮前往TD 200向导的每个新页。

在完成配置时，单击“完成”按钮。

在向导配置完成时，一个数据页会在您的项目中得到创建，其中包含您在TD 200向导中指定的信息和地址。供TD 200使用的参数和讯息信息存储于V内存中。

在向导配置完成时，一个数据页会在您的项目中得到创建，其中包含您在TD 200向导中指定的信息和地址。供TD 200使用的参数和讯息信息存储于V内存中。

通过调用TD 200向导和为新的配置指定新的起始地址，您可以创建多个TD 200配置。

**注释：**

- 使用STEP 7-Micro/WIN V3.2或更高版本下载的TD 200和TD 200C配置可在上载到新项目中后加以编辑。
- 如果想要在完成后检视已配置的参数块和讯息，您可以通过打开数据块编辑器，然后选取适当的数据页加以检查。
- 出于记录或调试目的，您也可打印向导配置。此打印输出将包括您在此向导中所做选择的记录，以及所创建用户讯息母本。请在STEP 7-Micro/WIN打印向导配置，请在[打印对话框](#)中复选向导配置复选框。

步骤	基本配置
1	<a href="#">编辑现有配置</a>
2	<a href="#">选择TD 200型号和版本</a>
3	<a href="#">启用标准菜单、密码及设置更新速率</a>
4	<a href="#">本地化显示</a>
5	<a href="#">配置小键盘按钮</a>
6	<a href="#">创建用户菜单和文字</a>
7	<a href="#">定义用户菜单和文字</a>
8	<a href="#">创建进程警报与文字</a>
9	<a href="#">定义警报选项与文字</a>
	指定语言集
8	<a href="#">选择要编辑、增加或删除的语言集</a>
9	<a href="#">本地化显示 (指定TD 200C语言集)</a>
	完成最后步骤
10	<a href="#">向导部分完成</a>
11	<a href="#">为配置分配内存</a>
12	<a href="#">生成项目元件</a>

### 第1步 编辑现有配置 (TD 200 v3.0或TD 200C v1.0)

[返回顶端](#)

[TD 200 v2.1或之前版本的向导步骤](#)

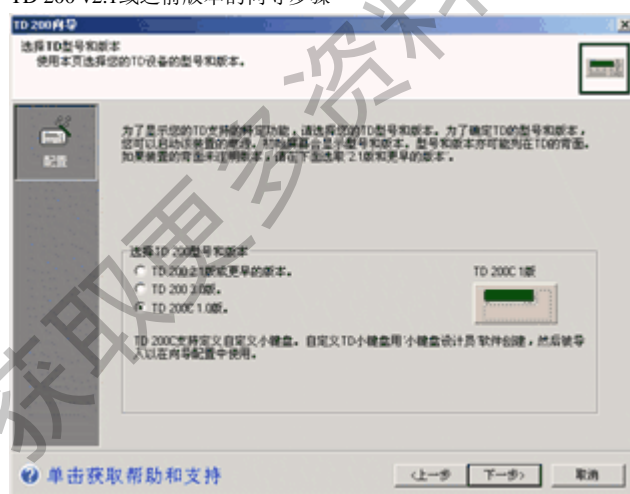
bml TD\_wiz\_CV1V3\_exist\_config.gif]

选取您想要编辑的配置。如果您的项目不存在现有配置，那么单击“下一步”前往第2步。

### 第2步 选择TD 200型号和版本 (TD 200 v3.0或TD 200C v1.0)

[返回顶端](#)

[TD 200 v2.1或之前版本的向导步骤](#)



不同的TD 200型号和版本支持不同的功能。为了正确配置TD 200，您必须选取所用TD 200的型号和版本。TD 200 v2.1使用与TD 200 v3.0和TD 200C 1.0不同的向导步骤。您可以通过单击适当按钮直接前往现有配置的各个部分。



[基本配置](#) 第1-5步

此配置部分令您可设定TD设备的参数，包括指定小键盘行为、语言选项，以及对PLC功能的保护程度。这属于必需部分。



用户菜单

[菜单](#) [选用的第6步](#) 这个部分令您可为此TD

200配置定义一个用户菜单。此用户菜单可用于提供指示性文字讯息，或协助机器的故障排除。配置用户菜单并非必需。



警报

[警报](#) [选用的第7步](#) 这个部分令您可为此TD

200配置定义位触发警报。警报是具有优先顺序的用户讯息，会在程序控制下显示。对警报的配置并非必需。



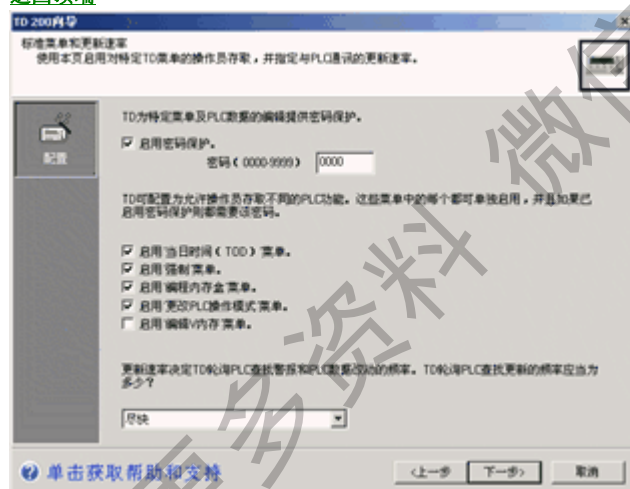
语言集

[语言](#) [选用的第7-8步](#)

如果您定义了警报或用户菜单，“语言集”图标就会出现；这就让您可以定义用户讯息的不同语言翻译。

### 第3步 启用标准菜单、密码及设置更新速率

[返回顶端](#)



此页令用户可以指定哪些TD

200菜单选项应为此应用程序出现。这些菜单可为方便用户或出于安全目的而启用或禁用。一旦在向导中启用了标准菜单选择，

贝搬 留察蛟涌惺保 没BI 档裁款D

200中使用该菜单。倘若某个菜单选择已在向导中被禁用，那么当此应用程序运行时，该菜单将无法在TD 200中使用。

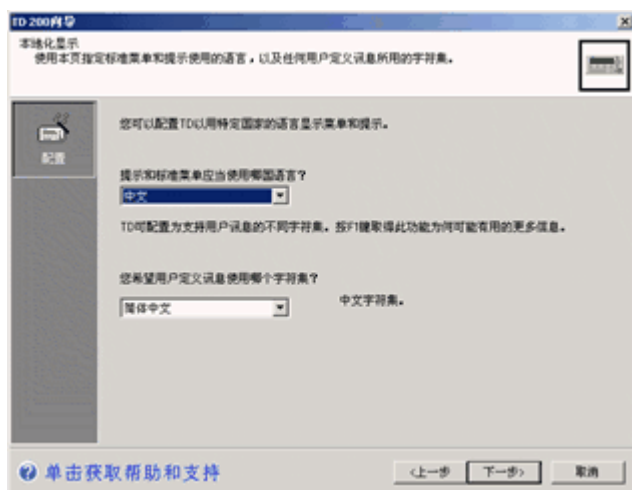
“密码保护”令您可启用一个四位数字密码（从0000到9999）。此密码控制着用户编辑嵌入讯息中的变量和使用菜单模式的能力。

如果启用密码保护，向导中会出现一个域，供设置密码之用。此密码不是CPU密码。此密码存储于TD 200内，只影响对该TD

200内编辑功能的使用。

### 第4步 本地化显示

[返回顶端](#)



请选择TD应为标准提示和菜单使用的国家语言。

请选择您创建的用户消息应使用的字符集。请注意，选取某字符集并非一定意指特定国家语言。但是，许多国家语言需要特定字符集，方能显示该语言中使用的所有字符。

若是定义了多个语言集（第8步）的配置，则本地化显示页会针对每个语言集出现。每语言集可指定一个不同的国家语言和（或）地址。

**国家语言：**此设定会影响TD 200的预配置菜单和提示。请选择您希望TD 200显示预配置菜单和提示的语言。

**字符集：**此设定决定了您在创建用户消息时可以使用的字符类型。

- 每行消息最多可显示20个单字节字符。

**注释：**当您在TD 200向导中键入或由STEP 7-Micro/WIN的数据块编辑器检查消息时，Latin 1 Bold（拉丁1粗体）字符不会显示为粗体；但这些字符会在TD 200显示单元上正确显示（即以粗体出现）。

- 简体中文字符为双字节字符。每行消息最多可显示10个双字节字符。

**注释：**简体中文字符集中有一些拉丁1字符。取决于您的键盘和（或）仿真器，拉丁1字符既可能为单字节格式，也可能为如同简体中文字符的双字节字符。如果您在同一条消息中将单字节字符与双字节字符混用，请切记单字节字符的后面只能接单字节字符或空格。

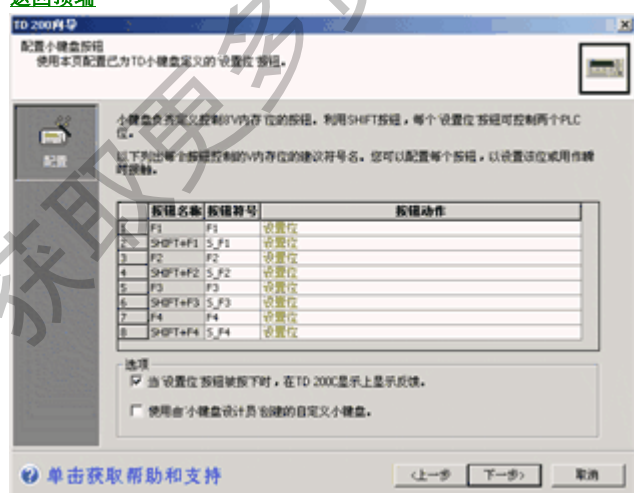
消息编辑器会显示10个双字节字符位置。每个字符位置可容纳下列组合之一：

- \* 一个双字节字符
- \* 两个单字节字符
- \* 一个单字节字符和一个单字节空格

一个字符位置中不能容纳一个单字节字符加一个双字节字符。

## 第5步 配置小键盘按钮

[返回顶端](#)



TD 200C和TD 200

V3的标准面板提供了四个“设置位”按钮和一个Shift按钮，可用于控制PLC内的总共8个V内存位。如果您指定自定义的面板设计，则可有最多20个按钮配置为“设置位”功能。这些按钮控制的位会在V内存中TD 200参数块的一个保留区内得到分配。

由向导的这个页面，您可以为小键盘按钮控制的每个位指定一个符号名。此向导将基于每个按钮的名称建议一个符号名。您可以



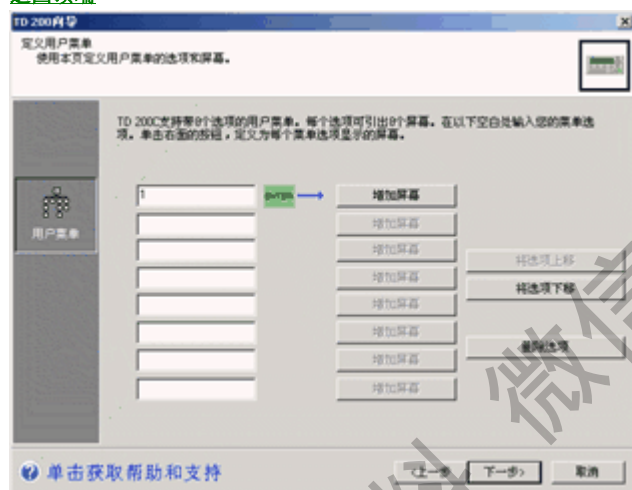
接受这些建议，或单击每个符号名进行编辑。在完成TD 200向导后，这些符号将会在向导创建的符号表中出现。这些符号自此可用于您的程序逻辑，以检测用户对TD 200小键盘按钮的按动。

每个按钮都可配置为设置相应PLC位或作为一个瞬时接触。一旦选取‘设置位’，每当用户按下TD小键盘上的按钮时，相应的V内存位就会得到设置。该位会保持在已设置状态，只能使用程序逻辑加以清除。倘若某按钮被选取为‘瞬时接触’，每当用户按下TD小键盘的该按钮时，相应的V内存位就会得到设置。该位只在此TD面板按钮按下期间处于已设置状态。当用户放开该按钮时，内存位就会被清除。

如果您已选取TD 200C作为目标型号，页面上将显示‘选择小键盘...’按钮。此按钮用于选取使用TD小键盘设计员应用程序定义的自定义TD 200C面板。按下‘选择小键盘...’按钮即可载入一Windows‘打开文件’对话框，令您可选取一个小键盘设计员项目文件。在选定面板文件后，此页面将会更新并列出在该面板文件中已定义的按钮。如需关于使用TD小键盘设计员工具创建自定义用户面板的详细说明，请参阅《SIMATIC文字显示 (TD) 用户手册》(SIMATIC Text Display (TD) User Manual)。

## 第6步 定义用户菜单和文字

[返回顶端](#)





TD 200和TD 200C支持一个将在运行时间由用户控制之下显示的用户菜单。用户能够使用TD 200小键盘上的标准按钮滚动此菜单，然后选取要显示的标题。一旦选取某菜单选项，用户能够使用箭头按钮在该菜单选择的屏恢 涛韶 F聊泰砂 怀 ?0个字符的文字，亦可能包括可编辑的PLC变量和图标。

您可以配置1-8个用户菜单选项，其中每个菜单选项可有1-8个屏幕。此用户菜单的显示由TD用户而不是PLC程序加以控制。如要创建菜单选项，请您希望使用的每个选择输入文字；然后单击‘编辑屏幕’按钮，定义该菜单选项的文字屏幕。



请针对此用户菜单的选择，编辑要显示的信息屏幕（TD 200为40个字符）。请使用屏幕编辑器创建用户讯息。您可用下列方式的组合编写您的讯息：

- 单击工具条  上的某个图标按钮，在屏幕内的当前光标位置插入恒壁急转 D 200C)。请注意，此图标不会在TD 200向显示中实际出现。由这个图标占据的位置将用一特殊颜色显示。
- 使用键盘为屏幕输入文字。
- 嵌入一个PLC内存值；方法为单击  按钮，然后指定您想要在该讯息中显示的数据之地址 C 扛釜聊蛔啮啜砂 ?个PLC数据引用。

您可以复选任何屏幕，表明该屏幕应做为默认的‘启动’屏幕。如果您删除了已选定的默认屏幕，那么将没有默认屏幕。在没有其他用户活动时，默认的启动屏幕将会在TD 200上显示。此默认屏幕也可包括PLC数据和图标。

当键入屏幕文字时，您可以按键盘的Insert键在插入模式和覆盖模式之间切换。屏幕编辑器所包括的微型工具条，提供了剪切、复制、粘贴及删除整个屏幕的功能。讯息内容亦可在菜单标题编辑器和警报编辑器之间移动：



将屏幕的一个副本放在剪贴板内，然后删除此屏幕；屏幕的总数就会减少一个。



将当前屏幕的一个副本放在剪贴板上。



当剪贴板上有TD 200讯息时，‘粘贴’按钮就可使用。粘贴讯息不会将该讯息从剪贴板上移除。此粘贴的讯息会被插入到屏幕列表中当前显示钠聊簧厦姘 U程 断11.峡钎聊坏淖茨 皂右桓触灰屹撒没 缙 延凶奋第 浚?) 的屏幕，此操作就无法进行。



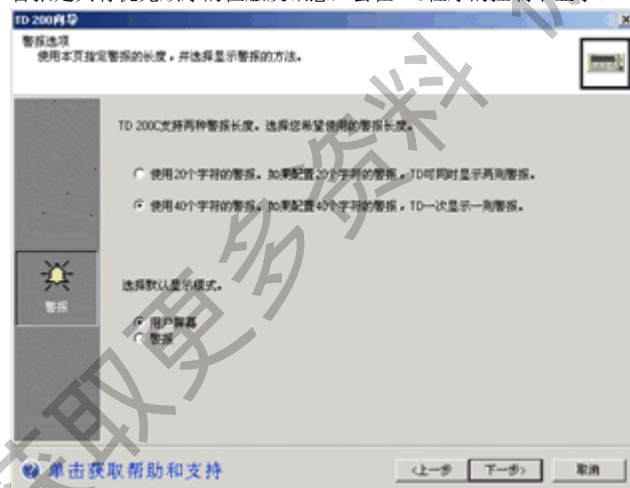
删除此屏幕并返回到菜单或屏幕定义页面，屏幕的总数就会减少一个。

倘若此向导配置定义了多个语言集，那么用户菜单简介页会允许您选取想要编辑的语言集。在这种情况下，您会看到现有语言集的一个下拉列表。当前语言集将为默认选择。请注意，您不能由此创建一个语言集，您只能选择针对某现有语言集编辑此用户菜单。新的语言需在‘语言集’页添加。请用您的主要语言创建菜单和屏幕讯息，然后编辑此菜单和屏幕讯息文字的替代语言版本。您必须使用主要语言增加或删除菜单选项、讯息屏幕及PLC数据引用。

## 第7步 定义警报选项与文字

[返回顶端](#)

警报是具有优先顺序的位触发讯息，会在PLC程序的控制下显示。





请选取所有警报的长度和显示模式。



您可配置多达80条警报。如要创建新警报，请单击‘新警报’按钮，修改识别该警报的默认符号名，然后输入讯息。请使用‘前一个警报’和‘下一个警报’按钮修改现有警报。新警报的创建顺序就是其优先顺序。此优先顺序由‘转到’下拉列表中括号内的警报号表明。

您可用下列方式的组合编写您的警报屏幕：

- 单击工具条  上的某个图标按钮，在屏幕内的当前光标位置插入 **瓜桓壁急轳日幌轶D 200C**。请注意，此图标不会在TD 200向导显示中实际出现。由这个图标占据的位置将用一特殊颜色显示。
- 用键盘输入文字。国际字符表和插入这些字符的ALT键-数字组合请见《SIMATIC文字显示 (TD) 用户手册》的附录A。
- 嵌入一个PLC内存值；方法为单击  按钮，然后指定您想要在该讯息中显示的数据之地址 **C刻蹇 女聊蚰啮噉吸砂 ?个PLC数据引用**。

当键入屏幕文字时，您可以按键盘的Insert键在插入模式和覆盖模式之间切换。

屏幕编辑器所包括的微型工具条，提供了剪切、复制、粘贴及删除整个警报屏幕的功能。讯息内容亦可在菜单标题编辑器和警报编辑器之间移动：



将警报屏幕的一个副本放在剪贴板内，然后删除此警报屏幕；警报的总数就会减少一个。



将当前警报屏幕的一个副本放在剪贴板上。



在执行粘贴操作时，您将有机会选择插入或覆盖当前警报屏幕。粘贴插入动作会插入一条新警报屏幕，其优先顺序比当前显示之警报高一个。警报的总数会增加一个。粘贴覆盖动作将覆盖当前显示的警报，不会创建新警报屏幕。请注意，剪切和粘贴可用于重新排列警报，以改变其优先顺序。如要变更某警报屏幕的优先顺序，请前往该警报屏幕，然后选取‘剪切’图标，就可将该警报屏幕从其当前位置移除并放在剪贴板上。然后前往该警报的新位置（通过视需要单击‘前一个警报’或‘下一个警报’），并且选取‘粘贴’按钮。当讯息框出现，询问您是要插入或覆盖时，请选取插入。该警报屏幕现就在列表中得到移动并改变其优先顺序。



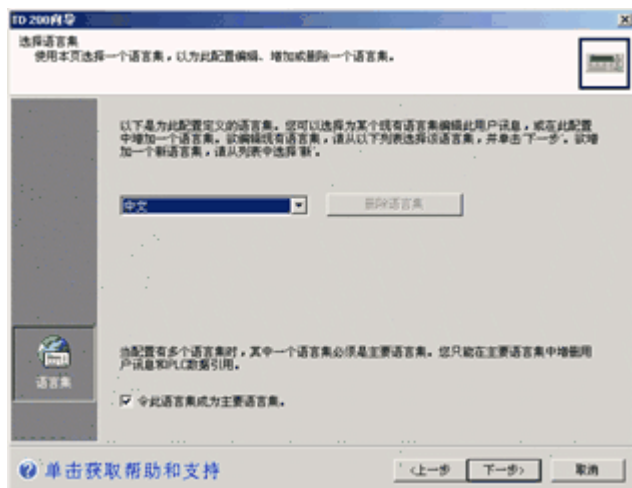
删除此警报屏幕，警报屏幕的总数就会减少一个。

倘若此向导配置定义了多个语言集，那么用户警报简介页面允许您选取想要编辑的语言集。在这种情况下，您会看到现有语言集的一个下拉列表。当前语言集将为默认选择。请注意，您不能由此创建一个语言集，您只能选择针对某现有语言集编辑此用户菜单。新的语言需在‘语言集’页添加。请用您的主要语言创建警报，然后编辑此警报文字的替代语言版本。您必须使用主要语言增加或删除警报及PLC数据引用。

### 第8步 选择要编辑、增加或删除的语言集

#### 返回顶端

如果您定义了用户菜单或位触发警报，“语言集”图标就会出现。此功能可帮助您制作TD 200项目的国际版本。如果您增加某个替代语言并翻译屏幕文字，则用户可用该替代语言显示TD 200的文字。



如要增加新语言，请单击下拉列表并选取‘新语言集’。



然后选取要复制的语言集。您可以将新语言集作为任何现有语言集的副本加以初始化。与此语言集关联的用户讯息就可供翻译。倘若此配置定义了多个语言集，其中一个语言集必须被指定为‘主要’。您只能用主要语言集执行下列操作：

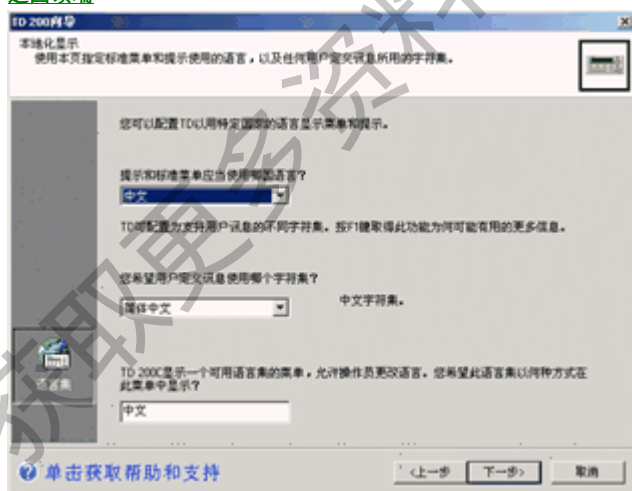
- 更改警报选项和增加或删除警报屏幕。
- 增加或删除用户菜单项。
- 增加或删除用户菜单屏幕。
- 增加、删除或编辑警报或菜单屏幕中的PLC数据引用。

**注释：**

与警报和PLC数据引用关联的符号名适用于所有语言集。举例来说，警报0的符号名对所有语言集都一样。

### 第9步 本地化显示（具有多个语言集）

[返回顶端](#)



**国家语言：**此设定会影响TD 200的预配置菜单和提示。请选取您希望TD 200显示预配置菜单和提示的语言。

**字符集：**此设定决定了您在创建用户讯息时可以使用字符类型。

- 每行讯息最多可显示20个单字节字符。

**注释：**当您在TD 200向导中键入或由STEP 7-Micro/WIN的数据块编辑器检查讯息时，Latin 1 Bold（拉丁1粗体）字符不会显示为粗体；但这些字符会在TD 200显示单元上正确显示（即以粗体出现）。

- 简体中文字符为双字节字符。每行讯息最多可显示10个双字节字符。

**注释：**简体中文字符集中有一些拉丁1字符。取决于您的键盘和（或）仿真器，拉丁1字符既可能为单字节格式，也可能为如同简体中文字符的双字节字符。如果您在同一条讯息中将单字节字符与双字节字符混用，请切记单字节字符的后面只能接单字节字符或空格。

讯息编辑器会显示10个双字节字符位置。每个字符位置可容纳下列组合之一：

- \* 一个双字节字符
- \* 两个单字节字符
- \* 一个单字节字符和一个单字节空格

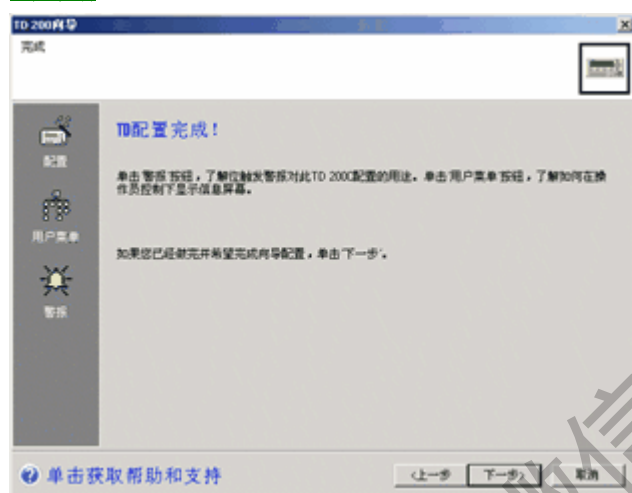
一个字符位置中不能容纳一个单字节字符加一个双字节字符。

语言集名称

任何配置的第一个语言集将得到一个默认名称；此名称与在向导的‘配置’部分指定的TD 200语言相符。您可以通过在‘选择语言集’部分选取此默认名称并编辑以修改之。

## 第10步 向导部分完成

[返回顶端](#)



配置

[基本配置](#) 选用的第1-5步

此配置部分令您可设定TD设备的参数，包括指定键盘行为、语言选项，以及对PLC功能的保护程度。这属于必需部分。



用户菜单

[菜单](#) 选用的第6步

这个部分令您可为此TD配置定义一个用户菜单。用户菜单并非必需。



警报

[警报](#) 选用的第7步 这个部分令您可定义位触发警报。警报并非必需。

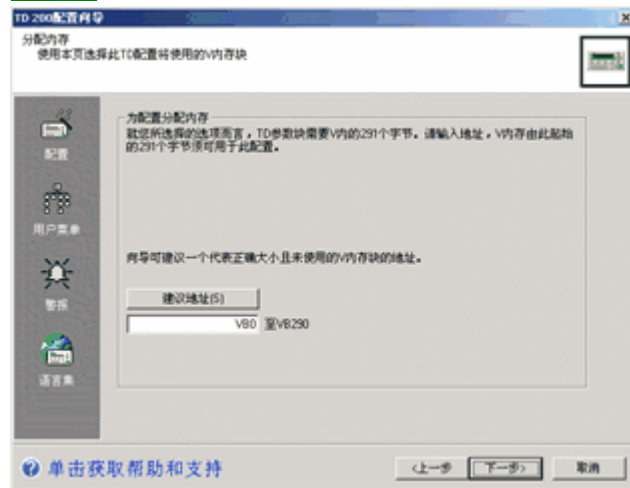


语言集

[语言](#) 选用的第7-8步

如果您定义了警报或用户菜单，“语言集”图标就会出现；这就让您可定义用户讯息的不同语言翻译。

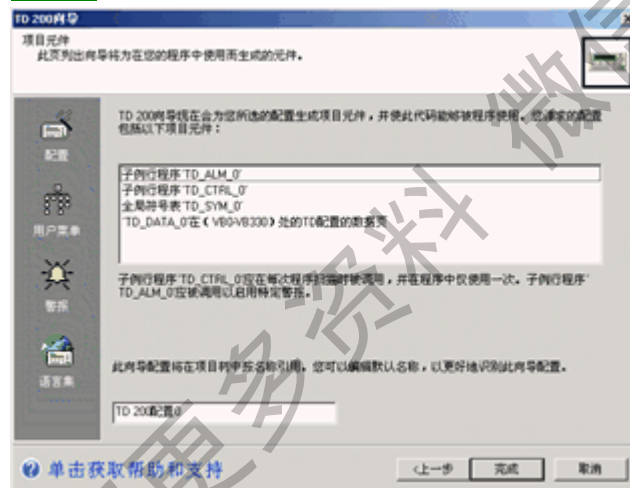
## 第11步 为配置分配内存

[返回顶端](#)

当您将特定配置初次到达此页时，该配置会自动得到建议的 V 内存地址。此建议地址已针对程序的交叉引用得到检查，确定是未怀疑的。

单击“建议地址”按钮，向导就会寻找下一个（具有足够容量的）可用 V 内存块。举例来说，假如您输入地址 VB300，并且此配置的大小是 132 个字节，那么“建议地址”功能将会在 VB432 开始寻找一个 V 内存块。下一个具有足够容量的可用内存块将得到使用。

(3) 摇 痢 1 坊 岙 桓 隆

**第 12 步 生成项目元件**[返回顶端](#)**TD 200 项目元件**

**TD\_CTRL\_x** （控制 TD 200 操作）这个子例行程序用于监视和控制 TD 200 设备的操作，在您的程序中应只调用一次，并且应在每个程序扫描时调用。

**TD\_ALM\_x** （启用警报）如果 TD 配置定义了警报，那么向导将会生成这个子例行程序，用于启用特定的警报。

**TD\_SYM\_x** （全局符号表）包含与 TD 200 按钮、警报及 PLC 数据关联的符号。

**TD\_DATA\_x** （数据块页）用于存储 TD 200 向导配置的 V 内存数据。

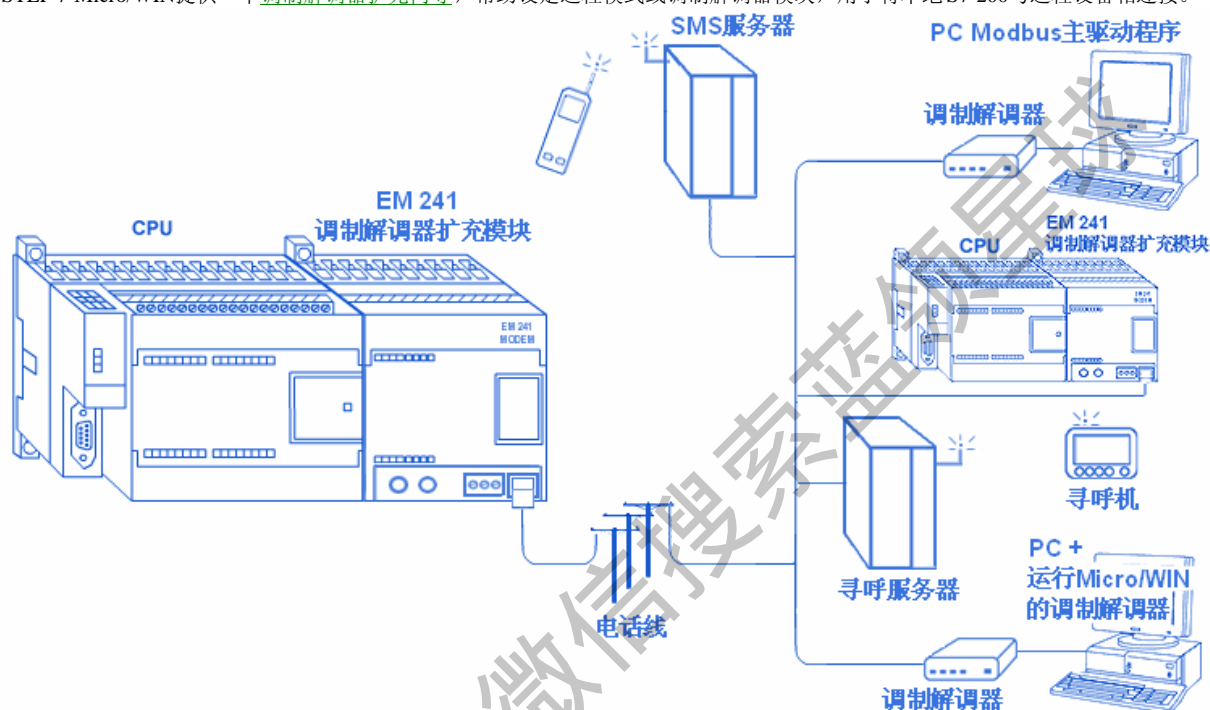
**另请参阅：**

[TD 200 SIMATIC 文字显示功能](#)

## 2.5.8 EM241调制解调器模块性能

EM 241调制解调器模块允许您将S7-200直接与模拟电话线连接，并支持S7-200和STEP 7-Micro/WIN之间的通讯。调制解调器模块还支持Modbus从属RTU协议。调制解调器模块与S7-200之间的通讯系通过扩充I/O总线完成。

STEP 7-Micro/WIN提供一个[调制解调器扩充向导](#)，帮助设定远程模式或调制解调器模块，用于将本地S7-200与远程设备相连接。



调制解调器模块是一个智能扩充模块，用于下表显示的S7-200 CPU。

CPU	说明
CPU 222 1.10版或更高版本	CPU 222 DC/DC/DC CPU 222 AC/DC/中继
CPU 224 1.10版或更高版本	CPU 224 DC/DC/DC CPU 224 AC/DC/中继
CPU 226 1.00版或更高版本	CPU 226 DC/DC/DC CPU 226 AC/DC/中继
CPU 226XM 1.00版或更高版本	CPU 226XM DC/DC/DC CPU 226XM AC/DC/中继

调制解调器模块在前侧面板上有8个状态LED。下表是有关状态LED的说明。

LED	说明
MF	模块故障 — 该LED开启时模块探测到一错误状态，例如： VDC外接电源 • I/O监视定时器超时 • 调制解调器故障 • 无24 与本地CPU通讯错误
MG	模块良好 — 本LED开启时未出现模块错误状态。如果配置表中出现错误或者用户为电话线接口选择了非法国家设置，模块状况LED会闪烁。检查STEP 7-Micro/WIN中的PLC信息屏幕或查阅SMW220中的数值（0号模块槽），了解有关配置错误的信息。
OH	摘机 — 本LED开启时EM 241正在使用电话线。
NT	无拨号音 — 本LED指示出现错误状态，当EM 241接到指令发送讯息但电话线上无拨号音时会开启。只有在EM 241被配置为拨号前检查拨号音时才会显示的错误状态。在拨号尝试之后，LED会保持开启约5秒钟。
RI	铃声指示器 — 本LED指示EM 241正在接受输入呼叫。
CD	载体探测 — 本LED指示已与远程调制解调器连接。
RX	接收数据 — 当调制解调器正在接收数据时，本LED会闪烁。
TX	传输数据 — 当调制解调器正在传输数据时，本LED会闪烁。

调制解调器模块提供以下通讯功能：

- [国际电话线接口](#)
- [用于编程和排除故障的与STEP 7 - Micro/WIN连接的调制解调器接口（通信服务）](#)

- [Modbus RTU协议](#)
- [支持数字和文字寻呼](#)
- [支持SMS讯息传送](#)
- [CPU至CPU或CPU至Modbus数据传送](#)
- [密码保护](#)
- [安全回叫](#)
- [调制解调器模块配置存储在CPU中](#)

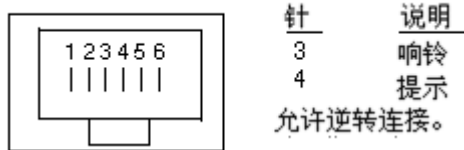
### 国际电话线接口

[返回顶端](#)

调制解调器模块为标准V.34（33.6 kBaud）、10位调制解调器，与大多数内装和外装PC调制解调器兼容。调制解调器模块不与11位调制解调器通讯。您使用安装在模块前侧的六位、四线RJ11接头将EM

241调制解调器模块与电话线连接（见下图）。可能需要一个适配器转换RJ11接头，以便与不同国家的标准电话线终端连接。请

卧哪 褂攻氛逝淦鸬油肺牡抵械南晗感畔 1



调制解调器和电话线接头由24

VDC电源供电，可连接至CPU传感器电源或外部电源。将调制解调器模块上的接地终端连接至系统接地终端。

当调制解调器接通电源时，调制解调器会根据具体国家的操作自动配置电话接口。位于调制解调器前侧的两个旋转开关用于选择国家。您必须在调制解调器模块电源开启之前将开关设为所需国家选择。请参阅 [国家代码](#) 中所支持国家的开关设置。

### STEP 7-Micro/WIN接口

[返回顶端](#)

调制解调器模块允许您通过电话线与STEP 7-Micro/WIN通讯（通信服务）。您无须对S7-200 CPU进行配置或编程，以便在与STEP 7-Micro/WIN配合使用时将EM 241用作远程调制解调器。

将调制解调器模块与STEP 7-Micro/WIN配合使用时，请遵循以下步骤：

1. 拔下S7-200 CPU电源接头，将调制解调器模块与I/O扩充总线连接。当S7-200 CPU电源开启时，请勿连接任何I/O模块。
2. 将电话线与调制解调器模块连接。如有必要，使用适配器。
3. 将24伏特直流电与调制解调器模块接线盒连接。
4. 将调制解调器模块接线盒接地接头与系统接地接头连接。
5. 设置国家开关。
6. 打开S7-200 CPU和调制解调器模块电源。
7. 配置STEP 7-Micro/WIN，使之与一台10位调制解调器通讯。

### Modbus RTU协议

[返回顶端](#)

您可以配置EM 241调制解调器模块，作为Modbus

RTU从属装置应答。调制解调器模块通过调制解调器接口接收Modbus请求、解释此类请求、并将数据在调制解调器模块和CPU之间传输。然后调制解调器模块生成Modbus应答，并将其通过调制解调器接口输出。

**注释：**如果调制解调器模块被配置作为Modbus RTU从属装置应答，STEP 7-Micro/WIN无法通过电话线与调制解调器模块通讯。

调制解调器模块支持以下Modbus功能：

功能	说明
01	读取线圈（输入）状态
02	读取输入状态
03	读取保持寄存器
04	读取输入（模拟输入）寄存器
05	写入单个线圈（输出）
06	预设单个寄存器
15	写入多个线圈（输出）
16	预设多个寄存器

Modbus功能4和16在一次请求中最多可允许读取或写入125个保持寄存器（250字节V内存）。功能5和15写入至CPU的输出图像寄

嫫鳌U庠十 悼梢杂糜没 C 缘蛎切砗



Modbus地址通常写为5或6个字符的数值，包含数据类型和偏移。第一个或第二个字符决定数据类型，最后四个字符在数据类型中选择适当的数值。Modbus主设备将地址映射至正确的Modbus功能。

下表显示调制解调器模块支持的Modbus地址以及Modbus地址与S7-200 CPU地址的映射。

使用STEP 7-Micro/WIN中的调制解调器扩充向导为调制解调器模块建立一个配置块，使之支持 Modbus RTU协议。必须在使用Modbus协议之前将调制解调器模块配置块下载至CPU数据块。

Modbus地址	S7-200 CPU地址
000001 000002 000003 ... 000127 000128	Q0.0 Q0.1 Q0.2 ... Q15.6 Q15.7
010001 010002 010003 ... 010127 010128	I0.0 I0.1 I0.2 ... I15.6 I15.7
030001 030002 030003 ... 030032	AIW0 AIW2 AIW4 ... AIW62
040001 040002 040003 ... 04xxxx	VW0 VW2 VW4 ... VW 2*(xxxx-1)

## 寻呼与SMS讯息传送

### [返回顶端](#)

调制解调器模块支持传送：

- 数字和文字寻呼讯息
- 向手机发送的SMS（简短讯息服务）讯息（受手机供应商支持）。

讯息和电话号码存储在调制解调器模块配置块中，该程序块必须下载至 S7-200

CPU中的数据块中。您可以使用调制解调器扩充向导为调制解调器模块配置块建立讯息和电话号码。调制解调器扩充向导还会建立程序代码，允许您的程序将讯息传送初始化。

### 数字寻呼

数字寻呼使用按键电话的音调向寻呼机发送数字数值。调制解调器模块向请求的寻呼服务拨号，等候语音讯息完成，然后发送与寻呼讯息中的数字对应的音调。寻呼讯息中可使用数字0至9、星号（\*）、A、B、C和D。寻呼机允镜男抢拧 A、B、C和D实际字符并非标准化字符，由寻呼机和寻呼服务供应商决定。

### 文字寻呼

文字寻呼允许将字母数字讯息传输至寻呼服务供应商，再由寻呼服务供应商传输至寻呼机。文字寻呼供应商通常设有一条接受文字寻呼的调制解调器线路。调制解调器模块使用电信定位器字母数字协议（TAP）将文字讯息传输至服务供应商。很多文字寻呼供应商使用该协议接收讯息。

### 简短讯息服务（SMS）

某些手机电话服务支持简短讯息服务（SMS）讯息传送，通常为与GSM兼容的手机电话服务。SMS允许调制解调器模块通过模拟电话线将讯息传送给SMS供应商。SMS供应商然后将讯息还书至手机电话，讯息在电话的文字显酒勉般允尽5髦平雅骸骸?械褂攻縲哦?黄誓帜甘 中 惊 TAP)和通用计算机协议（UCP）将讯息传送给SMS供应

### 文字和SMS讯息中的嵌入变量

调制解调器模块可在文字讯息中嵌入来自CPU的数据数值，并根据讯息中的一条规格为数据数值制定格式。

稍灾付? ∈ 阙竺婧陀顾炳氛 洽皇 约靶 ∈ 阙且恒鲍慷故且恒超汉拧5庇没C缘蛞 髦髦平雅骸骸? 样 湮淖 讯息时，调制解调器模块从CPU检索讯息、确定讯息中所需的CPU数值、从CPU检索这些数值、并在将讯息传输给服务

供应商之前制定格式和将数值放入文字讯息中。讯息传送供应商的电话号码、讯息和嵌入讯息的变量均通过多个CPU扫描周期从CPU读取。您的程序不应当在发送讯息时修改电话号码或讯息。可以在发送讯息的过程中继续更新嵌入讯息中的变量。如果讯息包含多个变量，这些变量通?CPU的多个扫描周期读取。如果您希望一则讯息中嵌入的所有变量保持一致，则在发送讯息后不得更改任何嵌入变量

## 数据传送

### [返回顶端](#)

调制解调器模块允许您的程序将数据通过电话线传输至另一台CPU或Modbus设备。数据传送和电话号码用调制解调器扩充向导配置，并存储在调制解调器模块配置块中。然后配置块被下载至 S7-200

CPU中的数据块中。调制解调器扩充向导还会建立程序代码，允许您的程序将数据传送初始化。

数据传送可以是远程设备读取数据请求、将数据写入远程设备的请求或读取和写入数据的请求。数据传送可以读取和/或写入1至100个字的数据。数据传送将数据输入和输出所连接CPU的V内存。

调制解调器扩充向导允许您建立一个数据传送，该数据传送包含从远程设备的单一读取、至远程设备的单一写入或二者兼备。

数据传送使用调制解调器模块的配置协议。调制解调器模块的配置支持PPI协议（该协议对STEP 7-

Micro/WIN作出应答），调制解调器模块使用PPI协议传输数据。如果调制解调器模块的配置支持Modbus

RTU协议，则使用Modbus协议传输数据传送。

远程设备的电话号码、数据传送请求和传送的数据通过多个CPU扫描周期从CPU读取。讯息正在传送时您的程序不应当修改电话

怕扁蜓断i A 硇猱 断(7) 洼 褪踯 挥Φ毙蒂恼 洼 偷氛 寻

如果远程设备是另一个调制解调器模块，数据传送可使用密码功能，方法是在电话号码配置中输入远程调制解调器模块的密码。

回叫功能不得与数据传送一起使用。

## 密码保护

### [返回顶端](#)

调制解调器模块的密码安全为选用功能，可使用调制解调器扩充向导启用。调制解调器模块使用的密码与CPU密码不同。调制解调器模块密码是一个分开的8字符（全部大写字母或数字）密码，在允许呼叫人存取所连接的CPU之前，呼叫人必须向调制解调器? 榷? 榷? 妹芳榷 8 妹芳眺魑 髦平雅骸骸? 榕渲每桐囊徊糠执焮(0) CPU的V内存中。必须将调制解调器模块配置块下载至所连接CPU的数据块中。

如果CPU在系统数据块中启用密码安全功能，呼叫者则必须提供CPU密码，才能存取任何密码保护功能。

## 安全回叫 返回顶端

调制解调器模块的回叫功能为选用功能，用调制解调器扩充向导配置。回叫功能为所连接的 CPU 提供额外的安全保护，方法是仅限从预定义的电话号码存取 CPU。当回叫功能被启用时，调制解调器模块回答所有输入呼叫、核实呼叫人、然后断开线路。如果是经授权呼叫人，调制解调器模块则为呼叫人拨预定义电话号码，并允许呼叫人存取 CPU。

调制解调器模块支持三种回叫模式：

- 向单一预定义电话号码回叫
- 向多个预定义电话号码回叫
- 向任意电话号码回叫

回叫模式是通过在调制解调器扩充向导中勾选适当的选项并定义回叫电话号码的方式选择。回叫电话号码存储在调制解调器模块配置块中，该程序块存储在所连接的 CPU 的数据块中。

回叫的最简单形式是单一预定义电话号码。如果仅在调制解调器模块配置块中存储一个回叫号码，每当调制解调器模块回答一个输入呼叫时，就会通知呼叫人回叫已启用，断开线路，然后拨叫配置块中指定的回叫号码。

调制解调器模块也支持多个预定义电话号码回叫。在该模式中，会要求呼叫人提供一个电话号码。如果提供的电话号码与调制解调器模块配置块中预定义的一个电话号码相符，调制解调器模块则会断开线路，然后使用配置块中的匹配电话号码回叫。用户最多可配置 250 个回叫号码。

当有多个预定义回叫号码时，与调制解调器连接时提供的回叫号码必须与调制解调器模块的配置块中的数目相符（除头两位数外）。例如，如果配置的回叫为 91(123)4569999（因为需要拨外线（9）和长途（1）），为回叫提供的数字可以是下列任何一个数

- 91(123)4569999
- 1(123)4569999
- (123)4569999

所有上述电话号码均被视为回叫匹配号码。调制解调器模块在执行回叫功能时使用配置块中的回叫电话号码，在上例中为 91(123)4569999。在配置多个回叫号码时，请核实所有的电话号码均应为独特的号码（除头两位数外）。在比较回叫号码时，仅使用电

昂怕脍械氛 肿址 T 谄冉匣咏泻怕脍保 黍悸嵌汉呕蛭e胖 嗟淖址

对任何电话号码的回叫在调制解调器扩充向导中设置，方法是在回叫配置过程中选择“启用向任何电话号码回叫”选项。如果该选项被选择，调制解调器模块回答输入呼叫，并要求提供一个回叫电话号码。呼叫人提供该电话号码后，调制解调器模块则断开线路，并拨叫该电话号码。该回叫模式仅提供一种向调制解调器模块的电话连接收取电话费用的方法，而不为 S7-200

CPU 提供任何安全保障。如果使用此种回叫模式，则应当使用调制解调器模块密码。

调制解调器模块密码和回叫功能可以同时启用。调制解调器模块要求呼叫人在执行回叫之前提供正确的密码。

## EM 241 配置表

### 返回顶端

所有的文字讯息、电话号码、数据传送信息、回叫号码和其他选项均存储在调制解调器模块配置表中，该表必须载入 CPU 的 V 内存小 5 号平獾骸酪 尸 汕虹莳媚 寄 5.髦平獾骸酪? 榕渲帽恚 STEP 7-Micro/WIN 然后将调制解调器模块配置表放入下载到 S7-200 CPU 的数据块中。

调制解调器模块在启动时和 CPU 的任何“停止向运行”过渡的五秒钟内读取该配置表。只要调制解调器模块与 STEP 7-Micro/WIN 联机，调制解调器模块就不会从 CPU 读取新的配置表。如果在调制解调器模块联机时下载新配置表，在联机通话结束

保 髦平獾骸酪? 榛幽寥 厶 厦渲帽恚

如果调制解调器模块探测到配置表中的错误，位于模块前侧的模块良好（MG）LED 会闪烁。检查 STEP 7-

Micro/WIN 中的 PLC 信息屏幕，或读取 SMW220（用于 0 号模块槽）中的数值，了解有关配置错误的信息。下表列出了调制解调器

? 榕渲么碧蟆日缙 褂玫髦平獾骸酪 尸 汕虹冀 5.髦平獾骸酪? 榕渲帽恚

STEP 7-Micro/WIN 会在建立配置表之前检查数据。

### 错误

### 说明

0000	无错误
0001	无 24 VDC 外接电源
0002	调制解调器故障
0003	无配置块 ID — 配置表启动时的 EM 241 识别对本模块无效。
0004	配置块超出范围 — 配置表指针没有指向 V 内存，或配置表的某些部分超出所连接 CPU V 内存的范围。
0005	配置错误 — 回叫启用，回叫电话号码数目等于 0 或大于 250。讯息数目超过 250。讯息电话号码数目超过 250，或讯息电话号码的长度超过 120 个字节。
0006	国家选择错误 — 两个旋转开关上的国家选择不是受支持的数值。
0007	电话号码过大 — 回叫启用，回叫号码长度超过最大限度。
0008至00FF	保留
01xx	回叫号码 xx 中出现错误 — 回叫电话号码 xx 中出现非法字符。第一个回叫号码的数值 xx 将是 1，第二个回叫号码的数值将是 2，其余依此类推。
02xx	电话号码 xx 中出现错误 — 讯息电话号码 xx 或数据传送电话号码中的一个域包含一个非法数值。第一个电话号码的数值 xx 将是 1，第二个电话号码的数值 xx 将是 2，其余依此类推。
03xx	讯息 xx 中出现错误 — 讯息或数据传送号码 xx 超过最大长度限制。第一则讯息的数值 xx 将是 1，第二则讯息的数值 xx 将是 2，其余依此类推。
0400至FFFF	保留

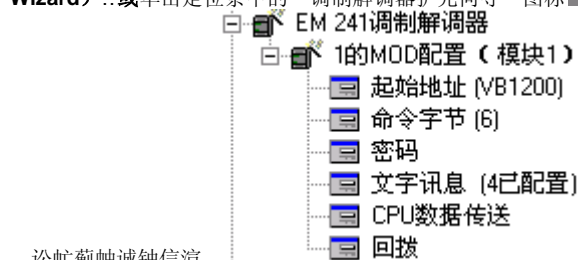
另请参阅：  
[调制解调器扩充向导概述](#)

## 2.5.9 EM241调制解调器向导概述

欲使用调制解调器扩充向导，请遵循以下步骤：

1. 选择菜单指令工具 (**T**ools) > 调制解调器扩充向导 (**M**odem Expansion

Wizard) ..或单击定位条中的“调制解调器扩充向导”图标  打开向导或打开指令树中的Wizards文件夹并随后打



2. 单击“下一个”按钮前往调制解调器扩充向导的每个新页。
3. 当您完成配置时，单击“完成”按钮。

调制解调器扩充向导允许您对以下任意一个项目进行编程：



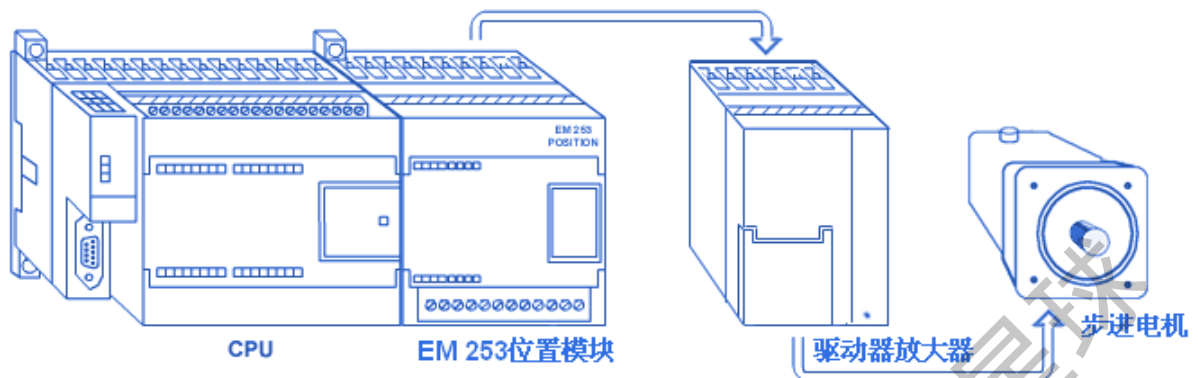
- 用于S7-200 PLC的模拟或手机调制解调器
- EM 241调制解调器模块的参数和操作

您希望配置哪一个项目？  
[模拟或手机调制解调器](#)  
[EM 241调制解调器扩充模块](#)

## 2.5.10 EM253位置模块性能

EM 253位置模块是一种S7-200特别功能模块，生成用于步进电机或伺服电机速度和位置开环控制装置的脉冲串。该模块通过扩充 I/O总线与S7-200通讯，并在I/O配置中显示为配备八个数字输出的智能模块。

根据存储在S7-200 V内存中的配置信息，位置模块生成控制移动所需的脉冲串。



为了简化应用程序中位置控制的使用，STEP 7-Micro/WIN提供一个[位置控制向导](#)，该向导允许您在几分钟时间内配置位置模块。STEP 7-Micro/WIN还提供一个[控制面板](#)，该控制面板允许您控制、监管和测试位置操作。

位置模块提供单轴开环位置控制所需的功能和性能：

- 高速控制，范围从每秒12次脉冲至每秒200,000次脉冲
- 跳动（S曲线）或线性加速 / 减速
- 测量系统配置的工程单位或脉冲
- 可配置间隙补偿
- 绝对位置控制
- 相对位置控制
- 手动位置控制
- 连续操作
- 最多为25个位置轮廓，每个轮廓最多有四种速度变化
- 四个参考点查找模式（每个序列有起始查找方向和最终接近方向选择）
- 可移动域接线接头

可使用STEP 7-Micro/WIN建立位置模块使用的所有配置和轮廓信息。该信息用您的程序块下载至S7-200。因为所有用于位置控制的信息均存储在S7-200，您可以无须重新为模块编程或配置模块即可更换位置模块。S7-200为位置模块接口保留8位程序图象输出寄存器（Q内存）。S7-200中的应用程序使用这些位控制位置模块作业。这8个输出位不与位置模块的任何实际域输出连接。

位置模块提供五个数字输入和四个数字输出，这些输入和输出为您的位置应用程序提供接口。请参阅下表。这些输入和输出与恢复？榭挥谕 磺 颀

信号	说明
STP	STP输入使模块停止。您可以在位置控制向导内选择希望使用的STP操作。
RPS	RPS（参考点开关）输入为绝对移动操作建立参考点或原位。
ZP	ZP（零脉冲）输入帮助建立参考点或原位。通常电机驱动程序 / 放大器电机每次转动时向ZP发出一次脉冲。
LMT+ LMT-	LMT+ 和 LMT- 输入被用于建立移动的最大限度。位置控制向导允许您配置LMT+ 和 LMT- 输入操作。
P0 P1 P0+, P0- P1+, P1-	P0和P1是控制电机移动和方向的开放式漏极晶体管脉冲输出。P0+、P0-和P1+、P1-是分别提供P0和P1相同功能的差分脉冲输出，同时提供优异的信号质量。开放式漏极输出和差分输出同时为现用。您根据电机驱动程序 / 放大器的接口要求选择使用哪一套脉冲输出。
DIS	DIS是用于禁用电机驱动程序 / 放大器的开放式漏极晶体管输出。
CLR	CLR是用于清除伺服脉冲计数器寄存器的开放式漏极晶体管输出。


另请参阅：  
[位置控制向导概述](#)

## 2.5.11 EM253位置PTO/PWM向导概述

**注释：**您必须为位置模块建立一个配置 / 轮廓表，以便使模块控制位置应用程序。位置控制向导还允许您脱机建立配置 / 轮廓表 D 梢裁诮挥氩沧拔恢媚？榭 S7-200 CPU连接的情况下建立配置。

欲使用位置控制向导，请遵循以下步骤：

1. 选择菜单命令工具 (I) >

位置控制向导..或单击浏览条中的“位置控制向导”图标打开此向导或打开[指令树](#)中的“向导”文件夹并随后

EM 253位置

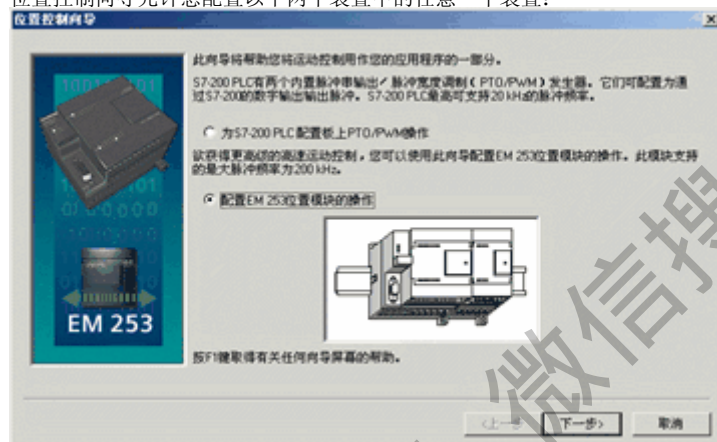
- 2的POS配置 (模块0)
  - 起始地址 (VB1000)
  - 命令字节 (6)
  - 测量单位 (cm)
  - 电机配置
  - 参考点设置
  - 运动轮廓 (2已配置)

PTO/PWM

- Q0.0的PTO配置 (输出Q0.0)
  - 起始地址 (VB5000)
  - 脉冲操作 (PTO)
  - 电机配置
  - 运动轮廓 (1已配置)

- 单击“下一步 >”按钮前往位置控制向导的每个新页。
- 完成配置后单击“完成”。

位置控制向导允许您配置以下两个装置中的任意一个装置：



机载PTO/PWM发生器  
PLC有两个可以配置用于输出脉冲串输出的内置脉冲串输出（PTO）/脉冲宽度调制（PWM）发生器。支持的最大脉冲速率为20kHz。



- EM 253位置模块  
253位置模块。模块支持的最大脉冲速率为200kHz。

如果您需要更高级的高速位置控制，则可配置EM

您完成的配置被存储在V内存内的[配置/轮廓表](#)中。位置模块使用的配置和轮廓信息是STEP 7-Micro/WIN项目的一部分，被下载至S7-200。因为位置控制所需的所有信息均存储在S7-200中，取代位置模块则与取代任何其他S7-200 I/O模块一样简便。

STEP 7-Micro/WIN还提供一个[EM 253控制面板](#)。该控制面板可简化您的应用程序位置控制部分的测试，方法是提供允许您控制、监控和测试位置操作的功能。EM 253控制面板允许您编辑向导设置，而无须通过向导屏幕返回。

欲存取EM 253控制面板：

- 单击定位条中的EM 253控制面板图标，或
- 选择菜单命令工具 (Tools) > EM 253控制面板... (EM 253 Control Panel)

您希望配置什么？

[PTO或PWM发生器](#)

[EM 253模块](#)

## 2.5.12 EM253控制板

为了协助您开发您的位置控制解决方案，STEP 7-Micro/WIN提供EM 253控制面板。操作、配置和诊断程序标记便于您在开发程序的启动和测试阶段监控和控制位置模块的操作。使用EM 253控制面板核实位置模块接线正确、调整配置数据以及测试每个移动轮廓。



**提示：** 在使用EM

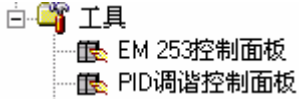
253控制面板之前，将位置控制项目下载至PLC。因为位置控制向导对程序块、数据块和系统块作出改动，请勿必下载所有三种模块。否则，位置模块可能没有正常操作所需的所有程序元件。在模块和控制面板开始作业之前，必须初始化模块的SM内存。当您用1.2.1或更高版本的固件将数据块和系统块下载至CPU时，初始化会自动完成。如果您有1.2版或更早版本的CPU，则必须执行POSx\_CTRL子例行程序，初始化特殊内存。位置控制向导将配置和轮廓数据存储存储在PLC V内存中。当您启动EM 253控制面板时，Micro/WIN尝试从PLC V内存读取有效配置数据。如果Micro/WIN无法与PLC通讯，您会收到一则通讯错误。如果V内存不包含有效位置配置，Micro/WIN会显示一则V内存配置错误。

请使用下列方法之一打开EM 253控制面板：

- 选取**工具(I)** > **EM 253控制面板** 菜单命令。



- 单击**浏览条**上的EM 253控制面板按钮。
- 打开**指令树**上的工具文件夹，然后打开EM 253控制面板。



操作标记



控制面板的操作标记允许您与位置模块操作对话。控制面板显示位置模块的当前速度、位置和方向。您还可以看到输入和输出 LED 的状态（除脉冲LED外）。

控制面板允许您与位置模块对话，方法是更改速度和方向、停止和启动工具以及使工具慢进（如果移动停止）。

您还可以生成下列位置命令：

- 启用手动操作 — 该命令允许您使用手动控制为工具定位。
- 运行位置轮廓 — 该命令允许您选择需要执行的轮廓。控制面板显示由位置模块执行的轮廓状态。
- 查找参考点 — 该命令使用配置的搜索模式查找参考点。
- 载入参考点偏移 — 在您使用手动控制使工具慢进至新的零位时，则可载入参考点偏移。
- 重新载入当前位置 — 该命令更新当前位置数值。
- 激活DIS输出并取消激活DIS输出 — 这些命令打开和关闭位置模块的DIS输出。
- 脉冲和CLR输出 — 该命令在位置模块的CLR输出中生成一个50毫秒的脉冲。
- 示教位置轮廓 — 该命令允许您在以手动方式放置工具时保存位置轮廓和步进的目标位置和速度。控制面板显示位置模块正在执行的轮廓状态。
- 载入模块配置 — 单击执行使模块从V内存读取配置。
- 移至绝对位置 — 该命令指定一个目标速度和移动的绝对位置。该选项要求定义0位置。
- 移动相对数量 — 该命令指定一个目标速度和您希望从当前位置移动的距离。您可以指定一个正向距离或负向距离。
- 重设模块命令接口 — 如果模块似乎不对命令作出应答，该命令会十分有用。该选项清除模块的命令字节，并设定 DONE位。

#### 配置标记



控制面板的配置标记允许您检视和修改存储在 S7-200 数据块中的位置模块的当前配置设置。

您必须选择 PLC 复选框中的“允许更新模块配置”，才能修改当前配置设置。

修改配置设置后，单击“更新配置”按钮，更新 STEP 7-Micro/Win 项目和 S7-200 数据块中的设置。

诊断程序标记



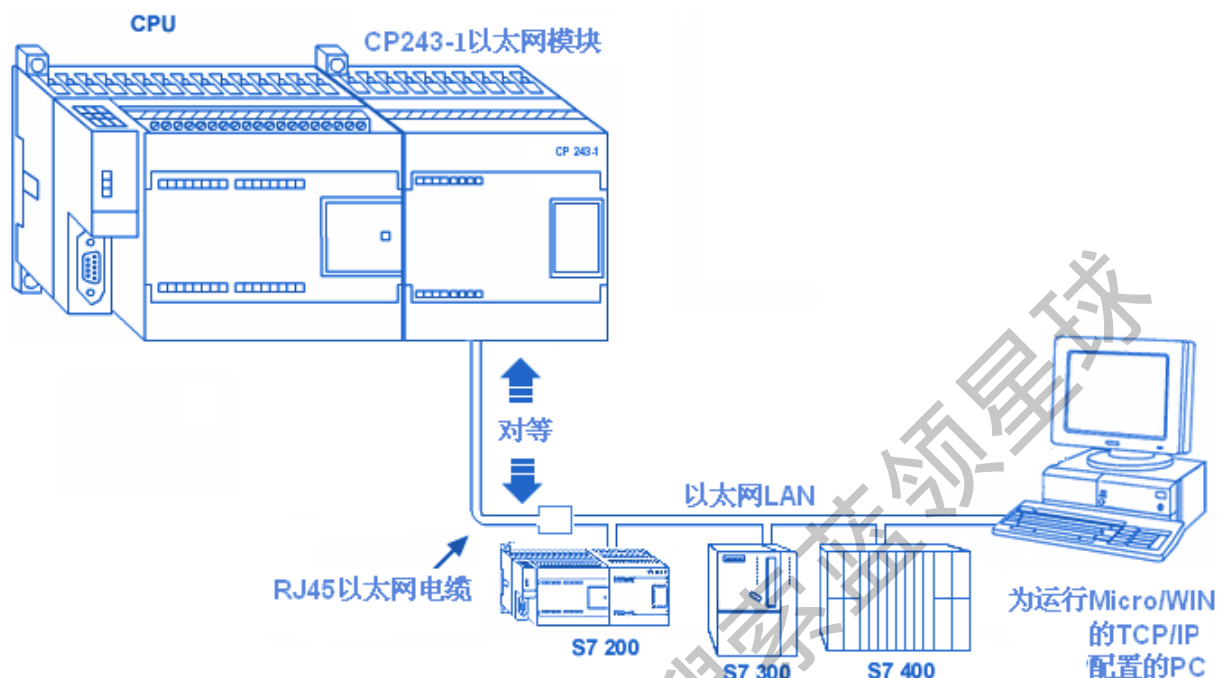


控制面板的诊断程序标记允许您检视有关位置模块的诊断信息。您可以检视有关位置模块的具体信息，例如 I/O 链接中的模块位置、模块类型和固件版本号以及用作模块命令字节的输出字节。控制面板显示因执行命令操作而导致的错误状况。您还可以检视位置模块报告的任何错误状况。

另请参阅：  
[位置模块的错误代码](#)

### 2.5.13 CP243-1 以太网模块性能

CP243-1 以太网模块是 S7-200 系列的通讯处理器，可使 S7-200 PLC 与工业以太网网络链接。以太网模块使用存储在 S7-200 V 内存中的配置信息生成以太网通讯所需的连接。



以太网模块包括以下功能：

- 根据TCP/IP和ISO标准进行通讯
- 工厂安装MAC地址
- 与其他S7设备的对等通讯
- 自动传感全双工或半双工通讯，10 MB和100 MB
- 多个连接（最多为8个）
- 客户机或服务配置选项
- 初始化、重新配置和数据传输指令（ETHx\_CTRL、ETHx\_CFG、ETHx\_XFR）

以太网模块允许S7-200 PLC与S7-300和S7-400设备通讯。您还可以与SIMATIC操作员面板（OP）建立通讯。您通过以太网向导配置以太网模块的通讯参数。

另请参阅：


[以太网向导概述](#)

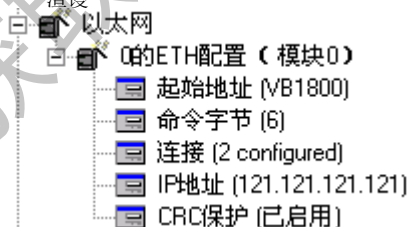
## 2.5.14 CP243-1以太网向导概述

欲使用以太网向导，请遵循以下步骤：

1. 选择菜单命令 **工具 > 以太网向导 (Tools > Ethernet Wizard)** ..



或单击浏览条中的“以太网向导”图标  打开向导或打开指令树中的Wizards文件夹并随后打开此向导或某现有向导

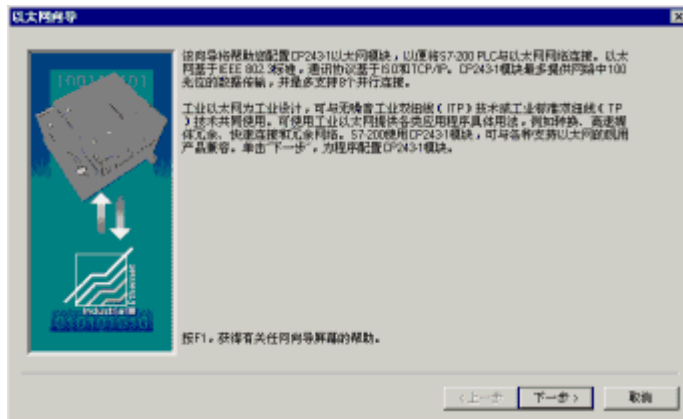


2. 单击“下一步 >”按钮，进入以太网向导的每个新篇章。
3. 完成配置后，单击“完成”按钮。

以太网向导是用于配置以太网模块的主要机制。

以太网向导帮助您配置以太网模块，以便将S7-200 PLC与工业以太网网络连接。然后将本配置存储在适当的数据块区域。您最多可配置8个异步并行连接。但一次只能建立一种此类以太网连接。您必须为您请求的每个连接重复连接配置程序。您一旦完成向导操作并下载配置，以太网模块即使用在S7-200 PLC的SM位置中提供的指针数值存取配置和应用程序数据。

以太网向导还会建立支持对等通讯的独特以太网通讯指令子例行程序。



另请参阅：

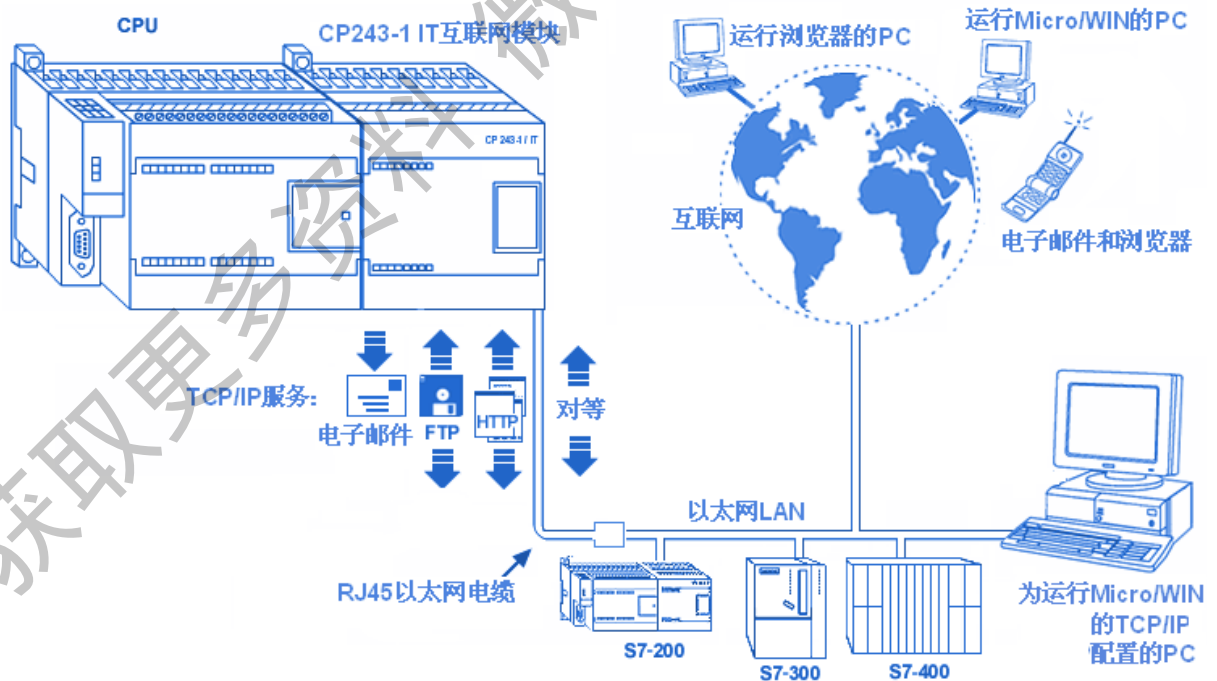
[CP243-1以太网模块性能](#)  
[以太网向导](#)

与以太网相关的指令：

[ETHx\\_CTRL子例行程序](#)  
[ETHx\\_CFG子例行程序](#)  
[ETHx\\_XFR子例行程序](#)

## 2.5.15 CP243-1 IT互联网模块性能

CP243-1 IT互联网模块是用于S7-200系列的通讯处理器，该模块允许S7-200 PLC用标准以太网LAN电缆与互联网连接。



互联网模块支持以下TCP/IP通讯服务。

- 使用用户ID和密码获得保密FTP（文件传输协议）和HTTP（超级文本传输协议）服务器登入
- 将带有嵌入PLC数据的电子邮件讯息发送至标准SMTP邮件服务器
- 将文件传输至远程服务器的FTP客户机服务
- 通过远程FTP客户机将文件传输至内部8 Mbyte快闪存文件系统并从该文件系统接收文件传输的FTP服务器服务
- 用于远程互联网浏览器存取的HTTP服务器服务

- 用于互联网通讯的S7-200程序指令（ETHx\_EMAIL和ETHx\_FTPC）
- 互联网性能的配置数据被直接存入互联网模块的快闪内存。

互联网模块还支持以太网模块提供的所有功能（CP243-1）

- 根据TCP/IP和ISO标准进行通讯
- 工厂安装MAC地址
- 与其他S7设备进行对等通讯
- 自动传感全双工或半双工通讯，10 MB和100 MB
- 多个连接（最多为8个）
- 以太网客户机或服务器配置选项
- 初始化、重新配置和数据传输程序指令（ETHx\_CTRL、ETHx\_CFG、ETHx\_XFR）

储存在V内存中的配置数据被用于生成以太网通讯所需的连接。必须下载向导修改的项目代码块，才能在PLC中操作和更新CP243-1 IT配置。

您必须通过互联网向导配置互联网模块的通讯参数。以太网连接开始操作后，您可以使用Windows DOS

FTP命令将三个向导生成的文件复制至CP243-1

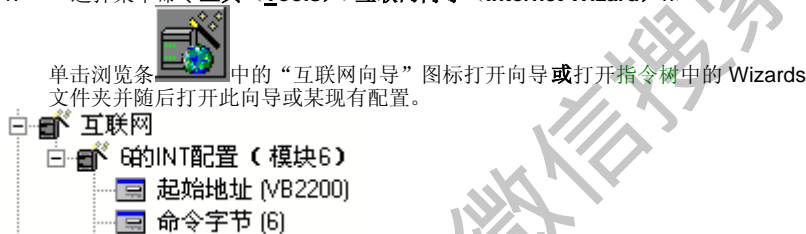
IT快闪内存。这三个文件（\*.edb、\*.fdb和\*.udb）在互联网向导中启用您选择的互联网通讯性能。

有关CP243-1 IT配置程序的详细总结，请参阅[互联网向导概述](#)标题中的内容。

## 2.5.16 CP243-1 IT互联网向导概述

欲启动互联网向导，请遵循以下步骤：

1. 选择菜单命令**工具 (Tools) > 互联网向导 (Internet Wizard) ...**



2. 单击“下一项”按钮，继续执行每一个新“互联网向导”段。
3. 完成配置后，单击“完成”按钮。

### 注释：

互联网模块必须由互联网向导配置，但完成互联网向导操作在总体配置程序中仅需执行一个步骤。请阅读下列步骤，了解完整CP243-1 IT配置程序。

### 先决条件：

运行STEP 7-Micro/WIN的PC必须与以太网连接，才能完成CP243-1

IT配置程序。安装Micro/WIN之前，必须安装PC的网络接口硬件和软件驱动程序。如果您的PC没有网络连接，解除安装STEP 7-Micro/WIN、安装或配置网络硬件和软件，然后重新安装STEP 7-Micro/WIN。

### CP243-1 IT模块配置和互联网通讯步骤

1. 请告诉您的网络管理员，您需要一个新的LAN连接和IP地址。网络管理员会为您指定一个IP地址，并告诉您是否需要使用BOOTP服务器程序。BOOTP服务通常由预先租用的DHCP服务器处理。

如果未使用BOOTP程序，则必须在完成互联网向导程序之前由网络管理员为您指定一个静态IP地址、子网掩模和网关地址。

如果使用BOOTP程序，您必须向网络管理员提供印在模块顶盖下方的独特的MAC地址（例如，08-00-06-70-8C-23）。网络管理员将为您指定一个对应的IP地址。如果其后用一个新模块取代CP243-1

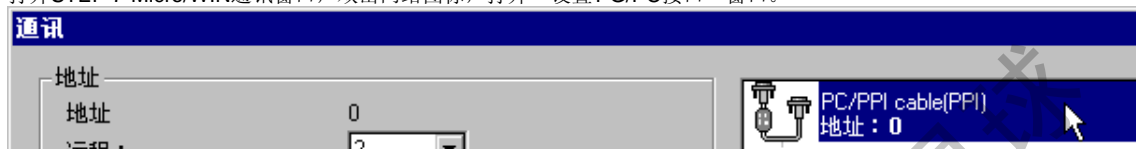
IT模块，网络管理员则必须保留IP地址，用于替换模块的MAC地址。

2. 断开CPU电源，将S7-200 CPU连接至CP243-1 IT带状电缆，并将24V电源连接至CP243-1 IT模块（+24V至“L+”和0V至“M”螺旋终端）。取下插入式内存盒（如果安装），“L+”和“M”螺旋终端，使电源启动重设不会盖写新的以
3. 启动S7-200 CPU和24V电源。用PC/PPI电缆、PC COM卡或以太网TCP/IP（如果使用BOOTP服务）通过集成CPU PPI COM端口建立STEP 7-Micro/WIN与CPU之间的通讯。互联网向导使用该初始通讯路径设置新的以太网通讯路径。该步骤的其余部分假设您在
4. 打开STEP 7-Micro/WIN项目，运行互联网向导。完成向导屏幕操作，并单击“完成”按钮。单击“是”，完成向导配置，并在程序性
5. 使用“文件>下载”命令，将新代码块从STEP 7-Micro/WIN传输至S7-200 PLC。下载所有的程序块、数据块和系统块。该下载传输设置互联网向导中指定的CP243-1 IT模块IP地址、子网地址、网关地址和管理员数据。
6. 取决于您的互联网模块的初始状态，可能不会自动重新配置。如果不重新配置，执行STEP 7-

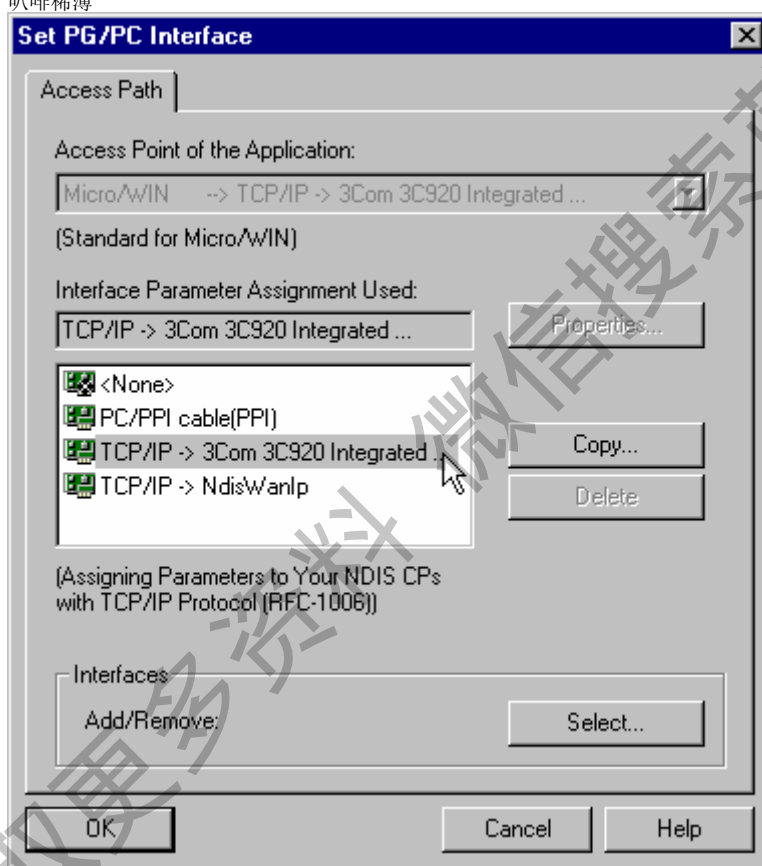
Micro/WIN “PLC>电源启动重设” 软件命令，或关闭PLC电源后再重新开启，初始化CP243-1 IT模块。

7. 用一根RJ45以太网电缆连接CP243-1 IT模块和以太网LAN。运行STEP 7-Micro/WIN的PC必须也有一个以太网LAN，才能完成配置。您可以使用Windows DOS命令提示窗口发出一条“ping xxx.xxx.xxx.xxx”命令。将模块的IP地址替换为xxx.xxx.xxx.xxx，测试PC和CP243-1 IT模块之间的以太网路径。
8. 当PC与CP243-1 IT以太网连接操作时，将STEP7-Micro/WIN至PLC通讯路径从CPU的集成PPI端口更改为CP243-1 IT以太网LAN连接。



打开STEP 7-Micro/WIN通讯窗口，双击网络图标，打开“设置PG/PC接口”窗口。



用鼠标单击PC使用的TCP/IP设备，选择PG/PC存取路径，然后单击“确认”按钮。单击存取路径更改核实窗口中的咖啡稀薄

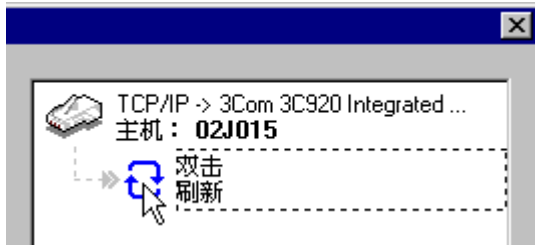


TCP/IP网络设备应当显示在“通讯”窗口的右上方。在“远程”域中输入CP243-1

IP地址。另一种方法是单击书本图标，打开IP地址浏览器，并选择一个以前指定的地址。您还可以用IP地址浏览器刷新图标，刷新列表。



双击网络刷新图标，更新通讯使用期限列表。



刷新的使用期限显示与以太网连接的CPU。核实已选择目标CPU。如下图所示，所选CPU和地址被增高。



单击“确认”，关闭“通讯”窗口。STEP 7-

Micro/WIN现在通过以太网LAN通讯，可以移除PPI端口电缆。所有的STEP 7-Micro/WIN性能均通过新以太网路径（运行 / 停止、下载、上载、状态收集等）提供。

9. 使用“文件 > 下载”命令通过以太网路径从STEP 7-Micro/WIN向CP243-1 IT模块传输配置。

成功完成步骤9所需的条件：

- 运行STEP 7-Micro/WIN的PC/PG必须通过以太网与CP243-1 IT模块连接。
- 当前项目中存在一个互联网向导配置。
- 在互联网向导配置中指定的模块位置实际存在一个CP243-1 IT模块。

该下载传输设置在互联网向导中指定的CP243-1 IT模块用户、电子邮件和FTP数据。如下图所示，下载窗口现在指示从STEP 7-Micro/WIN至CP243-1 IT模块的下载路径。



10. 互联网向导已建立操作您选择的性能的子例行程序（ETHx\_CTRL, ETHx\_CFG, ETHx\_XFR, ETHx\_EMAIL和ETHx\_FTPC）。打开指令树中的“调用子例行程序”文件夹，将指令放入程序块中，由程序执行这一功能。

如果您希望更改部分或全部配置参数，重新运行互联网向导：

- 如果您修改以太网配置参数，则必须下载所有的项目块，并重新执行电源启动重设。
- 如果您修改互联网参数（电子邮件、FTP或用户登录），则必须重新执行步骤9，以便重新载入\*.edb、\*.fdb和\*.udb文件。

11. 快闪存文件系统由工厂提供，已预先载入下表所示目录和文件集。作为范例提供的快闪存文件可供您使用或修改。除\_\_S7Sys目录外，所有的目录均可删除。每次CP243-1 IT初始化时，所有的默认目录均会被建立为空目录，并无法找到这些目录。您可以建立自己的网页和小程序，但如果M 媛 S7-200数据，则必须使用所提供的“放置 / 获取” JAVA组件。

编程细节请参阅CP243-1 IT技术参考资料：

用于工业以太网和信息技术的CP243-1 IT通讯处理器

#### SIMATIC NET

#### CP 243-1 IT Communications Processor for Industrial Ethernet and Information Technology

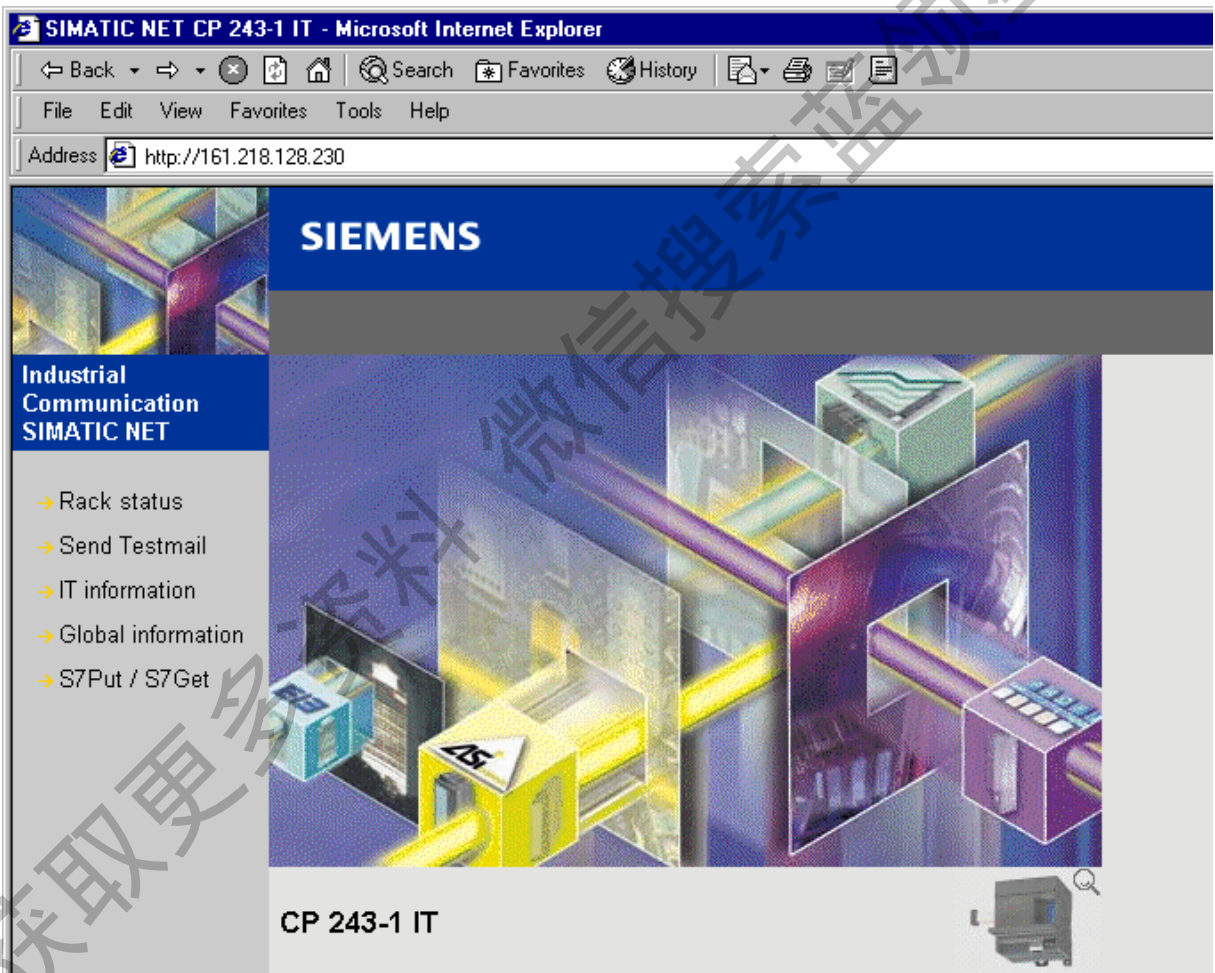
/flash:	快闪存文件系统根目录	
/_S7Sys	（用于FTP服务的读取保护）	内部使用
/images	内部使用	
*.gif	（用于状态页）	
/itsdb	内部使用	
bin		
sendmail.htm		
cp243_styles.css		
/applets	小程序目录	
*.jar		
/config	配置文件目录，最初为空目录	

<projectname>.edb  
 <projectname>.fdb  
 <projectname>.udb  
 /examples  
 /images  
 \*.gif  
 \*.jpg  
 \*.htm  
 /user  
 index.htm  
 readme.htm

CP243-1 IT 主页范例 JAVA 小程序和组件范例

用于用户文件

将 CP243-1 IT 模块地址输入互联网浏览器，检视示范网页。



另请参阅：

[CP243-1 IT 互联网模块性能  
互联网向导](#)

与互联网相关的指令：

[ETHx\\_EMAIL 子例行程序](#)  
[ETHx\\_FTPC 子例行程序](#)

与以太网相关的指令：

[ETHx\\_CTRL 子例行程序](#)  
[ETHx\\_CFG 子例行程序](#)



[ETHx XFR子例行程序](#)

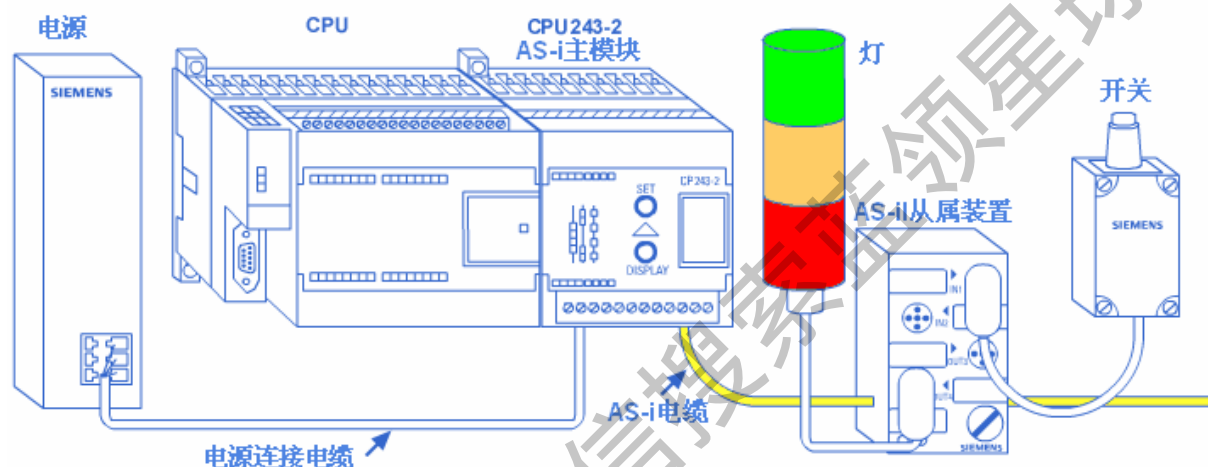
## 2.5.17 CP243-2 AS-i模块性能

CP243-2 AS-i主模块允许将AS-i网络连接至S7-200 PLC。

传动器 / 传感器接口或AS-i是用于最低级别自动化系统的单台主设备网络连接系统。CP243-2模块用作网络的AS-i主设备。传感器和传动器（AS-i从属装置）使用单条AS-i电缆，可通过AS-i主模块与PLC连接。AS-i主模块处理所有AS-i网络协调作业，并通过指定给主模块的I/O地址将来自传动器和传感器的数据和状态信息传送至PLC。AS-i从属装置是AS-i系统的输入和输出信道，只有在被AS-i主模块调用时才会成为现用。

AS-i向导可以为AS-i从属装置地址重新编程，帮助设置AS-i网络。

AS-i向导还可以配置项目，为I/O存取映射AS-i网络的从属装置。



注释：AS-i向导可用于第二代AS-i主模块CP243-2（1.x版或更高版本）。

另请参阅：  
[AS-i向导概述](#)

## 2.5.18 CP243-2 AS-i向导概述

欲使用AS-i向导，请遵循以下步骤：

1. 选择菜单命令 **工具 > AS-i向导** (**Tools > AS-i Wizard**)？或单击浏览条中的“AS-

向导”图标  打开此向导或打开 **指令树** 中的Wizards文件夹并随后打开此向导或某现有配置。

AS-i 的AS配置 (模块0)

- 起始地址 (VB100)
- 从设备类型

2. 单击“下一步 >”按钮前往AS-i向导的每个新部分。
3. 完成向导中的步骤后，单击“完成”按钮。

配置初始向导选项，以便：

[更改AS-i从属装置地址](#)

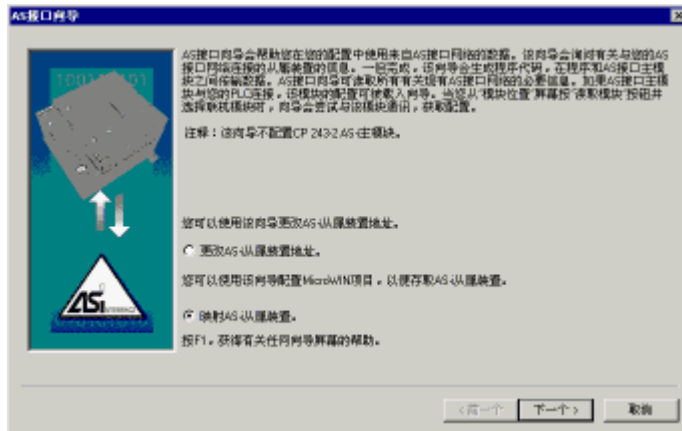
选择该选项，设置AS-i从属装置网络，方法是为存储在每个AS-i从属装置中的从属地址编程。欲为从属装置地址编程，您必须在STEP 7-Micro/WIN、连接CP243-2 AS-i主模块的S7-200 CPU和AS-i从属装置网络之间建立通讯。

或  
[配置项目，使之映射AS-i从属装置](#)

AS-i向导可帮助您建立在程序和AS-i从属装置之间传输数据所需的Micro/WIN项目代码，并为您提供映射至AS-i I/O位置的符号。

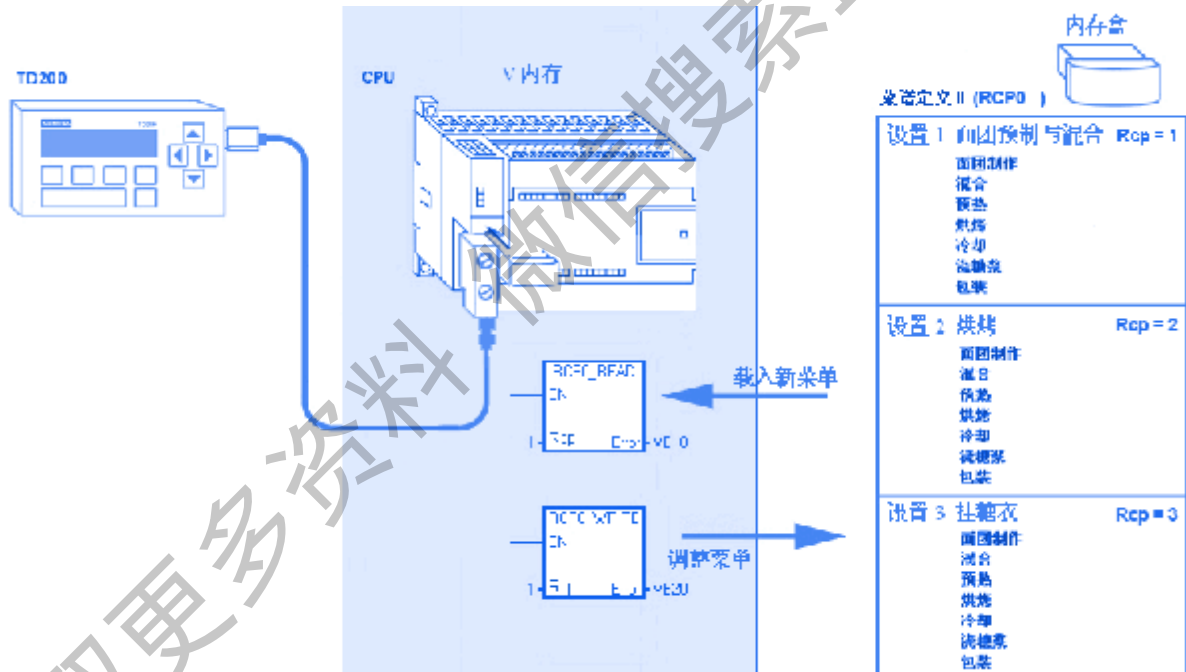
向导的这一部分可联机或脱机使用（未连接S7-200 CPU、CP243-

2或从属装置网络）。如果联机使用向导，向导可提供CP243-2模块信息，并进行AS-i网络比较。



另请参阅：  
AS-i模块性能

## 2.5.19 菜谱向导



注释：“菜谱向导”可以创建项目代码，用于在64千字节或256千字节永久性内存（EEPROM）盒中存储和获取菜谱数据。对菜谱蛇已端佳婿械闹C质加 S7-200 CPU 222、CPU 224、CPU 226第2.0版。

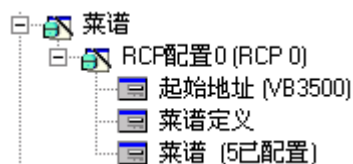
菜谱向导用于在内存盒中组织和载入自动化程序菜谱。将菜谱数据值转移到内存盒内，可以解放否则必需用来存储菜谱数据的 VÄ 注烟 0贰 P 碌牟似锥丁塞筒似资 搗峙淇稍诤似紫虹贾兄笨蛹 瓣6圆似资 荻暮竿 亩 赏u 甌略浦肋似紫虹蓟蚤捐 RCPx WRITE子例行程序的编程来完成。

欲使用菜谱向导，请遵循以下步骤：

1. 选择菜单命令工具 > 菜谱向导 (Tools > Recipe



Wizard) 或单击浏览条中的菜谱向导图标 打开此向导或打开指令树中的Wizards文件夹并随后打开此向导或某钟信渲漫



2. 单击"下一步 >"按钮前往菜谱向导的每个新页。
3. 完成向导中的步骤后，单击"完成"按钮。

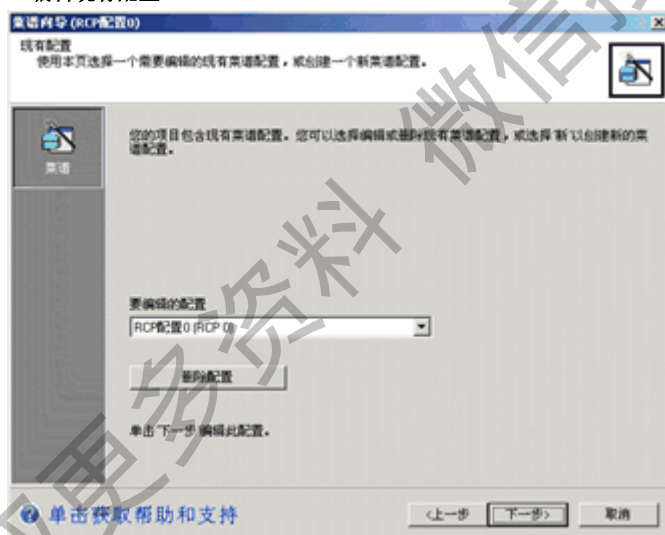
菜谱向导通过生成下列项目元件而创建存取内存盒接口的程序：

- 菜谱符号表，每个菜谱定义一个，用于关联菜谱符号名、V内存地址、菜谱注解
- 新的数据块标记，每个菜谱定义一个，用于为每个菜谱定义分配V内存地址和数值。您的程序可使用对这些V内存地址姆琶创媛（似资 尊
- 子例行程序RCPx\_READ将指定的菜谱数据值从内存盒复制到V内存。RCPx\_READ子例行程序用于将不同的菜谱载入V内存或刷新当前菜谱的V内存数值。
- 子例行程序RCPx\_WRITE将指定的菜谱数据值从V内存复制到内存盒。如果程序在V内存菜谱数据值中写入改动，并且您希望保存为永久菜谱更改，请使用RCPx\_WRITE。

### 步骤

- 1 编辑现有配置
- 2 菜谱定义
- 3 创建和编辑菜谱数据值
- 4 分配V内存
- 5 生成代码

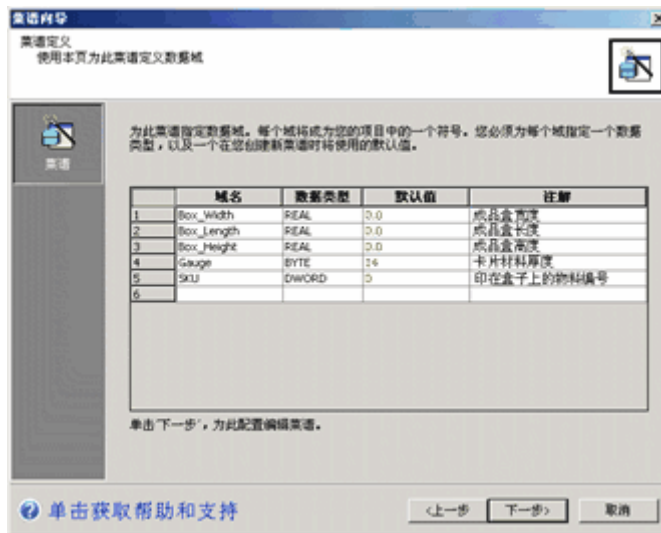
#### 第1步 编辑现有配置



单击配置下拉列表，从中选一个现有或新的菜谱定义。您最多可以有四个不同的菜谱定义。每个菜谱定义中可保存数百类似住

如果不存在现有菜谱配置，菜谱向导将由第2步开始。

#### 第2步 菜谱定义



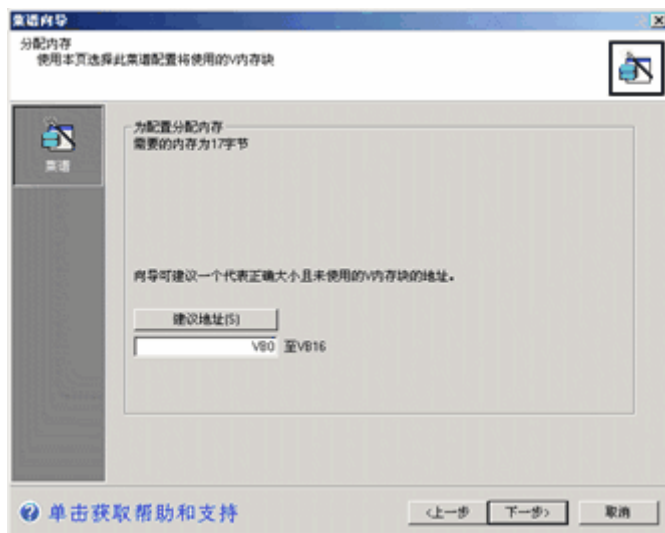
用鼠标单击以选取一个表数据单元，然后编辑该域。键入一个符号名作为域名。由下拉式列表中选一个数据类型。为成品盒宽度、成品盒长度、成品盒高度、卡片材料厚度、印在盒子上的物料编号。

### 第3步 创建和编辑菜谱数据值



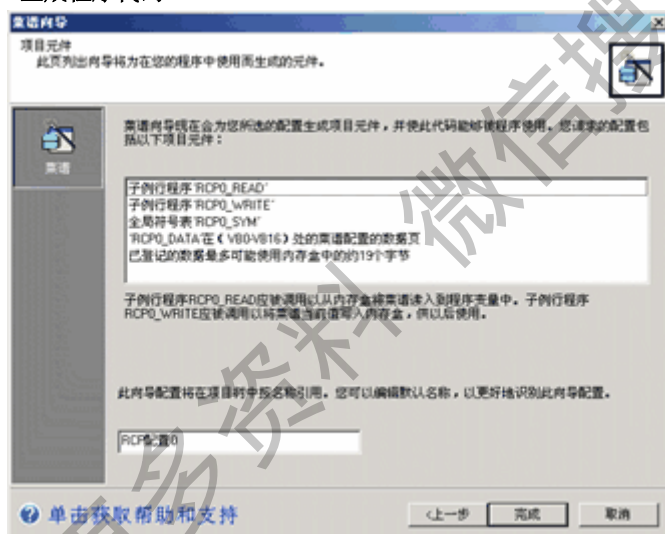
双击列标题，重新命名该菜谱，然后编辑默认值以创建新菜谱。单击菜谱数据的任何位置或注释列启用"新"按钮，然后单击"新"按钮插入新的菜谱。重复此过程即可基于当前菜谱定义插入、重新命名、编辑所有菜谱。

### 第4步 分配V内存



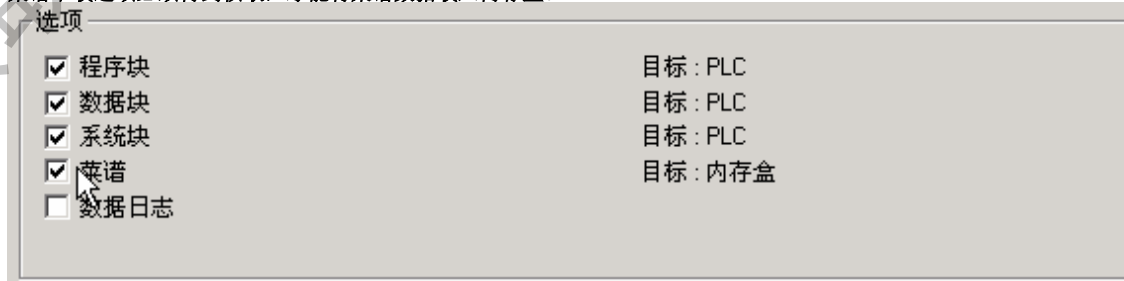
此向导将分配一个"数据块"标记以在V内存中存储菜谱配置。在数据块中指定您希望安放该配置的起始地址。您可以选择想要存放该配置块的V内存地址；如果您希望向导建议正确大小的未使用内存块地址，则单击"建议地址"按钮。配置块的大小根据您在向导中所作的具体选择不同而异。

### 第5步 生成程序代码



菜谱向导会为您的菜谱(符号表、数据块、子例行程序、内存盒数据)生成项目元件，并令您的程序可以使用这些元件。您必须将带有菜谱向导配置的项目下载至 S7-200 才能加以使用。如果某项目有菜谱配置，则下载窗口的默认设置为"菜谱"选项已被核取。

**菜谱下载选项必须得到核取，才能将菜谱数据载入内存盒。**



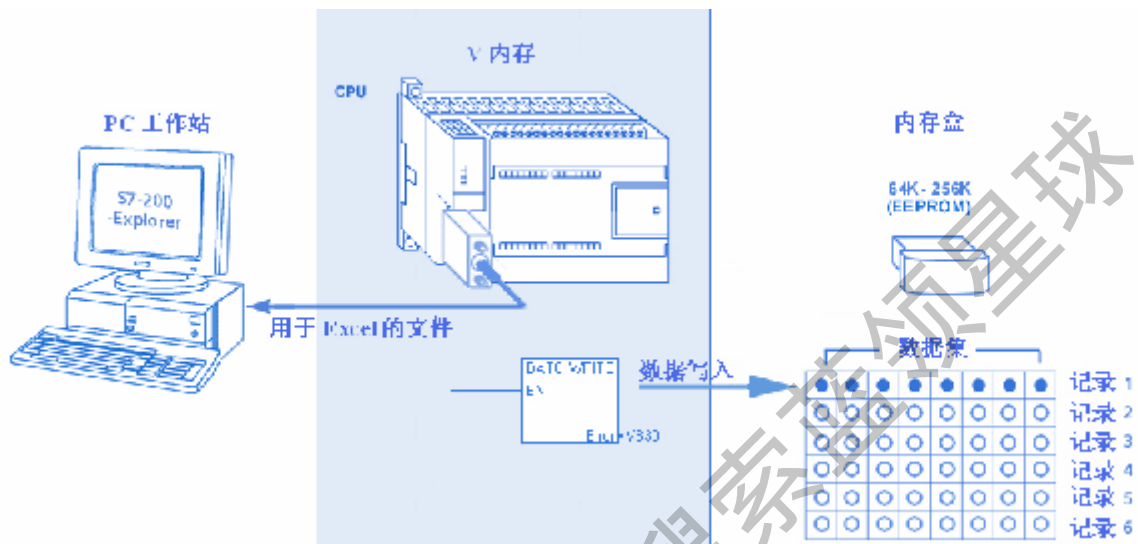
您的程序能够由用菜谱数据符号名指明的V内存地址读取菜谱数据。每执行一次RCPx\_READ就会将一个菜谱从内存盒粗频 V内存。

每执行一次RCPx\_WRITE就会将一个菜谱从V内存复制到内存盒。

另请参阅：

RCPx\_READ  
RCPx\_WRITE

## 2.5.20 数据日志向导



注释：“数据日志向导”可以创建项目代码，用于在64千字节或256千字节永久性内存（EEPROM）盒中存储数据。对数据日志向导已端挂婧械闹 C 质加赞7-200 CPU 222、CPU 224、CPU 226第2.0版。

数据日志向导可帮助您在内存盒内存储工艺测量数据。将工艺数据的存放转移到内存盒内，可以解放否则必需用来存储此数据的V内存地址。记录定义会被输入到数据日志向导内。对记录定义的改动可通过重新运行数据日志向导来完成。

S7-200 Explorer工具能够复制已存储在内存盒内的数据日志记录。使用S7-200

Explorer，找到内存盒图标，选取数据定义，然后上载有关的数据日志记录。由此上载的CSV（逗号分隔值）文件将会打开与CSV文件关联的Windows应用程序（例如MS Excel或MS Notepad）。请使用此Windows应用程序在您的磁盘驱动器上保存新数据日志CSV文件的一个永久副本。

欲使用数据日志向导，请遵循以下步骤：

1. 选择菜单命令工具 > 数据日志向导（Tools > Data Log



Wizard) 或单击浏览条中的数据日志向导图标 打开此向导或打开指令树中的Wizards文件夹并随后打开此向导



2. 单击“下一步 >”按钮前往数据日志向导的每个新页。
3. 完成向导中的步骤后，单击“完成”按钮。

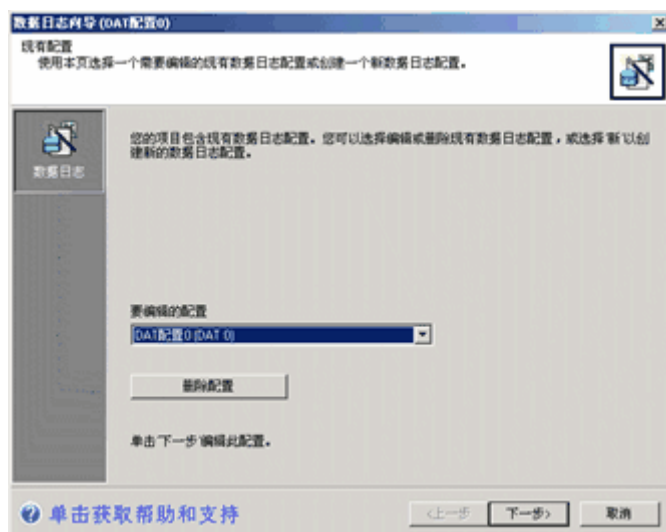
数据日志向导通过生成下列项目元件而创建存取内存盒接口的程序：

- 数据日志符号表，每个数据日志定义一个，用于关联数据日志符号名、V内存地址、数据日志注解
- 新的数据块标记，每个数据日志定义一个，用于为每个记录定义分配V内存地址。您的程序使用这些V内存地址来积累鼻叭罩炯锹肌
- 子例行程序DATx\_WRITE将指定的记录从V内存复制到内存盒。每执行一次DATx\_WRITE就会为已储存在内存盒内的日志数据添加一个新的数据记录。

### 步骤

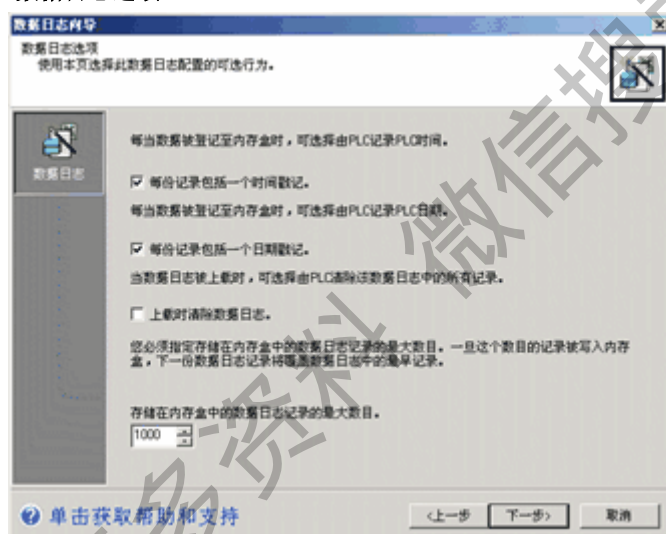
1. 编辑现有配置
2. 数据日志选项
3. 数据日志定义
4. 分配V内存
5. 生成程序代码

#### 第1步 编辑现有配置



单击配置下拉列表，从中选一个现有或新的数据日志定义。您最多可以有四个不同的记录定义。如果不存在现有数据日志配置，数据日志向导将由第2步开始。

### 第2步 数据日志选项



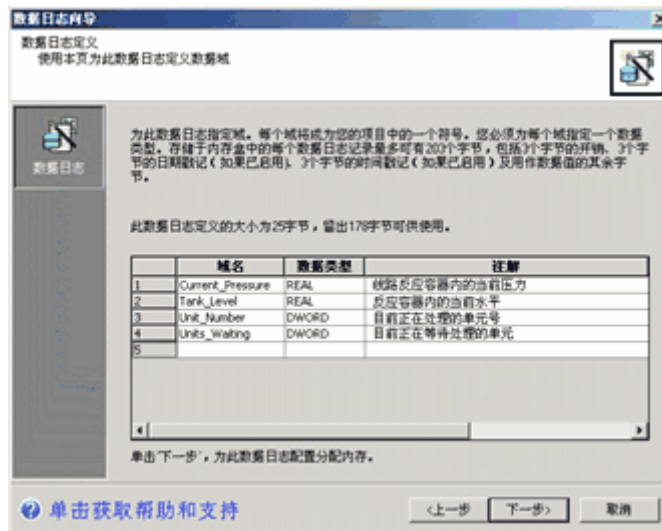
您可以让每份记录包括一个时间戳记。在核取这个选项后，每当用户程序发出数据日志写入命令，CPU就会在每份记录中自动包括一个整数时间戳记。核取此选项会令数据日志记录定义的最大长度缩短3个字节。

每份数据日志记录可添加一个日期戳记。在核取这个选项后，每当用户程序发出数据日志写入命令，CPU就会在每份记录中自动包括一个整数日期戳记。核取此选项会令数据日志记录定义的最大长度缩短3个字节。

您可选择每当数据日志被上载时，清除该数据日志中的全部记录。

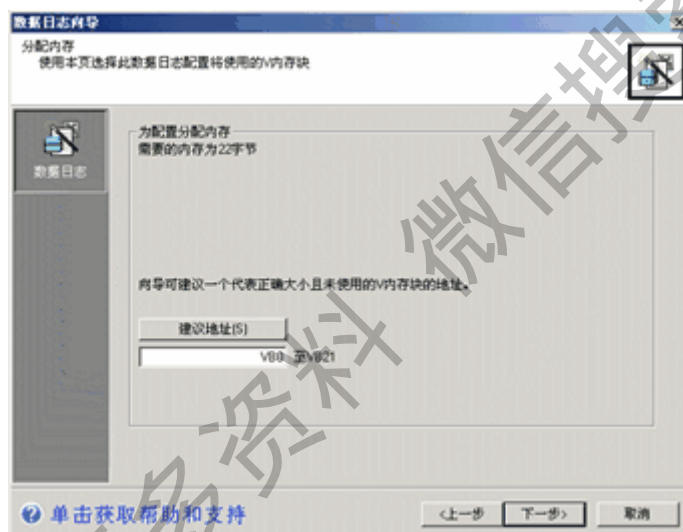
数据日志的形式为循环缓冲区，您必须指定存储在内存盒中数据日志记录的最大数目。此数目为CPU中的一个WORD (字) 值，且必须选取范围 (1-65535)。此选项的默认值为1000。在您下载具有数据日志配置的项目时，内存盒空间会被保留以存储最大数目的记录。

### 第3步 数据日志定义



用鼠标单击以选取一个表数据单元，然后编辑该域。键入一个符号名作为域名。由下拉式列表选取一个数据类型。为

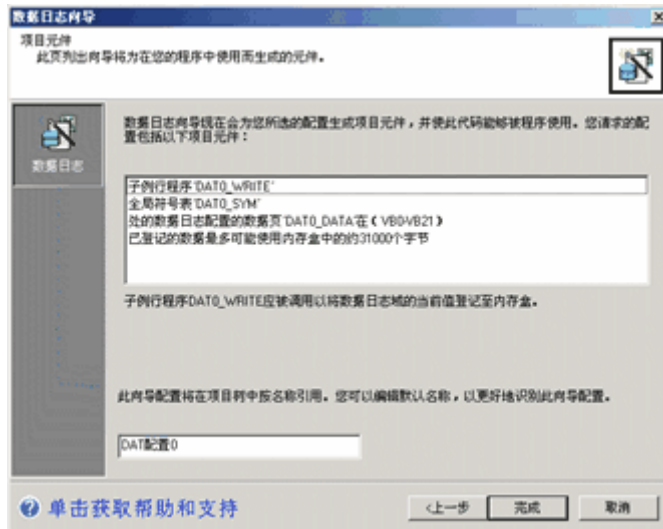
#### 第4步 分配V内存



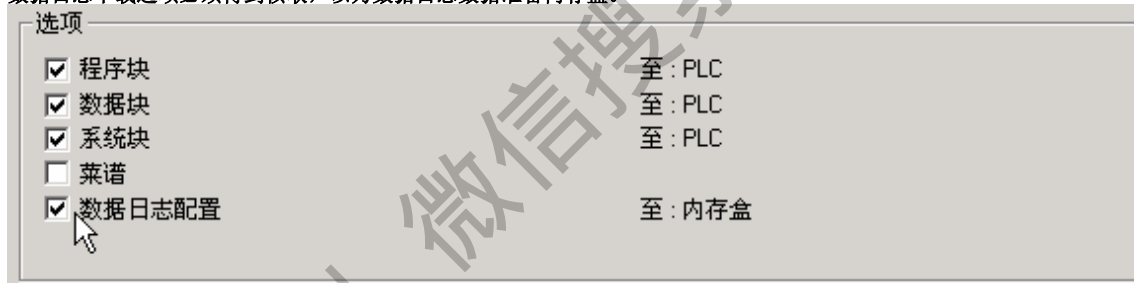
此向导将分配一个“数据块”标记以在V内存中存储数据日志配置。在数据块中指定您希望安放该配置的起始地址。您可以选择V内存地址，如果您希望向导建议正确大小的未使用内存块地址，则单击“建议地址”按钮。配置块的大小根据您在向导中所作的具体选择不同而异。

#### 第5步 生成程序代码





数据日志向导会为您的日志数据定义（符号表、数据块、子例行程序、内存盒数据）生成项目元件，并令您的程序可以使用这些元件。  
 您必须将带有数据日志向导配置的项目下载至S7-200才能加以使用。如果某项目有数据日志配置，则下载窗口的默认设置为“数据日志配置”选项已被核取。  
**数据日志下载选项必须得到核取，以为数据日志数据准备内存盒。**



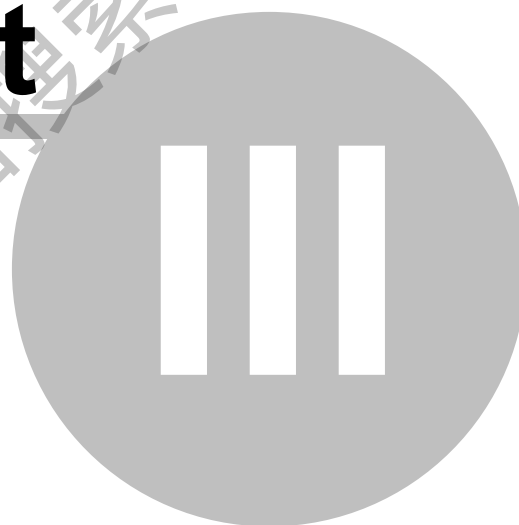
您的程序能够将新的工艺数据移动到由该记录符号名指明V内存地址，执行一DATx\_WRITE子例行程序以取得全部数据值的快照并在内存盒内保存一个新记录，使用S7-200 Explorer从内存盒中将所积累的记录作为CSV（逗号分隔值）文件上载。

另请参阅：  
[DATx\\_WRITE子例行程序](#)

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part



获取更多资料 微信搜索蓝领星球

### 3 LAD、FBD和STL程序编辑器

#### 3.1 阶梯逻辑编程

##### 3.1.1 SIMATIC LAD指令

 指令



##### 3.1.2 IEC 1131-3 LAD指令

 指令



获取更多资料 微信搜索 蓝领星球

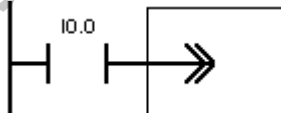


另请参阅：  
[使用IEC超载和非标准指令](#)

### 3.1.3 功率流指示灯

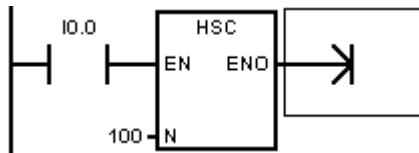
在有效的网络内，功率流从左侧的电源杆流出，通过用户输入的元素，最后到达输出。为了帮助您连接正确的功率流，LAD 网络中会显示功率流指示灯。如果功率流指示灯显示为红色，则表示网络中存在开路状况。您必须解决开路问题，网络才能成功编译。

开路功率流指示器



该元素在线圈输出中显示，表示网络中存在开路状况。您 **必须** 解决开路问题，网络才能成功编译。

供选用功率流指示器

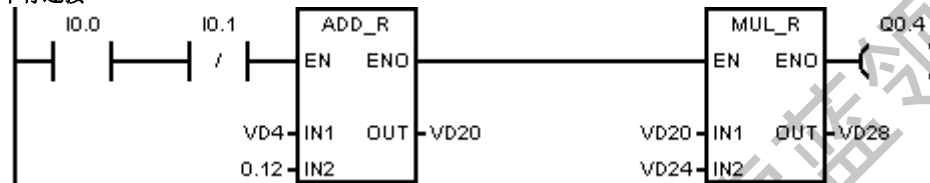


这表示在该点可能有额外逻辑附加在网络上，但并非必须的要求，因为没有该额外逻辑网络也能成功编译。该指示器将在方框元素的ENO功率流输出位置显示，并作为所有空网络的起点。

### 3.1.4 连接指令

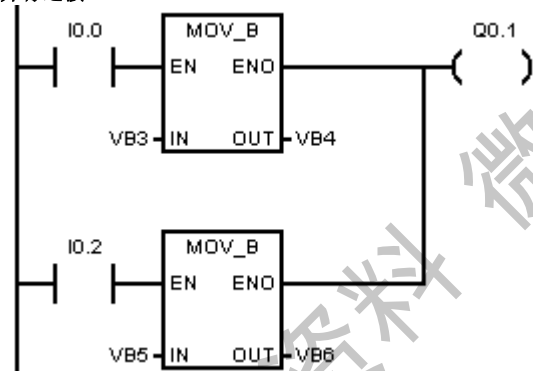
LAD提供水平线和垂直线，帮助以串行或并行方式连接指令引用。通过将一条指令的功率流输出与另一条指令的功率流输入

#### 串行连接



接点和方框可在逻辑系列中连接，线圈可用于终止此类系列。有时需要水平线用于连接一系列指令的功率流。此处使用水平线连接加上实数与乘以实数方框。

#### 并行连接



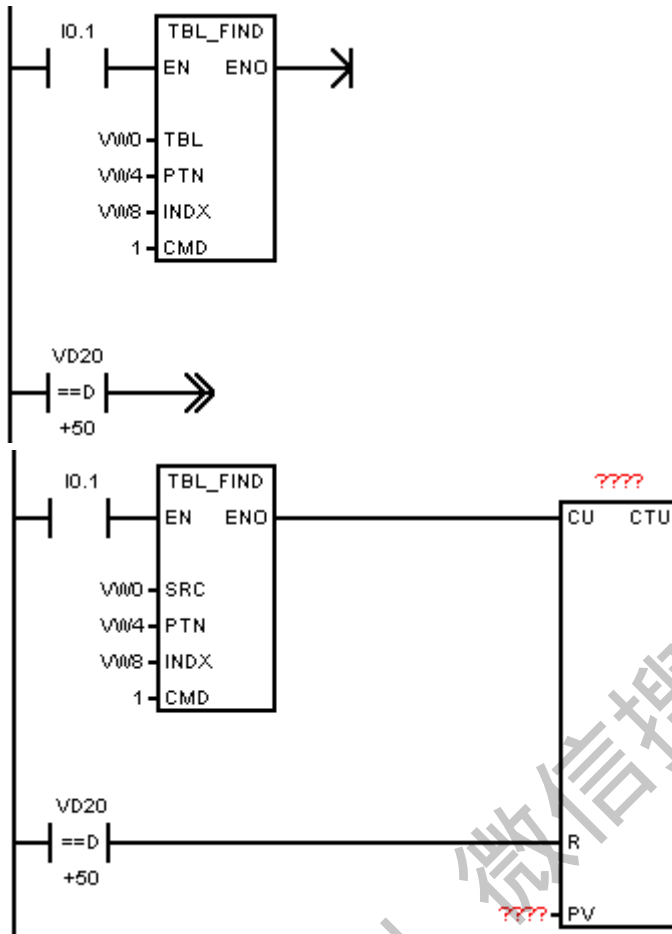
要求并行连接时，需要使用垂直线。垂直线可从原点单元格位置向上和/向下伸展。在上述配置中，“移动字节”指令是并行连接。

#### 自动功率流连接



将元素置于网络中时，阶梯编辑器自动将元素的功率流输入与左侧的任何功率流指示器相连。在上述网络中，LAD将功率流

#### 方框对齐



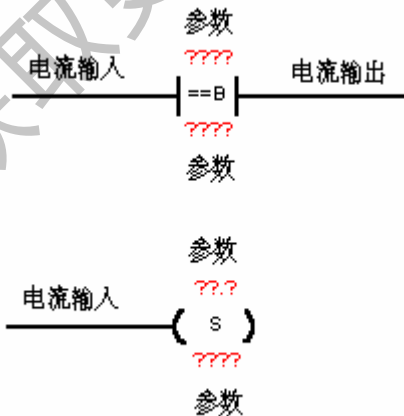
在该例中，“表查找”方框占据网络的许多行。为了使“向上计数”方框的第二个功率流输入与“比较双整数”接点的输出相连，方框被自动扩充。功率流被扩充，将新方框的功率流输入与网络中的功率流指示器相连。

将方框置于网络中时，如果方框包含两个或多个功率流输入，阶梯编辑器会自动将功率流输入与网络功率流指示器对齐。

### 3.1.5 指令参数

LAD共有三种不同的指令类型：接点、线圈和方框。

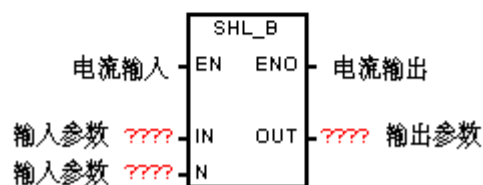
接点和线圈



指令标签在接点或线圈中部显示。功率流显示为从左侧进入和右侧流出。接点有功率流进入，功率流输出，最多可有两个参

数。线圈有功率流输入，最多有两个参数。

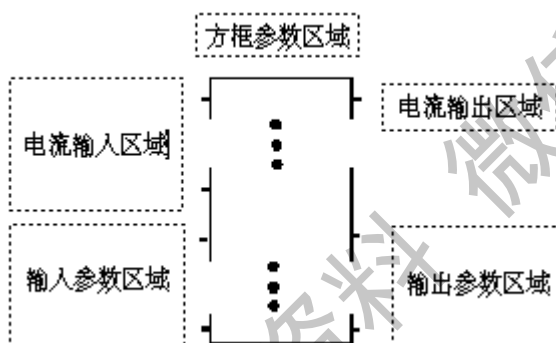
方框



用上述两种格式之一表示方框。指令标签位于指令顶端。

小行显示为从可能连接功率流或指定参数的方框向外突出。参数文本紧靠方框的左侧和右侧，方框顶部参数紧靠方框上端。

方框参数



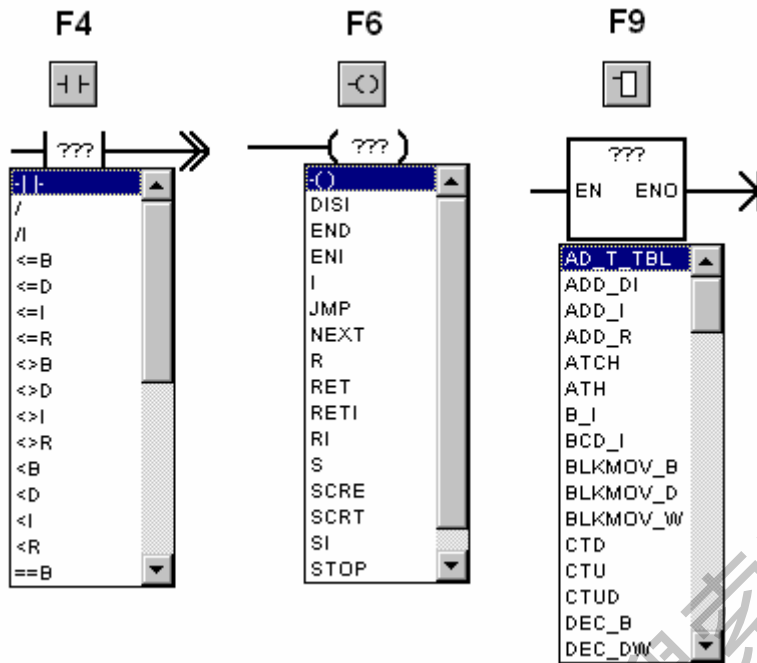
上图描述方框指令引用的五个区域：包括功率流输入、功率流输出、方框顶部参数、输入参数和输出参数。虽然只能有一个方框顶部参数和一个功率流输出，其他三个区域可包含多个功率流或参数。

### 3.1.6 通用指令

类属指令为在LAD中挑选和放置指令提供一种快速的键盘方法。按F4、F6或F9键（或相应的工具条按钮）时，一条类属指令会被放在光标位置，并在下方出现一个列表框。列表框包括一个相同类型全部指令的指令记忆符排序列表。F4键或工具条按钮允许其拥有一组线圈并显示一份线圈指令列表。F6键或工具条按钮放置类属线圈并显示一份线圈指令列表。F9键或工具条按钮放置类属方框并显示一份方框指令列表。

从列表框中选择的指令被放置在网络中，代替类属指令。

按<Escape>键可以延迟指令选择。单击类属指令标签或选择类属指令并按<Enter>时，列表框再次显示。



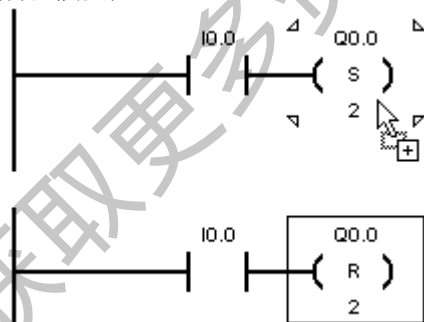
放置指令的全部规则也适用于放置类属指令以及放置从列表框中选择的指令。

### 3.1.7 保留参数

在LAD中，指令可以保留被盖写指令的参数赋值，但必须符合下列条件：

- 编辑器必须位于“重写”模式，指令必须直接放置在现有指令上。
- 指令必须属于相同指令引用类型。
- 对于接点和线圈，指令必须具有相同数目的参数。
- 对于方框，指令必须具有相同数目的功率流输入和输出，并在每个参数区具有相同数目的参数。

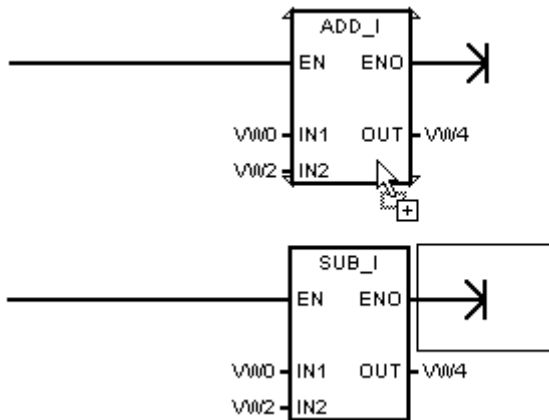
#### 保留线圈参数



在上例中，“重置”线圈保留“设置”线圈的参数赋值，因为两个指令引用均为线圈，并包含两个参数。

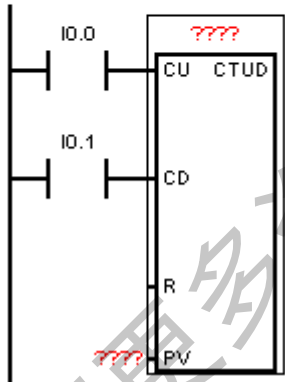
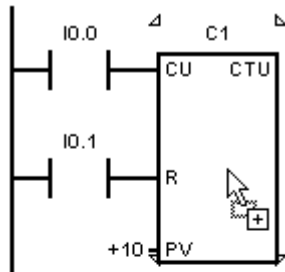
#### 保留方框参数





在该例中，减去整数保留加上整数的参数赋值。每个方框指令均包含一个功率流输入、一个功率流输出、无方框顶部参数、两个输入参数和一个输出参数。

#### 未保留的方框参数



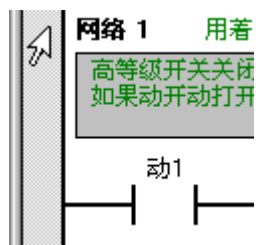
在最后的范例中，“向上 / 向下计数”不保留“向上计数”参数。指令不具有相同数目的功率流输入。

### 3.1.8 编辑指令参数

#### 剪切、复制、粘贴或删除多个网络

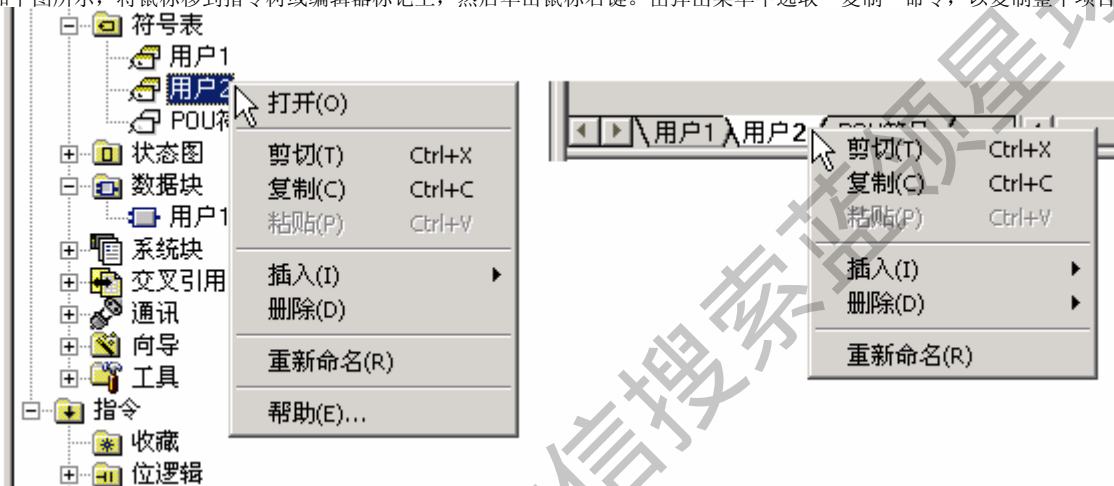
通过拖曳滑鼠或使用SHIFT键和UP（向上）、DOWN（向下）箭头键，您可以选择多个相邻的网络，用于剪切、复制、粘贴或删除选项。使用工具条按钮，从“编辑”菜单选择一条命令，或用滑鼠右键单击，调出编辑选项的弹出菜单。

如下图所示，将鼠标移到编辑器的左侧边缘（装订线区域），然后单击以开始对多个网络进行拖过选择。您不能选择部分网络。如果您尝试选择部分网络，会自动选择整个网络。



### 剪切、复制、粘贴项目元件

如下图所示，将鼠标移到指令树或编辑器标记上，然后单击鼠标右键。由弹出菜单中选取“复制”命令，以复制整个项目元件。



### 注释:

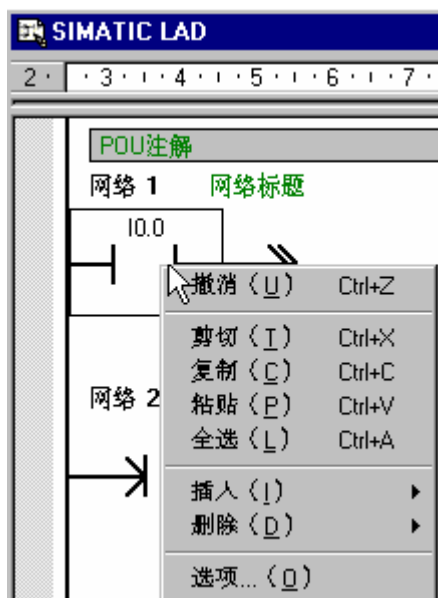
“复制”功能只在您选择了一个需要复制的项目后才能使用。您作出的选择确切描述了将复制到Windows剪贴板缓冲器的内容。

您可以在项目中选择以下目标

- 程序文字或数据域
- LAD、FBD、STL等编辑器中的指令
- 单个网络：将鼠标放在“网络”的左侧边缘上，然后单击
- 多个相邻网络：在选取第一个网络后，使用Shift+下箭头/上箭头或Shift+Page Up/Page Down来扩大或减少所包括的网络数目
- 某POU内的所有网络：请使用 **编辑** >
- **全选** 菜单命令，或用鼠标右键单击某指令树 POU 分支，或用鼠标右键单击某标记名
- 选定的数据块文字，或整个数据块标记页：  
当通过用鼠标右键单击编辑器标记名而复制整个数据块标记页时，随后的粘贴操作能够输出用制表符分隔的数据。举例来说，倘若将此数据块文字粘贴到MS Excel，则电子表格的多个单元行将得到填充。
- 选定的符号表行或列，或整个符号表标记页
- 选定的状态图行或列，或整个状态图标记页

### 编辑单元格、指令、地址和网络

- 当您单击程序编辑器中的空单元格时，会出现一个方框，显示您已经选择的单元格。您可以使用弹出菜单在空单元格
- 单击指令时，会在指令周围出现一个方框，显示您选择的指令。您可以使用弹出菜单在该位置剪切、拷贝或粘贴指令



- 单击指令参数时，会在域周围出现一个方框，显示您选择的参数。您可以使用弹出菜单撤销键入、剪切、拷贝、粘贴



- 当您在网络标题中单击时，您可以编辑项目的标题。您可以使用弹出菜单全选、取消键入的内容、对整个网络执行



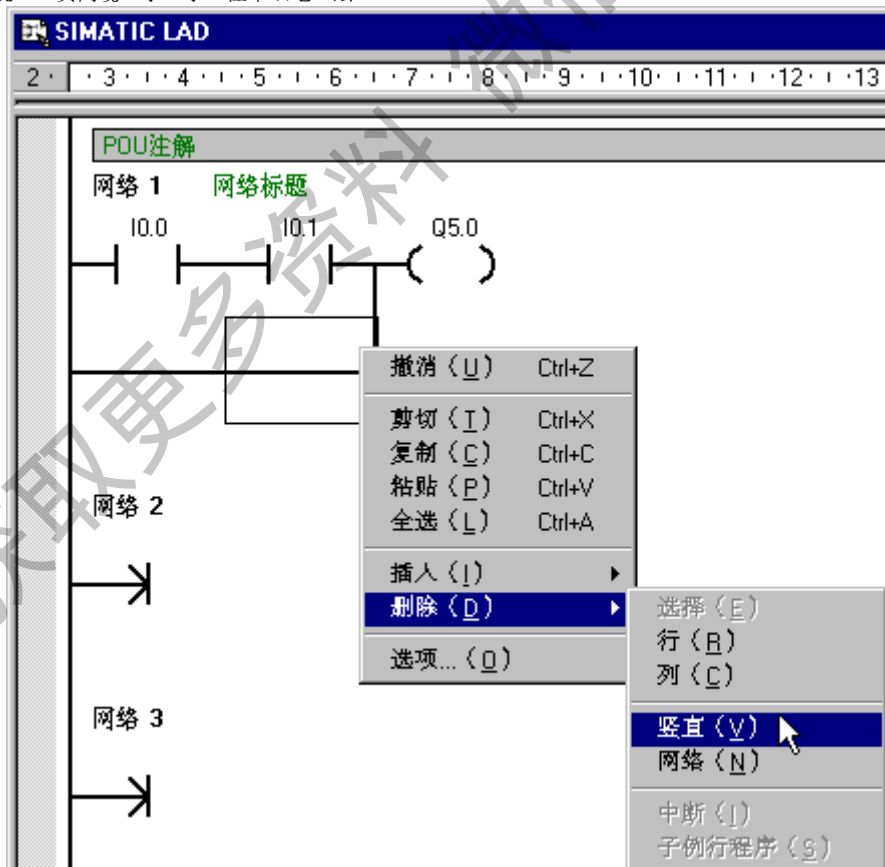
您也可以使用工具条按钮、标准窗口控制键和“编辑”菜单剪切、拷贝或粘贴选项。

#### 将元素移近

您可以剪切和粘贴元素和行，删除行或列；但是，程序编辑器要求元素之间有一定的间距。在某些情况下，您不能把一个元素拷

#### 删除元素

您可以使用DELETE（删除）或BACKSPACE（退格）键删除个别单元格；您可以用“编辑”菜单或滑鼠右键单击的方法调出弹



**注释：** 为了正确选择一条需要删除的垂直线，始终将光标放在垂直线左边的单元格上。

另请参阅：  
[快捷键 \(GS 1.2\)](#)

### 3.1.9 使用符号表帮助组织程序

使用下列方法之一打开符号表（用SIMATIC模式）或全局变量表（用IEC 1131-3模式）：



- 单击浏览条中的“符号表”按钮。
- 选择**检视 (View) > 符号表 (Symbol Table)** 菜单命令。
- 打开**指令树**中的符号表或全局变量文件夹，然后双击一个表格图标。

您可以使用项目中的**系统符号名**。预定义的系统符号表提供常用PLC系统功能的存取功能。PLC系统符号将功能名称与用于激活霉δ芳 PLC特殊内存位置相联系。

本标题讨论下列信息：

[在表格中编辑](#)

[在表格中指定符号赋值](#)

[引号的使用](#)

[插入附加行](#)

[使用定义、编辑或选择符号命令](#)

[按照列排序](#)

[检视重叠和未使用的符号](#)

[使用多个表格](#)

[建立附加符号表](#)

[在符号表之间移动](#)

[POU符号标签](#)

[符号和绝对视图](#)

[在符号和绝对地址视图之间切换](#)

[同时检视符号和绝对地址](#)

[理解符号](#)

[理解符号范围](#)

[使用间接记号](#)

[内存类型](#)

[输入错误和故障排除](#)

[提示](#)

[SIMATIC与IEC](#)

[SIMATIC和IEC表格举例](#)

[在符号表 / 全局变量表中指定符号赋值](#)

[返回顶端](#)

欲为地址或常数值指定符号，请遵循下列步骤：

1. 打开符号表 / 全局变量表（使用标题开始部分描述的一个步骤）。
2. 在“符号名”列键入符号名（例如，Input1）。允许使用的最大符号长度为23个字符。使用TAB、ENTER或ARROW键确认，并移至下一个单元格。

**注释：**

\* 在为符号指定地址或常数值之前，该符号一直显示为未定义符号（绿色波浪下划线）。完成“地址”列赋值后，绿色波浪下划线被删除。

\* 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在LAD、FBD和STL程序编辑器窗口中被一个波浪号(~)截断。您可将鼠标放在被截断的名称上，在工具提示中查看全名。

3. 在“地址”列中键入地址或数值（例如，V或I23）。  
（在IEC 1131-3编程模式中，键入地址后会自动增加正确的IEC“%”前缀）。
4. 如果您正在使用IEC全局变量表，在“数据类型”列的下拉列表中选择一个数据类型。  
（SIMATIC用户无需提供数据类型。）
5. 键入注解（选项：最多允许79个字符）。

**注释：**

在STEP 7-Micro/WIN中，您可以建立多个符号表（SIMATIC编程模式）或多个全局变量表（IEC 1131-



3编程模式)。但不允许将相同的符号名称多次用作全局符号赋值，在单个符号表中中和几个表内均不得如此。（相反，可允许根据您的选择在多个不同局部变量表中多次使用相同的符号名称。）

### 检视重叠和未使用的符号


[返回顶端](#)

欲检视符号表中的“重叠”列或“未使用的符号”列：

- 选择工具 (Tools) > 选项 (Options) 菜单项目。
- 选择“符号表”标记。
- 选择适当的复选框 (“显示重叠符号”和“显示未使用的符号”)。

“重叠”列显示绝对地址共享部分或全部相同内存之符号行的重叠图标 。如果同一个字面值有多个已定义的符号常量，那麽该图标将显示在重叠图标  下方。每次表格被修改时，“重叠”列被更新。

符号表			
· 3 · 1 · 4 · 1 · 5 · 1 · 6 · 1 · 7 · 1 · 8 · 1			
		符号	地址
1		符号3	VBO
2		符号4	VDO

“未使用的符号”列就在您的程序中未被引用的所有符号显示未用图标 。每次表格被修改时，该列被更新。

符号表			
· 3 · 1 · 4 · 1 · 5 · 1 · 6 · 1 · 7 · 1 · 8 · 1			
		符号	地址
1		符号1	I0.0
2		符号2	I0.1

### 引号的使用

[返回顶端](#)

在3.1版之前，全局符号名称必须放在双引号内。双引号已不再用于符号名。自4.0版起，双引号用于在符号表中起始和结束作为撤 琶 持档册ASCII常量字符串。此赋值创建了一个符号ASCII字符串常量。与此相似，单引号用于起始和结束ASCII字符常量（只限字节、字、双字）以创建一个符号ASCII字符常量。

### 插入附加行

[返回顶端](#)

使用下列方法之一在符号表 / 全局变量表中插入附加行：

- 选择菜单命令编辑 (Edit) > 插入 (Insert) > 行 (Row)。将在符号表 / 全局变量表光标的当前位置上方插入新行。
- 用鼠标右键单击符号表 / 全局变量表中的一个单元格。选择弹出菜单命令插入 (Insert) > 行 (Row)。将在光标的当前位置上方插入新行。
- 欲在符号表底部插入新行，将光标放在最后一行的任意一个单元格中，按“下箭头”键。

### 使用定义、编辑或选择符号命令

[返回顶端](#)

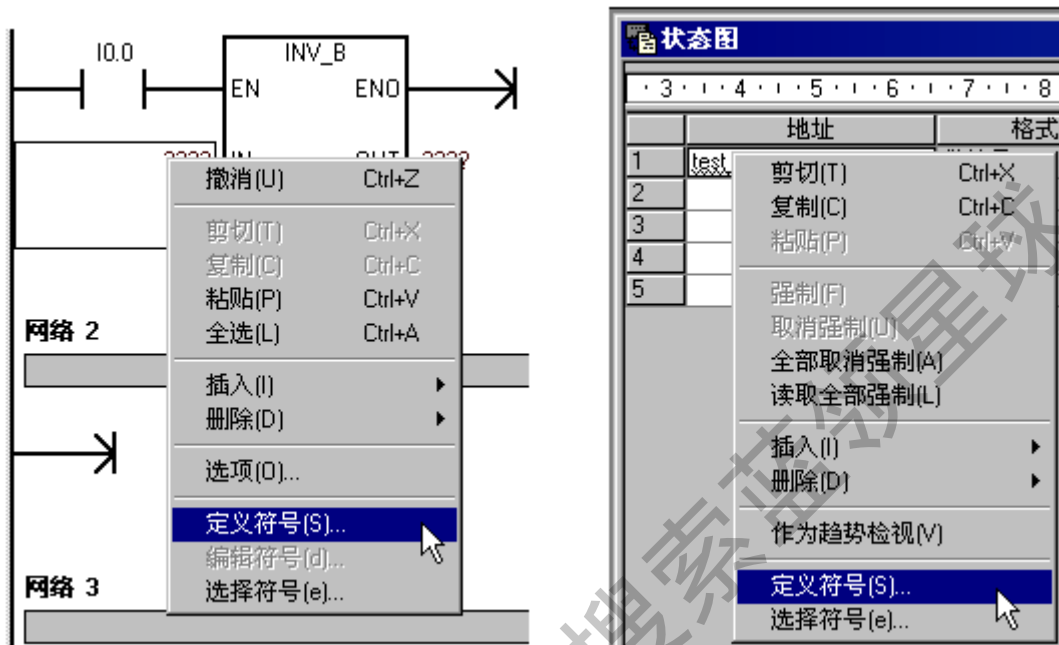
定义、编辑或选择符号等命令，允许您在使用程序编辑器或状态图时，定义新符号、从列表上选取现有符号、或编辑符号属性。新的或修改后的赋值将被自动加入到符号表内。



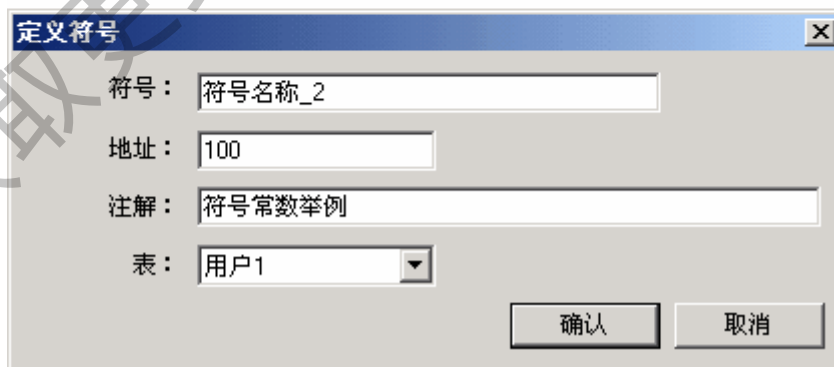
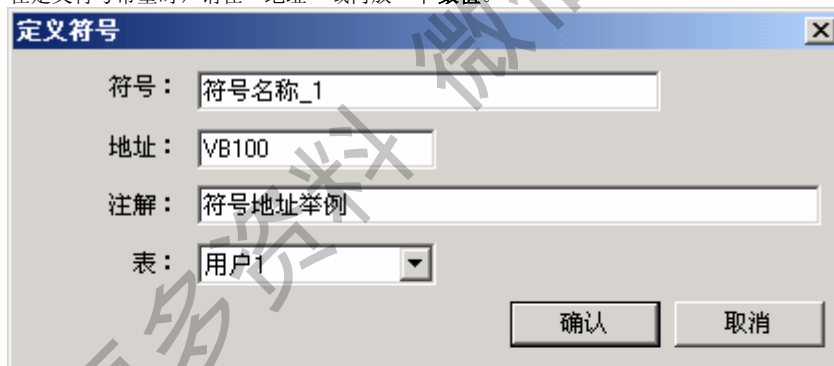
**提示：**只有在“符号编址”视图中工作时才能使用符号命令。（检查“检视”菜单下方的“符号编址”选项：标选符号表示已打开“符号编址”视图。）

欲激活“符号”命令：

- 用鼠标右键单击对某指令参数或“状态图”地址单元来说不完全（或不正确）的符号名，然后从弹出菜单选择“定义符号(S)...”。



- 定义新符号，编辑现有符号的属性，或由全部用户指定符号的概括中选取一个现有符号。  
在定义符号地址时，请在“地址”域内放一个地址。  
在定义符号常量时，请在“地址”域内放一个数值。



bml mw\_sym\_define\_b.gif}

- 单击“确定”确认作业，并关闭对话框，或单击“取消”，取消选择。

注释:



- 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在 LAD、FBD 和 STL 程序编辑器窗口中被一个波浪号 (~) 截断。您可将鼠标放在被截断的名称上，从而在工具提示中查看全名。

### 按照列排序

[返回顶端](#)

您可按照“名称”或“地址”列排序表格，排序时可按正向或逆向（字母）顺序排列。此表中的符号常量将在与地址关联的最后一个符号后面列出。所有非字符类型的符号常量接着将独立于显示格式地按数字顺序排列。ASCII 常量将出现在数字常量之后，地址 A 拷 置 螺 旋 钻

您可以使用检视菜单对某列进行排序。请单击列身以选取该列，请勿单击列顶的列标题（“名称”或“地址”）。

- 若按正向顺序（A 至 Z）排序列，单击“排序”按钮，或选择菜单命令**检视 (View) > 正向排序 (Sort Ascending)**。
- 若按逆向（Z 至 A）顺序排序列，单击“逆向排序”按钮，或选择菜单命令**检视 (View) > 逆向排序 (Sort Descending)**。

若要按相反顺序排序，单击列顶部的列头（“名称”或“地址”）。当您单击列头时，将在正向和逆向排序之间切换。


### 建立附加符号表


[返回顶端](#)

默认情况下，符号表窗口为用户定义的符号名称（USR1）显示一个标记。如果您对任何 POU 重新命名，符号表窗口显示一个被称为“POU 符号”的标记—如果您在使用“协议”功能，例如 USS 协议库，则窗口显示“USS 协议”标记。

可采用下列几种方法为用户定义的符号名称建立附加符号表。

- 从指令树用鼠标右键单击“符号表”文件夹，并选择弹出菜单命令**插入符号表 (Insert Symbol Table)**。
- 打开符号表窗口，并使用“编辑”菜单，或用鼠标右键单击，调出弹出菜单，并选择**插入 (Insert) > 表格 (Table)**。
- 您在建立程序网络时，可指定符号名，方法是在程序编辑器的指令参数域中键入有效的符号名。此种方法建议一组不


付 注 册 返 回 母 琶 5 七 整 散 (0) 锤 持 捣 疝 恻 卑 磁   
，将该组符号名传送到您指定地址的新“符号表”标记。欲为符号表重新命名，用鼠标右键单击指令树中的符号表标记，并选择“重新命名”，或双击“符号表”标记名。

**注释：**成功地插入新符号表后，一个新标记  会出现在符号表窗口的底部。在检视符号表时，请核实选择了正确的标记。您可以双击或用鼠标右键单击标记，为标记重新命名。

### 在符号表之间移动

[返回顶端](#)

如果建立了多个符号表，您可以使用下列任意一种方法存取表格：

- 如果已经打开符号表窗口，单击位于符号表窗口底部的标记 ，在符号表之间移动。
- 从指令树扩充“符号表”文件夹图标，并双击所需的“符号表”图标，检视符号表窗口中该表的标记。

### POU 符号标记

[返回顶端](#)

如果您已经为项目的任何 POU 或其他元件（例如数据块、状态图或符号表）指定符号名，将在符号表窗口的“POU 符号”标记中列出符号名赋值。

该标记为只读标记；您不能在此编辑赋值。

如果您希望更改赋值，您必须编辑该元件的“属性”对话框。（从指令树的“项目”分支，用鼠标右键单击元件，调出“属性

倍 曰 翻 颞 #  
所有符号赋值必须使用有效语法。（请参阅下面的 [输入错误](#)）。如果指定符号名赋值时违反了指南规定，编译程序时 STEP 7-Micro/WIN 会报告错误。

### 在符号和绝对地址视图之间切换

[返回顶端](#)



在符号表 / 全局变量表中建立符号和绝对地址或常量值的关联后，您可在操作数信息的符号和绝对显示之间切换。可采用下列

- 选择菜单命令 **检视 (View) > 符号编址 (Symbolic Addressing)**，在符号编址打开或关闭之间切换
- 使用 **Ctrl+Y** 快捷键在符号编址打开或关闭之间切换

“符号编址”菜单项目前面的标选标记表示已打开符号编址。默认条件下，当您打开第一个项目时，符号编址也被打开。

您不能在检视符号常量或其关联的字面值之间切换。这是因为您可以为同一个字面值定义多个符号常量。因此，假如您可以 ‘  
毛侧 儒 A 浚瑛TEP 7-  
Micro/WIN无法可靠地恢复原有的符号常量。出于此原因，禁用符号编址（通过主菜单或按CTRL+Y）将不会影响您项目中符号常量的显示。出于同样原因，如果您为某操作数输入了该常量的字面值，STEP 7-Micro/WIN将不会自动套用已定义的符号常量。

### 同时检视符号和绝对地址

[返回顶端](#)

欲在LAD、FBD或STL程序中同时检视符号地址和绝对地址，使用菜单命令 **工具 (Tools) > 选项 (Options)**，并选择“程序编辑器”标记。选择“显示符号和地址”。



- 提示：**
- 只有在打开 **符号地址视图** 时，才能在项目中显示符号地址。否则，即使您选择了“显示符号和地址”，也只会显示绝对地址。
  - 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在 LAD、FBD 和 STL 程序编辑器窗口中被一个波浪号 (~) 截断。您可将鼠标放在被截断的名称上，在工具提示中查看全名。

### 理解符号范围

[返回顶端](#)

符号表 / 全局变量表将符号名指定给 PLC 内存或 I/O 地址。在符号表 / 全局变量表中定义 **符号** 时，符号具有全局范围。这意味着您可在任何 **POU** 中使用该符号名称，代表位于该符号地址的数据。（相反，如果您使用局部变量表指定符号名称，则只能在定义该局部变量的 POU 范围中使用该局部变量。）

### 使用间接记号

[返回顶端](#)

与绝对地址相同，您可在符号中使用间接记号 (& 和 \*)。欲了解有关间接编址的详情，请参阅 [直接和间接编址](#)。

### 内存类型

[返回顶端](#)

您可为下列内存类型建立符号名：I、Q、M、SM、AI、AQ、V、S、C、T、HC

### 输入错误

[返回顶端](#)

输入错误用色彩显示，很容易发现：

- 非法语法 — **红色文字** 和 **红色波浪下划线**

例如：

	符号	地址
1	2name	I0.0
2		VBO
3	Begin	I0.2

（非法地址 VBO 是正确的）（逆转的关键字无法用作符号）

（第一个字符的号码字符非法）

- 非法使用 — **红色波浪下划线**

例如：

	符号	地址
1	Pump1	I0.0
2	Pump1	I0.0
3	SymConstant	123456
4	SymConstant	123456
5	SymConstant2	654321
6	SymConstant3	654321

(重复名称) (重复地址)

- 未定义符号 — 绿色波浪下划线

例如:

	符号	地址
1	<u>Pump1</u>	
2	Pump2	V0
3	Factor	3.14E-006
4	Neg_Value	-3333333

(名称带有无效地址) SIMATIC模式

(名称带有无效地址或数据类型) IEC模式

**注释:**

- 符号名称可包含某些字母数字字符和下划线，以及某些扩展字符（ASCII 128至ASCII 255）。但第一个字符只能是字母或扩展字符。
- 在3.1版之前，全局符号名称必须放在双引号中。双引号已不再用于符号名。自4.0版起，双引号用于在符号表中起始和结束作为某符号名之赋值的ASCII常量字符串。此赋值创建了一个符号ASCII字符串常量。与此相似，单引号用于起始和结束ASCII字符常量（仅限字节、字、双字）以创建一个符号ASCII字符常量。
- 不允许使用关键字作为符号名，也不允许第一个字符是数字或包含非字母数字字符或扩展字符集。
- 符号名称的最大许可长度为23个字符。
- 如果您希尝试建立符号赋值，以纠正程序中操作数“未定义符号”错误，但键入地址数值后忘记按TAB键、ENTER键  
蚣 芳 蹦 禱 爻 繇 蚩 唳 魔 保 僮 魔 廊 幌 允 尽 拔 炊 丁 宸 拧 贝 砦 蟆 N 私 饩 稣 庖 晃 侍 猓 禱 胤 疔 棠 翱 家 獾 支  
旁 淞 暗 司 贰 钡 卮 襪 校 TAB、ENTER或箭头键，完成地址赋值。

**建立符号提示**[返回顶端](#)

- 您可在任何时间指定符号赋值：在建立程序逻辑之前、之间或之后。
- 您可在“注解”列中增加注释，记录您的符号。
- 编程时，可使用“定义、编辑或选择符号”命令从列表选择新符号名或建立新符号名。
- 通过单击不同的表列，可对符号表/全局变量表中的数值进行检索。
- 通过拖拉边缘，可重定列尺寸。
- 您可使用文件(File) > 打印(Print)命令，并选择适当的复选框，打印符号表/全局变量表。
- 您一旦建立符号赋值，就可以通过启用符号信息表功能，在LAD/FBD/STL程序编辑器中的每隔网络滞后显示地址赋值和符号注解。

**SIMATIC和IEC表举例**[返回顶端](#)

符号表（SIMATIC编程模式）和全局变量表（IEC编程模式）在功能上的唯一区别是“数据类型”列，此为IEC编程的独特功能。

IEC-

1131是一种加强类型语言。指令操作数被定义为接受一种指定的数据类型。欲了解有关数据类型的详情，请参阅 [SIMATIC和IEC 1131-3数据类型](#)。

SIMATIC符号表举例:

	符号	地址	注解
1	动1	I0.0	
2	动2	I0.1	
3	停止1	I0.2	
4	停止2	I0.3	
5	高等级	I0.4	
6	低等级	I0.5	
7	复位	I0.7	
8			
9	泵1	Q0.0	
10	泵2	Q0.1	
11	搅拌机电机	Q0.2	
12	蒸阀	Q0.3	
13	排泄阀	Q0.4	
14	排泄泵	Q0.5	
15			
16	达到高等级	M0.1	
17	搅拌机计时	T37	
18	循环计数	C30	
19	Mix_Time_Minutes	15.3	
20	Mix_Temp_Cent_min	90	
21	Mix_Temp_Cent_max	95	
22	Process_ID	"P418369B"	

IEC全局变量表举例：

全局变量表					
	符号	地址	数据类型	注解	
1	Start_1	%I0.0	BOOL		
2	Start_2	%I0.1	BOOL		
3	Stop_1	%I0.2	BOOL		
4	Stop_2	%I0.3	BOOL		
5	High_Level	%I0.4	BOOL		
6	Low_Level	%I0.5	BOOL		
7	Reset	%I0.7	BOOL		
8					
9	Pump_1	%Q0.0	BOOL		
10	Pump_2	%Q0.1	BOOL		
11	Mixer_Motor	%Q0.2	BOOL		
12	Steam_Valve	%Q0.3	BOOL		
13	Drain_Valve	%Q0.4	BOOL		
14	Drain_Pump	%Q0.5	BOOL		
15					
16	High_Lev_Reached	%M0.0	BOOL		
17	Mix_Timer	%T37	TON		
18	Cycle_Counter	%C30	CTU		
19	Counter_Out	%M0.2	BOOL		
20	Counter_Current	%M\W3	INT		
21	Mix_Time_Minutes	15.3	REAL		
22	Mix_Temp_Cent_min	9	BYTE		
23	Mix_Temp_Cent_max	95	BYTE		
24	Process_ID	"P418369B"	STRING		

[返回顶端](#)

另请参阅:

[局部变量表](#)

[检视符号编址](#)

[网络符号信息表 \(LAD/FBD\)](#)

[编址概述](#)

[直接和间接编址](#)



[关键字列表](#)

[系统符号表](#)

[符号表标记 \(工具 > 选项\)](#)

### 3.1.10 为每个网络显示符号信息表

使用下列方法之一检视或隐藏LAD/FBD/STL程序编辑器窗口中的符号信息表:

- 单击  工具条按钮。
- 选择菜单命令 **检视 (View) > 符号信息表 (Symbol Information Table)**。
- 按 **Ctrl+T** 快捷键组合。
- 单击 **在项目中应用全部符号**  工具条按钮。应用全部符号命令不仅用所有的新、旧和修改符号名更新项目，而且在符号信息表打开和关闭之间切换。

您一旦选择了 **检视 (View) > 元件 (Component) > 符号表 (Symbol Table)** 菜单项目，所有打开的项目均会显示网络符号信息，直至您取消该菜单项目。

当您检视符号信息表时，符号名、绝对地址、数据类型和注解会在 LAD/FBD/STL 程序中每个网络下方按字母顺序显示。未定义

的符号名也包括在内。

该表列出该网络中所有符号的信息；不包含全局符号的网络不显示符号信息表。所有重复条目均被移除。

如果您位于IEC模式，表格显示一个附加数据类型列。

符号信息表无法编辑。

**网络 1 用涂料成分1和监视器罐填充储罐。**

起始\_1:I0.0    停止\_1:I0.2    高\_位:I0.4    泵\_1:Q0.0

泵\_1:Q0.0

泵_1	Q0.0	泵1控制 (1 = 运行, 0 = 停止)
高_位	I0.4	混合罐料位过高
起始_1	I0.0	泵1起始命令
停止_1	I0.2	泵1停止命令

**网络 2 用涂料成分2和监视器罐填充储罐。**

起始\_2:I0.1    停止\_2:I0.3    高\_位:I0.4    泵\_2:Q0.1

泵\_2:Q0.1

泵_2	Q0.1	泵2控制 (1 = 运行, 0 = 停止)
高_位	I0.4	混合罐料位过高
起始_2	I0.1	泵2起始命令
停止_2	I0.3	泵2停止命令




**提示：**用光标键浏览符号信息。

您可以从列边缘拖动分割器光标，重新确定表格大小。表格不得超过 32 个单元格。所有

无法放入一个单元格中的文本均以波浪号结束；工具提示显示所有无法容纳的文本。

打印程序时，使用“打印”对话框中的“选项”按钮启用符号信息表打印。

为了节省垂直方向的空间并检视更多的程序网络，您可以在符号信息表打开和关闭之间切换，方法是使用**检视 (View) >**

**元件 (Component) > 符号表 (Symbol Table)** 命令，或单击在项目中应用全部符号  工具条按钮。

使用“查找”功能替换文本。

#### 故障排除

如果在LAD/FBD/STL程序网络的下方未显示符号信息表，请考虑下列因素：

- 是否已启用符号信息表？检查“检视”菜单：符号信息表项目旁应当有一个标选符号。

- 网络中是否使用符号（或与符号对应的绝对地址）？不包含符号数值的网络不会显示符号信息表。
- 是否在符号表（IEC全局变量表）中输入了符号？在您完成赋值之前，不会显示地址或注解信息。

另请参阅：

[符号表 / 全局变量表](#)

[符号表标记（工具 > 选项）](#)（在您检视程序状态时，调整符号信息表的显示方法）

[应用程序用户参考手册](#)

[入门指南目录](#)

### 3.1.11 SIMATIC和IEC 1131-3数据类型

在STEP 7-Micro/WIN中编程时，在下列两种情况下您必须理解并使用数据类型：

- 如果您以IEC 1131-3模式编程，则必须为全局符号表中的所有数值选择数据类型。
- 如果您在局部变量表中赋值，则必须为每个局部变量定义数据类型。

当您明确地为数值指定数据类型时，您为STEP 7-

Micro/WIN提供清晰的指令，指明需要为数值分配多少内存空间（例如，数值100可被存储为字节、字或双字）以及如何表示数值（将0视作布尔值还是数值？）。

每个SIMATIC和IEC 1131-

3指令或参数化子例行程序均由精确定义识别。该定义可被称为签名。对于所有的标准指令，每个指令操作数允许使用的数据类型集均从签名获得。对于参数化子例行程序，用户通过局部变量表建立子例行程序签名。

STEP 7-Micro/WIN V3.0编程软件为SIMATIC模式执行简单的数据类型检查，为IEC 1131-

3模式执行细致的数据类型检查。这意味着为局部或全局变量指定一种数据类型时，软件在下表所列的每个等级中确保操作数数

基本数据类型	数据类型大小	说明	范围
布尔	1位	布尔	0至1
字节	8位	不带符号的字节	0至255
字节	8位	带符号的字节（SIMATIC模式仅限于SHRB指令）	-
128至+127			
字	16位	不带符号的整数	0至65,535
整数	16位	带符号的整数	-32768至+32767
双字	32位	不带符号的双整数	0至4294967295
双整数	32位	带符号的双整数	-2147483648至 +2147483647
实数	32位	IEEE 32 位浮点	+1.175495E-38至 +3.402823E+38 -1.175495E-38至
3.402823E+38			
字符串	2至255字节	ASCII字符串照原样存储在PLC内存中，形式为1字符串长度接ASCII数据字节	
	ASCII字符代码128至255		

复杂数据类型	说明	范围
TON	打开延迟计时器	1 ms 10 ms 100 ms T32, T96 T33 - T36, T97 - T100T37 - T63, T101 -
T255		
TOF	关闭延迟计时器	1 ms 10 msms T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 -
255		
TP	脉冲计时器（参阅注释1）	1 ms 10 msms T32, T96 T33 - T36, T97 - T100T37 -
T63, T101 - 255		
CTU	向上计数器	至255
CTD	向下计数器	至255
CTUD	向上 / 向下计数器	至255
SR	设置主要双稳态	无关
RS	重置主要双稳态	无关

#### STEP 7-Micro/WIN 提供三级数据类型检查

1. **IEC编辑器具有细致数据类型检查。**在该模式中，参数数据类型必须与符号或变量的数据类型完全匹配。欲支持细致数据类型检查，只有被指定数据类型类型的变量才能成功编译。使用细致数据类型检查时，数据类型为整数的变量对字指令参数无效。细致数据类型检查仅在IEC 1131-3模式中进行。

**用户选择的数据类型**                      **等同数据类型**

布尔	布尔
字节	字节
字	字
整数	整数
双字	双字
双整数	双整数
实数	实数
字符串	字符串

2. **处于SIMATIC模式中的局部变量表具有简单数据类型检查。** 在该模式中，当符号或变量被指定一个数据类型时，同时自动指定与所选数据类型位大小相匹配的所有数据类型。例如，如果用户选择双整数作为数据类型，局部变量会自动指定数据类型双字，因为二者均为32位数据类型。不自动指定实数数据类型，虽然实数也是32位数据类型。实数数据类型被定义为无其他等同数据类型，总是独特的类型。使用局部变量时，只在SIMATIC模式中执行简单数据类型检查。

**用户选择的数据类型**                      **等同数据类型**

布尔	布尔
字节	字节
字	字、整数
整数	字、整数
双字	双字、双整数
双整数	双字、双整数
实数	实数
字符串	字符串

3. **SIMATIC符号编辑器无数据类型检查。** 该模式仅限用于无法选择数据类型的SIMATIC全局变量。在该模式中，所有相同大小的数据类型被自动指定给符号。例如，编程软件将自动为赋予VD100地址的符号指定下列数据类型：双字、双整数和实数。

**为SIMATIC全局符号由大小决定的数据类型****用户选择的地址**                      **指定等同的数据类型**

V0.0	布尔
VB0	字节、字符串
VW0	字、整数
VDO	双字、双整数、实数

**数据类型检查的益处**

数据类型检查的益处是帮助用户避免常见编程错误。例如，如果指令支持带符号的数字，则在使用不带符号的数字作为指令操作数时，软件为该用法设置旗标会有益处。例如，关系比较<I是带符号的指令，显然，对于带符号的数据类型操作数，-1小于0。但当<I指令允许支持不带符号的数据类型时，程序员需确保绝不会发生下列情况。在程序运行时，对于<I指令，不带符号的40,000实际上小于0。程序员需确保对带符号的指令使用不带符号的数字不会超越正负数界限，否则会出现无法预测的结果。



**警告** 您应当确保带符号的数字用作不带符号的指令时不得超出正负数界限。若无法确保不带符号的数字用作带符号的指令时不超出正负数界限，可能致使程序或控制器操作出现无法预测的结果。无法预测的控制器操作可能导致人员伤亡或严重伤害及/或财产严重损坏。请务必核实不带符号的数字用作带符号的指令时不得超出正负数界限。

总之，在IEC 1131-3编辑模式中，细致数据类型检查通过对指令的非法数据类型生成错误，帮助程序员在编译过程中发现这些错误。SIMATIC编辑器不具有此一功能。

**在SIMATIC与IEC1131-3之间移植程序**

因为IEC1131-3属于细致数据类型，而SIMATIC不属于此一类型，STEP 7-Micro/WIN不提供在两种不同编辑模式之间移动程序的能力。程序员必须选择一种希望使用的编辑模式。

**超载指令**

超载指令支持一系列数据类型。仍然应用细致数据类型检查，在成功编译指令之前所有的操作数数据类型必须匹配。例如，请参阅下表中用于IEC超载ADD（加）指令。

<b>指令</b>	<b>允许使用的数据类型（细致数据类型检查）</b>	<b>允许使用的数据类型（数据</b>
-----------	----------------------------	---------------------

## 类型检查)

## 编译指令

ADD	整数	字、整数	ADD_I (加整数)
ADD	双整数	双字、双整数	ADD_D (加双整数)
ADD	实数	实数	ADD_R (加实数)

当所有操作数的数据类型均为双整数时，编译器生成“加双整数”指令。如果超载指令数据类型混合，会出现编译错误。根据数据类型检查的等级决定哪些为非法。在下例中，如果使用细致数据类型检查，则生成编译程序错误，但如果使用简单数据类型检查则成功编译。

ADD IN1 = INT, IN2 = WORD, IN3 = INT.

细致数据类型检查：引起编译错误。

简单数据类型检查：成功编译至ADD\_I (加整数)。

简单数据类型检查不会阻止出现常见运行时间编程错误。例如，使用简单数据类型检查时，编译程序不能帮助避免在运行时间(5) 韵鲁 < 暗檀碧媳 ADD 40000,1被解释为负数，而不是不带符号的40,001。

## 在IEC中为超载指令使用直接编址

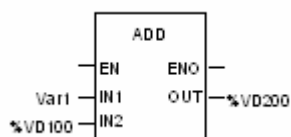
## IEC-

1131编程模式也允许使用直接表示的内存位置，作为指令参数配置的一部分。参数中可使用变量和内存位置。但是，使用直接表示的内存位置时必须清楚地理解数据类型的隐含意义，因为这些位置不包含明确的类型信息。另外，不能从任何超载IEC指

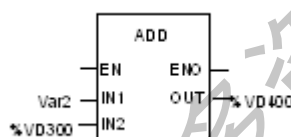
直接表示的参数数据类型通过检查指令中包含的其他类型的参数来决定。当指令参数配置使用某一指定类型的变量时，则假定

名称	地址	数据类型	注解
Var1		实数	这是浮点数变量。
Var2		双整数	这是双整数变量。
Var3		整数	这是整数变量。

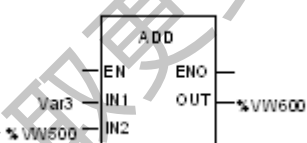
## 举例 说明



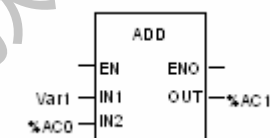
假定VD100和VD200为实数类型，因为Var1是实数类型。



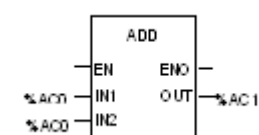
VD300和VD400为双整数类型，因为Var2是双整数类型。



假定VW500和VW600为整数类型，因为Var3是整数类型。



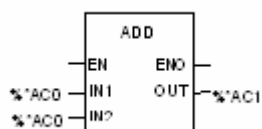
假定AC0和AC1为实数类型，因为Var1是实数类型。



这种配置被视作非法，因为无法确定类型。累加器中的数据类型可为双整数或



实数。



数或实数。

这种配置被视作非法，因为无法确定类型。累加器指针中的数据类型可为双整

### 使用数据类型转换指令

转换指令允许从一种数据类型移至另一种数据类型。STEP 7-Micro/WIN支持在下列简单数据类型之间的移动数值转换指令。

转换指令	IEC细致数据类型检查 允许的操作数	SIMATIC局部变量表中的简单数据类型检查
字节至整数	入： 字节 出： 整数	入： 字节 出： 字、整数
整数至字节	入： 整数 出： 字节	入： 字、整数 出： 字节
整数至双整数	入： 整数 出： 双整数	入： 字、整数 出： 双字、双整数
双整数至整数	入： 双整数 出： 整数	入： 双字、双整数 出： 字、整数
双整数至实数	入： 双整数 出： 实数	入： 双字、双整数 出： 实数
实数至双整数（进位）	入： 实数 出： 双整数	入： 实数 出： 双字、双整数
字符串至整数	入： 字符串 出： 双字（*指向字符串的指针） 出： 整数	入： 字符串，
双字（*指向字符串的指针）	出： 字、整数	
整数至字符串	入： 整数 出： 用双字代表的字符串（*指向字符串的指针）	入：
字、整数	出： 用双字代表的字符串（*指向字符串的指针）	
字符串至双整数	入： 字符串 出： 双字（*指向字符串的指针） 出： 双整数	入： 字符串，
双字（*指向字符串的指针）	出： 双字、双整数	
双整数至字符串	入： 双整数 出： 用双字代表的字符串（*指向字符串的指针）	入：
双字、双整数	出： 用双字代表的字符串（*指向字符串的指针）	
字符串至实数	入： 字符串 出： 双字（*指向字符串的指针） 出： 实数	入： 字符串
双字（*指向字符串的指针）	出： 实数	
实数至字符串	入： 实数 出： 用双字代表的字符串（*指向字符串的指针）	入： 实数
出： 用双字代表的字符串（*指向字符串的指针）		

在IEC 1131-

3编辑模式中，您可以使用超载移动指令在整数和字、双整数和双字之间转换。超载移动指令允许相同大小的数据类型自由移动

簧 杀喊氩砒蟆

IEC 1131-3超载移动	入	出
移动（整数至字）	整数	字
移动（字至整数）	字	整数
移动（双整数至双字）	双整数	双字
移动（双字至双整数）	双字	双整数

### 3.1.13 理解S7-200计时器指令

您可利用计时器执行时基计数功能。S7-200指令集提供三种不同类型的计时器。

- 打开延迟计时器（TON），用于单间隔计时
- 保留性打开延迟计时器（TONR），用于累计一定数量的定时间隔
- 关闭延迟计时器（TOF），用于延长时间以超过关闭（或假条件），例如马达关闭后使马达冷却。

计时器操作：

计时器类型	当前值 >= 预设值	启用输入“打开”	启用输入“关闭”
电源循环 / 首次扫描			
TON	计时器位打开，当前值继续计数 直至达到32,767		当前值记录时间
当前值=0	计时器位关闭，当前值=0		计时器位关闭，
TONR	计时器位打开，当前值继续计数 直至达到32,767		当前值记录时间
值保持最后的状态	计时器位关闭，可保持当前值(1)		计时器位及当前
TOF	计时器位关闭，当前值=预设值，停止计数		计时器位打开，当前值=0

从“打开”转换为“关闭”后计时器开始计数

计时器位关闭，当前值=0

(1) 可通过电源循环为保留目的选择保留性计时器当前值。请参阅 [保留范围标记—系统块配置](#) 中有关S7-300 CPU的内存保留信息。

请参阅文档光盘“提示和技巧”中的提示31，查阅使用打开延迟计时器（TON）的抽样程序。

#### 注释：

可用“重设”（R）指令重设任何计时器。“重设”指令执行下列操作：  
计时器位 = 关闭，计时器当前值 = 0

只能用“重设”指令重设TONR计时器。

重设后，TOF计时器要求启用输入从“打开”转换为“关闭”，以便重新启动。

#### 1毫秒分辨率

1毫秒计时器记录自现用1毫秒计时器启用以来1毫秒计时器间隔的数目。执行计时器指令即开始计时；但是，1毫秒计时器每毫秒更新一次（计时器位及计时器当前值），不与扫描循环同步。换言之，在超过1毫秒的扫描过程中，计时器位和计时器当前值将保持不变。  
计时器指令用于打开和重设计时器，如果是TONR计时器，则用于关闭计时器。  
因为可在1毫秒内的任意时刻启动计时器，预设值必须设为比最小要求计时器间隔大一个时间间隔。例如，使用1毫秒计时器时，为了保证时间间隔至少为56毫秒，则预设时间值应设为57。

#### 10毫秒分辨率

10毫秒计时器记录自现用10毫秒计时器启用以来10毫秒计时器间隔的数目。执行计时器指令即开始计时；但是，在每次扫描循环的开始更新10毫秒计时器，其方法是以当前值加上积累的10毫秒间隔的数目（自前一次扫描开始算起）（换言之，在整个扫描过程中，计时器当前值及计时器位保持不变）  
因为可在10毫秒内的任意时刻开始计时器，预设值必须设为比最小要求计时器间隔大一个时间间隔。例如，使用10毫秒计时器时，为了保证时间间隔至少为140毫秒，则预设时间值应设为15。

#### 100毫秒分辨率

100毫秒计时器记录自现用100毫秒计时器上一次更新以来100毫秒计时器间隔的数目。这种计时器的更新方法是在执行计时器指令时更新计数器的数目（自前一次扫描开始算起）。  
只有在执行计时器指令时才对100毫秒计时器的当前值进行更新。因此，如果启用了100毫秒计时器但并未对各扫描循环执行计时器指令，则仍不能更新计时器当前值并将丧失时间。同样，如果在单个扫描循环内多次执行100毫秒计时器指令，将向计时器的计数器增加100毫秒间隔数，赢得时间。只有在每次扫描循环仅执行一次计时器指令时，才应该使用100毫秒计时器。  
因为可在100毫秒内的任意时刻启动计时器，预设值必须设为比最小要求计时器间隔大一个时间间隔。例如，使用100毫秒计时器时，为了保证时间间隔至少为2100毫秒，则预设时间值应设为22。

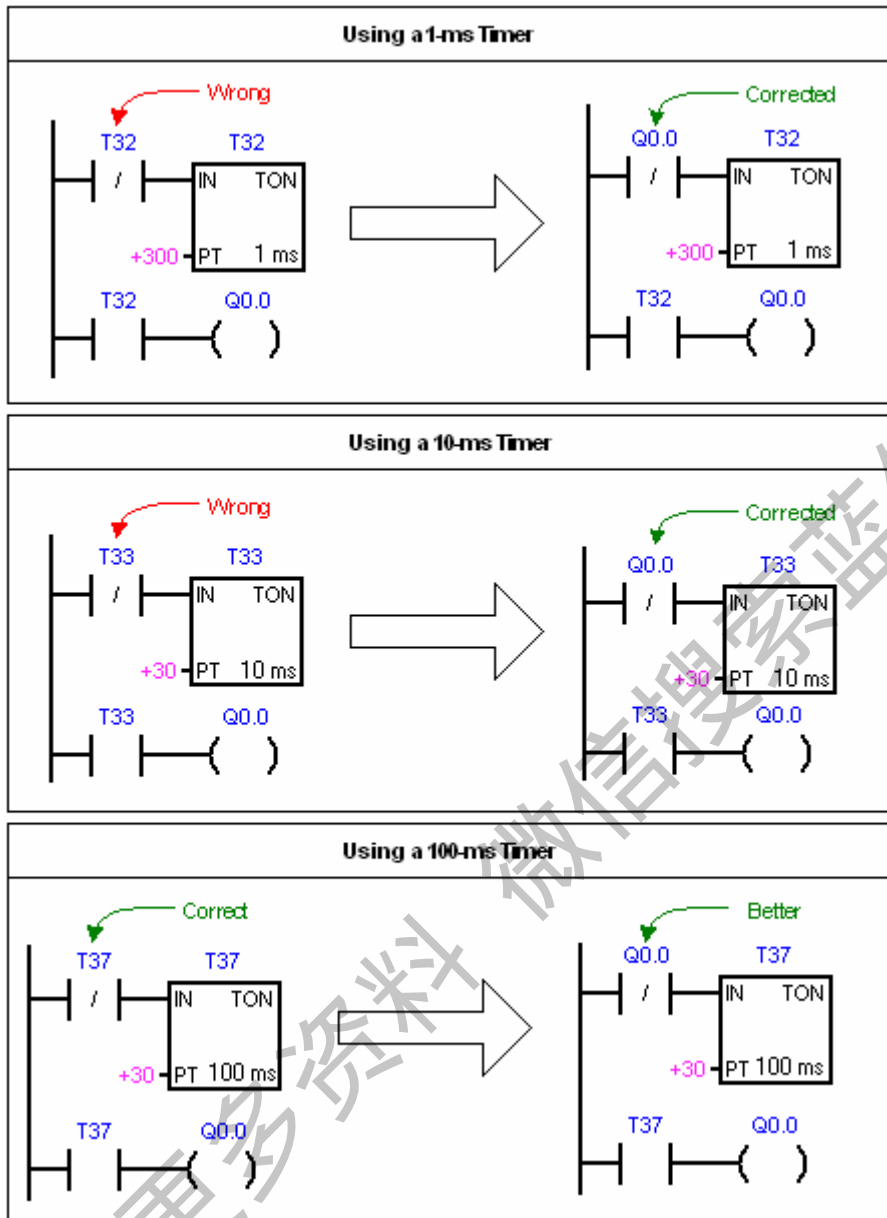
#### 更新计时器当前值

更新当前时间值有多种方式，其作用取决于如何使用计时器，如下图所示的计时器操作：

- 使用1毫秒计时器时，每次在执行正常关闭接点T32之后和执行正常打开接点T32之前更新计时器当前值时，Q0.0即打开进行一次扫描。
- 使用10毫秒计时器时，从不打开Q0.0，因为从扫描顶端至执行计时器方框均打开计时器位T33。一旦执行了计时器方框，计时器的当前值及T位均被设为零。执行正常打开接点T33时，T33及Q0.0均被关闭。
- 使用100毫秒计时器时，每当计时器当前值达到预设数值时，Q0.0时钟打开进行一次扫描。

通过使用正常关闭接点Q0.0代替计时器位作为计时器方框的启用输入，可保证每次计时器达到预设值时均打开输出Q0.0进行一次扫描。

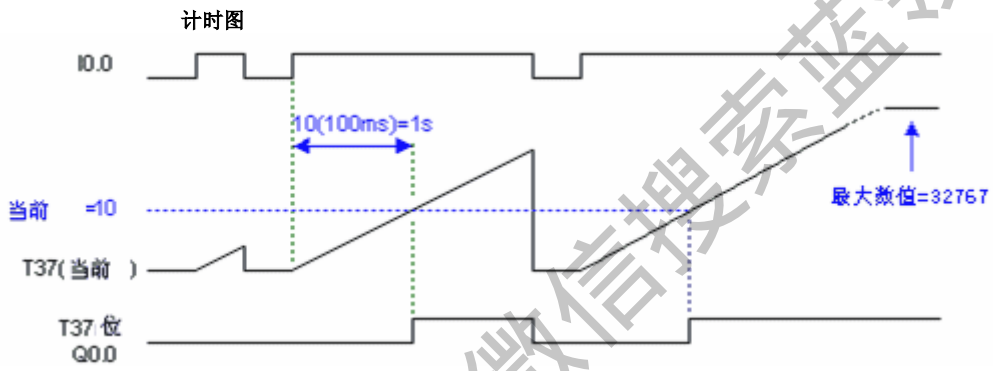
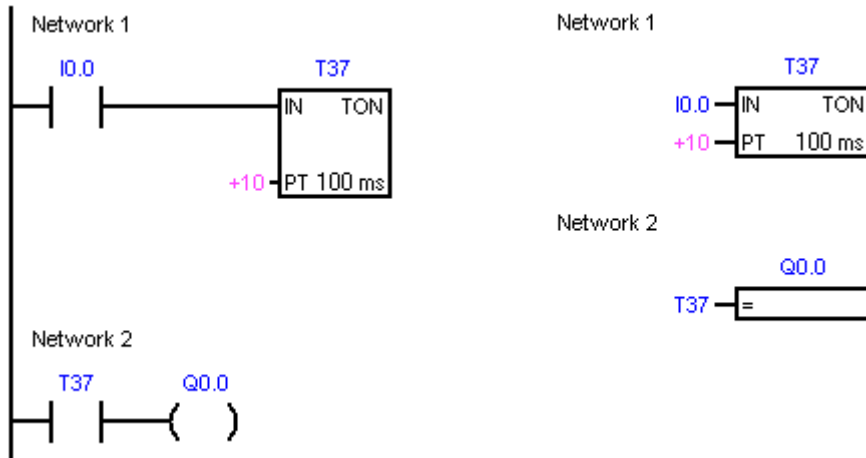
#### 自动重新触发—击计时器举例



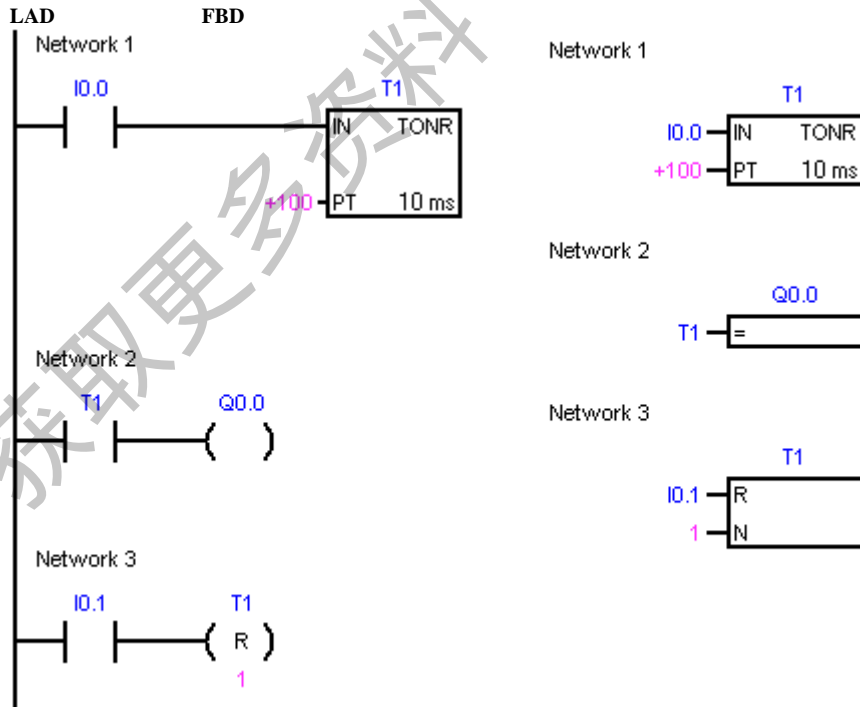
打开延迟计时器举例

LAD

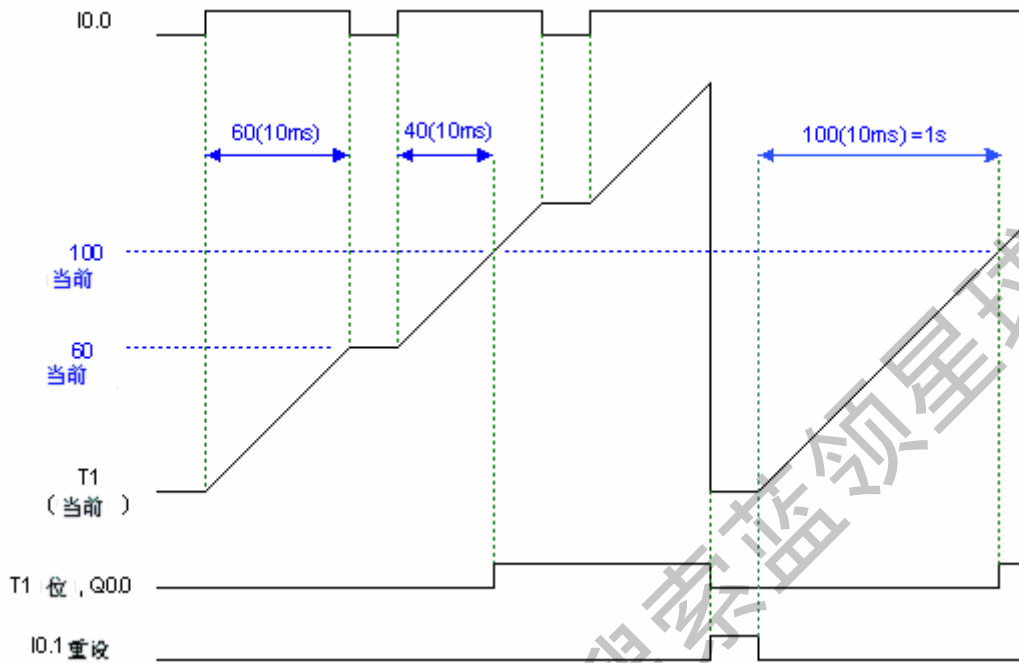
FBD



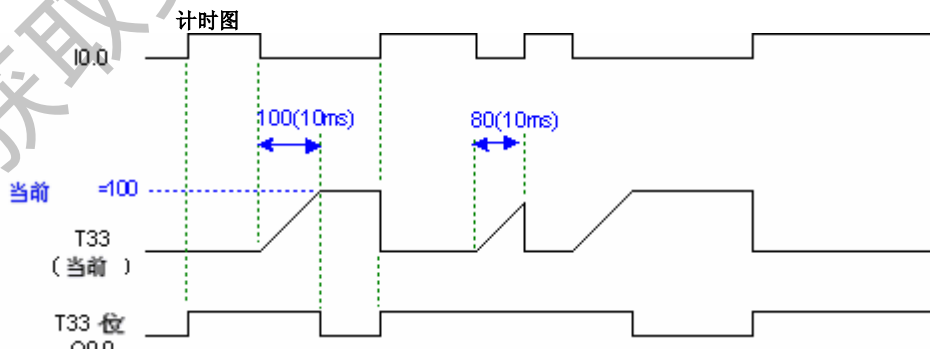
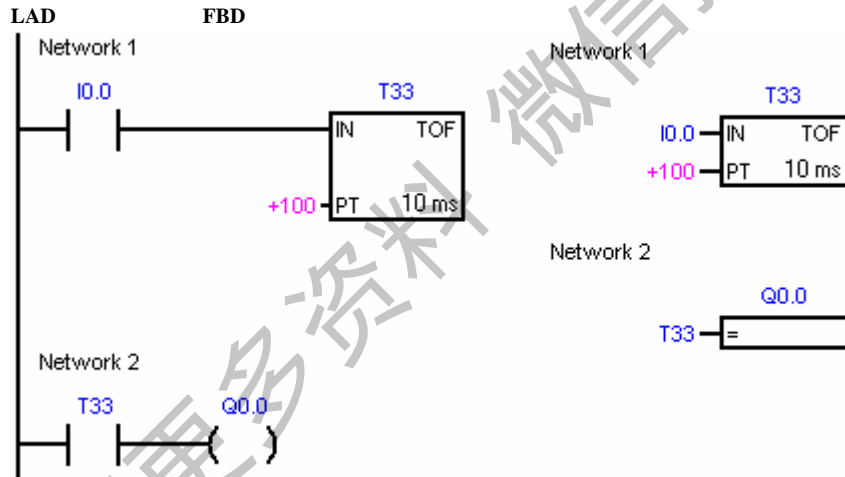
保留性打开延迟计时器举例



计时图



关闭延迟计时器举例



### 3.1.14 理解S7-200计数器指令

每次向上计数输入执行从关闭至打开转换时，向上计数（CTU）从该计数器的当前值向上计数。重设输入打开或执行重设指令时，计数器被重设。达到最大值（32,767）时，计数器停止。

每次向上计数输入执行从关闭至打开转换时，向上/向下计数器（CTUD）向上计数，每次向下计数输入执行从关闭至打开转换时，向下计数。达到最大值（32,767）时，向上计数输入的下一个上升边缘导致当前计数变成最小值（32,768）。与此相似，达到最小值（-32,768）时，向下计数输入的下一个上升边缘导致当前计数变成最大值（32,767）。

向上和向上/向下计数器有一个保持当前计数的当前值。计数器还有一个预设值（PV），每次执行计数器指令时，将预设值与当前值进行比较。如果当前值大于或等于预设值，C位关闭。否则，C位打开。

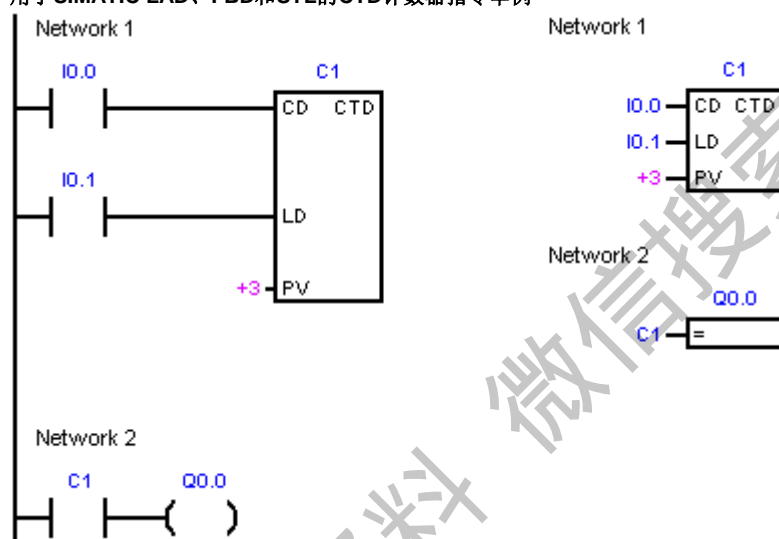
每次向下计数输入执行从关闭至打开转换时，向下计数器（CTD）从该计数器的当前值向下计数。载入输入打开时，计数器重设计数器位，并将预设值载入当前值。达到零时，计数器停止，计数器位（C位）打开。

当您使用重设指令重设计数器时，计数器位被重设，计数器当前值被设为零。使用计数器号码引用该计数器的当前值和C位。

#### 注释：

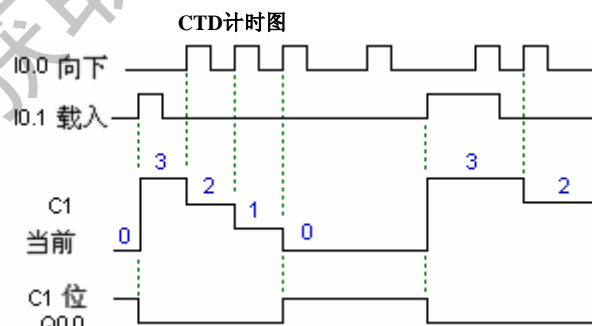
因为每台计数器有一个当前值，请勿将相同的号码指定给一台以上计数器。（向上计数器、向上/向下计数器和向下计数器存取相同的当前值。）

#### 用于SIMATIC LAD、FBD和STL的CTD计数器指令举例

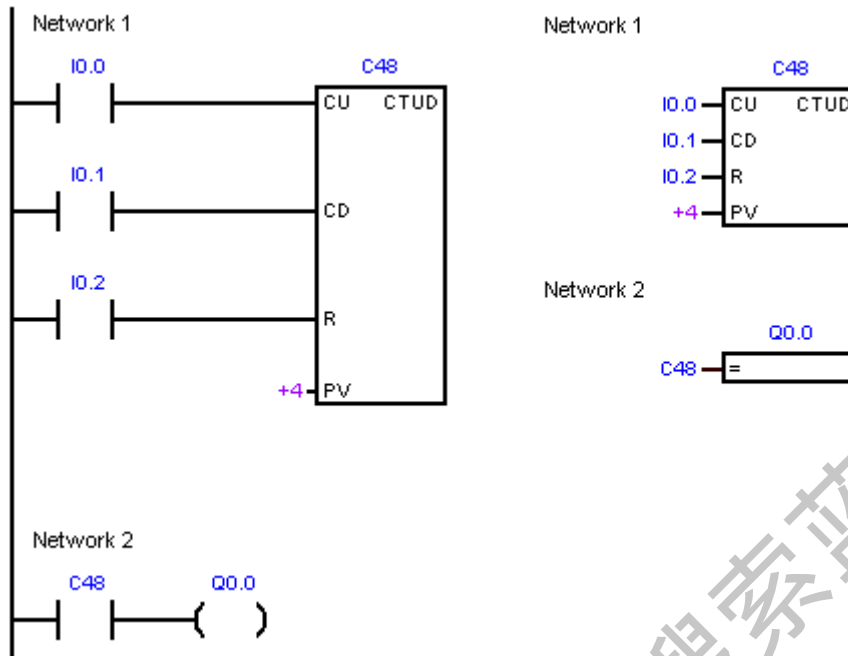


NETWORK 1 // 向下计数器C1当前值从3计数至0，  
// 其中IO.1关闭，IO.0"关闭"打开"递减C1当前值"  
// IO.1"打开"载入向下计数预设值3  
LD IO.0  
LD IO.1  
CTD C1 +3

NETWORK 2 // 当计数器C1当前值 = 0时，C1位"打开"  
LD C1  
= Q0.0



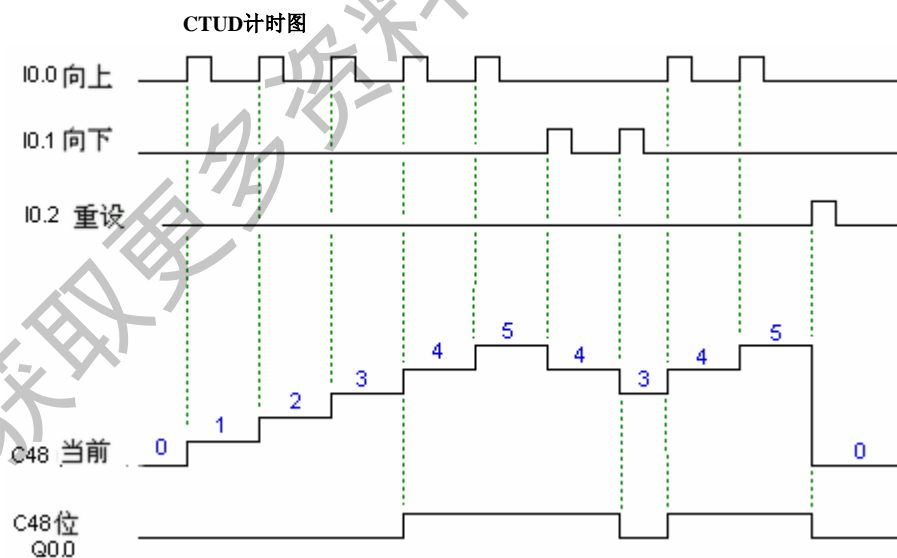
用于SIMATIC LAD、FBD和STL的CTUD计数器指令举例



```

NETWORK 1 // I0.0向上计数 - I0.1向下计数 - I0.2将当前值重设为0
LD I0.0
LD I0.1
LD I0.2
CTUD C48 +4

NETWORK 2 // 当前值>= 4时, 向上/向下计数计数器C48打开C48位
LD C48
= Q0.0
  
```



### 3.1.15 理解高速计数器指令

高速计数器对CPU扫描速率无法控制的高速事件进行计数，最多可配置12种不同的操作模式。高速计数器的最高计数频率取决于您的CPU类型。

每台计数器对支持此类功能的时钟、方向控制、重设和起始均有专用输入。对于二相计数器，两个时钟均可以最高速率运行。在正交模式中，可选择1乘以（1x）或4乘以（4x）最高计数速率。所有计数器均以最高速率运行，互不干扰。

本标题讨论下列主题：

[使用高速计数器](#)

[理解高速计数器的详细计时功能](#)

[为高速计数器连接输入线](#)

[高速计数器编址（HC）](#)

[理解不同的高速计数器](#)

[选择现用状态和1x/4x模式](#)

[高速计数器初始化顺序](#)

[控制字节](#)

[HSC模式](#)

[设置当前值和预设值](#)

[状态字节](#)

[为中断赋值](#)

## 使用高速计数器

[返回顶端](#)

通常高速计数器被用作鼓式计数器驱动器，以恒速旋转的转轴配有递增转轴编码器。转轴编码器提供每次旋转的指定计数以及每次旋转一个重设脉冲。转轴编码器的时钟和重设脉冲为高速计数器提供输入。

用最先的几个预设值载入高速计数器，并在当前计数小于当前预设值的期间内激活所需输出。当前计数等于预设值或重设时，计数器设置提供中断。

每次发生当前计数值等于预设值中断事件时，载入新预设值，并设置下一个输出状态。发生重设中断事件时，设置第一个预设值和第一个输出状态，并重复该循环。

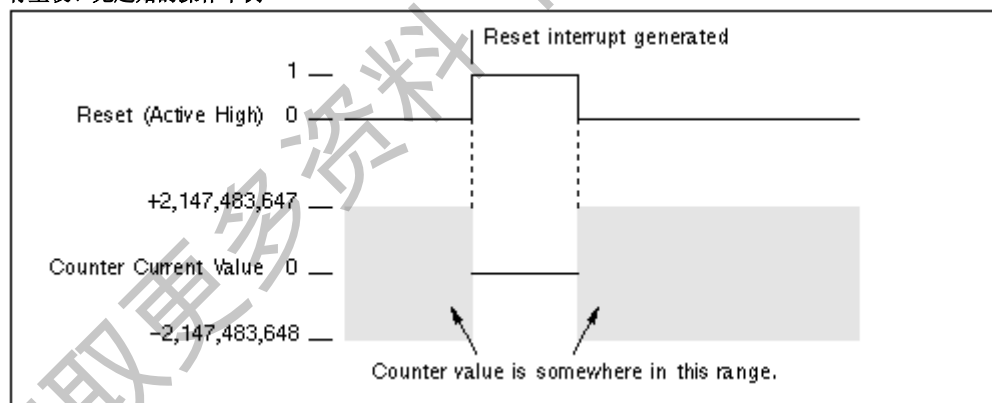
因为中断的发生速率远远低于高速计数器的计数速率，可对高速操作执行精确的控制，并对整体 PLC 扫描循环产生相对较小的影响。

## 理解高速计数器的详细计时功能

[返回顶端](#)

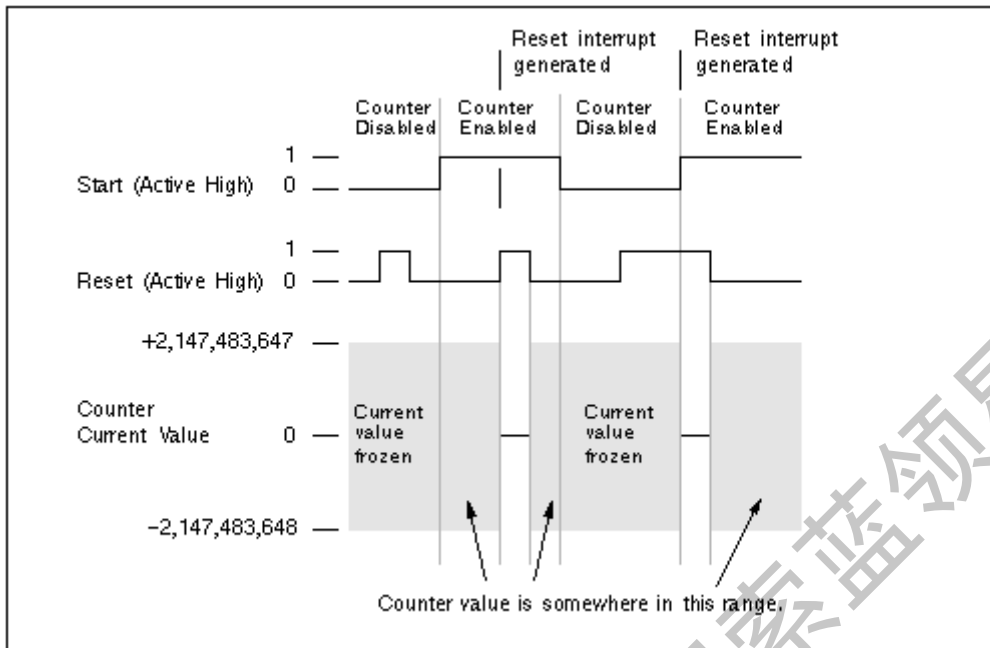
下列计时图显示根据模式分类的每台计数器的功能。在另一个计时图中显示重设和起始输入操作，并应用于所有使用重设和起始输入的模式。在重设和起始输入图中，重设和起始的现用状态均被编程为高级。

### 有重设、无起始的操作举例

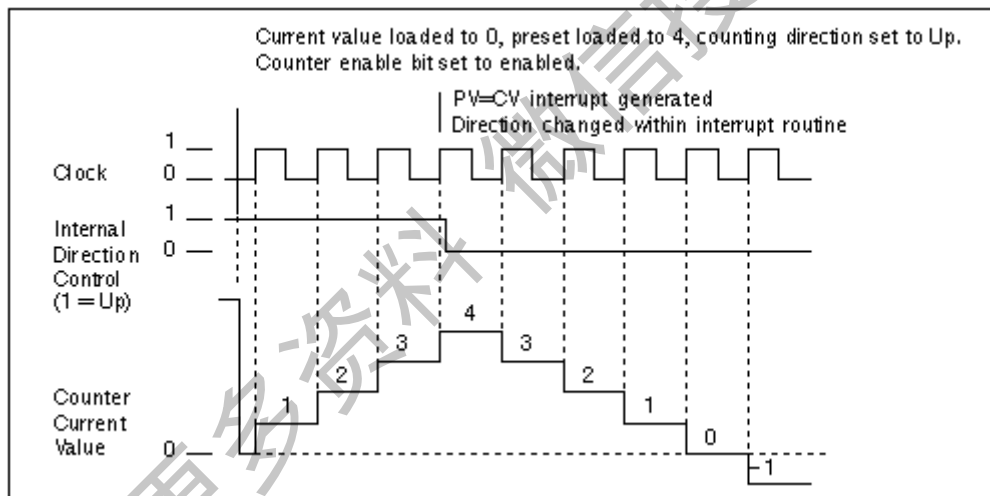


### 有重设和起始的操作举例

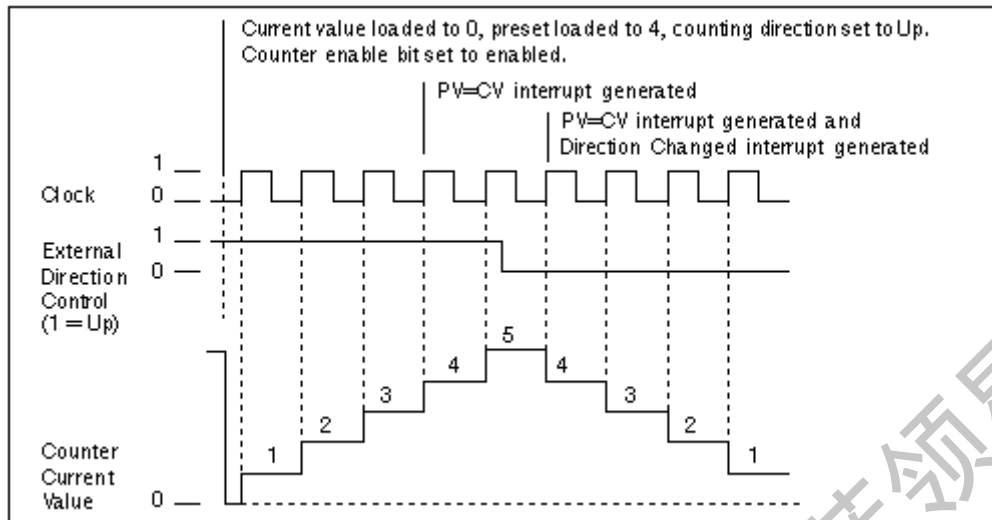




模式0、1和2操作举例

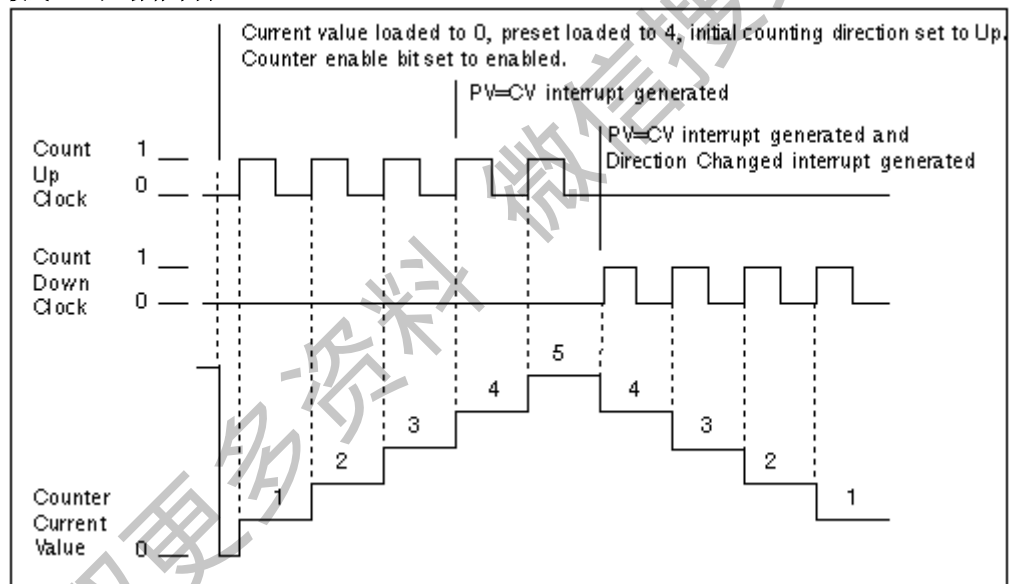


模式3、4和5操作举例

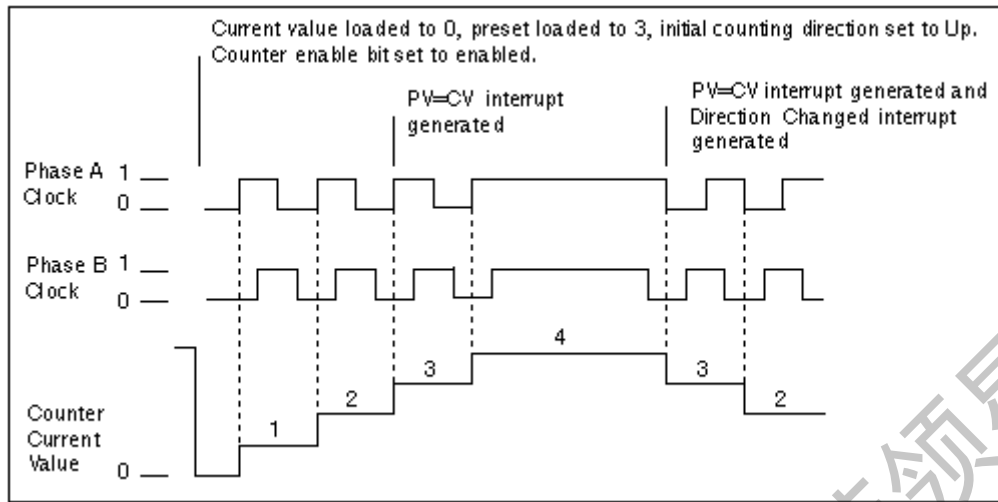


使用计数模式6、7和8时，上下时钟输入的上升边缘间隔0.3微秒，高速计数器可能认为这些事件同时发生。如果发生这种情况，在两种情况下，均不生成错误，而且计数器保持当前计数值。

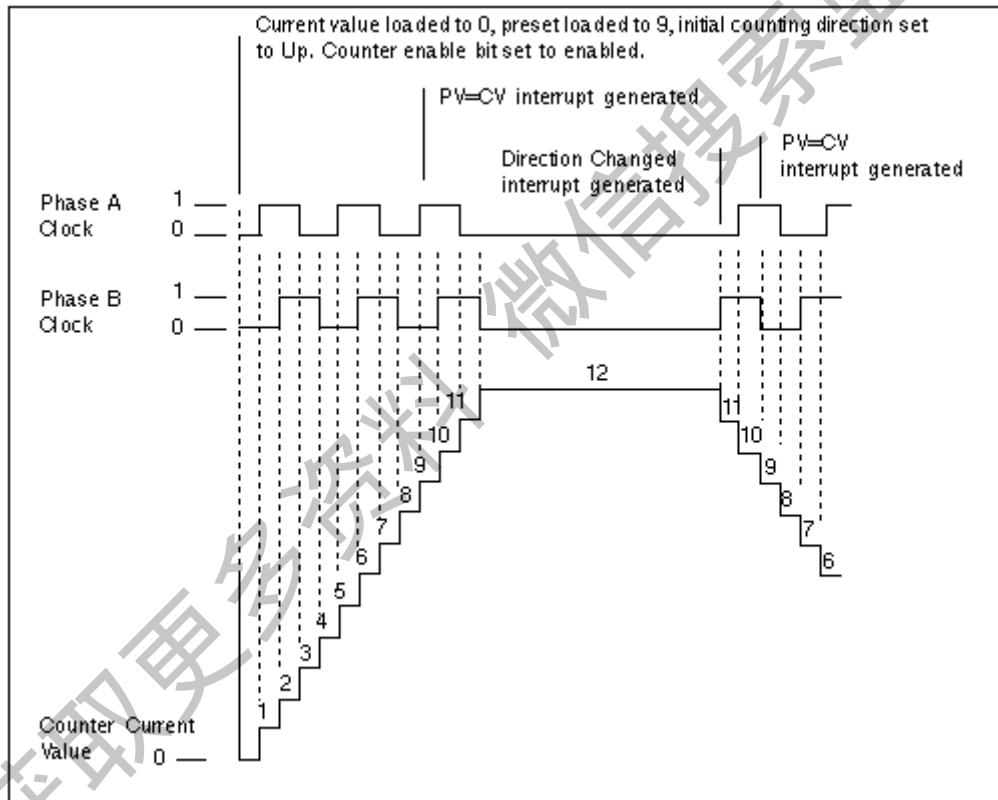
#### 模式6、7和8操作举例



#### 模式9、10和11操作举例（正交 1x 模式）



模式9、10和11操作举例（正交 4x 模式）



为高速计数器连接输入线

[返回顶端](#)

使用“高速计数器定义”指令定义计数器模式和输入。  
下表显示与高速计数器相关的用于时钟、方向控制、重设和起始功能的输入。

高速计数器专用输入

高速计数器	使用的输入
HSC0	I0.0, I0.1, O.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

有些高速计数器和边缘中断的输入点赋值存在某些重叠。同一个输入不能用于两种不同的功能；但是高速计数器当前模式未使用的任何输入均可用于其他目的。例如，如果在模式2中使用HSC0，模式2使用I0.0和I0.2，则I0.1可用于边缘中断或用于HSC3。如果所用的HSC0模式不使用输入I0.1，则该输入可用于HSC3或边缘中断。与此相似，如果所选的HSC0模式不使用I0.2，则该输入可用于边缘中断；如果所选HSC4模式不使用I0.4，则该输入可用于HSC5。

请注意HSC0的所有模式均使用I0.0，HSC4的所有模式均使用I0.3，因此当使用这些计数器时，这些输入点绝不会用于其他用途。

### HSC模式

[返回顶端](#)

HSC 模式	说明	输入			
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I0.2	I1.1
	HSC2	I1.2	I1.3	I1.1	I1.2
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	具有内部方向控制的单相计数器	时钟			
1		时钟		重设	
2		时钟		重设	开始
3	具有外部方向控制的单相计数器	时钟	方向		
4		时钟	方向	重设	
5		时钟	方向	重设	开始
6	具有两个时钟输入的两相计数器	正计时	倒计时		
7		正计时	倒计时	重设	
8		正计时	倒计时	重设	开始
9	A / B相正交计数器	时钟 A	时钟 B		
10		时钟 A	时钟 B	重设	
11		时钟 A	时钟 B	重设	开始
12	仅HSC0和HSC3支持12模式。 HSC0计数Q0.0所发脉冲的数目。 HSC3计数Q0.1所发脉冲的数目。				

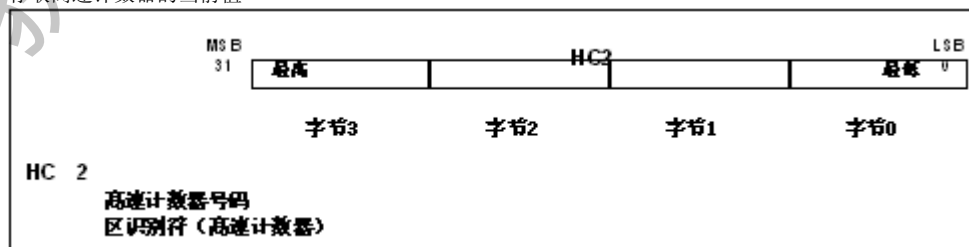
### 为高速计数器编址（HC）

[返回顶端](#)

欲存取高速计数器的计数值，您需要利用内存类型（HC）和计数器号码（例如HC0）指定高速计数器的地址。如下所示，高速计数器的当前值是只读数值，只能作为双字（32位）编址。

格式：HC[高速计数器号码]，以HC2为例。

存取高速计数器的当前值



### 理解不同的高速计数器

[返回顶端](#)

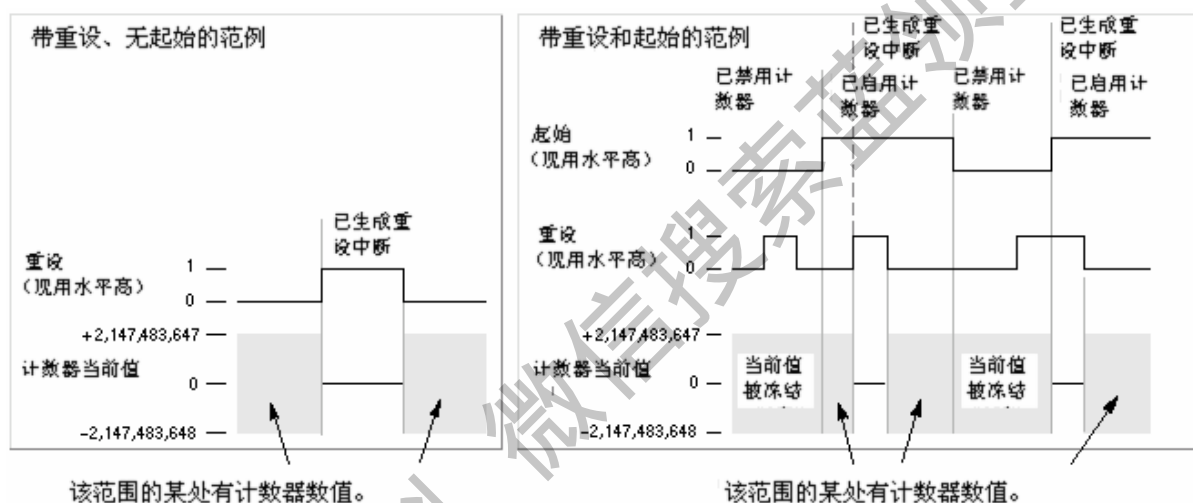
对于相同的计数器操作模式，所有计数器的功能均相同。如上所示，共有四种基本计数器模式类型。请注意并非每一种计数器均支持每种模式。您可以使用每种类型：无重设或起始输入、有重设但无起始或有起始和重设输入。

- 激活重设输入时，会清除当前值并保持清除状态直至取消激活重设。
- 激活起始输入时，会允许计数器计数。起始被取消激活时，计数器的当前值保持恒定，并忽略时钟事件。
- 如果在起始未激活时激活重设，则会忽略重设，当前值不变。如果激活重设输入时激活起始输入，则当前值被清除

使用高速计数器之前必须选择计数器模式，您可以利用 HDEF 指令（高速计数器定义）选择计数器模式。利用首次扫描内存位 S M0.1（首次扫描时该位打开，随后关闭），调用包含 HDEF 指令的子例行程序。

**选择激活状态和 1x/4x 模式**[返回顶端](#)

上图显示的重设和起始输入操作适用于使用重设和起始输入的所有模式。在重设和起始输入图形中，重设和起始均显示为现用状态编程为高位。



四台计数器有三个控制位，用于配置重设和起始输入的激活状态并选择 1x 或 4x 计数模式（仅限正交计数器）。这些控制位位于各自计数器的控制字节内，只在执行 HDEF 指令时才使用。

执行 HDEF 指令之前，必须将这些控制位设为所需的状态，否则计数器采用所选计数器模式的默认配置。重设输入和起始输入的仙柚梦 钟盟 礁撸 患割 俾饰 4x（或 4 乘以输入时钟频率）。一旦执行了 HDEF 指令，就不能再改变计数器设置，除非紫冉 CPU 设为 STOP（停止）模式。

**HDEF 控制位**

（仅在执行 HDEF 时使用）

HSC0	HSC1	HSC2	HSC4	说明
SM37.0	SM47.0	SM57.0	SM147.0	“重设”现用水平控制位**：0 =
重设现用水平高 1 = 重设现用水平低				
	SM47.1	SM57.1		“起始”现用水平控制位**：0 =
起始现用水平高 1 = 起始现用水平低				
SM37.2	SM47.2	SM57.2	SM147.2	“正交”计数器的计数速率选项：0 =
= 4x 计数速率 1 = 1x 计数速率				
**重设输入和起始输入的默认值为现用水平高，求积计数率为 4x（或 4 乘以输入时钟频率）。				

**控制字节**[返回顶端](#)

一旦定义了计数器和计数器模式，您就可以为计数器动态参数编程。每台高速计数器均有一个控制字节，允许完成以下作业：

- 启用或禁用计数器
- 控制方向（仅限模式 0、1 和 2）或初始化所有其他模式的计数方向
- 载入当前值

通过执行 HSC 指令可激活控制字节以及相关当前值和预设值检查。下表说明每个控制位。

**用于 HSC 参数的 SM 控制位**

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

### 设置当前值和预设值

[返回顶端](#)

每台高速计数器都有一个32位当前值和一个32位预设值，当前值和预设值均为带符号的整数。欲向高速计数器载入新的当前值和预设值，您必须设置包含当前值和 / 或预设值的控制字节及特殊内存字节。然后您必须执行 HSC指令，将新数值传输至高速计数器。下表说明用于包含新当前值和预设值的特殊内存字节。除控制字节以及新预设值和当前值保持字节外，还可以使用数据类型 HC（高速计数器当前值）加计数器号码（0、1、2、3、4或5）读取每台高速计数器的当前值。因此，读取操作可直接存取当前值，但只有用上述 HSC指令才能执行写入操作。

载入数值	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
新当前值	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
58						
新预设值	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162
62						

### 状态字节

[返回顶端](#)

为每台提供状态内存位的高速计数器提供状态字节，状态内存位表示当前计数方向以及当前值是否大于或等于预设值。下表定义每台高速计数器的状态位。

#### HSC0、HSC1、HSC2、HSC3、HSC4和HSC5的状态位

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

#### 注释:

只有在执行高速计数器中断例行程序时，状态位才有效。监控高速计数器状态的目的在于启用对正在执行的操作有重大影响的事件的中断程序。

### 为中断赋值

[返回顶端](#)

所有计数器模式均支持当前值等于预设值中断，使用外部重设输入的计数器模式支持将外部重设用中断。除模式 0、1和2以外，所有模式均支持外部重设用中断。

**注释:**

如果您尝试载入新的当前值或从外部重设中断例行程序内先禁用然后再重新启用高速计数器，会引起严重错误。

**高速计数器初始化顺序**

[返回顶端](#)

HSC1在以下初始化和操作顺序说明中被用作模型计数器。初始化说明假设 S7-200刚刚被放置在RUN（运行）模式中，因此首次扫描内存位为真。如果不是如此，请记住在进入RUN（运行）模式后，只能为刻齿轮廓 笔葱幸准 HDEF指令。为高速计数器第二次执行HDEF会生成运行时间错误，并不会改变该计数器首次执行HDEF时计数器的设置方式。虽然以下顺序分别显示如何更改方向、当前值和预设值，您可以按照相同的顺序更改所有这些数值或这些数值的任何组合，方法是以适当的方式设置SMB47数值，然后执行HSC指令。

**0、1或2初始化模式**

下列步骤说明如何为带内部方向的单相向上 / 向下计数器（模式 0、1或2）初始化HSC1:

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序  
蛇丝杉跽偈 枋奔渲葱胁(6)钩缘蚪拏垢 趵砣
2. 在初始化子例行程序中，根据所需的控制操作载入SMB47。例如：  
SMB47 = 16#F8产生下列结果：  
启动计数器  
写入新当前值  
写入新预设值  
将方向设置为向上计数  
将起始和重设输入设为现用水平高
3. 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）输入设为0，有外部重设但无起始设为1，有外部重设和起始设为2。
4. 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD52（双字尺寸数值）。
6. 为了捕获当前值等于预设值，将CV=P中断事件（事件13）附加于中断例行程序中，为中断编程。
7. 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
8. 执行全局中断启用指令（ENI），启用中断。
9. 执行HSC指令，使S7-200为HSC1编程。
10. 退出子例行程序。

**3、4或5初始化模式**

下列步骤说明如何为带外部方向的单相向上 / 向下计数器（模式 3、4或5）初始化HSC1:

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序  
蛇丝杉跽偈 枋奔渲葱胁(6)钩缘蚪拏垢 趵砣
2. 在初始化子例行程序中，根据所需的控制操作载入SMB47。例如：  
SMB47 = 16#F8产生下列结果：  
启用计数器  
写入新当前值  
写入新预设值  
设置HSC初始方向，向上计数  
将起始和重设输入设为现用水平高
3. 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）输入设为3，有外部重设但无起始设为4，有外部重设和起始设为5。
4. 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD52（双字尺寸数值）。
6. 为了捕获当前值等于预设值，将CV=P中断事件（事件13）附加于中断例行程序中，为中断编程。
7. 为了捕获方向改变，将方向改变中断事件（事件14）附加于中断例行程序中，为中断编程。
8. 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
9. 执行全局中断启用指令（ENI），启用中断。
10. 执行HSC指令，使S7-200为HSC1编程。
11. 退出子例行程序。

**6、7或8初始化模式**

下列步骤说明如何为带向上 / 向下时钟的双相向上 / 向下计数器（模式 6、7或8）初始化HSC1:

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序  
蛇丝杉跽偈 枋奔渲葱胁(6)钩缘蚪拏垢 趵砣
2. 在初始化子例行程序中，根据所需的控制操作载入SMB47。例如：  
SMB47 = 16#F8产生下列结果：  
启用计数器

写入新当前值  
 写入新预设值  
 设置HSC初始方向，向上计数  
 将起始和重设输入设为现用水平高

3. 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）设为6，有外部重设但无起始设为7，有外部重设和起始设为8。
4. 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD52（双字尺寸数值）。
6. 为了捕获当前值等于预设值，将CV=PV中断事件（事件13）附加于中断例行程序中，为中断编程。
7. 为了捕获方向改变，将方向改变中断事件（事件14）附加于中断例行程序中，为中断编程。
8. 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
9. 执行全局中断启用指令（ENI），启用中断。
10. 执行HSC指令，使S7-200为HSC1编程。
11. 退出子例行程序。

### 9、10或11初始化模式

下列步骤说明如何为A/B相正交计数器（模式9、10或11）初始化HSC1：

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序。
2. 在初始化子例行程序中，根据所需的控制操作载入SMB47。

例如（1x计数模式）：

SMB47 = 16#FC产生下列结果：

启用计数器  
 写入新当前值  
 写入新预设值  
 设置HSC初始方向，向上计数  
 将起始和重设输入设为现用水平高

例如（4x计数模式）：

SMB47 = 16#F8产生下列结果：

启用计数器  
 写入新当前值  
 写入新预设值  
 设置初始HSC方向，向上计数  
 将起始和重设输入设为现用水平高

3. 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）输入设为9，有外部重设但无起始设为10，有外部重设和起始设为11。
4. 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD52（双字尺寸数值）。
6. 为了捕获当前值等于预设值，将CV=PV中断事件（事件13）附加于中断例行程序中，为中断编程。
7. 为了捕获方向改变，将方向改变中断事件（事件14）附加于中断例行程序中，为中断编程。
8. 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
9. 执行全局中断启用指令（ENI），启用中断。
10. 执行HSC指令，使S7-200为HSC1编程。
11. 退出子例行程序。

### 12初始化模式

下面步骤描述了如何为计算由PTO0产生的脉冲数而初始化HSC0（模式12）。

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序。
2. 在初始化子例行程序中，根据所需的控制操作载入SMB47。例如：

SMB37 = 16#F8 产生下列结果：

启动计数器  
 写入新当前值  
 写入新预设值  
 将方向设置为向上计数  
 将起始和重设输入设为现用水平高

3. 在将HSC输入设为0且MODE（模式）输入设为12后执行HDEF指令。
4. 用所需的当前值载入SMD38（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD42（双字尺寸数值）。
6. 为了捕获当前值等于预设值事件，将CV = PV中断事件（事件13）附加于中断例行程序中，从而实现中断编程。如需关于中断处理的全部细节，请参阅讨论“中断指令”的章节。



7. 执行全局中断启用指令（ENI）以启用中断。
8. 执行HSC指令以使S7-200为HSC0编程。
9. 退出子例行程序。

#### 在0、1、2或12模式中改变方向

下列步骤说明如何配置HSC1，使带内部方向的单相计数器（模式0、1、2或12）改变方向：

1. 载入SMB47，写入所需的方向：  
SMB47 = 16#90  
启动计数器  
将HSC方向设为向下计数  
  
SMB47 = 16#98  
启动计数器  
将HSC方向设为向上计数
2. 执行HSC指令，使S7-200为HSC1编程。

#### 载入新当前值（任何模式）

下列步骤说明如何改变HSC1的计数器当前值（任何模式）：  
改变当前值强迫计数器在进行改动的过程中被禁用。计数器被禁用时，则不再计数或生成中断。

1. 载入SMB47，写入所需的当前值：  
SMB47 = 16#C0  
启用计数器  
写入新当前值
2. 用所需的当前值载入SMD48（双字尺寸）（载入零可加以清除）。
3. 执行HSC指令，使S7-200为HSC1编程。

#### 载入新预设值（任何模式）

下列步骤说明如何改变HSC1的计数器预设值（任何模式）：

1. 载入SMB47，写入所需的预设值：  
SMB47 = 16#A0  
启用计数器  
写入新预设值
2. 用所需的预设值载入SMD52（双字尺寸数值）。
3. 执行HSC指令，使S7-200为HSC1编程。

#### 禁用高速计数器（任何模式）

下列步骤说明如何禁用HSC1高速计数器（任何模式）：

1. 载入SMB47，禁用计数器：  
SMB47 = 16#00 禁用计数器
2. 执行HSC指令，禁用计数器。

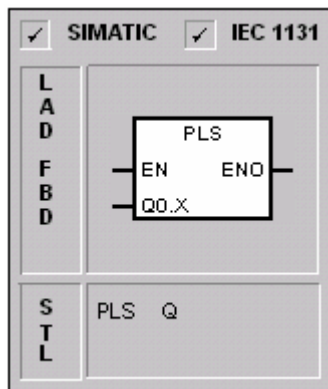
虽然上述操作步骤显示如何逐一改变方向、当前值和预设值，您也可以按照相同顺序，通过以适当方式设置 SMB47数值并执行HSC指令，改变全部数值或其中任何组合。

另请参阅：

[HSC向导](#)  
[HSC指令](#)  
[HDEF指令](#)

### 3.1.16 理解高速输出指令

操作数	数据类型	[[[ENDKEEPN]]]	
Q 常量 (0或1)		字	[[[ENDKEEPN]]]
内存范围	错误	S7-200 CPU指令支持	SIMATIC / 国际助记符
[[[ENDKEEPN]]]			
数据范围	ENO	CPU内存中的指令大小	编址内存 [[[ENDKEEPN]]]



**脉冲输出 (PLS)** 指令被用于控制在高速输入 (Q0.0和Q0.1) 中提供的“脉冲链输出” (PTO) 和“脉冲宽度调制” (PWM) 输出。PTO提供矩形波 (工作循环) 输出, 配备循环时间和脉冲数用户控制功能。

PWM提供连续性变量工作循环输出, 配备循环时间和脉冲宽度用户控制功能。脉冲输出范围 Q0.0至Q0.1

[特殊内存PTO / PWM高速输出寄存器](#)

### 理解S7-200高速输出指令

S7-

200有两台PTO/PWM生成器, 建立高速脉冲链或脉冲宽度调节波形。一台生成器指定给数字输出点 Q0.0, 另一台生成器指定给数字输出点Q0.1。一个指定的特殊内存 (SM) 位置为每台生成器存储以下数据: 一个控制字节 (8位数值)、一个脉冲计数值 (一个不带符号的32位数值) 和一个循环时间和脉冲宽度数值 (一个不带符号的16位数值)。

PTO/PWM生成器和进程图像寄存器共用Q0.0和Q0.1。PTO或PWM功能在Q0.0或Q0.1位置现用时, PTO/PWM生成器控制输出, 并禁止输出点的正常使用。输出波形不受进程图像寄存器状态、点强迫数值、执行立即输出指令的影响。PTO/PWM生成器非现用

保持住! 谱 柜 掏枷寞姆嫫螯 = 掏枷寞姆嫫骹韶去涑霾 4 蔚某脚己妥航兆刺 共 4 卧诩呱换虹臀膝 己徒嘎

#### 注释:

- 在启用PTO或PWM操作之前, 将用于Q0.0和Q0.1的进程图像寄存器设为0。
- 所有的控制位、循环时间、脉冲宽度和脉冲计数值的默认值均为0。
- PTO/PWM输出必须至少有10%的额定负载, 才能完成从关闭至打开以及从打开至关闭的顺利转换。
- 文档光盘“提示与技巧”中的提示7、22、23、30和50包含使用PTO/PWM操作PLS指令的程序。

脉冲链 (PTO) 功能提供矩形波 (50%工作循环) 输出或指定的脉冲数和指定的循环时间。脉冲宽度调制 (PWM) 功能提供带变量工作循环的固定循环时间输出。

每台PTO/PWM生成器有一个控制字节 (8位), 一个循环时间数值和脉冲宽度数值 (不带符号的16位数值) 和一个脉冲计数值 (不带符号的32位数值)。这些数值全部存储在特殊内存 (SM) 区域的指定位置。一旦设置这些特殊内存位的位置, 选择所需的值, 葱新此途涑蛸啮睿 PLS) 即启动操作。该指令使S7-200读取SM位置, 并为PTO/PWM生成器编程。

通过修改SM区域中 (包括控制字节) 要求的位置, 您可以更改 PTO或PWM的波形特征, 然后执行PLS指令。

您可以在任意时间向控制字节 (SM67.7或SM77.7) 的PTO/PWM启用位写入零, 禁用PTO或PWM波形的生成, 然后执行PLS指令

**注释:** 所有控制位、循环时间、脉冲宽度和脉冲计数值的默认值均为零。

**注释:** PTO/PWM输出必须至少有10%的额定负载, 才能完成从关闭至打开以及从打开至关闭的顺利转换。

#### PWM操作

PWM功能提供带变量工作循环的固定循环时间输出。可以微秒或毫秒为时基指定循环时间和脉冲宽度。

循环时间的范围从10微秒至65,535微秒, 或从2毫秒至65,535毫秒。

脉冲宽度时间范围从0微秒至65,535微秒或从0毫秒至65,535毫秒。



如上表所示, 设置脉冲宽度等于循环时间 (这使工作循环为100%) 使输出连续运行。设置脉冲宽度等于0 (这使工作循环为0%) 会关闭输出。

#### 脉冲宽度时间 / 循环时间

#### 反应

脉冲宽度时间  $\geq$  循环时间数值 工作循环为100%: 输出连续运行。

脉冲宽度时间 = 0 工作循环为0%: 输出关闭。

循环时间 < 2个时间单位 循环时间的默认值为两个时间单位。

有两种不同的方法可改变PWM波形的特征: 同步更新和异步更新。

- 同步更新: 如果不要求更改时基, 即可以执行同步更新。执行同步更新时, 波形特征的变化发生在循环边缘, 提供顺

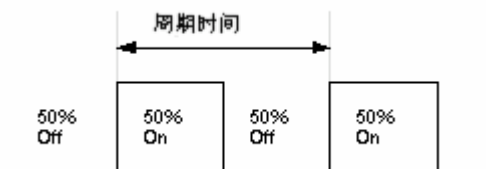
弧

- 异步更新：此为常见的PWM操作，脉冲宽度不同，但循环时间保持不变。因此，不要求时基改变。但是，如果要求改变PTO/PWM生成器的时基，则应使用异步更新。异步更新使PTO/PWM生成器被立即禁用，与PWM波形异步。这样可能造成控制设备状态暂时不稳。由于此一原因，建议使用同步PWM更新。选择可用于所有预计循环时间数值的时基。

控制字节中的PWM更新方法位（SM67.4或SM77.4）指定更新类型，在执行PLC指令时激活改动。请注意，如果时基改变，则会

#### PTO操作

PTO为指定的脉冲数和指定的循环时间提供矩形波（50%工作循环）输出。PTO可提供单脉冲链或多脉冲链（使用脉冲轮廓）。



循环时间范围从10微秒至65,535微秒或从2毫秒至65,535毫秒。

脉冲计数范围从1至4,294,967,295次脉冲。

为循环时间指定基数微秒或毫秒（例如75毫秒）会引起工作循环的失真。

下表定义脉冲计数和循环时间限制：

#### 脉冲计数 / 循环时间

#### 反应

循环时间 < 2个时间单位

循环时间的默认值为2个时间单位。

脉冲计数 = 0

脉冲计数的默认值为1次脉冲。

状态字节（SM66.7或SM76.7）中的PTO空闲位表示编程脉冲链已完成。另外，也可在脉冲链完成时激活中断例行程序。如果您使用多段操作，则在轮廓表完成时立即激活中断例行程序。请参阅以下多段管线连接。

PTO功能允许脉冲链链接或管线作业。现用脉冲链完成时，新的脉冲链输出立即开始。这样就保证了随后的输出脉冲链的连续性。

该管线作业可以两种方式中的一种完成：单段管线作业或多段管线作业。

#### 单段管线作业

在单段管线作业中，您负责更新下一个脉冲链的SM位置。初始PTO段一旦开始，您必须按照对第二个波形的要求立即修改SM位置（00伏沃葱 PLS指令。第二个脉冲链特征被保留在管线中，直至第一个脉冲链完成。管线中每次只能存储一个条目。第一个脉冲链一旦完成，第二个波形输出即开始，管线可用于新的脉冲链规格。您可以重复此步骤，设置下一个脉冲链的特征。

可在脉冲链之间平稳转换，下列情况除外：

- 如果执行时基改动
- 如果现用脉冲链在执行PLS指令捕获到新脉冲链设置之前完成

如果您在管线已满时尝试载入，状态寄存器（SM66.6或SM76.6）中的PTO溢出位被设置。进入RUN（运行）模式时，该位被初始化为0。如果您希望探测随后出现的溢出，则必须在探测到溢出之后以手动方式清除该位。

#### 多段管线作业

在多段管线作业中，S7-200从V内存中的轮廓表自动读取每个脉冲链段的特征。该模式中的SM位置是轮廓表的控制字节、状态字节和起始V内存偏移量（SMW

168或SMW178）。时基可以为微秒或毫秒，但该选项适用于轮廓表中的所有循环时间数值，但在轮廓运行时不得变更。然后可由执行PLS指令开始多段操作。

每段输入的长度均为8个字节，由一个16位循环时间数值、一个16位循环时间Δ数值和一个32位脉冲计数数值组成。

下表说明轮廓表的格式。多段PTO操作的另一个特征是能够通过指定每个脉冲的数量自动增加或减少循环时间。在循环时间Δ域

暗陶 痔嵩皂友 肥奔洌 谱 肥奔涸 卍 虬暗谈褐袴峒跽 肥奔涸H 羰 滴 悖 蛭 肥奔洌槐涸

如果您指定的循环时间Δ数值在一定数量的脉冲后导致非法循环时间，则会出现数学溢出条件。PTO功能被终止，输出转换成图

蠹拇嫫甯 司啤 4 送 猓 刺 纸冢 SM66.4或SM76.4)中的Δ计算错误位被设为一。

如果您以手动方式异常中止正在运行的PTO轮廓，状态字节（SM66.5或SM76.5）中的用户异常中止位则被设为一。

运行PTO轮廓时，SMB166（或SMB176）中提供当前现用段数。

#### 多段PTO操作的轮廓表格式

距离轮廓表起始位置的 字节偏移量

偏移量	段数	表格条目说明
0		轮廓段数目
1	#1	段数（1至255）；数值0生成非严重错误，生成无PTO输出
3		初始循环时间（2至65535个时基单位）
5		每次脉冲的循环时间Δ（带符号的数值）（-32768至32767个时基单位）
9		脉冲计数（1至4294967295）
11	#2	初始循环时间（2至65535个时基单位）
13		每个脉冲的循环时间Δ（带符号的数值）（-32768至32767个时基单位）
15		脉冲计数（1至4294967295）
::	#3	::

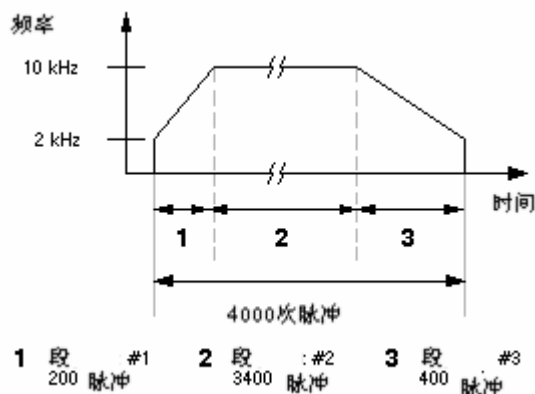
### 计算轮廓表数值

PTO/PWM生成器的多段管线作业功能在许多应用程序中都很有用，特别是步阶器电机控制中。

例如，您可以通过简单斜坡向上、运行和斜坡向下顺序或更复杂的顺序使用配备脉冲轮廓的 PTO，控制步进器马达，方法是定义恒速脉冲 255 个段的脉冲轮廓，每个段与一个斜坡向上、运行或斜坡向下操作相对应。

下图显示生成加速步进器马达（#1段）、按恒速操作马达（#2段）、随后减低马达速度（#3段）的输出波形所要求的轮廓表数值样本。

### 简单步阶器电机应用程序的频率与时间图形举例



在本例中：起始和终止脉冲频率为 2 kHz，最大脉冲频率为 10 kHz

，要求 4000 次脉冲才能达到所需的马达转动次数。因为用阶段（循环时间）表示轮廓表数值，而不使用频率，需要将给定频率数值转换成循环时间数值。因此，起始（最初）和终止（结束）循环时间为 500 ms，与最大频率对应的循环时间为 100 ms。在输出轮廓的加速部分，应在约 400 次脉冲时达到最大脉冲频率。轮廓减速部分应在约 400 次脉冲时完成。

您可以用以下公式确定 PTO/PWM 生成器用于调节某一特定段每次脉冲循环时间的  $\Delta$  循环时间数值：

$$\text{段的 } \Delta \text{ 周期时间} = \left| \text{End\_CT}_{\text{seg}} - \text{Init\_CT}_{\text{seg}} \right| / \text{Quantity}_{\text{seg}}$$

位置：  
 $\text{End\_CT}_{\text{seg}}$  = 该段的结束周期时间  
 $\text{Init\_CT}_{\text{seg}}$  = 该段的初始周期时间  
 $\text{Quantity}_{\text{seg}}$  = 该段的脉冲数

利用该公式，计算出的加速部分（或 #1 段）的  $\Delta$  循环时间为 -

2。与此相似，减速部分（或 #3 段）的  $\Delta$  循环时间为 1。因为 #2 段是输出波形的恒速部分，该段的  $\Delta$  循环时间为零。

假定轮廓表位于从 V500 开始的 V 内存中，以下显示用于生成所需波形的表值。您可以在程序中包含指令，将这些数值载入 V 内存。

### 轮廓表数值举例

V 内存地址	数值	说明
VB500	总段数	
VW501	初始循环时间	#1 段
VW503 -2	初始 $\Delta$ 循环时间	
VD505	脉冲数	
VW509	初始循环时间	#2 段
VW511	$\Delta$ 循环时间	
VD513	脉冲数	
VW517	初始循环时间	#3 段
VW519	$\Delta$ 循环时间	
VD521	脉冲数	

利用程序中的指令可将这些表值置于 V 内存中。另一种方法是定义数据块中的轮廓值。本标题结尾部分举例说明使用多段 PTO 操作某条轴。

为了确定波形段之间的转换是否可接受，您需要确定段中最后一次脉冲的循环时间。除非  $\Delta$  循环时间是 0，您必须计算段最后一

段的最后一次脉冲周期时间  $= \text{Init\_CT}_{\text{seg}} + (\text{Delta}_{\text{seg}} * (\text{Quantity}_{\text{seg}} - 1))$

位置 :  $\text{Init\_CT}_{\text{seg}} =$  该段的结束周期时间

$\text{Delta}_{\text{seg}} =$  该段的初始周期时间

$\text{Quantity}_{\text{seg}} =$  该段的脉冲数

上例是简化的情况，用于介绍目的，实际应用程序可能要求更复杂的波形轮廓。请记住：

- $\Delta$  循环时间只能指定为整数微秒或毫秒
- 可对每次脉冲执行循环时间修改

这两个项目产生的效果是计算某个特定段的  $\Delta$  循环时间数值可能要求一个循环方案。计算某个特定段的结束循环时间或脉冲数目时可能要求一定的灵活性。

特定轮廓段期限对确定正确的轮廓表数值程序有用。可利用以下公式计算完成特定轮廓段的时间长度：

段时间长度  $= \text{Quantity}_{\text{seg}} * (\text{Init\_CT} + ((\text{Delta}_{\text{seg}}/2) * (\text{Quantity}_{\text{seg}} - 1)))$

位置 :  $\text{Quantity}_{\text{seg}} =$  该段的结束周期时间

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

$\text{Delta}_{\text{seg}} =$  该段的脉冲数

#### PTO/PWM控制寄存器

PLS指令读取存储在指定的SM内存位置的数据，并以此为PTO/PWM生成器编程。SMB67控制PTO 0或PWM 0，SMB77控制PTO 1或PWM 1。PTO/

PWM控制寄存器表描述用于控制PTO/PWM操作的寄存器。您可以将下表用作快速参考，帮助确定放置在PTO/PWM控制寄存器中用于激活所需操作的数值。

您可以改变PTO或PWM波形的特征，方法是修改SM区（包括控制字节）中的位置，然后执行PLS指令。您可以在任何时间禁用PTO或PWM波形的生成，方法是向控制字节（SM67.7或SM77.7）的PTO/PWM启用位写入0，然后执行PLS指令。

状态字节中的PTO空闲位（SM66.7或SM76.7）表示编程脉冲链已完成。此外，可在脉冲链完成时激活中断例行程序。（请参阅中断指令说明和“通讯”指令。）如果您在使用多段操作，在轮廓表完成时激活中断例行程序。

以下条件设置SM66.4（或SM76.4）和SM66.5（或SM76.5）：

- 指定一个在数次脉冲后导致非法循环时间的循环时间  $\Delta$  数值生成一个数学溢出条件，该条件会终止PTO功能，并将“ $\Delta$  计算错误”位（SM66.4或SM76.4）设为1。输出回复为图像寄存器控制。
- 以手动方式异常中止（禁用）正在执行的PTO轮廓会将“用户异常中止”位（SM66.5或SM76.5）设为1。
- 尝试在管线已满的情况下载入会将PTO溢出位（SM66.6或SM76.6）设为1。如果您希望检测随后的溢出，您必须在检测到溢出后以手动方式清除该位。转换至RUN（运行）模式可将该位初始化为0。

**注释：**当您载入新脉冲计数（SMD72或SMD82）、脉冲宽度（SMW70或SMW80）或循环时间（SMW68或SMW78）时，在执行PLS指令之前，还需要在控制寄存器中设置适当的更新位。对于多段脉冲链操作，在执行PLS指令之前，您还必须载入轮廓表的起始偏移量（SMW168或SMW178）和轮廓表数值。

#### PTO/PWM控制寄存器

Q0.0	Q0.1	状态位		
SM66.4		SM76.4	PTO轮廓由于 $\Delta$ 计算错误异常中止	
		0 = 无错；	1 = 异常中止	
SM66.5		SM76.5	PTO轮廓由于用户命令异常中止	
		0 = 无错；	1 = 异常中止	
SM66.6		SM76.6	PTO管线溢出 / 下溢	0 =
无溢出；		1 = 溢出 / 下溢		
SM66.7		SM76.7	PTO空闲	0 =
进行中；		1 = PTO空闲		
Q0.0	Q0.1	控制位		
SM67.0		SM77.0	PTO/PWM更新循环时间值	0 =
无更新；		1 = 更新循环时间		
SM67.1		SM77.1	PWM更新脉冲宽度时间值	0 =
无更新；		1 = 更新脉冲宽度		
SM67.2		SM77.2	PTO更新脉冲计数值	0 =
无更新；		1 = 更新脉冲计数		
SM67.3		SM77.3	PTO/PWM时基选择	0 = 1
祗/tick；		1 = 1ms/tick		
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启用	0 = 禁用PTO/PWM； 1

= 启用PTO/PWM

Q0.0	Q0.1	其他PTO/PWM寄存器	
SMW68 65535)		SMW78	PTO/PWM循环时间值 (范围: 2至
SMW70		SMW80	PWM脉冲宽度值 (范围: 0至65535)
SMD72		SMD82	
PTO脉冲计数值 (范围: 1至4294967295)			
SMB166		SMB176	进行中的段数 (仅用于多段PTO操作)
SMW168		SMW178	
轮廓表起始位置, 用距离V0的字节偏移量表示 (仅用于多段PTO操作)			
SMB170		SMB180	线性轮廓状态字节
SMB171		SMB181	线性轮廓结果寄存器
SMB172		SMB182	手动模式频率寄存器

**PTO/PWM控制字节参考**

控制寄存器 (十六进制数值)	启用 脉冲计数	执行PLS指令的结果 选择模式 脉冲宽度	PTO段操作 循环时间	PWM更新方法
时基 16#81	是	PTO 载入	单段	衽 / 循环
16#84	是	PTO	单段	衽 / 循环
载入 16#85	是	PTO 载入	单段	衽 / 循环
载入 16#89	是	PTO 载入	单段	ms / 循环
16#8C	是	PTO	单段	ms / 循环
载入 16#8D	是	PTO 载入	单段	ms / 循环
16#A0	是	PTO	多段	衽 / 循环
16#A8	是	PTO	多段	ms / 循环
16#D1	是	PWM 载入		同步 衽 / 循环
16#D2	是 载入	PWM		同步 衽 / 循环
16#D3	是 载入	PWM 载入		同步 衽 / 循环
16#D9	是	PWM 载入		同步 ms / 循环
16#DA	是 载入	PWM		同步 ms / 循环
16#DB	是 载入	PWM 载入		同步 ms / 循环

**PTO/PWM初始化和操作顺序**

以下是初始化和操作顺序说明, 能够帮助您更好地理解 PTO和PWM功能操作。在整个顺序说明过程中一直使用脉冲输出 Q0.0。初始化说明假定S7-200刚置入RUN (运行) 模式, 因此首次扫描内存位为真实。如果不是如此或者如果必须对 PTO/PWM功能重新初始化, 您可以利用除首次扫描内存位之外的一个条件调用初始化例行程序。

**PWM初始化**

以下PWM初始化和操作顺序说明建议使用“首次扫描”位 (SM0.1) 初始化脉冲输出。使用“首次扫描”位调用初始化子例行程序可降低扫描时间, 因为随后的扫描无须调用该子例行程序。(仅需在转换为 RUN (运行) 模式后的首次扫描时设置“首次扫描”位。)但是, 您的应用程序可能有其他限制, 要求您初始化 (或重新初始化) 脉冲输出。在此种情况下, 您可以使用另一个条件调用初始化例行程序。

通常, 您用一个子例行程序为脉冲输出初始化PWM。您从主程序调用初始化子例行程序。使用首次扫描内存位 (SM0.1) 将脉冲输出初始化为0, 并调用子例行程序, 执行初始化操作。当您使用子例行程序调用时, 随后的扫描不再调用该子例行程序, 这样

从主程序建立初始化子例行程序调用后, 用以下步骤建立控制逻辑, 用于在初始化子例行程序中配置脉冲输出 Q0.0:

1. 通过将以下一个数值载入SMB67: 16#D3 (选择微秒递增) 或16#DB (选择毫秒递增) 的方法配置控制字节。  
两个数值均可启用PTO/PWM功能、选择PWM操作、设置更新脉冲宽度和循环时间数值、以及选择时基 (微秒或毫秒)。
2. 在SMW68中载入一个循环时间的字尺寸数值。
3. 在SMW70中载入脉冲宽度的字尺寸数值。
4. 执行PLS指令 (以便S7-200为PTO/PWM生成器编程)。
5. 欲为随后的脉冲宽度变化预载一个新控制字节数值 (选项), 在 SMB67: 16#D2 (微秒) 或16#DA (毫秒) 中载入下列数值之一。
6. 退出子例行程序。

**为PWM输出更改脉冲宽度**

如果您用16#D2或16#DA预载SMB67（请参阅以上第5步），您可以使用一个将脉冲宽度改变为脉冲输出（Q0.0）的子例行程序。建立对该子例行程序的调用后，使用以下步骤建立改变脉冲宽度的控制逻辑：

1. 在SMW70中载入新脉冲宽度的字尺寸数值。
2. 执行PLS指令，使S7-200为PTO/PWM生成器编程。
3. 退出子例行程序。

**PTO初始化—单段操作**

以下PWM初始化和操作顺序说明建议使用“首次扫描”位（SM0.1）初始化脉冲输出。使用“首次扫描”位调用初始化子例行程序可降低扫描时间，因为随后的扫描无须调用该子例行程序。（仅需在转换为RUN（运行）模式后的首次扫描时设置“首次扫描”位。）但是，您的应用程序可能有其他限制，要求您初始化（或重新初始化）脉冲输出。在此种情况下，您可以使用另一个条件调用初始化例行程序。

通常，您用一个子例行程序为脉冲输出初始化PWM。您从主程序调用初始化子例行程序。使用首次扫描内存位（SM0.1）将脉冲输出初始化为0，并调用子例行程序，执行初始化操作。当您使用子例行程序调用时，随后的扫描不再调用该子例行程序，这样

从主程序建立初始化子例行程序调用后，用以下步骤建立控制逻辑，用于在初始化子例行程序中配置脉冲输出 Q0.0:

1. 通过将以下一个数值载入SMB67: 16#85（选择微秒递增）或16#8D（选择毫秒递增）的方法配置控制字节。
2. 两个数值均可启用PTO/PWM功能、选择PWM操作、设置更新脉冲宽度和循环时间数值、以及选择时基（微秒或毫秒）。

在SMW68中载入一个循环时间的字尺寸数值。

3. 在SMD72中载入脉冲计数的双字尺寸数值。
4. （选项）如果您希望在脉冲链输出完成后立即执行相关功能，您可以将脉冲链完成事件（中断类别 19）附加于中断子谱缘颞 枋奔渲葱校 (8)峁┘峁垢 辖鞅某缘颞 卸媳嗜蹇 褂 ATCH指令并执行全局中断启用指令ENI。
5. 执行PLS指令，使S7-200为PTO/PWM生成器编程。
6. 退出子例行程序。

**改变PTO循环时间—单段操作**

对于单段PTO操作，您可以使用中断例行程序或子例行程序改变循环时间。欲使用单段PTO操作更改中断例行程序或子例行程序机械PTO循环时间，请遵循下列步骤：

- 设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基、设置更新循环时间数值），方法是在SMB67: 16#81（用于微秒）或16#89（用于毫秒）中载入下列一个数值。
- 在SMW68中，载入新循环时间的一个字尺寸数值。
- 执行PLS指令，使S7-200为PTO/PWM生成器编程。更新脉冲计数波形输出开始之前，CPU必须完成所有进行中的PTO。
- 退出中断例行程序或子例行程序。

**改变PTO脉冲计数—单段操作**

对于单段PTO操作，您可以使用中断例行程序或子例行程序改变脉冲计数。欲使用单段PTO操作在中断例行程序或子例行程序中改变PTO脉冲计数，请遵循下列步骤：

1. 设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基、设置更新循环时间数值），方法是在SMB67: 16#84（用于微秒）或16#8C（用于毫秒）中载入以下两个数值之一。
2. 在SMD72中，载入新脉冲计数的一个字尺寸数值。
3. 执行PLS指令（以便S7-200为PTO/PWM生成器编程）。开始用更新脉冲计数生成波形之前，S7-200完成所有进行中的PTO。
4. 退出中断例行程序或子例行程序。

**改变PTO循环时间和脉冲计数—单段操作**

对于单段PTO操作，您可以使用中断例行程序或子例行程序改变循环时间和脉冲计数。欲使用单段PTO操作更改中断例行程序或子例行程序机械PTO循环时间和脉冲计数，请遵循下列步骤：

1. 设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基、设置更新循环时间和脉冲计数数值），方法是在SMB67: 16#85（用于微秒）或16#8D（用于毫秒）中载入以下两个数值之一。
2. 在SMW68中，载入新循环时间的一个字尺寸数值。
3. 在SMD72中，载入新脉冲计数的一个字尺寸数值。
4. 执行PLS指令，使S7-200为PTO/PWM生成器编程。用更新脉冲计数和脉冲时间波形输出开始之前，CPU必须完成所有进行中的PTO。
5. 退出中断例行程序或子例行程序。

**PTO初始化—多段操作**

通常，您用一个子例行程序为多段操作的脉冲输出配置和初始化PTO。您从主程序调用初始化子例行程序。使用首次扫描内存位（SM0.1）将PTO使用的输出初始化为0，并调用子例行程序，执行初始化操作。当您使用“首次扫描”位调用初始化子例行程序保 婧蟪纳 璨辉俚饕酶米永 谱缘颞 皮 峁垢 枋奔渲葱小

从主程序建立对初始化例行程序的调用后，使用以下步骤建立控制逻辑，用于在初始化子例行程序中配置脉冲输出 Q0.0: 使用首次扫描内存位（SM0.1）将输出初始化为0，并调用您所需的子例行程序，执行初始化操作。这样会降低扫描时间执行，并提供结构更严谨的程序。

1. 通过将以下一个数值载入SMB67: 16#A0（选择微秒递增）或16#A8（选择毫秒递增）的方法配置控制字节。

- 两个数值均可启用PTO/PWM功能、选择PTO操作、选择多段操作、以及选择时基（微秒或毫秒）。
2. 在SMW168中载入一个字尺寸数值，用作轮廓表起始V内存偏移量。
  3. 使用V内存存在轮廓表中设置段值。确保“段数”域（表的第一个字节）正确无误。
  4. （选项）如果您希望在PTO轮廓完成后立即执行相关功能，您可以将脉冲链完成事件（中断类别 19）附加在中断子例行程序中，为中断编程。使用ATCH执行全局中断启用指令ENI。
  5. 执行PLS指令，使S7-200为PTO/PWM生成器编程。
  6. 退出子例行程序。

```

NETWORK 1 // 用于PWM范例的主程序
// 首次扫描时，将图像寄存器位设为低，并调用 SBR_0
LD SMO.1
R Q0.1 1
CALL SBR_0

```

```

NETWORK 2 // 设置程序中其他位置的MO.0，将脉冲宽度更改为50%工作循环
LD MO.0
EU
CALL SBR_1

```

```

NETWORK 1 // 子例行程序0开始
LD SMO.0
MOVB 16#DB SMB77 // 设置控制字节
// - 选择PWM操作
// - 选择毫秒递增和同步更新
// - 设置脉冲宽度和循环时间数值
// - 启用PWM功能
MOVW +10000 SMW78 // 将循环时间设为10,000毫秒
MOVW +1000 SMW80 // 将脉冲宽度设为1,000毫秒
PLS 1 // 激活PWM操作：PLS1=>Q0.1
MOVB 16#DA SMB77 // 重新载入控制字节，用于随后的脉冲宽度改动

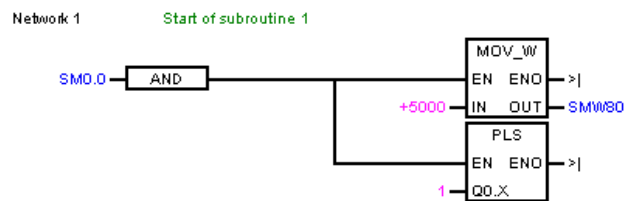
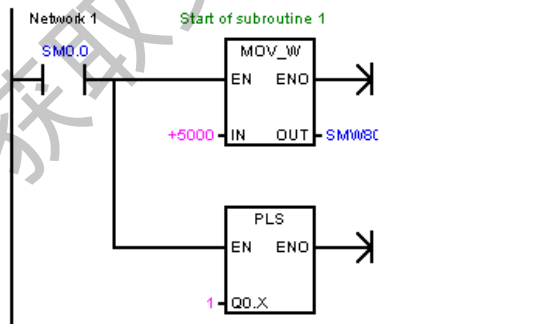
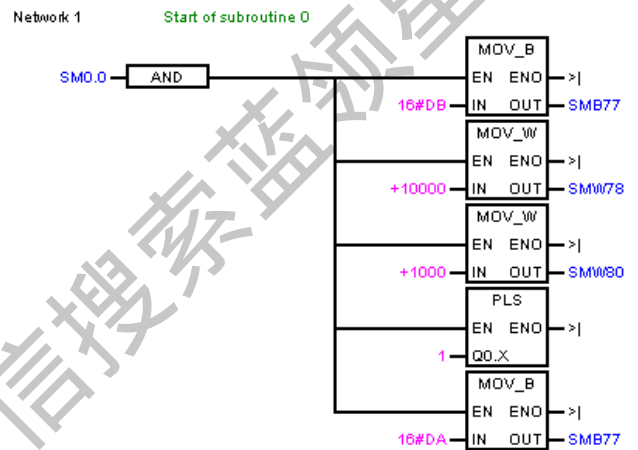
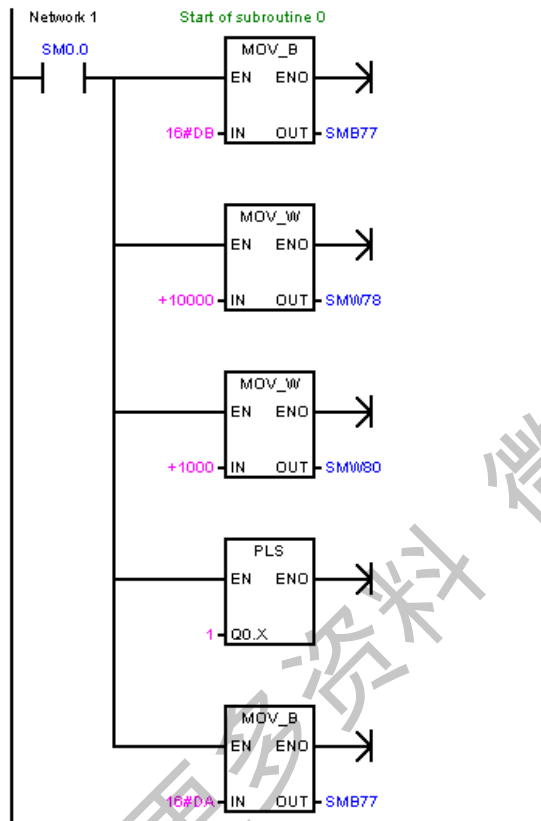
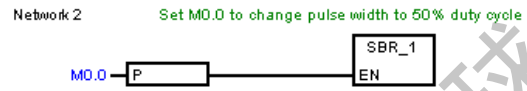
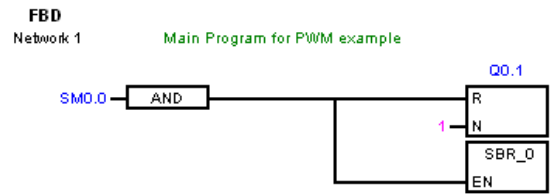
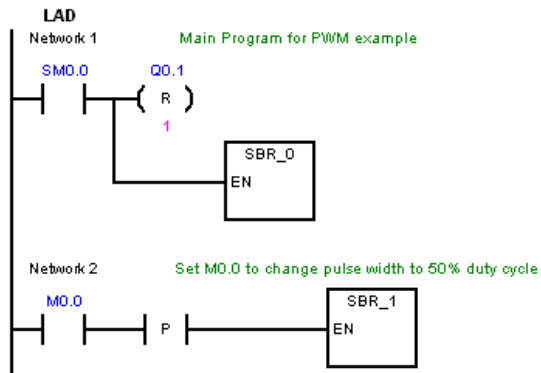
```

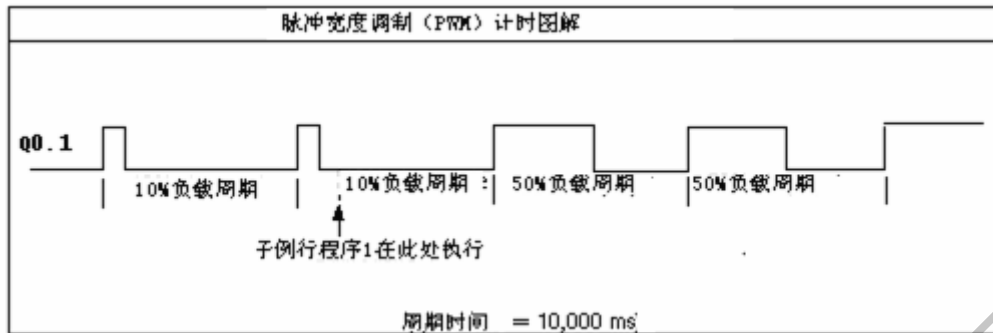
```

NETWORK 1 // 子例行程序1开始
LD SMO.0
MOVW +5000 SMW80 // 将脉冲宽度设为5000毫秒
PLS 1 // 断言脉冲宽度改动

```







### 使用单段操作的脉冲链输出举例

```

NETWORK 1 // 用于单段脉冲链操作的主程序 (PTO)
// 首次扫描时, 将图像寄存器位设为低
// 并调用子例行程序0
LD SMO. 1
R Q0. 0 1
CALL SBR_0

```

```

NETWORK 1 // 子例行程序0开始
LD SMO. 0
MOVB 16#8D SMB67 // 设置控制字节:
// - 选择PTO操作
// - 选择单段操作
// - 选择毫秒递增
// - 设置脉冲计数和循环时间数值
// - 启用PTO功能
MOVW +500 SMD72 // 将循环时间设为500毫秒。
MOVB +4 SMD72 // 将脉冲计数设为4次脉冲。
ATCH INT_0 19 // 将中断例行程序0定义为
// 处理PTO完成中断的中断。
ENI // 全局中断启用
PLS 0 // 激活PTO操作, PLS0 => Q0. 0
MOVB 16#89 SMB67 // 预载控制字节, 用于随后的
// 循环时间改动。

```

```

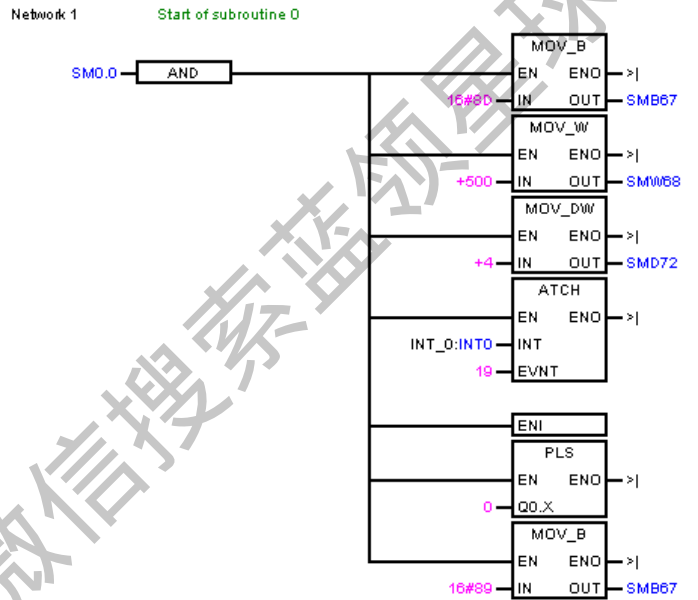
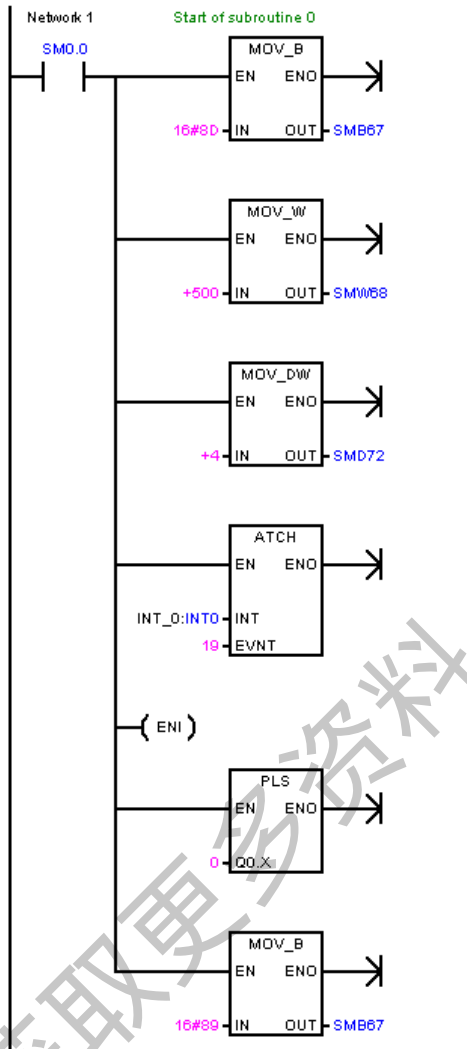
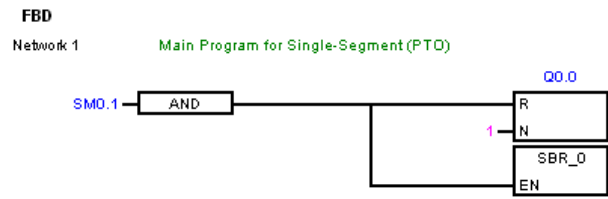
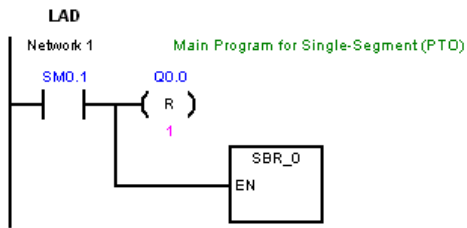
NETWORK 1 // 中断0开始
// 如果当前循环时间为500毫秒:
// 将循环时间设为1000毫秒, 并生成4次脉冲
LDW= SMD72 +500
MOVW +1000 SMD72
PLS 0
CRETI

```

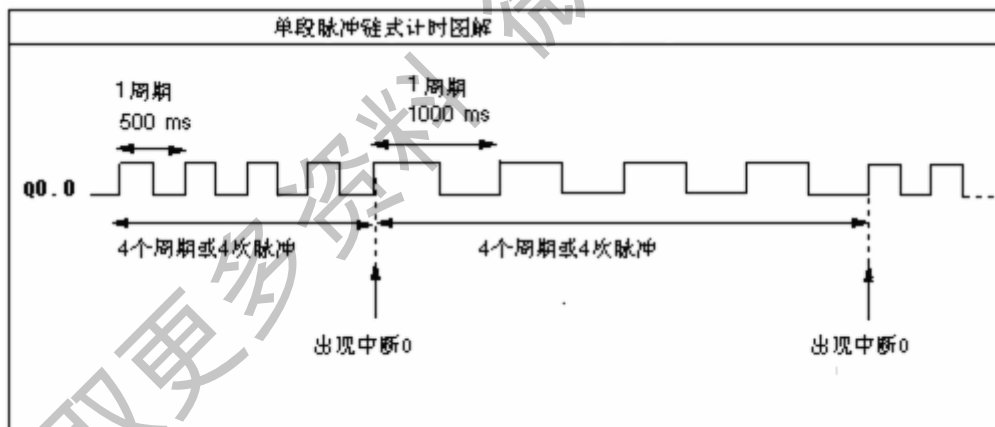
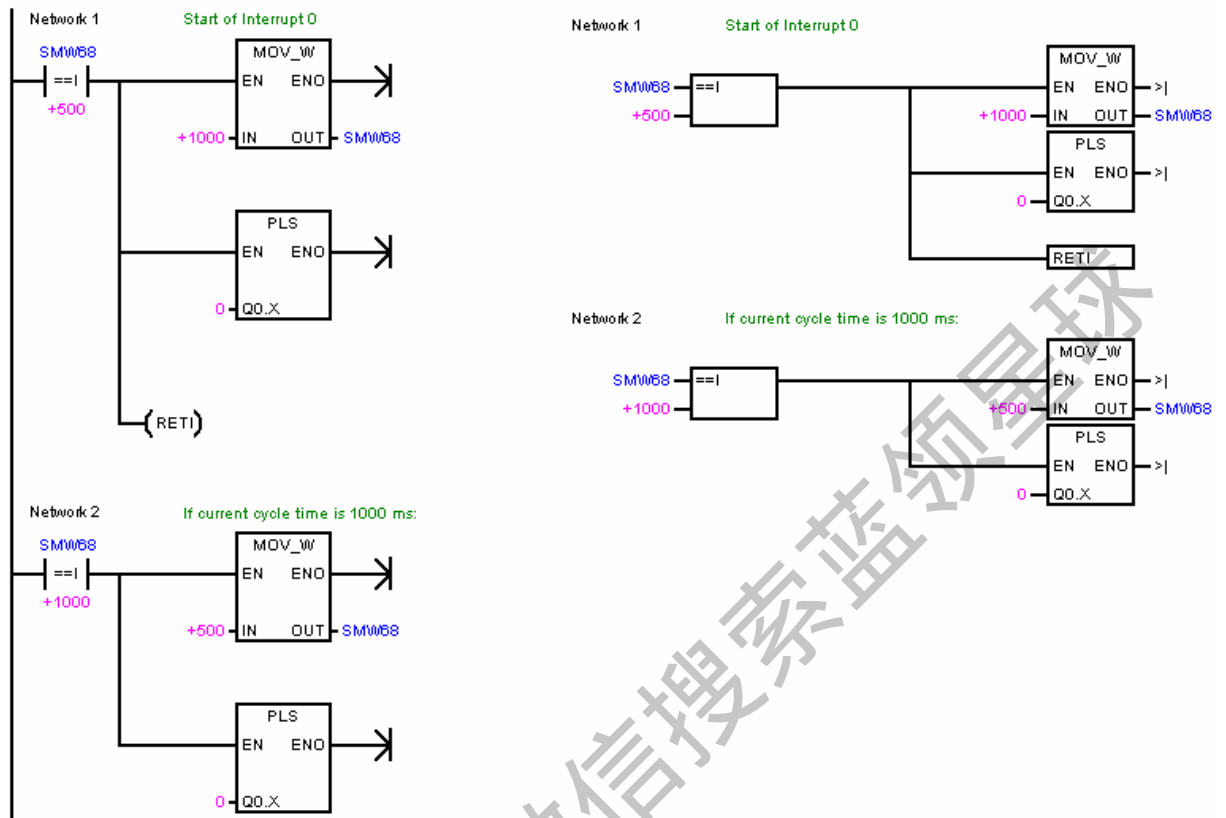
```

NETWORK 2
// 如果当前循环时间为1000毫秒:
// 将循环时间设为500毫秒, 并生成4次脉冲
LDW= SMD72 +1000
MOVW +500 SMD72
PLS 0

```



获取更多资料 微信搜索 蓝领星球



### 使用多段操作的脉冲链输出举例

```

NETWORK 1 // 用于多段脉冲链操作的主程序 (PTO)
// 首次扫描时, 将图像寄存器位设为低
// 并调用子例行程序0
LD SMO. 1
R Q0. 0 1
CALL SBR_0

```

```

NETWORK 1 // 子例行程序0开始
// 预载PTO轮廓表
LD SMO. 0

```

```

MOVB 3 VB500 // 将轮廓表段数设为3
// 段1:
MOVW +500 VW501 // 将段1的初始循环时间设为500毫秒
MOVW -2 VW503 // 将段1的Δ循环时间设为-2毫秒
MOVD +200 VD505 // 将段1中的脉冲数设为200
// 段2:
MOVW +100 VW509 // 将段2的初始循环时间设为100毫秒
MOVW +0 VW511 // 将段2的Δ循环时间设为0毫秒
MOVD +3400 VD513 // 将段2中的脉冲数设为3400
// 段3:
MOVW +100 VW517 // 将段3的初始循环时间设为100毫秒
MOVW +1 VW519 // 将段3的Δ循环时间设为1毫秒
MOVD +400 VD521 // 将段3中的脉冲数设为400

```

```

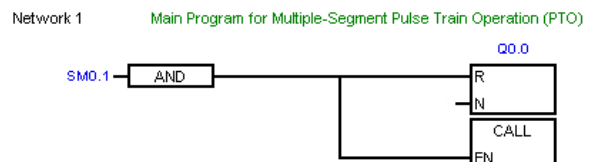
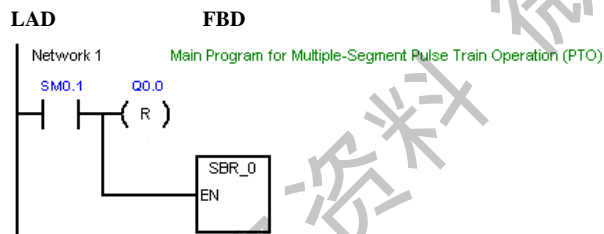
NETWORK 2
LD SMO.0
MOVB 16#A8 SMB67 // 设置控制字节:
// - 选择PTO操作
// - 选择多段操作
// - 选择毫秒递增
// - 启用PTO功能
MOVW +500 SMW168 // 将轮廓表的该起始地址
// 指定为V500。
ATCH INT_0 19 // 将中断例行程序0定义为
// 处理PTO完成中断的中断。
ENI // 全局中断启用
PLS 0 // 激活PTO操作, PLS0 => Q0.0
MOVB 16#89 SMB67 // 预载控制字节, 用于随后的
// 循环时间改动。

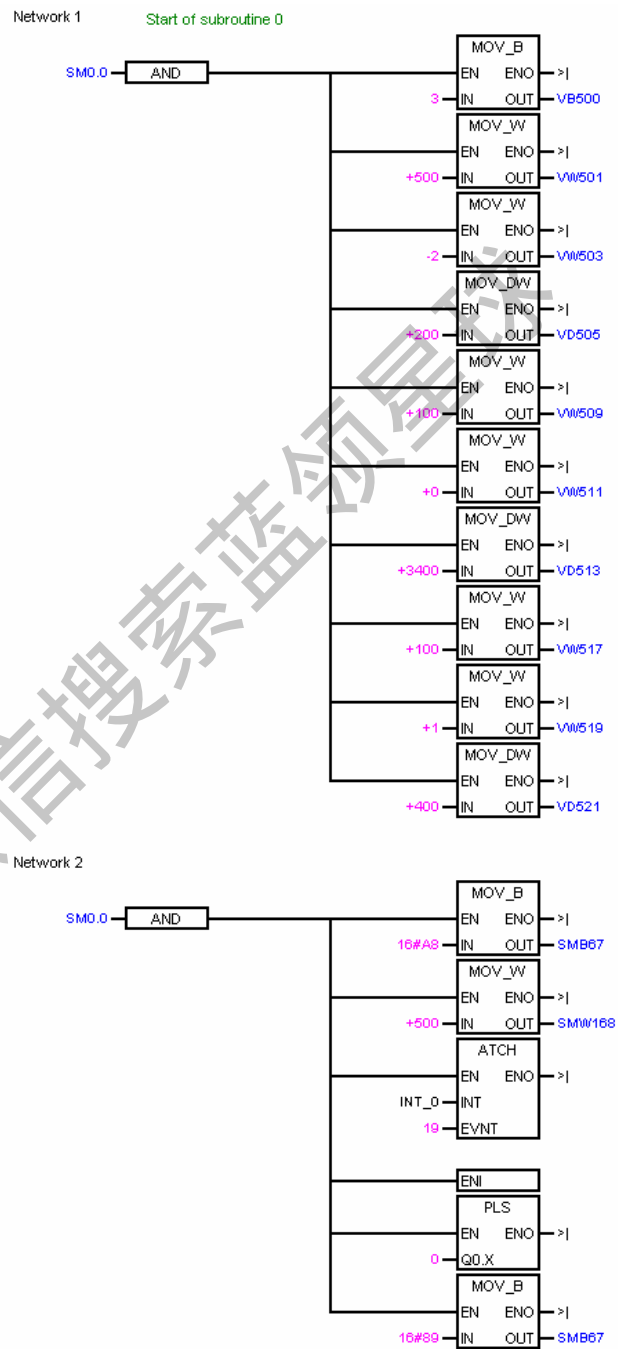
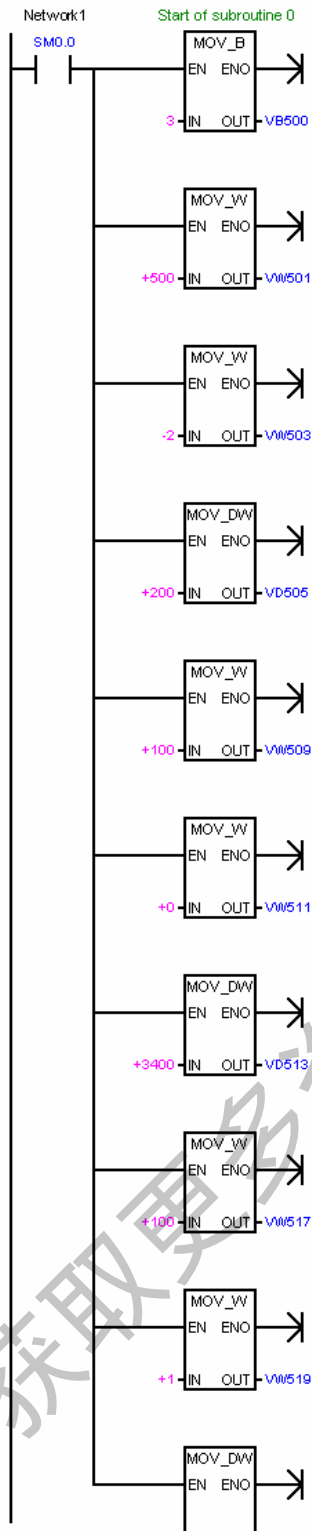
```

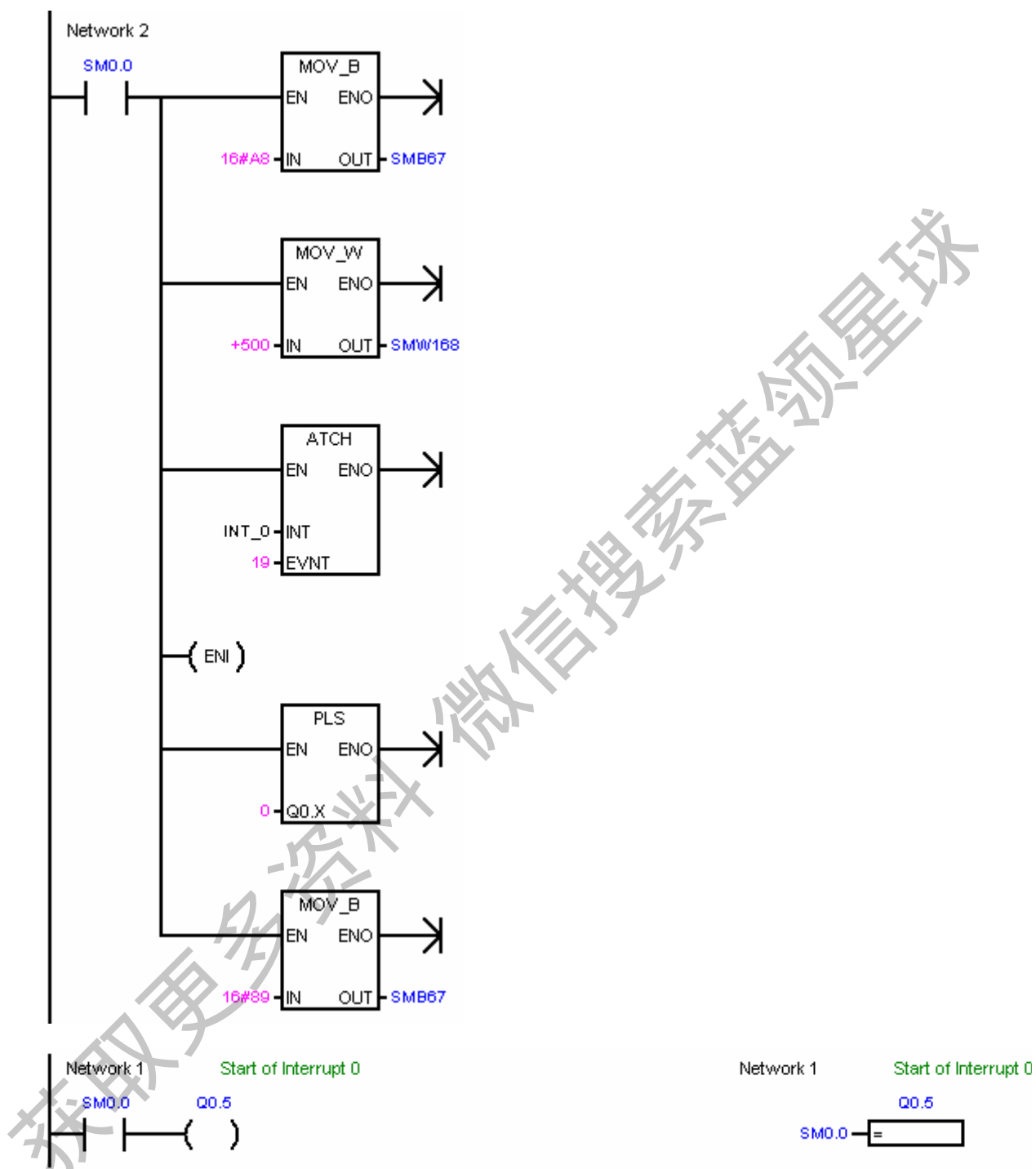
```

NETWORK 1 // 中断0开始
// PTO输出轮廓完成时, 打开输出Q0.5
LD SMO.0
= Q0.5

```







### 3.1.17 子例行程序编程

**调用指令** `[[[ENDKEEPN]]]`

本标题讨论下列主题：

- [使用子例行程序](#)
- [如何建立子例行程序](#)
- [如何终止子例行程序](#)
- [如何调用子例行程序](#)
- [嵌套和递归](#)

## 使用子例行程序

[\(返回顶端\)](#)

子例行程序帮助您对程序进行分块。主程序中使用的指令决定具体子例行程序的执行状况。当主程序调用子例行程序并执行时，子例行程序执行全部指令直至结束。然后，系统将控制返回至调用子例行程序网络中的主程序。子例行程序用于为程序分段和分块，使其成为较小的、更易管理的块。在程序中调试和维护时，您可以利用这项优势。通过使用较小的程序块，对这些区域和整个程序简单地进行调试和排除故障。只在需要时才调用程序块，可以更有效地使用 PLC，因为所械某缘蚩榭瞻芪才脞葱忻看紊琛

最后，如果子例行程序仅引用参数和局部内存，则可移动子例行程序。为了移动子例行程序，应避免使用任何全局变量 / 符号（I、Q、M、SM、AI、AQ、V、T、C、S、AC内存中的绝对地址）。如果子例行程序无调用参数（IN、OUT或IN\_OUT）或仅在LAD、FBD或STL中使用子例行程序，必须执行下列三项任务：

- 建立子例行程序
- 在子例行程序局部变量表中定义参数（如果有）
- 从适当的POU（从主程序或另一个子例行程序）调用子例行程序

当子例行程序被调用时，整个逻辑堆栈被保存，堆栈顶端被设为一，所有其他堆栈位置被设为零，控制被传输至调用子例行程序。子例行程序和调用例行程序共用累加器。由于子例行程序的使用，对累加器不执行保存或恢复操作。

### 用参数调用子例行程序

子例行程序可能包含交接的参数。参数在子例行程序的局部变量表中定义。参数必须有一个符号名（最多为 23 个字符）、一个变量类型和一个数据类型。可向子例行程序交接 16 个参数或从子例行程序交接 16 个参数。局部变量表中的变量类型域定义参数是否交接至子例行程序（IN）、交接至或交接出子例行程序（IN\_OUT）或交接出子例行程序（OUT）。下表说明子例行程序的参数类型。欲增加参数条目，将光标放在您希望增加的类型的变量类型域上（IN、IN\_OUT或OUT）。单极滑鼠右键，获得选项菜单。选择“插入”选项，然后选择“下一行”选项。在当前条目的下方会显示所选类型的另一个参数条目。

#### 调用参数类型

#### 说明

IN	参数被交接至子例行程序。如果参数是直接地址（例如 VB10），在指定位置的数值被交接至子例行程序。如果参数是间接地址，（例如 *AC1），位于指向位置的数值被交接至子例行程序。如果参数是数据常量（16#1234）或地址（&VB100），常量或地址数值被交接至子例行程序。
IN_OUT	位于指定参数位置的数值被交接至子例行程序，来自子例行程序的结果数值被返回至相同的位置。输入 / 输出参数不允许使用常量（例如 16#1234）和地址（例如 &VB100）。
OUT	来自子例行程序的结果数值被返回至指定的参数位置。常量（例如 16#1234）和地址（例如 &VB100）不允许用作输出。
TEMP	未用作交接参数的任何本地内存不得用于子例行程序中的临时存储。

局部变量表中的数据类型域定义参数的大小和格式。下表列出了参数类型。

#### 参数数据类型

#### 说明

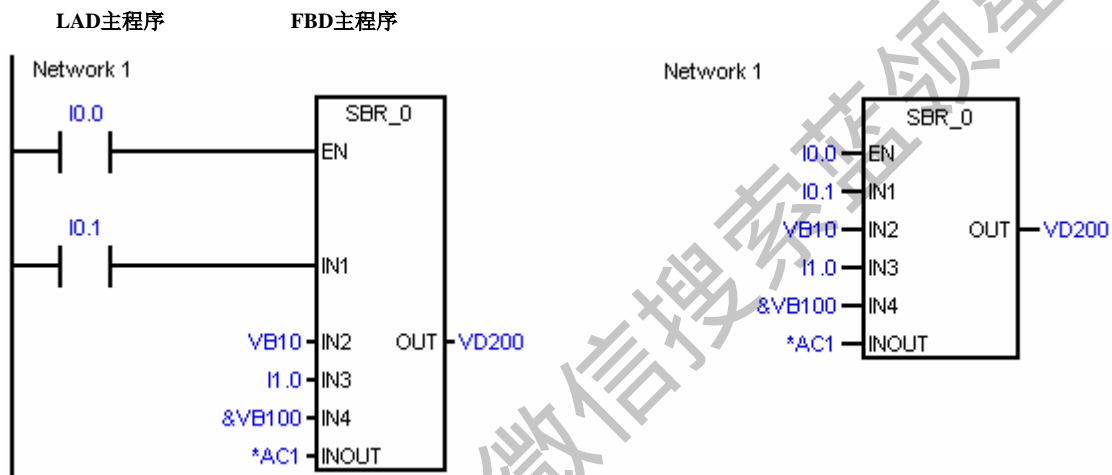
功率流	布尔功率流仅限于位（布尔）输入。该说明通知 STEP 7-Micro/WIN 该输入参数是依据位逻辑指令组合的功率流结果。布尔功率流输入必须首先在局部变量表中任何其他类型输入之前显示。只有输入参数允许这样使用。下例中的启用输入（EN）和 IN1 输入使用布尔逻辑。
布尔	该数据类型用于单位输入和输出。下例中的 IN3 是布尔输入。
字节、字、双字	这些数据类型分别识别 1、2 或 4 个字节不带符号的输入或输出参数。
整数、双整数	这些数据类型分别识别 2 或 4 个字节带符号的输入或输出参数。
实数	该数据类型识别单精度（4 个字节）IEEE 浮点数值。
字符串	此数据类型被用作字符串的四字节指针
功率流	布尔功率流只可用于位（布尔）位。此说明告诉 STEP 7-Micro/WIN，这个输入参数是基于某位逻辑指令组合的功率流结果。布尔功率流输入必须在局部变量表中最先出现，早于任何其他类型的输入。唯有输入参数可以此方式使用。下例中的启用输入（EN）输入和 IN1 输入使用了布尔逻辑。

### 子例行程序调用举例

#### 用于 SBR\_0 的局部变量表



	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
LB1	IN2	IN	BYTE	
L2.0	IN3	IN	BOOL	
LD3	IN4	IN	DWORD	
		IN		
LD7	INOUT	IN_OUT	REAL	
		IN_OUT		
LD11	OUT	OUT	REAL	
		OUT		



用Micro/WIN从LAD/FBD图形建立的STL代码在LAD、FBD或STL视图中启用显示。注释：Micro/WIN保留L内存（LB60-LB63）的四个上方字节，将其用于调用参数数据。

```

NETWORK 1
// L内存被用于保存布尔输入参数状态，
// 该参数在LAD和FBD中显示为功率流输入。这样可允许
// 本网络在LAD、FBD和STL编辑器中显示。
LD I0.0
= L60.0
LD I0.1
= L63.7
LD L60.0
CALL SBR_0 L63.7 VB10 I1.0 &VB100 *AC1 VD200

```

在STL编辑器中输入与以上显示相同的子例行程序调用的仅限 STL的范例 注释：STL程序员可使用该简化的调用程序

```


NETWORK 1
// 该网络只能在STL编辑器中显示，
// 因为被用作功率流输入的布尔参数
// 未保存至L内存。
LD I0.0
CALL SBR_0 I0.1 VB10 I1.0 &VB100 *AC1 VD200

```

#### 用于LAD和FBD

建立子例行程序和定义调用参数后，STEP 7-

Micro/WIN自动生成子例行程序调用方框指令。根据局部变量表中对该子例行程序的说明，调用指令包含输入 / 输出参数的正确

壳屠槽汀= (9)永 谐缘蚝媳 蚪 鱿衷溢噶钐髦小S 讷碾框 POU中插入子例行程序，从指令树中拖出子例行程序块  到急筷 湃育碾框 POU中。

#### 如何建立子例行程序

(返回顶端)

可采用下列一种方法建立子例行程序：

- 从“编辑”菜单，选择**插入 (Insert) > 子例行程序 (Subroutine)**
- 从“指令树”，用滑鼠右键单击“程序块”图标，并从弹出菜单选择**插入 (Insert) > 子例行程序 (Subroutine)**
- 从“程序编辑器”窗口，用滑鼠右键单击并从弹出菜单选择**插入 (Insert) > 子例行程序 (Subroutine)**

程序编辑器从先前的POU显示更改为新子例行程序。程序编辑器底部会出现一个新标记，代表新子例行程序。

此时，您可以对新子例行程序编程，或者保留子例行程序，返回您先前作业的POU位置：

- 如果您现在希望为子例行程序指定参数，您可以使用该子例行程序的**局部变量表**定义参数。

**注释：**

\*

请记住程序中每个POU都有一个独立的局部变量表。必须在选择该子例行程序标记后出现的局部变量表中为该子例行程序定义局部变量。编辑局部变量表时，必须确保已选择适当的标签。

\*\* 每个子例行程序调用的最大输入 / 输出参数限制为 16。如果您尝试下载超过该限制的程序，则会返回一则错误讯息。

- 选择子例行程序标记时，如果您希望为该子例行程序写入逻辑，在程序编辑器窗口中即可写入。
- 如果您希望对不同的POU编程，单击该POU的标签，以便在程序编辑器窗口中显示该POU。

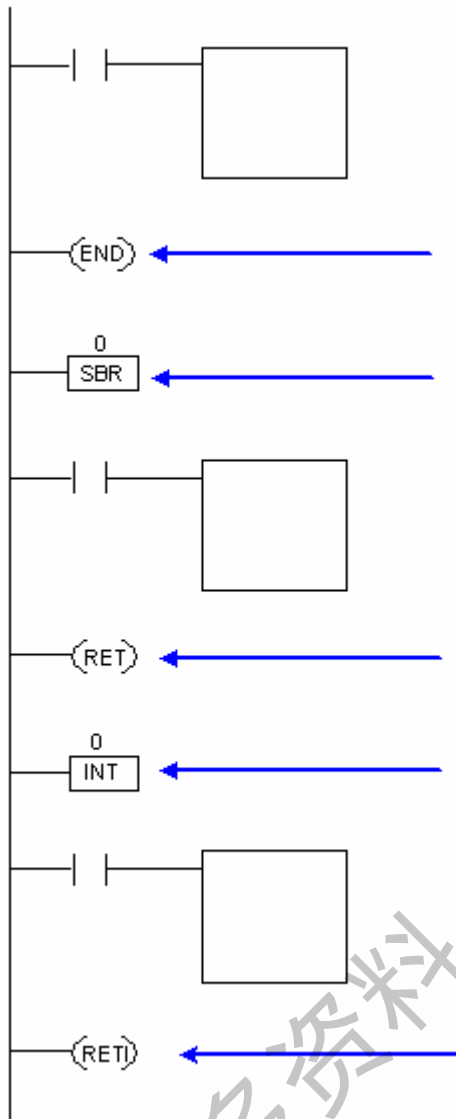
### 不使用RET指令终止子例行程序

([返回顶端](#))

在子例行程序中不得使用END（结束）指令。

编辑器自动插入无条件POU终止指令（END用于OB1，RET用于SBR，RETI用于INT）。以下显示一个范例。

箭头表示指令由Micro/WIN自动处理



### 如何调用子例行程序 (返回顶端)

插入新子例行程序并在该子例行程序的局部变量表中定义参数（如果有）后，您可在程序的另一个 POU 中放置一个子例行程序调用。（您可以从 OB1、另一个子例行程序或中断例行程序调用子例行程序；您不能从子例行程序本身调用子例行程序。）

#### LAD、FBD

对于 LAD 和 FBD 程序，在子例行程序局部变量表中为该子例行程序指定参数后，会生成一个定制调用方框指令。该调用指令自动包括子例行程序输入和输出参数的正确数目和类型。

欲在 LAD 或 FBD 程序的 POU 中插入调用指令：

1. 打开程序编辑器窗口中所需的 POU，滚动至您希望插入子例行程序调用的网络处。
2. 在指令树中，双击打开的“子例行程序”文件夹。您可将适当的调用指令从指令树拖放至程序编辑器中的正确的网络单元格中，或将光标放在程序编辑器中的单元格上，然后双击指令树中的调用指令。
3. 编辑程序中的调用指令参数并为每个参数指定有效操作数。有效操作数为：内存地址、常量、总体符号以及调用指令被放置的 POU 中的局部变量（并非被调用子例行程序中的局部变量）。

**注释：**如果您在子例行程序中插入一个调用指令，然后修改该子例行程序的局部变量表，调用指令则无效。您必须删

#### STL

欲在 STL 程序中插入调用，使用 [调用指令](#)。

调用子例行程序时，保存整个逻辑堆栈，堆栈顶值被设为 1，其他所有堆栈位置均设为 0，控制转移至被调用的子例行程序。该子

例行程序完成后，用调用时保存的数值恢复堆栈，控制返回调用例行程序。  
子例行程序和调用例行程序共用累加器。不因使用子例行程序对累加器执行保存或恢复操作。

### 嵌套和递归 (返回顶端)

程序中总共可有64个子例行程序（CPU 226XM可有128个子例行程序）。在主程序中，您可以嵌套子例行程序（在子例行程序中放置子例行程序调用指令），最大嵌套层数为8。您无法从中断例行程序嵌套子例行程序。子例行程序调用无法被放置在任何从中断例行程序调用的子例行程序中。递归（子例行程序调用自身）不被禁用，但您在子例行程序中使用递归时应小心。

## 3.1.18 中断例行程序编程

中断由事件驱动。在启动中断例行程序之前，必须使中断事件与发生该事件时您希望执行的程序段建立联系。使用“附加中断”指令（ATCH）建立中断事件（由**中断事件号码**指定）与程序段（由中断例行程序号码指定）之间的联系。将中断事件附加于中断事件。如果您使用全局禁用中断指令禁用所有的中断，中断事件的每次出现均被排队等候，直至使用全局启用中断指令重新启用中断。使用“拆卸中断”指令（DTCH）可拆卸中断事件与中断例行程序之间的联系，从而禁用单个中断事件。“拆卸”指令使中断返回正常操作状况。

### 了解S7-200如何处理中断例行程序

作为对关联的内部或外部事件的应答，执行中断例行程序。一旦中断例行程序的最后一条指令被执行，控制被返回至主程序。您可以用执行“从中断指令有条件返回”指令（CRETI）的方法退出例行程序。

### 使用中断例行程序指南和限制

中断程序为特殊内部或外部事件提供快速反应。您应当优化中断例行程序，执行某项具体任务，然后将控制返回至主程序。通过将中断例行程序保持为短小和简明扼要，可加快执行的速度，使其他程序不会受到长时间的延误。如果未能做到这一点，无法预料的情形可能导致主程序控制的装置出现非正常操作状况。

### 限制

您不得在中断例行程序中使用DISI、ENI、HDEF、LSCR和END指令。

### 中断的系统支持

由于接点、线圈和累加器逻辑可能受中断的影响，系统保存和重新载入说明累加器和指令操作状态的逻辑堆栈、累加器寄存器（SM）。这样可避免因分支至中断例行程序和从中断例行程序分支而导致的主程序中断。

### 在主程序和中断例行程序之间共享数据

您可以在主程序和一个或多个中断例行程序之间共享数据。因为无法预测S7-200何时可能生成中断，最好限制中断例行程序和程序中其他位置使用的变量数目。由于主程序中指令执行被中断事件中断时，并且不盖写程序其他位置使用的的数据。

您可以使用各种编程技巧，以确保在主程序和中断例行程序之间正确地共享数据。这些技巧限制存取共享内存位置的方法，或者使用共享内存位置预防出现指令序列中断。

- 对于共享单一变量的STL程序：如果共享数据是单字节、字或双字变量，且程序在STL中写入，则可用在非共享内存位置或累加器中存储共享数据操作数的直接数值的方法，确保正确的共享存取。
- 对于共享单一变量的LAD程序：如果共享数据是单字节、字或双字变量，且程序在LAD中写入，则可用建立仅使用“贫 敲噶眷 MOVB、MOVW、MOVD、MOVR）存取共享内存位置常规的方法，确保正确的共享存取。尽管很多LAD指令由STL指令的可中断序列组成，这些“移动”指令却是由单个STL指令组成，此类指令的执行不受中断事件的影响。
- 对于共享多个变量的STL或LAD程序：如果共享数据由各种相关的字节、字或双字组成，则可使用中断禁用/启用指令（DISI和ENI）控制中断例行程序的执行。在主程序中共享内存位置操作即将开始的点，禁用中断。一旦所有影响共享位置的措施均完成后，重新启用中断。在中断被禁用的时间内，不得执行中断例行程序，因此无法存取共享内存位置；但是，此种方法会导致对中断事件的延迟应答。

### 从中断例行程序调用子例行程序

您可以从中断例行程序调用一个子例行程序嵌套级别。在被调用的中断例行程序和子例行程序之间共享累加器和逻辑堆栈。

### S7-200支持的中断类型

S7-200支持以下中断例行程序类型：

- 通讯端口中断：S7-200生成允许程序控制通讯端口的事件。
- I/O中断：S7-200生成用于各种I/O状态不同变化的事件。这些事件允许程序对高速计数器、脉冲输出或输入的升高或降低状态作出应答。
- 时基中断：S7-200生成允许程序按照具体间隔作出应答的事件。

### 通讯端口中断

可用程序控制S7-

200的串行通讯端口。此种操作通讯端口的模式被称作自由端口模式。在自由端口模式中，程序定义波特率、每个字符的位、奇偶校验、流控、商、敲、邮、铡、焙、汀、按、涸、澈、卸、希、谐、缘、蚩、颀、耐、丁、O、呢、榻、瓮、卧、摹、按、浜、徒、邮、剿、澈、噶、唯

### I/O中断

I/O中断包括上升/下降边缘中断、高速计数器中断和脉冲链输出中断。S7-

200可生成输入（I0.0、I0.1、I0.2或I0.3）上升和/或下降边缘中断。可为每个此类输入点捕获上升边缘和下降边缘事件。这些上陆、当、吹、凳、录、捎、糜、诘、碛、驹、谱、录、(5)北、甌、育、12、创、淼、淖、纯、航

高速计数器中断允许您对诸如以下之类的条件作出应答：当前值达到预设值，可能与转轴旋转方向逆转对应的计数方向的改变或

计数器外部重设。每种此类高速计数器事件均允许针对按照可编程逻辑控制器扫描速度控制的高速事件采取实时措施。脉冲链输出中断发出输出预定数目脉冲完成的立即通知。脉冲链输出的最常用用法是步进器马达控制。

您可以用将中断例行程序附加在相关I/O事件上的方法，启用上述每种中断。

#### 时基中断

时基中断包括定时中断和计时器T32/T96中断。您可以使用定时中断基于循环指定需要采取的措施。循环时间被设为从1毫秒至255毫秒每1毫秒递增一次。您必须在SMB34中将定时中断的循环时间设为0，在SMB35中将定时中断的循环时间设为1。

每次计时器失效时，定时中断事件将控制传输给适当的中断例行程序。通常您使用定时中断控制模拟输入取样或定期执行PID环

您可将中断例行程序附加在定时中断事件上时，则启用定时中断，且计时开始。在附加的过程中，系统捕获循环时间数值，因此其后对SMB34和SMB35所作的改动不会影响循环时间。欲改动循环时间，您必须修改循环时间数值，然后将中断例行程序重新附加在定时中断事件上。重新附加时，定时中断功能从以前的附件中清除所有的累计时间，并开始用新数值计时。

时间中断被启用后，则持续运行，每当指定的时间间隔失效时，执行附加中断例行程序。如果您退出RUN（运行）模式或分离定时中断，定时中断被禁用。如果全局禁用中断指令被执行，定时中断继续进行。每次定时中断出现均排队等候（直至中断被启用或队列已满）。

计时器T32/T96中断允许对指定时间间隔完成及时作出应答。仅在1毫秒分辨率打开延迟（TON）和关闭延迟（TOF）计时器T32/T96中支持此类中断。否则T32和T96按照正常情况作业。一旦中断被启用，在S7-200中执行的正常1毫秒计时器更新的过程中，当用计时器的当前值等于预设时间数值时，即执行附加中断例行程序。您用将中断例行程序附加至T32/T96中断事件的方法，启用此类中断。

#### 中断优先级别和入队等候

S7-

200在中断各自的优先级别组内按照先来先服务的原则为中断提供服务。在任何时刻，只能执行一个用户中断例行程序。一旦桓鲋卸侠 谐缘蚩 贾葱校 蛞恢澈葱兄镣瓿伞 2 荒馨涣砒桓鲋卸侠 谐缘蛟は扰眈眨 词故款 晒畔燃伺鸬睦 谐缘颀U 洼 破砒桓鲋卸鲜猓(5) 闹卸先父拥却 惹

#### 每个中断队列最大条目数

队列	CPU 221、CPU 222 CPU 224	CPU 224 XP CPU 226和CPU 226XM
通讯队列	4	
I/O中断队列	16	
定时中断队列	8	

一般而言，出现的中断数目会超出队列能够容纳的数目。因此，队列溢出内存位（识别已经丢失的中断事件类型）由系统保持。下表显示中断队列溢出位。您应当仅在中断例行程序中使用这些位，因为当队列排空时这些位会被重设，控制被返回主程序。

#### 中断队列溢出位

说明（0 = 无溢出，1 = 溢出） SM位

通讯队列	SM4.0
I/O中断队列	SM4.1
定时中断队列	SM4.2

可采用下列一种方法建立中断例行程序：

- 从“编辑”菜单，选择**插入（Insert） > 中断（Interrupt）**。
- 从指令树，用鼠标右键单击“程序块”图标并从弹出菜单选择**插入（Insert） > 中断（Interrupt）**。
- 从“程序编辑器”窗口，从弹出菜单用鼠标右键单击**插入（Insert） > 中断（Interrupt）**。

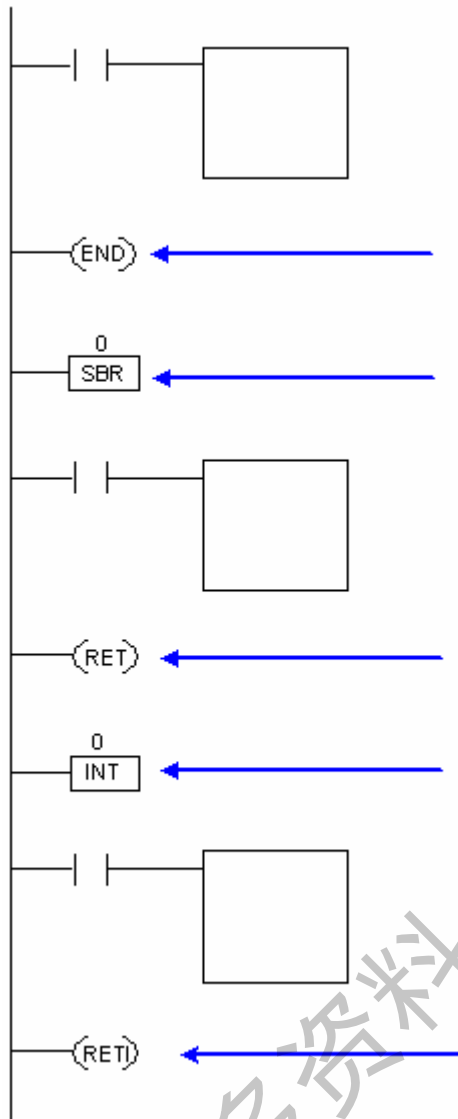
程序编辑器从先前的POU显示更改为新中断例行程序。在程序编辑器的底部会出现一个新标记，代表新中断例行程序。

一个程序中总共可有128个中断。在各自的优先赋值范围内，PLC采用先来先服务的原则为中断提供服务。在任何时刻，只能执行一个用户中断例行程序。一旦一个中断例行程序开始执行，则一直执行至完成。不能被另一个中断例行程序预先排空，即使是更高优先级别的例行程序。正在处理另一个中断时发生的中断入队等待处理。

#### 注释：

编辑器自动插入无条件POU终止指令（END用于OB1，RET用于SBR，RETI用于INT），以下显示一个范例。

箭头表示指令由Micro/WIN自动处理



另请参阅：  
[队列溢出错误（中断）](#)  
[中断事件时间间隔赋值](#)  
[中断事件优先级列表](#)  
[局部变量表](#)

**SIMATIC编程中断指令**

[LAD](#)  
[FBD](#)  
[STL](#)

**IEC编程中断指令**

[LAD](#)  
[FBD](#)

### 3.1.19 局部变量表编程

本部分讨论下列主题：  
[您的程序需要使用局部变量吗？](#)  
[理解局部变量](#)  
[局部变量的说明类型](#)

[局部变量数据类型检查](#)  
[如何隐藏 / 显示局部变量表](#)  
[如何在局部变量表中赋值](#)  
[局部变量表举例](#)

### 您的程序是否需要使用局部变量？

[返回顶端](#)

使用局部变量有两种原因：

- 您希望建立不引用绝对地址或全局符号的可移动子例行程序。
- 您希望使用临时变量（说明为 TEMP 的局部变量）进行计算，以便释放 PLC 内存。

如果以上说明对您的情形不适用，您无须使用局部变量；您可以在符号表（SIMATIC）或全局变量表（IEC）中定义符号数值，  
 械姆 攀 刀丁邇 直淞俊

### 理解局部变量

[返回顶端](#)

您可以使用程序编辑器的局部变量表指定对个别子例行程序或中断例行程序独特的变量。

局部变量可用作传递至子例行程序并可用于增加子例行程序的移动性或重新使用子例行程序的参数。

程序中的每个 POU 都有自己的局部变量表，配备 64 个字节的 L 内存。这些局部变量表允许您定义具有范围限制的变量；局部变量  
 辉消 8.帽淞康 POU 中才有效。相反，在每个 POU 中均有效的全局符号只能在符号表 / 全局变量表中定义。当您为全局符号和  
 植勘淞渴褂孟嗤 姆 芭 保g INPUT1)，定义局部变量的 POU 中的局部定义优先，全局定义用于其他 POU。

在局部变量表中赋值时，您指定说明类型（TEMP、IN、IN\_OUT 或 OUT）和数据类型（请参阅 SIMATIC 和 IEC 1131-  
 3 数据类型），但不指定内存地址；程序编辑器自动在 L 内存区中为所有的局部变量指定内存位置。

局部变量表符号地址分配会将一符号名与存储有关数据值的 L 内存地址关联。局部变量表不支持向符号名直接赋值的符号常量  
 (这在符号或全局变量表中是许可的)。



**提示：** PLC 不会将局部变量数据值初始为零。您必须在程序逻辑中初始化您使用的局部变量。

### 局部变量的说明类型

[返回顶端](#)

对局部变量赋值指定的类型取决于在其中赋值的 POU。主程序（OB1）、中断例行程序和子例行程序可使用临时（TEMP）变量  
 V 挥性谥葱铎械保 偈北淞坎坎指茫 槽葱型瓠珊瑚 偈北淞靠杀恢甌础 W 永 谱缘蛞部墙褂玫瑰貌问 IN、IN\_OUT、OUT）。

#### 说明类型

#### 说明

IN 调用 POU 提供的输入参数。

OUT 返回调用 POU 的输出参数。

IN\_OUT 数值由调用 POU 提供的参数，由子例行程序修改，然后返回调用 POU。

TEMPORARY 临时保存在局部数据堆栈中的临时变量。一旦 POU 完全执行，临时变量数值则无  
 法再用。在两次 POU 执行之间，临时变量不保持其数值。

### 局部变量数据类型检查

[返回顶端](#)

将局部变量作为子例行程序参数传递时，在该子例行程序局部变量表中指定的数据类型必须与调用 POU 中数值的数据类型相匹配

举例：

您从 OB1 调用 SBR0，将称为 INPUT1 的全局符号用作子例行程序的输入参数。

在 SBR0 的局部变量表中，您已经将一个称为 FIRST 的局部变量定义为输入参数。

当 OB1 调用 SBR0 时，INPUT1 数值被传递至 FIRST。

INPUT1 和 FIRST 的数据类型必须匹配。

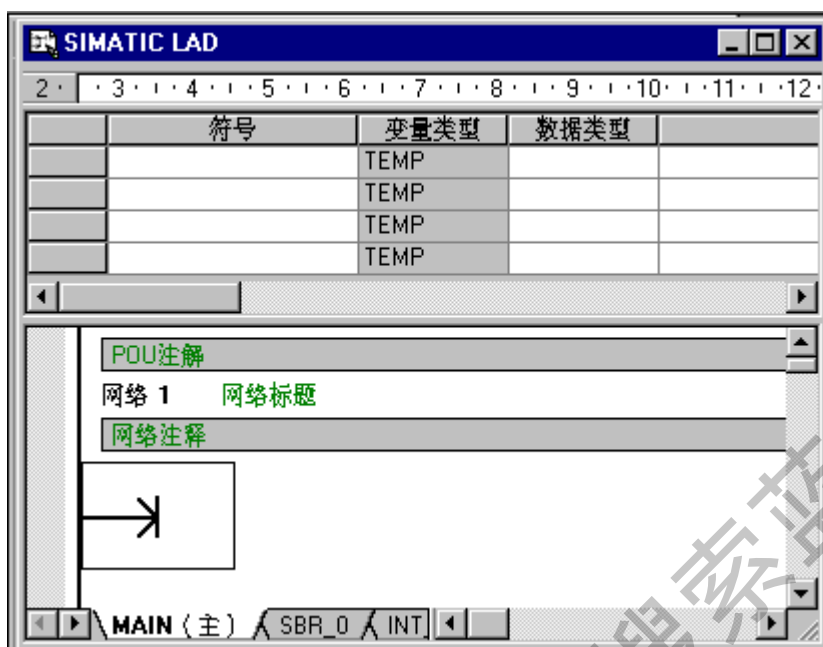
如果 INPUT1 是实数，FIRST 也是实数，则数据类型匹配。如果 INPUT1 是实数，但 FIRST 是整数，则数据类型不匹配，

挥芯勒 苏庖淮砦媳 缘蚤拍馨噤裔

### 检视 / 隐藏局部变量表

[返回顶端](#)

如果您将水平分裂条拉至程序编辑器窗口的顶部，则局部变量表不再显示，但仍旧存在。将分裂条下拉即可再次显示局部变量表。



#### 如何在局部变量表中赋值

[返回顶端](#)

##### 注释：\*

在程序中使用局部变量之前，在局部变量表中赋值最为有效。在程序中使用符号名时，程序编辑器首先检查适当 POU 的局部变量表，然后检查符号表 / 全局变量表。

如果符号名在这两处均未定义，程序编辑器则将其视为未定义的全局符号；此类符号用绿色波浪状下划线标明。程序编辑器不会自动重新读取局部变量表并对您的程序逻辑做出修正。如果您后来进行了定义该符号名的数据类型分配（在局部变量表中），您必须以手动方式在该符号名前插入一个井号（#），例如：#UndefinedLocalVar（在程序逻辑中）。

\*\* 每个子例行程序调用的输入 / 输出参数的最大限制是 16，如果您尝试下载的程序超过此一限制，会返回一则错误信息。

#### 如何输入第一个局部变量赋值

欲在局部变量表中赋值，请遵循以下步骤。

1. 确保正确的 POU 在程序编辑器窗口中显示，如有必要，单击所需的 POU 标记。（因为每个 POU 都有自己的局部变量表）
2. 如果局部变量表处于隐藏状态，下拉水平分裂条，显示局部变量表。（请参阅 [检视 / 隐藏局部变量表](#)。）
3. 为需要定义的变量选择具有正确说明类型的行，在“名称”域中为该变量键入一个名称。（如果您在 OB1 或中断例行程序中赋值，局部变量表只包含 TEMP 变量。如果您在子例行程序中赋值，局部变量表包含 IN、IN\_OUT、OUT 和 TEMP 变量。）您不需要在局部变量表中的变量名前加井号。井号只用在程序代码中的局部变量前。

##### 注释：

\* 局部变量名最多可包含 23 个字母数字字符和下划号，也可包含扩展字符（ASCII 128 至 ASCII 255）。第一个字符只能是字母或扩展字符。将关键字用作符号名属于非法，名称的第一个字符是数字，或名称包含非字母数字字符。

\*\* 局部变量表变量名被下载并存储于 CPU 内存中，使用较长的变量名可能减少用于存储程序的内存空间。

4. 在“数据类型”域中单击鼠标指针，并使用列表框为局部变量选择适当的数据类型。

##### 注释：

当您为局部变量指定为子例行程序的参数时，您必须保证为局部变量指定的数据类型不与子例行程序调用中使用的操作数发生冲突。（请参阅上文中的 [数据类型检查](#)。）

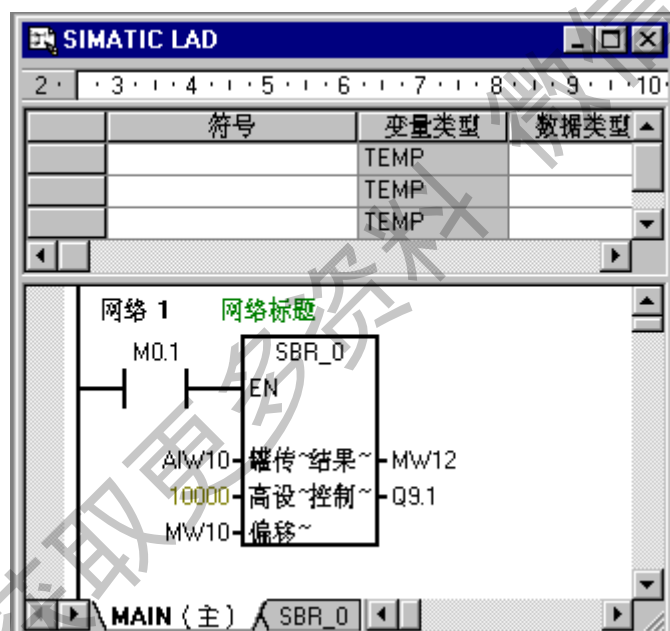
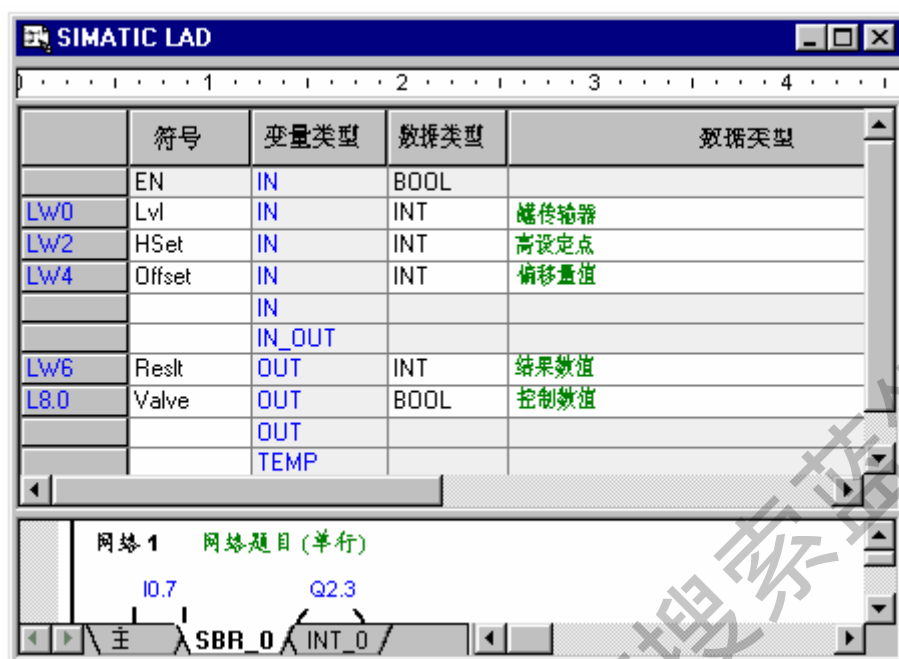
为“名称”和“数据类型”域提供数值后，程序编辑器会自动为局部变量指定 L 内存地址。

#### 如何在局部变量表中输入附加赋值

对于 OB1 和中断例行程序，局部变量表显示一组已被预先定义为 TEMP 变量的行。在 OB1 或中断例行程序中，只能使用这种说明类型。欲在表中增加更多行，只需单击最后一行中的一个单元格，然后使用 ENTER 键移过该行并向下移动，即自动生成一个新行。对于子例行程序，局部变量表显示按照以下顺序预先定义说明类型的一组行：IN、IN\_OUT、OUT 和 TEMP。您不能改变该顺序。局部变量在该表中的顺序必须符合当您为子例行程序调用指令进行操作数赋值时对应的操作数顺序。如果您希望增加附加局部变量，必须用鼠标右键单击现有行，并使用弹出菜单插入与单击行类型相同的另一局部变量。选择 **插入 (Insert) > 行 (Row)**，在所选行的上方插入新行，或选择 **插入 (Insert) > 行下方 (Below Row)**，在所选行下方插入新行。



局部变量表举例  
[返回顶端](#)



另请参阅:

[子例行程序](#)  
[SIMATIC和IEC 1131-3数据类型](#)  
[关键字](#)

### 3.1.20 使用数据块在PLC V内存中存储数据

使用下列一种方法存取数据块：



- 单击浏览器上的“数据块”按钮。
- 选择菜单命令**检视 (V) > 数据块 (D)**。
- 打开**指令树**中的“数据块”文件夹，然后双击某块页图标。

通过插入新数据块页标记，将您的数据块V内存赋值分成多个功能组：

- 单击数据块窗口，然后选取菜单命令**编辑(E) > 插入(I) > 数据块(D)**
- 在**指令树**中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取**插入(I) > 数据块(D)**
- 用鼠标右键单击数据块窗口，然后在弹出菜单中选取**插入(I) > 数据块(D)**
- 标记的最大数目为128。如果您使用向导，有关标记会被自动创建以支持向导功能。您可以创建的最大数目为（由MicroWin自动创建的标记数目）。请使用Windows剪贴板合并标记数据；方法为使用剪切和粘贴由一个标记转移到另一块页。

重新命名和保护数据块页标记：

- 在**指令树**中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取**重新命名**。您也可以在指令树内直接重新命名数据块页，方法为单击该标记页名称两次（动作要慢一些，以免解释成双击）；然后编辑该标记名。数据块编辑器提供相同的重新命名功能，方法为用鼠标右键直接单击该标记名。
- 在**指令树**中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取**属性**。由此，您可以重新命名该数据块标记和指定作者。属性对话框的**保护**标记令您能够用密码保护单个数据块标记。受保护的标记会显示锁图标。数据块编辑器向导会创建不能重新命名且包含只读数据值的受保护标记。

导入和导出数据块数据至ASCII文本文件

- [由ASCII文本文件导入数据块](#)
- [由ASCII文本文件导出数据块](#)

本标题讨论下列主题：

[在数据块中进行地址和数据值赋值](#)

[数据块举例](#)

[在数据块编辑器中使用“剪切”、“复制”和“粘贴”](#)

[使用ASCII常量的限制](#)

[理解和解决错误](#)

[将数据块下载至PLC](#)

[从PLC上载数据块](#)

[上载后保留数据块格式](#)

[从CPU RAM建立数据块](#)

在数据块中进行地址和数据值赋值

[返回顶端](#)

数据块仅允许您对V内存进行初始数据或ASCII字符赋值。您可以对V内存的字节 (V或VB)、字 (VW) 或双字 (VD) 赋值。注解（前面带双正斜线//）是选用项目。

- 数据块的第一行必须包含一个明确地址赋值（绝对或符号地址），其后的行可包含明确或隐含地址赋值。当您在单地址赋值后键入多个数据值或键入仅包含数据值的行时，由编辑器指定隐含地址赋值。编辑器根据先前的地址分配及数据值大小（字节、字或双字）指定适当的V内存数量。
- 数据块编辑器是一种自由格式文本编辑器，对特定类型的信息无具体域。键入一行后，按ENTER键，数据块编辑器格式校云氩妨小(6)尊(9)3.裸徊痘 V内存地址) 并重新显示行。数据块编辑器接受大小写字母并允许使用逗号、制砑 蚩崧痲 魁 小泛褪 葵抵 涑姆指解
- 在完成一赋值行后按CTRL-ENTER键组合，会令地址自动增加至下一个可用地址。

[常量格式详细说明](#)

[数据块一般规则](#)

```

数据块
2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18
VBO  248      //明确地址赋值：VBO数据值：248。

VB1  249, 250, 251 //单行中多个数据值。
           //隐含地址赋值：
           //VB2包含数据值250。
           //VB3包含数据值251。

VB4  252      //不能使用先前指定的地址（VBO-VB3）。

      253, 254, 255 //无明确地址赋值的行。
           //数据值隐含指定给VB5、VB6、VB7。

VW8  256, 257   //新数据类型（字）。隐含将数据值
           //257指定给V内存VB10-VB11。

X  65536      //地址赋值不能与先前的明确赋值发生冲突。
           //数据值65536要求双字数据类型（VD内存），
           //但上一个明确地址赋值
           //是字内存（VW8）。编辑器标志错误。

//您可以从地址中省略尺寸规格。
//然后您可以对随后的数据使用隐含地址赋值，
//不必考虑数据的尺寸。

V12  258      //字值明确指定给V12-V13。
      65537   //双字值隐含指定给V14-V17。

```

数据块举例

[返回顶端](#)

直接地址和数值

```

数据块
//
//数据块注解
//
VB0    255          //字节值从VB0开始
VW2    256          //字值从VW2开始
VD4    700.59      //双字真值从VD4开始
VB8    -35         //字节值从VB8开始
VW10   16#A        //字值HEX从VW10开始
VD14   146879     //双字值从VD14开始
VW20   2, 4, 16, 32, 64 //字值表从VD14开始
        -2, -4, -16, -32, -64 //扩展至多行的数值不能出现在列中
//第一列为内存地址保留
VB45   'Up'        //起始于VB45的双字节ASCII字符串
VW90   65535       //起始于VW90的字值
V100   255, 'abcd', 65535, 1.0 //起始于VB100的未定义数值：可混合不同尺寸
VB110  255         //与输入B110 255相同
VW120  256         //与输入W120 256相同
VD130  700.59     //与输入D130 700.59相同
V140   255, 'abc', 65535, 1.0 //与输入140相同：必须从列1开始
VW150  2#1010101010101010 //二进制字值
VD152  2#1100110011001100110011001100 //二进制双字值

```

符号地址和符号数值

符号表			
符号	地址	注解	
1	符号地址	VB1	符号地址
2	符号常数	200	符号常数

用户1 / POU符号 /

```

数据块
VB0    100
符号地址 符号常数 //为VB1指定一个200数值
用户1 /

```

替代二进制输入方法（上窗口）和结果二进制格式（下窗口）

```

数据块
VB0    2 1 0 true false off on
用户1 /

```



```

数据块
VB0    2#10010100 //按Enter键后二进制模式
用户1 /

```

在数据块编辑器中使用“剪切”、“复制”和“粘贴”  
[返回顶端](#)

当在数据块中执行剪切、复制、粘贴时，重要的是通过鼠标单击设置适当的上下文：

如果您单击指令树中的数据块标记页  图标，或者单击数据块窗口中的标记名 ，  
 则 霰县且成喜僮鳌 T 洼松舷挛闹杏檬孟暖壹 七 颯 突嵯允咀饕糜谩 黎 菘楸县且车牟僮鞞 霾说 7  
 如果您单击数据块编辑器中的文字区域，那么您将在数据块窗口中已选定(突出显示)的文字上操作。在此上下文中用鼠标右  
 七 颯 突嵯允咀饕糜谩 黎 菘楸县且衬谔 《 刁 淖稚系牟僮鞞 霾说 7

### 使用ASCII常量的限制

[返回顶端](#)

有效的ASCII赋值：

唯有字节地址(V或VB)能够与长ASCII常量(使用单引号或双引号格式)合用：

VBO 'A'、VWO 'AB'、VDO 'ABCD'

就3和5或更多字节而言，您必须使用V或VB地址前缀 VBO 'ABC'、VBO 'ABCDE'、VBO 'ABCDEFGH IJK'


双引号ASCII常量字符串格式存储一个前导长度字节和字符串 VBO "A"、VBO "AB"、VBO "ABC"、VBO "ABCD"、VBO "ABCDEFGH IJK"

[常量格式详细说明](#)

### 理解和解决错误

[返回顶端](#)

一旦在包含错误的行尾按ENTER键，立即会在数据块左页边显示输入错误。您必须纠正全部输入错误，才能成功地编译。

 vbbo 255  
 VBO 300

// (红色文本 - 非法语法错误) 应为vb0

// (红色曲线下划线 - 非法使用错误) 一个字节最大为255

引起输入错误的条件包括：

- 指定错误内存区 (V是唯一允许使用的内存区)
- 当数据值实际要求较大的尺寸时(例如，数据值256过大，无法在VB地址中存储—要求使用VW地址)，在地址赋值兄付 骋荒洼姆叽纾 卫纸译厝郑
- 在一行中使用错误序列：在数据值之后(而不是在数据值之前)键入内存地址
- 使用非法语法或无效数值
- 尝试使用符号，而不是使用绝对V内存地址(数据块中不允许使用符号)
- 未能适当地指定注解(双正斜线必须位于注解之前：//注解样本)如果数据块是现用窗口，您可以使用菜单命令 **PLC > 编译 (Compile)** 编译数据块。如果数据块不是现用窗口，您依然可以编译数据块：使用菜单命令 **PLC > 全部编译 (Compile All)**。

编译数据块时，如果编译程序发现错误，会在“输出窗口”显示错误。将光标置于“输出窗口”中的错误讯息上，双击该讯息  
 谿 崧椽翱溢那允境趵刚小  
 仅在编译后显示的错误包括：

编译数据块时，如果编译程序发现错误，会在“输出窗口”显示错误。将光标置于“输出窗口”中的错误讯息上，双击该讯息，  
 在数据块窗口中显示出错行。  
 仅在编译后显示的错误包括：

- 重复地址赋值(例如，如果您输入“VB1 249, 250”之类的行，则是对VB2进行250隐含赋值—您不得在别处对地址VB2进行其他不同的数据值赋值)
- 地址重叠(例如，如果您为VD0指定一个类似65536的双数值，则不得再对V1、V2或V3指定其他赋值，因为这些数值已被使用，是以VD0开始的双字的一部分)

### 将数据块下载至PLC

[返回顶端](#)

如果要编辑数据块，则需将数据块 **下载** 至PLC。只有在修改的数据块下载后您的编辑才会生效。  
 欲节省空间，您可以将您不希望下载至PLC的信息切换为打开/关闭。

### 从PLC上载数据块

[返回顶端](#)

您必须在STEP 7-Micro/WIN中打开一个项目，才能 **上载** 数据块。

如果PLC中的数据块与您打开的项目中的数据块不匹配(或者您打开的项目中没有数据块)，您只能上载该PLC的数据块地址  
 糠趾褪 莼挡糠郑 荒茆显刈13.袞 5 卍芬宰纸谿 蹈裕缴显兀 莼狄允 聘裕缴显兀ò 匕魏卧 模 H纛 谿】占  
 ，您可以将您不希望下载至PLC的信息切换为打开/关闭。

如果CPU中的数据块包含由Micro/WIN 4.0(或较新版本)下载的标记信息，那么由Micro/WIN

4.0的上载将会重新载入此标记结构。如果某数据块曾由较早的Micro/WIN版本下载且无标记信息，那么由Micro/WIN 4.0的上载将会将所有赋值放在同一个标记中。

#### 上载后保留DB格式

##### [返回顶端](#)

当您上载数据块时，下载至PLC的数据块条目格式被保留。如果您在一行中输入多个条目，当您上载时，格式不被保留。每个条目在一个不同的行中显示。例如，如果您在数据块中输入VB0 20、30、40、50并上载，格式更改为：

VB0	20
VB1	30
VB2	40
VB3	50

#### 从CPU RAM建立数据块

##### [返回顶端](#)

选择**PLC > 从RAM建立数据块 (Create Data Block from RAM)** 菜单命令，将CPU V内存保存至EEPROM。从CPU读取V内存当前值，然后将V内存图像作为数据块下载。

#### 注释：

进行此操作时，CPU必须处于STOP（停止）模式。

倘若CPU

RAM内有已修改的V内存地址，并且这些地址在上一次数据块下载中未得到指定，那么在您执行“从RAM建立数据块”命令时，这一研需牡牡 司坊岂恢付ù 恒鲂碌谋县敲 g 罐LC\_DATA1)。当程序的执行向新的V地址写入数据时，或当状态图被用于修改碌膺地址时，此情形就可能发生。

#### 另请参阅：

[PLC电源中断后的数据保留](#)

[常量格式详细说明](#)

[通讯配置](#)

[PLC类型选择](#)

[系统块 \(PLC配置\)](#)



[错误讯息](#)

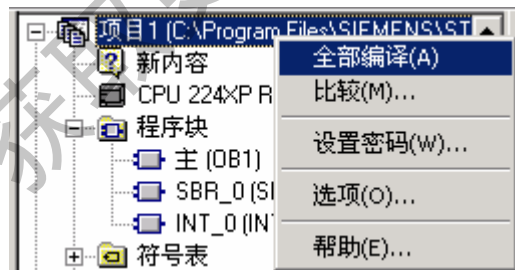
[入门指南目录](#)

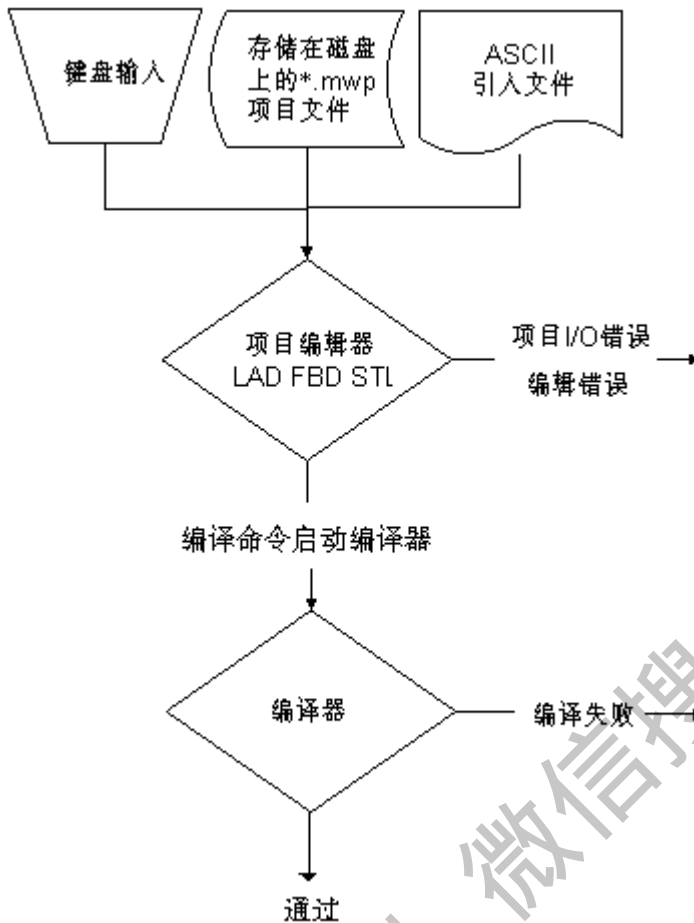
[应用程序用户参考手册](#)

## 3.1.21 编译项目

可使用下列一种方法启动STEP 7-Micro/WIN项目编译程序：

- 单击“编译” 按钮或选择菜单命令**PLC (PLC) > 编译 (Compile)**，编译现用窗口（程序块或数据块）。
- 单击“全部编译” 按钮或选择菜单命令**PLC (PLC) > 全部编译 (Compile All)**，编译全部项目元件（程序块、数据块和系统块）。
- 用鼠标右键单击**指令树**中的某个文件夹，然后由弹出菜单中选取编译命令。项目、程序块文件夹、系统块文件夹及数松榭募 卸加斜喊阮 睢





设置PLC类型 项目文件I/O错误 LAD/FBD编辑器错误 全部STEP 7-Micro/WIN编译器错误均列在输出窗口中。双击错误，编辑器将滚动至错误位置。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译器。

另请参阅：  
[应用程序用户参考手册](#)  
[入门指南目录](#)

[打开项目](#) [范围检查](#)

### 3.1.12 检视PLC内存用法的交叉引用总结

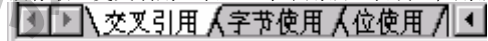
可使用下列一种方法检视“交叉引用”窗口：

- 选择菜单命令 **检视 (View) > 交叉引用 (Cross Reference)**



- 单击浏览条中的“交叉引用”按钮
- 打开 [指令树](#) 中的“交叉引用”文件夹，然后双击某引用或使用节点

欲存取“交叉引用”表、“字节用法”表或“位用法”表，单击位于“交叉引用”窗口底部的适当标记



本标题讨论下列主题：

[交叉引用表](#)  
[字节用法表](#)  
[位用法表](#)

[交叉引用表](#)  
[返回顶端](#)



**提示：** 您必须编译程序才能检视“交叉引用”表。

当您希望了解程序中是否已经使用和在哪里使用某一符号名或内存赋值时，可使用“交叉引用”表。“交叉引用”列表识别在程序中使用的所有操作数，并识别 POU、网络或行位置以及每次使用的操作数指令上下文。

**元素**指程序中使用的所有操作数。您可以在符号和绝对视图之间切换，改变全部操作数显示。（使用菜单命令 **检视 (View) > 符号编址 (Symbolic Addressing)**。）

**块**指使用操作数的 POU。

**位置**指使用操作数的行或网络。

**上下文**指使用操作数的程序指令。

LAD 交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1	LD
2 动2	MAIN (OB1)	网络 2	LD
3 停止1	MAIN (OB1)	网络 1	LD
4 停止2	MAIN (OB1)	网络 2	LD
5 高等级	MAIN (OB1)	网络 1	LD
6 高等级	MAIN (OB1)	网络 2	LD

FBD 交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1	OR
2 动2	MAIN (OB1)	网络 2	OR
3 停止1	MAIN (OB1)	网络 1	AND
4 停止2	MAIN (OB1)	网络 2	AND
5 高等级	MAIN (OB1)	网络 1	AND
6 高等级	MAIN (OB1)	网络 2	AND

STL 交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1, 行 1	LD
2 动2	MAIN (OB1)	网络 2, 行 1	LD
3 停止1	MAIN (OB1)	网络 1, 行 3	A
4 停止2	MAIN (OB1)	网络 2, 行 3	A
5 高等级	MAIN (OB1)	网络 1, 行 4	AN
6 高等级	MAIN (OB1)	网络 2, 行 4	AN

字节用法表

[返回顶端](#)





**提示：** 您必须编译程序才能检视“字节用法”表。

“字节用法”表允许您查看程序中使用了哪些字节以及在哪些内存区使用，还可帮助您识别重复赋值错误。

**b**表示已经指定一个内存位。

**B**表示已经指定一个内存字节。

**W**表示已经指定一个字（16位）。

**D**表示已经指定一个双字（32位）。

**X**用于计时器和计数器。

范例1：解释字节用法表

该字节用法表范例显示相关程序使用下列内存位置：MB0中一个位；计数器C30；计时器T37。

字节	9	8	7	6	5	4	3	2	1	0
MB0										b
C0										
C10										
C20										
C30										X
T0										
T10										
T20										
T30			X							

范例2：识别重复赋值错误

该范例程序从MB 10.0开始进行重复内存赋值。

```

网络 1  网络标题
LD      M10.0
AB=    MB10, VB0
AW=    MW10, VW0
AD=    MD10, VD0

```

可检查字节用法表，识别不适当的赋值。因为双字要求四个字节，VB0行中应有4个相邻的D。与此相似，因为字要求2个字节，VB0中应有2个相邻的W。MB10行存在相同的问题，此外在多个赋值语句中使用MB10.0。

字节	9	8	7	6	5	4	3	2	1	0
VB0							D	D	W	B
MB0										
MB10							D	D	W	b

位用法表

[返回顶端](#)



**提示：** 您必须编译程序才能查看位用法表。

位用法表允许您查看程序中已经使用了哪些内存地址，可精确至位级别，还可帮助您识别重复赋值错误。

**b**表示已经指定一个内存位。

**B**表示已经指定一个内存字节。

**W**表示已经指定一个字（16位）。

**D**表示已经指定一个双字（32位）。

**X**用于计时器和计数器。

范例1：解释位用法表

该位用法表范例显示相关程序使用下列内存位置：从字节I0、位0、1、2、3、4、5和7；从字节Q0、位0、1、2、3、4和5；从字节M0、位1。

位	7	6	5	4	3	2	1	0
I0.0	b		b	b	b	b	b	b
Q0.0			b	b	b	b	b	b
M0.0							b	

范例2：识别重复赋值错误

该范例程序从MB 10.0开始进行重复内存赋值。

```

网络 1 网络标题
LD M10.0
AB= MB10, VW0
AW= MW10, VV0
AD= MD10, VD0
  
```

可检查位用法表，识别不适当的赋值。在适当的赋值程序中，字节中间不得有位值。BBBBBBb无效，而BBBBBBBb则有效。

嘿 慕婁丁彩视糜滥指担虫 τ 16个相邻的W)和双字赋值(应有32个相邻的D)。

位	7	6	5	4	3	2	1	0
M0.0								
M1.0								
M2.0								
M3.0								
M4.0								
M5.0								
M6.0								
M7.0								
M8.0								
M9.0								
M10.0	B	B	B	B	B	B	B	b
M11.0	W	W	W	W	W	W	W	W
M12.0	D	D	D	D	D	D	D	D
M13.0	D	D	D	D	D	D	D	D

另请参阅:


[通讯配置](#)  
[PLC类型选项](#)  
[系统块 \(PLC配置\)](#)  
[错误](#)  
[在符号和绝对视图之间切换](#)

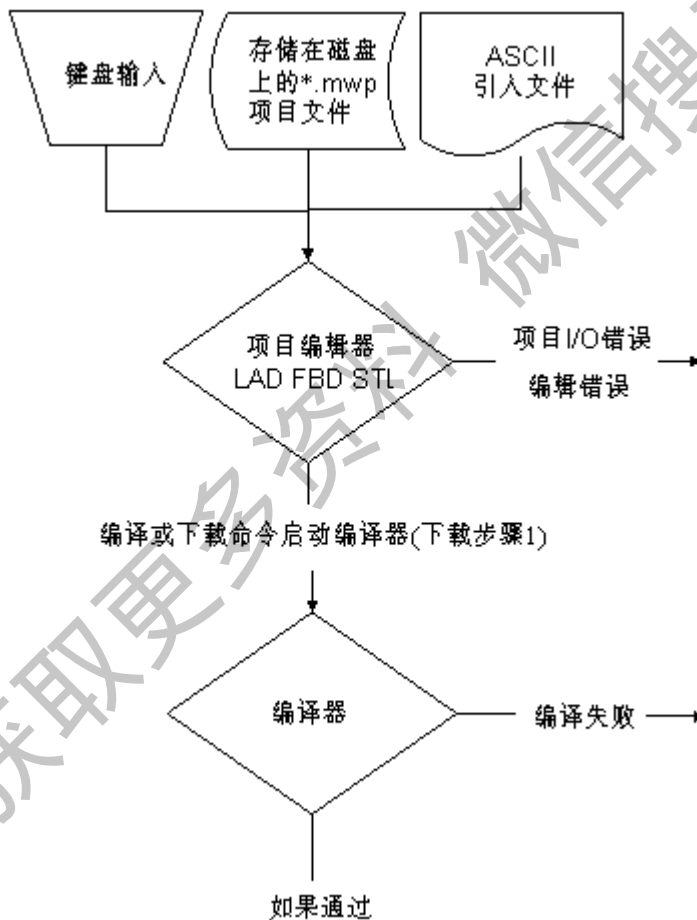
[入门指南目录](#)  
[应用程序用户参考手册](#)

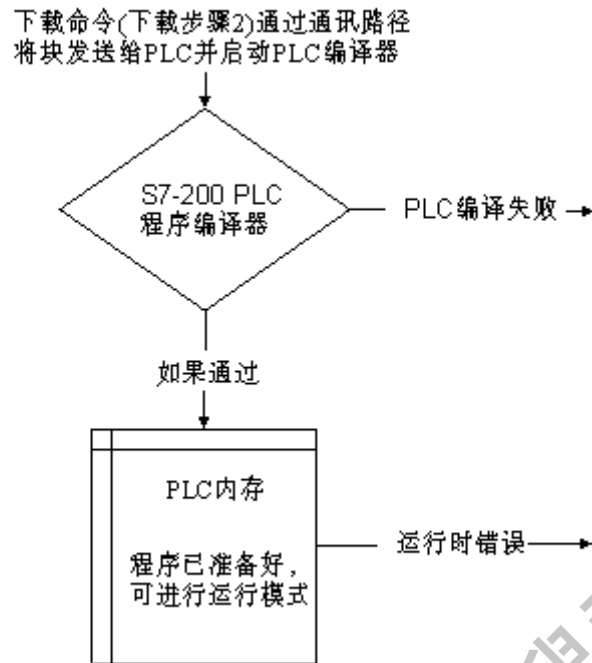
### 3.1.23 将程序下载至PLC

可使用下列一种方法从STEP 7-Micro/WIN向PLC下载项目元件:

- 单击“下载”  按钮。
- 选择菜单命令文件 (File) > 下载 (Download)。
- 按Ctrl+D快捷键组合。

单击“选项”按钮 ，显示和设置下载操作选项。‘上载下载管理器’窗口可帮助您控制上载、下载、清除及  
[UDM - 上载下载管理器详细说明](#)

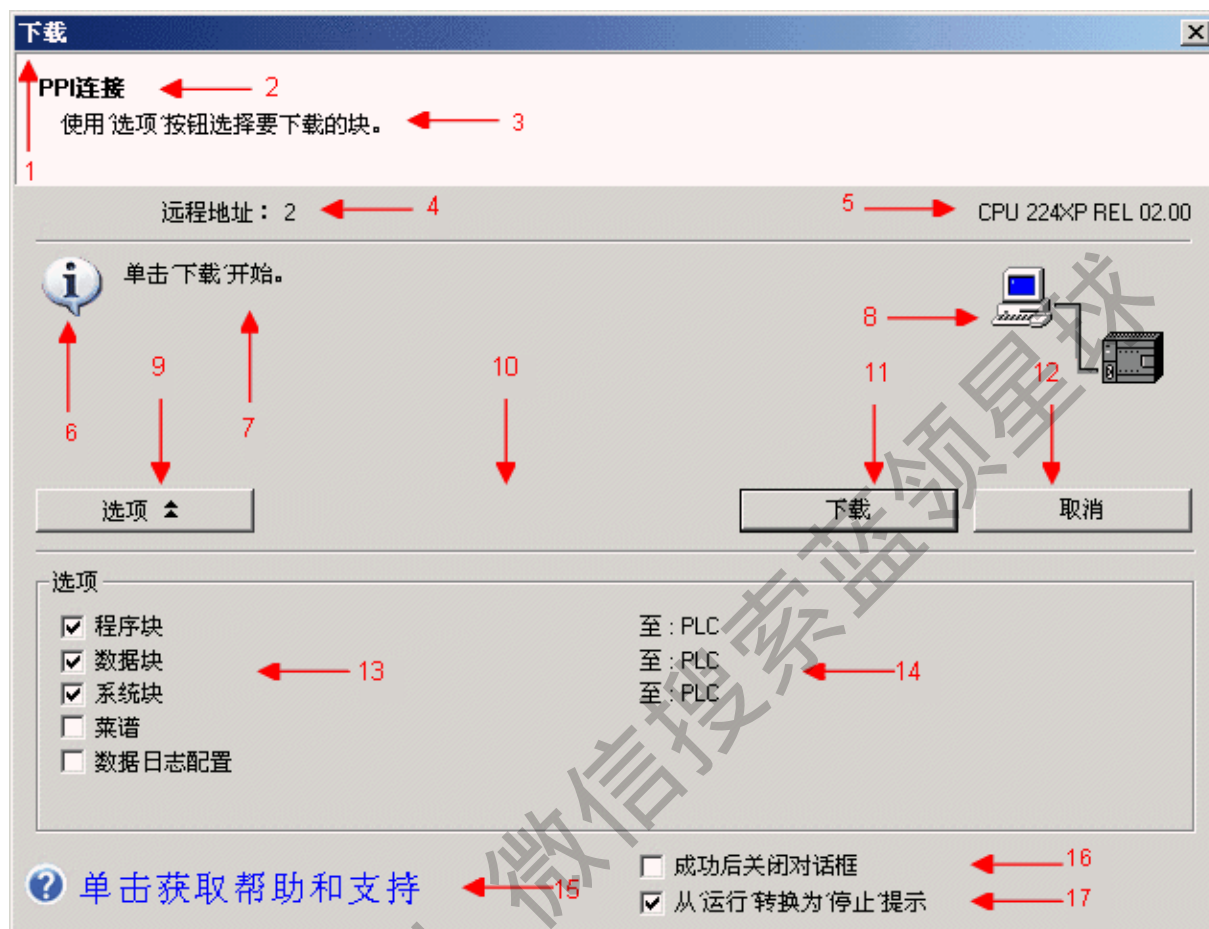




设置PLC类型 项目文件I/O错误 LAD/FBD编辑器错误 全部STEP 7-Micro/WIN编译器错误均列在输出窗口中。双击错误，编辑器滚动至错误位置。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译器。 **通讯错误** 欲下载（编辑器至PLC）或上载（PLC至编辑器），PLC通讯必须正常运行。确保网络硬件和PLC接头电缆均正常操作。PLC编译器确认PLC硬件支持全部程序指令、范围和结构。使用菜单命令 **PLC > 信息** 查看发现的第一条编译错误。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译程序，这样会将尽可能多的错误移至输出窗口，易于找到错误位置和改正错误。 **严重错误**和**非严重运行时间编程故障**由菜单命令 **PLC > 信息** 报告。

[打开项目](#) [范围检查](#)

UDM - 上下载管理器详细说明：（用于上载、下载、清除及编程内存操作）



1. 标题 - 将进行的操作
2. 连接类型 - 通讯连接类型
3. 操作说明与指令 - 进程说明与指示
4. CPU地址 - 通讯地址
5. CPU类型 - 目标CPU的版本和类型
6. 状态图标 - 表明进程的总体状态
7. 状态讯息区 - 显示错误和警告的区域
8. 动画图标 - 表明操作处理是否在进行中
9. 选项按钮 - 基于用户的优先选择隐藏或显示选项选择
10. 更改项目/通讯按钮 - 更改项目的CPU类型以与实际CPU相符，也允许通讯配置
11. 动作按钮 - 如果没有错误存在就开始操作
12. 取消按钮 - 在下一个停止点取消正在进行的操作
13. Option selections - 选取选项
14. Option status? 选定的选项状态
15. 帮助 - 打开MicroWin帮助（与按F1键相同）
16. 成功操作后关闭对话框 - 用于在成功操作后关闭此窗口的选项
17. 从‘运行’转换为‘停止’提示 - 当CPU从运行模式转换为停止模式时省略用户提示的选项。

#### 另请参阅:

[通讯概述 \(GS 6.1\)](#)

[如何测试通讯网络 \(GS 6.2\)](#)

[如何下载程序 \(GS 6.3\)](#)

[在运行模式中进行程序编辑](#)

[上载 \(文件 > 上载\)](#)

[如何从PLC上载 \(GS 8.3\)](#)

[STEP 7-Micro/WIN编译程序错误](#)

[如何纠正编译错误和下载错误 \(GS 6.4\)](#)

[应用程序用户参考手册](#)  
[入门指南目录](#)

### 3.1.24 使用状态监控确认运行时间性能

您成功地在运行的STEP 7-Micro/WIN编程设备和PLC之间建立通讯并向PLC下载程序后，则可利用“调试”工具条的诊断特征。可单击工具条按钮或从“调试”菜单列表选择项目，选择调试工具。

**调试工具条**



本标题讨论下列主题：

[什么是“状态”？](#)

[前提条件](#)

[收集状态的不同方法](#)

[以连续、瞬态图或触发暂停的方式收集状态](#)

[PLC模式](#)

[状态通讯与扫描循环](#)

[模拟程序条件（读取、写入、强制、取消强制）](#)

[检查交叉引用和元素用法](#)

[向程序下载改动](#)

[重设用于调试和编辑的帧设置](#)

[什么是“状态”？](#)

[返回顶端](#)

‘状态’这一术语指显示程序在PLC中执行时有关PLC数据当前值和功率流的信息。您可以使用状态图和程序状态窗口读取、写入和强制PLC数据值。

在控制程序的执行过程中，PLC数据的动态改变可用三种不同方式检视：

#### 图状态

在一表格中显示状态数据：每行指定一个要监视的PLC数据值。您指定一个内存地址、格式、当前值及新值（如果使用写入命令）。

#### 趋势显示

用随时间而变的PLC数据之绘图跟踪状态数据：您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势数据亦可在趋势视图中直接赋值。

#### 程序状态

在程序编辑器窗口中显示状态数据：当前PLC数据值会显示在引用该数据的STL语句或LAD/FBD图形旁边。LAD图形也显示功率流

#### 程序状态和图状态 (或趋势视图) 窗口可以同时运行：

在状态图窗口写入或强制PLC数据将PLC数据改动应用于程序状态窗口。

在程序状态窗口写入或强制PLC数据还会将新PLC数据改动应用于状态图窗口。

#### 检视状态数据的前提条件

[返回顶端](#)

您必须执行下列任务，才能监控状态数据或调试程序：


- 成功地在STEP 7-Micro/WIN和PLC之间建立通讯。
- 成功地向PLC下载程序。
- 欲检视连续状态更新，PLC必须位于RUN（运行）模式。否则，您只能看到对I/O的改动（如果有）。由于PLC程序辉溢葱校 I/O的改动不会对“状态”显示中的程序逻辑产生预期的影响。
- 如果您位于程序不在执行的区域（例如，子例行程序、中断例行程序或由于JMP指令被绕过的区域），不会有状态显示，因为不在对代码执行扫描。

#### 注释：

- STEP 7-Micro/WIN中项目的时间标记和PLC必须匹配，才能激活状态数据通讯。时间标记比较（在您尝试打开状态时自动执行）繁 STEP 7-Micro/WIN中的项目状态表示正确反映实际PLC程序。如果时间标记不匹配，您可以检查“时间标记不匹配”对话框中南吕 斜恚 私悻男 纛斐刹黄 涸
- 下载程序后，您还应将PLC转换回RUN（运行）模式。否则会报告STOP（停止）模式状态数据，但PLC无法执行程序，因此您将无法看到预期的逻辑操作。

#### 在程序编辑器窗口中收集状态的不同方法

[返回顶端](#)

单击“程序状态打开/关闭”按钮，或选择菜单命令**调试 (Debug) > 程序状态 (Program Status)**，在程序编辑器窗口中显示PLC数据状态。状态数据收集按以前选择的模式开始。请参阅以下说明。

LAD和FBD程序有两种不同的程序状态数据收集模式。选择**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单命令会在状态模式复选标记打开和关闭之间切换。必须在程序状态操作开始之前选择状态模式。

- **执行状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选后）



- **扫描结束状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被取消勾选后）



状态图始终使用“扫描结束”状态数据。以前的 STEP 7 Micro/WIN 版本用“扫描结束”状态数据建立 LAD 和 FBD 程序状态视图。如果您使用的是第一代 PLC，执行状态不受支持。如果您尝试执行执行状态，“扫描结束”状态开始。

#### 执行状态色彩：

- 程序被扫描时，电源横杆会带有色彩。
- 图形中的功率流用色彩表示。
- 接点？接点打开时，指令会带有色彩。
- 线圈？输出打开时，指令会带有色彩。
- 方框和 SUBR 指令？指令接通电源并准确无误地成功执行时，方框和 SUBR 指令带有色彩。
- 绿色计时器和计数器表示计时器和计数器包含有效数据。
- 红色表示执行的指令有误。
- 跳接和标签指令现用时，显示为功率流色彩。如果为非现用，则显示为灰色。
- 灰色（灰色为默认赋值）表示无功率流、指令未扫描（跳过或未调用）或位于 STOP（停止）模式的 PLC。
- 布尔功率流位（仅限 FBD）。

LAD、FBD 和 STL 执行状态视图显示操作数数值，并表示每条指令在位于扫描循环执行程序阶段时的功率流。执行状态可以显示可能被执行后续程序指令盖写的中间数据值。所有显示的 PLC 数据值均从一个程序扫描循环收集。

#### 扫描结束状态

- 接通电源或逻辑真的接点和线圈显示为蓝色（您可以选择 **工具 (Tools) > 选项 (Options)** 菜单项目并单击“色彩”标记，指定自己选择的色彩）。

“扫描结束”状态显示在程序扫描结束时读取的状态结果。这些结果可能不会反映 PLC 数据地址的所有数值改动，因为随后某绿蜂鸣铃语毓绿蛭 杞喂 翱瞻芊慈牒椭陔漆慈胧 怠 S 捐诳炯 PLC 扫描循环和相对慢速的 PLC 状态数据通讯之间存语乃俛快败稿 吧 杞喂 弊刺 允炯父鏊 枢 方喂 辈杉 氛 奠怠

#### STL 程序中程序状态举例：

当您开启 STL 中的状态时，程序编辑器窗口被分为一个代码区（左侧）和一个状态区（右侧）。可以根据您希望监控的数值类型

在 STL 状态中共有三个可监控的数值类别：

- **操作数** 每条指令最多可监控三个操作数。
- **逻辑堆栈** 最多可监控四个来自逻辑堆栈的最新数值。
- **指令状态位** 最多可监控十二个状态位。

**工具 (Tools) >**

**选项 (Options)** 对话框的 STL 状态标记允许您选择或取消选择任何此类数值类别。如果您选择一个项目，该项目不会在“状态”显示中出现。

	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+16095	
5	T32	位	2#0	
6	T32	带符号	+0	

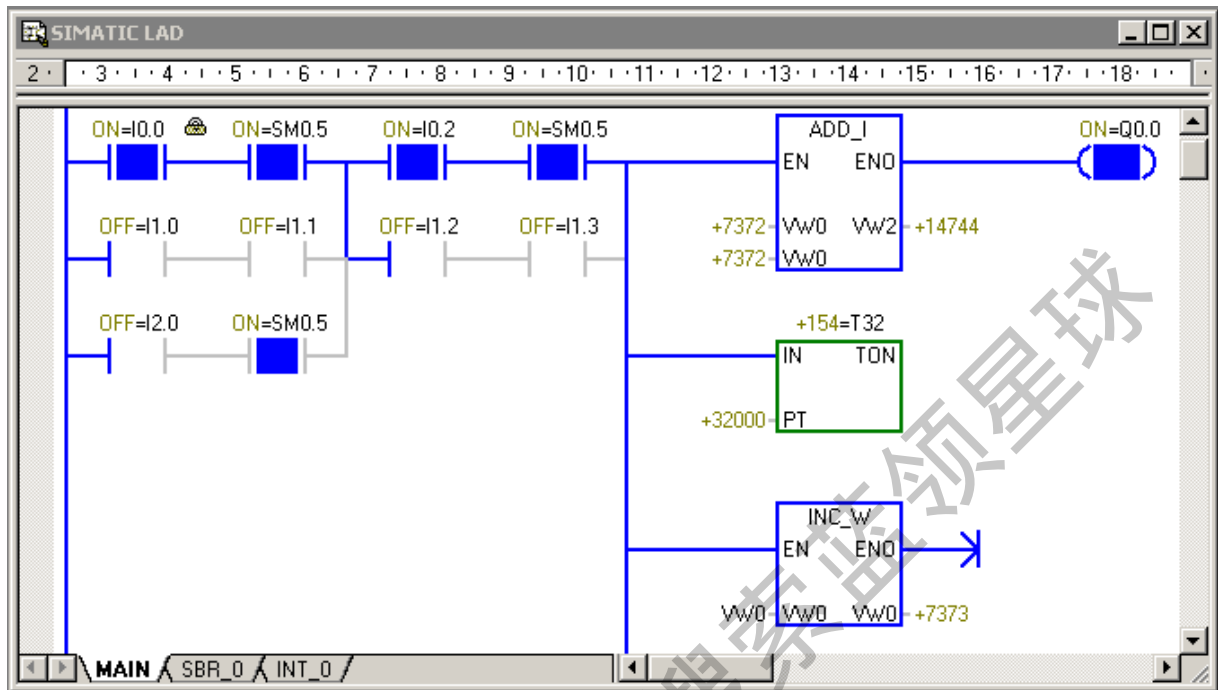


		操作数 1	操作数 2	操作数 3	0123	空
LD	I0.0	ON			1000	1
A	SM0.5	ON			1000	1
LD	I1.0	OFF			0100	0
A	I1.1	OFF			0100	0
OLD					1000	1
LD	I2.0	OFF			0100	0
A	SM0.5	ON			0100	1
OLD					1000	1
LD	I0.2	ON			1100	1
A	SM0.5	ON			1100	1
LD	I1.2	OFF			0110	0
A	I1.3	OFF			0110	0
OLD					1100	1
ALD					1000	1
LPS					1100	1
MOVW	VW0, VW2	+15919	+15919		1100	1
AENO					1100	1
+I	VW0, VW2	+15919	+31838		1100	1
AENO					1100	1
=	Q0.0	ON			1100	1
LRD					1100	1
TON	T32, +32000	+288	+32000		1100	1
LRD					1100	1
TNCW	VW0	+15920			1100	1

**LAD图形中执行状态举例：**

以下范例在LAD图形中使用LAD执行状态。状态图仅使用“扫描结束状态”数据收集。可在LAD图形和对应的状态图之间看出

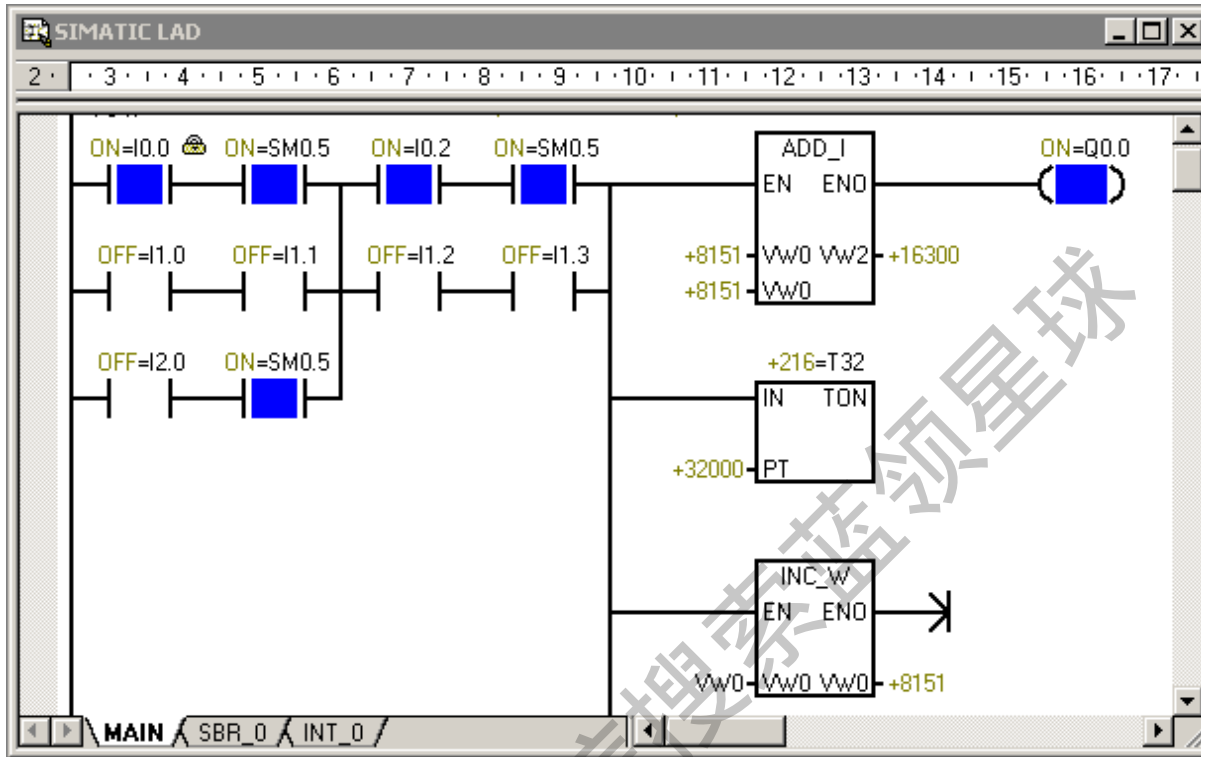
地址	格式	当前值	新数值
1	I0.0	位 2#1	
2	I0.2	位 2#1	
3		带符号	
4	VW0	带符号 +7652	
5	T32	位 2#0	
6	T32	带符号 +0	



#### LAD图形中扫描结束状态举例:

以下范例在LAD图形中使用“扫描结束状态”。由于两个窗口（程序状态和图状态）均使用相同的扫描 PLC结束数据，两个窗口允鞠嗤 PLC数据值。这种情形会引起误解，因为程序可以在收集最终“扫描结束”数值之前为相同的位置指定很多数值。中间临时数值不显示。在该范例中，ADD\_I方框未能正确执行，因为输入相加超出了带符号字整数的最大范围。与红色方框的执行状态范例不同，您必须读取状态值，才能看到Vw0 + Vw0 <> VW2。

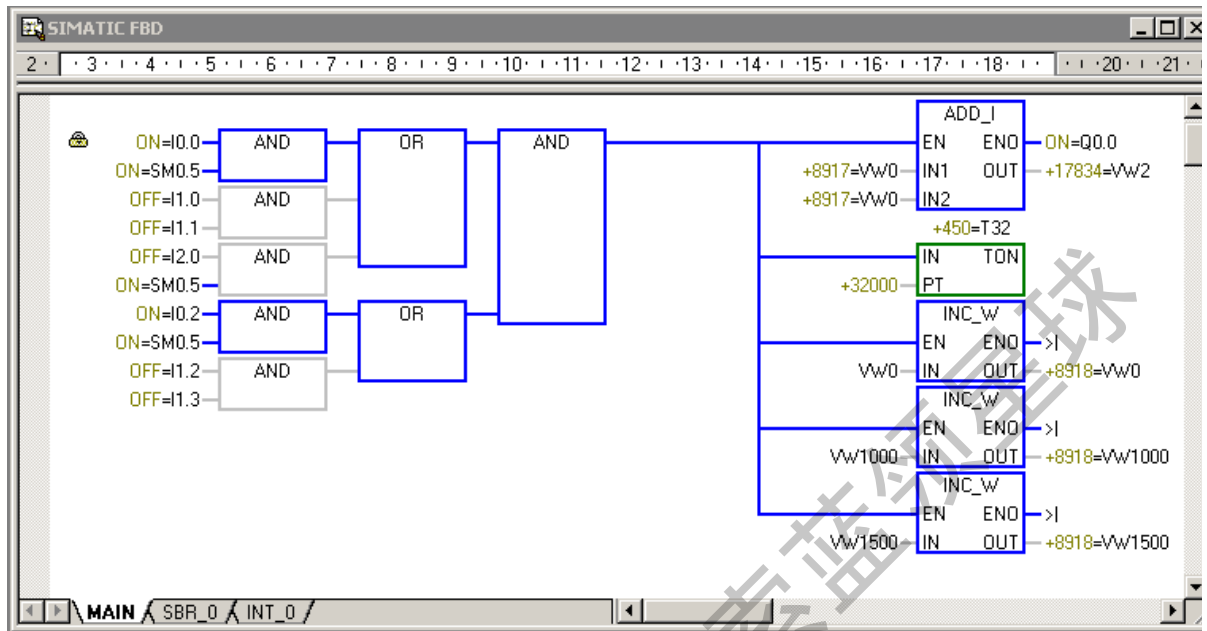
地址	格式	当前值	新数值
1	I0.0	位	2#1
2	I0.2	位	2#1
3		带符号	
4	VW0	带符号	+8418
5	T32	位	2#0
6	T32	带符号	+485



FBD图形中的执行状态举例:

地址	格式	当前值	新数值
1 I0.0	位	2#1	
2 I0.2	位	2#1	
3	带符号		
4 Vw0	带符号	+8958	
5 T32	位	2#0	
6 T32	带符号	+0	

获取更多资料



按照连续、瞬态图或触发（暂停程序状态）的方式收集状态值

[返回顶端](#)

#### 注释：

“连续”并非意味着实时；而是指编程设备不断地从 PLC 轮询状态信息，在屏幕中显示，并按按照通讯允许的速度更新显示。可能

#### 连续

- 打开程序编辑器窗口，并启动“程序状态”，当 PLC 位于 RUN（运行）模式时检视程序执行的连续状态更新。
- 打开状态图窗口，并启用“图状态”，当 PLC 位于 RUN（运行）模式时检视连续更新。



#### 瞬态图

- 将 PLC 转换为 STOP（停止）模式，收集单个状态更新。当 PLC 位于 STOP（停止）模式时，您可以使用“多次扫描”功能检视一次或多次扫描。您还可以引用网络中的特殊内存 SM0.1 首次扫描位，读取该网络的首次扫描状态值。
- 关闭图状态，如果您希望收集单次状态图更新并不希望将 PLC 转换至 STOP（停止）模式，使用“单次读取”功能。如 PLC 转换为 STOP（停止）模式，并开启“图状态”，您也会收集单次状态更新。此外，您可以在检视状态图时使用“多次扫描”和“首次扫描”功能。

#### 执行状态模式（LAD / FBD / STL）中的程序状态触发暂停

欲检视触发暂停 PLC 数据状态，您必须：

1. 将程序下载至 PLC，并将 PLC 设为 RUN（运行）模式。您还可以从 STOP（停止）模式开始触发暂停。如果您从 STOP（停止）模式开始，则可以捕获首次扫描。
2. 滚动程序编辑器窗口，显示您希望检视的程序网络。网络可以位于任何 POU（主程序、子例行程序或中断程序）中。
3. 使用 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单命令，将“使用执行状态”复选标记切换为打开。
4. 激活程序状态，开始状态数据通讯。



5. 单击“暂停程序状态” 工具条按钮，或用鼠标右键在程序编辑器中单击，并选择弹出菜单命令“暂停程序状态”。
6. 更新暂停之后，触发网络中的状态值在您检视时保持不变。当您准备移至另一个网络时，则单击“暂停程序状态” 工具条按钮，将暂停切换为关闭，并再次开始连续更新。

一旦获得新的状态信息，只要程序状态暂停功能被启用，该信息就会保持在屏幕上。在您取消选择程序状态暂停功能之前，无论 PLC 中实际变化速度多快，均不会进一步执行状态更新。如果没有程序状态暂停功能，屏幕就会不断刷新，状态数据在您读取和解释之前就可能消失（或者数据根本没有显示）。

#### PLC RUN / STOP（运行 / 停止）模式

[返回顶端](#)

使用以下一种方法改变PLC操作模式：

- 单击“运行”按钮，执行RUN（运行）模式，或者单击“停止”按钮，执行STOP（停止）模式。
- 选择**PLC > 运行 (RUN)** 菜单命令，执行RUN（运行）模式，或选择**PLC > 停止 (STOP)** 菜单命令，执行STOP（停止）模式。
- 以手动方式改变位于PLC上的模式开关。
- 在程序中插入一条STOP（停止）指令。

#### 注释：

欲使用STEP 7-Micro/WIN软件控制RUN / STOP（运行 / 停止）模式，必须在STEP 7-Micro/WIN和PLC之间存在一条通讯路径。此外，必须将PLC硬件模式开关设为TERM（终端）或RUN（运行）。将模式开关设为TERM（终端）不会改变PLC操作模式，但却允许STEP 7-Micro/WIN改变PLC操作模式。位于PLC前方的状态LED表示当前操作模式。当程序状态或状态图操作正在进行时，在STEP 7-Micro/WIN窗口右下方处附近的状态条上会出现一个RUN / STOP（运行 / 停止）指示灯。

虽然程序在STOP（停止）模式中不执行，PLC操作系统继续监控PLC（收集PLC RAM和I/O状态），将状态数据传递给STEP 7-Micro/WIN，并执行所有的“强制”或“取消强制”命令。当PLC位于STOP（停止）模式中时，您可以执行以下操作：

- 使用图状态或程序状态检视操作数的当前值。（此一步骤具有与执行“单次读取”相同的效果，因为程序未执行。）
- 您可以使用图状态或程序状态强制数值。使用图状态写入数值。
- 写入或强制输出。
- 执行有限次数扫描，并通过状态图和 / 或项目状态检视效果。

当PLC位于RUN（运行）模式时，您不得使用“首次扫描”或“多次扫描”功能。您可以在状态图中写入和强制数值，或使用LAD或FBD程序编辑器强制数值，方法与您在STOP（停止）模式中强制数值相同。您还可以执行以下操作（不得从STOP（停止）模式）：

- 使用图状态收集不断变化的PLC数据值的连续更新信息。（如果您希望使用单次更新，图状态必须关闭，才能使用“单次读取”命令。）
- 使用程序状态收集不断变化的PLC数据值的连续更新信息。
- 使用RUN（运行）模式功能中的“程序编辑”编辑程序，并将改动下载至PLC。

#### 状态通讯与扫描循环

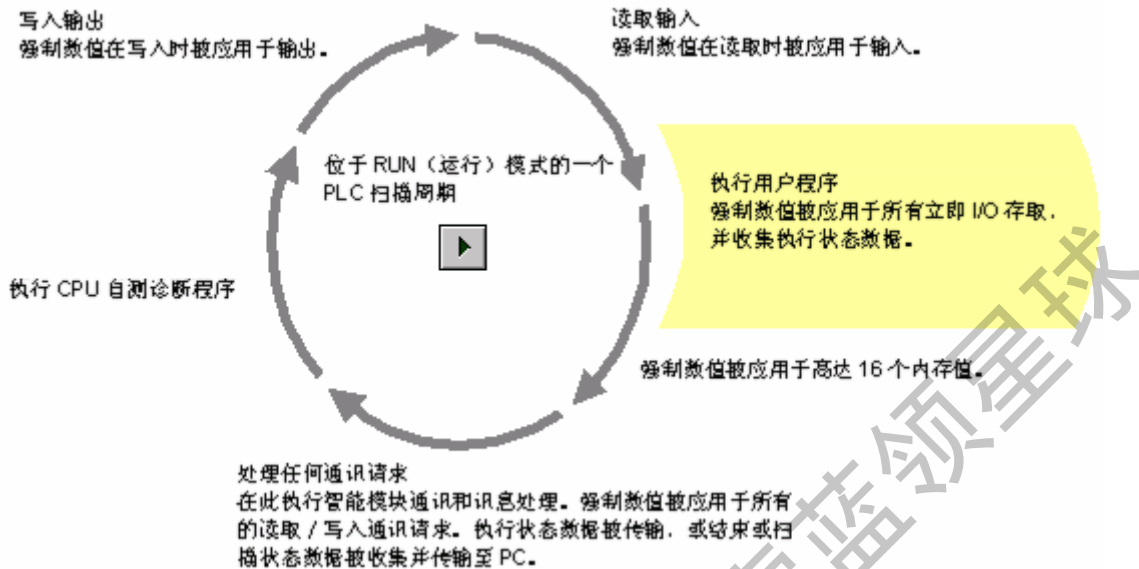
[返回顶端](#)

PLC在连续循环中读取输入、执行程序逻辑、写入输出和执行系统操作和通讯。该扫描循环速度极快，每秒执行多次。虽然STEP 7-Micro/WIN会快速发出状态请求，您应当认识到您检视的不是PLC中出现的每一个事件，这一点十分重要。

由于PLC和您观察的程序状态编程设备之间存在通讯时间滞后，您看到的显示的操作数数值总是在状态显示中改变之前即在PLC中改变，但没有在程序状态表示中改变。

如果您使用“扫描结束”状态模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被取消勾选时），您在几个扫描循环中采集数据。

如果您使用“执行状态”模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选时），所有显示的程序状态值一定来自同一个扫描循环。



**注释：**

一次只有一个状态窗口被授权运行状态。如果您尝试在任何其他与同一个 PLC 连接的 STEP 7 Micro/WIN 应用程序中运行状态，则会收到错误讯息。

**模拟进程条件（读取、写入、强制、取消强制）**


[返回顶端](#)

您可以模拟进程条件，方法是在程序状态操作过程中从程序编辑器和从状态图向操作数写入或强制新数值。

使用“程序状态”和“图状态”按钮或“调试”菜单命令，开始状态数据通讯，并启用调试工具。您还可以使用硬件输入模拟器开关（西门子（Siemens）可提供）以手动方式在数字输入打开和关闭之间切换，并观察 PLC 上的 LED 指示灯，观察数字输出的条件。

从“调试”工具条或“调试”菜单列表存取以下功能。






**单次读取**（仅限状态图）如果您希望获得一次“瞬态图”（对状态图中的所有数值一次更新），使用“单次读取”。默认值为状态图连续轮询 PLC，获取状态更新信息。当您单击“状态图”时，状态图会切换为关闭，“单次读取”黄紫漫



**全部写入**（仅限状态图）

在您完成对几行的“新数值”列的改动后，您可以使用“全部写入”，将所有需要的改动发送至 PLC。



**强制** 您可以在操作程序状态时从程序编辑器和状态图强制地址。欲强制程序状态地址，用鼠标右键单击一个参数，并选择从菜单列表强制。欲强制状态图地址为某一数值，您必须首先规定所需的数值，方法是读取该数值（如果您希望强制当前值）或键入该数值（如果您希望将地址强制为一个新数值）。您一旦使用了强制功能，则在每次扫描时该数值均被重新应用于地址，直至您取消强制地址。“强制”功能盖写立即读取或立即写入指令。“强制”功能还盖写被配置为在转换为 STOP（停止）模式时进入一个指定数值的输出：如果 CPU 进入 STOP（停止）模式，输出被设为强制数值，而非配置的数值。通过将 V 或 M 内存强制为字节、字或双字，将 AI 或 AQ 内存在偶数字节边界上（例如 AIW6 或 AIW14）或 I/O 点上（I 和 Q 位址）强制为字，模拟逻辑条件。您可以一次强制 16 个（V、M、AI 或 AQ）地址和所有的 I/O 位（所有 I 和 Q 位址）。 该图标表示该地址被明确强制。该地址数值在地址被取消强制之前无法改变。 该图标表示该地址被隐含强制。如果地址是一个被明确强制的较大地址的一部分，该地址则被认为是隐含强制。例如，如果 VW0 被强制，则 VB0 是隐含强制（VB0 是 VW0 的第一个字节）。隐含强制数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制 VD0（该地址包含 VB0、VB1、VB2 和 VB3），则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制，则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。



**取消强制** 对于程序状态和状态图，选择一个地址，并使用“取消强制”按钮从该特定地址移除



**取消强制** 对于程序状态和状态图，选择一个地址，并使用“取消强制”按钮从该特定地址移除

强制功能。您还可以选择（单击）一个参数，然后用鼠标右键单击该参数，查看“强制”和“取消强制”功能的弹出菜单。



#### 全部取消强制

如果您希望从所有的地址中移除强制功能，使用“全部取消强制”按钮。在应用“全部取消强制”之前，您无须选择单个地址。



#### 读取全部强制（仅限状态图）

当您使用“读取全部强制”功能时，状态图的“当前数值”列会为已经明确强制、隐含强制或部分隐含强制的所有地址显示一个图标。

### 检查交叉引用和元素用法

[返回顶端](#)

调试程序时，您可以决定是否增加、删除或编辑参数。

使用“交叉引用”窗口查看程序中当前指定参数的方法，这样可以防止您意外地重复赋值。

此外，如果您希望在RUN（运行）模式中执行程序编辑，并且某些网络使用正向或负向转换（EP和N接点或方框以及STL EU或ED指令），您一定要检查“交叉引用”信息中的EU/ED数目，以防重复指令号码。

### 向程序下载改动

[返回顶端](#)

一旦调试部分代码，如果PLC位于STOP（停止）模式，您可以编辑程序、下载程序、然后再次监控状态，查看改动是否发生作用。

您不必进入STOP（停止）模式就可以在RUN（运行）模式中执行程序编辑并向PLC下载较小的改动。



**警告** 在RUN（运行）模式中向PLC下载改动时，您的改动会立即影响程序操作。不得出错；程序编辑中的错误可能导致人员死亡或严重伤害和/或设备损坏。仅限合格人员在RUN（运行）模式中执行程序编辑。

### 重设用于调试和编辑的帧设置

[返回顶端](#)

欲重设将窗口和工具条的可视性、尺寸和位置设为默认值，选择**检视（View）> 帧（Frame）> 全部重设（Reset All）**菜单命令。

您有两套不同的工作空间窗口的便利，可在调试程序时获得较大的灵活性：窗口可视性和定位。这是比正常编辑空间更大的检视空间。当您进入程序状态时，您已经设置的任何窗口定位保持初始进入的状态；如果您关闭程序状态，窗口保持初始进入的状态。同样，任何在编辑程序时执行的窗口定位均在运行时与程序状态分开维护。

[返回顶端](#)

#### 另请参阅：

[如何在程序编辑器中显示状态（GS 7.2）](#)

[如何在状态图中显示状态（GS 7.3）](#)

[如何执行有限数目扫描（GS 7.4）](#)

[如何下载程序（GS 6.3）](#)

[时间标记不匹配错误](#)（确保编程设备中的项目与PLC中的项目相匹配）

[交叉引用和元素用法](#)（确保程序编辑不引起重复赋值）

[在运行模式中执行程序编辑](#)

[PLC运行/停止模式](#)


[在停止模式中写入和强制输出](#)

[安装和移除S7-200设备](#)

[接地和布线指南](#)

## 3.1.25 保存作业



您可以使用工具条上的“保存”按钮保存您的作业，或从“文件”菜单选择“保存”和“另存为”选项保存您的作业。



- “保存”允许您在作业中快速保存所有改动。（然而，您初次保存一个项目时，会被提示核实或修改当前项目名称和柯嫉哪 涎∠睢#
- “另存为”允许您修改当前项目的名称和 / 或目录位置。

当您首次建立项目时，STEP 7-

Micro/WIN提供默认值名称“Project1.mwp”。您可以接受或修改该名称；如果您接受该名称，下一个项目的默认名称将自动递增为“Project2.mwp”。

STEP 7-Micro/WIN项目的默认目录位置是位于“Microwin”目录中的称作“项目”的文件夹，您可以不接受该默认位置。

## 3.2 功能块图编程

### 3.2.1 SIMATIC FBD指令

指令







### 3.2.2 IEC 1131-3 FBD指令

白...指令



另请参阅:

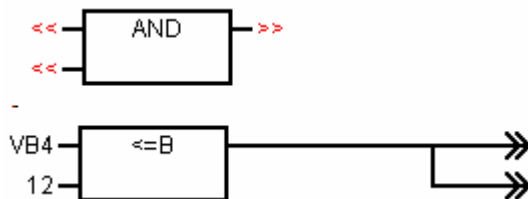
获取更多资料 微信搜索蓝领星球

使用IEC超载和非标准指令

### 3.2.3 逻辑流指示灯

在有效网络中，逻辑流从左向右流动，通过元素，最后到达输出。为了帮助您正确连接逻辑流，FBD提供“逻辑流”指示器

开路逻辑流指示器



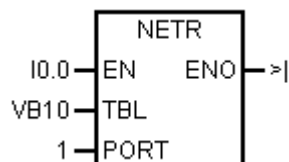
FBD为检测开路提供三种不同的视觉提示。您**必须**解决所有开路问题，网络才能成功编译。

检查AND（与）方框的逻辑流输入和输出。默认<< and >>

参数总是在红色文本中显示，表示网络中存在开路状况。这些逻辑流必须与其他指令连接或被指定一个有效参数。

在“比较字节”网络中，两个指示器在水平和垂直连接中显示一个开路。这些逻辑流在附加指令中必须继续直至到达终端。

开路逻辑流指示器



供选用逻辑流指示器显示为默认

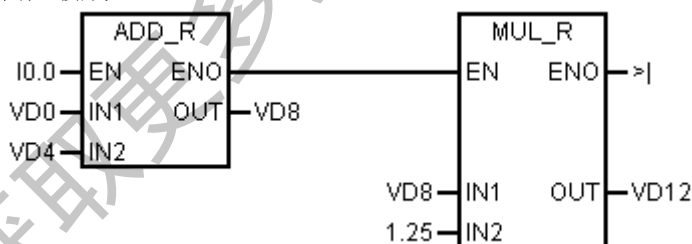
>|参数，通知您虽然不要求编译，但在该位置可附加逻辑或指定参数。该指示器用黑色文本在带 ENO指令的逻辑流输出位置

允

### 3.2.4 连接指令

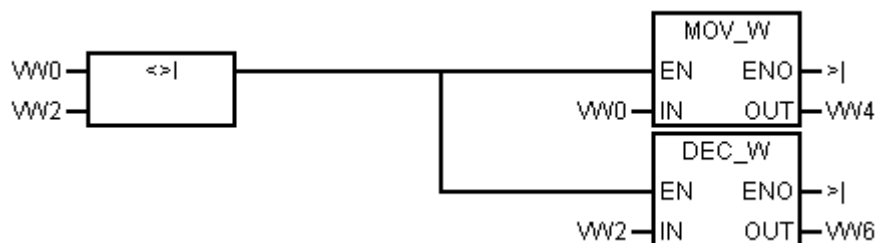
FBD提供水平和垂直帮助用串行或并行排列方式连接逻辑。将一条指令的逻辑流输出与另一条指令的逻辑流输入相连，

串行连接方框



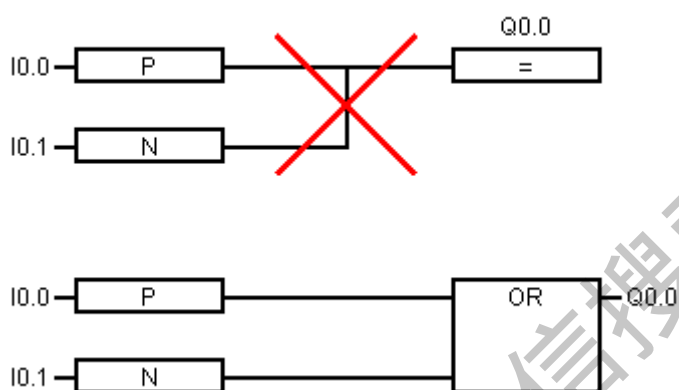
可在逻辑系列中连接方框。有时需要用水平连接指令系列逻辑流。此处使用水平连接加上实数和乘以实数方框。

多个方框分支连接



当需要用具体输出作为多个指令的输入时，可使用垂直行进行必要的连接。垂直行可从原点单元格位置向上和/或向下扩充。垂直行在此启用“比较整数”结果，将其用作两个方框的逻辑流输入。

#### 合并分支连接



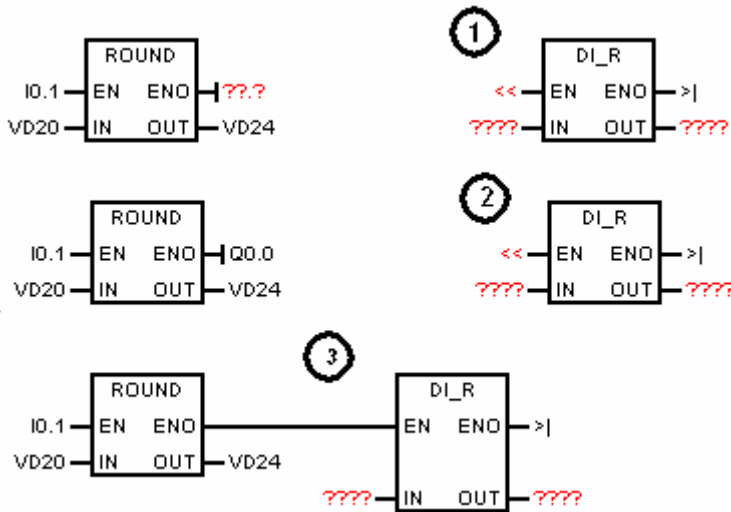
请注意垂直行不应用于将多个输出与一个具体输入连接，这种配置不能被编译。

反之，应当使用AND（与）或OR（或）方框连接多个逻辑流。

#### 自动逻辑流连接



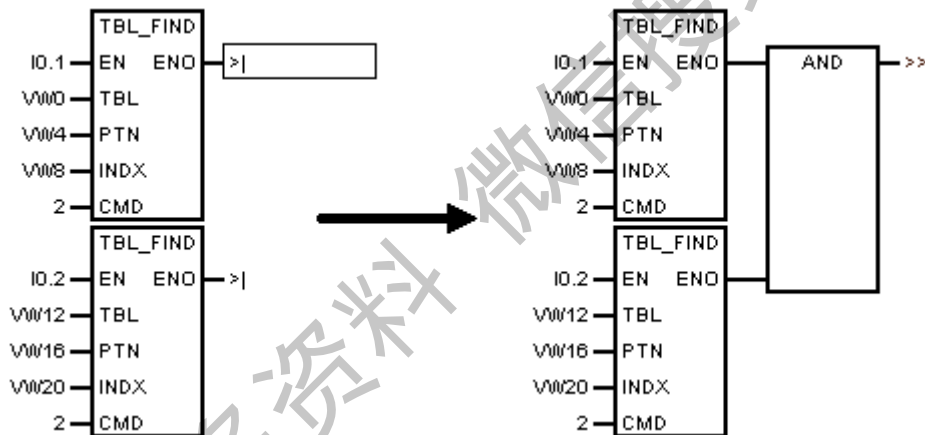
当一个元素被放入网络中时，FBD编辑器自动将元素的逻辑流输入与左侧的任何逻辑流指示器连接。在右侧的网络中，FBD远



逻辑流的自动扩充并非无条件。上面的第一个例子显示水平行未扩充至带有“立即”设置的逻辑流输出。

第二个例子显示当方框置于右侧时，指定的逻辑流参数未被逻辑流替代。

当方框被直接放入逻辑流输出中时，就会得到最后一个范例的结果。逻辑流输出参数被逻辑流替代。



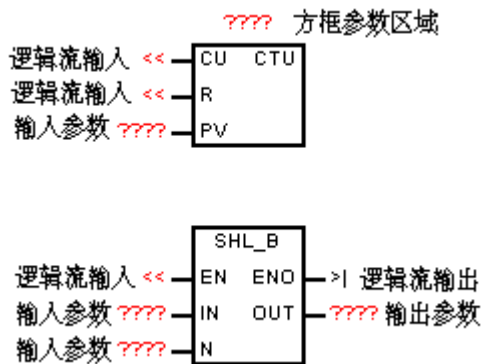
当方框被置于网络中时，方框包含两个或多个逻辑流输入，编辑器将自动将逻辑流输入与逻辑流指示器和网络输出对齐。

“表查找”方框占用网络中的许多行。为了使 AND（与）方框的第二个逻辑流输入与第二个方框的逻辑流输出相连，方框被自动扩充。

### 3.2.5 指令参数

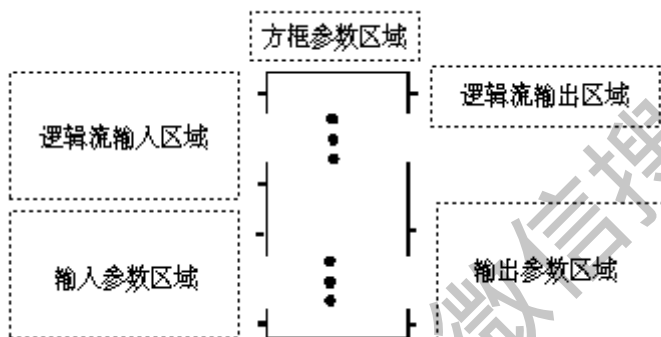
FBD中的全部指令均为方框。

方框参数



指令标签位于指令最顶端的单元格中。请注意标签也被指定为参数。

小行显示为从每个参数位置的方框向外突出。参数的指定文本紧靠方框的左侧和右侧，或方框顶部参数紧靠方框的上端。用红色显示的默认参数文本表示逻辑流对参数是否有效。



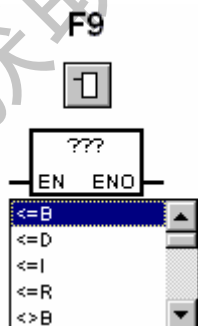
上图说明与五种不同参数类型相关的五个方框指令参数区。虽然只能有一个方框顶部参数和一个逻辑流输出参数，其他三个

### 3.2.6 通用指令

类属指令提供在FBD中快速挑选和放置指令的键盘方法。使用F9键或工具条按钮，您可以按类似以下的方法放置类属方框。娇娜路浇 毓峙判站斜枫颀 FBD指令标签以及项目的用户定义子例行程序。按键时，列表框增亮最佳匹配。出现正确的且涎 保 梢园 Enter键或双击接受新记忆符。

按Escape键取消列表框时，类属指令会保留。单击类属指令“???”标签或其附近区域或将光标放在类属指令上并按Enter键保 嶂甌孛允臼实钡牧斜颀蚶嘈汀

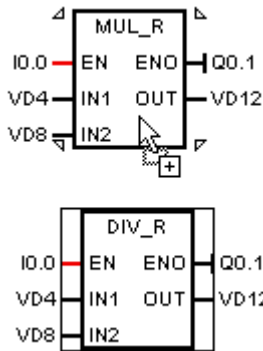
放置指令的全部规则也适用于类属方框和从列表框选择的任何指令。注释：与F4和F6键相关的AND（与）和OR（或）指令不从列表框中提供。



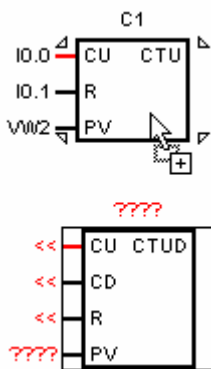
### 3.2.7 保留参数

在FBD中，指令引用可保留正在被重写的引用参数赋值，同时也保留“负数”和“立即”参数设置。必须符合下列条件：

- 编辑器必须处于“重写”模式，而且指令必须直接放在现有指令上。
- 指令在每个参数区必须具有相同数目的参数。



在本例中，“除以实数”保留“乘以实数”的参数赋值。每个方框指令都包含一个逻辑流输入参数、一个逻辑流输出参数、无方框顶部参数、两个输入参数以及一个输出参数。请注意保留第一个逻辑流输入的“负数”以及逻辑流输出的“立即”参数。



此处的“向上/向下计数”不保留“向上计数”参数。指令不具有相同数目的逻辑流输入参数。

### 3.2.8 编辑指令参数

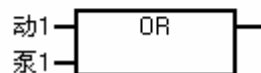
#### 剪切、复制、粘贴或删除多个网络

使用拖曳鼠标或SHIFT键和上、下箭头键，可选择多个相邻的网络，然后剪切、复制、粘贴或删除选项。利用工具条按钮，从“

如下图所示，将鼠标移到编辑器的左侧边缘（灰条区域），然后单击以开始对多个网络进行拖过选择。您不能选取部分网络。您不能选择部分网络。如果您尝试选择部分网络，会自动选择整个网络。

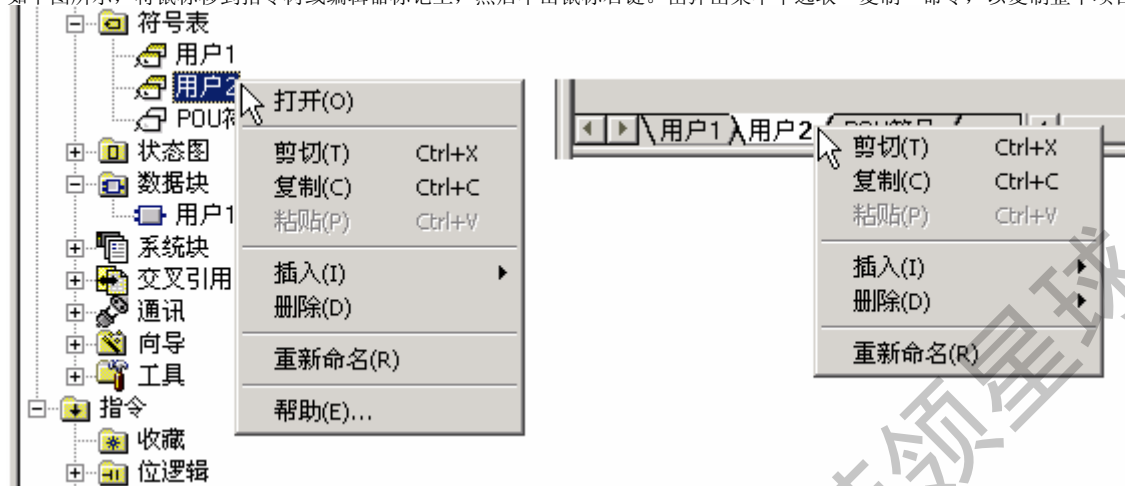
#### 网络 1 用着色配料1填充容并监控

高等级开关关闭等级  
如果动开打，维持泵状态



剪切、复制、粘贴项目元件

如下图所示，将鼠标移到指令树或编辑器标记上，然后单击鼠标右键。由弹出菜单中选取“复制”命令，以复制整个项目元件。



注释：

“复制”功能只在您选择了一个需要复制的项目后才能使用。您作出的选择确切描述了将复制到Windows剪贴板缓冲器的内容。

您可以在项目中选择以下目标

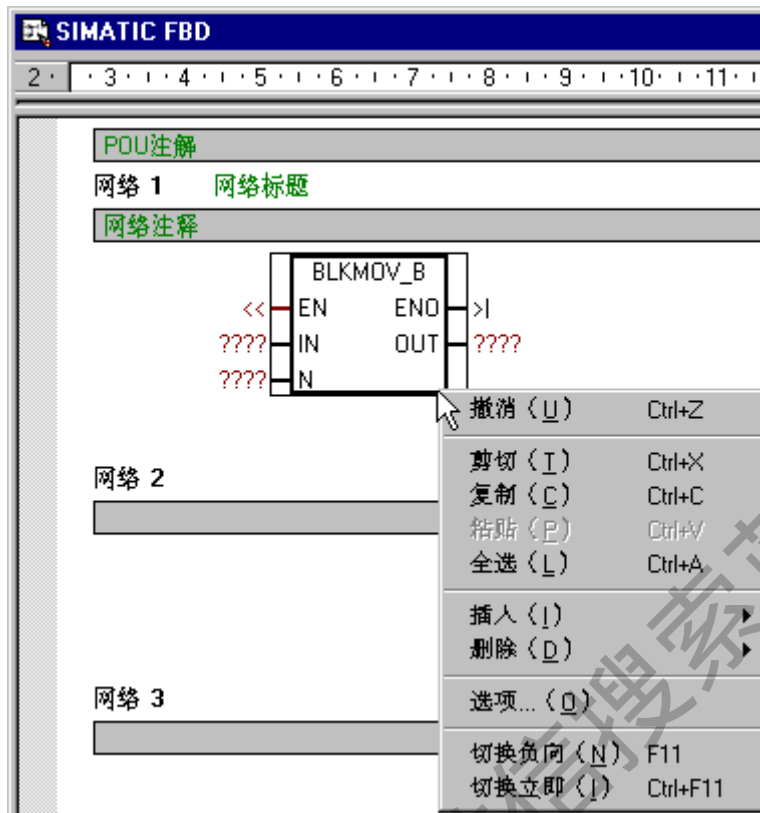
- 程序文字或数据域
- LAD、FBD、STL等编辑器中的指令
- 单个网络：将鼠标放在“网络”的左侧边缘上，然后单击
- 多个相邻网络：在选取第一个网络后，使用Shift+下箭头/上箭头或Shift+Page Up/Page Down来扩大或减少所包括的网络数目
- 某POU内的所有网络：请使用 **编辑 >**

**全选** 菜单命令，或用鼠标右键单击某指令树POU分支，或用鼠标右键单击某标记名

- 选定的数据块文字，或整个数据块标记页；  
当通过用鼠标右键单击编辑器标记名而复制整个数据块标记页时，随后的粘贴操作能够输出用制表符分隔的数据。举例来说，倘若将此数据块文字粘贴到MS Excel，则电子表格的多个单元行将得到填充。
- 选定的符号表行或列，或整个符号表标记页
- 选定的状态图行或列，或整个状态图标记页

#### 编辑单元格、指令、地址和网络

- 当您单击程序编辑器中的空单元格时，会出现一个方框，显示您选择的单元格。您可以使用弹出菜单将选项粘贴进空  
 び 襪谢蛟诟梦恢貌迦瓜恒魴濼小17.小4.怪毙谢蛸 纭D 部梢源涌盍び 竦奈恢蒙境 纭
- 当您单击一条指令时，会在指令周围出现一个方框，显示您选择的指令。您可以使用弹出菜单在指令上全选、撤消键  
 齋12.轻小8.粗苹蟀程 糝 逃之睿 06)诟梦恢貌迦膈蛸境 匕纭斜匾 一小17.谢蛸 纭D 籀梢酞u 貌说 孑媛 把之睢  
 倍曰翱颞

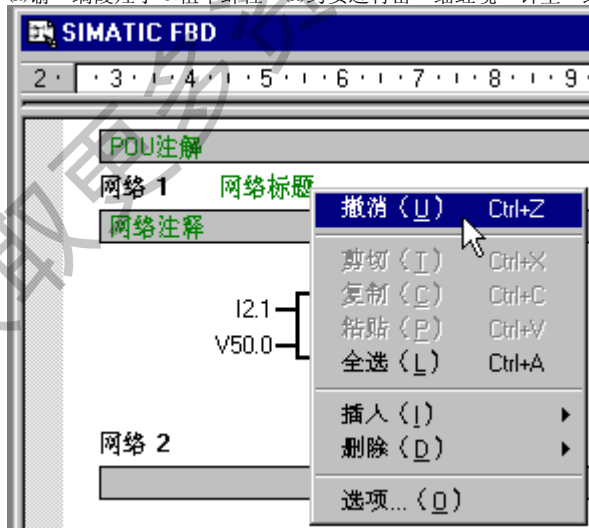


- 当您单击一个指令参数时，会在域周围出现一个方框，显示您选择的参数。您可以使用弹出菜单撤消键入、剪切、复制、粘贴、全选、插入、删除、选项、切换负向、切换立即。





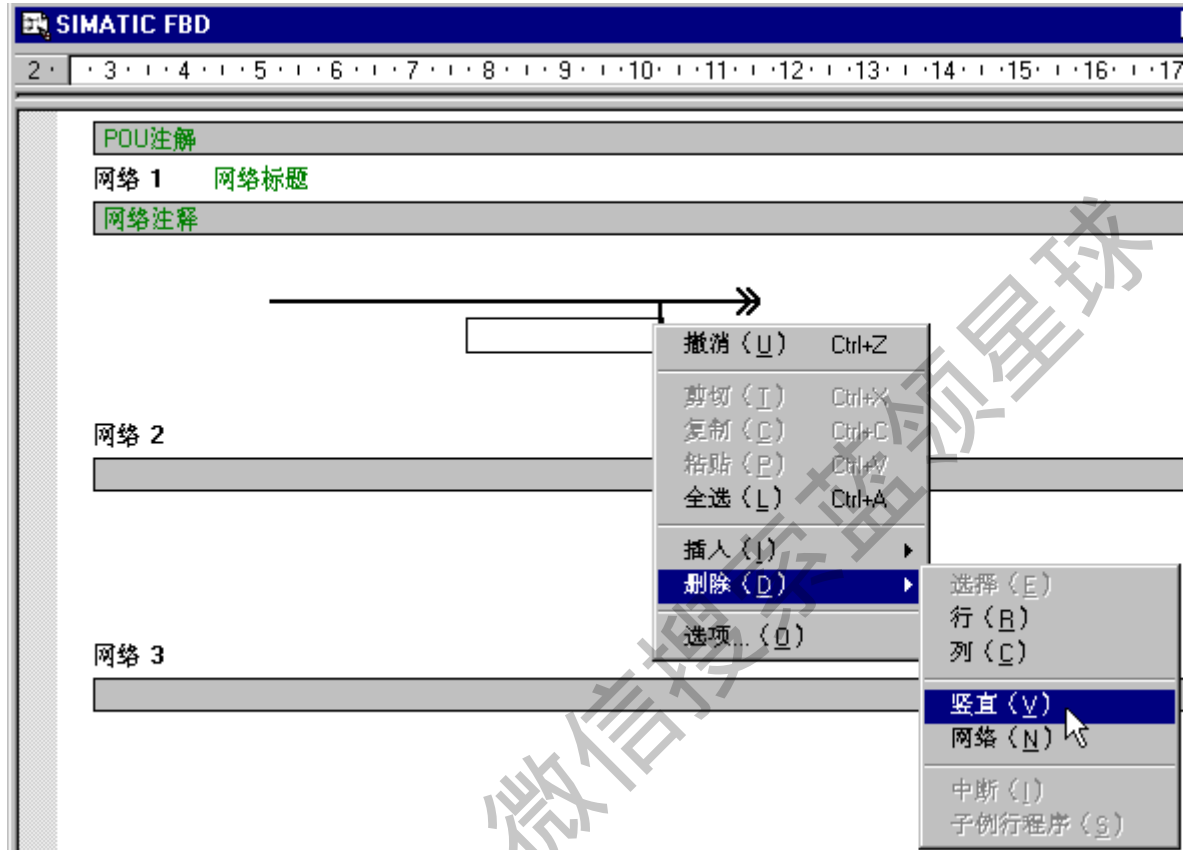
- 当您在网络标题行中单击时，您可以编辑项目的标题。您可以使用弹出菜单全选、撤消键入、在整个网络中执行编辑



您还可以使用工具条按钮、标准窗口控制键和“编辑”菜单剪切、复制或粘贴选项。

删除元素

您可以使用DELETE（删除）或BACKSPACE（退格）键删除单个单元格；您可以使用“编辑”菜单或用鼠标右键单击的方法删除垂直行。



**注释：** 为了正确选择用作删除的垂直行，始终将光标放在垂直行左边的单元格上。

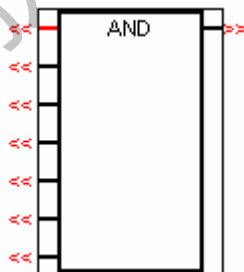
另请参阅：  
[快捷键 \(GS 1.2\)](#)

### 3.2.9 在AND（与）或OR（或）方框中增加输入行

使用下列方法之一在FBD程序中的AND（与）方框或OR（或）方框中增加另一个输入行：

- 将光标放在AND（与）方框或OR（或）方框中，然后单击“增加输入”按钮。
- 将光标放在AND（与）方框或OR（或）方框中，然后按“+”键，增加输入行。

重复此一程序，继续增加更多的输入。



另请参阅：



[FBD中的AND（与）和OR（或）用法](#)  
[如何在FBD中输入指令](#)

[应用程序用户参考手册](#)  
[入门指南目录](#)

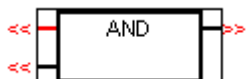
### 3.2.10 从AND（与）或OR（或）方框移除多余的输入行

在FBD程序中，如果增加了附加输入，从AND（与）方框或OR（或）方框移除输入行：



- 将光标放在AND（与）方框或OR（或）方框中，然后单击“移除输入”按钮。
- 将光标放在AND（与）方框或OR（或）方框中，然后按“-”键，删除输入行。

重复此步骤，移除其他额外的输入。



#### 注释：

支持单输入AND（与）方框。单输入AND（与）方框是单STL LD指令或单LAD接点的FBD程序视图。欲在程序中放置单输入AND（与）方框，使用一个双输入方框，然后用移除输入按钮删除一个输入。

#### 另请参阅：



[FBD中的AND（与）或OR（或）用法](#)  
[如何在FBD中输入指令](#)

[应用程序用户参考手册](#)  
[入门指南目录](#)

### 3.2.11 负（反转）逻辑连接

本标题讨论插入FBD指令的几种方法：

[指令树：拖放](#)  
[指令树：双击](#)  
[工具条按钮](#)  
[功能键](#)

当您插入指令时，可能还需要下列与本题目有关的信息：

[纠正指令位置](#)  
[如何划线](#)  
[插入与覆盖模式](#)

#### 相关标题：

[反转布尔输入](#)  
[将布尔输入或输出设为立即模式](#)  
[增加或移除附加输入（仅限AND/OR（与/或））](#)

**纠正指令位置：从左至右作业**

[返回顶端](#)

建立FBD网络时，始终从左至右作业，以便利用程序编辑器的特别功能：

- (当插入光标作为前一个方框的打开输出 >> 正确放置在同一行上时) 方框自动连接
- (当插入光标作为前一个方框的打开输出 >> 正确放置在同一行上时) 自动调整方框位置
- 自动调整方框尺寸 (以便启用正在插入的方框输入, 与前一个方框的打开输出 >> 对齐和连接)

当您从左至右作业, 并适当放置指令时 (将插入光标放在与前一条指令的输出 >> 相同的行中), 程序编辑器会自动连接和排列指令。

#### 为参数留出位置

将FBD程序编辑器视作由列和行组成的网格。

您不得将指令放在网络的最上一行; 光标必须放在第二行或下方。另外, 如果指令左边有输入, 则不能在程序编辑器的第一



**提示:** 当您在程序编辑器中工作时, 留心FBD工具条; 如果将光标放在程序编辑器的一个不允许放置特定类型线条或指令的区域中, 线条工具和指令按钮会被自动禁用。

**注释:** 将一条指令与另一条指令连接时, 不必在指令之间留出额外的空格。您可以在第一条指令的适当输出参数中 (>>) 直接放置第二条指令。

#### 将输出赋值更改为连接

如果您将参数值指定给指令的输出 >>, 但是后来决定将指令与另一条指令连接, 您可以直接在第一个输出上放置第二条指令。以前指定给该输出参数的数值 (例如 Q0.1) 会被自动盖写, 输出与替换指令连接。

#### 划线

##### 返回顶端

您可以使用程序工具条中的水平和垂直线, 或按住 CTRL 键并按键盘上的左、右、上或下箭头键, 必要时在网络元素之间划线 (例如, 建立分支)。



#### 插入与覆盖模式

##### 返回顶端

STEP 7-Micro/WIN 允许您在键盘上按“插入”键, 在两个编辑模式之间切换:

- 在插入模式 (当您按“插入”键时选择) 中, 如果您在另一条指令上放置一条指令, 程序编辑器移动现有指令, 为新指令留出位置。
- 在覆盖模式 (“插入”键未按下时的默认值) 中, 如果您在另一条指令上放置一条指令, 程序编辑器删除现有指令, 并用新指令替换现有指令。

#### 覆盖模式中的参数保留

- 如果您用具有相同概况的另一个方框替换 (覆盖) 一条指令, 您对旧参数所作的任何赋值都会转移至新参数。(即如果第二条指令具有与第一条指令相同的逻辑流输入数目、相同的输入地址参数数目、相同的逻辑流输出数目和相同的输出地址参数数目, 则当您用第二条指令覆盖第一条指令时, 参数赋值被保留。)
- 对于 AND (与) 方框和 OR (或) 方框, 即使您修改了原方框以便增加附加输入, 旧参数仍然保留。(例如, 如果您将 AND (与) 方框分配了四个输入赋值, 您用一个 OR (或) 方框覆盖 AND (与) 方框, OR (或) 方框会自动继承四个

#### 从指令树拖放

##### 返回顶端

#### 1. 选择指令。



**注释:** 位于 IEC 指令前的红点表示指令是标准 IEC 指令的非标准扩展, 因为有一条相似的指令是 SIMATIC 指令集的一部分

糜翎媛菽康摹

2. 将指令拖至所需的位置。



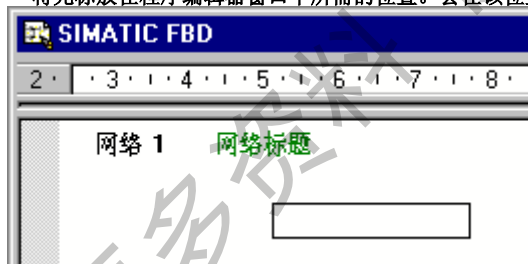
3. 松开鼠标按钮，在所需的位置放下指令。



注释：光标会自动阻止您在非法的位置放下指令（例如首列、网络标题或另一条指令的参数）。

#### 从指令树双击 [返回顶端](#)

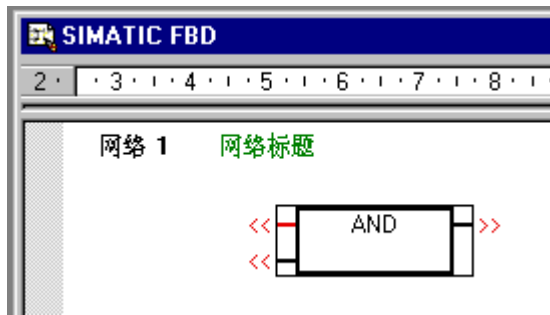
1. 将光标放在程序编辑器窗口中所需的位置。会在该位置周围出现一个选项方框。



2. 在指令树中，浏览至所需的指令并双击该指令。

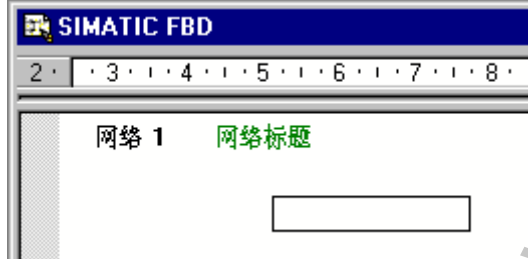


3. 双击指令后，指令出现在程序编辑器窗口中。

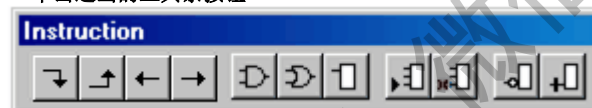


### 使用工具条按钮或功能键 [返回顶端](#)

1. 将光标放在程序编辑器窗口中所需的位置。会在该位置周围出现一个选择方框。



2. 单击适当的工具条按钮...



或使用适当的功能键（F4=AND（与）、F6=OR（或）、F9=通用方框）插入指令。

注释：不得将数值指定给类属指令框的参数；您必须如步骤3所示，选择一个具体的指令类型。

3. 会出现一个下拉列表。滚动或键入指令的前几个字母，浏览至所需的指令。双击所需的指令，或使用 ENTER 键，将其插入。（如果您这时不选择特定的指令种类，可以返回网络，单击类属指令的助记符区域（该区域包含 ???，而不是助记符），或选择指令，并按 ENTER 键，调回列表。）



### 3.2.12 为逻辑连接指定立即属性（直接I/O存取）

本标题讨论插入FBD指令的几种方法：

[指令树：拖放](#)  
[指令树：双击](#)  
[工具条按钮](#)  
[功能键](#)

当您插入指令时，可能还需要下列与本题目有关的信息：

[纠正指令位置](#)  
[如何划线](#)  
[插入与覆盖模式](#)

#### 相关标题:

[反转布尔输入](#)  
[将布尔输入或输出设为立即模式](#)  
[增加或移除附加输入 \(仅限AND/OR \(与/或\)\)](#)

#### 纠正指令位置: 从左至右作业

[返回顶端](#)

建立FBD网络时, 始终从左至右作业, 以便利用程序编辑器的特别功能:

- (当插入光标作为前一个方框的打开输出 >> 正确放置在同一行上时) 方框自动连接
- (当插入光标作为前一个方框的打开输出 >> 正确放置在同一行上时) 自动调整方框位置
- 自动调整方框尺寸 (以便启用正在插入的方框输入, 与前一个方框的打开输出 >> 对齐和连接)

当您从左至右作业, 并适当放置指令时 (将插入光标放在与前一条指令的输出 >> 相同的行中), 程序编辑器会自动连接和排列指令。

#### 为参数留出位置

将FBD程序编辑器视作由列和行组成的网格。

您不得将指令放置在网络的最上一行; 光标必须放在第二行或下方。另外, 如果指令左边有输入, 则不能在程序编辑器的第一行蟹胖弥噶脊 蛭 挥蟹胖貌问 奈恢谩



**提示:** 当您在程序编辑器中工作时, 留心FBD工具条; 如果将光标放在程序编辑器的一个不允许

放置特定类型线条或指令的区域中, 线条工具和指令按钮会被自动禁用。

**注释:** 将一条指令与另一条指令连接时, 不必在指令之间留出额外的空格。您可以在第一条指令的适当输出参数中 (>>) 直接放置第二条指令。

#### 将输出赋值更改为连接

如果您将参数值指定给指令的输出 >>, 但是后来决定将指令与另一条指令连接, 您可以直接在第一个输出上放置第二条指令。以前指定给该输出参数的数值 (例如Q0.1) 会被自动盖写, 输出与替换指令连接。

#### 划线

[返回顶端](#)

您可以使用程序工具条中的水平和垂直线, 或按住 CTRL 键并按键盘上的左、右、上或下箭头键, 必要时在网络元素之间划线 (例如, 建立分支)。



#### 插入与覆盖模式

[返回顶端](#)

STEP 7-Micro/WIN允许您在键盘上按“插入”键, 在两个编辑模式之间切换:

- 在插入模式 (当您按“插入”键时选择) 中, 如果您在另一条指令上放置一条指令, 程序编辑器移动现有指令, 为新指令留出位置。
- 在覆盖模式 (“插入”键未按下时的默认值) 中, 如果您在另一条指令上放置一条指令, 程序编辑器删除现有指令, 并用新指令替换现有指令。

#### 覆盖模式中的参数保留

- 如果您用具有相同概况的另一个方框替换 (覆盖) 一条指令, 您对旧参数所作的任何赋值都会转移至新参数。(即如果第二条指令具有与第一条指令相同的逻辑流输入数目、相同的输入地址参数数目、相同的逻辑流输出数目和相同的输出地址参数数目, 则当您用第二条指令覆盖第一条指令时, 参数赋值被保留。)
- 对于AND (与) 方框和OR (或) 方框, 即使您修改了原方框以便增加附加输入, 旧参数仍然保留。(例如, 如果您 ^ AND (与) 方框分配了四个输入赋值, 您用一个OR (或) 方框覆盖AND (与) 方框, OR (或) 方框会自动继承四个钟惶凉爵#

## 从指令树拖放

[返回顶端](#)

1. 选择指令。

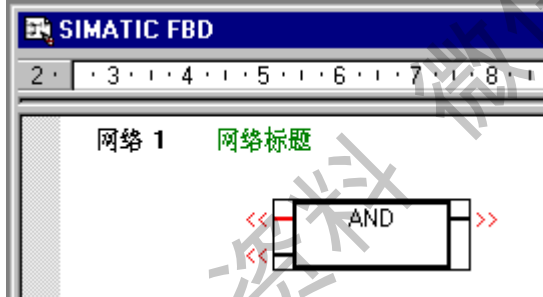


**注释：**位于IEC指令前的红点表示指令是标准IEC指令的非标准扩展，因为有一条相似的指令是SIMATIC指令集的一部分。

2. 将指令拖至所需的位置。



3. 松开鼠标按钮，在所需的位置放下指令。



**注释：**光标会自动阻止您在非法的位置放下指令（例如首列、网络标题或另一条指令的参数）。

## 从指令树双击

[返回顶端](#)

1. 将光标放在程序编辑器窗口中所需的位置。会在该位置周围出现一个选项方框。



2. 在指令树中，浏览至所需的指令并双击该指令。



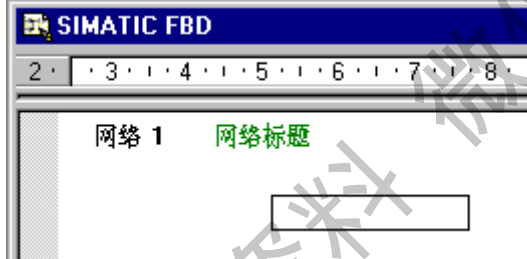


3. 双击指令后，指令出现在程序编辑器窗口中。

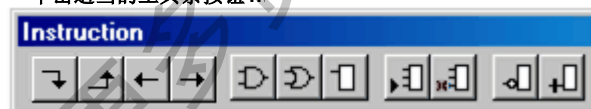


#### 使用工具条按钮或功能键 [返回顶端](#)

1. 将光标放在程序编辑器窗口中所需的位置。会在该位置周围出现一个选择方框。



2. 单击适当的工具条按钮...



或使用适当的功能键（F4=AND（与）、F6=OR（或）、F9=通用方框）插入指令。

注释：不得将数值指定给类属指令框的参数；您必须如步骤3所示，选择一个具体的指令类型。

3. 会出现一个下拉列表。滚动或键入指令的前几个字母，浏览至所需的指令。双击所需的指令，或使用 ENTER 键，将其插入。（如果您这时不选择特定的指令种类，可以返回网络，单击类属指令的助记符区域（该区域包含 ???，而不是助记符），或选择指令，并按 ENTER 键，调回列表。）



### 3.2.13 使用符号表帮助组织程序

使用下列方法之一打开符号表（用SIMATIC模式）或全局变量表（用IEC 1131-3模式）：



- 单击浏览条中的“符号表”按钮。
- 选择**检视 (View) > 符号表 (Symbol Table)** 菜单命令。
- 打开**指令树**中的符号表或全局变量文件夹，然后双击一个表格图标。

您可以使用项目中的**系统符号名**。预定义的系统符号表提供常用PLC系统功能的存取功能。PLC系统符号将功能名称与用于激活莓8芳 PLC特殊内存位置相联系。

本标题讨论下列信息：

[在表格中编辑](#)

[在表格中指定符号赋值](#)

[引号的使用](#)

[插入附加行](#)

[使用定义、编辑或选择符号命令](#)

[按照列排序](#)

[检视重叠和未使用的符号](#)

[使用多个表格](#)

[建立附加符号表](#)

[在符号表之间移动](#)

[POU符号标签](#)

[符号和绝对视图](#)

[在符号和绝对地址视图之间切换](#)

[同时检视符号和绝对地址](#)

[理解符号](#)

[理解符号范围](#)

[使用间接记号](#)

[内存类型](#)

[输入错误和故障排除](#)

[提示](#)

[SIMATIC与IEC](#)

[SIMATIC和IEC表格举例](#)

[在符号表 / 全局变量表中指定符号赋值](#)

[返回顶端](#)

欲为地址或常量值指定符号，请遵循下列步骤：

1. 打开符号表 / 全局变量表（使用标题开始部分描述的一个步骤）。
2. 在“符号名”列键入符号名（例如，Input1）。允许使用的最大符号长度为23个字符。使用TAB、ENTER或ARROW键确认，并移至下一个单元格。

**注释：**

\*

在为符号指定地址或常量值之前，该符号一直显示为未定义符号（绿色波浪下划线）。完成“地址”列赋值后，绿色波浪下划线被删除。

\*

如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在LAD、FBD和STL程序编辑器窗口中被一个波浪号(~)截断。您可将鼠标放在被截断的名称上，在工具提示中查看全名。

3. 在“地址”列中键入地址或数值（例如，V或I23）。
- （在IEC 1131-3编程模式中，键入地址后会自动增加正确的IEC“%”前缀）。
4. 如果您正在使用IEC全局变量表，在“数据类型”列的下拉列表选取一个数据类型。（SIMATIC用户无需提供数据类型。）
5. 键入注解（选项：最多允许79个字符）。

**注释：**


在STEP 7-Micro/WIN中，您可以建立多个符号表（SIMATIC编程模式）或多个全局变量表（IEC 1131-3编程模式）。但不允许将相同的符号名称多次用作全局符号赋值，在单个符号表和几个表内均不得如此。（相反，允许根据您的选择在多个不同局部变量表中多次使用相同的符号名称。）




[检视重叠和未使用的符号](#)


[返回顶端](#)



欲检视符号表中的“重叠”列或“未使用的符号”列：

- 选择工具 (**T**ools) > 选项 (**O**ptions) 菜单项目。
- 选择“符号表”标记。
- 选择适当的复选框 (“显示重叠符号”和“显示未使用的符号”)。

“重叠”列显示绝对地址共享部分或全部相同内存之符号行的重叠图标 。如果同一个字面值有多个已定义的符号常量，那该图标将显示为重叠图标。每次表格被修改时，“重叠”列被更新。

符号表			
· 3 · 1 · 4 · 1 · 5 · 1 · 6 · 1 · 7 · 1 · 8 · 1			
		符号	地址
1		符号3	V80
2		符号4	V80

“未使用的符号”列就在您的程序中未被引用的所有符号显示未用图标 。每次表格被修改时，该列被更新。

符号表			
· 3 · 1 · 4 · 1 · 5 · 1 · 6 · 1 · 7 · 1 · 8 · 1			
		符号	地址
1		符号1	I0.0
2		符号2	I0.1

**引号的使用**[返回顶端](#)

在3.1版之前，全局符号名称必须放在双引号内。双引号已不再用于符号名。自4.0版起，双引号用于在符号表中起始和结束作为撤 瑟 持档脍ASCII常量字符串。此赋值创建了一个符号ASCII字符串常量。与此相似，单引号用于起始和结束ASCII字符常量（只限字节、字、双字）以创建一个符号ASCII字符常量。

**插入附加行**[返回顶端](#)

使用下列方法之一在符号表 / 全局变量表中插入附加行：

- 选择菜单命令 **编辑 (E)dit** > **插入 (I)nsert** > **行 (R)ow**。将在符号表 / 全局变量表光标的当前位置上方插入新行。
- 用鼠标右键单击符号表 / 全局变量表中的一个单元格。选择弹出菜单命令 **插入 (I)nsert** > **行 (R)ow**。将在光标的当前位置上方插入新行。
- 欲在符号表底部插入新行，将光标放在最后一行的任意一个单元格中，按“下箭头”键。

**使用定义、编辑或选择符号命令**[返回顶端](#)

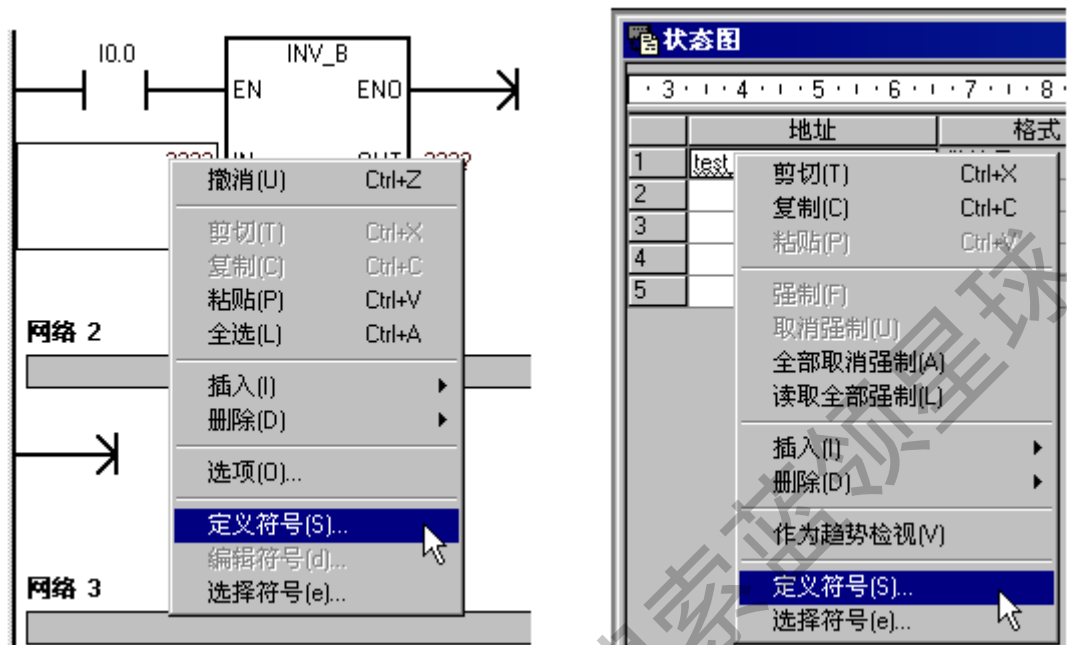
定义、编辑或选择符号等命令，允许您在使用程序编辑器或状态图时，定义新符号、从列表上选取现有符号、或编辑符号属性。新的或修改后的赋值将被自动加入到符号表内。



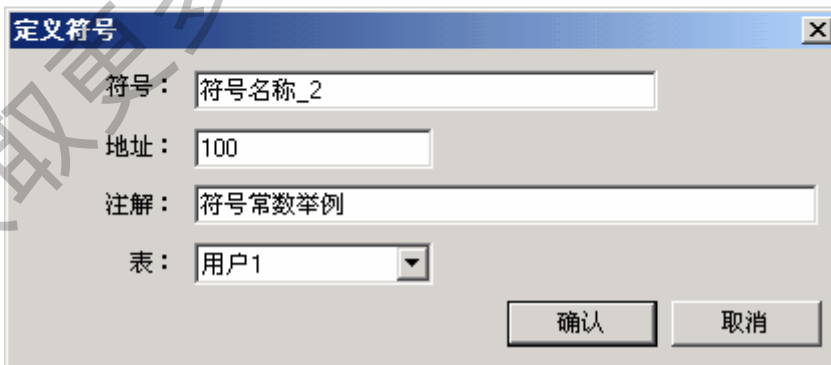
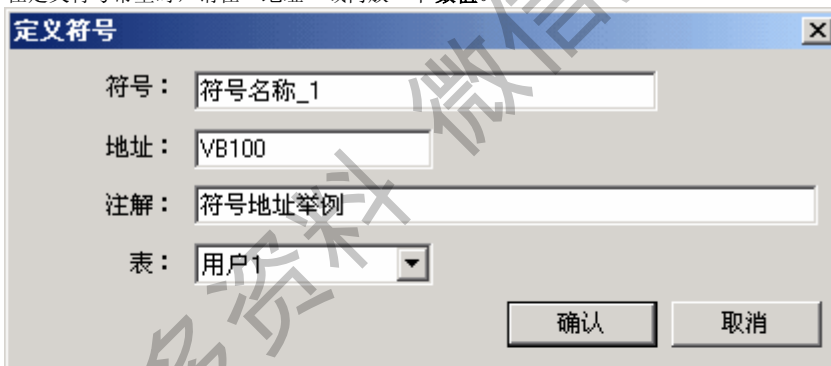
**提示：**只有在“符号编址”视图中工作时才能使用符号命令。（检查“检视”菜单下方的“符号编址”选项：标选符号表示已打开“符号编址”视图。）

欲激活“符号”命令：

1. 用鼠标右键单击对某指令参数或“状态图”地址单元来说不完全（或不正确）的符号名，然后从弹出菜单选择“定义”。



2. 定义新符号，编辑现有符号的属性，或由全部用户指定符号的概括中选取一个现有符号。  
在定义符号地址时，请在“地址”域内放一个地址。  
在定义符号常量时，请在“地址”域内放一个数值。



bml mw\_sym\_define\_b.gif)

3. 单击“确定”确认作业，并关闭对话框，或单击“取消”，取消选择。

**注释：**

- 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在 LAD、FBD 和 STL 程序编辑器窗口中被一个



波浪号 (~) 截断。您可将鼠标放在被截断的名称上，从而在工具提示中查看全名。

### 按照列排序

#### [返回顶端](#)

您可按照“名称”或“地址”列排序表格，排序时可按正向或逆向（字母）顺序排列。此表中的符号常量将在与地址关联的最后一个符号后面列出。所有非字符类型的符号常量接着将独立于显示格式地按数字顺序排列。ASCII常量将出现在数字常量之后，地址 A 拷 罇 蟪 魃 帧

您可以使用检视菜单对某列进行排序。请单击列身以选取该列，请勿单击列顶的列标题（“名称”或“地址”）。

- 若按正向顺序（A至Z）排序列，单击“排序”  按钮，或选择菜单命令 **检视 (View) > 正向排序 (Sort Ascending)**。
- 若按逆向（Z至A）顺序排序列，单击“逆向排序”  按钮，或选择菜单命令 **检视 (View) > 逆向排序 (Sort Descending)**。

若要按相反顺序排序，单击列顶部的列头（“名称”或“地址”）。当您单击列头时，将在正向和逆向排序之间切换。


### 建立附加符号表

#### [返回顶端](#)

默认情况下，符号表窗口为用户定义的符号名称（USR1）显示一个标记。如果您对任何POU重新命名，符号表窗口显示一个被称为“POU符号”的标记—如果您在使用“协议”功能，例如USS协议库，则窗口显示“USS协议”标记。

可采用下列几种方法为用户定义的符号名称建立附加符号表。

- 从指令树用鼠标右键单击“符号表”文件夹，并选择弹出菜单命令 **插入符号表 (Insert Symbol Table)**。
- 打开符号表窗口，并使用“编辑”菜单，或用鼠标右键单击，调出弹出菜单，并选择 **插入 (Insert) > 表格 (Table)**。
- 您在建立程序网络时，可指定符号名，方法是在程序编辑器的指令参数域中键入有效的符号名。此种方法建议一组不

付 讷 嫫 卬 返 姆 琶 5 弋 鳌 敖 00 锤 持 捣 疔 恻 卑 磁   
，将该组符号名传送至您指定地址的新“符号表”标记。欲为符号表重新命名，用鼠标右键单击指令树中的符号表标记，并选择“重新命名”，或双击“符号表”标记名。


**注释：**成功地插入新符号表后，一个新标记

会出现在符号表窗口的底部。在检视符号表时，请核实选择了正确的标记。您可以双击或用鼠标右键单击标记，为标记重新命名。

### 在符号表之间移动

#### [返回顶端](#)

如果建立了多个符号表，您可以使用下列任意一种方法存取表格：

- 如果已经打开符号表窗口，单击位于符号表窗口底部的标记 ，在符号表之间移动。
- 从指令树扩充“符号表”文件夹图标，并双击所需的“符号表”图标，检视符号表窗口中该表的标记。

### POU符号标记

#### [返回顶端](#)

如果您已经为项目的任何POU或其他元件（例如数据块、状态图或符号表）指定符号名，将在符号表窗口的“POU符号”标记中列出符号名赋值。

该标记为只读标记；您不能在此编辑赋值。

如果您希望更改赋值，您必须编辑该元件的“属性”对话框。（从指令树的“项目”分支，用鼠标右键单击元件，调出“属性”对话框）

所有符号赋值必须使用有效语法。（请参阅下面的 [输入错误](#)）。如果指定符号名赋值时违反了指南规定，编译程序时 STEP 7-Micro/WIN会报告错误。

### 在符号和绝对地址视图之间切换

#### [返回顶端](#)

在符号表 / 全局变量表中建立符号和绝对地址或常量值的关联后，您可在操作数信息的符号和绝对显示之间切换。可采用下列

- 选择菜单命令 **检视 (View) > 符号编址 (Symbolic Addressing)**，在符号编址打开或关闭之间切换
- 使用 **Ctrl+Y** 快捷键在符号编址打开或关闭之间切换

“符号编址”菜单项目前面的标选标记表示已打开符号编址。默认条件下，当您打开第一个项目时，符号编址也被打开。

您不能在检视符号常量或其关联的字面值之间切换。这是因为您可以为同一个字面值定义多个符号常量。因此，假如您可以 ‘  
STEP 7-  
Micro/WIN 无法可靠地恢复原有的符号常量。出于此原因，禁用符号编址（通过主菜单或按 CTRL+Y）将不会影响您项目中符号常量的显示。出于同样原因，如果您为某操作数输入了该常量的字面值，STEP 7-Micro/WIN 将不会自动套用已定义的符号常量。

### 同时检视符号和绝对地址

[返回顶端](#)

欲在 LAD、FBD 或 STL 程序中同时检视符号地址和绝对地址，使用菜单命令 **工具 (Tools) > 选项 (Options)**，并选择“程序编辑器”标记。选择“显示符号和地址”。



- 提示：**
- 只有在打开 **符号地址视图** 时，才能在项目中显示符号地址。否则，即使您选择了“显示符号和地址”，也只会显示绝对地址。
  - 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在 LAD、FBD 和 STL 程序编辑器窗口中被一个波浪号 (~) 截断。您可将鼠标放在被截断的名称上，在工具提示中查看全名。

### 理解符号范围

[返回顶端](#)

符号表 / 全局变量表将符号名指定给 PLC 内存或 I/O 地址。在符号表 / 全局变量表中定义 **符号** 时，符号具有全局范围。这意味着您可在任何 **POU** 中使用该符号名称，代表位于该符号地址的数据。（相反，如果您使用局部变量表指定符号名称，则只能在定义该局部变量的 POU 范围中使用该局部变量。）

### 使用间接记号

[返回顶端](#)

与绝对地址相同，您可在符号中使用间接记号 (& 和 \*)。欲了解有关间接编址的详情，请参阅 [直接和间接编址](#)。

### 内存类型

[返回顶端](#)

您可为下列内存类型建立符号名：I、Q、M、SM、AI、AQ、V、S、C、T、HC

### 输入错误

[返回顶端](#)

输入错误用色彩显示，很容易发现：

- 非法语法 — **红色文字** 和 **红色波浪下划线**

例如：

	符号	地址
1	2name	I0.0
2		VBO
3	Begin	I0.2

（非法地址 VBO 是正确的）（逆转的关键字无法用作符号）

（第一个字符的号码字符非法）

- 非法使用 — **红色波浪下划线**

例如：

	符号	地址
1	Pump1	I0.0
2	Pump1	I0.0
3	SymConstant	123456
4	SymConstant	123456
5	SymConstant2	654321
6	SymConstant3	654321

(重复名称) (重复地址)

- 未定义符号 — 绿色波浪下划线

例如:

	符号	地址
1	<u>Pump1</u>	
2	Pump2	V0
3	Factor	3.14E-006
4	Neg_Value	-3333333

(名称带有无效地址) SIMATIC模式

(名称带有无效地址或数据类型) IEC模式

**注释:**

- 符号名称可包含某些字母数字字符和下划线，以及某些扩展字符（ASCII 128至ASCII 255）。但第一个字符只能是字母或扩展字符。
- 在3.1版之前，全局符号名称必须放在双引号中。双引号已不再用于符号名。自4.0版起，双引号用于在符号表中起始和结束作为某符号名之赋值的ASCII常量字符串。此赋值创建了一个符号ASCII字符串常量。与此相似，单引号用于起始和结束ASCII字符常量（仅限字节、字、双字）以创建一个符号ASCII字符常量。
- 不允许使用关键字作为符号名，也不允许第一个字符是数字或包含非字母数字字符或扩展字符集。
- 符号名称的最大许可长度为23个字符。
- 如果您希尝试建立符号赋值，以纠正程序中操作数“未定义符号”错误，但键入地址数值后忘记按 TAB键、ENTER键  
蚣 芳 蹦 禱 爻 繇 蚩 唳 魔 保 僮 魔 廊 幌 允 尽 拔 炊 丁 宸 拧 贝 砦 蟆 N 私 伎 稣 庖 晃 侍 猓 禱 胤 疔 荣 翱 冢 獾 支  
旁 淞 暗 司 贰 钡 卮 襪 校 TAB、ENTER或箭头键，完成地址赋值。

**建立符号提示**[返回顶端](#)

- 您可在任何时间指定符号赋值：在建立程序逻辑之前、之间或之后。
- 您可在“注解”列中增加注释，记录您的符号。
- 编程时，可使用“定义、编辑或选择符号”命令从列表选择新符号名或建立新符号名。
- 通过单击不同的表列，可对符号表 / 全局变量表中的数值进行检索。
- 通过拖拉边缘，可重定列尺寸。
- 您可使用文件 (File) > 打印 (Print) 命令，并选择适当的复选框，打印符号表 / 全局变量表。
- 您一旦建立符号赋值，就可以通过启用符号信息表功能，在 LAD/ FBD /STL 程序编辑器中的每隔网络滞后显示地址赋值和符号注解。

**SIMATIC和IEC表举例**[返回顶端](#)

符号表 (SIMATIC编程模式) 和全局变量表 (IEC编程模式) 在功能上的唯一区别是“数据类型”列，此为 IEC编程的独特功能。

IEC-

1131是一种加强类型语言。指令操作数被定义为接受一种指定的数据类型。欲了解有关数据类型的详情，请参阅 [SIMATIC和IEC 1131-3数据类型](#)。

SIMATIC符号表举例:

符号表				
· 3 · · 4 · · 5 · · 6 · · 7 · · 8 · · 9 · · 10 · · 11 · · 12 · · 13 · · 14 · · 15 · · 16 · ·				
		符号	地址	注解
1		动1	I0.0	
2		动2	I0.1	
3		停止1	I0.2	
4		停止2	I0.3	
5		高等级	I0.4	
6		低等级	I0.5	
7		复位	I0.7	
8				
9		泵1	Q0.0	
10		泵2	Q0.1	
11		搅拌机电机	Q0.2	
12		蒸阀	Q0.3	
13		排泄阀	Q0.4	
14		排泄泵	Q0.5	
15				
16		达到高等级	M0.1	
17		搅拌机计时	T37	
18		循环计数	C30	
19		Mix_Time_Minutes	15.3	
20		Mix_Temp_Cent_min	90	
21		Mix_Temp_Cent_max	95	
22		Process_ID	"P418369B"	

USER1 / POU Symbols

IEC全局变量表举例：



符号	地址	数据类型	注解
1	Start_1	%I0.0	BOOL
2	Start_2	%I0.1	BOOL
3	Stop_1	%I0.2	BOOL
4	Stop_2	%I0.3	BOOL
5	High_Level	%I0.4	BOOL
6	Low_Level	%I0.5	BOOL
7	Reset	%I0.7	BOOL
8			
9	Pump_1	%Q0.0	BOOL
10	Pump_2	%Q0.1	BOOL
11	Mixer_Motor	%Q0.2	BOOL
12	Steam_Valve	%Q0.3	BOOL
13	Drain_Valve	%Q0.4	BOOL
14	Drain_Pump	%Q0.5	BOOL
15			
16	High_Lev_Reached	%M0.0	BOOL
17	Mix_Timer	%T37	TON
18	Cycle_Counter	%C30	CTU
19	Counter_Out	%M0.2	BOOL
20	Counter_Current	%M\W3	INT
21	Mix_Time_Minutes	15.3	REAL
22	Mix_Temp_Cent_min	9	BYTE
23	Mix_Temp_Cent_max	95	BYTE
24	Process_ID	"P418369B"	STRING

[返回顶端](#)

另请参阅:

[局部变量表](#)

[检视符号编址](#)

[网络符号信息表 \(LAD/FBD\)](#)

[编址概述](#)

[直接和间接编址](#)

[关键字列表](#)

[系统符号表](#)

[符号表标记 \(工具 > 选项\)](#)

### 3.2.14 为每个网络显示符号信息表

使用下列方法之一检视或隐藏LAD/FBD/STL程序编辑器窗口中的符号信息表:

- 单击  工具条按钮。
- 选择菜单命令 **检视 (View) > 符号信息表 (Symbol Information Table)**。
- 按 **Ctrl+T** 快捷键组合。
- 单击 **在项目中应用全部符号**  工具条按钮。应用全部符号命令不仅用所有的新、旧和修改符号名更新项目，而且在符号信息表打开和关闭之间切换。

您一旦选择了 **检视 (View) > 元件 (Component) > 符号表 (Symbol Table)** 菜单项目，所有打开的项目均会显示网络符号信息，直至您取消该菜单项目。

当您检视符号信息表时，符号名、绝对地址、数据类型和注解会在 LAD/FBD/STL 程序中每个网络下方按字母顺序显示。未定义

的符号名也包括在内。

该表列出该网络中所有符号的信息；不包含全局符号的网络不显示符号信息表。所有重复条目均被移除。

如果您位于IEC模式，表格显示一个附加数据类型列。

符号信息表无法编辑。

**网络 1 用涂料成分1和监视器罐填充储罐。**

起始\_1:I0.0    停止\_1:I0.2    高\_位:I0.4    泵\_1:Q0.0

泵\_1:Q0.0

泵_1	Q0.0	泵1控制 (1 = 运行, 0 = 停止)
高_位	I0.4	混合罐料位过高
起始_1	I0.0	泵1起始命令
停止_1	I0.2	泵1停止命令

**网络 2 用涂料成分2和监视器罐填充储罐。**

起始\_2:I0.1    停止\_2:I0.3    高\_位:I0.4    泵\_2:Q0.1

泵\_2:Q0.1

泵_2	Q0.1	泵2控制 (1 = 运行, 0 = 停止)
高_位	I0.4	混合罐料位过高
起始_2	I0.1	泵2起始命令
停止_2	I0.3	泵2停止命令




**提示：**用光标键浏览符号信息。

您可以从列边缘拖动分割器光标，重新确定表格大小。表格不得超过 32 个单元格。所有

无法放入一个单元格中的文本均以波浪号结束；工具提示显示所有无法容纳的文本。

打印程序时，使用“打印”对话框中的“选项”按钮启用符号信息表打印。

为了节省垂直方向的空间并检视更多的程序网络，您可以在符号信息表打开和关闭之间切换，方法是使用**检视 (View) >**

**元件 (Component) > 符号表 (Symbol Table)** 命令，或单击在项目中应用全部符号  工具条按钮。

使用“查找”功能替换文本。

#### 故障排除

如果在LAD/FBD/STL程序网络的下方未显示符号信息表，请考虑下列因素：

- 是否已启用符号信息表？检查“检视”菜单：符号信息表项目旁应当有一个标选符号。

- 网络中是否使用符号（或与符号对应的绝对地址）？不包含符号数值的网络不会显示符号信息表。
- 是否在符号表（IEC全局变量表）中输入了符号？在您完成赋值之前，不会显示地址或注解信息。

另请参阅：

[符号表 / 全局变量表](#)

[符号表标记（工具 > 选项）](#)（在您检视程序状态时，调整符号信息表的显示方法）

[应用程序用户参考手册](#)

[入门指南目录](#)

### 3.2.15 SIMATIC和IEC 1131-3数据类型

在STEP 7-Micro/WIN中编程时，在下列两种情况下您必须理解并使用数据类型：

- 如果您以IEC 1131-3模式编程，则必须为全局符号表中的所有数值选择数据类型。
- 如果您在局部变量表中赋值，则必须为每个局部变量定义数据类型。

当您明确地为数值指定数据类型时，您为STEP 7-

Micro/WIN提供清晰的指令，指明需要为数值分配多少内存空间（例如，数值100可被存储为字节、字或双字）以及如何表示数值（将0视作布尔值还是数值？）。

每个SIMATIC和IEC 1131-

3指令或参数化子例行程序均由精确定义识别。该定义可被称为签名。对于所有的标准指令，每个指令操作数允许使用的数据类型集均从签名获得。对于参数化子例行程序，用户通过局部变量表建立子例行程序签名。

STEP 7-Micro/WIN V3.0编程软件为SIMATIC模式执行简单的数据类型检查，为IEC 1131-

3模式执行细致的数据类型检查。这意味着为局部或全局变量指定一种数据类型时，软件在下表所列的每个等级中确保操作数数

基本数据类型	数据类型大小	说明	范围
布尔	1位	布尔	0至1
字节	8位	不带符号的字节	0至255
字节	8位	带符号的字节（SIMATIC模式仅限于SHRB指令）	-
128至+127			
字	16位	不带符号的整数	0至65,535
整数	16位	带符号的整数	-32768至+32767
双字	32位	不带符号的双整数	0至4294967295
双整数	32位	带符号的双整数	-2147483648至 +2147483647
实数	32位	IEEE 32 位浮点	+1.175495E-38至 +3.402823E+38 -1.175495E-38至
3.402823E+38			
字符串	2至255字节	ASCII字符串照原样存储在PLC内存中，形式为1字符串长度接ASCII数据字节	
	ASCII字符代码128至255		
复杂数据类型	说明		范围
TON	打开延迟计时器	1 ms 10 ms 100 ms	T32, T96 T33 - T36, T97 - T100T37 - T63, T101 -
T255			
TOF	关闭延迟计时器	1 ms 10 msms	T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 -
255			
TP	脉冲计时器（参阅注释1）	1 ms 10 msms	T32, T96 T33 - T36, T97 - T100T37 -
T63, T101 - 255			
CTU	向上计数器		至255
CTD	向下计数器		至255
CTUD	向上 / 向下计数器		至255
SR	设置主要双稳态		无关
RS	重置主要双稳态		无关

#### STEP 7-Micro/WIN 提供三级数据类型检查

1. **IEC编辑器具有细致数据类型检查。**在该模式中，参数数据类型必须与符号或变量的数据类型完全匹配。欲支持细致数据类型检查，只有被指定数据类型的数据类型的变量才能成功编译。使用细致数据类型检查时，数据类型为整数的变量对字指令参数无效。细致数据类型检查仅在IEC 1131-3模式中进行。

**用户选择的数据类型**                      **等同数据类型**

布尔	布尔
字节	字节
字	字
整数	整数
双字	双字
双整数	双整数
实数	实数
字符串	字符串

2. **处于SIMATIC模式中的局部变量表具有简单数据类型检查。** 在该模式中，当符号或变量被指定一个数据类型时，同时自动指定与所选数据类型位大小相匹配的所有数据类型。例如，如果用户选择双整数作为数据类型，局部变量会自动指定数据类型双字，因为二者均为32位数据类型。不自动指定实数数据类型，虽然实数也是32位数据类型。实数数据类型被定义为无其他等同数据类型，总是独特的类型。使用局部变量时，只在SIMATIC模式中执行简单数据类型检查。

**用户选择的数据类型**                      **等同数据类型**

布尔	布尔
字节	字节
字	字、整数
整数	字、整数
双字	双字、双整数
双整数	双字、双整数
实数	实数
字符串	字符串

3. **SIMATIC符号编辑器无数据类型检查。** 该模式仅限用于无法选择数据类型的SIMATIC全局变量。在该模式中，所有相同大小的数据类型被自动指定给符号。例如，编程软件将自动为赋予VD100地址的符号指定下列数据类型：双字、双整数和实数。

**为SIMATIC全局符号由大小决定的数据类型****用户选择的地址**                      **指定等同的数据类型**

V0.0	布尔
VB0	字节、字符串
VW0	字、整数
VDO	双字、双整数、实数

**数据类型检查的益处**

数据类型检查的益处是帮助用户避免常见编程错误。例如，如果指令支持带符号的数字，则在使用不带符号的数字作为指令操作数时，软件为该用法设置旗标会有益处。例如，关系比较<I是带符号的指令，显然，对于带符号的数据类型操作数，-1小于0。但当<I指令允许支持不带符号的数据类型时，程序员需确保绝不会发生下列情况。在程序运行时，对于<I指令，不带符号的40,000实际上小于0。程序员需确保对带符号的指令使用不带符号的数字不会超越正负数界限，否则会出现无法预测的结果。



**警告** 您应当确保带符号的数字用作不带符号的指令时不得超出正负数界限。若无法确保不带符号的数字用作带符号的指令时不超出正负数界限，可能致使程序或控制器操作出现无法预测的结果。无法预测的控制器操作可能导致人员死亡或严重伤害及/或财产严重损坏。请务必核实不带符号的数字用作带符号的指令时不得超出正负数界限。

总之，在IEC 1131-3编辑模式中，细致数据类型检查通过对指令的非法数据类型生成错误，帮助程序员在编译过程中发现这些错误。SIMATIC编辑器不具有此一功能。

**在SIMATIC与IEC1131-3之间移植程序**

因为IEC1131-3属于细致数据类型，而SIMATIC不属于此一类型，STEP 7-Micro/WIN不提供在两种不同编辑模式之间移动程序的能力。程序员必须选择一种希望使用的编辑模式。

**超载指令**

超载指令支持一系列数据类型。仍然应用细致数据类型检查，在成功编译指令之前所有的操作数数据类型必须匹配。例如，请参阅下表中用于IEC超载ADD（加）指令。

<b>指令</b>	<b>允许使用的数据类型（细致数据类型检查）</b>	<b>允许使用的数据类型（数据</b>
-----------	----------------------------	---------------------

## 类型检查)

## 编译指令

ADD	整数	字、整数	ADD_I (加整数)
ADD	双整数	双字、双整数	ADD_D (加双整数)
ADD	实数	实数	ADD_R (加实数)

当所有操作数的数据类型均为双整数时，编译器生成“加双整数”指令。如果超载指令数据类型混合，会出现编译错误。根据数据类型检查的等级决定哪些为非法。在下例中，如果使用细致数据类型检查，则生成编译程序错误，但如果使用简单数据类型检查则成功编译。

ADD IN1 = INT, IN2 = WORD, IN3 = INT.

细致数据类型检查：引起编译错误。

简单数据类型检查：成功编译至ADD\_I (加整数)。

简单数据类型检查不会阻止出现常见运行时间编程错误。例如，使用简单数据类型检查时，编译程序不能帮助避免在运行时间(5) 韵鲁< 暗檀碧媳 ADD 40000,1被解释为负数，而不是不带符号的40,001。

## 在IEC中为超载指令使用直接编址

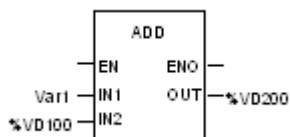
## IEC-

1131编程模式也允许使用直接表示的内存位置，作为指令参数配置的一部分。参数中可使用变量和内存位置。但是，使用直接表示的内存位置时必须清楚地理解数据类型的隐含意义，因为这些位置不包含明确的类型信息。另外，不能从任何超载 IEC指

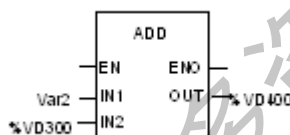
直接表示的参数数据类型通过检查指令中包含的其他类型的参数来决定。当指令参数配置使用某一指定类型的变量时，则假定

名称	地址	数据类型	注解
Var1		实数	这是浮点数变量。
Var2		双整数	这是双整数变量。
Var3		整数	这是整数变量。

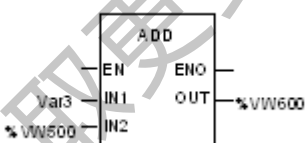
## 举例 说明



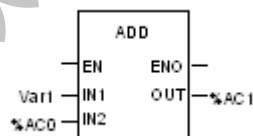
假定VD100和VD200为实数类型，因为Var1是实数类型。



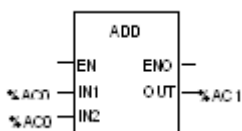
VD300和VD400为双整数类型，因为Var2是双整数类型。



假定VW500和VW600为整数类型，因为Var3是整数类型。

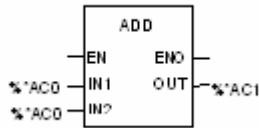


假定AC0和AC1为实数类型，因为Var1是实数类型。



这种配置被视作非法，因为无法确定类型。累加器中的数据类型可为双整数或

实数。



数或实数。

这种配置被视作非法，因为无法确定类型。累加器指针中的数据类型可为双整数或实数。

### 使用数据类型转换指令

转换指令允许从一种数据类型移至另一种数据类型。STEP 7-Micro/WIN支持在下列简单数据类型之间的移动数值转换指令。

转换指令	IEC细致数据类型检查 允许的操作数	SIMATIC局部变量表中的简单数据类型检查
字节至整数	入： 字节 出： 整数	入： 字节 出： 字、整数
整数至字节	入： 整数 出： 字节	入： 字、整数 出： 字节
整数至双整数	入： 整数 出： 双整数	入： 字、整数 出： 双字、双整数
双整数至整数	入： 双整数 出： 整数	入： 双字、双整数 出： 字、整数
双整数至实数	入： 双整数 出： 实数	入： 双字、双整数 出： 实数
实数至双整数（进位）	入： 实数 出： 双整数	入： 实数 出： 双字、双整数
字符串至整数	入： 字符串 出： 双字（*指向字符串的指针） 出： 整数	入： 字符串，
双字（*指向字符串的指针）出： 字、整数		
整数至字符串	入： 整数 出： 用双字代表的字符串（*指向字符串的指针）	入：
字、整数 出： 用双字代表的字符串（*指向字符串的指针）		
字符串至双整数	入： 字符串 出： 双字（*指向字符串的指针） 出： 双整数	入： 字符串，
双字（*指向字符串的指针）出： 双字、双整数		
双整数至字符串	入： 双整数 出： 用双字代表的字符串（*指向字符串的指针）	入：
双字、双整数 出： 用双字代表的字符串（*指向字符串的指针）		
字符串至实数	入： 字符串 出： 双字（*指向字符串的指针） 出： 实数	入： 字符串
双字（*指向字符串的指针）出： 实数		
实数至字符串	入： 实数 出： 用双字代表的字符串（*指向字符串的指针）	入： 实数
出： 用双字代表的字符串（*指向字符串的指针）		

在IEC 1131-

3编辑模式中，您可以使用超载移动指令在整数和字、双整数和双字之间转换。超载移动指令允许相同大小的数据类型自由移动。

IEC 1131-3超载移动	入	出
移动（整数至字）	整数	字
移动（字至整数）	字	整数
移动（双整数至双字）	双整数	双字
移动（双字至双整数）	双字	双整数

## 3.2.16 使用“查找”和“替换”进行修改

欲使用查找/替换或转入：

- 选择**编辑 (Edit)** > **查找 (Find)**、**编辑 (Edit)** > **替换 (Replace)** 或 **编辑 (Edit)** > **转入 (GoTo)** 菜单命令
- 按CTRL+F执行查找，按CTRL+H执行替换或按CTRL+G执行转入

如何使用：

**查找**  
**替换**

在何处使用

可以在程序编辑器窗口、局部变量表，符号表、状态图、交叉引用标记和数据块中使用“查找”、“替换”和“转入”。

如何作业

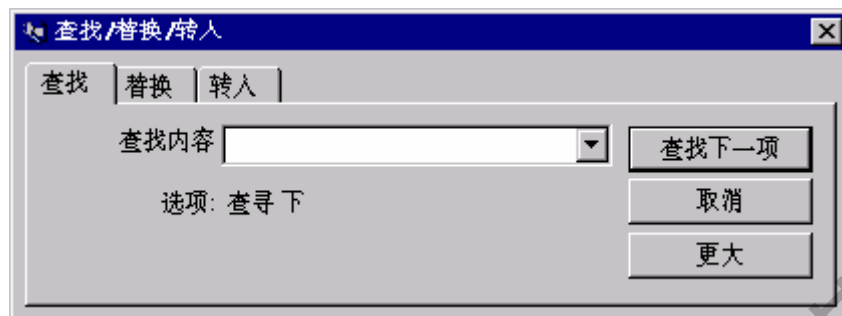
- “查找”功能允许您查找指定的字符串，例如操作数、网络标题或指令助记符。（“查找”不搜索网络注解，仅搜索

凌晏俊！安榆摇辈凰阉影 LAD和FBD中提供的网络符号信息表。）

- “替换”功能允许您替换指定的字符串。（“替换”不对指令助记符操作。）
- “转入”功能允许您通过指定网络数目或您希望浏览的行迅速移至另一个位置。

#### 查找功能

[返回顶端](#)



1. 在“查找内容”域中键入要搜索的字符串。
2. 欲移动下一个搜索字符串，单击“查找下一项”按钮。

**注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反映了操作数在代码中的存储方式。另外，如果您在 STL 中建立的程序包含在 LAD 或 FBD 编辑器中非法的网络，在 LAD 或 FBD 编辑器中查看该程序时，“查找下一项”命令不对这些网络执行搜索操作。

欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方式定制搜索：

- 您可以使用“搜索”列表框选择搜索方向。
- 您可以选择“大小写匹配”复选框，仅搜索与您键入的字符串大小写数值相同的字符串。
- 您可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的 SPAN（扩展）复选框，对所有的 POU（OB1、所有的子例行程序和中断例行程序）或局部变量表的所有实例、符号表或状态图进行搜索。
- 您可以指定对一定的行范围进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认段 B 荒 部梢约 驴 缙蚰泻牛 魁 阉鞞钠鸮已徒嘎
- 您可以指定是否通过选择适当的复选框搜索网络标题、POU 和网络注释和 / 或程序代码。

#### 替换功能

[返回顶端](#)



1. 在“查找内容”域中键入要搜索的字符串。
2. 在“替换内容”域中键入您希望用作替换搜索字符串的字符串。
3. 欲查找出现的搜索字符串，单击“查找下一项”按钮。

**注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反映了操作数在代码中的存储方式。另外，如果您在 STL 中建立的程序包含在 LAD 或 FBD 编辑器中非法的网络，在 LAD 或 FBD 编辑器中查看该程序时，“查找下一项”命令不对这些网络进行搜索操作。

4. 如果您希望替换字符串，单击“替换”。如果您已经仔细地定义搜索字符串而且没有误改的可能，您可以单击“全部替换”。

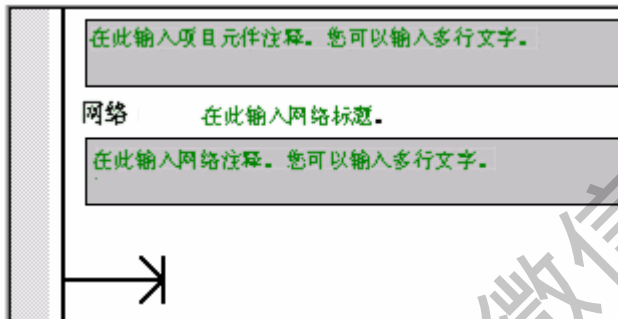
欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方法定制搜索：

- 您可以使用“搜索”列表框选择搜索方向。
- 您可以选择“大小写匹配”复选框，仅搜索与您在“查找内容”中键入字符串大小写数值相同的字符串。
- 您可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的SPAN（扩展）复选框，对所有的POU（OB1、所有的子例行程序和中断例行程序）或局部变量表的所有实例、符号表或状态图进行搜索。
- 您可以指定对一定的行范围进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认段。B 荒 部梢约 庐 缙蚰泻牛 魁 阉鞞钠鹦己徒嘎
- 您可以指定是否通过选择适当的复选框搜索网络标题、POU和网络注释和 / 或程序代码。

### 3.2.17 在程序中增加注解

FBD编辑器中共有四个注释级别：

- 项目元件注释
- 网络标题
- 网络注释
- 项目元件属性



#### 项目元件注释

在网络1上方的灰色方框中单击，输入POU注释。

您可以单击“切换POU注释”按钮  或选择和取消选择 **检视 (View) > POU注释 (POU Comments)** 选项，在POU注释ON（打开）（可视）或OFF（关闭）（隐藏）之间切换。


每条POU注释允许使用的最大字符数为4096。POU注释为供选用项目，可视时，始终位于POU的顶端，在第一个网络之前显示。

#### 网络标题

将光标放在网络标题行中的任何位置，输入一个识别逻辑网络的标题。网络标题中允许使用的最大字符数为127。

#### 网络注释

在网络1下方的灰色方框中单击，输入网络注释。您可以输入一条识别逻辑网络的注释，并输入一条有关网络内容的说明。

您可以单击“切换网络注释”按钮  或选择和取消选择 **检视 (View) > 网络注释 (Network Comments)** 选项，在网络注释ON（打开）（可视）或OFF（关闭）（隐藏）之间切换。

网络注释中允许使用的最大字符数为4,096。

#### 项目元件属性

您可以用以下任意一种方法存取“属性”标记：

用鼠标右键单击指令树中的 **POU**，并从鼠标右键菜单中选择“属性”。





用滑鼠右键单击程序编辑器窗口中的任何POU标记，并从弹出菜单选择“属性”。



在“属性”对话框中有两个标记：

- 一般
- 保护



#### 一般标记

“一般”标记允许您为子例行程序、中断例行程序和主程序块（OB1）重新编号和重新命名，并为项目指定一个作者。

#### 注释：

您不得将默认名称（由STEP 7-Micro/WIN指定的POU地址，例如SBR1用于子例行程序，或INT1用于中断例行程序）用作符号名，因为这样会构成重复赋值。如果您违反了符号名赋值的指南，STEP 7-Micro/WIN会在您尝试编译程序时报告一则错误。

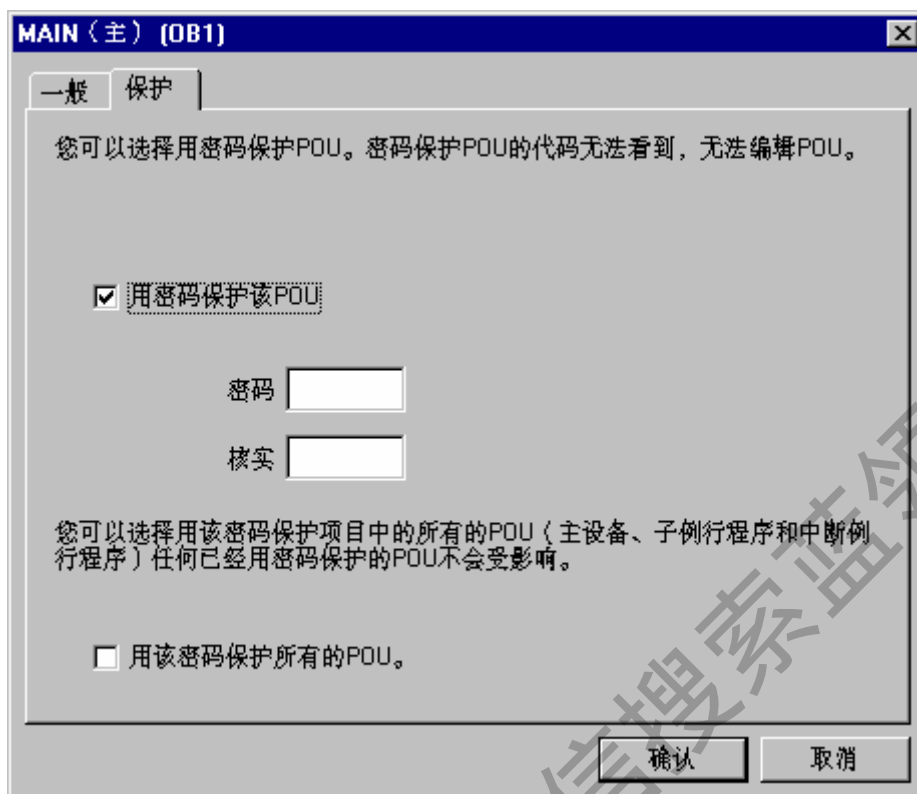
如果您在程序中为POU指定符号名，符号名会在程序代码中显示，即使“符号编址”视图未启用亦如此。符号表显示一个列出所有符号名赋值的特别标记（“POU符号”）。您只能检视该标记，无法编辑符号表中的条目。欲改动赋值，您必须编辑适当POU的“属性”对话框。

#### 保护标记

“保护”标记允许您选择用密码保护POU，使其他用户无法看到，并在下载时加密。

欲用密码保护POU：


1. 用鼠标右键单击“主POU”，并选择“属性...”
2. 选择“保护”标记。
3. 选择“用密码保护该POU”复选框。输入并核实由四个字符组成的密码。



您还可以选择“用密码保护所有POU”复选框，保护项目中的所有POU（主程序、子例行程序和中断例行程序）。

当POU受保护时，以下列方式在指令树中显示：



此外，还会在程序编辑器窗口的装订线中显示一个灰色锁定图标 .

欲取消锁定POU：

1. 浏览至“保护”标记。
2. 输入密码。
3. 单击“授权”。

您必须为您选择保护的每个POU重复该步骤。



### 3.2.18 理解S7-200计时器指令

您可利用计时器执行时基计数功能。S7-200指令集提供三种不同类型的计时器。

- 打开延迟计时器 (TON), 用于单间隔计时
- 保留性打开延迟计时器 (TONR), 用于累计一定数量的定时间隔
- 关闭延迟计时器 (TOF), 用于延长时间以超过关闭 (或假条件), 例如马达关闭后使马达冷却。

计时器操作:

计时器类型	当前值 >= 预设值	启用输入“打开”	启用输入“关闭”
电源循环 / 首次扫描			
TON	计时器位打开, 当前值继续计数 直至达到32,767		当前值记录时间
当前值=0	计时器位关闭, 当前值=0		
TONR	计时器位打开, 当前值继续计数 直至达到32,767		当前值记录时间
值保持最后的状态	计时器位关闭, 可保持当前值(1)		计时器位及当前值
TOF	计时器位关闭, 当前值=预设值, 停止计数		计时器位打开, 当前值=0
从“打开”转换为“关闭”后计时器开始计数		计时器位关闭, 当前值=0	

(1) 可通过电源循环为保留目的选择保留性计时器当前值。请参阅 [保留范围标记—系统块配置](#) 中有关S7?00 CPU的内存保留信息。

请参阅文档光盘“提示和技巧”中的提示31, 查阅使用打开延迟计时器 (TON) 的抽样程序。

注释:

可用“重置” (R) 指令重置任何计时器。“重置”指令执行下列操作:  
计时器位 = 关闭, 计时器当前值 = 0

只能用“重置”指令重置TONR计时器。

重置后, TOF计时器要求启用输入从“打开”转换为“关闭”, 以便重新启动。

**1毫秒分辨率**

1毫秒计时器记录自现用1毫秒计时器启用以来1毫秒计时器间隔的数目。执行计时器指令即开始计时; 但是, 1毫秒计时器每毫秒更新一次 (计时器位及计时器当前值), 不与扫描循环同步。换言之, 在超过1毫秒的扫描过程中, 计时器位和计时器当前值将

音胃 隆

计时器指令用于打开和重设计时器，如果是 TONR 计时器，则用于关闭计时器。

因为可在 1 毫秒内的任意时刻启动计时器，预设值必须设为比最小要求计时器间隔大一个时间间隔。例如，使用 1 毫秒计时器时，为了保证时间间隔至少为 56 毫秒，则预设时间值应设为 57。

#### 10 毫秒分辨率

10 毫秒计时器记录自现用 10 毫秒计时器启用以来 10 毫秒计时器间隔的数目。执行计时器指令即开始计时；但是，在每次扫描循环的开始更新 10 毫秒计时器，其方法是以当前值加上积累的 10 毫秒间隔的数目（自前一次扫描开始算起）（换言之，在整个扫描过程中，计时器当前值及计时器位保持不变）

因为可在 10 毫秒内的任意时刻开始计时器，预设值必须设为比最小要求计时器间隔大一个时间间隔。例如，使用 10 毫秒计时器时，为了保证时间间隔至少为 140 毫秒，则预设时间值应设为 15。

#### 100 毫秒分辨率

100 毫秒计时器记录自现用 100 毫秒计时器上一次更新以来 100 毫秒计时器间隔的数目。这种计时器的更新方法是在执行计时器指令时缘鼻爸导由匣 鄣 100 毫秒间隔的数目（自前一次扫描开始算起）。

只有在执行计时器指令时才对 100 毫秒计时器的当前值进行更新。因此，如果启用了 100 毫秒计时器但并未对各扫描循环执行计时器指令，则仍不能更新计时器当前值并将丧失时间。同样，如果在单个扫描循环内多次执行 100 毫秒计时器指令，将向计时器的鼻爸刀啮卧龟 100 毫秒间隔数，赢得时间。只有在每次扫描循环仅仅执行一次计时器指令时，才应该使用 100 毫秒计时器。

因为可在 100 毫秒内的任意时刻启动计时器，预设值必须设为比最小要求计时器间隔大一个时间间隔。例如，使用 100 毫秒计时器时，为了保证时间间隔至少为 2100 毫秒，则预设时间值应设为 22。

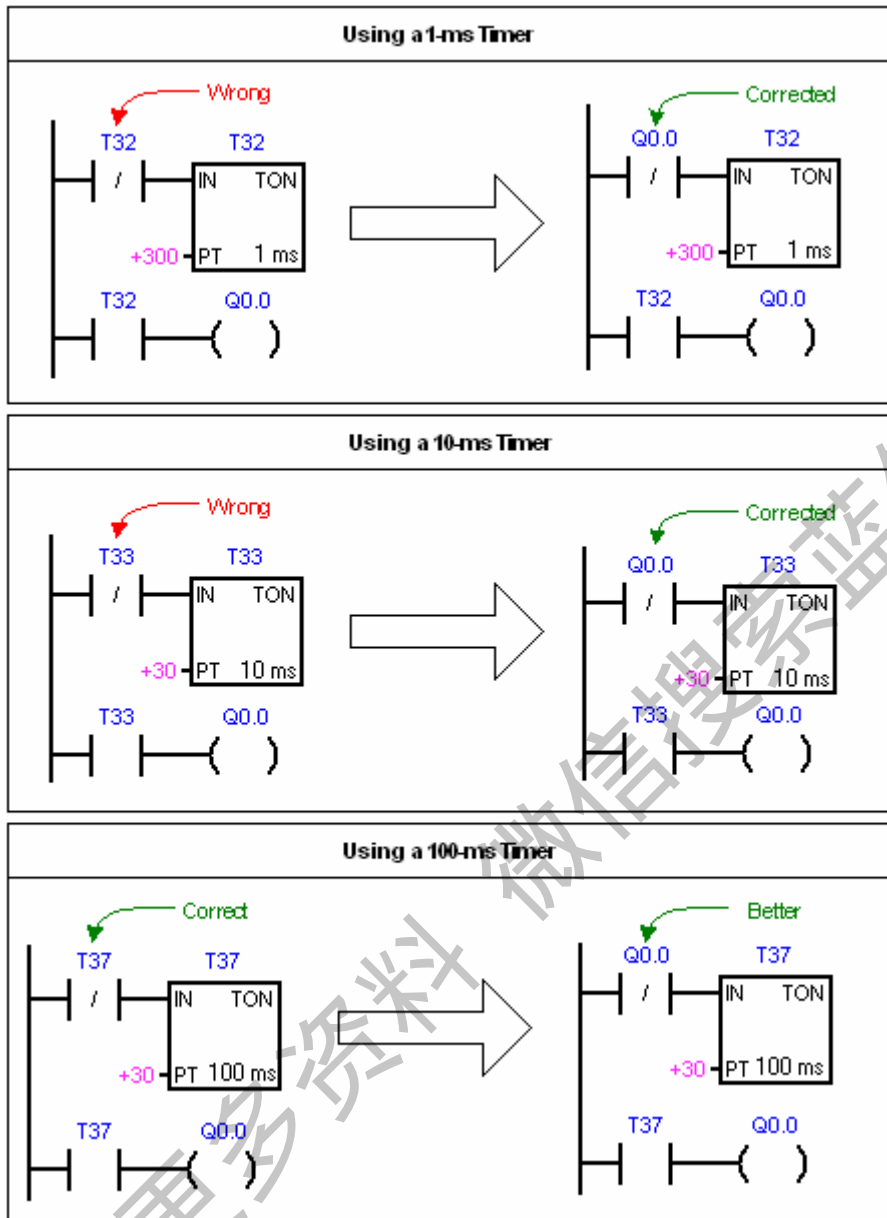
#### 更新计时器当前值

更新当前时间值有多种方式，其作用取决于如何使用计时器，如下图所示的计时器操作：

- 使用 1 毫秒计时器时，每次在执行正常关闭接点 T32 之后和执行正常打开接点 T32 之前更新计时器当前值时，Q0.0 即打开进行一次扫描。
- 使用 10 毫秒计时器时，从不打开 Q0.0，因为从扫描顶端至执行计时器方框均打开计时器位 T33。一旦执行了计时器方框，计时器的当前值及 T 位均被设为零。执行正常打开接点 T33 时，T33 及 Q0.0 均被关闭。
- 使用 100 毫秒计时器时，每当计时器当前值达到预设数值时，Q0.0 时钟打开进行一次扫描。

通过使用正常关闭接点 Q0.0 代替计时器位作为计时器方框的启用输入，可保证每次计时器达到预设值时均打开输出 Q0.0 进行一次扫描。

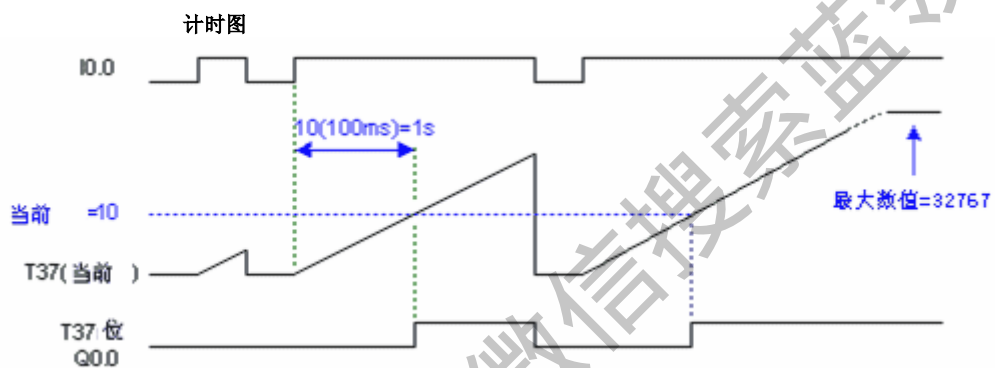
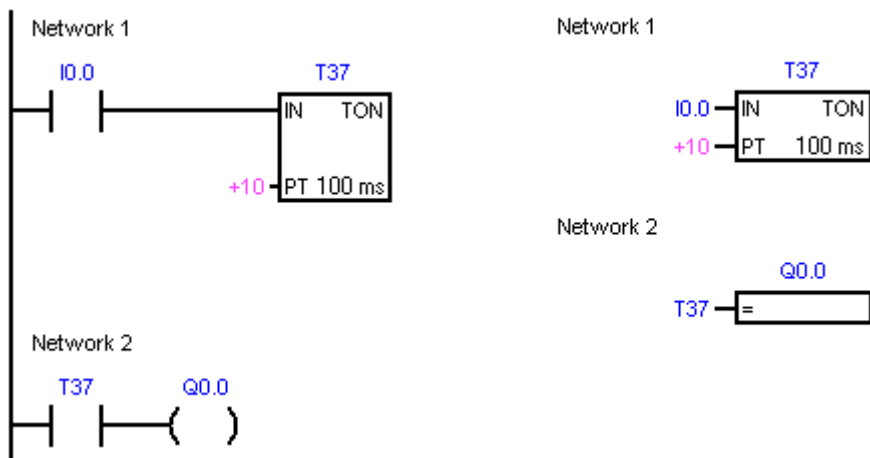
#### 自动重新触发—击计时器举例



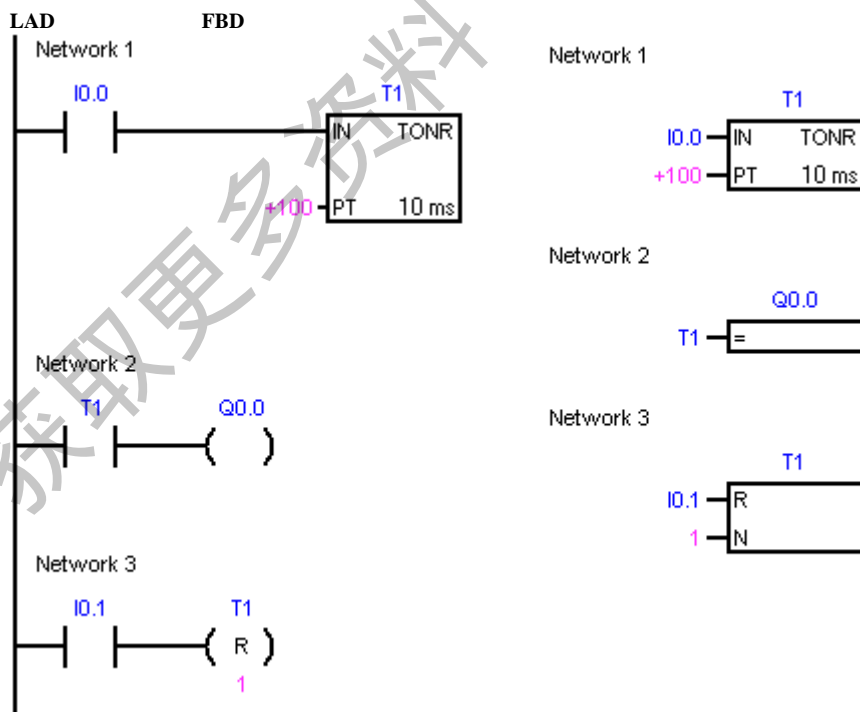
打开延迟计时器举例

LAD

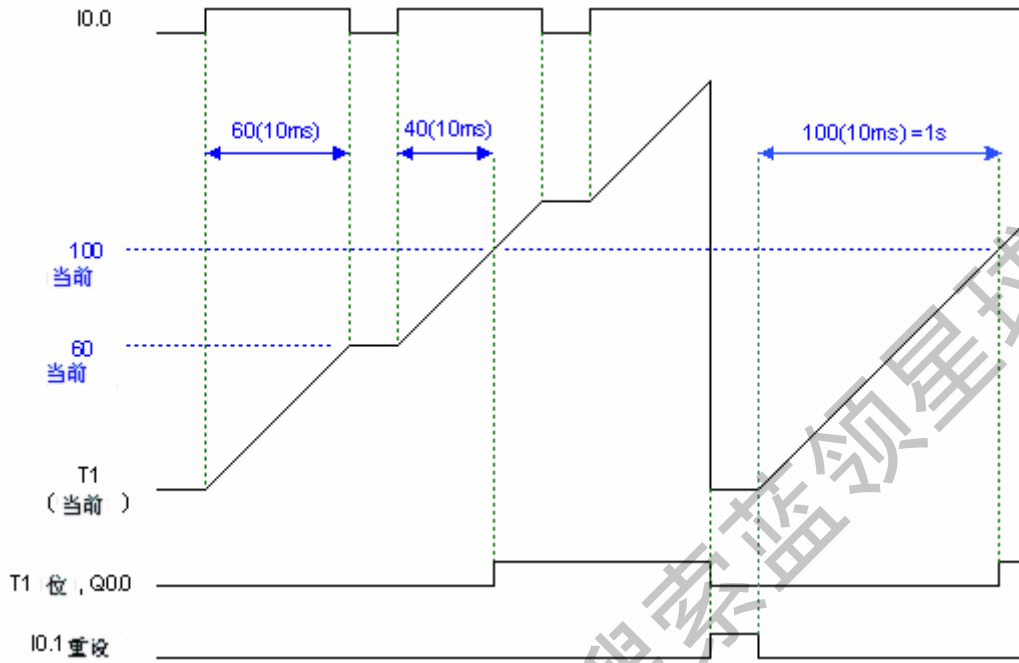
FBD



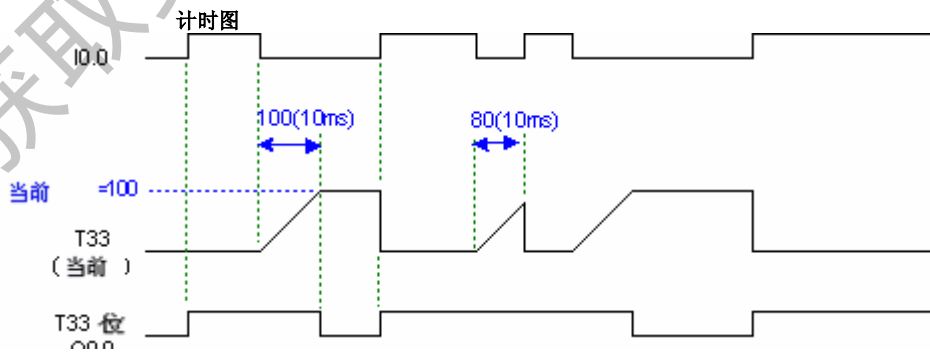
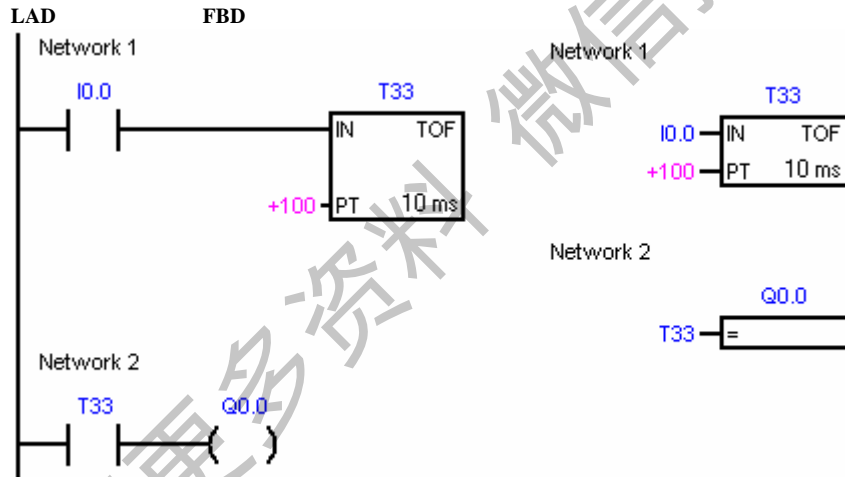
保留性打开延迟计时器举例  
FBD



计时图



关闭延迟计时器举例





### 3.2.19 理解S7-200计数器指令

每次向上计数输入执行从关闭至打开转换时，向上计数（CTU）从该计数器的当前值向上计数。重设输入打开或执行重设指令时，计数器被重设。达到最大值（32,767）时，计数器停止。

每次向上计数输入执行从关闭至打开转换时，向上/向下计数器（CTUD）向上计数，每次向下计数输入执行从关闭至打开转换时，向下计数。达到最大值（32,767）时，向上计数输入的下一个上升边缘导致当前计数变成最小值（32,768）。与此相似，达到最小值（-32,768）时，向下计数输入的下一个上升边缘导致当前计数变成最大值（32,767）。

向上和向上/向下计数器有一个保持当前计数的当前值。计数器还有一个预设值（PV），每次执行计数器指令时，将预设值与当前值进行比较。如果当前值等于预设值，计数器位（C位）打开。否则，C位关闭。

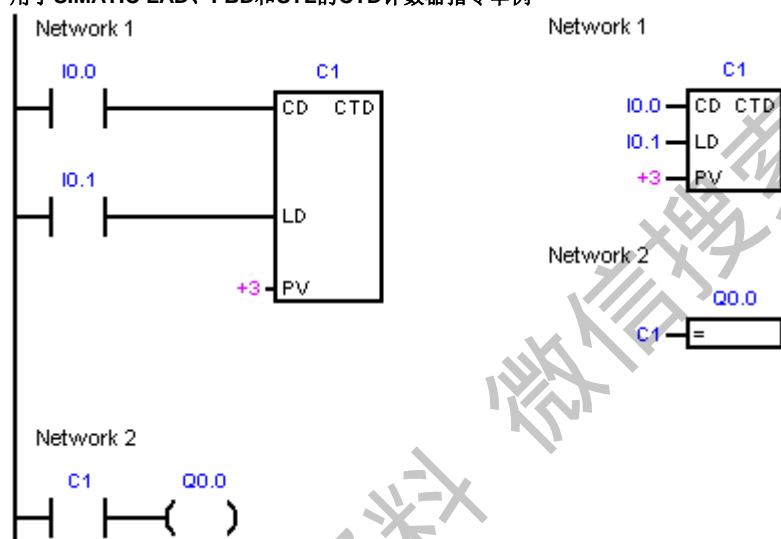
每次向下计数输入执行从关闭至打开转换时，向下计数器（CTD）从该计数器的当前值向下计数。载入输入打开时，计数器重设计数器位，并将预设值载入当前值。达到零时，计数器停止，计数器位（C位）打开。

当您使用重设指令重设计数器时，计数器位被重设，计数器当前值被设为零。使用计数器号码引用该计数器的当前值和C位。

#### 注释：

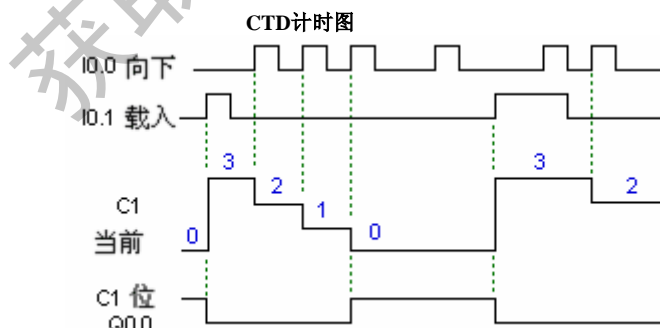
因为每台计数器有一个当前值，请勿将相同的号码指定给一台以上计数器。（向上计数器、向上/向下计数器和向下计数器存取相同的当前值。）

#### 用于SIMATIC LAD、FBD和STL的CTD计数器指令举例

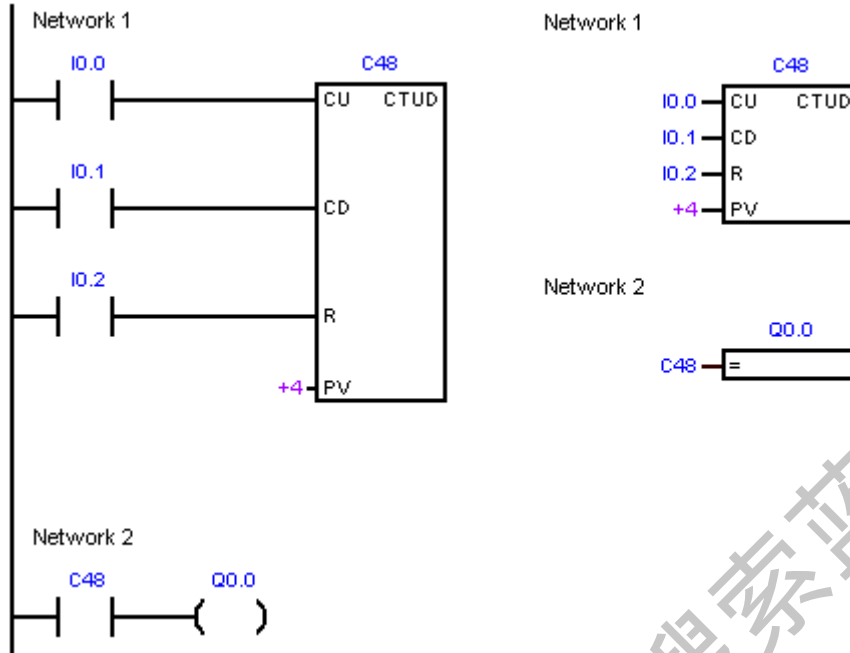


NETWORK 1 // 向下计数器C1当前值从3计数至0，  
// 其中IO.1关闭，IO.0"关闭-打开"递减C1当前值  
// IO.1"打开"载入向下计数预设值3  
LD IO.0  
LD IO.1  
CTD C1 +3

NETWORK 2 // 当计数器C1当前值 = 0时，C1位"打开"  
LD C1  
= Q0.0



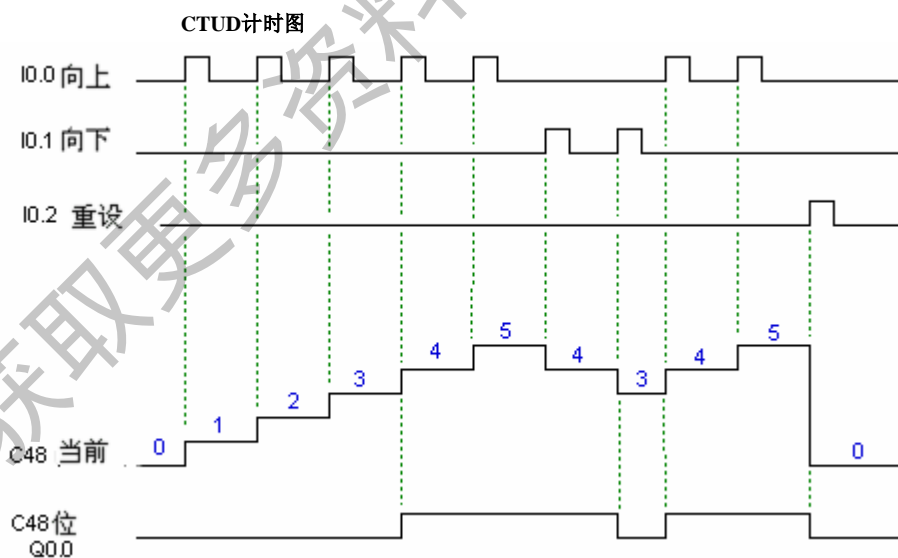
用于SIMATIC LAD、FBD和STL的CTUD计数器指令举例



```

NETWORK 1 // I0.0向上计数 - I0.1向下计数 - I0.2将当前值重设为0
LD I0.0
LD I0.1
LD I0.2
CTUD C48 +4

NETWORK 2 // 当前值 >= 4时, 向上 / 向下计数计数器C48打开C48位
LD C48
= Q0.0
  
```



### 3.2.20 理解高速计数器指令

高速计数器对 CPU 扫描速率无法控制的高速事件进行计数，最多可配置 12 种不同的操作模式。高速计数器的最高计数频率取决于您的 CPU 类型。

每台计数器对支持此类功能的时钟、方向控制、重设和起始均有专用输入。对于二相计数器，两个时钟均可以最高速率运行。在正交模式中，可选择1乘以（1x）或4乘以（4x）最高计数速率。所有计数器均以最高速率运行，互不干扰。

本标题讨论下列主题：

[使用高速计数器](#)  
[理解高速计数器的详细计时功能](#)  
[为高速计数器连接输入线](#)  
[高速计数器编址（HC）](#)  
[理解不同的高速计数器](#)  
[选择现用状态和1x/4x模式](#)  
[高速计数器初始化顺序](#)  
[控制字节](#)  
[HSC模式](#)  
[设置当前值和预设值](#)  
[状态字节](#)  
[为中断赋值](#)

### 使用高速计数器

[返回顶端](#)

通常高速计数器被用作鼓式计数器驱动器，以恒速旋转的转轴配有递增转轴编码器。转轴编码器提供每次旋转的指定计数以及每次旋转一个重设脉冲。转轴编码器的时钟和重设脉冲为高速计数器提供输入。

用最先的几个预设值载入高速计数器，并在当前计数小于当前预设值的期间内激活所需输出。当前计数等于预设值或重设时，计数器设置提供中断。

每次发生当前计数值等于预设值中断事件时，载入新预设值，并设置下一个输出状态。发生重设中断事件时，设置第一个预设值和第一个输出状态，并重复该循环。

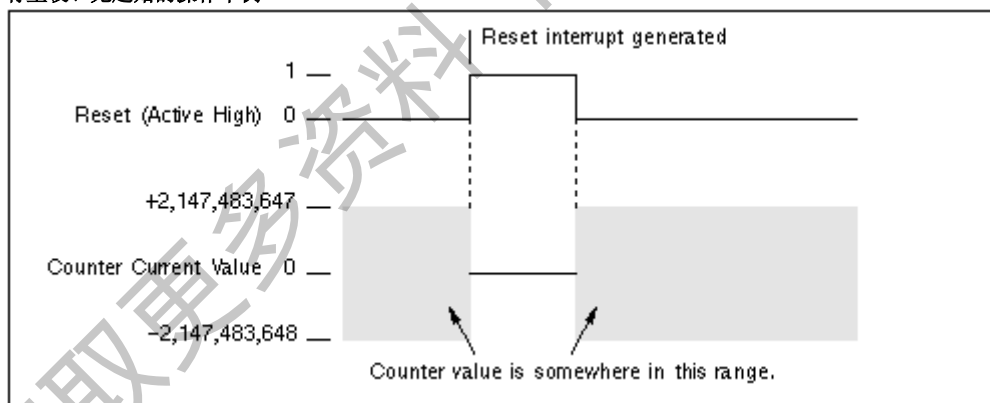
因为中断的发生速率远远低于高速计数器的计数速率，可对高速操作执行精确的控制，并对整体 PLC 扫描循环产生相对较小的影响。V卸细郊雕椒个市確次懶卸伙 谐绿蚌兄葱忻扛鲈唇氩男略と柚担 员憬 凶刺 卍啤# 碓恢址椒去窃诘 颯卸伙 谐绿蚌写 碓 械闹卸鲜录 #

### 理解高速计数器的详细计时功能

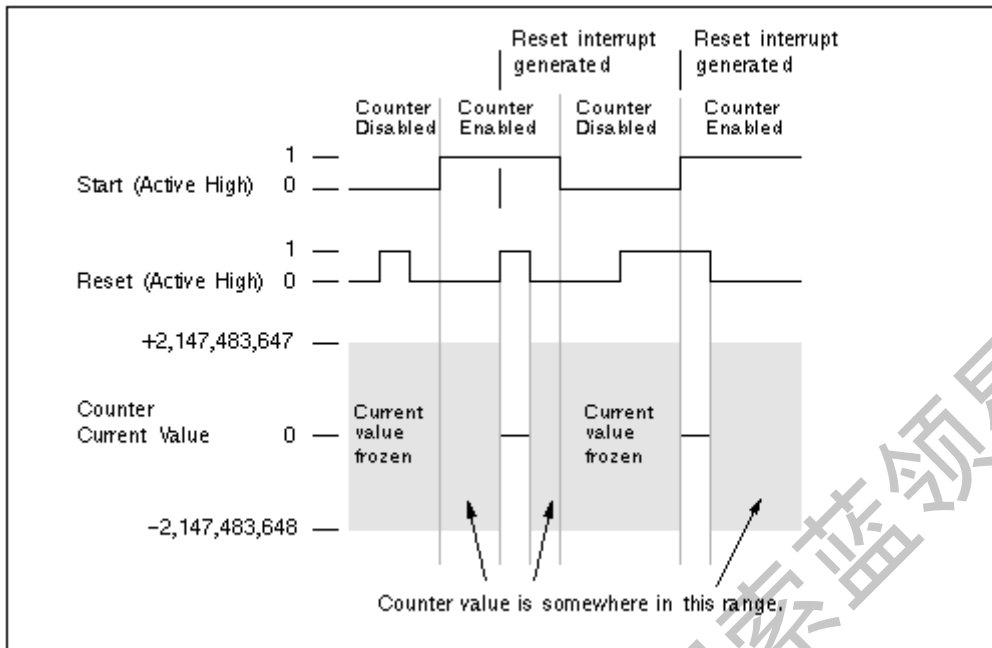
[返回顶端](#)

下列计时图显示根据模式分类的每台计数器的功能。在另一个计时图中显示重设和起始输入操作，并应用于所有使用重设和起始输入的模式。在重设和起始输入图中，重设和起始的现用状态均被编程为高级。

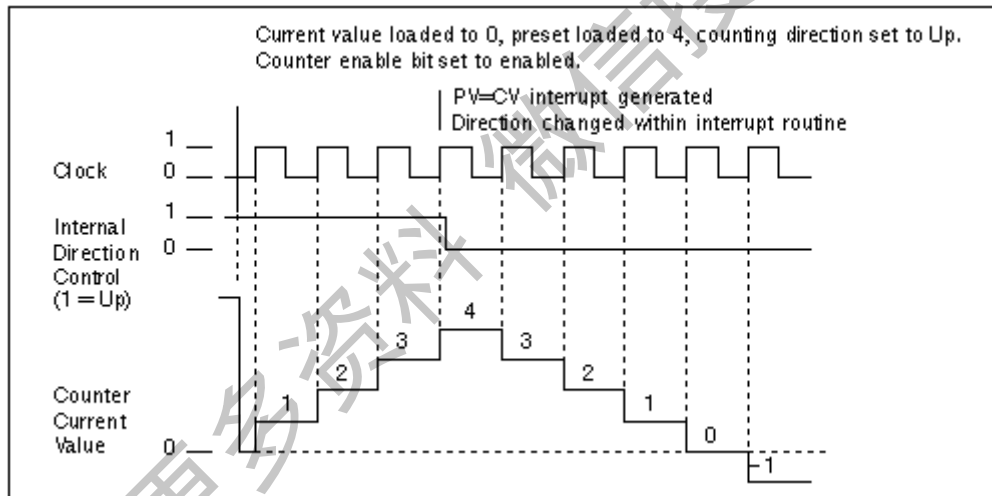
#### 有重设、无起始的操作举例



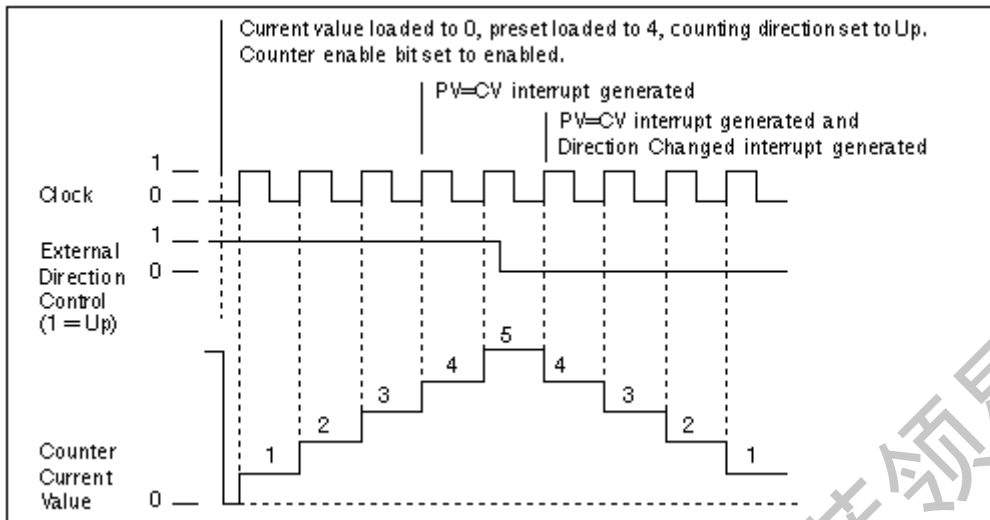
#### 有重设和起始的操作举例



模式0、1和2操作举例

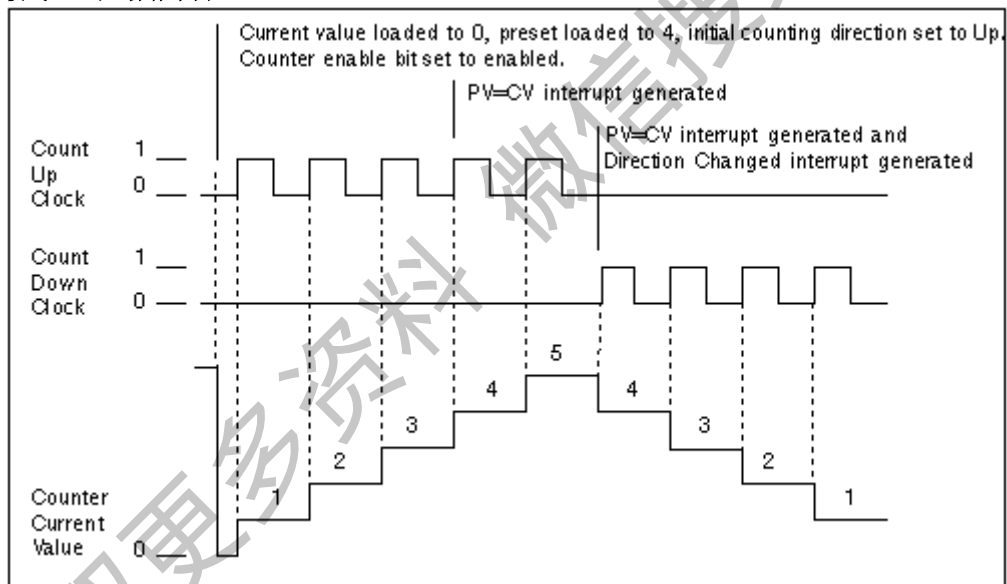


模式3、4和5操作举例

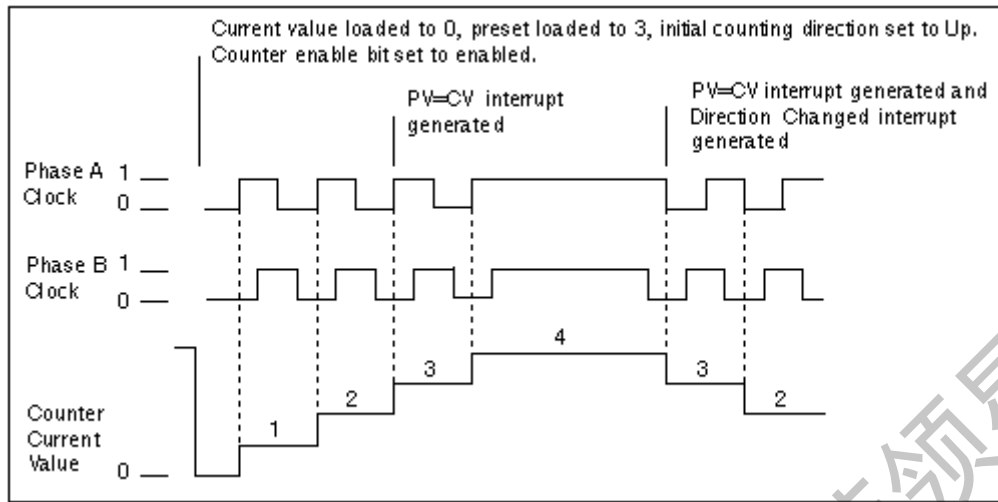


使用计数模式6、7和8时，上下时钟输入的上升边缘间隔0.3微秒，高速计数器可能认为这些事件同时发生。如果发生这种情况，在两种情况下，均不生成错误，而且计数器保持当前计数值。

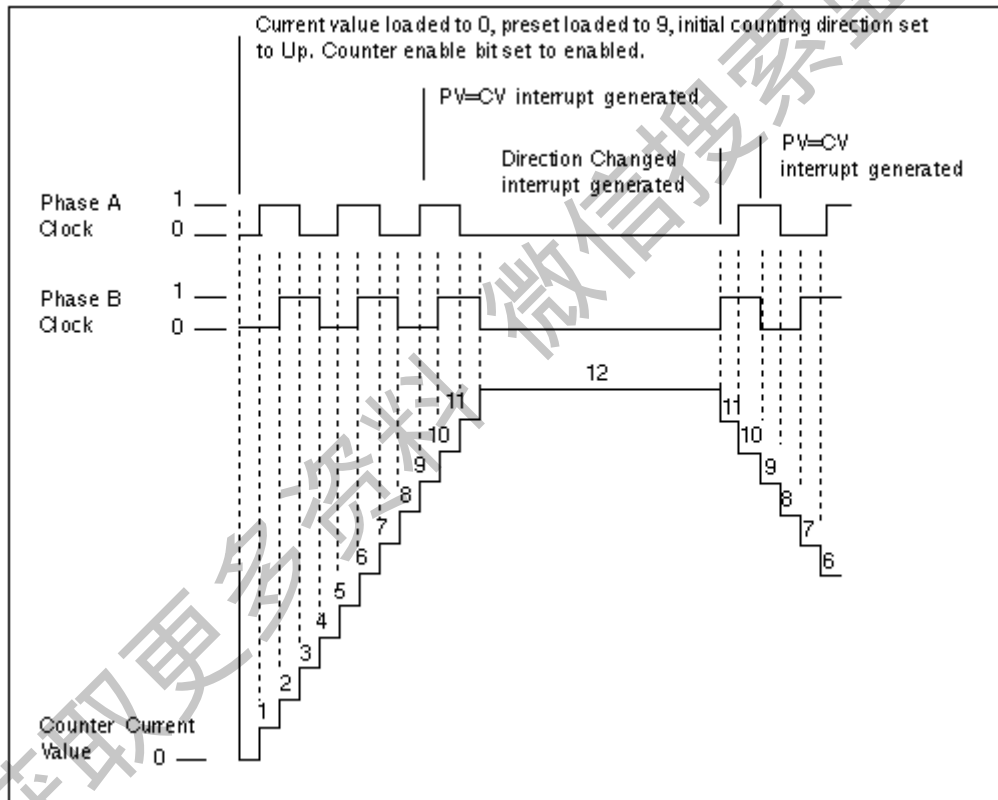
#### 模式6、7和8操作举例



#### 模式9、10和11操作举例（正交 1x 模式）



模式9、10和11操作举例（正交 4x 模式）



为高速计数器连接输入线

[返回顶端](#)

使用“高速计数器定义”指令定义计数器模式和输入。  
下表显示与高速计数器相关的用于时钟、方向控制、重设和起始功能的输入。

高速计数器专用输入

高速计数器	使用的输入
HSC0	I0.0, I0.1, 0.2
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

有些高速计数器和边缘中断的输入点赋值存在某些重叠。同一个输入不能用于两种不同的功能；但是高速计数器当前模式未使用的任何输入均可用于其他目的。例如，如果在模式2中使用HSC0，模式2使用I0.0和I0.2，则I0.1可用于边缘中断或用于HSC3。如果所用的HSC0模式不使用输入I0.1，则该输入可用于HSC3或边缘中断。与此相似，如果所选的HSC0模式不使用I0.2，则该输入可用于边缘中断；如果所选HSC4模式不使用I0.4，则该输入可用于HSC5。

请注意HSC0的所有模式均使用I0.0，HSC4的所有模式均使用I0.3，因此当使用这些计数器时，这些输入点绝不会用于其他用途。

#### HSC模式

[返回顶端](#)

HSC 模式	说明	输入			
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I0.2	I1.1
	HSC2	I1.2	I1.3	I1.1	I1.2
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	具有内部方向控制的单相计数器	时钟			
1		时钟		重设	
2		时钟		重设	开始
3	具有外部方向控制的单相计数器	时钟	方向		
4		时钟	方向	重设	
5		时钟	方向	重设	开始
6	具有两个时钟输入的两相计数器	正计时	倒计时		
7		正计时	倒计时	重设	
8		正计时	倒计时	重设	开始
9	A / B 相正交计数器	时钟 A	时钟 B		
10		时钟 A	时钟 B	重设	
11		时钟 A	时钟 B	重设	开始
12	仅HSC0和HSC3支持12模式。 HSC0计数Q0.0所发脉冲的数目。 HSC3计数Q0.1所发脉冲的数目。				

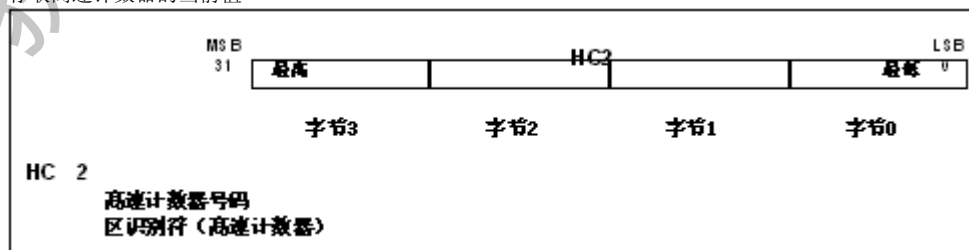
#### 为高速计数器编址 (HC)

[返回顶端](#)

欲存取高速计数器的计数值，您需要利用内存类型 (HC) 和计数器号码 (例如HC0) 指定高速计数器的地址。如下所示，高速计数器的当前值是只读数值，只能作为双字 (32位) 编址。

格式：HC[高速计数器号码]，以HC2为例。

存取高速计数器的当前值



理解不同的高速计数器

### 返回顶端

对于相同的计数器操作模式，所有计数器的功能均相同。如上所示，共有四种基本计数器模式类型。请注意并非每一种计数器均支持每种模式。您可以使用每种类型：无重设或起始输入、有重设但无起始或有起始和重设输入。

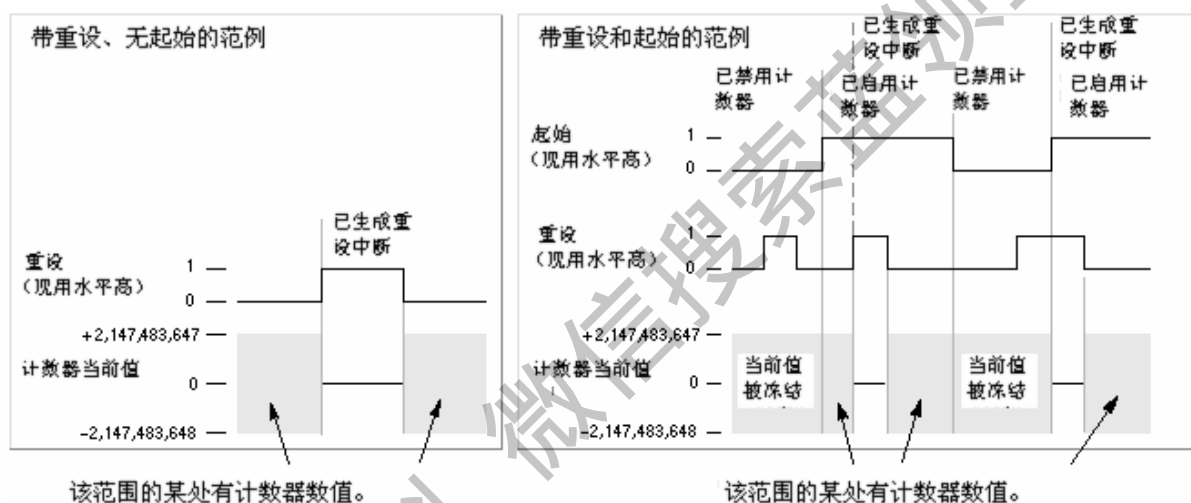
- 激活重设输入时，会清除当前值并保持清除状态直至取消激活重设。
- 激活起始输入时，会允许计数器计数。起始被取消激活时，计数器的当前值保持恒定，并忽略时钟事件。
- 如果在起始未激活时激活重设，则会忽略重设，当前值不变。如果激活重设输入时激活起始输入，则当前值被清除。

使用高速计数器之前必须选择计数器模式，您可以利用HDEF指令（高速计数器定义）选择计数器模式。利用首次扫描内存位SM0.1（首次扫描时该位打开，随后关闭），调用包含HDEF指令的子例行程序。

### 选择激活状态和1x/4x模式

#### 返回顶端

上图显示的重设和起始输入操作适用于使用重设和起始输入的所有模式。在重设和起始输入图形中，重设和起始均显示为现用状态编程为高位。



四台计数器有三个控制位，用于配置重设和起始输入的激活状态并选择1x或4x计数模式（仅限正交计数器）。这些控制位位于各自计数器的控制字节内，只在执行HDEF指令时才使用。

执行HDEF指令之前，必须将这些控制位设为所需的状态，否则计数器采用所选计数器模式的默认配置。重设输入和起始输入的默认配置为：重设（或4乘以输入时钟频率）。一旦执行了HDEF指令，就不能再改变计数器设置，除非CPU设为STOP（停止）模式。

### HDEF控制位

（仅在执行HDEF时使用）

HSC0	HSC1	HSC2	HSC4	说明
SM37.0	SM47.0	SM57.0	SM147.0	“重设”现用水平控制位**：0 =
	SM47.1	SM57.1		“起始”现用水平控制位**：0 =
SM37.2	SM47.2	SM57.2	SM147.2	“正交”计数器的计数速率选项：0 =
= 4x计数速率 1 = 1x计数速率				
**重设输入和起始输入的默认值为现用水平高，求积计数率为4x（或4乘以输入时钟频率）。				

### 控制字节

#### 返回顶端

一旦定义了计数器和计数器模式，您就可以为计数器动态参数编程。每台高速计数器均有一个控制字节，允许完成以下作业：

- 启用或禁用计数器
- 控制方向（仅限模式0、1和2）或初始化所有其他模式的计数方向
- 载入当前值

通过执行HSC指令可激活控制字节以及相关当前值和预设值检查。下表说明每个控制位。

### 用于HSC参数的SM控制位



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

### 设置当前值和预设值

[返回顶端](#)

每台高速计数器都有一个32位当前值和一个32位预设值，当前值和预设值均为带符号的整数。欲向高速计数器载入新的当前值和预设值，您必须设置包含当前值和 / 或预设值的控制字节及特殊内存字节。然后您必须执行 HSC指令，将新数值传输至高速计数器。下表说明用于包含新当前值和预设值的特殊内存字节。除控制字节以及新预设值和当前值保持字节外，还可以使用数据类型 HC（高速计数器当前值）加计数器号码（0、1、2、3、4或5）读取每台高速计数器的当前值。因此，读取操作可直接存取当前值，但只有用上述 HSC指令才能执行写入操作。

载入数值	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
新当前值	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
58						
新预设值	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162
62						

### 状态字节

[返回顶端](#)

为每台提供状态内存位的高速计数器提供状态字节，状态内存位表示当前计数方向以及当前值是否大于或等于预设值。下表定义每台高速计数器的状态位。

#### HSC0、HSC1、HSC2、HSC3、HSC4和HSC5的状态位

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

#### 注释:

只有在执行高速计数器中断例行程序时，状态位才有效。监控高速计数器状态的目的在于启用对正在执行的操作有重大影响的事件的中断程序。

### 为中断赋值

[返回顶端](#)

所有计数器模式均支持当前值等于预设值中断，使用外部重设输入的计数器模式支持将外部重设现用中断。除模式 0、1和2以外，所有模式均支持外部重设现用中断。除模式 0、1和2以外，所有模式均支持外部重设现用中断。

**注释：**

如果您尝试载入新的当前值或从外部重设中断例行程序内先禁用然后再重新启用高速计数器，会引起严重错误。

**高速计数器初始化顺序**

[返回顶端](#)

HSC1在以下初始化和操作顺序说明中被用作模型计数器。初始化说明假设 S7-200刚刚被放置在RUN（运行）模式中，因此首次扫描内存位为真。如果不是如此，请记住在进入RUN（运行）模式后，只能为刻啮倍剖 髦葱幸淮 HDEF指令。为高速计数器第二次执行HDEF会生成运行时间错误，并不会改变该计数器首次执行HDEF时计数器的设置方式。虽然以下顺序分别显示如何更改方向、当前值和预设值，您可以按照相同的顺序更改所有这些数值或这些数值的任何组合，方法是以适当的方式设置SMB47数值，然后执行HSC指令。

**0、1或2初始化模式**

下列步骤说明如何为带内部方向的单相向上 / 向下计数器（模式 0、1或2）初始化HSC1：

- 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序  
蛇丝杉跽偕 枋奔渲葱胁(6)钩缘蚪峇垢 趵砣
- 在初始化子例行程序中，根据所需的控制操作载入SMB47。例如：  
SMB47 = 16#F8产生下列结果：  
启动计数器  
写入新当前值  
写入新预设值  
将方向设置为向上计数  
将起始和重设输入设为现用水平高
- 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）输入设为0，有外部重设但无起始设为1，有外部重设和起始设为2。
- 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
- 用所需的预设值载入SMD52（双字尺寸数值）。
- 为了捕获当前值等于预设值，将CV=P中断事件（事件13）附加于中断例行程序中，为中断编程。
- 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
- 执行全局中断启用指令（ENI），启用中断。
- 执行HSC指令，使S7-200为HSC1编程。
- 退出子例行程序。

**3、4或5初始化模式**

下列步骤说明如何为带外部方向的单相向上 / 向下计数器（模式 3、4或5）初始化HSC1：

- 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序  
蛇丝杉跽偕 枋奔渲葱胁(6)钩缘蚪峇垢 趵砣
- 在初始化子例行程序中，根据所需的控制操作载入SMB47。例如：  
SMB47 = 16#F8产生下列结果：  
启用计数器  
写入新当前值  
写入新预设值  
设置HSC初始方向，向上计数  
将起始和重设输入设为现用水平高
- 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）输入设为3，有外部重设但无起始设为4，有外部重设和起始设为5。
- 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
- 用所需的预设值载入SMD52（双字尺寸数值）。
- 为了捕获当前值等于预设值，将CV=P中断事件（事件13）附加于中断例行程序中，为中断编程。
- 为了捕获方向改变，将方向改变中断事件（事件14）附加于中断例行程序中，为中断编程。
- 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
- 执行全局中断启用指令（ENI），启用中断。
- 执行HSC指令，使S7-200为HSC1编程。
- 退出子例行程序。

**6、7或8初始化模式**

下列步骤说明如何为带向上 / 向下时钟的双相向上 / 向下计数器（模式 6、7或8）初始化HSC1：

- 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序  
蛇丝杉跽偕 枋奔渲葱胁(6)钩缘蚪峇垢 趵砣
- 在初始化子例行程序中，根据所需的控制操作载入SMB47。例如：  
SMB47 = 16#F8产生下列结果：  
启用计数器

写入新当前值  
 写入新预设值  
 设置HSC初始方向，向上计数  
 将起始和重设输入设为现用水平高

3. 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）设为6，有外部重设但无起始设为7，有外部重设和起始设为8。
4. 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD52（双字尺寸数值）。
6. 为了捕获当前值等于预设值，将CV=P中断事件（事件13）附加于中断例行程序中，为中断编程。
7. 为了捕获方向改变，将方向改变中断事件（事件14）附加于中断例行程序中，为中断编程。
8. 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
9. 执行全局中断启用指令（ENI），启用中断。
10. 执行HSC指令，使S7-200为HSC1编程。
11. 退出子例行程序。

### 9、10或11初始化模式

下列步骤说明如何为A/B相正交计数器（模式9、10或11）初始化HSC1：

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序
2. 在初始化子例行程序中，根据所需的控制操作载入SMB47。

例如（1x计数模式）：

SMB47 = 16#FC产生下列结果：

启用计数器  
 写入新当前值  
 写入新预设值  
 设置HSC初始方向，向上计数  
 将起始和重设输入设为现用水平高

例如（4x计数模式）：

SMB47 = 16#F8产生下列结果：

启用计数器  
 写入新当前值  
 写入新预设值  
 设置初始HSC方向，向上计数  
 将起始和重设输入设为现用水平高

3. 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）输入设为9，有外部重设但无起始设为10，有外部重设和起始设为11。
4. 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD52（双字尺寸数值）。
6. 为了捕获当前值等于预设值，将CV=P中断事件（事件13）附加于中断例行程序中，为中断编程。
7. 为了捕获方向改变，将方向改变中断事件（事件14）附加于中断例行程序中，为中断编程。
8. 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
9. 执行全局中断启用指令（ENI），启用中断。
10. 执行HSC指令，使S7-200为HSC1编程。
11. 退出子例行程序。

### 12初始化模式

下面步骤描述了如何为计算由PTO0产生的脉冲数而初始化HSC0（模式12）。

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序
2. 在初始化子例行程序中，根据所需的控制操作载入SMB47。例如：

SMB37 = 16#F8 产生下列结果：

启动计数器  
 写入新当前值  
 写入新预设值  
 将方向设置为向上计数  
 将起始和重设输入设为现用水平高

3. 在将HSC输入设为0且MODE（模式）输入设为12后执行HDEF指令。
4. 用所需的当前值载入SMD38（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD42（双字尺寸数值）。
6. 为了捕获当前值等于预设值事件，将CV = PV中断事件（事件13）附加于中断例行程序中，从而实现中断编程。如需关于中断处理的全部细节，请参阅讨论“中断指令”的章节。

7. 执行全局中断启用指令（ENI）以启用中断。
8. 执行HSC指令以使S7-200为HSC0编程。
9. 退出子例行程序。

#### 在0、1、2或12模式中改变方向

下列步骤说明如何配置HSC1，使带内部方向的单相计数器（模式0、1、2或12）改变方向：

1. 载入SMB47，写入所需的方向：  
SMB47 = 16#90  
启动计数器  
将HSC方向设为向下计数  
  
SMB47 = 16#98  
启动计数器  
将HSC方向设为向上计数
2. 执行HSC指令，使S7-200为HSC1编程。

#### 载入新当前值（任何模式）

下列步骤说明如何改变HSC1的计数器当前值（任何模式）：  
改变当前值强迫计数器在进行改动的过程中被禁用。计数器被禁用时，则不再计数或生成中断。

1. 载入SMB47，写入所需的当前值：  
SMB47 = 16#C0  
启用计数器  
写入新当前值
2. 用所需的当前值载入SMD48（双字尺寸）（载入零可加以清除）。
3. 执行HSC指令，使S7-200为HSC1编程。

#### 载入新预设值（任何模式）

下列步骤说明如何改变HSC1的计数器预设值（任何模式）：

1. 载入SMB47，写入所需的预设值：  
SMB47 = 16#A0  
启用计数器  
写入新预设值
2. 用所需的预设值载入SMD52（双字尺寸数值）。
3. 执行HSC指令，使S7-200为HSC1编程。

#### 禁用高速计数器（任何模式）

下列步骤说明如何禁用HSC1高速计数器（任何模式）：

1. 载入SMB47，禁用计数器：  
SMB47 = 16#00 禁用计数器
2. 执行HSC指令，禁用计数器。

虽然上述操作步骤显示如何逐一改变方向、当前值和预设值，您也可以按照相同顺序，通过以适当方式设置 SMB47数值并执行HSC指令，改变全部数值或其中任何组合。

另请参阅：

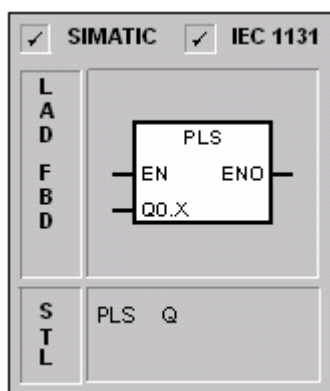
[HSC向导](#)

[HSC指令](#)

[HDEF指令](#)

### 3.2.21 理解高速输出指令

操作数	数据类型	[[[ENDKEEPN]]]	
Q 常量 (0或1)		字	[[[ENDKEEPN]]]
内存范围	错误	S7-200 CPU指令支持	SIMATIC / 国际助记符
[[[ENDKEEPN]]]			
数据范围	ENO	CPU内存中的指令大小	编址内存 [[[ENDKEEPN]]]



**脉冲输出 (PLS)** 指令被用于控制在高速输入 (Q0.0和Q0.1) 中提供的“脉冲链输出” (PTO) 和“脉冲宽度调制” (PWM) 输出。PTO提供矩形波 (工作循环) 输出, 配备循环时间和脉冲数用户控制功能。

PWM提供连续性变量工作循环输出, 配备循环时间和脉冲宽度用户控制功能。脉冲输出范围 Q0.0至Q0.1

[特殊内存PTO / PWM高速输出寄存器](#)

#### 理解S7-200高速输出指令

S7-

200有两台PTO/PWM生成器, 建立高速脉冲链或脉冲宽度调节波形。一台生成器指定给数字输出点 Q0.0, 另一台生成器指定给数字输出点Q0.1。一个指定的特殊内存 (SM) 位置为每台生成器存储以下数据: 一个控制字节 (8位数值)、一个脉冲计数值 (一个带符号的32位数值) 和一个循环时间和脉冲宽度数值 (一个不带符号的16位数值)。

PTO/PWM生成器和进程图像寄存器共用Q0.0和Q0.1。PTO或PWM功能在Q0.0或Q0.1位置现用时, PTO/PWM生成器控制输出, 并禁止输出点的正常使用。输出波形不受进程图像寄存器状态、点强迫数值、执行立即输出指令的影响。PTO/PWM生成器非现用

保持住! 谱 柜 掏枷窠姆嫫螯 = 掏枷窠姆嫫猓詔去涑霾 4 蔚某脚己妥航兆刺 共 4 卧诟啾换虹臀膝 己徒嘎

#### 注释:

- 在启用PTO或PWM操作之前, 将用于Q0.0和Q0.1的进程图像寄存器设为0。
- 所有的控制位、循环时间、脉冲宽度和脉冲计数值的默认值均为0。
- PTO/PWM输出必须至少有10%的额定负载, 才能完成从关闭至打开以及从打开至关闭的顺利转换。
- 文档光盘“提示与技巧”中的提示7、22、23、30和50包含使用PTO/PWM操作PLS指令的程序。

脉冲链 (PTO) 功能提供矩形波 (50%工作循环) 输出或指定的脉冲数和指定的循环时间。脉冲宽度调制 (PWM) 功能提供带变量工作循环的固定循环时间输出。

每台PTO/PWM生成器有一个控制字节 (8位), 一个循环时间数值和脉冲宽度数值 (不带符号的16位数值) 和一个脉冲计数值 (不带符号的32位数值)。这些数值全部存储在特殊内存 (SM) 区域的指定位置。一旦设置这些特殊内存位的位置, 选择所需的值, 葱新此途涑啮啻睿 PLS) 即启动操作。该指令使S7-200读取SM位置, 并为PTO/PWM生成器编程。

通过修改SM区域中 (包括控制字节) 要求的位置, 您可以更改 PTO或PWM的波形特征, 然后执行PLS指令。

您可以在任意时间向控制字节 (SM67.7或SM77.7) 的PTO/PWM启用位写入零, 禁用PTO或PWM波形的生成, 然后执行PLS指令

**注释:** 所有控制位、循环时间、脉冲宽度和脉冲计数值的默认值均为零。

**注释:** PTO/PWM输出必须至少有10%的额定负载, 才能完成从关闭至打开以及从打开至关闭的顺利转换。

#### PWM操作

PWM功能提供带变量工作循环的固定循环时间输出。可以微秒或毫秒为时基指定循环时间和脉冲宽度。

循环时间的范围从10微秒至65,535微秒, 或从2毫秒至65,535毫秒。

脉冲宽度时间范围从0微秒至65,535微秒或从0毫秒至65,535毫秒。



如上表所示, 设置脉冲宽度等于循环时间 (这使工作循环为 100%) 使输出连续运行。设置脉冲宽度等于0 (这使工作循环为0%) 会关闭输出。

#### 脉冲宽度时间 / 循环时间

#### 反应

脉冲宽度时间  $\geq$  循环时间数值 工作循环为100%: 输出连续运行。

脉冲宽度时间 = 0 工作循环为0%: 输出关闭。

循环时间 < 2个时间单位 循环时间的默认值为两个时间单位。

有两种不同的方法可改变PWM波形的特征: 同步更新和异步更新。

- 同步更新: 如果不要求更改时基, 即可以执行同步更新。执行同步更新时, 波形特征的变化发生在循环边缘, 提供顺

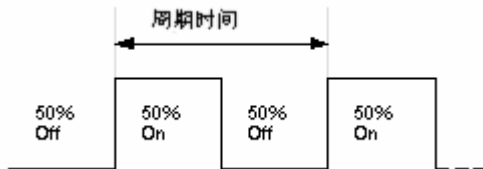
弧

- 异步更新：此为常见的PWM操作，脉冲宽度不同，但循环时间保持不变。因此，不要求时基改变。但是，如果要求改变PTO/PWM生成器的时基，则应使用异步更新。异步更新使PTO/PWM生成器被立即禁用，与PWM波形异步。这样可能造成控制设备状态暂时不稳。由于此一原因，建议使用同步PWM更新。选择可用于所有预计循环时间数值的时基。

控制字节中的PWM更新方法位（SM67.4或SM77.4）指定更新类型，在执行PLC指令时激活改动。请注意，如果时基改变，则会

#### PTO操作

PTO为指定的脉冲数和指定的循环时间提供矩形波（50%工作循环）输出。PTO可提供单脉冲链或多脉冲链（使用脉冲轮廓）。



循环时间范围从10微秒至65,535微秒或从2毫秒至65,535毫秒。

脉冲计数范围从1至4,294,967,295次脉冲。

为循环时间指定基数微秒或毫秒（例如75毫秒）会引起工作循环的失真。

下表定义脉冲计数和循环时间限制：

#### 脉冲计数 / 循环时间

#### 反应

循环时间 < 2个时间单位

循环时间的默认值为2个时间单位。

脉冲计数 = 0

脉冲计数的默认值为1次脉冲。

状态字节（SM66.7或SM76.7）中的PTO空闲位表示编程脉冲链已完成。另外，也可在脉冲链完成时激活中断例行程序。如果您使用多段操作，则在轮廓表完成时立即激活中断例行程序。请参阅以下多段管线连接。

PTO功能允许脉冲链链接或管线作业。现用脉冲链完成时，新的脉冲链输出立即开始。这样就保证了随后的输出脉冲链的连续性。

该管线作业可以两种方式中的一种完成：单段管线作业或多段管线作业。

#### 单段管线作业

在单段管线作业中，您负责更新下一个脉冲链的SM位置。初始PTO段一旦开始，您必须按照对第二个波形的要求立即修改SM位。第二个脉冲链特征被保留在管线中，直至第一个脉冲链完成。管线中每次只能存储一个条目。第一个脉冲链一旦完成，第二个波形输出即开始，管线可用于新的脉冲链规格。您可以重复此步骤，设置下一个脉冲链的特征。

可在脉冲链之间平稳转换，下列情况除外：

- 如果执行时基改动
- 如果现用脉冲链在执行PLS指令捕获到新脉冲链设置之前完成

如果您在管线已满时尝试载入，状态寄存器（SM66.6或SM76.6）中的PTO溢出位被设置。进入RUN（运行）模式时，该位被初始化为0。如果您希望探测随后出现的溢出，则必须在探测到溢出之后以手动方式清除该位。

#### 多段管线作业

在多段管线作业中，S7-200从V内存中的轮廓表自动读取每个脉冲链段的特征。该模式中的SM位置是轮廓表的控制字节、状态字节和起始V内存偏移量（SMW

168或SMW178）。时基可以为微秒或毫秒，但该选项适用于轮廓表中的所有循环时间数值，但在轮廓运行时不得变更。然后可由执行PLS指令开始多段操作。

每段输入的长度均为8个字节，由一个16位循环时间数值、一个16位循环时间Δ数值和一个32位脉冲计数数值组成。

下表说明轮廓表的格式。多段PTO操作的另一个特征是能够通过指定每个脉冲的数量自动增加或减少循环时间。在循环时间Δ域

如果您指定的循环时间Δ数值在一定数量的脉冲后导致非法循环时间，则会出现数学溢出条件。PTO功能被终止，输出转换成图

如果您以手动方式异常中止正在运行的PTO轮廓，状态字节（SM66.5或SM76.5）中的用户异常中止位则被设为一。

运行PTO轮廓时，SMB166（或SMB176）中提供当前现用段数。

运行PTO轮廓时，SMB166（或SMB176）中提供当前现用段数。

#### 多段PTO操作的轮廓表格式

距离轮廓表起始位置的 字节偏移量

轮廓段数目

表格条目说明

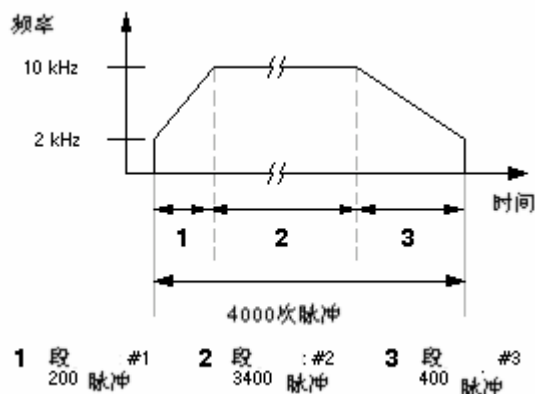
0		段数（1至255）；数值0生成非严重错误，生成无PTO输出
1	#1	初始循环时间（2至65535个时基单位）
3		每次脉冲的循环时间Δ（带符号的数值）（-32768至32767个时基单位）
5		脉冲计数（1至4294967295）
9	#2	初始循环时间（2至65535个时基单位）
11		每个脉冲的循环时间Δ（带符号的数值）（-32768至32767个时基单位）
13		脉冲计数（1至4294967295）
::	#3	::

**计算轮廓表数值**

PTO/PWM生成器的多段管线作业功能在许多应用程序中都很有用，特别是步阶器电机控制中。

例如，您可以通过简单斜坡向上、运行和斜坡向下顺序或更复杂的顺序使用配备脉冲轮廓的 PTO，控制步进器马达，方法是定义恒速啮啮 255个段的脉冲轮廓，每个段与一个斜坡向上、运行或斜坡向下操作相对应。

下图显示生成加速步进器马达（#1段）、按恒速操作马达（#2段）、随后减低马达速度（#3段）的输出波形所要求的轮廓表数值样本。

**简单步阶器电机应用程序的频率与时间图形举例**

在本例中：起始和终止脉冲频率为 2 kHz，最大脉冲频率为 10 kHz

，要求 4000 次脉冲才能达到所需的马达转动次数。因为用阶段（循环时间）表示轮廓表数值，而不使用频率，需要将给定频率数值转换成循环时间数值。因此，起始（最初）和终止（结束）循环时间为 500 ms，与最大频率对应的循环时间为 100 ms。在输出轮廓的加速部分，应在约 400 次脉冲时达到最大脉冲频率。轮廓减速部分应在约 400 次脉冲时完成。

您可以用以下公式确定 PTO/PWM 生成器用于调节某一特定段每次脉冲循环时间的  $\Delta$  循环时间数值：

$$\text{段的 } \Delta \text{ 周期时间} = | \text{End\_CT}_{\text{seg}} - \text{Init\_CT}_{\text{seg}} | / \text{Quantity}_{\text{seg}}$$

位置：  
 $\text{End\_CT}_{\text{seg}}$  = 该段的结束周期时间  
 $\text{Init\_CT}_{\text{seg}}$  = 该段的初始周期时间  
 $\text{Quantity}_{\text{seg}}$  = 该段的脉冲数

利用该公式，计算出的加速部分（或 #1 段）的  $\Delta$  循环时间为 -

2。与此相似，减速部分（或 #3 段）的  $\Delta$  循环时间为 1。因为 #2 段是输出波形的恒速部分，该段的  $\Delta$  循环时间为零。

假定轮廓表位于从 V500 开始的 V 内存中，以下显示用于生成所需波形的表值。您可以在程序中包含指令，将这些数值载入 V 内存。

**轮廓表数值举例**

V 内存地址	数值	说明
VB500	总段数	
VW501	初始循环时间	#1 段
VW503 -2	初始 $\Delta$ 循环时间	
VD505	脉冲数	
VW509	初始循环时间	#2 段
VW511	$\Delta$ 循环时间	
VD513	脉冲数	
VW517	初始循环时间	#3 段
VW519	$\Delta$ 循环时间	
VD521	脉冲数	

利用程序中的指令可将这些表值置于 V 内存中。另一种方法是定义数据块中的轮廓值。本标题结尾部分举例说明使用多段 PTO 操作某绿蝉鸣

为了确定波形段之间的转换是否可接受，您需要确定段中最后一次脉冲的循环时间。除非  $\Delta$  循环时间是 0，您必须计算段最后一温此必难 肥奔洌 蛭 檬 滴丛逐挪 兄付āJ 褂靡韵鹿 郊扑阔罟笠准温此必难 肥奔洌

段的最后一次脉冲周期时间  $= \text{Init\_CT}_{\text{seg}} + (\text{Delta}_{\text{seg}} * (\text{Quantity}_{\text{seg}} - 1))$

位置 :  $\text{Init\_CT}_{\text{seg}} =$  该段的结束周期时间

$\text{Delta}_{\text{seg}} =$  该段的初始周期时间

$\text{Quantity}_{\text{seg}} =$  该段的脉冲数

上例是简化的情况，用于介绍目的，实际应用程序可能要求更复杂的波形轮廓。请记住：

- $\Delta$  循环时间只能指定为整数微秒或毫秒
- 可对每次脉冲执行循环时间修改

这两个项目产生的效果是计算某个特定段的  $\Delta$  循环时间数值可能要求一个循环方案。计算某个特定段的结束循环时间或脉冲数目时可能要求一定的灵活性。

特定轮廓段期限对确定正确的轮廓表数值程序有用。可利用以下公式计算完成特定轮廓段的时间长度：

段时间长度  $= \text{Quantity}_{\text{seg}} * (\text{Init\_CT} + ((\text{Delta}_{\text{seg}}/2) * (\text{Quantity}_{\text{seg}} - 1)))$

位置 :  $\text{Quantity}_{\text{seg}} =$  该段的结束周期时间

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

$\text{Delta}_{\text{seg}} =$  该段的脉冲数

#### PTO/PWM控制寄存器

PLS指令读取存储在指定的SM内存位置的数据，并以此为PTO/PWM生成器编程。SMB67控制PTO 0或PWM 0，SMB77控制PTO 1或PWM 1。PTO/

PWM控制寄存器表描述用于控制PTO/PWM操作的寄存器。您可以将下表用作快速参考，帮助确定放置在PTO/PWM控制寄存器中用于激活所需操作的数值。

您可以改变PTO或PWM波形的特征，方法是修改SM区（包括控制字节）中的位置，然后执行PLS指令。您可以在任何时间禁用PTO或PWM波形的生成，方法是向控制字节（SM67.7或SM77.7）的PTO/PWM启用位写入0，然后执行PLS指令。

状态字节中的PTO空闲位（SM66.7或SM76.7）表示编程脉冲链已完成。此外，可在脉冲链完成时激活中断例行程序。（请参阅中断指令说明和“通讯”指令。）如果您在使用多段操作，在轮廓表完成时激活中断例行程序。

以下条件设置SM66.4（或SM76.4）和SM66.5（或SM76.5）：

- 指定一个在数次脉冲后导致非法循环时间的循环时间  $\Delta$  数值生成一个数学溢出条件，该条件会终止PTO功能，并将“ $\Delta$  计算错误”位（SM66.4或SM76.4）设为1。输出回复为图像寄存器控制。
- 以手动方式异常中止（禁用）正在执行的PTO轮廓会将“用户异常中止”位（SM66.5或SM76.5）设为1。
- 尝试在管线已满的情况下载入会将PTO溢出位（SM66.6或SM76.6）设为1。如果您希望检测随后的溢出，您必须在检测到溢出后以手动方式清除该位。转换至RUN（运行）模式可将该位初始化为0。

**注释：**当您载入新脉冲计数（SMD72或SMD82）、脉冲宽度（SMW70或SMW80）或循环时间（SMW68或SMW78）时，在执行PLS指令之前，还需要在控制寄存器中设置适当的更新位。对于多段脉冲链操作，在执行PLS指令之前，您还必须载入轮廓表的起始偏移量（SMW168或SMW178）和轮廓表数值。

#### PTO/PWM控制寄存器

Q0.0	Q0.1	状态位		
SM66.4		SM76.4	PTO轮廓由于 $\Delta$ 计算错误异常中止	
		0 = 无错；	1 = 异常中止	
SM66.5		SM76.5	PTO轮廓由于用户命令异常中止	
		0 = 无错；	1 = 异常中止	
SM66.6		SM76.6	PTO管线溢出 / 下溢	0 =
无溢出；		1 = 溢出 / 下溢		
SM66.7		SM76.7	PTO空闲	0 =
进行中；		1 = PTO空闲		
Q0.0	Q0.1	控制位		
SM67.0		SM77.0	PTO/PWM更新循环时间值	0 =
无更新；		1 = 更新循环时间		
SM67.1		SM77.1	PWM更新脉冲宽度时间值	0 =
无更新；		1 = 更新脉冲宽度		
SM67.2		SM77.2	PTO更新脉冲计数值	0 =
无更新；		1 = 更新脉冲计数		
SM67.3		SM77.3	PTO/PWM时基选择	0 = 1
祗/tick；		1 = 1ms/tick		
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启用	0 = 禁用PTO/PWM； 1



= 启用PTO/PWM

Q0.0	Q0.1	其他PTO/PWM寄存器	
SMW68 65535)		SMW78	PTO/PWM循环时间值 (范围: 2至
SMW70		SMW80	PWM脉冲宽度值 (范围: 0至65535)
SMD72		SMD82	
PTO脉冲计数值 (范围: 1至4294967295)			
SMB166		SMB176	进行中的段数 (仅用于多段PTO操作)
SMW168		SMW178	
轮廓表起始位置, 用距离V0的字节偏移量表示 (仅用于多段PTO操作)			
SMB170		SMB180	线性轮廓状态字节
SMB171		SMB181	线性轮廓结果寄存器
SMB172		SMB182	手动模式频率寄存器

**PTO/PWM控制字节参考**

控制寄存器 (十六进制数值)	启用 脉冲计数	执行PLS指令的结果 选择模式 脉冲宽度	PTO段操作 循环时间	PWM更新方法
时基 16#81	是	PTO 载入	单段	衽 / 循环
16#84 载入	是	PTO	单段	衽 / 循环
16#85 载入	是	PTO 载入	单段	衽 / 循环
16#89	是	PTO 载入	单段	ms / 循环
16#8C 载入	是	PTO	单段	ms / 循环
16#8D 载入	是	PTO 载入	单段	ms / 循环
16#A0	是	PTO	多段	衽 / 循环
16#A8	是	PTO	多段	ms / 循环
16#D1	是	PWM 载入		同步 衽 / 循环
16#D2	是 载入	PWM		同步 衽 / 循环
16#D3	是 载入	PWM 载入		同步 衽 / 循环
16#D9	是	PWM 载入		同步 ms / 循环
16#DA	是 载入	PWM		同步 ms / 循环
16#DB	是 载入	PWM 载入		同步 ms / 循环

**PTO/PWM初始化和操作顺序**

以下是初始化和操作顺序说明, 能够帮助您更好地理解 PTO和PWM功能操作。在整个顺序说明过程中一直使用脉冲输出 Q0.0。初始化说明假定S7-200刚置入RUN (运行) 模式, 因此首次扫描内存位为真实。如果不是如此或者如果必须对 PTO/PWM功能重新初始化, 您可以利用除首次扫描内存位之外的一个条件调用初始化例行程序。

**PWM初始化**

以下PWM初始化和操作顺序说明建议使用“首次扫描”位 (SM0.1) 初始化脉冲输出。使用“首次扫描”位调用初始化子例行程序可降低扫描时间, 因为随后的扫描无须调用该子例行程序。(仅需在转换为 RUN (运行) 模式后的首次扫描时设置“首次扫描”位。)但是, 您的应用程序可能有其他限制, 要求您初始化 (或重新初始化) 脉冲输出。在此种情况下, 您可以使用另一个条件调用初始化例行程序。

通常, 您用一个子例行程序为脉冲输出初始化PWM。您从主程序调用初始化子例行程序。使用首次扫描内存位 (SM0.1) 将脉冲输出初始化为0, 并调用子例行程序, 执行初始化操作。当您使用子例行程序调用时, 随后的扫描不再调用该子例行程序, 这样

从主程序建立初始化子例行程序调用后, 用以下步骤建立控制逻辑, 用于在初始化子例行程序中配置脉冲输出 Q0.0:

1. 通过将以下一个数值载入SMB67: 16#D3 (选择微秒递增) 或16#DB (选择毫秒递增) 的方法配置控制字节。  
两个数值均可启用PTO/PWM功能、选择PWM操作、设置更新脉冲宽度和循环时间数值、以及选择时基 (微秒或毫秒)。
2. 在SMW68中载入一个循环时间的字尺寸数值。
3. 在SMW70中载入脉冲宽度的字尺寸数值。
4. 执行PLS指令 (以便S7-200为PTO/PWM生成器编程)。
5. 欲为随后的脉冲宽度变化预载一个新控制字节数值 (选项), 在 SMB67: 16#D2 (微秒) 或16#DA (毫秒) 中载入下列数值之一。
6. 退出子例行程序。

### 为PWM输出更改脉冲宽度

如果您用16#D2或16#DA预载SMB67（请参阅以上第5步），您可以使用一个将脉冲宽度改变为脉冲输出（Q0.0）的子例行程序。建立对该子例行程序的调用后，使用以下步骤建立改变脉冲宽度的控制逻辑：

1. 在SMW70中载入新脉冲宽度的字尺寸数值。
2. 执行PLS指令，使S7-200为PTO/PWM生成器编程。
3. 退出子例行程序。

### PTO初始化—单段操作

以下PWM初始化和操作顺序说明建议使用“首次扫描”位（SM0.1）初始化脉冲输出。使用“首次扫描”位调用初始化子例行程序可降低扫描时间，因为随后的扫描无须调用该子例行程序。（仅需在转换为RUN（运行）模式后的首次扫描时设置“首次扫描”位。）但是，您的应用程序可能有其他限制，要求您初始化（或重新初始化）脉冲输出。在此种情况下，您可以使用另一个条件调用初始化例行程序。

通常，您用一个子例行程序为脉冲输出初始化PWM。您从主程序调用初始化子例行程序。使用首次扫描内存位（SM0.1）将脉冲输出初始化为0，并调用子例行程序，执行初始化操作。当您使用子例行程序调用时，随后的扫描不再调用该子例行程序，这样

从主程序建立初始化子例行程序调用后，用以下步骤建立控制逻辑，用于在初始化子例行程序中配置脉冲输出 Q0.0：

1. 通过将以下一个数值载入SMB67: 16#85（选择微秒递增）或16#8D（选择毫秒递增）的方法配置控制字节。
2. 两个数值均可启用PTO/PWM功能、选择PWM操作、设置更新脉冲宽度和循环时间数值、以及选择时基（微秒或毫秒）。

在SMW68中载入一个循环时间的字尺寸数值。

3. 在SMD72中载入脉冲计数的双字尺寸数值。
4. （选项）如果您希望在脉冲链输出完成后立即执行相关功能，您可以将脉冲链完成事件（中断类别 19）附加于中断子谱缘颞 枋奔渲葱校 (8)峁┘峁垢 辖鞅某缘颞 卸媳嗜蹇 褂 ATCH指令并执行全局中断启用指令ENI。
5. 执行PLS指令，使S7-200为PTO/PWM生成器编程。
6. 退出子例行程序。

### 改变PTO循环时间—单段操作

对于单段PTO操作，您可以使用中断例行程序或子例行程序改变循环时间。欲使用单段PTO操作更改中断例行程序或子例行程序机械PTO循环时间，请遵循下列步骤：

- 设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基、设置更新循环时间数值），方法是在SMB67: 16#81（用于微秒）或16#89（用于毫秒）中载入下列一个数值。
- 在SMW68中，载入新循环时间的一个字尺寸数值。
- 执行PLS指令，使S7-200为PTO/PWM生成器编程。更新脉冲计数波形输出开始之前，CPU必须完成所有进行中的PTO。
- 退出中断例行程序或子例行程序。

### 改变PTO脉冲计数—单段操作

对于单段PTO操作，您可以使用中断例行程序或子例行程序改变脉冲计数。欲使用单段PTO操作在中断例行程序或子例行程序中改变PTO脉冲计数，请遵循下列步骤：

1. 设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基、设置更新循环时间数值），方法是在SMB67: 16#84（用于微秒）或16#8C（用于毫秒）中载入以下两个数值之一。
2. 在SMD72中，载入新脉冲计数的一个字尺寸数值。
3. 执行PLS指令（以便S7-200为PTO/PWM生成器编程）。开始用更新脉冲计数生成波形之前，S7-200完成所有进行中的PTO。
4. 退出中断例行程序或子例行程序。

### 改变PTO循环时间和脉冲计数—单段操作

对于单段PTO操作，您可以使用中断例行程序或子例行程序改变循环时间和脉冲计数。欲使用单段PTO操作更改中断例行程序或永谱缘颞械 PTO循环时间和脉冲计数，请遵循下列步骤：

1. 设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基、设置更新循环时间和脉冲计数数值），方法是在SMB67: 16#85（用于微秒）或16#8D（用于毫秒）中载入以下两个数值之一。
2. 在SMW68中，载入新循环时间的一个字尺寸数值。
3. 在SMC72中，载入新脉冲计数的一个字尺寸数值。
4. 执行PLS指令，使S7-200为PTO/PWM生成器编程。用更新脉冲计数和脉冲时间波形输出开始之前，CPU必须完成所有进行中的PTO。
5. 退出中断例行程序或子例行程序。

### PTO初始化—多段操作

通常，您用一个子例行程序为多段操作的脉冲输出配置和初始化PTO。您从主程序调用初始化子例行程序。使用首次扫描内存位（SM0.1）将PTO使用的输出初始化为0，并调用子例行程序，执行初始化操作。当您使用“首次扫描”位调用初始化子例行程序保 婧蟪纳 璨辉俚饕酶米永 谱缘颞 皮 峁垢 枋奔渲葱小

从主程序建立对初始化例行程序的调用后，使用以下步骤建立控制逻辑，用于在初始化子例行程序中配置脉冲输出 Q0.0：使用首次扫描内存位（SM0.1）将输出初始化为0，并调用您所需的子例行程序，执行初始化操作。这样会降低扫描时间执行，并提供结构更严谨的程序。

1. 通过将以下一个数值载入SMB67: 16#A0（选择微秒递增）或16#A8（选择毫秒递增）的方法配置控制字节。

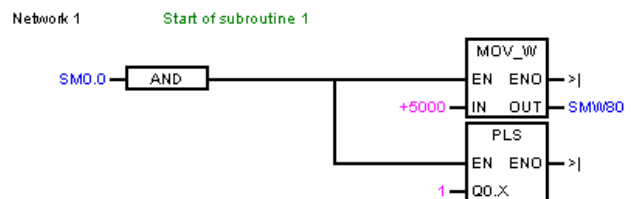
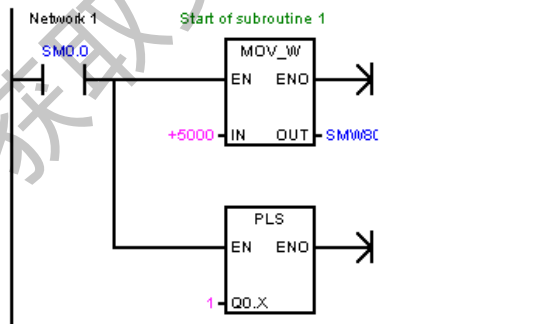
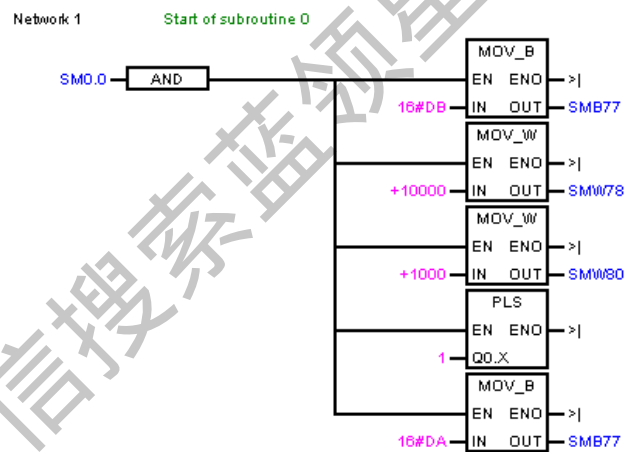
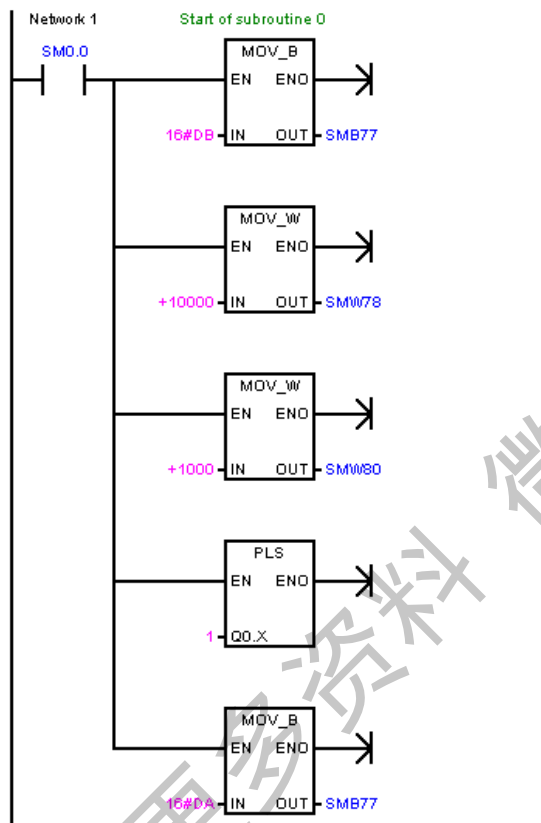
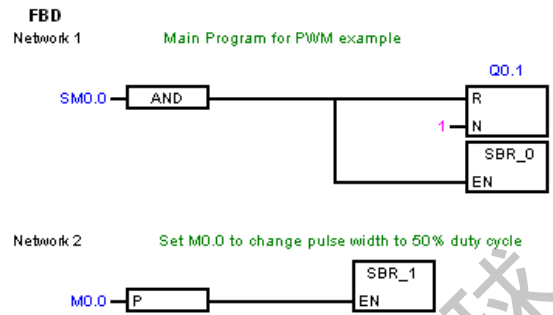
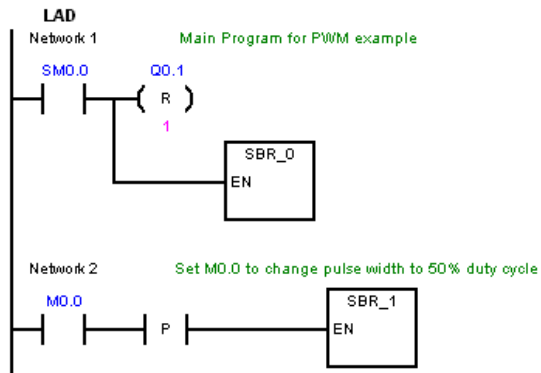
- 两个数值均可启用PTO/PWM功能、选择PTO操作、选择多段操作、以及选择时基（微秒或毫秒）。
2. 在SMW168中载入一个字尺寸数值，用作轮廓表起始V内存偏移量。
  3. 使用V内存存在轮廓表中设置段值。确保“段数”域（表的第一个字节）正确无误。
  4. （选项）如果您希望在PTO轮廓完成后立即执行相关功能，您可以将脉冲链完成事件（中断类别 19）附加在中断子例行程序中，为中断编程。使用ATCH执行全局中断启用指令ENI。
  5. 执行PLS指令，使S7-200为PTO/PWM生成器编程。
  6. 退出子例行程序。

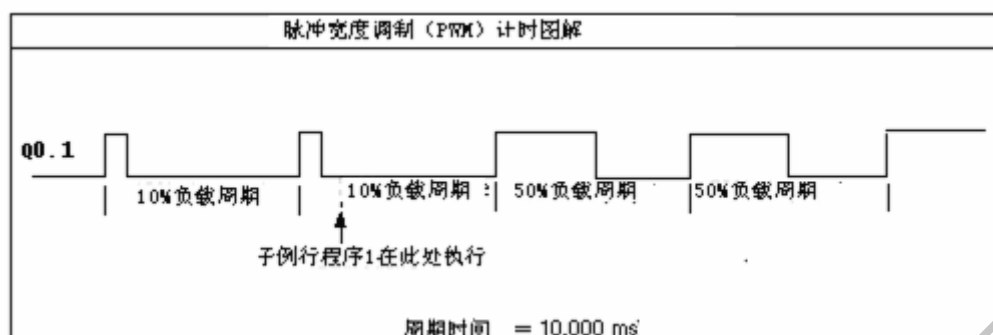
```
NETWORK 1 // 用于PWM范例的主程序
// 首次扫描时，将图像寄存器位设为低，并调用 SBR_0
LD SMO.1
R Q0.1 1
CALL SBR_0
```

```
NETWORK 2 // 设置程序中其他位置的MO.0，将脉冲宽度更改为50%工作循环
LD MO.0
EU
CALL SBR_1
```

```
NETWORK 1 // 子例行程序0开始
LD SMO.0
MOVB 16#DB SMB77 // 设置控制字节
// - 选择PWM操作
// - 选择毫秒递增和同步更新
// - 设置脉冲宽度和循环时间数值
// - 启用PWM功能
MOVW +10000 SMW78 // 将循环时间设为10,000毫秒
MOVW +1000 SMW80 // 将脉冲宽度设为1,000毫秒
PLS 1 // 激活PWM操作：PLS1=>Q0.1
MOVB 16#DA SMB77 // 重新载入控制字节，用于随后的脉冲宽度改动
```

```
NETWORK 1 // 子例行程序1开始
LD SMO.0
MOVW +5000 SMW80 // 将脉冲宽度设为5000毫秒
PLS 1 // 断言脉冲宽度改动
```





## 使用单段操作的脉冲链输出举例

```

NETWORK 1 // 用于单段脉冲链操作的主程序 (PTO)
// 首次扫描时, 将图像寄存器位设为低
// 并调用子例行程序0
LD SMO. 1
R Q0. 0 1
CALL SBR_0

```

```

NETWORK 1 // 子例行程序0开始
LD SMO. 0
MOVB 16#8D SMB67 // 设置控制字节:
// - 选择PTO操作
// - 选择单段操作
// - 选择毫秒递增
// - 设置脉冲计数和循环时间数值
// - 启用PTO功能
MOVW +500 SMD72 // 将循环时间设为500毫秒。
MOVD +4 SMD72 // 将脉冲计数设为4次脉冲。
ATCH INT_0 19 // 将中断例行程序0定义为
// 处理PTO完成中断的中断。
ENI // 全局中断启用
PLS 0 // 激活PTO操作, PLS0 => Q0. 0
MOVB 16#89 SMB67 // 预载控制字节, 用于随后的
// 循环时间改动。

```

```

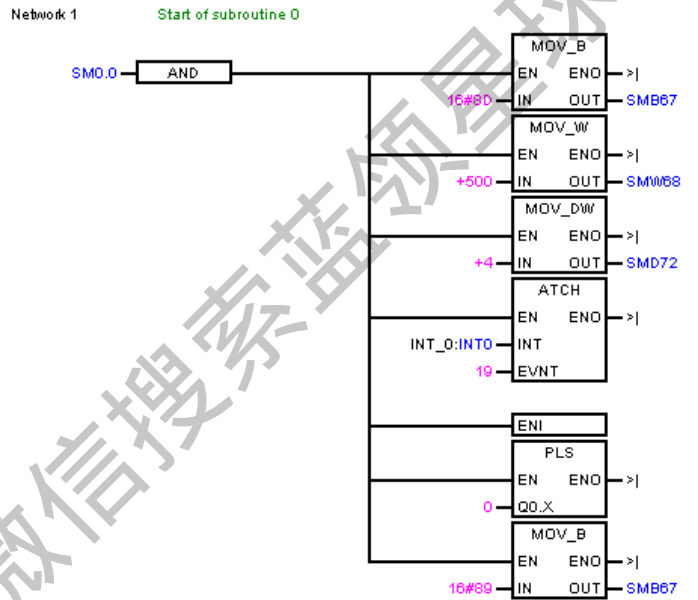
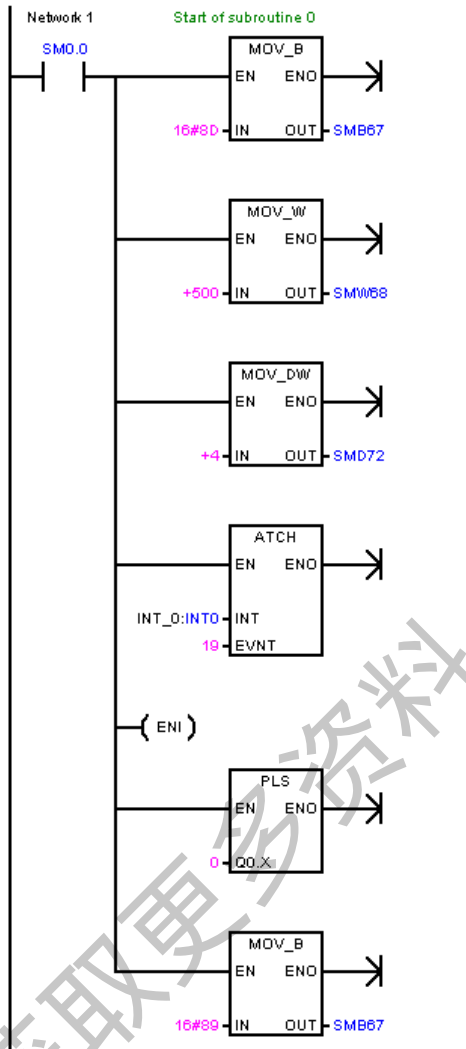
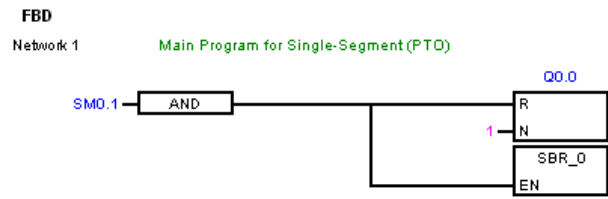
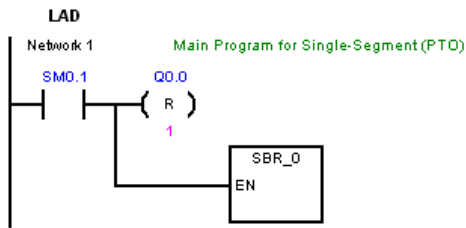
NETWORK 1 // 中断0开始
// 如果当前循环时间为500毫秒:
// 将循环时间设为1000毫秒, 并生成4次脉冲
LDW= SMD72 +500
MOVW +1000 SMD72
PLS 0
CRETI

```

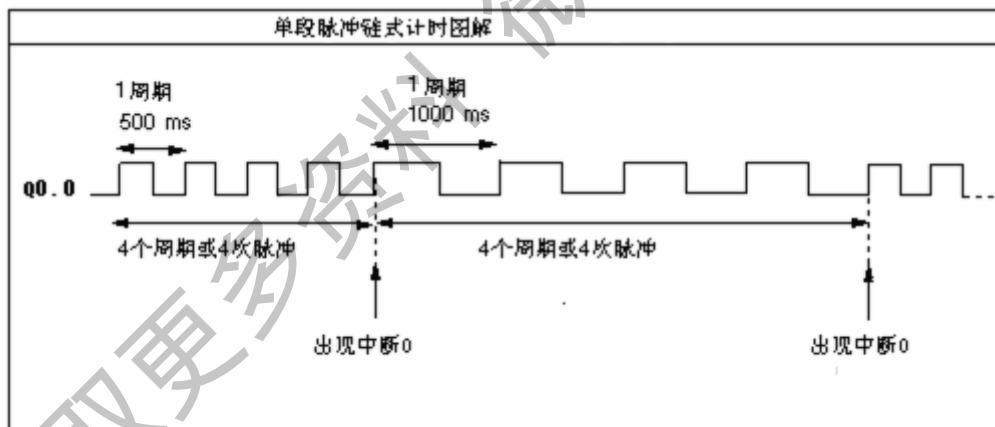
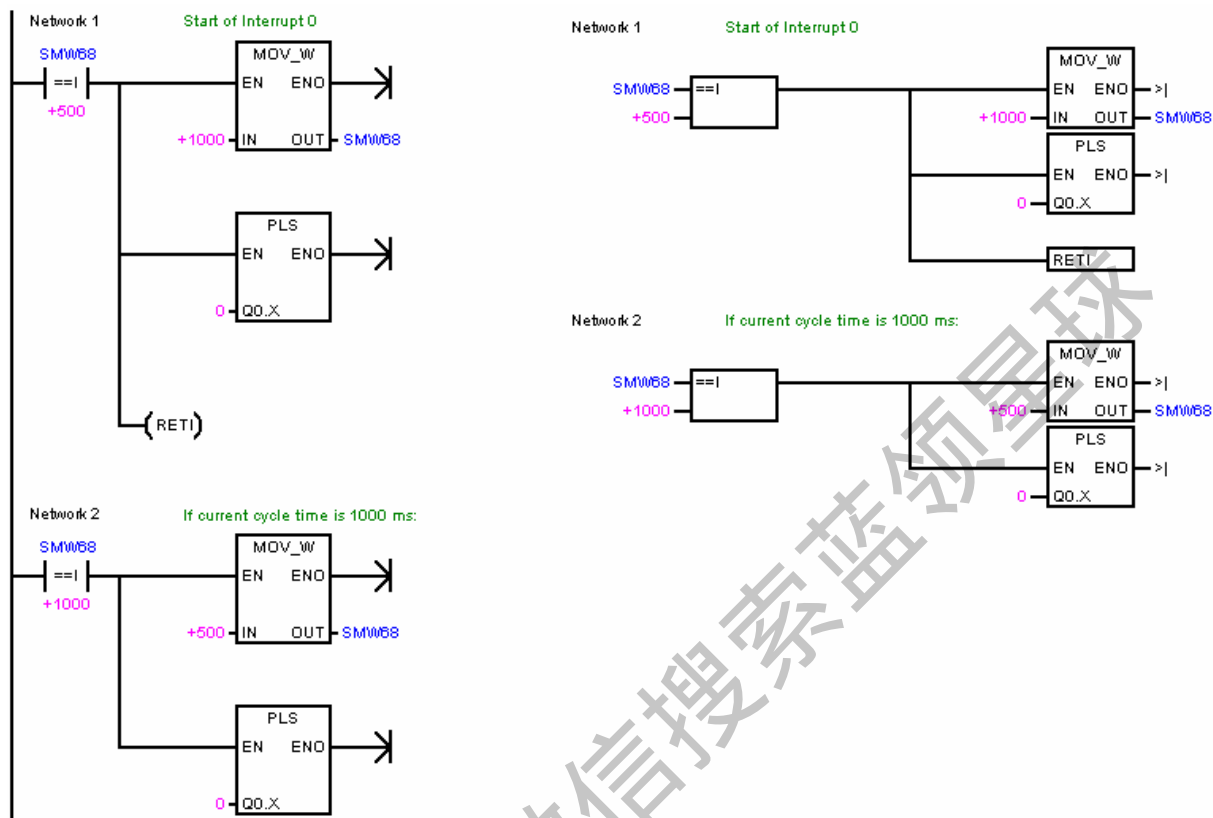
```

NETWORK 2
// 如果当前循环时间为1000毫秒:
// 将循环时间设为500毫秒, 并生成4次脉冲
LDW= SMD72 +1000
MOVW +500 SMD72
PLS 0

```



获取更多资料 微信搜索 蓝领星球



## 使用多段操作的脉冲链输出举例

```

NETWORK 1 // 用于多段脉冲链操作的主程序 (PTO)
// 首次扫描时, 将图像寄存器位设为低
// 并调用子例行程序0
LD SMO. 1
R Q0. 0 1
CALL SBR_0

```

```

NETWORK 1 // 子例行程序0开始
// 预载PTO轮廓表
LD SMO. 0

```

```

MOVB 3 VB500 // 将轮廓表段数设为3
// 段1:
MOVW +500 VW501 // 将段1的初始循环时间设为500毫秒
MOVW -2 VW503 // 将段1的Δ循环时间设为-2毫秒
MOVD +200 VD505 // 将段1中的脉冲数设为200
// 段2:
MOVW +100 VW509 // 将段2的初始循环时间设为100毫秒
MOVW +0 VW511 // 将段2的Δ循环时间设为0毫秒
MOVD +3400 VD513 // 将段2中的脉冲数设为3400
// 段3:
MOVW +100 VW517 // 将段3的初始循环时间设为100毫秒
MOVW +1 VW519 // 将段3的Δ循环时间设为1毫秒
MOVD +400 VD521 // 将段3中的脉冲数设为400

```

```

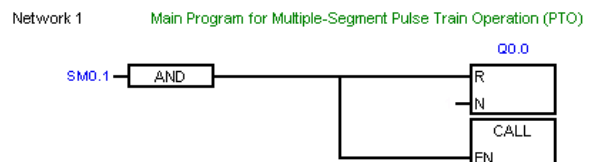
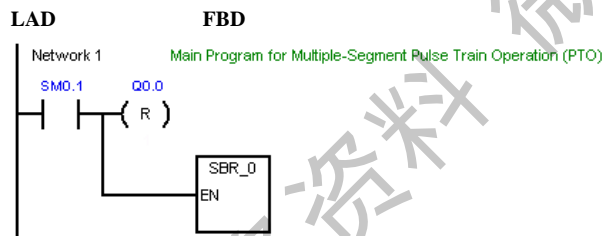
NETWORK 2
LD SMO.0
MOVB 16#A8 SMB67 // 设置控制字节:
// - 选择PTO操作
// - 选择多段操作
// - 选择毫秒递增
// - 启用PTO功能
MOVW +500 SMW168 // 将轮廓表的该起始地址
// 指定为V500。
ATCH INT_0 19 // 将中断例行程序0定义为
// 处理PTO完成中断的中断。
ENI // 全局中断启用
PLS 0 // 激活PTO操作, PLS0 => Q0.0
MOVB 16#89 SMB67 // 预载控制字节, 用于随后的
// 循环时间改动。

```

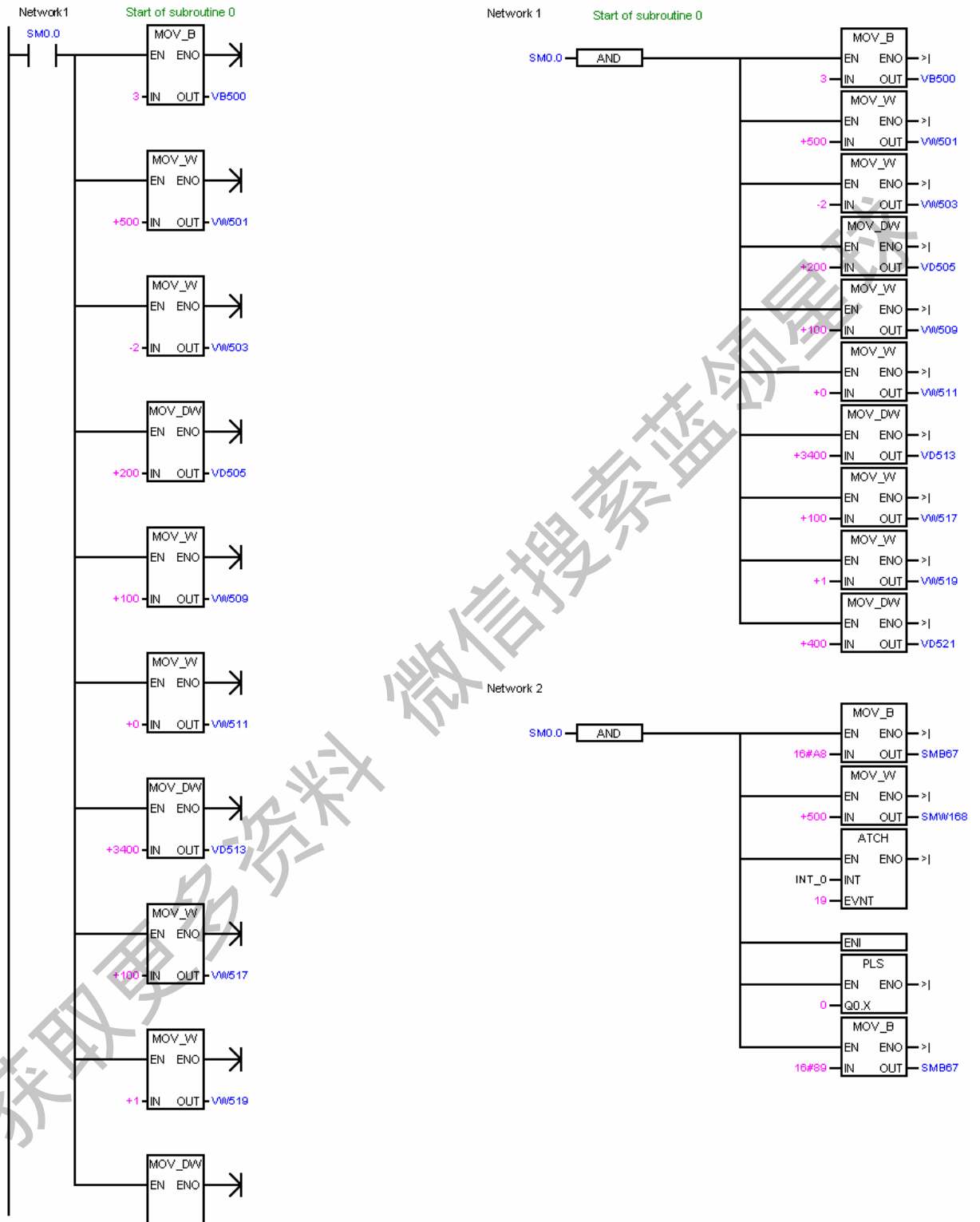
```

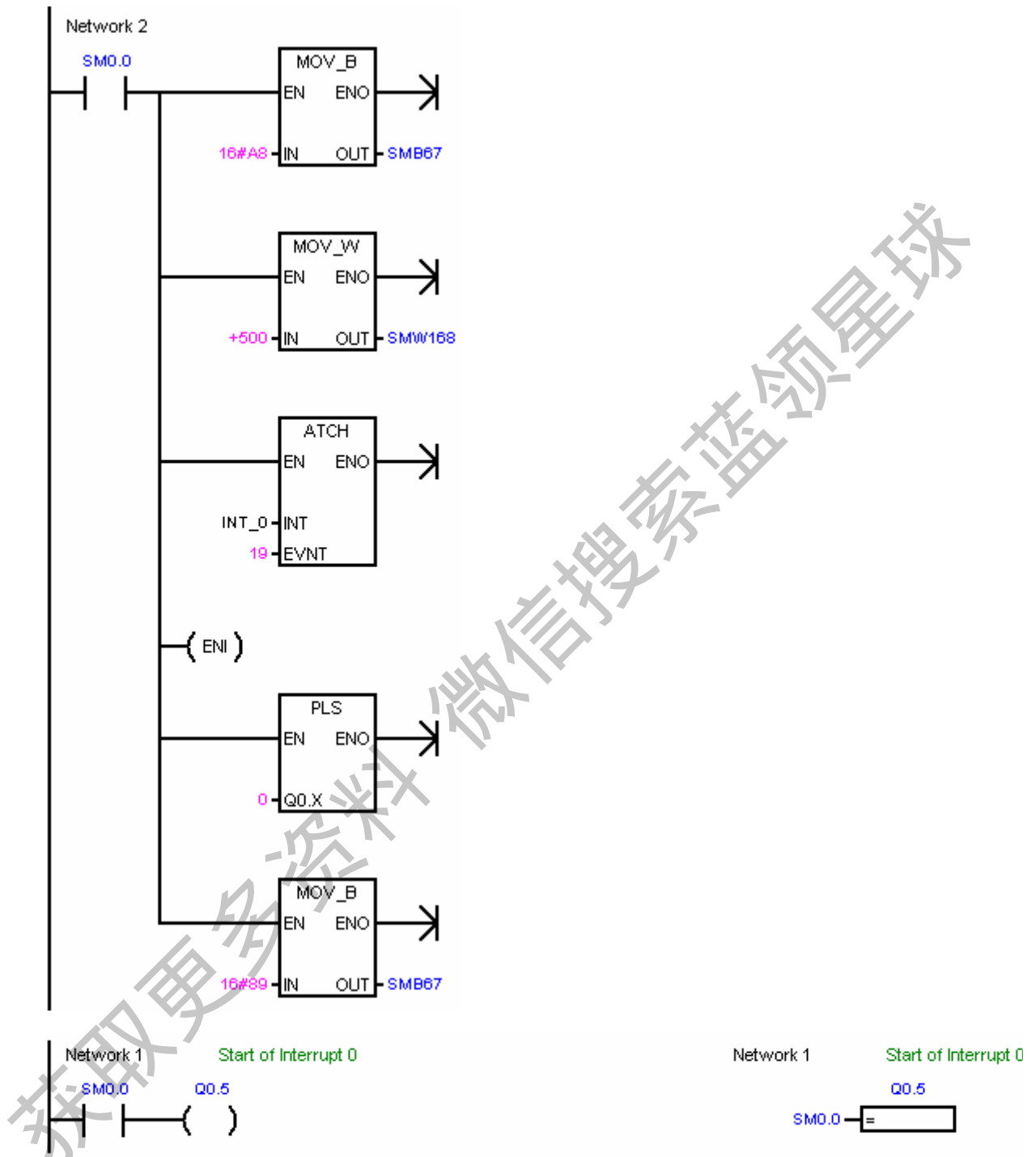
NETWORK 1 // 中断0开始
// PTO输出轮廓完成时, 打开输出Q0.5
LD SMO.0
= Q0.5

```









### 3.2.22 子例行程序编程

**调用指令** `[[[ENDKEEPN]]]`

本标题讨论下列主题：

- [使用子例行程序](#)
- [如何建立子例行程序](#)
- [如何终止子例行程序](#)
- [如何调用子例行程序](#)
- [嵌套和递归](#)

**使用子例行程序**[\(返回顶端\)](#)

子例行程序帮助您对程序进行分块。主程序中使用的指令决定具体子例行程序的执行状况。当主程序调用子例行程序并执行时，子例行程序执行全部指令直至结束。然后，系统将控制返回至调用子例行程序网络中的主程序。子例行程序用于为程序分段和分块，使其成为较小的、更易管理的块。在程序中调试和维护时，您可以利用这项优势。通过使用较小的程序块，对这些区域和整个程序简单地进行调试和排除故障。只在需要时才调用程序块，可以更有效地使用 PLC，因为所械某缘蚩榭瞻芪才脞葱忻看紊琛

最后，如果子例行程序仅引用参数和局部内存，则可移动子例行程序。为了移动子例行程序，应避免使用任何全局变量 / 符号（ I、Q、M、SM、AI、AQ、V、T、C、S、AC内存中的绝对地址）。如果子例行程序无调用参数（IN、OUT或IN\_OUT）或仅在LAD 讠嬷惺榭镁植勘淞浚 涂梢缘汲蚩永 谐缘蚤13. 涑既盲礞恒鱿牟俊

欲在程序中使用子例行程序，必须执行下列三项任务：

- 建立子例行程序
- 在子例行程序局部变量表中定义参数（如果有）
- 从适当的POU（从主程序或另一个子例行程序）调用子例行程序

当子例行程序被调用时，整个逻辑堆栈被保存，堆栈顶端被设为一，所有其他堆栈位置被设为零，控制被传输至调用子例行程 颀5 备米永 谐缘蛭甌端保 颜换指次 诘饕攻阅北A 舂氛 担 叩品砖氏饕美 谐缘颀

子例行程序和调用例行程序共用累加器。由于子例行程序的使用，对累加器不执行保存或恢复操作。

**用参数调用子例行程序**

子例行程序可能包含交接的参数。参数在子例行程序的局部变量表中定义。参数必须有一个符号名（最多为 23 个字符）、一个变量类型和一个数据类型。可向子例行程序交接 16 个参数或从子例行程序交接 16 个参数。局部变量表中的变量类型域定义参数是否交接至子例行程序（IN）、交接至或交接出子例行程序（IN\_OUT）或交接出子例行程 颀 OUT）。下表说明子例行程序的参数类型。欲增加参数条目，将光标放在您希望增加的类型的变量类型域上（IN、IN\_OUT 或 OUT）。单极滑鼠右键，获得选项菜单。选择“插入”选项，然后选择“下一行”选项。在当前条目的下方会显示所选类型的另一个参数条目。

**调用参数类型****说明**

**IN** 参数被交接至子例行程序。如果参数是直接地址（例如 VB10），在指定位置的数值被交接至子例行程序。如果参数是间接地址，（例如 \*AC1），位于指向位置的数值被交接至子例行程序。如果参数是数据常量（16#1234）或地址（&VB100），常量或地址数值被交接至子例行程序。

**IN\_OUT** 位于指定参数位置的数值被交接至子例行程序，来自子例行程序的结果数值被返回至相同的位置。输入 / 输出参数不允许使用常量（例如 16#1234）和地址（例如 &VB100）。

**OUT** 来自子例行程序的结果数值被返回至指定的参数位置。常量（例如 16#1234）和地址（例如 &VB100）不允许用作输出。

**TEMP** 未用作交接参数的任何本地内存不得用于子例行程序中的临时存储。

局部变量表中的数据类型域定义参数的大小和格式。下表列出了参数类型。

**参数数据类型****说明**

**功率流** 布尔功率流仅限于位（布尔）输入。该说明通知 STEP 7-Micro/WIN 该输入参数是依据位逻辑指令组合的功率流结果。布尔功率流输入必须首先在局部变量表中任何其他类型输入之前显示。只有输入参数允许这样使用。下例中的启用输入（EN）和 IN1 输入使用布尔逻辑。

**布尔** 该数据类型用于单位输入和输出。下例中的 IN3 是布尔输入。

**字节、字、双字** 这些数据类型分别识别 1、2 或 4 个字节不带符号的输入或输出参数。

**整数、双整数** 这些数据类型分别识别 2 或 4 个字节带符号的输入或输出参数。

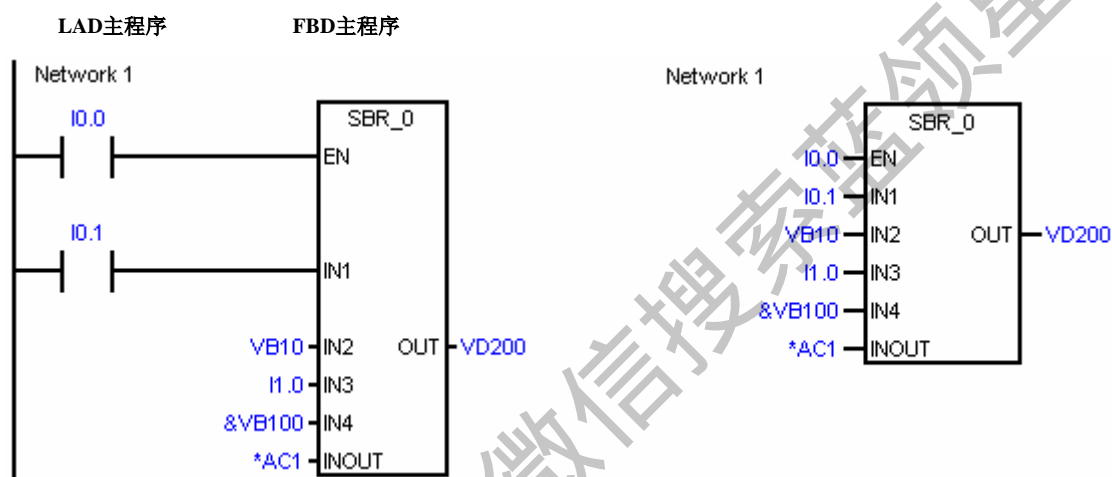
**实数** 该数据类型识别单精度（4 个字节）IEEE 浮点数值。

**字符串** 此数据类型被用作字符串的四字节指针

**功率流** 布尔功率流只可用于位（布尔）位。此说明告诉 STEP 7-Micro/WIN，这个输入参数是基于某位逻辑指令组合的功率流结果。布尔功率流输入必须在局部变量表中最先出现，早于任何其他类型的输入。唯有输入参数可以此方式使用。下例中的启用输入（EN）输入和 IN1 输入使用了布尔逻辑。

**子例行程序调用举例****用于 SBR\_0 的局部变量表**

	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
LB1	IN2	IN	BYTE	
L2.0	IN3	IN	BOOL	
LD3	IN4	IN	DWORD	
		IN		
LD7	INOUT	IN_OUT	REAL	
		IN_OUT		
LD11	OUT	OUT	REAL	
		OUT		



用Micro/WIN从LAD/FBD图形建立的STL代码在LAD、FBD或STL视图中启用显示。注释：Micro/WIN保留L内存（LB60-LB63）的四个上方字节，将其用于调用参数数据。

```

NETWORK 1
// L内存被用于保存布尔输入参数状态，
// 该参数在LAD和FBD中显示为功率流输入。这样可允许
// 本网络在LAD、FBD和STL编辑器中显示。
LD I0.0
= L60.0
LD I0.1
= L63.7
LD L60.0
CALL SBR_0 L63.7 VB10 I1.0 &VB100 *AC1 VD200

```

在STL编辑器中输入与以上显示相同的子例行程序调用的仅限STL的范例 注释：STL程序员可使用该简化的调用程序

```


NETWORK 1
// 该网络只能在STL编辑器中显示，
// 因为被用作功率流输入的布尔参数
// 未保存至L内存。
LD I0.0
CALL SBR_0 I0.1 VB10 I1.0 &VB100 *AC1 VD200

```

#### 用于LAD和FBD

建立子例行程序和定义调用参数后，STEP 7-

Micro/WIN自动生成子例行程序调用方框指令。根据局部变量表中对该子例行程序的说明，调用指令包含输入 / 输出参数的正确

冗余槽位 = (9) 永 谐 缘 蚝 蚪 鱿 衰 溢 噶 钐 髦 小 S 讷 碾 框 POU中插入子例行程序，从指令树中拖出子例行程序块  到 POU中。

#### 如何建立子例行程序

([返回顶部](#))

可采用下列一种方法建立子例行程序：

- 从“编辑”菜单，选择**插入 (Insert) > 子例行程序 (Subroutine)**
- 从“指令树”，用鼠标右键单击“程序块”图标，并从弹出菜单选择**插入 (Insert) > 子例行程序 (Subroutine)**
- 从“程序编辑器”窗口，用鼠标右键单击并从弹出菜单选择**插入 (Insert) > 子例行程序 (Subroutine)**

程序编辑器从先前的**POU**显示更改为新子例行程序。程序编辑器底部会出现一个新标记，代表新子例行程序。

此时，您可以对新子例行程序编程，或者保留子例行程序，返回您先前作业的 POU 位置：

- 如果您现在希望为子例行程序指定参数，您可以使用该子例行程序的**局部变量表**定义参数。

**注释：**

\*

请记住程序中每个 POU 都有一个独立的局部变量表。必须在选择该子例行程序标记后出现的局部变量表中为该子例行程序定义局部变量。编辑局部变量表时，必须确保已选择适当的标签。

\*\* 每个子例行程序调用的最大输入 / 输出参数限制为 16。如果您尝试下载超过该限制的程序，则会返回一则错误讯息。

- 选择子例行程序标记时，如果您希望为该子例行程序写入逻辑，在程序编辑器窗口中即可写入。
- 如果您希望对不同的 POU 编程，单击该 POU 的标签，以便在程序编辑器窗口中显示该 POU。

**不使用 RET 指令终止子例行程序**

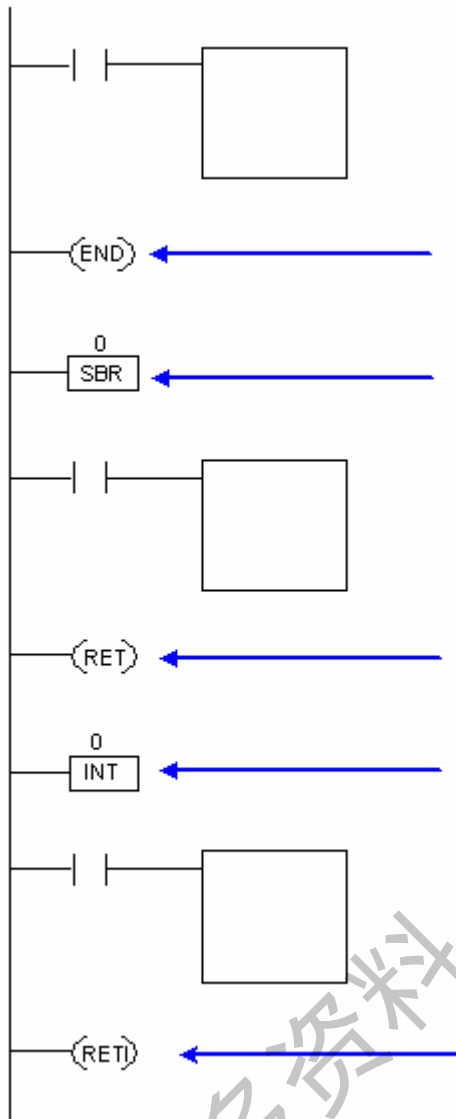
([返回顶端](#))

在子例行程序中不得使用 END (结束) 指令。

编辑器自动插入无条件 POU 终止指令 (END 用于 OB1, RET 用于 SBR, RETI 用于 INT)。以下显示一个范例。

箭头表示指令由 Micro/WIN 自动处理

获取更多资料 微信搜索 蓝领星球



### 如何调用子例行程序

(返回顶端)

插入新子例行程序并在该子例行程序的局部变量表中定义参数（如果有）后，您可在程序的另一个 POU 中放置一个子例行程序调用。（您可以从 OB1、另一个子例行程序或中断例行程序调用子例行程序；您不能从子例行程序本身调用子例行程序。）

#### LAD、FBD

对于 LAD 和 FBD 程序，在子例行程序局部变量表中为该子例行程序指定参数后，会生成一个定制调用方框指令。该调用指令自动包括子例行程序输入和输出参数的正确数目和类型。

欲在 LAD 或 FBD 程序的 POU 中插入调用指令：

1. 打开程序编辑器窗口中所需的 POU，滚动至您希望插入子例行程序调用的网络处。
2. 在指令树中，双击打开的“子例行程序”文件夹。您可将适当的调用指令从指令树拖放至程序编辑器中的正确的网络单元格中，或将光标放在程序编辑器中的单元格上，然后双击指令树中的调用指令。
3. 编辑程序中的调用指令参数并为每个参数指定有效操作数。有效操作数为：内存地址、常量、总体符号以及调用指令被放置的 POU 中的局部变量（并非被调用子例行程序中的局部变量）。

**注释：**如果您在子例行程序中插入一个调用指令，然后修改该子例行程序的局部变量表，调用指令则无效。您必须删除并重新插入调用指令。

#### STL

欲在 STL 程序中插入调用，使用 [调用指令](#)。

调用子例行程序时，保存整个逻辑堆栈，堆栈顶值被设为 1，其他所有堆栈位置均设为 0，控制转移至被调用的子例行程序。该子

例行程序完成后，用调用时保存的数值恢复堆栈，控制返回调用例行程序。  
子例行程序和调用例行程序共用累加器。不因使用子例行程序对累加器执行保存或恢复操作。

#### 嵌套和递归 (返回顶端)

程序中总共可有64个子例行程序（CPU 226XM可有128个子例行程序）。在主程序中，您可以嵌套子例行程序（在子例行程序中放置子例行程序调用指令），最大嵌套层任 8。您无法从中断例行程序嵌套子例行程序。子例行程序调用无法被放置在任何从中断例行程序调用的子例行程序中。递归（子例行程序调用自身）不被禁用，但您在子例行程序中使用递归时应当小心。

### 3.2.23 中断例行程序编程

中断由事件驱动。在启动中断例行程序之前，必须使中断事件与发生该事件时您希望执行的程序段建立联系。使用“附加中断”指令（ATCH）建立中断事件（由**中断事件号码**指定）与程序段（由中断例行程序号码指定）之间的联系。将中断事件附加于中侠 谱缘蚰保 弥卸献远 黄黎漫  
如果您使用全局禁用中断指令禁用所有的中断，中断事件的每次出现均被排队等候，直至使用全局启用中断指令重新启用中断。使用“拆卸中断”指令（DTCH）可拆卸中断事件与中断例行程序之间的联系，从而禁用单个中断事件。“拆卸”指令使中断返  
□醇せ翟蚩缓雕宰刺

#### 了解S7-200如何处理中断例行程序

作为对关联的内部或外部事件的应答，执行中断例行程序。一旦中断例行程序的最后一条指令被执行，控制被返回至主程序。您可以用执行“从中断指令有条件返回”指令（CRETI）的方法退出例行程序。

#### 使用中断例行程序指南和限制

中断程序为特殊内部或外部事件提供快速反应。您应当优化中断例行程序，执行某项具体任务，然后将控制返回至主程序。通过将中断例行程序保持为短小和简明扼要，可加快执行的速度，使其他程序不会受到长时间的延误。如果未能做到这一点，无法预料的情形可能导致主程序控制的装置出现非正常操作状况。

#### 限制

您不得在中断例行程序中使用DISI、ENI、HDEF、LSCR和END指令。

#### 中断的系统支持

由于接点、线圈和累加器逻辑可能受中断的影响，系统保存和重新载入说明累加器和指令操作状态的逻辑堆栈、累加器寄存器吞厥饴洼嫫晓 SM）。这样可避免因分支至中断例行程序和从中断例行程序分支而导致的主程序中断。

#### 在主程序和中断例行程序之间共享数据

您可以在主程序和一个或多个中断例行程序之间共享数据。因为无法预测 S7-200何时可能生成中断，最好限制中断例行程序和程序中其他位置使用的变量数目。由于主程序中指令执行被中断事件中中断时卸侠 谱缘蚤扇) 拇脞 岬贾鹿蚕砾 宛恢深您收稀J 褂弥卸侠 谱缘蚰植勘恣勘恻 匀繁V卸侠 谱缘蚪蚤褪味偈踟洼奴并且不盖写程序其他位置使用的数据。

您可以使用各种编程技巧，以确保在主程序和中断例行程序之间正确地共享数据。这些技巧限制存取共享内存位置的方法，或者使用共享内存位置预防出现指令序列中断。

- 对于共享单一变量的STL程序：如果共享数据是单字节、字或双字变量，且程序在STL中写入，则可用在非共享内存位置或累加器中存储共享数据操作数的直接数值的方法，确保正确的共享存取。
- 对于共享单一变量的LAD程序：如果共享数据是单字节、字或双字变量，且程序在LAD中写入，则可用建立仅使用“贫 敞噶睿 MOVB、MOVW、MOVD、MOVR）存取共享内存位置常规的方法，确保正确的共享存取。尽管很多LAD指令由STL指令的可中断序列组成，这些“移动”指令却是由单个STL指令组成，此类指令的执行不受中断事件的影响。
- 对于共享多个变量的STL或LAD程序：如果共享数据由各种相关的字节、字或双字组成，则可使用中断禁用/启用指令（DISI和ENI）控制中断例行程序的执行。在主程序中共享内存位置操作即将开始的点，禁用中断。一旦所有影响共享位置的措施均完成后，重新启用中断。在中断被禁用的时间内，不得执行中断例行程序，因此无法存取共享内存位置；但是，此种方法会导致对中断事件的延迟应答。

#### 从中断例行程序调用子例行程序

您可以从中断例行程序调用一个子例行程序嵌套级别。在被调用的中断例行程序和子例行程序之间共享累加器和逻辑堆栈。

#### S7-200支持的中断类型

S7-200支持以下中断例行程序类型：

- 通讯端口中断：S7-200生成允许程序控制通讯端口的事件。
- I/O中断：S7-200生成用于各种I/O状态不同变化的事件。这些事件允许程序对高速计数器、脉冲输出或输入的升高或降低状态作出应答。
- 时基中断：S7-200生成允许程序按照具体间隔作出应答的事件。

#### 通讯端口中断

可用程序控制S7-

200的串行通讯端口。此种操作通讯端口的模式被称作自由端口模式。在自由端口模式中，程序定义波特率、每个字符的位、奇夹Q 樞托 楞? 商崂 敖邮侧焙汀按 涸敞卸希 谱缘蚤 侧耐< 丁O呢榭鼠卧摹按 浜徒邮侧敞噶睇

#### I/O中断

I/O中断包括上升/下降边缘中断、高速计数器中断和脉冲链输出中断。S7-

200可生成输入（I0.0、I0.1、I0.2或I0.3）上升和/或下降边缘中断。可为每个此类输入点捕获上升边缘和下降边缘事件。这些上陆当吹凳录 捎糜诒硃驹谱录 (5) 北甌育12创 森淖纯航

高速计数器中断允许您对诸如以下之类的条件作出应答：当前值达到预设值，可能与转轴旋转方向逆转对应的计数方向的改变或

计数器外部重设。每种此类高速计数器事件均允许针对按照可编程逻辑控制器扫描速度控制的高速事件采取实时措施。脉冲链输出中断发出输出预定数目脉冲完成的立即通知。脉冲链输出的最常用用法是步进器马达控制。

您可以用将中断例行程序附加在相关I/O事件上的方法，启用上述每种中断。

#### 时基中断

时基中断包括定时中断和计时器T32/T96中断。您可以使用定时中断基于循环指定需要采取的措施。循环时间被设为从1毫秒至255毫秒每1毫秒递增一次。您必须在SMB34中将定时中断的循环时间设为0，在SMB35中将定时中断的循环时间设为1。

每次计时器失效时，定时中断事件将控制传输给适当的中断例行程序。通常您使用定时中断控制模拟输入取样或定期执行PID环

当您将中断例行程序附加在定时中断事件上时，则启用定时中断，且计时开始。在附加的过程中，系统捕获循环时间数值，因此其后对SMB34和SMB35所作的改动不会影响循环时间。欲改动循环时间，您必须修改循环时间数值，然后将中断例行程序重新附加在定时中断事件上。重新附加时，定时中断功能从以前的附件中清除所有的累计时间，并开始用新数值计时。

时间中断被启用后，则持续运行，每当指定的时间间隔失效时，执行附加中断例行程序。如果您退出RUN（运行）模式或分离定时中断，定时中断被禁用。如果全局禁用中断指令被执行，定时中断继续进行。每次定时中断出现均排队等候（直至中断被启用或队列已满）。

计时器T32/T96中断不允许对指定时间间隔完成及时作出应答。仅在1毫秒分辨率打开延迟（TON）和关闭延迟（TOF）计时器T32/T96中支持此类中断。否则T32和T96按照正常情况作业。一旦中断被启用，在S7-200中执行的正常1毫秒计时器更新的过程中，当用计时器的当前值等于预设时间数值时，即执行附加中断例行程序。您用将中断例行程序附加至T32/T96中断事件的方法，启用此类中断。

#### 中断优先级别和入队等候

S7-

200在中断各自的优先级别组内按照先来先服务的原则为中断提供服务。在任何时刻，只能执行一个用户中断例行程序。一旦桓鲋卸侠 谐缘蚩 贾葱校 蛞恢澈葱兄镣瓿伞 2 荒馨涣砒桓鲋卸侠 谐缘蛟は扰眠眨 词故款 晒畔燃伺鸬睦 谐缘颀U 洼 破砒桓鲋卸鲜猓(5) 闹卸先父拥却 惹

#### 每个中断队列最大条目数

队列	CPU 221、CPU 222 CPU 224	CPU 224 XP CPU 226和CPU 226XM
通讯队列	4	
I/O中断队列	16	
定时中断队列	8	

一般而言，出现的中断数目会超出队列能够容纳的数目。因此，队列溢出内存位（识别已经丢失的中断事件类型）由系统保持。下表显示中断队列溢出位。您应当仅在中断例行程序中使用这些位，因为当队列排空时这些位会被重设，控制被返回主程序。

#### 中断队列溢出位

说明（0 = 无溢出，1 = 溢出） SM位

通讯队列	SM4.0
I/O中断队列	SM4.1
定时中断队列	SM4.2

可采用下列一种方法建立中断例行程序：

- 从“编辑”菜单，选择**插入（Insert）> 中断（Interrupt）**。
- 从指令树，用鼠标右键单击“程序块”图标并从弹出菜单选择**插入（Insert）> 中断（Interrupt）**。
- 从“程序编辑器”窗口，从弹出菜单用鼠标右键单击**插入（Insert）> 中断（Interrupt）**。

程序编辑器从先前的POU显示更改为新中断例行程序。在程序编辑器的底部会出现一个新标记，代表新中断例行程序。

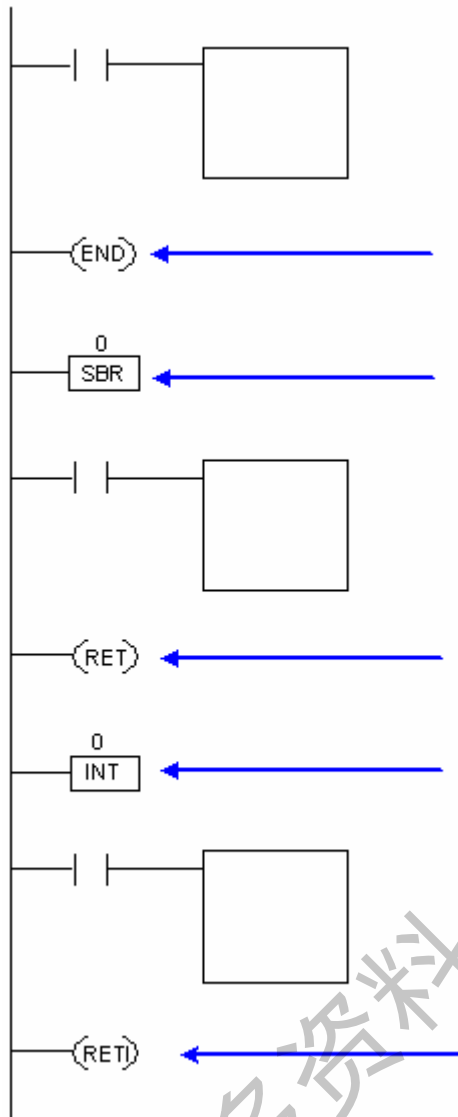
一个程序中总共可有128个中断。在各自的优先赋值范围内，PLC采用先来先服务的原则为中断提供服务。在任何时刻，只能执行一个用户中断例行程序。一旦一个中断例行程序开始执行，则一直执行至完成。不能被另一个中断例行程序预先排空，即使是更高优先级别的例行程序。正在处理另一个中断时发生的中断入队等待处理。

#### 注释：

编辑器自动插入无条件POU终止指令（END用于OB1，RET用于SBR，RETI用于INT），以下显示一个范例。

箭头表示指令由Micro/WIN自动处理





另请参阅：  
[队列溢出错误（中断）](#)  
[中断事件时间间隔赋值](#)  
[中断事件优先级列表](#)  
[局部变量表](#)

**SIMATIC编程中断指令**

[LAD](#)

[FBD](#)

[STL](#)

**IEC编程中断指令**

[LAD](#)

[FBD](#)

### 3.2.24 局部变量表编程

本部分讨论下列主题：  
[您的程序需要使用局部变量吗？](#)  
[理解局部变量](#)  
[局部变量的说明类型](#)

[局部变量数据类型检查](#)  
[如何隐藏/显示局部变量表](#)  
[如何在局部变量表中赋值](#)  
[局部变量表举例](#)

### 您的程序是否需要使用局部变量？

[返回顶端](#)

使用局部变量有两种原因：

- 您希望建立不引用绝对地址或全局符号的可移动子例行程序。
- 您希望使用临时变量（说明为TEMP的局部变量）进行计算，以便释放PLC内存。

如果以上说明对您的情形不适用，您无须使用局部变量；您可以在符号表（SIMATIC）或全局变量表（IEC）中定义符号数值，  
 械姆 攀 刀丁邇 直淞俊

### 理解局部变量

[返回顶端](#)

您可以使用程序编辑器的局部变量表指定对个别子例行程序或中断例行程序独特的变量。

局部变量可用作传递至子例行程序并可用于增加子例行程序的移动性或重新使用子例行程序的参数。

程序中的每个POU都有自己的局部变量表，配备64个字节的L内存。这些局部变量表允许您定义具有范围限制的变量；局部变量  
 辉消 8.帽淞康 POU中才有效。相反，在每个POU中均有效的全局符号只能在符号表/全局变量表中定义。当您为全局符号和  
 植勘淞渴褂孟嗤 姆 芭 保g INPUT1），定义局部变量的POU中的局部定义优先，全局定义用于其他POU。

在局部变量表中赋值时，您指定说明类型（TEMP、IN、IN\_OUT或OUT）和数据类型（请参阅SIMATIC和IEC 1131-  
 3数据类型），但不指定内存地址；程序编辑器自动在L内存区中为所有的局部变量指定内存位置。

局部变量表符号地址分配会将一符号名与存储有关数据值的L内存地址关联。局部变量表不支持向符号名直接赋值的符号常量  
 (这在符号或全局变量表中是许可的)。



**提示：** PLC不会将局部变量数据值初始为零。您必须在程序逻辑中初始化您使用的局部变量。

### 局部变量的说明类型

[返回顶端](#)

对局部变量赋值指定的类型取决于在其中赋值的POU。主程序（OB1）、中断例行程序和子例行程序可使用临时（TEMP）变量  
 V 挥性谥葱铎械保 偈北淞坎趴指茫 槽葱型瓠珊瑚 偈北淞靠杀恢甌础W永 谱缘蛞部墙褂玫瑰貌问 IN、IN\_OUT、OUT）。

#### 说明类型

#### 说明

IN 调用POU提供的输入参数。

OUT 返回调用POU的输出参数。

IN\_OUT 数值由调用POU提供的参数，由子例行程序修改，然后返回调用POU。

TEMPORARY 临时保存在局部数据堆栈中的临时变量。一旦POU完全执行，临时变量数值则无  
 法再用。在两次POU执行之间，临时变量不保持其数值。

### 局部变量数据类型检查

[返回顶端](#)

将局部变量作为子例行程序参数传递时，在该子例行程序局部变量表中指定的数据类型必须与调用POU中数值的数据类型相匹配

举例：

您从OB1调用SBR0，将称为INPUT1的全局符号用作子例行程序的输入参数。  
 在SBR0的局部变量表中，您已经将一个称为FIRST的局部变量定义为输入参数。

当OB1调用SBR0时，INPUT1数值被传递至FIRST。

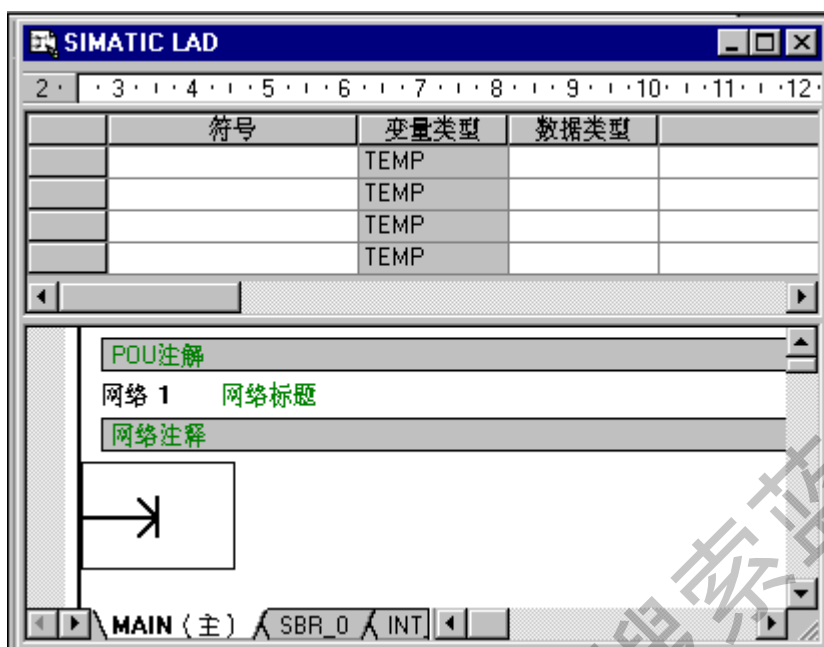
INPUT1和FIRST的数据类型必须匹配。

如果INPUT1是实数，FIRST也是实数，则数据类型匹配。如果INPUT1是实数，但FIRST是整数，则数据类型不匹配，  
 挥芯勒 苏庖准砦媳 缘蚤拍鞞噤

### 检视/隐藏局部变量表

[返回顶端](#)

如果您将水平分裂条拉至程序编辑器窗口的顶部，则局部变量表不再显示，但仍旧存在。将分裂条下拉即可再次显示局部变量表。



#### 如何在局部变量表中赋值

[返回顶端](#)

#### 注释：\*

在程序中使用局部变量之前，在局部变量表中赋值最为有效。在程序中使用符号名时，程序编辑器首先检查适当 POU 的局部变量表，然后检查符号表 / 全局变量表。

如果符号名在这两处均未定义，程序编辑器则将其视为未定义的全局符号；此类符号用绿色波浪状下划线标明。程序编辑器不会自动重新读取局部变量表并对您的程序逻辑做出修正。如果您后来进行了定义该符号名的数据类型分配（在局部变量表中），您必须以手动方式在该符号名前插入一个井号（#），例如：#UndefinedLocalVar（在程序逻辑中）。

\*\* 每个子例行程序调用的输入 / 输出参数的最大限制是 16，如果您尝试下载的程序超过此一限制，会返回一则错误信息。

#### 如何输入第一个局部变量赋值

欲在局部变量表中赋值，请遵循以下步骤。

1. 确保正确的 POU 在程序编辑器窗口中显示，如有必要，单击所需的 POU 标记。（因为每个 POU 都有自己的局部变量表）
2. 如果局部变量表处于隐藏状态，下拉水平分裂条，显示局部变量表。（请参阅 [检视 / 隐藏局部变量表](#)。）
3. 为需要定义的变量选择具有正确说明类型的行，在“名称”域中为该变量键入一个名称。（如果您在 OB1 或中断例行程序中赋值，局部变量表只包含 TEMP 变量。如果您在子例行程序中赋值，局部变量表包含 IN、IN\_OUT、OUT 和 TEMP 变量。）您不需要在局部变量表中的变量名前加井号。井号只用在程序代码中的局部变量前。

#### 注释：

\* 局部变量名最多可包含 23 个字母数字字符和下划线，也可包含扩展字符（ASCII 128 至 ASCII 255）。第一个字符只能是字母或扩展字符。将关键字用作符号名属于非法，名称的第一个字符是数字，或名称包含非字母数字字符。

\*\* 局部变量表变量名被下载并存储于 CPU 内存中，使用较长的变量名可能减少用于存储程序的内存空间。

4. 在“数据类型”域中单击鼠标指针，并使用列表框为局部变量选择适当的数据类型。

#### 注释：

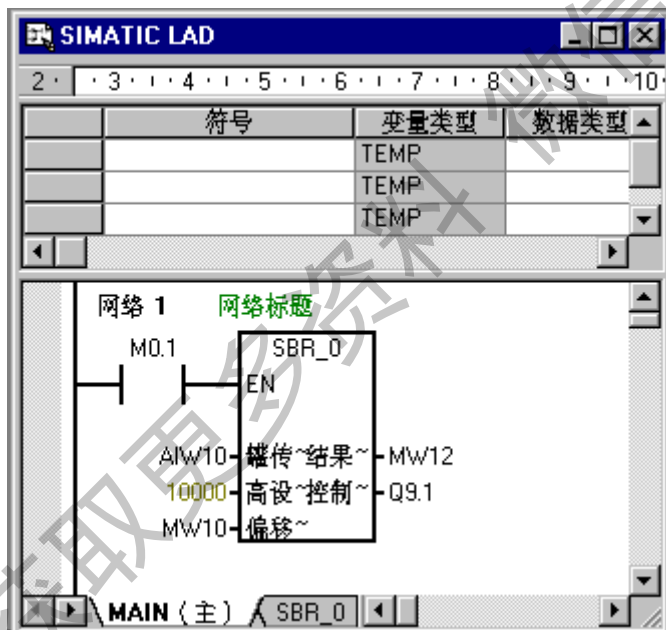
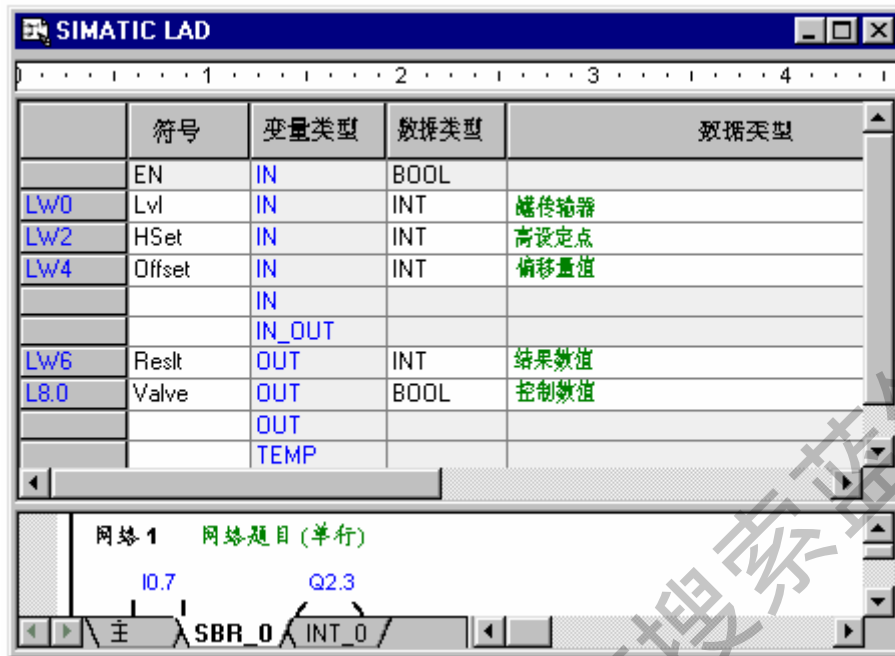
当您为局部变量指定为子例行程序的参数时，您必须保证为局部变量指定的数据类型不与子例行程序调用中使用的操作数发生冲突。（请参阅上文中的 [数据类型检查](#)。）

为“名称”和“数据类型”域提供数值后，程序编辑器会自动为局部变量指定 L 内存地址。

#### 如何在局部变量表中输入附加赋值

对于 OB1 和中断例行程序，局部变量表显示一组已被预先定义为 TEMP 变量的行。在 OB1 或中断例行程序中，只能使用这种说明类型。欲在表中增加更多行，只需单击最后一行中的一个单元格，然后使用 ENTER 键移过该行并向右移动，即自动生成一个新行。对于子例行程序，局部变量表显示按照以下顺序预先定义说明类型的一组行：IN、IN\_OUT、OUT 和 TEMP。您不能改变该顺序。局部变量在该表中的顺序必须符合您为子例行程序调用指令进行操作数赋值时对应的操作数顺序。如果您希望增加附加局部变量，必须用鼠标右键单击现有行，并使用弹出菜单插入与单击行类型相同的另一局部变量。选择 **插入 (Insert) > 行 (Row)**，在所选行的上方插入新行，或选择 **插入 (Insert) > 行下方 (Below Row)**，在所选行下方插入新行。

局部变量表举例  
[返回顶端](#)



另请参阅:

[子例行程序](#)  
[SIMATIC和IEC 1131-3数据类型](#)  
[关键字](#)

### 3.2.25 使用数据块在PLC V内存中存储数据

使用下列一种方法存取数据块：



- 单击浏览条上的“数据块”按钮。
- 选择菜单命令**检视 (V) > 数据块 (D)**。
- 打开**指令树**中的“数据块”文件夹，然后双击某块页图标。

通过插入新数据块页标记，将您的数据块V内存赋值分成多个功能组：

- 单击数据块窗口，然后选取菜单命令**编辑 (E) > 插入 (I) > 数据块 (D)**
- 在**指令树**中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取**插入 (I) > 数据块 (D)**
- 用鼠标右键单击数据块窗口，然后在弹出菜单中选取**插入 (I) > 数据块 (D)**
- 标记的最大数目为128。如果您使用向导，有关标记会被自动创建以支持向导功能。您可以创建的标记的最大数目为（由MicroWin自动创建的标记数目）。请使用Windows剪贴板合并标记数据；方法为使用剪切和粘贴由一个标记转移到另一块页。

重新命名和保护数据块页标记：

- 在**指令树**中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取**重新命名**。您也可以在指令树内直接重新命名数据块页，方法为单击该标记页名称两次（动作要慢一些，以免解释成双击）；然后编辑该标记名。数据块编辑器提供相同的重新命名功能，方法为用鼠标右键直接单击该标记名。
- 在**指令树**中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取**属性**。由此，您可以重新命名该数据块标记和指定作者。属性对话框的**保护**标记令您能够用密码保护单个数据块标记。受保护的标记会显示锁图标。数据块编辑器向导会创建不能重新命名且包含只读数据值的受保护标记。

导入和导出数据块数据至ASCII文本文件

- [由ASCII文本文件导入数据块](#)
- [由ASCII文本文件导出数据块](#)

本标题讨论下列主题：

[在数据块中进行地址和数据值赋值](#)

[数据块举例](#)

[在数据块编辑器中使用“剪切”、“复制”和“粘贴”](#)

[使用ASCII常量的限制](#)

[理解和解决错误](#)

[将数据块下载至PLC](#)

[从PLC上载数据块](#)

[上载后保留数据块格式](#)

[从CPU RAM建立数据块](#)

在数据块中进行地址和数据值赋值

[返回顶端](#)

数据块仅允许您对V内存进行初始数据或ASCII字符赋值。您可以对V内存的字节 (V或VB)、字 (VW) 或双字 (VD) 赋值。注解（前面带双正斜线//）是选用项目。

- 数据块的第一行必须包含一个明确地址赋值（绝对或符号地址），其后的行可包含明确或隐含地址赋值。当您在单地址赋值后键入多个数据值或键入仅包含数据值的行时，由编辑器指定隐含地址赋值。编辑器根据先前的地址分配及数据值大小（字节、字或双字）指定适当的V内存数量。
- 数据块编辑器是一种自由格式文本编辑器，对特定类型的信息无具体域。键入一行后，按ENTER键，数据块编辑器格式校核并重新显示行。数据块编辑器接受大小写字母并允许使用逗号、制表符。
- 在完成一赋值行后按CTRL-ENTER键组合，会令地址自动增加至下一个可用地址。

[常量格式详细说明](#)

[数据块一般规则](#)

```

数据块
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

VB0  248      //明确地址赋值：VB0数据值：248。

VB1  249, 250, 251 //单行中多个数据值。
      //隐含地址赋值：
      //VB2包含数据值250。
      //VB3包含数据值251。

VB4  252      //不能使用先前指定的地址（VB0-VB3）。

      253, 254, 255 //无明确地址赋值的行。
      //数据值隐含指定给VB5、VB6、VB7。

VW8  256, 257  //新数据类型（字）。隐含将数据值
      //257指定给V内存VB10-VB11。

X    65536     //地址赋值不能与先前的明确赋值发生冲突。
      //数据值65536要求双字数据类型（VD内存），
      //但上一个明确地址赋值
      //是字内存（VW8）。编辑器标志错误。

      //您可以从地址中省略尺寸规格。
      //然后您可以对随后的数据使用隐含地址赋值，
      //不必考虑数据的尺寸。

V12  258      //字值明确指定给V12-V13。
      65537    //双字值隐含指定给V14-V17。

```

数据块举例

[返回顶端](#)

直接地址和数值

```

数据块
//
//数据块注解
//
VB0    255           //字节值从VB0开始
VW2    256           //字值从VW2开始
VD4    700.59       //双字真值从VD4开始
VB8    -35          //字节值从VB8开始
VW10   16#A         //字值HEX从VW10开始
VD14   146879       //双字值从VD14开始
VW20   2, 4, 16, 32, 64 //字值表从VD14开始
      -2, -4, -16, -32, -64 //扩展至多行的数值不能出现在列中
//第一列为内存地址保留
VB45   'Up'         //起始于VB45的双字节ASCII字符串
VW90   65535        //起始于VW90的字值
V100   255, 'abcd', 65535, 1.0 //起始于VB100的未定义数值：可混合不同尺寸
VB110  255          //与输入B110 255相同
VW120  256          //与输入W120 256相同
VD130  700.59       //与输入D130 700.59相同
V140   255, 'abc', 65535, 1.0 //与输入140相同：必须从列1开始
VW150  2#1010101010101010 //二进制字值
VD152  2#1100110011001100110011001100 //二进制双字值

```

符号地址和符号数值

符号表			
符号	地址	注解	
1	符号地址	VB1	符号地址
2	符号常数	200	符号常数

用户1 \ POU符号 /

```

数据块
VB0    100
符号地址 符号常数 //为VB1指定一个200数值
用户1 /

```

替代二进制输入方法（上窗口）和结果二进制格式（下窗口）

```

数据块
VB0    2 1 0 true false off on
用户1 /

```

```



数据块
VB0    2#10010100 //按Enter键后二进制模式
用户1 /

```

在数据块编辑器中使用“剪切”、“复制”和“粘贴”

[返回顶端](#)

当在数据块中执行剪切、复制、粘贴时，重要的是通过鼠标单击设置适当的上下文：

如果您单击指令树中的数据块标记页  图标，或者单击数据块窗口中的标记名 ，  
 则 霰县且成喜僮鳌 T 洼松舷掌闹杏檬孟暖壹 七 颯 突嵯允咀饕糜谩 黎 菘楸县且车牟僮鞞 霾说 ？  
 如果您单击数据块编辑器中的文字区域，那么您将在数据块窗口中已选定(突出显示)的文字上操作。在此上下文中用鼠标右  
 七 颯 突嵯允咀饕糜谩 黎 菘楸县且衬谔 《 刁 淖 稚 系 牟 僮 鞞 霾 说 ？

### 使用ASCII常量的限制

[返回顶端](#)

有效的ASCII赋值：

唯有字节地址(V或VB)能够与长ASCII常量(使用单引号或双引号格式)合用：

VBO 'A'、VWO 'AB'、VDO 'ABCD'

就3和5或更多字节而言，您必须使用V或VB地址前缀 VBO 'ABC'、VBO 'ABCDE'、VBO 'ABCDEFGH IJK'



双引号ASCII常量字符串格式存储一个前导长度字节和字符串 VBO "A"、VBO "AB"、VBO "ABC"、VBO "ABCD"、VBO "ABCDEFGH IJK"

[常量格式详细说明](#)

### 理解和解决错误

[返回顶端](#)

一旦在包含错误的行尾按ENTER键，立即会在数据块左页边显示输入错误。您必须纠正全部输入错误，才能成功地编译。

 vbbo 255  
 vbo 300

// (红色文本 - 非法语法错误) 应为vb0

// (红色曲线下划线 - 非法使用错误) 一个字节最大为255

引起输入错误的条件包括：

- 指定错误内存区 (V是唯一允许使用的内存区)
- 当数据值实际要求较大的尺寸时(例如，数据值256过大，无法在VB地址中存储—要求使用VW地址)，在地址赋值兄付 骋荒洼姆叽纾 卫纸译厝郑
- 在一行中使用错误序列：在数据值之后(而不是在数据值之前)键入内存地址
- 使用非法语法或无效数值
- 尝试使用符号，而不是使用绝对V内存地址(数据块中不允许使用符号)
- 未能适当地指定注解(双正斜线必须位于注解之前：//注解样本)如果数据块是现用窗口，您可以使用菜单命令 **PLC > 编译 (Compile)** 编译数据块。如果数据块不是现用窗口，您依然可以编译数据块：使用菜单命令 **PLC > 全部编译 (Compile All)**。

编译数据块时，如果编译程序发现错误，会在“输出窗口”显示错误。将光标置于“输出窗口”中的错误讯息上，双击该讯息  
 谿 崧 椽 翱 溢 邢 允 境 趵 刚 小  
 仅在编译后显示的错误包括：

编译数据块时，如果编译程序发现错误，会在“输出窗口”显示错误。将光标置于“输出窗口”中的错误讯息上，双击该讯息，  
 在数据块窗口中显示出错行。  
 仅在编译后显示的错误包括：

- 重复地址赋值(例如，如果您输入“VB1 249, 250”之类的行，则是对VB2进行250隐含赋值—您不得在别处对地址VB2进行其他不同的数据值赋值)
- 地址重叠(例如，如果您为VD0指定一个类似65536的双数值，则不得再对V1、V2或V3指定其他赋值，因为这些数值已被使用，是以VD0开始的双字的一部分)

### 将数据块下载至PLC

[返回顶端](#)

如果要编辑数据块，则需将数据块 **下载** 至PLC。只有在修改的数据块下载后您的编辑才会生效。  
 欲节省空间，您可以将您不希望下载至PLC的信息切换为打开/关闭。

### 从PLC上传数据块

[返回顶端](#)

您必须在STEP 7-Micro/WIN中打开一个项目，才能 **上传** 数据块。

如果PLC中的数据块与您打开的项目中的数据块不匹配(或者您打开的项目中没有数据块)，您只能上传该PLC的数据块地址  
 糠 趾 褪 莼 挡 糠 郑 荒 茺 显 刈 13. 狡 5 卍 芬 宰 纸 谿 蹈 裕 缴 显 兀 莼 狄 允 聘 裕 缴 显 兀 ò 匕 魏 卧 模 日 纛 谿 占  
 ，您可以将您不希望下载至PLC的信息切换为打开/关闭。

如果CPU中的数据块包含由Micro/WIN 4.0(或较新版本)下载的标记信息，那么由Micro/WIN



4.0的上载将会重新载入此标记结构。如果某数据块曾由较早的Micro/WIN版本下载且无标记信息，那么由Micro/WIN 4.0的上载将会将所有赋值放在同一个标记中。

#### 上载后保留DB格式

##### [返回顶端](#)

当您上载数据块时，下载至PLC的数据块条目格式被保留。

如果您在一行中输入多个条目，当您上载时，格式不被保留。每个条目在一个不同的行中显示。

例如，如果您在数据块中输入VB0 20、30、40、50并上载，格式更改为：

```
VB0      20
VB1      30
VB2      40
VB3      50
```

#### 从CPU RAM建立数据块

##### [返回顶端](#)

选择**PLC > 从RAM建立数据块 (Create Data Block from RAM)** 菜单命令，将CPU V内存保存至EEPROM。

从CPU读取V内存当前值，然后将V内存图像作为数据块下载。

#### 注释：

进行此操作时，CPU必须处于STOP（停止）模式。

倘若CPU

RAM内有已修改的V内存地址，并且这些地址在上一次数据块下载中未得到指定，那么在您执行“从RAM建立数据块”命令时，这一研需牡牡 司坊岂恢付ù 恒鲂碌谋县敲 g 罐LC\_DATA1)。当程序的执行向新的V地址写入数据时，或当状态图被用于修改碌膺地址时，此情形就可能发生。

#### 另请参阅：

[PLC电源中断后的数据保留](#)

[常量格式详细说明](#)

[通讯配置](#)

[PLC类型选择](#)

[系统块 \(PLC配置\)](#)



[错误讯息](#)

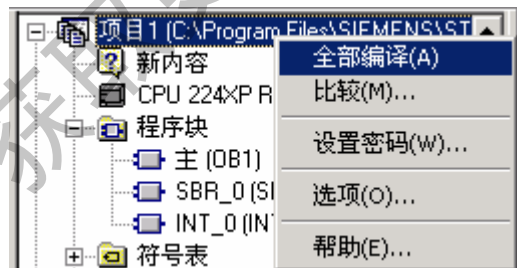
[入门指南目录](#)

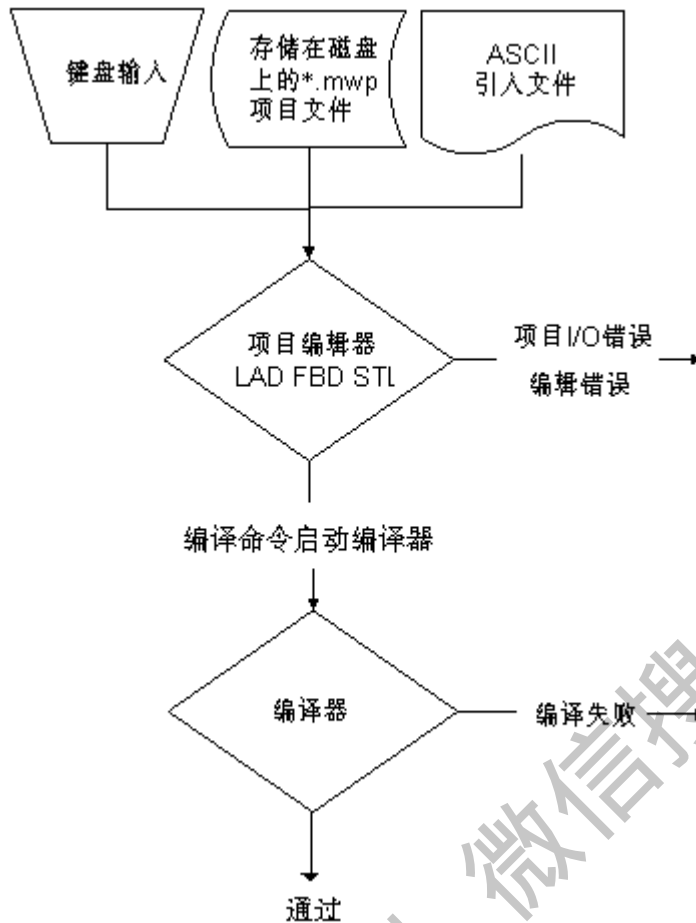
[应用程序用户参考手册](#)

## 3.2.26 编译项目

可使用下列一种方法启动STEP 7-Micro/WIN项目编译程序：

- 单击“编译” 按钮或选择菜单命令**PLC (PLC) > 编译 (Compile)**，编译现用窗口（程序块或数据块）。
- 单击“全部编译” 按钮或选择菜单命令**PLC (PLC) > 全部编译 (Compile All)**，编译全部项目元件（程序块、数据块和系统块）。
- 用鼠标右键单击**指令树**中的某个文件夹，然后由弹出菜单中选取编译命令。项目、程序块文件夹、系统块文件夹及数  
松榭募 卸加斜喊阮 睢





[打开项目](#) [范围检查](#)

设置PLC类型 项目文件 I/O 错误 LAD/FBD 编辑器错误 全部STEP 7-Micro/WIN 编译器错误均列在输出窗口中。双击错误，编辑器将滚动至错误位置。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译器。

另请参阅：  
[应用程序用户参考手册](#)  
[入门指南目录](#)

### 3.2.27 检视PLC内存用法的交叉引用总结

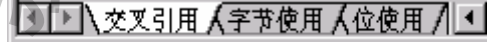
可使用下列一种方法检视“交叉引用”窗口：

- 选择菜单命令 **检视 (View) > 交叉引用 (Cross Reference)**



- 单击浏览条中的“交叉引用”按钮
- 打开 [指令树](#) 中的“交叉引用”文件夹，然后双击某引用或使用节点

欲存取“交叉引用”表、“字节用法”表或“位用法”表，单击位于“交叉引用”窗口底部的适当标记



本标题讨论下列主题：

[交叉引用表](#)  
[字节用法表](#)  
[位用法表](#)

[交叉引用表](#)  
[返回顶端](#)



**提示：** 您必须编译程序才能检视“交叉引用”表。

当您希望了解程序中是否已经使用和在哪里使用某一符号名或内存赋值时，可使用“交叉引用”表。“交叉引用”列表识别在程序中使用的所有操作数，并识别 POU、网络或行位置以及每次使用的操作数指令上下文。

**元素**指程序中使用的所有操作数。您可以在符号和绝对视图之间切换，改变全部操作数显示。（使用菜单命令 **检视 (View) > 符号编址 (Symbolic Addressing)**。）

**块**指使用操作数的 POU。

**位置**指使用操作数的行或网络。

**上下文**指使用操作数的程序指令。

LAD交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1	-
2 动2	MAIN (OB1)	网络 2	-
3 停止1	MAIN (OB1)	网络 1	-
4 停止2	MAIN (OB1)	网络 2	-
5 高等级	MAIN (OB1)	网络 1	-
6 高等级	MAIN (OB1)	网络 2	-

FBD交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1	OR
2 动2	MAIN (OB1)	网络 2	OR
3 停止1	MAIN (OB1)	网络 1	AND
4 停止2	MAIN (OB1)	网络 2	AND
5 高等级	MAIN (OB1)	网络 1	AND
6 高等级	MAIN (OB1)	网络 2	AND

STL交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1, 行 1	LD
2 动2	MAIN (OB1)	网络 2, 行 1	LD
3 停止1	MAIN (OB1)	网络 1, 行 3	A
4 停止2	MAIN (OB1)	网络 2, 行 3	A
5 高等级	MAIN (OB1)	网络 1, 行 4	AN
6 高等级	MAIN (OB1)	网络 2, 行 4	AN

字节用法表

[返回顶端](#)



**提示：** 您必须编译程序才能检视“字节用法”表。

“字节用法”表允许您查看程序中使用了哪些字节以及在哪些内存区使用，还可帮助您识别重复赋值错误。

**b**表示已经指定一个内存位。

**B**表示已经指定一个内存字节。

**W**表示已经指定一个字（16位）。

**D**表示已经指定一个双字（32位）。

**X**用于计时器和计数器。

范例1：解释字节用法表

该字节用法表范例显示相关程序使用下列内存位置：MB0中一个位；计数器C30；计时器T37。

字节	9	8	7	6	5	4	3	2	1	0
MB0										b
C0										
C10										
C20										
C30										X
T0										
T10										
T20										
T30				X						

范例2：识别重复赋值错误

该范例程序从MB 10.0开始进行重复内存赋值。

```

网络 1  网络标题
LD      M10.0
AB=    MB10, VB0
AW=    MW10, VW0
AD=    MD10, VD0

```

可检查字节用法表，识别不适当的赋值。因为双字要求四个字节，VB0行中应有4个相邻的D。与此相似，因为字要求2个字节，VB0中应有2个相邻的W。MB10行存在相同的问题，此外在多个赋值语句中使用MB10.0。

字节	9	8	7	6	5	4	3	2	1	0
VB0							D	D	W	B
MB0										
MB10							D	D	W	b

位用法表

[返回顶端](#)



**提示：** 您必须编译程序才能查看位用法表。

位用法表允许您查看程序中已经使用了哪些内存地址，可精确至位级别，还可帮助您识别重复赋值错误。

**b**表示已经指定一个内存位。

**B**表示已经指定一个内存字节。

**W**表示已经指定一个字（16位）。

**D**表示已经指定一个双字（32位）。

**X**用于计时器和计数器。

范例1：解释位用法表

该位用法表范例显示相关程序使用下列内存位置：从字节I0、位0、1、2、3、4、5和7；从字节Q0、位0、1、2、3、4和5；从字节M0、位1。

位	7	6	5	4	3	2	1	0
I0.0	b		b	b	b	b	b	b
Q0.0			b	b	b	b	b	b
M0.0							b	

范例2：识别重复赋值错误

该范例程序从MB 10.0开始进行重复内存赋值。

```

网络 1  网络标题
LD      M10.0
AB=     MB10, VW0
AW=     MW10, VWO
AD=     MD10, VDO
  
```

可检查位用法表，识别不适当的赋值。在适当的赋值程序中，字节中间不得有位值。BBBBBBb无效，而BBBBBBBb则有效。

嘿 慕婁丁彩视糜滥指担虫 τ 16个相邻的W)和双字赋值(应有32个相邻的D)。

位	7	6	5	4	3	2	1	0
M0.0								
M1.0								
M2.0								
M3.0								
M4.0								
M5.0								
M6.0								
M7.0								
M8.0								
M9.0								
M10.0	B	B	B	B	B	B	B	b
M11.0	W	W	W	W	W	W	W	W
M12.0	D	D	D	D	D	D	D	D
M13.0	D	D	D	D	D	D	D	D

另请参阅：


[通讯配置](#)  
[PLC类型选项](#)  
[系统块（PLC配置）](#)  
[错误](#)  
[在符号和绝对视图之间切换](#)

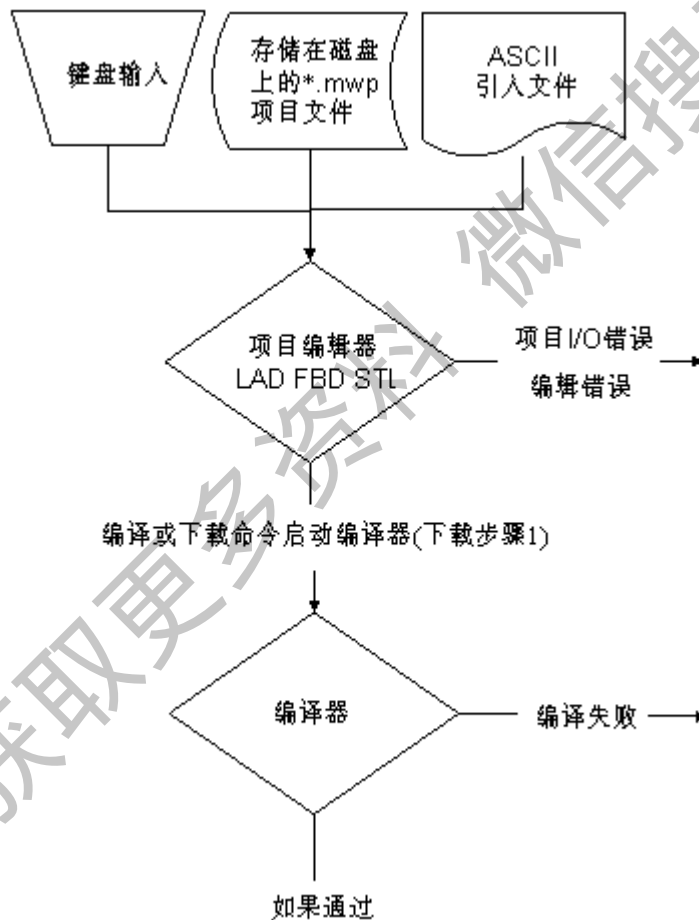
[入门指南目录](#)  
[应用程序用户参考手册](#)

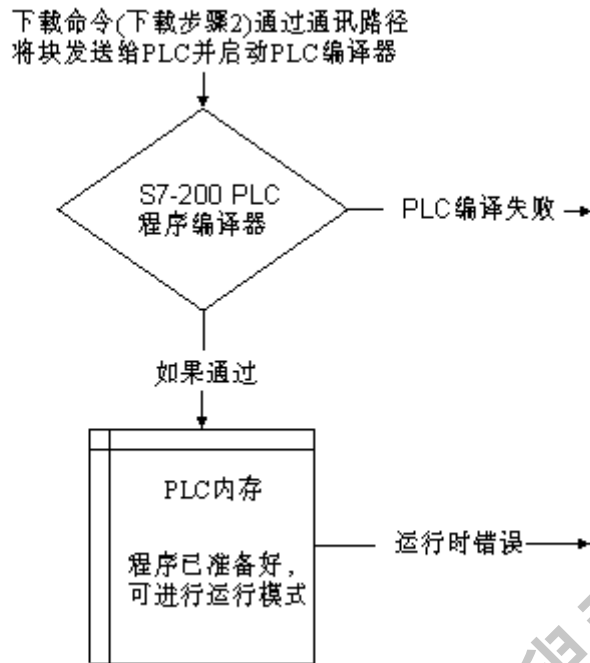
### 3.2.28 将程序下载至PLC

可使用下列一种方法从STEP 7-Micro/WIN向PLC下载项目元件：

- 单击“下载”  按钮。
- 选择菜单命令文件（File）> 下载（Download）。
- 按Ctrl+D快捷键组合。

单击“选项”按钮 ，显示和设置下载操作选项。‘上载下载管理器’窗口可帮助您控制上载、下载、清除及  
[UDM - 上载下载管理器详细说明](#)

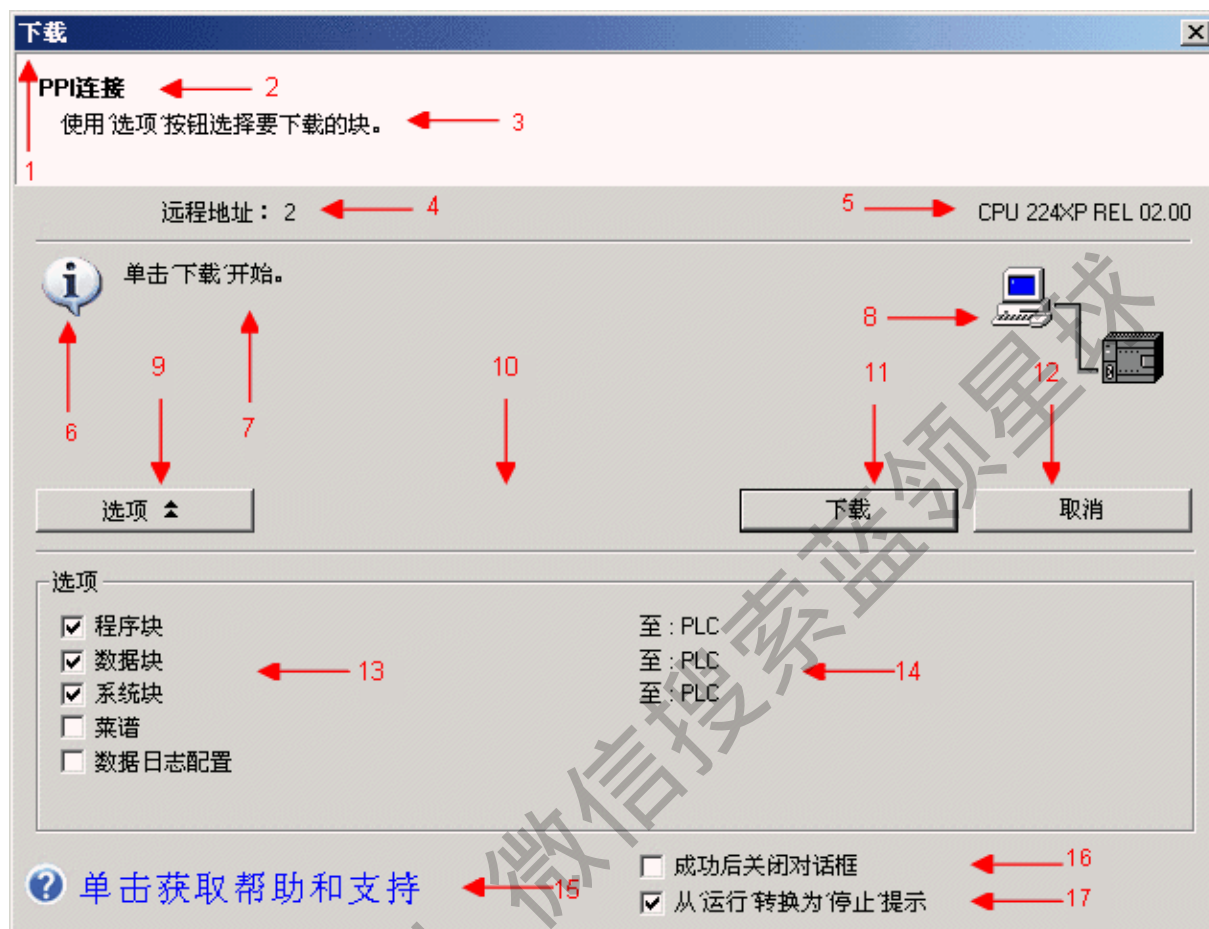




设置PLC类型 项目文件I/O错误 LAD/FBD编辑器错误 全部STEP 7-Micro/WIN编译器错误均列在输出窗口中。双击错误，编辑器滚动至错误位置。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译器。 **通讯错误** 欲下载（编辑器至PLC）或上载（PLC至编辑器），PLC通讯必须正常运行。确保网络硬件和PLC接头电缆均正常操作。PLC编译器确认PLC硬件支持全部程序指令、范围和结构。使用菜单命令 **PLC > 信息** 查看发现的第一条编译错误。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译程序，这样会将尽可能多的错误移至输出窗口，易于找到错误位置和改正错误。 **严重错误**和**非严重运行时间编程故障**由菜单命令 **PLC > 信息** 报告。

[打开项目](#) [范围检查](#)

UDM - 上载下载管理器详细说明：（用于上载、下载、清除及编程内存操作）



1. 标题 - 将进行的操作
2. 连接类型 - 通讯连接类型
3. 操作说明与指令 - 进程说明与指示
4. CPU地址 - 通讯地址
5. CPU类型 - 目标CPU的版本和类型
6. 状态图标 - 表明进程的总体状态
7. 状态讯息区 - 显示错误和警告的区域
8. 动画图标 - 表明操作处理是否在进行中
9. 选项按钮 - 基于用户的优先选择隐藏或显示选项选择
10. 更改项目/通讯按钮 - 更改项目的CPU类型以与实际CPU相符，也允许通讯配置
11. 动作按钮 - 如果没有错误存在就开始操作
12. 取消按钮 - 在下一个停止点取消正在进行的操作
13. Option selections - 选取选项
14. Option status? 选定的选项状态
15. 帮助 - 打开MicroWin帮助（与按F1键相同）
16. 成功操作后关闭对话框 - 用于在成功操作后关闭此窗口的选项
17. 从“运行”转换为“停止”提示 - 当CPU从运行模式转换为停止模式时省略用户提示的选项。

另请参阅：

[通讯概述 \(GS 6.1\)](#)

[如何测试通讯网络 \(GS 6.2\)](#)

[如何下载程序 \(GS 6.3\)](#)

[在运行模式中进行程序编辑](#)

[上载 \(文件 > 上载\)](#)

[如何从PLC上载 \(GS 8.3\)](#)

[STEP 7-Micro/WIN编译程序错误](#)

[如何纠正编译错误和下载错误 \(GS 6.4\)](#)



[应用程序用户参考手册](#)  
[入门指南目录](#)

### 3.2.29 使用状态监控确认运行时间性能

您成功地在运行的STEP 7-Micro/WIN编程设备和PLC之间建立通讯并向PLC下载程序后，则可利用“调试”工具条的诊断特征。可单击工具条按钮或从“调试”菜单列表选择项目，选择调试工具。

**调试工具条**



本标题讨论下列主题：

[什么是“状态”？](#)

[前提条件](#)

[收集状态的不同方法](#)

[以连续、瞬态图或触发暂停的方式收集状态](#)

[PLC模式](#)

[状态通讯与扫描循环](#)

[模拟程序条件（读取、写入、强制、取消强制）](#)

[检查交叉引用和元素用法](#)

[向程序下载改动](#)

[重设用于调试和编辑的帧设置](#)

[什么是“状态”？](#)

[返回顶端](#)

‘状态’这一术语指显示程序在PLC中执行时有关PLC数据当前值和功率流的信息。您可以使用状态图和程序状态窗口读取、写入和强制PLC数据值。

在控制程序的执行过程中，PLC数据的动态改变可用三种不同方式检视：

#### 图状态

在一表格中显示状态数据：每行指定一个要监视的PLC数据值。您指定一个内存地址、格式、当前值及新值（如果使用写入命令）。

#### 趋势显示

用随时间而变的PLC数据之绘图跟踪状态数据：

您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势数据亦可在趋势视图中直接赋值。

#### 程序状态

在程序编辑器窗口中显示状态数据：当前PLC数据值会显示在引用该数据的STL语句或LAD/FBD图形旁边。LAD图形也显示功率流

#### 程序状态和图状态 (或趋势视图) 窗口可以同时运行：

在状态图窗口写入或强制PLC数据将PLC数据改动应用于程序状态窗口。

在程序状态窗口写入或强制PLC数据还会将新PLC数据改动应用于状态图窗口。

#### 检视状态数据的前提条件

[返回顶端](#)

您必须执行下列任务，才能监控状态数据或调试程序：


- 成功地在STEP 7-Micro/WIN和PLC之间建立通讯。
- 成功地向PLC下载程序。
- 欲检视连续状态更新，PLC必须位于RUN（运行）模式。否则，您只能看到对I/O的改动（如果有）。由于PLC程序辉溢葱校 I/O的改动不会对“状态”显示中的程序逻辑产生预期的影响。
- 如果您位于程序不在执行的区域（例如，子例行程序、中断例行程序或由于JMP指令被绕过的区域），不会有状态显示，因为不在对代码执行扫描。

#### 注释：

- STEP 7-Micro/WIN中项目的时间标记和PLC必须匹配，才能激活状态数据通讯。时间标记比较（在您尝试打开状态时自动执行）繁 STEP 7-Micro/WIN中的项目状态表示正确反映实际PLC程序。如果时间标记不匹配，您可以检查“时间标记不匹配”对话框中南吕 斜恚 私饬男 纛斐刹黄 涸
- 下载程序后，您还应将PLC转换回RUN（运行）模式。否则会报告STOP（停止）模式状态数据，但PLC无法执行程序，因此您将无法看到预期的逻辑操作。

#### 在程序编辑器窗口中收集状态的不同方法

[返回顶端](#)

单击“程序状态打开/关闭”按钮，或选择菜单命令**调试 (Debug) > 程序状态 (Program Status)**，在程序编辑器窗口中显示PLC数据状态。状态数据收集按以前选择的模式开始。请参阅以下说明。

LAD和FBD程序有两种不同的程序状态数据收集模式。选择**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单命令会在状态模式复选标记打开和关闭之间切换。必须在程序状态操作开始之前选择状态模式。

- **执行状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选后）



- **扫描结束状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被取消勾选后）



状态图始终使用“扫描结束”状态数据。以前的 STEP 7 Micro/WIN 版本用“扫描结束”状态数据建立 LAD 和 FBD 程序状态视图。如果您使用的是第一代 PLC，执行状态不受支持。如果您尝试执行执行状态，“扫描结束”状态开始。

#### 执行状态色彩：

- 程序被扫描时，电源横杆会带有色彩。
- 图形中的功率流用色彩表示。
- 接点？接点打开时，指令会带有色彩。
- 线圈？输出打开时，指令会带有色彩。
- 方框和 SUBR 指令？指令接通电源并准确无误地成功执行时，方框和 SUBR 指令带有色彩。
- 绿色计时器和计数器表示计时器和计数器包含有效数据。
- 红色表示执行的指令有误。
- 跳接和标签指令现用时，显示为功率流色彩。如果为非现用，则显示为灰色。
- 灰色（灰色为默认赋值）表示无功率流、指令未扫描（跳过或未调用）或位于 STOP（停止）模式的 PLC。
- 布尔功率流位（仅限 FBD）。

LAD、FBD 和 STL 执行状态视图显示操作数数值，并表示每条指令在位于扫描循环执行程序阶段时的功率流。执行状态可以显示可能被执行后续程序指令盖写的中间数据值。所有显示的 PLC 数据值均从一个程序扫描循环收集。

#### 扫描结束状态

- 接通电源或逻辑真的接点和线圈显示为蓝色（您可以选择 **工具 (Tools) > 选项 (Options)** 菜单项目并单击“色彩”标记，指定自己选择的色彩）。

“扫描结束”状态显示在程序扫描结束时读取的状态结果。这些结果可能不会反映 PLC 数据地址的所有数值改动，因为随后某绿蜂鸣铃语毓绿蛭 杞嘤 翱瞻芊慈牒脍漆慈胧 怠 S 捐诳炯 PLC 扫描循环和相对慢速的 PLC 状态数据通讯之间存诤乃俛袂败稿 吧 杞嘤 弊刺 允炯父鑿 枢 方嘤 辈杉 氛 奠怠

#### STL 程序中程序状态举例：

当您开启 STL 中的状态时，程序编辑器窗口被分为一个代码区（左侧）和一个状态区（右侧）。可以根据您希望监控的数值类型

在 STL 状态中共有三个可监控的数值类别：

- **操作数** 每条指令最多可监控三个操作数。
- **逻辑堆栈** 最多可监控四个来自逻辑堆栈的最新数值。
- **指令状态位** 最多可监控十二个状态位。

**工具 (Tools) >**

**选项 (Options)** 对话框的 STL 状态标记允许您选择或取消选择任何此类数值类别。如果您选择一个项目，该项目不会在“状态”显示中出现。

	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+16095	
5	T32	位	2#0	
6	T32	带符号	+0	

SIMATIC STL

2 · 3 · 4 · 5 · 6 · 7 · 8 · 9 · 10 · 11 · 12 · 13 · 14 · 15 · 16 ·

		操作数 1	操作数 2	操作数 3	0123	空
LD	I0.0	ON			1000	1
A	SM0.5	ON			1000	1
LD	I1.0	OFF			0100	0
A	I1.1	OFF			0100	0
OLD					1000	1
LD	I2.0	OFF			0100	0
A	SM0.5	ON			0100	1
OLD					1000	1
LD	I0.2	ON			1100	1
A	SM0.5	ON			1100	1
LD	I1.2	OFF			0110	0
A	I1.3	OFF			0110	0
OLD					1100	1
ALD					1000	1
LPS					1100	1
MOVW	VW0, VW2	+15919	+15919		1100	1
AENO					1100	1
+I	VW0, VW2	+15919	+31838		1100	1
AENO					1100	1
=	Q0.0	ON			1100	1
LRD					1100	1
TON	T32, +32000	+288	+32000		1100	1
LRD					1100	1
TNCW	VW0	+15920			1100	1

MAIN | SBR\_0 | INT\_0 /

**LAD图形中执行状态举例：**

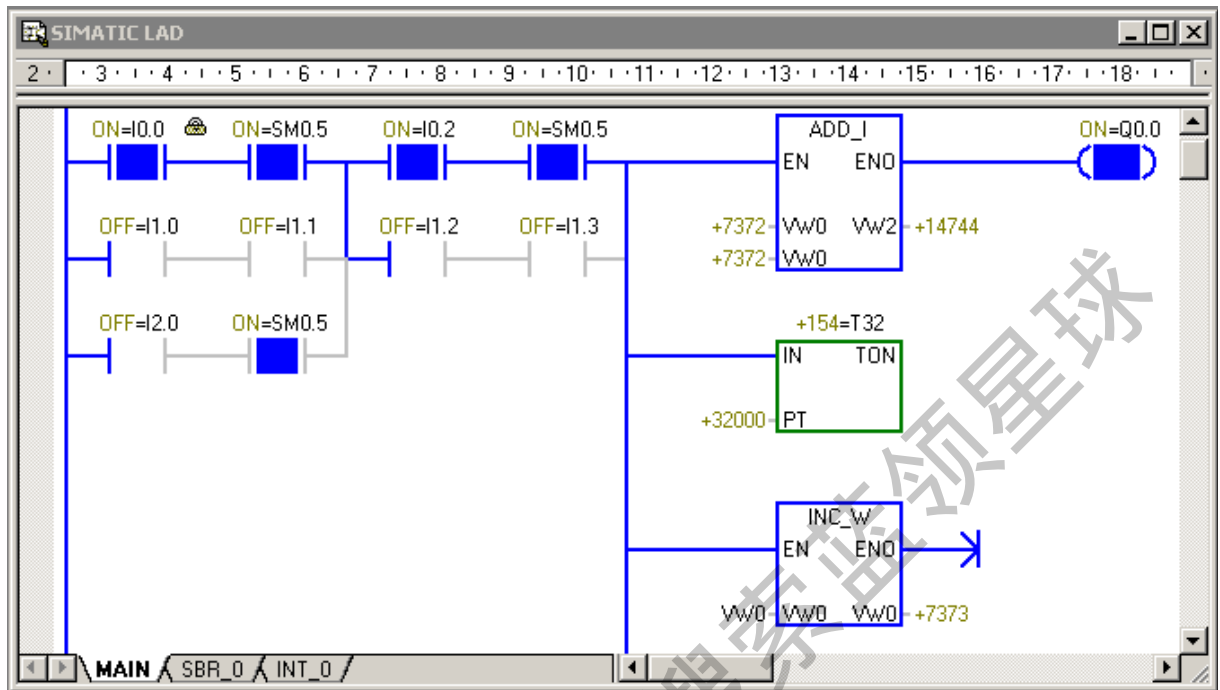
以下范例在LAD图形中使用LAD执行状态。状态图仅使用“扫描结束状态”数据收集。可在LAD图形和对应的状态图之间看出

状态图

3 · 4 · 5 · 6 · 7 · 8 · 9 · 10 · 11 · 12 · 13 · 14 · 15 · 16 · 17 · 18 · 19 · 20 ·

	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+7652	
5	T32	位	2#0	
6	T32	带符号	+0	

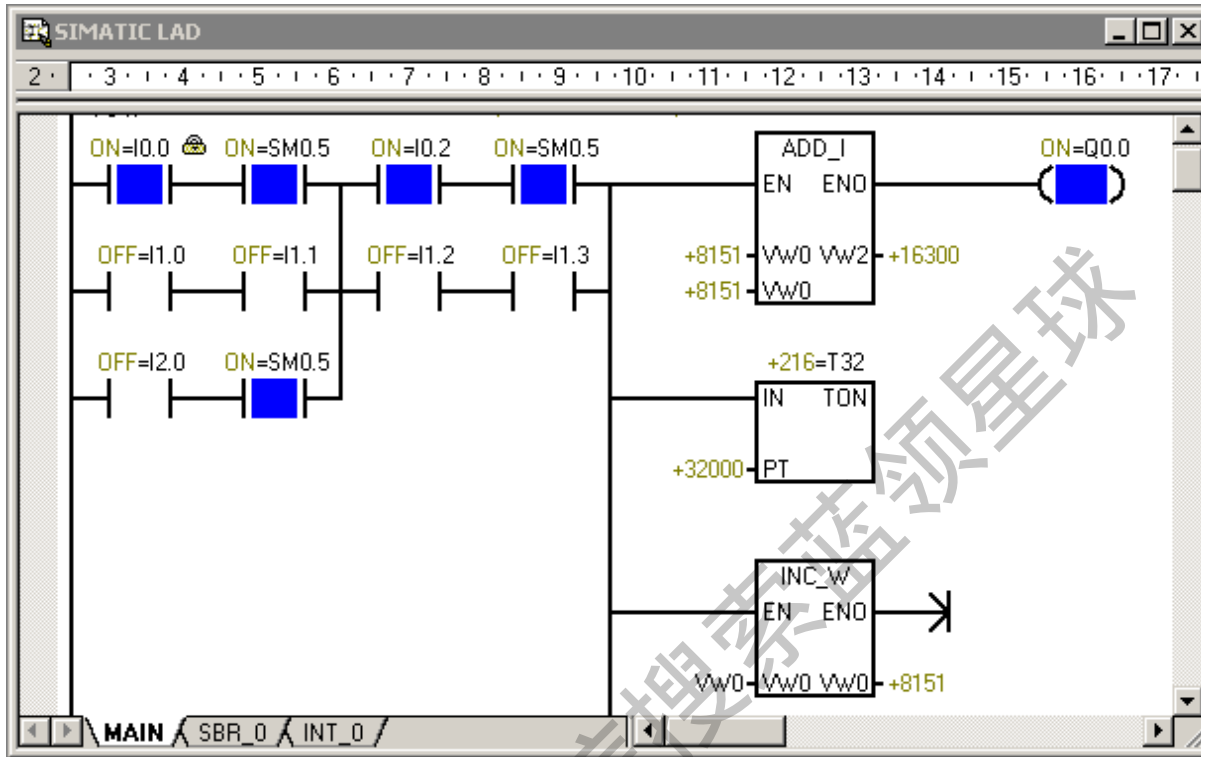
CHT1 /



#### LAD图形中扫描结束状态举例：

以下范例在LAD图形中使用“扫描结束状态”。由于两个窗口（程序状态和图状态）均使用相同的扫描 PLC结束数据，两个窗口允鞠嗤 PLC数据值。这种情形会引起误解，因为程序可以在收集最终“扫描结束”数值之前为相同的位置指定很多数值。中间临时数值不显示。在该范例中，ADD\_I方框未能正确执行，因为输入相加超出了带符号字整数的最大范围。与红色方框的执行状态范例不同，您必须读取状态值，才能看到 $Vw0 + Vw0 < VW2$ 。

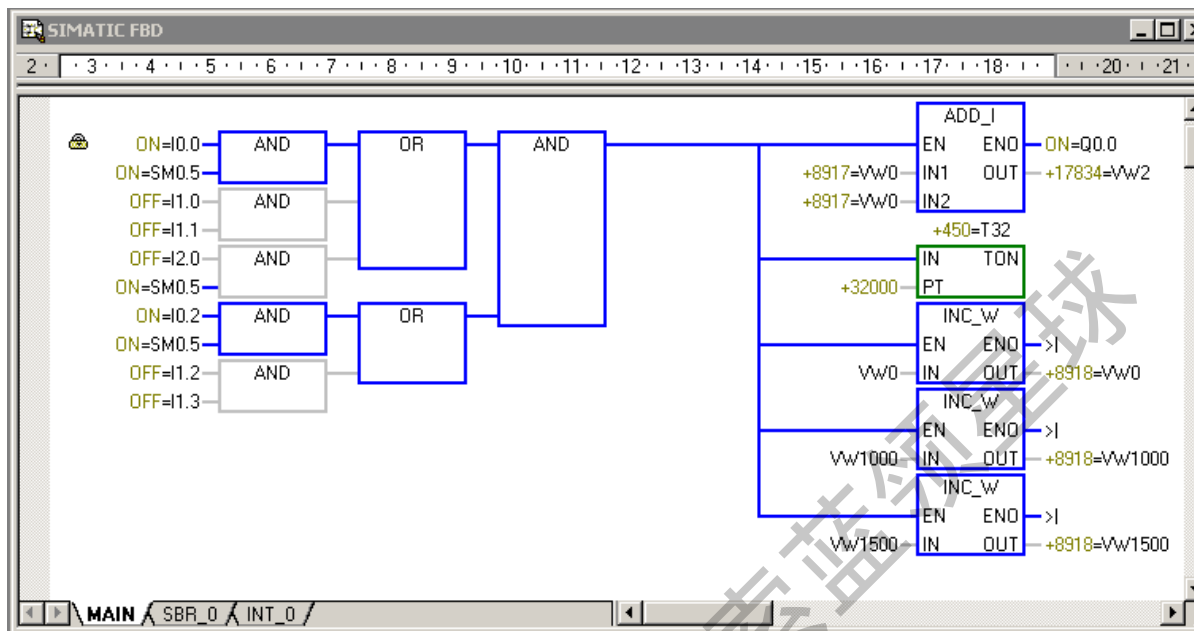
	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+8418	
5	T32	位	2#0	
6	T32	带符号	+485	



FBD图形中的执行状态举例:

地址	格式	当前值	新数值
1	I0.0	位	2#1
2	I0.2	位	2#1
3		带符号	
4	VW0	带符号	+8958
5	T32	位	2#0
6	T32	带符号	+0

获取更多资料



按照连续、瞬态图或触发（暂停程序状态）的方式收集状态值

[返回顶端](#)

#### 注释:

“连续”并非意味着实时；而是指编程设备不断地从 PLC 轮询状态信息，在屏幕中显示，并按按照通讯允许的速度更新显示。可能

#### 连续

- 打开程序编辑器窗口，并启动“程序状态”，当 PLC 位于 RUN（运行）模式时检视程序执行的连续状态更新。
- 打开状态图窗口，并启用“图状态”，当 PLC 位于 RUN（运行）模式时检视连续更新。



#### 瞬态图

- 将 PLC 转换为 STOP（停止）模式，收集单个状态更新。当 PLC 位于 STOP（停止）模式时，您可以使用“多次扫描”功能检视一次或多次扫描。您还可以引用网络中的特殊内存 SM0.1 首次扫描位，读取该网络的首次扫描状态值。
- 关闭图状态，如果您希望收集单次状态图更新并不希望将 PLC 转换至 STOP（停止）模式，使用“单次读取”功能。如 PLC 转换为 STOP（停止）模式，并开启“图状态”，您也会收集单次状态更新。此外，您可以在检视状态图时使用“多次扫描”和“首次扫描”功能。

#### 执行状态模式（LAD / FBD / STL）中的程序状态触发暂停

欲检视触发暂停 PLC 数据状态，您必须：

1. 将程序下载至 PLC，并将 PLC 设为 RUN（运行）模式。您还可以从 STOP（停止）模式开始触发暂停。如果您从 STOP（停止）模式开始，则可以捕获首次扫描。
2. 滚动程序编辑器窗口，显示您希望检视的程序网络。网络可以位于任何 POU（主程序、子例行程序或中断程序）中。
3. 使用 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单命令，将“使用执行状态”复选标记切换为打开。
4. 激活程序状态，开始状态数据通讯。



5. 单击“暂停程序状态” 工具条按钮，或用鼠标右键在程序编辑器中单击，并选择弹出菜单命令“暂停程序状态”。
6. 更新暂停之后，触发网络中的状态值在您检视时保持不变。当您准备移至另一个网络时，则单击“暂停程序状态” 工具条按钮，将暂停切换为关闭，并再次开始连续更新。

一旦获得新的状态信息，只要程序状态暂停功能被启用，该信息就会保持在屏幕上。在您取消选择程序状态暂停功能之前，无论 PLC 中实际变化速度多快，均不会进一步执行状态更新。如果没有程序状态暂停功能，屏幕就会不断刷新，状态数据在您读取和解释之前就可能消失（或者数据根本没有显示）。

#### PLC RUN / STOP（运行 / 停止）模式

[返回顶端](#)

使用以下一种方法改变PLC操作模式：

- 单击“运行”按钮，执行RUN（运行）模式，或者单击“停止”按钮，执行STOP（停止）模式。
- 选择**PLC > 运行 (RUN)** 菜单命令，执行RUN（运行）模式，或选择**PLC > 停止 (STOP)** 菜单命令，执行STOP（停止）模式。
- 以手动方式改变位于PLC上的模式开关。
- 在程序中插入一条STOP（停止）指令。

#### 注释：

欲使用STEP 7-Micro/WIN软件控制RUN / STOP（运行 / 停止）模式，必须在STEP 7-Micro/WIN和PLC之间存在一条通讯路径。此外，必须将PLC硬件模式开关设为TERM（终端）或RUN（运行）。将模式开关设为TERM（终端）不会改变PLC操作模式，但却允许STEP 7-Micro/WIN改变PLC操作模式。位于PLC前方的状态LED表示当前操作模式。当程序状态或状态图操作正在进行时，在STEP 7-Micro/WIN窗口右下方处附近的状态条上会出现一个RUN / STOP（运行 / 停止）指示灯。

虽然程序在STOP（停止）模式中不执行，PLC操作系统继续监控PLC（收集PLC RAM和I/O状态），将状态数据传递给STEP 7-Micro/WIN，并执行所有的“强制”或“取消强制”命令。当PLC位于STOP（停止）模式中时，您可以执行以下操作：

- 使用图状态或程序状态检视操作数的当前值。（此一步骤具有与执行“单次读取”相同的效果，因为程序未执行。）
- 您可以使用图状态或程序状态强制数值。使用图状态写入数值。
- 写入或强制输出。
- 执行有限次数扫描，并通过状态图和 / 或项目状态检视效果。

当PLC位于RUN（运行）模式时，您不得使用“首次扫描”或“多次扫描”功能。您可以在状态图中写入和强制数值，或使用LAD或FBD程序编辑器强制数值，方法与您在STOP（停止）模式中强制数值相同。您还可以执行以下操作（不得从STOP（停止）模式）：

- 使用图状态收集不断变化的PLC数据值的连续更新信息。（如果您希望使用单次更新，图状态必须关闭，才能使用“单次读取”命令。）
- 使用程序状态收集不断变化的PLC数据值的连续更新信息。
- 使用RUN（运行）模式功能中的“程序编辑”编辑程序，并将改动下载至PLC。

#### 状态通讯与扫描循环

[返回顶端](#)

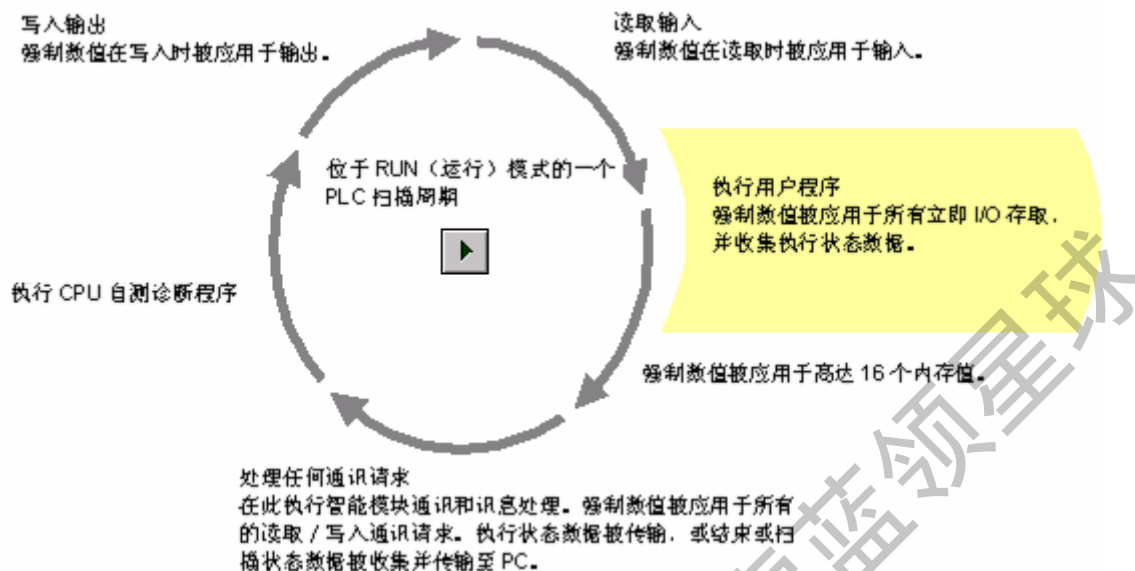
PLC在连续循环中读取输入、执行程序逻辑、写入输出和执行系统操作和通讯。该扫描循环速度极快，每秒执行多次。虽然STEP 7-Micro/WIN会快速发出状态请求，您应当认识到您检视的不是PLC中出现的每一个事件，这一点十分重要。

由于PLC和您观察的程序状态编程设备之间存在通讯时间滞后，您看到的显示的操作数数值总是在状态显示中改变之前即在PLC中改变，但没有在程序状态表示中改变。

如果您使用“扫描结束”状态模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被取消勾选时），您在几个扫描循环中采集数据。

如果您使用“执行状态”模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选时），所有显示的程序状态值一定来自同一个扫描循环。





**注释:**

一次只有一个状态窗口被授权运行状态。如果您尝试在任何其他与同一个 PLC 连接的 STEP 7 Micro/WIN 应用程序中运行状态，则会收到错误讯息。

**模拟进程条件 (读取、写入、强制、取消强制)**


[返回顶端](#)

您可以模拟进程条件，方法是在程序状态操作过程中从程序编辑器和从状态图向操作数写入或强制新数值。

使用“程序状态”和“图状态”按钮或“调试”菜单命令，开始状态数据通讯，并启用调试工具。您还可以使用硬件输入模拟器开关（西门子（Siemens）可提供）以手动方式在数字输入打开和关闭之间切换，并观察 PLC 上的 LED 指示灯，观察数字输出的条件。

从“调试”工具条或“调试”菜单列表存取以下功能。






**单次读取**（仅限状态图）如果您希望获得一次“瞬态图”（对状态图中的所有数值一次更新），使用“单次读取”。默认值为状态图连续轮询 PLC，获取状态更新信息。当您单击“状态图”时，状态图会切换为关闭，“单次读取” 卑磁ヶ黄黎漫




**全部写入**（仅限状态图）

在您完成对几行的“新数值”列的改动后，您可以使用“全部写入”，将所有需要的改动发送至 PLC。



**强制** 您可以在操作程序状态时从程序编辑器和状态图强制地址。欲强制程序状态地址，用鼠标右键单击一个参数，并选择从菜单列表强制。欲强制状态图地址为某一数值，您必须首先规定所需的数值，方法是读取该数值（如果您希望强制当前值）或键入该数值（如果您希望将地址强制为一个新数值）。您一旦使用了强制功能，则在每次扫描时该数值均被重新应用于地址，直至您取消强制地址。“强制”功能盖写立即读取或立即写入指令。“强制”功能还盖写被配置为在转换为 STOP（停止）模式时进入一个指定数值的输出：如果 CPU 进入 STOP（停止）模式，输出被设为强制数值，而非非配置的数值。通过将 V 或 M 内存强制为字节、字或双字，将 AI 或 AQ 内存在偶数字节边界上（例如 AIW6 或 AIW14）或 I/O 点上（I 和 Q 位址）强制为字，模拟逻辑条件。您可以一次强制 16 个（V、M、AI 或 AQ）地址和所有的 I/O 位（所有 I 和 Q 位址）。 该图标表示该地址被明确强制。该地址数值在地址被取消强制之前无法改变。 该图标表示该地址被隐含强制。如果地址是一个被明确强制的较大地址的一部分，该地址则被认为是隐含强制。例如，如果 VW0 被强制，则 VB0 是隐含强制（VB0 是 VW0 的第一个字节）。隐含强制数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制 VD0（该地址包含 VB0、VB1、VB2 和 VB3），则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制，则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。



**部分强制** 该图标表示该地址的一部分被部分强制。例如，如果 VW0 被明确强制，则 VW1 的一部分被强制（VW1 的第一个字节是 VW0 的第二个字节）。被部分强制的数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制 VD0（该地址包含 VB0、VB1、VB2 和 VB3），则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制，则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。



**取消强制** 对于程序状态和状态图，选择一个地址，并使用“取消强制”按钮从该特定地址移除

强制功能。您还可以选择（单击）一个参数，然后用鼠标右键单击该参数，查看“强制”和“取消强制”功能的弹出菜单。



#### 全部取消强制

如果您希望从所有的地址中移除强制功能，使用“全部取消强制”按钮。在应用“全部取消强制”之前，您无须选择单个地址。



#### 读取全部强制（仅限状态图）

当您使用“读取全部强制”功能时，状态图的“当前数值”列会为已经明确强制、隐含强制或部分隐含强制的所有地址显示一个图标。

### 检查交叉引用和元素用法

[返回顶端](#)

调试程序时，您可以决定是否增加、删除或编辑参数。

使用“交叉引用”窗口查看程序中当前指定参数的方法，这样可以防止您意外地重复赋值。

此外，如果您希望在RUN（运行）模式中执行程序编辑，并且某些网络使用正向或负向转换（EP和N接点或方框以及STL EU或ED指令），您一定要检查“交叉引用”信息中的EU/ED数目，以防重复指令号码。

### 向程序下载改动

[返回顶端](#)

一旦调试部分代码，如果PLC位于STOP（停止）模式，您可以编辑程序、下载程序、然后再次监控状态，查看改动是否发生作用。

您不必进入STOP（停止）模式就可以在RUN（运行）模式中执行程序编辑并向PLC下载较小的改动。



**警告** 在RUN（运行）模式中向PLC下载改动时，您的改动会立即影响程序操作。不得出错；程序编辑中的错误可能导致人员死亡或严重伤害和/或设备损坏。仅限合格人员在RUN（运行）模式中执行程序编辑。

### 重设用于调试和编辑的帧设置

[返回顶端](#)

欲重设将窗口和工具条的可视性、尺寸和位置设为默认值，选择**检视（View）> 帧（Frame）> 全部重设（Reset All）**菜单命令。

您有两套不同的工作空间窗口的便利，可在调试程序时获得较大的灵活性：窗口可视性和定位。这是比正常编辑空间更大的检视空间。当您进入程序状态时，您已经设置的任何窗口定位保持初始进入的状态；如果您关闭程序状态，窗口保持初始进入的状态。同样，任何在编辑程序时执行的窗口定位均在运行时与程序状态分开维护。

[返回顶端](#)

#### 另请参阅：

[如何在程序编辑器中显示状态（GS 7.2）](#)

[如何在状态图中显示状态（GS 7.3）](#)

[如何执行有限数目扫描（GS 7.4）](#)

[如何下载程序（GS 6.3）](#)

[时间标记不匹配错误](#)（确保编程设备中的项目与PLC中的项目相匹配）

[交叉引用和元素用法](#)（确保程序编辑不引起重复赋值）

[在运行模式中执行程序编辑](#)

[PLC运行/停止模式](#)


[在停止模式中写入和强制输出](#)

[安装和移除S7-200设备](#)

[接地和布线指南](#)

## 3.2.30 保存作业



您可以使用工具条上的“保存”按钮保存作业，或从“文件”菜单选择“保存”和“另存为”选项保存作业。



- “保存”允许您在作业中快速保存所有改动。（然而，您初次保存项目时，会被提示核实或修改当前项目名称和目录哪 诞之唯#
- “另存为”允许您修改当前项目的名称和 / 或目录位置。

当您首次建立项目时，STEP 7-

Micro/WIN的默认值名称为“Project1.mwp”。您可以接受或修改该名称；如果您接受该名称，下一个项目的默认名称将自动递增为“Project2.mwp”。

STEP 7-Micro/WIN项目的默认目录位置是位于“Microwin”目录中的称作“项目”的文件夹，您可以不接受该默认位置。

### 3.3 语句表编程

#### 3.3.1 SIMATIC STL指令

白... 指令





### 3.3.2 语句列表程序举例

下面列举的程序显示根据 [如何在STL中输入语句](#) 中的指南在STL编辑器中制定格式。您可能希望以类似的方式设定程序，由网络注解对网络的功能作出一般概述，由语句注解识别语句的功能。请注意示范程序使用了符号地址。

如果您希望在STEP 7-Micro/WIN中看到该程序（用STL或LAD），选择**文件 > 打开**，浏览至Microwin\Samples\English目录，增亮示范程序，并单击“打开”按钮。

#### 油漆混合器示范程序

```

NETWORK 1 //用1号成分装满储罐并监视储罐水平。
//如果“起始”开关打开，保持泵原有状态。
LD Start_1 //载入Start_1 I0.0数值。
O Pump_1 //或带有Pump_1 Q0.0数值。
A Stop_1 //并且带有Stop_1 I0.2。
AN High_Level //并且不带有High_Level I0.4。
= Pump_1 //将结果指定给Pump_1 Q0.0。
NETWORK 2 //用油漆成分2装满储罐并监视储罐。
//位置。如果“起始”开关打开，保持泵原有状态。
LD Start_2 //载入Start_2 I0.1数值。
O Pump_2 //或带有Pump_2 Q0.1数值。
A Stop_2 //并且带有Stop_2 I0.3。
AN High_Level //并且不带有High_Level I0.4。
= Pump_2 //将结果指定给Pump_2 Q0.1。
NETWORK 3 //如果达到高位，设定内存位。
LD High_Level //载入High_Level I0.4数值。
S High_Lev_Reached, 1 //将High_Lev_Reached M0.1设定为1（打开）。
NETWORK 4 //如果达到高位，设定起始定时器。
LD High_Lev_Reached //载入High_Level Reached M0.1数值。
TON Mix_Timer, +100 //定时器37，预设 = 100 (@0.1 s)
NETWORK 5 //打开混合器马达。
LDN Mix_Timer //不载入Mix_Timer T37数值。
A High_Lev_Reached //并且带有High_Lev_Reached M0.1。
= Mixer_Motor //将结果指定给Mixer_Motor Q0.2。
= Steam_Valve //将结果指定给Steam_Valve Q0.3。
NETWORK 6 //排空油漆混合储罐。
LD Mix_Timer //载入Mix_Timer T37数值。
AN Low_Level //并且不带有Low_Level I0.5。
= Drain_Valve //将结果指定给Drain_Valve Q0.4。
= Drain_Pump //将结果指定给Drain_Pump Q0.5。
NETWORK 7 //为每次循环计数。
LD Low_Level //载入Low_Level I0.5数值。
A Mix_Timer //以及Mix_Timer T37数值。
LD Reset //载入Reset I0.7数值。
CTU Cycle_Counter, +12 //计数器30，预设 = 12。
NETWORK 8 //如果达到“低位”，重置内存位，计时器超时。

```

```
LD      Low_Level           //载入Low_Level I0.5数值。
A       Mix_Timer           //以及Mix_Timer T37数值。
R       High_Lev_Reached, 1 //将High_Lev_Reached M0.1数值重设为0。
```

### 3.3.3 编辑指令参数

#### 步骤:

您可以在程序编辑器窗口中键入STL指令或用从指令树拖放的方式插入STL指令。

欲在STL编辑器中键入语句:

1. 将光标放在一行的开始。
2. 输入指令。(例如: LD)
3. 按SPACEBAR (空格键) 或TAB (制表符) 键。
4. 输入操作数, 操作数可以是绝对值 (例如: I0.0)、符号 (例如: Input1) 或常量 (例如: 100)。

带指令和地址的整行范例: LD I0.0)

**注释:** 如果一行有多个操作数, 您可以使用空格、制表符或逗号将其分开。

5. 如果无须注解, 按ENTER键, 移至下一行。

请勿使用MEND、RET、RETI、SBR或INT终止程序的POU (OB1、子程序、中断例行程序)。这些指令由编译程序自动插入蝗纒 POU中使用这些指令, 编译程序返回一则错误。

欲从指令树拖放:

1. 选择指令。



2. 将指令拖拉至所需的位置。



3. 松开鼠标按钮, 将指令放在所需的位置。



晏馍 或另一条指令的参数上)。

**注释:** 光标会自动阻止您将指令放在非法的位置 (例如, 放在网络

如果您将指令放在一行上, 该行会向下移动, 将新指令放在该行原来的位置。

#### 指南:

- 每行仅输入一个指令和 / 或注解。
- 按INSERT (插入) 键, 在键入时在“插入模式”和“覆盖模式”之间切换。“插入模式”是默认值。
- 欲分隔程序中的网络, 您可以使用位于构成网络的指令开始前的一行的关键字 NETWORK (网络)。(请勿键入网络怕耄坏蹦 喊朊被磷远 龟雍怕霁)

举例:

```
Network 1 //该行介绍程序的第一个网络
LD      I0.0 //在I0.0载入数值
O       I0.1 //或带有I0.1数值
//此处输入更多指令: 每个网络由多个指令组成
```

#### Network 2 //开始新的指令网络

分隔网络为供选用选项；该选项使查看配备 LAD 或 FBD 程序编辑器的 STL 程序更加简便。

- 当程序超过一个屏蔽时，使用滚动条或 PAGE UP/DOWN（上页 / 下页）键查看程序。

#### 另请参阅：

[编址简介 \(GS 2.2\)](#)

[如何在 STL 中输入程序注解 \(GS 5.3\)](#)

[语句列表程序范例 \(GS 5.4\)](#)

### 3.3.4 使用符号表帮助组织程序

使用下列方法之一打开符号表（用 SIMATIC 模式）或全局变量表（用 IEC 1131-3 模式）：



- 单击浏览条中的“符号表”按钮。
- 选择**检视 (View) > 符号表 (Symbol Table)** 菜单命令。
- 打开**指令树**中的符号表或全局变量文件夹，然后双击一个表格图标。

您可以使用项目中的**系统符号名**。预定义的系统符号表提供常用 PLC 系统功能的存取功能。PLC 系统符号将功能名称与用于激活霉 δ 芳 PLC 特殊内存位置相联系。

本标题讨论下列信息：

[在表格中编辑](#)

[在表格中指定符号赋值](#)

[引号的使用](#)

[插入附加行](#)

[使用定义、编辑或选择符号命令](#)

[按照列排序](#)

[检视重叠和未使用的符号](#)

[使用多个表格](#)

[建立附加符号表](#)

[在符号表之间移动](#)

[POU 符号标签](#)

[符号和绝对视图](#)

[在符号和绝对地址视图之间切换](#)

[同时检视符号和绝对地址](#)

[理解符号](#)

[理解符号范围](#)

[使用间接记号](#)

[内存类型](#)

[输入错误和故障排除](#)

[提示](#)

[SIMATIC 与 IEC](#)

[SIMATIC 和 IEC 表格举例](#)

[在符号表 / 全局变量表中指定符号赋值](#)

[返回顶端](#)

欲为地址或常量值指定符号，请遵循下列步骤：

1. 打开符号表 / 全局变量表（使用标题开始部分描述的一个步骤）。
2. 在“符号名”列键入符号名（例如，Input1）。允许使用的最大符号长度为 23 个字符。使用 TAB、ENTER 或 ARROW 键确认，并移至下一个单元格。

**注释：**

\*

在为符号指定地址或常量值之前，该符号一直显示为未定义符号（绿色波浪下划线）。完成“地址”列赋值后，绿色波浪下划线被删除。

\*

如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在 LAD、FBD 和 STL 程序编辑器窗口中被一个波浪号 (~) 截断。您可将鼠标放在被截断的名称上，在工具提示中查看全名。

3. 在“地址”列中键入地址或数值（例如，V 或 I23）。  
（在 IEC 1131-3 编程模式中，键入地址后会自动增加正确的 IEC “%” 前缀）。
4. 如果您正在使用 IEC 全局变量表，在“数据类型”列的下拉列表中选择一数据类型。  
（SIMATIC 用户无需提供数据类型。）
5. 键入注解（选项：最多允许 79 个字符）。

**注释:**


在STEP 7-Micro/WIN中, 您可以建立多个符号表 (SIMATIC编程模式) 或多个全局变量表 (IEC 1131-3编程模式)。但不允许将相同的符号名称多次用作全局符号赋值, 在单个符号表中中和几个表内均不得如此。(相反, 可允许根据您的选择在多个不同局部变量表中多次使用相同的符号名称。)



**检视重叠和未使用的符号**


[返回顶端](#)



欲检视符号表中的“重叠”列或“未使用的符号”列:

- 选择工具 (**T**ools) > 选项 (**O**ptions) 菜单项目。
- 选择“符号表”标记。
- 选择适当的复选框 (“显示重叠符号”和“显示未使用的符号”)。

“重叠”列显示绝对地址共享部分或全部相同内存之符号行的重叠图标 。如果同一个字面值有多个已定义的符号常量, 那该表格被修改时, “重叠”列被更新。

符号表			
· 3 · 1 · 4 · 1 · 5 · 1 · 6 · 1 · 7 · 1 · 8 · 1			
		符号	地址
1		符号3	VBO
2		符号4	VDO

“未使用的符号”列就在您的程序中未被引用的所有符号显示未用图标 。每次表格被修改时, 该列被更新。

符号表			
· 3 · 1 · 4 · 1 · 5 · 1 · 6 · 1 · 7 · 1 · 8 · 1			
		符号	地址
1		符号1	I0.0
2		符号2	I0.1

**引号的使用**

[返回顶端](#)

在3.1版之前, 全局符号名称必须放在双引号内。双引号已不再用于符号名。自4.0版起, 双引号用于在符号表中起始和结束作为撤 豈 持档册ASCII常量字符串。此赋值创建了一个符号ASCII字符串常量。

与此相似, 单引号用于起始和结束ASCII字符常量 (只限字节、字、双字) 以创建一个符号ASCII字符常量。

**插入附加行**

[返回顶端](#)

使用下列方法之一在符号表 / 全局变量表中插入附加行:

- 选择菜单命令 **编辑 (E)dit** > **插入 (I)nsert** > **行 (R)ow**。将在符号表 / 全局变量表光标的当前位置上方插入新行。
- 用鼠标右键单击符号表 / 全局变量表中的一个单元格。选择弹出菜单命令 **插入 (I)nsert** > **行 (R)ow**。将在光标的当前位置上方插入新行。
- 欲在符号表底部插入新行, 将光标放在最后一行的任意一个单元格中, 按“下箭头”键。

**使用定义、编辑或选择符号命令**

[返回顶端](#)

定义、编辑或选择符号等命令, 允许您在使用程序编辑器或状态图时, 定义新符号、从列表上选取现有符号、或编辑符号属性。新的或修改后的赋值将被自动加入到符号表内。

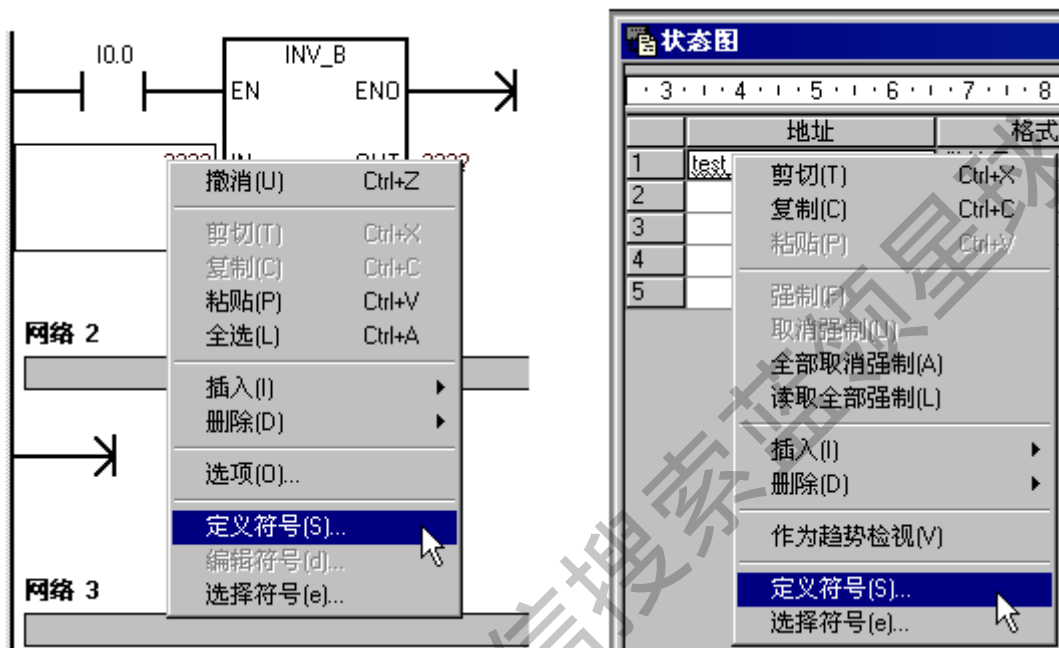


**提示:** 只有在“符号编址”视图中工作时才能使用符号命令。(检查“检视”菜单下方)

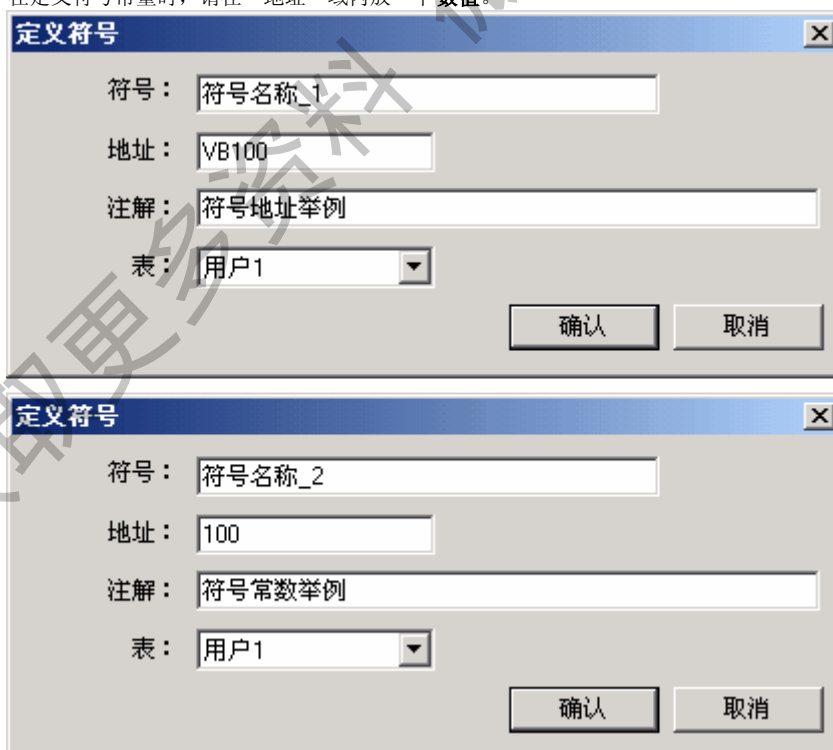
的“符号编址”选项：标选符号表示已打开“符号编址”视图。)

欲激活“符号”命令：

1. 用鼠标右键单击对某指令参数或“状态图”地址单元来说不完全（或不正确）的符号名，然后从弹出菜单选择“定义符号(S)...”。



2. 定义新符号，编辑现有符号的属性，或由全部用户指定符号的概括中选取一个现有符号。  
在定义符号地址时，请在“地址”域内放一个地址。  
在定义符号常量时，请在“地址”域内放一个数值。



bml mw\_sym\_define\_b.gif)



3. 单击“确定”确认作业，并关闭对话框，或单击“取消”，取消选择。



**注释：**

- 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在 LAD、FBD 和 STL 程序编辑器窗口中被一个波浪号 (~) 截断。您可将鼠标放在被截断的名称上，从而在工具提示中查看全名。

**按照列排序**[返回顶端](#)

您可按照“名称”或“地址”列排序表格，排序时可按正向或逆向（字母）顺序排列。此表中的符号常量将在与地址关联的最后一个符号后面列出。所有非字符类型的符号常量接着将独立于显示格式地按数字顺序排列。ASCII 常量将出现在数字常量之后，地址 A 拷 罾蟪毓帧。

您可以使用检视菜单对某列进行排序。请单击列身以选取该列，请勿单击列顶的列标题（“名称”或“地址”）。

- 若按正向顺序（A 至 Z）排序列，单击“排序”按钮，或选择菜单命令**检视 (View) > 正向排序 (Sort Ascending)**。
- 若按逆向（Z 至 A）顺序排序列，单击“逆向排序”按钮，或选择菜单命令**检视 (View) > 逆向排序 (Sort Descending)**。


若要按相反顺序排序，单击列顶部的列头（“名称”或“地址”）。当您单击列头时，将在正向和逆向排序之间切换。

**建立附加符号表**[返回顶端](#)

默认情况下，符号表窗口为用户定义的符号名称（USR1）显示一个标记。如果您对任何 POU 重新命名，符号表窗口显示一个被称为“POU 符号”的标记—如果您在使用“协议”功能，例如 USS 协议库，则窗口显示“USS 协议”标记。

可采用下列几种方法为用户定义的符号名称建立附加符号表。

- 从指令树用鼠标右键单击“符号表”文件夹，并选择弹出菜单命令**插入符号表 (Insert Symbol Table)**。
- 打开符号表窗口，并使用“编辑”菜单，或用鼠标右键单击，调出弹出菜单，并选择**插入 (Insert) > 表格 (Table)**。
- 您在建立程序网络时，可指定符号名，方法是在程序编辑器的指令参数域中键入有效的符号名。此种方法建议一组不


付 洼嫫 刈返姆 琶 5 七 鳌敖 (0) 锤持捣 疔恹卑磁 ，将该组符号名传送至您指定地址的新“符号表”标记。欲为符号表重新命名，用鼠标右键单击指令树中的符号表标记，并选择“重新命名”，或双击“符号表”标记名。

**注释：**成功地插入新符号表后，一个新标记

会出现在符号表窗口的底部。在检视符号表时，请核实选择了正确的标记。您可以双击或用鼠标右键单击标记，为标记重新命名。

**在符号表之间移动**[返回顶端](#)

如果建立了多个符号表，您可以使用下列任意一种方法存取表格：

- 如果已经打开符号表窗口，单击位于符号表窗口底部的标记 ，在符号表之间移动。
- 从指令树扩充“符号表”文件夹图标，并双击所需的“符号表”图标，检视符号表窗口中该表的标记。

**POU 符号标记**[返回顶端](#)

如果您已经为项目的任何 POU 或其他元件（例如数据块、状态图或符号表）指定符号名，将在符号表窗口的“POU 符号”标记中列出符号名赋值。

该标记为只读标记；您不能在此编辑赋值。

如果您希望更改赋值，您必须编辑该元件的“属性”对话框。（从指令树的“项目”分支，用鼠标右键单击元件，调出“属性”对话框）

所有符号赋值必须使用有效语法。（请参阅下面的[输入错误](#)）。如果指定符号名赋值时违反了指南规定，编译程序时 STEP 7-Micro/WIN 会报告错误。

### 在符号和绝对地址视图之间切换

[返回顶端](#)

在符号表 / 全局变量表中建立符号和绝对地址或常量值的关联后，您可在操作数信息的符号和绝对显示之间切换。可采用下列

- 选择菜单命令 **检视 (View) > 符号编址 (Symbolic Addressing)**，在符号编址打开或关闭之间切换
- 使用 **Ctrl+Y** 快捷键在符号编址打开或关闭之间切换

“符号编址”菜单项目前面的标选标记表示已打开符号编址。默认条件下，当您打开第一个项目时，符号编址也被打开。

您不能在检视符号常量或其关联的字面值之间切换。这是因为您可以为同一个字面值定义多个符号常量。因此，假如您可以

STEP 7-Micro/WIN 无法可靠地恢复原有的符号常量。出于此原因，禁用符号编址（通过主菜单或按 CTRL+Y）将不会影响您项目中符号常量的显示。出于同样原因，如果您为某操作数输入了该常量的字面值，STEP 7-Micro/WIN 将不会自动套用已定义的符号常量。

### 同时检视符号和绝对地址

[返回顶端](#)

欲在 LAD、FBD 或 STL 程序中同时检视符号地址和绝对地址，使用菜单命令 **工具 (Tools) > 选项 (Options)**，并选择“程序编辑器”标记。选择“显示符号和地址”。



**提示：** • 只有在打开 **符号地址视图** 时，才能在项目中显示符号地址。否则，即使您选择了“显示符号和地址”，也只会显示绝对地址。

- 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在 LAD、FBD 和 STL 程序编辑器窗口中被一个波浪号 (~) 截断。您可将鼠标放在被截断的名称上，在工具提示中查看全名。

### 理解符号范围

[返回顶端](#)

符号表 / 全局变量表将符号名指定给 PLC 内存或 I/O 地址。在符号表 / 全局变量表中定义 **符号** 时，符号具有全局范围。这意味着您可在任何 **POU** 中使用该符号名称，代表位于该符号地址的数据。（相反，如果您使用局部变量表指定符号名称，则只能在定义该局部变量的 POU 范围内使用该局部变量。）

### 使用间接记号

[返回顶端](#)

与绝对地址相同，您可在符号中使用间接记号 (& 和 \*)。欲了解有关间接编址的详情，请参阅 [直接和间接编址](#)。

### 内存类型

[返回顶端](#)

您可为下列内存类型建立符号名：I、Q、M、SM、AI、AQ、V、S、C、T、HC

### 输入错误

[返回顶端](#)

输入错误用色彩显示，很容易发现：

- 非法语法 — **红色文字** 和 **红色波浪下划线**

例如：

	符号	地址
1	<b>2name</b>	I0.0
2		<b>VBO0</b>
3	<b>Begin</b>	I0.2

(第一个字符的号码字符非法)

(非法地址 VBO 是正确的) (逆转的关键字无法用作符号)

- 非法使用 — **红色波浪下划线**

例如：

	符号	地址
1	Pump1	I0.0
2	Pump1	I0.0
3	SymConstant	123456
4	SymConstant	123456
5	SymConstant2	654321
6	SymConstant3	654321

(重复名称) (重复地址)

- 未定义符号 — 绿色波浪下划线

例如:

	符号	地址
1	Pump1	
2	Pump2	V0
3	Factor	3.14E-006
4	Neg_Value	-3333333

(名称带有无效地址) SIMATIC模式

(名称带有无效地址或数据类型) IEC模式

**注释:**

- 符号名称可包含某些字母数字字符和下划线，以及某些扩展字符（ASCII 128至ASCII 255）。但第一个字符只能是字母或扩展字符。
- 在3.1版之前，全局符号名称必须放在双引号中。双引号已不再用于符号名。自4.0版起，双引号用于在符号表中起始和结束作为某符号名之赋值的ASCII常量字符串。此赋值创建了一个符号ASCII字符串常量。与此相似，单引号用于起始和结束ASCII字符常量（仅限字节、字、双字）以创建一个符号ASCII字符常量。
- 不允许使用关键字作为符号名，也不允许第一个字符是数字或包含非字母数字字符或扩展字符集。
- 符号名称的最大许可长度为23个字符。
- 如果您希尝试建立符号赋值，以纠正程序中操作数“未定义符号”错误，但键入地址数值后忘记按TAB键、ENTER键  
蚣 芳 蹦 禱 爻 繇 蚩 唵 魔 保 僮 魔 廊 幌 允 尽 拔 炊 丁 宸 拧 贝 砦 蟆 N 私 伎 稣 庖 晃 侍 猓 禱 胤 疝 棠 翱 冢 獾 支 旁 淞 暗 司 贰 钡 卮 襪 校 TAB、ENTER或箭头键，完成地址赋值。

**建立符号提示**[返回顶端](#)

- 您可在任何时间指定符号赋值：在建立程序逻辑之前、之间或之后。
- 您可在“注解”列中增加注释，记录您的符号。
- 编程时，可使用“定义、编辑或选择符号”命令从列表选择新符号名或建立新符号名。
- 通过单击不同的表列，可对符号表/全局变量表中的数值进行检索。
- 通过拖拉边缘，可重定列尺寸。
- 您可使用文件(File) > 打印(Print)命令，并选择适当的复选框，打印符号表/全局变量表。
- 您一旦建立符号赋值，就可以通过启用符号信息表功能，在LAD/FBD/STL程序编辑器中的每隔网络滞后显示地址赋值和符号注解。

**SIMATIC和IEC表举例**[返回顶端](#)

符号表（SIMATIC编程模式）和全局变量表（IEC编程模式）在功能上的唯一区别是“数据类型”列，此为IEC编程的独特功能。

IEC-

1131是一种加强类型语言。指令操作数被定义为接受一种指定的数据类型。欲了解有关数据类型的详情，请参阅 [SIMATIC和IEC 1131-3数据类型](#)。

SIMATIC符号表举例:

符号表				
· 3 · · 4 · · 5 · · 6 · · 7 · · 8 · · 9 · · 10 · · 11 · · 12 · · 13 · · 14 · · 15 · · 16 ·				
		符号	地址	注解
1		动1	I0.0	
2		动2	I0.1	
3		停止1	I0.2	
4		停止2	I0.3	
5		高等级	I0.4	
6		低等级	I0.5	
7		复位	I0.7	
8				
9		泵1	Q0.0	
10		泵2	Q0.1	
11		搅拌机电机	Q0.2	
12		蒸阀	Q0.3	
13		排泄阀	Q0.4	
14		排泄泵	Q0.5	
15				
16		达到高等级	M0.1	
17		搅拌机计时	T37	
18		循环计数	C30	
19		Mix_Time_Minutes	15.3	
20		Mix_Temp_Cent_min	90	
21		Mix_Temp_Cent_max	95	
22		Process_ID	"P418369B"	

USER1 / POU Symbols

IEC全局变量表举例：

符号	地址	数据类型	注解
1	Start_1	%I0.0	BOOL
2	Start_2	%I0.1	BOOL
3	Stop_1	%I0.2	BOOL
4	Stop_2	%I0.3	BOOL
5	High_Level	%I0.4	BOOL
6	Low_Level	%I0.5	BOOL
7	Reset	%I0.7	BOOL
8			
9	Pump_1	%Q0.0	BOOL
10	Pump_2	%Q0.1	BOOL
11	Mixer_Motor	%Q0.2	BOOL
12	Steam_Valve	%Q0.3	BOOL
13	Drain_Valve	%Q0.4	BOOL
14	Drain_Pump	%Q0.5	BOOL
15			
16	High_Lev_Reached	%M0.0	BOOL
17	Mix_Timer	%T37	TON
18	Cycle_Counter	%C30	CTU
19	Counter_Out	%M0.2	BOOL
20	Counter_Current	%M\W3	INT
21	Mix_Time_Minutes	15.3	REAL
22	Mix_Temp_Cent_min	9	BYTE
23	Mix_Temp_Cent_max	95	BYTE
24	Process_ID	"P418369B"	STRING

[返回顶端](#)

另请参阅:

[局部变量表](#)

[检视符号编址](#)

[网络符号信息表 \(LAD/FBD\)](#)

[编址概述](#)

[直接和间接编址](#)

[关键字列表](#)

[系统符号表](#)

[符号表标记 \(工具 > 选项\)](#)

### 3.3.5 SIMATIC和IEC 1131-3数据类型

在STEP 7-Micro/WIN中编程时，在下列两种情况下您必须理解并使用数据类型：

- 如果您以IEC 1131-3模式编程，则必须为全局符号表中的所有数值选择数据类型。
- 如果您在局部变量表中赋值，则必须为每个局部变量定义数据类型。

当您明确地为数值指定数据类型时，您为STEP 7-

Micro/WIN提供清晰的指令，指明需要为数值分配多少内存空间（例如，数值100可被存储为字节、字或双字）以及如何表示数值（将0视作布尔值还是数值？）。

每个SIMATIC和IEC 1131-

3指令或参数化子例行程序均由精确定义识别。该定义可被称为签名。对于所有的标准指令，每个指令操作数允许使用的数据类型集均从签名获得。对于参数化子例行程序，用户通过局部变量表建立子例行程序签名。

STEP 7-Micro/WIN V3.0编程软件为SIMATIC模式执行简单的数据类型检查，为IEC 1131-

3模式执行细致的数据类型检查。这意味着为局部或全局变量指定一种数据类型时，软件在下表所列的每个等级中确保操作数数基槽陀脰噶针卜 噪ッ涸

基本数据类型	数据类型大小	说明	范围
布尔	1位	布尔	0至1
字节	8位	不带符号的字节	0至255
字节	8位	带符号的字节 (SIMATIC模式仅限用于 SHRB指令)	-
128至+127			
字	16位	不带符号的整数	0至65,535
整数	16位	带符号的整数	-32768至+32767
双字	32位	不带符号的双整数	0至4294967295
双整数	32位	带符号的双整数	-2147483648至 +2147483647
实数	32位	IEEE 32 位浮点	+1.175495E-38至 +3.402823E+38 -1.175495E-38至 -3.402823E+38
字符串	2至255字节	ASCII字符串照原样存储在PLC内存中, 形式为1字符串长度接ASCII数据字节	
	ASCII字符代码128至255		

复杂数据类型	说明	范围
TON	打开延迟计时器	1 ms 10 ms 100 ms T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255
TOF	关闭延迟计时器	1 ms 10 msms T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 - 255
TP	脉冲计时器 (参阅注释1)	1 ms 10 msms T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 - 255
CTU	向上计数器	至255
CTD	向下计数器	至255
CTUD	向上 / 向下计数器	至255
SR	设置主要双稳态	无关
RS	重设主要双稳态	无关

#### STEP 7-Micro/WIN 提供三级数据类型检查

- IEC编辑器具有细致数据类型检查。** 在该模式中, 参数数据类型必须与符号或变量的数据类型完全匹配。欲支持细致数据类型检查, 只有被指定数据类型的数据类型的变量才能成功编译。使用细致数据类型检查时, 数据类型为整数的变量对字指令参数无效。细致数据类型检查仅在 IEC 1131-3模式中进行。

#### 用户选择的数据类型 等同数据类型

布尔	布尔
字节	字节
字	字
整数	整数
双字	双字
双整数	双整数
实数	实数
字符串	字符串

- 处于SIMATIC模式中的局部变量表具有简单数据类型检查。** 在该模式中, 当符号或变量被指定一个数据类型时, 同时自动指定与所选数据类型大小相匹配的所有数据类型。例如, 如果用户选择双整数作为数据类型, 局部变量会自动指定数据类型双字, 因为二者均为32位数据类型。不自动指定实数数据类型, 虽然实数也是32位数据类型。实数数据类型被定义为无其他等同数据类型, 总是独特的类型。使用局部变量时, 只在 SIMATIC模式中执行简单数据类型检查。

#### 用户选择的数据类型 等同数据类型

布尔	布尔
字节	字节
字	字、整数
整数	字、整数
双字	双字、双整数
双整数	双字、双整数
实数	实数
字符串	字符串

3. **SIMATIC符号编辑器无数据类型检查。** 该模式仅限于无法选择数据类型的SIMATIC全局变量。在该模式中，所有相同大小的数据类型被自动指定给符号。例如，编程软件将自动为赋予VD100地址的符号指定下列数据类型：双字、双整数和实数。

#### 为SIMATIC全局符号由大小决定的数据类型

用户选择的地址	指定等同的数据类型
V0.0	布尔
VB0	字节、字符串
VW0	字、整数
VD0	双字、双整数、实数

#### 数据类型检查的益处

数据类型检查的益处是帮助用户避免常见编程错误。例如，如果指令支持带符号的数字，则在使用不带符号的数字作为指令操作数时，软件为该用法设置旗标会有益处。例如，关系比较<I是带符号的指令，显然，对于带符号的数据类型操作数，-1小于0。但当<I指令允许支持不带符号的数据类型时，程序员需确保绝不会发生下列情况。在程序运行时，对于<I指令，不带符号数40,000实际上小于0。程序员需确保对带符号的指令使用不带符号的数字不会超越正负数界限，否则会出现无法预测的结果。



**警告** 您应当确保带符号的数字用作不带符号的指令时不得超出正负数界限。若无法确保不带符号的数字用作带符号的指令时不超出正负数界限，可能致使程序或控制器操作出现无法预测的结果。无法预测的控制器操作可能导致人员死亡或严重伤害及/或财产严重损坏。请务必核实不带符号的数字用作带符号的指令时不得超出正负数界限。

总之，在IEC 1131-3编辑模式中，细致数据类型检查通过对指令的非法数据类型生成错误，帮助程序员在编译过程中发现这些错误。SIMATIC编辑器不具有此一功能。

#### 在SIMATIC与IEC1131-3之间移植程序

因为IEC1131-3属于细致数据类型，而SIMATIC不属于此一类型，STEP 7-Micro/WIN不提供在两种不同编辑模式之间移动程序的能力。程序员必须选择一种希望使用的编辑模式。

#### 超载指令

超载指令支持一系列数据类型。仍然应用细致数据类型检查，在成功编译指令之前所有的操作数数据类型必须匹配。例如，请参阅下表中用于IEC超载ADD（加）指令。

指令 类型检查)	允许使用的数据类型（细致数据类型检查） 编译指令	允许使用的数据类型（数据
ADD	整数 字、整数	ADD_I（加整数）
ADD	双整数 双字、双整数	ADD_D（加双整数）
ADD	实数 实数	ADD_R（加实数）

当所有操作数的数据类型均为双整数时，编译器生成“加双整数”指令。如果超载指令数据类型混合，会出现编译错误。根据数据类型检查的等级决定哪些为非法。在下例中，如果使用细致数据类型检查，则生成编译程序错误，但如果使用简单数据类型检查则成功编译。

ADD IN1 = INT, IN2 = WORD, IN3 = INT.

细致数据类型检查：引起编译错误。

简单数据类型检查：成功编译至ADD\_I（加整数）。

简单数据类型检查不会阻止出现常见运行时间编程错误。例如，使用简单数据类型检查时，编译程序不能帮助避免在运行时间(5) 韵鲁< 暗檀碧蝗 ADD 40000,1被解释为负数，而不是不带符号的40,001。

#### 在IEC中为超载指令使用直接编址

##### IEC-

1131编程模式也允许使用直接表示的内存位置，作为指令参数配置的一部分。参数中可使用变量和内存位置。但是，使用直接表示的内存位置时必须清楚地理解数据类型的隐含意义，因为这些位置不包含明确的类型信息。另外，不能从任何超载 IEC指 耀韶g嘈托畔iii 蛭 小噶鬲邮斩噶质 董嘈汀

直接表示的参数数据类型通过检查指令中包含的其他类型的参数来决定。当指令参数配置使用某一指定类型的变量时，则假定 兄笨颖碾镜牟问 美嘈汀

名称	地址	数据类型	注解
Var1		实数	这是浮点数变量。
Var2		双整数	这是双整数变量。

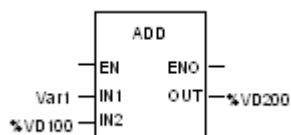
Var3

整数

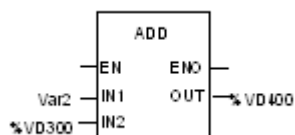
这是整数变量。

举例

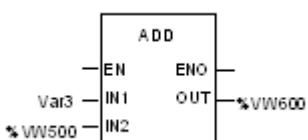
说明



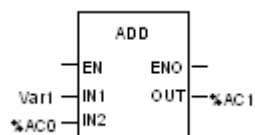
假定VD100和VD200为实数类型，因为Var1是实数类型。



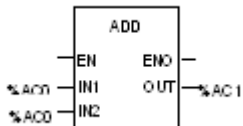
VD300和VD400为双整数类型，因为Var2是双整数类型。



假定VW500和VW600为整数类型，因为Var3是整数类型。

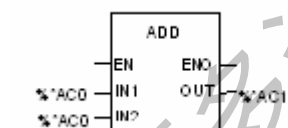


假定AC0和AC1为实数类型，因为Var1是实数类型。



实数。

这种配置被视为非法，因为无法确定类型。累加器中的数据类型可为双整数或



数或实数。

这种配置被视为非法，因为无法确定类型。累加器指针中的数据类型可为双整

**使用数据类型转换指令**

转换指令允许从一种数据类型移至另一种数据类型。STEP 7-Micro/WIN支持在下列简单数据类型之间的移动数值转换指令。

**转换指令****IEC细致数据类型检查 允许的操作数****SIMATIC局部变量表中的简单数据类型检查****允许的操作数**

字节至整数

入： 字节 出： 整数

入： 字节 出： 字、整数

整数至字节

入： 整数 出： 字节

入： 字、整数 出： 字节

整数至双整数

入： 整数 出： 双整数

入： 字、整数 出： 双字、双整数

双整数至整数

入： 双整数 出： 整数

入： 双字、双整数 出： 字、整数

双整数至实数

入： 双整数 出： 实数

入： 双字、双整数 出： 实数

实数至双整数（进位）

入： 实数 出： 双整数

入： 实数 出： 双字、双整数

字符串至整数

入： 字符串 出： 双字（\*指向字符串的指针） 出： 整数

入： 字符串，

双字（\*指向字符串的指针） 出： 字、整数

整数至字符串

入： 整数 出： 用双字代表的字符串（\*指向字符串的指针）

入：

字、整数 出： 用双字代表的字符串（\*指向字符串的指针）



字符串至双整数	入: 字符串	双字 (* 指向字符串的指针) 出: 双整数	入: 字符串,
双字 (* 指向字符串的指针) 出: 双字、双整数			
双整数至字符串	入: 双整数	出: 用双字代表的字符串 (* 指向字符串的指针)	入:
双字、双整数 出: 用双字代表的字符串 (* 指向字符串的指针)			
字符串至实数	入: 字符串	双字 (* 指向字符串的指针) 出: 实数	入: : 字符串
双字 (* 指向字符串的指针) 出: 实数			
实数至字符串	入: 实数	出: 用双字代表的字符串 (* 指向字符串的指针)	入: 实数
出: 用双字代表的字符串 (* 指向字符串的指针)			


在IEC 1131-

3编辑模式中, 您可以使用超载移动指令在整数和字、双整数和双字之间转换。超载移动指令允许相同大小的数据类型自由移动

#### IEC 1131-3超载移动

	入	出
移动 (整数至字)	整数	字
移动 (字至整数)	字	整数
移动 (双整数至双字)	双整数	双字
移动 (双字至双整数)	双字	双整数

### 3.3.6 STL程序书签

您可以在设置程序中的书签  , 使之便于在较长程序中的指定网络之间来回移动:

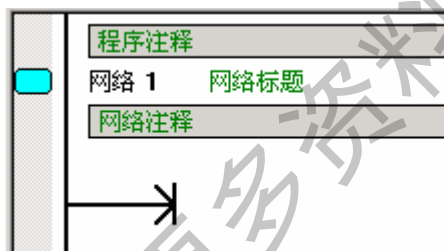


**切换书签:** 单击该按钮, 在由当前光标位置指定的程序网络位置设置或移除书签。

**下一个书签:** 单击该按钮, 向下移至程序的下一个带书签的网络。

**前一个书签:** 单击该按钮, 向上移至程序的前一个带书签的网络。

**移除所有的书签:** 单击该按钮, 移除程序中的所有当前书签。



另请参阅:

[应用程序用户参考手册](#)

[入门指南目录](#)

[编辑器比较: LAD、FBD、STL \(GS 2.5\)](#)

[程序编辑器](#)

### 3.3.7 使用“查找”和“替换”进行修改

欲使用“查找”/“替换”或“转入”:

- 选择 **编辑 > 查找 (Edit>Find)**、**编辑 > 替换 (Edit>Replace)** 或 **编辑 > 转入 (Edit>GoTo)** 菜单命令
- 按 CTRL+F 执行查找, CTRL+H 执行替换或 CTRL+G 执行转入

**如何使用:**

[查找](#)

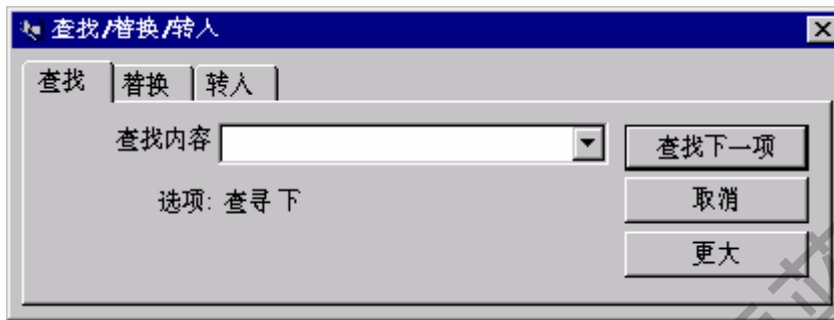
[替换](#)

**在何处使用**

可以在程序编辑器窗口、局部变量表, 符号表、状态图表、交叉引用标签和数据块中使用“查找”、“替换”和“转入”。

**如何使用**

- “查找”功能允许您查找指定的字符串，例如操作数、网络标题或指令助记符。（“查找”不搜索网络注解，仅搜索符号）
- “替换”功能允许您替换指定的字符串。（“替换”不对指令助记符操作。）
- “转入”功能允许您通过指定网络号码或您希望浏览至的行快速移至另一个位置。

**查找功能**[返回顶端](#)

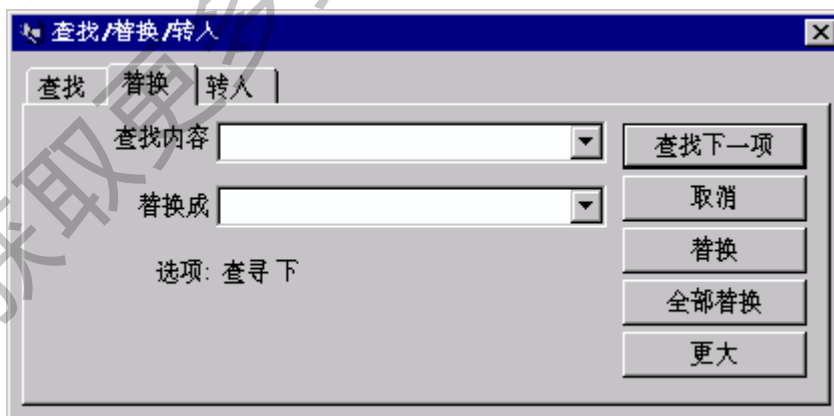
1. 在“查找内容”栏中键入要搜索的字符串。

2. 欲移至下一个搜索字符串，单击“查找下一项”按钮。

**注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反应了操作数在代码中的存储方式。另外，如果您在 STL 中创建的程序包含在 LAD 或 FBD 编辑器中非法的网络，在 LAD 或 FBD 编辑器中查看该程序时，“查找下一项”命令不对这些网络进行搜索操作。

欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方式定制搜索：

- 您可以使用“搜索”列表框选择搜索的方向。
- 您可以选择“大小写匹配”复选框，仅搜索与您键入字符串大小写数值相同的字符串。
- 您可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的“SPAN”复选框，对所有的 POU（OB1、所有的子程序和中断例行程序）或局部变量表的所有实例
- 7. 逐行逐列 逐行 逐列
- 您可以指定对一部分行进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认范围
- 荒 部梢约 庐 缙蚰泻牛 魑 阍鞞钠鸮已徒螻
- 您可以指定是否通过选择适当的复选框搜索网络标题、POU 和网络注解。

**替换功能**[返回顶端](#)

1. 在“查找内容”栏中键入要搜索的字符串。

2. 在“替换内容”栏中键入您希望用作替换搜索字符串的字符串。

3. 欲查找存在的搜索字符串，单击“查找下一项”按钮。

**注释：**在某些情况下，“查找下一项”命令可能看起来不按顺序搜索程序代码，但实际上这种看似不规则的顺序却反应了操作数在代码中的存储方式。另外，如果您在 STL 中创建的程序包含在 LAD 或 FBD 编辑器中非法的网络，在 LAD 或 FBD 编辑器中查看该程序时，“查找下一项”命令不对这些网络进行搜索操作。

4. 如果您希望替换字符串，单击“替换”。如果您已经仔细地定义搜索字符串而且没有误改的可能，您可以单击“全部  
 娥弧保 娥凰 写婁请淖址 才 胫鸚患徽梨扛蚤址

欲进一步定义搜索，必须单击“更多”按钮，该按钮允许您按照下列方法定制搜索：

- 您可以使用“搜索”列表框选择搜索的方向。
- 您可以选择“大小写匹配”复选框，仅搜索与您“查找内容”中键入字符串大小写数值相同的字符串。
- 您可以选择“全字”复选框，除去包含作为较长字的一部分的搜索短语的字符串。
- 您可以选择适当的“SPAN”复选框，对POU（OB1、所有的子程序和中断例行程序）或局部变量表、符号表或状态图  
 礙械乃 惶道 兴闹整
- 您可以指定对一部分行进行搜索，如果您在程序编辑器中选择了网络范围，它们将成为“查找”对话框中的默认范围  
 荒 部梢约 驴 缙蚰泻牛 魑 闹鞞钠鸚已徒嶶
- 您可以指定是否通过选择适当的复选框搜索网络标题、POU和网络注解。

### 3.3.8 为程序增加注解

#### 程序注解

所有STL中的注解前均须有双斜线；//

您可以将一整行作为注解；您可以把注解作为一行放在有效指令之后。您不能创建行中注解；双斜线之后的所有内容均被编译程序视作注解。

最大行长（包括注解）为255个字符。

举例：

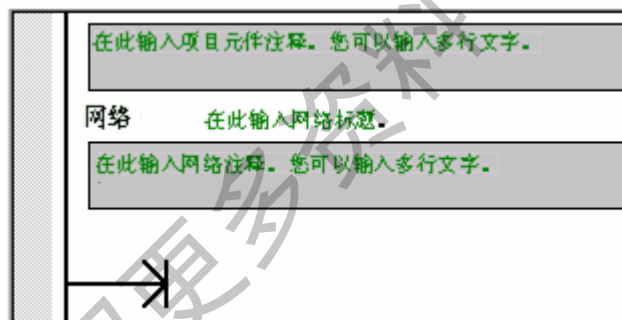
```
Network 1      //本行介绍程序的第一个网络
LD            I0.0      //在I0.0处载入数值
O            I0.1      //或带有I0.1数值
```

//此处输入更多指令：每个网络由多个指令组成

```
Network 2      //开始新的指令网络
```


在STL编辑器中共有四个注解级别：

- 项目组件注解
- 网络标题
- 网络注解
- 项目组件属性



#### 项目组件注解

在网络1上方的灰色方框中单击，输入POU注解。

您可以单击“切换POU注解”按钮  或选择和取消选择 **检视 (View) > POU注解 (POU Comments)** 选项的方法，在POU注解“打开”（可见）或“关闭”（隐藏）之间切换。


每条POU注解最多可使用4096个字符。POU注解是选用项目，显示时始终位于POU顶端，并在第一个网络之前显示。

#### 网络标题

将光标放在网络标题行中的任何位置，输入一个可识别该逻辑网络的标题。网络标题最多可使用 127个字符。

#### 网络注解

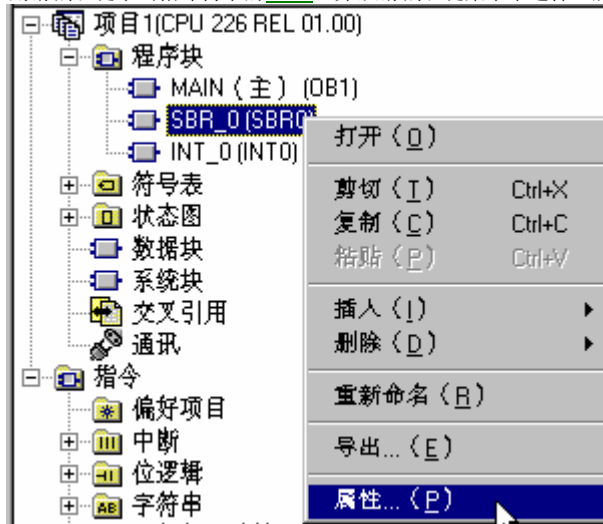
在网络1下方的灰色方框中单击，输入网络注解。您可以输入一条识别该逻辑网络的注解，并增加有关网络内容的说明。

您可以单击“切换网络注解”按钮  或选择和取消选择 **检视 (View) > 网络注解 (Network Comments)** 选项的方法，在网络注解“打开”（可见）或“关闭”（隐藏）之间切换。  
 网络注解中最多可使用4096个字符。

#### 项目组件属性

您可以用以下任何一种方法存取“属性”制表符：

用鼠标右键单击指令树中的 **POU**，并从鼠标右键菜单中选择“属性”。



用鼠标右键单击程序编辑器窗口中的 **POU** 制表符，并从弹出菜单中选择“属性”。



“属性”对话框中有两个制表符：

- 一般
- 保护



#### “一般”制表符

“一般”制表符允许您为子例行程序、中断例行程序和主程序块（OB1）重新编号和重新命名，并为项目指定一个作者。

#### 注释：

您不能将默认名称（由STEP 7-Micro/WIN指定的POU地址，例如，SBR1代表子例行程序或INT1代表中断例行程序）用作符号名，因为这将构成重复。STEP 7-Micro/WIN会报告一则错误。

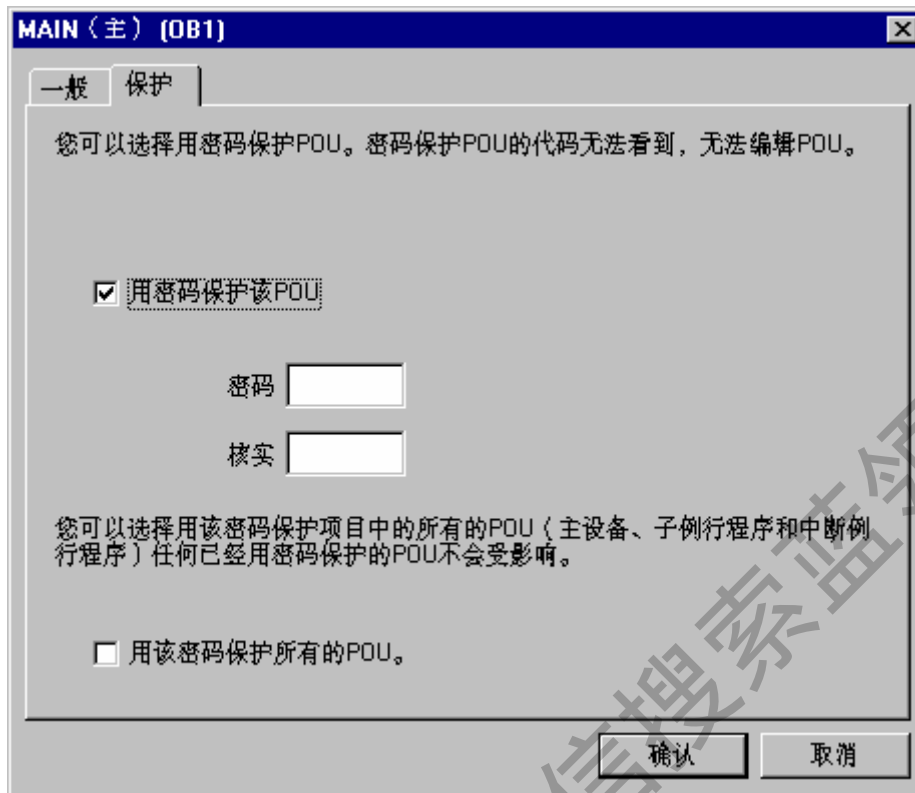
如果您为程序中的POU指定符号名，符号名会在程序代码中显示，即使您没有启用“符号编址”视图亦如此。“符号表”显示一个特殊的制表符（“POU符号”），该制表符列出所有符号名赋值。您只能检视该制表符，但无法从“符号表”编辑条目。欲更改赋值，您必须编辑适当POU的“属性”对话框。

#### “保护”制表符

“保护”制表符允许您选择用密码保护POU，以便使其他用户无法看到POU，并在下载时加密。

欲用密码保护POU：

1. 用鼠标右键单击“主POU”，并选择“属性...”。
2. 选择“保护制表符”。
3. 选择“密码—保护POU”复选框。输入并核实由四个字符组成的密码。



您还可以选择保护项目中的所有POU（主、子例行程序和中断例行程序），方法是选择“密码—保护所有POU”。

POU受到保护后，则会已以下方式在指令树中显示：



此外，会在程序编辑器窗口中显示一个灰色锁定图标 .

欲为POU解除锁定：

1. 浏览至“保护制表符”。
2. 输入密码。
3. 单击“授权”。

您必须为选择保护的每个POU重复该步骤。



### 3.3.9 理解S7-200计时器指令

您可利用计时器执行时基计数功能。S7-200指令集提供三种不同类型的计时器。

- 打开延迟计时器 (TON)，用于单间隔计时
- 保留性打开延迟计时器 (TONR)，用于累计一定数量的定时间隔
- 关闭延迟计时器 (TOF)，用于延长时间以超过关闭 (或假条件)，例如马达关闭后使马达冷却。

计时器操作：

计时器类型	当前值 >= 预设值	启用输入“打开”	启用输入“关闭”
电源循环 / 首次扫描			
TON	计时器位打开，当前值继续计数 直至达到32,767	当前值记录时间	计时器位关闭，
当前值=0	计时器位关闭，当前值=0		
TONR	计时器位打开，当前值继续计数 直至达到32,767	当前值记录时间	计时器位及当前
值保持最后的状态	计时器位关闭，可保持当前值(1)		
TOF	计时器位关闭，当前值=预设值，停止计数	计时器位打开，当前值=0	
从“打开”转换为“关闭”后计时器开始计数	计时器位关闭，当前值=0		

(1) 可通过电源循环为保留目的选择保留性计时器当前值。请参阅 [保留范围标记—系统块配置](#) 中有关S7?00 CPU的内存保留信息。

请参阅文档光盘“提示和技巧”中的提示31，查阅使用打开延迟计时器 (TON) 的抽样程序。

**注释：**

可用“重设”(R)指令重设任何计时器。“重设”指令执行下列操作：

计时器位=关闭，计时器当前值=0

只能用“重设”指令重设TONR计时器。

重设后，TOF计时器要求启用输入从“打开”转换为“关闭”，以便重新启动。

**1毫秒分辨率**

1毫秒计时器记录自现用1毫秒计时器启用以来1毫秒计时器间隔的数目。执行计时器指令即开始计时；但是，1毫秒计时器每毫秒更新一次（计时器位及计时器当前值），不与扫描循环同步。换言之，在超过1毫秒的扫描过程中，计时器位和计时器当前值将

啜胃 隆

计时器指令用于打开和重设计器，如果是TONR计时器，则用于关闭计时器。

因为可在1毫秒内的任意时刻启动计时器，预设值必须设为比最小要求计时器间隔大一个时间间隔。例如，使用1毫秒计时器时，为了保证时间间隔至少为56毫秒，则预设时间值应设为57。

#### 10毫秒分辨率

10毫秒计时器记录自现用10毫秒计时器启用以来10毫秒计时器间隔的数目。执行计时器指令即开始计时；但是，在每次扫描循环的开始更新10毫秒计时器，其方法是以当前值加上积累的10毫秒间隔的数目（自前一次扫描开始算起）（换言之，在整个扫描过程中，计时器当前值及计时器位保持不变）

因为可在10毫秒内的任意时刻开始计时器，预设值必须设为比最小要求计时器间隔大一个时间间隔。例如，使用10毫秒计时器时，为了保证时间间隔至少为140毫秒，则预设时间值应设为15。

#### 100毫秒分辨率

100毫秒计时器记录自现用100毫秒计时器上一次更新以来100毫秒计时器间隔的数目。这种计时器的更新方法是在执行计时器指令时更新当前值。只有当100毫秒间隔的数目（自前一次扫描开始算起）。

只有在执行计时器指令时才对100毫秒计时器的当前值进行更新。因此，如果启用了100毫秒计时器但并未对各扫描循环执行计时器指令，则仍不能更新计时器当前值并将丧失时间。同样，如果在单个扫描循环内多次执行100毫秒计时器指令，将向计时器的鼻爸刀啮卧龟100毫秒间隔数，赢得时间。只有在每次扫描循环仅仅执行一次计时器指令时，才应该使用100毫秒计时器。

因为可在100毫秒内的任意时刻启动计时器，预设值必须设为比最小要求计时器间隔大一个时间间隔。例如，使用100毫秒计时器时，为了保证时间间隔至少为2100毫秒，则预设时间值应设为22。

#### 更新计时器当前值

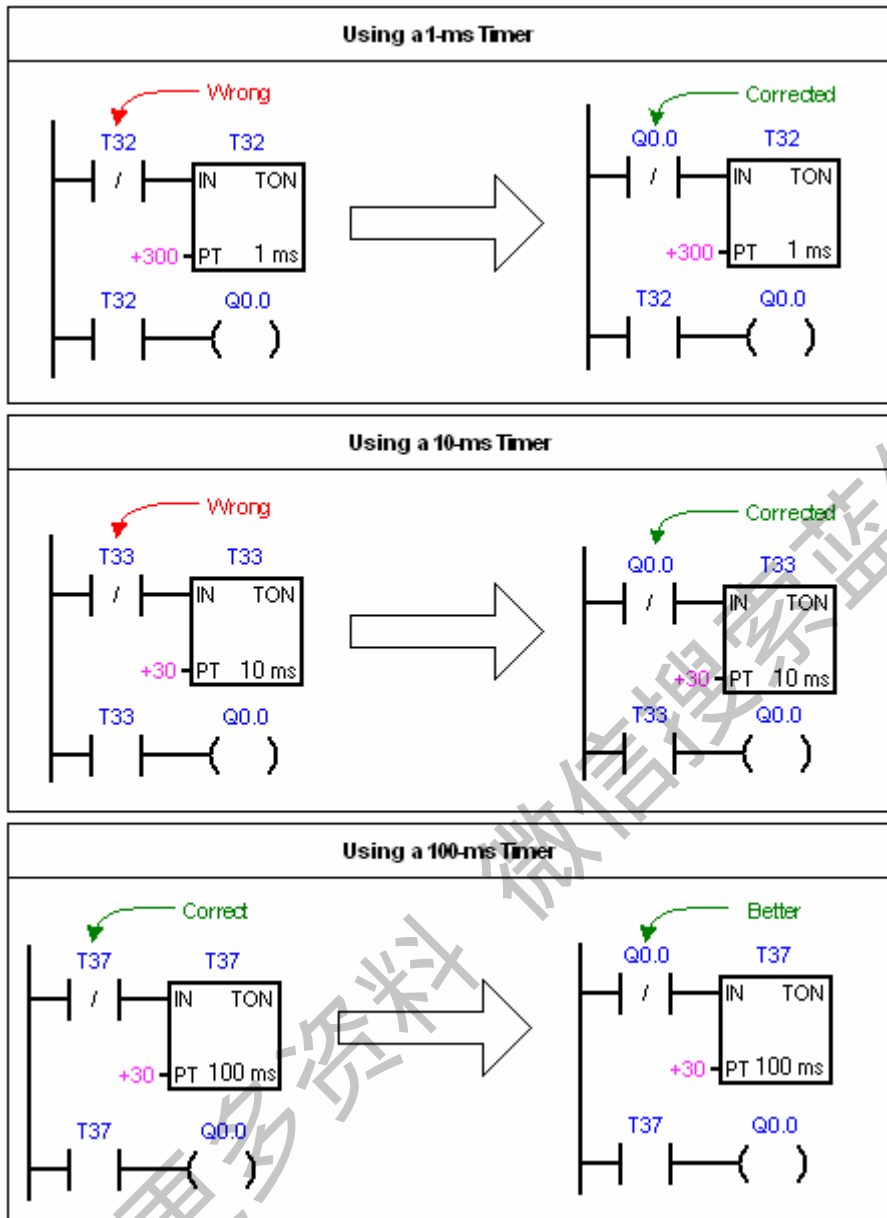
更新当前时间值有多种方式，其作用取决于如何使用计时器，如下图所示的计时器操作：

- 使用1毫秒计时器时，每次在执行正常关闭接点T32之后和执行正常打开接点T32之前更新计时器当前值时，Q0.0即打开进行一次扫描。
- 使用10毫秒计时器时，从不打开Q0.0，因为从扫描顶端至执行计时器方框均打开计时器位T33。一旦执行了计时器方框，计时器的当前值及T位均被设为零。执行正常打开接点T33时，T33及Q0.0均被关闭。
- 使用100毫秒计时器时，每当计时器当前值达到预设数值时，Q0.0时钟打开进行一次扫描。

通过使用正常关闭解点Q0.0代替计时器位作为计时器方框的启用输入，可保证每次计时器达到预设值时均打开输出Q0.0进行一次扫描。

#### 自动重新触发一击计时器举例

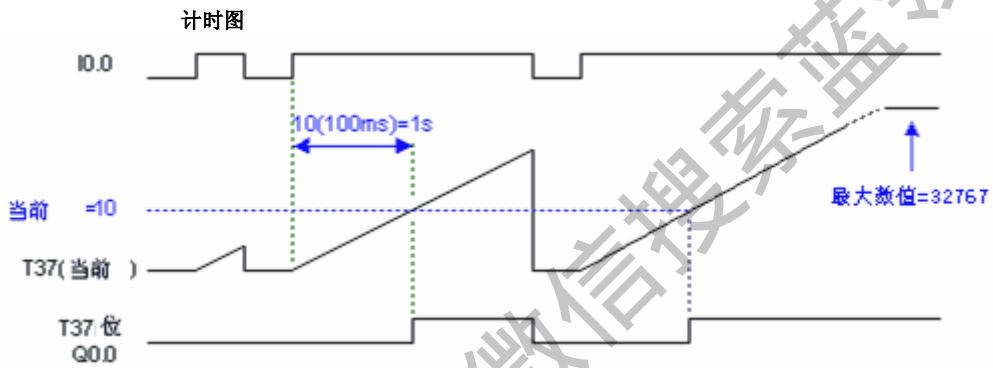
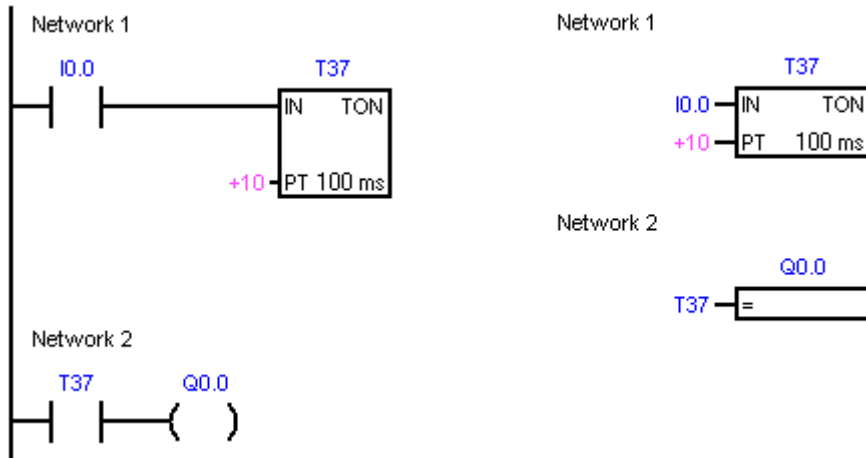




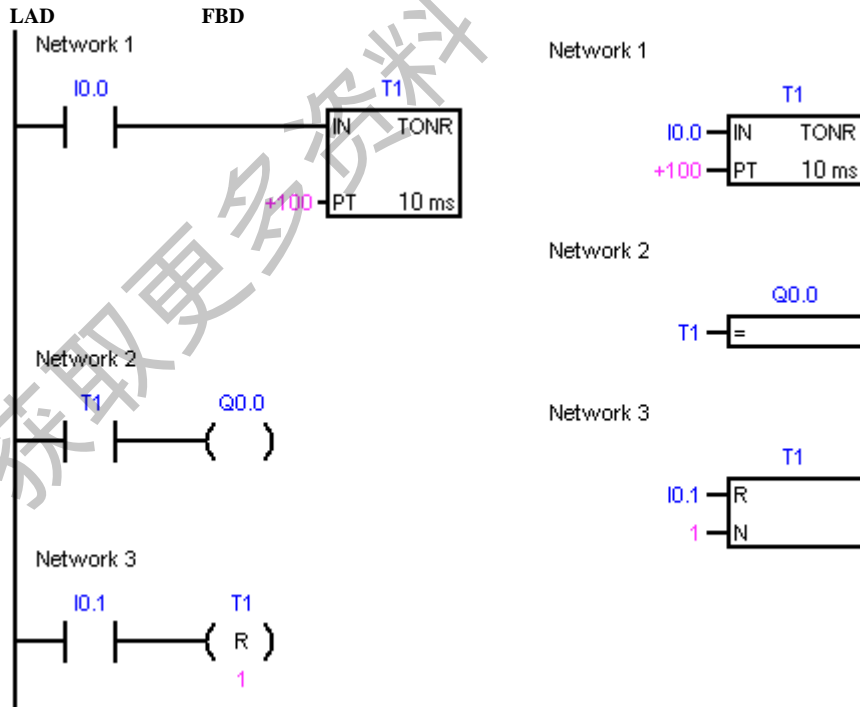
打开延迟计时器举例

LAD

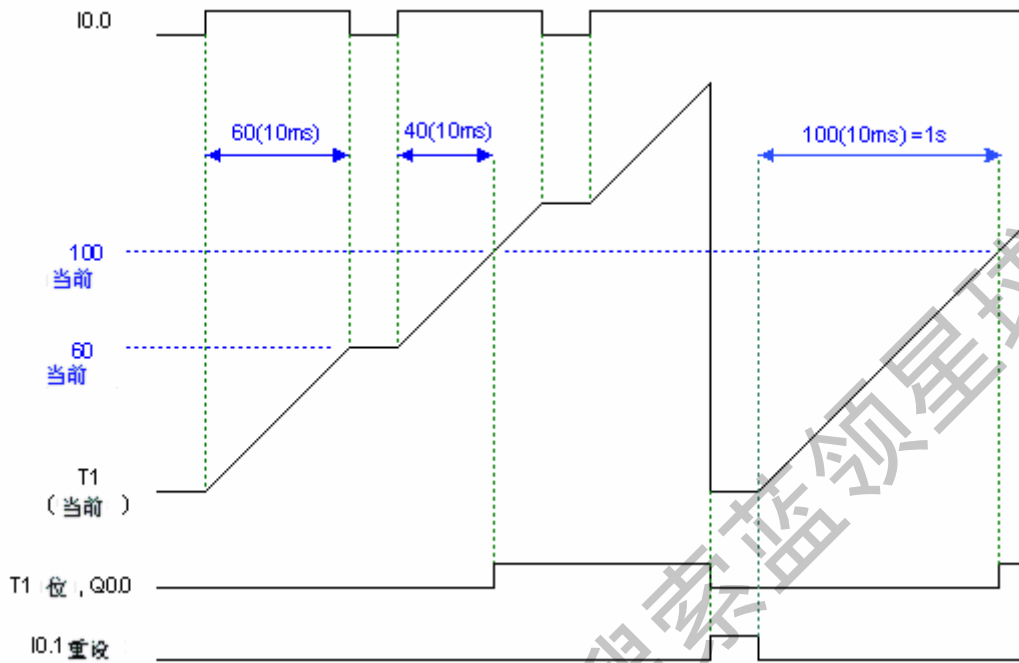
FBD



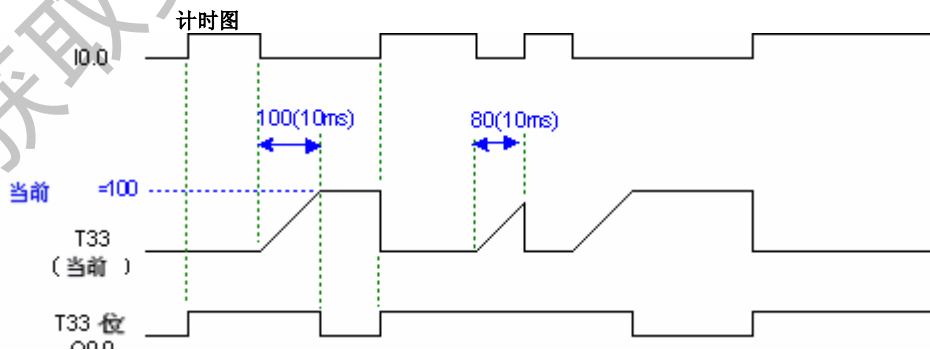
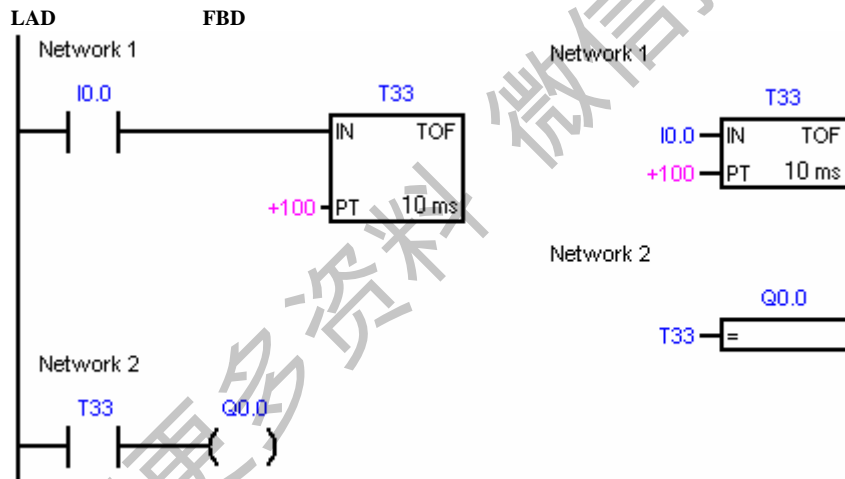
保留性打开延迟计时器举例



计时图



关闭延迟计时器举例



### 3.3.10 理解S7-200计数器指令

每次向上计数输入执行从关闭至打开转换时，向上计数（CTU）从该计数器的当前值向上计数。重设输入打开或执行重设指令时，计数器被重设。达到最大值（32,767）时，计数器停止。

每次向上计数输入执行从关闭至打开转换时，向上/向下计数器（CTUD）向上计数，每次向下计数输入执行从关闭至打开转换时，向下计数输入执行从关闭至打开转换时，向上计数输入的下一个上升边缘导致当前计数变成最小值（32,768）。与此相似，达到最小值（-32,768）时，向下计数输入的下一个上升边缘导致当前计数变成最大值（32,767）。

向上和向上/向下计数器有一个保持当前计数的当前值。计数器还有一个预设值（PV），每次执行计数器指令时，将预设值与当前值进行比较。如果当前值大于或等于预设值，C位关闭。否则，C位打开。

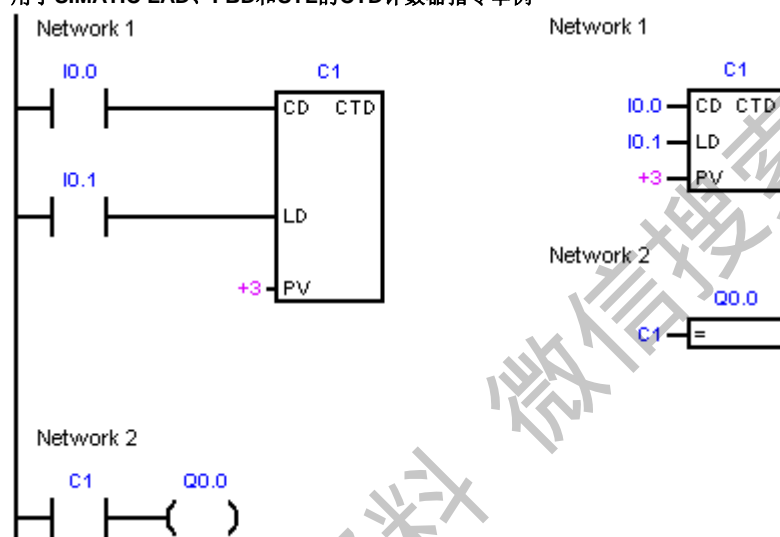
每次向下计数输入执行从关闭至打开转换时，向下计数器（CTD）从该计数器的当前值向下计数。载入输入打开时，计数器重设计数器位，并将预设值载入当前值。达到零时，计数器停止，计数器位（C位）打开。

当您使用重设指令重设计数器时，计数器位被重设，计数器当前值被设为零。使用计数器号码引用该计数器的当前值和C位。

#### 注释：

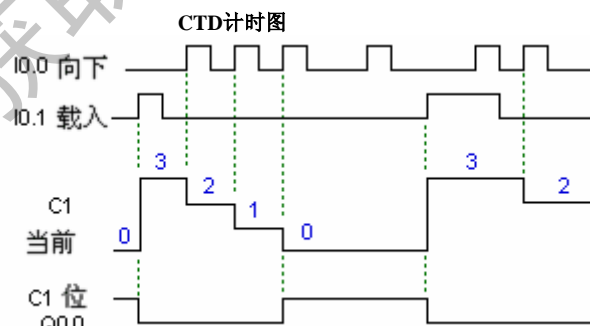
因为每台计数器有一个当前值，请勿将相同的号码指定给一台以上计数器。（向上计数器、向上/向下计数器和向下计数器存取相同的当前值。）

#### 用于SIMATIC LAD、FBD和STL的CTD计数器指令举例

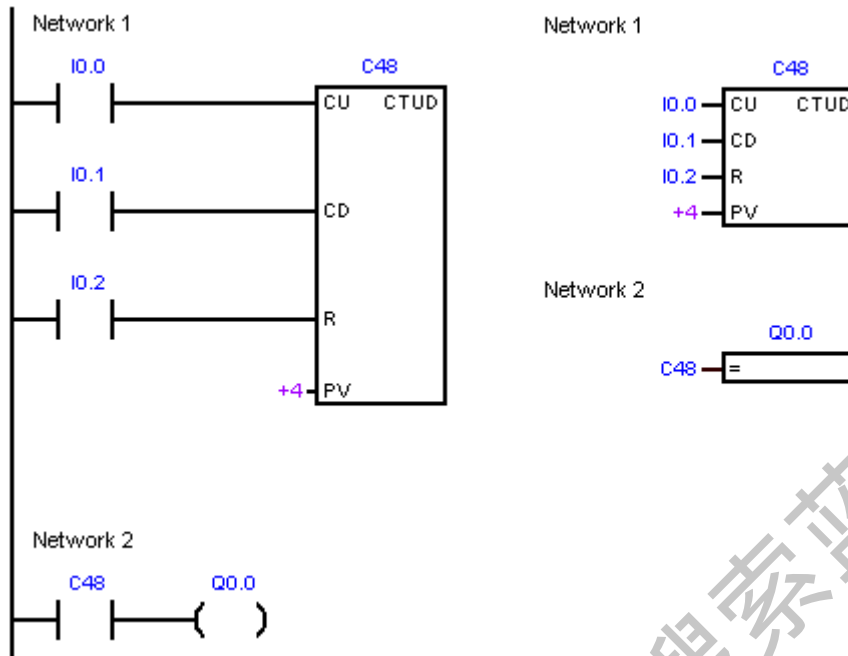


NETWORK 1 // 向下计数器C1当前值从3计数至0，  
// 其中IO.1关闭，IO.0"关闭"打开"递减C1当前值  
// IO.1"打开"载入向下计数预设值3  
LD IO.0  
LD IO.1  
CTD C1 +3

NETWORK 2 // 当计数器C1当前值 = 0时，C1位"打开"  
LD C1  
= Q0.0



用于SIMATIC LAD、FBD和STL的CTUD计数器指令举例

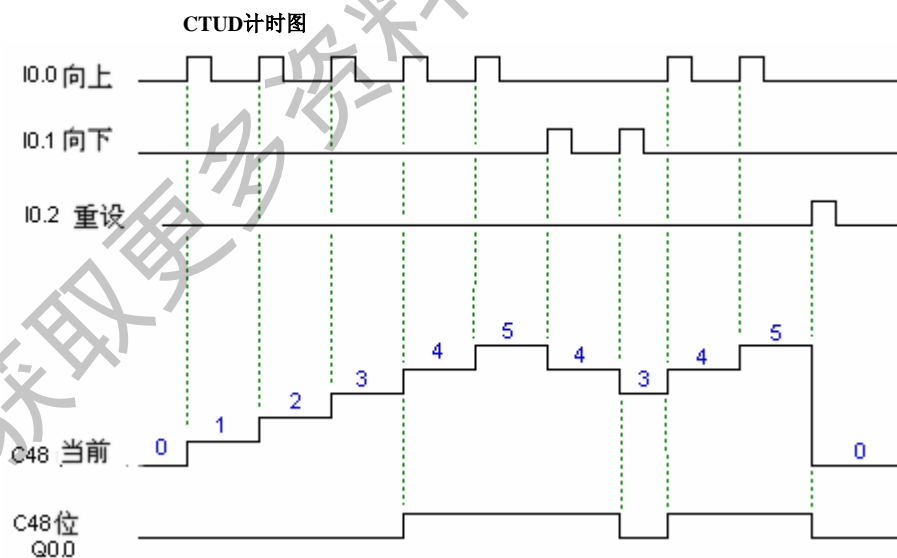


```

NETWORK 1 // I0.0向上计数 - I0.1向下计数 - I0.2将当前值重设为0
LD I0.0
LD I0.1
LD I0.2
CTUD C48 +4
  
```

```

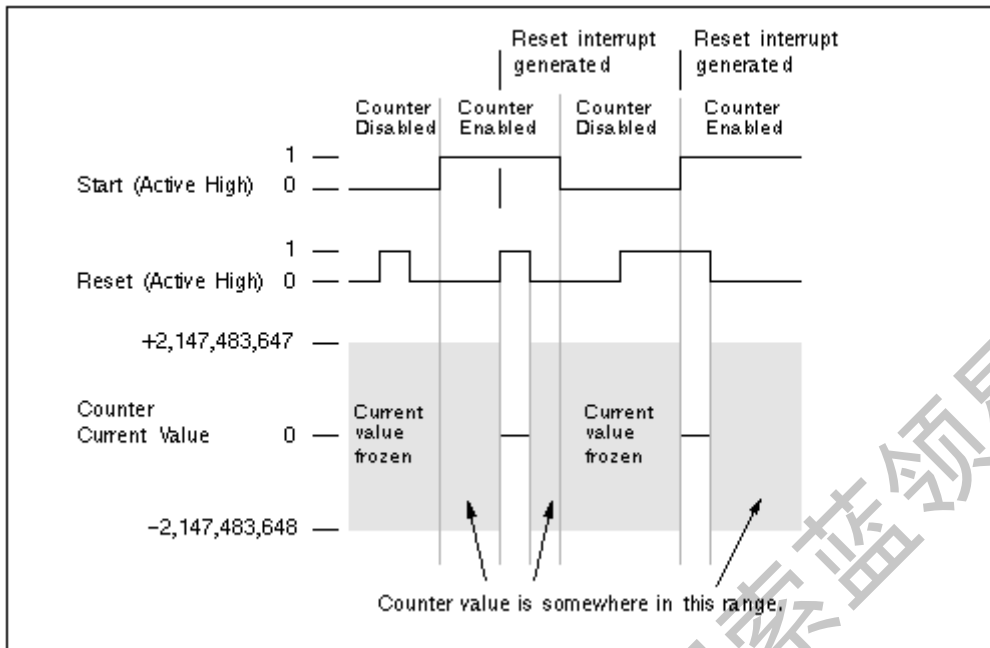
NETWORK 2 // 当前值 >= 4时, 向上 / 向下计数计数器C48打开C48位
LD C48
= Q0.0
  
```



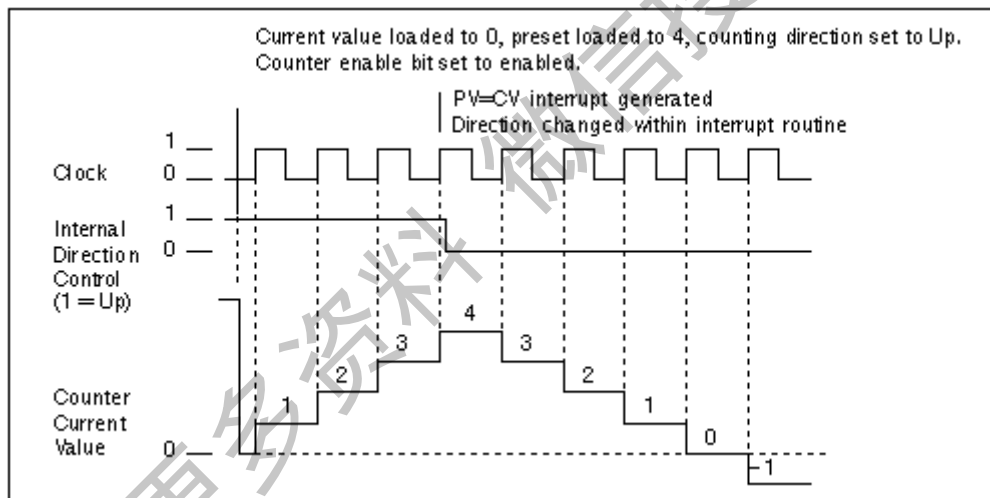
### 3.3.11 理解高速计数器指令

高速计数器对CPU扫描速率无法控制的高速事件进行计数，最多可配置12种不同的操作模式。高速计数器的最高计数频率取决于您的CPU类型。

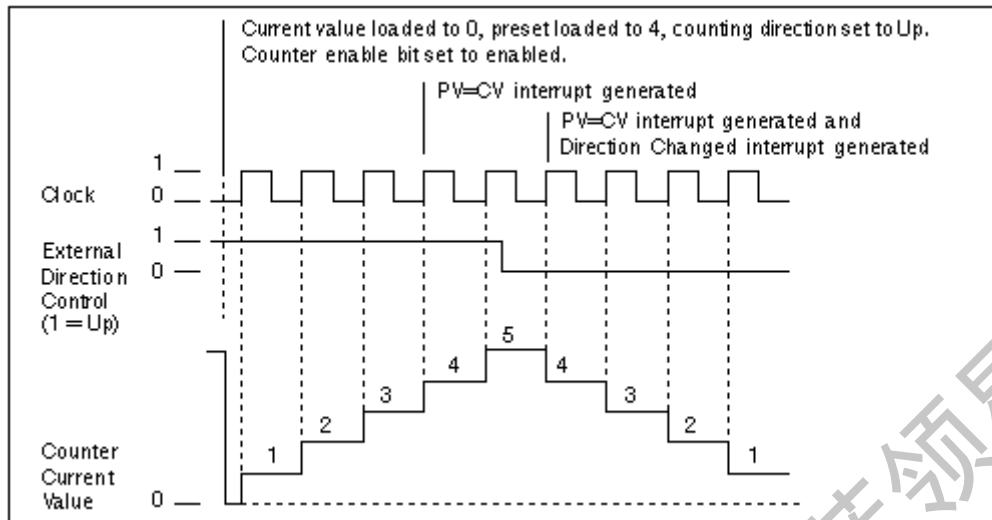




模式0、1和2操作举例

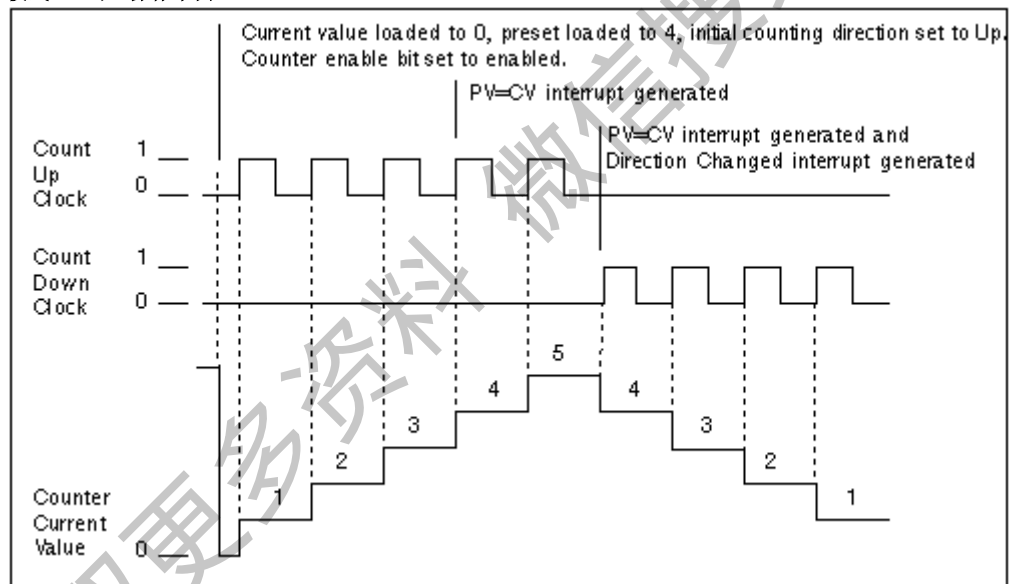


模式3、4和5操作举例



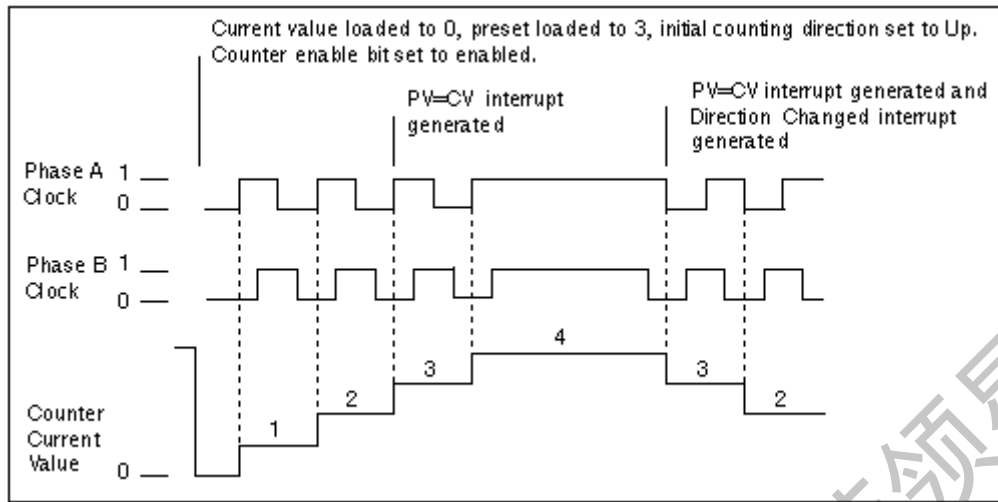
使用计数模式6、7和8时，上下时钟输入的上升边缘间隔0.3微秒，高速计数器可能认为这些事件同时发生。如果发生这种情况，在两种情况下，均不生成错误，而且计数器保持当前计数值。

#### 模式6、7和8操作举例

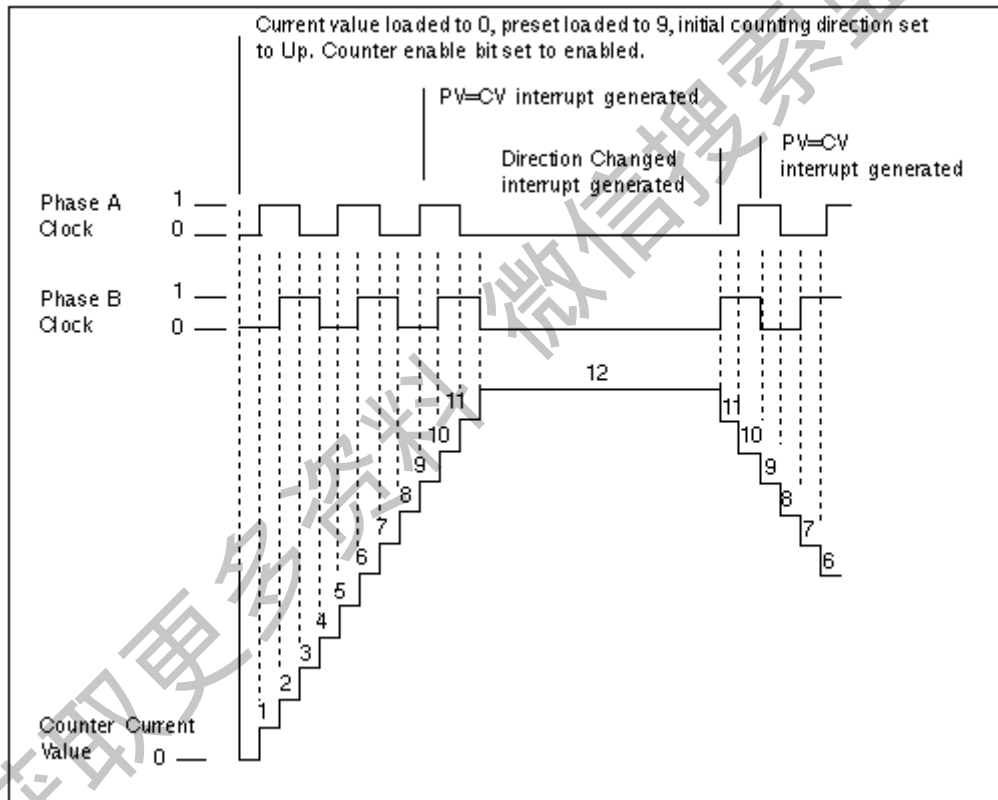


#### 模式9、10和11操作举例（正交 1x 模式）





模式9、10和11操作举例（正交 4x 模式）



为高速计数器连接输入线

[返回顶端](#)

使用“高速计数器定义”指令定义计数器模式和输入。  
下表显示与高速计数器相关的用于时钟、方向控制、重设和起始功能的输入。

高速计数器专用输入

高速计数器	使用的输入
HSC0	I0.0, I0.1, O.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

有些高速计数器和边缘中断的输入点赋值存在某些重叠。同一个输入不能用于两种不同的功能；但是高速计数器当前模式未使用的任何输入均可用于其他目的。例如，如果在模式2中使用HSC0，模式2使用I0.0和I0.2，则I0.1可用于边缘中断或用于HSC3。如果所用的HSC0模式不使用输入I0.1，则该输入可用于HSC3或边缘中断。与此相似，如果所选的HSC0模式不使用I0.2，则该输入可用于边缘中断；如果所选HSC4模式不使用I0.4，则该输入可用于HSC5。

请注意HSC0的所有模式均使用I0.0，HSC4的所有模式均使用I0.3，因此当使用这些计数器时，这些输入点绝不会用于其他用途。

### HSC模式

[返回顶端](#)

HSC 模式	说明	输入			
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I0.2	I1.1
	HSC2	I1.2	I1.3	I1.1	I1.2
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	具有内部方向控制的单相计数器	时钟			
1		时钟		重设	
2		时钟		重设	开始
3	具有外部方向控制的单相计数器	时钟	方向		
4		时钟	方向	重设	
5		时钟	方向	重设	开始
6	具有两个时钟输入的两相计数器	正计时	倒计时		
7		正计时	倒计时	重设	
8		正计时	倒计时	重设	开始
9	A / B 相正交计数器	时钟 A	时钟 B		
10		时钟 A	时钟 B	重设	
11		时钟 A	时钟 B	重设	开始
12	仅HSC0和HSC3支持12模式。 HSC0计数Q0.0所发脉冲的数目。 HSC3计数Q0.1所发脉冲的数目。				

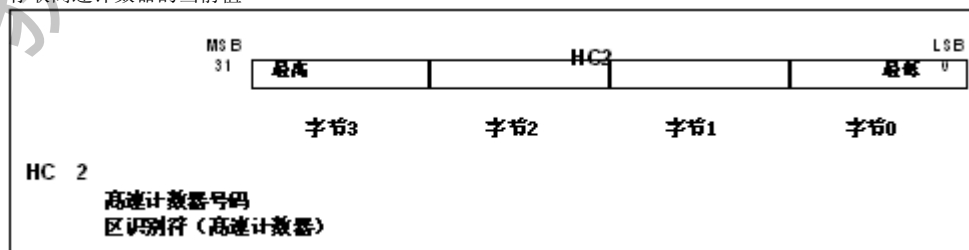
### 为高速计数器编址 (HC)

[返回顶端](#)

欲存取高速计数器的计数值，您需要利用内存类型 (HC) 和计数器号码 (例如HC0) 指定高速计数器的地址。如下所示，高速计数器的当前值是只读数值，只能作为双字 (32位) 编址。

格式: HC[高速计数器号码]，以HC2为例。

存取高速计数器的当前值



理解不同的高速计数器

[返回顶端](#)

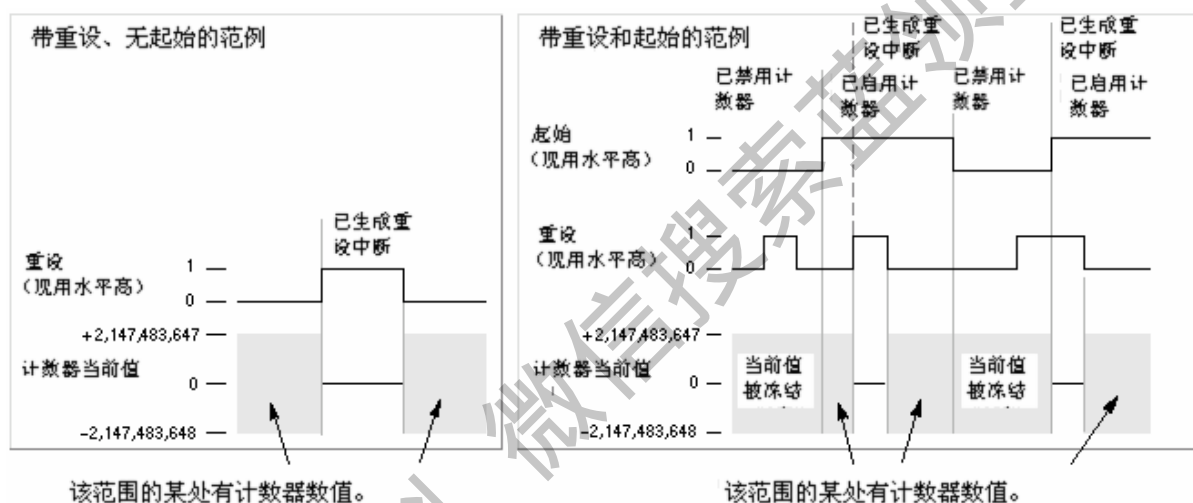
对于相同的计数器操作模式，所有计数器的功能均相同。如上所示，共有四种基本计数器模式类型。请注意并非每一种计数器均支持每种模式。您可以使用每种类型：无重设或起始输入、有重设但无起始或有起始和重设输入。

- 激活重设输入时，会清除当前值并保持清除状态直至取消激活重设。
- 激活起始输入时，会允许计数器计数。起始被取消激活时，计数器的当前值保持恒定，并忽略时钟事件。
- 如果在起始未激活时激活重设，则会忽略重设，当前值不变。如果激活重设输入时激活起始输入，则当前值被清除

使用高速计数器之前必须选择计数器模式，您可以利用 HDEF 指令（高速计数器定义）选择计数器模式。利用首次扫描内存位 S M0.1（首次扫描时该位打开，随后关闭），调用包含 HDEF 指令的子例行程序。

**选择激活状态和 1x/4x 模式**[返回顶端](#)

上图显示的重设和起始输入操作适用于使用重设和起始输入的所有模式。在重设和起始输入图形中，重设和起始均显示为现用状态编程为高位。



四台计数器有三个控制位，用于配置重设和起始输入的激活状态并选择 1x 或 4x 计数模式（仅限正交计数器）。这些控制位位于各自计数器的控制字节内，只在执行 HDEF 指令时才使用。

执行 HDEF 指令之前，必须将这些控制位设为所需的状态，否则计数器采用所选计数器模式的默认配置。重设输入和起始输入的仙柚梦 钟盟 礁撸 患割 俾饰 4x（或 4 乘以输入时钟频率）。一旦执行了 HDEF 指令，就不能再改变计数器设置，除非紫冉 CPU 设为 STOP（停止）模式。

**HDEF 控制位**

（仅在执行 HDEF 时使用）

HSC0	HSC1	HSC2	HSC4	说明
SM37.0	SM47.0	SM57.0	SM147.0	“重设”现用水平控制位**：0 =
重设现用水平高 1 = 重设现用水平低				
	SM47.1	SM57.1		“起始”现用水平控制位**：0 =
起始现用水平高 1 = 起始现用水平低				
SM37.2	SM47.2	SM57.2	SM147.2	“正交”计数器的计数速率选项：0 =
= 4x 计数速率 1 = 1x 计数速率				
**重设输入和起始输入的默认值为现用水平高，求积计数率为 4x（或 4 乘以输入时钟频率）。				

**控制字节**[返回顶端](#)

一旦定义了计数器和计数器模式，您就可以为计数器动态参数编程。每台高速计数器均有一个控制字节，允许完成以下作业：

- 启用或禁用计数器
- 控制方向（仅限模式 0、1 和 2）或初始化所有其他模式的计数方向
- 载入当前值

通过执行 HSC 指令可激活控制字节以及相关当前值和预设值检查。下表说明每个控制位。

**用于 HSC 参数的 SM 控制位**

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

### 设置当前值和预设值

[返回顶端](#)

每台高速计数器都有一个32位当前值和一个32位预设值，当前值和预设值均为带符号的整数。欲向高速计数器载入新的当前值和预设值，您必须设置包含当前值和 / 或预设值的控制字节及特殊内存字节。然后您必须执行 HSC 指令，将新数值传输至高速计数器。下表说明用于包含新当前值和预设值的特殊内存字节。除控制字节以及新预设值和当前值保持字节外，还可以使用数据类型 HC（高速计数器当前值）加计数器号码（0、1、2、3、4或5）读取每台高速计数器的当前值。因此，读取操作可直接存取当前值，但只有用上述 HSC 指令才能执行写入操作。

载入数值	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
新当前值	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
58						
新预设值	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162
62						

### 状态字节

[返回顶端](#)

为每台提供状态内存位的高速计数器提供状态字节，状态内存位表示当前计数方向以及当前值是否大于或等于预设值。下表定义每台高速计数器的状态位。

#### HSC0、HSC1、HSC2、HSC3、HSC4和HSC5的状态位

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

#### 注释：

只有在执行高速计数器中断例行程序时，状态位才有效。监控高速计数器状态的目的在于启用对正在执行的操作有重大影响的事件的中断程序。

### 为中断赋值

[返回顶端](#)

所有计数器模式均支持当前值等于预设值中断，使用外部重设输入的计数器模式支持将外部重设用中断。除模式 0、1和2以外，所有模式均支持外部重设输入。除模式 0、1和2以外，所有模式均支持外部重设输入。

**注释:**

如果您尝试载入新的当前值或从外部重设中断例行程序内先禁用然后再重新启用高速计数器, 会引起严重错误。

**高速计数器初始化顺序**

[返回顶端](#)

HSC1在以下初始化和操作顺序说明中被用作模型计数器。初始化说明假设 S7-200刚刚被放置在RUN (运行) 模式中, 因此首次扫描内存位为真。如果不是如此, 请记住在进入RUN (运行) 模式后, 只能为刻啮啮削 髦葱幸淮 HDEF指令。为高速计数器第二次执行HDEF会生成运行时间错误, 并不会改变该计数器首次执行HDEF时计数器的设置方式。虽然以下顺序分别显示如何更改方向、当前值和预设值, 您可以按照相同的顺序更改所有这些数值或这些数值的任何组合, 方法是以适当的方式设置SMB47数值, 然后执行HSC指令。

**0、1或2初始化模式**

下列步骤说明如何为带内部方向的单相向上 / 向下计数器 (模式 0、1或2) 初始化HSC1:

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用, 随后的扫描不再调用子例行程序  
蛇丝杉跽偕 枋奔渲葱胁(6)钩缘蚪昂垢 趵砣
2. 在初始化子例行程序中, 根据所需的控制操作载入SMB47。例如:  
SMB47 = 16#F8产生下列结果:  
启动计数器  
写入新当前值  
写入新预设值  
将方向设置为向上计数  
将起始和重设输入设为现用水平高
3. 执行HDEF指令, HSC输入设为1, 无外部重设或起始的MODE (模式) 输入设为0, 有外部重设但无起始设为1, 有外部重设和起始设为2。
4. 用所需的当前值载入SMD48 (双字尺寸数值) (载入零可加以清除)。
5. 用所需的预设值载入SMD52 (双字尺寸数值)。
6. 为了捕获当前值等于预设值, 将CV=P中断事件 (事件13) 附加于中断例行程序中, 为中断编程。
7. 为了捕获外部重设事件, 将外部重设中断事件 (事件15) 附加于中断例行程序中, 为中断编程。
8. 执行全局中断启用指令 (ENI), 启用中断。
9. 执行HSC指令, 使S7-200为HSC1编程。
10. 退出子例行程序。

**3、4或5初始化模式**

下列步骤说明如何为带外部方向的单相向上 / 向下计数器 (模式 3、4或5) 初始化HSC1:

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用, 随后的扫描不再调用子例行程序  
蛇丝杉跽偕 枋奔渲葱胁(6)钩缘蚪昂垢 趵砣
2. 在初始化子例行程序中, 根据所需的控制操作载入SMB47。例如:  
SMB47 = 16#F8产生下列结果:  
启用计数器  
写入新当前值  
写入新预设值  
设置HSC初始方向, 向上计数  
将起始和重设输入设为现用水平高
3. 执行HDEF指令, HSC输入设为1, 无外部重设或起始的MODE (模式) 输入设为3, 有外部重设但无起始设为4, 有外部重设和起始设为5。
4. 用所需的当前值载入SMD48 (双字尺寸数值) (载入零可加以清除)。
5. 用所需的预设值载入SMD52 (双字尺寸数值)。
6. 为了捕获当前值等于预设值, 将CV=P中断事件 (事件13) 附加于中断例行程序中, 为中断编程。
7. 为了捕获方向改变, 将方向改变中断事件 (事件14) 附加于中断例行程序中, 为中断编程。
8. 为了捕获外部重设事件, 将外部重设中断事件 (事件15) 附加于中断例行程序中, 为中断编程。
9. 执行全局中断启用指令 (ENI), 启用中断。
10. 执行HSC指令, 使S7-200为HSC1编程。
11. 退出子例行程序。

**6、7或8初始化模式**

下列步骤说明如何为带向上 / 向下时钟的双相向上 / 向下计数器 (模式 6、7或8) 初始化HSC1:

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用, 随后的扫描不再调用子例行程序  
蛇丝杉跽偕 枋奔渲葱胁(6)钩缘蚪昂垢 趵砣
2. 在初始化子例行程序中, 根据所需的控制操作载入SMB47。例如:  
SMB47 = 16#F8产生下列结果:  
启用计数器

写入新当前值  
 写入新预设值  
 设置HSC初始方向，向上计数  
 将起始和重设输入设为现用水平高

3. 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）设为6，有外部重设但无起始设为7，有外部重设和起始设为8。
4. 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD52（双字尺寸数值）。
6. 为了捕获当前值等于预设值，将CV=P中断事件（事件13）附加于中断例行程序中，为中断编程。
7. 为了捕获方向改变，将方向改变中断事件（事件14）附加于中断例行程序中，为中断编程。
8. 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
9. 执行全局中断启用指令（ENI），启用中断。
10. 执行HSC指令，使S7-200为HSC1编程。
11. 退出子例行程序。

### 9、10或11初始化模式

下列步骤说明如何为A/B相正交计数器（模式9、10或11）初始化HSC1：

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序。
2. 在初始化子例行程序中，根据所需的控制操作载入SMB47。

例如（1x计数模式）：

SMB47 = 16#FC产生下列结果：

启用计数器  
 写入新当前值  
 写入新预设值  
 设置HSC初始方向，向上计数  
 将起始和重设输入设为现用水平高

例如（4x计数模式）：

SMB47 = 16#F8产生下列结果：

启用计数器  
 写入新当前值  
 写入新预设值  
 设置初始HSC方向，向上计数  
 将起始和重设输入设为现用水平高

3. 执行HDEF指令，HSC输入设为1，无外部重设或起始的MODE（模式）输入设为9，有外部重设但无起始设为10，有外部重设和起始设为11。
4. 用所需的当前值载入SMD48（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD52（双字尺寸数值）。
6. 为了捕获当前值等于预设值，将CV=P中断事件（事件13）附加于中断例行程序中，为中断编程。
7. 为了捕获方向改变，将方向改变中断事件（事件14）附加于中断例行程序中，为中断编程。
8. 为了捕获外部重设事件，将外部重设中断事件（事件15）附加于中断例行程序中，为中断编程。
9. 执行全局中断启用指令（ENI），启用中断。
10. 执行HSC指令，使S7-200为HSC1编程。
11. 退出子例行程序。

### 12初始化模式

下面步骤描述了如何为计算由PTO0产生的脉冲数而初始化HSC0（模式12）。

1. 使用首次扫描内存位调用执行初始化操作的子例行程序。因为使用子例行程序调用，随后的扫描不再调用子例行程序。
2. 在初始化子例行程序中，根据所需的控制操作载入SMB47。例如：

SMB37 = 16#F8 产生下列结果：

启动计数器  
 写入新当前值  
 写入新预设值  
 将方向设置为向上计数  
 将起始和重设输入设为现用水平高

3. 在将HSC输入设为0且MODE（模式）输入设为12后执行HDEF指令。
4. 用所需的当前值载入SMD38（双字尺寸数值）（载入零可加以清除）。
5. 用所需的预设值载入SMD42（双字尺寸数值）。
6. 为了捕获当前值等于预设值事件，将CV = PV中断事件（事件13）附加于中断例行程序中，从而实现中断编程。如需关于中断处理的全部细节，请参阅讨论“中断指令”的章节。

7. 执行全局中断启用指令（EN1）以启用中断。
8. 执行HSC指令以使S7-200为HSC0编程。
9. 退出子例行程序。

#### 在0、1、2或12模式中改变方向

下列步骤说明如何配置HSC1，使带内部方向的单相计数器（模式0、1、2或12）改变方向：

1. 载入SMB47，写入所需的方向：  
SMB47 = 16#90  
启动计数器  
将HSC方向设为向下计数  
  
SMB47 = 16#98  
启动计数器  
将HSC方向设为向上计数
2. 执行HSC指令，使S7-200为HSC1编程。

#### 载入新当前值（任何模式）

下列步骤说明如何改变HSC1的计数器当前值（任何模式）：  
改变当前值强迫计数器在进行改动的过程中被禁用。计数器被禁用时，则不再计数或生成中断。

1. 载入SMB47，写入所需的当前值：  
SMB47 = 16#C0  
启用计数器  
写入新当前值
2. 用所需的当前值载入SMD48（双字尺寸）（载入零可加以清除）。
3. 执行HSC指令，使S7-200为HSC1编程。

#### 载入新预设值（任何模式）

下列步骤说明如何改变HSC1的计数器预设值（任何模式）：

1. 载入SMB47，写入所需的预设值：  
SMB47 = 16#A0  
启用计数器  
写入新预设值
2. 用所需的预设值载入SMD52（双字尺寸数值）。
3. 执行HSC指令，使S7-200为HSC1编程。

#### 禁用高速计数器（任何模式）

下列步骤说明如何禁用HSC1高速计数器（任何模式）：

1. 载入SMB47，禁用计数器：  
SMB47 = 16#00 禁用计数器
2. 执行HSC指令，禁用计数器。

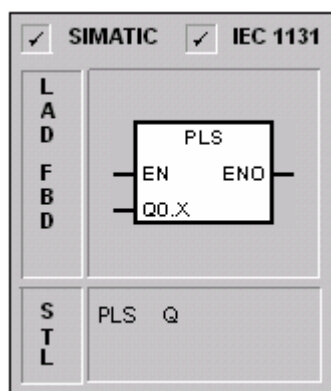
虽然上述操作步骤显示如何逐一改变方向、当前值和预设值，您也可以按照相同顺序，通过以适当方式设置 SMB47数值并执行HSC指令，改变全部数值或其中任何组合。

另请参阅：

[HSC向导](#)  
[HSC指令](#)  
[HDEF指令](#)

### 3.3.12 理解高速输出指令

操作数	数据类型	[[[ENDKEEPN]]]	
Q 常量 (0或1)		字	[[[ENDKEEPN]]]
<a href="#">内存范围</a>	<a href="#">错误</a>	<a href="#">S7-200 CPU指令支持</a>	<a href="#">SIMATIC / 国际助记符</a>
<a href="#">[[[ENDKEEPN]]]</a>			
<a href="#">数据范围</a>	<a href="#">ENO</a>	<a href="#">CPU内存中的指令大小</a>	<a href="#">编址内存</a> [[[ENDKEEPN]]]



**脉冲输出 (PLS)** 指令被用于控制在高速输入 (Q0.0和Q0.1) 中提供的“脉冲链输出” (PTO) 和“脉冲宽度调制” (PWM) 输出。PTO提供矩形波 (工作循环) 输出, 配备循环时间和脉冲数用户控制功能。

PWM提供连续性变量工作循环输出, 配备循环时间和脉冲宽度用户控制功能。脉冲输出范围 Q0.0至Q0.1

[特殊内存PTO / PWM高速输出寄存器](#)

### 理解S7-200高速输出指令

S7-

200有两台PTO/PWM生成器, 建立高速脉冲链或脉冲宽度调节波形。一台生成器指定给数字输出点 Q0.0, 另一台生成器指定给数字输出点Q0.1。一个指定的特殊内存 (SM) 位置为每台生成器存储以下数据: 一个控制字节 (8位数值)、一个脉冲计数值 (一个不带符号的32位数值) 和一个循环时间和脉冲宽度数值 (一个不带符号的16位数值)。

PTO/PWM生成器和进程图像寄存器共用Q0.0和Q0.1。PTO或PWM功能在Q0.0或Q0.1位置现用时, PTO/PWM生成器控制输出, 并禁止输出点的正常使用。输出波形不受进程图像寄存器状态、点强迫数值、执行立即输出指令的影响。PTO/PWM生成器非现用

保持住! 谱 柜 掏枷寞姆嫫螯 = 掏枷寞姆嫫骹韶去涑霾 4 蔚某脚己妥航兆刺 共 4 卧诩呱换虹臀膝 己徒嘎

#### 注释:

- 在启用PTO或PWM操作之前, 将用于Q0.0和Q0.1的进程图像寄存器设为0。
- 所有的控制位、循环时间、脉冲宽度和脉冲计数值的默认值均为0。
- PTO/PWM输出必须至少有10%的额定负载, 才能完成从关闭至打开以及从打开至关闭的顺利转换。
- 文档光盘“提示与技巧”中的提示7、22、23、30和50包含使用PTO/PWM操作PLS指令的程序。

脉冲链 (PTO) 功能提供矩形波 (50%工作循环) 输出或指定的脉冲数和指定的循环时间。脉冲宽度调制 (PWM) 功能提供带变量工作循环的固定循环时间输出。

每台PTO/PWM生成器有一个控制字节 (8位), 一个循环时间数值和脉冲宽度数值 (不带符号的16位数值) 和一个脉冲计数值 (不带符号的32位数值)。这些数值全部存储在特殊内存 (SM) 区域的指定位置。一旦设置这些特殊内存位的位置, 选择所需的值, 葱新此途涑蛸噶睿 PLS) 即启动操作。该指令使S7-200读取SM位置, 并为PTO/PWM生成器编程。

通过修改SM区域中 (包括控制字节) 要求的位置, 您可以更改 PTO或PWM的波形特征, 然后执行PLS指令。

您可以在任意时间向控制字节 (SM67.7或SM77.7) 的PTO/PWM启用位写入零, 禁用PTO或PWM波形的生成, 然后执行PLS指令

**注释:** 所有控制位、循环时间、脉冲宽度和脉冲计数值的默认值均为零。

**注释:** PTO/PWM输出必须至少有10%的额定负载, 才能完成从关闭至打开以及从打开至关闭的顺利转换。

#### PWM操作

PWM功能提供带变量工作循环的固定循环时间输出。可以微秒或毫秒为时基指定循环时间和脉冲宽度。

循环时间的范围从10微秒至65,535微秒, 或从2毫秒至65,535毫秒。

脉冲宽度时间范围从0微秒至65,535微秒或从0毫秒至65,535毫秒。



如上表所示, 设置脉冲宽度等于循环时间 (这使工作循环为 100%) 使输出连续运行。设置脉冲宽度等于0 (这使工作循环为0%) 会关闭输出。

#### 脉冲宽度时间 / 循环时间

#### 反应

脉冲宽度时间  $\geq$  循环时间数值 工作循环为100%: 输出连续运行。

脉冲宽度时间 = 0 工作循环为0%: 输出关闭。

循环时间 < 2个时间单位 循环时间的默认值为两个时间单位。

有两种不同的方法可改变PWM波形的特征: 同步更新和异步更新。

- 同步更新: 如果不要求更改时基, 即可以执行同步更新。执行同步更新时, 波形特征的变化发生在循环边缘, 提供顺



弧

- 异步更新：此为常见的PWM操作，脉冲宽度不同，但循环时间保持不变。因此，不要求时基改变。但是，如果要求改变PTO/PWM生成器的时基，则应使用异步更新。异步更新使PTO/PWM生成器被立即禁用，与PWM波形异步。这样可能造成控制设备状态暂时不稳。由于此一原因，建议使用同步PWM更新。选择可用于所有预计循环时间数值的时基。

控制字节中的PWM更新方法位（SM67.4或SM77.4）指定更新类型，在执行PLC指令时激活改动。请注意，如果时基改变，则会

#### PTO操作

PTO为指定的脉冲数和指定的循环时间提供矩形波（50%工作循环）输出。PTO可提供单脉冲链或多脉冲链（使用脉冲轮廓）。



循环时间范围从10微秒至65,535微秒或从2毫秒至65,535毫秒。

脉冲计数范围从1至4,294,967,295次脉冲。

为循环时间指定基数微秒或毫秒（例如75毫秒）会引起工作循环的失真。

下表定义脉冲计数和循环时间限制：

脉冲计数 / 循环时间	反应
循环时间 < 2个时间单位	循环时间的默认值为2个时间单位。
脉冲计数 = 0	脉冲计数的默认值为1次脉冲。

状态字节（SM66.7或SM76.7）中的PTO空闲位表示编程脉冲链已完成。另外，也可在脉冲链完成时激活中断例行程序。如果您使用多段操作，则在轮廓表完成时立即激活中断例行程序。请参阅以下多段管线连接。

PTO功能允许脉冲链链接或管线作业。现用脉冲链完成时，新的脉冲链输出立即开始。这样就保证了随后的输出脉冲链的连续性。

该管线作业可以两种方式中的一种完成：单段管线作业或多段管线作业。

#### 单段管线作业

在单段管线作业中，您负责更新下一个脉冲链的SM位置。初始PTO段一旦开始，您必须按照对第二个波形的要求立即修改SM位置（00伏沃葱 PLS指令。第二个脉冲链特征被保留在管线中，直至第一个脉冲链完成。管线中每次只能存储一个条目。第一个脉冲链一旦完成，第二个波形输出即开始，管线可用于新的脉冲链规格。您可以重复此步骤，设置下一个脉冲链的特征。

可在脉冲链之间平稳转换，下列情况除外：

- 如果执行时基改动
- 如果现用脉冲链在执行PLS指令捕获到新脉冲链设置之前完成

如果您在管线已满时尝试载入，状态寄存器（SM66.6或SM76.6）中的PTO溢出位被设置。进入RUN（运行）模式时，该位被初始化为0。如果您希望探测随后出现的溢出，则必须在探测到溢出之后以手动方式清除该位。

#### 多段管线作业

在多段管线作业中，S7-200从V内存中的轮廓表自动读取每个脉冲链段的特征。该模式中的SM位置是轮廓表的控制字节、状态字节和起始V内存偏移量（SMW

168或SMW178）。时基可以为微秒或毫秒，但该选项适用于轮廓表中的所有循环时间数值，但在轮廓运行时不得变更。然后可由执行PLS指令开始多段操作。

每段输入的长度均为8个字节，由一个16位循环时间数值、一个16位循环时间Δ数值和一个32位脉冲计数数值组成。

下表说明轮廓表的格式。多段PTO操作的另一个特征是能够通过指定每个脉冲的数量自动增加或减少循环时间。在循环时间Δ域

暗陶 痔嵩皂友 肥奔洌 谱 肥奔涸 卍 虬暗谈褐袴峒跽 肥奔涸H 羰 滴 悖 蛭 肥奔洌槐涸

如果您指定的循环时间Δ数值在一定数量的脉冲后导致非法循环时间，则会出现数学溢出条件。PTO功能被终止，输出转换成图

蠹拇嫫甯 司啤 4 送 猓 刺 纸冢 SM66.4或SM76.4)中的Δ计算错误位被设为一。

如果您以手动方式异常中止正在运行的PTO轮廓，状态字节（SM66.5或SM76.5）中的用户异常中止位则被设为一。

运行PTO轮廓时，SMB166（或SMB176）中提供当前现用段数。

#### 多段PTO操作的轮廓表格式

距离轮廓表起始位置的 字节偏移量	轮廓段数目	表格条目说明
0		段数（1至255）；数值0生成非严重错误，生成无PTO输出
1	#1	初始循环时间（2至65535个时基单位）
3		每次脉冲的循环时间Δ（带符号的数值）（-32768至32767个时基单位）
5		脉冲计数（1至4294967295）
9	#2	初始循环时间（2至65535个时基单位）
11		每个脉冲的循环时间Δ（带符号的数值）（-32768至32767个时基单位）
13		脉冲计数（1至4294967295）
::	#3	::

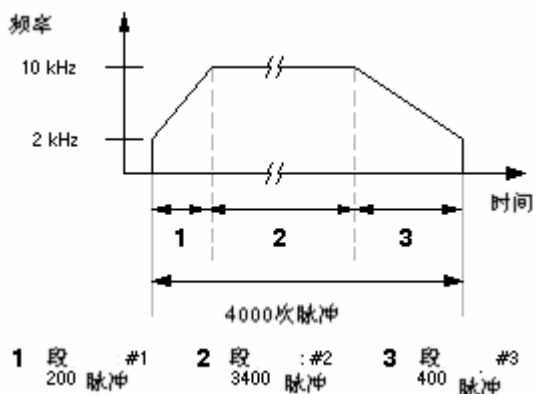
### 计算轮廓表数值

PTO/PWM生成器的多段管线作业功能在许多应用程序中都很有用，特别是步阶器电机控制中。

例如，您可以通过简单斜坡向上、运行和斜坡向下顺序或更复杂的顺序使用配备脉冲轮廓的 PTO，控制步进器马达，方法是定义恒速啮啮 255个段的脉冲轮廓，每个段与一个斜坡向上、运行或斜坡向下操作相对应。

下图显示生成加速步进器马达（#1段）、按恒速操作马达（#2段）、随后减低马达速度（#3段）的输出波形所要求的轮廓表数值样本。

### 简单步阶器电机应用程序的频率与时间图形举例



在本例中：起始和终止脉冲频率为 2 kHz，最大脉冲频率为 10 kHz

，要求 4000 次脉冲才能达到所需的马达转动次数。因为用阶段（循环时间）表示轮廓表数值，而不使用频率，需要将给定频率数值转换成循环时间数值。因此，起始（最初）和终止（结束）循环时间为 500 ms，与最大频率对应的循环时间为 100 ms。在输出轮廓的加速部分，应在约 400 次脉冲时达到最大脉冲频率。轮廓减速部分应在约 400 次脉冲时完成。

您可以用以下公式确定 PTO/PWM 生成器用于调节某一特定段每次脉冲循环时间的  $\Delta$  循环时间数值：

$$\text{段的 } \Delta \text{ 周期时间} = \left| \text{End\_CT}_{\text{seg}} - \text{Init\_CT}_{\text{seg}} \right| / \text{Quantity}_{\text{seg}}$$

位置：  
 $\text{End\_CT}_{\text{seg}}$  = 该段的结束周期时间  
 $\text{Init\_CT}_{\text{seg}}$  = 该段的初始周期时间  
 $\text{Quantity}_{\text{seg}}$  = 该段的脉冲数

利用该公式，计算出的加速部分（或 #1 段）的  $\Delta$  循环时间为 -

2。与此相似，减速部分（或 #3 段）的  $\Delta$  循环时间为 1。因为 #2 段是输出波形的恒速部分，该段的  $\Delta$  循环时间为零。

假定轮廓表位于从 V500 开始的 V 内存中，以下显示用于生成所需波形的表值。您可以在程序中包含指令，将这些数值载入 V 内存。

### 轮廓表数值举例

V 内存地址	数值	说明
VB500	总段数	
VW501	初始循环时间	#1 段
VW503 -2	初始 $\Delta$ 循环时间	
VD505	脉冲数	
VW509	初始循环时间	#2 段
VW511	$\Delta$ 循环时间	
VD513	脉冲数	
VW517	初始循环时间	#3 段
VW519	$\Delta$ 循环时间	
VD521	脉冲数	

利用程序中的指令可将这些表值置于 V 内存中。另一种方法是定义数据块中的轮廓值。本标题结尾部分举例说明使用多段 PTO 操作某条轴。

为了确定波形段之间的转换是否可接受，您需要确定段中最后一次脉冲的循环时间。除非  $\Delta$  循环时间是 0，您必须计算段最后一

段的最后一次脉冲周期时间  $= \text{Init\_CT}_{\text{seg}} + (\text{Delta}_{\text{seg}} * (\text{Quantity}_{\text{seg}} - 1))$

位置 :  $\text{Init\_CT}_{\text{seg}} =$  该段的结束周期时间

$\text{Delta}_{\text{seg}} =$  该段的初始周期时间

$\text{Quantity}_{\text{seg}} =$  该段的脉冲数

上例是简化的情况，用于介绍目的，实际应用程序可能要求更复杂的波形轮廓。请记住：

- $\Delta$  循环时间只能指定为整数微秒或毫秒
- 可对每次脉冲执行循环时间修改

这两个项目产生的效果是计算某个特定段的  $\Delta$  循环时间数值可能要求一个循环方案。计算某个特定段的结束循环时间或脉冲数目时可能要求一定的灵活性。

特定轮廓段期限对确定正确的轮廓表数值程序有用。可利用以下公式计算完成特定轮廓段的时间长度：

段时间长度  $= \text{Quantity}_{\text{seg}} * (\text{Init\_CT} + ((\text{Delta}_{\text{seg}}/2) * (\text{Quantity}_{\text{seg}} - 1)))$

位置 :  $\text{Quantity}_{\text{seg}} =$  该段的结束周期时间

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

$\text{Delta}_{\text{seg}} =$  该段的脉冲数

#### PTO/PWM控制寄存器

PLS指令读取存储在指定的SM内存位置的数据，并以此为PTO/PWM生成器编程。SMB67控制PTO 0或PWM 0，SMB77控制PTO 1或PWM 1。PTO/

PWM控制寄存器表描述用于控制PTO/PWM操作的寄存器。您可以将下表用作快速参考，帮助确定放置在PTO/PWM控制寄存器中用于激活所需操作的数值。

您可以改变PTO或PWM波形的特征，方法是修改SM区（包括控制字节）中的位置，然后执行PLS指令。您可以在任何时间禁用PTO或PWM波形的生成，方法是向控制字节（SM67.7或SM77.7）的PTO/PWM启用位写入0，然后执行PLS指令。

状态字节中的PTO空闲位（SM66.7或SM76.7）表示编程脉冲链已完成。此外，可在脉冲链完成时激活中断例行程序。（请参阅中断指令说明和“通讯”指令。）如果您在使用多段操作，在轮廓表完成时激活中断例行程序。

以下条件设置SM66.4（或SM76.4）和SM66.5（或SM76.5）：

- 指定一个在数次脉冲后导致非法循环时间的循环时间  $\Delta$  数值生成一个数学溢出条件，该条件会终止PTO功能，并将“ $\Delta$  计算错误”位（SM66.4或SM76.4）设为1。输出回复为图像寄存器控制。
- 以手动方式异常中止（禁用）正在执行的PTO轮廓会将“用户异常中止”位（SM66.5或SM76.5）设为1。
- 尝试在管线已满的情况下载入会将PTO溢出位（SM66.6或SM76.6）设为1。如果您希望检测随后的溢出，您必须在检测到溢出后以手动方式清除该位。转换至RUN（运行）模式可将该位初始化为0。

**注释：**当您载入新脉冲计数（SMD72或SMD82）、脉冲宽度（SMW70或SMW80）或循环时间（SMW68或SMW78）时，在执行PLS指令之前，还需要在控制寄存器中设置适当的更新位。对于多段脉冲链操作，在执行PLS指令之前，您还必须载入轮廓表的起始偏移量（SMW168或SMW178）和轮廓表数值。

#### PTO/PWM控制寄存器

Q0.0	Q0.1	状态位		
SM66.4		SM76.4	PTO轮廓由于 $\Delta$ 计算错误异常中止	
		0 = 无错；	1 = 异常中止	
SM66.5		SM76.5	PTO轮廓由于用户命令异常中止	
		0 = 无错；	1 = 异常中止	
SM66.6		SM76.6	PTO管线溢出 / 下溢	0 =
无溢出；		1 = 溢出 / 下溢		
SM66.7		SM76.7	PTO空闲	0 =
进行中；		1 = PTO空闲		
Q0.0	Q0.1	控制位		
SM67.0		SM77.0	PTO/PWM更新循环时间值	0 =
无更新；		1 = 更新循环时间		
SM67.1		SM77.1	PWM更新脉冲宽度时间值	0 =
无更新；		1 = 更新脉冲宽度		
SM67.2		SM77.2	PTO更新脉冲计数值	0 =
无更新；		1 = 更新脉冲计数		
SM67.3		SM77.3	PTO/PWM时基选择	0 = 1
祗/tick；		1 = 1ms/tick		
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启用	0 = 禁用PTO/PWM； 1

= 启用PTO/PWM

Q0.0	Q0.1	其他PTO/PWM寄存器	
SMW68 65535)		SMW78	PTO/PWM循环时间值 (范围: 2至
SMW70		SMW80	PWM脉冲宽度值 (范围: 0至65535)
SMD72		SMD82	
PTO脉冲计数值 (范围: 1至4294967295)			
SMB166		SMB176	进行中的段数 (仅用于多段PTO操作)
SMW168		SMW178	
轮廓表起始位置, 用距离V0的字节偏移量表示 (仅用于多段PTO操作)			
SMB170		SMB180	线性轮廓状态字节
SMB171		SMB181	线性轮廓结果寄存器
SMB172		SMB182	手动模式频率寄存器

**PTO/PWM控制字节参考**

控制寄存器 (十六进制数值)	启用 脉冲计数	执行PLS指令的结果 选择模式 脉冲宽度	PTO段操作 循环时间	PWM更新方法
时基 16#81	是	PTO 载入	单段	衽 / 循环
16#84	是	PTO	单段	衽 / 循环
载入 16#85	是	PTO 载入	单段	衽 / 循环
载入 16#89	是	PTO 载入	单段	ms / 循环
16#8C	是	PTO	单段	ms / 循环
载入 16#8D	是	PTO 载入	单段	ms / 循环
16#A0	是	PTO	多段	衽 / 循环
16#A8	是	PTO	多段	ms / 循环
16#D1	是	PWM 载入		同步 衽 / 循环
16#D2	是 载入	PWM		同步 衽 / 循环
16#D3	是 载入	PWM 载入		同步 衽 / 循环
16#D9	是	PWM 载入		同步 ms / 循环
16#DA	是 载入	PWM		同步 ms / 循环
16#DB	是 载入	PWM 载入		同步 ms / 循环

**PTO/PWM初始化和操作顺序**

以下是初始化和操作顺序说明, 能够帮助您更好地理解 PTO和PWM功能操作。在整个顺序说明过程中一直使用脉冲输出 Q0.0。初始化说明假定S7-200刚置入RUN (运行) 模式, 因此首次扫描内存位为真实。如果不是如此或者如果必须对 PTO/PWM功能重新初始化, 您可以利用除首次扫描内存位之外的一个条件调用初始化例行程序。

**PWM初始化**

以下PWM初始化和操作顺序说明建议使用“首次扫描”位 (SM0.1) 初始化脉冲输出。使用“首次扫描”位调用初始化子例行程序可降低扫描时间, 因为随后的扫描无须调用该子例行程序。(仅需在转换为 RUN (运行) 模式后的首次扫描时设置“首次扫描”位。)但是, 您的应用程序可能有其他限制, 要求您初始化 (或重新初始化) 脉冲输出。在此种情况下, 您可以使用另一个条件调用初始化例行程序。

通常, 您用一个子例行程序为脉冲输出初始化PWM。您从主程序调用初始化子例行程序。使用首次扫描内存位 (SM0.1) 将脉冲输出初始化为0, 并调用子例行程序, 执行初始化操作。当您使用子例行程序调用时, 随后的扫描不再调用该子例行程序, 这样

从主程序建立初始化子例行程序调用后, 用以下步骤建立控制逻辑, 用于在初始化子例行程序中配置脉冲输出 Q0.0:

1. 通过将以下一个数值载入SMB67: 16#D3 (选择微秒递增) 或16#DB (选择毫秒递增) 的方法配置控制字节。  
两个数值均可启用PTO/PWM功能、选择PWM操作、设置更新脉冲宽度和循环时间数值、以及选择时基 (微秒或毫秒)。
2. 在SMW68中载入一个循环时间的字尺寸数值。
3. 在SMW70中载入脉冲宽度的字尺寸数值。
4. 执行PLS指令 (以便S7-200为PTO/PWM生成器编程)。
5. 欲为随后的脉冲宽度变化预载一个新控制字节数值 (选项), 在 SMB67: 16#D2 (微秒) 或16#DA (毫秒) 中载入下列数值之一。
6. 退出子例行程序。

**为PWM输出更改脉冲宽度**

如果您用16#D2或16#DA预载SMB67（请参阅以上第5步），您可以使用一个将脉冲宽度改变为脉冲输出（Q0.0）的子例行程序。建立对该子例行程序的调用后，使用以下步骤建立改变脉冲宽度的控制逻辑：

1. 在SMW70中载入新脉冲宽度的字尺寸数值。
2. 执行PLS指令，使S7-200为PTO/PWM生成器编程。
3. 退出子例行程序。

**PTO初始化—单段操作**

以下PWM初始化和操作顺序说明建议使用“首次扫描”位（SM0.1）初始化脉冲输出。使用“首次扫描”位调用初始化子例行程序可降低扫描时间，因为随后的扫描无须调用该子例行程序。（仅需在转换为RUN（运行）模式后的首次扫描时设置“首次扫描”位。）但是，您的应用程序可能有其他限制，要求您初始化（或重新初始化）脉冲输出。在此种情况下，您可以使用另一个条件调用初始化例行程序。

通常，您用一个子例行程序为脉冲输出初始化PWM。您从主程序调用初始化子例行程序。使用首次扫描内存位（SM0.1）将脉冲输出初始化为0，并调用子例行程序，执行初始化操作。当您使用子例行程序调用时，随后的扫描不再调用该子例行程序，这样

从主程序建立初始化子例行程序调用后，用以下步骤建立控制逻辑，用于在初始化子例行程序中配置脉冲输出 Q0.0:

1. 通过将以下一个数值载入SMB67: 16#85（选择微秒递增）或16#8D（选择毫秒递增）的方法配置控制字节。
2. 两个数值均可启用PTO/PWM功能、选择PWM操作、设置更新脉冲宽度和循环时间数值、以及选择时基（微秒或毫秒）。

在SMW68中载入一个循环时间的字尺寸数值。

3. 在SMD72中载入脉冲计数的双字尺寸数值。
4. （选项）如果您希望在脉冲链输出完成后立即执行相关功能，您可以将脉冲链完成事件（中断类别 19）附加于中断子谱缘颞 枋奔渲葱校 (8)峁┘峁垢 辖鞅某缘颞 卸媳嗜蹇 褂 ATCH指令并执行全局中断启用指令ENI。
5. 执行PLS指令，使S7-200为PTO/PWM生成器编程。
6. 退出子例行程序。

**改变PTO循环时间—单段操作**

对于单段PTO操作，您可以使用中断例行程序或子例行程序改变循环时间。欲使用单段PTO操作更改中断例行程序或子例行程序机械PTO循环时间，请遵循下列步骤：

- 设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基、设置更新循环时间数值），方法是在SMB67: 16#81（用于微秒）或16#89（用于毫秒）中载入下列一个数值。
- 在SMW68中，载入新循环时间的一个字尺寸数值。
- 执行PLS指令，使S7-200为PTO/PWM生成器编程。更新脉冲计数波形输出开始之前，CPU必须完成所有进行中的PTO。
- 退出中断例行程序或子例行程序。

**改变PTO脉冲计数—单段操作**

对于单段PTO操作，您可以使用中断例行程序或子例行程序改变脉冲计数。欲使用单段PTO操作在中断例行程序或子例行程序中改变PTO脉冲计数，请遵循下列步骤：

1. 设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基、设置更新循环时间数值），方法是在SMB67: 16#84（用于微秒）或16#8C（用于毫秒）中载入以下两个数值之一。
2. 在SMD72中，载入新脉冲计数的一个字尺寸数值。
3. 执行PLS指令（以便S7-200为PTO/PWM生成器编程）。开始用更新脉冲计数生成波形之前，S7-200完成所有进行中的PTO。
4. 退出中断例行程序或子例行程序。

**改变PTO循环时间和脉冲计数—单段操作**

对于单段PTO操作，您可以使用中断例行程序或子例行程序改变循环时间和脉冲计数。欲使用单段PTO操作更改中断例行程序或永谱缘颞械 PTO循环时间和脉冲计数，请遵循下列步骤：

1. 设置控制字节（启用PTO/PWM功能、选择PTO操作、选择时基、设置更新循环时间和脉冲计数数值），方法是在SMB67: 16#85（用于微秒）或16#8D（用于毫秒）中载入以下两个数值之一。
2. 在SMW68中，载入新循环时间的一个字尺寸数值。
3. 在SMC72中，载入新脉冲计数的一个字尺寸数值。
4. 执行PLS指令，使S7-200为PTO/PWM生成器编程。用更新脉冲计数和脉冲时间波形输出开始之前，CPU必须完成所有进行中的PTO。
5. 退出中断例行程序或子例行程序。

**PTO初始化—多段操作**

通常，您用一个子例行程序为多段操作的脉冲输出配置和初始化PTO。您从主程序调用初始化子例行程序。使用首次扫描内存位（SM0.1）将PTO使用的输出初始化为0，并调用子例行程序，执行初始化操作。当您使用“首次扫描”位调用初始化子例行程序保 婧蟪纳 璨辉俚饕酶米永 谱缘颞 皮 峁垢 枋奔渲葱小

从主程序建立对初始化例行程序的调用后，使用以下步骤建立控制逻辑，用于在初始化子例行程序中配置脉冲输出 Q0.0: 使用首次扫描内存位（SM0.1）将输出初始化为0，并调用您所需的子例行程序，执行初始化操作。这样会降低扫描时间执行，并提供结构更严谨的程序。

1. 通过将以下一个数值载入SMB67: 16#A0（选择微秒递增）或16#A8（选择毫秒递增）的方法配置控制字节。

- 两个数值均可启用PTO/PWM功能、选择PTO操作、选择多段操作、以及选择时基（微秒或毫秒）。
2. 在SMW168中载入一个字尺寸数值，用作轮廓表起始V内存偏移量。
  3. 使用V内存存在轮廓表中设置段值。确保“段数”域（表的第一个字节）正确无误。
  4. （选项）如果您希望在PTO轮廓完成后立即执行相关功能，您可以将脉冲链完成事件（中断类别 19）附加在中断子例行程序中，为中断编程。使用ATCH执行全局中断启用指令ENI。
  5. 执行PLS指令，使S7-200为PTO/PWM生成器编程。
  6. 退出子例行程序。

```

NETWORK 1 // 用于PWM范例的主程序
// 首次扫描时，将图像寄存器位设为低，并调用 SBR_0
LD SMO.1
R Q0.1 1
CALL SBR_0

```

```

NETWORK 2 // 设置程序中其他位置的MO.0，将脉冲宽度更改为50%工作循环
LD MO.0
EU
CALL SBR_1

```

```

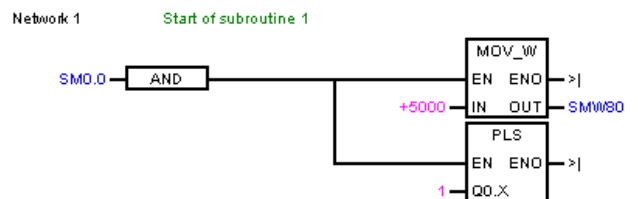
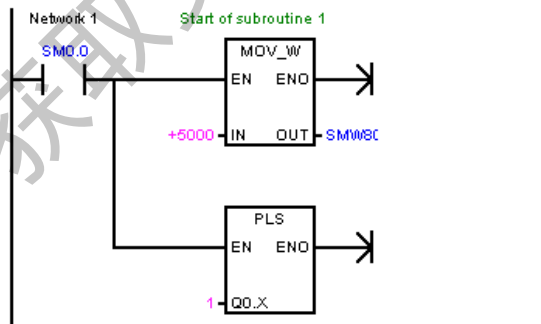
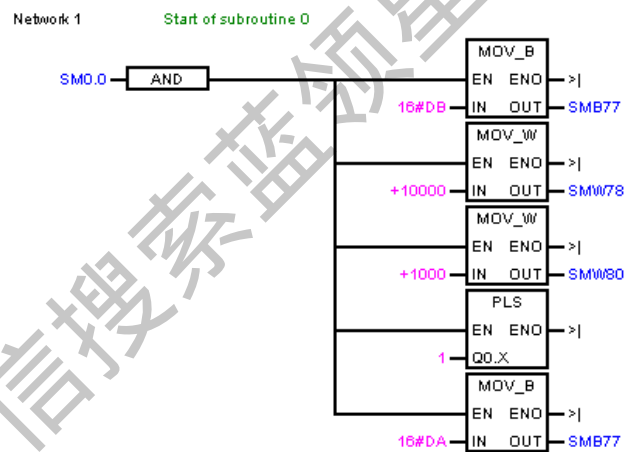
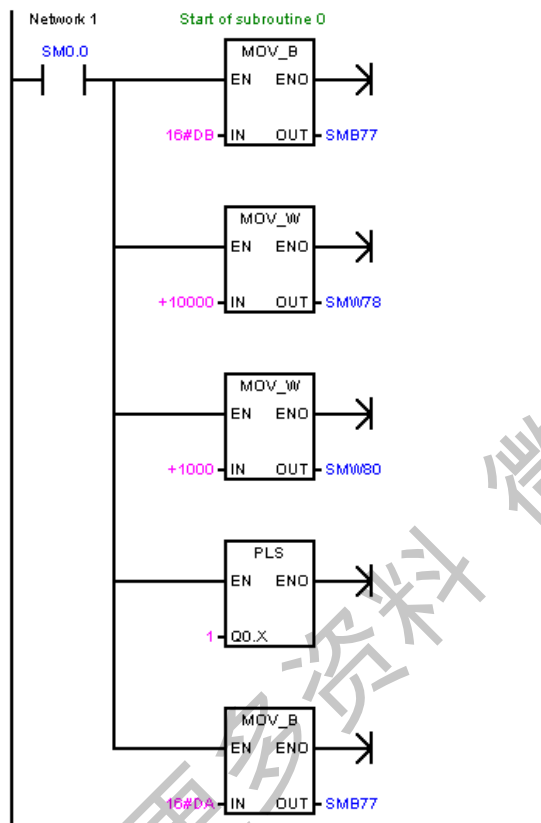
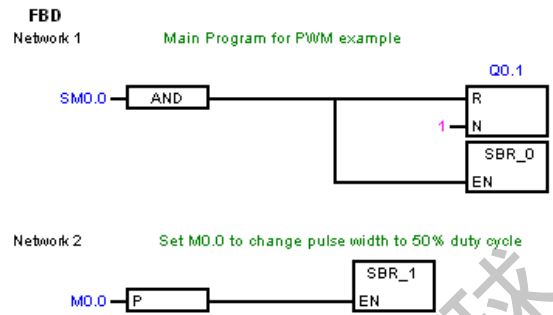
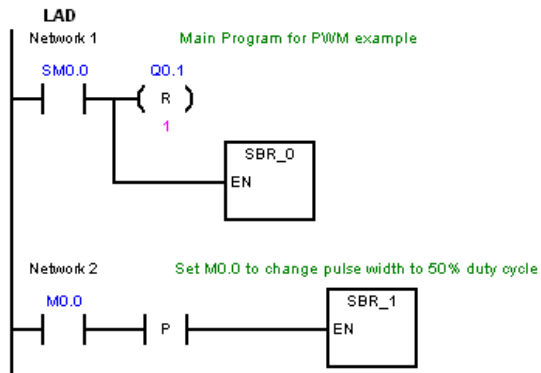
NETWORK 1 // 子例行程序0开始
LD SMO.0
MOVB 16#DB SMB77 // 设置控制字节
// - 选择PWM操作
// - 选择毫秒递增和同步更新
// - 设置脉冲宽度和循环时间数值
// - 启用PWM功能
MOVW +10000 SMW78 // 将循环时间设为10,000毫秒
MOVW +1000 SMW80 // 将脉冲宽度设为1,000毫秒
PLS 1 // 激活PWM操作：PLS1=>Q0.1
MOVB 16#DA SMB77 // 重新载入控制字节，用于随后的脉冲宽度改动

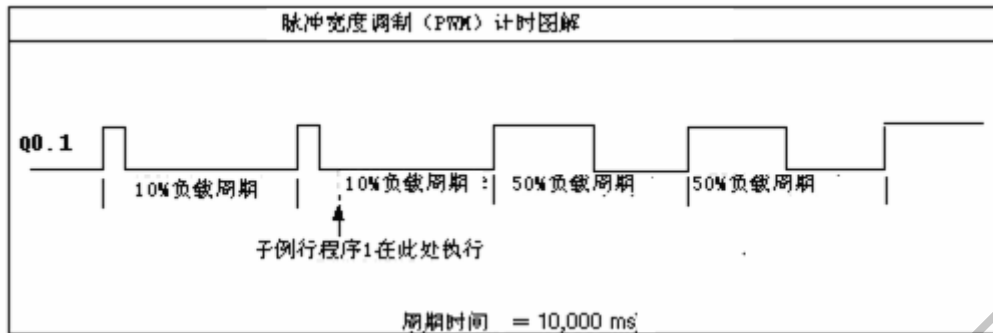
```

```

NETWORK 1 // 子例行程序1开始
LD SMO.0
MOVW +5000 SMW80 // 将脉冲宽度设为5000毫秒
PLS 1 // 断言脉冲宽度改动

```





### 使用单段操作的脉冲链输出举例

```

NETWORK 1 // 用于单段脉冲链操作的主程序 (PTO)
// 首次扫描时, 将图像寄存器位设为低
// 并调用子例行程序0
LD SMO. 1
R Q0. 0 1
CALL SBR_0

```

```

NETWORK 1 // 子例行程序0开始
LD SMO. 0
MOVB 16#8D SMB67 // 设置控制字节:
// - 选择PTO操作
// - 选择单段操作
// - 选择毫秒递增
// - 设置脉冲计数和循环时间数值
// - 启用PTO功能
MOVW +500 SMD72 // 将循环时间设为500毫秒。
MOVB +4 SMD72 // 将脉冲计数设为4次脉冲。
ATCH INT_0 19 // 将中断例行程序0定义为
// 处理PTO完成中断的中断。
ENI // 全局中断启用
PLS 0 // 激活PTO操作, PLS0 => Q0. 0
MOVB 16#89 SMB67 // 预载控制字节, 用于随后的
// 循环时间改动。

```

```

NETWORK 1 // 中断0开始
// 如果当前循环时间为500毫秒:
// 将循环时间设为1000毫秒, 并生成4次脉冲
LDW= SMD72 +500
MOVW +1000 SMD72
PLS 0
CRETI

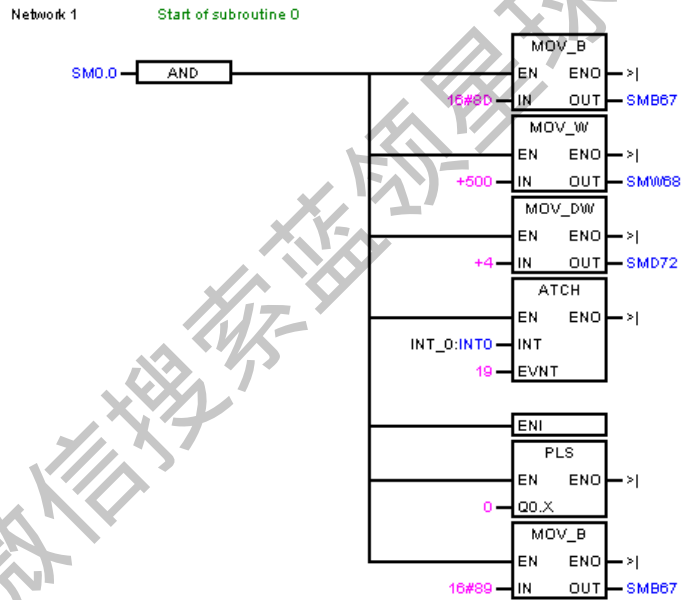
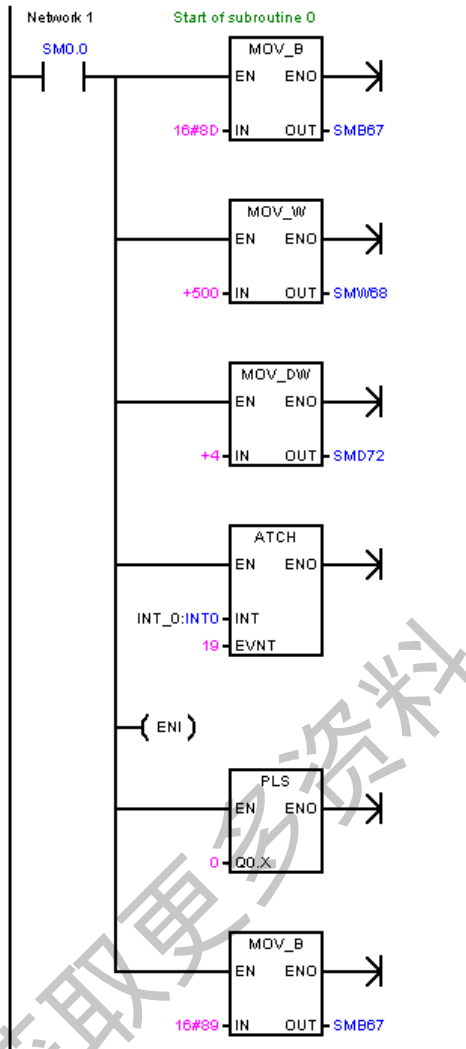
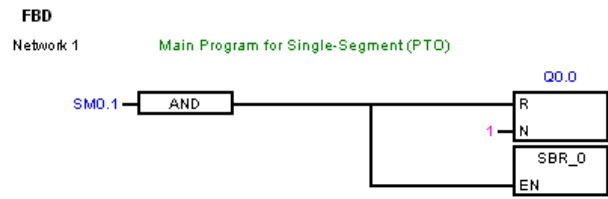
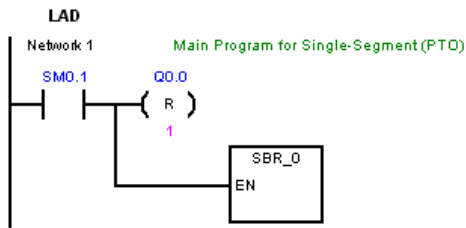
```

```

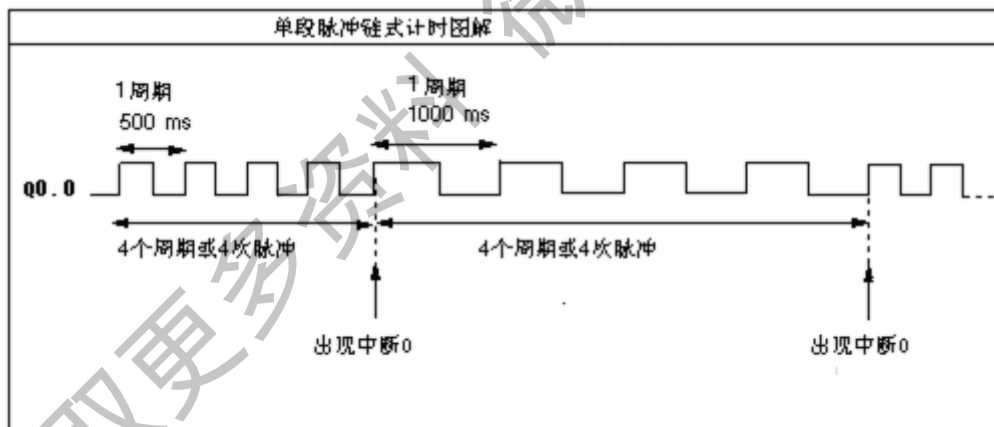
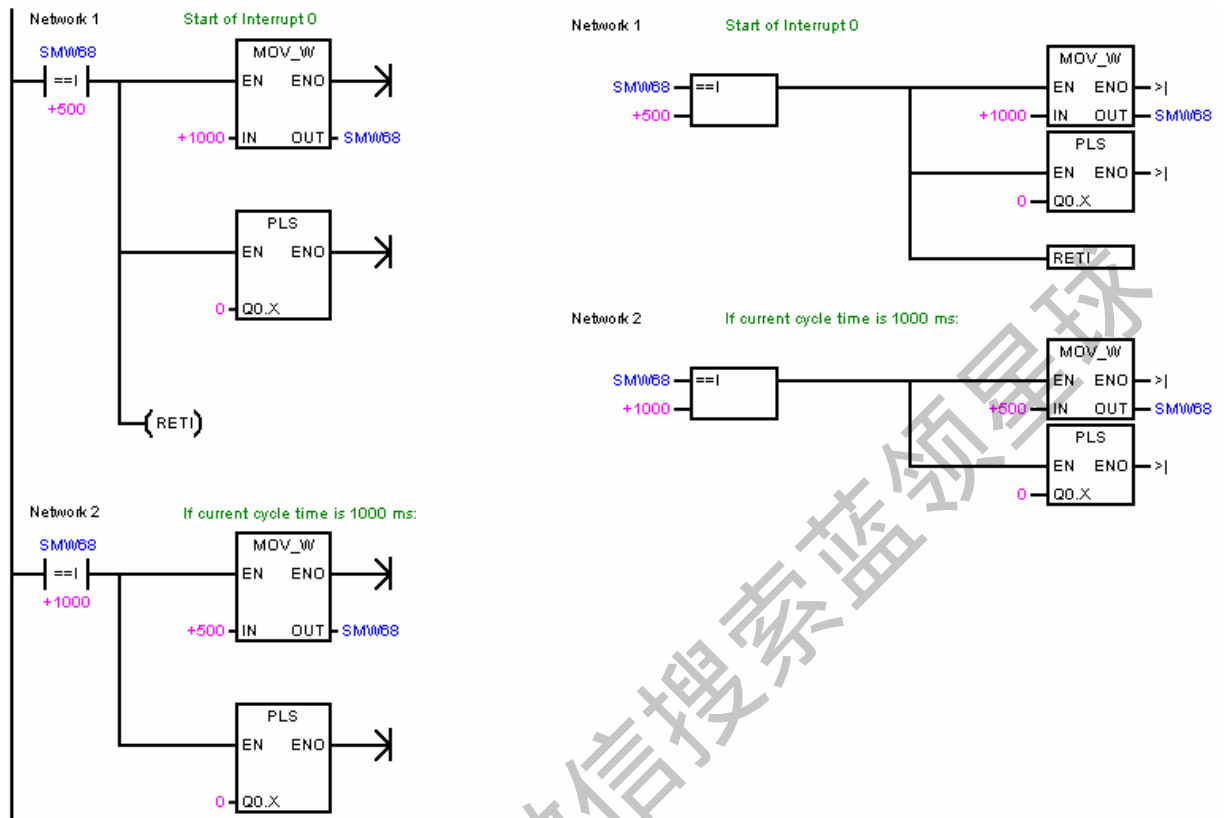
NETWORK 2
// 如果当前循环时间为1000毫秒:
// 将循环时间设为500毫秒, 并生成4次脉冲
LDW= SMD72 +1000
MOVW +500 SMD72
PLS 0

```





获取更多资料 微信搜索 蓝领星球



### 使用多段操作的脉冲链输出举例

```

NETWORK 1 // 用于多段脉冲链操作的主程序 (PTO)
// 首次扫描时, 将图像寄存器位设为低
// 并调用子例行程序0
LD SMO. 1
R Q0. 0 1
CALL SBR_0

```

```

NETWORK 1 // 子例行程序0开始
// 预载PTO轮廓表
LD SMO. 0

```

```

MOVB 3 VB500 // 将轮廓表段数设为3
// 段1:
MOVW +500 VW501 // 将段1的初始循环时间设为500毫秒
MOVW -2 VW503 // 将段1的Δ循环时间设为-2毫秒
MOVD +200 VD505 // 将段1中的脉冲数设为200
// 段2:
MOVW +100 VW509 // 将段2的初始循环时间设为100毫秒
MOVW +0 VW511 // 将段2的Δ循环时间设为0毫秒
MOVD +3400 VD513 // 将段2中的脉冲数设为3400
// 段3:
MOVW +100 VW517 // 将段3的初始循环时间设为100毫秒
MOVW +1 VW519 // 将段3的Δ循环时间设为1毫秒
MOVD +400 VD521 // 将段3中的脉冲数设为400

```

```

NETWORK 2
LD SMO.0
MOVB 16#A8 SMB67 // 设置控制字节:
// - 选择PTO操作
// - 选择多段操作
// - 选择毫秒递增
// - 启用PTO功能
MOVW +500 SMW168 // 将轮廓表的该起始地址
// 指定为V500。
ATCH INT_0 19 // 将中断例行程序0定义为
// 处理PTO完成中断的中断。
ENI // 全局中断启用
PLS 0 // 激活PTO操作, PLS0 => Q0.0
MOVB 16#89 SMB67 // 预载控制字节, 用于随后的
// 循环时间改动。

```

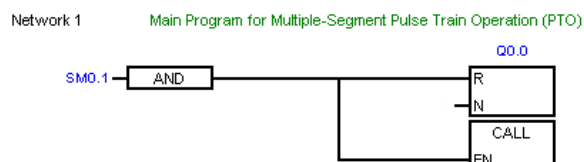
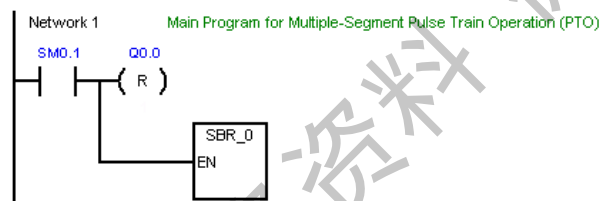
```

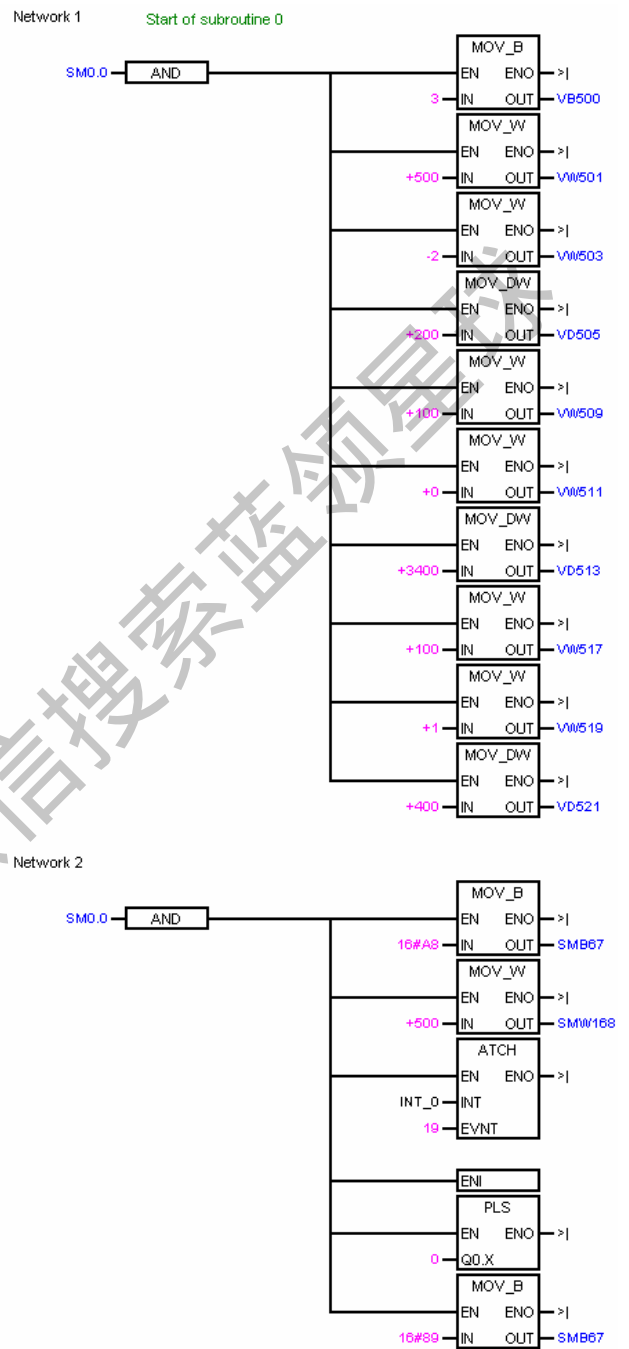
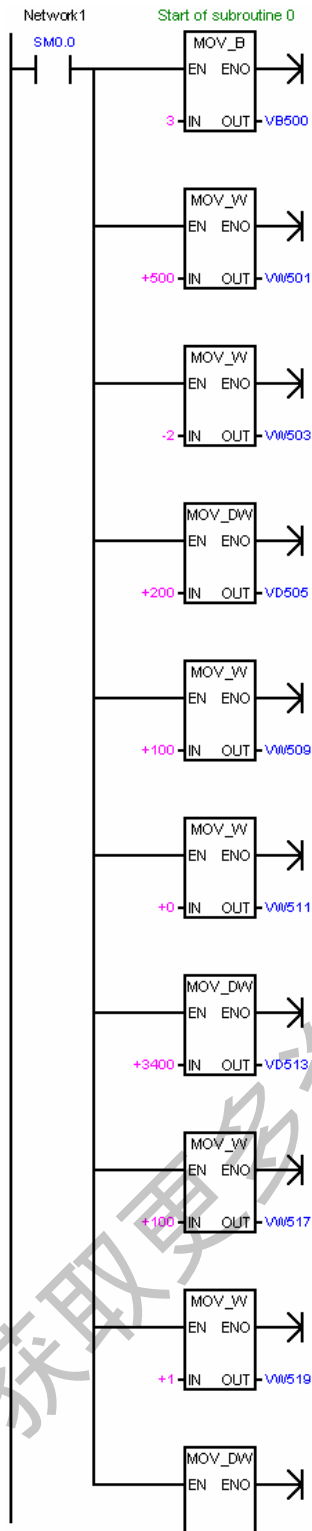
NETWORK 1 // 中断0开始
// PTO输出轮廓完成时, 打开输出Q0.5
LD SMO.0
= Q0.5

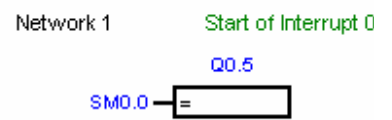
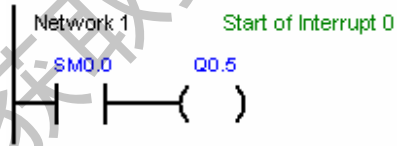
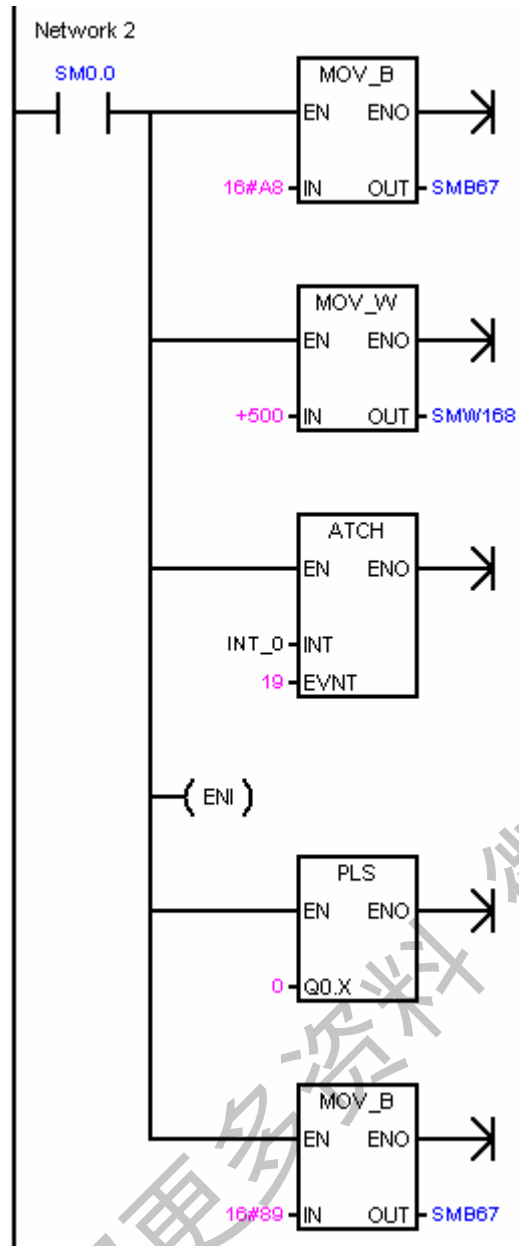
```

LAD

FBD







### 3.3.13 子例行程序编程

**调用指令**      `[[[ENDKEEPN]]]`

- 本标题讨论下列主题:
- [使用子例行程序](#)
- [如何建立子例行程序](#)
- [如何终止子例行程序](#)
- [如何调用子例行程序](#)
- [嵌套和递归](#)

## 使用子例行程序

(返回顶端)

子例行程序帮助您对程序进行分块。主程序中使用的指令决定具体子例行程序的执行状况。当主程序调用子例行程序并执行时，子例行程序执行全部指令直至结束。然后，系统将控制返回至调用子例行程序网络中的主程序。子例行程序用于为程序分段和分块，使其成为较小的、更易管理的块。在程序中调试和维护时，您可以利用这项优势。通过使用较小的程序块，对这些区域和整个程序简单地进行调试和排除故障。只在需要时才调用程序块，可以更有效地使用 PLC，因为所械某缘蚩榭瞻芪才脞葱忻看紊琛最后，如果子例行程序仅引用参数和局部内存，则可移动子例行程序。为了移动子例行程序，应避免使用任何全局变量 / 符号（I、Q、M、SM、AI、AQ、V、T、C、S、AC内存中的绝对地址）。如果子例行程序无调用参数（IN、OUT或IN\_OUT）或仅在LAD、FBD或STL中使用子例行程序，必须执行下列三项任务：

- 建立子例行程序
- 在子例行程序局部变量表中定义参数（如果有）
- 从适当的POU（从主程序或另一个子例行程序）调用子例行程序

当子例行程序被调用时，整个逻辑堆栈被保存，堆栈顶端被设为一，所有其他堆栈位置被设为零，控制被传输至调用子例行程序。子例行程序和调用例行程序共用累加器。由于子例行程序的使用，对累加器不执行保存或恢复操作。

### 用参数调用子例行程序

子例行程序可能包含交接的参数。参数在子例行程序的局部变量表中定义。参数必须有一个符号名（最多为 23 个字符）、一个变量类型和一个数据类型。可向子例行程序交接 16 个参数或从子例行程序交接 16 个参数。局部变量表中的变量类型域定义参数是否交接至子例行程序（IN）、交接至或交接出子例行程序（IN\_OUT）或交接出子例行程序（OUT）。下表说明子例行程序的参数类型。欲增加参数条目，将光标放在您希望增加的类型的变量类型域上（IN、IN\_OUT或OUT）。单击鼠标右键，获得选项菜单。选择“插入”选项，然后选择“下一行”选项。在当前条目的下方会显示所选类型的另一个参数条目。

### 调用参数类型

### 说明

IN	参数被交接至子例行程序。如果参数是直接地址（例如 VB10），在指定位置的数值被交接至子例行程序。如果参数是间接地址，（例如 *AC1），位于指向位置的数值被交接至子例行程序。如果参数是数据常量（16#1234）或地址（&VB100），常量或地址数值被交接至子例行程序。
IN_OUT	位于指定参数位置的数值被交接至子例行程序，来自子例行程序的结果数值被返回至相同的位置。输入 / 输出参数不允许使用常量（例如 16#1234）和地址（例如 &VB100）。
OUT	来自子例行程序的结果数值被返回至指定的参数位置。常量（例如 16#1234）和地址（例如 &VB100）不允许用作输出。
TEMP	未用作交接参数的任何本地内存不得用于子例行程序中的临时存储。

局部变量表中的数据类型域定义参数的大小和格式。下表列出了参数类型。

### 参数数据类型

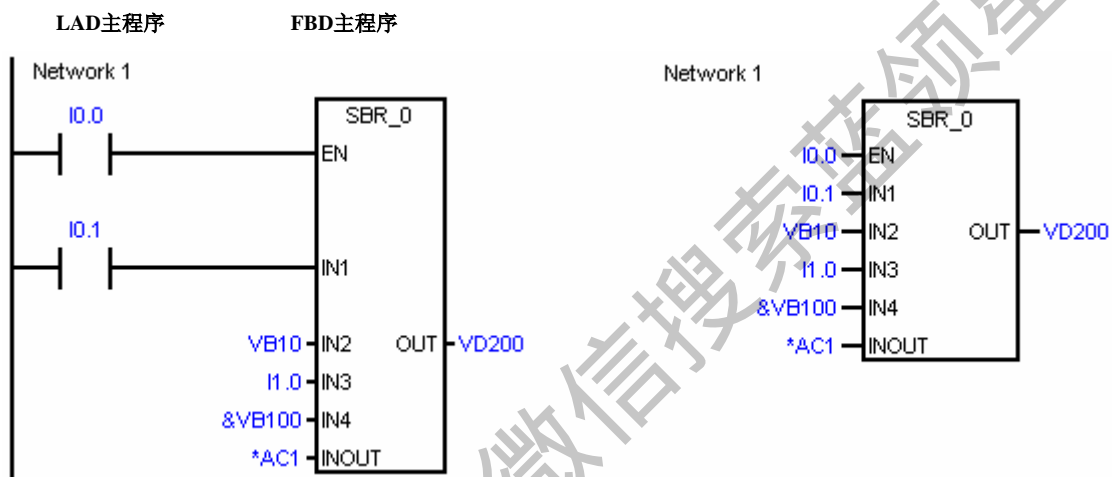
### 说明

功率流	布尔功率流仅限于位（布尔）输入。该说明通知 STEP 7-Micro/WIN 该输入参数是依据位逻辑指令组合的功率流结果。布尔功率流输入必须首先在局部变量表中任何其他类型输入之前显示。只有输入参数允许这样使用。下例中的启用输入（EN）和 IN1 输入使用布尔逻辑。
布尔	该数据类型用于单位输入和输出。下例中的 IN3 是布尔输入。
字节、字、双字	这些数据类型分别识别 1、2 或 4 个字节不带符号的输入或输出参数。
整数、双整数	这些数据类型分别识别 2 或 4 个字节带符号的输入或输出参数。
实数	该数据类型识别单精度（4 个字节）IEEE 浮点数值。
字符串	此数据类型被用作字符串的四字节指针
功率流	布尔功率流只可用于位（布尔）位。此说明告诉 STEP 7-Micro/WIN，这个输入参数是基于某位逻辑指令组合的功率流结果。布尔功率流输入必须在局部变量表中最先出现，早于任何其他类型的输入。唯有输入参数可以此方式使用。下例中的启用输入（EN）输入和 IN1 输入使用了布尔逻辑。

### 子例行程序调用举例

#### 用于 SBR\_0 的局部变量表

	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
LB1	IN2	IN	BYTE	
L2.0	IN3	IN	BOOL	
LD3	IN4	IN	DWORD	
		IN		
LD7	INOUT	IN_OUT	REAL	
		IN_OUT		
LD11	OUT	OUT	REAL	
		OUT		



用Micro/WIN从LAD/FBD图形建立的STL代码在LAD、FBD或STL视图中启用显示。注释：Micro/WIN保留L内存（LB60-LB63）的四个上方字节，将其用于调用参数数据。

```

NETWORK 1
// L内存被用于保存布尔输入参数状态，
// 该参数在LAD和FBD中显示为功率流输入。这样可允许
// 本网络在LAD、FBD和STL编辑器中显示。
LD I0.0
= L60.0
LD I0.1
= L63.7
LD L60.0
CALL SBR_0 L63.7 VB10 I1.0 &VB100 *AC1 VD200

```

在STL编辑器中输入与以上显示相同的子例行程序调用的仅限 STL的范例 注释：STL程序员可使用该简化的调用程序

```


NETWORK 1
// 该网络只能在STL编辑器中显示，
// 因为被用作功率流输入的布尔参数
// 未保存至L内存。
LD I0.0
CALL SBR_0 I0.1 VB10 I1.0 &VB100 *AC1 VD200

```

#### 用于LAD和FBD

建立子例行程序和定义调用参数后，STEP 7-

Micro/WIN自动生成子例行程序调用方框指令。根据局部变量表中对该子例行程序的说明，调用指令包含输入 / 输出参数的正确

壳屠槽汀= (9)永 谐缘蚝媳 蚪 鱿衷谥噶钐髦小S 讷碾框 POU中插入子例行程序，从指令树中拖出子例行程序块  到急簇 湃育碾框 POU中。

#### 如何建立子例行程序

(返回顶端)

可采用下列一种方法建立子例行程序：

- 从“编辑”菜单，选择**插入 (Insert) > 子例行程序 (Subroutine)**
- 从“指令树”，用鼠标右键单击“程序块”图标，并从弹出菜单选择**插入 (Insert) > 子例行程序 (Subroutine)**
- 从“程序编辑器”窗口，用鼠标右键单击并从弹出菜单选择**插入 (Insert) > 子例行程序 (Subroutine)**

程序编辑器从先前的POU显示更改为新子例行程序。程序编辑器底部会出现一个新标记，代表新子例行程序。

此时，您可以对新子例行程序编程，或者保留子例行程序，返回您先前作业的POU位置：

- 如果您现在希望为子例行程序指定参数，您可以使用该子例行程序的**局部变量表**定义参数。

**注释：**

\*

请记住程序中每个POU都有一个独立的局部变量表。必须在选择该子例行程序标记后出现的局部变量表中为该子例行程序定义局部变量。编辑局部变量表时，必须确保已选择适当的标签。

\*\* 每个子例行程序调用的最大输入 / 输出参数限制为 16。如果您尝试下载超过该限制的程序，则会返回一则错误讯息。

- 选择子例行程序标记时，如果您希望为该子例行程序写入逻辑，在程序编辑器窗口中即可写入。
- 如果您希望对不同的POU编程，单击该POU的标签，以便在程序编辑器窗口中显示该POU。

### 不使用RET指令终止子例行程序

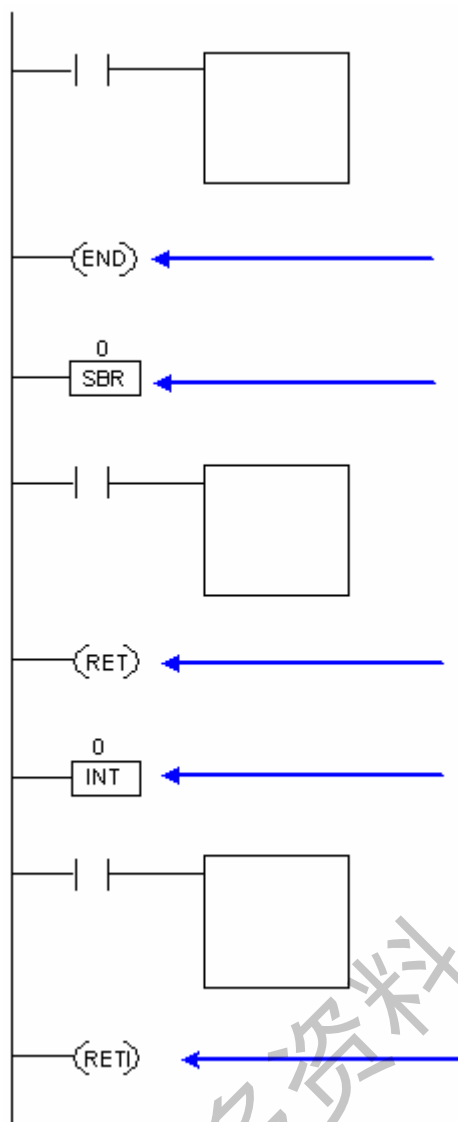
([返回顶端](#))

在子例行程序中不得使用END（结束）指令。

编辑器自动插入无条件POU终止指令（END用于OB1，RET用于SBR，RETI用于INT）。以下显示一个范例。

箭头表示指令由Micro/WIN自动处理





### 如何调用子例行程序

(返回顶端)

插入新子例行程序并在该子例行程序的局部变量表中定义参数（如果有）后，您可在程序的另一个 POU 中放置一个子例行程序调用。（您可以从 OB1、另一个子例行程序或中断例行程序调用子例行程序；您不能从子例行程序本身调用子例行程序。）

#### LAD、FBD

对于 LAD 和 FBD 程序，在子例行程序局部变量表中为该子例行程序指定参数后，会生成一个定制调用方框指令。该调用指令自动包括子例行程序输入和输出参数的正确数目和类型。

欲在 LAD 或 FBD 程序的 POU 中插入调用指令：

1. 打开程序编辑器窗口中所需的 POU，滚动至您希望插入子例行程序调用的网络处。
2. 在指令树中，双击打开的“子例行程序”文件夹。您可将适当的调用指令从指令树拖放至程序编辑器中的正确的网络单元格中，或将光标放在程序编辑器中的单元格上，然后双击指令树中的调用指令。
3. 编辑程序中的调用指令参数并为每个参数指定有效操作数。有效操作数为：内存地址、常量、总体符号以及调用指令被放置的 POU 中的局部变量（并非被调用子例行程序中的局部变量）。

**注释：**如果您在子例行程序中插入一个调用指令，然后修改该子例行程序的局部变量表，调用指令则无效。您必须删

#### STL

欲在 STL 程序中插入调用，使用 [调用指令](#)。

调用子例行程序时，保存整个逻辑堆栈，堆栈顶值被设为 1，其他所有堆栈位置均设为 0，控制转移至被调用的子例行程序。该子

例行程序完成后，用调用时保存的数值恢复堆栈，控制返回调用例行程序。  
子例行程序和调用例行程序共用累加器。不因使用子例行程序对累加器执行保存或恢复操作。

### 嵌套和递归 (返回顶端)

程序中总共可有64个子例行程序（CPU 226XM可有128个子例行程序）。在主程序中，您可以嵌套子例行程序（在子例行程序中放置子例行程序调用指令），最大嵌套层数 8。您无法从中断例行程序嵌套子例行程序。子例行程序调用无法被放置在任何从中断例行程序调用的子例行程序中。递归（子例行程序调用自身）不被禁用，但您在子例行程序中使用递归时应当小心。

## 3.3.14 中断例行程序编程

中断由事件驱动。在启动中断例行程序之前，必须使中断事件与发生该事件时您希望执行的程序段建立联系。使用“附加中断”指令（ATCH）建立中断事件（由中断事件号码指定）与程序段（由中断例行程序号码指定）之间的联系。将中断事件附加于中断事件。如果您使用全局禁用中断指令禁用所有的中断，中断事件的每次出现均被排队等候，直至使用全局启用中断指令重新启用中断。使用“拆卸中断”指令（DTCH）可拆卸中断事件与中断例行程序之间的联系，从而禁用单个中断事件。“拆卸”指令使中断返回正常操作状况。

### 了解S7-200如何处理中断例行程序

作为对关联的内部或外部事件的应答，执行中断例行程序。一旦中断例行程序的最后一条指令被执行，控制被返回至主程序。您可以用执行“从中断指令有条件返回”指令（CRETI）的方法退出例行程序。

### 使用中断例行程序指南和限制

中断程序为特殊内部或外部事件提供快速反应。您应当优化中断例行程序，执行某项具体任务，然后将控制返回至主程序。通过将中断例行程序保持为短小和简明扼要，可加快执行的速度，使其他程序不会受到长时间的延误。如果未能做到这一点，无法预料的情形可能导致主程序控制的装置出现非正常操作状况。

### 限制

您不得在中断例行程序中使用DISI、ENI、HDEF、LSCR和END指令。

### 中断的系统支持

由于接点、线圈和累加器逻辑可能受中断的影响，系统保存和重新载入说明累加器和指令操作状态的逻辑堆栈、累加器寄存器（SM）。这样可避免因分支至中断例行程序和从中断例行程序分支而导致的主程序中断。

### 在主程序和中断例行程序之间共享数据

您可以在主程序和一个或多个中断例行程序之间共享数据。因为无法预测 S7-200何时可能生成中断，最好限制中断例行程序和程序中其他位置使用的变量数目。由于主程序中指令执行被中断事件中断时，并且不盖写程序其他位置使用的的数据。

您可以使用各种编程技巧，以确保在主程序和中断例行程序之间正确地共享数据。这些技巧限制存取共享内存位置的方法，或者使用共享内存位置预防出现指令序列中断。

- 对于共享单一变量的STL程序：如果共享数据是单字节、字或双字变量，且程序在STL中写入，则可用在非共享内存位置或累加器中存储共享数据操作数的直接数值的方法，确保正确的共享存取。
- 对于共享单一变量的LAD程序：如果共享数据是单字节、字或双字变量，且程序在LAD中写入，则可用建立仅使用“贫 澈噶睿 MOVB、MOVW、MOVD、MOVR）存取共享内存位置常规的方法，确保正确的共享存取。尽管很多LAD指令由STL指令的可中断序列组成，这些“移动”指令却是由单个STL指令组成，此类指令的执行不受中断事件的影响。
- 对于共享多个变量的STL或LAD程序：如果共享数据由各种相关的字节、字或双字组成，则可使用中断禁用/启用指令（DISI和ENI）控制中断例行程序的执行。在主程序中共享内存位置操作即将开始的点，禁用中断。一旦所有影响共享位置的措施均完成后，重新启用中断。在中断被禁用的时间内，不得执行中断例行程序，因此无法存取共享内存位置；但是，此种方法会导致对中断事件的延迟应答。

### 从中断例行程序调用子例行程序

您可以从中断例行程序调用一个子例行程序嵌套级别。在被调用的中断例行程序和子例行程序之间共享累加器和逻辑堆栈。

### S7-200支持的中断类型

S7-200支持以下中断例行程序类型：

- 通讯端口中断：S7-200生成允许程序控制通讯端口的事件。
- I/O中断：S7-200生成用于各种I/O状态不同变化的事件。这些事件允许程序对高速计数器、脉冲输出或输入的升高或降低状态作出应答。
- 时基中断：S7-200生成允许程序按照具体间隔作出应答的事件。

### 通讯端口中断

可用程序控制S7-

200的串行通讯端口。此种操作通讯端口的模式被称作自由端口模式。在自由端口模式中，程序定义波特率、每个字符的位、奇夹Q 樞托 楞? 商崂 敖邮侧焙汀按 涸澈卸希 谱缘蚤 频率耐<丁O呢榭匍卧摹按 浜徒邮侧澈噶唯

### I/O中断

I/O中断包括上升/下降边缘中断、高速计数器中断和脉冲链输出中断。S7-

200可生成输入（I0.0、I0.1、I0.2或I0.3）上升和/或下降边缘中断。可为每个此类输入点捕获上升边缘和下降边缘事件。这些上陆当吹凳录 捐糜诒碲驹谱录 (5) 北甌育12创 森淖纯航

高速计数器中断允许您对诸如以下之类的条件作出应答：当前值达到预设值，可能与转轴旋转方向逆转对应的计数方向的改变或

计数器外部重置。每种此类高速计数器事件均允许针对按照可编程逻辑控制器扫描速度控制的高速事件采取实时措施。脉冲链输出中断发出输出预定数目脉冲完成的立即通知。脉冲链输出的最常用用法是步进器马达控制。

您可以用将中断例行程序附加在相关I/O事件上的方法，启用上述每种中断。

#### 时基中断

时基中断包括定时中断和计时器T32/T96中断。您可以使用定时中断基于循环指定需要采取的措施。循环时间被设为从1毫秒至255毫秒每1毫秒递增一次。您必须在SMB34中将定时中断的循环时间设为0，在SMB35中将定时中断的循环时间设为1。

每次计时器失效时，定时中断事件将控制传输给适当的中断例行程序。通常您使用定时中断控制模拟输入取样或定期执行PID环

您可将中断例行程序附加在定时中断事件上时，则启用定时中断，且计时开始。在附加的过程中，系统捕获循环时间数值，因此其后对SMB34和SMB35所作的改动不会影响循环时间。欲改动循环时间，您必须修改循环时间数值，然后将中断例行程序重新附加在定时中断事件上。重新附加时，定时中断功能从以前的附件中清除所有的累计时间，并开始用新数值计时。

时间中断被启用后，则持续运行，每当指定的时间间隔失效时，执行附加中断例行程序。如果您退出RUN（运行）模式或分离定时中断，定时中断被禁用。如果全局禁用中断指令被执行，定时中断继续进行。每次定时中断出现均排队等候（直至中断被启用或队列已满）。

计时器T32/T96中断允许对指定时间间隔完成及时作出应答。仅在1毫秒分辨率打开延迟（TON）和关闭延迟（TOF）计时器T32/T96中支持此类中断。否则T32和T96按照正常情况作业。一旦中断被启用，在S7-200中执行的正常1毫秒计时器更新的过程中，当用计时器的当前值等于预设时间数值时，即执行附加中断例行程序。您用将中断例行程序附加至T32/T96中断事件的方法，启用此类中断。

#### 中断优先级别和入队等候

S7-

200在中断各自的优先级别组内按照先来先服务的原则为中断提供服务。在任何时刻，只能执行一个用户中断例行程序。一旦桓鲋卸侠 谐缘蚩 贾葱校 蛞恢澈葱兄镣瓠伞 2 荒馨涣砒桓鲋卸侠 谐缘蛟は扰眠眨 词故款 晒畔燃伺鸬睦 谐缘颀U 洼 砒砒桓鲋卸鲜猓(5) 闹卸先父拥却 惹

#### 每个中断队列最大条目数

队列	CPU 221、CPU 222 CPU 224	CPU 224 XP CPU 226和CPU 226XM
通讯队列	4	
I/O中断队列	16	
定时中断队列	8	

一般而言，出现的中断数目会超出队列能够容纳的数目。因此，队列溢出内存位（识别已经丢失的中断事件类型）由系统保持。下表显示中断队列溢出位。您应当仅在中断例行程序中使用这些位，因为当队列排空时这些位会被重置，控制被返回主程序。

#### 中断队列溢出位

说明（0 = 无溢出，1 = 溢出） SM位

通讯队列	SM4.0
I/O中断队列	SM4.1
定时中断队列	SM4.2

可采用下列一种方法建立中断例行程序：

- 从“编辑”菜单，选择**插入（Insert） > 中断（Interrupt）**。
- 从指令树，用鼠标右键单击“程序块”图标并从弹出菜单选择**插入（Insert） > 中断（Interrupt）**。
- 从“程序编辑器”窗口，从弹出菜单用鼠标右键单击**插入（Insert） > 中断（Interrupt）**。

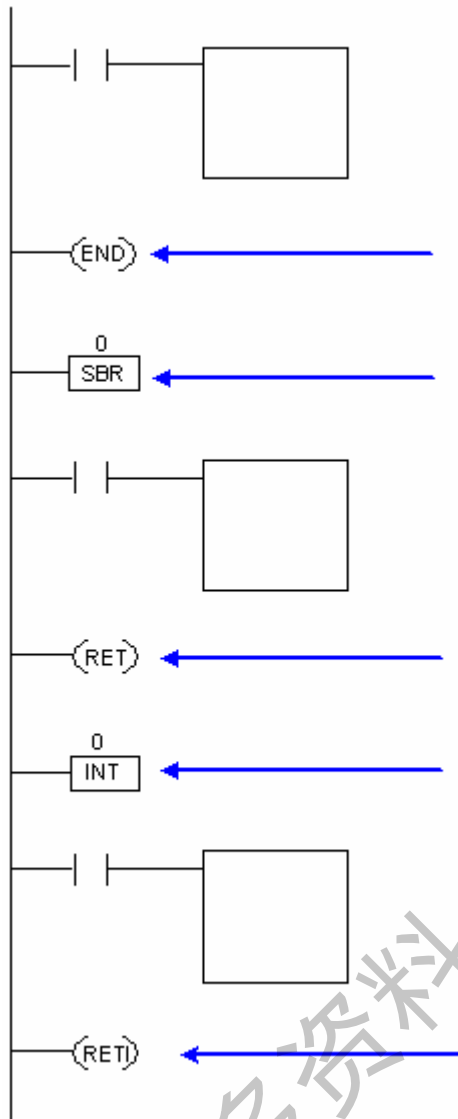
程序编辑器从先前的POU显示更改为新中断例行程序。在程序编辑器的底部会出现一个新标记，代表新中断例行程序。

一个程序中总共可有128个中断。在各自的优先赋值范围内，PLC采用先来先服务的原则为中断提供服务。在任何时刻，只能执行一个用户中断例行程序。一旦一个中断例行程序开始执行，则一直执行至完成。不能被另一个中断例行程序预先排空，即使是更高优先级别的例行程序。正在处理另一个中断时发生的中断入队等待处理。

#### 注释：

编辑器自动插入无条件POU终止指令（END用于OB1，RET用于SBR，RETI用于INT），以下显示一个范例。

箭头表示指令由Micro/WIN自动处理



另请参阅：  
[队列溢出错误（中断）](#)  
[中断事件时间间隔赋值](#)  
[中断事件优先级列表](#)  
[局部变量表](#)

**SIMATIC编程中断指令**

[LAD](#)  
[FBD](#)  
[STL](#)

**IEC编程中断指令**

[LAD](#)  
[FBD](#)

### 3.3.15 局部变量表编程

本部分讨论下列主题：  
[您的程序需要使用局部变量吗？](#)  
[理解局部变量](#)  
[局部变量的说明类型](#)

[局部变量数据类型检查](#)  
[如何隐藏 / 显示局部变量表](#)  
[如何在局部变量表中赋值](#)  
[局部变量表举例](#)

### 您的程序是否需要使用局部变量？

[返回顶端](#)

使用局部变量有两种原因：

- 您希望建立不引用绝对地址或全局符号的可移动子例行程序。
- 您希望使用临时变量（说明为 TEMP 的局部变量）进行计算，以便释放 PLC 内存。

如果以上说明对您的情形不适用，您无须使用局部变量；您可以在符号表（SIMATIC）或全局变量表（IEC）中定义符号数值，  
 械姆 攀 刀丁邇 直淞俊

### 理解局部变量

[返回顶端](#)

您可以使用程序编辑器的局部变量表指定对个别子例行程序或中断例行程序独特的变量。

局部变量可用作传递至子例行程序并可用于增加子例行程序的移动性或重新使用子例行程序的参数。

程序中的每个 POU 都有自己的局部变量表，配备 64 个字节的 L 内存。这些局部变量表允许您定义具有范围限制的变量：局部变量  
 辉消 8.帽淞康 POU 中才有效。相反，在每个 POU 中均有效的全局符号只能在符号表 / 全局变量表中定义。当您为全局符号和  
 植勘淞渴褂孟嗤 姆 芭 保g INPUT1)，定义局部变量的 POU 中的局部定义优先，全局定义用于其他 POU。

在局部变量表中赋值时，您指定说明类型（TEMP、IN、IN\_OUT 或 OUT）和数据类型（请参阅 SIMATIC 和 IEC 1131-  
 3 数据类型），但不指定内存地址；程序编辑器自动在 L 内存区中为所有的局部变量指定内存位置。

局部变量表符号地址分配会将一符号名与存储有关数据值的 L 内存地址关联。局部变量表不支持向符号名直接赋值的符号常量  
 (这在符号或全局变量表中是许可的)。



**提示：** PLC 不会将局部变量数据值初始为零。您必须在程序逻辑中初始化您使用的局部变量。

### 局部变量的说明类型

[返回顶端](#)

对局部变量赋值指定的类型取决于在其中赋值的 POU。主程序（OB1）、中断例行程序和子例行程序可使用临时（TEMP）变量  
 V 挥性谥葱铎械保 偈北淞坎坎指茫 槽葱型瓠珊媳 偈北淞靠杀恢甌础 W 永 谱缘蛭部墙褂玫瑰貌问 IN、IN\_OUT、OUT）。

#### 说明类型

#### 说明

IN 调用 POU 提供的输入参数。

OUT 返回调用 POU 的输出参数。

IN\_OUT 数值由调用 POU 提供的参数，由子例行程序修改，然后返回调用 POU。

TEMPORARY 临时保存在局部数据堆栈中的临时变量。一旦 POU 完全执行，临时变量数值则无  
 法再用。在两次 POU 执行之间，临时变量不保持其数值。

### 局部变量数据类型检查

[返回顶端](#)

将局部变量作为子例行程序参数传递时，在该子例行程序局部变量表中指定的数据类型必须与调用 POU 中数值的数据类型相匹配

举例：

您从 OB1 调用 SBR0，将称为 INPUT1 的全局符号用作子例行程序的输入参数。

在 SBR0 的局部变量表中，您已经将一个称为 FIRST 的局部变量定义为输入参数。

当 OB1 调用 SBR0 时，INPUT1 数值被传递至 FIRST。

INPUT1 和 FIRST 的数据类型必须匹配。

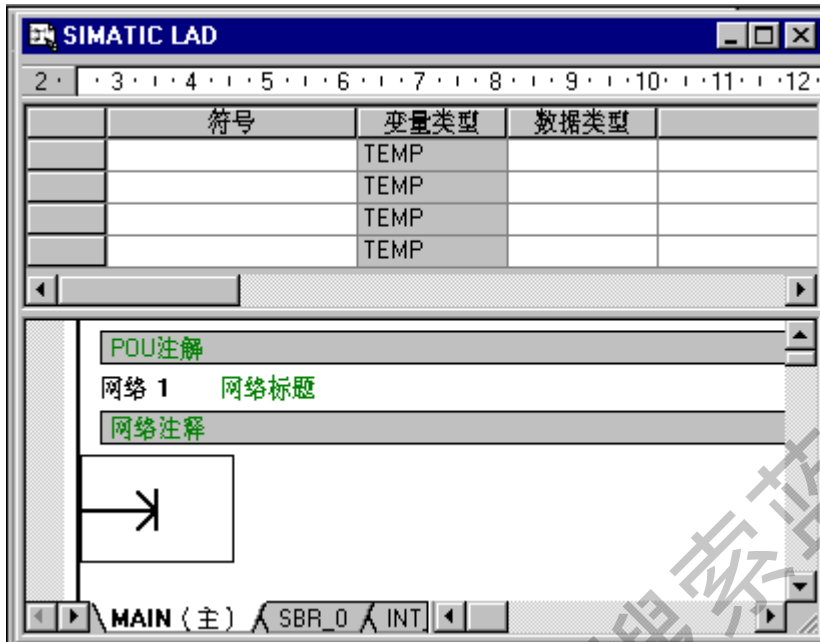
如果 INPUT1 是实数，FIRST 也是实数，则数据类型匹配。如果 INPUT1 是实数，但 FIRST 是整数，则数据类型不匹配，

挥芯勒 苏庖淮砦媳 缘蚤拍鞑噤裔

### 检视 / 隐藏局部变量表

[返回顶端](#)

如果您将水平分裂条拉至程序编辑器窗口的顶部，则局部变量表不再显示，但仍旧存在。将分裂条下拉即可再次显示局部变量表。



#### 如何在局部变量表中赋值

[返回顶端](#)

#### 注释：\*

在程序中使用局部变量之前，在局部变量表中赋值最为有效。在程序中使用符号名时，程序编辑器首先检查适当 POU 的局部变量表，然后检查符号表 / 全局变量表。

如果符号名在这两处均未定义，程序编辑器则将其视为未定义的全局符号；此类符号用绿色波浪状下划线标明。程序编辑器不会自动重新读取局部变量表并对您的程序逻辑做出修正。如果您后来进行了定义该符号名的数据类型分配（在局部变量表中），您必须以手动方式在该符号名前插入一个井号（#），例如：#UndefinedLocalVar（在程序逻辑中）。

\*\* 每个子例行程序调用的输入 / 输出参数的最大限制是 16，如果您尝试下载的程序超过此一限制，会返回一则错误信息。

#### 如何输入第一个局部变量赋值

欲在局部变量表中赋值，请遵循以下步骤。

1. 确保正确的 POU 在程序编辑器窗口中显示，如有必要，单击所需的 POU 标记。（因为每个 POU 都有自己的局部变量表）
2. 如果局部变量表处于隐藏状态，下拉水平分裂条，显示局部变量表。（请参阅 [检视 / 隐藏局部变量表](#)。）
3. 为需要定义的变量选择具有正确说明类型的行，在“名称”域中为该变量键入一个名称。（如果您在 OB1 或中断例行程序中赋值，局部变量表只包含 TEMP 变量。如果您在子例行程序中赋值，局部变量表包含 IN、IN\_OUT、OUT 和 TEMP 变量。）您不需要在局部变量表中的变量名前加井号。井号只用在程序代码中的局部变量前。

#### 注释：

\* 局部变量名最多可包含 23 个字母数字字符和下划线，也可包含扩展字符（ASCII 128 至 ASCII 255）。第一个字符只能是字母或扩展字符。将关键字用作符号名属于非法，名称的第一个字符是数字，或名称包含非字母数字字符。

\*\* 局部变量表变量名被下载并存储于 CPU 内存中，使用较长的变量名可能减少用于存储程序的内存空间。

4. 在“数据类型”域中单击鼠标指针，并使用列表框为局部变量选择适当的数据类型。

#### 注释：

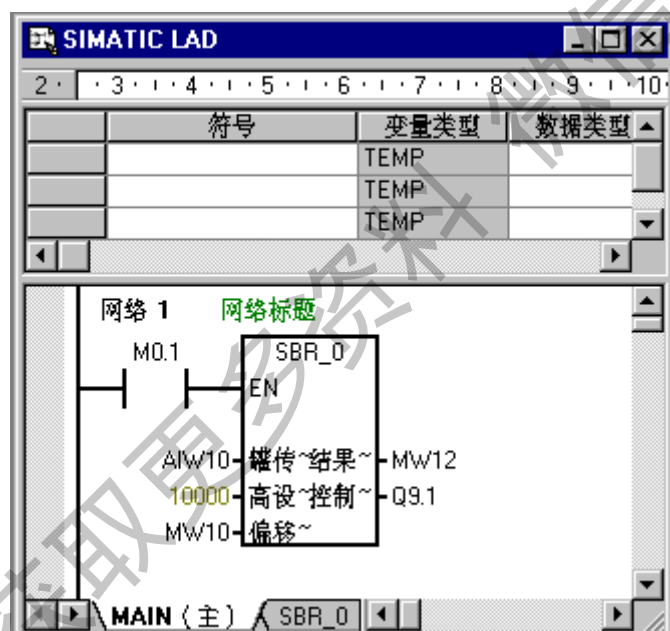
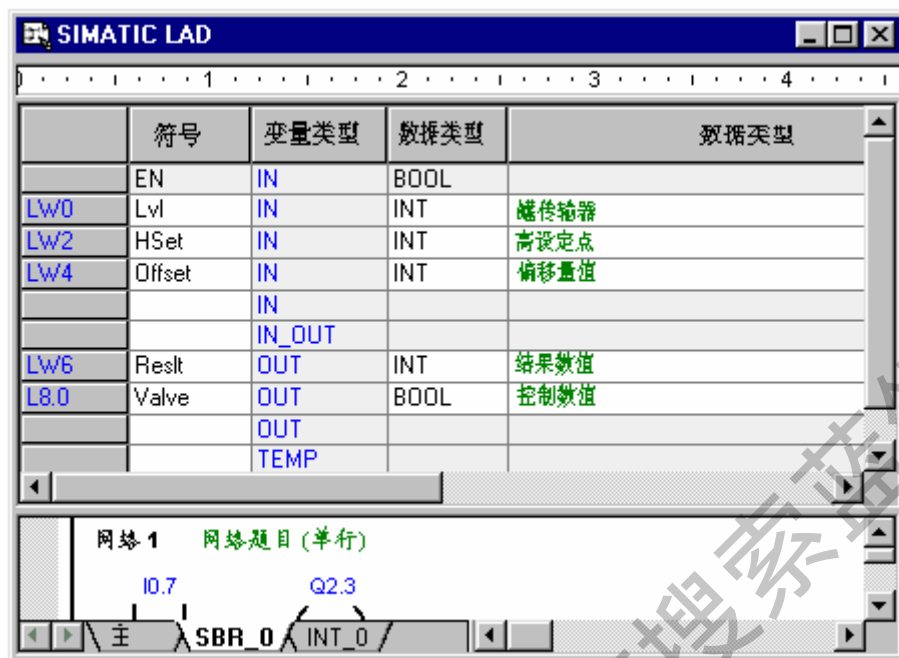
当您为局部变量指定为子例行程序的参数时，您必须保证为局部变量指定的数据类型不与子例行程序调用中使用的操作数发生冲突。（请参阅上文中的 [数据类型检查](#)。）

为“名称”和“数据类型”域提供数值后，程序编辑器会自动为局部变量指定 L 内存地址。

#### 如何在局部变量表中输入附加赋值

对于 OB1 和中断例行程序，局部变量表显示一组已被预先定义为 TEMP 变量的行。在 OB1 或中断例行程序中，只能使用这种说明类型。欲在表中增加更多行，只需单击最后一行中的一个单元格，然后使用 ENTER 键移过该行并向下移动，即自动生成一个新行。对于子例行程序，局部变量表显示按照以下顺序预先定义说明类型的一组行：IN、IN\_OUT、OUT 和 TEMP。您不能改变该顺序。局部变量在该表中的顺序必须符合您为子例行程序调用指令进行操作数赋值时对应的操作数顺序。如果您希望增加附加局部变量，必须用鼠标右键单击现有行，并使用弹出菜单插入与单击行类型相同的另一局部变量。选择 **插入 (Insert) > 行 (Row)**，在所选行的上方插入新行，或选择 **插入 (Insert) > 行下方 (Below Row)**，在所选行下方插入新行。

局部变量表举例  
[返回顶端](#)



另请参阅:

[子例行程序](#)  
[SIMATIC和IEC 1131-3数据类型](#)  
[关键字](#)

### 3.3.16 使用数据块在PLC V内存中存储数据

使用下列一种方法存取数据块：



- 单击浏览条上的“数据块”按钮。
- 选择菜单命令**检视 (V) > 数据块 (D)**。
- 打开**指令树**中的“数据块”文件夹，然后双击某块页图标。

通过插入新数据块页标记，将您的数据块V内存赋值分成多个功能组：

- 单击数据块窗口，然后选取菜单命令**编辑 (E) > 插入 (I) > 数据块 (D)**
- 在**指令树**中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取**插入 (I) > 数据块 (D)**
- 用鼠标右键单击数据块窗口，然后在弹出菜单中选取**插入 (I) > 数据块 (D)**
- 标记的最大数目为128。如果您使用向导，有关标记会被自动创建以支持向导功能。您可以创建的标记的最大数目为（由MicroWin自动创建的标记数目）。请使用Windows剪贴板合并标记数据；方法为使用剪切和粘贴由一个标记转移到另一块页。

重新命名和保护数据块页标记：

- 在**指令树**中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取**重新命名**。您也可以在指令树内直接重新命名数据块页，方法为单击该标记页名称两次（动作要慢一些，以免解释成双击）；然后编辑该标记名。数据块编辑器提供相同的重新命名功能，方法为用鼠标右键直接单击该标记名。
- 在**指令树**中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取**属性**。由此，您可以重新命名该数据块标记和指定作者。属性对话框的**保护**标记令您能够用密码保护单个数据块标记。受保护的标记会显示锁图标。数据块编辑器向导会创建不能重新命名且包含只读数据值的受保护标记。

导入和导出数据块数据至ASCII文本文件

- [由ASCII文本文件导入数据块](#)
- [由ASCII文本文件导出数据块](#)

本标题讨论下列主题：

[在数据块中进行地址和数据值赋值](#)

[数据块举例](#)

[在数据块编辑器中使用“剪切”、“复制”和“粘贴”](#)

[使用ASCII常量的限制](#)

[理解和解决错误](#)

[将数据块下载至PLC](#)

[从PLC上载数据块](#)

[上载后保留数据块格式](#)

[从CPU RAM建立数据块](#)

在数据块中进行地址和数据值赋值

[返回顶端](#)

数据块仅允许您对V内存进行初始数据或ASCII字符赋值。您可以对V内存的字节 (V或VB)、字 (VW) 或双字 (VD) 赋值。注解（前面带双正斜线//）是选用项目。

- 数据块的第一行必须包含一个明确地址赋值（绝对或符号地址），其后的行可包含明确或隐含地址赋值。当您在单地址赋值后键入多个数据值或键入仅包含数据值的行时，由编辑器指定隐含地址赋值。编辑器根据先前的地址分配及数据值大小（字节、字或双字）指定适当的V内存数量。
- 数据块编辑器是一种自由格式文本编辑器，对特定类型的信息无具体域。键入一行后，按ENTER键，数据块编辑器格式并重新显示行。数据块编辑器接受大小写字母并允许使用逗号、制表符。
- 在完成一赋值行后按CTRL-ENTER键组合，会令地址自动增加至下一个可用地址。

[常量格式详细说明](#)

[数据块一般规则](#)



```

数据块
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
VBO 248 //明确地址赋值：VBO数据值：248。
VB1 249, 250, 251 //单行中多个数据值。
//隐含地址赋值：
//VB2包含数据值250。
//VB3包含数据值251。
VB4 252 //不能使用先前指定的地址（VBO-VB3）。
253, 254, 255 //无明确地址赋值的行。
//数据值隐含指定给VB5、VB6、VB7。
VW8 256, 257 //新数据类型（字）。隐含将数据值
//257指定给V内存VB10-VB11。
//地址赋值不能与先前的明确赋值发生冲突。
X 65536 //数据值65536要求双字数据类型（VD内存），
//但上一个明确地址赋值
//是字内存（VW8）。编辑器标志错误。
//您可以从地址中省略尺寸规格。
//然后您可以对随后的数据使用隐含地址赋值，
//不必考虑数据的尺寸。
V12 258 //字值明确指定给V12-V13。
65537 //双字值隐含指定给V14-V17。

```

数据块举例

[返回顶端](#)

直接地址和数值

```

数据块
//
//数据块注解
//
VB0    255          //字节值从VB0开始
VW2    256          //字值从VW2开始
VD4    700.59      //双字真值从VD4开始
VB8    -35         //字节值从VB8开始
VW10   16#A        //字值HEX从VW10开始
VD14   146879      //双字值从VD14开始
VW20   2, 4, 16, 32, 64 //字值表从VD14开始
      -2, -4, -16, -32, -64 //扩展至多行的数值不能出现在列中
//第一列为内存地址保留
VB45   'Up'        //起始于VB45的双字节ASCII字符串
VW90   65535       //起始于VW90的字值
V100   255, 'abcd', 65535, 1.0 //起始于VB100的未定义数值：可混合不同尺寸
VB110  255         //与输入B110 255相同
VW120  256         //与输入W120 256相同
VD130  700.59      //与输入D130 700.59相同
V140   255, 'abc', 65535, 1.0 //与输入140相同：必须从列1开始
VW150  2#1010101010101010 //二进制字值
VD152  2#1100110011001100110011001100 //二进制双字值

```

符号地址和符号数值

符号表			
· 3 · · 4 · · 5 · · 6 · · 7 · · 8 · · 9 · · 10 · · 11 · · 12 · · 13 · · 14 · · 15 · · 16 · · 17 · · 18 · ·			
	符号	地址	注解
1	符号地址	VB1	符号地址
2	符号常数	200	符号常数

用户1 / POU符号 /

```

数据块
2 · · 3 · · 4 · · 5 · · 6 · · 7 · · 8 · · 9 · · 10 · · 11 · · 12 · · 13 · · 14 · · 15 · · 16 · · 17 · · 18 · ·
VB0    100
符号地址 符号常数 //为VB1指定一个200数值
用户1 /

```

替代二进制输入方法（上窗口）和结果二进制格式（下窗口）

```

数据块
2 · · 3 · · 4 · · 5 · · 6 · · 7 · · 8 · · 9 · · 10 · · 11 · · 12 · · 13 · · 14 · · 15 · · 16 · · 17 · · 18 · ·
v0.2 1 0 true false off on
用户1 /

```



```

数据块
2 · · 3 · · 4 · · 5 · · 6 · · 7 · · 8 · · 9 · · 10 · · 11 · · 12 · · 13 · · 14 · · 15 · · 16 · · 17 · · 18 · ·
VB0    2#10010100 //按Enter键后二进制模式
用户1 /

```

在数据块编辑器中使用“剪切”、“复制”和“粘贴”  
[返回顶端](#)

当在数据块中执行剪切、复制、粘贴时，重要的是通过鼠标单击设置适当的上下文：

如果您单击指令树中的数据块标记页  图标，或者单击数据块窗口中的标记名 ，  
 则 霰县且成喜僮鳌 T 洼松舷挛闹杏檬孟暖壹 七 颯 突嵯允咀饕糜谩 黎 菘楸县且车牟僮鞞 霏说 7  
 如果您单击数据块编辑器中的文字区域，那么您将在数据块窗口中已选定(突出显示)的文字上操作。在此上下文中用鼠标右  
 七 颯 突嵯允咀饕糜谩 黎 菘楸县且车牟僮鞞 霏说 7

### 使用ASCII常量的限制

[返回顶端](#)

有效的ASCII赋值：

唯有字节地址(V或VB)能够与长ASCII常量(使用单引号或双引号格式)合用：

VBO 'A'、VWO 'AB'、VDO 'ABCD'

就3和5或更多字节而言，您必须使用V或VB地址前缀 VBO 'ABC'、VBO 'ABCDE'、VBO 'ABCDEFGH IJK'



双引号ASCII常量字符串格式存储一个前导长度字节和字符串 VBO "A"、VBO "AB"、VBO "ABC"、VBO "ABCD"、VBO "ABCDEFGH IJK"

[常量格式详细说明](#)

### 理解和解决错误

[返回顶端](#)

一旦在包含错误的行尾按ENTER键，立即会在数据块左页边显示输入错误。您必须纠正全部输入错误，才能成功地编译。

	vbbo	255	// (红色文本 - 非法语法错误) 应为vb0
	VBO	300	// (红色曲线下划线 - 非法使用错误) 一个字节最大为255

引起输入错误的条件包括：

- 指定错误内存区 (V是唯一允许使用的内存区)
- 当数据值实际要求较大的尺寸时 (例如，数据值256过大，无法在VB地址中存储—要求使用VW地址)，在地址赋值兄付 骋荒洼姆叽纾 卫纸译厝郑
- 在一行中使用错误序列：在数据值之后 (而不是在数据值之前) 键入内存地址
- 使用非法语法或无效数值
- 尝试使用符号，而不是使用绝对V内存地址 (数据块中不允许使用符号)
- 未能适当地指定注解 (双正斜线必须位于注解之前：//注解样本) 如果数据块是现用窗口，您可以使用菜单命令 **PLC > 编译 (Compile)** 编译数据块。如果数据块不是现用窗口，您依然可以编译数据块：使用菜单命令 **PLC > 全部编译 (Compile All)**。

编译数据块时，如果编译程序发现错误，会在“输出窗口”显示错误。将光标置于“输出窗口”中的错误讯息上，双击该讯息 谿 崧椽翱溢那允境趵刚小  
 仅在编译后显示的错误包括：

编译数据块时，如果编译程序发现错误，会在“输出窗口”显示错误。将光标置于“输出窗口”中的错误讯息上，双击该讯息，在数据块窗口中显示出错行。仅在编译后显示的错误包括：

- 重复地址赋值 (例如，如果您输入“VB1 249, 250”之类的行，则是对VB2进行250隐含赋值—您不得在别处对地址VB2进行其他不同的数据值赋值)
- 地址重叠 (例如，如果您为VDO指定一个类似65536的双数值，则不得再对V1、V2或V3指定其他赋值，因为这些数值已被使用，是以VDO开始的双字的一部分)

### 将数据块下载至PLC

[返回顶端](#)

如果要编辑数据块，则需将数据块 **下载** 至PLC。只有在修改的数据块下载后您的编辑才会生效。欲节省空间，您可以将您不希望下载至PLC的信息切换为打开 / 关闭。

### 从PLC上载数据块

[返回顶端](#)

您必须在STEP 7-Micro/WIN中打开一个项目，才能 **上载** 数据块。

如果PLC中的数据块与您打开的项目中的数据块不匹配 (或者您打开的项目中没有数据块)，您只能上载该 PLC的数据块地址 糠趾褪 莼挡糠郑 荒茺显刈13.袞 5 卍芬宰纸谿 蹈裕缴显兀 莼狄允 聘裕缴显兀ò 匕魏卧 模 H纛 谿】占，您可以将您不希望下载至PLC的信息切换为打开 / 关闭。

如果CPU中的数据块包含由Micro/WIN 4.0 (或较新版本) 下载的标记信息，那么由Micro/WIN

4.0的上载将会重新载入此标记结构。如果某数据块曾由较早的Micro/WIN版本下载且无标记信息，那么由Micro/WIN 4.0的上载将会将所有赋值放在同一个标记中。

#### 上载后保留DB格式

##### [返回顶端](#)

当您上载数据块时，下载至PLC的数据块条目格式被保留。如果您在一行中输入多个条目，当您上载时，格式不被保留。每个条目在一个不同的行中显示。例如，如果您在数据块中输入VB0 20、30、40、50并上载，格式更改为：

VB0	20
VB1	30
VB2	40
VB3	50

#### 从CPU RAM建立数据块

##### [返回顶端](#)

选择**PLC > 从RAM建立数据块 (Create Data Block from RAM)** 菜单命令，将CPU V内存保存至EEPROM。从CPU读取V内存当前值，然后将V内存图像作为数据块下载。

#### 注释：

进行此操作时，CPU必须处于STOP（停止）模式。

倘若CPU

RAM内有已修改的V内存地址，并且这些地址在上一次数据块下载中未得到指定，那么在您执行“从RAM建立数据块”命令时，这一研需牡牡 司坊岂恢付ù 恒鲂碌谋县敲 g 罐LC\_DATA1)。当程序的执行向新的V地址写入数据时，或当状态图被用于修改碌膺地址时，此情形就可能发生。

#### 另请参阅：

[PLC电源中断后的数据保留](#)

[常量格式详细说明](#)

[通讯配置](#)

[PLC类型选择](#)

[系统块 \(PLC配置\)](#)



[错误讯息](#)

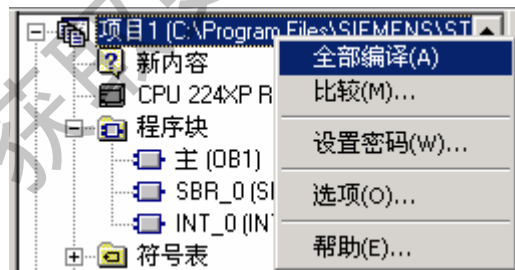
[入门指南目录](#)

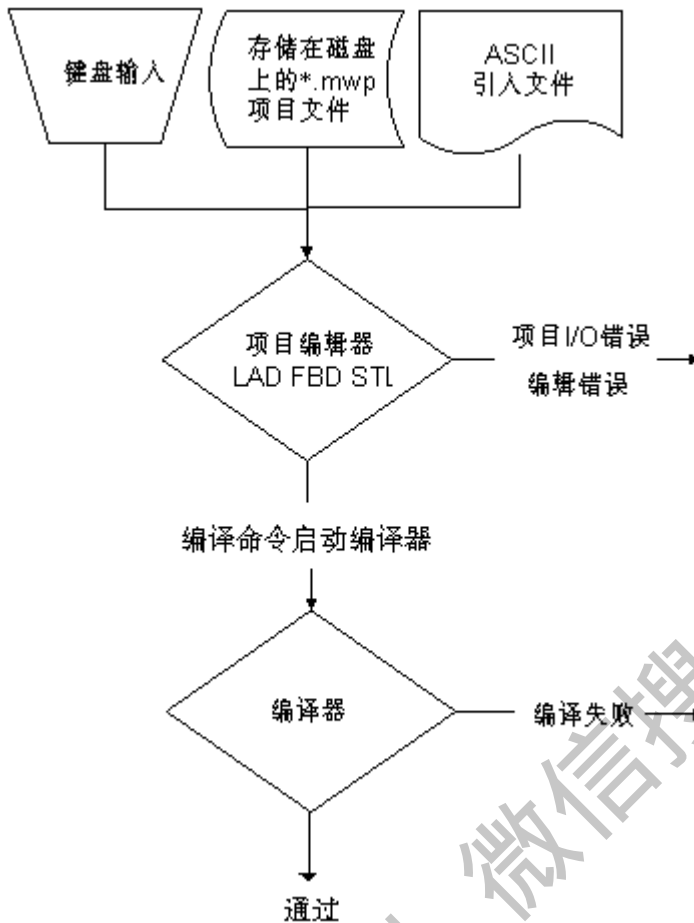
[应用程序用户参考手册](#)

### 3.3.17 编译项目

可使用下列一种方法启动STEP 7-Micro/WIN项目编译程序：

- 单击“编译” 按钮或选择菜单命令**PLC (PLC) > 编译 (Compile)**，编译现用窗口（程序块或数据块）。
- 单击“全部编译” 按钮或选择菜单命令**PLC (PLC) > 全部编译 (Compile All)**，编译全部项目元件（程序块、数据块和系统块）。
- 用鼠标右键单击**指令树**中的某个文件夹，然后由弹出菜单中选取编译命令。项目、程序块文件夹、系统块文件夹及数





设置PLC类型 项目文件I/O错误 LAD/FBD编辑器错误 全部STEP 7-Micro/WIN编译器错误均列在输出窗口中。双击错误，编辑器将滚动至错误位置。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译器。

[打开项目](#) [范围检查](#)

另请参阅：  
[应用程序用户参考手册](#)  
[入门指南目录](#)

### 3.3.18 检视PLC内存用法的交叉引用总结

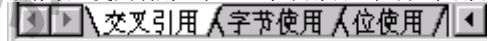
可使用下列一种方法检视“交叉引用”窗口：

- 选择菜单命令 **检视 (View) > 交叉引用 (Cross Reference)**



- 单击浏览条中的“交叉引用”按钮
- 打开 [指令树](#) 中的“交叉引用”文件夹，然后双击某引用或使用节点

欲存取“交叉引用”表、“字节用法”表或“位用法”表，单击位于“交叉引用”窗口底部的适当标记



本标题讨论下列主题：

[交叉引用表](#)  
[字节用法表](#)  
[位用法表](#)

[交叉引用表](#)  
[返回顶端](#)



**提示：** 您必须编译程序才能检视“交叉引用”表。

当您希望了解程序中是否已经使用和在哪里使用某一符号名或内存赋值时，可使用“交叉引用”表。“交叉引用”列表识别在程序中使用的所有操作数，并识别 POU、网络或行位置以及每次使用的操作数指令上下文。

**元素**指程序中使用的所有操作数。您可以在符号和绝对视图之间切换，改变全部操作数显示。（使用菜单命令 **检视 (View) > 符号编址 (Symbolic Addressing)**。）

**块**指使用操作数的 POU。

**位置**指使用操作数的行或网络。

**上下文**指使用操作数的程序指令。

LAD 交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1	LD
2 动2	MAIN (OB1)	网络 2	LD
3 停止1	MAIN (OB1)	网络 1	LD
4 停止2	MAIN (OB1)	网络 2	LD
5 高等级	MAIN (OB1)	网络 1	LD
6 高等级	MAIN (OB1)	网络 2	LD

FBD 交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1	OR
2 动2	MAIN (OB1)	网络 2	OR
3 停止1	MAIN (OB1)	网络 1	AND
4 停止2	MAIN (OB1)	网络 2	AND
5 高等级	MAIN (OB1)	网络 1	AND
6 高等级	MAIN (OB1)	网络 2	AND

STL 交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1, 行 1	LD
2 动2	MAIN (OB1)	网络 2, 行 1	LD
3 停止1	MAIN (OB1)	网络 1, 行 3	A
4 停止2	MAIN (OB1)	网络 2, 行 3	A
5 高等级	MAIN (OB1)	网络 1, 行 4	AN
6 高等级	MAIN (OB1)	网络 2, 行 4	AN

字节用法表

[返回顶部](#)



**提示：** 您必须编译程序才能检视“字节用法”表。

“字节用法”表允许您查看程序中使用了哪些字节以及在哪些内存区使用，还可帮助您识别重复赋值错误。

**b**表示已经指定一个内存位。

**B**表示已经指定一个内存字节。

**W**表示已经指定一个字（16位）。

**D**表示已经指定一个双字（32位）。

**X**用于计时器和计数器。

范例1：解释字节用法表

该字节用法表范例显示相关程序使用下列内存位置：MB0中一个位；计数器C30；计时器T37。

字节	9	8	7	6	5	4	3	2	1	0
MB0										b
C0										
C10										
C20										
C30										X
T0										
T10										
T20										
T30			X							

范例2：识别重复赋值错误

该范例程序从MB 10.0开始进行重复内存赋值。

```

网络标题
LD M10.0
AB= MB10, VB0
AW= MW10, VW0
AD= MD10, VD0

```

可检查字节用法表，识别不适当的赋值。因为双字要求四个字节，VB0行中应有4个相邻的D。与此相似，因为字要求2个字节，VB0中应有2个相邻的W。MB10行存在相同的问题，此外在多个赋值语句中使用MB10.0。

字节	9	8	7	6	5	4	3	2	1	0
VB0							D	D	W	B
MB0										
MB10							D	D	W	b

位用法表

[返回顶端](#)



**提示：** 您必须编译程序才能查看位用法表。

位用法表允许您查看程序中已经使用了哪些内存地址，可精确至位级别，还可帮助您识别重复赋值错误。

**b**表示已经指定一个内存位。

**B**表示已经指定一个内存字节。

**W**表示已经指定一个字（16位）。

**D**表示已经指定一个双字（32位）。

**X**用于计时器和计数器。

范例1：解释位用法表

该位用法表范例显示相关程序使用下列内存位置：从字节I0、位0、1、2、3、4、5和7；从字节Q0、位0、1、2、3、4和5；从字节M0、位1。

位	7	6	5	4	3	2	1	0
I0.0	b		b	b	b	b	b	b
Q0.0			b	b	b	b	b	b
M0.0							b	

范例2：识别重复赋值错误

该范例程序从MB 10.0开始进行重复内存赋值。

```

网络 1  网络标题
LD      M10.0
AB=     MB10, VW0
AW=     MW10, VV0
AD=     MD10, VD0
  
```

可检查位用法表，识别不适当的赋值。在适当的赋值程序中，字节中间不得有位值。BBBBBBb无效，而BBBBBBBb则有效。

（注：16个相邻的W）和双字赋值（应有32个相邻的D）。

位	7	6	5	4	3	2	1	0
M0.0								
M1.0								
M2.0								
M3.0								
M4.0								
M5.0								
M6.0								
M7.0								
M8.0								
M9.0								
M10.0	B	B	B	B	B	B	B	b
M11.0	W	W	W	W	W	W	W	W
M12.0	D	D	D	D	D	D	D	D
M13.0	D	D	D	D	D	D	D	D



另请参阅:


[通讯配置](#)  
[PLC类型选项](#)  
[系统块 \(PLC配置\)](#)  
[错误](#)  
[在符号和绝对视图之间切换](#)

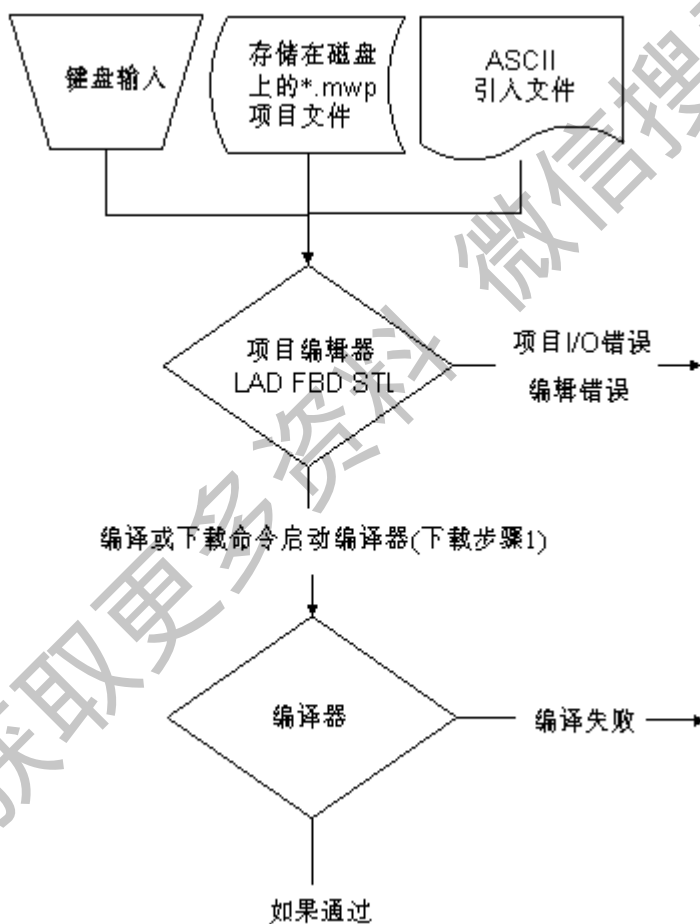
[入门指南目录](#)  
[应用程序用户参考手册](#)

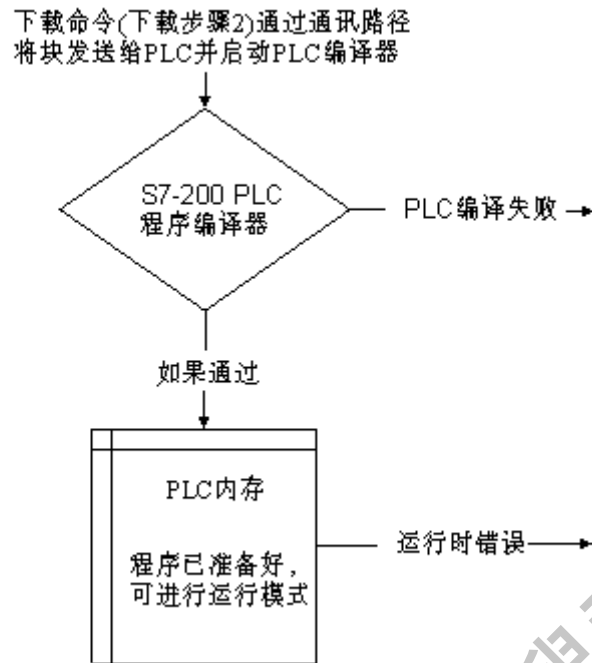
### 3.3.19 将程序下载至PLC

可使用下列一种方法从STEP 7-Micro/WIN向PLC下载项目元件:

- 单击“下载”  按钮。
- 选择菜单命令文件 (File) > 下载 (Download)。
- 按Ctrl+D快捷键组合。

单击“选项”按钮 ，显示和设置下载操作选项。‘上载下载管理器’窗口可帮助您控制上载、下载、清除及  
[UDM - 上载下载管理器详细说明](#)

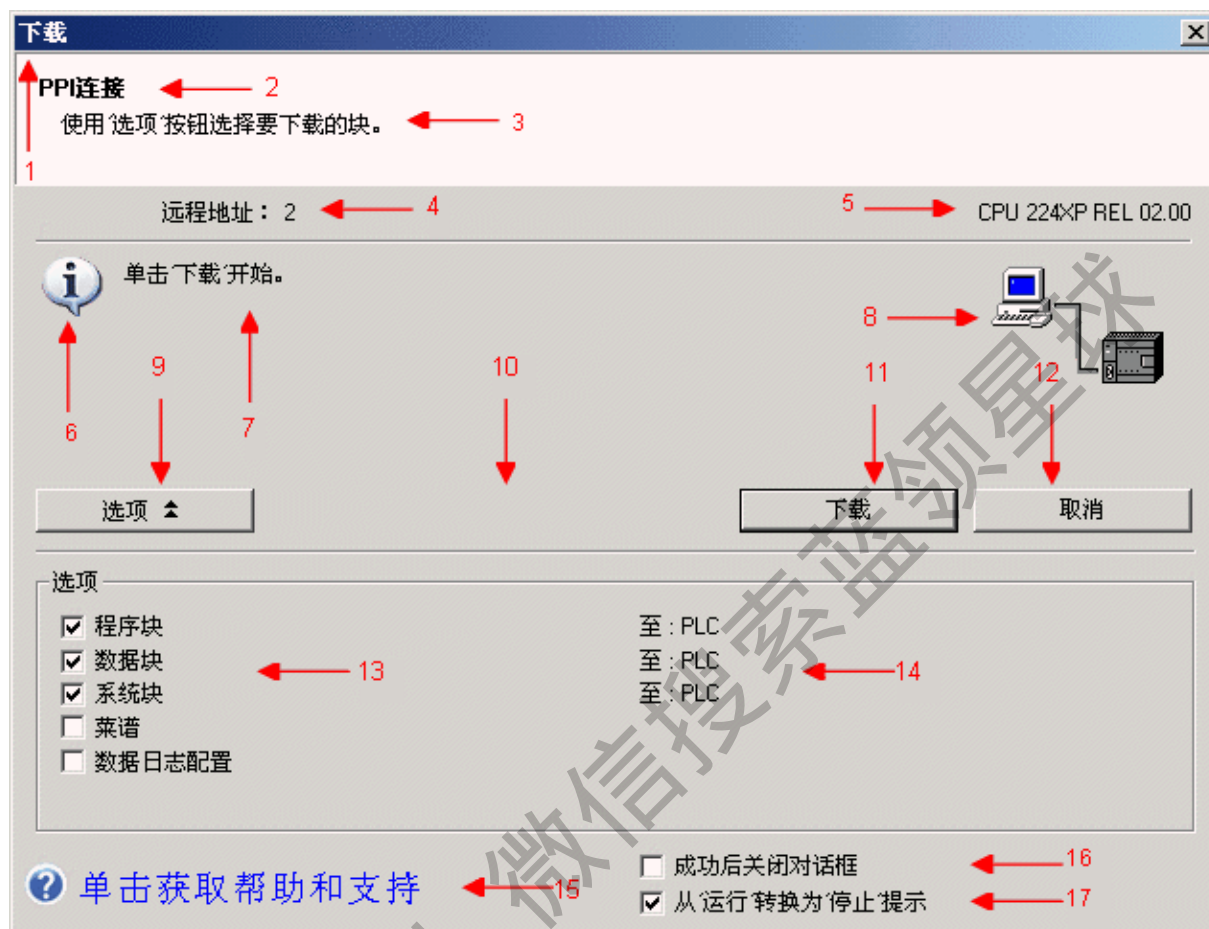




设置PLC类型 项目文件I/O错误 LAD/FBD编辑器错误 全部STEP 7-Micro/WIN编译器错误均列在输出窗口中。双击错误，编辑器滚动至错误位置。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译器。 **通讯错误** 欲下载（编辑器至PLC）或上载（PLC至编辑器），PLC通讯必须正常运行。确保网络硬件和PLC接头电缆均正常操作。PLC编译器确认PLC硬件支持全部程序指令、范围和结构。使用菜单命令 **PLC > 信息** 查看发现的第一条编译错误。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译程序，这样会将尽可能多的错误移至输出窗口，易于找到错误位置和改正错误。 **严重错误**和**非严重运行时间编程故障**由菜单命令 **PLC > 信息** 报告。

[打开项目](#) [范围检查](#)

UDM - 上下载管理器详细说明：（用于上载、下载、清除及编程内存操作）



1. 标题 - 将进行的操作
2. 连接类型 - 通讯连接类型
3. 操作说明与指令 - 进程说明与指示
4. CPU地址 - 通讯地址
5. CPU类型 - 目标CPU的版本和类型
6. 状态图标 - 表明进程的总体状态
7. 状态讯息区 - 显示错误和警告的区域
8. 动画图标 - 表明操作处理是否在进行中
9. 选项按钮 - 基于用户的优先选择隐藏或显示选项选择
10. 更改项目/通讯按钮 - 更改项目的CPU类型以与实际CPU相符，也允许通讯配置
11. 动作按钮 - 如果没有错误存在就开始操作
12. 取消按钮 - 在下一个停止点取消正在进行的操作
13. Option selections - 选取选项
14. Option status? 选定的选项状态
15. 帮助 - 打开MicroWin帮助（与按F1键相同）
16. 成功操作后关闭对话框 - 用于在成功操作后关闭此窗口的选项
17. 从‘运行’转换为‘停止’提示 - 当CPU从运行模式转换为停止模式时省略用户提示的选项。

#### 另请参阅:

[通讯概述 \(GS 6.1\)](#)

[如何测试通讯网络 \(GS 6.2\)](#)

[如何下载程序 \(GS 6.3\)](#)

[在运行模式中进行程序编辑](#)

[上载 \(文件 > 上载\)](#)

[如何从PLC上载 \(GS 8.3\)](#)

[STEP 7-Micro/WIN编译程序错误](#)

[如何纠正编译错误和下载错误 \(GS 6.4\)](#)

[应用程序用户参考手册](#)  
[入门指南目录](#)

### 3.3.20 使用状态监控确认运行时间性能

您成功地在运行的STEP 7-Micro/WIN编程设备和PLC之间建立通讯并向PLC下载程序后，则可利用“调试”工具条的诊断特征。可单击工具条按钮或从“调试”菜单列表选择项目，选择调试工具。

**调试工具条**



本标题讨论下列主题：

[什么是“状态”？](#)

[前提条件](#)

[收集状态的不同方法](#)

[以连续、瞬态图或触发暂停的方式收集状态](#)

[PLC模式](#)

[状态通讯与扫描循环](#)

[模拟程序条件（读取、写入、强制、取消强制）](#)

[检查交叉引用和元素用法](#)

[向程序下载改动](#)

[重设用于调试和编辑的帧设置](#)

[什么是“状态”？](#)

[返回顶端](#)

‘状态’这一术语指显示程序在PLC中执行时有关PLC数据当前值和功率流的信息。您可以使用状态图和程序状态窗口读取、写入和强制PLC数据值。

在控制程序的执行过程中，PLC数据的动态改变可用三种不同方式检视：

#### 图状态

在一表格中显示状态数据：每行指定一个要监视的PLC数据值。您指定一个内存地址、格式、当前值及新值（如果使用写入命令）。

#### 趋势显示

用随时间而变的PLC数据之绘图跟踪状态数据：

您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势数据亦可在趋势视图中直接赋值。

#### 程序状态

在程序编辑器窗口中显示状态数据：当前PLC数据值会显示在引用该数据的STL语句或LAD/FBD图形旁边。LAD图形也显示功率流

#### 程序状态和图状态 (或趋势视图) 窗口可以同时运行：

在状态图窗口写入或强制PLC数据将PLC数据改动应用于程序状态窗口。

在程序状态窗口写入或强制PLC数据还会将新PLC数据改动应用于状态图窗口。

#### 检视状态数据的前提条件

[返回顶端](#)

您必须执行下列任务，才能监控状态数据或调试程序：


- 成功地在STEP 7-Micro/WIN和PLC之间建立通讯。
- 成功地向PLC下载程序。
- 欲检视连续状态更新，PLC必须位于RUN（运行）模式。否则，您只能看到对I/O的改动（如果有）。由于PLC程序辉溢葱校 I/O的改动不会对“状态”显示中的程序逻辑产生预期的影响。
- 如果您位于程序不在执行的区域（例如，子例行程序、中断例行程序或由于JMP指令被绕过的区域），不会有状态显示，因为不在对代码执行扫描。

#### 注释：

- STEP 7-Micro/WIN中项目的时间标记和PLC必须匹配，才能激活状态数据通讯。时间标记比较（在您尝试打开状态时自动执行）繁 STEP 7-Micro/WIN中的项目状态表示正确反映实际PLC程序。如果时间标记不匹配，您可以检查“时间标记不匹配”对话框中南吕 斜恚 私悱男 纛斐刹黄 洵
- 下载程序后，您还应将PLC转换回RUN（运行）模式。否则会报告STOP（停止）模式状态数据，但PLC无法执行程序，因此您将无法看到预期的逻辑操作。

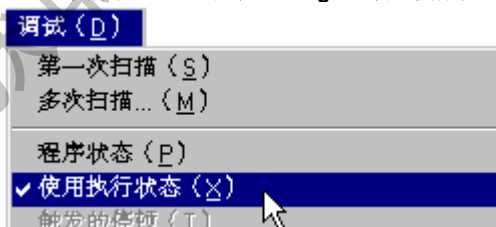
#### 在程序编辑器窗口中收集状态的不同方法

[返回顶端](#)

单击“程序状态打开/关闭”按钮，或选择菜单命令**调试 (Debug) > 程序状态 (Program Status)**，在程序编辑器窗口中显示PLC数据状态。状态数据收集按以前选择的模式开始。请参阅以下说明。

LAD和FBD程序有两种不同的程序状态数据收集模式。选择**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单命令会在状态模式复选标记打开和关闭之间切换。必须在程序状态操作开始之前选择状态模式。

- **执行状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选后）



- **扫描结束状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被取消勾选后）



状态图始终使用“扫描结束”状态数据。以前的 STEP 7 Micro/WIN 版本用“扫描结束”状态数据建立 LAD 和 FBD 程序状态视图。如果您使用的是第一代 PLC，执行状态不受支持。如果您尝试执行执行状态，“扫描结束”状态开始。

#### 执行状态色彩：

- 程序被扫描时，电源横杆会带有色彩。
- 图形中的功率流用色彩表示。
- 接点？接点打开时，指令会带有色彩。
- 线圈？输出打开时，指令会带有色彩。
- 方框和 SUBR 指令？指令接通电源并准确无误地成功执行时，方框和 SUBR 指令带有色彩。
- 绿色计时器和计数器表示计时器和计数器包含有效数据。
- 红色表示执行的指令有误。
- 跳接和标签指令现用时，显示为功率流色彩。如果为非现用，则显示为灰色。
- 灰色（灰色为默认赋值）表示无功率流、指令未扫描（跳过或未调用）或位于 STOP（停止）模式的 PLC。
- 布尔功率流位（仅限 FBD）。

LAD、FBD 和 STL 执行状态视图显示操作数数值，并表示每条指令在位于扫描循环执行程序阶段时的功率流。执行状态可以显示可能被执行后续程序指令盖写的中间数据值。所有显示的 PLC 数据值均从一个程序扫描循环收集。

#### 扫描结束状态

- 接通电源或逻辑真的接点和线圈显示为蓝色（您可以选择 **工具 (Tools) > 选项 (Options)** 菜单项目并单击“色彩”标记，指定自己选择的色彩）。

“扫描结束”状态显示在程序扫描结束时读取的状态结果。这些结果可能不会反映 PLC 数据地址的所有数值改动，因为随后某绿蜂鸣铃语毓绿蛭 杞喂 翱瞻芊慈牒椭陔漆慈胧 怠 S 捐诳炯 PLC 扫描循环和相对慢速的 PLC 状态数据通讯之间存语乃俛快败稿 吧 杞喂 弊刺 允炯父鑿 枢 方喂 辈杉 氛 奠怠

#### STL 程序中程序状态举例：

当您开启 STL 中的状态时，程序编辑器窗口被分为一个代码区（左侧）和一个状态区（右侧）。可以根据您希望监控的数值类型

在 STL 状态中共有三个可监控的数值类别：

- **操作数** 每条指令最多可监控三个操作数。
- **逻辑堆栈** 最多可监控四个来自逻辑堆栈的最新数值。
- **指令状态位** 最多可监控十二个状态位。

**工具 (Tools) >**

**选项 (Options)** 对话框的 STL 状态标记允许您选择或取消选择任何此类数值类别。如果您选择一个项目，该项目不会在“状态”显示中出现。

	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+16095	
5	T32	位	2#0	
6	T32	带符号	+0	

The screenshot shows the SIMATIC STL editor window. The title bar reads 'SIMATIC STL'. The main area contains a table of instructions with their execution status. The status column shows '0123' and a '变' (change) indicator. The instructions are as follows:

指令	操作数 1	操作数 2	操作数 3	0123	变
LD	I0.0	ON		1000	1
A	SM0.5	ON		1000	1
LD	I1.0	OFF		0100	0
A	I1.1	OFF		0100	0
OLD				1000	1
LD	I2.0	OFF		0100	0
A	SM0.5	ON		0100	1
OLD				1000	1
LD	I0.2	ON		1100	1
A	SM0.5	ON		1100	1
LD	I1.2	OFF		0110	0
A	I1.3	OFF		0110	0
OLD				1100	1
ALD				1000	1
LPS				1100	1
MOVW	VW0, VW2	+15919	+15919	1100	1
AENO				1100	1
+I	VW0, VW2	+15919	+31838	1100	1
AENO				1100	1
=	Q0.0	ON		1100	1
LRD				1100	1
TON	T32, +32000	+288	+32000	1100	1
LRD				1100	1
TNCW	VW0	+15920		1100	1

The status bar at the bottom shows 'MAIN | SBR\_0 | INT\_0'.

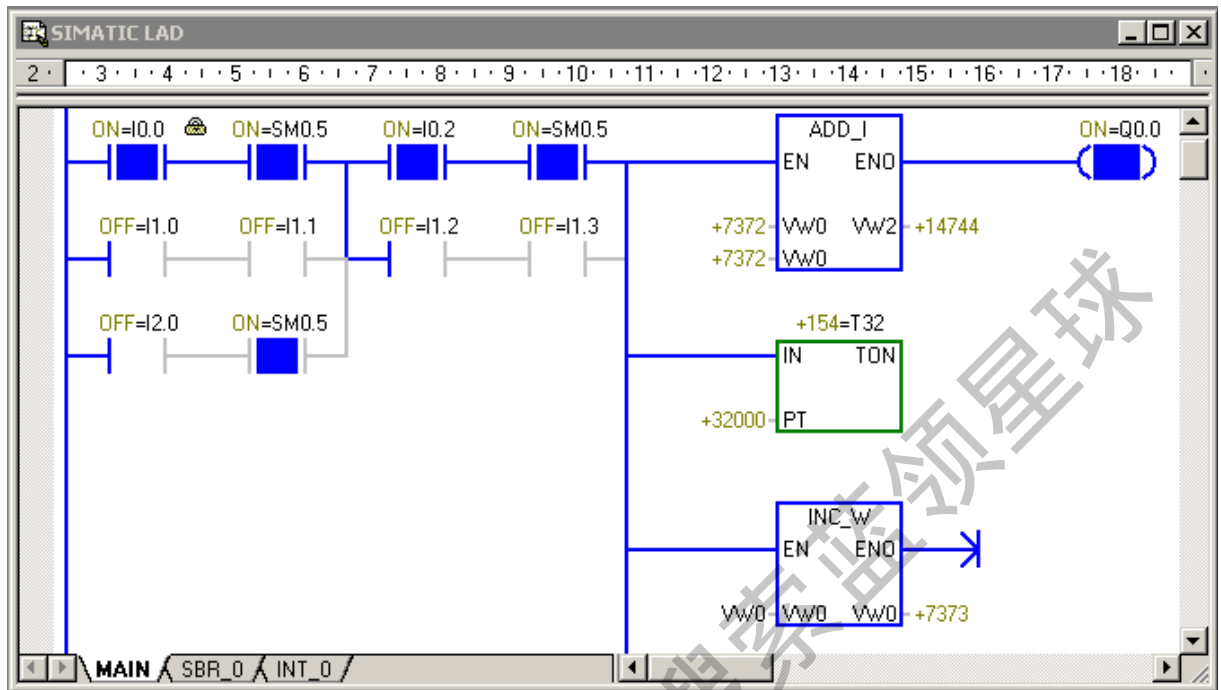
#### LAD图形中执行状态举例：

以下范例在LAD图形中使用LAD执行状态。状态图仅使用“扫描结束状态”数据收集。可在LAD图形和对应的状态图之间看出

The screenshot shows the '状态图' (Status Diagram) window. The title bar reads '状态图'. The main area contains a table with columns for '地址' (Address), '格式' (Format), '当前值' (Current Value), and '新数值' (New Value). The data is as follows:

地址	格式	当前值	新数值
1	I0.0	位 2#1	
2	I0.2	位 2#1	
3		带符号	
4	VW0	带符号 +7652	
5	T32	位 2#0	
6	T32	带符号 +0	

The status bar at the bottom shows 'CHT1'.

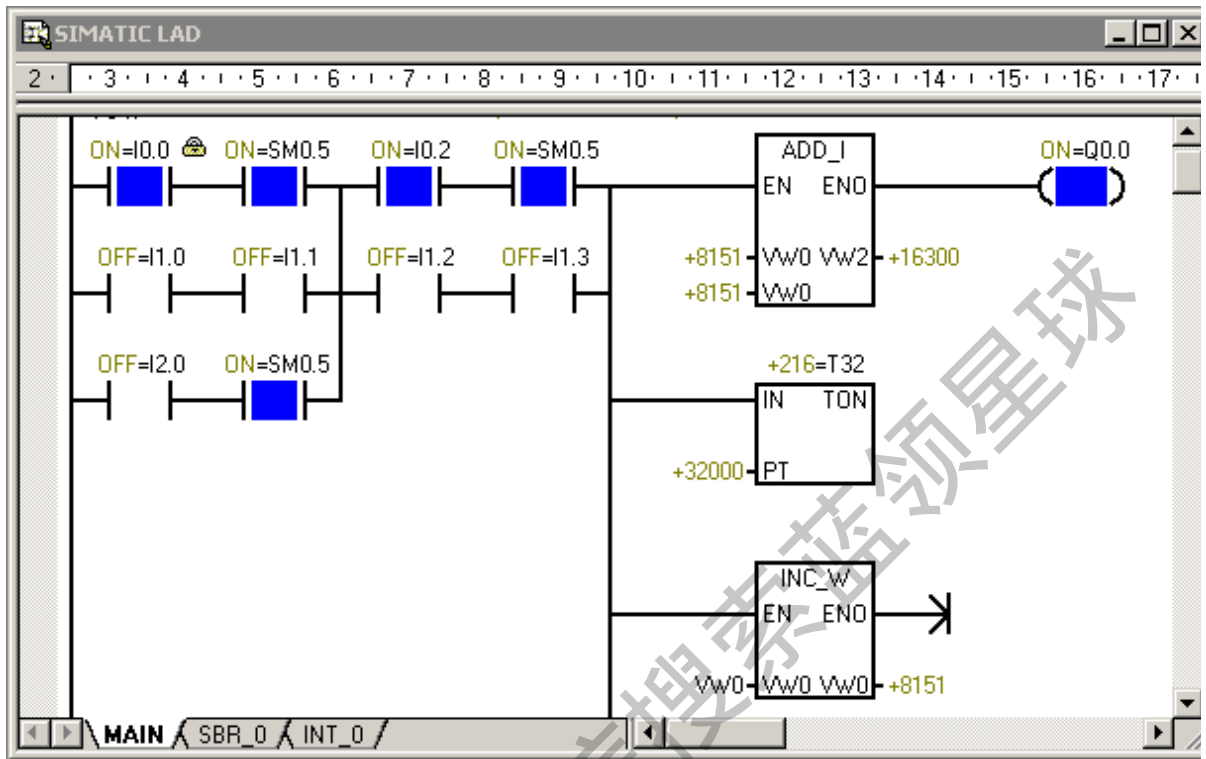


#### LAD图形中扫描结束状态举例：

以下范例在LAD图形中使用“扫描结束状态”。由于两个窗口（程序状态和图状态）均使用相同的扫描 PLC结束数据，两个窗口允鞠嗤 PLC数据值。这种情形会引起误解，因为程序可以在收集最终“扫描结束”数值之前为相同的位置指定很多数值。中间临时数值不显示。在该范例中，ADD\_I方框未能正确执行，因为输入相加超出了带符号字整数的最大范围。与红色方框的执行状态范例不同，您必须读取状态值，才能看到Vw0 + Vw0 <> VW2。

地址	格式	当前值	新数值
1	I0.0	位	2#1
2	I0.2	位	2#1
3		带符号	
4	VW0	带符号	+8418
5	T32	位	2#0
6	T32	带符号	+485

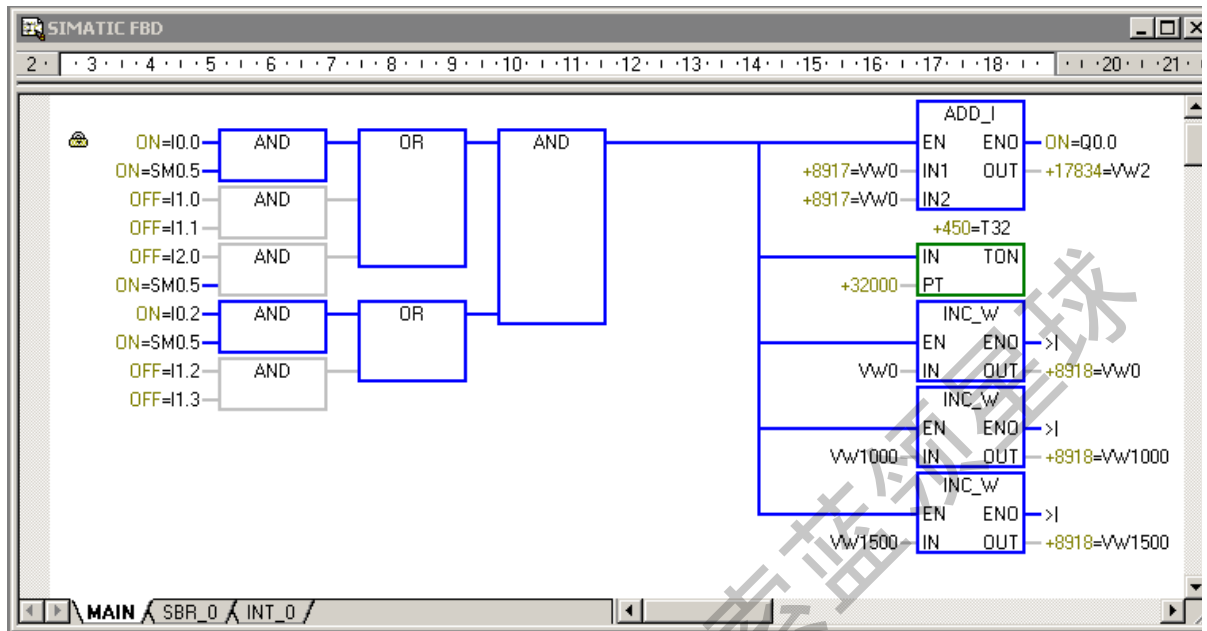




FBD图形中的执行状态举例:

地址	格式	当前值	新数值
1 I0.0	位	2#1	
2 I0.2	位	2#1	
3	带符号		
4 Vw0	带符号	+8958	
5 T32	位	2#0	
6 T32	带符号	+0	

获取更多资料



按照连续、瞬态图或触发（暂停程序状态）的方式收集状态值

[返回顶端](#)

#### 注释：

“连续”并非意味着实时；而是指编程设备不断地从 PLC 轮询状态信息，在屏幕中显示，并按按照通讯允许的速度更新显示。可能

#### 连续

- 打开程序编辑器窗口，并启动“程序状态”，当 PLC 位于 RUN（运行）模式时检视程序执行的连续状态更新。
- 打开状态图窗口，并启用“图状态”，当 PLC 位于 RUN（运行）模式时检视连续更新。



#### 瞬态图

- 将 PLC 转换为 STOP（停止）模式，收集单个状态更新。当 PLC 位于 STOP（停止）模式时，您可以使用“多次扫描”功能检视一次或多次扫描。您还可以引用网络中的特殊内存 SM0.1 首次扫描位，读取该网络的首次扫描状态值。
- 关闭图状态，如果您希望收集单次状态图更新并不希望将 PLC 转换至 STOP（停止）模式，使用“单次读取”功能。如 PLC 转换为 STOP（停止）模式，并开启“图状态”，您也会收集单次状态更新。此外，您可以在检视状态图时使用“多次扫描”和“首次扫描”功能。

#### 执行状态模式（LAD / FBD / STL）中的程序状态触发暂停

欲检视触发暂停 PLC 数据状态，您必须：

1. 将程序下载至 PLC，并将 PLC 设为 RUN（运行）模式。您还可以从 STOP（停止）模式开始触发暂停。如果您从 STOP（停止）模式开始，则可以捕获首次扫描。
2. 滚动程序编辑器窗口，显示您希望检视的程序网络。网络可以位于任何 POU（主程序、子例行程序或中断程序）中。
3. 使用 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单命令，将“使用执行状态”复选标记切换为打开。
4. 激活程序状态，开始状态数据通讯。



5. 单击“暂停程序状态” 工具条按钮，或用鼠标右键在程序编辑器中单击，并选择弹出菜单命令“暂停程序状态”。
6. 更新暂停之后，触发网络中的状态值在您检视时保持不变。当您准备移至另一个网络时，则单击“暂停程序状态” 工具条按钮，将暂停切换为关闭，并再次开始连续更新。

一旦获得新的状态信息，只要程序状态暂停功能被启用，该信息就会保持在屏幕上。在您取消选择程序状态暂停功能之前，无论 PLC 中实际变化速度多快，均不会进一步执行状态更新。如果没有程序状态暂停功能，屏幕就会不断刷新，状态数据在您读取和解释之前就可能消失（或者数据根本没有显示）。

#### PLC RUN / STOP（运行 / 停止）模式

[返回顶端](#)

使用以下一种方法改变PLC操作模式：

- 单击“运行”按钮，执行RUN（运行）模式，或者单击“停止”按钮，执行STOP（停止）模式。
- 选择**PLC > 运行 (RUN)** 菜单命令，执行RUN（运行）模式，或选择**PLC > 停止 (STOP)** 菜单命令，执行STOP（停止）模式。
- 以手动方式改变位于PLC上的模式开关。
- 在程序中插入一条STOP（停止）指令。

#### 注释：

欲使用STEP 7-Micro/WIN软件控制RUN / STOP（运行 / 停止）模式，必须在STEP 7-Micro/WIN和PLC之间存在一条通讯路径。此外，必须将PLC硬件模式开关设为TERM（终端）或RUN（运行）。将模式开关设为TERM（终端）不会改变PLC操作模式，但却允许STEP 7-Micro/WIN改变PLC操作模式。位于PLC前方的状态LED表示当前操作模式。当程序状态或状态图操作正在进行时，在STEP 7-Micro/WIN窗口右下方处附近的状态条上会出现一个RUN / STOP（运行 / 停止）指示灯。

虽然程序在STOP（停止）模式中不执行，PLC操作系统继续监控PLC（收集PLC RAM和I/O状态），将状态数据传递给STEP 7-Micro/WIN，并执行所有的“强制”或“取消强制”命令。当PLC位于STOP（停止）模式中时，您可以执行以下操作：

- 使用图状态或程序状态检视操作数的当前值。（此一步骤具有与执行“单次读取”相同的效果，因为程序未执行。）
- 您可以使用图状态或程序状态强制数值。使用图状态写入数值。
- 写入或强制输出。
- 执行有限次数扫描，并通过状态图和 / 或项目状态检视效果。

当PLC位于RUN（运行）模式时，您不得使用“首次扫描”或“多次扫描”功能。您可以在状态图中写入和强制数值，或使用LAD或FBD程序编辑器强制数值，方法与您在STOP（停止）模式中强制数值相同。您还可以执行以下操作（不得从STOP（停止）J 绞褂茫

- 使用图状态收集不断变化的PLC数据值的连续更新信息。（如果您希望使用单次更新，图状态必须关闭，才能使用“单次读取”命令。）
- 使用程序状态收集不断变化的PLC数据值的连续更新信息。
- 使用RUN（运行）模式功能中的“程序编辑”编辑程序，并将改动下载至PLC。

#### 状态通讯与扫描循环

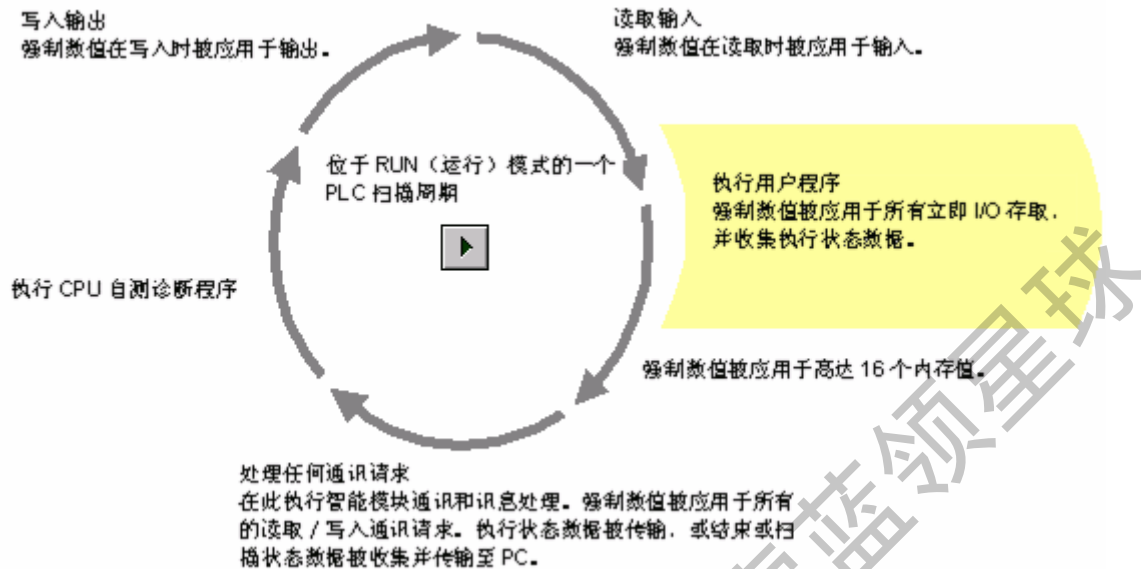
[返回顶端](#)

PLC在连续循环中读取输入、执行程序逻辑、写入输出和执行系统操作和通讯。该扫描循环速度极快，每秒执行多次。虽然STEP 7-Micro/WIN会快速发出状态请求，您应当认识到您检视的不是PLC中出现的每一个事件，这一点十分重要。

由于PLC和您观察的程序状态编程设备之间存在通讯时间滞后，您看到的显示的操作数数值总是在状态显示中改变之前即在PLC中改变，但没有在程序状态表示中改变。

如果您使用“扫描结束”状态模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被取消勾选时），您在几个扫描循环中采集数据。

如果您使用“执行状态”模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选时），所有显示的程序状态值一定来自同一个扫描循环。



**注释:**

一次只有一个状态窗口被授权运行状态。如果您尝试在任何其他与同一个 PLC 连接的 STEP 7 Micro/WIN 应用程序中运行状态，则会收到错误信息。

**模拟进程条件（读取、写入、强制、取消强制）**

[返回顶端](#)

您可以模拟进程条件，方法是在程序状态操作过程中从程序编辑器和从状态图向操作数写入或强制新数值。

使用“程序状态”和“图状态”按钮或“调试”菜单命令，开始状态数据通讯，并启用调试工具。您还可以使用硬件输入模拟器开关（西门子（Siemens）可提供）以手动方式在数字输入打开和关闭之间切换，并观察 PLC 上的 LED 指示灯，观察数字输出的条件。

从“调试”工具条或“调试”菜单列表存取以下功能。






**单次读取**（仅限状态图）如果您希望获得一次“瞬态图”（对状态图中的所有数值一次更新），使用“单次读取”。默认值为状态图连续轮询 PLC，获取状态更新信息。当您单击“状态图”时，状态图会切换为关闭，“单次读取”图标变为黄色。




**全部写入**（仅限状态图）

在您完成对几行的“新数值”列的改动后，您可以使用“全部写入”，将所有需要的改动发送至 PLC。



**强制** 您可以在操作程序状态时从程序编辑器和状态图强制地址。欲强制程序状态地址，用鼠标右键单击一个参数，并选择从菜单列表强制。欲强制状态图地址为某一数值，您必须首先规定所需的数值，方法是读取该数值（如果您希望强制当前值）或键入该数值（如果您希望将地址强制为一个新数值）。您一旦使用了强制功能，则在每次扫描时该数值均被重新应用于地址，直至您取消强制地址。“强制”功能盖写立即读取或立即写入指令。“强制”功能还盖写被配置为在转换为 STOP（停止）模式时进入一个指定数值的输出：如果 CPU 进入 STOP（停止）模式，输出被设为强制数值，而非配置的数值。通过将 V 或 M 内存强制为字节、字或双字，将 AI 或 AQ 内存在偶数字节边界上（例如 AIW6 或 AIW14）或 I/O 点上（I 和 Q 位址）强制为字，模拟逻辑条件。您可以一次强制 16 个（V、M、AI 或 AQ）地址和所有的 I/O 位（所有 I 和 Q 位址）。 该图标表示该地址被明确强制。该地址数值在地址被取消强制之前无法改变。 该图标表示该地址被隐含强制。如果地址是一个被明确强制的较大地址的一部分，该地址则被认为是隐含强制。例如，如果 VW0 被强制，则 VB0 是隐含强制（VB0 是 VW0 的第一个字节）。隐含强制数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制 VD0（该地址包含 VB0、VB1、VB2 和 VB3），则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制，则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。



**部分强制** 该图标表示该地址的一部分被部分强制。例如，如果 VW0 被明确强制，则 VW1 的一部分被强制（VW1 的第一个字节是 VW0 的第二个字节）。被部分强制的数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制 VD0（该地址包含 VB0、VB1、VB2 和 VB3），则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制，则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。



**取消强制** 对于程序状态和状态图，选择一个地址，并使用“取消强制”按钮从该特定地址移除

强制功能。您还可以选择（单击）一个参数，然后用鼠标右键单击该参数，查看“强制”和“取消强制”功能的弹出菜单。



#### 全部取消强制

如果您希望从所有的地址中移除强制功能，使用“全部取消强制”按钮。在应用“全部取消强制”之前，您无须选择单个地址。



#### 读取全部强制（仅限状态图）

当您使用“读取全部强制”功能时，状态图的“当前数值”列会为已经明确强制、隐含强制或部分隐含强制的所有地址显示一个图标。

### 检查交叉引用和元素用法

[返回顶端](#)

调试程序时，您可以决定是否增加、删除或编辑参数。

使用“交叉引用”窗口查看程序中当前指定参数的方法，这样可以防止您意外地重复赋值。

此外，如果您希望在RUN（运行）模式中执行程序编辑，并且某些网络使用正向或负向转换（EP和N接点或方框以及STL EU或ED指令），您一定要检查“交叉引用”信息中的EU/ED数目，以防重复指令号码。

### 向程序下载改动

[返回顶端](#)

一旦调试部分代码，如果PLC位于STOP（停止）模式，您可以编辑程序、下载程序、然后再次监控状态，查看改动是否发生作用。

您不必进入STOP（停止）模式就可以在RUN（运行）模式中执行程序编辑并向PLC下载较小的改动。



**警告** 在RUN（运行）模式中向PLC下载改动时，您的改动会立即影响程序操作。不得出错；程序编辑中的错误可能导致人员死亡或严重伤害和/或设备损坏。仅限合格人员在RUN（运行）模式中执行程序编辑。

### 重设用于调试和编辑的帧设置

[返回顶端](#)

欲重设将窗口和工具条的可视性、尺寸和位置设为默认值，选择**检视（View）> 帧（Frame）> 全部重设（Reset All）**菜单命令。

您有两套不同的工作空间窗口的便利，可在调试程序时获得较大的灵活性：窗口可视性和定位。这是比正常编辑空间更大的检视空间。当您进入程序状态时，您已经设置的任何窗口定位保持初始进入的状态；如果您关闭程序状态，窗口保持初始进入的状态。同样，任何在编辑程序时执行的窗口定位均在运行时与程序状态分开维护。

[返回顶端](#)

#### 另请参阅：

[如何在程序编辑器中显示状态（GS 7.2）](#)

[如何在状态图中显示状态（GS 7.3）](#)

[如何执行有限数目扫描（GS 7.4）](#)

[如何下载程序（GS 6.3）](#)

[时间标记不匹配错误](#)（确保编程设备中的项目与PLC中的项目相匹配）

[交叉引用和元素用法](#)（确保程序编辑不引起重复赋值）

[在运行模式中执行程序编辑](#)

[PLC运行/停止模式](#)

[在停止模式中写入和强制输出](#)

[安装和移除S7-200设备](#)

[接地和布线指南](#)

## 3.3.21 保存作业



您可以使用工具条上的“保存”按钮保存作业，或从“文件”菜单选择“保存”和“另存为”选项保存作业。



- “保存”允许您在作业中快速保存所有改动。（然而，您初次保存一个项目时，会被提示核实或修改当前项目名称和柯嫉哪 涎∠睢#
- “另存为”允许您修改当前项目的名称和 / 或目录位置。

当您首次创建项目时，STEP 7-

Micro/WIN的默认值名称为**Project1.mwp**。您可以接受或修改该名称；如果您接受该名称，下一个项目的默认名称将自动递增为 **Project2.mwp**。

STEP 7-Micro/WIN项目的默认目录位置是位于STEP-7-MicroWIN目录中的称作“项目”的文件夹，您可以不接受该默认位置。

获取更多资料 微信搜索 蓝领星球

获取更多资料 微信搜索蓝领星球

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part

# IV

获取更多资料 微信搜索蓝领星球



## 4 LAD、STL和FBD指令集

### 4.1 ENO用法

ENO是LAD和FBD

中方框的布尔输出。ENO允许您以串行（水平方向）方式连接方框，不允许以并行（垂直方向）方式连接方框。如果方框在 EU 输入位置有功率流，且方框执行无错误，则 ENO 输出将功率流传输至下一个元素。如果方框执行过程中检测到错误，则在生成错误的方框位置终止功率流。

ENO与EN（EN=ENO）具有相同的状态图，除非操作执行有误。例如，DIV\_I指令为除数为0提供ENO=0。

STL中没有ENO输出，但与带ENO输出的LAD和FBD指令相对的STL指令设置一个特殊的ENO位。可使用STL指令 [AENO](#)、[CAND](#)、[ENO](#) 存取该位，该位也可用于生成与方框ENO位相同的效果。

### 4.2 使用IEC超载和非标准指令

#### IEC超载指令

以下列举在IEC模式中编程和使用超载指令的要点。

- 超载指令的输入参数与输出参数的尺寸必须一致：例如，如果您的输入是字数值，则输出也必须为字数值，而不能是字节数值或双字数值。
- 超载指令的任何变量操作数必须与该指令的其他参数数据类型相同。
- 至少一个操作数必须是变量并指定数据类型。

下列IEC指令属于超载指令。

ADD	AND	SHL	LT
SUB	OR	SHR	GT
MUL	XOR	ROL	LE
DIV	NOT	ROR	GE
INC	MOVE	EQ	
DEC	BLKMOVE	NE	

#### IEC非标准指令

指令树中指令前面的红色菱形符号◆表示非标准IEC指令。

另请参阅：

[编译器数据类型检查](#)

### 4.3 受S7-200 CPU模型支持的指令

#### S7-21x CPU指令

[SIMATIC程序](#)

[STL](#) [语句表](#)  
[LAD和FBD](#) [梯形图和功能块图](#)

[IEC-1131程序](#)

[LAD和FBD](#) [梯形图和功能块图](#)

#### S7-22x CPU指令

[SIMATIC程序](#)

[STL](#) [语句表](#)  
[LAD和FBD](#) [梯形图和功能块图](#)

[IEC-1131程序](#)

[LAD和FBD](#) [梯形图和功能块图](#)

### 4.4 SIMATIC / 国际助记符

说明

SIMATIC

国际

梯形图形	KOP		LAD	
功能块图形		FUP		FBD
语句列表	AWL		STL	
输入	E		I	
输出	A		Q	
变量内存	V		V	
内存	M		M	
模拟输入	AE		AI	
模拟输出	AA		AQ	
计时器	T		T	
计数器	Z		C	
高速计数器		HC		HC
SCR	S		S	
特殊内存	SM		SM	
累加器	AC		AC	
局部变量内存		L		L

另请参阅：

[编址概述 \(GS 2.2\)](#)

[内存类型和属性](#)

[直接和间接编址](#)

[CPU内存地址范围](#)

[特殊内存地址](#)

[符号表 / 全局变量表](#)

## 4.5 SIMATIC LAD指令

### 4.5.1 位逻辑

指令

位逻辑





### 4.5.2 时钟



SET\_RTC 设置实时时钟

### 4.5.3 通讯



### 4.5.4 比较





获取更多资料 微信搜索 蓝领星球



### SIMATIC LAD转换指令

[[[ENDKEEPN]]]

#### 4.5.5 转换

指令

转换



获取更多资料 微信搜索蓝领星球



### 4.5.6 计数器



### 4.5.7 浮点数学



获取更多资料 微信搜索 蓝领星球



### 4.5.8 整数数学

指令

整数数学



获取更多资料 微信搜索蓝领星球

### 4.5.9 中断

指令  
中断



### 4.5.10 逻辑操作

指令  
逻辑操作



获取更多资料 微信搜索 蓝领星球



### 4.5.11 移动



### 4.5.12 程序控制



获取更多资料 微信搜索 蓝领星球



### 4.5.13 移位 / 旋转



### 4.5.14 字符串





白...指令  
白...比较



白...指令  
白...转换



### 4.5.15 表

白...指令  
白...表格



微信搜索蓝领星球

## 4.5.16 计时器



## 4.5.17 子例行程序

输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
SBR_N	0至63	常量 用于CPU?21、CPU?22、CPU?24： 用于CPU 224XP和CPU
226		0至127 字[[[ENDKEEPN]]]
IN		V, I, Q, M, SM, S, T, C, L, 功率流 VB, IB, QB, MB, SMB, SB, LB,
AC, *VD, *LD, *AC, 常量 VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, 常量 VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, &VB, &IB, &QB, &MB, &T, &C, &SB, &AI, &AQ, &SMB, 常量 *VD, *LD, *AC,		布尔字节字、整数双字、双整数 字符串[[[ENDKEEPN]]]
常量		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
IN/OUT		*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD,
*VD, *LD, *AC		布尔字节字、整数双字、双整数[[[ENDKEEPN]]]
OUT		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AQW, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD,		布尔字节字、整数双字、双整数[[[ENDKEEPN]]]
*LD, *AC		

注释：AC地址必须不少于1SM地址必须不少于30[[[ENDKEEPN]]]

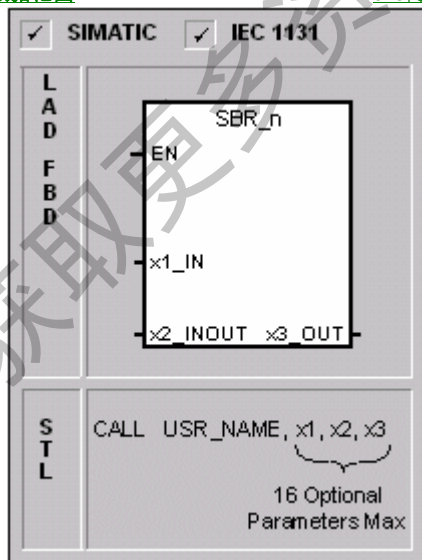
内存范围 错误 S7-200 CPU指令支持

SIMATIC / 国际助记符

[[[ENDKEEPN]]]

数据范围 CPU内存中的指令大小

编址内存 [[[ENDKEEPN]]]



**调用子例行程序 (CALL)** 指令将控制转换给子例行程序 (SBR\_n)。您可以使用带参数或不带参数的“调用子例行程序”指令。在子例行程序完成执行后，控制返回至“调用子例行程序”之后的指令。每个子例行程序调用的输入 / 输出参数最大限制为16。如果您尝试下载的程序超过此一限制，会返回一则错误讯息。

如果您为子例行程序指定一个符号名，例如 USR\_NAME，该符号名会出现在指令树的“子例行程序”文件夹中。将参数值指定给局部内存的顺序由 CALL 指定，参数从 L.0 开始。2.

一至八个连续位参数值被指定给从 Lx.0 开始持续至 Lx.7 的单字节。3.

字节、字和双字数值被指定给局部内存，位于字节边界（LBx、LWx 或 LDx）位置。在带参数的“调用子例行程序”指令中，问 既朕肱永 谱缘蚓植勘淤

表中定义的变量完全匹配。参数顺序必须以输入参数开始，其次是输入 / 输出参数，然后是输出参数。为带安数 =

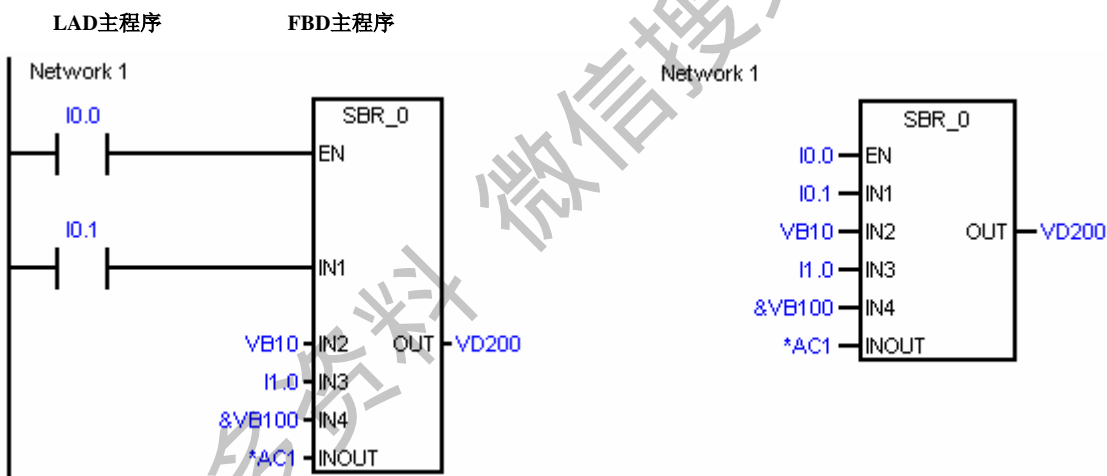
0调用设置 ENO 的错误条件：0008 超出最大子例行程序嵌套

注释：位于指令树中的子例行程序名称的工具提示显示每个参数的名称。

#### 子例行程序调用举例

用于 SBR\_0 的局部变量表

	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
LB1	IN2	IN	BYTE	
L2.0	IN3	IN	BOOL	
LD3	IN4	IN	DWORD	
		IN		
LD7	INOUT	IN_OUT	REAL	
		IN_OUT		
LD11	OUT	OUT	REAL	
		OUT		



由 Micro/WIN 从 LAD/FBD 图形建立的 STL 代码，可在 LAD、FBD 或 STL 视图中启用显示

注释：Micro/WIN 保留 L 内存的最上方的 4 个字节（LB60-LB63），用于调用参数数据。

```

NETWORK 1
// L内存被用于保存布尔输入参数状态，
// 此类参数在LAD和FBD中被显示为功率流输入。
// 从而允许本网络在LAD、FBD和STL编辑器中显示。
LD I0.0
= L60.0
LD I0.1
= L63.7
LD L60.0
CALL SBR_0 L63.7 VB10 I1.0 &VB100 *AC1 VD200

```

用 STL 编辑器输入具有与以上显示相同的子例行程序的 STL 范例 注释：STL 程序员可使用该简化的调用程序。

```

NETWORK 1
// 本网络只能在STL编辑器中显示，
// 因为用作功率流输入的布尔参数
// 未在L内存中保存。
LD I0.0
CALL SBR_0 I0.1 VB10 I1.0 &VB100 *AC1 VD200

```

另请参阅：  
[子例程序用法详情](#)

## 4.6 SIMATIC FBD指令

### 4.6.1 位逻辑

指令  
位逻辑



### 4.6.2 时钟

指令  
时钟



### 4.6.3 通讯

指令  
通讯



#### 4.6.4 比较

白...指令  
白...比较



获取更多资料 微信搜索 蓝领星球



#### 4.6.5 转换



获取更多资料 微信搜索 蓝领星球





### SIMATIC FBD计数器指令

[[[ENDKEEPN]]]

#### 4.6.6 计数器

- 指令
- +1 计数器



### 4.6.7 浮点数学

指令  
浮点数学



### 4.6.8 整数数学

指令  
整数数学



获取更多资料 微信搜索 蓝领星球



#### 4.6.9 中断



#### 4.6.10 逻辑操作



微信搜索蓝领星球

获取更多资料



#### 4.6.11 移动



#### 4.6.12 程序控制





#### 4.6.13 移位 / 旋转

指令

移位 / 旋转



微信搜索蓝领星球

获取更多资料



#### 4.6.14 字符串

指令  
字符串



指令  
比较



指令  
转换





### 4.6.15 表

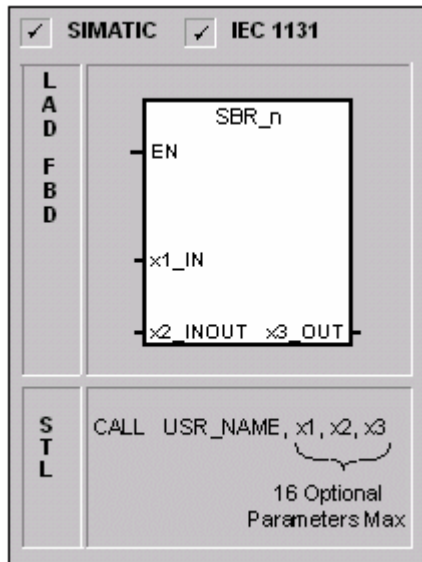


### 4.6.16 计时器



### 4.6.17 子例行程序

输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
SBR_N		常量 用于CPU?21 、CPU?22 、CPU?24 :
0至63		用于CPU 224XP和CPU
226		0至127 字[[[ENDKEEPN]]]
IN		V, I, Q, M, SM, S, T, C, L, 功率流 VB, IB, QB, MB, SMB, SB, LB,
AC, *VD, *LD, *AC, 常量 VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, 常量 VD, ID, QD, MD, SMD, SD,		&SB, &AI, &AQ, &SMB, 常量 *VD, *LD, *AC,
LD, AC, HC, *VD, *LD, *AC, &VB, &IB, &QB, &MB, &T, &C,		布尔 字节 字、整数 双字、双整数 字符串[[[ENDKEEPN]]]
常量		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
IN/OUT		*VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD,
*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, *VD, *LD, *AC		布尔 字节 字、整数 双字、双整数[[[ENDKEEPN]]]
OUT		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AQW, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD,		布尔 字节 字、整数 双字、双整数[[[ENDKEEPN]]]
*LD, *AC		
注释: AC地址必须不少于1SM地址必须不少于30[[[ENDKEEPN]]]		
<a href="#">内存范围</a>	<a href="#">错误</a>	<a href="#">S7-200 CPU指令支持</a>
[[[ENDKEEPN]]]		<a href="#">SIMATIC / 国际助记符</a>
<a href="#">数据范围</a>		<a href="#">编址内存 [[[ENDKEEPN]]]</a>
		<a href="#">CPU内存中的指令大小</a>



**调用子例行程序 (CALL)** 指令将控制转换给子例行程序 (SBR\_n)。您可以使用带参数或不带参数的“调用子例行程序”指令。在子例行程序完成执行后，控制返回至“调用子例行程序”之后的指令。每个子例行程序调用的输入 / 输出参数最大限制为16。如果您尝试下载的程序超过此一限制，会返回一则错误讯息。

如果您为子例行程序指定一个符号名，例如 **USR\_NAME**，该符号名会出现在指令树的“子例行程序”文件夹中。将参数值指定给子例行程序的输入 / 输出参数。1. 参数值指定给局部内存的顺序由 **CALL** 指定，参数从 **L.0** 开始。2. 一至八个连续位参数值被指定给从 **Lx.0** 开始持续至 **Lx.7** 的单字节。3.

字节、字和双字数值被指定给局部内存，位于字节边界 (**LBx**、**LWx** 或 **LDx**) 位置。在带参数的“调用子例行程序”指令中，参数顺序必须以输入参数开始，其次是输入 / 输出参数，然后是输出参数。为带参数 =

**调用设置 ENO 的错误条件：0008** 超出最大子例行程序嵌套

**注释：** 位于指令树中的子例行程序名称的工具提示显示每个参数的名称。

#### 子例行程序调用举例

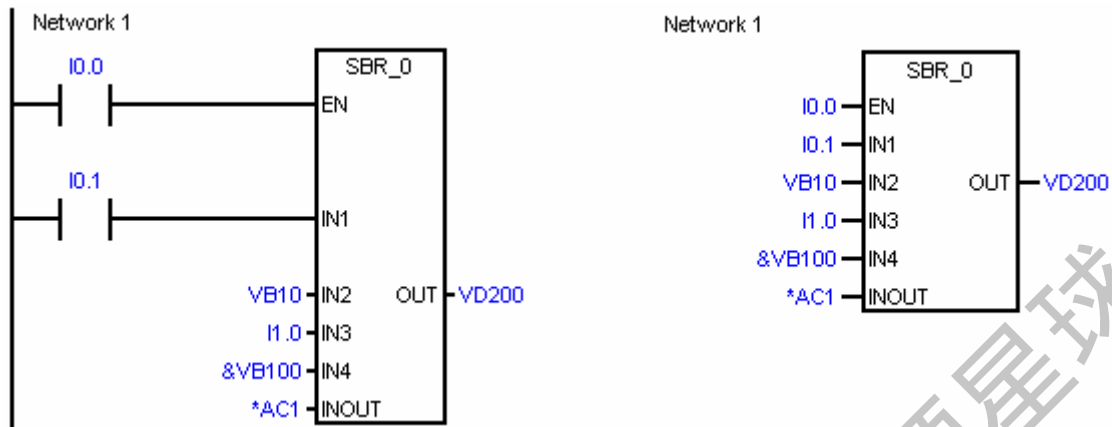
用于 **SBR\_0** 的局部变量表

	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
<b>L0.0</b>	IN1	IN	BOOL	
<b>LB1</b>	IN2	IN	BYTE	
<b>L2.0</b>	IN3	IN	BOOL	
<b>LD3</b>	IN4	IN	DWORD	
		IN		
<b>LD7</b>	INOUT	IN_OUT	REAL	
		IN_OUT		
<b>LD11</b>	OUT	OUT	REAL	
		OUT		

LAD主程序

FBD主程序





由Micro/WIN从LAD/FBD图形建立的STL代码，可在LAD、FBD或STL视图中启用显示  
 注释：Micro/WIN保留L内存的最上方的4个字节（LB60-LB63），用于调用参数数据。

```

NETWORK 1
// L内存被用于保存布尔输入参数状态，
// 此类参数在LAD和FBD中被显示为功率流输入。
// 从而允许本网络在LAD、FBD和STL编辑器中显示。
LD I0.0
= L60.0
LD I0.1
= L63.7
LD L60.0
CALL SBR_0 L63.7 VB10 I1.0 &VB100 *AC1 VD200
  
```

用STL编辑器输入具有与以上显示相同的子例程序的STL范例 注释：STL程式员可使用该简化的调用程序。

```

NETWORK 1
// 本网络只能在STL编辑器中显示，
// 因为用作功率流输入的布尔参数
// 未在L内存中保存。
LD I0.0
CALL SBR_0 I0.1 VB10 I1.0 &VB100 *AC1 VD200
  
```

另请参阅：  
[子例行程序用法详情](#)

## 4.7 SIMATIC STL指令

### 4.7.1 位逻辑

指令  
 位逻辑





获取更多资料 微信搜索 蓝领星球



### SIMATIC STL时钟指令

[[[ENDKEEPN]]]

指令

#### 4.7.2 时钟

指令

时钟



#### 4.7.3 通讯

指令

通讯



#### 4.7.4 比较

指令

比较



微信搜索 蓝领星球  
获取更多资料



LDB<= IN1, IN2 载入字节小于或等于



获取更多资料 微信搜索蓝领星球



... LDD<= IN1,IN2 载入双字小于或等于

获取更多资料 微信搜索蓝领星球



获取更多资料 微信搜索 蓝领星球



#### 4.7.5 转换



获取更多资料 微信搜索 蓝领星球



#### 4.7.6 计数器



#### 4.7.7 浮点数学







#### 4.7.8 整数数学

指令  
整数数学



获取更多资料 微信搜索 蓝领星球



### 4.7.9 中断



### 4.7.10 逻辑操作





#### 4.7.11 移动



#### 4.7.12 程序控制





#### 4.7.13 移位 / 旋转

指令  
移位 / 旋转





### 4.7.14 字符串

指令  
字符串



指令  
比较



OS= IN1, IN2 OR (或) 字符串等于



指令  
转换



获取更多资料 微信搜索 蓝领星球



### SIMATIC STL表格指令

[[[ENDKEEPN]]]

#### 4.7.15 表



#### 4.7.16 计时器



## 4.7.17 子例行程序

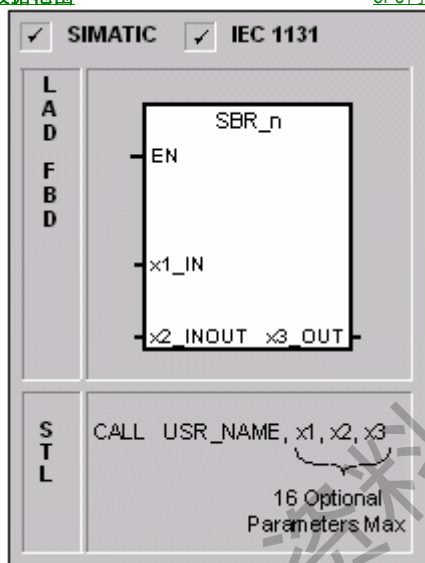
输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
SBR_N		常量 用于CPU?21 、 CPU?22 、 CPU?24 : 用于CPU 224XP和CPU
0至63		字[[[ENDKEEPN]]]
226		0至127
IN		V, I, Q, M, SM, S, T, C, L, 功率流 VB, IB, QB, MB, SMB, SB, LB,
AC, *VD, *LD, *AC, 常量 VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, 常量 VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, &VB, &IB, &QB, &MB, &T, &C, &SB, &AI, &AQ, &SMB, 常量 *VD, *LD, *AC, 布尔字节字、整数 双字、双整数 字符串[[[ENDKEEPN]]]		
IN/OUT		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, 布尔字节字、整数 双字、双整数[[[ENDKEEPN]]]		
OUT		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AQW, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC 布尔字节字、整数 双字、双整数[[[ENDKEEPN]]]		

注释: AC地址必须不少于1SM地址必须不少于30[[[ENDKEEPN]]]

内存范围 错误 S7-200 CPU指令支持 SIMATIC / 国际助记符

[[[ENDKEEPN]]]

数据范围 CPU内存中的指令大小 编址内存 [[[ENDKEEPN]]]



**调用子例行程序 (CALL)** 指令将控制转换给子例行程序 (SBR\_n)。您可以使用带参数或不带参数的“调用子例行程序”指令。在子例行程序完成执行后，控制返回至“调用子例行程序”之后的指令。每个子例行程序调用的输入 / 输出参数最大限制为16。如果您尝试下载的程序超过此一限制，会返回一则错误讯息。

如果您为子例行程序指定一个符号名，例如 **USR\_NAME**，该符号名会出现在指令树的“子例行程序”文件夹中。将参数值指定给局部内存的顺序由 **CALL** 指定，参数从 **L.0** 开始。

1. 参数值指定给局部内存的顺序由 **CALL** 指定，参数从 **L.0** 开始。  
2. 一至八个连续位参数值被指定给从 **Lx.0** 开始持续至 **Lx.7** 的单字节。  
3. 字节、字和双字数值被指定给局部内存，位于字节边界 (**LBx**、**LWx** 或 **LDx**) 位置。在带参数的“调用子例行程序”指令中，表中定义的变量完全匹配。参数顺序必须以输入参数开始，其次是输入 / 输出参数，然后是输出参数。

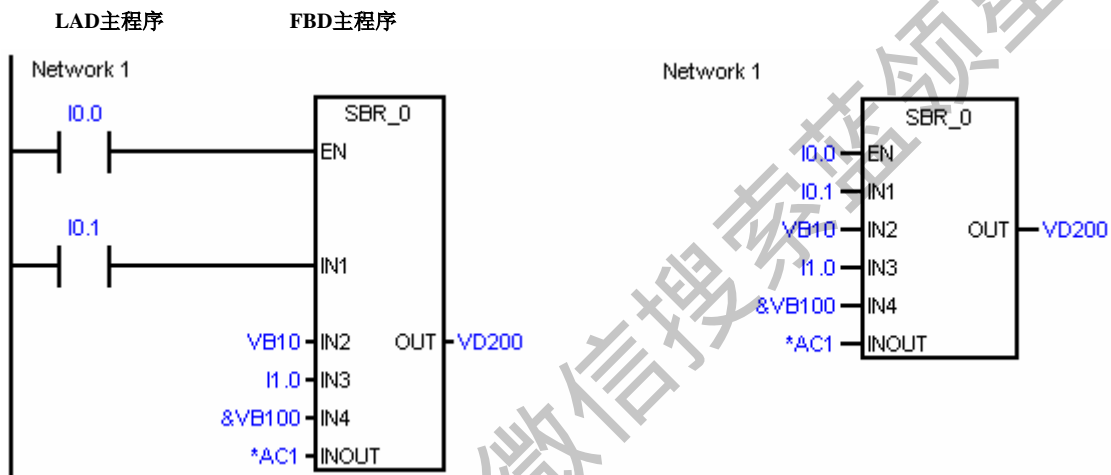
**为带参数 = 0调用设置ENO的错误条件: 0008 超出最大子例行程序嵌套**

**注释:** 位于指令树中的子例行程序名称的工具提示显示每个参数的名称。

## 子例行程序调用举例

用于 **SBR\_0** 的局部变量表

	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
LB1	IN2	IN	BYTE	
L2.0	IN3	IN	BOOL	
LD3	IN4	IN	DWORD	
		IN		
LD7	INOUT	IN_OUT	REAL	
		IN_OUT		
LD11	OUT	OUT	REAL	
		OUT		



由Micro/WIN从LAD/FBD图形建立的STL代码，可在LAD、FBD或STL视图中启用显示  
 注释：Micro/WIN保留L内存的最上方的4个字节（LB60-LB63），用于调用参数数据。

```

NETWORK 1
// L内存被用于保存布尔输入参数状态，
// 此类参数在LAD和FBD中被显示为功率流输入。
// 从而允许本网络在LAD、FBD和STL编辑器中显示。
LD I0.0
= L60.0
LD I0.1
= L63.7
LD L60.0
CALL SBR_0 L63.7 VB10 I1.0 &VB100 *AC1 VD200
  
```

用STL编辑器输入具有与以上显示相同的子例行程序的STL范例 注释：STL程序员可使用该简化的调用程序。

```

NETWORK 1
// 本网络只能在STL编辑器中显示，
// 因为用作功率流输入的布尔参数
// 未在L内存中保存。
LD I0.0
CALL SBR_0 I0.1 VB10 I1.0 &VB100 *AC1 VD200
  
```

另请参阅：  
[子例行程序用法详情](#)



## 4.8 IEC LAD指令

### 4.8.1 位逻辑



注释：  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

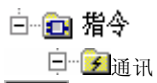
### 4.8.2 时钟





注释：  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.8.3 通讯



注释：  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.8.4 比较





另请参阅:

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

[使用IEC超载和非标准指令](#)

### 4.8.5 转换

指令

转换



获取更多资料 微信搜索 蓝领星球



注释：  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.8.6 计数器



注释：  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.8.7 中断



注释：

每个SIMATIC指令的红色菱形块♦表示已扩展至标准IEC 1131-3指令集。

#### 4.8.8 逻辑操作

指令  
逻辑操作



另请参阅：  
[使用IEC超载和非标准指令](#)

#### 4.8.9 数学

指令  
数学



注释：  
每个SIMATIC指令的红色菱形块♦表示已扩展至标准IEC 1131-3指令集。

另请参阅：  
[使用IEC超载和非标准指令](#)

#### 4.8.10 移动

指令  
移动





**注释:**  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

**另请参阅:**  
[使用IEC超载和非标准指令](#)

### 4.8.11 数字



**注释:**  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.8.12 程序控制





注释:

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.8.13 移位 / 旋转

指令

移位 / 旋转



注释:

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

另请参阅:

[使用IEC超载和非标准指令](#)

### 4.8.14 字符串

指令

字符串





指令  
比较



指令  
转换



注释：  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.8.15 表

指令  
表格



注释：  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。



## 4.8.16 计时器



## 4.8.17 子例行程序

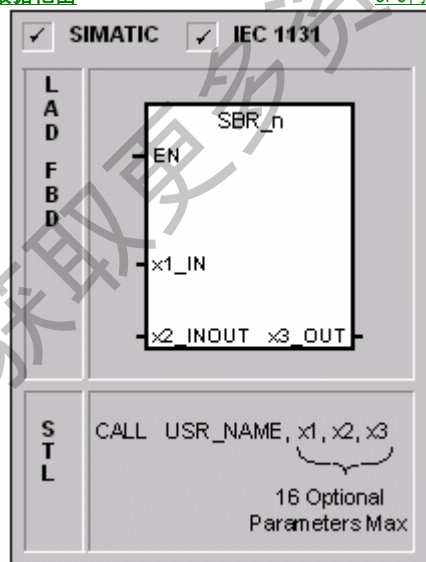
输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
SBR_N		常量 用于CPU?21、CPU?22、CPU?24： 用于CPU 224XP和CPU
0至63		字[[[ENDKEEPN]]]
226		0至127 字[[[ENDKEEPN]]]
IN		V, I, Q, M, SM, S, T, C, L, 功率流 VB, IB, QB, MB, SMB, SB, LB,
AC, *VD, *LD, *AC, 常量 VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, 常量 VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, &VB, &IB, &QB, &MB, &T, &C, &SB, &AI, &AQ, &SMB, 常量 *VD, *LD, *AC,		布尔字节字、整数双字、双整数字符串[[[ENDKEEPN]]]
常量		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
IN/OUT		*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD,
*VD, *LD, *AC		布尔字节字、整数双字、双整数[[[ENDKEEPN]]]
OUT		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AQW, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC		布尔字节字、整数双字、双整数[[[ENDKEEPN]]]

注释：AC地址必须不少于1SM地址必须不少于30[[[ENDKEEPN]]]

内存范围 错误 S7-200 CPU指令支持 SIMATIC / 国际助记符

[[[ENDKEEPN]]]

数据范围 CPU内存中的指令大小 编址内存 [[[ENDKEEPN]]]



调用子例行程序（CALL）指令将控制转换给子例行程序（SBR\_n）。您可以使用带参数或不带参数的“调用子例行程序”指令。在子例行程序完成执行后，控制返回至“调用子例行程序”之后的指令。每个子例行程序调用的输入/输出参数最大限制为16。如果您尝试下载的程序超过此一限制，会返回一则错误讯息。

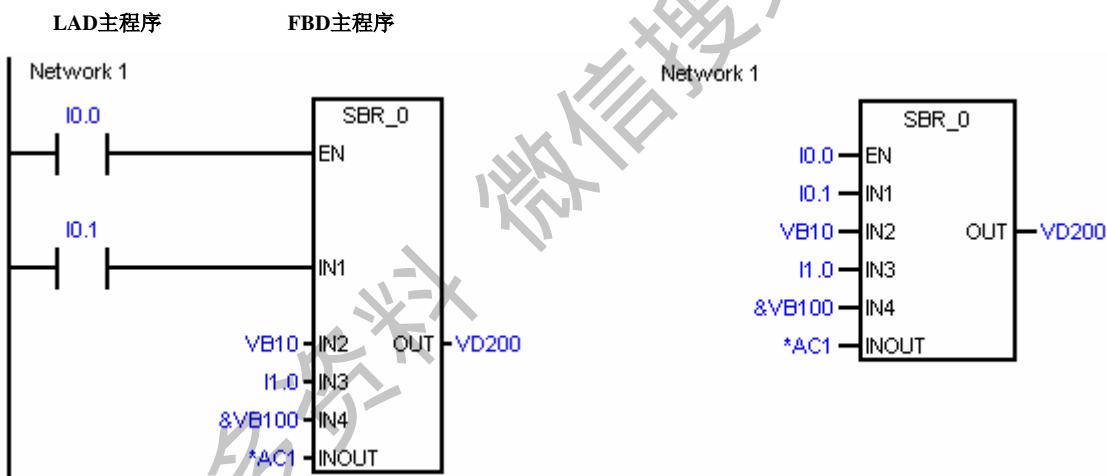
如果您为子例行程序指定一个符号名，例如 `USR_NAME`，该符号名会出现在指令树的“子例行程序”文件夹中。将参数值指定给局部内存的顺序由 `CALL` 指定，参数从 `L.0` 开始。1. 参数值指定给局部内存的顺序由 `CALL` 指定，参数从 `L.0` 开始。2. 一至八个连续位参数值被指定给从 `Lx.0` 开始持续至 `Lx.7` 的单字节。3. 字节、字和双字数值被指定给局部内存，位于字节边界（`LBx`、`LWx`或`LDx`）位置。在带参数的“调用子例行程序”指令中，表中定义的变量完全匹配。参数顺序必须以输入参数开始，其次是输入 / 输出参数，然后是输出参数。为带安数 = 0调用设置 `ENO` 的错误条件：0008 超出最大子例行程序嵌套

注释：位于指令树中的子例行程序名称的工具提示显示每个参数的名称。

## 子例行程序调用举例

用于 `SBR_0` 的局部变量表

	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
<code>L0.0</code>	IN1	IN	BOOL	
<code>LB1</code>	IN2	IN	BYTE	
<code>L2.0</code>	IN3	IN	BOOL	
<code>LD3</code>	IN4	IN	DWORD	
		IN		
<code>LD7</code>	INOUT	IN_OUT	REAL	
		IN_OUT		
<code>LD11</code>	OUT	OUT	REAL	
		OUT		



由 `Micro/WIN` 从 `LAD/FBD` 图形建立的 `STL` 代码，可在 `LAD`、`FBD` 或 `STL` 视图中启用显示

注释：`Micro/WIN` 保留 `L` 内存的最上方的 4 个字节（`LB60-LB63`），用于调用参数数据。

```

NETWORK 1
// L内存被用于保存布尔输入参数状态，
// 此类参数在LAD和FBD中被显示为功率流输入。
// 从而允许本网络在LAD、FBD和STL编辑器中显示。
LD I0.0
= L60.0
LD I0.1
= L63.7
LD L60.0
CALL SBR_0 L63.7 VB10 I1.0 &VB100 *AC1 VD200

```

用 `STL` 编辑器输入具有与以上显示相同的子例行程序的 `STL` 范例 注释：`STL` 程序员可使用该简化的调用程序。

```

NETWORK 1
// 本网络只能在STL编辑器中显示，
// 因为用作功率流输入的布尔参数
// 未在L内存中保存。
LD I0.0
CALL SBR_0 I0.1 VB10 I1.0 &VB100 *AC1 VD200

```

另请参阅：  
[子例行程序用法详情](#)

## 4.9 IEC FBD指令

### 4.9.1 位逻辑

指令  
位逻辑



注释：  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.9.2 时钟

指令  
时钟



**注释:**

每个SIMATIC指令的红色菱形块♦表示已扩展至标准IEC 1131-3指令集。

### 4.9.3 通讯

白... 指令  
白... 通讯

**注释:**

每个SIMATIC指令的红色菱形块♦表示已扩展至标准IEC 1131-3指令集。

### 4.9.4 比较

白... 指令  
白... 比较

**另请参阅:**

[使用IEC超载和非标准指令](#)

### 4.9.5 转换



获取更多资料 微信搜索 蓝领星球



注释:

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.9.6 计数器



注释:

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.9.7 中断



注释:

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.9.8 逻辑操作

指令  
逻辑操作



另请参阅：  
[使用IEC超载和非标准指令](#)

### 4.9.9 数学

指令  
数学



注释：  
每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

另请参阅：  
[使用IEC超载和非标准指令](#)

### 4.9.10 移动

指令  
移动





**注释:**

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

**另请参阅:**

[使用IEC超载和非标准指令](#)

### 4.9.11 数字

指令

数字



**注释:**

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.9.12 程序控制

指令

程序控制







**注释:**

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

### 4.9.13 移位 / 旋转

指令

移位 / 旋转



**注释:**

每个SIMATIC指令的红色菱形块表示已扩展至标准IEC 1131-3指令集。

**另请参阅:**

[使用IEC超载和非标准指令](#)

### 4.9.14 字符串

指令

字符串





**注释:**  
每个SIMATIC指令的红色菱形块♦表示已扩展至标准IEC 1131-3指令集。

#### 4.9.15 表



**注释:**  
每个SIMATIC指令的红色菱形块♦表示已扩展至标准IEC 1131-3指令集。

#### 4.9.16 计时器





## 4.9.17 子例行程序

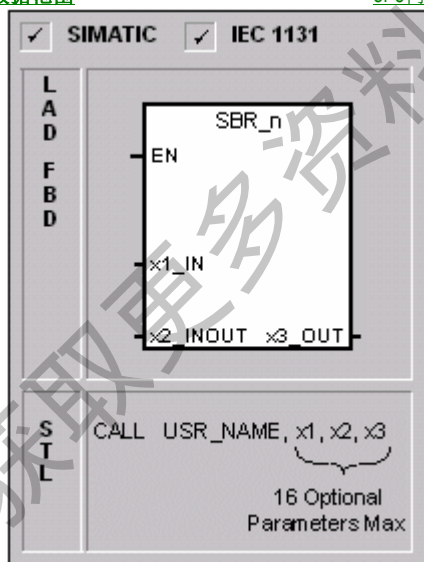
输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
SBR_N 0至63		常量 用于CPU?21、CPU?22、CPU?24 : 用于CPU 224XP和CPU
226		0至127 字[[[ENDKEEPN]]]
IN		V, I, Q, M, SM, S, T, C, L, 功率流 VB, IB, QB, MB, SMB, SB, LB,
AC, *VD, *LD, *AC, 常量 VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC, 常量 VD, ID, QD, MD, SMD, SD,		&SB, &AI, &AQ, &SMB, 常量 *VD, *LD, *AC,
LD, AC, HC, *VD, *LD, *AC, &VB, &IB, &QB, &MB, &T, &C,		布尔字节字、整数双字、双整数 字符串[[[ENDKEEPN]]]
常量		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
IN/OUT		布尔字节字、整数双字、双整数[[[ENDKEEPN]]]
*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD,		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
*AC		布尔字节字、整数双字、双整数[[[ENDKEEPN]]]
OUT		V, I, Q, M, SM, S, T, C, L VB, IB, QB, MB, SMB, SB, LB, AC,
*VD, *LD, *AC VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AQW, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD,		布尔字节字、整数双字、双整数[[[ENDKEEPN]]]
*LD, *AC		

注释: AC地址必须不少于1SM地址必须不少于30[[[ENDKEEPN]]]

[内存范围](#) [错误](#) [S7-200 CPU指令支持](#) [SIMATIC / 国际助记符](#)

[\[\[\[ENDKEEPN\]\]\]](#)

[数据范围](#) [CPU内存中的指令大小](#) [编址内存 \[\[\[ENDKEEPN\]\]\]](#)



**调用子例行程序 (CALL)** 指令将控制转换给子例行程序 (SBR\_n)。您可以使用带参数或不带参数的“调用子例行程序”指令。在子例行程序完成执行后，控制返回至“调用子例行程序”之后的指令。每个子例行程序调用的输入 / 输出参数最大限制为16。如果您尝试下载的程序超过此一限制，会返回一则错误讯息。

如果您为子例行程序指定一个符号名，例如 **USR\_NAME**，该符号名会出现在指令树的“子例行程序”文件夹中。将参数值指定给子例行程序指定一个符号名，例如 **USR\_NAME**，该符号名会出现在指令树的“子例行程序”文件夹中。将参数值指定给子例行程序指定一个符号名，例如 **USR\_NAME**，该符号名会出现在指令树的“子例行程序”文件夹中。

1. 参数值指定给局部内存的顺序由 **CALL** 指定，参数从 **L.0** 开始。2.

一至八个连续位参数值被指定给从 **Lx.0** 开始持续至 **Lx.7** 的单字节。3.

字节、字和双字数值被指定给局部内存，位于字节边界 (**LBx**、**LWx** 或 **LDx**) 位置。在带参数的“调用子例行程序”指令中，

表中定义的变量完全匹配。参数顺序必须以输入参数开始，其次是输入 / 输出参数，然后是输出参数。 **为带参数 =**

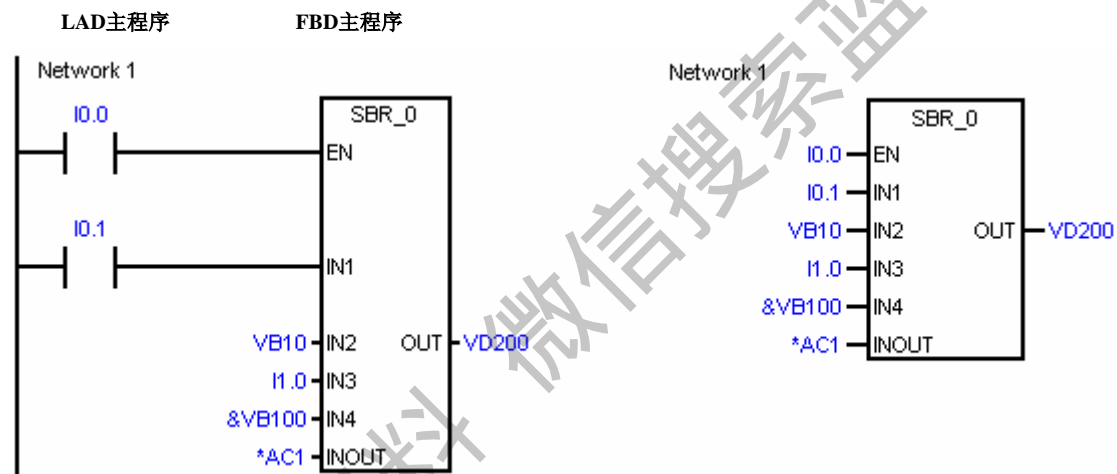
**0调用设置ENO的错误条件：0008 超出最大子例行程序嵌套**

**注释：**位于指令树中的子例行程序名称的工具提示显示每个参数的名称。

#### 子例行程序调用举例

用于SBR\_0的局部变量表

	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
L0.0	IN1	IN	BOOL	
LB1	IN2	IN	BYTE	
L2.0	IN3	IN	BOOL	
LD3	IN4	IN	DWORD	
		IN		
LD7	INOUT	IN_OUT	REAL	
		IN_OUT		
LD11	OUT	OUT	REAL	
		OUT		



由Micro/WIN从LAD/FBD图形建立的STL代码，可在LAD、FBD或STL视图中启用显示

注释：Micro/WIN保留L内存的最上方的4个字节（LB60-LB63），用于调用参数数据。

```

NETWORK 1
// L内存被用于保存布尔输入参数状态，
// 此类参数在LAD和FBD中被显示为功率流输入。
// 从而允许本网络在LAD、FBD和STL编辑器中显示。
LD I0.0
= L60.0
LD I0.1
= L63.7
LD L60.0
CALL SBR_0 L63.7 VB10 I1.0 &VB100 *AC1 VD200

```

用STL编辑器输入具有与以上显示相同的子例行程序的STL范例 注释：STL程式员可使用该简化的调用程序。

```

NETWORK 1
// 本网络只能在STL编辑器中显示，
// 因为用作功率流输入的布尔参数
// 未在L内存中保存。
LD I0.0
CALL SBR_0 I0.1 VB10 I1.0 &VB100 *AC1 VD200

```

另请参阅：  
子例行程序用法详情

获取更多资料 微信搜索蓝领星球

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part



V

获取更多资料 微信搜索 蓝领星球

## 5 SM特殊内存赋值和功能

### 5.1 SMB 0-29（预定义S7-200只读内存）

#### 5.1.1 SMB0系统状态位

特殊内存字节0（SM0.0 - SM0.7）提供八个位，在每次扫描循环结尾处由S7-200 CPU更新。程序可以读取这些位的状态，然后根据位值作出决定。

S7-200 符号名	SM 地址	用户程序读取SMB0状态数据
Always_On	SM0.0	该位总是打开。
First_Scan_On	SM0.1	首次扫描循环时该位打开，一种用途是调用初始化子例行程序。
Retentive_Lost	SM0.2	如果保留性数据丢失，该位为一次扫描循环打开。该位可用作错误内存位或激活特殊启动顺序的机制。
RUN_Power_Up	SM0.3	从电源开启条件进入RUN（运行）模式时，该位为一次扫描循环打开。该位
Clock_60s	SM0.4	该位提供时钟脉冲，该脉冲在1分钟的周期时间内OFF（关闭）30秒，ON（打开）30秒
Clock_1s	SM0.5	该位提供时钟脉冲，该脉冲在1秒钟的周期时间内OFF（关闭）0.5秒，ON（打开）0.5秒。该位提供便于使用的延迟或1秒钟时钟脉冲。
Mode_Switch	SM0.7	该位表示“模式”开关的当前位置（关闭 = “终止”位置，打开 = “运行”位置）。开关位于RUN（运行）位置时，您可以使用该位启用自由端口模式，可使用转换至“终止”位置的方法重新启用带PC / 编程设备的正常通讯。

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

#### 5.1.2 SMB1指令执行状态位

特殊内存字节1（SM1.0 - SM1.7）为各种不同的指令提供执行状态，例如表格和数学运算。这些位在执行时由指令设置和重新设置。程序可以读取位值

S7-200 符号名	SM 地址	用户程序读取SMB1状态数据
Result_0	SM1.0	当操作结果为零时，某些指令的执行打开该位。
Overflow_Illegal	SM1.1	当溢出结果或检测到非法数字数值时，某些指令的执行打开该位。
Neg_Result	SM1.2	数学操作产生负结果时，该位打开。
Divide_By_0	SM1.3	尝试除以零时，该位打开。
Table_Overflow	SM1.4	“增加至表格”指令尝试过度填充表格时，该位打开。
Table_Empty	SM1.5	LIFO或FIFO指令尝试从空表读取时，该位打开。
Not_BCD	SM1.6	尝试将非BCD数值转换为二进制数值时，该位打开。
Not_Hex	SM1.7	当ASCII数值无法转换成有效的十六进制数值时，该位打开。

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

#### 5.1.3 SMB2自由端口接收字符

特殊内存字节2是自由端口接收字符缓冲器。在自由端口模式中接收的每个字符均被置于该位置，易于程序存取。

S7-200 符号名	SM 地址	用户程序读取SMB2，以便载入输入自由端口数据
Receive_Char	SMB2	该字节包含在自由端口通讯过程中从端口0或端口1接收的每个字符。

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

### 5.1.4 SMB3自由端口奇偶校验错误

SMB3用于自由端口模式，包含在接收字符中检测到奇偶错误时设置的奇偶错误位。当检测到奇偶错误时，打开 SM3.0。在程序接受和读取存储在SMB2中的讯息字符数值之前，使用该位测试自由端口讯息字符是否有传输错误。

S7-200 符号名	SM 地址	用户程序读取SMB3，确认输入自由端口数据
Parity_Err	SM3.0	该位表示在端口0和端口1中出现奇偶校验错误。（0 = 无错；1 = 错误）
	SM3.1 ?SM3.7	保留

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

### 5.1.5

### SMB4中断队列溢出、运行时间程序错误、中断启用、自由端口传输器闲置、数值强制

特殊内存字节4（SM4.0 - SM4.7）包含中断队列溢出位和一个显示中断是启用还是禁用的位（SM 4.4）。这些位表示中断发生速率比可处理速率更快，或中断被全局中断禁用指令禁用。其他位表示：

- 运行时间程序错误
- 自由端口传输器状态
- 任何PLC内存数值是否目前被强制

S7-200 符号名	SM 地址	用户程序读取SMB4状态数据
Comm_Int_Ovr	**SM4.0	通讯中断队列溢出时，该位打开。
Input_Int_Ovr	**SM4.1	输入中断队列溢出时，该位打开。
Timed_Int_Ovr	**SM4.2	定时中断队列溢出时，该位打开。
RUN_Err	SM4.3	检测到运行时间编程错误时，该位打开。
Int_Enable	SM4.4	该位反映全局中断启用状态。启用中断时，该位打开。
Xmit0_Idle	SM4.5	传输器闲置（端口0）时，该位打开。
Xmit1_Idle	SM4.6	传输器闲置（端口1）时，该位打开。
Force_On	SM4.7	当任何内存位置被强制时该位打开（仅限22x）。

\*\* 仅限在中断例行程序中使用状态位 4.0、4.1和4.2。队列空置且控制返回主程序时，这些状态位被重设。

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

### 5.1.6 SMB5 I/O错误状态位

特殊内存字节5（SM5.0 - SM5.7）包含表示在I/O系统中检测到的错误条件状态位。这些位为检测到的I/O错误提供概述。

S7-200 符号名	SM 地址	用户程序读取SMB5错误状态数据
IO_Err	SM5.0	如果存在任何I/O错误，该位打开。
Too_Many_D_IO	SM5.1	如果过多数字I/O点与I/O总线连接，该位打开。
Too_Many_A_IO	SM5.2	如果过多模拟I/O点与I/O总线连接，该位打开。
Too_Many_IM	SM5.3	如果过多智能I/O模块与I/O总线连接，该位打开。
	SM5.4	保留
	SM5.5	保留
	SM5.6	保留
DP_Err	SM5.7	如果存在DP标准总线故障，该位打开（仅限S7-215）。

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)



### 5.1.7 SMB6 CPU代码寄存器

特殊内存字节6是CPU标识寄存器。SM6.4 - SM6.7识别PLC的类型。SM6.0 - SM6.3为将来使用保留。

S7-200 符号名称	SM 地址	用户程序读取SMB6 CPU标识符数据								
		MSB				LSB				
CPU_ID	SMB6	7							0	
		x	x	x	x	r	r	r	r	
	SM6.4 - SM6.7	0	0	0	0	= CPU 212 / CPU 222				
		0	0	1	0	= CPU 214 / CPU 224				
		0	1	1	0	= CPU 221				
		1	0	0	0	= CPU 215				
		1	0	0	1	= CPU 216 / CPU 226 / CPU 226XM				
	SM6.0 - SM6.3					r	r	r	r	= 保留

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

### 5.1.8 SMB8-SMB21 I/O模块代码和错误寄存器

SMB8至SMB21以成对字节组织，用于扩充模块0至6。每对偶数字节是模块标识寄存器。这些字节识别模块类型、I/O类型以及输入和输出次数。每对奇数字节是模块错误寄存器。这些字节提供该模块I/O中检测到的任何错误。

	偶数字节 I/O 模块标识寄存器				奇数字节 I/O 模块错误寄存器				
	MSB		LSB		MSB		LSB		
	7								
	M	t	t	A	i	i	Q	Q	
M: 存在模块	0	= 存在			C:	0	= 无错误		
	1	= 不存在				1	= 配置错误		
tt: 模块类型	0	0	= 非智能 I/O 模块		b:	0	= 无错误		
	0	1	= 智能 I/O 模块			1	= 总线故障或奇偶校验错误		
	1	0	= 保留		r:	0	= 超出范围错误		
	1	1	= 保留			1	= 超出范围错误		
A: I/O 类型	0	= 数字			P:	0	= 无错误		
	1	= 模拟				1	= 无用户权利错误		
ii: 输入	0	0	= 无输入		f:	0	= 无错误		
	0	1	= 2 AI 或 8 DI			1	= 熔断丝错误		
	1	0	= 4 AI 或 16 DI		t:	0	= 无错误		
	1	1	= 8 AI 或 32 DI			1	= 接线盒松开错误		
QQ: 输出	0	0	= 无输出						
	0	1	= 2 AQ 或 8 DQ						
	1	0	= 4 AQ 或 16 DQ						
	1	1	= 8 AQ 或 32 DQ						

S7-200 符号名称	SM 地址	用户程序读取SMB8-SMB21 模块标识符和错误数据
EM0_ID	SMB8	模块 0 标识符寄存器
EM0_Err	SMB9	模块 0 错误寄存器
EM1_ID	SMB10	模块 1 标识符寄存器
EM1_Err	SMB11	模块 1 错误寄存器
EM2_ID	SMB12	模块 2 标识符寄存器
EM2_Err	SMB13	模块 2 错误寄存器
EM3_ID	SMB14	模块 3 标识符寄存器
EM3_Err	SMB15	模块 3 错误寄存器
EM4_ID	SMB16	模块 4 标识符寄存器
EM4_Err	SMB17	模块 4 错误寄存器
EM5_ID	SMB18	模块 5 标识符寄存器
EM5_Err	SMB19	模块 5 错误寄存器
EM6_ID	SMB20	模块 6 标识符寄存器
EM6_Err	SMB21	模块 6 错误寄存器

另请参阅：

[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.1.9 SMW22-SMW26扫描时间

SMW22、SMW24和SMW26包含有关扫描时间的信息。您可以毫秒为单位读取最后一次扫描时间、最小扫描时间和最大扫描时间。

S7-200 符号名	SM 地址	用户程序读取SMW22-SMW26扫描时间数据
Last_Scan	SMW22	该字提供最后一次扫描的扫描时间。
Minimum_Scan	SMW24	该字提供自进入RUN（运行）模式以来记录的最小扫描时间。
Maximum_Scan	SMW26	该字提供自进入RUN（运行）模式以来记录的最大扫描时间。

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.1.10 SMB28-SMB29模拟调整

特殊内存字节28和29包含与模拟调整0和1轴角位置对应的数字值。模拟调整电位器位于CPU前方存取门后方。用一把小螺丝刀调整电位器（沿顺时针方向增加，或沿逆时针方向减少）。此类只读数值可被程序用于各种不同的功能，例如，为计时器或计数器更新当前值，输入或改动预设值或设置限制。模拟调整有一个0至255的额定范围，以及72计数的重合性。

S7-200 符号名	SM 地址	用户程序读取SMB28-SMB29，获取电位器位置数据
Pot0_Value	SMB28	该字节存储随模拟调节0输入的数值。
Pot1_Value	SMB29	该字节存储随模拟调节1输入的数值。

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.2 SMB 30-194（预定义S7-200读取 / 写入内存）

### 5.2.1 SMB30和SMB130自由端口控制寄存器

SMB30控制端口0的自由端口通讯；SMB130控制端口1的自由端口通讯。您可以从SMB30和SMB130读取或向SMB30和SMB130写入。这些字节配置各自的通讯端口，进行自由端口操作，并提供自由端口或系统协议支持选择。

S7-200 符号名称	SM 地址		位格式	位格式								
	端口 0	端口 1		MSB				LSB				
P0_Config P1_Config	<b>SMB30</b>	<b>SMB130</b>		7								0
			pp:	p	p	d	b	b	b	m	m	
	SM30.6 – SM30.7	SM130.6 – SM130.7		0	0							= 无奇偶校验
				0	1							= 偶校验
				1	0							= 无奇偶校验
				1	1							= 奇校验
	SM30.0 – SM30.1	SM130.0 – SM130.1	d:									0 = 每个字符 8 个数据位 1 = 每个字符 7 个数据位
	SM30.2 – SM30.4	SM130.2 – SM130.4	bbb:									0 0 0 = 38,400 bps 0 0 1 = 19,200 bps 0 1 0 = 9,600 bps 0 1 1 = 4,800 bps 1 0 0 = 2,400 bps 1 0 1 = 1,200 bps 1 1 0 = 115,200 bps (CPU 1.2 或更高版本) 1 1 1 = 57,600 bps (CPU 1.2 或更高版本)
P0_Config_0 P1_Config_0	SM30.0	SM130.0	mm:									0 0 = 点对点接口协议 (PPI / 从属模式) 0 1 = 自由点协议 1 0 = PPI / 主模式 1 1 = 保留 (PPI / 从属模式默认值)
	SM30.1	SM130.1										<b>注释：</b> 当您选择代码 mm = 10 (PPI 主模式) 时，CPU 在网络上成为主设备，允许执行 NETR 和 NETW 指令。字节 2 至 7 在 PPI 模式中被忽略。

另请参阅：

[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.2.2 SMB31-SMW32永久内存（EEPROM）写入控制

您可以将存储在V内存中的一个数值保存至受程序控制的永久内存（EEPROM）中。欲执行此一功能，将需要保存的位置地址置 SMW32。然后，将命令载入SMB31，保存该数值。一旦您载入保存数值的命令，在CPU重设SM31.7表示保存操作已经完成之前，您不得改变V内存中的数值。

在每次扫描结尾处，CPU检查是否发出将数值保存至永久内存的命令。如果发出命令，指定的数值被保存在永久内存中。

用户程序：写入SMB31和SMB32，设置和控制V内存数据，保存至EEPROM																																						
在开始保存操作之前，设置SM31.7并用0测试写入SM31.7，开始保存操作																																						
S7-200 符号名称	SM 地址	软件命令																																				
Save_to EEPROM	SMB31	<table border="1"> <thead> <tr> <th>MSB</th> <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>LSB</th> </tr> </thead> <tbody> <tr> <td>c</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>s</td> <td>s</td> </tr> </tbody> </table>	MSB	7	6	5	4	3	2	1	0	LSB	c	0	0	0	0	0	0	0	s	s																
MSB	7	6	5	4	3	2	1	0	LSB																													
c	0	0	0	0	0	0	0	s	s																													
Save_to EEPROM_7	SM31.7	c: 0 = 无执行保存操作的请求 1 = 用户程序请求CPU将数据保存至固定内存 在每次保存操作完成后，CPU重设该位。																																				
	SM31.0 - SM31.1	ss: 需要保存的数值大小 <table border="1"> <tbody> <tr> <td>0</td> <td>0</td> <td>= 字节</td> </tr> <tr> <td>0</td> <td>1</td> <td>= 字节</td> </tr> <tr> <td>1</td> <td>0</td> <td>= 字</td> </tr> <tr> <td>1</td> <td>1</td> <td>= 双字</td> </tr> </tbody> </table>	0	0	= 字节	0	1	= 字节	1	0	= 字	1	1	= 双字																								
0	0	= 字节																																				
0	1	= 字节																																				
1	0	= 字																																				
1	1	= 双字																																				
V内存地址																																						
Save_to_Address	SMW32	<table border="1"> <thead> <tr> <th>MSB</th> <th>15</th> <th>14</th> <th>13</th> <th>12</th> <th>11</th> <th>10</th> <th>9</th> <th>8</th> <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>LSB</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> 字节数据：需要保存的数据的V内存地址被存储在SMW32中。该数值作为一个V0的整数偏移量输入。执行保存操作时，存储在该V内存地址的数值被保存至固定内存中的对应V内存位置（EEPROM）。	MSB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB																		
MSB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB																					

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.2.3 SMB34-SMB35定时中断的时间间隔寄存器

特殊内存字节34和35控制中断0和中断1的时间间隔。您可以指定从1毫秒至255毫秒\*\*的时间间隔（以1毫秒为增量）。相应的定时卸鲜录 郊釉溢卸侠 谐缘蜂惺保 CPU捕获时间间隔数值。欲改变时间间隔，您必须将定时中断事件重新附加在相同的或不闹卸侠 谐缘蜂小S梅掷胧录 姆椒口罩苟去澈卸鲜录

S7-200 符号名	SM 地址	以毫秒为单位的定时中断间隔
Time_0_Intrvl	SMB34	定时中断0：时间间隔数值（以1毫秒为增量，从1毫秒至255毫秒**）。
Time_1_Intrvl	SMB35	定时中断1：时间间隔数值（以1毫秒为增量，从1毫秒至255毫秒**）。

\*\*对于21x系列，时间间隔（以1毫秒为增量）从5毫秒至255毫秒。

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.2.4 SMB36-SMB65 HSC0、HSC1和HSC2高速计数寄存器

SMB36至SMB65被用于监控和控制高速计数器HSC0、HSC1和HSC2的操作。

S7-200 符号名称	SM 地址	功能
HSC0_Status	<b>SMB36</b>	<b>HSC0 计数器状态</b> <b>注释：</b> 计数器状态位仅限在执行由高速计数器事件触发的中断例行程序时有效。
	SM36.0 – SM36.4	保留
HSC0_Status_5	SM36.5	HSC0 当前计数方向状态位：1 = 向上计数
HSC0_Status_6	SM36.6	HSC0 当前值等于预设值状态位：1 = 等于
HSC0_Status_7	SM36.7	HSC0 当前值大于预设值状态位：1 = 大于
HSC0_Ctrl	<b>SMB37</b>	<b>HSC0 控制</b> <b>注释：</b> 当您使用 HSC 外部重设中断事件时，请勿尝试载入新当前值或禁用，然后从附加在该事件上的中断例行程序内重新启用高速计数器。这会导致严重错误状况。
HSC0_Reset_Level	SM37.0	重设的现用水平控制位：0 = 重设现用水平高；1 = 重设现用水平低
	SM37.1	保留
HSC0_Rate	SM37.2	累积计数器的 HSC0 计数速率选择：0 = 4x 计数速率；1 = 1x 计数速率
HSC0_Dir	SM37.3	HSC0 方向控制位：1 = 向上计数
HSC0_Dir_Update	SM37.4	HSC0 更新方向：1 = 更新方向
HSC0_PV_Update	SM37.5	HSC0 更新预设值：1 = 在 HSC0 预设中写入新预设值
HSC0_CV_Update	SM37.6	HSC0 更新当前值：1 = 在 HSC0 当前值中写入新当前值
HSC0_Enable	SM37.7	HSC0 启用位：1 = 启用
HSC0_CV	<b>SMD38</b>	<b>HSC0 新当前值</b> 双数值：SMD38 被用于将 HSC0 当前值设为您选择的任何数值。带有您希望使用的新当前值的 SMD38 则将 SM37.6 写入 1。新当前值被写入 HSC0 当前值。
HSC0_PV	<b>SMD42</b>	<b>HSC0 新预设值</b> 双数值：SMD42 被用于将 HSC0 预设值设为您选择的任何数值。带有您希望使用的新预设值的 SMD42 则将 SM37.5 写入 1。新预设值被写入 HSC0 预设值。
HSC1_Status	<b>SMB46</b>	<b>HSC1 计数器状态</b> <b>注释：</b> 计数器状态位仅限在执行由高速计数器事件触发的中断例行程序时有效。
	SM46.0 – SM46.4	保留
HSC1_Status_5	SM46.5	HSC1 当前计数方向状态位：1 = 向上计数
HSC1_Status_6	SM46.6	HSC1 当前值等于预设值状态位：1 = 等于
HSC1_Status_7	SM46.7	HSC1 当前值大于预设值状态位：1 = 大于

HSC1_Ctrl	<b>SMB47</b>	<b>HSC1 控制</b> <b>注释:</b> 当您使用 HSC 外部重设中断事件时, 请勿尝试载入新当前值或禁用, 然后从附加在该事件上的中断例行程序内重新启用高速计数器。这会导致严重错误状况。
HSC1_Reset_Level	SM47.0	重设的 HSC1 现用水平控制位: 0 = 重设现用水平高; 1 = 重设现用水平低
HSC1_Start_Level	SM47.1	启用的 HSC1 现用水平控制位: 0 = 启动现用水平高; 1 = 启用现用水平低
HSC1_Rate	SM47.2	求积计数器的 HSC1 计数速率选择: 0 = 4x 计数速率; 1 = 1x 计数速率
HSC1_Dir	SM47.3	HSC1 方向控制位: 1 = 向上计数
HSC1_Dir_Update	SM47.4	HSC1 更新方向: 1 = 更新方向
HSC1_PV_Update	SM47.5	HSC1 更新预设值: 1 = 在 HSC1 预设中写入新预设值
HSC1_CV_Update	SM47.6	HSC1 更新当前值: 1 = 在 HSC1 当前值中写入新当前值
HSC1_Enable	SM47.7	HSC1 启用位: 1 = 启用
HSC1_CV	<b>SMD48</b>	<b>HSC1 新当前值</b> 双字数值: SMD48 被用于将 HSC1 当前值设为您选择的任何数值。带有您希望使用的新当前值的 SMD48 则将 SM47.6 写入 1。新当前值被写入 HSC1 当前值。
HSC1_PV	<b>SMD52</b>	<b>HSC1 新预设值</b> 双字数值: SMD52 被用于将 HSC1 预设值设为您选择的任何数值。带有您希望使用的新预设值的 SMD52 则将 SM47.5 写入 1。新预设值被写入 HSC1 预设值。
HSC2_Status	<b>SMB56</b>	<b>HSC2 计数器状态</b> <b>注释:</b> 计数器状态位仅限在执行由高速计数器事件触发的中断例行程序时有效。
	SM56.0 – SM56.4	保留
HSC2_Status_5	SM56.5	HSC2 当前计数方向状态位: 1 = 向上计数
HSC2_Status_6	SM56.6	HSC2 当前值等于预设值状态位: 1 = 等于
HSC2_Status_7	SM56.7	HSC2 当前值大于预设值状态位: 1 = 大于
HSC2_Ctrl	<b>SMB57</b>	<b>HSC2 控制</b> <b>注释:</b> 当您使用 HSC 外部重设中断事件时, 请勿尝试载入新当前值或禁用, 然后从附加在该事件上的中断例行程序内重新启用高速计数器。这会导致严重错误状况。
HSC2_Reset_Level	SM57.0	重设的 HSC2 现用水平控制位: 0 = 重设现用水平高; 1 = 重设现用水平低
HSC2_Start_Level	SM57.1	启用的 HSC2 现用水平控制位: 0 = 启动现用水平高; 1 = 启用现用水平低
HSC2_Rate	SM57.2	求积计数器的 HSC2 计数速率选择: 0 = 4x 计数速率; 1 = 1x 计数速率
HSC2_Dir	SM57.3	HSC2 方向控制位: 1 = 向上计数
HSC2_Dir_Update	SM57.4	HSC2 更新方向: 1 = 更新方向
HSC2_PV_Update	SM57.5	HSC2 更新预设值: 1 = 在 HSC2 预设中写入新预设值
HSC2_CV_Update	SM57.6	HSC2 更新当前值: 1 = 在 HSC2 当前值中写入新当前值
HSC2_Enable	SM57.7	HSC2 启用位: 1 = 启用
HSC2_CV	<b>SMD58</b>	<b>HSC2 新当前值</b> 双字数值: SMD58 被用于将 HSC2 当前值设为您选择的任何数值。带有您希望使用的新当前值的 SMD58 则将 SM57.6 写入 1。新当前值被写入 HSC2 当前值。
HSC2_PV	<b>SMD62</b>	<b>HSC2 新预设值</b> 双字数值: SMD62 被用于将 HSC2 预设值设为您选择的任何数值。带有您希望使用的新预设值的 SMD62 则将 SM57.5 写入 1。新预设值被写入 HSC2 预设值。

另请参阅:  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.2.5 SMB66-SMB85 PTO / PTM高速输出寄存器

SMB66至SMB85被用于监控和控制PLC（脉冲）指令的脉冲链输出和脉冲宽度调制功能。

S7-200 符号名称	SM 地址	功能
PTO0_Status	<b>SMB66</b>	<b>PTO0 状态</b>
	SM66.0 – SM66.3	保留
PLS0_Err_Abort	SM66.4	PTO0 概况异常中止：0 = 无错，1 = 由于 $\delta$ 计算错误异常中止
PLS0_Cmd_Abort	SM66.5	PTO0 概况异常中止：0 = 未被用户命令异常中止；1 = 被用户命令异常中止
PLS0_Ovr	SM66.6	PTO0 管道溢出使用外部管道时由系统清除，否则必须由用户重设：0 = 无溢出；1 = 管道溢出
PLS0_Idle	SM66.7	PTO0 空闲位：0 = PTO 正在执行，1 = PTO 空闲
PLS0_Ctrl	<b>SMB67</b>	<b>Q0.0 的控制脉冲链式输出和脉冲宽度调制</b>
PLS0_Cycle_Update	SM67.0	PTO0/PWM0 更新周期时间值：1 = 写入新周期时间
PWM0_PW_Update	SM67.1	PWM0 更新脉冲宽度值：1 = 写入新脉冲宽度
PTO0_PC_Update	SM67.2	PTO0 更新脉冲计数值：1 = 写入新脉冲计数
PLS0_TimeBase	SM67.3	PTO0/PWM0 时基：0 = 1 $\mu$ s/滴答，1 = 1 ms/滴答
PWM0_Sync	SM67.4	同步更新 PWM0：0 = 异步更新，1 = 同步更新
PTO0_Op	SM67.5	PTO0 操作：0 = 单段操作（存储在 SM 内存中的周期时间和脉冲计数）；1 = 多段操作（存储在 V 内存中的概况表）。
PLS0_Select	SM67.6	PTO0/PWM0 模式选择：0 = PTO；1 = PWM
PLS0_Enable	SM67.7	PTO0/PWM0 启用位：1 = 启用
PLS0_Cycle	<b>SMW68</b>	<b>周期时间值、脉冲链式或脉冲宽度调制输出 0</b> 字节数据：PTO0/PWM0 周期时间值（2 至 65535 个时基单位）
PWM0_PW	<b>SMW70</b>	<b>脉冲宽度调制输出 0 的脉冲宽度值</b> 字节数据：PWM0 脉冲宽度值（0 至 65535 个时基单位）
PTO0_PC	<b>SMD72</b>	<b>脉冲链式输出 0 的脉冲计数值</b> 双字节数据：PTO0 脉冲计数值（1 至 $2^{32}-1$ ）
PTO1_Status	<b>SMB76</b>	<b>PTO1 状态</b>
	SM76.0 – SM76.3	保留
PLS1_Err_Abort	SM76.4	PTO1 概况异常中止：0 = 无错，1 = 由于 $\delta$ 计算错误异常中止
PLS1_Cmd_Abort	SM76.5	PTO1 概况异常中止：0 = 未被用户命令异常中止；1 = 被用户命令异常中止
PLS1_Ovr	SM76.6	PTO1 管道溢出使用外部管道时由系统清除，否则必须由用户重设：0 = 无溢出；1 = 管道溢出
PLS1_Idle	SM76.7	PTO1 空闲位：0 = PTO 正在执行，1 = PTO 空闲
PLS1_Ctrl	<b>SMB77</b>	<b>Q0.1 的控制脉冲链式输出和脉冲宽度调制</b>
PLS1_Cycle_Update	SM77.0	PTO1/PWM1 更新周期时间值：1 = 写入新周期时间
PWM1_PW_Update	SM77.1	PWM1 更新脉冲宽度值：1 = 写入新脉冲宽度
PTO1_PC_Update	SM77.2	PTO1 更新脉冲计数值：1 = 写入新脉冲计数
PLS1_TimeBase	SM77.3	PTO1/PWM1 时基：0 = 1 $\mu$ s/滴答，1 = 1 ms/滴答
PWM1_Sync	SM77.4	同步更新 PWM1：0 = 异步更新，1 = 同步更新
PTO1_Op	SM77.5	PTO1 操作：0 = 单段操作（存储在 SM 内存中的周期时间和脉冲计数）；1 = 多段操作（存储在 V 内存中的概况表）。
PLS1_Select	SM77.6	PTO1/PWM1 模式选择：0 = PTO；1 = PWM
PLS1_Enable	SM77.7	PTO1/PWM1 启用位：1 = 启用
PLS1_Cycle	<b>SMW78</b>	<b>周期时间值、脉冲链式或脉冲宽度调制输出 1</b> 字节数据：PTO1/PWM1 周期时间值（2 至 65535 个时基单位）
PWM1_PW	<b>SMW80</b>	<b>脉冲宽度调制输出 1 的脉冲宽度值</b> 字节数据：PWM1 脉冲宽度值（0 至 65535 个时基单位）
PTO1_PC	<b>SMD82</b>	<b>脉冲链式输出 1 的脉冲计数值</b> 双字节数据：PTO1 脉冲计数值（1 至 $2^{32}-1$ ）

另请参阅：  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.2.6 SMB86-SMB94和SMB186-SMB194接收讯息控制

SMB86至SMB94以及SMB186至SMB194被用于控制和读取有关“接收讯息”指令的状态。

S7-200 符号名称	SM 地址		位格式	接收讯息状态字节								
	端口 0	端口 1		MSB				LSB				
P0_Stat_Rcv	SMB86			7								0
P1_Stat_Rcv		SMB186		n	r	e	0	0	t	c	P	
P0_Stat_Rcv_7	SM86.7		n:	1	= 接收讯息被用户禁用命令终止							
P1_Stat_Rcv_7		SM186.7										
P0_Stat_Rcv_6	SM86.6		r:	1	= 接收讯息终止: 输入参数错误或缺少起始或结束条件							
P1_Stat_Rcv_6		SM186.6										
P0_Stat_Rcv_5	SM86.5		e:	1	= 收到结束符							
P1_Stat_Rcv_5		SM186.5										
P0_Stat_Rcv_2	SM86.2		t:						1	= 接收讯息终止: 计时器失效		
P1_Stat_Rcv_2		SM186.2										
P0_Stat_Rcv_1	SM86.1		c:							1	= 接收讯息终止: 收到最大字符计数	
P1_Stat_Rcv_1		SM186.1										
P0_Stat_Rcv_0	SM86.0		p:								1	= 由于奇偶校验错误接收讯息终止
P1_Stat_Rcv_0		SM186.0										

	SM 地址		位格式	接收讯息控制字节								
	端口 0	端口 1		MSB				LSB				
P0_Ctrl_Rcv	SMB87			7								0
P1_Ctrl_Rcv		SMB187		en	sc	ec	il	c/m	Tmr	bk		0
P0_Ctrl_Rcv_7	SM87.7		en:	0	= 接收讯息功能被禁用							
P1_Ctrl_Rcv_7		SM187.7		1	= 接收讯息功能被启用							
P0_Ctrl_Rcv_6	SM87.6		sc:	0	= 忽略 SMB88 或 SMB188							
P1_Ctrl_Rcv_6		SM187.6		1	= 使用 SMB88 或 SMB188 数值检测讯息开始							
P0_Ctrl_Rcv_5	SM87.5		ec:	0	= 忽略 SMB89 或 SMB189							
P1_Ctrl_Rcv_5		SM187.5		1	= 使用 SMB89 或 SMB189 数值检测讯息结束							
P0_Ctrl_Rcv_4	SM87.4		il:	0	= 忽略 SMMV90 或 SMB190							
P1_Ctrl_Rcv_4		SM187.4		1	= 使用 SMMV90 数值检测空行条件							
P0_Ctrl_Rcv_3	SM87.3		c/m:	0	= 计时器是字符间计时器							
P1_Ctrl_Rcv_3		SM187.3		1	= 计时器是讯息计时器							
P0_Ctrl_Rcv_2	SM87.2		tmr:					0	= 忽略 SMMV92 或 SMMV192			
P1_Ctrl_Rcv_2		SM187.2						1	= 如果超出 SMMV92 或 SMMV192 中的时间阶段, 则终止接收			
P0_Ctrl_Rcv_1	SM87.1		bk:						0	= 忽略断开条件		
P1_Ctrl_Rcv_1		SM187.1							1	= 将断开条件用作讯息检测开始		

讯息中断控制字节的位被用于定义用于识别讯息的标准。定义讯息开始和结束标准。欲确定讯息开始, 两套被执行逻辑 AND (与) 的讯息标准开始必须真实, 且必须连续出现 (连续表示闲置行后面是起始字符或分隔符后面是起始字符)。欲确定讯息结束, 被启用的讯息标准结束被执行逻辑 OR (或)。以下是起始和停止标准公式:

讯息开始 =  $il * sc + bk * sc$

讯息结束 =  $ec + tmr +$  达到的最大字符计数

S7-200 符号名称	SM 地址		位格式	为訊息标准开始编程								
	端口 0	端口 1		MSB				LSB				
P0_Ctrl_Rcv P1_Ctrl_Rcv	SMB87	SMB187		en	sc	ec	il	c/m	Tim r	bk	0	
			空闲行检测	0			1			0		SMW90 > 0
			起始字符检测	1			0			0		SMW90 = 无关
			断开检测	0			0			1		SMW90 = 无关
			对请求的任何应答	0			1			0		SMW90 = 0 (如果无应答, 讯息计时器可用于终止接收)
			断开和起始字符	1			0			1		SMW90 = 无关
			空闲行和起始字符	1			1			0		SMW90 > 0
			空闲行和起始字符 (非法)	1			1			0		SMW90 = 0

**注释:** 接收会由超时运行或奇偶校验错误 (如启用) 自动终止

S7-200 符号名称	SM 地址		描述
	端口 0	端口 1	
P0_Start_Char P1_Start_Char	SMB88	SMB188	讯息字符开始。
P0_End_Char P1_End_Char	SMB89	SMB189	讯息字符结束。
P0_Idle_Time P1_Idle_Time	SMW90	SMW190	字数据: 以毫秒为单位给出的空闲行时段。空闲行时间失败后收到的第一个字符是新讯息的开始。
P0_Timeout P1_Timeout	SMW92	SMW192	字数据: 以毫秒为单位给出的字符间/讯息间计时器超时数值。如果超过时段, 接收讯息被终止。
P0_Max_Char P1_Max_Char	SMB94	SMB194	最大接收字符数 (1 至 255 个字节)。 <b>注释:</b> 该范围必须设为预期的最大缓冲器大小, 即使字符计数讯息终止未使用亦如此。

另请参阅:  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.2.7 SMW98 I/O扩充总线 - 通讯错误

SMW98提供有关扩充I/O总线中的错误数目的信息。

S7-200 符号名

EM\_Parity\_Err  
及用户写入零时被清除。

SM Byte (Read/Write)

SMW98

I/O扩充总线? 错误总结

每当扩充I/O总线中检测到奇偶校验错误时, 该字均被递增。电源开启时以

另请参阅:  
[所有特殊内存赋值列表](#)  
[系统符号表](#)



## 5.2.8 SMB130-SMB165 HSC3、HSC4和HSC5高速计数寄存器

SMB136至SMB165被用于监控和控制高速计数器HSC3、HSC4和HSC5操作。

S7-200 符号名称	SM 地址	说明
HSC3_Status	<b>SMB136</b>	<b>HSC3 计数器状态</b> <b>注释：</b> 计数器状态位仅限在执行由高速计数器事件触发的中断例行程序时有效。
	SM136.0 – SM136.4	保留
HSC3_Status_5	SM136.5	HSC3 当前计数方向状态位：1 = 向上计数
HSC3_Status_6	SM136.6	HSC3 当前值等于预设值状态位：1 = 等于
HSC3_Status_7	SM136.7	HSC3 当前值大于预设值状态位：1 = 大于
HSC3_Ctrl	<b>SMB137</b>	<b>HSC3 控制</b>
	SM137.0 – SM137.2	保留
HSC3_Dir	SM137.3	HSC3 方向控制位：1 = 向上计数
HSC3_Dir_Update	SM137.4	HSC3 更新方向：1 = 更新方向
HSC3_PV_Update	SM137.5	HSC3 更新预设值：1 = 在 HSC3 预设中写入新预设值
HSC3_CV_Update	SM137.6	HSC3 更新当前值：1 = 在 HSC3 当前值中写入新当前值
HSC3_Enable	SM137.7	HSC3 启用位：1 = 启用
HSC3_CV	<b>SMD138</b>	<b>HSC3 新当前值</b> 双数值：SMD138 被用于将 HSC3 当前值设为您选择的任何数值。带有您希望使用的新当前值的 SMD138 则将 SM137.6 写入 1。新当前值被写入 HSC3 当前值。
HSC3_PV	<b>SMD142</b>	<b>HSC3 新预设值</b> 双数值：SMD142 被用于将 HSC3 预设值设为您选择的任何数值。带有您希望使用的新预设值的 SMD142 则将 SM137.5 写入 1。新预设值被写入 HSC3 预设值。
HSC4_Status	<b>SMB146</b>	<b>HSC4 计数器状态</b> <b>注释：</b> 计数器状态位仅限在执行由高速计数器事件触发的中断例行程序时有效。
	SM146.0 – SM146.4	保留
HSC4_Status_5	SM146.5	HSC4 当前计数方向状态位：1 = 向上计数
HSC4_Status_6	SM146.6	HSC4 当前值等于预设值状态位：1 = 等于
HSC4_Status_7	SM146.7	HSC4 当前值大于预设值状态位：1 = 大于

获取更多资料

HSC4_Ctrl	<b>SMB147</b>	<b>HSC4 控制</b> <b>注释:</b> 当您使用 HSC 外部重设中断事件时, 请勿尝试写入新当前值或禁用, 然后从附加在该事件上的中断例行程序内重新启用高速计数器。这会导致严重错误状况。 重设的 HSC4 现用水平控制位: 0 = 重设现用水平高; 1 = 重设现用水平低
HSC4_Reset_Level	SM147.0 SM147.1	保留
HSC4_Rate	SM147.2	累积计数器的 HSC4 计数速率选择: 0 = 4x 计数速率; 1 = 1x 计数速率
HSC4_Dir	SM147.3	HSC4 方向控制位: 1 = 向上计数
HSC4_Dir_Update	SM147.4	HSC4 更新方向: 1 = 更新方向
HSC4_PV_Update	SM147.5	HSC4 更新预设值: 1 = 在 HSC4 预设中写入新预设值
HSC4_CV_Update	SM147.6	HSC4 更新当前值: 1 = 在 HSC4 当前值中写入新当前值
HSC4_Enable	SM147.7	HSC4 启用位: 1 = 启用
HSC4_CV	<b>SMD148</b>	<b>HSC4 新当前值</b> 双数值: SMD148 被用于将 HSC4 当前值设为您选择的任何数值。带有您希望使用的新当前值的 SMD148 则将 SM147.6 写入 1。新当前值被写入 HSC4 当前值。
HSC4_PV	<b>SMD152</b>	<b>HSC4 新预设值</b> 双数值: SMD152 被用于将 HSC4 预设值设为您选择的任何数值。带有您希望使用的新预设值的 SMD152 则将 SM147.5 写入 1。新预设值被写入 HSC4 预设值。
HSC5_Status	<b>SMB156</b>	<b>HSC5 计数器状态</b> <b>注释:</b> 计数器状态位仅限在执行由高速计数器事件触发的中断例行程序时有效。
	SM156.0 – SM156.4	保留
HSC5_Status_5	SM156.5	HSC5 当前计数方向状态位: 1 = 向上计数
HSC5_Status_6	SM156.6	HSC5 当前值等于预设值状态位: 1 = 等于
HSC5_Status_7	SM156.7	HSC5 当前值大于预设值状态位: 1 = 大于
HSC5_Ctrl	<b>SMB157</b>	<b>HSC5 控制</b>
	SM156.0 – SM156.2	保留
HSC5_Dir	SM157.3	HSC5 方向控制位: 1 = 向上计数
HSC5_Dir_Update	SM157.4	HSC5 更新方向: 1 = 更新方向
HSC5_PV_Update	SM157.5	HSC5 更新预设值: 1 = 在 HSC5 预设中写入新预设值
HSC5_CV_Update	SM157.6	HSC5 更新当前值: 1 = 在 HSC5 当前值中写入新当前值
HSC5_Enable	SM157.7	HSC5 启用位: 1 = 启用
HSC5_CV	<b>SMD158</b>	<b>HSC5 新当前值</b> 双数值: SMD158 被用于将 HSC5 当前值设为您选择的任何数值。带有您希望使用的新当前值的 SMD158 则将 SM157.6 写入 1。新当前值被写入 HSC5 当前值。
HSC5_PV	<b>SMD162</b>	<b>HSC5 新预设值</b> 双数值: SMD162 被用于将 HSC5 预设值设为您选择的任何数值。带有您希望使用的新预设值的 SMD162 则将 SM157.5 写入 1。新预设值被写入 HSC5 预设值。

另请参阅:  
[所有特殊内存赋值列表](#)  
[系统符号表](#)

## 5.2.9 用于 PLC (脉冲) 指令的 SMB166-SMB194 PTO 概要表

SMB166 至 SMB194 被用于显示现有轮廓步骤数目和 V 内存中的轮廓表地址。

S7-200 符号名	SM 地址	脉冲链输出配置
PTO0_Step	SMB166	PTO0 现有轮廓步骤的当前条目数
	SMB167	保留
PTO0_Profile	SMW168	字数据类型: PTO0 轮廓表 V 内存地址, 作为 V0 的偏移量给出
	SMW170	线性轮廓状态字节
	SMW171	线性轮廓结果寄存器
	SMW172	手动模式频率寄存器
	SMB173 ?SMB175	保留
PTO1_Step	SMB176	PTO1 现有轮廓步骤的当前条目数
	SMB177	保留
PTO1_Profile	SMW178	字数据类型: PTO1 轮廓表 V 内存地址, 作为 V0 的偏移量给出
	SMW180	线性轮廓状态字节

SMW181	线性轮廓结果寄存器
SMW182	手动模式频率寄存器

另请参阅:

[所有特殊内存赋值列表](#)

[系统符号表](#)

## 5.2.10 为智能扩充模块提供的状态信息保留的SMB200-SMB549

如下表所示，SMB200至SMB299为智能扩充模块提供的状态信息所保留，例如EM 277 PROFIBUS-DP模块。请参阅您的扩充模块文档中有关模块如何使用SMB200至SMB549的信息。

**注释:**

为智能模块指定SM位置的方法在2.2版和更高的版本中已经改变。

如果您使用的是2.2版以前的CPU版本，您应当将所有的智能模块放入与CPU临近的槽中，且在所有非智能模块之前，以确保兼容性。

### 特殊内存字节SMB200至SMB549

0号槽中的 智能模块	1号槽中的 智能模块	2号槽中的 智能模块	3号槽中的 智能模块	4号槽中的 智能模块	5号槽中的 智能模块	6号槽中的 智能模块	说明
SMB200至 SMB215	SMB250至 SMB265	SMB300至 SMB315	SMB350至 SMB365	SMB400至 SMB415	SMB450至 SMB465	SMB500至 SMB515	模块名称 (16个ASCII字符)
SMB216至 SMB219	SMB266至 SMB269	SMB316至 SMB319	SMB366至 SMB369	SMB416至 SMB419	SMB466至 SMB469	SMB516至 SMB519	
SMB519	S/W修订号码 (4个ASCII字符) xxxx						
SMW220	SMW270	SMW320	SMW370	SMW420	SMW470	SMW520	SMW
SMW220	错误代码						
SMB222至 SMB249	SMB272至 SMB299	SMB322至 SMB349	SMB372至 SMB399	SMB422至 SMB449	SMB472至 SMB499	SMB522至 SMB549	具体模块类型信息

另请参阅:

[所有特殊内存赋值列表](#)

[系统符号表](#)

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part



# VI

获取更多资料 微信搜索 蓝领星球

## 6 PLC操作和选项

### 6.1 停止和运行PLC操作模式

使用下列一种方法改变PLC操作模式：

- 单击“运行”按钮，进入RUN（运行）模式，或单击“停止”按钮，进入STOP（停止）模式。
- 选择PLC (PLC) > 运行 (RUN) 菜单命令，进入RUN（运行）模式，或选择PLC > 停止 (PLC > STOP) 菜单命令，进入STOP（停止）模式。
- 以手动方式更改位于PLC中的模式开关。
- 在程序中插入STOP（停止）指令。

#### 注释：

欲使用STEP 7-Micro/WIN软件控制RUN/STOP（运行 / 停止）模式，在STEP 7-Micro/WIN和PLC之间必须存在通讯路径。另外，PLC硬件模式开关必须设为TERM（终端）或RUN（运行）。将模式开关设为TERM（终端）并不改变PLC的操作模式，但允许STEP 7-Micro/WIN更改PLC操作模式。PLC前方的状态LED表示当前操作模式。当程序状态或状态图操作在进行时，在STEP 7-Micro/WIN窗口右下角附近的状态条上有一个RUN/STOP（运行 / 停止）指示灯。

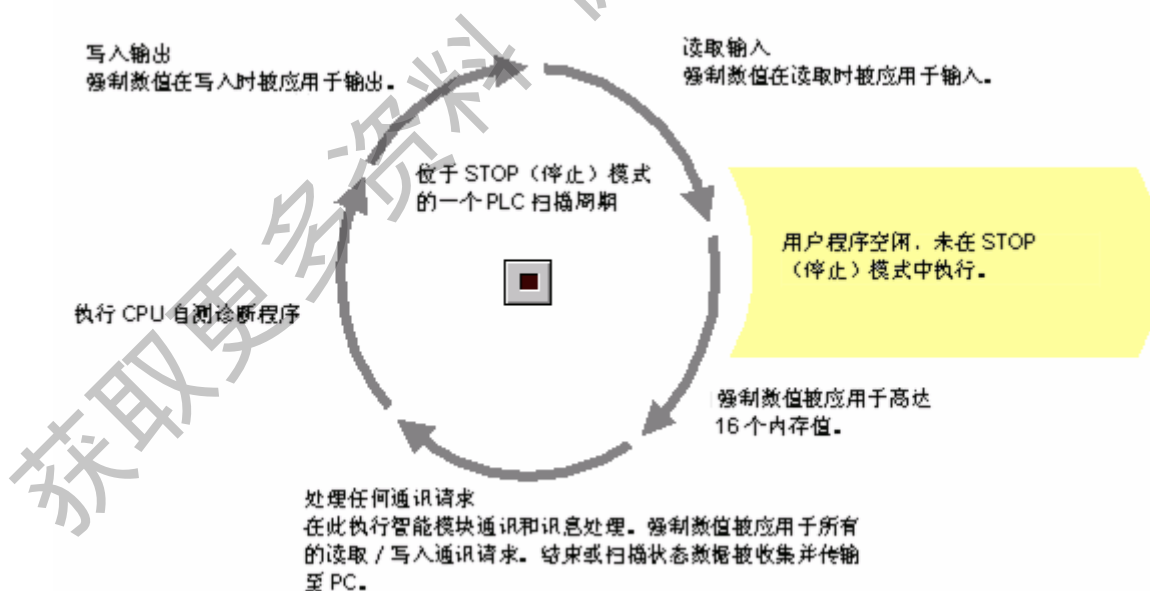
#### PLC操作模式详情：

PLC有两种操作模式：STOP（停止）和RUN（运行）模式。在STOP（停止）模式中，您可以建立 / 编辑程序。在STOP（停止）模式中不得执行程序。但在RUN（运行）模式中可执行程序。此外，在RUN（运行）模式中，您可以建立、编辑和监控程序操作和数据。提供的调试协助可增强追踪程序操作和识别编程问题的能力。

可在STOP（停止）模式中使用调试协助，例如首次扫描和多次扫描功能，并导致预定扫描次数从STOP（停止）更改为RUN（运行）模式。

PLC操作系统存储严重错误，并强制从RUN（运行）模式更改为STOP（停止）模式。如果PLC检测到严重错误，在严重错误条件依然存在时不允许从STOP（停止）模式更改为RUN（运行）模式。PLC操作系统功能存储非严重错误，用于检查，但不会使模式从RUN（运行）更改为STOP（停止）。

在STOP（停止）模式中，PLC处于半空闲状态。用户程序执行被中断；执行输入更新；用户中断条件被禁用。下图描述PLC在STOP（停止）模式中遵守的时间表。

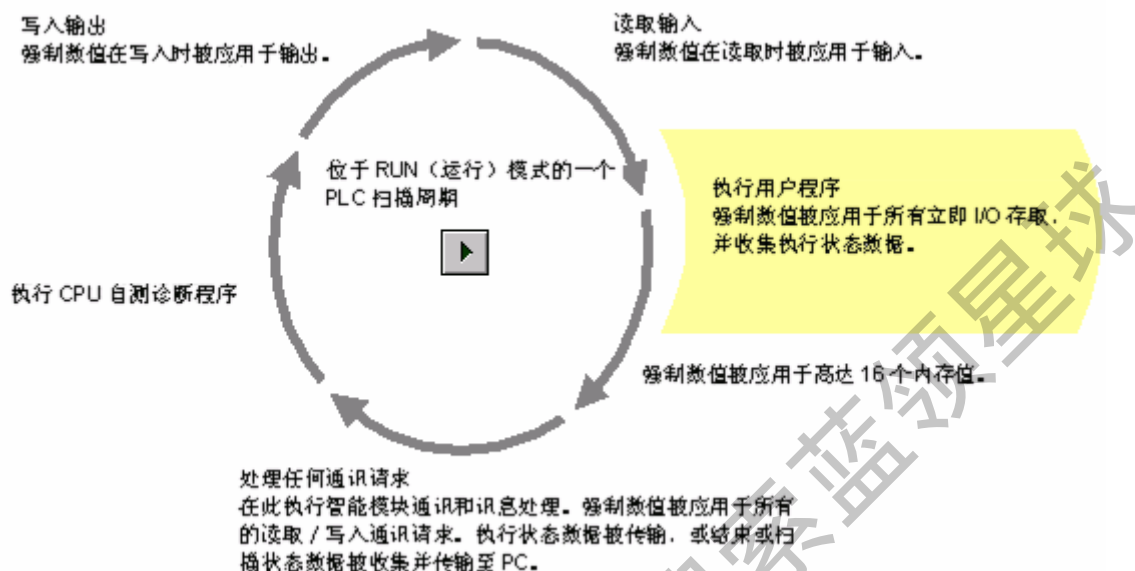


发生通讯中断时，PLC收到讯息并在适当的情况下执行请求。当PLC仍然处于STOP（停止）模式时，图像寄存器作出I/O数值更改。一种例外的情况是对图像寄存器盖写I/O数值改动的强制功能。处于STOP（停止）模式时，您能够载入、上载或删除用户程序

一台或多台设备尝试通过通讯端口与PLC通讯时，PLC按顺序对每一项请求作出应答。PLC不尝试阻止一台通讯设备的行动干扰

自我诊断检查包括定期检查操作系统EEPROM、I/O模块状态检查和在每次存取扩充I/O时执行的I/O扩充总线一致性检查。

在RUN（运行）模式中，PLC读取输入、执行程序、写入输出、对通讯请求作出应答、更新智能模块、执行内部管理工作以及对中断条件作出应答。PLC不支持用于RUN（运行）模式执行循环的固定扫描时间。根据发生的优先顺序对这些行动（中断除外）



每次扫描循环开始时先读取当前输入位数值，并将这些数值写入输入图象寄存器。输入位无相应的实际输入，但与具有实际输入位位于同一个字节中，每次输入更新循环时，除非输入位被强制，否则会在图象寄存器中被设为零。

读取输入后，程序从第一条指令开始执行，直至遇到结束指令。遇到结束指令时，PLC检查系统的智能模块是否需要服务。如果秤 寥⊙断 2.13. 断 4.媧Ⓚ浴 返南乱唤锥位撼迤髦小

在扫描循环的讯息处理阶段，从通讯端口接收的讯息被处理。完成的应答被置于一旁，等待在适当的时间传输给给通讯请求方。

自我诊断检查包括对操作系统EEPROM和用户程序内存以及I/O模块状态检查进行定期检查。

最后，输出图象寄存器数值被写入输出模块，从而完成一次扫描循环。

另请参阅：

[在停止模式中写入和强制输出](#)

[在运行模式中进行程序编辑](#)

[应用程序用户参考手册](#)

[入门指南目录](#)

## 6.2 在停止模式写入和强制输出

欲在STOP（停止）模式中启用“写入”和“强制”输出，选择 **调试（Debug） > 在停止模式中写入和强制输出（Write-Force Outputs in STOP）** 菜单命令。

S7-200 PLCs支持当PLC处于STOP（停止）模式时写入和强制输出（模拟和数字）。出于安全考虑，您必须明确要求在STEP 7-Micro/WIN中启用该功能。菜单选项 **调试（Debug） > 在停止模式中写入-强制输出（Write-Force Outputs in STOP）** 使您能够在PLC位于STOP（停止）模式时写入或强制输出。



**警告** 在写入或强制输出时，如果S7-200 PLC与其他设备相连，这些改动可能被传输至该装置，可能引起该装置操作无法预料，亦可能造成人员死亡和严重伤害及/或装置损坏。

每次打开STEP 7-

Micro/WIN时，菜单选项设为默认值未标选，防止您在PLC处于STOP（停止）模式时写入或强制输出。标选菜单选项会为当前项目的当前编辑操作启用写入和强制功能。打开不同的项目时，菜单选项会返回默认状态，禁止用户在PLC处于STOP（停止）模式时写入或强制输出地址。

另请参阅：

[PLC运行/停止模式](#)

[在运行模式中进行程序编辑](#)

[应用程序用户参考手册](#)

[入门指南目录](#)

## 6.3 控制程序元素

### 阶梯程序

在LAD程序中，逻辑的基本元素用接点、线圈和方框表示。构成完整电路的一套互联元素被称为网络。

硬连线输入由被称为接点的符号表示。正常状态下打开的接点在关闭时启用功率流。接点亦可在正常状态下关闭。在这种情况下，打开接点时出现功率流。

硬连线输出由被称为线圈的符号表示。线圈有功率流时，输出被打开。

方框是代表在PLC中执行的复杂操作的符号。方框可简化操作编程。例如，[计时器](#)和[计数器](#)用方框表示。

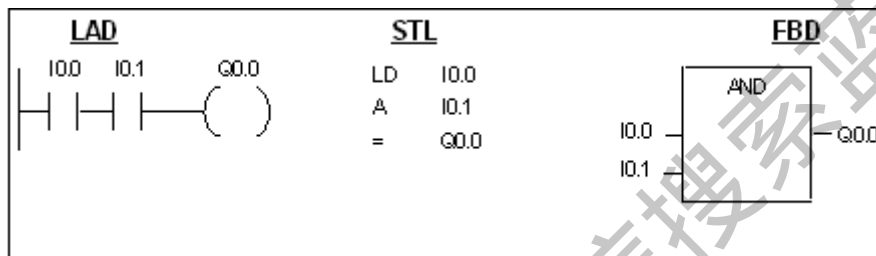
### STL程序

STL程序元素用执行所需功能的一套指令表示。与阶梯程序使用的图形显示不同，STL程序用文本格式显示。

### FBD程序

FBD程序元素用方框表示。[AND / OR \(与 / 或\) 方框 \(门\)](#)用于处理布尔信号，处理方式与阶梯接点相同。

下图显示不同的表示方法。



## 6.4 系统块配置

系统块可用于配置S7-200 CPU选项。

使用下列方法之一检视和编辑系统块，设置CPU选项：

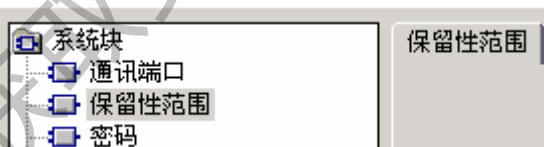
- 单击[浏览条](#)中的“系统块”按钮。
- 选择[检视 \(View\) > 元件 \(Components\) > 系统块 \(System Block\)](#) 菜单命令。
- 打开[指令树](#)中的“系统块”文件夹，然后打开某配置页。

当项目的CPU类型和版本能够支持特定选项时，这些系统块配置选项将被启用。

在下载或上载系统块之前，您必须成功地建立PC（STEP 7-Micro/WIN的位置）与CPU之间的通讯。

- 然后即可下载一个修改的系统块，以便为CPU提供新系统配置。
- 您也可以从CPU上载一个现有系统块，以便使STEP 7-Micro/WIN项目配置与CPU相匹配。

单击系统块树上分支即可修改项目配置。



下面列出全部系统块配置。

[端口](#)  
[保留范围](#)  
[密码](#)

[输出表](#)  
[数字输出表](#)  
[模拟输出表](#)

[输入过滤器](#)  
[数字输入过滤器](#)  
[模拟输入过滤器](#)

[脉冲捕获位](#)  
[背景时间](#)  
[EM \(扩充模块\) 配置](#)  
[配置LED](#)  
[扩大内存](#)

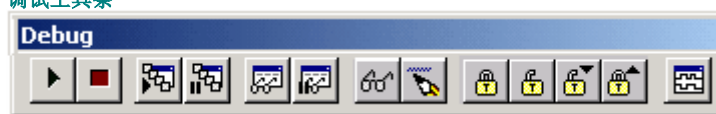
另请参阅：  
[编译程序](#)  
[应用程序用户参考手册](#)  
[入门指南目录](#)

## 6.5 如何使用状态工具确认程序逻辑

您成功地在运行的STEP 7-

Micro/WIN编程设备和PLC之间建立通讯并向PLC下载程序后，则可利用“调试”工具条的诊断特征。可单击工具条按钮或从“调试”菜单列表选择项目，选择调试工具。

[调试工具条](#)



本标题讨论下列主题：

[什么是“状态”？](#)  
[前提条件](#)  
[收集状态的不同方法](#)  
[以连续、瞬态图或触发暂停的方式收集状态](#)  
[PLC模式](#)  
[状态通讯与扫描循环](#)



[模拟程序条件（读取、写入、强制、取消强制）](#)  
[检查交叉引用和元素用法](#)  
[向程序下载改动](#)  
[重设用于调试和编辑的帧设置](#)

### 什么是“状态”？

[返回顶端](#)

‘状态’这一术语指显示程序在PLC中执行时有关PLC数据当前值和功率流的信息。您可以使用状态图和程序状态窗口读取、写入和强制PLC数据值。

在控制程序的执行过程中，PLC数据的动态改变可用三种不同方式检视：

#### 图状态

在一表格中显示状态数据：每行指定一个要监视的PLC数据值。您指定一个内存地址、格式、当前值及新值（如果使用写入命令）。

#### 趋势显示

用随时间而变的PLC数据之绘图跟踪状态数据：

您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势数据亦可在趋势视图中直接赋值。

#### 程序状态

在程序编辑器窗口中显示状态数据：当前PLC数据值会显示在引用该数据的STL语句或LAD/FBD图形旁边。LAD图形也显示功率流

### 程序状态和图状态 (或趋势视图) 窗口可以同时运行：

在状态图窗口写入或强制PLC数据将PLC数据改动应用于程序状态窗口。

在程序状态窗口写入或强制PLC数据还会将新PLC数据改动应用于状态图窗口。

### 检视状态数据的前提条件

[返回顶端](#)

您必须执行下列任务，才能监控状态数据或调试程序：


- 成功地在STEP 7-Micro/WIN和PLC之间建立通讯。
- 成功地向PLC下载程序。
- 欲检视连续状态更新，PLC必须位于RUN（运行）模式。否则，您只能看到对I/O的改动（如果有）。由于PLC程序辉溢葱校 I/O的改动不会对“状态”显示中的程序逻辑产生预期的影响。
- 如果您位于程序不在执行的区域（例如，子例行程序、中断例行程序或由于JMP指令被绕过的区域），不会有状态显示，因为不在对代码执行扫描。

### 注释：

- STEP 7-Micro/WIN中项目的时间标记和PLC必须匹配，才能激活状态数据通讯。时间标记比较（在您尝试打开状态时自动执行）繁 STEP 7-Micro/WIN中的项目状态表示正确反映实际PLC程序。如果时间标记不匹配，您可以检查“时间标记不匹配”对话框中南吕 斜恚 私饬男 繇斐刹黄 涸
- 下载程序后，您还应将PLC转换回RUN（运行）模式。否则会报告STOP（停止）模式状态数据，但PLC无法执行程序，因此您将无法看到预期的逻辑操作。

### 在程序编辑器窗口中收集状态的不同方法

[返回顶端](#)

单击“程序状态打开 / 关闭”按钮 ，或选择菜单命令 **调试 (Debug) > 程序状态 (Program Status)**，在程序编辑器窗口中显示PLC数据状态。状态数据收集按以前选择的模式开始。请参阅以下说明。

LAD和FBD程序有两种不同的程序状态数据收集模式。选择 **调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单命令会在状态模式复选标记打开和关闭之间切换。必须在程序状态操作开始之前选择状态模式。

- **执行状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选后）



- 扫描结束状态（当调试（Debug）> 使用执行状态（Use Execution Status）菜单项目被取消勾选后）



状态图始终使用“扫描结束”状态数据。以前的 STEP 7 Micro/WIN 版本用“扫描结束”状态数据建立 LAD 和 FBD 程序状态视图。如果您使用的是第一代 PLC，执行状态不受支持。如果您尝试执行执行状态，“扫描结束”状态开始。

#### 执行状态色彩：

- 程序被扫描时，电源横杆会带有色彩。
- 图形中的功率流用色彩表示。
- 接点？接点打开时，指令会带有色彩。
- 线圈？输出打开时，指令会带有色彩。
- 方框和 SUBR 指令？指令接通电源并准确无误地成功执行时，方框和 SUBR 指令带有色彩。
- 绿色计时器和计数器表示计时器和计数器包含有效数据。
- 红色表示执行的指令有误。
- 跳接和标签指令现用时，显示为功率流色彩。如果为非现用，则显示为灰色。
- 灰色（灰色为默认赋值）表示无功率流、指令未扫描（跳过或未调用）或位于 STOP（停止）模式的 PLC。
- 布尔功率流位（仅限 FBD）。

LAD、FBD 和 STL 执行状态视图显示操作数数值，并表示每条指令在位于扫描循环执行程序阶段时的功率流。执行状态可以显示可能被执行后续程序指令盖写的中间数据值。所有显示的 PLC 数据值均从一个程序扫描循环收集。

#### 扫描结束状态

- 接通电源或逻辑真的接点和线圈显示为蓝色（您可以选择 **工具（Tools）> 选项（Options）** 菜单项目并单击“色彩”标记，指定自己选择的色彩）。

“扫描结束”状态显示在程序扫描结束时读取的状态结果。这些结果可能不会反映 PLC 数据地址的所有数值改动，因为随后某绿蜂鸣铃语酱硫缘蛭 杞嘤 翱瞻苈慈牒椭恹漆慈胧 怠 S 捐诳炯 PLC 扫描循环和相对慢速的 PLC 状态数据通讯之间存诘乃俱快吹稿 吧 杞嘤 弊刺 允炯父鏊 枢 方嘤 辈杉 氛 奠怠

#### STL 程序中程序状态举例：

当您开启 STL 中的状态时，程序编辑器窗口被分为一个代码区（左侧）和一个状态区（右侧）。可以根据您希望监控的数值类型

在 STL 状态中共有三个可监控的数值类别：

- **操作数** 每条指令最多可监控三个操作数。
- **逻辑堆栈** 最多可监控四个来自逻辑堆栈的最新数值。
- **指令状态位** 最多可监控十二个状态位。

**工具（Tools）>**

**选项（Options）** 对话框的 STL 状态标记允许您选择或取消选择任何此类数值类别。如果您选择一个项目，该项目不会在“状态”显示中出现。

**状态图**

	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+16095	
5	T32	位	2#0	
6	T32	带符号	+0	

CHT1 /

---

**SIMATIC STL**

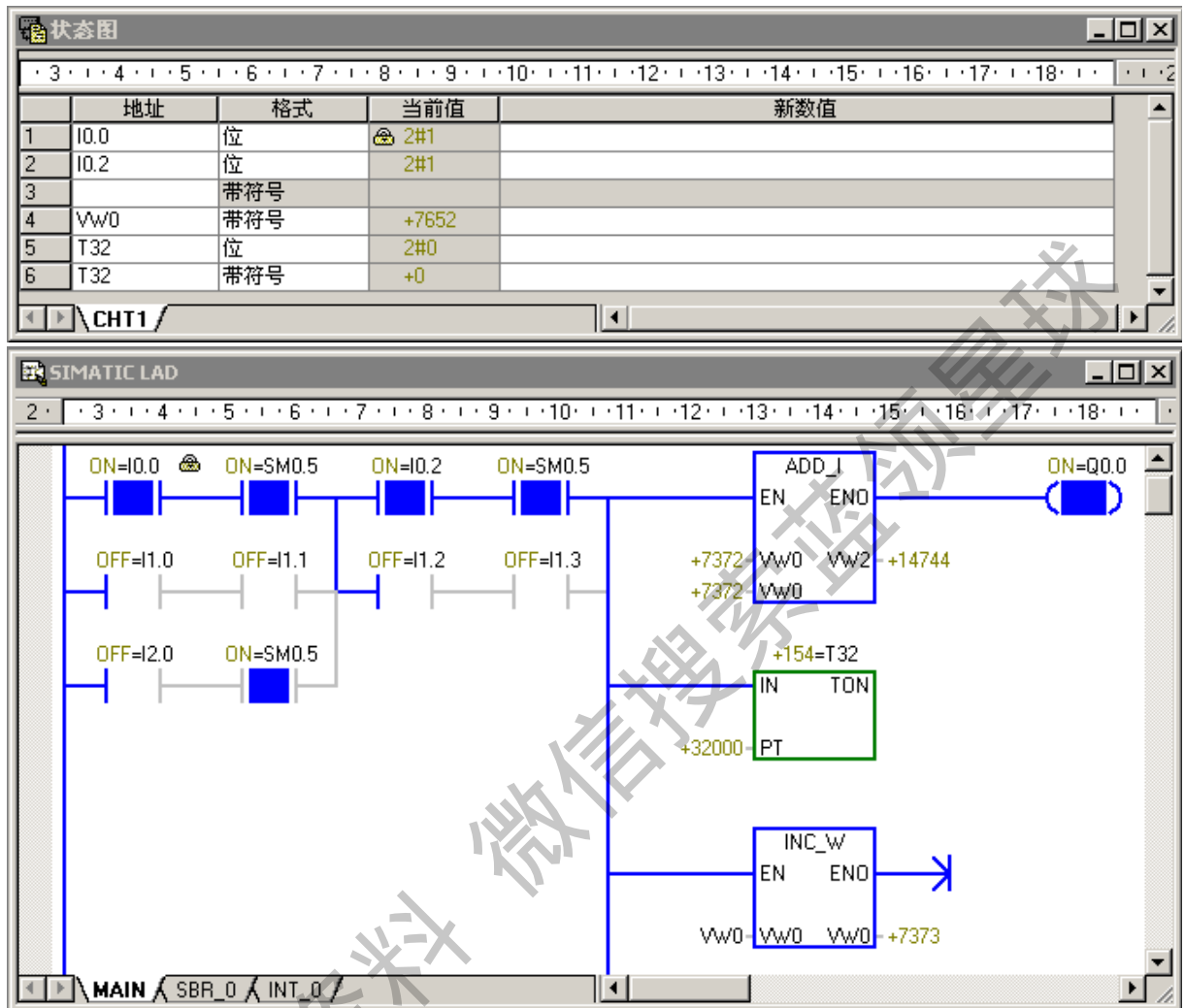
	操作数 1	操作数 2	操作数 3	0123	↑
LD I0.0	ON			1000	1
A SM0.5	ON			1000	1
LD I1.0	OFF			0100	0
A I1.1	OFF			0100	0
OLD				1000	1
LD I2.0	OFF			0100	0
A SM0.5	ON			0100	1
OLD				1000	1
LD I0.2	ON			1100	1
A SM0.5	ON			1100	1
LD I1.2	OFF			0110	0
A I1.3	OFF			0110	0
OLD				1100	1
ALD				1000	1
LPS				1100	1
MOVW VW0, VW2	+15919	+15919		1100	1
AENO				1100	1
+I VW0, VW2	+15919	+31838		1100	1
AENO				1100	1
= Q0.0	ON			1100	1
LRD				1100	1
TON T32, +32000	+288	+32000		1100	1
LRD				1100	1
TNOFF VW0	+15920			1100	1

MAIN / SBR\_0 / INT\_0 /

**LAD图形中执行状态举例:**

以下范例在LAD图形中使用LAD执行状态。状态图仅使用“扫描结束状态”数据收集。可在LAD图形和对应的状态图之间看出

烟筒谋消 渣允白 抵械哪承 厂收模 8 貌收鄂怯措盗刺 版占 伞煌 J 胶筒煌 耐久洞 砾咚魁 隆



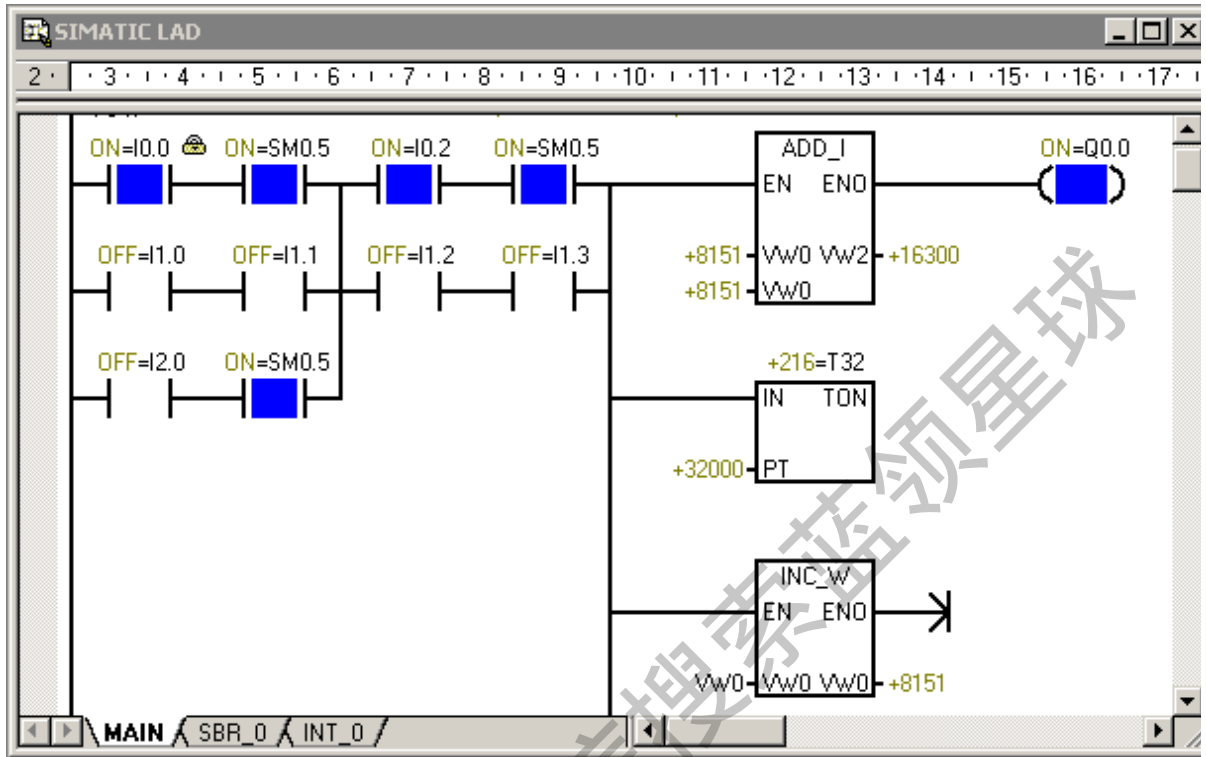
#### LAD图形中扫描结束状态举例：

以下范例在LAD图形中使用“扫描结束状态”。由于两个窗口（程序状态和图状态）均使用相同的扫描 PLC 结束数据，两个窗口允鞠啻 PLC 数据值。这种情形会引起误解，因为程序可以在收集最终“扫描结束”数值之前为相同的位置指定很多数值。中间临时数值不显示。在该范例中，**ADD\_I**方框未能正确执行，因为输入相加超出了带符号字整数的最大范围。与红色方框的执行状态范例不同，您必须读取状态值，才能看到  $Vw0 + Vw0 < Vw2$ 。

The screenshot displays the SIMATIC Manager interface. The top window, titled '状态图' (Status Diagram), shows a table of PLC data:

地址	格式	当前值	新数值
1	位	2#1	
2	位	2#1	
3	带符号		
4	带符号	+8418	
5	位	2#0	
6	带符号	+485	

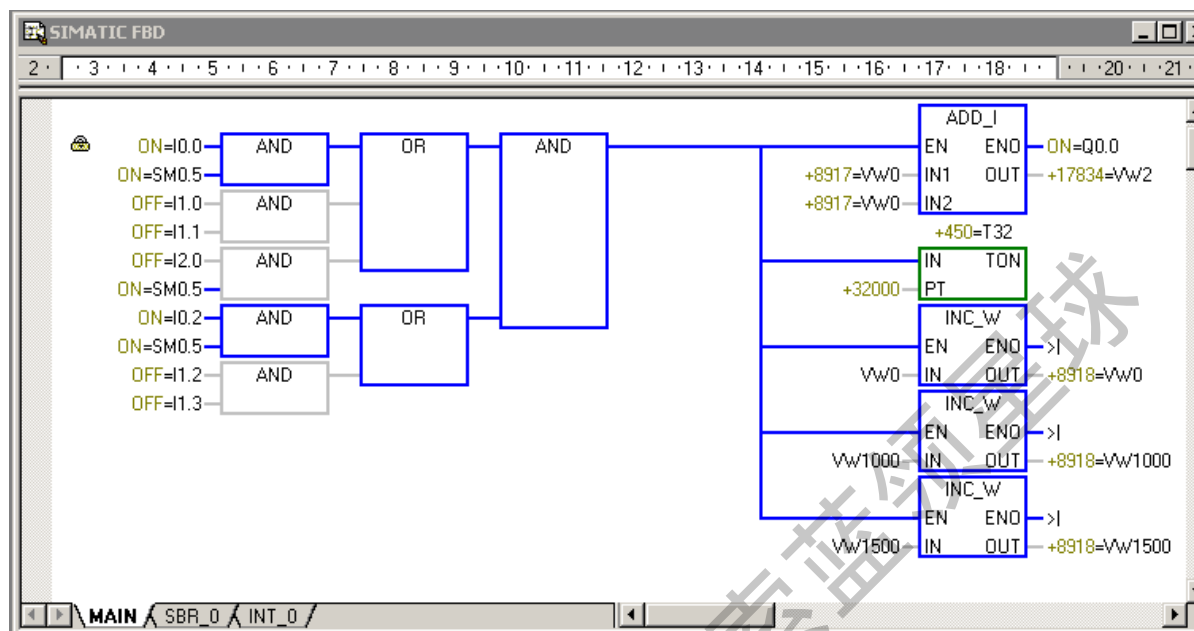
The status bar at the bottom indicates the current network is **CHT1**.



FBD图形中的执行状态举例:

地址	格式	当前值	新数值
1 I0.0	位	2#1	
2 I0.2	位	2#1	
3	带符号		
4 Vw0	带符号	+8958	
5 T32	位	2#0	
6 T32	带符号	+0	

获取更多资料



按照连续、瞬态图或触发（暂停程序状态）的方式收集状态值

[返回顶端](#)

**注释：**

“连续”并非意味着实时；而是指编程设备不断地从 PLC 轮询状态信息，在屏幕中显示，并按按照通讯允许的速度更新显示。可能

**连续**

- 打开程序编辑器窗口，并启动“程序状态”，当 PLC 位于 RUN（运行）模式时检视程序执行的连续状态更新。
- 打开状态图窗口，并启用“图状态”，当 PLC 位于 RUN（运行）模式时检视连续更新。



**瞬态图**

- 将 PLC 转换为 STOP（停止）模式，收集单个状态更新。当 PLC 位于 STOP（停止）模式时，您可以使用“多次扫描”功能检视一次或多次扫描。您还可以引用网络中的特殊内存 SM0.1 首次扫描位，读取该网络的首次扫描状态值。
- 关闭图状态，如果您希望收集单次状态图更新并不希望将 PLC 转换至 STOP（停止）模式，使用“单次读取”功能。如 PLC 转换为 STOP（停止）模式，并开启“图状态”，您也会收集单次状态更新。此外，您可以在检视状态图时使用“多次扫描”和“首次扫描”功能。

**执行状态模式（LAD / FBD / STL）中的程序状态触发暂停**

欲检视触发暂停 PLC 数据状态，您必须：

1. 将程序下载至 PLC，并将 PLC 设为 RUN（运行）模式。您还可以从 STOP（停止）模式开始触发暂停。如果您从 STOP（停止）模式开始，则可以捕获首次扫描。
2. 滚动程序编辑器窗口，显示您希望检视的程序网络。网络可以位于任何 POU（主程序、子例行程序或中断程序）中。
3. 使用 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单命令，将“使用执行状态”复选标记切换为打开。
4. 激活程序状态，开始状态数据通讯。



5. 单击“暂停程序状态” 工具条按钮，或用鼠标右键在程序编辑器中单击，并选择弹出菜单命令“暂停程序状态”。
6. 更新暂停之后，触发网络中的状态值在您检视时保持不变。当您准备移至另一个网络时，则单击“暂停程序状态” 工具条按钮，将暂停切换为关闭，并再次开始连续更新。

一旦获得新的状态信息，只要程序状态暂停功能被启用，该信息就会保持在屏幕上。在您取消选择程序状态暂停功能之前，无论 PLC 中实际变化速度多快，均不会进一步执行状态更新。如果没有程序状态暂停功能，屏幕就会不断刷新，状态数据在您读取和解释之前就可能消失（或者数据根本没有显示）。

**PLC RUN / STOP（运行 / 停止）模式**

[返回顶端](#)

使用以下一种方法改变PLC操作模式：

- 单击“运行”按钮，执行RUN（运行）模式，或者单击“停止”按钮，执行STOP（停止）模式。
- 选择**PLC > 运行 (RUN)** 菜单命令，执行RUN（运行）模式，或选择**PLC > 停止 (STOP)** 菜单命令，执行STOP（停止）模式。
- 以手动方式改变位于PLC上的模式开关。
- 在程序中插入一条STOP（停止）指令。

#### 注释：

欲使用STEP 7-Micro/WIN软件控制RUN / STOP（运行 / 停止）模式，必须在STEP 7-Micro/WIN和PLC之间存在一条通讯路径。此外，必须将PLC硬件模式开关设为TERM（终端）或RUN（运行）。将模式开关设为TERM（终端）不会改变PLC操作模式，但却允许STEP 7-Micro/WIN改变PLC操作模式。位于PLC前方的状态LED表示当前操作模式。当程序状态或状态图操作正在进行时，在STEP 7-Micro/WIN窗口右下方处附近的状态条上会出现一个RUN / STOP（运行 / 停止）指示灯。

虽然程序在STOP（停止）模式中不执行，PLC操作系统继续监控PLC（收集PLC RAM和I/O状态），将状态数据传递给STEP 7-Micro/WIN，并执行所有的“强制”或“取消强制”命令。当PLC位于STOP（停止）模式中时，您可以执行以下操作：

- 使用图状态或程序状态检视操作数的当前值。（此一步骤具有与执行“单次读取”相同的效果，因为程序未执行。）
- 您可以使用图状态或程序状态强制数值。使用图状态写入数值。
- 写入或强制输出。
- 执行有限次数扫描，并通过状态图和 / 或项目状态检视效果。

当PLC位于RUN（运行）模式时，您不得使用“首次扫描”或“多次扫描”功能。您可以在状态图中写入和强制数值，或使用LAD或FBD程序编辑器强制数值，方法与您在STOP（停止）模式中强制数值相同。您还可以执行以下操作（不得从STOP（停止）

- 使用图状态收集不断变化的PLC数据值的连续更新信息。（如果您希望使用单次更新，图状态必须关闭，才能使用“单次读取”命令。）
- 使用程序状态收集不断变化的PLC数据值的连续更新信息。
- 使用RUN（运行）模式功能中的“程序编辑”编辑程序，并将改动下载至PLC。

#### 状态通讯与扫描循环

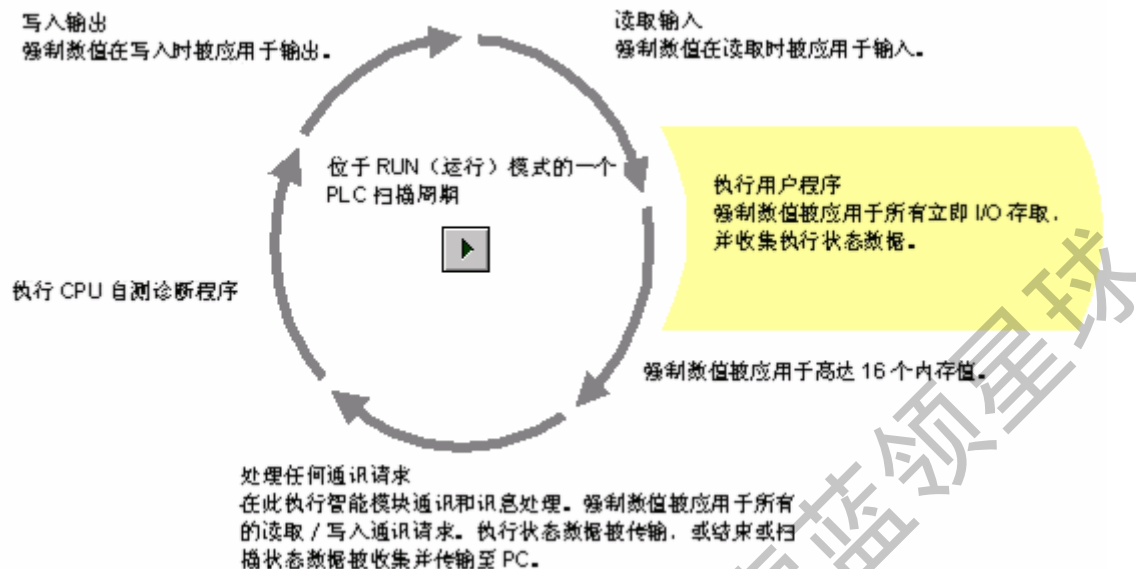
[返回顶端](#)

PLC在连续循环中读取输入、执行程序逻辑、写入输出和执行系统操作和通讯。该扫描循环速度极快，每秒执行多次。虽然STEP 7-Micro/WIN会快速发出状态请求，您应当认识到您检视的不是PLC中出现的每一个事件，这一点十分重要。

由于PLC和您观察的程序状态编程设备之间存在通讯时间滞后，您看到的显示的操作数数值总是在状态显示中改变之前即在PLC中改变，但没有在程序状态表示中改变。

如果您使用“扫描结束”状态模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被取消勾选时），您在几个扫描循环中采集数据。

如果您使用“执行状态”模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选时），所有显示的程序状态值一定来自同一个扫描循环。

**注释:**

一次只有一个状态窗口被授权运行状态。如果您尝试在任何其他与同一个 PLC 连接的 STEP 7 Micro/WIN 应用程序中运行状态，则会收到错误讯息。

**模拟进程条件 (读取、写入、强制、取消强制)**

[返回顶端](#)

您可以模拟进程条件，方法是在程序状态操作过程中从程序编辑器和从状态图向操作数写入或强制新数值。

使用“程序状态”和“图状态”按钮或“调试”菜单命令，开始状态数据通讯，并启用调试工具。您还可以使用硬件输入模拟器开关（西门子 (Siemens) 可提供）以手动方式在数字输入打开和关闭之间切换，并观察 PLC 上的 LED 指示灯，观察数字输出的条件。

从“调试”工具条或“调试”菜单列表存取以下功能。



**单次读取** (仅限状态图) 如果您希望获得一次“瞬态图” (对状态图中的所有数值一次更新)，使用“单次读取”。默认值为状态图连续轮询 PLC，获取状态更新信息。当您单击“状态图”时，状态图会切换为关闭，“单次读取”按钮变为灰色。



**全部写入** (仅限状态图)

在您完成对几行的“新数值”列的改动后，您可以使用“全部写入”，将所有需要的改动发送至 PLC。



**强制** 您可以在操作程序状态时从程序编辑器和状态图强制地址。欲强制程序状态地址，用鼠标右键单击一个参数，并选择从菜单列表强制。欲强制状态图地址为某一数值，您必须首先规定所需的数值，方法是读取该数值 (如果您希望强制当前值) 或键入该数值 (如果您希望将地址强制为一个新数值)。您一旦使用了强制功能，则在每次扫描时该数值均被重新应用于地址，直至您取消强制地址。“强制”功能盖写立即读取或立即写入指令。“强制”功能还盖写被配置为在转换为 STOP (停止) 模式时进入一个指定数值的输出：如果 CPU 进入 STOP (停止) 模式，输出被设为强制数值，而非配置的数值。通过将 V 或 M 内存强制为字节、字或双字，将 AI 或 AQ 内存在偶数字节边界上 (例如 AIW6 或 AIW14) 或 I/O 点上 (I 和 Q 位址) 强制为字，模拟逻辑条件。您可以一次强制 16 个 (V、M、AI 或 AQ) 地址和所有的 I/O 位 (所有 I 和 Q 位址)。 该图标表示该地址被明确强制。该地址数值在地址被取消强制之前无法改变。 该图标表示该地址被隐含强制。如果地址是一个被明确强制的较大地址的一部分，该地址则被认为是隐含强制。例如，如果 VW0 被强制，则 VB0 是隐含强制 (VB0 是 VW0 的第一个字节)。隐含强制数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制 VD0 (该地址包含 VB0、VB1、VB2 和 VB3)，则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制，则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。



**取消强制** 对于程序状态和状态图，选择一个地址，并使用“取消强制”按钮从该特定地址移除强制。



强制功能。您还可以选择（单击）一个参数，然后用鼠标右键单击该参数，查看“强制”和“取消强制”功能的弹出菜单。



#### 全部取消强制

如果您希望从所有的地址中移除强制功能，使用“全部取消强制”按钮。在应用“全部取消强制”之前，您无须选择单个地址。



#### 读取全部强制（仅限状态图）

当您使用“读取全部强制”功能时，状态图的“当前数值”列会为已经明确强制、隐含强制或部分隐含强制的所有地址显示一个图标。

### 检查交叉引用和元素用法

[返回顶端](#)

调试程序时，您可以决定是否增加、删除或编辑参数。

使用“交叉引用”窗口查看程序中当前指定参数的方法，这样可以防止您意外地重复赋值。

此外，如果您希望在RUN（运行）模式中执行程序编辑，并且某些网络使用正向或负向转换（EP和N接点或方框以及STL EU或ED指令），您一定要检查“交叉引用”信息中的EU/ED数目，以防重复指令号码。

### 向程序下载改动

[返回顶端](#)

一旦调试部分代码，如果PLC位于STOP（停止）模式，您可以编辑程序、下载程序、然后再次监控状态，查看改动是否发生作用。

您不必进入STOP（停止）模式就可以在RUN（运行）模式中执行程序编辑并向PLC下载较小的改动。



**警告** 在RUN（运行）模式中向PLC下载改动时，您的改动会立即影响程序操作。不得出错；程序编辑中的错误可能导致人员死亡或严重伤害和/或设备损坏。仅限合格人员在RUN（运行）模式中执行程序编辑。

### 重设用于调试和编辑的帧设置

[返回顶端](#)

欲重设将窗口和工具条的可视性、尺寸和位置设为默认值，选择**检视（View）> 帧（Frame）> 全部重设（Reset All）**菜单命令。

您有两套不同的工作空间窗口的便利，可在调试程序时获得较大的灵活性：窗口可视性和定位。这是比正常编辑空间更大的检视空间。当您进入程序状态时，您已经设置的任何窗口定位保持初始进入的状态；如果您关闭程序状态，窗口保持初始进入的状态。同样，任何在编辑程序时执行的窗口定位均在运行时与程序状态分开维护。

[返回顶端](#)

#### 另请参阅：

[如何在程序编辑器中显示状态（GS 7.2）](#)

[如何在状态图中显示状态（GS 7.3）](#)

[如何执行有限数目扫描（GS 7.4）](#)

[如何下载程序（GS 6.3）](#)

[时间标记不匹配错误](#)（确保编程设备中的项目与PLC中的项目相匹配）

[交叉引用和元素用法](#)（确保程序编辑不引起重复赋值）

[在运行模式中执行程序编辑](#)

[PLC运行/停止模式](#)

[在停止模式中写入和强制输出](#)

[安装和移除S7-200设备](#)

[接地和布线指南](#)

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part

# VIII

获取更多资料 微信搜索 蓝领星球

## 7 设置通讯

### 7.1 了解S7-200网络通讯的基本知识

S7-200的设计旨在解决您的通讯和联网需求，不仅支持最简单的网络，而且支持更复杂的网络。S7-200还提供允许您与其他设备通讯的工具，例如，使用自身通讯协议的打印机和计量器。STEP 7-Micro/WIN使设置和配置您的网络变得简便易行。

本标题讨论以下内容：

[主从设备网络定义](#)  
[波特率和网络地址定义](#)  
[为STEP 7-Micro/WIN设置波特率和网络地址](#)  
[S7-200 CPU设置波特率和网络地址](#)  
[设置远程地址](#)  
[在网络上搜索CPU](#)

#### 主从设备网络定义

[返回顶端](#)

S7-200支持主从设备网络，可在网络中用作主设备或从属设备，而STEP 7-Micro/WIN始终用作主设备。



**提示**如果您使用Windows NT和PC/PPI电缆，则无法在网络中使用任何其他主设备。

- **主设备：**网络中的主设备是可以向网络中的另一台设备发出请求的设备。主设备还能够对来自网络中其他主设备的请求作出应答。通常，主设备包括STEP 7-Micro/WIN、人机接口设备，例如TD 200和S7-300或S7-400 PLC。当S7-200向另一台S7-200请求信息时，则用作主设备（对等通讯）。



**提示**TP070无法用在配备另一台主设备的网络上。

- **从属设备：**配置为从属设备的设备，只能对来自一台主设备的请求作出应答；从属设备从来不会主动发出请求。S7-200在大多数网络中用作从属设备。作为从属设备，S7-200对网络主设备（例如，操作员面板或STEP 7-Micro/WIN）发出的请求作出应答。

#### 波特率和网络地址定义

[返回顶端](#)

数据在网络中的传输速度称为**波特率**，通常以千波特（kbaud）兆波特（Mbaud）为单位。波特率测量在某一特定时间内传输的位数。19.2 kbaud的波特率表示每秒19200位的传输速率。

在某一特定网络上通讯的每台设备必须被配置为以相同的波特率传输数据。因此，网络上速度最快的波特率由与网络连接的速度最快的设备决定。

下表列出S7-200支持的波特率：

网络	波特率
标准网络	9.6 kbaud至187.5 kbaud
使用EM 277的网络	9.6 kbaud至12 Mbaud
使用自由端口模式的网络	1200 baud至115.2 kbaud

网络地址是您为网络上每台设备指定的一个独特号码。该独特的网络地址确保将数据传输至正确的设备，并从正确的设备检索数据。S7-200支持0至126的网络地址。对于配备两个端口的S7-200，每个端口可以有一个网络地址。下表列出了S7-200设备的默认（工厂）设置：

S7-200设备	默认地址
STEP 7-Micro/WIN	
HMI（TD 200、TP或OP）	1
S7-200 CPU	

#### 为STEP 7-Micro/WIN设置波特率和网络地址

[返回顶端](#)

如果您需要为STEP 7-

Micro/WIN配置波特率和网络地址，您可以从PG/PC接口对话框操作。波特率必须与网络上的其他设备相同，网络地址必须具卸捞甌浴

通常，您无须更改STEP 7-Micro/WIN的网络地址（0）。如果您的网络包含另一个程序包，例如STEP 7，您可能需要更改STEP 7-Micro/WIN的网络地址。

为STEP 7-Micro/WIN配置波特率和网络地址十分简便：



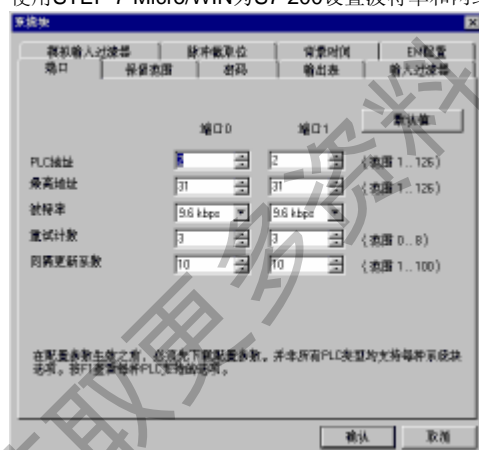
1. 单击应用程序左侧的“PG/PC接口”按钮。
2. 单击“设置PG/PC接口”对话框中的“属性”按钮。
3. 为STEP 7-Micro/WIN选择网络地址。
4. 为STEP 7-Micro/WIN选择波特率。

### 为S7-200设置波特率和网络地址

[返回顶端](#)

您可以为S7-200配置波特率和网络地址。S7-200系统块存储CPU与网络连接时使用的波特率和网络地址。您为S7-200选择参数后，必须在改动生效之前将系统块下载至S7-200。每台S7-200 CPU的默认波特率为9.6 kbaud，默认网络地址为2。

使用STEP 7-Micro/WIN为S7-200设置波特率和网络地址：



1. 选择浏览条中的“系统块”图标或选择**检视 > 元件 > 系统块 (View > Component > System Block)** 菜单命令。
2. 为S7-200选择网络地址。
3. 为S7-200选择波特率。
4. 将系统块下载至S7-200。

### 设置远程地址

[返回顶端](#)

您可以通过设置称为“远程”地址的参数方便地更改通讯目标PLC。在您建立通讯频道之前，必须设置“远程”地址，使之与勘S7-200相匹配。波特率也必须匹配。

欲更改远程地址：



1. 单击浏览条中的“通讯”图标或选择**检视 > 元件 > 通讯 (View > Component > Communications)** 菜单命令。
2. 在“通讯”对话框中，选择远程地址。确保COM端口的参数（波特率）、远程CPU端口和PC/PPI电缆与目标（远程）PLC和网络相匹配。否则通讯会失败。

### 在网上搜索CPU

[返回顶端](#)

STEP 7-Micro/WIN允许您搜索和识别与您的网络连接的CPU。搜索S7-200 PLC时，您还可以按照特定波特率搜索网络或选择所有波特率。

1. 如果您希望搜索所有的波特率，打开“通讯”对话框，并选择对应的选项。
2. 双击“刷新”图标，开始搜索。

## 7.2 我的通讯选择是什么？

S7-200 CPU支持以下一种或多种通讯能力，此类通讯能力允许您配置应用程序要求的网络性能和功能。

您的网络可支持以下一种或多种通讯协议：

- [点至点接口 \(PPI\)](#)
- [多点接口 \(MPI\)](#)
- [PROFIBUS](#)

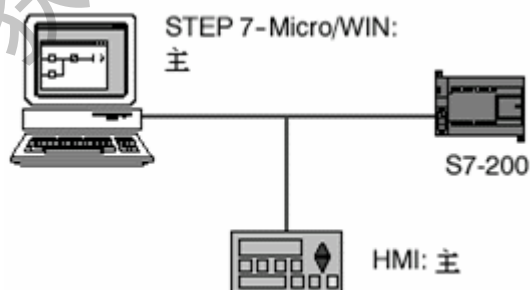
根据通讯协议的开放系统互连 (OSI) 七层模型，此类协议在令牌环网络中使用，该网络符合《欧洲标准 EN 50170》中定义的PROFIBUS标准。此类协议是基于字符的异步协议，配有一个起始位、八个数据位、偶数校验和一个停止位。M < 吨 U 【鲇愕厥雅钠鹦己屯 V 梢址 (0) 试春湍康恼镜 1 贰 (0) 才褪 蕊煽哨约燧楹汀 4 死嘈 榭稍谕 柜璧 缙贤 痹诵校 互相不会干扰，但每个协议的波特率必须相同。

请参阅[PG / PC与S7-200 CPU通讯选项表](#)中的通讯连接选项总结。

### PPI协议

[返回顶端](#)

PPI是一个主从设备协议：主设备向从属设备发出请求，从属设备作出应答。



从属设备不主动发出讯息，而是等候主设备向其发出请求或查询，要求应答。

主设备通过由PPI协议管理的共享连接与从属设备通讯。PPI不限制能够与任何一台从属设备通讯的主设备数目；但是，您无法在网络中安装32台以上主设备。

选择“PPI高级协议”允许网络设备在设备之间建立逻辑连接。若使用“PPI高级协议”，每台设备可提供的连接数目有限。下表显示由S7-200提供的连接数目。

如果您在用户程序中启用PPI主设备模式，S7-200

CPU可在处于RUN（运行）模式时用作主设备。启用PPI主设备模式后，您可以使用“网络读取”（NETR）或“网络写入”（NETW）从其他S7-200 CPU读取数据或向S7-200 CPU写入数据。S7-200用作PPI主设备时，作为从属设备应答来自其他主设备的请求。

您可以使用PPI协议与所有的S7-200 CPU通讯。欲与EM 277通讯，您必须启用“PPI高级协议”。

模块	波特率	连接
S7-200 CPU	端口0	端口1
kbaud或187.5 kbaud	个 4个	.6 kbaud、19.2 kbaud或187.5 kbaud 9.6 kbaud、19.2
EM 277模块		9.6 kbaud至12 Mbaud 每个模块6个

请遵循以下步骤设置PPI参数：

1. 在“站参数”区域的PPI标记上，选择“地址”方框中的一个数字。该数字表示您希望将STEP 7-Micro/WIN放置在可编程控制器网络中的位置。您正在运行STEP 7-Micro/WIN的个人计算机的默认站址是0。网络上的第一台PLC的默认站址是2。网络上的每台设备（PC、PLC等）必须有一个独特的站址；请勿将相同的地址指定给多台设备。
2. 在“超时”方框中选择一个数值。该数值代表您希望通讯驱动程序尝试建立连接花费的时间。默认值应当有足够的长时间。
3. 确定您是否希望将STEP 7-Micro/WIN用在配备多台主设备的网络上。您可以保留“多台主设备网络”方框中的勾选符号，除非您使用的是调制解调器连接。STEP 7-Micro/WIN不支持该功能。
4. 按照您希望STEP 7-Micro/WIN网络上通讯的速率设置传输速率。PPI电缆支持9.6 kbaud、19.2 kbaud和187.5 kbaud。
5. 选择最高站址（节点地址）。此为STEP 7-Micro/WIN停止查找网络上的其他主设备的地址。



图7 属性 ?PC/PPI 电缆 (PPI) 对话框，PPI标记

6. 单击“本地连接”标记。请参阅图8。
7. 在“本地连接”标记中，选择PC/PPI电缆与之连接的COM端口。如果您使用的是调制解调器，选择调制解调器连接COM端口，并选择“使用调制解调器”复选框。
8. 单击“确定”，退出“设置PG/PC接口”对话框。

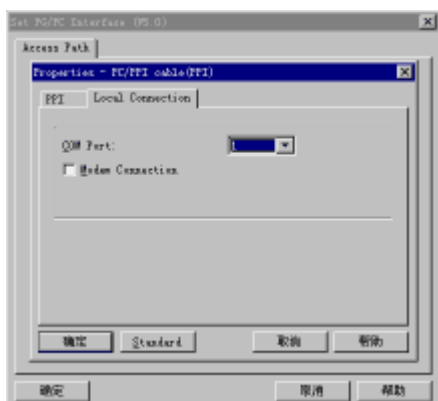


图8 属性 ?PC/PPI 电缆 (PPI) 对话框, 本地连接标记

#### 注释:

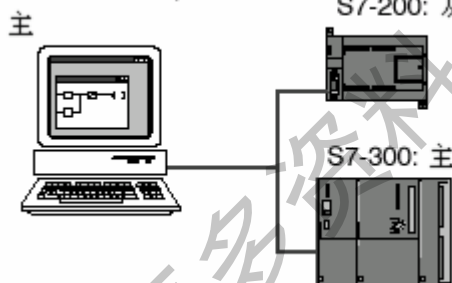
- 与S7-200 CPU通讯时, STEP 7-Micro/WIN的默认值是多台主设备PPI协议。该协议允许STEP 7-Micro/WIN与其他主设备 (TD 200和操作员面板) 同时在网络中存在。可勾选PG/PC接口“PC/PPI电缆属性”对话框中的“多台主设备网络”复选框, 启用该模式。
- 在使用单台主设备协议时, STEP 7-Micro/WIN还支持单台主设备PPI协议。STEP 7-Micro/WIN假设PPI协议是网络上的唯一主设备, 不与其他主设备共享网络。用调制解调器或噪音很高的网络传输时, 请勾选PG/PC接口“PC/PPI电缆属性”对话框中“多台主设备网络”复选框内的勾选符号, 选择单台主设备模式。

#### MPI协议

[返回顶端](#)

MPI允许主设备与主设备或主设备与从属设备之间的通讯。

#### STEP 7-Micro/WIN:



欲与S7-200 CPU通讯, STEP 7-Micro/WIN建立主设备与从属设备连接。MPI协议无法与用作主设备的S7-200 CPU通讯。网络设备通过任何两台设备之间的 (由MPI协议管理的) 不同连接进行通讯。设备之间的通讯受 S7-200 CPU或EM 277模块支持的限制。请参阅下表中的受 S7-200支持的限制。

对于MPI协议, S7-300和S7-400 PLC使用XGET和XPUT指令从S7-200 CPU读取数据和向S7-200 CPU写入数据。有关这些指令的信息, 请参阅您的 S7-300或S7-400编程手册。

如果您在使用CP

5511、CP5611或MPI及PPI协议, 从“设置PG/PC接口”对话框, 单击“属性..”按钮, 会显示XXX卡 (PPI) 的属性表, 其中“XXX”表示您安装的卡类型。例如, MPI-ISA。请参阅图10。

#### 注释:

当您与S7-200 CPU 215的1号端口通讯时, 请使用MPI协议。

请遵循以下步骤设置MPI参数:

1. 在PPI标记中, 选择“地址”方框中的一个数字。该数字表示您希望将 STEP 7-Micro/WIN在可编程控制器网络中放置的位置。
2. 在“超时”方框中选择一个数值。该数值代表您希望通讯驱动程序尝试建立连接花费的时间。默认值应当有足够的时
3. 按照您希望STEP 7-Micro/WIN在网络上通讯的速率设置传输速率。

4. 选择最高站址（节点地址）。此为STEP 7-Micro/WIN停止查找网络上的其他主设备的地址。
5. 单击“确定”退出“设置PG/PC接口”对话框。

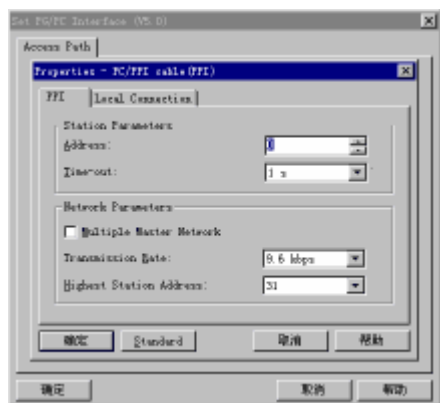


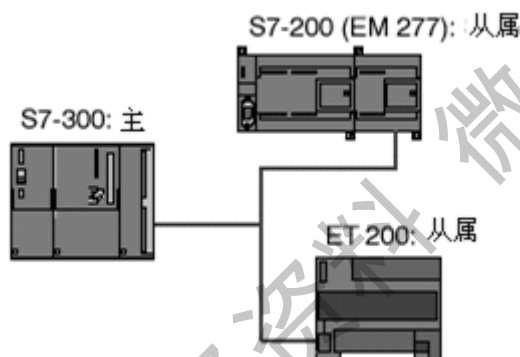
图10 MPI-ISA卡（PPI）属性表

## PROFIBUS协议

[返回顶端](#)

PROFIBUS协议用于与分布式I/O设备（远程I/O）进行高速通讯。各类制造商提供多种PROFIBUS设备。此类设备从简单的输

通常，PROFIBUS网络有一台主设备和几台从属I/O设备。



主设备的配置可了解连接的I/O从属设备的类型以及连接的地址。主设备初始化网络，并核实网络上的从属设备与配置相符。主设备不断将输出数据写入从属设备，并从从属设备读取输入数据。

当一台DP主设备成功地配置了一台从属设备后，该主设备则拥有该从属设备。如果网络上还有第二台主设备，该主设备对第一台主设备拥有的从属设备只有有限的存取权。

## 7.3 设置通讯

### 从何处开始设置通讯？

您可以在从下列点开始设置通讯：

- 在STEP 7-Micro/WIN软件安装的最后一个步骤
- 在STEP 7-Micro/WIN中，选择PG/PC接口对话框
- Windows控制面板的“设置PG/PC”接口图标

### 如何选择通讯配置？

下表显示可能的硬件配置以及STEP 7-Micro/WIN支持的波特率。表1用于CPU 22x模块，表2用于Profibus DP连接，表3用于CPU 21x模块，表4用于以太网和互联网连接。

U = 受支持



表 1 S7- 22x CPU 通讯						Windows 版本
S7-200 CPU 型号	微机 I/O 端口	西门子 I/O 硬件	硬件和电缆类型	数据 COM 速率	通讯协议	2000 XP
221 222 224 224XP 226	串行 COM RS232	PC/PPI 电缆	将微机端口与 S7-200 CPU 连接的西门子电缆转换器。会执行向 RS485 信号电平的线内转换。	9.6 千波特 至 19.2 千波特	<sup>1</sup> PPI 单台主设备	✓
		RS-232/PPI 多台主设备电缆		9.6 千波特 至 187.5 千波特	<sup>1</sup> PPI 单台主设备 <sup>2</sup> 多台主设备 <sup>3</sup> PPI 高级	✓ ✓ ✓
	串行 COM USB	USB/PPI 多台主设备电缆		9.6 千波特 至 187.5 千波特	PPI 单台主设备	✓
					多台主设备 <sup>1</sup> PPI 高级	✓ ✓
	I/O 总线卡	CP 5511	已集成在 PC 或 PCI 卡 (第 3 版或更高版本) 和 MPI 电缆内	9.6 千波特 至 187.5 千波特	PPI 多台主设备	✓
					<sup>1</sup> PPI 高级	✓
					<sup>2</sup> MPI 或 PROFIBUS	✓
					CP 5511	笔记本微机的 II 型 PCMCIA 卡和 MPI 电缆
	CP 5512	笔记本微机的 II 型或 III 型 PCMCIA 卡和 MPI 电缆	9.6 千波特 至 187.5 千波特	多台主设备 <sup>1</sup> PPI 高级 <sup>2</sup> MPI 或 PROFIBUS	✓ ✓ ✓	

<sup>1</sup> 支持使用 PPI 连接的外部调制解调器 (PC COM 端口—RS232 电缆—本地调制解调器—电话 / 无线连接—远程调制解调器—PC/PPI 电缆—S7-200)

<sup>2</sup> 高级 PPI 协议与多台主设备 PPI 协议之比较:

优点: 高级 PPI 设定值在程序员和 CPU 之间建立一种更直接的逻辑连接。

高级 PPI 协议能在几台 CPU 使用 NETR (网络读取) 和 NETW (网络写入) 指令时改善复杂网络的通讯状况。

缺点: 较之多台主设备 PPI, 高级 PPI 限制可供使用的连接数目。

<sup>3</sup> MPI 协议不支持 9.6 Kbps。19.2 Kbps 是所支持的最低速率。

表2 S7-215、S7-22x EM277 DP 通讯						Windows 版本
S7-200 CPU 型 号	微机 I/O 端 口	西门子 I/O 硬件	硬件和电缆类型	数据 COM 速 率	通讯协议	2000 XP
215 DP 端口 22x. 带 EM 277 DP 模块	串行 COM RS232	RS-232/PPI 多台主设备 电缆	将微机端口与 CPU 连接的西 门子电缆转换器。会执行向 RS485 信号电平的线内转换。	9.6 千波特 至 187.5 千波特	PPI 高级	✓
		USB/PPI 多台 主设备电缆		9.6 千波特 至 187.5 千波特	PPI 高级	✓
	串行 COM RS232	微机适配器	与微机串行 COM 连接的西门 子电缆转换器。会执行自 RS232 至 RS485 和波特率的线 内转换。作为网络主设备工 作。	19.2 千波特 至 1.5 兆波特	MPI 或 PROFIBUS	✓
	I/O 总线 卡	CP 5511	已集成在 PC 或 PCI 卡 (第 3 版或更高版本) 和 MPI 电缆 内	9.6 千波特 至 12 兆波特	PPI 高级	✓
				9.6 千波特 至 12 兆波特	<sup>3</sup> MPI 或 PROFIBUS	✓
		CP 5512	笔记本微机的 II 型 PCMCIA 卡和 MPI 电缆	9.6 千波特 至 12 兆波特	PPI 高级	✓
9.6 千波特 至 12 兆波特				<sup>3</sup> MPI 或 PROFIBUS	✓	

<sup>3</sup> MPI 协议不支持 9.6 Kbps。19.2 Kbps 是所支持的最低速率。

表 3 S7-21x 通讯						Windows 版本	
S7-200 CPU 型号	微机 I/O 端口	西门子 I/O 硬件	硬件和电缆类型	数据 COM 速率	通讯协议	2000 XP	
212 214 215 端口 0 216	串行 COM RS232	PC/PPI 电缆	将微机端口与 S7-200 CPU 连接的西门子电缆。会执行向 RS485 信号电平的线内转换。	9.6 千波特 19.2 千波特	<sup>1</sup> PPI 单台主设备	✓	
					9.6 千波特 19.2 千波特	<sup>1</sup> PPI 单台主设备	✓
				<sup>2</sup> PPI 多台主设备		✓	
		<sup>3</sup> PPI 高级		✓			
		串行 COM USB		USB/PPI 多台主设备电缆	9.6 千波特 19.2 千波特	PPI 单台主设备	✓
						PPI 多台主设备	✓
	<sup>3</sup> PPI 高级		✓				
	I/O 总线卡	CP 5511	已集成在 PC 或 PCI 卡 (第 3 版或更高版本) 和 MPI 电缆内	9.6 千波特 19.2 千波特	PPI 多台主设备	✓	
					<sup>3</sup> PPI 高级	✓	
				<sup>3</sup> MPI 或 PROFIBUS	✓		
		CP 5511		笔记本微机的 II 型 PCMCIA 卡和 MPI 电缆	9.6 千波特 19.2 千波特	PPI 多台主设备	✓
						<sup>3</sup> PPI 高级	✓
<sup>3</sup> MPI 或 PROFIBUS					✓		
CP 5512	笔记本微机的 II 型或 III 型 PCMCIA 卡和 MPI 电缆	9.6 千波特 19.2 千波特	PPI 多台主设备	✓			
			<sup>3</sup> PPI 高级	✓			
		<sup>3</sup> MPI 或 PROFIBUS	✓				

<sup>1</sup> 支持使用 PPI 连接的外部调制解调器 (PC COM 端口—RS232 电缆—本地调制解调器—电话 / 无线连接—远程调制解调器—PC/PPI 电缆—S7-200)

<sup>2</sup> 高级 PPI 协议与多台主设备 PPI 协议之比较:

优点: 高级 PPI 设定值在程序员和 CPU 之间建立一种更直接的逻辑连接。

高级 PPI 协议能在几台 CPU 使用 NETR (网络读取) 和 NETW (网络写入) 指令时改善复杂网络的通讯状况。

缺点: CPU

212 和 214 必须为 1.10 发行版或更新版本。较之多台主设备 PPI, 高级 PPI 限制可供使用的连接数目。

<sup>3</sup> MPI 协议不支持 9.6 Kbps。19.2 Kbps 是所支持的最低速率。

表 4 S7-22x CP243-1 CP243-1 IT 以太网通讯					Windows 版本
S7-200 CPU 型号	以太网接口	通讯协议	数据 COM 速率	I/O 硬件	2000 XP
22x. 配备 CP243-1 以太网模块或 CP 243-1 IT 互联网模块	Windows TCP / IP	S7 服务	自动感测 10-100 兆波特半或全双工	非西门子以太网卡	✓
				CP 1512	✓
	SIMATIC Net	S7 服务	自动感测 10-100 兆波特半或全双工	CP 1612	✓
				CP 1613	✓

## 7.4 仅使用S7-200设备的网络配置样本

本标题显示以下网络配置范例

[仅使用S7-200设备的网络配置范例](#)

[使用S7-200、S7-300和S7-400设备的网络配置范例](#)

[PROFIBUS-DP网络配置范例](#)

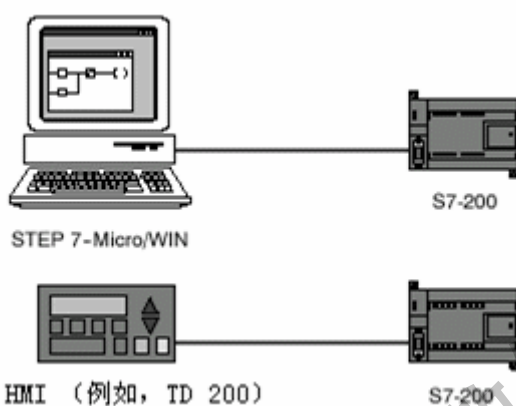
[工业以太网网络范例](#)

仅使用S7-200设备的网络配置范例

[返回顶端](#)

单台主设备PPI网络

下图显示两个网络范例。



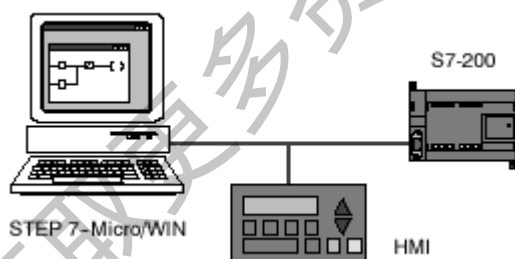
对于简单的单台主设备网络，编程站和S7-200 CPU通过PC/PPI电缆或安装在编程站中的通讯处理器（CP）卡连接。在第一个范例中，编程站（STEP 7-Micro/WIN）是网络主设备。在第二个范例中，一台人机接口（HMI）设备（例如TD 200、TP或OP）是网络主设备。

在两个网络范例中，S7-200 CPU是对来自主设备的请求作出应答的从属设备。

对于单台主设备PPI配置，您将STEP 7-Micro/WIN配置为使用PPI协议：选择单台主设备、多台主设备或PPI高级协议。

多台主设备PPI网络

下图显示配备一台从属设备的多台主设备网络范例。

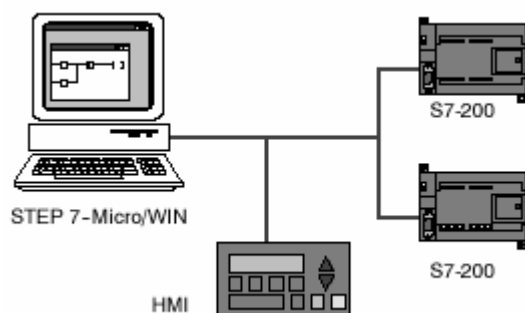


编程站（STEP 7-Micro/WIN）使用CP卡或PC/PPI电缆，STEP 7-Micro/WIN和HMI设备共享设备。

STEP 7-Micro/WIN和HMI设备均为主设备，必须具有不同的网络地址。S7-200 CPU是从属设备。

对于多台主设备存取一台从属设备的网络，将STEP 7-Micro/WIN配置为使用PPI协议，并启用多台主设备驱动程序。PPI高级协议是最佳选择。您还可以购买PPI多台主设备电缆，用于多台主设备网络。如果使用此种电缆，多台主设备和高级PPI检查盒则无任何意义。电缆无须配置即会自动调整为适当的设置。

下图显示多台主设备与多台从属设备通讯的PPI网络。



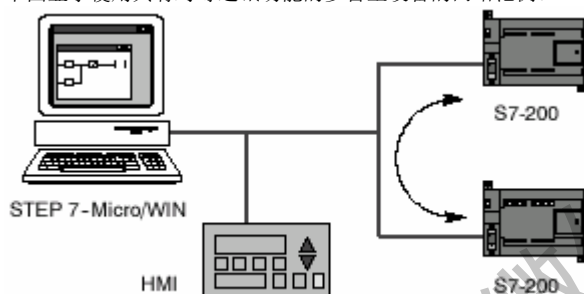
在该范例中，STEP 7-Micro/WIN和HMI均可请求从任何S7-200 CPU从属设备获得数据。STEP 7-Micro/WIN和HMI设备共享网络。

所有的设备（主设备和从属设备）均有不同的网络地址。

对于配备多台主设备和多台从属设备的PPI网络，您将STEP 7-Micro/WIN配置为使用PPI协议，并启用多台主设备驱动程序。PPI高级协议是最佳选择。

#### 复杂的PPI网络

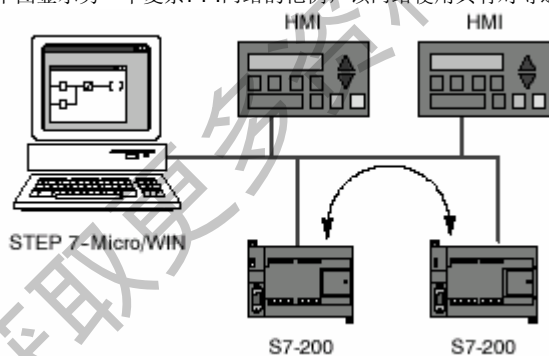
下图显示使用具有对等通讯功能的多台主设备的网络范例。



STEP 7-Micro/WIN和HMI设备在网络上从S7-200 CPU读取数据和向S7-200 CPU写入数据，S7-200 CPU使用“网络读取”（NETR）和“网络写入”（NETW）指令相互读取和写入数据（对等通讯）。

对于此类复杂的PPI网络，将STEP 7-Micro/WIN配置为使用PPI协议，并启用多台主设备驱动程序。PPI高级协议是最佳选择。

下图显示另一个复杂PPI网络的范例，该网络使用具有对等通讯功能的多台主设备。



在该范例中，每台HMI监管一台S7-200 CPU。

S7-200 CPU使用NETR和NETW命令相互读取和写入数据（对等通讯）。

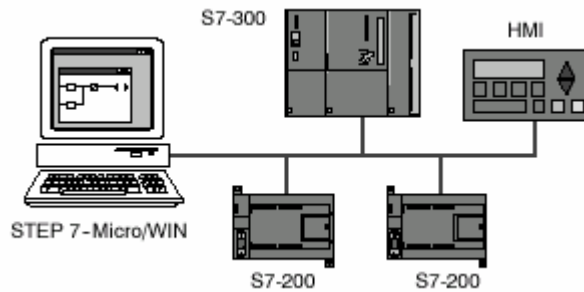
对于该网络，将STEP 7-Micro/WIN配置为使用PPI协议，并启用多台主设备驱动程序。PPI高级协议是最佳选择。

#### 使用S7-200、S7-300和S7-400设备的网络配置范例

[返回顶端](#)

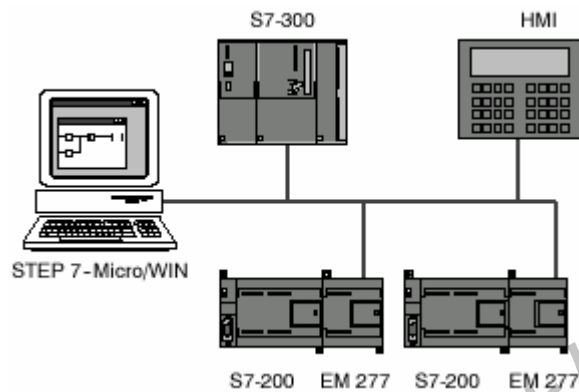
#### 波特率最高为187.5 kBaud的网络

在以上显示的网络范例中，S7-300使用XPUT和XGET指令与S7-200 CPU通讯。S7-300无法在主设备模式中S7-200 CPU通讯。



#### 波特率超过187.5 kbaud（最高为12 Mbaud）的网络

对于超过187.5 kbaud的波特率，S7-200 CPU必须使用EM 277模块与网络连接。



编程站（STEP 7-Micro/WIN）必须通过通讯处理器（CP）卡连接。

在该配置范例中，S7-300可使用XPUT和XGET指令与S7-200通讯，XMI可监控S7-200或S7-300。

EM 277始终是从属设备。

STEP 7-Micro/WIN可通过EM 277对任何一台S7-200 CPU进行编程或监控。

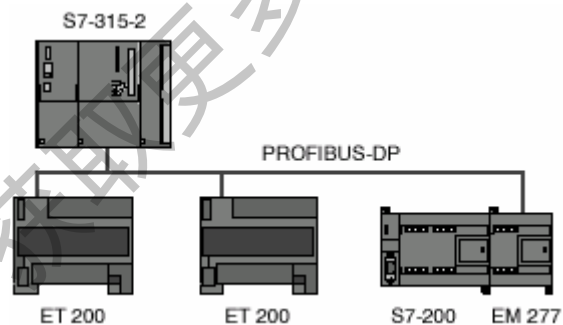
欲与EM 277通讯，将STEP 7-Micro/WIN配置为使用PPI协议，并启用PPI高级驱动程序。

#### PROFIBUS-DP网络配置范例

[返回顶端](#)

#### 将S7-315-2用作PROFIBUS主设备以及将EM 277用作PROFIBUS从属设备的网络

下图显示将S7-315-2用作PROFIBUS主设备的PROFIBUS网络范例。EM 277模块用作PROFIBUS从属设备。

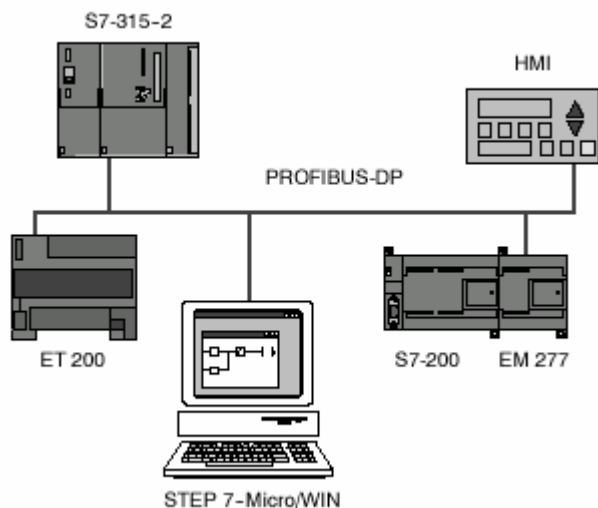


S7-315-2 DP可从EM 277读取或向EM 277写入从1个字节至128个字节的的数据。S7-315-2 DP在S7-200中读取或写入V内存位置。

该网络支持从9600 baud至12 Mbaud的波特率。

#### 配备STEP 7-Micro/WIN和HMI的网络

下图显示将S7-315-2 DP用作PROFIBUS主设备以及将EM 277用作PROFIBUS从属设备的网络范例。



在该配置中，HMI通过EM 277监控S7-200。STEP 7-Micro/WIN通过EM 277为S7-200编程。该网络支持从9600 baud至12 Mbaud的波特率。（对于超过19.2 kbaud的波特率，STEP 7-Micro/WIN要求使用CP卡。）对于通用FMS配置文件，必须配置CP卡。

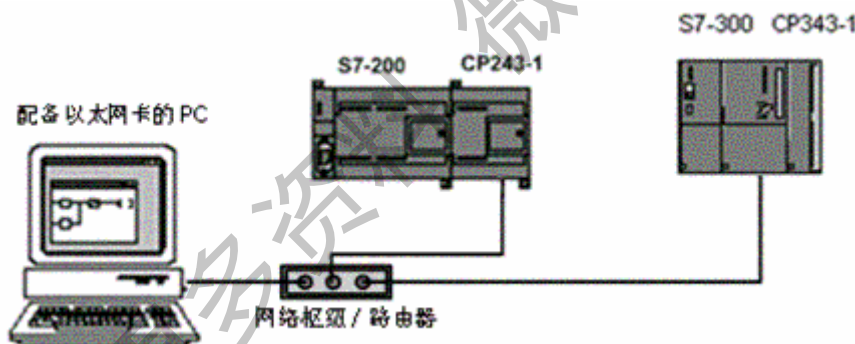
对于CP卡，将STEP 7-

Micro/WIN配置为使用PROFIBUS协议。如果网络中只有DP设备，选择DP或标准配置文件。如果网络中没有DP设备，则选择虫茫 DP/FMS) 配置文件。

#### 工业以太网网络范例

[返回顶端](#)

CP243-1以太网模块是允许S7-200 PLC与工业以太网网络链接的S7-200系列的通讯处理器。



以太网模块最多支持8个异步并行连接，从而允许S7-200 PLC与S7-300和S7-400设备通讯。您还可以建立与SIMATIC操作员面板（OP）的通讯。

欲与工业以太网网络通讯，STEP 7-Micro/WIN使用[以太网向导](#)建立以太网连接。

有关使用以太网模块的详情，请参阅[以太网通讯](#)。

## 7.5 安装和移除通讯接口

您可以使用图5中显示的“安装/取消安装接口”对话框安装或移除通讯硬件。对话框左侧是您可以安装的硬件类型列表，右侧是当前已安装的硬件类型列表。如果您使用Windows NT 4.0操作系统，在“已安装”列表框下方有一个“资源”按钮。

#### 安装硬件：

欲安装硬件，请遵循下列步骤：

1. 如图5所示，在“设置PG/PC接口”对话框中，单击“选择”，存取“安装/取消安装接口”对话框。
2. 从“选择”列表框，选择您已有的硬件类型。在窗口下方会显示您的选择说明。
3. 单击“安装”。
4. 硬件安装完成后，单击“关闭”。会出现“设置PG/PC接口”对话框，您的选择在“已使用的接口参数分配”列表框

中显示。

#### 移除硬件：

欲移除硬件，请遵循下列步骤：

1. 从右侧的已安装列表框中选择硬件。
2. 单击“取消安装”。
3. 移除硬件完成后，单击“关闭”。会出现“设置 PG/PC 接口”对话框，您的选择在“已使用接口参数分配”列表框中显示。

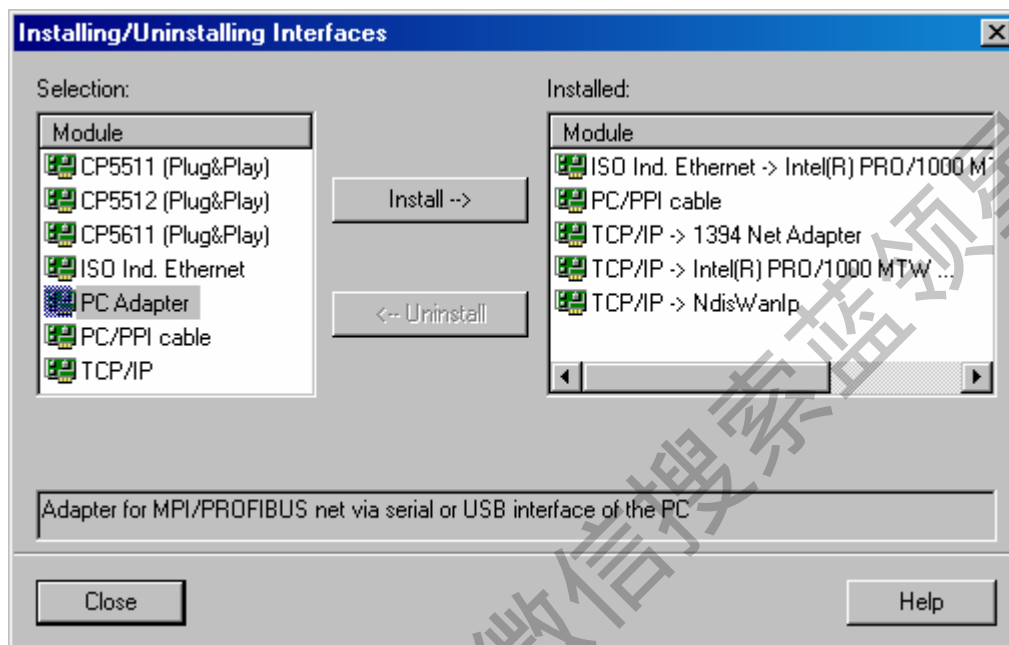


图5 安装 / 移除接口对话框

## 7.6 建立网络

### 一般指南

- 在可能受雷电电涌影响的任何布线上，始终安装适当的电涌抑制设备。
- 避免将低压信号导线和通讯电缆放入与交流电导线和快速转换的大功率直流电导线同一个线盒中。始终成对布线，将零线或中性线与热线或信号载线成对使用。
- S7-200 CPU 的通讯端口未绝缘。请考虑使用 RS-485 中继器为网络提供绝缘保护。



**当心** 带有不同的参考电位的互连装置会使无用的电流流入互连电缆。

此类无用的电流会造成通讯错误或损坏装置。

核实即将与通讯电缆连接的所有装置均共用一个电路参考电位或具有绝缘保护，以防无用的电流流入。详情请参阅 [使用绝缘](#) 或 [返回慕用摩偷缙凡慰嫉缥坏](#) 中的说明。

本标题讨论以下内容：

[确定网络的距离、传输速率和电缆](#)

[在网络中使用中继器](#)

[选择网络电缆](#)

[接头插针分配](#)

[偏流和设置网络电缆终端](#)

[在网络中使用HMI设备](#)

[用自由端口模式建立用户定义协议](#)

[在RS-232设备中使用PC/PPI电缆和自由端口模式](#)

[确定网络的距离、传输速率和电缆](#)

[返回顶端](#)



如上表所示，网络段的最大长度由两个因素决定：绝缘（使用 RS-485 中继器）和波特率。

当您连接不同电位的设备时，要求有绝缘装置。当地装置之间距离很远时，会存在不同的参考电位。即使接地装置之间的距离

波特率	未绝缘的 CPU 端口**	配备中继器或 EM 277 的 CPU 端口
9.6 kbaud 至 187.5 kbaud	米	1000 米
500 kbaud	不受支持	400 米
1 Mbaud 至 1.5 Mbaud	不受支持	200 米
3 Mbaud 至 12 Mbaud	不受支持	100 米

\*\* 不使用绝缘器或中继器时允许的最大距离为 50 米，此为网络段中第一个节点至最后一个节点的距离。

### 在网络中使用中继器

[返回顶端](#)

RS-485 中继器为网络段提供偏流和终端。您可以因以下目的使用中继器：

- **增加网络长度：**在网络中增加一台中继器可允许您将网络长度增加 50 米。如果您将两台中继器连接在一起，中间无
- **在网络中增加设备：**每个网络段最多可连接 32 台设备，波特率为 9600 时的最大长度为 50 米。使用一台中继器允许您
- **为不同的网络段绝缘：**为网络绝缘可提高传输质量，方法是将可能具有不同接地电位的网络段隔开。

网络中的一台中继器被计作网络段中的一个节点，即使未为该网络段指定网络地址亦如此。以下是配备中继器的网络范例。



### 选择网络电缆

[返回顶端](#)

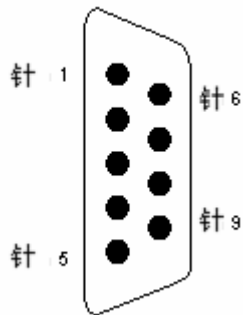
S7-200 网络使用双绞线电缆的 RS-485 标准。下表显示网络电缆的规格。您可以在一个网络段中最多连接 32 台设备。

规格	说明
电缆类型	屏蔽式，双绞线
回路电阻	$\leq 115 \Omega/\text{km}$
有效电容	30 pF/m
额定阻抗	约为 135 $\Omega$ 至 160 $\Omega$ （频率 = 3 MHz 至 20 MHz）
衰减	0.9 dB/100 m（频率 = 200 kHz）
横断面核心区域	0.3 mm <sup>2</sup> 至 0.5 mm <sup>2</sup>
电缆直径	8 mm $\pm$ 0.5 mm

### 接头插针分配

[返回顶端](#)

S7-200 CPU 中的通讯端口是符合《欧洲标准 EN 50170》中定义的 PROFIBUS 标准并与 RS-485 兼容的九针超小型 D 接头。下表显示为通讯端口提供实际连接的接头，并描述通讯端口插针分配状况。



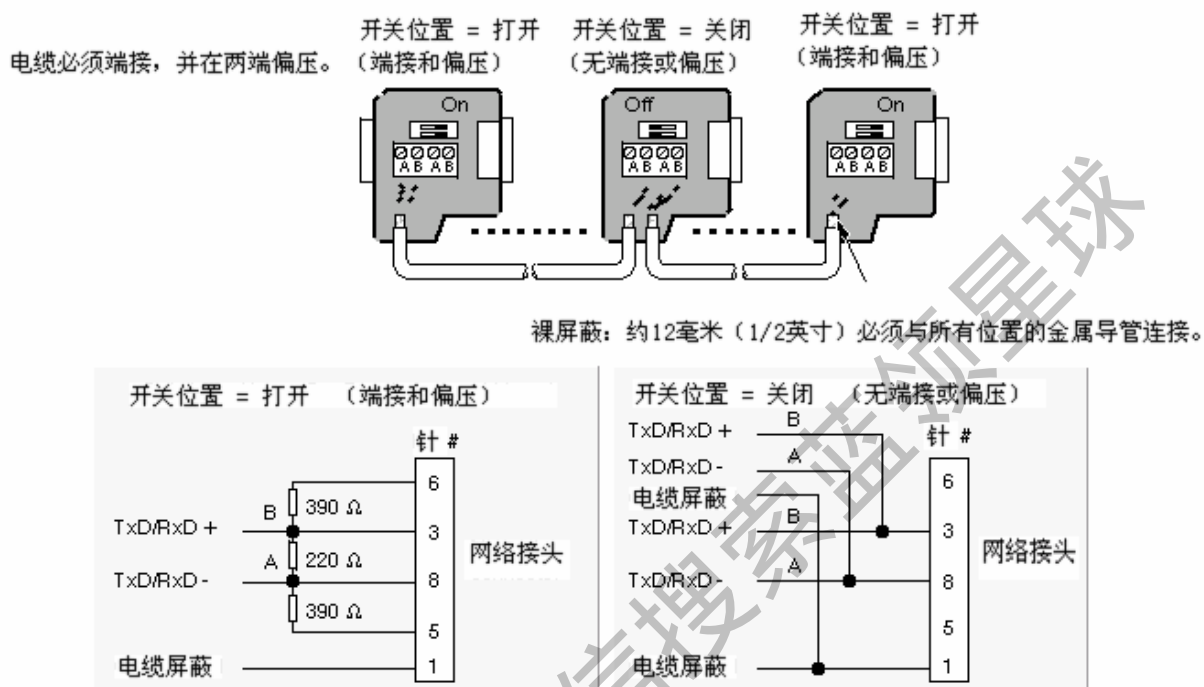
插针号码	信号	0号端口 / 1号端口
1	屏蔽	机壳接地
2	V回流	逻辑中性线
2	RS-485信号B	RS-485信号B
4	请求发送	RTS (TTL)
5	V回流	逻辑中性线
6	+5 V	+5 V, 100 Ω串联电阻器
7	+24 V	+24 V
8	RS-485信号A	RS-485信号A
9	不相关	10位协议选择 (输入)
接头外壳	屏蔽	机壳接地

#### 偏流和设置网络电缆终端

[返回顶端](#)

Siemens提供两种网络接头，可用于方便地将多台设备与网络连接：标准网络接头和包括编程端口的接头，该编程端口允许您暗陶净 HMI设备与网络连接，而不会干扰任何现有网络连接。编程端口接头通过编程端口传输所有来自 S7-200 CPU的信号（包括电源插针），这对于连接由 S7-200 CPU（例如TD 200）供电的设备尤其有用。

两种接头均有两套终端螺丝，用于连接输入和输出网络电缆。两种接头还配有开关，可选择性网络偏流和终端。下图显示电缆油返钠胀女 骺椭斩俗纯觥



### 使用网络上的HMI设备

[返回顶端](#)

S7-200 CPU支持多种Siemens和其他制造商生产的HMI设备。虽然其中一些HMI设备（例如，TD 200或TP070）不允许您选择设备使用的通讯协议，其他设备（例如，OP7和TP170）则允许您选择设备的通讯协议。

如果HMI设备允许您选择通讯协议，请考虑以下指南：

- 若是与S7-200 CPU通讯端口连接的HMI设备，如果网络上没有其他设备，为HMI设备选择PPI或MPI协议。
- 若是与EM 277 PROFIBUS模块连接的HMI设备，选择MPI或PROFIBUS协议。
  - 如果配备HMI设备的网络包括S7-300或S7-400 PLC，为HMI设备选择MPI协议。
  - 如果配备HMI设备的网络是PROFIBUS网络，为HMI设备选择PROFIBUS协议，并选择一个与PROFIBUS网络上其他主设备一致的配置文件。
- 若是与S7-200 CPU通讯端口连接的HMI设备，如果尚未被配置为主设备，为HMI设备选择PPI协议。PPI高级协议是最佳选择。MPI和PROFIBUS协议不支持将S7-200 CPU用作主设备。

### 用自由端口模式建立用户定义协议

[返回顶端](#)

自由端口模式允许程序控制S7-200

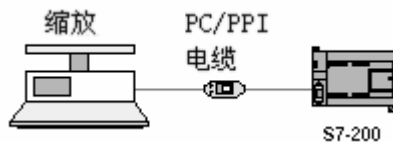
CPU的通讯端口。您可以使用自由端口模式使用户定义通讯协议与多种智能设备通讯。自由端口模式支持ASCII和二进制协议。

欲启用自由端口模式，可使用特殊内存字节SMB30（用于0号端口）和SMB130（用于1号端口）。您的程序使用以下功能控制通讯端口的操作：

- **传输指令（XMT）和传输中断：**“传输”指令允许S7-200 CPU从COM端口最多传输255个字符。传输完成时，传输中断向S7-200中的程序发出通知。
- **接收字符中断：**接收字符中断通知用户程序在COM端口中收到一个字符。程序则可根据正在执行的协议处理该字符。
- **接收指令（RCV）：**“接收”指令从COM端口接收整条讯息，完全收到讯息后，为用户程序生成中断。您使用S7-200的SM内存配置“接收”指令，根据定义的条件开始和停止讯息接收。“接收”指令允许程序根据具体字符或时间间隔开始或停止讯息。大多数协议可用“接收”指令执行。

自由端口模式仅限在S7-200处于RUN（运行）模式时才成为现用。将S7-200设为STOP（停止）模式会使所有的自由端口通讯暂停，通讯端口则返回至在S7-200系统块中配置的PPI协议设置。

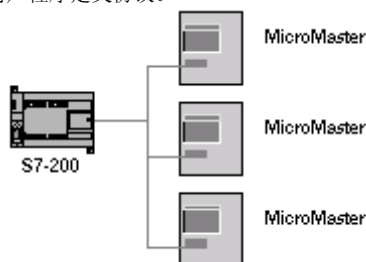
**网络配置**      **说明**  
在RS-



232连接中使用自由端口

- 即 S7-200 CPU，该电子标尺有一个RS-232端口。
- CPU上的RS-485端口连接。
- CPU使用自由端口与标尺通讯。
- kbaud。
- 用户程序定义协议。

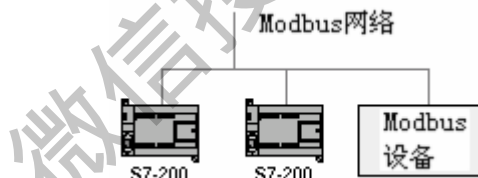
举例：使用配备电子标尺的S7-200 CPU，该电子标尺有一个RS-232端口。PC/PPI电缆将标尺上的RS-232端口与S7-200



使用USS协议

- MicroMaster 驱动器的S7-200 CPU。
- STEP 7-Micro/WIN提供一个USS库。
- CPU是主设备，驱动器是从属设备。有关USS程序范例，请参阅文档光盘中的“提示与技巧”一节。请参阅第28条提示。

举例：使用配备SIMODRIVE



建立模仿另一个网络中从属设备的用户程序

- 航 S7-200 CPU与Modbus网络连接。
- 200中的用户程序模仿Modbus从属设备。
- Micro/WIN提供一个Modbus库。有关Modbus程序范例，请参阅文档光盘中的“提示与技巧”一节。请参阅第41条提示。

S7-STEP 7-

举

**在RS-232设备中使用PC/PPI电缆和自由端口模式**

[返回顶端](#)

您可以使用PC/PPI电缆和自由端口通讯功能将S7-200 CPU与很多与RS-232标准兼容的设备连接。

当数据从RS-232端口传输至RS-485端口时，PC/PPI电缆位于“传输”模式。当电缆闲置或从RS-485端口向RS-232端口传输数据时，电缆位于“接收”模式。一旦检测到RS-232传输线路上的字符时，电缆立即从“接收”模式转换为“传输”模式。

PC/PPI电缆支持1200 baud至115.2

kbaud的波特率。使用PC/PPI电缆外壳上的DIP开关将电缆配置为正确的波特率。下表显示波特率和开关位置。

波特率	周转时间（毫秒）	设置（1 = 向上）
38400至115200	.5	
19200	.0	
9600	.0	
4800	.0	
2400	.0	
1200	.0	

当RS-232传输线路位于闲置状态的时间达到定义的电缆周转时间时，电缆开关返回“接收”模式。

如果您在使用自由端口通讯的系统中使用PC/PPI电缆，S7-200 CPU中的程序必须在以下情形下能够理解周转时间：

- S7-200对RS-232设备传输的讯息作出应答。  
当S7-200从RS-232设备接收到请求讯息时，S7-200必须延迟应答讯息的传输，延迟时间必须大于或等于电缆的周转时间。
- RS-232设备对从S7-200 CPU传输的讯息作出应答。  
当S7-200 CPU从RS-232设备接收应答讯息后，S7-

200必须延迟下一则请求信息的传输，延迟时间必须大于或等于电缆的周转时间。

在上述两种情形下，延迟均允许PC/PPI电缆有足够的时间从“传输”模式转换为“接收”模式，以便从RS-485端口向RS-232端口传输数据。

## 7.7 在网络中使用调制解调器和STEP 7-Micro/WIN

在STEP 7-Micro/WIN和S7-200 PLC之间通讯时有两种不同的调制解调器配置：

- PC调制解调器连接：PC调制解调器连接可以是内装或外装调制解调器（模拟、无线电和手机调制解调器）。此类调制解调器连接被视作本地连接。
- PLC调制解调器连接：PLC调制解调器连接包括将S7-200 PLC与EM 241调制解调器扩充模块或远程调制解调器连接。此类调制解调器连接被视作远程连接。

本标题讨论以下内容：

[为PC调制解调器连接设置本地调制解调器](#)

[为11位通讯配置本地调制解调器](#)

[为PC调制解调器连接配置本地调制解调器](#)

[为外装PC调制解调器连接配置无线电调制解调器](#)

[为外装PC调制解调器连接配置手机调制解调器](#)

[使用EM 241调制解调器扩充模块配置PLC远程调制解调器连接](#)

[将调制解调器与PC/PPI电缆配合使用](#)

**为PC调制解调器连接设置本地调制解调器**

[返回顶端](#)

在开始配置PC调制解调器连接之前，您必须首先通过Windows控制面板设置本地调制解调器。

STEP 7-Micro/WIN

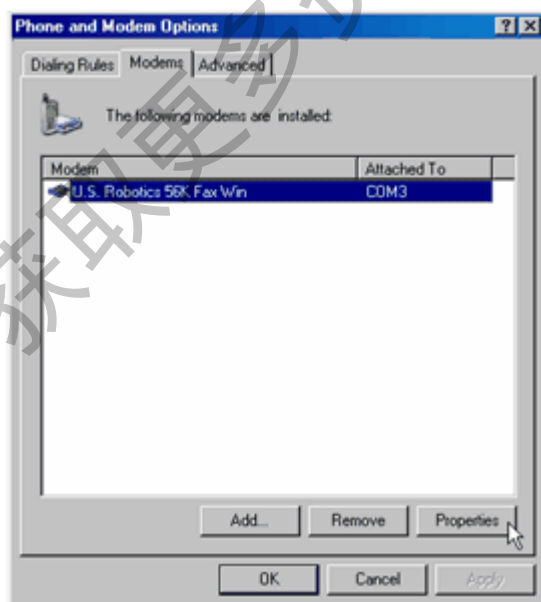
3.2版使用标准Windows‘电话和调制解调器选项’选择和配置电话调制解调器。可在Windows控制面板下方找到调制解调器选项。

将Windows设置选项用于调制解调器允许您：

- 使用Windows支持的大多数内装和外装调制解调器。
- 对Windows支持的大多数调制解调器使用标准配置。
- 使用标准Windows拨号规则选择地点、国家和地区代码支持、脉冲或音调拨号、以及电话卡支持。
- 与EM 241调制解调器模块通讯时，使用较高的波特率。

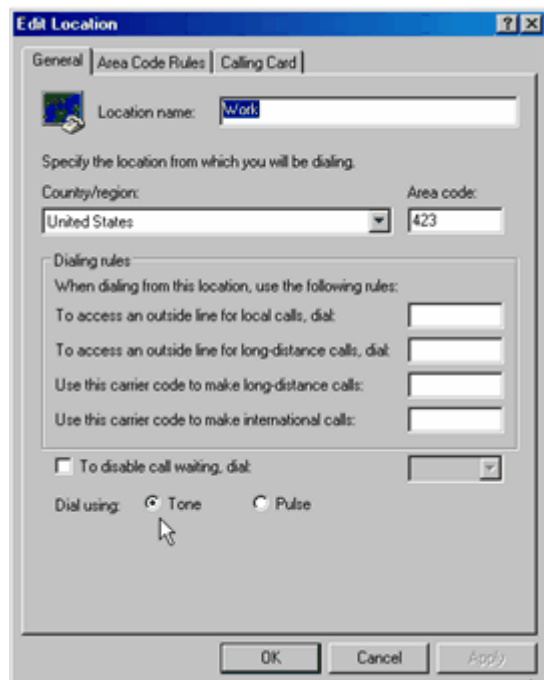
欲存取“调制解调器属性”对话框，执行以下两个步骤之一：

- 选择桌面上的“控制面板”图标，然后单击“调制解调器”。
- 如果是Windows 2000或XP，选择开始 > 设置 > 控制面板 > 电话和调制解调器选项。



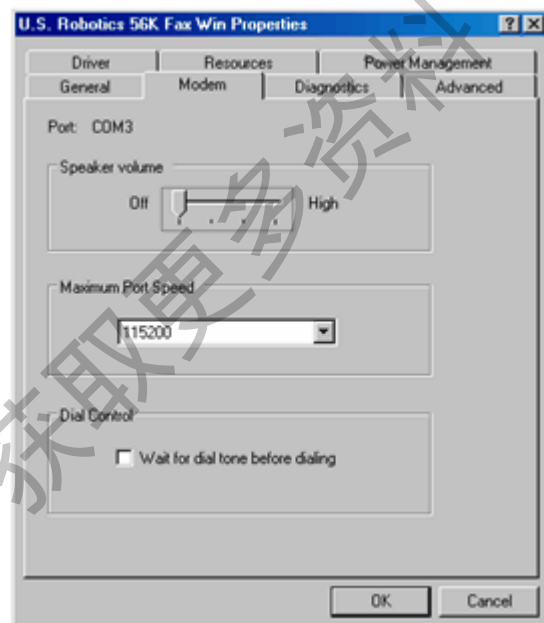
### 电话和调制解调器选项 -

**调制解调器**标记允许您设置本地调制解调器。您可以从Windows支持的调制解调器列表中选择您的调制解调器，或者单击“增0...”按钮，增加另一台调制解调器。如果您的调制解调器未列入Windows调制解调器对话框中，请与您的调制解调器供货商联系，索取Windows的调制解调器配置文件。如果您的调制解调器不支持Windows驱动程序，可使用标准9600 baud。



**编辑位置 - 拨号规则**允许您选择“音调”或“脉冲”拨号。您还可以配置拨号位置，并为每个实际位置指定具体的拨号属性。

单击**调制解调器**标记上的**属性**按钮，为调制解调器配置以下选项：



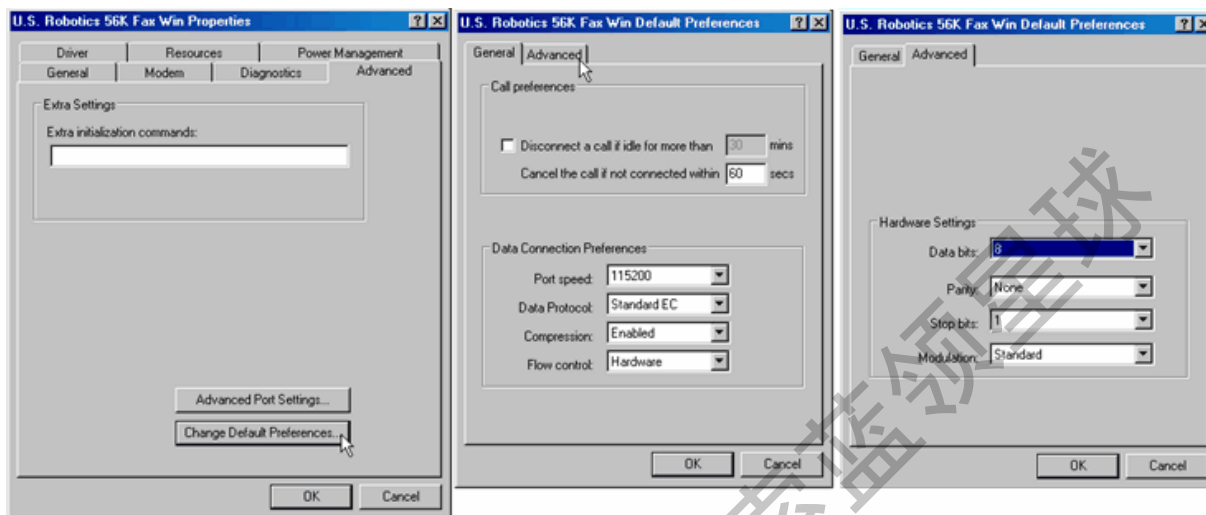
### 调制解调器标记

**注释：**根据正在配置的调制解调器类型，可使用不同的选项。调制解调器支持并通过调制解调器注册表配置识别的功能定义可

Windows波特率和帧选择仅控制PC串行端口的设置。Windows要求调制解调器具有自动设置波特和自动设置奇偶校验的功能。该波特率确定通讯速率。调制解调器被选择时，不使用PG/PC接口中的波特率设置。

用此种方法配置还有助于提供与EM

241 调制解调器模块波特率选项一致的附加波特率选项。如果调制解调器允许用超过 19,200 的波特率操作，可通过该配置获得镜氏髦平獾骸鞅为毕实囊恢卵=瘵



#### 高级连接设置

高级连接属性可通过“高级...”按钮存取。此类选项不要求理解具体调制解调器命令。此类域仅限在调制解调器支持这些功能

其他调制解调器命令可在**额外的初始化命令**域中指定。这些是无须在对话域中理解的具体调制解调器 AT 命令。

#### 为 11 位通讯配置本地调制解调器

##### [返回顶端](#)

几种调制解调器可为 10 位或 11 位通讯提供支持。每种调制解调器的工厂配置默认值为 10 位通讯。欲将本地调制解调器配置为 11 位通讯，必须改动两项配置：

- 首先，必须将具体 11 位调制解调器命令放置在 Windows 调制解调器对话框的“附加设置”域中。例如，MultiTech 调平獾骸鞅 11 位命令为 \$EB1。
- 其次，必须为 Windows 串行端口配置选择偶数校验。一旦本地调制解调器配置为 11 位通讯，通常无法通过 Windows 的 10 位通讯。这是因为调制解调器一旦位于 11 位模式就无法自动执行奇偶校验。将调制解调器重设为 10 位的最简单的方法是将电源循环至调制解调器位置。

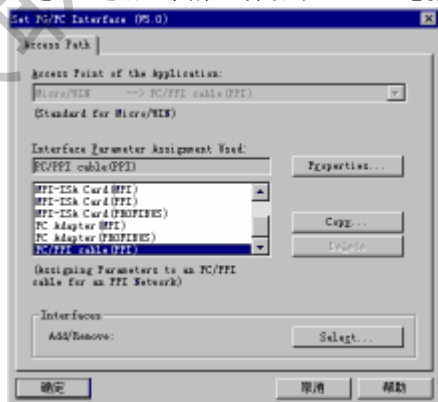
仅将“连接首选项”配置为 11 位操作还不够。还必须通过 AT 命令，命令调制解调器进入 11 位模式。具体 AT 命令根据调制解调器

#### 为 PC 调制解调器连接配置本地调制解调器

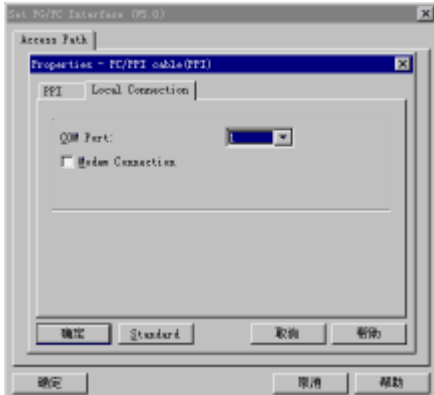
##### [返回顶端](#)

通过 Windows 控制面板设置本地调制解调器后，使用 Micro/WIN 中的“设置 PG/PC 接口”对话框，将调制解调器连接作为通讯

1. 进入“通讯”对话框，并双击“PC/PPI 电缆”，打开 PG/PC 接口。



2. 选择PPI电缆，并单击“属性”方框。



3. 在“本地连接”标记上，选择“调制解调器连接”方框。单击“确定”。  
4. 双击“调制解调器”标记，显示“调制解调器连接”对话框。

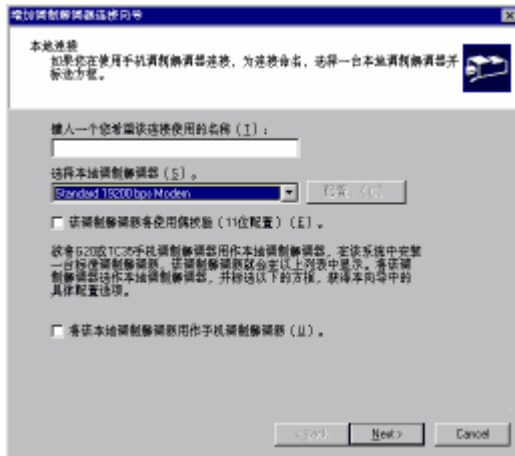


5. 单击“设置”按钮，显示“调制解调器连接设置”对话框。您可以编辑当前使用的调制解调器的设置，或者选择增加一个新配置。

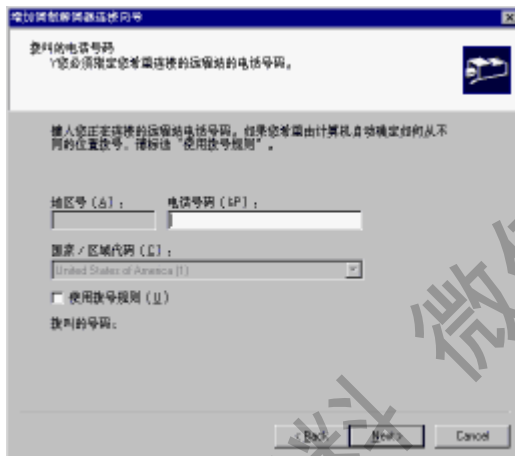


6. 单击“增加”按钮，启动“增加调制解调器连接”向导。连接指识别的名称与 STEP 7-Micro/WIN和特定S7-200 PLC之间通讯实际属性之间的联系。这些属性包括电话号码、超时以及任何其他与连接相关的独特特征。

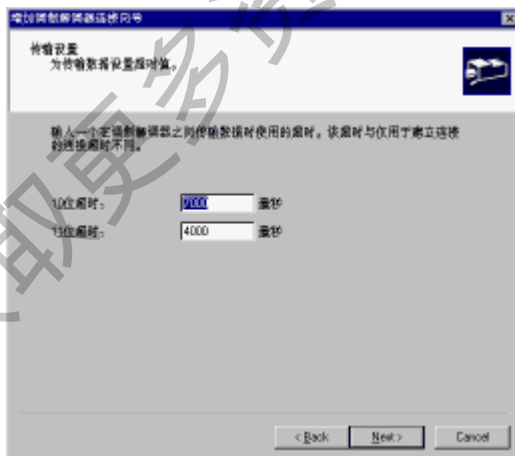




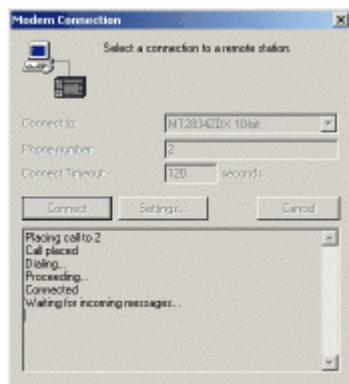
7. 键入连接名称, 并从调制解调器下拉列表中选择一台需要配置的调制解调器。列表包括通过操作系统安装的每台调制解调器和无线电调制解调器。单击“配置..”按钮, 设置调制解调器属性。



8. 输入您正在连接的远程站电话号码。您还可以选择激活拨号规则。



9. 指定STEP 7-Micro/WIN等候PLC认可通讯请求的时间。  
10. 单击“完成”, 结束“调制解调器连接”配置。



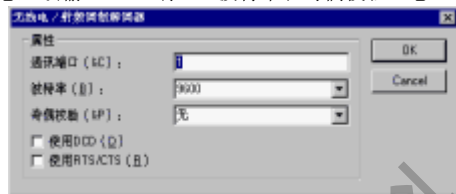
11. 单击“连接”按钮，使用“调制解调器连接”对话框建立调制解调器连接。您可以指定等候建立连接的总时间。默认值为90秒。一旦开始执行，则在位于对话框底部的文字方框中显示详细的拨号状态信息。连接后，在该对话框中编辑的功能被禁用，对话框允许执行的唯一操作是使用“断开”按钮断开或选择“取消”继续操作。

**注释：**可修改任何以前配置的电话号码，但这些更改不随连接一起保存。而是在 STEP 7-Micro/WIN 应用程序设为现用时被用作临时改动。欲将其设为永久性改动，使用“调制解调器连接向导”编辑配置。

### 配置无线电调制解调器

[返回顶端](#)

如果您希望配置无线电调制解调器，则必须从第一个“调制解调器连接向导”屏幕中的下拉列表选择“无线电 / 射频调制解调器”。您还必须输入 COM 端口、波特率和奇偶校验。您还可以指定无线电是使用载波信号还是流量控制信号。



### 配置手机调制解调器

[返回顶端](#)

您可以选择将 M20 或 TC35 手机调制解调器用作本地调制解调器，方法是首先在系统上安装标准 9600 调制解调器驱动程序，并选择“调制解调器连接向导”第一个屏幕中的“将该本地调制解调器用作手机调制解调器”复选框。

将该本地调制解调器用作手机调制解调器 (U)。

在“特殊配置设置”对话框中，必须输入 PIN 命令和 PIN 号码。此类设置允许配置与手机类型调制解调器相关的任何授权命令。在向任何远程调制解调器拨号和建立任何连接之前设置此类设置。

您还必须以秒为单位输入一个超时数值，为调制解调器授权。如果您使用的是 M20 手机，还须输入一个授权后超时数值。接受 PIN 之后，在成功拨号之前，M20 最多可延迟 2 分钟。



欲获得有关M20或TC35手机调制解调器故障排除的信息，请参阅[Siemens M20和TC35手机调制解调器故障排除](#)。

### 配置远程PLC调制解调器连接

#### [返回顶端](#)

如果远程调制解调器是模拟调制解调器或手机调制解调器，启动“调制解调器扩充”向导，并选择配置模拟或手机调制解调器W 裱 虹贾械闹噶睿 柚煤捕噎淘冻痰髦平獾骸颞 糜 S7-200 CPU。

欲获得有关配置远程PLC的详情，请参阅[模拟或手机调制解调器配置](#)。

### 将调制解调器与PC/PPI电缆配合使用

#### [返回顶端](#)

您可以使用PC/PPI电缆将调制解调器的RS-232通讯端口与S7-200 CPU连接。PC/PPI电缆上的1、2和3号开关设置波特率。4号开关选择10位或11位PPI协议。5号开关选择数据通讯装置（DCE或DTE）模式。6号开关（如果有）选择PC/PPI电缆RS-232端口上的RTS信号操作。

SIEMENS		独立的PC/PPI电缆			
PPI	双列直插式开关 #	波特率	开关 #	模式	PC
	115.2-38.4K	000	4	1= 10 位	
	19.2	001	5	0= 11 位	
1	9.6K	010	5	1= DTE	
0	2.4K	100	6	0= DCE	
	1.2K	101	6	1= 用于XMT的RTS	
				0= 始终为RTS	

调制解调器通常使用RS-232控制信号（例如RTS、CTS和DTR）允许计算机控制调制解调器。将调制解调器与PC/PPI电缆配合使用时，您必须将调制解调器设置为DTE模式。当S7-200用调制解调器与STEP 7-Micro/WIN连接时，仅限使用4号开关。否则，将4号开关设为11位模式，以确保其他设备的正常操作。

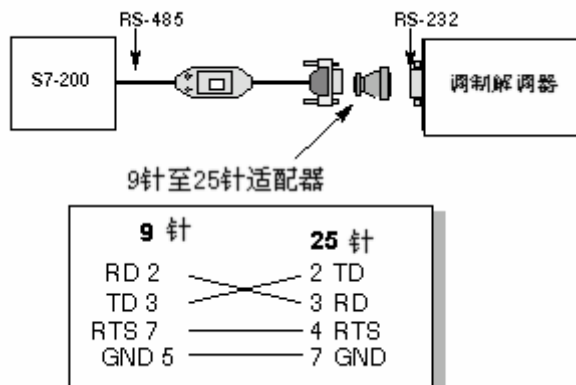
PC/PPI电缆的4号开关选择PPI协议的10位或11位模式。当S7-200用调制解调器与STEP 7-Micro/WIN连接时，仅限使用4号开关。否则，将4号开关设为11位模式，以确保其他设备的正常操作。

PC/PPI电缆的5号开关允许您将电缆的RS-232端口设为DCE或DTE模式。如果您在STEP 7-Micro/WIN上使用PC/PPI电缆或者PC/PPI电缆与计算机连接，将PC/PPI电缆设为DCE模式。如果您将PC/PPI电缆与调制解调器（DCE设备）配合使用，则将PC/PPI电缆设为DTE模式。这样就无须在PC/PPI电缆和调制解调器之间安装一台盖髦平獾骸颞 范勺渲攻髦平獾骸颞 蟪拿 睿 氩樵哪 褂袂木吁必髦平獾骸颞 得魁壮怠 盍髦平獾骸颞 淅淦H【站诤髦平獾骸颞 沱暗慕油罚 贲孔廊恍枰 褂靡惶 9针至25针适配器。

PC/PPI电缆的6号开关选择RS-232接头上的RTS信号操作。当S7-200在RS-485端口上传输数据时，选择“将RTS用于XMT”会将RTS信号设为现用，当S7-200在接收数据时，会将RTS信号设为非现用。选择“始终使用RTS”会将RTS信号设为在PC/PPI电缆的RS-

232端口上始终现用，而无论S7-200是在传输数据还是接收数据。当PC/PPI电缆设为DTE模式时，6号开关仅会影响RTS信号。

下图显示一台共用调制解调器适配器的插针分配状况。



下表显示位于DTE模式的PC/PPI电缆RS-485和RS-232端口的插针号码和功能。表7-10显示位于DCE模式的PC/PPI电缆RS-485和RS-232端口的插针号码和功能。当PC/PPI电缆位于DTE模式时，仅限提供RTS。

#### RS-485接头外接插针

插针	信号说明	插针
1	接地 (RS-485逻辑接地)	
2	V回流 (RS-485逻辑接地)	
3	信号B (RxD/TxD+)	
4	RTS (TTL级别)	
5	接地 (RS-485逻辑接地)	
6	NC (无连接)	
7	V电源	7
8	信号A (RxD/TxD-)	
9	协议选择	9

#### RS-232 DTE接头外接插针\*\*

插针	信号说明	插针
1	未使用: 检测到数据载波 (DCD)	
2	接收数据 (RD) (向PC/PPI电缆输出)	
3	传输数据 (TD) (从PC/PPI电缆输入)	
4	未使用: 数据传输准备就绪 (DTR)	
5	接地 (RS-232逻辑接地)	
6	未使用: 数据集准备就绪 (DSR)	
	请求发送 (RTS) (输出 - 可选择开关)	
8	未使用: 清除以便发送 (CTS)	
	未使用: 环状指示器 (RI)	

\*\*调制解调器可能要求将内孔接头转换为凸形接头，并将9针转换为25针。

#### RS-485接头外接插针

插针	信号说明	插针
1	接地 (RS-485逻辑接地)	
2	V回流 (RS-485逻辑接地)	
3	信号B (RxD/TxD+)	
4	RTS (TTL级别)	
5	接地 (RS-485逻辑接地)	
6	NC (无连接)	
7	V电源	7
8	信号A (RxD/TxD-)	
9	协议选择	9

#### RS-232 DCE接头外接插针

插针	信号说明	插针
1	未使用: 检测到数据载波 (DCD)	
2	接收数据 (RD) (向PC/PPI电缆输出)	
3	传输数据 (TD) (从PC/PPI电缆输入)	
4	未使用: 数据传输准备就绪 (DTR)	
5	接地 (RS-232逻辑接地)	
6	未使用: 数据集准备就绪 (DSR)	
	未使用: 请求发送 (RTS)	
8	未使用: 清除以便发送 (CTS)	
	未使用: 环状指示器 (RI)	

## 7.8 高级标题

本标题讨论以下内容:

- [优化网络性能](#)
- [计算网络的令牌轮转时间](#)
- [比较令牌轮转时间](#)
- [理解链接网络设备的连接](#)
- [复杂网络作业](#)

### 优化网络性能

[返回顶部](#)

下列因素会影响网络性能 (波特率和主设备对网络性能的影响最大):

- **波特率：**接受所有设备支持的最高波特率在操作中网络影响最大。
- **网络上的主设备数目：**尽量减少网络上的主设备数目也可增强网络性能。网络上的每台主设备会增加网络的附加要
- **选择主设备和从属设备地址：**应当设置主设备地址，以便所有的主设备均具有连续的地址，地址间无间隔。每当主
- **间隔升级因子（GUF）：**仅在S7-200 CPU作为PPI主设备操作时使用，GUF通知S7-200检查其他主设备的地址间隔频率。您使用STEP 7-Micro/WIN为CPU端口在CPU配置中设置GUF。这样配置的S7-200仅会定期检查地址间隔。如果GUF=1，在S7-200每持有令牌两次即检查一次地址间隔。如果在主设备之间不存在地址间隔，GUF对性能不会产生任何影响。将GUF设为很大的数目会在将主设备联机时造成长时间的延迟，因为检查地址的频率较低。GUF的默认值为10。
- **最高站址（HAS）：**仅在S7-200 CPU作为PPI主设备操作时使用，HAS定义主设备应当查找另一台主设备的最高地址。您使用STEP 7-Micro/WIN为CPU端口在CPU配置中设置HAS。设置HAS会限制网络中最后一台主设备（最高地址）必须检查的地址间隔。限制地址间隔的大小会将查找另一台主设备和将另一台主设备联机所要求的时间降至最低。最高站址对从属设备地址无任何影响：主设备依然可以与地址大于HAS的从属设备通讯。通常将所有主设备上的最高站址设为相同的数值。该地址应当大于或等于最高主设备地址。HAS的默认值为31。

### 计算网络令牌轮转时间

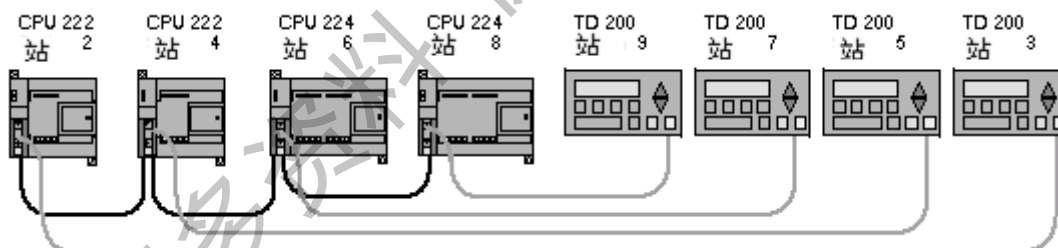
[返回顶端](#)

在令牌传递网络中，唯一可以起始通讯的站是持有令牌的站。令牌轮转时间（令牌轮转至逻辑环中的每台主设备所需的时间）

下图提供一个网络范例，作为计算多台主设备网络令牌轮转时间举例。在该范例中，TD 200（3号站）与CPU 222（2号站）通讯，TD 200（5号站）与CPU 222（4号站）通讯。两个CPU 224模块使用NETR和NETW指令从其他S7-200搜集数据：CPU 224（6号站）向2号、4号和8号站发送讯息，CPU 224（8号站）向2号、4号和6号站发送讯息。在该网络中，共有六个主设备站（四台TD 200装置和两个CPU 224模块）和两个从属设备站（两个CPU 222模块）。

#### 注释：

请参阅文档光盘中的“提示与技巧”一节中有关令牌轮转的说明。请参阅第42条提示。



欲使主设备发送讯息，主设备必须持有令牌。例如：当3号站持有令牌时，向2号站发出一则请求讯息，然后将令牌传递给5号站。5号站接着向4号站发出一则请求讯息，再将令牌传递给6号站。6号站接着向2号、4号或8号站发出一则讯息，并将令牌传递给7号站。这一发出讯息并传递令牌的程序沿着逻辑环继续执行，从3号站至5号、6号、7号、8号、9号站，最后返回3号站。令牌必须沿整个逻辑环轮转，主设备才能发送信息请求。如果是包含6个站的逻辑环，发送一则每个令牌持有读取或写入一个双字节数值（四个字节的数据）的请求讯息，在9600 baud波特率时令牌轮转时间约为900微秒。增加每则讯息存取的数据字节数或站数会延长令牌轮转时间。

令牌轮转时间由每个站持有令牌的时间决定。您可以通过增加每台主设备持有令牌的时间决定多台主设备网络的令牌轮转时间。PPI主设备模式（位于网络的PPI协议下方），您可以向其他S7-200发送讯息，方法是在S7-200上使用“网络读取”和“网络写入”指令。如果您使用这些指令发送讯息，则可以根据以下假设使用下列公式计算令牌大碌穆肿 奔洌好看纬钟辛轧剖泵扛稣痉(7)鸵辉蚯蛄蛄磺蛄第嵌粤 菸恢梅 3.龋亩寥】蝓慈蛄蛄蛄蛄 S7-200中使用一台通讯缓冲器不会发生冲突；不存在扫描时间超过约10微秒的S7-200。

令牌持有时间 (Thold) = (128个附加位 + n个数据字符) × 11位 / 字符 × 1/波特率  
 令牌轮转时间 (Trot) = 主设备1的Thold + 主设备2的Thold + ... + 主设备m的Thold  
 其中 n是数据字符（字节）数目  
 m是主设备数目

以下公式计算上图显示的范例的轮转时间（一“位时间”等于一个信号传递阶段的时间）：

$$T \text{ (令牌持有时间)} = (128 + 4 \text{ 个字符}) \times 11 \text{ 位 / 字符} \times 1/9600 \text{ 位时间/秒}$$

$$T \text{ (令牌轮转时间)} = \frac{\text{每台主设备 } 151.25 \text{ 微秒}}{\text{每台主设备 } 151.25 \text{ 微秒} * 6 \text{ 台主设备}} = 907.5 \text{ 微秒}$$



**提示：** SIMATIC NET COM PROFIBUS软件提供一台分析仪，用于确定网络的性能。

### 比较令牌轮转时间

[返回顶端](#)

下表显示令牌轮转时间与站数、数据量和波特率之比较。时间系基于在 S7-200 CPU或其他主设备上使用“网络读取”（NETR）和“网络写入”（NETW）指令的情况进行计算。

波特率	传输的字节	主设备数目		
9.6 kbaud	2		.30	.44
		.74	.89	.03
		.33	.48	.18
		16	.33	.50
		.83	.99	.16
19.2 kbaud	2	.49	.65	.32
			.15	.22
		.37	.44	.52
		.67	.74	.59
		16	.17	.25
187.5 kbaud	2	.41	.50	.58
		.74	.83	.66
			.009	.013
		7	.022	.026
		5	.039	.043
1	2	16	.011	.016
		.026	.031	.037
		.047	.052	.04

### 理解链接网络设备的连接

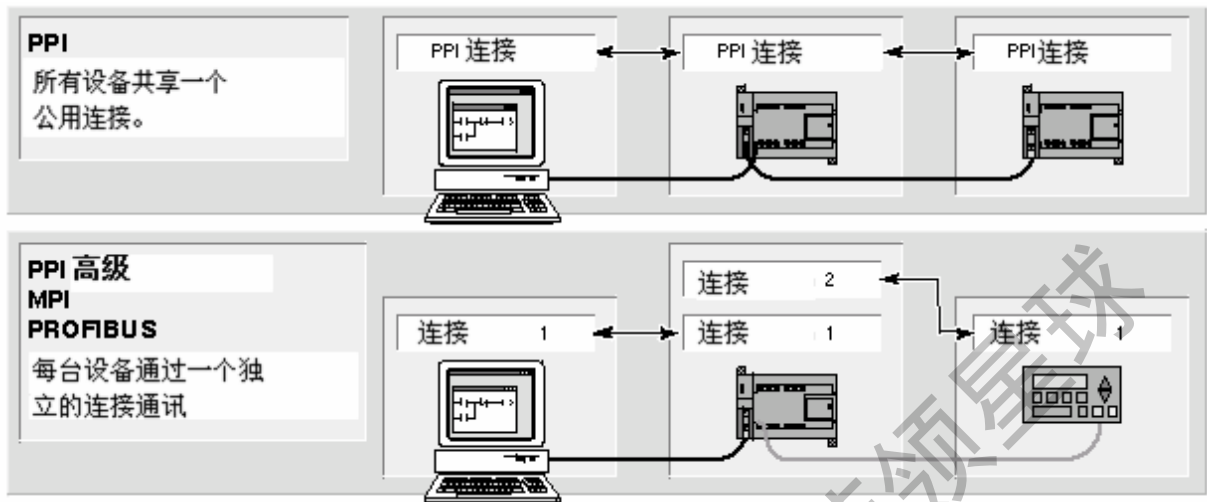
[返回顶端](#)

网络设备通过个别连接进行通讯，这些个别连接是主设备和从属设备之间的专用链接。如下图所示，通讯协议在处理连接的方

- PPI协议在所有网络设备中使用一个共享连接。
- PPI高级协议、MPI和PROFIBUS协议在任何两台相互通讯的设备之间使用独立的连接。
- TCP/IP工业以太网协议使用8个连接。

当使用PPI高级协议、MPI或PROFIBUS时，第二台主设备不得介入已经在一台主设备和一台从属设备之间建立的连接。S7-200 CPU和EM 277始终为STEP 7-

Micro/WIN保留一个连接，为HMI设备保留一个连接。其他主设备不得使用这些保留的连接。这样可确保当主设备在使用支持连接的协议（例如PPI高级协议）时，始终至少将一个编程站和一台HMI设备与S7-200 CPU或EM 277连接。



如上表所示，S7-200 CPU或EM 277提供具体连接数目。S7-200 CPU的每个端口（0号端口和1号端口）最多可支持四个独立的连接。（这样可允许S7-200 CPU最多有八个连接。）但共享PPI连接除外。EM 277支持六个连接。

模块	波特率	连接	协议
S7-200 CPU	0号端口	kbaud	9.6 kbaud、19.2 kbaud或187.5 kbaud
	1号端口		PPI、PPI高级协议、MPI和PROFIBUS **
EM 277模块	每个模块6个	9.6 kbaud至12 Mbaud	PPI高级协议、MPI和PROFIBUS
			TCP/IP以太网
CP243-1以太网模块		每个模块8个	TCP/IP以太网

\*\*对于S7-200 CPU的0号端口和1号端口，与作为从属设备的S7-200通讯时，仅能使用MPI和PROFIBUS。

### 复杂网络作业

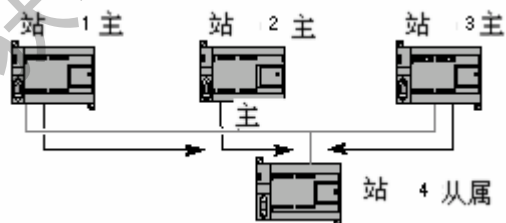
[返回顶端](#)

对于S7-200，复杂网络通常有多台S7-200主设备，使用“网络读取”（NETR）和“网络写入”（NETW）指令与PPI网络上的其他设备通讯。复杂网络通常会发生厥亥收希 璋 魁璞赣氨邮羯璞竿く丁

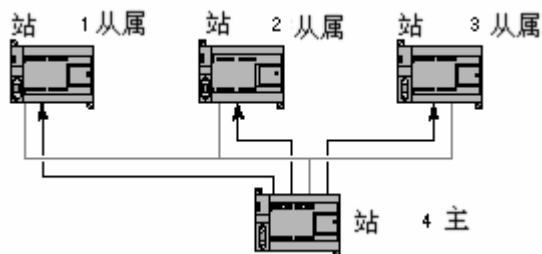
如果网络以较低的波特率（如9.6 kbaud或19.2 kbaud）运行，则每台主设备在传递令牌之前即完成作业（读取或写入）。但是，主设备以187.5 kbaud的速率向从属设备发出一则请求，然后传递令牌，这样就在从属设备中留下一则未完成的请求。

下图显示具有潜在通讯冲突的网络。在该网络中，1号、2号和3号站是主设备，使用NETR或NETW指令与4号站通讯。NETR和NETW指令使用PPI协议，因此所有的CPU可在4号站中共享一个PPI连接。

在该范例中，1号站向4号站发出一则请求。对于超过19.2 kbaud的波特率，1号站则将令牌传递给2号站。如果2号站尝试向4号站发出请求，来自2号站的请求会被拒绝，因为来自1号站那那笠廊准娶淞T 4号站完成对1号站的应答之前，所有向4号站发出的请求均会被拒绝。只有在应答完成后，另一台主设备才能向4号站发出请求。



为了避免在4号站出现此类通讯端口冲突，如下图所示，请考虑将4号站设为网络上的唯一主设备。4号站则向其他CPU发出读。 慈那呐蟆 4 酥峙渲貌唤佳扇繁 2 换岗鱿滞く冻连唬 铨杉跽僇缺嗵口魁璞复 吹母郊右 媳 雇 缙 行V卦俗螯



但是，某些应用程序则无法选择减少网络上的主设备数目。您有几台主设备时，则必须管理令牌轮转时间，并确保网络未超出。如果使令牌返回至主设备所需的时间超过该预定令牌轮转时间，则不允许该主设备发出请求。只有在实际令牌轮转时间小于预定令牌轮转时间时，才允许该主设备发出请求。

CPU的最高站址（HAS）和波特率设置决定预定令牌轮转时间。下表列出了预定的轮转时间。

HSA	.6 kbaud	.2 kbaud	.5 kbaud
HSA=15	.613秒	0.307秒	31微秒
HSA=31	.040秒	0.520秒	53微秒
HSA=63	.890秒	0.950秒	97微秒
HSA=126	.570秒	1.790秒	183微秒

对于较低的波特率，例如9.6 kbaud和19.2

kbaud，主设备在传递令牌之前等待对请求的应答。因为就扫描时间而言，处理请求/应答循环可能会需要相对较长的时间，所以在某些情况下，只在极少的情况下允许主设备处理请求。

例如：如果有一个装有10台主设备的网络，以9.6

kbaud的速率传输1个字节，配备最高站址为15的HAS。在该范例中，每台主设备始终有一则讯息准备发送。如上表所示，该网络预定的轮转时间为0.613秒。但是，根据以上比较令牌轮转时间表中所列的性能数据，该网络要求的实际令牌轮转时间为1.48秒。因为实际令牌轮转时间大于预定令牌轮转时间，在其后的令牌轮转之前不允许某些主设备传输讯息。

在改善实际令牌轮转时间大于预定令牌轮转时间这一情形时您有两个基本的选择：

- 您可以通过降低网络上的主设备的数目，减少实际令牌轮转时间。取决于您使用的应用程序，这可能不是一个可行的解决方案。
- 您可以通过增加网络上的所有主设备的HAS，增加预定令牌轮转时间。

增加HAS可能会为网络带来另一种问题，会影响S7-

200转换至主设备模式和进入网络的时间。如果您使用一台定时器以确保“网络读取”或“网络写入”指令在指定的时间内完成，那么S7-200作为主设备在网络中增加会使指令超时。您可以通过减少网络中所有主设备的间隔更新因子（GUF）尽量减少增加主设备带来的影响。

由于以187.5

kbaud将请求发送和保留在从属设备中的方式不同，在选择预定令牌轮转时间时，您应当留出额外的时间。对于187.5 kbaud的波特率，实际令牌轮转时间应当约为预定令牌轮转时间的一半。

欲确定令牌轮转时间，使用以上比较令牌轮转时间表中的性能数据，确定完成“网络读取”和“网络写入”操作所需的时间。

例如，HMI设备（例如TD 200）所需的时间，使用传输16个字节的性能数据。将网络上的每台设备所需的时间相加，计算令牌轮转时间。将所有时间相加，得到令牌轮转时间。

例如：如果一个网络以9.6 kbaud的速率运行，配备四台TD 200和四台S7-200，每台S7-200每秒向另一台S7-200写入10个字节的数据。利用表7-12计算该网络的具体传输时间：

四台TD 200设备传输16个字节的数据	=	0.66秒
四台S7-200传输10个字节的数据	=	0.63秒
总计令牌轮转时间	=	1.29秒

为了允许该网络有足够的时间在一个令牌轮转期间处理所有的请求，将HAS设为63。选择一个大于最大令牌轮转时间（1.29秒）的HAS（例如1.83秒）可确保每台设备在每次令牌轮转时均可传输数据。

为了帮助改善多台主设备网络的可靠性，您还应当考虑采取以下步骤：

- 改变HMI设备的更新速率，以便在两次更新之间留出更多的时间。例如，将TD 200的更新速率从“尽快”更改为“每秒一次”。
- 将NETR操作或NETW操作组合在一起，减少请求次数（以及处理请求的网络附加要求）。例如，使用读取8个字节囊淮 NETR操作，而不是使用每次读取4个字节的两次NETR操作。处理4个字节的两次请求所需的时间比处理8个字节



囊淮吻呐笏 璧氛奔涑と枚嗜

- 更改CPU主设备的更新速率，防止主设备尝试以比令牌轮转时间更快的速度更新。

## 7.9 以太网通讯

CP243-1以太网模块是用于使S7-200 PLC与工业以太网网络链接的S7-200系列的通讯处理器。

以太网模块包括下列功能：

- 根据TCP/IP和ISO通讯标准进行通讯
- 工厂安装MAC地址
- 与其他S7设备进行对等通讯
- 自动传感全双工或半双工通讯，10 MB和100 MB
- 多个连接（最多为8个）
- 客户机或服务器配置选项
- 数据初始化、重新配置和传输指令（ETHx\_CTRL、ETHx\_CFG、ETHx\_XFR）

在使用以太网模块时，共有三种不同类型的通讯选项：

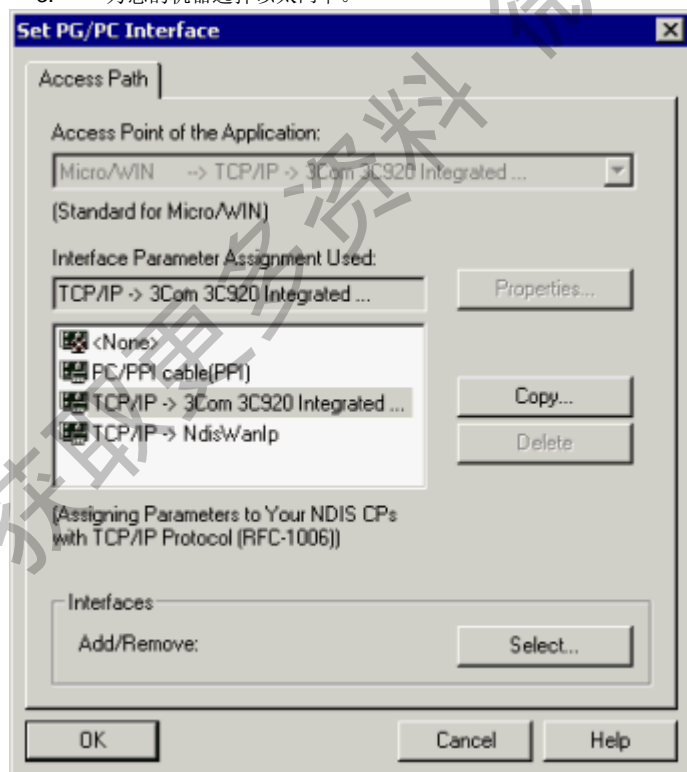
1. 将以太网模块与STEP 7-Micro/WIN PLC连接
2. 将以太网模块与其他S7元件（S7-200/S7-300）连接
3. 将以太网模块与OPC应用程序连接

您一旦使用[以太网向导](#)建立了以太网配置后，则必须通过STEP 7-Micro/WIN设置以太网通讯。

**注释：**如果您使用的是Windows 95 OS，则必须在完成新安装之前单独安装TCP/IP网络功能。详情请参阅自述文件。

请遵循以下步骤设置与TCP/IP以太网网络的通讯：

1. 单击浏览条中的“通讯”图标或选择**检视 > 元件 > 通讯**菜单命令。
2. 双击“通讯设置”窗口中的“存取点”图标。
3. 为您的机器选择以太网卡。



4. 单击“设置PG/PC接口”对话框中的“属性”按钮。
5. 一旦完成配置，单击“设置PG/PC接口”对话框中的“确认”。

6. 在“通讯设置”对话框中，使用“地址浏览器”图标，为对应的CP243-1以太网模块选择或输入IP地址。唯一勾选的IP地址是您已经配置的IP地址。
7. 双击“刷新”图标，尝试与指定的IP地址连接。
  - 如果连接失败，“通讯”窗口中的IP地址会显示为“不存在”。
  - 如果连接成功，但STEP 7-Micro/WIN无法确定PLC类型，IP地址则被显示为“未知”。
  - 如果连接和读取均成功，会在“通讯”窗口中显示适当的PLC图标。
8. 欲核实新连接已设为现用，双击“通讯”窗口中的“PLC”图标。会出现“PLC信息”窗口，CP243-1以太网模块中的CFG LED会打开。



一旦建立了与以太网模块的连接，您可以检视模块的报告方式。  
欲存取该信息：

- 双击“通讯设置”窗口中的模块图标，或
- 选择PLC > 信息。会出现“PLC信息”对话框。双击所列的CP243-1以太网模块条目。

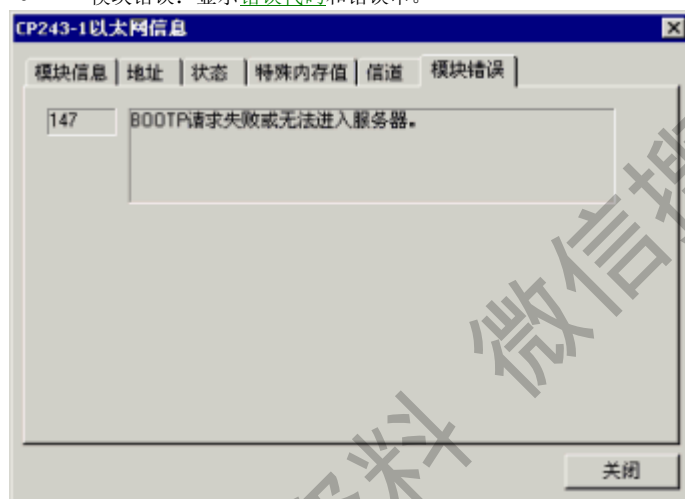


您可以使用以太网模块信息框检视以下信息：

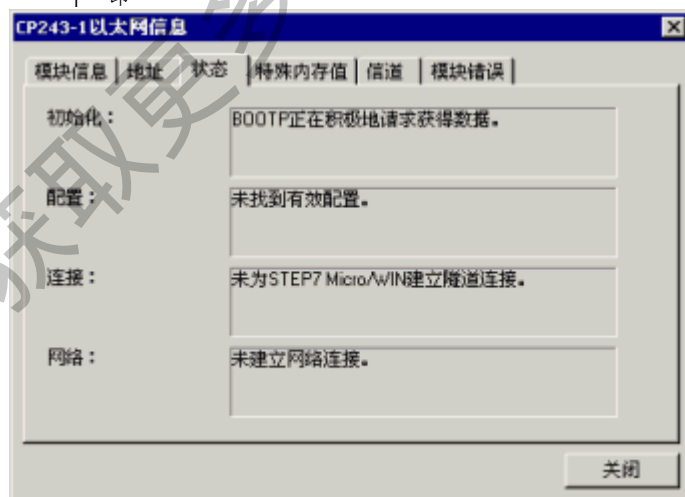
- 模块信息：显示模块类型、固件版本和CP243-1以太网模块硬件修订版。



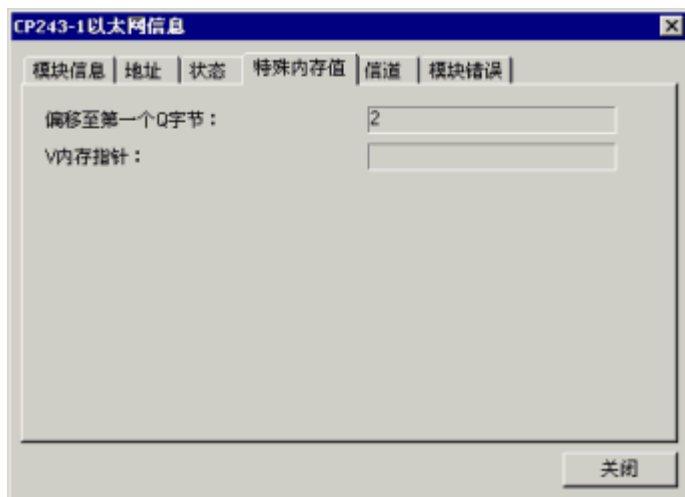
- 模块错误: 显示错误代码和错误串。



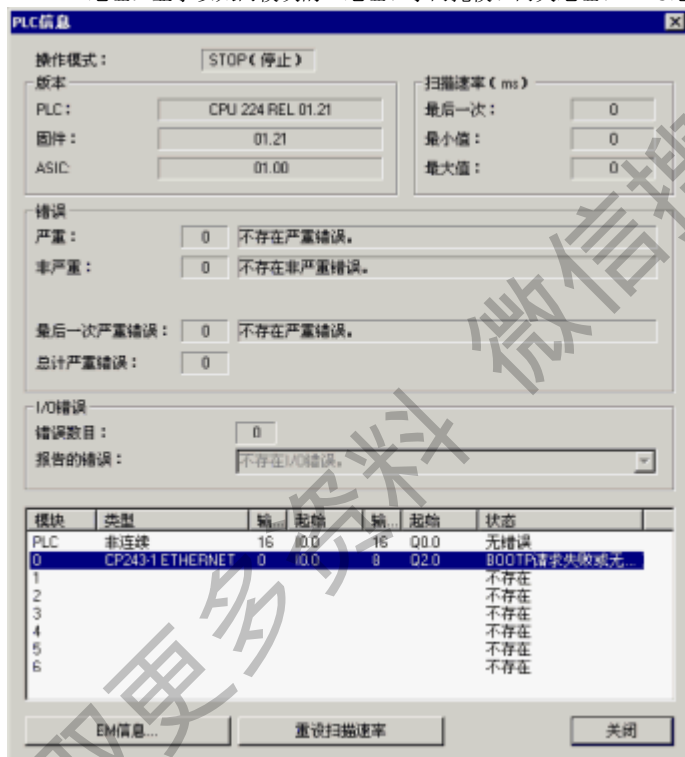
- CP243-1状态: 显示有关不同设备之间模块连接的信息。在第二个域中显示正在使用的以太网配置类型。显示 STEP 7-Micro/WIN与以太网模块的连接是否已在第三个域中建立。在第四个域中显示以太网模块是否已与以太网局域网 (LAN) 连接。



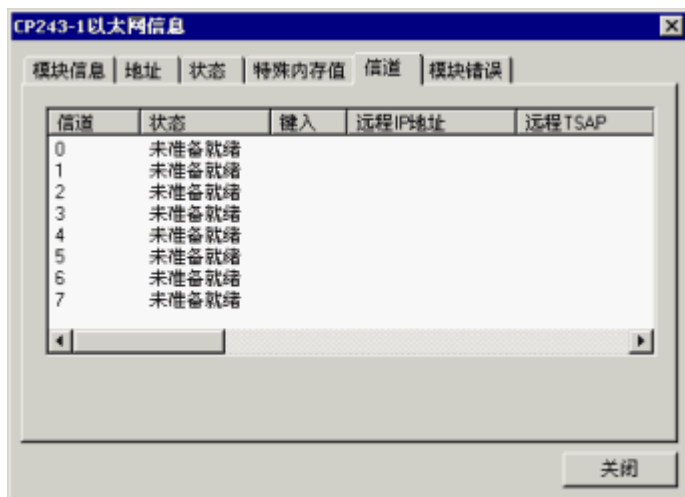
- 特殊内存数值: 显示模块的Q字节地址和指向存储模块配置的数据块地址的V内存指针。



- 地址：显示以太网模块的IP地址、子网掩模、网关地址、MAC地址。



- 信道：显示连接信道、连接站（已配置、未配置、准备就绪、未准备就绪）。连接类型（客户机、服务器）、远程 IP 地址、远程 TSAP 数值、本地 TSAP 数值、“保持现用”被启用或禁用、以及连接的错误状态。请注意，当您使用以太网向导时，须输入所有此类信息。



另请参阅：  
[以太网向导概述](#)  
[以太网向导](#)

获取更多资料 微信搜索 蓝领星球

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part



获取更多资料 微信搜索 蓝领星球

## 8 如何使用PLC内存

### 8.1 内存类型和属性

区域	说明 作为双字存取	作为位存取 可保留	作为字节存取 可强迫[[[ENDKEEPN]]]	作为字存取		
I 写入	离散输入和图象寄存器 否	是	读取/写入	读取/写入	读取/写入	读取/
Q 写入	离散输出和图象寄存器 否	是	读取/写入	读取/写入	读取/写入	读取/
M	内部内存位 是	是	读取/写入	读取/写入	读取/写入	读取/写入
SM SM29为只读内存区)	特殊内存位 (SM0 - 否)	读取/写入	读取/写入	读取/写入	读取/写入	否
V	变量内存 是	读取/写入	读取/写入	读取/写入	读取/写入	是
T 读取/写入	计时器当前值和计时器位 否	T当前 - 是	T位 - 否	T位 读取/写入	否	T当前
C 读取/写入	计数器当前值和计数器位 否	C当前 - 是	C位 - 否	C位 读取/写入	否	C当前
HC 否	高速计数器当前值 否	否	否	否	否	只读
AI	模拟输入 是	否	否	只读	否	否
AQ	模拟输出 是	否	否	只写	否	否
AC 否	累加器寄存器 否	否	读取/写入	读取/写入	读取/写入	读取/写入
L 否	局部变量内存 否	读取/写入	读取/写入	读取/写入	读取/写入	读取/写入
S	SCR 否	读取/写入	读取/写入	读取/写入	读取/写入	否

另请参阅:

[编址概述 \(GS 2.2\)](#)

[直接和间接编址](#)

[CPU内存地址范围](#)

[特殊内存地址](#)

[符号表 / 全局变量表](#)

### 8.2 直接和间接编址

当您写入程序时，您可以使用以下三种模式之一，为指令操作数编址：

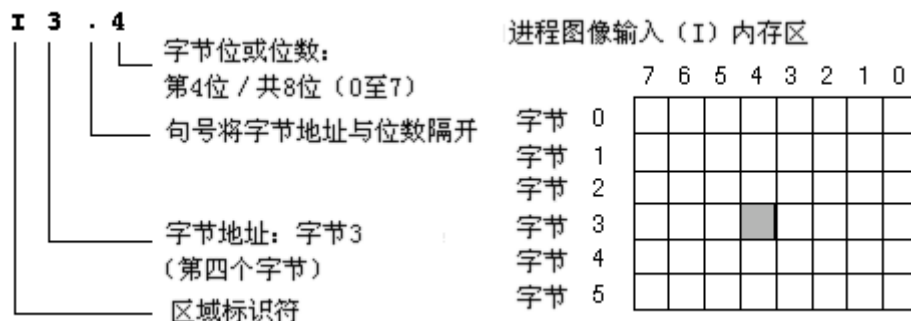
- 直接
- 符号
- 间接

#### 直接编址

S7-

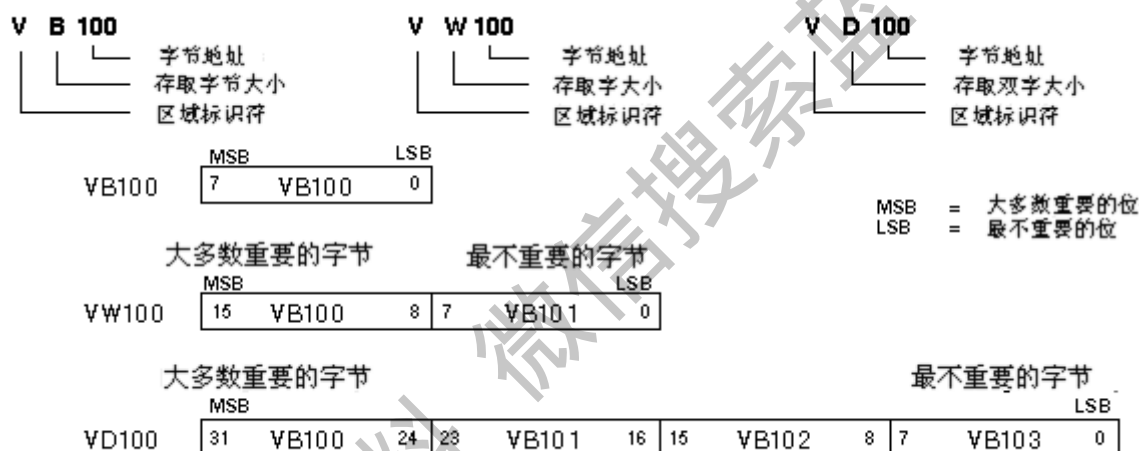
200在具有独特地址的不同内存位置存储信息。您可以明确识别您希望存取的内存地址。这将允许程序直接存取信息。直接编址付 诿媿 4.笨『臀恢茫焕 纾 VW790指内存区中的字位置790。

欲存取内存区中的一个位，您需要指定地址，包括内存区标识符、字节地址和前面带一个句号的位数。下图显示存取位（亦称为“字节位”编址）的一个范例。在该范例中，内存区和字节地址（1=输入，3=字节3）后面是一个句号（“.”），用于分隔位址（位4）。



您可以使用字节地址格式将大多数内存区 (V、I、Q、M、S、L和SM) 的数据存取为字节、字或双字。欲存取内存中数据的字节、字或双字，您必须与指定地址相似的方法指定地址。如下图所示，这包括区域标识符、数据大小指定和字节、字或双字的字节地址。

其他内存区中的数据 (例如，T、C、HC和累加器) 可使用地址格式存取，地址格式包括区域标识符和设备号码。



### 符号编址

符号编址使用字母数字字符组合来识别地址。符号常量使用符号名识别常量或 ASCII 字符值。

对于 SIMATIC 程序，您用符号表进行全局符号赋值。对于 IEC 程序，您使用全局变量表进行全局符号赋值。如果您在 SIMATIC 符

符号编址功能从“检视”菜单控制。名称旁的复选符号表示该功能已打开。否则，所有的地址均仅显示为绝对地址。



**提示：**如果您为局部和全局级别的地址使用相同的名称，局部用法会优先。即如果程序编辑器在局部变量表中发现某一特定程序块的名称定义，则使用该定义。如果未发现定义，程序编辑器会检查符号表。例如，您将 PumpOn 定义为全局符号。您同时也在 SBR2 中（不是 SBR1 中）将其定义为局部变量。编译程序时，在 SBR2 中使用将局部定义用于 PumpOn，在 SBR1 中将全局定义用于 PumpOn。

### 间接编址

间接编址使用指针存取内存中的数据。指针是包含另一个内存位置地址的双字内存位置。您只能将 V 内存位置、L 内存位置或累加器寄存器 (AC1、AC2、AC3) 用作指针。欲建立指针，您必须使用“移动双字”指令，将间接编址内存位置移至指针位置。指

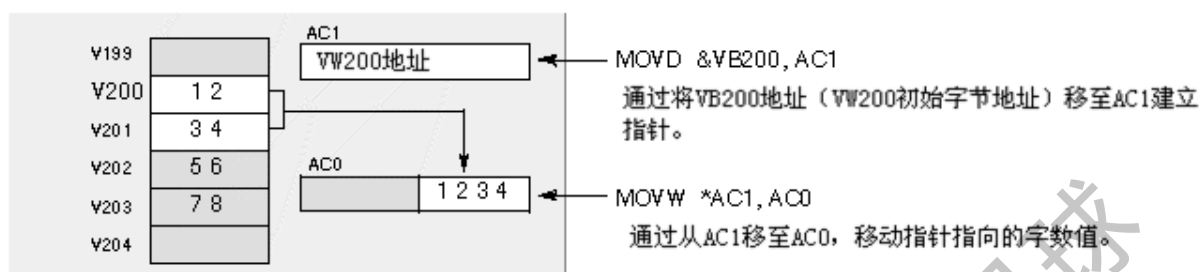
S7-

200 允许指针存取以下内存区：I、Q、V、M、S、T\* (仅限当前值) 和 C (仅限当前值)。您不能使用间接编址存取单个位或存取 AI、AQ、HC、SM 或内存区。

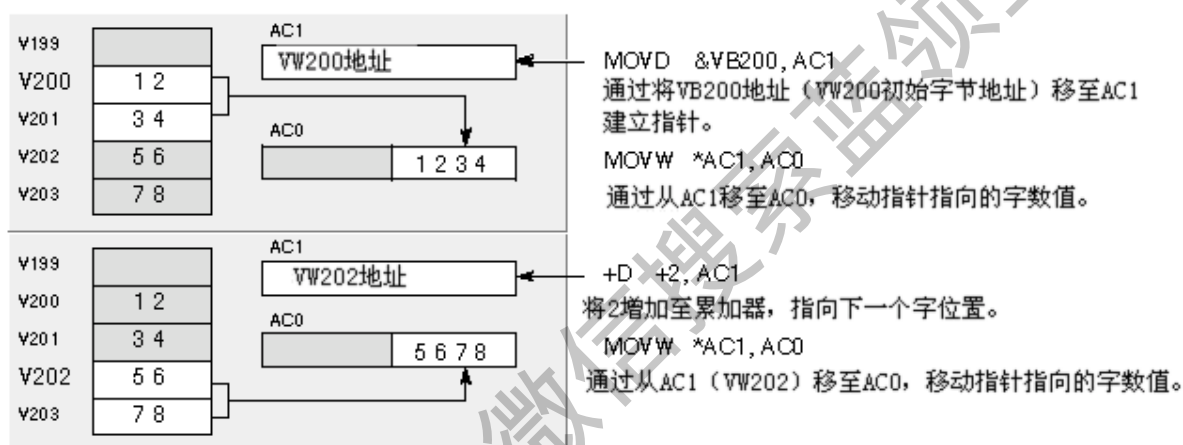
欲间接存取内存区数据，输入一个“和”符号 (&) 和需要编址的内存位置，建立一个该位置的指针。指令的输入操作数前必须幸桓航昂汀狈 (牛 &)，表示内存位置的地址 (而非内存位置的内容) 将被移入在指令输出操作数中识别的位置 (指针)。



在指令操作数前面输入一个星号 (\*) 指定该操作数是一个指针。如下图所示, 输入 \*AC1 指定 AC1 是“移动字” (MOVW) 指令引用的字长度数值的指针。在该范例中, 在 VB200 和 VB201 中存储的数值被移至累加器 AC0。



如下图所示, 您可以改动指针数值。由于指针是 32 位数值, 使用双字指令修改指针数值。可使用简单算术操作 (例如加或递增) 修改指针数值。



**提示:** 欲间接存取字节, 将指针用一递增或递减。欲间接存取字, 将指针用二递增或递减。

欲间接存取双字, 将指针用四递增或递减。如果您从使终止位置超出 V 内存上限的起始位置递增双字, 程序执行时会收到一则运行时间错误讯息。欲存取计时器或计数器当前值 (此为字数值), 将指针用二递增或递减。

另请参阅:

[编址概述 \(GS 2.2\)](#)  
[内存类型和属性](#)  
[CPU 内存地址范围](#)  
[特殊内存地址](#)  
[符号表 / 全局变量表](#)

## 8.3 位存取

欲存取位, 指定位址, 位址包含区域标识符和字节位号。零是所有数据区的第一个地址。字节号后面的小数点用于区别字节号和位号。位址是从 0 至 7 的十进制数字。

例如: I0.0

另请参阅:

[编址概述 \(GS 2.2\)](#)  
[常量 \(数字范围\)](#)  
[内存类型和属性](#)  
[直接和间接编址](#)  
[CPU 内存地址范围](#)  
[特殊内存地址](#)  
[符号表 / 全局变量表](#)

## 8.4 字节、字或双字存取

欲存取字节、字或双字, 指定地址, 地址包含区域标识符, 指定数据大小的字母以及地址号码。

例如: VB200 存取 V 内存地址字节 200 VW200 存取 V 内存地址字节 200 和 201 VD200

存取V内存地址字节200、201、202和203

另请参阅：

[编址概述 \(GS 2.2\)](#)

[常量 \(数字范围\)](#)

[内存类型和属性](#)

[直接和间接编址](#)

[CPU内存地址范围](#)

[特殊内存地址](#)

[符号表 / 全局变量表](#)

## 8.5 CPU内存地址范围

被存取： 226[[[ENDKEEPN]]]	内存类型	CPU 221	CPU 222	CPU 224	CPU
位 (字节.位)	V	0.0 - 2047.7	0.0 - 2047.7	0.0 - 5119.7	V
1.22 0.0 - 8191.7	V 2.00 0.0 - 10239.7 XP	0.0 - 5119.7	V 1.23 0.0 - 10239.7	V 2.00	
I	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	
M	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	0.0 - 31.7	
SM	0.0 - 179.7	0.0 - 299.7	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	
T	0 - 255	0 - 255	0 - 255	0 - 255	
C	0 - 255	0 - 255	0 - 255	0 - 255	
L	0.0 - 59.7	0.0 - 59.7	0.0 - 59.7	0.0 - 59.7	
字节	VB	0 - 2047	0 - 2047	0 - 5119	V 1.22 0 - 8191 V 2.00 0 - 10239
XP	0 25119	V 1.23 0 - 10239	V 2.00		
IB	0 - 15	0 - 15	0 - 15	0 - 15	
QB	0 - 15	0 - 15	0 - 15	0 - 15	
MB	0 - 31	0 - 31	0 - 31	0 - 31	
SMB	0 - 179	0 - 299	0 - 549	0 - 549	
SB	0 - 31	0 - 31	0 - 31	0 - 31	
LB	0 - 59	0 - 59	0 - 59	0 - 59	
AC	0 - 3	0 - 3	0 - 3	0 - 3	
字	VW	0 - 2046	0 - 2046	0 - 5118	V 1.22 0 - 8190 V 2.00 0 - 10238
XP	0 - 5118	V 1.23 0 - 10238	V 2.00		
IW	0 - 14	0 - 14	0 - 14	0 - 14	
QW	0 - 14	0 - 14	0 - 14	0 - 14	
MW	0 - 30	0 - 30	0 - 30	0 - 30	
SMW	0 - 178	0 - 298	0 - 548	0 - 548	
SW	0 - 30	0 - 30	0 - 30	0 - 30	
T	0 - 255	0 - 255	0 - 255	0 - 255	
C	0 - 255	0 - 255	0 - 255	0 - 255	
LW	0 - 58	0 - 58	0 - 58	0 - 58	
AC	0 - 3	0 - 3	0 - 3	0 - 3	
AIW	0 - 30	0 - 30	0 - 62	0 - 62	
AQW	0 - 30	0 - 30	0 - 62	0 - 62	
双字	VD	0 - 2044	0 - 2044	0 - 5116	V 1.22 0 - 8188 V 2.00 0 - 10236
XP	0 - 5116	V 1.23 0 - 10236	V 2.00		
ID	0 - 12	0 - 12	0 - 12	0 - 12	
QD	0 - 12	0 - 12	0 - 12	0 - 12	
MD	0 - 28	0 - 28	0 - 28	0 - 28	
SMD	0 - 176	0 - 296	0 - 546	0 - 546	
SD	0 - 28	0 - 28	0 - 28	0 - 28	
LD	0 - 56	0 - 56	0 - 56	0 - 56	

AC	0 - 3	0 - 3	0 - 3	0 - 3
HC	0 - 5	0 - 5	0 - 5	0 - 5

另请参阅:

[编址概述 \(GS 2.2\)](#)  
[内存类型和属性](#)  
[直接和间接编址](#)  
[特殊内存地址](#)  
[符号表 / 全局变量表](#)

## 8.6 SIMATIC和IEC 1131-3数据类型

在STEP 7-Micro/WIN中编程时, 在下列两种情况下您必须理解并使用数据类型:

- 如果您以IEC 1131-3模式编程, 则必须为全局符号表中的所有数值选择数据类型。
- 如果您在局部变量表中赋值, 则必须为每个局部变量定义数据类型。

当您明确地为数值指定数据类型时, 您为STEP 7-

Micro/WIN提供清晰的指令, 指明需要为数值分配多少内存空间(例如, 数值100可被存储为字节、字或双字)以及如何表示数值(将0视作布尔值还是数字值?)。

每个SIMATIC和IEC 1131-

3指令或参数化子例行程序均由精确定义识别。该定义可被称为签名。对于所有的标准指令, 每个指令操作数允许使用的数据类型集均从签名获得。对于参数化子例行程序, 用户通过局部变量表建立子例行程序签名。

STEP 7-Micro/WIN V3.0编程软件为SIMATIC模式执行简单的数据类型检查, 为IEC 1131-3模式执行细致的数据类型检查。这意味着为局部或全局变量指定一种数据类型时, 软件在下表所列的每个等级中确保操作数数据类型正确。

基本数据类型	数据类型大小	说明	范围
布尔	1位	布尔	0至1
字节	8位	不带符号的字节	0至255
字节	8位	带符号的字节 (SIMATIC模式仅限于SHRB指令)	-
128至+127			
字	16位	不带符号的整数	0至65,535
整数	16位	带符号的整数	-32768至+32767
双字	32位	不带符号的双整数	0至4294967295
双整数	32位	带符号的双整数	-2147483648至+2147483647
实数	32位	IEEE 32位浮点	+1.175495E-38至+3.402823E+38 -1.175495E-38至-3.402823E+38
字符串	2至255字节	ASCII字符串照原样存储在PLC内存中, 形式为1字符串长度接ASCII数据字节	
	ASCII字符代码128至255		
复杂数据类型	说明		范围
TON	打开延迟计时器	1 ms 10 ms 100 ms	T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255
TOF	关闭延迟计时器	1 ms 10 msms	T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255
TP	脉冲计时器 (参阅注释1)	1 ms 10 msms	T32, T96 T33 - T36, T97 - T100 T37 - T63, T101 - T255
CTU	向上计数器		至255
CTD	向下计数器		至255
CTUD	向上 / 向下计数器		至255
SR	设置主要双稳态		无关
RS	重设主要双稳态		无关

### STEP 7-Micro/WIN 提供三级数据类型检查

1. **IEC编辑器具有细致数据类型检查。** 在该模式中, 参数数据类型必须与符号或变量的数据类型完全匹配。欲支持细致数据类型检查, 在细致数据类型检查中, 只有被指定数据类型的数据类型的变量才能成功编译。使用细致数据类型检查时, 数据类型为整数的变量对字指令参数无效。细致数据类型检查仅在IEC 1131-3模式中进行。

用户选择的数据类型                      等同数据类型

布尔	布尔
字节	字节
字	字
整数	整数
双字	双字
双整数	双整数
实数	实数
字符串	字符串

2. **处于SIMATIC模式中的局部变量表具有简单数据类型检查。** 在该模式中，当符号或变量被指定一个数据类型时，同时自动指定与所选数据类型位大小相匹配的所有数据类型。例如，如果用户选择双整数作为数据类型，局部变量会自动指定数据类型双字，因为二者均为32位数据类型。不自动指定实数数据类型，虽然实数也是32位数据类型。实数数据类型被定义为无其他等同数据类型，总是独特的类型。使用局部变量时，只在SIMATIC模式中执行简单数据类型检查。

#### 用户选择的数据类型                      等同数据类型

布尔	布尔
字节	字节
字	字、整数
整数	字、整数
双字	双字、双整数
双整数	双字、双整数
实数	实数
字符串	字符串

3. **SIMATIC符号编辑器无数据类型检查。** 该模式仅限于无法选择数据类型的SIMATIC全局变量。在该模式中，所有相同大小的数据类型被自动指定给符号。例如，编程软件将自动为赋予VD100地址的符号指定下列数据类型：双字、双整数和实数。

#### 为SIMATIC全局符号由大小决定的数据类型

用户选择的地址	指定等同的数据类型
V0.0	布尔
VB0	字节、字符串
VW0	字、整数
VDO	双字、双整数、实数

#### 数据类型检查的益处

数据类型检查的益处是帮助用户避免常见编程错误。例如，如果指令支持带符号的数字，则在使用不带符号的数字作为指令操作数时，软件为该用法设置旗标会有益处。例如，关系比较<I是带符号的指令，显然，对于带符号的数据类型操作数，-1小于0。但当<I指令允许支持不带符号的数据类型时，程序员需确保绝不会发生下列情况。在程序运行时，对于<I指令，不带符号诺 40,000实际上小于0。程序员需确保对带符号的指令使用不带符号的数字不会超越正负数界限，否则会出现无法预测的结果。



**警告** 您应当确保带符号的数字用作不带符号的指令时不得超出正负数界限。若无法确保不带符号的数字用作带符号的指令时不超出正负数界限，可能致使程序或控制器操作出现无法预测的结果。无法预测的控制器操作可能导致人员伤亡或严重伤害及/或财产严重损坏。请务必核实不带符号的数字用作带符号的指令时不得超出正负数界限。

总之，在IEC 1131-3编辑模式中，细致数据类型检查通过对指令的非法数据类型生成错误，帮助程序员在编译过程中发现这些错误。SIMATIC编辑器不具有此一功能。

#### 在SIMATIC与IEC1131-3之间移植程序

因为IEC1131-3属于细致数据类型，而SIMATIC不属于此一类型，STEP 7-Micro/WIN不提供在两种不同编辑模式之间移动程序的能力。程序员必须选择一种希望使用的编辑模式。

#### 超载指令

超载指令支持一系列数据类型。仍然应用细致数据类型检查，在成功编译指令之前所有的操作数数据类型必须匹配。例如，请参阅下表中用于IEC超载ADD（加）指令。

指令 类型检查)	允许使用的数据类型（细致 数据类型检查） 编译指令	允许使用的数据类型（数据
-------------	------------------------------	--------------

ADD	整数	字、整数	ADD_I (加整数)
ADD	双整数	双字、双整数	ADD_D (加双整数)
ADD	实数	实数	ADD_R (加实数)

当所有操作数的数据类型均为双整数时，编译器生成“加双整数”指令。如果超载指令数据类型混合，会出现编译错误。根据数据类型检查的等级决定哪些为非法。在下例中，如果使用细致数据类型检查，则生成编译程序错误，但如果使用简单数据类型检查则成功编译。

ADD IN1 = INT, IN2 = WORD, IN3 = INT.

细致数据类型检查：引起编译错误。

简单数据类型检查：成功编译至ADD\_I (加整数)。

简单数据类型检查不会阻止出现常见运行时间编程错误。例如，使用简单数据类型检查时，编译程序不能帮助避免在运行时间(5) 韵鲁< 暗檀碧媳 ADD 40000,1被解释为负数，而不是不带符号的40,001。

### 在IEC中为超载指令使用直接编址

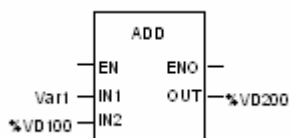
#### IEC-

1131编程模式也允许使用直接表示的内存位置，作为指令参数配置的一部分。参数中可使用变量和内存位置。但是，使用直接表示的内存位置时必须清楚地理解数据类型的隐含意义，因为这些位置不包含明确的类型信息。另外，不能从任何超载 IEC指

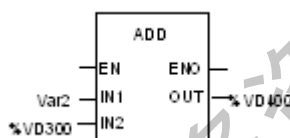
直接表示的参数数据类型通过检查指令中包含的其他类型的参数来决定。当指令参数配置使用某一指定类型的变量时，则假定

名称	地址	数据类型	注解
Var1		实数	这是浮点数变量。
Var2		双整数	这是双整数变量。
Var3		整数	这是整数变量。

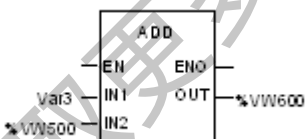
#### 举例 说明



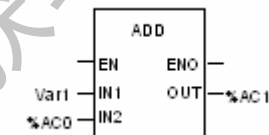
假定VD100和VD200为实数类型，因为Var1是实数类型。



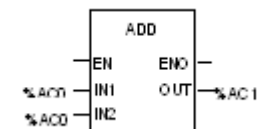
VD300和VD400为双整数类型，因为Var2是双整数类型。



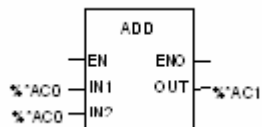
假定VW500和VW600为整数类型，因为Var3是整数类型。



假定AC0和AC1为实数类型，因为Var1是实数类型。



这种配置被视为非法，因为无法确定类型。累加器中的数据类型可为双整数或实数。



数或实数。

这种配置被视为非法，因为无法确定类型。累加器指针中的数据类型可为双整

### 使用数据类型转换指令

转换指令允许从一种数据类型移至另一种数据类型。STEP 7-Micro/WIN支持在下列简单数据类型之间的移动数值转换指令。

#### 转换指令

#### IEC细致数据类型检查 允许的操作数

#### SIMATIC局部变量表中的简单数据类型检查

#### 允许的操作数

字节至整数	入： 字节 出： 整数	入： 字节 出： 字、整数
整数至字节	入： 整数 出： 字节	入： 字、整数 出： 字节
整数至双整数	入： 整数 出： 双整数	入： 字、整数 出： 双字、双整数
双整数至整数	入： 双整数 出： 整数	入： 双字、双整数 出： 字、整数
双整数至实数	入： 双整数 出： 实数	入： 双字、双整数 出： 实数
实数至双整数（进位）	入： 实数 出： 双整数	入： 实数 出： 双字、双整数
字符串至整数	入： 字符串 出： 双字（*指向字符串的指针） 出： 整数	入： 字符串，
双字（*指向字符串的指针）出： 字、整数		
整数至字符串	入： 整数 出： 用双字代表的字符串（*指向字符串的指针）	入：
字、整数 出： 用双字代表的字符串（*指向字符串的指针）		
字符串至双整数	入： 字符串 出： 双字（*指向字符串的指针） 出： 双整数	入： 字符串，
双字（*指向字符串的指针）出： 双字、双整数		
双整数至字符串	入： 双整数 出： 用双字代表的字符串（*指向字符串的指针）	入：
双字、双整数 出： 用双字代表的字符串（*指向字符串的指针）		
字符串至实数	入： 字符串 出： 双字（*指向字符串的指针） 出： 实数	入： 字符串
双字（*指向字符串的指针）出： 实数		
实数至字符串	入： 实数 出： 用双字代表的字符串（*指向字符串的指针）	入： 实数
出： 用双字代表的字符串（*指向字符串的指针）		

在IEC 1131-

3编辑模式中，您可以使用超载移动指令在整数和字、双整数和双字之间转换。超载移动指令允许相同大小的数据类型自由移动

簧 杀喊氮砒蟆

#### IEC 1131-3超载移动

	入	出
移动（整数至字）	整数	字
移动（字至整数）	字	整数
移动（双整数至双字）	双整数	双字
移动（双字至双整数）	双字	双整数

## 8.7 系统电源中断后的数据保留

使用下列方法之一检视系统块，设置CPU选项：

- 单击浏览条中的“系统块”按钮。
- 选择检视（View）> 元件（Components）> 系统块（System Block）菜单命令。
- 单击指令树中的“系统块”图标。

然后单击“系统块”树的保留范围分支。

选择在电源循环时您希望保留的内存区。为V、M、T或C内存输入新值，然后将这些修改下载到CPU。



**提示：**保留范围赋值的数字地址限制随CPU型号和版本不同而异。使用PLC > 类型菜单命令为项目设置目标CPU型号，启用错误检查，并使用默认按钮载入的数值与目标CPU的实际数值保持一致。

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

在默认情况下，所有的V、M、T和C内存均被设为**保留**。您可以重新定义保留范围，将一些内存设为**非保留**。这些赋值配置断电时仍保持有效。

### CPU电源中断后的数据保留

您最多可定义六个保留范围，选择您希望在电源循环过程中保留的内存区。您可将下列内存区中的地址范围定义为保留：V、M、C和T。对于计时器，只能保留保留计时器（TONR），而且只有计时器和计数器的当前值可定义为保留。每次通电时计时器和计数器位均被清除。

M内存的前14个字节的默认设置是非保留。

CPU具有超级电容，可在CPU断电后保存RAM数据。有些CPU型号支持延长可保留RAM数据时间的选用电池盒。电池盒只有在超级电容完全放电后才提供电源。

**断电时：**M内存的前14个字节（MB0至MB13），如果被配置为保留，在CPU模块失去电源时被永久性保存在EEPROM中。

**通电时：**CPU检查RAM内存，核实超级电容或电池已成功地将数据保存在RAM中的数据。如果RAM数据被成功保存，RAM内存的保留区不变。永久V内存（在EEPROM中）的相应区域被复制至CPU RAM中的非保留区。用户程序和CPU配置也从EEPROM恢复。CPU RAM的所有其他非保留区均被设为零。

**通电时：**如果未保存RAM的内容（例如长时间断电后），CPU清除RAM（包括保留和非保留范围），并为通电后的首次扫描设置默认值（如SM 0.2）。用户程序和CPU配置然后从EEPROM（E<sup>2</sup>）复制至CPU RAM。此外，EEPROM中的V内存永久区域和M内存永久区域（如果被定义为保留）从EEPROM复制至CPU RAM。CPU RAM的所有其他区域均被设为零。

### S7-200 CPU内存地址限制

CPU

XP 版本	.10 全部	.91 全部	0.91 .00 全部	.92 .01 全部	.00 .10 至	.01 .11
1.22 上限	2.00	2.00	全部			
V	1023.7 ü		ü	ü	ü	ü
V	2047.7		ü	ü		
V	4096.7	ü	ü	ü	ü	ü
V	5119.7	ü			ü	
V	8191.7					ü
V	10239.7					

VE	127.7						
VE	199.7						
VE	511.7						
VE	1023.7						
VE	2047.7						
VE	5119.7						
VE	8191.7						
VE	10239.7						
CPU							
版本	全部	全部	全部	全部	全部	全部	全部
M	6(W)						
M	15.7						
M	31.7						
CPU							
版本	全部	全部	全部	全部	全部	全部	全部



T	4(W)	ü					
T	64(W)		ü				
T	128(W)			ü			
T	256(W)				ü		ü
C	4(W)	ü					
C	64(W)		ü				
C	128(W)			ü			
C	256(W)	ü	ü	ü	ü	ü	ü

另请参阅:

[系统块](#)

[应用程序用户参考手册](#)

[入门指南目录](#)

## 8.8 常量

### 常数数值范围

数据大小:	不带符号的整数范围		带符号的整数范围		十六进制数字:	十进制数字:	十六
	十进制数字:	十六进制数字:	十进制数字:	十六进制数字:			
B (字节)	0至255	0至FF	-128至+127			80至7F	
W (字)	0至65535	0至FFFF	-32768至+32767			8000至7FFF	
D (双字)	0至4294967295	0至FFFFFF	0至FFFFFF			-2147483648至	
+2147483647	8000 0000至 7FFF FFFF						
数据大小:	十进制实数 (正数)		十进制实数 (负数)				
D (双字)	+1.175495E-38至+3.402823E+38		-1.175495E-38至-3.402823E+38				

### 常量格式标识符

程序可在许多指令中使用字节、字或双字大小的数据常量。格式标识符控制如何显示常量数值（二进制数字、十进制数字、十进制实数、十六进制数字、ASCII字符串）。

假设程序常量为十进制数字，除非您使用格式标识符：

**2#** 二进制数字  
**16#** 十六进制数字

'abcd'

ASCII常量，使用单引号字符起始与结束ASCII字符。对在参数列表中指定常量的绝大多数指令有效。作为数据字节存储。

"string"

ASCII常量字符串，使用双引号字符起始与结束ASCII字符。对在参数列表中指定常量字符串的绝大多数指令有效。作为长度字节后接数据字节存储。

### 二进制常量举例

数字基数	分隔符	常量数值
举例: 2#1101	2 #	1101
举例: 2#1101_1111	2 #	11011111

### 十六进制常量举例

数字基数	分隔符	常量数值
举例: 16#3FB2	16 #	3FB2
16#A_B_C_D	<b>注释:</b> 可使用底线，以便增强可读性	

### ASCII常量字符范围

ASCII常量字符的有效范围为ASCII 32至ASCII 255，但不包括DEL (删除) 字符、单引号字符、双引号字符。在此范围之外的ASCII字符必须使用特殊\$字符格式。

**ASCII原义字符常量举例 (单引号格式)**

与允许将常量作为输入的指令合用

参数	操作数	数据类型	内存图		
IN1, IN2	IB, QB, MB, SMB, VB, SB, LB, AC, 常量, *VD, *LD, *AC	字节	无单引号格式的前导长度字节		
<b>数据</b> VB1	ASCII常量输入的编辑器支持 VB2	VB3	有效地址举例 VB4	VB5	VB0
'A'	程序和数据块	VB0	A		
'AB'	程序和数据块	VB0, VW0	A		B
'ABC'	数据块	VB0	A	B	C
'ABCD'	程序和数据块	VB0, VD0	A		B
C	D				
'ABCDE'	数据块	VB0	A	B	C
D	E				
'ABCDEF'	数据块	VB0	A		B
C	D	E	F		

**ASCII原义常量字符串举例 (双引号格式)**

与允许将常量字符串作为输入的指令合用

参数	操作数	数据类型	内存图		
IN	VB, 常量字符串, *VD, *LD, *AC	字符串	双引号格式所用的前导长度字节		
<b>数据</b> VB1	ASCII常量字符串输入的编辑器支持 VB2	VB3	有效地址举例 VB4	VB5	VB0 VB6
"A"	程序和数据块	VB0	1		A
"AB"	程序和数据块	VB0	2		A
B					
"ABC"	程序和数据块	VB0	3		A
B	C				
"ABCD"	程序和数据块	VB0	4		A
B	C	D			
"ABCDE"	程序和数据块	VB0	5		A
B	C	D	E		
"ABCDEF"	程序和数据块	VB0	6		A
B	C	D	E	F	



提示:就ASCII常量而言, \$是一个格式字符, 用于在字符串中指明某特殊字符。

**使用\$格式字符的字符串示例:**

当您下载或上载数据块及主程序、子例行程序和中断例行程序时, 使用\$格式字符修饰ASCII原义常量。

**字符串项目****存储在PLC内存中的数据**

```
'Cost $$50'    "Cost $$50"    Cost $50
'Enter '$name$' "Enter $"name$""    Enter 'name' Enter "name"
```

下表列举嵌套控制字符的技巧。

代码	解释
\$\$	单美元符号
'	单引号字符
"	双引号字符
SL 或 \$I	换行字符

\$N 或 \$n	新行字符
\$P 或 \$p	进纸, 新页
\$R 或 \$r	回车字符
\$T 或 \$t	制表符, 即制表字符
\$1f	\$ 指定ASCII代码1f后的两个十六进制数字。ASCII 1f 十六进制数字 = ASCII 31 十进制数字

**另请参阅:**[编址概述 \(GS 2.2\)](#)[内存类型和属性](#)[直接和间接编址](#)[CPU内存地址范围](#)[特殊内存地址](#)[符号表 / 全局变量表](#)

## 8.9 关键字

这些关键字是ASCII程序文件的保留字, 不能用作符号名或POU名称。

**注释:**

下列所有关键字均用大写字母标记, 但不区分大小写, 可键入小写、大写或大小写混合字母。

ANY (任何) (不支持? 保留作为将来用)  
 ARRAY (阵列) (不支持? 保留作为将来用)  
 BEGIN (开始)  
 BOOL (布尔)  
 BYTE (字节)  
 CHAR (字符)  
 DATA\_BLOCK\_TAB  
 DATE (日期) (不支持? 保留作为将来用)  
 DATE\_AND\_TIME (日期\_和\_时间) (不支持? 保留作为将来用)  
 DINT (双整数)  
 DT (不支持? 保留作为将来用)  
 DWORD (双字)  
 END\_DATA\_BLOCK\_TAB  
 END\_FUNCTION (结束\_功能)  
 END\_FUNCTION\_BLOCK (结束\_功能\_块)  
 END\_ORGANIZATION\_BLOCK (结束\_组织\_块)  
 END\_STRUCT (结束\_结构)  
 END\_TYPE (结束\_类型)  
 END\_VAR (结束\_变量)  
 FALSE (假)  
 FUNCTION (功能)  
 FUNCTION\_BLOCK (功能\_块)  
 INT (整数)  
 NETWORK (网络)  
 OF (不支持? 保留作为将来用)  
 OFF (关闭)  
 ON (打开)  
 ORGANIZATION\_BLOCK (组织\_块)  
 REAL (实数)  
 STRING (字符串)  
 STRUCT (结构) (不支持? 保留作为将来用)  
 TIME (时间) (不支持? 保留作为将来用)  
 TIME\_OF\_DAY (当日时间) (不支持? 保留作为将来用)  
 TITLE (标题)  
 TOD (不支持? 保留作为将来用)  
 TRUE (真)  
 UDT (不支持? 保留作为将来用)  
 VAR (变量)  
 VAR\_IN\_OUT (变量\_入\_出)  
 VAR\_INPUT (变量\_输入)  
 VAR\_OUTPUT (变量\_输出)  
 WORD (字)  
 DB  
 FC (不支持? 保留作为将来用)  
 FB (不支持? 保留作为将来用)  
 SFB  
 OB  
 SDB

SBR  
 SUBROUTINE\_BLOCK (子例行程序\_块)  
 END\_SUBROUTINE\_BLOCK (结束\_子例行程序\_块)  
 INTERRUPT\_BLOCK (中断\_块)  
 END\_INTERRUPT\_BLOCK (结束\_中断\_块)  
 所有语句列表指令助记符均为关键字。  
 所有带大小标识符的内存类型助记符均为关键字。

另请参阅:

[ASCII文本文件\(.awl\)的词汇规则](#)

## 8.10 ASCII (.awl) 文本文件的语法规则

语法说明

[	表示方括号中的内容为选项
{	表示内容可被写入零次或多次
(	圆括号中的项目具有最高优先级, 通常定义一个表达符号
.. ..	左侧项目或右侧项目有效
关键字	关键字大写。不区分大小写; 可使用小写、大写或大小写混合。
..	表示范围, B..Z表示B至Z之间的所有字符
CR	表示行回车(行结尾字符)

SIMATIC S7-200词汇规则

字母和数字	
字母	A B .. Z a b .. z
数字	0 1 .. 9
位数字	0 1 .. 7
二进制数字	0 1 _
十六进制数字	数字 A .. F a .. f _
字符	字母 ASCII 128 .. ASCII 255 _
第一标识字符	字符 空格
其他标识字符	字符 数字 空格
局部变量(局部符号)	[#]标识字符

注释:

输入时, 用户不必键入井(#)字符, 如果已经定义局部变量, 则自动加井号(#)。输入时, 所有的连续空格被转换成单个空格, 然后自动用一个下划线代替每个空格。最后, 连续的多个下划线自动被减少成一个下划线。局部变量或局部符号的长度必须至少为一个字符, 而且不能超出该字符集中的 23个字符。

常量

布尔常量	真 假 开 关	注意: 不区分大小写
常量说明符	B# BYTE# W# WORD# DW# DWORD#	注意: 不区分大小写
不带符号的整数	[常量说明符]数字{数字}	
数字	数字{数字}	
带符号的数字	[+ -]数字	
带符号的整数	[常量说明符]带符号的数字	
二进制整数	[常量说明符]2#二进制数字{二进制数字}	
十六进制整数	[常量说明符]16#十六进制数字{十六进制数字}	
指数	(E e)带符号的数字	
实数常量	带符号的数字.{??}	
实数常量	数字{指数}	
实数常量	带符号的数字.数字{指数}	
可打印的字符	ASCII 32 .. ASCII 255 不包括 DEL 和 '	
双字符十六进制	十六进制数字 十六进制数字	
字符表示	可打印的字符 \$\$ \$L \$l \$N \$n \$P \$p \$R \$r \$T \$t \$' \$"\$ \$二字符 十六进制 注意: 开头的 \$	
号是控制字符	(也称为还原代码)	

还原代码	解释
\$\$	单美元号
\$'	单引号字符
\$"	双引号字符
\$L 或 \$l	行回车字符
\$N 或 \$n	新行字符
\$P 或 \$p	行回车、新页
\$R 或 \$r	行回车字符
\$T 或 \$t	制表字符

ASCII字符常量 [常量说明符]字符表示  
ASCII字符串常量 "字符表示"

常量 布尔常量 | 二进制数字 | 不带符号的整数 | 带符号的整数 | 二进制整数 | 十六进制整数 | 实数常量 | ASCII  
常量

**注释:**

ASCII常量必须与字符说明符相匹配,例如**B#a**有效,但**B#ab**无效。**B#**要求一个并仅限一个字符,**W#**确切要求两个字符,**DW#**确切要求四个字符。任何其他字符数目都将导致错误。

**程序注解**

注解字符 可打印的字符 CR  
注解 //{注解字符}CR

**编址**

离散输入 I | E 注释:不区分大小写  
离散输出 Q | A 注释:不区分大小写  
模拟输入 AI | AE 注释:不区分大小写  
模拟输出 AQ | AA 注释:不区分大小写  
db 内存 DB 注释:不区分大小写  
变量内存 V 注释:不区分大小写  
标志内存 M 注释:不区分大小写  
系统内存 SM 注释:不区分大小写  
阶段内存 S 注释:不区分大小写  
累加器 AC 注释:不区分大小写  
计时器 T 注释:不区分大小写  
计数器 C | Z 注释:不区分大小写  
高速计数器 HC | HZ 注释:不区分大小写  
数据块 DB 注释:不区分大小写  
局部内存 L 注释:不区分大小写

位点 不带符号的数字 . 0 | 1 | ? | 7  
尺寸前缀 B | W | D 注释:不区分大小写  
直接前缀 离散输入 | 离散输出 | 变量内存 | 旗标内存 | 系统内存 | 阶段内存  
外围前缀 模拟输入 | 模拟输出  
计时器计数器前缀 计时器 | 计数器

位地址 直接前缀 [X] 位点  
字节字双字地址 直接前缀大小前缀不带符号的数字  
外围地址 外围前缀 [W] 不带符号的数字  
计时器计数器地址 计时器计数器前缀 ( (X不带符号的数字) | 不带符号的数字)  
高速计数器地址 高速计数器不带符号的数字  
累加器地址 累加器[大小前缀]不带符号的数字

db 号码 DB 不带符号的数字

db 地址 [db 号码.]数据块 ([X] 位点) | (大小前缀 不带符号的数字)

直接地址 位地址 | 字节 字 双字 地址 | 外围地址 | 计时器计数器地址 | 高速计数器地址 | 累加器地址 | db  
地址

间接地址 \* ( (累加器 [D|d]) | (变量内存 D|d) | (局部内存 D|d) | ([db 号码.]数据块 D|d) )  
不带符号的数字

地址 & (变量内存 | 离散输入 | 离散输出 | 计时器 | 计数器 | 旗标内存 | 系统内存 ([db 号码.]数据块)  
尺寸前缀不带符号的数字

**程序结构****注释:**

关键字不区分大小写,除非另外注明。

块标题 { 空白空间 } (标题 = 注解字符 CR | 注解)  
块注解 {??}  
网络标题 { 空白空间 } (标题 = 注解字符 CR | 注解)  
网络注解 {??}  
线路注解 {??}  
代码注解 {??}  
网络号码 { 空间 | 不带符号的数字 }  
操作数 { 空白空间 } 全局变量 | 局部变量 | 常量 | 直接地址 | 间接地址 | 地址  
指令助记符 (字母 | 数字 | \_) { 字母 | 数字 | \_ }

指令 语句列表	指令助记符 [ (空白空间) { 空白空间 } 操作数 { (,  空白空间) { 空白空间 } 操作数}]; 代码注解 {??   行注解}
网络	网络 网络号码 (CR   [网络标题] [网络注解]) 语句列表
ob号码 组织块 CR	OB 不带符号的数字 组织块 (ob 号码) (CR   [块标题] [块注解]) ob 说明 BEGIN [网络] END_ORGANIZATION_BLOCK CR
sbr号码 子例行程序块 END_SUBROUTINE_BLOCK CR	SBR 不带符号的数字 子例行程序块 (sbr 号码) (CR [块标题] [块注解]) sbr 说明 BEGIN [网络] END_SUBROUTINE_BLOCK CR
整数数字 中断块	INT 不带符号的数字 中断块 (整数数字) (CR [块标题] [块注解]) int 说明 BEGIN [网络] END_INTERRUPT_BLOCK CR
<b>局部变量说明</b>	
ob说明	[var说明]
int说明	[var说明]
sbr说明	[io var说明] ob说明
io var说明	[var输入说明] [var入出说明] [var输出说明]
var输入说明	VAR_INPUT CRdecl_list END_VAR CR
var输出说明	VAR_OUTPUT CR
	decl_listEND_VAR CR
var入出说明	VAR_IN_OUT CRdecl_listEND_VAR CR
var说明	VAR CRdecl_list END_VAR CR
decl列表	{标识符: 复杂类型;} {注解}
类型	基本类型   用户类型   字符串类型
复杂类型	基本类型   数列类型   结构类型   用户类型   字符串类型
初始数值	常量   整数 (初始数值)   初始数值, 初始数值
基本类型	布尔   字节   字   双字   字符   整数   双整数   实数
<b>数据块</b>	
<b>注释:</b>	
	数据块 *.txt 文件与程序 *.txt 文件分开且必须分别导入
数据块标记页1	DATA_BLOCK_TAB (name_1) (CR [块注解] [V分配] END_DATA_BLOCK_TAB CR
数据块标记页n	DATA_BLOCK_TAB (name_n) (CR [块注解] [V分配] END_DATA_BLOCK_TAB

另请参阅:  
[ASCII文件关键字列表](#)

## 8.11 CPU内存中的指令大小

	CPU 221	CPU 222	CPU 224	CPU 226	CPU
<b>226XM[[[ENDKEEPN]]]</b>					
最大程序尺寸	2048个字	2048个字	4096个字	4096个字	8192个字[[[ENDKEEPN]]]
字节地址	<b>参数大小 (以字节为单位) [[[ENDKEEPN]]]</b>				
直接地址	<b>2个字节</b> (用于字节偏移量0~4095) 或 <b>4个字节</b> (用于字节偏移量 > 4096) [[[ENDKEEPN]]]				
间接地址	<b>4个字节</b> (用于*VD、LD) 或 <b>0个字节</b> (用于*AC) [[[ENDKEEPN]]]				
累加器 (AC)	<b>0个字节[[[ENDKEEPN]]]</b>				
<b>位址</b>	<b>[[[ENDKEEPN]]]</b>				
直接地址	<b>2个字节</b> (用于字节偏移量0-511) 或 <b>4个字节</b> (用于字节地址 > 511) [[[ENDKEEPN]]]				
<b>常量</b>	<b>[[[ENDKEEPN]]]</b>				

字节、字、双字、字符串 间接地址 请参阅助记符表中的第4列。第4列表明可为常量的参数及该常量的大小。BC - 1个字节常量; 1个字节 WC - 2个字节 DW - 双字常量; 4个字节 StrC - 字符串常量: 0-126个字节 + 1个字节的长度域。& - 间接地址常量: 4个字节 (例如: &VB0) [[[ENDKEEPN]]]

计算某指令长度: 确定您使用的参数类型, 在助记符表中找到该指令, 并增加“基本大小”(第2列)和正确的“参数大小”(第3列)。“参数注释”(第4列)提供了确定指令长度所需的附加信息。最大指令长度为255字节。如果超出最大长度, 程序就会发回错误讯息。

[[[ENDKEEPN]]]STL助记符和参数 参数注释[[[ENDKEEPN]]]		基本大小字节	参数大小 字节	
LD	N	2	0	字节偏移量0-
63:	(仅限I、Q、M、S、SM、T、C)			
LDN	N	2	0	字节偏移量0-
63:	(仅限I、Q、M、S、SM、T、C)			
A	N	2	0	字节偏移量0-
63:	(仅限I、Q、M、S、SM、T、C)			
AN	N	2	0	字节偏移量0-
63:	(仅限I、Q、M、S、SM、T、C)			
O	N	2	0	字节偏移量0-
63:	(仅限I、Q、M、S、SM、T、C)			
ON	N	2	0	字节偏移量0-
63:	(仅限I、Q、M、S、SM、T、C)			
=	N	2	0	字节偏移量0-
63:	(仅限I、Q、M、S、SM、T、C)			
LD	N	1	p1	p1: 位址
LDN	N	1	p1	p1: 位址
A	N	1	p1	p1: 位址
AN	N	1	p1	p1: 位址
O	N	1	p1	p1: 位址
ON	N	1	p1	p1: 位址
=	N	1	p1	p1: 位址
LDI	N	1	p1	p1: 位址
LDNI	N	1	p1	p1: 位址
AI	N	1	p1	p1: 位址
ANI	N	1	p1	p1: 位址
OI	N	1	p1	p1: 位址
ONI	N	1	p1	p1: 位址
=I	N	1	p1	p1: 位址
+D	IN1, IN2	2	p1+p2	p2: DWC
-D	IN1, IN2	2	p1+p2	p2: DWC
*D	IN1, IN2	3	p1+p2	p1: DWC
/D	IN1, IN2	3	p1+p2	p1: DWC
+I	IN1, IN2	2	p1+p2	p2: WC
-I	IN1, IN2	2	p1+p2	p2: WC
*I	IN1, IN2	3	p1+p2	p1: WC
/I	IN1, IN2	3	p1+p2	p1: WC
+R	IN1, IN2	3	p1+p2	p1: DWC
-R	IN1, IN2	3	p1+p2	p1: DWC
*R	IN1, IN2	3	p1+p2	p1: DWC
/R	IN1, IN2	3	p1+p2	p1: DWC
AB<	N1, N2	3	p1+p2	p1, p2: BC
AB<=	N1, N2	2	p1+p2	p1, p2: BC
AB<>	N1, N2	3	p1+p2	p1, p2: BC
AB=	N1, N2	2	p1+p2	p1, p2: BC
AB>	N1, N2	3	p1+p2	p1, p2: BC
AB>=	N1, N2	2	p1+p2	p1, p2: BC
AD<	N1, N2	3	p1+p2	p1, p2: DWC
AD<=	N1, N2	2	p1+p2	p1, p2: DWC
AD<>	N1, N2	3	p1+p2	p1, p2: DWC
AD=	N1, N2	2	p1+p2	p1, p2: DWC
AD>	N1, N2	3	p1+p2	p1, p2: DWC

AD>=	N1, N2	2		p1+p2	p1, p2: DWC
AENO	1	0			
ALD	1	0			
ANDB	IN1, IN2	2		p1+p2	p1: BC
ANDD	IN1, IN2	2		p1+p2	p1, p2: DWC
ANDW	IN1, IN2	2		p1+p2	p1, p2: WC
AR<	N1, N2	3		p1+p2	p1, p2: DWC
AR<=	N1, N2	3		p1+p2	p1, p2: DWC
AR<>	N1, N2	3		p1+p2	p1, p2: DWC
AR=	N1, N2	3		p1+p2	p1, p2: DWC
AR>	N1, N2	3		p1+p2	p1, p2: DWC
AR>=	N1, N2	3		p1+p2	p1, p2: DWC
AS<>	S1, S2	7		p1+p2	p1, p2: StrC
AS=	S1, S2	7		p1+p2	p1, p2: StrC
ATCH	INT, EVENT		2		2
ATH	IN, OUT, LEN		5	p1+p2+p3	p3: BC
ATT	DATA, TABLE		2	p1+p2	p1: WC
AW<	N1, N2	3		p1+p2	p1, p2: WC
AW<=	N1, N2	2		p1+p2	p1, p2: WC
AW<>	N1, N2	3		p1+p2	p1, p2: WC
AW=	N1, N2	2		p1+p2	p1, p2: BC
AW>	N1, N2	3		p1+p2	p1, p2: WC
AW>=	N1, N2	2		p1+p2	p1, p2: WC
BCDI	IN	2		p1	
BIR	IN, OUT	3		p1+p2	
BITIM	OUT	3	p1		
BIW	IN, OUT	3		p1+p2	p1: BC
BMB	IN, OUT, N	5		p1+p2+p3	p3: BC
BMD	IN, OUT, N	5		p1+p2+p3	p3: BC
BMW	IN, OUT, N	5		p1+p2+p3	p3: BC
BTI	IN, OUT	3		p1+p2	p1: BC
CALL	N	1		2	
CEVNT	EVNT	3	1		
CFND	S1, S2, R	9		p1+p2+p3	p1, p2: StrC
CITIM	IN,OUT	3	p1+p2		
COS	IN, OUT	3		p1+p2	p1: DWC
CRET	1	0			
CRETI	1	0			
CSCRE	1	0			
CTD	CXXX, PV	3		2+p2	p2: WC
CTU	CXXX, PV	2		2+p2	p2: WC
CTUD	CXXX, PV	2		2+p2	p2: WC
DECB	IN	2		p1	
DECD	IN	2		p1	
DECO	IN, OUT	4		p1+p2	p1: BC
DECW	IN	2		p1	
DISI	1	0			
DIV	IN1, IN2	2		p1+p2	p2: WC
DLED	IN	3		p1	p1: BC
DTA	IN, OUT, FORMAT	5		p1+p2+p3	p1: DWC, p3: BC
DTCH	EVENT	2		1	
DTI	IN, OUT	3		p1+p2	p1: DWC
DTR	IN, OUT	3		p1+p2	p1: DWC
DTS	IN, S, FMT	5		p1+p2+p3	p1: DWC, p3: BC



ED	1	0			
ENCO	IN, OUT	4		p1+p2	p1: WC
END	1	0			
ENI	1	0			
EU	1	0			
EXP	IN, OUT	3		p1+p2	p1: DWC
FIFO	TABLE, DATA	2		p1+p2	
FILL	IN, OUT, N	5		p1+p2+p3	p1: WC p3: BC
FND<	SRC, PATRN, INDX	5		p1+p2+p3	p2: WC
FND<>	SRC, PATRN, INDX	5		p1+p2+p3	p2: WC
FND=	SRC, PATRN, INDX	5		p1+p2+p3	p2: WC
FND>	SRC, PATRN, INDX	5		p1+p2+p3	p2: WC
FOR	INDEX, INITIAL, FINAL	5		p1+p2+p3	p2, p3: WC
GPA	ADDR, PORT	3		p1+1	
HDEF	HSC, MODE	2		2	
HIDE	N, PASSWD	3		6	
HSC	N	2	3		
HTA	IN, OUT, LEN	5		p1+p2+p3	p3: BC
IBCD	IN	2		p1	
INCB	IN	2		p1	
INCD	IN	2		p1	
INCW	IN	2		p1	
INT	N	1		2	
INVB	IN	2		p1	
INVD	IN	2		p1	
INW	IN	2		p1	
ITA	IN, OUT, FORMAT	5		p1+p2+p3	p1: WC, p3: BC
ITB	IN, OUT	3		p1+p2	p1: WC
ITD	IN, OUT	3		p1+p2	p1: WC
ITS	IN, S, FMT	5		p1+p2+p3	p1: WC, p3: BC
JMP	N	1	2		
LBL	N	1	2		
LDB<=	N1, N2	2		p1+p2	p1, p2: BC
LDB<	N1, N2	3		p1+p2	p1, p2: BC
LDB<>	N1, N2	3		p1+p2	p1, p2: BC
LDB=	N1, N2	2		p1+p2	p1, p2: BC
LDB>=	N1, N2	2		p1+p2	p1, p2: BC
LDB>	N1, N2	3		p1+p2	p1, p2: BC
LDD<	N1, N2	3		p1+p2	p1, p2: DWC
LDD<=	N1, N2	2		p1+p2	p1, p2: DWC
LDD<>	N1, N2	3		p1+p2	p1, p2: DWC
LDD=	N1, N2	2		p1+p2	p1, p2: DWC
LDD>	N1, N2	3		p1+p2	p1, p2: DWC
LDD>=	N1, N2	2		p1+p2	p1, p2: DWC
LDR<	N1, N2	3		p1+p2	p1, p2: DWC
LDR<=	N1, N2	3		p1+p2	p1, p2: DWC
LDR<>	N1, N2	3		p1+p2	p1, p2: DWC
LDR=	N1, N2	3		p1+p2	p1, p2: DWC
LDR>	N1, N2	3		p1+p2	p1, p2: DWC
LDR>=	N1, N2	3		p1+p2	p1, p2: DWC
LDS	X	1	1		
LDS<>	S1, S2	7		p1+p2	p1, p2: StrC
LDS=	S1, S2	7		p1+p2	p1, p2: StrC
LDW<	N1, N2	3		p1+p2	p1, p2: WC

LDW<=	N1, N2	2		<b>p1+p2</b>	p1、p2: WC
LDW<>	N1, N2	3		<b>p1+p2</b>	p1、p2: WC
LDW=	N1, N2	2		<b>p1+p2</b>	p1、p2: WC
LDW>	N1, N2	3		<b>p1+p2</b>	p1、p2: WC
LDW>=	N1, N2	2		<b>p1+p2</b>	p1、p2: WC
LIFO	TABLE, DATA		2		<b>p1+p2</b>
LN	IN, OUT	3		<b>p1+p2</b>	p1: DWC
LPP	1	0			
LPS	1	0			
LRD	1	0			
LSCR	S	2		<b>p1</b>	p1: 位址
MEND	1	0			
MOVB	IN, OUT	2		<b>p1+p2</b>	p1: BC
MOVD	IN, OUT	2		<b>p1+p2</b>	p1: DWC、&
MOVR	IN, OUT	3		<b>p1+p2</b>	p1: DWC
MOVW	IN, OUT	2		<b>p1+p2</b>	p1: WC
MUL	IN1, IN2	2		<b>p1+p2</b>	p2: WC
NETR	TABLE, PORT		2		<b>p1+1</b>
NETW	TABLE, PORT		2		<b>p1+1</b>
NEXT	1	0			
NOP	N	1		1	
NOT	1	0			
OB<	N1, N2	3		<b>p1+p2</b>	p1、p2: BC
OB<=	N1, N2	2		<b>p1+p2</b>	p1、p2: BC
OB<>	N1, N2	3		<b>p1+p2</b>	p1、p2: BC
OB=	N1, N2	2		<b>p1+p2</b>	p1、p2: BC
OB>	N1, N2	3		<b>p1+p2</b>	p1、p2: BC
OB>=	N1, N2	2		<b>p1+p2</b>	p1、p2: BC
OD<	N1, N2	3		<b>p1+p2</b>	p1、p2: DWC
OD<=	N1, N2	2		<b>p1+p2</b>	p1、p2: DWC
OD<>	N1, N2	3		<b>p1+p2</b>	p1、p2: DWC
OD=	N1, N2	2		<b>p1+p2</b>	p1、p2: DWC
OD>	N1, N2	3		<b>p1+p2</b>	p1、p2: DWC
OD>=	N1, N2	2		<b>p1+p2</b>	p1、p2: DWC
OLD	1	0			
OR<	N1, N2	3		<b>p1+p2</b>	p1、p2: DWC
OR<=	N1, N2	3		<b>p1+p2</b>	p1、p2: DWC
OR<>	N1, N2	3		<b>p1+p2</b>	p1、p2: DWC
OR=	N1, N2	3		<b>p1+p2</b>	p1、p2: DWC
OR>	N1, N2	3		<b>p1+p2</b>	p1、p2: DWC
OR>=	N1, N2	3		<b>p1+p2</b>	p1、p2: DWC
ORB	IN1, IN2	2		<b>p1+p2</b>	p1: BC
ORD	IN1, IN2	2		<b>p1+p2</b>	p1、p2: DWC
ORW	IN1, IN2	2		<b>p1+p2</b>	p1、p2: WC
OS<>	S1, S2	7		<b>p1+p2</b>	p1、p2: StrC
OS=	S1, S2	7		<b>p1+p2</b>	p1、p2: StrC
OW<	N1, N2	3		<b>p1+p2</b>	p1、p2: WC
OW<=	N1, N2	2		<b>p1+p2</b>	p1、p2: WC
OW<>	N1, N2	3		<b>p1+p2</b>	p1、p2: WC
OW=	N1, N2	2		<b>p1+p2</b>	p1、p2: BC
OW>	N1, N2	3		<b>p1+p2</b>	p1、p2: WC
OW>=	N1, N2	2		<b>p1+p2</b>	p1、p2: WC
PCALL	N, [P0, 屨	7		<b>(2+p0) + . (2+pX)</b>	p0-

pX: BC、WC、DWC、StrC

PID	TABLE, LOOP	2	2	p1+1	
PLS	X	2	3		
R	S_BIT, N	4		p1+p2	p1: 位址, p2: BC
RCV	TABLE, PORT	2		p1+1	
RET	1	0			
RETI	1	0			
RI	S_BIT, N	4		p1+p2	p1: 位址 (只限Q), p2: BC
RLB	IN, N	4		p1+p2	p2: BC
RLD	IN, N	4		p1+p2	p2: BC
RLW	IN, N	4		p1+p2	p2: BC
ROUND	IN, OUT	3		p1+p2	p1: DWC
RRB	IN, N	4		p1+p2	p2: BC
RRD	IN, N	4		p1+p2	p2: BC
RRW	IN, N	4		p1+p2	p2: BC
RTA	IN, OUT, FORMAT	5		p1+p2+p3	p1: DWC, p3: BC
RTS	IN, S, FMT	5		p1+p2+p3	p1: DWC, p3: BC
S	S_BIT, N	4		p1+p2	p1: 位址, p2: BC
SBR	N	1	2		
SCAT	S1, S2	7		p1+p2	p1: StrC
SCPY	S1, S2	7		p1+p2	p1: StrC
SCRE	1	0			
SCRT	S	2		p1	p1: 位址
SEG	IN, OUT	4		p1+p2	p2: BC
SFND	S1, S2, R	9		p1+p2+p3	p1, p2: StrC
SHRB	DATA, S_BIT, N	5		p1+p2+p3	p1, p2: 位址, p3: BC
SI	S_BIT, N	4		p1+p2	p1: 位址 (只限Q), p2: BC
SIN	IN, OUT	3		p1+p2	p1: DWC
SLB	IN, N	4		p1+p2	p2: BC
SLD	IN, N	4		p1+p2	p2: BC
SLEN	S, R	7		p1+p2	p1: StrC
SLW	IN, N	4		p1+p2	p2: BC
SPA	ADDR, PORT	3		p1+1	
SQRT	IN, OUT	3		p1+p2	p1: DWC
SRB	IN, N	4		p1+p2	p2: BC
SRD	IN, N	4		p1+p2	p2: BC
SRW	IN, N	4		p1+p2	p2: BC
SSCPY	S1, P, N, S2		11	p1+p2+p3+p4	p1: StrC, p2, p3: BC
STD	S, P, OUT	9		p1+p2+p3	p1: StrC p2: BC
STI	S, P, OUT	9		p1+p2+p3	p1: StrC p2: BC
STOP	1	0			
STR	S, P, OUT	9		p1+p2+p3	p1: StrC p2: BC
SWAP	IN	2		p1	
TAN	IN, OUT	3		p1+p2	p1: DWC
TODR	T	2		p1	
TODRX	T	3		p1	
TODW	T	2		p1	
TODWX T	3		p1		
TOF	TXXX, PT	3		2+p2	p2: WC
TON	TXXX, PT	2		2+p2	p2: WC
TONR	TXXX, PT	2		2+p2	p2: WC
TRUNC	IN, OUT	3		p1+p2	p1: DWC
WDR	1				
XMT	TABLE, PORT	2		p1+1	

XORB	IN1, IN2	2	<b>p1+p2</b>	p1: BC
XORD	IN1, IN2	2	<b>p1+p2</b>	p1、p2: DWC
XORW	IN1, IN2	2	<b>p1+p2</b>	p1、p2: WC

获取更多资料 微信搜索 蓝领星球

获取更多资料 微信搜索蓝领星球

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part



IX

获取更多资料 微信搜索蓝领星球

## 9 如何执行常见任务

### 9.1 设置工作区选项

**工具 > 选项** 菜单命令设置:

#### 通用标记

默认编辑器  
助记符集  
测量系统  
时间格式  
日期格式  
SIMATIC或IEC 1131-3编程模式  
语言

#### 项目标记

默认文件位置  
默认PLC类型  
系统符号表

#### 色彩标记

字体和色彩首选项

#### 程序编辑器标记

栅格宽度 (以像素为单位)  
字体类型、样式和大小  
显示符号和地址  
在放置指令后启用操作数编辑  
输入后格式化STL代码

#### 符号表标记

字体和色彩首选项  
显示重叠、未使用和库符号

#### 数据块标记

字体和色彩首选项

#### 状态图标记

字体和色彩首选项  
显示符号和地址

#### 交叉引用标记

字体和色彩首选项  
显示符号和地址

#### 输出窗口标记

字体和色彩首选项

#### 指令树标记

字体和色彩首选项  
启用指令树自动压缩

#### 浏览条标记

字体和色彩首选项

#### 打印标记

字体和色彩首选项

#### STL状态标记

观察数值、操作数数目和堆栈位数目  
启用指令状态位监控

**工具 > 定制** 菜单命令设置:

#### 命令标记

检视或隐藏工具提示  
选择三维或扁平按钮  
配置工具条按钮

#### 增加工具标记

将您喜爱的应用程序增加至“工具”菜单

## 9.2 编辑程序

### 9.2.4 使用数据块在V内存中存储数据

### 9.2.4 使用数据块在V内存中存储数据

### 9.2.4 使用数据块在V内存中存储数据

### 9.2.4 使用数据块在V内存中存储数据

使用下列一种方法存取数据块：




- 单击浏览条上的“数据块”按钮。
- 选择菜单命令 **检视 (V) > 数据块 (D)**。
- 打开 **指令树** 中的“数据块”文件夹，然后双击某块页图标。

通过插入新数据块页标记，将您的数据块V内存赋值分成多个功能组：

- 单击数据块窗口，然后选取菜单命令 **编辑 (E) > 插入 (I) > 数据块 (D)**
- 在 **指令树** 中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取 **插入 (I) > 数据块 (D)**
- 用鼠标右键单击数据块窗口，然后在弹出菜单中选取 **插入 (I) > 数据块 (D)**
- 标记的最大数目为128。如果您使用向导，有关标记会被自动创建以支持向导功能。您可以创建的标记的最大数目为（128 - 由MicroWin自动创建的标记数目）。请使用Windows剪贴板合并标记数据；方法为使用剪切和粘贴由一个标记转移到另一块页。

重新命名和保护数据块页标记：

- 在 **指令树** 中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取 **重新命名**。您也可以在指令树内直接重新命名数据块页，方法为单击该标记页名称两次（动作要慢一些，以免解释成双击）；然后编辑该标记名。数据块编辑器提供相同的重新命名功能，方法为用鼠标右键直接单击该标记名。
- 在 **指令树** 中，用鼠标右键单击数据块页图标，然后在弹出菜单中选取 **属性**。由此，您可以重新命名该数据块标记和

指定作者。属性对话框的 **保护** 标记令您能够用密码保护单个数据块标记。受保护的标记会显示锁图标 。数据块编辑时，向向导会创建不能重新命名且包含只读数据值的受保护标记。

导入和导出数据块数据至ASCII文本文件

- [由ASCII文本文件导入数据块](#)
- [由ASCII文本文件导出数据块](#)

本标题讨论下列主题：

[在数据块中进行地址和数据值赋值](#)

[数据块举例](#)

[在数据块编辑器中使用“剪切”、“复制”和“粘贴”](#)

[使用ASCII常量的限制](#)

[理解和解决错误](#)

[将数据块下载至PLC](#)

[从PLC上载数据块](#)

[上载后保留数据块格式](#)

[从CPU RAM建立数据块](#)

[在数据块中进行地址和数据值赋值](#)

[返回顶端](#)

数据块仅允许您对V内存进行初始数据或ASCII字符赋值。您可以对V内存的字节 (V或VB)、字 (VW) 或双字 (VD) 赋值。注解（前面带双斜线//）是选用项目。

- 数据块的第一行必须包含一个明确地址赋值（绝对或符号地址），其后的行可包含明确或隐含地址赋值。当您在单地址赋值后键入多个数据值或键入仅包含数据值的行时，由编辑器指定隐含地址赋值。编辑器根据先前的地址分配及数据值大小（字节、字或双字）指定适当的V内存数量。
- 数据块编辑器是一种自由格式文本编辑器，对特定类型的信息无具体域。键入一行后，按 ENTER 键，数据块编辑器格式交校云氩 妨小(6) 尊(9)13.猓徊痘 V内存地址) 并重新显示行。数据块编辑器接受大小写字母并允许使用逗号、制砑 蚩崙痲 魑 卍泛褪 莸抵 涑姆指解



- 在完成一赋值行后按CTRL-ENTER键组合，会令地址自动增加至下一个可用地址。

[常量格式详细说明](#)

**数据块一般规则**

```

VB0    248           //明确地址赋值：VB0数据值：248。

VB1    249, 250, 251 //单行中多个数据值。
           //隐含地址赋值：
           //VB2包含数据值250。
           //VB3包含数据值251。

VB4    252           //不能使用先前指定的地址（VB0-VB3）。

           253, 254, 255 //无明确地址赋值的行。
           //数据值隐含指定给VB5、VB6、VB7。

VW8    256, 257      //新数据类型（字）。隐含将数据值
           //257指定给V内存VB10-VB11。

//地址赋值不能与先前的明确赋值发生冲突。
X      65536         //数据值65536要求双字数据类型（VD内存），
           //但上一个明确地址赋值
           //是字内存（VW8）。编辑器标志错误。

//您可以从地址中省略尺寸规格。
//然后您可以对随后的数据使用隐含地址赋值，
//不必考虑数据的尺寸。

V12    258           //字值明确指定给V12-V13。
           65537      //双字值隐含指定给V14-V17。

```

[数据块举例](#)

[返回顶端](#)

**直接地址和数值**

```

数据块
//
//数据块注解
//
VB0    255           //字节值从VB0开始
VW2    256           //字值从VW2开始
VD4    700.59       //双字真值从VD4开始
VB8    -35           //字节值从VB8开始
VW10   16#A         //字值HEX从VW10开始
VD14   146879       //双字值从VD14开始
VW20   2, 4, 16, 32, 64 //字值表从VD14开始
      -2, -4, -16, -32, -64 //扩展至多行的数值不能出现在列中
//第一列为内存地址保留
VB45   'Up'         //起始于VB45的双字节ASCII字符串
VW90   65535        //起始于VW90的字值
V100   255, 'abcd', 65535, 1.0 //起始于VB100的未定义数值：可混合不同尺寸
VB110  255          //与输入B110 255相同
VW120  256          //与输入W120 256相同
VD130  700.59       //与输入D130 700.59相同
V140   255, 'abc', 65535, 1.0 //与输入140相同：必须从列1开始
VW150  2#1010101010101010 //二进制字值
VD152  2#1100110011001100110011001100 //二进制双字值

```

符号地址和符号数值

符号表			
符号	地址	注解	
1	符号地址	VB1	符号地址
2	符号常数	200	符号常数

```

数据块
2. VB0 100
   符号地址 符号常数 //为VB1指定一个200数值

```

替代二进制输入方法（上窗口）和结果二进制格式（下窗口）

```

数据块
2. VB0 2 1 0 true false off on

```



```

数据块
2. VB0 2#10010100 //按Enter键后二进制模式

```

在数据块编辑器中使用“剪切”、“复制”和“粘贴”  
[返回顶端](#)

当在数据块中执行剪切、复制、粘贴时，重要的是通过鼠标单击设置适当的上下文：

如果您单击指令树中的数据块标记页  图标，或者单击数据块窗口中的标记名 ，  
 则 霰县且成喜僮鳌 T 洼松舷挛闹杏檬孟暖壹 七 颯 突嵯允咀饕糜谩 黎 菘楸县且车牟僮鞞 霾说 ？  
 如果您单击数据块编辑器中的文字区域，那么您将在数据块窗口中已选定(突出显示)的文字上操作。在此上下文中用鼠标右  
 七 颯 突嵯允咀饕糜谩 黎 菘楸县且衬谔 《 刁 淖 稚 系 牟 僮 鞞 霾 说 ？

### 使用ASCII常量的限制

[返回顶端](#)

有效的ASCII赋值：

唯有字节地址(V或VB)能够与长ASCII常量(使用单引号或双引号格式)合用：

VBO 'A'、VWO 'AB'、VDO 'ABCD'

就3和5或更多字节而言，您必须使用V或VB地址前缀 VBO 'ABC'、VBO 'ABCDE'、VBO 'ABCDEFGH IJK'



双引号ASCII常量字符串格式存储一个前导长度字节和字符串 VBO "A"、VBO "AB"、VBO "ABC"、VBO "ABCD"、VBO "ABCDEFGH IJK"

[常量格式详细说明](#)

### 理解和解决错误

[返回顶端](#)

一旦在包含错误的行尾按ENTER键，立即会在数据块左页边显示输入错误。您必须纠正全部输入错误，才能成功地编译。

	vbbo	255		// (红色文本 - 非法语法错误) 应为vb0
	VBO	300		// (红色曲线下划线 - 非法使用错误) 一个字节最大为255

引起输入错误的条件包括：

- 指定错误内存区 (V是唯一允许使用的内存区)
- 当数据值实际要求较大的尺寸时(例如，数据值256过大，无法在VB地址中存储—要求使用VW地址)，在地址赋值兄付 骋荒洼姆叽纾 卫纸译厝郑
- 在一行中使用错误序列：在数据值之后(而不是在数据值之前)键入内存地址
- 使用非法语法或无效数值
- 尝试使用符号，而不是使用绝对V内存地址(数据块中不允许使用符号)
- 未能适当地指定注解(双正斜线必须位于注解之前：//注解样本)如果数据块是现用窗口，您可以使用菜单命令 **PLC > 编译 (Compile)** 编译数据块。如果数据块不是现用窗口，您依然可以编译数据块：使用菜单命令 **PLC > 全部编译 (Compile All)**。

编译数据块时，如果编译程序发现错误，会在“输出窗口”显示错误。将光标置于“输出窗口”中的错误讯息上，双击该讯息  
 谿 崧椽翱溢邢允境趵刚小  
 仅在编译后显示的错误包括：

编译数据块时，如果编译程序发现错误，会在“输出窗口”显示错误。将光标置于“输出窗口”中的错误讯息上，双击该讯息，  
 在数据块窗口中显示出错行。  
 仅在编译后显示的错误包括：

- 重复地址赋值(例如，如果您输入“VB1 249, 250”之类的行，则是对VB2进行250隐含赋值—您不得在别处对地址VB2进行其他不同的数据值赋值)
- 地址重叠(例如，如果您为VDO指定一个类似65536的双数值，则不得再对V1、V2或V3指定其他赋值，因为这些数值已被使用，是以VDO开始的双字的一部分)

### 将数据块下载至PLC

[返回顶端](#)

如果要编辑数据块，则需将数据块 **下载** 至PLC。只有在修改的数据块下载后您的编辑才会生效。  
 欲节省空间，您可以将您不希望下载至PLC的信息切换为打开/关闭。

### 从PLC上载数据块

[返回顶端](#)

您必须在STEP 7-Micro/WIN中打开一个项目，才能 **上载** 数据块。

如果PLC中的数据块与您打开的项目中的数据块不匹配(或者您打开的项目中没有数据块)，您只能上载该 PLC的数据块地址  
 糠趾褪 莼挡糠郑 荒茆显刈13.袞 5 卍芬宰纸谿 蹈裕缴显兀 莼狄允 聘裕缴显兀ò 匕魏卧 模 H纛 谿】占  
 ，您可以将您不希望下载至PLC的信息切换为打开/关闭。

如果CPU中的数据块包含由Micro/WIN 4.0(或较新版本)下载的标记信息，那么由Micro/WIN

4.0的上载将会重新载入此标记结构。如果某数据块曾由较早的Micro/WIN版本下载且无标记信息，那么由Micro/WIN 4.0的上载将会将所有赋值放在同一个标记中。

#### 上载后保留DB格式

##### [返回顶端](#)

当您上载数据块时，下载至PLC的数据块条目格式被保留。如果您在一行中输入多个条目，当您上载时，格式不被保留。每个条目在一个不同的行中显示。例如，如果您在数据块中输入VB0 20、30、40、50并上载，格式更改为：

VB0	20
VB1	30
VB2	40
VB3	50

#### 从CPU RAM建立数据块

##### [返回顶端](#)

选择**PLC > 从RAM建立数据块 (Create Data Block from RAM)** 菜单命令，将CPU V内存保存至EEPROM。从CPU读取V内存当前值，然后将V内存图像作为数据块下载。

#### 注释：

进行此操作时，CPU必须处于STOP（停止）模式。

倘若CPU

RAM内有已修改的V内存地址，并且这些地址在上一次数据块下载中未得到指定，那么在您执行“从RAM建立数据块”命令时，这一研需牡牡 司坊岂恢付ù 恒鲂碌谋县敲 g 罐LC\_DATA1)。当程序的执行向新的V地址写入数据时，或当状态图被用于修改碌腓地址时，此情形就可能发生。

#### 另请参阅：

[PLC电源中断后的数据保留](#)

[常量格式详细说明](#)

[通讯配置](#)

[PLC类型选择](#)

[系统块 \(PLC配置\)](#)

[错误讯息](#)

[入门指南目录](#)

[应用程序用户参考手册](#)

## 9.2.5 查找 / 替换 / 转入编辑操作

可使用下列一种方法存取“查找”、“替换”和“转入”操作：

- 选择菜单命令**编辑 (Edit) > 查找 (Find)**、**编辑 (Edit) > 替换 (Replace)** 或 **编辑 (Edit) > 转入 (Go To)**。
- 按快捷键组合**Ctrl+F**、**Ctrl+H**或**Ctrl+G**。

可对程序、局部变量表、数据块、符号表或状态图执行“查找”、“替换”和“转入”操作。

选择一种编辑器，了解如何使用“查找 / 替换 / 转入”命令的详情：

[LAD](#)

[FBD](#)

[STL](#)

## 9.2.6 使用交叉引用和元素用法

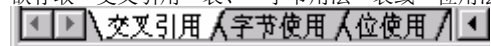
可使用下列一种方法检视“交叉引用”窗口：

- 选择菜单命令**检视 (View) > 交叉引用 (Cross Reference)**



- 单击浏览条中的“交叉引用”按钮
- 打开[指令树](#)中的“交叉引用”文件夹，然后双击某引用或使用节点

欲存取“交叉引用”表、“字节用法”表或“位用法”表，单击位于“交叉引用”窗口底部的适当标记



本标题讨论下列主题：

[交叉引用表](#)  
[字节用法表](#)  
[位用法表](#)

**交叉引用表**

[返回顶端](#)



**提示：** 您必须编译程序才能检视“交叉引用”表。

当您希望了解程序中是否已经使用和在哪里使用某一符号名或内存赋值时，可使用“交叉引用”表。“交叉引用”列表识别在程序中使用的所有操作数，并识别 POU、网络或行位置以及每次使用的操作数指令上下文。

**元素**指程序中使用的操作数。您可以在符号和绝对视图之间切换，改变全部操作数显示。（使用菜单命令 **检视 (View) > 符号编址 (Symbolic Addressing)**。）

**块**指使用操作数的 POU。

**位置**指使用操作数的行或网络。

**上下文**指使用操作数的程序指令。

LAD交叉引用列表举例

	元素	块	位置	
1	动1	MAIN (OB1)	网络 1	- -
2	动2	MAIN (OB1)	网络 2	- -
3	停止1	MAIN (OB1)	网络 1	- -
4	停止2	MAIN (OB1)	网络 2	- -
5	高等级	MAIN (OB1)	网络 1	- -
6	高等级	MAIN (OB1)	网络 2	- -

FBD交叉引用列表举例

	元素	块	位置	
1	动1	MAIN (OB1)	网络 1	OR
2	动2	MAIN (OB1)	网络 2	OR
3	停止1	MAIN (OB1)	网络 1	AND
4	停止2	MAIN (OB1)	网络 2	AND
5	高等级	MAIN (OB1)	网络 1	AND
6	高等级	MAIN (OB1)	网络 2	AND

STL交叉引用列表举例

元素	块	位置	
1 动1	MAIN (OB1)	网络 1,行 1	LD
2 动2	MAIN (OB1)	网络 2,行 1	LD
3 停止1	MAIN (OB1)	网络 1,行 3	A
4 停止2	MAIN (OB1)	网络 2,行 3	A
5 高等级	MAIN (OB1)	网络 1,行 4	AN
6 高等级	MAIN (OB1)	网络 2,行 4	AN

### 字节用法表

[返回顶端](#)



**提示：** 您必须编译程序才能检视“字节用法”表。

“字节用法”表允许您查看程序中使用了哪些字节以及在哪些内存区使用，还可帮助您识别重复赋值错误。

**b**表示已经指定一个内存位。

**B**表示已经指定一个内存字节。

**W**表示已经指定一个字（16位）。

**D**表示已经指定一个双字（32位）。

**X**用于计时器和计数器。

范例1：解释字节用法表

该字节用法表范例显示相关程序使用下列内存位置：MB0中一个位；计数器C30；计时器T37。

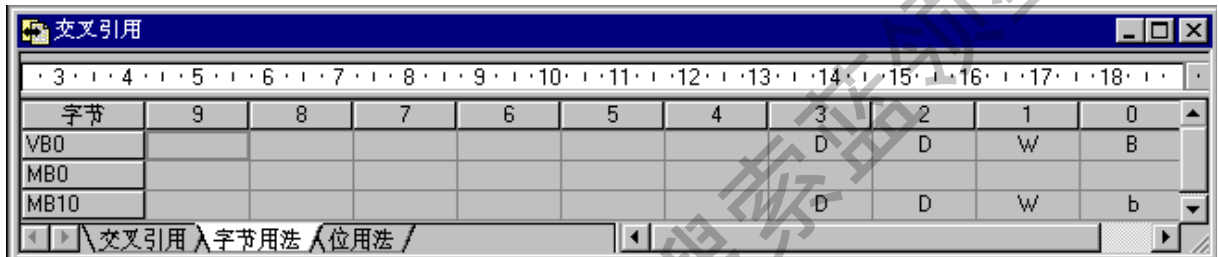
字节	9	8	7	6	5	4	3	2	1	0
MB0										b
C0										
C10										
C20										
C30										X
T0										
T10										
T20										
T30			X							

范例2：识别重复赋值错误

该范例程序从MB 10.0开始进行重复内存赋值。



可检查字节用法表，识别不适当的赋值。因为双字要求四个字节，VB0行中应有4个相邻的D。与此相似，因为字要求2个字节，VB0中应有2个相邻的W。MB10行存在相同的问题，此外在多个赋值语句中使用MB10.0。



#### 位用法表

[返回顶端](#)



**提示：** 您必须编译程序才能查看位用法表。

位用法表允许您查看程序中已经使用了哪些内存地址，可精确至位级别，还可帮助您识别重复赋值错误。

**b**表示已经指定一个内存位。

**B**表示已经指定一个内存字节。

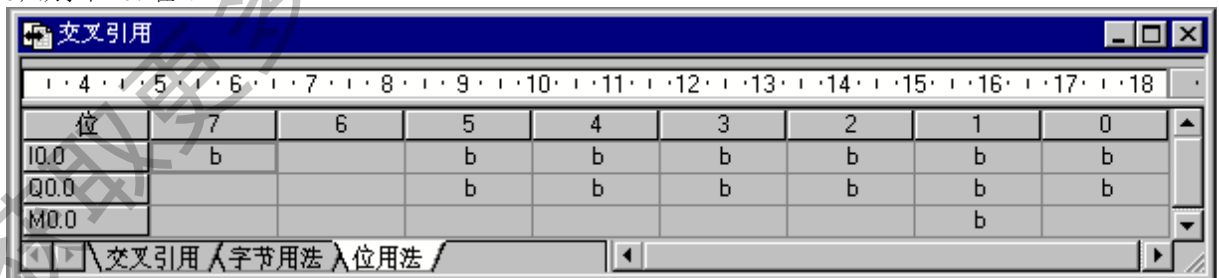
**W**表示已经指定一个字（16位）。

**D**表示已经指定一个双字（32位）。

**X**用于计时器和计数器。

范例1：解释位用法表

该位用法表范例显示相关程序使用下列内存位置：从字节I0、位0、1、2、3、4、5和7；从字节Q0、位0、1、2、3、4和5；从字节M0、位1。



范例2：识别重复赋值错误

该范例程序从MB 10.0开始进行重复内存赋值。



可检查位用法表，识别不适当的赋值。在适当的赋值程序中，字节中间不得有位值。BBBBBBb无效，而BBBBBBB则有效。嗤 慕娑丁彩视糜溢指持担虫 τ 16个相邻的W)和双字赋值(应有32个相邻的D)。

位	7	6	5	4	3	2	1	0
M0.0								
M1.0								
M2.0								
M3.0								
M4.0								
M5.0								
M6.0								
M7.0								
M8.0								
M9.0								
M10.0	B	B	B	B	B	B	B	b
M11.0	W	W	W	W	W	W	W	W
M12.0	D	D	D	D	D	D	D	D
M13.0	D	D	D	D	D	D	D	D

另请参阅：

[通讯配置](#)

[PLC类型选项](#)

[系统块 \(PLC配置\)](#)

[错误](#)



[在符号和绝对视图之间切换](#)

[入门指南目录](#)

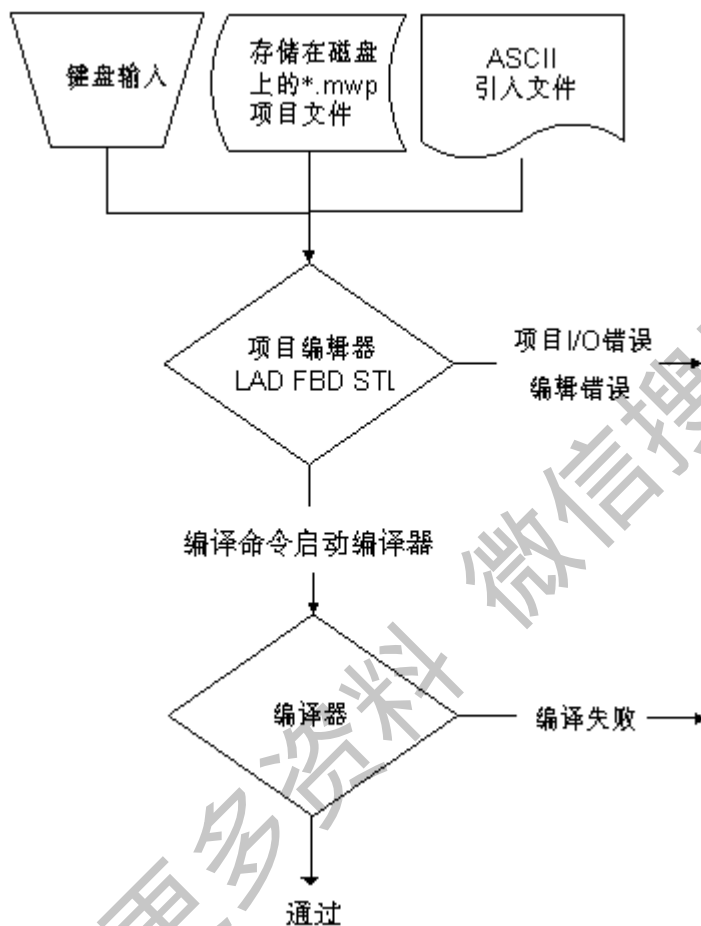
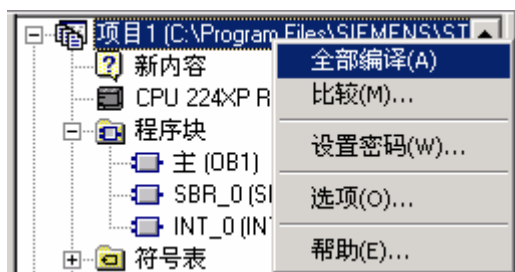
[应用程序用户参考手册](#)

## 9.2.7 成功编译

可使用下列一种方法启动STEP 7-Micro/WIN项目编译程序：

- 单击“编译”  按钮或选择菜单命令 **PLC (PLC) > 编译 (Compile)**，编译现用窗口（程序块或数据块）。
- 单击“全部编译”  按钮或选择菜单命令 **PLC (PLC) > 全部编译 (Compile All)**，编译全部项目元件（程序块、数据块和系统块）。
- 用鼠标右键单击 **指令树** 中的某个文件夹，然后由弹出菜单中选取编译命令。项目、程序块文件夹、系统块文件夹及数  
崧榭募 卸加斜喊阮 睢





设置PLC类型 项目文件I/O错误 LAD/FBD编辑器错误 全部STEP 7-Micro/WIN编译器错误均列在输出窗口中。双击错误，编辑器将滚动至错误位置。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译器。

另请参阅：  
[应用程序用户参考手册](#)  
[入门指南目录](#)

[打开项目](#) [范围检查](#)

### 9.3 使用全局符号和局部变量<sup>3</sup> 定义全局符号

使用下列方法之一打开符号表（用SIMATIC模式）或全局变量表（用IEC 1131-3模式）：

- 单击浏览条中的“符号表”按钮。
- 选择**检视 (View) > 符号表 (Symbol Table)** 菜单命令。
- 打开**指令树**中的符号表或全局变量文件夹，然后双击一个表格图标。

您可以使用项目中的[系统符号名](#)。预定义的系统符号表提供常用PLC系统功能的存取功能。PLC系统符号将功能名称与用于激活

霉 8 芳 PLC特殊内存位置相联系。

本标题讨论下列信息：

[在表格中编辑](#)

[在表格中指定符号赋值](#)

[引号的使用](#)

[插入附加行](#)

[使用定义、编辑或选择符号命令](#)

[按照列排序](#)

[检视重叠和未使用的符号](#)

[使用多个表格](#)

[建立附加符号表](#)

[在符号表之间移动](#)

[POU符号标签](#)

[符号和绝对视图](#)

[在符号和绝对地址视图之间切换](#)

[同时检视符号和绝对地址](#)

[理解符号](#)

[理解符号范围](#)

[使用间接记号](#)

[内存类型](#)

[输入错误和故障排除](#)

[提示](#)

[SIMATIC与IEC](#)

[SIMATIC和IEC表格举例](#)

[在符号表 / 全局变量表中指定符号赋值](#)

[返回顶端](#)

欲为地址或常量值指定符号，请遵循下列步骤：

1. 打开符号表 / 全局变量表（使用标题开始部分描述的一个步骤）。
2. 在“符号名”列键入符号名（例如，Input1）。允许使用的最大符号长度为23个字符。使用TAB、ENTER或ARROW键确认，并移至下一个单元格。

**注释：**

\* 在为符号指定地址或常量值之前，该符号一直显示为未定义符号（绿色波浪下划线）。完成“地址”列赋值后，绿色波浪下划线被删除。

\* 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在LAD、FBD和STL程序编辑器窗口中被一个波浪号(~)截断。您可将鼠标放在被截断的名称上，在工具提示中查看全名。

3. 在“地址”列中键入地址或数值（例如，V或I23）。  
（在IEC 1131-3编程模式中，键入地址后会自动增加正确的IEC“%”前缀。）
4. 如果您正在使用IEC全局变量表，在“数据类型”列的下拉列表中选择一种数据类型。  
（SIMATIC用户无需提供数据类型。）
5. 键入注解（选项：最多允许79个字符）。

**注释：**


在STEP 7-Micro/WIN中，您可以建立多个符号表（SIMATIC编程模式）或多个全局变量表（IEC 1131-3编程模式）。但不允许将相同的符号名称多次用作全局符号赋值，在单个符号表中中和几个表内均不得如此。（相反，可允许根据您的选择在多个不同局部变量表中多次使用相同的符号名称。）

[检视重叠和未使用的符号](#)


[返回顶端](#)





欲检视符号表中的“重叠”列或“未使用的符号”列：

- 选择**工具 (Tools) > 选项 (Options)** 菜单项目。
- 选择“符号表”标记。
- 选择适当的复选框（“显示重叠符号”和“显示未使用的符号”）。

“重叠”列显示绝对地址共享部分或全部相同内存之符号行的重叠图标。如果同一个字面值有多个已定义的符号常量，那麽该图标将显示为重叠。每次表格被修改时，“重叠”列被更新。

符号表				
· 3 · 1 · 4 · 1 · 5 · 1 · 6 · 1 · 7 · 1 · 8 · 1				
			符号	地址
1			符号3	VBO
2			符号4	VDO

“未使用的符号”列就在您的程序中未被引用的所有符号显示未用图标。每次表格被修改时，该列被更新。

符号表				
· 3 · 1 · 4 · 1 · 5 · 1 · 6 · 1 · 7 · 1 · 8 · 1				
			符号	地址
1			符号1	I0.0
2			符号2	I0.1

### 引号的使用

[返回顶端](#)

在3.1版之前，全局符号名称必须放在双引号内。双引号已不再用于符号名。自4.0版起，双引号用于在符号表中起始和结束作为撇 琶 持档册ASCII常量字符串。此赋值创建了一个符号ASCII字符串常量。与此相似，单引号用于起始和结束ASCII字符常量（仅限字节、字、双字）以创建一个符号ASCII字符常量。

### 插入附加行

[返回顶端](#)

使用下列方法之一在符号表 / 全局变量表中插入附加行：

- 选择菜单命令 **编辑 (Edit) > 插入 (Insert) > 行 (Row)**。将在符号表 / 全局变量表光标的当前位置上方插入新行。
- 用鼠标右键单击符号表 / 全局变量表中的一个单元格。选择弹出菜单命令 **插入 (Insert) > 行 (Row)**。将在光标的当前位置上方插入新行。
- 欲在符号表底部插入新行，将光标放在最后一行的任意一个单元格中，按“下箭头”键。

### 使用定义、编辑或选择符号命令

[返回顶端](#)

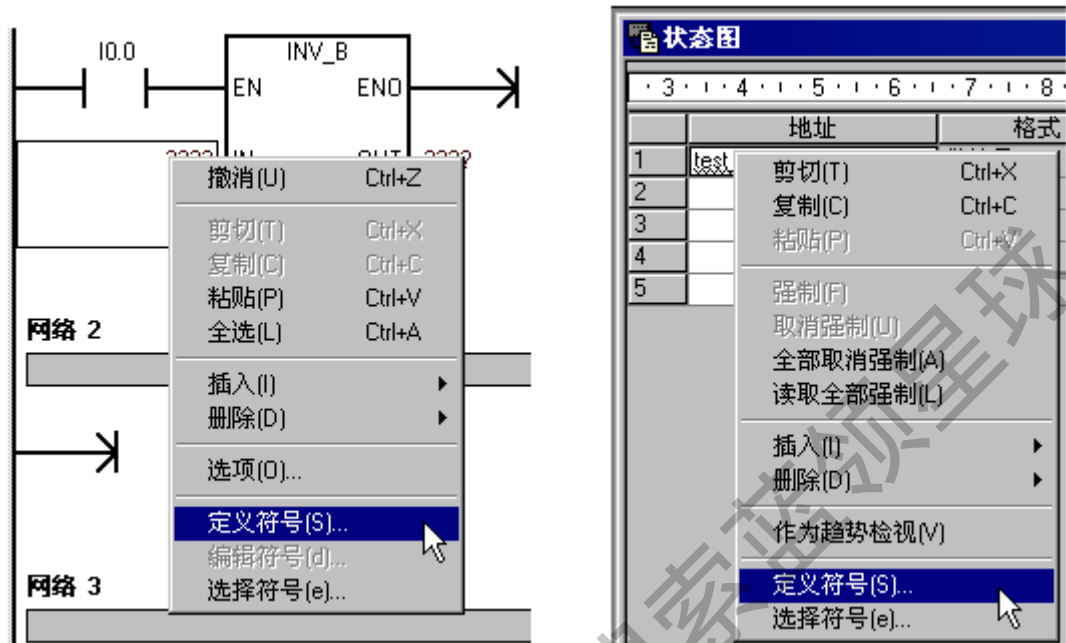
定义、编辑或选择符号等命令，允许您在使用程序编辑器或状态图时，定义新符号、从列表上选取现有符号、或编辑符号属性。新的或修改后的赋值将被自动加入到符号表内。



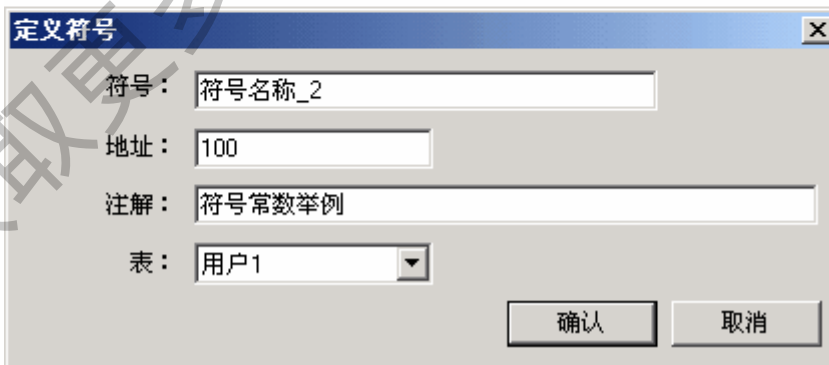
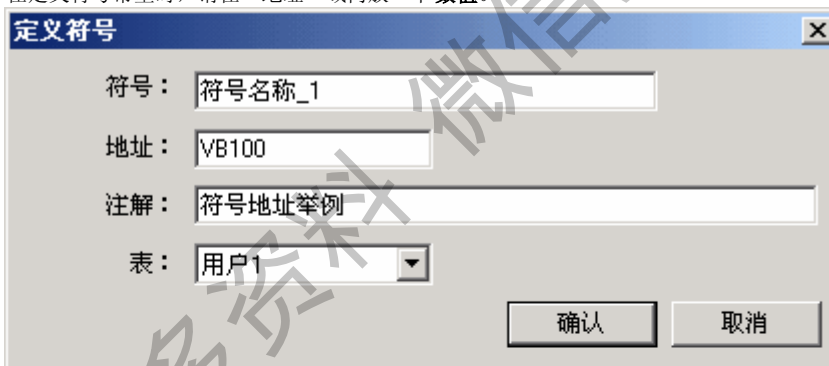
**提示：**只有在“符号编址”视图中工作时才能使用符号命令。（检查“检视”菜单下方的“符号编址”选项：标选符号表示已打开“符号编址”视图。）

欲激活“符号”命令：

1. 用鼠标右键单击对某指令参数或“状态图”地址单元来说不完全（或不正确）的符号名，然后从弹出菜单选择“定义”。



2. 定义新符号，编辑现有符号的属性，或由全部用户指定符号的概括中选取一个现有符号。  
在定义符号地址时，请在“地址”域内放一个地址。  
在定义符号常量时，请在“地址”域内放一个数值。



bml mw\_sym\_define\_b.gif)

3. 单击“确定”确认作业，并关闭对话框，或单击“取消”，取消选择。

**注释:**

- 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在 LAD、FBD 和 STL 程序编辑器窗口中被一个



波浪号 (~) 截断。您可将鼠标放在被截断的名称上，从而在工具提示中查看全名。

### 按照列排序

#### [返回顶端](#)

您可按照“名称”或“地址”列排序表格，排序时可按正向或逆向（字母）顺序排列。此表中的符号常量将在与地址关联的最后一个符号后面列出。所有非字符类型的符号常量接着将独立于显示格式地按数字顺序排列。ASCII常量将出现在数字常量之后，地址 A 拷 罇蟪毓帧

您可以使用检视菜单对某列进行排序。请单击列身以选取该列，请勿单击列顶的列标题（“名称”或“地址”）。

- 若按正向顺序（A至Z）排序列，单击“排序”  按钮，或选择菜单命令 **检视 (View) > 正向排序 (Sort Ascending)**。
- 若按逆向（Z至A）顺序排序列，单击“逆向排序”  按钮，或选择菜单命令 **检视 (View) > 逆向排序 (Sort Descending)**。

若要按相反顺序排序，单击列顶部的列头（“名称”或“地址”）。当您单击列头时，将在正向和逆向排序之间切换。


### 建立附加符号表


#### [返回顶端](#)

默认情况下，符号表窗口为用户定义的符号名称（USR1）显示一个标记。如果您对任何POU重新命名，符号表窗口显示一个被称为“POU符号”的标记—如果您在使用“协议”功能，例如USS协议库，则窗口显示“USS协议”标记。

可采用下列几种方法为用户定义的符号名称建立附加符号表。

- 从指令树用鼠标右键单击“符号表”文件夹，并选择弹出菜单命令 **插入符号表 (Insert Symbol Table)**。
- 打开符号表窗口，并使用“编辑”菜单，或用鼠标右键单击，调出弹出菜单，并选择 **插入 (Insert) > 表格 (Table)**。
- 您在建立程序网络时，可指定符号名，方法是在程序编辑器的指令参数域中键入有效的符号名。此种方法建议一组不


付 讷嫫 卬返姆 琶 5 弋 鳌敖 00 锤持捣 疔恻卑磁 ，将该组符号名传送至您指定地址的新“符号表”标记。欲为符号表重新命名，用鼠标右键单击指令树中的符号表标记，并选择“重新命名”，或双击“符号表”标记名。

**注释：**成功地插入新符号表后，一个新标记  会出现在符号表窗口的底部。在检视符号表时，请核实选择了正确的标记。您可以双击或用鼠标右键单击标记，为标记重新命名。

### 在符号表之间移动

#### [返回顶端](#)

如果建立了多个符号表，您可以使用下列任意一种方法存取表格：

- 如果已经打开符号表窗口，单击位于符号表窗口底部的标记 ，在符号表之间移动。
- 从指令树扩充“符号表”文件夹图标，并双击所需的“符号表”图标，检视符号表窗口中该表的标记。

### POU符号标记

#### [返回顶端](#)

如果您已经为项目的任何POU或其他元件（例如数据块、状态图或符号表）指定符号名，将在符号表窗口的“POU符号”标记中列出符号名赋值。

该标记为只读标记；您不能在此编辑赋值。

如果您希望更改赋值，您必须编辑该元件的“属性”对话框。（从指令树的“项目”分支，用鼠标右键单击元件，调出“属性”对话框）

所有符号赋值必须使用有效语法。（请参阅下面的 [输入错误](#)）。如果指定符号名赋值时违反了指南规定，编译程序时 STEP 7-Micro/WIN 会报告错误。

### 在符号和绝对地址视图之间切换

#### [返回顶端](#)

在符号表 / 全局变量表中建立符号和绝对地址或常量值的关联后，您可在操作数信息的符号和绝对显示之间切换。可采用下列

- 选择菜单命令**检视 (View) > 符号编址 (Symbolic Addressing)**，在符号编址打开或关闭之间切换
- 使用**Ctrl+Y**快捷键在符号编址打开或关闭之间切换

“符号编址”菜单项目前面的标选标记表示已打开符号编址。默认条件下，当您打开第一个项目时，符号编址也被打开。

您不能在检视符号常量或其关联的字面值之间切换。这是因为您可以为同一个字面值定义多个符号常量。因此，假如您可以 ‘  
毛侧 儒A浚瑋TEP 7-  
Micro/WIN无法可靠地恢复原有的符号常量。出于此原因，禁用符号编址（通过主菜单或按CTRL+Y）将不会影响您项目中符号常量的显示。出于同样原因，如果您为某操作数输入了该常量的字面值，STEP 7-Micro/WIN将不会自动套用已定义的符号常量。

## 同时检视符号和绝对地址

[返回顶端](#)

欲在LAD、FBD或STL程序中同时检视符号地址和绝对地址，使用菜单命令**工具 (Tools) > 选项 (Options)**，并选择“程序编辑器”标记。选择“显示符号和地址”。



- 提示：**
- 只有在打开**符号地址视图**时，才能在项目中显示符号地址。否则，即使您选择了“显示符号和地址”，也只会显示绝对地址。
  - 如果您选择了同时显示项目操作数的符号视图和绝对视图，较长的符号名在LAD、FBD和STL程序编辑器窗口中被一个波浪号 (~) 截断。您可将鼠标放在被截断的名称上，在工具提示中查看全名。

## 理解符号范围

[返回顶端](#)

符号表 / 全局变量表将符号名指定给PLC内存或I/O地址。在符号表 / 全局变量表中定义**符号**时，符号具有全局范围。这意味着您可在任何**POU**中使用该符号名称，代表位于该符号地址的数据。（相反，如果您使用局部变量表指定符号名称，则只能在定义该局部变量的POU范围中使用该局部变量。）

## 使用间接记号

[返回顶端](#)

与绝对地址相同，您可在符号中使用间接记号 (& 和 \*)。欲了解有关间接编址的详情，请参阅[直接和间接编址](#)。

## 内存类型

[返回顶端](#)

您可为下列内存类型建立符号名：I、Q、M、SM、AI、AQ、V、S、C、T、HC

## 输入错误

[返回顶端](#)

输入错误用色彩显示，很容易发现：

- 非法语法 — **红色文字**和**红色波浪下划线**

例如：

	符号	地址
1	2name	I0.0
2		VBO
3	Begin	I0.2

（非法地址VBO是正确的）（逆转的关键字无法用作符号）

（第一个字符的号码字符非法）

- 非法使用 — **红色波浪下划线**

例如：

	符号	地址
1	Pump1	I0.0
2	Pump1	I0.0
3	SymConstant	123456
4	SymConstant	123456
5	SymConstant2	654321
6	SymConstant3	654321

(重复名称) (重复地址)

- 未定义符号 — 绿色波浪下划线

例如:

	符号	地址
1	<u>Pump1</u>	
2	Pump2	V0
3	Factor	3.14E-006
4	Neg_Value	-3333333

(名称带有无效地址) SIMATIC模式

(名称带有无效地址或数据类型) IEC模式

**注释:**

- 符号名称可包含某些字母数字字符和下划线, 以及某些扩展字符 (ASCII 128至ASCII 255)。但第一个字符只能是字母或扩展字符。
- 在3.1版之前, 全局符号名称必须放在双引号中。双引号已不再用于符号名。自4.0版起, 双引号用于在符号表中起始和结束作为某符号名之赋值的ASCII常量字符串。此赋值创建了一个符号ASCII字符串常量。与此相似, 单引号用于起始和结束ASCII字符常量 (仅限字节、字、双字) 以创建一个符号ASCII字符常量。
- 不允许使用关键字作为符号名, 也不允许第一个字符是数字或包含非字母数字字符或扩展字符集。
- 符号名称的最大许可长度为23个字符。
- 如果您希尝试建立符号赋值, 以纠正程序中操作数“未定义符号”错误, 但键入地址数值后忘记按 TAB键、ENTER键  
蚣 芳 蹦 禱 爻 纛 蚩 唳 魔 保 僮 魔 廊 幌 允 尽 拔 炊 丁 宸 拧 贝 砦 蟆 N 私 伎 稣 庖 晃 侍 猓 禱 胤 疝 棠 翱 冢 獾 支  
旁 淞 暗 司 贰 钡 卮 襪 校 TAB、ENTER或箭头键, 完成地址赋值。

**建立符号提示**[返回顶端](#)

- 您可在任何时间指定符号赋值: 在建立程序逻辑之前、之间或之后。
- 您可在“注解”列中增加注释, 记录您的符号。
- 编程时, 可使用“定义、编辑或选择符号”命令从列表选择新符号名或建立新符号名。
- 通过单击不同的表列, 可对符号表 / 全局变量表中的数值进行检索。
- 通过拖拉边缘, 可重定列尺寸。
- 您可使用文件 (File) > 打印 (Print) 命令, 并选择适当的复选框, 打印符号表 / 全局变量表。
- 您一旦建立符号赋值, 就可以通过启用符号信息表功能, 在 LAD/ FBD /STL 程序编辑器中的每隔网络滞后显示地址赋值和符号注解。

**SIMATIC和IEC表举例**[返回顶端](#)

符号表 (SIMATIC编程模式) 和全局变量表 (IEC编程模式) 在功能上的唯一区别是“数据类型”列, 此为 IEC编程的独特功能。

IEC-

1131是一种加强类型语言。指令操作数被定义为接受一种指定的数据类型。欲了解有关数据类型的详情, 请参阅 [SIMATIC和IEC 1131-3数据类型](#)。

SIMATIC符号表举例:

符号表				
	符号	地址	注解	
1	动1	I0.0		
2	动2	I0.1		
3	停止1	I0.2		
4	停止2	I0.3		
5	高等级	I0.4		
6	低等级	I0.5		
7	复位	I0.7		
8				
9	泵1	Q0.0		
10	泵2	Q0.1		
11	搅拌机电机	Q0.2		
12	蒸阀	Q0.3		
13	排泄阀	Q0.4		
14	排泄泵	Q0.5		
15				
16	达到高等级	M0.1		
17	搅拌机计时	T37		
18	循环计数	C30		
19	Mix_Time_Minutes	15.3		
20	Mix_Temp_Cent_min	90		
21	Mix_Temp_Cent_max	95		
22	Process_ID	"P418369B"		

USER1 / POU Symbols

IEC全局变量表举例：



符号	地址	数据类型	注解
1	Start_1	%I0.0	BOOL
2	Start_2	%I0.1	BOOL
3	Stop_1	%I0.2	BOOL
4	Stop_2	%I0.3	BOOL
5	High_Level	%I0.4	BOOL
6	Low_Level	%I0.5	BOOL
7	Reset	%I0.7	BOOL
8			
9	Pump_1	%Q0.0	BOOL
10	Pump_2	%Q0.1	BOOL
11	Mixer_Motor	%Q0.2	BOOL
12	Steam_Valve	%Q0.3	BOOL
13	Drain_Valve	%Q0.4	BOOL
14	Drain_Pump	%Q0.5	BOOL
15			
16	High_Lev_Reached	%M0.0	BOOL
17	Mix_Timer	%T37	TON
18	Cycle_Counter	%C30	CTU
19	Counter_Out	%M0.2	BOOL
20	Counter_Current	%M\W3	INT
21	Mix_Time_Minutes	15.3	REAL
22	Mix_Temp_Cent_min	9	BYTE
23	Mix_Temp_Cent_max	95	BYTE
24	Process_ID	"P418369B"	STRING

[返回顶端](#)

另请参阅:

[局部变量表](#)

[检视符号编址](#)

[网络符号信息表 \(LAD/FBD\)](#)

[编址概述](#)

[直接和间接编址](#)

[关键字列表](#)

[系统符号表](#)

[符号表标记 \(工具 > 选项\)](#)

## 9.4 显示每个网络的符号信息表

使用下列方法之一检视或隐藏LAD/FBD/STL程序编辑器窗口中的符号信息表:

- 单击  工具条按钮。
- 选择菜单命令 **检视 (View) > 符号信息表 (Symbol Information Table)**。
- 按 **Ctrl+T** 快捷键组合。
- 单击 **在项目中应用全部符号**  工具条按钮。应用全部符号命令不仅用所有的新、旧和修改符号名更新项目，而且在符号信息表打开和关闭之间切换。

您一旦选择了 **检视 (View) > 元件 (Component) > 符号表 (Symbol Table)** 菜单项目，所有打开的项目均会显示网络符号信息，直至您取消勾选菜单项目。

当您检视符号信息表时，符号名、绝对地址、数据类型和注解会在 LAD/FBD/STL程序中每个网络下方按字母顺序显示。未定义

的符号名也包括在内。

该表列出该网络中所有符号的信息；不包含全局符号的网络不显示符号信息表。所有重复条目均被移除。

如果您位于IEC模式，表格显示一个附加数据类型列。

符号信息表无法编辑。

**网络 1 用涂料成分1和监视器罐填充储罐。**

起始\_1:I0.0      停止\_1:I0.2      高\_位:I0.4      泵\_1:Q0.0

泵\_1:Q0.0

泵_1	Q0.0	泵1控制 (1 = 运行, 0 = 停止)
高_位	I0.4	混合罐料位过高
起始_1	I0.0	泵1起始命令
停止_1	I0.2	泵1停止命令

**网络 2 用涂料成分2和监视器罐填充储罐。**

起始\_2:I0.1      停止\_2:I0.3      高\_位:I0.4      泵\_2:Q0.1

泵\_2:Q0.1

泵_2	Q0.1	泵2控制 (1 = 运行, 0 = 停止)
高_位	I0.4	混合罐料位过高
起始_2	I0.1	泵2起始命令
停止_2	I0.3	泵2停止命令




**提示：**用光标键浏览符号信息。

您可以从列边缘拖动分割器光标，重新确定表格大小。表格不得超过 32 个单元格。所有

无法放入一个单元格中的文本均以波浪号结束；工具提示显示所有无法容纳的文本。

打印程序时，使用“打印”对话框中的“选项”按钮启用符号信息表打印。

为了节省垂直方向的空间并检视更多的程序网络，您可以在符号信息表打开和关闭之间切换，方法是使用**检视 (View) >**

**元件 (Component) > 符号表 (Symbol Table)** 命令，或单击在项目中应用全部符号  工具条按钮。

使用“查找”功能替换文本。

#### 故障排除

如果在LAD/FBD/STL程序网络的下方未显示符号信息表，请考虑下列因素：

- 是否已启用符号信息表？检查“检视”菜单：符号信息表项目旁应当有一个标选符号。

- 网络中是否使用符号（或与符号对应的绝对地址）？不包含符号数值的网络不会显示符号信息表。
- 是否在符号表（IEC全局变量表）中输入了符号？在您完成赋值之前，不会显示地址或注解信息。

另请参阅：

[符号表 / 全局变量表](#)

[符号表标记（工具 > 选项）](#)（在您检视程序状态时，调整符号信息表的显示方法）

[应用程序用户参考手册](#)

[入门指南目录](#)

## 9.5 定义局部变量

本部分讨论下列主题：

[您的程序需要使用局部变量吗？](#)

[理解局部变量](#)

[局部变量的说明类型](#)

[局部变量数据类型检查](#)

[如何隐藏 / 显示局部变量表](#)

[如何在局部变量表中赋值](#)

[局部变量表举例](#)

**您的程序是否需要使用局部变量？**

[返回顶端](#)

使用局部变量有两种原因：

- 您希望建立不引用绝对地址或全局符号的可移动子例行程序。
- 您希望使用临时变量（说明为TEMP的局部变量）进行计算，以便释放PLC内存。

如果以上说明对您的情形不适用，您无须使用局部变量；您可以在符号表（SIMATIC）或全局变量表（IEC）中定义符号数值，  
械姆 攀 刀丁邇 直淞俊

**理解局部变量**

[返回顶端](#)

您可以使用程序编辑器的局部变量表指定对个别子例行程序或中断例行程序独特的变量。

局部变量可用作传递至子例行程序并可用于增加子例行程序的移动性或重新使用子例行程序的参数。

程序中的每个POU都有自己的局部变量表，配备64个字节的L内存。这些局部变量表允许您定义具有范围限制的变量；局部变量  
辉消 8.帽淞康 POU中才有效。相反，在每个POU中均有效的全局符号只能在符号表 / 全局变量表中定义。当您为全局符号和  
植勘淞渴褂孟嗤 姆 芭 保g INPUT1），定义局部变量的POU中的局部定义优先，全局定义用于其他POU。

在局部变量表中赋值时，您指定说明类型（TEMP、IN、IN\_OUT或OUT）和数据类型（请参阅SIMATIC和IEC 1131-  
3数据类型），但不指定内存地址；程序编辑器自动在L内存区中为所有的局部变量指定内存位置。

局部变量表符号地址分配会将一符号名与存储有关数据值的L内存地址关联。局部变量表不支持向符号名直接赋值的符号常量  
(这在符号或全局变量表中是许可的)。



**提示：** PLC不会将局部变量数据值初始为零。您必须在程序逻辑中初始化您使用的局部变量。

**局部变量的说明类型**

[返回顶端](#)

对局部变量赋值指定的类型取决于在其中赋值的POU。主程序（OB1）、中断例行程序和子例行程序可使用临时（TEMP）变量  
V 挥性谥葱铎械保 偈北淞坎趴捎茫 槽葱型甌珊瑚 偈北淞靠杀恢甌础W 永 谱缘蛭部墙褂玫甯貌问 IN、IN\_OUT、OUT）。

**说明类型**

**说明**

IN

调用POU提供的输入参数。

OUT

返回调用POU的输出参数。

IN\_OUT

数值由调用POU提供的参数，由子例行程序修改，然后返回调用POU。

TEMPORARY

临时保存在局部数据堆栈中的临时变量。一旦POU完全执行，临时变量数值则无法再用。在两次POU执行之间，临时变量不保持其数值。

### 局部变量数据类型检查

[返回顶端](#)

将局部变量作为子例行程序参数传递时，在该子例行程序局部变量表中指定的数据类型必须与调用 POU 中数值的数据类型相匹配。

举例：

您从 OB1 调用 SBR0，将称为 INPUT1 的全局符号用作子例行程序的输入参数。

在 SBR0 的局部变量表中，您已经将一个称为 FIRST 的局部变量定义为输入参数。

当 OB1 调用 SBR0 时，INPUT1 数值被传递至 FIRST。

INPUT1 和 FIRST 的数据类型必须匹配。

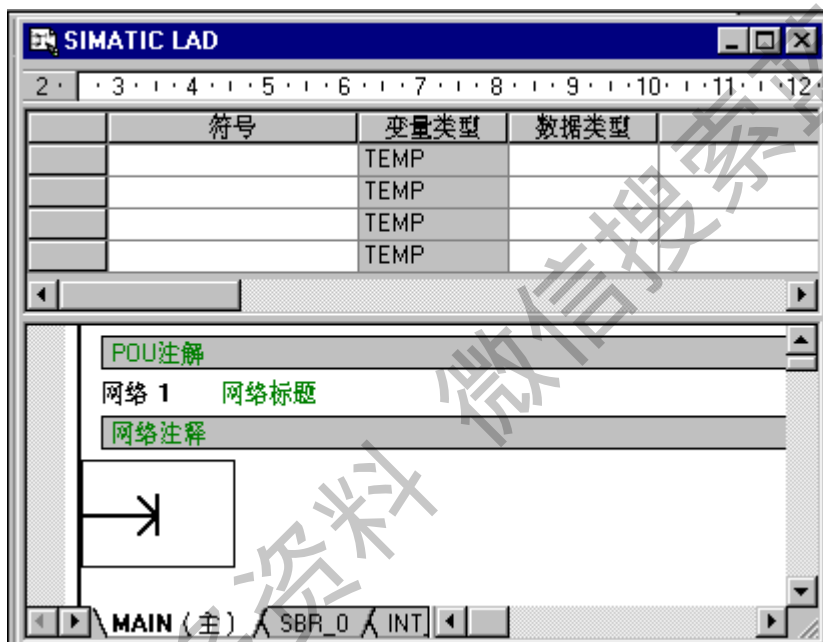
如果 INPUT1 是实数，FIRST 也是实数，则数据类型匹配。如果 INPUT1 是实数，但 FIRST 是整数，则数据类型不匹配，

挥芯勒 苏庖淮砦蝗 缘蚤拍鞞喊舜

### 检视 / 隐藏局部变量表

[返回顶端](#)

如果您将水平分裂条拉至程序编辑器窗口的顶部，则局部变量表不再显示，但仍旧存在。将分裂条下拉即可再次显示局部变量表。



### 如何在局部变量表中赋值

[返回顶端](#)

注释：\*

在程序中使用局部变量之前，在局部变量表中赋值最为有效。在程序中使用符号名时，程序编辑器首先检查适当 POU 的局部变量表，然后检查符号表 / 全局变量表。

如果符号名在这两处均未定义，程序编辑器则将其视为未定义的全局符号；此类符号用绿色波浪状下划线标明。程序编辑器不会自动重新读取局部变量表并对您的程序逻辑做出修正。如果您后来进行了定义该符号名的数据类型分配（在局部变量表中），您必须以手动方式在该符号名前插入一个井号（#），例如：#UndefinedLocalVar（在程序逻辑中）。

\*\* 每个子例行程序调用的输入 / 输出参数的最大限制是 16，如果您尝试下载的程序超过此一限制，会返回一则错误信息。

### 如何输入第一个局部变量赋值

欲在局部变量表中赋值，请遵循以下步骤。

1. 确保正确的 POU 在程序编辑器窗口中显示，如有必要，单击所需的 POU 标记。（因为每个 POU 都有自己的局部变量表）
2. 如果局部变量表处于隐藏状态，下拉水平分裂条，显示局部变量表。（请参阅 [检视 / 隐藏局部变量表](#)。）
3. 为需要定义的变量选择具有正确说明类型的行，在“名称”域中为该变量键入一个名称。（如果您在 OB1 或中断例行程序中赋值，局部变量表只包含 TEMP 变量。如果您在子例行程序中赋值，局部变量表包含 IN、IN\_OUT、OUT 和 TEMP 变量。）您不需要在局部变量表中的变量名前加井号。井号只用在程序代码中的局部变量前。

注释：

- \* 局部变量名最多可包含23个字母数字字符和下划线，也可包含扩展字符（ASCII 128至 ASCII 255）。第一个字符只能是字母或扩展字符。将关键字用作符号名属于非法，名称的第一个字符是数字，或名称包含非字母数字字符。
  - \*\* 局部变量表变量名被下载并存储于CPU内存中，使用较长的变量名可能减少用于存储程序的内存空间。
4. 在“数据类型”域中单击鼠标指针，并使用列表框为局部变量选择适当的数据类型。

**注释：**

当您为局部变量指定为子例行程序的参数时，您必须保证为局部变量指定的数据类型不与子例行程序调用中使用的操作数发生冲突。（请参阅上文中的[数据类型检查](#)。）

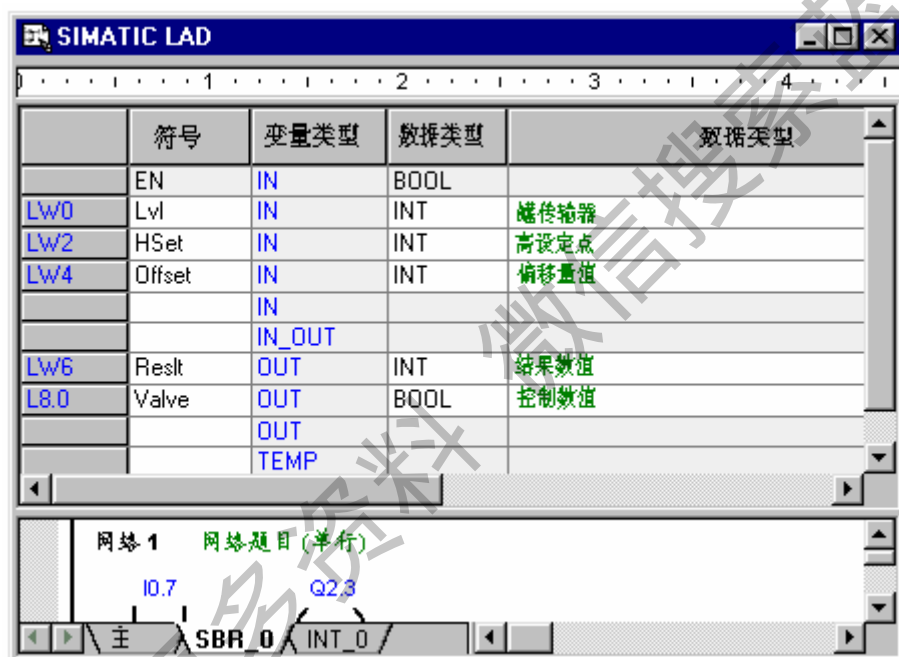
为“名称”和“数据类型”域提供数值后，程序编辑器会自动为局部变量指定L内存地址。

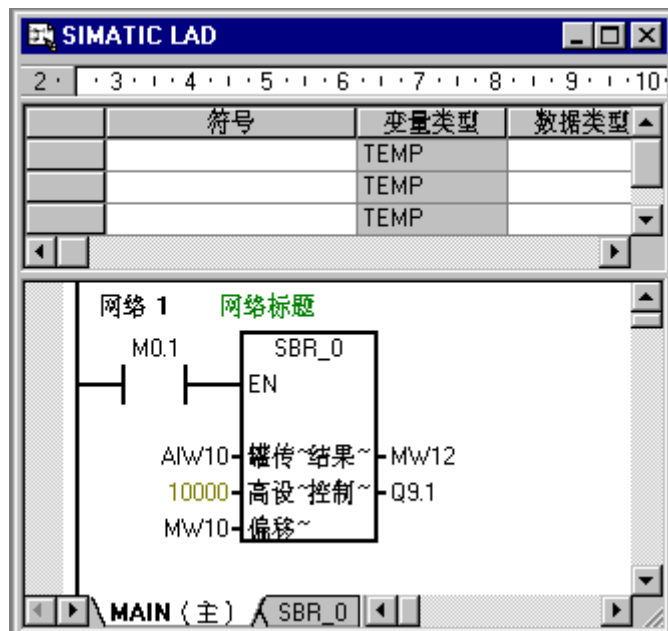
**如何在局部变量表中输入附加赋值**

对于OB1和中断例行程序，局部变量表显示一组已被预先定义为TEMP变量的行。在OB1或中断例行程序中，只能使用这种说明类型。欲在表中增加更多行，只需单击最后一行中的一个单元格，然后使用ENTER键移过该行并向下移动，即自动生成一个新行。对于子例行程序，局部变量表显示按照以下顺序预先定义说明类型的一组行：IN、IN\_OUT、OUT和TEMP。您不能改变该顺序。局部变量在该表中的顺序必须符合当您为子例行程序调用指令进行操作数赋值时对应的操作数顺序。如果您希望增加附加局部变量，必须用鼠标右键单击现有行，并使用弹出菜单插入与单击行类型相同的另一局部变量。选择**插入 (Insert) > 行 (Row)**，在所选行的上方插入新行，或选择**插入 (Insert) > 行下方 (Below Row)**，在所选行下方插入新行。

**局部变量表举例**

[返回顶端](#)





另请参阅：

子例行程序  
SIMATIC和IEC 1131-3数据类型  
关键字

## 9.6 PLC设置

### 9.6.1 读取PLC信息

欲检视PLC信息，例如PLC型号和版本号、操作模式、扫描速率、I/O模块配置、CPU和I/O模块错误及事件历史日志，选择菜单命令**PLC (PLC) > 信息 (Information)**。

如果您的CPU上连着智能模块，要检视其信息，请选取合适的模块列表项，然后单击EM信息按钮。如此就会出现一个单独的对话框。该窗口的标题为“模块信息”。

**重设扫描速率**按钮允许您刷新三个“扫描速率”域中的次数。

如果您的PLC类型支持“事件历史”日志，该按钮就会被启用。请单击“事件历史”按钮检视通电、模式转换及严重错误的日志。

#### DP标准协议状态（仅用于DP端口通讯启用的按钮）

- DP状态：端口的当前状态（初始化，未初始化等）。
- 主地址：控制CPU的主设备地址。
- 输出缓冲器地址：用于从主设备传输信息的起始V内存地址。
- 输出字节数目：配置用于从主设备接收传输存储的字节数目。
- 输入字节数目：配置用于传输至主设备数据存储的字节数目。

#### 通讯端口状态


- 端口0地址：为端口0指定的地址。
- 端口1地址：为端口1指定的地址。

### 9.6.2 选择目标PLC类型（型号和版本）

如欲使用“PLC类型”对话框，请选择**PLC (PLC) >**

**类型 (Type)** 菜单命令。您也可以打开**指令树**中的CPU节点，或者用鼠标右键单击CPU节点并由弹出菜单中选取**PLC类型**命令。



通过选择PLC类型，您启用STEP 7-Micro/WIN，执行指令和参数检查，可帮助防止在建立程序时发生错误。此外，在您为项目指定 PLC类型后，指令树用红色标记 x:  表示对PLC无效的任何指令。

如果您打开一个新项目，没有进行 PLC类型选择即开始编辑程序，STEP 7-Micro/WIN编辑器和编译程序允许您对并非在所有的 S7-200

PLC型号中均受支持的指令、地址和 PLC功能进行编程。如果您在不受目标 PLC支持的项目中使用指令、地址或 PLC功能，当 (6) 韵略回乍 渴保 岂 PLC拒绝。

本标题讨论下列主题：

[指定PLC类型选择](#)

[设置默认PLC类型](#)

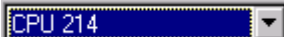
[让STEP 7-Micro/WIN读取远程PLC类型](#)

[类型不匹配](#)

### 指定PLC 类型选择

[返回顶端](#)

从“PLC类型”对话框中的列表框选择一种 PLC。例如，如果您有 PLC 214，列表框则显示：



作这项选择时，PLC 214最新固件版本的范围和功能限制被应用。但是，您可能使用的是较早的 PLC 214固件版本，它们不支持最新 PLC

214固件版本中提供的功能。为了确保在执行范围检查时考虑到您的 PLC型号和您的固件版本，您可以让 STEP 7-Micro/WIN 3从PLC直接读取信息。

**注释：**

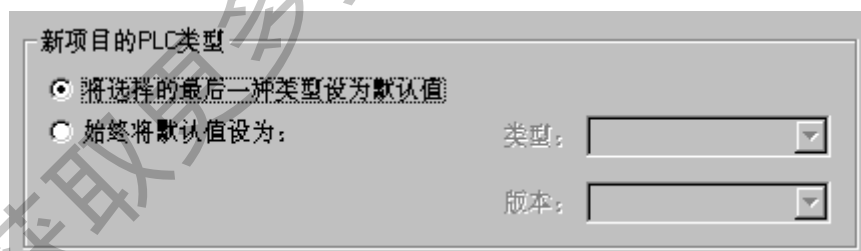
当您指定 PLC类型选择时，不需要配置通讯；如果您希望 STEP 7-Micro/WIN为您读取远程 PLC类型，您只需建立通讯。

### 设置默认PLC类型

[返回顶端](#)

您可以选择让新项目始终使用最后选择的 PLC类型，或者让新项目始终将默认值设为指定 PLC类型。选择 **工具 (Tools) > 选项 (Options)** 并单击“项目”标记，设置默认 PLC类型。

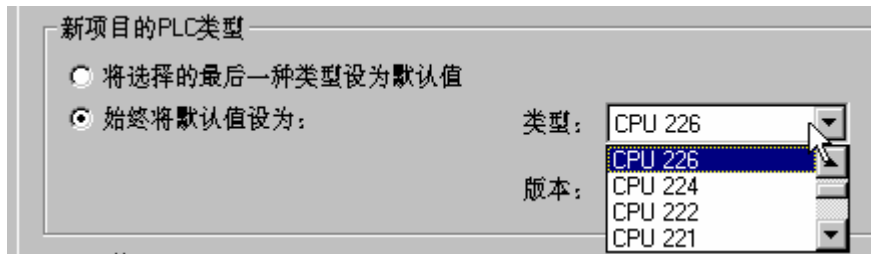
- 如果您选择将默认值设为最后选择的 PLC类型，您每次建立新项目时，该项目的 PLC类型均与最后一个项目的 PLC类型相同。



- 如果您选择“始终将默认值设为：”选项，建立的每个新项目均设为您从下拉列表中选择的 PLC类型。当前 PLC类型包含在该列表中。如果当前 PLC类型属于具体版本（使用“读取 PLC”功能设置），您还可以选择让每个新项目使用该 PLC 类型。


**注释：**


在新安装 STEP 7-Micro/WIN 时，默认选项被设为“始终将默认值设为：CPU 221”。



### 让STEP 7-Micro/WIN读取远程PLC类型

[返回顶端](#)

欲使STEP 7-Micro/WIN读取PLC型号和固件版本，单击“PLC类型”对话框中的  按钮。

**注释：**您必须已经成功地建立通讯才能读取PLC型号和固件信息。您可以使用“PLC类型”对话框中的  按钮，建立与PLC的通讯，或者在有多个PLC与通讯网络连接时选择一个目标PLC。



### 类型不匹配

[返回顶端](#)

如果您尝试向PLC下载程序，但PLC与项目的PLC类型选择不匹配，会出现错误讯息。您的选项取决于不匹配的类型

项目选择	实际PLC类型	您的选项
CPU 21x 类型 CPU 22x 类型 单击“将类型设为远程PLC”，修改选择	CPU 21x 类型 CPU 22x 类型	因为两个PLC版本均为同一代产品。 您可以不修改PLC类型选择下载，或
CPU 21x 类型 单击“将类型设为远程PLC”，修改选择。	CPU 22x 类型	您可以不修改PLC类型选择下载，或
CPU 22x 类型 远程PLC”，修改选择。	CPU 21x 类型	您必须取消下载，或单击“将类型设为

### 另请参阅：

[通讯编辑器](#)  
[PLC内存地址范围](#)  
[指令支持S7-200 CPU](#)  
[中断事件优先级列表](#)

## 9.6.3 使用系统块配置PLC

系统块可用于配置S7-200 CPU选项。  
使用下列方法之一检视和编辑系统块，设置CPU选项：

- 单击 [浏览条](#) 中的“系统块”  按钮。
- 选择 **检视 (View) > 元件 (Components) > 系统块 (System Block)** 菜单命令。
- 打开 [指令树](#) 中的‘系统块’文件夹，然后打开某配置页。 

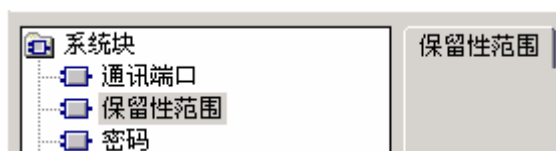
当项目的CPU类型和版本能够支持特定选项时，这些系统块配置选项将被启用。

在下载或上载系统块之前，您必须成功地建立PC（STEP 7-Micro/WIN的位置）与CPU之间的通讯。

- 然后即可下载一个修改的系统块，以便为CPU提供新系统配置。
- 您也可以从CPU上载一个现有系统块，以便使STEP 7-Micro/WIN项目配置与CPU相匹配。

单击系统块树上分支即可修改项目配置。





下面列出全部系统块配置。

[端口](#)  
[保留范围](#)  
[密码](#)

**输出表**  
[数字输出表](#)  
[模拟输出表](#)

**输入过滤器**  
[数字输入过滤器](#)  
[模拟输入过滤器](#)  
[脉冲捕获位](#)  
[背景时间](#)  
[EM \(扩充模块\) 配置](#)  
[配置LED](#)  
[扩大内存](#)

**另请参阅:**  
[编译程序](#)  
[应用程序用户参考手册](#)  
[入门指南目录](#)

## 9.6.4 设置当日时钟时间

欲检视或设置存储在PLC中的当前时间和数据，选择菜单命令 **PLC (PLC) > 当日时间时钟 (Time of Day Clock)**。

您可以设置“当日时间时钟”对话框中的时间和日期。PLC型号214、215、216、221、222、224和226支持“当日时间时钟”及TODR/TODW程序指令。

CPU 222、CPU 224、CPU 226 (2.0版或更高版本) 支持夏时制时间的自动调整。

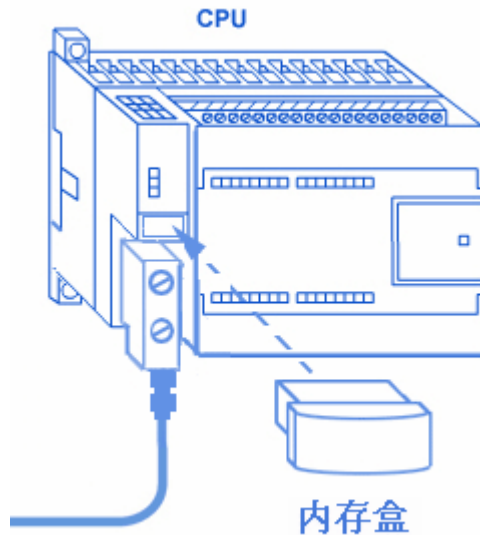
**另请参阅:**  
[TODR/TODW程序指令](#)

## 9.6.5 将PLC与PC内存中的打开项目进行比较

欲将STEP 7-Micro/WIN项目元件与PLC进行比较，选择菜单命令 **PLC (PLC) > 比较 (Compare)**。

请单击有关复选框，选取要比较的项目元件。在STEP 7-Micro/WIN程序编辑器中当前打开的项目与CPU中包含的项目之间进行比较。

## 9.6.6 为内存盒编程



欲为程序和项目数据提供可移动EEPROM存储的供选用内存盒编程，选择菜单命令**PLC (PLC) > 为内存盒编程 (Program Memory Cartridge)**。

在内存盒中存储下列元素：

- 用户程序
- 存储在永久V内存区的数据
- 系统块（PLC配置）

此外，当与64千字节或256千字节内存盒一起使用时，2.0版或更高版本的CPU 224、CPU 224XP及CPU 226能够存储下列内容：

- 菜谱和数据日志配置
- 作为执行数据日志指令之结果而存储的数据
- 用户文件 - （用户文件用S7 Explorer工具而不是‘编程内存盒’命令载入）

单击选项按钮

选项 ▾

以显示或设置下载选项。‘上载下载管理器’窗口可帮助您控制上载、下载、清除及编程内存盒。

[UDM - 上载下载管理器详细说明](#)

### 复制至内存盒

只有在PLC处于电源开启状态、位于STOP（停止）模式并且已安装内存盒时，才能将程序从PLC RAM复制至内存盒。



### 当心

静电放电可能损坏内存盒或PLC中的插座。处理磁带时，与接地导电板接触和 / 或佩戴接地腕带。在导电容器中存放磁带。

您可以在PLC位于电源开启状态时安装或移除内存盒。欲安装内存盒，从PLC中取下塑料罩并将内存盒插入PLC。（内存盒带有怕光 V 3 0 沧啊 # 上注婧邪沧巴甌珊媳 湊障铝肋街趟粗警缘颀

1. 如果程序尚未被下载至PLC，下载程序。
2. 使用菜单命令**PLC > 为内存盒编程 (Program Memory Cartridge)**将程序复制至内存盒。

### 使用内存盒恢复程序和内存

欲从内存盒向PLC传输程序，您必须对装有内存盒的PLC执行电源循环。电源循环后，PLC执行下列任务（安装内存盒时）：

1. 清除RAM
2. 内存盒的内容被复制至PLC RAM
3. 用户程序、系统块（CPU配置）和V内存被复制至永久EEPROM

### 注释：

为装有空白内存盒或在不同型号的PLC中编程内存盒的PLC开启电源时会引起错误。由较早CPU型号编程的内存盒可被较新的CPU型号读取。由较新型号编程的内存盒无法被较早的型号读取，并会被拒绝。

移除内存盒，再次开启电源。然后即可插入内存盒并开始编程。

另请参阅：

[下载 \(文件 > 下载\)](#)

## 9.6.7 PLC电源开启重设



欲从PLC清除严重错误并返回RUN（运行）模式，选择菜单命令**PLC (PLC) > 电源启动重设 (Power-Up Reset)**。如果操作PLC存在严重错误，将SF（系统错误）指示灯设为打开，程序停止执行。您必须将PLC模式重设为STOP（停止），然后再设置为RUN（运行），才能清除错误。可远程使用**PLC (PLC) > 电源启动重设 (Power-Up Reset)** 命令或以手动方式重设PLC模式开关完成此项任务。

## 9.7 上载和下载程序

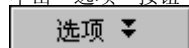
### 9.7.1 上载

可使用下列一种方法从PLC将项目元件上载至STEP 7-Micro/WIN程序编辑器：

- 单击“上载”  按钮。
- 选择菜单命令**文件 (File) > 下载 (Upload)**。
- 按快捷键组合**Ctrl+U**。

欲上载（PLC至编辑器），PLC通讯必须正常运行。确保网络硬件和PLC接头电缆正常操作。选择想要的块（程序块、数据块或低晨棕  显氏某缘蛟  复制到当前打开的项目；于是您就可保存已上载的程序。

单击“选项”按钮



，显示和设置上载操作选项。‘上载下载管理器’窗口可帮助您控制上载、下载、清除及编程内存盒操作。

[UDM - 上载下载管理器详细说明](#)

另请参阅：

[密码](#)

[如何从PLC上载 \(GS 8.3\)](#)

[通讯概述 \(GS 6.1\)](#)

[如何测试通讯网络 \(GS 6.2\)](#)

[如何下载程序 \(GS 6.3\)](#)

[下载 \(文件 > 下载\)](#)

[在运行模式中进行程序编辑](#)

[应用程序用户参考手册](#)

[入门指南目录](#)

### 9.7.2 下载

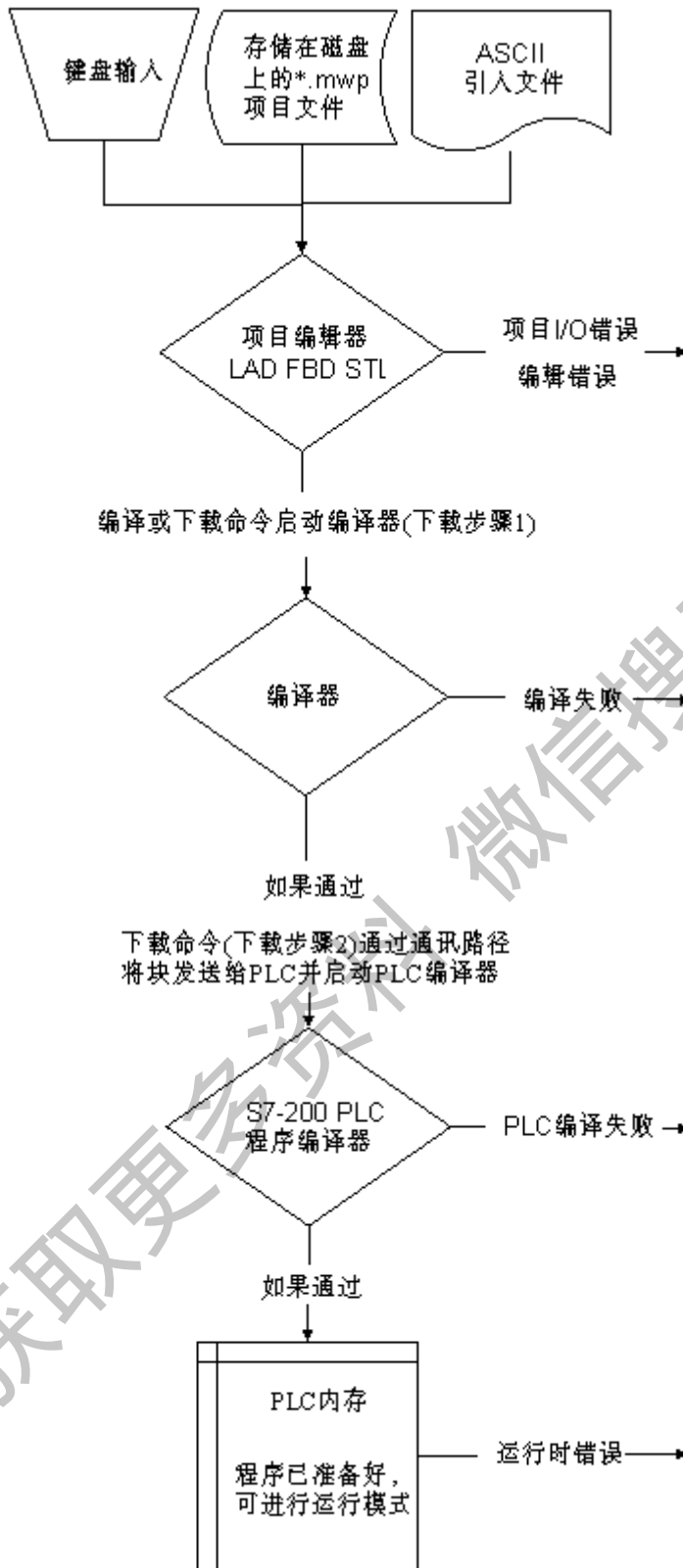
可使用下列一种方法从STEP 7-Micro/WIN向PLC下载项目元件：

- 单击“下载”  按钮。
- 选择菜单命令**文件 (File) > 下载 (Download)**。
- 按**Ctrl+D**快捷键组合。



单击“选项”按钮，显示和设置下载操作选项。‘上载下载管理器’窗口可帮助您控制上载、下载、清除及

[UDM - 上载下载管理器详细说明](#)

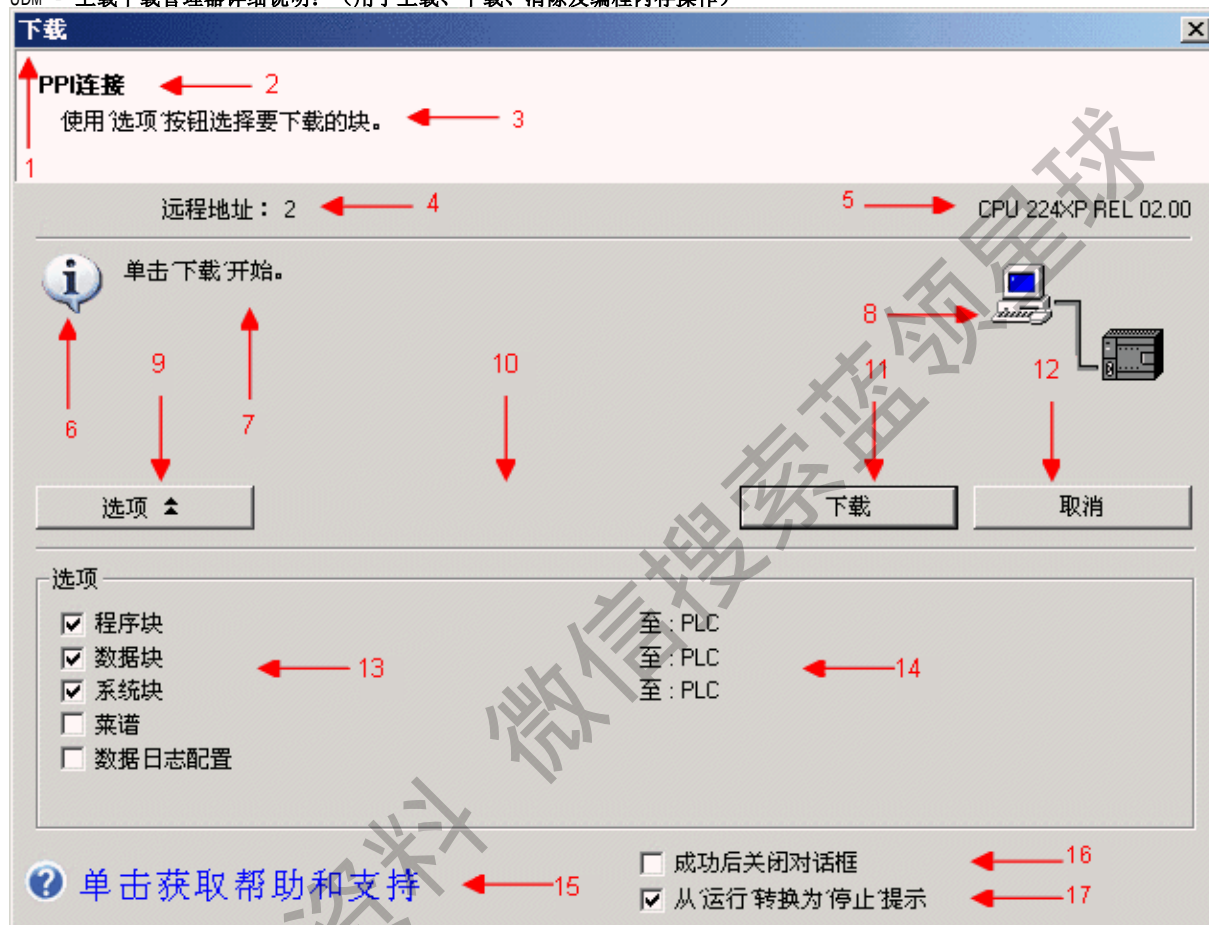


设置PLC类型 项目文件I/O错误 LAD/FBD编辑器错误 全部STEP 7-Micro/WIN编译器错误均列在输出窗口中。双击错误，编辑器滚动至错误位置。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译器。 **通讯错误** 欲下载（编辑器至PLC）或上载（PLC至编辑器），PLC通讯必须正常运行。确保网络硬件和PLC接头电缆均正常操作。 PLC编译器确认PLC硬件支持全部程序指令、范围和结构。

[打开项目](#) [范围检查](#)

使用菜单命令 **PLC > 信息** 查看发现的第一条编译错误。使用菜单命令 **PLC > 类型** 为具体PLC型号和版本设置编译程序，这样会将尽可能多的错误移至输出窗口，易于找到错误位置和改正错误。**严重错误**和**非严重运行时间编程故障**由菜单命令 **PLC > 信息** 报告。

UDM - 上下载管理器详细说明：（用于上载、下载、清除及编程内存操作）



1. [标题](#) - 将进行的操作
2. [连接类型](#) - 通讯连接类型
3. [操作说明与指令](#) - 进程说明与指示
4. [CPU地址](#) - 通讯地址
5. [CPU类型](#) - 目标CPU的版本和类型
6. [状态图标](#) - 表明进程的总体状态
7. [状态讯息区](#) - 显示错误和警告的区域
8. [动画图标](#) - 表明操作处理是否在进行中
9. [选项按钮](#) - 基于用户的优先选择隐藏或显示选项选择
10. [更改项目/通讯按钮](#) - 更改项目的CPU类型以与实际CPU相符，也允许通讯配置
11. [动作按钮](#) - 如果没有错误存在就开始操作
12. [取消按钮](#) - 在下一个停止点取消正在进行的操作
13. [Option selections](#) - 选取选项
14. [Option status ?](#) 选定的选项状态
15. [帮助](#) - 打开MicroWin帮助（与按F1键相同）
16. [成功操作后关闭对话框](#) - 用于在成功操作后关闭此窗口的选项
17. [从“运行”转换为“停止”提示](#) - 当CPU从运行模式转换为停止模式时省略用户提示的选项。

另请参阅：

[通讯概述 \(GS 6.1\)](#)

[如何测试通讯网络 \(GS 6.2\)](#)

[如何下载程序 \(GS 6.3\)](#)

[在运行模式中进行程序编辑](#)

[上载（文件 > 上载）](#)  
[如何从PLC上载（GS 8.3）](#)

[STEP 7-Micro/WIN编译程序错误](#)  
[如何纠正编译错误和下载错误（GS 6.4）](#)

[应用程序用户参考手册](#)  
[入门指南目录](#)

## 9.7.3 清除PLC内存

欲清除PLC内存的指定区域，选择菜单命令 **PLC (PLC) > 清除 (Clear)**（您必须首先将PLC设为STOP（停止）模式。）



### 警告

清除PLC内存影响数字和模拟输出的状态。工厂默认值是在PLC位于STOP（停止）模式时，将数字输出重设为零（关闭）。但是，如果您已经使用系统块的“输出表”标记定义数字输出的“安全状态”，您需要注意删除系统块会删除该“安全状态”定义。下一次您将PLC从RUN（运行）模式转入STOP（停止）模式时，数字输出会被设为零（工厂默认值）。此外，当您删除程序块时，模拟输出被冻结为当前值。直至您下载新的程序块，对模拟输出状态进行更改的唯一方法是使用状态图。清除PLC内存时，如果S7-200 PLC与设备相连，对数字输出状态的更改可传输至装置。而且，您删除程序块后，因为输出被“冻结”，与模拟输出相连的爸梦ø凑罩付姆绞接 答。如果清除了PLC内存，却没有仔细考虑对数字和模拟输出的影响，设备可能按照无法预料的方式操作，可能导致人员死亡或严重伤害及/或设备损坏。始终遵循正确的安全预防措施，在清除PLC内存之前确保程序处于安全状态。

单击选项按钮  以显示或设置下载选项。‘上载下载管理器’窗口可帮助您控制上载、下载、清除及编程内存威胁。

[UDM - 上载下载管理器详细说明](#)

**注释：**清除PLC内存时，当日时间时钟不变。

### 忘记PLC密码怎么办

如果您忘记了PLC密码，则必须清除PLC内存并重新载入程序。清除PLC内存使PLC处于STOP（停止）模式，并将PLC重设为工厂默认值，但PLC地址、波特率和当日时间时钟除外。

欲清除PLC中的程序：

1. 欲清除PLC中的程序，选择菜单命令 **PLC > 清除.. (Clear)**，显示“清除”对话框。
2. 选择所有的复选框，并单击“确定”按钮，核实执行的步骤。
3. 如果已经配置密码，**STEP 7-Micro/WIN**会显示一个密码授权对话框。欲清除密码，在密码授权对话框中输入 **CLEARPLC**，继续执行“全部清除”操作。（**CLEARPLC** 密码不区分大小写。）

“全部清除”操作不从内存盒移除程序。因为内存盒存储密码和程序，您必须对内存盒重新编程才能删除丢失的密码。

另请参阅：

[密码—系统块配置](#)

[应用程序用户参考手册](#)  
[入门指南目录](#)

## 9.7.4 解释错误讯息

[STEP 7-Micro/WIN编译程序错误](#)  
[通讯错误](#)  
[LAD/FBD编辑程序错误](#)  
[项目文件I/O错误](#)

[PLC严重错误](#)  
[PLC编译及运行时间错误](#)  
[PLC非严重错误内存位置](#)

[时间戳记不匹配](#)

[EM241调制解调器模块错误](#)

[EM253位置模块错误](#)  
[CP243-1以太网 / 互联网模块错误和ETHx\\_CTRL](#)  
[CP243-1信道和ETHx-XFR/ETHx\\_CFG错误](#)  
[CP243-1 IT ETHx\\_EMAIL错误](#)  
[CP243-1 IT ETHx\\_FTPC错误](#)  
[CP243-2 AS-i 模块错误](#)

[USS协议指令错误](#)  
[Modbus协议指令错误](#)

## 9.8 使用向导

### 9.8.1 HSC指令向导

欲使用HSC向导：  
选择菜单命令工具 (I) >



指令向导...，然后选取HSC；或单击浏览条中的指令向导图标，然后选取HSC；或打开指令树中的“向导”文件夹并单击 HSC 向导。

#### 步骤

- 1 [选择计数器类型和操作模式](#)
- 2 [指定初始参数](#)
- 3 [程序中中断事件 / 多个步骤](#)
- 4 [生成代码](#)

LAD或STL中的“高速计数器”指令（HSC）由S7-200提供，控制速率超过PLC扫描速率的事件。HSC操作十分灵活，有多种确定计数器行为方式的初始化选项。欲以适当的方式配置HSC，您需要在SM内存中设置几个参数，为操作模式配置一个具体的HSC，将中断例行程序附加在具体的计数器事件上，最后启用计数器操作。

**注释：**在使用HSC向导之前，必须编译程序。如果您尚未编译程序，向导在HSC配置进程开始时提示您编译程序。

HSC向导向您查询初始化选项，然后为指定的配置生成程序代码和数据块代码。从第一个屏幕选择HSC后，向导向您

#### 步骤 1 选择计数器类型和操作模式

从列表框（依赖PLC）选择须配置的计数器。  
从列表框选择操作模式。您的计数器选择决定可用的模式。

#### 步骤 2 指定初始参数

向导为子例行程序指定一个默认名称。您可以指定一个不同的名称，但请勿使用现有子例行程序名称。  
为计数器CV和PV指定一个双字地址、全局符号或整数文字。

**注释：**如果您为PV或CV输入一个未定义的符号，则在定义符号之前无法执行步骤3。向导会提示您在“定义符号”对话框中定义符号。

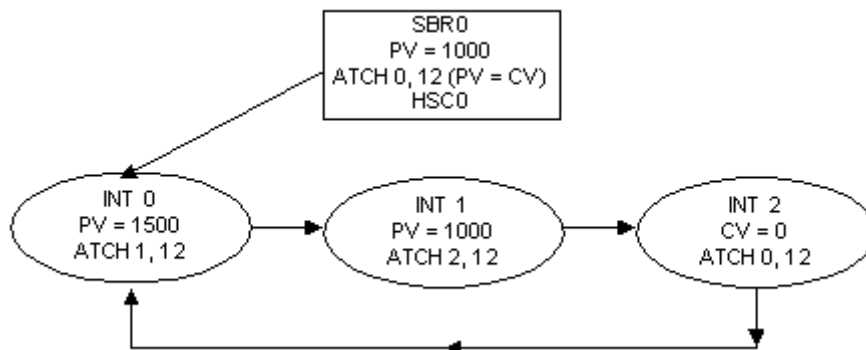
当您为PV和CV选择地址或符号时，更新PV和CV的选项被禁用。您必须修改生成的代码，才能修改或传递数值。根据定义，这意味着代码是可重新载入。  
指定初始计数方向。

#### 步骤 3 程序中中断事件 / 程序多种步骤

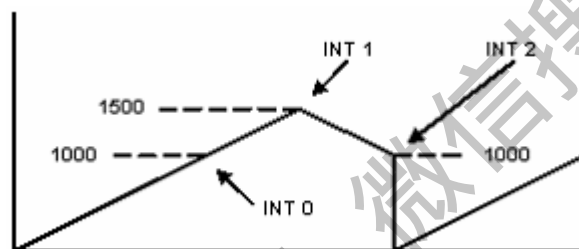
您的计数器类型和操作模式选择决定可用的[中断事件](#)。

当您选择对[当前数值等于预设值事件（CV = PV）](#)进行编程时，向导允许您指定多种步骤计数器执行。  
为了理解多种步骤HSC执行，请考虑下列内容：

在已编程的中断例行程序中，您可以选择将相同的中断事件重新连接到另一个中断例行程序。这将有效地在计数器操作过程中更改HSC的一些[动态参数](#)，并将一个不同的中断例行程序连接到相同的中断事件。最后一个中断例行程序连接到相同的中断事件。最后一个是每个步骤的说明。



- SBR 0** 该子例行程序包含计数器初始化。计数器的当前值被指定为 0 (CV = 0)。计数器的预设值被指定为 1000 (PV = 1000)。计数方向为 UP (向上)。事件 12 (HSC0 CV = PV) 被连接至 INT 0，计数器被启动。
- INT 0** 当计数器达到第一个预设值 1000 时，执行 INT 0。计数器预设值被更改为 1500，方向不变。事件 12 (HSC0 CV = PV) 被重新连接至 INT 1，计数器被重新启动。
- INT 1** 计数器再次达到预设值 (1500) 时，执行 INT 1。此时，我们将预设值更改成 1000 (PV = 1000)，将计数方向更改为 DOWN (向下)，将 INT 1 连接至事件 12 (HSC0 CV = PV)，并重新启动计数器。
- INT 2** 当计数器向下计数至预设值 1000 时，执行 INT 2。此时，我们将当前值设为 0 (CV = 0)，并将计数方向更改为 UP (向上)。事件 12 被重新连接至 INT 0，至此则完成了计数器操作环路。下图显示计数器当前值随时间的变化。每个 (CV = PV) 中断事件均标有为该事件调用的 INT 例行程序。



时间 ->

#### 步骤 4 生成代码

完成 HSC 参数配置后，您可以检查计数器使用的子例行程序 / 中断例行程序列表。该列表仅供参考。

单击“完成”按钮，允许向导为 HSC 生成必要的程序代码。代码包括由用于高速计数器的初始化代码组成的子例行程序。

**注释：**欲启用计数器操作，您必须从主程序中将 [调用](#) 放置在包含初始化代码的子例行程序中。使用 SM0.1 或边缘触发指令。

另请参阅：  
[理解高速计数器指令](#)  
[HSC 指令](#)  
[HDEF 指令](#)

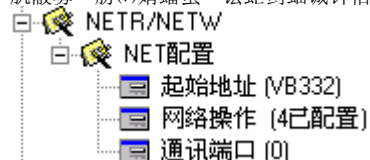


## 9.8.2 NETR/NETW指令向导

欲使用NETR/NETW向导：  
选择菜单命令工具 (I) >



指令向导...，然后选取NETR/NETW；或单击浏览条中的指令向导图标，然后选取NETR/NETW；或打开指令树中的“向



**注释：**本向导仅限于第二代CPU。如果您在一个3.1x项目中有一个向导配置，项目依然工作。但是，向导不会识别该项目，您

### 步骤

- 1 指定您需要的网络操作数目
- 2 指定端口号码和子例行程序名称
- 3 指定网络操作
- 4 分配V内存
- 5 生成代码

为了帮助与网络连接PLC之间的数据交换，S7-200支持网络读取 (NETR) 和网络写入 (NETW) 指令。NETR指令从远程PLC中的指定地址读取一定数量的配置数据。NETW指令向远程PLC中的指定地址写入一定数量的配置数据。NETR和NETW指令操作由数据表中的前七个字节控制。表

**注释：**在使用HSC向导之前，程序必须被编译程序，且位于符号编址模式。如果您尚未编译程序，向导在HSC配置进程开始时提示您编译程序。向导将向您查询初始化选项，然后为指定配置生成程序代码和数据块代码。从第一个屏幕选择NETR/NETW后，向导

### 步骤 1 指定您需要的网络操作数目

**注释：**如果项目包含使用STEP 7 Micro/WIN 3.2版建立的现有NETR/NETW配置，您必须在继续执行步骤1之前选择编辑其中一个现有配置或建立一个新配置。

可使用NETR/NETW指令向导，简化网络操作配置。向导向您查询初始化选项，并根据您的选择生成完整的配置。向导

### 步骤 2 指定端口号码和子例行程序名称

**注释：**项目可能包含一个NETR/NETW向导配置。所有以前建立的配置均被自动载入向导。向导会提示您完成以下两个

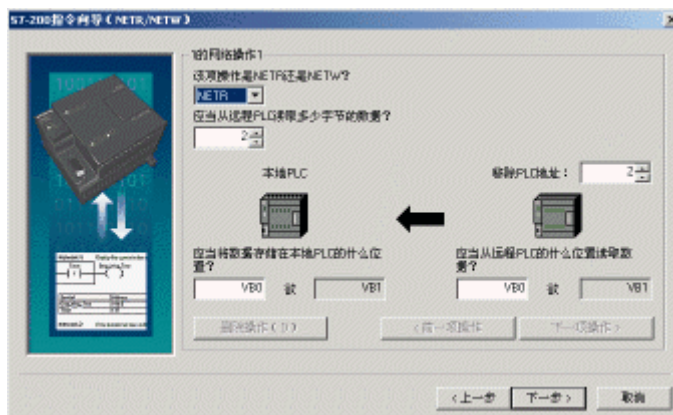
- 选择编辑现有配置，方法是单击“下一步>”。
- 选择从项目中删除现有配置，方法是选择“删除”复选框，并单击“完成”。

如果不存在以前的配置，向导会查询以下信息：

1. 必须被设为在PPI主模式中通讯。您指定操作将通过哪一个PLC端口通讯。
2. 向导建立一个用于执行具体网络操作的参数化子例行程序。向导还为子例行程序指定一个默认名称。您可以编辑默认名称。

### 步骤 3 指定网络操作

对于每项网络操作，您需要提供下列信息：



1. 指定操作是NETR还是NETW。
2. 指定从远程PLC (NETR) 读取的数据字节数或向远程PLC (NETW) 写入的数据字节数。
3. 指定您希望用于通讯的远程PLC网络地址。
4. 如果您在配置NETR, 指定以下内容:
  - 数据存储在本地PLC中的位置。有效操作数: VB、IB、QB、MB、LB。
  - 从远程PLC读取数据的位置。有效操作数: VB、IB、QB、MB、LB。
5. 如果您在配置NETW, 指定以下内容:
  - 数据存储在本地PLC中的位置。有效操作数: VB、IB、QB、MB、LB。
  - 向远程PLC写入数据的位置。有效操作数: VB、IB、QB、MB、LB。

#### 步骤 4 分配V内存

对于您配置的每一项网络操作, 要求有 12个字节的V内存。您指定可放置配置的V内存起始地址。向导会自动为您建议一个地址, 但您可以编辑该地址。请注意, 如果您使用重叠地址, 向导在显示一则警告后, 允许您进入下一步。

**注释:** 在NETR操作中, 核实将存储数据的变量没有与分配给向导的内存地址重叠。

#### 步骤 5 生成程序代码

回答这些查询后, 单击“完成”, S7-

200指令向导将为您指定的网络操作生成代码。由向导建立的子例行程序成为项目的一部分。欲在程序中启用网络通讯, 在主程序块中调用执行子例行程序 (NET\_EXE)。每次扫描循环时, 使用SM0.0调用该子例行程序。这样会启动配置网络操作执行。为每项网络操作建立的数据处理子例行程序会在适当时间被自动调用。

另请参阅:

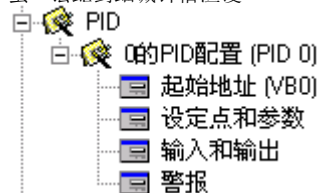
[NET\\_EXE指令](#)  
[标准NETR/NETW指令](#)

### 9.8.3 PID指令向导

欲使用PID向导:  
选择菜单命令 **工具(I) >**



指令向导...，然后选取PID；或单击浏览条中的指令向导图标，然后选取PID；或打开指令树中的“向导”文件夹并随后



#### 注释：

本向导仅限于第二代CPU。如果您在一个3.1x项目中有一个向导配置，项目依然工作。但是，向导不会识别该项目，您无法编

#### 步骤

- 1 指定环路号码
- 2 设置环路参数
- 3 环路输入和输出选项
- 4 环路警报选项
- 5 为计算指定内存区
- 6 指定初始化子例行程序和中断例行程序
- 7 生成代码

#### S7-

200指令向导的PID公式可用于简化PID操作配置。向导向您查询初始化选项，然后为指定配置生成程序代码和数据块

**注释：**在使用PID向导之前，程序必须被编译并位于符号编址模式。如果您尚未编译，向导会在PID配置进程开始提示您进行编译。

从第一个屏幕选择PID后，向导向您查询下列信息：

#### 步骤 1 指定环路号码

**注释：**如果项目包含使用STEP 7 Micro/WIN

3.2版建立的现有PID配置，您必须在继续执行步骤1之前选择编辑其中一个现有配置或建立一个新配置。

您指定配置哪一个PID环路。

#### 步骤 2 设置环路参数

**注释：**参数表地址的符号名已经由向导指定。PID向导生成的代码使用参数表地址建立操作数，作为参数表中的相对

环路定点是您为向导生成的子例行程序提供的一个参数。

1. 指定环路定点（SP）应当如何缩放。为“低范围”和“高范围”选择任何实数。默认值是0.0和100.0之间的一个实数。
2. 指定下列环路参数：

- 增益时间
- 样本时间
- 积分时间
- 派生时间

欲了解环路参数如何影响PID计算性能，请参阅PID计算。

#### 步骤 3 设置环路输入和输出选项

环路进程变量（PV）是您为向导生成的子例行程序指定的一个参数。

向导向您查询以下环路输入选项：

1. 指定环路进程变量（PV）应当如何缩放。您可以选择：
  - 单极（可编辑默认范围0至32000）
  - 双极（可编辑默认范围-32000至32000）
  - 20%偏移量（设置范围6400至3200，不可变更）

您还必须输入以下环路输出选项：

2. 指定环路输出应当如何缩放。您可以选择：
  - 输出类型（模拟或数字）

**注释：**如果您选择配置数字输出类型，则必须以秒为单位输入“任务循环时间”。

- 缩放（单极、双击或20%偏移量）

#### 步骤 4 设置环路警报选项

该向导为各种环路条件提供输出。当达到警报条件时，输出被设置。

指定您希望识别警报输入的哪些条件：

- 启用低警报（PV）并设置从0.0到高警报限制的标准化低警报限制
- 启用高警报（PV）并设置从低警报限制到1的标准化高警报限制
- 启用模拟输入模块错误，并指定输入模块附加在PLC上的位置

#### 步骤 5 为计算指定内存区

**PID指令**使用V内存中的一个36个字节的参数表，存储用于控制环路操作的参数。PID计算还要求一个“暂存区”，用于

您还可以选择增加PID的手动控制。位于手动模式时，PID计算不执行，环路输出不改变。

**注释：**

当PID位于手动模式时，输出应当通过向“手动输出”参数写入一个标准化数值（0.00至1.00）的方法控制输出，而不是用直接改变输出的方法控制输出。这样会在PID返回自动模式时提供无障碍式转换。

#### 步骤 6 指定子例行程序和中断例行程序

**注释：**

如果项目包含一个现用PID配置，已经建立的中断例行程序被列为唯读。因为项目中的所有配置共享一个公用中断例行

向导为初始化子例行程序和中断例行程序指定默认名称。您可以编辑默认名称。

#### 步骤 7 生成PID代码

该屏幕显示PID向导生成的POU列表，并对如何综合进程作出简要说明。

回答这些查询后，单击“完成”，S7-

200指令向导将为您指定的配置生成程序代码和数据块代码。由向导建立的子例行程序和中断例行程序成为项目的一部分。

欲在程序中启用该配置，每次扫描循环时，使用SM0.0从主程序块调用该子例行程序。该代码配置PID

0。该子例行程序初始化PID控制逻辑使用的变量，并启动PID中断“PID\_EXE”例行程序。根据PID样本时间循环调用PID中断例行程序。

另请参阅：

[PIDx\\_INIT \(PID向导子例行程序\)](#)

[PID算法](#)

[PID环路表](#)

[PID调谐控制面板](#)

**注释：**


具有PID向导配置的项目不直接使用标准PID指令。如果您使用PID向导配置，您的程序必须使用CALL

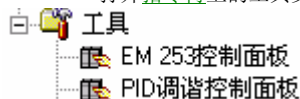
PIDxINIT来激活PID向导子例行程序。

[PID \(标准PID指令\)](#)

## 9.8.4 PID调谐控制板

使用以下一种方法打开PID调谐控制面板：

- 选取工具(I) > PID调谐控制面板 菜单命令。
- 单击浏览条上的PID调谐控制面板  按钮。
- 打开指令树上的工具文件夹，然后打开PID调谐控制面板。



### PID调谐兼容性

- PID环路自动调谐不能用于固件版本早于2.00发行版的控制器上的PID指令。
- 在早于STEP 7-Micro/WIN 4.0发行版的版本中使用PID向导创建或手工生成的PID指令，能够以手动输入调谐参数的校正方式进行调谐。这些PID指令无法使用自动调谐。
- 在早于STEP 7-Micro/WIN 4.0发行版的版本中使用PID向导创建的PID指令环路表不符合自动调谐格式。但是，如果您按照自动调谐格式修改环路表，那么PID指令可有自动调谐算法在其上执行。

[理解自动调谐](#)  
[扩展的环路表](#)  
[先决条件](#)  
[自动滞后和自动偏差](#)  
[自动调谐序列](#)  
[例外条件](#)  
[关于PV超出范围的注释（结果代码3）](#)  
[PID调谐控制面板](#)

### 理解自动调谐

S7-200 PLC现已包含PID自动调谐能力，并且STEP 7-Micro/WIN也已添加一PID调谐控制面板。这两项功能相结合，就大大加强了S7-200 Micro PLC系列产品的用途和便利。

自动调谐能够由用户程序从操作员板发起，或由PID调谐控制面板发起。PID环路能够一次一个地得到自动调谐；如有必要，也可八个环路同时进行自动调谐。PID自动调谐会计算增益、积分时间（重设）、微分时间（速率）等调谐数值的建议（接近最优）数值，并且允许您选择调谐环路的快速、中速、慢速或很慢应答。

借助于PID调谐控制面板，您可以发起自动调谐进程、中止自动调谐进程、用图形方式监视其结果。此控制面板不仅显示可能产生的警告，还能显示调谐过程的详细数据。

S7-200使用的自动调谐算法，是基于K. J. 朗格和T.

H. 朗格于1984年提出的一项称为继电器反馈的方法。在过去二十年里，继电器反馈已应用于范围广大的行业。

继电器反馈的核心，是在一原本稳定的进程中产生一个小幅度但却持续的振荡。基于此振荡的周期和在进程变量中观察到的幅度改变，此进程的终极频率和终极增益可得到确定。然后，使用这些终极增益值和值，PID自动调谐器会就增益、重设、速率诸调谐参数提供建议。

所建议的数值取决于您选择的该进程环路的应答速度。您可以选取快速、中速、慢速或很慢等应答。取决于您的进程，快速应答可能过调且将对应于某欠阻尼调谐状态。中速应答可能濒临过调且将对应于某临界阻尼调谐状态。慢速应答可能没有任何过调且将对应于某过阻尼调谐状态。很慢应答可能没有过调且将对应于某严重过阻尼调谐状态。

自动调谐器除建议调谐值外，还能够自动确定滞后和高峰PV偏差的数值。这些参数用于在限制由自动调谐器建立的持续振荡之幅度的同时，减轻进程噪声的效应。

此PID自动调谐器能够为直接作用和反作用P、PI、PD、PID诸环路确定建议调谐值。

PID自动调谐器的目的，在于确定一组调谐参数，从而为您的环路之最优数值提供合理的近似。由建议调谐值开始，您将能够进行进一步的优化。

### 先决条件

您想要自动调谐的环路必须处于自动模式。环路输出必须由PID指令的执行来控制。如果环路处于手动模式，自动调谐将会失败。

在发起自动调谐操作前，您的进程必须已达到某种稳定态；也就是说，PV已经达到设定点（或就P型环路而言，PV与设定点的某偏差已稳定）。

在开始自动调谐时，环路输出值最好能够在控制范围的中点附近。自动调谐过程会通过向环路输出中做出小步长改变，在进程中

建立一个振荡。倘若环路输出很接近其控制范围的任何一端，由自动调谐过程引入的步进改变可能令输出值试图超出最小或最大范围极限。

如果发生此情况，就可能导致自动调谐错误状态的产生，并且肯定会得出较不理想的建议值。

### 自动滞后和自动偏差

滞后参数规定了PV (process variable、进程变量) 相对于设定点的允许漂移 (加或减)；此漂移不会令继电器控制器改变输出。此数值用于最小化PV信号中噪声的效应，从而更精确地确定此进程的固有振荡频率。

如果您选择自动确定滞后值，PID自动调谐器将会进入滞后确定序列。此序列包括对进程变量进行一段时间的抽样，然后计算抽样

为了使抽样具有统计意义，样本集必须至少有100个样本。假设环路的抽样时间为200毫秒，那么取得100个样本需要20秒。抽样

在取得全部样本后，标准偏差将得到计算。滞后值被定义为标准偏差的两倍。此计算的滞后值将会写入环路表的实际滞后域 (AHYS)。



#### 提示：

在自动滞后序列工作过程中，正常的PID计算不会进行。因此，在发起自动调谐序列前，此进程处于稳定态是至关重要的。这样

偏差参数规定了欲用的PV围绕设定点之峰到峰摆动。如果您选择自动确定此数值，欲用的PV偏差将为滞后值乘以4.5。输出将被

### 自动调谐序列

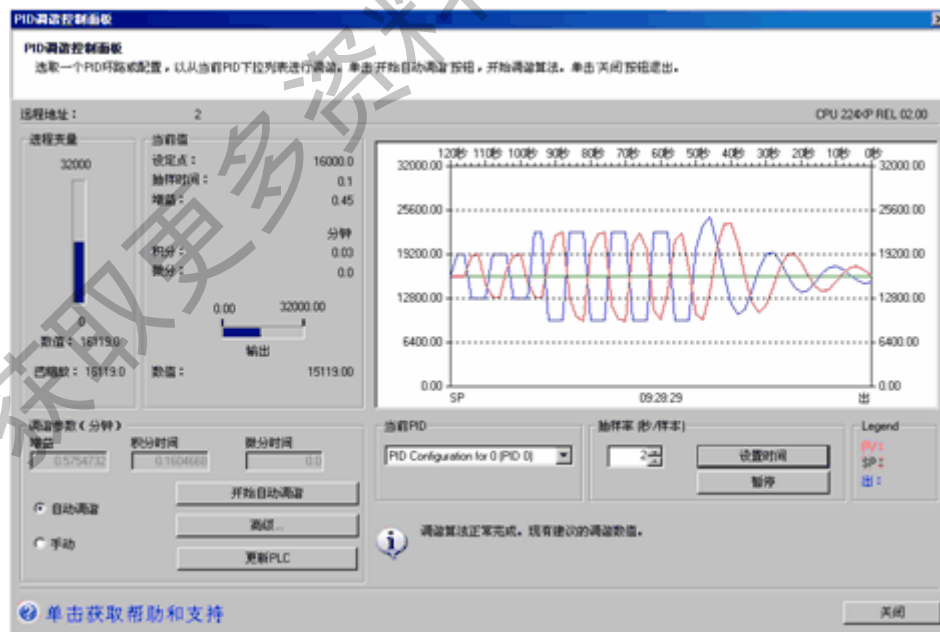
在确定了滞后值和偏差值后，自动调谐序列就会开始。此调谐过程开始于在环路输出上施加初始输出步长。

此输出值的改变应会导致进程变量值的相应改变。当输出变化将PV推离设定点足够远以至于超出滞后边界时，自动调谐器就会检测到

调谐器会继续对PV进行抽样和等待下一个过零事件。此序列总共需要十二个过零才能完成。所观察到的峰到峰PV值幅度 (峰值误差) 和过零发生的速率与此进程的动态特性直接相关。

在自动调谐过程初期，输出步长值会按比例调整一次，从而令随后的PV峰到峰摆动更接近欲用的偏差值。一旦做出此调整后，新的输出步长值将写入环路表的“实际步长”域 (ASTEP)。

如果两次过零之间的时间超出过零监视间隔时间，自动调谐序列将以错误告终。过零监视间隔时间的默认值为两小时。



上面的屏幕图象，显示了在某直接作用环路的自动调谐序列中，输出和进程变量的行为。PID调谐控制面板被用于发起和监视此

请注意为了让此进程经历小幅度振荡（如PV值所证实），自动调谐器是如何切换输出的。此PV振荡的频率和幅度，预示了进程增益和固有频率。

基于在自动调谐过程中收集的进程频率和增益信息，终极增益和终极频率将被计算得出，由此将计算增益（环路增益）、重设（积分时间）、速率（微分时间）的建议值。



**提示：**目前的环路参数确定自动调谐器将计算的调谐值。对于只有“增益”和“积分时间”的PI环路，远 麓称鹊 扑闼碌脑鲟婧突 质奔渲担 丁桐奈 7.质奔浣 0.0（无微分动作）。如果在“进程变量”值中有噪音，“微分时间”参数则无法正常作业。

#### 例外条件

调谐的执行过程可能产生三种警告状况。这些警告用环路表ASTAT域的三个位报告；这些位一经设置，在自动调谐序列被再次发

- 如果偏差值小于滞后值的4倍，警告0就会发出。这项检查在实际已知滞后值时进行，该数值取决于自动滞后设置。
- 如果在自动调谐过程的头2.5个循环中收集到的两个高峰误差值之差超过滞后值的8倍，警告1就会发出。
- 如果测量的平均峰值误差小于滞后值的4倍，警告2就会发出。

除了这些警告状况外，还可能有一些错误状况。下表列出了这些错误状况，以及对每项错误之原因的描述。

#### 调谐执行中的错误状况

结果代码（在ARES内）	条件
01	由用户中止 EN位在调谐进行中被清除
02	由于过零监视超时而中止 半循环持续时间超过过零监视间隔
03	由于进程超出范围而中止 PV超出范围：-在自动滞后序列过程中，或者 -在第四次过零前两次，或者 -在第四次过零后
04	由于滞后值超出最大值而中止 用户指定的滞后值或自动确定的滞后值 > 最大值
05	由于非法配置值而中止 下列范围检查错误：-初始环路输出值 < 0.0 或 > 1.0-用户指定的偏差值 ≤ 滞后值或 > 最大值 -初始输出步长 ≤ 0.0 或 > 最大偏差 -过零监视间隔时间 < 最小值-环路表中的抽样时间值为负数
06	由于数字错误而中止 遇到非法的浮点数或除数为零
07	执行PID指令时无功率流（手动模式） 当自动调谐在进行中或被请求时PID指令以无功率流的方式执行
08	自动调谐只可用于P、PI、PD或PID等环路 环路类型不是P、PI、PD或PID

#### 关于PV超出范围的注释（结果代码3）

进程变量只要大于0.0且小于1.0，就会被自动调谐器视为在范围内。

如果在自动滞后序列中发现PV超出范围，调谐将立即中止，并且会发出进程超出范围错误。

如果在调谐序列的起始点和第四个过零之间发现PV超出范围，输出步长值将被减半，调谐序列将从头开始。如果在重新启动后于第一个过零后检测到第二个PV超出范围事件，调谐就会中止，并且发出进程超出范围错误结果。

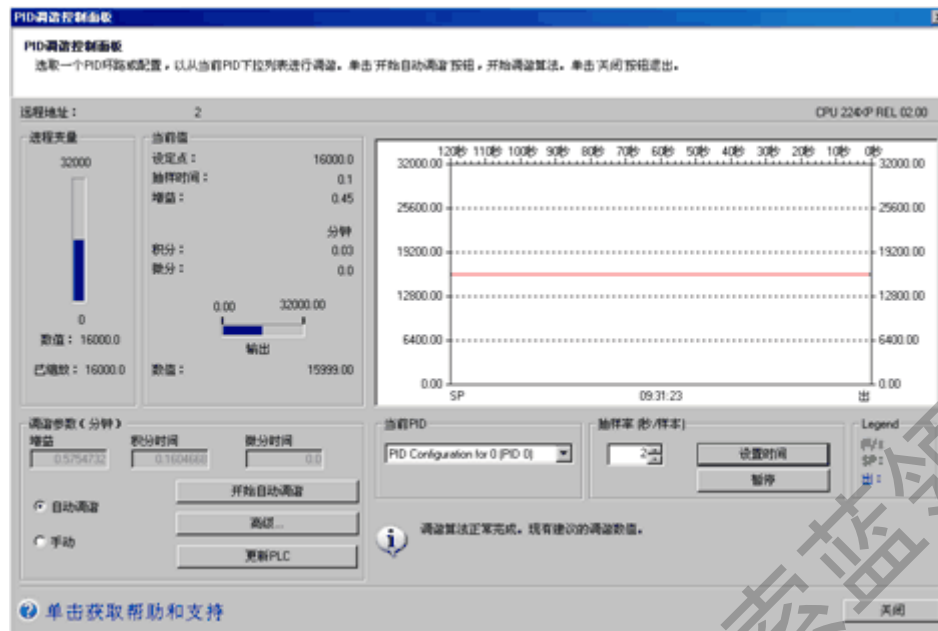
在第四个过零之后的任何PV超出范围事件会导致调谐的立即中止，并且会给出进程超出范围错误结果。

#### PID调谐控制面板

##### STEP 7-

Micro/WIN包括一个PID调谐控制面板，令您可用图形方式监视PID环路的行为。此外，这个控制面板也令您可发起自动调谐序列

兄垢眯蛄校 约疤子媒丁桐牡麓持禱蝻 约旱牡麓持怠



欲使用此控制面板，您必须在与一个S7-200

PLC通讯，并且此PLC内必须存在向导生成的PID环路配置。此PLC必须处于“运行”模式，以允许控制面板显示某PID环路的操作。夏娴钠聊煌椒蠓从沉丝 刁泼妍必哪 掀聊弧

控制面板在此屏幕左侧顶部显示目标PLC的站地址（远程地址）。此屏幕右侧顶部显示了PLC类型和版本号。在远程地址域下面，有进程变量的条形图表示及已缩放数值和未缩放数值。PV条形图右侧紧接着当前值区域。

当前值区域显示了设定点、抽样时间、增益、积分时间、微分时间的数值。输出值用一水平条形图及其数值显示。当前值区域右边是一图形显示。

此图形显示用不同颜色绘出PV、SP及输出作为时间函数的曲线。PV和SP共用此图左侧的垂直尺度，输出的垂直尺度在此图右侧。

此屏幕底部左侧是“调谐参数（分钟）”区域。增益、积分时间、微分时间诸值会显示在此区域内。圆点按钮会表明所显示的是增益、积分时间、微分时间的“当前”、“建议”或“手动”数值。您可以单击此圆点按钮，显示三个来源之一的这些数值。如要修改调谐参数，请单击手动圆点按钮。

您可以使用“更新PLC”按钮，就正在监视的PID环路，将所显示的增益、积分时间、微分时间诸值传输给PLC。您可以使用“开甲远 廢场卑磁 づ远 廢承站小 R 坏 干远 廢承站畔舳 翱 甲远 廢场卑磁 せ 岂 涑 伞 巴 V 棺远 廢场卑磁 了

图形显示的正下方是“当前PID”选择区域，具有一个拉下菜单，供您选取想要用此控制面板监视的PID环路。

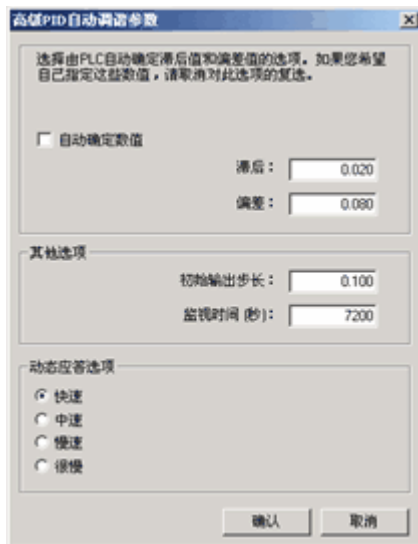
在“抽样速率”区域，您可以选取图形显示的抽样速率，其范围为每个样本1至480秒。您可以编辑抽样速率，然后使用“设置时间”来套用此更改。此图的时间尺度会得到自动调整，从而最佳显示新速率的数据。

您可以通过按“暂停”按钮冻结此图。按“继续”即可以选定的速率继续进行数据抽样。如要清除此图，请在图中用鼠标右键单击，然后选取“清除”。

图选项区域的右边是图例，指明PV、SP、输出等绘图所用的颜色。

当前PID选择区域正下方用于显示正在执行操作的相关信息。





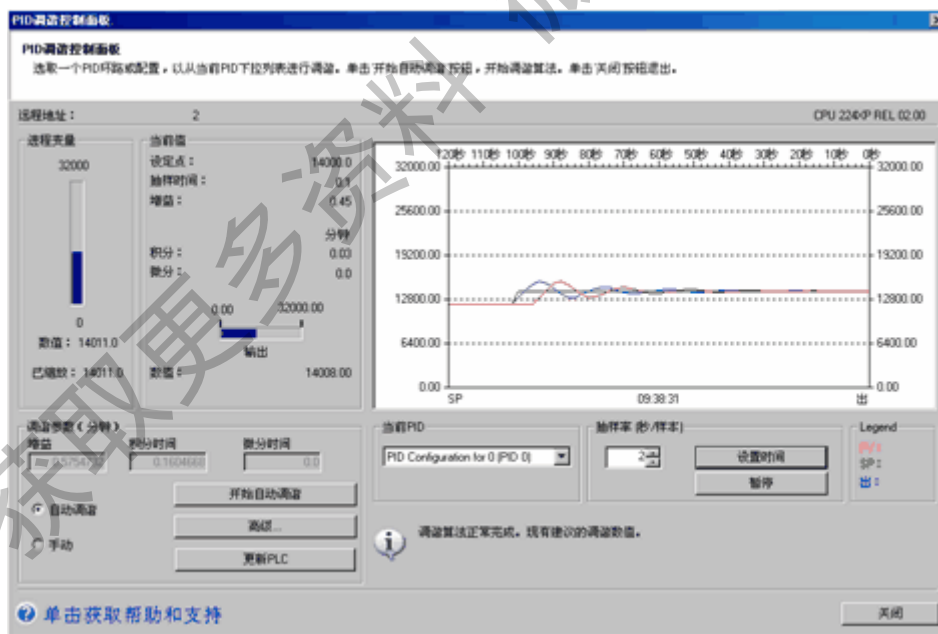
“调谐参数”区域内的“高级...”按钮令您可进一步配置自动调谐过程的参数。上面所示为高级屏幕图象。

由高级屏幕，您可以复选让自动调谐器自动确定滞后值和偏差值（默认设置）的方框；或在这些域输入特定数值，从而最小化自动调谐过程对进程的扰动。

在“其他选项”区域，您可以指定初始输出步长和过零监视超时期限。

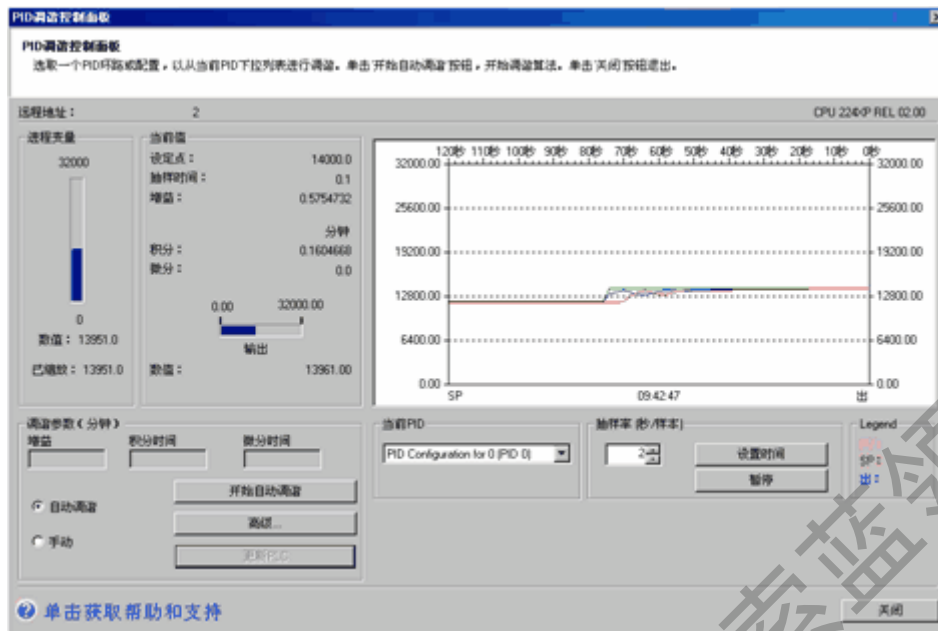
在“动态应答选项”区域，请针对您的进程，单击对应于您希望的环路应答类型的圆点按钮。“动态应答”指环路对进程变化的应答速度，而不是指进程的速度。取决于您的进程，快速应答可能过调且将对应于某欠阻尼调谐状态。中速应答可能濒临过调且将对应于某临界阻尼调谐状态。慢速应答可能没有任何过调且将对应于某过阻尼调谐状态。很慢应答可能没有过调且将对应于某严重过阻尼调谐状态。

一旦做出欲用的选择，请单击‘确定’返回至PID调谐控制面板的主屏幕。



在完成自动调谐序列且已将建议的调谐参数传输至PLC后，您可以使用控制面板监视环路对设定点步进变更的应答。上面的屏幕截图显示了对设定点（范围22000至14000）的应答。

请注意此进程使用原始调谐参数的过调和阻尼的长程振荡行为。



上面的屏幕图象，显示了在套用选取快速应答后由自动调谐过程确定的数值时，环路对设定点变更（由12000至14000）的应答。请注意就此进程而言已经没有过调，但仍有些微振荡。如欲以应答速度为代价消除此振荡，您需要选取中速或慢速应答，然后重新运行自动调谐过程。

一旦针对有关环路的良好起始点后，您可使用控制面板微调有关参数，并监视环路对设定点变更的应答。以此方式，您可以精调有关进程，以在您的应用中获得最优应答。

#### 另请参阅：

[PID算法](#)  
[PID环路表](#)

#### PID向导

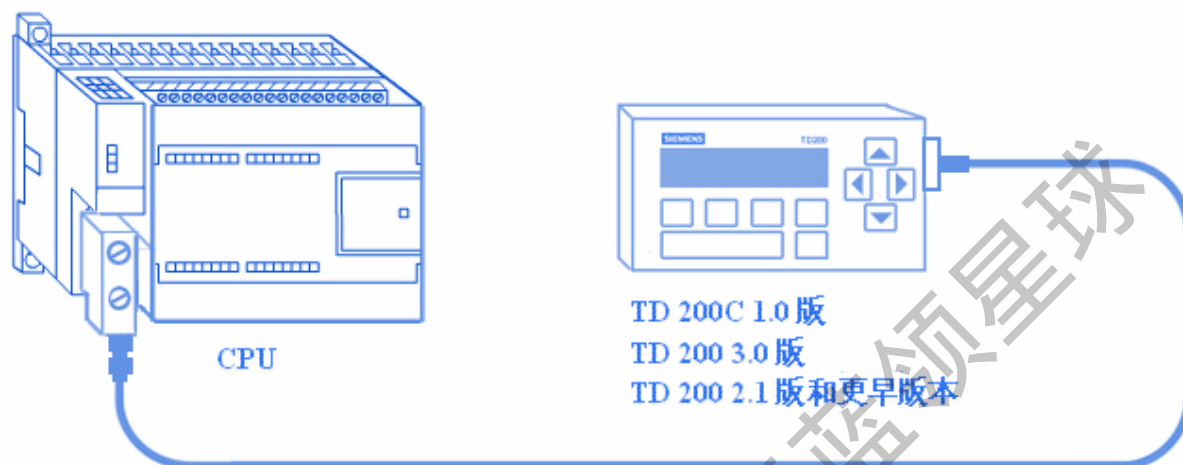
[PIDx\\_INIT \(PID向导子例行程序\)](#)

#### 注释：

具有PID向导配置的项目不直接使用标准PID指令。如果您使用PID向导配置，您的程序必须使用CALL PIDxINIT来激活PID向导子例行程序。

[PID \(标准PID指令\)](#)

## 9.8.5 TD 200 文字显示功能



请使用TD 200向导配置下列TD 200版本

- TD 200 v2.1或之前版本具有9个固定功能按钮
- TD 200 v3.0支持多个语言集、自定义用户菜单、用字符串显示的PLC数据，并且带有9个固定功能按钮（与TD 200第1.0发行版的项目向后兼容）
- TD 200C v1.0不仅具有TD 200 v3.0的全部功能，还有可配置的小键盘面板布局（使用TD小键盘设计员工具）、更多的诊断功能（载入内存盒、‘停止至运行’命令、编辑PLC内存），以及多达20个可配置功能按钮（与TD 200第1.0发行版的项目向后兼容）

TD 200向导会引导您逐步完成TD 200的配置，协助您设置TD 200功能的参数，以及输入用户讯息。如果您需要特定TD 200信息以完成配置，请参阅《SIMATIC文字显示（TD）用户手册》（SIMATIC Text Display (TD) User Manual）。在向导配置完成时，您在向导中指定的信息和地址将用于生成数据页、符号表及指令子例行程序。通过调用TD 200向导和为新的配置指定新的起始地址，您可以创建多个TD 200配置。

另请参阅：

[TD 200 v2.1或之前版本的TD 200向导步骤](#)  
[TD 200 v3.0或TD 200C v1.0的TD 200向导步骤](#)

## 9.8.6 TD 200向导 (TD 200 v2.1版或之前版本)



通过选取菜单命令 **工具(T) > TD 200向导? T** 或单击浏览条上的TD 200向导图表，启动TD 200向导

单击“下一步”按钮前往TD 200向导的每个新页。

在完成配置时，单击“完成”按钮。

在向导配置完成时，一个数据页会得到创建，其中包含您在TD 200向导中指定的信息和地址。此信息存储于V内存中，并且每个位置均有注解。

通过调用TD 200向导和为新的配置指定新的起始地址，您可以创建多个TD 200配置。

注释：

- 使用STEP 7-Micro/WIN V3.2或更高版本下载的TD 200配置可在上载到新项目中后加以编辑。
- 如果想要在完成后检视已配置的参数块和讯息，您可以打开数据块编辑器，然后选择查看适当的数据页。

步骤

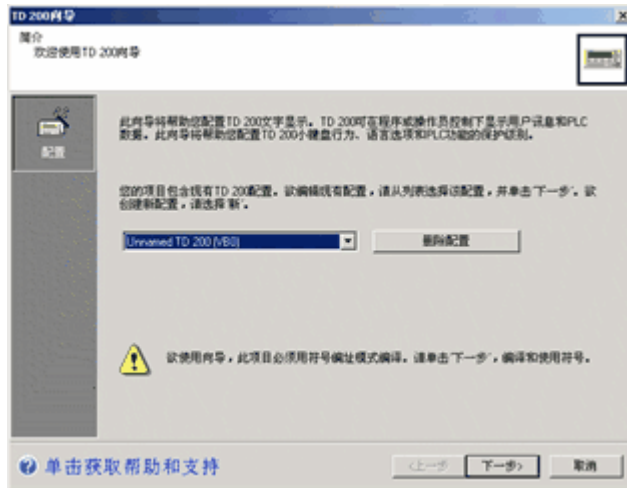
- 1 [编辑现有配置](#)
- 2 [选择TD型号和版本](#)
- 3 [本地化显示](#)
- 4 [标准菜单（每日时间、强制、密码）](#)
- 5 [功能键和更新速率](#)
- 6 [警报选项](#)
- 7 [分配内存](#)
- 8 [警报（输入讯息）](#)

## 9 生成项目元件

### 第1步 编辑现有配置 (TD 200 v2.1或之前版本)

[返回顶端](#)

[TD 200 v3.0或TD 200C v1.0的向导步骤](#)

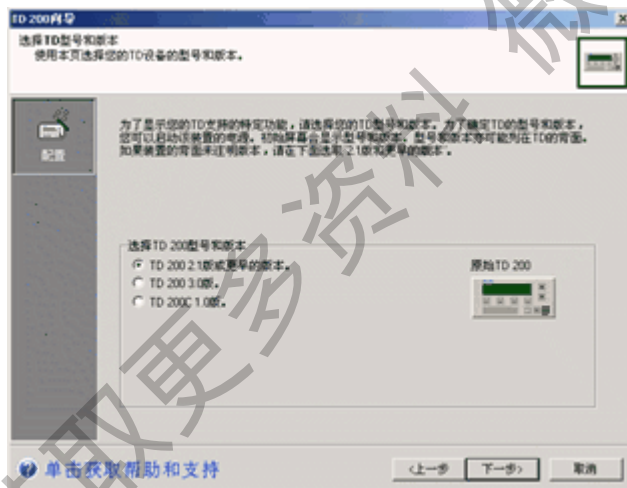


选取您想要编辑的配置。如果您的项目不存在现有配置，那么单击“下一步”前往第2步。

### 第2步 选择TD 200型号和版本 (TD 200 v2.1或之前版本)

[返回顶端](#)

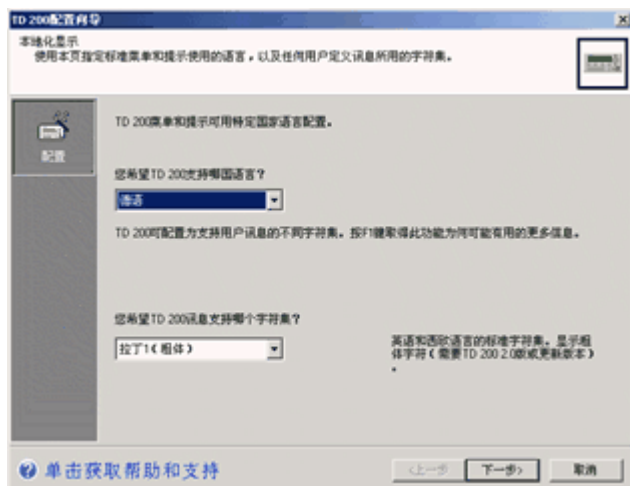
[TD 200 v3.0或TD 200C v1.0的向导步骤](#)



不同的TD 200型号和版本支持不同的功能。为了正确配置TD 200，您必须选取所用TD 200的型号和版本。TD 200 v3.0和TD 200C v1.0使用与TD 200 v2.1不同的向导步骤。

### 第3步 本地化显示

[返回顶端](#)



**国家语言：**此设定会影响TD 200的预配置菜单和提示。请选择您希望TD 200用于预配置菜单和提示的语言。

**字符集：**此设定决定了您在创建用户讯息时可以使用的字符类型。

- 每行讯息最多可显示20个单字节字符。

**注释：**当您在TD 200向导中键入或由STEP 7-Micro/WIN的数据块编辑器检查讯息时，Latin 1 Bold (拉丁1粗体) 字符不会显示为粗体；但这些字符会在TD 200显示单元上正确显示（即以粗体出现）。

- **注释：**简体中文字符集中有一些拉丁1字符。取决于您的键盘和(或)仿真器，拉丁1字符既可能为单字节格式，也可能为双字节格式。例如：荒美拥丿纸讠址 蚩崙瘰

讯息编辑器会显示10个双字节字符位置。每个字符位置可容纳下列组合之一：

- \* 一个双字节字符
- \* 两个单字节字符
- \* 一个单字节字符和一个单字节空格

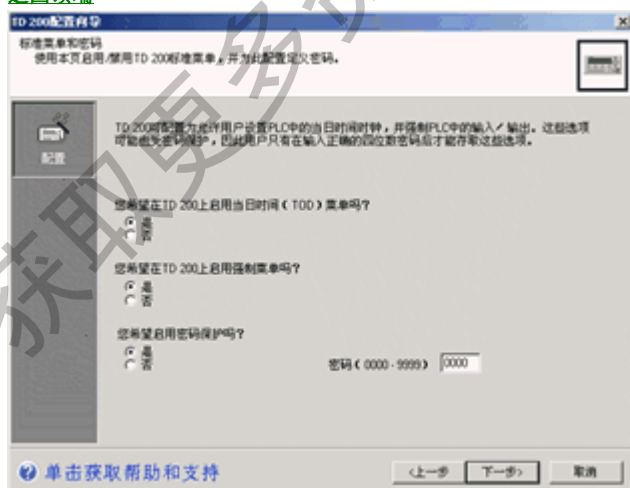
一个字符位置中不能容纳一个单字节字符加一个双字节字符。



**提示：**如果您在使用2.0版之前的TD 200版本，请勿使用默认字符集（拉丁1）。请只使用原版TD 200字符集或条形图字符集。倘若选择了所用TD 200版本不支持的字符集，在下次使用新配置启动TD 200时，您将收到一个关于参数块的非具体错误讯息。如要纠正此错误，请更改字符集选择，然后将修正后的配置下载到TD 200上。如果您不知道所用TD 200的版本，请将它关闭后重新启动。版本号会在接通电源时显示。

#### 第4步 标准菜单（每日时间、强制、密码）

[返回顶端](#)



“每日时间 (TOD)”和“强制”菜单选择令您可选择是否启用“TOD时钟”菜单和“强制”菜单。一旦启用某个选择，用户可在TD 200中使用相应菜单。若某个选择未启用，该菜单不会在TD 200的菜单模式中出现。

“密码保护”令您可启用一个四位数字密码（从0000到9999）。此密码控制着用户编辑嵌入讯息中的变量和使用菜单模式的能力。如果启用密码保护，向导中会出现一个域，供设置密码之用。此密码不是CPU密码。此密码存储于TD 200内，只影响对该TD 200内编辑功能的使用。

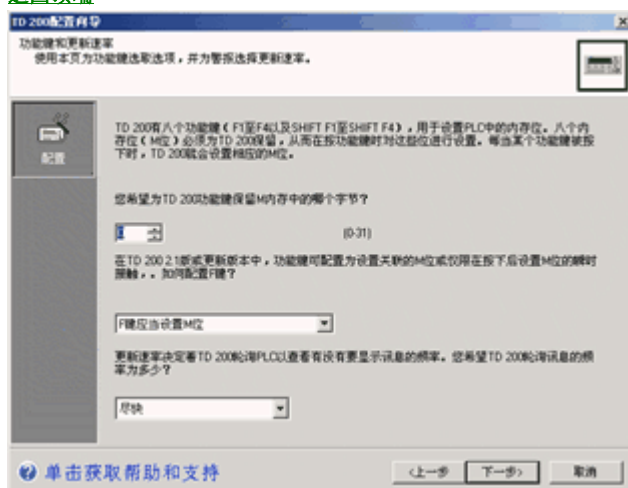


### 提示:

嵌入数据在向导中后面出现的一个屏幕上加以配置，但此处的密码保护选项会影响到您在后面能否选择“数据的用户编辑”。

## 第5步 功能键和更新速率

[返回顶端](#)



TD 200具有8个用于设置内存位或作为瞬时接触的功能键（F1至F4和SHIFT F1至SHIFT F4）。为了读取TD 200功能键，内存的八个位（M位）必须为TD 200保留。每当某个功能键被按下时，TD 200就会设置相应的M位。

- 请为TD 200输入一个M内存字节地址。
- 选取TD 200的F键行为。这些F键能够将M位设置为如同锁存接触或瞬时接触那样操作。



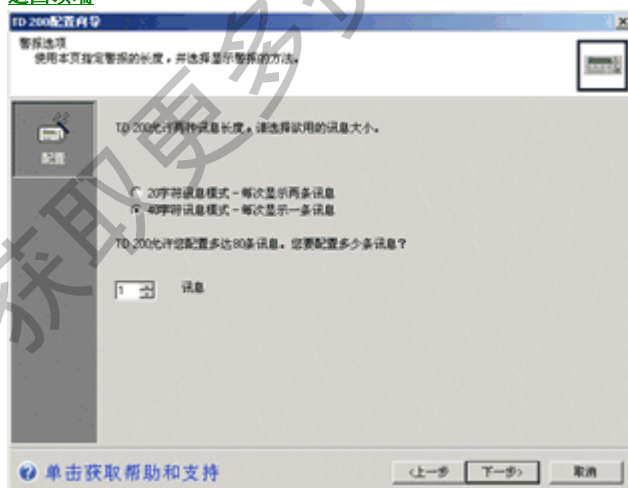
### 警告 每当某个功能键被按下时，TD

200就会设置相应的M位。如果您不欲使用功能键，并且因此没有指定M字节地址，TD 200为这些功能键指定的默认值为字节M0。如果您的程序使用M0中的位，并且某用户按下任何功能键，TD 200就会设置M0中的相应位，从而重写已由程序赋予该位的数值。对M位的无意变更可能令您的程序有意外举动。不可预知的控制器操作可能造成人员死亡或严重伤害，以及设备损坏。即使您的程序不使用功能键，也请总是保留一个M区地址。更新速率决定着TD 200轮询PLC以查看有没有要显示信息的频率。

- 请由列表框中选取一个更新速率。

## 第6步 警报选项

[返回顶端](#)



TD 200允许两种消息大小。请选取欲用的大小：

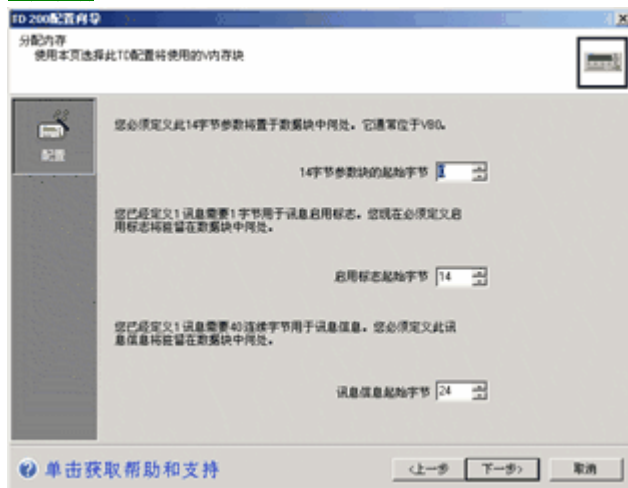
- 20字符消息模式 — 每次显示两条消息
- 40字符消息模式 — 每次显示一条消息

TD 200允许您配置多达80条消息。

- 请输入您想要创建的讯息数目。

### 第7步 分配内存

[返回顶端](#)



存储参数定义的位置必须得到指定，正常位置为VB0。

- 请输入参数块的起始字节地址（最多可包含14个字节）。
- 请输入讯息启用标志的起始字节地址。
- 请输入讯息内容的起始字节地址。

### 第8步 警报（输入讯息）

[返回顶端](#)



在20字符域中输入您的讯息。

**注释：** 原版TD

200字符集和条形图字符集必须使用特别ALT键组合键入讯息字符。所有其他字符集可用普通键盘和（或）仿真器键入。

您能够复制当前可见的讯息并粘贴到下一条讯息。

在“讯息浏览”页中，您可以用下列两种方式之一转到另一条讯息：

- 单击“< 上一个讯息”和“下一个讯息 >”按钮，或者
- 从“转到讯息”框中选取适当讯息，然后单击“转到讯息”按钮，即可直接提取任何讯息。

#### 原版TD 200字符集的ALT键组合

TD 200支持对特定国际字符和特殊字符的显示，这些字符是在TD 200向导中使用ALT键组合加以输入。如要使用ALT键组合：

1. 确定键盘上的NUM LOCK已启用。
2. 按住ALT键。
3. 使用键盘上的数字小键盘，键入该字符的4位数码。
4. 放开ALT键。

请注意，您在TD 200向导中输入的某些字符，在数据块编辑器中有不同的表示。但是，这些字符在TD 200中的显示将与您在TD 200向导中的输入一样。

°	ALT + 0248	m	ALT + 0230	e	ALT +
0238					
å	ALT + 0134	ñ	ALT + 0164	b	ALT +
0225					
ä	ALT + 0132	ö	ALT + 0148	S	ALT +
0159					
æ	ALT + 0145	a	ALT + 0224	ü	ALT +
0129					
À	ALT + 0143	s	ALT + 0229		ALT +
0157					
Æ	ALT + 0146	W	ALT + 0234	Þ	ALT +
0195					
ç	ALT + 0155	P	ALT + 0227	Û	ALT +
0180					

### 条形图字符集的ALT键组合

如要在TD

200向导中输入条形图字符，您必须使用下列ALT键组合。请注意，这些字符在此向导（及数据块编辑器）中的表示，与在TD 200本身中不同，但会在TD 200中正确显示。

□	ALT + 0200
□□	ALT + 0201
□□□	ALT + 0202
□□□□	ALT + 0203
□□□□□	ALT + 0204
↑	ALT + 0194



### 嵌入数据

- 将光标放在内的正确位置，然后单击“嵌入数据”按钮。
- 选择数据格式和显示格式。
- 输入小数点右边要显示的位数。

当心：嵌入数据值右端的字符具有该数值在TD 200显示中的锚固点的作用。在TD

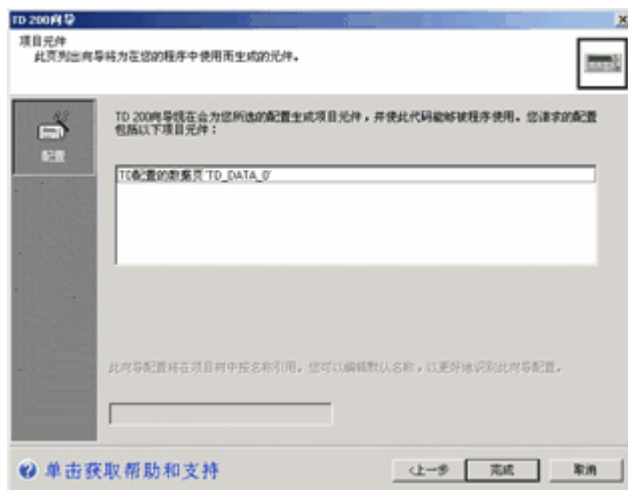
200显示上的讯息中，数据值始终以该锚固点为基准向右对齐。随着数据值的加大，它将在锚固点的左边使用更多的空间

- 选取用户是否必须确认收到该讯息。
- 选择是否允许用户编辑此数据。（如果您在第3步中未启用“密码保护”，则可使用“上一步”按钮返回至该屏幕，然后更改您的选择。使用“上一步”按钮不会丢失在之前的屏幕上已做出的选择。）
- 如果允许编辑，请选择“数据的用户编辑”是否应受密码保护。此外，请选择当讯息被启用后，该数据是否应变成可

### 第9步 生成项目元件

[返回顶端](#)





TD 200向导会为您选定的配置（程序块和数据块）生成项目元件，并令此代码可供您的程序使用。您必须下载TD 200配置块（数据块）、系统块及程序块至S7-200 CPU，才能更新此配置。

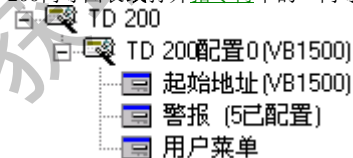
另请参阅：

[TD 200 SIMATIC文字显示功能](#)

### 9.8.7 TD 200向导 (TD 200 v3.0版或TD 200C v1.0版)



通过选取菜单命令 **工具(T) > TD 200向导? (W)** 或单击浏览条上的TD 200向导图标或打开 **指令树** 中的“向导”文件夹并随后打开此向导或某现有配置，启动TD 200向导。



单击“下一步”按钮前往TD 200向导的每个新页。

在完成配置时，单击“完成”按钮。

在向导配置完成时，一个数据页会在您的项目中得到创建，其中包含您在TD 200向导中指定的信息和地址。供TD 200使用的参数和讯息信息存储于V内存中。

在向导配置完成时，一个数据页会在您的项目中得到创建，其中包含您在TD 200向导中指定的信息和地址。供TD 200使用的参数和讯息信息存储于V内存中。

通过调用TD 200向导和为新的配置指定新的起始地址，您可以创建多个TD 200配置。

注释：

- 使用STEP 7-Micro/WIN V3.2或更高版本下载的TD 200和TD 200C配置可在上载到新项目中后加以编辑。
- 如果想要在完成后检视已配置的参数块和讯息，您可以通过打开数据块编辑器，然后选取适当的数据页加以检查。
- 出于记录或调试目的，您也可打印向导配置。此打印输出将包括您在此向导中所做选择的记录，以及所创建用户讯息母本。请在STEP 7-Micro/WIN打印向导配置，请在[打印对话框](#)中复选向导配置复选框。

步骤	基本配置
1	<a href="#">编辑现有配置</a>
2	<a href="#">选择TD 200型号和版本</a>
3	<a href="#">启用标准菜单、密码及设置更新速率</a>
4	<a href="#">本地化显示</a>
5	<a href="#">配置小键盘按钮</a>
6	<a href="#">创建用户菜单和文字</a>
7	<a href="#">定义用户菜单和文字</a>
8	<a href="#">创建进程警报与文字</a>
9	<a href="#">定义警报选项与文字</a>
	指定语言集
8	<a href="#">选择要编辑、增加或删除的语言集</a>
9	<a href="#">本地化显示 (指定TD 200C语言集)</a>
	完成最后步骤
10	<a href="#">向导部分完成</a>
11	<a href="#">为配置分配内存</a>
12	<a href="#">生成项目元件</a>

### 第1步 编辑现有配置 (TD 200 v3.0或TD 200C v1.0)

[返回顶端](#)

[TD 200 v2.1或之前版本的向导步骤](#)

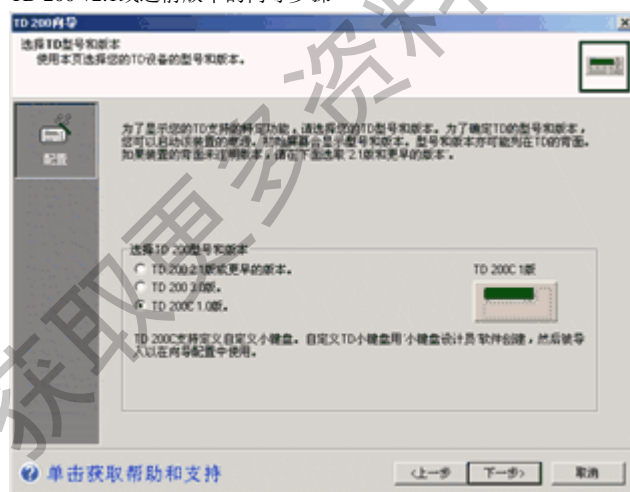
bml TD\_wiz\_CV1V3\_exist\_config.gif]

选取您想要编辑的配置。如果您的项目不存在现有配置，那么单击“下一步”前往第2步。

### 第2步 选择TD 200型号和版本 (TD 200 v3.0或TD 200C v1.0)

[返回顶端](#)

[TD 200 v2.1或之前版本的向导步骤](#)



不同的TD 200型号和版本支持不同的功能。为了正确配置TD 200，您必须选取所用TD 200的型号和版本。TD 200 v2.1使用与TD 200 v3.0和TD 200C 1.0不同的向导步骤。您可以通过单击适当按钮直接前往现有配置的各个部分。



配置

[基本配置](#) [第1-5步](#)

此配置部分令您可设定TD设备的参数，包括指定小键盘行为、语言选项，以及对PLC功能的保护程度。这属于必需部分。



[菜单](#) [选用的第6步](#) 这个部分令您可为此TD

200配置定义一个用户菜单。此用户菜单可用于提供指示性文字讯息，或协助机器的故障排除。配置用户菜单并非必需。



[警报](#) [选用的第7步](#) 这个部分令您可为此TD

200配置定义位触发警报。警报是具有优先顺序的用户讯息，会在程序控制下显示。对警报的配置并非必需。

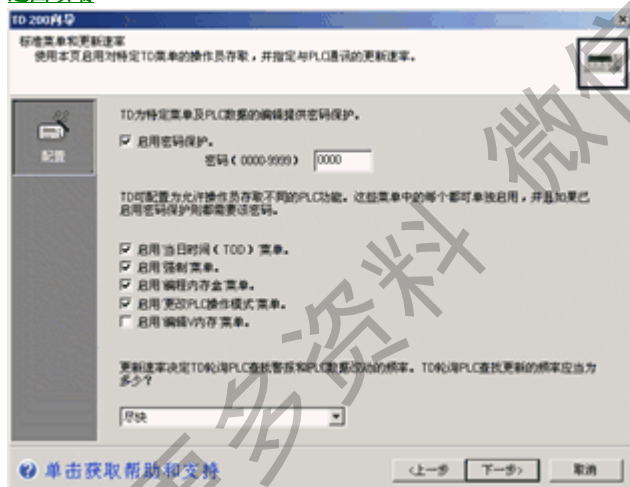


[语言](#) [选用的第7-8步](#)

如果您定义了警报或用户菜单，“语言集”图标就会出现；这就让您可以定义用户讯息的不同语言翻译。

### 第3步 启用标准菜单、密码及设置更新速率

[返回顶端](#)



此页令用户可以指定哪些TD

200菜单选项应为此应用程序出现。这些菜单可为方便用户或出于安全目的而启用或禁用。一旦在向导中启用了标准菜单选择，

贝撒 留察蛟涌惺保 没BI 档裁款D

200中使用该菜单。倘若某个菜单选择已在向导中被禁用，那么当此应用程序运行时，该菜单将无法在TD 200中使用。

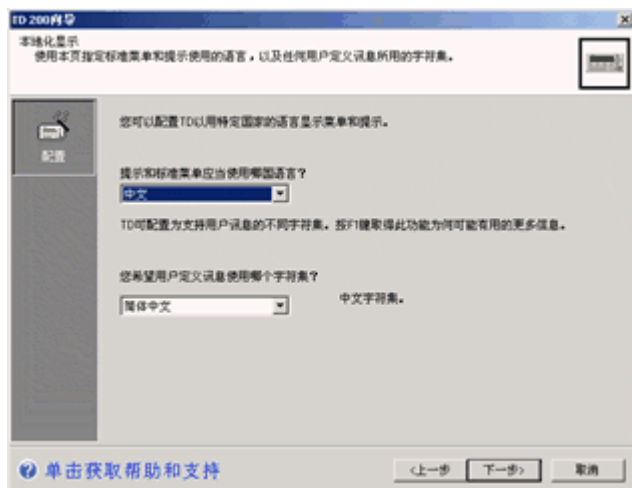
“密码保护”令您可启用一个四位数字密码（从0000到9999）。此密码控制着用户编辑嵌入讯息中的变量和使用菜单模式的能力。

如果启用密码保护，向导中会出现一个域，供设置密码之用。此密码不是CPU密码。此密码存储于TD 200内，只影响对该TD

200内编辑功能的使用。

### 第4步 本地化显示

[返回顶端](#)



请选取TD应为标准提示和菜单使用的国家语言。

请选取您创建的用户讯息应使用的字符集。请注意，选取某字符集并非一定意指特定国家语言。但是，许多国家语言需要特定字符集，方能显示该语言中使用的所有字符。

若是定义了多个语言集（第8步）的配置，则本地化显示页会针对每个语言集出现。每语言集可指定一个不同的国家语言和（或）地址。

**国家语言：**此设定会影响TD 200的预配置菜单和提示。请选取您希望TD 200显示预配置菜单和提示的语言。

**字符集：**此设定决定了您在创建用户讯息时可以使用的字符类型。

- 每行讯息最多可显示20个单字节字符。

**注释：**当您在TD 200向导中键入或由STEP 7-Micro/WIN的数据块编辑器检查讯息时，Latin 1 Bold（拉丁1粗体）字符不会显示为粗体；但这些字符会在TD 200显示单元上正确显示（即以粗体出现）。

- 简体中文字符为双字节字符。每行讯息最多可显示10个双字节字符。

**注释：**简体中文字符集中有一些拉丁1字符。取决于您的键盘和（或）仿真器，拉丁1字符既可能为单字节格式，也可能为如同简体中文字符的双字节字符。如果您在同一条讯息中将单字节字符与双字节字符混用，请切记单字节字符的后面只能接单字节字符或空格。

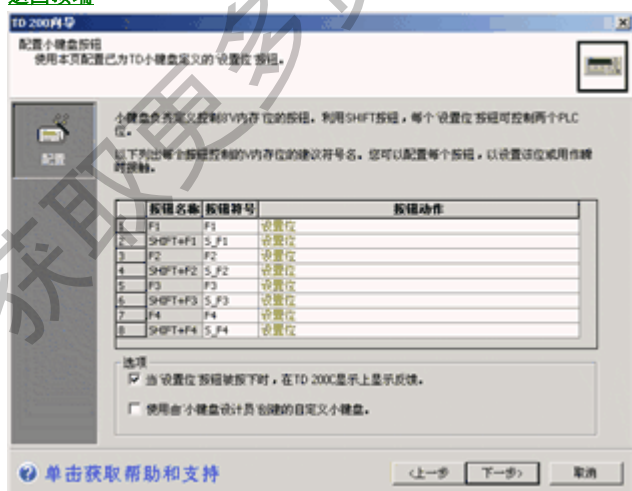
讯息编辑器会显示10个双字节字符位置。每个字符位置可容纳下列组合之一：

- \* 一个双字节字符
- \* 两个单字节字符
- \* 一个单字节字符和一个单字节空格

一个字符位置中不能容纳一个单字节字符加一个双字节字符。

## 第5步 配置小键盘按钮

[返回顶端](#)



TD 200C和TD 200

V3的标准面板提供了四个“设置位”按钮和一个Shift按钮，可用于控制PLC内的总共8个V内存位。如果您指定自定义的面板设计，则可有最多20个按钮配置为“设置位”功能。这些按钮控制的位会在V内存中TD 200参数块的一个保留区内得到分配。

由向导的这个页面，您可以为小键盘按钮控制的每个位指定一个符号名。此向导将基于每个按钮的名称建议一个符号名。您可以

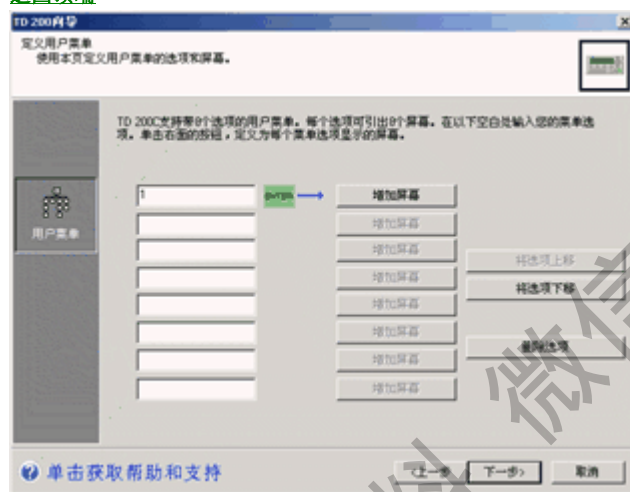
接受这些建议，或单击每个符号名进行编辑。在完成TD 200向导后，这些符号将会在向导创建的符号表中出现。这些符号自此可用于您的程序逻辑，以检测用户对TD 200小键盘按钮的按动。

每个按钮都可配置为设置相应PLC位或作为一个瞬时接触。一旦选取‘设置位’，每当用户按下TD小键盘上的按钮时，相应的V内存位就会得到设置。该位会保持在已设置状态，只能使用程序逻辑加以清除。倘若某按钮被选取为‘瞬时接触’，每当用户按下TD小键盘的该按钮时，相应的V内存位就会得到设置。该位只在此TD面板按钮按下期间处于已设置状态。当用户放开该按钮时，肫湲元 肫内存位就会被清除。

如果您已选取TD 200C作为目标型号，页面上将显示‘选择小键盘...’按钮。此按钮用于选取使用TD小键盘设计员应用程序定义的自定义TD 200C面板。按下‘选择小键盘...’按钮即可载入一Windows‘打开文件’对话框，令您可选取一个小键盘设计员项目文件。在选定面板文件后，此页面将会更新并列出在该面板文件中已定义的按钮。如需关于使用TD小键盘设计员工具创建自定义用户面板的详细说明，请参阅《SIMATIC文字显示 (TD) 用户手册》(SIMATIC Text Display (TD) User Manual)。

## 第6步 定义用户菜单和文字

[返回顶端](#)

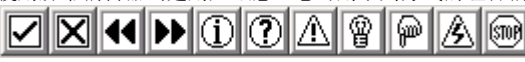



TD 200和TD 200C支持一个将在运行时间由用户控制之下显示的用户菜单。用户能够使用TD 200小键盘上的标准按钮滚动此菜单，然后选取要显示的标题。一旦选取某菜单选项，用户能够使用箭头按钮在该菜单选择的屏恢 湲韶 F聊黍砂 怀 ?0个字符的文字，亦可能包括可编辑的PLC变量和图标。

您可以配置1-8个用户菜单选项，其中每个菜单选项可有1-8个屏幕。此用户菜单的显示由TD用户而不是PLC程序加以控制。如要创建菜单选项，请您希望使用的每个选择输入文字；然后单击‘编辑屏幕’按钮，定义该菜单选项的文字屏幕。



请针对此用户菜单的选择，编辑要显示的信息屏幕（TD 200为40个字符）。请使用屏幕编辑器创建用户讯息。您可用下列方式的组合编写您的讯息：

- 单击工具条  上的某个图标按钮，在屏幕内的当前光标位置插入恒壁急转 D 200C)。请注意，此图标不会在TD 200向显示中实际出现。由这个图标占据的位置将用一特殊颜色显示。
- 使用键盘为屏幕输入文字。
- 嵌入一个PLC内存值；方法为单击  按钮，然后指定您想要在该讯息中显示的数据之地址 C 扛釜聊蛔啮啜砂 ?个PLC数据引用。

您可以复选任何屏幕，表明该屏幕应做为默认的‘启动’屏幕。如果您删除了已选定的默认屏幕，那么将没有默认屏幕。在没有其他用户活动时，默认的启动屏幕将会在TD 200上显示。此默认屏幕也可包括PLC数据和图标。

当键入屏幕文字时，您可以按键盘的Insert键在插入模式和覆盖模式之间切换。屏幕编辑器所包括的微型工具条，提供了剪切、复制、粘贴及删除整个屏幕的功能。讯息内容亦可在菜单标题编辑器和警报编辑器之间移动：



将屏幕的一个副本放在剪贴板内，然后删除此屏幕；屏幕的总数就会减少一个。



将当前屏幕的一个副本放在剪贴板上。



当剪贴板上有TD 200讯息时，‘粘贴’按钮就可使用。粘贴讯息不会将该讯息从剪贴板上移除。此粘贴的讯息会被插入到屏幕列表中当前显示钠聊簧厦姘 U 程 断 11.唛钎聊坏淖茨 皂右桓触灰屹撒没 缙 延凶奋第 浚?) 的屏幕，此操作就无法进行。



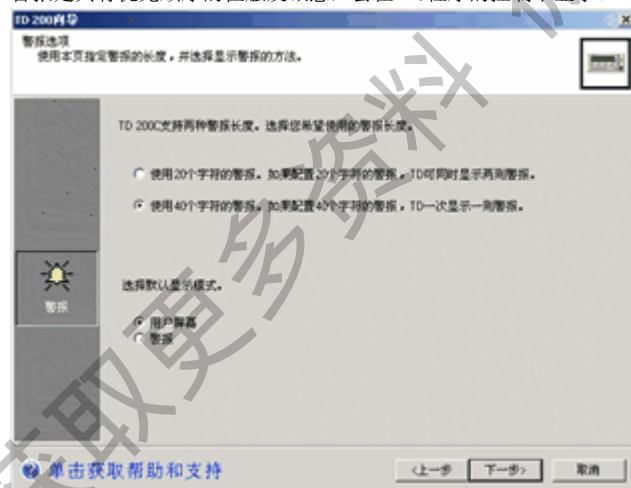
删除此屏幕并返回到菜单或屏幕定义页面，屏幕的总数就会减少一个。

倘若此向导配置定义了多个语言集，那么用户菜单简介页会允许您选取想要编辑的语言集。在这种情况下，您会看到现有语言集的一个下拉列表。当前语言集将为默认选择。请注意，您不能由此创建一个语言集，您只能选择针对某现有语言集编辑此用户菜单。新的语言需在‘语言集’页添加。请用您的主要语言创建菜单和屏幕讯息，然后编辑此菜单和屏幕讯息文字的替代语言版本。您必须使用主要语言增加或删除菜单选项、讯息屏幕及PLC数据引用。

## 第7步 定义警报选项与文字

[返回顶端](#)

警报是具有优先顺序的位触发讯息，会在PLC程序的控制下显示。





请选取所有警报的长度和显示模式。



您可配置多达80条警报。如要创建新警报，请单击‘新警报’按钮，修改识别该警报的默认符号名，然后输入讯息。请使用‘前一个警报’和‘下一个警报’按钮修改现有警报。新警报的创建顺序就是其优先顺序。此优先顺序由‘转到’下拉列表中括号内的警报号表明。

您可用下列方式的组合编写您的警报屏幕：

- 单击工具条  上的某个图标按钮，在屏幕内的当前光标位置插入该图标（例如：TD 200C）。请注意，此图标不会在TD 200向导显示中实际出现。由这个图标占据的位置将用一特殊颜色显示。
- 用键盘输入文字。国际字符表和插入这些字符的ALT键-数字组合请见《SIMATIC文字显示（TD）用户手册》的附录A。
- 嵌入一个PLC内存值；方法为单击  按钮，然后指定您想要在该讯息中显示的数据之地址（例如：M0.0）或PLC数据引用。

当键入屏幕文字时，您可以按键盘的Insert键在插入模式和覆盖模式之间切换。

屏幕编辑器所包括的微型工具条，提供了剪切、复制、粘贴及删除整个警报屏幕的功能。讯息内容亦可在菜单标题编辑器和警报编辑器之间移动：



将警报屏幕的一个副本放在剪贴板内，然后删除此警报屏幕；警报的总数就会减少一个。



将当前警报屏幕的一个副本放在剪贴板上。



在执行粘贴操作时，您将有机会选择插入或覆盖当前警报屏幕。粘贴插入动作会插入一条新警报屏幕，其优先顺序比当前显示之警报高一个。警报的总数会增加一个。粘贴覆盖动作将覆盖当前显示的警报，不会创建新警报屏幕。请注意，剪切和粘贴可用于重新排列警报，以改变其优先顺序。如要变更某警报屏幕的优先顺序，请前往该警报屏幕，然后选取‘剪切’图标，就可将该警报屏幕从其当前位置移除并放在剪贴板上。然后前往该警报的新位置（通过视需要单击‘前一个警报’或‘下一个警报’），并且选取‘粘贴’按钮。当讯息框出现，询问您是要插入或覆盖时，请选取插入。该警报屏幕现就在列表中得到移动并改变其优先顺序。



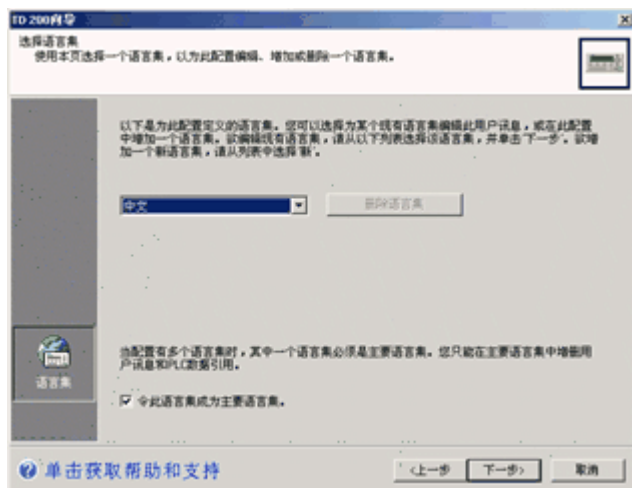
删除此警报屏幕，警报屏幕的总数就会减少一个。

倘若此向导配置定义了多个语言集，那么用户警报简介页面允许您选取想要编辑的语言集。在这种情况下，您会看到现有语言集的一个下拉列表。当前语言集将为默认选择。请注意，您不能由此创建一个语言集，您只能选择针对某现有语言集编辑此用户菜单。新的语言需在‘语言集’页添加。请用您的主要语言创建警报，然后编辑此警报文字的替代语言版本。您必须使用主要语言增加或删除警报及PLC数据引用。

### 第8步 选择要编辑、增加或删除的语言集

#### 返回顶端

如果您定义了用户菜单或位触发警报，“语言集”图标就会出现。此功能可帮助您制作TD 200项目的国际版本。如果您增加某个替代语言并翻译屏幕文字，则用户可用该替代语言显示TD 200的文字。



如要增加新语言，请单击下拉列表并选取‘新语言集’。



然后选取要复制的语言集。您可以将新语言集作为任何现有语言集的副本加以初始化。与此语言集关联的用户讯息就可供翻译。倘若此配置定义了多个语言集，其中一个语言集必须被指定为‘主要’。您只能用主要语言集执行下列操作：

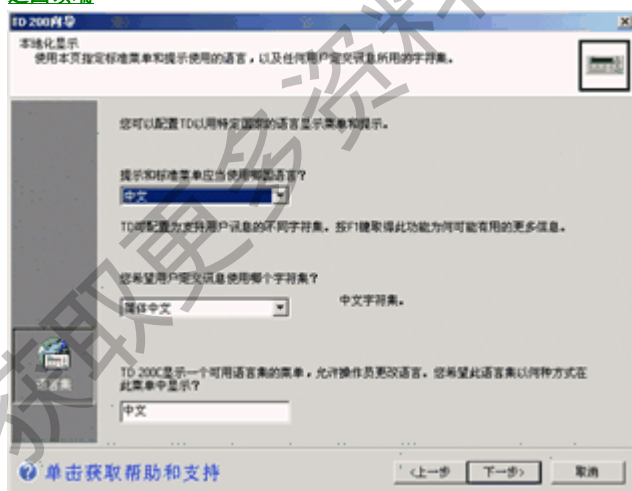
- 更改警报选项和增加或删除警报屏幕。
- 增加或删除用户菜单项。
- 增加或删除用户菜单屏幕。
- 增加、删除或编辑警报或菜单屏幕中的PLC数据引用。

**注释：**

与警报和PLC数据引用关联的符号名适用于所有语言集。举例来说，警报0的符号名对所有语言集都一样。

### 第9步 本地化显示（具有多个语言集）

[返回顶端](#)



**国家语言：**此设定会影响TD 200的预配置菜单和提示。请选取您希望TD 200显示预配置菜单和提示的语言。

**字符集：**此设定决定了您在创建用户讯息时可以使用字符类型。

- 每行讯息最多可显示20个单字节字符。

**注释：**当您在TD 200向导中键入或由STEP 7-Micro/WIN的数据块编辑器检查讯息时，Latin 1 Bold（拉丁1粗体）字符不会显示为粗体；但这些字符会在TD 200显示单元上正确显示（即以粗体出现）。

- 简体中文字符为双字节字符。每行讯息最多可显示10个双字节字符。

**注释：**简体中文字符集中有一些拉丁1字符。取决于您的键盘和（或）仿真器，拉丁1字符既可能为单字节格式，也可能为如同简体中文字符的双字节字符。如果您在同一条讯息中将单字节字符与双字节字符混用，请切记单字节字符的后面只能接单字节字符或空格。



讯息编辑器会显示10个双字节字符位置。每个字符位置可容纳下列组合之一：

- \* 一个双字节字符
- \* 两个单字节字符
- \* 一个单字节字符和一个单字节空格

一个字符位置中不能容纳一个单字节字符加一个双字节字符。

语言集名称

任何配置的第一个语言集将得到一个默认名称；此名称与在向导的‘配置’部分指定的TD 200语言相符。您可以通过在‘选择语言集’部分选取此默认名称并编辑以修改之。

## 第10步 向导部分完成

[返回顶端](#)



配置

[基本配置](#) 选用的第1-5步

此配置部分令您可设定TD设备的参数，包括指定键盘行为、语言选项，以及对PLC功能的保护程度。这属于必需部分。



用户菜单

[菜单](#) 选用的第6步

这个部分令您可为此TD配置定义一个用户菜单。用户菜单并非必需。



警报

[警报](#) 选用的第7步 这个部分令您可定义位触发警报。警报并非必需。

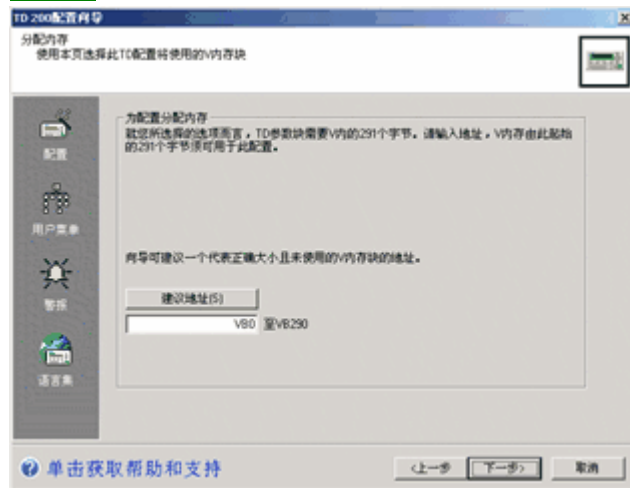


语言集

[语言](#) 选用的第7-8步

如果您定义了警报或用户菜单，“语言集”图标就会出现；这就让您可以定义用户讯息的不同语言翻译。

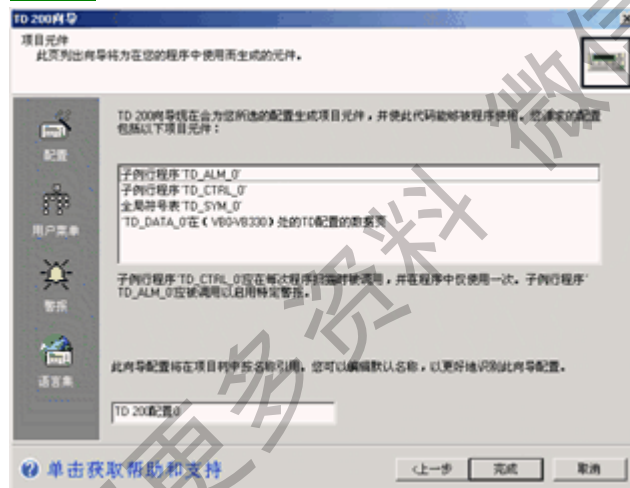
## 第11步 为配置分配内存

[返回顶端](#)

当您将特定配置初次到达此页时，该配置会自动得到建议的V内存地址。此建议地址已针对程序的交叉引用得到检查，确定是未怀疑的。

单击“建议地址”按钮，向导就会寻找下一个（具有足够容量的）可用V内存块。举例来说，假如您输入地址VB300，并且此配置的大小是132个字节，那么“建议地址”功能将会在VB432开始寻找一个V内存块。下一个具有足够容量的可用内存块将得到使用。

(3) 摇 痢 1 坊 岙 恒 隆

**第12步 生成项目元件**[返回顶端](#)**TD 200项目元件**

**TD\_CTRL\_x** （控制TD 200操作）这个子例行程序用于监视和控制TD 200设备的操作，在您的程序中应只调用一次，并且应在每个程序扫描时调用。

**TD\_ALM\_x** （启用警报）如果TD配置定义了警报，那么向导将会生成这个子例行程序，用于启用特定的警报。

**TD\_SYM\_x** （全局符号表）包含与TD 200按钮、警报及PLC数据关联的符号。

**TD\_DATA\_x** （数据块页）用于存储TD 200向导配置的V内存数据。

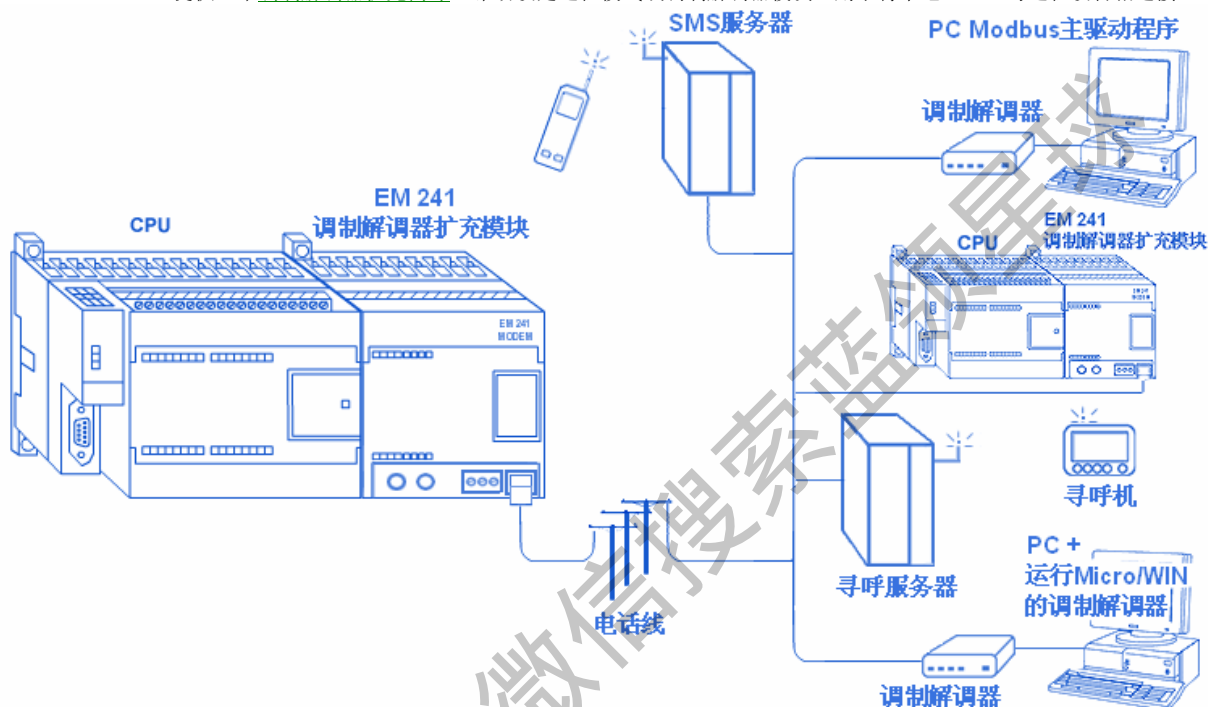
**另请参阅：**

[TD 200 SIMATIC文字显示功能](#)

## 9.8.8 EM241调制解调器模块性能

EM 241调制解调器模块允许您将S7-200直接与模拟电话线连接，并支持S7-200和STEP 7-Micro/WIN之间的通讯。调制解调器模块还支持Modbus从属RTU协议。调制解调器模块与S7-200之间的通讯系通过扩充I/O总线完成。

STEP 7-Micro/WIN提供一个[调制解调器扩充向导](#)，帮助设定远程模式或调制解调器模块，用于将本地S7-200与远程设备相连接。



调制解调器模块是一个智能扩充模块，用于下表显示的S7-200 CPU。

CPU	说明
CPU 222 1.10版或更高版本	CPU 222 DC/DC/DC CPU 222 AC/DC/中继
CPU 224 1.10版或更高版本	CPU 224 DC/DC/DC CPU 224 AC/DC/中继
CPU 226 1.00版或更高版本	CPU 226 DC/DC/DC CPU 226 AC/DC/中继
CPU 226XM 1.00版或更高版本	CPU 226XM DC/DC/DC CPU 226XM AC/DC/中继

调制解调器模块在前侧面板上有8个状态LED。下表是有关状态LED的说明。

LED	说明
MF	模块故障 — 该LED开启时模块探测到一错误状态，例如： • VDC外接电源 • I/O监视定时器超时 • 调制解调器故障 • 无24 与本地CPU通讯错误
MG	模块良好 — 本LED开启时未出现模块错误状态。如果配置表中出现错误或者用户为电话线接口选择了非法国家设置，模块状况LED会闪烁。检查STEP 7-Micro/WIN中的PLC信息屏幕或查阅SMW220中的数值（0号模块槽），了解有关配置错误的信息。
OH	摘机 — 本LED开启时EM 241正在使用电话线。
NT	无拨号音 — 本LED指示出现错误状态，当EM 241接到指令发送讯息但电话线上无拨号音时会开启。只有在EM 241被配置为拨号前检查拨号音时才会显示的错误状态。在拨号尝试之后，LED会保持开启约5秒钟。
RI	铃声指示器 — 本LED指示EM 241正在接受输入呼叫。
CD	载体探测 — 本LED指示已与远程调制解调器连接。
RX	接收数据 — 当调制解调器正在接收数据时，本LED会闪烁。
TX	传输数据 — 当调制解调器正在传输数据时，本LED会闪烁。

调制解调器模块提供以下通讯功能：

- [国际电话线接口](#)
- [用于编程和排除故障的与STEP 7 - Micro/WIN连接的调制解调器接口（通信服务）](#)

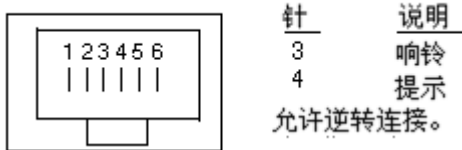
- [Modbus RTU协议](#)
- [支持数字和文字寻呼](#)
- [支持SMS讯息传送](#)
- [CPU至CPU或CPU至Modbus数据传送](#)
- [密码保护](#)
- [安全回叫](#)
- [调制解调器模块配置存储在CPU中](#)

### 国际电话线接口

[返回顶端](#)

调制解调器模块为标准V.34（33.6 kBaud）、10位调制解调器，与大多数内装和外装PC调制解调器兼容。调制解调器模块不与11位调制解调器通讯。您使用安装在模块前侧的六位、四线RJ11接头将EM

241调制解调器模块与电话线连接（见下图）。可能需要一个适配器转换RJ11接头，以便与不同国家的标准电话线终端连接。请



调制解调器和电话线接头由24

VDC电源供电，可连接至CPU传感器电源或外部电源。将调制解调器模块上的接地终端连接至系统接地终端。

当调制解调器接通电源时，调制解调器会根据具体国家的操作自动配置电话接口。位于调制解调器前侧的两个旋转开关用于选择国家。您必须在调制解调器模块电源开启之前将开关设为所需国家选择。请参阅 [国家代码](#) 中所支持国家的开关设置。

### STEP 7-Micro/WIN接口

[返回顶端](#)

调制解调器模块允许您通过电话线与STEP 7-Micro/WIN通讯（通信服务）。您无须对S7-200 CPU进行配置或编程，以便在与STEP 7-Micro/WIN配合使用时将EM 241用作远程调制解调器。

将调制解调器模块与STEP 7-Micro/WIN配合使用时，请遵循以下步骤：

1. 拨下S7-200 CPU电源接头，将调制解调器模块与I/O扩充总线连接。当S7-200 CPU电源开启时，请勿连接任何I/O模块。
2. 将电话线与调制解调器模块连接。如有必要，使用适配器。
3. 将24伏特直流电与调制解调器模块接线盒连接。
4. 将调制解调器模块接线盒接地接头与系统接地接头连接。
5. 设置国家开关。
6. 打开S7-200 CPU和调制解调器模块电源。
7. 配置STEP 7-Micro/WIN，使之与一台10位调制解调器通讯。

### Modbus RTU协议

[返回顶端](#)

您可以配置EM 241调制解调器模块，作为Modbus

RTU从属装置应答。调制解调器模块通过调制解调器接口接收Modbus请求、解释此类请求、并将数据在调制解调器模块和CPU之间传输。然后调制解调器模块生成Modbus应答，并将其通过调制解调器接口输出。

**注释：**如果调制解调器模块被配置作为Modbus RTU从属装置应答，STEP 7-Micro/WIN无法通过电话线与调制解调器模块通讯。

调制解调器模块支持以下Modbus功能：

功能	说明
01	读取线圈（输入）状态
02	读取输入状态
03	读取保持寄存器
04	读取输入（模拟输入）寄存器
05	写入单个线圈（输出）
06	预设单个寄存器
15	写入多个线圈（输出）
16	预设多个寄存器

Modbus功能4和16在一次请求中最多可允许读取或写入125个保持寄存器（250字节V内存）。功能5和15写入至CPU的输出图像寄

Modbus地址通常写为5或6个字符的数值，包含数据类型和偏移。第一个或第二个字符决定数据类型，最后四个字符在数据类型中选择适当的数值。Modbus主设备将地址映射至正确的Modbus功能。

下表显示调制解调器模块支持的Modbus地址以及Modbus地址与S7-200 CPU地址的映射。

使用STEP 7-Micro/WIN中的调制解调器扩充向导为调制解调器模块建立一个配置块，使之支持 Modbus RTU协议。必须在使用Modbus协议之前将调制解调器模块配置块下载至CPU数据块。

Modbus地址	S7-200 CPU地址
000001 000002 000003 ... 000127 000128	Q0.0 Q0.1 Q0.2 ... Q15.6 Q15.7
010001 010002 010003 ... 010127 010128	I0.0 I0.1 I0.2 ... I15.6 I15.7
030001 030002 030003 ... 030032	AIW0 AIW2 AIW4 ... AIW62
040001 040002 040003 ... 04xxxx	VW0 VW2 VW4 ... VW 2*(xxxx-1)

## 寻呼与SMS讯息传送

### [返回顶端](#)

调制解调器模块支持传送：

- 数字和文字寻呼讯息
- 向手机发送的SMS（简短讯息服务）讯息（受手机供应商支持）。

讯息和电话号码存储在调制解调器模块配置块中，该程序块必须下载至 S7-200

CPU中的数据块中。您可以使用调制解调器扩充向导为调制解调器模块配置块建立讯息和电话号码。调制解调器扩充向导还会建立程序代码，允许您的程序将讯息传送初始化。

### 数字寻呼

数字寻呼使用按键电话的音调向寻呼机发送数字数值。调制解调器模块向请求的寻呼服务拨号，等候语音讯息完成，然后发送与寻呼讯息中的数字对应的音调。寻呼讯息中可使用数字0至9、星号（\*）、A、B、C和D。寻呼机允镜男抢拧 A、B、C和D实际字符并非标准化字符，由寻呼机和寻呼服务供应商决定。

### 文字寻呼

文字寻呼允许将字母数字讯息传输至寻呼服务供应商，再由寻呼服务供应商传输至寻呼机。文字寻呼供应商通常设有一条接受文字寻呼的调制解调器线路。调制解调器模块使用电信定位器字母数字协议（TAP）将文字讯息传输至服务供应商。很多文字寻呼供应商使用该协议接收讯息。

### 简短讯息服务（SMS）

某些手机电话服务支持简短讯息服务（SMS）讯息传送，通常为与GSM兼容的手机电话服务。SMS允许调制解调器模块通过模拟电话线将讯息传送给SMS供应商。SMS供应商然后将讯息还书至手机电话，讯息在电话的文字显酒勉般允尽5髦平雅骸骸?械褂攻縲哦?黄誓帜甘 中 惊 TAP)和通用计算机协议（UCP）将讯息传送给SMS供应

赖D 梢越 SMS讯息仅发送给支持调制解调器线路上的此类协议的SMS供应商。

### 文字和SMS讯息中的嵌入变量

调制解调器模块可在文字讯息中嵌入来自CPU的数据数值，并根据讯息中的一条规格为数据数值制定格式。梢灾付? ∈ 阙竺婧陀顾炳氛 洽皇 约靶 ∈ 阙且恒鲍慷故且恒超汉拧 5庇没C缘蛞 髦髦平雅骸骸? 样 湮淖

讯息时，调制解调器模块从CPU检索讯息、确定讯息中所需的CPU数值、从CPU检索这些数值、并在将讯息传输给服务供应商之前制定格式和将数值放入文字讯息中。讯息传送供应商的电话号码、讯息和嵌入讯息的变量均通过多个CPU扫描周期从CPU读取。您的程序不应当在发送讯息时修改电话号码或讯息。可以在发送讯息的过程中继续更新嵌入讯息中的变量。如果讯息包含多个变量，这些变量通过CPU的多个扫描周期读取。如果您希望一则讯息中嵌入的所有变量保持一致，则在发送讯息后不得更改任何嵌入变量

## 数据传送

### [返回顶端](#)

调制解调器模块允许您的程序将数据通过电话线传输至另一台CPU或Modbus设备。数据传送和电话号码用调制解调器扩充向导配置，并存储在调制解调器模块配置块中。然后配置块被下载至 S7-200

CPU中的数据块中。调制解调器扩充向导还会建立程序代码，允许您的程序将数据传送初始化。

数据传送可以是远程设备读取数据请求、将数据写入远程设备的请求或读取和写入数据的请求。数据传送可以读取和/或写入1至100个字的数据。数据传送将数据输入和输出所连接CPU的V内存。

调制解调器扩充向导允许您建立一个数据传送，该数据传送包含从远程设备的单一读取、至远程设备的单一写入或二者兼备。

数据传送使用调制解调器模块的配置协议。调制解调器模块的配置支持PPI协议（该协议对STEP 7-

Micro/WIN作出应答），调制解调器模块使用PPI协议传输数据。如果调制解调器模块的配置支持Modbus

RTU协议，则使用Modbus协议传输数据传送。

远程设备的电话号码、数据传送请求和传送的数据通过多个CPU扫描周期从CPU读取。讯息正在传送时您的程序不应当修改电话怕扁蜒断 i A 硇猱 断(7) 洼 褪踯 挥Φ毙蒂恼 洼 偷氛 寻

如果远程设备是另一个调制解调器模块，数据传送可使用密码功能，方法是在电话号码配置中输入远程调制解调器模块的密码。

回叫功能不得与数据传送一起使用。

## 密码保护

### [返回顶端](#)

调制解调器模块的密码安全为选用功能，可使用调制解调器扩充向导启用。调制解调器模块使用的密码与CPU密码不同。调制解调器模块密码是一个分开的8字符（全部大写字母或数字）密码，在允许呼叫人存取所连接的CPU之前，呼叫人必须向调制解调器? 榷躬 L妹劳舜 8妹劳眺魑 髦平雅骸骸? 榕渣每桐囊徊糠执焮(0) CPU的V内存中。必须将调制解调器模块配置块下载至所连接CPU的数据块中。

如果CPU在系统数据块中启用密码安全功能，呼叫者则必须提供CPU密码，才能存取任何密码保护功能。

## 安全回叫 返回顶端

调制解调器模块的回叫功能为选用功能，用调制解调器扩充向导配置。回叫功能为所连接的 CPU 提供额外的安全保护，方法是仅限从预定义的电话号码存取 CPU。当回叫功能被启用时，调制解调器模块回答所有输入呼叫、核实呼叫人、然后断开线路。如果是经授权呼叫人，调制解调器模块则为呼叫人拨预定义电话号码，并允许呼叫人存取 CPU。

调制解调器模块支持三种回叫模式：

- 向单一预定义电话号码回叫
- 向多个预定义电话号码回叫
- 向任意电话号码回叫

回叫模式是通过在调制解调器扩充向导中勾选适当的选项并定义回叫电话号码的方式选择。回叫电话号码存储在调制解调器模块配置块中，该程序块存储在所连接的 CPU 的数据块中。

回叫的最简单形式是单一预定义电话号码。如果仅在调制解调器模块配置块中存储一个回叫号码，每当调制解调器模块回答一个输入呼叫时，就会通知呼叫人回叫已启用，断开线路，然后拨叫配置块中指定的回叫号码。

调制解调器模块也支持多个预定义电话号码回叫。在该模式中，会要求呼叫人提供一个电话号码。如果提供的电话号码与调制解调器模块配置块中预定义的一个电话号码相符，调制解调器模块则会断开线路，然后使用配置块中的匹配电话号码回叫。用户最多可配置 250 个回叫号码。

当有多个预定义回叫号码时，与调制解调器连接时提供的回叫号码必须与调制解调器模块的配置块中的数目相符（除头两位数字外）。例如，如果配置的回叫为 91(123)4569999（因为需要拨外线（9）和长途（1）），为回叫提供的数字可以是下列任何一个数

- 91(123)4569999
- 1(123)4569999
- (123)4569999

所有上述电话号码均被视为回叫匹配号码。调制解调器模块在执行回叫功能时使用配置块中的回叫电话号码，在上例中为 91(123)4569999。在配置多个回叫号码时，请核实所有的电话号码均为独特的号码（除头两位数字外）。在比较回叫号码时，仅使用电

昂怕脍械氛 肿址 T 谄冉匣咏泻拍脞保 黍悸嵌汉呕蛸e胖 嗟淖址

对任何电话号码的回叫在调制解调器扩充向导中设置，方法是在回叫配置过程中选择“启用向任何电话号码回叫”选项。如果该选项被选择，调制解调器模块回答输入呼叫，并要求提供一个回叫电话号码。呼叫人提供该电话号码后，调制解调器模块则断开

线路，并拨叫该电话号码。该回叫模式仅提供一种向调制解调器模块的电话连接收取电话费用的方法，而不为 S7-200

CPU 提供任何安全保障。如果使用此种回叫模式，则应当使用调制解调器模块密码。

调制解调器模块密码和回叫功能可以同时启用。调制解调器模块要求呼叫人在执行回叫之前提供正确的密码。

## EM 241 配置表

### 返回顶端

所有的文字讯息、电话号码、数据传送信息、回叫号码和其他选项均存储在调制解调器模块配置表中，该表必须载入 CPU 的 V 内存小 5 毫平獾骸酪 尸蚯蚯蛸 寄 5.毫平獾骸酪? 榕渲帽恚 STEP 7-Micro/WIN 然后将调制解调器模块配置表放入下载至 S7-200 CPU 的数据块中。

调制解调器模块在启动时和 CPU 的任何“停止向运行”过渡的五秒钟内读取该配置表。只要调制解调器模块与 STEP 7-

Micro/WIN 联机，调制解调器模块就不会从 CPU 读取新的配置表。如果在调制解调器模块联机时下载新配置表，在联机通话结束

保 毫平獾骸酪? 榛幽寥 〇 厦渲帽恚

如果调制解调器模块探测到配置表中的错误，位于模块前侧的模块良好（MG）LED 会闪烁。检查 STEP 7-

Micro/WIN 中的 PLC 信息屏幕，或读取 SMW220（用于 0 号模块槽）中的数值，了解有关配置错误的信息。下表列出了调制解调器

? 榕渲么碧蜈日缙 褂玫毫平獾骸酪 尸蚯蚯蛸 5.毫平獾骸酪? 榕渲帽恚 STEP 7-Micro/WIN 会在建立配置表之前检查数据。

### 错误

### 说明

0000	无错误
0001	无 24 VDC 外接电源
0002	调制解调器故障
0003	无配置块 ID — 配置表启动时的 EM 241 识别对本模块无效。
0004	配置块超出范围 — 配置表指针没有指向 V 内存，或配置表的某些部分超出所连接 CPU V 内存的范围。
0005	配置错误 — 回叫启用，回叫电话号码数目等于 0 或大于 250。讯息数目超过 250。讯息电话号码数目超过 250，或讯息电话号码的长度超过 120 个字节。
0006	国家选择错误 — 两个旋转开关上的国家选择不是受支持的数值。
0007	电话号码过大 — 回叫启用，回叫号码长度超过最大限度。
0008 至 00FF	保留
01xx	回叫号码 xx 中出现错误 — 回叫电话号码 xx 中出现非法字符。第一个回叫号码的数值 xx 将是 1，第二个回叫号码的数值将是 2，其余依此类推。
02xx	电话号码 xx 中出现错误 — 讯息电话号码 xx 或数据传送电话号码中的一个域包含一个非法数值。第一个电话号码的数值 xx 将是 1，第二个电话号码的数值 xx 将是 2，其余依此类推。
03xx	讯息 xx 中出现错误 — 讯息或数据传送号码 xx 超过最大长度限制。第一则讯息的数值 xx 将是 1，第二则讯息的数值 xx 将是 2，其余依此类推。
0400 至 FFFF	保留

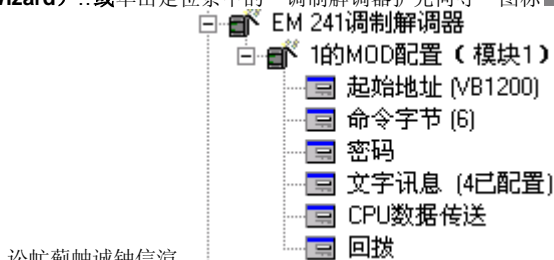
另请参阅：  
[调制解调器扩充向导概述](#)

### 9.8.9 EM241调制解调器模块和调制解调器向导概述

欲使用调制解调器扩充向导，请遵循以下步骤：

1. 选择菜单指令工具 (**T**ools) > 调制解调器扩充向导 (**M**odem Expansion

**Wizard**) ..或单击定位条中的“调制解调器扩充向导”图标  打开向导或打开指令树中的Wizards文件夹并随后打



讼蚯蓟蚰诚钟信渲 ..

2. 单击“下一个”按钮前往调制解调器扩充向导的每个新页。
3. 当您完成配置时，单击“完成”按钮。

调制解调器扩充向导允许您对以下任意一个项目进行编程：



- 用于S7-200 PLC的模拟或手机调制解调器
- EM 241调制解调器模块的参数和操作

您希望配置哪一个项目？

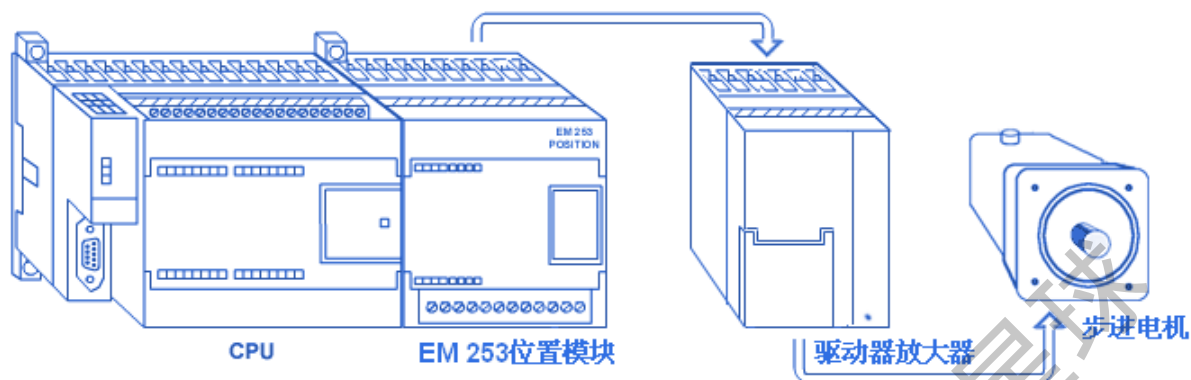
[模拟或手机调制解调器](#)

[EM 241调制解调器扩充模块](#)

### 9.8.10 EM253位置模块性能

EM 253位置模块是一种S7-200特别功能模块，生成用于步进电机或伺服电机速度和位置开环控制装置的脉冲串。该模块通过扩充 I/O总线与S7-200通讯，并在I/O配置中显示为配备八个数字输出的智能模块。

根据存储在S7-200 V内存中的配置信息，位置模块生成控制移动所需的脉冲串。



为了简化应用程序中位置控制的使用，STEP 7-Micro/WIN提供一个[位置控制向导](#)，该向导允许您在几分钟时间内配置位置模块。STEP 7-Micro/WIN还提供一个[控制面板](#)，该控制面板允许您控制、监管和测试位置操作。

位置模块提供单轴开环位置控制所需的功能和性能：

- 高速控制，范围从每秒12次脉冲至每秒200,000次脉冲
- 跳动（S曲线）或线性加速 / 减速
- 测量系统配置的工程单位或脉冲
- 可配置间隙补偿
- 绝对位置控制
- 相对位置控制
- 手动位置控制
- 连续操作
- 最多为25个位置轮廓，每个轮廓最多有四种速度变化
- 四个参考点查找模式（每个序列有起始查找方向和最终接近方向选择）
- 可移动域接线接头

可使用STEP 7-Micro/WIN建立位置模块使用的所有配置和轮廓信息。该信息用您的程序块下载至S7-200。因为所有用于位置控制的信息均存储在S7-200，您可以无须重新为模块编程或配置模块即可更换位置模块。S7-200为位置模块接口保留8位程序图象输出寄存器（Q内存）。S7-200中的应用程序使用这些位控制位置模块作业。这8个输出位不与位置模块的任何实际域输出连接。

位置模块提供五个数字输入和四个数字输出，这些输入和输出为您的位置应用程序提供接口。请参阅下表。这些输入和输出与

信号	说明
STP	STP输入使模块停止。您可以在位置控制向导内选择希望使用的STP操作。
RPS	RPS（参考点开关）输入为绝对移动操作建立参考点或原位。
ZP	ZP（零脉冲）输入帮助建立参考点或原位。通常电机驱动程序 / 放大器电机每次转动时向ZP发出一次脉冲。
LMT+ LMT-	LMT+ 和 LMT- 输入被用于建立移动的最大限度。位置控制向导允许您配置LMT+ 和 LMT- 输入操作。
P0 P1 P0+, P0- P1+, P1-	P0和P1是控制电机移动和方向的开放式漏极晶体管脉冲输出。P0+、P0-和P1+、P1-是分别提供P0和P1相同功能的差分脉冲输出，同时提供优异的信号质量。开放式漏极输出和差分输出同时为现用。您根据电机驱动程序 / 放大器的接口要求选择使用哪一套脉冲输出。
DIS	DIS是用于禁用电机驱动程序 / 放大器的开放式漏极晶体管输出。
CLR	CLR是用于清除伺服脉冲计数器寄存器的开放式漏极晶体管输出。

另请参阅：  
[位置控制向导概述](#)


### 9.8.11 EM253位置和PTO/PWM向导概述

**注释：**您必须为位置模块建立一个配置 / 轮廓表，以便使模块控制位置应用程序。位置控制向导还允许您脱机建立配置 / 轮廓表。S7-200 CPU连接的情况下建立配置。

欲使用位置控制向导，请遵循以下步骤：

1. 选择菜单命令工具 (I) >



位置控制向导..或单击浏览条中的“位置控制向导”图标打开此向导或打开指令树中的“向导”文件夹并随后

EM 253位置

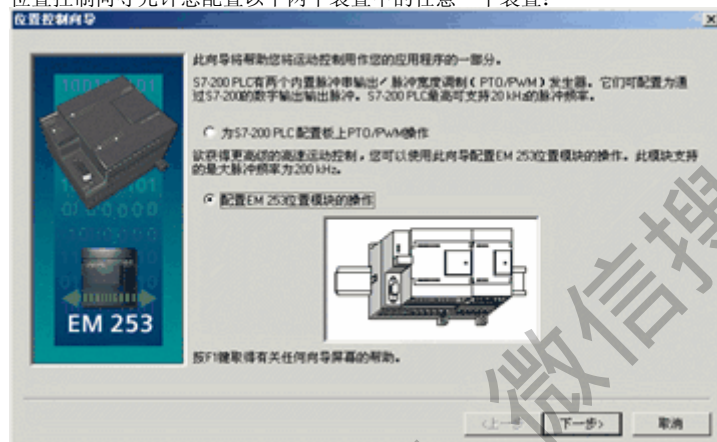
- 2的POS配置 (模块0)
  - 起始地址 (VB1000)
  - 命令字节 (6)
  - 测量单位 (cm)
  - 电机配置
  - 参考点设置
  - 运动轮廓 (2已配置)

PTO/PWM

- Q0.0的PTO配置 (输出Q0.0)
  - 起始地址 (VB5000)
  - 脉冲操作 (PTO)
  - 电机配置
  - 运动轮廓 (1已配置)

- 单击“下一步 >”按钮前往位置控制向导的每个新页。
- 完成配置后单击“完成”。

位置控制向导允许您配置以下两个装置中的任意一个装置：



机载PTO/PWM发生器  
PLC有两个可以配置用于输出脉冲串输出的内置脉冲串输出（PTO）/脉冲宽度调制（PWM）发生器。支持的最大脉冲速率为20kHz。



- EM 253位置模块  
253位置模块。模块支持的最大脉冲速率为200kHz。

如果您需要更高级的高速位置控制，则可配置EM

您完成的配置被存储在V内存内的配置/轮廓表中。位置模块使用的配置和轮廓信息是STEP 7-Micro/WIN项目的一部分，被下载至S7-200。因为位置控制所需的所有信息均存储在S7-200中，取代位置模块则与取代任何其他S7-200 I/O模块一样简便。

STEP 7-Micro/WIN还提供一个EM 253控制面板。该控制面板可简化您的应用程序位置控制部分的测试，方法是提供允许您控制、监控和测试位置操作的功能。EM 253控制面板允许您编辑向导设置，而无须通过向导屏幕返回。

欲存取EM 253控制面板：

- 单击定位条中的EM 253控制面板图标，或
- 选择菜单命令工具 (Tools) > EM 253控制面板... (EM 253 Control Panel)

您希望配置什么？

[PTO或PWM发生器](#)

[EM 253模块](#)

## 9.8.12 EM253控制板

为了协助您开发您的位置控制解决方案，STEP 7-Micro/WIN提供EM 253控制面板。操作、配置和诊断程序标记便于您在开发程序的启动和测试阶段监控和控制位置模块的操作。使用EM 253控制面板核实位置模块接线正确、调整配置数据以及测试每个移动轮廓。



**提示：** 在使用EM

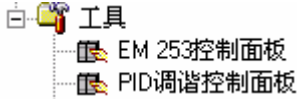
253控制面板之前，将位置控制项目下载至PLC。因为位置控制向导对程序块、数据块和系统块作出改动，请勿必下载所有三种模块。否则，位置模块可能没有正常操作所需的所有程序元件。在模块和控制面板开始作业之前，必须初始化模块的SM内存。当您用1.2.1或更高版本的固件将数据块和系统块下载至CPU时，初始化会自动完成。如果您有1.2版或更早版本的CPU，则必须执行POSx\_CTRL子例行程序，初始化特殊内存。位置控制向导将配置和轮廓数据存储于PLC V内存中。当您启动EM 253控制面板时，Micro/WIN尝试从PLC V内存读取有效配置数据。如果Micro/WIN无法与PLC通讯，您会收到一则通讯错误。如果V内存不包含有效位置配置，Micro/WIN会显示一则V内存配置错误。

请使用下列方法之一打开EM 253控制面板：

- 选取**工具(I)** > **EM 253控制面板** 菜单命令。



- 单击**浏览条**上的EM 253控制面板按钮。
- 打开**指令树**上的工具文件夹，然后打开EM 253控制面板。



操作标记



控制面板的操作标记允许您与位置模块操作对话。控制面板显示位置模块的当前速度、位置和方向。您还可以看到输入和输出 LED 的状态（除脉冲LED外）。

控制面板允许您与位置模块对话，方法是更改速度和方向、停止和启动工具以及使工具慢进（如果移动停止）。

您还可以生成下列位置命令：

- 启用手动操作 — 该命令允许您使用手动控制为工具定位。
- 运行位置轮廓 — 该命令允许您选择需要执行的轮廓。控制面板显示由位置模块执行的轮廓状态。
- 查找参考点 — 该命令使用配置的搜索模式查找参考点。
- 载入参考点偏移 — 在您使用手动控制使工具慢进至新的零位时，则可载入参考点偏移。
- 重新载入当前位置 — 该命令更新当前位置数值。
- 激活DIS输出并取消激活DIS输出 — 这些命令打开和关闭位置模块的DIS输出。
- 脉冲和CLR输出 — 该命令在位置模块的CLR输出中生成一个50毫秒的脉冲。
- 示教位置轮廓 — 该命令允许您在以手动方式放置工具时保存位置轮廓和步进的目标位置和速度。控制面板显示位置模块正在执行的轮廓状态。
- 载入模块配置 — 单击执行使模块从V内存读取配置。
- 移至绝对位置 — 该命令指定一个目标速度和移动的绝对位置。该选项要求定义0位置。
- 移动相对数量 — 该命令指定一个目标速度和您希望从当前位置移动的距离。您可以指定一个正向距离或负向距离。
- 重设模块命令接口 — 如果模块似乎不对命令作出应答，该命令会十分有用。该选项清除模块的命令字节，并设定 DONE位。

#### 配置标记



控制面板的配置标记允许您检视和修改存储在 S7-200 数据块中的位置模块的当前配置设置。

您必须选择 PLC 复选框中的“允许更新模块配置”，才能修改当前配置设置。

修改配置设置后，单击“更新配置”按钮，更新 STEP 7-Micro/Win 项目和 S7-200 数据块中的设置。

#### 诊断程序标记

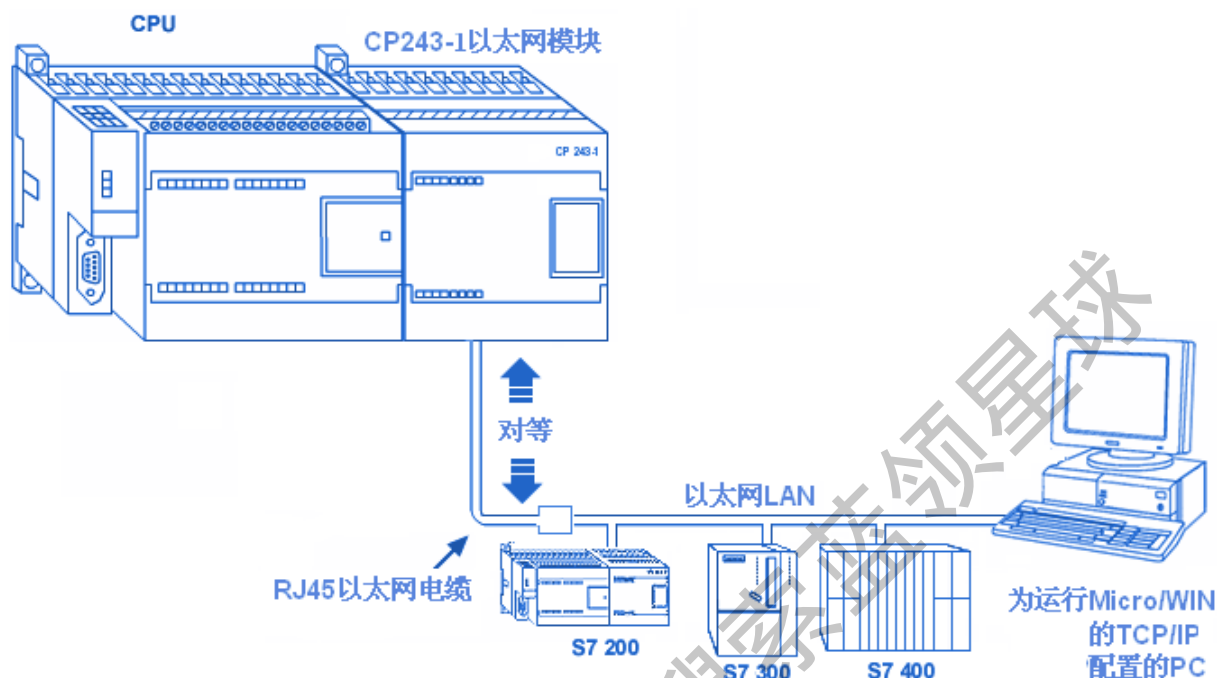


控制面板的诊断程序标记允许您检视有关位置模块的诊断信息。您可以检视有关位置模块的具体信息，例如 I/O 链接中的模块位置、模块类型和固件版本号以及用作模块命令字节的输出字节。控制面板显示因执行命令操作而导致的错误状况。您还可以检视位置模块报告的任何错误状况。

另请参阅：  
[位置模块的错误代码](#)

### 9.8.13 CP243-1 以太网模块性能

CP243-1 以太网模块是 S7-200 系列的通讯处理器，可使 S7-200 PLC 与工业以太网网络链接。以太网模块使用存储在 S7-200 V 内存中的配置信息生成以太网通讯所需的连接。



以太网模块包括以下功能：

- 根据TCP/IP和ISO标准进行通讯
- 工厂安装MAC地址
- 与其他S7设备的对等通讯
- 自动传感全双工或半双工通讯，10 MB和100 MB
- 多个连接（最多为8个）
- 客户机或服务配置选项
- 初始化、重新配置和数据传输指令（ETHx\_CTRL、ETHx\_CFG、ETHx\_XFR）

以太网模块允许S7-200 PLC与S7-300和S7-400设备通讯。您还可以与SIMATIC操作员面板（OP）建立通讯。您通过以太网向导配置以太网模块的通讯参数。

另请参阅：


[以太网向导概述](#)

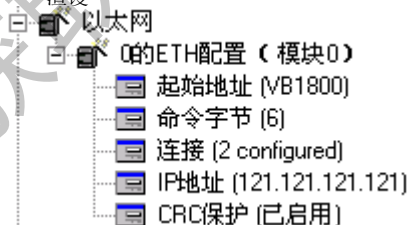
## 9.8.14 CP243-1以太网向导概述

欲使用以太网向导，请遵循以下步骤：

1. 选择菜单命令 **工具 > 以太网向导 (Tools > Ethernet Wizard)** ..



或单击浏览条中的“以太网向导”图标  打开向导或打开指令树中的Wizards文件夹并随后打开此向导或某现有向导

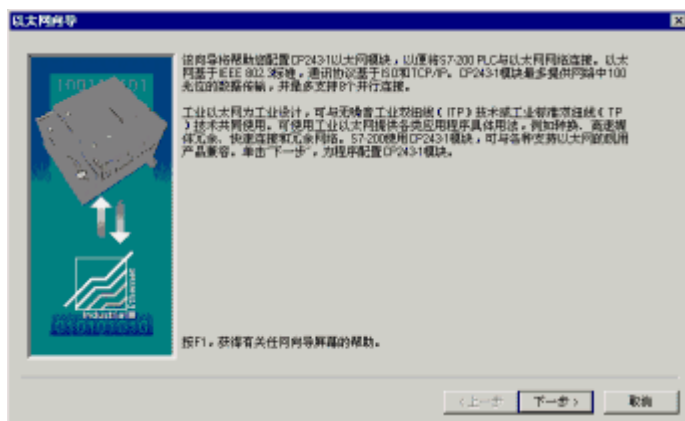


2. 单击“下一步 >”按钮，进入以太网向导的每个新篇章。
3. 完成配置后，单击“完成”按钮。

以太网向导是用于配置以太网模块的主要机制。

以太网向导帮助您配置以太网模块，以便将S7-200 PLC与工业以太网网络连接。然后将本配置存储在适当的数据块区域。您最多可配置8个异步并行连接。但一次只能建立一种此类以太网连接。您必须为您请求的每个连接重复连接配置程序。您一旦完成向导操作并下载配置，以太网模块即使用在S7-200 PLC的SM位置中提供的指针数值存取配置和应用程序数据。

以太网向导还会建立支持对等通讯的独特以太网通讯指令子例行程序。



另请参阅：

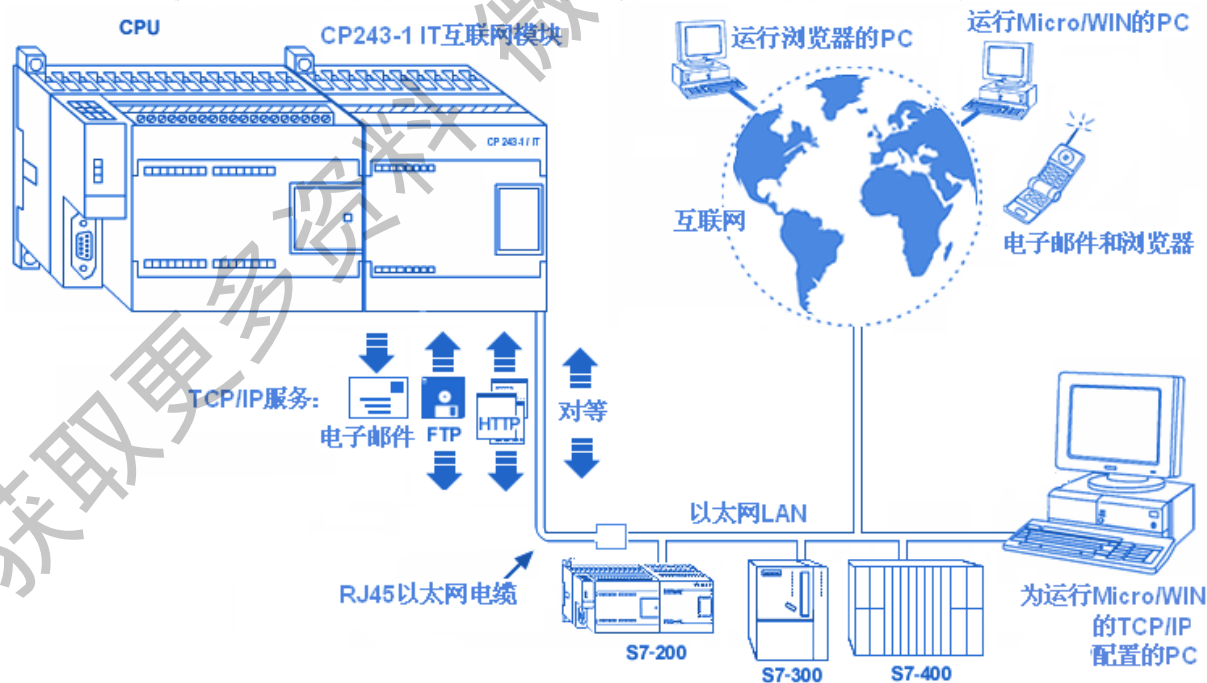
[CP243-1以太网模块性能](#)  
[以太网向导](#)

与以太网相关的指令：

[ETHx\\_CTRL子例行程序](#)  
[ETHx\\_CFG子例行程序](#)  
[ETHx\\_XFR子例行程序](#)

### 9.8.15 CP243-1 IT互联网模块性能

CP243-1 IT互联网模块是用于S7-200系列的通讯处理器，该模块允许S7-200 PLC用标准以太网LAN电缆与互联网连接。



互联网模块支持以下TCP/IP通讯服务。

- 使用用户ID和密码获得保密FTP（文件传输协议）和HTTP（超级文本传输协议）服务器登入
- 将带有嵌入PLC数据的电子邮件讯息发送至标准SMTP邮件服务器
- 将文件传输至远程服务器的FTP客户机服务
- 通过远程FTP客户机将文件传输至内部8 Mbyte快闪存文件系统和从该文件系统接收文件传输的FTP服务器服务
- 用于远程互联网浏览器存取的HTTP服务器服务

- 用于互联网通讯的S7-200程序指令（ETHx\_EMAIL和ETHx\_FTPC）
- 互联网性能的配置数据被直接存入互联网模块的快闪内存。

互联网模块还支持以太网模块提供的所有功能（CP243-1）

- 根据TCP/IP和ISO标准进行通讯
- 工厂安装MAC地址
- 与其他S7设备进行对等通讯
- 自动传感全双工或半双工通讯，10 MB和100 MB
- 多个连接（最多为8个）
- 以太网客户机或服务器配置选项
- 初始化、重新配置和数据传输程序指令（ETHx\_CTRL、ETHx\_CFG、ETHx\_XFR）

储存在V内存中的配置数据被用于生成以太网通讯所需的连接。必须下载向导修改的项目代码块，才能在PLC中操作和更新CP243-1 IT配置。

您必须通过互联网向导配置互联网模块的通讯参数。以太网连接开始操作后，您可以使用Windows DOS FTP命令将三个向导生成的文件复制至CP243-1

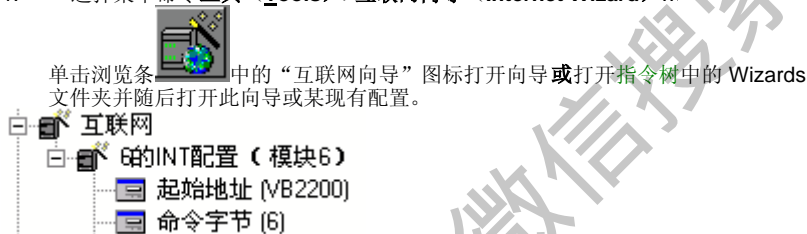
IT快闪内存。这三个文件（\*.edb、\*.fdb和\*.udb）在互联网向导中启用您选择的互联网通讯性能。

有关CP243-1 IT配置程序的详细总结，请参阅[互联网向导概述](#)标题中的内容。

## 9.8.16 CP243-1 IT互联网向导概述

欲启动互联网向导，请遵循以下步骤：

1. 选择菜单命令工具（Tools）>互联网向导（Internet Wizard）...



2. 单击“下一项”按钮，继续执行每一个新“互联网向导”段。
3. 完成配置后，单击“完成”按钮。

### 注释：

互联网模块必须由互联网向导配置，但完成互联网向导操作在总体配置程序中仅需执行一个步骤。请阅读下列步骤，了解完整CP243-1 IT配置程序。

### 先决条件：

运行STEP 7-Micro/WIN的PC必须与以太网连接，才能完成CP243-1

IT配置程序。安装Micro/WIN之前，必须安装PC的网络接口硬件和软件驱动程序。如果您的PC没有网络连接，解除安装STEP 7-Micro/WIN、安装或配置网络硬件和软件，然后重新安装STEP 7-Micro/WIN。

### CP243-1 IT模块配置和互联网通讯步骤

1. 请告诉您的网络管理员，您需要一个新的LAN连接和IP地址。网络管理员会为您指定一个IP地址，并告诉您是否需要使用BOOTP服务器程序。BOOTP服务通常由预先租用的DHCP服务器处理。

如果未使用BOOTP程序，则必须在完成互联网向导程序之前由网络管理员为您指定一个静态IP地址、子网掩模和网关地址。

如果使用BOOTP程序，您必须向网络管理员提供印在模块顶盖下方的独特的MAC地址（例如，08-00-06-70-8C-23）。网络管理员将为您指定一个对应的IP地址。如果其后用一个新模块取代CP243-1 IT模块，网络管理员则必须保留IP地址，用于替换模块的MAC地址。

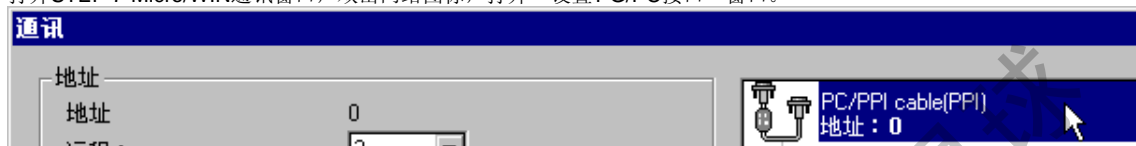
2. 断开CPU电源，将S7-200 CPU连接至CP243-1 IT带状电缆，并将24V电源连接至CP243-1 IT模块（+24V至“L+”和0V至“M”螺旋终端）。取下插入式内存盒（如果安装），“M”螺旋终端，使电源启动重设不会盖写新的以
3. 启动S7-200 CPU和24V电源。用PC/PPI电缆、PC COM卡或以太网TCP/IP（如果使用BOOTP服务）通过集成CPU PPI COM端口建立STEP 7-Micro/WIN与CPU之间的通讯。互联网向导使用该初始通讯路径设置新的以太网通讯路径。该步骤的其余部分假设您在
4. 打开STEP 7-Micro/WIN项目，运行互联网向导。完成向导屏幕操作，并单击“完成”按钮。单击“是”，完成向导配置，并在程序性
5. 使用“文件>下载”命令，将新代码块从STEP 7-Micro/WIN传输至S7-200 PLC。下载所有的程序块、数据块和系统块。该下载传输设置互联网向导中指定的CP243-1 IT模块IP地址、子网地址、网关地址和管理员数据。
6. 取决于您的互联网模块的初始状态，可能不会自动重新配置。如果不重新配置，执行STEP 7-



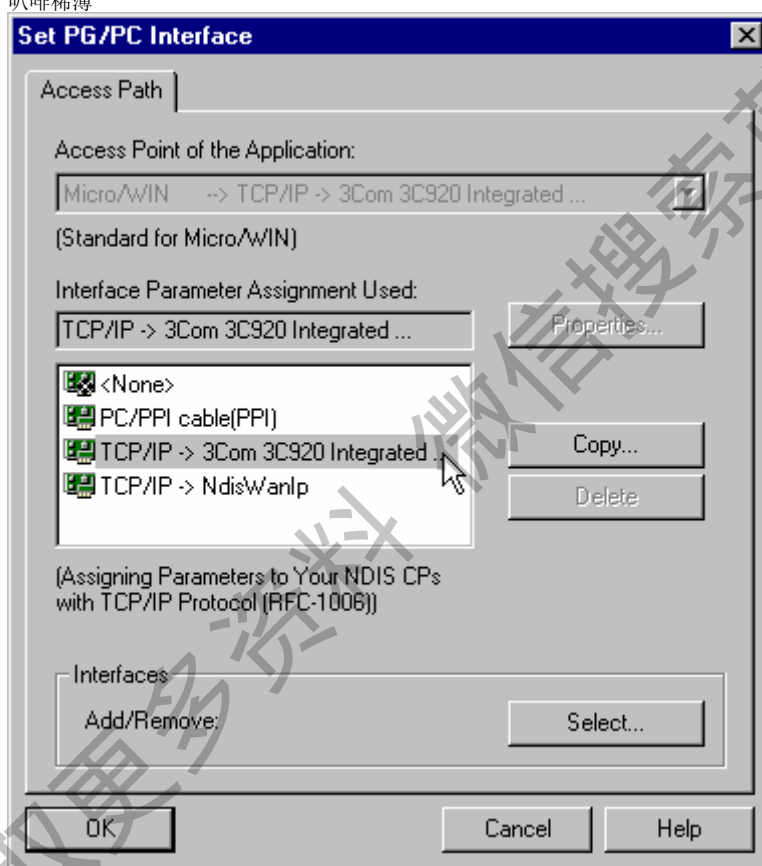
Micro/WIN “PLC>电源启动重设” 软件命令，或关闭PLC电源后再重新开启，初始化CP243-1 IT模块。

7. 用一根RJ45以太网电缆连接CP243-1 IT模块和以太网LAN。运行STEP 7-Micro/WIN的PC必须也有一个以太网LAN，才能完成配置。您可以使用Windows DOS命令提示窗口发出一条“ping xxx.xxx.xxx.xxx”命令。将模块的IP地址替换为xxx.xxx.xxx.xxx，测试PC和CP243-1 IT模块之间的以太网路径。
8. 当PC与CP243-1 IT以太网连接操作时，将STEP7-Micro/WIN至PLC通讯路径从CPU的集成PPI端口更改为CP243-1 IT以太网LAN连接。



打开STEP 7-Micro/WIN通讯窗口，双击网络图标，打开“设置PG/PC接口”窗口。

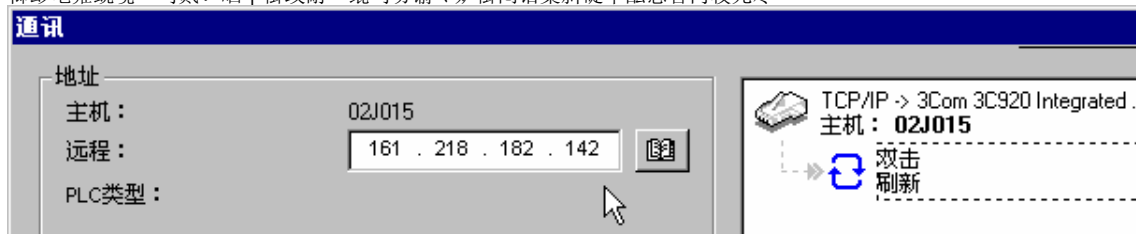


用鼠标单击PC使用的TCP/IP设备，选择PG/PC存取路径，然后单击“确认”按钮。单击存取路径更改核实窗口中的咖啡稀薄

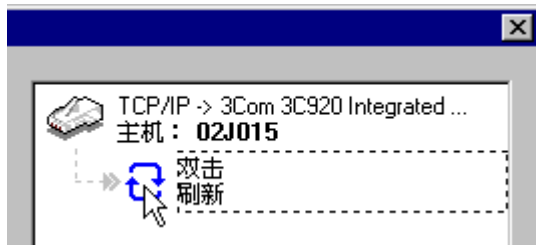


TCP/IP网络设备应当显示在“通讯”窗口的右上方。在“远程”域中输入CP243-1

IP地址。另一种方法是单击书本图标，打开IP地址浏览器，并选择一个以前指定的地址。您还可以用IP地址浏览器刷新图标刷新。



双击网络刷新图标，更新通讯使用期限列表。



刷新的使用期限显示与以太网连接的CPU。核实已选择目标CPU。如下图所示，所选CPU和地址被增高。



单击“确认”，关闭“通讯”窗口。STEP 7-

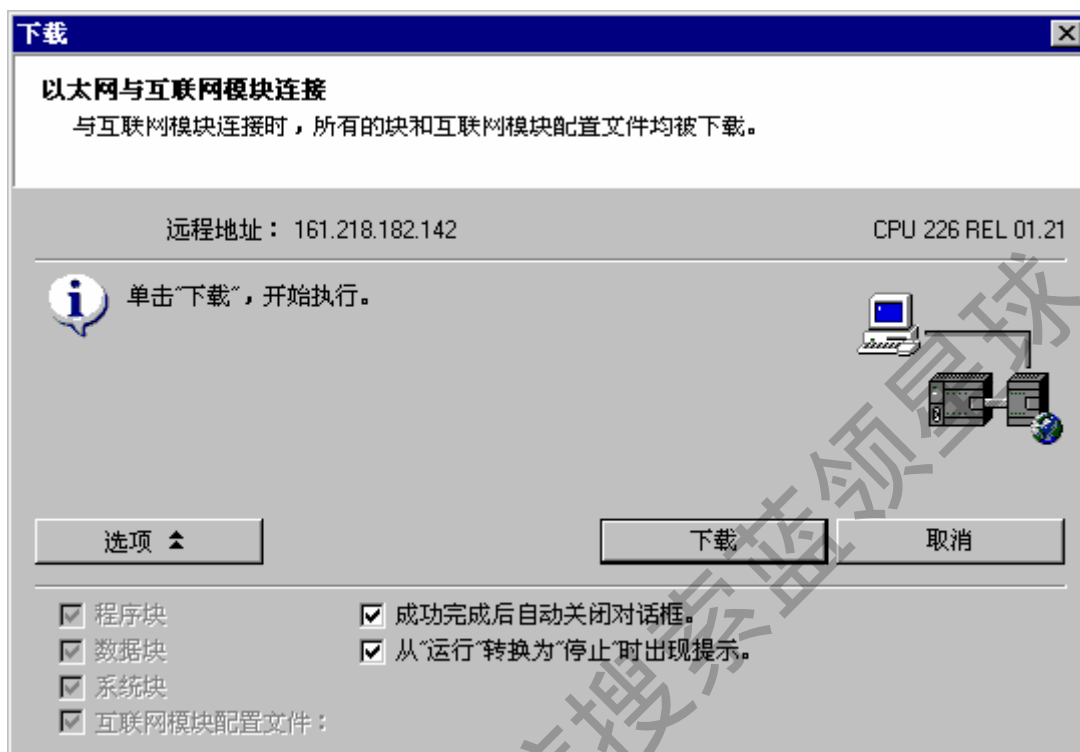
Micro/WIN现在通过以太网LAN通讯，可以移除PPI端口电缆。所有的STEP 7-Micro/WIN性能均通过新以太网路径（运行 / 停止、下载、上载、状态收集等）提供。

9. 使用“文件 > 下载”命令通过以太网路径从STEP 7-Micro/WIN向CP243-1 IT模块传输配置。

成功完成步骤9所需的条件：

- 运行STEP 7-Micro/WIN的PC/PG必须通过以太网与CP243-1 IT模块连接。
- 当前项目中存在一个互联网向导配置。
- 在互联网向导配置中指定的模块位置实际存在一个CP243-1 IT模块。

该下载传输设置在互联网向导中指定的CP243-1 IT模块用户、电子邮件和FTP数据。如下图所示，下载窗口现在指示从STEP 7-Micro/WIN至CP243-1 IT模块的下载路径。



10. 互联网向导已建立操作您选择的性能的子例行程序（ETHx\_CTRL, ETHx\_CFG, ETHx\_XFR, ETHx\_EMAIL和ETHx\_FTPC）。打开指令树中的“调用子例行程序”文件夹，将指令放入程序块中，由程序执行这一功能。

如果您希望更改部分或全部配置参数，重新运行互联网向导：

- 如果您修改以太网配置参数，则必须下载所有的项目块，并重新执行电源启动重设。
- 如果您修改互联网参数（电子邮件、FTP或用户登录），则必须重新执行步骤9，以便重新载入\*.edb、\*.fdb和\*.udb文件。

11. 快闪存文件系统由工厂提供，已预先载入下表所示目录和文件集。作为范例提供的快闪存文件可供您使用或修改。除\_\_S7Sys目录外，所有的目录均可删除。每次CP243-1 IT初始化时，所有的默认目录均会被建立为空目录，并无法找到这些目录。您可以建立自己的网页和小程序，但如果M 媛 S7-200数据，则必须使用所提供的“放置 / 获取” JAVA组件。

编程细节请参阅CP243-1 IT技术参考资料：

用于工业以太网和信息技术的CP243-1 IT通讯处理器

#### SIMATIC NET

#### CP 243-1 IT Communications Processor for Industrial Ethernet and Information Technology

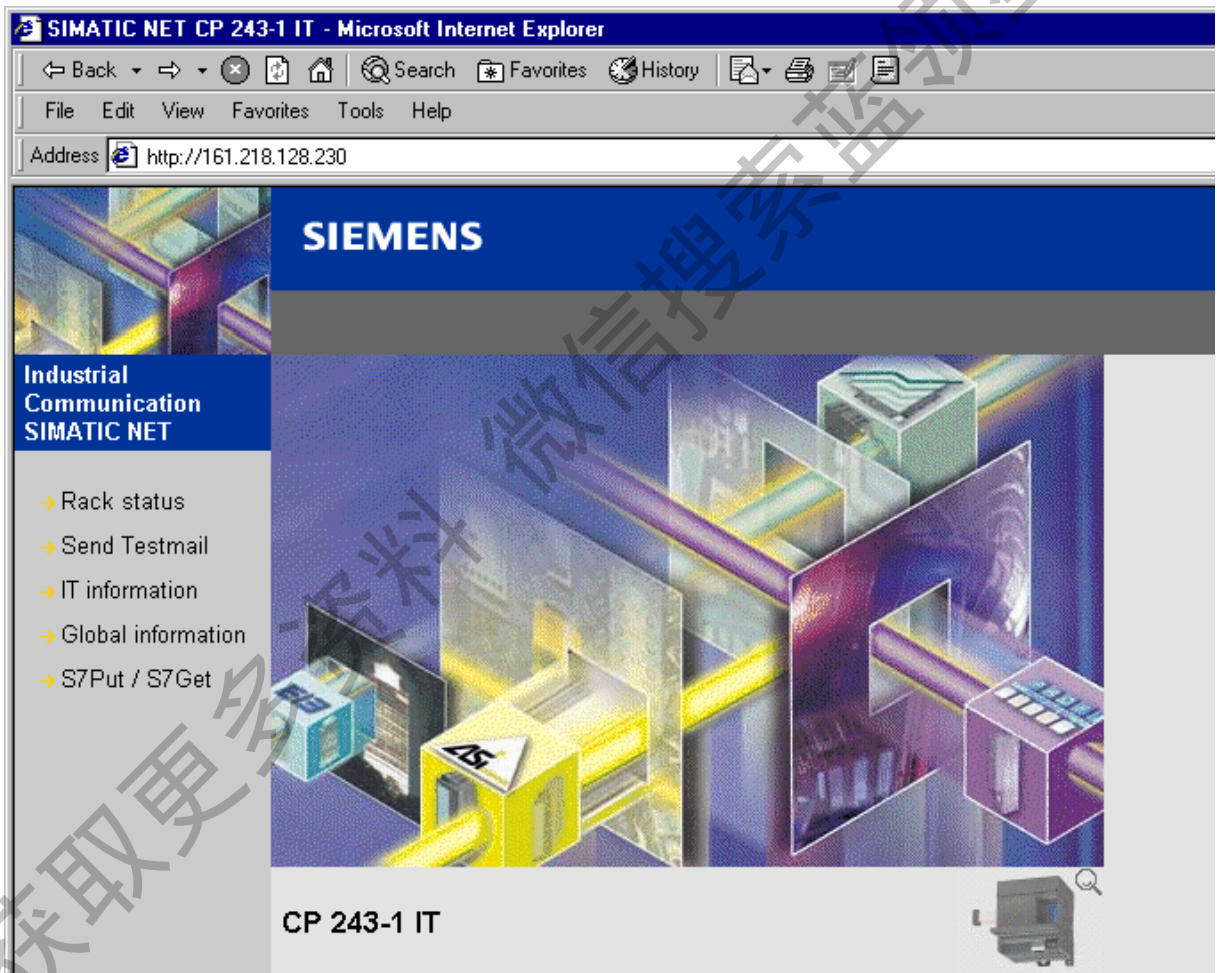
/flash:	快闪存文件系统根目录	
/_S7Sys	（用于FTP服务的读取保护）	内部使用
/images		内部使用
*.gif	（用于状态页）	
/itsdb		内部使用
bin		
sendmail.htm		
cp243_styles.css		
/applets		小程序目录
*.jar		
/config		配置文件目录，最初为空目录

<projectname>.edb  
 <projectname>.fdb  
 <projectname>.udb  
 /examples  
 /images  
 \*.gif  
 \*.jpg  
 \*.htm  
 /user  
 index.htm  
 readme.htm

CP243-1 IT 主页范例 JAVA 小程序和组件范例

用于用户文件

将 CP243-1 IT 模块地址输入互联网浏览器，检视示范网页。



另请参阅：

[CP243-1 IT 互联网模块性能  
互联网向导](#)

与互联网相关的指令：

[ETHx\\_EMAIL 子例行程序](#)  
[ETHx\\_FTFC 子例行程序](#)

与以太网相关的指令：

[ETHx\\_CTRL 子例行程序](#)  
[ETHx\\_CFG 子例行程序](#)

[ETHx XFR子例行程序](#)

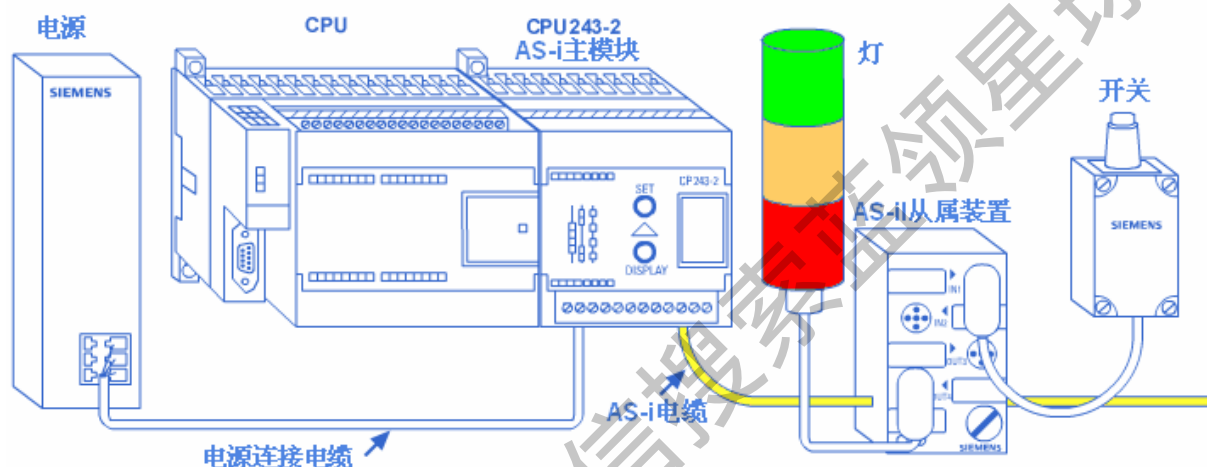
## 9.8.17 CP243-2 AS-i模块性能

CP243-2 AS-i主模块允许将AS-i网络连接至S7-200 PLC。

传动器 / 传感器接口或AS-i是用于最低级别自动化系统的单台主设备网络连接系统。CP243-2模块用作网络的AS-i主设备。传感器和传动器（AS-i从属装置）使用单条AS-i电缆，可通过AS-i主模块与PLC连接。AS-i主模块处理所有AS-i网络协调作业，并通过指定给主模块的I/O地址将来自传动器和传感器的数据和状态信息传送至PLC。AS-i从属装置是AS-i系统的输入和输出信道，只有在被AS-i主模块调用时才会成为现用。

AS-i向导可以为AS-i从属装置地址重新编程，帮助设置AS-i网络。

AS-i向导还可以配置项目，为I/O存取映射AS-i网络的从属装置。



注释：AS-i向导可用于第二代AS-i主模块CP243-2（1.x版或更高版本）。

另请参阅：  
[AS-i向导概述](#)

## 9.8.18 CP243-2 AS-i向导概述

欲使用AS-i向导，请遵循以下步骤：

1. 选择菜单命令 **工具 > AS-i向导** (**Tools > AS-i Wizard**)？或单击浏览条中的“AS-

向导”图标  打开此向导或打开 [指令树](#) 中的Wizards文件夹并随后打开此向导或某现有配置。

AS-i  
的AS配置 (模块0)  
起始地址 (VB100)  
从设备类型

2. 单击“下一步 >”按钮前往AS-i向导的每个新部分。
3. 完成向导中的步骤后，单击“完成”按钮。

配置初始向导选项，以便：

[更改AS-i从属装置地址](#)

选择该选项，设置AS-i从属装置网络，方法是为存储在每个AS-i从属装置中的从属地址编程。欲为从属装置地址编程，您必须在STEP 7-Micro/WIN、连接CP243-2 AS-i主模块的S7-200 CPU和AS-i从属装置网络之间建立通讯。

或

[配置项目，使之映射AS-i从属装置](#)

AS-i向导可帮助您建立在程序和AS-i从属装置之间传输数据所需的Micro/WIN项目代码，并为您提供映射至AS-i I/O位置的符号。

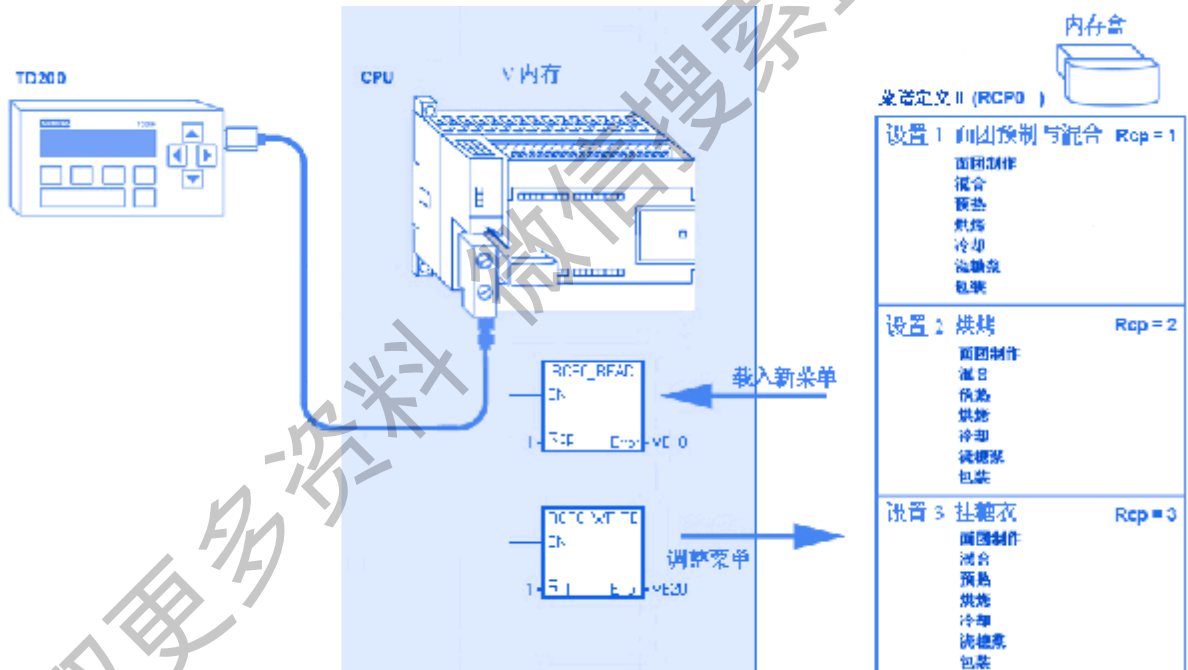
向导的这一部分可联机或脱机使用（未连接S7-200 CPU、CP243-

2或从属装置网络）。如果联机使用向导，向导可提供CP243-2模块信息，并进行AS-i网络比较。



另请参阅：  
AS-i 模块性能

## 9.8.19 菜谱向导



注释：“菜谱向导”可以创建项目代码，用于在64千字节或256千字节永久性内存（EEPROM）盒中存储和获取菜谱数据。对菜谱数据已就绪的机械闸C质加 S7-200 CPU 222、CPU 224、CPU 226第2.0版。

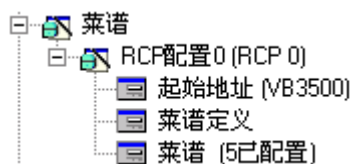
菜谱向导用于在内存盒中组织和载入自动化程序菜谱。将菜谱数据值转移到内存盒内，可以解放否则必需用来存储菜谱数据的 V<sub>A</sub> 注册。P 录伞似锥丁塞筒似资 捣峙淇稍诤似紫虹贾兄笨蛹 瓣6圆似资 荻暮竿 亩 赏u 甌略诵肋似紫虹蓟蚤捐 RCPx WRITE子例行程序的编程来完成。

欲使用菜谱向导，请遵循以下步骤：

1. 选择菜单命令工具 > 菜谱向导 (Tools > Recipe



Wizard) 或单击浏览条中的菜谱向导图标 打开此向导或打开指令树中的 Wizards 文件夹并随后打开此向导或某钟信渲漫



2. 单击"下一步 >"按钮前往菜谱向导的每个新页。
3. 完成向导中的步骤后，单击"完成"按钮。

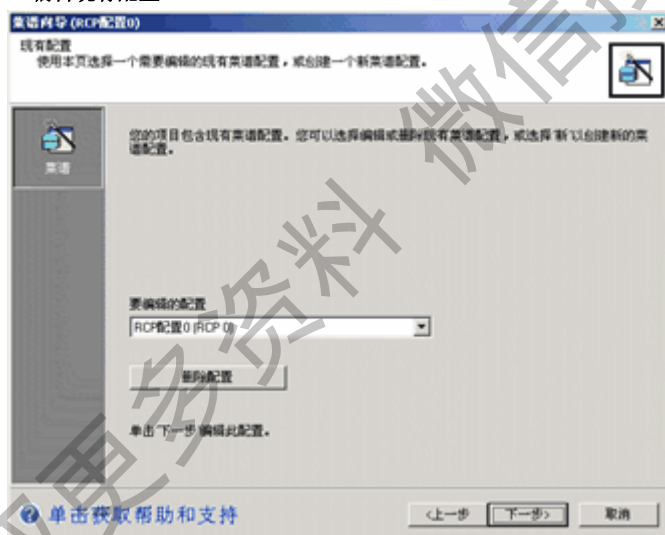
菜谱向导通过生成下列项目元件而创建存取内存盒接口的程序：

- 菜谱符号表，每个菜谱定义一个，用于关联菜谱符号名、V内存地址、菜谱注解
- 新的数据块标记，每个菜谱定义一个，用于为每个菜谱定义分配V内存地址和数值。您的程序可使用对这些V内存地址姆琶创媛（似资 尊
- 子例行程序RCPx\_READ将指定的菜谱数据值从内存盒复制到V内存。RCPx\_READ子例行程序用于将不同的菜谱载入V内存或刷新当前菜谱的V内存数值。
- 子例行程序RCPx\_WRITE将指定的菜谱数据值从V内存复制到内存盒。如果程序在V内存菜谱数据值中写入改动，并且您希望保存为永久菜谱更改，请使用RCPx\_WRITE。

### 步骤

- 1 编辑现有配置
- 2 菜谱定义
- 3 创建和编辑菜谱数据值
- 4 分配V内存
- 5 生成代码

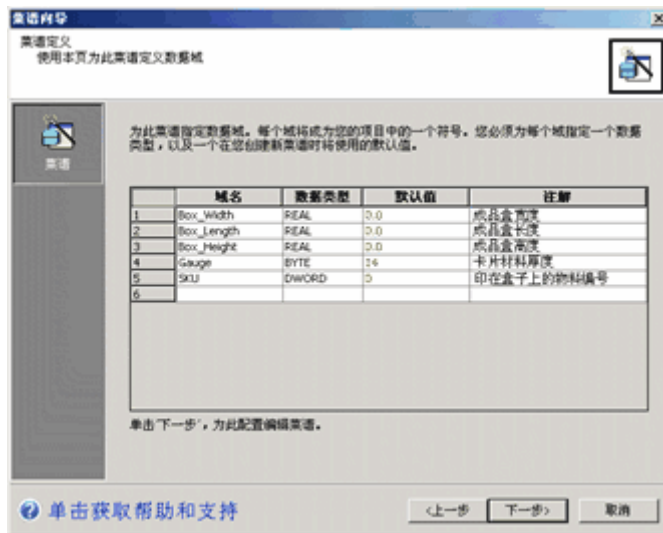
#### 第1步 编辑现有配置



单击配置下拉列表，从中选一个现有或新的菜谱定义。您最多可以有四个不同的菜谱定义。每个菜谱定义中可保存数百个类似住

如果不存在现有菜谱配置，菜谱向导将由第2步开始。

#### 第2步 菜谱定义



用鼠标单击以选取一个表数据单元，然后编辑该域。键入一个符号名作为域名。由下拉式列表选取一个数据类型。为成品盒宽度、成品盒长度、成品盒高度、卡片材料厚度、印在盒子上的物料编号。

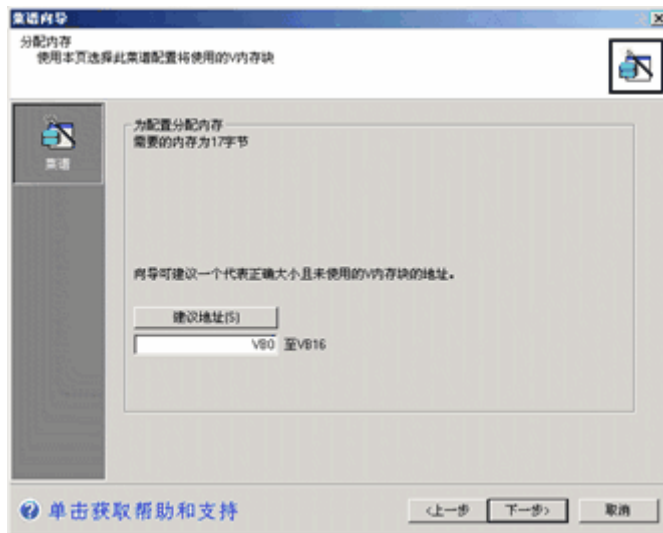
### 第3步 创建和编辑菜谱数据值



双击列标题，重新命名该菜谱，然后编辑默认值以创建新菜谱。单击菜谱数据的任何位置或注释列启用"新"按钮，然后单击"新"按钮插入新的菜谱。重复此过程即可基于当前菜谱定义插入、重新命名、编辑所有菜谱。

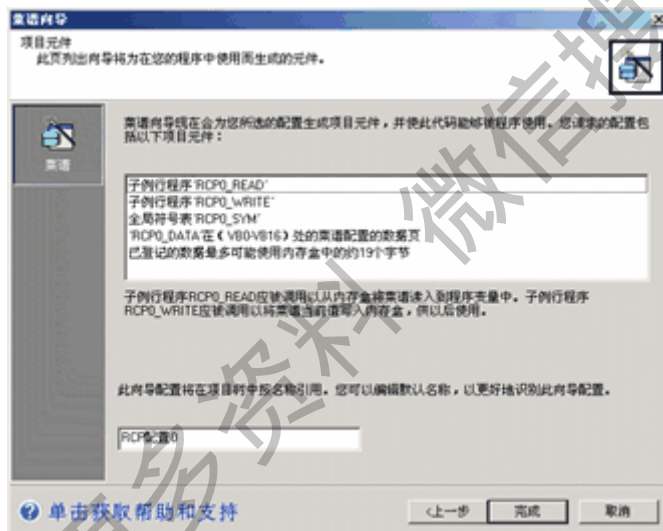
### 第4步 分配V内存





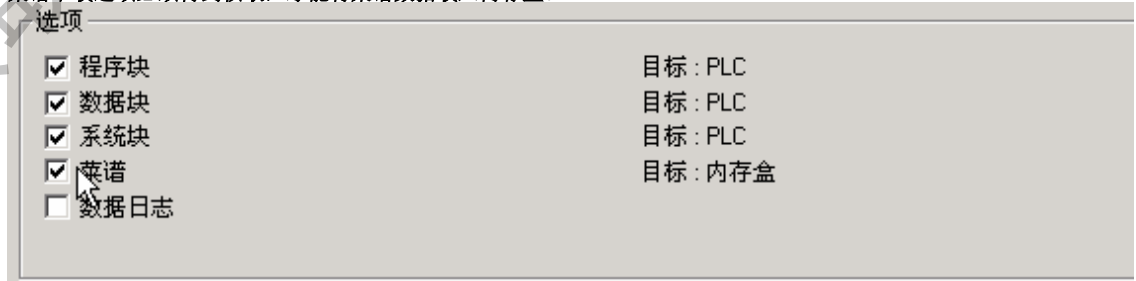
此向导将分配一个"数据块"标记以在V内存中存储菜谱配置。在数据块中指定您希望安放该配置的起始地址。您可以选择想要存放该配置块的V内存地址；如果您希望向导建议正确大小的未使用内存块地址，则单击"建议地址"按钮。配置块的大小根据您在向导中所作的具体选择不同而异。

### 第5步 生成程序代码



菜谱向导会为您的菜谱(符号表、数据块、子例行程序、内存盒数据)生成项目元件，并令您的程序可以使用这些元件。您必须将带有菜谱向导配置的项目下载至 S7-200 才能加以使用。如果某项目有菜谱配置，则下载窗口的默认设置为"菜谱"选项已被核取。

**菜谱下载选项必须得到核取，才能将菜谱数据载入内存盒。**



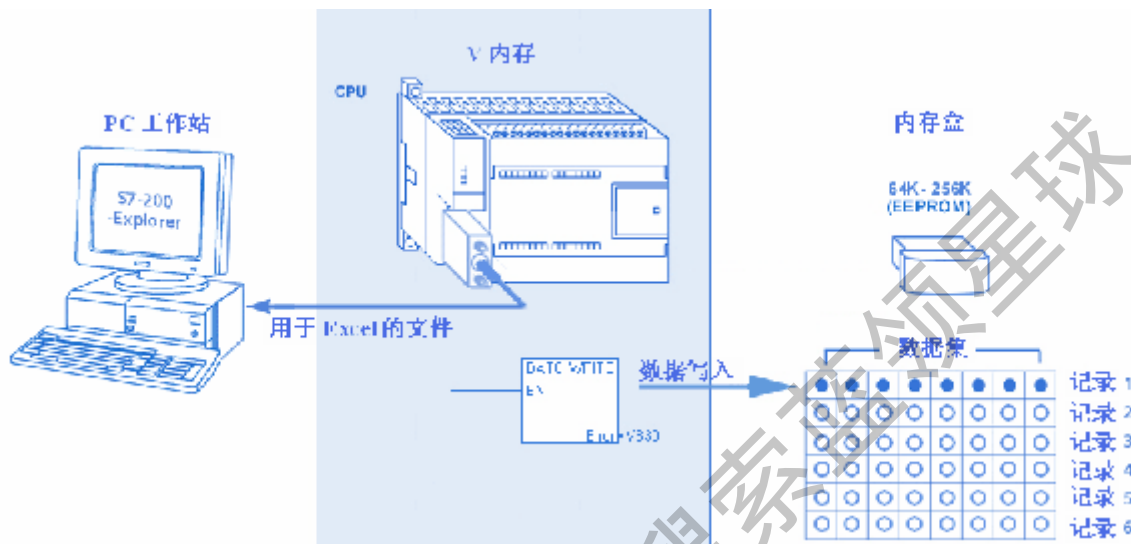
您的程序能够由用菜谱数据符号名指明的V内存地址读取菜谱数据。每执行一次RCPx\_READ就会将一个菜谱从内存盒粗频 V内存。

每执行一次RCPx\_WRITE就会将一个菜谱从V内存复制到内存盒。

另请参阅：

RCPx\_READ  
RCPx\_WRITE

## 9.8.20 数据日志向导



注释：“数据日志向导”可以创建项目代码，用于在64千字节或256千字节永久性内存（EEPROM）盒中存储数据。对数据日志向导已端挂婧械闹 C 质加赞7-200 CPU 222、CPU 224、CPU 226第2.0版。

数据日志向导可帮助您在内存盒内存储工艺测量数据。将工艺数据的存放转移到内存盒内，可以解放否则必需用来存储此数据的V内存地址。记录定义会被输入到数据日志向导内。对记录定义的改动可通过重新运行数据日志向导来完成。

S7-200 Explorer工具能够复制已存储在内存盒内的数据日志记录。使用S7-200

Explorer，找到内存盒图标，选取数据定义，然后上载有关的数据日志记录。由此上载的CSV（逗号分隔值）文件将会打开与CSV文件关联的Windows应用程序（例如MS Excel或MS Notepad）。请使用此Windows应用程序在您的磁盘驱动器上保存新数据日志CSV文件的一个永久副本。

欲使用数据日志向导，请遵循以下步骤：

1. 选择菜单命令工具 > 数据日志向导（Tools > Data Log



Wizard) 或单击浏览条中的数据日志向导图标  打开此向导或打开指令树中的Wizards文件夹并随后打开此向导



2. 单击“下一步 >”按钮前往数据日志向导的每个新页。
3. 完成向导中的步骤后，单击“完成”按钮。

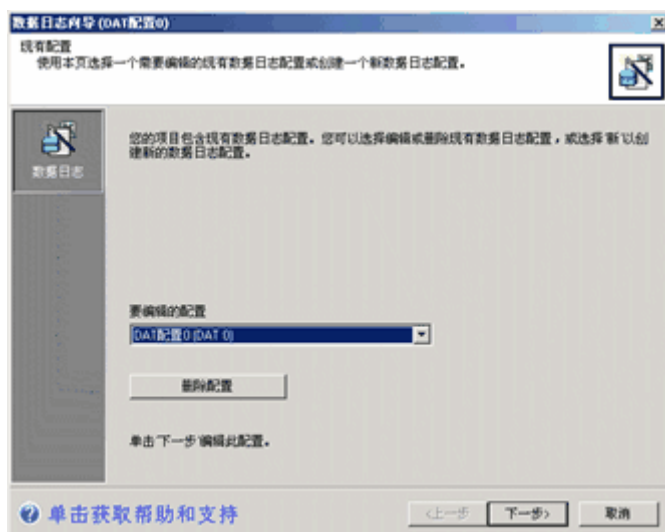
数据日志向导通过生成下列项目元件而创建存取内存盒接口的程序：

- 数据日志符号表，每个数据日志定义一个，用于关联数据日志符号名、V内存地址、数据日志注解
- 新的数据块标记，每个数据日志定义一个，用于为每个记录定义分配V内存地址。您的程序使用这些V内存地址来积累鼻叭罩炯锹肌
- 子例行程序DATx\_WRITE将指定的记录从V内存复制到内存盒。每执行一次DATx\_WRITE就会为已储存在内存盒内的日志数据添加一个新的数据记录。

### 步骤

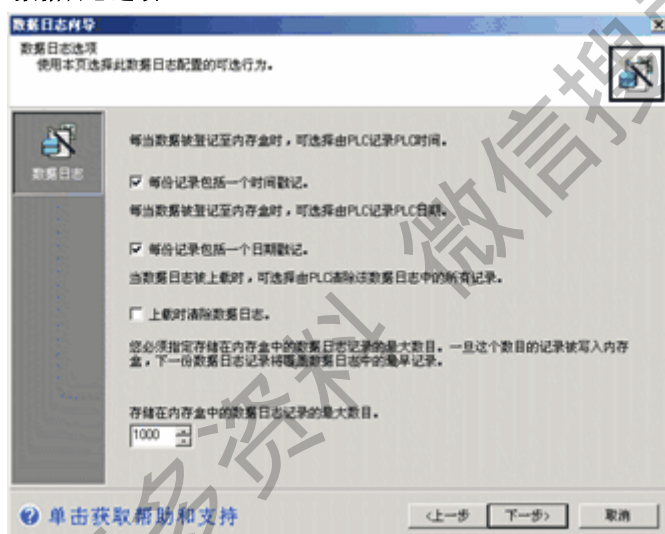
1. 编辑现有配置
2. 数据日志选项
3. 数据日志定义
4. 分配V内存
5. 生成程序代码

#### 第1步 编辑现有配置



单击配置下拉列表，从中选一个现有或新的数据日志定义。您最多可以有四个不同的记录定义。如果不存在现有数据日志配置，数据日志向导将由第2步开始。

### 第2步 数据日志选项



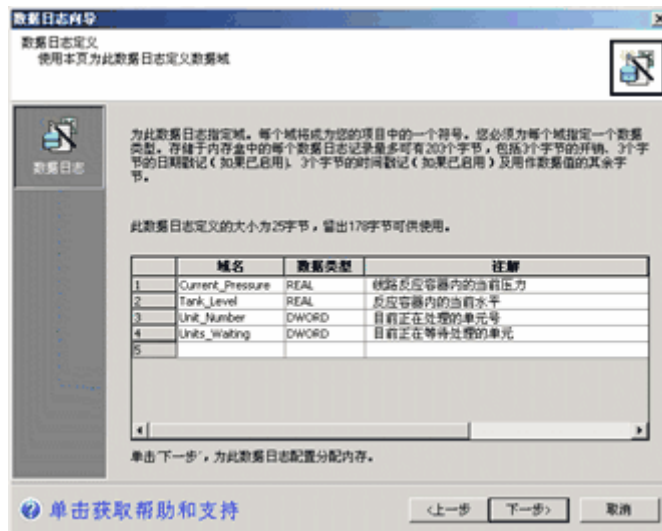
您可以让每份记录包括一个时间戳记。在核取这个选项后，每当用户程序发出数据日志写入命令，CPU就会在每份记录中自动包括一个整数时间戳记。核取此选项会令数据日志记录定义的最大长度缩短3个字节。

每份数据日志记录可添加一个日期戳记。在核取这个选项后，每当用户程序发出数据日志写入命令，CPU就会在每份记录中自动包括一个整数日期戳记。核取此选项会令数据日志记录定义的最大长度缩短3个字节。

您可选择每当数据日志被上载时，清除该数据日志中的全部记录。

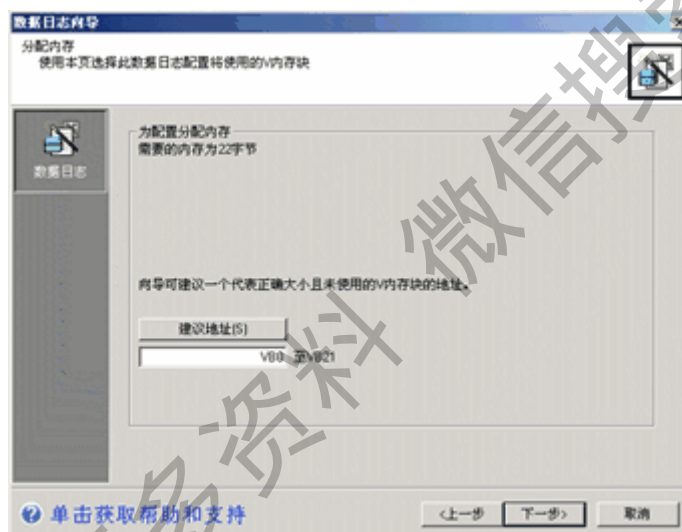
数据日志的形式为循环缓冲区，您必须指定存储在内存盒中数据日志记录的最大数目。此数目为CPU中的一个WORD (字) 值，且必须选取范围 (1-65535)。此选项的默认值为1000。在您下载具有数据日志配置的项目时，内存盒空间会被保留以存储最大数目的记录。

### 第3步 数据日志定义



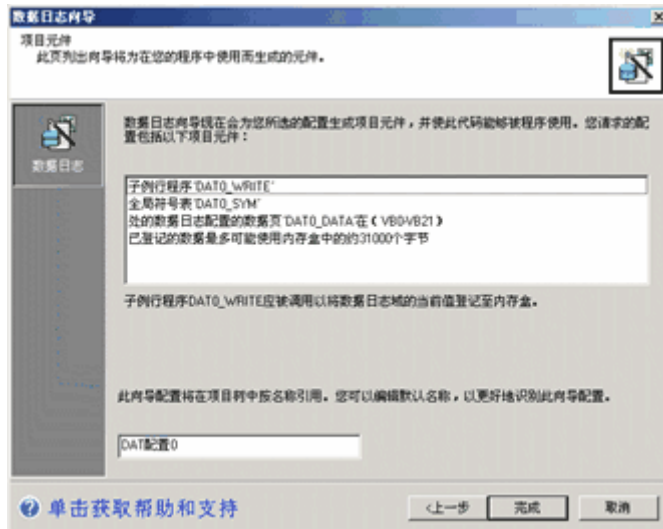
用鼠标单击以选取一个表数据单元，然后编辑该域。键入一个符号名作为域名。由下拉式列表中选取一个数据类型。为

#### 第4步 分配V内存

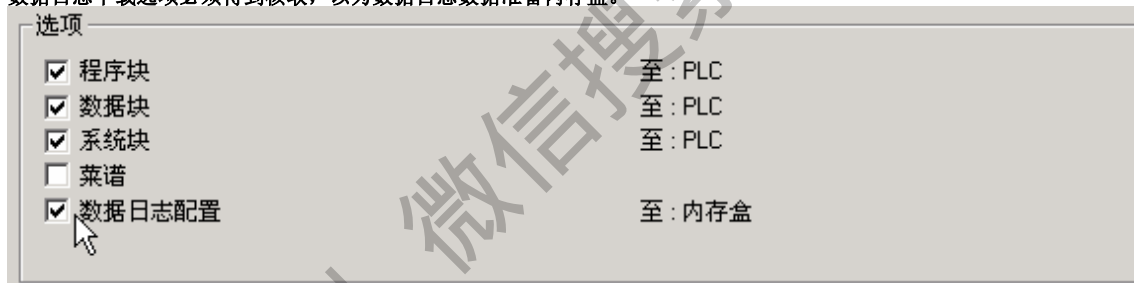


此向导将分配一个“数据块”标记以在V内存中存储数据日志配置。在数据块中指定您希望安放该配置的起始地址。您可以选择V内存地址，如果您希望向导建议正确大小的未使用内存块地址，则单击“建议地址”按钮。配置块的大小根据您在向导中所作的具体选择不同而异。

#### 第5步 生成程序代码



数据日志向导会为您的日志数据定义（符号表、数据块、子例行程序、内存盒数据）生成项目元件，并令您的程序可以使用这些元件。  
 您必须将带有数据日志向导配置的项目下载至S7-200才能加以使用。如果某项目有数据日志配置，则下载窗口的默认设置为“数据日志配置”选项已被核取。  
**数据日志下载选项必须得到核取，以为数据日志数据准备内存盒。**



您的程序能够将新的工艺数据移动到由该记录符号名指明V内存地址，执行一DATx\_WRITE子例行程序以取得全部数据值的快照并在内存盒内保存一个新记录，使用S7-200 Explorer从内存盒中将所积累的记录作为CSV（逗号分隔值）文件上传。

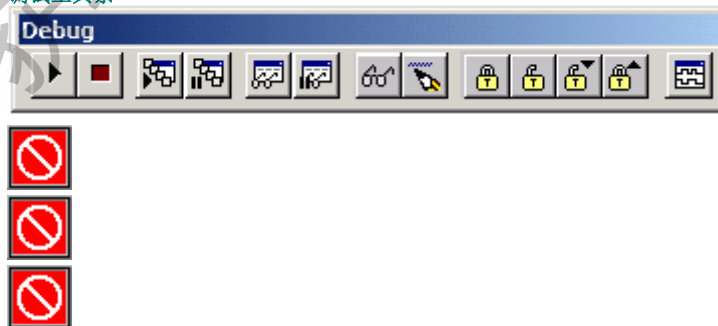
另请参阅：  
[DATx\\_WRITE子例行程序](#)

## 9.9 监控和调试程序

### 9.9.1 调试和监控特征概述

您成功地在运行的STEP 7-Micro/WIN编程设备和PLC之间建立通讯并向PLC下载程序后，则可利用“调试”工具条的诊断特征。可单击工具条按钮或从“调试”菜单列表选择项目，选择调试工具。

调试工具条





本标题讨论下列主题：

[什么是“状态”？](#)

[前提条件](#)

[收集状态的不同方法](#)

[以连续、瞬态图或触发暂停的方式收集状态](#)

[PLC模式](#)

[状态通讯与扫描循环](#)

[模拟程序条件（读取、写入、强制、取消强制）](#)

[检查交叉引用和元素用法](#)

[向程序下载改动](#)

[重设用于调试和编辑的帧设置](#)

## 什么是“状态”？

[返回顶端](#)

‘状态’这一术语指显示程序在 PLC 中执行时有关 PLC 数据当前值和功率流的信息。您可以使用状态图和程序状态窗口读取、写入和强制 PLC 数据值。

在控制程序的执行过程中，PLC 数据的动态改变可用三种不同方式检视：

### 图状态

在一表格中显示状态数据：每行指定一个要监视的 PLC 数据值。您指定一个内存地址、格式、当前值及新值（如果使用写入命令）。

### 趋势显示

用随时间而变的 PLC 数据之绘图跟踪状态数据：

您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势数据亦可在趋势视图中直接赋值。

### 程序状态

在程序编辑器窗口中显示状态数据：当前 PLC 数据值会显示在引用该数据的 STL 语句或 LAD/FBD 图形旁边。LAD 图形也显示功率流。

## 程序状态和图状态 (或趋势视图) 窗口可以同时运行：

在状态图窗口写入或强制 PLC 数据将 PLC 数据改动应用于程序状态窗口。

在程序状态窗口写入或强制 PLC 数据还会将新 PLC 数据改动应用于状态图窗口。

## 检视状态数据的前提条件

[返回顶端](#)

您必须执行下列任务，才能监控状态数据或调试程序：


- 成功地在STEP 7-Micro/WIN和PLC之间建立通讯。
- 成功地向PLC下载程序。
- 欲检视连续状态更新，PLC必须位于RUN（运行）模式。否则，您只能看到对I/O的改动（如果有）。由于PLC程序辉溢葱校 I/O的改动不会对“状态”显示中的程序逻辑产生预期的影响。
- 如果您位于程序不在执行的区域（例如，子例行程序、中断例行程序或由于JMP指令被绕过的区域），不会有状态显示，因为不在对代码执行扫描。

**注释：**

- **STEP 7-Micro/WIN**中项目的时间标记和PLC必须匹配，才能激活状态数据通讯。时间标记比较（在您尝试打开状态时自动执行）繁 STEP 7-Micro/WIN中的项目状态表示正确反映实际PLC程序。如果时间标记不匹配，您可以检查“时间标记不匹配”对话框中南吕 斜志 私悱男 縲斐刹黄 涸
- 下载程序后，您还应将PLC转换回RUN（运行）模式。否则会报告STOP（停止）模式状态数据，但PLC无法执行程序，因此您将无法看到预期的逻辑操作。

**在程序编辑器窗口中收集状态的不同方法**

[返回顶端](#)

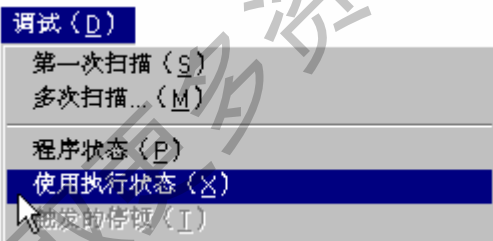
单击“程序状态打开 / 关闭”按钮 ，或选择菜单命令 **调试 (Debug) > 程序状态 (Program Status)**，在程序编辑器窗口中显示PLC数据状态。状态数据收集按以前选择的模式开始。请参阅以下说明。

LAD和FBD程序有两种不同的程序状态数据收集模式。选择 **调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单命令会在状态模式复选标记打开和关闭之间切换。必须在程序状态操作开始之前选择状态模式。

- **执行状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选后）



- **扫描结束状态**（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被取消勾选后）



状态图始终使用“扫描结束”状态数据。以前的STEP 7 Micro/WIN版本用“扫描结束”状态数据建立LAD和FBD程序状态视图。如果您使用的是第一代PLC，执行状态不受支持。如果您尝试执行执行状态，“扫描结束”状态开始。

**执行状态色彩：**

- 程序被扫描时，电源横杆会带有色彩。
- 图形中的功率流用色彩表示。
- 接点？接点打开时，指令会带有色彩。
- 线圈？输出打开时，指令会带有色彩。
- 方框和SUBR指令？指令接通电源并准确无误地成功执行时，方框和SUBR指令带有色彩。
- 绿色计时器和计数器表示计时器和计数器包含有效数据。
- 红色表示执行的指令有误。
- 跳接和标签指令现用时，显示为功率流色彩。如果为非现用，则显示为灰色。
- 灰色（灰色为默认赋值）表示无功率流、指令未扫描（跳过或未调用）或位于STOP（停止）模式的PLC。

- 布尔功率流位（仅限FBD）。

LAD、FBD和STL执行状态视图显示操作数数值，并表示每条指令在位于扫描循环执行程序阶段时的功率流。执行状态可以显示可能被执行后续程序指令盖写的中间数据值。所有显示的 PLC数据值均从一个程序扫描循环收集。

#### 扫描结束状态

- 接通电源或逻辑真的接点和线圈显示为蓝色（您可以选择 **工具（Tools）> 选项（Options）** 菜单项目并单击“色彩”标记，指定自己选择的色彩）。

“扫描结束”状态显示在程序扫描结束时读取的状态结果。这些结果可能不会反映 PLC数据地址的所有数值改动，因为随后某缘蜂鸣铃语酱硫缘蜂 杞嘤 翱瞻芊慈牒楠甌深慈胧 怠 S 捐赃炯 PLC扫描循环和相对慢速的PLC状态数据通讯之间存诘乃保快败稿 吧 杞嘤 弊刺 允炯父玺 枢 方嘤 辈杉 氛 奠怠

#### STL程序中程序状态举例：

当您开启STL中的状态时，程序编辑器窗口被分为一个代码区（左侧）和一个状态区（右侧）。可以根据您希望监控的数值类型

在STL状态中共有三个可监控的数值类别：

- **操作数** 每条指令最多可监控三个操作数。
- **逻辑堆栈** 最多可监控四个来自逻辑堆栈的最新数值。
- **指令状态位** 最多可监控十二个状态位。

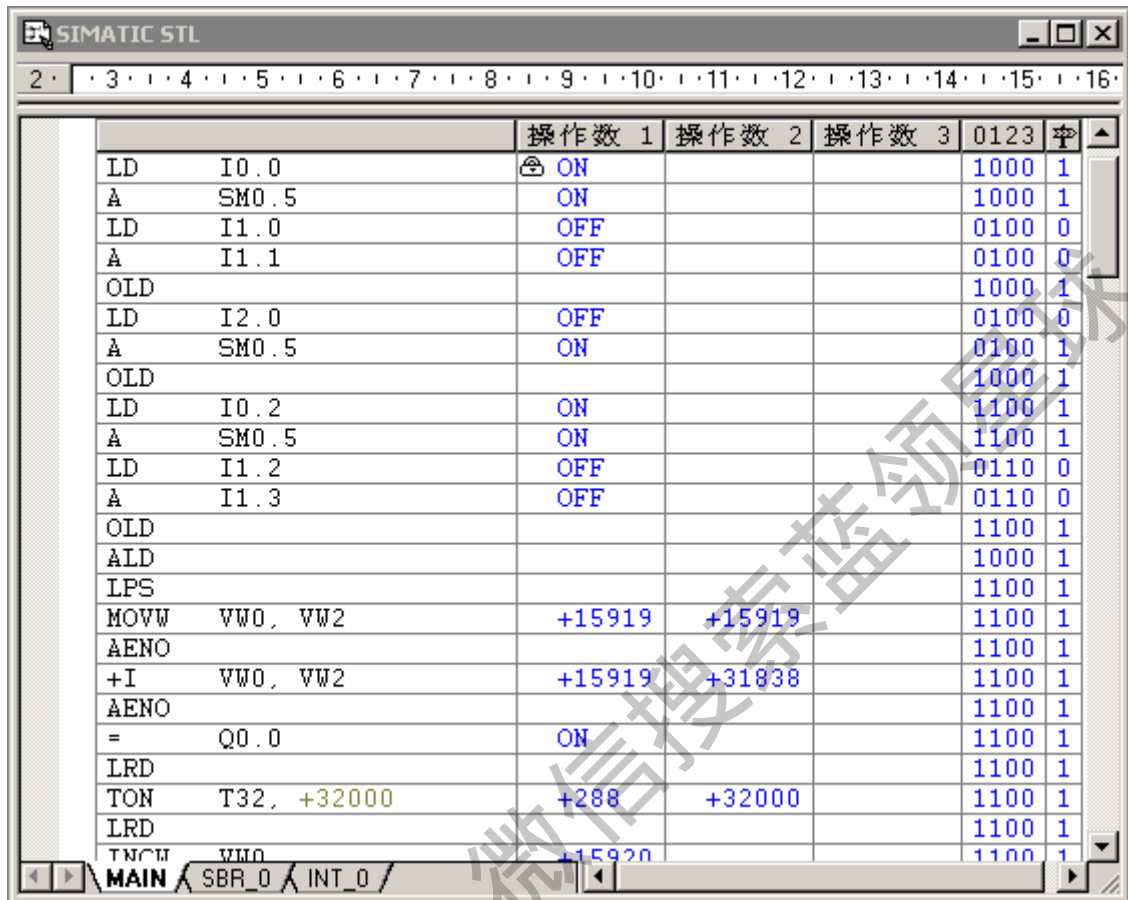
**工具（Tools）>**

**选项（Options）** 对话框的STL状态标记允许您选择或取消选择任何此类数值类别。如果您选择一个项目，该项目不会在“状态”显示中出现。

	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+16095	
5	T32	位	2#0	
6	T32	带符号	+0	

Navigation bar: < > CHT1 < >



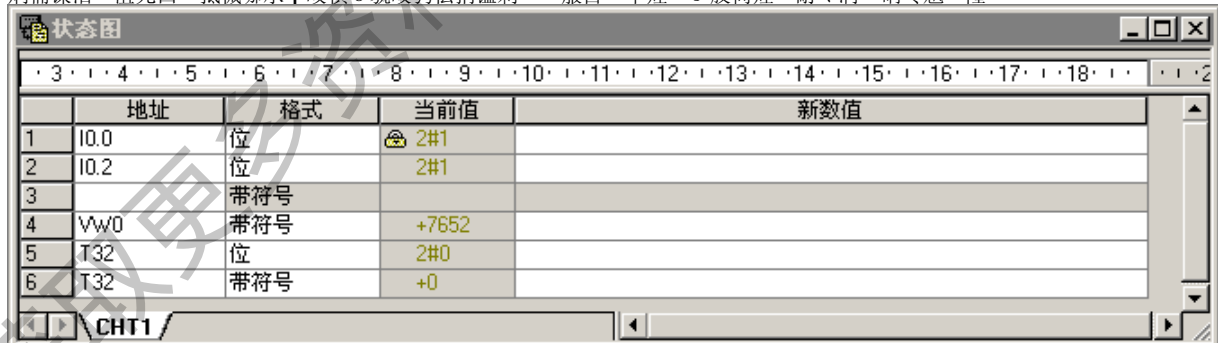


		操作数 1	操作数 2	操作数 3	0123	空
LD	I0.0	ON			1000	1
A	SM0.5	ON			1000	1
LD	I1.0	OFF			0100	0
A	I1.1	OFF			0100	0
OLD					1000	1
LD	I2.0	OFF			0100	0
A	SM0.5	ON			0100	1
OLD					1000	1
LD	I0.2	ON			1100	1
A	SM0.5	ON			1100	1
LD	I1.2	OFF			0110	0
A	I1.3	OFF			0110	0
OLD					1100	1
ALD					1000	1
LPS					1100	1
MOVW	VW0, VW2	+15919	+15919		1100	1
AENO					1100	1
+I	VW0, VW2	+15919	+31838		1100	1
AENO					1100	1
=	Q0.0	ON			1100	1
LRD					1100	1
TON	T32, +32000	+288	+32000		1100	1
LRD					1100	1
TNCW	VW0	+15920			1100	1

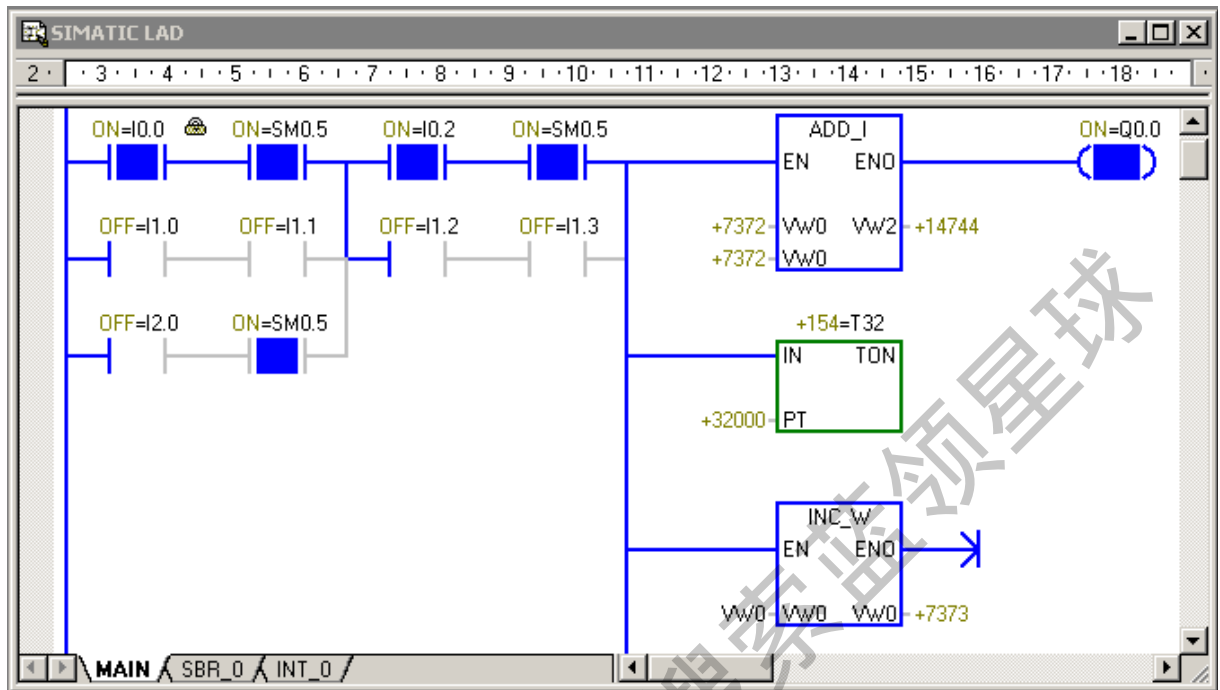
**LAD图形中执行状态举例：**

以下范例在LAD图形中使用LAD执行状态。状态图仅使用“扫描结束状态”数据收集。可在LAD图形和对应的状态图之间看出

炯俑谋道 范允白 抵械哪承 收稂8 貌改鸱怯捎盗刺 服占 牟焯 J胶筒焯 耐久洞 铜咚魑 隆



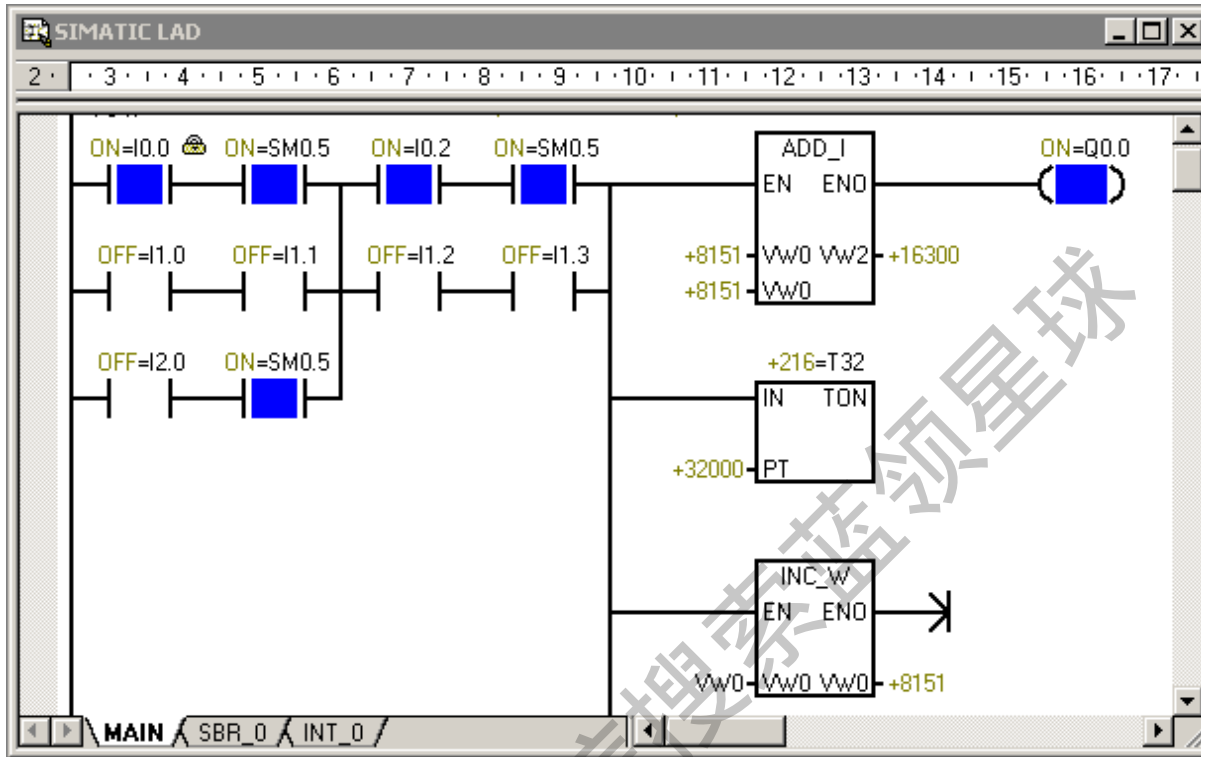
	地址	格式	当前值	新数值
1	I0.0	位	2#1	
2	I0.2	位	2#1	
3		带符号		
4	VW0	带符号	+7652	
5	T32	位	2#0	
6	T32	带符号	+0	



#### LAD图形中扫描结束状态举例:

以下范例在LAD图形中使用“扫描结束状态”。由于两个窗口（程序状态和图状态）均使用相同的扫描 PLC结束数据，两个窗口允鞠嗤 PLC数据值。这种情形会引起误解，因为程序可以在收集最终“扫描结束”数值之前为相同的位置指定很多数值。中间临时数值不显示。在该范例中，ADD\_I方框未能正确执行，因为输入相加超出了带符号字整数的最大范围。与红色方框的执行状态范例不同，您必须读取状态值，才能看到Vw0 + Vw0 <> VW2。

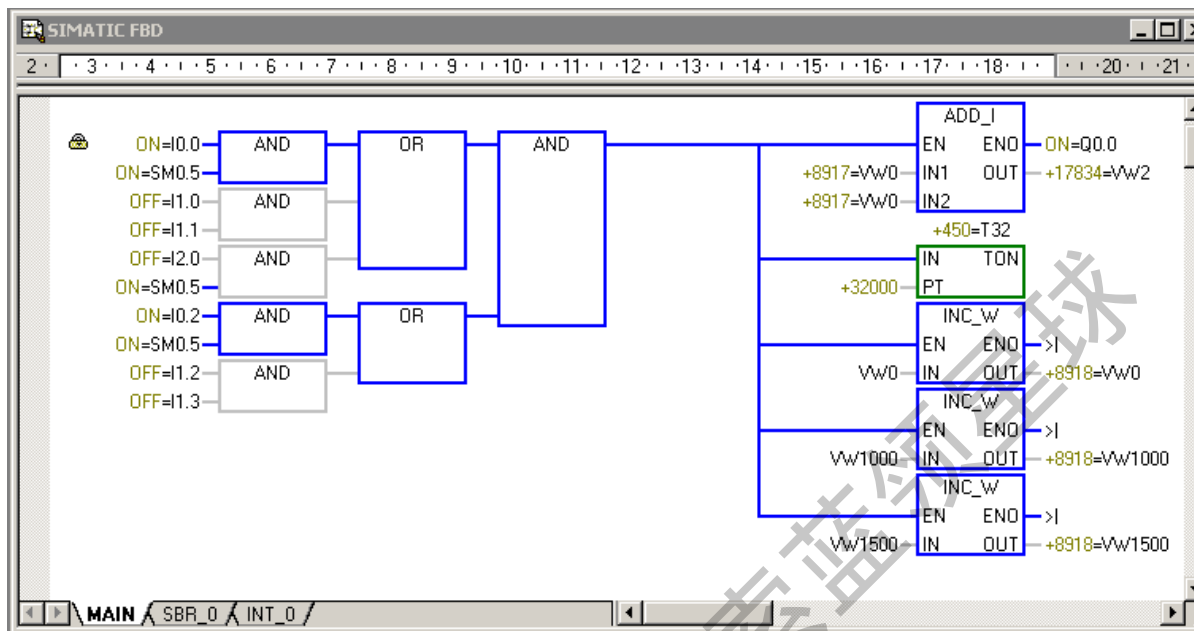
地址	格式	当前值	新数值
1	I0.0	位	2#1
2	I0.2	位	2#1
3		带符号	
4	VW0	带符号	+8418
5	T32	位	2#0
6	T32	带符号	+485



FBD图形中的执行状态举例:

地址	格式	当前值	新数值
1 I0.0	位	2#1	
2 I0.2	位	2#1	
3	带符号		
4 VW0	带符号	+8958	
5 T32	位	2#0	
6 T32	带符号	+0	

获取更多资料



按照连续、瞬态图或触发（暂停程序状态）的方式收集状态值

[返回顶端](#)

**注释：**

“连续”并非意味着实时；而是指编程设备不断地从 PLC 轮询状态信息，在屏幕上显示，并按照通讯允许的速度更新显示。可能

**连续**

- 打开程序编辑器窗口，并启动“程序状态”，当 PLC 位于 RUN（运行）模式时检视程序执行的连续状态更新。
- 打开状态图窗口，并启用“图状态”，当 PLC 位于 RUN（运行）模式时检视连续更新。


**瞬态图**


- 将 PLC 转换为 STOP（停止）模式，收集单个状态更新。当 PLC 位于 STOP（停止）模式时，您可以使用“多次扫描”功能检视一次或多次扫描。您还可以引用网络中的特殊内存 SM0.1 首次扫描位，读取该网络的首次扫描状态值。
- 关闭图状态，如果您希望收集单次状态图更新并不希望将 PLC 转换至 STOP（停止）模式，使用“单次读取”功能。如 PLC 转换为 STOP（停止）模式，并开启“图状态”，您也会收集单次状态更新。此外，您可以在检视状态图时使用“多次扫描”和“首次扫描”功能。

**执行状态模式（LAD / FBD / STL）中的程序状态触发暂停**

欲检视触发暂停 PLC 数据状态，您必须：

1. 将程序下载至 PLC，并将 PLC 设为 RUN（运行）模式。您还可以从 STOP（停止）模式开始触发暂停。如果您从 STOP（停止）模式开始，则可以捕获首次扫描。
2. 滚动程序编辑器窗口，显示您希望检视的程序网络。网络可以位于任何 POU（主程序、子例行程序或中断程序）中。
3. 使用 **调试（Debug） > 使用执行状态（Use Execution Status）** 菜单命令，将“使用执行状态”复选标记切换为打开。
4. 激活程序状态，开始状态数据通讯。

5. 单击“暂停程序状态” 工具条按钮，或用鼠标右键在程序编辑器中单击，并选择弹出菜单命令“暂停程序状态”。
6. 更新暂停之后，触发网络中的状态值在您检视时保持不变。当您准备移至另一个网络时，则单击“暂停程序状态”



 工具条按钮，将暂停切换为关闭，并再次开始连续更新。

一旦获得新的状态信息，只要程序状态暂停功能被启用，该信息就会保持在屏幕上。在您取消选择程序状态暂停功能之前，无论 PLC 中实际变化速度多快，均不会进一步执行状态更新。如果没有程序状态暂停功能，屏幕就会不断刷新，状态数据在您读取和解释之前就可能消失（或者数据根本没有显示）。

**PLC RUN / STOP（运行 / 停止）模式**

[返回顶端](#)

使用以下一种方法改变PLC操作模式：

- 单击“运行”按钮，执行RUN（运行）模式，或者单击“停止”按钮，执行STOP（停止）模式。
- 选择**PLC > 运行 (RUN)** 菜单命令，执行RUN（运行）模式，或选择**PLC > 停止 (STOP)** 菜单命令，执行STOP（停止）模式。
- 以手动方式改变位于PLC上的模式开关。
- 在程序中插入一条STOP（停止）指令。

#### 注释：

欲使用STEP 7-Micro/WIN软件控制RUN / STOP（运行 / 停止）模式，必须在STEP 7-Micro/WIN和PLC之间存在一条通讯路径。此外，必须将PLC硬件模式开关设为TERM（终端）或RUN（运行）。将模式开关设为TERM（终端）不会改变PLC操作模式，但却允许STEP 7-Micro/WIN改变PLC操作模式。位于PLC前方的状态LED表示当前操作模式。当程序状态或状态图操作正在进行时，在STEP 7-Micro/WIN窗口右下方处附近的状态条上会出现一个RUN / STOP（运行 / 停止）指示灯。

虽然程序在STOP（停止）模式中不执行，PLC操作系统继续监控PLC（收集PLC RAM和I/O状态），将状态数据传递给STEP 7-Micro/WIN，并执行所有的“强制”或“取消强制”命令。当PLC位于STOP（停止）模式中时，您可以执行以下操作：

- 使用图状态或程序状态检视操作数的当前值。（此一步骤具有与执行“单次读取”相同的效果，因为程序未执行。）
- 您可以使用图状态或程序状态强制数值。使用图状态写入数值。
- 写入或强制输出。
- 执行有限次数扫描，并通过状态图和 / 或项目状态检视效果。

当PLC位于RUN（运行）模式时，您不得使用“首次扫描”或“多次扫描”功能。您可以在状态图中写入和强制数值，或使用LAD或FBD程序编辑器强制数值，方法与您在STOP（停止）模式中强制数值相同。您还可以执行以下操作（不得从STOP（停止）模式）：

- 使用图状态收集不断变化的PLC数据值的连续更新信息。（如果您希望使用单次更新，图状态必须关闭，才能使用“单次读取”命令。）
- 使用程序状态收集不断变化的PLC数据值的连续更新信息。
- 使用RUN（运行）模式功能中的“程序编辑”编辑程序，并将改动下载至PLC。

#### 状态通讯与扫描循环

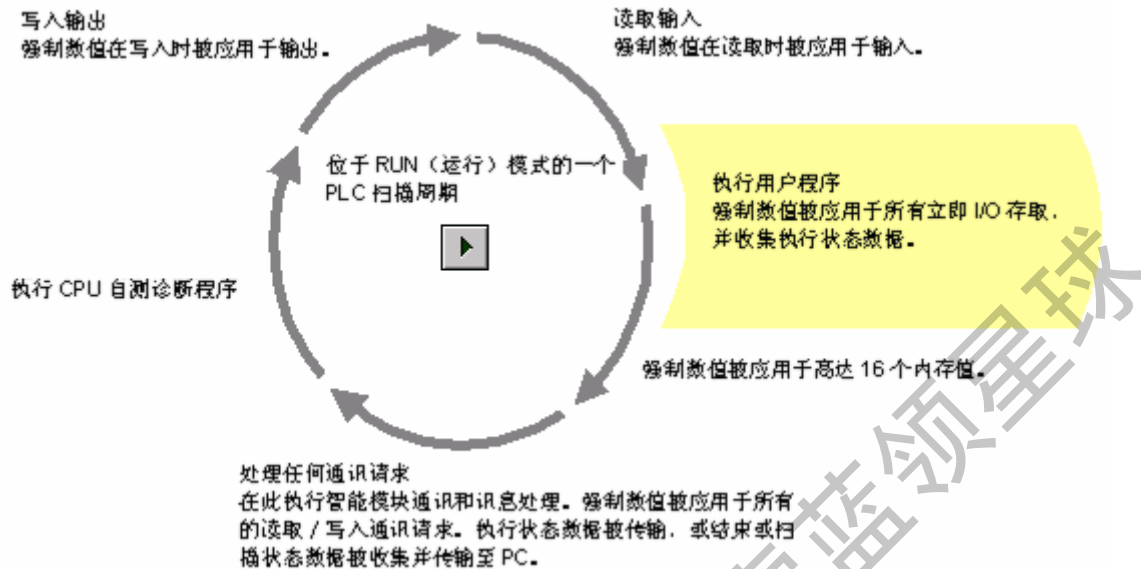
[返回顶端](#)

PLC在连续循环中读取输入、执行程序逻辑、写入输出和执行系统操作和通讯。该扫描循环速度极快，每秒执行多次。虽然STEP 7-Micro/WIN会快速发出状态请求，您应当认识到您检视的不是PLC中出现的每一个事件，这一点十分重要。

由于PLC和您观察的程序状态编程设备之间存在通讯时间滞后，您看到的显示的操作数数值总是在状态显示中改变之前即在PLC中改变，但没有在程序状态表示中改变。

如果您使用“扫描结束”状态模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被取消勾选时），您在几个扫描循环中采集数据。

如果您使用“执行状态”模式检视程序状态（当**调试 (Debug) > 使用执行状态 (Use Execution Status)** 菜单项目被勾选时），所有显示的程序状态值一定来自同一个扫描循环。



**注释:**

一次只有一个状态窗口被授权运行状态。如果您尝试在任何其他与同一个 PLC 连接的 STEP 7 Micro/WIN 应用程序中运行状态，则会收到错误信息。

**模拟进程条件（读取、写入、强制、取消强制）**


[返回顶端](#)

您可以模拟进程条件，方法是在程序状态操作过程中从程序编辑器和从状态图向操作数写入或强制新数值。

使用“程序状态”和“图状态”按钮或“调试”菜单命令，开始状态数据通讯，并启用调试工具。您还可以使用硬件输入模拟器开关（西门子（Siemens）可提供）以手动方式在数字输入打开和关闭之间切换，并观察 PLC 上的 LED 指示灯，观察数字输出的条件。

从“调试”工具条或“调试”菜单列表存取以下功能。






**单次读取**（仅限状态图）如果您希望获得一次“瞬态图”（对状态图中的所有数值一次更新），使用“单次读取”。默认值为状态图连续轮询 PLC，获取状态更新信息。当您单击“状态图”时，状态图会切换为关闭，“单次读取”按钮变为黄色。




**全部写入**（仅限状态图）

在您完成对几行的“新数值”列的改动后，您可以使用“全部写入”，将所有需要的改动发送至 PLC。



**强制** 您可以在操作程序状态时从程序编辑器和状态图强制地址。欲强制程序状态地址，用鼠标右键单击一个参数，并选择从菜单列表强制。欲强制状态图地址为某一数值，您必须首先规定所需的数值，方法是读取该数值（如果您希望强制当前值）或键入该数值（如果您希望将地址强制为一个新数值）。您一旦使用了强制功能，则在每次扫描时该数值均被重新应用于地址，直至您取消强制地址。“强制”功能盖写立即读取或立即写入指令。“强制”功能还盖写被配置为在转换为 STOP（停止）模式时进入一个指定数值的输出：如果 CPU 进入 STOP（停止）模式，输出被设为强制数值，而非非配置的数值。通过将 V 或 M 内存强制为字节、字或双字，将 AI 或 AQ 内存在偶数字节边界上（例如 AIW6 或 AIW14）或 I/O 点上（I 和 Q 位址）强制为字，模拟逻辑条件。您可以一次强制 16 个（V、M、AI 或 AQ）地址和所有的 I/O 位（所有 I 和 Q 位址）。 该图标表示该地址被明确强制。该地址数值在地址被取消强制之前无法改变。 该图标表示该地址被隐含强制。如果地址是一个被明确强制的较大地址的一部分，该地址则被认为是隐含强制。例如，如果 VW0 被强制，则 VB0 是隐含强制（VB0 是 VW0 的第一个字节）。隐含强制数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制 VD0（该地址包含 VB0、VB1、VB2 和 VB3），则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制，则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。



**部分强制** 该图标表示该地址的一部分被部分强制。例如，如果 VW0 被明确强制，则 VW1 的第一个字节是 VW0 的第二个字节。被部分强制的数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制 VD0（该地址包含 VB0、VB1、VB2 和 VB3），则被计数为您可以强制的 16 个内存数值之一。如果您将 VB0、VB1、VB2 和 VB3 作为分开的实体强制，则计数为您可以强制的 16 个内存数值中的四个数值。所有被强制的数值均存储在 CPU 的永久性 EEPROM 内存中。



**取消强制** 对于程序状态和状态图，选择一个地址，并使用“取消强制”按钮从该特定地址移除

强制功能。您还可以选择（单击）一个参数，然后用鼠标右键单击该参数，查看“强制”和“取消强制”功能的弹出菜单。



#### 全部取消强制

如果您希望从所有的地址中移除强制功能，使用“全部取消强制”按钮。在应用“全部取消强制”之前，您无须选择单个地址。



#### 读取全部强制（仅限状态图）

当您使用“读取全部强制”功能时，状态图的“当前数值”列会为已经明确强制、隐含强制或部分隐含强制的所有地址显示一个图标。

### 检查交叉引用和元素用法

[返回顶端](#)

调试程序时，您可以决定是否增加、删除或编辑参数。

使用“交叉引用”窗口查看程序中当前指定参数的方法，这样可以防止您意外地重复赋值。

此外，如果您希望在RUN（运行）模式中执行程序编辑，并且某些网络使用正向或负向转换（EP和N接点或方框以及STL EU或ED指令），您一定要检查“交叉引用”信息中的EU/ED数目，以防重复指令号码。

### 向程序下载改动

[返回顶端](#)

一旦调试部分代码，如果PLC位于STOP（停止）模式，您可以编辑程序、下载程序、然后再次监控状态，查看改动是否发生作用。

您不必进入STOP（停止）模式就可以在RUN（运行）模式中执行程序编辑并向PLC下载较小的改动。



**警告** 在RUN（运行）模式中向PLC下载改动时，您的改动会立即影响程序操作。不得出错；程序编辑中的错误可能导致人员死亡或严重伤害和/或设备损坏。仅限合格人员在RUN（运行）模式中执行程序编辑。

### 重设用于调试和编辑的帧设置

[返回顶端](#)

欲重设将窗口和工具条的可视性、尺寸和位置设为默认值，选择**检视（View）> 帧（Frame）> 全部重设（Reset All）**菜单命令。

您有两套不同的工作空间窗口的便利，可在调试程序时获得较大的灵活性：窗口可视性和定位。这是比正常编辑空间更大的检视空间。当您进入程序状态时，您已经设置的任何窗口定位保持初始进入的状态；如果您关闭程序状态，窗口保持初始进入的状态。同样，任何在编辑程序时执行的窗口定位均在运行时与程序状态分开维护。

[返回顶端](#)

#### 另请参阅：

[如何在程序编辑器中显示状态（GS 7.2）](#)

[如何在状态图中显示状态（GS 7.3）](#)

[如何执行有限数目扫描（GS 7.4）](#)

[如何下载程序（GS 6.3）](#)

[时间标记不匹配错误](#)（确保编程设备中的项目与PLC中的项目相匹配）

[交叉引用和元素用法](#)（确保程序编辑不引起重复赋值）

[在运行模式中执行程序编辑](#)

[PLC运行/停止模式](#)



[在停止模式中写入和强制输出](#)

[安装和移除S7-200设备](#)

[接地和布线指南](#)

## 9.9.2 停止和运行PLC操作模式

使用下列一种方法改变PLC操作模式：

- 单击“运行”按钮，进入RUN（运行）模式，或单击“停止”按钮，进入STOP（停止）模式。
- 选择PLC（PLC）> 运行（RUN）菜单命令，进入RUN（运行）模式，或选择PLC > 停止（PLC >

- **STOP** 菜单命令，进入STOP（停止）模式。
- 以手动方式更改位于PLC中的模式开关。
- 在程序中插入STOP（停止）指令。

**注释：**

欲使用STEP 7-Micro/WIN软件控制RUN/STOP（运行 / 停止）模式，在STEP 7-Micro/WIN和PLC之间必须存在通讯路径。另外，PLC硬件模式开关必须设为TERM（终端）或RUN（运行）。将模式开关设为TERM（终端）并不改变PLC的操作模式，但允许STEP 7-Micro/WIN更改PLC操作模式。PLC前方的状态LED表示当前操作模式。

当程序状态或状态图操作在进行时，在STEP 7-Micro/WIN窗口右下角附近的状态条上有一个RUN/STOP（运行 / 停止）指示灯。

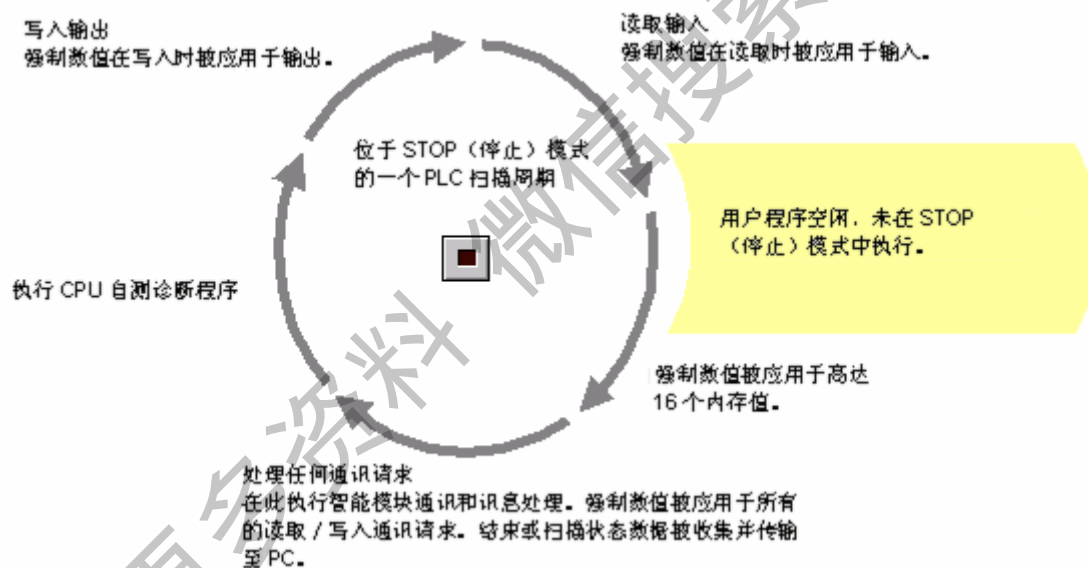
**PLC操作模式详情：**

PLC有两种操作模式：STOP（停止）和RUN（运行）模式。在STOP（停止）模式中，您可以建立 / 编辑程序。在STOP（停止）模式中不得执行程序。但在RUN（运行）模式中可执行程序。此外，在RUN（运行）模式中，您可以建立、编辑和监控程序操作和数据。提供的调试协助可增强追踪程序操作和识别编程问题的能力。

可在STOP（停止）模式中使用调试协助，例如首次扫描和多次扫描功能，并导致预定扫描次数从STOP（停止）更改为RUN（运行）模式。

PLC操作系统存储严重错误，并强制从RUN（运行）模式更改为STOP（停止）模式。如果PLC检测到严重错误，在严重错误条件依然存在时不允许从STOP（停止）模式更改为RUN（运行）模式。PLC操作系统功能存储非严重错误，用于检查，但不会使模式从RUN（运行）更改为STOP（停止）。

在STOP（停止）模式中，PLC处于半空闲状态。用户程序执行被中断；执行输入更新；用户中断条件被禁用。下图描述PLC在STOP（停止）模式中遵守的时间表。



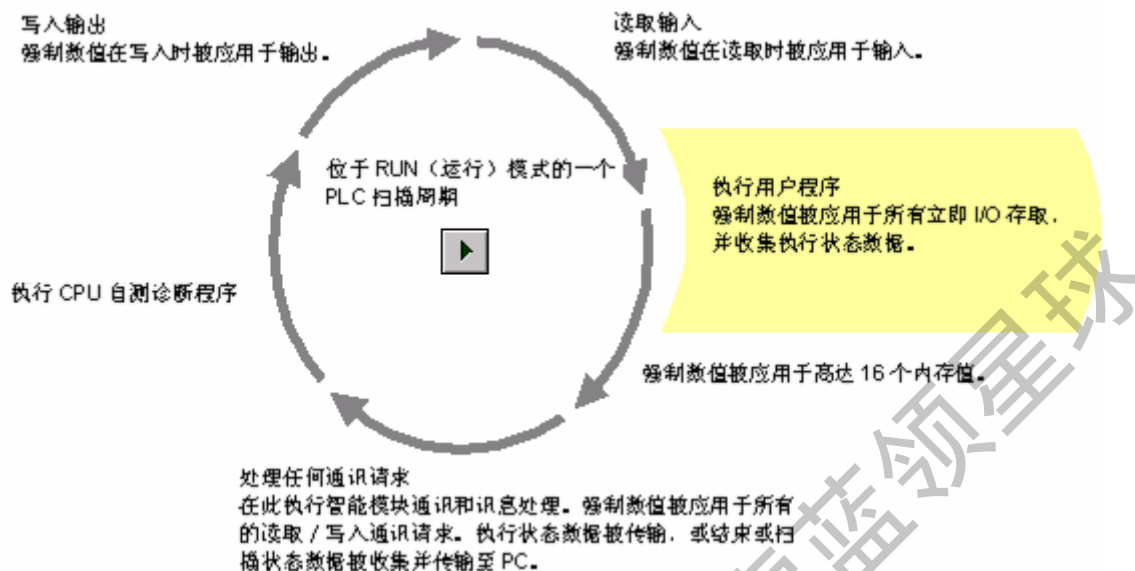
发生通讯中断时，PLC收到讯息并在适当的情况下执行请求。当PLC仍然处于STOP（停止）模式时，图像寄存器作出I/O数值更改。一种例外的情况是对图像寄存器盖写I/O数值改动的强制功能。处于STOP（停止）模式时，您能够载入、上载或删除用户程序

一台或多台设备尝试通过通讯端口与PLC通讯时，PLC按顺序对每一项请求作出应答。PLC不尝试阻止一台通讯设备的行动干扰

自我诊断检查包括定期检查操作系统EEPROM、I/O模块状态检查和在每次存取扩充I/O时执行的I/O扩充总线一致性检查。

在RUN（运行）模式中，PLC读取输入、执行程序、写入输出、对通讯请求作出应答、更新智能模块、执行内部管理工作以及对中断条件作出应答。PLC不支持用于RUN（运行）模式执行循环的固定扫描时间。根据发生的优先顺序对这些行动（中断除外）





每次扫描循环开始时先读取当前输入位数值, 并将这些数值写入输入图象寄存器。输入位无相应的实际输入, 但与具有实际输入位位于同一个字节中, 每次输入更新循环时, 除非输入位被强制, 否则会在图象寄存器中被设为零。

读取输入后, 程序从第一条指令开始执行, 直至遇到结束指令。遇到结束指令时, PLC 检查系统的智能模块是否需要服务。如果秤 寥 断 2.13. 断 4.娟 00 谔 返南乱唤锥位撼遮髦小

在扫描循环的讯息处理阶段, 从通讯端口接收的讯息被处理。完成的应答被置于一旁, 等待在适当的时间传输给通讯请求方。

自我诊断检查包括对操作系统 EEPROM 和用户程序内存以及 I/O 模块状态检查进行定期检查。

最后, 输出图象寄存器数值被写入输出模块, 从而完成一次扫描循环。

另请参阅:

[在停止模式中写入和强制输出](#)

[在运行模式中进行程序编辑](#)

[应用程序用户参考手册](#)

[入门指南目录](#)

### 9.9.3 执行一组固定数目的扫描

您可以指定 PLC 对程序执行有限次数扫描 (从 1 次扫描到 65,535 次扫描)。通过选择 PLC 运行的扫描次数, 您可以在程序改变进程 淋渴倍云浣 屑噉于 第一次扫描时, SM0.1 数值为 1 (打开)。

**执行单次扫描:**

1. PLC 必须位于 STOP (停止) 模式。如果不是已经位于 STOP (停止) 模式, [将 PLC 转换成停止模式](#)。
2. 从菜单条选择 **调试 (Debug) > 首次扫描 (First Scan)**。


**执行多次扫描:**

1. PLC 须位于 STOP (停止) 模式。如果不是已经位于 STOP (停止) 模式, [将 PLC 转换成停止模式](#)。
2. 欲执行多次扫描, 从菜单条选择 **调试 (Debug) > 多次扫描 (Multiple Scans)**。出现“执行扫描”对话框。



3. 输入所需的扫描次数数值, 单击“确定”, 确认您的选择并取消对话框。

**注释:**

当您准备好恢复正常程序操作时，请务必将PLC转回RUN（运行）模式（单击工具条中的“运行”按钮或选择PLC > 运行（RUN））。

#### 另请参阅：

[调试和监控特征概述（GS 7.1）](#)  
[如何在编辑器窗口中显示状态（GS 7.2）](#)  
[如何在状态图中显示状态（GS 7.3）](#)

[如何下载程序（GS 6.3）](#)  
[时间标记不匹配错误（确保编程设备内的项目与PLC内的项目相匹配）](#)  
[交叉引用和元素用法（确保程序编辑不引起重复赋值）](#)  
[在运行模式中进行程序编辑](#)  
[PLC运行 / 停止模式](#)  
[在停止模式中写入和强制输出](#)

## 9.9.5 在运行模式中编辑程序



**警告** 在RUN（运行）模式中向PLC下载改动时，您的改动会立即影响程序操作。因此没有防范错误的余地；编程编辑中的错误可能引起人员死亡和严重伤害及 / 或装置损坏。仅限合格人员在RUN（运行）模式中执行程序编辑。

本标题讨论下列主题：

[概述](#)  
[在运行模式中编辑的前提条件](#)  
[可能发生的问题](#)  
[处理正向或负向转换（EU、ED指令）](#)  
[在运行模式中初启程序编辑](#)  
[下载编辑](#)  
[指定CPU分配（背景时间）](#)  
[在运行模式中编辑程序的固件 / 硬件支持](#)

### 概述

[返回顶端](#)

在RUN（运行）模式中执行程序编辑功能允许您不必转换至STOP（停止）模式即可对程序作出更改并将改动下载至PLC。

- 您不必关机即可对当前程序作出较小的改动。例如：更改参数值。
- 您可以利用此一功能更快地调试程序。例如：对正常打开或正常关闭开关进行反转逻辑操作。

向真实程序（与模拟程序相反，调试程序时可能使用模拟程序）下载改动时，下载之前务必要仔细考虑可能对机器和机器操作员造成的各种可能的安全问题。

### 在运行模式中编辑的前提条件

[返回顶端](#)

只有满足下列前提条件才能在RUN（运行）模式中向PLC下载程序编辑：

- 程序必须成功地编译。
- 您必须已经成功地在运行STEP 7-Micro/WIN的个人计算机与PLC之间建立通讯。
- 目标PLC的固件必须支持在RUN（运行）模式中进行程序编辑的功能。
- 您尝试下载的程序必须符合STEP 7-Micro/WIN 3.0版或更高版本接受的POUs顺序。如果您上载的程序由较低版本的STEP 7-Micro/WIN建立，则程序可能无须遵守此一顺序。如果程序元素顺序有误，STEP 7-Micro/WIN将要求您在下载改动之前将PLC转换为STOP（停止）模式。下载程序时，STEP 7-Micro/WIN会自动修复POU顺序。随后下载程序时，您可以在RUN（运行）模式中使用程序编辑功能。

如果在完成编辑并成功下载之前，PLC被转为STOP（停止）模式，编辑操作会被异常中止。

### 可能发生的问题

[返回顶端](#)

为了帮助您决定是在RUN（运行）模式还是在STOP（停止）模式向PLC下载程序编辑，请考虑下列有关RUN（运行）模式编辑囊蚤兀

- 如果在RUN（运行）模式编辑中删除输出的控制逻辑，输出保持其最后状态，直至下一次电源循环或转换为STOP（停止）模式。
- 如果在RUN（运行）模式编辑中删除正在运行的HSC或PTO/PWM功能，这些功能继续运行，直至下一次电源循环或转换为STOP（停止）模式。
- 如果在RUN（运行）模式编辑中删除ATCH指令但中断例行程序本身尚未删除，则中断例行程序继续执行，直至下一次电源循环或转换为STOP（停止）模式。与此相似，如果DTCH指令被删除，则中断例行程序不会关闭，直至下一次电源坊蜃 晃 STOP（停止）模式。
- 如果在RUN（运行）模式编辑中增加ATCH指令，并以遇到首次扫描旗标为条件，这些事件不会被激活，直到下一次电源循环或出现STOP-to-RUN（停止至运行）模式转换。
- 如果在RUN（运行）模式编辑中删除ENI指令，中断例行程序继续操作，直至下一次电源循环或出现RUN-to-STOP（运行至停止）模式转换。
- 如果在RUN（运行）模式编辑中修改接收方框的表格地址，而且接收方框在从旧程序向新程序转换的过程中被激活，蚪邮盍氛 茶恍慈剔杀森 卩貳V挥性冢厂蚰俊 卩返模 冫鼻放邮涨脓管甬珊澎趴 际褂咪碌 卩罚 蛭 丫 猿缘蚪 辛吮嗦 缘蚪 谛碌 卩分醒罢沂 苾ũ 床淮壁冢 NETR和NETW指令功能与此相似。
- 以首次扫描旗标为条件的逻辑在因为电源循环或从STOP-to-RUN（运行至停止）模式转换而进入RUN（运行）模式后才被执行，因为首次扫描旗标不受RUN（运行）模式编辑的影响

**处理正向或负向转换（EU、ED指令）**[返回顶端](#)

在RUN（运行）模式编辑过程中，您可以对包含正向或负向转换（EU或ED指令）的网络进行改动，STEP 7-Micro/WIN暂时为程序中的每条EU/ED指令分配一个号码。当您在RUN（运行）模式中激活“程序编辑”功能时，“交叉引用贝翱溢械脑 曠梅ũ 梈谐葩鼻爸付ũ乃 EU/ED指令。禁止使用重复的EU/ED指令。改动程序时请参考该列表。


**在运行模式中初启程序编辑**[返回顶端](#)

欲在RUN（运行）模式中初启程序编辑，选择菜单命令 **调试（Debug） > 在运行模式中进行程序编辑（Program Edit in RUN）**。

**注释：**

如果程序中有用STEP 7-Micro/WIN编写的尚未保存的材料，会提示您将程序保存至编程设备的硬盘中。您可以使用与CPU中的项目名称相同的名称，或吒谋涿 疲豪 纾 聘 奈 BACKUP.MWP。

会出现警告对话框。选择“继续”按钮，确认您希望继续在RUN（运行）模式中编辑程序。这会使程序从CPU上载至STEP 7-Micro/WIN，您可以在此作出必要的编辑改动。

当程序编辑器可以接受更改时，鼠标会变为CPU光标 。完成所需的改动后，您必须将改动下载，才会在CPU中生效。是否将程序编辑保存在编程设备的硬盘中是您的个人选择，保存时是否采用与项目较早版本相同的名称也是您的个人选择。采用相同的名称会使较早版本被盖写。

**下载编辑**[返回顶端](#)

在RUN（运行）模式中执行程序编辑的过程中，您只能下载程序块（OB1），而不能下载系统块或数据块。

开始下载后，您不能在STEP 7-Micro/WIN中执行其他任务，直至下载完成。

检查输出窗口是否存在任何编译错误（例如，重复EU或ED号码）。您可以双击错误讯息，在STEP 7-Micro/WIN程序编辑器窗口中调出出错的网络。

**指定CPU分配（背景时间）**[返回顶端](#)

在RUN（运行）模式中开始下载程序时，CPU需要一些时间对修改的程序进行背景编译（CPU同时继续执行当前载入的程序）。您可以使用系统块的“背景时间”标记更改CPU花费在编译修改程序上的时间。默认值是扫描时间的10%；您最多可将其指定为50%。较高的背景时间值可加快正在下载的程序编译速度，但会延长总体扫描时间，这可能会对当前执行程序产生不利的影响。

#### 在运行模式中为程序编辑提供固件 / 硬件支持

[返回顶端](#)

下列CPU型号支持在RUN（运行）模式中进行程序编辑：

- CPU 224 1.10版（或更高版本）
- CPU 226 1.00版（或更高版本）

[返回顶端](#)

另请参阅：

[运行 / 停止模式](#)

[在停止模式中写入和强制输出](#)

[调试和监控功能概述（GS 7.1）](#)

[背景时间](#)

[下载（文件 > 下载）](#)

[如何下载程序（GS 6.3）](#)

[正向和负向转换](#)

## 9.9.6 使用状态图

您可以建立一个或多个状态图，在将程序下载至PLC之后监管和调试程序操作。打开一个图，检视或编辑图的内容。开启图，收集数据。

在控制程序的执行过程中，状态图数据的动态改变可用两种不同方式检视：

#### 图状态

在一表格中显示状态数据：每行指定一个要监视的PLC数据值。您指定一个内存地址、格式、当前值及新值（如果使用写入命令）。

#### 趋势显示

用随时间而变的PLC数据之绘图跟踪状态数据。

您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势数据亦可在趋势视图中直接赋值。

使用以下一种方法打开状态图：

- 单击**浏览条**上的“状态图”按钮。
- 选择**检视（View） > 元件（Component） > 状态图（Status Chart）**菜单命令。
- 打开**指令树**中的“状态图”文件夹，然后双击“图”图标。
- 如果您在项目中有一个以上状态图，使用位于“状态图”窗口底部的“图”标记


 在图之间移动。

如果您已经打开一个空状态图，编辑图行，增加您希望监管的PLC数据地址。

使用以下一种方法启动在状态图中载入PLC数据的通讯：

- 欲连续收集状态图信息，开启状态图：使用菜单命令 **调试（Debug） > 图状态（Chart**

**Status）**或使用“图状态”工具条按钮。

- 欲获得单个数值“瞬态图”，使用“单次读取”功能：使用菜单命令 **调试（Debug） > 单次读取（Single Read）**或使用“单次读取”工具条按钮。（但是，如果您已经开启图状态，“单次读取”功能被禁用。）



**提示：** • 打开图并不意味着检视状态。您必须开启图，才能收集状态信息。

- 如果图空置，开启图没有任何意义：您必须首先将程序数值（操作数）放在“地址”列中，并为“格式”列中的每一个数据选择数据类型，“建立”图。
- 您可以用于STOP（停止）模式的PLC收集状态，检查初始或最终条件。PLC必须位于RUN（运行）模式，才能从连续执行的程序收集数据。

如果PLC位于RUN（运行）模式，程序在连续扫描的状况下执行。您可以开启图状态，连续更新状态图数值。作为替代开启图状  
牧俄恢址椒ā 稍允褂漫暗丕味寥 愈δ埽 占 刺 际 档牡彡航八蔡 肌薄  
您在检视状态图时，还可以将PLC设为STOP（停止）模式，使用“首次扫描”或“多次扫描”功能，监管有限数目扫描的程序操作。



**图状态提示：** • 无论PLC位于何种操作模式，锁定始终会显示。

- 如果PLC位于RUN（运行）或STOP（停止）模式，可使用写入、强制和取消强制功能。但是，在STOP（停止）模式中，写入输出（Q和AQ）选项只有在**调试（Debug）> 在停止中写入-强制输出（Write-Force Outputs in Stop）**菜单项目被勾选时才能使用。
- 当PLC位于RUN（运行）或STOP（停止）模式时，状态数值被显示。
- 强制和取消强制工具条按钮选项仅在用于带有V、M、AI和AQ内存类型的字节、字和双字长度时才被启用。
- 强制和取消强制工具条按钮和弹出菜单选项在用于带有I和Q内存类型的位、字节、字和双字长度时被启用。工具条按钮只有在所有选项可以被适当强制/取消强制（多行选项时如此）时才会变成可用。
- 强制和取消强制工具条按钮和弹出菜单选项在用于所有完全或部分隐含强制数值时被禁用。
- 明确强制数值可以被强制为一个新数值，而无需取消强制后再强制新数值。

本标题讨论下列主题：

[状态图举例](#)

[建立图](#)

[使用多个状态图](#)

[数据格式](#)

[单次读取与连续图状态](#)

[写入与强制数值](#)

[在状态图中使用调试功能](#)

[编辑捷径](#)

[趋势显示详细说明](#)

### 状态图举例

[返回顶端](#)

打开状态图窗口与启动状态图不同。您可以打开状态图进行编辑或检查，但除非您（从“调试”菜单或工具条）发出“单次读取”命令或（从“调试”菜单或工具条）启动状态图，否则不会在“当前数值”列中显示状态信息。

利用“单次读取”特征（只有在关闭图状态时才能使用）检查状态图时，从PLC收集当前数值，并在“当前数值”列显示，但PLC执行程序时并不对其进行更新。

（从“调试”菜单或工具条）启动图状态时，在连续循环的基础上从PLC收集当前数值。从PLC接收到改变时，对“当前数值”新解隆

您可以利用“新数值”列指定（写入或强制）您确定的一个数值。如果“新数值”欲空置，并且在“当前数值”域中存在一个数值，则当前数值被强制。

	地址	格式	当前数值	新数值
1	起始_1	位	2#1	
2	起始_2	位	2#0	
3	停止_1	位	2#1	
4	停止_2	位	2#0	
5	高位	位	2#0	
6	低位	位	2#0	
7	重置	位	2#0	
8		带符号		
9	泵_1	位	2#1	
10	泵_2	位	2#0	
11	混合器_泵	位	2#0	
12	蒸汽_阀	位	2#0	
13	排放_阀	位	2#1	
14	排放_泵	位	2#1	
15		带符号		
16	高位_已达到	位	2#1	
17	混合_定时器	带符号	+7426	
18	循环_计数器	带符号	+0	

## 建立图

[\(返回顶端\)](#)

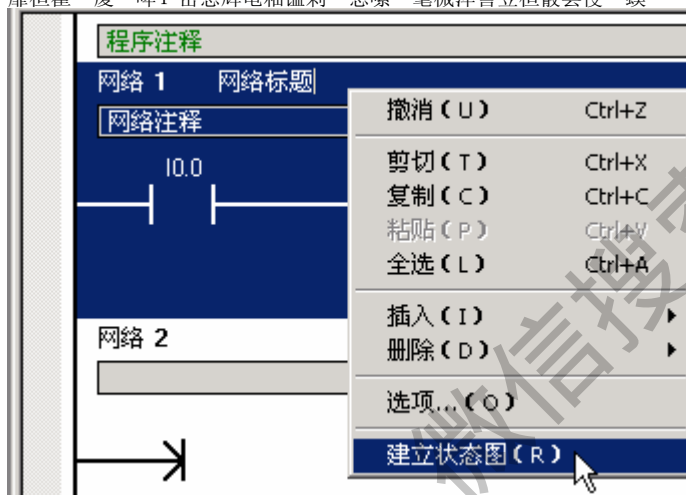
在状态图中，您可以输入地址或定义符号名，从程序监管或修改数值。计时器或计数器数值可以显示为位或字。如果您将计时器或计数器数值显示为位，则会显示输出状态（输出打开或关闭）。如果您将计时器或计数器显示为字，则使用当前数值。欲建立状态图，遵循以下步骤。

- 在“地址”域中为每个需要的数值输入地址（或符号名）。  
PLC内存地址范围中列出的大多数内存类型有效，数据常量、累加器和高速计数器除外。  
欲编辑地址单元格，使用箭头键或鼠标选择您希望编辑的单元格。  
如果您开始键入，域被清除，输入新字符。  
如果您单击鼠标或按“F2”键，域被增亮，您可以使用箭头键将编辑光标移至您希望编辑的位置。

地址	
1	I0.0

或

增亮程序代码的一部分，用鼠标右键调出弹出菜单，并选择命令 **建立状态图 (Create Status Chart)**。新图包含可以收集状态的选择区域中每个独特操作数的一个条目。条目在程序中按照其出现的顺序排列，图获  
靡桓瞿 厦 啤 P 峦急辉龟釉谥刺 急唛 髦械淖罟笠桓藪昙侵 蟆



每次您选择 **建立状态图 (Create Status Chart)** 时，只能增加头150个地址。

**注释：**

一个项目最多可存储32个状态图。

- 如果元素是位（例如，I、Q或M），格式在“格式”列中被设为位。如果元素是字节、字或双字，选择“格式”列中的单元格，并双击或按SPACEBAR或ENTER键，在有效格式中循环，直至显示适当的格式。

格式
位

欲插入附加行，使用“编辑”菜单或用鼠标右键单击状态图中的一个单元格，调出弹出菜单，并选择菜单命令 **插入 (Insert) > 行 (Row)**。新行被插入在状态图中光标当前位置的上方。您还可以将光标放在最后一行的任何一个单元格中，并按下箭头键，在状态图底部插入一行。



- 提示：**
  - 您可以从符号表选择地址，并将地址复制至状态图，更快地建立图。
  - 您可以建立多个状态图。这样可允许您将元素分为逻辑群组，以便每个群组可以在一个不同的图中检视。则可不必在一个包括全部元素的长列表中滚动。

## 使用多个状态图

[\(返回顶端\)](#)

欲建立一个新状态图，核实“图状态”已经关闭，然后：

- 从指令树，用鼠标右键单击“状态图”文件夹，并选择弹出菜单命令 **插入 (Insert) > 图 (Chart)**。
- 打开状态图窗口，并使用“编辑”菜单或用鼠标右键单击，调出弹出菜单，选择 **插入 (Insert) > 图 (Chart)**。

**注释：**

- 成功地插入新状态图后，在状态图窗口底部会出现一个新标记 ，并在指令树  
翔溢小白刺 肌蓖急哈路搅碗饨 04桓鲈 H梁 刺 贾甌旅 檬孟暖壹 七髦喝钍髦械亩杂 v 急转 03=癭  
爸甌旅 保 蛋 鳌小白刺 肌北县敲 啤 S 谕贾 涅贫 恍璧 七 魁 壁耐急昙恰
- 有时位于标记部分右侧的滚动按钮可能会遮住标记。如果看不到标记，拖曳标记区域和滚动按钮之间的分界线，以便

看到更多的标记。

### 数据格式

[\(返回顶端\)](#)

您为数值指定的数据格式决定如何在状态图中显示该数值。  
在下例中，V0.0（以及V0和V0.0）包含数值1。

	地址	格式	当前数值
1	V0.0	位	2#1
2	V0	二进制	2#0000_0001
3	V0	不带符号	1
4	V0	带符号	+1
5	V0	十六进制	16#01
6	V0	ASCII	'\$01'
7	V0	字符串	'\$01\$00\$00\$01\$00\$00\$00\$00\$00\$0
8	V0	浮点数	2.350989E-038





**提示：** • 位和二进制数值前面都带有数字2和井号（#）。十六进制数值前面带有数字16和井号（#）。 • 带符号和不带符号的数值均使用十进制数。

写入与读取相反。当您读取字符串格式的VB地址时，第一个字节首先读取，这代表字符串的长度。然后再读取其他字符串。鼻爸邓孀址 旧场 隆日继 粤俄恢指拾较允 VB地址，则会看到长度字节数值。字符串长度被写入第一个字节，其后是字符串本身。

### 单次读取与连续图状态

[\(返回顶端\)](#)

- 欲获得数值的单次“瞬态图”，使用“单次读取”功能：使用菜单命令 **调试 (Debug) > 单次读取 (Single Read)** 或使用“单次读取”  工具条按钮。（但是如果您已经启动图状态，“单次读取”功能则被禁用。）
- 欲连续收集状态图信息，启动图状态：使用菜单命令 **调试 (Debug) > 图状态 (Chart Status)** 或使用“图状态”  工具条按钮。

### 写入与强制数值

[\(返回顶端\)](#)

“全部写入”功能允许您向程序写入一个或多个数值，模拟一种条件或一系列条件。然后您可以运行程序或使用状态图（以及程序状态（如果需要））监控运行状况。执行程序时，您用“全部写入”功能修改的数值可能被新数值盖写。

“强制”功能允许您模拟逻辑条件，方法是强制V（字节、字或双字）、M（字节、字或双字）、AI（字）或AQ（字）或实际条件（通过强制配备I和Q地址的I/O点）。当您在状态图中使用“强制”功能强制一个或多个程序数值时，每次扫描循环时都会有效地重设数值。虽然在单次扫描过程中，程序可修改强制数值，但新扫描开始时，会重新应用强制数值。由于PLC执行速度很快，PLC向您的编程设备传输状态信息和在编程设备屏幕上显示信息所需的延误时间很长，您可能无法观察到强制数值的任何改变。



**提示：** • 如果您将一个数值写入I/O点，您可能永远不会看到该数值在状态图中作为当前值显示，因为在下一次扫描时程序可能写入一个不同的数值（与您在状态图中写入的数值不同）。使用“强制”功能将数值指定给I/O更加有效。

- 图状态允许您在“新数值”列键入一个所需的数值。请记住，当PLC收到强制命令时，PLC会将操作数强制为当前值。这是一个快速的程序，但不是瞬时程序。操作数数值有可能在您单击STEP 7-Micro/WIN中的状态显示时间和强制命令被PLC收到时间之间发生变化。一旦锁定图标在操作数旁显示，您可以看到在PLC中被强制的数值。如果数值未被正确强制，您可以使用状态图在“新数值”列中键入您希望使用的确切数值，并从图状态强制操作数。




**警告** 在程序中强制数值时，在程序每次扫描时将操作数重设为该数值，与输入 / 输出条件或其他正常情况下对操作数值有影响的程序逻辑无关。强制可能导致程序操作无法预料，无法预料的程序操作可能导致人员死亡或严重伤害和 / 或设备损坏。强制功能是调试程序的辅助工具，切勿为了弥补处理设备的故障而执行强制。仅限合格人员使用强制功能。强制程序数值后，务必通知所有授权维修或调试程序的人员。

## 在状态图中使用调试功能

[\(返回顶端\)](#)

您可以在程序状态操作过程中从程序编辑器和状态图将新数值写入或强制至操作数，用此种方法模拟进程条件。

使用“图状态”按钮或“调试”菜单命令，开始状态数据通讯。

您可以从“调试”菜单或“调试”工具条存取以下“调试”功能。






**单次读取**（仅限状态图）欲获得“瞬态图”，即对全部数值程序状态的一次更新，使用“单次读取”功能。在默认情况下，图状态从PLC连续获取状态更新信息。单击状态图且图状态关闭时，则启用“单次读取”按钮。



**全部写入**（仅限状态图）对状态图中的“新数值”列的改动完成后，您可以使用“全部写入”，将所需的改动传送到PLC。



**强制** 欲将地址强制为某一数值，您必须首先规定所需的数值，可通过读取数值（如果您希望强制当前数值）或键入数值（如果您希望将地址强制为新数值）来完成。一旦您单击“强制”按钮，每次扫描都会将数值重新应用于该地址，直至您对该地址执行取消强制。“强制”功能盖写立即读取或立即写入指令。“强制”功能还盖写被配置为在转换为STOP（停止）模式时进入一个指定数值的输出：如果CPU进入STOP（停止）模式，输出被设为强制数值，而非非配置的数值。通过将V或M内存强制为字节、字或双字，将AI或AQ内存在偶数字节边界上（例如AIW6或AIW14）或I/O点上（I和Q位址）强制为字，模拟逻辑条件。您可以一次强制16个（V、M、AI或AQ）地址和所有的I/O位（所有AI和Q位地址）。该图标表示该地址被明确强制。该图单元格数值在地址被取消强制之前无法改变。该图标表示该地址被隐含强制。如果地址是一个被明确强制的较大地址的一部分，该地址则被认为是隐含强制。例如，如果VW0被强制，则VB0是隐含强制（VB0是VW0的第一个字节）。隐含强制数值无法自身取消强制。您必须取消强制较大的地址，然后才能改变该地址数值。如果您强制VD0（该地址包含VB0、VB1、VB2和VB3），则被计数为您可以强制的16个内存数值之一。如果您将VB0、VB1、VB2和VB3作为分开的实体强制，则计数为您可以强制的16个内存数值中的四个数值。所有被强制的数值均存储在CPU的永久性EEPROM内存中。该图标表示该地址的一部分被部分强制。例如，如果VW0被明确强制，则VW1的一部分被强制（VW1的第一个字节是VW0的第二个字节）。被部分强制的数值无法自身取消强制。您必须取消强制该地址的较大地址。如果您强制VW1，则VW0的一部分被强制（VW0的第一个字节是VW1的第二个字节）。被部分强制的数值无法自身取消强制。您必须取消强制该地址的较大地址。



**取消强制** 对于程序状态和状态图，选择一个地址并使用“取消强制”从该特定地址移除强制。您也可以选择（单击）一个参数，然后用鼠标右键单击该参数，调出“强制”和“取消强制”功能弹出菜单。



**全部取消强制**（仅限状态图）如果您希望从全部地址移除强制，使用“全部取消强制”功能。应用“全部取消强制”之前不必选择单个地址。



**读取全部强制**（仅限状态图）使用“读取全部强制”特征时，状态图的“当前数值”列为已经被明确强制、隐含强制或部分隐含强制的所有地址显示一个图标。

## 图编辑捷径

[\(返回顶端\)](#)

- 欲插入或附加一个具有下一个顺序地址和相同数据格式的新行，选择一个“地址”单元格，并按 **ENTER** 键。
- 欲在特定地址的所有可能数据格式中循环，选择“数据格式”单元格，并连续按 **ENTER** 键。欲检视所有可供使用的数据格式，按 **ENTER** 键。
- 欲移至图的下一个单元格，按 **TAB** 键。
- 欲调整列宽，将鼠标指针放在列边缘，直至光标变成“调整大小”光标，然后将列边缘拉至所需的位置。
- 欲选择一整行（用于剪切或复制），单击行号。
- 欲插入一个新行，选择一个单元格或用鼠标右键单击。选择下拉菜单命令 **插入 (Insert) > 行 (Row)**。行被插入在光标位置；其后的行则向下移动一行。
- 欲在图当前最后一行下方插入一行，将光标放在最后一行的任何一个单元格中，并按下箭头键。
- 欲删除一个单元格或行，增亮单元格或行，并用鼠标右键单击。选择弹出菜单命令 **删除 (Delete) >**




**选项 (Selection)**。如果删除一行，其后的行（如果有）则向上移动一行。

- 欲选择整个状态图，在行号上方的左上角单击一次。

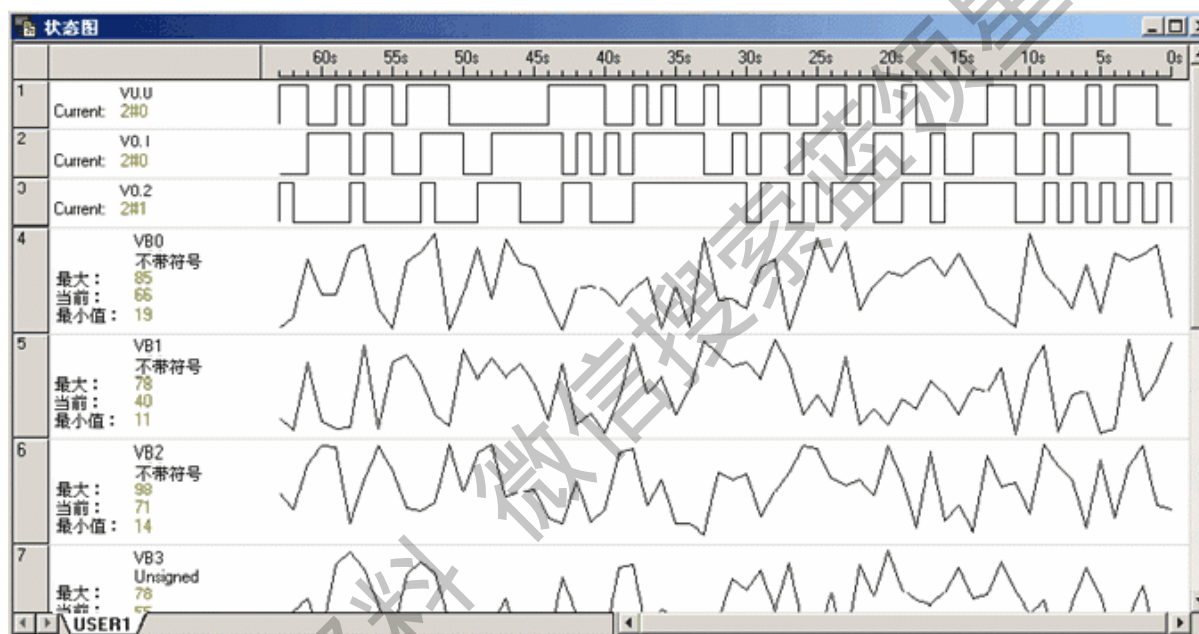
### 趋势显示详细说明



(返回顶端)

您可以使用下列方法在状态图的表格视图和趋势视图之间切换：

- 使用菜单命令 **检视(V) > 检视趋势(T)**。
- 用鼠标右键单击状态图，然后复选或取消复选 **作为趋势检视** 命令
- 单击调试工具条的趋势视图按钮 

趋势图用随时间而变的PLC数据之图形绘图跟踪状态数据：您可以就现有的状态图在表格视图和趋势视图之间切换。新的趋势



倘若在选取切换趋势按钮  之前没有选取图状态按钮 ，供趋势功能使用的状态收集将不会开始。如要启动PLC数据收

您可以通过单击暂停趋势视图按钮  或选取 **调试(D) > 暂停趋势图(P)** 命令，凝固趋势绘图。

### 趋势视图词汇表

**趋势显示** 随时间而变的PLC数据之图形绘图，通过选取“切换趋势”按钮由预配置的状态图创建。

**趋势绘图**

就在状态图中已配置的每个地址而言，该地址的PLC数据之屏幕位置会以时间为自变量得到显示。您不能调整趋势绘图的宽度。

**数据点或趋势点**

数据点意指出自PLC中某地址的单个数据。数据点是在相应地址的趋势绘图中赋予最后一个像素的数值。

**Range**

趋势数据的绘图在最小值和最大值之间多个数据的一个跨度上做出。此范围通常定义了数值的某个固定跨度。

**自由范围**

如果没有给趋势绘图选取范围，则最小值和最大值之间的数据范围会动态变化。

**抽样速率或收集速率**

定义由PLC收集数据的速率。此速率将是固定的，并且将基于选定的趋势时间单位计算得出。

**趋势窗口**


用于描述对趋势数据进行绘图的时间窗口；换句话说，由趋势绘图屏幕宽度代表的时间跨度。

**实时趋势**

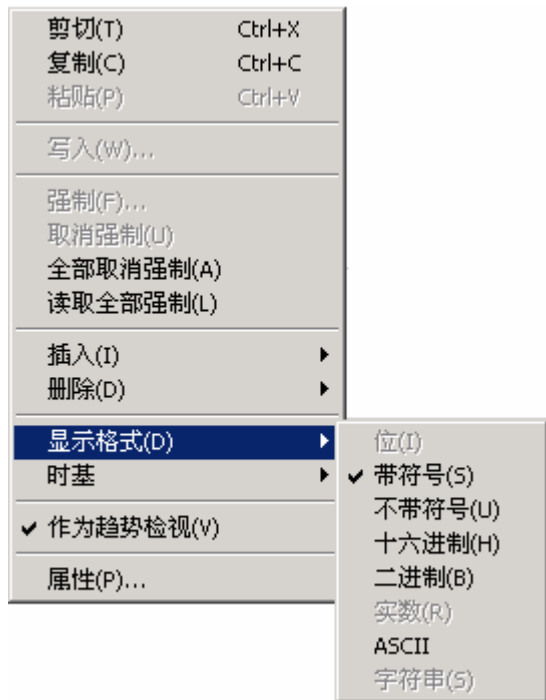
实时趋势意指以实时方式收集和在趋势窗口中显示数据。一旦此数据填满趋势窗口，最先的数据就会从趋势绘图上消失。

趋势图的单次读取和写入按钮与状态图上的相应按钮作用相同。‘强制’、‘取消强制’、‘全部取消强制’及‘读取全部强制’的作用与在状态图中相似。

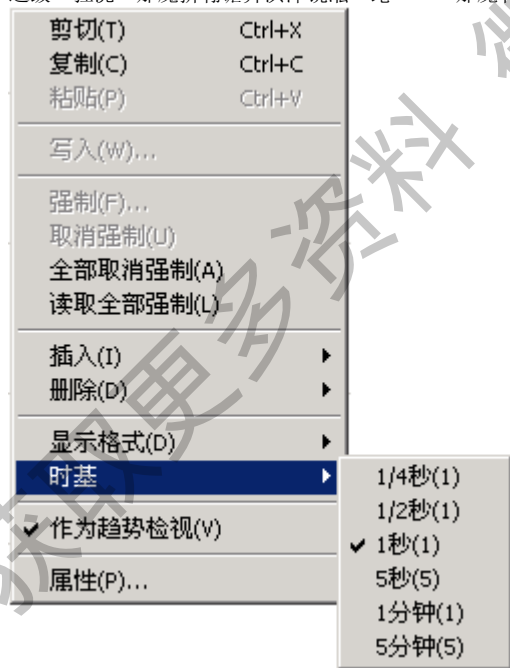
实时趋势功能不支持历史趋势。这意味着超出一个趋势窗口之时间跨度的趋势数据不会得到保留。

将光标放在分隔趋势行的横线上直至出现双箭头光标 。在光标为双箭头时，向上拖动以减少或向下拖动以增加高度。

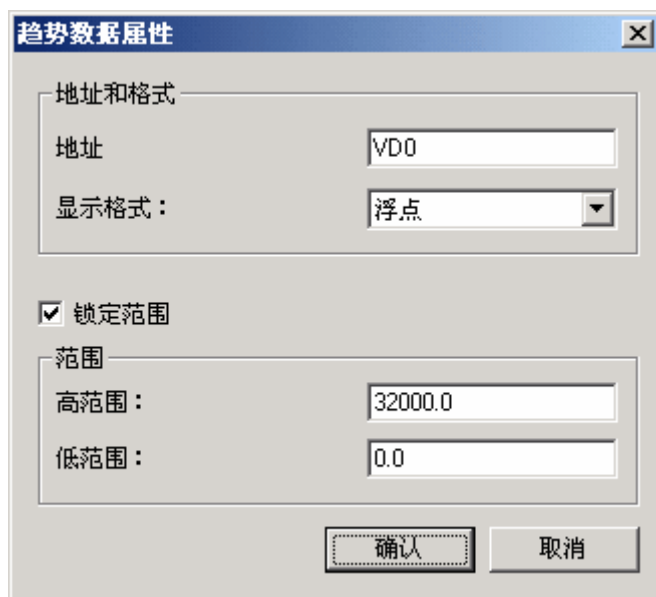
在单击并选定一趋势行后，用鼠标右键单击会显示一弹出菜单，由此可指定该行的显示格式。变更显示格式及添加或删除行也会改变状态图的表格视图。



您可以用秒或分为单位设置趋势绘图的时基，其默认值为1秒。配置的时基会在趋势图上用s（秒）或m（分）显示，其位置为



在单击并选定一趋势行后，用鼠标右键单击会显示一弹出菜单，由此可指定此行的趋势数据属性。



由“趋势数据属性”对话框，您可以对趋势绘图做出下列变更：

- 编辑地址（例如将VW0更改为VW10）。
- 增加一个新地址槽 如果选定的点以前没有得到配置。
- 如前面的上下文菜单讨论所述，更改显示格式类型。
- 基于选定的显示格式设置范围。

如果显示格式被修改，所有数据将会重新显示，并且可能因选定的新格式而改变。如果这些格式为数值兼容，例如带符号或不带符号，此趋势绘图将只会由可能的范围更改而改变。如果有些数值现在是在旧范围之外，并且范围已锁定，那么此绘图将在最大或最小范围值处截短。如果此绘图配置为自由范围，那么如有必要，这些数据将使用新的范围值重新显示。

按照默认规定，固定范围跨度不会启用；因此，这些趋势图绘图将为自由范围。如要能够针对给定趋势点选取范围值，请参见【段 A 备用】[输入输出】](#)。

由此对话框对地址或显示格式做出的任何更改将同时影响状态图和趋势图。

另请参阅：

[调试和监控特征概述 \(GS 7.1\)](#)

[如何在程序编辑器中显示状态 \(GS 7.2\)](#)

[如何执行有限次数扫描 \(GS 7.4\)](#)

[如何下载程序 \(GS 6.3\)](#)

[时间标记不匹配错误 \(确保编程设备中的项目与 PLC 中的项目相匹配\)](#)

[交叉引用和元素用法 \(确保程序编辑不引起重复赋值\)](#)

[在运行模式中进行程序编辑](#)


[PLC 运行 / 停止模式](#)

[在停止模式中写入和强制输出](#)

## 9.10 打印

### 9.10.1 打印项目和文档

请使用下列方法之一打印程序和项目文档：

- 单击“打印”按钮。
- 选择文件 (File) > 打印 (Print) 菜单命令。
- 按 Ctrl+P 快捷键组合。

本标题讨论下列主题：

[选择打印机和色彩选项](#)

[从单个项目元件打印网络或行范围](#)

[打印多个项目元件](#)

[改动打印顺序](#)

[选择打印选项](#)  
[页面设置](#)  
[如何打印帮助标题](#)  
[打印系统数据块信息](#)

### 选择打印机和色彩选项

[返回顶端](#)

在“打印机名称”列表框中显示的选项取决于为您的个人计算机 / 编程设备配置的打印机。



**提示：**如果您的打印机是黑白打印机，核实已经取消选择“彩色打印”复选框。否则，打印件中的网络注解可能不清楚。

### 从单个项目元件打印网络和行范围

[返回顶端](#)

您可以使用多个程序块、符号表和状态图。如果您希望从程序的单个程序块打印一系列网络，或从单个符号表或状态图打印一行网络，请按照以下步骤操作。

- 选择适当的复选框，并使用“范围”域指定打印的元素，或
- 增亮一段文本、网络或行，并选择“打印”。

如果您选择增亮项目的一节用于打印，则应检查以下条目：

- 在“打印内容 / 顺序”帧中已选取的正确编辑器。
- 在“范围”条目框中选择正确的POU（如适用）。
- POU“范围”条目框空闲正确的单选按钮，“范围”条目框中显示正确的数字。

例如，您可能希望从USR1符号表打印行6-20，则应采取以下两个步骤之一：

- 仅选择“打印内容 / 顺序”题目下方的“符号表”复选框以及“范围”下方的“USR1”复选框，定义打印范围6至20，或
- 在符号表中增亮6-20行，并选择“打印”。



**提示：**如果您希望继续选择打印程序的其他元件，您对第一个元件进行的“范围”选择则被禁用，将打印每个元件的全部范围。例如，您可以检查梯形图并指定网络范围。如果您选择希望打印的其他程序元件，例如符号表，梯形图的范围选择被禁用，“打印”命令将打印LAD网络的全部范围以及所有符号表信息。

#### 打印多个项目元件

[返回顶端](#)

选择“打印内容”标题下方您需要的所有元件复选框。请记住，打印多个项目元件时不能进行范围选择。



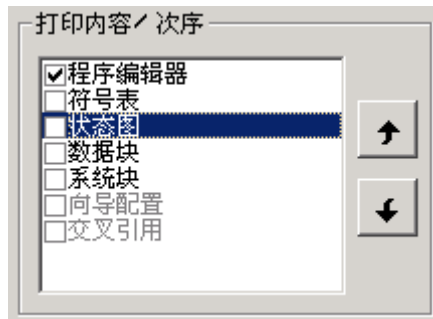
**提示：**每次单击复选框时，即在打开或关闭复选框之间切换。如果选择几个复选框，而您却不需所有这些复选框，单击您不需要的复选框，取消选择这些复选框。

#### 改动打印顺序

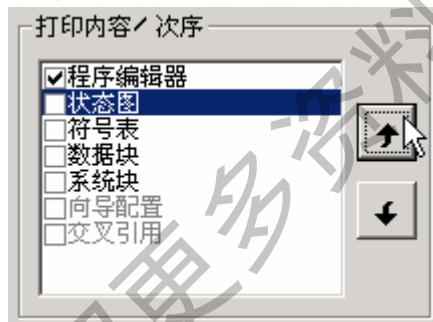
[返回顶端](#)

您可以增亮具体元素，用上箭头和下箭头将元素移至适当的位置，改变打印顺序。例如，如果您希望首先打印状态图：

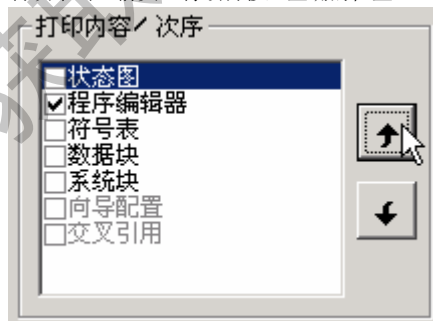
增亮您希望移动的项目。



单击位于帧右侧的上箭头，将项目移动一个位置。



再次单击上箭头，将项目移至正确的位置。

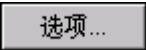


除非另行说明，默认打印顺序如下：  
打印区                      默认顺序

程序编辑器 (LAD、FBD、STL)	1
符号表	2
状态图	3
数据块	4
系统块	5
向导配置	6
交叉引用	7

### 选择打印选项

[返回顶端](#)

从“打印”对话框，单击  按钮，更改打印选项。

单击标记，修改打印选项。选择或清除复选框，在选项打开和关闭之间切换：

单击“程序编辑器标记”配置：

**属性** 打印程序组织单元的作者和时间标记属性。

**局部变量表** 打印所有选择的POU局部变量表。

**如有可能，不要分割网络** 在新页中打印网络（仅限LAD/FBD）。

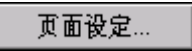
单击“数据块标记”配置：

**属性** 打印数据块属性。

### 页面设置

[返回顶端](#)

欲格式化项目打印件，选择文件 (File) > 页面设置 (Page

Setup) 菜单命令，或从主“打印”对话框单击“页面设置”  按钮。



使用“页面设置”对话框的“页边距”域输入：

- 上方页边距
- 下方页边距
- 左方页边距
- 右方页边距

在“页面设置”对话框中，您还可以输入页眉 / 页脚文本的位置、页面方向和纸张来源。

单击“页眉 / 页脚”按钮以设置页眉 / 页脚文本。



“页眉 / 页脚”对话框提供页眉或页脚中使用的条目工具条。欲选择某一条目，将光标置于您希望选择的条目域中（页眉或页脚），并单击适当的按钮。对于页脚，您可利用批注框附加选项。批注框允许您建立带标题的域（例如，“签名”或“注解”），以便更好地定制页脚空间。

“页眉 / 页脚”工具条按钮如下：



**项目名称**（完整路径）



**对象名称** 对于新项目，打印时页眉会自动包括对象名称（POU、符号表、状态图或数据块）。打印在较早版本STEP 7-Micro/WIN中建立的项目时，您可在页眉中增加对象名称：单击“对象名称”按钮或在页眉 / 页脚对话框的“页眉文本”域中键入%OBJECT。



**日期**



**时间**



**页号**



**左对齐**（页眉 / 页脚）



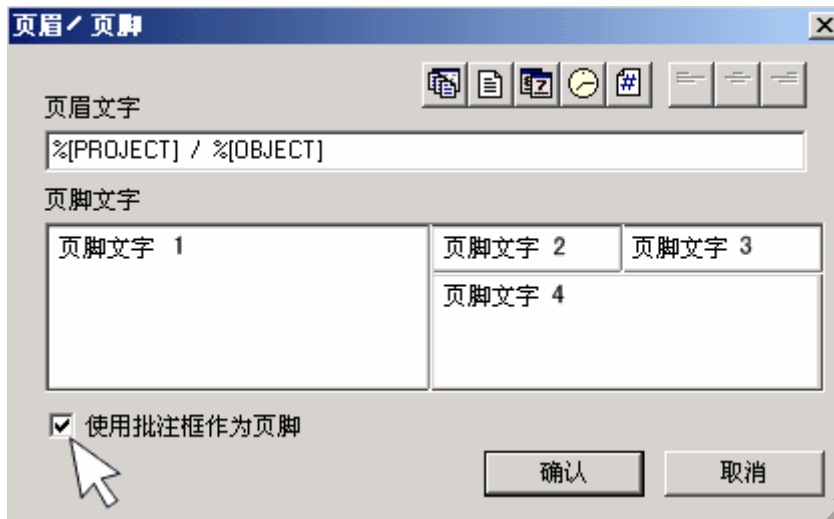
**居中**（页眉 / 页脚）



**右对齐**（页眉 / 页脚）

选择选项后，在打印件页眉或页脚中键入您希望显示的文字。您可能需要在页眉和页脚文字之间插入空格。单击“确定”按钮，确认设置，或单击“取消”按钮，取消设置。

单击“使用批注框作为页脚”复选框，即可在框架内添加您自己的页脚文字。



欲检视页眉和页脚，选择菜单命令文件（Print）> 打印预览（打印预览）。

#### 如何打印帮助：

[返回顶端](#)

您可从“目录”浏览器打印各组标题或单个标题。

- 欲从“目录”浏览器打印书中的所有标题，选择书标题，并使用位于浏览器底部的“打印”按钮。
- 欲从“目录”浏览器打印单个标题，选择题目标题，并使用位于浏览器底部的“打印”按钮。
- 欲从单个标题中打印，使用位于题目标题上方的“打印”按钮。

#### 打印系统数据块信息

[返回顶端](#)

育从“系统数据块”对话框打印信息，选择“打印”对话框的“打印内容 / 顺序”域下方的“系统块”复选框。信息按下列顺序打印：

端口信息  
保留范围  
密码  
背景时间  
输出表  
输入过滤器  
模拟输入过滤器  
脉冲捕获位

#### 注释：

您不能改动打印件的格式；标准字体和大小是 ARIAL 10点。  
当“系统数据块”对话框打开时，不能打印。

系统数据块打印件包括：

- 引言段落
- 用蓝色粗体下划线文本标出的条目
- 用6个点分开的标题和第一行
- 用12个点分开的最后一行文本和下一个标题

#### 注释：

- 选择“打印系统块”选项后，“范围”域变成非现用。
- 必须下载配置参数，才能使之生效。并非所有的 PLC 类型均支持每一个系统块选项。对于当前选择的 PLC 无效的数据被排除在打印件之外。

#### 另请参阅：

[如何打印 \(GS 8.1\)](#)



[打印预览](#)  
[应用程序用户参考手册](#)  
[入门指南目录](#)

获取更多资料 微信搜索蓝领星球

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part



获取更多资料 微信搜索 蓝领星球

## 10 错误讯息

### 10.1 STEP 7-Micro/WIN编译程序错误

**全部编译**指令按照下列顺序编译全部三个块。

1. 程序块 (OB1、全部子例行程序、全部中断例行程序)
2. 数据块 (DB1)
3. 系统块 (SDB0)

**编译**指令编译当前块，当前块为现用编辑程序。

编译程序块 (OB1、全部子例行程序、全部中断例行程序) 的过程中，编译程序在编译每个独立网络时遵守下列规则。

1. 编译程序检查对不存在的子例行程序的调用。
2. 编译程序检查对子例行程序的无效调用。
3. 编译程序检查指令操作数的非法语法。
4. 编译程序确认每个指令操作数，而且在 IEC 1131-3 的情况下，编译程序解决超载指令问题。
5. 编译程序检查无效网络结构。
6. 编译程序检查因选择的 PLC 中不存在的特征而造成的错误。

下面列举 STEP 7-Micro/WIN 编译程序错误代码。

十进制	错误代码	说明
1		所选 PLC 类型不支持 ENO 的使用。
2		非法操作数组合，OUT 中的直接地址操作数不可包含 IN2 中的直接地址的任何字节。
3		非法操作数组合，OUT 中的直接地址操作数不可包含 IN2 中的间接地址的任何四个字节。
4		非法操作数组合，OUT 中的相同直接地址累加器不能用作 IN2 中的间接地址累加器。
5		非法操作数组合，OUT 中的直接地址操作数的最低位字节不可包含 IN2 中的直接地址的任何字节。
6		非法操作数组合，OUT 中的直接地址操作数的最低位字节不可包含 IN2 中的间接地址的任何四个字节。
7		非法操作数组合，OUT 中的间接地址操作数不可包含 IN1 中的直接地址的任何字节。
8		非法操作数组合，OUT 中的间接地址操作数不可包含 IN1 或 IN2 中的间接地址的任何四个字节。
9		非法操作数组合，OUT 中的累加器操作数不能与 IN1 中使用的累加器相同。
10		非法操作数组合，OUT 中的累加器操作数不能与 IN2 中使用的累加器相同。
11		所选 PLC 类型不支持该指令。
12		所选 PLC 类型不支持参数化子例行程序。
13		所选 PLC 类型不支持带条件的 END (结束) 指令。
14		所选 PLC 类型不支持子例行程序。
15		无法解决超载指令问题。至少一个操作数必须是具有相关数据类型的局部或全局变量。
16		无法解决超载指令问题。并非全部超载操作数均具有相同大小或数据类型。
32		指令操作数非法语法。
33		指令操作数未定义全局符号或局部变量。
34		指令操作数编址模式无效。
35		指令操作数内存区无效。
36		指令操作数大小或数据类型无效。
37		指令操作数内存编址范围无效。
38		编译前必须先定义类属指令。
39		调用指令无对应的子例行程序。
42		无效间接内存区。
43		无效模拟输入或输出地址。
44		开路。
45		短路。
46		逆转功率流方向。
47		无效网络或网络太复杂无法编译。
48		非法放置正向转换、负向转换或 NOT (非) 接点。
49		计数器指令必须是网络中的唯一输出。
50		未知编译错误。
51		A LBL、NEXT、NOP、SCR 或 SCRE 指令在同一网络中不能有任何其他指令。
52		接点必须位于输出指令之前。
53		指令 END、RET 及 RETI 只能用作有条件输出。编译程序自动处理这些指令的无条件使用。
54		网络缺少输出指令。
55		指令操作数直接量 / 常量使用非法。
56		指令操作数直接量 / 常量数值超出范围。
57		指令操作数直接量 / 常量类型非法。
58		多堆栈指令 (CTU、CTD、CTUD、SBRx) 不允许用垂直线互联布尔输入逻辑。

59	在与多堆栈指令（CTU、CTD、CTUD、SBRx）相同的网络中不允许使用附加输出指令。
62	正向边缘号码重复使用
63	负向边缘号码重复使用
64	指令不支持ENO的使用。
65	输出线圈不支持功率流从线圈右侧流出。
66	指令未被识别为有效标准指令。
67	操作数数目与标准指令签名不匹配。
2700	非法STL指令

#### 系统数据块编译错误及警告

2750	所选PLC类型不支持设置端口1选项。
2751	所选PLC类型不支持设置模拟输入过滤选项。
2752	所选PLC类型不支持设置脉冲截取位选项。
2753	所选PLC类型不支持设置输出表格选项。
2754	所选PLC类型不支持配置输出表格的高位字节。
2755	对于所选的PLC类型，为端口'指定的波特率非法。
2756	对于所选的PLC类型，保留范围''非法。

#### 数据块编译错误

2770	指定的地址已经被分配一个数值。
2771	对于所选的PLC类型，地址无效。
2772	数据块条目非法语法。
2773	数值对于指定的地址过大。
2774	超出内部标记文本大小。
2775	非法DB 语法。

## 10.2 通讯错误

#### 十进制错误代码

#### 说明

300	未知通讯错误。
301	PLC处于错误模式，或者RUN/STOP（运行/停止）开关不在终端位置。
302	请求的操作不可能完成。
303	上载顺序错误。
304	请求的操作不可能完成。
305	请求的操作不可能完成。
306	PLC受密码保护，不允许进行请求的操作。
307	通讯语法名称错误。
308	发生编译错误。请检查非严重错误，了解详情。
309	数据块类型未知。
310	数据块位于EEPROM中。
311	数据块不位于PLC中。
312	配置信息中指定了一个无效参数。
313	数据块过大，PLC内存无法容纳。
314	通讯数据域的长度超出预料。
315	违反内部协调规则。
316	PLC受密码保护，不允许执行请求的命令。
317	密码中语法错误。
318	输入了不正确的密码。
319	链接已经合法化。存取已获得授权。
320	已解除链接。
321	PLC不受密码保护。
322	指定了无效的时间。
323	输入了无效的波特率。
324	指定了无效的端口号码。
325	尝试打开通讯端口失败。
326	关闭端口操作失败。
327	向PLC传输数据时出错。
328	从PLC接收数据时出错。
329	通讯超时。请检查端口号码、网址、波特率和连接的电缆。
330	存在网络地址错误。地址无效或者本地和远程地址之间存在冲突。
331	未设置PLC时钟。
332	PLC硬件故障。
333	非法对象存取。
334	无效地址。

335	不支持数据类型。
336	对象不存在或长度错误。
337	端口打开时无效请求。
338	无效通讯请求。
339	无效协议。
340	通讯错误：未找到MPI设备。
341	通讯错误：设备中断繁忙。
342	通知：PLC版本只支持地址改变。不修改最高地址、波特率、重试计数和间隔更新系数。
343	通知：操作系统不支持多台主设备PPI，协议被重设为单台主设备PPI。
344	未改变操作模式。
345	程序内的时间戳记与PLC内的时间戳记不匹配。

**调制解调器特定错误**

370	向调制解调器传输命令错误。
371	从调制解调器接收命令应答错误。
372	断开调制解调器应答错误。
373	建立调制解调器连接错误。
374	调制解调器初始化错误。
375	调制解调器必须处于联机状态才能执行该项操作。
376	检测到繁忙信号。
377	未选择本地调制解调器。
378	未选择远程调制解调器。
379	检测到调制解调器设置并在''位置编程，但调制解调器无法利用当前COM端口通讯。

**10.3 LAD/FBD编辑程序错误****十进制错误代码****说明**

500	水平线不能位于网络的第一列。
501	水平线不能位于网络的最后一列。
502	水平线不能位于成块单元上。
503	编辑程序处于报告状态时不允许进行编辑。
504	水平线不能与带立即标号的参数连接。
505	不能为与水平线或方框连接的参数设置立即标号。
506	不能为输出参数设置负标号。
507	只能修改功率流或布尔参数的标号。
509	不能从AND（与）或OR（或）方框删除两个默认输入。
510	方框在此位置必须有功率流输入。
511	不能删除行，由此产生的网络为非法网络。
512	带输入的方框指令不能位于网络的第一列。
513	竖线不能直接位于FUNCTION_BLOCK方框上方的成块单元上。
514	方框不能直接位于FUNCTION_BLOCK方框上方的成块单元上。
515	不能切换对参数化子例行程序的无效引用标号。
516	竖线不能位于网络的第一列。
517	由于主输入或输出上的立即标号，剪贴板中的方框不能位于当前位置。
518	方框不能位于网络的第一行。
519	由于次级功率流输入上的立即标号，不能剪切或复制方框。
600	竖线不能位于线圈上。
601	竖线不能位于方框上。
602	向上竖线不能位于网络的第一行。
603	向下竖线不能位于网络的最后一行。
604	竖线不能位于网络的最后一列。
605	接点和水平线不能位于网络的最后一列。
606	接点和水平线不能位于成块单元上。
607	线圈不能位于竖线上。
608	线圈不能位于逻辑紧靠右侧的位置。
609	线圈不能位于成块单元上。
610	方框不能位于网络的最后一列。
611	方框不能位于网络的最后一行。
612	因为网络空间不足，不能放置方框。
613	方框不能位于网络的第一列。
614	请求指令有一个无效签名，无法提取。
615	方框不能位于成块单元上。
616	因为将在该输出中附加功率流，方框在此位置必须有功率流输出。
617	永远不应显示该错误串。

618	编辑程序正在报告状态时，不能进行编辑。
619	请求将使网络非法。
620	未找到请求的元素。
621	未找到请求的元素。
622	请求将超出网络中允许使用的最大列数。
623	请求将超出网络中允许使用的最大行数。
624	不能删除列。由此产生的网络将是非法网络。
625	编辑程序根据需要增加和移除功率流符号，用户不能删除这些符号。
626	子例行程序无法直接自我调用。
627	提供的字符串与一条与类属指令类型发生冲突的指令相同。
628	该POU包含一个或多个对参数化子例行程序无效的引用。
629	不能编辑对参数化子例行程序无效的引用操作数。
630	不能对参数化子例行程序无效的引用执行操作。
631	不能提取有效子例行程序引用，该引用被删除。
632	不能执行请求的操作。
633	由于对参数化子例行程序的引用无效，不能执行请求的操作。

## 10.4 项目文件I/O错误

十进制错误代码	说明
1000	文件I/O过程中出现未说明的错误。
1001	无法找到请求的文件。
1002	全部或部分路径无效。
1003	超出最大打开文件数目。
1004	无法存取文件。
1005	尝试使用无效文件句柄。
1006	无法移除当前工作目录。
1007	无更多目录条目。
1008	尝试设置文件指针错误。
1009	出现硬件错误。
1010	未载入SHARE.EXE，或交换了共享区域。
1011	尝试锁定已经锁定的区域。
1012	保存操作过程中出现磁盘已满错误。
1013	已达到文件结尾。
1014	指定的文件是无效项目文件。
1015	时间戳记不匹配。
1016	无法在当前模式中打开IEC-1131项目。
1017	无法在当前模式中打开SIMATIC项目。
1018	项目文件包含一个或多个 STEP 7-Micro/WIN 不支持的旧结构。

## 10.5 PLC严重错误

严重错误导致PLC停止执行程序。取决于错误的严重程度，严重错误可以使PLC无法执行任何一个或全部功能。处理严重错误哪康脑谟谟 PLC进入安全状态，使PLC能对现有错误条件的询问作出应答。

检测到严重错误时，PLC执行下列任务。

- 更改为STOP（停止）模式
- 打开系统故障LED和停止LED
- 关闭输出

PLC一直处于该状态，直到严重错误得到纠正。下表列举可从PLC读取的严重错误代码及其说明。

STEP 7-Micro/WIN在PLC信息对话框中显示由PLC生成的错误代码以及简要说明。欲存取该对话框，选择菜单命令 **PLC > 信息...**

一旦纠正了引起严重错误的条件后，必须采用下列方法之一重新启动PLC。

- 关闭PLC电源后再重新打开，或
- 将模式开关从TERM（有条件）或RUN（运行）转至STOP（停止）位置。（只有在PLC出现严重错误和故障指示灯

重新启动PLC会清除严重错误条件并开始启动诊断测试。如果发现另一个严重错误条件，PLC再次设置系统故障LED；否则，PLC开始正常操作。

有几种可能的错误条件会使PLC无法通讯，在这种情况下，您无法检视PLC错误代码。此类错误表示硬件故障，要求修理PLC？  
 棕桓谋涑缘蚯冗 PLC内存不能解决这一问题。

十六进制错误代码	说明
0000	不存在严重错误
0001	用户程序校验和错误
0002	编译阶梯程序校验和错误
0003	扫描监视程序超时错误
0004	内部EEPROM失败
0005	用户程序内部EEPROM校验和错误
0006	配置参数内部EEPROM 校验和错误
0007	强迫数据内部EEPROM校验和错误
0008	默认输出表数值内部EEPROM校验和错误
0009	用户数据、DB1内部EEPROM校验和错误
000A	内存盒故障
000B	用户程序内存盒校验和错误
000C	配置参数内存盒校验和错误
000D	强迫数据内存盒校验和错误
000E	默认输出表数值内存盒校验和错误
000F	用户数据、DB1内存盒校验和错误
0010	内部软件错误
0011	比较接点间接编址错误
0012	比较节点非法浮点数值错误
0013	内存盒空白或该S7-200无法理解程序
0014	比较节点范围错误

## 10.6 PLC编译和运行时间错误

编译和运行时间错误属于非严重错误。非严重错误可能降低PLC的某些性能，但不会阻止PLC执行用户程序或更新I/O。

- **运行时间错误**是在RUN（运行）模式中检测到的非严重错误，在程序可监控和评估的特别内存（SM）位中反应。（欲了解有关这些SM位的详情，请参考[PLC非严重错误内存位置](#)。）启动时，PLC读取I/O配置，并将该信息存储在SM内存盒中。启动时，PLC读取I/O配置，并将该信息存储在SM内存盒中。如果PLC检测到I/O配置的区别，则会设置模块I/O配置。PLC欲重置配置改变位，模块I/O必须再次与存储在系统数据内存中的I/O配置相匹配。
- **运行时间编程错误**是在执行程序的过程中，您或程序造成的非严重错误条件。例如，一个编译程序时有效的间接地址指针被程序执行修改为指向超出范围的地址。使用[PLC > 信息](#)命令确定出现的错误类型。
- 只有通过修改用户程序才能纠正运行时间编程错误。运行时间编程错误在下次从STOP（停止）模式转换为RUN（运行）模式时会被清除。
- 在PLC编译程序时在下载中检测到**编译错误**（或程序编译错误）。如果存在编译错误，编译会暂停，并检索先前的程序（该程序自从先前编译以来为有效程序）。所有这些步骤均在PLC位于STOP（停止）模式时进行。可使用[PLC > 信息](#)命令显示编译错误。

### 十六进制错误代码

#### 编译错误（非严重）

0080	蚬 CPU无法生成可执行代码；请缩小程序大小	程
0081	幌乱纤唤 绶殖啥喔璧	堆
0082	日噶睿患徽槽噶航 欠	非
0083	ÛMEND，或主程序中不允许存在指令；增加MEND指令或移除不正确的指令	缺
0084	ò	保
0085	ÛFOR；增加FOR指令或删除NEXT指令	缺
0086	ÛNEXT；增加NEXT指令或删除FOR指令	缺
0087	俦昵 LBL、INT、子例行程序）；增加适当的标签	缺
0088	ÛRET或子例行程序中不允许存在指令；在子例行程序结尾处增加RET或移除不正确的指令	缺
0089	ÛRETI或中断例行程序中不允许存在指令；在中断例行程序结尾处增加RETI或移除不正确的指令	缺
008A		保

008B		至S
CR段非法JMP或从SCR段非法JMP		
008C		重
幢呢 LBL、INT、SBR)；为其中一个标签重新命名		非
008D		非
ū呢 LBL、INT、SBR)；核实未超出允许使用的标签数		非
0090		非
ú问 缓耸抵噶铃市礪褂攻牟问		范
0091		指
T砮媳ò ū 卍畔iii 患穀椴僮魔 段		
0092		
累剖 虻嬖洼砮媳ò m剖 畔iii 缓耸底奋蠢剖		
0093	超过 FOR/NEXT嵌套层数	
0095	缺少LSCR指令(载入SCR)	
0096	缺少SCRE指令(SCR结束)或在SCRE前出现不允许使用的指令	
0097	用户程序包含不带号码及带号码的EU/ED指令	
0098	尝试在配备不带号码EU/ED指令的程序中执行运行时间编辑	
0099	过多隐含程序段	
009B	非法索引(字符串操作,已指定该操作中的一个起始位置数值0)	
009C	超出最大指令长度	
009D	SDB0内检测到非法参数	
十六进制错误代码		运
惺奔得嗜坦收希ù茄现兀		无
0000		
砮		在
0001		
葱 HDEF方框之前启用HSC方框		输
0002		输
脛卸戏峙涑逋噶 峙渲烈丫 峙涓 HSC的点		尝
0003		尝
救峙涑逋噶 峙渲烈丫 峙涓 淙脛卸匣蚌涑 HSC的HSC		间
0004		TO
裁溢卸侠 谱缘蚌兄葱 ENI、DISI或HDEF指令		超
0005		在
裁谕甌傻偏桓 HSC/PLS之前执行第二个带有相同号码的HSC/PLS,与主程序中的 HSC/PLS发生中断例行程序冲突		尝
0006		在
颖喃反砮		时
0007		尝
DW(日写入时间)或TODR(日读取时间)数据错误		PT
0008		
蜚奋笱没e永 谱缘蚯短撞闯		
0009		
丝 0中同时执行XMT/RCV指令		
000A		
酩ù 葱辛礻惶踉糜谱噍 HSC的HDEF指令重新定义HSC		
000B		
丝 1中同时执行XMT/RCV指令		
000C		
雍肋淮嬖		
000D		
灾圃露丁速钟寐鞞迨涑		
000E		
○概况段号被设为0		
000F		
*		
冉辖拥闾噶航杏匏椒欠去 质		在
0010		当
°PTO操作模式不允许的命令		非
0011		非
°PTO命令代码		
0012		
°PTO轮廓表		
0013		
°PID环路表		



0091		范
T 砒螳ò ù 洩畔iii 患穀椴僮魔 段		
0092		指
罟剖 虻砒螳ò ù 剖 畔iii 缓耸底奋蠢剖		
0094	用地址信息向非易失性内存写入范围错误	
009A	在用户中断中尝试转入自由端口模式	
009B	非法索引（字符串操作，已指定该操作中的一个起始位置数值 0）	
*比较接点错误是唯一生成严重和非严重条件的错误。		
生成非严重错误条件的原因是保存错误的程序地址。		

另请参阅：  
[PLC非严重错误内存位置](#)

## 10.7 PLC非严重错误内存位置

非严重错误可能降低 PLC 的某些性能，但不会使 PLC 无法执行用户程序和更新 I/O。为了帮助您调试程序，与错误条件相关的信息存储在特别内存（SM）中，用户程序可存取这些特别内存位置。例如，如果您不希望存在某些非严重错误条件时继续处于 RUN（运行）模式，您可以让用户程序在出现不良条件时强迫转换至 STOP（停止）模式。

下表列举并说明特别内存非严重错误信息。

SM位	非严重错误说明	SM字节	非严重错误说明
SM0.2	保留的数据丢失	SMB9	模块0 I/O错误字节
SM1.3	除以0错误	SMB11	模块1 I/O错误字节
SM3.0	奇偶校验错误	SMB13	模块2 I/O错误字节
SM4.0	通讯中断队列溢出	SMB15	模块3 I/O错误字节
SM4.1	输入中断队列溢出	SMB17	模块4 I/O错误字节
SM4.2	定时中断队列溢出	SMB19	模块5 I/O错误字节
SM4.3	运行时间编程故障	SMB21	模块6 I/O错误字节
SM5.0	I/O 错误（设置任何I/O错误位）		
SM5.1	过多数字点		
SM5.2	过多模拟点		

## 10.8 时间戳记不符

该警告讯息表示项目时间戳记与 PLC 中程序的时间戳记不匹配。这可能说明程序不同，在这种情况下，继续当前操作可能有危险。

### 哪些措施可以修改程序时间戳记？

每个程序包含两个不同的时间戳记：“建立”时间戳记和“最近修改”时间戳记。利用“新项目”选项建立项目时，设置建立时间戳记。

“最近修改”时间戳记用于表示用户最近修改程序的时间。有许多条件均可造成设置“最近修改”时间戳记：

1. 编辑指令或程序块编辑程序中的操作数。
2. 增加、删除或修改局部或全局符号。
3. 增加或删除POU。
4. 编译程序块。
5. 下载程序块（会自动编译程序块，从而设置最近修改时间戳记）。

请注意，虽然所有这些措施会设置最后修改的时间戳记，但并不一定表示程序不同。因此，STEP 7-Micro/WIN提供“比较”选项，允许您确定程序是否真正不同。

### 如何确定程序是否真正不同？

您可以按“比较”按钮，比较 PLC 中的程序块和项目的程序块，比较结果允许您确定是否继续执行状态操作。



#### 当心

如果比较结果表明 PLC 中的程序与项目并不完全相同，继续执行状态操作将导致无法预测的程序表现。

### 如何使用时间戳记同步？

向 PLC 下载一个新项目可使时间戳记同步，从而允许您运行状态。

## 10.9 EM241调制解调器模块错误代码

错误	说明
0	无错误
<b>电话线错误</b>	
1	未出现拨号音
2	线路繁忙
3	拨号错误
4	无回答
5	连接超时（1分钟内无连接）
6	连接异常终止或出现未知应答
<b>命令错误</b>	
7	数字寻呼讯息包含非法数字
8	电话号码（电话输入）超出范围
9	讯息或数据传送（讯息或数据输入）超出范围
10	文字讯息或数据传送讯息错误
11	讯息传送或数据传送电话号码错误
12	作业不允许（例如，尝试设置为零）
<b>服务供应商错误</b>	
13	讯息传送服务无应答（超时）
14	因未知原因讯息服务中断
15	用户异常中断讯息（禁用命令位）
<b>TAP — 服务供应商返回的文字寻呼和SMS讯息错误</b>	
16	收到远程中断（服务供应商异常终止通话）
17	讯息服务不接受登录进入（不正确的密码）
18	讯息服务不接受信息块（检查和或传输错误）
19	讯息服务不接受信息块（原因不详）
<b>UCP — 服务供应商返回的SMS讯息错误</b>	
20	未知错误
21	检查和错误
22	句法错误
23	系统不支持作业（非法命令）
24	此时不允许作业
25	呼叫阻拦启用（黑名单）
26	呼叫者地址无效
27	授权故障
28	合法化代码故障
29	GA无效
30	不允许重复
31	重复合法化代码，故障
32	不允许优先呼叫
33	优先呼叫合法化代码，故障
34	不允许紧急讯息
35	紧急讯息合法化代码，故障
36	不允许反向收费
37	反向收费合法化代码，故障
38	不允许延迟交付
39	新AC无效
40	不允许新合法化代码
41	标准文字无效d
42	时间期限无效
43	系统不支持讯息类型
44	讯息过长
45	请求的标准文字无效

46	讯息类型对寻呼机类型无效
47	SMSC中未发现讯息
48	保留
49	保留
50	用户挂断
51	不支持传真群组
52	不支持传真讯息类型

## 数据传送错误

53	讯息超时（远程设备无应答）
54	远程CPU忙于上载或下载
55	存取错误（内存超出范围，非法数据类型）
56	通讯错误（未知应答）
57	应答中的检查和或CRC错误
58	为回叫设置的远程EM 241（不允许）
59	远程EM 241拒绝提供的密码
60至127	保留

## 指令使用错误

128	无法处理请求。模块忙于处理另一项请求或该请求无 START（开始）脉冲。
129	调制解调器模块错误： -
	• 调制解调器模块位置或用调制解调器扩充向导配置的 Q 内存地址与实际位置或内存地址不符。 • 请参阅 <a href="#">SMB8至SMB21（I/O模块ID和错误寄存器）</a> 。

## 10.10 EM253位置模块错误代码

## 指令错误代码

错误代码	说明
0	无错误
1	被用户异常中止
2	配置错误 使用EM 253控制面板诊断程序标记检视错误代码
3	非法命令
4	由于无有效配置被异常中止 使用EM 253控制面板诊断程序标记检视错误代码
5	由于无用户权利被异常中断
6	由于无定义的参考点被异常中断
7	由于STP输入现用被异常中断
8	由于LMT-输入现用被异常中断
9	由于LMT+输入现用被异常中断
10	由于执行位置故障被异常中断
11	没有为指定的轮廓配置的轮廓块
12	非法操作模式
13	操作模式不支持本命令
14	轮廓块中的非法步进数目
15	非法方向改动
16	非法距离
17	在达到目标速度之前发生RPS触发
18	RPS现用区域宽度不足
19	速度超出范围
20	无足够距离执行所需的速度改动
21	非法位置
22	未知零位置
23至127	保留
128	位置模块正在忙于处理另一个指令
129	位置模块错误
130	位置模块未启用

131 由于模块错误或模块未被启用，位置模块无法使用（请参阅 POSx\_CTRL 状态）  
132至255 保留

#### 模块错误代码

错误代码	说明
0	无错误
1	无用户权利
2	未出现配置块
3	配置块指示字错误
4	配置块尺寸超出可供使用的V内存
5	非法配置块格式
6	指定的轮廓过多
7	非法STP_RSP规格
8	非法LMT_-RSP规格
9	非法LMT+_RSP规格
10	非法FILTER_TIME规格
11	非法MEAS_SYS规格
12	非法RP_CFG规格
13	非法PLS/REV数值
14	非法UNITS/REV数值
15	非法RP_ZP_CNT数值
16	非法JOG_INCREMENT数值
17	非法MAX_SPEED数值
18	非法SS_SPEED数值
19	非法RP_FAST数值
20	非法RP_SLOW数值
21	非法JOG_SPEED数值
22	非法ACCEL_TIME数值
23	非法DECEL_TIME数值
24	非法JERK_TIME数值
25	非法BKLSH_COMP数值

## 10.11 以太网 / 互联网模块和ETHx\_CTRL指令的CP243-1错误代码

双击在PLC>信息窗口底部显示的以太网模块，检视状态和模块错误代码。  
请查阅CP243-1 IT技术参考资料中的编程细节：

SIMATIC NET  
CP243-1 IT通讯处理器  
工业以太网和信息技术

错误字 十六进制	说明 十进制	反应 / 校正	功能?? (ETHx_)
16#0001	1	底板总线超时。	自动重新启动
16#000D	13	由于用户程序触发了一个新配置，所有数据传输被终止。	_CTRL
重新启动系统	_CTRL		
16#0030	48	无法在指定的时间内用CPU调用配置。	自动重新启动
	_CTRL		
16#0031	49	在S7-200 CPU内存中未找到带有正确语法的CDB配置。	
检查配置。	_CTRL		
16#0032	50	配置数据（CDB、NDB、IDB）的CRC检查和错误。	
用STEP 7 Micro/WIN设置 CP 243-1 IT新配置。		检查用户程序，查看配置数据是否被意外盖写。	_CTRL
16#0033	51	CP 243-1 IT的配置数据包含错误或未正确存储。	用STEP 7
Micro/WIN设置 CP 243-1 IT新配置。	_CTRL		
16#0034	52	CDB指针错误或未载入CDB。	
核实在用户程序开始调用子程序ETHx_CTRL。		检查CDB指针（在SM内存区字节偏移46至49）。	
_CTRL			
16#0035	53	已传输配置的格式识别符错误。	检查配置。
	_CTRL		

16#0036	54	配置中的TSAP不具有独特性或出现一次以上。	检查配置。
16#0038	56	配置不具有独特性（IP地址错误，客户机和服务器均未配置，STEP 7	
Micro/WIN 32频道未激活）。		检查配置。	
16#003A	58	CP 243-1 IT的模块名称在配置中被更改。	检查配置。
16#003B	59	配置有一个无效IP地址。	检查配置。
16#003C	60	配置有一个无效网关地址。	检查配置。
16#003D	61	配置在“保持现用”参数中有一个无效数值。	检查配置。
16#003E	62	未从内存或S7-200 CPU或经由BOOTP收到有效配置。	
从S7-200 CPU内存或经由BOOTP服务器尝试循环，获取一个有效配置。			
16#0042	66	NDB在读取/写入作业或长度中包含语法错误。	检查配置。
16#0093	147	BOOTP作业失败。	自动重新启动
16#0094	148	BOOTP服务器回答包含无效数据。	自动重新启动
16#0095	149	TCP/IP堆栈不接受指定的监管时间（保持现用）。	
自动重新启动			
16#0096	150	TCP/IP堆栈不接受为客户机指定的IP地址。	自动重新启动
16#0097	151	TCP/IP堆栈不接受指定的子网掩模。	自动重新启动
16#0098	152	TCP/IP堆栈不接受指定的网关地址。	自动重新启动
16#00B0	176	IDB包含无效数值。	用STEP 7 Micro/WIN准备配置。
16#00B1	177	IDB的CRC检查失败。	用STEP 7 Micro/WIN准备配置。
检查用户程序，查看配置数据是否被意外盖写。			
16#00B2	178	IDB中未指定管理员名称。	用STEP 7 Micro/WIN准备配置。
16#00B3	179	管理员密码检查失败或未指定密码。	用STEP 7
Micro/WIN准备配置。			
16#00B7	183	管理员密码过短。	用STEP 7 Micro/WIN准备配置。
16#00BB	187	错误IDB格式	用STEP 7 Micro/WIN准备配置。
16#00BD	189	文件系统初始化失败。	从文件系统删除文件。
16#00BF	191	无法在文件系统中设置系统文件。	
从文件系统中删除文件。			
16#00F0	240	S7-200 CPU无法识别CP 243-1 IT。	检查S7-
200系统的配置和设置。			
16#00F1	241	输出字节地址与S7-200系统上的CP 243-1	
IT当前位置不兼容，根据配置要求，应当能够经由该输出字节地址存取CP 243-1 IT。			检查S7-
200系统的配置和设置。			
16#0100	256	底板总线超时。	自动重新启动
to			
to			
16#0108	264		
16#8080	32896	CP 243-1 IT尚未完全启动。	必要时检查24
V电源。			

## 10.12 信道和ETHx-XFR/ETHx\_CFG的CP243-1错误代码

双击在PLC>信息窗口底部显示的以太网模块，检视频道状态。  
请查阅CP243-1 IT技术参考资料中的编程细节：

SIMATIC NET  
CP243-1 IT通讯处理器  
工业以太网和信息技术

错误字节 十六进制	说明 十进制	反应 / 校正	晓得??
--------------	-----------	---------	------

		(ETHx_)	
16#01	1	底板总线超时。	自动重新启动 _XFR_CFG_EMAIL
_FTPC			
16#02	2	由于“异常中止”命令，数据存取被终止。	
可启动一项新读取/写入作业。			_XFR_EMAIL_FTPC
16#03	3	传输参数“数据”或“邮件”或OP超出配置区域范围。	
检查用户程序或配置。			_XFR_EMAIL_FTPC
16#04	4	无法在S7级别设置连接。	建立连接的尝试重复。
_XFR			
16#05	5	连接被终止或在未准备就绪的频道上尝试执行读取/写入作业。	
检查通讯对方的连接路径或通讯对方的配置。			_XFR
		无法建立连接或连接被终止，或CP243-1 IT上的电子邮件配置文件	
被更改，或尝试执行一项作业，但电子邮件服务未设置正确的参数。			检查通讯对方的连接路径。
检查CP 243-1 IT上的.edb文件是否具有独特性和完整性。			EMAIL
		无法建立连接或连接被终止，或CP 243-1 IT上的FTP配置被更改，或	
尝试执行一项作业，但FTP客户机服务未设置正确的参数。			检查通讯对方的连接路径。 检查CP 243-1
IT上的.fdb文件是否具有独特性和完整性。			_FTPC
		FTP连接被终止。	检查通讯对方的连接路径。
检查FTP服务器配置的完整性。			_FTPC
		无法建立连接或连接被终止，或尝试执行一项作业，但HTTP服务器	
未设置正确的参数。			检查通讯对方的连接路径。 检查HTTP服务器配置的完整性。
16#06	6	应答包包含逻辑错误。	可触发一项新读取/写入作业。 检查配置。
			_XFR_EMAIL_FTPC
16#07	7	读取作业失败。	可触发一项新读取/写入作业。
应当检查读取作业的参数。			_XFR
16#08	8	写入作业失败。	可触发一项新读取/写入作业。
应当检查写入作业的参数。			_XFR
16#09	9	频道未配置。	可触发带其他参数的新读取/写入作业。
			_XFR_EMAIL_FTPC
16#0A	10	频道被配置为“服务器”，但却尝试触发读取/写入作业。	
可触发带其他参数的新读取/写入作业。			_XFR
16#0B	11	以前的读取/写入作业尚未完成。	可触发一项新读取/写入作业。
评估以前的读取/写入作业的“完成”		返回参数。	_XFR_EMAIL_FTPC
16#0C	12	无效作业识别符	可触发一项新读取/写入作业。
			_XFR_EMAIL_FTPC
16#0D	13	由于用户程序触发了一个新配置，所有数据传输被终止。	重新启动系统
			_XFR_CFG_EMAIL
16#0E	14	无法从CPU读取需要传输的数据。	
检查配置，尤其是.fdb文件的配置。			_FTPC
16#0F	15	无法将需要传输的数据写入CPU。	
检查配置，尤其是.fdb文件的配置。			_FTPC
16#10	16	无法与一个已配置电子邮件服务器连接。	
检查与已配置电子邮件服务器的连接路径。 检查配置，尤其是.edb文件的配置。			EMAIL
16#11	17	注册被电子邮件服务器拒绝。	检查并确定电子邮件服务器的配置是否正确。 检查配置，尤其是.edb文件的配置。
			EMAIL
16#12	18	生成需要发送的电子邮件时出错。	
检查配置，尤其是.edb文件和电子邮件中使用的占位符的配置。			EMAIL
16#13	19	发现电子邮件配置不正确。	检查配置，尤其是.edb文件和电子邮件中使用的占位符的配置。
			EMAIL
16#14	20	生成的电子邮件文字过长。	检查配置，尤其是.edb文件和电子邮件中使用的占位符的配置。
			EMAIL
16#17	23	无法与已配置FTP服务器建立连接。	检查与FTP服务器的通讯路径。
检查配置，尤其是.fdb文件的配置。			_FTPC
16#18	24	用FTP服务器注册失败。	检查并确定FTP服务器配置是否正确。
检查配置，尤其是.fdb文件的配置。			_FTPC
16#19	25	FTP传输协议错误	检查并确定FTP服务器配置是否正确。
检查并确定是否总是能够存取该文件。		检查配置，尤其是.fdb文件的配置。	_FTPC
16#1A	26	传输文件时超出内存范围或指定的文件长度与实际文件长度不符。	
检查并确定该文件是否有足够的内存空间。 检查配置，尤其是.fdb文件的配置。			_FTPC
16#1B	27	经由FTP传输数据时出错。	检查并确定FTP服务器配置是否正确。
检查并确定是否存在网络故障。 检查配置，尤其是.fdb文件的配置。			_FTPC
16#80	128	不存在外部24 V。	应用电压或等到准备就绪。 _XFR_CFG
_EMAIL_FTPC			
16#81	129	该频道未准备就绪或发生故障。	等到准备就绪。
评估子程序ETHx_CTRL的返回参数。			_XFR_CFG_EMAIL_FTPC
16#82	130	该频道繁忙或该指令无“起始”脉冲。	等到准备就绪。
激活“起始”输入。			_XFR_CFG_EMAIL_FTPC
16#83	131	用一个非法频道号码启动一项作业。	可触发一项新读取/写入作业。

检查用户程序。 16#84	132	<code>_XFR_EMAIL_FTPC</code> 用一个非法数据块号码启动一项作业。	可触发一项新读取 / 写入作业。
检查用户程序。		<code>_XFR_EMAIL_FTPC</code>	

## 10.13 ETHx\_EMAIL指令的CP243-1 IT错误代码

双击在PLC>信息窗口底部显示的以太网模块，检视频道状态。  
请查阅CP243-1 IT技术参考资料中的编程细节：

SIMATIC NET  
CP243-1 IT通讯处理器  
工业以太网和信息技术

错误字节 十六进制	十进制	说明	反应 / 校正	触发?? (ETHx_)
16#01 _FTPC	1	底板总线超时。	自动重新启动	<code>_XFR_CFG_EMAIL</code>
16#02 可启动一项新读取 / 写入作业。	2	由于“异常中止”命令，数据存取被终止。		<code>_XFR_EMAIL_FTPC</code>
16#03 检查用户程序或配置。	3	传输参数“数据”或“邮件”或OP超出配置区域范围。		<code>_XFR_EMAIL_FTPC</code>
16#04 _XFR	4	无法在S7级别设置连接。	建立连接的尝试重复。	
16#05 检查通讯对方的连接路径或通讯对方的	5	连接被终止或在未准备就绪的频道上尝试执行读取 / 写入作业。 配置。		<code>_XFR</code>
被更改，或尝试执行一项作业，但电子邮件服务未设置正确的参数。 检查CP 243-1 IT上的.edb文件是否具有独特性和完整性。		无法建立连接或连接被终止，或CP 243-1 IT上的电子邮件配置文件 检查通讯对方的连接路径。		<code>_EMAIL</code>
尝试执行一项作业，但FTP客户机服务未设置正确的参数。 IT上的.fdb文件是否具有独特性和完整性。		无法建立连接或连接被终止，或CP 243-1 IT上的FTP配置被更改，或 检查通讯对方的连接路径。检查CP 243-1 FTP连接被终止。		<code>_FTP</code>
检查FTP服务器配置的完整性。		检查通讯对方的连接路径。		<code>_FTP</code>
未设置正确的参数。		无法建立连接或连接被终止，或尝试执行一项作业，但HTTP服务器 检查HTTP服务器配置的完整性。		
16#06 _XFR_EMAIL_FTPC	6	应答包包含逻辑错误。	可触发一项新读取 / 写入作业。	检查配置。
16#07 应当检查读取作业的参数。	7	读取作业失败。	可触发一项新读取 / 写入作业。	
16#08 应当检查写入作业的参数。	8	写入作业失败。	可触发一项新读取 / 写入作业。	
16#09 _XFR_EMAIL_FTPC	9	频道未配置。	可触发带其他参数的新读取 / 写入作业。	
16#0A 可触发带其他参数的新读取 / 写入作业。	10	频道被配置为“服务器”，但却尝试触发读取 / 写入作业。		<code>_XFR</code>
16#0B 评估以前的读取 / 写入作业的“完成”	11	以前的读取 / 写入作业尚未完成。 返回参数。	可触发一项新读取 / 写入作业。	<code>_XFR_EMAIL_FTPC</code>
16#0C _XFR_EMAIL_FTPC	12	无效作业识别符	可触发一项新读取 / 写入作业。	
16#0D _XFR_CFG_EMAIL	13	由于用户程序触发了一个新配置，所有数据传输被终止。	重新启动系统	
16#0E 检查配置，尤其是.fdb文件的配置。	14	无法从CPU读取需要传输的数据。		<code>_FTP</code>
16#0F 检查配置，尤其是.fdb文件的配置。	15	无法将需要传输的数据写入CPU。		<code>_FTP</code>
16#10 检查与已配置电子邮件服务器的连接路	16	无法与一个已配置电子邮件服务器连接。 径。检查配置，尤其是.edb文件的配置。		<code>_EMAIL</code>
16#11 正确。检查配置，尤其是.edb文件的配置。	17	注册被电子邮件服务器拒绝。	检查并确定电子邮件服务器的配置是否	<code>_EMAIL</code>
16#12 检查配置，尤其是.edb文件和电子邮件	18	生成需要发送的电子邮件时出错。 中使用的占位符的配置。		<code>_EMAIL</code>
16#13 中使用的占位符的配置。	19	发现电子邮件配置不正确。	检查配置，尤其是.edb文件和电子邮件	<code>_EMAIL</code>

16#14	20	生成的电子邮件文字过长。	检查配置，尤其是.edb文件和电子邮件
中使用的占位符的配置。		_EMAIL	
16#17	23	无法与已配置FTP服务器建立连接。	检查与FTP服务器的通讯路径。
检查配置，尤其是.fdb文件的配置。		_FTPC	
16#18	24	用FTP服务器注册失败。	检查并确定FTP服务器配置是否正确。
检查配置，尤其是.fdb文件的配置。		_FTPC	
16#19	25	FTP传输协议错误	检查并确定FTP服务器配置是否正确。
检查并确定是否总是能够存取该文件。		检查配置，尤其是.fdb文件的配置。	_FTPC
16#1A	26	传输文件时超出内存区范围或指定的文件长度与实际文件长度不符。	
检查并确定该文件是否有足够的内存空间。		检查配置，尤其是.fdb文件的配置。	_FTPC
16#1B	27	经由FTP传输数据时出错。	检查并确定FTP服务器配置是否正确。
检查并确定是否存在网络故障。		检查配置，尤其是.fdb文件的配置。	_FTPC
16#80	128	不存在外部24 V。	应用电压或等到准备就绪。
_EMAIL _FTPC			_XFR_CFG
16#81	129	该频道未准备就绪或发生故障。	等到准备就绪。
评估子程序ETHx_CTRL的返回参数。		_XFR_CFG_EMAIL_FTPC	
16#82	130	该频道繁忙或该指令无“起始”脉冲。	等到准备就绪。
激活“起始”输入。		_XFR_CFG_EMAIL_FTPC	
16#83	131	用一个非法频道号码启动一项作业。	可触发一项新读取 / 写入作业。
检查用户程序。		_XFR_EMAIL_FTPC	
16#84	132	用一个非法数据块号码启动一项作业。	可触发一项新读取 / 写入作业。
检查用户程序。		_XFR_EMAIL_FTPC	

## 10.14 ETHx\_FTPC指令的CP243-1 IT错误代码

双击在PLC>信息窗口底部显示的以太网模块，检视频道状态。  
请查阅CP243-1 IT技术参考资料中的编程细节：

SIMATIC NET  
CP243-1 IT通讯处理器  
工业以太网和信息技术

错误字节 十六进制	十进制	说明	反应 / 校正	晓岔?? (ETHx_)
16#01	1	底板总线超时。	自动重新启动	_XFR_CFG_EMAIL
_FTPC				
16#02	2	由于“异常中止”命令，数据存取被终止。		
可启动一项新读取 / 写入作业。			_XFR_EMAIL_FTPC	
16#03	3	传输参数“数据”或“邮件”或OP超出配置区域范围。		
检查用户程序或配置。			_XFR_EMAIL_FTPC	
16#04	4	无法在S7级别设置连接。	建立连接的尝试重复。	
_XFR				
16#05	5	连接被终止或在未准备就绪的频道上尝试执行读取 / 写入作业。		
检查通讯对方的连接路径或通讯对方的配置。			_XFR	
被更改，或尝试执行一项作业，但电子邮件服务未设置正确的参数。			无法建立连接或连接被终止，或CP243-1 IT上的电子邮件配置文件	检查通讯对方的连接路径。
检查CP243-1 IT上的.edb文件是否具有独特性和完整性。			_EMAIL	
尝试执行一项作业，但FTP客户机服务未设置正确的参数。			无法建立连接或连接被终止，或CP243-1 IT上的FTP配置被更改，或	检查通讯对方的连接路径。检查CP243-1
IT上的.fdb文件是否具有独特性和完整性。			_FTPC	
检查FTP服务器配置的完整性。			FTP连接被终止。	检查通讯对方的连接路径。
未设置正确的参数。			_FTPC	
			无法建立连接或连接被终止，或尝试执行一项作业，但HTTP服务器	检查通讯对方的连接路径。检查HTTP服务器配置的完整性。
16#06	6	应答包包含逻辑错误。		可触发一项新读取 / 写入作业。检查配置。
_XFR_EMAIL_FTPC				
16#07	7	读取作业失败。		可触发一项新读取 / 写入作业。
应当检查读取作业的参数。			_XFR	
16#08	8	写入作业失败。		可触发一项新读取 / 写入作业。
应当检查写入作业的参数。			_XFR	
16#09	9	频道未配置。		可触发带其他参数的新读取 / 写入作业。
_XFR_EMAIL_FTPC				
16#0A	10	频道被配置为“服务器”，但却尝试触发读取 / 写入作业。		
可触发带其他参数的新读取 / 写入作业。			_XFR	



16#0B	11	以前的读取 / 写入作业尚未完成。	可触发一项新读取 / 写入作业。
评估以前的读取 / 写入作业的“完成”		返回参数。	_XFR_EMAIL_FTFC
16#0C	12	无效作业识别符	可触发一项新读取 / 写入作业。
_XFR_EMAIL_FTFC			
16#0D	13	由于用户程序触发了一个新配置，所有数据传输被终止。	重新启动系统
		_XFR_CFG_EMAIL	
16#0E	14	无法从CPU读取需要传输的数据。	
检查配置，尤其是.fdb文件的配置。		_FTFC	
16#0F	15	无法将需要传输的数据写入CPU。	
检查配置，尤其是.fdb文件的配置。		_FTFC	
16#10	16	无法与一个已配置电子邮件服务器连接。	
检查与已配置电子邮件服务器的连接路径。		检查配置，尤其是.edb文件的配置。	_EMAIL
16#11	17	注册被电子邮件服务器拒绝。	检查并确定电子邮件服务器的配置是否正确。
检查配置，尤其是.edb文件的配置。		_EMAIL	
16#12	18	生成需要发送的电子邮件时出错。	
检查配置，尤其是.edb文件和电子邮件		中使用的占位符的配置。	_EMAIL
16#13	19	发现电子邮件配置不正确。	检查配置，尤其是.edb文件和电子邮件
中使用的占位符的配置。		_EMAIL	
16#14	20	生成的电子邮件文字过长。	检查配置，尤其是.edb文件和电子邮件
中使用的占位符的配置。		_EMAIL	
16#17	23	无法与已配置FTP服务器建立连接。	检查与FTP服务器的通讯路径。
检查配置，尤其是.fdb文件的配置。		_FTFC	
16#18	24	用FTP服务器注册失败。	检查并确定FTP服务器配置是否正确。
检查配置，尤其是.fdb文件的配置。		_FTFC	
16#19	25	FTP传输协议错误	检查并确定FTP服务器配置是否正确。
检查并确定是否总是能够存取该文件。		检查配置，尤其是.fdb文件的配置。	_FTFC
16#1A	26	传输文件时超出内存范围或指定的文件长度与实际文件长度不符。	
检查并确定该文件是否有足够的内存空		间。检查配置，尤其是.fdb文件的配置。	_FTFC
16#1B	27	经由FTP传输数据时出错。	检查并确定FTP服务器配置是否正确。
检查并确定是否存在网络故障。		检查配置，尤其是.fdb文件的配置。	_FTFC
16#80	128	不存在外部24 V。	应用电压或等到准备就绪。
_EMAIL_FTFC			_XFR_CFG
16#81	129	该频道未准备就绪或发生故障。	等到准备就绪。
评估子程序ETHx_CTRL的返回参数。		_XFR_CFG_EMAIL_FTFC	
16#82	130	该频道繁忙或该指令无“起始”脉冲。	等到准备就绪。
激活“起始”输入。		_XFR_CFG_EMAIL_FTFC	
16#83	131	用一个非法频道号码启动一项作业。	可触发一项新读取 / 写入作业。
检查用户程序。		_XFR_EMAIL_FTFC	
16#84	132	用一个非法数据块号码启动一项作业。	可触发一项新读取 / 写入作业。
检查用户程序。		_XFR_EMAIL_FTFC	

## 10.15 CP243-2 AS-i模块错误代码

错误代码	说明[[[ENDKEEPN]]]
0000	工作完成，无错误。
0081	AS-i从属装置地址不正确。
0082	AS-i从属装置未激活（不在LAS中）。
0083	AS接口中出错。
0084	AS-i主模块当前状态中不允许使用的命令。
0085	存在一个带有地址0的AS-i从属装置。
0086	AS-i从属装置有非法配置数据（I/O或ID代码）。
00A1	在AS接口中未找到带地址的AS-i从属装置。
00A2	存在一个带地址0的AS-i从属装置。
00A3	AS接口中已经存在一个带新地址的AS-i从属装置。
00A4	无法删除AS-i从属装置地址。
00A5	无法设置AS-i从属装置地址。
00A6	无法永久性存储AS-i从属装置地址。
00A7	读取扩展ID1代码错误。
00A8	目标地址不可行（例如，将B从属装置地址用于标准从属装置）。
00B1	根据配置文件7.4传输字符串时出现长度错误。
00B2	根据配置文件7.4传输字符串时出现协议错误。
00F8	未知工作号码或工作参数。
00F9	AS-i主模块检测到一则EEPROM错误。
0x01	AS-i主模块错误：？模块位于错误位置
其他模块错误	？模块未配置
0x02	库规格过大。

0x03  
0x05

AS-i主模块未准备就绪。  
AS-i主模块正忙于处理地址变更请求或命令代码。

获取更多资料 微信搜索 蓝领星球

获取更多资料 微信搜索蓝领星球

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part



XI

获取更多资料 微信搜索蓝领星球

## 11 库

### 11.1 一般库概述

共有两种不同类型的库：

- 西门子（Siemens）公司提供的库（USS协议和Modbus协议）
- 用户定义的库

西门子指令库用一张名为STEP 7-Micro/WIN Add-On: Instruction Library (STEP 7-Micro/WIN附件：指令库)的单独CD盘出售，订购编号为6ES7 830 2BC00 0YX0。您只要已购买且曾经安装第1.1版西门子指令库，之后安装的任何STEP 7-Micro/WIN V3.2x版或V4.0版升级会无额外收费地自动升级您的指令库（当指令库有所添加或修改时）。

**注释：**

- 21x序列PLC不支持库。
- 以前的库指令版本无法用于带有新库指令的相同项目。
- 一旦您升级至STEP 7-Micro/WIN 3.2版，以前安装的库指令不再在Micro/WIN指令树中出现，而会将新库指令载入指令树。
- 您必须具有管理员特权登录，才能安装指令库。
- 用户定义的库不得使用与STEP 7-Micro/WIN指令库光盘中的库相同的名称。

以下是有关两种西门子（Siemens）公司提供的库的简要说明：

#### USS协议

USS协议库支持MicroMaster 3（MM3）和Micromaster 4（MM4）驱动器。STEP 7-Micro/WIN指令库通过包括专门设计使用USS协议与驱动器通讯的预配置子例行程序和中断例行程序的方法，使控制MicroMasterC 调速器 蜗蜗格D 梢岳 USS指令控制实际驱动器，并读取 / 写入驱动器参数。

您可以在STEP 7-

Micro/WIN指令树的“库”文件夹中找到这些指令。当您选择一条USS指令时，会自动增加一个或多个相关子例行程序（USS1至USS7）。

#### Modbus协议

STEP 7-

Micro/WIN指令库通过包括专门设计用于Modbus通讯的预配置子例行程序和中断例行程序的方法，使与Modbus主设备通讯更加简便。您可以利用Modbus从属协议指令配置S7-200，将其用作Modbus RTU从属设备，与Modbus主设备通讯。

您可以在STEP 7-Micro/WIN指令树的“库”文件夹中找到这些指令。您可以利用这些新指令将S7-200用作Modbus从属设备。当您选择一条Modbus从属指令时，会在您的项目中自动增加一个或多个相关子例行程序。

**注释：**

欲了解有关“用户定义库”的详情，请参阅[用户定义库指导](#)。

**另请参阅：**

[用户定义库简介](#)

[USS协议简介](#)

[Modbus协议简介](#)

### 11.2 用户定义库概述

指令库的起点是STEP 7-

Micro/WIN项目。您可以使用任何包含子例行程序的现有项目生成指令库。但是，您会发现建立一个专用库源码更加简便。如果要用POU的方法将库集成到项目中，请参阅[在其他STEP 7-Micro/WIN项目中包括供使用的库](#)。

该标题讨论以下主题：

[全局内存用法](#)

[分配全局内存](#)

[指令和参数命名](#)

[将POU注释用于文档编制](#)

[调用库指令](#)

[调试库指令](#)

[分布指令库](#)

[改动包括在库中的POU](#)

[使用STEP 7-Micro/WIN中的版本检查功能](#)

**全局内存用法**

[返回顶端](#)

欲清除可能出现库和项目之间内存使用的冲突，指令库不应当使用全局 PLC 资源。可能无法完全排除指令库使用全局资源的状况

- 尽量用子例行程序参数代替全局内存  
使用子例行程序参数，尽量减少库对全局内存的依赖性。您可以指定库指令使用的内存。例如，您可以有一个计算数值的子例行程序，并将该输出存储在一个 V 内存位置。程序的其余部分则会读取该 V 内存位置，以便确定计算的结果。如果您
- 使用临时变量进行计算  
将临时变量用于计算和临时结果可尽量减少指令库对全局内存的使用。临时变量（在局部变量表中对子例行程序和中断例行程序进行定义）从局部内存分配，不与项目发生冲突。局部内存并非持久性内存，无法用于在指令调用之间保存数据。其使用期是说明局部内存的子例行程序或中断例行程序的执行阶段。
- 需要使用 V 内存时  
有时您必须在包括在指令库中的子例行程序和中断例行程序中使用全局 V 内存。此时，为在这些 POU 中使用的所有 V 内存

符号	地址	注释
1 Output1	VB0	
2 Output2	VB10	
3 Output3	VB20	

此处有三个符号，分别在 VB0、VB10 和 VB20 位置说明。库编译对此作出解释，说明库要求 21 个字节的 V 内存。但是，如果我们按照以下方式重新分配符号，符号则使用 V 连续字节的 V 内存。库编译对此作出解释，说明库仅要求三个字节的 V 内存。

符号	地址	注释
1 Output1	VB0	
2 Output2	VB1	
3 Output3	VB2	

**注释：**在重新分配符号地址时应当小心，以便统一内存用法。通常在地址之间存在一种内在的关系，使地址不会丢失。例如

- 内存初始化  
指令库不包括数据块部分。因此，如果库使用的 V 内存必须初始化，则应用代码完成。
- 使用累加器  
用累加器进行计算经常会减少特定解决方案所需的代码数量，因为累加器可以被确定为字节、字或双字。但是，累加器是全局资源，如果用于包括在库中的子例行程序或中断例行程序，您应当留心不要销毁累加器的目录。请遵循以下步骤，在库 POU 中安全地使用累加器：
  1. 在 POU 开始位置，使用移动指令将累加器当前目录复制至局部内存。这样可保存目录，以便以后恢复。
  2. POU 现在可按照需要的方式使用累加器，并根据需要重新分配数值。
  3. 在 POU 的结束位置，使用移动指令恢复累加器的目录，从局部内存将保存的数值复制回累加器。

#### 分配全局内存 [返回顶端](#)

如果库要求使用部分 V 内存，您为库可以使用的内存块指定一个起始地址。没有必要在调用其中一条库指令后立即为库分配内存

- 选择文件 > 库内存菜单选项。只有在项目使用指令库时才会启用该选项。



您在指导中建立的库（模拟缩放）不要求使用V内存，对话框会说明这一点。如果库要求使用V内存，您必须为库的使用指定一

### 指令和参数命名

[返回顶端](#)

当您设计成为库指令的子例行程序时，在为子例行程序和子例行程序参数命名时要格外小心。在为成为库中的指令的子例行程序

- 选择准确地描述指令目的的名称。请记住，使用库的人可能对每条指令的详细内容并不了解。请选择清楚地描述指令
- 避免使用类属名称。通过避免使用类属名称，即可尽量减少名称与用户已经使用的子例行程序名称重复的可能性。如
- 当心LAD和FBD方框标题的可视空间限制。在LAD和FBD编辑器中，对指令名称（指令名称作为方框指令标题显示）可以显示的字符数有限制。确切的可视字符数取决于用户选择的字体大小和单元格宽度。
- 考虑国际性。如果您打算在全球各地分发指令库，请考虑使用不同的语言容易识别的指令名称。避免使用方言、俚

很多此类相同的命名指南也同样适用于指令参数。您为局部变量表中的子例行程序定义指令参数名称。请记住，输入和输出参数名称可能是用户用于确定如何使用指令的唯一文档。定义的名称应有助于用户确定每个参数所代表的内容。

### 将POU注释用于文档编制

[返回顶端](#)

如果您选择用密码保护库指令，库用户则无法看到指令中的源码。但是，他们能够看到构成库的每个子例行程序和中断例行程序的POU指令。POU注释是记录每条指令功能的最佳位置。请考虑使用POU注释提供下列详细信息：

1. 该指令具体执行什么功能？POU注释是给出指令功能说明的最佳位置。
2. 如何调用指令？例如，如果指令仅限在首次扫描时调用，或者如果指令依赖于成功地完成另一条指令，应当使用POU
3. 指令使用何种全局内存？如果指令直接参考全局内存，POU内存应当提醒用户这一用法，以便用户对程序进行设计，
4. 指令如何指示错误？很多指令需要通过输出参数或全局内存位置指示错误条件。向用户提供可能产生的错误说明以及

### 调用库指令

[返回顶端](#)

一旦在项目中输入库指令，编辑指令在各方面均与编辑子例行程序调用相同。您完全按照指定子例行程序调用的方式指定指令参数和控制参数执行的逻辑。

### 调试库指令

[返回顶端](#)

在建立指令库时，最好建立一些使用新库的测试项目，在将库分发给其他用户之前测试库指令功能。您可能会发现原先没有意识到的指令和原来的源项目之间的依赖性。当您测试指令库时，尽量避免对别人使用库的方法作出任何假设。请记住，使用库的人不具备与您一样对库操作的详细知识。

### 分发指令库

[返回顶端](#)

库编译输出是一个磁盘文件（mwl文件），该文件包含STEP 7-

Micro/WIN识别和使用指令库所需的所有内容。如果您希望向其他 STEP 7-Micro/WIN用户分发指令库，您可以将该文件复制至一个软盘或将文件附在电子邮件上，为用户提供一份文件的拷贝。欲使用向您提供的指令库，将mwl文件复制至您的计算机的硬盘驱动器，供使用。

### 改动库中包括的POU

#### [返回顶端](#)

如果您希望对库中的代码作出改动，打开作为库来源的项目，编辑包括在库中的 POU。当您重新编译库时，所作的改动被包括在新的编译程序中。

当您改动库时，您应当更新版本。欲更新版本：

- 打开“项目编译器选项”对话框。
- 在“库属性”标记上，更新版本域。
- 单击“确定”。
- 重新编译库。

**注释：**如果您不改动库文件存储的位置或改动库文件名，库的新版本则会盖写现有版本，基本上清除现有的版本。当您改动库版本时，您应当改动库文件名，或改动库存储的目录。

### 使用STEP 7-Micro/WIN中的版本检查功能

#### [返回顶端](#)

当您分发库的更新版本供您的用户使用，请务必记住现有项目不会“自动”更新，使用新版本。

为了帮助用户在指令库的不同版本之间转换，STEP 7-Micro/WIN提供版本检查功能。

版本检查功能查看当前项目中使用的每条库指令。如果包含的库有一个不同的指令版本，则会采取以下两项措施之一：

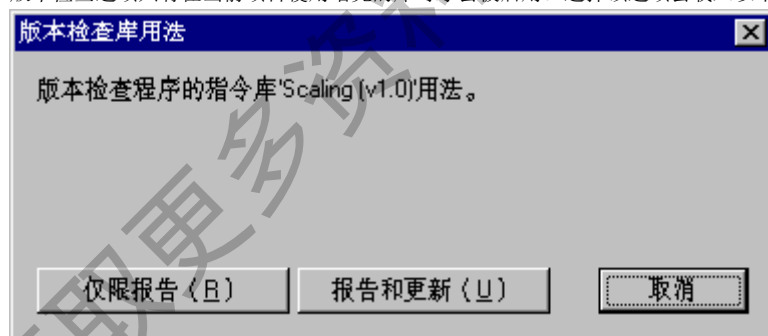
1. 如果指令的两个版本的签名相同，则构成该指令的POU会在项目中用来自库新版本的POU更新。出现此种情形时，无
2. 但是，如果指令的两个版本的签名不匹配，则不会进行自动更新。此时所涉及的POU将会从用户的项目中被删除。如

如果库的内部分配要求有改动，也会在用户项目中更新。

欲更新现有项目，使用库的最新版，（在包括新版库之后）打开项目，并有鼠标右键单击指令树中的库条目。上下文菜单会按以下方式显示：



版本检查选项只有在当前项目使用增亮的库时才会被启用。选择该选项会载入以下对话框：



从该对话框，您可以选择对当前项目执行版本更新或只需扫描当前项目，以便确定需要采取何种措施。

- **仅限报告：**“仅限报告”选项对程序进行扫描，以便确定如果您决定将项目更新至最新库版本应当作出何种改动。报
- **报告和更新：**“报告和更新”选项对程序进行扫描，以便确定需要作出何种改动以及如何执行这些改动。在执行更新

## 11.3 用户定义库辅导

以下指导将帮助您设计您的指令库。

欲遵循本指导的步骤：

- 从STEP 7-Micro/WIN电子手册光盘上的“提示与技巧”复制tip038.mwp。



- 将项目粘贴至您的计算机的硬盘驱动器上。
- 用STEP 7-Micro/WIN打开项目。

tip038.mwp建立三个模拟缩放例行程序，这三个例行程序在项目中作为子例行程序执行。每个子例行程序又执行一个不同类型的

- **Scale\_I\_to\_R**: 该子例行程序缩放从整数范围至浮点范围的模拟数值。初始（整数）值是向子例行程序的输入。缩放的
- **Scale\_R\_to\_I**: 该子例行程序缩放从浮点范围至整数范围的模拟数值。初始（浮点）值是向子例行程序的输入。缩放的
- **Scale\_R\_to\_R**: 该子例行程序缩放从浮点范围至另一个浮点范围的模拟数值。初始值（与输入范围一致）是向子例行程序的输入。子例行程序的输出是缩放至输出范围的模拟值。输入的高范围和低范围以及输出的高范围和低范围均为向子例行程序的输入。

该项目已经作为可下载S7-200项目执行。我们将使用该项目建立一个指令库，并将该新库包括在其他 STEP 7-Micro/WIN项目中使用。

### 任务1: 从项目建立指令库

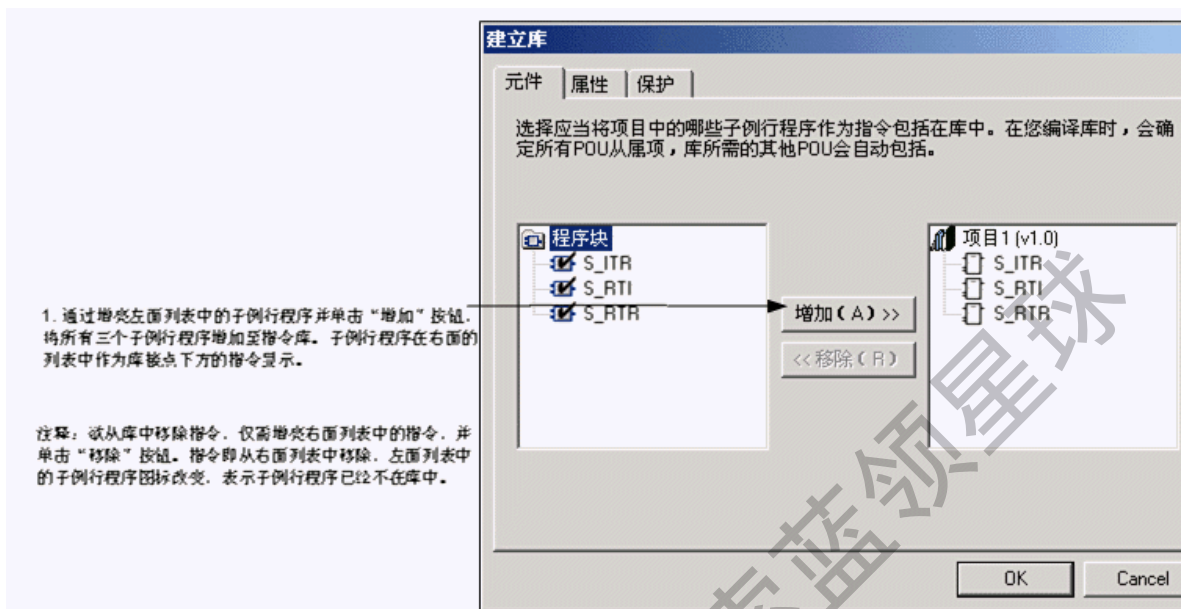
欲从项目建立指令库:

- 选择**文件 > 建立库**菜单选项。

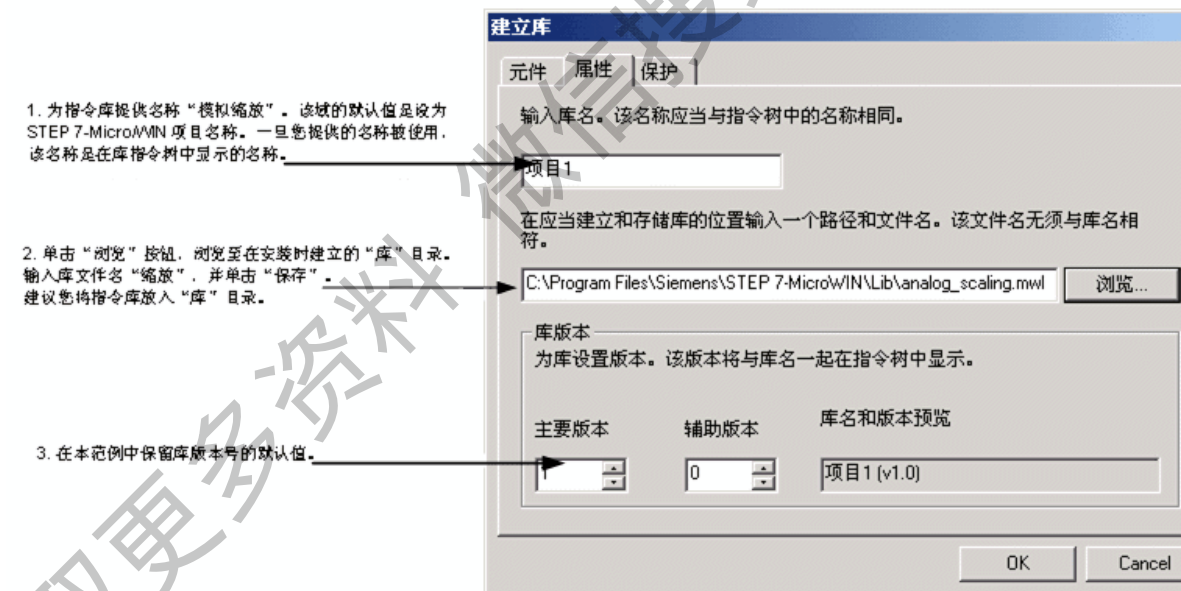


在每个标记上指定以下信息:

- 元件标记

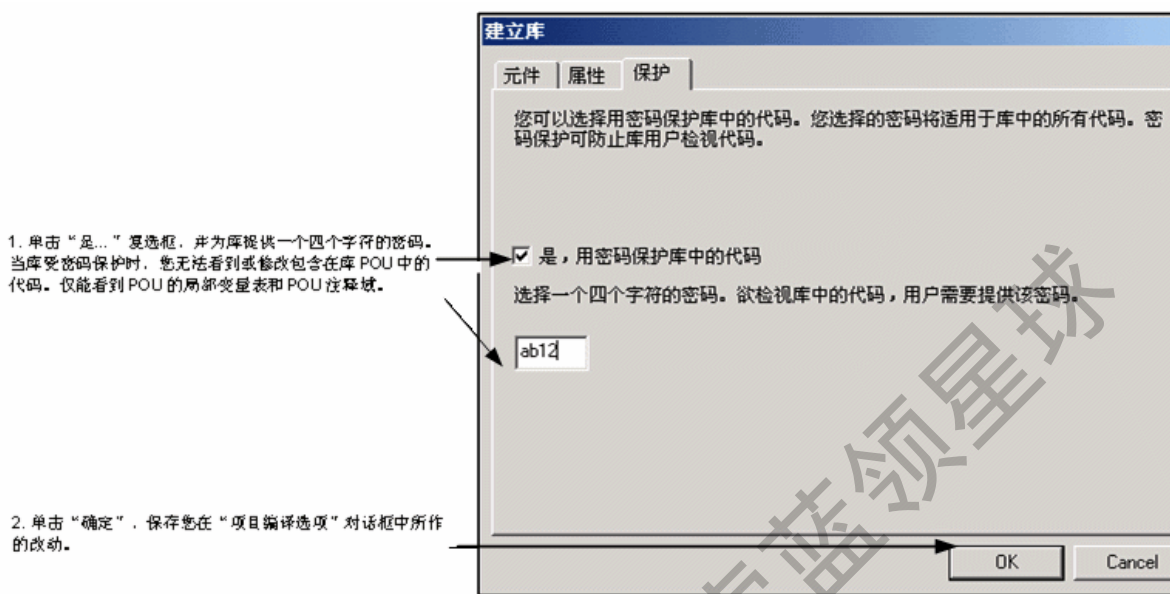


- 属性标记



有关命名的详情，请参阅[指令与参数命名](#)。

- 保护标记



有关密码保护的详情，请参阅[将POU注释用于文档编制](#)。

请单击“确定”，核实您的选择。

### 任务2：编译库

单击“确定”时，您定义的指令库会自动编译和写入硬盘驱动器。输出窗口中的讯息与正常项目编译不同。只有包括在库中的 POU 被编译。当库成功编译后，会建立库文件，并写入计算机的硬盘驱动器。

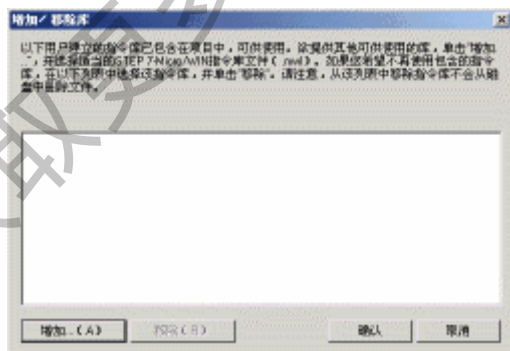
您一旦成功编译库，且库文件被写入磁盘，您就可以使用指令库。欲在另一个项目中使用指令库：

- 保存对项目所作的改动。
- 打开一个新项目。

### 任务3：在其他STEP 7-Micro/WIN项目中包括库

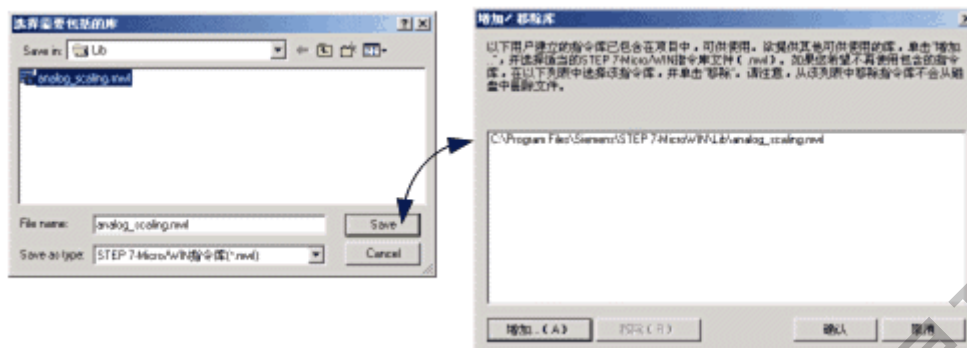
打开一个新项目后，在指令树中查找库中库节点。

- 用鼠标右键单击指令树中的“库”节点。
- 选择“增加 / 移除库”，浏览至您保存库文件的目录。

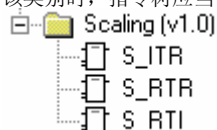


默认值为对话框从安装时建立的库目录开始。因为您已经在该目录中建立库，对话框应当包含库文件 Scaling.mwl。

- 单击“增加...”，将库增加至项目。
- 选择 Scaling.mwl，并单击“保存”。“选项”对话框应当如以下方式显示。单击“确定”。



一旦完成此一步骤，查看指令树中的“库”节点。您应当看到您建立的指令库已经作为一个类别增加在“库”的下方。当您扩充该类别时，指令树应当以下列方式显示：



请注意，库名是您在“建立库”对话框中输入的名称，库版本会自动附加在库名上。您选择的子例行程序在库中作为指令显示。

#### 任务4：调用库指令

一旦包括了库，您就可以使用从指令树拖放的方式或使用 F9 列表（在 LAD 和 FBD 中）放置库指令。从模拟缩放库调用 `Scale_I_to_R` 指令：

1. 在程序编辑器窗口打开所需的 POU，并滚动至您希望插入调用的网络。
2. 在指令树中，以双击的方式打开“库”文件夹。您可以从指令树拖动适当的库指令，并将指令放置在程序编辑器中的适当网络单元格中，或将光标放在程序编辑器的单元格上，然后双击指令树中的库指令。
3. 在程序中编辑调用指令参数，并为每个参数指定有效的操作数。

当您调用库指令时，可能会在项目中增加一个或多个 POU。请记住，库指令实际上是包含在库中的子例行程序。当您调用库指令时，执行该指令的子例行程序会自动从库复制至 STEP 7-Micro/WIN 项目。当您继续调用时，就不会再出现此种情形，因为所要求的子例行程序已经成为项目的一部分。您会注意到已经自动在项目中增加一个新 POU（`Scale_I_to_R`）。该 POU 在项目树中限制，在程序编辑器中有一个标记。您放置的指令现在成为对该子例行程序的调用。有关调用库指令的进一步详情，请参阅 [调用库指令](#)。

#### 任务5：改动库版本

当您对指令库作出改动时，您应当更新库版本，以便启用 STEP 7-Micro/WIN 中的版本检查功能，该功能可帮助您用最新改动更新现有项目。

欲改动库版本：

- 打开“项目编译选项”对话框
- 将“编译项目”作为一个 STEP 7-Micro/WIN 指令库选项选择
- 选择“库属性”标记
- 更新版本域
- 单击“确定”
- 重新编译库

出现在输出窗口中的编译讯息应当显示带有新版本的库名。

**注释：**当您改动库版本时，您应当改动库文件名，或改动库存储的目录。如果您不改动库文件存储的位置或改动库文件名，则会用库的新版本盖写现有版本，清除现有版本。有关改动版本的进一步详情，请参阅 [使用版本检查功能](#)。

另请参阅：  
[用户定义库简介](#)

## 11.4 USS 协议

### 11.4.1 使用 USS 协议库

本标题讨论以下主题：  
[USS 协议简介](#)  
[使用 USS 协议的要求](#)

[计算与驱动器通讯要求的时间](#)

## 编程细节:

- [USS\\_INIT](#)
- [USS\\_CTRL](#)
- [USS\\_RPM\\_X](#)
- [USS\\_WPM\\_x](#)
- [USS协议程序举例](#)
- [USS指令执行错误](#)

## USS协议简介

[返回顶端](#)

西门子指令库用一张名为STEP 7-Micro/WIN Add-On: Instruction Library (STEP 7-Micro/WIN附件: 指令库) 的单独CD盘出售, 订购编号为6ES7 830 2BC00 0YX0。您只要已购买且曾经安装第1.1版西门子指令库, 之后安装的任何STEP 7-Micro/WIN V3.2x版或V4.0版升级会无额外收费地自动升级您的指令库 (当指令库有所添加或修改时)。

STEP 7-Micro/WIN指令库通过包括专门设计用于USS协议与驱动器通讯的预配置子例行程序和中断例行程序使控制 MicroMaster 驱动器的作业更加简便。您可以使用USS指令控制实际驱动器和读取 / 写入驱动器参数。

您可以在STEP 7-

Micro/WIN指令树的“库”文件夹中找到这些指令。当您选择一条USS指令时, 会自动增加一条或多条相关子例行程序 (USS1至USS7)。

## 使用USS协议的要求

[返回顶端](#)

STEP 7-Micro/WIN指令库提供14个子例行程序、3个中断例行程序和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协议将端口0指定用于USS通讯。  
您使用USS\_INIT指令为端口0选择USS或PPI。(USS指用于SIMOTION MicroMaster驱动器的USS协议)。选择USS协议与驱动器通讯后, 您不得将端口0用于任何其他用途, 包括与STEP 7-Micro/WIN通讯。

在使用USS协议应用程序的程序开发过程中, 您应当在计算机中使用CPU 226、CPU 226XM或EM 277 PROFIBUS—与计算机中PROFIBUS CP连接的DP模块。这第二个通讯端口允许STEP 7-Micro/WIN在USS协议运行时监控应用程序。

- USS指令影响与端口0上自由端口通讯相关的所有SM位置。
- USS指令使用14个子例行程序和3个中断例行程序。
- USS指令可将程序要求的内存量最多增加至3050个字节。根据具体使用的USS指令, 这些指令的支持例行程序可使控制程序的内存操作至少增加2150个字节, 最多可达3050个字节。
- USS指令的变量要求一个400个字节V内存块。该内存块的起始地址由用户指定, 保留用于USS变量。
- 某些USS指令也要求有一个16个字节的通讯缓冲器。作为指令的参数, 您为该缓冲器在V内存中提供一个起始地址。建议为USS指令的每个实例指定一个独特的缓冲器。
- 执行计算时, USS指令使用累加器AC0至AC3。您还可以在程序中使用累加器; 但是, 累加器中的数值应由USS指令改
- USS指令无法用于中断例行程序。



**提示** 欲改动端口0操作, 使之返回PPI, 以便与STEP 7-Micro/WIN通讯, 使用另一条USS\_INIT指令, 重新分配端口0。您还可以将S7-200上的模式开关设为STOP (停止) 模式。这样会重置端口0的参数。请注意, 停止与驱动器的通讯同时也会使驱动器停止。

## 计算与驱动器通讯所需的时间

[返回顶端](#)

与驱动器通讯与S7-200扫描异步。S7-200通常已完成几次扫描, 才会完成一次驱动器通讯事项。以下因素帮助决定要求的时间量: 存在的驱动器数目、波特率和S7-200的扫描时间。

在使用参数存取指令时, 某些驱动器要求较长的延迟。参数存取所要求的时间量由驱动器的类型和存取的参数决定。

在USS\_INIT指令指定端口0使用USS协议后, S7-

200按下表显示的间隔时间定期轮询所有的现用驱动器。您必须为每台驱动器设置超时参数, 才能留出该轮询时间。

## 通讯时间

波特率	现用驱动器轮询之间的时间 (无现用参数存取指令)
1200	ms (最大) 乘以驱动器数目
2400	ms (最大) 乘以驱动器数目
4800	ms (最大) 乘以驱动器数目
9600	ms (最大) 乘以驱动器数目
19200	ms (最大) 乘以驱动器数目
38400	ms (最大) 乘以驱动器数目
57600	ms (最大) 乘以驱动器数目
115200	ms (最大) 乘以驱动器数目



**提示** 一次仅限将一条USS\_RPM\_x or USS\_WPM\_x指令设为现用。每条指令的“完成”输出应当在用户逻辑初始化一条新指令之前指示完成。每台驱动器仅限使用一条USS\_CTRL指令。

另请参阅：  
[USS协议程序举例](#)

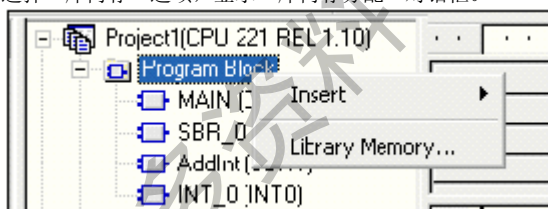
[USS\\_INIT](#)  
[USS\\_CTRL](#)  
[USS\\_RPM\\_X](#)  
[USS\\_WPM\\_x](#)

[USS指令执行错误](#)

## 11.4.2 使用USS协议指令

欲在S7-200控制器程序中使用USS协议指令，请遵循以下步骤：

- 在程序中插入USS\_INIT指令，并仅限在一次扫描中执行USS\_INIT指令。您可以使用USS\_INIT指令初始化或改动USS通讯参数。  
当您插入USS\_INIT指令时，几个隐藏子例行程序和中断例行程序会自动增加至程序中。
- 仅限为每台现用驱动器在程序中放置一条USS\_CTRL指令。  
您可以按照要求增加任何数目的USS\_RPM\_x和USS\_WPM\_x指令，但一次仅限其中一条指令设为现用。
- 用滑鼠右键单击(获取菜单)指令树中的“程序块”节点，为库指令分配V内存。  
选择“库内存”选项，显示“库内存分配”对话框。



- 配置驱动器参数，使之与程序中使用的波特率和地址相匹配。
- 连接S7-200和驱动器之间的通讯电缆。  
确保与驱动器连接的所有控制设备，例如S7-200，均用一根短粗的电缆与驱动器相同的接地线或汇接点连接。

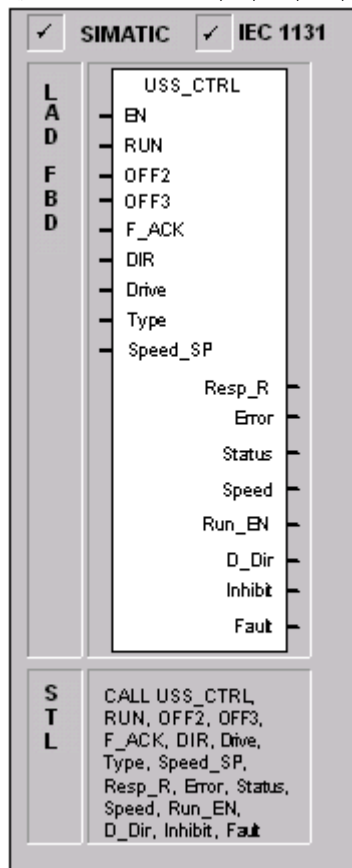


**当心** 具有不同参考电势的互连设备会引起不需要的电流流入互连电缆。这些不需要的电流会造成通讯错误或损坏设备。  
确保所有需要与通讯电缆连接的设备共享一个公用电路参考电势或相互独立，以防不需要的电流流入。屏蔽必须系在机壳接地味 9针接头的1号针上。建议您将MicroMaster驱动器上的2-0V终端系在机壳接地线上。

## 11.4.3 USS\_CTRL USS协议

输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
运行, OFF2, OFF3, F_ACK, DIR		I, Q, M, S, SM, T, C, V, L,
功率流	布尔[[[ENDKEEPN]]]	
Resp_R, Run_EN, D_Dir, 禁止, 故障		I, Q, M, S, SM, T, C, V, L
驱动器, 类型	VB, IB, QB, MB, SB, SMB, LB, AC, 常量, *VD, *AC, *LD	布尔[[[ENDKEEPN]]] 字节[[[ENDKEEPN]]]
错误	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	字节[[[ENDKEEPN]]]
状态	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	字[[[ENDKEEPN]]]

Speed\_SP            VD, ID, QD, MD, SD, SMD, LD, AC, 常量, \*VD, \*AC, \*LD,            实数[[[ENDKEEPN]]]  
速度                VD, ID, QD, MD, SD, SMD, LD, AC, \*VD, \*AC, \*LD            实数[[[ENDKEEPN]]]



**USS\_CTRL**指令被用于控制ACTIVE（现用）MicroMaster驱动器。USS\_CTRL指令将选择的命令放在通讯缓冲器中，然后送至编返那 颯 DRIVE（驱动器）参数，条件是已在USS\_INIT指令的ACTIVE（现用）参数中选择该驱动器。

仅限为每台驱动器指定一条USS\_CTRL指令。

某些驱动器仅将速度作为正值报告。如果速度为负值，驱动器将速度作为正值报告，但逆转D\_Dir（方向）位。

EN位必须打开，才能启用USS\_CTRL指令。该指令应当始终启用。

RUN（运行）（RUN/STOP（运行/停止））表示驱动器是打开（1）还是关闭（0）。当RUN（运行）位打开时，MicroMaster驱动器收到一条命令，按指定的速度和方向开始运行。为了使驱动器运行，必须符合以下条件：

- DRIVE（驱动器）在USS\_INIT中必须被选为ACTIVE（现用）。
- OFF2和OFF3必须被设为0。
- FAULT（故障）和INHIBIT（禁止）必须为0。

当RUN（运行）关闭时，会向MicroMaster驱动器发出一条命令，将速度降低，直至马达停止。OFF2位被用于允许MicroMaster驱动器兄镍V埂 OFF3位被用于命令MicroMaster驱动器迅速停止。

Resp\_R（收到应答）位确认从驱动器收到应答。对所有的现用驱动器进行轮询，查找最新驱动器状态信息。每次S7-200从驱动器收到应答时，Resp\_R位均会打开，进行一次扫描，所有以下数值均被更新。

F\_ACK（故障确认）位被用于确认驱动器中的故障。当F\_ACK从0转为1时，驱动器清除故障。

DIR（方向）位表示驱动器应当移动的方向。

“驱动器”（驱动器地址）输入是MicroMaster驱动器的地址，向该地址发送USS\_CTRL命令。有效地址：0至31。“类型”（驱动器类型）输入是MicroMaster驱动器的类型。将MicroMaster 3（或更早版本）驱动器的类型设为0。将MicroMaster 4驱动器的类型设为1。

Speed\_SP（速度定点）是作为全速百分比的驱动器速度。Speed\_SP的负值会使驱动器逆转旋转方向。范围：-200.0%至200.0%。

“错误”是一个包含对驱动器最新通讯请求结果的错误字节。USS指令执行错误标题定义可能因执行指令而导致的错误条件。

“状态”是驱动器返回的状态字原始数值。下图显示“标准状态字”和“主反馈”的状态位。

“速度”是作为全速百分比的驱动器速度。范围：-200.0%至200.0%。

Run\_EN（运行启用）表示驱动器是运行（1）还是停止（0）。

D\_Dir表示驱动器的旋转方向。

“禁止”表示驱动器上的禁止位状态（0 - 不禁止，1 -

禁止）。欲清除禁止位，“故障”位必须关闭，RUN（运行）、OFF2和OFF3输入也必须关闭。

“故障”表示故障位状态（0 - 无故障，1 -

故障）。驱动器显示故障代码。（有关您的驱动器，请参阅用户手册）。欲清除故障位，纠正引起故障的原因，并打开F\_ACK位。

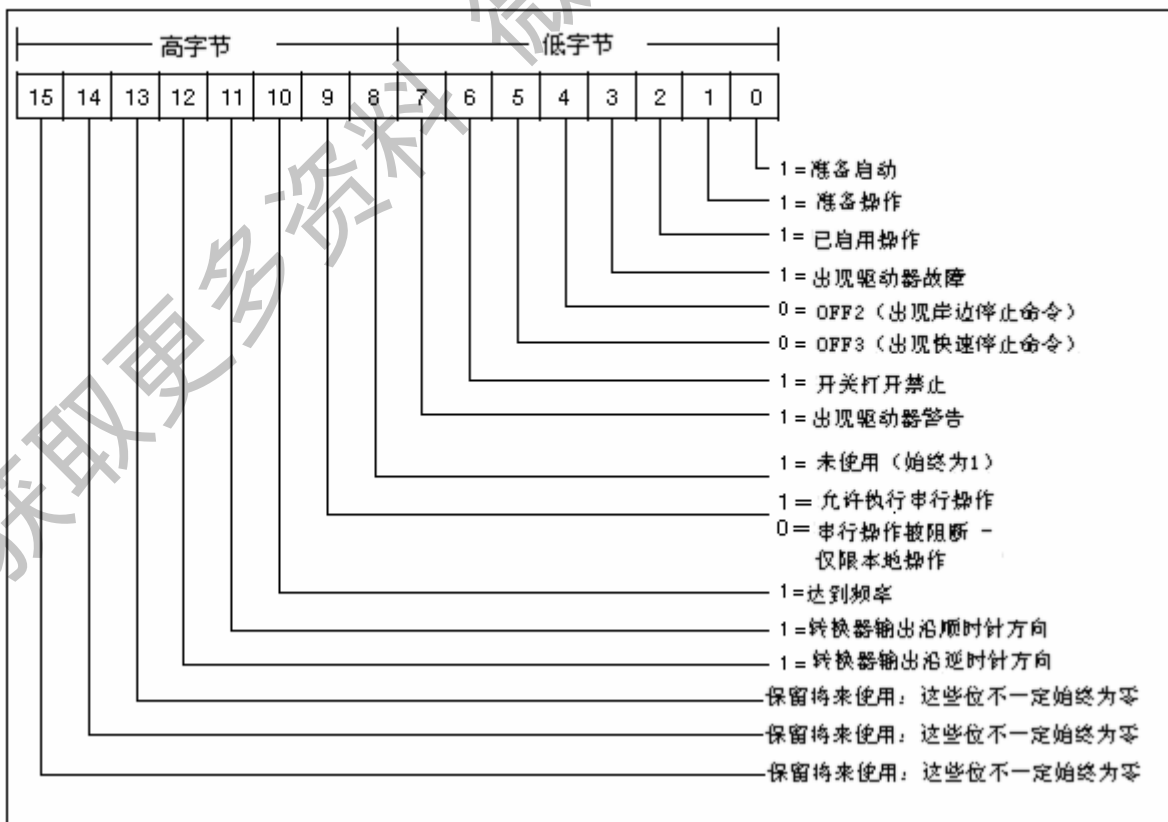
**MicroMaster 3和主反馈的标准状态字状态位**



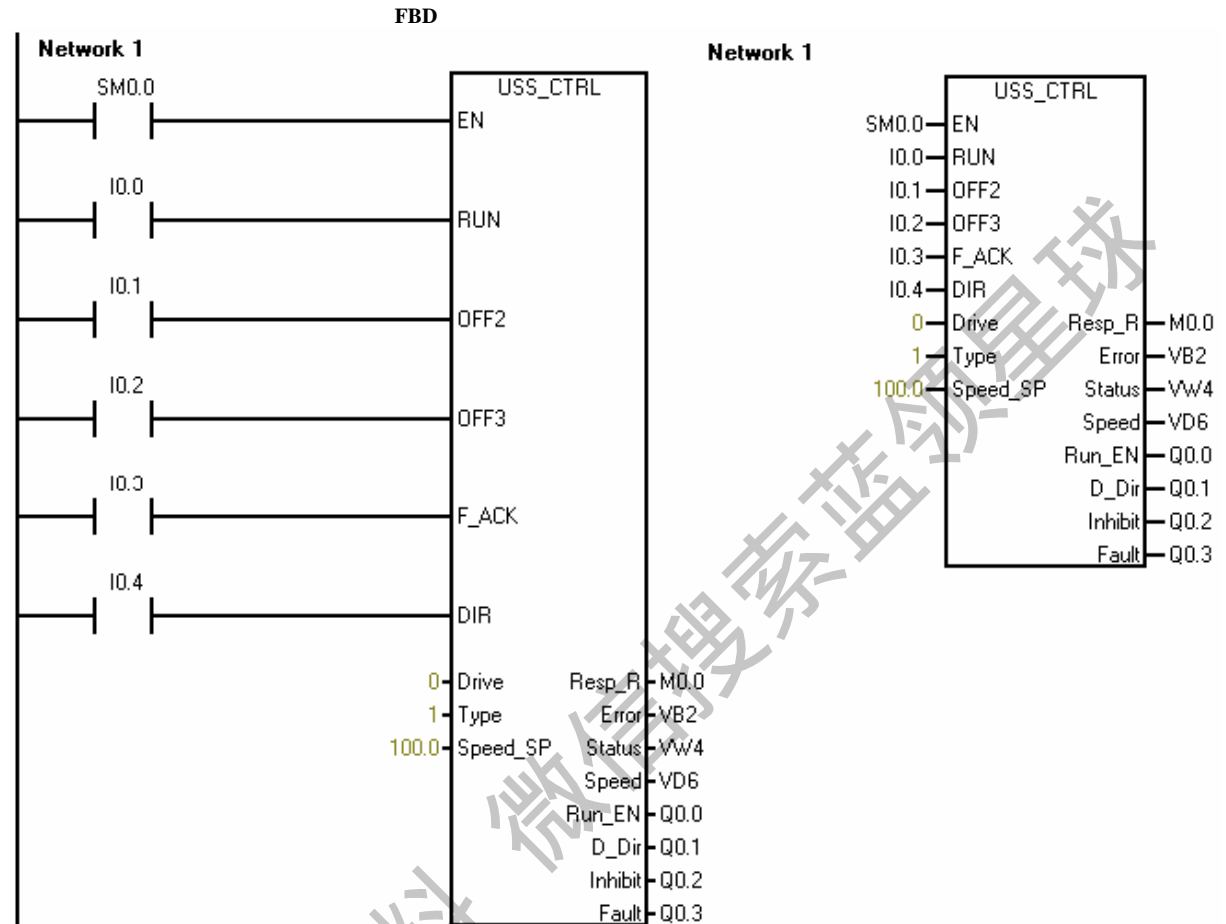
保留将来使用：这些位不一定始终为零

MicroMaster 4和主反馈的标准状态字状态位

保留将来使用：这些位不一定始终为零





USS\_CTRL 举例  
LAD

## STL

```

NETWORK 1 // MM4驱动器0的控制方框
LD SMO.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

```

## 仅在STL中显示

```

NETWORK 1 // MM4驱动器0的控制方框
LD SMO.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

```

## 另请参阅:

[USS协议程序举例](#)  
[使用USS协议指令](#)

[USS\\_INIT](#)

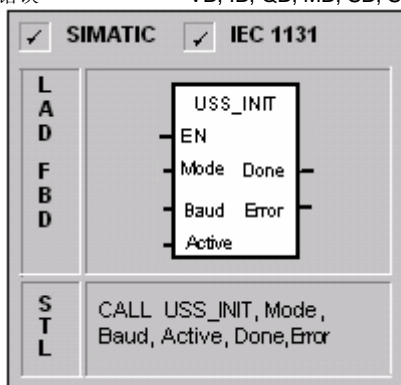
[USS\\_RPM\\_X](#)

[USS\\_WPM\\_x](#)

USS指令执行错误

## 11.4.4 USS\_INIT USS协议

输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
模式	VB, IB, QB, MB, SB, SMB, LB, AC, 常量, *VD, *AC, *LD	字节[[[ENDKEEPN]]]
波特率、现用	VD, ID, QD, MD, SD, SMD, LD, AC, 常量, *VD, *AC, *LD	双字[[[ENDKEEPN]]]
完成	I, Q, M, S, SM, T, C, V, L	布尔[[[ENDKEEPN]]]
错误	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	字节[[[ENDKEEPN]]]



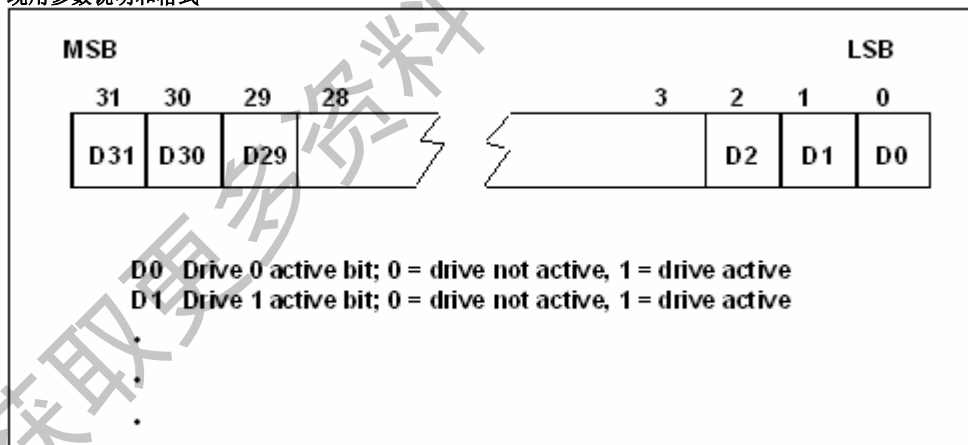
USS\_INIT指令被用于启用和初始化或禁用MicroMaster驱动器通讯。在使用任何其他USS协议指令之前，必须执行USS\_INIT指令。EN输入打开时，在每次扫描时执行该指令。

仅限为通讯状态的每次改动执行一次USS\_INIT指令。使用边缘检测指令，以脉冲方式打开EN输入。欲改动初始化参数，执行一跌USS\_INIT指令。

USS输入数值选择通讯协议：输入值1将端口0分配给USS协议，并启用该协议；输入值0将端口0分配给PPI，并禁用USS协议。BAUD（波特率）将波特率设为1200、2400、4800、9600、19200、38400、57600或115200。

ACTIVE（现用）表示现用驱动器。某些驱动器仅支持地址0至30。请参阅以下图解中的现用驱动器输入说明和格式。会在背景中对任何带有“现用”标记的驱动器自动进行轮询，以便控制驱动器、收集状态和预防驱动器中的串行链接超时。可使用驱动 topic中的驱动器扫描时间表计算驱动器状态轮询之间的时间。请参阅“驱动器设置”标题中有关如何配置串行链接超时参数（P093）的信息。

现用参数说明和格式



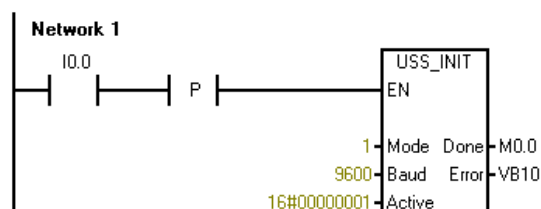
当USS\_INIT指令完成时，DONE（完成）输出打开。“错误”输出字节包含执行指令的结果。USS指令执行错误标题定义可能因葱兄噶碗贾碌拇碧笔登

下表显示USS\_INIT子例行程序的操作数和数据类型。

USS\_INIT举例

LAD

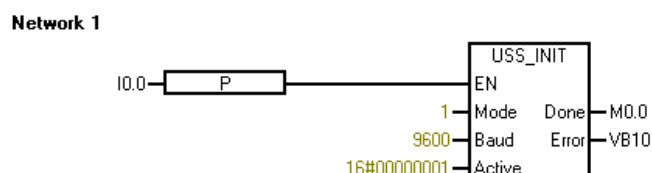
FBD



```

STL
LD I0.0
EU
CALL USS_INIT, 1, 9600, 16#00000001, M0.0, VB10

```



另请参阅:

[USS协议程序举例](#)  
[使用USS协议指令](#)

[USS\\_CTRL](#)

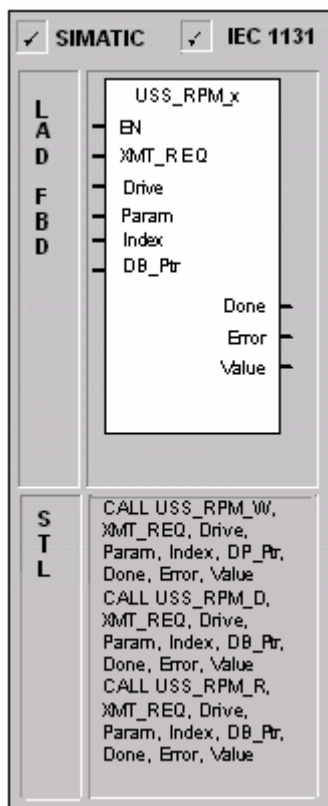
[USS\\_RPM\\_x](#)

[USS\\_WPM\\_x](#)

[USS指令执行错误](#)

#### 11.4.5 USS\_RPM\_x USS协议

输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
XMT_REQ	I, Q, M, S, SM, T, C, V, L, 功率流受上升边缘检测元素条件限制。	布尔[[[ENDKEEPN]]]
驱动器	VB, IB, QB, MB, SB, SMB, LB, AC, 常量, *VD, *AC, *LD	字节[[[ENDKEEPN]]]
参数, 索引	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, 常量, *VD, *AC, *LD	字[[[ENDKEEPN]]]
DB_Ptr	&VB	双字[[[ENDKEEPN]]]
完成	I, Q, M, S, SM, T, C, V, L	布尔[[[ENDKEEPN]]]
错误	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	字节[[[ENDKEEPN]]]
数值	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AQW, *VD, *AC, *LD	VD, ID, QD, MD, SD, SMD, LD, *VD, 字 双字, 实数[[[ENDKEEPN]]]



USS协议共有三条读取指令：

- USS\_RPM\_W指令读取不带符号的字参数。
- USS\_RPM\_R指令读取浮点参数。
- USS\_RPM\_D指令读取不带符号的双字参数。

一次仅限将一条读取（USS\_RPM\_x）或写入（USS\_WPM\_x）指令设为现用。

当MicroMaster驱动器确认收到命令或发送一则错误条件时，USS\_RPM\_x事项完成。当该进程等待应答时，逻辑扫描继续执行。EN位必须打开，才能启用请求传输，并应当保持打开，直至设置“完成”位，表示进程完成。例如，当XMT\_REQ输入打开，在每次扫描时向MicroMaster传输一条USS\_RPM\_x请求。因此，XMT\_REQ输入应当通过一个边缘检测元素以脉冲方式打开，该检测元素为EN输入的每次正转移传输一则请求。

“驱动器”输入是MicroMaster驱动器的地址，USS\_RPM\_x指令被发送至该地址。单台驱动器的有效地址是0至31。

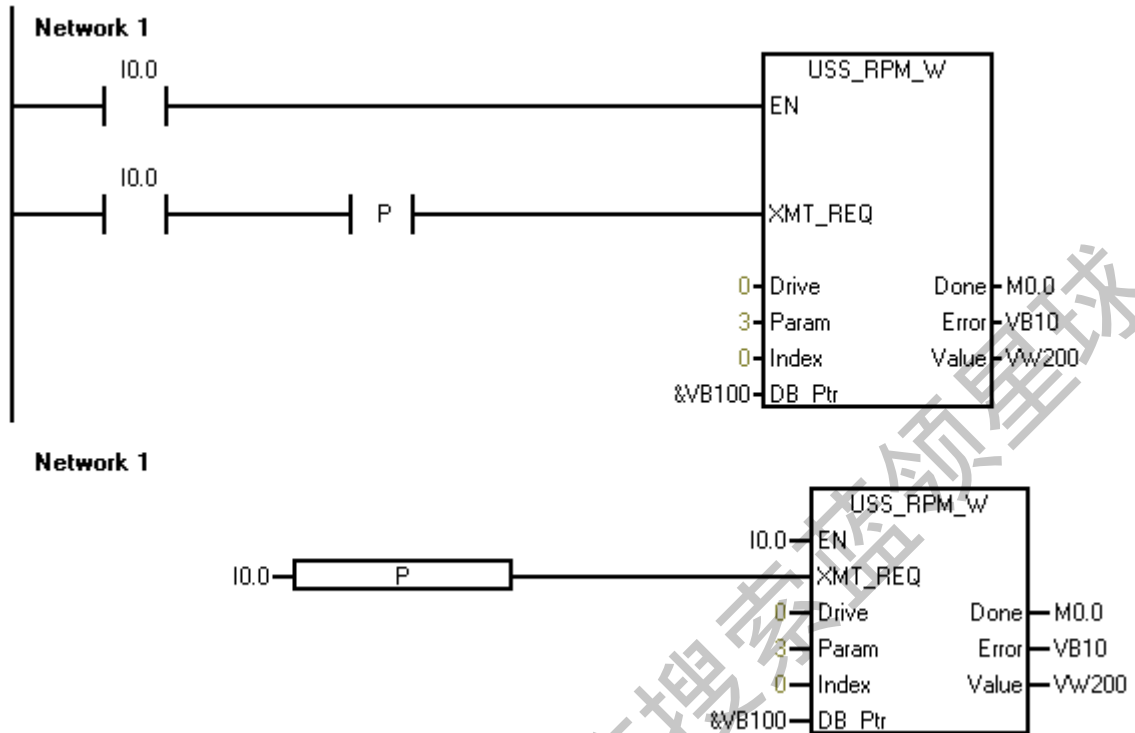
“参数”是参数号码。“索引”是需要读取参数的索引值。“数值”是返回的参数值。必须向DB\_Ptr输入提供16个字节的缓冲器地址。该缓冲器被USS\_RPM\_x指令用于存储向MicroMaster驱动器发出的命令结果。

当USS\_RPM\_x指令完成时，“完成”输出被打开，“错误”输出字节和“数值”输出包含执行指令的结果。[USS指令执行错误标](#)

#### USS\_RPM\_x举例

##### LAD和FBD

注释：两个触点必须具有相同的地址。



```

STL
NETWORK 1
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

```

// 两个触点必须具有相同的地址  
// 与下一个触点相同的地址  
// 与前一个触点相同的地址

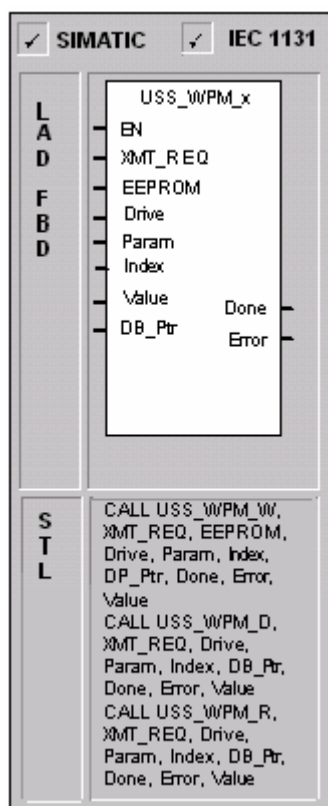
另请参阅：  
[USS协议程序举例](#)  
[使用USS协议指令](#)

[USS\\_INIT](#)  
[USS\\_CTRL](#)  
[USS\\_WPM\\_x](#)

[USS指令执行错误](#)

## 11.4.6 USS\_WPM\_x USS协议

输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
XMT_REQ	I, Q, M, S, SM, T, C, V, L, 功率流受上升边缘检测元素条件限制。	布尔[[[ENDKEEPN]]]
EEPROM	I, Q, M, S, SM, T, C, V, L, 功率流	[[[ENDKEEPN]]]
驱动器	VB, IB, QB, MB, SB, SMB, LB, AC, 常量, *VD, *AC, *LD	字节[[[ENDKEEPN]]]
参数, 索引	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, 常量, AC *VD, *AC, *LD	字[[[ENDKEEPN]]]
数值	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AQW, *VD, *AC, *LD VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD	字 双字, 实数[[[ENDKEEPN]]]
DB_PTR	&VB	双字[[[ENDKEEPN]]]
完成	I, Q, M, S, SM, T, C, V, L	布尔[[[ENDKEEPN]]]
错误	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	字节[[[ENDKEEPN]]]



USS协议共有三种写入指令：

- USS\_WPM\_W指令写入不带符号的字参数。
- USS\_WPM\_R指令写入浮点参数。
- USS\_WPM\_D指令写入不带符号的双字参数。

一次仅限将一条读取（USS\_RPM\_x）或写入（USS\_WPM\_x）指令设为现用。

当MicroMaster驱动器确认收到命令或发送一则错误条件时，USS\_WPM\_x事项完成。当该进程等待应答时，逻辑扫描继续执行。EN位必须打开，才能启用请求传输，并应当保持打开，直至设置“完成”位，表示进程完成。例如，当XMT\_REQ输入打开，在每次扫描时向MicroMaster传输一条USS\_WPM\_x请求。因此，XMT\_REQ输入应当通过一个边缘检测元素以脉冲方式打开，该检测

当驱动器打开时，EEPROM输入启用对驱动器的RAM和EEPROM的写入，当驱动器关闭时，仅启用对RAM的写入。请注意，该

“驱动器”输入是MicroMaster驱动器的地址，USS\_RPM\_x指令被发送至该地址。单台驱动器的有效地址是0至31。

“参数”是参数号码。“索引”是需要读取参数的索引值。“数值”是需要写入驱动器RAM中的参数值。对于MicroMaster 3驱动器，您还可以将该数值写入驱动器的EEPROM，根据您配置P971（EEPROM存储控制）的方式而定。

必须向DB\_Ptr输入提供16个字节的缓冲器地址。该缓冲器被USS\_WPM\_x指令用于存储向MicroMaster驱动器发出的命令结果。

当USS\_WPM\_x指令完成时，“完成”输出被打开，“错误”输出字节包含执行指令的结果。USS指令执行错误标题定义可能因执行指令导致的错误条件。



#### 当心

当您使用USS\_WPM\_x指令更新存储在驱动器EEPROM中的参数集时，您必须确保未超出向EEPROM写入周期（约50,000）

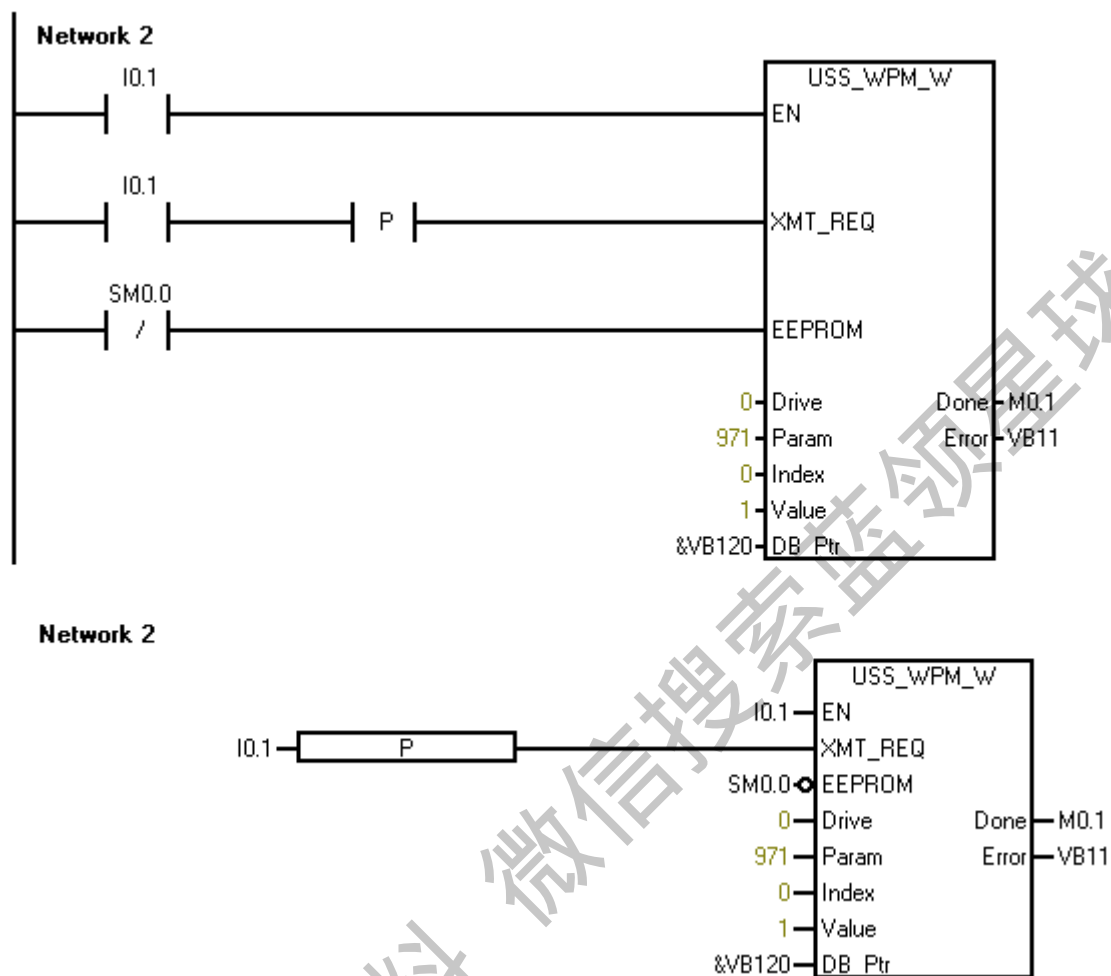
如果要求向驱动器参数频繁写入，如果是MM3驱动器，您应当首先将驱动器中的EEPROM存储控制参数设为零，如果是MM4C

EEPROM输入。

#### USS\_WPM\_x举例

##### LAD和FBD

注释：两个触点必须具有相同的地址。



```

STL
NETWORK 2
LD      I0.1           // 两个触点必须具有相同的地址
=       L60.0          // 与下一个触点相同的地址
LD      I0.1           // 与前一个触点相同的地址
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

```

另请参阅：  
[USS协议程序举例](#)  
[使用USS协议指令](#)

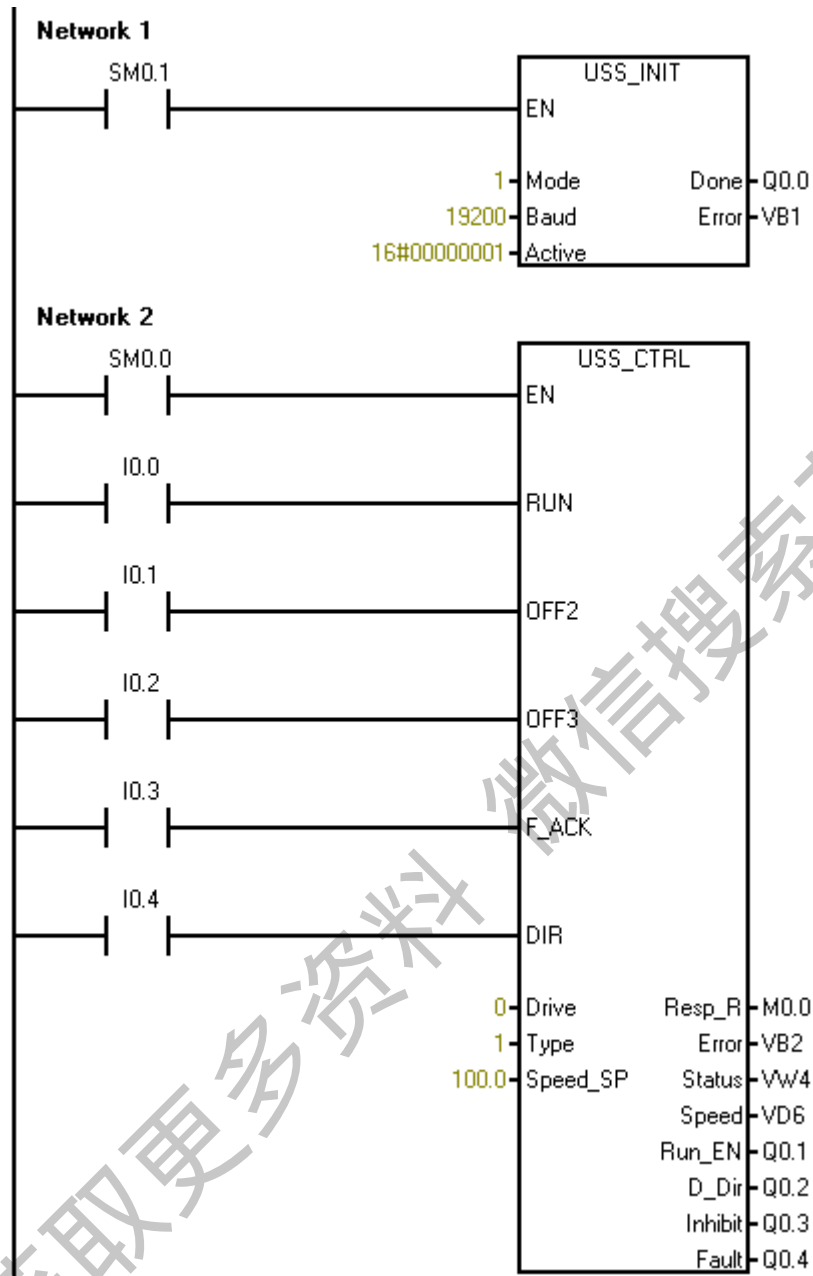
[USS\\_INIT](#)  
[USS\\_RPM\\_X](#)  
[USS\\_CTRL](#)

[USS指令执行错误](#)

## 11.4.7 USS协议程序举例

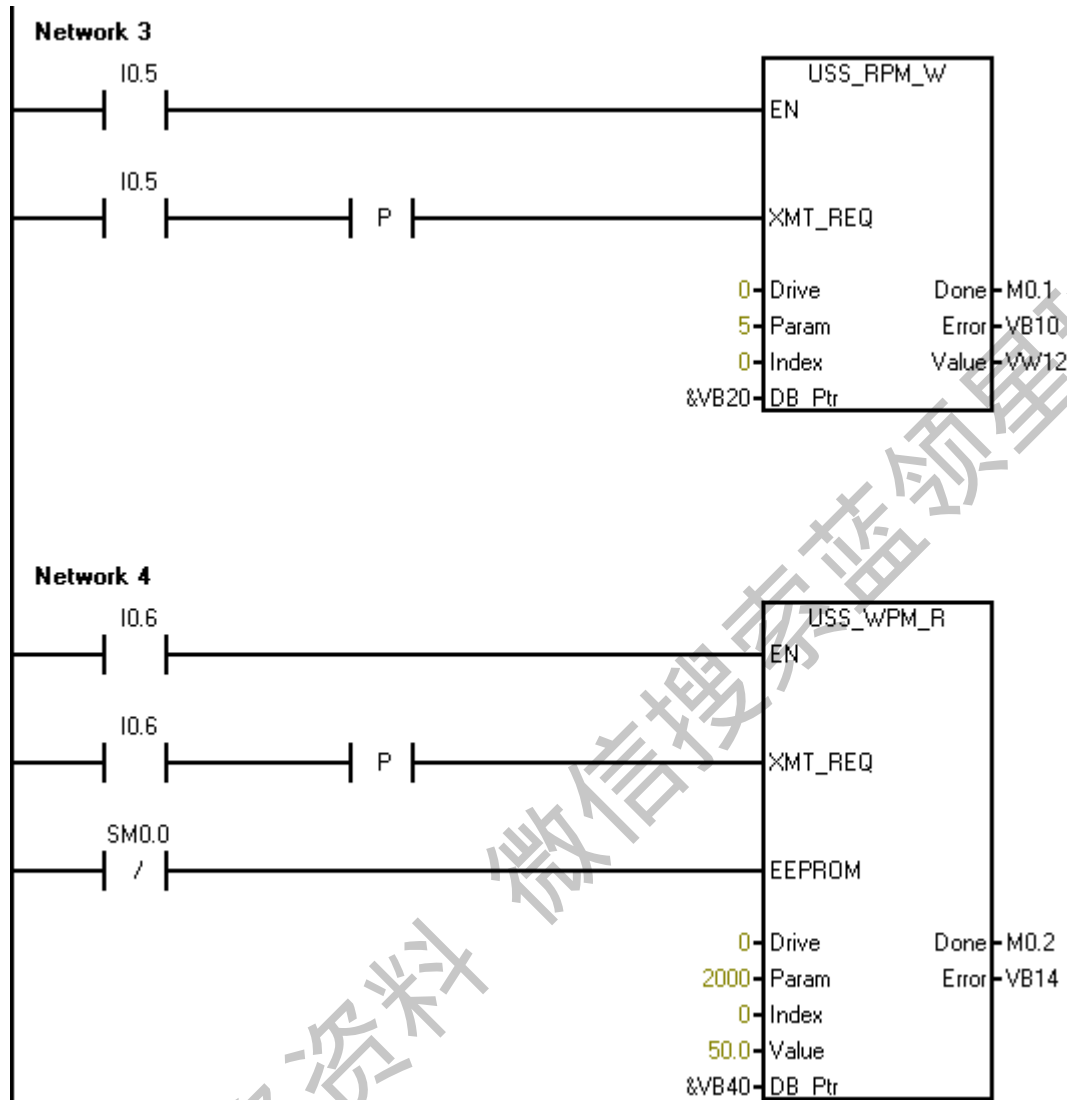
[LAD举例](#)  
[FBD举例](#)  
[STL举例](#)

LAD举例  
[返回顶端](#)



获取更多资料 微信搜索 蓝领星球

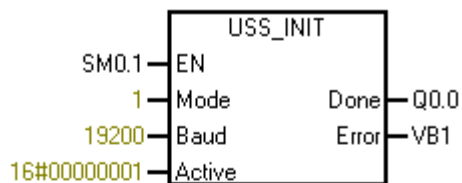




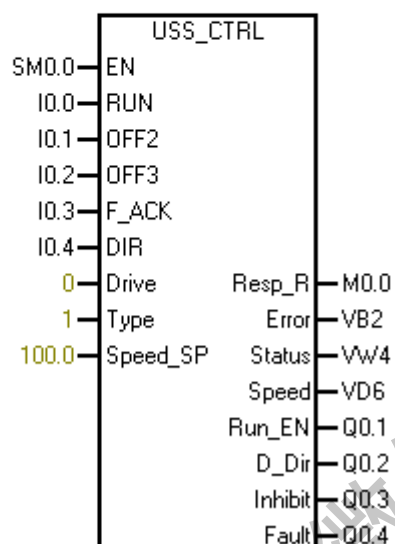
FBD举例  
[返回顶端](#)

获取更多资料 微信搜索 蓝领星球

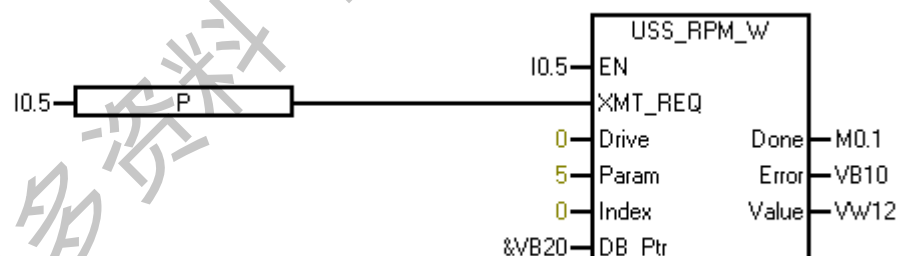
## Network 1



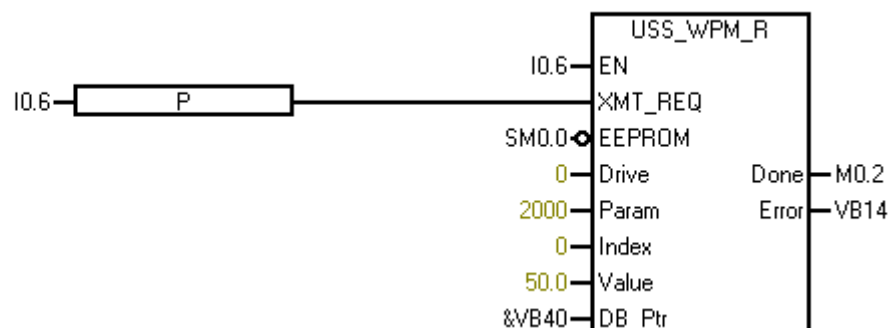
## Network 2



## Network 3



## Network 4



STL举例

**返回顶端**

```

NETWORK 1          // 初始化USS写移：在首次扫描时，在19200位置
                   // 为端口0启用USS协议，将驱动器地址“0”设为现用
LD      SMO.1
CALL   USS_INIT, 1, 19200, 16#00000001, Q0.0, VB1

NETWORK 2          // 为驱动器0控制参数
LD      SMO.0
CALL   USS_CTRL, I0.0, I0.1, I0.2, I0.3, I0.4, 0, 1, 100.0, MO.0, VB2, VW4, VD6, Q0.1, Q0.2,
Q0.3, Q0.4

NETWORK 3          // 从驱动器0读取字参数
                   // 读取参数5索引0
LD      I0.5
=       L60.0      // 将I0.5状态保存至一个临时
LD      I0.5      // L位置，以便本网络在LAD中显示。
EU
=       L63.7      // 至一个临时L位置，以便传送至子例行程序。
LD      L60.0
CALL   USS_RPM_W, L63.7, 0, 5, 0, &VB20, MO.1, VB10, VW12
NETWORK 4          // 在驱动器0中写入一个字参数
                   // 写入参数2000索引0
LD      I0.6
=       L60.0
LD      I0.6
EU
=       L63.7
LDN    SMO.0
=       L63.6
LD      L60.0
CALL   USS_WPM_R, L63.7, L63.6, 0, 2000, 0, 50.0, &VB40, MO.2, VB14

```

**另请参阅：**[使用USS协议指令](#)[USS\\_INIT](#)[USS\\_CTRL](#)[USS\\_RPM\\_X](#)[USS\\_WPM\\_x](#)[USS指令执行错误](#)**11.4.8 USS指令执行错误**

错误代码	说明
0	无错。
1	驱动器不应答。
2	检测到来自驱动器的应答中检验和错误。
3	检测到来自驱动器的应答中奇偶校验错误。
4	来自用户程序的干扰造成错误。
5	尝试非法命令。
6	提供非法驱动器地址。
7	未为USS协议设置通讯端口。
8	通讯端口正在忙于处理指令。
9	驱动器速度输入超出范围。
10	驱动器应答长度不正确。
11	驱动器应答第一个字符不正确。
12	驱动器应答中的字符长度不受USS指令支持。
13	错误的驱动器应答。
14	提供的DB_Ptr地址不正确。
15	提供的参数号不正确。
16	选择了无效协议。
17	USS现用，不允许改动。
18	指定了非法波特率。
19	无通讯：驱动器未设为现用。
20	驱动器应答中的参数或数值不正确或包含错误代码。
21	返回一个双数值，而不是请求的字数值。
22	返回一个字数值，而不是请求的双数值。

**另请参阅：**[USS协议程序举例](#)

[使用USS协议指令](#)

[USS\\_INIT](#)

[USS\\_CTRL](#)

[USS\\_RPM\\_X](#)

[USS\\_WPM\\_x](#)

## 11.5 Modbus协议库

### 11.5.1 使用Modbus协议库

西门子指令库用一张名为STEP 7-Micro/WIN Add-On: Instruction Library (STEP 7-Micro/WIN附件: 指令库) 的单独CD盘出售, 订购编号为6ES7 830 2BC00 0YX0。您只要已购买且曾经安装第1.1版西门子指令库, 之后安装的任何STEP 7-Micro/WIN V3.2x版或V4.0版升级会无额外收费地自动升级您的指令库 (当指令库有所添加或修改时)。

STEP 7-

Micro/WIN指令库包含为Modbus通讯专门设计预先配置的子例行程序和中断例行程序, 使与Modbus的通讯更加简便。有了Modbus从属协议指令, 您可以配置S7-200, 将其用作Modbus RTU从属设备, 与Modbus主设备通讯。

您可以在STEP 7-Micro/WIN指令树的“协议库”文件夹中找到这些指令。有了这些新指令, 您可以将S7-200用作Modbus从属设备。当您选择一条Modbus从属指令时, 会在项目中自动增加一个或多个相关子例行程序。

该标题讨论下列问题:

[使用Modbus协议的要求](#)

[Modbus协议的初始化和执行时间](#)

[Modbus编址](#)

[配置符号表](#)

**使用Modbus协议的要求**

[返回顶端](#)

Modbus从属协议指令使用下列S7-200资源:

- 初始化Modbus从属协议为Modbus从属协议通讯指定端口0。  
当端口0被用于Modbus从属协议通讯时, 就无法用于任何其他用途, 包括与STEP 7-Micro/WIN通讯。MBUS\_INIT指令控制将端口0指定给Modbus从属协议或PPI。
- Modbus从属协议指令影响与端口0中自由端口通讯相关的所有SM位置。
- Modbus从属协议指令使用3个子例行程序和2个中断例行程序。
- Modbus从属协议指令要求两个Modbus从属指令, 支持例行程序有1857个字节的程序空间。
- Modbus从属协议指令的变量要求779个字节的V内存程序块。该程序块的起始地址由用户指定, 专门保留用于Modbus变量。



**提示** 欲将端口0操作更替回PPI以便与STEP 7-Micro/WIN通讯, 用另一条MBUS\_INIT指令重新指定端口0。您还可以将S7-200上的模式开关设为“停止”模式。这样会将参数重设为端口0。

**Modbus协议的初始化和执行时间**

[返回顶端](#)

Modbus通讯使用CRC (循环冗余校验) 确保通讯信息的完整性。Modbus从属协议使用预先计算数值的表格减少处理讯息的时间。CRC表初始化约需425微秒。初始化在MBUS\_INIT子例行程序中进行, 通常在进入“运行”模式后用户程序首次扫描时执行。如果MBUS\_INIT子例行程序和其他初始化程序要求的时间超过500微秒扫描监视时间, 您需要重设监视定时器, 并将输出保持策略复位。 匕纒 一鼓? 藁 媳 丿 涑瞿? 榭喇佃去逼鬲捎孟蚰? 械涑鲂慈氩姆椒日瘳琛G 氩卧 [监视重设指令](#)。

当MBUS\_SLAVE子例行程序执行请求时, 扫描时间会延长。由于大多数时间用于计算Modbus CRC, 对于请求中和应答中的每个字节, 扫描时间会延长约650微秒。最大请求/应答 (读取或写入120个字) 使扫描时间延长约165毫秒。

**Modbus编址**

[返回顶端](#)

Modbus地址通常被写为包含数据类型和偏移的5个或6个字符的数值。第一个或第二个字符决定数据类型, 最后四个字符在数据类型中选择适当的数值。然后, Modbus主设备将地址映射至正确的功能。Modbus从属指令支持下列地址:

- 00001至000128是映射至Q0.0 - Q15.7的离散输出,
- 010001至010128是映射至I0.0 - I15.7的离散输入

- 030001至030032是映射至AIW0至AIW62的模拟输入寄存器
- 040001至04xxxx是映射至V内存的保持寄存器

所有Modbus地址均以1为基位。下表显示映射至S7-200地址的Modbus地址。

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	Hold Start
040002	HoldStart+2
040003	HoldStart+4
...	...
04xxxx	HoldStart+2 x (xxxx-1)

Modbus从属协议允许您限制Modbus主设备可存取的输入、输出、模拟输入和保持寄存器（V内存）数量。

MBUS\_INIT指令的MaxIQ参数指定Modbus主设备允许存取的离散输入或输出（Is或Qs）的最大数目。

MBUS\_INIT指令的MaxAI参数指定Modbus主设备允许存取的输入寄存器（AIWs）的最大数目。

MBUS\_INIT指令的MaxHold参数指定Modbus主设备允许存取的保持寄存器（V内存字）的最大数目。

请参阅MBUS\_INIT指令说明，了解有关为Modbus从属协议设定内存限制的进一步详情。

#### 配置符号表

[返回顶端](#)

在您为第一个符号输入一个地址后，符号表会自动计算和指定表中的其余符号。

您应当为表格指定一个起始V位置，该位置占用779个字节。确保指定的Modbus从属符号不会与指定给Modbus保持寄存器的V内存地址重叠。V内存的MBUS\_INIT指令中有HoldStart和MaxHold参数。如果内存区域出现任何重叠，MBUS\_INIT指令会返回一则错误消息。

另请参阅：

[MBUS\\_INIT指令](#)

[MBUS\\_SLAVE指令](#)

[Modbus从属协议执行错误](#)

## 11.5.2 使用Modbus从属协议指令

欲在S7-200程序中使用Modbus从属协议指令，请执行以下步骤：

1. 在程序中插入MBUS\_INIT指令，并仅限为一次扫描执行MBUS\_INIT指令。您可以使用MBUS\_INIT指令初始化或更改Modbus通讯参数。  
当您插入MBUS\_INIT指令时，会在程序中自动增加几个隐藏的子例行程序和中断例行程序。
2. 为Modbus从属协议指令要求的779个字节连续V内存指定一个起始地址。
3. 仅将一条MBUS\_SLAVE指令放入程序中。每次扫描时均调用该指令，为所有收到的请求提供服务。
4. 连接S7-200上的端口0和Modbus主设备之间的通讯电缆。



### 当心

带有不同参考电势的互连装置会使不需要的电流从互连电缆中通过。这些不需要的电流会引起通讯错误或损坏装置。确保所有用通讯电缆连接的装置共享一个共用电路参考或单独隔离，以防不需要的电流通过。

累加器 (AC0、AC1、AC2、AC3) 被Modbus从属指令使用，在“交叉参考”列表中显示。执行之前，Modbus从属指令累加器值当Modbus从属指令完成之前还原至累加器，以确保在执行Modbus从属指令时累加器中的所有用户数据被保留。

Modbus从属协议指令支持Modbus RTU协议。这些指令使用S7-200的自由端口实用程序支持最常用的Modbus功能。以下是受支持的Modbus功能：

功能	说明
1	读取单个 / 多个线圈 (离散输出) 状态。功能1返回任何数目输出点 (Qs) 的打开 / 关闭状态。
2	读取单个 / 多个接触 (离散输入) 状态。功能2返回任何数目输入点 (Is) 的打开 / 关闭状态。
3	读取单个 / 多个保持寄存器。功能3返回V内存的内容。保持寄存器是Modbus中的字数值，允许您在一次请求中最多读取120个字。
4	读取单个 / 多个输入寄存器。功能4返回模拟输入数值。
5	写入单个线圈 (离散输出)。功能5将离散输出点设为一个具体数值。该点并非强制数值，程序可以盖写Modbus请求写入的数值。
6	写入单个保持寄存器。功能6将一个单个保持寄存器数值写入S7-200的V内存。
15	写入多个线圈 (离散输出)。功能15将离散输出数值写入S7-200的Q图象寄存器。起始输出点必须在字节边界 (Q0.0或Q2.0) 开始，写入的输出数目必须是8的倍数。这是对Modbus从属协议指令的一个限制。这些点并非强制数值，程序可以盖写Modbus请求写入的数值。
16	写入多个保持寄存器。功能16将多个保持寄存器写入S7-200的V内存。在一个请求中最多可写入120个字。

### 另请参阅：

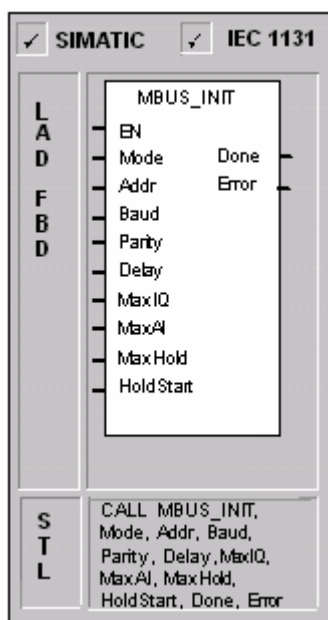
[MBUS\\_INIT指令](#)

[MBUS\\_SLAVE指令](#)

[Modbus从属协议执行错误](#)

## 11.5.3 MBUS\_INIT指令

输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
模式、地址、奇偶校验		VB, IB, QB, MB, SB, SMB, LB, AC, 常量, *VD, *AC,
*LD	字节[[[ENDKEEPN]]]	
波特、HoldStart		VD, ID, QD, MD, SD, SMD, LD, AC, 常量, *VD, *AC,
*LD	双字[[[ENDKEEPN]]]	
延迟、MaxIQ、MaxAI、MaxHold		VW, IW, QW, MW, SW, SMW, LW, AC, 常量, *VD, *AC,
*LD	字[[[ENDKEEPN]]]	
完成	I, Q, M, S, SM, T, C, V, L	布尔[[[ENDKEEPN]]]
错误	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	字节[[[ENDKEEPN]]]



MBUS\_INIT指令被用于启用和初始化或禁用Modbus通讯。在使用MBUS\_SLAVE指令之前，必须正确执行MBUS\_INIT指令。指令的“完成”位，才能继续执行下一条指令。在每次扫描且EN输入打开时执行该指令。

应当在每次通讯状态改变时执行MBUS\_INIT指令。因此，EN输入应当通过一个边缘检测元素用脉冲打开，或者仅在首次扫描时执行。

“模式”输入数值选择通讯协议：输入数值1将端口0指定给Modbus协议并启用协议；将输入数值0指定给PPI，并禁用Modbus协议。

“波特”参数将波特率设为1200、2400、4800、9600、19200、38400、57600或115200。

“地址”参数将地址设为1和247之间（包括1和247）的数值。请参阅《S7-200可编程控制器系统手册》中的范例程序。

“奇偶校验”参数被设为与Modbus主设备奇偶校验相匹配。可接受的数值为：

- 0-无奇偶校验
- 1-奇数奇偶校验
- 2-偶数奇偶校验

“延迟”参数通过将指定的毫秒数增加至标准Modbus讯息超时的方法延长标准Modbus讯息结束超时条件。该参数的典型数值在那咄 编嫌 0。如果您在使用带有纠错功能的调制解调器，将延迟设为50至100毫秒的数值。如果您在使用扩展频谱无线电，映借 10至100毫秒的数值。“延迟”数值可以是0至32767毫秒。

MaxIQ参数将供Modbus地址00xxxx和01xxxx使用的I和Q点数设为0至128之间的数值。数值0禁用所有向输入和输出的读取。建议 MaxIQ数值是128，该数值可在S7-200中存取所有的I和Q点。

MaxAI参数将供Modbus地址03xxx使用的字输入（AI）寄存器数目设为0至32之间的数值。数值0禁用模拟输入的读数。建议使用 MaxAI数值如下，这些数值可允许存取所有的S7-200模拟输入：

- 0用于CPU 221
- 16用于CPU 222
- 32用于CPU 224、226和226XM

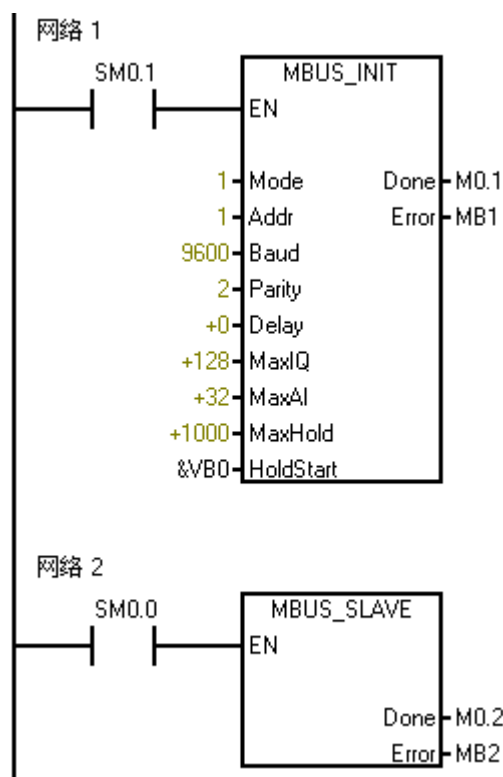
MaxHold参数设定供Modbus地址04xxx使用的V内存中的字保持寄存器数目。例如，为了允许主设备存取2000个字节的V内存，将MaxHold设为1000个字的数值（保持寄存器）。

HoldStart参数是V内存中保持寄存器的起始地址。该数值一般被设为VB0，因此HoldStart参数被设为&VB0（VB0地址）。其他V地址从VB0开始。Modbus主设备可存取V内存MaxHold个字数，从HoldStart开始。

MBUS\_INIT指令完成时，“完成”输出打开。“错误”输出字节包含执行指令的结果。请参阅 [Modbus从属协议执行错误代码](#)中可能因执行指令而产生的错误条件。

#### 程序举例

LAD

**STL**

NETWORK 1 //首次扫描时初始化Modbus从属装置协议  
 //将从属装置地址设为1, 将端口0设为9600波特, 将奇偶校验设为偶数,  
 //允许存取所有的I、Q和AI数值, 允许存取1000台保存寄存器(2000个字节)  
 //从VB0开始。

LD SM0.1

CALL MBUS\_INIT, 1, 1, 9600, 2, +0, +128, +32, +1000, &VB0, M0.1, MB1

**NETWORK 2**

//每次扫描时执行Modbus从属装置协议

LD SM0.0

CALL MBUS\_SLAVE, M0.2, MB2

另请参阅:

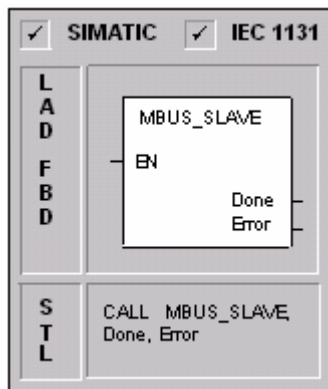
[MBUS\\_SLAVE指令](#)

[Modbus从属协议执行错误](#)

**11.5.4 MBUS\_Slave指令**

输入 / 输出	操作数	数据类型[[[ENDKEEPN]]]
完成	I, Q, M, S, SM, T, C, V, L	布尔[[[ENDKEEPN]]]
错误	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	字节[[[ENDKEEPN]]]





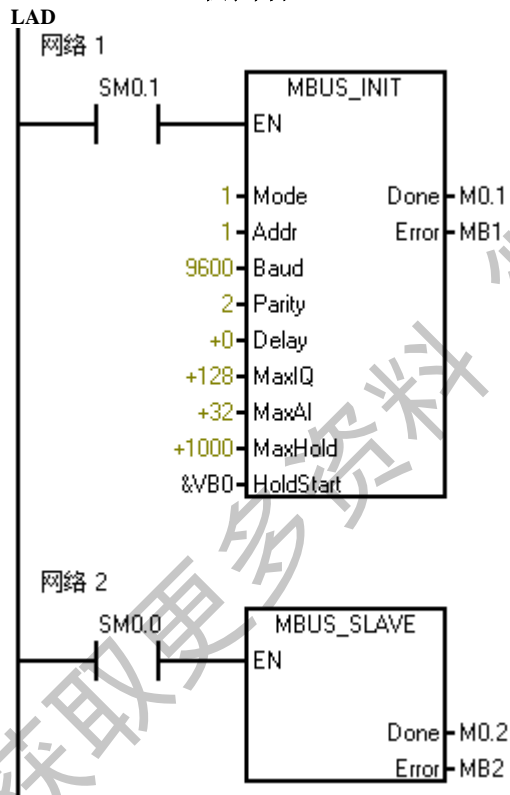
MBUS\_SLAVE指令被用于为Modbus主设备发出的请求服务，并且必须在每次扫描时执行，以便允许该指令检查和回答 ModbusC 响 在每次扫描且EN输入开启时执行该指令。MBUS\_SLAVE指令无输入参数。

当MBUS\_SLAVE指令对Modbus请求作出应答时，“完成”输出打开。如果没有需要服务的请求时，“完成”输出关闭。

“错误”输出包含执行该指令的结果。该输出只有在“完成”打开时才有效。

如果“完成”关闭，错误参数不会改变。请参阅 [Modbus从属协议执行错误代码](#) 中可能因执行该指令而导致的错误条件。请参阅《S7-200可编程控制器系统手册》中的范例程序。

#### 程序举例



#### STL

```

NETWORK 1 //首次扫描时初始化Modbus从属装置协议
//将从属装置地址设为1，将端口0设为9600波特，将奇偶校验设为偶数，
//允许存取所有的I、Q和AI数值，允许存取1000台保存寄存器（2000个字节）
//从VBO开始。
LD SM0.1
CALL MBUS_INIT, 1, 1, 9600, 2, +0, +128, +32, +1000, &VBO, M0.1, MB1

NETWORK 2
//每次扫描时执行Modbus从属装置协议
LD SM0.0
CALL MBUS_SLAVE, M0.2, MB2

```

另请参阅：  
[MBUS\\_INIT指令](#)  
[Modbus从属协议执行错误](#)

### 11.5.5 Modbus从属协议执行错误代码

错误代码	说明
0	无错误
1	内存范围错误
2	非法波特率或奇偶校验
3	非法从属地址
4	非法Modbus参数值
5	保持寄存器与Modbus从属符号重叠
6	收到奇偶校验错误
7	收到CRC错误
8	非法功能请求 / 功能不受支持
9	请求中的非法内存地址
10	从属功能未启用

获取更多资料 微信搜索蓝领星球

获取更多资料 微信搜索蓝领星球

## Top Level Intro

This page is printed before a new  
top-level chapter starts

# Part

# XIII

获取更多资料 微信搜索 蓝领星球

## 12 快速参考手册

本标题总结了以下条目的主要信息:

[特殊内存位](#)

[中断事件说明](#)

[S7-200 CPU内存范围和功能总结](#)

[高速计数器HSC0、HSC1、HSC2、HSC3、HSC4、HSC5](#)

[S7-200指令](#)

### 特殊内存位

SM0.0	始终打开	SM1.0	操作结果 = 0
SM0.1	首次扫描	SM1.1	溢出或非法数值
SM0.2	保留性数据丢失	SM1.2	负数结果
SM0.3	电源开启	SM1.3	除以0
SM0.4	秒关闭 / 30秒打开	SM1.4	表格已满
SM0.5	.5秒关闭 / 0.5秒打开	SM1.5	表格空置
SM0.6	关闭1次扫描 / 打开1次扫描	SM1.6	BCD至二进制转换错误
SM0.7	在RUN (运行) 位置转换	SM1.7	ASCII至十六进制转换错误

### 按优先级顺序排列的中断事件

事件号码	中断说明	优先级别群组	在群组中的优先级
8	端口0: 接收字符	通讯 (最高)	0
9	端口0: 传输完成		0
23	端口0: 接收讯息完成		0
24	端口1: 接收讯息完成		1
25	端口1: 接收字符		1
26	端口1: 传输完成		1
19	PTO 0完成中断	离散 (中等)	0
20	PTO 1完成中断		1
0	I0.0, 上升边缘		2
2	I0.1, 上升边缘		3
4	I0.2, 上升边缘		4
6	I0.3, 上升边缘		5
1	I0.0, 下降边缘		6
3	I0.1, 下降边缘		7
5	I0.2, 下降边缘		8
7	I0.3, 下降边缘		9
12	HSC0 CV=PV (当前值 = 预设值)		10
27	HSC0方向改变		11
28	HSC0外部重设		12
13	HSC1 CV=PV (当前值 = 预设值)		13
14	HSC1方向输入改变		14
15	HSC1外部重设		15
16	HSC2 CV=PV		16
17	HSC2方向改变		17
18	HSC2外部重设		18
32	HSC3 CV=PV (当前值 = 预设值)		19
29	HSC4 CV=PV (当前值 = 预设值)		20
30	HSC4方向改变		21
31	HSC4外部重设		22
33	HSC5 CV=PV (当前值 = 预设值)		23
10	定时中断0	定时 (最低)	0
11	定时中断1		1
21	计时器T32 CT=PT中断		2
22	计时器T96 CT=PT中断		3

### 内存范围和功能

说明	CPU 221	CPU 222	CPU 224	CPU 226
用户程序大小	4千字节	4千字节	8千字节	V 1.22, V2.0 12千字节 XP
8千字节 V 1.23 16千字节 XM, V2.0				
用户数据大小	2千字节	2千字节	5千字节 V 1.22 8千字节 V 2.0	10千字节 XP
5千字节 V 1.23 10千字节 XM, V2.0				
进程图像输入寄存器	I0.0至I15.7	I0.0至I15.7	I0.0至I15.7	I0.0至I15.7
进程图像输出寄存器	Q0.0至Q15.7	Q0.0至Q15.7	Q0.0至Q15.7	Q0.0至Q15.7
模拟输入 (仅限读取)	--	AIW0至AIW30		AIW0至AIW62
AIW0至AIW62				
模拟输出 (仅限写入)	--	AQW0至AQW30		AQW0至AQW62
AQW0至AQW62				

变量内存 (V) V 2.0 VB0至VB10239 XP	VB0至VB2047 VB0至VB5119	VB0至VB2047 V 1.23VB0至VB10239 XM, V 2.0	VB0至VB5119 V 1.22VB0至VB8119
Local memory (L) LB0至LB59	LB0至LB59	LB0至LB59	LB0至LB59
本地内存 (L) 1 LB0至LB63	LB0至LB63	LB0至LB63	LB0至LB63
位内存 (M) M0.0至M31.7	M0.0至M31.7	M0.0至M31.7	M0.0至M31.7
特殊内存 (SM) SM0.0至SM29.7	仅限读取y SM0.0至SM549.7SM0.0至SM29.7	SM0.0至SM179.7SM0.0至SM29.7	SM0.0至SM299.7 SM0.0至SM549.7SM0.0至SM29.7
计时器保留性打开延迟	10毫秒 1毫秒	100毫秒 10毫秒	100毫秒 打开 / 关闭延迟 1毫秒
T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	256 (T0至T255)T0、T64 T1至T4、和 T65至T68 T5至T31、和 256 (T0至T255)T0、T64 T1至T4、和 T65至T68
T5至T31、和 T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	T5至T31、和 T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	T5至T31、和 T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	256 (T0至T255)T0、T64 T1至T4、和 T65至T68
T65至T68 T5至T31、和 T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	T65至T68 T5至T31、和 T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	T65至T68 T5至T31、和 T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	256 (T0至T255)T0、T64 T1至T4、和 T65至T68
T1至T4、和 T65至T68 T5至T31、和 T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	T1至T4、和 T65至T68 T5至T31、和 T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	T1至T4、和 T65至T68 T5至T31、和 T69至T95 T32、T96 T33至T36、和 T97至T100 T37至T63、和 T101至T255	256 (T0至T255)T0、T64 T1至T4、和 T65至T68
计数器	C0至C255	C0至C255	C0至C255
高速计数器	HC0、HC3、HC4和HC5	HC0、HC3、HC4和HC5	HC0、HC3、HC4和HC5
HC0至HC5	HC0至HC5	HC0至HC5	HC0至HC5
连续控制中继 (S) S0.0至S31.7	S0.0至S31.7	S0.0至S31.7	S0.0至S31.7
累加器寄存器	AC0至AC3	AC0至AC3	AC0至AC3
跳接 / 标签	0至255	0至255	0至255
调用 / 子例行程序	0至63	0至63	0至63 V 1.230至127 XM, V 2.0
中断例行程序	0至127	0至127	0至127
正数 / 负数转换	256	0至127	0至127
PID环路	0至7	0至7	0至7
端口	端口0	端口0	Port 0 Port 0, Port 1 XP
端口0、端口1	端口0	端口0	Port 0 Port 0, Port 1 XP

### 高速计数器HSC0、HSC3、HSC4和HSC5

模式	HSC0 I0.0 I0.5 Clk	HSC3 I0.1 I0.4 Clk	HSC4 I0.2 重设	HSC5 I0.1 Clk	I0.3 Clk	I0.4 Clk
0	Clk	Clk	重设	Clk	Clk	Clk
1	Clk 重设	Clk	重设	Clk	Clk	Clk
2	Clk	Clk	重设	Clk	Clk	Clk
3	Clk	方向	重设	Clk	Clk	Clk
4	Clk	方向	重设	Clk	Clk	Clk
5	Clk	方向	重设	Clk	Clk	Clk
6	Clk向上	Clk向下	重设	Clk	Clk向上	Clk向上
7	Clk向下	Clk向上	重设	Clk	Clk向下	Clk向下
8	Clk向下 重设	Clk向下	重设	Clk	Clk向上	Clk向上
9	相位A	相位B	重设	Clk	相位A	相位A
10	相位B	相位A	重设	Clk	相位B	相位B
11	相位B 重设	相位B	重设	Clk	相位A	相位A

### 高速计数器HSC1和HSC2

模式	HSC1 I0.6 I1.4 Clk	HSC2 I0.7 I1.5 Clk	I1.0 重设	I1.1 启动	I1.2 Clk
0	Clk	Clk	重设	启动	Clk
1	Clk 重设	Clk	重设	启动	Clk
2	Clk 重设	Clk	重设	启动	Clk
3	Clk	启动 方向	重设	启动	Clk

方向					
4	Clk	方向	重设		Clk
方向	重设				
5	Clk	方向	重设	启动	Clk
方向	重设	启动			
6	Clk向上	Clk向下			Clk向上
Clk向下					
7	Clk向上	Clk向下	重设		Clk向上
Clk向下	重设				
8	Clk向上	Clk向下	重设	启动	Clk向上
Clk向下	重设	启动			
9	相位A	相位B			相位A
相位B					
10	相位A	相位B	重设		相位A
相位B	重设				
11	相位A	相位B	重设	启动	相位A
相位B	重设	启动			

## 指令

布尔指令		数学、递增和递减指令			
LD	BitLDI	Bit LDN	位 LDNI	位	载入立即载入 载入Not (非)
载入Not (非) 立即	+I		IN1, OUT+D		IN1, OUT +R
OUT	加整数、双整数或实数	IN1+OUT=OUT			
A	位AI	位 AN	位 ANI	位	AND (与) AND (与) 立即 AND Not (与非)
AND Not (与非) 立即	-I		IN1, OUT-D		IN1, OUT -R
OUT	减去整数、双整数或实数	OUT-IN1=OUT			
O	位OI	位 ON	位 ONI	位	OR (或) OR (或) 立即 OR Not (或非) OR
Not (或非) 立即	MUL		IN1, OUT	乘以整数 (16*16->32)	
LDBx	IN1, IN2	载入字节比较结果	IN1 (x:<, <=, =, >=, >, <> ) IN2	*I	IN1, OUT
*D	IN1, OUT *R		IN1, IN2	乘以整数、双整数或实数	IN1 * OUT = OUT
ABx	IN1, IN2	字节比较AND (与) 结果	IN1 (x:<, <=, =, >=, >, <>) IN2	DIV	IN1,
OUT	除以整数 (16/16->32)				
OBx	IN1, IN2	字节比较OR (或) 结果	IN1 (x:<, <=, =, >=, >, <>) IN2	/I	IN1,
OUT/D,	IN1, OUT /R		IN1, OUT	除以整数、双整数或实数	OUT / IN1 = OUT
LDWx	IN1, IN2	载入字比较结果	IN1 (x:<, <=, =, >=, >, <>) IN2	SQRT	IN,
OUT	平方根				
AWx	IN1, IN2	字比较AND (与) 结果	IN1 (x:<, <=, =, >=, >, <>) IN2	LN	IN,
OUT	自然对数				
OWx	IN1, IN2	字比较OR (或) 结果	IN1 (x:<, <=, =, >=, >, <>) IN2	EXP	IN,
OUT	自然指数				
LDDx	IN1, IN2	载入双字比较结果	IN1 (x:<, <=, =, >=, >, <>) IN2	SIN	IN,
OUT	正弦				
ADx	IN1, IN2	双字比较AND (与) 结果	IN1 (x:<, <=, =, >=, >, <>) IN2	COS	IN,
OUT	余弦				
ODx	IN1, IN2	双字比较OR (或) 结果	IN1 (x:<, <=, =, >=, >, <>) IN2	TAN	IN,
OUT	正切				
LDRx	IN1, IN2	载入实数比较结果	IN1 (x:<, <=, =, >=, >, <>) IN2	INCB	OUT
INCW	OUT INCD	OUT	递增字节、字或双字		
ARx	IN1, IN2	实数比较AND (与) 结果	IN1 (x:<, <=, =, >=, >, <>) IN2	DECB	OUT
DECW	OUT DECD	OUT	递减字节、字或双字		
ORx	IN1, IN2	实数比较OR (或) 结果	IN1 (x:<, <=, =, >=, >, <>) IN2	PID	TBL,
LOOP	PID环路				
NOT	堆栈否定				
<b>计时器和计数器指令</b>					
EU	上升边缘检测	TON	Txxx, PTTOF	Txxx, PT TONR	Txxx,
PT	打开-延迟计时器 关闭-延迟计时器	保留性打开-延迟计时器			
ED	下降边缘检测				
=	位=I	位	指定数值 指定立即数值	CTU	Cxxx, PV
CTD	Cxxx, PV CTUD	Cxxx, PV BITIM	OUT CITIM		IN, OUT
向上计数	向下计数	向上 / 向下计数	开始间隔时间	计算间隔时间	
S	位, N	设置位范围			
R	位, N	重设位范围			
<b>实时时间时钟指令</b>					
读取当日时间时钟	读取扩展的当日时间时钟				
SI	位, N	设置立即位范围	TODR	T TODRX	T
写入当日时间时钟	写入扩展的当日时间时钟				
RI	位, N	重设立即位范围	TODW	T TODWX	T
<b>程序控制指令</b>					
LDSx	IN1, IN2	载入字符串比较结果	END	程序有条件结束	
	IN1 (x: =, <>) IN2	STOP	转入STOP (停止) 模式		
ASx	IN1, IN2	字符串比较AND (与) 结果	WDR	监视器重设 (300毫秒)	
	IN1 (x: =, <>) IN2	DLED	IN	诊断LED	

OSx	IN1, IN2	字符串比较OR (或) 结果	JMP	N	
		跳接至定义的标签			
ALD	IN1 (x: =, <=>) IN2	AND (与) 载入	CALL	N	定义要跳接至的标签
		最多为16个选用参数]			调用子例行程序[N1, ...]
OLD		OR (或) 载入	CRET		从SBR有条件返回
LPS		逻辑进栈 (堆栈控制)	FOR		INDX, INIT, FINAL NEXT
		用于 / 下一个环路			
LRD		逻辑读取 (堆栈控制)			<b>序列控制中继</b>
		LSCR	N		载入
LPP		逻辑出栈 (堆栈控制)	SCRT	N	转移
LDS	N	载入堆栈 (堆栈控制)	CSCRE		有条件结束
AENO	AND (与) ENO	SCRE			序列
<b>移动、移位和旋转指令</b>			<b>表格、查找和转换指令</b>		
MOVB	IN, OUT	移动			ATT DATA,
TBL		在表格中增加数据			
MOVW	IN, OUT			LIFO	TBL, DATA
		从表格中获取数据			
MOVD	IN, OUT			FIFO	TBL, DATA
MOVR	IN, OUT				
BIR	IN, OUTBIW				
FND=	TBL, PTN, INDX	FND<>	TBL, PTN, INDX		在表格中查找与比较相符的数据值
BMB	IN, OUT, N	块移动		FND<	TBL, PTN, INDX
BMW	IN, OUT, N			FND>	TBL, PTN, INDX
BMD	IN, OUT, N				
SWAP	IN	交换字节	FILL	IN, OUT, N	用格式填充内存空间
SHRB	DATA, S_BIT, N	移位寄存器		BCDI	OUT
IBCD	OUT	将BCD转换为整数	将整数转换为BCD		
SRB	OUT, NSRW	OUT, N	SRD	OUT, N	向右移位 字节 字
BTI	IN, OUTITB	IN, OUT	ITD	IN, OUT	DTI IN,
OUT		将字节转换为整数	将整数转换为字节	将整数转换为双整数	将双整数转换为整数
SLB	OUT, NSLW	OUT, N	SLD	OUT, N	向左移位 字节 字
DTR	IN, OUTTRUNC	IN, OUT	ROUND	IN, OUT	将双字转换为实数
RRB	OUT, NRRW	OUT, N	RRD	OUT, N	向右旋转 字节 字
ATH	IN, OUT, LENHTA	IN, OUT, LEN	ITA	IN, OUT, FMT	DTA IN, OUT, FM
RTA	IN, OUT, FM				
RLB	OUT, NRLW	OUT, N	RLD	OUT, N	Rotate Left 字节 字
DECO	IN, OUTENCO	IN, OUT	SEG	IN, OUT	解码编码
		生成7段格式			
<b>Logical Instructions</b>			<b>ITS</b>		
ANDB	IN1, OUTANDW	IN1, OUT	ANDD	IN1, OUT	逻辑AND (与) 字节
ORB	IN1, OUTORW	IN1, OUT	ORD	IN1, OUT	逻辑OR (或) 字节
STR	STR, INDX, OUT	STR, INDX, OUT	STI	STR, INDX, OUT	STD STR, INDX, OUT
XORB	IN1, OUT	逻辑XOR (异-或) 字节			
XORW	IN1, OUT	字			
XORD	IN1, OUT	双字			
INVB	OUTINVV	OUT	INVD	OUT	反转 字节
CEVNT	EVNT	禁用中断	DISI	ATCH	INT, EVNT
		将中断例程序附加在事件上			清除中断事件
<b>String Instructions</b>			<b>通讯指令</b>		
SLEN	IN, OUTSCAT	IN, OUT	SCPY	IN, OUT	SSCPY IN, INDX, N, OUT
CFND	IN1, IN2, OUT	IN1, IN2, OUT			字符串长度并置字符串 复制字符串
RVCV	TBL, PORT	TBL, PORT	NETR	TBL, PORT	GPA ADDR, PORT
SPA	ADDR, PORT	自由端口传输			自由端口接收讯息 网络读取 网络写入 获取端口地址 设置端口地址
<b>高速指令</b>			<b>Q</b>		
HDEF	HSC, MODE	HSC	PLS	Q	定义高速计数器代码
		驱动高速计数器 脉冲输出			



获取更多资料 微信搜索蓝领星球

# Index

## - 1 -

100毫秒分辨率（计时器） 194, 293, 376  
 10毫秒分辨率（计时器） 194, 293, 376  
 1x/4x模式（高速计数器） 200, 299, 382  
 1毫秒分辨率（计时器） 194, 293, 376

## - 2 -

22x系列: 571  
 22x系列: CPU内存地址范围 571

## - A -

A/B相正交计数器 200, 299, 382  
 ALT键: 7, 112, 128, 134, 636, 642  
 ALT键:TD 200键 128, 134, 636, 642  
 ALT键:捷径键 7, 112  
 AN 66, 357  
 AND（与）指令: 468  
 AND（与）指令:AND（与）方框（FBD） 267, 268, 520  
 ASCII:ASCII词汇规则 581  
 ASCII:常量举例 578  
 ASCII:文件关键字 580  
 ASCII导入文件词汇规则 581  
 ASCII导入文件规则 581  
 ASCII导入文件句法 581  
 AS-i主模块: 162, 670  
 AS-i主模块:AS-i模块的错误代码 722  
 AS-i主模块:AS-i向导概述 162, 670  
 AS-i主模块:CP243-2 AS-i模块性能 162, 670  
 AWL文件导入 581

## - B -

Building Status Charts 92, 693

## - C -

CALL 453, 464, 480, 490, 500  
 Chart Program Status 92, 693

COM端口 532  
 CP243-1 IT: 156, 664  
 CP243-1 IT:互联网模块性能 156, 664  
 CP243-1 IT:互联网向导概述 157, 665  
 CP243-1 IT:以太网 / 互联网模块的错误代码 717  
 CP243-1: 154, 662  
 CP243-1:以太网 / 互联网模块的错误代码 717  
 CP243-1:以太网模块性能 154, 662  
 CP243-1:以太网向导概述 155, 663  
 CP243-2 AS-i:AS-i模块性能 162, 670  
 CP243-2 AS-i:AS-i向导概述 162, 670  
 CPU 711  
 CPU: 17, 615, 712  
 CPU: 与PC内存中打开的项目进行比较 618  
 CPU:22x系列 615  
 CPU:22x系列内存地址范围 571  
 CPU:编译和运行时间错误 712  
 CPU:当日时钟时间 618  
 CPU:非严重错误内存位置 714  
 CPU:检视CPU信息 615  
 CPU:类型选择 615  
 CPU:模式 74, 247, 346, 429, 521, 678  
 CPU:内存磁带 619  
 CPU:删除CPU内存 623  
 CPU:适当的空间 12  
 CPU:数据块 235, 334, 417, 593  
 CPU:系统块 520, 617  
 CPU:运行 / 停止模式 518, 688  
 CPU:执行扫描循环 17  
 CPU:指令支持 442  
 CPU代码寄存器（SMB6） 506  
 CPU内存中的指令大小 583  
 CP卡 545

## - D -

DP通讯: 505, 615  
 DP通讯:DP总线故障（SMB5） 505  
 DP通讯:PLC信息 615

## - E -

EEPROM 575, 711  
 EEPROM中永久性V数据存储（SMB31- SMB32） 508  
 EM 241调制解调器: 144, 652

EM 241调制解调器:EM 241调制解调器模块性能  
144, 652  
EM 241调制解调器:EM 241调制解调器向导概述  
148, 656  
EM  
241调制解调器:错误代码 (MODx\_MSG和MODx\_XF  
R) 715  
EM 253: 151, 659  
EM 253位置: 148, 656  
EM 253位置:错误代码 716  
EM 253位置:控制面板 151, 659  
EM 253位置:位置模块向导概述 149, 657  
EM 253位置:位置模块性能 148, 656  
EN 442  
ENO用法 442

## - F -

FBD Instructions (IEC):字符串指令 498  
FBD 指令 (SIMATIC):字符串指令 463  
FBD编辑器: 42  
FBD编辑器:编程简介 42  
FBD编辑器:地址 47  
FBD编辑器:反转输入逻辑 42  
FBD编辑器:快捷键 7, 112  
FBD编辑器:立即参数 42  
FBD编辑器:书签 370  
FBD编辑器:图形建立 44, 268, 271  
FBD编辑器:网络 44  
FBD编辑器:移除输入 (工具条按钮) 268  
FBD编辑器:与其他编辑器比较 20  
FBD编辑器:元素 42  
FBD编辑器:增加输入 (工具条按钮) 267  
FBD编辑器:指令 44, 268, 271  
FBD编辑器:注解 51, 289  
FBD编辑器程序条目错误 60  
FBD指令 (IEC) 258  
FBD指令 (IEC):比较指令 493  
FBD指令 (IEC):程序控制指令 497  
FBD指令 (IEC):计时器指令 499  
FBD指令 (IEC):计数器指令 495  
FBD指令 (IEC):逻辑指令 496  
FBD指令 (IEC):时钟指令 492  
FBD指令 (IEC):数学指令 496  
FBD指令 (IEC):数字指令 497  
FBD指令 (IEC):通讯指令 493  
FBD指令 (IEC):位逻辑指令 492

FBD指令 (IEC):移动指令 496  
FBD指令 (IEC):移位 / 旋转指令 498  
FBD指令 (IEC):中断指令 495  
FBD指令 (IEC):转换指令 494  
FBD指令 (SIMATIC) 257  
FBD指令 (SIMATIC):比较指令 456  
FBD指令 (SIMATIC):表格指令 464  
FBD指令 (SIMATIC):程序控制指令 461  
FBD指令 (SIMATIC):浮点指令 (实数数学) 459  
FBD指令 (SIMATIC):计时器指令 464  
FBD指令 (SIMATIC):计数器指令 458  
FBD指令 (SIMATIC):逻辑指令 460  
FBD指令 (SIMATIC):时钟指令 455  
FBD指令 (SIMATIC):通讯指令 455  
FBD指令 (SIMATIC):位逻辑指令 455  
FBD指令 (SIMATIC):移动指令 461  
FBD指令 (SIMATIC):移位 / 旋转指令 462  
FBD指令 (SIMATIC):整数数学指令 459  
FBD指令 (SIMATIC):中断指令 460  
FBD指令 (SIMATIC):转换指令 457  
FBD中的逻辑流指示器 259  
Force:from Status Chart View 92, 693

## - H -

HDEF控制位 200, 299, 382  
HMI 541, 545  
HSC: 200, 299, 382  
HSC: 向导 116, 624  
HSC:HSC0/HSC1/和HSC2高速计数器寄存器 (SMB  
36-SMB65) 509  
HSC:HSC3/HSC4/和HSC5高速计数器寄存器 (SMB  
136-SMB165) 514  
HSC:当前值和预设值 200, 299, 382  
HSC:定义控制位 200, 299, 382  
HSC:模式和输入赋值 200, 299, 382  
HSC:设置动态参数 200, 299, 382  
HSC操作模式 200, 299, 382  
HSC当前值和预设值 200, 299, 382  
HSC的初始化顺序 200, 299, 382  
HSC动态参数 200, 299, 382

## - I -

I/O错误 / 状态 712, 714

I/O错误状态 (SMB5) 505  
 I/O扩充总线 - 通讯错误 (SMW98) 513  
 I/O模块代码和错误寄存器 (SMB8-SMB21) 506  
 I/O模块--请参阅扩充模块 516  
 IEC 1131-3编程模式 23  
 IEC 1131-3编程模式: 191, 284, 366, 572  
 IEC 1131-3编程模式: IEC和SIMATIC数据类型  
 191, 284, 366, 572  
 IEC 1131-3编程模式:超载指令 442  
 IEC 1131-3编程模式:非标准指令 442  
 IEC 1131-3编程模式:全局变量表 182, 275, 359,  
 602  
 IEC FBD Instructions:字符串指令 498  
 IEC FBD指令: 258  
 IEC FBD指令:比较指令 493  
 IEC FBD指令:程序控制指令 497  
 IEC FBD指令:计时器指令 499  
 IEC FBD指令:计数器指令 495  
 IEC FBD指令:逻辑指令 496  
 IEC FBD指令:时钟指令 492  
 IEC FBD指令:数学指令 496  
 IEC FBD指令:数字指令 497  
 IEC FBD指令:通讯指令 493  
 IEC FBD指令:位逻辑指令 492  
 IEC FBD指令:移动指令 496  
 IEC FBD指令:移位 / 旋转指令 498  
 IEC FBD指令:中断指令 495  
 IEC FBD指令:转换指令 494  
 IEC LAD指令: 172  
 IEC LAD指令:比较指令 483  
 IEC LAD指令:表格指令 489  
 IEC LAD指令:程序控制指令 487  
 IEC LAD指令:计时器指令 490  
 IEC LAD指令:计数器指令 485  
 IEC LAD指令:逻辑指令 486  
 IEC LAD指令:时钟指令 482  
 IEC LAD指令:数学指令 486  
 IEC LAD指令:数字指令 487  
 IEC LAD指令:通讯指令 483  
 IEC LAD指令:位逻辑指令 482  
 IEC LAD指令:移动指令 486  
 IEC LAD指令:移位 / 旋转指令 488  
 IEC LAD指令:中断指令 485  
 IEC LAD指令:转换指令 484  
 IEC LAD指令:字符串指令 488  
 INT 229, 328, 411

## - L -

LAD 指令 (IEC):中断指令 485  
 LAD/FBD: 710  
 LAD/FBD:编辑程序错误 710  
 LAD编辑器: 20  
 LAD编辑器:编程概述 24  
 LAD编辑器:程序条目错误 40  
 LAD编辑器:地址 29  
 LAD编辑器:书签 370  
 LAD编辑器:与其他编辑器比较 20  
 LAD编辑器:元素 24  
 LAD编辑器:指令 26  
 LAD编辑器:注解 32  
 LAD指令 (IEC): 172  
 LAD指令 (IEC):比较指令 483  
 LAD指令 (IEC):表格指令 489  
 LAD指令 (IEC):程序控制指令 487  
 LAD指令 (IEC):计时器指令 490  
 LAD指令 (IEC):计数器指令 485  
 LAD指令 (IEC):逻辑指令 486  
 LAD指令 (IEC):时钟指令 482  
 LAD指令 (IEC):数学指令 486  
 LAD指令 (IEC):数字指令 487  
 LAD指令 (IEC):通讯指令 483  
 LAD指令 (IEC):位逻辑指令 482  
 LAD指令 (IEC):移动指令 486  
 LAD指令 (IEC):移位 / 旋转指令 488  
 LAD指令 (IEC):转换指令 484  
 LAD指令 (IEC):字符串指令 488  
 LAD指令 (SIMATIC) 172, 443  
 LAD指令 (SIMATIC):比较指令 444  
 LAD指令 (SIMATIC):表格指令 452  
 LAD指令 (SIMATIC):程序控制指令 450  
 LAD指令 (SIMATIC):浮点指令 (实数数学) 447  
 LAD指令 (SIMATIC):计时器指令 453  
 LAD指令 (SIMATIC):计数器指令 447  
 LAD指令 (SIMATIC):逻辑指令 449  
 LAD指令 (SIMATIC):时钟指令 444  
 LAD指令 (SIMATIC):通讯指令 444  
 LAD指令 (SIMATIC):移动指令 450  
 LAD指令 (SIMATIC):移位 / 旋转指令 451  
 LAD指令 (SIMATIC):整数数学指令 448  
 LAD指令 (SIMATIC):中断指令 449  
 LAD指令 (SIMATIC):转换指令 446  
 LAD指令 (SIMATIC):字符串指令 451

LAD中的功率流指示器 173  
 Locks:Status Chart View Force and Unforce 92, 693  
 LVT: 231, 330, 413, 612  
 LVT:ASCII文本文件说明 581  
 LVT:编程 231, 330, 413, 612  
 LVT:编址 18  
 L内存 231, 330, 413, 612

## - M -

MaxAI (Modbus) 749  
 MaxHold (Modbus) 749  
 MaxIQ (Modbus) 749  
 MBUS\_INIT 751  
 MBUS\_Slave 753  
 Micro/DOS项目 104  
 Micro/WIN (早期版本) 104  
 Modbus库:简介 726  
 Modbus协议库 749  
 Modbus协议库:MBUS\_INIT 751  
 Modbus协议库:MBUS\_Slave 753  
 Modbus协议库:Modbus功能 750  
 Modbus协议库:编址 749  
 Modbus协议库:初始化和执行时间 749  
 Modbus协议库:从属协议执行错误代码 755  
 Modbus协议库:配置符号表 749  
 Modbus协议库:使用Modbus指令 750  
 Modbus协议库:要求 749  
 MPI通讯: 615  
 MPI通讯:PLC信息 615

## - N -

NETR Wizard 118, 626  
 NETW Wizard 118, 626

## - O -

OR (或) 指令:OR (或) 方框 (FBD) 267, 268

## - P -

PC/PPI电缆 545  
 PCL: 17  
 PID 120, 627

PID环路控制:PID调谐控制面板 122, 630  
 PID向导 (成比例整数派生环路) 120, 627  
 PLC 714  
 PLC > 编译命令 239, 338, 421, 601  
 PLC > 清除命令 623  
 PLC: 615, 712  
 PLC:与PC内存中打开的项目进行比较 618  
 PLC:22x系列 615  
 PLC:22x系列内存地址范围 571  
 PLC:编译和运行时间错误 712  
 PLC:当日时钟时间 (PLC时钟设置) 618  
 PLC:检视PLC信息 615  
 PLC:类型选择 615  
 PLC:模式 74, 247, 346, 429, 521, 678  
 PLC:内存磁带 619  
 PLC:删除PLC内存 623  
 PLC:适当的空间 12  
 PLC:数据块 235, 334, 417, 593  
 PLC:系统块 520, 617  
 PLC:严重错误 711  
 PLC:运行/停止模式 518, 688  
 PLC:执行扫描循环 17  
 PLC:指令支持 442  
 PLS 210, 309, 392  
 PLS (脉冲) 指令PTO概要表 (SMB166-SMB194) 515  
 POU: 580  
 POU:保留关键字 580  
 PPI通讯: 615  
 PPI通讯:PLC信息 615  
 PPI通讯:单台主设备网络 541  
 PPI通讯:多台主设备网络 541  
 PPI通讯:复杂的网络 541  
 PROFIBUS通讯: 516, 534  
 PROFIBUS通讯:SMB200-SMB549智能模块状态 516  
 PROFIBUS通讯:通讯选择 534  
 Programs:Chart Program Status 92, 693  
 PTO / PWM高速输出寄存器 (SMB66 ?SMB85) 511  
 PTO/PWM初始化 210, 309, 392  
 PTO位置向导 149, 657  
 PWM位置向导 149, 657

## - R -

RS-232 DTE/DCE接头外接插针 550

RS-232设备 545

RS-485接头外接插针 550

**- S -**S7-200 CPU型号提供的版本指令支持（固件）  
442

S7-200固件版本 442, 615

S7-200设备安装指南 12

SBR 224, 323, 406

SIMATIC FBD 指令:逻辑指令 460

SIMATIC FBD 指令:移动指令 461

SIMATIC FBD 指令:中断指令 460

SIMATIC FBD指令 257

SIMATIC FBD指令:比较指令 456

SIMATIC FBD指令:表格指令 464

SIMATIC FBD指令:程序控制指令 461

SIMATIC FBD指令:浮点指令（实数数学） 459

SIMATIC FBD指令:计时器指令 464

SIMATIC FBD指令:计数器指令 458

SIMATIC FBD指令:时钟指令 455

SIMATIC FBD指令:通讯指令 455

SIMATIC FBD指令:位逻辑指令 455

SIMATIC FBD指令:移位 / 旋转指令 462

SIMATIC FBD指令:转换指令 457

SIMATIC FBD指令:字符串指令 463

SIMATIC LAD 指令:程序控制指令 450

SIMATIC LAD 指令:逻辑指令 449

SIMATIC LAD 指令:移动指令 450

SIMATIC LAD指令: 172

SIMATIC LAD指令:比较指令 444

SIMATIC LAD指令:表格指令 452

SIMATIC LAD指令:浮点指令（实数数学） 447

SIMATIC LAD指令:计时器指令 453

SIMATIC LAD指令:计数器指令 447

SIMATIC LAD指令:时钟指令 444

SIMATIC LAD指令:通讯指令 444

SIMATIC LAD指令:位逻辑指令 443

SIMATIC LAD指令:移位 / 旋转指令 451

SIMATIC LAD指令:整数数学指令 448

SIMATIC LAD指令:中断指令 449

SIMATIC LAD指令:转换指令 446

SIMATIC LAD指令:字符串指令 451

SIMATIC STL指令: 356

SIMATIC STL指令:比较指令 468

SIMATIC STL指令:表格指令 479

SIMATIC STL指令:程序控制指令 476

SIMATIC STL指令:浮点指令（实数数学） 473

SIMATIC STL指令:计时器指令 479

SIMATIC STL指令:计数器指令 473

SIMATIC STL指令:逻辑指令 475

SIMATIC STL指令:时钟指令 468

SIMATIC STL指令:通讯指令 468

SIMATIC STL指令:位逻辑指令 466

SIMATIC STL指令:移动指令 476

SIMATIC STL指令:移位 / 旋转指令 477

SIMATIC STL指令:整数数学指令 474

SIMATIC STL指令:中断指令 475

SIMATIC STL指令:转换指令 472

SIMATIC: 182, 191, 275, 284, 359, 366, 442, 572, 602

SIMATIC:SIMATIC / 国际助记符 442

SIMATIC:分配符号（全局） 182, 275, 359, 602

SIMATIC:符号表 182, 275, 359, 602

SIMATIC:全局符号 191, 284, 366, 572

SMB0系统状态位 504

SMB136-SMB165

HSC3、HSC4和HSC5高速计数器寄存器 514

SMB166-SMB194 PTO PLS（脉冲）指令概要表  
515

SMB1指令执行状态位 504

SMB200-SMB549智能模块状态 516

SMB28-SMB29模拟调整 507

SMB2自由端口接收字符 504

SMB30和SMB130自由端口控制寄存器 507

SMB31和SMW32永久性内存（EEPROM）写入控制  
508SMB34-SMB35用于定时中断的时间间隔寄存器  
508

SMB36-SMB65

HSC0、HSC1和HSC2高速计数器寄存器 509

SMB3自由端口奇偶校验错误 505

SMB4中断队列溢出、运行时间程序错误、中断启用  
(19)杂啥丝洼 涂粪兄漫(6) 当磺恐? 505

SMB5 I/O错误状态位 505

SMB6 CPU代码寄存器 506

SMB66-SMB85 PTO / PWM高速输出寄存器 511

SMB86-SMB94和SMB186-SMB194接收讯息控制  
512

SMB8-SMB21 I/O模块代码和错误寄存器 506

SMS讯息传送（调制解调器） 144, 652

SMW22-SMW26扫描时间 507

SMW98 I/O扩充总线 - 通讯错误 513

Status:Chart Status 92, 693

STEP 7-Micro/DOS项目 104

STL SIMATIC 指令: 356  
 STL SIMATIC 指令:计数器 473  
 STL SIMATIC 指令:通讯 468  
 STL SIMATIC 指令:位逻辑 466  
 STL SIMATIC指令:比较 468  
 STL SIMATIC指令:表格 479  
 STL SIMATIC指令:程序控制 476  
 STL SIMATIC指令:浮点指令 (实数数学) 473  
 STL SIMATIC指令:计时器 479  
 STL SIMATIC指令:逻辑 475  
 STL SIMATIC指令:时钟 468  
 STL SIMATIC指令:移动 476  
 STL SIMATIC指令:移位 / 旋转 477  
 STL SIMATIC指令:整数数学 474  
 STL SIMATIC指令:指令 356  
 STL SIMATIC指令:中断 475  
 STL SIMATIC指令:转换 472  
 STL SIMATIC指令:字符串指令 478  
 STL编辑器: 20  
 STL编辑器:程序输入项错误 70  
 STL编辑器:地址 61, 358  
 STL编辑器:范例程序 66, 357  
 STL编辑器:举例 453, 464, 480, 490, 500  
 STL编辑器:书签 370  
 STL编辑器:与其他编辑器比较 20  
 STL编辑器:语句 61, 358  
 STL编辑器:注解 62, 372

## - T -

TD 200文字显示: 128, 636  
 TD 200文字显示:TD 200功能 128, 636  
 TD 200文字显示:TD 200向导 (TD 200 v2.1或之前版本) 128, 636  
 TD 200文字显示:TD 200向导 (TD 200 v3.0或TD 200C v1.0) 134, 642

## - U -

Unforce:from Status Chart View 92, 693  
 USS库:USS\_CTRL 735  
 USS库:USS\_INIT 739  
 USS库:USS\_RPM\_D 740  
 USS库:USS\_RPM\_R 740  
 USS库:USS\_RPM\_W 740  
 USS库:USS\_WPM\_D 742  
 USS库:USS\_WPM\_R 742

USS库:USS\_WPM\_W 742  
 USS库:简介 726  
 USS库:样本协议程序 744  
 USS库:执行错误 748

## - V -

V内存 (数据块) 235, 334, 417, 593

## - W -

Windows NT资源对话按钮 544  
 Windows控制面板 (调制解调器) 550

## - ? -

安全回叫 (调制解调器) 144, 652  
 安装 / 移除通讯接口 544  
 安装尺寸 13  
 安装和移除S7-200模块 13  
 版本 (固件) 615  
 帮助 (联机) 11  
 保存作业:在FBD中 60, 355  
 保存作业:在STL中 71, 438  
 保留范围标记一系统块配置 575  
 保留关键字 580  
 保留数据块格式 235, 334, 417, 593  
 保留图解修订中的参数: 177, 263  
 保留图解修订中的参数:FBD中 263  
 保留图解修订中的参数:LAD中 177  
 本地调制解调器 550  
 比较: 20, 23  
 比较:编程模式 (SIMATIC、IEC) 23  
 比较:编辑器 (LAD/FBD/STL) 20  
 比较指令: 444, 456, 468, 483, 493  
 比较指令:IEC (LAD) 483  
 比较指令:IEC (FBD) 493  
 比较指令:SIMATIC (FBD) 456  
 比较指令:SIMATIC (LAD) 444  
 比较指令:SIMATIC (STL) 468  
 闭环回路控制 (PID) 120, 627  
 边缘触发逻辑: 200, 299, 382  
 边缘触发逻辑:中断 (高速计数器) 200, 299, 382  
 编程 (比较) 23  
 编辑程序 (LAD/FBD/STL):错误代码 710  
 编辑程序元素:在FBD中 54, 263

- 编辑程序元素:在LAD中 35, 178  
 编辑程序元素:在STL中 67  
 编辑程序元素:在运行模式中 691  
 编辑器 (LAD/FBD/STL) : 20  
 编辑器 (LAD/FBD/STL) :比较 20  
 编辑指令元素: 54, 263  
 编译CPU程序:PLC信息 (错误) 615  
 编译Micro/WIN程序 708  
 编译Micro/WIN程序: 239, 244, 338, 343, 421, 426, 601, 620, 708  
 编译Micro/WIN程序:编译 (PLC > 编译) 239, 338, 421, 601  
 编译Micro/WIN程序:错误代码 712  
 编译Micro/WIN程序:数据类型检查 191, 284, 366, 572  
 编译Micro/WIN程序:下载 (文件 > 下载) 244, 343, 426, 620  
 编译程序错误 708  
 编译程序错误 (Micro/WIN程序) 708  
 编址 (Modbus协议库) 749  
 编址概况:在PLC内存中 568  
 编址概述: 18  
 编址概述:绝对与符号 18  
 编址概述:在符号和绝对地址视图之间切换 182, 275, 359, 602  
 编址概述:直接与间接 568  
 编址简介:在FBD中 47  
 编址简介:在LAD中 29  
 编址简介:在STL中 61, 358  
 变量 18  
 表格指令: 452, 464, 479, 489  
 表格指令:IEC LAD 489  
 表格指令:SIMATIC FBD 464  
 表格指令:SIMATIC LAD 452  
 表格指令:SIMATIC STL 479  
 波特率 (通讯) 532, 541  
 不分大小写(ASCII) 581  
 布尔输入 443, 455, 466, 570  
 布线指南 15  
 菜单条: 6  
 菜谱向导: 163, 671  
 菜谱向导:向导概述 163, 671  
 参数: 520, 617  
 参数:配置PLC (系统块) 520, 617  
 操作模式: 518, 615, 688  
 操作模式:PLC运行 / 停止模式 518, 688  
 操作模式:读取PLC信息 615  
 操作数数据类型 191, 284, 366, 572  
 测试通讯 72  
 插入程序元素 (LAD/FBD) : 35, 54, 178, 263  
 插入程序元素 (LAD/FBD) :在FBD中 54, 263  
 插入程序元素 (LAD/FBD) :在LAD中 35, 178  
 插入与覆盖模式 26, 44, 268, 271  
 插入与覆盖模式: 61, 358  
 插入与覆盖模式:在FBD中 44, 268, 271  
 插入与覆盖模式:在LAD中 26  
 插入与覆盖模式:在STL中 61, 358  
 插针分配: 545, 550  
 插针分配:调制解调器 550  
 插针分配:网络连接器 545  
 查找: 38  
 查找:在FBD中 58, 287  
 查找:在LAD中 38  
 常量 (数字范围) : 578  
 常量 (数字范围) :举例 578  
 超载指令 (IEC) 191, 284, 366, 442, 572  
 程序: 17  
 程序:保存 (FBD) 60, 355  
 程序:保存 (LAD) 41, 256  
 程序:保存 (STL) 71, 438  
 程序:操作 17  
 程序:从一个段复制至另一个段 103  
 程序:调试和监控状态:调试和监控 74, 247, 346, 429, 521, 678  
 程序:结构 581  
 程序:内存磁带 619  
 程序:下载 73  
 程序:执行有限次数扫描 100, 690  
 程序编辑器 691  
 程序大小 583  
 程序举例: 199, 200, 210, 235, 298, 299, 309, 334, 381, 382, 392, 417, 593  
 程序举例:高速计数器 200, 299, 382  
 程序举例:计数器 199, 298, 381  
 程序举例:脉冲输出 210, 309, 392  
 程序举例:数据快 235, 334, 417, 593  
 程序控制指令: 450, 461, 476, 487, 497  
 程序控制指令:IEC (FBD) 497  
 程序控制指令:SIMATIC (FBD) 461  
 程序控制指令:SIMATIC (LAD) 450  
 程序控制指令:SIMATIC (STL) 476  
 程序控制指令IEC (LAD) 487  
 程序库建立 729  
 程序样本: 235, 334, 417, 593  
 程序指令CPU支持表 442



- 初始化PPI主模式 (NETR/NETW向导) 118, 626
- 初始化和执行时间 749
- 初始化和执行时间:Modbus协议库 749
- 初始化模式 (高速计数器) 200, 299, 382
- 传输速率 545
- 窗口: 6
- 窗口:组件 6
- 从PLC概述上载 102
- 从程序V内存向EEPROM写入 (SMB31- SMB32) 508
- 从程序执行EEPROM写入 (SMB31- SMB32) 508
- 从程序状态视图强制 83
- 从程序状态视图取消强制 83
- 从属设备网络 532
- 从自由端口接收的奇偶校验错误 (SMB3) 505
- 存取: 11
- 存取:查找程序元素 597
- 存取:数据区域的属性 568
- 存取:这是什么? 11
- 错误 708, 711
- 错误: 623
- 错误:DP通讯总线故障 (SMB5) 505
- 错误:FBD程序条目错误 60
- 错误:I/O错误状态 505
- 错误:I/O模块代码和错误寄存器 (SMB8-SMB21) 506
- 错误:LAD/FBD编辑程序错误总结 710
- 错误:LAD程序条目错误 40
- 错误:Modbus从属协议错误 755
- 错误:MODx\_MSG和MODx\_XFR指令 715
- 错误:PLC信息 615
- 错误:STL程序输入项错误 70
- 错误:编译和下载 73
- 错误:编译和运行时间错误 (PLC程序) 712
- 错误:从自由端口接收的奇偶校验错误 (SMB3) 505
- 错误:错误讯息总结 623
- 错误:内存位置 (PLC非严重错误) 714
- 错误:频道和ETHx\_XFR / \_CFG / \_EMAIL / \_FTPC指令的错误代码 718, 720, 721
- 错误:时间标记不匹配 (PC/PLC程序差别) 714
- 错误:数据保留错误 575
- 错误:通讯错误 709
- 错误:位置模块错误 716
- 错误:严重错误 (PLC) 711
- 错误:以太网 / 互联网模块和ETHx\_CTRL子例行程序的错误代码 717
- 错误代码 708
- 打开早期的Micro/WIN和Micro/DOS项目版本 104
- 打印 (文件菜单) 700
- 打印: 11
- 打印:帮助标题 11
- 打印:项目 101
- 大小 (CPU内存中的指令) 583
- 大于指令: 444, 456
- 大于指令:大于实数 444, 456
- 大于指令:大于双整数 444, 456
- 大于指令:大于整数 444, 456
- 大于指令:大于字节 444, 456
- 代码生成 (向导) 116, 118, 120, 624, 626, 627
- 单段PTO操作 210, 309, 392
- 单台主设备PPI网络 541
- 当前值和预设值: 194, 199, 200, 293, 298, 299, 376, 381, 382
- 当前值和预设值:为计时器 194, 293, 376
- 当前值和预设值:用于HSC 200, 299, 382
- 当前值和预设值:用于计数器 199, 298, 381
- 当日时钟时间命令 618
- 等于指令 (比较): 444, 456
- 等于指令 (比较):等于实数 444, 456
- 等于指令 (比较):等于双整数 444, 456
- 等于指令 (比较):等于整数 444, 456
- 等于指令 (比较):等于字节 444, 456
- 递增 / 递减指针 568
- 第一次扫描 100, 690
- 点至点接口 (PPI) 534
- 电感负载 15
- 电位器调整 (SMB28-SMB29) 507
- 电源启动重设命令 620
- 电源预算 12
- 电源中断 (PLC) 575
- 调试工具条 74, 247, 346, 429, 521, 678
- 调用子例行程序 224, 323, 406, 453, 464, 480, 490, 500
- 调用子例行程序: 453, 464, 480, 490, 500
- 调用子例行程序:嵌套 453, 464, 480, 490, 500
- 调制解调器: 550
- 调制解调器:电话调制解调器 550
- 调制解调器:调制解调器连接向导 550
- 调制解调器:调制解调器适配器 550
- 调制解调器:配置本地调制解调器 550
- 调制解调器:配置调制解调器连接 550
- 调制解调器:配置远程调制解调器 550
- 调制解调器:使用PC/PPI电缆进行调制解调器连接 550

- 调制解调器:无线电调制解调器 550  
 调制解调器扩充向导: 144, 652  
 调制解调器扩充向导:EM 241调制解调器模块性能 144, 652  
 调制解调器扩充向导:EM 241调制解调器向导概述 148, 656  
 调制解调器扩充向导:错误代码 715  
 调制解调器模拟 / 手机 550  
 定义局部变量 231, 330, 413, 612  
 定义全局符号 182, 275, 359, 602  
 定制: 592  
 定制:STEP 7-Micro/WIN工作区 592  
 定制:工作区外观 (GS 1.4) 如何: 12  
 读取PLC信息 615  
 队列中断溢出 (SMB4) 505  
 多次扫描 100, 690  
 多点接口 (MPI) 534  
 多段PTO操作 210, 309, 392  
 多台主设备PPI网络 541  
 反转布尔值输入 (FBD) 42  
 范例程序: 224, 229, 323, 328, 406, 411  
 范例程序:中断例行程序 229, 328, 411  
 范例程序:子例行程序 224, 323, 406  
 范围 (CPU内存) 571  
 范围检查 615  
 方框对齐—LAD 174  
 非标准指令 (IEC) 442  
 非法句法: 580  
 非法句法:关键字 580  
 非法语法: 47, 182, 275, 359, 602  
 非法语法:符号名 47  
 非法语法:双引号 182, 275, 359, 602  
 非严重错误 712  
 非严重错误内存位置 714  
 分配:符号—局部变量 182, 275, 359, 602  
 浮点指令 (实数数学): 447, 459, 473  
 浮点指令 (实数数学):SIMATIC (LAD) 447  
 浮点指令 (实数数学):SIMATIC (FBD) 459  
 浮点指令 (实数数学):SIMATIC (STL) 473  
 符号编址: 18  
 符号编址:保留关键字 580  
 符号编址:概述 18  
 符号编址:网络符号信息 189, 282, 610  
 符号表 182, 275, 359, 602  
 符号表:Modbus协议库 749  
 符号表:SIMATIC符号表 / LEC全局变量表 182, 275, 359, 602  
 符号表:复制表格信息 103  
 符号的局部范围 18  
 附加AND (与) / OR (或) 输入 (FBD) 42  
 复杂的网络: 541, 557  
 复杂的网络:范例 541  
 复杂的网络:高级标题 557  
 复制 / 剪切和粘贴: 35, 178  
 复制 / 剪切和粘贴:项目段 103  
 复制 / 剪切和粘贴:在FBD中 54, 263  
 复制 / 剪切和粘贴:在LAD中 35, 178  
 复制 / 剪切和粘贴:在STL中 67  
 赋值: 231, 330, 413, 612  
 赋值:符号-局部变量 231, 330, 413, 612  
 覆盖模式 26, 44, 61, 268, 271, 358  
 覆盖模式:FBD 44, 268, 271  
 覆盖模式:LAD 26  
 覆盖模式:STL 61, 358  
 盖写模式:在STL中 67  
 概述 24  
 概述: 17, 18  
 概述:LAD编程 24  
 概述:调试和监控 74, 247, 346, 429, 521, 678  
 概述:控制程序 17  
 概述:扫描循环 17  
 概述:通讯 72  
 高级标题 (通讯) 557  
 高速计数器: 200, 299, 382  
 高速计数器:HSC0/HSC1/和HSC2寄存器 (SMB36-SMB65) 509  
 高速计数器:HSC3/HSC4/和HSC5高速计数器寄存器 (SMB136-SMB165) 514  
 高速计数器:HSC向导 116, 624  
 高速计数器:当前值和预设值 200, 299, 382  
 高速计数器:模式和输入赋值 200, 299, 382  
 高速计数器:设置动态参数 200, 299, 382  
 高速输出 210, 309, 392  
 工具提示 29, 47, 61, 358  
 工具条 108  
 工作区 6  
 工作区: 12  
 工作区:关于安排的提示 12  
 工作区:组件定义 6  
 功能块图元素—请参阅FBD 42  
 固件/硬件 (PLC支持) 571  
 固件 / 硬件 (PLC支持) 442, 615  
 故障排除 72  
 故障排除:编译和下载 73

- 故障排除:测试通讯 72
- 关键字 (保留) 580
- 关于安排工作区的提示 12
- 国际 / SIMATIC助记符 (比较) 442
- 国际电子技术委员会 (IEC) 23
- 互联网模块: 156, 664
- 互联网模块:CP243-1 IT互联网模块性能 156, 664
- 互联网模块:互联网模块的错误代码 717
- 互联网模块:互联网向导概述 157, 665
- 划线 (连接指令) 26, 44, 268, 271
- 环路控制 (PID): PID调谐控制面板 122, 630
- 环路控制 (PID): 120, 627
- 环路控制 (PID):警报选项 120, 627
- 计时器 (理解S7-200计时器) 194, 293, 376
- 计时器指令: 453, 464, 479, 490, 499
- 计时器指令:IEC FBD 499
- 计时器指令:IEC LAD 490
- 计时器指令:SIMATIC FBD 464
- 计时器指令:SIMATIC LAD 453
- 计时器指令:SIMATIC STL 479
- 计数器指令: 199, 200, 298, 299, 381, 382, 447, 458, 473, 485, 495
- 计数器指令:IEC (LAD) 485
- 计数器指令:IEC (FBD) 495
- 计数器指令:SIMATIC (FBD) 458
- 计数器指令:SIMATIC (LAD) 447
- 计数器指令:SIMATIC (STL) 473
- 计数器指令:理解S7-200计数器指令 199, 298, 381
- 计数器指令:理解高速计数器指令 200, 299, 382
- 间隔更新因子 (网络配置) 557
- 间接和直接编址 568
- 监控状态 74, 247, 346, 429, 521, 678
- 减指令: 473
- 减指令:实数 473
- 剪贴板 (剪切 / 复制 / 粘贴) 108
- 检视: 108
- 检视:检视 / 隐藏符号信息表 108
- 简介:FBD编程 42
- 简介:STL编程 61, 358
- 建立网络 545
- 键组合 (快捷键) 7, 112
- 箭头 7, 112
- 将PLC与PC内存中打开的项目进行比较 618
- 交叉引用和元素用法 240, 339, 422, 597
- 交流电输出 15
- 阶梯逻辑—参阅LAD 24
- 接地指南 15
- 接口安装 / 移除 (通讯) 544
- 接口参数通讯 544
- 接收讯息控制 (SMB86-SMB94和SMB186-SMB194) 512
- 接头插针分配 545
- 进程图像寄存器 17
- 警告 708
- 警告: 708
- 局部变量表: 231, 330, 413, 612
- 局部变量表:ASCII文本文件说明 581
- 局部变量表:编程 231, 330, 413, 612
- 局部变量表:编址 18
- 举例程序: 194, 293, 376
- 举例程序:计时器 194, 293, 376
- 句法 (非法关键字使用) 580
- 绝对地址视图 (符号表 / 全局变量表) 182, 275, 359, 602
- 绝缘指南 15
- 开路功率流指示器 (LAD) 173
- 开路逻辑流指示器 (FBD) 259
- 控制程序元素 520
- 控制寄存器 (PTO/PWM) 210, 309, 392
- 控制面板 (位置模块) 151, 659
- 库简介 726
- 快捷键 7, 112
- 快速参考手册 758
- 扩充模块: 505, 516
- 扩充模块:模块代码和错误寄存器 (SMB8-SMB21) 506
- 扩充模块:模块错误状态 (SMB5) 505
- 扩充模块:智能模块状态 (SMB200-SMB549) 516
- 扩充模式: 615
- 扩充模式:PLC信息 615
- 类型 (PLC菜单) 615
- 类型不匹配 (项目至PLC) 615
- 类型检查数据 191, 284, 366, 572
- 类属指令 (F4/ F6/ F9键): 176, 262
- 类属指令 (F4/ F6/ F9键):FBD中 262
- 类属指令 (F4/ F6/ F9键):LAD中 176
- 累加器 (Modbus) 750
- 理解: 194, 199, 200, 293, 298, 299, 376, 381, 382
- 理解:S7-200计时器指令 194, 293, 376
- 理解:S7-200计数器指令 199, 298, 381
- 理解:高速计数器指令 200, 299, 382
- 立即指令: 42, 263
- 立即指令:保留立即参数 (仅限FBD) 263
- 立即指令:在FBD中立即输入 / 输出 42

- 连接: 174, 200, 259, 299, 382  
 连接:FBD程序中的指令 259  
 连接:LAD程序中的指令 174  
 连接:输入布线(高速计数器) 200, 299, 382  
 了解S7-200网络通讯的基本知识 532  
 令牌轮转时间(通讯) 557  
 逻辑指令(布尔): 449, 460, 475, 486, 496  
 逻辑指令(布尔):IEC(LAD) 486  
 逻辑指令(布尔):IEC(FBD) 496  
 逻辑指令(布尔):SIMATIC(FBD) 460  
 逻辑指令(布尔):SIMATIC(LAD) 449  
 逻辑指令(布尔):SIMATIC(STL) 475  
 脉冲输出 210, 309, 392  
 密码保护:TD 200 128, 134, 636, 642  
 模拟调整(SMB28-SMB29) 507  
 模拟调制解调器 550  
 目录:用户参考手册 108  
 目录:入门指南 4  
 内存: 568, 571  
 内存:22x系列地址范围 571  
 内存:保留范围配置 575  
 内存:磁带(程序) 619  
 内存:非严重错误地址标识符 714  
 内存:类型和属性 568  
 内存:清除PLC内存 623  
 内存:指令大小 583  
 内存L 714  
 配置: 520, 617  
 配置:PLC系统参数 520, 617  
 配置符号表(Modbus) 749  
 偏流和网络电缆终端 545  
 嵌入数据(TD 200 v2.1或之前版本) 128, 636  
 嵌入数据(TD 200 v3.0或TD 200C v1.0向导) 134, 642  
 嵌套子例行程序 224, 323, 406  
 强制: 74, 247, 346, 429, 521, 678  
 强制数值指示灯(SMB4) 505  
 切换反转(FBD) 42  
 切换立即(FBD) 42  
 清除PLC内存 623  
 取消强制: 74, 247, 346, 429, 521, 678  
 全局变量表/符号表 182, 275, 359, 602  
 全局与局部变量 18  
 如何:变更尺寸和去除局部变量表 12  
 如何:检视和去除工作区组件 12  
 如何:纠正编译和下载错误 73  
 如何:使用联机帮助 12  
 如何:在STL中编译 71, 438  
 如何:在窗口标签之间滚动 12  
 如何控制程序作业 17  
 如何在LAD中编译 41, 256  
 入门指南目录 4  
 扫描循环: 17, 518, 688  
 扫描循环:扫描时间(SMW22-SMW26) 507  
 扫描循环:SMB0 504  
 扫描循环:概述 17  
 扫描循环:扫描速率(PLC信息) 615  
 扫描循环:运行/停止模式 518, 688  
 扫描循环:执行单次扫描 100, 690  
 扫描循环:执行多次扫描 100, 690  
 删除程序元素 67  
 删除程序元素: 35, 54, 178, 263  
 删除程序元素:在FBD中 54, 263  
 删除程序元素:在LAD中 35, 178  
 删除程序元素:在STL中 67  
 上下文相关的帮助 11  
 上载(文件>上载) 620  
 设置高速计数器的动态参数 200, 299, 382  
 设置通讯 537  
 设置页面(文件菜单) 700  
 生成程序代码(向导): 116, 118, 120, 624, 626, 627  
 生成程序代码(向导):HSC 116, 624  
 生成程序代码(向导):NETR/NETW 118, 626  
 生成程序代码(向导):PID 120, 627  
 时间标记不匹配(PC/PLC程序差别) 714  
 时钟设置(当日时间) 618  
 时钟指令: 444, 455, 468, 482, 492  
 时钟指令:IEC(LAD) 482  
 时钟指令:IEC(FBD) 492  
 时钟指令:SIMATIC(FBD) 455  
 时钟指令:SIMATIC(LAD) 444  
 时钟指令:SIMATIC(STL) 468  
 实数转换指令: 191, 284, 366, 572  
 实数转换指令:实数常量(数字范围) 191, 284, 366, 572  
 使用Modbus从属协议指令 750  
 使用Modbus协议的要求 749  
 使用Modbus协议库 749  
 使用USS协议指令 735  
 使用向导编程 24  
 收集状态 74, 247, 346, 429, 521, 678  
 手机调制解调器 550  
 首次扫描 504

- 输入 26, 44, 61, 268, 271, 358
- 输入: 42
- 输入:FBD指令 44, 268, 271
- 输入:LAD指令 26
- 输入:STL语句 61, 358
- 输入:反转FBD输入 42
- 输入:使FBD输入或输出成为立即 42
- 输入:增加或删除附加输入 42
- 数据: 235, 334, 417, 593
- 数据:保留 575
- 数据:常量 578
- 数据:初始程序 235, 334, 417, 593
- 数据:类型 191, 284, 366, 572
- 数据块: 708
- 数据块:保留格式 235, 334, 417, 593
- 数据块:编译错误 708
- 数据快: 235, 334, 417, 593
- 数据快:建立 235, 334, 417, 593
- 数据类型 (SIMATIC and IEC 1131-3) 191, 284, 366, 572
- 数据类型定义 231, 330, 413, 612
- 数据区域的属性 568
- 数学 486
- 数学: 496
- 数学:IEC数学指令 (FBD) 496
- 数学:IEC数学指令 (LAD) 486
- 数学:SIMATIC浮点数学 (FBD) 459
- 数学:SIMATIC浮点数学 (LAD) 447
- 数学:SIMATIC浮点数学 (STL) 473
- 数学:SIMATIC整数数学 (FBD) 459
- 数学:SIMATIC整数数学 (LAD) 448
- 数学:SIMATIC整数数学 (STL) 474
- 数字指令 (IEC): 487, 497
- 数字指令 (IEC):IEC (LAD) 487
- 数字指令 (IEC):IEC (FBD) 497
- 双字地址属性 568, 570
- 所有编辑器中的书签 370
- 锁定: 74, 247, 346, 429, 521, 678
- 锁定:程序状态视图强制和取消强制 83
- 替换: 38
- 替换:在FBD中 58, 287
- 替换:在LAD中 38
- 替换:在STL中 69, 370
- 停止模式 518, 688
- 停止模式:PLC菜单 518, 688
- 停止模式:写入和强制输出 519
- 通讯: 72
- 通讯:PROFIBUS 534
- 通讯:安装 / 移除接口 544
- 通讯:测试 72
- 通讯:错误代码 709
- 通讯:点至点接口 (PPI) 534
- 通讯:端口状态 615
- 通讯:多点接口 (MPI) 534
- 通讯:概述 72
- 通讯:配置 534
- 通讯:设置对话框 534
- 通讯:下载程序 73
- 通讯:选择 534
- 通讯:硬件配置表 537
- 通讯指令: 444, 455, 468, 483, 493
- 通讯指令:IEC (LAD) 483
- 通讯指令:IEC (FBD) 493
- 通讯指令:SIMATIC (FBD) 455
- 通讯指令:SIMATIC (LAD) 444
- 通讯指令:SIMATIC (STL) 468
- 网络 (程序段): 44, 103, 189, 282, 610
- 网络 (程序段):FBD 44
- 网络 (程序段):复制项目段 103
- 网络 (程序段):网络符号信息表 189, 282, 610
- 网络 (通讯) 72
- 网络 (通讯): 72, 534
- 网络 (通讯):建立网络 545
- 网络 (通讯):偏流和网络电缆终端 545
- 网络 (通讯):网络地址 532
- 网络 (通讯):网络读取 / 网络写入向导 118, 626
- 网络 (通讯):网络配置 534
- 网络 (通讯):网络配置范例 541
- 网络 (通讯):选择网络电缆 545
- 网络配置范例 541
- 网络配置范例:仅使用S7-200设备 541
- 网络配置范例:使用PROFIBUS-DP 541
- 网络配置范例:使用S7-200/S7-300/S7-400设备 541
- 为HSC设置当前值和预设值 200, 299, 382
- 位存取: 240, 339, 422, 570, 597
- 位存取:内存类型和属性 568
- 位存取:位存取 570
- 位存取:位用法表 (程序用法) 240, 339, 422, 597
- 位逻辑指令 443
- 位逻辑指令: 443, 455, 466, 482, 492
- 位逻辑指令:IEC (LAD) 482
- 位逻辑指令:IEC (FBD) 492
- 位逻辑指令:SIMATIC (FBD) 455
- 位逻辑指令:SIMATIC (LAD) 443

- 位逻辑指令:SIMATIC (STL) 466
- 位置控制向导: 148, 656
- 位置控制向导:EM 253位置控制面板 151, 659
- 位置控制向导:EM 253位置模块性能 148, 656
- 位置控制向导:错误代码 716
- 位置控制向导:向导概述 149, 657
- 文件 > 上载命令 620
- 文件 > 下载命令 244, 343, 426, 620
- 文件I/O错误 711
- 无效符号名 29, 47, 70
  
- 西门子 (Siemens) 公司提供的库 (USS和Modbus) 726
- 系统块 (配置) 520, 617
- 系统状态位 (SMB0) 504
- 下载 (文件 > 下载) 244, 343, 426, 620
- 下载程序 73
- 现用状态 (高速计数器) 200, 299, 382
- 线圈 (LAD) 520
- 向导:CP243-1 IT互联网向导概述 157, 665
- 向导:CP243-1 IT互联网性能 156, 664
- 向导:CP243-1 以太网向导概述 155, 663
- 向导:CP243-1 以太网性能 154, 662
- 向导:CP243-2 AS-i模块性能 162, 670
- 向导:CP243-2 AS-i向导概述 162, 670
- 向导:EM 241调制解调器模块性能 144, 652
- 向导:EM 241调制解调器向导概述 148, 656
- 向导:EM 253位置模块性能 148, 656
- 向导:HSC 116, 624
- 向导:NETR/NETW 118, 626
- 向导:PID 120, 627
- 向导:TD 200 (TD 200 v3.0或TD 200C v1.0) Wizard 134, 642
- 向导:TD 200功能 128, 636
- 向导:TD 200向导 (TD 200 v2.1或之前版本) 128, 636
- 向导:菜谱 163, 671
- 向导:调制解调器连接向导 550
- 向导:数据日志 167, 675
- 向导:位置模块向导概述 149, 657
- 项目: 20
- 项目:保存 (FBD) 60, 355
- 项目:保存 (LAD) 41, 256
- 项目:保存 (STL) 71, 438
- 项目:创建 (STL) 61
- 项目:打开早期的Micro/WIN和Micro/DOS项目版本 104
- 项目:打印 (文件菜单) 700
- 项目:复制 102
- 项目:建立 (FBD) 41
- 项目:建立 (LAD) 24
- 项目:移动 102
- 项目:邮寄 102
- 项目:与PLC进行比较 618
- 项目:元件概述 20
- 项目:之间复制 103
- 项目:重新命名 102
- 项目文件I/O错误 711
- 小于指令 (SIMATIC LAD/FBD): 444, 456
- 小于指令 (SIMATIC LAD/FBD):小于实数 444, 456
- 小于指令 (SIMATIC LAD/FBD):小于双整数 444, 456
- 小于指令 (SIMATIC LAD/FBD):小于整数 444, 456
- 小于指令 (SIMATIC LAD/FBD):小于字节 444, 456
- 选择PLC类型 615
- 寻呼 (调制解调器) 144, 652
- 严重错误 711
- 严重错误 (PLC) 711
- 页脚 (打印机页面设置) 700
- 页面设置 (文件菜单) 700
- 页首 (打印机页面设置) 700
- 一击 (计时器) 194, 293, 376
- 移除: 108, 268, 544
- 移除:书签 108
- 移除:输入 (FBD工具条按钮) 268
- 移除:通讯接口 544
- 移除:通讯硬件 544
- 移除PLC类型 615
- 移动指令: 450, 461, 476, 486, 496
- 移动指令:IEC (FBD) 496
- 移动指令:IEC (LAD) 486
- 移动指令:SIMATIC (FBD) 461
- 移动指令:SIMATIC (LAD) 450
- 移动指令:SIMATIC (STL) 476
- 移位 / 旋转指令: 451, 462, 477, 488, 498
- 移位 / 旋转指令:IEC (FBD) 498
- 移位 / 旋转指令:IEC (LAD) 488
- 移位 / 旋转指令:SIMATIC (FBD) 462
- 移位 / 旋转指令:SIMATIC (LAD) 451
- 移位 / 旋转指令:SIMATIC (STL) 477
- 以太网模块: 154, 662
- 以太网模块:CP243-1以太网性能 154, 662

- 以太网模块:频道和ETHx\_XFR / \_CFG / \_EMAIL / \_FTPC指令的错误代码 718, 720, 721
- 以太网模块:以太网 / 互联网模块和ETHx\_CTRL指令的错误代码 717
- 以太网模块:以太网通讯 562
- 以太网模块:以太网向导概述 155, 663
- 抑制电路指南 15
- 应用程序用户参考手册 108
- 硬件接口安装和移除 (通讯) 544
- 硬件配置表 537
- 硬件支持 18, 442, 615
- 用户参考手册 108
- 用户定义库: 726
- 用户定义库:指导 729
- 用于图形编辑器的指令参数: 175, 261
- 用于图形编辑器的指令参数:FBD中 261
- 用于图形编辑器的指令参数:LAD中 175
- 有效操作数 191, 284, 366, 572
- 语句表—参阅STL 66, 357
- 语句表程序范例 66, 357
- 元素用法 240, 339, 422, 597
- 远程调制解调器 550
- 远程站址 532
- 允许的操作数 191, 284, 366, 572
- 运行模式 (PLC菜单) 518, 688
- 运行模式编辑 691
- 运行时间程序错误 (SMB4) 505
- 运行时间和PLC编译错误 712
- 在LAD中保存作业 41, 256
- 在STL中查找 69, 370
- 在编辑器中显示程序状态 83
- 在停止模式中强制和写入输出 519
- 在停止模式中写入和强制输出 519
- 在网络中使用调制解调器和STEP 7-Micro/WIN 550
- 在运行模式中编辑 691
- 在子例行程序中递归 224, 323, 406
- 早期的Micro/WIN、Micro/DOS项目版本 104
- 增加指令:增加输入 (FBD工具条按钮) 267
- 整数数学指令总结: 448, 459, 474
- 整数数学指令总结:SIMATIC (LAD) 448
- 整数数学指令总结:SIMATIC (FBD) 459
- 整数数学指令总结:SIMATIC (STL) 474
- 正交模式 (HSC) 200, 299, 382
- 执行和初始化时间 749
- 执行和初始化时间:Modbus协议库 749
- 执行有限次数扫描 100, 690
- 执行状态 74, 247, 346, 429, 521, 678
- 直接I/O存取: 42, 263
- 直接I/O存取:保留FBD中的参数 263
- 直接I/O存取:在FBD中输入 / 输出 42
- 直接和间接编址 568
- 直流电输出 15
- 指定: 29
- 指定:地址 29
- 指令:整数数学指令 459
- 指令树: 6
- 指令执行状态位 (SMB1) 504
- 指令总结: 442
- 指令总结:IEC超载和非标准 442
- 指令总结:S7-200 CPU型号提供的支持 442
- 指针数值 191, 284, 366, 572
- 指针值 568
- 智能I/O模块状态 (SMB200-SMB549) 516
- 中断例行程序:
- 用于定时中断的时间间隔寄存器 (SMB34-SMB35) 508
- 中断例行程序:建立中断例行程序 229, 328, 411
- 中断例行程序:全局中断启用状态 (SMB4) 505
- 中断例行程序:中断队列溢出 (SMB4) 505
- 中断指令: 449, 460, 475, 485, 495
- 中断指令:IEC (FBD) 495
- 中断指令:SIMATIC (IEC) 485
- 中断指令:SIMATIC (FBD) 460
- 中断指令:SIMATIC (LAD) 449
- 中断指令:SIMATIC (STL) 475
- 中继器 545
- 终端块接头 (移除和重新安装) 13
- 重设CPU硬件 620
- 重设尺寸: 231, 330, 413, 612
- 重设尺寸:局部变量表 231, 330, 413, 612
- 主菜单 108
- 主地址 (DP通讯) 615
- 主设备网络 532
- 助记符: 442
- 助记符:SIMATIC和国际 442
- 注解 (项目和程序): 32, 51, 289
- 注解 (项目和程序):在FBD中 51, 289
- 注解 (项目和程序):在LAD中 32
- 注解 (项目和程序):在STL中 62, 372
- 转换指令: 446, 457, 472, 484, 494
- 转换指令:IEC (LAD) 484
- 转换指令:IEC (FBD) 494
- 转换指令:SIMATIC (FBD) 457
- 转换指令:SIMATIC (LAD) 446

- 转换指令:SIMATIC (STL) 472  
 转入: 38, 58, 69, 287, 370  
 转入:在FBD中 58, 287  
 转入:在LAD中 38  
 转入:在STL中 69, 370  
 状态: 74, 247, 346, 429, 521, 678  
 状态:程序状态 83  
 状态:复制图信息 103  
 状态:执行有限次数扫描 100, 690  
 状态:执行状态 83  
 状态错误(时间标记不匹配) 714  
 状态条: 6  
 状态图: 6  
 资源对话按钮(Windows NT) 544  
 子例行程序: 224, 323, 406  
 子例行程序:如何调用 224, 323, 406  
 子例行程序:如何建立 224, 323, 406  
 子例行程序:如何终止 224, 323, 406  
 字符串指令: 451, 463, 478, 488, 498  
 字符串指令:IEC (FBD) 498  
 字符串指令:IEC (LAD) 488  
 字符串指令:SIMATIC (LAD) 451  
 字符串指令:SIMATIC (FBD) 463  
 字符串指令:SIMATIC (STL) 478  
 字符集 47  
 字符设置 29  
 字节存取: 570  
 字节存取:不等于指令(比较) 468  
 字节存取:大于指令(比较) 468  
 字节存取:等于指令(比较) 468  
 字节存取:小于指令(比较) 468  
 字节存取:字节/字/双字存取 570  
 字节存取:字节用法表(程序用法) 240, 339, 422, 597  
 字指令:内存类型和属性 568  
 自动调谐(PID) 122, 630  
 自动功率流连接(LAD) 174  
 自动逻辑流连接(FBD) 259  
 自由端口模式 545  
 自由端口模式:自由端口接收字符(SMB2) 504  
 自由端口模式:自由端口奇偶校验错误(SMB3) 505  
 自由端口模式:SMB86-SMB94和SMB186-SMB194接收讯息控制 512  
 自由端口模式:自由端口控制寄存器(SMB30和SMB31) 507  
 自由端口模式:自由模式传输器闲置(SMB4) 505  
 组合分支连接(FBD) 259  
 最高站址(HAS) 557



Endnotes 2... (after index)

获取更多资料 微信搜索蓝领星球

Back Cover

获取更多资料 微信搜索蓝领星球