

# SIEMENS

## SIMATIC

### 用于S7-300/400 系统和标准功能的系统软件

参考手册

2004年1月版  
A5E00446508-01

前言, 内容

组织块	1
SFC的公共参数	2
复制功能和块功能	3
用于控制程序执行的SFC	4
用于处理系统时钟的SFC	5
用于处理运行系统计时器的SFC	6
用于传送数据记录的SFC	7
符合PNO AK 1131的DPV1 SFB	8
用于处理时间中断的SFC	9
用于处理延时中断的SFC	10
用于处理同步错误的SFC	11
用于处理中断和异步错误的SFC	12
用于诊断的SFC	13
用于更新过程映像和处理位域的SFC和S FB	14
用于寻址模块的系统功能	15
用于分布式I/O的SFC	16
用于全局数据通讯的SFC	17
S7通讯和S7基本通讯的概述	18
S7通讯	19
用于未组态S7连接的通讯SFC	20
PROFINET	21
生成与块相关的消息	22
IEC定时器和IEC计数器	23
IEC功能	24
用于集成控制的SFB	25
用于紧凑型CPU的SFB	26
用于H CPU的SFC	27
集成功能 (对于带集成I/O的CPU)	28
塑料技术	29
诊断数据	30
系统状态列表(SSL)	31
事件	32
SFC和SFB列表	33
书目, 词汇表, 索引	

## 安全指南

本手册包括了保证人身安全与保护本产品及连接的设备所应遵守的注意事项。这些注意事项在手册中均以如下符号加以突出，并根据危险等级标明如下：



---

### 危险

表示如果不采取适当的预防措施，将导致死亡、严重的人身伤害或财产损失。

---



---

### 警告

表示如果不采取适当的预防措施，可能导致死亡、严重的人身伤害或财产损失。

---



---

### 当心

表示如果不采取适当的预防措施，可能导致轻微的人身伤害。

---

---

### 当心

表示如果不采取适当的预防措施，可能导致财产损失。

---

---

### 须知

提醒您注意有关产品、产品使用的特别重要的信息，或者是文档的特定部分。

---

## 合格人员

只有**合格人员**才允许安装和操作该设备。合格人员是指被授权按照既定安全惯例和标准，对线路、设备和系统进行调试、接地和标记的人员。

## 正确使用

请注意如下事项：



---

### 警告

该设备及其部件只能用于产品目录或技术说明书中所描述的范畴，并且只能与由西门子公司认可或推荐的第三方厂商提供的设备或部件一起使用。

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

---

## 商标

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

本文档中的其它一些标志也是注册商标，如果任何第三方出于个人目的而使用，都会侵犯商标所有者的权利。

### 版权所有 © Siemens AG 2004 保留所有权利

未经明确的书面许可，不得复制、传播或使用本手册或所含内容。违者应对造成的损失承担责任。保留所有权利，包括实用新型或设计的专利许可权及注册权。

Siemens AG  
Bereich Automation and Drives  
Geschäftsgebiet Industrial Automation Systems  
Postfach 4848, D- 90327 Nuernberg

Siemens Aktiengesellschaft

### 免责声明

我们已检查过本手册中的内容与所述硬件和软件相符。由于差错在所难免，我们不能保证完全一致。我们会定期审查本手册中的内容，并在后续版本中进行必要的更正。欢迎提出改进意见。

©Siemens AG 2004  
技术数据如有改动，恕不另行通知。

A5E00446508-01

# 前言

## 用途

本手册综述了在S7-300和S7-400的CPU的操作系统中所包含的组织块(OB)、系统功能(SFC)、系统和标准功能块(SFB)及IEC功能。附录部分描述了诊断数据、系统状态列表(SZL)和事件。

---

### 注意

请参考下列文件中的参考书目 - “S7-300自动化系统CPU规范: CPU 31xC与CPU 31x”和“S7-300自动化系统CPU规范: CPU 312IFM - 318-2 DP” /70/, 或“自动化系统S7-400: CPU规范”参考手册/101/, 或指令列表: S7-400可编程控制器/102/(适用于您的CPU的任意版本), 以了解各个CPU上可以使用的功能和块的详细资料。特定CPU的CFB属性和S7发信功能在/70/和 [/101/](#)。

---

欲了解有关CPU操作系统、程序设计及CPU的通讯和诊断能力的内容, 请参见“配置硬件和通讯连接STEP 7 V5.3”手册/234/关于如何在程序中调用功能和功能块的说明参见语言描述部分。

使用STEP 7标准软件为所有这些功能编程, 并分配参数。关于如何使用本软件的说明请参见“使用STEP 7 V5.3编程”手册 /231/ 和STEP 7在线帮助。

## 目标用户

本手册的目标读者为: 熟悉控制过程、负责为可编程逻辑控制器编写程序的程序员和工程师。

**STEP 7文档数据包**

下表显示了STEP 7文档的总览：

文档	用途	订货号
STEP 7基础信息 <ul style="list-style-type: none"> <li>• STEP 7 V5.3, 使用入门手册</li> <li>• 使用STEP 7 V5.3编程</li> <li>• 配置硬件与通讯连接, STEP 7 V5.3</li> <li>• 从S5到S7, 变频器手册</li> </ul>	提供给技术人员的基础信息, 描述了使用STEP 7和S7-300/400可编程控制器来实现控制任务的方法。	6ES7810-4CA07-8BW0
STEP 7参考书目 <ul style="list-style-type: none"> <li>• 用于S7-300/400的梯形图(LAD)/功能块图(FBD)/语句表(STL)手册</li> <li>• S7-300/400 的标准函数及系统函数</li> </ul>	提供了参考信息, 并描述了编程语言LAD、FBD、STL、标准函数以及系统函数, 扩充了STEP 7基础信息的范围。	6ES7810-4CA07-8BW1

在线帮助	用途	订货号
STEP 7帮助	以在线帮助的形式, 提供了使用STEP 7进行编程和组态硬件的基础信息。	STEP 7标准软件中的一部分。
STL/LAD/FBD帮助参考 SFB/SFC帮助参考 组织块帮助参考	上下文相关参考信息。	STEP 7标准软件中的一部分。

## 在线帮助

集成于软件中的在线帮助是对本手册的补充。提供在线帮助的目的在于，在使用该软件时提供详细的支持。

该帮助系统通过一些界面集成于软件中：

- 在**帮助**菜单中提供了多个菜单命令：**内容**命令打开关于STEP 7帮助信息的索引。
- **使用帮助**提供了有关使用在线帮助的详细指示。
- 上下文相关帮助提供关于当前语境的信息，例如，打开的对话框或激活的窗口。通过单击“帮助”按钮或按下F1键可以打开上下文相关的帮助。
- 状态栏提供了另一种形式的上下文相关帮助。当把鼠标指针放在菜单命令上时，它会显示每个菜单命令的简要解释。
- 当把鼠标指针放在工具栏的各个图标上时，一会之后，它也显示该图标的简要解释。

如果您更喜欢以打印格式阅读在线帮助的信息，则可以将单个的帮助主题、书本或整个在线帮助打印出来。

本手册是从基于HTML的STEP 7帮助中摘录下来的。由于该手册和在线帮助的结构几乎一致，因而可以很容易地在该手册和在线帮助间进行切换。

## 对文档的反馈

为了帮助我们为您和STEP 7未来的用户提供尽可能最好的文档，我们需要您的支持。如果您有任何关于本手册或在线帮助的批评或建议，请填妥位于本手册最后的调查表，并发送到所提供的地址。请注明您本人对文档的等级评价。

## 其它手册

下列手册对各种S7-300 CPU和S7-400 CPU以及S7-300模块和S7-400模块进行了描述：

- 欲了解S7-300可编程逻辑控制器的内容，请参考手册：“Automatisierungssystem S7-300 CPU-Daten: CPU 31xC und CPU 31x” und “Automatisierungssystem S7-300 CPU-Daten: CPU 312 IFM bis 318-2 DP” /70/、“S7-300、M7-300可编程控制器模块规范”/71/和指令列表/72/中。
- 欲了解S7-400可编程逻辑控制器的内容，请参考手册：“Automatisierungssystem S7-400: CPU Daten”/101/和指令列表/102/中。

### 如何使用本手册

本手册包含了下列主题：

- 第1章说明所有组织块的功能。
- 第2章描述公用参数RET\_VAL、REQ和BUSY。
- 第3章至第29章描述SFC、SFB和IEC-FC。
- 第30章至第33章包含对诊断数据结构的描述、对本手册中描述的SZL-ID、可能的事件、SFC列表、SFB和FC的概述，以及对SDB的概述。
- 参考书目包含了更多其它手册的列表。
- 词汇表解释了重要的术语。
- 索引帮助您迅速地定位文本部分和主题部分。

### 约定

对其它手册和文档的引用用两个斜线/.../加数字表示。这些数字指的是参考书目中列出的手册的标题。

### 特殊注意事项

系统功能可能被中断。如果有任何适用于某些SFC或某些情况的限制，在特定的SFC的描述中有解释。

### 更多支持

如果有任何技术问题，请联系西门子代表或代理商。

您可以在如下网页中查找联系人：

<http://www.siemens.com/automation/partner>

### 培训中心

西门子提供了很多培训教程，帮助您熟悉SIMATIC S7自动化系统。请联系当地的培训中心，或位于德国纽伦堡(D 90327)的培训总部，以获取详细信息。

电话： +49(911) 895-3200。

<http://www.sitrain.com/>

**A&D技术支持**

遍布世界各处，24小时服务：



<b>全球(纽伦堡)</b> <b>技术支持</b>  每年365天，每天24小时 电话： +49(180) 5050-222 传真： +49(180) 5050-223 电子邮件： adsupport@siemens.com  格林威治 标准时间： +1:00		
<b>欧洲/非洲(纽伦堡)</b> <b>许可证</b>  当地时间： 周一至周五，8:00 - 5:00 PM 电话： +49(180) 5050-222 传真： +49(180) 5050-223 电子邮件： adsupport@siemens.com  格林威治 标准时间： +1:00	<b>美国(约翰逊城)</b> <b>技术支持和授权</b>  当地时间： 周一至周五，8:00 - 5:00 PM 电话： +1(423) 262 2522 传真： +1(423) 262 2289 电子邮件： simatic.hotline@sea.siemens.com  格林威治 标准时间： -5:00	<b>亚洲/澳洲(北京)</b> <b>技术支持和授权</b>  当地时间： 周一至周五，8:00 - 5:00 PM 电话： +86 10 64 75 75 75 传真： +86 10 64 74 74 74 电子邮件： adsupport.asia@siemens.com  格林威治 标准时间： +8:00
SIMATIC热线以及授权热线所使用的语言通常为德语和英语。		

## Internet服务和支特

除文档以外，还在Internet上在线提供了知识产权信息，网址如下：

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

可在其中查找如下内容：

- 公司简讯，经常提供产品的最新信息。
- 相应文档资料，可通过“服务和支特”中的搜索功能查找。
- 论坛，世界各地的用户和专家可以在此交流经验。
- 当地自动化和驱动办事处。
- 在“服务”页面下提供了关于现场服务、维修、备件等信息。



# 目录

<b>1</b>	<b>组织块</b>	<b>1-1</b>
1.1	组织块(OB)概述 .....	1-1
1.2	程序循环组织块(OB1) .....	1-4
1.3	时钟中断组织块 (OB10到OB17) .....	1-6
1.4	时间延迟中断组织块 (OB20到OB23) .....	1-10
1.5	周期性中断组织块 (OB30到OB38) .....	1-12
1.6	硬件中断组织块 (OB40到OB47) .....	1-14
1.7	状态中断OB(OB 55) .....	1-16
1.8	更新中断OB(OB 56) .....	1-17
1.9	制造商特定中断OB(OB57) .....	1-18
1.10	多值计算中断组织块(OB60) .....	1-19
1.11	同步周期性中断OB(OB 61到OB 64) .....	1-21
1.12	I/O冗余出错OB(OB70) .....	1-22
1.13	CPU冗余出错OB(OB72) .....	1-24
1.14	通讯冗余出错OB(OB73) .....	1-27
1.15	时间出错组织块(OB80) .....	1-28
1.16	电源出错组织块(OB81) .....	1-30
1.17	诊断中断组织块(OB82) .....	1-32
1.18	插入/删除模块中断组织块(OB83) .....	1-34
1.19	CPU硬件故障组织块(OB84) .....	1-37
1.20	优先级出错组织块(OB85) .....	1-38
1.21	机架故障组织块(OB86) .....	1-42
1.22	通讯出错组织块(OB87) .....	1-45
1.23	处理中断OB(OB 88) .....	1-47
1.24	后台组织块(OB90) .....	1-48
1.25	启动组织块(OB100、OB101和OB102) .....	1-50
1.26	编程出错组织块(OB121) .....	1-55
1.27	I/O访问出错组织块(OB122) .....	1-57
<b>2</b>	<b>SFC的公共参数</b>	<b>2-1</b>
2.1	通过输出参数RET_VAL判断出错 .....	2-1
2.2	异步SFC的REQ、RET_VAL和BUSY参数的含义 .....	2-4

<b>3</b>	<b>复制功能和块功能</b>	<b>3-1</b>
3.1	使用SFC20 “BLKMOV”复制存储区域 .....	3-1
3.2	使用SFC81 “UBLKMOV”不间断地复制变量 .....	3-4
3.3	使用SFC21 “FILL”初始化存储区 .....	3-6
3.4	使用SFC22 “CREAT_DB”创建数据块.....	3-8
3.5	使用SFC23 “DEL_DB”删除数据块.....	3-10
3.6	使用SFC24 “TEST_DB”测试数据块.....	3-12
3.7	使用SFC25 “COMPRESS”压缩用户存储器 .....	3-13
3.8	使用SFC44 “REPL_VAL”传送一个替换值到累加器1 .....	3-15
3.9	使用SFC82 “CREA_DBL”在装载存储器中生成数据块 .....	3-16
3.10	使用SFC83 “READ_DBL”从装载存储器的数据块中读取 数据.....	3-19
3.11	使用SFC84 “WRIT_DBL”在装载存储器中写入数据块 .....	3-21
3.12	使用SFC85 “CREA_DB”创建数据块.....	3-23
<b>4</b>	<b>用于控制程序执行的SFC</b>	<b>4-1</b>
4.1	使用SFC43 “RE_TRIGR”重新触发循环时间监视 .....	4-1
4.2	使用SFC46 “STP”将CPU切换为STOP .....	4-1
4.3	使用SFC47 “WAIT”延时用户程序执行 .....	4-2
4.4	使用SFC35 “MP_ALM”触发多处理器中断 .....	4-3
4.5	使用SFC104 “CiR”控制CiR.....	4-4
<b>5</b>	<b>用于处理系统时钟的SFC</b>	<b>5-1</b>
5.1	使用SFC0 “SET_CLK”设定TOD .....	5-1
5.2	使用SFC1 “READ_CLK”读取时间 .....	5-2
5.3	使用SFC48 “SNC_RTCB”同步子时钟 .....	5-3
5.4	使用SFC100 “SET_CLKS”设定时间日期和TOD状态 .....	5-4
<b>6</b>	<b>用于处理运行系统计时器的SFC</b>	<b>6-1</b>
6.1	运行时间定时器.....	6-1
6.2	使用SFC101 “RTM”控制运行时间定时器.....	6-2
6.3	使用SFC2 “SET_RTM”设置运行时间定时器.....	6-4
6.4	使用SFC3 “CTRL_RTM”启动和停止运行时间定时器 .....	6-5
6.5	使用SFC4 “READ_RTM”读取运行时间定时器.....	6-6
6.6	使用SFC64 “TIME_TCK”读取系统时间 .....	6-7
<b>7</b>	<b>用于传送数据记录的SFC</b>	<b>7-1</b>
7.1	读写一条数据记录 .....	7-1
7.2	使用SFC54 “RD_DPARM”读取定义的参数 .....	7-3
7.3	使用SFC102 “RD_DPARA”读取预定义参数 .....	7-4
7.4	使用SFC55 “WR_PARM”写动态参数 .....	7-5
7.5	使用SFC56 “WR_DPARM”写一条数据记录.....	7-7
7.6	用SFC57 “PARM_MOD” 将参数分配给模块.....	7-8
7.7	使用SFC58 “WR_REC”写入数据记录.....	7-11
7.8	使用SFC59 “RD_REC”读一个数据记录.....	7-13
7.9	SFC 55到59的进一步出错信息 .....	7-18

<b>8</b>	<b>符合PNO AK 1131的DPV1 SFB</b>	<b>8-1</b>
8.1	使用SFB “RDREC”从DP从站读一个数据记录 .....	8-1
8.2	使用SFB53“WRREC”将数据记录写入DP从站 .....	8-3
8.3	用SFB54“RALRM” STATUS[3]从DP从站接收中断 .....	8-5
8.4	用SFB75“SALRM”向DP主站发送中断 .....	8-14
<b>9</b>	<b>用于处理时钟中断的SFC</b>	<b>9-1</b>
9.1	处理时钟中断 .....	9-1
9.2	SFC 28到31的特征 .....	9-2
9.3	使用SFC28 “SET_TINT”设置日时钟中断.....	9-4
9.4	使用SFC29 “CAN_TINT”取消时钟中断.....	9-5
9.5	使用SFC30 “ACT_TINT”激活时钟中断.....	9-6
9.6	使用SFC31 “QRY_TINT”查询日时钟中断.....	9-7
<b>10</b>	<b>用于处理延时中断的SFC</b>	<b>10-1</b>
10.1	处理延时中断 .....	10-1
10.2	使用SFC32 “SRT_DINT”启动延时中断.....	10-3
10.3	使用SFC34 “QRY_DINT”查询日时钟中断 .....	10-4
10.4	使用SFC33 “CAN_DINT”取消延时中断 .....	10-5
<b>11</b>	<b>用于处理同步出错的SFC</b>	<b>11-1</b>
11.1	屏蔽同步出错 .....	11-1
11.2	使用SFC36 “MSK_FLT”屏蔽同步出错 .....	11-10
11.3	使用SFC37 “DMSK_FLT”解除屏蔽同步出错 .....	11-11
11.4	使用SFC38 “READ_ERR”读取出错寄存器 .....	11-12
<b>12</b>	<b>用于处理中断和异步出错的SFC</b>	<b>12-1</b>
12.1	延迟和禁用中断和异步出错.....	12-1
12.2	使用SFC39 “DIS_IRT”禁用新中断和异步出错的处理 .....	12-3
12.3	使用SFC40 “EN_IRT”启用新中断和异步出错的处理 .....	12-5
12.4	使用SFC41 “DIS_AIRT”延迟更高优先级中断和异步出错的处理 .....	12-7
12.5	使用SFC42 “EN_AIRT”启用处理较高优先级的中断和异步出错 .....	12-8
<b>13</b>	<b>用于诊断的SFC</b>	<b>13-1</b>
13.1	系统诊断 .....	13-1
13.2	使用SFC6 “RD_SINFO”读取OB启动信息.....	13-1
13.3	使用SFC51 “RDSYSST”读取系统状态列表或部分列表.....	13-4
13.4	使用SFC52 “WR_USMSG”将自定义诊断事件写入诊断缓冲区 .....	13-10
13.5	使用SFC78 “OB_RT”确定OB程序运行时间.....	13-14
13.6	使用SFC87 “C_DIAG”诊断当前连接状态.....	13-18
13.7	使用SFC103 “DP_TOPOL”识别DP主站系统的总线拓扑 .....	13-23
<b>14</b>	<b>用于更新过程映像和处理位域的SFC和SFB</b>	<b>14-1</b>
14.1	使用SFC26 “UPDAT_PI”更新过程映像输入表.....	14-1
14.2	使用SFC27 “UPDAT_PO”更新过程映像输出表 .....	14-3
14.3	使用SFC79 “SET”在I/O区域中设置位域 .....	14-5
14.4	使用SFC 126 “SYNC_PI”在同步循环中更新过程映像分区输入表 .....	14-6
14.5	使用SFC 127 “ISO_PO”在同步循环中更新过程映像分区输出表 .....	14-8
14.6	使用SFC80 “RSET”复位I/O区域中的位域.....	14-10
14.7	使用SFB 32 “DRUM”实现一个操作序列 .....	14-11

<b>15</b>	<b>用于寻址模块的系统功能</b>	<b>15-1</b>
15.1	使用SFC5 “GADR_LGC”查询模块的逻辑基址.....	15-1
15.2	使用SFC49 “LGC_GADR”查询属于一个逻辑地址的模块 插槽.....	15-3
15.3	使用SFC50 “RD_LGADR”查询模块的所有逻辑地址 .....	15-5
<b>16</b>	<b>用于分布式I/O的SFC</b>	<b>16-1</b>
16.1	使用SFC7 “DP_PRAL”在DP主站上触发硬件中断 .....	16-1
16.2	用SFC11 “DPSYC_FR”同步DP从站组 .....	16-4
16.3	用SFC12 “D_ACT_DP”激活和取消激活DP从站 .....	16-10
16.4	用SFC13 “DPNRM_DG”读取DP从站的诊断数据(从站诊断) .....	16-14
16.5	使用SFC14 “DPRD_DAT”读取DP标准从站的连续数据.....	16-18
16.6	使用SFC15 “DPWR_DAT”将连续数据写入到DP标准从站 .....	16-20
<b>17</b>	<b>用于全局数据通讯的SFC</b>	<b>17-1</b>
17.1	使用SFC60 “GD_SND”发送一个GD信息包 .....	17-1
17.2	通过SFC61 “GD_RCV”编程接受已接收到的GD信息包 .....	17-4
<b>18</b>	<b>通讯和S7基本通讯的概述</b>	<b>18-1</b>
18.1	S7通讯块和S7基本通讯块之间的差别 .....	18-1
18.2	数据的一致性 .....	18-4
18.3	S7通讯块概述 .....	18-6
18.4	用于S7基本通讯的功能块总览 .....	18-8
<b>19</b>	<b>S7通讯</b>	<b>19-1</b>
19.1	用于S7通讯的SFB/FB和SFC/FC的公用参数.....	19-1
19.2	用于组态的S7连接的SFB启动例行程序.....	19-5
19.3	SFB对故障如何反应.....	19-7
19.4	通过SFB8/FB8“USEND”进行无协调的数据发送.....	19-9
19.5	通过SFB/FB9“URCV”进行无协调的数据接收 .....	19-12
19.6	通过SFB/FB12 “BSEND”发送分段数据 .....	19-15
19.7	通过SFB/FB13 “BRCV”接收分段数据 .....	19-18
19.8	通过SFB/FB15“PUT”向远程CPU写入数据 .....	19-21
19.9	通过SFB/FB14GET”从远程CPU中读取数据.....	19-24
19.10	使用SFB16“PRINT”将数据发送到打印机.....	19-27
19.11	通过SFB 19“START”在远程设备上开始一个暖重启或冷 重启.....	19-33
19.12	通过SFB 20“STOP”将远程设备切换到STOP状态 .....	19-36
19.13	通过SFB 21“RESUME”在远程设备上开始一个热重启.....	19-38
19.14	使用SFB 22“STATUS”查询远程伙伴的状态 .....	19-40
19.15	通过SFB23 “USTATUS”接收远程设备的状态改变 .....	19-42
19.16	通过SFC62 “CONTROL”查询属于SFB实例的连接状态 .....	19-44
19.17	通过FC62 “C_CNTRL”查询连接状态 .....	19-46
19.18	S7通讯SFB/FB的工作存储器要求.....	19-48

<b>20</b>	<b>用于未组态S7连接的通讯SFC</b>	<b>20-1</b>
20.1	通讯SFC的公用参数 .....	20-1
20.2	用于未组态S7连接的通讯SFC的出错信息 .....	20-2
20.3	使用SFC65 “X_SEND”将数据发送给在本地S7站外的一个通讯伙伴 .....	20-7
20.4	通过SFC66 “X_RCV”从本地S7站以外的通讯伙伴中接收 数据 .....	20-8
20.5	通过SFC68 “X_PUT”将数据写入本地S7站以外的通讯伙伴 .....	20-11
20.6	通过SFC67 “X_GET”从本地S7站以外的通讯伙伴中读取 数据 .....	20-13
20.7	通过SFC69 “X_ABORT”中止已存在的、到本地S7站以外的通讯伙伴的连接 .....	20-15
20.8	使用SFC72 “I_GET”从本地S7站内的一个通讯伙伴上读取数据 .....	20-16
20.9	使用SFC73 “I_PUT”将数据写入到本地S7站内的一个通讯伙伴 .....	20-18
20.10	通过SFC74 “I_ABORT”中止已存在的、到本地S7站内的通讯伙伴的连接 .....	20-20
<b>21</b>	<b>PROFINet</b>	<b>21-1</b>
21.1	SFC112、113和114背景信息 .....	21-1
21.2	使用SFC112 “PN_IN”更新用于PROFINet组件的用户程序接口的输入 .....	21-4
21.3	使用SFC113 “PN_OUT”更新用于PROFINet组件的用户程序接口的输出 .....	21-5
21.4	使用SFC114 “PN_DP”更新DP互连 .....	21-6
<b>22</b>	<b>生成与块相关的消息</b>	<b>22-1</b>
22.1	关于使用SFB生成块相关消息的介绍 .....	22-1
22.2	使用SFB 36 “NOTIFY”生成无需确认的块相关消息 .....	22-5
22.3	使用SFB31 “NOTIFY_8P”生成无确认显示的块相关消息 .....	22-7
22.4	使用SFB 33 “ALARM”生成需要确认的块相关消息 .....	22-10
22.5	使用SFB35 “ALARM_8P”生成针对八个信号的带有关联值的块相关消息 .....	22-13
22.6	使用SFB34 “ALARM_8”生成针对八个信号的不附带关联值的块相关消息 .....	22-16
22.7	使用SFB37 “AR_SEND”发送归档数据 .....	22-18
22.8	使用SFC10 “DIS_MSG”禁止与块相关的消息、 与符号相关的消息以及组状态消息 .....	22-20
22.9	使用SFC9 “EN_MSG”启用块相关、符号相关和组状态消息 .....	22-22
22.10	用于生成与块相关的消息的SFB的启动特性 .....	22-24
22.11	用于生成与块相关的消息的SFB如何响应故障 .....	22-25
22.12	使用SFC生成与块相关的消息简介 .....	22-26
22.13	使用SFC17 “ALARM_SQ”生成可确认的与块相关的消息以及使用SFC18 “ALARM_S”生成永久确认的与块相关的消息 .....	22-29
22.14	使用SFC19 “ALARM_SC”查询上一ALARM_SQ/ ALARM_DQ进入事件消息的确认状态 .....	22-32
22.15	使用SFC 107 “ALARM_DQ”和108 “ALARM_D” 生成可确认和永久确认的块相关消息 .....	22-33
22.16	使用SFC105 “READ_SI”读取动态系统资源 .....	22-35
22.17	使用SFC106 “READ_SI”读取动态系统资源 .....	22-38
<b>23</b>	<b>IEC定时器和IEC计数器</b>	<b>23-1</b>
23.1	使用SFB 3“TP”生成一个脉冲 .....	23-1
23.2	使用SFB 4“TON”生成一个接通延迟 .....	23-3
23.3	使用SFB 5“TOF”生成一个关闭延迟 .....	23-5
23.4	使用SFB 0 “CTU”递增计数 .....	23-7
23.5	使用SFB 1 “CTD”递减计数 .....	23-8
23.6	使用SFB2 “CTUD”递增/递减计数 .....	23-9

<b>24</b>	<b>IEC功能</b>	<b>24-1</b>
24.1	概述.....	24-1
24.2	IEC功能的技术数据.....	24-3
24.3	日期和时间作为复杂数据类型.....	24-5
24.4	时间功能.....	24-6
24.5	比较DATE_AND_TIME变量.....	24-10
24.6	比较STRING变量.....	24-13
24.7	编辑数值.....	24-16
24.8	STL示例.....	24-17
24.9	STL示例.....	24-18
24.10	编辑STRING变量.....	24-19
24.11	转换数据类型格式.....	24-24
<b>25</b>	<b>用于集成控制的SFB</b>	<b>25-1</b>
25.1	使用SFB41/FB41 “CONT_C”实现连续控制.....	25-1
25.2	使用SFB42/FB42 “CONT_S”进行步控制.....	25-8
25.3	使用SFB43/FB43 “PULSEGEN”生成脉冲.....	25-14
25.4	PULSEGEN块的实例.....	25-25
<b>26</b>	<b>用于紧凑型CPU的SFB</b>	<b>26-1</b>
26.1	通过SFB44 “Analog”使用模拟量输出进行定位.....	26-1
26.2	通过SFB46 “DIGITAL”使用数字量输出进行定位.....	26-13
26.3	使用SFB47 “COUNT”控制计数器.....	26-23
26.4	使用SFB48 “FREQUENC”控制频率测量.....	26-28
26.5	使用SFB49 “PULSE”控制脉宽调制.....	26-32
26.6	使用SFB60 “SEND_PTP”发送数据(ASCII, 3964(R)).....	26-35
26.7	使用SFB61 “RCV_PTP”接收数据(ASCII, 3964(R)).....	26-38
26.8	使用SFB62 “RES_RCVB”删除接收缓冲区(ASCII, 3964(R)).....	26-41
26.9	使用SFB63 “SEND_RK”发送数据(512(R)).....	26-43
26.10	使用SFB64 “FETCH_RK”获取数据(RK 512).....	26-47
26.11	使用SFB65 “SERVE_RK”接收和提供数据(RK 512).....	26-52
26.12	SFB 60至65的附加出错信息.....	26-57
<b>27</b>	<b>用于H CPU的SFC</b>	<b>27-1</b>
27.1	在H系统中使用SFC90 “H_CTRL”控制操作.....	27-1
<b>28</b>	<b>集成功能(对于带集成I/O的CPU)</b>	<b>28-1</b>
28.1	SFB29(HS_COUNT).....	28-1
28.2	SFB30(FREQ_MES).....	28-3
28.3	SFB38(HSC_A_B).....	28-4
28.4	SFB39(POS).....	28-5
<b>29</b>	<b>Plastics Techology</b>	<b>29-1</b>
29.1	SFC63 (AB_CALL).....	29-1
<b>30</b>	<b>诊断数据</b>	<b>30-1</b>
30.1	诊断数据结构概述.....	30-1
30.2	诊断数据.....	30-2
30.3	通道专有诊断数据的结构.....	30-4

<b>31</b>	<b>系统状态列表(SSL)</b>	<b>31-1</b>
31.1	系统状态列表(SSL)概述.....	31-1
31.2	部分SSL列表的结构.....	31-3
31.3	SSL-ID .....	31-4
31.4	可能的部分系统状态列表 .....	31-5
31.5	SSL-ID W#16#xy11 - 模块标识 .....	31-6
31.6	SSL-ID W#16#xy12 - CPU特征 .....	31-7
31.7	SSL-ID W#16#xy13 - 存储区域 .....	31-9
31.8	SSL-ID W#16#xy14 - 系统区 .....	31-10
31.9	SSL-ID W#16#xy15 - 块类型 .....	31-12
31.10	SSL-ID W#16#xy19 - 模块LED的状态.....	31-13
31.11	SSL-ID W#16#xy1C - 组件标识.....	31-15
31.12	SSL ID W#16#xy25 - 将过程映像分区分配到OB.....	31-18
31.13	SSL-ID W#16#xy32 - 通讯状态数据 .....	31-21
31.14	SSL-ID W#16#0132, 索引为W#16#0005的部分列表的 数据记录摘录 .....	31-22
31.15	SSL-ID W#16#0132, 索引为W#16#0008的部分列表的 数据记录摘录 .....	31-23
31.16	SSL-ID为W#16#0132、索引为W#16#000B的部分列表摘 录的数据记录 ....	31-25
31.17	SSL-ID为W#16#0132、索引为W#16#000C的部分列表摘 录的数据记录 ....	31-26
31.18	SSL-ID W#16#0232, 索引为W#16#0004的部分列表的 数据记录摘录 .....	31-27
31.19	SSL-ID W#16#xy71 - H CPU的组信息 .....	31-28
31.20	SSL-ID W#16#xy74 - 模块LED的状态.....	31-31
31.21	SSL-ID W#16#xy75 - H系统中切换的DP从站 .....	31-33
31.22	SSL-ID W#16#xy90 - DP主站系统信息 .....	31-35
31.23	SSL-ID W#16#xy91 - 模块的状态信息.....	31-37
31.24	SSL-ID W#16#xy92 - 机架/站的状态信息 .....	31-41
31.25	SSL-ID W#16#xy95 - 扩展DP主站系统信息.....	31-44
31.26	SSL-ID W#16#xyA0 - 诊断缓冲区 .....	31-46
31.27	SSL-ID W#16#00B1 - 模块的诊断信息.....	31-47
31.28	SSL-ID W#16#00B2 - 对应物理地址的诊断数据记录1 .....	31-49
31.29	SSL-ID W#16#00B3 - 带逻辑基址的模块诊断数据.....	31-50
31.30	SSL-ID W#16#00B4 - DP从站的诊断数据 .....	31-51
<b>32</b>	<b>事件</b>	<b>32-1</b>
32.1	事件和事件标识符 .....	32-1
32.2	事件等级1 - 标准OB事件.....	32-3
32.3	事件等级2 - 异步出错.....	32-3
32.4	事件等级3 - 异步出错.....	32-4
32.5	事件等级4 - 停止事件和其它模式改变 .....	32-7
32.6	事件等级5 - 模式运行期事件 .....	32-10
32.7	事件等级6 - 通讯事件.....	32-11
32.8	事件等级7 - H/F事件 .....	32-12
32.9	事件等级8 - 模块的诊断事件 .....	32-14
32.10	事件等级9 - 标准用户事件.....	32-16
32.11	事件等级A和B - 自由用户事件 .....	32-17
32.12	保留的事件等级.....	32-18

<b>33</b>	<b>SFC和SFB列表</b>	<b>33-1</b>
33.1	按编号排序的SFC列表.....	33-1
33.2	按字母排序的SFC列表.....	33-4
33.3	按编号排序的SFB列表.....	33-7
33.4	按字母排序的SFB列表.....	33-9

参考书目

词汇表

索引



# 1 组织块

## 1.1 组织块(OB)概述

### 何为组织块？

组织块(OB)是指CPU的操作系统与用户程序之间的接口。OB用于执行特定的程序段：

- 启动CPU时
- 在循环或定时执行过程中
- 出错时
- 发生硬件中断时。

组织块按照已分配的优先级来执行。

### 哪些OB可用？

并非所有的CPU均可处理STEP 7中可用的所有OB。请参阅操作列表/72/和/102/以确定用户CPU中所包含的OB。

### 在何处查找详细信息？

请参阅在线帮助和下列手册以获取更多信息：

- /70/：此手册包含描述不同S7-300 CPU性能的技术数据。
- /101/：此手册包含描述不同S7-400 CPU性能的技术数据。

下表中给出了属于每个OB的启动事件和缺省优先级。

OB	启动事件	缺省优先级	说明
OB1	启动结束或OB1结束	1	自由循环
OB10	时钟中断0	2	未指定缺省时间
OB11	时钟中断1	2	
OB12	时钟中断2	2	
OB13	时钟中断3	2	
OB14	时钟中断4	2	
OB15	时钟中断5	2	
OB16	时钟中断6	2	
OB17	时钟中断7	2	
OB20	延时中断0	3	未指定缺省时间
OB21	延时中断1	4	
OB22	延时中断2	5	
OB23	延时中断3	6	
OB30	周期性中断0(缺省时间间隔: 5s)	7	周期性中断
OB31	周期性中断1(缺省时间间隔: 2s)	8	
OB32	周期性中断2(缺省时间间隔: 1s)	9	
OB33	周期性中断3(缺省时间间隔: 500ms)	10	
OB34	周期性中断4(缺省时间间隔: 200ms)	11	
OB35	周期性中断5(缺省时间间隔: 100ms)	12	
OB36	周期性中断6(缺省时间间隔: 50ms)	13	
OB37	周期性中断7(缺省时间间隔: 20ms)	14	
OB38	周期性中断8(缺省时间间隔: 10ms)	15	
OB40	硬件中断 0	16	硬件中断
OB41	硬件中断 1	17	
OB42	硬件中断 2	18	
OB43	硬件中断 3	19	
OB44	硬件中断 4	20	
OB45	硬件中断 5	21	
OB46	硬件中断 6	22	
OB47	硬件中断 7	23	
OB55	状态中断	2	DPV1中断
OB56	更新中断	2	
OB57	制造商特定中断	2	
OB60	SFC35 “MP_ALM”调用	25	多值计算中断
OB 61	同步周期性中断1	25	同步周期性中断
OB 62	同步周期性中断2	25	
OB 63	同步周期性中断3	25	
OB 64	同步周期性中断4	25	

OB	启动事件	缺省优先级	说明
OB70	I/O冗余出错(仅在H CPU中)	25	冗余出错中断
OB72	CPU冗余出错(仅在H CPU中)	28	
OB 73	通讯冗余出错OB(仅在H CPU中)	25	
OB80	时间出错	26、28 <sup>1)</sup>	异步出错中断
OB81	电源故障	25、28 <sup>1)</sup>	
OB82	诊断中断	25、28 <sup>1)</sup>	
OB83	插入/删除模块中断	25、28 <sup>1)</sup>	
OB84	CPU硬件故障	25、28 <sup>1)</sup>	
OB85	程序出错	25、28 <sup>1)</sup>	
OB86	扩展机架、DP主站系统或用于分布式I/O的站故障	25、28 <sup>1)</sup>	
OB87 OB 88	通讯出错 处理中断	25、28 <sup>1)</sup> 28	
OB90	暖重启或冷重启，或删除OB90中正在执行的块，或将在CPU上装载OB90或终止OB90	29 <sup>2)</sup>	后台循环
OB100	暖重启	27 <sup>1)</sup>	启动
OB101	热重启	27 <sup>1)</sup>	
OB102	冷重启	27 <sup>1)</sup>	
OB121	编程出错	导致出错的OB优先级	同步出错中断
OB122	I/O访问出错	导致出错的OB优先级	

<sup>1)</sup> 优先级27和28在启动的优先级模型中有效。

<sup>2)</sup> 优先级29对应于优先级0.29。这就意味着后台循环的优先级低于自由循环。

## 1.2 程序循环组织块(OB1)

### 描述

S7 CPU的操作系统定期执行OB1。执行OB1后，操作系统将再次启动它。完成启动后，将启动OB1的循环执行。可调用OB1中的其它功能块(FB、SFB)或功能(FC、SFC)。

### 了解OB1的操作

在运行期受监视的所有OB中，OB1的优先级最低，换言之，除OB90之外的其它所有OB均可中断OB1的执行。以下事件可导致操作系统调用OB1：

- 完成启动。
- 已完成OB1的执行(上一循环)。

执行OB1后，操作系统发送全局数据。重新启动OB1之前，操作系统会将过程映像输出表写入输出模块中、更新过程映像输入表以及接收CPU的任何全局数据。

S7监视最大的扫描时间，并确保最大的响应时间。最大扫描时间的值预置为150ms。用户可设置一个新值，也可在用户程序中的任意位置使用SFC43“RE\_TRIGR”来重新启动时间监视。如果用户程序超出了OB1的最大周期，则操作系统将调用OB80(时间错误OB)；如果OB80没有编程，则CPU将转为STOP模式。

除了监视最大扫描时间外，还可以保证最小扫描时间。操作系统将延迟启动新循环(将过程映像输出表写入输出模块中)，直至达到最小扫描时间为止。

请参阅手册 */70/* 和 */101/* 以了解参数“最大”和“最小”扫描时间的范围。可使用STEP 7更改参数设置。

## OB1的本地数据

下表描述了OB1的临时(TEMP)变量。变量名为OB1的缺省名称。

变量	类型	描述
OB1_EV_CLASS	BYTE	事件等级和标识符: B#16#11: OB1激活
OB1_SCAN_1	BYTE	<ul style="list-style-type: none"> <li>• B#16#01: 完成暖重启</li> <li>• B#16#02: 完成热重启</li> <li>• B#16#03: 完成主循环</li> <li>• B#16#04: 完成冷重启</li> <li>• B#16#05: 主站-保留站切换和“停止”上一主站之后新主站CPU的首个OB1循环</li> </ul>
OB1_PRIORITY	BYTE	优先级1
OB1_OB_NUMBR	BYTE	OB编号(01)
OB1_RESERVED_1	BYTE	保留
OB1_RESERVED_2	BYTE	保留
OB1_PREV_CYCLE	INT	上一次扫描的运行时间(ms)
OB1_MIN_CYCLE	INT	自上次启动后的最小周期(ms)
OB1_MAX_CYCLE	INT	从上次启动后的最大周期(ms)
OB1_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

## 1.3 时钟中断组织块 (OB10到OB17)

### 描述

STEP 7提供了多达八个OB (OB10到OB17)，这些OB可单次运行，也可定期运行。可使用SFC或STEP 7为CPU分配参数，以便以下列时间间隔处理这些OB：

- 单次
- 每分钟
- 每小时
- 每天
- 每周
- 每月
- 每个月末

---

### 注意

对于每月执行的时钟中断OB，只可将1、2、...28号作为起始日期。

---

### 了解时钟中断OB的操作

要启动时钟中断，必须先设置中断，然后再将其激活。

有以下三种可能的启动方式：

- 自动启动时钟中断。一旦使用STEP 7设置并激活了时钟中断，即会发生此情况。下表说明了使用STEP 7激活时钟中断的基本可能情况。
- 使用STEP 7设置日时钟中断，然后通过调用程序中的SFC30 “ACT-TINT”来激活它。
- 通过调用SFC28 “SET\_TINT”来设置日时钟中断，然后通过调用SFC30 “ACT\_TINT”来激活它。

时间间隔	描述
未激活	时钟中断未执行，即使将其装载到CPU中也是如此。可通过调用SFC30来激活它。
仅激活一次	时间OB按指定时间运行一次后，即被自动取消。 用户程序可使用SFC28和SFC30来复位并重新激活OB。
定期激活	当发生时钟中断时，CPU将根据当前的时间和周期计算时间中断的下一次启动的时间。

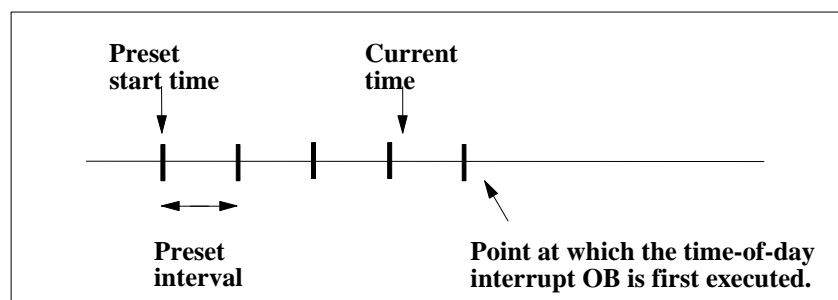
当前后拨动时钟时，时间中断的特性将如**234**

### 注意

如果以如下方式组态时间中断，即单次处理相应的OB要，则DATE\_AND\_TIME必须为过去的时间(相对于CPU的实际时间)。

如果以如下方式组态时间中断，即定期处理相应的OB，而启动DATE\_AND\_TIME为过去的时间，随后将在下次预期的时间处理该时间中断。下图对此进行了说明。

可以使用SFC 39至42禁用或延迟和重新启用时钟中断。



### 影响时钟中断OB的条件

由于时间中断仅以指定的时间间隔发生，因此在执行用户程序期间，某些条件可能会影响OB的操作。下表列出了其中的一些条件，并说明了该条件对执行时间中断OB的影响。

条件	结果
用户程序调用SFC29 (CAN_TINT)并取消时间中断。	操作系统清除了时钟中断的启动事件(DATE_AND_TIME)。必须再次设置启动事件并在再次调用OB之前激活它。
用户程序试图激活时间中断OB，但未将OB加载到CPU中。	操作系统调用OB85。如果OB85尚未编程(装载到CPU中)，则CPU将转为STOP模式。
当同步或更正CPU的系统时钟时，用户提前设置了时间并跳过时间OB的启动事件日期或时间。	操作系统调用OB80并对时间OB的编号和OB80中的启动事件信息进行编码。 随后操作系统将运行一次时间OB，而不管本应执行此OB的次数。OB80的启动事件信息给出了第一次跳过时间OB的DATE_AND_TIME。
当同步或更正CPU的系统时钟时，推后设置了时间以使OB的启动事件、日期或时间得以重复。	<b>S7-400-CPU和CPU 318:</b> 如果在推后设置时钟之前已激活了时间OB，则不会再次调用它。 <b>S7-300-CPU:</b> 执行时间OB。
CPU通过暖重启或冷重启运行。	由SFC组态的所有时间OB会被改回在STEP 7中指定的组态。 如果已为相应OB的单个启动组态了时间中断，并使用STEP 7对其进行了设置，并将其激活，则当所组态的启动时间为已过去的时间(相对于CPU的实时时钟)时，会在暖重启或冷重启操作系统后调用一次OB。
当发生下一时间间隔的启动事件时，仍执行时间OB。	操作系统调用OB80。如果OB80没有编程，则CPU转为STOP模式。 如果装载了OB80，则会首先执行OB80和时钟中断OB，然后再执行请求的中断。



## 时钟中断OB的本地数据

下表给出了时间中断OB的临时(TEMP)变量。变量名为OB10的缺省名称。

变量	类型	描述
OB10_EV_CLASS	BYTE	事件等级和标识符: B#16#11 = 中断处于激活状态
OB10_STRT_INFO	BYTE	B#16#11: OB10的启动请求 (B#16#12: OB11的启动请求) : : (B#16#18: OB17的启动请求)
OB10_PRIORITY	BYTE	分配的优先级; 缺省值为2
OB10_OB_NUMBR	BYTE	OB编号(10至17)
OB10_RESERVED_1	BYTE	保留
OB10_RESERVED_2	BYTE	保留
OB10_PERIOD_EXE	WORD	OB以指定的时间间隔执行: W#16#0000: 单次 W#16#0201: 每分钟一次 W#16#0401: 每小时一次 W#16#1001: 每天一次 W#16#1201: 每周一次 W#16#1401: 每月一次 W#16#1801: 每年一次 W#16#2001: 月末
OB10_RESERVED_3	INT	保留
OB10_RESERVED_4	INT	保留
OB10_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

## 1.4 时间延迟中断组织块 (OB20到OB23)

### 描述

S7提供多达四个在指定延迟后执行的OB (OB20到OB23)。每个延时OB均可通过调用SFC32 (SRT\_DINT)来启动。延迟时间是SFC的一个输入参数。

当用户程序调用SFC32 (SRT\_DINT)时，需要提供OB编号、延迟时间和用户专用的标识符。经过指定的延迟后，OB将会启动。还可取消尚未启动的延时中断。

### 了解延时中断OB的操作

延迟时间(同OB编号一起传送给SFC32的值，单位为ms)到期后，操作系统将启动相应的OB。

要使用延时中断，必须执行以下任务：

- 必须调用SFC32 (SRT\_DINT)。
- 必须将延时中断OB作为用户程序的一部分下载到CPU。

只有当CPU处于RUN模式下时才会执行延时OB。暖重启或冷重启将清除延时OB的所有启动事件。如果延时中断还未启动，则可使用SFC33 (CAN\_DINT)取消执行。

延迟时间的分辨率为1ms。已到期的延迟时间可立即再次启动。可使用SFC34 (QRY\_DINT)查询延时中断的状态。

如果发生以下事件之一，操作系统将调用异步出错OB：

- 如果操作系统试图启动一个尚未装载的OB，并且用户在调用SFC32 “SRT\_DINT”时指定了其编号。
- 如果在完全执行延时OB之前发生延时中断的下一个启动事件。

可使用SFC 39至42来禁用或延迟并重新使能延迟中断。

## 延时中断OB的本地数据

下表描述了延时中断OB的临时(TEMP)变量。变量名为OB20的缺省名称。

变量	类型	描述
OB20_EV_CLASS	BYTE	事件等级和标识符： B#16#11：中断处于激活状态
OB20_STRT_INF	BYTE	B#16#21：OB20的启动请求 (B#16#22：OB21的启动请求) (B#16#23：OB22的启动请求) (B#16#24：OB23的启动请求)
OB20_PRIORITY	BYTE	分配的优先级：缺省值3(OB20)至6(OB23)
OB20_OB_NUMBR	BYTE	OB编号(20至23)
OB20_RESERVED_1	BYTE	保留
OB20_RESERVED_2	BYTE	保留
OB20_SIGN	WORD	用户ID：通过调用SFC32 (SRT_DINT)的输入参数SIGN
OB20_DTIME	TIME	已组态的延迟时间(单位为ms)
OB20_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

## 1.5 周期性中断组织块 (OB30到OB38)

### 描述

S7提供了多达九个的循环中断OB (OB30到OB38)，它们以固定的时间间隔来中断用户程序。下表给出了循环中断OB的缺省时间间隔和优先级。

OB编号	缺省时间间隔	缺省优先级
OB30	5s	7
OB31	2s	8
OB32	1s	9
OB33	500ms	10
OB34	200ms	11
OB35	100ms	12
OB36	50ms	13
OB37	20ms	14
OB38	10ms	15

### 了解周期性中断OB的操作

循环中断OB的等距启动时间是由时间间隔和相位偏移量决定的。请参阅/234/，以了解OB的启动时间、周期以及相位偏移量之间的关系。

#### 注意

必须确保每个周期性中断OB的运行时间远远小于其时间间隔。如果因时间间隔已到期，如果在预期的再次执行前未完全执行循环中断OB，则启动时间错误OB (OB80)。稍后将执行导致错误的循环中断。

可使用SFC 39至42来禁用或延迟，并重新启用周期性中断

请参阅用户特定CPU的说明书，了解参数时间间隔、优先级和相位偏移量的范围。可使用STEP 7来更改参数设置。

## 周期性中断OB的本地数据

下表给出了循环中断OB的临时(TEMP)变量。变量名为OB35的缺省名称。

变量	类型	描述
OB35_EV_CLASS	BYTE	事件等级和标识符 B#16#11: 中断处于激活状态
OB35_STRT_INF	BYTE	B#16#30: 具有特殊标准的周期性中断OB的启动请求(仅适用于H-CPU, 且仅当对其进行了明确组态后) B#16#31 : OB30的启动请求 : B#16#36 : OB35的启动请求 : B#16#39 : OB38的启动请求
OB35_PRIORITY	BYTE	分配的优先级: 缺省值为7(OB30)至15(OB38)
OB35_OB_NUMBR	BYTE	OB编号(30至38)
OB35_RESERVED_1	BYTE	保留
OB35_RESERVED_2	BYTE	保留
OB35_PHASE_OFFSET	WORD	相位偏移量[ms]
OB35_RESERVED_3	INT	保留
OB35_EXC_FREQ	INT	时间间隔(单位为ms)
OB35_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

## 1.6 硬件中断组织块 (OB40到OB47)

### 描述

S7提供了多达八个的独立硬件中断，每一中断都具有自己的OB。

通过STEP 7分配参数，可以为将要触发硬件中断的每个信号模块指定以下内容：

- 在何种条件下触发硬件中断的通道。
- 要分配给个别通道组的硬件中断OB(缺省情况下，将由OB40来处理所有的硬件中断)。

借助于CP和FM，可使用其各带软件来分配这些参数。

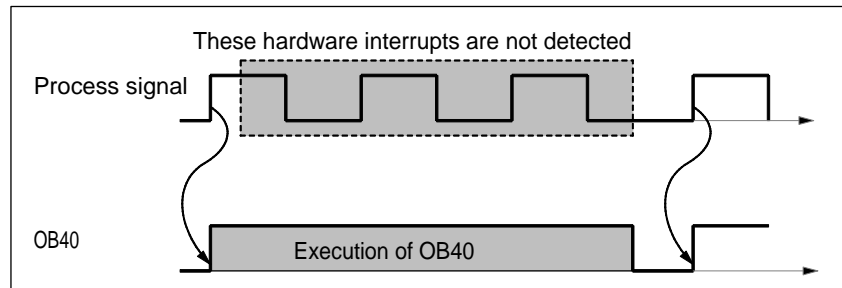
可使用STEP 7选择各个硬件中断OB的优先级。

### 了解硬件中断OB的操作

在模块触发硬件中断后，操作系统将标识插槽以及相应的硬件中断OB。如果此OB的优先级高于当前激活的优先级，则将启动该OB。当执行完此硬件中断OB后，将发送通道指定的确认。

如果在对硬件中断进行标识和确认的这段时间内，在同一模块中发生了触发硬件中断的另一事件，则应用以下规则：

- 如果该事件发生在先前触发硬件中断的通道中，则新中断丢失。在下图中，将根据数字输入模块的通道实例对此进行说明。触发事件是上升沿。硬件中断OB是OB40。



- 如果该事件发生在同一模块的另一通道中，通常不会触发任何硬件中断。然而此中断不会丢失，而是在确认当前激活的硬件中断后被触发。

如果因来自另一模块中的硬件中断而使某一硬件中断被触发，并且其OB当前处于激活状态，则将记录新请求并且在OB空闲时对其进行处理。

可使用SFC 39至42来禁用或延迟，并重新启用硬件中断。

既可以使用STEP 7，也可以使用SFC 55至57为模块的硬件中断分配参数。

### 硬件中断OB的本地数据

下表给出了硬件中断OB的临时(TEMP)变量。变量名为OB40的缺省名称。

变量	类型	描述
OB40_EV_CLASS	BYTE	事件等级和标识符： B#16#11：中断处于激活状态
OB40_STRT_INF	BYTE	<ul style="list-style-type: none"> <li>• B#16#41：通过中断线1中断</li> <li>• B#16#42：通过中断线2中断 (仅限S7-400)</li> <li>• B#16#43：通过中断线3中断 (仅限S7-400)</li> <li>• B#16#44：通过中断线4中断 (仅限S7-400)</li> <li>• B#16#45：WinAC：通过PC触发中断</li> </ul>
OB40_PRIORITY	BYTE	分配的优先级：缺省值为16(OB40)至23(OB47)
OB40_OB_NUMBR	BYTE	OB编号(40至47)
OB40_RESERVED_1	BYTE	保留
OB40_IO_FLAG	BYTE	输入模块：B#16#54 输出模块：B#16#55
OB40_MDL_ADDR	WORD	触发中断的模块的逻辑基址
OB40_POINT_ADDR	DWORD	<ul style="list-style-type: none"> <li>• 对于数字模块： 模块上具有输入状态的位域(位0对应于第一个输入)可在给定模块的说明中找到为模块中的通道分配的从OB40_POINT_ADDR起始的位。</li> <li>• 对于模拟模块： 位域，指出哪个通道已超出哪条限制 (有关结构的详细信息，请参阅71/ 或101/ )。</li> <li>• 对于CP或IM： 模块中断状态(非用户相关)</li> </ul>
OB40_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

### 注意

如果使用了具有DPV1功能的CPU，可借助于SFB54 "RALRM"获取有关中断的其它信息，这些信息不止包含OB的起始信息。在S7兼容模式下操作DP主站时，这一点也同样适用。

## 1.7 状态中断OB(OB 55)

### 注意

状态中断OB (OB 55)仅对具有DPV1功能的CPU可用。

### 描述

如果状态中断是通过DPV1从站的插槽触发，则CPU操作系统调用OB55。  
如果DPV1从站的组件(模块或机架)更改了其操作模式(例如由RUN改为STOP)，则可能会发生这种情况。有关可触发状态中断的事件的详细信息，请参阅DPV1从站制造商的文档。

### 状态中断OB的本地数据

下表给出了状态中断OB的临时(TEMP)变量。所选变量的名称为OB55的缺省名称。

变量	数据类型	描述
OB55_EV_CLASS	BYTE	事件等级和标识符： B#16#11(即将发生的事件)
OB55_STRT_INF	BYTE	B#16#55(OB55的启动请求)
OB55_PRIORITY	BYTE	已组态的优先级，缺省值为2
OB55_OB_NUMBR	BYTE	OB编号(55)
OB55_RESERVED_1	BYTE	保留
OB55_IO_FLAG	BYTE	输入模块：B#16#54 输出模块：B#16#55
OB55_MDL_ADDR	WORD	中断触发组件(模块)的逻辑基址
OB55_LEN	BYTE	中断所提供的数据块长度
OB55_TYPE	BYTE	中断类型状态中断的ID
OB55_SLOT	BYTE	中断触发组件(模块)的插槽号
OB55_SPEC	BYTE	说明符 <ul style="list-style-type: none"> <li>位0至1：中断说明符</li> <li>位2：Add_Ack</li> <li>位3至7：顺序号</li> </ul>
OB55_DATE_TIME	DATE_AND_TIME	调用OB时的日期和时间

### 注意

通过OB55调用SFB54“RALRM”，可获得DP消息帧所含中断的完整辅助信息。



## 1.8 更新中断OB(OB 56)

### 注意

更新中断OB (OB 56)仅对具有DPV1功能的CPU可用。

### 描述

如果更新中断是通过DPV1从站的插槽触发，则CPU操作系统调用OB56。  
如果用户更改了DPV1从站插槽的参数(通过本地或远程访问)，则可能会发生这种情况。有关可触发更新中断的事件的详细信息，请参阅DPV1从站制造商的文档。

### 更新中断OB的本地数据

下表给出了更新中断OB的临时(TEMP)变量。所选变量名为OB56的缺省名称。

变量	数据类型	描述
OB56_EV_CLASS	BYTE	事件等级和标识符： B#16#11(即将发生的事件)
OB56_STRT_INF	BYTE	B#16#56(OB56的启动请求)
OB56_PRIORITY	BYTE	组态的优先级，缺省值为2
OB56_OB_NUMBR	BYTE	OB编号(56)
OB56_RESERVED_1	BYTE	保留
OB56_IO_FLAG	BYTE	输入模块：B#16#54 输出模块：B#16#55
OB56_MDL_ADDR	WORD	中断触发组件(模块)的逻辑基址
OB56_LEN	BYTE	中断所提供的数据块长度
OB56_TYPE	BYTE	中断类型更新中断的ID
OB56_SLOT	BYTE	中断触发组件(模块)的插槽号
OB56_SPEC	BYTE	说明符 <ul style="list-style-type: none"> <li>位0至1：中断符</li> <li>位2：Add_Ack</li> <li>位3至7：顺序号</li> </ul>
OB56_DATE_TIME	DATE_AND_TIME	调用OB时的日期和时间

### 注意

通过OB56调用SFB54“RALRM”，可获得DP消息帧所含中断的完整辅助信息。

## 1.9 制造商特定中断OB(OB57)

### 注意

制造商特定中断的OB (OB57)仅对具有DPV1功能的CPU可用。

### 描述

如果制造商特定中断是通过DPV1从站的插槽触发的，则CPU操作系统调用OB57。

### 制造商特定中断的OB的本地数据

下表给出制造商特定中断的OB的临时(TEMP)变量。所选变量名为OB57的缺省名称。

变量	数据类型	描述
OB57_EV_CLASS	BYTE	事件等级和标识符： B#16#11(即将发生的事件)
OB57_STRT_INF	BYTE	B#16#57(OB57的启动请求)
OB57_PRIORITY	BYTE	已组态的优先级，缺省值为2
OB57_OB_NUMBR	BYTE	OB编号(57)
OB57_RESERVED_1	BYTE	保留
OB57_IO_FLAG	BYTE	输入模块：B#16#54 输出模块：B#16#55
OB57_MDL_ADDR	WORD	中断触发组件(模块)的逻辑基址
OB57_LEN	BYTE	中断所提供的数据块长度
OB57_TYPE	BYTE	中断类型制造商特定中断的ID
OB57_SLOT	BYTE	中断触发组件(模块)的插槽号
OB57_SPEC	BYTE	说明符 <ul style="list-style-type: none"> <li>位0至1：中断说明符</li> <li>位2：Add_Ack</li> <li>位3至7：顺序号</li> </ul>
OB57_DATE_TIME	DATE_AND_TIME	调用OB时的日期和时间

### 注意

通过OB57调用SFB54“RALRM”，可获得DP消息帧所含中断的完整辅助信息。

## 1.10 多值计算中断组织块(OB60)

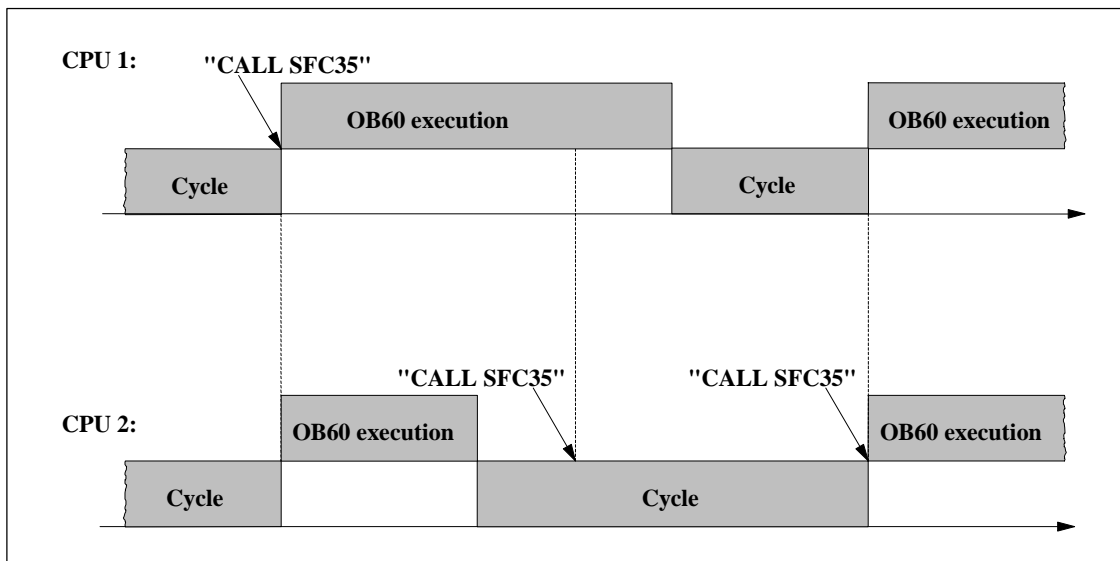
### 描述

使用多值计算中断，可确保CPU的反应在多值计算过程中与事件同步。与信号模块所触发的硬件中断比较而言，多值计算中断只能由CPU输出。

### 了解多值计算中断OB的操作

多值计算中断通过调用SFC35 “MP\_ALM”来触发。通过调用SFC35 “MP\_ALM”触发多值计算中断。在多值计算期间，导致总线段上所有CPU上的同步的OB60启动，除非禁用(通过SFC39 “DIS\_IRT”)或延迟(通过SFC41 “DIS\_AIRT”)了OB60。如果尚未将OB60装载到CPU上，则该CPU会在中断前返回上一优先级，并继续在此执行程序。在单处理器操作中以及在使用分段机架时，仅在用户调用SFC35 “MP\_ALM”的CPU上启动OB60。

当用户程序调用SFC35 “MP\_ALM”时，应为所有的CPU提供一个作业ID。这就允许用户对特定事件作出反应。如果用户在不同的CPU上以不同方式对OB60进行编程，将导致OB的执行时间不同。在这种情况下，CPU将在不同时间返回到被中断的优先级。如果下一个多值计算中断由CPU输出，而另一个CPU仍在忙于执行上一个多值计算中断的OB60，则OB60既不会在请求时启动，也不会属于该总线段的任何其它CPU上启动。下图以两个CPU为例对此进行了说明。所调用SFC35的函数值将通知您具体结果。



### 多值计算中断OB的本地数据

下表说明了多值计算中断OB的临时(TEMP)变量。变量名为OB60的缺省名称。

变量	数据类型	描述
OB60_EV_CLASS	BYTE	事件等级和ID: B#16#11: 中断处于激活状态
OB60_STRT_INF	BYTE	B#16#61: 由自身的CPU触发的多值计算中断 B#16#62: 由其它CPU触发的多值计算中断
OB60_PRIORITY	BYTE	分配的优先级: 缺省值为25
OB60_OB_NUMBR	BYTE	OB编号: 60
OB60_RESERVED_1	BYTE	保留
OB60_RESERVED_2	BYTE	保留
OB60_JOB	INT	作业ID: SFC35 "MP_ALM"的输入变量JOB
OB60_RESERVED_3	INT	保留
OB60_RESERVED_4	INT	保留
OB60_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME。

## 1.11 同步周期性中断OB(OB 61到OB 64)

### 描述

通过同步循环中断可选择在具有DP循环的同步循环中启动程序。OB 61充当同步循环中断TSAL1的接口OB。可以将OB 61的优先级设置为0 (取消选定OB)和2到26之间的数。



### 当心

在使用L或T命令(例如L PEB、T PAB)直接访问, 以及使用SFC 14 “DPRD\_DAT”和15 “DPWR\_DAT”时, 请避免访问已为其过程映像分区分配了到OB6x (同步循环中断)的连接I/O区域

### 同步周期性中断OB的本地数据

下表说明了同步周期性中断OB的临时(TEMP)变量。变量名为OB 61的缺省名称。

变量	数据类型	描述
OB61_EV_CLASS	BYTE	事件等级和ID: B#16#11: 中断处于激活状态
OB61_STRT_INF	BYTE	B#16#64: OB 61 的启动请求 B#16#67: OB 64的启动请求
OB61_PRIORITY	BYTE	分配的优先级; 缺省值: 25
OB61_OB_NUMBR	BYTE	OB编号: 61 ... 64
OB61_RESERVED_1	BYTE	保留
OB61_RESERVED_2	BYTE	保留
OB61_GC_VIOL	BOOL	GC违例
OB61_FIRST	BOOL	启动或停止状态后的首次使用
OB61_MISSED_EXEC	BYTE	自上次执行OB 61以来启动OB 61失败的次数
OB61_DP_ID	BYTE	同步DP主站系统的DP主站系统标识号
OB61_RESERVED_3	BYTE	保留
OB61_RESERVED_4	WORD	保留
OB61_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME。

## 1.12 I/O冗余出错OB(OB70)

### 注意

I/O冗余出错OB (OB70)仅可与H CPU结合使用。

### 描述

当在PROFIBUS DP上存在冗余丢失时(例如, 激活DP主站发生总线故障, 或DP从站的接口模块出错), 或者当具有连接I/O的DP从站的激活DP主站发生改变时, H CPU的操作系统将调用OB70。

如果发生了启动事件而OB70没有编程, 则CPU不会转为STOP模式。如如果装载了OB70且H系统处于冗余模式, 则在两个CPU上执行OB70。H系统仍处于冗余模式。

### I/O冗余OB的本地数据

下表给出了I/O冗余错误OB的临时(TEMP)变量。所选变量名为OB70的缺省名称。

变量	类型	描述
OB70_EV_CLASS	BYTE	事件等级和ID: <ul style="list-style-type: none"> <li>• B#16#72: 离开事件</li> <li>• B#16#73: 进入事件</li> </ul>
OB70_FLT_ID	BYTE	出错代码(可能值: B#16#A2、B#16#A3
OB70_PRIORITY	BYTE	优先级, 可通过STEP 7分配 (硬件配置)
OB70_OB_NUMBR	BYTE	OB编号(70)
OB70_RESERVED_1	WORD	保留
OB70_INFO_1	WORD	取决于出错代码
OB70_INFO_2	WORD	取决于出错代码
OB70_INFO_3	WORD	取决于出错代码
OB70_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

下表给出了启动OB70的事件。

OB70_EV_CLASS	OB70_FLT_ID	OB70的启动事件
B#16#72	B#16#A2	DP主站或DP主站系统分别发生故障
B#16#73/B#16#72	B#16#A3	冗余丢失/返回DP从站的冗余

取决于错误代码的变量具有如下含义：

出错代码:	位	含义
<ul style="list-style-type: none"> <li>• B#16#A2</li> <li>- OB70_INFO_1:</li> <li>- OB70_INFO_2:</li> <li>- OB70_INFO_3:</li> </ul>	0至7: 8至15:	受影响DP主站的逻辑基址 保留 保留 受影响DP主站的DP主站系统ID
<ul style="list-style-type: none"> <li>• B#16#A3</li> <li>- OB70_INFO_1:</li> <li>- OB70_INFO_2:</li> <li>- OB70_INFO_3</li> </ul>	0至14: 15:  0至7: 8至15:	DP主站的逻辑基址 受影响的DP从站: 逻辑基址(使用S7从站时)或诊断地址(使用DP标准从站时)。 I/O标识 受影响的DP从站: DP站编号 DP主站系统标识号

#### 注意

如果使用了具有DPV1功能的CPU，可借助于SFB54“RALRM”获取有关中断的其它信息，这些信息不止包含OB的起始信息。在S7兼容模式下操作DP主站时，这一点也同样适用。

## 1.13 CPU冗余出错OB(OB72)

### 注意

只在H CPU中存在CPU冗余错误OB (OB72)。

### 描述

如果发生下列事件之一，则H CPU的操作系统将调用OB72：

- CPU冗余丢失
- 保留主站切换
- 同步出错
- SYNC模块中的出错
- 更新中止
- 比较出错(例如，RAM、PIQ)

在适当的启动事件后，OB72由处于RUN或STARTUP模式下的所有CPU执行。

### CPU冗余出错OB的本地数据

下表给出了CPU冗余错误OB的临时(TEMP)变量。OB72的缺省名称已被用作变量名。

变量	类型	描述
OB72_EV_CLASS	BYTE	事件等级和ID： B#16#78：离开事件 B#16#73、B#16#75、B#16#79：进入事件
OB72_FLT_ID	BYTE	出错代码(可能值：B#16#01, B#16#02, B#16#03, B#16#20, B#16#21, B#16#22, B#16#23, B#16#31, B#16#33, B#16#34, B#16#35, B#16#40, B#16#41, B#16#42, B#16#43, B#16#44, B#16#50, B#16#51, B#16#52, B#16#53, B#16#54, B#16#55, B#16#56, B#16#C1, B#16#C2)
OB72_PRIORITY	BYTE	优先级，可通过STEP 7分配(硬件配置)
OB72_OB_NUMBR	BYTE	OB编号(72)



变量	类型	描述
OB72_RESERVED_1	WORD	<p>仅适用于出错代码B#16#03:</p> <ul style="list-style-type: none"> <li>• 高字节: OB72_INFO_2和OB72_INFO_3的内容ID <ul style="list-style-type: none"> <li>- 0: OB72_INFO-2和OB72_INFO_3无任何意义</li> <li>- B#16#C4: 在故障检测模式执行后, 通过待机主站切换(对于OB72_INFO_3=W#16#0001)或不通过待机主站切换(对于OB72_INFO_3=W#16#0002)跳转到冗余模式。OB72_INFO_2被保留。</li> <li>- B#16#CD: OB72_INFO_2和OB72_INFO_3中包含优先级的实际锁定时间 &gt; 15</li> </ul> </li> <li>- 低字节: 保留</li> </ul>
OB72_INFO_1	WORD	<p>仅适用于出错代码B#16#C2:</p> <ul style="list-style-type: none"> <li>- 高字节: 超出监视时间的ID: <ul style="list-style-type: none"> <li>- 1: 扫描周期增加</li> <li>- 2: I/O死区时间</li> <li>- 3: 通讯时间延迟</li> </ul> </li> <li>- 低字节: 当前的更新尝试</li> </ul>
OB72_INFO_2	WORD	<p>仅适用于出错代码B#16#03和OB72_RESERVED_1=B#16#CD: 优先级的实际锁定时间 &gt; 15ms的高位字</p>
OB72_INFO_3	WORD	<p>仅适用于出错代码B#16#03:</p> <ul style="list-style-type: none"> <li>• OB72_RESERVED_1=B#16#C4: <ul style="list-style-type: none"> <li>- W#16#0001: 在执行故障检测模式后, <b>通过</b>待机主站切换跳转到冗余操作</li> <li>- W#16#0002: 在执行故障检测模式后, <b>不通过</b>待机主站切换跳转到冗余操</li> </ul> </li> <li>• OB72_RESERVED_1=B#16#CD: 优先级的实际锁定时间 &gt; 15ms的高位字</li> </ul>
OB82_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

下表给出了启动OB72的事件。

OB72_EV_CLASS	OB72_FLT_ID	OB72的启动事件
B#16#73	B#16#01	因CPU故障导致冗余丢失(2个中的1个)
B#16#73	B#16#02	由用户触发的保留设备上的STOP模式而导致冗余丢失(2个中的1个)
B#16#73	B#16#03	H 系统(2个中的1个)转为冗余模式
B#16#73	B#16#20	RAM比较中的出错
B#16#73	B#16#21	比较过程映像输出值时出错
B#16#73	B#16#22	比较存储器位、定时器或计数器时出错
B#16#73	B#16#23	识别出不同操作系统数据
B#16#73	B#16#31	因主站故障而导致待机主站切换
B#16#73	B#16#33	因运行期间的系统修改而导致待机主站切换
B#16#73	B#16#34	因同步模块连接问题而导致待机主站切换
B#16#73	B#16#35	由90“H_CTRL”触发的备用主站切换
B#16#73	B#16#40	因等待时间已到期而导致用户程序出现同步错误
B#16#73	B#16#41	因在不同的同步点等待而导致用户程序出现同步出错
B#16#73	B#16#42	因在不同的同步点等待而导致操作系统出现同步出错
B#16#73	B#16#43	因等待时间已到期而导致操作系统出现同步错误
B#16#73	B#16#44	因数据出错而导致操作系统出现同步出错
B#16#79	B#16#50	无SYNC模块
B#16#79	B#16#51	在未上电的情况下, 修改SYNC模块
B#16#79/B#16#78	B#16#52	删除/插入的SYNC模块
B#16#79	B#16#53	在未复位的情况下, 修改SYNC模块
B#16#79	B#16#54	SYNC模块: 机架号被分配了两次
B#16#79/B#16#78	B#16#55	SYNC模块出错/已排除
B#16#79	B#16#56	在SYNC模块上设置了非法机架号
B#16#73	B#16#C1	更新中止
B#16#73	B#16#C2	因在第n( $1 \leq n \leq$ 在因超出监视时间而导致的中止后所进行的更新尝试的最大可能次数)次尝试期间超出了监视时间而导致更新尝试中止。

## 1.14 通讯冗余出错OB(OB73)

### 注意

通讯冗余出错OB(OB73)仅在CPU 417-4H的固化程序版本V2.0.x中可用。

### 描述

当容错S7连接中发生首次冗余丢失时，H CPU的操作系统将调用OB73(只有在S7通讯中才会有容错S7连接。更多信息，请参见“S7-400 H可编程控制器，容错系统。”)。如果其它容错S7连接发生了冗余丢失，则不会再有OB73启动。

直到为具有容错功能的所有S7连接恢复冗余后，才会出现另一个OB73启动。

如果发生了启动事件且OB73没有编程，CPU不会转为STOP模式。

### CPU冗余出错OB的本地数据

下表给出了通讯冗余错误OB的临时(TEMP)变量。OB73的缺省名称已被用作变量名。

变量	类型	描述
OB73_EV_CLASS	BYTE	事件等级和ID: B#16#73、B#16#72
OB73_FLT_ID	BYTE	出错代码(可能值: B#16#E0)
OB73_PRIORITY	BYTE	分配的优先级: 缺省值为25
OB73_OB_NUMBR	BYTE	OB编号(73)
OB73_RESERVED_1	WORD	保留
OB73_INFO_1	WORD	(与用户无关)
OB73_INFO_2	WORD	(与用户无关)
OB73_INFO_3	WORD	(与用户无关)
OB73_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

下表给出了启动OB73的事件。

OB73_FLT_ID	OB 73的启动事件
B#16#E0	通讯中的冗余丢失/问题已排除

## 1.15 时间出错组织块(OB80)

### 描述

无论何时执行OB时出错，S7-300 CPU的操作系统将调用OB80。此类错误包括：超出周期时间、执行OB时出现确认错误、提前了时间而使OB的启动时间被跳过、在CiR后恢复RUN模式。例如，如果在上一次调用之后发生了某一周期性中断OB的启动事件，而同一OB此时仍在执行中，则操作系统将调用OB80。

如果OB80尚未编程，则CPU将转为STOP模式。

可以使用SFC 39至42禁用或延迟和重新启用时间出错OB。

### 注意

如果因超出了扫描时间而导致在同一扫描周期内调用了两次OB80，则CPU转为STOP模式。通过在程序中的适当的点调用SFC43“RE\_TRIGR”，可防止这种情况。

### 时间出错OB的本地数据

下表说明了时间出错OB的临时(TEMP)变量。变量名为OB80的缺省名称。

变量	类型	描述
OB80_EV_CLASS	BYTE	事件等级和标识符：B#16#35
OB80_FLT_ID	BYTE	出错代码：(可能值： B#16#01, B#16#02, B#16#05, B#16#06, B#16#07, B#16#08, B#16#09, B#16#0A)
OB80_PRIORITY	BYTE	优先级：OB 80以优先级26在RUN模式下运行，如果发生OB请求缓冲区溢出，则以优先级28运行
OB80_OB_NUMBR	BYTE	OB编号(80)
OB80_RESERVED_1	BYTE	保留
OB80_RESERVED_2	BYTE	保留
OB80_ERROR_INFO	WORD	出错信息：取决于出错代码
OB80_ERR_EV_CLASS	BYTE	导致出错的启动事件的事件等级
OB80_ERR_EV_NUM	BYTE	导致出错的启动事件的事件编号
OB80_OB_PRIORITY	BYTE	出错信息：取决于出错代码
OB80_OB_NUM	BYTE	出错信息：取决于出错代码
OB80_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

取决于错误代码的变量具有如下含义：

出错代码:	位	含义
<ul style="list-style-type: none"> <li>• B#16#01</li> <li>- OB80_ERROR_INFO:</li> <li>- OB80_ERR_EV_CLASS:</li> <li>- OB80_ERR_EV_NUM:</li> <li>- OB80_OB_PRIORITY:</li> <li>- OB80_OB_NUM</li> </ul>		<p>超过周期。</p> <p>上一扫描周期的运行时间(ms)。</p> <p>触发中断的事件的等级。</p> <p>触发中断的事件的编号。</p> <p>出错时执行的OB的优先级。</p> <p>出错时执行的OB的编号。</p>
<ul style="list-style-type: none"> <li>• B#16#02</li> <li>- OB80_ERROR_INFO:</li>   <li>- OB80_ERR_EV_CLASS:</li> <li>- OB80_ERR_EV_NUM:</li> <li>- OB80_OB_PRIORITY</li>   <li>- OB80_OB_NUM:</li> </ul>		<p>仍在执行调用的OB。</p> <p>被调用块的各个临时变量取决于下列各项</p> <ul style="list-style-type: none"> <li>• OB80_ERR_EV_CLASS和</li> <li>• OB80_ERR_EV_NUM。</li> <li>• 触发中断的事件的等级。</li> <li>• 触发中断的事件的编号。</li> <li>• 导致出错的OB的优先级(例如: 对于本应启动, 但却未能启动的OB30/优先级7, 其级别为7”。</li> <li>• 导致出错的OB的编号(例如: 对于本应启动, 但却未能启动的OB30, 其编号为30”。</li> </ul>
<ul style="list-style-type: none"> <li>• B#16#05和</li> <li>• B#16#06</li> <li>- OB80_ERROR_INFO:</li>   <li>- OB80_ERR_EV_CLASS:</li> <li>- OB80_ERR_EV_NUM:</li> <li>- OB80_OB_PRIORITY:</li> <li>- OB80_OB_NUM:</li> </ul>	置位位0: 置位位7: 位8至15:	<p>因向前拨动时钟而导致时间中断已流逝。</p> <p>在HOLD模式后返回RUN模式时时间中断已流逝。</p> <p>时钟中断0的启动时间为过去的时间。</p> <p>时钟中断7的启动时间为过去的时间。</p> <p>未用</p> <p>未用</p> <p>未用</p> <p>未用</p>
<ul style="list-style-type: none"> <li>• B#16#07</li> </ul> <p>有关参数的含义, 请参见出错代码B#16#02。</p>		<p>当前优先级的OB请求缓冲区溢出(优先级的每个OB启动请求均会被输入到相应OB请求缓冲区中; 在完成OB之后, 将会删除该条目。如果优先级的OB启动请求多于相应OB请求缓冲区中所允许的最大条目数, 则通过错误代码B#16#07调用OB80)。</p>
<ul style="list-style-type: none"> <li>• B#16#08</li> </ul> <p>有关参数的含义, 请参见出错代码B#16#02。</p>		<p>同步周期性中断时间出错</p>
<ul style="list-style-type: none"> <li>• B#16#09</li> </ul> <p>有关参数的信息, 请参考出错代码B#16#02。</p>		<p>因中断负载过高而导致中断丢失</p>
B#16#0A <ul style="list-style-type: none"> <li>• OB80_ERROR_INFO:</li> </ul>		<p>在CiR之后恢复RUN模式</p> <p>CiR同步时间(单位为ms)</p>

## 1.16 电源出错组织块(OB81)

### 描述

只要发生由错误或故障所触发的事件，而此错误或故障又与电源(仅在S7-400上)或备用电池(当事件进入和离开时)有关，则S7-300 CPU的操作系统调用OB81。

在S7-400中，如果已使用BATT.INDIC开关激活了电池测试功能，则只有在出现电池故障时才会调用OB81。

如果OB81没有编程，则CPU不会转为STOP模式。

可以使用SFC 39至42禁用或延迟，并重新启用电源出错OB。

### 电源出错OB的本地数据

下表说明了电源出错OB的临时(TEMP)变量。变量名为OB81的缺省名称。

变量	类型	描述
OB81_EV_CLASS	BYTE	事件等级和标识符： B#16#38：离开事件 B#16#39：进入事件
OB81_FLT_ID	BYTE	出错代码：(可能值) B#16#21、B#16#22、B#16#23、B#16#25、B#16#26、 B#16#27、B#16#31、B#16#32、B#16#33)
OB81_PRIORITY	BYTE	优先级，可通过STEP 7分配(硬件配置) 例如，RUN模式的可能值为：2-26
OB81_OB_NUMBR	BYTE	OB编号(81)
OB81_RESERVED_1	BYTE	保留
OB81_RESERVED_2	BYTE	保留
OB81_RACK_CPU	WORD	<ul style="list-style-type: none"> <li>位0至2：机架号</li> <li>位3：0= 待机CPU，1= 主CPU</li> <li>位4至7：1111</li> </ul>
OB81_RESERVED_3	BYTE	仅与错误代码B#16#31、B#16#32以及B#16#33相关
OB81_RESERVED_4	BYTE	
OB81_RESERVED_5	BYTE	
OB81_RESERVED_6	BYTE	
OB81_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

变量OB81\_RESERVED\_i (3 £ I £ 6)指示其中的备用电池(错误代码B#16#31)、备用电压(错误代码B#16#32)或24V电源(错误代码B#16#33)出现故障或返回的扩展机架。下表给出了为扩展机分配变量OB81\_RESERVED\_i (3 i 6)中各个位的情况。

	OB81_RESERVED_6	OB81_RESERVED_5	OB81_RESERVED_4	OB81_RESERVED_3
位0	保留	扩展机架8	扩展机架16	保留
位1	扩展机架1	扩展机架9	扩展机架17	保留
位2	扩展机架2	扩展机架10	扩展机架18	保留
位3	扩展机架3	扩展机架11	扩展机架19	保留
位4	扩展机架4	扩展机架12	扩展机架20	保留
位5	扩展机架5	扩展机架13	扩展机架21	保留
位6	扩展机架6	扩展机架14	保留	保留
位7	扩展机架7	扩展机架15	保留	保留

变量OB81\_RESERVED\_i中各个位的含义(对于相关的扩展机架)如下:

当事件发生时,将对其中至少有一个电池或备用电压或24伏电源出现故障的扩展机架进行标记(置位相应的位)。对于其中至少有一个电池或备用电压或24伏电源在先前发生过故障的扩展机架,将不再对其进行标记。

当排除了事件并且至少在一个扩展机架上恢复备份后,将就此发出信号(置位相应的位)。

下表给出了启动OB81的事件:

OB81_EV_CLASS	OB81_FLT_ID	含义
B#16#39/B#16#38	B#16#21:	至少一个中央机架的备用电池电量耗尽/问题已排除(BATTF) 注意: 只要两个电池中有一个电池出现故障,就会发生此事件(如果存在冗余的备用电池)。如果第二个电池恰巧也发生了故障,则该事件不会再次出现。
B#16#39/B#16#38	B#16#22:	中央机架中的备用电压出现故障/问题已排除(BAF)
B#16#39/B#16#38	B#16#23:	中央机架中的24伏电源出现故障/问题已排除。
B#16#39/B#16#38	B#16#25:	至少在一个冗余中央机架中的至少有一个备用电池的电量耗尽/问题已排除(BATTF)
B#16#39/B#16#38	B#16#26:	至少一个冗余中央机架中的备用电压出现故障/问题已排除(BAF)
B#16#39/B#16#38	B#16#27:	至少一个冗余中央机架中的24伏电源出现故障
B#16#39/B#16#38	B#16#31:	至少在一个扩展机架中的至少有一个备用电池的电量耗尽/问题已排除(BATTF)。
B#16#39/B#16#38	B#16#32:	至少一个扩展机架中的备用电压出现故障/问题已排除(BAF)
B#16#39/B#16#38	B#16#33:	至少一个扩展机架中的24伏电源出现故障/问题已排除。

## 1.17 诊断中断组织块(OB82)

### 描述

如果具有诊断功能的模块(已为其启用了诊断中断)检测到错误, 则它会输出一个诊断中断的请求给CPU(当事件进入和离开时)。则操作系统调用OB82。

OB82的局部变量包含逻辑基址和四字节的故障模块的诊断数据(请参见下表)。

如果OB82尚未编程, 则CPU转为STOP模式。

可以使用SFC 39至42禁用或延迟, 并重新启用诊断中断OB。

### 诊断中断OB的本地数据

下表说明了诊断中断OB的临时(TEMP)变量。变量名为OB82的缺省名称。

变量	类型	描述
OB82_EV_CLASS	BYTE	事件等级和标识符: <ul style="list-style-type: none"> <li>• B#16#38: 离开事件</li> <li>• B#16#39: 进入事件</li> </ul>
OB82_FLT_ID	BYTE	出错代码(B#16#42)
OB82_PRIORITY	BYTE	• 优先级, 可通过STEP 7分配(硬件配置)
OB82_OB_NUMBR	BYTE	OB编号(82)
OB82_RESERVED_1	BYTE	保留
OB82_IO_FLAG	BYTE	<ul style="list-style-type: none"> <li>• 输入模块: B#16#54</li> <li>• 输出模块: B#16#55</li> </ul>
OB82_MDL_ADDR	WORD	发生故障的模块的逻辑基址
OB82_MDL_DEFECT	BOOL	模块发生故障
OB82_INT_FAULT	BOOL	内部故障
OB82_EXT_FAULT	BOOL	外部故障
OB82_PNT_INFO	BOOL	通道故障
OB82_EXT_VOLTAGE	BOOL	外部电压故障
OB82_FLD_CONNCTR	BOOL	未插入前面板连接器
OB82_NO_CONFIG	BOOL	未组态模块
OB82_CONFIG_ERR	BOOL	模块中的参数不正确
OB82_MDL_TYPE	BYTE	<ul style="list-style-type: none"> <li>• 位0至3: 模块等级</li> <li>• 位4: 存在通道信息</li> <li>• 位5: 存在用户信息</li> <li>• 位6: 来自替换者的诊断中断</li> <li>• 位7: 保留</li> </ul>
OB82_SUB_MDL_ERR	BOOL	子模块丢失或存在出错
OB82_COMM_FAULT	BOOL	通讯问题



变量	类型	描述
OB82_MDL_STOP	BOOL	操作模式(0: RUN, 1: STOP)
OB82_WTCH_DOG_FLT	BOOL	内部电源故障
OB82_INT_PS_FLT	BOOL	内部电源发生故障
OB82_PRIM_BATT_FLT	BOOL	电池用尽
OB82_BCKUP_BATT_FLT	BOOL	整个备份失败
OB82_RESERVED_2	BOOL	保留
OB82_RACK_FLT	BOOL	扩展机架故障
OB82_PROC_FLT	BOOL	处理器故障
OB82_EPROM_FLT	BOOL	EPROM故障
OB82_RAM_FLT	BOOL	RAM故障
OB82_ADU_FLT	BOOL	ADC/DAC出错
OB82_FUSE_FLT	BOOL	保险丝断开
OB82_HW_INTR_FLT	BOOL	硬件中断丢失
OB82_RESERVED_3	BOOL	保留
OB82_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

### 注意

如果使用了具有DPV1功能的CPU，可借助于SFB54 "RALRM"获取有关中断的其它信息，这些信息不止包含OB的起始信息。在S7兼容模式下操作DP主站时，这一点也同样适用。

## 1.18 插入/删除模块中断组织块(OB83)

### 描述

在下列情况下，CPU操作系统会调用OB 83：

- 插入/删除已组态模块后
- 在STEP 7下修改模块参数以及在运行期间将更改下载至CPU后

可借助SFC 39至42禁用/延迟/启用插入/删除中断OB。

### 模块插入和删除

每次在RUN、STOP和STARTUP模式期间删除或插入已组态模块时，都会生成插入/删除中断(在这些模式下一定不能删除电源模块、CPU、适配器模块以及IM)。此中断会导致在诊断缓冲区和所涉及CPU的系统状态列表中生成一个条目。如果CPU处于RUN模式，则还会启动插入/删除OB。如果此OB尚未编程，CPU将转为STOP模式。

随后，系统会以数秒钟的时间间隔来轮询S7-400模块以检测插入或删除。为使CPU能够检测到S7-400模块的删除和插入，在删除和插入操作之间必须至少有2s的时间间隔。对于其它模块而言，此最短时间值会稍大一些。

如果在RUN模式下删除了已组态的模块，则启动OB83。由于仅以1s为时间间隔来监视模块的存在，因此当直接访问模块或更新过程映像时，可能会首先检测到访问错误。

如果在RUN模式下将一个模块插入到已组态的插槽中，则操作系统将检查所插入模块的类型是否与所记录的组态相符。如果模块类型匹配，则启动OB83并分配参数。

### 重新组态模块

当在运行(CiR)期间修改系统组态时，可为现有模块重新分配参数。参数的这种重新分配通过将所需参数数据记录传送至模块实现。

步骤如下：

1. 在STEP 7下已为模块分配了新参数，且在RUN模式中将此组态下载到CPU时，将启动OB 83(启动事件W#16#3367)。相关的OB启动信息为逻辑基址(OB83\_MDL\_ADDR)和模块类型(OB83\_MDL\_TYPE)。从现在起，模块I/O数据可能不正确，这就意味着没有任何SFC忙于将数据记录发送到此模块。
2. 在执行OB 83后将重新分配模块参数。

3. 在分配参数后，将重新启动OB 83(假如成功地分配参数，则启动事件W#16#3267，如果失败，则启动事件W#16#3968)。模块的I/O数据响应与其在插入中断后的响应相同，也就是说，当前它们可能不正确。现在可再次调用SFC以便将数据记录发送到模块。

### OB 83的本地数据

下表说明了插入/删除模块中断OB的临时(TEMP)变量。变量名为OB 83的缺省名称。

变量	类型	描述
OB83_EV_CLASS	BYTE	事件等级和标识符： <ul style="list-style-type: none"> <li>• B#16#32: 重新分配模块参数结束</li> <li>• B#16#33: 重新分配模块参数开始</li> <li>• B#16#38: 模块已插入</li> <li>• B#16#39: 模块被删除或未响应，或者参数分配结束</li> </ul>
OB83_FLT_ID	BYTE	出错代码：(可能值：B#16#61, B#16#63, B#16#64, B#16#65, B#16#67, B#16#68, B#16#84)
OB83_PRIORITY	BYTE	<ul style="list-style-type: none"> <li>• 优先级，可通过STEP 7分配(硬件配置)</li> </ul>
OB83_OB_NUMBR	BYTE	OB编号(83)
OB83_RESERVED_1	BYTE	模块或子模块/接口模块的标识
OB83_MDL_TD	BYTE	范围： <ul style="list-style-type: none"> <li>• B#16#54: 外围输入(PI)</li> <li>• B#16#55: 外设输出(PQ)</li> </ul>
OB83_MDL_ADDR	WORD	受影响模块的逻辑基址； 如果它是混合模块，则该地址为模块中所使用的最小逻辑地址。如果混合块中的I地址和O地址相等，则逻辑基址为接收事件标识符的地址。
OB83_RACK_NUM	WORD	<ul style="list-style-type: none"> <li>• 如果OB83_RESERVED_1 = B#16#A0: 接口模块编号</li> <li>• 如果OB83_RESERVED_1 = B#16#C4: 机架号或DP站的编号(低字节)和DP主站系统标识号(高字节)</li> </ul>
OB83_MDL_TYPE	WORD	受影响模块的模块类型： <ul style="list-style-type: none"> <li>• W#16#X5XX: 模拟模块</li> <li>• W#16#X8XX: 功能模块</li> <li>• W#16#XCXX: CP</li> <li>• W#16#XFXX: 数字模块</li> </ul> X: 与用户无关的数值
OB83_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

下表给出了启动OB83的事件：

OB83_EV_CLASS	OB83_FLT_ID	含义
B#16#39	B#16#61	模块已删除或未响应 OB83_MDL_TYPE: 实际模块类型
B#16#38	B#16#61	模块已插入。模块类型正确 OB83_MDL_TYPE: 实际模块类型
B#16#38	B#16#63	模块已插入，但模块类型不正确 OB83_MDL_TYPE: 实际模块类型
B#16#38	B#16#64	模块已插入，但存在问题(无法读取模块ID) OB83_MDL_TYPE: 已组态的模块类型
B#16#38	B#16#65	模块已插入，但模块参数分配有出错 OB83_MDL_TYPE: 实际模块类型
B#16#39	B#16#66	模块未响应，负载电压出错
B#16#38	B#16#66	模块再次响应，已更正负载电压出错
B#16#33	B#16#67	启动模块重新组态
B#16#32	B#16#67	结束模块重新组态
B#16#39	B#16#68	模块重新组态因出错而终止
B#16#38	B#16#84	接口模块已插入
B#16#39	B#16#84	接口模块已删除

### 注意

如果使用了具有DPV1或PROFINET功能的CPU，可借助于SFB54“RALRM”获取有关中断的其它信息，这些信息不止包含OB的起始信息。在S7兼容模式下操作DP主站时，这一点也同样适用。

## 1.19 CPU硬件故障组织块(OB84)

### 描述

在下列情况下，CPU中的OS将调用OB84：

- 已检测到并更正了内存出错之后
- 对于S7-400H：如果两个CPU之间的冗余链接的性能下降

可以使用SFC 39至42禁用或延迟CPU硬件出错OB，然后再次启用它。

### 硬件故障OB的本地数据

下表包括了CPU硬件故障的临时(TEMP)变量。变量名为OB84的缺省名称。

变量	类型	描述
OB84_EV_CLASS	BYTE	事件等级和标识符： <ul style="list-style-type: none"> <li>• B#16#38：离开事件</li> <li>• B#16#35、B#16#39 进入事件</li> </ul>
OB84_FLT_ID	BYTE	出错代码(B#16#81、B#16#82、B#16#83、B#16#85、B#16#86、B#16#87)
OB84_PRIORITY	BYTE	优先级，可通过STEP 7分配(硬件配置)
OB84_OB_NUMBR	BYTE	OB编号(84)
OB84_RESERVED_1	BYTE	保留
OB84_RESERVED_2	BYTE	保留
OB84_RESERVED_3	WORD	保留
OB84_RESERVED_4	DWORD	保留
OB84_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

下表给出了启动OB84的事件：

OB84_EV_CLASS	OB84_FLT_ID	OB 84的启动事件
B#16#39	B#16#81	接口出错，进入
B#16#38	B#16#81	接口出错，离开
B#16#35	B#16#82	在操作系统中检测到内存出错并已更正
B#16#35	B#16#83	已检测到并更正的内存出错的累积
B#16#35	B#16#85	PC操作系统中的出错
B#16#39	B#16#86	H-Sync链接的性能受到负面影响
B#16#35	B#16#87	已检测到多位内存出错并已更正

## 1.20 优先级出错组织块(OB85)

### 描述

只要发生下列事件之一，CPU的操作系统即调用OB85：

- 尚未装载的OB(OB81除外)的启动事件。
- 操作系统访问模块时出错。
- 在系统更新过程映像期间出现I/O访问错误(如果由于组态原因，未禁止OB85的调用)。

### 注意

如果OB85尚未编程，则当检测到这些事件其中之一时，CPU会转为STOP模式。

可以使用SFC 39至42禁用或延迟，并重新启用优先级出错OB。

### 优先级出错OB的本地数据

下表说明了优先级出错OB的临时(TEMP)变量。变量名为OB85的缺省名称。

变量	类型	描述
OB85_EV_CLASS	BYTE	事件等级和标识符：B#16#35 B#16#38 (仅限于出错代码B#16#B3和B#16#B4) B#16#39(仅限于出错代码B#16#B1、B#16#B2、 B#16#B3和B#16#B4)
OB85_FLT_ID	BYTE	出错代码(可能值：B#16#A1、B#16#A2、B#16#A3、 B#16#A4、B#16#B1、B#16#B2、B#16#B3、B#16#B4)
OB85_PRIORITY	BYTE	• 优先级，可通过STEP 7分配(硬件配置)
OB85_OB_NUMBR	BYTE	OB编号(85)
OB85_RESERVED_1	BYTE	保留
OB85_RESERVED_2	BYTE	保留
OB85_RESERVED_3	INT	保留
OB85_ERR_EV_CLASS	BYTE	引起出错的事件的等级
OB85_ERR_EV_NUM	BYTE	引起出错的事件的编号
OB85_OB_PRIOR	BYTE	出错时处于激活状态的OB的优先级
OB85_OB_NUM	BYTE	出错时处于激活状态的OB的编号
OB85_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

如果要对依赖于可能错误代码的OB85进行编程，建议按如下方式组织局部变量：

变量	类型
OB85_EV_CLASS	BYTE
OB85_FLT_ID	BYTE
OB85_PRIORITY	BYTE
OB85_OB_NUMBR	BYTE
OB85_DKZ23	BYTE
OB85_RESERVED_2	BYTE
OB85_Z1	WORD
OB85_Z23	DWORD
OB85_DATE_TIME	DATE_AND_TIME

下表给出了启动OB85的事件：

OB85_EV_CLASS	OB85_FLT_ID	含义
B#16#35	B#16#A1	<p>由于使用STEP 7创建了组态，用户的程序或操作系统将为未加载到CPU上的OB创建一个启动事件。</p> <ul style="list-style-type: none"> <li>OB85_Z1：由OB85_Z23确定的被调用OB的各个局部变量。</li> <li>OB85_Z23： <ul style="list-style-type: none"> <li>高位字：导致OB调用的事件的等级和编号</li> <li>低位字，高字节：程序级别以及出错时处于活动状态的OB</li> <li>低位字，低字节：激活的OB</li> </ul> </li> </ul>
B#16#35	B#16#A2	<p>由于使用STEP 7创建了组态，用户的程序或操作系统将为未装载到CPU上的OB创建一个启动事件。</p> <p>OB85_Z1和OB85_Z23(对于OB85_FLT_ID=B#16#A1)</p>
B#16#35	B#16#A3	<p>操作系统访问模块时出错</p> <ul style="list-style-type: none"> <li>OB85_Z1：操作系统的出错ID <ul style="list-style-type: none"> <li>高字节：1= 集成的功能，2=IEC定时器</li> <li>低字节：0= 无故障分辨率，1= 未装载块，2= 区域长度出错，3= 写保护出错</li> </ul> </li> <li>OB85_Z23： <ul style="list-style-type: none"> <li>高位字：块编号</li> <li>低位字：导致出错的MC7命令的相对地址。块类型必须取自OB 85_DKZ23(B#16#88：OB，B#16#8C：FC，B#16#8E：FB，B#16#8A：DB)。</li> </ul> </li> </ul>
B#16#35	B#16#A4	无法对PROFINet接口DB进行寻址
B#16#34	B#16#A4	可以再次对PROFINet接口DB进行寻址

OB85_EV_CLASS	OB85_FLT_ID	含义
B#16#39	B#16#B1	<p>更新输入的过程映像时出现I/O访问出错</p> <ul style="list-style-type: none"> <li>• OB85_DKZ23: 过程映像传送(在此期间发生I/O访问出错)类型的ID <ul style="list-style-type: none"> <li>- B#16#10: 字节访问</li> <li>- B#16#20: 字访问</li> <li>- B#16#30: 双字访问</li> <li>- B#16#57: 传送已组态的一致性范围</li> </ul> </li> <li>• OB85_Z1: 留作CPU内部使用: 模块的逻辑基址 如果OB85_RESERVED_2的值为B#16#76, 则OB85_Z1会接收到受影响SFC(SFC14、15、26或27)的返回值。</li> <li>• OB85_Z23: <ul style="list-style-type: none"> <li>- 字节0: 部分过程映像编号</li> <li>- 字节1: 如果OB85_DKZ23=B#16#10、20或30, 则不相关; 如果OB85_DKZ23=B#16#57, 则为以字节为单位的一致性范围的长度</li> <li>- 字节2和3: 如果OB85_DKZ23=B#16#10、20或30, 则为导致PAE的I/O地址; 如果OB85_DKZ23=B#16#57, 则为一致性范围的逻辑起始地址</li> </ul> </li> </ul>
B#16#39	B#16#B2	<p>当将输出过程映像传送到输出模块时出现I/O访问出错</p> <p>OB85_DKZ23、OB85_Z1和OB85_Z23(对于OB85_FLT_ID=B#16#B1)</p>
<p>如果您已为系统过程映像表更新组态了I/O访问出错的重复OB85调用, 则会获得出错代码B#16#B1和B#16#B2。</p>		
B#16#39/B#16#38	B#16#B3	<p>当更新输入的过程映像、进入/离开事件时出现I/O访问出错</p> <ul style="list-style-type: none"> <li>• OB85_DKZ23: 过程映像传送(在此期间发生I/O访问出错)类型的ID <ul style="list-style-type: none"> <li>- B#16#10: 字节访问</li> <li>- B#16#20: 字访问</li> <li>- B#16#30: 双字访问</li> <li>- B#16#57: 传送组态的一致性范围</li> </ul> </li> <li>• OB85_Z1: 留作CPU内部使用: 模块的逻辑基址 如果OB85_RESERVED_2的值为B#16#76, 则OB85_Z1会接收到受影响SFC(SFC14、15、26或27)的返回值。</li> <li>• OB85_Z23: <ul style="list-style-type: none"> <li>- 字节0: 部分过程映像编号</li> <li>- 字节1: 如果OB85_DKZ23=B#16#10、20或30, 则不相关; 如果OB85_DKZ23=B#16#57, 则为以字节为单位的一致性范围的长度</li> <li>- 字节2和3: 如果OB85_DKZ23=B#16#10、20或30, 则为导致PAE的I/O地址; 如果OB85_DKZ23=B#16#57, 则为一致性范围的逻辑起始地址</li> </ul> </li> </ul>



OB85_EV_CLASS	OB85_FLT_ID	含义
B#16#39/B#16#38	B#16#B4	当更新输出的过程映像时出现I/O访问出错，进入/离开事件 OB85_DKZ23、OB85_Z1、OB85_Z23(对于 OB85_FLT_ID=B#16#B3)
如果已为由系统进行的过程映像表更新组态了I/O访问错误进入和离开事件的OB85调用，则会获得错误代码B#16#B3和B#16#B4。冷重启或暖重启后，在下次更新过程映像表期间，对于非现有输入和输出的所有访问均报告为I/O访问错误。		

## 1.21 机架故障组织块(OB86)

### 描述

只要在分布式I/O (PROFIBUS DP或PROFINET IO)中检测到中央扩展机架(不带S7-300)、DP主站系统或站故障(进入事件与离开事件时), CPU的操作系统调用OB86。

如果OB86尚未编程, 当检测到此种类型的出错时, CPU将转为STOP模式。

可使用SFC 39至42禁用或延迟, 并重新启用OB86。

### 机架故障OB的本地数据

下表说明了机架故障OB的临时(TEMP)变量。变量名为OB86的缺省名称。

变量	类型	描述
OB86_EV_CLASS	BYTE	事件等级和标识符: <ul style="list-style-type: none"> <li>• B#16#38: 离开事件</li> <li>• B#16#39: 进入事件</li> </ul>
OB86_FLT_ID	BYTE	出错代码: (可能的数值: B#16#C1, B#16#C2, B#16#C3, B#16#C4, B#16#C5, B#16#C6, B#16#C7, B#16#C8)
OB86_PRIORITY	BYTE	<ul style="list-style-type: none"> <li>• 优先级, 可通过STEP 7分配(硬件配置)</li> </ul>
OB86_OB_NUMBR	BYTE	OB编号(86)
OB86_RESERVED_1	BYTE	保留
OB86_RESERVED_2	BYTE	保留
OB86_MDL_ADDR	WORD	取决于出错代码
OB86_RACKS_FLTD	数组[0..31] 布尔型	取决于出错代码
OB86_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

如果要对依赖于可能错误代码的OB86进行编程, 建议按如下方式组织局部变量:

变量	类型
OB86_EV_CLASS	BYTE
OB86_FLT_ID	BYTE
OB86_PRIORITY	BYTE
OB86_OB_NUMBR	BYTE
OB86_RESERVED_1	BYTE
OB86_RESERVED_2	BYTE
OB86_MDL_ADDR	WORD
OB86_Z23	DWORD
OB86_DATE_TIME	DATE_AND_TIME

下表给出了启动OB86的事件:

OB86_EV_CLASS	OB86_FLT_ID	含义
B#16#39	B#16#C1	<p>扩展机架故障</p> <ul style="list-style-type: none"> <li>• OB86_MDL_ADDR: IM的逻辑基址</li> <li>• OB86_Z23: 针对于每个可能的扩展机架包含一个位: 导致调用OB86的每个扩展机架均报告为已出现故障(置位各个位)。不再显示之前出现故障的扩展机架。 <ul style="list-style-type: none"> <li>- 位0: 始终为0</li> <li>- 位1: 扩展机架1</li> <li>- :</li> <li>- 位21: 扩展机架21</li> <li>- 位22至29: 始终为0</li> <li>- 位30: 在SIMATIC S5区域中至少有一个扩展机架出现了故障</li> <li>- 位31: 始终为0</li> </ul> </li> </ul>
B#16#38	B#16#C1	<p>扩展机架再次可运行</p> <p>OB86_MDL_ADDR(对于OB86_FLT_ID=B#16#C1)。再次可运行的扩展机架在OB86_Z23中进行报告(置位各个位)。</p>
B#16#38	B#16#C2	<p>扩展机架再次可运行(扩展机架因预期组态与实际组态不符而出现故障)</p> <ul style="list-style-type: none"> <li>• OB86_MDL_ADDR: IM的逻辑基址</li> <li>• OB86_Z23: 针对每个可能扩展机架包含一个位, 请参见OB86_FLT_ID B#16#C1。 已置位的含义: 在受影响机架中: <ul style="list-style-type: none"> <li>- 存在具有出错类型ID的模块</li> <li>- 已组态的模块丢失</li> <li>- 至少有一个模块故障。</li> </ul> </li> </ul>
B#16#39	B#16#C3	<p>分布式I/O设备: DP主站系统故障</p> <p>只有进入事件会导致启动出错代码为B#16#C3的OB 86。离开事件可启动出错代码为B#16#C4, 且事件等级为B#16#38的OB 86: 每个DP从站的返回会启动OB86</p> <ul style="list-style-type: none"> <li>• OB86_MDL_ADDR: DP主站的逻辑基址</li> <li>• OB86_Z23: DP主站ID: <ul style="list-style-type: none"> <li>- 位0至7: 保留</li> <li>- 位8至15: DP主站系统标识号</li> <li>- 位16至31: 保留</li> </ul> </li> </ul>
B#16#39/B#16#38	B#16#C4	<p>DP站故障</p> <ul style="list-style-type: none"> <li>• OB86_MDL_ADDR: DP主站的逻辑基址</li> <li>• OB86_Z23: 受影响DP从站的地址: <ul style="list-style-type: none"> <li>- 位0至7: DP站编号</li> <li>- 位8至15: DP主站系统标识号</li> <li>- 位16至30: S7从站的逻辑基址或者标准DP从站的诊断地址</li> <li>- 位31: I/O标识符</li> </ul> </li> </ul>

OB86_EV_CLASS	OB86_FLT_ID	含义
B#16#39/B#16#38	B#16#C5	DP站故障 OB86_MDL_ADDR和OB86_Z23(对于FLT_ID=B#16#C4)
B#16#38	B#16#C6	扩展机架可再次运行，但模块参数仍存在错误。 <ul style="list-style-type: none"> <li>• OB86_MDL_ADDR: IM的逻辑基址</li> <li>• OB86_Z23: 包含针对于每个可能的扩展机架的一个位:                             <ul style="list-style-type: none"> <li>- 位0: 始终为0</li> <li>- 位1: 扩展机架1</li> <li>- :</li> <li>- 位21: 扩展机架21</li> <li>- 位22至30: 保留</li> <li>- 位31: 始终为0</li> </ul> </li> <li>• 置位时的含义(在所涉及的扩展机架中):                             <ul style="list-style-type: none"> <li>- 存在具有出错类型标识符的模块</li> <li>- 存在缺少参数或具有出错参数的模块。</li> </ul> </li> </ul>
B#16#38	B#16#C7	返回DP站，但模块参数分配存在出错 <ul style="list-style-type: none"> <li>• OB86_MDL_ADDR: DP主站的逻辑基址</li> <li>• 受影响DP从站的地址:                             <ul style="list-style-type: none"> <li>- 位0至7: DP站编号</li> <li>- 位8至15: DP主站系统标识号</li> <li>- 位16至30: DP从站的逻辑基址</li> <li>- 位31: I/O标识符</li> </ul> </li> </ul>
B#16#38	B#16#C8	返回DP站，但所配置的组态和实际组态不符 <ul style="list-style-type: none"> <li>• OB86_MDL_ADDR: DP主站的逻辑基址</li> <li>• OB86_Z23: 受影响DP从站的地址:                             <ul style="list-style-type: none"> <li>- 位0至7: DP站编号</li> <li>- 位8至15: DP主站系统标识号</li> <li>- 位16至30: DP从站的逻辑基址</li> <li>- 位31: I/O标识符</li> </ul> </li> </ul>

**注意**

如果使用呢具有DPV1功能的CPU，可借助于SFB54 “RALRM”获取有关中断的其它信息，这些信息不止包含OB的起始信息。在S7兼容模式下操作DP主站时，这一点也同样适用。

## 1.22 通讯出错组织块(OB87)

### 描述

只要发生由通讯出错导致的事件，CPU的操作系统就会调用OB87。

如果OB87尚未编程，CPU不会转为STOP模式。

可以使用SFC 39至42禁用或延迟，并重新启用通讯出错OB。

### OB87的本地数据

下表说明了通讯出错OB的临时(TEMP)变量。变量名为OB87的缺省名称。

变量	类型	描述
OB87_EV_CLASS	BYTE	事件等级和标识符: B#16#35
OB87_FLT_ID	BYTE	出错代码: (可能值: B#16#D2、B#16#D3、B#16#D4、B#16#D5、 B#16#E1、B#16#E2、B#16#E3、B#16#E4、B#16#E5、 B#16#E6)
OB87_PRIORITY	BYTE	• 优先级, 可通过STEP 7分配(硬件配置)
OB87_OB_NUMBR	BYTE	OB编号(87)
OB87_RESERVED_1	BYTE	保留
OB87_RESERVED_2	BYTE	保留
OB87_RESERVED_3	WORD	取决于出错代码
OB87_RESERVED_4	DWORD	取决于出错代码
OB87_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

取决于错误代码的变量具有如下含义:

出错代码:	字节/字	含义
<ul style="list-style-type: none"> <li>• B#16#D2:</li> <li>• B#16#D3</li> <li>• B#16#D4:</li> <li>• B#16#D5</li> <li>OB87_RESERVED_3:</li> <li>OB87_RESERVED_4:</li> </ul>		当前无法传送诊断条目。 无法传送同步消息(主站)。 因时钟同步而导致非法时间跳转。 接收同步时间时出错(从站)。 未包含更多的信息。 未包含更多的信息。
<ul style="list-style-type: none"> <li>• B#16#E1:</li> <li>• B#16#E3:</li> <li>• B#16#E4:</li> <li>OB87_RESERVED_3:</li> <li>OB87_RESERVED_4:</li> </ul>	高字节: 低字节:	在全局数据通讯期间帧ID不正确。 在全局数据通讯期间帧长度出错。 收到非法的GD信息包编号。 接口ID(0: K总线, 1: MPI) GD电路编号 未包含更多的信息。

出错代码:	字节/字	含义
<ul style="list-style-type: none"> <li>B#16#E2:</li> <li>OB87_RESERVED_3:</li> <li>OB87_RESERVED_4:</li> </ul>	高位字: 低位字:	无法将GD信息包状态输入到DB中 DB编号 未包含更多的信息。 <ul style="list-style-type: none"> <li>GD回路数目(高字节),</li> <li>GD信息包编号(低字节)</li> </ul>
<ul style="list-style-type: none"> <li>B#16#E5:</li> <li>OB87_RESERVED_3:</li> <li>OB87_RESERVED_4:</li> </ul>	高位字: 低位字:	通过通讯功能块进行数据交换期间出现DB访问出错 留作CPU内部使用。 包含导致出错的MC7命令的块编号。 导致出错的MC7命令的相对地址。

块类型可从OB\_87\_RESERVED\_1读取(B#16#88: OB, B#16#8A: DB, B#16#8C: FC, B#16#8E: FB)。

出错代码:	含义
<ul style="list-style-type: none"> <li>B#16#E6:</li> <li>OB87_RESERVED_3:</li> <li>OB87_RESERVED_4:</li> </ul>	无法将GD组状态输入到DB中。 DB编号。 未包含更多的信息。

## 1.23 处理中断OB(OB 88)

### 描述

程序块执行被中止后，CPU操作系统将调用OB 88。导致此中断的原因可能是：

- 同步出错的嵌套深度过大
- 块调用(U堆栈)的嵌套深度过大
- 分配本地数据时出错

如果未对OB 88编程且程序块执行被中止，则CPU进入STOP模式(事件ID W#16#4570)。

如果在优先级28下中止了程序块执行，则CPU进入STOP模式。

可借助于SFC 39至42禁用、延迟和启用处理中断OB。

### 处理中断OB的本地数据

下表给出了处理中断OB的临时(TEMP)变量。OB 88的缺省名称被选为变量名。

变量	数据类型	描述
OB88_EV_CLASS	BYTE	事件等级和ID: B#16#35
OB88_SW_FLT	BYTE	出错代码 可能值: <ul style="list-style-type: none"> <li>• B#16#73: 同步出错的嵌套深度过大</li> <li>• B#16#75: 块调用(B堆栈)的嵌套深度过大</li> <li>• B#16#76: 分配本地数据时出错</li> </ul>
OB88_PRIORITY	BYTE	• 优先级: 28
OB88_OB_NUMBR	BYTE	OB编号(88)
OB88_BLK_TYPE	BYTE	出现出错的块的类型: <ul style="list-style-type: none"> <li>• B#16#88: OB</li> <li>• B#16#8C: FC</li> <li>• B#16#8E: FB</li> <li>• B#16##00: 无法确定中断源</li> </ul>
OB88_RESERVED_1	BYTE	保留
OB88_FLT_PRIORITY	BYTE	导致出错的OB的优先级
OB88_FLT_OB_NUMBR	BYTE	导致出错的OB的编号
OB88_BLK_NUM	WORD	具有导致出错的MC7指令的块编号
OB88_PRG_ADDR	WORD	导致出错的MC7指令的相对地址
OB88_DATE_TIME	DATE_AND_TIME	OB调用的日期和TOD

## 1.24 后台组织块(OB90)

### 描述

使用STEP 7，可以监视最大扫描周期并能确保最小扫描周期。如果包含所有嵌套中断和系统活动在内的OB1的执行时间少于指定的最小扫描周期，则操作系统将作出如下反应：

- 调用后台OB(如果它存在于CPU中)。
- 延迟下一次OB1启动(如果OB90在CPU中不存在)。

### 了解OB90的操作

在所有OB中，OB90的优先级最低。任何系统活动和中断都会将其中断(甚至在最小周期到期后由OB1中断)，并且只有在尚未达到所选最小扫描周期的情况下才会恢复。对此有一个例外，SFC和SFB的执行在OB90中启动。这两项以OB1的优先级执行，因此不会被OB1中断。在此没有OB90的时间监视。

在下列情况下，将从第一条指令开始处理OB90中的用户程序：

- 暖重启、冷重启或热重启后
- 在删除一个正在OB90中执行的块之后(用STEP 7)
- 在RUN模式下将OB90装载到CPU之后
- 终止后台循环后

---

### 注意

对于最小扫描周期与周期监视时间差别不大的组态，后台OB中的SFC和SFB调用有可能意外地超出周期时间。

---



## OB90的本地数据

下表介绍了OB90的临时(TEMP)变量。变量名为OB90的缺省名称。

变量	数据类型	描述
OB90_EV_CLASS	BYTE	事件等级和标识符: B#16#11: 激活
OB90_STRT_INF	BYTE	<ul style="list-style-type: none"> <li>• B#16#91: 暖重启/冷重启/热重启</li> <li>• B#16#92: 已删除块</li> <li>• B#16#93: 在RUN模式下将OB90下载到CPU</li> <li>• B#16#95: 终止后台循环</li> </ul>
OB90_PRIORITY	BYTE	优先级: 29(对应于优先级0.29)
OB90_OB_NUMBR	BYTE	OB编号(90)
OB90_RESERVED_1	BYTE	保留
OB90_RESERVED_2	BYTE	保留
OB90_RESERVED_3	INT	保留
OB90_RESERVED_4	INT	保留
OB90_RESERVED_5	INT	保留
OB90_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

## 1.25 启动组织块(OB100、OB101和OB102)

### 启动类型

以下几种启动类型有所区别

- 热重启(不使用S7-300和S7-400H)
- 冷重启
- 冷重启

下表中列出了启动过程中操作系统将调用哪个OB。

启动类型	对应的OB
热重启	OB101
暖重启	OB100
冷重启	OB102

有关启动类型的更多详细信息，请参阅手册“**使用STEP 7编程**”、“**使用STEP7组态硬件和连接**”以及“**S7-400H PLC**”。

### 启动事件

CPU按如下所述执行启动：

- 上电后
- 只要将模式选择器由STOP切换为RUN-P时
- 使用通讯功能(编程设备中的菜单命令或者通过调用不同CPU上的通讯功能块19“START”或21“RESUME”)发出请求后。
- 多值计算的同步
- 在链接之后的H系统中(仅适用于待机的CPU)

根据启动事件、特定的CPU及其参数，将调用适当的启动OB (OB100、OB101或OB102)。通过相配的编程，可针对循环程序进行某些设置(例外：在H系统中，当待机CPU已链接时，在该待机CPU上存在启动，但却不会调用启动OB)。

## 启动OB的本地数据

下表介绍了启动OB的临时(TEMP)变量。变量名为OB100的缺省名称。

变量	类型	描述
OB10x_EV_CLASS	BYTE	事件等级和标识符: <b>B#16#13</b> : 激活
OB10x_STARTUP	BYTE	启动请求: <ul style="list-style-type: none"> <li>• B#16#81: 手动暖重启</li> <li>• B#16#82: 自动暖重启</li> <li>• B#16#83: 请求手动热重启</li> <li>• B#16#84: 请求自动热重启</li> <li>• B#16#85: 请求手动冷重启</li> <li>• B#16#86: 请求自动冷重启</li> <li>• B#16#87: 主站: 请求手动冷重启</li> <li>• B#16#88: 主站: 请求自动冷重启</li> <li>• B#16#8A: 主站: 请求手动暖重启</li> <li>• B#16#8B: 主站: 请求自动暖重启</li> <li>• B#16#8C: 备用: 手动重新启动请求</li> <li>• B#16#8D: 备用: 自动重新启动请求</li> </ul>
OB10x_PRIORITY	BYTE	优先级: 27
OB10x_OB_NUMBR	BYTE	OB编号(100、101或102)
OB10x_RESERVED_1	BYTE	保留
OB10x_RESERVED_2	BYTE	保留
OB10x_STOP	WORD	导致CPU停止的事件的编号
OB10x_STRT_INFO	DWORD	有关当前启动的辅助信息
OB10x_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

下表给出了变量OB100\_STR\_INFO和OB101\_STR\_INFO。

位号	含义	可能的二进制值	说明
31 - 24	启动信息	0000 xxxx	机架号0(仅限H CPU)
		0100 xxxx	机架号1(仅限H CPU)
		1000 xxxx	机架号2(仅限H CPU)
		0001 xxxx	多值计算(只对S7-400)
		0010 xxxx	分段机架中多个CPU的操作 (只对S7-400)
		xxxx xxx0	预期组态与实际组态之间无任何 差异(仅限S7-300)
		xxxx xxx1	预期组态与实际组态之间存在差 异(仅限S7-300)

位号	含义	可能的二进制值	说明
		xxxx xx0x	预期组态与实际组态之间无任何差异
		xxxx xx1x	预期组态与实际组态之间存在差异
		xxxx x0xx	非H CPU
		xxxx x1xx	H CPU
		xxxx 0xxx	最后一次上电时用于时间标志的时钟未由电池供电
		xxxx 1xxx	最后一次上电时用于时间标志的时钟由电池供电
23 - 16	刚完成启动	0000 0001	多值计算中的暖重启, 未根据参数分配来更改CPU中的设置 (仅限S7-400)
		0000 0011	利用模式选择器触发重新启动 (暖重启)
		0000 0100	经由MPI通过命令触发重新启动 (暖重启)
		0000 0101	多值计算中的冷重启, 未根据参数分配来更改CPU中的设置 (仅限S7-400)
		0000 0011	利用模式选择器触发冷重启
		0000 1000	经由MPI通过命令触发冷重启
		0000 1010	多值计算中的热重启, 未根据参数分配来更改CPU中的设置 (仅限S7-400)
		0000 1011	利用模式选择器触发热重启 (仅限S7-400)
		0000 1100	经由MPI通过命令触发热重启 (仅限S7-400)
		0001 0000	在由电池支持的上电后, 自动重新启动(暖重启)
		0001 0001	根据参数分配在由电池支持的上电后进行冷重启
		0001 0011	利用模式选择器触发重新启动 (暖重启); 最后一次上电由电池支持
		0001 0100	经由MPI通过命令触发重新启动 (暖重启); 最后一次上电由电池支持
		0010 0000	在经过由电池支持的上电后, 自动重新启动(暖重启) (由系统复位存储器)
		0010 0001	在经过由电池支持的上电后, 执行冷重启(由系统复位存储器)

位号	含义	可能的二进制值	说明
		0010 0011	利用模式选择器触发重新启动(暖重启); 最后一次上电未受电池支持
		0010 0100	经由MPI通过命令触发重新启动(暖重启); 最后一次上电未受电池支持
		1010 0000	根据参数分配在经过由电池支持的上电后自动热重启(仅限S7-400)
15 - 12	自动启动的许可	0000	自动启动非法, 请求存储器复位
		0001	自动启动非法, 有必要进行参数修改等
		0111	允许自动重新启动(暖重启)
		1111	允许自动重新启动(暖重启/热重启) (仅限S7-400)
11 - 8	手动启动的许可	0000	自动启动非法, 请求存储器复位
		0001	启动非法, 有必要进行参数修改等
		0111	允许重新启动(暖重启)
		1111	允许重新启动(暖重启/热重启) (仅限S7-400)
7 - 0	上一次上电时对自动启动的有效干预或设置	0000 0000	无启动
		0000 0001	多值计算中的暖重启, 未根据参数分配来更改CPU上的设置 (仅限S7-400)
		0000 0011	由模式选择器触发的重新启动(暖重启)
		0000 0100	经由MPI通过命令触发重新启动(暖重启)
		0000 0101	多值计算中的热重启, 未根据参数分配来更改CPU中的设置 (只对S7-400)
		0000 0111	利用模式选择器触发冷重启
		0000 1000	经由MPI通过命令触发冷重启
		0000 1010	多值计算中的热重启, 未根据参数分配来更改CPU中的设置 (只对S7-400)
		0000 1011	利用模式选择器触发热重启 (只对S7-400)

位号	含义	可能的二进制值	说明
		0000 1100	经由MPI通过命令触发热重启 (只对S7-400)
		0001 0000	在由电池支持的上电后，自动重新启动(暖重启)
		0001 0001	根据参数分配在由电池支持的上电后进行冷重启
		0001 0011	利用模式选择器触发重新启动(暖重启)；最后一次由电池支持上电
		0001 0100	经由MPI通过命令触发重新启动(暖重启)；最后一次由电池支持通电
		0010 0000	在经过由电池支持的上电后，自动重新启动(暖重启) (由系统复位存储器)
		0010 0001	根据参数分配在由电池支持的上电后进行冷重启
		0010 0011	利用模式选择器触发重新启动(暖重启)；最后一次上电非电池支持
		0010 0100	经由MPI通过命令触发重新启动(暖重启)；最后一次上电非电池支持
		1010 0000	根据参数分配在经过由电池支持的上电后自动热重启 (仅限S7-400)

## 1.26 编程出错组织块(OB121)

### 描述

只要发生同程序处理相关的错误所导致的事件，CPU的操作系统即调用OB121。  
例如，如果用户程序调用了尚未装载到CPU中的块，将会调用OB121。

### 了解编程出错OB的操作

执行OB121时所处优先级与中断块的优先级相同。

如果OB121未编程，则CPU将由RUN模式改变为STOP模式。

S7提供了下列SFC，用于在执行程序期间屏蔽和取消屏蔽OB121的启动事件：

- SFC36 (MSK\_FLT)：屏蔽特定的出错代码
- SFC37 (DMSK\_FLT)：取消屏蔽由SFC36屏蔽的出错代码
- SFC38 (READ\_ERR)：读取出错寄存器

### 编程出错OB的本地数据

下表介绍了编程出错OB的临时(TEMP)变量。变量名为OB121的缺省名称。

变量	类型	描述
OB121_EV_CLASS	BYTE	事件等级和标识符：B#16#25
OB121_SW_FLT	BYTE	出错代码： (可能值：B#16#21、B#16#22、B#16#23、B#16#24、 B#16#25、B#16#26、B#16#27、B#16#28、B#16#29、 B#16#30、B#16#31、B#16#32、B#16#33、B#16#34、 B#16#35、B#16#3A、B#16#3C、B#16#3D、B#16#3E、 B#16#3F)
OB121_PRIORITY	BYTE	优先级 = 出错OB的优先级
OB121_OB_NUMBR	BYTE	OB编号(121)
OB121_BLK_TYPE	BYTE	错误所在块的类型(使用S7-300时，未在此处输入有效值)：B#16#88：OB，B#16#8A：DB， B#16#8C：FC，B#16#8E：FB
OB121_RESERVED_1	BYTE	保留
OB121_FLT_REG	WORD	出错源(取决于出错代码)。 例如： <ul style="list-style-type: none"> <li>• 发生转换出错的寄存器</li> <li>• 不正确的地址(读/写出错)</li> <li>• 不正确的定时器/计数器/块编号</li> <li>• 不正确的存储区域</li> </ul>
OB121_BLK_NUM	WORD	具有导致出错的MC7命令的块编号(未在此处为S7-300输入有效的编号)
OB121_PRG_ADDR	WORD	导致出错的MC7命令的相对地址 (未在此处为S7-300输入有效值)
OB121_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME

取决于错误代码的变量具有如下：

出错代码:	含义
B#16#21: OB121_FLT_REG:	BCD转换出错 所涉及寄存器的ID(W#16#0000: 累加器1)
B#16#22: B#16#23: B#16#28: B#16#29:  OB121_RESERVED_1:	读取时区域长度出错 写入时区域长度出错 利用位地址不为0的指针对字节、字或双字进行读取访问。 利用位地址不为0的指针对字节、字或双字进行写访问。 字节地址不正确。可从OB121_RESERVED_1读取数据区和访问类型。 <ul style="list-style-type: none"> <li>• 位7至4访问类型。 <ul style="list-style-type: none"> <li>- 0: 位访问,</li> <li>- 1: 字节访问,</li> <li>- 2: 字访问,</li> <li>- 3: 双字访问</li> </ul> </li> <li>• 位3至0存储区域: <ul style="list-style-type: none"> <li>- 0: I/O区域</li> <li>- 1: 过程映像输入表</li> <li>- 2: 过程映像输出表</li> <li>- 3: 位存储器</li> <li>- 4: 全局DB</li> <li>- 5: 背景数据块</li> <li>- 6: 自己的本地数据</li> <li>- 7: 发话人的本地数据</li> </ul> </li> </ul>
B#16#24: B#16#25: OB121_FLT_REG:	读取时发生范围出错 写入时发生范围出错 在低字节中包含非法区域的ID(自己的本地数据区的B#16#86)
B#16#26: B#16#27: OB121_FLT_REG:	定时器编号出错 计数器编号出错 编号非法
B#16#30: B#16#31: B#16#32: B#16#33: OB121_FLT_REG:	对写保护全局DB的写访问 对写保护背景数据块的写访问 访问全局DB时DB编号出错 访问背景数据块时DB编号出错 DB编号非法



出错代码:	含义
B#16#34:	调用FC时FC编号出错
B#16#35:	调用FB时FB编号出错
B#16#3A:	访问了尚未装载的DB; DB编号位于允许的范围内
B#16#3C:	访问了尚未装载的FC; FC编号位于允许的范围内
B#16#3D:	访问了尚未装载的SFC; SFC编号位于允许的范围内
B#16#3E:	访问了尚未装载的FB; FB编号位于允许的范围内
B#16#3F:	访问了尚未装载的SFB; SFB编号位于允许的范围内
OB121_FLT_REG:	编号非法

## 1.27 I/O访问出错组织块(OB122)

### 描述

只要在访问模块上的数据时出错，CPU的操作系统即调用OB122。例如，如果在访问I/O模块上的数据时，CPU检测到读取错误，操作系统将调用OB122。

### 了解I/O访问出错OB的操作

执行OB122时所处优先级与中断OB的优先级相同。如果OB122未编程，则CPU将由RUN模式改变为STOP模式。

S7提供了下列SFC，用于在执行程序期间屏蔽和取消屏蔽OB122的启动事件：

- SFC36 (MSK\_FLT)：屏蔽特定的出错代码
- SFC37 (DMSK\_FLT)：取消屏蔽由SFC36屏蔽的出错代码
- SFC38 (READ\_ERR)：读取出错寄存器

### I/O访问出错OB的本地数据

下表说明了I/O访问出错OB的临时(TEMP)变量。变量名为OB122的缺省名称。

变量	类型	描述
OB122_EV_CLASS	BYTE	事件等级和标识符：B#16#29
OB122_SW_FLT	BYTE	出错代码： <b>B#16#42</b> <ul style="list-style-type: none"> <li>- 对于S7-300和CPU 417：I/O访问错误，读取对于所有其它S7-400 CPU：发生错误后在第一次读取访问过程中出错</li> </ul> <b>B#16#43</b> <ul style="list-style-type: none"> <li>- 对于S7-300和CPU 417：I/O访问错误，写对于所有其它S7-400 CPU：发生错误后在第一次写访问过程中出错</li> </ul> <b>B#16#44</b> <ul style="list-style-type: none"> <li>- (仅适用于S7-400，不包括CPU 417)发生出错后在第n(n &gt; 1)次读取访问过程中出错</li> </ul> <b>B#16#45</b> <ul style="list-style-type: none"> <li>- (仅适用于S7-400，不包括CPU 417)发生出错后在第n(n &gt; 1)次写访问过程中出错</li> </ul>
OB122_PRIORITY	BYTE	优先级： <ul style="list-style-type: none"> <li>• 出错OB的优先级</li> </ul>
OB122_OB_NUMBR	BYTE	OB编号(122)
OB122_BLK_TYPE	BYTE	出错块的类型(B#16#88：OB， B#16#8C：FC，B#16#8E：FB)(未在此处为S7-300输入有效编号)

变量	类型	描述
OB122_MEM_AREA	BYTE	存储器区域和访问类型： <ul style="list-style-type: none"><li>• 位7至4: 访问类型<ul style="list-style-type: none"><li>- 0: 位访问</li><li>- 1: 字节访问</li><li>- 2: 字访问</li><li>- 3: 双字访问</li></ul></li><li>• 位3至0: 内存区域<ul style="list-style-type: none"><li>- 0: I/O区域</li><li>- 1: 输入的过程映像</li><li>- 2: 输出的过程映像</li></ul></li></ul>
OB122_MEM_ADDR	WORD	出错的内存地址
OB122_BLK_NUM	WORD	具有导致出错的MC7命令的块编号(未在此处为S7-300输入有效的编号)
OB122_PRG_ADDR	WORD	导致出错的MC7命令的相对地址(未在此处为S7-300输入有效编号)
OB122_DATE_TIME	DATE_AND_TIME	调用OB时的DATE_AND_TIME



## 2 SFC的公共参数

### 2.1 通过输出参数RET\_VAL判断出错

#### 出错信息的类型

在用户程序中执行的系统功能(SFC)将指示CPU是否能够成功执行SFC的功能。您可获取有关已出现的任何出错的信息，这些信息将以如下两种方式显示：

- 在状态字的BR位中
- 在输出参数RET\_VAL(返回值)中

---

#### 注意

在判断某个特定SFC的输出参数前，您应始终执行以下步骤：

- 首先，判断状态字的BR位。
- 然后，检查输出参数RET\_VAL。

如果BR位指示出现了出错或者如果RET\_VAL包含一个常规出错代码，则您不得判断该SFC的输出参数！

---

#### 返回值中的出错信息

系统功能(SFC)通过在状态字的二进制结果位(BR)中输入值“0”来指示执行时出现了一个出错。某些系统功能会在称为返回值(RET\_VAL)输出的输出处提供附加出错代码。如果在输出参数RET\_VAL中输入一个常规出错(相关解释请参见下文)，则只能通过状态字BR位中的值“0”指示。

返回值为整型数据类型(INT)。返回值与值“0”之间的关系指示出在执行该功能时是否出错。

在CPU上执行SFC	BR	返回值	整数符号
出现出错	0	小于“0”	负 (符号位为“1”)
未出现出错	1	大于或等于“0”	正 (符号位为“0”)

## 对出错信息的响应

RET\_VAL中有如下两种不同类型的出错代码：

- 所有系统功能均可输出的常规出错代码以及
- 系统功能可输出的特定出错代码，该代码与其特定功能相关。

您可以编写程序，使其对系统功能执行期间出现的出错作出响应。这样，您便可防止由第一个出错而导致的更多出错的出现。

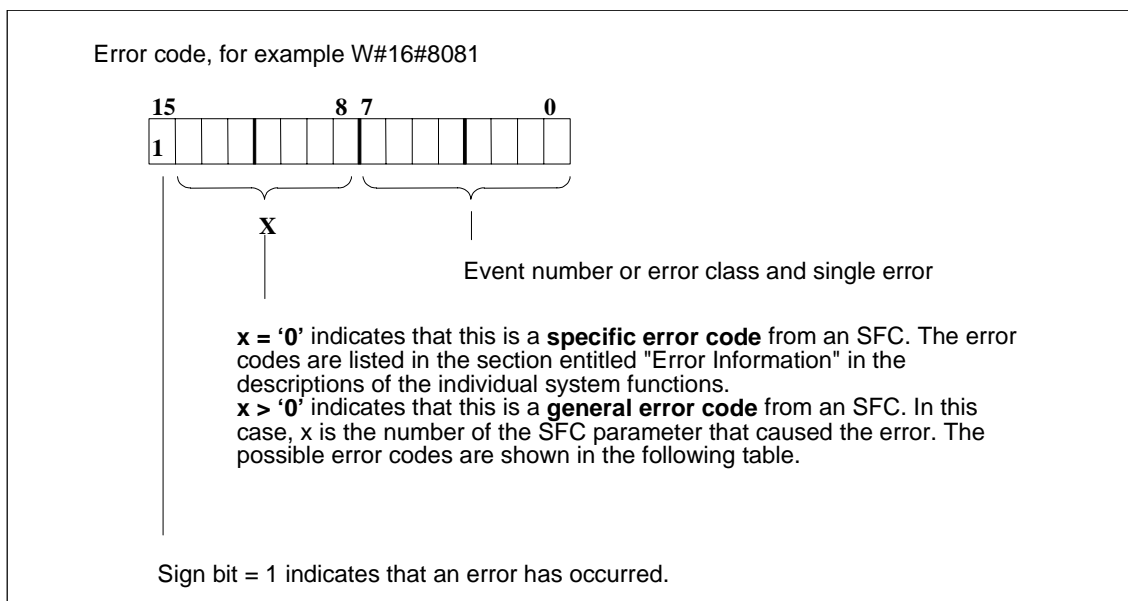
## 常规和特定出错信息

系统功能的返回值(RET\_VAL)将提供下列两种出错代码类型之一：

- 与可在任何系统功能中出现的出错相关的常规出错代码。
- 仅与特定系统功能相关的特定出错代码。

虽然输出参数RET\_VAL的数据类型为整型(INT)，但会根据十六进制值对系统功能的出错代码进行编组。如果您要检查某个返回值并将该值与本手册中列出的出错代码进行比较，则请用十六进制格式显示出错代码。

下图给出了十六进制格式的系统功能出错代码结构。

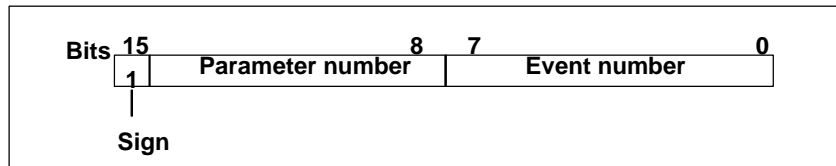


## 常规出错信息

常规出错代码指示可在任何系统功能中出现的出错。常规出错代码由下列两种数字组成：

- 介于1到111之间的参数编号，其中1指示SFC的第一个参数，2指示第二个参数，等等。
- 介于0到127之间的事件编号。该事件编号指示出现了同步出错。

下表列出了常规出错的代码以及每个出错的解释。



## 注意

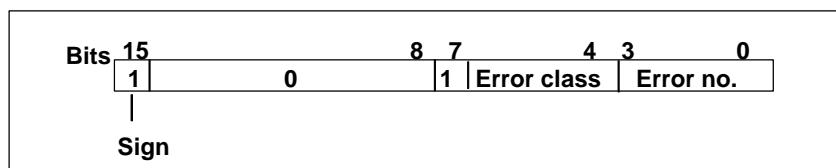
如果在RET\_VAL中输入了一个常规出错代码，则可能会出现下列情况：

- 与SFC相关的操作可能已开始或者已经完成。
- 可能在执行该操作时已经出现了特定SFC出错。但由于之后出现了常规出错而可能不再指示该特定出错。

## 特定出错信息

某些系统功能(SFC)有一个可提供特定出错代码的返回值。此出错代码指示在某一特定系统功能执行期间出现了一个与该功能相关的出错(参见下图)。特定出错代码由下列两种数字组成：

- 介于0到7之间的出错类别。
- 介于0到15之间的出错编号。



## 常规出错代码

下表解释了返回值的常规出错代码。该出错代码用十六进制格式显示。每个代码编号中的字母x仅仅是一个占位符，它代表引起该出错的系统功能参数的编号。

出错代码(W#16#...)	说明
8x7F	内部出错 此出错代码指示参数x处的内部出错。此出错不是由用户引起的，用户无法消除。
8x01	ANY参数中的非法语法ID
8x22 8x23	读参数时出现范围长度出错。 写参数时出现范围长度出错。 此出错代码指示参数x是全部还是部分位于地址范围之外，或者指示ANY参数的位范围长度不是8的倍数。
8x24 8x25	读参数时发生范围出错。 写参数时发生范围出错。 此出错代码指示参数x位于一个对于系统功能非法的范围之内。有关非法范围的信息，请参见各个功能的描述。
8x24 8x25	读参数时发生范围出错。 写参数时发生范围出错。 此出错代码指示参数x位于一个对于系统功能非法的范围之内。有关非法范围的信息，请参见各个功能的描述。
8x26	该参数包含一个过大的定时器编号。 此出错代码指示在参数x中指定的定时器不存在。
8x27	该参数包含一个过大的计数器编号(计数器编号出错)。 此出错代码指示在参数x中指定的计数器不存在。
8x28 8x29	读参数时发生地址对齐出错。写参数时发生地址对齐出错。 此出错代码指示参数x的参考是一个不等于0的位地址。
8x30 8x31	该参数位于一个只读的全局DB中。 该参数位于一个只读的背景数据块中。 此出错代码指示参数x位于一个只读数据块中。如果该数据块是用系统功能本身打开的，则系统功能将始终返回值W#16#8x30。
8x32 8x34 8x35	该参数包含一个过大的DB编号(DB编号出错)。 该参数包含一个过大的FC编号(FC编号出错)。 该参数包含一个过大的FB编号(FB编号出错)。 此出错代码指示参数x包含一个大于允许最大编号的块编号。
8x3A 8x3C 8x3E	该参数包含一个未装载的DB编号。 该参数包含一个未装载的FC编号。 该参数包含一个未装载的FB编号。
8x42 8x43	在系统尝试从外设输入区读参数时出现访问出错。 在系统尝试将参数写入外设输出区时出现访问出错。
8x44 8x45	在出现一个出错后，在第n(n > 1)个读访问中出现出错。 在出现一个出错后，在第n(n > 1)个写访问中出现出错。 此出错代码指示对所需参数的访问被拒绝。

## 2.2 异步SFC的REQ、RET\_VAL和BUSY参数的含义

### 异步SFC

异步运行的SFC是这样一种SFC：在完成其功能前，它们将被调用一次以上。下列SFC始终异步执行，或者在某些情况下才会异步执行：



- SFC7 “DP\_PRAL”
- SFC11 “DPSYC\_FR”
- SFC12 “D\_ACT\_DP”
- SFC13 “DPNRM\_DG”
- SFC51 “RDSYSST”
- SFC55 “WR\_PARM”
- SFC56 “WR\_DPARM”
- SFC57 “PARM\_MOD”
- SFC58 “WR\_REC”
- SFC59 “RD\_REC”
- SFC65 “X\_SEND”
- SFC67 “X\_GET”
- SFC68 “X\_PUT”
- SFC69 “X\_ABORT”
- SFC72 “I\_GET”
- SFC73 “I\_PUT”
- SFC74 “I\_ABORT”
- SFC82 “CREA\_DBL”
- SFC83 “READ\_DBL”
- SFC84 “WRIT\_DBL”
- SFC90 “H\_CTRL”
- SFC102 “RD\_DPARA”
- SFC103 “DP\_TOPOL”
- SFC114 “PN\_DP”

## 标识作业

如果您用以上所列的某个SFC触发一个硬件中断、将控制命令输出到DP从站、启动数据传送或者中止未组态连接，然后在当前作业完成之前再次调用同一SFC，则SFC的执行结果取决于第二次调用是否涉及相同的作业。

下表解释了哪些输入参数可为这些SFC指定该作业。如果这些参数与一个尚未完成的作业的参数相匹配，则SFC调用将被视为后续调用。

SFC	作业的标识符为...
7 "DP_PRAL"	IOID、LADDR
11 "DPSYC_FR"	LADDR、GROUP、MODE
12 "D_ACT_DP"	LADDR
13 "DPNRM_DG"	LADDR
51 "RDSYSST"	SSL_ID、INDEX
55 "WR_PARM"	IOID、LADDR、RECNUM
56 "WR_DPARM"	IOID、LADDR、RECNUM
57 "PARM_MOD"	IOID、LADDR
58 "WR_REC"	IOID、LADDR、RECNUM
59 "RD_REC"	IOID、LADDR、RECNUM
65 "X_SEND"	DEST_ID、REQ_ID
67 "X_GET"	DEST_ID、VAR_ADDR
68 "X_PUT"	DEST_ID、VAR_ADDR
69 "X_ABORT"	DEST_ID
72 "I_GET"	IOID、LADDR、VAR_ADDR
73 "I_PUT"	IOID、LADDR、VAR_ADDR
74 "I_ABORT"	IOID、LADDR
82 "CREA_DBL"	LOW_LIMIT、UP_LIMIT、COUNT、ATTRIB、SRCBLK
83 "READ_DBL"	SRCBLK、DSTBLK
84 "WRIT_DBL"	SRCBLK、DSTBLK
90 "H_CTRL"	MODE、SUBMODE
102 "RD_DPARA"	LADDR、RECNUM
103 "DP_TOPOL"	DP_ID
114 "PN_DP"	-

## 输入参数REQ

REQ(请求)输入参数只用于启动作业：

- 如果您为一个当前未激活的作业调用SFC，则该作业将用REQ = 1进行启动(情况1)。
- 如果某特定作业已启动但尚未完成，并且您再次调用SFC来执行相同的作业(例如，在一个循环中断OB中)，则SFC不会判断REQ(情况2)。

## 输出参数RET\_VAL和BUSY

作业执行状态由输出参数RET\_VAL和BUSY指示。

另请参见通过输出参数RET\_VAL判断出错中的注意。

- 在第1种情况中(第一次用REQ=1进行调用), 如果系统资源可用且输入参数正确, 则会将W#16#7001输入到RET\_VAL中。然后将BUSY置位。

如果所需要的系统资源当前正在使用中或者输入参数有出错, 则会将相应的出错代码输入到RET\_VAL中, 且BUSY具有值0。

- 在第2种情况中(在相同的作业已激活时进行调用), 会将W#16#7002输入到RET\_VAL中(这是一个说明该作业仍在处理中的警告), 并将BUSY置位。
- 下列内容适用于作业的最后一次调用:
  - 对于SFC13 “DPNRM\_DG,” SFC67 “X\_GET”和SFC72 “I\_GET”, 如果未出现任何出错, 则会将提供的数据的数目以正的字节数的形式输入到RET\_VAL中。然后, BUSY便会具有值0。  
如果出现一个出错, 则RET\_VAL将包含出错信息。这种情况下, 不得使用BUSY进行判断。
  - 对于SFC59  
“RD\_REC”, 如果未出现任何出错, 则会将以字节为单位的数据记录大小或者0值输入到RET\_VAL中(请参见用SFC59 “RD\_REC”读一个数据记录!)。这种情况下, BUSY具有值0。如果出现一个出错, 则会将出错代码输入到RET\_VAL中, 且BUSY具有值0。)
  - 对于所有其它SFC, 如果执行作业时未出现任何出错, 则会将0输入到RET\_VAL中, 且BUSY具有值0。如果出现一个出错, 则会将出错代码输入到RET\_VAL中, 且BUSY具有值0。

---

### 注意

如果第一次和最后一次调用一起执行, 则RET\_VAL和BUSY的执行结果与最后一次调用之描述相同。

---

## 概述

下表概括了上面所述的关系。特别是，其给出了在已调用SFC但未执行完作业的情况下输出参数的可能值。

### 注意

在每次调用之后，您必须评估您程序中相关的输出参数。

在作业执行期间，调用类型、REQ、RET、RET\_VAL和BUSY之间的关系。

调用编号	调用类型	REQ	RET_VAL	BUSY
1	第一次调用	1	W#16#7001	1
			出错代码:	0
2到(n - 1)	中间调用	不相关	W#16#7002	1
N	最后一次调用	不相关	如果未出现任何出错，则为W#16#0000 (例外: SFC59 "RD_REC"(如果目标区域大于传送的数据记录)及 SFC13 "DPNRM_DG"、SFC67 "X_GET" 和SFC72 "I_GET")	0
			如果出现了出错，则为出错代码	0

## 3 复制功能和块功能

### 3.1 使用SFC20 “BLKMOV”复制存储区域

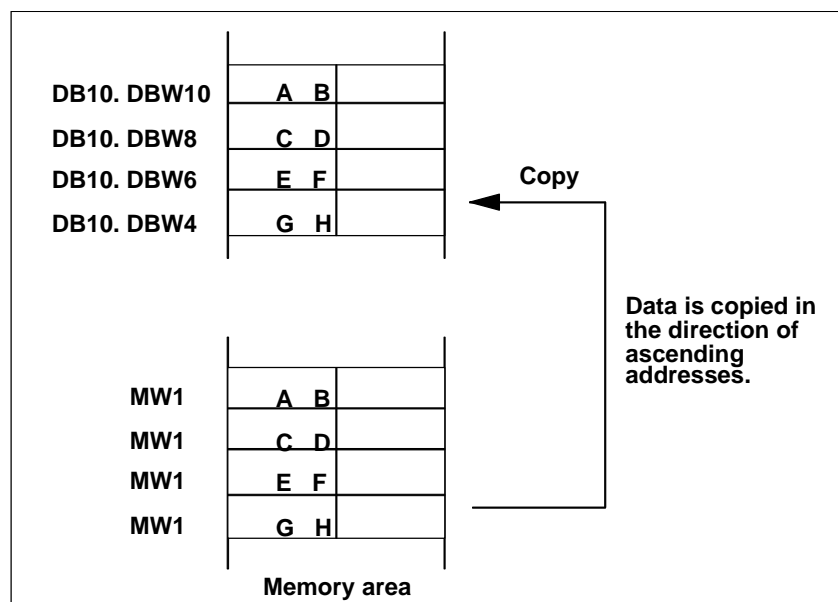
#### 描述

使用SFC20 “BLKMOV”(块移动)可将存储器区(= 源区域)的内容复制到另一个存储器区(= 目标区域)。

以下为允许的源区域:

- 部分数据块
- 存储位
- 输入的过程映像分区(部分过程映像)
- 输出的过程映像分区(部分过程映像)

源参数可以是与程序执行无关的装载存储器中数据块(用关键字UNLINKED编译的数据块)的一部分。



## 中断功能

只要源区域不是只存在于装载存储器中的数据块的一部分，就没有嵌套深度的限制。

但是，如果在从与程序执行无关的数据块复制时SFC20被中断，则SFC20不能再嵌套执行。

参数	声明	数据类型	存储区域	描述
SRCBLK	INPUT	ANY	I、Q、M、D、L	指定被复制的存储区(源区域)。不允许指定STRING类型的数组。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	在功能执行时如有故障发生，此参数值中包含故障代码。
DSTBLK	OUTPUT	ANY	I、Q、M、D、L	指定要将数据复制到存储区(目标区域)。不允许指定STRING类型的数组。

### 注意

源区域与目标区域一定不可以交叉。如果指定的目标区域大于源区域，此功能只将源区域中包含的数据复制到目标区域。

如果指定的目标区域小于源区域，则只复制目标区域中能容纳的数据。

如果ANY指针(源或目标)的类型为BOOL，则指定的长度必须能够被8整除；否则此SFC将不被执行。

源或目标参数(或二者都)的数据类型也可以为STRING。如果源为字符串，将只复制该字符串的当前字符。如果目标为字符串，复制字符的当前长度被写入。不能复制“STRING数组”。这意味着只有“STRING 1”是允许的。

如果已用STEP 7在数据块或本地数据区中定义了源和目标字符串，并要在SRCBLK和DSTBLK参数中使用这些字符串，SFC20也会将源字符串的“最大长度”和“实际长度”发送到目标字符串。为了防止发生这种情况，请自行创建在SRCBLK和DSTBLK参数中使用的ANY指针。请使用STRING数据类型完成此操作。

**特性：**如果用SFC20 BLKMOV将未链接的数据块复制并装载到RAM(例如，通过编程

设备)，SFC将被延时多达几毫秒。这会导致一个长的OB循环时间，可能会使循环监控报警。所以应避免在CPU通过SFC20复制块时装载这类块。

## 出错信息

出错代码 (W# 16#...)	说明
0000	无故障
8091	嵌套深度超出。
8092	SFC20 “BLKMOV”不能被执行因为进入的数据块是不能被执行的。需用SFC 83实现复制功能。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 3.2 使用SFC81 “UBLKMOV”不间断地复制变量

### 描述

使用SFC81 “UBLKMOV”(不间断的块移动)，可以将存储区(= 源区域)的内容原封不动地复制到另一个存储区(=目标区域)。此复制操作不能被其它操作系统任务打断。

使用SFC81 “UBLKMOV”，可以复制除下列之外的所有存储器区：

- 下列块类型：FB、SFB、FC、SFC、OB、SDB
- 计数器
- 定时器
- 外设I/O区存储区
- 与运行无关的块

可以复制最大512字节的数据。请注意不同的CPU此数值不同；可以参阅指令列表。

源参数也可以加入未链接数据块(用关键字UNLINKED编译的数据块)的装载存储器中！

### 中断能力，中断响应时间

复制不可以被中断。注意如果使用SFC81 “UBLKMOV”，可以增加CPU的中断响应时间。

参数	声明	类型	存储区域	描述
SRCBLK	INPUT	ANY	I、Q、M、D、L	指定要复制的存储区(源区域)。不允许指定STRING类型的数组。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如功能执行时发生故障，则此参数中包含一个故障代码。
DSTBLK	OUTPUT	ANY	I、Q、M、D、L	指定要将数据复制到的存储区(目标区域)。不允许指定STRING类型的数组。



---

**注意**

源区域与目标区域一定不可以交叉。如果指定的目标区域大于源区域，此功能只将源区域中包含的数据复制到目标区域。

如果指定的目标区域小于源区域，则只复制目标区域中能容纳的数据。

如果ANY指针(源或目标)的类型为**BOOL**，则指定的长度必须能够被**8**整除；否则此**SFC**不被执行。

如果ANY指针的类型为**STRING**，定义的长度必须为**1**。

---

**出错信息**

出错代码(W#16#...)	描述
0000	无故障
8091	源区域在未链接数据块中。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 3.3 使用SFC21 “FILL”初始化存储区

#### 描述

使用SFC21 “FILL”，可以初始化一个存储区(目标区域)，数据采用另一个存储区(源区域)的内容。此SFC拷贝数据直到充满整个存储区。

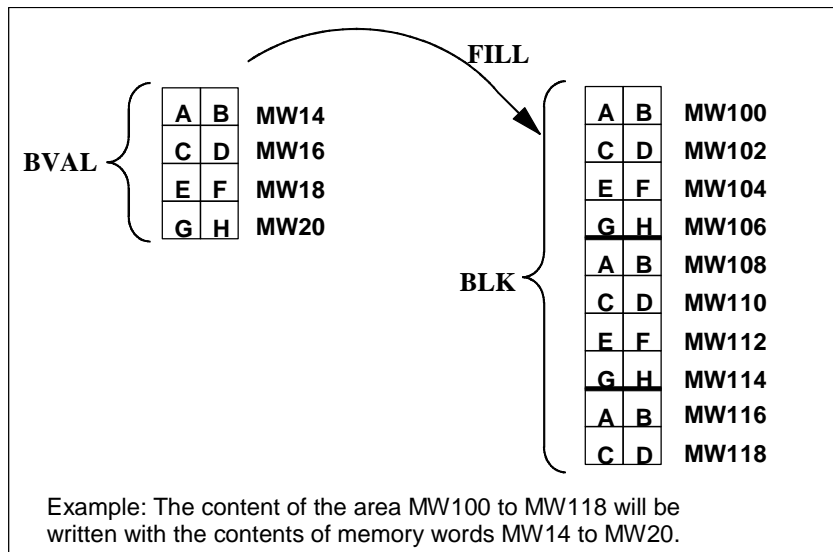
#### 注意

源与目标区域一定不可交叉。

如果目标区域的长度不为输入参数BVAL的整数倍，则目标区域永远不能写满到最后一字节。

如果目标区域长度小于源区域，只拷贝目标区域能容纳的数据。

如果ANY指针(源或目标)的类型为BOOL，则指定的长度必须能够被8整除；否则此SFC将不被执行。



## 例外

下列项目不能用作源区域：

- 计数器
- 定时器

使用SFC21不能将数据写到下列单元中：

- 下列块类型：FB、SFB、FC、SFC、SDB，
- 计数器，
- 定时器，
- 外设I/O存储区。

参数	声明	数据类型	存储区域	描述
BVAL	INPUT	ANY	I、Q、M、D、L	参数BVAL包含用于初始化目标区域的区域(源区域)的值或说明。不允许使用STRING数组。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	在功能执行时发生故障，则此参数中包含一个故障代码。
BLK	OUTPUT	ANY	I、Q、M、D、L	参数BLK包含被初始化的区域(目标区域)的说明。不允许使用STRING数组。

## 数据以结构形式输入

如果将一个结构作为输入参数传送，注意以下事项：

STEP 7总是把一个结构的长度定义为偶数字节数。因此，如果声明一个包含奇数字节数的结构，需要另加一字节的存储空间。

### 例如

声明一个结构如下：

```
TYP_5_BYTE_STRUCTURE : STRUCT
    BYTE_1_2 : WORD
    BYTE_3_4 : WORD
    BYTE_5 : BYTE
END_STRUCT
```

声明的结构“TYP\_5\_BYTE\_STRUCTURE”需要6字节存储空间。

## 出错信息

SFC21 “FILL”只提供常规(非特定的)出错信息。请参见使用输出参数RET\_VAL判断故障。

### 3.4 使用SFC22 “CREAT\_DB”创建数据块

#### 描述

使用SFC22 “CREAT\_DB”(创建数据块)，可以创建不包含初始化数据的数据块。同时，该数据块中会包含随机数据。此SFC可以创建一个可选择长度的数据块，数据块号在一个特定范围内。SFC将给数据块分配一个可能的最低编号。如果要创建特定编号的数据块，只需选择将上限和下限指定为同一值的范围。在用户程序中已存在的数据块编号不可以再使用。数据块的长度必须为偶数字节。

#### 中断能力

SFC22 “CREAT\_DB”可被更高优先级的组织块中断。如果SFC22 “CREAT\_DB”再次被更高优先级的组织块调用，则出现W#16#8091故障。

参数	声明	数据类型	存储区域	描述
LOW_LIMIT	INPUT	WORD	I、Q、M、D、L、 常数	下限值是分配给数据块号的最小值。
UP_LIMIT	INPUT	WORD	I、Q、M、D、L、 常数	上限值是分配给数据块号的最大值。
COUNT	INPUT	WORD	I、Q、M、D、L、 常数	计数值指定您想要为数据块保留的数据字节的数目。偶数字节 (最大65534)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果功能执行时发生故障，此参数值中包含故障代码。
DB_NUMBER	OUTPUT	WORD	I、Q、M、D、L	此数据块编号是所创建数据块的编号。如果出错，产生，(RET_VAL的位15被设置)值0被输入到DB_NUMBER中。

## 出错信息

出错代码 (W#16#...)	说明
0000	未出错。
8091	调用了嵌套的SFC22。
8092	当前不能执行“创建数据块”功能，可能因为 <ul style="list-style-type: none"> <li>• “压缩用户存储器”功能正在执行。</li> <li>• “压缩用户程序”功能当前处于激活状态。</li> <li>• H CPU正在执行链接或者升级功能。</li> <li>• 安装了WinAC软件的CPU探测到一个计算机操作系统故障。</li> </ul>
80A1	数据块编号出错： <ul style="list-style-type: none"> <li>• 编号为0。</li> <li>• 编号超出本CPU规定的的数据块编号值。</li> <li>• 参数下限 &gt; 上限。</li> </ul>
80A2	数据块长度出错： <ul style="list-style-type: none"> <li>• 长度为0。</li> <li>• 长度指定为奇数。</li> <li>• 长度超出CPU中的允许值。</li> </ul>
80B1	没有可用的数据块编号。
80B2	存储空间不够。
80B3	连续存储空间不够(纠正方法：压缩存储器！)
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 3.5 使用SFC23 “DEL\_DB”删除数据块

#### 描述

使用SFC23 “DEL\_DB”(删除数据块), 可删除存在于CPU的工作存储器以及装载存储器(如果存在)中的数据块。此数据块必须没有在当前或任何更低的优先级中打开, 换言之, 此数据一定不是位于两个数据块寄存器中的任意一个或B堆栈中。否则, CPU在调用SFC23时将启动OB°121。如果OB°121不存在, 则CPU停机。

#### 注意

不建议使用SFC23 “DEL\_DB”删除背景数据块。此操作总会导致程序出错。请避免使用SFC23删除背景数据块!

下表说明何时可以使用SFC23 “DEL\_DB”删除数据块。

如果...	那么...
数据块是由SFC22 “CREAT_DB”创建的,	SFC23 可以删除。
数据块由STEP 7传送到CPU, 且未使用关键字UNLINKED创建,	SFC23 可以删除。
数据块存储于闪存卡中,	SFC 23不能删除。

#### 中断功能

SFC23 “DEL\_DB”可被具有更高优先权的优先级中断。此时如再次调用此SFC, 则调用失败并且故障W#16#8091写入参数RET\_VAL中。

参数	声明	数据类型	存储区域	描述
DB_NUMBER	INPUT	WORD	I、Q、M、D、L、常数	要删除的数据块的编号
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息

## 出错信息

出错代码 (W#16#...)	说明
0000	未出错。
8091	SFC23被嵌套调用，且超出了CPU的最大嵌套深度。
8092	当前不能执行“删除数据块”功能，可能因为。 <ul style="list-style-type: none"> <li>• 当前激活了“压缩用户存储器”功能。</li> <li>• 正在上传将要删除的数据块。</li> <li>• H CPU正在执行链接或升级功能。</li> <li>• 在装有WinAC软件的计算机操作系统中CPU检测到一个故障</li> </ul>
80A1	输入参数DB_NUMBER出错：实际选择的参数 <ul style="list-style-type: none"> <li>• 为0。</li> <li>• 大于所用CPU允许的数据块编号最大值。</li> </ul>
80B1	指定编号的数据块编号在CPU中不存在。
80B2	指定编号的数据块是由关键字UNLINKED创建的。
80B3	数据块位于闪存卡上。
80B4	数据块不能被删除。可能的原因： <ul style="list-style-type: none"> <li>• 该数据块属于F程序。</li> <li>• 该数据块是用于S7通讯的块的背景数据块(仅限S7-400)。</li> <li>• 该数据块是技术数据块。</li> </ul>
80C1	因为暂时资源紧张，当前不能执行“删除数据块”功能。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 3.6 使用SFC24 “TEST\_DB”测试数据块

#### 描述

使用SFC24 “TEST\_DB”(测试数据块)，可获取位于CPU的工作存储器中的数据块的信息。此SFC查询所选数据块的数据字节数并检查其是否为只读数据块。

参数	声明	数据类型	存储区域	描述
DB_NUMBER	INPUT	WORD	I、Q、M、D、L、 常数	被测试的数据块编号
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
DB_LENGTH	OUTPUT	WORD	I、Q、M、D、L	所选数据块包含的数据字节数。
WRITE_PROT	OUTPUT	BOOL	I、Q、M、D、L	关于该数据块写保护标识符的信息 (1表示只读)。

#### 出错信息

故障代码(W#16#...)	说明
0000	未出错。
80A1	输入参数DB_NUMBER出错：实际选择的参数 <ul style="list-style-type: none"> <li>• 为0</li> <li>• 大于所用CPU允许的最大数据块编号。</li> </ul>
80B1	指定编号的数据块在CPU中不存在。
80B2	数据块是用关键字UNLINKED创建的。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。



### 3.7 使用SFC25 “COMPRESS”压缩用户存储器

#### 存储器中的空隙

当数据块被删除或重新装载几次后，在装载存储器和工作存储器中就会产生空隙。这些空隙会减少存储区的可用空间。

#### 描述

使用SFC25 “COMPRESS”可以压缩装载存储器和工作存储器的RAM区。压缩功能的启用与在RUN-P模式(模式选择)下外部启动CPU类似。

如果压缩功能由外部启动且一直激活，则调用SFC25将产生一个出错信息。

#### 注意

使用SFC25 “COMPRESS”无法移动长度大于1000字节的数据块。这意味着压缩之后工作存储器中仍有可能存在空隙。

参数	声明	数据类型	存储区域	描述
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
BUSY	OUTPUT	BOOL	I、Q、M、D、L	指示调用SFC25开始的压缩功能是否仍在执行。(1表示仍在执行。)
DONE	OUTPUT	BOOL	I、Q、M、D、L	指示调用SFC25开始的压缩功能是否已经完成。(1表示成功完成。)

#### 检查压缩功能

一旦调用SFC25 “COMPRESS”，压缩功能就开始执行了。但是，不能检查是否已成功压缩存储器空间。

如果要检查压缩功能，需遵循以下步骤：

循环调用SFC25。每次调用后先查看参数RET\_VAL。如果其值为0，则可查看参数BUSY和DONE。如果BUSY = 1且DONE = 0，则表示压缩功能仍在进行。当BUSY值变为0且DONE的值变为1时，表示压缩功能已成功完成。此后再调用SFC25，压缩功能会再次启动。

## 出错信息

故障代码(W#16#...)	说明
0000	未出错。压缩功能已通过调用SFC25启动。 在这种情况下，通过检查用户程序中的输出参数BUSY和DONE(见上文)仅能提供有用的信息。
8091	压缩功能由外部启动而且仍在执行。
8092	当前不能执行“压缩用户存储器”功能，因为 <ul style="list-style-type: none"><li>• “删除数据块”功能由STEP 7从外部启动而且仍在执行</li><li>• 测试和启动功能当前需要一个特定的块(例如，状态)</li><li>• “复制块”功能已由外部触发而且仍在执行。</li><li>• H-CPU正在执行链接或升级功能。</li></ul>
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 3.8 使用SFC44 “REPL\_VAL”传送一个替换值到累加器1

#### 描述

使用SFC44 “REPL\_VAL”(替换值)，可以传送一个值到累加器1中，这个值存在于可以引发故障的优先级中。

#### 限制条件：只能是位于同步故障组织块中

只能在同步故障组织块(OB121、OB122)中调用SFC44 “REPL\_VAL”。

#### 应用示例

如果一个输入模块损坏到不能读取任何值的程度，则每次扫描此模块时OB 122都会启动。使用SFC44 “REPL\_VAL”，可将OB 122中适当的值传送到中断优先级的累加器1中，这样程序就可以使用此替换值。选择此替换值的信息(例如，出错的块或受到影响的地址)位于OB 122的局部变量中。

参数	声明	数据类型	存储区域	描述
VAL	INPUT	DWORD	I、Q、M、D、L、常数	替换值
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果此功能执行时出错，返回值会包含故障代码。

#### 出错信息

故障代码(W#16#...)	说明
0000	未出错。替换值被输入。
8080	SFC44不是被同步故障组织块 (OB 121、OB 122)调用。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 3.9 使用SFC82 “CREA\_DBL”在装载存储器中生成数据块

#### 描述

使用SFC82 “CREA\_DBL”(在装载存储器中创建数据块)，可在装载存储器(微型存储卡)中创建新的数据块。SFC82使用指定范围中的一个编号生成一个缺省值大小的数据块。该数据块的编号为指定范围内的最小可能值。可以通过分配指定编号范围的上限和下限值来生成一个特定编号的数据块。不能把已在用户程序中存在的数据块编号分配给新生成的数据块。如果相同编号的数据块已存在于工作存储器和/或装载存储器中，或者该数据块为复制得到的，则SFC终止运行，并生成一条故障消息。

---

#### 注意

使用SFC24 “TEST\_DB”，可以确定相同编号的数据块是否已经存在。

---

参数SRCBLK (源块)指向的数据区的内容被写入到数据块中。此数据区必须是数据块或数据块中的一个区域。为了保持一致，当SFC82被执行时(即，只要BUSY参数值为TRUE时)不能改变此数据区。

具有READ\_ONLY属性的数据块只能由SFC82创建和初始化。

SFC82 不改变用户程序的校验和。

## 工作原理

SFC82 “CREA\_DBL”异步运行，也就是说，包含多次SFC调用。在REQ = 1时，调用SFC 82启动创建数据块功能。

作业状态通过输出参数RET\_VAL和BUSY显示。

另请参见异步运行SFC的REQ、RET\_VAL和BUSY的含义。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	REQ = 1: 请求生成数据块
LOW_LIMIT	INPUT	WORD	I、Q、M、D、L	SFC分配给数据块编号的范围的下限值
UP_LIMIT	INPUT	WORD	I、Q、M、D、L	SFC分配给数据块编号的范围的上限值
COUNT	INPUT	WORD	I、Q、M、D、L	计数值指定了数据块的字节数。此处，必须指定偶数字节。
ATTRIB	INPUT	BYTE	I、Q、M、D、L	数据块的属性:
				位0 = 1: UNLINKED: 该数据块只存在于装载存储器中。
				位1 = 1: READ_ONLY: 该数据块具有写保护。
				位2 = 1: NON_RETAIN: 该数据块没有掉电保持。
				位3至7: 备用
SRCBLK	INPUT	ANY	D	初始化当前数据块的数据区的指针
RET_VAL	OUTPUT	INT	I、Q、M、D、L	故障代码
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 工作还未结束。
DB_NUM	OUTPUT	WORD	I、Q、M、D、L	生成的数据块的编号

出错信息

出错代码 (W#16#...):	说明
0000	无故障
0081	目标区域大于源区域。 源区域中内部全部写到目标区域。剩余空间用0填满。
7000	REQ=0首次调用：无数据传输BUSY为0。
7001	REQ=1首次调用：无数据传输BUSY为1。
7002	中间调用(与REQ不相关)：数据传输已激活；BUSY为1。
8081	源区域大于目标区域。 <b>目标区域写满后。源区域中的其余字节将被忽略。</b>
8091	调用了嵌套的SFC82。
8092	“生成数据块”功能当前无法执行因为 <ul style="list-style-type: none"> <li>• “压缩应用存储器”操作正在进行</li> <li>• 已经达到CPU的最大块数量。</li> </ul>
8093	没有数据块或者一个不在工作存储器中的数据块指定给了参数SRCBLK。
8094	一个不被支持的属性被指定给参数ATTRIB
80A1	数据块编号故障： <ul style="list-style-type: none"> <li>• 编号为0</li> <li>• 下限 &gt; 上限</li> </ul>
80A2	数据块长度故障： <ul style="list-style-type: none"> <li>• 长度为0</li> <li>• 长度为奇数</li> <li>• 长度超出CPU允许的值</li> </ul>
80B1	没有可用的数据块编号
80B2	存储器不够用
80BB	装载存储器不够用
80C0	目标区域正在被另外一个SFC或通讯操作占用。
80C3	当前达到了同时激活的SFC82 的最大数目。
8xyy	一般故障的代码，例如： <ul style="list-style-type: none"> <li>• 源数据块不存在，或为复制版</li> <li>• 数据块所在的源区域不存在</li> </ul> 请参见使用输出参数RET_VAL判断故障。

### 3.10 使用SFC83 “READ\_DBL”从装载存储器的数据块中读取数据

#### 描述

使用SFC83 “READ\_DBL”(读取装载存储器中的数据块)可将装载存储器(微型存储卡)中的数据块或数据块的一个数据区复制到目标数据块的数据区中。目标数据块必须与运行有关；即不能用属性UNLINKED创建。在复制期间，装载存储器的内容不变。

为了保持一致，在SFC83执行时(即只要BUSY参数值为TRUE时)切勿改变目标区域。

下列限制条件适用于参数SRCBLK(源块)和DSTBLK(目标块)：

- 对于BOOLEAN类型的ANY指针，长度必须可被8整除。
- 对于BOOLEAN类型的ANY指针，长度必须可被1整除。

如果需要，可以使用SFC24 “TEST\_DB”确定源区域的长度。

---

#### 注意

SFC83异步执行。所以不适合从装载存储器中频繁(或反复)读取变量。

---

#### 注意

一旦开始，作业就必须被完成。如果达到了同时激活的SFC 83的最大数量，而此时具有更高优先权的优先级也调用SFC 83，将返回故障代码W#16#80C3。因此，并不马上重启高优先级的任务。

---

## 工作方法

SFC83 “READ\_DBL”是异步运行的，也就是说，处理过程包含此SFC的多次调用。  
REQ = 1时调用SFC83功能启动。

作业状态可通过输出参数RET\_VAL和BUSY显示。

参见异步运行SFC的参数REQ、RET\_VAL和BUSY的含义。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	REQ = 1: 读请求
SRCBLK	INPUT	ANY	D	装载存储器中读取的数据块的数据区的指针
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 读取过程尚未终止。
DSTBLK	OUTPUT	ANY	D	指向要写入的工作存储器的数据块的指针

## 出错信息

出错代码 (W#16#...):	描述
0000	无故障
8081	源范围大于目标范围。 将目标范围完全写入。源区域的其余字节将被忽略。
7000	REQ=0的第一次调用: 无激活的数据传送; BUSY值为0。
7001	第一次调用, REQ=1: 无激活的数据传送; BUSY值为1。
7002	中间调用(与REQ不相关): 数据传送已经激活; BUSY值为1。
0081	目标范围大于源范围。 <b>将源区域完全写入目标区域。目标区域的其余字节不变。</b>
80C0	该目标当前正由另一个SFC或通讯操作处理。
8093	针对参数DSTBLK未指示任何数据块或指示一个不在工作存储器中的数据块。
80B1	针对参数SRCBLK未指示任何数据块, 或者此处指示的数据块不是装载存储器中的对象(例如, 由SFC22生成的DB)。
80B4	不能读取具有F属性的DB。
80C0	目标DB当前正由另一个SFC或通讯功能处理。
80C3	当前达到了同时激活的SFC 83的最大数目。
8xyy	常规出错代码, 请参见使用输出参数RET_VAL判断出错



### 3.11 使用SFC84 “WRIT\_DBL”在装载存储器中写入数据块

#### 描述

使用SFC84 “WRIT\_DBL”(在装载存储器中写入数据块), 可将工作存储器中某一DB或DB区域中的内容传送到“微存储卡”中的DB或DB区域中。此源DB必须与处理相关; 即它一定不是用属性UNLINKED创建的。然而, 它也可以使用SFC22 “CREAT\_DB”生成。

为确保一致性, 一定不要在处理SFC84 的同时改变源区域(也就是说, 只要BUSY参数的值为TRUE)。

下列限制适用于参数SRCBLK(源块)和DSTBLK(目标块):

- 对于BOOLEAN类型的ANY指针, 长度必须可被8整除。
- 对于STRING类型的ANY指针, 长度必须等于1。

如果需要, 可以使用SFC24 “TEST\_DB”确定目标DB的长度。

如果您描述通过SFC生成的DB, SFC84不会改变用户程序的校验和。然而, 在写入装载DB时, 该DB中的第一个条目会改变用户程序的校验和。

#### 注意

SFC84不适于向装载存储器频繁(或循环)写入变量。这是因为“微存储卡”技术表明, 只能对“微存储卡”进行特定数量的写访问。有关详细信息, 请参见参考手册“SIMATIC S7-300可编程控制器CPU数据: CPU 31xC和CPU 31x”。

#### 工作原理

SFC84 “WRIT\_DBL”以异步方式工作, 也就是说, 处理涉及多个SFC调用。通过使用REQ = 1调用SFC84来启动作业。

作业状态通过输出参数RET\_VAL和BUSY显示。

异步操作SFC中REQ、RET\_VAL和BUSY的含义。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	REQ = 1: 写请求
SRCBLK	INPUT	ANY	D	指向要从中读取内容的工作存储器中DB的指针
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 写操作尚未结束。
DSTBLK	OUTPUT	ANY	D	指向要向其中写入的装载存储器中数据块的指针

## 出错信息

出错代码 (W#16#...):	描述
0000	无故障
0081	目标区域大于源区域。 将源区域完全写入目标区域。目标区域的其余字节不变。
7000	REQ=0首次调用：没有数据传输；BUSY值为0。
7001	REQ=1首次调用：没有数据传输；BUSY值为1。
7002	中间调用(与REQ无关)：数据传输功能激活；BUSY值为1。
8081	源区域大于目标区域。 <b>目标区域充满。源区域的其余字节将被忽略。</b>
8092	不正确的工作模式：当SFC 84处于激活状态时，CPU进入STOP模式。在下次切换到RUN的位置会提供此故障代码。再次调用SFC 84。
8093	针对参数SRCBLK未指示任何数据块或指示一个不在工作存储器中的数据块。
80B1	没有任何数据块或者不是装载存储器中的对象的数据块分配给参数DSTBLK (例如，由SFC22生成的数据块)。
80B4	不得更改具有F属性的数据块
80C0	目标区域当前正在被另外一个SFC或通讯操作占用。实例：您从CPU向编程设备(PG)上传数据块。您要使用SFC 84更改此数据块的内容。
80C3	当前达到了同时激活的SFC 84的最大数目。
8xyy	常规故障代码，请参见使用输出参数RET_VAL判断故障。

## 3.12 使用SFC85 “CREA\_DB”创建数据块

### 描述

使用SFC85 “CREA\_DB”(创建数据块)，可在用户程序中创建不带任何缺省值的数据块。相反，此数据块包含随机数据。SFC将以指定区域中的编号和缺省大小创建一个数据块。SFC给该数据块分配一个可能的最低编号。要创建具有特定编号的数据块，请为缺省区域的上限和下限输入同样的编号。用户程序中已经存在的数据块编号不可以再使用。数据块的长度必须是偶数字节。

根据为ATTRIB参数所做的选择，创建的数据块具有属性RETAIN或NON\_RETAIN:

- **RETAIN**表示数据块创建于工作存储器的掉电保持区域。也就是说数据块的当前值在每次断电/通电转换以及每次重新启动(热重启)之后会被保留。
- **NON\_RETAIN**表示数据块创建于工作存储器的非掉电保持区域。也就是说数据块的当前值在每次断电/通电转换以及每次重新启动(热重启)之后不确定。

如果不区分掉电保持与非掉电保持工作存储器，则ATTRIB参数会被忽略。也就是说数据块的值在每次断电/通电转换以及每次重新启动(热重启)之后会被保留。

## 中断功能

## SFC85

“CREA\_DB”可由更高优先级的组织块中断。如果SFC85 “CREA\_DB”再次被更高优先级的组织块调用，则该调用将被拒绝，并出现W#16#8091故障。

参数	声明	数据类型	存储区域	描述
LOW_LIMIT	INPUT	WORD	I、Q、M、D、L、 常数	下限值是可分配给数据块号的最小编号。
UP_LIMIT	INPUT	WORD	I、Q、M、D、L、 常数	上限值是可分配给数据块号的最大编号。
COUNT	INPUT	WORD	I、Q、M、D、L、 常数	计数值为数据块的数据字节数。必须指定偶数字节(最大65534)。
ATTRIB	INPUT	BYTE	I、Q、M、D、L、 常数	数据块属性： <ul style="list-style-type: none"> <li>• B#16#00: RETAIN</li> <li>• B#16#04: NON_RETAIN</li> </ul>
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果功能执行时出错，则返回值包含故障代码。
DB_NUMBER	OUTPUT	WORD	I、Q、M、D、L	创建的数据块的编号。在出错的情况下，(RET_VAL的第15位置1)DB_NUMBER置0。

## 出错信息

出错代码 (W#16#...)	描述
0000	无故障
8091	调用了嵌套的SFC85。
8092	“创建数据块”功能当前不能被执行，因为 <ul style="list-style-type: none"> <li>• “压缩用户存储区”功能正在执行</li> <li>• 安装了WinAC软件的CPU探测到一个计算机操作系统故障。</li> </ul>
8094	ATTRIB中的值无效
80A1	数据块编号出错： <ul style="list-style-type: none"> <li>• 编号为0</li> <li>• 编号超出本CPU规定的数据块的编号值</li> <li>• 下限 &gt; 上限</li> </ul>
80A2	数据块长度出错： <ul style="list-style-type: none"> <li>• 长度为0</li> <li>• 指定了一个奇数字节长度</li> <li>• 长度超出CPU中的允许值</li> </ul>
80B1	没有可用的数据块编号。
80B2	存储空间不够。
80B3	连续存储空间不够(压缩存储器！)
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。



## 4 用于控制程序执行的SFC

### 4.1 使用SFC43 “RE\_TRIGR”重新触发循环时间监视

#### 描述

使用SFC43 “RE\_TRIGR”(再触发监视狗)，再触发循环时间监控。

#### 参数

SFC43 “RE\_TRIGR”没有参数。

#### 出错信息

SFC43 “RE\_TRIGR”没有提供任何出错信息。

### 4.2 使用SFC46 “STP”将CPU切换为STOP

#### 描述

使用SFC46 “STP”(停止)，可将CPU切换为STOP。

#### 参数

SFC46 “STP”没有任何参数。

#### 出错信息

SFC46 “STP”没有提供任何出错信息。

## 4.3 使用SFC47 “WAIT”延时用户程序执行

### 描述

使用SFC47 “WAIT”，可在用户程序中设定延时或等待时间。最长可设定32767 s。最短时间依CPU型号而定，并与SFC47的执行时间相同。

### 中断能力

SFC47 “WAIT”可被优先级更高的组织块OB中断。

---

### 注意

**(仅限S7-300，且不包括CPU 318)**

使用SFC47所编写的等待时间是最短时间。由于嵌套优先级的执行时间以及系统负荷，该时间会有所延长。

---

参数	声明	数据类型	存储区域	描述
WT	INPUT	INT	I、Q、M、D、L、常数	参数WT包含以 s为单位的等待时间。

### 出错信息

SFC47 “WAIT”没有提供任何出错信息。

---



## 4.4 使用SFC35 “MP\_ALM”触发多处理器中断

### 描述

调用SFC35 “MP\_ALM”将触发多处理器中断功能。这时同步启动所有CPU的OB 60程序块。某机架上的单一处理器操作模式下，它的OB60只会被SFC35启动。

可以使用JOB输入参数指明多处理器中断的原因。此工作标识符同时传送到所有的CPU，再由多处理器中断程序(OB 60)识别(请参见“STEP 7编程”在线文档)。

可在用户程序中任何地方调用SFC35 “MP\_ALM”。但是只能在RUN状态下才有效，如果在STARTUP状态调用，多处理器中断功能无效。在这种情况下函数值将提示您。

参数	声明	数据类型	存储区域	描述
JOB	INPUT	BYTE	I、Q、M、D、L、常数	作业标识符：取值范围：1到15
RET_VAL	OUTPUT	INT	I、Q、M、D、L	功能执行时出错，返回值将包含故障代码。

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	输入参数JOB中的值非法。
80A0	OB60执行的时候本CPU或下一个CPU中的下一个中断没有完成。
80A1	状态出错(STARTUP状态代替了RUN)状态。
8xyy	常规故障代码，请参见使用输出参数RET_VAL判断故障

## 4.5 使用SFC104 “CiR”控制CiR

### 描述

RUN状态下用SFC104 “CiR”执行再组态功能：

- 可以完全禁用CiR。此时，会始终拒绝将修改的组态从PG下载到CPU。在使用SFC104 “CiR”进行复位之前，会维持锁定状态。
- 可以设定CiR同步到CiR禁止的时间上限值。在这种情况下，仅当CPU用于评估已修改组态所需的时间小于此指定时间时，修改后的组态才能从PG下载到CPU。
- 可以确定是否使能CiR。如果使能或者条件使能，参数A\_FT显示CiR同步时间值的实际上限值。

### 注意

在CiR同步时输出被禁止，输入值无效。

### 参数

参数	声明	数据类型	存储区域	描述
MODE	INPUT	BYTE	I、Q、M、D、L、 常数	作业ID号 允许值： <ul style="list-style-type: none"> <li>• 0：信息功能</li> <li>• 1：使能CiR功能(在此，CiR同步时间上限值已经预置)</li> <li>• 2：完全禁用CiR</li> <li>• 3：有条件禁用CiR。参数FRZ_TIME射定CiR同步时间上限值。</li> </ul>
FRZ_TIME	INPUT	TIME	I、Q、M、D、L、 常数	“冻结时间” CiR同步时间上限值(单位为ms) 允许的取值范围：200...2500ms(缺省值： 1000ms) 注意：只有MODE=3时FRZ_TIME 才有效。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果运行时出错，则返回值将包含故障 代码。 当MODE=0时，RET_VAL显示CiR是否 使能。
A_FT	OUTPUT	TIME	I、Q、M、D、L	当前CiR同步时间的上限值

## 出错信息

出错代码 (W#16#...)	说明
0000	作业执行过程中未出错。(此故障代码仅当MODE=1或MODE=2或MODE=3时才有效。)
0001	CiR使能。(此故障代码仅当MODE=0时才有效。)
0002	CiR已完全被禁用。(此故障代码仅当MODE=0时才有效。)
0003	CiR被有条件禁用。(此故障代码仅当MODE=0时才有效。)
8001	CPU没有准备CiR操作。您正在使用H系统(单一操作)中的H CPU，或者正在多处理器模式下使用标准CPU。
8002	MODE中的值非法
8003	FRZ_TIME中的值非法
8xyy	常规故障代码，请参见使用输出参数RET_VAL判断故障

## SFC104的应用示例

例如，SFC104 “CiR”可用于确保没有初始化CiR操作，在此过程中要求CPU处于最高性能或适当的状态。

为此，请在增加处理过程之前执行下列CPU程序段：

- 通过MODE = 2调用SFC104(完全禁用CiR操作)

在增加处理过程之后，执行下列CPU程序段：

- 通过MODE = 1调用SFC104(重新启用CiR操作)或者通过MODE = 3调用SFC 104(有条件禁用CiR)



## 5 用于处理系统时钟的SFC

### 5.1 使用SFC0 “SET\_CLK”设定TOD

#### 描述

使用SFC0 “SET\_CLK” (设定系统时钟)，可以设定CPU时钟的时间和日期。调用SFC0会启动时钟。时钟将从设定时间和设定日期开始运行。

如果这个时钟是主时钟，则在调用SFC 0的同时，CPU启动同步时间的功能。可使用STEP 7设定同步周期。

参数	声明	数据类型	存储区域	描述
PDT	INPUT	DT	D、L	在输入PDT时，输入要设定的日期和时间。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	功能执行时如有故障，则返回值会包含故障代码。

#### 日期和时间

以数据类型DT输入日期和时间。例如：1995年1月15日，上午10点30分30秒，输入为：DT#1995-01-15-10: 30: 30。输入时间值精确到秒级。由SFC 0 “SET\_CLK”从日期推算周的天数。

可以使用FC 3 “D\_TOD\_DT”创建数据类型DT(请参见时间功能：FC 1、FC 3、FC 6、FC 7、FC 8、FC 33、FC 34、FC 35、FC 40)。

#### 出错信息

出错代码(W#16#...)	说明
0000	无故障
8080	日期出错
8081	时间出错
8xyy	常规故障代码，请参见使用输出参数RET_VAL判断故障

## 5.2 使用SFC1 “READ\_CLK”读取时间

### 描述

使用SFC1 “READ\_CLK” (读取系统时钟), 可以读取当前CPU系统时钟的日期和时间。

参数	声明	数据类型	存储区域	描述
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能执行期间出错, 返回值会包含故障代码。
CDT	OUTPUT	DT	D、L	CDT输出当前日期和时间值。

### 出错信息

请参见使用输出参数RET\_VAL判断故障

### 5.3 使用SFC48 “SNC\_RTCB”同步子时钟

#### 定义：子时钟的同步

子时钟的同步是指将日期和时间从一个总线段的主时钟(例如，S7-400 K总线、MPI或S7背板总线)传送到该总线段的所有子时钟。

#### 描述

使用SFC48 “SNC\_RTCB” (实时时钟同步)，可以同步总线程序段上的所有子时钟，并且与组态同步时间间隔无关。只有调用SFC48的CPU设置为该总线段上的主时钟时，同步功能才有效。使用STEP 7设定相关参数。使用STEP 7分配相关的参数。

参数	声明	数据类型	存储区域	描述
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能执行期间出错，返回值会包含故障代码。

#### 出错信息

出错代码(W#16#...)	说明
0000	在同步期间未产生故障。
0001	对任何总线而言，现时钟都没有设成主时钟。
8xyy	常规故障代码，请参见使用输出参数RET_VAL判断故障

## 5.4 使用SFC100 “SET\_CLKS”设定时间日期和TOD状态

### 描述

使用SFC100 “SET\_CLKS”设定CPU的TOD和TOD状态。

### 当心

只有使用SFC100，CPU的时间日期才能同步。否则，每次同步，主站的TOD状态都被调用。这就使得每个SFC的设定值被刷新。

通过MODE参数，可以指定只更改TOD、只更改TOD状态还是同时更改二者。下表对此进行了解释：

MODE (B#16#...)	含义
01	设定TOD SFC的调用相当于SFC 0 “SET_CLK”的调用。 输入参数CORR、SUMMER和ANN_1无效。
02	设定TOD状态 输入参数PDT无效。保留的输入参数形成下列TOD状态： <ul style="list-style-type: none"> <li>• 带符号的修正值</li> <li>• 时间制式</li> <li>• 夏令时/冬令时标志</li> </ul> CPU决定TOD的分辨率。FALSE显示TOD状态位同步失败。 TOD保持不变。
03	设定TOD与TOD状态

### 注意

通过SFC51 “RDSYSST”读取SSL ID W#16#0132标志W#16#0008，可以确定CPU的当前TOD状态。



参数	声明	数据类型	存储区域	描述
MODE	INPUT	BYTE	I、Q、M、D、L、 常数	操作模式 允许值:
				B#16#01: 设定TOD
				B#16#02: 设定TOD状态
				B#16#03: 设定TOD与TOD状态
PDT	INPUT	DT	D、L	TOD预置
CORR	INPUT	INT	I、Q、M、D、L、 常数	修正值(0.5h模式) 允许值范围: -24到+26
SUMMER	INPUT	BOOL	I、Q、M、D、L、 常数	夏令时/冬令时标识符: <ul style="list-style-type: none"> <li>• 0 = 冬令时</li> <li>• 1 = 夏令时</li> </ul>
ANN_1	INPUT	BOOL	I、Q、M、D、L、 常数	时间制式 1: 从下一个小时起夏令时转换成冬令时 (或反之)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	故障代码:

## 出错信息

出错代码 (W#16#...):	说明
0000	无故障
8080	MODE超出允许值范围
8081	CORR超出允许值范围 (仅限MODE = B#16#02或MODE = B#16#03有效)
8082	PDT超出允许值范围: 非法日期或TOD
8xyy	常规故障代码, 请参见使用输出参数RET_VAL判断故障



## 6 用于处理运行系统计时器的SFC

### 6.1 运行时间定时器

#### 简介

CPU具有若干运行时间定时器(请参照有关CPU技术资料)。

- SFC 2、3和4用于设置、启动、停止或读取CPU 16位运行时间定时器。
- SFC101 “RTM”用于设置、启动、停止或读取CPU 32位运行时间定时器。

---

#### 注意

SFC 2、3和4也可用于32位运行时间定时器。但是，在这种情况下这个运行时间定时器将以16位模式工作(取值范围：0到32767小时)。

参见：具有SSL ID W#16#0132索引W#16#000B的部分列表摘录的数据记录。

---

#### 应用

运行时间定时器有各种应用方式：

- 测试CPU的运行时间
- 测试控制器或相关设备的运行时间。

#### 运行时间定时器的特征

启动时，运行时间定时器开始记录最后一个数值。如果要求在不同的初始值启动，这个值必须自己定义(SFC2或SFC101，MODE=4)。如果CPU进入STOP状态，或者您停止运行时间定时器，则CPU将记录运行时间定时器的当前值。在执行CPU热重启或冷启动后，必须重新启动运行时间定时器(SFC3或SFC101，MODE=1)。

#### 取值范围：

- 带有16位运行时间定时器的CPU：0到32767小时
- 带有32位运行时间定时器的CPU：0到 $(2E31) - 1$  小时 = 2.147.483.647小时

## 6.2 使用SFC101 “RTM”控制运行时间定时器

### 描述

调用SFC101 “RTM” (运行时间定时器)可设置、启动、停止和读取CPU的32位运行时间定时器。

调用SFC51 “RDSYSST”，SZL\_ID=W#16#0132，INDEX=W#16#000B(运行时间定时器0 ... 7)或INDEX=W#16#000C(运行时间定时器8 ... 15)读取CPU 32位运行时间定时器的值，(参见具有SSL ID W#16#0132索引W#16#000B的部分列表摘录的数据记录。

参数	声明	数据类型	存储区域	描述
NR	INPUT	BYTE	I、Q、M、D、L、 常数	运行时间定时器的编号(允许值: 0 ... 15)
MODE	INPUT	BYTE	I、Q、M、D、L、 常数	作业标识符: <ul style="list-style-type: none"> <li>• 0: 读取(状态写入CQ, 当前值写入CV)。运行时间定时器达到(2E31) -1 小时, 它会停止运行并显示这个最高值, 并输出“溢出”出错消息。</li> <li>• 1: 启动(从上一计数值开始)</li> <li>• 2: 停止</li> <li>• 4: 设置(在PV中的设定值)</li> <li>• 5: 设置(在PV中的设定值)并启动</li> <li>• 6: 设置(在PV中的设定值)并停止</li> </ul>
PV	INPUT	DINT	I、Q、M、D、L、 常数	运行时间定时器的新数值
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果该功能运行时出错, 则返回值将包含故障代码。
CQ	OUTPUT	BOOL	I、Q、M、D、L	运行时间定时器状态(1: 运行)
CV	OUTPUT	DINT	I、Q、M、D、L	运行时间定时器当前值

### 带有16位运行时间定时器的程序兼容性

SFC 2 “SET\_RTM”、SFC 3 “CTRL\_RTM”和SFC 4 “READ\_RTM”可用于操作32位运行时间定时器。但此时, 32位运行时间定时器只能当作16位使用(取值范围: 0到32767小时)。

具有SSL ID W#16#0132和索引W#16#0008的部分列表摘录以16位模式显示32位运行时间定时器0到7。也就是说, 您可以继续使用为具有16位运行时间定时器的CPU开发的程序, 这些程序使用具有SSL ID W#16#0132和索引W#16#0008的部分列表摘录。

## 出错信息

出错代码 (W#16#...)	说明
0000	无故障
8080	运行时间定时器编号出错
8081	负值被传送到参数PV。
8082	运行时间定时器溢出。
8091	输入参数MODE的值非法。
8xyy	常规故障代码，请参见使用输出参数RET_VAL判断故障

## 6.3 使用SFC2 “SET\_RTM”设置运行时间定时器

### 描述

使用SFC2 “SET\_RTM” (设置运行时间定时器)，可将CPU的运行时间定时器设置为所选值。可以设置的运行时间定时器数量取决于所使用的CPU。

参数	声明	数据类型	存储区域	描述
NR	INPUT	BYTE	I、Q、M、D、L、 常数	输入NR包含您要设置的运行时间定时器的编号(允许值：0到7)。
PV	INPUT	INT	I、Q、M、D、L、 常数	参数PV用于运行时间定时器设置(缺省值)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在执行功能的同时出错，则返回值包含故障代码。

### 出错信息

出错代码(W#16#...)	说明
0000	无故障
8080	运行时间定时器的编号出错
8081	负值被传送到参数PV。
8xyy	常规故障代码，请参见使用输出参数RET_VAL判断故障

## 6.4 使用SFC3 “CTRL\_RTM”启动和停止运行时间定时器

### 描述

SFC3 “CTRL\_RTM” (控制运行时间定时器)用于启动或停止CPU的运行时间定时器。

参数	声明	数据类型	存储区域	描述
NR	INPUT	BYTE	I、Q、M、D、L、 常数	输入NR包含您要启动或停止的运行时间定时器的编号(允许值：0到7)。
S	INPUT	BOOL	I、Q、M、D、L、 常数	输入S可启动或停止运行时间定时器。当您想要停止计数器时，请将信号状态设置为“0”。当您想要启动计数器时，请将信号状态设置为“1”。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在执行功能时出错，则返回值将包含故障代码。

### 出错信息

出错代码(W#16#...)	说明
0000	无故障
8080	运行时间定时器的编号故障
8xyy	常规故障代码，请参见使用输出参数RET_VAL判断故障

## 6.5 使用SFC4 “READ\_RTM”读取运行时间定时器

### 描述

使用SFC4 “READ\_RTM” (读取运行时间定时器), 可以读取运行时间定时器。SFC4以输出数据的形式提供当前的运行时间, 以及计数器的状态, 例如, “停止”或“计数”。

如果运行时间定时器运行超过32767小时, 它将停在计数值32767并输出“溢出”的故障消息。

参数	声明	数据类型	存储区域	描述
NR	INPUT	BYTE	I、Q、M、D、L、常数	输入NR包含您要读取的运行时间定时器的编号(允许值: 0到7)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能运行时出错, 则返回值包含故障代码。
CQ	OUTPUT	BOOL	I、Q、M、D、L	CQ显示运行时间定时器是否运行。信号状态“0”表示运行时间定时器已被停止。信号状态“1”表示运行时间定时器正在运行。
CV	OUTPUT	INT	I、Q、M、D、L	CV显示运行时间定时器的当前值。

### 出错信息

出错代码(W#16#...)	说明
0000	无故障
8080	运行时间定时器的编号出错
8081	运行时间定时器溢出
8xyy	常规故障代码, 请参见使用输出参数RET_VAL判断故障



## 6.6 使用SFC64 “TIME\_TCK”读取系统时间

### 描述

使用SFC64 “TIME\_TCK” (时间刻度报时信号)，可以读取CPU的系统时间。系统时间是一个循环时间计数器，技术范围周期性从0统计到最大值~2147483647毫秒的“时间计数器”。如果出现上溢溢出时，系统时间将从0开始重新计数。系统时间的分辨率和精度对于S7-400和CPU 318系统时间的分辨率和精度为1毫秒，而对于所有其它S7-300 CPU则为10毫秒。只有CPU的操作模式才影响系统时间只受CPU工作模式的影响。

### 应用

例如，比较SFC64两次执行的结果，测试执行周期。

### 系统时间和模式

模式	系统时间...
启动	...刷新
RUN	
STOP	...停止并且保持当前值
热重启 (不适用于S7-300和S7-400 H)	...延续进入停止前保存的值
暖重启	...删除并从“0”重新开始
冷重启	

参数	声明	数据类型	存储区域	描述
RET_VAL	OUTPUT	TIME	I、Q、M、D、L	RET_VAL参数读取的系统时间取值范围：0到 $2^{31}-1$ 毫秒。

### 出错信息

SFC64 “TIME\_TCK”不提供任何出错信息。



# 7 用于传送数据记录的SFC

## 7.1 读写一条数据记录

### 原理

某些模块具有只写的系统数据区，可由程序向其传送数据记录。此区域包含编号从0到最大为240的数据记录。并不是每个模块都包含所有数据记录(参见下表)。

另外，一些模块具有只读的系统数据区，可由程序从中读一个数据记录。此区域包含编号从0到最大为240的数据记录。并不是每个模块都包含所有数据记录(参见下表)。

---

### 注意

有的模块同时具有这两种系统数据区。在物理上这些区域是分开的，在逻辑结构上它们是相同的。

---

### 只写的系统数据区

下表显示只写的系统数据区的结构。此表还显示数据记录的允许长度以及利用哪些SFC写这些数据记录。

数据记录号	内容	长度	限制	可用于写操作的SFC
0	参数	S7-300: 2到14字节	只能由 S7-400写	56 "WR_DPARM" 57 "PARM_MOD"
1	参数	S7-300: 2到14字节 数据记录0和1 的总和正好是16字节。	-	55 "WR_PARM" 56 "WR_DPARM" 57 "PARM_MOD"
2 - 127	用户数据	每个 = 240字节	-	55 "WR_PARM" 56 "WR_DPARM" 57 "PARM_MOD" 58 "WR_REC"
128 - 240	参数	每个 = 240字节	-	55 "WR_PARM" 56 "WR_DPARM" 57 "PARM_MOD" 58 "WR_REC"

## 只读的系统数据区

下表显示只读的系统数据区的结构。此表还显示数据记录的允许长度以及利用哪些SFC读取这些数据记录。

数据记录号	内容	长度	可用于读操作的SFC
0	模块专用诊断数据 (标准设定用于整个系统)	4字节	51 "RDSYSST" (SSL_ID 00B1H) 59 "RD_REC"
1	通道专用诊断数据 (包括数据记录0)	<ul style="list-style-type: none"> <li>• S7-300: 16字节</li> <li>• S7-400: 4 – 220字节</li> </ul>	51 "RDSYSST" (SSL_ID 00B2H和 00B3H) 59 "RD_REC"
2 - 127	用户数据	每个 ≤ 240字节	59 "RD_REC"
128 - 240	诊断数据	每个 ≤ 240字节	59 "RD_REC"

## 系统资源

如果有若干个异步数据记录传送接连被启动，而它们之间只有很短的时间间隔，则操作系统对系统资源的分配可保证所有的任务都被执行且相互之间不发生冲突。

如果所有可用的系统资源都在使用，则在参数RET\_VAL中会有显示。这种临时出错状况可通过简单的任务重复加以补救。

可“同时”激活的SFC任务的最大数量依CPU而定。详细信息请参考/70/和/101/。

## 7.2 使用SFC54 “RD\_DPARM”读取定义的参数

### 描述

通过SFC54 “RD\_DPARM” (读取定义的参数), 可从用STEP7组态的系统数据中读取指定地址模块的、编号为RECNUM的数据记录。读取的数据记录被传送到由参数RECORD开辟的目标区域。

参数	声明	数据类型	存储区域	描述
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	地址区的ID: B#16#54 =外设输入(PI) B#16#55 =外设输出(PQ) 若为混合型模块, 则指定最低地址的区域ID。 如果地址相同, 则指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的逻辑基址。对于混合模块, 指定两个地址中较低的一个。
RECNUM	INPUT	BYTE	I、Q、M、D、L、 常数	数据记录号(允许值: 0 – 240)
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果读取的数据记录长度与目标区域匹配, 且传送时未出错, 返回值是被读取的数据记录的字节数。 如果在功能激活时出错, 则返回值包含故障代码。
RECORD	OUTPUT	ANY	I、Q、M、D、L	被读一个数据记录的目标区域。只允许字节数据类型。

### 出错信息

用SFC57 “PARM\_MOD” 将参数分配给模块

## 7.3 使用SFC102 “RD\_DPARA”读取预定义参数

### 描述

通过SFC102 “RD\_DPARA”，可从从STEP7组态的系统数据中读取所选模块的编号为RECNUM的数据记录。读取的数据记录被传送到由参数RECORD开辟的目标区域。

### 工作原理

SFC102 “RD\_DPARA”以异步方式操作，也就是说，处理过程包含SFC的多次调用。在REQ = 1时调用SFC102以启动任务。

作业状态通过输出参数RET\_VAL和BUSY显示。

异步操作SFC中REQ、RET\_VAL和BUSY的含义。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	REQ = 1：读请求
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的任何地址。对于输出地址，必须置位最高位。
RECNUM	INPUT	BYTE	I、Q、M、D、L、 常数	数据记录号(允许值：0至240)
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能激活时出错，则返回值包含故障代码。此外：读取的数据记录的长度，单位为字节(如果读取的数据记录适合于目标区域且传送时未出错)。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1：任务尚未结束
RECORD	OUTPUT	ANY	I、Q、M、D、L	取回数据记录的目标区域。只允许字节数据类型

### 出错信息

参见使用SFC57 “PARM\_MOD”组态模块

## 7.4 使用SFC55 “WR\_PARM”写动态参数

### 描述

通过SFC55 “WR\_PARM” (写参数)，可将数据记录RECORD传送到指定地址的模块。传送到模块去的参数不会被STEP7组态的参数覆盖。

### 必要条件

- 要传送的数据记录不允许是静态的。

(请参考/71/和/101/ 有关静态数据记录的详细信息。)

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1: 写请求
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	地址区域的标识号: B#16#54 =外设输入(PI) B#16#55 =外设输出(PQ) 若为混合模块, 则指定最低地址的区域ID。 如果地址相同, 则指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的逻辑基址。对于混合模块, 则指定两个地址中较低的一个。
RECNUM	INPUT	BYTE	I、Q、M、D、L、 常数	数据记录号
RECORD	INPUT	ANY	I、Q、M、D、L	数据记录
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能激活时出错, 则返回值包含故障代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 写操作尚未完成。

### 输入参数RECORD

要传送的数据是在第一个SFC调用时从参数RECORD读取的。在第一个SFC调用期间, 从参数RECORD读取要传送的数据。如果传送数据记录使用的时间比调用过程的持续时间长, 参数RECORD的内容对于后续的SFC调用(同一作业)不再有意义。

## 出错信息

用SFC57“PARM\_MOD”将参数分配给模块

---

### **注意 (只对S7-400)**

如果出现常规出错W#16#8544，只是表明对于至少一个字节的包含数据记录的I/O存储区访问被拒绝。数据传送继续进行。

---



## 7.5 使用SFC56 “WR\_DPARM”写一条数据记录

### 描述

通过SFC56 “WR\_DPARM” (写一条数据记录)，可将STEP 7组态数据中编号为RECNUM的数据记录传送到指定地址的模块。使用本功能，数据记录是静态还是动态都没有关系。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1: 写请求
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	地址区域的标识号: B#16#54 = 外围输入(PI) B#16#55 = 外围输出(PO) 如果模块是混合模块, 指定最低地址的区域ID。如果两个地址相同, 则指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的逻辑基址。 对于混合模块, 则指定两个地址中较低的一个。
RECNUM	INPUT	BYTE	I、Q、M、D、L、 常数	数据记录号
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能激活时出错, 则返回值包含故障代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 写操作尚未完成。

### 出错信息

用SFC57 “PARM\_MOD” 将参数分配给模块

## 7.6 用SFC57 “PARM\_MOD” 将参数分配给模块

### 描述

通过SFC57 “PARM\_MOD” (为模块分配参数), 可将由STEP 7组态生成的某一模块中的全部数据记录传送到该模块。使用本功能, 数据记录是静态还是动态都没有关系。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1: 写请求
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	地址区域的标识号: B#16#54 = 外围输入(PI) B#16#55 = 外设输出(PQ) 若是混合模块, 则指定最小地址的区域标识号。如果地址相同, 则指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的逻辑基址。 对于混合模块, 指定两个地址中较低的一个。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能激活时出错, 则返回值包含故障代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 写操作尚未完成。

### 出错信息

“真实的”出错信息(故障代码W#16#8xyz)可分为两类:

- 暂时故障(故障代码W#16#80A2到80A4、80Cx):  
对于此类故障, 可能在不采取任何措施的情况下就会消除, 换句话说, 重新调用该SFC (必要时多次调用)是可行的。  
例如, 所需资源当前正在使用(W#16#80C3)时, 即会出现暂时故障。
- 永久故障(故障代码W#16#809x、80A1、80Bx、80Dx):  
如果不采取措施, 此类故障就不会消除。只有消除故障后, 再次调用SFC才能成功。  
例如, 在RECORD中输入长度出错(W#16#80B1)即为一个永久故障。

### 注意

如果使用SFC 55、56或57向DPV1从站传送数据记录且该从站运行于DPV1模式, DP主站会按如下原则判断它从此从站接收的出错信息:

如果故障代码处于从W#16#8000到W#16#80FF或从W#16#F000到W#16#FFFF的区间内, DP主站会将出错信息传送给SFC。如果出错代码代码不在上述区间, 则CPU将数值W#16#80A2传送给SFC, 并挂起此从站。

SFC54 “RD\_DPARM”、SFC55 “WR\_PARM”、SFC56 “WR\_DPARM”和SFC57 “PARM\_MOD”的出错信息。

出错代码 (W#16#...)	说明	限制
7000	首次调用REQ=0: 无数据传送被激活; BUSY值为0。	-
7001	首次调用REQ=1: 数据传送已开始; BUSY值为1。	分布式I/O
7002	中间调用(REQ无关联): 数据传送激活; BUSY值为1。	分布式I/O
8090	指定的逻辑基址无效: SDB1/SDB2x中没有赋值, 或者没有基址。	-
8092	在ANY参考中指定的类型不是BYTE。	适用于S7-400的 SFC54 “RD_PARM”和 SFC55 “WR_PARM”
8093	此SFC不允许由LADDR和IOID所指定的模块(允许下列模块: S7-300模块用于S7-300、S7-400模块用于S7-400、S7-DP 模块用于S7-300和S7-400)。	-
80A1	当向模块发送数据记录时, 收到消极的应答(模块已拆除或在 传送期间损坏)。	1)
80A2	DP协议在第2层出错, 可能DP从站的硬件/接口故障	分布式I/O 1)
80A3	DP协议与用户接口/用户出错。	分布式I/O 1)
80A4	通讯总线上的通讯问题。	CPU与外部DP接口模块之 间出现故障1)
80B0	SFC不能用于此类型模块, 模块不识别数据记录。	1)
80B1	传送的数据记录的长度不正确。对于SFC54 “RD_PARM”: 由RECORD打开的目标区域太短。	-
80B2	组态的插槽未占用。	1)
80B3	实际模块类型与SDB1中要求的模块类型不 匹配。	1)
80C1	对于模块的同一数据记录, 上一写任务的数据还未被模块处理 数据。	1)
80C2	模块正在处理对一个CPU允许的最大数量的任务。	1)

出错代码 (W#16#...)	说明	限制
80C3	需要的资源(存储区等)当前被占用。	1)
80C4	内部临时出错。无法处理任务。 <ul style="list-style-type: none"> <li>重复执行该任务。如果经常发生此故障，请检查您的系统是否存在电干扰源。</li> </ul>	1)
80C5	分布式I/O不存在。	分布式I/O 1)
80C6	由于优先级的中止(热重启或后台操作)，数据记录传送已停止	分布式I/O 1)
80D0	在对应模块的SDB中无条目。	-
80D1	数据记录在对应模块的SDB中未组态(数据记录号 = 241被STEP 7拒绝)。	-
80D2	根据模块类型标识符，该模块不能被分配参数。	-
80D3	不能访问该SDB，因为它不存在。	-
80D4	SDB结构出错：SDB内部指针指向SDB以外的值。	仅对S7-300
80D5	数据记录是静态的。	仅对SFC55 “WR_PARM”
8xyy	一般故障代码，参见使用输出参数RET_VAL判断故障	

1 在SFC54 “RD\_DPARM”中不会发生

## 7.7 使用SFC58 “WR\_REC”写入数据记录

### 描述

通过SFC58 “WR\_REC” (写入记录), 可将包含在RECORD中的数据记录传送到指定地址的模块。

调用SFC58时, 给输入参数REQ赋值1, 可启动写任务。如果写任务可以立即执行, 此SFC将在输出参数BUSY中返回值0。如果BUSY值为1, 说明写任务尚未完成。

### 注意

如果已用一个GSD文件(版本3以上的GSD)对DPV1从站进行了组态, 且DP主站的DP接口已设置为“S7兼容”, 则不能使用SFC53从用户程序的I/O模块读一个数据记录。其原因是这种情况下的DP主站将寻址出错的插槽(已组态的插槽+3)。

纠正方法: 将DP主站的接口设置为“DPV1”。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1: 写请求
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	地址区域的标识号: B#16#54 = 外围输入(PI) B#16#55 = 外设输出(PQ) 若是混合模块, 指定最低地址的区域ID。 若地址相同, 指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的逻辑基本地址。对于混合模块, 则指定两个地址中较低的一个。
RECNUM	INPUT	BYTE	I、Q、M、D、L、 常数	数据记录号(允许值: 2 – 240)
RECORD	INPUT	ANY	I、Q、M、D、L	数据记录。只允许数据类型BYTE。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能激活时出错, 则返回值包含故障代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 写操作尚未完成。

## 输入参数RECORD

要传送的数据是在第一次SFC调用时从参数RECORD读取的。如果数据记录的传送占用的时间长于一个SFC调用的持续时间，则参数RECORD的内容不再与后续的SFC调用(同一任务)相关。

## 出错信息

使用SFC59 “RD\_REC”读一个数据记录

---

### 注意

如果出现常规出错W#16#8544，只是表明对于至少一个字节包含数据记录的I/O存储区的访问遭到拒绝。数据传送继续进行

---

## 7.8 使用SFC59 “RD\_REC”读一个数据记录

### 描述

通过SFC59 “RD\_REC” (读取记录), 可从指定地址的模块中读取编号为RECNUM的数据记录。通过调用SFC59且将输入参数REQ置1启动读任务。如果读任务可以立即执行, 此SFC在输出参数BUSY中返回值0。如果BUSY值为1, 则读任务尚未完成(参见“通过输出参数RET\_VAL判断出错”)。如果数据传送没有出错, 读取的数据记录将被传送到由RECORD参数指定的目标区域中。

### 注意

如果从1997年2月以前购买的FM或CP(下文称“旧式模块”)中读取编号 > 1的数据记录, SFC59的响应会有所不同。这一特殊状况在章节“使用旧的S7-300 FM和CP数据记录编号 >1”中有说明。

### 注意

如果使用GSD文件(版本3以上的GSD)对DPV1从站进行组态, 且DP主站的DP接口设置为“S7兼容”则一定不能在用户程序中通过SFC59从I/O模块读一个数据记录。其原因是这种情况下的DP主站将寻址出错的插槽(已组态的插槽+3)。

纠正方法: 将DP主站的接口设置为“DPV1”。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1: 读请求
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	地址区域的标识号: B#16#54 = 外围输入(PI) B#16#55 = 外设输出(PQ) 若是混合模块, 指定最低地址的区域ID。 如果两个地址相同, 指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的逻辑基本地址。对于混合模块, 则指定两个地址中较低的一个。
RECNUM	INPUT	BYTE	I、Q、M、D、L、 常数	数据记录号(允许值0 – 240)
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能激活时出错, 则返回值包含故障代码。如果目标区域大于传送的数据记录且传送中未出现出错, 则返回值是被传送的数据记录实际字节数(可能值: +1 - +240)。

参数	声明	数据类型	存储区域	描述
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 读操作尚未完成。
RECORD	OUTPUT	ANY	I、Q、M、D、L	读一个数据记录的目标区域。 异步执行SFC59时, 要确保每次调用时参数RECORD的实际值具有相同的长度信息。 只允许数据类型BYTE。

### 输出参数RET\_VAL

- 如果在执行此功能时出错, 则返回值包含故障代码。
- 如果未出错, RET\_VAL包含下列内容:
  - 0: 整个目标区域都被所选数据记录填充(数据记录也有可能不完整)。
  - 传送的数据记录的实际字节长度(可能值: +1 - + 240)  
(如果目标区域大于传送的数据记录)。

---

#### 注意

如果出现常规出错W#16#8545, 这仅表明对于至少一个字节包含数据记录的I/O存储区的访问遭到拒绝。数据记录已被模块正确读取并写入I/O存储区。

---

### 设定RECORD

---

#### 注意

如果要确保每次都读取完整的数据记录, 可将目标区长度选为241字节。如果数据传送无错, RET\_VAL包含被传送数据记录的实际字节数。

---



## 使用旧的S7-300 FM和CP数据记录号 > > 1

如果要使用SFC59 “RD\_REC”从旧的S7-300 FM和S7-300 CP中读取编号大于1的数据记录，请切记下列要点：

- 如果目标区域长度大于数据记录所需的实际长度，则数据不传入RECORD。  
RET\_VAL的值为W#16#80B1。
- 如果目标区域小于所请求数据记录的实际长度，CPU从记录的开始处开始读取指定量的字节(在RECORD长度信息中指定)，并将这些字节输入RECORD中。  
RET\_VAL值为0。
- 如果RECORD中指定的长度与数据记录的所需实际长度相同，CPU读一个数据记录并将其输入RECORD中。  
RET\_VAL值为0。

## 出错信息

下表中的“真实的”出错信息(故障代码W#16#8xyz)可分为两类：

- 暂时故障(故障代码W#16#80A2到80A4、80Cx)：  
对于此类故障，有可能在不采取任何措施的情况下故障即会消除，换句话说，再次调用该SFC (必要时多次调用)是可行的。  
例如，所需资源当前正在使用(W#16#80C3)时，即会出现暂时故障。
- 永久出错(出错代码W#16#809x、80A1、80Bx、80Dx)：  
如果不采取动作，此类出错就不会消除。只有消除出错后，重新调用SFC才能成功。例如，在RECORD中输入出错长度(W#16#80B1)即为一个永久出错。

---

### 注意

如果利用SFC58 “WR\_REC”向DPV1从站传送数据记录或利用SFC59 “RD\_REC”从DPV1从站读一个数据记录且此从站正以DPV1模式运行，则DP主站会按如下原则判断所接收的来自从站的出错信息：

如果出错信息处于从W#16#8000到W#16#80FF或从W#16#F000到W#16#FFFF的区间内，DP主站会将出错信息传送给SFC。如果出错代码代码不在上述区间，则CPU将数值W#16#80A2传送给SFC，并挂起从站。

对于来自DPV1从站出错信息的说明，请参见使用SFB 54 “RALRM”接收来自DP从站的中断STATUS[3]。

---

针对SFC58 “WR\_REC”和SFC59 “RD\_REC”的出错信息。

出错代码 (W#16#...)	说明	限制
7000	首次调用REQ=0: 无数据传送被激活; BUSY值为0。	-
7001	首次调用REQ=1: 无数据传送被激活; BUSY值为1。	分布式I/O
7002	中间调用(REQ无关联): 数据传送已经激活; BUSY值为1。	分布式I/O
8090	指定的逻辑基址无效: SDB1/SDB2x中没有赋值, 或者没有基址。	-
8092	在ANY参考中指定的类型不是BYTE。	只对S7-400
8093	本SFC不允许用于由LADDR和IOID所指定的模块(允许下列模块: S7-300模块用于S7-300、S7-400模块用于S7-400、 S7-DP模块用于S7-300和S7-400)。	-
80A0	当从模块读取数据时, 收到消极的应答 <ul style="list-style-type: none"> <li>模块在读任务期间被拆除或损坏</li> <li>另外, 对于H系统: 备用CPU一侧的I/O不可用(例如, CPU处于STOP状态)</li> </ul>	仅限SFC59 “RD_REC”
80A1	向模块发送数据记录时收到消极的应答 <ul style="list-style-type: none"> <li>该模块已在传送期间拆除或损坏</li> <li>另外, 对于H系统: 备用CPU一侧的I/O不可用(例如, CPU处于STOP状态)</li> </ul>	仅限SFC58 “WR_REC”
80A2	DP协议的第2层出错	分布式I/O
80A3	DP协议与用户接口/用户出错	分布式I/O
80A4	通讯总线上的通讯问题	出错发生在CPU与外部DP接口模块之间。
80B0	SFC不能用于此类型模块。 模块不能识别数据记录。 不允许使用数据记录号w 241。 对于SFC58 (WR_REC), 不允许数据记录号0和1。	-
80B1	RECORD参数中定义的长度不正确。	<ul style="list-style-type: none"> <li>SFC58 “WR_REC”: 长度不正确</li> <li>SFC59 “RD_REC” (仅在使用旧S7-300 FM和S7-300 CP时): 指定长度 &gt; 记录长度</li> <li>对于SFC13 “DPNRM_DG”: 指定长度 &gt; 记录长度</li> </ul>

出错代码 (W#16#...)	说明	限制
80B2	组态的插槽未占用。	-
80B3	实际模块类型与SDB1中要求的模块类型不匹配	-
80C0	对SFC59 (RD_REC): 模块有数据记录, 但是还没有要读取的数据。 对于SFC13 (DPNRM_DG): 诊断数据不存在。	仅限SFC59 “RD_REC” 或SFC13 “DPNRM_DG”
80C1	对于模块的同一数据记录, 上一个写任务的数据还未被模块处理。	-
80C2	模块正在处理对一个CPU所可能的最大数量的任务。	-
80C3	需要的资源(存储区等)当前被占用。	-
80C4	内部暂时故障。无法处理任务。 • 重复执行该任务。如果经常发生此故障, 请检查您的系统是否存在电干扰源。	-
80C5	分布式I/O不存在。	分布式I/O
80C6	由于优先级的中止(热重启或后台), 数据记录传送已停止	分布式I/O
8xyy	一般故障代码, 请参见使用输出参数RET_VAL判断故障	

## 7.9 SFC 55到59的进一步出错信息

### 只对S7-400

使用S7-400时SFC 55到59也可返回出错信息W#16#80Fx。在此情况下，将产生无法定位的故障。此时，请联系维护部门。

## 8 符合PNO AK 1131的DPV1 SFB

### 8.1 使用SFB “RDREC”从DP从站读一个数据记录

---

#### 注意

SFB52 “RDREC”接口与“符合IEC 61131-3标准的PROFIBUS指南PROFIBUS 通讯和代理功能块”标准中定义的FB “RDREC”完全相同。

---

#### 描述

利用SFB52 “RDREC” (读取记录), 可以从已通过ID寻址的DP从站组件(模块或子模块)中读取编号为INDEX的数据记录。

在MLEN中指定要读取的最多字节数。目标区域RECORD的选定长度至少应等于MLEN字节的长度。

输出参数VALID如为TRUE, 则表明已将数据记录成功传送到目标区域RECORD中。此时, 输出参数LEN包含所取得的数据的长度(以字节计)。

输出参数ERROR用以指示是否发生数据记录传送出错。如果发生出错, 则输出参数STATUS包含出错信息。

---

#### 注意

如果使用GSD文件(版本3以上的GSD)对DPV1从站进行组态, 且DP主站的DP接口设置为“S7兼容”则一定不能在用户程序中通过SFC52从I/O模块读一个数据记录。其原因是这种情况下的DP主站将寻址出错的插槽(已组态的插槽+3)。

纠正方法: 将DP主站的接口设置为“DPV1”。

---

#### 工作原理

SFC52 “RDREC”以异步方式工作, 即处理过程涉及多个SFC调用。通过使用REQ = 1调用SFC52来启动作业。

作业状态通过输出参数BUSY和输出参数STATUS的2、3字节显示。其中, STATUS的字节2和3对应于异步工作SFC的输出参数RET\_VAL(参见异步工作SFC中REQ、RET\_VAL和BUSY的含义)。

如果输出参数BUSY = FALSE, 则表明数据记录传送完成。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、常数	REQ = 1: 传送数据记录
ID	INPUT	DWORD	I、Q、M、D、L、常数	DP从站组件(模块或子模块)的逻辑地址 对于输出模块, 必须设置位15(例如对于地址5: ID: =DW#16#8005)。 对于组合模块, 应确定两个地址中的较小地址。
INDEX	INPUT	INT	I、Q、M、D、L、常数	数据记录号。
MLEN	INPUT	INT	I、Q、M、D、L、常数	要获取的数据记录信息的最大长度 (以字节计)
VALID	OUTPUT	BOOL	I、Q、M、D、L	新数据记录已被接收且有效
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 读取过程尚未终止。
ERROR	OUTPUT	BOOL	I、Q、M、D、L	ERROR = 1: 发生读出错。
STATUS	OUTPUT	DWORD	I、Q、M、D、L	调用ID(字节2和3)或出错代码
LEN	OUTPUT	INT	I、Q、M、D、L	所获取数据记录信息的长度
RECORD	IN_OUT	ANY	I、Q、M、D、L	所获取数据记录的目标区域。

## 出错信息

参见用SFB54“RALRM” STATUS[3]从DP从站接收中断

## 8.2 使用SFB53“WRREC”将数据记录写入DP从站

---

### 注意

SFB52 “WRREC”接口与在“符合IEC 61131-3标准的PROFIBUS指南PROFIBUS 通讯和代理功能块”标准中定义的FB “WRREC”完全相同。

---

### 描述

利用SFB52 “WRREC” (写入记录)，可以向已通过ID寻址的DP从站组件(模块或子模块)传送编号为INDEX的数据记录。

指定要传送的数据记录的字节长度。因此，源区域RECORD的选定长度至少应等于LEN字节的长度。

输出参数DONE如为TRUE，则表明已将数据记录成功传送到DP从站。

输出参数ERROR用以指示是否发生数据记录传送出错。如果发生出错，则输出参数STATUS包含出错信息。

---

### 注意

如果已用一个GSD文件(版本3以上的GSD)对DPV1从站进行了组态，且DP主站的DP接口已设置为“S7兼容”，则不能使用SFC53将数据记录写入用户程序的I/O模块。因为此时DP主站将寻址出错的插槽(已组态的插槽+3)。

纠正方法：将DP主站的接口设置为“DPV1”。

---

### 工作原理

SFC53 “WRREC”以异步方式工作，即处理过程涉及多个SFC调用。通过使用REQ = 1调用SFC53来启动作业。

作业状态通过输出参数BUSY和输出参数STATUS的2、3字节显示。其中，STATUS的字节2和3对应于异步工作SFC的输出参数RET\_VAL(参见异步工作SFC中REQ、RET\_VAL和BUSY的含义)。

请注意，必须为属于同一个数作业的所有SFB53调用的实际参数RECORD分配同一个值。该规则同样适用于LEN参数。

如果输出参数BUSY = FALSE，则表明数据记录传送完成。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、常数	REQ = 1: 传送数据记录
ID	INPUT	DWORD	I、Q、M、D、L、常数	DP从站组件(模块或子模块)的逻辑地址 对于输出模块，必须设置位15(例如对于地址5: ID: =DW#16#8005)。 对于组合模块，应确定两个地址中的较小地址。
INDEX	INPUT	INT	I、Q、M、D、L、常数	数据记录号。
LEN	INPUT	INT	I、Q、M、D、L、常数	要传送的数据记录的最大字节长度
DONE	OUTPUT	BOOL	I、Q、M、D、L	数据记录已传送
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 写操作尚未结束。
ERROR	OUTPUT	BOOL	I、Q、M、D、L	ERROR = 1: 发生写出错。
STATUS	OUTPUT	DWORD	I、Q、M、D、L	调用ID(字节2和3)或出错代码
RECORD	IN_OUT	ANY	I、Q、M、D、L	数据记录

## 出错信息

参见用SFB54“RALRM” STATUS[3]从DP从站接收中断



## 8.3 用SFB54“RALRM” STATUS[3]从DP从站接收中断

### 注意

SFB52 “RALRM”接口与“符合IEC 61131-3标准的PROFIBUS指南PROFIBUS通讯和代理功能块”标准中定义的FB “RALRM”完全相同。

### 描述

#### SFB

“RALRM”将从外围设备模块(集中结构)或者从DP从站组件接收含所有相应信息的中断。然后将此信息提供给输出参数。

输出参数中的信息包含被调用OB的启动信息以及中断源的信息。

由于要检查外围设备中断，故只能在由CPU操作系统启动的中断OB中调用SFB54。

### 注意

如果要在启动事件并非由外围设备触发的OB中调用SFB54 “RALRM”，则SFB输出中提供的信息便会相应减少。

确保在不同的OB中调用SFB 54 “RALRM”时使用不同的背景数据块。如果要在相关中断OB外部计算SFB54调用结果数据，则对每个OB启动事件应该使用单独的背景数据块。

### SFB 54调用

可以在三种工作模式(MODE)下调用SFB54 “RALRM”。下表予以说明。

MODE	SFB54 ...
0	...在输出参数ID中显示触发中断的组件并将输出参数NEW设置为TRUE。
1	...描述所有输出参数，与中断触发组件无关。
2	...检查是否是输入参数F_ID中指定的组件触发了中断。 <ul style="list-style-type: none"> <li>• 如果未触发中断，则NEW = FALSE</li> <li>• 如果已触发中断，则NEW = TRUE，且描述所有其它输出参数</li> </ul>

参数	声明	数据类型	存储区域	描述
MODE	INPUT	INT	I、Q、M、D、L、常数	工作模式
F_ID	INPUT	DWORD	I、Q、M、D、L、常数	要从中接收中断的组件(模块)的逻辑起始地址
MLEN	INPUT	INT	I、Q、M、D、L、常数	要接收的数据中断信息最大长度(以字节计)
NEW	OUTPUT	BOOL	I、Q、M、D、L	已接收新中断。
STATUS	OUTPUT	DWORD	I、Q、M、D、L	SFB或DP主站的出错代码
ID	OUTPUT	DWORD	I、Q、M、D、L	已从中接收中断的组件(模块)的逻辑起始地址。 位15包含I/O ID: 0代表输入地址; 1代表输出地址
LEN	OUTPUT	INT	I、Q、M、D、L	已接收中断信息的长度
TINFO	IN_OUT	ANY	I、Q、M、D、L	(任务信息) OB启动和管理信息的目标范围
AINFO	IN_OUT	ANY	I、Q、M、D、L	(中断信息) 标题信息和附加中断信息的目标区域 至少应为AINFO提供MLEN字节。

**当心**

如果选择的目标区域TINFO或AINFO能容纳的字节数过少，则SFB54不能输入完整信息。

**目标区域TINFO的数据结构**

字节	含义
0至19	当前已调用SFB54的OB的启动信息： <ul style="list-style-type: none"> <li>• 字节0到11：结构类似于SFC6SP;“RD_SINFO”中的参数TOP_SI</li> <li>• 字节12到19：请求OB的日期和时间</li> </ul>
20至27	管理信息

## 管理信息的结构

TINFO 的字节号	数据类型	含义					
20	BYTE	集中式:	0				
		分布式:	DP主站系统标识号(可能的值为1到255)				
21	BYTE	集中式:	模块机架号(可能的值: 0至31)				
		分布式:	DP站号(可能的值: 0至127)				
22	BYTE	集中式:	0				
		分布式:	• 位0至3:	从站类型	0000:	DP	
					0001:	DPS7	
					0010:	DPS7 V1	
				0011:	DPV1		
				自0100起:	保留		
			• 位4到7:	配置文件类型	0000:	DP	
					自0001起:	保留	
23	BYTE	集中式:	0				
		分布式:	• 位0至3:	中断信息类型	0000:	透明 (中断起源于已组态的分布式模块)	
					0001:	代表 (源自一个非DPV1从站或一个未组态的插槽的中断)	
					0010:	生成 (中断在CPU中生成)	
					自0011起:	保留	
				• 位4到7:	结构版本	0000:	初始
						自0001起:	保留
24	BYTE	集中式:	0				
		分布式:	DP主站接口的标记				
			• 位0 = 0:	中断源于集成DP接口			
			• 位0 = 1:	中断源于外部DP接口			
			• 位1到7:	保留			

TINFO的字节号	数据类型	含义	
25	BYTE	集中式:	0
		分布式:	DP从站接口的标记
			<ul style="list-style-type: none"> <li>位0: 诊断消息帧的EXT_DIAG_Bit位或0(如果中断中不存在此位)</li> </ul>
			<ul style="list-style-type: none"> <li>位1到7: 保留</li> </ul>
26至27	WORD	集中式:	0
		分布式:	PROFIBUS ID编号

### 目标区域AINFO的数据结构

字节	含义	
0至3	报头信息	
4至223	附加中断信息: 针对相关中断的模块特定数据:	
	<ul style="list-style-type: none"> <li>集中式:</li> </ul>	ARRAY[0]到ARRAY[220]
	<ul style="list-style-type: none"> <li>分布式:</li> </ul>	ARRAY[0]到ARRAY[59]

### 报头信息的结构

字节	数据类型	含义	
0	BYTE	已接收中断信息的长度(以字节计)	
		<ul style="list-style-type: none"> <li>本地:</li> </ul>	1至224
		<ul style="list-style-type: none"> <li>分布式:</li> </ul>	4至63
1	BYTE	本地:	保留
		分布式:	中断类型的ID
			1: 诊断中断
			2: 硬件中断
			3: 删除中断
			4: 插入中断
			5: 状态中断
			6: 更新中断
			31: 扩展设备、DP主站系统或DP站出现故障
			32至126 制造商特定 中断
2	BYTE	中断触发组件的插槽号	
3	BYTE	本地:	保留

字节	数据类型	含义	
		分布式:	识别
		位0和1	0: 无更多信息 1: 即将发生的事件, 插槽已损坏 2: 离开的事件, 插槽不再损坏 3: 离开的事件, 插槽仍损坏
		位2:	Add_Ack
		位3到7:	序号

### 目标区域AINFO的数据结构

根据调用SFB54的各个OB的不同, 可能仅会部分改写目标区域TINFO和AINFO。  
关于分别输入哪些信息, 请参考下表。

中断类型	OB	TINFO OB状态 信息	TINFO 管理 信息	AINFO 标题 信息	AINFO 附加中断 信息	
硬件中断	4x	是	是	是	集中式:	否
					分布式:	由DP从站交付
状态中断	55	是	是	是	是	
更新中断	56	是	是	是	是	
制造商特定中断	57	是	是	是	是	
外围设备 冗余出错	70	是	是	否	否	
诊断中断	82	是	是	是	集中式:	数据记录1
					分布式:	由DP从站交付
删除/插入中断	83	是	是	是	集中式:	否
					分布式:	由DP从站交付
模块机架/站故障	86	是	是	否	否	
...	所有 其它 OB	是	否	否	否	

## 出错信息

输出参数STATUS包含一些信息。它被解释为ARRAY[1...4] OF BYTE，出错信息的结构如下：

域元素	名称:	含义
STATUS[1]	Function_Num	<ul style="list-style-type: none"> <li>B#16#00(如果无故障)</li> <li>DPV1-PDU的功能ID: 在出错情况下, 与B#16#80进行或链接。如果未使用DPV1协议单元: B#16#C0。</li> </ul>
STATUS[2]	Error Decode	出错ID的位置
STATUS[3]	Error_Code_1	出错ID
STATUS[4]	Error_Code_2	制造商特定出错ID扩展

STATUS[2]可具有以下值:

Error Decode (B#16#...)	源	含义
00到7F	CPU	无故障或无警告
80	DPV1	因不符合IEC 61158-6而出错
81到8F	CPU	B#16#8x显示在SFB的第n个调用参数中存在出错。
FE、FF	DP配置文件	配置文件特定出错

STATUS[3]可具有以下值:

Error Decode (B#16#...)	Error_Code_1 (B#16#...)	按照DVP1的解释	含义
00	00		无故障, 无警告
70	00	保留, 拒绝	初始调用; 未传送活动数据记录
	01	保留, 拒绝	初始调用; 已开始传送数据记录
	02	保留, 拒绝	中间调用; 已激活数据记录传送
80	90	保留, 通过	逻辑起始地址无效
	92	保留, 通过	ANY指针的非法类型
	93	保留, 通过	通过ID或F_ID寻址的DP组件未组态。
	95		获取附加中断信息时在H系统中出错(通过外部DP接口在本地或分布式I/O获取附加中断信息时, 此出错输出为“组出错”。) 注意: 连接或更新时, 附加中断信息可能会暂时不可用。

Error_Decode (B#16#...)	Error_Code_1 (B#16#...)	按照DVP1的解释	含义
	96		H系统中发生主备机切换，先前的主CPU已进入STOP模式。当时正在处理一个OB。SFB54不能提供OB启动信息、管理信息、标题信息或附加中断信息。可使用SFC6“RD_SINFO”读出OB启动信息。此外，还可以使用SFC13“DPNRM_DG”同步读取OB4x、55、56、57、82和83的受影响DP从站的当前诊断帧(地址信息来自OB启动信息)。
	A0	读出错	读模块时出现否定应答。
	A1	写出错	写模块时出现否定应答
	A2	模块故障	在第2层出现DP协议出错
	A3	保留，通过	直接数据链路映射或用户接口/用户出现DP协议出错
	A4	保留，通过	总线通讯中断
	A5	保留，通过	-
	A7	保留，通过	DP从站或模块已被占用(临时性出错)
	A8	版本冲突	DP从站或模块报告出现不兼容版本
	A9	功能不被支持	DP从站或模块不支持该功能
	AA到AF	用户特定	DP从站或模块报告其应用程序中出现制造商特定出错。请检查制造商提供的DP从站或模块文档。
	B0	索引无效	数据记录在模块中为未知 数据记录号 256，非法
	B1	写长度出错	在参数RECORD中指定的长度出错；对于SFB54： AINFO中出现长度出错
	B2	插槽无效	未使用已组态的插槽。
	B3	类型冲突	实际模块类型与指定模块类型不同
	B4	区域无效	DP从站或模块报告访问的是无效区域
	B5	状态冲突	DP从站或模块未就绪
	B6	访问被拒绝	DP从站或模块拒绝访问
	B7	范围无效	DP从站或模块报告参数或值的范围无效
	B8	参数无效	DP从站或模块报告参数无效
	B9	类型无效	DP从站或模块报告类型无效
	BA到BF	用户特定	DP从站或模块报告在访问时出现制造商特定出错。 请检查制造商提供的DP从站或模块文档。
	C0	读限制冲突	模块有数据记录，不过没有已读过的数据。

Error_Decode (B#16#...)	Error_Code_1 (B#16#...)	按照DVP1的解释	含义
	C1	写限制冲突	针对模块的相同数据记录的上一次写请求数据尚未被该模块处理。
	C2	资源忙	模块当前正在处理CPU允许的最大可能作业数。
	C3	资源不可用	所需操作资源当前已被占用。
	C4		内部临时出错。无法执行作业。 重复执行该作业。如果此出错频繁发生，请检查设备的电气干扰源。
	C5		DP从站或模块不可用。
	C6		由于取消优先级而使数据记录传送被取消
	C7		由于重启动(热重启)或冷重启DP主站而使作业被取消
	C8到CF		DP从站或模块报告制造商特定资源出错。请检查制造商提供的DP从站或模块文档。
	Dx	用户特定	DP从站特定。参考DP从站的描述。
81	00到FF		初始调用参数中存在出错(对于SFB54: MODE)
	00		非法工作模式
82	00到FF		第2个调用参数中存在出错
:	:		:
88	00到FF		第8个调用参数中存在出错(对于SFB54: TINFO)
	01		语法ID出错
	23		超出数量帧或目标区域过小
	24		范围ID出错
	32		DB/DI编号超出用户范围
	3A		对于区域ID DB/DI, DB/DI编号为空或指定DB/DI不存在
89	00到FF		第9个调用参数中存在出错(对于SFB54: AINFO)
	01		语法ID出错
	23		超出数量帧或目标区域过小
	24		范围ID出错
	32		DB/DI编号超出用户范围
	3A		对于区域ID DB/DI, DB/DI编号为空或指定DB/DI不存在
8A	00到FF		第10个调用参数中存在出错
:	:		:



Error_Decode (B#16#...)	Error_Code_1 (B#16#...)	按照DVP1的解释	含义
8F	00到FF		第15个调用参数中存在出错
FE、FF	00到FF		配置文件特定出错

出现DPV1出错时，DP主站会将STATUS[4]传递给CPU和SFB。如果不出现DPV1出错，则此值被设置为0，但对于SFB52则会出现以下例外情况：

- STATUS[4]包含RECORD中的目标区域长度(如果MLEN > RECORD中的目标区域长度)
- STATUS[4]=MLEN(如果实际数据记录长度 < MLEN < RECORD中的目标区域长度)

## 8.4 用SFB75“SALRM”向DP主站发送中断

---

### 注意

SFB75“SALRM”接口与在标准“符合IEC 61131-3标准的PROFIBUS指南PROFIBUS通讯和代理功能块”中定义的FB“SALRM”完全相同。

---

### 描述

SFB75“SALRM”从智能从站的用户程序向相关DP主站发送一个插槽中断，该DP主站在过渡区域(虚拟插槽)中访问该中断。这将启动DP主站中的相关OB。

可以与中断一起发送附加中断特定信息。在DP主站中，可以使用SFB54“RALRM”读取所有附加信息。

SFB75只能用在S7兼容模式中。

- DP: 主站使用GSD文件集成I从站。
- S7兼容: 在STEP 7中，通过HW配置将I从站与一个主站链接在一起。

### 工作原理

SFB75“SALRM”是异步SFB，这表示它的处理涉及多个SFB调用。通过使用REQ = 1调用SFB75来启动中断的传送。

在DP主站确认或取消中断处理之前，发送过程始终保持活动状态。

输出参数BUSY与输出参数STATUS的字节2和3指示作业状态，其中STATUS的字节2和3对应于异步工作SFC的RET\_VAL输出参数(参见异步SFC中参数REQ、RET\_VAL和BUSY的含义)

如果输出参数BUSY = FALSE，则中断传送完成。

## 识别作业

当使用SFB 75向DP主站启动中断传送后，如果在完成当前作业前再次调用此SFB，则此SFB的进一步响应将取决于新调用是否与前一个调用完全相同。

如果参数ID和TYPE与尚未完成的作业相匹配，则将会按顺序处理对SFB的新调用。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、常数	REQ = 1: 传送中断
ID	INPUT	DWORD	I、Q、M、D、L、常数	DP主站的过渡区域(虚拟插槽)的所有逻辑地址，与在DP从站中看到的相同(站的诊断地址和插槽2的逻辑地址除外)。请在低字中查找相关信息。在高字中输入零。位15包含I/O ID: 0 = 输入地址, 1 = 输出地址。
ATYPE	INPUT	INT	I、Q、M、D、L、常数	中断类型 中断类型的ID。可能的值: <ul style="list-style-type: none"> <li>• 1: 诊断中断</li> <li>• 2: 硬件中断</li> </ul>
ASPEC	INPUT	INT	I、Q、M、D、L、常数	中断标识符: <ul style="list-style-type: none"> <li>• 0: 无更多信息</li> <li>• 1: 进入的事件, 插槽有故障</li> <li>• 2: 离开的事件, 插槽不再有故障</li> <li>• 3: 离开的事件, 插槽仍存在故障</li> </ul>
LEN	INPUT	INT	I、Q、M、D、L、常数	要发送的附加中断信息的长度 (以字节计) 最大值: 16
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE=1: 中断已传送
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 中断传送仍忙
ERROR	OUTPUT	BOOL	I、Q、M、D、L	ERROR = 1: 出现出错。
STATUS	OUTPUT	DWORD	I、Q、M、D、L	出错信息
AINFO	IN_OUT	ANY	I、Q、M、D、L	中断信息 附加中断信息的源区域

## ATYPE参数

下表显示ATYPE的所有允许值，其中OB是在相关主站中启动的，且在DP模式中，允许使用给定的中断类型。

ATYPE	DPV1标准中的含义	S7 DP主站中的相关OB	DP模式	
			DP	S7兼容
1	诊断中断	诊断中断OB(OB82)	-	是
2	硬件中断	硬件中断OB(OB 40到47)	-	是

### 注意

除上表提供的信息外，中断类型的使用范围还受到DP主站的限制。

## 中断类型与S7主站工作模式的依存关系

对于S7兼容模式下的从站(在S7主站上工作)，仅当相关DP主站处于RUN模式(DP: 工作)下时才能发送硬件和诊断中断。如果DP主站处于STOP模式(DP: 清除)，中断将被禁用，SFB75将返回出错信息W#16#80C8。这种情况下，由用户负责在以后发送中断。

## ASPEC参数

按照适用标准，此参数将显示虚拟插槽的诊断状态。因此，仅当在发送诊断中断时，才能为ASPEC分配非零值。

由于诊断中断的S7附加中断信息(数据记录0)包含进入/离开信息(参见字节0中的位0)，必须如下描述附加中断信息字节0中的位0 (模块故障)：

ASPEC	AINFO中的“模块故障”位
0	0
1	1
2	0
3	1

## LEN参数

LEN参数用于确定要发送的附加中断信息的长度(以字节计)。允许的最大取值范围是0到16。

对于所有可能的中断类型，下表显示在智能从站各种模式中，LEN的取值。

中断类型	DP	S7兼容
诊断中断	-	4至16
硬件中断	-	4

下表显示为LEN分配一个不等于AINFO长度的值(以字节计)后，SFB75的响应。

LEN的值	SFB 75响应
<= 为AINFO指定的长度	SFB75向DP主站发送一个中断。传送的附加中断信息的字节数与在LEN中指定的字节数相同。
超出允许的取值范围(< 0或 > 16)	SFB75不发送中断。 出错信息: W#16#80B1, STATUS[4]=B#16#FF
> 为AINFO指定的长度	SFB75向DP主站发送一个中断。传送的附加中断信息的字节数与在AINFO长度信息中指定的字节数相同。 出错信息: W#16#00B1, STATUS[4]=AINFO中的长度信息

## AINFO参数

AINFO是附加中断信息的源区域。就智能从站而言，可以用想要的任何值填充此区域。但是，如果您使用的是S7系列的DP主站，则随中断一起发送的附加信息必须符合S7惯例。

如果要发送**诊断中断**(ATYPE=1)，则必须在数据记录0中输入正确的相关值，如必要还要在数据记录1中输入正确的相关值。

下表是向您建议的一组S7兼容条目。此时已设置“模块故障”位(参见上文)。在实际命名该位之前，此建议与缺省分配(恢复的智能从站或站在加电或者从STOP转换到RUN后的现有分配)保持一致。

数据记录编号	分配
0	B#16#01、0B、00、00
1	对于S7兼容模式: 数据记录0 + 12字节(含零)

(参见诊断数据结构概述)

## SFB75调用对模块状态信息和组出错LED(SF)的影响

对于所有CPU，智能从站的插槽属性都存储在模块状态信息中(参见SSL-ID W#16#xy91 - 模块状态信息)

使用SFB75发送诊断中断时，智能从站的操作系统会对基于AINFO字节0中的位0(此位已作为“模块故障”添加到状态信息中)的从站本地模块状态信息和SF LED产生影响。但I从站中未创建任何诊断缓冲区条目，并且未启动诊断中断OB。

## DP主站与I从站之间模块状态信息的一致性

下文介绍了各种情况，并论述了它们对模块状态信息的影响：

- 站恢复(导致在DP主站和I从站中启动OB86)  
这将对I从站和S7主站中的模块状态信息产生影响(将复位“模块故障”。)。如果在站恢复后，从用户角度来看I从站中存在故障，则必须通过SFB75调用将这些故障报告给DP主站。
- DP主站中STOP到RUN工作模式的过渡(导致I从站中启动OB82) I从站中的模块状态信息保持不变。DP主站将复位相关模块状态信息中的“模块故障”位。

为确保在S7兼容模式下DP主站和I从站之间的模块状态信息能保持一致，必须按如下所述对I从站重新操作：

- 对于每个无错的虚拟插槽，使用SFB75向DP主站发送一个离开的诊断中断。
- 对于每个有故障的虚拟插槽，使用SFB75向DP主站发送一个进入的诊断中断。
- I从站中STOP到RUN工作模式的过渡(导致DP主站中启动OB 82) DP主站中的模块状态信息保持不变；将复位I从站中的信息(复位“模块故障”)。  
为确保在S7兼容模式下DP主站和I从站之间的模块状态信息能保持一致，必须按如下所述对I从站重新操作：
  - 对于每个无错的虚拟插槽，使用SFB75向DP主站发送一个离开的诊断中断。
  - 对于每个有故障的虚拟插槽，使用SFB75向DP主站发送一个进入的诊断中断。

---

### 注意

由于SFB75功能异步运行，无法在启动OB过程中结束SFB75调用。换言之，在完成调用之前，这些调用必须在循环程序中运行。

---

**注意**

上文提到的主站和I从站模块状态信息之间的所有差别只能发生在通过SFB75接收诊断中断的那些插槽上。也就是说上文论述的纠正方法仅适用于此类插槽。

**出错信息**

STATUS输出参数包含出错信息。如果被解释为ARRAY[1 ... 4] OF BYTE，则此出错信息的结构将如下表所示

域元素	含义
STATUS[1]	<ul style="list-style-type: none"> <li>B#16#00: 无故障</li> <li>B#16#C0: I从站检测到的出错</li> </ul>
STATUS[2]、STATUS[3]	对应于SFC的RET_VAL输出参数
STATUS[4]	B#16#00, LEN和AINFO之间的一些长度冲突除外。这些例外显示在下表中。

下表显示SFB 75的所有特定出错信息。

ERROR	STATUS[2,3] (W#16# ...)	说明
0	0000	作业已完成，且没有出错。如果LEN < AINFO的长度，则只能传送附加中断信息中的LEN字节。
0	00B1	LEN > AINFO的长度。作业已完成。已传送AINFO中的附加中断信息。STATUS[4]将接收AINFO的长度。
0	7000	REQ=0的初始调用(空循环)。未发送中断。BUSY的值为0。
0	7001	REQ=1的初始调用。作业已启动。BUSY的值为1。
0	7002	中间调用(与REQ无关)。发送的中断尚未得到DP主站的确认。BUSY的值为1。
1	8090	在ID中指定的地址超出允许的地址范围或者未对其进行组态。
1	8091	<ul style="list-style-type: none"> <li>您已在组态中禁用该中断。</li> <li>此类从站不允许使用该中断。</li> </ul>
1	8092	AINFO中的数据无效(BYTE和BLOCK-DB是有效类型)
1	8093	ID属于虚拟插槽。不能从其中请求任何中断。
1	80B0	ASPEC <ul style="list-style-type: none"> <li>与AINFO的字节0中的位0不匹配</li> <li>使用的中断类型的值必须为0</li> <li>超出允许的取值范围</li> </ul>
1	80B1	LEN超出允许的取值范围。STATUS[4]接收B#16#FF。
1	80B5	不允许在DP主站调用SFB75。

ERROR	STATUS[2,3] (W#16# ...)	说明
	80C3	所需资源(内存等)此时已被占用。
1	80C5	分布式I/O设备此时不可用(即站出现故障)
1	80C8	在当前DP主站工作模式下不允许使用该功能(DP主站是S7主站, 并且处于STOP模式下)。



## 9 用于处理时钟中断的SFC

### 9.1 处理时钟中断

#### 定义

时钟中断将导致操作系统调用一个时钟中断OB(OB10到OB17)。

#### 调用条件

必须满足下列条件，操作系统才能够调用时钟中断OB：

- 必须使用下列工具为时钟中断OB分配了参数(启动日期和时间、执行)
  - STEP 7或
  - SFC28 “SET\_TINT”(在用户程序中)。
- 必须使用下列工具激活时钟中断OB
  - STEP 7或
  - SFC30 “ACT\_TINT”(在用户程序中)。
- 不得使用STEP 7取消选择时钟中断OB。
- 时钟中断OB必须存在于CPU中。
- 如果通过SFC30 “ACT\_TINT”设置中断，且指定OB执行**仅一次**，则当前时间必须尚未超过启动日期和时间。如果选择**定期**执行，则将在下一个周期结束(启动时间 + 指定周期时间的倍数)时调用日时钟中断OB。

#### 提示

可以使用STEP 7为时钟中断分配参数，然后在用户程序(SFC30 “ACT\_TINT”)中激活中断。

#### SFC28到SFC31的用途

以下各部分所介绍的系统功能SFC28到SFC31的用途如下：

- 设置日时钟中断(SFC28 “SET\_TINT”)
- 取消时钟中断(SFC29 “CAN\_TINT”)
- 激活时钟中断(SFC30 “ACT\_TINT”)
- 查询日时钟中断(SFC31 “QRY\_TINT”)

## 9.2 SFC 28到31的特征

### 状况与影响...

下表列出一些不同的状况，并说明它们对时钟中断的影响。

如果...	那么...
设置日时钟中断(通过调用SFC28; SET_TINT)	将取消当前的时钟中断。
取消时钟中断(通过调用SFC29; CAN_TINT)	将清除启动日期和时间。必须重新设置日时钟中断，然后才能将其激活。
调用不存在的时钟中断OB。	将自动生成优先级出错，这意味着操作系统将调用OB85。 如果OB85不存在，CPU将切换到STOP模式。
如果将实时时钟同步或向前调整时钟	
<ul style="list-style-type: none"> <li>时钟向前调整</li> </ul>	如果因为向前调整了时间而跳过启动日期/时间： <ul style="list-style-type: none"> <li>操作系统将调用OB80<sup>1)</sup>。</li> <li>调用OB80之后，操作系统还将调用未在OB80<sup>2)</sup>中操作的每个被跳过的日时钟中断OB(调用一次，无论跳过多少个周期)。</li> </ul> 如果OB80不存在，CPU将切换到STOP模式。
<ul style="list-style-type: none"> <li>时钟向后调整</li> </ul>	如果在时钟向后调整的这段时间里已调用时钟中断OB，则在第二次到达该时间时不会再调用它们。

- 1) OB80包含编码的启动事件信息，用以指示哪些时钟中断OB因向前调整了时钟而无法调用。启动事件信息中的时间对应于向前调整的时间。
- 2) 在跳过后被激活的日时钟中断的启动事件信息中的时间对应于第一个被跳过的日时钟中断的启动时间。

### 暖重启或冷重启

暖重启或冷重启过程中，用户程序中通过SFC指定的所有日时钟中断设置均将被清除。

然后，使用STEP 7设置的参数将生效。

## 执行时钟中断OB

下表列出了“执行”参数的不同影响。可以通过STEP 7或SFC28“SET\_TINT”设置此参数(输入参数PERIOD)。

日时钟中断OB的执行	响应
无 (只能通过STEP 7设置)	即使CPU中存在时钟中断OB，也会不执行。 可以在用户程序中使用SFC28“SET_TINT”重新分配参数(设置日时钟中断)。
单次	调用时钟中断OB后，将取消时钟中断。然后，可以重新设置和激活此中断。
定期 (每分钟、每小时、每天、每周、每月、每年)	如果激活中断时启动日期和时间已过，则时钟中断OB将在下一可能点(“启动日期/时间 + 所选周期的倍数”)上中断循环程序。 在极少数情况下，当再次调用时钟中断OB时，其处理可能尚未完成。 结果： <ul style="list-style-type: none"> <li>• 时间出错(操作系统将调用OB80；如果OB80不存在，则CPU将切换到STOP模式)。</li> <li>• 将在以后执行时钟中断OB。</li> </ul>

### 9.3 使用SFC28 “SET\_TINT”设置日时钟中断

#### 描述

通过SFC28 “SET\_TINT” (设置日时钟中断)，可以设置日时钟中断组织块的启动日期和时间。将忽略指定的启动时间的秒和毫秒值，并将其设置为0。

参数	声明	数据类型	存储区域	描述
OB_NR	INPUT	INT	I、Q、M、D、L、 常数	在时间SDT + PERIOD的倍数处启动的OB的编号(OB10到OB17)。
SDT	INPUT	DT	D、L、常数	启动日期和时间：将忽略指定的启动时间的秒和ms值 并将其设置为0。 如果要设置每月启动日时钟中断OB，则只能使用日期1、2、... 28号作为启动日期。
PERIOD	INPUT	WORD	I、Q、M、D、L、 常数	从启动点SDT开始的周期： W#16#0000 = 一次 W#16#0201 = 每分钟 W#16#0401 = 每小时 W#16#1001 = 每日 W#16#1202 = 每周 W#16#1401 = 每月 W#16#1801 = 每年 W#16#2001 = 月末
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能处于激活状态时出错， 则RET_VAL的实际参数将包含出错代码。

#### 出错信息

出错代码(W#16#...)	说明
0000	未出错
8090	不正确的参数OB_NR
8091	不正确的参数SDT
8092	不正确的参数PERIOD
80A1	设置的启动时间已过。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 9.4 使用SFC29 “CAN\_TINT”取消时钟中断

### 描述

使用SFC29 “CAN\_TINT” (设置日时钟中断)，可以取消一个已激活的日时钟中断组织块

参数	声明	数据类型	存储区域	描述
OB_NR	INPUT	INT	I、Q、M、D、L、 常数	OB的编号，将在其中取消启动日期和时间(OB10到OB17)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能处于激活状态时出错，则RET_VAL的实际参数将包含出错代码。

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	不正确的参数OB_NR
80A0	没有为时钟中断OB指定启动日期/时间
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 9.5 使用SFC30 “ACT\_TINT”激活时钟中断

### 描述

使用SFC30 “ACT\_TINT” (激活日时钟中断)，可以激活一个日时钟中断组织块。

参数	声明	数据类型	存储区域	描述
OB_NR	INPUT	INT	I、Q、M、D、L、 常数	将激活的OB的编号(OB10到OB17)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能处于激活状态时出错， 则RET_VAL的实际参数将包含出错代码。

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	不正确的参数OB_NR。
80A0	没有为相应的日时钟中断OB设置启动日期/时间。
80A1	激活的时间已过。只有当选择“执行 = 一次”时才会出现此出错。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 9.6 使用SFC31 “QRY\_TINT”查询日时钟中断

### 描述

使用系统功能SFC31 “QRY\_TINT” (查询日时钟中断)，可以在输出参数STATUS上显示日时钟中断组织块的状态。

参数	声明	数据类型	存储区域	描述
OB_NR	INPUT	INT	I、Q、M、D、L、 常数	将被查询状态的OB的编号(OB10到OB17)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能处于激活状态时出错，则RET_VAL的实际参数将包含出错代码。
STATUS	OUTPUT	WORD	I、Q、M、D、L	时钟中断的状态；参见下表。

### 输出参数STATUS

位	值	含义
0	0	操作系统已启用时钟中断。
1	0	接受新的时钟中断。
2	0	未激活时钟中断或时间已过。
3	-	-
4	0	未加载时钟中断OB。
5	0	当前活动的测试功能未禁用日时钟中断OB的执行。
6	0	时钟中断基于基准时间
	1	时钟中断基于当地时间

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	不正确的参数OB_NR
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。





## 10 用于处理延时中断的SFC

### 10.1 处理延时中断

#### 定义

调用SFC32 “SRT\_DINT”后，操作系统将在指定的延迟时间过去后生成一个中断，也就是说，调用所选择的延时中断OB。此中断被称为延时中断。此中断称为延时中断。

#### 调用条件

必须满足下列条件，操作系统才能调用延时中断OB：

- 必须通过SFC32 “SRT\_DINT”启动延时中断OB。
- 不得使用STEP 7取消选择延时中断OB。
- 延时中断OB必须存在于CPU中。

#### SFC32到SFC34的用途

以下各部分所介绍的系统功能SFC32到SFC34的用途如下：

- 启动延时中断(SFC32 “SRT\_DINT”)
- 取消延时中断(SFC33 “CAN\_DINT”)
- 查询日时钟中断(SFC34 “QRY\_DINT”)。

### 状况与影响...

下表列出了一些不同状况，并说明了它们对延时中断的影响。

如果...	且...	那么...
延时中断被启动 (通过调用SFC32 "SRT_DINT")。	该延时中断已经启动。	将重写延迟时间，并重新启动延时中断。
	在调用时延时中断OB不存在。	操作系统将生成优先级出错 (调用OB85)。 如果OB85不存在，CPU将切换到 STOP模式。
	CPU切换为RUN模式前，已在 启动OB中启动中断且延迟时 间已过。	将延迟对延时中断OB的调用， 直到CPU处于RUN模式。
延迟时间已过。	先前启动的延时中断OB 仍在执行。	操作系统将生成时间出错 (调用OB80)。如果OB80 不存在，CPU将切换到 STOP模式。

### 暖重启和冷重启

暖重启或冷重启期间，用户程序中使用SFC指定的所有延时中断设置均将被清除。

### 在启动OB中启动

可以在启动OB中启动延时中断。要调用延时中断OB，必须满足下列两个条件：

- 延迟时间必须已过。
- CPU必须处于RUN模式。

如果延迟时间已过而CPU未处于RUN模式，则延时中断OB调用将被延迟，直到CPU处于RUN模式。然后，将在执行OB1中的第一条指令前调用延时中断OB。

## 10.2 使用SFC32 “SRT\_DINT”启动延时中断

### 描述

通过SFC32 “SRT\_DINT” (启动延时中断), 可以在延迟时间过去(参数DTIME)后立即启动调用延时中断组织块的延时中断。

使用SIGN参数, 可以输入用于标识延时中断开始的标识符。执行指定的OB时, DTIME和SIGN值将再次显示在该OB的启动事件信息中。

参数	声明	数据类型	存储区域	描述
OB_NR	INPUT	INT	I、Q、M、D、L、 常数	将在延时后启动的OB的编号 (OB20到OB23)。
DTIME	INPUT	TIME	I、Q、M、D、L、 常数	延时值(1到60000ms) 要实现更长时间的延时, 例如, 可以通过在 延时中断OB中使用计数器来实现。
SIGN	INPUT	WORD	I、Q、M、D、L、 常数	调用延时中断OB时将显示在启动事件 信息中的标识符。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在系统功能处于激活状态时出错, 则RET_VAL的实际参数将包含出错代码。

### 精度

只要没有中断事件延迟该调用, 调用SFC32 “SRT\_DINT”和启动延时中断OB之间的时间最多比所选时间少**一毫秒**。

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	不正确的参数OB_NR
8091	不正确的参数DTIME
8xyy	常规出错信息, 请参见使用输出参数RET_VAL评估 出错。

## 10.3 使用SFC34 “QRY\_DINT”查询日时钟中断

### 描述

通过SFC34 “QRY\_DINT” (查询日时钟中断), 可以查询日时钟中断OB的状态。  
延时中断由组织块OB20到OB23管理。

参数	声明	数据类型	存储区域	描述
OB_NR	INPUT	INT	I、Q、M、D、L、 常数	将查询其STATUS的OB的编号 (OB20到OB23)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在处理功能期间出错, 则RET_VAL 的实际参数 将包含出错代码。
STATUS	OUTPUT	WORD	I、Q、M、D、L	有关延时中断的状态, 参见下表。

### 输出参数STATUS

位	值	含义
0	0	操作系统已启用延时中断。
1	0	未拒绝新的延时中断。
2	0	未激活延时中断或时间已过。
3	-	-
4	0	未加载延时中断OB。
5	0	当前活动的测试功能未禁用延时中断OB的执行。

### 出错信息

出错代码(W#16#...)	说明
0000	未出错
8090	不正确的参数OB_NR
8xyy	常规出错信息, 请参见使用输出参数RET_VAL评估出错。

## 10.4 使用SFC33 “CAN\_DINT”取消延时中断

### 描述

通过SFC33 “CAN\_DINT” (取消延时中断), 可以取消已启动的延时中断(参见使用SFC32 “SRT\_DINT”启动延时中断)。这样, 将不调用延时中断OB。然后, 将不调用延时中断OB。

参数	声明	数据类型	存储区域	描述
OB_NR	INPUT	INT	I、Q、M、D、L、 常数	将取消的OB的编号(OB20到OB23)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能处于激活状态时出错, 则RET_VAL的实际参数 将包含出错代码。

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	不正确的参数OB_NR
80A0	未启动延时中断。
8xyy	常规出错信息, 请参见使用输出参数RET_VAL评估出错。



# 11 用于处理同步出错的SFC

## 11.1 屏蔽同步出错

### 引言

同步出错是编程和访问出错。使用不正确的地址区域、编号或不正确的地址进行编程将导致此类出错的发生。**屏蔽**这些同步出错意味着：

- 屏蔽掉的同步出错不会触发出错OB调用，且不会导致所编程的备选处理。
- CPU将在出错寄存器中“记录”被屏蔽掉的已发生出错。

通过调用SFC36 “MSK\_FLT”执行屏蔽。

**取消屏蔽**出错意味着取消先前设置的屏蔽，并清除当前优先级的事件状态寄存器中的相应位。使用如下操作取消屏蔽：

- 调用SFC37 “DMSK\_FLT”
- 在完成当前优先级时取消屏蔽。

如果一个出错在取消屏蔽后又发生，则操作系统将启动相应的出错OB。可以编程OB121实现对编程出错的响应处理，编程OB122实现对访问出错的响应处理。

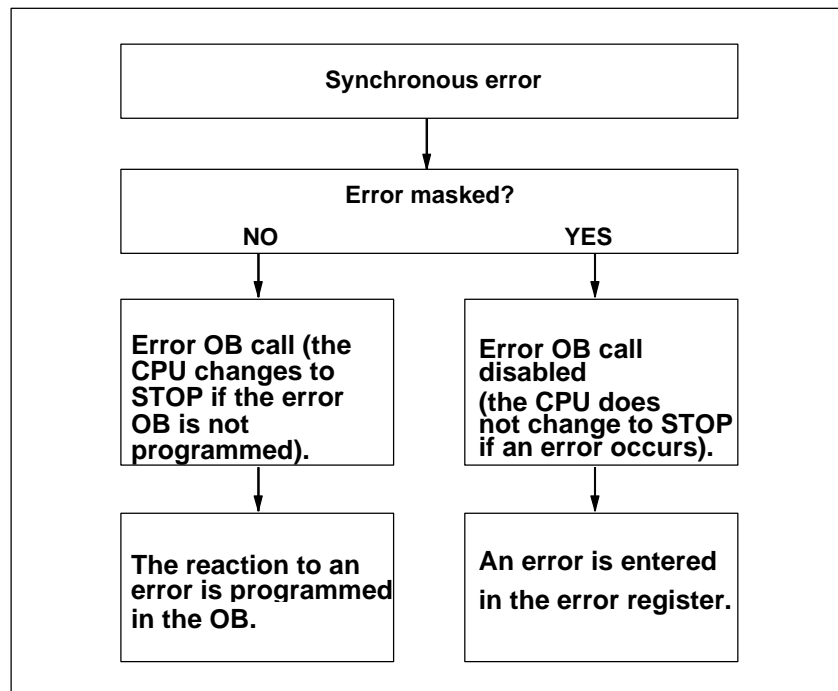
可以使用SFC38 “READ\_ERR”读出被屏蔽掉的已发生出错。

**注意：**注意：对于S7-300 (CPU 318除外)，无论是否屏蔽出错，出错都将进入诊断缓冲区，CPU的组出错LED都将亮起。

### 出错的常规处理

如果用户程序中出现编程和访问出错，可以不同方式做出响应：

- 可以对出现出错时操作系统调用的相应出错OB进行编程。
- 可以针对每个优先级单独禁用出错OB调用。这种情况下，当特定优先级中发生此类出错时，CPU将不切换到STOP模式。CPU将出错输入到出错寄存器中。但是，从此条目中无法识别出错的发生时间或频率。



## 过滤器

同步出错被分配给特殊的位模式(被称为**出错过滤器(屏蔽)**)。此出错过滤器也在SFC 36、37和38的输入和输出参数中。

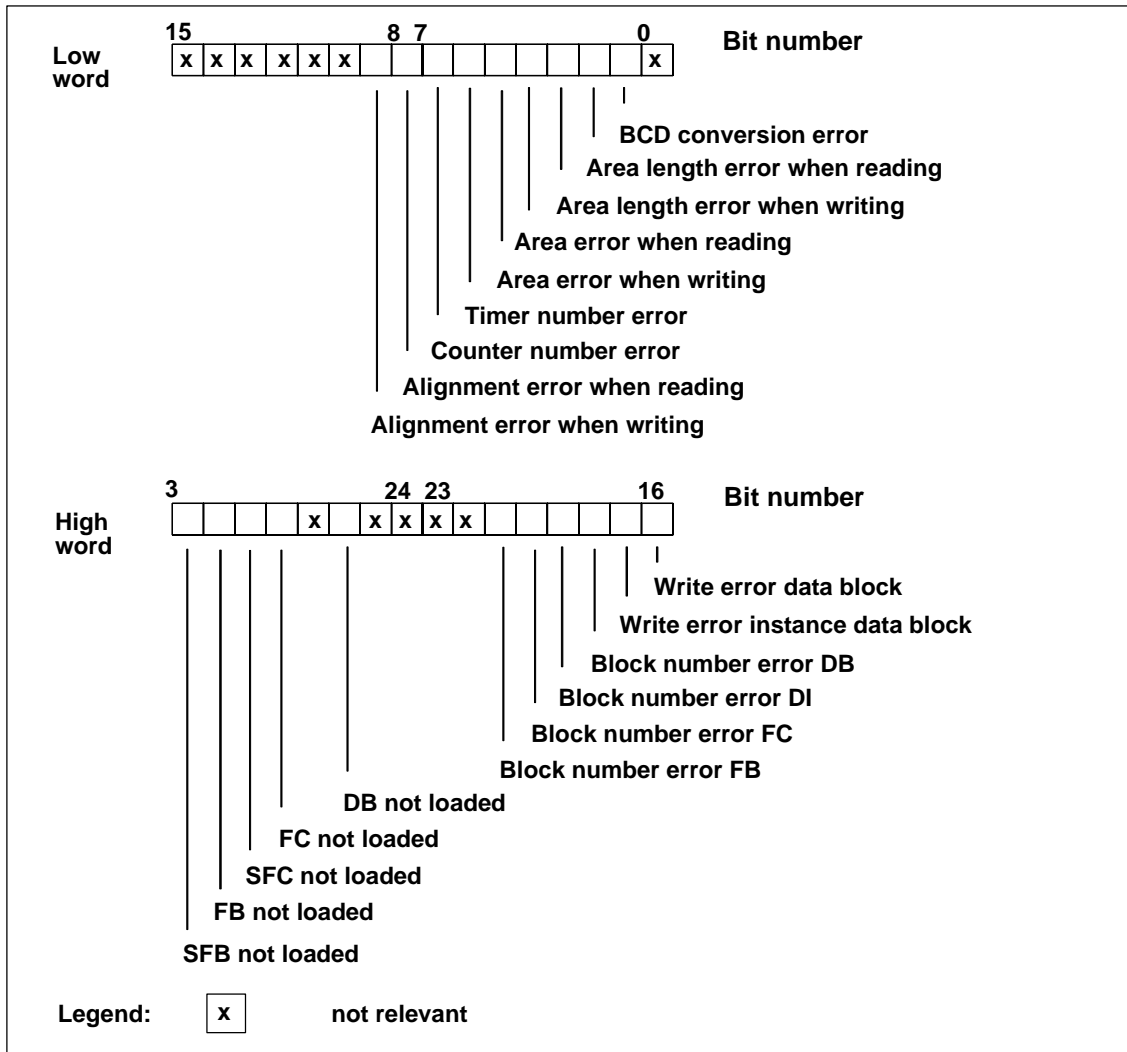
同步出错分为可以使用两种出错过滤器屏蔽的编程出错和访问出错。这些出错过滤器的图解如下。



### 编程出错过滤器

下图展示了编程出错的出错过滤器的位模式。编程出错的出错过滤器位于参数 PRGFLT\_...中。

请参考“编程出错的可能出错原因低位字”或“编程出错的可能出错原因高位字”



### 注意

编程出错过滤器的高位字中的29 (“未加载SFC”)和31 (“未加载SFB”)位仅适用于S7-400和CPU 318。

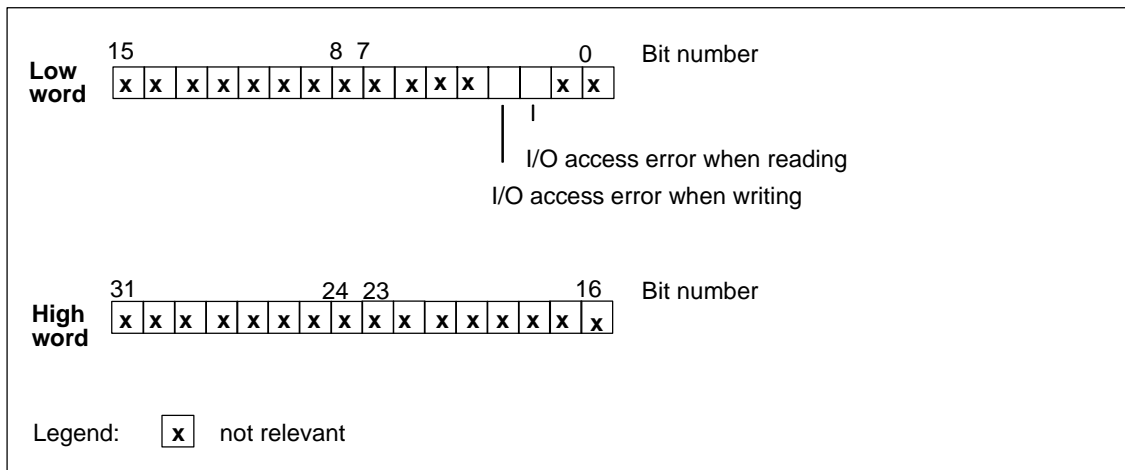
### 不相关位

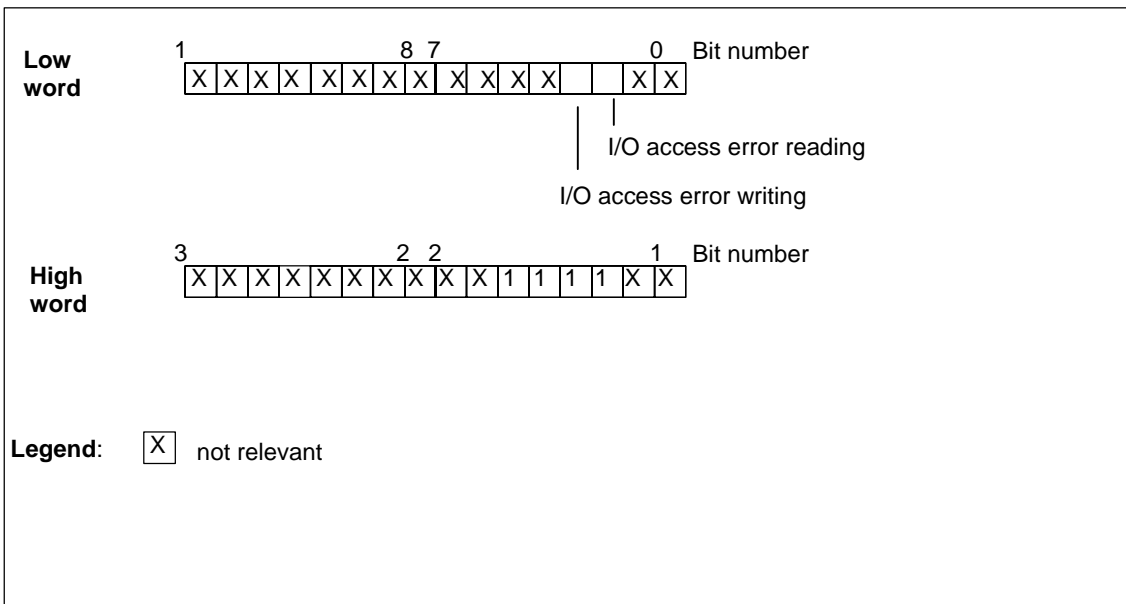
在上图中，**x**表示...

• ...输入参数	用于SFC36、37、38	= “0”
• ...输出参数	用于SFC36、37	= “1”用于S7-300 = “0”用于S7-400
	对于SFC38	= “0”

### 所有CPU的访问出错过滤器

下图显示了除CPU 417和CPU 417H外所有CPU的访问出错的出错过滤器的位模式。访问出错的出错过滤器在参数ACCFLT\_...中。有关访问出错的解释，请参考表格“除CPU 318外所有CPU 31x的可能出错原因”或“所有CPU 41x和CPU 318的可能出错原因”。





### 不相关位CPU 41x和CPU 318

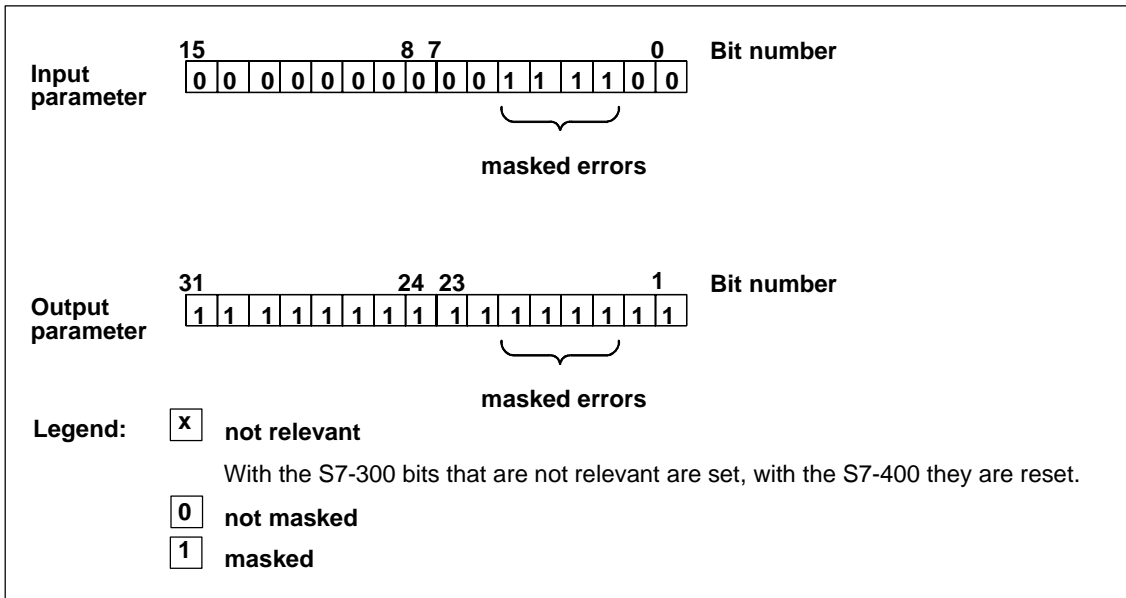
在上图中，x表示...

• ...输入参数	用于SFC36、37、38	= “0”
• ...输出参数	用于SFC36、37	= “0”
	对于SFC38	= “0”

### CPU 31x (不是CPU 318)的实例

下图显示了除CPU 417和CPU 417H外所有CPU的访问出错的出错过滤器的位模式。

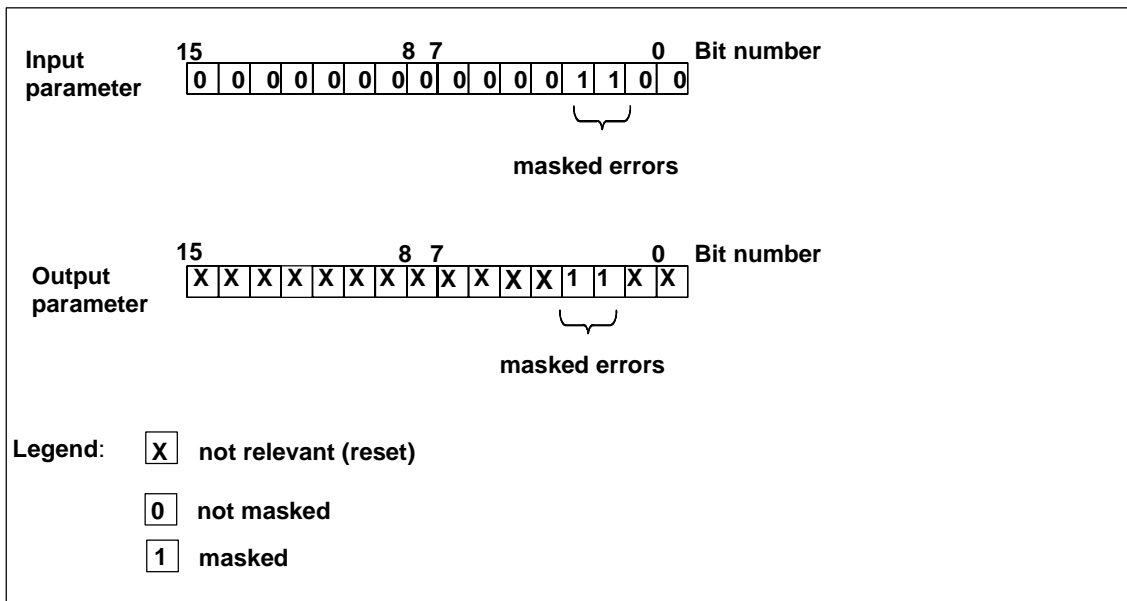
- 作为SFC36的输入参数
- 作为SFC36的输出参数



### CPU 417和CPU 417H的实例

下图说明CPU 417中和CPU 417H中用于访问出错的出错过滤器的低位字节及所有已屏蔽的出错是如何显示的。

- 作为SFC36的输入参数
- 作为SFC36的输出参数



### 编程出错低位字

下表列出了分配给用于编程出错的出错过滤器的低位字的出错。此表还列出了出错的可能原因。

编程出错的可能原因，低位字

出错	事件标识符 (W#16#...)	发生出错...
BCD转换出错	2521	...当将转换的值不是BCD数字时(例如, 5E8)
读取时区域长度出错	2522	...当使用的地址不全在可能的地址区域内时。 实例: 虽然存储器区域仅有256字节长, 但必须读取 MW 320。
写入时区域长度出错	2523	...当使用的地址不全在可能的地址区域内时。 实例: 虽然存储器区域仅有256字节长, 但必须将值写入 MW 320中。
读取时的区域出错	2524	...当使用间接跨区域寻址时, 为地址指定了不正确的区域标识符。 实例:            正确:     LAR1 P#E 12.0 L W[AR1, P#0.0] 不正确: LAR1 P#12.0 L W[AR1, P#0.0] 对于此操作, 将发送区域长度出错信号。

写入时区域出错	2525	<p>...当使用间接跨区域寻址时，为地址指定了不正确的区域标识符。</p> <p>实例：           正确：       LAR1 P#E 12.0                                   TW[AR1, P#0.0]</p> <p>                  不正确：   LAR1 P#12.0                                   TW[AR1, P#0.0]</p> <p>对于此操作，将发送区域长度出错信号。</p>
定时器编号出错	2526	<p>...当访问不存在的定时器时。</p> <p>实例： SP T [MW 0]，其中MW 0 = 129；虽然只有128个定时器可用，但必须启动定时器129。</p>
计数器编号出错	2527	<p>...当访问不存在的计数器时。</p> <p>实例： CU C [MW 0]，其中MW 0 = 600；虽然只有512个计数器可用，但必须启动计数器600 (CPU 416-D)。</p>
读取时的对齐出错	2528	<p>...当寻址字节、字或双字地址，而位地址 ≠ 0时。</p> <p>实例：           正确：       LAR1 P#M12.0                                   LB[AR1, P#0.0]</p> <p>                  不正确：   LAR1 P#M12.4                                   LB[AR1, P#0.0]</p>
写入时的对齐出错	2529	<p>...当寻址字节、字或双字地址，而位地址 ≠ 0时。</p> <p>实例：           正确：       LAR1 P#M12.0                                   TB[AR1, P#0.0]</p> <p>                  不正确：   LAR1 P#M12.4                                   TB[AR1, P#0.0]</p>

## 编程出错高位字

下表列出了分配给用于编程出错的出错过滤器的高位字的出错。同时列出了可能的出错原因。

编程出错的可能原因，高位字

出错	事件标识符 (W#16#...)	发生出错...
写入出错数据块	2530	...要写入的数据块为只读。
写入出错背景数据块	2531	...要写入的背景数据块为只读。
块编号出错DB	2532	...必须打开的数据块的编号高于允许的最大编号。
块编号出错DI	2533	...必须打开的背景数据块的编号高于允许的最大编号。
块编号出错FC	2534	...调用的功能的编号高于允许的最大编号。
块编号出错FB	2535	...调用的功能块的编号高于允许的最大编号。
未加载DB	253A	...要打开的数据块未加载。

出错	事件标识符 (W#16#...)	发生出错...
未加载FC	253C	...调用的功能未加载。
SFC不存在	253D	...调用的系统功能不存在。
未加载FB	253E	...要调用的功能块未加载。
SFB不存在	253F	...调用的系统/标准功能块不 存在。

## 访问出错

下表列出了分配给所有CPU的访问出错的出错过滤器的出错。同时列出了可能的出错原因

出错	事件标识符(W#16#...)	发生出错...
读取时的I/O 访问出错	2942	...未将任何信号模块分配给 I/O区域中的地址。 或 ...未在所选模块监视狗时间 内确认对此I/O区域的访问(超时)。
写入时的I/O 访问出错	2943	...未将任何信号模块分配给 I/O区域中的地址。 或 ...未在所选模块监视狗时间 内确认对此I/O区域的访问(超时)。

## 11.2 使用SFC36 “MSK\_FLT”屏蔽同步出错

### 描述

通过SFC36 “MSK\_FLT” (屏蔽同步出错)，可以控制CPU对同步出错的响应处理。通过此SFC，可以使用出错过滤器屏蔽同步出错(参见屏蔽同步出错)。当调用SFC36时，必须在当前优先级中屏蔽同步出错。

如果在输入参数中将同步出错过滤器的单个位设置为“1”，则先前设置的其它位将保留其值“1”。这样，就可以获得可以使用输出参数读取的新出错过滤器。已屏蔽的同步出错将不调用OB，而只是输入到出错寄存器中。可以通过SFC38 “READ\_ERR”读取出错寄存器。

参数	声明	数据类型	存储区域	描述
PRGFLT_SET_MASK	INPUT	DWORD	I、Q、M、D、L、 常数	将屏蔽 编程出错
ACCFLT_SET_MASK	INPUT	DWORD	I、Q、M、D、L、 常数	将屏蔽访问出错
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
PRGFLT_MASKED	OUTPUT	DWORD	I、Q、M、D、L	屏蔽编程出错
ACCFLT_MASKED	OUTPUT	DWORD	I、Q、M、D、L	屏蔽访问出错

### 出错信息

出错代码(W#16#...)	说明
0000	尚未屏蔽任何出错。
0001	至少已屏蔽了一项出错。 不过，其它出错也将屏蔽。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。



## 11.3 使用SFC37 “DMSK\_FLT”解除屏蔽同步出错

### 描述

通过SFC37 “DMSK\_FLT”(取消屏蔽同步出错)，可以取消屏蔽通过SFC “MSK\_FLT”屏蔽的出错。为此，必须在输入参数中将出错过滤器的相应位设置为“1”。通过SFC37调用，可以取消屏蔽当前优先级的相应同步出错。同时，在出错寄存器中清除这些条目。使用输出参数可以读取新的出错过滤器。

参数	声明	数据类型	存储区域	描述
PRGFLT_RESET_MASK	INPUT	DWORD	I、Q、M、D、L、 常数	将被取消屏蔽编程出错
ACCFLT_RESET_MASK	INPUT	DWORD	I、Q、M、D、L、 常数	将被取消屏蔽访问出错
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
PRGFLT_MASKED	OUTPUT	DWORD	I、Q、M、D、L	仍被屏蔽的编程出错
ACCFLT_MASKED	OUTPUT	DWORD	I、Q、M、D、L	仍被屏蔽的编程出错

### 出错信息

出错代码(W#16#...)	说明
0000	已取消屏蔽所有指定的出错。
0001	至少有一项出错未屏蔽。 不过，其它出错将被取消屏蔽。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 11.4 使用SFC38 “READ\_ERR”读取出错寄存器

### 描述

使用SFC38 “READ\_ERR” (读取出错寄存器), 可以读取出错寄存器。出错寄存器的结构对应于编程和访问出错过滤器的结构, 这些结构可以通过SFC36和SFC37作为输入参数编程。

在输入参数中, 输入要从出错寄存器读取的同步出错。调用SFC38时, 将从出错寄存器读取所需的条目, 同时将这些条目清除。

出错寄存器包含的信息可以让您了解, 在当前优先级中哪些被屏蔽的同步出错至少发生了一次。如果相应位被置位, 则意味着相应的屏蔽同步出错至少发生了一次。

参数	声明	数据类型	存储区域	描述
PRGFLT_QUERY	INPUT	DWORD	I、Q、M、D、L、 常数	查询编程出错
ACCFLT_QUERY	INPUT	DWORD	I、Q、M、D、L、 常数	查询访问出错
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
PRGFLT_CLR	OUTPUT	DWORD	I、Q、M、D、L	发生的编程出错
ACCFLT_CLR	OUTPUT	DWORD	I、Q、M、D、L	发生的访问出错

### 出错信息

出错代码(W#16#...)	说明
0000	已屏蔽所有被查询的出错。
0001	至少有一项被查询的出错未屏蔽。
8xyy	常规出错信息, 请参见使用输出参数RET_VAL评估出错。

## 12 用于处理中断和异步出错的SFC

### 12.1 延迟和禁用中断和异步出错

#### SFC39到SFC42的用途

通过这些SFC，可以达到下列目的：

- 通过SFC39 “DIS\_IRT”，对所有后续CPU周期禁用中断和异步出错。
- 通过SFC41 “DIS\_AIRT”，延迟更高的优先级直到OB结束。
- 通过SFC40 “EN\_IRT”或SFC42 “EN\_AIRT”，启用中断和异步出错。

您在用户程序中编程中断和异步出错的处理。同时还必须编程相应的OB。

#### SFC41和SFC42的优点

通过SFC41 “DIS\_AIRT”禁用更高优先级的中断和异步出错，然后再通过SFC52 “EN\_AIRT”重新启用它们，从而延迟更高优先级的中断和异步出错，这样做具有以下优点：

被延迟的中断数由CPU进行计数。如果已延迟中断和异步出错，而且也在标准FC自身中禁用并重新启用了这些中断和异步出错，则不能通过标准FC取消延迟。

## 中断等级

中断分为多个等级。下表列出了所有中断等级和相应的OB。

中断等级	OB
日历中断	OB10到OB17
延时中断	OB20到OB23
周期性中断	OB30到OB38
硬件中断	OB40到OB47
DPV1中断	OB55到OB57
多值计算中断	OB60
冗余出错中断	OB70、OB72
异步出错中断	OB80到OB87(参见下文)
同步出错中断	OB121、OB122 (可以通过SFC36到SFC38屏蔽或取消屏蔽同步出错中断的处理)

## 异步出错

下表列出了可以在用户程序中通过OB做出响应的所有异步出错。

异步出错	OB
时间出错(例如, 超出周期时间)	OB80
电源出错(例如, 电池故障)	OB81
诊断中断(例如, 信号模块上有故障的保险丝)	OB82
删除/插入模块中断	OB83
CPU硬件故障(例如, 存储卡已拆除)	OB84
程序出错	OB85
机架故障	OB86
通讯出错	OB87

## 12.2 使用SFC39 “DIS\_IRT”禁用新中断和异步出错的处理

### 描述

通过SFC39 “DIS\_IRT”(禁用中断), 可以禁用新中断和异步出错的处理。也就是说, 如果发生中断, 则CPU操作系统的处理如下:

- **既不**调用中断OB或异步出错OB,
- **也不**触发常规响应处理(如果未编程中断OB或异步出错OB的话)。

如果禁用中断和异步出错, 则这对于所有优先级都有效。只能通过调用SFC40 “EN\_IRT”(参见通过输出参数RET\_VAL判断出错)部分0)或者通过暖重启或冷重启重新取消“DIS\_IRT”的影响。

发生中断和异步出错时, 操作系统是否将其写入诊断缓冲区取决于为MODE选择的输入参数设置。

### 注意

请记住, 如果在程序中使用了SFC39 “DIS\_IRT”, 则将丢失发生的所有中断!

参数	声明	数据类型	存储区域	描述
MODE	INPUT	BYTE	I、Q、M、D、L、 常数	指定禁用哪些中断和异步出错。
OB_NR	INPUT	INT	I、Q、M、D、L、 常数	OB编号
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能处于激活状态时出错, 则返回值将包含出错代码。

### MODE

MODE	含义
00	将禁用所有新发生的中断和异步出错。(未禁用同步出错。)将值0赋值给OB_NR参数。继续在诊断缓冲区中 输入条目。

MODE	含义
01	禁用所有新发生的、属于指定中断等级的事件。通过为中断等级指定最小可能的OB编号来标识该中断等级(即使CPU中不存在此OB)。 实例： <ul style="list-style-type: none"> <li>• 如果要禁用所有监视狗中断，请在OB_NR中指定30(即使OB30不是CPU的第一个监视狗OB。)</li> <li>• 如果要禁用所有硬件中断，请在OB_NR中指定40。</li> </ul> 继续在诊断缓冲区中输入条目。
01	禁用所有新发生的、属于指定中断等级的事件。通过如下指定来标识中断等级： <ul style="list-style-type: none"> <li>• 时钟中断： 10</li> <li>• 延时中断： 20</li> <li>• 周期性中断： 30</li> <li>• 硬件中断： 40</li> <li>• 用于DPV1的中断： 50</li> <li>• 多值计算中断： 60</li> <li>• 冗余出错中断： 70</li> <li>• 异步出错中断： 80</li> </ul> 继续在诊断缓冲区中输入条目。
02	禁用所有新发生的指定中断。使用OB编号 指定中断。继续在诊断缓冲区中输入条目。
80	禁用所有新发生的中断和异步出错，不再将它们输入到诊断缓冲区中。操作系统在诊断缓冲区中输入事件W#16#5380。
81	禁用所有新发生的、属于指定中断等级的事件，且不再将它们输入到诊断缓冲区中。操作系统在诊断缓冲区中输入事件W#16#5380。
82	禁用所有新发生的、属于指定中断的事件，且不再将它们输入到诊断缓冲区中。操作系统在诊断缓冲区中输入事件W#16#5380。

## 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	输入参数OB_NR包含非法值。
8091	输入参数MODE包含非法值。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 12.3 使用SFC40 “EN\_IRT”启用新中断和异步出错的处理

### 描述

使用SFC40 “EN\_IRT” (启用中断)，可以启用先前通过SFC39 “DIS\_IRT”禁用的新中断和异步出错的处理。也就是说，如果发生中断事件，则CPU操作系统将以下列方式中的一种进行处理：

- 调用中断OB或异步出错OB。
- 如果未编程中断OB或异步出错OB，则触发标准响应处理。

参数	声明	数据类型	存储区域	描述
MODE	INPUT	BYTE	I、Q、M、D、L、 常数	指定将启用哪些中断和异步出错。
OB_NR	INPUT	INT	I、Q、M、D、L、 常数	OB编号
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在功能处于激活状态时出错，则返回值将包含出错代码。

### MODE

MODE (B#16#...)	含义
0	启用所有新发生的中断和异步出错。
1	启用所有新发生的、属于指定中断等级的事件。通过如下指定来标识中断等级： <ul style="list-style-type: none"> <li>• 时钟中断： 10</li> <li>• 延时中断： 20</li> <li>• 周期性中断： 30</li> <li>• 硬件中断： 40</li> <li>• 用于DPV1的中断： 50</li> <li>• 多值计算中断： 60</li> <li>• 冗余出错中断： 70</li> <li>• 异步出错中断： 80</li> </ul>
2	启用所指定中断的所有新发生的事件。使用OB编号 指定中断。

## 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	输入参数OB_NR包含非法值。
8091	输入参数MODE包含非法值。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。



## 12.4 使用SFC41 “DIS\_AIRT”延迟更高优先级中断和异步出错的 处理

### 描述

使用SFC41 “DIS\_AIRT”(禁用报警中断)，可以延迟优先级高于当前OB的中断OB和异步出错OB的处理。可以在一个OB中多次调用SFC41。SFC41调用将由操作系统进行计数。这些调用中的每一个都将保持有效，直至明确地通过SFC42 “EN\_AIRT”调用再次取消或直至完全处理完当前OB为止。

一旦重新启用后，只要通过SFC42 “EN\_AIRT”重新启用或当前OB已经执行完毕，则SFC41有效期间发生的中断和异步出错就将被处理。

参数	声明	数据类型	存储区域	描述
RET_VAL	OUTPUT	INT	I、Q、M、D、L	延迟数(= SFC41调用数)

### 返回值

下表列出通过RET\_VAL参数输出的SFC41返回值。

返回值	描述
n	“n”表示处理被禁用的次数，即SFC41调用的次数(只有n = 0时，才会重新启用中断处理；参见使用SFC42 “EN_AIRT”启用更高优先级中断和异步出错的处理)。

## 12.5 使用SFC42 “EN\_AIRT”启用处理较高优先级的中断和异步出错

### 描述

通过SFC42 “EN\_AIRT”(启用报警中断), 可以启用先前通过SFC41 “DIS\_AIRT”禁用的更高优先级中断和异步出错的处理。必须通过SFC41调用来取消每个SFC42调用。

### 实例

例如, 如果通过五次SFC41调用禁用了五次中断, 则必须通过五次SFC42调用来取消这些调用。

参数	声明	数据类型	存储区域	描述
RET_VAL	OUTPUT	INT	I、Q、M、D、L	完成SFC42或发生出错消息时仍编程的延迟数。

### 返回值和出错信息

请参考使用输出参数RET\_VAL评估出错

通过输出参数RET\_VAL判断出错“使用输出参数RET\_VAL评估出错”一章介绍了如何评估RET\_VAL参数的出错信息。此章还包含了SFC的常规出错信息。下表包含了可以通过RET\_VAL参数输出的、SFC42特定的所有出错信息。

返回值和出错信息	描述
N	“n”表示尚未通过SFC41调用取消的SFC42调用次数(只有当“n” = 0时, 才会重新启用中断处理)。
W#16#8080	虽然已启用中断处理, 但已再次调用了此功能。

## 13 用于诊断的SFC

### 13.1 系统诊断

CPU将保留有关可编程逻辑控制器状态的内部数据。通过系统诊断功能，可以读取最重要的数据。可以使用STEP 7在编程设备上显示某些数据。

还可以使用SFC “RD\_SINFO”和“RDSYSST”访问系统诊断所需的数据。

### 13.2 使用SFC6 “RD\_SINFO”读取OB启动信息

#### 描述

通过SFC6 “RD\_SINFO”(读取启动信息)，可以读取有关下列各项的启动信息：

- 尚未完全执行的、被调用的最近一个OB，  
和
- 将要启动的最近一个启动OB。

任一情况下均没有时间标志。如果在OB100或OB101或OB102中调用，将返回两条相同的启动信息消息。

参数	声明	数据类型	存储区域	描述
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
TOP_SI	OUTPUT	STRUCT	D、L	当前OB的启动信息
START_UP_SI	OUTPUT	STRUCT	D、L	上次启动的启动OB的启动信息

## TOP\_SI和START\_UP\_SI

输出参数TOP\_SI和START\_UP\_SI是有着相同单元的两个结构(参见下表)。

结构单元	数据类型	描述
EV_CLASS	BYTE	<ul style="list-style-type: none"> <li>0到3位: 事件标识符</li> <li>位4至7: 事件等级</li> </ul>
EV_NUM	BYTE	事件编号
PRIORITY	BYTE	优先级编号
NUM	BYTE	OB编号
TYP2_3	BYTE	数据标识符2_3: 标识在ZI2_3中输入的信息
TYP1	BYTE	数据标识符1: 标识在ZI1中输入的信息
ZI1	WORD	附加信息1
ZI2_3	DWORD	附加信息2_3

### 注意

表格中列出的结构单元和OB的临时变量有相同内容。

但请注意, 各个OB的临时变量可以有不同的名称和不同的数据类型。同时, 请注意每个OB的调用接口包括OB请求的日期和时间等附加信息。

EV\_CLASS结构单元的4到7位包含事件等级。此处可以为下列值:

- 1: 来自标准OB的启动事件
- 2: 来自同步出错OB的启动事件
- 3: 来自异步出错OB的启动事件

PRIORITY结构单元提供属于当前OB的优先级(请参见章节通过输出参数RET\_VAL判断出错)。

除这两个单元外, NUM也是相关的。NUM包含当前OB或上次启动的启动OB的编号。

## 实例

上次启用且未完全处理的OB用作OB80。上次启动的启动OB用作OB100。

下表显示了SFC6 “RD\_SINFO”的参数TOP\_SI的结构单元及OB80的各局部变量的分配。

TOP_SI		OB80	
结构单元	数据类型	局部变量	数据类型
EV_CLASS	BYTE	OB80_EV_CLASS	BYTE
EV_NUM	BYTE	OB80_FLT_ID	BYTE
PRIORITY	BYTE	OB80_PRIORITY	BYTE
NUM	BYTE	OB80_OB_NUMBR	BYTE
TYP2_3	BYTE	OB80_RESERVED_1	BYTE
TYP1	BYTE	OB80_RESERVED_2	BYTE
ZI1	WORD	OB80_ERROR_INFO	WORD
ZI2_3	DWORD	OB80_ERR_EV_CLASS	BYTE
		OB80_ERR_EV_NUM	BYTE
		OB80_OB_PRIORITY	BYTE
		OB80_OB_NUM	BYTE

下表显示了SFC6 “RD\_SINFO”的参数START\_UP\_SI的结构单元及OB100的各局部变量的分配。

START_UP_SI		OB100	
结构单元	数据类型	局部变量	数据类型
EV_CLASS	BYTE	OB100_EV_CLASS	BYTE
EV_NUM	BYTE	OB100_STRTUP	BYTE
PRIORITY	BYTE	OB100_PRIORITY	BYTE
NUM	BYTE	OB100_OB_NUMBR	BYTE
TYP2_3	BYTE	OB100_RESERVED_1	BYTE
TYP1	BYTE	OB100_RESERVED_2	BYTE
ZI1	WORD	OB100_STOP	WORD
ZI2_3	DWORD	OB100_STRT_INFO	DWORD

## 出错信息

SFC6 “RD\_SINFO”不提供任何特定出错信息，而只提供常规出错信息。在标题为“SFC的常规参数”部分的使用输出参数RET\_VAL评估出错中详细介绍了常见出错代码及如何评估这些出错。

## 13.3 使用SFC51 “RDSYSST”读取系统状态列表或部分列表

### 描述

通过系统功能SFC51 “RDSYSST” (读取系统状态)，可以读取系统状态列表或部分系统状态列表。

调用SFC51时，通过将值“1”赋给输入参数REQ来启动读取。如果可以立即读取系统状态，则SFC将在BUSY输出参数中返回值0。如果BUSY包含值1，则尚未完成读取功能。

### 注意

如果使用SSL-ID W#16#00B1或W#16#00B2或W#16#00B3，在诊断中断OB中调用SFC51 “RDSYSST”，并访问启动该诊断中断的模块，则将会立即读取系统状态。

### 系统资源

如果以较短间隔一个接一个地启动几个异步读取功能(使用SSL\_ID W#16#00B4、W#16#4C91、W#16#4092、W#16#4292和W#16#4692及可能的W#16#00B1和W#16#00B3启动的作业)，则操作系统将确保执行所有读取作业且彼此不互相干扰。如果达到系统资源的极限，则将在RET\_VAL中加以指示。可以通过重复执行作业来纠正这种临时出错状况。

最多可“同时”激活多少SFC51作业取决于CPU。可以在/70/和/101/中找到此信息。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1: 启动处理
SSL_ID	INPUT	WORD	I、Q、M、D、L、 常数	将要读取的系统状态列表或部分列表(部分列表将在第0章中说明)的SSL-ID。
INDEX	INPUT	WORD	I、Q、M、D、L、 常数	部分列表中对象的类型或编号。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果执行SFC时出错，则RET_VAL参数将包含出错代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	TRUE: 尚未完成读取。
SSL_HEADER	OUTPUT	STRUCT	D、L	参见下文。

参数	声明	数据类型	存储区域	描述
DR	OUTPUT	ANY	I、Q、M、L、D	SSL列表读取或SSL部分列表读取的目标区域： <ul style="list-style-type: none"> <li>如果仅读取了SSL列表的标题信息，则不能评估DR的值，而只能评估SSL_HEADER的值。</li> <li>否则，LENTHDR和N_DR的乘积将指示已在DR中输入了多少字节。</li> </ul>

## SSL\_HEADER

SSL\_HEADER参数是一个如下定义的结构：

```
SSL_HEADER:  STRUCT
LENTHDR:    WORD
N_DR:       WORD
END_STRUCT
```

LENTHDR是SSL列表或SSL部分列表的数据记录的长度。

- 如果仅读取了SSL列表的标题信息，则N\_DR包含属于它的数据记录数。
- 否则，N\_DR包含传送到目标区域的数据记录数。

## 出错信息

出错代码 (W#16#...)	描述
0000	无错。
0081	结果域过短。(但是，仍然将尽可能多地提供数据记录。SSL标题指示此数值。)
7000	REQ=0的第一次调用：无激活的数据传送；BUSY值为0。
7001	第一次调用，REQ=1：已启动数据传送；BUSY包含值1。
7002	中间调用(与REQ无关)：数据传送已经激活；BUSY值为1。
8081	结果域过短(没有可容纳一个数据记录的足够空间)。
8082	SSL_ID出错或在CPU或SFC中未知。
8083	INDEX出错或不允许。
8085	因为系统中存在问题，信息目前不可用(例如，因为资源不足)。
8086	因为出现系统出错，无法读一个数据记录(总线、模块、操作系统)。
8087	因为模块不存在或不能进行应答，无法读一个数据记录。
8088	因为实际的模块标识符与预期的模块标识符不同，无法读一个数据记录。
8089	因为模块不能进行诊断，无法读一个数据记录。
80A2	DP协议出错(第2层出错)(临时出错)

出错代码 (W#16#...)	描述
80A3	关于用户接口/用户的DP协议出错(临时出错)
80A4	与通讯总线有关的通讯问题(CPU和外部DP接口模块间出错)
80C5	分布式I/O不可用(临时出错)。
80C6	因为优先级中止(重新启动或后台操作)，数据记录传送已停止
80D2	因为模块不能进行诊断，无法读一个数据记录。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## SSL\_ID

### 注意

有关可以通过SFC51 “RDSYSST”读取的部分列表的信息，参见

- I70I(对于S7-300)
- 下表(对于S7-400)。

SSL_ID(W#16#...)	部分列表	INDEX (W#16#...)
	<b>模块标识符</b>	
0111	一个标识数据记录	
	模块的标识	0001
	基本硬件的标识	0006
	基本硬件的标识	0007
	<b>CPU特征</b>	
0012	所有特征	不相关
0112	一个组的特征	
	<b>MC7处理单元</b>	0000
	时间系统	0100
	系统特性	0200
	MC7语言描述	0300
	SFC的可用性	0400
0F12	仅SSL部分列表标题信息	不相关
	<b>用户存储器区域</b>	
0113	指定的存储器区域的一个数据记录	
	工作存储器	0001
	<b>系统区域</b>	
0F14	所有系统区域的数据记录	不相关
0F14	仅SSL部分列表标题信息	不相关
	<b>模块类型</b>	
0015	所有模块类型的数据记录	不相关



SSL_ID(W#16#...)	部分列表	INDEX (W#16#...)
	<b>模块LED的状态(不能从所有CPU中读取, 参见102)</b>	
0019	所有LED的状态	不相关
0F19	仅SSL部分列表标题信息	不相关
	<b>一个组件的标识</b>	
001C	所有组件的标识	不相关
011C	一个组件的标识	
	自动化系统的名称	0001
	模块名称	0002
	模块的系统标识符	0003
	版权条目	0004
	模块的序列号	0005
	为操作系统保留	0006
	模块类型名称	0007
	存储卡的序列号	0008
	CPU模块的制造商和配置文件	0009
	模块的位置指定	000A
	模块的位置标识	000B
0F1C	仅SSL部分列表标题信息	不相关
	<b>过程映像分区和CPU的分配</b>	
0025	将所有过程映像分区分配到OB	不相关
0125	过程映像分区到相应OB的分配	过程映像分区编号。
0225	OB到相应过程映像分区的分配	OB编号。
0F25	仅SSL部分列表标题的信息	不相关
	<b>通讯状态数据</b>	
0132	一个通讯单元的状态数据	
	诊断	0005
	时间系统	0008
	运行时间定时器(16位)	000B
	运行时间定时器(32位)	000C
0232	一个通讯单元的状态数据	
	CPU保护级别和操作员控制设置	0004
	<b>H CPU组信息</b>	
0071	有关H系统当前状态的信息	不相关
0F71	仅SSL部分列表标题信息	不相关
	<b>模块LED的状态(不能从所有CPU中读取, 参见102)</b>	
0174	LED的状态	LED标识符
	<b>在H系统中切换的DP从站</b>	
0C75	H系统与切换的DP从站间的通讯状态	DP从站接口的诊断地址

SSL_ID(W#16#...)	部分列表	INDEX (W#16#...)
	<b>DP主站系统信息</b>	
0090	CPU已知的DP主站系统信息	0000
0190	有关DP主站系统的信息	DP主站系统标识号
0F90	仅SSL部分列表标题信息	0000
	<b>模块状态信息(最多提供27个数据记录)</b>	
0091	所有插入的模块/子模块的状态信息	不相关
0191	所有带不正确类型标识符的、未取消激活的模块/机架的模块状态信息	不相关
0291	所有存在故障且未取消激活的模块的模块状态信息	不相关
0391	所有无法获得的模块的模块状态信息	不相关
0591	主机模块的所有子模块的模块状态信息	不相关
0991	指定机架中主机模块的所有子模块的模块状态信息	机架或DP主站系统标识符
0A91	所有DP主站系统的模块状态信息	不相关
0C91	中央组态中的或连接到集成DP通讯处理器的模块的模块状态信息	逻辑基址
4C91	连接到外部DP通讯处理器的模块的模块状态信息	逻辑基址
0D91	指定机架/DP站中的所有模块的模块状态信息	机架或DP主站系统标识符, 或DP主站系统标识符和站号
0E91	所有已分配模块的模块状态信息	不相关
	<b>机架/站的状态信息</b>	
0092	中央组态的机架/DP主站系统的站的预期状态	0/DP主站系统 标识号
4092	连接到外部DP接口的DP主站系统的站的预期状态	DP主站系统标识符
0192	通过集成DP接口连接的DP主站系统中的站的激活状态	DP主站系统标识符
0292	中央组态中的机架的当前状态/DP主站系统的站的当前状态	0/DP主站系统标识符
0392	至少一块电池出现故障后, CPU的机架/模块机架中备用电池的状态	0
0492	CPU的所有机架/模块机架的总电池备用状态	0
0592	CPU的所有机架/模块机架的24-V电源的状态	0
4292	通过外部DP接口模块连接的DP主站系统的站的当前状态	DP主站系统标识符
0692	中央组态中扩展机架的诊断状态/通过集成DP接口模块连接的DP主站系统中的站的诊断状态	0/DP主站系统标识符

SSL_ID(W#16#...)	部分列表	INDEX (W#16#...)
4692	中央组态中的扩展机架/通过集成DP接口连接的DP主站系统的站的	DP主站系统标识符
	<b>扩充的DP主站系统信息</b>	
0195	有关DP主站系统的扩充信息	DP主站系统标识符
0F95	仅SSL部分列表标题的信息	0000
	<b>诊断缓冲区(最多提供21个数据记录)</b>	
00A0	当前活动工作模式中可以提供的所有条目	不相关
01A0	最新条目, 编号在索引中指定	数量
0FA0	仅SSL部分列表标题信息	不相关
	<b>关于模块的诊断数据</b>	
00B1	一个模块的前四个诊断字节(数据记录0)	逻辑基址
00B2	一个模块的所有诊断数据(_ 220字节, 数据记录1) (无DP模块)	机架、插槽
00B3	一个模块的所有诊断数据(_ 220字节, 数据记录1)	逻辑基址
00B4	DP从站的诊断数据	组态的诊断地址

## 13.4 使用SFC52 “WR\_USMSG”将自定义诊断事件写入诊断缓冲区

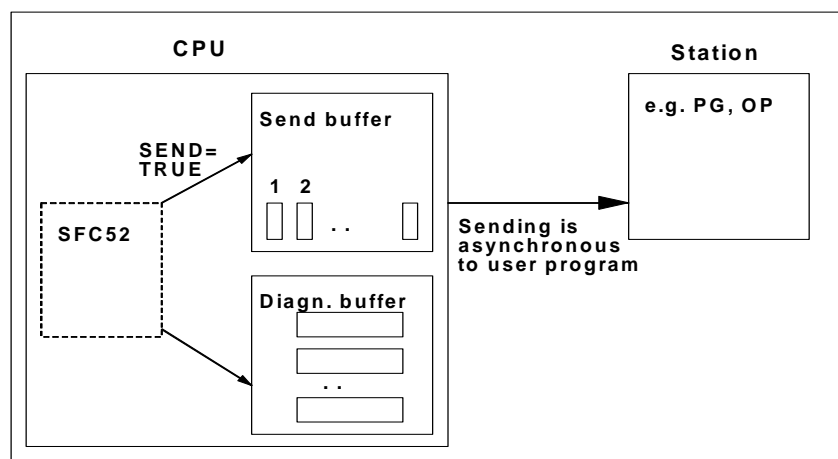
### 描述

通过SFC52 “WR\_USMSG” (将用户元素写入诊断缓冲区), 可以将自定义诊断事件写入诊断缓冲区。还可以将相应的诊断信息发送到为此目的而登录的所有站(通过设置输入参数SEND = TRUE)。如果出错, 则输出参数RET\_VAL将提供出错信息。

### 发送自定义诊断消息

SFC52将自定义诊断事件写入诊断缓冲区。然后, 还可以将相应的诊断消息发送到为此目的而登录的任何站(通过设置输入参数SEND = TRUE)。接下来, 自定义诊断消息将写入发送缓冲区并自动发送到登录的站。

可以检查当前是否能够发送自定义诊断消息。为此, 可以通过参数SSL\_ID = W#16#0132和INDEX = W#16#0005调用SFC51 “RDSYSST”。所获取的数据记录的第四个字指示当前能(1)否(0)发送用户元素。



### 发送缓冲区满

只有在发送缓冲区未滿时, 才能在发送缓冲区中输入诊断消息。可以在发送缓冲区中输入的条目数取决于所用CPU的类型。

如果发送缓冲区已滿, 则:

- 诊断事件仍输入诊断缓冲区中,
- 参数RET\_VAL指示发送缓冲区已滿(RET\_VAL = W#16#8092)。

## 未登录站

如果要发送自定义诊断消息(SEND = TRUE)，而未登录任何站，则

- 自定义诊断事件将输入诊断缓冲区中，
- 参数RET\_VAL指示未登录任何站(RET\_VAL = W#16#8091或W#16#8091。如果使用较早版本的CPU，将显示值W#16#8091)。

## 常规结构

诊断缓冲区中元素的内部结构如下：

字节	内容
1和2	事件标识符
3	优先级等级
4	OB编号
5和6	保留
7和8	附加信息1
9、10、11和12	附加信息2
13至20	时间标志

## 事件标识符

每个事件分配一个事件标识符。

## 附加信息

这是有关事件的附加信息。每个事件的附加信息可以不同。创建诊断事件时，可以自己决定这些条目的内容。

发送自定义诊断消息时，可在(事件标识符指定的)消息文本中将附加信息合并为关联值。

## 时间标志

时间标志的类型为Date\_and\_Time。

参数	声明	数据类型	存储区域	描述
SEND	INPUT	BOOL	I、Q、M、D、L、 常数	启用发送自定义诊断消息到所有已登录的站
EVENTN	INPUT	WORD	I、Q、M、D、L、 常数	事件标识符 - 由您分配事件标识符。 而不是由 消息服务器进行分配。
INFO1	INPUT	ANY	I、Q、M、D、L	1个字长的附加信息
INFO2	INPUT	ANY	I、Q、M、D、L	2个字长的附加信息
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息

## SEND

如果SEND = TRUE，则自定义诊断消息将发送到所有已登录的站。只有当站已经登录且发送缓冲区未滿时，才会发送消息。元素的发送将与用户程序异步。

## EVENTN

EVENTN参数包含用户事件的事件标识符。可输入类型为W#16#8xyz、W#16#9xyz、W#16#Axyz、W#16#Bxyz的事件标识符。

W#16#8xyz和W#16#9xyz格式的标识符属于预定义事件，W#16#Axyz和W#16#Bxyz格式的标识符属于自由定义的事件。

进入状态的事件由x = 1指示，离开状态的事件由x = 0指示。对于A和B等级的事件，“yz”为在消息组态中分配给消息的十六进制格式的消息编号。

事件标识符的结构在第26.1部分介绍。

## INFO1

INFO1参数包含一个字长的信息。INFO1允许使用下列数据类型：

- WORD
- INT
- ARRAY [0 - 1] OF CHAR

可以在消息文本中将参数INFO1合并为关联值，从而向该消息添加最新信息。

**INFO2**

INFO2参数包含两个字长的信息。INFO2允许使用下列数据类型：

- DWORD
- DINT
- REAL
- TIME
- ARRAY [0 - 3] OF CHAR

可以在消息文本中将参数INFO2合并为关联值，从而向该消息添加最新信息。

**出错信息**

出错代码(W#16#...)	说明
0000	无故障
0091	未登录任何站(诊断事件输入诊断缓冲区中)
8083	INFO1的数据类型不允许使用
8084	INFO2的数据类型不允许使用
8085	EVENTN不允许使用
8086	INFO1的长度不允许
8087	INFO2的长度不允许
8091	(只有当使用较早版本的CPU时，才会显示此出错代码。)未登录任何站(诊断事件输入诊断缓冲区中)。
8092	目前不可以发送，发送缓冲区已满(诊断事件输入诊断缓冲区中)。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 13.5 使用SFC78 “OB\_RT”确定OB程序运行时间

### 描述

通过SFC78 “OB\_RT”，可以确定不同时间段各OB的运行时间。

### 注意

对于所需OB，无论它当前是否加载，SFC78都将返回上一次为它记录的时间值。SFC78数据不会被删除或覆盖，而是在重启动(暖重启)后被复位。

### 参数

参数	声明	数据类型	存储区域	含义
OB_NR	INPUT	INT	I、Q、M、D、L	要查询上次计算的时间的OB。 有效的编号为CPU的OB组态中的所有编号(OB 121和OB 122除外)。同步出错处理时间包括在处理导致该出错的OB所需的时间中。 如果声明OB 121和122， 或者声明未在CPU中组态的OB，将导致出错消息。 OB_NR=0时，将传送已调用SFC的OB的数据。在OB 121或OB 122中调用SFC78且OB_NR=0时，将输出触发OB的中断的所有时间数据，包括OB 12x中的时间数据。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果处理此功能时出错，则返回值将包含出错代码。否则，RET_VAL将包含这些数据对应的OB的编号。
PRIO	OUTPUT	INT	I、Q、M、D、L	所查询OB的优先级在PRIO中输出



参数	声明	数据类型	存储区域	含义
LAST_RT	OUTPUT	DINT	I、Q、M、D、L	<p>指定OB的最近一次执行的运行时间(微秒)。 如果您请求其运行时间的OB当前正在处理： 在所需OB的当前执行期间，第一次调用SFC78后，LAST_RT将输出该OB的上次总执行运行时间。 在所需OB的当前执行期间，每次随后调用SFC78时，Last_RT将</p> <ul style="list-style-type: none"> <li>• 指示DW#16#FFFF FFFF(如果所需OB已在OB_NR=0时调用SFC78)。</li> <li>• 指示所需OB的上一次执行的运行时间(如果此OB未在OB_NR=0时调用SFC78)。</li> </ul> <p>注意：LAST_RT中不包括具有更高优先级的OB导致的中断时间。</p>
LAST_ET	OUTPUT	DINT	I、Q、M、D、L	<p>指定OB从调用到执行结束之间的时间间隔(微秒)，即上一次完成的指定OB的处理所使用的时间。 如果您请求其运行时间的OB当前正在处理： 在所需OB处理期间，第一次调用SFC78后，LAST_ET指示从上次完成OB调用到指定OB执行结束之间的时间间隔。 在所需OB的当前执行期间，每次随后调用SFC78时，Last_ET将</p> <ul style="list-style-type: none"> <li>• 指示DW#16#FFFF FFFF(如果所需OB已在OB_NR=0时调用SFC78)。</li> <li>• 指示从上次完成OB调用到指定OB执行结束之间的时间间隔(如果OB_NR=0时此OB尚未调用SFC78的话)。</li> </ul> <p>注意：LAST_RT中包括具有更高优先级的OB导致的中断时间。</p>

参数	声明	数据类型	存储区域	含义
CUR_T	OUTPUT	DINT	I、Q、M、D、L	指定的、当前处理的OB的OB请求的时间(以微秒为单位的相对时间值)。如果指定的OB不是正被处理, 则CUR_T = 0。 注意: 系统时间是从0到2.147.483.647(微秒)计数的计数器。计数器溢出后将从0重新开始计数。
CUR_RT	OUTPUT	DINT	I、Q、M、D、L	指定OB已用的执行时间(微秒)。如果指定的OB不是正被处理或尚未处理, 则CUR_RT = 0。处理后, 运行时间数据输入到LAST_RT中, CUR_RT将设置为零。 注意: CUR_RT中不包括具有更高优先级的OB导致的中断时间。
CUR_ET	OUTPUT	DINT	I、Q、M、D、L	从调用当前处理的指定OB起所经过的时间(以微秒为单位的运行时间)。如果指定的OB不是正被处理, 则CUR_ET = 0。处理后, 运行时间数据将输入到LAST_RT中, CUR_ET将设置为零。 注意: CUR_ET中包括具有更高优先级的OB导致的中断时间。
NEXT_ET	OUTPUT	DINT	I、Q、M、D、L	如果在完成当前请求前, 队列中还有更多指定OB的调用, 则NEXT_ET指示实际时间与下一请求的执行时间之间的时间间隔(微秒)。如果对此OB来讲, 除当前正排队或正处理的启动事件外, 不存在其它启动事件, 则NEXT_ET = 0。 WinLC RTX不使用此参数。 注意: NEXT_ET中包括具有更高优先级的OB导致的中断时间。

这些时间还包括任何嵌套的同步出错中断(OB 121、OB 122)处理的运行时间。

### 注意

如果在OB\_NR中声明在CPU的动态项目数据中存在的OB编号, 而OS并没有调用相应的OB, 或者未将其下载到CPU中, 则RET\_VAL将包含指定的OB编号, PRIO将包含指定OB的组态优先级(缺省值, 如果需要的话), 而LAST\_RT将包含值DW#16#FFFF FFFF。

## 出错信息

事件类别出错代码	说明
1至102	要传送信息到其中的OB的编号。
W#16#8080	OB_NR参数包含非法值。
W#16#8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 13.6 使用SFC87 “C\_DIAG”诊断当前连接状态

### 描述

可以使用SFC87 “C\_DIAG”诊断所有S7连接和所有容错S7连接(或其局部连接)的当前状态。

恰当地评估这些连接数据能够发现S7连接故障及当前S7连接故障，并在合适的时机报告给可视化操作系统。监视的连接可以是自动化系统间的连接，也可以是自动化系统与可视化操作系统间的连接。

---

### 注意

CPU工作状态的改变((RUN -> STOP -> RUN)不会影响所组态的连接的状态。

例外：H站从“冗余”系统状态变为“停止”系统状态时，与待机CPU的所有容错连接中的局部连接将断开。

另一方面，在掉电后，所有组态的连接都将重新建立，这将改变连接状态。

启动期间或启动后第一次调用SFC87，连接信息将有所不同，这取决于CPU的上一工作模式是STOP还是POWER OFF。

---

### 工作模式

SFC87 “C\_DIAG”是异步SFC操作，也就是说，处理可以扩展到跨多个SFC调用。

通过使用REQ=1调用SFC87来启动作业。

如果可以立即执行作业，SFC将在输出参数BUSY中返回值0。如果BUSY的值为1，则作业仍处于激活状态。

### 何时调用SFC87?

要识别S7连接的故障和当前S7连接，可以在已启动的周期性中断OB中调用SFC87，例如，由操作系统每10秒调用一次。

因为连接状态通常不发生改变，所以仅当它们自上次调用(MODE=B#16#02时的调用，参见下文)后发生改变时，才应通过这些循环调用将连接数据复制到用户程序中。

## 如何调用SFC87?

SFC87 “C\_DIAG”提供了四种可能的操作模式这在下表中给予介绍。

MODE (B#16#...)	SFC将连接数据复制到用户程序	SFC将确认信息传送到操作系统
00	否	是
01	是	是
02	<ul style="list-style-type: none"> <li>• 是, 如果连接数据已改变</li> <li>• 否, 如果连接数据未改变</li> </ul>	是
03	是	否

通过将确认消息传送到操作系统来确认自上次调用SFC87SP;  
(通过MODE=B#16#00、01或02)后连接数据的状态改变。

### 注意

如果在处于“有条件复制”模式(MODE=B#16#02)下的周期性中断OB中操作SFC87, 则必须确保CPU冷重启后, 目标区域中不包含初始化值。可以通过在“带确认的无条件复制”模式(MODE=B#16#01)下调用一次SFC87, 以便在OB 102中实现这一点。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	要激活的控制参数请求 REQ = 1: 初始化该作业(如果尚未启动)

参数	声明	数据类型	存储区域	描述
MODE	INPUT	BYTE	I、Q、M、D、L、 常数	<p>作业指定可能的值：</p> <ul style="list-style-type: none"> <li>• <b>B#16#00</b>: SFC不复制连接数据，仅将确认信息传送到操作系统。</li> <li>• <b>B#16#01</b>: 无论状态如何改变，SFC均将所有连接数据复制到用户程序中，并将确认信息传送到操作系统。</li> <li>• <b>B#16#02</b>: 如果连接数据已改变，则SFC会将其复制到用户程序中。如未改变，则不复制。在两种情况下，SFC均将确认信息传送到操作系统中。</li> <li>• <b>B#16#03</b>: SFC将连接数据复制到用户程序中(无论改变后的状态如何)。而不将确认消息传送到操作系统中。</li> </ul>
RET_VAL	OUTPUT	INT	I、Q、M、D、L	返回值(出错代码或作业状态)
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 作业未完成。
N_CON	OUTPUT	INT	I、Q、M、D、L	.DIS_PCON或.DIS_CON的值为TRUE的CON_ARR中最后一个结构的索引。因此，在用户程序中只需检查CON_ARR的第一个N_CON元素。 注意：注意：CON_ARR域中的第一个结构具有索引1。
CON_ARR	OUTPUT	ANY	I、Q、M、D、L	<p>用于接收的连接数据的目标区域。 只允许使用数据类型BYTE。 将给每个连接分配一个结构。 选择一个目标区域大小，使之在连接了CPU允许的最大数目的连接时仍然可以接收所有结构。</p>

## 组织目标区域CON\_ARR

读取目标区域是一个结构域。将给每个连接分配一个结构。

最初，该域无需填入有效条目，域的两个有效条目间可以包含无效条目。

连接不按连接引用排序。

### 注意

如果将连接数据从操作系统复制到所选择的目标区域，则可以确保连接的日期一致性

## 结构组织

参数	数据类型	描述
CON_ID	WORD	在NETPRO中为此连接分配的连接引用 W#16#FFFF：无效指定，也就是说，尚未组态连接。 如果设置CON_ARR[i].DIS_PCON或CON_ARR[i].DIS_CON (参见下文)，则自上次调用SFC87后已重新组态或删除了此连接。
STAT_CON	BYTE	S7连接或冗余S7连接的当前状态 可能的值： <ul style="list-style-type: none"> <li>• B#16#00：S7连接未建立</li> <li>• B#16#10：冗余S7连接未建立</li> <li>• B#16#01：当前正在建立S7连接</li> <li>• B#16#11：当前正在建立冗余S7连接</li> <li>• B#16#02：S7连接已建立</li> <li>• B#16#12：冗余S7连接已建立，但不是冗余状态</li> <li>• B#16#13：冗余S7连接已建立</li> </ul>
PROD_CON	BYTE	运行连接的局部连接号。 可能的值：0、1、2、3
STBY_CON	BYTE	备用连接的局部连接号(B#16#FF：无备用连接) 可能的值：0、1、2、3 注意：只有冗余S7连接才可以有备用连接。

参数	数据类型	描述
DIS_PCON	BOOL	上次调用SFC后，如果CON_ARR[j].STAT_CON发生W#16#12 -> W#16#13和W#16#13 -> W#16#12跳转，则CON_ARR[j].DIS_PCON将被设为1。所有其它连接状态的变化不会影响CON_ARR[j].DIS_PCON。 注意： <ul style="list-style-type: none"> <li>• 如果MODE=B#16#01和02，则将连接数据复制到目标区域时，将复位与DIS_PCON对应的操作系统位。</li> <li>• MODE=B#16#03时，与DIS_PCON对应的操作系统位将保持不变。</li> </ul>
RES0	BYTE	保留(B#16#00)
RES1	BYTE	保留(B#16#00)

### 出错信息

出错代码 (W#16#...)	描述
0000	<ul style="list-style-type: none"> <li>• MODE=B#16#00、01或02：自上次调用后无连接状态变化(结构元素STAT_CON)。已执行调用，未出错。</li> <li>• MODE=B#16#03：已执行复制过程，未出错。</li> </ul>
0001	<ul style="list-style-type: none"> <li>• MODE=B#16#00、01或02：自上次调用后至少有一个连接发生连接状态变化(结构元素STAT_CON)。已执行作业，未出错。</li> <li>• MODE=B#16#03：RET_VAL W#16#0001不可能。</li> </ul>
7000	REQ=0的首次调用。不能处理在MODE中指定的作业。BUSY值为0。
7001	REQ=1的首次调用。已初始化MODE中所指定的作业。BUSY值为1
7002	中间调用(与REQ无关)。作业仍在运行。BUSY值为1。
8080	MODE参数中的非法值。
8081	CON_ARR参数中的非法数据类型。
8082	CON_ARR参数中的长度描述过短。SFC未将任何数据复制到目标区域中。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。



## 13.7 使用SFC103 “DP\_TOPOL”识别DP主站系统的总线拓扑

### 描述

调用具有诊断中继器的SFC103 “DP\_TOPOL”，以启动对所选DP主站系统的总线拓扑的识别。调用SFC103时将寻址连接到DP主站系统的所有诊断中继器。

---

### 注意

在某个给定时间，只能识别一个DP主站系统的拓扑。

---

拓扑识别是线路出错时显示详细出错位置的先决条件。安装或更改DP主站系统的物理结构后，请通过SFC103再次识别拓扑。

对物理结构的更改包括：

- 改变电缆长度
- 添加或删除节点或具有中继器功能的组件
- 改变节点地址

每个诊断中继器在拓扑表中输入结果(所有节点的PROFIBUS地址，及其到诊断中继器的绝对距离)。可以在STEP 7下读取这些值。

建议在OB1或监视狗中断OB中调用SFC103。S7-300不允许在一个启动OB中进行此调用。

SFC将诊断中继器返回的出错消息写入输出DPR和DPRI。如果所选DP主站系统中有多个诊断中继器报告出错，则SFC只在DPR和DPRI中输入与第一个生成出错消息的诊断中继器相关的信息。调用SFC13 “DPNRM\_DG”或STEP 7，可以读取完整的诊断信息。如果诊断中继器未报告任何出错，则输出DPR和DPRI的值为NULL。

### 操作方式

SFC103 “DP\_TOPOL”以异步方式工作，也就是说，它的执行需要调用几个SFC。通过REQ = 1调用SFC103，可以启动总线拓扑的识别，通过R=1调用SFC103，可以中止这一过程。

输出参数RET\_VAL和BUSY中指示了作业状态，另请参见异步工作SFC的REQ、RET\_VAL和BUSY的含义。

---

### 注意

识别拓扑可能要花费几分钟的时间。

---

## 作业标识

输入参数DP\_ID确定一个请求。

如果在完成拓扑识别前再次调用SFC103“DP\_TOPOL”，SFC的进一步响应将取决于新请求是否与先前的请求相同：如果参数DP\_ID与尚未完成的作业相对应，则SFC调用将被解释为调用序列的一部分，将在RET\_VAL中输入值W#16#7002。但如果是新作业，则会被CPU拒绝。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1：启动总线拓扑的识别
R	INPUT	BOOL	I、Q、M、D、L、 常数	R=1：中止拓扑识别
DP_ID	INPUT	INT	I、Q、M、D、L、 常数	将要识别的拓扑所属的DP主站系统的DP主站系统标识符
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果处理此功能期间发生出错，则返回值将包含出错代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1：仍在进行总线拓扑识别。
DPR	OUTPUT	BYTE	I、Q、M、D、L	生成出错消息的诊断中继器的PROFIBUS地址
DPRI	OUTPUT	BYTE	I、Q、M、D、L	生成出错消息的诊断中继器的测量段： <ul style="list-style-type: none"> <li>• 位0 = 1：段DP2出错</li> <li>• 位1 = 1：段DP3出错</li> <li>• 位4 = 1：段DP3有临时中断</li> <li>• 位5 = 1：段DP3有永久中断</li> </ul>
<p><b>永久出错：</b>系统检测到阻止成功进行拓扑识别的永久网络出错。可以通过SFC13“DPNRM_DG”或STEP 7获取诊断信息，以查看细节</p> <p><b>临时出错：</b>系统检测到防止成功进行拓扑识别的临时网络出错。这可能是由于不良接触或反复出错造成的。这些中断的特性使得无法确定出错源的准确位置。</p>				

## 出错信息

看到下表中的“真实”出错信息(出错代码W#16#8xyz)时，我们应区分两类事件：

- 临时出错(出错代码W#16#80A2 - 80A4、80C3、80C5)：  
此类出错可能无需用户操作即可消除，这意味着再次调用SFC是有帮助的(如必要，则多次调用)。  
临时出错的实例：所需资源当前正在使用中(W#16#80C3)。
- 永久出错(出错代码W#16#8082、80B0、80B2)：  
此类型的出错不能自行更正。只有在消除此出错后，SFC的新调用才有意义。  
永久出错实例：DP主站/CPU不支持此服务。(W#16#80B0)。

出错代码 (W#16#...)	说明
0000	作业已完成，未出错。
7000	REQ=0的第一次调用。未启动总线拓扑识别。BUSY = 0。
7001	REQ=1的初始调用。已请求总线拓扑识别。BUSY = 1。
7002	中间调用(与REQ不相关)：仍在进行拓扑识别。BUSY = 1。
7010	已请求取消拓扑识别。但没有与指定的DP_ID匹配的正在处理中的作业BUSY = 0。
7011	初始调用，R=1。已启动取消拓扑识别的请求。BUSY = 1。
7012	中间调用：尚未完成拓扑识别。BUSY = 1。
7013	最终调用：已取消拓扑识别。BUSY = 0。
8082	没有以指定DP_ID组态的DP主站系统。
80A2	拓扑识别期间出错；要了解更多详细信息，请参考输出参数DPR和DPRI。
80A3	拓扑识别期间出错：监视狗超时。
80A4	am K总线上的通讯出错
80B0	DP主站/CPU不支持此服务。
80B2	拓扑识别期间出错：在所选DP主站系统中未找到诊断中继器。
80C3	所需资源当前正在使用。可能原因：您启动了第二个拓扑识别周期(在某给定时间只允许一个拓扑识别周期)或H CPU正在执行连接或更新操作。
80C5	DP主站系统当前不可用。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。



## 14 用于更新过程映像和处理位域的SFC和SFB

### 14.1 使用SFC26 “UPDAT\_PI”更新过程映像输入表

#### 描述

通过SFC26 “UPDAT\_PI”(更新过程映像), 可以更新OB1过程映像输入表(=过程映像段0)或通过STEP 7定义的过程映像输入段。

如果您指定在出现I/O访问出错时重复发出信号以更新系统过程映像表, 则SFC26将不断地更新所选的过程映像表。

否则, SFC26将只在系统不更新所选过程映像段时才更新过程映像表, 换句话说, 只在以下情况下进行更新:

- 未将此过程映像段分配给一个中断OB时, 或者
- 选择了过程映像段0, 并在组态中禁用了OB1过程映像段的更新时。

---

#### 注意

通过STEP 7分配给输入过程映像分区的每个逻辑地址不再属于OB 1的过程映像输入表。

通过SFC26更新输入过程映像分区时, 不能也通过SFC126 “SYNC\_PI”对其进行更新。

---

OB1过程映像输入表及分配给某个中断OB的过程映像输入段的更新不受SFC26调用的影响。

参数	声明	数据类型	存储区域	描述
PART	INPUT	BYTE	I、Q、M、D、L、 常数	要更新的过程映像输入段的编号。 最大数值范围(取决于CPU): 0 - 15 (0表示OB1过程映像, n(其中, 1 <n <15) 表示过程映像段n)
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
FLADDR	OUTPUT	WORD	I、Q、M、D、L	出现访问出错时引起出错的第一个字节的地址。

## 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	输入参数PART的值非法。
8091	指定的过程映像段未定义或不在CPU上允许的过程映像表区域中。
8092	系统通过OB更新了过程映像段，而您尚未组态所有I/O访问出错的重复发送信号。SFC26“UPDAT_PI”未更新过程映像
80A0	更新过程中检测到访问出错。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

---

### 注意

对于定义了大于32字节的一致性区域的DP标准从站，如果将SFC26“UPDAT\_PI”用于其过程映像分区，则也可以获得来自SFC14“DPRD\_DAT”的出错代码。

---

## 14.2 使用SFC27 “UPDAT\_PO”更新过程映像输出表

### 描述

通过SFC27 “UPDAT\_PO”(更新过程输出), 可以将OB1过程映像输出表(=过程映像段0)或通过STEP 7定义的过程映像段的信号状态传送到输出模块。

如果已为该部分过程映像指定一致性范围, 则相应数据将一致地传送到相应外围模块。

### 注意

通过STEP 7分配给输出过程映像分区的每个逻辑地址不再属于OB 1的过程映像输出表。

通过SFC27更新输出过程映像分区时, 不能也通过SFC127 “SYNC\_PO”对其进行更新。

OB1过程映像输出表及分配给中断OB的过程映像输出段的传送不受SFC27调用的影响。

参数	声明	数据类型	存储区域	描述
PART	INPUT	BYTE	I、Q、M、D、L、 常数	要更新的过程映像输出段的编号。最大数值范围(取决于CPU): 0 - 15。 (0表示OB1过程映像, n(其中 $1 \leq n \leq 15$ )表示过程映像段n)
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
FLADDR	OUTPUT	WORD	I、Q、M、D、L	出现访问出错时引起出错的第一个字节的地址。

## 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	输入参数PART的值非法。
8091	指定的过程映像段未定义或不在CPU上允许的过程映像区中。
80A0	更新过程中检测到访问出错。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

---

### 注意

对于定义了大于32字节的一致性区域的DP标准从站，如果将SFC27 “UPDAT\_PO”用于其过程映像分区，则也可以获得来自SFC15 “DPWR\_DAT”的出错代码。

---



## 14.3 使用SFC79 “SET”在I/O区域中设置位域

### 描述

调用SFC79 “SET”(设置输出范围)具有以下作用:

- 置位使用参数N和SA选择的外围I/O区域中的位域。
- 同时也置位过程映像输出表中的相应位，而无论它们是否在过程映像段中。

位域必须是分配给过程映像的外围I/O区域的组成部分。

对于所选位域部分，如果没有插入模块，则SFC79仍将尝试置位整个位域。然后在RET\_VAL中返回相应的出错信息。

### 注意

执行SFC79时，总是将全部字节写入I/O区域中。

如果以参数N和SA选择的位域不以字节界限起始，则调用SFC79具有以下作用:

- 位于要传送到外围I/O区域的第一个和最后一个字节中的，并且不属于所选位域的位，包含了过程映像输出表中相应位的值。这可以导致意外的响应，如启动电机或关闭冷却系统。
- 属于所选位域的位按上述说明置位。

如果将0值赋给参数N，则调用SFC79将不会产生任何作用。如果未置位主控继电器，则调用SFC79将不会产生任何作用。

参数	声明	数据类型	存储区域	描述
N	INPUT	INT	I、Q、M、D、L、常数	要置位的位的编号
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
SA	OUTPUT	POINTER	P	指向要置位的第一个位的指针

### 出错信息

评估参数RET\_VAL的出错信息的方法在使用输出参数RET\_VAL评估出错中进行了介绍。此章还包括SFC的常规出错信息。SFC79不通过RET\_VAL参数提供任何特定出错信息。

## 14.4 使用SFC 126 “SYNC\_PI”在同步循环中更新过程映像分区输入表

### 描述

通过SFC 126 “SYNC\_PI”，可以在同步循环中更新过程映像分区输入表。链接到DP循环的用户程序可以使用此SFC，连续、同步地更新过程映像分区中的输入数据。

SFC 126可以被中断，且只能在OB 61、62、63和64中调用。

为确保稳定执行SFC 126，应满足下列条件(仅限S7-400)：

- DP时钟周期 > 约5.0ms
- 从站更新时间 < DP时钟周期 - 4.0ms

---

### 注意

只有当在HW Config中将受影响的过程映像分区分配给相关OB后，才允许在OB 61到64中调用SFC 126 “SYNC\_PI”。通过SFC 126更新过程映像分区输入表时，不能也通过SFC 26 “UPDAT\_PI”对其进行更新。

---



---

### 当心(仅限S7-400)

请避免直接访问(例如使用L PEB命令)您正在使用SFC 123进行处理的过程映像分区区域。

如果忽略了这条规则，在某些情况下，可能会接收不到当前值。

---

## 参数

参数	声明	数据类型	值范围	缺省值	含义
PART	INPUT	BYTE	1至30		要在同步循环中更新的过程映像分区输入表的编号。
RET_VAL	OUTPUT	INT			出错信息
FLADDR	OUTPUT	WORD			出现访问出错时引起出错的第一个字节的地址。

## 出错信息

事件类别出错代码	说明
W#16#8090	参数PART中包含非法值和/或在此OB中不允许更新指定的过程映像分区输入表。未更新过程映像分区输入表。
W#16#8091	指定的过程映像分区尚未定义或不在CPU的可访问过程映像区域中。未更新过程映像分区输入表。
W#16#80A0	更新过程中检测到访问出错。受影响的输入被设置为“0”。
W#16#80A1	更新时间晚于允许的访问时窗。未更新过程映像分区输入表。
W#16#80C1	更新时间早于允许的存取窗口。未更新过程映像分区输入表。
W#16#8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 注意

对于定义了大于32字节的一致性区域的DP标准从站，如果将SFC 126 “SYNC\_PI”用于其过程映像分区，则也可以获得来自SFC 14 “DPRD\_DAT”的出错代码。

## 14.5 使用SFC 127 “ISO\_PO”在同步循环中更新过程映像分区输出表

### 描述

通过SFC 127 “SYNC\_PO”，可以在同步循环中更新过程映像分区输出表。链接到DP循环的用户程序可以使用此SFC同步地更新过程映像分区中的输出数据，并将它们一致地传送到I/O设备。

SFC 127可以被中断，且只能在OB 61、62、63和64中调用。

为确保稳定可靠地执行SFC 127，应满足下列条件(仅限S7-400):

- DP时钟周期 > 大约5.0ms
- 从站更新时间 < DP时钟周期 - 4.0ms

### 注意

只有当在HW Config中将受影响的过程映像分区分配给相关OB后，才允许在OB 61到64中调用SFC 127 “SYNC\_PO”。通过SFC 127更新过程映像分区。通过SFC 127更新输出过程映像分区时，不能也通过SFC 27 “UPDAT\_PO”对其进行更新。



### 当心(仅限S7-400)

避免直接访问(例如，L PEB命令)正在通过SFC 127处理的过程映像分区的区域。如果无视此规则，则写入尝试将失败

### 参数

参数	声明	数据类型	值范围	缺省值	含义
PART	INPUT	BYTE	1至30		要在同步循环中更新的过程映像分区输出表的编号。
RET_VAL	OUTPUT	INT			如果处理此功能时出错，则返回值将包含出错代码。
FLADDR	OUTPUT	WORD			引起出错的第一个字节的地址。

## 出错信息

事件类别出错代码	说明
W#16#0001	一致性警告。过程映像分区表的更新跨两个DP循环。但始终传送一个从站中的数据。
W#16#8090	参数PART中包含非法值和/或在此OB中不允许更新指定的过程映像分区输出表。未将输出传送到I/O设备中。未更改过程映像分区输出表。
W#16#8091	指定的过程映像分区尚未定义或不在CPU的可访问过程映像区域中。未将输出传送到I/O设备中。未更改过程映像分区输出表。
W#16#80A0	更新过程中检测到访问出错。未将输出传送到I/O设备中。未更改过程映像分区输出表。
W#16#80A1	更新时间晚于允许的存取窗口或者DP主站未更新输出数据。未将输出传送到I/O设备中。未更改过程映像分区输出表。
W#16#80C1	更新时间早于允许的存取窗口。未将输出传送到I/O设备中。过程映像分区输出表保持原样。
W#16#8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 注意

对于定义了大于32字节的一致性区域的DP标准从站，如果将SFC 127 “SYNC\_PO”用于其过程映像分区，则也可以获得来自SFC 15 “DPWR\_DAT”的出错代码。

## 14.6 使用SFC80 “RSET”复位I/O区域中的位域

### 描述

调用SFC80 “RSET”(复位输出范围)具有以下作用：

- 复位通过参数N和SA复位选择的外围I/O区域中的位域。
- 同时也复位过程映像输出表中的相应位，而无论它们是否在过程映像段中。

位域必须位于分配给过程映像的外围I/O区域中。

对于所选位域部分，如果没有插入模块，则SFC80仍将尝试复位整个位域。然后在RET\_VAL中返回相应的出错信息。

### 注意

执行SFC80时，总是将全部字节写入外围I/O区域中。

如果以参数N和SA选择的位域不是以字节界限起始，则调用SFC80具有以下作用：

- 要传送到外围I/O区域的第一个和最后一个字节中的，并且不属于所选位域中的位，包含了过程映像输出表中相应位的值。这可以导致意外的响应，如启动电机或关闭冷却系统。
- 属于所选位域的位按上述说明置位。

如果将0值赋给参数N，则调用SFC80将不会产生任何作用。如果未设置主控继电器，则调用SFC80将不会产生任何作用。

参数	声明	数据类型	存储区域	描述
N	INPUT	INT	I、Q、M、D、L、常数	要复位的位的编号
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
SA	OUTPUT	POINTER	P	指向要复位的第一个位的指针

### 出错信息

评估参数RET\_VAL的出错信息的方法在使用输出参数RET\_VAL评估出错中进行了介绍。此章还包括SFC的常规出错信息。SFC80不通过RET\_VAL参数提供任何特定出错信息。

## 14.7 使用SFB 32 “DRUM”实现一个操作序列

### 描述

SFB 32 “DRUM”可以实现一个最多16步的操作序列。通过参数DSP指定第一步的编号，通过参数LST\_STEP指定最后一步的编号。

在每个操作步骤中，将写入所有16个输出位OUT0到OUT15和输出参数OUT\_WORD (在其中将输出位汇集在一起)。将用指定的OUT\_VAL数组的相应位或前一步骤的相应输出位的值为输出位赋值。赋哪个值取决于S\_MASK参数中屏蔽位的设置(参见下表)。

---

### 注意

屏蔽位的缺省值为0。如果要更改一个或多个屏蔽位的当前设置，可以在背景数据块中实施改动。

---

当与先前的SFB调用相比，JOG输入出现上升沿时，SFB 32 “DRUM”将切换到下一步骤。如果SFB已到达最后一步，则JOG的上升沿将置位变量Q和EOD；DCC的值为0；SFB将停留在最后的步骤中，直至在RESET输入上设置1。

也可以通过参数赋值，使系统根据时间切换到下一步。为此，必须将DRUM\_EN参数设置为1。然后，操作序列在下列情况下切换到下一步骤：

- 用于当前步骤的事件位EVENT被置位，和
- 为当前步骤设定的时间已过。

此时间是对当前步骤有效的DTBP时间基准和时间因数(来自S\_PRESET数组)的乘积

---

### 注意

只有当置位相应的事件位EVENT后，当前步骤(DCC)中剩余的执行时间才会减少。

---

如果在调用SFB时在RESET输入端设置1，则操作序列将转到分配给DSP输入的步骤。

### 注意

如果将DRUM\_EN设置为1，则可以实现以下特殊情况：

- 完全根据时间启用各步骤(通过选择EVENT = 1，其中 $DSP \leq i \leq LST\_STEP$ 。
- 完全根据事件启用各步骤(通过将DTBP的值设置为0，使用事件位EVENT来选择)。

也可以通过JOG输入，随时(即使DRUM\_EN=1)移动到操作序列中的下一个步骤。

第一次调用块时，必须在RESET输入中设置1。

操作序列处于最后一个步骤(DSC的值为LST\_STEP)且此步骤的执行时间已过时，将置位输出Q和EOD，并且SFB将停留在最后一步，直到在RESET输入中设置1。

DRUM定时器只在STARTUP和RUN模式下运行。

在冷重启过程中，操作系统将复位SFB 32 “DRUM”，但在暖重启过程中则不复位。如果要在暖重启后初始化SFB 32 “DRUM”，可以在OB100中通过RESET = 1来调用它。

参数	声明	数据类型	存储区域	描述
RESET	INPUT	BOOL	I、Q、M、D、L、常数	信号电平1将复位操作序列。第一次调用块时，必须将RESET设置为1。
JOG	INPUT	BOOL	I、Q、M、D、L、常数	如果操作序列尚未进入最后一步，则上升沿(与上一次SFB调用相比)会将其切换到下一个步骤。将根据您赋给DRUM_EN的值启用下一个步骤。
DRUM_EN	INPUT	BOOL	I、Q、M、D、L、常数	控制参数，指定是否能够根据时间切换到下一个步骤(1：能够根据时间切换)
LST_STEP	INPUT	BYTE	I、Q、M、D、L、常数	最后一步的编号；可能的值：1至16
EVENT, $1 \leq i \leq 16$	INPUT	BOOL	I、Q、M、D、L、常数	事件位号i(属于步骤 i)
OUT <sub>j</sub> , $0 \leq j \leq 15$	OUTPUT	BOOL	I、Q、M、D、L	输出位号j(与OUT_WORD的位号j相同)
Q	OUTPUT	BOOL	I、Q、M、D、L	状态参数，指定为最后一个步骤选择的执行时间是否已过去。
OUT_WORD	OUTPUT	WORD	I、Q、M、D、L、P	将输出位汇集到一起形成的变量
ERR_CODE	OUTPUT	WORD	I、Q、M、D、L、P	如果执行SFB期间出错，则ERR_CODE将包含出错信息。
JOG_HIS	VAR	BOOL	I、Q、M、D、L、常数	(与用户无关：上一个SFB调用的JOG输入参数)



参数	声明	数据类型	存储区域	描述
EOD	VAR	BOOL	I、Q、M、D、L、常数	与输出参数Q相同
DSP	VAR	BYTE	I、Q、M、D、L、P、常数	第一步的编号；可能的值：1至16
DSC	VAR	BYTE	I、Q、M、D、L、P、常数	当前步骤的编号
DCC	VAR	DWORD	I、Q、M、D、L、P、常数	当前步骤中仍剩余的执行时间(ms) (仅当DRUM_EN = 1且将相应事件位设置为 = 1时才相关)
DTBP	VAR	WORD	I、Q、M、D、L、P、常数	对所有步骤有效的基准时间(以毫秒为单位)
PREV_TIME	VAR	DWORD	I、Q、M、D、L、常数	(与用户无关：上一个SFB调用的系统时间)
S_PRESET	VAR	ARRAY of WORD	I、Q、M、D、L、常数	包含每个步骤的时间因数的一维数组。有效下标为：[1 - 16]。这种情况下，S_PRESET [x]包含步骤x的时间因数。
OUT_VAL	VAR	ARRAY of BOOL	I、Q、M、D、L、常数	包含在每个步骤中输出的数值的二维数组(如果尚未使用S_MASK进行屏蔽)。有效下标为：[1 - 16, 0 - 15]。这种情况下，OUT_VAL [x, y]包含在步骤x中赋给输出位OUTy的值。
S_MASK	VAR	ARRAY of BOOL	I、Q、M、D、L、常数	包含每个步骤的屏蔽位的二维数组。有效下标为：[1 - 16, 0 - 15]。这种情况下，S_MASK [x, y]包含将在步骤x中输出的第y个值的屏蔽位。 屏蔽位的含义： <ul style="list-style-type: none"> <li>0: 上一步骤的值已赋给相应的输出位。</li> <li>1: 来自OUT_VAL的相应值已赋给相应的输出位。</li> </ul>

## 出错信息

如果出现下表所列情况中的一种，SFB 32  
“DRUM”将保持其当前状态不变，并将置位ERR\_CODE输出。

ERR_CODE(W#16#...)	说明
0000	无故障
8081	LST_STEP的值非法
8082	DSC的值非法
8083	DSP的值非法
8084	乘积 $DCC = DTBP * S\_PRESET[DSC]$ 超出值 $2^{32}-1$ (大约24.86天)

## 15 用于寻址模块的系统功能

### 15.1 使用SFC5 “GADR\_LGC”查询模块的逻辑基址

#### 描述

基于信号模块的通道，可以知道相应模块的插槽和用户数据地址区域中的偏移量。通过SFC5 “GADR\_LGC” (将地理地址转换为逻辑地址)，可以获取模块的相应逻辑地址，即最小输入和输出地址。

参数	声明	数据类型	存储区域	描述
SUBNETID	INPUT	BYTE	I、Q、M、D、L、 常数	区域标识符： <ul style="list-style-type: none"> <li>0，如果插槽位于机架0(中央机架)或1到21(扩展机架)中(其中之一)。</li> <li>相应的分布式I/O系统的DP主站标识符(如果插槽位于分布式I/O设备中)。</li> </ul>
RACK	INPUT	WORD	I、Q、M、D、L、 常数	<ul style="list-style-type: none"> <li>机架号(如果区域标识符为0)。</li> <li>分布式I/O设备的站号(如果区域标识符 &gt; 0)。</li> </ul>
SLOT	INPUT	WORD	I、Q、M、D、L、 常数	插槽号
SUBSLOT	INPUT	BYTE	I、Q、M、D、L、 常数	子模块插槽(如果没有可以插入的子模块，则必须在此指定0)
SUBADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的用户数据地址区域中的 偏移量
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
IOID	OUTPUT	BYTE	I、Q、M、D、L	区域标识符： <b>B#16#54</b> ： 外围输入(PI) <b>B#16#55</b> ： 外设输出(PQ) 如果为混合模块，则SFC将提供较小地址的区域标识符。如果地址相同，则SFC将提供标识符 <b>B#16#54</b> 。
LADDR	OUTPUT	WORD	I、Q、M、D、L	模块的逻辑基址

## 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8094	没有用指定SUBNETID组态子网。
8095	RACK参数的值非法。
8096	SLOT参数的值非法。
8097	SUBSLOT参数的值非法。
8098	SUBADDR参数的值非法。
8099	插槽未组态或由使用组合地址(ET 200S)的子模块占用。
809A	未组态所选插槽的子地址。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 15.2 使用SFC49 “LGC\_GADR”查询属于一个逻辑地址的模块插槽

### 描述

通过SFC49 “LGC\_GADR”(将逻辑地址转换为地理地址)，可以获得属于逻辑地址的模块插槽及模块的用户数据地址区域中的偏移量。

### 注意

不能将SFC49 “LGC\_GADR”用于使用组合地址(ET 200S)的子模块。

参数	声明	数据类型	存储区域	描述
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	地址区域的标识符： <ul style="list-style-type: none"> <li>• B#16#00: LADDR的第15位指定输入(位15=0)或输出地址(位15=1)是否存在。</li> <li>• B#16#54 = 外围输入(PI)</li> <li>• B#16#55 = 外围输出(PO)</li> </ul> 如果模块为混合模块，则指定最小地址的区域标识符。如果地址相同，则指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	逻辑地址。对于混合模块，指定两个地址中较小的一个。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
AREA	OUTPUT	BYTE	I、Q、M、D、L	区域标识符：这将指定必须如何解释剩余的其它输出参数。
RACK	OUTPUT	WORD	I、Q、M、D、L	机架号
SLOT	OUTPUT	WORD	I、Q、M、D、L	插槽号
SUBADDR	OUTPUT	WORD	I、Q、M、D、L	相应模块的用户数据地址区域中的偏移量。

### 注意

注意如果在故障安全模式中使用S7-400H PLC，并在以参数LADDR调用SFC49时指定连接的DP从站中的模块的逻辑地址，则将在RACK参数的高字节中提供活动通道的DP主站系统标识符。如果不存在活动通道，则将相关DP主站系统的DP主站系统标识符输出到主CPU。

## 输出参数AREA

输出参数AREA指定必须采用何种方式解释输出参数RACK、SLOT和SUBADDR (参见下表)。

AREA的值	系统	RACK、SLOT和SUBADDR的含义
0	S7-400	RACK: 模块机架号。 SLOT: 插槽号。 SUBADDR: 逻辑地址与逻辑基址间的差值
1	S7-300	RACK: 模块机架号。 SLOT: 插槽号。 SUBADDR: 逻辑地址与逻辑基址间的差值
2	DP	RACK: (低字节)站号 RACK: (高字节) DP主站系统标识符 SLOT: 站中的插槽号 SUBADDR: 相应模块的用户数据地址区域中的偏移量。
3	S5 P区域	RACK: 模块机架号。 SLOT: 适配室的插槽号 SUBADDR: S5 x范围中的地址
4	S5 O区域	RACK: 模块机架号。 SLOT: 适配室的插槽号 SUBADDR: S5 x范围中的地址
5	S5 IM3区域	RACK: 模块机架号。 SLOT: 适配室的插槽号 SUBADDR: S5 x范围中的地址
6	S5 IM4区域	RACK: 模块机架号。 SLOT: 适配室的插槽号 SUBADDR: S5 x范围中的地址

## 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	指定的逻辑地址无效或者IOID参数的值非法
8xyy	常规出错信息, 请参见使用输出参数RET_VAL评估出错。

## 15.3 使用SFC50 “RD\_LGADR”查询模块的所有逻辑地址

### 描述

从模块的一个逻辑地址开始。通过SFC50 “RD\_LGADR” (读取模块逻辑地址), 可以获取此模块的所有已声明的逻辑地址。先前您已通过STEP 7将地址分配给模块。SFC50按升序将获取的逻辑地址输入到PEADDR域或PAADDR域中。

参数	声明	数据类型	存储区域	描述
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	区域标识符: <ul style="list-style-type: none"> <li>B#16#00: LADDR的第15位指定输入(位15=0)或输出地址(位15=1)是否存在。</li> <li>B#16#54: 外围输入(PI)</li> <li>B#16#55: 外设输出(PQ)</li> </ul>
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	一个逻辑地址
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
PEADDR	OUTPUT	ANY	I、Q、M、D、L	用于PI地址的域, 域元素的数据类型必须为WORD。
PECOUNT	OUTPUT	INT	I、Q、M、D、L	返回的PI地址数
PAADDR	OUTPUT	ANY	I、Q、M、D、L	用于PQ地址的域, 域的数据类型必须为WORD。
PACOUNT	OUTPUT	INT	I、Q、M、D、L	返回的PQ地址数

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8090	指定的逻辑地址无效或者IOID参数的值非法。
80A0	输出参数PEADDR中的出错: 域元素的数据类型不是 WORD。
80A1	输出参数PAADDR中的出错: 域元素的数据类型不是 WORD。
80A2	输出参数PEADDR中的出错: 指定的域不能容纳所有逻辑地址。
80A3	输出参数PAADDR中的出错: 指定的域不能容纳所有逻辑地址。
8xyy	常规出错信息, 请参见使用输出参数RET_VAL评估出错。





## 16 用于分布式I/O的SFC

### 16.1 使用SFC7 “DP\_PRAL”在DP主站上触发硬件中断

#### 描述

利用SFC7 “DP\_PRAL”，可以从智能从站的用户程序触发DP主站上的硬件中断。此中断将启动DP主站上的OB40。

使用输入参数AL\_INFO能够识别硬件中断的原因。此中断标识符被传送至DP主站，您可以判断OB40(变量OB40\_POINT\_ADDR)中的标识符。

被请求的硬件中断将由输入参数IOID和LADDR唯一确定。对于传送存储器中的每个已组态地址区域，无论何时都能准确地触发一个硬件中断。

#### SFC的工作方式

SFC7 “DP\_PRAL”以异步方式运行，换言之，它通过多个SFC调用来执行。通过使用REQ=1调用SFC7启动硬件中断请求。

作业状态由输出参数RET\_VAL和BUSY指示，参见异步SFC中参数REQ、RET\_VAL和BUSY的含义在DP主站上执行完OB40时，该作业便完成。在DP主站上执行完OB40时，该作业便完成。

---

#### 注意

如果将DP从站作为标准从站运行，则当DP主站获得诊断帧后，作业便会立即完成。

---

## 识别一个作业

输入参数IOID和LADDR可唯一确定作业。

如果已在DP从站中调用了SFC7“DP\_PRAL”，并在主站确认已请求的硬件中断之前再次调用此SFC，则SFC的反应方式主要取决于新调用是否涉及同一个作业：如果参数IOID和LADDR与某个尚未完成的作业相匹配，便将SFC调用视为继续调用，而不考虑参数AL\_INFO的值，并将值W#16#7002输入到RET\_VAL中。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1: DP主站上属于从站的 硬件中断
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	传送存储器中地址范围的标识符(以DP从站的角度): B#16#54= 外围设备输入(PI) B#16#55= 外围设备输出(PQ) 属于混合模块的范围的标识符是两个地址中较低的那个地址。如果地址相同,则指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	传送存储器中地址范围 的起始地址(以DP从站的角度)。 如果是属于混合模块的范围,则指定两个地址中的较低地址。
AL_INFO	INPUT	DWORD	I、Q、M、D、L、 常数	中断ID 将被传送至要在DP主站上启动的OB40(变量OB40_POINT_ADDR)。 如果要运行带有远程主站的智能从站,必须对 主站上的诊断帧进行判断。 (参见/70/)
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在激活功能时出现出错,出错,返回值将包含出错代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 触发的硬件中断尚未 得到DP主站的确认。

## 出错信息

出错代码(W#16#...)	说明
0000	作业已执行且未出错。
7000	REQ=0的首次调用。硬件中断请求未激活；BUSY的值为0。
7001	REQ=1的首次调用。已将硬件中断请求发送至DP主站；BUSY的值为1。
7002	中间调用(与REQ无关)：已触发的硬件中断尚未得到DP主站的确认；BUSY的值为1。
8090	传送存储器中地址范围的起始地址不正确。
8091	中断被阻塞(由用户组态阻塞)
8093	参数IOID和LADDR将对不具备处理硬件中断请求功能的模块寻址。
80B5	不允许在DP主站中调用。
80C3	所需资源(内存等)此时已被占用。
80C5	分布式I/O设备此时不可用(即站出现故障)。
80C8	在当前DP主站操作模式下不允许使用该功能。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 16.2 用SFC11 “DPSYC\_FR”同步DP从站组

### 描述

利用SFC11 “DPSYC\_FR”，可以同步一组或多组DP从站。

该功能可将以下控制命令之一或这些命令的组合发送到相关组中：

- SYNC(同步输出并冻结DP从站的输出状态)
- UNSYNC(取消SYNC控制命令)
- FREEZE(冻结DP从站中从已冻结输入中读取的输入状态)
- UNFREEZE(取消FREEZE控制命令)

---

### 注意

注意，执行重新启动/暖重启时，控制命令SYNC和FREEZE仍保持有效。

另外请注意，在给定时间内只能发起一个SYNC-/UNSYNC请求或一个FREEZE/UNFREEZE请求。

---

### 必要条件

在发送上文列出的控制命令之前，必须使用STEP 7将DP从站分配给各个组(参见231)。必须了解将哪个DP从站分配给哪个组以及该组的编号，还要知道各个组对SYNC/FREEZE的反应。

### SFC的工作方式

SFC11 “DPSYC\_FR”是异步SFC；换言之，它的执行要涉及多个SFC调用。作业通过用REQ=1调用SFC11启动。

作业的状态由输出参数RET\_VAL和BUSY指出，另请参考异步SFC中参数REQ、RET\_VAL和BUSY的含义。

### 识别一个作业

如果已触发一个SYNC/FREEZE作业，且在第一个作业完成之前再次调用了SFC11，则SFC的响应取决于新调用是否针对同一个作业。如果输入参数LADDR、GROUP和MODE匹配，则SFC调用将被视为后续调用。

## 写DP模块输出

对DP模块输出的写操作可通过下列方式触发：

- 通过向DP I/O传送命令，
- 通过将过程映像输出表写入模块中(在OB1结束处通过操作系统或通过调用SFC27 “UPDAT\_PO”),
- 通过调用SFC15 “DPWR\_DAT”。

正常操作下，DP主站会将输出字节循环(在PROFIBUS DP总线周期内)传送给DP从站的输出。

如果希望同时将某些输出数据(可能分布于多个从站)应用于过程输出，可以使用SFC11 “DPSYC\_FR”向相关DP主站发送SYNC命令。

## SYNC的作用

利用SYNC控制命令，选定组的DP从站将切换至“同步”模式。换言之，DP主站传送当前输出数据并指示相关DP从站冻结各自输出。利用随后的输出帧，DP从站将输出数据输入到内部缓冲区中，输出的状态保持不变。

执行完每一条SYNC控制命令后，选定组的DP从站都会将其内部缓冲区中的输出数据应用到过程的输出中。

使用SFC11 “DPSYC\_FR”发送UNSYNC控制命令后，输出只是再次循环更新。

---

### 注意

如果选定组的DP从站当前未连接到网络上或者在发送控制命令时出现故障，那么这些从站将不会切换至SYNC模式。SFC的返回值中不会传达此信息。

---

## 读取DP模块的输入数据

按照下列方式读取DP模块的输入数据：

- 对DP I/O使用装载命令，
- 更新过程映像输入表时(启动OB1时通过操作系统或通过调用SFC26 “UPDAT\_PI”),
- 通过调用SFC14 “DPRD\_DAT”。

正常操作下，DP主站将从其DP从站循环(在PROFIBUS DP总线周期内)接收此输入数据，并向CPU提供这些数据。

如果希望同时从过程读取某些输入数据(可能分布于多个从站)，可以使用SFC11 “DPSYC\_FR”向相关DP主站发送FREEZE命令。

## FREEZE的作用

利用FREEZE控制命令，可将DP从站切换到“冻结”模式，换言之，DP主站指示DP从站冻结当前输入状态。然后，将冻结的数据传送到CPU的输入区域。

执行每条FREEZE控制命令后，DP从站都会再次冻结各自的输入状态。

仅当用SFC11 “DPSYC\_FR”发送UNFREEZE控制命令后，DP主站才再次开始循环接收当前输入状态。

---

### 注意

如果选定组的DP从站当前未连接到网络上或者在发送控制命令时出现故障，那么这些从站将不会切换至FREEZE模式。SFC的返回值中不会传达此信息。

---

## 数据的一致性

由于SFC11 “DPSYC\_FR”功能是非循环的且能被更高优先级中断，因而在使用SFC “DPSYC\_FR”时应确保过程映像与实际输入输出保持一致。

如果您遵循下列一致性规则，这一点将得到保证：

- 为“SYNC输出”和“FREEZE输入”定义适当的过程映像区域(仅适用于S7-400)。首次调用SYNC作业之前调用SFC27 “UPDAT\_PO”。在最后一次调用FREEZE作业之后立即调用SFC26 “UPDAT\_PI”。
- 或者：对于与SYNC作业有关的输出和与FREEZE作业有关的输入，仅使用直接I/O访问。在SYNC作业处于激活状态时不得对这些输出进行写操作，在FREEZE作业处于激活状态时不得读取这些输入。

## 使用SFC15和SFC14

如果使用SFC15

“DPWR\_DAT”，则必须在向相关输出发送SYNC作业之前完成此SFC。

如果使用SFC14

“DPRD\_DAT”，则必须在向相关输入发送FREEZE作业之前完成此SFC。

## SFC11 “DPSYC\_FR”和启动

用户必须自行负责在启动OB中发送SYNC和FREEZE控制命令。

如果希望在启动用户程序后，一组或多组输出处于“同步”模式，则必须在启动期间初始化这些输出，并用SYNC控制命令完整执行SFC11 “DPSYC\_FR”。

如果希望在启动用户程序后，一组或多组输入处于FREEZE模式，则必须在启动期间用FREEZE控制命令对这些输入完整执行SFC11 “DPSYC\_FR”。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、 L、 常数	电平触发的控制参数 REQ=1：触发SYNC/FREEZE作业
LADDR	INPUT	WORD	I、Q、M、D、 L、 常数	DP主站的逻辑地址
GROUP	INPUT	BYTE	I、Q、M、D、 L、 常数	组选择 位0 = 1：选择了组1 位1 = 1：选择了组2 ： 位7 = 1：选择了组8 可以为每个作业选择多个组。 值B#16#0无效。

参数	声明	数据类型	存储区域	描述
MODE	INPUT	BYTE	I、Q、M、D、 L、 常数	<p>作业ID(编码符合EN 50 170 Volume 2, PROFIBUS)</p> <p>位0:           保留(值0)</p> <p>位1:           保留(值0)</p> <p>位2           = 1:     执行UNFREEZE               = 0:     无意义</p> <p>位3           = 1:     执行FREEZE               = 0:     无意义</p> <p>位4           = 1:     执行UNSYNC               = 0:     无意义</p> <p>位5           = 1:     执行SYNC               = 0:     无意义</p> <p>位6:           保留(值0)</p> <p>位7:           保留(值0)</p> <p>可能的值:</p> <ul style="list-style-type: none"> <li>• 每个作业只有一个ID: <ul style="list-style-type: none"> <li>- B#16#04(UNFREEZE)</li> <li>- B#16#08(FREEZE)</li> <li>- B#16#10(UNSYNC)</li> <li>- B#16#20(SYNC)</li> </ul> </li> <li>• 每个作业有多个ID: <ul style="list-style-type: none"> <li>- B#16#14(UNSYNC, UNFREEZE)</li> <li>- B#16#18(UNSYNC, FREEZE)</li> <li>- B#16#24(SYNC, UNFREEZE)</li> <li>- B#16#28(SYNC, FREEZE)</li> </ul> </li> </ul>
RET_VAL	OUTPUT	INT	I、Q、M、D、 L	<p>如果在函数处于激活状态时出错， 则返回值将包含出错代码。</p> <p>每次执行完块后 都必须对RET_VAL进行判断。</p>
BUSY	OUTPUT	BOOL	I、Q、M、D、 L	<p>BUSY=1: SYNC/FREEZE作业尚未完成。</p>



## 出错信息

出错代码(W#16#...)	说明
0000	作业已完成且没有出错。
7000	REQ=0的首次调用。用LADDR、GROUP和MODE指定的作业未激活；BUSY的值为0。
7001	REQ=1的首次调用。用LADDR、GROUP和MODE指定的作业已被触发；BUSY的值为1。
7002	中间调用(与REQ无关)。已激活的SYNC /FREEZE作业仍处于激活状态；BUSY的值为1。
8090	用LADDR选定的模块不是DP主站。
8093	对于用LADDR(DP主站的组态或版本)选择的模块，不允许使用此SFC。
8094	GROUP参数无效
8095	MODE参数无效
80B0	用GROUP选定的组未组态。
80B1	用GROUP选定的组未分配给此CPU。
80B2	用MODE指定的SYNC作业不允许在用GROUP选定的组中使用。
80B3	用MODE指定的FREEZE作业不允许在用GROUP选定的组中使用。
80C2	DP主站上出现临时资源短缺：DP主站当前正在处理CPU所允许的最多作业数。
80C3	目前不能激活此SYNC/UNSYNC作业，原因是在任何时候都只能触发一个SYNC/UNSYNC作业。检查用户程序。
80C4	目前不能激活此FREEZE/UNFREEZE作业，原因是在任何时候都只能触发一个FREEZE-/UNFREEZE作业。检查您的用户程序。
80C5	分布式I/O不可访问：DP子系统故障
80C6	由于CPU断开I/O连接，作业被中止
80C7	由于DP主站上的暖重启或冷重启，作业被中止
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 16.3 用SFC12 “D\_ACT\_DP”激活和取消激活DP从站

### 描述

利用SFC12 “D\_ACT\_DP”，可以明确地取消激活及再次激活已组态的DP从站。此外，还可以确定当前是激活还是取消激活每个已分配的DP从站。

SFC12 不能用在通过DP/PA链接与DP主站系统相连的PROFIBUS PA现场设备上。

---

### 注意

如果所有SFC12 作业均忙，则不能将已修改的组态到从PG下载到CPU中(在CiR处理期间)。

在CiR处理期间，CPU在接收已修改组态的下载时，会拒绝SFC12的启动请求。

---

### 用途

如果在实际并不存在或非当前所需的CPU中组态DP从站，则该CPU仍会继续以正常时间间隔访问这些DP从站。而在取消激活这些从站后，则其它CPU访问也将停止。利用这种方法，有可能达到最快的DP总线周期，同时不再发生相应的出错事件。

### 应用实例

从机器制造厂商的角度看，在连续生产机器的过程中，有大量的设备选件可供使用。但交付的每台机器仅是选定组件的一个组合。

制造商将所有这些可能的机器选件均组态为DP从站，以便创建并维护包含所有可能选件的通用用户程序。利用SFC12，可以取消激活所有DP从站，这些从站在机器启动时并不存在。

与此类似的情况是机床，可以使用大量加工选件，但实际上常常用到的仅是其中的一小部分。这些工具被作为DP从站来实现。利用SFC12，用户程序可以激活当前需要的工具而取消激活那些以后才需要的工具。

### SFC的工作方式

SFC12 “D\_ACT\_DP”以异步方式运行，换言之，它通过多个SFC调用来执行。通过使用REQ=1调用SFC12启动请求。

作业状态由输出参数RET\_VAL和BUSY指示；参见参考“HB用于S7--300/400系统和标准功能的系统软件”的第2.2节。

## 识别一个作业

如果已启动一个取消激活或激活的作业，并在该作业完成前再次调用SFC12，则SFC的反应方式主要取决于新调用是否涉及同一个作业：如果参数LADDR匹配，SFC调用被解释为继续调用。

## 取消激活DP从站

用SFC12

取消激活DP从站后，其过程输出将被设置为已组态的替换值或0(安全状态)。分配的DP主站将不再继续寻址此DP从站。DP主站或CPU上的出错LED不会将取消激活的DP从站识别为出现故障或丢失。

而取消激活的DP从站输入的过程映像将用0更新，即将其作为有故障的DP从站处理。

如果要使用程序直接访问先前取消激活的DP从站的用户数据，将调用I/O访问出错OB(OB 122)，并在诊断缓冲区中输入相应的启动事件。如果尝试使用SFC(即SFC59 “RD\_REC”)访问取消激活的DP从站，则会在RET\_VAL中收到一条出错信息(与访问不可用的DP从站相同)。

取消激活DP从站不会启动程序出错OB(OB 85)，即使其输入或输出属于要更新的系统范围的过程映像。诊断缓冲区中不输入任何条目。

取消激活DP从站不会启动机架故障OB(OB 86)，操作系统也不会向诊断缓冲区输入条目。

如果用SFC12 取消激活DP站后该站出现故障，则操作系统不对该故障进行检测。因而不会后续启动OB86或诊断缓冲区条目。只有在重新激活该站后才能检测到站故障，并在RET\_VAL中指示出来。

如果希望取消激活作为交叉通讯中发送器的DP从站，建议您首先取消激活用于检测发送器要将哪些输入数据传送到DP主站的接收器(监听器)。仅当执行此步骤后才取消激活发送器。

## 激活DP从站

用SFC12重新激活DP从站时，将由指定的DP主站对其进行组态和分配参数(与故障站的恢复相同)。从站能够传送用户数据时，激活操作便完成。

激活DP从站不启动程序出错OB(OB85)，即使其输入或输出属于要更新的系统范围的过程映像。并且不会向诊断缓冲区输入条目。

激活DP从站不启动机架故障OB(OB86)，操作系统也不会向诊断缓冲区输入条目。

如果尝试使用SFC12激活一个已取消激活、并且实物已经从DP总线分离的从站，则CPU上的LED“DP-BUSF”将闪烁约一分钟。

这个监视期到时后，SFC会返回出错消息W#16#80A2，并关闭LED。

从站仍保持取消激活状态。如果以后从站重新连接到DP总线，则必须用SFC12重新激活它。

**注意**

激活DP从站可能需要花费一定时间。因此，如果希望取消当前的激活作业，可对LADDR使用同一个数值和MODE=2再次启动SFC12。重复调用SFC12，直至RET\_VAL = 0指示成功取消激活。

如果希望激活参与交叉通讯的DP从站，建议您首先激活发送器，然后激活接收器(监听器)。

**CPU启动**

根据启动模式的不同，CPU操作系统对DP从站的作用如下：

- 在冷重启和暖重启模式下，从站被自动激活。
- 在热重启模式下，从站的激活状态保持不变，即已激活的从站保持已激活状态，取消激活的从站保持取消激活状态。

CPU启动之后，将定期尝试联系所有已组态但未取消激活的从站，这些从站或者不存在或者未响应。

**注意**

启动OB不支持对SFC12的调用。

**参数**

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、常数	电平触发的控制参数REQ=1：执行激活或取消激活
MODE	INPUT	BYTE	I、Q、M、D、L、常数	作业ID。可能的值： <ul style="list-style-type: none"> <li>• 0：是激活还是取消激活已寻址DP从站的请求信息DP从站的请求信息</li> <li>• 1：激活DP从站</li> <li>• 2：取消激活DP从站</li> </ul>
LADDR	INPUT	WORD	I、Q、M、D、L、常数	DP从站的所有逻辑地址
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在处理功能时出现出错，则返回值将包含一个出错代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	激活代码： <ul style="list-style-type: none"> <li>• BUSY=1：作业仍处于激活状态。</li> <li>• BUSY=0：作业已被终止。</li> </ul>

## 出错信息

出错代码(W#16#...)	说明
0000	作业已完成且没有出错。
0001	DP从站已激活。(仅当MODE = 0时才可能出现此出错代码。)
0002	DP从站已被取消激活(仅当MODE = 0时才可能出现此出错代码)
7000	REQ=0的首次调用。用LADDR指定的作业未激活；BUSY的值为0。
7001	REQ=1的首次调用。用LADDR指定的作业已被触发；BUSY的值为1。
7002	中间调用(与REQ无关)。已激活的作业仍处于激活状态；BUSY的值为1。
8090	<ul style="list-style-type: none"> <li>尚未用LADDR中指定的地址组态模块。</li> <li>将CPU作为I从站运行，并且已在LADDR中为此I从站指定了一个地址。</li> </ul>
8092	当前没有为已寻址DP从站处理任何激活作业(仅当MODE = -1时才可能出现此出错代码。)
8093	没有将任何DP从站分配给LADDR中指定的地址(未提交任何项目)，或者参数MODE为未知
80A1	已寻址DP从站不能被参数化(仅当MODE = 1时才可能出现此出错代码)。 注意： 仅当已激活从站在参数化期间再次出现故障时，CPU才提供此信息。 如果单个模块的参数化未成功，则SFC将返回出错信息W#16#0000
80A2	已寻址DP从站未返回确认。
80A3	相关的DP主站不支持此功能。
80A4	CPU不支持外部DP主站的这一功能。
80A6	DP从站中出现插槽出错；无法访问用户数据(仅当MODE=1时才可能出现此出错代码)。 注意： 仅当在参数化之后SFC结束之前激活从站出现故障时，SFC才会返回此出错信息。如果仅是一个模块不可用，则SFC将返回出错信息W#16#0000。
80C1	SFC12已启动，并用另一逻辑地址继续(仅当MODE=1时才可能出现此出错代码)。
80C3	<ul style="list-style-type: none"> <li>临时资源出错：CPU当前正在处理允许的最多数目的激活和取消激活作业。(仅当MODE = 1和MODE = 2时才可能出现此出错代码。)</li> <li>CPU正在忙于接收已修改的组态。当前无法启用/禁用DP从站。</li> </ul>
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 16.4 用SFC13 “DPNRM\_DG”读取DP从站的诊断数据(从站诊断)

### 从站诊断

每个DP从站都会提供一些结构符合“EN 50 170 Volume 2, PROFIBUS”的从站诊断数据。要读出此诊断数据，需要使用SFC13 “DPNRM\_DG”。

有关从站诊断数据的基本结构，请参考下表；更多信息，则请参考DP从站手册。

字节	含义
0	站状态1
1	站状态2
2	站状态3
3	主站号
4	供应商ID(高字节)
5	供应商ID(低字节)
6...	更多从站特定诊断信息

### 描述

利用SFC13 “DPNRM\_DG”(读取DP从站的诊断数据)，可读取DP从站中格式符合“EN 50 170 Volume 2, PROFIBUS”的当前诊断数据。在经过无错数据传送之后，已读取的数据被输入到由RECORD指示的目标区域中。

通过将1分配给SFC13调用中的输入参数REQ可开始读取作业。

## 功能

读取作业异步执行，换言之，它需要多个SFC13调用。作业状态由输出参数RET\_VAL和BUSY指示，另请参考异步SFC中参数REQ、RET\_VAL和BUSY的含义。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1: 读请求
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	DP从站的已组态诊断地址。 <b>注意:</b> 必须以十六进制格式输入地址。例如， 诊断地址1022表示： LADDR: =W#16#3FE。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在激活功能时出现出错，出错，则返回值将包含一个出错代码。如果未出现出错，则实际传送的数据长度将输入到RET_VAL中。
RECORD	OUTPUT	ANY	I、Q、M、D、L	已读取的诊断数据的目标区域。仅允许使用BYTE数据类型。要读取的数据记录的最小长度或目标区域为6。要发送的数据记录的最大长度为240。标准从站可提供长度超出240字节的诊断数据，最多可达到最大值244字节。这种情况下，将前240个字节传送至目标区域，并在数据中设置溢出位。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 读取作业尚未完成。

## 输入参数RECORD

CPU按照以下方式判断已读诊断数据的实际长度：

- 如果为RECORD指定的长度小于提供的数据字节数，则数据将被放弃并在RET\_VAL中输入相应出错代码。
- 如果为RECORD指定的长度大于或等于提供的数据字节数，则目标区域将接受该数据并以正数形式在RET\_VAL中输入实际长度。

---

### 注意

必须确保RECORD的实际参数在属于一个作业的所有调用中均匹配。  
作业由LADDR输入参数唯一识别。

---

## 诊断数据超出240字节的标准从站

对于标准诊断数据字节数介于241和244之间的标准从站，请注意以下几点：

- 如果为RECORD指定的长度小于240字节，则数据将被放弃，并在RET\_VAL中输入相应出错信息。
- 如果为RECORD指定的长度大于或等于240个字节，则标准诊断数据中的前240个字节将被传送至目标区域，并在数据中设置溢出位。

## 输出参数RET\_VAL

- 如果在执行功能时出错，则返回值中会包含出错代码。
- 如果在数据传送期间未出现出错，则RET\_VAL中将包含读取的数据长度(以正数表示的字节数)。

---

### 注意

在DP从站中读取的数据数量取决于它的诊断状态。

---

## 出错信息

关于如何判断RET\_VAL参数的出错信息，请参见SFC的公共参数。本章还包含SFC的常规出错信息。SFC13特定的出错信息是SFC59“RD\_REC”出错信息的子集，参见使用SFC59“RD\_REC”读一个数据记录。



### **S7-400的系统资源**

如果某个当前未进行处理的作业调用SFC13“DPNRM\_DG”，则将使用S7-400中的CPU资源(内存空间)。可以为多个DP从站连续调用SFC13，前提是未超过CPU允许的“同时”激活SFC13作业的最大数目。在**/101/**中可以找到此类作业的最大数目。

如果“同时”激活多个作业，则将执行所有这些作业，并且彼此不会干扰。

如果达到系统资源的限值，则将在RET\_VAL中指出。此时请重新执行作业。

## 16.5 使用SFC14 “DPRD\_DAT”读取DP标准从站的连续数据

### 数据的一致性

参考“S7通讯和S7基本通讯概述 – 数据的一致性”。

### SFC14的用途

需要SFC14 “DPRD\_DAT”的原因是使用用于访问I/O或过程映像输入表的装载指令最多只能读出四个连续字节。

---

#### 注意

如果需要，还可以通过输入的过程映像读取连续数据。要确定CPU是否支持此功能，请参考“S7-300可编程控制器，硬件与安装”手册。

所有S7-400 CPU均支持此功能。

---



---

#### 当心

在使用SFC14P;“DPRD\_DAT”时，应避免访问已分配了带有OB6x连接(时钟中断)的过程映像分区的I/O区域。

---

### 描述

利用SFC14 “DPRD\_DAT” (读取DP标准从站的连续数据)，能读取DP标准从站的连续数据。

为查找此处适用的最大长度，请参考下列手册：

- S7-300可编程控制器，硬件与安装
- ET 200S接口子模块，IM151-7 CPU
- 基本子模块BM147CPU

对于S7-400 CPU，最大长度是32字节。

如果在数据传送期间未出现出错，则已读取的数据将被输入到由RECORD标识的目标区域中。

目标区域的长度必须与用STEP 7为选定模块组态的长度相同。

如果从具有模块化设计或具有多个DP标识符的DP标准从站读取数据，则通过指定组态的起始地址，每个SFC14调用只能访问一个模块/DP标识符的数据。

参数	声明	数据类型	存储区域	描述
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的I区域中已组态的起始地址，将从该处读取数据。 <b>注意：</b> 必须以十六进制格式输入地址。例如， 诊断地址100表示： LADDR: =W#16#64。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在激活功能时出现出错，出错，则返回值将包含一个出错代码。
RECORD	OUTPUT	ANY	I、Q、M、D、L	已读取的用户数据的目标区域。必须与用STEP 7为选定模块配置的长度完全相同。仅允许使用数据类型BYTE。

## 出错信息

### 注意

如果访问DPV1从站，这些从站的出错信息可从DP主站转发到SFC。有关此出错信息的详细信息，请参见用SFB 54 “RALRM” STATUS[3]从DP从站接收中断。

出错代码(W#16#...)	说明
0000	未出错。
8090	<ul style="list-style-type: none"> <li>没有为指定的逻辑基址组态模块或者</li> <li>忽略了关于连续数据长度的限制或者</li> <li>没有以十六进制格式在LADDR参数中输入起始地址。</li> </ul>
8092	在ANY引用中，指定了一个非BYTE的类型。
8093	在LADDR中指定的逻辑地址处没有可从中读取连续数据的DP模块。
80A0	访问I/O设备时检测到访问出错。
80B0	外部DP接口模块从站出现故障。
80B1	指定目标区域的长度不等于用STEP 7组态的用户数据的长度。
80B2	外部DP接口模块出现系统出错。
80B3	外部DP接口模块出现系统出错。
80C0	模块尚未读取数据。
80C2	外部DP接口模块出现系统出错。
80Fx	外部DP接口模块出现系统出错。
87xy	外部DP接口模块出现系统出错。
808x	外部DP接口模块出现系统出错。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 16.6 使用SFC15 “DPWR\_DAT”将连续数据写入到DP标准从站

### 数据的一致性

参考 S7通讯和S7基本通讯概述 – 数据的一致性。

### SFC15的用途

需要SFC15 “DPWR\_DAT”的原因是使用用于访问I/O或过程映像输入表的传送指令最多只能写入四个连续字节。

---

#### 注意

如果需要，还可以通过输入的过程映像读取连续数据。要确定CPU是否支持此功能，请参考“S7-300可编程控制器，硬件与安装”手册。

所有S7-400 CPU均支持此功能。

---



---

#### 当心

在使用SFC15 “DPRD\_DAT”时，应避免访问已分配了带有OB6x连接(时钟中断)的过程映像分区的I/O区域。

---

### 描述

利用SFC15 “DPWR\_DAT” (向DP标准从站写入连续数据)，可将RECORD中的数据始一致地传送至已寻址的DP标准从站，如必要，还可以传送至过程映像(即如果已将DP标准从站的各个地址区域组态为过程映像中的一致性范围)：

- S7-300可编程控制器，硬件与安装
- ET 200S接口子模块，IM151-7 CPU
- 基本子模块BM147CPU

对于S7-400 CPU，最大长度是32字节。

数据是同步传送的，即完成SFC时也完成了写作业。

源区域的长度必须与用STEP 7为选定模块组态的长度相同。

如果DP标准从站是模块化设计，则只能访问DP从站的一个模块。

参数	声明	数据类型	存储区域	描述
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	模块的过程映像输出区域中已组态的起始地址，数据将被写入该地址。 <b>注意：</b> 必须以十六进制格式输入地址。例如， 诊断地址100表示： LADDR: =W#16#64。
RECORD	INPUT	ANY	I、Q、M、D、L	要写入用户数据的源区域。必须与用STEP 7为选定模块组态的长度完全相同。仅允许使用数据类型BYTE。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在激活功能时出现出错，出错，则返回值将包含一个出错代码。

## 出错信息

### 注意

如果访问DPV1从站，这些从站的出错信息可从DP主站转发到SFC。有关此出错信息的详细信息，请参见用SFB 54 “RALRM” STATUS[3]从DP从站接收中断。

出错代码(W#16#...)	说明
0000	未出错。
808x	外部DP接口模块出现系统出错。
8090	<ul style="list-style-type: none"> <li>没有为指定的逻辑基址组态模块或者</li> <li>忽略了关于连续数据长度的限制或者</li> <li>没有以十六进制格式在LADDR参数中输入起始地址。</li> </ul>
8092	在ANY引用中，指定了一个非BYTE的类型。
8093	在LADDR中指定的逻辑地址处没有要写入连续数据的DP模块/PROFINet IO设备。
80A1	访问I/O设备时检测到访问出错。
80B0	外部DP接口模块从站出现故障。
80B1	指定源区域的长度不等于用STEP 7组态的用户数据的长度。
80B2	外部DP接口模块出现系统出错。
80B3	外部DP接口模块出现系统出错。

出错代码(W#16#...)	说明
80C1	模块中先前写作业中的数据尚未被 模块处理。
80C2	外部DP接口模块出现系统出错。
80Fx	外部DP接口模块出现系统出错。
85xy	外部DP接口模块出现系统出错。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 17 用于全局数据通讯的SFC

### 17.1 使用SFC60 “GD\_SND”发送一个GD信息包

#### 描述

通过SFC60 “GD\_SND”(全局数据发送)，采集GD信息包的数据，并通过在GD信息包中指定的路径发送。必须已经使用STEP 7对GD信息包进行组态。

可以在用户程序的任何地方调用SFC60 “GD\_SND”。

扫描速率以及系统在循环检测点处所进行的数据采集和发送并不受SFC60用的影响。

#### 中断能力

SFC60 “GD\_SND”可以被具有更高优先级的程序中断。还可以在更高的优先级中为同一个GD信息包再次调用SFC60。

然后，以更高的优先级收集、发送数据。当程序返回到被中断的SFC时，将立即终止执行此程序块，并丢弃已经接收到的数据。

此过程意味着，在具有更高优先级的处理过程中，传送的是一致性数据(从全局数据的意义上定义一致性)。

#### GD的数据的一致性

下列规则适用于从各种存储区域采集到的和发送的数据的一致性。

下列数据是一致的：

- 简单数据类型(位、字节、字和双字)
- 数据类型为字节、字和双字的数组，最大数据长度取决于特定的CPU类型。

### 确保整个GD信息包的一致性

CPU上发送数据的GD信息包所具有的结构不会自动保证所采集到的数据是一致的。例如，当信息包由字节数组组成，而字节的数目超过了指定CPU所允许的最大长度时，就会发生这种情况。

然而，如果需要整个GD信息包具有一致性，请在程序中遵守下列操作步骤：

- 通过调用SFC39 “DIS\_IRT”或SFC41 “DIS\_AIRT”，禁止或延迟出现具有更高优先级的中断和异步出错。
- 调用SFC60 “GD\_SND.”
- 通过调用SFC40 “EN\_IRT”或SFC42 “EN\_AIRT”，重新允许具有更高优先级的中断和异步出错。

参数	声明	数据类型	存储区域	描述
CIRCLE_ID	INPUT	BYTE	I、Q、M、D、L、 常数	要发送的GD信息包所在的GD环的数目。在用STEP 7组态全局数据时，指定这个数字。 允许使用的数值：1到16。可以在CPU的技术数据中找到允许使用的GD环的最大数目。
BLOCK_ID	INPUT	BYTE	I、Q、M、D、L、 常数	要在所选择的GD环中发送的GD信息包的数目。在用STEP 7组态全局数据时设置此数字。 允许使用的数值：1到3。可以在CPU的技术数据中找到允许使用的GD环的最大数目。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息



## 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8081	并未组态通过参数CIRCLE_ID和BLOCK_ID所选择的GD信息包。
8082	参数CIRCLE_ID或BLOCK_ID的数值非法，或这两个参数的数值都非法。
8083	执行SFC时出错。出错类型输入到为状态信息组态的变量中。可以通过用户程序对此信息进行评估。
8084	提前终止SFC的执行，因为在更高的优先级中为同一个GD信息包重新调用了参见“中断能力”。
8085	在将状态信息输入到所组态的变量中时出错。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 注意

在每次调用SFC60之后，都应该评估相应GD信息包的状态，并且，如果有必要，则复位该状态。

## 17.2 通过SFC61 “GD\_RCV”编程接受已接收到的GD信息包

### 描述

通过SFC61 “GD\_RCV”(全局数据接收), 从进入的GD帧中为单个GD信息包提取数据, 然后输入接收到的GD信息包中。必须已经通过STEP 7对此进行了组态。

可以在用户程序的任何地方调用SFC61 “GD\_SND”。

扫描速率以及系统在循环检测点处所进行的数据提取并不受SFC61调用的影响。

### 中断能力

SFC61可以被具有更高优先级的程序中断, 然而, 其前提是必须保证为全局数据定义的数据的一致性。如果函数的处理被中断, 则可能是: 为同一个GD信息包以更高优先级再次调用了SFC61。

然后, 将数据输入到在更高优先级中接收到的GD信息包中。当程序返回到中断的SFC时, 将立即终止该过程。

### GD的数据的一致性

下列规则适用于在各种内存区域输入的数据的一致性。

下列数据是一致的:

- 简单数据类型(位、字节、字和双字)
- 数据类型为字节、字和双字的数组, 最大数据长度取决于接收CPU。

### 确保整个GD信息包的一致性

接收CPU上的GD信息包所具有的结构不会自动保证它的数据来自一个相同的帧。例如, 当信息包由三个GD单元组成时就是这种情况。

然而, 如果需要整个GD信息包具有一致性, 请在程序中遵守下列步骤:

- 通过调用SFC39 “DIS\_IRT”或SFC41 “DIS\_AIRT”, 禁止或延迟出现具有更高优先级的中断和异步出错。
- 调用SFC60 “GD\_SND.”

- 通过调用SFC40 “EN\_IRT”或SFC42 “EN\_AIRT”，重新允许具有更高优先级的中断和异步出错。

参数	声明	数据类型	存储区域	描述
CIRCLE_ID	INPUT	BYTE	I、Q、M、D、L、 常数	用于输入进入的GD信息包的GD环的数目。 允许使用的数值：1到16。可以在CPU的技术数据中找到允许使用的GD环的最大数目。
BLOCK_ID	INPUT	BYTE	I、Q、M、D、L、 常数	所选择的GD环中的GD信息包的数目，将在其中输入进入的数据。在全局数据的组态过程中，通过STEP 7指定该数目。 允许使用的数值：1到3。可以在CPU的技术数据中找到允许使用的GD环的最大数目。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息

## 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8081	并未组态通过参数CIRCLE_ID和BLOCK_ID所选择的GD信息包。
8082	参数CIRCLE_ID或BLOCK_ID的数值非法，或这两个参数的数值都非法。
8083	执行SFC时出错。出错类型被输入到为状态信息组态的变量中。可以在用户程序中评估该数据。
8084	因为以更高的优先级为同一个GD信息包重新调用了SFC61 (参见“中断能力”)，提早终止了SFC的执行。
8085	在将状态信息输入到所组态的变量中时出错。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 注意

在每次调用SFC61之后，都应该评估相应GD信息包的状态，并且，如果有必要，则复位该状态。



## 18 通讯和S7基本通讯的概述

### 18.1 S7通讯块和S7基本通讯块之间的差别

#### 选择标准

除了全局数据通讯以外，SIMATIC S7可编程控制器的CPU/FM之间还有其它两种数据交换方式：

- 对于未组态的S7连接，使用通讯SFC进行数据交换
- 对于已组态的S7连接，使用通讯SFB进行数据交换

所选择的方法取决于所使用的SIMATIC S7可编程控制器(S7-300、S7-400)和用于数据交换的其它参数。可以基于下表包含的标准列表。

标准	用于未组态S7连接的通讯SFC	用于已组态的S7连接的通讯SFB
程序块的可用性	S7-300和S7-400作为SFC	S7-300作为FB和FC S7-400作为SFB和SFC
通讯连接	连接没有组态。当激活SFC时连接建立。在数据已经传送完毕之后，连接或者保持建立，或者由参数控制而终止。如果暂时无法建立连接，则相应作业将无法发送。	在系统配置中永久配置连接。
切换到STOP模式	如果启动数据传送的CPU切换到STOP模式，则将终止它所建立的所有连接。	此连接将保持在STOP模式。
到一个伙伴的几个连接	在任何时刻，一个连接最多只能连接到一个通讯伙伴。	可以建立到同一个伙伴的几个连接。
地址范围	可以在本地站点或在MPI子网中寻址的模块	可以在MPI网络、PROFIBUS或工业以太网上寻址的模块。

标准	用于未组态S7连接的通讯SFC	用于已组态的S7连接的通讯SFB
通讯伙伴数	可以一个接一个查询的通讯伙伴数不受可用的连接资源限制(参见/70/,/101/)(当程序正在运行时,可以重新建立和终止连接。)	可以同时获得的通讯伙伴数受可用的连接资源限制。同时还取决于正在使用的CPU(参见/70/,/101/).
最大用户数据长度	保证76字节的用户数据长度。	可传送的最大用户数据长度取决于块类型(USEND/URCV、GET等)和通讯伙伴(S7-300、S7-400或M7)。
每个块调用所传送的变量数	只能传送一个变量。	<ul style="list-style-type: none"> <li>• S7-300: 一个变量</li> <li>• S7-400: 最多四个变量</li> </ul>
功能块分类	用于S7基本通讯的SFC是系统函数。因此,它们并不需要用户存储区。	用于S7基本通讯的SFB/FB是系统函数块。因此,它们需要背景数据块来提供实际参数和静态数据。
地址参数的动态修改	可以进行地址参数的动态修改:在激活的作业结束时,可以寻址其它通讯伙伴。	<p>S7-300:可以在功能块激活时重新组态寻址参数。当前一个作业关闭时,新参数开始有效。</p> <p>S7-400:不能进行地址参数的动态修改:连接由第一次块调用指定并固定,且在下一次暖重启或冷重启之前保持不变。</p>

### 通过S7-400实现IEC 61131-5

通过下列功能块来实现IEC标准61131-5:

- USEND(SFB8)/URCV(SFB9)
- BSEND(SFB12)/BRCV(SFB13)
- PUT(SFB15)/GET(SFB14), 相当于READ/WRITE
- STATUS(SFB22)/USTATUS(SFB23)
- ALARM(SFB33)
- NOTIFY(SFB36)
- START(SFB19)、STOP(SFB20)和RESUME(SFB21)实现用于程序控制函数的调用接口。

### 通过S7-300实现IEC 61131-5

通过下列功能块来实现IEC标准61131-5:

- USEND(SFB8)/URCV(SFB9)
- BSEND(SFB12)/BRCV(SFB13)
- PUT(SFB15)/GET(SFB14), 相当于READ/WRITE

## 18.2 数据的一致性

### 定义

可以被一致性过程同时修改的数据区的大小被称为一致数据区。大于一致数据区的数据区将被认为整体失真。

也就是说，属于一组且大于一致数据区的数据区可以同时包含新的一致数据块部分和旧的一致数据块部分。

### 实例

如果通讯块被中断，例如，被具有更高优先级的硬件中断OB所中断，则可能会引起数据不一致。如果此OB中的用户程序现在改变数据，而一部分数据已经由通讯块处理过，则已传送的数据：

- 部分来自处理硬件中断之前的时间
- 部分来自处理硬件中断之后的时间
- 也就是说，这些数据将不一致(不连贯)。

### 影响

如果数据的较大数据包以一致格式传送，则传送可能会被中断。这可能会，例如，增加CPU中的中断反应时间。

这表示：需要绝对一致传送的数据量越大，系统的中断反应时间越长。

### SIMATIC上的数据的一致性

如果用户程序包含访问公用数据的通讯函数，例如BSEND/BRCV，则可以通过使用参数DONE”来协调对此数据区的访问。这样，便可以在用户程序中确保通过通讯块本地传送的通讯区的数据的一致性。

然而，当使用S7通讯函数时，例如PUT/GET或通过OP通讯进行写/读时，在编程或组态期间必须考虑一致数据区的大小，因为在目标设备(服务器)的用户程序中没有通讯块可用于向用户程序发送同步通讯数据。

在S7-300和C7-300中(例外：CPU 318-2 DP)，在操作系统的周期检测点上，通讯数据以32个字节的数据块为单位，一致地复制到用户存储区。对于较大数据区，不能保证数据的一致性。如果要求定义数据的一致性，则用户程序中的通讯数据不可以超出32个字节(根据版本的不同，最大为8个字节)。



与此相反，在S7-400中，通讯数据不是在周期检测点上处理，而是在程序循环周期的固定时间段上进行处理。变量的一致性由系统来确保。

随后，可使用PUT/GET函数或变量的读/写来一致访问这些通讯区，例如通过OS上的OP。

## 建议

关于数据的一致性的更多信息，请参考各个块的信息描述以及[通过SIMATIC进行通讯手册](#)。

## 18.3 S7通讯块概述

### 分类

对于S7通讯，需要进行连接组态。通过SFB/FB或SFC/FC，在应用程序中调用集成的通讯功能。

可以将这些块分成下列几个类别：

- 用于数据交换的SFB/FB
- 用于改变工作状态的SFB
- 用于查询工作状态的SFB
- 用于查询连接的SFC/FC

### 可用的S7-300/400通讯块

- 用于S7-400的块位于“标准库”中。
- 用于S7-300的可装载块位于“SIMATIC\_NET\_CP”库中。

如果希望运行S7-300块，则需要S7-300系列中的SIMATIC NET CP。欲知详情，请参见相关文档。

### 用于数据交换的SFB/FB

用于数据交换的通讯SFB/FB被用于在两个通讯伙伴之间交换数据。如果只在本地模块上存在SFB，则称为单边数据交换。如果在本地模块和远程模块上都存在SFB/FB，则称为双边数据交换。

块 S7-400	块 S7-300	描述	简要描述
SFB8 SFB9	FB -8 FB9	USEND URCV	无确认的快速数据交换，不考虑通讯伙伴上通讯函数(URCV)的相应执行情况(例如，操作和维护消息)。也就是说，通讯伙伴上的数据可以被更多最新的数据所覆盖。
SFB12 SFB13	FB12 FB13	BSEND/ BRCV	到通讯伙伴的固定数据块传送。也就是说，在通讯伙伴中的接收函数(BRCV)接受该数据之前，数据传送不会结束。
SFB14	FB -14	GET	通讯伙伴用户程序中，由程序控制的变量读功能，无更多的通讯功能。
SFB15	FB15	PUT	通讯伙伴用户程序中，由程序控制的变量写功能，无更多的通讯功能。
SFB16	FB16	PRINT	向打印机发送数据(只用于S7-400)

### 用于改变工作状态的SFB

通过用于改变工作状态的SFB，可以控制远程设备的工作状态。

通过用于改变工作状态的SFB进行的数据交换是单边数据交换。

S7-400中使用的块		简要描述
SFB19	START	如果S7/M7-300/400或C7-300 CPU处于STOP操作模式，则该功能块将触发这些模块的重启动(RESTART)功能。
SFB20	STOP	如果S7/M7-300/400或C7-300 CPU处于RUN、HALT操作模式或正在启动，则该功能块将触发这些模块的停机(STOP)功能。
SFB21	RESUME	如果S7-400 CPU处于STOP模式，则该功能块将触发此模块，使其继续运行。

### 用于查询工作状态的SFB

通过用于查询工作状态的SFB，可以获得远程设备的工作状态信息。

使用SFB“STATUS”时，数据交换是单边的；而使用SFB“USTATUS”时，数据交换是双边的。

S7-400中使用的块		简要描述
SFB22	STATUS	在用户请求时，提供通讯伙伴的工作状态(S7-400-CPU、M7-300/400)。
SFB23	USTATUS	如果已经设置了相应的连接属性(发送工作状态消息)，则当CPU的工作状态发生改变时，将接收S7-400-CPU的工作状态。

### 用于查询连接的SFC/FC

S7-400中使用的块	S7-300中使用的块	简要描述
SFC62 CONTROL		查询连接的状态，该连接属于某个SFB/FB的一个实例。
	FC 62 C CNTRL	通过连接标识号查询连接的状态

#### 提示：

还可以使用SFC87 C\_DIAG来完成实际连接状态的诊断(只用于S7-400)。

### 实例程序

STEP 7提供了一个S7-400的示例程序，该程序给出了如何使用SFB进行S7通讯。这个示例程序的名称是step7\examples\com\_sfb。它位于路径文件夹“..\STEP7\Examples\ZDT01\_10”中的示例程序中。

## 18.4 用于S7基本通讯的功能块总览

### 用于S7基本通讯的SFC的分类

S7基本通讯不需要连接组态。通过用户程序中的SFC调用集成的通讯函数。

SFC分为两类：

- 用于在S7 CPU和其它有通讯功能的模块之间交换数据的SFC，条件是通讯伙伴属于相同的S7站(开头以I”标识，表示**内部**)。
- 用于在S7 CPU和其它有通讯功能的模块之间交换数据的SFC，条件是通讯伙伴连接到公共MPI子网(开头以X”标识，表示**外部**)。

对于与其它子网中的站的通讯，不能使用用于S7基本通讯的SFC。

用于基本通讯的SFC可以运行在S7-300和S7-400系列的所有CPU上。通过这些CPU，还可以向S7-200的CPU中写入变量，以及从S7-200的CPU中读取变量。

### 用于外部通讯的SFC

块		简要描述
SFC65/ SFC66	X_SEND/ X_RCV	确保将数据块传送给通讯伙伴。也就是说，在通讯伙伴中的接收函数(X_RCV)接受该数据之前，数据传送不会结束。
SFC67	X_GET	读取通讯伙伴中的变量，而不需要在通讯伙伴上运行相应的SFC。在通讯伙伴的操作系统中实现此功能。
SFC68	X_PUT	向通讯伙伴写入变量，而不需要在通讯伙伴上运行相应的SFC。在通讯伙伴的操作系统中实现此功能。
SFC69	X_ABORT	中止一个已存在的、无正在传送数据的连接。然后再在通讯连接的两端释放相应的连接资源。

## 用于内部通讯的SFC

块		简要描述
SFC72	I_GET	读取通讯伙伴中的变量，而不需要在通讯伙伴上运行相应的SFC。在通讯伙伴的操作系统中实现此功能。
SFC73	I_PUT	向通讯伙伴写入变量，而不需要在通讯伙伴上运行相应的SFC。在通讯伙伴的操作系统中实现此功能。
SFC74	I_ABORT	中止一个已存在的、无正在传送数据的连接。然后再在通讯连接的两端释放相应的连接资源。

## 实例程序

随STEP 7提供了两个用于S7基本通讯的SFC的实例程序。这两个实例程序包含在内容step7\examples\com\_SFC1和step7\examples\com\_SFC2中。

## 最大用户数据长度

在S7-300和S7-400的所有CPU上都集成了用于未组态的S7连接的通讯SFC。

所有SFC都保证可以传送76字节的用户数据(参数SD或RD)。可以在各个块上找到准确的字节数。

## 到通讯伙伴的连接

通过用于未组态的S7连接的通讯SFC，在SFC执行时建立连接。根据分配给CONT输入参数的数值，或者保持建立连接，或者在数据交换结束时终止连接。也就是说，通讯具有下列特征：

- 可以相继访问到的通讯伙伴数多于可以同时访问到的通讯伙伴数(数目取决于特定的CPU，参见/70/, /101/).
- 如果由于所有连接资源(在本地CPU或在通讯伙伴上)都正被使用，当前不能建立任何与通讯伙伴的连接，则通过RET\_VAL来指示。必须在随后的某个合适的时间点上重新触发此作业。然而，并不能保证随后可以成功建立连接。如必要，检查程序中连接资源的使用情况，并使用带有更多资源的CPU。

已存在的、用于已组态的S7连接的通讯SFB的连接，不能被用于未组态的S7连接的通讯SFC使用。

一旦触发了一个作业，则为此作业建立的连接只能用于此特定作业。只有在当前作业结束之后，才能执行其它包含相同通讯伙伴的作业。

**注意**

如果程序中包含了几个涉及到同一个通讯伙伴的作业，则必须确保在随后的某个合适的时间点上重新调用为其RET\_VAL输入了W#16#80C0的SFC。

**识别一个作业**

如果已经通过一个用于未组态S7连接的通讯SFC触发了一次数据传送或一个连接中止，并在当前传送结束之前重新调用这个SFC，则SFC的反应取决于新调用是否包含了相同的作业。下表解释了哪个输入参数为每个SFC指定一个作业。如果参数与还没有结束的作业相匹配，则将此SFC调用算作一个后继调用。

块		用于识别作业的标识符
SFC65	X_SEND	DEST_ID、REQ_ID
SFC67	X_GET	DEST_ID、VAR_ADDR
SFC68	X_PUT	DEST_ID、VAR_ADDR
SFC69	X_ABORT	DEST_ID
SFC72	I_GET	IOID、LADDR、VAR_ADDR
SFC73	I_PUT	IOID、LADDR、VAR_ADDR
SFC74	I_ABORT	IOID、LADDR

**响应中断**

用于未组态S7连接的通讯SFC可以被具有更高优先级的OB中断。如果带有完全相同作业的不同SFC被中断OB重新调用，则中止此第二次调用，并在RET\_VAL中输入一个相应的条目。然后继续执行中断SFC。

**访问CPU的工作存储器**

不管要传送的用户数据量有多大，操作系统的通讯函数以最大长度的数据域来访问CPU的工作存储器，因而不会因为通讯函数的使用而延长中断反应时间。

根据通过STEP 7设置的通讯引起的最大周期性负载，在作业的执行期间，操作系统的通讯函数可以多次访问工作存储器。

## 客户机切换到STOP

如果启动作业(并因此建立连接)的CPU在数据传送期间切换到STOP，则将终止由它所建立的所有连接。

## 更改程序

对于程序中所有可以立即对用于未组态S7连接的通讯SFC调用产生影响的部分，只能在处于STOP模式时对此进行修改。特别地，这包括删除包含用于未组态S7连接的通讯SFC调用的FC、FB或OB。

在修改程序之后，必须执行暖重启或冷重启。

不遵守这些规则，可能会使资源保持已分配状态，以及可编程控制器随后进入未定义的状态。





# 19 S7通讯

## 19.1 用于S7通讯的SFB/FB和SFC/FC的公用参数

### 分类

根据功能，用于组态S7连接的通讯SFB/FB的参数可分为以下五个类别：

1. 控制参数，用于激活块。
2. 寻址参数，用于寻址远程通讯伙伴。
3. 发送参数，指向将要发送到远程伙伴的数据区。
4. 接收参数，指向用于输入从远程伙伴接收到的数据的数据区。
5. 状态参数，用于监视块是否已经无错地完成了它的任务，或用于分析任何已经发生的出错。

### 控制参数

只有当调用SFB/FB时，相应的控制参数具有已定义的信号状态(例如被置位)，或自从前一次SFB/FB调用之后，信号状态发生了特定变化(例如，上升沿)时，数据交换才会被激活。

---

#### S7-300的注意事项

对于第一次调用，需要将参数REQ设置为FALSE。

---

## 寻址参数

参数	描述
ID	指向本地连接描述(由STEP 7连接组态指定)。
R_ID	<p>使用R_ID参数来指定一对发送SFB和接收SFB: 发送端SFB/FB的R_ID参数必须和接收端SFB的R_ID参数相同。</p> <p>这允许几对SFB/FB通过相同的逻辑连接进行通讯。</p> <ul style="list-style-type: none"> <li>• R_ID必须以DW#16#wxyzWXYZ的形式指定。</li> <li>• 通过R_ID指定的一个逻辑连接的几个程序块对, 对于这个连接来说必须是唯一的。</li> </ul>

参数PI\_NAME只在相关SFB中进行描述(只用于S7-400)。

### 注意

**S7-300:** 当寻址参数ID和R ID已激活时, 可以改变这些参数。在前一个作业关闭之后, 当新建一个作业时新参数开始生效。在此, 可以在一个实例中链接多个FB对。

**提示:** 下列一些方法可以用于保存背景数据块, 进而保存工作存储器:

1. 通过变量ID, 可以通过一个数据实例块来使用多个连接。
2. 通过变量R\_ID, 可以为一个作业定义多个发送和接收FB对的标识符。
3. 方法1和方法2可以同时使用。
4. 请注意, 新参数在最后一个作业执行完毕之后才开始生效。如果激活了发送作业, 则发送和接收FB的R\_ID参数必须相互匹配。

**S7-400:** 寻址参数ID和R ID只在第一次调用块时计算(实际参数或来自实例的预定义值)。因此, 第一次调用块时所指定的与远程伙伴的通讯关系(连接)会持续到下一次暖重启或冷重启。

## 状态参数

通过状态参数, 可以监视块是否已经正确地完成了它的任务, 或块是否还处于工作状态。状态参数还可以用于指示是否出错。

### 注意

状态参数只在一个周期内有效, 即从SFB/FB调用之后的第一个指令开始到下一个SFB/FB调用为止。所以, 必须在每个块周期之后评估这些参数。

## 发送和接收参数

如果没有使用SFB/FB的所有发送或接收参数，则第一个未使用的参数必须是NIL指针(参见/232/), 并且已使用的参数必须逐个连续放置，不能有任何间隔。

---

### S7-400的注意事项

在第一次调用期间，ANY指针指定可以为此作业传送的最大用户数据量。即，在CPU的工作存储器中创建一个通讯缓冲区，以确保数据的一致性。这个缓冲区最多占用480个字节的工作存储器。如果当CPU处于RUN模式时，没有通过SFB调用重新装载块，则建议在暖重启或冷重启OB中运行第一次调用。

在后来的调用中可以发送/接收任意数量的数据，但是不能超过第一次调用所发送/接收的数据量。

本规则的例外情况是BSEND和BRCV系统功能块。通过这两个功能块，每个作业可以发送多达64 K字节的数据(参见通过SFB/FB 12“BSEND”发送块相关数据和通过SFB/FB 13“BRCV”接收块相关数据)。

---

对于用于双边通讯的SFB/FB:

- 在发送端和接收端，所使用的SD\_i和RD\_i参数的数目必须匹配。
- 在发送端和接收端，属于一对的SD\_i和DR\_i参数的数据类型必须匹配。
- 根据SD\_i参数发送的数据量不得超出相应RD\_i参数指定的可用区域(不适用于BSEND/BRCV)。

ERROR = 1和STATUS = 4表示用户违反了上面的规则。

此最大用户数据长度取决于远程伙伴是S7-300还是S7-400。

## 用户数据大小

通过SFB/FB、USEND、URCV、GET和PUT，要发送的数据量不得超过最大用户数据长度。最大用户数据大小取决于：

- 所使用的块类型和
- 通讯伙伴。

下面的表格列出了使用带1-4个变量的SFB/FB时，可保证的最小用户数据大小：

块	伙伴：S7-300/C7-300	伙伴：S7-400/M7 M7到M7
PUT/GET	160字节	400字节
USEND/URCV	160字节	440字节
BSEND/BRCV	32768字节	65534字节

关于用户数据大小的更多信息，请参见各CPU技术数据。

## 精确的用户数据大小

如果上面指定的用户数据大小不够，则可以如下确定用户数据的最大字节长度：

1. 首先，从下面的表格中读取有效的通讯数据块大小：

本地CPU	远程CPU	数据块大小，以字节为单位
S7-300	ANY	240(S7-300)
S7-400	S7-300/C7-300	240(S7-400)
S7-400	S7-400或CPU 318	480
S7-400	M7模块	480
M7模块	M7模块	960

2. 使用下面表格中的数值来读取最大可能的用户数据长度，以字节为单位。  
此表适用于SD<sub>i</sub>、RD<sub>i</sub>和ADDR<sub>i</sub>区域的偶数长度。

数据块大小	SFB/FB	所使用的参数SD <sub>i</sub> 、RD <sub>i</sub> 和ADDR <sub>i</sub> 的数目			
		1	2	3	4
240(S7-300)	PUT/GET/ USEND	160	-	-	-
240(S7-300, 通过集成接口)	PUT	212	-	-	-
	GET	222	-	-	-
	USEND	212	-	-	-
240(S7-400)	PUT	212	196	180	164
	GET	222	218	214	210
	USEND	212	-	-	-
480	PUT	452	436	420	404
	GET	462	458	454	450
	USEND	452	448	444	440
960	PUT	932	916	900	884
	GET	942	938	934	930
	USEND	932	928	924	920

## 19.2 用于组态的S7连接的SFB启动例行程序

### 必要条件

在S7-400的下列描述中，假设下列条件成立：

- 在模块上存在连接描述(SDB)。
- 所组态的连接已经建立。
- ID的实际参数与为每个SFB组态的连接ID匹配。

### 暖重启和冷重启

在暖重启和冷重启期间，所有SFB都设置为NO\_INIT状态。存储在背景数据块中的实际参数不会改变。

### 通过SFB执行暖重启和冷重启，以实现双边数据交换

通常，用于双边数据交换的带有SFB的两个模块不会同时暖重启或冷重启。SFB的反应受下列规则的控制：

接收块(SFB URVCV、BRVCV)的反应如下：

- 如果在暖重启或冷重启时，SFB已经接收了一个作业，但是还没有确认这个作业，则SFB将产生一个序列中止帧(CFB、BRVCV)，并立即跳转到NO\_INIT状态。
- 通过SFB BRVCV，即使已经发送了序列中止，仍然可以接收其他数据段。将在本地丢弃该数据帧。
- SFB URVCV立即切换到NO\_INIT状态。

发送块(SFB USEND、BSEND)的反应如下：

- 如果SFB BSEND已经启动了一个还没有完成的作业序列，则当启动暖重启或冷重启时，它将发送一个序列中止。然后立即跳转到NO\_INIT状态。将在本地丢弃此后到达的应答。
- 如果当请求暖重启或冷重启时，SFB BSEND已经发送或接收了一个序列中止，则它立即切换到NO\_INIT状态。
- 在所有其它情形中，只要SFB只发送消息(例如，SFB USEND)，则将中止本地处理，且SFB立即跳转到NO\_INIT状态。

### 通过SFB执行暖重启和冷重启，以实现单边数据交换

可以认为，在连接建立之后，通讯伙伴上的服务器是运行的，即，服务器可以在任何时刻处理作业或输出消息。

发送作业并期待应答的SFB对完全重启的反应如下：

当前处理被中止，然后CFB立即跳转到NO\_INIT状态。如果在暖重启或冷重启之前发送的作业应答随后到达，则将在本地丢弃该应答。

在前面作业的应答接收到之前，新作业可能已经发送。

输出或接收消息的SFB反应如下：

- 中止当前处理，然后CFB立即跳转到NO\_INIT状态。
- 通过SFB USTATUS，在本地丢弃在NO\_INIT和DISABLED状态期间到达的消息。

### 对热重启的反应

只有在暖重启或冷重启期间，用于S7通讯的SFB才被设置为NO\_INIT状态。也就是说，它们的反应和可以在热重启之后重新恢复运行的用户功能块一样。

### 对存储器复位的反应

存储器复位将终止所有连接。由于在存储器复位之后，对于用户程序来说，暖重启或冷重启是唯一可以运行的启动类型，所以所有用于S7通讯的SFB (如果仍然存在) 都被设置为NO\_INIT状态并被初始化。作为对连接终止的反应，存储器没有复位的模块中的伙伴块将切换到IDLE、ENABLED或DISABLED状态。

## 19.3 SFB对故障如何反应

下面讲述了S7-400中用于S7通讯的SFB对故障如何反应。

### 连接被终止

监视分配给SFB实例的连接。

如果终止连接，则SFB的反应取决于它的内部状态。

如果在块处于IDLE或ENABLED状态时检测到连接断开，则SFB反应如下：

- SFB跳转到ERROR状态，并在ERROR和STATUS输出参数上输出出错ID“通讯故障”。
- 当下一次调用时，功能块返回到其初始状态，并重新检查此连接。

不处于IDLE或DISABLED状态的通讯SFB反应如下：

- SFB中止处理，立即或在下一个块调用时切换到ERROR状态，并在ERROR和STATUS输出参数上输出出错ID“通讯故障”。
- 当下一次调用SFB时，功能块切换到IDLE、DISABLED或ENABLED状态。在IDLE和ENABLED状态中重新检查连接。

如果在此期间重新建立连接，则也会执行此过程。

### 电源掉电

带备用电池的电源掉电，随后重新启动，这将终止所有已建立的连接。因此，上面讲到的各点适用于所有有关的功能块。

如果发生带备用电池的电源掉电，随后自动暖重启或冷重启，则终止连接和暖重启或冷重启的各个相关事项都适用。

在无备用电池的自动暖重启或冷重启特例中，电源恢复之后自动执行存储器复位，用于S7通讯的SFB的反应可参见“用于S7通讯的SFB启动例行程序”一节中的描述。

### 对操作模式切换的反应

如果操作模式在STOP、START、RUN和HOLD状态之间切换，则通讯SFB保持其当前状态(例外：在暖重启或冷重启期间，它将切换到NO\_INIT状态)。这既适用于单边通讯SFB，也适用于双边通讯SFB。

### 到用户程序的出错接口

如果在通讯SFB的处理期间出错，则它将切换到ERROR状态。在ERROR输出参数置位为1的同时，将相应出错ID输入到STATUS输出参数。可以在用户程序中判断此出错信息。

可能发生的出错实例：

- 收集发送数据时出错。
- 复制接收数据到接收区时出错(例如，尝试访问一个并不存在的DB)。
- 要发送的数据区的长度和在伙伴SFB中指定的接收区的长度不匹配。



## 19.4 通过SFB8/FB8“USEND”进行无协调的数据发送

### 描述

SFB/FB “USEND”向类型为“URCV”的远程伙伴SFB/FB发送数据。执行发送过程而不需要和SFB/FB伙伴进行协调。也就是说，在进行数据传送时不需要伙伴SFB/FB进行确认。

**S7-300:** 在REQ的上升沿处发送数据。在REQ的每个上升沿处传送参数R\_ID、ID和SD\_1。在每个作业结束之后，可以给R\_ID、ID和SD\_1参数分配新数值。

**S7-400:** 在控制输入REQ的上升沿处发送数据。通过参数SD\_1到SD\_4来指向要发送的数据，但并非都需要用到所有四个发送参数。

然而，必须确保参数SD\_1到SD\_4/SD\_1和RD\_1到RD\_4/RD\_1(在相应通讯伙伴SFB/FB“URCV”上)所定义的区域在以下几个方面保持一致：

- 编号
- 长度，和
- 数据类型。

参数R\_ID必须在两个SFB中完全相同。

如果传送成功完成，则通过状态参数DONE来表示，此时其逻辑数值为1。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求，在上升沿激活数据交换
ID	INPUT	WORD	M、D、常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
R_ID	INPUT	DWORD	I、Q、M、D、L、常数	寻址参数R_ID，参见用于S7通讯的SFB和SFC的公用参数
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数： 0：作业还未启动或仍然在运行 1：作业已经无错地执行完毕

参数	声明	数据类型	存储区域	描述
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0 STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS 提供详细信息 ERROR = 1 出错。 STATUS提供关于出错类型的详细信息。
S7-300: SD_1  S7-400: SD_i (1 i 4)	IN_OUT	ANY	M、D、T、Z  I、Q、M、D、T、C	指针，指向第i个发送数据区。 只允许使用下列数据类型：BOOL(不允许： 位域)、BYTE、CHAR、WORD、INT、 DWORD、DINT、REAL、DATE、TOD、 TIME、S5TIME、DATE_AND_TIME、 COUNTER、TIMER。 <b>注意：</b> 如果使用ANY指针访问一个数据块，则必须要指 定该数据块(例如：P# DB10.DBX5.0 Byte 10)。

## 出错信息

ERROR	STATUS (十进制)	说明
0	11	警告： <ul style="list-style-type: none"> <li>由于前一个作业还没有结束，所以不能执行新作业。</li> <li>正在以较低优先级处理此作业。。</li> </ul>
0	25	通讯已经启动。作业正在处理。
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>没有装载连接描述(本地或远程)</li> <li>连接被中断(例如：电缆断线、CPU关闭、或CP处于STOP模式)</li> <li>没有建立到通讯伙伴的连接</li> </ul>
1	4	发送数据区指针SD_i中的出错，包括数据长度或数据类型。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)。
1	12	当调用SFB时 <ul style="list-style-type: none"> <li>指定了一个不属于SFB“USEND”的背景数据块</li> <li>没有指定任何背景数据块，而指定了全局数据块。</li> <li>没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>
1	18	在连接ID中已经存在R_ID。

ERROR	STATUS (十进制)	说明
1	20	<ul style="list-style-type: none"> <li>• S7-400: 没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用, 则解决方案是压缩工作存储器。</li> <li>• H系统: 在更新期间不能进行SFB的第一次调用, 因为正在创建工作存储器中的通讯缓冲区。同样在更新期间不能改变工作存储器。</li> <li>• S7-300: <ul style="list-style-type: none"> <li>- 超出并行作业/实例的最大数目</li> <li>- CPU在RUN模式时, 实例超载。</li> <li>- 当第一次调用时可以</li> </ul> </li> </ul>
1	27	CPU中没有用于此功能块的功能代码。

### 数据的一致性

**S7-300:** 为确保数据的一致性, 只能在当前发送操作结束之后, 重新向发送区SD\_1写入数据。状态参数DONE的数值变为1时就是这种情形。

**S7-400和S7-300通过集成接口:** 当激活一个发送操作时(REQ的上升沿), 从用户程序中复制要从发送区SD\_i发送的数据。在块调用之后, 可以在不破坏当前发送数据的前提下向这些区域写入数据。

---

### 注意

只有当DONE状态参数数值为1时, 发送操作才完成。

---

## 19.5 通过SFB/FB9“URCV”进行无协调的数据接收

### 描述

SFB/FB9 “URCV”从类型为“USEND”的远程伙伴SFB/FB中异步接收数据，并把接收到的数据复制到组态的接收区域内。

当程序块准备好接收数据时，EN\_R输入处的逻辑值为1。可以通过EN\_R=0来取消一个已激活的作业。

**S7-300:** 在EN\_R的每个上升沿处应用参数R\_ID、ID和RD\_1。在每个作业结束之后，可以给R\_ID、ID和RD\_1参数分配新数值。

**S7-400:** 通过参数RD\_1到RD\_4来指向接收数据区。

但是，必须确保参数RD\_i/RD\_1和SD\_i/SD\_1(在相应通讯伙伴SFB/FB“USEND”上)所定义的区域在以下几个方面保持一致：

- 编号
- 长度，和
- 数据类型。

通过NDR状态参数逻辑数值为1来指示已经成功完成复制处理过程。

参数R\_ID必须在两个SFB/FB上完全相同。

参数	声明	数据类型	存储区域	描述
EN_R	INPUT	BOOL	I、Q、M、D、L、 常数	经激活可进行接收的控制参数在输入置位时发出信号，通知该对象准备接收。
ID	INPUT	WORD	M、D、常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
R_ID	INPUT	DWORD	I、Q、M、D、L、 常数	寻址参数R_ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
NDR	OUTPUT	BOOL	I、Q、M、D、L	NDR状态参数： 0：作业还未启动或仍然在运行 1：作业已经无错地执行完毕

参数	声明	数据类型	存储区域	描述
ERROR	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0 STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS 提供详细信息 ERROR = 1 出错。 STATUS提供关于出错类型的详细 信息。
S7-300: RD_1  S7-400: RD_i (1 i 4)	IN_OUT	ANY	M、D、T、Z  I、Q、M、D、T、 Z	指针，指向第i个接收数据区： 只允许使用下列数据类型：BOOL(不允许： 位域)、BYTE、CHAR、WORD、INT、 DWORD、DINT、REAL、DATE、TOD、 TIME、S5TIME、DATE_AND_TIME、 COUNTER、TIMER。 <b>注意：</b> 如果使用ANY指针访问一个数据块，则必须要 指定该数据块(例如：P# DB10.DBX5.0 Byte 10)。

## 出错信息

ERROR	STATUS (十进制)	说明
0	9	超限警告：较旧的接收数据被新的接收数据覆盖。
0	11	警告：正在以较低优先级处理接收数据。
0	25	通讯已经启动。作业正在处理。
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>没有装载连接描述(本地或远程)</li> <li>连接被中断(例如：电缆断线、CPU关闭、或CP处于STOP模式)</li> <li>没有建立到通讯伙伴的连接</li> </ul>
1	4	接收区指针RD_i中出错，包括数据长度或数据类型。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)
1	12	当调用CFB时 <ul style="list-style-type: none"> <li>指定了一个不属于SFB“URCV”的背景数据块</li> <li>没有指定任何背景数据块，而指定了一个全局数据块。</li> <li>没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>
1	18	在连接ID中已经存在R_ID。
1	19	相应的SFB/FB“USEND”正在发送数据，其速度快于SFB/FB“URCV”能够将它们复制到接收区的速度。

ERROR	STATUS (十进制)	说明
1	20	<ul style="list-style-type: none"> <li>• S7-400: 没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用, 则解决方案是压缩工作存储器。</li> <li>• H系统: 在更新期间不能进行SFB的第一次调用, 因为正在创建工作存储器中的通讯缓冲区。同样地, 在更新期间不能改变工作存储器。</li> <li>• S7-300: <ul style="list-style-type: none"> <li>- 超出并行作业/实例的最大数目</li> <li>- CPU在RUN模式时, 实例超载。</li> <li>- 当第一次调用时可以</li> </ul> </li> </ul>
1	27	CPU中没有用于此功能块的功能代码。

### 数据的一致性

如果遵守下列要点, 则可以保证数据接收的一致性:

- S7-300: 在状态参数NDR数值变化为1之后, 必须立即重新调用FB 9“URCV”(EN\_R为0)。这可以保证在计算完接收到的数据之前, 接收区不会被覆盖。
- 在调用块(控制输入EN\_R取值1)之前, 完成评估接收区(RD\_1)。

S7-400: 在状态参数NDR数值变化为1之后, 在接收区(RD\_i)中有新的接收数据。新的块调用可能引起这些数据被新的接收数据覆盖。如果要防止这种情况发生, 则必须调用SFB 9“URCV”(例如通过周期性的块处理)(EN\_R的数值为0), 一直到完成对接收数据的处理为止。

## 19.6 通过SFB/FB12 “BSEND”发送分段数据

### 描述

SFB/FB12“BSEND”向类型为“BRCV”的远程伙伴SFB/FB发送数据。相对于通过所有其他类型的SFB/FB通讯，通过这种类型的数据传送，可以在通讯伙伴之间为所组态的S7连接传输更多的数据，即可以为S7-300发送多达32768个字节，为S7-400发送多达65534个字节，以及通过集成接口为S7-300发送多达65534个字节的数据。

要发送的数据区是分段的。各个分段单独发送给通讯伙伴。通讯伙伴在接收到最后一个分段时对此分段进行确认，该过程与相应SFB/FB“BRCV”的调用无关。

**S7-300:** 在REQ的上升沿处激活发送作业。在REQ的每个上升沿处传送参数R\_ID、ID、SD\_1和LEN。在一个作业结束之后，可以给R\_ID、ID、SD\_1和LEN参数分配新的数值。为了进行分段数据的传送，必须在用户程序中周期性地调用块。

由SD\_1指定起始地址和要发送数据的最大长度。可以通过LEN来确定数据域的作业指定长度。

**S7-400和S7-300通过集成接口:** 在调用块之后，当在控制输入REQ上有上升沿时，发送作业被激活。发送用户存储区中的数据与处理用户程序是异步执行的。

由SD\_1指定起始地址和要发送数据的最大长度。可以通过LEN来确定数据域的作业指定长度。在这种情况下，LEN替换SD\_1的长度区域。

参数R\_ID必须在相应的两个SFB/FB上完全相同。

如果在控制输入R处有上升沿，则当前数据传送将被取消。

如果传送成功完成，则通过将状态参数DONE的数值设置为1来进行指示。

如果状态参数DONE或ERROR的数值为1，则在前一个发送处理结束之前，不能处理新的发送作业。

由于是异步数据传送，所以只有在通过调用伙伴SFB/FB而检索到前一个数据时，新传送才能启动。在数据被检索到之前，当调用SFB/FB“BSEND”时，将给出状态值7(参见下表)。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求，在上升沿激活数据交换。
R	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数复位，当仍然进行数据交换时，将在上升沿处激活中止的数据交换。
ID	INPUT	WORD	M、D、常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数

参数	声明	数据类型	存储区域	描述
R_ID	INPUT	DWORD	I、Q、M、D、L、 常数	寻址参数R_ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数使用通过CP 441到S5或辅助设备连接，R_ID包含了远程设备的地址信息。要了解更多信息，请参考CP 441的描述。
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数： 0：作业还未启动或仍然在运行 1：作业已经无错地执行完毕
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0 STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS 提供详细信息 ERROR = 1 出错。 STATUS提供关于出错类型的详细信息。
SD_1	IN_OUT	ANY	S7-300：M、D S7-400：I、Q、M、 D、T、Z	指针，指向发送区。 只允许使用下列数据类型：BOOL(不允许： 位域)、BYTE、CHAR、WORD、INT、 DWORD、DINT、REAL、DATE、TOD、 TIME、S5TIME、DATE_AND_TIME、 COUNTER、TIMER。 <b>注意：</b> 如果使用ANY指针访问一个数据块，则必须要 指定该数据块(例如：P# DB10.DBX5.0 Byte 10)。
LEN	IN_OUT	WORD	I、Q、M、D、L	要发送的数据域的长度，以字节为单位。



## 出错信息

下表包含了所有SFB/FB12特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	11	警告： <ul style="list-style-type: none"> <li>• 由于前一个作业还没有结束，所以不能执行新作业。</li> <li>• 正在以较低优先级处理此作业。</li> </ul>
0	25	通讯已经启动。作业正在处理。
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>• 没有装载连接描述(本地或远程)</li> <li>• 连接被中断(例如：电缆断线、CPU关闭、或CP处于STOP模式)</li> <li>• 没有建立到通讯伙伴的连接</li> </ul>
1	2	来自伙伴SFB/FB的否定应答。无法执行该功能。
1	3	R_ID在ID指定的连接上未知，或接收块还未调用。
1	4	发送区指针SD_1中出错，包括数据长度或数据类型，或者通过LEN传送数值0。
1	5	复位请求已执行。
1	6	伙伴SFB/FB处于DISABLED状态(EN_R数值为0)。同时还检查BRCV块的输入参数与BSEND块的一致性。
1	7	伙伴SFB/FB处于出错状态。 在最后一次数据传送之后，没有重新调用过接收块。
1	8	访问用户存储区中的远程对象被拒绝：用于相应SFB/FB13“BRCV”的目标区域太小。 相应的SFB/FB13“BRCV”报告ERROR = 1、STATUS = 4或ERROR = 1、STATUS = 10。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)。
1	12	当调用SFB时 <ul style="list-style-type: none"> <li>• 指定了一个不属于SFB12的背景数据块。</li> <li>• 没有指定任何背景数据块，而是指定了全局数据块。</li> <li>• 没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>
1	18	在连接ID中已经存在R_ID。
1	20	<ul style="list-style-type: none"> <li>• S7-400：没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用，则解决方案是压缩工作存储器。</li> <li>• H系统：在更新期间不能进行SFB的第一次调用，因为正在创建工作存储器中的通讯缓冲区。同样地，在更新期间不能改变工作存储器。</li> <li>• S7-300： <ul style="list-style-type: none"> <li>- 超出并行作业/实例的最大数目</li> <li>- CPU在RUN模式时，实例超载。</li> <li>- 当第一次调用时可以</li> </ul> </li> </ul>
1	27	CPU中没有用于此功能块的功能代码。

## 数据的一致性

为确保数据的一致性，只能在当前发送操作结束之后，重新向发送区SD\_i写入数据。状态参数DONE的数值变为1时就是这种情形。

## 19.7 通过SFB/FB13 “BRCV”接收分段数据

### 描述

SFB/FB13“BRCV”接收来自类型为“BSEND”的远程伙伴SFB/FB的数据。在收到每个数据段后，向伙伴SFB/FB发送一个确认帧，同时更新LEN参数。

在块调用完毕，并且在控制输入EN\_R数值为1之后，块准备接收数据。可以通过EN\_R=0来取消一个已激活的作业。

由RD\_1指定起始地址和接收区的最大长度。由LEN指示已接收数据域的长度。

**S7-300:** 在EN\_R的每个上升沿处应用参数R\_ID、ID和RD\_1。在每个作业结束之后，可以给R\_ID、ID和RD\_1参数分配新数值。为了进行分段数据的传送，必须在用户程序中循环地调用块。

**S7-400和S7-300通过集成接口:** 从用户存储区中接收数据与处理用户程序是异步执行的。

参数R\_ID必须在相应的两个SFB/FB上完全相同。

通过状态参数NDR的数值为1来指示所有数据段的无错接收。接收到的数据保持不变，直到通过EN\_R=1来重新调用SFB/FB13为止。

如果在数据的异步接收期间调用块，则将引发一个警告，该警告通过STATUS参数输出；如果当控制输入EN\_R数值为0时进行调用，则接收将被终止，并且SFB/FB将返回到它的初始状态。

参数	声明	数据类型	存储区域	描述
EN_R	INPUT	BOOL	I、Q、M、D、L、常数	经激活可进行接收的控制参数在输入置位时发出信号，通知该对象准备接收。
ID	INPUT	WORD	M、D、常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
R_ID	INPUT	DWORD	I、Q、M、D、L、常数	寻址参数R_ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数 使用通过CP 441到S5或辅助设备的连接，R_ID包含了远程设备的地址信息。要了解更多信息，请参考CP 441的描述。
NDR	OUTPUT	BOOL	I、Q、M、D、L	NDR状态参数： 0：作业还没有启动，或仍然处于激活状态 1：作业已成功完成。

参数	声明	数据类型	存储区域	描述
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0 STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS 提供详细信息 ERROR = 1 出错。 STATUS提供关于出错类型的详细 信息。
RD_1	IN_OUT	ANY	S7-300: M、D S7-400: I、Q、M、 D、T、C	指针，指向接收区。长度信息指定了要接收的 数据块的最大长度。  只允许使用下列数据类型：BOOL(不允许： 位域)、BYTE、CHAR、WORD、INT、 DWORD、DINT、REAL、DATE、TOD、 TIME、S5TIME、DATE_AND_TIME、 COUNTER、TIMER。 <b>注意：</b> 如果使用ANY指针访问一个数据块，则必须要 指定该数据块(例如：P# DB10.DBX5.0 Byte 10)。
LEN	IN_OUT	WORD	I、Q、M、D、L	已接收的数据长度，以字节为单位。

## 出错信息

下表包含了所有SFB/FB13特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	11	警告：正在以较低优先级处理接收数据。
0	17	警告：块异步接收数据。LEN参数显示已接收的数据量，以字节为单位。
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>没有装载连接描述(本地或远程)</li> <li>连接被中断(例如：电缆断线、CPU关闭、或CP处于STOP模式)</li> <li>没有建立到通讯伙伴的连接</li> </ul>
1	2	函数无法执行(协议出错)。
1	4	接收区指针RD_1中出错，包括数据长度或数据类型。发送数据域长于接收区。
1	5	复位已接收的请求，传送未完成。

ERROR	STATUS (十进制)	说明
1	8	相应SFB/FB12“BSEND”中发生访问出错。在发送完最后一个有效数据段之后，报告ERROR = 1和STATUS = 4或ERROR = 1和STATUS = 10。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)。
1	12	当调用SFB时 <ul style="list-style-type: none"> <li>指定了一个不属于SFB13的背景数据块。</li> <li>没有指定任何背景数据块，而是指定了全局数据块。</li> <li>没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>
1	18	在连接ID中已经存在R_ID。
1	20	<ul style="list-style-type: none"> <li><b>S7-400:</b> 没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用，则解决方案是压缩工作存储器。</li> <li><b>H系统:</b> 在更新期间不能进行SFB的第一次调用，因为正在创建工作存储器中的通讯缓冲区。同样地，在更新期间不能改变工作存储器。</li> <li><b>S7-300:</b> <ul style="list-style-type: none"> <li>超出并行作业/实例的最大数目</li> <li>CPU在RUN模式时，实例超载。</li> <li>当第一次调用时可以</li> </ul> </li> </ul>
1	27	CPU中没有用于此功能块的功能代码。

## 数据的一致性

如果遵守下列要点，则可以保证数据接收的一致性：

- 在再次调用块(控制输入EN\_R数值为1)之前，完成计算上次使用的接收区(RD\_1)的数值。

## 接收数据的特例(只用于S7-400)

如果在相应发送块发送第一个数据段之前，准备接受数据的带有BRCV块的接收CPU(即，已经在EN\_R = 1时进行了块调用)切换到STOP模式，则将发生下列事件：

- 接收CPU切换到STOP模式之后的第一个作业内的数据全部输入到接收区内。
- 伙伴SFB“BSEND”接收到一个肯定应答。
- 处于STOP模式的接收CPU不再能接受任何附加的BSEND作业。
- 只要CPU保持在STOP模式，NDR和LEN数值就都为0。

为了防止丢失关于接收数据的信息，必须在接收CPU上执行热重启，并以EN\_R = 1来调用SFB13“BRCV”。

## 19.8 通过SFB/FB15“PUT”向远程CPU写入数据

### 描述

通过使用SFB/FB15“PUT”，可以将数据写入到远程CPU。

**S7-300:** 在REQ的上升沿处发送数据。在REQ的每个上升沿处传送参数ID、ADDR\_1和SD\_1。在每个作业结束之后，可以给ID、ADDR\_1和SD\_1参数分配新数值。

**S7-400:** 在控制输入REQ的上升沿处启动SFB。在此过程中，将指向要写入数据的区域(ADDR\_i)的指针和数据(SD\_i)发送到伙伴CPU。

远程伙伴将所需要的数据保存在随数据一起提供的地址下面，并返回一个执行确认。

必须要确保通过参数ADDR\_i和SD\_i定义的区域在编号、长度和数据类型方面相互匹配。

如果没有产生任何出错，则在下一个SFB/FB调用时，通过状态参数DONE来指示，其数值为1。

只有在最后一个作业完成之后，才能再次激活写作业。

远程CPU可以处于RUN或STOP工作状态。

如果正在写入数据时发生访问故障，或如果执行检查过程中出错，则出错和警告信息将通过ERROR和STATUS输出表示。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求，在上升沿激活数据交换。
ID	INPUT	WORD	M、D、常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数： 0：作业还未启动或仍然处于激活状态 1：作业已经无错地执行完毕
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0 STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS提供详细信息。 ERROR = 1 出错。 STATUS提供关于出错类型的详细信息。

参数	声明	数据类型	存储区域	描述
S7-300: ADDR_1	IN_OUT	ANY	M、D	指针，指向伙伴CPU中要用于写入数据的区域。
S7-400: ADDR_i (1 i 4)			I、Q、M、D、T、C	
S7-300: SD_1	IN_OUT	ANY	S7-300: M、D S7-400: I、Q、M、 D、T、C	<p>指针，指向本地CPU中包含了要发送的数据的区域。</p> <p>只允许使用下列数据类型：BOOL(不允许：位域)、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、COUNTER、TIMER。</p> <p><b>注意：</b> 如果使用ANY指针访问一个数据块，则必须要指定该数据块 (例如：P# DB10.DBX5.0 Byte 10)。</p>
S7-400: SD_i (1 i 4)				

## 出错信息

下表包含了所有SFB/FB14特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	11	警告： <ul style="list-style-type: none"> <li>由于前一个作业还没有结束，所以不能执行新作业。</li> <li>正在以较低优先级处理此作业。</li> </ul>
0	25	通讯已经启动。作业正在处理。
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>没有装载连接描述(本地或远程)</li> <li>连接被中断(例如：电缆断线、CPU关闭、或CP处于STOP模式)</li> <li>没有建立到通讯伙伴的连接</li> </ul>
1	2	来自伙伴设备的否定应答。无法执行该函数。
1	4	发送区指针SD_i中出错，包括数据长度或数据类型。
1	8	在伙伴CPU上发生访问出错。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)
1	12	当调用SFB时， <ul style="list-style-type: none"> <li>指定了一个不属于SFB15的背景数据块。</li> <li>没有指定任何背景数据块，而是指定了一个共享数据块。</li> <li>没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>

ERROR	STATUS (十进制)	说明
1	20	<ul style="list-style-type: none"> <li>• S7-400: 没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用, 则解决方案是压缩工作存储器。</li> <li>• H系统: 在更新期间不能进行SFB的第一次调用, 因为正在创建工作存储器中的通讯缓冲区。同样地, 在更新期间不能改变工作存储器。</li> </ul> <p>S7-300:</p> <ul style="list-style-type: none"> <li>- 超出并行作业/实例的最大数目</li> <li>- CPU在RUN模式时, 实例超载。</li> <li>- 当第一次调用时可以</li> </ul>
1	27	CPU中没有用于此功能块的功能代码。

### S7-300的数据的一致性

为了确保数据的一致性, 在当前发送处理结束之前, 不能重新向发送区SD\_I写入数据。状态参数DONE数值为1时便是这种情况。

### S7-400和S7-300通过集成接口的数据的一致性:

当激活一个发送操作时(在REQ的上升沿), 从用户程序中复制要从发送区SD\_i发送的数据。在块调用之后, 可以在不破坏当前发送数据的前提下向这些区域写入数据。

---

#### 注意

只有当DONE状态参数数值为1时, 发送操作才完成。

---

## 19.9 通过SFB/FB14“GET”从远程CPU中读取数据

### 描述

可以通过SFB/FB14“GET”，从远程CPU中读取数据。

**S7-300:** 在REQ的上升沿处读取数据。在REQ的每个上升沿处传送参数ID、ADDR\_1和RD\_1。在每个作业结束之后，可以分配新数值给ID、ADDR\_1和RD\_1参数。

**S7-400:** 在控制输入REQ的上升沿处启动SFB。在此过程中，要读取的区域的相关指针(ADDR\_i)被发送到伙伴CPU。

远程伙伴返回此数据。

在下一个SFB/FB调用处，已接收的数据被复制到组态的接收区(RD\_i)中。

必须要确保通过参数ADDR\_i和RD\_i定义的区域在长度和数据类型方面要相互匹配。

通过状态参数NDR数值1来指示此作业已完成。

只有在前一个作业已经完成之后，才能重新激活读作业。

远程CPU可以处于工作状态RUN或STOP。

如果正在读取数据时发生访问故障，或如果数据类型检查过程中出错，则出错和警告将通过ERROR和STATUS输出。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求，在上升沿激活数据交换。
ID	INPUT	WORD	IM、D、常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
NDR	OUTPUT	BOOL	I、Q、M、D、L	NDR状态参数： 0：作业还未启动或仍然处于激活。 1：作业已成功完成。
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0 STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS提供详细信息。 ERROR = 1 产生一个出错。STATUS给出了关于出错类型的详细信息。



参数	声明	数据类型	存储区域	描述
S7-300: ADDR_1	IN_OUT	ANY	S7-300: M、D	指针，指向伙伴CPU中要被读取的区域。
S7-400: ADDR_i (1 i 4)			S7-400: I、Q、M、D、T、C	
S7-300: RD_1	IN_OUT	ANY	S7-300: M、D	指针，指向本地CPU中用于输入读取数据的区域。 只允许使用下列数据类型： BOOL(不允许：位域)、BYTE、CHAR、WORD、 INT、DWORD、DINT、REAL、COUNTER、 TIMER。 <b>注意：</b> 如果使用ANY指针访问一个数据块，则必须要指定 该数据块(例如：P# DB10.DBX5.0 Byte 10)。
S7-400: RD_i (1 i 4)			S7-400: I、Q、M、D、T、C	

## 出错信息

下表包含了所有SFB/FB14出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	11	警告： <ul style="list-style-type: none"> <li>由于前一个作业还没有结束，所以不能执行新作业。</li> <li>正在以较低优先级处理此作业。</li> </ul>
0	25	通讯已经启动。作业正在处理。
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>没有装载连接描述(本地或远程)</li> <li>连接被中断(例如：电缆断线、CPU关闭、或CP处于STOP模式)</li> <li>没有建立到通讯伙伴的连接</li> </ul>
1	2	来自伙伴设备的否定应答。无法执行该功能。
1	4	接收区指针RD_i中出错，包括数据长度或数据类型。
1	8	在伙伴CPU上发生访问出错。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)
1	12	当调用SFB时， <ul style="list-style-type: none"> <li>指定了一个不属于SFB14的背景数据块。</li> <li>没有指定任何背景数据块，而是指定了一个共享数据块。</li> <li>没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>

ERROR	STATUS (十进制)	说明
1	20	<ul style="list-style-type: none"><li>• S7-400: 没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用, 则解决方案是压缩工作存储器。</li><li>• H系统: 在更新期间不能进行SFB的第一次调用, 因为正在创建工作存储器中的通讯缓冲区。同样地, 在更新期间不能改变工作存储器。</li></ul> S7-300: <ul style="list-style-type: none"><li>- 超出并行作业/实例的最大数目</li><li>- CPU在RUN模式时, 实例超载。</li><li>- 当第一次调用时可以</li></ul>
1	27	CPU中没有用于此功能块的功能代码。

### 数据的一致性

如果遵守了下面的要点, 则能保证数据接收处于一致性状态:

在启动其他作业之前, 评估当前正在使用的接收区RD\_i部分。

## 19.10 使用SFB16“PRINT”将数据发送到打印机

### 描述

SFB 16“PRINT”将数据和格式指令发送到远程打印机，例如，通过CP 441。

当在控制输入REQ上有上升沿时，格式描述(FORMAT)和数据(SD\_i)被发送到通过ID和PRN\_NR选择的打印机上。

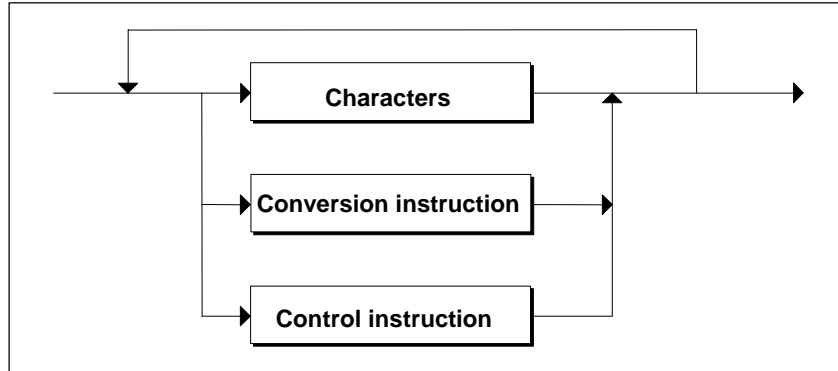
如果并未使用所有四个发送区，则必须确保用SD\_1参数描述第一个区域，用SD\_2参数描述第二个区域(如果存在)，用SD\_3描述第三个区域(如果存在)。

通过DONE状态参数来指示作业的成功执行，通过ERROR和STATUS参数来指示发生的出错。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求，在上升沿激活数据交换。
ID	INPUT	WORD	M、D、 常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数： 0：作业还未启动或仍然在运行 1：作业已经无错地执行完毕
ERROR STATE	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0和STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS提供 详细信息 ERROR = 1 出错。 STATUS提供关于出错类型的详细 信息。
PRN_NR	IN_OUT	BYTE	I、Q、M、D、L	打印机数目
格式	IN_OUT	STRING	I、Q、M、D、L	格式描述
SD_i (1≤i≤4)	IN_OUT	ANY	M、D、T、C	指针，指向“第i个”发送数据区。 只允许使用下列数据类型：BOOL(不允许： 位域)、BYTE、CHAR、WORD、INT、 DWORD、DINT、REAL、DATE、TOD、 TIME、S5TIME、DATE_AND_TIME、 COUNTER、TIMER。 <b>注意：</b> 如果使用ANY指针访问一个数据块，则必须 要指定该数据块(例如：P# DB10.DBX5.0 Byte 10)。

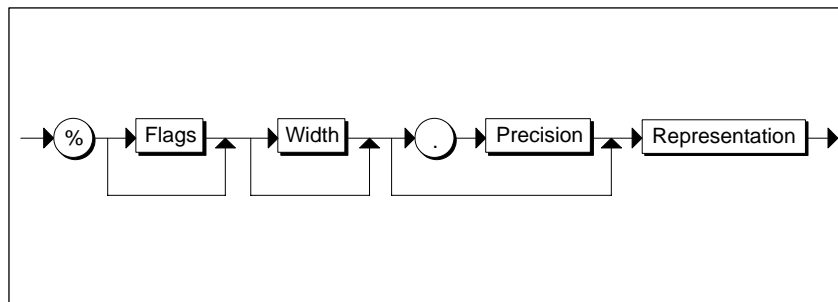
## In\_out参数FORMAT

FORMAT字符串包含了可打印的字符和格式元素。它具有如下结构：



对于每个要打印的发送区(SD\_1到SD\_4)，在FORMAT中必须要有一个转换指令。转换指令依据发送区格式化的顺序，依次应用于各个发送区(SD\_i)。字符和指令可以按照任意顺序一个接一个地排列。

- 字符  
可以使用下列一些字符：
  - 所有可打印的字符
  - \$\$ (美元字符)、\$(单引号)、\$L和\$I(换行)、\$P和\$P(页面)、\$R和\$r(回车)、\$T和\$t(制表键)



转换指令的语法图

转换指令的元素	含义
标记	<ul style="list-style-type: none"> <li>• 无： 右对齐输出</li> <li>• -: 左对齐输出</li> </ul>
宽度	<ul style="list-style-type: none"> <li>• 无： 以标准表达式输出</li> <li>• n: 精确地输出n个字符。如果输出是右对齐，则字符前面可能会有空格，而对于左对齐，则在字符后面输出空格。</li> </ul>

转换指令的元素	含义
精度	精度只与表达式A、D、F和R相关(参见下表)。 <ul style="list-style-type: none"> <li>• 无: 以标准表达式输出</li> <li>• 0: 在F和R表达式中没有小数点或小数位输出</li> <li>• n:               <ul style="list-style-type: none"> <li>- 对于F和R: 小数点和n个小数位的输出</li> <li>- 对于A和D(日期): 年的位数: 可以使用的数值为2和4。</li> </ul> </li> </ul>
表达式	下表包含了: <ul style="list-style-type: none"> <li>• 可以使用的表达式</li> <li>• 可以用于每个表达式的数据类型</li> <li>• 用于每个表达式的标准格式(如果没有在FORMAT参数中指定宽度和精度, 则打印输出使用标准表达式), 以及它们的最大长度</li> </ul>

下表给出了在FORMAT参数的转换指令中可以使用的表达式模式。

表达式	可用的数据类型	实例	长度	注释
A、a	DATE	25.07.1996	10	-
	DWORD			
C、c	CHAR	K	1	-
	BYTE	M	1	
	WORD	KL	2	
	DWORD	KLMN	4	
	ARRAY of CHAR	KLMNOP	字符数	
	ARRAY of BYTE			
D、d	DATE	1996-07-25	10	-
	DWORD			
F、f	REAL	0.345678	8	-
	DWORD			
H、h	所有数据类型, 包括 ARRAY of BYTE	根据数据类型	根据数据类型	十六进制表达式
I、i	INT	- 32 768	最大6	-
	WORD	- 2 147 483 648	最大11	

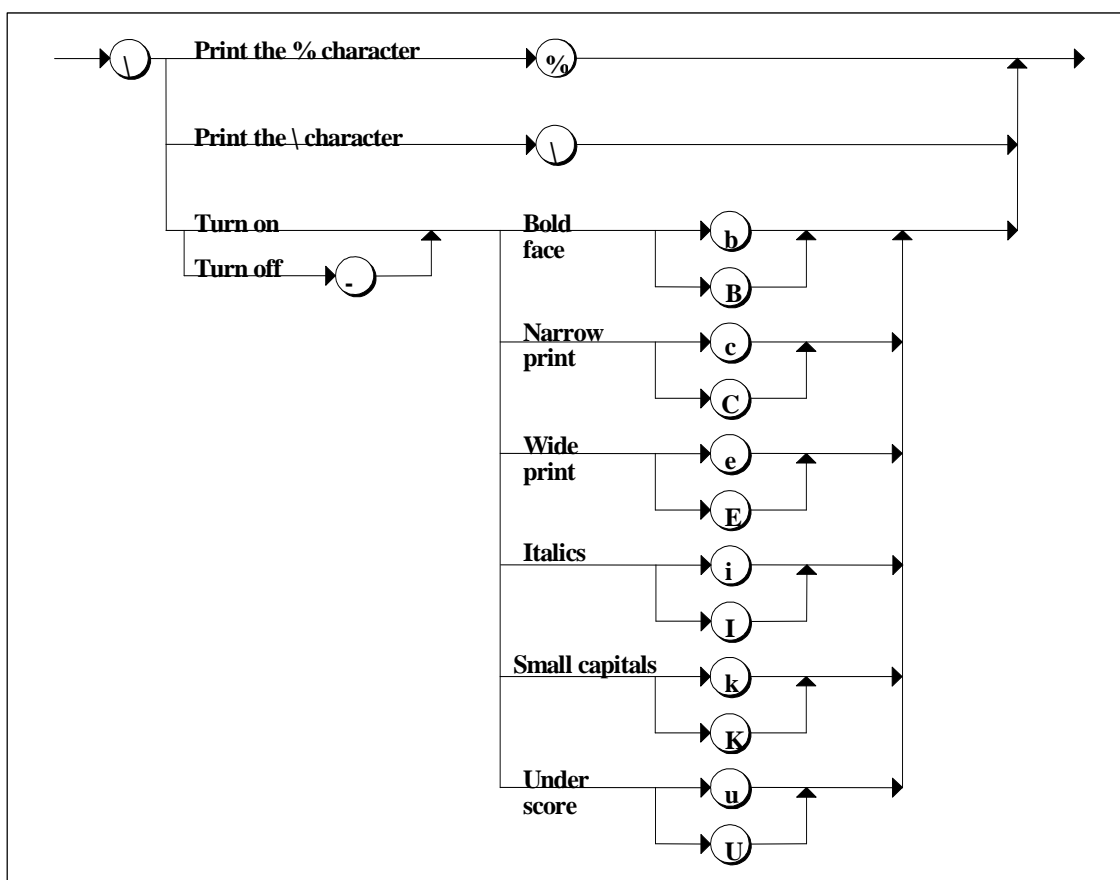
表达式	可用的数据类型	实例	长度	注释
N、n	WORD	文本输出	-	相应发送区SD_i包含了一个指向要打印文本的引用(编号)。文本在创建可打印字符串的模块(例如, CP 441)上。如果在指定的编号下找不到任何文本, 则输出*****。
R、r	REAL	0.12E-04	8	-
	DWORD			
S、s	STRING	文本输出		-
T、t	TIME	2d_3h_10m_5s_250ms	最大21	如果出错, 则输出*****。
	DWORD			
U、u	BYTE	255	最大3	-
	WORD	65 535	最大5	
	DWORD	4 294 967 295	最大10	
X、x	BOOL	1	1	-
	BYTE	101 ..	8	
	WORD	101 ..	16	
	DWORD	101 ..	32	
Z、z	TIME_OF_DAY(TOD)	15: 38: 59.874	12	-

此表中的某些数据, 是为表达式指定的最大长度, 所以实际长度可以更短一些。

### 注意

对于数据类型C和S, 以下几点取决于所使用的打印机:

- 可以打印哪些字符
  - 对于不能打印的字符, 打印机将打印什么, 除非打印机驱动程序对于这些字符有转换表。
- 
- 控制指令  
通过使用控制指令, 可以执行下列一些任务:
    - 打印字符%和\
    - 改变打印机设置。



控制指令的语法图

如果尝试进行某些工作，例如，禁止一种还没有启用的字体，或执行打印机无法识别的一个函数，则控制指令将被忽略。下表包含了使用FORMAT输入/输出参数时可能发生的出错。

出错	打印机输出
转换指令无法执行	根据缺省表达式的(最大)长度或指定宽度输出*字符。
指定的宽度太小	在表达式A、C、D、N、S、T和Z中，打印和所选择的宽度指定的字符数一样多的字符。对于所有其它表达式，打印*字符来补足指定的宽度。
转换指令太多	没有发送区指针SD_i的转换指令将被忽略。
转换指令太少	不打印没有转换指令的发送区。
未定义或不支持的转换指令	打印输出*****。
不完整的转换指令	打印输出*****。
未定义或不支持的转换指令	不符合上图中显示的语法的控制指令将被忽略。

## 出错信息

下表包含了所有SFB16“PRINT”特有的出错信息，这些信息可以通过参数ERROR和STATUS打印输出。

ERROR	STATUS (十进制)	说明
0	11	警告： <ul style="list-style-type: none"> <li>• 由于前一个作业还没有结束，所以不能执行新作业。</li> <li>• 正在以较低优先级处理此作业。</li> </ul>
0	25	通讯已经启动。作业正在处理。
1	1	通讯故障，例如 <ul style="list-style-type: none"> <li>• 没有装载连接描述(本地或远程)</li> <li>• 连接中断(例如，电缆断线、CPU关闭、CP处于STOP模式)</li> </ul>
1	2	来自打印机的否定应答。无法执行该功能。
1	3	在ID指定的通讯链接上不能识别PRN_NR。
1	4	在FORMAT输入/输出参数或在发送区指针SD_i中，在数据长度或数据类型方面出错。
1	6	远程打印机处于OFFLINE状态。
1	7	远程打印机不在正确状态(例如，缺纸)。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)。
1	13	在FORMAT输入/输出参数中出错
1	20	<ul style="list-style-type: none"> <li>• S7-400：没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用，则解决方案是压缩工作存储器。</li> <li>• H系统：在更新期间不能进行SFB的第一次调用，因为正在创建工作存储器中的通讯缓冲区。同样地，在更新期间不能改变工作存储器。</li> </ul>

## 可传送的数据量

可以传送到远程打印机的数据量不得超出最大长度。

此最大数据长度按照下面的公式来计算：

最大长度 = 420 - 格式

格式是FORMAT参数的当前长度，以字节为单位。要打印的数据可以分布在一个或者多个发送区。



## 19.11 通过SFB 19“START”在远程设备上开始一个暖重启或冷重启

### 描述

如果在SFB19“START”的控制输入REQ上有上升沿，则这将激活由ID寻址的远程设备上的暖重启或冷重启。如果远程系统是一个容错系统，则重启请求的结果就取决于参数PI\_NAME：启动请求或者仅对此系统中的一个CPU有效，或者对此系统中的所有CPU有效。如果远程设备是一个CPU，则必须满足下列条件：

- CPU必须处于STOP模式。
- CPU的钥匙开关必须设置为“RUN”或“RUN-P”。

一旦完成暖重启或冷重启，设备切换到RUN模式，并发送一个肯定执行应答。接收到肯定应答之后，状态参数DONE置位为1，如果发生任何出错，则通过状态参数ERROR和STATUS来指示这些出错。

只有在最近一次完全重启完成之后，才能在相同的远程设备上再次激活暖重启或冷重启。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求，在上升沿激活SFB
ID	INPUT	WORD	I、Q、M、D、 常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数： 0：作业还未启动或仍然在运行 1：作业已经无错地执行完毕
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0和STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS提供 详细信息  ERROR = 1 出错。 STATUS提供关于出错类型的详细 信息。

参数	声明	数据类型	存储区域	描述
PI_NAME	IN_OUT	ANY	I、Q、M、D、T、C	<p>指针，指向存储要启动的程序的名称(ASCII代码)的存储区域。此名称所包含的字符不能超过32个。</p> <p>对于S7 PLC，名称必须是P_PROGRAM。</p> <p>对于H系统，可以使用下列名称：</p> <ul style="list-style-type: none"> <li>• P_PROGRAM(启动作业对H系统中的所有CPU都有效)。</li> <li>• P_PROG_0(启动作业对H系统中机架0中的CPU有效)。</li> <li>• P_PROG_1(启动作业对H系统中机架1中的CPU有效)。</li> </ul>
ARG	IN_OUT	ANY	I、Q、M、D、T、C	<p>执行参数。</p> <ul style="list-style-type: none"> <li>• 如果没有给ARG分配数值，则在远程设备上执行暖重启。</li> <li>• 如果分配数值“C”，则在远程设备上执行冷重启(如果远程设备支持这种类型的启动)。</li> </ul>
IO_STATE	IN_OUT	BYTE	I、Q、M、D、L	<p>当前不相关。如果通讯伙伴是S7可编程控制器，则不给这个参数分配任何数值。。</p>

## 出错信息

下表包含了所有SFB19特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	11	<p>警告：</p> <ul style="list-style-type: none"> <li>• 由于前一个作业还没有结束，所以不能执行新作业。</li> <li>• 正在以较低优先级处理此作业。</li> </ul>
0	25	<p>通讯已经启动。作业正在处理。</p>
1	1	<p>通讯故障，例如：</p> <ul style="list-style-type: none"> <li>• 没有装载连接描述(本地或远程)</li> <li>• 连接中断(例如，电缆断线、CPU关闭、CP处于STOP模式)</li> </ul>
1	2	<p>来自伙伴设备的否定应答。无法执行该函数。</p>
1	3	<p>无法识别在PI_NAME中输入的程序名称。</p>
1	4	<p>指针PI_NAME或ARG出错，包括数据长度或数据类型。</p>
1	7	<p>无法在伙伴设备上执行完全重启。</p>
1	10	<p>不能访问本地用户存储区(例如，访问一个已经删除的数据块)</p>

ERROR	STATUS (十进制)	说明
1	12	当调用SFB时， <ul style="list-style-type: none"><li>指定了一个不属于SFB19的背景数据块。</li><li>没有指定任何背景数据块，而是指定了一个共享数据块。</li><li>没有找到任何背景数据块(从PG中装载新背景数据块)。</li></ul>
1	20	<ul style="list-style-type: none"><li><b>S7-400:</b> 没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用，则解决方案是压缩工作存储器。</li><li><b>H系统:</b> 在更新期间不能进行SFB的第一次调用，因为正在创建工作存储器中的通讯缓冲区。同样地，在更新期间不能改变工作存储器。</li></ul>

## 19.12 通过SFB 20“STOP”将远程设备切换到STOP状态

### 描述

如果在SFB20 “STOP”的控制输入REQ上有上升沿，则将由ID寻址的远程设备切换到STOP模式。当设备处于RUN、HOLD或STARTUP模式时，可以进行模式切换。

如果远程系统是一个容错系统，则重新启动请求的结果就取决于参数PI\_NAME：启动请求或者仅对此系统中的一个CPU有效，或者对此系统中的所有CPU有效。

通过状态参数DONE的数值为1来指示作业的成功执行。如果发生任何出错，则通过状态参数ERROR和STATUS来指示这些出错。

只有在前一个SFB20 调用结束之后，才能再次在相同远程设备上启动模式切换。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求，在上升沿激活SFB
ID	INPUT	WORD	I、Q、M、D、 常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数： 0：作业还未启动或仍然在运行 1：作业已经无错地执行完毕
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0和STATUS的数值： 0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS提供详细信息。 ERROR = 1 出错。 STATUS提供关于出错类型的详细信息。
PI_NAME	IN_OUT	ANY	I、Q、M、D	指针，指向存储要启动的程序的名称(ASCII代码)的存储区域。此名称所包含的字符不能超过32个。 对于S7 PLC，名称必须是P_PROGRAM。 对于H系统，可以使用下列名称： <ul style="list-style-type: none"> <li>• P_PROGRAM(启动作业对H系统中的所有CPU都有效)。</li> <li>• P_PROG_0(启动作业对H系统中机架0中的CPU有效)。</li> <li>• P_PROG_1(启动作业对H系统中机架1中的CPU有效)。</li> </ul>

参数	声明	数据类型	存储区域	描述
IO_STATE	IN_OUT	BYTE	I、Q、M、D、L	当前不相关。如果通讯伙伴是S7可编程控制器，则不给这个参数分配任何数值。

## 出错信息

下表包含了所有SFB20特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	11	警告： <ul style="list-style-type: none"> <li>• 由于前一个作业还没有结束，所以不能执行新作业。</li> <li>• 正在以较低优先级处理此作业。</li> </ul>
0	25	通讯已经启动。作业正在处理。
1	1	通讯故障，例如 <ul style="list-style-type: none"> <li>• 没有装载连接描述(本地或远程)</li> <li>• 连接中断(例如，电缆断线、CPU关闭、CP处于STOP模式)</li> </ul>
1	2	来自伙伴设备的否定应答。无法执行该函数。
1	3	无法识别在PI_NAME中输入的程序名称。
1	4	指针PI_NAME中出错，包括数据长度或数据类型。
1	7	伙伴设备已经处于STOP状态。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)
1	12	当调用SFB时， <ul style="list-style-type: none"> <li>• 指定了一个不属于SFB20的背景数据块。</li> <li>• 没有指定任何背景数据块，而是指定了一个共享数据块。</li> <li>• 没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>
1	20	<ul style="list-style-type: none"> <li>• S7-400: 没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用，则解决方案是压缩工作存储器。</li> <li>• H系统: 在更新期间不能进行SFB的第一次调用，因为正在创建工作存储器中的通讯缓冲区。同样地，在更新期间不能改变工作存储器。</li> </ul>

## 19.13 通过SFB 21“RESUME”在远程设备上开始一个热重启

### 描述

如果在SFB21“RESUME”的控制输入REQ上有上升沿，则将在通过ID选择的远程设备上激活一次热重启。

如果远程设备是一个CPU，则必须满足下列条件：

- CPU必须处于STOP模式。
- CPU的钥匙开关必须设置为“RUN”或“RUN-P”。
- 当通过STEP 7创建组态时，允许进行手动热重启。
- 必须没有阻止热重启的其他条件。

一旦完成热重启，设备将切换到RUN模式，并发送一个肯定执行应答。接收到肯定应答之后，状态参数DONE置位为1。所发生的出错由状态参数ERROR和STATUS指出。

只有在前一个热重启结束之后，才能在同一个远程设备上重新激活重启动。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求，在上升沿激活SFB
ID	INPUT	WORD	I、Q、M、D、 常数	寻址参数ID，参见用于S7通讯的SFB/FB和SF C/FC的公用参数
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数： 0：作业还未启动或仍然在运行 1：作业已经无错地执行完毕
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0和STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS 提供详细信息 ERROR = 1 出错。STATUS提供关于出错类型 的详细信息。
PI_NAME	IN_OUT	ANY	I、Q、M、D	指针，指向存储要启动的程序的名称(ASCII代码)的 存储区域。此名称所包含的字符不能超过 32个。对于S7 PLC，名称必须是P_PROGRAM。

参数	声明	数据类型	存储区域	描述
ARG	IN_OUT	ANY	I、Q、M、D、T、C	执行参数。当前不相关。如果通讯伙伴是S7可编程控制器，则不给这个参数分配任何数值。
IO_STATE	IN_OUT	BYTE	I、Q、M、D、L	当前不相关。如果通讯伙伴是S7可编程控制器，则不给这个参数分配任何数值。

## 出错信息

下表包含了所有SFB21特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	11	警告： <ul style="list-style-type: none"> <li>由于前一个作业还没有结束，所以不能执行新作业。</li> <li>正在以较低优先级处理此作业。</li> </ul>
0	25	通讯已经启动。作业正在处理。
1	1	<ul style="list-style-type: none"> <li>通讯故障，例如，没有装载连接描述(本地或远程)</li> <li>连接中断(例如，电缆断线、CPU关闭、CP处于STOP模式)</li> </ul>
1	2	来自伙伴设备的否定应答。无法执行该函数。
1	3	无法识别在PI_NAME中输入的程序名称。
1	4	指针PI_NAME或ARG出错，包括数据长度或数据类型。
1	7	不能执行热重启
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)
1	12	当调用SFB时， <ul style="list-style-type: none"> <li>指定了一个不属于SFB21的背景数据块。</li> <li>没有指定任何背景数据块，而是指定了一个共享数据块。</li> <li>没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>
1	20	<ul style="list-style-type: none"> <li><b>S7-400:</b> 没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用，则解决方案是压缩工作存储器。</li> <li><b>H系统:</b> 在更新期间不能进行SFB的第一次调用，因为正在创建工作存储器中的通讯缓冲区。同样地，在更新期间不能改变工作存储器。</li> </ul>

## 19.14 使用SFB 22“STATUS”查询远程伙伴的状态

### 描述

通过使用SFB22 “STATUS”，可以查询远程通讯伙伴的状态。

如果在控制输入REQ上有上升沿，则将作业发送到远程伙伴。判断回复，以确定是否产生了故障。如果没有产生任何出错，则通过下一个SFB调用将接收到的状态复制到变量PHYS、LOG和LOCAL。通过状态参数NDR的数值为1来指示作业完成。

只有最后一次查询结束之后，才能再次查询同一个通讯伙伴的状态。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求，在上升沿激活SFB
ID	INPUT	WORD	I、Q、M、D、 常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
NDR	OUTPUT	BOOL	I、Q、M、D、L	NDR状态参数： 0：作业还未启动或仍然在运行 1：作业已经无错地执行完毕
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0和STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H： 警告，STATUS提供详细信息。 ERROR = 1 出错。 STATUS提供关于出错类型的 详细信息。
PHYS	IN_OUT	ANY	I、Q、M、D	物理状态(最小长度：一字节)。 可能的值： • 10H 执行功能 • 13H 服务请求
LOG	IN_OUT	ANY	I、Q、M、D	逻辑状态(最小长度：一字节)。 可能的数值： • 00H 允许状态改变
LOCAL	IN_OUT	ANY	I、Q、M、D	如果伙伴设备是一个S7 CPU时的状态 (最小长度：两字节)



## 输入/输出参数LOCAL

如果通讯伙伴是一个S7 CPU，则输入/输出参数LOCAL包含它的当前状态。  
第一字节保留，第二字节包含状态的ID。

操作模式	相应标识符
STOP	00H
暖重启	01H
RUN	02H
热重启	03H
HOLD	04H
冷重启	06H
RUN_R	09H
LINK-UP	0BH
UPDATE	0CH

## 出错信息

下表包含了所有SFB22特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	11	警告： <ul style="list-style-type: none"> <li>由于前一个作业还没有结束，所以不能执行新作业。</li> <li>正在以较低优先级处理此作业。</li> </ul>
0	25	通讯已经启动。作业正在处理。
1	1	通讯故障，例如 <ul style="list-style-type: none"> <li>没有装载连接描述(本地或远程)</li> <li>连接中断(例如，电缆断线、CPU关闭、CP处于STOP模式)</li> </ul>
1	2	来自伙伴设备的否定应答。无法执行该函数。
1	4	PHYS、LOG或LOCAL中出错，包括数据长度或数据类型。
1	8	访问远程对象被拒绝。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)。
1	12	当调用SFB时， <ul style="list-style-type: none"> <li>指定了一个不属于SFB22的背景数据块。</li> <li>没有指定任何背景数据块，而是指定了一个共享数据块。</li> <li>没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>
1	20	<ul style="list-style-type: none"> <li><b>S7-400</b>：没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用，则解决方案是压缩工作存储器。</li> <li><b>H系统</b>：在更新期间不能进行SFB的第一次调用，因为正在创建工作存储器中的通讯缓冲区。同样地，在更新期间不能改变工作存储器。</li> </ul>

## 19.15 通过SFB23 “USTATUS”接收远程设备的状态改变

### 描述

SFB23 “USTATUS”接收远程通讯伙伴的设备状态改变。如果在STEP 7中进行了相应组态，当发生状态改变时，通讯伙伴主动发送它的状态。

当调用CFB并且有来自通讯伙伴的数据帧时，如果控制输入EN\_R数值为1，则在下一次调用SFB时，将状态信息输入到变量PHYS、LOG和LOCAL中。通过状态参数NDR的数值为1来指示作业完成。

必须在USTATUS所使用的连接上允许进行工作状态消息传送。

### 注意

每个连接只能使用一个SFB23 实例。

参数	声明	数据类型	存储区域	描述
EN_R	INPUT	BOOL	I、Q、M、D、L	经激活可进行接收的控制参数在输入置位时发出信号，通知该对象准备接收。
ID	INPUT	WORD	I、Q、M、D、常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
NDR	OUTPUT	BOOL	I、Q、M、D、L	NDR状态参数： 0：作业还未启动或仍然在运行 1：作业已经无错地执行完毕
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0和STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS提供详细信息。 ERROR = 1 出错。STATUS提供关于出错类型的详细信息。
PHYS	IN_OUT	ANY	I、Q、M、D	物理状态(最小长度：一字节)。 可能的值： • 10H 执行功能 • 13H 服务请求
LOG	IN_OUT	ANY	I、Q、M、D	逻辑状态(最小长度：一字节) 可能的数值： • 00H 允许状态改变
LOCAL	IN_OUT	ANY	I、Q、M、D	如果伙伴设备是一个S7 CPU时的状态(最小长度：一字节)

### 输入/输出参数LOCAL

如果通讯伙伴是一个S7 CPU，则输入/输出参数LOCAL包含它的当前状态。第一字节保留，第二字节包含状态的ID。

操作模式	相应标识符
STOP	00H
暖重启	01H
RUN	02H
热重启	03H
HOLD	04H
冷重启	06H
RUN(H系统状态: 冗余)	09H
LINK-UP	0BH
UPDATE	0CH

## 出错信息

下表包含了所有SFB23特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	9	超限警告：使用新设备状态所覆盖较旧的设备状态。
0	11	警告：正在以较低优先级处理接收数据。
0	25	通讯已经启动。作业正在处理。
1	1	通讯故障，例如 <ul style="list-style-type: none"> <li>没有装载连接描述(本地或远程)</li> <li>连接中断(例如，电缆断线、CPU关闭、CP处于STOP模式)</li> </ul>
1	4	PHYS、LOG或LOCAL中出错，包括数据长度或数据类型。
1	10	不能访问本地用户存储区(例如，访问一个已经删除的数据块)。
1	12	当调用SFB时， <ul style="list-style-type: none"> <li>指定了一个不属于SFB23的背景数据块。</li> <li>没有指定任何背景数据块，而是指定了一个共享数据块。</li> <li>没有找到任何背景数据块(从PG中装载新背景数据块)。</li> </ul>
1	18	对于此ID所标识的连接，已经有一个用于SFB23“USTATUS”的实例。
1	19	远程CPU发送数据的速度快于用户程序中SFB可以接收的速度。
1	20	<ul style="list-style-type: none"> <li><b>S7-400</b>：没有足够的工作存储器可用。如果仍然有足够的总工作存储器可用，则解决方案是压缩工作存储器。</li> <li><b>H系统</b>：在更新期间不能进行SFB的第一次调用，因为正在创建工作存储器中的通讯缓冲区。同样地，在更新期间不能改变工作存储器。</li> </ul>

## 19.16 通过SFC62 “CONTROL”查询属于SFB实例的连接状态

### 描述

通过SFC62 “CONTROL”，可以为S7-400查询属于本地通讯SFB实例的连接状态。

在通过将控制输入EN\_R置1来调用系统函数之后，查询通讯SFB实例的连接的当前状态，通过I\_DB选择此实例。

参数	声明	数据类型	存储区域	描述
EN_R	INPUT	BOOL	I、Q、M、D、L、 常数	经激活可进行接收的控制参数在输入置位时发出信号，通知该对象准备接收。
I_DB	INPUT	BLOCK_DB	I、Q、M、D、L、 常数	背景数据块的数目
OFFSET	INPUT	WORD	I、Q、M、D、L、 常数	多重背景数据块中的数据记录数(如果没有多重背景数据块存在，则必须在这里输入0)。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	ERROR和STATUS状态参数，出错显示： ERROR=0和STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS 提供详细信息。  ERROR = 1 出错。STATUS提供关于出错类型的 详细信息。
I_TYP	OUTPUT	BYTE	I、Q、M、D、L	属于所选实例的块类型的标识符
I_STATE	OUTPUT	BYTE	I、Q、M、D、L	<ul style="list-style-type: none"> <li>• = 0：从最后一次冷重启/暖重启或装载之后，还没有调用相应的SFB实例。</li> <li>• &lt;&gt;0：从最后一次冷重启/暖重启或装载之后，至少已经调用了一次相应的SFB实例。</li> </ul>
I_CONN	OUTPUT	BOOL	I、Q、M、D、L	相应连接的状态， 可能的数值： <ul style="list-style-type: none"> <li>• 0：连接断开或没有建立</li> <li>• 1：连接存在</li> </ul>
I_STATUS	OUTPUT	WORD	I、Q、M、D、L	所查询的通讯SFB实例的状态参数STATUS

## 输出参数I\_TYP

下表列出了不同SFB类型和相应的标识符

SFB类型	标识符(W#16#...)
USEND	00
URCV	01
BSEND	04
BRCV	05
GET	06
PUT	07
PRINT	08
START	0B
STOP	0C
RESUME	0D
STATUS	0E
USTATUS	0F
ALARM	15
ALARM_8	16
ALARM_8P	17
NOTIFY	18
AR_SEND	19
NOTIFY_8P	1A
(不存在SFB; I_DB或OFFSET出错)	FF

## 出错信息

SFC62 “CONTROL”的输出参数RET\_VAL可能会为下列两个数值：

- 等于0000H： 在SFC的执行期间没有出错。
- 8000H： 在SFC的执行过程中出错。

### 注意

即使输出参数RET\_VAL指示了数值0000H，还是应该评估输出参数ERROR和STATUS。

ERROR	STATUS (十进制)	说明
1	10	无法访问本地用户存储区(例如，将存储器字节指定为I_TYP的实际参数，但是在正在使用的CPU中此存储器字节并不存在)。
1	12	对于通过I_DB指定的数字， <ul style="list-style-type: none"> <li>• 没有任何背景数据块，而有一个共享数据块。</li> <li>• 没有任何数据块，或实例已经损坏。</li> </ul>

## 19.17 通过FC62 “C\_CNTRL”查询连接状态

### 描述

通过FC62 “C\_CNTRL”，查询S7-300的连接状态。

在通过将控制输入EN\_R置1调用系统函数之后，查询通过ID确定的通讯的当前状态。

参数	声明	数据类型	存储区域	描述
EN_R	INPUT	BOOL	I、Q、M、D、L、 常数	经激活可进行接收的控制参数，在输入置位时发出准备接收信号。
ID	INPUT	WORD	M、D、常数	寻址参数ID，参见用于S7通讯的SFB/FB和SFC/FC的公用参数
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I、Q、M、D、L I、Q、M、D、L	状态参数ERROR和STATUS，出错显示： ERROR=0和STATUS的数值： 等于0000H：既不是警告也不是出错 <>不等于0000H：警告，STATUS提供详细信息。  ERROR = 1 出错。STATUS提供关于出错类型的详细信息。
C_CONN	OUTPUT	BOOL	I、Q、M、D、L	相应连接的状态。 可能的值： <ul style="list-style-type: none"> <li>• 0：连接断开或没有建立。</li> <li>• 1：连接已经建立。</li> </ul>
C_STATUS	OUTPUT	WORD	I、Q、M、D、L	连接状态： <ul style="list-style-type: none"> <li>• W#16#0000：连接还没有建立</li> <li>• W#16#0001：连接正在建立</li> <li>• W#16#0002：连接已经建立</li> <li>• W#16#000F：无连接状态数据可用（例如在CP启动时）</li> <li>• W#16#00FF：连接还没有组态</li> </ul>

## 出错信息

在FC 62“C\_CNTRL”中，输出参数RET\_VAL可以为下列数值：

- 等于0000H： FC执行时没有出错。
- 8000H： FC执行时出错。

---

### 注意

即使输出参数RET\_VAL显示了数值0000H，还是应该评估输出参数ERROR和STATUS。

---

ERROR	STATUS (十进制)	描述
1	10	CP访问出错。当前有其他作业正在运行。以后再重复执行此作业。
1	27	CPU中没有用于此功能块的功能代码。

## 19.18 S7通讯SFB/FB的工作存储器要求

为了平稳地进行工作，根据用户数据(代码区域)的大小，需要给S7通讯功能块SFB/FB分配CPU工作存储器中的一块临时存储区域。所占用的存储区域的大小如下表所示：

S7-300中使用的块		所需要的工作存储器中的存储空间，以字节为单位
FB8	USEND	块：4583字节，实例：368字节
FB9	URCV	块：4880字节，实例：370字节
FB12	BSEND	块：5284字节，实例：372字节
FB13	BRCV	块：5258字节，实例：374字节
FB14	GET	块：4888字节，实例：336字节
FB15	PUT	块：4736字节，实例：384字节
FC 62	C_CNTRL	块：546字节

### 中断特性的注意事项

在S7-300中，只能以一个优先级调用SIMATIC\_NET通讯块。

S7-400中使用的块		所需要的工作存储器中的存储空间，以字节为单位
SFB8/ SFB9	USEND/ URCV	68 + 第一次从SD_1, ...SD_4/RD_1, ...RD_4中调用时所指示的用户数据的长度
SFB12/ SFB13	BSEND/ BRCV	54
SFB14	GET	88 + 第一次从RD_1, ...RD_4中调用时所指示的用户数据的长度
SFB15	PUT	108 + 第一次从SD_1, ...RD_4中调用时所指示的用户数据的长度
SFB16	PRINT	78 + FORMAT的长度规定 + 第一次从SD_1, ...SD_4中调用时指定的用户数据的长度
SFB19	START	52 + 当第一次从PI_NAME和ARG调用时所指示的参数的长度。
SFB20	STOP	48 + 当第一次从PI_NAME调用时所指示的参数的长度。
SFB21	RESUME	52 + 当第一次从PI_NAME和ARG调用时所指示的参数的长度。
SFB22	STATUS	50
SFB23	USTATUS	50



## 20 用于未组态S7连接的通讯SFC

### 20.1 通讯SFC的公用参数

#### 输入参数REQ

输入参数REQ(请求激活)是电平触发控制参数。其用于触发作业(数据传送或连接中止):

- 如果为当前没有激活的作业调用SFC，则通过REQ=1来触发该作业。如果当第一次调用通讯SFC时，没有到通讯伙伴的连接，则在数据传送开始之前首先建立连接。
- 如果触发一个作业，并且当为同样的作业重新调用SFC时该作业还没有结束，则SFC不能使用REQ。

#### 输入参数REQ\_ID(只适用于SFC65 和SFC66 )

输入参数REQ\_ID用于识别发送数据。该参数被发送CPU的操作系统传送到通讯伙伴的CPU的SFC66 “X\_RCV”中。

对于下列两种情况，在接收端上需要REQ\_ID参数:

- 如果在一个发送CPU上通过不同参数REQ\_ID调用几个SFC 65 “X\_SEND”，并将数据传送到一个通讯伙伴。
- 如果使用SFC65 “X\_SEND”，从几个发送CPU发送数据到一个通讯伙伴。

通过计算REQ\_ID，可以将接收到的数据保存到不同的存储区域中。

#### 输出参数RET\_VAL和BUSY

通讯SFC是异步执行的，也就是说作业的执行会在多个SFC调用中持续。输出参数RET\_VAL和BUSY指示作业的状态。参见异步SFC中参数REQ、RET\_VAL和BUSY的含义

## 输入参数CONT

输入参数CONT(继续)是一个控制参数。通过此参数可以决定：作业结束之后是否保持建立与通讯伙伴的连接。

- 如果在第一次调用时选择CONT=0，则在数据传送结束之后将再次终止连接。然后，连接可重新用于与新通讯伙伴进行数据交换。

这种方法确保了只在实际使用时才占用连接资源。

- 如果在第一次调用时选择CONT=1，则在数据传送结束时保持建立连接。

这种方法用于，例如，在两个站之间循环地交换数据。

---

### 注意

通过CONT=1建立的连接可以明确地通过SFC69 “X\_ABORT”或SFC74 “I\_ABORT”来终止。

---

## 20.2 用于未组态S7连接的通讯SFC的出错信息

### 出错信息

“SFC 65到SFC 74的特有出错信息”表给出的用于SFC 65到SFC 74的“实际”出错信息可以分类如下：

出错代码(W#16#...)	说明
809x	执行SFC的CPU上产生的出错
80Ax	永久通讯出错
80Bx	通讯伙伴上出错
80Cx	暂时出错

## SFC 65到SFC 74的特有出错信息

出错代码 (W#16#...)	解释(常规)	解释(用于特定SFC)
0000	执行无错完成。	SFC69 “X_ABORT”和SFC74 “I_ABORT”: REQ=1, 并且没有建立指定的连接。
		SFC66 “X_RCV”: EN_DT=1和RD=NIL
00xy	-	SFC66 “X_RCV”, NDA=1和RD<>NIL: RET_VAL 包含已接收数据的长度(EN_DT=0), 或复制到RD的数据长度 (EN_DT=1)。
		SFC67 “X_GET”: RET_VAL包含已接收的数据块长度。
		SFC72 “I_GET”: RET_VAL包含已接收的数据块长度。
7000	-	SFC65 “X_SEND”、 SFC67 “X_GET”、 SFC68 “X_PUT”、 SFC69 “X_ABORT”、 SFC72 “I_GET”、 SFC73 “I_PUT”和 SFC74 “I_ABORT”: 通过REQ = 0调用 (调用而不执行), BUSY数值为0, 没有数据传送被激活。
		SFC66 “X_RCV”: EN_DT=0/1, NDA=0
7001	第一次调用, REQ=1: 数据传送被触发; BUSY数值为1。	-
7002	中间调用(REQ与此不相关): 数据传送已经激活; BUSY数值为1。	SFC69 “X_ABORT”和SFC74 “I_ABORT”: 中间调用, 使用 REQ=1
8090	指定的通讯伙伴目标地址无效, 例如: <ul style="list-style-type: none"> <li>出错的IOID</li> <li>存在出错的基址</li> <li>出错的MPI地址 (&gt; 126)</li> </ul>	-
8092	SD或RD中出错, 例如: 不允许寻址本地数据区。	SFC65 “X_SEND”, 例如 <ul style="list-style-type: none"> <li>SD长度非法</li> <li>SD=NIL非法</li> </ul>
		SFC66 “X_RCV”, 例如 <ul style="list-style-type: none"> <li>已接收的数据超过RD指定的区域可以容纳的数据量。</li> <li>RD的数据类型是BOOL, 但是已接收的数据长于一字节。</li> </ul>
		SFC67 “X_GET”和SFC72 “I_GET”, 例如 <ul style="list-style-type: none"> <li>RD非法长度</li> <li>RD的长度或数据类型和已接收的数据不匹配。</li> <li>不允许RD=NIL。</li> </ul>

出错代码 (W#16#...)	解释(常规)	解释(用于特定SFC)
		SFC68 "X_PUT"和SFC73 "I_PUT", 例如 <ul style="list-style-type: none"> <li>• SD长度非法</li> <li>• SD=NIL是非法的</li> </ul>
8095	已经以较低优先级执行此功能块。	-
80A0	在已接收的应答中出错	SFC68 "X_PUT"和SFC73 "I_PUT": 通讯伙伴不支持在发送CPU的SD中指定的数据类型。
80A1	通讯故障: 在一个已存在的连接终止之后再调用SFC	-
80B0	不能获得对象, 例如, 没有装载数据块	使用SFC67 "X_GET"、SFC68 "X_PUT"、SFC72 "I_GET"和SFC73 "I_PUT"时可能会发生这种出错
80B1	在ANY指针中出错。要发送的数据区的长度不正确。	-
80B2	硬件出错: 模块不存在 <ul style="list-style-type: none"> <li>• 已组态的插槽并未使用。</li> <li>• 实际模块类型和希望的类型不匹配</li> <li>• 分布式外围I/O不可用。</li> <li>• 在相应SDB中没有用于此模块的条目。</li> </ul>	使用SFC67 "X_GET"、SFC68 "X_PUT"、SFC72 "I_GET"和SFC73 "I_PUT"时可能会发生这种出错
80B3	数据可能是只读或只写, 例如, 写保护的数据块	使用SFC67 "X_GET"、SFC68 "X_PUT"、SFC72 "I_GET"和SFC73 "I_PUT"时可能会发生这种出错
80B4	ANY指针中的数据类型出错, 或不允许使用指定类型的ARRAY。	SFC67 "X_GET"、SFC68 "X_PUT"、SFC72 "I_GET"和SFC73 "I_PUT": 通讯伙伴不支持在VAR_ADDR中指定的数据类型。
80B5	由于使用了非法模式, 执行被拒绝	使用SFC65 "X_SEND"时可能会发生这种出错
80B6	已接收的应答包含不能识别的出错代码。	-

出错代码 (W#16#...)	解释(常规)	解释(用于特定SFC)
80B7	已传送的数据的数据类型和/或长度与伙伴CPU上用于写入据的区域不相符。	使用SFC68 “X_PUT”和SFC73 “I_PUT”时可能会发生这种出错
80B8	-	SFC65 “X_SEND”: 通讯伙伴的SFC66 “X_RCV”不允许数据接收(RD=NIL)。
80B9	-	SFC65 “X_SEND”: 通讯伙伴已经识别出数据块(SFC66 “X_RCV”调用, EN_DT=0), 但是还没有输入到用户程序, 因为通讯伙伴处于STOP模式。
80BA	通讯伙伴的响应和通讯帧不相符。	-
80C0	指定的连接正在被其他作业使用。	-
80C1	执行SFC的CPU上资源不足, 例如: 正在模块上执行的不同发送作业数已经达到允许的最大数目。 连接资源正在使用, 例如, 正用于接收数据。	-
80C2	在通讯伙伴上, 资源暂时不足, 例如: <ul style="list-style-type: none"> <li>• 通讯伙伴当前正在处理最大数目的作业。</li> <li>• 所需要的资源、存储器等正被使用。</li> <li>• 没有足够的工作存储器可用。(压缩存储器)。</li> </ul>	-

出错代码 (W#16#...)	解释(常规)	解释(用于特定SFC)
80C3	<p>在连接建立中出错， 例如：</p> <ul style="list-style-type: none"><li>• 本地S7站没有连接到MPI子网。</li><li>• 已经在MPI子网上对站点进行了编址。</li><li>• 无法访问通讯伙伴。</li><li>• 在通讯伙伴上，资源暂时不足。</li></ul>	-

## 20.3 使用SFC65 “X\_SEND”将数据发送给在本地S7站外的一个通讯伙伴

### 描述

通过SFC65 “X\_SEND”，发送数据到本地S7站以外的通讯伙伴。

在通讯伙伴上使用SFC66 “X\_RCV”接收数据。

在通过REQ=1调用SFC之后再发送数据。

必须要确保由参数SD(在发送CPU上)定义的发送区小于或等于由参数RD(在通讯伙伴上)定义的接收区。如果SD是BOO数据类型，则RD必须也是BOOL数据类型。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“请求激活”，参见用于S7基本通讯的SFC的公用参数
CONT	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“继续”，参见用于S7基本通讯的SFC的公用参数
DEST_ID	INPUT	WORD	I、Q、M、D、L、 常数	地址参数“目标ID”。 该参数包含了通讯伙伴的MPI地址。 通过STEP 7组态了此参数。
REQ_ID	INPUT	DWORD	I、Q、M、D、L、 常数	作业标识符。此参数用于识别通讯伙伴上的数据。
SD	INPUT	ANY	I、Q、M、D	指向发送区。允许使用下列数据类型： BOOL、BYTE、CHAR、WORD、 INT、DWORD、DINT、REAL、 DATE、TOD、TIME、S5TIME、 DATE_AND_TIME和这些数据类型的 数组，BOOL数据类型除外。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在函数执行过程中出错，则返回值 包含相应的出错代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1：发送还没有结束。 BUSY=0：发送已经结束或不存在已经激活的发送函数。

### 数据的一致性

数据以一致状态发送。

### 出错信息

参见用于未组态S7连接的通讯SFC的出错信息和通过输出参数RET\_VAL判断出错。

## 20.4 通过SFC66 “X\_RCV”从本地S7站以外的通讯伙伴中接收数据

### 描述

通过SFC66 “X\_RCV”，接收本地S7站以外的一个或多个通讯伙伴通过SFC65 “X\_SEND”发送的数据。

With SFC66 “X\_RCV,”

- 可以检查数据是已经发送还是正在等待复制。数据被操作系统输入到内部队列。
- 可以将队列中最早的数据块复制到所选择的接收区。

参数	声明	数据类型	存储区域	描述
EN_DT	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“激活数据传送”。通过数值0，可以检查是否至少有一个数据块正在等待被输入到接收区。数值1复制队列中最早的数据块到RD指定的工作存储区域。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在函数执行过程中出错，则返回值包含相应的出错代码。如果没有出错，则RET_VAL包含下列内容： <ul style="list-style-type: none"> <li>• W#16#7000，如果EN_DT=0/1且NDA=0。 在这种情况下，队列中没有数据块。</li> <li>• 如果EN_DT=0且NDA=1，在队列中输入的最早的数据块的长度是一个以字节为单位的正数。</li> <li>• 如果EN_DT=1和NDA=1，复制到RD接收区的数据块长度是一个以字节为单位的正数。</li> </ul>
REQ_ID	OUTPUT	DWORD	I、Q、M、D、L	SFC “X_SEND”的作业标识符，它的数据位于队列的开头，即，是队列中最早的数据。如果队列中没有数据块，则REQ_ID数值为0。
NDA	OUTPUT	BOOL	I、Q、M、D、L	状态参数“已到达的新数据”。 <p>NDA=0:</p> <ul style="list-style-type: none"> <li>• 队列中没有数据块。</li> </ul> <p>NDA=1:</p> <ul style="list-style-type: none"> <li>• 队列至少包含了一个数据块。 (SFC66调用，EN_DT=0)。</li> <li>• 队列中最早的数据块被复制到用户程序中。(SFC66调用，EN_DT=1)。</li> </ul>



参数	声明	数据类型	存储区域	描述
RD	OUTPUT	ANY	I、Q、M、D	指向接收数据区域。允许使用下列数据类型：BOOL、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TOD、TIME、S5_TIME、DATE_AND_TIME和这些数据类型的数组，BOOL数据类型除外。 如果要丢弃队列中最早的数据块，则将数值NIL分配给RD。 接收区的最大长度是76字节。

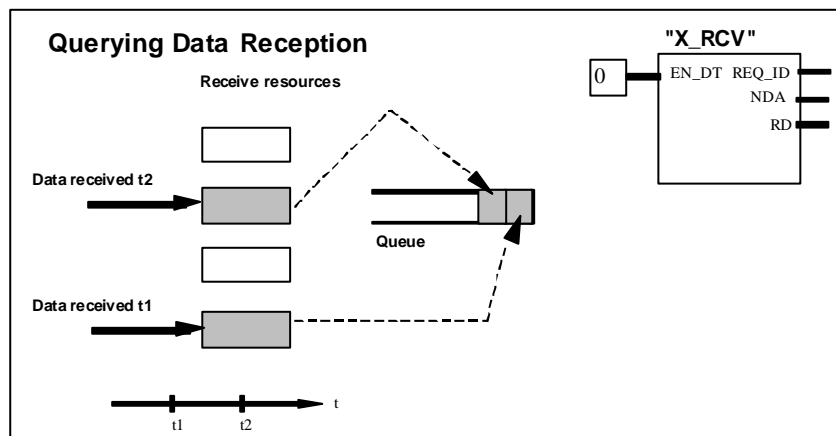
### 通过EN\_DT=0指示数据接收

只要来自通讯伙伴的数据到达，操作系统就按照接收顺序将这些数据输入到队列中。

如果要检查队列中是否至少有一个数据块，则通过EN\_DT=0调用SFC66，并按如下方式计算输出参数NDA：

- NDA=0意味着队列并未包含数据块。REQ\_ID与此不相关。RET\_VAL数值为W#16#7000。
- NDA=1意味着队列中至少有一个数据块可以被读取。

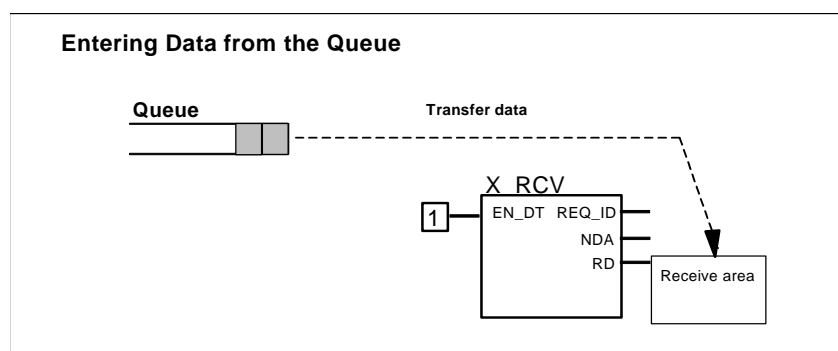
这种情况下，应该判断输出参数RET\_VAL和REQ\_ID(如果可用)。RET\_VAL包含以字节为单位的数据块长度，REQ\_ID包含发送块的作业标识符。如果队列中有多个数据块，则REQ\_ID和RET\_VAL属于队列中最早的数据块。



数据接收

## 通过EN\_DT=1在接收区中输入数据

当通过EN\_DT=1调用SFC66 “X\_RCV”时，队列中最早的数据块被复制到RD指定的工作存储器区域。RD必须大于或等于由SD参数定义的相应SFC65 “X\_SEND”的发送区。如果输入参数SD是BOOL数据类型，则RD必须也是BOOL数据类型。如果要在不同区域内输入接收到的数据，则可以查询REQ\_ID (SFC调用，EN\_DT = 0)，并在继续调用中(EN\_DT = 1)选择一个合适的RD。如果在复制数据时没有出错，则RET\_VAL包含以字节为单位的所复制的数据块的长度，并给发送方发送一个肯定应答。



数据接收

## 放弃数据

如果不想从队列中输入数据，则将数值NIL分配给RD(参见/232/)。在这种情况下，发送方接收一个否定应答(相应SFC65 “X\_SEND”的RET\_VAL，数值为W#1680B8)。SFC66 “X\_RCV”的RET\_VAL数值为0。

## 数据的一致性

在通过EN\_DT=1和RETVAl=W#16#00xy进行调用之后，接收区RD包含了新数据。这些数据可能会被随后的功能块调用所覆盖。为了防止发生这种情况，在评估完接收数据之前，请不要使用相同的接收区RD来调用SFC66 “X\_RCV”。

## 切换到STOP模式

如果CPU切换到STOP模式，

- 则否定应答所有新到达的作业。
- 否定应答所有已经到达并存储在队列中的作业。
  - 如果在STOP之后暖重启或冷重启，则所有数据块都被放弃。
  - 如果在STOP之后重启动且在切换到STOP模式之前查询队列(通过调用SFC66 “X\_RCV”，EN\_DT=0)，则属于最早作业的数据块被输入用户程序(S7-300和S7-400H系统不可以)。否则数据块将被丢弃。

所有其他数据块都将被丢弃。

## 连接中止

如果连接被终止，则将放弃已经在队列中、属于此连接的作业。

例外：如果此作业是队列中最早的作业，并且已经通过调用SFC66 “X\_RCV” (EN\_DT=0)检测到它的存在，则可以通过EN\_DT=1将它输入到接收区。

## 出错信息

参见用于未组态S7连接的通讯SFC的出错信息和通过输出参数RET\_VAL判断出错。

## 20.5 通过SFC68 “X\_PUT”将数据写入本地S7站以外的通讯伙伴

### 描述

通过SFC68 “X\_PUT”，将数据写入不在同一个本地S7站中的通讯伙伴。在通讯伙伴上没有相应SFC。

在通过REQ=1调用SFC之后，激活读作业。此后，可以继续调用SFC，直到BUSY=0指示接收到应答为止。

必须要确保由SD参数(在发送CPU上)定义的发送区和由VAR\_ADDR参数(在通讯伙伴上)定义的接收区长度相同。SD的数据类型还必须和VAR\_ADDR的数据类型相匹配。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“请求激活”，参见用于S7基本通讯的SFC的公用参数
CONT	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“继续”，参见控制参数“请求激活”，参见用于S7基本通讯的SFC的公用参数
DEST_ID	INPUT	WORD	I、Q、M、D、L、 常数	地址参数“目标ID”。它包含通讯伙伴的MPI地址。通过STEP7组态了此参数。
VAR_ADDR	INPUT	ANY	I、Q、M、D	指向伙伴CPU上要从中读取数据的区域。必须选择通讯伙伴支持的数据类型。

参数	声明	数据类型	存储区域	描述
SD	INPUT	ANY	I、Q、M、D	指向本地CPU中包含要发送数据的区域的引用。允许使用下列数据类型： BOOL、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TOD、TIME、S5_TIME、DATE_AND_TIME 和这些数据类型的数组，BOOL数据类型除外。 SD的长度必须和通讯伙伴上的VAR_ADDR参数的长度相同。 SD的数据类型必须与VAR_ADDR相匹配。 发送区的最大长度是76字节
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在函数执行时出错，则返回值包含相应出错信息。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1：发送还没有结束。 BUSY=0：发送已结束，或当前没有激活的发送函数。

### 切换到STOP模式

如果CPU切换到STOP模式，则将终止通过SFC 68 “X\_PUT”建立的连接。不再继续发送数据。如果当CPU改变操作模式时发送数据已经复制到内部缓冲区，则缓冲区的内容将被丢弃。

### 通讯伙伴切换到STOP模式

如果通讯伙伴的CPU切换到STOP模式，不会影响通过SFC68 “X\_PUT”进行的数据传送。处于STOP模式的通讯伙伴也可以写数据。

### 数据的一致性

数据以一致状态发送。

### 出错信息

参见用于未组态S7连接的通讯SFC的出错信息和通过输出参数RET\_VAL判断出错。

## 20.6 通过SFC67 “X\_GET”从本地S7站以外的通讯伙伴中读取数据

### 描述

通过SFC67 “X\_GET”，可以从本地S7站以外的通讯伙伴中读取数据。在通讯伙伴上没有相应SFC。

在通过REQ=1调用SFC之后，读作业被激活。此后，可以继续调用SFC，直到BUSY=0指示数据接收为止。然后，RET\_VAL便包含了以字节为单位的、已接收的数据块的长度。

必须要确保由RD参数定义的接收区(在接收CPU上)至少和由VAR\_ADDR参数定义的要读取的区域(在通讯伙伴上)一样大。RD的数据类型还必须和VAR\_ADDR的数据类型相匹配。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、常数	控制参数“请求激活”，参见用于S7基本通讯的SFC的公用参数
CONT	INPUT	BOOL	I、Q、M、D、L、常数	控制参数“继续”，参见用于S7基本通讯的SFC的公用参数
DEST_ID	INPUT	WORD	I、Q、M、D、L、常数	地址参数“目标ID”。它包含了通讯伙伴的MPI地址。通过STEP 7组态了此参数。
VAR_ADDR	INPUT	ANY	I、Q、M、D	指向伙伴CPU上要从中读取数据的区域。必须选择通讯伙伴支持的数据类型。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在函数执行过程中出错，则返回值包含相应的出错代码。 如果没有出错，RET_VAL包含复制到接收区RD的数据块的长度，它是一个以字节为单位的正数。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 接收还没有结束。 BUSY=0: 接收已经结束，或当前没有激活的接收作业。

参数	声明	数据类型	存储区域	描述
RD	OUTPUT	ANY	I、Q、M、D	<p>指向接收区(接收数据区)。允许使用下列数据类型：BOOL、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TOD、TIME、S5_TIME、DATE_AND_TIME和这些数据类型的数组，BOOL数据类型除外。</p> <p>接收区RD必须至少和通讯伙伴上要读取的数据区域VAR_ADDR一样长。RD的数据类型还必须和VAR_ADDR的数据类型相匹配。</p> <p>接收区的最大长度是76字节。</p>

### 切换到STOP模式

如果CPU切换到STOP模式，则将终止通过SFC67 “X\_GET”建立的连接位于操作系统缓冲区中的、已接收到的数据是否会丢失，取决于所执行的重启动类型：

- 在热重启之后(S7-300和S7-400H系统不可以)，数据被复制到由RD定义的区域中。
- 在暖重启或冷重启之后，数据被丢弃。

### 通讯伙伴切换到STOP模式

如果通讯伙伴的CPU切换到STOP模式，这不会影响通过SFC67 “X\_GET”进行的数据传送。处于STOP模式的通讯伙伴也可以读取数据。

### 数据的一致性

数据以一致状态接收。

### 出错信息

参见用于未组态S7连接的通讯SFC的出错信息和通过输出参数RET\_VAL判断出错。

## 20.7 通过SFC69 “X\_ABORT”中止已存在的、到本地S7站以外的通讯伙伴的连接

### 描述

通过SFC69 “X\_ABORT”，终止一个通过SFC X\_SEND、X\_GET或X\_PUT建立的、到不在同一个本地S7站的通讯伙伴的连接。如果属于X\_SEND、X\_GET或X\_PUT的作业已结束(BUSY = 0)，则在调用SFC69 “X\_ABORT”之后，将释放在通讯两端使用的连接资源。如果属于X\_SEND、X\_GET或X\_PUT的作业还没有结束(BUSY = 1)，则在连接中止之后重新通过REQ = 0和CONT = 0调用相关的SFC，然后等待BUSY = 0。只有这样才能重新释放所有连接资源。只能在有SFC “X\_SEND”、“X\_PUT”或“X\_GET”的通讯端点上才可以调用SFC69 “X\_ABORT”。通过REQ=1来调用SFC，激活中止的连接。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“请求激活”，参见用于S7基本通讯的SFC的公用参数
DEST_ID	INPUT	WORD	I、Q、M、D、L、 常数	地址参数“目标ID”。它包含了通讯伙伴的MPI地址。通过STEP 7组态此参数。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在函数执行过程中出错，则返回值包含相应的出错代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1：连接中止还没有完成。 BUSY=0：连接中止已经完成。

### 切换到STOP模式

如果CPU切换到STOP模式，则结束通过SFC69 “X\_ABORT”启动的连接中止。

### 通讯伙伴切换到STOP模式

如果通讯伙伴的CPU切换到STOP模式，不会影响通过SFC69 “X\_ABORT”进行的连接中止。连接被终止。

### 出错信息

参见用于未组态S7连接的通讯SFC的出错信息和通过输出参数RET\_VAL判断出错。

## 20.8 使用SFC72 “I\_GET”从本地S7站内的一个通讯伙伴上读取数据

### 描述

通过SFC72 “I\_GET”，可以从同一个本地S7站内的通讯伙伴中读取数据。通讯伙伴可以位于中央机架，也可以位于扩展机架或分布式机架中。一定要通过STEP 7分配分布式通讯伙伴到本地CPU。在通讯伙伴上没有相应SFC。

在通过REQ=1调用SFC之后，激活写作业。此后，可以继续调用SFC，直到BUSY=0指示数据接收为止。然后，RET\_VAL便包含了以字节为单位的、已接收的数据块的长度。

必须要确保由RD参数定义的接收区(在接收CPU上)至少和由VAR\_ADDR参数定义的要读取的区域(在通讯伙伴上)一样大。RD的数据类型还必须和VAR\_ADDR的数据类型相匹配。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“请求激活”，参见用于S7基本通讯的SFC的公用参数
CONT	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“继续”，参见用于S7基本通讯的SFC的公用参数
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	通讯伙伴模块的地址范围标识符： B#16#54= 外围设备输入(PI) B#16#55= 外围设备输出(PQ) 混合模块的范围标识符是两个地址中较小的一个。如果两个地址相同，则指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	通讯伙伴模块的逻辑地址。如果是一个混合模块，则指定两个地址中较小的一个。
VAR_ADDR	INPUT	ANY	I、Q、M、D	指向伙伴CPU中要写入数据的区域。选择通讯伙伴支持的数据类型。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在函数执行过程中出错，则返回值包含相应的出错代码。 如果没有出错，RET_VAL包含复制到接收区RD的数据块的长度，它是一个以字节为单位的正数。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 接收还没有结束。 BUSY=0: 接收已经结束，或当前没有激活的接收作业。
RD	OUTPUT	ANY	I、Q、M、D	指向接收区(接收数据区)。允许使用下列数据类型：BOOL、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TOD、TIME、S5_TIME、DATE_AND_TIME和这些数据类型的数组，BOOL数据类型除外。 接收区RD必须至少和通讯伙伴上要读取的数据区域VAR_ADDR一样长。RD的数据类型还必须和VAR_ADDR的数据类型相匹配。 接收区的最大长度是94字节。



### 切换到STOP模式

如果CPU切换到STOP模式，则将终止通过SFC72 “I\_GET”建立的连接。位于操作系统缓冲区中的、已接收到的数据是否会丢失，取决于所执行的重启动类型：

- 在热重启之后(S7-300和S7-400H系统不可以)，数据被复制到由RD定义的区域中。
- 在暖重启或冷重启之后，数据被丢弃。

### 通讯伙伴切换到STOP模式

如果通讯伙伴的CPU切换到STOP模式，不会影响通过SFC72 “I\_GET”进行的数据传送。处于STOP模式的通讯伙伴也可以读取数据。

### 数据的一致性

数据以一致状态接收。

### 出错信息

参见用于未组态S7连接的通讯SFC的出错信息和通过输出参数RET\_VAL判断出错。

## 20.9 使用SFC73 “I\_PUT”将数据写入到本地S7站内的一个通讯伙伴

### 描述

通过SFC73 “I\_PUT”，将数据写入相同本地S7站内的通讯伙伴。通讯伙伴可以位于中央机架，也可以位于扩展机架或分布式机架中。一定要通过STEP 7分配分布式通讯伙伴到本地CPU。在通讯伙伴上没有相应SFC。

在使用REQ控制输入上的信号电平为1调用SFC之后，激活发送作业。

必须要确保由SD参数(在发送CPU上)定义的发送区和由VAR\_ADDR参数(在通讯伙伴上)定义的接收区长度相同。SD的数据类型还必须和VAR\_ADDR的数据类型相匹配。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“请求激活”，参见用于S7基本通讯的SFC的公用参数
CONT	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“继续”，参见用于S7基本通讯的SFC的公用参数
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	通讯伙伴模块的地址范围标识符： B#16#54= 外围设备输入(PI) B#16#55= 外围设备输出(PQ) 混合模块的范围标识符是两个地址中较小的一个。如果两个地址相同，则指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	通讯伙伴模块的逻辑地址。如果是一个混合模块，则指定两个地址中较小的一个。
VAR_ADDR	INPUT	ANY	I、Q、M、D、L	指向通讯伙伴上要写入数据的区域。 选择通讯伙伴支持的数据类型。
SD	INPUT	ANY	I、Q、M、D	指向本地CPU上包含要发送数据的区域。允许使用下列数据类型：BOOL、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TOD、TIME、S5TIME、DATE_AND_TIME以及这些数据类型的数组，BOOL数据类型除外。 SD必须和通讯伙伴上的参数VAR_ADDR长度相同。SD的数据类型还必须和VAR_ADDR的数据类型相匹配。 发送区的最大长度是94字节

参数	声明	数据类型	存储区域	描述
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在函数执行过程中出错，则返回值包含相应的出错代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1：发送还没有结束。 BUSY=0：发送已经结束，或没有发送函数被激活。

### 切换到STOP模式

如果CPU切换到STOP模式，则将终止使用SFC73 “I\_PUT”建立的连接。不再继续发送数据。如果当CPU改变操作模式时发送数据已经复制到内部缓冲区，则缓冲区的内容将被丢弃。

### 通讯伙伴切换到STOP模式

如果通讯伙伴的CPU切换到STOP模式，不会影响通过SFC73 “I\_PUT”进行的数据传送。处于STOP模式的通讯伙伴也可以写数据。

### 数据的一致性

数据以一致状态发送。

### 出错信息

参见用于未组态S7连接的通讯SFC的出错信息和通过输出参数RET\_VAL判断出错。

## 20.10 通过SFC74 “I\_ABORT”中止已存在的、到本地S7站内的通讯伙伴的连接

### 描述

通过SFC74 “I\_ABORT”，终止通过SFC72 “I\_GET”或SFC73 “I\_PUT”建立的、到同一个本地S7站内的通讯伙伴的连接。如果属于I\_GET或I\_PUT的作业已结束(BUSY = 0)，则在调用SFC74 “I\_ABORT”之后，将释放在通讯两端使用的连接资源。

如果属于I\_GET或I\_PUT的作业还没有结束(BUSY = 1)，则在连接中止之后重新通过REQ = 0和CONT = 0调用相关的SFC，然后等待BUSY = 0。只有这样才能重新释放所有连接资源。

只能在有SFC “I\_PUT”或“I\_GET”的通讯端上(即在客户端上)调用SFC74 “I\_ABORT”。

通过REQ=1来调用SFC，激活中止的连接。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数“请求激活”，参见用于S7基本通讯的SFC的公用参数
IOID	INPUT	BYTE	I、Q、M、D、L、 常数	通讯伙伴模块的地址范围标识符： B#16#54= 外围设备输入(PI) B#16#55= 外围设备输出(PQ) 混合模块的范围标识符是两个地址中较小的一个。如果两个地址相同，则指定B#16#54。
LADDR	INPUT	WORD	I、Q、M、D、L、 常数	通讯伙伴模块的逻辑地址。如果是一个混合模块，则指定两个地址中较小的一个。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在函数执行过程中出错，则返回值包含相应的出错代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1：连接中止还没有完成。 BUSY=0：连接中止已经完成。

### **切换到STOP模式**

如果CPU切换到STOP模式，则结束通过SFC74 “I\_ABORT”启动的连接中止。

### **通讯伙伴切换到STOP模式**

如果通讯伙伴的CPU切换到STOP模式，不会影响通过SFC74 “I\_ABORT”进行的连接中止。连接被终止。

### **出错信息**

参见用于未组态S7连接的通讯SFC的出错信息和通过输出参数RET\_VAL判断出错。



# 21 PROFINet

## 21.1 SFC112、113和114背景信息

---

### 注意

缺省情况下操作系统会在扫描周期检查点更新PROFINet接口以及DP互连。不过，如果在组态中禁用了自动更新(例如，为使CPU可以更从容地进行时间调配)，则必须自己执行更新。为此，请在适当的时间调用SFC 112到114。

组态中只能以组的形式禁用此处提到的更新。

---

### 影子存储器

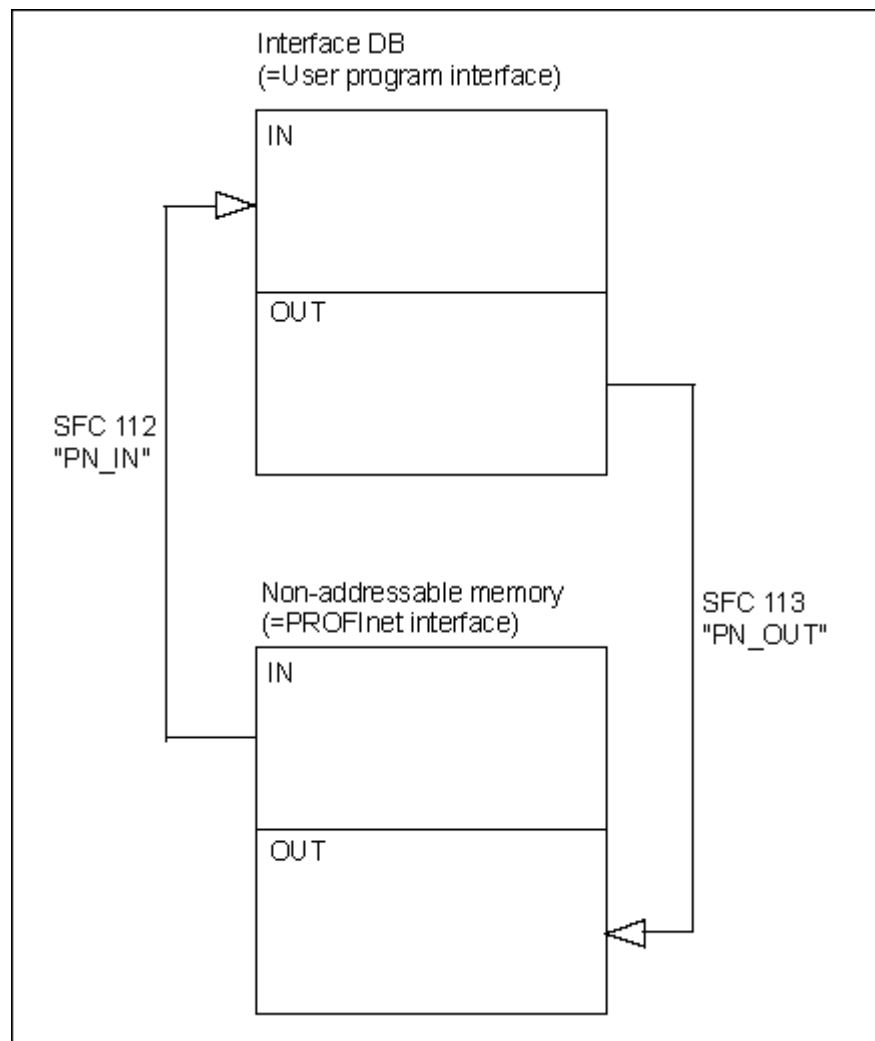
接口数据块是PROFINet组件的用户程序接口。接口数据块是PROFINet组件的用户程序接口。为确保关联程序部分运行时输入与输出一致，各个接口数据块都有一个由操作系统进行管理的结构完全相同的存储器区。该区叫做“影子存储器”。在程序中，您只能访问接口数据块，而其它(外部)PROFINet组件则只能访问影子存储器。这样安排可以防止在访问接口数据块的输入和输出时发生冲突。

### 更新PROFINet接口

通过使用影子存储器实现的数据的一致性意味着更新PROFINet接口分以下两步进行：

- 接口数据块的输入将在PROFINet组件程序启动前被复制到影子存储器的输入
- 接口数据块的输出将在PROFINet组件程序运行后被复制到影子存储器的输出

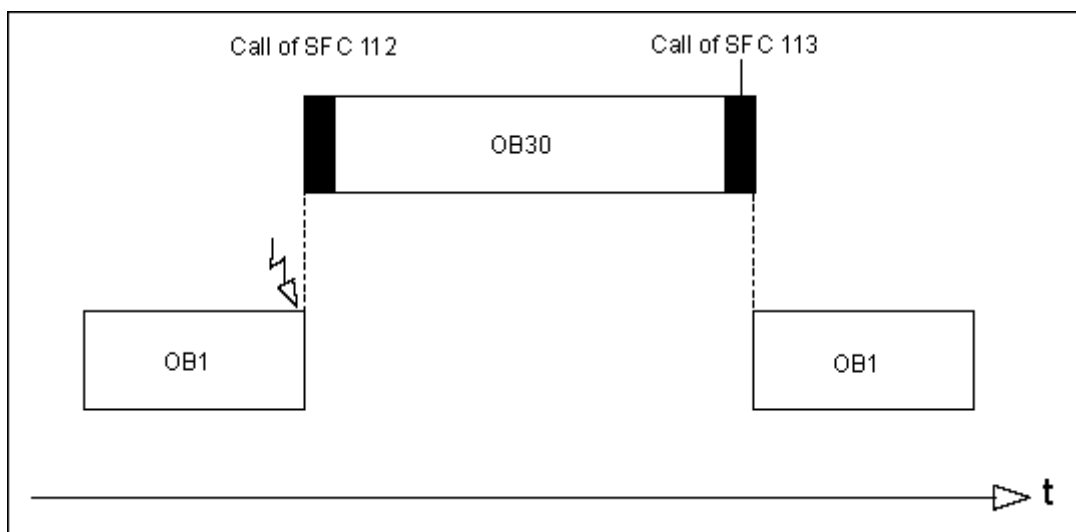
根据在SIMATIC管理器中对PROFINet组件的组态方式，确定是由操作系统还是通过SFC 112和113来更新PROFINet。下图所示为通过SFC 112和113完成更新的方式。



系统范围更新始终发生在扫描周期检查点。

如果通过SFC 112和113执行更新，则在包含PROFINet组件关联程序的OB的开始处调用SFC 112，在此OB的结束处调用SFC 113。下图以OB30 为实例说明这一过程。





### 更新DP互连

根据在SIMATIC管理器中对PROFINet组件的组态方式，确定是由操作系统还是通过SFC 114来更新DP互连。

## 21.2 使用SFC112 “PN\_IN”更新用于PROFINet组件的用户程序接口的输入

### 描述

使用SFC112 “PN\_IN”可以将PROFINet中的输入数据从PROFINet组件的影子存储器复制到关联的接口数据块。完成SFC后，应用程序即可以使用当前这些输入数据。

参数	声明	数据类型	存储区域	描述
DBNO	INPUT	WORD	I、Q、M、D、L、 常数	接口数据块的DB编号
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8002	DB编号与组件组态中的编号不符。
8004	DB编号与组件组态中的编号相符，但尚未装载该DB。
8006	CPU中的接口数据块有写保护。
80B1	读取或写入时发生长度出错。组件组态不适合载入的DB。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 参见：

SFC 112、113和114背景信息

## 21.3 使用SFC113 “PN\_OUT”更新用于PROFINet组件的用户程序接口的输出

### 描述

使用SFC113 “PN\_OUT”可以将应用程序中生成的输出数据从PROFINet组件的接口数据块复制到关联的影子存储器。完成SFC后，其它PROFINet组件可以使用当前这些输入数据。

参数	声明	数据类型	存储区域	描述
DBNO	INPUT	WORD	I、Q、M、D、L、 常数	接口数据块的DB编号
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8002	DB编号与组件组态中的编号不符。
8004	DB编号与组件组态中的编号相符，但尚未装载该DB。
8006	CPU中的接口数据块有写保护。
80B1	读取或写入时发生长度出错。组件组态不适合载入的DB。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 参见:

SFC 112、113和114背景信息

## 21.4 使用SFC114 “PN\_DP”更新DP互连

### 描述

使用SFC 114 “PN\_DP”可以更新下列所有互连：

- 本地PROFIBUS上PROFINet组件间的互连
- 本地PROFIBUS上PROFINet组件与外部PROFINet组件间的周期性传输互连。这种互连应用于网络之间(工业以太网与PROFIBUS DP间)。

### 功能

SFC114 “PN\_DP”以异步方式工作。这意味着其处理过程覆盖若干个SFC调用。要开始更新DP互连，请使用REQ=1调用SFC114。

输出参数RET\_VAL和BUSY中指示了作业状态，另请参见异步工作SFC的REQ、RET\_VAL和BUSY的含义。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L、 常数	REQ = 1: 启动DP互连更新
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在函数执行过程中出错，则返回值包含相应的出错代码。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: DP互连更新尚未完成。

### 出错信息

出错代码(W#16#...)	说明
0000	作业已无误地执行
7000	第一次调用时REQ = 0: 未启动DP互连更新。BUSY的值为0。
7001	第一次调用时REQ = 1: BUSY的值为1。
7002	后续调用(与REQ不相关)。DP互连更新尚未完成。BUSY的值为1。
8095	在较高优先级启动了另一个DP互连更新。不过，较低优先级更新(由操作系统或SFC114 执行)仍在进行。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### 参见：

SFC 112、113和114背景信息

## 22 生成与块相关的消息

### 22.1 关于使用SFB生成块相关消息的介绍

#### 用于生成块相关消息的SFB

可以在程序中通过调用以下SFB中的一个生成块相关消息：

- SFB36 “NOTIFY”
- SFB31 “NOTIFY\_8P”
- SFB33 “ALARM”
- SFB34 “ALARM\_8”
- SFB35 “ALARM\_8P”

上述SFB具有以下属性：

- 除SFB 36 “NOTIFY”和31 “NOTIFY\_8P”以外，调用块时所有检测到的0 -> 1或1 -> 0信号转变都会引起消息传送。
- 如果设置了缺省报告过程(禁用了确认触发报告)，则在调用块时如果检测到任何信号转变，SFB 33 “ALARM”、34 “ALARM\_8”和35 “ALARM\_8P”也将生成一条消息。  
相反，如果启用了确认触发报告，则并非所有信号转变都会触发消息传送(有关详细信息，请参见下文)。
- 块执行之后，将完整读取关联值(输入SD\_i)，并将其分配给消息(请参见SFB/FB的公用参数及S7通讯的SFC/FC的公用参数中的“发送和接收参数”。)  
在与高优先级的一致性比较方面，以下关联值是一致的：  
每个关联值SD\_i都具有固有的一致性。

可以通过状态参数DONE、ERROR和STATUS监视块的处理状态(请参见SFB/FB的公用参数及S7通讯的SFC/FC的公用参数中的“状态参数”)。

---

#### 注意

只有在初次调用块(实例的实际参数或已定义值)时才会对参数ID和EV\_ID进行判断。

---

## 登录显示设备

要使用于生成块相关消息的SFB能够在检测到信号转变时发送消息，事先必须至少登录一台用于显示块相关消息的显示设备(例如，WinCC)。如果没有登录，参数STATUS = 1。

## 信号转变检测

消息块的每个实例可以使用一个包含两个存储器块的消息存储器。

首先将对该消息存储器清零。然后将在输入端SIG或输入端SIG\_1至SIG\_8中的一个检测到的信号转变输入到第一个存储器块。该存储器块将保持占用状态，直到相应消息传送完毕。

然后将在输入端SIG或输入端SIG\_1至SIG\_8中的一个检测到的下一个信号转变输入到第二个存储器块。如果第一个块仍处于占用状态，在任何情况下均将覆盖第二个存储器块。

输出参数ERROR和STATUS将指示这一消息丢失情况。该信息还会与下一个可以传送的消息一并传递给在线显示设备。

清空第一个存储器块后，将把第二个存储器块的数据传递给第一个块。这样，第二个存储器块再次被清空。

## 确认触发报告

将SFB 33 “ALARM”、34 “ALARM\_8”和35 “ALARM\_8P”与确认触发报告一起使用，可以减少系统的消息通讯量。

也就是说，在初次生成信号的进入消息(信号由0转变为1)之后，只有当用户在显示设备上确认第一个消息后，才会继续生成后续消息。确认标识离开消息(信号由1转变为0)后，下一个消息将显示在显示设备上。出现必须进行确认的进入消息(信号由0转变为1)时，消息周期将重启。可以通过此方法使用显示设备来控制信号转变消息(离开消息除外)。

以全局方式为STEP 7组态中的CPU设置SFB 33到35的报告方法(启用或禁用确认触发报告)。缺省情况下将禁用确认触发报告。

为确保系统内消息评估的一致性，应确认是否所有显示设备都能够处理确认触发报告。

---

### 关于不能处理确认触发报告的显示设备的操作注意事项

已启用确认触发报告的CPU只会将消息分配给能够处理此报告方法的显示设备。如果没有显示设备能够处理确认触发报告，CPU将不会发送任何消息。此种情况将通过ERROR=1和STATUS=1指示出来。

---

## 使用SFB 33 “ALARM”、34 “ALARM\_8”和35 “ALARM\_8P”确认消息

采用集中确认原则。在显示设备上确认消息后，确认信息将首先发送给生成该消息的CPU。然后从CPU将确认信息分配到为确认而登录的所有站。

确认的将是信号而不是单个消息。例如，如果指示了信号的若干个上升沿，并且确认了进入状态的事件，将把具有相同消息编号的所有先前事件视为已确认的事件。

### 确认显示

SFB36 “NOTIFY”和SFB31 “NOTIFY\_8P”没有确认指示。可以检查SFB33 “ALARM”的输出参数ACK\_UP和ACK\_DN及SFB 35 “ALARM\_8P”和34 “ALARM\_8”的输出参数ACK\_STATE。如果控制参数EN\_R的值为1，将在调用块时更新这些输出。

### 通过SFC或显示设备(WinCC)启用和禁用消息

在某些情况下，隐藏消息会有帮助(例如，重新组态系统时)。因此，可以在显示设备或程序中禁用和启用消息。禁用/启用应用于所有为特定消息而登录的站。禁用的消息将保持禁用状态，直到再次将其启用。

通过ERROR和STATUS输出参数(ERROR = 1, STATUS = 21)发送禁用消息通知。

### SFB生成块相关消息所需的存储器资源

为了使SFB能够正常运行，必须在CPU通讯缓冲区中为其提供中介存储器空间，具体大小取决于关联值数据的长度。有关占用的存储器大小的信息，请参考下表。

块类型	所需的CPU工作存储器空间(字节)
NOTIFY	200 + 2 * 初次调用时在SD_1、...SD_10指定的关联值的长度
NOTIFY_8P	200 + 2 * 初次调用时在SD_1、...SD_10指定的关联值的长度
ALARM	200 + 2 * 初次调用时在SD_1、...SD_10指定的关联值的长度
ALARM_8	100
ALARM_8P	200 + 2 * 初次调用时在SD_1、...SD_10指定的关联值的长度
AR_SEND	54

## 可以传送的数据量

与NOTIFY、NOTIFY\_8P、ALARM和ALARM\_8P SFB的关联值SD<sub>i</sub>一并传送的数据不得超过最大长度。最大数据长度的计算方法如下：

$\text{maxleng} = \min(\text{pdu\_local}, \text{pdu\_remote}) - \text{diff} - 4 * \text{使用的SD}_i\text{参数的数量}$

此处：

- $\min(\text{pdu\_local}, \text{pdu\_remote})$ 是数据字段pdu<sub>local</sub>和pdu<sub>remote</sub>的最小值
- pdu<sub>local</sub>是本地CPU的数据字段的最大长度(请参见CPU的技术数据)
- pdu<sub>remote</sub>是显示设备的数据字段的最大长度
- diff = 48(如果启用确认触发报告)； diff = 44(如果禁用确认触发报告)。

### 实例：

CPU 414-2通过工业以太网向WinCC发送消息。确认触发报告处于禁用状态。

使用的关联值为SD<sub>1</sub>、SD<sub>2</sub>和SD<sub>3</sub>。

pdu<sub>local</sub> = 480字节， pdu<sub>remote</sub> = 480字节

使用的SD<sub>i</sub>参数的数量： 3

### 因此：

$\text{maxleng} = \min(480, 480) - 44 - 4 * 3 = 480 - 44 - 12 = 424$

每个SFB可以传送的数据的最大长度为424字节。



## 22.2 使用SFB 36 “NOTIFY”生成无需确认的块相关消息

### 描述

SFB36 “NOTIFY”对信号进行监视。它在上升沿(进入状态的事件)和下降沿(离开状态的事件)均会生成消息。最多可以将十个关联值与消息一起发送。消息将发送到为此目的登录的所有站。初次调用SFB时，将发送具有当前信号状态的消息。

检测到沿时将查询关联值，并将其分配给消息。

每个SFB36 “NOTIFY”的实例可以使用一个包含两个存储器块的存储器。有关将信号转变保存到中介存储器的信息，请参考关于使用SFB生成块相关消息的介绍中的“信号转变检测”一节。

SFB36 “NOTIFY”符合IEC 1131-5标准。

参数	声明	数据类型	存储区域	描述
SIG	INPUT	BOOL	I、Q、M、D、L	要监视的信号
ID	INPUT	WORD	I、Q、M、D、L、 常数	消息的数据通道： 仅在初次调用时判断W#16#EEEE ID。
EV_ID	INPUT	DWORD	I、Q、M、D、L、 常数	仅在初次调用时判断消息号(不允许为0) EV_ID。此后每次使用相应的背景数据块 调用SFB36时，都将应用初次调用所使用 的消息号。 STEP 7会自动分配消息号以确保消息号的一 致性。用户程序内的消息号必须唯一。
SEVERITY	INPUT	WORD	I、Q、M、D、L、 常数	事件评价：可能的值： 0至127(值为0表示最高权重)。 此参数与该消息的处理无关。
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数：消息生成已完成
ERROR	OUTPUT	BOOL	I、Q、M、D、L	ERROR状态参数 ERROR=TRUE表示处理过程中出错。 有关详细信息，请参考参数STATUS。
STATUS	OUTPUT	WORD	I、Q、M、D、L	STATUS参数：显示出错信息

参数	声明	数据类型	存储区域	描述
SD_i, 1 i 10	IN_OUT	ANY	I、Q、M、D、T、C	<p>i-th关联值</p> <p>只允许使用下列数据类型：BOOL (不允许：位域)、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TOD、TIME、S5TIME、DATE_AND_TIME。</p> <p><b>注意：</b> 如果ANY指针访问某个DB，则务必指定该DB(例如：P# DB10.DBX5.0 Byte 10)。</p>

## 出错信息

下表包含了所有SFB36特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS (十进制)	说明
0	11	消息丢失，无法发送信号转变/消息
0	22	<ul style="list-style-type: none"> <li>指向关联值SD_i的指针出错： <ul style="list-style-type: none"> <li>涉及数据长度或数据类型</li> <li>用户存储器中的关联值无法访问(例如，由于DB被删除或区域长度出错) 激活的消息没有或最终与有效数量的关联值一起发送</li> </ul> </li> <li>为SEVERITY选择的实际参数超过了允许的范围。激活的消息将在SEVERITY=127下发送。</li> </ul>
0	25	启动了通讯。正在处理消息。
1	1	通讯问题：连接中止或未登录
1	4	初次调用时： <ul style="list-style-type: none"> <li>指定的EV_ID超出了允许的范围或</li> <li>ANY指针SD_i有形式出错</li> <li>超出了每个SFB33 可以为CPU发送的最大存储器区</li> </ul>
1	10	无法访问本地用户存储器(例如，访问的DB已被删除)
1	12	调用SFB时： <ul style="list-style-type: none"> <li>指定了一个不属于SFB36的背景数据块。</li> <li>指定的是共享数据块而不是背景数据块</li> </ul>
1	18	EV_ID已由SFB 31或33至36中的一个使用。
1	20	工作存储器空间不足。 H系统：更新过程中调用了此SFB
1	21	禁用了带有指定EV_ID的消息

## 22.3 使用SFB31 “NOTIFY\_8P”生成无确认显示的块相关消息

### 描述

SFB31 “NOTIFY\_8P”表示将SFB36 “NOTIFY”扩展到八个信号。

在至少检测到一个信号转变时生成消息。始终在初次调用SFB 31时生成消息。为所有八个信号分配一个公用消息号，该消息号在显示设备上拆分成八个子消息。

每个SFB31 “NOTIFY\_8P”的实例可以使用一个包含两个存储器块的存储器。有关将信号转变保存到中介存储器的信息，请参考关于使用SFB生成块相关消息的介绍中的“信号转变检测”一节。

---

### 注意

无论消息是否丢失，显示设备均显示最近的两次信号转变。

---



---

### 警告

在CPU中调用SFB31 “NOTIFY\_8P”前，必须确保所有连接的显示设备均可以识别此块。PLC必须至少采用以下的一个软件版本，才能确保这一点：**STEP 7 V5.1 Service Pack 3**及以上版本、**WinCC V5.1 Hot Fix 1**及以上版本、**PCS7 V5.2 Service Pack 2**及以上版本、**SIMATIC Device Driver V5.6**及以上版本。否则，将关闭PLC与连接的显示设备间的通讯。在这种情况下，将无法再使用系统连接的显示设备访问系统。

---

参数	声明	数据类型	存储区域	描述
SIG_i, 1 . i . 8	INPUT	BOOL	I、Q、M、D、L。	要监视的i-th信号
ID	INPUT	WORD	常数 (I、Q、M、D、L)	消息的数据通道： 仅在初次调用时判断W#16#EEEE ID。
EV_ID	INPUT	DWORD	常数 (I、Q、M、D、L)	消息号(不允许：0) 仅在初次调用时判断EV_ID。此后每次使用相应的背景数据块调用SFB31时，都将应用初次调用所使用的消息号。 <b>STEP 7</b> 会自动分配消息号以确保消息号的一致性。用户程序内的消息号必须唯一。
SEVERITY	INPUT	WORD	常数 (I、Q、M、D、L)	事件权重 有效值：0至127(值为0 = 最高权重)； 缺省值：64 此参数与消息处理无关。
DONE	OUTPUT	BOOL	I、Q、M、D、L	状态参数DONE：消息生成已完成。
ERROR	OUTPUT	BOOL	I、Q、M、D、L	状态参数ERROR ERROR=TRUE表示处理过程中出错。 有关详细信息，请参考参数STATUS。
STATUS	OUTPUT	WORD	I、Q、M、D、L	状态参数STATUS：显示出错信息
SD_i, 1 . i . 10	IN_OUT	ANY	I、Q、M、D、T、Z	i-th关联值 只允许使用以下类型的数据：BOOL (不允许：位域)、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TOD、TIME、S5TIME、DATE_AND_TIME。 <b>注意：</b> 如果ANY指针访问某个DB，则务必指定该DB。 (例如：P# DB10.DBX5.0 Byte 10)

## 出错信息

下表包含所有可通过ERROR和STATUS参数输出的SFB31特有的出错信息。

ERROR	STATUS(十进制)	说明
0	11	消息丢失
0	11	消息丢失, 至少一个信号转变/消息无法发送
0	22	<ul style="list-style-type: none"> <li>指向关联值SD_i的指针出错:</li> <li>与数据长度/类型有关</li> <li>无法访问用户存储器中的关联值(例如, 由于DB已被删除或区域长度出错)</li> </ul> 激活的消息未与或被要求与最大允许数量的关联值一起传送。 <ul style="list-style-type: none"> <li>所选择的SEVERITY实际参数超出了上限。激活的消息将在SEVERITY=127下发送。</li> </ul>
0	25	启动了通讯。正在处理消息。
1	1	通讯出错: 通讯关闭或无登录
1	4	初次调用时 <ul style="list-style-type: none"> <li>指定的EV_ID超出了允许的范围</li> <li>ANY指针SD_i中有形式出错</li> <li>已超过CPU每个SFB31可以传送的最大存储器区长度</li> </ul>
1	10	无法访问本地用户存储器(例如, 尝试访问的DB已被删除)
1	12	调用SFB时 <ul style="list-style-type: none"> <li>指定了一个不属于SFB31的背景数据块。</li> <li>指定的是全局数据块而不是背景数据块</li> </ul>
1	18	EV_ID已由SFB 31或33至36中的一个使用。
1	20	工作存储器空间不足。 H系统: 更新过程中调用了此SFB
1	21	锁定了带有指定EV_ID的消息。

## 22.4 使用SFB 33 “ALARM”生成需要确认的块相关消息

### 描述

SFB33 “ALARM”对信号进行监视。

缺省模式(即, 禁用确认触发报告): 该块在上升沿(进入状态的事件)和下降沿(离开状态的事件)均会生成消息。最多可以将十个关联值与消息一起发送。

启用了确认触发报告: 为信号生成进入消息后, 在显示设备上确认该进入消息时, 该块才会继续生成消息。

(另请参见关于使用SFB生成块相关消息的介绍。)

消息将发送到为此目的登录的所有站。

初次调用SFB时, 将发送具有当前信号状态的消息。

将在上升沿重置ACK\_UP输出。收到登录的显示设备对进入状态的事件的确认时, 将对其进行设置。

ACK\_DN输出的情况与此类似: 将在下降沿对其重置。从登录的显示设备收到对离开状态的事件的确认时, 将对其进行设置。从登录的显示设备收到确认后, 即将确认信息传递给为此目的登录的所有其它站。

每个SFB33 “ALARM”的实例可以使用一个包含2个存储器块的消息存储器。有关将信号转变保存到中介存储器的信息, 请参考关于使用SFB生成块相关消息的介绍中的“信号转变检测”一节。

SFB33 “ALARM”符合IEC 1131-5标准。

参数	声明	数据类型	存储区域	描述
EN_R	INPUT	BOOL	I、Q、M、D、L、 常数	为接收该项而启用的控制参数决定初次调用块时更新(EN_R=1)还是不更新(EN_R=0)输出ACK_UP和ACK_DN。如果EN_R=0, 输出参数ACK_UP和ACK_DN将保持不变。
SIG	INPUT	BOOL	I、Q、M、D、L	要监视的信号
ID	INPUT	WORD	I、Q、M、D、L、 常数	消息的数据通道: 仅在初次调用时判断W#16#EEEE ID。

参数	声明	数据类型	存储区域	描述
EV_ID	INPUT	DWORD	I、Q、M、D、L、 常数	仅在初次调用时判断消息号(不允许为0) EV_ID。此后每次使用相应的背景数据块调用SFB33时，都将应用初次调用所使用的消息号。 <b>STEP 7</b> 会自动分配消息号以确保消息号的一致性。用户程序内的消息号必须唯一。
SEVERITY	INPUT	WORD	I、Q、M、D、L、 常数	事件权重 有效值：0至127(值为0表示最高权重) 此参数与该消息的处理无关。
DONE	OUTPUT	BOOL	I、Q、M、D、L	<b>DONE</b> 状态参数：消息生成 已完成
ERROR	OUTPUT	BOOL	I、Q、M、D、L	<b>ERROR</b> 状态参数 <b>ERROR=TRUE</b> 表示处理过程中出错。 有关详细信息，请参考参数 <b>STATUS</b> 。
STATUS	OUTPUT	WORD	I、Q、M、D、L	<b>STATUS</b> 参数：显示出错信息
ACK_DN	OUTPUT	BOOL	I、Q、M、D、L	在显示设备上确认了离开状态的事件 初始化状态：1
ACK_UP	OUTPUT	BOOL	I、Q、M、D、L	在显示设备上确认了进入状态的事件 初始化状态：1
SD_i, 1 i 10	IN_OUT	ANY	I、Q、M、D、T、C	<b>i-th</b> 关联值 只允许使用下列数据类型：BOOL (不允许：位域)、BYTE、CHAR、 WORD、INT、DWORD、DINT、 REAL、DATE、TOD、TIME、 S5TIME、DATE_AND_TIME。 <b>注意：</b> 如果ANY指针访问某个DB，则务必指 定该DB(例如：P# DB10.DBX5.0 Byte 10)。

## 出错信息

下表包含所有可通过ERROR和STATUS参数输出的SFB33特有的出错信息。

ERROR	STATUS(十进制)	说明
0	11	警告：消息丢失，无法发送信号转变或信号消息。
0	22	<ul style="list-style-type: none"> <li>• 指向关联值SD_i的指针出错：               <ul style="list-style-type: none"> <li>- 涉及数据长度或数据类型</li> <li>- 无法访问用户存储器中的关联值 (例如，由于DB已被删除或区域长度出错)</li> <li>- 激活的消息未与关联值一起发送</li> </ul> </li> <li>• 所选的SEVERITY实际参数超出了允许的范围。激活的消息将在SEVERITY=127下发送。</li> </ul>
0	25	启动了通讯。正在处理消息。
1	1	通讯问题：连接中止或未登录 启用了确认触发报告时：临时显示，如果没有显示设备可以支持确认触发报告
1	4	初次调用时： <ul style="list-style-type: none"> <li>• 指定的EV_ID超出了允许的范围或</li> <li>• ANY指针SD_i有形式出错</li> <li>• 超出了每个SFB33可以为CPU发送的最大存储器区</li> </ul>
1	10	无法访问本地用户存储器(例如，访问的DB已被删除)
1	12	调用SFB时： <ul style="list-style-type: none"> <li>• 指定了不属于SFB 33的背景数据块</li> <li>• 指定的是共享数据块而不是背景数据块</li> </ul>
1	18	EV_ID已由SFB 31或33至36中的一个使用。
1	20	工作存储器空间不足。 H系统：更新过程中调用了FSB
1	21	禁用了带有指定EV_ID的消息

### 注意

初次调用块后，ACK\_UP和ACK\_DN输出的值将是1，并假定SIG输入的先前值为0。



## 22.5 使用SFB35 “ALARM\_8P”生成针对八个信号的带有关联值的块相关消息

### 描述

SFB35 “ALARM\_8P”表示将SFB33 “ALARM”线性扩展至八个信号。

只要未启用确认触发报告，在一个或多个信号处检测到信号转变时即会生成消息(例外：初次调用块时将始终发送消息)。所有八个信号拥有公用消息ID，它在显示设备上被拆分成八个独立的消息。可以分别确认各个消息或确认一组消息。

可以使用ACK\_STATE输出参数在程序中处理单个消息的确认状态。如果禁用或启用ALARM\_8P块的消息，在任何情况下均会影响整个ALARM\_8P块。无法禁用或启用单个信号。

每个SFB35 “ALARM\_8P”的实例可以使用一个带有两个存储器块的存储器。

关于将信号转变保存到中介存储器的信息，请参考关于使用SFB生成块相关消息的介绍中的“信号转变检测”一节。

参数	声明	数据类型	存储区域	描述
EN_R	INPUT	BOOL	I、Q、M、D、L、 常数	为接收该项而启用的控制参数决定调用块时更新(EN_R=1)还是不更新(EN_R=0)输出ACK_STATE。
SIG_i, 1 = i = 8	INPUT	BOOL	I、Q、M、D、L	要监视的i(th)信号
ID	INPUT	WORD	I、Q、M、D、L、 常数	消息的数据通道： 仅在初次调用时判断W#16#EEEE ID。
EV_ID	INPUT	DWORD	I、Q、M、D、L、 常数	仅在初次调用时判断消息号(不允许为0)EV_ID。此后每次使用相应的背景数据块调用SFB35时，都将应用初次调用所使用的消息号。  STEP 7会自动分配消息号以确保消息号的一致性。用户程序内的消息号必须唯一。
SEVERITY	INPUT	WORD	I、Q、M、D、L、 常数	事件权重 有效值：0至127(0表示最高权重) 此参数与该消息的处理无关。
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数：消息生成已完成

参数	声明	数据类型	存储区域	描述
ERROR	OUTPUT	BOOL	I、Q、M、D、L	ERROR状态参数 ERROR=TRUE表示处理过程中出错。 有关详细信息，请参考参数STATUS。
STATUS	OUTPUT	WORD	I、Q、M、D、L	STATUS参数：显示出错信息
ACK_STATE	OUTPUT	WORD	I、Q、M、D、L	包含所有八个消息的当前确认状态的位字段： (1：事件已确认，0：事件未确认)： <ul style="list-style-type: none"> <li>0到7位分别对应SIG_1至SIG_7的进入事件</li> <li>8到15位分别对应SIG_1至SIG_7的离开事件</li> </ul> 初始化状态：W#16#FFFF，即已确认所有进入和离开事件
SD_j, 1 j 10	IN_OUT	ANY	I、Q、M、D、T、C	j-th关联值 关联值将应用于所有消息。只允许使用下列数据类型：BOOL(不允许：位域)、BYTE、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TOD、TIME、S5TIME、DATE_AND_TIME。 <b>注意：</b> 如果ANY指针访问某个DB，则务必指定该DB(例如：P# DB10.DBX5.0 Byte 10)。

## 出错信息

下表包含了所有SFB35特有的出错信息，这些信息可以通过参数ERROR和STATUS输出表示。

ERROR	STATUS(十进制)	说明
0	11	警告：消息丢失，至少一个信号转变/消息无法发送。
0	22	<ul style="list-style-type: none"> <li>指向关联值SD_i的指针出错：               <ul style="list-style-type: none"> <li>与数据长度或数据类型有关</li> <li>无法访问用户存储器中的关联值(例如，由于DB已被删除或区域长度出错)</li> </ul> </li> </ul> 激活的消息未与关联值一起发送 <ul style="list-style-type: none"> <li>所选的SEVERITY实际参数超出了允许的范围。激活的消息将在SEVERITY=127下发送。</li> </ul>
0	25	启动了通讯。正在处理消息。

ERROR	STATUS(十进制)	说明
1	1	通讯问题：连接中止或未登录 启用了确认触发报告时：临时显示，如果没有显示设备可以支持确认触发报告
1	4	初次调用时： <ul style="list-style-type: none"> <li>指定的EV_ID超出了允许的范围或</li> <li>ANY指针SD_i有形式出错</li> <li>超出了每个SFB35可以为CPU发送的最大存储器区</li> </ul>
1	10	无法访问本地用户存储器(例如，访问的DB已被删除)
1	12	调用SFB时： <ul style="list-style-type: none"> <li>指定了一个不属于SFB35的背景数据块。</li> <li>指定的是共享数据块而不是背景数据块</li> </ul>
1	18	EV_ID已由SFB 31或33至36中的一个使用。
1	20	工作存储器空间不足。 H系统：更新过程中调用了此SFB
1	21	禁用了带有指定EV_ID的消息。

**注意**

初次调用块后，将设置ACK\_STATE输出的所有位，并假定输入SIG\_i( $1 \leq i \leq 8$ )的先前值为0。

## 22.6 使用SFB34 “ALARM\_8”生成针对八个信号的不附带关联值的块相关消息

### 描述

除了没有关联值SD\_1至SD\_10外，SFB34 “ALARM\_8”与SFB35 “ALARM\_8P”完全相同。

参数	声明	数据类型	存储区域	描述
EN_R	INPUT	BOOL	I、Q、M、D、L、 常数	为接收该项而启用的控制参数决定调用块时更新(EN_R=1)还是不更新(EN_R=0)输出ACK_STATE。
SIG_i, 1≤i≤8	INPUT	BOOL	I、Q、M、D、L	要监视的i(th)信号
ID	INPUT	WORD	I、Q、M、D、L、 常数	消息的数据通道：仅在初次调用时判断W#16#EEEE ID。
EV_ID	INPUT	DWORD	I、Q、M、D、L、 常数	仅在初次调用时判断消息号(不允许为0)EV_ID。此后每次使用相应的背景数据块调用SFB34时，都将应用初次调用所使用的消息号。  STEP 7会自动分配消息号以确保消息号的一致性。用户程序内的消息号必须唯一。
SEVERITY	INPUT	WORD	I、Q、M、D、L、 常数	事件权重 有效值：0至127(值为0表示最高权重)。此参数与该消息的处理无关。
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数：消息生成已完成
ERROR	OUTPUT	BOOL	I、Q、M、D、L	ERROR状态参数 ERROR=TRUE表示处理过程中出错。有关详细信息，请参考参数STATUS。
STATUS	OUTPUT	WORD	I、Q、M、D、L	STATUS参数：显示出错信息
ACK_STATE	OUTPUT	WORD	I、Q、M、D、L	包含所有八个消息的当前确认状态的位字段：(1：事件已确认，0：事件未确认)： <ul style="list-style-type: none"> <li>0到7位分别对应SIG_1至SIG_8的进入事件</li> <li>8到15位分别对应SIG_1至SIG_8的离开事件</li> </ul> 初始化状态：W#16#FFFF，即已确认所有进入和离开事件

## 出错信息

下表包含所有可通过ERROR和STATUS参数输出的SFB 34特有的出错信息。

ERROR	STATUS(十进制)	说明
0	11	警告：消息丢失，至少一个信号转变/消息无法发送。
0	22	所选的SEVERITY实际参数超出了允许的范围。激活的消息将在SEVERITY=127下发送。
0	25	启动了通讯。正在处理消息。
1	1	通讯问题：连接中止或未登录 启用了确认触发报告时：临时显示，如果没有显示设备可以支持确认触发报告
1	4	初次调用时，指定的EV_ID超出了允许的范围。
1	10	无法访问本地用户存储器(例如，访问的DB已被删除)
1	12	<ul style="list-style-type: none"> <li>• 调用SFB时：</li> <li>• 指定了一个不属于SFB34的背景数据块。</li> <li>• 指定的是共享数据块而不是背景数据块</li> </ul>
1	18	EV_ID已由SFB 31或33至36中的一个使用。
1	20	工作存储器空间不足。 H系统：更新过程中调用了此SFB
1	21	禁用了带有指定EV_ID的消息

### 注意

初次调用块后，将设置ACK\_STATE输出的所有位，并假定输入SIG\_i(1≤i≤8)的先前值均为0。

## 22.7 使用SFB37 “AR\_SEND”发送归档数据

### 描述

SFB37 “AR\_SEND”将归档数据发送给为此目的登录的操作员界面系统。这些系统把登录消息中的相关归档号通知CPU。视CPU可用的存储空间和已用地址区，归档数据最大长度可达65534字节。在归档数据的结构中，必须考虑所使用的操作员界面系统的缺省值。

数据的发送在调用块后由控制输入REQ的上升沿激活。归档数据的起始地址由SD\_1指定，数据字段的长度由LEN指定。数据传送与用户程序的执行异步。DONE状态参数的值为1表示传送成功完成。控制输入R的上升沿中止数据的传送。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	控制参数请求
R	INPUT	BOOL	I、Q、M、D、L、 常数	控制参数重置：当前作业已中止
ID	INPUT	WORD	I、Q、M、D、L、 常数	消息的数据通道：仅在初次调用时判断W#16#EEEE ID。
AR_ID	INPUT	DWORD	I、Q、M、D、L、 常数	归档号(不允许为0) 仅在初次调用时判断AR_ID。此后，每次使用相应背景数据块调用SFB37时，都将使用初次调用时的归档号。STEP 7会自动分配消息号以确保消息号的一致性。
DONE	OUTPUT	BOOL	I、Q、M、D、L	DONE状态参数：发送已完成
ERROR	OUTPUT	BOOL	I、Q、M、D、L	ERROR状态参数 ERROR=TRUE表示处理过程中出错。 有关详细信息，请参考参数STATUS。
STATUS	OUTPUT	WORD	I、Q、M、D、L	STATUS参数：显示出错信息
SD_1	IN_OUT	ANY	I、Q、M、D、T、C	指向归档数据的指针。不对长度规范进行判断。 只允许使用下列数据类型：BOOL (不允许：位域)、CHAR、WORD、INT、DWORD、DINT、REAL、DATE、TOD、TIME、S5TIME、DATE_AND_TIME。 归档数据必须具有PLC特有的结构。 <b>注意：</b> 如果使用ANY指针访问一个数据块，则必须要指定该数据块(例如：P# DB10.DBX5.0 Byte 10)。
LEN	IN_OUT	WORD	I、Q、M、D、L	要发送的数据字段的长度(字节)

## 出错信息

下表包含所有可通过ERROR和STATUS参数输出的SFB37特有的出错信息。

ERROR	STATUS(十进制)	说明
0	11	警告：新作业未激活，因为前一作业仍在进行。
0	25	通讯已经启动。作业正在处理。
1	1	通讯问题
1	2	否定确认，无法执行功能
1	3	没有针对指定AR_ID的登录。
1	4	<ul style="list-style-type: none"> <li>归档数据指针SD_1中存在涉及数据长度或数据类型的出错。</li> <li>初次调用时，指定的AR_ID超出了允许的范围。</li> </ul>
1	5	所请求的重置已执行。
1	7	由于当前功能已完成或未激活(块处于不正确的状态)，因此与RESET作业无关。
1	10	无法访问本地用户存储器(例如，访问的DB已被删除)。
1	12	调用SFB时： <ul style="list-style-type: none"> <li>指定了一个不属于SFB37的背景数据块。</li> <li>指定的是共享数据块而不是背景数据块</li> </ul>
1	18	AR_ID已由SFB37使用。
1	20	工作存储器空间不足。 H系统：更新过程中调用了此SFB

## 数据的一致性

为确保数据的一致性致性限制为206，只有在完成当前发送过程后，才能写入当前使用的发送区域部分。当状态参数DONE的值为1时，便是这种情况。

## 22.8 使用SFC10 “DIS\_MSG”禁止与块相关的消息、与符号相关的消息以及组状态消息

### 描述

可以使用SFC10 “DIS\_MSG”(禁用消息)禁由SFB生成的块相关消息及禁用符号相关消息(SCAN)和组状态消息。使用输入参数MODE和MESGN来选择要禁用的消息。只有在未使用SFC10 禁用消息时，才能调用SFC10 “DIS\_MSG”并成功禁用消息。

对于准备在调用SFC10时发送但仍然位于内部缓冲区中的消息，将无法再行禁用，而会发送出去。禁用的消息在“NOTIFY”、“NOTIFY\_8P”、“ALARM”、“ALARM\_8P”和“ALARM\_8” SFB的ERROR和STATUS输出中给予指示。

通过在调用SFC10 时为REQ输入参数赋值1来启动对消息的禁用。

### SFC10 的工作方式

禁用以异步方式执行，换言之，它可以在若干个SFC10调用过程中一直保持激活状态。另请参见异步工作SFC的REQ、RET\_VAL和BUSY含义：

- 初次调用时(REQ = 1)，SFC10 检查输入参数，并尝试占据所需的系统资源。如果尝试成功，将在RET\_VAL中输入值W#16#7001、设置BUSY并启动对消息的禁用。如果不成功，将在RET\_VAL中输入出错信息并终止作业。在这种情况下不会评价BUSY。
- 如果期间进行了更多调用，将在RET\_VAL中输入值W#16#7002(仍然由CPU执行作业)并设置BUSY。进行更多调用不会影响当前作业。
- 最后一次调用SFB时，如果没有出错，将在RET\_VAL中输入值W#16#0000。届时BUSY的值将是0。如果出错，将在RET\_VAL中输入出错信息且不会评价BUSY。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	REQ = 1: 触发禁用
MODE	INPUT	BYTE	I、Q、M、D、L、 常数	有关用于选择要禁用消息的参数，请参见下表
MESGN	INPUT	DWORD	I、Q、M、D、L、 常数	只有在将MODE设置为5、6、7时，消息号才有意义。可以通过它禁用单个消息。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	有关出错信息，请参见“出错信息”表
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 尚未取消禁用。



**MODE输入参数**

下表列出了允许MODE输入参数使用的值：

值	含义
0	使用SFB生成CPU的所有块相关、符号相关和组状态消息
1	使用SFB生成CPU的所有块相关消息，即所有由“NOTIFY”、“NOTIFY_8P”、“ALARM”、“ALARM_8P”和“ALARM_8” SFB生成的消息
2	CPU的所有组状态消息
3	CPU的所有符号相关消息(SCAN)
5	“符号相关消息”类的单个消息
6	“使用SFB生成的块相关消息”类的单个消息
7	“组状态消息”类的单个消息

**出错信息**

出错代码(W#16#...)	说明
0000	禁用终止而没有出错。
7000	第一次调用时REQ = 0：未激活禁用。
7001	第一次调用时REQ = 1：触发了禁用。
7002	再次调用：禁用已激活。
8081	访问参数时出错
8082	MODE含有违规值。
8083	消息号超出了允许的值范围。
8084	没有登录使用MODE(及可能的情况下使用MESGN)指定的消息。
80C3	无法禁用要在MODE(及可能的情况下在MESGN)中禁用的消息，因为目前SFC10已在禁用消息。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 22.9 使用SFC9 “EN\_MSG”启用块相关、符号相关和组状态消息

### 描述

使用SFC9 “EN\_MSG”(启用消息)可以启用先前禁用的块相关、符号相关和组状态消息。或者在显示设备上禁用消息，或者使用SFC10 “DIS\_MSG”禁用。

使用MODE和MESGN输入参数指定要启用的消息。只有在SFC9 尚未有效地启用消息时，才能使用SFC9 “EN\_MSG”成功地启用消息。

通过为SFC9 的REQ输入参数赋值1来启动启用功能。

### SFC10 的工作方式

启用以异步方式执行，换言之，它可以在若干个SFC9调用过程中一直保持激活状态。另请参见异步工作SFC的REQ、RET\_VAL和BUSY的含义：

- 初次调用时(REQ = 1)，SFC9 检查输入参数，并尝试占用所需的系统资源。如果尝试成功，将在RET\_VAL中输入值W#16#7001、设置BUSY并启动对消息的启用。如果不成功，将在RET\_VAL中输入出错信息并终止作业。在这种情况下不会评价BUSY。
- 如果期间进行了更多调用，将在RET\_VAL中输入值W#16#7002(仍然由CPU 执行作业)并设置BUSY。进行更多调用不会影响当前作业。
- 最后一次调用SFB时，如果没有出错，将在RET\_VAL中输入值W#16#0000。届时BUSY的值将是0。如果出错，将在RET\_VAL中输入出错信息且不会评价BUSY。

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L	REQ = 1: 触发启用
MODE	INPUT	BYTE	I、Q、M、D、L、 常数	用于选择要启用消息 的参数
MESGN	INPUT	DWORD	I、Q、M、D、L、 常数	只有在将MODE设置为5、6、7时， 消息号才有意义。可以通过它启用 单个消息。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息，请参见表格。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1: 尚未取消启用。

## MODE输入参数

下表列出了允许MODE输入参数使用的值。

值	含义
0	使用SFB生成CPU的所有块相关、符号相关和组状态消息
1	使用SFB生成CPU的所有块相关消息，换言之，即所有使用“NOTIFY”、“NOTIFY_8P”、“ALARM”、“ALARM_8P”和“ALARM_8” SFB生成的消息
2	CPU的所有组状态消息
3	CPU的所有符号相关消息(SCAN)
5	“符号相关消息”类的单个消息
6	“使用SFB生成的块相关消息”类的单个消息
7	“组状态消息”类的单个消息

## 出错信息

出错代码(W#16#...)	说明
0000	禁用终止而没有出错。
7000	第一次调用时REQ = 0: 未激活启用。
7001	第一次调用时REQ = 1: 已触发启用。
7002	再次调用: 启用已激活。
8081	访问参数时出错
8082	MODE含有违规值。
8083	消息号超出了允许的值范围。
8084	没有登录使用MODE(及可能的情况下使用MESGN)指定的消息。
80C3	目前无法启用要在MODE(及可能的情况下在MESGN)中启用的消息，因为SFC9已在启用消息。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 22.10 用于生成与块相关的消息的SFB的启动特性

### 重新启动(暖重启)

在暖重启过程中，将把用于生成块相关消息的SFB的背景数据块设置为NO\_INIT (未初始化)状态。存储在背景数据块中的实际参数未发生变化。将在下次调用块时判断参数ID和EV\_ID。

### 完全重启

在完全重启过程中，将把用于生成块相关消息的SFB的背景数据块内容重置为其初始值。

### 热重启

在热重启过程中，用于生成块相关消息的SFB的特性类似于具有继续执行功能的用户功能块。它们从中断点继续执行。

### 存储器复位

只要复位存储器，就将使所有连接终止，从而也没有为消息而登录的站。将删除用户程序。如果插入了闪存卡，与执行相关的程序段将再次从该卡加载到CPU上，随后CPU执行暖重启或冷重启(在任何情况下均隐式执行冷重启，因为清空存储器后将初始化所有用户数据)。

## 22.11 用于生成与块相关的消息的SFB如何响应故障

### 连接故障

对分配给SFB实例的连接进行故障监视。如果某个连接发生故障，将从为块相关消息而登录的站内部CPU列表中删除所涉及的站。删除这些站的所有未决消息。

如果在连接发生故障后其它站仍然处于登录状态，它们将继续接收消息。只有在没有任何登录到站的连接时，SFB才会停止发送消息。ERROR和STATUS输出参数会指示这种情况(ERROR = 1, STATUS = 1)。

### 到用户程序的出错接口

如果在执行用于生成块相关消息的SFB的过程中发生出错，将把ERROR输出参数设置为1，STATUS输出参数将具有相应的出错标识符。可以在程序中判断此出错信息。

可能的出错实例：

- 由于资源不足而无法发送
- 访问要监视的信号中的一个时出错。

## 22.12 使用SFC生成与块相关的消息简介

### 用于生成块相关消息的SFC

可以使用下列SFC生成块相关消息：

- SFC17 “ALARM\_SQ”
- SFC18 “ALARM\_S”
- SFC107 “ALARM\_DQ”
- SFC108 “ALARM\_D”

以上SFC具有下列属性：

- 可以在登录的显示设备上确认信号状态为1时由SFC17 “ALARM\_SQ”和SFC107 “ALARM\_DQ”发送的消息。始终隐式确认SFC18 “ALARM\_S”和SFC108 “ALARM\_D”的消息。生成消息的不是检测到的边沿变化，而是每个SFC调用。有关更多详细信息，请参考使用SFC17 “ALARM\_SQ”生成可确认的块相关消息及使用SFC18 “ALARM\_S”生成永久公认的块相关消息
- “ALARM\_S”和使用SFC 17 “ALARM\_DQ”和108 “ALARM\_D”生成可确认的块相关消息
- 执行块后，将完整读取关联值SD\_1，并将其分配给消息。在一致性方面，与高优先级类相比，以下关联值是一致的：
  - 简单数据类型(位、字节、字及双字)
  - 数据类型字节的数组，其最大长度取决于具体CPU特定的最大长度(请参见 **171**, **101**)。)

### SFC19 “ALARM\_SC”

使用SFC19 “ALARM\_SC”可以查询下列内容：

- 上一“进入状态消息”的确认状态和上一SFC17 /SFC107 调用的信号状态，或
- 上一SFC18 /SFC108 调用的信号状态。

### 登录显示设备

只有在至少为块相关消息登录了一台显示设备时，用于生成块相关消息的SFC才会在其被调用时发送消息。

## 消息存储

为避免在通讯系统的通讯量大时丢失消息，SFC 17、18、107和108都可以缓存两个消息。

不过，如果丢失了消息，会在RET\_VAL中通知用。下次可以发送消息时，会将此情况通知登录的显示设备。

## 使用SFC 17 “ALARM\_SQ”和107 “ALARM\_DQ”确认消息

如果在显示设备上确认了“进入事件消息”，将先把此确认信息发送给生成消息的CPU。随后由CPU将该确认信息分配给为此目的登录的所有站。

## 禁用和启用消息

对于使用SFC17 “ALARM\_SQ”、SFC18 “ALARM\_S”、SFC107 “ALARM\_DQ”或SFC108 “ALARM\_D”生成的块相关消息，无法在禁用后再启用。

## 包含SFC17 /SFC18 调用的程序的更改

---

### 注意

使用SFC17 /SFC18 调用下载一个已在CPU中的块时，可能会出现的情况是：前一个块已发送了进入状态消息，但新块未发送相应的离开状态消息。这意味着消息将保留在CPU的内部消息存储器中。这种情况还会发生在使用SFC17/SFC18删除块时。

将CPU转入STOP状态，然后执行暖重启或冷重启，即可从CPU的内部消息存储器中删除此类消息。

---

## 包含SFC17 /SFC18 调用的程序的更改

即使程序可能包含SFC107 和/或SFC108 调用，所述程序修改仍可能使消息驻留在内部消息存储器中，并因此永久性地占用系统资源。

与由SFC17 /SFC18 调用占用的系统资源相反，可以在不将CPU切换为STOP模式的情况下，释放由SFC107 /SFC108 调用占用的系统资源。这是通过使用SFC106 “DEL\_SI”来执行的，请参见使用SFC106 “DEL\_SI”释放动态占用的系统资源。在通过调用SFC106 “DEL\_SI”释放动态占用的系统资源前，可能需要借助SFC105 “READ\_SI”获取有关CPU当前动态占用的系统资源的信息。请参见使用SFC105 “READ\_SI”读取动态占用的系统资源。

## 可以传送的数据的量

使用SFC ALARM\_S、ALARM\_SQ、ALARM\_D和ALARM\_DQ的伴随值SD可以传送的数据量不能超过最大长度。数据长度的计算方法如下：

$$\text{maxleng} = \min(\text{pdu\_local}, \text{pdu\_remote}) - 48$$

定义：

- pdu\_local: CPU数据块(SZL\_ID W#16#0131、INDEX 1、变量pdu)的最大长度
- pdu\_remote: 显示设备数据块的最大长度

**实例：**

一个CPU 414-1将向一个编程设备PG 760(通过MPI)发送消息。

pdu\_local = 480字节， pdu\_remote = 480字节，

结果：

$$\text{maxleng} = \min(480, 480) - 48 = 480 - 48 = 432$$

因此，每个SFC可以传送的最大数据长度为432字节。



## 22.13 使用SFC17 “ALARM\_SQ”生成可确认的与块相关的消息以及使用SFC18 “ALARM\_S”生成永久确认的与块相关的消息

---

### 注意

由于新创建的程序提供经改进的系统资源管理选项，因此只应使用SFC 107和108。

---

### 描述

每次调用SFC17 “ALARM\_SQ”和SFC18 “ALARM\_S”时，它们都会生成一个可以向其添加关联值的消息。把该消息发送给为此而登录的所有站。SFC17 和SFC18 提供了简单的消息发送机制。必须确保当触发信号SIG的值是上一调用的反值时，只调用SFC17 或SFC18 。如果不是这种情况，将在RET\_VAL中给予指示，且不会发送消息。初次调用SFC17 或SFC18 时，必须确保SIG输入的值为1。否则，RET\_VAL将包含出错信息，且不会发送消息。

---

### 注意

请务必在先前已分配了适当系统属性的FB中调用SFC17 和SFC18 ！  
有关向块分配系统属性的更多详细信息，请参考/234/。

---

### 系统资源的使用

使用SFC17 “ALARM\_SQ”和18 “ALARM\_S”生成消息时，操作系统在信号周期过程中使用一个系统资源。

对于SFC18 “ALARM\_S”，信号周期从SFC调用SIG=1开始持续到SIG=0的另一个调用。对于SFC17 “ALARM\_SQ”，此时期还包括直到其中一个已报告的显示设备确认进入信号的时间(必要时)。

如果在信号周期中，生成消息的块超载或被删除，相关系统资源将保持占用状态，直到下次重启(暖重启)。

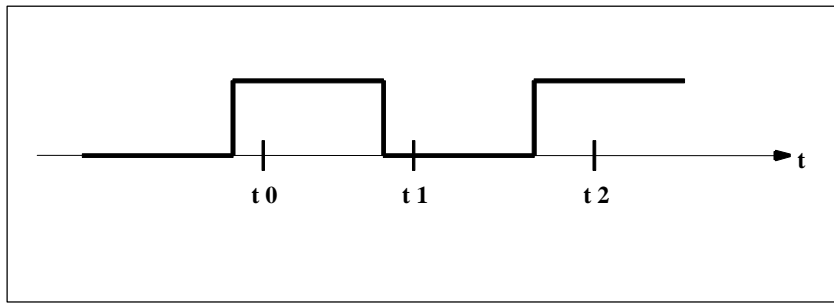
## 确认消息

在受监视信号的状态为1时，可以确认由SFC17 “ALARM\_SQ”发送的消息。可以使用SFC19 “ALARM\_SC”查询上一“进入事件消息”的确认状态和上次SFC调用的信号状态。始终隐式确认使用SFC18 “ALARM\_S”发送的消息。使用SFC18 “ALARM\_SC”可以查询上次SFC19调用的信号状态。

## 信号状态的临时存储

SFC17 “ALARM\_SQ”和SFC18 “ALARM\_S”会临时占用系统资源。它们在此处输入上两个信号状态(包括时间标志和关联值)等信息。如果在未发送上两个“有效”SFC调用的信号状态(信号溢出)时调用SFC17 或SFC18，将放弃当前和上一信号状态，并在缓冲区中设置溢出ID。在下一个可能的时机，将发送倒数第二个信号和溢出标识符。

实例：



t0、t1和t2是调用SFC17 或SFC18所在的点。如果t0和t1的信号状态未在时间t2发送，将放弃t1和t2的信号状态，并会为t0的信号状态设置溢出标识符。

## 实例溢出

如果SFC17 或SFC18 调用的数量大于CPU系统资源的最大数量，可能会导致资源不足(实例溢出)。将通过RET\_VAL中的信息和登录的显示设备指明这种情况。

SFC17 或SFC18 调用的最大数量取决于具体的CPU。可以在/70/和/101/中找到此信息。

参数	声明	数据类型	存储区域	描述
SIG	INPUT	BOOL	I、Q、M、D、L	触发消息的信号
ID	INPUT	WORD	I、Q、M、D、L、 常数	消息的数据通道: W#16#EEEE
EV_ID	INPUT	DWORD	I、Q、M、D、L、 常数	消息号(不允许为0)
SD	INPUT	ANY	I、Q、M、D、T、C	相关值 最大长度: 12字节 允许使用以下数据类型 BOOL(不允许: 位域)、BYTE、CHAR、 WORD、INT、DWORD、DINT、 REAL、DATE、TOD、TIME、S5TIME、 DATE_AND_TIME、COUNTER、 TIMER。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息

## 出错信息

出错代码(W#16#...)	说明
0000	未出错。
0001	<ul style="list-style-type: none"> <li>关联值超过了允许的最大长度或</li> <li>无法访问用户存储器(例如, 访问的DB已被删除)。发送消息。</li> <li>关联值指向本地数据区中的某个值。发送消息。</li> </ul>
0002	警告: 已经使用了最后一部分可用的消息确认存储器。
8081	指定的EV_ID超出了允许的范围。
8082	由于CPU不再有可以供SFC生成块相关消息的资源, 消息丢失。
8083	由于已存在相同的信号更改但尚无法发送(信号溢出), 消息丢失。
8084	当前SFC17 或SFC18 调用中, 触发消息的信号(SIG)的值与上次调用时的值相同。
8085	没有针对指定EV_ID的登录
8086	已在较低优先级类执行了针对指定EV_ID的SFC 调用。
8087	初次调用SFC17 或SFC18 时, 消息触发信号的值为0。
8088	指定的EV_ID已由另一个系统资源(分配给SFC17、18、107、108)使用。
8xyy	常规出错信息, 请参见使用输出参数RET_VAL评估出错。

## 22.14 使用SFC19 “ALARM\_SC”查询上一ALARM\_SQ/ ALARM\_DQ进入事件消息的确认状态

### 描述

使用SFC19 “ALARM\_SC”可以查询下列内容：

- 上一ALARM\_SQ/ALARM\_DQ进入状态消息的确认状态和上次调用SFC17 “ALARM\_SQ”/SFC107 “ALARM\_DQ”时触发消息的信号状态，或
- 上次调用SFC18 “ALARM\_S”/SFC108 “ALARM\_D”时触发消息的信号状态。

假定在消息组态期间分配了消息号，将通过SFC19 “ALARM\_SC”访问SFC17 或SFC18 /SFC107 /SFC108 临时占用存储器的唯一消息号引用消息或信号。

参数	声明	数据类型	存储区域	描述
EV_ID	INPUT	DWORD	I、Q、M、D、L、 常数	上次SFC调用信号状态的消息号或想要查询的上一进入状态消息(仅限使用SFC17和SFC107！)的确认状态。
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息
STATE	OUTPUT	BOOL	I、Q、M、D、L	上次SFC调用触发消息的信号状态
Q_STATE	OUTPUT	BOOL	I、Q、M、D、L	如果指定的EV_ID参数属于SFC18 /SFC108调用：1
				如果指定的EV_ID参数属于SFC17 /SFC107调用：上一进入状态消息的确认状态： 0：未确认 1：已确认

### 出错信息

出错代码(W#16#...)	说明
0000	未出错。
8081	指定的EV_ID超出了允许的范围。
8082	当前没有针对此EV_ID而占据存储器空间(可能的原因：相应的信号状态尚不是1或信号状态已恢复为0)。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 22.15 使用SFC 107 “ALARM\_DQ”和108 “ALARM\_D”生成可确认和永久确认的块相关消息

### 描述

每次调用，SFC 107 “ALARM\_DQ”和108 “ALARM\_D”都会生成一个可向其附加关联值的消息。从而与SFC 17 “ALARM\_SQ”和18 “ALARM\_S”保持一致。

使用SFC 107 “ALARM\_DQ”和108 “ALARM\_D”生成消息时，操作系统会在信号周期过程中临时占用系统资源。

SFC108 “ALARM\_D”的信号周期始于SIG=1的SFC调用，终于SIG=0的新调用。对于SFC107 “ALARM\_DQ”，它的周期可能因时间超时而延长，一直到在登录的显示设备上确认进入信号为止。

对于SFC108 “ALARM\_D”，信号周期从SFC调用SIG=1开始持续到SIG=0的另一个调用。对于SFC107 “ALARM\_DQ”，此时期还包括直到其中一个已报告的显示设备确认进入信号的时间(如有必要)。

如果在信号周期过程中，生成消息的块超载或被删除，相关系统资源将保持占用状态，直到下次重启(暖重启)。

与SFC 17和18相比，SFC 107 “ALARM\_DQ”和108 “ALARM\_D”所增加的功能是：可以对这些占用的系统资源进行管理：

- 借助SFC105 “READ\_SI”，可以获取与占用的系统资源有关的信息。
- 可以使用SFC106 “DEL\_SI”再次释放占用的系统资源。这对于永久占用的系统资源具有特殊意义。例如，如果在程序更改过程中删除了包含SFC107或SFC108调用的FB，当前占用的系统资源将保持占用状态，直到下次重启(暖重启)。更改程序并重新加载包含SFC 107或SFC 108调用的FB时，可能会发生SFC 107和108不再生成消息的情况。

SFC 107和108包含的参数比SFC 17和18多一个，即输入CMP\_ID。使用此输入可以将由SFC 107和108生成的消息分配到逻辑区，例如，系统的各个部分。如果调用FB中的SFC107/SFC108，显然需要将相应的背景数据块号分配到CMP\_ID。

参数	声明	数据类型	存储区域	描述
SIG	INPUT	BOOL	I、Q、M、D、L	消息触发信号
ID	INPUT	WORD	I、Q、M、D、L、 常数	消息的数据通道：W#16#EEEE
EV_ID	INPUT	DWORD	I、Q、M、D、L、 常数	消息号(不允许：0)

参数	声明	数据类型	存储区域	描述
CMP_ID	INPUT	DWORD	I、Q、M、D、L、 常数	组件标识符(不允许: 0) 将相应消息分配到的部分系统的ID 建议值: <ul style="list-style-type: none"> <li>• 低位字: 1至65535</li> <li>• 高位字: 0</li> </ul> 如果使用上述建议值, 将不会遇到与SIEMENS程序数据包冲突的情况。
SD	INPUT	ANY	I、Q、M、D、T、 C	相关值 最大长度: 12字节 只允许使用BOOL型数据(不允许: 位域)、BYTE、CHAR、WORD、INT、 DWORD、DINT、REAL、DATE、 TOD、TIME、S5TIME、 DATE_AND_TIME。
RET_VAL	OUTPUT	INT	E、A、M、D、L	出错信息

## 出错信息

出错代码 (W#16#...):	说明
0000	未出错。
0001	<ul style="list-style-type: none"> <li>• 关联值的长度超出了允许的最大长度, 或</li> <li>• 无法访问用户存储器(例如, 访问的DB已被删除)。发送激活的消息。</li> <li>• 关联值指向本地数据区中的某个值。发送消息。</li> </ul>
0002	警告: 最后一部分可用的消息确认存储器已被占用。
8081	指定的EV_ID超出了有效范围。
8082	由于CPU不再有可以供SFC生成块相关消息的资源, 消息丢失。
8083	由于已存在相同的信号转变但尚无法发送(信号溢出), 消息丢失。
8084	对于当前和以前的SFC107 -/SFC-108调用, 消息触发信号SIG具有相同的值。
8085	没有针对指定EV_ID的登录。
8086	已在较低优先级类处理了针对指定EV_ID的SFC调用。
8087	初次调用SFC107 /SFC108 时, 消息触发信号的值为0。
8088	指定的EV_ID已由另一个系统资源(分配给SFC17、18、107、108)使用。
8089	已将CMP_ID赋值为0。
8xyy	常规出错信息, 请参见使用输出参数RET_VAL评估出错。

## 22.16 使用SFC105 “READ\_SI”读取动态系统资源

### 使用SFC107和108生成消息时，动态占用的系统资源有何变化

使用SFC 107 “ALARM\_DQ”和108 “ALARM\_D”生成消息时，操作系统会临时占用系统存储器空间。

例如，如果不删除CPU中存在的包含SFC107 或SFC108 调用的FB，可能会发生相应系统资源保持永久占用状态的情况。如果重新装载加载包含SFC107或SFC108调用的FB，则可能发生SFC107和108无法再得到正确处理的情况。

### 描述

使用SFC105 “READ\_SI”可以读取生成消息时由SFC107和108占用的系统资源。这是通过在此处使用EV\_ID和CMP\_ID值来实现的。这些值通过SI\_ID参数传给SFC105 “READ\_SI”。

SFC105 “READ\_SI”具有下表说明的四种可能的操作模式。通过MODE参数设置所需的操作模式。

MODE	读取SFC107 /SFC108 占用的哪些系统资源？
1	全部(SI_ID: =0的SFC105 调用)
2	EV_ID: =ev_id的SFC107 -/SFC108 调用(SI_ID: =ev_id的SFC105 调用)占用的系统资源
3	CMP_ID: =cmp_id的SFC107 -/SFC108 调用(SI_ID: =ev_id的SFC105 调用)占用的系统资源
0	由于指定的目标域SYS_INST过小，不能通过上一调用在MODE=1或MODE=3下读取的其它系统资源

### 工作原理

如果在MODE=1或MODE=3下调用SFC105 时未选择足够大的SYS\_INST目标区，该区将包含通过MODE参数选择的所有当前占用的系统资源的内容。

系统资源负载大将导致相应的SFC运行时间较长。也就是说，高CPU性能负载可能导致超过最大可组态周期监视时间。

可以使用下列方法解决此运行时间问题：选择一个相对较小的SYS\_INST目标区。如果SFC无法在SYS\_INST中输入要读取的所有系统资源，RET\_VAL=W#16#0001会发送通知。在这种情况下，请在MODE=0下并使用与前一调用相同的SI\_ID调用SFC105，直到RET\_VAL的值为W#16#0000。

**注意**

由于操作系统不会组合属于读取作业的SFC105调用，因此应以相同优先级执行所有SFC105调用。

**目标区SYS\_INST的结构**

获取的已占用系统资源的目标区必须位于DB内。应将目标区适当定义为结构域，结构的构成如下：

结构单元	数据类型	描述
SFC_NO	WORD	占用系统资源的SFC的编号
LEN	BYTE	结构的长度(字节)，包括SFC_NO和LEN: B#16#0C
SIG_STAT	BOOL	信号状态
ACK_STAT	BOOL	进入事件的确认状态(上升沿)
EV_ID	DWORD	消息号
CMP_ID	DWORD	部分系统ID

参数	声明	数据类型	存储区域	描述
MODE	INPUT	INT	I、Q、M、D、L、 常数	作业标识符 允许值 <ul style="list-style-type: none"> <li>1: 读取所有系统资源</li> <li>2: 读取调用SFC107 -/SFC108时使用EV_ID = ev_id占用的系统资源</li> <li>3: 读取调用SFC107 -/SFC108时使用CMP_ID = cmp_id占用的系统资源</li> <li>0: 后续调用</li> </ul>
SI_ID	INPUT	DWORD	I、Q、M、D、L、 常数	要读取的系统资源的ID 允许值 <ul style="list-style-type: none"> <li>0, 如果MODE=1</li> <li>消息号ev_id, 如果MODE=2</li> <li>用于标识系统部分的ID cmp_id, 如果MODE=3</li> </ul>
RET_VAL	OUTPUT	INT	I、Q、M、D、L	返回值(出错信息或作业状态)
N_SI	OUTPUT	INT	I、Q、M、D、L	包含SYS_INT的输出系统资源数
SYS_INT	OUTPUT	ANY	D	已获取系统资源的目标区。



## 出错信息

出错代码 (W#16#...):	说明
0000	未出错。
0001	由于所选的SYS_INT目标范围过窄，因此无法读取所有系统资源。
8081	(仅限MODE=2或3)已将SI_ID赋值为0。
8082	(仅限MODE=1)已将一个0不同值赋值到SI_ID。
8083	(仅限MODE=0)已将SI_ID赋值为与前一SFC调用(MODE=1或3)SI_ID不同的值。
8084	为MODE分配了非法值。
8085	SFC105 正在另一个OB中接受处理。
8086	系统资源的目标区SYS_INST过小。
8087	DB中不存在目标区SYS_INST。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

## 22.17 使用SFC106 “READ\_SI”读取动态系统资源

### 使用SFC 107和108生成消息时动态系统资源有何变化

使用SFC 107 “ALARM\_DQ”和108 “ALARM\_D”生成消息时，操作系统会临时占用系统存储器空间。

例如，如果不删除CPU中存在的包含SFC107 或SFC108 调用的FB，可能会发生相应系统资源保持永久占用状态的情况。如果重新加载包含SFC107或SFC108调用的FB，则可能发生SFC107和108无法再得到正确处理的情况。

### 描述

使用SFC106 “DEL\_SI”可以删除当前使用的系统资源。

SFC106 “READ\_SI”具有下表说明的三种可能的操作模式。通过MODE参数设置所需的操作模式。

MODE	删除SFC107/SFC108占用的哪些系统资源？
1	全部(SI_ID: =0的SFC106 调用)
2	EV_ID: =ev_id的SFC107 -/SFC108 调用(SI_ID: =ev_id的SFC106 调用)占用的系统资源
3	CMP_ID: =cmp_id的SFC107 -/SFC108 调用(SI_ID: =ev_id的SFC106 调用)占用的系统资源

参数	声明	数据类型	存储区域	描述
MODE	INPUT	INT	I、Q、M、D、L、 常数	作业标识符 允许值 <ul style="list-style-type: none"> <li>1: 删除所有系统资源</li> <li>2: 删除调用SFC107/SFC108时使用EV_ID = ev_id占用的系统资源</li> <li>3: 删除调用SFC107/SFC108时使用CMP_ID = cmp_id占用的系统资源</li> </ul>
SI_ID	INPUT	DWORD	I、Q、M、D、L、 常数	要删除的系统资源的ID允许值 <ul style="list-style-type: none"> <li>0, 如果MODE=1</li> <li>消息号ev_id, 如果MODE=2</li> <li>用于标识系统部分的ID cmp_id, 如果MODE=3</li> </ul>
RET_VAL	OUTPUT	INT	I、Q、M、D、L	出错信息

## 出错信息

出错代码 (W#16#...):	说明
0000	未出错。
8081	(仅限MODE=2或3)已将SI_ID赋值为0。
8082	(仅限MODE=1)已将一个0不同值赋值到SI_ID。
8084	为MODE分配了非法值。
8085	当前正在处理SFC106。
8086	由于在调用SFC106时至少有一个所选择的系统资源正在接受处理，因此无法删除全部所选择的系统资源。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。



## 23 IEC定时器和IEC计数器

### 23.1 使用SFB 3“TP”生成一个脉冲

#### 描述

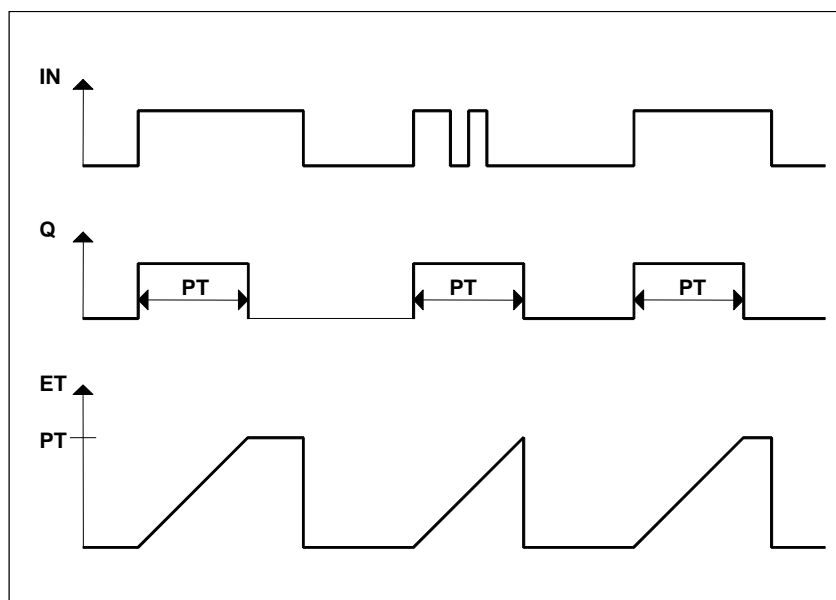
SFB3 “TP”生成了长度为PT的脉冲。定时器只在STARTUP和RUN模式下运行。

输入IN的信号上升沿启动脉冲。在时间PT内输出Q将保持置位状态，无论输入信号是否有变化(换言之，即使IN输入在时间PT用完前从0变回1)。ET输出提供输出Q处于置位状态的时间。ET输出的最大值是PT输入的值。输出ET在输入IN变为0时复位，但复位不会在时间PT用完前进行。

SFB3“TP”符合IEC 1131-3标准。

操作系统会在冷重启期间重置SFB3“TP”的实例。如果希望在暖重启后初始化此SFB的实例，必须在OB100中使用PT = 0ms调用SFB3。如果此SFB的实例位于另一个块中，可以，例如通过初始化更高级别的块，重置这些实例。

#### 时序图



参数	声明	数据类型	存储区域	描述
IN	INPUT	BOOL	I、Q、M、D、L、 常数	开始输入
PT	INPUT	TIME	I、Q、M、D、L、 常数	脉冲的持续时间。PT必须为正值。 (注意：值范围由TIME数据类型确定)
Q	OUTPUT	BOOL	I、Q、M、D、L	时间的状态
ET	OUTPUT	TIME	I、Q、M、D、L	用完的时间

## 23.2 使用SFB 4“TON”生成一个接通延迟

### 描述

SFB4“TON”使信号上升沿延迟时间PT。定时器只在STARTUP和RUN模式下运行。

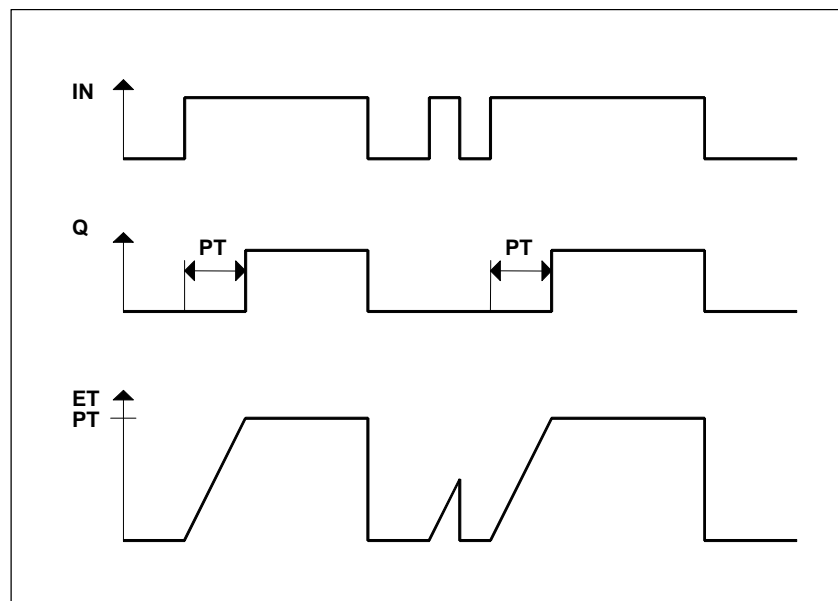
IN输入的上升沿会导致时间PT用完后输出Q出现上升沿。随后Q将保持置位状态，直到IN输入再次变为0。如果IN输入在时间PT用完前变为0，输出Q将保持其设置0不变。

ET输出提供IN输入的上一上升沿后过去的时间。其最大值为PT输入的值。IN输入变为0时将复位ET。

SFB4“TON”符合IEC 1131-3标准。

操作系统会在冷重启期间重置SFB4“TON”的实例。如果希望在暖重启后初始化此SFB的实例，必须在OB100中使用PT = 0ms调用SFB4。如果此SFB的实例位于另一个块中，可以，例如通过初始化更高级别的块，重置这些实例。

### 时序图



参数	声明	数据类型	存储区域	描述
IN	INPUT	BOOL	I、Q、M、D、L、 常数	开始输入
PT	INPUT	TIME	I、Q、M、D、L、 常数	IN输入上升沿延迟的时间。 PT必须为正值。 (注意：值范围由TIME数据类型确定)
Q	OUTPUT	BOOL	I、Q、M、D、L	时间的状态
ET	OUTPUT	TIME	I、Q、M、D、L	用完的时间



## 23.3 使用SFB 5“TOF”生成一个关闭延迟

### 描述

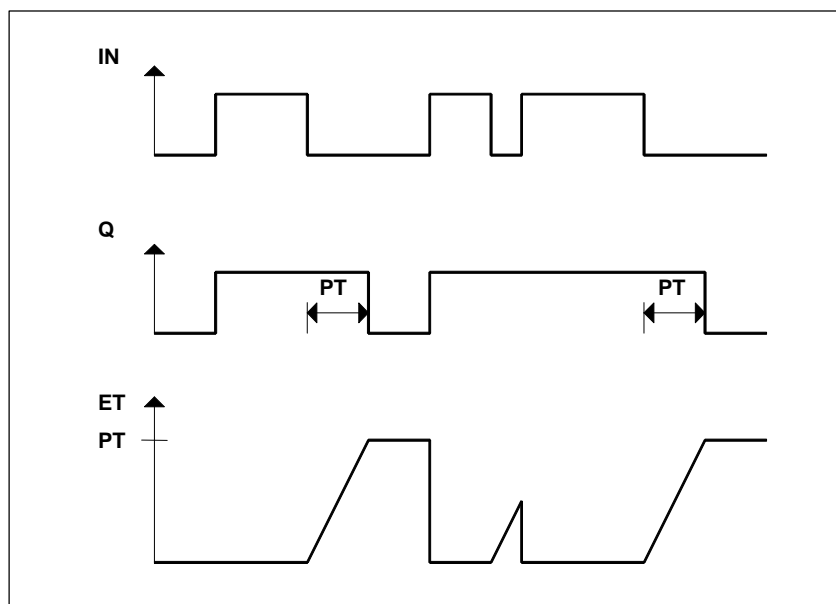
SFB5“TOF”使下降沿延迟时间PT。定时器只在STARTUP和RUN模式下运行。

IN输入的上升沿会导致输出Q出现上升沿。IN输入的下降沿会导致输出Q的下降沿延迟时间PT。如果在时间PT用完前IN输入变回1，输出Q将保持设置1不变。ET输出提供IN输入上一下降沿后经过的时间。但其最大值为PT输入的值。IN输入变为1时将重置ET。

SFB5“TOF”符合IEC 1131-3标准。

操作系统会在冷重启期间重置SFB5“TOF”的实例。如果希望在暖重启后初始化此SFB的实例，必须在OB100中使用PT = 0ms调用SFB5。如果此SFB的实例位于另一个块中，可以，例如通过初始化更高级别的块，重置这些实例。

### 时序图



参数	声明	数据类型	存储区域	描述
IN	INPUT	BOOL	I、Q、M、D、L、 常数	开始输入
PT	INPUT	TIME	I、Q、M、D、L、 常数	IN输入下降沿延迟的时间。PT必须为正值。 (注意：值范围由TIME数据类型确定)
Q	OUTPUT	BOOL	I、Q、M、D、L	时间的状态
ET	OUTPUT	TIME	I、Q、M、D、L	用完的时间

## 23.4 使用SFB 0 “CTU”递增计数

### 描述

可以使用SFB0 “CTU”递增计数。CU输入出现上升沿将使计数器递增1(与上一SFB调用相比)。如果计数值达到上限32767，将不再递增。CU输入出现任何后续上升沿都将不再起作用。

不管CU输入的当前值是什么，R输入信号级别为1时将把计数器重置为0。

Q输出指示当前计数值是否大于或等于预设值PV。

SFB0 “CTU”符合IEC 1131-3标准。

操作系统会在冷重启期间重置SFB0 “CTU”的实例。如果希望在暖重启后初始化此SFB的实例，必须在OB100中使用R = 1调用SFB 0。如果此SFB的实例位于另一个块中，可以(例如)通过初始化更高级别的块重置这些实例。

参数	声明	数据类型	存储区域	描述
CU	INPUT	BOOL	I、Q、M、D、L、 常数	计数器输入
R	INPUT	BOOL	I、Q、M、D、L、 常数	重置输入R优先于CU。
PV	INPUT	INT	I、Q、M、D、L、 常数	预设值。有关PV的作用，请参考参数Q。
Q	OUTPUT	BOOL	I、Q、M、D、L	计数器的状态：Q具有下列值 <ul style="list-style-type: none"> <li>• 1，如果<math>CV \geq PV</math></li> <li>• 0，其它情况</li> </ul>
CV	OUTPUT	INT	I、Q、M、D、L	当前计数值 (允许值：0至32767)

## 23.5 使用SFB 1 “CTD”递减计数

### 描述

可以使用SFB1 “CTD”递减计数。CD输入出现上升沿将使计数器递减(与上一SFB调用相比)。如果计数值达到下限 -32768，将不再递减。CD输入再出现任何上升沿将不再起作用。

不管CD输入的值是多少，LOAD输入信号级别为1时将把计数器设置为预设值PV。

Q输出指示当前计数值是否小于或等于0。

SFB1 “CTD”符合IEC 1131-3标准。

操作系统会在冷重启期间重置SFB1 “CTD”的实例。如果希望在暖重启后初始化此SFB的实例，必须在OB100中使用LOAD = 1和PV = 所需的CV初始值调用SFB 1。如果此SFB的实例位于另一个块中，可以，例如通过初始化更高级别的块，重置这些实例。

参数	声明	数据类型	存储区域	描述
CD	INPUT	BOOL	I、Q、M、D、L、 常数	计数输入
LOAD	INPUT	BOOL	I、Q、M、D、L、 常数	加载输入。LOAD优先于CD。
PV	INPUT	INT	I、Q、M、D、L、 常数	预设值。LOAD输入的信号级别为1时，将把计数器预置为PV。
Q	OUTPUT	BOOL	I、Q、M、D、L	计数器的状态：Q具有下列值： <ul style="list-style-type: none"> <li>• 1，如果CV _ 0</li> <li>• 0，其它情况</li> </ul>
CV	OUTPUT	INT	I、Q、M、D、L	当前计数值(允许值： -32 768至32767)

## 23.6 使用SFB2 “CTUD”递增/递减计数

### 描述

可以使用SFB2 “CTUD”递增和递减计数。出现上升沿时计数值按以下方式变化(与上一SFB调用相比):

- 出现在输入CU时递增1
- 出现在输入CD时递减1。

如果计数值达到极限值，计数器将做如下反应:

- 达到下限 -32768时将不再递减
- 达到上限32767时将不再递增。

如果在一个周期内输入CU和输入CD都有上升沿，计数器将保持其当前值不变。此反应不符合IEC 1131-3标准。在该标准中，如果两个信号同时处于激活状态，CU输入优先。已向IEC提交有关对此更改的提议。

不管CU和CD输入的值是什么，LOAD输入的信号级别为1时将把计数器预置为值PV。

不管CU、CD和LOAD输入的值是什么，R输入的信号级别为1时将把计数器重置为0。QU输出指示当前计数值是否大于或等于预设值PV；QD输出指示该值是否小于或等于0。

操作系统会在冷重启期间重置SFB2 “CTUD”。如果希望在暖重启后初始化SFB2 “CTUD”，必须在OB100中使用下列参数调用SFB2：

- 使用R = 1(使用块递增计数时)
- 使用R = 0和LOAD = 1及PV = 必需的CV初始值(使用块递减计数时)
- 如果此SFB的实例位于另一个块中，可以，例如通过初始化更高级别的块，重置这些实例。

参数	声明	数据类型	存储区域	描述
CU	INPUT	BOOL	I、Q、M、D、L、 常数	递增计数输入。
CD	INPUT	BOOL	I、Q、M、D、L、 常数	递减计数输入
R	INPUT	BOOL	I、Q、M、D、L、 常数	重置输入。R优先于LOAD。
LOAD	INPUT	BOOL	I、Q、M、D、L、 常数	装载输入。LOAD优先于CU和CD。
PV	INPUT	INT	I、Q、M、D、L、 常数	预设值。LOAD输入的信号级别为1时，将把计数器设置为预设值PV。

参数	声明	数据类型	存储区域	描述
QU	OUTPUT	BOOL	I、Q、M、D、L	递增计数器的状态： QU具有下列值 <ul style="list-style-type: none"><li>• 1, 如果<math>CV \geq PV</math></li><li>• 0, 其它情况</li></ul>
QD	OUTPUT	BOOL	I、Q、M、D、L	递减计数器的状态： QD具有下列值 <ul style="list-style-type: none"><li>• 1, 如果<math>CV \leq 0</math></li><li>• 0, 其它情况</li></ul>
CV	OUTPUT	INT	I、Q、M、D、L	当前计数值 (允许值: -32768至32767)

## 24 IEC功能

### 24.1 概述

您可以将以下国际电工委员会(IEC)功能从STEP 7库“S7libs\Stdlib30”复制到程序目录。

名称	IEC块系列	功能
FC3 D_TOD_DT	转换	将DATE和TIME_OF_DAY组合为DT
FC6 DT_DATE	转换	从DT提取DATE
FC7 DT_DAY	转换	从DT提取星期
FC8 DT_TOD	转换	从DT提取TIME_OF_DAY
FC33 S5TI_TIM	转换	数据类型转换, 从S5TIME到TIME
FC40 TIM_S5TI	转换	数据类型转换, 从TIME到S5TIME
FC16 I_STRNG	转换	数据类型转换, 从INT到STRING
FC5 DI_STRNG	转换	数据类型转换, 从DINT到STRING
FC30 R_STRNG	转换	数据类型转换, 从REAL到STRING
FC38 STRNG_I	转换	数据类型转换, 从STRING到INT
FC37 STRNG_DI	转换	数据类型转换, 从STRING到DINT
FC39 STRNG_R	转换	数据类型转换, 从STRING到REAL
FC9 EQ_DT	DT	比较DT, 求等于
FC12 GE_DT	DT	比较DT, 求大于或等于
FC14 GT_DT	DT	比较DT, 求大于
FC18 LE_DT	DT	比较DT, 求小于或等于
FC23 LT_DT	DT	比较DT, 求小于
FC28 NE_DT	DT	比较DT, 求不等于
FC10 EQ_STRNG	字符串	比较STRING, 求等于
FC13 GE_STRNG	字符串	比较STRING, 求大于或等于
FC15 GT_STRNG	字符串	比较STRING, 求大于
FC19 LE_STRNG	字符串	比较STRING, 求小于或等于
FC24 LT_STRNG	字符串	比较STRING, 求小于
FC29 NE_STRNG	字符串	比较STRING, 求不等于
FC21 LEN	字符串	STRING变量的长度
FC20 LEFT	字符串	STRING变量的左侧部分
FC32 RIGHT	字符串	STRING变量的右侧部分
FC26 MID	字符串	STRING变量的中间部分
FC2 CONCAT	字符串	组合两个STRING变量
FC17 INSERT	字符串	在STRING变量中插入

名称	IEC块系列	功能
FC4 DELETE	字符串	在STRING变量中删除
FC31 REPLACE	字符串	在STRING变量中替换
FC11 FIND	字符串	在STRING变量中查找
FC1 AD_DT_TM	浮点数数学运算	时间加持续时间
FC35 SB_DT_TM	浮点数数学运算	时间减持续时间
FC34 SB_DT_DT	浮点数数学运算	两个时间值相减
FC22 LIMIT	浮点数数学运算	限定值
FC25 MAX	浮点数数学运算	选择最大值
FC27 MIN	浮点数数学运算	选择最小值
FC36 SEL	浮点数数学运算	二进制选择

有关符合IEC的通讯块的更多信息，请参考系统功能块(SFB/SFC)的上下文相关帮助中的通讯功能(参见：*S7通讯块和S7基本通讯块之间的差别*)



## 24.2 IEC功能的技术数据

### 存储器要求

下表显示每个国际电工技术委员会(IEC)功能所需要的工作存储器字节数、装入存储器字节数和本地数据所需的字节数。

功能号	名称	大小(字节数)		本地数据 (字节数)
		工作存储器	装载存储器	
FC3	D_TOD_DT	634	810	12
FC6	DT_DATE	340	466	10
FC7	DT_DAY	346	472	10
FC8	DT_TOD	114	210	6
FC33	S5TI_TIM	94	208	2
FC40	TIM_S5TI	104	208	6
FC16	I_STRNG	226	340	10
FC5	DI_STRNG	314	440	18
FC30	R_STRNG	528	684	28
FC38	STRNG_I	292	420	12
FC37	STRNG_DI	310	442	12
FC39	STRNG_R	828	1038	30
FC9	EQ_DT	96	194	2
FC12	GE_DT	174	288	4
FC14	GT_DT	192	310	4
FC18	LE_DT	168	280	4
FC23	LT_DT	192	310	4
FC28	NE_DT	96	194	2
FC10	EQ_STRNG	114	220	4
FC13	GE_STRNG	162	282	8
FC15	GT_STRNG	158	278	8
FC19	LE_STRNG	162	282	8
FC24	LT_STRNG	158	278	8
FC29	NE_STRNG	150	266	8
FC21	LEN	38	132	2
FC20	LEFT	200	320	8
FC32	右箭头	230	350	8
FC26	MID	302	390	8
FC2	CONCAT	358	452	14
FC17	INSERT	488	644	20
FC4	DELETE	376	512	8
FC31	替换	562	726	20
FC11	查找	236	360	14
FC1	AD_DT_TM	1350	1590	22
FC35	SB_DT_TM	1356	1596	22

功能号	名称	大小(字节数)		本地数据 (字节数)
		工作存储器	装载存储器	
FC34	SB_DT_DT	992	1178	30
FC22	LIMIT	426	600	12
FC25	MAX	374	532	8
FC27	MIN	374	532	8
FC36	SEL	374	560	8

## 24.3 日期和时间作为复杂数据类型

### DATE\_AND\_TIME的实际参数

DATE\_AND\_TIME数据类型与ARRAY、STRING和STRUCT一起都属于复杂数据类型类别。复杂数据类型的允许存储区域是数据块(D)区域和本地数据(L)区域。

因为DATE\_AND\_TIME是复杂数据类型，在语句中使用DATE\_AND\_TIME作为形式参数时只能以下列形式提供实际参数：

- 指定块的变量声明表中的特定于块的符号
- 数据块的符号名称，例如“DB\_sys\_info.System\_Time”，由以下部分组成：
  - 在符号表中为数据块号定义的名称(例如DB5的名称为“DB\_sys\_info”)
  - 在数据块中为DATE\_AND\_TIME元素定义的名称(例如DB5中含有的DATE\_AND\_TIME数据类型变量的名称为“System\_Time”)

不能将常数作为实际参数传送复杂数据类型(包括DATE\_AND\_TIME)的形式参数。也不能将绝对地址作为实际参数传送给DATE\_AND\_TIME。

## 24.4 时间功能

### 描述 FC1 AD\_DT\_TM

功能FC1向一个时间(格式DT)值加上一段时间(格式TIME)，产生一个新的时间值(格式DT)。时间(参数T)必须在从DT#1990-01-01-00: 00: 00.000到DT#2089-12-31-23: 59: 59.999的范围内。此功能不运行输入检查。如果加法结果超出了有效范围，则将结果限制为相应值，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
T	INPUT	DATE_AND_TIME	D、L	DT格式的时间
D	INPUT	TIME	I、Q、M、D、L、常数	TIME格式的持续时间
RET_VAL	OUTPUT	DATE_AND_TIME	D、L	总和，格式为DT

只能为输入参数T和输出参数分配一个符号定义的变量。

### 描述FC3 D\_TOD\_DT

功能FC3将数据格式DATE和TIME\_OF\_DAY(TOD)组合在一起，并将这些格式转换为数据类型格式DATE\_AND\_TIME(DT)。输入值IN1必须在限定值DATE#1990-01-01和DATE#2089-12-31之间。(不检查此值。) 此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
IN1	INPUT	DATE	I、Q、M、D、L、常数	输入变量，格式为DATE
IN2	INPUT	TIME_OF_DAY	I、Q、M、D、L、常数	输入变量，格式为TOD
RET_VAL	OUTPUT	DATE_AND_TIME	D、L	返回值，格式为DT

只能为返回值分配一个符号定义的变量。

**描述FC6 DT\_DATE**

功能FC6从格式DATE\_AND\_TIME提取数据类型格式DATE。DATE值必须在限定值DATE#1990-1-1和DATE#2089-12-31之间。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
IN	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
RET_VAL	OUTPUT	DATE	I、Q、M、D、L	返回值，格式为DATE

只能为输入分配一个符号定义的变量。

**描述 FC7 DT\_DAY**

功能FC7从格式DATE\_AND\_TIME提取周时。以数据类型格式INTEGER提供周时：

- 1 = 星期日
- 2 = 星期一
- 3 = 星期二
- 4 = 星期三
- 5 = 星期四
- 6 = 星期五
- 7 = 星期六

此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
IN	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
RET_VAL	OUTPUT	INT	I、Q、M、D、L	返回值，格式为INT

只能为输入分配一个符号定义的变量。

**描述 FC8 DT\_TOD**

功能FC8从格式DATE\_AND\_TIME提取数据类型格式TIME\_OF\_DAY。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
IN	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
RET_VAL	OUTPUT	TIME_OF_DAY	I、Q、M、D、L	返回值，格式为TOD

只能为输入分配一个符号定义的变量。

**描述FC33 S5TI\_TIM**

功能FC33将数据类型格式S5TIME转换为格式TIME。如果转换结果超出TIME范围，则将结果限制为相应值，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
IN	INPUT	S5TIME	I、Q、M、D、L、常数	输入变量，格式为S5TIME
RET_VAL	OUTPUT	TIME	I、Q、M、D、L	返回值，格式为TIME

**描述 FC34 SB\_DT\_DT**

功能FC34，两个时间值(格式DT)相减，结果为持续时间(格式TIME)。时间值必须位于从DT#1990-01-01-00: 00: 00.000到DT#2089-12-31-23: 59: 59.999的范围内。此功能不运行输入检查。如果第一个时间(参数DT1)大于(晚于)第二个(参数DT2)，则结果为正；如果第一个时间小于(早于)第二个，则结果为负。如果减法结果超出TIME范围，则将结果限制到相应值，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
DT1	INPUT	DATE_AND_TIME	D、L	第一个DT格式的时间
DT2	INPUT	DATE_AND_TIME	D、L	第二个DT格式的时间
RET_VAL	OUTPUT	TIME	I、Q、M、D、L	差格式为TIME

只能为输入参数分配一个符号定义的变量。

**描述 FC35 SB\_DT\_TM**

功能FC35从一个时间值中(格式DT)减去持续时间(格式TIME)，结果产生一个新的时间值(格式DT)。时间(参数T)必须在DT#1990-01-01-00: 00: 00.000和DT#2089-12-31-23: 59: 59.999之间。此功能不进行输入检查。如果减法结果超出了有效范围，则将结果限制为相应值，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
T	INPUT	DATE_AND_TIME	D、L	DT格式的时间
D	INPUT	TIME	I、Q、M、D、L、常数	TIME格式的持续时间
RET_VAL	OUTPUT	DATE_AND_TIME	D、L	差，格式为DT

只能为输入参数T和输出参数分配一个符号定义的变量。

## 24.5 比较DATE\_AND\_TIME变量

### 描述 FC9 EQ\_DT

功能FC9比较两个DATE\_AND\_TIME数据类型格式变量的内容，检查它们是否相等，并将比较结果输出为返回值。如果参数DT1的时间与参数DT2的时间相等，则返回值的信号状态为“1”。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
DT1	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
DT2	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

### 描述FC12 GE\_DT

功能FC12比较两个DATE\_AND\_TIME数据类型格式变量的内容，检查其中一个是否大于或等于另一个，并将比较结果作为返回值。如果参数DT1的时间大于(晚于)参数DT2的时间，或者两个时间相同，则返回值的信号状态为“1”。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
DT1	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
DT2	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。



**描述FC14 GT\_DT**

功能FC14比较两个DATE\_AND\_TIME数据类型格式变量的内容，检查其中一个是否大于另一个，并将比较结果作为返回值。如果参数DT1的时间大于(晚于)参数DT2的时间，则返回值的信号状态为“1”。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
DT1	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
DT2	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

**描述FC18 LE\_DT**

功能FC18比较两个DATE\_AND\_TIME数据类型格式变量的内容，检查其中一个是否小于或等于另一个，并将比较结果作为返回值。如果参数DT1的时间小于(早于)参数DT2的时间，或者两个时间相同，则返回值的信号状态为“1”。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
DT1	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
DT2	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

**描述FC23 LT\_DT**

功能FC23比较两个DATE\_AND\_TIME数据类型格式变量的内容，检查其中一个是否小于另一个，并将比较结果作为返回值。如果参数DT1的时间小于(早于)参数DT2的时间，则返回值的信号状态为“1”。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
DT1	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
DT2	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

**描述FC28 NE\_DT**

功能FC28比较两个DATE\_AND\_TIME数据类型格式变量的内容，检查它们是否不相等，并将比较结果作为返回值。如果参数DT1的时间不等于参数DT2的时间，则返回值的信号状态为“1”。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
DT1	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
DT2	INPUT	DATE_AND_TIME	D、L	输入变量，格式为DT
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

## 24.6 比较STRING变量

### 描述FC10 EQ\_STRNG

功能FC9比较两个STRING数据类型格式变量的内容，检查它们是否相等，并将比较结果作为返回值。如果参数S1的字符串与参数S2的字符串相同，则返回值的信号状态为“1”。

此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
S1	INPUT	STRING	D、L	输入变量，格式为STRING
S2	INPUT	STRING	D、L	输入变量，格式为STRING
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

### 描述FC13 GE\_STRNG

功能FC13比较两个STRING数据类型格式变量的内容，检查第一个是否大于或等于另一个，并将比较结果作为返回值。如果参数S1的字符串大于或等于参数S2的字符串，则返回值的信号状态为“1”。

使用字符的ASCII码比较字符(例如“a”大于“A”)。从左开始比较。第一个不同的字符决定比较结果。如果开始字符都相同，则较长的字符串较大。

此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
S1	INPUT	STRING	D、L	输入变量，格式为STRING
S2	INPUT	STRING	D、L	输入变量，格式为STRING
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

**描述FC15 GT\_STRNG**

功能FC15比较两个STRING数据类型格式变量的内容，检查第一个是否大于另一个，并将比较结果作为返回值。如果参数S1的字符串大于参数S2的字符串，则返回值的信号状态为“1”。

使用字符的ASCII码比较字符(例如“a”大于“A”)。从左开始比较。第一个不同的字符决定比较结果。如果开始字符都相同，则较长的字符串较大。

此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
S1	INPUT	STRING	D、L	输入变量，格式为STRING
S2	INPUT	STRING	D、L	输入变量，格式为STRING
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

**描述FC19 LE\_STRNG**

功能FC19比较两个STRING数据类型格式变量的内容，检查第一个是否小于或等于另一个，并将比较结果作为返回值。如果参数S1的字符串小于或等于参数S2的字符串，则返回值的信号状态为“1”。

使用字符的ASCII码比较字符(例如“a”小于“A”)。从左开始比较。第一个不同的字符决定比较结果。如果开始字符都相同，则较短的字符串较小。

此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
S1	INPUT	STRING	D、L	输入变量，格式为STRING
S2	INPUT	STRING	D、L	输入变量，格式为STRING
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

**描述FC24 LT\_STRNG**

功能FC24比较两个STRING数据类型格式变量的内容，检查第一个是否小于另一个，并将比较结果作为返回值。如果参数S1的字符串小于参数S2的字符串，则返回值的信号状态为“1”。

使用字符的ASCII码比较字符(例如“a”小于“A”)。从左开始比较。第一个不同的字符决定比较结果。如果较长字符串的左侧部分和较短字符串相同,则较短的字符串较小。

此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
S1	INPUT	STRING	D、L	输入变量，格式为STRING
S2	INPUT	STRING	D、L	输入变量，格式为STRING
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

**描述FC29 NE\_STRNG**

功能FC29比较两个STRING数据类型格式变量的内容，检查它们是否不相等，并将比较结果作为返回值。如果参数S1的字符串不等于参数S2的字符串，则返回值的信号状态为“1”。

此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
S1	INPUT	STRING	D、L	输入变量，格式为STRING
S2	INPUT	STRING	D、L	输入变量，格式为STRING
RET_VAL	OUTPUT	BOOL	I、Q、M、D、L	比较结果

只能为输入参数分配一个符号定义的变量。

## 24.7 编辑数值

### 描述FC22 LIMIT

功能FC22将变量的数值限制到可分配有参数的限定值。INT、DINT和REAL数据类型变量可作为输入值。所有分配有参数的变量都必须为相同的数据类型。使用ANY指针识别变量类型。下限值(参数MN)必须小于/等于上限值(参数MX)。

如果下列任一条件为真，则输出值保持不变，并将状态字的二进制结果(BR)位设置为“0”：

- 已分配有参数的变量的数据类型无效
- 所有分配参数的变量不具有相同的数据类型
- 下限值大于上限值
- REAL变量不能代表有效浮点数。

参数	声明	数据类型	存储区域	描述
MN	INPUT	ANY	I、Q、M、D、L	下限
IN	INPUT	ANY	I、Q、M、D、L	输入变量
MX	INPUT	ANY	I、Q、M、D、L	上限
RET_VAL	OUTPUT	ANY	I、Q、M、D、L	限定的输出变量

### 描述FC25 MAX

功能FC25选择三个数值型变量值中的最大值。INT、DINT和REAL数据类型变量可作为输入值。所有分配参数的变量都必须为相同的数据类型。使用ANY指针识别变量类型。

如果下列任一条件为真，则输出值保持不变，并将状态字的二进制结果(BR)位设置为“0”：

- 已分配有参数的变量的数据类型无效
- 所有分配参数的变量不具有相同的数据类型
- REAL变量不能代表有效浮点数。

参数	声明	数据类型	存储区域	描述
IN1	INPUT	ANY	I、Q、M、D、L	第一个输入值
IN2	INPUT	ANY	I、Q、M、D、L	第二个输入值
IN3	INPUT	ANY	I、Q、M、D、L	第三个输入值
RET_VAL	OUTPUT	ANY	I、Q、M、D、L	输入值的最大值

## 24.8 STL示例

```
CALL FC 25
  IN1      : = P#M 10.0 DINT 1
  IN2      : = MD20
  IN3      : = P#DB1.DBX 0.0 DINT 1
  RET_VAL  : = P#M 40.0 DINT 1
=
M 0.0
```

注意:

必须为ANY指针输入允许的数据类型INT、DINT和REAL。也允许输入如“MD20”这样的参数，但必须“Symbol”定义“MD20”的相应数据类型。

### 描述FC27 MIN

功能FC27选择三个数值型变量值中的最小值。INT、DINT和REAL数据类型变量可作为输入值。所有分配参数的变量都必须为相同的数据类型。使用ANY指针识别变量类型。

如果下列任一条件为真，则输出值保持不变，并将状态字的二进制结果(BR)位设置为“0”:

- 已分配有参数的变量的数据类型无效
- 所有分配参数的变量不具有相同的数据类型
- REAL变量不能代表有效浮点数。

参数	声明	数据类型	存储区域	描述
IN1	INPUT	ANY	I、Q、M、D、L	第一个输入值
IN2	INPUT	ANY	I、Q、M、D、L	第二个输入值
IN3	INPUT	ANY	I、Q、M、D、L	第三个输入值
RET_VAL	OUTPUT	ANY	I、Q、M、D、L	输入值的最小值

## 24.9 STL示例

```
CALL FC 27
  IN1      : = P#M 10.0 DINT 1
  IN2      : = MD20
  IN3      : = P#DB1.DBX 0.0 DINT 1
  RET_VAL  : = P#M 40.0 DINT 1
=
M 0.0
```

注意:

必须为ANY指针输入允许的数据类型INT、DINT和REAL。也允许输入如“MD20”这样的参数，但必须“Symbol”定义“MD20”的相应数据类型。

### 描述FC36 SEL

功能FC36根据一个开关(参数G)在两个变量值中选择其一。符合数据位宽度要求的所有数据类型的变量均可作为参数IN0和IN1的输入值。例如字节、字和双字(不包括数据类型DT和STRING)。输入变量和输出变量必须为相同的数据类型。

如果下列任一条件为真，则输出值保持不变，并将状态字的二进制结果(BR)位设置为“0”:

- 已分配有参数的变量的数据类型无效
- 所有分配参数的变量不具有相同的数据类型
- REAL变量不能代表有效浮点数。

参数	声明	数据类型	存储区域	描述
G	INPUT	BOOL	I、Q、M、D、L	选择开关
IN0	INPUT	ANY	I、Q、M、D、L	第一个输入值
IN1	INPUT	ANY	I、Q、M、D、L	第二个输入值
RET_VAL	OUTPUT	ANY	I、Q、M、D、L	所选择的输入值



## 24.10 编辑STRING变量

### 描述FC2 CONCAT

功能FC2将两个STRING变量连接在一起，组成一个字符串。如果结果字符串比在输出参数中提供的变量长，则将结果字符串限制到设置的最大长度，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
IN1	INPUT	STRING	D、L	输入变量，格式为STRING
IN2	INPUT	STRING	D、L	输入变量，格式为STRING
RET_VAL	OUTPUT	STRING	D、L	连接字符串

只能为参数分配一个符号定义的变量。

### 描述FC4 DELETE

功能FC4从字符串中位置P的字符(含此字符)开始删除多个(L)字符。如果L和/或P等于零，或者P大于输入字符串的当前长度，则返回输入字符串。如果L加P的和大于输入字符串，则一直删除到输入字符串的末尾。如果L和/或P为负数，则返回空字符串，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
IN	INPUT	STRING	D、L	要从其中进行删除操作的STRING变量
L	INPUT	INT	I、Q、M、D、L、 常数	要删除的字符数
P	INPUT	INT	I、Q、M、D、L、 常数	要删除的第一个字符的位置
RET_VAL	OUTPUT	STRING	D、L	结果字符串

只能为输入参数IN和输出参数分配一个符号定义的变量。

**描述FC11 FIND**

功能FC11提供第二个字符串(IN2)在第一个字符串(IN1)中的位置。查找从左侧开始；报告第一次出现字符串的位置。如果在第一个字符串中未找到第二个字符串，则返回零。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
IN1	INPUT	STRING	D、L	要在其中进行查找的STRING变量
IN2	INPUT	STRING	D、L	要查找的STRING变量
RET_VAL	OUTPUT	INT	I、Q、M、D、L	找到的字符串位置

只能为输入参数IN1和IN2分配一个符号定义的变量。

**描述FC17 INSERT**

功能FC17将参数IN2的字符串插入到参数IN1的字符串中位置P的字符之后。如果P等于零，则将第二个字符串插入到第一个字符串之前。如果P大于第一个字符串的当前长度，则将第二个字符串附加到第一个字符串。如果P为负，则输出空字符串，并将状态字的二进制结果(BR)位设置为“0”。如果结果字符串比在输出参数中提供的变量长，也将二进制结果位设置为“0”；在这种情况下，将结果字符串限制到设置的最大长度。

参数	声明	数据类型	存储区域	描述
IN1	INPUT	STRING	D、L	要向其中进行插入操作的STRING变量
IN2	INPUT	STRING	D、L	要插入的STRING变量
P	INPUT	INT	I、Q、M、D、L、 常数	插入位置
RET_VAL	OUTPUT	STRING	D、L	结果字符串

只能为输入参数IN1和IN2及输出参数分配一个符号定义的变量。

**描述FC20 LEFT**

功能FC20提供字符串的前L个字符(其中L代表数字)。如果L大于STRING变量的当前长度,则返回输入值。如果L = 0并且输入值为空字符串,则返回空字符串。如果L为负数,则返回空字符串,并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
IN	INPUT	STRING	D、L	输入变量,格式为STRING
L	INPUT	INT	I、Q、M、D、L、 常数	左侧字符串的长度
RET_VAL	OUTPUT	STRING	D、L	输出变量的格式为STRING

只能为参数IN和返回值分配一个符号定义的变量。

**描述FC21 LEN**

STRING变量包含两个长度:最大长度(定义变量时在方括号中给出)和当前长度(当前有效字符数)。当前长度必须小于或等于最大长度。字符串占用的字节数为最大长度加2。

功能FC21输出字符串的当前长度(有效字符数)作为返回值。空字符串(' ')的长度为零。最大长度为254。此功能不报告任何出错。

参数	声明	数据类型	存储区域	描述
S	INPUT	STRING	D、L	输入变量,格式为STRING
RET_VAL	OUTPUT	INT	I、Q、M、D、L	有效字符数

只能为输入参数分配一个符号定义的变量。

### 描述FC26 MID

功能FC26提供字符串的中间部分(从第P个字符(含第P个字符)开始的L个字符)。如果L加P的和超出STRING变量的当前长度,则返回的字符串为从第P个字符到输入值末尾。在所有其它情况下(P超出当前长度,P和/或L等于零或为负),返回空字符串,并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
IN	INPUT	STRING	D、L	输入变量,格式为STRING
L	INPUT	INT	I、Q、M、D、L、 常数	中间字符串的长度
P	INPUT	INT	I、Q、M、D、L、 常数	第一个字符的位置
RET_VAL	OUTPUT	STRING	D、L	输出变量的格式为STRING

只能为参数IN和返回值分配一个符号定义的变量。

### 描述FC31 REPLACE

功能FC31使用第二个字符串(IN2)替换第一个字符串(IN1)中从位置P(含此字符)开始的多个(L)字符。如果L等于零,则返回第一个字符串。如果P等于零或一,则从第一个字符(含此字符)开始替换字符串。如果P超出第一个字符串,则将第二个字符串附加到第一个字符串。如果L和/或P为负数,则返回空字符串,并将状态字的二进制结果(BR)位设置为“0”。如果结果字符串比在输出参数中提供的变量长,也将二进制结果位设置为“0”;在这种情况下,将结果字符串限制到设置的最大长度。

参数	声明	数据类型	存储区域	描述
IN1	INPUT	STRING	D、L	要在其中进行替换的STRING变量
IN2	INPUT	STRING	D、L	要插入的STRING变量
L	INPUT	INT	I、Q、M、D、L、 常数	要替换的字符数
P	INPUT	INT	I、Q、M、D、L、 常数	要替换的第一个字符的位置
RET_VAL	OUTPUT	STRING	D、L	结果字符串

只能为输入参数IN1和IN2及输出参数分配一个符号定义的变量。

**描述FC32 RIGHT**

功能FC32提供字符串的后L个字符(其中L代表数字)。如果L大于STRING变量的当前长度,则返回输入值。如果L = 0并且输入值为空字符串,则返回空字符串。如果L为负数,则返回空字符串,并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
IN	INPUT	STRING	D、L	输入变量,格式为STRING
L	INPUT	INT	I、Q、M、D、L、 常数	右侧字符串的长度
RET_VAL	OUTPUT	STRING	D、L	输出变量的格式为STRING

只能为参数IN和返回值分配一个符号定义的变量。

## 24.11 转换数据类型格式

### 描述FC5 DI\_STRNG

功能FC5将DINT数据类型格式变量转换为字符串。显示的字符串以符号开头。如果在返回参数中给出的变量太短，则不进行转换，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
I	INPUT	DINT	I、Q、M、D、L、 常数	输入值
RET_VAL	OUTPUT	STRING	D、L	结果字符串

只能为输出参数分配一个符号定义的变量。

### 描述FC16 I\_STRNG

功能FC16将INT数据类型格式变量转换为字符串。显示的字符串以符号开头。如果在返回参数中给出的变量太短，则不进行转换，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
I	INPUT	INT	I、Q、M、D、L、 常数	输入值
RET_VAL	OUTPUT	STRING	D、L	结果字符串

只能为输出参数分配一个符号定义的变量。

### 描述FC30 R\_STRNG

功能FC30将REAL数据类型格式变量转换为字符串。显示的字符串有14位：

±v.nnnnnnnE±xx	±	符号
	v	小数点前1位
	n	小数点后7位
	x	2位指数

如果在返回参数中给出的变量太短，或者参数IN未提供有效的浮点数，则不进行转换，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
IN	INPUT	REAL	I、Q、M、D、L、 常数	输入值
RET_VAL	OUTPUT	STRING	D、L	结果字符串

只能为输出参数分配一个符号定义的变量。

### 描述FC37 STRNG\_DI

功能FC37将字符串转换为DINT数据类型格式变量。字符串中的第一个字符可以是符号或数字，随后的字符必须是数字。如果字符串的长度等于零或大于11，或者字符串中有无效字符，则不进行转换，并将状态字的二进制结果(BR)位设置为“0”。如果转换结果超出DINT范围，则将结果限制到相应值，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
S	INPUT	STRING	D、L	输入字符串
RET_VAL	OUTPUT	DINT	I、Q、M、D、L	结果

只能为输入参数分配一个符号定义的变量。

**描述FC38 STRNG\_I**

功能FC38将字符串转换为INT数据类型格式变量。字符串中的第一个字符可以是符号或数字，随后的字符必须是数字。如果字符串的长度等于零或大于6，或者字符串中有无效字符，则不进行转换，并将状态字的二进制结果(BR)位设置为“0”。如果转换结果超出INT范围，则将结果限制到相应值，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
S	INPUT	STRING	D、L	输入字符串
RET_VAL	OUTPUT	INT	I、Q、M、D、L	结果

只能为输入参数分配一个符号定义的变量。

**描述FC39 STRNG\_R**

功能FC39将字符串转换为REAL数据类型格式变量。字符串必须为以下格式：

±v.nnnnnnnE±xx	±	符号
	v	小数点前1位
	n	小数点后7位
	x	2位指数

如果字符串长度小于14，或者不是如上所示的结构，则不进行任何转换，并将状态字的二进制结果(BR)位设置为“0”。如果转换结果超出REAL范围，则将结果限制到相应值，并将状态字的二进制结果(BR)位设置为“0”。

参数	声明	数据类型	存储区域	描述
S	INPUT	STRING	D、L	输入字符串
RET_VAL	OUTPUT	REAL	I、Q、M、D、L	结果

只能为输入参数分配一个符号定义的变量。



## 25 用于集成控制的SFB

### 25.1 使用SFB41/FB41 “CONT\_C”实现连续控制

#### 引言

SFB/FB “CONT\_C”(连续控制器)在SIMATIC S7可编程逻辑控制器上使用，通过持续的输入和输出变量来控制工艺过程。在参数分配期间，可以通过激活或取消激活PID控制器的子功能使控制器适应过程的需要。使用参数分配工具可以轻松完成分配(菜单路径：**开始 > Simatic > Step7 > 分配PID控制参数**)。**开始 > Simatic > Step7 > 分配PID控制(英文)**中提供了在线电子手册。

#### 应用

可以使用该控制器作为PID固定设定值控制器或在多循环控制中作为层叠、混料或比率控制器。控制器的功能基于采样控制器的PID控制算法，采样控制器带有一个模拟信号；如果需要的话，还可以扩展控制器的功能，增加一个脉冲生成器环节，以产生脉宽调制的输出信号，用于带有比例执行器的两步或三步控制器。

---

#### 注意

只有在以固定时间间隔调用块时，在控制块中计算的才是正确的。为此，应该在周期性中断OB(OB30到OB38)中调用控制块。在CYCLE参数中输入采样时间。

---

#### 描述

除了设定值和过程值分支中的功能，SFB/FB还通过持续调节变量输出和手动影响操作值的选项实现了完整的PID控制器。

下文提供了对这些子功能的详细说明：

#### 设定值分支

以浮点格式在SP\_INT输入键入设定值。

#### 过程变量分支

可以外设(I/O)或以浮点格式输入过程变量。CRP\_IN功能根据以下公式将PV\_PER外设值转换为介于 -100和 +100 %间的浮点格式值:

$$\text{Output of CPR\_IN} = \text{PV\_PER} * \frac{100}{27648}$$

PV\_NORM函数根据下面的公式标准化CRP\_IN的输出:

$$\text{PV\_NORM的输出} = (\text{CPR\_IN的输出}) * \text{PV\_FAC} + \text{PV\_OFF}$$

PV\_FAC的缺省值为1, PV\_OFF的缺省值为0。

### 误差信号

设定值和过程变量间的差异就是出错信。为消除由于操作变量量化导致的小幅恒定振荡(例如, 在使用PULSEGEN进行脉宽调制时), 将死区应用于出错信号(DEADBAND)。如果DEADB\_W = 0, 则死区功能关闭。

### PID算法

PID算法用于定位计算。比例、积分(INT)和微分(DIF)操作以并联方式连接, 因而可以分别激活或取消激活。这使对P、PI、PD和PID控制器进行组态成为可能。还可以对纯I和D控制器进行组态。

### 手动值

可以在手动和自动模式间进行切换。在手动模式下, 调节变量被修正到手动选择的数值。积分器(INT)内部设置为LMN - LMN\_P - DISV, 微分单元(DIF)设置为0并在内部进行匹配。这样, 切换到自动模式就不会导致操作值的突然变化。

### 操作值

使用LMNLIMIT函数, 可以将操作值限制到所选择的数值上。输入变量超过限制时, 信号位会给予指示。

LMN\_NORM函数根据下面的公式标准化LMNLIMIT的输出:

$$\text{LMN} = (\text{LMNLIMIT的输出}) * \text{LMN\_FAC} + \text{LMN\_OFF}$$

LMN\_FAC的缺省值为1, LMN\_OFF的缺省值为0。

操作值也可以使用外部值格式。CPR\_OUT功能根据以下公式将浮点值LMN转换为外设值:

$$\text{LMN\_PER} = \text{LMN} * \frac{27648}{100}$$

### 前馈控制

可以在DISV输入前馈干扰变量。

## 初始化

SFB41 “CONT\_C”有一个在输入参数COM\_RST = TRUE时自动运行的初始化例行程序。

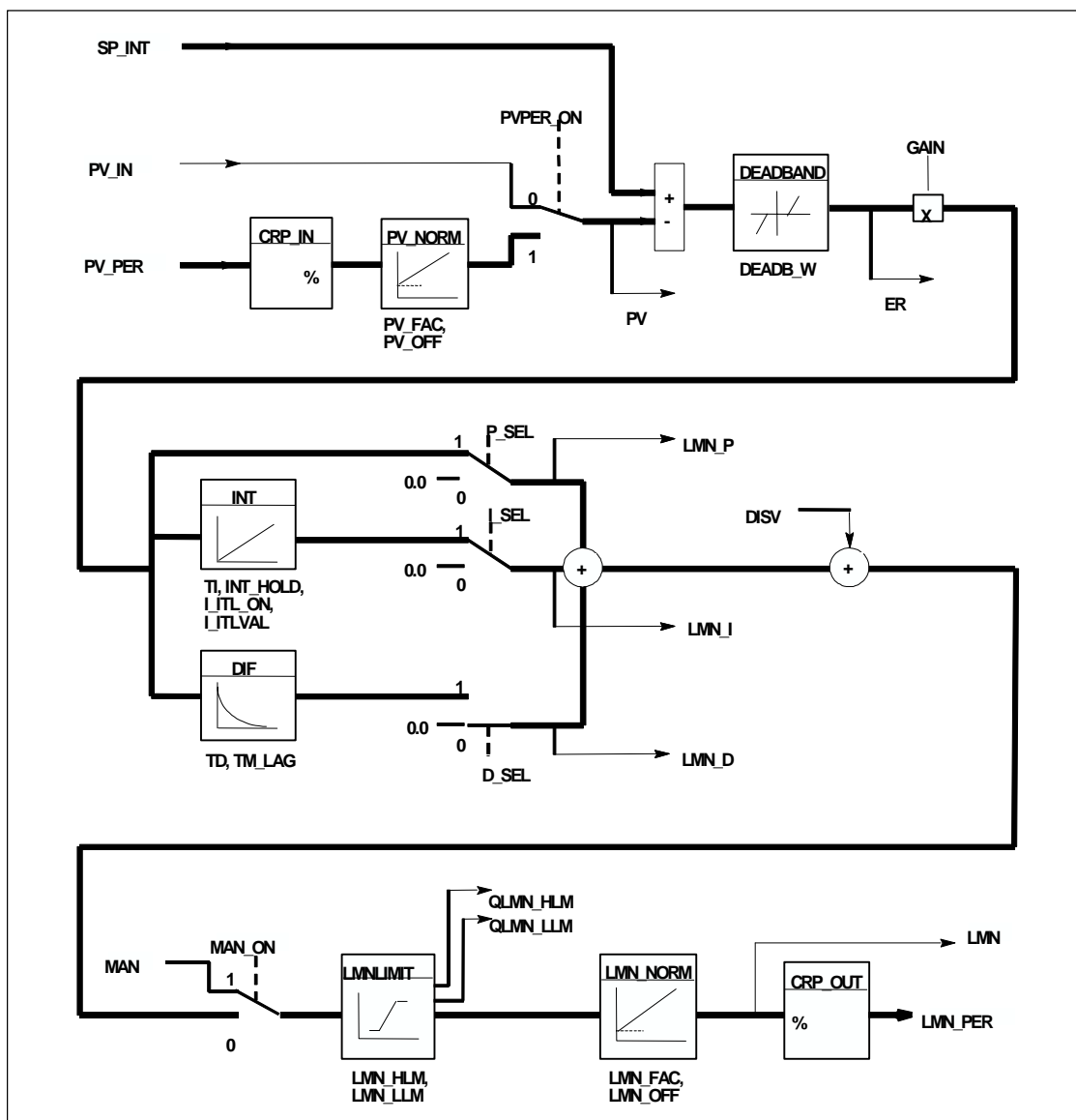
在初始化过程中，将把积分器内部设置为初始化值I\_ITVAL。当在周期性中断优先级中调用积分器时，它便从这个数值开始，继续工作。

所有其它输出都被设置成各自的缺省值。

## 出错信息

不使用出错输出参数RET\_VAL。

## CONT\_C方框图



## 输入参数

下表包含对SFB 41/FB 41 “CONT\_C”输入参数的说明。

参数	数据类型	取值范围:	缺省值	描述
COM_RST	BOOL		FALSE	<b>COMPLETE RESTART</b> 该块有一个在设置输入COM_RST时自动执行的初始化程序。
MAN_ON	BOOL		TRUE	<b>MANUAL VALUE ON</b> 如果设置输入“启用比例作用”，将中断控制回路。手动值作为操作值进行设置。
PVPER_ON	BOOL		FALSE	<b>PROCESS VARIABLE PERIPHERAL ON</b> 如果从I/O读取过程变量，必须将输入PV_PER连接到I/O，且必须设置输入“启用过程变量外设”。
P_SEL	BOOL		TRUE	<b>PROPORTIONAL ACTION ON</b> 可以在PID算法中单独激活或取消激活PID操作。如果设置输入“启用积分作用”，将启用P操作。
I_SEL	BOOL		TRUE	<b>INTEGRAL ACTION ON</b> 可以在PID算法中单独激活或取消激活PID操作。如果设置输入“积分作用打开”，将启用I操作。
INT_HOLD	BOOL		FALSE	<b>INTEGRAL ACTION HOLD</b> 设置输入“积分作用暂停”可以“冻结”积分器的输出。
I_ITL_ON	BOOL		FALSE	<b>INITIALIZATION OF THE INTEGRAL ACTION ON</b> 设置输入“积分作用初始化打开”可以将积分器的输出连接到输入I_ITL_VAL。
D_SEL	BOOL		FALSE	<b>DERIVATIVE ACTION ON</b> 可以在PID算法中单独激活或取消激活PID操作。如果设置输入“启用微分作用”，将启用D操作。
循环	TIME	>= 1ms	T#1s	<b>SAMPLING TIME</b> 块调用间的时间必须为常数。“采样时间”输入指定块调用之间的时间。
SP_INT	REAL	-100.0至 +100.0 (%)或物理值 1)	0.0	<b>INTERNAL SETPOINT</b> “内部设定值”输入用于指定设定值。
PV_IN	REAL	-100.0至 +100.0 (%)或物理值 1)	0.0	<b>PROCESS VARIABLE IN</b> 可以在“过程变量输入”输入设置初始化值，也可以连接浮点格式的外部过程变量。

参数	数据类型	取值范围:	缺省值	描述
PV_PER	WORD		W#16#0000	PROCESS VARIABLE PERIPHERAL 将I/O格式的过程变量连接到“过程变量外设”输入处的控制器。
MAN	REAL	-100.0至 +100.0 (%) 或物理 值 2)	0.0	MANUAL VALUE “手动值”输入用于通过操作员界面功能设置手动值。
GAIN	REAL		2.0	PROPORTIONAL GAIN “比例值”输入指定控制器增益。
TI	TIME	>= CYCLE	T#20s	RESET TIME “复位时间”输入决定积分器的时间响应。
TD	TIME	>= CYCLE	T#10s	DERIVATIVE TIME “微分时间”输入决定微分单元的时间响应。
TM_LAG	TIME	>= CYCLE/2	T#2s	TIME LAG OF THE DERIVATIVE ACTION D操作的算法包括可以在“微分作用的时间延迟”输入分配的时间延迟。
DEADB_W	REAL	>= 0.0(%) 或物理 值 1)	0.0	DEAD BAND WIDTH 将死区应用于出错。“死区带宽”输入决定死区的大小。
LMN_HLM	REAL	LMN_LLM ...100.0(%) 或物理 值 2)	100.0	MANIPULATED VALUE HIGH LIMIT 操作值始终受上限和下限的限制。“操作值上限”输入指定上限。
LMN_LLM	REAL	-100.0... LMN_HLM(%) 或物理 值 2)	0.0	MANIPULATED VALUE LOW LIMIT 操作值始终受上限和下限的限制。“操作值下限”输入指定下限。
PV_FAC	REAL		1.0	PROCESS VARIABLE FACTOR “过程变量因子”输入与过程变量相乘。该输入用于调整过程变量范围。
PV_OFF	REAL		0.0	PROCESS VARIABLE OFFSET 将“过程变量偏移量”输入与过程变量相加。该输入用于调整过程变量范围。
LMN_FAC	REAL		1.0	MANIPULATED VALUE FACTOR 将“操作值因子”输入与操作值相乘。该输入用于调整操作值范围。
LMN_OFF	REAL		0.0	MANIPULATED VALUE OFFSET 将“操作值偏移量”与操作值相加。该输入用于调整操作值范围。

参数	数据类型	取值范围:	缺省值	描述
I_ITLVAL	REAL	-100.0至 +100.0 (%) 或物理 值 2)	0.0	INITIALIZATION VALUE OF THE INTEGRAL ACTION 可以在输入I_ITL_ON设置积分器的输出。将初始化值应用于输入“积分作用的初始化值”。
DISV	REAL	-100.0至 +100.0 (%) 或物理 值 2)	0.0	DISTURBANCE VARIABLE 为进行前馈控制，将干扰变量连接到输入“干扰变量”。

- 1) 设定值和过程变量分支中的参数具有相同的单位
- 2) 操作值分支的参数具有相同的单位

## 输出参数

下表包含对SFB41/FB41 “CONT\_C”输出参数的说明。

参数	数据类型	取值范围:	缺省值	描述
LMN	REAL		0.0	MANIPULATED VALUE 有效的操作值为“操作值”输出处的浮点格式输出。
LMN_PER	WORD		W#16#0000	MANIPULATED VALUE PERIPHERAL 将I/O格式的操作值连接到“操作值外设”输出的控制器。
QLMN_HLM	BOOL		FALSE	HIGH LIMIT OF MANIPULATED VALUE REACHED 操作值始终受上限和下限的限制。如果输出为“达到操作值上限”，则表明已超过上限。
QLMN_LLM	BOOL		FALSE	LOW LIMIT OF MANIPULATED VALUE REACHED 操作值始终受上限和下限的限制。如果输出为“达到操作值下限”，则表明已超过下限。
LMN_P	REAL		0.0	PROPORTIONAL COMPONENT “比例组件”输出包含调节变量的比例组件。
LMN_I	REAL		0.0	INTEGRAL COMPONENT “积分组件”输出包含操作值的积分组件。

参数	数据类型	取值范围:	缺省值	描述
LMN_D	REAL		0.0	DERIVATIVE COMPONENT “微分组件”输出包含 操作值的微分组件。
PV	REAL		0.0	PROCESS VARIABLE 有效过程变量为“过程变量”输出处的输出。
ER	REAL		0.0	ERROR SIGNAL 有效出错为“出错信号”输出处的输出。

## 25.2 使用SFB42/FB42 “CONT\_S”进行步控制

### 引言

SFB/FB CONT\_S”(步控制器)在SIMATIC 7可编程逻辑控制器上使用，通过集成执行机构的数字操作值输出信号来控制工艺过程。在参数分配期间，可以通过激活或取消激活PI步控制器的子功能使控制器适应过程的需要。使用参数分配工具可以轻松完成分配(菜单路径：**开始 > Simatic > Step7 > 分配PID控制参数**)。**开始 > Simatic > Step7 > 分配PID控制(英文)**中提供了在线电子手册。

### 应用

可以将该控制器作为PI固定设定值控制器使用，或在层叠、混料或比率控制器的辅助控制回路中使用，但不能作为主控制器使用。控制器的功能基于采样控制器的PI控制算法，通过用于从模拟量驱动信号生成二进制输出信号的函数来实现。

从CPU 314 IFM的FB V1.5或V1.1.0开始增加了以下功能：

TI = T#0ms时，可以禁用控制器的积分组件，从而可以将块作为比例控制器使用。

由于该控制器在没有任何位置反馈信号的情况下工作，内部计算的调节变量将不会与信号控制元素位置完全相符。如果调节变量(ER \* GAIN)为负值，将进行调整。随后该控制器会设置输出QLMNDN(操作值信号下限)，直到设置了LMNR\_LS(位置反馈信号下限)为止。

该控制器还可作为控制器层叠中的辅助执行机构使用。设定值输入SP\_INT用于分配控制元素位置。在这种情况下，必须将实际值输入和参数TI(集成时间)设置为零。一个应用实例是采用热量输出控制方法对温度进行调节，该方法通过阀进行脉冲暂停控制和冷却能力控制。在这种情况下，要完全关闭阀，应将调节变量(ER \* GAIN)设置为负值。

---

### 注意

只有在以固定时间间隔调用块时，在控制块中计算的才是正确的。为此，应该在周期性中断OB(OB30到OB38)中调用控制块。在CYCLE参数中输入采样时间。

---



## 描述

除了过程值分支中的功能，SFB还通过数字操作值输出和手动影响操作值选项实现了完整的PI控制器。步控制器在没有位置反馈信号的情况下工作。

下文详细描述了部分功能：

### 设定值分支

以浮点格式在SP\_INT输入键入设定值。

### 过程变量分支

可以外设(I/O)或以浮点格式输入过程变量。CRP\_IN功能根据以下公式将PV\_PER外设值转换为介于 -100和 +100 %间的浮点格式值：

$$\text{Output of CPR\_IN} = \text{PV\_PER} * \frac{100}{27648}$$

PV\_NORM函数根据下面的公式标准化CRP\_IN的输出：

$$\text{PV\_NORM的输出} = (\text{CPR\_IN的输出}) * \text{PV\_FAC} + \text{PV\_OFF}$$

PV\_FAC的缺省值为1，PV\_OFF的缺省值为0。

### 误差信号

设定值和过程变量间的差异就是出错信号。为消除由于操作变量量化(例如，由于制动器阀操作值的精度有限)导致的小幅恒定振荡，将死区应用于出错信号(DEADBAND)。如果DEADB\_W = 0，则死区功能关闭。

### PI步算法

SFB/FB在没有位置反馈信号的情况下工作。PI算法的I操作和假定的位置反馈信号在一个积分器(INT)中计算，并作为反馈值与其余P操作进行比较。将差异应用于三步元素(THREE\_ST)和创建执行机构脉冲的脉冲发生器(PULSEOUT)。调整三步元素的阈值可以降低控制器的切换频率。

### 前馈控制

可以在DISV输入前馈干扰变量。

## 初始化

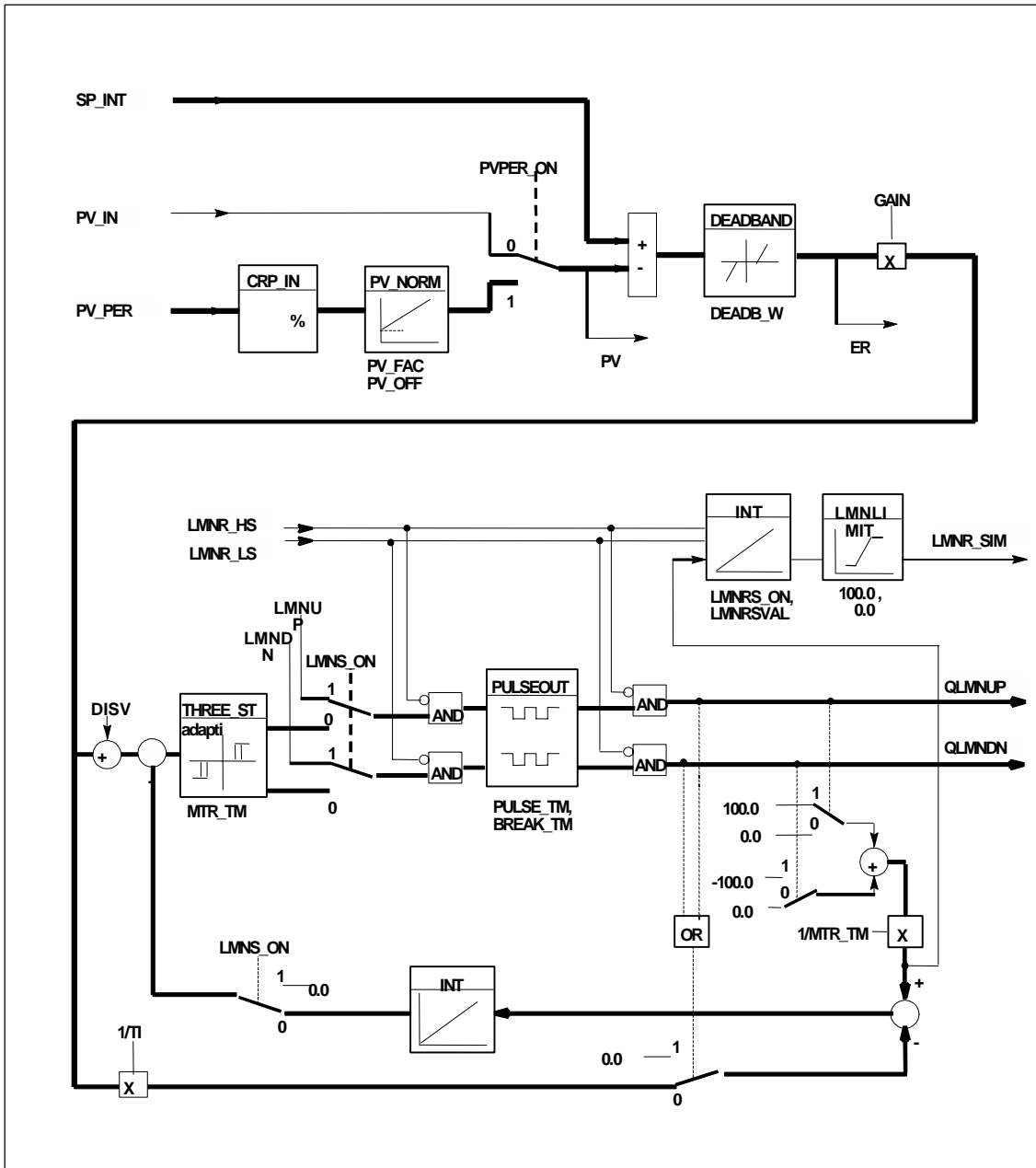
SFB/FB42 “CONT\_S”有一个在输入参数COM\_RST = TRUE时自动运行的初始化程序。

所有其它输出都被设置成各自的缺省值。

## 出错信息

不使用出错输出参数RET\_VAL。

方框图



## 输入参数

下表包含对SFB 42/FB 42 “CONT\_S”输入参数的说明。

参数	数据类型	值	缺省值	描述
COM_RST	BOOL		FALSE	COMPLETE RESTART 该块有一个在设置输入COM_RST时自动执行的初始化程序。
LMNR_HS	BOOL		FALSE	HIGH LIMIT OF POSITION FEEDBACK SIGNAL 将“执行机构位于上限止点”信号连接到“位置反馈信号的上限”输入。LMNR_HS=TRUE意味着执行器在上限处停止。
LMNR_LS	BOOL		FALSE	LOW LIMIT OF POSITION FEEDBACK SIGNAL 将“执行机构位于下限止点”信号连接到“位置反馈信号的下限”输入。LMNR_LS=TRUE表示执行机构位于下限置点。
LMNS_ON	BOOL		TRUE	MANUAL ACTUATING SIGNALS ON 激励信号处理在“启用手动激励信号”输入中切换为手动。
LMNUP	BOOL		FALSE	ACTUATING SIGNALS UP 使用手动激励值信号时，在输入“激励信号增加”设置输出信号QLMNUP。
LMNDN	BOOL		FALSE	ACTUATING SIGNALS DOWN 使用手动激励值信号时，在输入“激励信号降低”设置输出信号QLMNDN。
PVPER_ON	BOOL		FALSE	PROCESS VARIABLE PERIPHERAL ON 如果从I/O读入过程变量，必须将输入PV_PER连接到I/O，且必须设置输入“启用过程变量外设”。
循环	TIME	>= 1ms	T#1s	SAMPLING TIME 块调用之间的时间必须为常数。“采样时间”输入指定块调用间的时间。
SP_INT	REAL	-100.0... +100.0 (%) 或物理值 1)	0.0	INTERNAL SETPOINT “内部设定值”输入用于指定设定值。
PV_IN	REAL	-100.0... +100.0 (%) 或物理值 1)	0.0	PROCESS VARIABLE IN 可以在“过程变量输入”输入中设置初始化值，也可以连接浮点格式的外部过程变量。
PV_PER	WORD		W#16#0000	PROCESS VARIABLE PERIPHERAL 将I/O格式的过程变量连接到“过程变量外设”输入处的控制器。

参数	数据类型	值	缺省值	描述
GAIN	REAL		2.0	PROPORTIONAL GAIN “比例增益”输入设置控制器增益。
TI	TIME	>= CYCLE	T#20s	RESET TIME “复位时间”输入决定积分器的时间响应。
DEADB_W	REAL	0.0...100.0 (%) 或物理值 1)	1.0	DEAD BAND WIDTH 将死区应用于出错。“死区带宽”输入决定死区的大小。
PV_FAC	REAL		1.0	PROCESS VARIABLE FACTOR 将“过程变量因子”输入与过程变量相乘。该输入用于调整过程变量范围。
PV_OFF	REAL		0.0	PROCESS VARIABLE OFFSET 将“过程变量偏移量”输入与过程变量相加。它以此来调整过程变量的范围。
PULSE_TM	TIME	>= CYCLE	T#3s	MINIMUM PULSE TIME 使用参数“最小脉冲时间”可以分配最小脉冲宽度。
BREAK_TM	TIME	>= CYCLE	T#3s	MINIMUM BREAK TIME 使用参数“最小中断时间”可以分配最小中断宽度。
MTR_TM	TIME	>= CYCLE	T#30s	MOTOR ACTUATING TIME 在“电机动作时间”参数中输入执行机构在挡块间移动所需的时间。
DISV	REAL	- 100.0...100.0(%) 或物理值 2)	0.0	DISTURBANCE VARIABLE 为进行前馈控制，将干扰变量连接到输入“干扰变量”。

- 1) 设定值和过程变量分支中的参数具有相同的单位
- 2) 操作值分支的参数具有相同的单位

## 输出参数

下表包含对SFB42/FB42 “CONT\_S”输出参数的说明。

参数	数据类型	值	缺省值	描述
QLMNUP	BOOL		FALSE	ACTUATING SIGNAL UP 如果设置输出“激励信号增加”，将打开激励阀。
QLMNDN	BOOL		FALSE	ACTUATING SIGNAL DOWN 如果设置输出“激励信号降低”，将打开激励阀。
PV	REAL		0.0	PROCESS VARIABLE 有效过程变量为“过程变量”输出处的输出。
ER	REAL		0.0	ERROR SIGNAL 有效出错为“出错信号”输出处的输出。

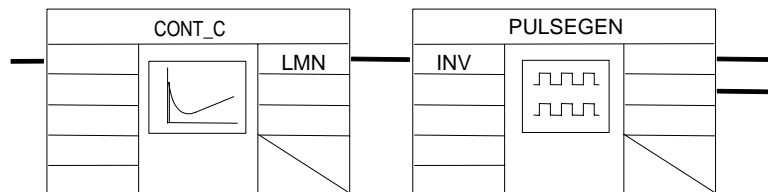
## 25.3 使用SFB43/FB43 “PULSEGEN”生成脉冲

### 引言

SFB43 PULSEGEN”(脉冲发生器)用于构建具有比例执行机构脉冲输出的PID控制器。文档 > 英文 > STEP7 PID控制中提供了电子手册。

### 应用

使用SFB/FB PULSEGEN”可以组态具有脉宽调制功能的PID两/三步控制器。该功能通常与连续控制器“CONT\_C”结合使用。



---

### 注意

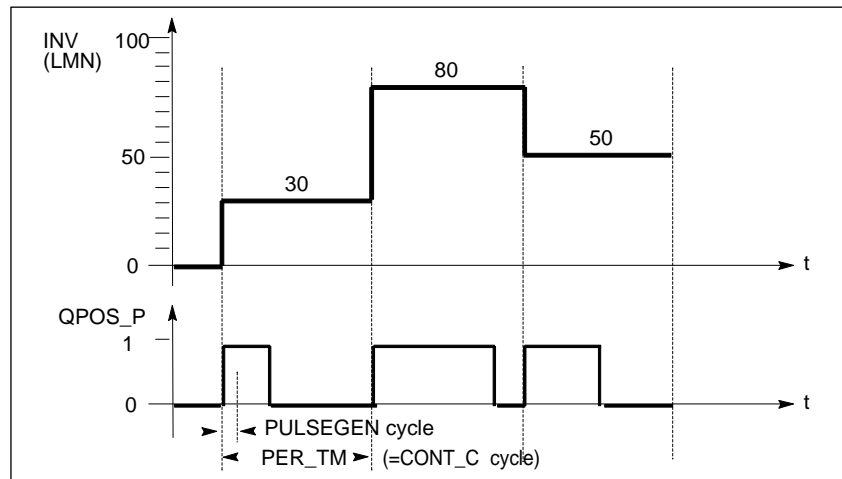
只有在以固定时间间隔调用块时，在控制块中计算的才是正确的。为此，应该在周期性中断OB(OB30到OB38)中调用控制块。在CYCLE参数中输入采样时间。

---

## 描述

PULSEGEN功能通过脉宽调制将输入变量INV = ID控制器的操作值)转换为具有恒定周期的脉冲列，从而与输入变量的更新周期及必须在PER\_TM中分配的周期相应。

在每个周期内，脉冲的持续时间和输入变量成比例。分配给PER\_TM的周期与SFB/FB “PULSEGEN”的处理周期不完全相同。PER\_TM周期由若干个SFB/FB “PULSEGEN”处理周期组成，因此可以将每个PER\_TM周期SFB/FB PULSEGEN”调用的数量作为脉宽调制精度的衡量标准。

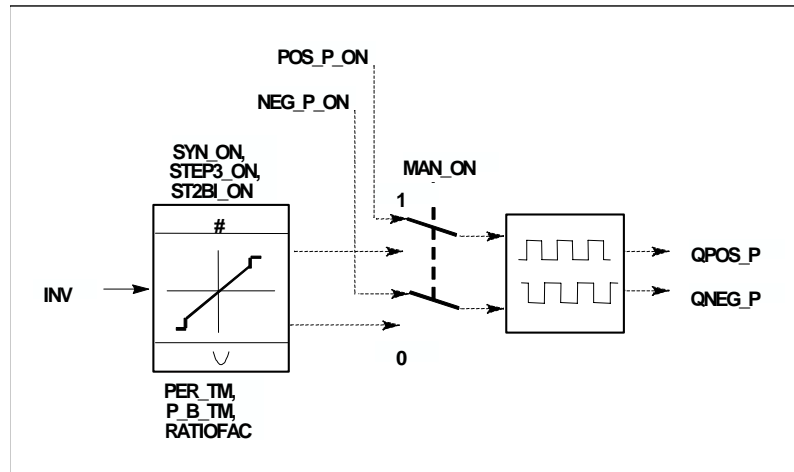


## 脉宽调制

输入变量为30%、每个PER\_TM有10个SFB/FB “PULSEGEN”调用的含义如下：

- 前三次SFB/FB “PULSEGEN”调用(10次调用的30%)在QPOS输出为“一”
- 后七次SFB/FB “PULSEGEN”调用(10次调用的70%)在QPOS输出为“零”

## 方框图



## 操作值的精度

“采样比率”为1: 10(CONT\_C调用与PULSEGEN调用之比)时，此实例中的操作值精度将限制在10%，换言之，设置的输入值INV只能在QPOS输出通过以10%为步长的脉冲宽度进行模拟。

精度随每个CONT\_C调用的SFB/FB “PULSEGEN”调用数的增加而增加。

例如，如果调用PULSEGEN的频率是调用CONT\_C频率的100倍，则获得的操作值范围的精度为1%。

## 注意

调用频率必须由用户自己编程指定。

## 自动同步

可以将脉冲输出与更新输入变量INV(例如，CONT\_C)的块同步。这将确保输入变量的变化能尽快地以脉冲方式输出。

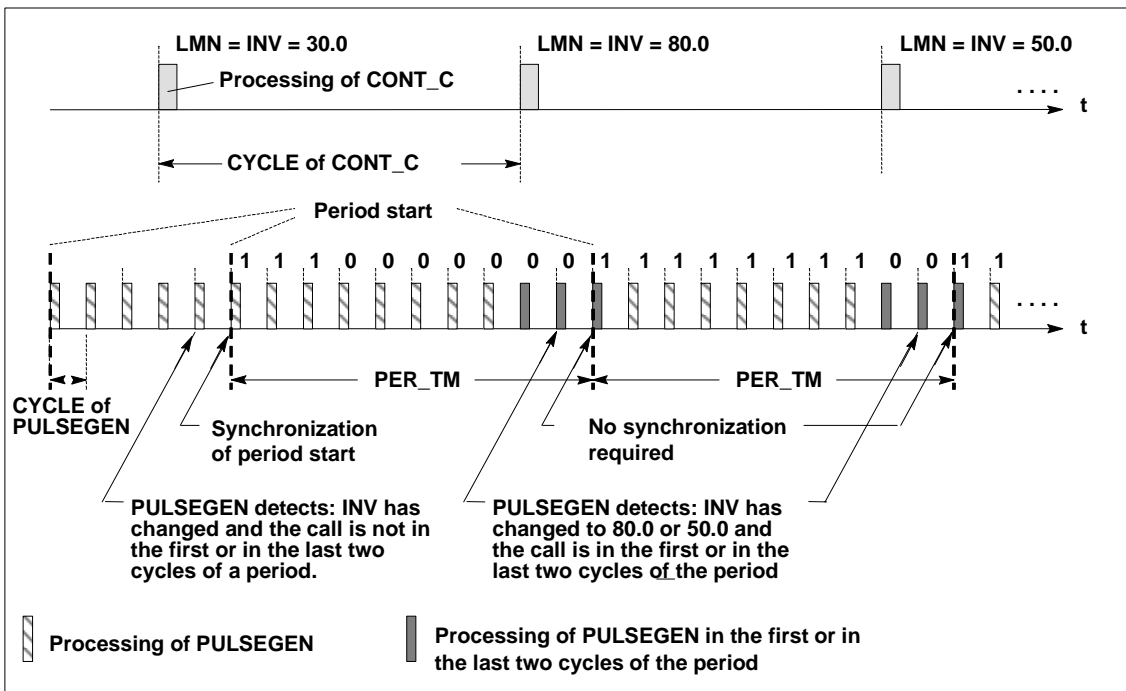
脉冲发生器按照周期PER\_TM的时间间隔计算输入值INV，并将此数值转换成相应时长的脉冲信号。

然后，由于计算INV的周期性中断等级通常较低，所有在INV更新之后，脉冲发生器应该尽快地启动离散值到脉冲信号的转换。

为此，程序块使用下列步骤同步周期的起始点：

如果INV发生变化，而块调用不在周期的第一个或最后两个调用循环中，则执行同步。将重新计算脉冲宽度，并在下一周期与新时期一起输出。





可以在SYN\_ON输入处禁止自动同步(= FALSE)。

### 注意

随着新的周期的开始，将在同步后于脉冲信号中近似精确地模拟INV的原值(即LMN值)。

### 模式

根据分配给脉冲发生器的参数，可以组态具有三步输出或双极或单极两步输出的PID控制器。下表给出了可能的模式所对应的开关组合的设置。

	开关		
模式	MAN_ON	STEP3_ON	ST2BI_ON
三步控制	FALSE	TRUE	ANY
具有双极控制范围的两步控制 (-100%至 +100%)	FALSE	FALSE	TRUE
具有单极控制范围的两步控制 (0 % ... 100 %)	FALSE	FALSE	FALSE
手动模式	TRUE	ANY	ANY

### 三步控制

在“三步控制”模式下，激励信号可以具有三种状态。将二进制输出信号QPOS\_P和QNEG\_P的值分配给执行机构的状态。  
下表给出了一个温度控制实例：

输出信号	执行器		
	加热	关闭	冷却
QPOS_P	TRUE	FALSE	FALSE
QNEG_P	FALSE	FALSE	TRUE

基于输入变量，使用特征曲线计算脉冲持续时间。特性曲线的形状由最小脉冲或最小中断时间及比率因子定义。

比率因子的正常值为1。

曲线中的“转折”由最小脉冲或最小中断时间引起。

#### 最小脉冲或最小断开时间

一个适当的最小脉冲或最小断开时间P\_B\_TM可以防止短暂开/关的次数，避免由此而降低切换元件和执行器的工作寿命。

---

#### 注意

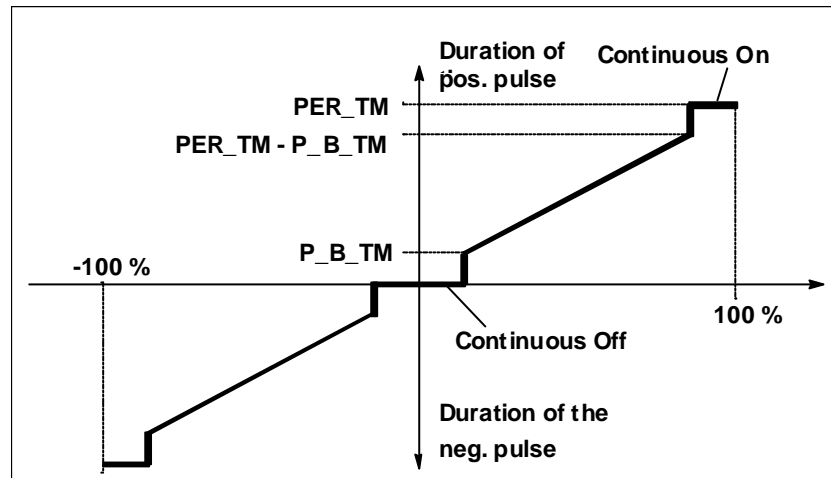
输入变量LMN处那些小绝对值会受到抑制，否则它们会生成较P\_B\_TM更短的脉冲宽度。将生成比(PER\_TM - P\_B\_TM)长的脉冲宽度的大输入值设置为100%或-100%。

---

通过将输入变量(%)与周期时间相乘获得正脉冲或负脉冲的宽度。

$$I\text{Period time} = \frac{\text{INV}}{100} * \text{PER\_TM}$$

下图显示的是三步控制器的对称曲线(比率因子 = 1)。



### 非对称三步控制

使用比率因子RATIOFAC可以更改正脉冲与负脉冲宽度的比率。例如，在一个热过程中，可以此为加热和冷却过程使用不同的系统时间常数。

比率因子还影响最小脉冲或最小断开时间。比率因子 < 1 意味着负脉冲的阈值与比率因子相乘。

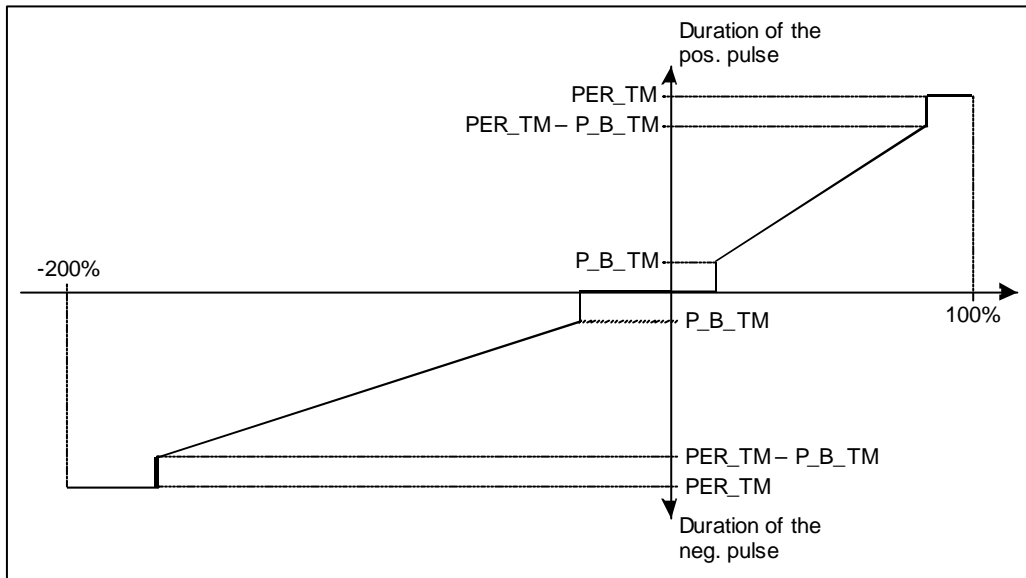
#### 比率因子 < 1

比率因子会缩短将输入变量与周期时间相乘所得到的负脉冲输出脉冲宽度。

$$\text{Dur. of positive pulse} = \frac{\text{INV}}{100} * \text{PER\_TM}$$

$$\text{Dur. of negative pulse} = \frac{\text{INV}}{100} * \text{PER\_TM} * \text{RATIOFAC}$$

下图显示三步控制器的非对称曲线(比率因子 = 0.5):



### 比率因子 > 1

计算输入变量和周期时间的乘积所得的正脉冲输出的脉宽，因比率因子的存在而减少。

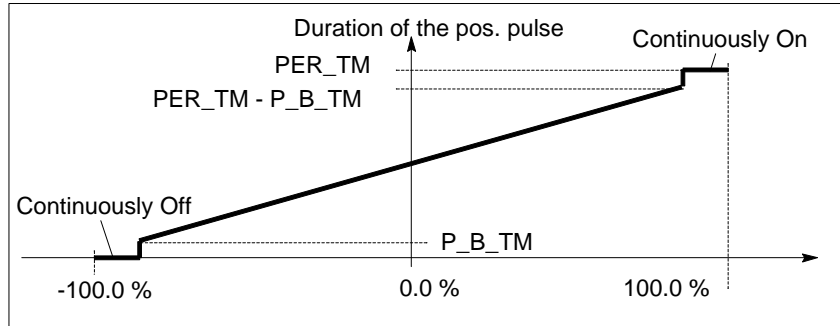
$$\text{Dur. of negative pulse} = \frac{\text{INV}}{100} * \text{PER\_TM}$$

$$\text{Dur. of positive pulse} = \frac{\text{INV}}{100} * \frac{\text{PER\_TM}}{\text{RATIOFAC}}$$

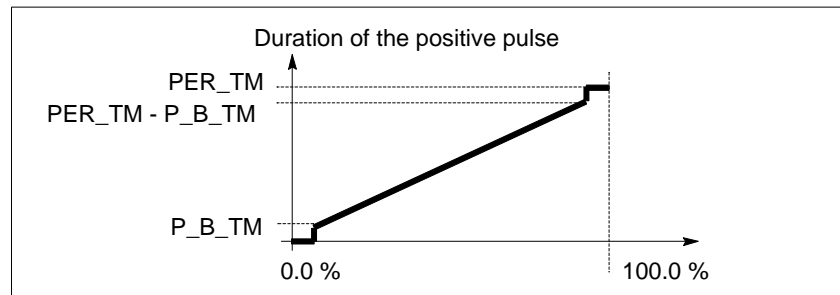
## 两步控制

在两步控制中，只会将PULSEGEN的正脉冲输出QPOS\_P连接到开/关执行机构。视所使用的操作值范围，两步控制器将具有双极或单极操作值范围。

### 具有双极调节变量范围的两步控制 (-100%至100%)



### 具有单级调节变量范围的两步控制 (0%至100%)



如果控制回路中的两步控制器的连接需要使用逻辑取反的二进制信号来激励脉冲，可在QNEG\_P获得取反输出信号。

脉冲	执行器	
	开	关
QPOS_P	TRUE	FALSE
QNEG_P	FALSE	TRUE

## 两步/三步控制的手动模式

在手动模式(MAN\_ON = TRUE)下，无论INV为何值，均可使用信号POS\_P\_ON和NEG\_P\_ON设置三步或两步控制器的二进制输出。

控制	POS_P_ON	NEG_P_ON	QPOS_P	QNEG_P
三步控制	FALSE	FALSE	FALSE	FALSE
	TRUE	FALSE	TRUE	FALSE
	FALSE	TRUE	FALSE	TRUE
	TRUE	TRUE	FALSE	FALSE
两步控制	FALSE	ANY	FALSE	TRUE
	TRUE	ANY	TRUE	FALSE

## 初始化

SFB/FB “PULSGEN”有一个在输入参数COM\_RST = TRUE时自动运行的初始化程序。

所有信号输出都设置为0。

## 出错信息

不使用出错输出参数RET\_VAL。

## 输入参数

参数	数据类型	取值范围:	缺省值	描述
INV	REAL	-100.0...100.0(%)	0.0	INPUT VARIABLE 将模拟操作值连接到输入参数“输入变量”。
PER_TM	TIME	>=20*CYCLE	T#1s	PERIOD TIME 使用“周期”输入参数输入脉宽调制的恒定周期。它对应于控制器的采样时间。脉冲发生器采样时间和控制器采样时间的比率决定脉宽调制的精度。
P_B_TM	TIME	>= CYCLE	T#0ms	MINIMUM PULSE/BREAK TIME 可以在输入参数“最小脉冲或最小中断时间”分配最小脉冲或最小中断时间。

参数	数据类型	取值范围:	缺省值	描述
RATIOFAC	REAL	0.1 ... 10.0	1.0	<b>RATIO FACTOR</b> 可以使用输入参数“比率因子”来改变负脉冲宽度与正脉冲宽度的比率。在热处理中,这使(例如)为加热和冷却补偿不同的时间常数(例如,在使用电加热和水冷却的工艺中)成为可能。
STEP3_ON	BOOL		TRUE	<b>THREE STEP CONTROL ON</b> “启用三步控制”输入参数可以激活此模式。在三步控制中,两个输出信号都是有效的。
ST2BI_ON	BOOL		FALSE	<b>TWO STEP CONTROL FOR BIPOLAR MANIPULATED VALUE RANGE ON</b> 使用输入参数“启用双极操作值范围的两步控制”可以在“双极操作值的两步控制”模式和“单极操作值范围的两步控制”模式间进行选择。必须设置参数STEP3_ON = FALSE。
MAN_ON	BOOL		FALSE	<b>MANUAL MODE ON</b> 通过设置输入参数“启用手动模式”可以手动设置输出信号。
POS_P_ON	BOOL		FALSE	<b>POSITIVE PULSE ON</b> 在采用三步控制的手动模式下,可以在输入参数“启用正脉冲”设置输出信号QPOS_P。在两步控制的手动模式中,QNEG_P和QPOS_P的设置必须始终相反。
NEG_P_ON	BOOL		FALSE	<b>NEGATIVE PULSE ON</b> 在采用三步控制的手动模式下,可以在输入参数“启用负脉冲”设置输出信号QNEG_P。在两步控制的手动模式中,QNEG_P和QPOS_P的设置必须始终相反。
SYN_ON	BOOL		TRUE	<b>SYNCHRONIZATION ON</b> 通过设置输入参数“启用同步”,可以自动与更新输入变量INV的块同步。这样可以确保变化的输入变量的输出速度尽可能达到脉冲速度。
COM_RST	BOOL		FALSE	<b>COMPLETE RESTART</b> 该块有一个在设置了COM_RST输入时自动执行的初始化程序
CYCLE	TIME	>= 1ms	T#10ms	<b>SAMPLING TIME</b> 块调用间的时间必须为常数。“采样时间”输入指定块调用间的时间。

---

**注意**

在此块中，输入参数的数值并没有限制。在这里并不检查参数。

---

**输出参数**

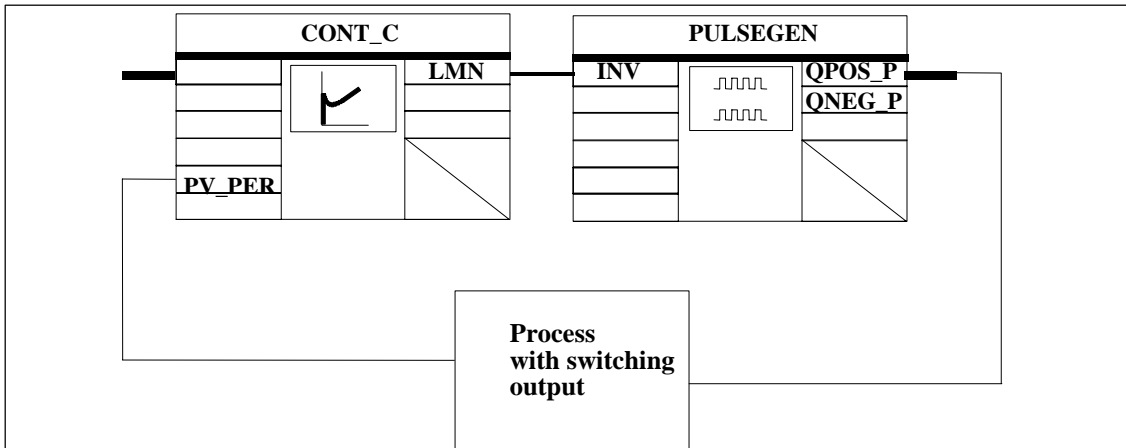
参数	数据类型	值	缺省值	描述
QPOS_P	BOOL		FALSE	OUTPUT POSITIVE PULSE 将在要输出脉冲时设置输出参数“输出正脉冲”。在三步控制中，始终输出正脉冲。在两步控制中，QNEG_P始终设置为与QPOS_P反向。
QNEG_P	BOOL		FALSE	OUTPUT NEGATIVE PULSE 将在要输出脉冲时设置输出参数“输出负脉冲”。在三步控制中，始终输出负脉冲。在两步控制中，QNEG_P始终设置为与QPOS_P反向。



## 25.4 PULSEGEN块的实例

### 控制回路

使用连续控制器CONT\_C和脉冲发生器PULSEGEN，用户可以实现具有比例执行机构切换输出的固定设定值控制器。下图显示控制回路的信号流。

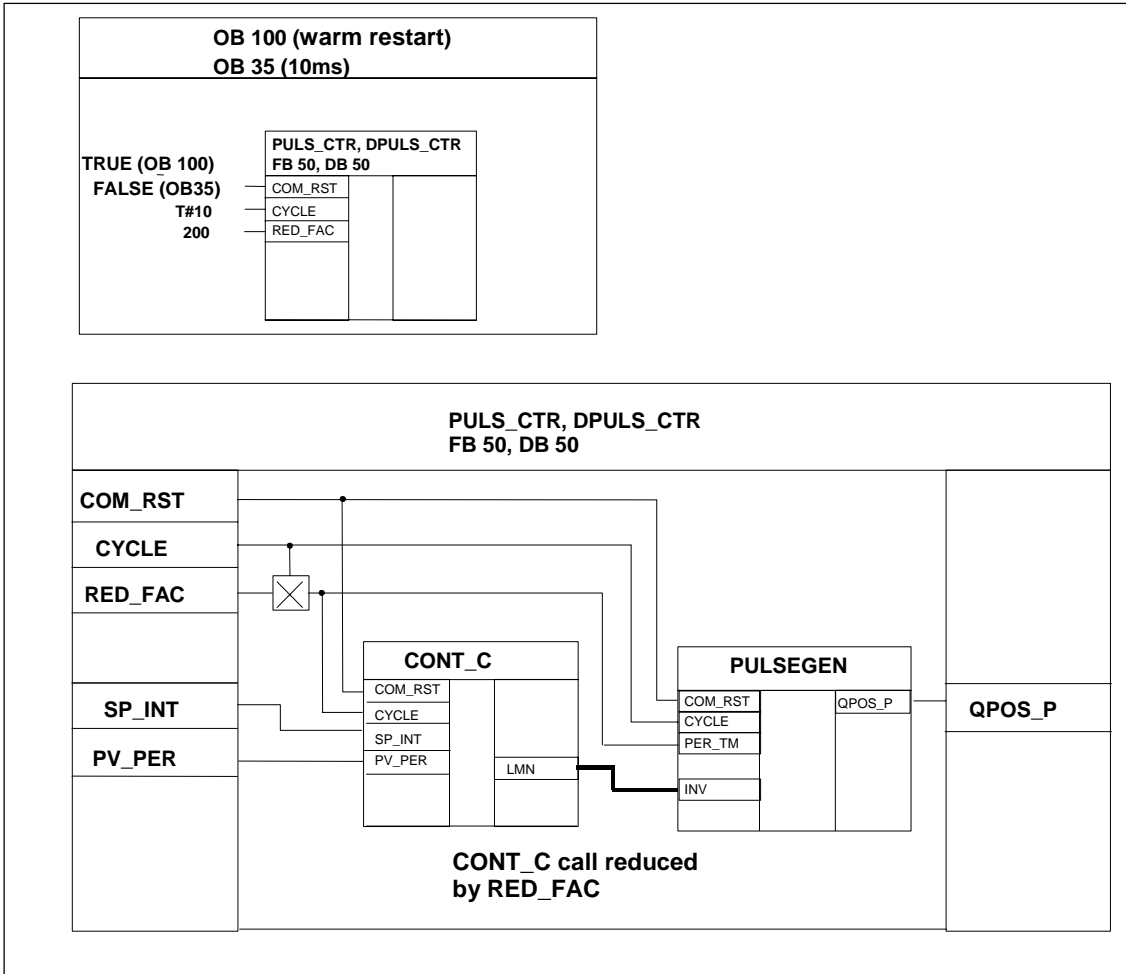


连续控制器CONT\_C形成操作值LMN，该操作值由脉冲发生器PULSEGEN转换为脉冲/中断信号QPOS\_P或QNEG\_P。

### 调用块并连接块

具有比例执行机构PULS\_CTRL切换输出的固定设定值控制器由块CONT\_C和PULSEGEN组成。执行块调用，使CONT\_C每2秒钟(=CYCLE\*RED\_FAC)调用一次，而PULSEGEN每10ms(=CYCLE)调用一次。OB35的循环周期设置成10ms。下图显示了互连情况。

在暖重启过程中，将在OB100中调用块PULS\_CTRL并将输入COM\_RST设置为TRUE。



## FB PULS\_CTR的STL程序

地址	声明	名称	类型	注释
0.0	in	SP_INT	REAL	设定值
4.0	in	PV_PER	WORD	过程变量外设
6.0	in	RED_FAC	INT	调用换算系数
8.0	in	COM_RST	BOOL	完全重启
10.0	in	循环	TIME	采样时间
14.0	out	QPOS_P	BOOL	驱动信号
16.0	stat	DI_CONT_C	FBCONT_C	计数器
142.0	stat	DI_PULSEGEN	FBPULSEGEN	计数器
176.0	stat	SCount	INT	计数器
0.0	TEMP	TCycCtr	TIME	控制器采样时间

STL	描述
<pre> A    #COM_RST JCN  M001 L    0 T    #sCount </pre>	//初始化程序
<pre> M001: L    #CYCLE       L    #RED_FAC       *D       T    #tCycCtr </pre>	//计算控制器采样时间
<pre> L    #sCount L    1 -I T    #sCount L    0 &lt;=I </pre>	//递减计数器并与零比较
<pre> JCN    M002 CALL   #DI_CONT_C COM_RST : =#COM_RST CYCLE :  =#tCycCtr SP_INT : =#SP_INT PV_PER : =#PV_PER L      #RED_FAC T      #sCount M002: L    #DI_CONT_C.LMN       T    #DI_PULSEGEN.INV       CALL #DI_PULSEGEN       PER_TM : =#tCycCtr       COM_RST : =#COM_RST       CYCLE :  =#CYCLE       QPOS_P : =#QPOS_P       BE </pre>	//条件块调用和设置计数器



## 26 用于紧凑型CPU的SFB

### 26.1 通过SFB44 “Analog”使用模拟量输出进行定位

#### 描述

要通过用户程序控制定位功能，请使用**SFB ANALOG (SFB44)**。

固定分配的模拟输出通过  $\pm 10\text{V}$  的电压(**电压信号**)或  $\pm 20\text{mA}$  的电流(**电流信号**)控制功率级。

- 加速阶段(**RAM\_UP**)过后，驱动器以速度(**VSetpoint**)接近目标。
- 到达由CPU计算出的**制动点**时，将开始减速(**RAMP\_DN**)，一直到转换点。
- 一旦到达**转换点**，将继续以慢速(**VCreep**)运行。
- 驱动器在到达**截止点**时关闭。
- 每个“Step Approach”的转换点和截止点由已指定的**转换差值**和**截止差值**的参数值确定。可以用不同方式确定前进运动(正方向)和反向运动(负方向)的转换差值和截止差值。
- 到达截止点时，运行即告结束(**WORKING = FALSE**)。随后即可开始新的运行。
- 当实际位置值到达**目标范围**时，即达到指定的目标(**POS\_RCD = TRUE**)。如果实际位置值逐渐偏移，而又未启动新的运行，“到达位置”信号将不会再次复位。

当转换差值小于截止差值时，驱动器将从制动点减速到速度设定值0。

#### 基本参数:

在此描述适用于所有操作模式的SFB参数。各操作模式的专用参数将随该操作模式一并描述。

参数:

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
<b>LADDR</b>	INPUT	WORD	0	CPU专用	W#16#0310	子模块的I/O地址，用户在“HW配置”中指定。 如果E和A地址不相等，则必须指定二者中较低的一个。
<b>CHANNEL</b>	INPUT	INT	2	0	0	通道号
<b>STOP</b>	INPUT	BOOL	4.4	TRUE/FALSE	FALSE	停止运行使得 <b>STOP = TRUE</b> ，则可以提前停止/中断运行。
<b>ERR_A</b>	INPUT	BOOL	4.5	TRUE/FALSE	FALSE	收集对外部出错的确认 经 <b>ERR_A = TRUE</b> 确认后，外部出错即被清除。
<b>SPEED</b>	INPUT	DINT	12	慢速最高为 1,000,000 脉冲/s 不高于在参数中声明的最大速度	1000	轴加速到速度为“Setpoint”。 运行中无法改变速度。
<b>WORKING</b>	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
<b>ACT_POS</b>	OUTPUT	DINT	18	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	实际位置值
<b>MODE_OUT</b>	OUTPUT	INT	22	0、1、3、4、5	0	激活的/已组态的操作模式

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
<b>ERR</b>	OUTPUT	WORD	24	每位 “0”或“1”:	0	外部出错: 位2: 零点监视 第11位: 行程监视 (始终为1) 位12: 操作范围监视 第13位: 实际值监视 第14位: 目标归位 监视 位15: 目标范围监视 剩余的其它位保留 备用
<b>ST_ENBLD</b>	OUTPUT	BOOL	26.0	TRUE/FALSE	TRUE	如果满足下列所有 条件, CPU会设置 启动使能: <ul style="list-style-type: none"> <li>• 没有未决STOP (STOP = FALSE)</li> <li>• 没有未决外部出 错(ERR = 0)</li> <li>• 已设置驱动器使 能(DRV_EN = TRUE)</li> <li>• 没有活动的定位 运行(WORKING = FALSE)</li> </ul>
<b>ERROR</b>	OUTPUT	BOOL	26.1	TRUE/FALSE	FALSE	开始/恢复运行时出错
<b>STATUS</b>	OUTPUT	WORD	28.0	W#16#0000至 W#16#FFFF	W#16#0000	出错编号

未分配给块的参数(本地统计数据):

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
ACCEL	STATIC	DINT	30	1到100,000 脉冲/s <sup>2</sup>	100	加速度 运行中无法更改。
DECEL	STATIC	DINT	34	1到100,000 脉冲/s <sup>2</sup>	100	减速度 运行中无法更改。
CHGDIFF_P	STATIC	DINT	38	0到 +10 <sup>8</sup> 个脉冲	1000	正转换差值: 正“转换差值”定义驱动器以慢速继续向前运行的转换点。
CUTOFF-DIFF_P	STATIC	DINT	42	0到 +10 <sup>8</sup> 个脉冲	100	正截止差值: “正截止差值”定义关闭驱动器慢速向前运行时的截止点。
CHGDIFF_M	STATIC	DINT	46	0到 +10 <sup>8</sup> 个脉冲	1000	负转换差值: “负转换差值”定义驱动器以慢速反向继续运行的转换点。
CUTOFF-DIFF_P	STATIC	DINT	50	0到 +10 <sup>8</sup> 个脉冲	100	负截止差值: “负截止差值”定义关闭驱动器慢速反向运行时的截止点。
PARA	STATIC	BOOL	54.0	TRUE/FALSE	FALSE	参数已分配给轴



参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
<b>DIR</b>	STATIC	BOOL	54.1	TRUE/FALSE	FALSE	实际的/上一方向 FALSE = 前进 (正方向) TRUE = 反向 (负方向)
<b>CUTOFF</b>	STATIC	BOOL	54.2	TRUE/FALSE	FALSE	驱动器在截止范围内 (从截止点到下一运行的 起点)
<b>CHGOVER</b>	STATIC	BOOL	54.3	TRUE/FALSE	FALSE	驱动器在转换范围内 (到达慢速和下一运行 起点之间)
<b>RAMP_DN</b>	STATIC	BOOL	54.4	TRUE/FALSE	FALSE	驱动器减速(从制动点 到转换点)
<b>RAMP_UP</b>	STATIC	BOOL	54.5	TRUE/FALSE	FALSE	驱动器加速(从起点到 达到速度SPEED (VSetpoint))
<b>DIST_TO_ GO</b>	STATIC	DINT	56	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	实际行进距离
<b>LAST_TRG</b>	STATIC	DINT	60	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	上一个/当前目标 <ul style="list-style-type: none"> <li>绝对法: 运行启动时 LAST_TRG = 当前绝对目标( TARGET)。</li> <li>相对法: 运行启动时 LAST_TRG = LAST_TRG为前 一运行指定的 +/- 距离 (TARGET)。</li> </ul>

## “点动”模式的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动器使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	正向点动(上升沿)
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	负向点动(上升沿)
MODE_IN	INPUT	INT	6	0、1、3、4、5	1	操作模式, 1 = 点动
WORKING	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
ACT_POS	OUTPUT	DINT	18	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 个脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	22	0、1、3、4、5	0	激活的/已组态的操作模式

## “参考运行”模式的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动器使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	在正方向参考运行 (上升沿)
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	在负方向参考运行 (上升沿)
MODE_IN	INPUT	INT	6	0、1、3、4、5	1	操作模式, 3 = “参考运行”
WORKING	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
SYNC	OUTPUT	BOOL	16.3	TRUE/FALSE	FALSE	SYNC = TRUE: 轴已同步
ACT_POS	OUTPUT	DINT	18	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 个脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	22	0、1、3、4、5	0	激活的/已组态的操作模式

## “相对法”模式的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动器使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	向正方向运行(上升沿)
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	向负方向运行(上升沿)
MODE_IN	INPUT	INT	6	0、1、3、4、5	1	操作模式, 4 = 相对法
TARGET	INPUT	DINT	8	0到 $10^9$ 个脉冲	1000	以脉冲计算的距离 (只允许正值)
WORKING	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
POS_RCD	OUTPUT	BOOL	16.1	TRUE/FALSE	FALSE	到达位置
ACT_POS	OUTPUT	DINT	18	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	22	0、1、3、4、5	0	激活的/已组态的操作 模式

## “绝对法”的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动器使能
START	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	开始运行(上升沿)
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	向正方向运行 (上升沿)
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	向负方向运行 (上升沿)
MODE_IN	INPUT	INT	6	0、1、3、4、5	1	操作模式, 5 = 绝对法
TARGET	INPUT	DINT	8	线性轴 $-5 \times 10^8$ 到 $+5 \times 10^8$ 旋转轴: 0到旋转轴终点 -1	1000	以脉冲计算的目标
WORKING	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
POS_RCD	OUTPUT	BOOL	16.1	TRUE/FALSE	FALSE	到达位置
ACT_POS	OUTPUT	DINT	18	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	22	0、1、3、4、5	0	激活的/已组态的操作 模式

“设置参考点”作业的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
SYNC	OUTPUT	BOOL	16.3	TRUE/FALSE	FALSE	轴已同步

未分配给块的参数(本地统计数据):

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
JOB_REQ	STATIC	BOOL	76.0	TRUE/FALSE	FALSE	作业初始化 (上升沿)
JOB_DONE	STATIC	BOOL	76.1	TRUE/FALSE	TRUE	可以启动新作业
JOB_ERR	STATIC	BOOL	76.2	TRUE/FALSE	FALSE	故障作业
JOB_ID	STATIC	INT	78	1、2	0	作业, 1 = “设置参考点”
JOB_STAT	STATIC	WORD	80	W#16#0000至 W#16#FFFF	W#16#0000	作业出错编号
JOB_VAL	STATIC	DINT	82	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 个脉冲	0	参考点坐标的作业 参数

## “清除剩余距离”作业的参数

未分配给块的参数(本地统计数据):

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
JOB_REQ	STATIC	BOOL	76.0	TRUE/FALSE	FALSE	作业初始化 (上升沿)
JOB_DONE	STATIC	BOOL	76.1	TRUE/FALSE	TRUE	可以启动新作业
JOB_ERR	STATIC	BOOL	76.2	TRUE/FALSE	FALSE	故障作业
JOB_ID	STATIC	INT	78	1、2	0	作业, 2 = “清除剩余距离”
JOB_STAT	STATIC	WORD	80	W#16#0000至 W#16#FFFF	W#16#0000	作业出错编号
JOB_VAL	STATIC	DINT	82	-	0	任意设置

## “长度测量”操作的参数

此操作开始于数字输入的上升沿。没有特定的输入参数。

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
MSR_DONE	OUTPUT	BOOL	16.2	TRUE/FALSE	FALSE	长度测量完成

未分配给块的参数(本地统计数据):

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
BEG_VAL	STATIC	DINT	64	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	开始长度测量的 实际位置值
END_VAL	STATIC	DINT	68	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	完成长度测量的 实际位置值
LEN_VAL	STATIC	DINT	72	0到 $10^9$ 个脉冲	0	测量的长度

## 出错信息

## 操作模式出错(ERROR = TRUE)

如果检测到出错，输出参数ERROR将置为TRUE。**STATUS**参数给出了发生出错的原因。

事件类别 出错代码	说明
W#16#2002	SFB出错，请使用SFB44
W#16#2004	通道号(CHANNEL)出错。将通道设置为“0”
W#16#3001	由于调用同一SFB时发生作业出错而拒绝运行作业。更正各自的JOB参数
W#16#3002	驱动器工作时不允许更改MODE_IN。请等待当前定位运行结束。
W#16#3003	未知操作模式(MODE_IN)。允许模式为1(点动)、3(参考运行)、4(相对法)和5(绝对法)。
W#16#3004	同一时间只允许一个启动请求。有效的启动请求为DIR_P、DIR_M或START。
W#16#3005	只有“绝对法”操作模式允许START。使用DIR_P或DIR_M启动运行
W#16#3006	线性轴和“绝对法”操作模式不允许使用DIR_P或DIR_M。通过START启动运行
W#16#3007	轴未同步。只有同步轴时才能使用“绝对法”。
W#16#3008	清除工作范围。只有在点动模式下才允许将运行返回工作位置。
W#16#3101	由于没有为轴指定参数，因此禁止启动通过HWConfig设置“定位”子模块的参数
W#16#3102	由于未使能驱动器，因此禁止启动。在SFB上设置“使能驱动器”(DRV_EN=TRUE)
W#16#3103	由于设置了STOP，因此禁止启动。在SFB中清除STOP(STOP=FALSE)
W#16#3104	由于驱动器当前正在执行定位运行(WORKING=TRUE)，禁止启动。请等待当前定位运行结束。
W#16#3105	由于至少有一个未决出错尚未清除，因此禁止启动。首先，排除并清除所有外部出错，然后恢复运行。
W#16#3202	SPEED中的速度设定值出错。速度设定值超出最高为1000000脉冲/s的慢速允许范围，尽管未超出参数指定的最大速度。
W#16#3203	ACCEL的加速设定值超出1到100,000脉冲/s <sup>2</sup> 的范围。
W#16#3204	DECEL的减速设定值超出1到100,000脉冲/s <sup>2</sup> 的范围。
W#16#3206	SPEED的速度设定值必须高于/等于参数化的参考频率。
W#16#3301	转换/截止差值过高。将最大转换/截止差值设置为10 <sup>8</sup>
W#16#3304	截止差值过低。截止差值大小至少必须为目标范围的一半。
W#16#3305	转换差值过低。转换差值大小至少必须为目标范围的一半。
W#16#3401	目标设置超出工作范围。对于线性轴和Step Approach，目标设置必须在软件限位开关范围内(含此限值)。
W#16#3402	目标设置出错。对于旋转轴，目标设置必须大于0并小于旋转轴终点值。
W#16#3403	距离设置出错。“相对法”的行进距离设定值必须为正值。
W#16#3404	距离设置出错。所得结果(绝对目标坐标)必须大于-5x10 <sup>8</sup> 。

事件类别 出错代码	说明
W#16#3405	距离设置出错。所得结果(绝对目标坐标)必须小于 $5 \times 10^8$ 。
W#16#3406	距离设置出错。所得结果(绝对目标坐标)必须在工作范围(目标范围的 +/- 一半)内
W#16#3501	行进距离过长。目标坐标 + 实际剩余距离必须大于/等于 $-5 \times 10^8$
W#16#3502	行进距离过长。目标坐标 + 实际剩余距离必须小于/等于 $5 \times 10^8$
W#16#3503	行进距离过短。正向的行进距离必须大于指定的正向截止差值
W#16#3504	行进距离过短。负向的行进距离必须大于指定的负向截止差值
W#16#3505	行进距离过短，或正向的限位开关已经启动。正向最后可接近的目标(工作范围或行进距离限制)与实际位置过近
W#16#3506	行进距离过短，或负方向的限位开关已经启动。负向最后可接近的目标(工作范围或行进距离限制)与实际位置过近

### 作业出错(JOB\_ERR = TRUE)

如果检测到出错，输出参数JOB\_ERROR将置为TRUE。参数JOB\_ERROR给出了出错原因。

事件类别 出错信息	说明
W#16#4001	未设置轴参数。通过HWConfig设置“定位”子模块的参数
W#16#4002	由于仍在定位，无法执行作业。等到WORKING = FALSE，再重复执行作业。
W#16#4004	未知作业。检查作业标识号，然后重复执行作业。
W#16#4101	对于线性轴，参考点坐标不能超出工作范围的限制。
W#16#4102	对于线性轴，设置的参考点坐标 + 实际剩余距离必须大于/等于 $-5 \times 10^8$ 。
W#16#4103	对于线性轴，设置的参考点坐标 + 实际剩余距离必须小于/等于 $5 \times 10^8$ 。
W#16#4104	对于线性轴，设置的参考点坐标 + 距起始点的实际剩余差值必须大于/等于 $-5 \times 10^8$ 。
W#16#4105	对于线性轴，设置的参考点坐标 + 距起始点的实际剩余差值必须小于/等于 $-5 \times 10^8$ 。
W#16#4106	对于旋转轴，参考点坐标不能小于0，也不能大于/等于旋转轴终点。

## 外部出错(ERR)

技术电路会监视运行、行进距离和连接的外围设备。首先必须已在“驱动器”、“轴”和“编码器”参数画面窗体中打开监视功能。

触发监视单元时将报告外部故障。外部出错的发生与已启动的操作无关。必须使用ERR\_A = TRUE清除外部出错。

SFB参数ERR (WORD)的置位对应外部出错。

监视	出错代码:	ERR-WORD中的位
零脉冲(零标记)	W#16#0004	2
行进距离	W#16#0800	11
工作范围	W#16#1000	12
实际值	W#16#2000	13
目标位置	W#16#4000	14
目标范围	W#16#8000	15

## 系统出错

系统出错由BIE = FALSE指示。系统出错由读取/写入背景数据块引起或由多次调用SFB引起。



## 26.2 通过SFB46 “DIGITAL”使用数字量输出进行定位

### 描述

通过用户程序使用**SFB DIGITAL (SFB46)**控制定位功能。

已为驱动器固定地分配了四个24V数字输出。它们用于控制功率级。数字输出根据控制模式组态控制方向和速度级(快速/慢速)。

通过两个相位偏移为90度的非对称24V增量传感器测量距离。

- 首先，以(**V<sub>Rapid</sub>**)速度接近目标。
- 到达**转换点**时速度切换为慢速(**V<sub>Creep</sub>**)。
- 驱动器在到达**关闭点**时关闭。
- 每个Step Approach的切换点和关闭点由为**转换差值**和**截止差值**声明的参数值确定。前进运动(正方向)和反向运动(负方向)的转换差值和截止差值可以用不同方式确定。
- 到达截止点时，运行即告结束(**WORKING = FALSE**)。随后即可开始新的运行。
- 当实际位置值到达**归位目标**时，即达到指定的目标(**POS\_RCD = TRUE**)。如果实际位置值逐渐偏移，而未启动新的运行，“到达位置”信号将不会再次复位。

### 基本参数:

在此将说明**SFB**参数。这些参数适用于所有操作模式。各操作模式的专用参数将随各操作模式一并描述。

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
<b>LADDR</b>	INPUT	WORD	0	CPU专用	W#16#0310	子模块的I/O地址，由用户在“HW配置”中指定。 如果E和A地址不相等，则必须指定二者中较低的一个。
<b>CHANNEL</b>	INPUT	INT	2	0	0	通道号:
<b>STOP</b>	INPUT	BOOL	4.4	TRUE/FALSE	FALSE	停止运行 使得 <b>STOP = TRUE</b> ，可以提前停止/中断运行。

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
ERR_A	INPUT	BOOL	4.5	TRUE/FALSE	FALSE	收集对外部出错的确认 经ERR_A = TRUE 确认后, 外部出错即被 清除。
SPEED	INPUT	BOOL	12.0	TRUE/FALSE	FALSE	用于“快速/慢速”模式的 两个速度级 TRUE = 快速模式 FALSE = 慢速模式
WORKING	OUTPUT	BOOL	14.0	TRUE/FALSE	FALSE	正在运行
ACT_POS	OUTPUT	DINT	16	-5x10 <sup>8</sup> 到 5x10 <sup>8</sup> 个脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0、1、3、4、 5	0	激活的/已组态的操作 模式
ERR	OUTPUT	WORD	22	每位 “0”或“1”:	0	外部出错: 第2位: 零标记监视 第11位: 行程监视 (始终为1) 第12位: 操作范围监视 第12位: 实际值监视 第12位: 目标位置监视 第15位: 目标位置 监视。 剩余的其它位保留备用
ST_ENBLD	OUTPUT	BOOL	24.0	TRUE/FALSE	TRUE	如果满足下列所有 条件, 则使能CPU: <ul style="list-style-type: none"> <li>• 没有未决STOP (STOP = FALSE)</li> <li>• 没有未决外部出错 (ERR = 0)</li> <li>• 已设置驱动器使能 (DRV_EN = TRUE)</li> <li>• 没有活动的定位运 行(WORKING = FALSE)</li> </ul>
ERROR	OUTPUT	BOOL	24.1	TRUE/FALSE	FALSE	开始/恢复运行时出错
STATUS	OUTPUT	WORD	26.0	W#16#0000至 W#16#FFFF	W#16#0000	出错编号

## 未分配给块的参数(本地统计数据):

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
CHGDIFF_P	STATIC	DINT	28	0到 $+10^8$ 个脉冲	1000	正切换差值: “正转换差值”定义驱动器以慢速继续向前运行的点。
CUTOFF-DIFF_P	STATIC	DINT	32	0到 $+10^8$ 个脉冲	100	正截止差值: “正截止差值”定义关闭驱动器慢速向前运行时的截止点。
CHGDIFF_M	STATIC	DINT	36	0到 $+10^8$ 个脉冲	1000	负转换差值: “负转换差值”定义驱动器以慢速继续向前运行的点。
CUTOFF-DIFF_P	STATIC	DINT	40	0到 $+10^8$ 个脉冲	100	负截止差值: “正截止差值”定义关闭驱动器慢速反向运行时的截止点。
PARA	STATIC	BOOL	44.0	TRUE/FALSE	FALSE	参数已分配给轴
DIR	STATIC	BOOL	44.1	TRUE/FALSE	FALSE	实际的/上一方向 FALSE = 前进(正方向) TRUE = 反向(负方向)
CUTOFF	STATIC	BOOL	44.2	TRUE/FALSE	FALSE	驱动器在截止范围内 (从截止位置到下一运行的起点)

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
CHGOVER	STATIC	BOOL	44.3	TRUE/FALSE	FALSE	驱动器在转换范围内 (从到达慢速的点到下一运行的起点)
DIST_TO_GO	STATIC	DINT	46	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	实际剩余距离
LAST_TRG	STATIC	DINT	50	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	上一个/当前目标 <ul style="list-style-type: none"> <li>绝对法: 运行启动时 LST_TRG = 当前绝对目标 (TARGET)。</li> <li>相对法: 运行启动时 LST_TRG = LAST_TRG 为前一运行的指定 +/- 距离 (TARGET)。</li> </ul>

#### “点动”模式的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动器使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	正向点动(上升沿)
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	负向点动(上升沿)
MODE_IN	INPUT	INT	6	0、1、3、4、5	1	操作模式, 1 = 点动
WORKING	OUTPUT	BOOL	14.0	TRUE/FALSE	FALSE	正在运行
ACT_POS	OUTPUT	DINT	16	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0、1、3、4、5	0	激活的/已组态的操作 模式

## “参考运行”模式的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动器使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	在正方向参考运行 (上升沿)
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	在负方向参考运行 (上升沿)
MODE_IN	INPUT	INT	6	0、1、3、4、5	1	操作模式，3 =“参考运行”
WORKING	OUTPUT	BOOL	14.0	TRUE/FALSE	FALSE	正在运行
SYNC	OUTPUT	BOOL	14.3	TRUE/FALSE	FALSE	SYNC = TRUE: 轴已同步
ACT_POS	OUTPUT	DINT	16	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 个脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0、1、3、4、5	0	激活的/已组态的操作 模式

## “相对法”模式的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动器使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	向正方向运行(上升沿)
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	向负方向运行(上升沿)
MODE_IN	INPUT	INT	6	0、1、3、4、 5	1	操作模式， 4 = 相对法
TARGET	INPUT	DINT	8	0到10 <sup>9</sup> 个脉冲	1,000	以脉冲计算的距离 (只允许正值)
WORKING	OUTPUT	BOOL	14.0	TRUE/FALSE	FALSE	正在运行
POS_RCD	OUTPUT	BOOL	14.1	TRUE/FALSE	FALSE	到达位置
ACT_POS	OUTPUT	DINT	16	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 个脉 冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0、1、3、4、 5	0	激活的/已组态的操作 模式

“绝对法”的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动器使能
START	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	开始运行(上升沿)
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	向正方向运行 (上升沿)
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	向负方向运行 (上升沿)
MODE_IN	INPUT	INT	6	0、1、3、4、5	1	操作模式, 5 = 绝对法
TARGET	INPUT	DINT	8	线性轴 -5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 旋转轴: 0到旋转轴终点 -1	1,000	以脉冲计算的目标
WORKING	OUTPUT	BOOL	14.0	TRUE/FALSE	FALSE	正在运行
POS_RCD	OUTPUT	BOOL	14.1	TRUE/FALSE	FALSE	到达位置
ACT_POS	OUTPUT	DINT	16	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 个脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0、1、3、4、5	0	激活的/已组态的操作 模式

“设置参考点”作业的参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
SYNC	OUTPUT	BOOL	14.3	TRUE/FALSE	FALSE	轴已同步

## 未分配给块的参数(本地统计数据):

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
JOB_REQ	STATIC	BOOL	66.0	TRUE/FALSE	FALSE	作业初始化 (上升沿)
JOB_DONE	STATIC	BOOL	66.1	TRUE/FALSE	TRUE	可以启动新作业
JOB_ERR	STATIC	BOOL	66.2	TRUE/FALSE	FALSE	故障作业
JOB_ID	STATIC	INT	68	1、2	0	作业, 1 = “设置参考点”
JOB_STAT	STATIC	WORD	70	W#16#0000到 W#16#FFFF	W#16#0000	作业出错编号
JOB_VAL	STATIC	DINT	72	$-5 \times 10^8$ 到 $+5 \times 10^8$ 个脉冲	0	参考点坐标的作业 参数

## “清除剩余距离”作业的参数

## 未分配给块的参数(本地统计数据):

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
JOB_REQ	STATIC	BOOL	66.0	TRUE/FALSE	FALSE	作业初始化(上升沿)
JOB_DONE	STATIC	BOOL	66.1	TRUE/FALSE	TRUE	可以启动新作业
JOB_ERR	STATIC	BOOL	66.2	TRUE/FALSE	FALSE	故障作业
JOB_ID	STATIC	IINT	68	1、2	0	作业, 2 = “清除剩余距离”
JOB_STAT	STATIC	WORD	70	0到FFFF十六 进制	0	作业出错编号
JOB_VAL	STATIC	DINT	72	-	0	无。

## “长度测量”功能的参数

此操作开始于数字输入的上升沿。没有特定的输入参数。

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
MSR_DONE	OUTPUT	BOOL	14.2	TRUE/FALSE	FALSE	长度测量完成

### 未分配给块的参数(本地统计数据):

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
BEG_VAL	STATIC	DINT	54	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 个脉冲	0	开始长度测量的实位置值
END_VAL	STATIC	DINT	58	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 个脉冲	0	完成长度测量的实际位置值
LEN_VAL	STATIC	DINT	62	0到10 <sup>9</sup> 个脉冲	0	测量的长度

## 出错信息

### 操作模式出错(ERROR = TRUE)

如果检测到出错，输出参数ERROR将设置为TRUE。**STATUS**参数显示发生出错的原因。

事件类别 出错代码	说明
W#16#2001	SFB出错，请使用SFB46
W#16#2004	通道号(CHANNEL)出错。将通道设置为“0”
W#16#3001	由于调用同一SFB时发生作业出错而拒绝运行作业。更正各自的JOB参数
W#16#3002	驱动器工作时不允许更改MODE_IN。请等待当前定位运行结束。
W#16#3003	未知操作模式(MODE_IN)。允许模式为1(点动)、3(参考运行)、4(相对法)和5(绝对法)。
W#16#3004	同一时间只允许一个启动请求。有效的启动请求为DIR_P、DIR_M或START。
W#16#3005	只有“绝对法”操作模式允许START。使用DIR_P或DIR_M启动运行
W#16#3006	线性轴和“绝对法”操作模式不允许DIR_P或DIR_M。使用START启动运行



事件类别 出错代码	说明
W#16#3007	轴未同步。只有同步轴时才能使用“绝对法”。
W#16#3008	清除工作范围。只有在点动模式下才允许将运行返回工作位置。
W#16#3101	由于未设置轴参数，因此禁止启动。通过HWConfig设置“定位”子模块的参数
W#16#3102	由于未使能驱动器，因此禁止启动。在SFB上设置“驱动器使能”(DRV_EN=TRUE)
W#16#3103	由于设置了STOP，因此禁止启动。在SFB中清除STOP(STOP=FALSE)
W#16#3104	由于驱动器当前正在执行定位运行(WORKING=TRUE)，因此禁止启动。请等待当前定位运行结束。
W#16#3105	由于至少有一个未决出错尚未清除，因此禁止启动。首先，排除并清除所有外部出错，然后重新启动运行。
W#16#3201	SPEED的速度设定值出错。使用数字量输出进行定位时，只能使用“慢速”(0)和“快速”(1)。
W#16#3301	转换/截止差值过高。将最大转换/截止差值设置为 $10^8$
W#16#3303	转换差值过低。转换差值必须高于/等于截止差值。
W#16#3304	截止差值过低。截止差值大小必须至少为目标范围的一半。
W#16#3401	目标设置超出工作范围。对于线性轴和Step Approach，目标设置必须在软件限位开关范围内(含此限值)。
W#16#3402	目标设置出错。对于旋转轴，目标设置必须大于0并小于旋转轴终点值。
W#16#3403	距离设置出错。“相对法”的行进距离设定值必须为正值。
W#16#3404	距离设置出错。所得结果(绝对目标坐标)必须大于 $-5 \times 10^8$ 。
W#16#3405	距离设置出错。所得结果(绝对目标坐标)必须小于 $5 \times 10^8$ 。
W#16#3406	距离设置出错。所得结果(绝对目标坐标)必须在工作范围(目标范围的 +/- 一半)内
W#16#3501	行进距离过长。目标坐标 + 实际剩余距离必须大于/等于 $-5 \times 10^8$
W#16#3502	行进距离过长。目标坐标 + 实际剩余距离必须小于/等于 $5 \times 10^8$
W#16#3503	行进距离过短。正方向的行进距离必须大于指定的正向截止差值
W#16#3504	行进距离过短。负方向的行进距离必须大于指定的负向截止差值
W#16#3505	行进距离过短，或正方向的限位开关已经启动。正向最后可接近的目标(工作范围或行进距离限制)与实际位置过近。
W#16#3506	行进距离过短，或负方向的限位开关已经启动。负方向最后可接近的目标(工作范围或行进距离限制)与实际位置过近。

### 作业出错(JOB\_ERR = TRUE)

如果检测到出错，输出参数JOB\_ERROR将置为TRUE。参数JOB\_ERROR给出了出错原因。

事件类别 出错代码	说明
W#16#4001	未设置轴参数。通过HWConfig设置“定位”子模块的参数
W#16#4002	由于仍在定位，无法执行作业。只有定位未运行时才能执行作业。 等到WORKING = FALSE，再重复执行作业。
W#16#4004	未知作业。检查作业标识号，然后重复执行作业。
W#16#4101	对于线性轴，参考点坐标不能超出工作范围的限制。
W#16#4102	对于线性轴，指定的参考点坐标 + 实际剩余距离必须大于/等于 $-5 \times 10^8$ 。
W#16#4103	对于线性轴，指定的参考点坐标 + 实际剩余距离必须小于/等于 $5 \times 10^8$ 。
W#16#4104	对于线性轴，指定的参考点坐标 + 距起始点的实际剩余差值必须大于/等于 $-5 \times 10^8$ 。
W#16#4105	对于线性轴，指定的参考点坐标 + 距起始点的实际剩余差值必须小于/等于 $-5 \times 10^8$ 。
W#16#4106	对于旋转轴，参考点坐标不能小于0，也不能大于/等于旋转轴终点。

### 外部出错(ERR)

技术电路会监视运行、行进距离和连接的外围设备。首先必须已在“驱动器”、“轴”和“编码器”参数画面窗体中打开监视功能。

触发监视单元时将报告外部故障。外部出错的发生与已启动的操作无关。必须使用ERR\_A = TRUE清除外部出错。

外部出错的SFB参数ERR(WORD)的置位。

监视	出错代码:	ERR-WORD中的位
零脉冲(零标记)	W#16#0004	2
行进距离	W#16#0800	11
工作范围	W#16#1000	12
实际值	W#16#2000	13
进入目标	W#16#4000	14
目标范围	W#16#8000	15

### 系统出错

系统出错由BIE = FALSE指示。系统出错由读取/写入背景数据块引用或多次调用SFB引起。

## 26.3 使用SFB47 “COUNT”控制计数器

### 描述

要通过用户程序控制定位功能，请使用**SFB COUNT (SFB47)**。

可以执行下列操作：

- 通过软件门**SW\_GATE**启动/停止计数器
- 使能/控制输出**DO**
- 检索状态位**STS\_CMP**、**STS\_OFLW**、**STS\_UFLW**和**STS\_ZP**
- 检索实际的计数器值**COUNTVAL**
- 读取/写入内部计数器寄存器的作业

### 参数

参数	声明	数据类型	地址 (背景数据块)	取值范围：	缺省值	描述
<b>LADDR</b>	INPUT	WORD	0	CPU专用	W#16#0300	子模块的I/O地址，由用户在“HW配置”中指定。 如果E和A地址不相等，则必须指定二者中较低的一个。
<b>CHANNEL</b>	INPUT	INT	2	CPU 312C: 0到1 CPU 313C: 0到2 CPU 314C: 0至3	0	通道号：
<b>SW_GATE</b>	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	用于启动/停止计数器的软件门
<b>CTRL_DO</b>	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	输出使能
<b>SET_DO</b>	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	控制输出
<b>JOB_REQ</b>	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	作业初始化(上升沿)

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
<b>JOB_ID</b>	INPUT	WORD	6	W#16#0000 无功能作业 W#16#0001 写计数值 W#16#0002 写载入值 W#16#0004 写比较值 W#16#0008 写滞后 W#16#0010 写脉冲周期 W#16#0082 读载入值 W#16#0084 读比较值 W#16#0088 读滞后 W#16#0090 读脉冲周期	W#16#0000	作业号
<b>JOB_VAL</b>	INPUT	DINT	8	$-2^{31}$ 至 $+2^{31}-1$	0	写作业的值。
<b>STS_GATE</b>	OUTPUT	BOOL	12.0	TRUE/FALSE	FALSE	内部门的状态
<b>STS_STRT</b>	OUTPUT	BOOL	12.1	TRUE/FALSE	FALSE	硬件门的状态 (开始输入)
<b>STS_LTCH</b>	OUTPUT	BOOL	12.2	TRUE/FALSE	FALSE	锁定输入的状态
<b>STS_DO</b>	OUTPUT	BOOL	12.3	TRUE/FALSE	FALSE	输出状态
<b>STS_C_DN</b>	OUTPUT	BOOL	12.4	TRUE/FALSE	FALSE	反方向状态 显示的始终为计数的 最后方向。第一次 调用SFB后, STS_C_DN值为 FALSE。

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
STS_C_UP	OUTPUT	BOOL	12.5	TRUE/FALSE	FALSE	正方向状态 显示的始终为计数的最后方向。第一次调用SFB后, STS_C_DN的值为TRUE。
COUNTVAL	OUTPUT	DINT	14	$-2^{31}$ 至 $+2^{31}-1$	0	实际计数值
LATCHVAL	OUTPUT	DINT	18	$-2^{31}$ 至 $+2^{31}-1$	0	实际锁定值
JOB_DONE	OUTPUT	BOOL	22.0	TRUE/FALSE	TRUE	可以启动新作业
JOB_ERR	OUTPUT	BOOL	22.1	TRUE/FALSE	FALSE	故障作业
JOB_STAT	OUTPUT	WORD	24	0至W#16# FFFF	0	作业出错编号

**注意**

如果已通过组态界面将“输出的反应”设置为“无比较”，则以下各项有效：

- 输出将以正常输出方式切换。
- SFB的输入参数CTRL\_DO和SET\_DO未激活。
- 状态位STS\_DO和STS\_CMP(IDB中的状态比较器)保持复位状态。

**未分配给块的参数(本地统计数据):**

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
STS_CMP	STATIC	BOOL	26.3	TRUE/FALSE	FALSE	状态比较器。 通过RES_STS 复位。 状态位STS_CMP指示 满足或已满足比较器 的比较条件。 STS_CMP还指示已设置 输出(STS_DO = TRUE)

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
STS_OFLW	STATIC	BOOL	26.5	TRUE/FALSE	FALSE	状态溢出 通过RES_STS复位。
STS_UFLW	STATIC	BOOL	26.6	TRUE/FALSE	FALSE	状态下溢 通过RES_STS复位。
STS_ZP	STATIC	BOOL	26.7	TRUE/FALSE	FALSE	过零状态 通过RES_STS复位。 只为无主计数方向的 计数器设置。 用于指示过零。 计数器设置为0或从载 入值=0开始计数时， 也会设置此值。
JOB_OVAL	STATIC	DINT	28	$-2^{31}$ 至 $+2^{31}-1$	0	读作业的输出值。
RES_STS	STATIC	BOOL	32.2	TRUE/FALSE	FALSE	复位状态位。 将状态位STS_CMP、 STS_OFLW、 STS_UFLW和 STS_ZP复位。 复位状态位时需要调 用两次SFB。

### 注意

有关使用SFB47的更多信息，请参见S7-300可编程控制器CPU 31xC技术功能手册。

## 出错信息

### 作业出错

如果发生作业出错，将置**JOB\_ERR = TRUE**。**JOB\_STAT**中将给出精确的出错原因。

事件类别 出错代码	说明
W#16#0121	比较值过低。
W#16#0122	比较值过高。
W#16#0131	滞后过小。
W#16#0132	滞后过大。
W#16#0141	脉冲周期过短。
W#16#0142	脉冲周期过长。
W#16#0151	载入值过低。
W#16#0152	载入值过高。
W#16#0161	计数器值过低。
W#16#0162	计数器值过高。
W#16#01FF	非法的作业号。

### 系统出错

发生系统出错后，将置**BIE = False**。

事件类别 出错代码	说明
W#16#8001	操作模式出错或参数出错。在“组态硬件”中设置正确的操作模式，或使用与已设置的操作模式相匹配的 <b>SFB</b>
W#16#8009	通道号非法。将通道号设置为 3(CPU专用值)。

## 26.4 使用SFB48 “FREQUENC”控制频率测量

### 描述

可通过用户程序操作频率计数器。这种情况下，必须使用**SFB FREQUENC (SFB48)**。

可以执行下列操作：

- 通过软件门**SW\_GATE**开始/停止
- 使能/控制输出**DO**
- 检索状态位**STS\_CMP**、**STS\_OFLW**和**STS\_UFLW**
- 检索实际频率值**MEAS\_VAL**
- 读取/写入内部频率计数器寄存器的作业

### 参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
<b>LADDR</b>	INPUT	WORD	0	CPU专用	W#16#0300	子模块的I/O地址，由用户在“HW配置”中指定。如果I和O地址不同，则必须指定二者中较低的一个。
<b>CHANNEL</b>	INPUT	INT	2	CPU 312C: 0到1 CPU 313C: 0到2 CPU 314C: 0至3	0	通道号:
<b>SW_GATE</b>	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	用于启动/停止频率测量的软件门
<b>MAN_DO</b>	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	手动输出控制使能
<b>SET_DO</b>	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	控制输出
<b>JOB_REQ</b>	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	作业初始化(上升沿)



参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
<b>JOB_ID</b>	INPUT	WORD	6	W#16#0000 = 无功能作业 W#16#0001 = 写下限 W#16#0001 = 写上限 W#16#0004 = 写集成时间 W#16#0081 = 读下限 W#16#0081 = 读上限 W#16#0084 = 读集成时间	0	作业号
<b>JOB_VAL</b>	INPUT	DINT	8	$-2^{31}$ 至 $+2^{31}-1$	0	写作业的值。
<b>STS_GATE</b>	OUTPUT	BOOL	12.0	TRUE/FALSE	FALSE	内部门的状态
<b>STS_STRT</b>	OUTPUT	BOOL	12.1	TRUE/FALSE	FALSE	硬件门的状态 (开始输入)
<b>STS_DO</b>	OUTPUT	BOOL	12.2	TRUE/FALSE	FALSE	输出状态
<b>STS_C_DN</b>	OUTPUT	BOOL	12.3	TRUE/FALSE	FALSE	反方向状态。 显示始终为计数的 最后方向。第一次 调用SFB后， STS_C_DN值为 FALSE。
<b>STS_C_UP</b>	OUTPUT	BOOL	12.4	TRUE/FALSE	FALSE	正方向状态 显示的始终为计数的 最后方向。第一次 调用SFB后， STS_C_UP的值为 TRUE。
<b>MEAS_VAL</b>	OUTPUT	DINT	14	0至 $+2^{31}-1$	0	实际频率值
<b>COUNTVAL</b>	OUTPUT	DINT	18	$-2^{31}$ 至 $+2^{31}-1$	0	实际计数值(每次 门在0时打开，开始 计数)
<b>JOB_DONE</b>	OUTPUT	BOOL	22.0	TRUE/FALSE	TRUE	可以启动新作业
<b>JOB_ERR</b>	OUTPUT	BOOL	22.1	TRUE/FALSE	FALSE	故障作业
<b>JOB_STAT</b>	OUTPUT	WORD	24	W#16#0000至 W#16#FFFF	W#16#0000	作业出错编号

**注意**

如果已通过组态界面将“输出的反应”设置为“无比较”，则以下各项有效：

- 输出将以正常输出方式切换。
- SFB输入参数MAN\_DO和SET\_DO未激活。
- 状态位STS\_DO保持复位状态。

**未分配给块的参数(本地统计数据):**

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
STS_CMP	STATIC	BOOL	26.3	TRUE/FALSE	FALSE	测量结束状态。 通过RES_STS复位。 每次时间间隔期满后， 将更新测量值。 在此，测量结束由状态位 STS_CMP报告
STS_OFLW	STATIC	BOOL	26.5	TRUE/FALSE	FALSE	状态溢出。 通过RES_STS复位。
STS_UFLW	STATIC	BOOL	26.6	TRUE/FALSE	FALSE	状态下溢 通过RES_STS复位。
JOB_OVAL	STATIC	DINT	28	$-2^{31}$ 至 $2^{31}-1$	0	读作业的输出值。
RES_STS	STATIC	BOOL	32.2	TRUE/FALSE	FALSE	复位状态位。 复位状态位 STS_CMP、 STS_OFLW、 STS_UFLW。 复位状态位时需要 调用两次SFB。

**注意**

有关使用SFB48的更多信息，请参见S7-300可编程控制器CPU 31xC技术功能手册。

## 作业出错

如果发生作业出错，则**JOB\_ERR = TRUE**。**JOB\_STAT**中将给出精确的出错原因。

事件类别 出错代码	说明
W#16#0221	集成时间过低。
W#16#0222	集成时间过高。
W#16#0231	频率下限过低。
W#16#0232	频率上限过高。
W#16#0241	频率上限过低。
W#16#0242	频率上限过高。
W#16#02FF	作业号非法。

## 系统出错

发生系统出错后，将置**BIE = False**。

事件类别 出错代码	说明
W#16#8001	操作模式出错或参数出错。在“组态硬件”中设置正确的操作模式，或使用与已设置的操作模式相匹配的 <b>SFB</b>
W#16#8009	通道号非法。将通道号设置为 <b>3</b> (CPU的专用值)。

## 26.5 使用SFB49 “PULSE”控制脉宽调制

### 描述

要通过用户程序控制脉宽调制，请使用**SFB ANALOG(SFB49)**。

可以执行下列操作：

- 通过软件门**SW\_EN**开始/结束
- 使能/控制输出**DO**
- 检索状态位**STS\_EN**、**STS\_STRT**和**STS\_DO**
- 输入输出值
- 读/写寄存器的作业

### 参数

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	含义
<b>LADDR</b>	INPUT	WORD	0	CPU专用	W#16#0300	子模块的I/O地址，由您在“HW配置”中指定。 如果E和A地址不相等，则必须指定二者中较低的一个。
<b>CHANNEL</b>	INPUT	INT	2	CPU 312C: 0到1 CPU 313C: 0到2 CPU 314C: 0至3	0	通道号:
<b>SW_EN</b>	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	用于开始/停止输出的软件门
<b>MAN_DO</b>	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	手动输出控制使能
<b>SET_DO</b>	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	控制输出
<b>OUTP_VAL</b>	INPUT	INT	6.0	单位ppm: 0至1,000 S7模拟值: 0至27,648	0	缺省输出值 如果输入的输出值 > 1000或27648, CPU会将其限制为1,000或27,648

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	含义
JOB_REQ	INPUT	BOOL	8.0	TRUE/FALSE	FALSE	作业初始化(上升沿)
JOB_ID	INPUT	WORD	10	W#16#0000 = 无功能作业 W#16#0001 = 写周期 W#16#0001 = 写延时 W#16#0004 = 写最小脉冲周期 W#16#0081 = 读周期 W#16#0081 = 读延时 W#16#0084 = 读最小脉冲周期	W#16#0000	作业号
JOB_VAL	INPUT	DINT	12	$-2^{31}$ 至 $+2^{31}-1$	0	写作业的值。
STS_EN	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	状态使能
STS_STRT	OUTPUT	BOOL	16.1	TRUE/FALSE	FALSE	硬件门的状态 (开始输入)
STS_DO	OUTPUT	BOOL	16.2	TRUE/FALSE	FALSE	输出状态
JOB_DONE	OUTPUT	BOOL	16.3	TRUE/FALSE	TRUE	可以启动新作业
JOB_ERR	OUTPUT	BOOL	16.4	TRUE/FALSE	FALSE	故障作业
JOB_STAT	OUTPUT	WORD	18	W#16#0000至 W#16#FFFF	W#16#0000	作业出错编号

## 未分配给块的参数(本地统计数据):

参数	声明	数据类型	地址 (背景数据块)	取值范围:	缺省值	描述
JOB_OVAL	OUTPUT	DINT	20	$-2^{31}$ 至 $2^{31}-1$	0	读作业的输出值。

**注意**

有关使用SFB 49的更多信息，请参见S7-300可编程控制器CPU 31xC技术功能手册。

## 作业出错

如果发生作业出错，则**JOB\_ERR = TRUE**。**JOB\_STAT**中将给出精确的出错原因。

事件类别 出错代码	说明
W#16#0411	周期过短。
W#16#0412	周期过长。
W#16#0421	延时过短。
W#16#0422	延时过长。
W#16#0431	最小脉冲周期过短。
W#16#0432	最小脉冲周期过长。
W#16#04FF	作业号非法。

## 系统出错

发生系统出错后，将置**BIE = False**。

事件类别 出错代码	说明
W#16#8001	操作模式出错或参数出错。在“组态硬件”中设置正确的操作模式，或使用与已设置的操作模式相匹配的 <b>SFB</b>
W#16#8009	通道号非法。将通道号设置为 <b>3</b> (CPU的专用值)。

## 26.6 使用SFB60 “SEND\_PTP”发送数据(ASCII, 3964(R))

### 描述

可通过**SFB SEND\_PTP (SFB60)**从DB传送数据块。

调用数据块且控制输入**REQ**为上升沿后，将执行发送操作。

要传送的数据范围由**SD\_1**(DB编号和起始地址)确定。数据块的长度由**LEN**确定。

要能使**SFB**处理作业，必须用**R(复位)=FALSE**调用。当控制输入**R**为上升沿时，将中止当前传送并将**SFB**复位为基本状态。被中止的作业将以出错消息(**STATUS**输出)终止。

对于子模块，应在**LADDR**中声明已在“HW配置”中指定的I/O地址。

如果作业无故障终止，则将**DONE**设置为**TRUE**，如果作业出错终止，则将**ERROR**设置为**TRUE**。

如果处理作业后**DONE=TRUE**，则表示：

- 使用**ASCII**驱动程序时：数据已传送给通讯伙伴。不保证通讯伙伴已收到所有数据。
- 使用过程**3964(R)**时：数据已传送给通讯伙伴，并且通讯伙伴已确认收到。不保证数据传送到通讯伙伴的**CPU**。

在**STATUS**中，**CPU**将指示出错，或出现警告时指示各自的事件ID。

“复位”(R=TRUE) **SFB**时，还将输出**DONE**或**ERROR/STATUS**。

如果已发生出错，则复位二进制结果**BIE**。如果块无错终止，二进制结果的状态为**TRUE**。

### 注意

**SFB**中不包括参数检查。如果参数化出现出错，**CPU**可能会进入**STOP**模式。

### 背景数据块

执行**SFB SEND\_PTP**时还需要背景数据块。**DB**编号通过调用传递。不允许访问背景数据块中的数据。

参数

参数	声明	数据类型	取值范围:	缺省值	描述
REQ	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“请求”: 在上升沿激活数据交换。
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”。中止作业。锁定传送。
LADDR	INPUT	WORD	CPU专用	W#16#03FF	子模块的I/O地址, 由用户在“HW配置”中指定。
DONE	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数(此参数仅为调用持续时间设置): 1. FALSE表示作业尚未开始或仍在执行。 2. TRUE表示作业已执行, 未出错。
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数(此参数仅为调用持续时间设置): 作业完成, 无故障
STATUS	OUTPUT	WORD	W#16#0000至 W#16#FFFF	W#16#0000	状态参数(此参数仅为调用持续时间设置): 要显示状态, 应将STATUS复制到空闲数据区) 根据ERROR位的不同, STATUS具有下列含义: <ul style="list-style-type: none"> <li>• ERROR=FALSE: STATUS的值为 W#16#0000: 既不是警告也不是出错 STATUS的值 &lt;&gt; W#16#0000: 警告, STATUS提供详细信息。</li> <li>• ERROR=TRUE: 发生出错, STATUS提供有关出错类型的详细信息。</li> </ul>



参数	声明	数据类型	取值范围:	缺省值	描述
<b>SD_1:</b>	INPUT/ OUTPUT	ANY	CPU专用	0	发送参数: 在此输入下列值: 1. 要从中传送数据的DB 的编号。 2. 数据字节编号, 数据将从此 编号开始传送。 例如: 从DB 10的第2字节开始 -> DB10.DBB2
<b>LEN</b>	INPUT/ OUTPUT	INT	1 - 1024	1	在此声明要传送的数据块的 长度。(在此间接设置长度。)

### 数据的一致性

数据的一致性限制为206个字节。如果要使超过206个字节的数据传送保持一致性, 必须注意以下的问题:

除非传送已终止, 否则请勿向发送范围SD\_1的当前使用部分写入数据。  
这种情况下, 状态参数DONE的值为TRUE。

## 26.7 使用SFB61 “RCV\_PTP”接收数据(ASCII, 3964(R))

### 描述

使用**SFB RCV\_PTP (SFB61)**可以接收数据，然后将其归档到数据块。

数据块被调用后，且控制输入**EN\_R**为**TRUE**时，数据块即准备就绪接收数据。将参数**EN\_R**的信号状态置为**FALSE**，则可取消当前传送。被取消的作业将以出错消息(**STATUS**输出)终止。只要参数**EN\_R**的信号状态置为**FALSE**，此输入关闭。

接收区域在**RD\_1**(DB编号和起始地址)中声明。数据块长度在**LEN**中声明。

要能使**SFB**处理作业，必须用**R(复位)=FALSE**调用。当控制输入**R**为上升沿时，将中止当前传送并将**SFB**复位为基本状态。被取消的作业将以出错消息(**STATUS**输出)终止。

对于子模块，应在**LADDR**中声明已在“HW配置”中指定的I/O地址。

如果作业无故障终止，则将**NDR**置为**TRUE**，如果作业出错终止，则将**ERROR**置为**TRUE**。

在**STATUS**中，CPU将指示出错，或者出现警告时，指示各自的事件。

“复位”(R=TRUE) SFB时，还将输出**NDR**或**ERROR/STATUS**(参数**LEN = 16#00**)。

如果已发生出错，则复位二进制结果**BIE**。如果块无错终止，二进制结果的状态为**TRUE**。

### 注意

**SFB**中不包括参数检查。如果组态出现出错，CPU可能会跳转到**STOP**模式。

### 背景数据块

执行**SFB RCV\_PTP**时还需要背景数据块。DB编号通过调用传递。不允许访问背景数据块中的数据。

## 参数

参数	声明	数据类型	取值范围:	缺省值	描述
EN_R	INPUT	BOOL	TRUE/ FALSE	FALSE	控制参数“使能接收”： 接收使能
R	INPUT	BOOL	TRUE/ FALSE	FALSE	控制参数“复位”。 中止作业。
LADDR	INPUT	WORD	CPU专用	W#16#03FF	子模块的I/O地址，由用户在“HW配置” 中指定。
NDR	OUTPUT	BOOL	TRUE/ FALSE	FALSE	作业完成，无故障，数据被接受。 <ul style="list-style-type: none"> <li>FALSE表示作业尚未开始或仍在执行</li> <li>TRUE表示作业已成功完成。</li> </ul>
ERROR	OUTPUT	BOOL	TRUE/ FALSE	FALSE	状态参数(此参数仅为调用持续时间 设置): 作业完成，无故障
STATUS	OUTPUT	WORD	W#16#0000 至 W#16#FFFF	W#16#0000	状态参数(此参数仅为调用持续时间 设置): 要显示状态，应将STATUS 复制到空闲数据区) 根据ERROR位的不同，STATUS 具有下列含义: <ul style="list-style-type: none"> <li>ERROR=FALSE: STATUS的值为W#16#0000: 既不是警告也不是出错 STATUS的值 &lt;&gt; W#16#0000: 警告，STATUS提供详细信息。</li> <li>ERROR=TRUE: 发生出错，STATUS提供有关出 错类型的详细信息。</li> </ul>
RD_1	INPUT/ OUTPUT	ANY	CPU专用	0	接收参数: 在此声明: 1. 要存储已接收数据的DB的编号。 2. 数据字节编号，数据将从此编号 开始存储。 例如: 从DB 20的第5字节开始 -> DB10.DBB2
LEN	INPUT/ OUTPUT	INT	0至1024	0	数据长度(字节数)输出

## 数据的一致性

数据的一致性限制为206字节。如果要使超过206字节的数据传送保持一致性，必须注意如下问题：

全部数据都已接收(NDR = TRUE)前，请勿访问接收DB。然后，锁定接收DB(EN\_R = FALSE)，直到完成数据处理。

## 26.8 使用SFB62 “RES\_RCVB”删除接收缓冲区(ASCII, 3964(R))

### 描述

使用**SFB RES\_RECV(SFB62)**可以清除模块的整个输入缓冲区。系统将放弃存储的所有报文。调用**SFB RES\_RCVB**时，会存储进入的消息帧。

调用数据块后且控制输入**REQ**为上升沿，将激活作业。作业可在多个调用(程序周期)间运行。

要能使**SFB**处理作业，调用**R(复位)=FALSE**调用它。当控制输入**R**为上升沿时，将取消删除过程并将**SFB**复位为基本状态。被取消的作业将以出错消息(**STATUS**输出)终止。

对于子模块，应在**LADDR**中声明已在“HW配置”中指定的I/O地址。

如果作业无故障终止，则将**DONE**置为**TRUE**，如果作业出错终止，则将**ERROR**置为**TRUE**。

在**STATUS**中，CPU将指示出错，或者出现警告时，指示各自的事件ID。

“复位”(R=TRUE) **SFB**时，还将输出**DONE**或**ERROR/STATUS**。

如果已发生出错，则复位二进制结果**BIE**。如果块无错终止，二进制结果的状态将为**TRUE**。

### 注意

**SFB**中不包括参数检查。如果参数化出现出错，CPU可能会跳转到**STOP**模式。

### 背景数据块

执行**SFB RES\_RCVB**时还需要背景数据块。DB编号通过调用传递。不允许访问背景数据块中的数据。

## 参数

参数	声明	数据类型	取值范围:	缺省值	描述
REQ	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“请求”： 在上升沿激活作业。
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”。 中止作业。
LADDR	INPUT	WORD	CPU专用	W#16#03FF	在“HW配置”中设置的子模块的I/O地址。
DONE	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数(此参数仅为调用持续时间设置): <ul style="list-style-type: none"> <li>FALSE表示作业尚未开始或仍在执行。</li> <li>TRUE表示作业已执行, 未出错。</li> </ul>
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数(此参数仅为调用持续时间设置): 作业完成, 无故障
STATUS	OUTPUT	WORD	W#16#0000至 W#16#FFFF	W#16#0000	状态参数(此参数仅为调用持续时间设置): 要显示状态, 应将STATUS复制到空闲数据区) 根据ERROR位的不同, STATUS具有下列含义: <ul style="list-style-type: none"> <li>ERROR=FALSE: STATUS的值为W#16#0000: 既不是警告也不是出错 STATUS的值 &lt;&gt; W#16#0000: 警告, STATUS提供详细信息。</li> <li>ERROR=TRUE: 发生出错, STATUS提供有关出错类型的详细信息。</li> </ul>

## 26.9 使用SFB63 “SEND\_RK”发送数据(512(R))

### 描述

可通过**SFB SEND\_PTP (SFB63)**从数据块发送数据。

调用数据块后且控制输入**REQ**为上升沿，将执行发送操作。

要传送的数据范围由**SD\_1**(DB编号和起始地址)确定。数据块的长度由**LEN**确定。

在**SFB**中，还可声明伙伴站的接收范围。CPU在消息帧报头中输入此信息(请参见附录F)，并将其传送给伙伴。

目标由以下各项指定：**CPU**号**R\_CPU**(仅适用于多处理器通讯)、**R\_TYPE**中的数据块类型(数据块(DB)和扩展数据块(DX))、**R\_DBNO**中的数据块号以及**R\_OFFSET**中距写入第一字节的偏移量。

在**R\_CF\_BYT**和**R\_CF\_BIT**中，声明伙伴CPU上的连接存储器字节和位。

在参数**SYNC\_DB**中，声明将在其中存储数据的DB，该数据用于所有**SFB**在启动和同步期间的初始化。用户程序中所有**SFB**的DB编号必须相同。

要使**SFB**能够处理作业，必须用**R(复位)=FALSE**调用它。当控制输入**R**为上升沿时，将取消当前发送操作并将**SFB**复位为基本状态。被取消的作业将以出错消息(**STATUS**输出)终止。

对于子模块，应在**LADDR**中声明已在“HW配置”中指定的I/O地址。

如果作业无故障终止，则将**DONE**设置为TRUE，如果作业出错终止，则将**ERROR**设置为TRUE。

处理完作业且**DONE = TRUE**后，数据被发送到通讯伙伴，通讯伙伴确认收到并将其传送到伙伴的CPU。

在**STATUS**中，CPU将指示出错，或者出现警告时，指示相应的事件标识符。

“复位”(R=TRUE) **SFB**时，还会输出**DONE**或**ERROR/STATUS**。

如果已发生出错，将复位二进制结果**BIE**。如果块无错终止，二进制结果的状态将为TRUE。

### 注意

**SFB**中不包括参数检查。如果为CPU分配了出错的参数，它可能会跳转到**STOP**模式。

### 背景数据块

执行**SFB SEND\_RK**时还需要背景数据块。DB编号通过调用传递。不允许访问背景数据块中的数据。

## 发送数据的特性

“发送”数据时，要考虑下列特性：

- 使用RK512，只能发送偶数数据。如果声明了奇数长度(LEN)的数据，将会在所发送的数据后额外追加具有值“O”的附加填充字节。
- 在RK512中，只能声明偶数偏移量。如果声明了奇数偏移量，数据将从下一个较小偶数偏移量开始存储在伙伴站中。

实例：偏移量是7，则从第6字节开始存储数据。

## 参数

参数	声明	数据类型	取值范围：	缺省值	描述
SYNC_DB	INPUT	INT	CPU专用，不允许零	0	存储用于RK-SFB的同步的公用数据的数据块的编号(最小长度 = 240 字节)。
REQ	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“请求”： 在上升沿激活作业。
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”。 中止作业。
LADDR	INPUT	WORD	CPU专用	W#16#03FF	子模块的I/O地址，由用户在“HW配置”中指定。
R_CPU	INPUT	INT	0至4	1	伙伴CPU的CPU编号 (仅用于多处理器操作)
R_TYPE	INPUT	CHAR	“D”、“X”	“D”	伙伴CPU上的地址类型 (只允许大写字母) “D”：数据块 “X”：扩展数据块
R_DBNO	INPUT	INT	0 - 255	0	伙伴CPU上的数据块编号
R_OFFSET	INPUT	INT	0至510 (只限偶数值)	0	伙伴CPU上的数据字节编号
R_CF_BYT	INPUT	INT	0 - 255	255	伙伴CPU上的连接存储器位 (255：表示：无连接存储器位)
R_CF_BIT	INPUT	INT	0至7	0	伙伴CPU上的连接存储器位
DONE	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数(此参数仅为调用持续时间设置)： 1. FALSE表示作业尚未开始或仍在执行。 2. TRUE表示作业已执行，未出错。
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数(此参数仅为调用持续时间设置)： 作业完成，无故障



参数	声明	数据类型	取值范围:	缺省值	描述
<b>STATUS</b>	OUTPUT	WORD	W#16#0000 至 W#16#FFFF	W#16#0000	<p>状态参数(此参数仅为调用持续时间设置): 要显示状态, 应将STATUS复制到空闲数据区)</p> <p>根据ERROR位, STATUS具有下列含义:</p> <ul style="list-style-type: none"> <li>• <b>ERROR=FALSE:</b> STATUS的值为W#16#0000: 既不是警告也不是出错 STATUS的值 &lt;&gt; W#16#0000: 警告, STATUS提供详细信息。</li> <li>• <b>ERROR=TRUE:</b> 发生出错, STATUS提供有关出错类型的详细信息。</li> </ul>
<b>SD_1:</b>	INPUT/ OUTPUT	ANY	CPU专用	0	<p>发送参数:</p> <p>在此声明:</p> <ol style="list-style-type: none"> <li>1. 要从中传送数据的DB的编号。</li> <li>2. 数据字节编号, 数据将从此编号开始传送。</li> </ol> <p>例如: 从DB 10的第2字节开始 -&gt; DB10.DBB2</p>
<b>LEN</b>	INPUT/ OUTPUT	INT	1 - 1024	1	<p>在此声明要传送的数据块的长度。 (在此间接设置长度。)</p>

## 消息帧中的声明

下表显示了RK 512消息帧的消息帧报头中的声明。

S7自动化系统上的源 (本地CPU)	目标 伙伴CPU	消息帧报头, 字节		
		3/4指令类型	5/6 D-DBNR/D 偏移量	7/8计数单位为 偏移量
数据块	数据块	AD	DB/DW	字
数据块	扩展数据块	AD	DB/DW	字

所用缩写的解释:

D-DBNR	目标数据块编号
D偏移量	目标起始地址
DW	以字计的偏移量

## 数据的一致性

数据的一致性限制为128字节。如果要使超过128字节的数据传送保持一致性，必须注意如下问题：

除非传送已终止，否则请勿向发送范围SD\_1的当前使用部分写入数据。这种情况下，状态参数DONE的值为TRUE。

## 26.10 使用SFB64 “FETCH RK”获取数据(RK 512)

### 描述

**SFB FETCH\_RK(SFB64)**用于从伙伴获取数据块并将其存储在数据块中。

调用数据块后且控制输入**REQ**为上升沿，将执行发送操作。

存储所获取数据的区域在**RD\_1**(DB编号和起始地址)中声明。数据块长度在**LEN**中声明。

在**SFB**中，还可指定从中获取数据的伙伴区域。CPU在**RK512**消息帧报头中输入该信息，并将其传送给伙伴。

伙伴区域由以下各项确定：**R\_CPU**中的CPU编号(只适用于多处理器通讯)、**R\_TYPE**中的数据类型(数据块、扩展数据块、存储器位、输入、输出、计数器和定时器)、**R\_DBNO**中的数据块编号(数据块、扩展数据块、存储器位、输入、输出、计数器和定时器)和**R\_OFFSET**中获取第一字节的偏移量。

在**R\_CF\_BYT**和**R\_CF\_BIT**中，可声明伙伴CPU上的连接存储字节和连接存储位。

在参数**SYNC\_DB**中，声明将在其中存储数据的DB，该数据用于所有**SFB**在启动和同步期间的初始化。用户程序中所有**SFB**的DB编号必须相同。

要能使**SFB**处理作业，必须用**R(复位)=FALSE**调用它。当控制输入**R**为上升沿时，将取消当前传送，并将**SFB**复位为基本状态。取消的作业将以出错消息(**STATUS**输出)结束。

对于子模块，应在**LADDR**中声明已在“HW配置”中指定的I/O地址。

如果作业无故障终止，则将**DONE**置为**TRUE**，如果作业出错终止，则将**ERROR**置为**TRUE**。

在**STATUS**中，CPU将指示出错，或者出现警告时，指示各自的事件ID。

“复位”(R=TRUE) **SFB**时，还将输出**DONE**或**ERROR/STATUS**。

如果已发生出错，则复位二进制结果**BIE**。如果块无错终止，二进制结果的状态为**TRUE**。

### 注意

**SFB**中不包括参数检查。如果参数化出错，CPU可能会跳转到**STOP**模式。

### 注意

从CPU获取数据时，必须为CPU编程**SFB “SERVE\_RK”**。

## 背景数据块

执行SFB FETCH\_RK时还需要背景数据块。DB编号通过调用传递。不允许访问背景数据块中的数据。

## (扩展)数据块的特性

从数据块或扩展数据块“获取数据”时，请注意下列特性：

- 使用RK512，只能发送偶数数据。如果输入奇数长度(LEN)，则会传送一个附加字节。但是，在目标数据块中，始终输入正确的数据数。
- 在RK512中，只能声明偶数偏移量。如果声明了奇数偏移量，数据将从下一个较小偶数偏移量开始存储在伙伴站中。

实例：偏移量是7，则从第6字节开始存储数据。

## 定时器和计数器的特性

从通讯伙伴获取定时器或计数器时，必须注意要为每个定时器或计数器拿来两个字节。例如，如果要获取10个计数器，则必须声明长度为20。

## 参数

注意：在SFB中，值的范围都以德国存储器惯例表示。

参数	声明	数据类型	取值范围：	缺省值	描述
SYNC_DB	INPUT	INT	CPU专用，不允许零	0	存储用于RK-SFB同步的公用数据的数据块的编号(最小长度 = 240字节)。
REQ	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“请求”：在上升沿激活作业。
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”。中止作业。
LADDR	INPUT	WORD	CPU专用	W#16#03FF	子模块的I/O地址，由用户在“HW配置”中指定。
R_CPU	INPUT	INT	0至4	1	伙伴CPU的CPU编号(仅用于多处理器操作)

参数	声明	数据类型	取值范围:	缺省值	描述
R_TYPE	INPUT	CHAR	“D”、“X”、“M”、 “E”、“A”、“Z”、 “T”	“D”	伙伴CPU上的地址类型 “D”: 数据块 “X”: 扩展数据块 “M”: 存储器位 “E”: 输入 “A”: 输出 “Z”: 计数器 “T”: 定时器
R_DBNO	INPUT	INT	0 - 255	0	伙伴CPU上的数据块编号
R_OFFSET	INPUT	INT	参考表格: ”数据源(伙伴 CPU)功能块中的 参数”	0	伙伴CPU上的数据字节编号
R_CF_BYT	INPUT	INT	0 - 255	255	伙伴CPU上的连接存储器位 (255: 表示: 无连接存储 器位)
R_CF_BIT	INPUT	INT	0至7	0	伙伴CPU上的连接存储器位
DONE	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数(此参数仅为调用 持续时间设置): <ul style="list-style-type: none"> <li>FALSE表示作业尚 未开始或仍在执行。</li> <li>TRUE表示作业已 执行, 未出错。</li> </ul>
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数(此参数仅为调用持 续时间设置): 作业完成, 无故障

参数	声明	数据类型	取值范围:	缺省值	描述
<b>STATUS</b>	OUTPUT	WORD	W#16#0000至 W#16#FFFF	W#16#0000	<p>状态参数(此参数仅为调用持续时间设置): 要显示状态, 应将STATUS复制到空闲数据区)</p> <p>根据ERROR位的不同, STATUS具有下列含义:</p> <ul style="list-style-type: none"> <li>• <b>ERROR=FALSE:</b> STATUS的值为 W#16#0000: 既不是警告也不是出错STATUS的值 &lt;&gt; W#16#0000: 警告, STATUS提供详细信息。</li> <li>• <b>ERROR=TRUE:</b> 发生出错, STATUS提供有关出错类型的详细信息。</li> </ul>
<b>RD_1</b>	INPUT/ OUTPUT	ANY	CPU专用	0	<p>接收参数:</p> <p>在此声明:</p> <ol style="list-style-type: none"> <li>1. 存储获取数据的数据块编号。</li> <li>2. 数据字节编号, 获取的数据将从此编号开始存储。</li> </ol> <p>例如: 从DB 10的第2字节开始 -&gt; DB10.DBB2</p>
<b>LEN</b>	INPUT/ OUTPUT	INT	1 - 1024	1	<p>在此声明要获取的数据块的字节长度。 每个定时器和每个计数器必须声明两字节。 (在此间接设置长度。)</p>

## 数据源(伙伴CPU)SFB中的参数

下表给出了可发送的数据类型。  
R\_OFFSET的值由伙伴CPU确定。

伙伴CPU上的源	R_TYP	R_NO	R_OFFSET(单位: 字节)
数据块	"D"	0 - 255	0 - 510 只适用偶数值
扩展数据块	"X":	0 - 255	0 - 510 只适用偶数值
存储器位	"M"	不相关	0 - 255
输入	"E"	不相关	0 - 255
输出	"A"	不相关	0 - 255
计数器	"Z"	不相关	0 - 255
定时器	"T"	不相关	0 - 255

## 消息帧中的声明

下表给出了RK512消息帧的消息帧报头中的声明。

伙伴CPU上的源	目标, S7自动化系统(本地CPU)	消息帧报头, 字节		
		3/4指令类型	5/6 S-DBNR/S 偏移量	7/8数目, 单位为
数据块	数据块	ED	DB/DW	字
扩展数据块	数据块	EX	DB/DW	字
存储器位	数据块	EM	字节地址	字节
输入	数据块	EI	字节地址	字节
输出	数据块	EO	字节地址	字节
计数器	数据块	EC	计数器编号	字
定时器	数据块	ET	定时器编号	字

所用缩写的解释:

S-DBNO	源数据块编号
S偏移量	源起始地址

## 数据的一致性

数据的一致性限制为128字节。如果要使超过128字节的数据传送保持一致性, 必须注意如下问题:

除非传送已终止, 否则请勿向发送范围SD\_1的当前使用部分写入数据。这种情况下, 状态参数DONE的值设置为TRUE。

## 26.11 使用SFB65 “SERVE\_RK”接收和提供数据(RK 512)

### 描述

使用**SFB SERVE\_RK (SFB65)**接收和提供数据。

- 接收数据：数据存储在由伙伴在RK512消息帧报头中指定的区域中。通讯伙伴执行“发送数据”(SEND作业)作业时，需要调用**SFB**。
- 提供数据：数据从由伙伴在RK512消息帧报头中的指定的区域中获取。通讯伙伴执行“获取数据”(FETCH作业)作业时，需要调用**SFB**。

块被调用后，且控制输入**EN\_R**的值为**TRUE**时，即准备就绪接收数据。通过将参数**EN\_R**的信号状态置为**FALSE**，可取消当前传送。被取消的作业将以出错消息(**STATUS**输出)终止。只要参数**EN\_R**的信号状态置为**FALSE**，则输入关闭。

在参数**SYNC\_DB**中，声明将在其中存储数据的**DB**，该数据用于所有**SFB**在启动和同步期间的初始化。用户程序中所有**SFB**的**DB**编号必须相同。

要能使**SFB**处理作业，必须用**R(复位)=FALSE**调用它。当控制输入**R**为上升沿时，将取消当前传送，并将**SFB**复位为基本状态。被取消的作业将以出错消息(**STATUS**输出)终止。

对于子模块，应在**LADDR**中声明已在“HW配置”中指定的I/O地址。

如果作业无故障终止，则将**NDR**置为**TRUE**，如果作业出错终止，则将**ERROR**置为**TRUE**。

调用**SFB**时，如果**NDR=TRUE**，CPU将在参数**L\_TYPE**、**L\_DBNO**和**L\_OFFSET**中指定要将数据存储到的区域或从中获取数据的区域。系统还会为调用显示相关作业的**L\_CF\_BYT**和**L\_CF\_BIT**以及长度**LEN**参数。

在**STATUS**中，CPU将指示出错，或者出现警告时，指示各自的事件标ID(请参考附录)。

“复位”(R=TRUE) **SFB**时，还将输出**NDR**或**ERROR/STATUS**(参数**LEN = 16#00**)。

如果已发生出错，则复位二进制结果**BIE**。如果块无错终止，二进制结果的状态为**TRUE**。

### 注意

**SFB**中不包括参数检查。如果组态出现出错，CPU可能会进入**STOP**模式。



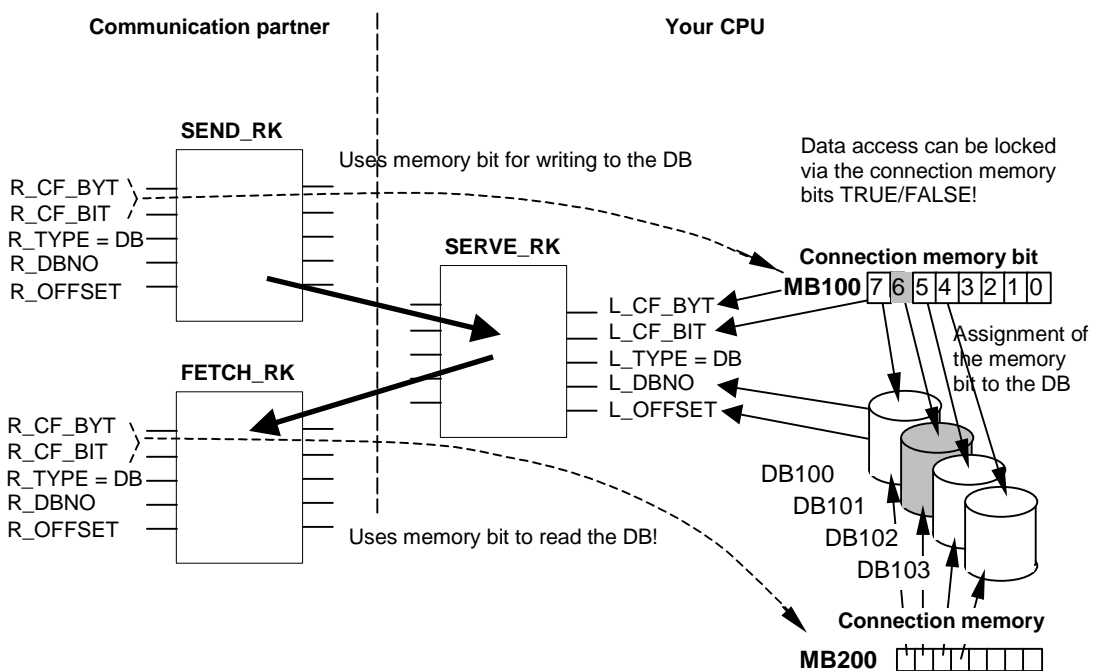
## 背景数据块

执行SFB SERVE\_RK时还需要背景数据块。DB编号通过调用传递。不允许访问背景数据块中的数据。

## 如何使用连接存储器位

通过连接存储器位可以锁定或使能通讯伙伴的SEND和FETCH作业。从而可防止覆盖或读取还未处理的数据。

可以为每个作业都指定连接存储器位。



## 实例：使用连接存储器位执行SEND\_RK

在本例中，通讯伙伴将数据传送到用户的CPU上的DB 101

1. 在用户的CPU中，将连接存储器位100.6设置为FALSE。
2. 在通讯伙伴中，为SEND作业指定连接存储器位100.6(参数R\_CF\_BYT、R\_CF\_BIT)。

连接存储器位在RK 512消息帧报头中传送到CPU。

处理作业前，CPU会验证在RK512消息帧报头中指定的连接存储器位。只有在CPU上将连接存储器位置为FALSE时，才会处理作业。如果连接存储器位置为TRUE，出错消息“32 hex”将通过响应消息帧传送到通讯伙伴。

数据传送到DB101后，SFB SERVE将把用户CPU上的连接存储器100.6置为TRUE。另外，在调用的持续时间，SFB SERVE上将输出连接存储器字节和位(如果NDR = TRUE)。

3. 在用户程序中判断连接存储器(连接存储器100.6 = TRUE)时，可查看作业是否已完成以及是否可以处理所传送的数据。
4. 在用户程序中处理完数据后，必须将连接存储器100.6复位到FALSE。否则，伙伴将不能再次无错执行该作业。

## 参数

注意：在SFB中，值的范围都以德国存储器惯例表示。

参数	声明	数据类型	取值范围:	缺省值	描述
SYNC_DB	INPUT	INT	CPU专用	0	存储用于RK-SFB同步的公用数据的数据块编号(最小长度 = 240字节)。
EN_R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“使能接收”作业使能
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”。中止作业。
LADDR	INPUT	WORD	CPU专用	W#16#03FF	子模块的I/O地址，由用户在“HW配置”中指定。
NDR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数“新数据就绪”(此参数仅为调用持续时间设置): 1. FALSE表示作业尚未开始或仍在执行。 2. TRUE表示作业已成功执行。
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数(此参数仅为调用持续时间设置): 作业完成，无故障

参数	声明	数据类型	取值范围:	缺省值	描述
<b>STATUS</b>	OUTPUT	WORD	W#16#0000至W#16#FFFF	W#16#0000	<p>状态参数(此参数仅为调用持续时间设置): 要显示状态, 应将STATUS复制到空闲数据区)</p> <p>根据ERROR位的不同, STATUS具有下列含义:</p> <ul style="list-style-type: none"> <li>• <b>ERROR=FALSE:</b> STATUS的值为W#16#0000: 既不是警告也不是出错STATUS的值 &lt;&gt; W#16#0000: 警告, STATUS提供详细信息。</li> <li>• <b>ERROR=TRUE:</b> 发生出错, STATUS提供有关出错类型的详细信息。</li> </ul>
<b>L_TYPE</b>	OUTPUT	CHAR	<p>“D”</p> <p>“D”、“X”、“M”、“E”、“A”、“Z”、“T”</p>	''	<p>接收数据: 本地CPU上的目标区域类型(只允许大写字母): “D”: 数据块</p> <p>提供数据: 本地CPU上的源区域类型(只允许大写字母): “D”: 数据块 “M”: 存储器位 “E”: 输入 “A”: 输出 “Z”: 计数器 “T”: 定时器</p> <p>该参数仅对调用的持续时间设置。</p>
<b>L_DBNO</b>	OUTPUT	INT	CPU专用	0	本地CPU上的数据块编号。该参数仅对调用的持续时间设置。
<b>L_OFFSET</b>	OUTPUT	INT	0 - 510	0	本地CPU上的数据字节编号。该参数仅对调用的持续时间设置。

参数	声明	数据类型	取值范围:	缺省值	描述
L_CF_BYT	OUTPUT	INT	0 - 255	0	本地CPU上的连接存储器字节。该参数仅对调用的持续时间设置。 (255: 表示: 无连接存储器位)
L_CF_BIT	OUTPUT	INT	0至7	0	本地CPU上的连接存储器位。该参数仅对调用的持续时间设置。
LEN	INPUT/ OUTPUT	INT	0至1024	0	消息帧的长度, 以字节计算的数目(该参数仅对调用的持续时间设置)。

### 数据的一致性

数据的一致性限制为128字节。如果要使超过128字节的数据传送保持一致性, 必须注意如下问题:

使用连接存储器功能。数据全部传送完毕前不要访问数据(对该作业指定的连接存储器的判断; 如果NDR = TRUE, 则连接存储器在调用SFB时处于激活状态)。完成数据处理前, 不要将连接存储器复位为FALSE。

## 26.12 SFB 60至65的附加出错信息

### 出错信息

下表给出了不同的事件类别和事件标识号。

“SFB参数组态中的出错”		
事件类别 出错代码	事件	纠正方法
W#16#0301	源/目标数据类型非法或不存在。 非法范围(起始地址, 长度)。 DB非法或不存在。 其它数据类型非法或不存在。 连接存储器字节或位标识号无效。	验证参数设置, 如有需要进行更正。 伙伴在消息帧报头中传递了非法参数。 验证参数, 如有需要则创建一个块。 有关允许的数据类型的信息, 请参考作业表。 伙伴在消息帧报头中传递了出错参数。
W#16#0303	无法访问的范围	验证参数。有关允许的起始地址和长度的信息, 请参考作业表, 或者伙伴在消息帧报头中提供了一个出错参数。
“作业处理出错”		
事件类别 出错代码	事件	纠正方法
W#16#0501	当前作业因重启动或复位而中止。	重复执行被中止的作业。当通过PG重新分配参数时, 应确保写入接口前没有其它正在处理的作业。
W#16#0502	处于此操作状态时, 不允许该作业(例如, 没有为设备接口分配任何参数)。	为设备接口分配参数。
W#16#050E	<ul style="list-style-type: none"> <li>消息帧长度无效</li> <li>在允许的最大长度内, 未出现在参数中分配的消息结束标识号。</li> </ul>	<ul style="list-style-type: none"> <li>消息帧长度 &gt; 1024 字节。选择较小的消息帧长度</li> <li>在发送缓冲区中期望的位置添加消息结束标识号。</li> </ul>
W#16#0513	数据类型出错(DB ...): 未知数据类型或非法数据类型(例如, DE) 在SFB中指定的源和目标数据类型不匹配。	有关允许的数据类型及其组合的信息, 请参考作业表。
W#16#0515	协调存储器中声明的位编号出错。	允许的位编号为0至7
W#16#0516	指定的CPU编号太高。	允许的CPU编号为0、1、2、3或4
W#16#0517	传送的数据 > 1024字节, 过大	将作业拆分成几个较小长度的作业。

“作业处理出错”		
事件类别 出错代码	事件	纠正方法
W#16#051D	发送/接收作业因以下原因中止 <ul style="list-style-type: none"> <li>• 通讯块复位</li> <li>• 重新分配了参数</li> </ul>	重复调用通讯块。
W#16#0522	已开始了新的SEND作业，即使前一作业尚未完成。	前一发送作业以DONE或ERROR终止前，不要开始新的SEND作业。
“处理伙伴作业时的出错”(仅限RK512)		
事件类别 出错代码	事件	纠正方法
W#16#0601	第一个指令字节出错 (非00或FFH)	伙伴中的基本报头结构出错。如有必要，可通过将接口测试仪挂接至数据链路来检验伙伴设备的出错情况。
W#16#0602	第三个指令字节出错 (非A、0或E)	伙伴中的基本报头结构出错。如有必要，可通过将接口测试仪挂接至数据链路来检验伙伴的出错情况。
W#16#0603	后续消息帧的第三个指令字节出错 (指令与第一个消息帧中的指令不同)	伙伴中的基本报头结构出错。如有必要，可通过将接口测试仪挂接至数据链路来检验伙伴的出错情况。
W#16#0604	第四个指令字节出错(指令字符出错)	伙伴的基本报头结构出错或请求了非法指令组合。检查允许的指令。如有必要，可通过将接口测试仪挂接至数据链路来检验伙伴的出错情况。
W#16#0606	第五个指令字节出错 (非法数据块编号)	有关允许的数据块编号、起始地址或长度，请参考作业表。
W#16#0607	第五个或第六个指令字节出错 (起始地址过高)	有关允许的数据块编号、起始地址或长度，请参考作业表。
W#16#0609	第九个或第十个指令字节出错(该数据类型不允许协调存储器位或位号太高)。	伙伴中的基本报头结构出错。有关何时允许协调存储器位的信息，请参考作业表。
W#16#060A	第十个指令字节出错(非法CPU编号)	伙伴中的基本报头结构出错。

“发送出错”		
事件类别 出错代码	事件	纠正方法
W#16#0701	<b>仅限3964(R):</b> 发送第一个重复: <ul style="list-style-type: none"> <li>发送消息帧时检测出出错</li> <li>伙伴以否定确认字符(NCC)请求了重复。</li> </ul>	重复不代表出错。但是,可能表示数据链接中断或伙伴的出错操作。如果在允许的最大重复次数内未传送消息帧,会报告一个出错编号,说明首次发生的出错。
W#16#0702	<b>仅限3964(R):</b> 建立连接时出错 传送STX后,收到NCC或其它字符(除DLE或STX外)。	如有必要,可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0703	<b>仅限3964(R):</b> 超出确认时间(QVZ): 传送STX后,伙伴在确认延迟时间内没有响应。	伙伴设备太慢或尚未作好接收准备,或数据链路已中断。如有必要,可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0704	<b>仅限3964(R):</b> 由伙伴中止: 发送操作正在进行时,从伙伴中收到一个或多个字符。	检查伙伴是否因以下原因也指示一项出错:尚未收到所有已传送数据(例如,由于数据链路中断)、未决致命出错、或伙伴设备的出错行为。如有必要,可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0705	<b>仅限3964(R):</b> 传送正在进行时收到否定确认	检查伙伴是否因以下原因也指示一个出错:尚未收到所有已传送数据(例如,由于数据链路中断)、未决致命出错、或伙伴设备的出错行为。如有必要,可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0706	<b>仅限3964(R):</b> 传送结束出错: <ul style="list-style-type: none"> <li>伙伴最终以NCC或任何字符(除DLE外)拒绝了消息帧</li> <li>确认字符(DLE)接收过早。</li> </ul>	检查伙伴是否因以下原因也指示一个出错:尚未收到所有已传送数据(例如,由于数据链路中断)、未决致命出错、或伙伴设备的出错行为。如有必要,可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0707	<b>仅限3964(R):</b> 超出传送结束确认延迟时间/响应监视狗时间: 连接断开后,伙伴未在QVZ内以DLE ETX发出响应。	伙伴设备太慢或中断。如有必要,可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0708	<b>仅限ASCII驱动程序:</b> 超出XON的等待时间。	通讯伙伴中断、太慢或切换到离线。如有必要,检查通讯伙伴或改变参数。

“发送出错”		
事件类别 出错代码	事件	纠正方法
W#16#0709	<b>仅限3964(R):</b> 无法建立连接，超出了允许的启动尝试次数。	检查接口线或传送参数。 还要在伙伴方检查CPU和CP间接收功能的参数是否已正确分配。
W#16#070A	<b>仅限3964(R):</b> 无法传送数据，超出了允许的传送尝试次数。	检查接口线或传送参数。
W#16#070B	<b>仅限3964(R):</b> 由于两个伙伴都设置为高优先级，无法解决初始化冲突。	改变参数。
W#16#070C	<b>仅限3964(R):</b> 由于两个伙伴都设置为低优先级，无法解决初始化冲突。	改变参数。
“接收出错”		
事件类别 出错代码	事件	纠正方法
W#16#0801	<b>仅限3964(R):</b> 等待第一个重复： 收到消息帧时，检测到出错，并且CPU以否定确认(NCC)向伙伴请求重复。	重复不代表出错。但是，可能表示数据链路中断或伙伴的出错行为。如果在允许的最大重复次数内未传送消息帧，会报告一个出错编号，说明首次发生的出错。
W#16#0802	<b>仅限3964(R):</b> 建立连接时出错 <ul style="list-style-type: none"> <li>• 空闲时收到一个或多个字符 (除NCC或STX外)</li> <li>• 收到STX之后，伙伴未等待DLE响应即发送了其它字符。</li> </ul> 伙伴设备启动后： <ul style="list-style-type: none"> <li>• 启动伙伴设备后，CPU收到未定义字符。</li> </ul>	如有必要，可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0805	<b>仅限3964(R):</b> 逻辑接收出错： 接收DLE之后，又收到了另一个字符 (除DLE和ETX外)。	检查伙伴是否在消息帧报头和数据字符串中加入两个DLE，或是否通过DLE ETX建立了连接。如有必要，可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。



“接收出错”		
事件类别 出错代码	事件	纠正方法
W#16#0806	超出字符延迟时间(CDT): <ul style="list-style-type: none"> <li>在CDT内未收到两个后续字符。</li> </ul> 仅限3964(R): <ul style="list-style-type: none"> <li>1. 建立连接后发送DLE时, 在CDT内未收到字符。</li> </ul>	伙伴设备太慢或中断。如有必要, 可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0807	消息帧长度非法: 收到消息帧的长度为0。	收到长度为0的消息帧并不代表出错。检查通讯伙伴发送的消息帧中为何没有用户数据。
W#16#0808	仅限3964(R): 块检查字符出错BCC: 内部生成的BCC值与伙伴在通讯链接结束时收到的BCC不一致。	检查通讯是否严重中断。这种情况下, 还可以见到偶发出错代码。如有必要, 可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0809	仅限3964(R): 块重复的延迟时间已到期	在通讯伙伴和用户模块的参数中声明相同的块延迟时间。如有必要, 可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#080A	没有空闲输入缓冲区: 没有可用于接收的已清空的输入缓冲区。	必须增加调用SFB RCV的频率。
W#16#080C	传送出错 <ul style="list-style-type: none"> <li>检测到传送出错(奇偶校验/停止位/溢出出错)。</li> </ul> 仅限3964(R): <ul style="list-style-type: none"> <li>如果在空闲模式收到中断的字符, 则立即报告出错, 以便尽快发现对数据链路的干扰影响。</li> </ul> 仅限3964(R): <ul style="list-style-type: none"> <li>如果这种出错在发送和接收操作期间发生, 则启动重复。</li> </ul>	对数据链路的干扰会引起消息帧的重复, 并因此降低用户数据的吞吐量。不检测出错的危险也会增加。改变系统结构或连线。 检查通讯伙伴的数据链路或双方设备上设置的波特率、奇偶校验和停止位是否相同。

“接收出错”		
事件类别 出错代码	事件	纠正方法
W#16#080D	中断:	再次建立链接或启动伙伴设备。
W#16#080E	未启用数据流控制时, 输入缓冲区溢出。	必须在用户程序中更为频繁地调用用于接收的SFB, 或必须为通讯分配数据流控制参数。
W#16#0810	校验出错	检查通讯伙伴的数据链路或双方设备上设置的波特率、奇偶校验和停止位是否相同。
W#16#0811	字符帧出错	检查通讯伙伴的数据链路或双方设备上设置的波特率、奇偶校验和停止位是否相同。 改变系统结构或连线。
“接收出错”		
事件类别 出错代码	事件	纠正方法
W#16#0812	仅限ASCII驱动程序: CPU发送XOFF后, 又接收到其它字符。	重新分配通讯伙伴的参数或加快处理数据。
W#16#0814	仅限ASCII驱动程序: 由于未使用数据流控制执行操作, 丢失了一个或多个消息帧。	尽可能使用数据流控制操作。利用全部输入缓冲区。在基本参数中, 将“对CPU STOP模式的反应”参数设置为“继续运行”。
W#16#0816	收到的消息帧长度超出指定的最大长度。	需要在伙伴站中更正。
“从通讯伙伴收到具有出错的响应消息帧或出错消息帧”		
事件类别 出错代码	事件	纠正方法
W#16#0902	<b>仅限RK 512:</b> 伙伴站的存储器访问出错(无存储器) 伙伴设备为SIMATIC S5时: <ul style="list-style-type: none"> <li>• 显示字范围出错</li> <li>• 无数据范围(DB/DX除外)</li> <li>• 数据范围不足(DB/DX除外)</li> </ul>	检查伙伴是否配备了所需的数据范围、该数据范围是否有足够大小, 或检查所调用SFB的参数。 检查在SFB中指定的长度。
W#16#0903	<b>仅限RK 512:</b> 伙伴站中的DB/DX访问出错(DB/DX不存在或过短) 伙伴设备为SIMATIC S5时: <ul style="list-style-type: none"> <li>• DB/DX不存在</li> <li>• DB/DX过短</li> <li>• DB/DX编号非法</li> </ul> FETCH作业超出了允许的源范围。	检查伙伴是否配备了所需的数据范围、该数据范围是否有足够大小, 或检查所调用SFB的参数。 检查在SFB中指定的长度。

“从通讯伙伴收到具有出错的响应消息帧或出错消息帧”		
事件类别 出错代码	事件	纠正方法
W#16#0904	<b>仅限RK 512:</b> 伙伴报告“不允许此作业类型”。	由于CPU从未输出系统指令，伙伴行为出错。
W#16#0905	<b>仅限RK 512:</b> 伙伴出错或SIMATIC S5伙伴出错： <ul style="list-style-type: none"> <li>• 不允许的源/目标类型</li> <li>• 伙伴设备发生存储器出错</li> <li>• 伙伴CP/CPU通讯出错</li> <li>• 伙伴PLC处于STOP模式</li> </ul>	检查伙伴是否能够传送所需的数据类型。 检查伙伴的硬件结构。 将伙伴的PLC切换到RUN模式。
W#16#0908	<b>仅限RK 512:</b> 伙伴设备发现同步出错：消息帧序列中断。	自身重新启动或重新启动伙伴的PLC时出现此出错。这是系统正常的启动操作。无需纠正。在运行期间，由于之前的出错，也可能出现此出错。否则，可以视为伙伴的出错行为。
W#16#0909	<b>仅限RK 512:</b> 伙伴设备中，DB/DX被协调存储器位锁定。	在伙伴程序中：处理完最后的发送数据后，复位协调存储器位！ 程序：重复作业！
W#16#090A	<b>仅限RK 512:</b> 伙伴发现消息帧报头出错：3. 报头中的指令字节出错	检查该出错是否由于干扰或出错的伙伴行为引起。通过挂接到数据链路的接口测试仪对此进行检查。
W#16#090C	<b>仅限RK 512:</b> 伙伴检测到消息帧长度(总长度)出错。	检查该出错是否由于干扰或出错的伙伴行为引起。通过挂接到数据链路的接口测试仪对此进行检查。
W#16#090D	<b>仅限RK 512:</b> 到目前为止，伙伴设备未进行重新启动。	重新启动伙伴PLC或将工作模式选择开关设置为RUN。
W#16#090E	<b>仅限RK 512:</b> 在响应消息帧中收到未知出错编号。	检查该出错是否由于干扰或出错的伙伴行为引起。通过挂接到数据链路的接口测试仪对此进行检查。

“CPU在伙伴的响应消息帧中检测到出错”		
事件类别 出错代码	事件	纠正方法
W#16#0A02	<b>仅限RK 512:</b> 收到的响应消息帧的结构出错 (第1字节不是00或FF)	如有必要, 可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0A03	<b>仅限RK 512:</b> 收到的响应消息帧所含数据过多或不足。	如有必要, 可通过将接口测试仪挂接至数据链路来检验伙伴的出错状况。
W#16#0A05	<b>仅限RK 512:</b> 在监视时间内, 没有来自伙伴的响应消息帧。	伙伴设备的速度很慢吗? 此出错也经常因前一出错显示。例如, 发送FETCH消息帧后会显示接收过程出错(事件类别8)。 原因: 由于干扰或监视狗超时, 无法接收响应消息帧。如果伙伴设备在响应最后收到的FETCH消息帧之前重新启动, 也可能产生此出错。
“警告”		
事件类别 出错代码	事件	纠正方法
W#16#0B01	输入缓冲区的装载量超出其容量的2/3	增加调用接收块的频率, 以避免输入缓冲区溢出。

## 27 用于H CPU的SFC

### 27.1 在H系统中使用SFC90 “H\_CTRL”控制操作

#### 描述

使用SFC90 “H\_CTRL”，可在下列方面影响H系统：

- 可防止主站CPU中的待机链接。通过SFC90 “H\_CTRL”取消此设置，或者H系统进入STOP模式时即禁用此功能。  
系统会存储所有来自待机CPU与主站链接的请求。
- 可在主站CPU上禁用更新。通过SFC90 “H\_CTRL”取消此设置，或者H系统进入STOP模式时即禁用此功能。  
系统会存储所有来自待机CPU的更新请求。



---

#### 当心

如果已经禁用更新但尚未连接，硬件系统仍可如先前一样确定连接状态。请注意，主站CPU在连接时不处理任何删除/插入中断、站故障/返回的中断或机架故障/返回的中断。

---

- 可从循环自检中删除测试组件，再次添加该组件或立即启动。如果已经禁用了循环自检中的一个或多个测试组件，使用SFC90 “H\_CTRL”取消该设置或H系统进入STOP模式前，被禁用的组件始终保持禁用状态。

#### 注意：

如果在冗余系统中使用CPU 414-4H或417-4H，请注意以下内容：如果禁用组件超过24小时，CPU会进入STOP模式。对于冗余系统，适用规则规定某些测试必须在24小时之内完成。

下表说明了输入参数MODE和SUBMODE允许的组合。

作业	MODE 输入	SUBMODE 输入
禁用链接	3	0
重新使能链接	4	0
禁用更新	1	0
重新使能更新	2	0
在循环自检中删除在SUBMODE中指定的测试组件。 测试组件只能删除一次。	20	0.1,...5
向循环自检中重新添加在SUBMODE中指定的测试组件。 测试组件只有先前已删除过才能再次添加。	21	0.1,...5
立即启动在SUBMODE中指定的测试组件。 此测试组件不能被删除。	22	0.1,...5

下表给出了使用SUBMODE输入值对循环自检的各测试组件的分配情况。(仅与输入MODE的值20、21和22相关)

SUBMODE 的值	相关测试组件
0	SP7 – ASIC – Test
1	代码存储器测试
2	数据存储器测试
3	操作系统代码检验和测试
4	代码块检验和测试
5	冗余操作中编码、时间、标记和数据块的比较。

## SFC的工作方式

SFC90 “H\_CTRL”是异步SFC，换言之，其执行可贯穿多个SFC调用。

通过用REQ=1调用SFC 90启动作业。

如果可以立即执行作业，SFC会在BUSY输出参数中返回0。即使测试包括多个周期(MODE=22, RET\_VAL=W#16#0001)，长期测试例行程序的初始化也以第一个SFC调用(BUSY=0)结束。如果BUSY的值为1，则作业仍处于活动状态(参见异步SFC的REQ、RET\_VAL和BUSY参数的含义)。

## 识别作业

作业由输入参数MODE和SUBMODE指定。如果这些参数与尚未完成的作业匹配，则SFC调用为后续调用。

## 参数

参数	声明	数据类型	存储区域	描述
REQ	INPUT	BOOL	I、Q、M、D、L 常数	电平触发的控制参数REQ=1： 触发作业
MODE	INPUT	BYTE	I、Q、M、D、L 常数	作业
SUBMODE	INPUT	WORD	I、Q、M、D、L 常数	二级作业
RET_VAL	OUTPUT	INT	I、Q、M、D、L	如果在执行功能时出错，返回值中会包含 出错代码。确保每次执行块后，都判断 RET_VAL。
BUSY	OUTPUT	BOOL	I、Q、M、D、L	BUSY=1： 作业尚未完成。

## 出错信息

出错代码(W#16#...)	说明
0000	作业已无误地执行
7000	第一次调用时REQ = 0：作业未激活；BUSY的值为0。
7001	第一次调用时REQ = 1：作业已启动；BUSY的值为1。
7002	后续调用(与REQ不相关)。激活的作业仍在运行；BUSY的值为1。
0001	<ul style="list-style-type: none"> <li>MODE=1时：已禁用更新。</li> <li>MODE=3时：已禁用链接。</li> <li>MODE=22时：测试组件已运行，不能重新启动。</li> </ul>
8082	<ul style="list-style-type: none"> <li>MODE=1时：已激活更新，不能再禁用。</li> <li>MODE=3时：已激活链接，不能再禁用。</li> <li>MODE=20时：指示的测试组件从循环自检中删除。</li> <li>MODE=21时：指示的测试组件尚未从循环自检中删除。</li> <li>MODE=22时：指示的测试组件已从循环自检中删除，无法执行。</li> </ul>
8090	MODE输入参数值无效。
8091	SUBMODE输入参数值无效。
8xyy	常规出错信息，请参见使用输出参数RET_VAL评估出错。

### SFC90使用实例

通过SFC90 “H\_CTRL”，可以确保需要CPU的最大资源时未启动链接和更新。

要实现此目的，可在进程活动增加期间前，将下列程序段加入主站CPU。

- 以MODE = 3和SUBMODE = 0调用SFC90(禁用链接)
- 以MODE = 1和SUBMODE = 0调用SFC90(禁用更新)

在活动增加期间结束时，将下列程序段加入主站CPU：

- 以MODE = 4和SUBMODE = 0调用SFC90(重新启用链接)
- 以MODE = 2和SUBMODE = 0调用SFC90(重新启用更新)。



## 28 集成功能(对于带集成I/O的CPU)

### 28.1 SFB29(HS\_COUNT)

#### 描述

通过SFB 29 “HS\_COUNT” (计数器), 可影响具有集成I/O的CPU的集成计数器功能的以下方面:

- 设置和输入起始值。
- 选择和设置比较值。
- 使能计数器。
- 使能数字输出。
- 读取当前计数值和当前比较值。
- 查询计数值和比较值之间的关系。

#### 更多信息

##### SFB29

各参数及集成功能计数器参数和CPU硬件输入与输出参数的含义, 在S7-300可编程程控制器, 集成功能手册中有详细描述。

#### 参数

参数	声明	数据类型	存储区域	描述
PRES_COUNT	INPUT	DINT	I、Q、M、D、L、 常数	计数器的起始值
PRES_COMP_A	INPUT	DINT	I、Q、M、D、L、 常数	新比较值COMP_A
PRES_COMP_B	INPUT	DINT	I、Q、M、D、L、 常数	新比较值COMP_B
EN_COUNT	INPUT		I、Q、M、D、L	使能计数器
EN_DO	INPUT	BOOL	I、Q、M、D、L、 常数	使能数字输出
SET_COUNT	INPUT	BOOL	I、Q、M、D、L、 常数	为起始值PRES_COUNT 设置输出
SET_COMP_A	INPUT	BOOL	I、Q、M、D、L、 常数	为比较值COMP_A设置输出

参数	声明	数据类型	存储区域	描述
SET_COMP_B	INPUT	BOOL	I、Q、M、D、L、 常数	为比较值COMP_B设置输出
COUNT	OUTPUT	DINT	I、Q、M、D、L	计数器的实际值
COMP_A	OUTPUT	DINT	I、Q、M、D、L	当前比较值COMP_A
COMP_B	OUTPUT	DINT	I、Q、M、D、L	当前比较值COMP_B
STATUS_A	OUTPUT	BOOL	I、Q、M、D、L	状态位STATUS_A 1: COUNT COMP_A 0: COUNT COMP_A
STATUS_B	OUTPUT	BOOL	I、Q、M、D、L	状态位STATUS_B 1: COUNT COMP_B 0: COUNT COMP_B

## 28.2 SFB30(FREQ\_MES)

### 描述

通过SFB 30 “FREQ\_MES” (频率计), 可影响具有集成I/O的CPU的集成频率计功能的以下方面:

- 选择和设置比较值。
- 输出测量的频率。
- 读取当前比较值。
- 查询测量的频率与比较值的关系。

### 更多信息

#### SFB30

各参数及集成频率计功能参数和CPU硬件输入与输出参数的含义, 在*S7-300可编程控制器, 集成功能手册*中有详细描述。

### 参数

参数	声明	数据类型	存储区域	描述
PRES_U_LIMIT	INPUT	DINT	I、Q、M、D、L、 常数	新(上限)比较值U_LIMIT
PRES_L_LIMIT	INPUT	DINT	I、Q、M、D、L、 常数	新(下限)比较值L_LIMIT
SET_U_LIMIT	INPUT	BOOL	I、Q、M、D、L、 常数	为新比较值U_LIMIT设置输入
SET_L_LIMIT	INPUT	BOOL	I、Q、M、D、L、 常数	为新比较值L_LIMIT设置输入
FREQ	OUTPUT	DINT	I、Q、M、D、L	测得的频率(以MHz计)
U_LIMIT	OUTPUT	DINT	I、Q、M、D、L	当前比较值(上限)
L_LIMIT	OUTPUT	DINT	I、Q、M、D、L	当前比较值(下限)
STATUS_U	OUTPUT	BOOL	I、Q、M、D、L	状态位 “1”: FREQ > U_LIMIT “0”: FREQ ≤ U_LIMIT
STATUS_L	OUTPUT	BOOL	I、Q、M、D、L	状态位 “1”: FREQ < L_LIMIT “0”: FREQ ≥ L_LIMIT

## 28.3 SFB38(HSC\_A\_B)

### 描述

通过SFB 38 (HSC\_A\_B)，可影响具有集成输入/输出的CPU的集成A/B计数器功能以下的方面：

- 指定和采用起始值
- 指定和设置比较值
- 使能计数器
- 使能数字输出
- 读取当前计数值和当前比较值
- 查询与比较值相关的计数值

SFB 38 (HSC\_A\_B)可在集成功能的背景数据块的用户程序中读写数据。A/B计数器由两个计数器A和B组成，这两个计数器既可同时计数又相互独立(可进行向上计数和向下计数)。

两个计数器的运行方式相同，记录的计数脉冲频率可达10千赫。

### 更多信息

SFB38 参数的确切含义及集成功能A/B计数器的参数和CPU硬件输入与输出的参数，在手册S7-300可编程控制器，集成功能CPU 312 IFM/314 IFM中有详细描述。

### 参数

参数	声明	数据类型	存储区域	描述
PRES_COMP	INPUT	DINT	I、Q、M、D、L、 常数	新比较值COMP
EN_COUNT	INPUT	BOOL	I、Q、M、D、L、 常数	使能计数器
EN	INPUT	BOOL	I、Q、M、D、L、 常数	使能数字输出
SET_COMP	INPUT	BOOL	I、Q、M、D、L、 常数	为比较值COMP设置输入
COUNT	OUTPUT	DINT	I、Q、M、D、L	计数器的实际值
COMP	OUTPUT	DINT	I、Q、M、D、L	当前比较值COMP
ENO	OUTPUT	BOOL	I、Q、M、D、L	出错处理： 1：执行中无故障 0：执行时出错

## 28.4 SFB39(POS)

### 描述

通过SFB39(POS)，可以影响具有集成输入/输出的CPU的集成定位功能。SFB39(POS)具有下列功能：

- 同步
- 执行微动模式
- 定位

集成定位功能的SFB 39 (POS)可从用户程序中读取数据并写入具有集成功能的背景数据块。集成定位功能可从非对称的24 V增量式编码器获得最高频率为10千赫的信号。通过CPU 314 IFM的指定集成输出，它可控制快速/慢速模式或频率转换器(控制定位)

### 更多信息

SFB39 参数的确切含义及集成功能A/B计数器的参数和CPU硬件输入与输出的参数，在手册S7-300可编程控制器，集成功能CPU 312 IFM/314 IFM中有详细描述。

### 参数

参数	声明	数据类型	存储区域	描述
EN	INPUT	BOOL	I、Q、M、D、L、 常数	数字输入使能
DEST_VAL	INPUT	DINT	I、Q、M、D、L、 常数	集成定位功能的目标位置
REF_VAL	INPUT	DINT	I、Q、M、D、L、 常数	同步参考点
SWITCH_OFF_DIFF	INPUT	WORD	I、Q、M、D、L、 常数	行程增量中的关闭差值 (关闭点和目标位置的差值)
PRES_COMP	INPUT	DINT	I、Q、M、D、L、 常数	新比较值COMP
BREAK	INPUT	BYTE	I、Q、M、D、L、 常数	控制遍历移动的最大模拟值
POS_MODE1、 POS_MODE2	INPUT	BOOL	I、Q、M、D、L、 常数	启动和执行微动模式
POS_STRT	INPUT	BOOL	I、Q、M、D、L、 常数	在上升沿启动定位操作
SET_POS	INPUT	BOOL	I、Q、M、D、L、 常数	出现上升沿时，输入参数REF_VAL 的值将作为新实际值

参数	声明	数据类型	存储区域	描述
ENO	OUTPUT	BOOL	I、Q、M、D、L	出错处理： 1：执行中无故障 0：执行时出错
ACTUAL_POS	OUTPUT	DINT	I、Q、M、D、L	当前实际值
POS_READY (状态消息)	OUTPUT	BOOL	I、Q、M、D、L	如果POS_READY=1，定位/微动完成
REF_VALID (状态消息)	OUTPUT	BOOL	I、Q、M、D、L	是否到达参考点开关
POS_VALID (状态消息)	OUTPUT	BOOL	I、Q、M、D、L	轴的实际位置已与集成功能的实际位置同步

## 29 塑料技术

### 29.1 SFC63 (AB\_CALL)

#### 描述

SFC63 (AB\_CALL)可调用汇编码块。  
汇编码块是用“C”语言编写或在汇编器中编写、然后编译的逻辑块。

#### 应用

只能对CPU 614使用汇编码块。

#### 更多信息

SFC63 各参数的含义在CPU 614的文档中详细说明。编写汇编码块有单独的编程指南。

#### 参数

参数	声明	数据类型	存储区域	描述
AB_NUMBER	INPUT	WORD	I、Q、M、D、L、 常数	要调用的汇编码块的位
CALL_REASON	INPUT	WORD	I、Q、M、D、L、 常数	调用SFC的组织块、对计算DB指针(参数 DB_NUMBER), 或激活调试器
DB_NUMBER	INPUT	WORD	I、Q、M、D、L、 常数	DB指针的编号
RET_VAL	OUTPUT	WORD	I、Q、M、D、L	SFC的返回值





## 30 诊断数据

### 30.1 诊断数据结构概述

#### 系统数据的数据记录0和1

模块的诊断数据位于系统数据区的数据记录0和1中(参见读/写一条数据记录)。

- 数据记录0包含4字节的诊断数据，用来描述信号模块的当前状态。
- 数据记录1包含
  - 4字节的诊断数据(也在数据记录0中)和
  - 模块专用的诊断数据。

#### 诊断数据的结构和内容

该部分描述诊断数据各字节的结构和内容。

出错时，相应位将置为“1”。

## 30.2 诊断数据

诊断数据的结构和内容:

字节	位	含义	备注	数据记录	
0	0	模块故障		0和1	
	1	内部出错			
	2	外部出错			
	3	通道出错			
	4	无外部辅助电压			
	5	无前连接器			
	6	无参数分配			
	7	模块参数出错			
1	0 至 3	模块等级	0101: 模拟模块 0000: CPU 1000: 功能模块 1100: CP 1111: 数字模块 0011: DP标准从站 1011: I从站 0100: IM	0和1	
	4	存在通道信息			
	5	存在用户信息			
	6	来自替换项的诊断中断			
	7	保留			
	2	0	无存储卡或存储卡出错		0和1
		1	通讯问题		
		2	模式	0: RUN 1: STOP	
3		响应周期监视			
4		内部模块电源电压故障			
5		电池耗尽			
6		整个备用电池故障			
7		保留			
3	0	扩展机架故障		0和1	
	1	处理器故障			
	2	EPROM出错			
	3	RAM出错			
	4	ADC/DAC出错			
	5	保险丝断开			
	6	硬件中断丢失			
	7	保留			

字节	位	含义	备注	数据记录
4	0 至 6	通道类型	B#16#70: 数字量输入 B#16#72: 数字量输出 B#16#71: 模拟量输入 B#16#73: 模拟量输出 B#16#74: FM-POS B#16#75: FM-REG B#16#76: FM-ZAEHL B#16#77: FM-TECHNO B#16#78: FM-NCU B#16#79至 B#16#7D: 保留 B#16#7E: US300 B#16#7F: 保留	1
	7	是否还有其它通道类型?	0: 否 1: 是	
5	0 至 7	模块每个通道的诊断位数输出。	每个通道的诊断位数向上取整到字节边界	1
6	0 至 7	模块每种通道类型的通道数	如果一个模块上存在不同的通道类型, 则对于每种通道类型, 在数据记录1 中从字节4开始向前 重复这一结构。	1
7	0	通道出错通道0/通道组0	通道出错向量的第一字节(通道出错向量的长度取决于通道数并向上取整到字节边界。)	1
	1	通道出错通道1/ 通道组1		
	2	通道出错通道2/ 通道组2		
	3	通道出错通道3/ 通道组3		
	4	通道出错通道4/ 通道组4		
	5	通道出错通道5/通道组5 通道组5		
	6	通道出错通道6/ 通道组6		
	7	通道出错通道7/ 通道组7		
...	-	通道专有出错(参见通道专有诊断数据的结构)		1

### 30.3 通道专有诊断数据的结构

#### 通道专有出错

从紧随通道出错向量的字节开始，指示模块每个通道的通道专有出错。下表显示不同通道类型的通道专有诊断数据的结构。各个位的含义如下：

- 1 = 出错
- 0 = 无错

#### 模拟量输入通道

模拟量输入通道的诊断字节：

位	含义	备注
0	组态/参数分配出错	可通过SFC52 和EVENTN = W#16#8x50发出信号
1	共模出错	可通过SFC52 和EVENTN = W#16#8x51发出信号
2	P短路	可通过SFC52 和EVENTN = W#16#8x52发出信号
3	M短路	可通过SFC52 和EVENTN = W#16#8x53发出信号
4	断线	可通过SFC52 和EVENTN = W#16#8x54发出信号
5	参考通道出错	可通过SFC52 和EVENTN = W#16#8x55发出信号
6	电流低于测量范围	可通过SFC52 和EVENTN = W#16#8x56发出信号
7	电流高于测量范围	可通过SFC52 和EVENTN = W#16#8x57发出信号

#### 模拟量输出通道

模拟量输出通道的诊断字节：

位	含义	备注
0	组态/参数分配出错	可通过SFC52 和EVENTN = W#16#8x60发出信号
1	共模出错	可通过SFC52 和EVENTN = W#16#8x61发出信号
2	P短路	可通过SFC52 和EVENTN = W#16#8x62发出信号
3	M短路	可通过SFC52 和EVENTN = W#16#8x63发出信号
4	断线	可通过SFC52 和EVENTN = W#16#8x64发出信号
5	0	保留
6	空载电压	可通过SFC52 和EVENTN = W#16#8x66发出信号
7	0	保留

## 数字量输入通道

数字量输入通道的诊断字节：

位	含义	备注
0	组态/参数分配出错	可通过SFC52 和EVENTN = W#16#8x70发出信号
1	接地出错	可通过SFC52 和EVENTN = W#16#8x71发出信号
2	P短路(传感器)	可通过SFC52 和EVENTN = W#16#8x72发出信号
3	M短路	可通过SFC52 和EVENTN = W#16#8x73发出信号
4	断线	可通过SFC52 和EVENTN = W#16#8x74发出信号
5	无传感器电源	可通过SFC52 和EVENTN = W#16#8x75发出信号
6	0	保留
7	0	保留

## 数字量输出通道

数字量输出通道的诊断字节：

位	含义	备注
0	组态/参数分配出错	可通过SFC52 和EVENTN = W#16#8x80发出信号
1	接地出错	可通过SFC52 和EVENTN = W#16#8x81发出信号
2	P短路	可通过SFC52 和EVENTN = W#16#8x82发出信号
3	M短路	可通过SFC52 和EVENTN = W#16#8x83发出信号
4	断线	可通过SFC52 和EVENTN = W#16#8x84发出信号
5	保险丝断开	可通过SFC52 和EVENTN = W#16#8x86发出信号
6	空载电压	可通过SFC52 和EVENTN = W#16#8x86发出信号
7	过热	可通过SFC52 和EVENTN = W#16#8x87发出信号



## 31 系统状态列表(SSL)

### 31.1 系统状态列表(SSL)概述

本章描述与下列各项相关的系统状态列表的所有部分列表：

- CPU
- 模块，其部分列表并非模块特有。(例如SSL-ID W#16#00B1、W#16#00B2、W#16#00B3)。

模块特有的部分列表，例如CP和FM的部分列表，将在特定模块的描述中介绍。

#### 定义：系统状态列表

系统状态列表(SSL)用于描述可编程逻辑控制器的当前状态。SSL的内容只能通过信息功能进行读取，而不能修改。换言之，部分列表是虚拟列表，只是在有特殊请求时由CPU的操作系统所创建。

只能通过SFC51 “RDSYSST”读取一个系统状态列表。

#### 内容

系统状态列表包含下列内容的有关信息：

- 系统数据
- CPU中的模块状态数据
- 模块的诊断数据
- 诊断缓冲区

#### 系统数据

系统数据是CPU固有的或者已分配的特征数据。它们提供了以下内容的有关信息：

- CPU的组态
- 优先级的状态
- 通讯

### **模块状态数据**

模块状态数据描述由系统诊断功能监视的组件的当前状态。

### **模块的诊断数据**

分配给CPU的具有诊断能力的模块的诊断数据直接存储在模块上。

### **诊断缓冲区**

诊断缓冲区包含的诊断条目按出现的顺序排列。



## 31.2 部分SSL列表的结构

### 基础

可以用SFC51 “RDSYSST”读取部分列表和部分列表的摘录。可使用参数SSL\_ID和INDEX指定要读取的内容。

### 结构

部分列表由下列各项组成：

- 一个报头和
- 数据记录。

### 文件头

部分列表的报头由下列各项组成：

- SSL-ID
- 索引
- 部分列表数据记录的长度(以字节计)
- 部分列表中包含的数据记录数目。

### 索引

对于某些部分列表或部分列表摘录，必须指定对象类型ID或对象编号。索引即用于此目的。如果不需要该部分信息，则其内容是不相关的。

### 数据记录

部分列表中的数据记录有规定长度。这取决于部分列表中的信息。如何使用数据记录中的数据字也视特定的部分列表而定。

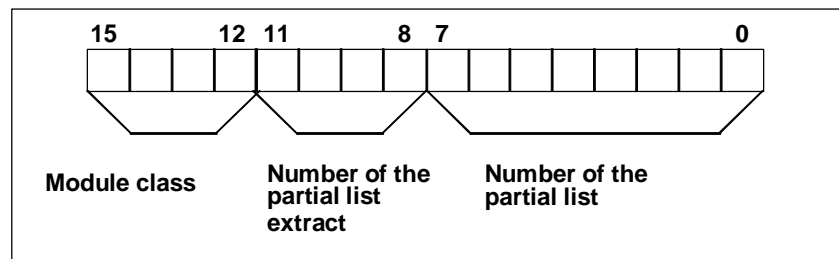
### 31.3 SSL-ID

#### SSL-ID

每个部分系统状态列表都有一个编号。可以根据编号输出完整的部分列表或摘录。预定义了可能的部分列表摘录，并由一个数字标识。**SSL-ID**由部分列表的编号、部分列表摘录的编号和模块等级组成。

#### 结构

SSL-ID为一个字长。SSL-ID中的各个位的含义如下所示：



SCL\_ID的结构

#### 模块等级

模块等级实例：classes:

模块等级	编码(二进制)
CPU	0000
IM	0100
FM	1000
CP	1100

#### 部分列表摘录的编号

部分列表摘录的编号及其含义取决于其所属的特定系统状态列表。通过部分列表摘录的编号，可以指定要读取部分列表的哪一个子集。

#### 部分列表的编号

使用部分列表的编号，可以指定要读取系统状态列表的哪一个部分列表。

## 31.4 可能的部分系统状态列表

### 子集

任何一个模块都仅具有所有可能的部分列表的一个子集。哪些部分列表可用取决于特定的模块。

### 可能的SSL部分列表

下表列出了所有可能的部分列表及其包含在SSL-ID中的编号。

部分列表	SSL-ID
模块标识	W#16#xy11
CPU特征	W#16#xy12
用户存储器区域	W#16#xy13
系统区域	W#16#xy14
块类型	W#16#xy15
模块LED的状态	W#16#xy19
过程映像分区和OB之间的分配	W#16#xy25
通讯状态数据	W#16#xy32
H CPU组信息	W#16#xy71
模块LED的状态	W#16#xy74
在H系统中切换的DP从站	W#16#xy75
模块状态信息	W#16#xy91
机架/站的状态信息	W#16#xy92
扩充的DP主站系统信息	W#16#xy95
CPU的诊断缓冲区	W#16#xyA0
模块诊断信息(数据记录0)	W#16#00B1
模块诊断信息(数据记录1)、地理地址	W#16#00B2
模块诊断信息(数据记录1)、逻辑地址	W#16#00B3
DP从站的诊断数据	W#16#00B4

## 31.5 SSL-ID W#16#xy11 - 模块标识

### 用途

如果读取SSL-ID为W#16#xy11的系统状态列表，则可获得该模块的模块标识。

### 文件头

系统状态列表SSL-ID W#16#xy11的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID W#16#0111: 单个标识数据记录
INDEX	特定数据记录的编号 W#16#0001: 模块的标识 W#16#0006: 基本硬件的标识 W#16#0007: 基本固化程序的标识
LENTHDR	W#16#001C: 一个数据记录为14个字(28字节)
N_DR	数据记录的数目

### 数据记录

系统状态列表SSL-ID W#16#xy11的数据记录具有如下结构：

名称	长度	含义
索引	1个字	标识数据记录的索引
MIFB	20字节	索引W#16#0007: 保留 索引W#16#0001和W#16#0006: 模块的订货号；字符串由19个字符和一个空格(20H)组成；例如，对于CPU 314为：“6ES7314-0AE01-0AB0”
BGTyp	1个字	保留
Ausbg1	1个字	索引W#16#0001: 模块的版本 索引W#16#0006和W#16#0007: “V”和版本ID的第一个数字
Ausbg2	1个字	索引W#16#0001: 保留 索引W#16#0006和W#16#0007: 版本ID的其余 数字

## 31.6 SSL-ID W#16#xy12 - CPU特征

### 用途

根据所使用的硬件，CPU模块具有不同的特征。将为每个特征分配一个ID。如果读取SSL-ID为W#16#xy12的部分列表，则可获得该模块的特征。

### 文件头

部分列表SSL-ID W#16#xy12的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID： W#16#0012：所有特征 W#16#0112：组的特征 可在INDEX参数中指定组。 W#16#0F12：部分列表报头信息
INDEX	组 W#16#0000：MC7处理单元 W#16#0100：时间系统 W#16#0200：系统响应 W#16#0300：CPU的MC7语言描述 W#16#0400：SFC的可用性
LENTHDR	W#16#0002：一个数据记录为1个字长(2字节)
N_DR	数据记录的数目

### 数据记录

部分列表SSL-ID W#16#xy12的数据记录为一个字长。为每个特征都输入了一个标识符。一个特征标识符为一个字长。

### 特征标识符

下表列出了所有的特征标识符。

识别	含义
W#16#0000 - 00FF	MC7处理单元(具有索引0000的组)
W#16#0001	MC7处理生成代码
W#16#0002	MC7翻译器
W#16#0100 - 01FF	时间系统(具有索引0100的组)
W#16#0101	1ms的分辨率
W#16#0102	10ms的分辨率
W#16#0103	无实时时钟

识别	含义
W#16#0104	BCD日时钟格式
W#16#0105	所有时间功能(设置日时钟、设置和读取日时钟、日时钟同步; 从日时钟和主日时钟)
W#16#0200 - 02FF	系统响应(具有索引0200的组)
W#16#0201	可使用多处理器模式
W#16#202	可进行冷重启、暖重启和热重启
W#16#203	可进行冷重启和热重启
W#16#204	可进行暖重启和冷重启
W#16#205	只能进行暖重启
W#16#0206	通过使用预定义资源可在运行期间进行新的分布式I/O组态
W#16#0207	处于单机模式的H-CPU: 通过使用预定义资源可在运行期间进行新的分布式I/O组态
W#16#0300 - 03FF	CPU的MC7语言描述(具有索引0300的组)
W#16#0301	保留
W#16#0302	所有32位的定点指令
W#16#0303	所有浮点型指令
W#16#0304	sin、asin、cos、acos、tan、atan、sqr、sqrt、ln、exp
W#16#0305	具有相应指令(ENT、PUSH、POP、LEAVE)的累加器3/累加器4
W#16#0306	主控继电器指令
W#16#0307	具有相应指令的地址寄存器1已存在
W#16#0308	具有相应指令的地址寄存器2已存在
W#16#0309	区域交叉寻址操作
W#16#030A	区域内寻址操作
W#16#030B	位存储器(M)的所有存储器间接寻址指令
W#16#030C	数据块(DB)的所有存储器间接寻址指令
W#16#030D	数据块(DI)的所有存储器间接寻址指令
W#16#030E	本地数据(L)的所有存储器间接寻址指令
W#16#030F	FC中用于参数传送的所有指令
W#16#0310	过程映像输入(I)的存储器位边沿指令
W#16#0311	过程映像输出(Q)的存储器位边沿指令
W#16#0312	位存储器(M)的存储器位边沿指令
W#16#0313	数据块(DB)的存储器位边沿指令
W#16#0314	数据块(DI)的存储器位边沿指令
W#16#0315	本地数据(L)的存储器位边沿指令
W#16#0316	FC位的动态判断
W#16#0317	具有相应指令的动态本地数据区
W#16#0318	保留
W#16#0319	保留
W#16#0401	SFC87 "C_DIAG"可用
W#16#0402	SFC88 "C_CNTRL"可用

## 31.7 SSL-ID W#16#xy13 - 存储区域

### 用途

如果读取SSL-ID为W#16#xy13的部分列表，则可获得有关该模块存储区域的信息。

### 文件头

部分列表SSL-ID W#16#xy13的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID W#16#0113: 一个存储区域的数据记录 可使用INDEX参数指定存储区域。
INDEX	指定一个存储区域(仅针对SSL-ID W#16#0113) W#16#0001: 工作存储器
LENTHDR	W#16#0024: 一个数据记录为18个字长(36字节)
N_DR	数据记录的数目

### 数据记录

部分列表SSL-ID W#16#xy13的数据记录具有如下结构：

名称	长度	含义
索引	1个字	存储区域的索引 W#16#0001: 工作存储器
代码	1个字	存储器类型: W#16#0001: 易失性存储器(RAM) W#16#0002: 非易失性存储器(FEPROM) W#16#0003: 混合性存储器(RAM + FEPROM)
长度	2个字	所选存储器的总容量(区域1和区域2的合计容量)
模式	1个字	存储器的逻辑模式 位0: 易失性存储区域 位1: 非易失性存储区域 位2: 混合性存储区域 对于工作存储器: 位3: 代码和数据分开 位4: 代码和数据在一起
Granu	1个字	值始终为0
Ber1	2个字	易失性存储区域的大小(以字节计)。
Belegt1	2个字	正在使用的易失性存储区域的大小
Block1	2个字	易失性存储区域中的最大空闲块 如果为0: 无信息可用或者无法确定。
Ber2	2个字	非易失性存储区域的大小(以字节计)
Belegt2	2个字	正在使用的非易失性存储区域的大小
Block2	2个字	非易失性存储区域中的最大空闲块 如果为0: 无信息可用或者无法确定。

## 31.8 SSL-ID W#16#xy14 - 系统区

### 用途

如果读取SSL-ID为W#16#xy14的部分列表，则可获得有关该模块的系统区域信息。

### 文件头

部分列表SSL-ID W#16#xy14的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID W#16#0014: 模块的所有系统区域 W#16#0F14: 仅用于部分列表报头信息
INDEX	不相关
LENTHDR	W#16#0008: 一个数据记录为4个字长(8字节)
N_DR	数据记录的数目 分配的数据记录数目必须至少为9个。如果选择的目标区域过小，则SFC51将不会提供数据记录。

### 数据记录

部分列表SSL-ID W#16#xy14的数据记录的结构如下：

名称	长度	含义
索引	1个字	系统区域的索引 W#16#0001: PII(以字节计) W#16#0002: PIQ(以字节计) W#16#0003: 存储器(以字节为容量单位) <b>注意:</b> 此索引仅由CPU提供，其中，标记数可在一个字内显示。如果您的CPU不提供此值，则必须估计索引W#16#0008的值。 W#16#0004: 定时器(数量) W#16#0005: 计数器(数量) W#16#0006: 逻辑地址区域中的字节数 W#16#0007: 本地数据(CPU的整个本地数据区(以字节为单位)) <b>注意:</b> 此索引仅由CPU提供，其中，标记数可在一个字内显示。如果您的CPU不提供此值，则必须估计索引W#16#0009的值。 W#16#0008: 存储器(以字节计) W#16#0009: 本地数据(CPU的整个本地数据区(以千字节计))



名称	长度	含义
代码	1个字	存储器类型 W#16#0001: 易失性存储器(RAM) W#16#0002: 非易失性存储器(FEPROM) W#16#0003: 混合性存储器(RAM和FEPROM)
数量	1个字	系统区域的元素数
Reman	1个字	掉电保护元素数

## 31.9 SSL-ID W#16#xy15 - 块类型

### 用途

如果读取SSL-ID为W#16#xy15的部分列表，则可获得该模块上存在的块类型。

### 文件头

部分列表SSL-ID W#16#xy15的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID W#16#0015: 模块的所有块类型的数据记录
INDEX	不相关
LENTHDR	W#16#0006: 一个数据记录为5个字长(10字节)
N_DR	数据记录的数目

### 数据记录

部分列表SSL-ID W#16#xy15的数据记录的结构如下：

名称	长度	含义
索引	1个字	块类型编号 W#16#0800: OB W#16#0A00: DB W#16#0B00: SDB W#16#0C00: FC W#16#0E00: FB
MaxAnz	1个字	以下类型的块的最大数目 OB: 一个CPU可能具有的最大OB数 DB: DB可能的最大数目(包括DB0) SDB: SDB可能的最大数目(包括 SDB2) FC和FB: 可装载块可能的最大数目
MaxLng	1个字	待装载对象的总量最大值(以千字节计)
Maxabl	2个字	一个块的工作存储器部分的最大长度(以字节为单位)

## 31.10 SSL-ID W#16#xy19 - 模块LED的状态

### 用途

如果读取SSL-ID为W#16#xy19的部分列表，则可获得模块LED的状态。

### 注意

如果要读取一个H CPU的部分列表W#16#16#xy19，请记住这只在非冗余的H工作模式中才能实现。

### 文件头

部分列表W#16#xy19的报头结构如下所示：

内容	含义
SSL-ID	部分系统状态列表的SSL-ID W#16#0019 所有LED的状态 W#16#0119 一个LED的状态
LENTHDR	W#16#0004: 一个数据记录为2个字长(4字节)
N_DR	数据记录的数目

### 数据记录

SSL-ID为W#16#xy19的部分列表的数据记录具有如下结构：

名称	长度	含义
索引	1个字	LED标识符(仅与SSL-ID W#16#0119相关) W#16#0001: SF(组出错) W#16#0002: INTF(内部出错) W#16#0003: EXTF(外部出错) W#16#0004: RUN W#16#0005: STOP W#16#0006: FRCE(强制) W#16#0007: CRST(重启动) W#16#0008: BAF(总线上的电池故障/超载、电池电压短路) W#16#0009: USR(用户自定义) W#16#000A: USR1(用户自定义) W#16#000B: BUS1F(总线出错接口1) W#16#000C: BUS2F(总线出错接口2) W#16#000D: REDF(冗余出错) W#16#000E: MSTR(主站) W#16#000F: RACK0(机架号0) W#16#0010: RACK1(机架号1) W#16#0011: RACK2(机架号2) W#16#0012: IFM1F(接口出错接口模块1) W#16#0013: IFM2F(接口出错接口模块2)

## 系统状态列表(SSL)

---

名称	长度	含义
led_on	1字节	LED的状态： 0: 关闭 1: 打开
led_blink	1字节	LED的闪烁状态： 0: 不闪烁 1: 正常闪烁(2赫兹) 2: 缓慢闪烁(0.5赫兹)

## 31.11 SSL-ID W#16#xy1C - 组件标识

### 用途

如果读取SSL-ID为W#16#xy1C的部分列表，则可识别CPU或PLC。

### 文件头

部分列表W#16#xy1C的报头结构如下所示：

内容	含义	
SSL-ID	部分列表摘录的SSL-ID	
	W#16#001C:	所有组件的标识
	W#16#011C:	一个组件的标识
	W#16#0F1C:	仅限SSL部分列表报头信息
INDEX	SSL ID为W#16#011C的部分系统状态列表的组件标识	
	W#16#0001:	PLC的名称
	W#16#0002:	模块名称
	W#16#0003:	模块的设备标识
	W#16#0004:	版权条目
	W#16#0005:	模块的序列号
	W#16#0006:	为操作系统保留
	W#16#0007:	模块类型名称
	W#16#0008:	存储卡的序列号 对无法接收存储卡的模块将不提供数据记录。
	W#16#0009:	CPU模块的制造商和配置文件
	W#16#000A:	模块的OEM ID
	W#16#000B:	模块的位置标识
LENTHDR	W#16#0022:	一个数据记录为17个字长(34字节)
N_DR	数据记录的数目	

## 数据记录

SSL-ID为W#16#xy1C的部分列表的数据记录具有如下结构：

- INDEX = W#16#0001

名称	长度	含义
索引	1个字	组件标识 W#16#0001
名称	12个字	PLC的名称 (最多24个字符；当使用短名称时，则用B#16#00填补间隙)
Res	4个字	保留

- INDEX = W#16#0002

名称	长度	含义
索引	1个字	组件标识 W#16#0002
名称	12个字	模块名称 (最多24个字符；当使用短名称时，则用B#16#00填补间隙)
Res	4个字	保留

- INDEX = W#16#0003

名称	长度	含义
索引	1个字	组件标识 W#16#0003
Tag	16个字	模块的设备标识 (最多32个字符；当使用短设备标识时，则用B#16#00填补间隙)

- INDEX = W#16#0004

名称	长度	含义
索引	1个字	组件标识 W#16#0004
版权所有	13个字	连续的字符序列 “西门子原厂设备”
Res	3字	保留

- INDEX = W#16#0005

名称	长度	含义
索引	1个字	组件ID: W#16#0005
serialn	12个字	模块的序列号；最大长度为24个字符的字符串。短编号用B#16#00填补。 注意：对于SIMATIC组件，此序列号全球唯一且与CPU硬件永久相关，即该序列号在执行固化程序更新后保持不变。
res	4个字	保留

- INDEX = W#16#0006

为操作系统保留相应的数据记录。

- INDEX = W#16#0007

名称	长度	含义
索引	1个字	组件ID: W#16#0007
Cputypname	16个字	模块类型名称; 最大长度为32个字符的字符串。短名称用B#16#00填补。

- INDEX = W#16#0008

名称	长度	含义
索引	1个字	组件ID: W#16#0008
sn_mc/mmc	16个字	存储卡/微存储卡的序列号; 最大长度为32个字符的字符串。短名称用B#16#00填补。 <ul style="list-style-type: none"> <li>• 西门子序列号: 仅序列号, 无索引</li> <li>• S7微存储卡的产品序列号(PSN): “MMC”加序列号(PSN)</li> <li>• S7存储卡的序列号: “MC”加序列号</li> </ul> 如果未安装任何存储卡, 则字符串在“MMC”或“MC”后即结束。

- INDEX = W#16#0009

名称	长度	含义
索引	1个字	组件ID: W#16#0009
manufacturer_id	1个字	请参见“PROFIBUS配置文件指南, 第1部分 - 标识与维护功能”
profile_id	1个字	请参见“PROFIBUS配置文件指南, 第1部分 - 标识与维护功能”
profile_specific_typ	1个字	请参见“PROFIBUS配置文件指南, 第1部分 - 标识与维护功能”
res	13个字	保留

- INDEX = W#16#000A

名称	长度	含义
索引	1个字	组件ID: W#16#000A
oem_copyright_string	13个字	OEM版权ID; 最大长度为20个字符的字符串。短名称用B#16#00填补。
oem_id	1个字	OEM ID; 由西门子签发。
oem_add_id	2个字	OEM附加ID; 可由用户签发。

- INDEX = W#16#000B

名称	长度	含义
索引	1个字	组件ID: W#16#000B
loc_id	16个字	位置标识; 最大长度为32个字符的字符串。短名称用B#16#00填补。

## 31.12 SSL ID W#16#xy25 - 将过程映像分区分配到OB

### 用途

SSL ID为W#16#xy25的部分列表给出了将过程映像分区分配到OB的方法。

此列表提供有关下列内容的信息

- 已分配给指定OB的过程映像分区通过系统进行更新
- 已分配给指定时钟同步中断OB(OB 61到64)的过程映像分区通过调用SFC 126 “SYNC\_PI” 和127 “SYNC\_PO”更新。  
DP主站系统和时钟同步中断OB之间的分配可在SSL W#16#xy95中找到。

### 文件头

SSL ID为W#16#xy25的部分列表的报头具有如下结构：

内容	含义
SSL ID	部分列表摘录的SSL ID <ul style="list-style-type: none"> <li>• W#16#0025: 将过程映像分区分配给CPU中的OB</li> <li>• W#16#0125: 将过程映像分区分配给相应的OB 在参数INDEX中指定过程映像分区ID。</li> <li>• W#16#0225: 将OB分配给过程映像分区 在参数INDEX中指定OB编号。 注意：时钟同步中断OB(OB 61到64)是可以分配给多个过程映像分区的唯一OB。</li> <li>• W#16#0F25: 仅限SSL部分列表报头的信息</li> </ul>
INDEX	<ul style="list-style-type: none"> <li>• 对于SSL ID W#16#0025: 不相关</li> <li>• 对于SSL ID W#16#0125: 过程映像分区ID</li> <li>• 对于SSL ID W#16#0225: OB编号。</li> <li>• 对于SSL ID W#16#0F25: 不相关</li> </ul>
LENTHDR	W#16#0004: 一个数据记录为2个字长(4字节)
N_DR	数据记录的数目



## 数据记录

部分列表摘录SSL-ID W#16#xy25的数据记录具有如下结构:

名称	长度	含义
tpa_nr	1字节	部分过程映像ID
tpa_use	1字节	过程映像分区和OB之间的分配类型: <ul style="list-style-type: none"> <li>• 位0 = 1: 输入的过程映像分区被分配给指定的OB, 并由系统进行更新。</li> <li>• 位1 = 1: 输出的过程映像分区被分配给指定的OB, 并由系统进行更新。</li> <li>• 位2 = 1: 输入的过程映像分区将被分配给指定的时钟同步中断OB。可通过调用SFC126 "SYNC_PI"在此OB中更新过程映像分区。</li> <li>• 位3 = 1: 输出的过程映像分区被分配给指定的时钟同步中断OB。可通过调用SFC126 "SYNC_PO"在此OB中更新过程映像分区。</li> <li>• 位4至7: 0</li> </ul>
ob_nr	1字节	OB编号
res	1字节	保留

## 部分列表摘录

- **SSL ID = W#16#0025的部分列表摘录:**  
在组态中已分配给一个OB的所有过程映像分区的数据记录将会按升序返回。对于未分配给任何OB的过程映像分区, 此ob\_nr参数值为零。这种情况下, 将不会为过程映像分区返回数据记录。
- **SSL ID = W#16#0125的部分列表摘录:**  
如果在组态中已将寻址的过程映像分区分配给一个OB, 则会返回数据记录。如果尚未分配OB, 则不会返回任何数据记录。

### 注意

OB 1被永久分配给过程映像分区0。因此, 请求有关过程映像分区0的信息时, 总会接收到数据记录。

- **SSL ID = W#16#0225的部分列表摘录:**  
对于已分配给调用OB的所有过程映像分区将返回一个数据记录。如果尚未将过程映像分区分配给调用OB, 则不会返回任何数据记录。

### 注意

可以将多个过程映像分区分配给时钟同步中断OB。在这种情况下, 会返回多个数据记录。

- **SSL ID = W#16#0F25的部分列表摘录:**  
返回的数据记录的最大数目。

有关数据记录含义的实例

SFC51 的调用参数	返回的变量	说明
SZL_ID = W#16#0125, INDEX = W#16#0008	tpa_nr = B#16#08, tpa_use = B#16#03, ob_nr = B#16#15	将返回一个数据记录。 输入/输出过程映像8被分配给OB 21，并由系统更新该过程映像。
SZL_ID = W#16#0125, INDEX = W#16#0009	-	未返回任何数据记录。 因此：过程映像分区9未被分配给任何OB。
SZL_ID = W#16#0225, INDEX = W#16#003D	tpa_nr = B#16#0A, tpa_use = B#16#C0, ob_nr = B#16#3D  tpa_nr = B#16#10, tpa_use = B#16#C0, ob_nr = B#16#3D	返回两个数据记录。 将输入和输出的过程映像10和16分配给OB 61。 可通过调用SFC 126和127在OB 61中将其更新。
SZL_ID = W#16#0225, INDEX = W#16#0001	tpa_nr = B#16#00, tpa_use = B#16#03, ob_nr = B#16#01	返回一个数据记录。 过程映像0分配给OB 1，用于输入和输出。它们将由系统进行更新。

### 31.13 SSL-ID W#16#xy32 - 通讯状态数据

#### 用途

如果读取SSL-ID为W#16#xy32的部分列表，则可获得模块通讯的状态数据。

#### 文件头

部分列表SSL-ID W#16#xy32的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL ID <ul style="list-style-type: none"> <li>W#16#0132: CPU的一个通讯部分的状态数据(始终为一个数据记录)。可以用INDEX参数指定CPU的通讯部分。</li> <li>W#16#0232: 一个通讯部分的状态数据(在一个处于RUN-REDUNDANT模式、单独模式或单机操作的H系统中，将返回2个数据记录。)。可以用INDEX参数指定CPU的通讯部分。</li> </ul>
INDEX	通讯部分 <ul style="list-style-type: none"> <li>对于SSL ID W#16#0132:               <ul style="list-style-type: none"> <li>W#16#0005            诊断</li> <li>W#16#0008            时间系统</li> <li>W#16#000B            时间系统</li> <li>W#16#000C            时间系统</li> </ul> </li> <li>对于SZL ID W#16#0232:               <ul style="list-style-type: none"> <li>W#16#0004            CPU保护级别、操作员控制设置和版本ID</li> </ul> </li> </ul>
LENTHDR	W#16#0028: 一个数据记录为20个字长(40字节)
N_DR	数据记录的数目

#### 数据记录

部分列表SSL-ID W#16#0132的数据记录始终为20个字长。数据记录具有不同的内容。具体内容取决于INDEX参数，换言之，取决于数据记录所属的CPU的通讯部分。

### 31.14 SSL-ID W#16#0132, 索引为W#16#0005的部分列表的数据记录摘录

#### 内容

SSL ID为W#16#0132、索引为W#16#0005的部分列表摘录包含有关模块上的诊断状态的信息。

#### 数据记录

SSL-ID为W#16#0132、索引为W#16#0005的部分列表摘录的数据记录具有如下结构：

名称	长度	含义
索引	1个字	W#16#0005: 诊断
Erw	1个字	扩展功能 0: 否 1: 是
发送	1个字	自动发送 0: 否 1: 是
Moeg	1个字	当前可以发送用户自定义的诊断消息 0: 否 1: 是
Res	16个字	保留

## 31.15 SSL-ID W#16#0132, 索引为W#16#0008的部分列表的数据记录摘录

### 内容

SSL-ID为W#16#0132、索引为W#16#0008的部分列表摘录包含有关模块上的时间系统状态的信息。

### 数据记录

SSL-ID为W#16#01032、索引W#16#0008的部分列表摘录的数据记录具有如下结构：

名称	长度	含义
索引	1个字	W#16#0008: 时间系统状态
Zykl	1个字	同步帧的周期
Korr	1个字	时间的修正因子
时钟0	1个字	运行系统计时器0: 以小时计
时钟1	1个字	运行系统计时器1: 以小时计
时钟2	1个字	运行系统计时器2: 以小时计
时钟3	1个字	运行系统计时器3: 以小时计
时钟4	1个字	运行系统计时器4: 以小时计
时钟5	1个字	运行系统计时器5: 以小时计
时钟6	1个字	运行系统计时器6: 以小时计
时钟7	1个字	运行系统计时器7: 以小时计
时间	4个字	当前日期和时间(格式: DATE_AND_TIME)
bszl_0到bszl_1	2字节	运行系统计时器已激活(位 =1: 运行系统计时器已激活)
bszl_0	1字节	位x: 运行系统计时器x, $0 \leq x \leq 7$
bszl_1	1字节	保留
bszü_0到bszü_1	2字节	运行系统计时器溢出(位 = 1: 溢出)
bszü_0	1字节	位x: 运行系统计时器x, $0 \leq x \leq 7$
bszü_1	1字节	保留
状态	1个字	时间状态(用于位分配, 参见下文)
Res	3字节	保留
status_valid	1字节	变量状态的有效性: B#16#01: 状态有效

状态

位	缺省值	描述
15	0	修正值的符号 (0: 正号, 1: 负号)
14 - 10	00000	修正值 此参数允许将帧中的基本时间修正为当地时间: 当地时间 = 基本时间 + 修正值 * 0.5 h 此修正考虑时区及夏令时和冬令时(标准时间)的时差。
9	0	保留
8	0	保留
7	0	通知时间 此参数指示下一时间调整是否也会包括从夏令时到冬令时(标准时间)的转换(反之亦然)。 (0: 未进行任何调整, 1: 进行了调整)。
6	0	夏令时/冬令时(标准时间)指示符 此参数指示使用修正值计算的当地时间是夏令时还是冬令时。 (0: 冬令时, 1: 夏令时)
5	0	参数未被S7使用。
4 - 3	00	时间分辨率 此参数指示传送的时钟时间的分辨率。 (00: 0.001秒, 01: 0.01秒, 10: 0.1秒, 11: 1秒)
2	0	参数未被S7使用。
1	0	参数未被S7使用。
0	0	同步失败 此参数指示来自外部时间主站(例如SICLOCK)的帧中传送的时间是否同步 (0: 同步失败, 1: 已同步) 注意: 仅当存在持续的外部时间同步时, 对CPU中此位的判断才有意义。

## 31.16 SSL-ID为W#16#0132、索引为W#16#000B的部分列表摘录的数据记录

### 内容

SSL ID为W#16#0132、索引为W#16#000B的部分列表摘录包含有关模块的32位运行系统计时器0 ... 7的状态的信息。

### 注意

SSL ID为W#16#0132、索引为W#16#0008的部分列表摘录将以16位模式显示运行系统计时器。

这将允许用户使用为CPU开发的程序，这些程序与16位的运行系统计时器和SSL ID为W#16#0132、索引为W#16#0008的部分列表摘录一起运行。

### 数据记录

SSL ID为W#16#0132、索引为W#16#000BB的部分列表摘录的数据记录具有如下结构：

名称	长度	含义
索引	1个字	W#16#000B: 时间系统状态
bszl_0	1字节	位x: 运行系统计时器x的状态, 0 x 7(位 = 1: 运行系统计时器忙)
bszl_1	1字节	保留
bszü_0	1字节	位x: 运行系统计时器x的溢出, 0 x 7(位 = 1: 溢出)
bszü_1	1字节	保留
时钟0	2个字	运行系统计时器0: 以小时计
时钟1	2个字	运行系统计时器1: 以小时计
时钟2	2个字	运行系统计时器2: 以小时计
时钟3	2个字	运行系统计时器3: 以小时计
时钟4	2个字	运行系统计时器4: 以小时计
时钟5	2个字	运行系统计时器5: 以小时计
时钟6	2个字	运行系统计时器6: 以小时计
时钟7	2个字	运行系统计时器7: 以小时计
Res	1个字	保留

### 31.17 SSL-ID为W#16#0132、索引为W#16#000C的部分列表摘录的数据记录

#### 内容

SSL ID为W#16#0132、索引为W#16#000C的部分列表摘录将包含有关模块的32位运行系统计时器8 ... 15的状态信息。

#### 数据记录

SSL ID为W#16#0132、索引为W#16#000CC的部分列表摘录的数据记录具有如下结构：

名称	长度	含义
索引	1个字	W#16#000C: 时间系统状态
Bszl_0	1字节	位x: 运行系统计时器(8+x)的状态, 0 x 7(位 = 1: 运行系统计时器忙)
Bszl_1	1字节	保留
Bszü_0	1字节	位x: 运行系统计时器(8+x)的溢出, 0 x 7(位 = 1: 溢出)
Bszü_1	1字节	保留
时钟8	2个字	运行系统计时器8: 以小时计
时钟9	2个字	运行系统计时器9: 以小时计
时钟10	2个字	运行系统计时器10: 以小时计
时钟11	2个字	运行系统计时器11: 以小时计
时钟12	2个字	运行系统计时器12: 以小时计
时钟13	2个字	运行系统计时器13: 以小时计
时钟14	2个字	运行系统计时器14: 以小时计
时钟15	2个字	运行系统计时器15: 以小时计
Res	1个字	保留



## 31.18 SSL-ID W#16#0232, 索引为W#16#0004的部分列表的数据记录摘录

### 内容

SSL-ID为W#16#0232、索引为W#16#0004的部分列表摘录将包含有关下列内容的信息：CPU保护级别、硬件配置和用户程序的操作员模式切换与版本标识的设置。

在处于RUN-REDUNDANT模式的H系统中，对于每个冗余CPU都将返回一个数据记录。

### 数据记录

SSL-ID为W#16#0232、索引为W#16#0004的部分列表摘录的数据记录具有如下结构：

名称	长度	含义
索引	1个字	<ul style="list-style-type: none"> <li>字节1: B#16#04: CPU保护级别、操作员控制设置和版本标识</li> <li>字节0: 标准CPU: B#16#00 H CPU: 位0至2: 机架号 位3: 0 = 待机CPU, 1 = 主站CPU 位4到7: 1111</li> </ul>
sch_schal	1个字	以模式选择器(1、2、3)设置的保护级别
sch_par	1个字	以参数设置的保护级别(0、1、2、3; 0: 无密码, 保护级别无效)
sch_rel	1个字	CPU的有效保护级别
bart_sch	1个字	模式选择器设置(1: RUN, 2: RUN-P, 3: STOP, 4: MRES, 0: 未定义或无法确定)
anl_sch	1个字	启动开关设置(1: CRST, 2: WRST, : 0: 未定义、不存在或无法确定)
ken_rel	1个字	用于有效版本标识的ID(0: 无效)
ken_ver1_hw	1个字	硬件配置的版本ID 1
ken_ver2_hw	1个字	硬件配置的版本ID 2
ken_ver1_awp	1个字	用户程序的版本ID 1
ken_ver2_awp	1个字	用户程序的版本ID 2
Res	8个字	保留

## 31.19 SSL-ID W#16#xy71 - H CPU的组信息

### 用途

SSL-ID为W#16#xy71的部分列表摘录包含有关H系统当前状态的信息。

### 文件头

部分列表SSL-ID W#16#xy71的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID: W#16#0071: 有关H系统当前状态的信息 W#16#0F71: 仅SSL部分列表标题信息
INDEX	W#16#0000
LENTHDR	W#16#0010: 一个数据记录为8个字长(16字节)
N_DR	W#16#0001: 数据记录的数目

### 数据记录

部分列表摘录ID W#16#xy71的数据记录具有如下结构：

内容	长度	含义
Redinf	2字节	关于冗余的信息 W#16#0011: 2个H系统中的一个 W#16#0012: 1个H系统(共2个)
Mwstat1	1字节	状态字节1 位0: 保留 位1: 保留 位2: 保留 位3: 保留 位4: 机架0中CPU的H状态 =0: 待机CPU =1: 主站CPU 位5: 机架1中CPU的H状态 =0: 待机CPU =1: 主站CPU 位6: 保留 位7: 保留

内容	长度	含义
Mwstat2	1字节	状态字节2 位0: 同步链接01的状态: CPU 0和CPU 1之间的同步 =0: 不可能 =1: 可能 位1: 0 位2: 0 位3: 保留 位4: = 0: CPU未插入机架0中 =1: CPU插入到机架0中 (在冗余模式中: 位4 = 0) 位5: = 0: CPU未插入机架1中 =1: CPU插入到机架1中 (在冗余模式中: 位5 = 0) 位6: 保留 位7: 自上次重新启用后的待机主站转换 =0: 否 =1: 是
Hsfcinfo	2字节	SFC90 "H_CTRL"的信息字 位0: = 0: 重新启用未激活 =1: 重新启用已激活 位1: = 0: 待机更新已启用 =1: 待机更新已禁用 位2: = 0: 待机链接已启用 =1: 待机链接已禁用 位3: 保留 位4: 保留 位5: 保留 位6: 保留 位7: = 1: 按更新请求进行更新 位8: = 1: 无更新请求而更新
Samfehl	2字节	保留
Bz_cpu_0	2字节	机架0中CPU的模式 W#16#0001: STOP(更新) W#16#0002: STOP(复位存储器) W#16#0003: STOP(自检) W#16#0004: STOP(内部) W#16#0005: STARTUP(冷重启) W#16#0006: STARTUP(暖重启) W#16#0007: STARTUP(热重启) W#16#0008: RUN(单独模式) W#16#0009: RUN-R(冗余模式) W#16#000A: HOLD W#16#000B: LINK-UP W#16#000C: UPDATE W#16#000D: DEFECTIVE W#16#000E: SELFTEST W#16#000F: NO POWER
Bz_cpu_1	2字节	机架1中CPU的模式 (与bz_cpu_0值相同)
Bz_cpu_2	2字节	保留
Cpu_valid	1字节	变量bz_cpu_0和bz_cpu_1的有效性 B#16#01: bz_cpu_0有效 B#16#02: bz_cpu_1有效 B#16#03: bz_cpu_0和bz_cpu_1有效
保留	1字节	保留

## 31.20 SSL-ID W#16#xy74 - 模块LED的状态

### 用途

对于标准CPU(如存在)和H CPU读取部分列表SSL-ID W#16#xy74, 将获得模块LED的状态。

如果H CPU处于非冗余的H模式, 将获得被寻址的CPU的LED状态。如果H CPU处于RUN-REDUNDANT模式, 则将返回所有冗余的H CPU的LED状态。

### 文件头

部分列表SSL-ID W#16#xy74的报头结构如下所示:

内容	含义
SSL-ID	部分列表摘录的SSL-ID W#16#0174 一个LED的状态。 可使用INDEX参数选择LED。
INDEX	LED标识符(仅与SSL-ID W#16#0174相关) W#16#0001: SF(组出错) W#16#0002: INTF(内部出错) W#16#0003: EXTF(外部出错) W#16#0004: RUN W#16#0005: STOP W#16#0006: FRCE(强制) W#16#0007: CRST(冷重启) W#16#0008: BAF(总线上的电池故障/超载、 电池电压短路) W#16#0009: USR(用户自定义) W#16#000A: USR1(用户自定义) W#16#000B: BUS1F(总线出错接口1) W#16#000C: BUS2F(总线出错接口2) W#16#000D: REDF(冗余出错) W#16#000E: MSTR(主站) W#16#000F: RACK0(机架号0) W#16#0010: RACK1(机架号1) W#16#0011: RACK2(机架号2) W#16#0012: IFM1F(接口出错接口模块1) W#16#0013: IFM2F(接口出错接口模块2)
LENTHDR	W#16#0004: 一个数据记录为2个字长(4字节)
N_DR	数据记录的数目

## 数据记录

部分列表摘录SSL-ID W#16#0074的数据记录具有如下结构:

名称	长度	含义
cpu_led_ID	1个字	<ul style="list-style-type: none"> <li>• 字节0               <ul style="list-style-type: none"> <li>- 标准CPU: B#16#00</li> <li>- H-CPU: 位0至2: 机架号 位3: 0 = 待机CPU, 1 = 主站CPU 位4到7: 1111</li> </ul> </li> </ul> 字节1: LED标识符
led_on	1字节	LED的状态: 0: 关闭 1: 打开
led_blink	1字节	LED的闪烁状态: 0: 不闪烁 1: 正常闪烁(2赫兹) 2: 缓慢闪烁(0.5赫兹)

## 31.21 SSL-ID W#16#xy75 - H系统中切换的DP从站

### 用途

对于处于冗余H工作模式的H系统的CPU，如果读取部分列表SSL-ID W#16#xy75，将获得有关H系统和切换的DP从站之间通讯状态的信息。

该部分列表将报告当前用于与DP从站之间通讯的DP主站系统接口模块被插入到哪一个机架中。

### 文件头

部分列表SSL-ID W#16#xy75的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID W#16#0C75: H系统与切换的 DP从站间的通讯状态 可使用INDEX参数选择DP从站。
INDEX	DP从站接口模块的诊断地址
LENTHDR	W#16#0010: 一个数据记录为8个字长(16字节)
N_DR	W#16#0001: 数据记录的数目

### 数据记录

部分列表ID SSL-ID W#16#xy75的数据记录具有如下结构：

名称	长度	含义
adr1_bgt0	1个字	DP主站接口模块插入到机架0中的 DP从站接口模块的第一个地址部分： DP主站系统标识号和站号
adr2_bgt0	1个字	DP主站接口模块插入到机架0中的DP从站接口模块的第二个 地址部分： 插槽和子模块插槽
adr1_bgt1	1个字	DP主站接口模块插入到机架1中的 DP从站接口模块的第一个地址部分： DP主站系统标识号和站号
adr2_bgt1	1个字	DP主站接口模块插入到机架1中的DP从站接口模块的第二个 地址部分： 插槽和子模块插槽
Res	2个字	保留
Logadr	1个字	DP从站接口模块的诊断地址： <ul style="list-style-type: none"> <li>• 位0到14: 逻辑基址</li> <li>• 位15: I/O标识符(0 = 输入, 1 = 输出)</li> </ul>

名称	长度	含义
Slavestatus	1个字	通讯状态： <ul style="list-style-type: none"><li>• 位0 = 1: 无权访问DP主站接口模块插入到机架0中的DP从站接口模块</li><li>• 位1 = 1: 无权访问DP主站接口模块插入到机架1中的DP从站接口模块</li><li>• 位2到7: 保留(每位 = 0)</li><li>• 位8 = 1: 两个通讯通道均运行正常; 当前正在通过机架0中的DP主站接口模块进行通讯</li><li>• 位9 = 1: 两个通讯通道均运行正常; 当前正在通过机架1中的DP主站接口模块进行通讯</li><li>• 位10到15: 保留(每位 = 0)</li></ul>

## 31.22 SSL-ID W#16#xy90 - DP主站系统信息

### 用途

如果读取SSL-ID为W#16#xy90的部分列表，将获得CPU已知的所有DP主站系统的状态信息。

### 文件头

部分列表SSL-ID W#16#xy90的报头结构如下所示：

内容	含义	
SSL-ID	部分列表摘录的SSL-ID	
	W#16#0090:	CPU已知的所有DP主站系统的信息
	W#16#0190:	一个DP主站系统的信息
	W#16#0F90:	仅限SSL部分列表报头信息
INDEX	<ul style="list-style-type: none"> <li>对于SSL-ID为W#16#0190的部分列表摘录： 低字节：B#16#00 高字节：DP主站系统标识号</li> <li>对于SSL-ID为W#16#0090和W#16#0F90的部分列表摘录： W#16#0000</li> </ul>	
LENTHDR	W#16#000E:	一个数据记录为7个字长(14字节)
N_DR	数据记录的数目 <ul style="list-style-type: none"> <li>对于SSL-ID位W#16#0190的部分列表摘录： 0到1</li> <li>对于SSL-ID为W#16#0090的部分列表摘录：                             <ul style="list-style-type: none"> <li>对于标准CPU： 0至14</li> <li>对于H系统： 0 - 12(在除冗余以外的所有系统状态下) 0 - 2 x 12(在冗余系统状态下)</li> </ul> </li> </ul>	



## 数据记录

部分列表ID 为W#16#xy90的数据记录具有如下结构：

名称	长度	含义	
dp_m_id	1字节	DP主站系统标识号	
rack_dp_m	1字节	DP主站的机架号 <ul style="list-style-type: none"> <li>对于标准CPU: 0</li> <li>对于H系统: 0或1</li> </ul>	
Steckpl_dp_m	1字节	DP主站的插槽或CPU的插槽(具有集成的DP接口)	
Subm_dp_m	1字节	<ul style="list-style-type: none"> <li>对于集成的DP接口: DP主站的接口编号: <ul style="list-style-type: none"> <li>1: X2</li> <li>2: X1</li> <li>3: IF1</li> <li>4: IF2</li> </ul> </li> <li>对于外部DP接口: 0</li> </ul>	
Logadr	1个字	DP主站的逻辑起始地址	
dp_m_sys_cpu	1个字	保留	
dp_m_sys_dpm	1个字	保留	
dp_m_state	1字节	DP主站系统的其它属性	
		位0:	DP模式 <ul style="list-style-type: none"> <li>0: S7兼容</li> <li>1: DPV1</li> </ul>
		位1	DP循环 <ul style="list-style-type: none"> <li>0: 不等距</li> <li>1: 等距</li> </ul>
		位2到6:	保留
		位7:	DP主站类型 <ul style="list-style-type: none"> <li>0: 集成的DP主站</li> <li>1: 外部DP主站</li> </ul>
保留	3字节	保留	

#### 有关多值计算的注意(只对S7-400)

所有部分列表仅提供被分配给一个CPU的模块的有关信息。因此，在多值计算模式中，必须对所有CPU进行采样以便获得所有连接模块的数据。

## 31.23 SSL-ID W#16#xy91 - 模块的状态信息

### 用途

如果读取SSL-ID 为W#16#xy91的部分列表，将获得已分配给CPU的模块的状态信息。

### 注意

不能为具有压缩地址(ET 200S)的子模块判断SSL-ID W#16#xy91。

### 文件头

部分列表SSL-ID W#16#xy91的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID
W#16#0091	所有插入式模块和子模块的模块状态信息(只对S7-400)
W#16#0191	所有具有出错模块标识符的未取消激活的模块/机架的状态信息(只对S7-400)
W#16#0291	所有有故障且未取消激活的模块的模块状态信息(只对S7-400)
W#16#0391	所有不可用模块的模块状态信息(只对S7-400)
W#16#0591	主机模块的所有子模块的模块状态信息
W#16#0991	DP主站系统的模块状态信息
W#16#0A91	所有DP子模块和DP主站系统的状态信息(仅S7-300，不包括CPU 318-2 DP)
W#16#0C91	中央机架中的模块或具有逻辑基址的集成式DP接口中的模块的模块状态信息
W#16#4C91	通过逻辑基址连接到外部DP接口的模块的模块状态信息 如果使用四个以上的外部DP接口，则可能产生出错结果RET_VAL W#16#80A4。
W#16#0D91	指定机架/指定站(DP)中所有模块的模块状态信息
W#16#0E91	所有已组态模块的模块状态信息

内容	含义
INDEX	<ul style="list-style-type: none"> <li>对于SSL-ID为W#16#4C91的部分列表摘录： <ul style="list-style-type: none"> <li>S7-400: 位0到14: 模块的逻辑基址 位15: 0 = 输入, 1 = 输出</li> <li>S7-300: 模块起始地址</li> </ul> </li> <li>对于SSL-ID为W#16#4C91的部分列表摘录(只对S7-400): 位0到14 : 模块的逻辑基址 位15 : 0 = 输入, 1 = 输出</li> <li>对于SSL-ID为W#16#0091、W#16#0191、W#16#0291、W#16#0391、W#16#0491、W#16#0591、W#16#0A91、W#16#0E91、W#16#0F91的部分列表摘录: INDEX不相关, 所有模块(在机架和分布式I/O中)</li> </ul> <p>对于SSL-ID为W#16#0991的部分列表摘录:</p> <p>W#16#00xx 机架中所有的模块和子模块 (xx包含机架号)</p> <p>W#16#xx00 DP主站系统的所有模块 (xx包含DP主站系统标识号)</p> <p>W#16#xxyy DP站的所有模块 (xx包含DP主站系统标识号, yy包含站号)</p>
LENTHDR	W#16#0010: 一个数据记录为8个字长(16字节)
N_DR	数据记录的数目。根据产品而定, 在SFC51中传送的记录数可能较少

对于W#16#0091、W#16#0191和W#16#0F91, 每个机架将另外提供两个数据记录:

- 一个电源记录(只要它存在并已被规划)和
- 一个机架记录。
- 如果是集中式结构, 则记录的顺序将为: PS、插槽1、插槽2、.....、插槽18、机架。

部分列表ID W#16#xy91的数据记录具有如下结构:

名称	长度	含义
Adr1	1个字	物理地址的机架数目 (DP主站系统标识号和DP的站号)
Adr2	1个字	插槽和子模块插槽
Logadr	1个字	第一个已分配的逻辑I/O地址(基址)
Solltyp	1个字	保留
isttyp	1个字	保留
报警	1个字	保留(00xx=CPU编号: 1-4)

名称	长度	含义
Eastat	1个字	I/O状态 位0 = 1: 模块出错 (由诊断中断检测) 位1 = 1: 模块存在 位2 = 1: 模块不存在 位3 = 1: 模块已禁用 位4 = 1: 站出错(仅限典型插槽) 位5 = 1: M7: 模块可以作为子模块的主机 S7: 此模块/站中的CiR事件忙或尚未完成 位6 = 1: 对于S7-400保留 位7 = 1: 局部总线分段中的模块 位8至15: 逻辑地址的数据ID (输入: B#16#B4, 输出: B#16#B5, 外部DP接口: B#16#FF)
Ber_bgbr	1个字	区域ID/模块宽度 位0到2: 模块宽度 位3: 保留 位4到6: 区域ID 0 = S7-400 1 = S7-300 2 = ET区域 3 = P区域 4 = Q区域 5 = IM3区域 6 = IM4区域 位7: 保留

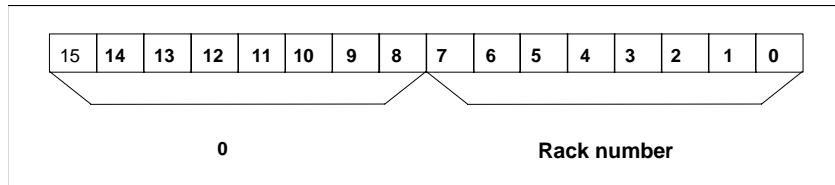
在某些模块中，下列值将显示在记录中：

名称	PS (只对S7-400)	CPU	IFM-CPU (S7-300)	机架 (只对S7-400)
Adr1	机架号	如上所述的标准信息	如上所述的标准信息	机架号
Adr2	W#16#01FF	W#16#0200或 W#16#0200到 W#16#1800	W#16#0200	W#16#00FF
Logadr	W#16#0000	W#16#7FFF	W#16#007C	W#16#0000
Solltyp	如上所述的标准信息	W#16#00C0或 W#16#0081或 W#16#0082	W#16#00C0	如上所述的标准信息
Eastat	W#16#0000	如上所述的标准信息	如上所述的标准信息	W#16#0000
Ber_bgbr	W#16#0000	W#16#0011或 W#16#0001或 W#16#0002	W#16#0011	W#16#0000

**Adr1**

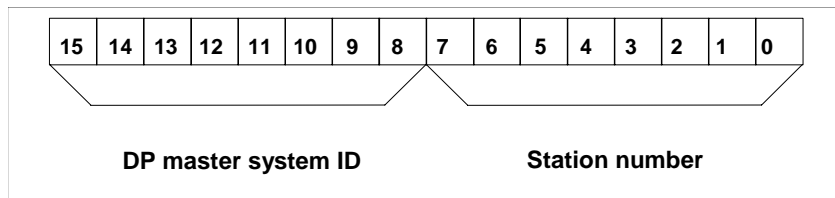
参数adr1包含:

- 作为中央机架安装时的机架号



作为中央机架安装时参数adr1的各个位

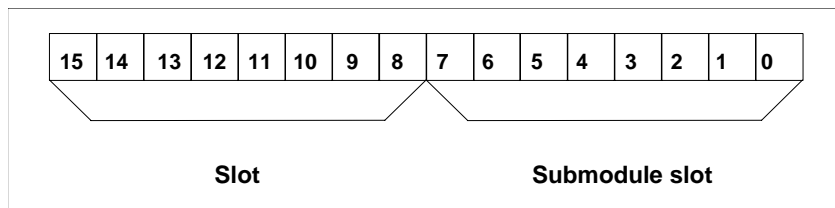
- 分布式组态中的  
DP主站系统标识号  
站号。



在分布式组态中, 参数adr1的位。

**Adr2**

参数adr2包含插槽和子模块插槽。



参数adr2的位。

**有关多值计算的信息(只对S7-400)**

所有部分列表仅提供分配给一个CPU的模块的有关信息。因此, 在多值计算模式中, 必须采样所有的CPU以获取所有连接模块的数据。

## 31.24 SSL-ID W#16#xy92 - 机架/站的状态信息

### 用途

如果读取部分列表SSL-ID W#16#xy92，将获得有关一个DP主站系统的集中安装的机架及站的期望和当前的硬件组态的信息。

### 通过SFC51 “RDSYSST”用S7-400 CPU读取SSL

如果通过SFC51读取部分列表，则必须使SFC51的参数SSL\_ID与INDEX相匹配。

SSL_ID	INDEX
W#16#0092或 W#16#0292或 W#16#0692或	通过 <b>集成</b> DP开关连接的DP主站系统的DP主站系统标识号。
W#16#4092或 W#16#4292或 W#16#4692或	通过 <b>外部</b> DP开关连接的DP主站系统的DP主站系统标识号。

### 文件头

部分列表SSL-ID W#16#xy92的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID: W#16#0092: 通过集成DP接口连接的DP主站系统的中央机架/站的期望状态 W#16#4092: 通过外部DP接口连接的DP主站系统的站的期望状态 W#16#0292: 通过集成DP接口连接的DP主站系统的中央机架/站的实际状态 W#16#0392: 至少一个电池发生故障时CPU机架/站的电池充电缓冲状态 W#16#0492: 一个CPU的所有机架/站的整个电池充电缓冲的状态 W#16#0592: 一个CPU的所有机架/站的24V电源状态 W#16#4292: 通过外部DP接口连接的DP主站系统的站的实际状态 W#16#0692: 中央组态中的扩展机架/通过集成DP接口的DP主站系统的站的“正常”状态 W#16#4692: 中央组态中的扩展机架/通过集成DP接口的DP主站系统的站的
INDEX	0/DP主站系统标识号
LENTHDR	W#16#0010: 一个数据记录为8个字长(16字节)
N_DR	数据记录的数目

## 数据记录

SSL ID为W#16#xy92的部分列表的数据记录具有如下结构：

内容	长度	含义
status_0到 status_15	16字节	机架状态/站状态或备份状态。(备份状态仅与DP模块相关) W#16#0092: 位 = 0: 机架/站未组态 位 = 1: 机架/站已组态 W#16#4092 位 = 0: 站未组态 位 = 1: 站已组态 W#16#0192: 位 = 0: 站未组态或已组态并已激活 位 = 1: 站已组态并已激活 W#16#0292: 位 = 0: 机架/站故障、已取消激活或未组态 位 = 1: 机架/站存在、已激活、尚未发生故障 W#16#4292: 位 = 0: 站故障、已取消激活或未组态 位 = 1: 站存在、已激活、尚未发生故障 W#16#0692: 位 = 0: 扩展机架/一个站的所有模块存在、 可用并且正常，站已激活 位 = 1: 扩展机架/一个站至少有1个模块不 “正常”或者该站已取消激活 W#16#4692: 位 = 0: 一个站的所有模块存在、可用并且 正常，该站已激活 位 = 1: 一个站至少有1个模块不“正常”或者 该站已取消激活
status_0	1字节	位0: 中央机架(INDEX = 0)或 站1(INDEX tu0) 位1: 1. 扩展机架或站2 : : 位7: 7. 扩展机架或站8
status_1	1字节	位0: 8. 扩展机架或站9 : : 位7: 15. 扩展机架或站16
status_2	1字节	位0: 16. 扩展机架或站17 : : 位5: 21 扩展机架或站22 位6: 0 或站23 位7: 0 或站24
status_3	1字节	位0: 0 或站 25 : 位5: 0 或站30 位6: 扩展机架(SIMATIC S5区域) 或站31 位7: 0 或站32

内容	长度	含义
status_4	1字节	位0: 0 或站 33 : 位7: 0 或站40
:		
:		
status_15	1字节	位0: 0 或站 121 : 位7: 0 或站128

---

**有关多值计算的信息(只对S7-400)**

所有部分列表仅提供分配给一个CPU的模块的有关信息。因此，在多值计算模式中，必须采样所有的CPU以获取所有连接模块的数据。

---



## 31.25 SSL-ID W#16#xy95 - 扩展DP主站系统信息

### 用途

SSL ID为W#16#xy95的部分列表提供有关CPU已知的所有DP主站系统的扩展状态信息。与SSL ID为W#16#xy90的部分列表相比较，此列表包含与DP主站系统的时钟同步有关的附加信息。

### 文件头

部分列表SSL ID W#16#xy95的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL ID
	W#16#0195: 有关DP主站系统的扩充信息
	W#16#0F95: 仅限与SSL部分列表的报头有关的信息
INDEX	<ul style="list-style-type: none"> <li>对于SSL ID为W#16#0195的部分列表摘录： 低字节：B#16#00 高字节：DP主站系统标识号</li> <li>对于SSL ID为W#16#0F95的部分列表摘录： W#16#0000</li> </ul>
LENTHDR	W#16#0028: 一个数据记录为20个字长(40字节)
N_DR	数据记录的数目： 对于SSL ID为W#16#0195的部分列表摘录：0到1

### 数据记录

ID为W#16#xy95的部分列表的数据记录具有如下结构：

名称	长度	含义
dp_m_id	1字节	DP主站系统标识号
rack_dp_m	1字节	DP主站的模块机架号 <ul style="list-style-type: none"> <li>对于标准CPU：0</li> <li>对于H系统：0或1</li> </ul>
steckpl_dp_m	1字节	DP主站的插槽或CPU的插槽(具有集成的DP接口)
subm_dp_m	1字节	<ul style="list-style-type: none"> <li>对于集成的DP接口：DP主站的接口ID： <ul style="list-style-type: none"> <li>1: X2</li> <li>2: X1</li> <li>3: IF1</li> <li>4: IF2</li> </ul> </li> <li>对于外部DP接口：0</li> </ul>

名称	长度	含义	
logadr	2字节	DP主站的逻辑起始地址	
dp_m_sys_cpu	2字节	保留	
dp_m_sys_dpm	2字节	保留	
dp_m_state	1字节	DP主站系统的更多属性	
		位0:	DP模式 <ul style="list-style-type: none"> <li>• 0: S7兼容</li> <li>• 1: DPV1</li> </ul>
		位1:	DP循环 <ul style="list-style-type: none"> <li>• 0: 不等距</li> <li>• 1: 等距</li> </ul>
		位2到6:	保留
		位7:	DP主站的类型 <ul style="list-style-type: none"> <li>• 0: 集成的DP主站</li> <li>• 1: 外部DP主站</li> </ul>
保留	3字节	保留	
tsal_ob	1字节	分配的时钟同步中断OB(只有DP循环等距时才相关)	
保留	1字节	保留	
波特率	4字节	DP主站系统的传送速率(十六进制值)	
dp_iso_takt	4字节	等距DP循环的期间(单位 $\mu$ s)	
保留	16字节	保留	

#### 有关多值计算的信息(只对S7-400)

所有部分列表仅提供分配给一个CPU的模块的有关信息。因此，在多值计算模式中，必须采样所有的CPU以获取所有连接模块的数据。

## 31.26 SSL-ID W#16#xyA0 - 诊断缓冲区

### 用途

如果读取SSL-ID 为W#16#xyA0的部分列表，可获得该模块诊断缓冲区中的条目。

### 文件头

部分列表SSL-ID W#16#xyA0的报头结构如下所示：

内容	含义
SSL-ID	部分列表摘录的SSL-ID： W#16#00A0： 当前模式下所有条目可用 W#16#01A0： 最新条目；使用INDEX参数指定最新条目的数目。 如果诊断缓冲区中的消息数小于已组态的最大消息数，SFC51会通过此部分列表摘录提供无效值。因此，应避免未备份的断电损失！ W#16#0FA0： 仅限部分列表报头信息
INDEX	仅适用于SSL-ID W#16#01A0：最新条目的数目
LENTHDR	W#16#0014： 一个数据记录为10个字长(20字节)
N_DR	数据记录的数目

### 数据记录

部分列表SSL-ID W#16#xyA0的数据记录具有如下结构：

名称	长度	含义
ID	1个字	事件标识符
信息	5字	有关事件及其结果的信息
时间	4个字	事件的时间标志

### 诊断缓冲区

使用STEP 7可在诊断缓冲区中获取事件的更多信息。

## 31.27 SSL-ID W#16#00B1 - 模块的诊断信息

### 用途

如果读取SSL-ID为W#16#00B1部分列表，将获取具有诊断能力的模块的前4个诊断字节。

### 文件头

部分列表SSL-ID W#16#00B1的报头结构如下所示：

	含义
SSL-ID	W#16#00B1
INDEX	位0到位14: 逻辑基址 位15: 0 = 输入, 1 = 输出
LENTHDR	W#16#0004: 一个数据记录为2个字长(4字节)
N_DR	1

### 数据记录

部分列表SSL-ID W#16#00B1的数据记录具有如下结构：

名称	长度	含义
byte1	1字节	位0: 模块故障/正常(组故障ID) 位1: 内部故障 位2: 外部故障 位3: 存在通道出错 位4: 无外部辅助电压 位5: 无前连接器 位6: 未分配参数给模块 位7: 模块的参数出错
byte2	1字节	位0到位3: 模块等级(CPU、FM、CP、IM、SM位4: ...) 位4: 存在通道信息 位5: 存在用户信息 位6: 来自替换的诊断中断 位7: 保留(用0初始化)
byte3	1字节	位0: 用户模块不正确/不存在 位1: 通讯故障 位2: RUN/STOP模式(0 = RUN, 1 = STOP) 位3: 监视狗已响应 位4: 内部模块电源故障 位5: 电池耗尽(BFS) 位6: 整个缓冲区故障 位7: 保留(用0初始化)

---

名称	长度	含义
byte4	1字节	位0: 扩展机架故障(由IM检测) 位1: 处理器故障 位2: EPROM出错 位3: RAM出错 位4: ADC/DAC出错 位5: 保险丝熔断 位6: 硬件出错丢失 位7: 保留(用0初始化)

---

**有关多值计算的信息(只对S7-400)**

所有的部分列表只提供分配给一个CPU的模块的有关信息。因此，在多值计算模式中，必须对所有CPU进行采样以便获得所有连接模块的数据。

---

## 31.28 SSL-ID W#16#00B2 - 对应物理地址的诊断数据记录1

### 用途

如果读取SSL-ID为W#16#00B2的部分列表，将获得集中机架中的模块的诊断数据记录1(不适用于DP或子模块)。使用机架和插槽号指定编号。

### 文件头

部分列表SSL-ID W#16#00B2的报头结构如下所示：

内容	含义
SSL-ID	W#16#00B2
INDEX	W#16#xxyy:    xx包含机架号 yy包含插槽号
LENTHDR	数据记录的长度取决于模块。
N_DR	1

### 数据记录

部分列表SSL-ID W#16#00B2的数据记录的大小及其内容取决于特定模块。有关详细信息，请参考/70/、/101/以及描述相关模块的手册。

---

#### 有关多值计算的信息(只对S7-400)

所有的部分列表只提供分配给一个CPU的模块有关信息。因此，在多值计算模式中，必须对所有CPU进行采样以便获得所有连接模块的数据。

---

## 31.29 SSL-ID W#16#00B3 - 带逻辑基址的模块诊断数据

### 用途

如果读取SSL-ID 为W#16#00B3的部分列表，将获得模块的所有诊断数据。还可以获取DP和子模块的此信息。使用模块的逻辑基址来选择模块。

### 文件头

部分列表SSL-ID W#16#00B3的报头结构如下所示：

内容	含义
SSL-ID	W#16#00B3
INDEX	位0到位14：逻辑基址 位15：0 = 输入，1 = 输出
LENTHDR	数据记录的长度取决于模块。
N_DR	1

### 数据记录

部分列表SSL-ID W#16#00B3的数据记录的大小及其内容取决于具体模块。有关详细信息，请参考 [/70/](#)、[/101/](#)以及描述相关模块的手册。

---

#### 有关多值计算的信息(只对S7-400)

所有部分列表仅提供分配给一个CPU的模块的有关信息。因此，在多值计算模式中，必须采样所有的CPU以获取所有连接模块的数据。

---

#### 注意

使用SFC51，只能在OB82外部读取SSL-ID为W#16#00B3的部分列表。

---

### 31.30 SSL-ID W#16#00B4 - DP从站的诊断数据

#### 用途

如果读取SSL-ID 为W#16#00B4的部分列表，将获得DP从站的诊断数据。此诊断数据按照EN 50 170 Volume 2，PROFIBUS标准构建。可使用已组态的诊断地址选择模块。

#### 文件头

部分列表SSL-ID W#16#00B4的报头结构如下所示：

内容	含义
SSL-ID	W#16#00B4
INDEX	DP从站的已组态诊断地址
LENTHDR	数据记录的长度。最大长度为240字节。对于具有大于240字节的诊断数据长度(最多可达244字节)的标准从站，只读取前240字节并在数据中置位溢出位。
N_DR	1

#### 数据记录

部分列表SSL-ID W#16#00B4的数据记录具有如下结构：

名称	长度	含义
状态1	1字节	站状态1
状态2	1字节	站状态2
状态3	1字节	站状态3
stat_nr	1字节	主站号
ken_hi	1字节	供应商ID(高字节)
ken_lo	1字节	供应商ID(低字节)
....	....	特定从站的更多专用诊断数据

#### 有关多值计算的信息(只对S7-400)

所有的部分列表只提供分配给一个CPU的模块有关信息。因此，在多值计算模式中，必须对所有CPU进行采样以便获得所有连接模块的数据。



## 32 事件

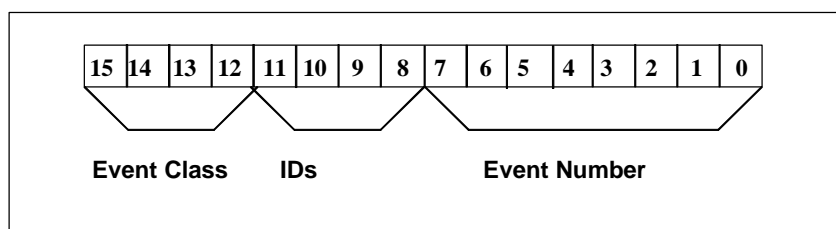
### 32.1 事件和事件标识符

#### 事件

在SIMATIC S7可编程逻辑控制器中所有事件有编号。允许将消息文本与事件相关。

#### 事件标识符

每个事件分配一个事件标识符。事件标识符的结构如下：



事件标识符的结构。

#### 事件等级

事件等级如下所示：

编号	事件等级
1	标准OB事件
2	同步出错
3	异步出错
4	模式转换
5	运行期事件
6	通讯事件
7	故障安全和容错系统事件
8	模块上的标准诊断数据
9	预定义用户事件
A、B	可自由定义的事件
C、D、E	保留
F	除CPU以外的模块(例如, CP、FM)的事件

## 识别

标识符用于区分事件的类型。四个位具有下列意义：

事件标识符中的位号	含义
8	= 0 事件离开状态 = 1 事件进入状态
9	= 1 诊断缓冲区中的条目
10	= 1 内部出错
11	= 1 外部出错

## 32.2 事件等级1 - 标准OB事件

事件标识符	事件
W#16#1381	请求手动暖重启
W#16#1382	请求自动暖重启
W#16#1383	请求手动热重启
W#16#1384	请求自动热重启
W#16#1385	请求手动冷重启
W#16#1386	请求自动冷重启
W#16#1387	主CPU: 请求手动冷重启
W#16#1388	主CPU: 请求自动冷重启
W#16#138A	主CPU: 请求手动暖重启
W#16#138B	主CPU: 请求自动暖重启
W#16#138C	待机CPU: 请求手动热重启
W#16#138D	待机CPU: 请求自动热重启

## 32.3 事件等级2 - 异步出错

事件标识符	事件	OB
W#16#2521	BCD转换出错	OB 121
W#16#2522	读取时区域长度出错	OB 121
W#16#2523	写入时区域长度出错	OB 121
W#16#2524	读取时的区域出错	OB 121
W#16#2525	写入时区域出错	OB 121
W#16#2526	定时器编号出错	OB 121
W#16#2527	计数器编号出错	OB 121
W#16#2528	读取时的对齐出错	OB 121
W#16#2529	写入时的对齐出错	OB 121
W#16#2530	访问DB时发生写出错	OB 121
W#16#2531	访问DI时发生写出错	OB 121
W#16#2532	打开DB时块编号出错	OB 121
W#16#2533	打开DI时块编号出错	OB 121
W#16#2534	调用FC时块编号出错	OB 121
W#16#2535	调用FB时块编号出错	OB 121
W#16#253A	未加载DB	OB 121
W#16#253C	未加载FC	OB 121
W#16#253D	未装载SFC	OB 121
W#16#253E	未加载FB	OB 121
W#16#253F	未装载SFB	OB 121
W#16#2942	I/O访问出错, 读取	OB 122
W#16#2943	I/O访问出错, 写入	OB 122
W#16#2544	I/O访问出错, 在第n次读取访问时发生(n>1)	OB 122

## 32.4 事件等级3 - 异步出错

事件标识符	事件	OB
W#16#3501	超出周期时间。	OB 80
W#16#3502	用户接口(OB或FRB)请求出错	OB 80
W#16#3503	处理优先级时延迟过长	-
W#16#3505	由于新时钟的设置而跳过日时钟中断	OB 80
W#16#3506	在从HOLD变为RUN时跳过时钟中断	OB 80
W#16#3507	多个OB请求出错导致内部缓冲区溢出	OB 80
W#16#3508	同步周期中断定时出错	OB 80
W#16#3509	中断负载过大导致中断丢失	OB 80
W#16#350A	在CiR之后恢复RUN模式	OB 80
W#16#3921/3821	<b>BATTF</b> : 中央机架的备用电池至少有一个发生故障/ 问题已排除  注意: 只有当其中一个备用电池出现故障(如果有冗余备用 电池)时, 才会出现事件进入状态。如果其它备用电池也发生 故障, 则不会再次出现该事件。	OB 81
W#16#3922/3822	<b>BAF</b> : 中央机架的备用电压发生故障/问题已排除	OB 81
W#16#3923/3823	中央机架的24V电源故障/问题已排除	OB 81
W#16#3925/3825	<b>BATTF</b> : 冗余中央机架的备用电池至少有一个发生故障/ 问题已排除	OB 81
W#16#3926/3826	<b>BAF</b> : 冗余中央机架的备用电压发生故障/问题已排除	OB 81
W#16#3917/3827	冗余中央机架的24V电源发生故障/问题已排除	OB 81
W#16#3931/3831	<b>BATTF</b> : 扩展机架的备用电池至少有一个发生故障/问题已 排除	OB 81
W#16#3932/3832	<b>BAF</b> : 扩展机架上的备用电压发生故障/问题已排除	OB 81
W#16#3933/3833	至少有一个扩展机架上的24V电源发生故障/问题已排除	OB 81
W#16#3942	模块出错	OB 82
W#16#3842	模块正常	OB 82
W#16#3861	模块/接口模块已插入, 模块类型正确	OB 83
W#16#3961	模块/接口模块已删除, 不能寻址	OB 83
W#16#3863	模块/接口模块已插入, 但模块类型出错	OB 83
W#16#3864	模块/接口模块已插入, 但出现问题(类型标识号不可读)	OB 83
W#16#3865	模块已插入, 但模块参数分配出错	OB 83
W#16#3866	可以再次寻址模块, 负载电压出错已排除	OB 83
W#16#3966	无法寻址模块, 负载电压出错	OB 83
W#16#3367	启动模块重新组态	OB 83
W#16#3267	结束模块重新组态	OB 83

事件标识符	事件	OB
W#16#3968	模块重组态已结束, 有出错	OB 83
W#16#3571	在嵌套层中嵌套深度过高	OB 88
W#16#3572	主控继电器的嵌套深度过高	OB 88
W#16#3573	同步出错后的嵌套深度过高	OB 88
W#16#3574	块调用(U堆栈)的嵌套深度过高	OB 88
W#16#3575	块调用(B堆栈)的嵌套深度过高	OB 88
W#16#3576	本地数据分配出错	OB 88
W#16#3578	未知指令	OB 88
W#16#357A	跳转指令指向到块外部的目标	OB 88
W#16#3884	接口模块已插入	OB 83
W#16#3984	接口模块已删除	OB 83
W#16#3981	输入状态时发生接口出错	OB 84
W#16#3881	离开状态时发生接口出错	OB 84
W#16#3582	检测到存储器出错并已由操作系统更正	OB 84
W#16#3583	已检测和更正备忘录出错的累积	OB 84
W#16#3585	PC操作系统中出错(仅限LC RTX)	OB 84
W#16#3986	H-Sync链接的性能受到负面影响	OB 84
W#16#3587	已检测到多位内存出错并已更正	OB 84
W#16#35A1	未找到用户接口(OB或FRB)	OB 85
W#16#35A2	未装载OB (因组态而由SFC或操作系统启动)	OB 85
W#16#35A3	操作系统访问块时出错	OB 85
W#16#35A4	无法对PROFINet接口DB进行寻址	OB 85
W#16#34A4	可以再次对PROFINet接口DB进行寻址	OB 85
W#16#39B1	更新过程映像输入表时发生I/O访问出错	OB 85
W#16#39B2	将过程映像传送到输出模块时发生I/O访问出错	OB 85
W#16#39B3/38B3	更新过程映像输入表时发生I/O访问出错	OB 85
W#16#39B4/38B4	将过程映像传送到输出模块时发生I/O访问出错	OB 85
W#16#38C1	扩展机架可再次操作(1到21), 离开状态	OB 86
W#16#39C1	扩展机架故障(1到21), 进入状态	OB 86
W#16#38C2	扩展机架可再次操作,但设定值与实际组态不匹配	OB 86
W#16#39C3	分布式I/O: 主站系统故障进入状态	OB 86
W#16#39C4	分布式I/O: 站故障, 进入状态	OB 86
W#16#38C4	分布式I/O: 站故障, 离开状态	OB 86
W#16#39C5	分布式I/O: 站故障, 进入状态	OB 86
W#16#38C5	分布式I/O: 站故障, 离开状态	OB 86
W#16#38C6	扩展机架可再次操作, 但模块参数分配出错	OB 86
W#16#38C7	DP: DP: 站可再次操作, 但模块参数分配出错	OB 86
W#16#38C8	DP: DP: 站可再次操作, 但设定值和实际组态不匹配	OB 86

事件标识符	事件	OB
W#16#35D2	目前不能发送诊断条目	OB 87
W#16#35D3	不能发送同步帧	OB 87
W#16#35D4	因同步而产生非法时间跳跃	OB 87
W#16#35D5	采用同步时间时出错	OB 87
W#16#35E1	GD中的帧ID不正确	OB 87
W#16#35E2	不能在DB中输入GD包状态	OB 87
W#16#35E3	GD中的帧长度出错	OB 87
W#16#35E4	接收到非法GD包编号	OB 87
W#16#35E5	在已组态的S7连接的通信SFB中访问 DB出错	OB 87
W#16#35E6	无法在DB中输入GD所有的状态	OB 87

## 32.5 事件等级4 - 停止事件和其它模式改变

事件标识符	事件
W#16#4300	备用电源开启
W#16#4301	模式从STOP转换为STARTUP
W#16#4302	模式从STARTUP转换为RUN
W#16#4303	停止已激活的开关而导致STOP模式
W#16#4304	PG STOP操作或SFB20 “STOP”导致STOP模式
W#16#4305	HOLD: 到达断点
W#16#4306	HOLD: 退出断点
W#16#4307	PG操作启动存储器复位
W#16#4308	开关设置启动存储器复位
W#16#4309	存储器复位自动启动(加电未备份)
W#16#430A	退出HOLD, 转换为STOP
W#16#430D	由其它多值计算的CPU导致STOP模式
W#16#430E	已执行存储器复位
W#16#430F	因一个CPU的STOP模式而导致模块的STOP模式
W#16#4510	CPU数据范围发生STOP违例
W#16#4318	CiR启动
W#16#4319	CiR已完成
W#16#4520	DEFECTIVE: 无法进入STOP模式
W#16#4521	DEFECTIVE: 处理指令的处理器发生故障
W#16#4522	DEFECTIVE: 时钟芯片发生故障
W#16#4523	DEFECTIVE: 时钟脉冲发生器发生故障
W#16#4524	DEFECTIVE: 定时器更新功能发生故障
W#16#4525	DEFECTIVE: 多值计算同步失败
W#16#4926	DEFECTIVE: I/O访问监视狗发生故障
W#16#4527	DEFECTIVE: I/O访问监视发生故障
W#16#4528	DEFECTIVE: 扫描时间监视发生故障
W#16#4530	DEFECTIVE: 内部存储器的存储器测试出错
W#16#4931	STOP或DEFECTIVE: 存储器子模块的存储器测试出错
W#16#4532	DEFECTIVE: 核心资源发生故障
W#16#4933	检验和出错
W#16#4934	DEFECTIVE: 存储器不可用
W#16#4935	DEFECTIVE: 因监视狗/处理器例外而取消
W#16#4536	DEFECTIVE: 开关发生故障
W#16#4540	STOP: 内部工作存储器的内存扩展有间隙。第一个存储器扩展过小或丢失。
W#16#4541	因优先级系统而导致STOP模式
W#16#4542	因对象管理系统而导致STOP模式
W#16#4543	因测试功能而导致STOP模式
W#16#4544	因诊断系统而导致STOP模式
W#16#4545	因通信系统而导致STOP模式

事件标识符	事件
W#16#4546	因CPU存储器管理而导致STOP模式
W#16#4547	因过程映像管理而导致STOP模式
W#16#4548	因I/O管理而导致STOP模式
W#16#4949	因持续硬件中断而导致STOP模式
W#16#454A	因组态而导致STOP模式：在STARTUP过程中CPU装载了STEP 7取消选定的OB
W#16#494D	因I/O出错而导致STOP模式
W#16#494E	因电源故障而导致STOP模式
W#16#494F	因组态出错而导致STOP模式
W#16#4550	DEFECTIVE：内部系统出错
W#16#4555	无法重新启动，监视时间已过
W#16#4556	STOP：通信系统请求存储器复位
W#16#4357	模块监视狗启动
W#16#4358	一个或多个模块操作未准备就绪
W#16#4959	一个或多个模块操作准备未就绪
W#16#4562	因编程出错而导致STOP模式(OB未装载或不可用)
W#16#4563	因I/O访问出错而导致STOP模式(OB未装载或不可用)
W#16#4567	因H事件而导致STOP模式
W#16#4568	因时间出错而导致STOP模式(OB未装载或不可用)
W#16#456A	因诊断中断而导致STOP模式(OB未装载或不可用)
W#16#456B	因删除/插入模块而导致STOP模式(OB未装载或不可用)
W#16#456C	因CPU硬件出错而导致STOP模式(OB未装载或不可用，或无FRB)
W#16#456D	因程序序列出错而导致STOP模式(OB未装载或不可用)
W#16#456E	因通信出错而导致STOP模式(OB未装载或不可用)
W#16#456F	因机架故障OB而导致STOP模式(OB未装载或不可用)
W#16#4570	因过程中断而导致STOP模式(OB未装载或不可用)
W#16#4571	因嵌套堆栈出错而导致STOP模式
W#16#4572	因主控继电器堆栈出错而导致STOP模式
W#16#4573	因超出同步出错的嵌套深度而导致STOP模式
W#16#4574	因超出优先级堆栈中的中断堆栈嵌套深度而导致STOP模式
W#16#4575	因超出优先级堆栈中的块堆栈嵌套深度而导致STOP模式
W#16#4576	因分配本地数据时出错而导致STOP模式
W#16#4578	因未知操作码而导致STOP模式
W#16#457A	因代码长度出错而导致STOP模式
W#16#497C	由集成技术导致STOP模式
W#16#457B	由板载I/O上未装载DB导致STOP模式
W#16#457F	因STOP命令导致STOP模式
W#16#4580	STOP：备份缓冲区内容不一致(未转换到RUN模式)
W#16#4590	因内部功能超载而导致STOP模式
W#16#49A0	因参数分配出错或不允许的设定值和实际扩展变动而导致STOP模式：启动被封锁。
W#16#49A1	因参数分配出错而导致STOP模式：请求存储器复位
W#16#49A2	因参数修改出错而导致STOP模式：启动禁用
W#16#49A3	因参数修改出错而导致STOP模式：请求存储器复位



事件标识符	事件
W#16#49A4	STOP: 组态数据不一致
W#16#49A5	STOP: 分布式I/O: 装载的组态信息 不一致
W#16#49A6	STOP: 分布式I/O: 组态信息无效
W#16#49A7	STOP: 分布式I/O: 无组态信息
W#16#49A8	STOP: 由分布式I/O接口模块指示出错
W#16#43B0	成功更新固化程序
W#16#49B1	固化程序更新数据不正确
W#16#49B2	固化程序更新: 硬件版本与固化程序不匹配
W#16#49B3	固化程序更新: 模块类型与固化程序不匹配
W#16#43B4	固化程序保险丝异常
W#16#43B6	固化程序更新被冗余模块取消
W#16#49D0	LINK-UP因协调规则违例而中止
W#16#49D1	LINK-UP/UPDATE序列中止
W#16#49D2	链接过程中, 待机CPU因主CPU上的STOP模式变为STOP模式
W#16#43D3	待机CPU为STOP模式
W#16#49D4	主机为STOP模式, 因为伙伴CPU也是主机(链接出错)
W#16#45D5	因子PLC的CPU存储器组态不匹配而拒绝LINK-UP
W#16#45D6	因子PLC的系统程序不匹配而拒绝LINK-UP
W#16#49D7	因用户程序或组态改变而拒绝LINK-UP
W#16#45D8	DEFECTIVE: 检测到因其它出错导致的硬件故障
W#16#45D9	因SYNC模块出错而导致STOP模式
W#16#45DA	因H CPU间同步出错而导致STOP模式
W#16#43DC	转换链接过程中中断
W#16#45DD	因运行测试或其它在线功能而拒绝LINK-UP
W#16#43DE	因第n次尝试时超过监视时间而中断, 启动新的更新尝试
W#16#43DF	更新因超出完成最大尝试数后的监视时间而最终中断, 需要用户干预。
W#16#43E0	链接后从单独模式切换
W#16#43E1	更新后从链接切换
W#16#43E2	从更新变为冗余模式
W#16#43E3	主CPU: 从冗余模式变为单独模式
W#16#43E4	待机CPU: 在出错搜索模式后从冗余模式切换
W#16#43E5	待机CPU: 在链接或STOP后从出错搜索模式切换
W#16#43E6	待机CPU上中止了链接
W#16#43E7	待机CPU上中止了更新
W#16#43E8	待机CPU: 启动后从链接切换
W#16#43E9	待机CPU: 更新后从启动切换
W#16#43F1	备用主站切换
W#16#43F2	不兼容H-CPU的连接被系统程序阻塞
W#16#42F3	检测到检验和出错, 并由操作系统更正
W#16#42F4	待机CPU: 主CPU中已锁定通过SFC90的连接/更新

## 32.6 事件等级5 - 模式运行期事件

事件标识符	事件
W#16#530D	STOP模式下新的启动信息
W#16#5311	忽略模块的“未就绪”消息而启动
W#16#5545	在RUN模式下启动系统重组态
W#16#5445	在RUN模式下启动系统重组态
W#16#5380	中断和异步出错的诊断缓冲器条目禁用
W#16#5395	分布式I/O: DP主站复位
W#16#5481	所有运行系统软件的许可证再次完成。
W#16#5498	因CiR而与DP主站系统不再不一致
W#16#5581	运行系统软件的一个或几个许可证丢失。
W#16#558A	组态和已插入CPU的MLFB间的差异
W#16#558B	组态和已插入CPU的固化程序版本的差异
W#16#5598	因CiR与DP主站系统可能的不一致性的开始
W#16#5960	切换时参数分配出错
W#16#5961	参数分配出错
W#16#5962	参数分配出错阻止启动
W#16#5963	参数分配出错并请求存储器复位
W#16#5966	切换时参数分配出错
W#16#5969	参数分配出错并阻止启动
W#16#5371	分布式I/O: 与DP主站的同步结束
W#16#5979/5879	来自DP接口的诊断信息: EXTF LED打开/关闭
W#16#597C	DP全局控制命令失败或被移动
W#16#597C	DP命令“全局控制”失败或被移动
W#16#59A0	CPU中无法关联中断
W#16#59A1	集成技术中组态出错
W#16#53A2	技术固化程序下载成功
W#16#59A3	下载集成技术时出错
W#16#53A4	技术DB下载失败
W#16#55A5	版本冲突: 具有集成技术的内部接口
W#16#53FF	复位至出厂设置

## 32.7 事件等级6 - 通讯事件

事件标识符	事件
W#16#6316	启动可编程控制器时接口出错
W#16#6390	微存储卡格式化完成
W#16#6500	连接ID在模块中出现两次
W#16#6501	连接资源不足
W#16#6502	连接描述出错
W#16#6510	判断EPROM时在背景数据块中检测到CFB结构出错
W#16#6514	GD包编号在模块中出现两次
W#16#6515	组态信息中的规定长度不一致
W#16#6521	无可用的存储器子模块和内部存储器
W#16#6522	非法存储器子模块：更换子模块并复位存储器
W#16#6523	因访问子模块出错而请求存储器复位
W#16#6524	因块报头出错而请求存储器复位
W#16#6526	因存储器更换而请求存储器复位
W#16#6527	已更换存储器，因此无法重新启动
W#16#6528	STOP/HOLD模式下的对象处理功能，无法重新启动
W#16#6529	“装载用户程序”功能过程中无法启动
W#16#652A	因块在用户存储器中出现两次而不能启动
W#16#652B	因子模块的块过长而不能启动 - 更换子模块
W#16#652C	因子模块中有非法OB而不能启动
W#16#6532	因子模块中有非法组态信息而不能启动
W#16#6533	因子模块内容无效而请求存储器复位
W#16#6534	不启动：块在子模块中出现一次以上
W#16#6535	不启动：无足够内存从子模块中传送块
W#16#6536	不启动：子模块中有非法模块编号
W#16#6537	不启动：子模块中包含非法长度的块
W#16#6538	块的本地数据或写保护ID(对于DB)对于CPU非法
W#16#6539	块中有非法命令(由编译器检测到)
W#16#653A	因子模块上的本地OB数据过短而请求存储器复位
W#16#6543	不启动：块类型非法
W#16#6544	不启动：“与处理相关的”属性非法
W#16#6545	源语言非法
W#16#6546	组态信息量已达到最大值
W#16#6547	将参数分配给模块时发生参数分配出错(不在P总线上，取消下载)
W#16#6548	块检查过程中发生似真的出错
W#16#6549	块中有结构性出错
W#16#6550	在CRC中的块有出错
W#16#6551	块没有CRC
W#16#6560	SCAN溢出
W#16#6881	离开状态时发生接口出错
W#16#6905/6805	组态连接中有资源问题/已排除
W#16#6981	输入状态时发生接口出错

## 32.8 事件等级7 - H/F事件

事件标识符	事件	OB
W#16#72A2	DP主站或DP主站系统发生故障	OB 70
W#16#72A3	DP从站上的冗余恢复	OB 70
W#16#7301	CPU故障造成冗余丢失(2个中的1个)	OB 72
W#16#7302	用户触发待机STOP状态造成的冗余丢失(2个中的1个)	OB 72
W#16#7303	H系统(2个中的1个)变为冗余模式	OB 72
W#16#7323	在操作系统数据中发现偏差	OB 72
W#16#7331	因主站故障而导致待机主站切换	OB 72
W#16#7333	因运行期间系统修改而发生待机主站切换	OB 72
W#16#7334	因同步模块的通信出错而发生待机主站切换	
W#16#7335	由SFC90“H_CTRL”触发的备用站主站切换	OB 72
W#16#7340	因等待时间已到期而在用户程序中发生同步出错	OB 72
W#16#7341	因在不同的同步点等待而在用户程序中发生同步出错	OB 72
W#16#7342	因在不同的同步点等待而在操作系统中发生同步出错	OB 72
W#16#7343	因等待时间已到期而在操作系统中发生同步出错	OB 72
W#16#7344	因数据不正确而在操作系统中发生同步出错	OB 72
W#16#734A	因执行了SFC90“H_CTRL”而触发的“重新启用”作业。	OB 72
W#16#73A3	DP从站中冗余丢失	OB 70
W#16#73D8	安全模式禁用	
W#16#73E0/72E0	通讯中冗余丢失/问题已排除	OB 73
W#16#7520	RAM比较出错	OB 72
W#16#7521	过程映像输出值的比较出错	OB 72
W#16#7522	存储器位、定时器或计数器的比较出错	OB 72
W#16#73C1	更新过程已取消	OB 72
W#16#73C2	因第n次尝试期间超出监视时间而中止更新(1 n 因超过监视时间而终止后的最大可能更新尝试数目)	OB 72
W#16#75D1	安全程序：内部CPU出错	
W#16#75D2	安全程序出错：周期超时	
W#16#75D6	安全程序中的数据在输出至F I/O之前损坏	
W#16#75D7	安全程序中的数据在输出至伙伴F CPU之前损坏	
W#16#75D9	DB中有无效实型数	
W#16#75DA	安全程序：安全数据格式出错	
W#16#73DB/72DB	安全程序：安全模式已启用/已禁用	
W#16#75DC	运行期间组、内部协议出错	

事件标识符	事件	OB
W#16#75DD/74DD	安全程序：故障安全运行组关闭已启用/已禁用	
W#16#75DE/74DE	安全程序：F程序关闭已启用/已禁用	-
W#16#75DF/74DF	F程序初始化开始/结束	-
W#16#75E1	安全程序：FB“F_PLK”、“F_PLK_O”、“F_CYC_CO”、“F_TEST”或“F_TESTC”中出错	
W#16#7934	因SYNC模块连接问题切换待机主站	OB 72
W#16#7950	同步模块丢失	OB 72
W#16#7951	未上电时SYNC模块发生改变	OB 72
W#16#7952/7852	删除/插入的SYNC模块	OB 72
W#16#7953	SYNC模块的改变未复位	OB 72
W#16#7954	SYNC模块：机架号被分配了两次	OB 72
W#16#7955/7855	SYNC模块出错/已排除	OB 72
W#16#7956	SYNC模块中设置了非法机架号	OB 72
W#16#7960	冗余I/O：数字输入时偏差时间超时，出错尚未本地化	
W#16#7961	冗余I/O，数字输入出错：偏差时间过期后信号变化	
W#16#7962	冗余I/O：数字输入出错	
W#16#796F	冗余I/O：全局禁用了I/O	
W#16#7970	冗余I/O：数字输出出错	
W#16#7980	冗余I/O：模拟输入的偏差时间超时	
W#16#7981	冗余I/O：模拟输入出错	
W#16#7990	冗余I/O：模拟输出出错	
W#16#79D3/78D3	PROFIsafe与F I/O间的通讯出错	
W#16#79D4/78D4	F CPU之间与安全相关的通讯中出错	
W#16#79D5/78D5	F CPU之间与安全相关的通讯中出错	
W#16#75E2	安全程序：区域长度出错	

## 32.9 事件等级8 - 模块的诊断事件

事件标识符	事件	模块类型
W#16#8x00	模块故障/正常	ANY
W#16#8x01	内部出错	
W#16#8x02	外部出错	
W#16#8x03	通道出错	
W#16#8x04	无外部辅助电压	
W#16#8x05	无前置连接器	
W#16#8x06	无参数分配	
W#16#8x07	模块中的参数不正确	
W#16#8x30	用户子模块不正确/未找到	
W#16#8x31	通讯问题	
W#16#8x32	工作模式: RUN/STOP(STOP: 进入状态, RUN: 离开状态)	
W#16#8x33	时间监视响应(监视狗)	
W#16#8x34	内部模块电源故障	
W#16#8x35	BATTF: 电池用尽	
W#16#8x36	所有的备份失败	
W#16#8x40	扩展机架故障	
W#16#8x41	处理器故障	
W#16#8x42	EPROM出错	
W#16#8x43	RAM出错	
W#16#8x44	ADC/DAC出错	
W#16#8x45	保险丝熔断	
W#16#8x46	硬件中断丢失	
W#16#8x50	组态/参数分配出错	模拟输入
W#16#8x51	共模出错	
W#16#8x52	相位短路	
W#16#8x53	接地短路	
W#16#8x54	断线	
W#16#8x55	参考通道出错	
W#16#8x56	低于测量范围	
W#16#8x57	高于测量范围	
W#16#8x60	组态/参数分配出错	模拟输出
W#16#8x61	共模出错	
W#16#8x62	相位短路	
W#16#8x63	接地短路	
W#16#8x64	断线	
W#16#8x66	空载电压	
W#16#8x70	组态/参数分配出错	数字输入
W#16#8x71	机壳接地故障	
W#16#8x72	相位短路(传感器)	

事件标识符	事件	模块类型
W#16#8x73	接地短路(传感器)	
W#16#8x74	断线	
W#16#8x75	无传感器电源	
W#16#8x80	组态/参数分配出错	数字输出
W#16#8x81	机壳接地故障	
W#16#8x82	相位短路	
W#16#8x83	接地短路	
W#16#8x84	断线	
W#16#8x85	保险丝断开	
W#16#8x86	空载电压	
W#16#8x87	温度过高	
W#16#8xB0	计数器模块, 信号A出错	FM
W#16#8xB1	计数器模块, 信号B出错	
W#16#8xB2	计数器模块, 信号N出错	
W#16#8xB3	计数器模块, 通道间传递的值不正确	
W#16#8xB4	计数器模块, 5.2V传感器电源故障	
W#16#8xB5	计数器模块, 24V传感器电源故障	

## 32.10 事件等级9 - 标准用户事件

事件标识符	事件
W#16#9001	自动模式
W#16#9101	手动模式
W#16#9x02	OPEN/CLOSED、ON/OFF
W#16#9x03	手动命令启用
W#16#9x04	单元保护命令(OPEN/CLOSED)
W#16#9x05	过程启用
W#16#9x06	系统保护命令
W#16#9x07	过程值监视响应
W#16#9x08	调节变量监视响应
W#16#9x09	系统偏差大于允许值
W#16#9x0A	限制位置出错
W#16#9x0B	运行出错
W#16#9x0C	命令执行出错(序列发生器)
W#16#9x0D	工作状态运行 > OPEN
W#16#9x0E	工作状态运行 > CLOSED
W#16#9x0F	命令阻塞
W#16#9x11	过程状态OPEN/ON
W#16#9x12	过程状态CLOSED/OFF
W#16#9x13	过程状态的中间位置
W#16#9x14	通过AUTO将过程状态设置为ON
W#16#9x15	通过手动设置过程状态为ON
W#16#9x16	通过保护性命令将过程状态设置为ON
W#16#9x17	通过AUTO将过程状态设置为OFF
W#16#9x18	通过手动将过程状态设置为OFF
W#16#9x19	通过保护性命令将过程状态设置为OFF
W#16#9x21	接近时出现功能出错
W#16#9x22	离开时出现功能出错
W#16#9x31	执行器(DE/WE)限位OPEN
W#16#9x32	执行器(DE/WE)限位未OPEN
W#16#9x33	执行器(DE/WE)限位CLOSED
W#16#9x34	执行器(DE/WE)限位未CLOSED
W#16#9x41	非法状态, 容差时间耗尽
W#16#9x42	非法状态, 容差时间未耗尽
W#16#9x43	互锁出错, 容差时间 = 0
W#16#9x44	互锁出错, 容差时间 > 0
W#16#9x45	无反应
W#16#9x46	非法退出最终状态, 容差时间 = 0
W#16#9x47	非法退出最终状态, 容差时间 > 0
W#16#9x50	信号范围上限USR



事件标识符	事件
W#16#9x51	测量范围上限UMR
W#16#9x52	信号范围下限LSR
W#16#9x53	测量范围下限LMR
W#16#9x54	报警上限UAL
W#16#9x55	警告上限UWL
W#16#9x56	容差上限UTL
W#16#9x57	容差下限LTL
W#16#9x58	警告下限LWL
W#16#9x59	报警下限LAL
W#16#9x60	GRAPH7步骤进入/离开
W#16#9x61	GRAPH7互锁出错
W#16#9x62	GRAPH7执行出错
W#16#9x63	GRAPH7出错已通知
W#16#9x64	GRAPH7出错已确认
W#16#9x70	趋势在正方向超出
W#16#9x71	趋势在负方向超出
W#16#9x72	无反应
W#16#9x73	非法退出最终状态
W#16#9x80	超过限定值, 容差时间 = 0
W#16#9x81	超过限定值, 容差时间 > 0
W#16#9x82	低于限定值, 容差时间 = 0
W#16#9x83	低于限定值, 容差时间 > 0
W#16#9x84	梯度超出, 容差时间 = 0
W#16#9x85	梯度超出, 容差时间 > 0
W#16#9x86	低于梯度, 容差时间 = 0
W#16#9x87	低于梯度, 容差时间 > 0
W#16#9190/9090	用户参数分配出错进入/离开
W#16#91F0	溢出
W#16#91F1	下溢
W#16#91F2	除以0
W#16#91F3	计算操作非法

### 32.11 事件等级A和B - 自由用户事件

事件标识符	事件
W#16#Axyz	用户可用的事件
W#16#Bxyz	

## 32.12 保留的事件等级

### 保留

以下的事件等级将被保留，供将来扩展：

- C
- D
- E
- F 为集中机架外的模块保留(例如，CP或FM)

## 33 SFC和SFB列表

### 33.1 按编号排序的SFC列表

编号	短名称	功能
SFC 0	SET_CLK	设置系统时钟
SFC1	READ_CLK	读系统时钟
SFC2	SET_RTM	设置运行系统计时器
SFC3	CTRL_RTM	启动/停止运行系统计时器
SFC4	READ_RTM	读运行系统计时器
SFC5	GADR_LGC	查询通道的逻辑地址
SFC6	RD_SINFO	读OB的启动信息
SFC7	DP_PRAL	在DP主站上触发硬件中断
SFC9	EN_MSG	启用与块相关、符号相关及组状态消息
SFC10	DIS_MSG	禁用与块相关、符号相关及组状态消息
SFC11	DPSYC_FR	同步DP从站组
SFC12	D_ACT_DP	取消激活和激活DP从站
SFC13	DPNRM_DG	读取DP从站的诊断数据(从站诊断)
SFC14	DPRD_DAT	从标准DP从站读取一致性数据
SFC15	DPWR_DAT	将连续数据写入DP标准从站
SFC17	ALARM_SQ	生成可确认的与块相关的消息
SFC18	ALARM_S	生成永久确认的与块相关的消息
SFC19	ALARM_SC	查询上一个ALARM_SQ输入状态消息的确认状态
SFC20	BLKMOV	复制变量
SFC21	FILL	初始化存储区域
SFC22	CREAT_DB	创建数据块
SFC23	DEL_DB	删除数据块
SFC24	TEST_DB	测试数据块
SFC25	COMPRESS	压缩用户存储器
SFC26	UPDAT_PI	更新过程映像更新表
SFC27	UPDAT_PO	更新过程映像输出表
SFC28	SET_TINT	设置日时钟中断
SFC29	CAN_TINT	取消时钟中断
SFC30	ACT_TINT	激活时钟中断
SFC31	QRY_TINT	查询日时钟中断
SFC32	SRT_DINT	启动延时中断

编号	短名称	功能
SFC33	CAN_DINT	取消延时中断
SFC34	QRY_DINT	查询日时钟中断
SFC35	MP_ALM	触发多处理器中断
SFC36	MSK_FLT	屏蔽同步出错
SFC37	DMSK_FLT	取消屏蔽同步出错
SFC38	READ_ERR	读出错寄存器
SFC39	DIS_IRT	禁用新的中断和异步出错
SFC40	EN_IRT	启用新的中断和异步出错
SFC41	DIS_AIRT	延迟高优先级的中断和异步出错
SFC42	EN_AIRT	启用高优先级的中断和异步出错
SFC43	RE_TRIGR	再触发循环时间监控
SFC44	REPL_VAL	将替换值传送到累加器1
SFC46	STP	将CPU切换为STOP
SFC47	WAIT	延时用户程序执行
SFC48	SNC_RTCB	同步从站时钟
SFC49	LGC_GADR	查询属于一个逻辑地址的模块插槽
SFC50	RD_LGADR	查询模块的所有逻辑地址
SFC51	RDSYSST	读系统状态列表或部分列表
SFC52	WR_USMSG	将用户自定义的诊断事件写入诊断缓冲区
SFC54	RD_PARM	读已定义的参数
SFC55	WR_PARM	写动态参数
SFC56	WR_DPARM	写一条数据记录
SFC57	PARM_MOD	为一个模块分配参数
SFC58	WR_REC	写一条数据记录
SFC59	RD_REC	读一条数据记录
SFC60	GD_SND	发送一个GD信息包
SFC61	GD_RCV	获取一个收到的GD信息包
SFC62	CONTROL	查询属于一个通讯SFB实例的连接状态
SFC63	AB_CALL	组合代码块
SFC64	TIME_TCK	读系统时间
SFC65	X_SEND	将数据发送到本地S7站外部的通讯伙伴
SFC66	X_RCV	接收来自本地S7站外部的通讯伙伴的数据
SFC67	X_GET	读来自本地S7站外部的通讯伙伴的数据
SFC68	X_PUT	将数据写入本地S7站外部的通讯伙伴
SFC69	X_ABORT	中止与本地S7站外部的通讯伙伴之间的现有连接
SFC72	I_GET	读来自本地S7站内的通讯伙伴的数据
SFC73	I_PUT	将数据写入本地S7站内的通讯伙伴
SFC74	I_ABORT	中止与本地S7站内的通讯伙伴之间的现有连接
SFC78	OB_RT	确定OB程序运行时间
SFC79	SET	设置输出范围
SFC80	RSET	重置输出范围

编号	短名称	功能
SFC81	UBLKMOV	不可中断的块移动
SFC82	CREA_DBL	在装载存储器中创建数据块
SFC83	READ_DBL	从装载存储器中的数据块读
SFC84	WRIT_DBL	从装载存储器中的数据块写
SFC85	CREA_DB	创建数据块
SFC87	C_DIAG	实际连接状态的诊断
SFC90	H_CTRL	控制H系统中的操作
SFC100	SET_CLKS	设置日时钟和TOD状态
SFC101	RTM	处理运行系统计时器
SFC102	RD_DPARA	重新定义的参数
SFC103	DP_TOPOL	标识DP主站系统中的总线拓扑
SFC104	CiR	控制CiR
SFC105	READ_SI	读动态系统资源
SFC106	DEL_SI	删除动态系统资源
SFC107	ALARM_DQ	生成始终可确认的与块相关的消息
SFC108	ALARM_D	生成始终可确认的与块相关的消息
SFC112	PN_IN	更新PROFInet组件的用户程序接口中的输入
SFC113	PN_OUT	更新PROFInet组件的用户程序接口中的输出
SFC114	PN_DP	更新DP互连
SFC126	SYNC_PI	以同步循环的方式更新过程映像分区输入表
SFC127	SYNC_PO	以同步循环的方式更新过程映像分区输出表

\* 只有CPU 614具有SFC63 “AB\_CALL”。关于详细描述，请参见相应的手册。

## 33.2 按字母排序的SFC列表

短名称	编号	功能
AB_CALL	SFC63	组合代码块
ACT_TINT	SFC30	激活时钟中断
ALARM_D	SFC108	生成永久可确认的与块相关的消息
ALARM_DQ	SFC107	生成永久可确认的与块相关的消息
ALARM_S	SFC18	生成永久确认的与块相关的消息
ALARM_SC	SFC19	查询上一个ALARM_SQ输入状态消息的确认状态
ALARM_SQ	SFC17	生成可确认的与块相关的消息
BLKMOV	SFC20	复制变量
C_DIAG	SFC87	实际连接状态的诊断
CAN_DINT	SFC33	取消延时中断
CAN_TINT	SFC29	取消时钟中断
CiR	SFC104	控制CiR
COMPRESS	SFC25	压缩用户存储器
CONTROL	SFC62	查询属于一个通讯SFB实例的连接状态
CREA_DB	SFC85	创建数据块
CREA_DBL	SFC82	在装载存储器中生成数据块
CREAT_DB	SFC22	创建数据块
CTRL_RTM	SFC3	启动/停止运行系统计时器
D_ACT_DP	SFC12	取消激活和激活DP从站
DEL_DB	SFC23	删除数据块
DEL_SI	SFC106	删除动态占用的系统资源
DIS_AIRT	SFC41	延迟高优先级的中断和异步出错
DIS_IRT	SFC39	禁用新的中断和异步出错
DIS_MSG	SFC10	禁用与块相关、符号相关及组状态消息
DMSK_FLT	SFC37	取消屏蔽同步出错
DP_PRAL	SFC7	在DP主站上触发硬件中断
DP_TOPOL	SFC103	标识DP主站系统中的总线拓扑
DPNRM_DG	SFC13	读取DP从站的诊断数据(从站诊断)
DPRD_DAT	SFC14	从标准DP从站读取一致性数据
DPSYC_FR	SFC11	同步DP从站组
DPWR_DAT	SFC15	将连续数据写入DP标准从站
EN_AIRT	SFC42	启用高优先级的中断和异步出错
EN_IRT	SFC40	启用新的中断和异步出错
EN_MSG	SFC9	启用与块相关、符号相关及组状态消息
FILL	SFC21	初始化存储区域
GADR_LGC	SFC5	查询通道的逻辑地址
GD_RCV	SFC61	获取一个收到的GD信息包
GD_SND	SFC60	发送一个GD信息包
H_CTRL	SFC90	控制H系统中的操作

短名称	编号	功能
I_ABORT	SFC74	中止与本地S7站内的通讯伙伴之间的现有连接
I_GET	SFC72	读来自本地S7站内的通讯伙伴的数据
I_PUT	SFC73	将数据写入本地S7站内的通讯伙伴
LGC_GADR	SFC49	查询属于一个逻辑地址的模块插槽
MP_ALM	SFC35	触发多处理器中断
MSK_FLT	SFC36	屏蔽同步出错
PARM_MOD	SFC57	为一个模块分配参数
PN_DP	SFC114	更新DP互连
PN_IN	SFC112	更新PROFINet组件的用户程序接口中的输入
PN_OUT	SFC113	更新PROFINet组件的用户程序接口中的输出
QRY_DINT	SFC34	查询延时中断
QRY_TINT	SFC31	查询日时钟中断
RD_DPARA	SFC102	重新定义的参数
RD_LGADR	SFC50	查询模块的所有逻辑地址
RD_PARM	SFC54	读已定义的参数
RD_REC	SFC59	读一个数据记录
RD_SINFO	SFC6	读OB的启动信息
RDSYSST	SFC51	读系统状态列表或部分列表
RE_TRIGR	SFC43	再触发循环时间监控
READ_CLK	SFC1	读系统时钟
READ_DBL	SFC83	SFC83
READ_ERR	SFC38	读出错寄存器
READ_RTM	SFC4	读运行系统计时器
READ_SI	SFC105	读动态占用的系统资源
REPL_VAL	SFC44	将替换值传送到累加器1
RSET	SFC80	重置输出范围
RTM	SFC101	处理运行系统计时器
SET	SFC79	设置输出范围
SET_CLK	SFC0	设置系统时钟
SET_CLKS	SFC100	设置日时钟和TOD状态
SET_RTM	SFC2	设置运行系统计时器
SET_TINT	SFC28	设置日时钟中断
SNC_RTCB	SFC48	同步从站时钟
SRT_DINT	SFC32	启动延时中断
STP	SFC46	将CPU切换为STOP
SYNC_PI	SFC126	以同步循环的方式更新过程映像分区输入表
SYNC_PO	SFC127	以同步循环的方式更新过程映像分区输出表
TEST_DB	SFC24	测试数据块
TIME_TCK	SFC64	读系统日时钟
UBLKMOV	SFC81	不可中断的块移动
UPDAT_PI	SFC26	更新过程映像更新表
UPDAT_PO	SFC27	更新过程映像输出表

短名称	编号	功能
WAIT	SFC47	延时用户程序执行
WR_DPARM	SFC56	写一条数据记录
WR_PARM	SFC55	写动态参数
WR_REC	SFC58	写入数据记录
WR_USMSG	SFC52	将用户自定义的诊断事件写入诊断缓冲区
WRIT_DBL	SFC84	从装载存储器中的数据块写
X_ABORT	SFC69	中止与本地S7站外部的通讯伙伴之间的现有连接
X_GET	SFC67	读来自本地S7站外部的通讯伙伴的数据
X_PUT	SFC68	将数据写入本地S7站外部的通讯伙伴
X_RCV	SFC66	接收来自本地S7站外部的通讯伙伴的数据
X_SEND	SFC65	将数据发送到本地S7站外部的通讯伙伴

\* 只有CPU 614存在SFC63“AB\_CALL”。关于详细描述，请参见相应的手册。



### 33.3 按编号排序的SFB列表

编号	短名称	功能
SFB 0	CTU	递增计数器
SFB1	CTD	递减计数器
SFB2	CTUD	递增/递减计数器
SFB3	TP	生成脉冲
SFB4	TON	生成接通延迟
SFB5	TOF	生成断开延迟
SFB8	USEND	数据的不对等的发送
SFB9	URCV	数据的不对等的接收
SFB12	BSEND	发送分段数据
SFB13	BRCV	接收分段数据
SFB14	GET	从远程CPU读数据
SFB15	PUT	将数据写入远程CPU
SFB16	PRINT	将数据发送到打印机
SFB19	START	在远程设备上开始一个暖重启或冷重启
SFB20	STOP	将远程设备切换为STOP状态
SFB21	RESUME	在远程设备上开始一个热重启
SFB22	STATUS	查询远程伙伴的状态
SFB23	USTATUS	接收远程设备的状态
SFB29	HS_COUNT*	计数器(高速计数器, 集成功能)
SFB30	FREQ_MES*	频率计(频率计, 集成功能)
SFB31	NOTIFY_8P	生成没有确认指示的与块相关的消息
SFB32	DRUM	执行序列发生器
SFB33	ALARM	生成具有确认显示的与块相关的消息
SFB34	ALARM_8	生成与块相关的消息, 其中没有用于表示8个信号的值
SFB35	ALARM_8P	生成与块相关的消息, 其中带有用于表示8个信号的值
SFB36	NOTIFY	生成没有确认显示的与块相关的消息
SFB37	AR_SEND	发送归档数据
SFB38	HSC_A_B*	计数器A/B(集成功能)
SFB39	POS*	定位(集成功能)
SFB41	CONT_C <sup>1)</sup>	连续控制
SFB42	CONT_S <sup>1)</sup>	步进控制
SFB43	PULSEGEN <sup>1)</sup>	脉冲生成
SFB44	ANALOG <sup>2)</sup>	使用模拟量输出定位
SFB46	DIGITAL <sup>2)</sup>	使用数字量输出定位
SFB47	COUNT <sup>2)</sup>	控制计数器
SFB48	FREQUENC <sup>2)</sup>	控制频率测量
SFB49	PULSE <sup>2)</sup>	控制脉宽调制
SFB52	RDREC	从DP从站读一个数据记录
SFB53	WRREC	将数据记录写入DP从站

编号	短名称	功能
SFB54	RALRM	从DP从站接收中断
SFB60	SEND_PTP <sup>2)</sup>	发送数据(ASCII、3964(R))
SFB61	RECV_PTP <sup>2)</sup>	接收数据(ASCII、3964(R))
SFB62	RES_RECV <sup>2)</sup>	删除接收缓冲区(ASCII、3964(R))
SFB63	SEND_RK <sup>2)</sup>	发送数据(RK 512)
SFB64	FETCH_RK <sup>2)</sup>	获取数据(RK 512)
SFB65	SERVE_RK <sup>2)</sup>	接收和提供数据(RK 512)
SFB75	SALRM	将中断发送到DP主站

- \* SFB29 “HS\_COUNT”和SFB30 “FREQ\_MES”仅存在于CPU 312 IFM和CPU 314 IFM上。  
SFB 38 “HSC\_A\_B”和SFB 39 “POS”仅存在于CPU 314 IFM上。关于详细描述，请参见/73/。
- 1) SFB 41 “CONT\_C”、SFB 42 “CONT\_S”和SFB 43 “PULSEGEN”仅存在于CPU 314 IFM上。
- 2) SFB 44到49及60到65仅存在于S7-300C CPU上。

### 33.4 按字母排序的SFB列表

短名称	编号	功能
ALARM	SFB33	生成具有确认的与块相关的消息
ALARM_8	SFB34	生成与块相关的消息，其中没有用于表示8个信号的值
ALARM_8P	SFB35	生成与块相关的消息，其中带有用于表示8个信号的值
ANALOG	SFB44 <sup>2)</sup>	使用模拟量输出定位
AR_SEND	SFB37	发送归档数据
BRCV	SFB13	接收分段数据
BSEND	SFB12	发送分段数据
CONT_C <sup>1)</sup>	SFB41	连续控制
CONT_S <sup>1)</sup>	SFB42	步进控制
COUNT	SFB47 <sup>2)</sup>	控制计数器
CTD	SFB1	递减计数器
CTU	SFB0	递增计数器
CTUD	SFB2	递增/递减计数器
DIGITAL	SFB46 <sup>2)</sup>	使用数字量输出定位
DRUM	SFB32	执行序列发生器
FETCH_RK	SFB64	接收数据(RK 512)
FREQ_MES*	SFB30	频率计(频率计, 集成功能)
FREQUENC	SFB48 <sup>2)</sup>	控制频率测量
GET	SFB14	从远程CPU读数据
HSC_A_B*	SFB38	计数器A/B(集成功能)
HS_COUNT*	SFB29	计数器(高速计数器, 集成功能)
NOTIFY	SFB36	生成没有确认显示的与块相关的消息
NOTIFY_8P	SFB31	生成没有确认指示的与块相关的消息
POS*	SFB39	定位(集成功能)
PRINT	SFB16	将数据发送到打印机
PULSE	SFB49 <sup>2)</sup>	控制脉宽调制
PULSEGEN <sup>1)</sup>	SFB43	脉冲生成
PUT	SFB15	将数据写入远程CPU
RALRM	SFB54	从DP从站接收中断
RDREC	SFB52	从DP从站读一个数据记录
RECV_PTP	SFB61 <sup>2)</sup>	接收数据(ASCII、3964(R))
RES_RECV	SFB62 <sup>2)</sup>	删除接收缓冲区(ASCII、3964(R))
RESUME	SFB21	在远程设备上开始一个热重启
SALRM	SFB75	将中断发送到DP主站
SEND_PTP	SFB60 <sup>2)</sup>	发送数据(ASCII、3964(R))
SEND_RK	SFB63 <sup>2)</sup>	发送数据(RK 512)
SERVE_RK	SFB65 <sup>2)</sup>	接收和提供数据(RK 512)
START	SFB19	在远程设备上开始一个暖重启或冷重启
STATUS	SFB22	查询远程伙伴的状态

短名称	编号	功能
STOP	SFB20	将远程设备切换为STOP状态
TOF	SFB5	生成断开延迟
TON	SFB4	生成接通延迟
TP	SFB3	生成脉冲
URCV	SFB9	数据的不对等的接收
USEND	SFB8	数据的不对等的发送
USTATUS	SFB23	接收远程设备的状态

- \*) 只有CPU 312 IFM和CPU 314 IFM存在SFB 29 “HS\_COUNT”和SFB 30 “FREQ\_MES”。  
SFB 38 “HSC\_A\_B”和SFB 39 “POS”仅存在于CPU 314 IFM上。关于详细信息，请参见/73/。
- 1) SFB 41 “CONT\_C”、SFB 42 “CONT\_S”和SFB 43 “PULSEGEN”仅存在于CPU 314 IFM上。
  - 2) SFB 44到49及60到65仅存在于S7-300C CPU上。

## 参考书目

- /30/使用入门： 使用STEP 7 V5.3
- /70/手册：“S7-300自动化系统CPU规范：  
CPU 31xC与CPU 31x”和  
“S7-300自动化系统CPU规范： CPU 312IFM - 318-2 DP”
- /71/参考手册： S7-300、M7-300可编程控制器  
模块规范
- /72/指令列表： S7-300可编程控制器
- /101/参考手册： 自动化系统S7-400： CPU规范
- /102/指令列表： S7-400可编程控制器
- /231/手册： 组态硬件和通讯连接，STEP 7 V5.3版本
- /232/参考手册： S7-300和S7-400的语句表(STL)
- /233/参考手册： S7-300和S7-400的梯形图(LAD)
- /234/手册： 使用STEP 7 V5.3编程
- /236/参考手册： S7-300和S7-400的功能块图(FBD)
- /250/手册： 用于S7-300和S7-400编程的  
结构控制语言(SCL)
- /251/手册： S7-300和S7-400的S7-GRAPH，  
顺序控制系统编程
- /252/手册： S7-300和S7-400的S7-HiGraph，  
状态图编程
- /254/手册： 用于S7和M7的连续功能图(CFC)  
连续功能图编程
- /270/手册： S7-300和S7-400的S7-PDIAG  
“组态LAD、STL和FBD的过程诊断”
- /350/用户手册： SIMATIC 7，  
标准控制器



# 词汇表

## 字母

### ACCU(累加器)

累加器为 CPU 中的寄存器，可充当用于装载和传送操作的缓冲区，还可用于比较、数学运算以及转换操作。

### CPU 操作系统

CPU 操作系统组织未链接到特定控制任务的 CPU 的所有功能和过程。

### OB1

组织块 OB1 是用于循环程序处理的系统程序的用户接口。

### OB 优先级

CPU 的操作系统可区分各种优先级，例如，循环程序处理、硬件中断控制的程序处理。组织块(OB)被分配给每个优先级，S7 用户可在其中对反应进行编程。OB 具有不同的优先级，这样就可以在同时出现两个 OB 时能够以正确的顺序对其进行处理，并允许优先级高的 OB 中断那些优先级低的 OB。S7 用户可以更改标准优先级。

### P 算法

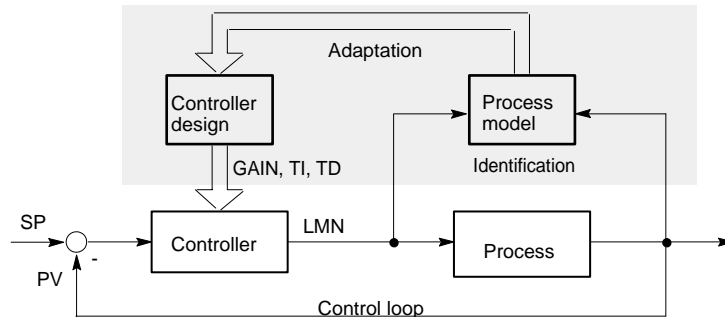
用于计算输出信号的算法，其中错误信号与受控变量更改之间存在比例关系。  
特征： 稳定状态错误信号，不用于包括停滞时间在内的过程。

### PI 算法

用于计算输出信号的算法，其中，受控变量的变化由一个与错误信号成比例的分量，以及一个与错误信号和时间成比例的 I 分量组成。特征： 无稳定状态错误信号，同 I 算法相比补偿更快，适用于所有过程。

## PID Algorithm

用于计算输出信号的算法，由错误信号的乘运算、积分和微分组成。PID 算法为并行结构。特征：如果过程的停滞时间不大于其它时间常数，即可获得高级的控制质量。



## Pulse Duration Modulation

脉冲持续时间调制是在不连续输出位置影响受控变量的一种方法。以百分比形式计算的受控值在受控变量输出位置会被转换为比例信号脉冲时间  $T_p$ ，例如， $100\% T_p = T_A$  或者 = CYCLE。

## SCAN

CPU 中集成的一项操作系统功能，用于以设定的时间间隔来扫描和检测信号以便确定是否发生了信号更改。

## STEP 7

用于为 SIMATIC S7 控制器创建用户程序的编程软件。

### STEP 7 编程语言

SIMATIC S7 控制器的编程语言。S7 程序员能够以不同表示类型来使用 STEP 7：a) 语句表、b) 控制系统流程图、c) 梯形图逻辑。

## STL

语句表。



## B

### 闭环控制

闭环控制器是一种设备，在该设备中会不断地计算错误信号，并生成激励信号以便迅速地消除错误信号且不发生过冲。

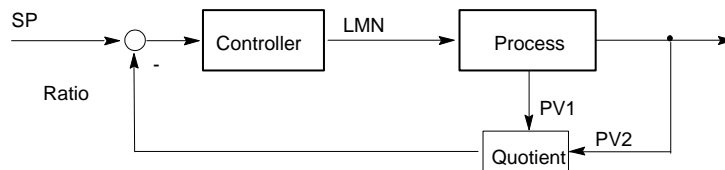
### 比例执行机构

脉冲持续时间调制

### 比率控制

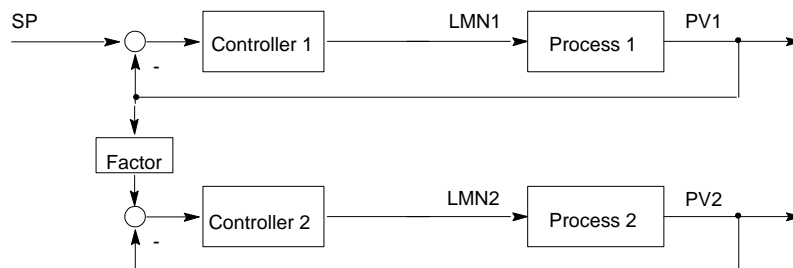
- 单回路比率控制器

当两个过程变量的比率比这两个变量的绝对值更重要时，将使用单回路比率控制器。



- 多回路比率控制器

在多回路比率控制器中，两个过程变量  $PV1$  和  $PV2$  的比率必须为常数。为此，将从第一个控制回路的过程变量来计算第二个控制回路的设定值。即使过程变量  $PV1$  发生动态更改，该比率也将保持不变。



## 变量

变量定义一种可在 **STEP 7** 用户程序中使用的、具有可变内容的数据。变量由地址(例如, **M 3.1**)和数据类型(例如, **BOOL**)组成, 并用符号表示(例如, **MOTOR\_ON**)。

## 变量声明

变量声明包括符号名、数据类型条目, 可能还会有缺省值、地址和注释条目。

## 标准功能

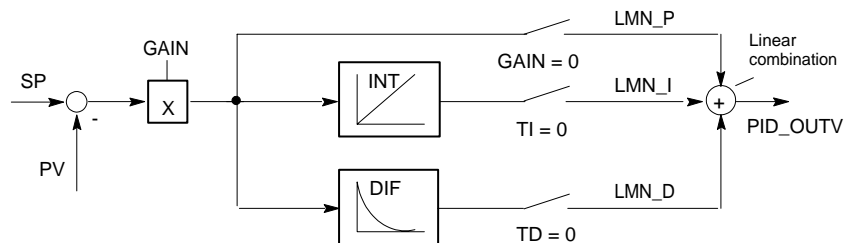
标准功能是指可从 **SIEMENS** 获得的功能块, 用于执行复杂任务。

## 标准功能块

标准功能块是指可从 **SIEMENS** 获得的功能块, 用于执行复杂任务。

## 并行结构

并行结构是控制器中一种特殊类型的信号处理(数学处理)方式。**P**、**I** 和 **D** 组件以相互并行的方式进行计算(彼此间没有任何交互), 然后再进行总计。



## 步骤控制器

步骤控制器是具有不连续输出的准连续控制器(以及具有 **I** 操作的机动执行机构)。该执行机构具有三步式响应, 例如, 向上 - 停止 - 向下(或者打开 - 暂停 - 关闭)。(三步控制器)。

## C

### 参数

1. 参数是指 S7 逻辑块的变量

(请参见块参数、实际参数、形式参数)

2. 用于设置模块特性的变量

(每个模块具有一个或多个)

每个可组态模块在出厂时均具有基本的参数设置，不过可使用 STEP 7 来更改此设置。

(每个模块具有一个或多个)。

有两种类型的参数：

静态和动态参数，参数，静态/参数，动态。

### 参数，动态

与静态参数不同，模块的动态参数可在操作期间由用户程序通过调用 SFC (例如，模拟模块的限定值)进行更改。

### 参数，静态

与动态参数不同，模块的静态参数不能由用户程序更改而只能使用 STEP 7 进行更改，例如，数字输入模块的输入延迟。

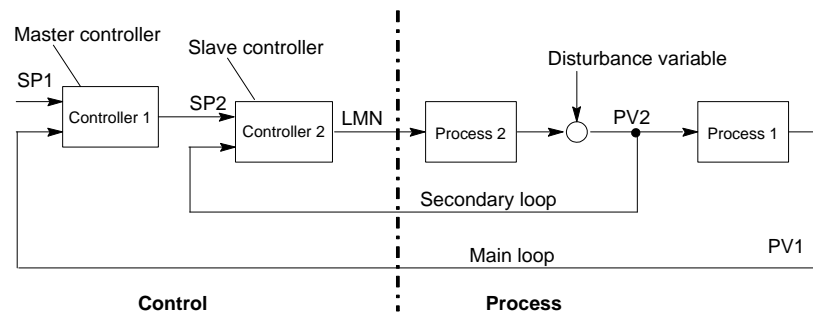
### 常数

“常数”为逻辑块中常数值标记值的标记值。常数用于提高程序的可读性。例如，不直接输入值(例如 10)，而是在功能块中输入标记值“Max\_iteration\_count”。当调用块时，再输入该常数的值(例如 10)。

### 层叠控制

层叠控制涉及一系列互连的控制器，其中主控制器根据主过程变量的即时错误信号来调整二级(从属)控制器的设定值。

可通过包含其它的过程变量来改进层叠控制系统。在适当的地点测量二级过程变量 PV2，并对其进行控制，使其接近参考设定值(主控制器的输出 SP2)进行控制。主控制器将过程变量 PV1 控制为固定设定值 SP1，并对 SP2 进行设置以便在不发生过冲的情况下尽快达到预期目标。



## 重新启动

启动 CPU 时(例如, 当模式选择器从 STOP 切换到 RUN 或当接通电源时), 在循环程序处理开始之前(OB1), 首先处理组织块 OB100(重新启动)或组织块 OB101(重新启动; 只在 S7-400 中)或组织块 OB 102(完全重新启动)(完全重新启动)。重新启动时, 将读入过程映像输入表, 并且会在 STEP 7 用户程序处理被上一次停止(STOP, 断电)中断的位置重新启动它。

## 处理 OB 时的错误

如果系统程序识别出一特定错误(例如, S7 中的访问错误), 它将会调用指定的组织块, 在该组织块中 CPU 对错误的响应可由用户程序来设置。

## 错误(系统错误)

系统错误是指可能在 PLC 内部(不在过程中)发生的错误。例如, 系统错误可以是 CPU 中的程序错误和模块中的故障。

## 错误 OB

错误 OB 是指用户可用来对错误的反应进行编程的组织块。不过, 只有在错误未导致 PLC 停止的情况下才可以对错误的反应编程。每种错误类型均有一个错误 OB。(例如, S7 中寻址错误的错误 OB、访问错误的错误 OB。)

## 错误反应

对运行错误的反应。操作系统可按以下方式作出反应: 将 PLC 更改为 STOP 状态、调用用户可在其中对反应进行编程的组织块, 或者显示错误。

## D

## 单向通讯

当使用通讯 SFB 交换数据时, 单向通讯与双向通讯之间存在区别。当只有本地模块上存在 SFB 时, 通讯是单向的, 例如, SFB "GET"。

## 地址

地址是指赋予某个内存位置或一系列内存位置的标识符，例如：输入 I 12.1；位存储器 MW25；数据块 DB3。

## F

### 符号编程

STEP 7 编程语言允许使用符号名称代替 STEP 7 地址。例如，可将 STEP 7 地址“Q 1.1”替换为“Valve 17”。

STEP 7 中的符号表还会在地址和所分配的符号名称之间创建一个链接。

## G

### 工具

用于组态和编程的软件特征部件。

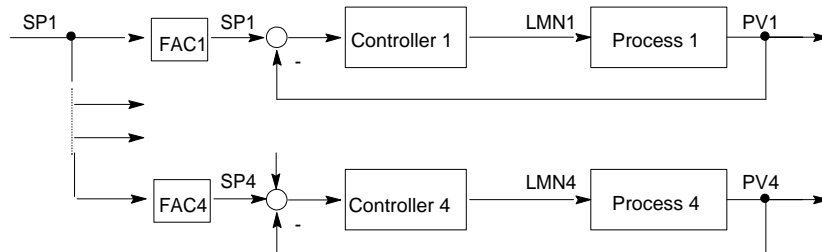
### 故障代码(DB)

故障代码是指用户程序中包含用户数据的区域。有所有逻辑块均可对其进行访问的共享故障代码，还有同特定功能块(FB)调用有关的情景故障代码。

## H

### 混合控制

混合控制涉及一个控制器结构，在该结构中，总量 SP 的设定值被转换为各个组件组件百分比。混合因子 FAC 的合计必须为 1(= 100%)。



## J

### 集成控制器

集成控制器是操作系统中经过预先编程的控制器块变量，包含闭环控制应用程序的最重要功能。用户可使用软件开关来选择或取消选择功能。

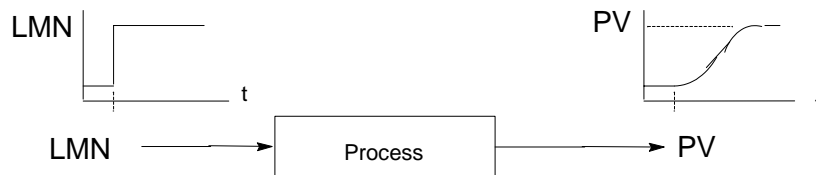
### 积分组件

控制器的积分组件。

在过程变量(或错误信号)发生阶跃变化后，输出变量将以斜坡函数随时间进行变化，并且变化的比率与积分作用因子  $KI(= 1/TI)$  成比例。闭合控制回路中的积分组件具有校正控制器输出变量的作用，直到错误信号变成零为止。

### 进程

过程是系统的一部分，其中，过程变量受到受控变量的影响(通过更改能量或质量的级别)。可将过程分为执行机构和被控制的实际过程。



## K

### 控制回路

控制回路是指过程输出(过程变量)与控制器输入之间以及控制器输出(调节变量)与过程输入之间的连接，这样，控制器和过程便形成了一个闭合回路。

### 控制器参数

控制器参数是指对给定回路或过程特征的控制器响应进行静态或动态改变的特征值。

### 控制设备

用于确定过程变量的整套设备。该设备由控制器、受控设备(例如，执行机构)以及传感器(测量设备)组成。

## 控制系统组消息

当标准诊断事件被输入诊断缓冲区中时由 CPU 操作系统生成的组消息。

## L

### 两步控制器

两步控制器是只能为受控变量设置两种状态(例如, 开 - 关)的控制器。

### 连续控制器

连续控制器是控制器中的一种, 在该控制器中, 错误信号的每一变化均会引起调节变量的变化。这样便可采用调节变量范围内的任意值。

### 逻辑块

在 SIMATIC S7 中, 逻辑块是指包含部分 STEP 7 用户程序的块。其它类型的块是仅包含数据的故障代码。下表给出了逻辑块的类型:

- 组织块(OB)
- 功能块(FB)
- 功能(FC)
- 系统功能块(SFB)
- 系统功能(SFC)

### 逻辑运算结果(RLO)

逻辑运算结果(RLO)是指处理器中的当前信号状态, 用于进一步的二进制信号处理。上一 RLO 的信号状态将决定是否执行某些运算。

## M

### 模块参数

模块参数是指可用于设置模块特性的值。可在用户程序中修改其中的某些参数, 这取决于特定的模块。

## Q

### 启动 OB

依据启动模式选择器的设置(只对 S7-400)启动原因(掉电后恢复供电、使用模式选择器手动由 STOP 切换为 RUN 或者从编程设备执行命令),操作系统将会调用启动组织块“完全重新启动”或“重新启动”(仅存在于 S7-400 中)。例如,在启动 OB 中,SIMATIC S7 用户能够对系统在掉电后再次启动的方式进行编程。

指令(STEP 5 或 STEP 7)是用文本语言所创建的程序中最小的部分。它表示处理器的命令。

### 启动事件

启动事件是指诸如错误或中断之类的定义事件,用于提示操作系统调用相应的组织块。

### 启动事件信息

启动事件信息是组织块(OB)的一部分。启动事件信息为 S7 用户提供了与触发调用 OB 的事件有关的详细信息。启动事件信息包含事件编号(由事件等级和事件 ID 组成)、事件时间标志和其它信息(例如,中断激活信号模块的地址)。

### 启动信息

当操作系统调用组织块时,操作系统将传送可在用户程序中进行解释的启动信息。

## S

### 三步控制器

只能采用三种离散状态的控制器:例如,“加热 - 断开 - 冷却”或者“右 - 停止 - 左”(步骤控制器)。

### 实际参数

当调用功能块(FB)或功能(FC)时,实际参数将替换形式参数,例如,形式参数“REQ”将被替换为实际参数“1 3.6。”



## 事件控制的程序执行

对于事件控制的程序执行，循环用户程序的运行将由启动事件(优先级)中断。如果发生启动事件，则会在调用和执行下一条指令和已分配组织块之前中断当前正在执行的块。然后循环程序执行将从中断点处继续。

## 时钟中断

时钟中断属于 **SIMATIC S7** 程序执行中的一种优先级。它在指定日期(或天)和时间(例如，9:50 或每小时或每分钟)生成。随后将执行相应的组织块。

## 输入参数

输入参数仅存在于功能和功能模块中。借助于输入参数，数据被传送到调用块中进行处理。

## 双向通讯

当使用通讯 **SFB** 交换数据时，单向通讯与双向通讯之间存在区别。当在本地模块和远程模块上存在 **SFB** 时，通讯是双向的，例如，通讯 **SFB "USEND"**和"**URCV**"。

## 随附值

生成消息时可随该消息以及与变量或地址状态有关的给定信息一起输出的值。

## T

### 同步错误

同步错误是指分配给用户程序中特定位置的运行错误(例如，访问 **I/O** 模块时的错误)。当发生这些错误时，操作系统将调用相应的组织块，用户可在这些组织块中对反应进行编程。

## X

### 系统功能(**SFC**)

系统功能(**SFC**)是集成在 **CPU** 操作系统中的功能，并可根据需要在 **STEP 7** 用户程序中调用此项功能。

### 系统功能块(SFB)

系统功能块(SFB)是集成在 CPU 操作系统中的功能块，可在需要在 STEP 7 用户程序中调用此功能块。

### 系统诊断

检测和评估系统诊断事件。

### 系统诊断事件

在 CPU 的诊断缓冲区中生成并用于初始化操作系统的条目。

### 显示设备

用于显示过程结果的设备。

### 消息

事件发生的报告。消息可以输出到一个有合适组态的显示设备，包含诸如优先级、位置和消息事件时间之类的信息，以及关于状态转变(进入状态/离开状态)的信息。

### 消息号

为消息所分配并用于标识此消息(例如，用于确认)的唯一编号。

### 消息组态

消息组态是指通过相应文本和属性来创建和编辑消息与消息模板，它涉及诸如与块相关的消息、与符号相关的消息以及诊断消息等。

### 形式参数

形式参数是指可为其分配参数的逻辑块中实际参数的占位符。在 FB 和 FC 中，形式参数由用户声明；在 SFB 和 SFC 中，形式参数已存在。当调用某个块时，会将实际参数分配给形式参数，这样被调用的块会使用最新值。形式参数属于块的本地数据，可被声明为输入、输出以及输入/输出参数。

### 寻址

在用户程序中分配地址。可将地址分配给一个内存位置或一系列的内存位置(例如：输入 I 12.1；位存储器 MW25)。

## W

### 完全重新启动

CPU 的启动模式。下面信息适用于完全重新启动模式：从工作存储器中删除由 SFC 生成的故障代码，且给所有其它故障代码分配它们在装载存储器中的缺省值。还将复位过程映像、定时器、计数器和存储位 - 与剩余的存储器组态无关。在完全重新启动期间，CPU 执行 OB 102、读取过程映像输入表，然后从第一个 OB 1 指令开始继续处理用户程序。

### 位存储器

这是 1 位内存位置。位存储器允许使用 STEP 7 的基本操作进行写和读取访问(使用位、字节、字和双字进行寻址)。用户可使用位存储器地址区来保存中间结果。

## Y

### 延时中断

延时中断属于 SIMATIC S7 程序执行中的一种优先级。当用户程序中的定时器过期时，将生成延时中断。随后将执行相应的组织块。

### 异步错误

异步错误是指未分配给用户程序中任何特定位置的运行时错误(例如，电源错误、扫描时间超限)。当发生这些错误时，操作系统将调用相应的组织块，用户可在这些组织块中对反应进行编程。

### 硬件中断

作为过程中特定事件的结果，硬件中断会由具有中断功能的模块触发。硬件中断将被报告给 CPU。然后再根据此中断的优先级来处理分配的组织块。

### 用户程序

用户程序包含所有语句和声明以及信号处理的数据，可利用这些数据来控制系统或过程。它被分配给可编程模块(模块，可编程)，并可在称作块的较小单元中进行构建。

## 用户程序错误

在 SIMATIC S7 PLC 中处理用户程序期间可能发生的错误(与过程错误不同)。  
在 SIMATIC S7 PLC 中处理用户程序期间发生的错误(与过程错误相反)。操作系统使用错误 OB (优先级系统)以及系统功能的状态字和输出参数来处理错误。

## 用于未组态连接的通讯 SFC

通讯 SFC 是用于数据交换和用于中止由通讯 SFC 创建的现有连接的系统功能。

## 优先级

当为组织块分配优先级时, 您将确定当前激活用户程序的中断能力, 以使高优先级的  
事件可中断低优先级的事件。

## 优先级别

CPU 的操作系统最多具有 28 个优先级, 为这些优先级分配了各种组织块。优先  
级可决定 OB 之间的中断能力。如果某个优先级中含有多个 OB, 则这些 OB 不会  
相互中断, 而是依次被执行。

## 与符号相关的消息

为符号表中的符号(输入、输出、位存储器、故障代码)组态消息时所使用的消息。  
在组态过程中, 必须为用于监视信号的 SCAN 功能设置时间间隔。

## 与块相关的消息

为具有消息功能的块(FB 或 DB)组态的消息。

## 语句表

语句表为 STEP 7 的汇编语言。在 STL 中处理程序时, 各条指令与 CPU 处理程序时  
所采用的顺序一致。

## 远程设备

远程设备是指可在网络中获得的设备, 例如, 打印机或计算机。与本地设备不同的是,  
在安装这些设备时, 必须为其分配网络地址。

## 运行系统错误

在 PLC 中(不在过程中)执行用户程序期间发生的错误。

## Z

### 组态连接的通讯 SFB

通讯 SFB 是用于数据交换和程序管理的系统功能块。

数据交换实例：SEND、RECEIVE、GET。

程序管理实例：将通讯伙伴的 CPU 设置为 STOP 状态，查询通讯伙伴的 CPU 的 STATUS。

### 诊断

诊断功能集成了所有系统诊断，并且包括对 PLC 中错误的识别、解释和报告。

### 诊断缓冲区

诊断缓冲区是 CPU 中的一个存储器区，所有诊断事件按照其发生的先后顺序存储在该区域中。

### 诊断数据

诊断数据是指错误消息中所包含的信息(诊断事件、时间标志)。

### 诊断条目

诊断事件在诊断缓冲区中使用诊断条目进行描述。

### 诊断消息

诊断消息由经过处理的诊断事件组成，并从 CPU 发往显示设备。

### 诊断中断

诊断模块使用发往 CPU 的诊断中断来报告识别出的系统错误。

### 指令

指令(STEP 5 或 STEP 7)是用文本语言所创建的程序中最小的部分。它表示处理器的命令。

### 子编号

要监视信号的编号(如果消息块可监视多个信号)。

### 自定义诊断

检测和评估自定义诊断事件。

### 自定义的诊断事件

由用户检测到的诊断事件，可将其置于诊断缓冲区中(使用 SFC52)。

### 自定义诊断消息

报告发生自定义诊断事件的消息。

### 中断

SIMATIC S7 优先级系统可识别 10 种不同的优先级，这些优先级控制用户程序的处理。中断属于这些优先级，例如，硬件中断。当发生中断时，操作系统将自动调用用户可在其中对所需反应进行编程的组织块(例如，在功能块中)。

### 组错误

由模块前面板上的 LED 显示屏所指示的错误消息(仅限在 S7-300 中)。只要所涉及的模块中存在错误(内部错误和外部错误)，LED 即会亮起。

### 组织块(OB)

组织块构成了 CPU 操作系统与用户程序之间的接口。用户程序的处理顺序在组织块中指定。



# 索引

## 字母

ACCFLT\_ESR 11-12  
ACCFLT\_MASKED 11-10、11-11  
ACCFLT\_QUERY 11-12  
ACCFLT\_RESET\_MASK 11-11  
ACCFLT\_SET\_MASK 11-10  
ACT\_TINT 9-6  
AD\_DT\_TM 24-6  
ADC/DAC 出错 30-2  
ALARM 22-10  
ALARM\_8 22-16  
ALARM\_8P 22-13  
ALARM\_D 22-33  
ALARM\_DQ 22-33  
ALARM\_S 22-29  
ALARM\_SC 22-32  
ALARM\_SQ 22-29  
AR\_SEND 22-18  
BCD 转换出错 11-7  
BLK 3-7  
BLKMOV 3-1  
BRCV 19-20  
BSEND 19-17  
BVAL 3-6  
C\_CNTRL 19-46  
C\_DIAG 13-18  
CAN\_DINT 10-5  
CAN\_TINT 9-5  
CDT 5-2  
CHANNEL 30-2、30-3  
    出错 30-2  
    信息 30-2  
CiR 4-4、4-5  
CiR 操作 1-34  
COMPRESS 3-13  
CONCAT 24-19  
CONT\_C 25-3  
CONT\_S 25-8  
CONTROL 19-44  
COUNT 3-8  
CPU 4-1、12-1、12-2、31-7、31-8  
    使用 SFC 46 STP 进入停机状态 4-1  
    特征 31-7  
    硬件故障 12-2  
CPU 417 和 CPU 417H 的访问出错 11-1  
CPU 417 和 CPU 417H 的访问出错过滤器 11-1  
CPU 冗余出错 OB(OB72) 1-24  
CPU 硬件故障 OB 1-37  
CPU 硬件故障组织块(OB84) 1-37  
CQ 6-6

CREA\_DB 3-23  
CREA\_DBL 3-16  
CREATE\_DB 3-8  
CTD 23-8  
CTRL\_RTM 6-5  
CTU 23-7  
CTUD 23-9  
CV 6-6  
D\_TOD\_DT 24-6  
DATE 5-1  
DB\_NUMBER 3-8  
DEL\_DB 3-10  
DEL\_SI 22-38  
DELETE 24-19  
DI\_STRNG 24-24  
DIS\_AIRT 12-7  
DIS\_IRT 12-3  
DIS\_MSG 22-20  
DMSK\_FLT 11-11  
DP\_PRAL 16-1  
DP\_TOPOL 13-23  
DPNRM\_DG 16-14  
DPRD\_DAT 16-18  
DPSYC\_FR 16-4  
DPWR\_DAT 16-20  
DP 主站系统信息 31-35  
DP 从站组 16-4  
    同步 16-4  
DRUM 14-12  
DSTBLK 3-2  
DT\_DATE 24-7  
DT\_DAY 24-7  
DT\_TOD 24-8  
DTIME 10-3  
EN\_AIRT 12-8  
EN\_IRT 12-5  
EN\_MSG 22-22  
EQ\_DT 24-10  
EQ\_STRNG 24-13  
EVENTN 13-10  
FB 12 BSEND 19-15  
FB 13 BRCV 19-18  
FB 8 USEND 19-9  
FB 9 URCV 19-12  
FC1 24-6  
FC10 24-13  
FC11 24-20  
FC12 24-10  
FC13 24-13  
FC14 24-11  
FC15 24-14  
FC16 24-24  
FC17 24-20  
FC18 24-11  
FC19 24-14



- FC2 24-19
- FC20 24-21
- FC21 24-21
- FC22 24-16
- FC23 24-12
- FC24 24-15
- FC25 24-16
- FC26 24-22
- FC27 24-17
- FC28 24-12
- FC29 24-15
- FC3 24-6
- FC30 24-25
- FC31 24-22
- FC32 24-23
- FC33 24-8
- FC34 24-8
- FC35 24-9
- FC36 24-18
- FC37 24-25
- FC38 24-26
- FC39 24-26
- FC4 24-19
- FC5 24-24
- FC6 24-7
- FC62 19-46
- FC7 24-7
- FC8 24-8
- FC9 24-10
- FILL 3-6
- FIND 24-20
- GADR\_LGC 15-1
- GD\_RCV 17-4
- GD\_SND 17-1
- GD 信息包 17-1、17-2、17-3、17-4、17-5
  - 通过 SFC 60 发送 17-1
  - 通过 SFC 61 编程接受 17-4
- GE\_DT 24-10
- GE\_STRNG 24-13
- GET 19-24
- GET 和 PUT SFC 的数据的一致性 18-4
- GT\_DT 24-11
- GT\_STRNG 24-14
- H/F 事件 32-12
- H\_CTRL 27-1
- I/O 冗余出错 OB(OB70) 1-22
- I/O 区域中的位域 14-5
  - 使用 SFC 79 设置 14-5
- I/O 访问出错 11-9
  - 写入时 11-7、11-8、11-9
  - 读取时 11-7、11-8、11-9
- I/O 访问出错组织块(OB122) 1-57
- I\_ABORT 20-20
- I\_GET 20-16
- I\_PUT 20-18
- I\_STRNG 24-24
- IEC 功能的技术数据 24-3
- INFO1 13-10
- INFO2 13-12、13-13
- INSERT 24-20
- IOID 7-5、7-7、7-8、7-11、7-13
- LADDR 7-5、7-7、7-8、7-9、7-11、7-13
- LE\_DT 24-11
- LE\_STRNG 24-14
- LEFT 24-21
- LEN 24-21
- LGC\_GADR 15-3
- LIMIT 24-16
- LOW\_LIMIT 3-8
- LT\_DT 24-12
- LT\_STRNG 24-15
- MAX 24-16
- MID 24-22
- MIN 24-17
- MODE 12-3、12-5
- MP\_ALM 4-3
- MSK\_FLT 11-10
- M 短路 30-4、30-5
  - 模拟量输出模块 30-4
  - 模拟量输入模块 30-4
- NE\_DT 24-12
- NE\_STRNG 24-15
- NOTIFY 22-5
- NOTIFY\_8P 22-7
- NR 6-4、6-5、6-6
- OB 10到 OB 17 1-6
- OB 20到 OB 23 1-10
- OB 55 1-16
- OB 56 1-17
- OB 73 1-27
- OB 83 1-34、1-35
- OB 88 1-47
- OB) 1-34
- OB\_NR 9-4、9-5、9-6、9-7、10-3、10-4、10-5、12-3、12-4、12-5
- OB\_RT 13-14
- OB100、OB101 和 OB102 1-50
- OB121 1-55、1-57
- OB30 到 OB38 1-12
- OB40 到 OB47 1-14
- OB57 1-18
- OB60 1-19、4-3
- OB70 1-22
- OB72 1-24、1-26
- OB80 1-28
- OB81 1-30
- OB82 1-32
- OB84 1-37
- OB85 1-38
- OB86 1-42、1-43
- OB87 1-45
- OB 的本地数据 31-31
- P sh 30-4
- PARAM\_MOD 7-8
- PERIOD 9-4
- PN\_DP 21-6
- PN\_IN 21-4
- PRGFLT\_ESR 11-12
- PRGFLT\_MASKED 11-10、11-11
- PRGFLT\_QUERY 11-12
- PRGFLT\_RESET\_MASK 11-11
- PRGFLT\_SET\_MASK 11-10

- PRINT 19-30
- PROFINet 接口 21-1
- PROFINet 组件 21-4、21-5、21-6
  - 更新互连 21-6
  - 更新用户程序接口输出 21-5
  - 更新用户程序接口输入 21-4
- PULSEGEN 25-15、25-16、25-21
- PULSEGEN 块的实例 25-25
- PUT 19-21
- PV 6-4
- P 短路 30-4、30-5
  - 模拟量输出模块 30-4
  - 模拟量输入模块 30-4
- QRY\_DINT 10-4
- QRY\_TINT 9-7
- R\_STRNG 24-25
- RALRM 8-5
- RAM 出错 30-2
- RD\_DPARA 7-4
- RD\_LGADR 15-5
- RD\_REC 7-13
- RD\_SINFO 13-1
- RDREC 8-1
- RDSYSST 13-4、31-1
- RE\_TRIGR 4-1
- READ\_CLK 5-2
- READ\_DBL 3-19
- READ\_ERR 11-12
- READ\_RTM 6-6
- READ\_SI 22-35
- RECNUM 7-5、7-7、7-11、7-13
- RECORD 7-5、7-6、7-11、7-12
- REPL\_VAL 3-15
- REPLACE 24-22
- RESUME 19-38
- RTM 6-2
- S 6-5
- S5TI\_TIM 24-8
- S7-300 连接的状态 19-46
- S7 通讯的工作存储器要求 19-48
- S7 通讯块 SFB 的分类和对工作存储区的要求 18-6
- S7 通讯块和 S7 基本通讯块之间的差别 18-1
- SALRM 8-14
- SB\_DT\_DT 24-8
- SB\_DT\_TM 24-9
- SDT 9-4
- SEL 24-18
- SEND 13-10
- SET\_CLK 5-1
- SET\_CLKS 5-4
- SET\_RTM 6-4
- SET\_TINT 9-4
- SFB 19-2、19-3、19-5、19-6、19-7、19-44
  - 参数分类 19-1
  - 启动反应 19-5
  - 对出错和故障的反应 19-7
  - 查询属于 SFB 实例的连接状态 19-44
- SFB 0 CTU 23-7
- SFB 1 CTD 23-8
- SFB 12 BSEND 19-15
- SFB 13 BRCV 19-18
- SFB 16 PRINT 19-27
- SFB 19 START 19-33
- SFB 2 CTUD 23-9
- SFB 20 STOP 19-36
- SFB 21 RESUME 19-38
- SFB 22 STATUS 19-40
- SFB 23 "USTATUS" 19-42、19-43
- SFB 23 USTATUS 19-42
- SFB 29(HS\_COUNT) 28-1
- SFB 3 TP 23-1
- SFB 30(FREQ\_MES) 28-3
- SFB 31 NOTIFY\_8P 22-7
- SFB 32 DRUM 14-11
- SFB 33 ALARM 22-10
- SFB 34 ALARM\_8 22-16
- SFB 35 ALARM\_8P 22-13
- SFB 36 NOTIFY 22-5
- SFB 37 AR\_SEND 22-18
- SFB 38(HSC\_A\_B) 28-4
- SFB 39(POS) 28-5
- SFB 4 TON 23-3
- SFB 41 CONT\_C 25-1
- SFB 42 CONT\_S 25-8
- SFB 43 PULSEGEN 25-14
  - 非对称三步控制 25-19
  - 两步控制 25-14
  - 三步控制 25-17、25-18、25-19、25-22、25-23、25-24
  - 自动同步 25-16、25-17
- SFB 44 26-10
- SFB 46 26-20
- SFB 47 26-26
- SFB 48 26-30
- SFB 49 26-33
- SFB 5 TOF 23-5
- SFB 52 RDREC 8-1
- SFB 53 WRREC 8-3
- SFB 54 RALRM 8-5
- SFB 60 26-35、26-57
  - 附加出错信息 26-57
- SFB 60 至 65 的附加出错信息 26-57
- SFB 61 26-38、26-57
  - 附加出错信息 26-57
- SFB 62 26-41、26-57
  - 附加出错信息 26-57
- SFB 63 26-43、26-57
  - 附加出错信息 26-57
- SFB 64 26-47、26-57
  - 附加出错信息 26-57
- SFB 65 26-52、26-57
  - 附加出错信息 26-57
- SFB 75 SALRM 8-14
- SFB 8 USEND 19-9
- SFB 9 URVCV 19-12
- SFB ANALOG 26-1
- SFB COUNT 26-23
- SFB DIGITAL 26-13
- SFB FETCH RK 26-47
- SFB FREQUENC 26-28
- SFB PULSE 26-32

- SFB RCV\_PTP 26-38  
SFB RES\_RCVB 26-41  
SFB SEND\_PTP 26-35、26-43  
SFB SERVE\_RK 26-52、26-53  
SFB 列表  
    按数字排序 33-7  
    按字母排序 33-9  
SFC 0 SET\_CLK 5-1  
SFC 1 READ\_CLK 5-2  
    参数 5-2  
SFC 10 DIS\_MSG 22-20  
SFC 100 SET\_CLKS 5-4  
SFC 101 RTM 6-2  
SFC 102 RD\_DPARA 7-4  
SFC 103 DP\_TOPOL 13-23  
SFC 104 CIR 4-4  
SFC 105 READ\_SI 22-35  
SFC 106 DEL\_SI 22-38  
SFC 107 ALARM\_DQ 22-33  
SFC 108 ALARM\_D 22-33  
SFC 11 DPSYC\_FR 16-4  
SFC 112 "PN\_IN" 21-1、21-4  
    背景信息 21-1  
SFC 112、113 和 114 背景信息: 21-1  
SFC 113 "PN\_OUT" 21-1  
    背景信息 21-1  
SFC 114 "PN\_DP" 21-1、21-6  
    背景信息 21-1  
SFC 12 D\_ACT\_DP 16-10  
SFC 126 SYNC\_PI 14-6  
SFC 127 SYNC\_PO 14-8  
SFC 13 DPNRM\_DG 16-14  
SFC 14 DPRD\_DAT 16-18  
SFC 15 DPWR\_DAT 16-20  
SFC 17 ALARM\_SQ 22-29  
SFC 18 ALARM\_S 22-29  
SFC 19 ALARM\_SC 22-32  
SFC 2 SET\_RTM 6-4  
SFC 20 BLKMOV 3-1  
SFC 21 FILL 3-6  
SFC 22 CREAT\_DB 3-8  
    出错信息 3-8  
SFC 22 CREATE\_DB 3-8  
SFC 23 DEL\_DB 3-10  
    出错信息 3-10  
SFC 25 COMPRESS 3-13  
SFC 26 UPDAT\_PI 14-1  
SFC 27 UPDAT\_PO 14-3  
SFC 28 - 31 的特征 9-2  
SFC 28 SET\_TINT 9-4  
SFC 29 CAN\_TINT 9-5  
SFC 3 CTRL\_RTM 6-5  
SFC 30 ACT\_TINT 9-6  
SFC 31 QRY\_TINT 9-7  
SFC 32 SRT\_DINT 10-3  
SFC 33 CAN\_DINT 10-5  
SFC 34 QRY\_DINT 10-4  
    出错信息 10-4  
SFC 35 MP\_ALM 4-3  
SFC 36 MSK\_FLT 11-10  
SFC 37 DMSK\_FLT 11-11  
SFC 38 READ\_ERR 11-12  
SFC 39 DIS\_IRT 12-3  
SFC 4 READ\_RTM 6-6  
SFC 40 EN\_IRT 12-5  
    出错信息 12-5  
SFC 41 DIS\_AIRT 12-7  
    返回值 12-7  
SFC 42 EN\_AIRT 12-8  
    出错信息 12-8  
    返回值 12-8  
SFC 43 RE\_TRIGR 4-1  
SFC 44 REPL\_VAL 3-15  
SFC 46 STP 4-1  
SFC 46 TIME\_TCK 6-7  
SFC 47 WAIT 4-2  
SFC 48 SNC\_RTCB 5-3  
SFC 49 LGC\_GADR 15-3  
SFC 5 GADR\_LGC 15-1  
SFC 50 RD\_LGADR 15-5  
SFC 51 RDSYSST 13-4、31-1  
SFC 52 WR\_USRMSG 13-10  
SFC 55 WR\_PARM 7-5  
SFC 55 到 59 的进一步出错信息 7-18  
SFC 56 WR\_DPARM 7-7  
SFC 57 PARM\_MOD 7-8  
SFC 58 WR\_REC 7-11  
    参数 7-11  
SFC 59 RD\_REC 7-13  
SFC 6 RD\_SINFO 13-1  
SFC 60 GD\_SND 17-1  
SFC 61 GD\_RCV 17-4  
SFC 62 CONTROL 19-44  
SFC 63(AB\_CALL) 29-1  
SFC 64 TIME\_TICK 6-7  
    参数 6-7  
SFC 65 X\_SEND 20-7  
SFC 66 X\_RCV 20-8  
SFC 67 X\_GET 20-13  
SFC 68 X\_PUT 20-11  
SFC 69 X\_ABORT 20-15  
SFC 7 DP\_PRAL 16-1  
SFC 72 I\_GET 20-16  
SFC 73 I\_PUT 20-18  
SFC 74 I\_ABORT 20-20  
SFC 78 "OB\_RT" 13-14  
SFC 79 SET 14-5  
SFC 80 RSET 14-10  
SFC 82 CREA\_DBL 3-16  
SFC 83 READ\_DBL 3-19  
SFC 84 WRIT\_DBL 3-21  
SFC 85 CREA\_DB 3-23  
SFC 87 C\_DIAG 13-18  
SFC 9 EN\_MSG 22-22  
SFC 90 H\_CTRL 27-1  
SFC113 "PN\_OUT" 21-5  
SFC114 "PN\_DP" 21-6  
SFC 列表  
    按数字排序 33-1  
    按字母排序 33-4  
SIGN 10-3  
SNC\_RTCB 5-3  
SRCBLK 3-1

SRT\_DINT 10-3  
 SSL ID W#16#xy25 - 将过程映像分区分配到 OB 31-18  
 SSL ID 为 W#16#0132、索引为 W#16#000B 的部分列表摘录的数据记录 31-25  
 SSL ID 为 W#16#0132、索引为 W#16#000C 的部分列表摘录的数据记录 31-26  
 SSL\_HEADER 13-4  
 SSL-ID 31-4  
 SSL-ID W#16#00B1 - 模块的诊断信息 31-47  
 SSL-ID W#16#00B2 - 对应物理地址的诊断数据记录 1 31-49  
 SSL-ID W#16#00B3 - 带逻辑基址的模块诊断数据 31-50  
 SSL-ID W#16#00B4 - DP 从站的诊断数据 31-51  
 SSL-ID W#16#0132, 索引为 W#16#0005 的部分列表的数据记录摘录 31-22  
 SSL-ID W#16#0132, 索引为 W#16#0008 的部分列表的数据记录摘录 31-23  
 SSL-ID W#16#0232, 索引为 W#16#0004 的部分列表的数据记录摘录 31-27  
 SSL-ID W#16#0x75 - H 系统中切换的 DP 从站 31-33  
 SSL-ID W#16#xy11 - 模块标识 31-6  
 SSL-ID W#16#xy12 - CPU 特征 31-7  
 SSL-ID W#16#xy13 - 存储区域 31-9  
 SSL-ID W#16#xy14 - 系统区 31-10  
 SSL-ID W#16#xy15 - 块类型 31-12  
 SSL-ID W#16#xy19 - 模块 LED 的状态 31-13  
 SSL-ID W#16#xy1C - 组件标识 31-15  
 SSL-ID W#16#xy32 - 通讯状态数据 31-21  
 SSL-ID W#16#xy71 - H CPU 的组信息 31-28  
 SSL-ID W#16#xy74 - 模块 LED 的状态 31-31  
 SSL-ID W#16#xy90 - DP 主站系统信息 31-35  
 SSL-ID W#16#xy91 - 模块的状态信息 31-37  
 SSL-ID W#16#xy92 - 机架/站的状态信息 31-41  
 SSL-ID W#16#xy95 - 扩展 DP 主站系统信息 31-44  
 SSL-ID W#16#xyA0 - 诊断缓冲区 31-46  
 START 19-33、19-34  
 STATUS 9-7、10-4、19-40、19-41  
 STEP 7 1-12、1-28、1-30、1-32、1-38、1-39、1-55、1-57  
 OB 的类型  
   OB121 1-55  
   OB122 1-57  
   OB35 1-12  
   OB80 1-28  
   OB81 1-30  
   OB82 1-32  
   OB85 1-38  
 STOP 19-36、19-37  
   将远程设备切换到 19-36  
 STP 4-1  
 STRNG\_DI 24-25  
 STRNG\_I 24-26  
 STRNG\_R 24-26

SYNC\_PI 14-6  
 SYNC\_PO 14-8  
 TIME\_TCK 6-7  
 TOF 23-5  
 TON 23-3  
 TP 23-1  
 UP\_LIMIT 3-8  
 UPDAT\_PI 14-1  
 UPDAT\_PO 14-3  
 URCV 19-12  
 USEND 19-9  
 USTATUS 19-42  
 WAIT 4-2  
 WR\_DPARM 7-7  
 WR\_PARM 7-5  
 WR\_REC 7-11  
 WR\_USRMSG 13-10  
 WRIT\_DBL 3-21  
 WRREC 8-3  
 WT 4-2  
 X\_ABORT 20-15  
 X\_GET 20-13  
 X\_PUT 20-11  
 X\_RCV 20-8  
 X\_SEND 20-7

## B

保留的事件等级 32-18  
 保险丝断开 30-2  
 备用电池 30-2  
   故障 30-2  
 比较 DATE\_AND\_TIME 变量 24-10、24-11、24-12  
 比较 STRING 变量 24-13、24-14、24-15  
 编程 1-12、1-28、1-30、1-32、1-38、1-55、1-57  
 OB 的类型  
   OB121 1-55  
   OB122 1-57  
   OB35 1-12  
   OB80 1-28  
   OB81 1-30  
   OB82 1-32  
   OB85 1-38  
 编辑 STRING 变量 24-19、24-20、24-21、24-22、24-23  
 编程出错 11-1  
 编程出错过滤器 11-1  
 编程出错组织块(OB121) 1-55  
 编辑数值 24-16、24-17、24-18  
 变量声明表 1-12、1-28、1-55、1-57  
   对于 OB121 1-55  
   对于 OB122 1-57  
   对于 OB35 1-12  
   对于 OB80 1-28  
 标识 DP 主站系统中的总线拓扑 13-23  
 标准 OB 事件 32-3  
 标准用户事件 32-16  
 部分 SSL 列表的结构 31-3

## C

- 参数 2-6、2-7、3-1、3-2、3-3、3-7、3-8、3-9、4-2、5-2、6-4、6-5、6-6、6-7、7-1、7-5、7-7、7-8、7-10、7-11、7-12、7-13、7-14、7-16、7-17、9-4、9-5、9-6、9-7、10-3、10-4、10-5、11-10、11-11、11-12、12-3、12-4、12-5、12-6、13-10、13-11、13-12、13-13
- ACCFLT\_ESR 11-12
- ACCFLT\_MASKED 11-10、11-11
- ACCFLT\_QUERY 11-12
- ACCFLT\_RESET\_MASK 11-11
- ACCFLT\_SET\_MASK 11-10
- BLK 3-6
- BVAL 3-6、3-7
- CDT 5-2
- COUNT 3-8
- CQ 6-6
- CV 6-6
- DB\_NUMBER 3-8
- DSTBLK 3-1
- DTIME 10-3
- EVENTN 13-12、13-13
- INFO1 13-12、13-13
- INFO2 13-10
- IOID 7-5、7-7、7-8、7-9、7-11、7-13、7-16
- LADDR 7-5、7-7、7-8、7-11、7-13、7-16
- LOW\_LIMIT 3-8
- MODE 12-3、12-4、12-5、12-6
- NR 6-4、6-5、6-6
- OB\_NR 9-4、9-5、9-6、9-7、10-3、10-4、10-5、12-3、12-5、12-6
- PERIOD 9-4
- PRGFLT\_ESR 11-12
- PRGFLT\_MASKED 11-10、11-11
- PRGFLT\_QUERY 11-12
- PRGFLT\_RESET\_MASK 11-11
- PRGFLT\_SET\_MASK 11-10
- PV 6-4
- RECNUM 7-5、7-7、7-11、7-13
- RECORD 7-5、7-11
- REQ 与异步 SFC 的 2-4
- SDT 9-4
- SEND 13-10、13-11、13-12
- SFC 1 READ\_CLK 5-2
- SFC 51 及 55 到 59 中的 BUSY 2-4
- SFC 51 及 55 到 59 中的 RET\_VAL 2-4
- SFC 58 WR\_REC 7-11
- SFC 64 TIME\_TICK 6-7
- SRCBLK 3-2
- STATUS 9-7、10-4
- UP\_LIMIT 3-8
- WT 4-2
- 信号模块的 7-1
- 写一条数据记录 7-7
- 参考通道出错 30-4
  - 模拟量输入模块 30-4
- 参数分配出错 30-4
  - 模拟量输入模块 30-4
  - 模拟量输出模块 30-4
  - 数字量输入模块 30-4
- 插入/删除 OB(OB 83) 1-34
- 超出测量范围 30-4
  - 模拟量输入模块 30-4
- 程序出错 12-2
- 重新启动(暖重启)或完全重启 19-33
  - 在远程设备上执行 19-33
- 重新启动(热重新启动) 19-33
- 重新启动 1-50、1-51、1-52、1-53、1-54、19-38、19-39
  - 在远程设备上启动 19-38
- 重新组态模块 1-34
- 出错 11-1、11-2、11-3、11-4、11-6、11-7、11-8、11-9、12-2、30-2、30-3
  - ADC/DAC 30-2
  - EPROM 30-2
  - RAM 30-2
  - 编程 11-1、11-2、11-3、11-7、11-8
  - 访问 11-1、11-2、11-4、11-6、11-9
  - 屏蔽 11-1
  - 同步 11-1、11-2
  - 异步 12-1
- 出错 OB 1-12、1-28、1-30、1-32、1-38、11-1
  - OB 的类型
    - OB35 1-12
    - OB80 1-28、1-29
    - OB81 1-30
    - OB82 1-32
    - OB85 1-38、1-39、1-40、1-41
- 出错处理 11-1
- 出错过滤器 11-2、11-3、11-4、11-6、11-7、11-8、11-9
  - 编程出错 11-1、11-3、11-7、11-8
  - 访问出错 11-1、11-2、11-4、11-6、11-9
- 出错寄存器 11-1、11-12
  - 使用 SFC 38 RD\_ERR 11-12
- 出错检测 1-12、1-28、1-55、1-57
  - OB 的类型
    - OB121 1-55
    - OB122 1-57
    - OB35 1-12
    - OB80 1-28
- 出错信息 3-9、3-10、3-11、10-4、12-6、12-8
  - SFC 22 CREAT\_DB 3-8
  - SFC 23 DEL\_DB 3-10
  - SFC 34 QRY\_DINT 10-4
  - SFC 40 EN\_IRT 12-5
  - SFC 42 EN\_AIRT 12-8
- 出错中断 12-1
  - 同步 12-2
  - 异步 12-1、12-2
- 处理器故障 30-2
- 处理时钟中断 9-1
- 处理延时中断 10-1
- 处理中断 OB(OB 88) 1-47
- 初始化存储区 3-6
  - 使用 SFC 21 FILL 3-6

传送参数 7-5、7-7  
  使用 SFC 55 WR\_PARM 7-5  
  使用 SFC 56 WR\_DPARM 7-7  
创建数据块 3-23  
从通讯伙伴接收数据并将其存储在数据块(RK 512)中 26-52  
存储卡 30-2  
存储器区 31-9

## D

递减计数器 23-8  
递增/递减计数 23-9  
递增计数器 23-7  
电池用尽 30-2  
电流低于测量范围 30-4  
  模拟量输入模块 30-4  
电源 30-2  
  故障 30-2  
电源出错 12-2  
电源出错组织块(OB81) 1-30  
调用汇编码块 29-1  
定时器编号出错 11-1  
读取 13-4、16-16、16-18  
  DP 从站的连续数据 16-18  
  DP 从站的诊断数据 16-14  
  使用 SFC 51 RDSYSST 13-4  
读取时间 5-2  
  使用 SFC 1 READ\_CLK 5-2  
读取系统时间 6-7  
  使用 SFC 64 TIME\_TCK 6-7  
读取系统状态 13-4  
  使用 SFC 51 RDSYSST 13-4  
断开延迟 23-5  
  生成 23-5  
断线 30-4、30-5  
  模拟量输出模块 30-4  
  模拟量输入模块 30-4  
对齐出错 11-8  
  读取时 11-1  
  写入时 11-1  
多值计算中断 12-2  
多值计算中断组织块(OB60) 1-19

## F

发送分段数据 19-15  
  通过 FB 12 19-15  
  通过 SFB 12 19-15  
返回值 12-7、12-8  
  SFC 41 DIS\_AIRT 12-7  
  SFC 42 EN\_AIRT 12-8  
访问出错 11-1  
访问出错过滤器 11-4  
复制变量 3-1  
  使用 SFC 20 BLKMOV 3-2

## G

概述 18-8、24-1  
更新 DP 互连 21-6  
更新过程映像输出表 14-3  
更新过程映像输入表 14-1  
更新中断 1-17  
更新中断 OB(OB 56) 1-17  
共模出错 30-4  
  模拟量输出模块 30-4  
  模拟量输入模块 30-4  
故障代码 3-20  
关于使用 SFB 生成块相关消息的介绍 22-1  
过 SFB 23 "USTATUS"接收远程设备的状态改变 19-42  
过滤器 11-1  
  出错 11-1、11-2、11-4、11-6、11-7、11-8、11-9

## H

后台组织块(OB90) 1-48

## J

集成计数器功能 28-1  
集成频率计功能 28-3  
计数器 11-8  
  编号出错 11-8  
计数器(CPU 312) 28-1  
机架故障 1-42、1-43、12-2  
机架故障组织块(OB86) 1-42  
将全部或部分数据块传送给通讯伙伴 26-35、26-43  
将全部或部分数据块传送给通讯伙伴(RK 512) 26-47  
接地出错 30-5  
  数字量输入模块 30-4  
接口 DB 21-1  
接收分段数据 19-18  
  通过 FB 13 19-18  
  通过 SFB 13 19-18  
接收来自通讯伙伴的数据并将其填入数据块。 26-38  
接通延迟 23-3  
  生成 23-3

## K

可能的部分系统状态列表 31-5  
空载电压 30-4、30-5  
  模拟量输出模块 30-4  
控制 25-1、25-4、25-6、25-8、25-9、25-12  
  使用 SFB 41 实现连续控制 25-1  
  使用 SFB 42 进行步控制 25-8  
块编号出错 FC 11-8  
块类型 31-12  
扩展机架故障 30-2

**L**

类型 ID 31-4  
  模块的 31-4  
连接 13-18、13-19、13-20、13-21、13-22、  
  19-46  
  使用 SFC 87 诊断 13-18  
  状态 S-300(FC62) 19-46  
临时变量(TEMP) 1-57  
  OB 所需的 1-57  
逻辑地址 15-1、15-5  
  模块的查询所有地址 15-5  
  通道的查询 15-1

**M**

脉冲 23-1、23-2  
  生成 23-1  
脉宽调制 25-14、25-15、25-22  
模块 30-2、30-3、31-4、31-6  
  Fault 30-2  
  标识 31-6  
  类型 ID 31-4  
模块插槽 15-3  
  逻辑地址的查询 15-3  
模块的诊断数据 30-1  
模式更改 32-7  
模块诊断数据 31-49、31-50  
模块诊断信息 31-47  
模块状态信息 31-37

**N**

内部出错 30-2  
暖重启 1-50

**P**

屏蔽 11-1  
  出错 11-1  
屏蔽同步出错 11-1  
频率计(CPU 312) 28-3

**Q**

启动 1-50、1-51、1-52、1-53  
启动组织块(OB100  
  OB101 和 OB102) 1-50  
清除输入缓冲区 26-41  
取消屏蔽 11-1  
  出错事件 11-1  
  读取时 11-1  
  写入时 11-1  
区域长度出错 11-7、11-8  
  读取时 11-1  
  写入时 11-1  
区域出错 11-7、11-8

**R**

日期和时间作为复杂数据类型 24-5

**S**

删除/插入模块中断 12-2  
设备状态改变 19-42  
  接收远程设备的状态改变 19-42  
设定 TOD 5-1  
  使用 SFC 0 SET\_CLK 5-1  
生成数据块 3-16  
  使用 SFC 82 "CREA\_DBL 3-16  
时间 5-1  
事件 13-10、13-11、13-12、13-13、32-1、  
  32-2  
  ID 13-11、13-12、32-1、32-2  
  等级 32-1  
时间出错 12-2  
时间出错组织块(OB80) 1-28  
事件和事件标识符 32-1  
事件等级 1 - 标准 OB 事件 32-3  
事件等级 2 - 异步出错 32-3  
事件等级 3 - 异步出错 32-4  
事件等级 4 - 停止事件和其它模式改变 32-7  
事件等级 5 - 模式运行期事件 32-10  
事件等级 6 - 通讯事件 32-11  
事件等级 7 - H/F 事件 32-12  
事件等级 8 - 模块的诊断事件 32-14  
事件等级 9 - 标准用户事件 32-16  
事件等级 A 和 B - 自由用户事件 32-17  
时间功能 24-6、24-7、24-8、24-9  
  使用 SFB 0"CTU"进行递增计数 23-7  
  使用 SFB 1 "CTD"递减计数 23-8  
  使用 SFB 16"PRINT"将数据发送到打印机 19-27  
  使用 SFB 19"START"在远程设备上开始一个暖重  
  启或冷重启 19-33  
  使用 SFB 2 "CTUD"递增/递减计数 23-9  
  使用 SFB 22 "STATUS"查询远程伙伴的状态 19-  
  40  
  使用 SFB 3 "TP"生成一个脉冲 23-1  
  使用 SFB 31 "NOTIFY\_8P"生成无确认显示的块  
  相关消息 22-7  
  使用 SFB 32"DRUM"实现一个操作序列 14-11  
  使用 SFB 33 "ALARM"生成需要确认的块相关消  
  息 22-10  
  使用 SFB 34 "ALARM\_8"生成针对八个信号的不  
  附带关联值的块相关消息 22-16  
  使用 SFB 35"ALARM\_8P"生成带 8 个信号伴随值  
  的与块相关的消息 22-13  
  使用 SFB 36 "NOTIFY"生成无需确认的块相关消  
  息 22-5  
  使用 SFB 37"AR\_SEND"发送归档数据 22-18  
  使用 SFB 4"TON"生成一个接通延迟 23-3  
  使用 SFB 41/FB 41 "CONT\_C"实现连续控制  
  25-1  
  使用 SFB 42/FB 42 "CONT\_S"进行步控制 25-8

- 使用 SFB 43 "PULSEGEN"生成脉冲 25-14
- 使用 SFB 5 "TOF"生成断开延迟 23-5
- 使用 SFB 52"RDREC"从 DP 从站读一个数据记录 8-1
- 使用 SFB 53 "WRREC"将数据记录写入 DP 从站 8-3
- 使用 SFC 0 "SET\_CLK"设定 TOD 5-1
- 使用 SFC 100 "SET\_CLKS"设定时间日期和 TOD 状态 5-4
- 使用 SFC 101 "RTM"控制运行时间定时器 6-2
- 使用 SFC 102 "RD\_DPARA"读取预定义参数 7-4
- 使用 SFC 105 "READ\_SI"读取动态系统资源 22-35
- 使用 SFC 106 "DEL\_SI"释放动态系统资源 22-38
- 使用 SFC 107 "ALARM\_DQ"生成可确认和永久确认的块相关消息 22-33
- 使用 SFC 108 "ALARM\_D"生成可确认和永久确认的块相关消息 22-33
- 使用 SFC 11"DPSYC\_FR"同步 DP 从站组 16-4
- 使用 SFC 126 "SYNC\_PI"在同步循环中更新过程映像分区输入表 14-6
- 使用 SFC 127 "SYNC\_PO"在同步循环中更新过程映像分区 14-8
- 使用 SFC 13"DPNRM\_DG"(从站诊断)读取 DP 从站的诊断数据 16-14
- 使用 SFC 14"DPRD\_DAT"读取 DP 标准从站的连续数据 16-18
- 使用 SFC 15"DPWR\_DAT"将连续数据写入到 DP 标准从站 16-20
- 使用 SFC 2 "SET\_RTM"设置运行时间定时器 6-4
- 使用 SFC 20 3-1
- 使用 SFC 21 "FILL"初始化存储区 3-6
- 使用 SFC 22 "CREAT\_DB"创建数据块 3-8
- 使用 SFC 23 "DEL\_DB"删除数据块 3-10
- 使用 SFC 24"TEST\_DB"测试数据块 3-12
- 使用 SFC 25 "COMPRESS"压缩用户存储器 3-13
- 使用 SFC 26"UPDAT\_PI"更新过程映像输入表 14-1
- 使用 SFC 27"UPDAT\_PO"更新过程映像输出表 14-3
- 使用 SFC 28"SET\_TINT"设置日时钟中断 9-4
- 使用 SFC 29 "CAN\_TINT"取消时钟中断 9-5
- 使用 SFC 3 "CTRL\_RTM"启动和停止运行时间定时器 6-5
- 使用 SFC 30 "ACT\_TINT"激活时钟中断 9-6
- 使用 SFC 31 "QRY\_TINT"查询日时钟中断 9-7
- 使用 SFC 32 "SRT\_DINT"启动延时中断 10-3
- 使用 SFC 33 "CAN\_DINT"取消延时中断 10-5
- 使用 SFC 34 "QRY\_DINT"查询日时钟中断 10-4
- 使用 SFC 35 "MP\_ALM"触发多处理器中断 4-3
- 使用 SFC 36"MSK\_FLT"屏蔽同步出错 11-10
- 使用 SFC 37"DMSK\_FLT"解除屏蔽同步出错 11-11
- 使用 SFC 38"READ\_ERR"读取出错寄存器 11-12
- 使用 SFC 39"DIS\_IRT"禁用新中断和异步出错的 处理 12-3
- 使用 SFC 4 "READ\_RTM"读取运行时间定时器 6-6
- 使用 SFC 40 "EN\_IRT"启用新中断和异步出错的 处理 12-5
- 使用 SFC 41"DIS\_AIRT"延迟更高优先级中断和 异步出错的 处理 12-7
- 使用 SFC 42"EN\_AIRT"启用处理较高优先级的中 断和异步出错 12-8
- 使用 SFC 43 "RE\_TRIGR"再触发循环时间监控 4-1
- 使用 SFC 44"REPL\_VAL"将替换值传送到累加器 1 中 3-15
- 使用 SFC 46 "STP"将 CPU 切换为 STOP 4-1
- 使用 SFC 47 "WAIT"延时用户程序执行 4-2
- 使用 SFC 48 "SNC\_RTCB"同步子时钟 5-3
- 使用 SFC 49"LGC\_GADR"查询属于一个逻辑地 址的模块插槽 15-3
- 使用 SFC 5"GADR\_LGC"查询模块的逻辑基址 15-1
- 使用 SFC 50"RD\_LGADR"查询模块的所有逻辑 地址 15-5
- 使用 SFC 51"RDSYSST"读取系统状态列表或部 分列表 13-4
- 使用 SFC 52"WR\_USMSG"将自定义诊断事件写 入到诊断缓冲区中 13-10
- 使用 SFC 54"RD\_DPARM"读取定义的参数 7-3
- 使用 SFC 55 "WR\_PARM"写动态参数 7-5
- 使用 SFC 56"WR\_DPARM"写一条数据记录 7-7
- 使用 SFC 58"WR\_REC"写入数据记录 7-11
- 使用 SFC 59 "RD\_REC"读一个数据记录 7-13
- 使用 SFC 6"RD\_SINFO"读取 OB 的启动信息 13-1
- 使用 SFC 60"GD\_SND"发送一个 GD 信息包 17-1
- 使用 SFC 61"GD\_RCV"编程接受已接收的 GD 信 息包 17-4
- 使用 SFC 64 "TIME\_TCK"读取系统时间 6-7
- 使用 SFC 65"X\_SEND"将数据发送给在本地 S7 站外的一个通讯伙伴 20-7
- 使用 SFC 66"X\_RCV"接收来自本地 S7 站外一个 通讯伙伴的数据 20-8
- 使用 SFC 69"X\_ABORT"中止到本地 S7 站外的 一个通讯伙伴的现有连接 20-15
- 使用 SFC 6 读取 OB 启动信息 13-1
- 使用 SFC 7"DP\_PRAL"在 DP 主站上触发硬件中 断 16-1
- 使用 SFC 72 "I\_GET"从本地 S7 站内的一个通讯 伙伴上读取数据 20-16
- 使用 SFC 78 }"OB\_RT"确定 OB 程序运行时间 13-14
- 使用 SFC 79"SET"在 I/O 区域中设置位域 14-5
- 使用 SFC 80 "RSET"复位 I/O 区域中的位域 14-10



- 使用 SFC 81 "UBLKMOV"不间断地复制变量 3-4
  - 使用 SFC 82 "CREA\_DBL"在装载存储器中生成数据块 3-16
  - 使用 SFC 83 "READ\_DBL"从装载存储器的数据块中读取数据 3-19
  - 使用 SFC 84 "WRIT\_DBL"在装载存储器中写入数据块 3-21
  - 使用 SFC 85 "CREA\_DB"创建数据块 3-23
  - 使用 SFC 87 "C\_DIAG"诊断当前连接状态 13-18
  - 使用 SFC 87 诊断 13-18
  - 使用 SFC 90"H\_CTRL"控制 H 系统中的操作 27-1
  - 使用 SFC1"READ\_CLK"读取时间 5-2
  - 使用 SFC104"CiR"控制 CiR 4-4
  - 使用 SFC17"ALARM\_SQ"生成可确认的与块相关的消息以及使用 SFC18"ALARM\_S"生成永久确认的与块相关的消息 22-29
  - 使用 SFC19"ALARM\_SC"查询最后一个 ALARM\_SQ 进入事件消息的确认状态 22-32
  - 使用 SFC73 "I\_PUT"将数据写入到本地 S7 站内的一个通讯伙伴 20-18
  - 使用 SFC9 22-22
  - 使用 SFC 生成与块相关的消息简介 22-26
  - 时钟 5-1
    - 同步 5-1
    - 主站 5-1
  - 时钟中断 9-1、9-2、9-3、9-4、9-5、9-6、9-7、12-1
    - OB 9-1
    - 调用条件 9-1
    - 使用 SFC 28 SET\_TINT 设置 9-4
    - 使用 SFC 29 CAN\_TINT 取消 9-5
    - 使用 SFC 30 ACT\_TINT 激活 9-6
    - 使用 SFC 31 QRY\_TINT 查询 9-7
    - 冷重启 9-2
    - 暖重启 9-2
    - 状况影响 9-2
  - 时钟中断 OB(OB 10到 OB 17)) 1-6
  - 数据的不对等的发送 19-9
    - 通过 FB 8 19-9
    - 通过 SFB 8 19-9
  - 数据的不对等的接收 19-12
    - 通过 FB 9 19-12
    - 通过 SFB 9 19-12
  - 数据记录 7-1、7-2、7-11、7-13、7-14、7-15、7-16、7-17、8-1、8-2、8-3、8-4
    - 使用 SFC 58 WR\_REC 写 7-11
    - 使用 SFC 59 RD\_REC 7-13
    - 写入 7-1
    - 用 SFB 52 RDREC 从 DP 从站读取 8-1
    - 用 SFB 53 WRREC 对 DP 从站进行写操作 8-3
    - 读取 7-1
  - 数据记录的写和读 7-1
  - 数据块 3-8、3-10、3-19、3-20、3-21、3-22
    - 使用 SFC 22 CREAT\_DB 创建 3-8
    - 使用 SFC 23 删除 3-10
    - 使用 SFC 83 "READ\_DBL"从装载存储器的数据块中读取数据 3-19
  - 双向 DP 从站 31-33
- ## T
- 替换值 3-15
    - 使用 SFC 44 REPL\_VAL 写入至 ACCU 1 3-15
  - 停止事件 32-7
  - 同步 5-1
    - 时钟 5-1
  - 同步出错 1-55、1-57、11-1、11-10、11-11、32-3
    - OB121 1-55
    - OB122 1-57
    - 使用 SFC 36 MSK\_FLT 屏蔽 11-10
    - 使用 SFC 37 DMSK\_FLT 取消屏蔽 11-11
  - 同步周期性中断 OB(OB61到 OB 64) 1-21
  - 通讯 12-2、31-21
    - 出错 12-1
    - 中断 12-2
    - 状态数据 31-21
  - 通讯 SFC 18-9、18-10、18-11
  - 通讯 SFC 的公用参数 20-1
  - 通讯出错组织块(OB87) 1-45
  - 通讯冗余出错 OB(OB73) 1-27
  - 通讯事件 32-11
  - 通过 SFB 14 19-24
  - 通过 SFB 15 19-21
  - 通过 SFB 20 "STOP"将远程设备切换到 STOP 状态 19-36
  - 通过 SFB 21 "RESUME"在远程设备上开始一个热重启 19-38
  - 通过 SFB 8/FB 8"USEND"进行无协调的数据发送 19-9
  - 通过 SFB/FB 12 "BSEND"发送分段数据 19-15
  - 通过 SFB/FB 13 "BRCV"接收分段数据 19-18
  - 通过 SFB/FB 14"GET"从远程 CPU 中读取数据 19-24
  - 通过 SFB/FB 14 从远程 CPU 中读取数据 19-24
  - 通过 SFB/FB 15"PUT"向远程 CPU 写入数据 19-21
  - 通过 SFB/FB 15 向远程 CPU 写入数据 19-21
  - 通过 SFB/FB 9 "URCV"进行无协调的数据接收 19-12
  - 通过 SFC 62"CONTROL"查询属于 SFB 实例的连接状态 19-44
  - 通过 SFC 67 "X\_GET"从本地 S7 站以外的通讯伙伴中读取数据 20-13
  - 通过 SFC 68 "X\_PUT"将数据写入本地 S7 站以外的通讯伙伴 20-11
  - 通过 SFC74 "I\_ABORT"中止已存在的、到本地 S7 站内的通讯伙伴的连接 20-20
  - 通过输出参数 RET\_VAL 判断出错 2-1
  - 通过用户程序控制计数器 26-23

通过用户程序控制脉宽调制 26-32  
 通过用户程序控制频率测量 26-28  
 通过用户程序使用模拟量输出控制定位 26-1  
 通过用户程序使用数字量输出控制定位 26-13  
 通道专有诊断数据的结构 30-4

## W

外部出错 30-2  
 无参数分配 30-2  
 无辅助电压 30-2  
 无前连接器 30-2  
 完全重启 1-50、19-33、19-34

## X

系统区域 31-10  
 系统数据 31-1  
 系统诊断 13-1  
 系统状态列表 31-1、31-5  
   部分列表 31-5  
 系统状态列表(SSL)概述 31-1  
 写 16-20、16-21、16-22  
   将连续数据入 DP 标准从站 16-20  
 写出错 11-8  
   情景块 11-1  
   数据块 11-8  
 序列发生器 14-11、14-12  
   实现 14-11

## Y

异步 SFC 的 REQ、  
   RET\_VAL 和 BUSY 参数的含义 2-4  
 异步出错 1-28、12-1、12-3、12-4、12-5、  
 12-7、12-8、32-4  
   OB80 1-28  
   使用 SFC 39 DIS\_IRT 禁用 12-3  
   使用 SFC 40 EN\_IRT 启用 12-5  
   使用 SFC 41 DIS\_AIRT 延迟 12-7  
   使用 SFC 42 EN\_AIRT 启用 12-8  
 延迟和禁止中断以及异步出错 12-1  
 延迟时间 10-1  
 延时用户程序 4-2  
   使用 SFC 47 WAIT 4-2  
 延时中断 1-10、10-1、10-2、10-3、10-4、  
 10-5、  
 12-1  
   调用条件 10-1  
   使用 SFC 32 SRT\_DINT 启动 10-3  
   使用 SFC 33 CAN\_DINT 取消 10-5  
   使用 SFC 34 QRY\_DINT 查询 10-4  
   在启动 OB 中启动 10-1  
   状况影响 10-1  
 延时中断 OB(OB 20到 OB 23) 1-10  
 硬件中断 12-1、30-2  
   丢失 30-2

硬件中断 OB 1-14  
 硬件中断组织块(OB40-OB47) 1-14  
 影子存储器 21-1  
 右箭头 24-23  
 优先级出错 OB 1-38  
 优先级出错组织块(OB85) 1-38  
 优先级等级 1-9、1-10、1-12、1-14、1-19、  
 1-22、1-24、1-29、1-30、1-32、1-37、1-38  
 、1-42、1-45、1-47、1-48、1-50、1-55、1-  
 57、11-12、13-1、32-7

### OB 的类型

OB121 1-55  
 OB122 1-57  
 OB35 1-12  
 OB80 1-28  
 OB81 1-30  
 OB82 1-32  
 OB85 1-38  
 用 SFB 54 "RALRM" STATUS[3]从 DP 从站接收  
 中断 8-5  
 用 SFB 75 "SALRM"向 DP 主站发送中断 8-14  
 用 SFC 12 "D\_ACT\_DP"激活和取消激活 DP 从  
 站 16-10  
 用 SFC 57 "PARM\_MOD" 将参数分配给模块  
 7-8  
 用户信息 30-2  
 用户存储器 3-13  
   使用 SFC 25 压缩 3-13  
 用于 S7 通讯的 SFB/FB 和 SFC/FC 的公用参数  
 19-1  
 用于 S7 通讯的 SFB 对故障如何反应 19-7  
 用于生成与块相关的消息的 SFB 如何响应故障  
 22-25  
 用于生成与块相关的消息的 SFB 的启动特性  
 22-24  
 用于未组态 S7 连接的通讯 SFC 18-9、18-10、  
 18-11  
   分类 18-8  
 用于未组态 S7 连接的通讯 SFC 的出错信息  
 20-2  
 用于组态的 S7 连接的 SFB 启动例行程序 19-5  
 远程伙伴的状态 19-40  
   查询 19-40  
 运行时间定时器 6-1  
 运行系统定时器 6-1、6-4、6-5、6-6  
   使用 SFC 2 SET\_RTM 设置 6-4  
   使用 SFC 3 CTRL\_RTM 启动 6-5  
   使用 SFC 3 CTRL\_RTM 停止 6-5  
   使用 SFC 4 READ\_RTM 读出 6-6  
   取值范围 6-1  
   特征 6-1

## Z

在 H 系统中使用 SFC 90 控制操作 27-1  
 再触发循环时间监控 4-1  
   使用 SFC 43 RE\_TRIGR 4-1  
 诊断缓冲区 11-1、31-1、31-46

- 诊断事件 32-14
  - 诊断数据 7-2、30-1、30-2、31-1、31-2
    - 的状态 31-1
    - 内容 30-1
    - 信号模块的 7-1、31-1
    - 结构 30-1
  - 诊断数据结构概述 30-1
  - 诊断中断 12-2、30-2
    - 来自替换件 30-2
  - 诊断中断组织块(OB82) 1-32
  - 制造商特定中断 1-18
  - 制造商特定中断的 OB(OB57) 1-18
  - 中断 1-6、1-10、1-11、1-12、1-13、1-14、1-15、1-16、1-17、1-18、1-34、12-1、12-3、12-4、12-5、12-7、12-8
    - DPV1 中断 1-16、1-17、1-18
    - 插入/拆卸中断 1-34
    - 等级 12-1、12-2
    - 更新中断 1-17
    - 使用 SFC 39 DIS\_IRT 禁用 12-3
    - 使用 SFC 40 EN\_IRT 启用 12-5
    - 使用 SFC 41 DIS\_AIRT 延迟 12-7
    - 使用 SFC 42 EN\_AIRT 启用 12-8
    - 循环(OB35) 1-12
    - 延时 1-10、1-11
    - 硬件中断 OB 1-14、1-15
    - 制造商特定中断 1-18
    - 状态中断 1-16
  - 中断 OB 1-32
    - 日时钟中断 1-7
    - 诊断中断 1-32
  - 中断等级 1-34
  - 周期监视时间 30-2
  - 周期性中断 1-12、12-2
    - OB35 1-12、1-13
  - 周期性中断组织块(OB30到 OB38) 1-12
  - 主时钟 5-1
  - 转换数据类型格式 24-24、24-25、24-26
  - 状况影响
    - 执行和响应 9-2
  - 状态中断 1-16
  - 状态中断 OB(OB 55) 1-16
  - 子钟 5-3
    - 同步 5-3
  - 自由用户事件 32-17
  - 组合代码块
    - 调用 29-1
  - 组件标识 31-15
  - 组态 30-4
    - 出错
      - 模拟量输出模块 30-4
      - 数字量输入模块 30-4
      - 模拟量输入模块 30-4
  - 组织块(OB) 1-6、1-10、1-12、1-16、1-17、1-18、1-19、1-28、1-30、1-32、1-34、1-37、1-38、1-42、1-45、1-47、1-48、1-50、1-55、1-57
  - CPU 硬件故障 OB(OB84) 1-37
  - OB121 1-55
  - OB122 1-57
  - 背景 OB(OB90) 1-48
  - 插入/删除 OB(OB ) 1-34
  - 重新启动 OB(OB101) 1-50
  - 处理中断 OB(OB 88) 1-47
  - 多值计算中断 OB(OB60) 1-19
  - 更新中断 OB(OB 56) 1-17
  - 机架故障 OB(OB86) 1-42
- 类型
- OB35 1-12
  - OB80 1-28
  - OB81 1-30
  - OB82 1-32
  - OB85 1-38
- 启动 OB(OB100 和 101) 1-50
- 时钟中断 OB(OB 10到 OB 17) 1-6
- 通讯出错 OB(OB87) 1-45
- 完全重启 OB(OB100) 1-50
- 延时中断 OB(OB20到 OB23) 1-10
- 优先级出错 OB(OB85) 1-38
- 诊断中断 OB(OB82) 1-32
- 状态中断 OB(OB 55) 1-16
- 制造商特定中断的 OB(OB57) 1-18
- 组织块(OB)概述 1-1

