

# SIEMENS

## SIMATIC

### S7-300/400的系统软件和标准功能

参考手册

12/2002 版

前言	1
组织块	2
SFC 常用参数	3
拷贝与块功能	4
用于控制程序执行的 SFCS	5
SFCS 控制系统时钟	6
SFCS 控制运行时间定时器	7
用于传送数据记录的 SFC	8
根据 PNO AK1131 的 DPV1 SKB	9
用于日时钟中断操作的 SFC	10
处理延时中断 SFC	11
处理同步故障的 SFC	12
处理中断和异步故障的 SFC	13
用于诊断的 SFC	14
用于刷新过程映象和处理位区域的 SFC 与 SFB	15
模板寻址的系统功能	16
分布式 I/O 使用的 SFC	17
用于全局数据通讯的 SFC	18
S7 通讯及 S7 基础通讯概述	19
S7 通讯	20
SFC 非组态 S7 连接通讯	21
生成块相关的信息	22
IEC 定时器和 IEC 计数器	23
IEC 功能集	24
用于集成控制功能的 SFB	25
用于紧凑型 CPU 的 SFB	26
用于 H CPUS 的 SFCS	27
集成功能（用于带集成 I/O 的 CPUS）	28
塑料工艺	29
诊断数据	30
系统状态列表（SSL）	31
事件	32
SFC 和 SFB 表	
术语	



# 前言

## 目的

本手册旨在提供全面的组织块(OB)、系统功能(SFC)、系统和标准功能(SFC)和 S7-300、S7-400 的 CPU 操作系统里的 IEC 功能。在附录中描述了诊断数据、系统状态列表 (SZL) 以及事件。

## 注意

关于 CPU 中提供的这些功能与块的详细信息，请参见“S7-300 可编程控制器，硬件和安装手册”/70/或“S7-400/M7-400 可编程控制器模板规范手册”参考手册/101/和指令表 7-400 可编程控制器/102/（相关 CPU 版本）中关于这些功能的详细描述。关于相关 CPU 的 CFB 和 S7 信号功能特性，请参见/70/和/101/中详细描述。

关于CPU操作系统、编程设计和通讯、诊断功能，请参考“硬件配置和通讯连接STEP 7 V5.2”手册/234。如何在程序中调用功能和功能块，参见编程语言解释。

使用STEP 7 标准软件，可以对这些功能进行编程和参数赋值，关于该软件的使用，请参见“STEP 7 V5.2 编程手册”/231/和STEP 7 的在线帮助。

## 读者

这本手册是为熟悉控制过程的编程人员和工程师编写的，这些人员将对可编程控制器编写程序。

## STEP 7 文件包

打印版本 06/2000 是文件包“STEP 7 基本信息”的一部分。

下列表格显示 STEP 7 文件的概缆：

文件	目的	定货号
STEP 7 基本信息 <ul style="list-style-type: none"><li>STEP 7 V5.2 入门手册</li><li>STEP 7 V5.2 编程</li><li>STEP 7 V5.2 硬件配置和通讯连接</li><li>从 S5 到 S7，程序变换手册</li></ul>	为技术人员提供基本信息，实现运用 S7-300/400 可编程控制器完成控制任务。	6ES7810-4CA06-8BA0
STEP 7 参考资料 <ul style="list-style-type: none"><li>S7-300/400 手册梯形图 (LAD) /功能块图 (FBD) /语句表 (STL)</li><li>S7-300/400 标准和系统功能</li></ul>	提供参考信息和对于编程语言 LAD, FBD, STL、标准和系统功能的扩展描述。	6ES7810-4CA06-8BR0

在线帮助	目的	定货号
STEP 7 帮助	STEP 7 编程和硬件组态基本信息在线帮助。	STEP 7 标准软件的一部分
STL/LAD/FBD 参考帮助 SFB/SFC 参考帮助 组织块参考帮助	上下文相关的参考信息	STEP 7 标准软件的一部分

## 在线帮助

手册被整合在软件中的在线帮助中。

当使用软件的时候，在线帮助会提供给你详细的支持。

这个帮助系统通过数个界面被集成在软件中：

- 有一些菜单指令你能在帮助菜单中选择：内容指令可打开 STEP 7 的帮助索引。
- 运用帮助可提供详细的在线帮助说明。
- 上下文的帮助提供关于现在的上下文的资讯。例如，一个开着的对话框或一个打开着的窗口。按“帮助”按钮或按 F1 键可获取帮助。
- 状态条提供上下文帮助另外的一种形式。当鼠标指针被放置在菜单命令的时候，将显示每个菜单命令的简短解释。
- 当鼠标指针被放置在图标上，少时，可显示工具栏中每个图标的简短解释。

如果你喜欢以打印格式阅读自在线帮助信息，你可打印出具体的帮忙主题、手册或整个的在线帮助。

这本手册是摘录自 STEP 7 的 HTML 帮助文件。它和在线帮助几乎为同一结构，所以可很容易地在手册和在线帮助之间进行切换。

## 文件反馈

为了为您及 STEP 7 的将来用户提供更好的文件，我们需要你的支持。如果你有与这一本手册或在线帮助有关的任何提议，请填写手册最后的调查表，并寄到所述地址。若附上您对文件的评价，将不胜感谢。

## 其它手册

各种不同的 S7-300 和 S7-400 CPU 和 S7-300 和 S7-400 模块在下列手册中进行阐述：

- S7-300 可编程序控制器，参见手册：“S7 300 可编程序控制器，硬件和安装”/70/，“S7-300，M7-300 可编程序控制器模板规范”/71/和指令表/72/。
- S7-400 可编程序控制器，参见手册：“S7 -400 可编程序控制器，硬件和安装”，“S7-400/M7-400 可编程序控制器模板规范”/101/和指令表/102/。

## 如何使用这一本手册

这一本手册包括下列各项主题：

- 第一章解释全部组织块的功能；
- 第二章描述通用参数 RET\_VAL、REQ 和 BUSY；
- 第三章到第二十八章描述 SFC、SFB 和 IEC-FC；
- 第二十九章到第三十二章描述诊断数据的结构、SZL-Id 的总览、可能的事件，以及本手册所涉及的 SFC，SFB 和 FC 列表，SDB 的总览；
- 包含其它手册列表的参考书目；
- 解释重要术语的术语表；
- 帮助你快速查找章节和主题的索引。

## 约定

其它手册和文件参考通过“/.../”中的数字表示。这些数字是指参考书目中所列手册的标题。

#### 特殊注意事项

系统功能可以被中断。如果对某 SFC 或某种情况有限制，请参见具体的 SFC 说明。

#### 培训中心

西门子提供一些培训课程帮助大家熟悉 SIMATIC S7 自动控制系统。细节请联系本地区培训中心或与德国纽伦堡 D 90327 培训中心联系：

电话：+49 (911) 895-3200

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

# 目录

<b>1</b>	<b>组织块</b> .....	<b>1-1</b>
1.1	组织块总览 (OB) .....	1-1
1.2	程序循环组织块 (OB1) .....	1-3
1.3	日期时间中断组织块 (OB10 到 OB17) .....	1-4
1.4	延时中断组织块 (OB20 到 OB23) .....	1-7
1.5	循环中断组织块 (OB30 到 OB38) .....	1-8
1.6	硬件中断组织块 (OB40 到 OB47) .....	1-9
1.7	状态中断组织块 (OB 55) .....	1-11
1.8	更新中断组织块 (OB 56) .....	1-12
1.9	属于制造厂商中断组织块 (OB57) .....	1-13
1.10	多处理器断组织块 (OB60) .....	1-14
1.11	周期同步中断 OB (OB 61 至 OB 64) .....	1-15
1.12	I/O 冗余故障 OB (OB70) .....	1-16
1.13	CPU 冗余故障 OB (OB72) .....	1-18
1.14	通讯冗余故障 OB (OB73) .....	1-20
1.15	时间故障组织块 (OB80) .....	1-21
1.16	电源故障组织块 (OB81) .....	1-23
1.17	诊断中断组织块 (OB82) .....	1-25
1.18	插入/移出模板中断组织块 (OB83) .....	1-26
1.19	CPU 硬件故障组织块 (OB84) .....	1-29
1.20	优先级故障组织块 (OB85) .....	1-29
1.21	机架故障组织块 (OB86) .....	1-33
1.22	通讯故障组织块 (OB87) .....	1-36
1.23	过程中断 OB (OB 88) .....	1-37
1.24	背景组织块 (OB90) .....	1-39
1.25	启动组织块 (OB100, OB101 和 OB102) .....	1-40
1.26	编程故障组织块 (OB121) .....	1-44
1.27	I/O 访问故障组织块 (OB122) .....	1-46
<b>2</b>	<b>SFC 常用参数</b> .....	<b>2-1</b>
2.1	利用输出类型参数 RET_VAL 作错误诊断.....	2-1
2.2	异步 SFC 中 REQ, RET_VAL 和 BUSY 参数的含义.....	2-4
<b>3</b>	<b>拷贝与块功能</b> .....	<b>3-1</b>
3.1	用 SFC20 “BLKMOV” 拷贝变量 .....	3-1
3.2	用 SFC81 “UBLKMOV” 不间断地拷贝变量.....	3-3
3.3	用 SFC21 “FILL” 初始化存储区 .....	3-4
3.4	用 SFC22 “CREAT_DB” 生成数据块.....	3-5
3.5	用 SFC23 “DEL_DB” 删除数据块 .....	3-7
3.6	用 SFC24 “TEST_DB” 测试数据块.....	3-8
3.7	用 SFC25 “COMPRESS” 压缩用户存储器.....	3-8

3.8	用 SFC44 “REPL_VAL” 传送一个替代值到累加器 1 .....	3-9
3.9	用 SFC82 “CREA_DBL” 在装载存储器中生成数据块 .....	3-10
3.10	用 SFC83 “READ_DBL” 从装载存储器的数据块中读取数据 .....	3-12
3.11	用 SFC84 “WRIT_DBL” 写数据到装载存储器中的数据块 .....	3-14
<b>4</b>	<b>用于控制程序执行的 SFC .....</b>	<b>4-1</b>
4.1	SFC43 “RE_TRIGR” 再触循环时间监控 .....	4-1
4.2	SFC46 “STP” 使 CPU 进入停机状态 .....	4-1
4.3	SFC 47 延时用户程序执行 .....	4-2
4.4	SFC35 “MP_ALM” 触发多 CPU 中断 .....	4-2
4.5	SFC104 “CiR” 控制 CiR .....	4-3
<b>5</b>	<b>SFC 控制系统时钟 .....</b>	<b>5-1</b>
5.1	SFC “SET_CLK” 设定 TOD .....	5-1
5.2	SFC1 “READ_CLK” 读取时间 .....	5-1
5.3	SFC48 “SNC_RTCB” 同步子时钟 .....	5-2
5.4	SFC100 “SET_CLKS” 设定日期时间和时间日期状态 .....	5-3
<b>6</b>	<b>SFC 控制运行时间定时器 .....</b>	<b>6-1</b>
6.1	运行时间定时器 .....	6-1
6.2	SFC101 “RTM” 控制运行时间定时器 .....	6-2
6.3	SFC2 “SET_RTM” 设置运行时间定时器 .....	6-3
6.4	SFC3 “CTRL_RTM” 启动, 运行时间定时器启/停 .....	6-3
6.5	SFC4 “READ_RTM” 读取运行时间定时器 .....	6-4
6.6	SFC64 “TIME_TCK” 读取系统时间 .....	6-5
<b>7</b>	<b>用于传送数据记录的 SFC .....</b>	<b>7-1</b>
7.1	数据记录的写和读 .....	7-1
7.2	利用 SFC 54 “RD_DPARM” 读取定义的参数 .....	7-2
7.3	利用 SFC 102 “RD_DPARA” 读取定义的参数 .....	7-3
7.4	利用 SFC 55 “WR_PARM” 写动态数据 .....	7-4
7.5	利用 SFC 56 “WR_DPARM” 写默认的参数 .....	7-5
7.6	利用 SFC 57 “PARM_MOD” 为模块指派参数 .....	7-6
7.7	利用 SFC 58 “WR_REC” 写数据记录 .....	7-8
7.8	利用 SFC 59 “RD_REC” 读数据记录 .....	7-9
7.9	关于 SFC 55 至 59 进一步的出错信息 .....	7-12
<b>8</b>	<b>根据 PNO AK1131 的 DPV1 SKB .....</b>	<b>8-1</b>
8.1	利用 SFB 52 “RDREC” 读来自 DP 从站的数据记录 .....	8-1
8.2	利用 SFB53 “WRREC” 向 DP 从站写数据记录 .....	8-2
8.3	利用 SFB 54 “RALRM” 接收来自 DP 从站的中断 .....	8-3
8.4	利用 SFB 75 “SALRM” 向 DP 从站发送中断 .....	8-11
<b>9</b>	<b>用于日时钟中断操作的 SFC .....</b>	<b>9-1</b>
9.1	日时期中断 .....	9-1
9.2	SFC28~SFC31 描述 .....	9-2
9.3	利用 SFC28 “SET_TINT” 设置日时钟中断 .....	9-3

9.4	利用 SFC29 “CAN_TINT” .....	9-4
9.5	利用 SFC30 “ACT_TINT” 启动日时钟中断 .....	9-4
9.6	利用 SFC 31 “QRY_TINT” 查询日时钟中断 .....	9-5
<b>10</b>	<b>处理延时中断 SFC .....</b>	<b>10-1</b>
10.1	处理延时中断 .....	10-1
10.2	用 SFC32 “SRT_DINT” 启动延时中断 .....	10-2
10.3	用 SFC 34 “QRY_DINT” 查询一个延时中断 .....	10-3
10.4	用 SFC33 “CAN_DINT” 取消一个延时中断 .....	10-4
<b>11</b>	<b>处理同步故障的 SFC .....</b>	<b>11-1</b>
11.1	屏蔽同步故障 .....	11-1
11.2	用 SFC 36 “MSK_FLT” 屏蔽同步故障 .....	11-7
11.3	用 SFC 37 “DMSK_FLT” 解除同步故障的屏蔽 .....	11-8
11.4	用 SFC 38 “READ_ERR” 读故障寄存器中信息 .....	11-9
<b>12</b>	<b>处理中断和异步故障的 SFC .....</b>	<b>12-1</b>
12.1	延迟和去活中断和异步故障 .....	12-1
12.2	用 SFC 39 “DIS_IRT” 去活新的中断和异步故障的处理 .....	12-2
12.3	用 SFC 40 “EN_IRT” 激活新中断和异步故障的处理 .....	12-4
12.4	用 SFC 41 “DIS_AIRT” 推迟一个高优先权的中断和异步故障的处理 .....	12-5
12.5	用 SFC 42 “EN_AIRT” 激活具有高优先权的中断和异步故障的处理 .....	12-6
<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	由 SFC 6 “RD_SINFO” 读取 OB 启动信息 .....	13-1
13.3	由 SFC 51 “RDSYSST” 读取系统状态信息或者部分表 .....	13-3
13.4	用 SFC 52 “WR_USMSG” 在诊断缓冲器中写入一个用户定义的诊断事件 .....	13-9
13.5	用 SFC 78 “OB_RT” 确定 OB 程序运行时间 .....	13-12
13.6	用 SFC 87 “C_DIAG” 诊断当前的连接状态 .....	13-14
13.7	通过 SFC 103 “DP_TOPOL” 识别 DP 主站系统的总线拓扑 .....	13-18
<b>14</b>	<b>用于刷新过程映象和处理位区域的 SFC 与 SFB .....</b>	<b>14-1</b>
14.1	用 SFC 26 “UPDAT_PI” 刷新过程映象输入表 .....	14-1
14.2	用 SFC 27 刷新过程映象输出表 .....	14-2
14.3	用 SFC 126 “SYNC_PI” 同步刷新过程映象区输入表 .....	14-3
14.4	用 SFC 127 “SYNC_PO” 同步刷新过程映象区输出表 .....	14-4
14.5	用 SFC 79 “SET” 置位 I/O 区域中的位区域 .....	14-5
14.6	用 SFC 80 “REST” 复位 I/O 区域中的位区域 .....	14-6
14.7	用 SFB 32 “DRUM” 执行一段顺序程序 .....	14-7
<b>15</b>	<b>寻址模板的系统功能 .....</b>	<b>15-1</b>
15.1	用 SFC 5 “GADR_LGC” 查询模板的逻辑起始地址 .....	15-1
15.2	用 SFC 49 “LGC_GADR” 查询逻辑地址所属的插槽 .....	15-2
15.3	用 SFC 50 “RD_LGADR” 查询一个模板所有的逻辑地址 .....	15-4
<b>16</b>	<b>分布式 I/O 使用的 SFC .....</b>	<b>16-1</b>
16.1	SFC 7 “DP_PRAL”：在 DP 主站上触发硬件中断 .....	16-1



---

16.2	SFC 11 “DPSYC_FR”：同步 DP 从主站组.....	16-3
16.3	SFC 12 取消和激活 DP 从站.....	16-7
16.4	使用 SFC 13 “DPNRM_DG” 读 DP 从站诊断数据（从站诊断）.....	16-10
16.5	使用 SFC 14 “DPRD_DAT” 读取 DP 标准从站的连续数据.....	16-12
16.6	使用 SFC 15 “DPWR_DAT” 向 DP 标准从站写连续数据.....	16-14
17	用于全局数据通讯的 SFC.....	17-1
17.1	利用 SFC 60 “GD_SND” 传送一个全局数据包.....	17-1
17.2	利用 SFC 61 “GD_RCV” 接收全局数据包.....	17-3
18	S7 通讯及 S7 基础通讯概述.....	18-1
18.1	S7 通讯及 S7 基础通讯所使用块的区别.....	18-1
18.2	数据连续性.....	18-3
18.3	S7 通讯块的概述.....	18-4
18.4	S7 基础通讯块的概述.....	18-6
19	S7 通讯.....	19-1
19.1	用于 S7 通讯的 SFC/FC，SFB/FB 常见参数.....	19-1
19.2	SFB 需组态连接的启动路线.....	19-3
19.3	SFB 对于错误的反映.....	19-5
19.4	使用 SFB/FB 8 U_SEND 非协调发送数据.....	19-6
19.5	使用 SFB/FB 9 U_RECV 非协调接受数据.....	19-8
19.6	用 SFB/FB 12 B_SEND 发送段数据.....	19-10
19.7	用 SFB/FB 13 B_RCV 接收段数据.....	19-12
19.8	使用 SFB/FB 15 “PUT” 写数据到远端 CPU.....	19-15
19.9	使用 SFB/FB 14 “GET” 读来自远端 CPU 数据.....	19-17
19.10	使用 SFB 16 “PRINT” 发送数据到打印.....	19-19
19.11	用 SFB 19 “START” 在远端设备上初始化一个暖冷启动.....	19-25
19.12	用 SFB 20 “STOP” 在改变远端设备到停止状态.....	19-27
19.13	用 SFB 21 “RESUME” 在远端设备上初始化一个 HOT RESTART.....	19-28
19.14	用 SFB 22 “STATUS” 块询问远端对象的状态.....	19-30
19.15	用 SFB 23 “USTATUS” 接收远端设备的状态.....	19-32
19.16	用 SFB 62 “CONTROL” 查询所属连接的状态.....	19-34
19.17	使用 FC 62 “C_CNTRL” 查询连接状态.....	19-36
19.18	S7 SFB/FB 通讯的工作存储器需求.....	19-37
20	SFC 非组态 S7 连接通讯.....	20-1
20.1	SFC 通讯的通用参数.....	20-1
20.2	SFC 非组态 S7 连接通讯的错误信息.....	20-2
20.3	使用 SFC65 “X_SEND” 发送数据到外部通讯对象 S7 站.....	20-5
20.4	使用 SFC66 “X_RCV” 接收来自外部通讯对象 S7 站的数据.....	20-6
20.5	使用 SFC68 “X_PUT” 写数据到外部通讯对象 S7.....	20-9
20.6	使用 SFC67 “X_GET” 读来自外部通讯对象 S7 站的数据.....	20-10
20.7	使用 SFC69 “X_ABORT” 中断一个连到外部 S7 站的已存在连接.....	20-11
20.8	使用 SFC73 “I_PUT” 写数据到内部通讯对象 S7 站.....	20-12
20.9	使用 SFC72 “I_GET” 读来自内部通讯对象 S7 站的数据.....	20-13

20.10	使用 SFC72 “I_ABORT” 中断一个已存在到本方 S7 站的连接 .....	20-15
<b>21</b>	<b>生成块相关的信息 .....</b>	<b>21-1</b>
21.1	用 SFB 生成块相关信息简介 .....	21-1
21.2	用 SFB 36 “NOTIFY” 生成无需确认的块相关的信息 .....	21-4
21.3	用 SFB 31 “NOTIFY_8P” 生成无需确认的块相关的信息 .....	21-6
21.4	用 SFB 33 “ALARM” 生成需确认的块相关的信息 .....	21-8
21.5	用 SFB 35 “ALARM_8P” 生成 8 个信号的带相关数据的块相关的信息 .....	21-10
21.6	用 SFB 34 “ALARM_8” 生成 8 个信号的不带相关数据的块相关的信息 .....	21-12
21.7	用 SFB 37 “AR_SEND” 发送存档数据 .....	21-14
21.8	用 SFC 10 “DIS_MSG” 禁止使能块相关的、符号相关的以及组状态信息 .....	21-16
21.9	用 SFC 9 “EN_MSG” 使能块相关的、符号相关的以及组状态信息。 .....	21-18
21.10	生成块相关的信息的 SFB 的启动响应 .....	21-19
21.11	生成块相关的信息的 SFB 对故障的响应 .....	21-20
21.12	用 SFC 生成块相关的信息简介 .....	21-20
21.13	用 SFC 17 “ALARM_SQ” 生成可确认的与块相关的信息和用 SFC 18 “ALARM_S” 生成永久确认的块相关的信息 .....	21-22
21.14	用 SFC 19 “ALARM_SC” 查询最后 ALARM_SQ/ALARM_ DQ 进入事件信息的确认状态 .....	21-24
21.15	用 SFC 107 “ALARM_DQ” 与 108 “ALARM_D” 生成可确认的与永久确认的块相关的信息 .....	21-25
21.16	用 SFC 105 “READ_SI” 读取动态系统资源 .....	21-27
21.17	用 SFC 106 “DEL_SI” 删除动态系统资源 .....	21-29
<b>22</b>	<b>IEC 定时器和 IEC 计数器 .....</b>	<b>22-1</b>
22.1	利用 SFB 3 “TP” 生成一个脉冲信号 .....	22-1
22.2	利用 SFB 4 “TON” 生成一个延时接通信号 .....	22-2
22.3	利用 SFB 5 “TOF” 产生一个延时断开信号 .....	22-3
22.4	利用 SFB 0 “CTU” 实现加计数功能 .....	22-4
22.5	利用 SFB 1 “CTD” 实现成计数功能 .....	22-5
22.6	利用 SFB 2 “CTUD” 实现加/成计数功能 .....	22-5
<b>23</b>	<b>IEC 功能集 .....</b>	<b>23-1</b>
23.1	概述 .....	23-1
23.2	IEC 功能的技术数据 .....	23-2
23.3	复杂数据类型日期和时间 .....	23-4
23.4	Time-of-Day 功能 .....	23-4
23.5	DATE_AND_TIME 变量比较 .....	23-7
23.6	STRING 变量比较 .....	23-9
23.7	数值编辑 .....	23-11
23.8	STL 举例 .....	23-12
23.9	STL 举例 .....	23-13
23.10	STRING 变量编辑 .....	23-14
23.11	数据类型格式转换 .....	23-17

---

24	用于集成控制功能的 SFB.....	24-1
24.1	连续调节功能 SFB 41/FB 41 “CONT_C” .....	24-1
24.2	步进调节功能 SFB 42/FB 42 “CONT_S” .....	24-8
24.3	脉冲发生功能 SFB 43/FB 43 “PULSEGEN” .....	24-13
24.4	PULSEGEN 功能块举例.....	24-21
25	用于紧凑型 CPU 的 SFB .....	25-1
25.1	使用 SFB 44 “Analog” 实现模拟量输出定位.....	25-1
25.2	使用 SFB 46 “DIGITAL” 实现数字量输出定位.....	25-8
25.3	使用 SFB 47 “COUNT” 控制计数器.....	25-16
25.4	使用 SFB 48 “FREQUENC” 控制频率测量 .....	25-19
25.5	使用 SFB 49 “PULSE” 控制脉宽调制 .....	25-22
25.6	使用 SFB 60 “SEND_PTP” 发送数据 (ASCII, 3964 (R)) .....	25-24
25.7	使用 SFB 61 “RCV_PTP” 接收数据 (ASCII, 3964 (R)) .....	25-26
25.8	使用 SFB 62 “RES_RCVB” 清除接收缓冲区 (ASCII, 3964 (R)) .....	25-28
25.9	使用 SFB 63 “SEND_RK” 发送数据 (512 (R)) .....	25-29
25.10	使用 SFB 64 “FETCH_RK” 获取数据 (RK 512) .....	25-32
25.11	使用 SFB 65 “SERVE_RK” 接收和提供数据 (RK 512) .....	25-36
25.12	SFB60 到 SFB65 的附加错误信息 .....	25-40
26	用于 H CPUs 的 SFC .....	26-1
26.1	使用 SFC 90 “H_CTRL” 在 H 系统中控制操作 .....	26-1
27	集成功能 (用于带集成 I/O 的 CPUs) .....	27-1
27.1	SFB 29 (HS_COUNT) .....	27-1
27.2	SFB 30 (FREQ_MES) .....	27-2
27.3	SFB 38 (HSC_A_B) .....	27-3
27.4	SFB 39 (POS) .....	27-4
28	塑料工艺 .....	28-1
28.1	SFB 63 (AB_调用) .....	28-1
29	诊断数据 .....	29-1
29.1	诊断数据的结构概览.....	29-1
29.2	诊断数据.....	29-1
29.3	通道特定的诊断数据的结构.....	29-3
30	系统状态列表 (SSL) .....	30-1
30.1	系统状态列表 (SSL) 概述.....	30-1
30.2	局部系统状态列表的结构.....	30-2
30.3	SSL-ID .....	30-3
30.4	可能出现的局部系统状态列表 .....	30-4
30.5	SSL-ID W#16#xy11 - 模块标识.....	30-5
30.6	SSL-ID W#16#xy12 - CPU 特性 .....	30-6
30.7	SL-ID W#16#xy13 - 存储区域 .....	30-8
30.8	SSL-ID W#16#xy14 - 系统区域 .....	30-9
30.9	SSL-ID W#16#xy15 - 块类型.....	30-10

30.10	SSL-ID W#16#xy19 – 模板 LED 的状态 .....	30-11
30.11	SSL-ID W#16#xy1C – 元件识别 .....	30-12
30.12	SSL-ID W#16#xy22 – 中断状态 .....	30-15
30.13	SSL ID W#16#xy25 – 给 OB 分配过程映像区 .....	30-17
30.14	SSL-ID W#16#xy32 – 通讯状态数据 .....	30-19
30.15	SSL-ID W#16#0132 索引号为 W#16#0005 的局部列表摘要数据记录 .....	30-20
30.16	SSL-ID W#16#0132 索引号为 W#16#0008 的局部列表摘要数据记录 .....	30-21
30.17	SSL-ID W#16#0132 索引号为 W#16#000B 的局部列表摘要数据记录 .....	30-23
30.18	SSL-ID W#16#0132 索引号为 W#16#000C 的局部列表摘要数据记录 .....	30-24
30.19	SSL-ID W#16#0232 索引号为 W#16#0004 的局部列表摘要数据记录 .....	30-25
30.20	SSL-ID W#16#xy71 – 冗余 CPU 组信息 .....	30-26
30.21	SSL-ID W#16#xy74 - 模板 LED 灯状态 .....	30-28
30.22	SSL-ID W#16#xy75 – H 系统中切换的 DP 从站 .....	30-30
30.23	SSL-ID W#16#xy90 – DP 主站系统信息 .....	30-31
30.24	SSL-ID W#16#xy91 – 模块状态信息 .....	30-33
30.25	SSL-ID W#16#xy92 – 机架/站状态信息 .....	30-36
30.26	SSL-ID W#16#xy95 – 扩展的 DP 主系统 .....	30-39
30.27	SSL-ID W#16#xyA0 – 诊断缓冲区 .....	30-41
30.28	SSL-ID W#16#00B1 – 模块诊断信息 .....	30-42
30.29	SSL-ID W#16#00B2 – 带物理地址的诊断数据纪录 1 .....	30-43
30.30	SSL-ID W#16#00B3 – 带逻辑基本地址的模块诊断数据 .....	30-44
30.31	SSL-ID W#16#00B4 – 一个 DP 从站的诊断数据 .....	30-45
<b>31</b>	<b>事件 .....</b>	<b>31-1</b>
31.1	事件和事件 ID .....	31-1
31.2	事件级别 1 – 标准 OB 事件 .....	31-2
31.3	事件级别 2 – 同步错误 .....	31-3
31.4	事件级别 3 – 非同步错误 .....	31-4
31.5	事件级别 4 – 停机事件和模式转换 .....	31-6
31.6	事件级别 5 – 模式运行时间事件 .....	31-9
31.7	事件级别 6 – 通信事件 .....	31-10
31.8	事件级别 7 – H/F 事件 .....	31-12
31.9	事件级别 8 – 模块的诊断事件 .....	31-14
31.10	事件级别 9 – 标准用户事件 .....	31-16
31.11	事件级别 A 和 B – 自由用户事件 .....	31-18
31.12	保留的事件级别 .....	31-18
<b>32</b>	<b>SFC 和 SFB 表 .....</b>	<b>32-1</b>
32.1	SFC 表, 按编号排序 .....	32-1
32.2	SFC 表, 按字母排序 .....	32-4
32.3	SFB 表, 按编号排序 .....	32-7
32.4	SFB 表, 按字母排序 .....	32-9
<b>术语</b>	<b>.....</b>	<b>术语-1</b>

# 1 组织块

## 1.1 组织块总览（OB）

### 什么是组织块？

组织块是在操作系统和用户程序之间的接口。OB 用于执行具体的程序部分：

- 在 CPU 启动时
- 在一个循环或时钟执行时
- 当发生故障时
- 当发生硬件中断时

组织块根据其优先级执行。

### 可使用哪些组织块？

不是所有的 CPU 都能处理所有 STEP 7 中的组织块。至于在 CPU 中包括哪些 OB，请参见操作列表/72/和/102/。

### 哪可以找到更多信息？

参见在线帮助和以下手册可以得到更多的信息：

- /70/：该手册包含有描述不同 S7-300 CPU 功能的技术数据。
- /101/：该手册包含有描述不同 S7-400 CPU 功能的技术数据。

下列表格中包含每一个 OB 的启动事件及对应的优先级。

OB	启动事件	默认优先级	解释
OB1	启动结束或 OB1 执行结束	1	自由循环
OB10	日期时间中断 0	2	没有指定时间
OB11	日期时间中断 1	2	
OB12	日期时间中断 2	2	
OB13	日期时间中断 3	2	
OB14	日期时间中断 4	2	
OB15	日期时间中断 5	2	
OB16	日期时间中断 6	2	
OB17	日期时间中断 7	2	
OB20	延时中断 0	3	没有指定时间
OB21	延时中断 1	4	
OB22	延时中断 2	5	
OB23	延时中断 3	6	

OB	启动事件	默认优先级	解释
OB30	循环中断 0 (默认时间间隔: 5s)	7	循环中断
OB31	循环中断 1 (默认时间间隔: 2s)	8	
OB32	循环中断 2 (默认时间间隔: 1s)	9	
OB33	循环中断 3 (默认时间间隔: 500ms)	10	
OB34	循环中断 4 (默认时间间隔: 200ms)	11	
OB35	循环中断 5 (默认时间间隔: 100ms)	12	
OB36	循环中断 6 (默认时间间隔: 50ms)	13	
OB37	循环中断 7 (默认时间间隔: 20ms)	14	
OB38	循环中断 8 (默认时间间隔: 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	刷新中断	2	
OB57	制造厂商特殊中断	2	
OB60	SFC 35 “MP_ALM” 调用	25	多处理器中断
OB61	同步循环中断 1	25	同步循环中断
OB62	同步循环中断 2	25	
OB63	同步循环中断 3	25	
OB64	同步循环中断 4	25	
OB70	I/O 冗余故障 (只在 H 型号 CPU)	25	冗余故障中断
OB72	CPU 冗余故障 (只在 H 型号 CPU)	28	
OB73	通讯冗余故障 OB (只在 H 型号 CPU)	25	
OB80	时间故障	26, 28 <sup>1)</sup>	同步故障中断
OB81	电源故障	25, 28 <sup>1)</sup>	
OB82	诊断中断	25, 28 <sup>1)</sup>	
OB83	插/拔模板中断	25, 28 <sup>1)</sup>	
OB84	CPU 硬件故障	25, 28 <sup>1)</sup>	
OB85	程序错误	25, 28 <sup>1)</sup>	
OB86	扩展机架、DP 主站系统或分布式 I/O 从站故障	25, 28 <sup>1)</sup>	
OB87	通讯故障	25, 28 <sup>1)</sup>	
OB88	过程中断	28	
OB90	暖或冷启动或删除一个正在 OB90 中执行的块或装载一个 OB90 到 CPU 或中止 OB90	29 <sup>2)</sup>	背景循环

OB	启动事件	默认优先级	解释
OB100	暖启动	27 <sup>1)</sup>	启动
OB101	热启动	27 <sup>1)</sup>	
OB102	冷启动	27 <sup>1)</sup>	
OB121 OB122	编程错误 I/O 访问错误	引起错误 OB 的优 先级 引起错误 OB 的优 先级	同步错误中断

<sup>1)</sup> 优先级 27 和 28 在优先级启动模式中是有效的。

<sup>2)</sup> 优先级 29 对应于优先级 0.29。这意味着背景循环比自由循环具有更低的优先级。

## 1.2 程序循环组织块（OB1）

### 描述

S7 CPU 操作系统周期性地执行组织块 OB1 程序。当 OB1 执行完毕，操作系统再次启动它。CPU 启动后，OB1 被循环执行。你可以在 OB1 中调用其它功能块（FB，SFB）或功能（FC，SFC）。

### 理解 OB1 的运行

OB1 的优先级最低。其循环时间被监控。即除 OB90 以外，其它所有 OB 均可打断 OB1 的执行。

以下事件可导致操作系统调用 OB1：

- CPU 启动完毕。
- OB1 执行到上一个循环周期结束。

OB1 执行完后，操作系统发送全局数据。再次启动 OB1 之前，操作系统会将输出映像区数据写入输出模板，刷新输入映像区并接收全局数据。

S7 监视最长循环时间，保证最长的响应时间。最长循环时间被预置为 150ms。你可以设一个新值或通过 SFC43 “RE\_TRIGR” 重新启动时间监视功能。如果你的程序超过了 OB1 最长循环时间，操作系统将调用 OB80（时间错误 OB）；如果 OB80 不存在，则 CPU 停机。

除了监视最长循环时间，还可以保证最短循环时间。操作系统将延长下一个新循环（将输出映像区数据传送到输出模板）直到最短循环时间到。

参见手册/70/和/101/参数“最长”、“最短”循环时间的范围。你可以运用 STEP 7 软件更改参数设置。

### OB1 的局部数据

以下表格描述了 OB1 的临时变量（TEMP）。变量名是 OB1 的缺省名称。

变量	类型	描述
OB1_EV_CLASS	字节	事件等级和标识码: B#16#11: OB1 激活
OB1_SCAN_1	字节	<ul style="list-style-type: none"> <li>• B#16#01: 暖启动完成</li> <li>• B#16#02: 热启动完成</li> <li>• B#16#03: 主循环完成</li> <li>• B#16#04: 冷启动完成</li> <li>• B#16#05: 当前一个主站 CPU 停机, 后备新主站 CPU 的第一次 OB1 循环</li> </ul>
OB1_PRIORITY	字节	优先级 1
OB1_OB_NUMBR	字节	OB 号 (01)
OB1_RESERVED_1	字节	备用
OB1_RESERVED_2	字节	备用
OB1_PREV_CYCLE	整数	上一次 OB1 的循环时间 (ms)
OB1_MIN_CYCLE	整数	自 CPU 启动, 最短一次 OB1 的循环 (ms)
OB1_MAX_CYCLE	整数	自 CPU 启动, 最长一次 OB1 的循环 (ms)
OB1_DATE_TIME	DATE_AND_TIME	OB 被调用的日期和时间

## 1.3 日期时间中断组织块 (OB10 到 OB17)

### 描述

STEP 7 提供多达 8 个 OB (OB10 到 OB17), 它们可以运行一次或周期性地运行。你可以运用 SFC 或 STEP 7 给 CPU 分配参数, 这样 OB 将以下列的间隔运行:

- 一次
- 每一分钟
- 每小时
- 每天
- 每周
- 每月
- 每月底

### 理解日期时间中断组织块 OB 的运行

在启动日期时间中断时, 你必须首先设置和激活中断。

以下三种方式可以设置和激活中断:

- 自动启动日期时间中断。你可通过 STEP 7 设置并激活中断。下表显示通过 STEP 7 激活日期时间中断的基本可能性。
- 你可以在 STEP 7 中, 通过调用 SFC30 “ACT-TINT”, 设置日期时间中断。



- 你可以在 STEP 7 中，通过调用 SFC28 “SET\_TINT” 并调用 SFC30 “ACT\_TINT”，设置日期时间中断。

时间间隔	描述
没有激活	日期时间中断没有激活，即使已调入 CPU。它可以调用 SFC30 激活。
只激活一次	当日期时间中断运行一次后，日期时间中断 OB 被自动取消。你可以通过调用 SFC28 和 SFC30 重设和重新激活日期时间中断 OB。
周期性地激活	当日期时间中断发生后，CPU 根据当前的时间计算下一次的启动时间。

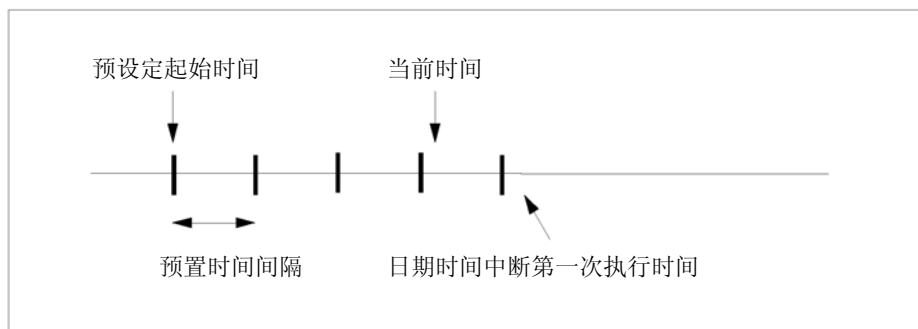
当你向前或向后调整时钟时，日期时间中断的行为请参见在/234/。

**注意**

如果你设定日期时间中断相应的 OB 是执行一次，那么日期时间 (DATE\_AND\_TIME) 不能是过去（与 CPU 的实时时钟相关）的日期时间。

如果你设定日期时间中断相应的 OB 是周期性的执行，日期时间 (DATE\_AND\_TIME) 是过去的日期和时间，那么日期时间中断将按下图所示在下次执行。

你可以用 SFC39 到 SFC42 禁止、延迟和重新使能日期时间中断。



**影响日期时间中断 OB 的条件**

因为日期时间中断只在规定的的时间间隔发生，某些条件会在程序执行时影响 OB 的运行。下面表格显示了一些影响日期时间中断 OB 运行的条件和描述。

条件	结果
你编程调用 SFC29 (CAN_TINT)，并取消日期时间中断。	操作系统清除日期时间中断的事件日期时间 (DATE_AND_TIME)。在此 OB 被调用之前，你必须重新设置并激活它。
你编程激活日期时间中断 OB，但此 OB 在 CPU 中不存在。	操作系统调用 OB85。如果 OB85 在 CPU 中不存在，CPU 将停机。
当同步或校正 CPU 系统时钟时，你设置的时间超前并跳过了日期时间中断 OB 的启动事件。	操作系统调用 OB80 并编码日期时间中断 OB 号和启动信息到 OB80 中的事件信息。 操作系统运行一次日期时间中断 OB，不考虑这个 OB 应该运行多少次。OB80 的启动事件信息显示为 OB 第一次被跳过的日期和时间 (DATE_AND_TIME)。

条件	结果
当同步或校正 CPU 系统时钟时，时间被向后设置，于是 OB 的启动事件、日期或时间被重复。	如果日期时间 OB 在时钟修改之前已激活，那么此 OB 不再被调用。
CPU 经过一个暖启动或冷启动。	任何日期时间 OB 被 SFC 配置，那么它将被列入回到 STEP7 的组态中去。 如果你组态了一个启动一次的日期时间中断 OB 在 STEP7 中，组态的时间是过去（相对于 CPU 的实时时钟），并激活它，经过一个操作系统暖启动或冷启动，这个 OB 将被调用一次。
当下一个时间间隔启动事件发生，日期时间 OB 仍被执行。	操作系统调用 OB80。如果 OB80 不存在，则 CPU 停机。 如果 OB80 被装载，那么 OB80 和日期时间中断 OB 都执行第一次，并且第二个中断请求也被执行。

#### 日期时间中断 OB 的局部数据

下列表格描述了日期时间中断的临时变量（TEMP）。变量名是 OB10 缺省名称。

变量	类型	描述
OB10_EV_CLASS	字节	事件级和识别码：B#16#11=中断激活
OB10_STRT_INFO	字节	B#16#11：启动请求 OB10 （B#16#12：启动请求 OB11） ： ： （B#16#18：启动请求 OB17）
OB10_PRIORITY	字节	分配的优先级：默认 2
OB10_OB_NUMBR	字节	OB 号（10 到 17）
OB10_RESERVED_1	字节	保留
OB10_RESERVED_2	字节	保留
OB10_PERIOD_EXE	字	OB 以特殊的间隔运行： W#16#0000：一次 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：每月底
OB10_RESERVED_3	整数	保留
OB10_RESERVED_4	整数	保留
OB10_DATE_TIME	日期和时间 （DATE_AND_TIME）	当 OB 被调用时的日期和时间。

## 1.4 延时中断组织块（OB20 到 OB23）

### 描述

S7 提供多达 4 个 OB（OB20 到 OB23），它们经过一段指定的延时后运行。每一个 OB 在调用 SFC32（SRT\_DINT）后被启动。延时时间在 SFC 参数中设定。

当你编程 SFC32（SRT\_DINT）时，需提供 OB 号、延时时间和一个用户指定的标识符。指定的延时到，OB 启动。你也可以取消还没有启动的延时中断。

### 理解延时中断 OB 的运行

当延时时间到（毫秒时间和 OB 号一起传给 SFC32）时，操作系统启动相应的 OB。

运用延时中断，你必须做下列事项：

- 你必须调用 SFC32（SRT\_DINT）；
- 你必须下载日期时间中断 OB 到 CPU 中。

延时 OB 只有在 CPU 处于运行状态时才运行。一个暖启动或冷启动清除任何延时 OB 的启动事件。如果延时中断还没有启动，你可以调用 SFC33（CAN\_DINT）去取消它的执行。

延时间隔为 1 毫秒，如果超出则再次启动调用 SFC34（QRY\_DINT）查询延时中断的状态。

如果以下事件发生，操作系统调用一个异步错误 OB：

- 如果你在调用 SFC32（SRT\_DINT）时启动的是没有下载到 CPU 中的 OB，并且你还指定了其它编号。
- 如果在一个延时中断 OB 执行未结束，下一个延时中断启动事件发生。

你可以运用 SFC 39 到 SFC 42 取消、延时和再使能延时中断。

### 延时中断组织块 OB 的局部数据

下面表格描述了延时中断 OB 的临时变量。变量名是 OB20 的缺省名称。

变量	类型	描述
OB20_EV_CLASS	字节	事件级别和识别码： B#16#11：中断激活
OB20_STRT_INF	字节	B#16#21：OB20 启动请求 (B#16#21：OB21 启动请求) (B#16#22：OB22 启动请求) (B#16#23：OB23 启动请求)
OB20_PRIORITY	字节	分配的优先级：默认值为 3（OB20）到 6（OB23）
OB20_OB_NUMBR	字节	OB 号（20 到 23）
OB20_RESERVED_1	字节	保留
OB20_RESERVED_2	字节	保留
OB20_SIGN	字	用户号：为调用 SFC32（SRT_DINT）输入参数标记
OB20_DTIME	时间	以毫秒形式组态延时时间
OB20_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

## 1.5 循环中断组织块（OB30 到 OB38）

### 描述

S7 提供多达 9 个循环中断组织块 OB（OB30 到 OB38），它们经过一段固定的时间间隔中断你的程序。下面表格显示了循环中断 OB 默认的时间间隔和优先级。

OB 号	默认的时间间隔	默认的优先级
OB30	5 秒	7
OB31	2 秒	8
OB32	1 秒	9
OB33	500 毫秒	10
OB34	200 毫秒	11
OB35	100 毫秒	12
OB36	50 毫秒	13
OB37	20 毫秒	14
OB38	10 毫秒	15

### 理解循环中断组织块 OB 的运行

循环中断 OB 的等距时间间隔，是由时间间隔和相位偏移量确定。参见/234/中 OB 启动时间、时间周期和相位偏移量的关系。

#### 注意

你必须确认循环中断 OB 每一次运行的时间一定要短于中断的间隔。如果一个循环中断 OB 没有执行完，循环中断时间到，又要求循环中断 OB 运行，则时间错误组织块 OB80 启动。循环中断导致错误程序的运行。

你可以运用 SFC39 到 42 取消或延时和再次使能循环中断。

对于你的 CPU 需参见相应其 CPU 的有关参数，时间间隔、优先级、相位偏置。你可以用 STEP7 软件修改参数设置。

### 循环中断组织块 OB 的局部数据

下面表格描述了循环中断组织块 OB 的临时变量（TEMP）。变量名称是 OB35 的缺省名称。

变量	类型	描述
OB35_EV_CLASS	字节	事件级别和识别码 B#16#11: 中断激活
OB35_STRT_INF	字节	B#16#30: 循环中断组织块 OB 的启动需求, 只对于特殊标准 (只有 H 型 CPU 并且明确地为其组态) B#16#31 : OB30 启动需求 : B#16#36 : OB35 启动需求 : B#16#39 : OB38 启动需求
OB35_PRIORITY	字节	分配的优先级: 默认 7 (OB30) 到 15 (OB38)
OB35_OB_NUMBR	字节	OB 号 (30 到 38)
OB35_RESERVED_1	字节	保留
OB35_RESERVED_2	字节	保留
OB35_PHASE_OFFSET	字	相位偏移[毫秒]
OB35_RESERVED_3	整数	保留
OB35_EXC_FREQ	整数	时间间隔, 以毫秒计
OB35_DATE_TIME	DATE_AND_TIME	当 OB 被调用时, 日期时间中的日期和时间

## 1.6 硬件中断组织块 (OB40 到 OB47)

### 描述

S7 提供多达 8 个独立的硬件中断组织块 OB (OB40 到 OB47)。

通过 STEP 7 进行参数赋值, 可以为会触发硬件中断的每一个信号模板指定以下参数。

- 哪个通道在哪种条件下触发一个硬件中断。
- 哪那一个硬件中断 OB 被分配到单独的通道组。(作为默认, 所有硬件中断被 OB40 处理)

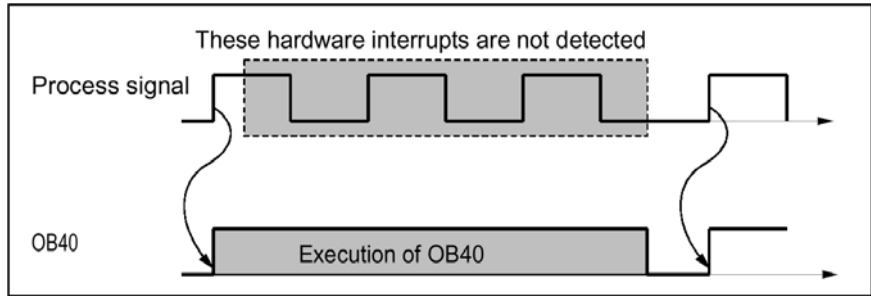
运用 CP 和 FM 模板, 你可以用它们自己的软件设置这些参数。

你可以运用 STEP 7 为每一个硬件中断 OB 选择优先级。

### 理解硬件中断组织块 OB 的运行

在硬件中断被模板触发之后, 操作系统识别相应的槽和相应的硬件中断 OB。如果这个 OB 比当前激活的 OB 优先级高, 则启动该 OB。在硬件中断 OB 执行之后, 将发送通道确认。如果在处理硬件中断的同时, 同一中断模板上有另一个硬件中断, 这个新的中断的识别与确认过程如下:

- 如果事件发生在以前触发硬件中断的通道，旧的硬件中断触发程序正在执行，则新中断丢失。如下面图所示。图中例子是一个数字量输入模板的通道。触发信号是上升沿。硬件中断 OB 是 OB40。



- 如果这个事件发生在同一模板的另一个通道，那么没有硬件中断能被触发。但是这个中断没有丢失，在确认当前激活硬件之后将被触发。

如果一个硬件中断被触发并且它的 OB 正在由于另一个模板的硬件中断而激活，则记录新的中断申请，在空闲后会执行该中断。

用 SFC 39 – 42 可以禁止、延时和再使能硬件中断。

你可以用 STEP 7 中的 SFC 55 到 57，为硬件中断模板赋值参数。

#### 硬件中断 OB 的局部数据

下列表格中描述了硬件中断 OB 的临时变量（TEMP）。变量名是 OB40 的缺省名称。

变量	类型	描述
OB40_EV_CLASS	字节	事件级别和诊断号： B#16#11：中断被激活
OB40_STRT_INF	字节	<ul style="list-style-type: none"> <li>B#16#41：中断通过中断线 1</li> <li>B#16#42：中断通过中断线 2（只对 S7-400）</li> <li>B#16#43：中断通过中断线 3（只对 S7-400）</li> <li>B#16#44：中断通过中断线 4（只对 S7-400）</li> <li>B#16#45：WinAC 通过 PC 触发的中断</li> </ul>
OB40_PRIORITY	字节	分配优先级：默认 16（OB40）到 23（OB47）
OB40_OB_NUMBR	字节	OB 号（40 到 47）
OB40_RESERVED_1	字节	保留
OB40_IO_FLAG	字节	输入模板：B#16#54 输出模板：B#16#55
OB40_MDL_ADDR	字	触发中断模块的逻辑地址
OB40_POINT_ADDR	双字	<ul style="list-style-type: none"> <li>数字模板： 带有模板输入状态的位字段（0 位对应第一个输入）</li> <li>模拟模板： 带有极限信息输入通道的位字段（结构详见/71/或/101/）</li> </ul>

变量	类型	描述
		<ul style="list-style-type: none"> <li>CP 或 IM: 模块中断状态（不是用户相关的）</li> </ul>
OB40_DATE_TIME	日期和时间	OB 被调用的日期和时间

**注意**

如果你有用 DPV1 能力的 CPU，你可以通过 SFB54 “RELRM” 获取额外的信息，此信息超出了 OB 的启动信息。当你运行工业现场总线 DP 主站在 S7 兼容模式下，此信息也会提供。

## 1.7 状态中断组织块（OB 55）

**注意**

一个状态中断组织块 OB（OB 55）只有在有 DPV1 能力的 CPU 中存在。

**描述**

如果状态中断通过 DPV1 从站槽触发，CPU 操作系统调用 OB55。这可能是一个 DPV1 从站组件更改其操作模式的情况。例如从运行到停止。事件触发一个状态中断的精确信息，参见 DPV1 从站手册。

**状态中断组织块 OB 的局部变量**

下面表格包含了状态中断的临时变量（TEMP）。选择的变量名是 OB55 的缺省名称。

变量	类型	描述
OB55_EV_CLASS	字节	事件级别和标识：B#16#11（即将发生的事件）
OB55_STRT_INF	字节	B#16#55：（OB55 启动请求）
OB55_PRIORITY	字节	组态的优先级，默认值为 2
OB55_OB_NUMBR	字节	OB 号（55）
OB55_RESERVED_1	字节	备用
OB55_IO_FLAG	字节	输入模板：B#16#54 输出模板：B#16#55
OB55_MDL_ADDR	字	中断触发组件（模板）的逻辑地址
OB55_LEN	字节	中断提供的数据块长度
OB55_TYPE	字节	“状态中断”的中断 ID 号
OB55_SLOT	字节	中断触发组件（模板）的槽号
OB55_SPEC	字节	详细说明 <ul style="list-style-type: none"> <li>位 0 至 1：中断详细说明</li> <li>位 2：ADD_Ack</li> <li>位 3 至 7：Seq. No.e</li> </ul>

变量	类型	描述
OB55_DATE_TIME	DATE_AND_TIME	被调用 OB 的日期时间

**注意**

你可以在 OB55 中通过调用 SFB54 “RALRM”，从工业现场总线 DP 信息报文中得到更新信息。

## 1.8 刷新中断组织块（OB 56）

**注意**

一个刷新中断组织块 OB（OB56）只有在有 DPV1 能力的 CPU 中存在。

**描述**

如果更新中断通过 DPV1 从站槽触发，CPU 运行系统调用 OB56。这可能是如果你变换了 DPV1 从站槽的参数（通过本地或远程访问）。精确的刷新中断触发事件信息，参见 DPV1 从站手册。

**更新中断组织块 OB 的局部变量**

下面表格包含了更新中断的临时变量（TEMP）。选择的变量名是 OB56 的缺省名称。

变量	类型	描述
OB56_EV_CLASS	字节	事件级别和标识：B#16#11（即将发生的事件）
OB56_STRT_INF	字节	B#16#56（OB56 启动请求）
OB56_PRIORITY	字节	组态的优先级，默认值为 2
OB56_OB_NUMBR	字节	OB 号（56）
OB56_RESERVED_1	字节	备用
OB56_IO_FLAG	字节	输入模板：B#16#54 输出模板：B#16#55
OB56_MDL_ADDR	字	中断触发组件（模板）的逻辑地址
OB56_LEN	字节	中断提供的数据块长度
OB56_TYPE	字节	“刷新中断”的中断 ID 号
OB56_SLOT	字节	中断触发组件（模板）的槽号
OB56_SPEC	字节	详细说明 <ul style="list-style-type: none"> <li>• 位 0 至 1：中断详细说明</li> <li>• 位 2：ADD_Ack</li> <li>• 位 3 至 7：Seq. No.</li> </ul>
OB56_DATE_TIME	DATE_AND_TIME	被调用 OB 的日期时间



**注意**

你可以在 OB55 中通过调用 SFB54 “RALRM”，从工业现场总线 D 信息报文中得到更多信息。

## 1.9 制造商用中断组织块（OB57）

**注意**

制造商用中断组织块 OB（OB57）只有在有 DPV1 能力的 CPU 中存在。

**描述**

如果制造商用中断通过 DPV1 从站槽触发，CPU 运行系统调用 OB57。

**制造商用中断组织块 OB 的局部变量**

下表包含了制造商用中断的临时变量（TEMP）。选择的变量名是 OB57 的缺省名称。

变量	类型	描述
OB57_EV_CLASS	字节	事件级别和标识：B#16#11（即将发生的事件）
OB57_STRT_INF	字节	B#16#57（OB57 启动请求）
OB57_PRIORITY	字节	组态的优先级，默认值为 2
OB57_OB_NUMBR	字节	OB 号（57）
OB57_RESERVED_1	字节	备用
OB57_IO_FLAG	字节	输入模板：B#16#54 输出模板：B#16#55
OB57_MDL_ADDR	字	中断触发组件（模板）的逻辑地址
OB57_LEN	字节	中断提供的数据块长度
OB57_TYPE	字节	“属于制造厂商中断”的中断 ID 号
OB57_SLOT	字节	中断触发组件（模板）的槽号
OB57_SPEC	字节	详细说明 <ul style="list-style-type: none"> <li>• 位 0 至 1：中断详细说明</li> <li>• 位 2：ADD_Ack</li> <li>• 位 3 至 7：Seq. no.</li> </ul>
OB57_DATE_TIME	DATE_AND_TIME	被调用 OB 的日期时间

**注释**

你可以在 OB57 中通过调用 SFB54 “RALRM”，从工业现场总线 DP 信息报文中得到更多信息。

## 1.10 多处理器中断组织块（OB60）

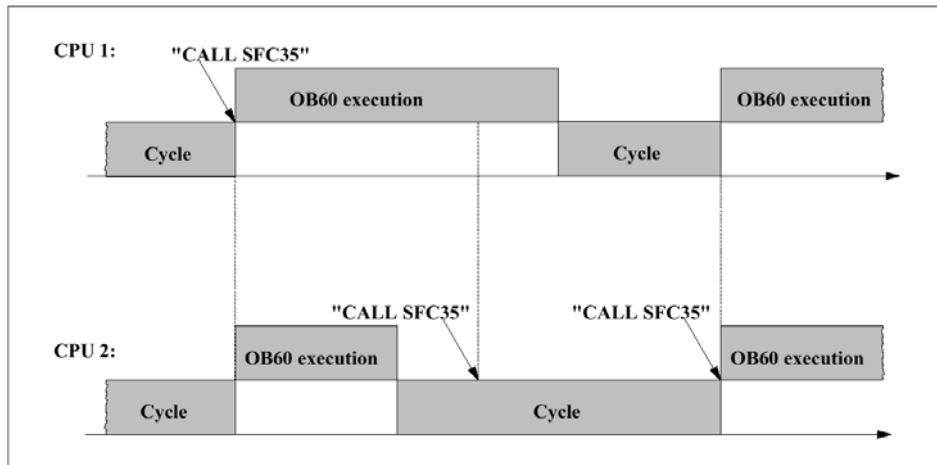
### 描述

使用多处理器中断，你可以确定在多处理器运行期间 CPU 的反应被同步到一个事件。和硬件中断由信号模板触发相反，多处理器中断仅由 CPU 输出。

### 理解多处理器中断 OB 的运行

多处理器中断通过调用 SFC35 “MP\_ALM” 触发。在多处理器期间，它使得总线段中的所有 CPU 的 OB60 同步启动，除非你封锁 OB60 (用 SFC39 “DIS\_IRT”) 或延迟它 (用 SFC41 “DIS\_AIRT”)。如果你没有把 OB60 下载到 CPU，CPU 返回中断之前的上一优先级并在哪里继续程序执行。在单处理器运行时和使用分段的机架，OB60 仅在你调用 SFC35 “MP\_ALM” 的 CPU 中启动。

当你的程序调用 SFC35 “MP\_ALM”，你要给出任务 ID。这一 ID 传送到所有 CPU。这使得你对特定事件作出反应。如果你在不同的 CPU 编写不同的 OB60，这可能导致 OB 的执行时间不同。在这种情况下，CPU 在不同的时间返回被中断的优先级。如果当一个 CPU 仍在忙于执行前一个多处理器中断的 OB60 时一个 CPU 又发出多处理器中断，那么不但发出请求的而且属于同一总线的其它所有 CPU 的 OB60 都不启动。两个 CPU 的举例如下图所示。你可以通过所调用的 SFC35 的功能值得到信息。



### 多处理器中断 OB 的局部数据

下表描述了多处理器中断 OB 的临时（TEMP）变量。变量名是 OB60 的缺省名称。

变量	类型	描述
OB60_EV_CLASS	字节	事件代码和 ID: B#16#11: 中断被激活
OB60_STRT_INF	字节	<ul style="list-style-type: none"> <li>B#16#61: 多处理器中断由本 CPU 触发</li> <li>B#16#62: 多处理器中断由其它 CPU 触发</li> </ul>
OB60_PRIORITY	字节	所选的优先级: 缺省 25
OB60_OB_NUMBR	字节	OB 号: 60
OB60_RESERVED_1	字节	备用
OB60_RESERVED_2	字节	备用
OB60_JOB	整数	任务 ID: SFC35 “MP_ALM” 的输入变量 JOB
OB60_RESERVED_3	整数	备用
OB60_RESERVED_4	整数	备用
OB60_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

## 1.11 周期同步中断 OB (OB 61 至 OB 64)

### 描述

周期同步中断给你这种与 DP 周期以同步周期启动程序的选择。OB61 作为接口 OB 提供给周期同步中断 TSAL1。你可以在 0 (不选择 OB) 和从 2 至 26 之间为 OB 61 设定优先级。

### 周期同步中断 OB 的局部数据

下表描述了周期同步中断 OB 的临时 (TEMP) 变量。变量名为 OB 61 的缺省名称。

变量	类型	描述
OB61_EV_CLASS	字节	事件代码和 ID: B#16#11: 中断被激活
OB61_STRT_INF	字节	B#16#64: OB 61 的启动请求 : B#16#67: OB 64 的启动请求
OB61_PRIORITY	字节	选择的优先级; 缺省: 25
OB61_OB_NUMBR	字节	OB 号: 61 ... 64
OB61_RESERVED_1	字节	备用
OB61_RESERVED_2	字节	备用
OB61_GC_VIOL	BOOL	GC 违反
OB61_FIRST	BOOL	启动或停止状态后首先使用
OB61_MISSED_EXEC	字节	从 OB 61 最后一次执行以来 OB61 启动失败的次数
OB61_DP_ID	字节	同步 DP 主站系统的 DP 主站系统 ID

变量	类型	描述
OB61_RESERVED_3	字节	备用
OB61_RESERVED_4	字	备用
OB61_RESERVED_5	字	备用
OB61_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

## 1.12 I/O 冗余故障 OB (OB70)

### 注意

I/O 冗余故障 OB (OB70) 仅可用在 H CPU。

### 描述

当 PROFIBUS DP 上的冗余丢失（例如，主动的 DP 主站总线故障或 DP 从站的接口模板出现故障时）或当主动的 DP 主站的 DP 从站连接的 I/O 改变时，H CPU 的操作系统调用 OB70。

如果启动事件出现且 OB70 未编程时 CPU 不变为 STOP 方式。如果 OB70 已下装且如果该 H 系统处于冗余方式，两个 CPU 都执行 OB70。H 系统保持冗余方式。

### I/O 冗余 OB 的局部数据

下表含有 I/O 冗余故障 OB 的临时 (TEMP) 变量。变量名为 OB70 的缺省名称。

变量	类型	描述
OB70_EV_CLASS	字节	事件级别和 ID: <ul style="list-style-type: none"> <li>• B#16#72: 离去事件</li> <li>• B#16#73: 到来事件</li> </ul>
OB70_FLT_ID	字节	故障代码 (可能值: B#16#A2, B#16#A3)
OB70_PRIORITY	字节	优先级: 可通过 STEP7 选择 (硬件组态)
OB70_OB_NUMBR	字节	OB 号 (70)
OB70_RESERVED_1	字	备用
OB70_INFO_1	字	根据故障代码
OB70_INFO_2	字	根据故障代码
OB70_INFO_3	字	根据故障代码
OB70_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

下表显示 OB70 启动时的事件。

OB70_FLT_ID	OB70 的启动事件
B#16#A2	DP 主站或 DP 主站系统故障
B#16#A3	冗余丢失/DP 从站的冗余恢复

根据故障代码这些变量有以下意义：

故障代码	位	意义
B#16#A2 OB70_INFO_1: OB70_INFO_2: OB70_INFO_3:	0 至 7: 8 至 15:	出问题的 DP 主站的逻辑基地址 备用 备用 出问题的 DP 主站的 DP 主站系统 ID
B#16#A3 OB70_INFO_1: OB70_INFO_2:  OB70_INFO_3	0 至 14:  15:  0 至 7: 8 至 15:	DP 主站的逻辑基地址 受影响的 DP 从站: 逻辑基地址, 如果使用 S7 从站或, 诊断地址, 如果使用 DP 标准从站。 I/O 标识 受影响的 DP 从站 DP 站的数量 DP 主站系统 ID

#### 注意

如果你使用具有 DPV1 能力的 CPU, 你可以借助于 SFB54 “RALRM” 获得超出 OB 启动信息的进一步中断信息。当你使用 DP 主站以 S7 兼容方式时也是一样。

## 1.13 CPU 冗余故障 OB（OB72）

### 注意

CPU 冗余故障 OB（OB72）仅用于 H CPU。

### 描述

当下列事件出现时 H CPU 的操作系统调用 OB72:

- CPU 冗余丢失
- 预定的主站切换
- 同步故障
- SYNC 模板故障
- 刷新终止
- 比较故障（例如，RAM，PIQ）

所有处于 RUN 或 STARTUP 方式的 CPU 随着相应的启动事件执行 OB72。

### CPU 冗余故障 OB 的局部数据

下表含有 CPU 冗余故障 OB 的临时（TEMP）变量。OB72 的缺省名用做变量名。

变量	类型	描述
OB72_EV_CLASS	字节	事件级别和 ID: <ul style="list-style-type: none"> <li>• B#16#78: 离去事件</li> <li>• B#16#73, B#16#75, B#16#79: 到来事件</li> </ul>
OB72_FLT_ID	字节	故障代码（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	字节	优先级: 可通过 STEP 7 选择（硬件组态）
OB72_OB_NUMBR	字节	OB 号（72）
OB72_RESERVED_1	字	仅用于故障代码 B#16#03: <ul style="list-style-type: none"> <li>• 高字节: OB72_1INFO_2 和 OB72_1INFO_3 内容的 ID <ul style="list-style-type: none"> <li>- 0: OB72_INFO-2 和 OB72_INFO_3 没意义</li> <li>- B#16#C4: 伴随着备用主站切换（如果 OB72_INFO_3=W#16#0001）或不伴随备用主站切换（如果 OB72_INFO_3=W#16#0002）执行了排故方式之后转换到冗余方式。OB72_INFO_2 预留。</li> <li>- B#16#CD: OB72_INFO_2 和 OB72_INFO_3 含有优先级 &gt;15 的实际封锁时间</li> </ul> </li> </ul>

变量	类型	描述
		<ul style="list-style-type: none"> <li>低字节：预留</li> </ul>
OB72_INFO_1	字	仅用于故障代码 B#16#C2： <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: 扫描周期时间增加</li> <li>2: I/O 无效时间</li> <li>3: 通讯时间延迟</li> </ul> </li> <li>低字节：当前的刷新尝试</li> </ul>
OB72_INFO_2	字	仅用于故障代码 B#16#03 且 OB72_RESERVED_1=B#16#CD：以 ms 优先级>15 的实际封锁时间的高字节
OB72_INFO_3	字	仅用于故障代码 B#16#03： <ul style="list-style-type: none"> <li>OB72_RESERVED_1=B#16#C4：               <ul style="list-style-type: none"> <li>W#16#0001：伴随着备用主站切换执行了排故方式之后转换到冗余方式</li> <li>W#16#0002：不伴随备用主站切换的执行了排故方式之后转换到冗余方式</li> </ul> </li> <li>OB72_RESERVED_1=B#16#C4：优先级 &gt; 15 的以 ms 实际封锁时间的高字节</li> </ul>
OB82_DATE_TIME	DATE_AND_TIME	OB被调用时的日期和时间

下表显示导致 OB72 启动的事件。

OB72_FLT_ID	OB72 的启动事件
B#16#01	由于 CPU 故障冗余丢失（二中之一）
B#16#02	由于用户触发 STOP 而冗余丢失（二中之一）
B#16#03	H 系统（二中之一）变为冗余方式
B#16#20	RAM 比较故障
B#16#21	比较过程映像输出值故障
B#16#22	比较标志、定时器或计数器故障
B#16#23	识别出不同的操作系统数据
B#16#31	由于主站故障备用主站切换
B#16#33	在运行时由于系统改变备用主站切换
B#16#34	由于同步模切连接问题备用主站切换
B#16#35	由 90 “H_CTRL” 触发备用主站切换
B#16#40	由于超出等待时间用户程序中的同步故障
B#16#41	由于在不同的同步点等待用户程序中的同步故障
B#16#42	由于在不同的同步点等待操作系统中的同步故障
B#16#43	由于超出等待时间操作系统中的同步故障
B#16#44	由于错误的操作数据操作系统中的同步故障
B#16#50	无 SYNC 模块

OB72_FLT_ID	OB72 的启动事件
B#16#51	未通电 SYNC 模块改变
B#16#52	SYNC 模块移出/插入
B#16#53	未复位 SYNC 模块改变
B#16#54	SYNC 模块：机架号选择两次
B#16#55	SYNC 模块故障/排除
B#16#56	SYNC 模块上非法机架号设置
B#16#C1	刷新中止
B#16#C2	在几次尝试期间 ( $1 \leq n \leq$ 由于监视时间超出中止后刷新尝试的最大可能数) 因为监视时间超出刷新尝试中止

## 1.14 通讯冗余故障 OB (OB73)

### 注意

通讯冗余故障 OB (OB73) 仅对 CPU417-4H 硬件版本 V2.0 有效。

### 描述

当首次冗余丢失出现在容错 S7 连接时（容错 S7 连接仅用于 S7 通讯。更多信息，参阅“S7-400H 可编程控制器，容错系统”），H CPU 操作系统调用 OB73。如果冗余丢失出现在以后的容错 S7 连接时，不再有 OB73 启动。

直到所有容错 S7 连接的冗余恢复之前，OB73 都不出现再次启动。

如果启动事件发生且 OB73 未编程，CPU 不变为 STOP 方式。

### CPU 冗余故障 OB 的局部数据

下表含有通讯冗余故障 OB 的临时 (TEMP) 变量。OB73 的缺省名用做变量名。

变量	类型	描述
OB73_EV_CLASS	字节	事件代码和 ID: B#16#73, B#16#72
OB73_FLT_ID	字节	故障代码 (可能值: B#16#E0)
OB73_PRIORITY	字节	选择的优先级: 缺省 25
OB73_OB_NUMBR	字节	OB 号 (73)
OB73_RESERVED_1	字	保留
OB73_INFO_1	字	(对用户无效)
OB73_INFO_2	字	(对用户无效)
OB73_INFO_3	字	(对用户无效)
OB73_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间



下表显示导致 OB73 启动的事件。

OB73_FLT_ID	OB73 的启动事件
B#16#E0	通讯中冗余丢失/故障排除

## 1.15 时间故障组织块（OB80）

### 描述

当执行一个 OB 时出现故障时 S7-300 CPU 的操作系统调用 OB80。这样的故障包括：循环时间超出、执行 OB 时应答故障、向前移动时间以致于跃过了 OB 的启动时间、CiR 后恢复 RUN 方式。如果，例如，当循环中断 OB 仍在执行前一次调用时，该 OB 块的启动事件发生，操作系统调用 OB80。

如果 OB80 未被编程，CPU 变为 STOP 方式。

你可以使用 SFC39 至 42 封锁或延时和再使能时间故障 OB。

### 注意

如果在同一个扫描周期中由于扫描时间超出 OB80 被调用两次，CPU 就变为 STOP 方式。你可以通过在程序中适当的位置调用 SFC43 “RE\_TRIGR” 来避免这种情况。

### 时间故障 OB 的局部数据

下表描述了时间故障 OB 的临时（TEMP）变量。变量名是 OB80 的缺省名。

变量	类型	描述
OB80_EV_CLASS	字节	事件级别和标识：B#16#35
OB80_FLT_ID	字节	故障代码：（允许值：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	字节	优先级：在 RUN 方式时 OB80 以优先级 26 运行，在 OB 请求缓冲区的溢出事件时以优先级 28 运行
OB80_OB_NUMBR	字节	OB 号（80）
OB80_RESERVED_1	字节	保留
OB80_RESERVED_2	字节	保留
OB80_ERROR_INFO	字	故障信息：根据故障代码
OB80_ERR_EV_CLAS	字节	引起故障的启动事件的事件级别
OB80_ERR_EV_NUM	字节	引起故障的启动事件的事件号
OB80_OB_PRIORITY	字节	故障信息：根据故障代码
OB80_OB_NUM	字节	故障信息：根据故障代码
OB80_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

根据故障代码这些变量有以下意义：

故障代码	位	意义
B#16#01 OB80_ERROR_INFO: OB80_ERR_EV_CLASS: OB80_ERR_EV_NUM: OB80_OB_PRIORITY: OB80_OB_NUM		循环时间超出。 最后一个扫描周期的运行时间 (ms)。 触发中断的事件的级别。 触发中断的事件的号码。 当故障出现时正在执行的 OB 的优先级。 当故障出现时正在执行的 OB 的号码。
B#16#02 OB80_ERROR_INFO:  OB80_ERR_EV_CLASS: OB80_ERR_EV_NUM: OB80_OB_PRIORITY  OB80_OB_NUM:		调用的 OB 仍在执行。 由 OB80_ERR_EV_CLASS 和 OB80_ERR_EV_NUM 确定的调用块的有关临时变量。 触发中断的事件的级别。 触发中断的事件的号码。 引起故障的 OB 的优先级 (例如：“7”指 OB30/ 优先级 7 它应该启动，但不能启动)。 引起故障的 OB 的号码 (例如：“30”指 OB30 它应该启动，但不能启动。)
B#16#05 和 B#16#06 OB80_ERROR_INFO:  OB80_ERR_EV_CLASS: OB80_ERR_EV_NUM: OB80_OB_PRIORITY: OB80_OB_NUM:	位 0 置位: 位 7 置位: 位 8 至 15:	由于向前移动时间日期跳过时间中断。 HOLD 之后返回 RUN 日期跳过时间中断。 日期时间中断 0 的启动时间在过去。 日期时间中断 7 的启动时间在过去。 未用 未用 未用 未用 未用
B#16#07 参数的意义参见故障代码 B#16#02。		当前优先级的 OB 请求缓冲区溢出。 一个优先级的每个 OB 启动请求将输入到相应的 OB 请求缓冲区，OB 完成之后该输入被删除。如果 一个优先级的 OB 启动请求多于相应的 OB 请求 缓冲区中最大允许输入数，OB80 将被调用且故障 代码 B#16#07)。
B#16#08 参数的意义参见故障代码 B#16#02。		同步循环中断时间故障
B#16#09 参数的信息参见故障代码 B#16#02		由于高中断装载中断丢失
B#16#0A OB80_ERROR_INFO:		CiR 之后恢复 RUN CiR 同步时间以 ms 为单位

## 1.16 电源故障组织块（OB81）

### 描述

当有与由电源（仅对 S7-400）或后备电池（当到来和离去事件）有关的故障触发的事件发生时，S7-300 CPU 的操作系统调用 OB81。

在 S7-400 中，如果电池测试功能已通过 BATT.INDIC 开关激活，仅在电池故障的事件中调用 OB81。

如果 OB81 未编程，CPU 保持为 STOP 方式。

你可以用 SFC39 至 42 来禁用、延时或再使能电源故障 OB。

### 电源故障 OB 的局部数据

下表描述了电源故障 OB 的临时（TEMP）变量。变量名是 OB81 的缺省名。

变量	类型	描述
OB81_EV_CLASS	字节	事件级别和标识： B#16#38：离去事件 B#16#39：到来事件
OB81_FLT_ID	字节	故障代码（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	字节	<ul style="list-style-type: none"> <li>优先级；可通过 STEP 7 选择（硬件组态）</li> <li>例如，RUN 方式的可能值：2-26</li> </ul>
OB81_OB_NUMBR	字节	OB 号（81）
OB81_RESERVED_1	字节	保留
OB81_RESERVED_2	字节	保留
OB81_MDL_ADDR	整数	<ul style="list-style-type: none"> <li>位 0 至 2： 机架号</li> <li>位 3： 0=备用 CPU, 1=主站 CPU</li> <li>位 4 至 7： 1111</li> </ul>
OB81_RESERVED_3	字节	仅与故障代码 B#16#31, B#16#32 and B#16#33 有关
OB81_RESERVED_4	字节	
OB81_RESERVED_5	字节	
OB81_RESERVED_6	字节	
OB81_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

变量 OB81\_RESERVED\_*i*, 3 ≤ *i* ≤ 6 显示扩展机架上后备电池（故障代码 B#16#31）、后备电压（故障代码 B#16#32）或 24V 电源（故障代码 B#16#33）故障或返回。

下表显示变量 OB81\_RESERVED\_*i*, 3 ≤ *i* ≤ 6 哪位对应哪个扩展机架。

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

变量 OB81\_RESERVED\_I 中的位有下述意义（对相关的扩展机架）：

当事件发生时，扩展机架被标出（相应位置位）电池或后备电压或 24 V 电源故障。更早期的扩展机架电池或后备电压或 24 V 电源故障不再显示。

当事件排除且至少一个扩展机架后备恢复，它也被表示（相应位置位）。

变量 OB81\_FLT\_ID 有下列意义：

OB81_FLT_ID	意义
B#16#21:	中央机架的至少一个后备电池耗尽/问题排除（BATTF） 如果仅两个电源中的一个故障（如果是冗余后备电池）事件也发生。如果第二个电池也发生故障，事件不再出现。
B#16#22:	中央机架的后备电压故障/问题排除（BAF）
B#16#23:	中央机架的 24 V 电源故障/问题排除。
B#16#25:	至少一个冗余中央机架中至少一个后备电池耗尽/问题排除（BATTF）
B#16#26:	至少一个冗余中央机架中后备电压故障（BAF）
B#16#27:	至少一个冗余中央机架 24 V 供给故障
B#16#31:	至少一个扩展机架的至少一个后备电池耗尽/问题排除（BATTF）
B#16#32:	至少一个扩展机架的后备电压故障（BAF）
B#16#33:	至少一个扩展机架的 24 V 电源故障/问题排除。

## 1.17 诊断中断组织块（OB82）

### 描述

如果模块具有诊断能力又使能了诊断中断，当它检测到故障时，它输出一个诊断中断请求给 CPU（当到来和离去事件）。于是操作系统调用 OB82。

OB82 的局部变量含有故障模板四个字节诊断数据（见下表）。

如果 OB82 未被编程，CPU 变为 STOP 方式。

你可以用 SFC39 至 42 来禁用或延时并再使能诊断中断 OB。

### 诊断中断 OB 的局部数据

下表描述了诊断中断 OB 的临时（TEMP）变量。变量名是 OB82 的缺省名。

变量	类型	描述
OB82_EV_CLASS	字节	事件级别和标识： <ul style="list-style-type: none"> <li>• B#16#38：离去事件</li> <li>• B#16#39：到来事件</li> </ul>
OB82_FLT_ID	字节	故障代码（B#16#42）
OB82_PRIORITY	字节	• 优先级；可通过 STEP 7 选择（硬件组态）
OB82_OB_NUMBR	字节	OB 号（82）
OB82_RESERVED_1	字节	备用
OB82_IO_FLAG	字节	<ul style="list-style-type: none"> <li>• 输入模板：B#16#54</li> <li>• 输出模板：B#16#55</li> </ul>
OB82_MDL_ADDR	字	故障发生处模板的逻辑基地址
OB82_MDL_DEFECT	BOOL	模板故障
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	前连接器未插入
OB82_NO_CONFIG	BOOL	模板未组态
OB82_CONFIG_ERR	BOOL	模板参数不正确
OB82_MDL_TYPE	字节	<ul style="list-style-type: none"> <li>• 位 0 至 3：模板级别</li> <li>• 位 4：通道信息存在</li> <li>• 位 5：用户信息存在</li> <li>• 位 6：来自替代的诊断中断</li> <li>• 位 7：备用</li> </ul>
OB82_SUB_MDL_ERR	BOOL	子模板丢失或有故障
OB82_COMM_FAULT	BOOL	通讯问题
OB82_MDL_STOP	BOOL	操作方式（0：RUN，1：STOP）

变量	类型	描述
OB82_WTCH_DOG_FLT	BOOL	看门狗定时器响应
OB82_INT_PS_FLT	BOOL	内部电源故障
OB82_PRIM_BATT_FLT	BOOL	整个后备故障
OB82_BCKUP_BATT_FLT	BOOL	完全后背故障
OB82_RESERVED_2	BOOL	备用
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 故障
OB82_FUSE_FLT	BOOL	熔断器跳闸
OB82_HW_INTR_FLT	BOOL	硬件中断丢失
OB82_RESERVED_3	BOOL	备用
OB82_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

#### 注意

如果你有用 DPV1 能力的 CPU 借助 SFB54 你可以得到超出 OB 启动信息的中断的进一步信息。当你运行 DP 主站以 S7 兼容方式时也是一样。

## 1.18 模板插/拔中断组织块（OB83）

### 描述

在下列情况下 CPU 操作系统调用 OB83:

- 组态的模板插入/拔出之后
- 在 STEP 7 下修改了模板参数并在 RUN 状态下装所作修改到 CPU 之后  
你可以借助 SFC39 至 42 封锁/延时/使能插入/拔出中断 OB。

### 模板插入和拔出

在 RUN、STOP 和 STARTUP 方式时每次组态的模板被插入或拔出，就产生了插入/拔出中断（电源模板、CPU、适配模块和 IM 不能在这种方式下移出）。该中断引起有关 CPU 的诊断缓冲区和系统状态表的记录。如果 CPU 在 RUN 方式也启动插入/拔出 OB。如果该 OB 未编程，CPU 变为 STOP 方式。

系统以秒间隔查询-400 模板以检测插入或拔出。为使 CPU 能检测出 S7-400 模板的拔出和插入，拔出和插入之间最小时间间隔必须超出两秒。对其它模板最小时间更高一些。

如果你在 RUN 方式下拔出组态的模板，OB83 被启动。由于仅以一秒的间隔监视模板的存在，如果模板被直接访问或当过程映像被刷新时可能首先检测出访问故障。

如果你在 RUN 方式下在组态的插槽中插入一块模板，操作系统检查插入的模板类型是否与组态记录一致。如果模板类型匹配，于是 OB83 被启动并且参数被赋值。

### 重新组态模板

当你在运行时，修改你的系统组态（CiR）你可以重新选择已存在模板的参数。参数的重新选择通过传送所需的参数数据到模板来进行。

过程如下：

1. 在你用 STEP 7 为一块模板选择了新的参数并且在 RUN 方式下下装该参数组态到 CPU 后 OB83 启动（启动事件 W#16#3367）。相应的 OB 启动信息为逻辑基地址（OB83\_MDL\_ADDR）和模块类型（OB83\_MDL\_TYPE）。模板 I/O 数据现在可能不对，这意味着不可能有 SFC 忙于发送数据记录到该模板。
2. OB 83 执行之后模板参数被重新赋值。
3. 参数赋值之后 OB 83 将再次启动（启动事件 W#16#3267，成功提供参数赋值，或 W#16#3968 如果故障）。模板 I/O 数据响应与插入中断后的响应相同，也就是，现在它们可能不正确。现在你可以再次调用 SFC 发送数据记录到该模板。

### OB 83 的局部数据

下表描述插入/拔出模板中断 OB 的临时（TEMP）变量。变量名是 OB83 的缺省名。

变量	类型	描述
OB83_EV_CLASS	字节	事件级别和标识： <ul style="list-style-type: none"> <li>• B#16#32：模板参数赋值结束</li> <li>• B#16#33：模板参数赋值启动</li> <li>• B#16#38：模板插入</li> <li>• B#16#39：模板拔出或无反应，或参数赋值结束</li> </ul>
OB83_FLT_ID	字节	故障代码：（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	字节	优先级，可通过 STEP 7 选择（硬件组态）
OB83_OB_NUMBR	字节	OB 号（83）
OB83_RESERVED_1	字节	块模板或接口模板标识
OB83_MDL_TD	字节	范围： <ul style="list-style-type: none"> <li>• B#16#54：外设输入（PI）</li> <li>• B#16#55：外设输出（PQ）</li> </ul>
OB83_MDL_ADDR	字	有关模板的逻辑基地址
OB83_RACK_NUM	字	<ul style="list-style-type: none"> <li>• 如果 OB83_RESERVED_1 = B#16#A0：接口模板号</li> <li>• 如果 OB83_RESERVED_1 = B#16#C4：机架号或 DP 站号（低字节）或 DP 主站系统 ID（高字节）</li> </ul>

变量	类型	描述
OB83_MDL_TYPE	字	有关模板的模板类型： <ul style="list-style-type: none"> <li>• W#16#X5XX: 模拟量模板</li> <li>• W#16#X8XX: 功能模板</li> <li>• W#16#XCXX: CP</li> <li>• W#16#XFX: 数字量模板</li> </ul> X : 数值对用户无效
OB83_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

由故障代码 OB83\_FLT\_ID 决定的变量 OB83\_MDL\_TYPE 有下列意义:

故障代码	意义
B#16#61:  OB83_MDL_TYPE:	模板插入。模板类型正确 (对事件级别 B#16#38) 模板拔出或无响应 (对事件级别 B#16#39) 实际模板类型
B#16#63 OB83_MDL_TYPE:	模板插入但模板类型不正确 实际模板类型
B#16#64:  OB83_MDL_TYPE:	模板插入但有问题 (读不出模板 ID) 组态模板类型
B#16#65: OB83_MDL_TYPE:	模板插入但模板参数赋值故障 实际模板类型
W#16#3866	模板再次响应, 装载电压故障改正
W#16#3966	模板不再次响应, 装载电压故障改正
W#16#3367	模板重新组态启动
W#16#3267	模板重新组态结束
W#16#3968	模板重新组态结束有故障
W#16#3884	接口模板插入
W#16#3984	接口模板拔出

**注意**

如果你使用具有 DPV1 能力的 CPU, 借助 SFB54 “RALRM”, 你可以得到 OB 启动信息之外的中断的进一步信息。当你运行 DP 主站以 S7 兼容方式时也是一样。



## 1.19 CPU 硬件故障组织块（OB84）

### 描述

当前 CPU 的 OS 不再调用 OB84。

### 硬件故障 OB 的局部数据

下表含有 CPU 硬件故障的临时（TEMP）变量。变量名是 OB84 的缺省名。

变量	类型	描述
OB84_EV_CLASS	字节	事件级别和标识： <ul style="list-style-type: none"> <li>• B#16#38：离去事件</li> <li>• B#16#39：到来事件</li> </ul>
OB84_FLT_ID	字节	故障代码（B#16#81）
OB84_PRIORITY	字节	优先级，可通过 STEP 7 选择（硬件组态）
OB84_OB_NUMBR	字节	OB 号（84）
OB84_RESERVED_1	字节	备用
OB84_RESERVED_2	字节	备用
OB84_RESERVED_3	字	备用
OB84_RESERVED_4	双字	备用
OB84_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

## 1.20 优先级故障组织块（OB85）

### 描述

当下列事件之一发生时，CPU 的操作系统调用 OB85：

- 未下装的 OB 的启动事件（OB81 除外）。
- 当操作系统访问模板时故障。
- 在由系统刷新过程映像期间 I/O 访问故障（如果 OB85 调用没有在组态中封锁）。

### 注意

如果 OB85 未编程，当检测到这些事件之一时 CPU 变为 STOP 方式。

你可以使用 SFC39 至 42 封锁或延时并使能优先级故障 OB。

### 优先级故障 OB 的局部数据

下表描述优先级故障 OB 的临时（TEMP）变量。变量名是 OB85 的缺省名。

变量	类型	描述
OB88_EV_CLASS	字节	事件级别和标识: B#16#35 B#16#38 (仅有故障代码 B#16#B3 和 B#16#B4) B#16#39 (仅有故障代码 B#16#B1 和 B#16#B2, B#16#B3 和 B#16#B4)
OB85_FLT_ID	字节	故障代码 (可能值: B#16#A1, B#16#A2, B#16#A3, B#16#B1, B#16#B2, B#16#B3, B#16#B4)
OB85_PRIORITY	字节	优先级, 可通过 STEP 7 选择 (硬件组态)
OB85_OB_NUMBR	字节	OB 号 (85)
OB85_RESERVED_1	字节	备用
OB85_RESERVED_2	字节	备用
OB85_RESERVED_3	整数	备用
OB85_ERR_EV_CLASS	字节	引起故障的事件级别
OB85_ERR_EV_NUM	字节	引起故障的事件号码
OB85_OB_PRIOR	字节	当故障发生时被激活的 OB 的优先级
OB85_OB_NUM	字节	当故障发生时被激活的 OB 的号码
OB85_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

如果你要根据可能的故障代码编写 OB85, 我们提示你按下述组织局部数据:

变量	类型
OB85_EV_CLASS	字节
OB85_FLT_ID	字节
OB85_PRIORITY	字节
OB85_OB_NUMBR	字节
OB85_DKZ23	字节
OB85_RESERVED_2	字节
OB85_Z1	字
OB85_Z23	双字
OB85_DATE_TIME	DATE_AND_TIME

根据故障代码，修改过的和缺省的变量比较有下述意义：

故障代码	字节/字	意义
B#16#A1 和  B#16#A2  OB85_Z1: OB85_Z23:	高字: 低字: 高字节/ 低字节	用 STEP 7 生成的组态结果，你的程序或操作系统产生的 OB 未下装到 CPU 的启动事件。 用 STEP 7 生成的组态结果，你的程序或操作系统产生的 OB 未下装到 CPU 的启动事件。 调用的 OB 的相关局部变量由 OB85_Z23 决定。 引起 OB 调用的事件的级别和号码  故障时的程序级和激活的 OB
B#16#A3 OB85_Z1:       OB85_Z23	高字节:  低字节:     高字: 低字:	当操作系统访问模块时故障 操作系统的故障 ID 1: 集成功能 2: IEC 定时器 0: 无故障分辨能力 1: 块未装载 2: 区域长度故障 3: 写保护故障 块号 引起故障的 MC7 命令的相对地址。块类型必须从 OB85_DKZ23 得到 (B#16#88: OB, B#16#8C: FC, B#16#8E: FB, B#16#8A: DB)
B#16#B1 和 B#16#B2:		当刷新输入的过程映像时 I/O 访问故障 当传送输出的过程映像到输出模板时 I/O 访问故障
OB85_DKZ23:		I/O 访问故障发生时过程映像传送类型的 ID <ul style="list-style-type: none"> <li>• B#16#10: 字节访问</li> <li>• B#16#20: 字节访问</li> <li>• B#16#30: 双字访问</li> <li>• B#16#57: 传送组态的一致性区域</li> </ul>
OB85_Z1:		由 CPU 内部使用占用: 模板的逻辑基地址 如果 OB85_RESERVED_2 有数值 B#16#76, OB85_Z1 接收有关 SFC (SFC 14, 15, 26 和 27) 的返回值。
OB85_Z23:	字节 0:	过程映像部分号。
	字节 1:	<ul style="list-style-type: none"> <li>• 无效, 如果 OB85_DKZ23=B#16#10, 20 或 30</li> <li>• 一致性区域长度, 单位字节, 如果 OB85_DKZ23= B#16#57</li> </ul>
	字节 2 和 3:	<ul style="list-style-type: none"> <li>• PII 引起的 I/O 地址, 如果 OB85_DKZ23= B#16#10、20 或 30</li> <li>• 一致性区域的逻辑起始地址, 如果 OB85_DKZ23= B#16#57</li> </ul>

故障代码	字节/字	意义
如果你已经组态了系统过程映像表刷新的 I/O 访问故障的重复的 OB85 调用你会获得故障代码 B#16#B1 和 B#16#B2。		
B#16#B3: B#16#B4:		当刷新输入的过程映像时 I/O 访问故障，到来/离去事件 当传送输出过程映像至输出模板时 I/O 访问故障，到来/离去事件
OB85_DKZ23:		I/O 访问故障发生时过程映像传送类型的 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: 传送组态的一致性区域</li> </ul>
OB85_Z1:		由 CPU 内部使用占用：模板的逻辑基地址 如果 OB85_RESERVED_2 有数值 B#16#76，OB85_Z1 接收有关 SFC (SFC 14, 15, 26 和 27) 的返回值。
OB85_Z23:	字节 0:	过程映像部份号。
	字节 1:	<ul style="list-style-type: none"> <li>• 无效，如果 OB85_DKZ23=B#16#10, 20 或 30</li> <li>• 一致性区域的长度，单位字节，如果 OB85_DKZ23= B#16#57</li> </ul>
	字节 2 和 3:	<ul style="list-style-type: none"> <li>• PII 引起的 I/O 地址，如果 OB85_DKZ23= B#16#10, 20 或 30</li> <li>• 一致性区域的逻辑起始地址，如果 OB85_DKZ23= B#16#57</li> </ul>
如果你已经组态了用于系统刷新过程映像表的 I/O 访问故障和离去事件的 OB85 调用，你会获得故障代码 B#16#B3 和 B#16#B4。在冷或暖启动之后，在下一过程映像表刷新时，所有不存在的输入和输出的访问将作为 I/O 访问故障报告。		

## 1.21 机架故障组织块（OB86）

### 描述

当扩展机架（不是 CPU 318）、DP 主站系统、或分布式 I/O 中从站故障时（当到来和离去事件时），CPU 的操作系统调用 OB86。

如果 OB86 未编程，当检测到这类故障时，CPU 变为 STOP 方式。

你可以使用 SFC 39 至 42 封锁或延时并使能 OB86。

### 机架故障 OB 的局部数据

下表描述机架故障 OB 的临时（TEMP）变量。变量名为 OB86 的缺省名。

变量	类型	描述
OB86_EV_CLASS	字节	事件级别和标识： <ul style="list-style-type: none"> <li>• B#16#38: 离去事件</li> <li>• B#16#39: 到来事件</li> </ul>
OB86_FLT_ID	字节	故障代码： （可能值 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	字节	优先级，可通过 STEP 7 选择（硬件组态）
OB86_OB_NUMBR	字节	OB 号（86）
OB86_RESERVED_1	字节	备用
OB86_RESERVED_2	字节	备用
OB 86_MDL_ADDR	字	根据故障代码
OB86_RACKS_FLTD	布尔矩阵 [0 ..31]	根据故障代码
OB86_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

如果你要根据可能的故障代码编写 OB86，我们建议你按下述组织局部数据：

变量	类型
OB86_EV_CLASS	字节
OB86_FLT_ID	字节
OB86_PRIORITY	字节
OB86_OB_NUMBR	字节
OB86_RESERVED_1	字节
OB86_RESERVED_2	字节
OB86_MDL_ADDR	字
OB86_Z23	双字
OB86_DATE_TIME	DATE_AND_TIME

根据故障代码，变量的内容有下列意义：

故障代码	意义
B#16#39C1: OB86_MDL_ADDR: OB86_Z23:	扩展机架故障 IM 的逻辑基地址 含有每个可能的扩展机架的一个位： 位 0： 总为 0 位 1： 第 1 扩展机架 ： 位 21： 第 21 扩展机架 位 22 至 29： 总为 0 位 30： 在 SIMATIC S5 区中至少一个扩展机架故障 位 31： 总为 0
B#16#38C1:	扩展机架重新运行

意义：当事件发生时，报告导致 OB86 调用的扩展机架故障（对应它们的位被置位）。更早已经故障的扩展机架不再显示。当故障修复时，再次激活的扩展机架在故障代码中报告（对应它们的位被置位。）

故障代码	意义
B#16#C2: OB86_MDL_ADDR: OB86_Z23:	扩展机架恢复并且预期的和实际的组态不一致。 IM 的逻辑基地址 含有对应每一可能扩展机架的一位，见故障代码 B#16#C1。 <ul style="list-style-type: none"> <li>• 一个位当被置位时的意义（有关的扩展机架）：                             <ul style="list-style-type: none"> <li>- 不正确类型 ID 的模板存在</li> <li>- 组态的模板丢失</li> <li>- 至少一块模板有故障</li> </ul> </li> </ul>
B#16#C3:  OB86_MDL_ADDR: OB86_Z23:	分布式 I/Os：主站系统故障。（仅到来事件引起 OB86 带着故障代码 B#16#C3 启动。离去状态启动 OB86 带有故障代码 B#16#C4 和事件级别 B#16#38。每个 DP 从站恢复启动 OB86）。 DP 主站的逻辑基地址。 DP 主站系统 ID 位 0 至 7： 备用 位 8 至 15： DP 主站系统 ID 位 16 至 31： 备用
B#16#C4: B#16#C5:  OB86_MDL_ADDR: OB86_Z23:	DP 站故障 DP 站有问题 DP 主站的逻辑基地址。  有关的 DP 从站地址： 位 0 至 7： DP 站号 位 8 至 15： DP 主站系统 ID 位 16 至 30： S7 从站的逻辑基地址或标准 DP 从站的诊断地址 位 31： I/O 标识

故障代码	意义
B#16#C6: OB86_MDL_ADDR: OB86_Z23:	扩展机架再次运行但模板参数赋值出错 IM 的逻辑基地址 含有对应每一可能扩展机架的一位： 位 0: 总为 0 位 1: 第 1 个扩展机架 : 位 21: 第 21 个扩展机架 位 22 至 30: 备用 位 31: 总为 0 当一位被置位时意义（有关扩展机架）： 不正确类型标识的模板存在 模板丢失或模板的参数不正确
B#16#C7: OB86_MDL_ADDR: OB86_Z23:	DP 站恢复，但模板参数赋值出错 DP 主站的逻辑基地址。 相应的 DP 从站的地址： 位 0 至 7: DP 站号 位 8 至 15: DP 主站系统 ID 位 16 至 30: DP 从站的逻辑基地址 位 31: I/O 标识
B#16#C8: OB86_MDL_ADDR: OB86_Z23:	DP 站恢复，然而组态与实际组态不一致 DP 主站的逻辑基地址 相应的 DP 从站的地址： 位 0 至 7: DP 站号 位 8 至 15: DP 主站系统 ID 位 16 至 30: DP 从站的逻辑基地址 位 31: I/O 标识

**注意**

如果你使用具有 DPV1 能力的 CPU，借助 SFB54 “RALRM”，你可以得到超出 OB 启动信息的中断的进一步信息。当你运行 DP 主站以 S7 兼容方式时也是一样。

## 1.22 通讯故障组织块（OB87）

### 描述

当导致通讯故障的事件发生时 CPU 的操作系统调用 OB87。

如果 OB87 未编程，CPU 不进入 STOP 方式。

你可以用 SFC39 至 42 封锁或延时并再使能通讯故障 OB。

### OB87 的局部数据

下表描述通讯故障 OB 的临时（TEMP）变量。变量名为 OB87 的缺省名。

变量	类型	描述
OB87_EV_CLASS	字节	事件级别和标识 B#16#35:
OB87_FLT_ID	字节	故障代码: (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	字节	优先级, 可通过 STEP 7 选择 (硬件组态)
OB87_OB_NUMBR	字节	OB 号 (87)
OB87_RESERVED_1	字节	备用
OB87_RESERVED_2	字节	备用
OB87_RESERVED_3	字	根据故障代码
OB87_RESERVED_4	双字	根据故障代码
OB87_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

根据故障代码变量有下述意义:

故障代码	字节/字	意义
B#16#D2: B#16#D3: B#16#D4: B#16#D5: OB87_RESERVED_3: OB87_RESERVED_4:		当前的诊断记录传送不可能 同步信息不能传送 (主站) 由于时钟同步非法的日时间跳转。 当接收同步时间时故障 (从站)。 不含进一步信息。 不含进一步信息。
B#16#E1: B#16#E3: B#16#E4: OB87_RESERVED_3: OB87_RESERVED_4:	高字节: 低字节:	在全局数据通讯期间不正确的帧 ID 在全局数据通讯期间帧长度故障 收到非法 GD 包号码。 接口 ID (0: K 总线, 1:MPI) GD 包号 不含进一步信息。



故障代码	字节/字	意义
B#16#E2: OB87_RESERVED_3: OB87_RESERVED_4:	高字: 低字:	GD 包状态不能记录在 DB 中 DB 号 不含进一步信息。 GD 包号 (高字节) GD 包号 (低字节)
B#16#E5: OB87_RESERVED_3: OB87_RESERVED_4:	高字: 低字:	在通过通讯功能块数据交换期间对 DB 访问故障 预留给 CPU 内部使用。 含有引起故障的 MC7 命令的块的号码。 含有引起故障的 MC7 命令的相对地址。

从 OB\_87\_RESERVED\_1 可读到块类型 (B#16#88: OB, B#16#8A: DB, B#16#8C: FC, B#16#8E: FB)

故障代码	意义
B#16#E6: OB87_RESERVED_3: OB87_RESERVED_4:	GD 包状态不能记录在 DB 中 DB 号 不含进一步信息。

## 1.23 过程中断 OB (OB 88)

### 描述

在一个程序块的执行被中断后 CPU 的操作系统调用 OB88。这种中断可能的原因是：

- 同步故障的嵌套深度太高
- 块调用的嵌套深度 (B-堆栈) 太高
- 在局部数据定位中故障

如果你未编程 OB88 并且程序块执行中断, CPU 进入 STOP 方式 (事件 ID W#16#4570)。

如果程序块在优先级 28 中执行被中断, CPU 进入 STOP 方式。

你可以借助 SFC39 至 42 封锁、延时并使能过程中断 OB。

### 过程中断 OB 的局部数据

下表含有过程中断 OB 的临时 (TEMP) 变量。OB88 的缺省名被选为变量名。

变量	类型	描述
OB88_EV_CLASS	字节	事件级别和 ID: B#16#35
OB88_SW_FLT	字节	故障代码: <ul style="list-style-type: none"> <li>• B#16#73: 同步故障的嵌套深度太高</li> <li>• B#16#75: 块调用的嵌套深度 (B-堆栈) 太高</li> <li>• B#16#76: 在局部数据定位中故障</li> </ul>
OB88_PRIORITY	字节	优先级: 28
OB88_OB_NUMBR	字节	OB 号 (88)
OB88_BLK_TYPE	字节	出现故障的块的类型: <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: 不能决定中断源</li> </ul>
OB88_RESERVED_1	字节	备用
OB88_FLT_PRIORITY	字节	引起故障的 OB 的优先级
OB88_FLT_OB_NUMBR	字节	引起故障的 OB 的号码
OB88_BLK_NUM	字	引起故障的 MC7 指令的块的号码
OB88_PRG_ADDR	字	引起故障的 MC7 指令的相对地址
OB88_DATE_TIME	DATE_AND_TIME	OB 调用的日期和时间

## 1.24 背景组织块（OB90）

### 描述

STEP 7 可以监视最大扫描周期时间，并可以保证最小扫描周期时间。如果 OB1 包括所有的嵌套中断和系统功能的执行时间小于你所指定的最小扫描周期时间，操作系统反应如下：

- 它调用背景 OB（如果 CPU 中 OB90 已存在）。
- 它延时下一次 OB1 启动（如果在 CPU 中 OB90 不存在）。

### 理解 OB90 的运行

OB90 在所有 OB 中优先级最低。它可以被任何系统功能和任务所中断（甚至被 OB1 在最小周期时间超出时）并仅在如果所选的最小扫描周期时间还没有达到时保持。一个例外是在 OB90 中启动的 SFC 和 SFB 的执行。它们用 OB1 优先级的执行并且不会被 OB1 中断。没有 OB90 的时间监视。OB90 中的用户程序在下述情况下被从第一条指令处开始处理：

- 暖、冷或热启动之后
- OB90 中正在执行的块被删除之后（用 STEP 7）
- 在 RUN 方式下装 OB90 到 CPU 之后
- 结束了背景周期之后

### 注意

当组态中最小扫描周期时间和周期监视时间之间差距不大时，在背景 OB 中 SFC 和 SFB 调用可能会导致周期时间超出预期。

### OB90 的局部数据

下表描述 OB90 的临时（TEMP）变量。变量名为 OB90 的缺省名。

变量	类型	描述
OB90_EV_CLASS	字节	事件级别和标识：B#16#11：激活
OB90_STRT_INF	字节	<ul style="list-style-type: none"> <li>• B#16#91：暖启动/冷启动/热启动</li> <li>• B#16#92：块删除</li> <li>• B#16#93：在 RUN 方式下装 OB90 至 CPU</li> <li>• B#16#95：背景周期结束</li> </ul>
OB90_PRIORITY	字节	优先级：29（相当于优先级 0.29）
OB90_OB_NUMBR	字节	OB 号（90）
OB90_RESERVED_1	字节	备用
OB90_RESERVED_2	字节	备用
OB90_RESERVED_3	整数	备用
OB90_RESERVED_4	整数	备用
OB90_RESERVED_5	整数	备用
OB90_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

## 1.25 启动组织块（OB100, OB101 和 OB102）

### 启动的类型

下列启动类型是有区别的

- 热启动（S7-300 和 S7-400H 没有）
- 暖启动
- 冷启动

在下表中，你可以看到在启动期间哪个 OB 被操作系统调用。

启动类型	相应的 OB
热启动	OB101
暖启动	OB100
冷启动	OB102

关于启动类型更详细的信息，参见手册“用 STEP 7”编程、“用 STEP 7 组态硬件和连接”和“S7-400H PLC”。

### 启动事件

由于下列原因 CPU 执行启动：

- POWER ON 之后
- 当你把方式选择开关从 STOP 拨到 RUN 时
- 使用通讯功能请求之后（来自编程装置的手动命令或在另一 CPU 上通过调用通讯功能块 19 “START”或 21 “RESUME”）。
- 多处理器的同步
- 在 H 系统中在连接上之后（仅在备用的 CPU）

根据启动事件，特定的 CPU，和它的参数，相应的启动 OB（OB100, OB101, 或 OB102）被调用。随着适当的编程，你可以对你的循环程序做某些设置（例外：在 H 系统，当备用 CPU 被连接上时，备用的 CPU 上有启动但不调用启动 OB）。

### 启动 OB 的局部数据

下表描述启动 OB 的临时（TEMP）变量。

变量名为 OB100 的缺省名。

变量	类型	描述
OB10_EV_CLASS	字节	事件级别和标识: B#16#13: 激活
OB10x_STRTUP	字节	启动请求: <ul style="list-style-type: none"> <li>• B#16#81: 手动暖启动</li> <li>• B#16#82: 自动暖启动</li> <li>• B#16#83: 手动热启动请求</li> <li>• B#16#84: 自动热启动请求</li> <li>• B#16#85: 手动冷启动请求</li> <li>• B#16#86: 自动冷启动请求</li> <li>• B#16#87: 主站: 手动冷启动请求</li> <li>• B#16#88: 主站: 自动冷启动请求</li> <li>• B#16#8A: 主站: 手动暖启动请求</li> <li>• B#16#8B: 主站: 自动暖启动请求</li> <li>• B#16#8C: 备用: 手动启动请求</li> <li>• B#16#8D: 备用: 自动启动请求</li> </ul>
OB10x_PRIORITY	字节	优先级: 27
OB10x_OB_NUMBR	字节	OB 号 (100, 101, 或 102)
OB10x_RESERVED_1	字节	备用
OB10x_RESERVED_2	字节	备用
OB10x_STOP	字	引起 CPU 停机事件的号码
OB10x_STRT_INFO	双字	关于当前启动的进一步信息
OB10x_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间

下表显示变量 OB100\_STR\_INFO 和 OB101\_STR\_INFO。

位号	意义	可能的二进制值	解释
31 – 24	启动信息	0000 xxxx	机架号 0 (仅 H CPU)
		0100 xxxx	机架号 1 (仅 H CPU)
		1000 xxxx	机架号 2 (仅 H CPU)
		0001 xxxx	多处理器 (仅 S7-400)
		0010 xxxx	这段机架超过一个 CPU 在运行 (仅 S7-400)
		xxxx xxx0	期望和实际组态之间无不同 (仅 S7-300)
		xxxx xxx1	期望和实际组态之间不同 (仅 S7-300)
		xxxx xx0x	期望和实际组态之间无不同
		xxxx xx1x	期望和实际组态之间不同
		xxxx x0xx	不是 H CPU
		xxxx x1xx	H CPU
		xxxx 0xxx	在最后 POWER ON 时间标时时钟不是电池后备

位号	意义	可能的二进制值	解释
		xxxx 1xxx	在最后 POWER ON 时间标时时钟是电池后备
23 – 16	启动刚完成	0000 0001	根据参数赋值未改变 CPU 上设置的多处理器暖启动（仅 S7-400）
		0000 0011	方式选择开关触发的启动（暖）
		0000 0100	通过 MPI 由命令触发的启动（暖）
		0000 0101	根据参数赋值未改变 CPU 上设置的多处理器冷启动（仅 S7-400）
		0000 0011	方式选择开关触发的冷启动
		0000 1000	通过 MPI 由命令触发的冷启动
		0000 1010	根据参数赋值未改变 CPU 上设置的多处理器热启动（仅 S7-400）
		0000 1011	方式选择开关触发的热启动（仅 S7-400）
		0000 1100	通过 MPI 由命令触发的热启动（仅 S7-400）
		0001 0000	在电池后背 POWER ON 之后的自动启动（热）
		0001 0001	根据参数赋值在电池后备 POWER ON 之后的冷启动
		0001 0011	方式选择开关触发的启动（暖），最后 POWER ON 电池后备
		0001 0100	通过 MPI 由命令触发的启动(暖),最后 POWER ON 电池后备
		0010 0000	在电池后备 POWER ON 之后(由系统存储器复位)自动启动（暖）
		0010 0001	在电池后备 POWER ON 之后(由系统存储器复位)冷启动
		0010 0011	方式选择开关触发的启动（暖），最后 POWER ON 无电池后备
		0010 0100	通过 MPI 由命令触发的启动(暖),最后 POWER ON 无电池后备
		1010 0000	根据参数赋值在电池后备 POWER ON 之后的热启动（仅 S7-400）
15 – 12	允许自动启动吗	0000	自动启动非法，需存储器复位
		0001	自动启动非法，需修改参数等
		0111	自动启动允许（暖）
		1111	自动启动允许（暖/热）（仅 S7-400）
11 – 8	允许手动启动吗	0000	启动非法，需存储器复位
		0001	启动非法，需修改参数等
		0111	启动允许（暖）
		1111	启动允许（暖/热）（仅 S7-400）

位号	意义	可能的二进制值	解释
7-0	最后有效的干涉或在 POWER ON 自动启动的设置	0000 0000	无启动
		0000 0001	根据参数赋值在 CPU 上的设置无改变的多处理器暖启动（仅 S7-400）
		0000 0011	方式选择开关触发的启动（暖）
		0000 0100	通过 MPI 由命令触发的启动（暖）
		0000 0101	根据参数赋值在 CPU 上的设置无改变的多处理器热启动（仅 S7-400）
		0000 0111	方式选择开关触发的冷启动
		0000 1000	通过 MPI 由命令触发的冷启动
		0000 1010	根据参数赋值在 CPU 上的设置无改变的多处理器热启动（仅 S7-400）
		0000 1011	方式选择开关触发的热启动（仅 S7-400）
		0000 1100	通过 MPI 由命令触发的热启动（仅 S7-400）
		0001 0000	在电池后备 POWER ON 之后自动启动（暖）
		0001 0001	根据参数赋值在电池后备 POWER ON 之后的冷启动
		0001 0011	方式选择开关触发的启动（暖），最后 POWER ON 电池后备
		0001 0100	通过 MPI 由命令触发的启动（暖），最后 POWER ON 电池后备
		0010 0000	在电池后备 POWER ON 之后自动启动（暖）（由系统复位存储器）
		0010 0001	根据参数赋值在电池后备 POWER ON 之后的冷启动
		0010 0011	方式选择开关触发的启动（暖），最后 POWER ON 无电池后备
		0010 0100	通过 MPI 由命令触发的启动（暖），最后 POWER ON 无电池后备
		1010 0000	根据参数赋值在电池后备 POWER ON 之后自动热启动（仅 S7-400）

## 1.26 编程故障组织块（OB121）

### 描述

当有关程序处理的故障引起的事件发生时 CPU 的操作系统调用 OB121。例如，如果程序调用了 CPU 中存在的块，OB121 就被调用。

### 理解编程故障 OB 的运行

OB121 与被中断的块在同一优先级中执行。

如果 OB121 未编程，CPU 从 RUN 方式进入 STOP 方式。

S7 提供了下列 SFC 在你的程序执行期间屏蔽和解除屏蔽 OB121 的启动事件：

- SFC36 (MSK\_FLT)：屏蔽特定的故障代码
- SFC37 (DMSK\_FLT)：解除被 SFC36 屏蔽的故障代码
- SFC38 (READ\_ERR)：读故障寄存器

### 编程故障 OB 的局部数据

下表描述编程故障 OB 的临时 (TEMP) 变量。变量名为 OB121 的缺省名。

变量	类型	描述
OB121_EV_CLASS	字节	事件级别和标识：B#16#25
OB121_SW_FLT	字节	故障代码： (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	字节	优先级=出现故障的 OB 的优先级
OB121_OB_NUMBR	字节	OB 号 (121)
OB121_BLK_TYPE	字节	故障出现的块的类型 (在 S7-300 时无有效值在这里记录): B#16#88: OB, B#16#8A: DB, B#16#8C: FB, B#16#8E: FB)
OB121_RESERVED_1	字节	备用
OB121_FLT_REG	字	故障源 (根据故障代码)。 例如： <ul style="list-style-type: none"> <li>• 转换故障发生的寄存器</li> <li>• 不正确的地址 (读/写故障)</li> <li>• 不正确的定时器/计数器/块号码</li> <li>• 不正确的存储区</li> </ul>
OB121_BLK_NUM	字	引起故障的 MC7 命令的块的号码 (S7-300 无效)
OB121_PRG_ADDR	字	引起故障的 MC7 命令的块的号码 (S7-300 无效)
OB121_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间



根据故障代码变量有下述意义：

故障代码	意义
B#16#21: OB121_FLT_REG:	BCD 转换故障 有关的寄存器的 ID (W#16#0000: 累加器 1)
B#16#22: B#16#23: B#16#28: B#16#29:  OB121_FLT_REG:  OB121_RESERVED_1:	当读时区域长度故障 当写时区域长度故障 用指针读访问字节、字、或双字时位地址不为 0。 用指针写访问字节、字、或双字时位地址不为 0。 不正确的字节地址。可以从 OB121_RESERVED_1 中读出数据区和访问类型。 位 7 至 4 访问类型。 0: 位访问 1: 字节访问 2: 字访问 3: 双字访问 位 3 至 0 存储区: 0: I/O 区 1: 过程映象输入表 2: 过程映象输出表 3: 位存储器 4: 全局 DB 5: 背景 DB 6: 自己的局部数据 7: 调用者的局部数据
B#16#24: B#16#25: OB121_FLT_REG:	当读时范围故障 当写时范围故障 在低字节含有非法区的 ID (B#16#86 自己的局部数据区)
B#16#26: B#16#27: OB121_FLT_REG:	定时器号码故障 计数器号码故障 非法的号码
B#16#30: B#16#31: B#16#32: B#16#33: OB121_FLT_REG:	写访问至写保护的全局 DB 写访问至写保护的背景 DB 访问全局 DB 时 DB 号码故障 访问背景 DB 时 DB 号码故障 非法的 DB 号
B#16#34: B#16#35: B#16#3A: B#16#3C: B#16#3D: B#16#3E: B#16#3F: OB121_FLT_REG:	在 FC 调用时 FC 号码故障 在 FB 调用时 FB 号码故障 访问未下装的 DB, DB 号在允许范围 访问未下装的 FC, FC 号在允许范围 访问未下装的 SFC, SFC 号在允许范围 访问未下装的 FB, FB 号在允许范围 访问未下装的 SFB, SFB 号在允许范围 非法号码

## 1.27 I/O 访问故障组织块（OB122）

### 描述

当对模板的数据访问出现故障时 CPU 的操作系统调用 OB122。例如，如果 CPU 对 I/O 模板的数据访问时检测到读故障，操作系统调用 OB122。

### 理解 I/O 访问故障 OB 的运行

OB122 与被中断的块在同一优先级中执行。如果 OB122 未编程，CPU 从 RUN 方式进入 STOP 方式。

S7 提供了下列 SFC 在你的程序执行期间屏蔽和解除屏蔽 OB122 的启动事件：

- SFC36 (MSK\_FLT)：屏蔽特定的故障代码
- SFC37 (DMSK\_FLT)：解除被 SFC36 屏蔽的故障代码
- SFC38 (READ\_ERR)：读故障寄存器

### I/O 访问故障 OB 的局部数据

下表描述 I/O 访问故障 OB 的临时 (TEMP) 变量。变量名为 OB122 的缺省名。

变量	类型	描述
OB122_EV_CLASS	字节	事件级别和标识：B#16#29
OB122_SW_FLT	字节	故障代码： <ul style="list-style-type: none"> <li>• B#16#42 对 S7-300 和 CPU 417：I/O 访问故障，读操作对所有其它的 S7-400CPU：在故障出现之后在第一次读访问期间故障</li> <li>• B#16#43 对 S7-300 和 CPU 417：I/O 访问故障，写操作对所有其它的 S7-400 CPU：在故障出现之后在第一次写访问期间故障</li> <li>• B#16#44 (仅对 S7-400，不包括 CPU 417) 在故障出现之后在第 n 次 (n &gt; 1) 写访问期间故障</li> <li>• B#16#45 (仅对 S7-400，不包括 CPU 417) 在故障出现之后在第 n 次 (n &gt; 1) 写访问期间故障</li> </ul>
OB122_PRIORITY	字节	优先级： 出现故障的 OB 的优先级
OB122_OB_NUMBR	字节	OB 号 (122)
OB122_BLK_TYPE	字节	故障出现的块的类型 (B#16#88：OB，B#16#8A：DB，B#16#8C：FC，B#16#8E：FB) (对 S7-300 无有效值在这里记录)

变量	类型	描述
OB122_MEM_AREA	字节	存储器区和访问类型： <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：存储器区<ul style="list-style-type: none"><li>– 0：I/O 区</li><li>– 1：过程映像输入</li><li>– 2：过程映像输出</li></ul></li></ul>
OB122_MEM_ADDR	字	出现故障的存储器地址
OB122_BLK_NUM	字	引起故障的 MC7 命令的块的号码（S7-300 无效值在这里记录）
OB122_PRG_ADDR	字	引起故障的 MC7 命令的相对地址（S7-300 无效值在这里记录）
OB122_DATE_TIME	DATE_AND_TIME	OB 被调用时的日期和时间



## 2 SFC 常用参数

### 2.1 利用输出类型参数 RET\_VAL 作故障诊断

#### 故障信息类型:

一个在用户程序中执行的系统功能（SFC）表明了该 CPU 是否有能力正确地执行 SFC 中的功能。你可以通过以下两种途径获取故障信息：

- 通过状态字中的 BR 位；
- 通过输出类型参数 RET\_VAL（返回值）

---

#### 注意:

在对 SFC 特定的输出类型参数进行检测之前，你应该遵循以下步骤。

- 首先，查看状态字中的 BR 位；
- 然后，检查输出类型参数 RET\_VAL。

如果 BR 位显示有错误发生或者 RET\_VAL 包含有故障代码，这时你就不能对 SFC 的输出类型参数做检测了。

---

#### 返回值中的故障信息:

系统功能（SFC）要求当状态字中的二进制结果位 BR 被置 0 时要产生一个错误，一些系统功能在返回值（RET\_VAL）端预制了一些附加的故障代码。

如果输出类型参数 RET\_VAL 端产生了一个一般的错误（见以下解释），只在状态字的 BR 位显示“0”。

返回值的数据类型为整数（INT），返回值和与之相关的“0”值表明是否在执行该功能时产生了错误。

CPU 执行的 SFC	BR	返回值	整数符号
有故障	0	小于“0”	负值（符号位为 1）
无故障	1	大于等于“0”	正值（符号位为 0）

#### 故障信息的反馈

在 RET\_VAL 端有以下两种不同类型的故障代码：

- 一般故障代码，所有的系统功能都可以输出。
- 特殊的故障代码，具有该特殊功能的系统功能才可以输出。

你可以编写自己的用户程序，它会对执行系统功能时产生的错误有影响，这样你就可以避免由于第一个错误的产生而引发更多的错误。

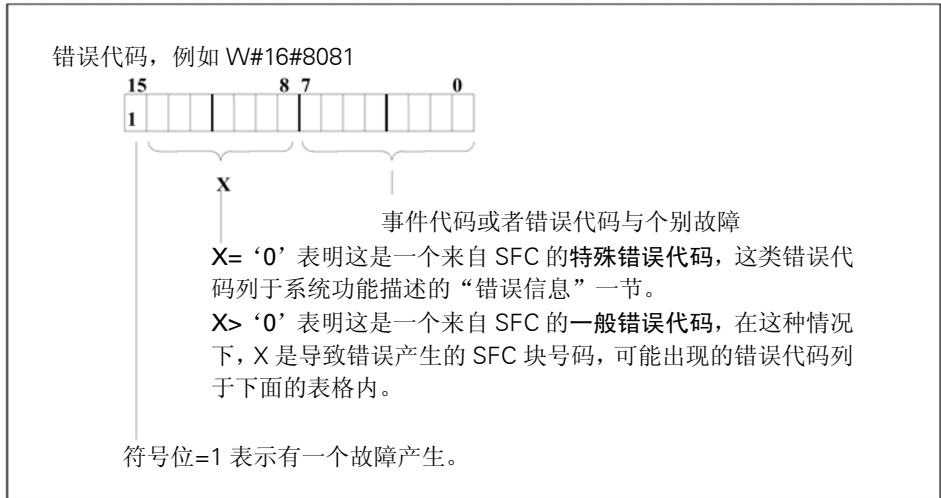
#### 一般和特殊的故障信息

系统功能的返回值（RET\_VAL）反映了以下两种故障代码之一。

- 一般故障代码，它涉及到了任何系统功能产生的错误。
- 特殊故障代码，它只涉及到了某些特殊的系统功能。

虽然输出类型参数 RET\_VAL 是整型 (INT) 数据，但是系统功能的故障代码是按十六进制数据进行分类，如果你想检测某一个返回值，并且要将它与本手册中列出的故障代码作比较，那么显示的故障代码用十六进制格式表示。

下图为用十六进制格式表示的系统功能故障代码的结构。

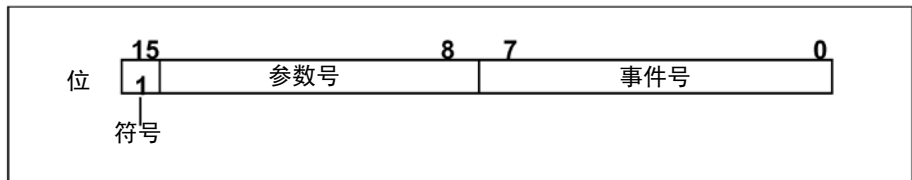


### 一般故障信息

一般故障信息用来表示在任意系统功能中产生的错误，一般故障代码包括有以下两个号码：

- 由 1 到 111 的参数号，这里 1 表示 SFC 的第 1 个参数，2 表示 SFC 的第 2 个参数等。
- 由 0 到 127 的事件号，事件号表示产生了一个同步错误。

下图列出了一般故障代码格式和每一个故障的说明。



### 注意：

如果 RET\_VAT 产生了一个故障代码，可能会产生以下情况：

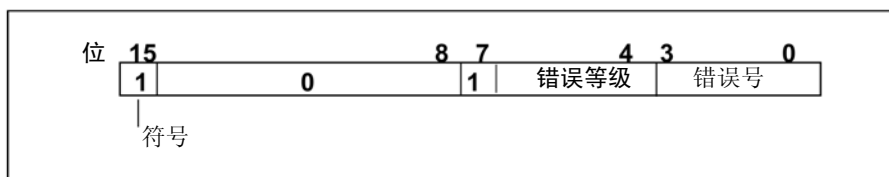
- 与 SFC 相关的动作可能已经开始或者已经完成。
- 当一个动作执行后可能会产生一个特殊的 SFC 错误，然而，作为后续发生的一般故障的结果，该特殊故障将不再指示。

### 特殊错误信息

一些系统功能 (SFC) 具有特定故障代码的返回值，这些故障代码表示在执行某一个特殊的系统功能时所产生的特殊错误 (见图)。一个特殊的故障代码包含有以下两个号码。

- 从 0~7 的故障等级

- 从 0~15 的错误号



### 一般故障代码

下表列出了作为返回值的一般故障代码，故障代码以十六进制格式表示。

在每一段代码中的 X 表示导致该故障产生的系统功能参数号。

故障代码 (W#16#...)	解释
8X7F	内部错误 这个错误表示在参数 X 处产生了一个内部错误，这个错误不是由用户引起的，也不能被用户去除。
8X01	在 ANY 变量处有非法的 ID
8X22	读取参数时长度范围错误。
8X23	写参数时长度范围错误。 这个错误表示参数 X 完全或者部分位于地址范围以外，也可能是 ANY 变量的位长度不是 8 的倍数。
8X24	读取参数时范围错误
8X25	写参数时范围错误 这个错误表示参数 X 位于非法的系统功能区域内，参阅个别功能描述中沙土范围信息的说明。
8X26	参数所包含的定时器号过大。 这个错误表示参数 X 中指定的定时器不存在
8X27	参数所包含的计数器过大（计数器号错误） 这个参数表示参数 X 中指定的计数器不存在。
8X28	读取参数时检验错误
8X29	写参数时检验错误 这个故障代码表示参考地址位 X 不等于。
8X30	参数位于只读的全局 DB。
8X31	参数位于只读的背景 DB 这个故障代码表明参数 X 位于只读数据块，如果数据块被系统功能自己打开，系统功能返回值总是 W#16#8X30。
8X32	参数所包括的 DB 号过大（DB 号错误）
8X34	参数所包括的 FC 号过大（FC 号错误）
8X35	参数所包括的 FB 号过大（FB 号错误） 这个故障代码表示参数 X 所包含的块号大于允许的最大值
8X3A	参数所包括的 DB 块没有装载。
8X3C	参数所包括的 FC 块没有装载。
8X3E	参数所包括的 FB 块没有装载。

故障代码 (W#16#...)	解释
8X42	当系统从外部输入区读取参数时产生的访问错误。
8X43	当系统向外部输出区与参数时产生的访问错误。
8X44	错误发生后第 n 次 (n>1) 读访问错误
8X45	错误发生后第 n 次 (n>1) 写访问错误 该故障代码表明对所需参数的访问被拒绝。

## 2.2 异步 SFC 中 REQ, RET\_VAL 和 BUSY 参数的含义

### 异步 SFC

SFC 异步运行是指 SFC 在完成操作之前被调用多次，以下的 SFC 经常被用于异步操作或某些特定场合。

- SFC 7 “DP\_PRAL”
- SFC 11 “DPSYC\_FR”
- SFC 12 “D\_ACT\_DP”
- SFC 13 “DPNRM\_DG”
- SFC 51 “RDSYSST”
- SFC 55 “WR\_PARM”
- SFC 56 “WR\_DPARM”
- SFC 57 “PARM\_MOD”
- SFC 58 “WR\_REC”
- SFC 59 “RD\_REC”
- SFC 65 “X\_SEND”
- SFC 67 “X\_GET”
- SFC 68 “X\_PUT”
- SFC 69 “X\_ABORT”
- SFC 72 “I\_GET”
- SFC 73 “I\_PUT”
- SFC 74 “I\_ABORT”
- SFC82 “CREA\_DBL”
- SFC83 “READ\_DBL”
- SFC84 “WRIT\_DBL”
- SFC90 “H\_CTRL”
- SFC102 “RD\_DPARA”
- SFC103 “DP\_TOPOL”



## 作业识别

如果你触发一个硬件中断，并且向 DP 从站发出输出控制指令，启动数据传输或者取消了一个上面列出的某一个 SFC 的非设定连接，然后在当前工作完成之间再次调用相同的 SFC，对 SFC 的影响要看是否第二次调用包含有相同的内容。

SFC	作业识别
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

## 输入参数 REQ

输入类型参数用于单独的启动某一事件：

- 如果你调用 SFC 作为一个当前没有激活的事件，这个事件只有在 REQ=1（第一种情况）时被激活。
- 如果某一个特定的事件已经被激活并且还没有完成，而且你又调用了 SFC 执行相同的事件（例如：在一个循环中断 OB 中）那么 REQ 不被 SFC 检测（第二种情况）。

## 输出参数 RET\_VAL 和 BUSY

作业运行状态由输出类型参数 RET\_VAL 和 BUSY 来表示。

参考并且要注意利用输出类型参数 RET\_VAL 作故障检测

- 情况 1（首次调用 REQ=1）如果有足够的系统资源并且输入参数正确，RET\_VAL 为 W#16#7001，那么 BUSY 被置位。  
如果系统资源被占用或者输入的参数有误，相当的故障代码会被写入 RET\_VAL 并且 BUSY 为 0。
- 情况 2（重复调用时）W#16#7002 被写入 RET\_VAL（这是一个提示工作正在运行的报警），并且 BUSY 被置位。
- 以下情况用于最后一次调用：
  - 如果没有错误，则 SFC13 “DPNRM\_DG”，SFC67 “X\_GET” 和 SFC72 “I\_GET” 的 RET\_VAL 参数显示所传送数据的数量数值以字节为单位，并且 BUSY 值为 0。  
如果没有错误，RET\_VAL 显示错误信息，BUSY 值为 0。
  - 如果没有错误，RET\_VAL 显示 SFC 59 “RD\_REC” 的值（字节），BUSY 值为 0。  
如果没有错误，RET\_VAL 显示故障代码，BUSY 值为 0。
  - 对所有 SFC 来说，如果没有错误，RET\_VAL 值为 0，BUSY 值为 0。  
如果出错，RET\_VAL 显示故障代码，BUSY 值为 0。

**注意**

如果第一次和最后一次调用，同时发生，RET\_VAL 和 BUSY 显示的是最后一次调用后的值。

**概述**

下表提供给你一个综上所述的所有关系的概况，特别是当一个 SFC 被调用但是这个事件还没有执行完时，在输出类型参数中可能会出现值。

**注意**

在每次调用后，你必须对你程序中的输出类型参数进行检查

在执行某一个事件时 Call，REQ，RET，RET\_VAL 和 BUSY 之间的关系。

调用次数	调用类型	REQ	RET_VAL	BUSY
1	第一次调用	1	W#16#7001	1
			故障代码	0
2 to (n-1)	中间调用	无关系	W#16#7002	1
N	最后一次调用	无关系	如果没有故障产生 W#16#0000（例外： SFC59 “RD_DEL” 如果目标地址大于数据传输的记录值，SFC13“DPNRM_DG”， “SFC67” X_GET 和 SFC72 “I_GET”）	0
			产生故障的故障代码	0

## 3 拷贝与块功能

### 3.1 用 SFC20 “BLKMOV” 拷贝变量

#### 描述

通过 SFC20 “BLKMOV”（块移动）可以将一个存储区（源存储区域）的内容拷贝到另外一个存储区（目标区域）。

用 SFC20 “BLKMOV” 可以拷贝除下列之外的所有存储区：

- 程序块：FB, SFB, FC, SFC, OB, SDB
- 计数器
- 定时器
- 外设 I/O 的存储区

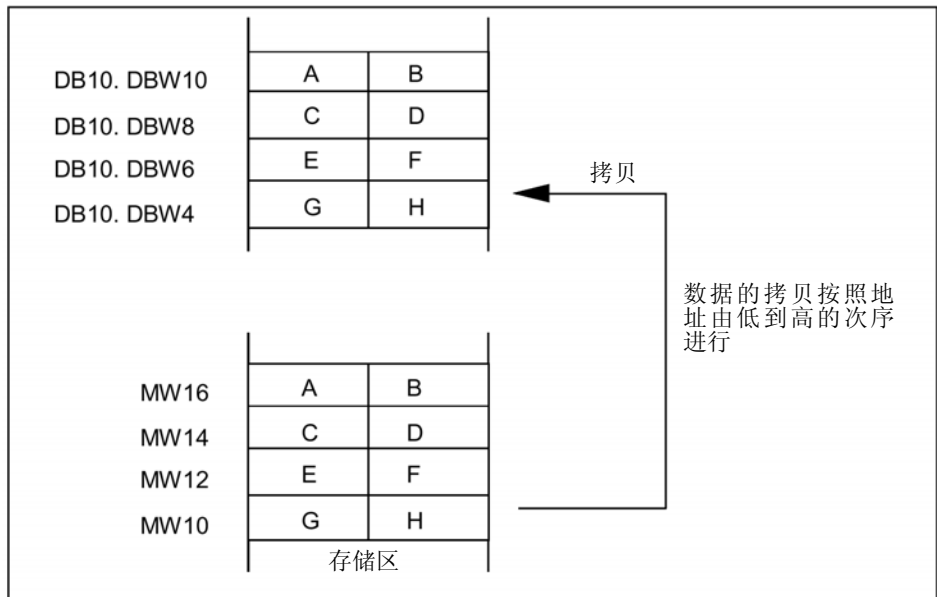
源数据可以是与程序执行无关的装载存储器中数据块的一部分（此数据块是由关键字 UNLINKED 编译的）

---

#### 注意

如果 CPU 中装有 SFC83, 那么必须用此 SFC 读取装载存储器中与程序执行无关的数据块, 如果使用 SFC20 会产生一个故障信息 W#16#8092。

---



#### 中断能力

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，如果当从一个与当前程序执行无关的数据块中拷贝时 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	说明
SRCBLK	INPUT	ANY	I, Q, M, D, L	指定被拷贝的存储区（源区域）。不允许指定字符串类型的数组。
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	指定拷贝的存储区（目标区域），不允许指定字符串类型的数组。

#### 注意

源区域与目标区域不可以交叉。如果目标区域大于源区域，只把源区域中包含的数据拷到其中。

如果目标区域小于源区域，则只拷贝目标区域中能容纳的数据。

如果“ANY”类型（源或目标区域）的数据为 BOOL，定义的长度必须可以被 8 除，否则此 SFC 将不被执行。

源或目标参数（或两者都）也可以 STRING 数据类型。如源是字符串，只拷贝字符串的当前字符。如果目标是字符串，拷贝字符的当前长度被写入，ARRAY OF STRING 不能被拷贝，这意味着只有“STRING1”是允许的。

**特殊性能：**如果一个被链接的数据块通过 SFC20BLKMOV 拷贝并装载到了 RAM 中，例如通过编程器，则此 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 拷贝块时装载这类块。

#### 故障信息

故障信息 (W#16#...)	说明
0000	无故障
8091	嵌套深度超出。 源区域不在相关执行的数据块中。

## 3.2 用 SFC81 “UBLKMOV” 不间断地拷贝变量

### 描述

通过 SFC81 “UBLKMOV”（不间断的块移动）可以将存储区（源区域）的内容原封不动地拷贝到另一个存储区（目标区域），此拷贝操作不能被其它操作系统的任务打断。

用 SFC81 “UBLKMOV”，可以拷贝除下列之外的所有存储区：

- 程序块：FB, SFB, FC, SFC, OB, SDB
- 计数器
- 定时器
- 外设 I/O 存储区
- 与运行无关的块

可以拷贝最大 512 字节的数据，注意不同的 CPU 此数值不同，可以在指令集中找到相关信息。

### 中断能力，中断响应时间

拷贝不可以被中断。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	说明
SROBLK	INPUT	ANY	I, Q, M, D, L	指定被拷贝的存储区（源区域）不允许指定字符串类型的数组。
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	指定拷贝的存储区（目标区域）不允许指定字符串类型的数组。

### 注意

源与目标区域不可以交叉。如果目标区域大于源区域，只把源区域中包含的数据拷到其中。如果目标区域小于源区域，则只拷贝目标区域中能容纳的数据。

如果“ANY”类型（源或目标区域）的数据为 BOOL，定义的长度必须可以被 8 除，否则此 SFC 不被执行。

如果“ANY”类型的数据为 STRING，定义的长度必须等于 1。

### 故障信息

故障代码 (W#16#...)	说明
0000	无故障
8091	源区域为位于未链接的数据块中
8092	“拷贝变量”功能不能被执行因为进入的数据块是不能被执行的，需用 SFC83 实现拷贝功能。

### 3.3 用 SFC21 “FILL” 初始化存储区

**描述**

通过 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。

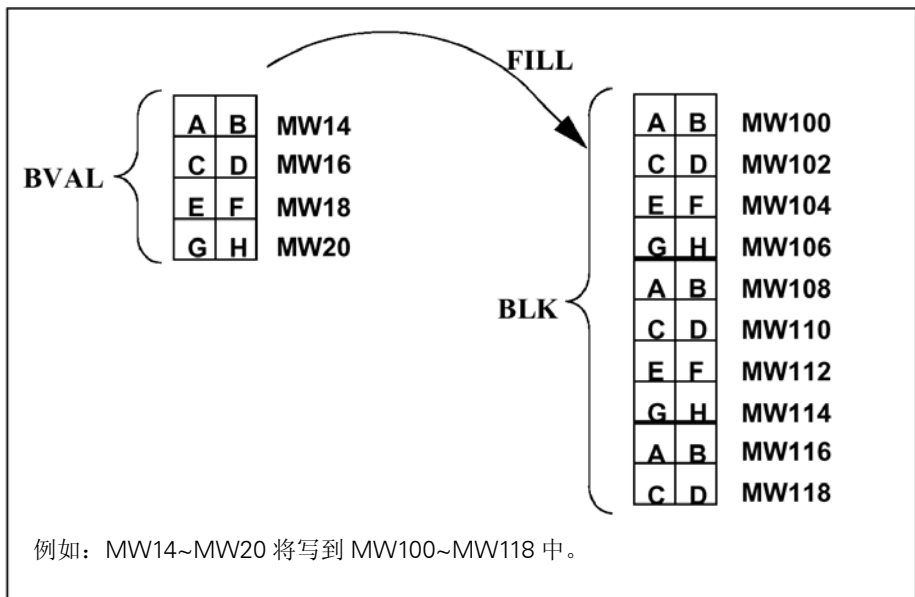
**注意**

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个字节。

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。

如果任意类型（源或目标区域）的数据为布尔量，则指定的长度必须可以被 8 除，否则此 SFC 将不被执行。



**例外**

通过 SFC21 不能将数据写到下列单元中：

- 程序块：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	说明
BVAL	INPUT	ANY	I, Q, M, D, L	参数 BVAL 包含用于初始化目标区域的源区域的值或说明。不允许使用字符串数组。
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，不允许使用字符串类型的数组。

#### 数据以结构形式输入

如果输入数据作为结构传输，注意以下事项：

STEP7 总是把一个结构的长度定义为偶数个字节数，所以，如果声明一个包含奇数个字节的结构，需要另加一个字节的存储空间。

例如

如下声明一个结构

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

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

## 3.4 用 SFC22 “CREAT\_DB” 生成数据块

#### 描述

用 SFC22“CREAT\_DB”（生成数据块），可以生成一个不包含初始化数据的数据块，此 SFC 可以生成一个可选长度的数据块，数据块号在一个特定范围内，系统功能给数据块分配一个可能的最低编号。如果想生成一个特殊编号的功能块，可取最高或最低限值，在用户程序中已存在的 OB 块编号不可以再使用，DB 块的长度必须是偶数个字节。

#### 中断能力

SFC22 “CREAT\_DB” 可被更高优先级的 OB 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	说明
Low_LIMIT	INPUT	WORD	I, Q, M, D, L, 常数	下限值是可以分配给 DB 块号的最小值。
UP_LIMIT	INPUT	WORD	I, Q, M, D, L, 常数	上限值是可以分配给 DB 块号的最大值。
COUNT	INPUT	WORD	I, Q, M, D, L, 常数	计数值为 DB 块的字节号, 这里必须指定偶数个字节 (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	生成的数据块的编号, 如果出现故障 (RET_VAL 的第 15 位置 1), DB_NUMBER 置 0。

## 故障信息

故障代码 (W#16#...)	说明
0000	无故障发生
8091	调用了嵌套的 SFC-22
8092	“生成数据块”功能不能被执行因为 <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”功能正在进行。</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	数据块编号错误 <ul style="list-style-type: none"> <li>编号为 0</li> <li>编号超出本 CPU 规定的数据库的编号值</li> <li>参数下限大于上限</li> </ul>
80A2	数据块长度错误 <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 (提示: 压缩存储器!)



### 3.5 用 SFC23 “DEL\_DB” 删除数据块

#### 描述

用 SFC23 “DEL\_DB”（删除数据块）可以删除一个存在于工作存储器 and 当前存在于装载存储器中的数据块。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，此数据块一定不是位于两个 DB 寄存器中的任意一个或 B 堆栈中。否则，CPU 在调用 SFC23 时启动 OB121。如果 OB121 不存在，则 CPU 停机。

下表解释何时可以通过 SFC23 “DEL\_DB” 删除数据块。

如果	那么
数据块是由 SFC22 “CREAT_DB” 产生的	SFC23 可删除它
数据块是通块 STEP7 传送到 CPU 中，并且不是通过关键字 UNLINKED 生成的	SFC23 可删除它
数据块存储于闪存卡中	SF23 不能删除它

#### 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	说明
DB_NUMBER	INPUT	WORD	I, Q, M, D, L, 常数	被删除数据块的编号
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息

故障代码 (W#16#...)	说明
0000	无故障发生
8091	SFC23 嵌套调用，并且 CPU 使用的回嵌套深度超出最大值。
8092	当前不能执行删除数据块功能因为 <ul style="list-style-type: none"> <li>• “压缩用户存储器” 功能当前激活</li> <li>• “存储用户程序” 功能当前激活</li> <li>• 正在上装被删除的数据块</li> <li>• H CPU 正在执行链接或者升级功能</li> <li>• 在装有 WinAC 软件的计算机的操作系统中 CPU 检测到一个故障</li> </ul>
80A1	输入参数 DB_NUMBER 故障：实际选择的参数 <ul style="list-style-type: none"> <li>• 为 0</li> <li>• 大于 CPU 使用的数据块编号的最大值</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 产生的
80B3	数据块位于闪存卡上
80B4	数据块不能被删除，因为它是下一库中的块
80C1	因为暂时资源紧张，删除数据块的功能当前不能被执行

### 3.6 用 SFC24 “TEST\_DB” 测试数据块

#### 描述

通过 SFC24 “TEST\_DB”（测试数据块），可以得到位于 CPU 的工作存储器中的数据块的信息。SFC 查询所选数据块的编号并检查其是否为只读数据块。

参数	声明	数据类型	存储区域	说明
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	故障信息
DB_LENGTH	OUTPUT	WORD	I, Q, M, D, L	所选数据块包含的字节数
WRITE_PROT	OUTPUT	BOOL	I, Q, M, D, L	关于数据块写保护标记的信息（1 表示只读）

#### 故障信息

故障代码 (W#16#...):	说明
0000	无故障发生
80A1	输入参数 DB_NUMBER 故障：实际选择的参数 <ul style="list-style-type: none"> <li>• 为 0</li> <li>• 大于 CPU 使用的数据块编号的最大值</li> </ul>
80B1	指定编号的数据块在 CPU 中不存在
80B2	数据块是通过关键字 UNLINRED 产生的

### 3.7 用 SFC25 “COMPRESS” 压缩用户存储器

#### 存储器的空隙

当数据块被删除或重装几次后，在工作存储器和装载存储器中就会产生空隙。这些空隙减少了存储区的可用空间。

#### 描述

通过 SFC25 “COMPRESS”，可以压缩工作存储器和装载存储器的 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并且一直激活，则调用 SFC25 将产生一个故障信息。

---

#### 注意

长度大于 1000 字节的数据块不能被 SFC25 “COMPRESS” 移动，这意味着压缩之后空隙仍存在。

---

参数	声明	数据类型	存储区域	说明
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
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结束（1 表示已成功结束）

### 检查压缩功能

一旦 SFC25 “COMPRESS” 被调用，压缩功能就开始执行了。但是，不能检查存储器空间是否被成功压缩。

如果想检查压缩功能，需遵循以下步骤：

循环调用 SFC25，每次调用后首先查看 KET\_VAL。如果此值为 0，则可查看参数 BUSY 和 DONE。如果 BUSY=1 和 DONE=0，表示压缩功能仍在进行。如果 BUSY 变为 0，DONE 变为 1，则表示压缩已成功完成。此后再调用 SFC25，压缩功能再次开始执行。

### 故障信息

故障代码 (W#16#...)	说明
0000	无故障发生，压缩功能已通过调用 SFC25 开始，在这种情况下通过检查用户程序中的输出参数 BUSY 和 DONE 仅能提供有用的信息。
8091	压缩功能由外部启动并且仍在执行。
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”功能是由外部启动且仍在执行</li> <li>测试和启动功能当前需要一个特殊的块（例如，状态）</li> <li>“拷贝块”功能已由外部触发并且仍在执行</li> <li>H-CPU 正在执行链接或升级功能</li> </ul>

## 3.8 用 SFC44 “REPL\_VAL” 传送一个替代值到累加器 1

### 描述

通过 SFC44 “REPL\_VAL”（替换值），可以传送一个参数到累加器 1 中，这个参数存在于可以发出故障的优先级中。

**限制条件：**只能是位于同步故障 OB 中

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

### 应用示例

如果一块输入模板损坏到不能读取任何值的程度，则每次扫描此模板时 OB122 都启动。用 SFC44 “REPL\_VAL”，可以把 OB122 中的参数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代值。选择此替代值（例如，发生故障的块或者受到影响的地址）的信息在 OB122 的局部变量中。

参数	声明	数据类型	存储区域	说明
VAL	INPUT	DWORD	I, Q, M, D, L, 常数	替代值
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如果在功能执行时发生故障，此参数中包含一个故障代码

故障信息

故障代码 (W#16#...)	说明
0000	无故障发生，替代值被输入
8080	SFC44 不是被同步故障组织块 (OB121, OB122) 调用

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

描述

通过 SFC82 “CREA\_DBL” (在装载存储器中生成数据块) 可以在装载存储器中 (存储器卡) 中生成一个新的数据块。SFC82 生成一个缺省值大小的数据块，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此数据块为拷贝得到的，则 SFC 终止运行，并生成一条故障信息。

注意

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

参数 SRCBLK 指向的数据区的内容被写到数据块中。该数据区必须是一个数据类型为 BLOCK\_DB 的数据块或者是来自数据块的区域。为了保持一致，当 SFC82 被执行时不能改变该数据区。

一个带有只读属性的数据块只能由 SFC82 生成和初始化。

SFC82 不改变用户程序的检查和。

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 SFC82，启动生成数据块功能。

作业状态通过输出参数 RET\_VAL 和 BUSY 显示。

关于异步运行 SFC，请参见 REQ、RET\_VAL 和 BUSY 的说明。

参数	声明	数据类型	存储区域	说明
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, 必须指定偶数个字节
ATTRIB	INPUT	BYTE	I, Q, M, D, L	数据块的属性:
				位 0 = 1: UNLINKED: 此数据块只存在于装载存储器中
				位 1 = 1: READ_ONLY: 此数据块具有写保护
				位 2 = 1: NON_RETAIN: 此数据块不保持
				位 3 到 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

## 故障信息

故障代码 (W#16#...)	说明
0000	无故障
0081	目标区域比源区域大 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 无关): 数据传输已激活; BUSY 为 1
8081	源区域比目标区域大 目标区域写满后, 源区域中的其它数据被忽略
8091	调用了嵌套的 SFC82
8092	“生成一个数据块”功能当前不能被执行因为 <ul style="list-style-type: none"> <li>• “压缩应用存储器”操作正在进行</li> <li>• H CPU 在连接或升级状态</li> <li>• Windows NT 操作系统瘫痪 (蓝屏)</li> </ul>
8093	没有块或者没有与此操作有关的块指定给参数 SRCBLK (初始化块)
8094	一个不被支持的属性被指定给参数 ATTRIB

故障代码 (W#16#...)	说明
80A1	数据块编号故障： <ul style="list-style-type: none"> <li>• 编号为 0</li> <li>• 下限大于上限</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 允许的数值</li> </ul>
80B1	没有可用的数据块编号
80B2	存储器不够用
80B3	连续存储空间不够（需压缩）
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	需要的系统资源正被占用着
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拷贝版</li> <li>• 数据块所在的源区域不存在</li> </ul>

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

#### 描述

通过 SFC83 “READ\_DBL”（读装载存储器中的数据块）可以将装载存储器（存储器卡）一个数据类型为 BLOOK\_DB 的数据块或者数据块的数据区放到目标数据块的数据区中。目标数据块必须与运行有关（关键字 UNLINKED=0）。源区域可以与运行无关（关键字 UNLINKED=1）。在读操作中装载存储器的内容不改变。

为了保持一致，在 SF83 执行时一定要不要改变目标区域。

下面的限制条件适用于参数 SRCBLK 和 DSTBLK

- 如果“ANY”数据类型为布尔量，则长度必须可被 8 除。
- 如果“ANY”数据类型为布尔量，则长度必须可被 1 除。

如果需要，可以通过 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地读取变量。

---

**注意**

一旦开始，作业就必须被完成，即使具有更高优先权的任务也需要同样的资源。如果高的优先级显示代码为 80C3 的故障，并不马上重启高优先级制的任务。而是必须要等到此块的操作完成后。

**工作方式**

SFC83 “READ\_DBL” 是异步运行的，即处理过程包含此 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 与 BUSY 的说明。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	E, A, 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	故障代码
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1, 读操作还未结束
DSTBLK	OUTPUT	ANY	D	目标数据块数据区的指针

**故障信息**

故障代码 (W#16#...)	说明
0000	无故障
8081	源区域比目标区域大 目标区域充满，源区域中其它数据被忽略
7000	REQ=0 首次调用：没有数据传输；BUSY 为 0
7001	REQ=1 首次调用：没有数据传输；BUSY 为 1
7002	中间调用（与 REQ 无关）：数据传输功能激活；BUSY 为 1
0081	目标区域比源区域大 源区域中数据全部写到目标区域。目标区域中其它数据保持原状
80C0	目标区域正被另外一个 SFC 或通讯操作占用
8093	没有块或者没有与此操作有关的块指定给参数 DSTBLK（初始化块）
80B1	没有块或者没有与此操作有关的块指定给参数 SKCBLK（初始化块）
80B4	具有 F 属性的数据块不能擦写
80C3	需要的操作资源正在被占用着
8xyy	一般故障的代码

### 3.11 用 SFC84 “WRIT\_DBL” 写数据到装载存储器中的数据块

#### 描述

通过 SFC84 “WRIT\_DBL”（在装载存储器中写数据块）可以将一个源数据块写到装载存储器（存储器卡）中数据类型为 BLOCK\_DB 的数据块中或者数据块的数据区中。装载存储器中参数 DSTBLK 所指的数据块可以与运行有关，也可以无关。参数 SRCBLK 所指的源区域可以是数据类型为 BLOCK\_DB 的数据块或者工作存储器中的数据块（内容）。参数 SRCBLK 所指的源数据块必须与运行有关（关键字 UNLINKED=0）。

源数据块也可以是通过 SFC22 “CREAT\_DB” 生成的。

为了保持一致，在 SF83 执行时一定要不要改变源区域。

下列限制条件适用于参数 SRCBLK 与 DSTBLK：

- 如果“ANY”数据类型为 BOOL，则长度必须可被 8 除
- 如果“ANY”数据类型为 STRING，则长度必须等于 1

如果需要，可以通过 SFC24 “TEST\_DB” 确定目标数据块的长度

SFC82 不改变用户程序的数据和。如果数据块是通过此 SFC 生成的，但是第一次进入此数据块写数据时就改变用户程序的数据和。

#### 注意

SFC84 异步执行，所以不适合频繁地写数据到装载存储器。请注意由于技术原因进入微存储器卡的写操作有一定的局限性。

#### 操作原理

SFC84 “CREA\_DBL” 异步运行，即处理过程包含此 SFC 的多次调用。

REQ=1 时调用 SFC84 功能启动。

作业状态可通过参数 RET\_VAL 与 BUSY 显示。

参见异步运行的 SFC 的参数 REQ, RET\_VAL 与 BUSY 的说明。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	REQ = 1: 写请求
SRCBLK	INPUT	ANY	D	源数据块数据区的指针
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障代码
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1 写操作还未结束
DSTBLK	OUTPUT	ANY	D	装载存储器中被写的数据块数据区的指针



## 故障信息

故障代码 (W#16#...)	说明
0000	无故障
0081	目标区域比源区域大 源区域中内容全部写到目标区域。目标区域中其它数据保持原状
7000	REQ=0 首次调用: 没有数据传输, BUSY 为 0
7001	REQ=1 首次调用: 没有数据传输, BUSY 为 1
7002	中间调用 (与 REQ 无关), 数据传输功能激活, BUSY 为 1
8081	源区域比目标区域大 目标区域充满, 源区域中其它数据被忽略
8092	Windows NT 操作系统瘫痪 (蓝屏)
8093	没有块或者没有与此操作有关的块指定给参数 SRCBLK (初始化块)
80B1	没有块或者没有与此操作有关的块指定给参数 DSTBLK (初始化块)
80B4	具有 F 属性的数据块不能擦写
80C0	目标区域正在被另外一个 SFC 或者通讯操作占用。
80C3	需要的操作资源正被占用着
8xyy	一般故障的代码



## 4 用于控制程序执行的 SFC

### 4.1 用 SFC43 “RE\_TRIGR” 再触发循环时间监控

#### 描述

用 SFC43 “RE\_TRIGR”（再触发看门狗）再触发循环时间监控。

#### 参数

SFC43 “RE\_TRIGR” 没有参数。

#### 故障信息

SFC43 “RE\_TRIGR” 没有提供任何故障信息。

### 4.2 用 SFC46 “STP” 使 CPU 进入停机状态

#### 描述

SFC46 “STP” 使 CPU 进入停机状态。

#### 参数

SFC46 “STP” 没有参数。

#### 故障信息

SFC46 “STP” 没有提供任何故障信息。

## 4.3 用 SFC 47 “WAIT” 延时用户程序执行

### 描述:

用 SFC 47 “WAIT” 在用户程序中设定延时或等待时间，最长可设定 32767 $\mu$ s。最短时间依 CPU 型号而定，并且与 SFC47 的执行时间相同。

### 中断能力

SFC 47 “WAIT” 可以被优先级更高的 OB 块中断。

### 注意

(S7-300, 不包括 CPU318)

SFC 47 所编写的等待时间是最小时间。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	存储单元	描述
WT	INPUT	INT	I, Q, M, D, L, 常数	参数 WT 包含等待时间, 单位为 $\mu$ s

### 故障信息

SFC47 “WAIT” 没有提供任何故障信息。

## 4.4 用 SFC35 “MP\_ALM” 触发多处理器中断

### 描述

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

用 JOB 输入参数指明多处理器中断的原因，工作标识符同时传送到所有的 CPU，再由多处理器中断程序识别（参考 STEP7 编程在线文本）。

可以在用户程序中任何地方调用 SFC35 “MP\_ALM”，但是只能在 RUN 状态才有效，如果在 STARTUP 状态调用，多处理器中断功能无效。

参数	声明	数据类型	存储单元	描述
JOB	INPUT	BYTE	I, Q, M, D, L, 常数	作业标识符 取值范围: 1-15
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能执行时如有错误 则这个参数显示故障代码

## 故障信息

故障代码 (W#16#...)	注释
0000	没有故障
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)

## 4.5 用 SFC104 “CiR” 控制 CiR

## 描述

RUN 状态下用 SFC104 “CiR” 执行再组态功能。

- PG 向 CPU 下载修改后的组态无效。这时只能用 SFC104 “CiR”，下载才会有效。
- 可以设定 CiR 同步到 CiR 禁止的时间上限值，如果 CPU 要求少于这个时间值，修改后的组态才能从 PG 下载到 CPU。
- 你可以决定是否使能 CiR 功能。如果使能或条件使能的情况下，参数 A\_FT 显示 CiR 的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	存储单元	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 (CiR 同步时间上限值已经预置)</li> <li>• 2: 完全禁止 CiR 功能</li> <li>• 3: 有条件禁止 CiR。参数 FRE_TIME 设定 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: ms 允许范围: 200-2500ms (预置: 1000ms) 注意: 只有 MODE=3 时 FRE_TIME 才有效

参数	声明	数据类型	存储单元	描述
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如运行时如果出错, 则显示故障代码 当 MODE=0, RET_VAT 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

#### 故障信息

故障代码 (W#16#...)	说明
0000	作业运行, 没有故障 (这个代码只有 MODE=1 或 2, 3 时有效)
0001	CiR 使能 (这个故障代码只有从 MODE=0 时有效)
0002	CiR 完全不使能 (这个故障代码只有从 MODE=0 时有效)
0003	CiR 条件使能 (这个故障代码只有从 MODE=0 时有效)
8001	CPU 没有准备 CiR 操作。应使用 H 系统在 H CPU (单一操作) 或在多 CPU 模式用标准 CPU。
8002	MODE 中的值非法
8003	FRZ_TIME 中的值非法

#### SFC104 应用实例

SFC104 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。因此在增加处理过程之前, 首先执行下列 CPU 程序段。

- 以 MODE=2 (CiR 不使能) 调用 SFC104

在增加处理过程之后执行下列 CPU 程序段:

- 以 MODE=1 (CiR 准备使能) 或 MODE=3 (CiR 条件不使能) 调用 SFC104

## 5 用于控制系统时钟的 SFC

### 5.1 用 SFC0 “SET\_CLK” 设定 TOD

#### 描述:

用 SFC0 “SET\_CLK”（设定系统时钟）设定 CPU 的时间和日期。执行 SFC0 启动时钟，时钟从设定时间和日期开始运行。如果这个时钟是主时钟，调用 SFC0 的同时，CPU 启动同步时间的功能，用 STEP 7 设定同步周期。

参数	声明	数据类型	存储单元	描述
PDT	INPUT	DT	D, L	输入设定的时间和日期
RET_VAL	OUTPUT	INT	I, Q, M, D, L	执行时如有故障，显示故障代码

#### 日期和时间

用数据类型 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 SFC0 “SET\_CLK” 从日期推算周的天数。可以用 FC3 “D\_TOD\_DT” 生成数据类型 DT。（见时间日期功能：FC1，FC3，FC6，FC7，FC8，FC33，FC34，FC35，FC40）。

#### 故障信息

出错代码 (W#16#...)	说明
0000	没有故障
8080	日期出错
8081	时间出错

### 5.2 用 SFC1 “READ\_CLK” 读取时间

#### 描述

用 SFC1 “READ\_CLK” 读取当前 CPU 系统时间和日期。

参数	声明	数据类型	存储单元	描述
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如果故障，则显示故障代码
CDT	OUTPUT	DT	D, L	CDT 输出当前时间和日期值

#### 故障信息

参见调用输出参数 RET\_VAL 评估故障的有关章节。

## 5.3 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（例如 S7-400、K 总线、MPI 或 S7 背板总线）传送到总线段的所有子时钟。

### 描述

SFC48 “SNC\_RTCB”（时钟同步）用于同步总线段上所有子时钟，并且与组态同步间隔无关。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。用 STEP7 设定相关参数。

参数	声明	数据类型	存储单元	描述
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如果出错，显示出错代码

### 故障信息

故障代码 (W#16#...)	
0000	同步期间没有故障
0001	对任何总线而言，现时钟都没有设成主时钟



## 5.4 用 SFC100 “SET\_CLKS” 设定日期时间和 TOD 状态

### 描述

用 SFC100 “SET\_CLKS” 设定 CPU 的时间日期和 TOD 状态

### 提示

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 参数指定是否改变时间日期, TOD 状态或二者都改变。下列表中有详细解释:

MODE (W#16#...)	注释
01	SFC 的调用相当于 SFC “SET_CLK” 的调用设定 TOD。 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>• 带符号的准确值</li> <li>• 时间制式</li> <li>• 显示夏季/冬季时间</li> <li>• CPU 决定 TOD 的分辨率 FALSE 显示 TOD 状态位同步失败。</li> </ul>
03	设定 TOD 和 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 的当前时间日期状态。

参数	声明	数据类型	存储单元	描述
MODE	INPUT	BYTE	I, Q, M, D, L, 常数	操作模式允许值:
				<u>B#16#01:</u> 设定 TOD
				<u>B#16#02:</u> 设定 TOD 状态
				<u>B#16#03:</u> 设定 TOD 和 TOD 状态
PDT	INPUT	DT	D, L	TOD 预置
CORR	INPUT	INT	I, Q, M, D, L, 常数	正确值 (0.5h 模式) 允许值范围: -24~+26
SUMMER	INPUT	BOOL	I, Q, M, D, L, 常数	夏季/冬季标识符 <ul style="list-style-type: none"> <li>• 0=冬季时间</li> <li>• 1=夏季时间</li> </ul>
ANN_1	INPUT	BOOL	I, Q, M, D, L,	时间制式

参数	声明	数据类型	存储单元	描述
			常数	1: 从下一个小时起夏季时间转换成冬季时间或反之
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障代码

故障信息

故障代码 (W#16#...)	说明
0000	没有故障
8080	MODE 超出允许值范围
8081	CORR 超出允许值范围 (仅 MODE=B#16#02 或 MODE=B#16#03 有效)
8082	PDT 超出允许值范围: 非法日期或时间日期

## 6 用于控制运行时间定时器的 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）

- SFC2、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

SFC2、3 和 4 也可用于 32 位运行时间定时器，这种情况下这个运行时间定时器只能运行在 16 位模式（取值范围：0—32767 小时）。也请参见用 SSL ID=W#16#0132 指针 W#16#000B 部分列表摘取的数据记录

---

#### 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性

启动的时候，运行时间定时器开始记录最后的数值，如果要求在不同的初值启动，这个值必须自己定义（SFC2 或 SFC101，MODE=4）。如果 CPU 进入 STOP 状态或你使运行时间定时器停止，CPU 记录运行时间定时器的当前值。当 CPU 执行暖启动或冷启动的时候，运行时间定时器重新启动（SFC3 或 SFC101，MODE=1）

#### 数值范围：

- 带有 16 位运行时间定时器的 CPU：0~32767 小时
- 带有 32 位运行时间定时器的 CPU：0~(2E31)-1 小时=2,147,483,647 小时

## 6.2 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）用于设置、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 CPU32 位运行时间定时器的值（参照数据记录表 SSL\_ID=W#16#0132 INDEX=W#16#000B）。

参数	声明	数据类型	存储单元	描述
NR	INPUT	BYTE	I, Q, M, D, L, 常数	运行时间定时器号码（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有溢出错误信息输出</li> <li>• 1: 启动（在最后的计数值）</li> <li>• 2: 停止</li> <li>• 3: 设置（在 PV 的设定值）</li> <li>• 4: 设置（在 PV 的设定值）并启动</li> <li>• 5: 设置（在 PV 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	该功能运行时如果出错，则返回值包含一个故障代码
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

SFC2 “SET\_RTM”、SFC3 “CTRL\_RTM” 和 SFC4 “READ\_RTM” 可用于操作 32 位运行时间定时器，但这时，32 位运行时间定时器只能当作 16 位使用（取值范围：0~32767 小时）。

形参 SSL ID=W#16#0132 和 INDEX=W#16#0008，在 16 位模式下显示 32 位运行时间定时器 0~7。你也可以用形参 SSL ID=W#16#0132，INDEX=W#16#0008 以 16 位的模式编写用户程序。

### 故障信息

故障代码（W#16#...）	说明
0000	没有故障
8080	运行时间定时器号码错误
8081	一个负值传送到 PV 参数
8082	运行时间定时器溢出
8091	输入参数 MODE 的值非法

## 6.3 用 SFC2 “SET\_RTM” 设置运行时间定时器

### 描述

用 SFC2 “SET\_RTM” 设定 CPU 运行时间定时器。

根据 CPU 选择运行时间定时器号码。

参数	声明	数据类型	存储单元	描述
NR	INPUT	BYTE	I, Q, M, D, L, 常数	输入运行时间定时器号码（允许值：0—7）
PV	INPUT	INT	I, Q, M, D, L, 常数	PV 参数用于运行时间定时器设定
RET_VAL	OUTPUT	INT	I, Q, M, D, L	运行时如果出错返回值显示出故障信息

### 故障信息

故障代码 (W#16#...)	说明
0000	没有故障
8080	运行时间定时器号码错误
8081	一个负值传送到 DV 参数

## 6.4 用 SFC3 “CTRL\_RTM” 启/停运行时间定时器

### 描述

SFC3 “CTRL\_RTM” 用于启动、停止 CPU 的运行时间定时器。

参数	声明	数据格式	存储单元	描述
NR	INPUT	BYTE	I, Q, M, D, L, 常数	被启动或停止的运行时间定时器的号码（允许范围：0-7）
S	INPUT	BOOL	I, Q, M, D, L, 常数	控制运行时间计时器的启动或停止。 0: 停止 1: 启动
RET_VAL	OUTPUT	INT	I, Q, M, D, L	运行中如有故障，显示故障代码。

### 故障信息

故障代码 (W#16#...)	说明
0000	没有故障
8080	运行时间定时器号码错误

## 6.5 用 SFC4 “READ\_RTM” 读取运行时间定时器

### 描述

SFC4 “READ\_RTM”（读运行时间定时器）读取运行时间定时器，SFC4 提供当前的运行时间和计数器状态，如“停止”或“计数”。

如果运行时间定时器运行超出 32767 小时，它会停在 32767 小时，并输出“溢出”的故障信息。

参数	声明	数据类型	存储单元	描述
NR	INPUT	BYTE	I, Q, M, D, L, 常数	被读取的运行时间定时器的号码 (取值范围: 0—7)
RET_VAL	OUTPUT	INT	I, Q, M, D, L	运行时如出错, 显示故障代码
CQ	OUTPUT	BOOL	I, Q, M, D, L	CQ 显示运行时间定时器是否运行。 0: 停止 1: 运行
CV	OUTPUT	INT	I, Q, M, D, L	CV 显示运行时间定时器当前值

### 故障信息

故障代码 (W#16#...)	说明
0000	没有故障
8080	运行时间定时器号码错误
8081	运行时间定时器溢出

## 6.6 用 SFC64 “TIME\_TCK” 读取系统时间

### 描述

SFC64 “TIME\_TCK”（报时信号）读取 CPU 的系统时间。系统时间是循环时间计数器，计数范围 0~2147483647ms。如果系统时间溢出，则计数从 0 开始。S7-400 和 CPU318 系统时间精度为 1ms。所有 S7-300CPU 系统时间精度为 10ms。只有 CPU 的操作模式才影响系统时间。

### 应用

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
运行	
停止	停止并保持当前值
热启动 (不包括 S7-300 和 S7-400H)	延续进入停止前保存的值
暖启动	删除并且启动时值为“0”
冷启动	

参数	声明	数据类型	存储单元	描述
RET_VAL	OUTPUT	TIME	I, Q, M, D, L	读取的系统时间 取值范围: $0-2^{31}-1$ ms

### 故障信息

参见 SFC 64 “TIME\_TCK” 的说明。





## 7 用于传送数据记录的 SFC

### 7.1 数据记录的写和读

#### 原理

某些模块具有只写的系统数据区域，可由程序向其传送的数据记录。该区域包含的数据记录编号从 0 开始最大为 240。并非每一模块都包含全部类型的数据记录（参见下表）。

另外，一些模块具有只读的系统数据区域，可通过程序从中读取数据记录。该区域包含的数据记录编号从 0 开始最大为 240。并非每一模块都包含全部类型的数据记录（参见下表）。

---

#### 注意

有些模块同时具备上述两种系统数据区域。在物理上这些区域是分开的，在逻辑结构上它们是相同的。

---

#### 只写的系统数据区域

下表显示了只写的系统数据区域的结构。该表还显示了数据记录的可能长度以及利用哪些 SFC 写这些数据记录。

数据记录号	内容	长度	限制	可用 SFC 写
0	参数	S7-300: 2 至 14 字节	只能由 S7-400 写	56 “WR_DPARAM” 57 “PARAM_MOD”
1	参数	S7-300: 2 至 14 字节 数据记录 0 和 1 的总和 正好是 16 字节	-	55 “WR_PARAM” 56 “WR_DPARAM” 57 “PARAM_MOD”
2 至 127	用户数据	每个 240 字节	-	55 “WR_PARAM” 56 “WR_DPARAM” 57 “PARAM_MOD” 58 “WR_REC”
128 至 240	参数	每个 240 字节	-	55 “WR_PARAM” 56 “WR_DPARAM” 57 “PARAM_MOD” 58 “WR_REC”

#### 只读的系统数据区域

下表显示了只读的系统数据区域的结构。该表还显示了数据记录的可能长度以及利用哪些 SFC 写这些数据记录。

数据记录号	内容	长度	可用 SFC 读
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 至 220 字节</li> </ul>	51 “RDSYSST” (SSL_ID 00B2H 和 00B3H) 59 “RD_REC”
2 至 127	用户数据	每个 ≤240 字节	59 “RD_REC”
128 至 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并且相互之间不发生冲突。

若所有可用的系统资源都被使用，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。详细信息参见/70/和/101/。

## 7.2 用 SFC 54 “RD\_DPARM” 读取定义的参数

### 描述

利用 SFC 54 “RD\_DPARM” (读定义的参数)，你可以从 STEP 7 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至由参数 RECORD 开辟的目标区域。

参数	声明	数据类型	存储区	说明
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) 若是混合型模块，指定最低地址的 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。对混合型模块，指定低端两个地址
RECNUM	INPUT	BYTE	I, Q, M, D, L, 常数	数据记录编号 (0 至 240)
RET_VAL	OUTPUT	INT	I, Q, M, D, L	若在功能激活时发生错误，返回值是故障代码。 若数据记录长度与目标区域匹配，且传送无错误，返回值是被读取的数据记录字节数。
RECORD	OUTPUT	ANY	I, Q, M, D, L	被读取数据记录的目标区域。只允许字节数据类型。

## 故障信息

用 SFC 57 “PARM\_MOD” 给模板分配参数。

## 7.3 用 SFC 102 “RD\_DPARA” 读取预定义的参数

## 说明

利用 SFC102 “RD\_DPARA”，你可以从 STEP7 组态的系统数据中读取选定模块的编号为 RECNUM 的数据记录。读取的数据记录被传送至由参数 RECORD 开辟的目标区域。

## 工作原理

SFC102 “RD\_DPARA” 是非同步工作的，即处理过程包括 SFC 的多重调用。调用 SFC102 时要使 REQ = 1 而启动任务。

任务状态通过输出参数 RET\_VAL 和 BUSY 显示。

可参阅非同步工作的 SFC 中关于 REQ, RET\_VAL 和 BUSY 的含义。

参数	声明	数据类型	存储区	说明
REQ	INPUT	BOOL	I, Q, M, D, L	REQ = 1: 读请求
LADDR	INPUT	WORD	I, Q, M, D, L, 常数	模块逻辑起始地址, 对混合模块组须指定低端的两个地址。
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	若在功能激活时发生错误, 返回值是故障代码。若数据长度与目标区域匹配且传送无错误, 返回值是被读取数据记录的字节数。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 任务还未结束
RECORD	OUTPUT	ANY	I, Q, M, D, L	被读取数据记录的目标区域。只允许字节数据类型。

## 故障信息

参见 SFC 57 “PARM\_MOD” 的说明。

## 7.4 用 SFC 55 “WR\_PARM” 写动态数据

### 说明

利用 SFC 55 “WR\_PARM”（写参数），你可以将数据记录 RECORD 传送至指定地址的模块。传送到模块去的参数不会由 STEP7 组态的参数所覆盖。

### 必要条件

被传送的数据记录不允许是静态的。（参阅/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	说明
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	地址区的 ID: B#16#54 = 外设输入 (PI) B#16#55 = 外设输出 (PQ) 若是混合型模块, 指定最低地址的 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。对混合型模块, 指定低端两个地址
RECNUM	INPUT	BYTE	I, Q, M, D, L, 常数	数据记录编号
RECORD	INPUT	ANY	I, Q, M, D, L	数据记录
RET_VAL	OUTPUT	INT	I, Q, M, D, L	若在功能激活时发生错误, 返回值是 故障代码。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 写未完成

### 输入参数 RECORD

被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次 SFC 调用的持续时间, 则参数 RECORD 的内容不再与后续的 SFC 调用相关 (对于相同的任务)。

### 故障信息

参见“用 SFC 57 “PARM\_MOD” 分配模板参数。

### 注

(只对 S7-400)

如果出现常规错误 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 存储区的访问被拒绝。数据传送继续进行。

## 7.5 用 SFC 56 “WR\_DPARM” 写缺省参数

### 说明

利用 SFC 56 “WR\_DPARM”（写缺省参数），你可以将编号为 RECNUM 的数据记录从 STEP 7 组态的数据中传送到指定地址的模块。使用本功能，数据记录是静态或动态都无关系。

参数	声明	数据类型	存储区	说明
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	地址区的 ID: B#16#54 = 外设输入 (PI) B#16#55 = 外设输出 (PQ) 若是混合型模块, 指定最低地址的 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。对混合型模块, 指定低端两个地址
RECNUM	INPUT	BYTE	I, Q, M, D, L, 常数	数据记录编号
RET_VAL	OUTPUT	INT	I, Q, M, D, L	若在功能激活时发生错误, 返回值包含故障代码。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 写还未完成

### 故障信息

参见“用 SFC 57 “PARM\_MOD” 分配模板参数。

## 7.6 用 SFC 57 “PARM\_MOD” 分配模块参数

### 说明

利用 SFC 57 “PARM\_MOD”（为模块指派参数），你可以把由 STEP7 组态的生成的某一模块的全部数据记录传送到该模块。使用本功能，数据记录是静态或动态都无关系。

参数	声明	数据类型	存储区	说明
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	地址区的 ID: B#16#54 = 外设输入 (PI) B#16#55 = 外设输出 (PQ) 若是混合型模块, 指定最低地址的 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。对混合型模块, 指定低端两个地址
RET_VAL	OUTPUT	INT	I, Q, M, D, L	若在功能激活时发生错误, 返回值是故障代码。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 写未完成

### 故障信息

“真实的”故障信息（故障代码 W#16#8xyz）可以分为两个级别：

- 暂时故障（故障代码 W#16#80A2 至 80A4, 80Cx）：  
这类故障有可能在不采取任何措施的情况下便被消除，换言之建议再次调用 SFC（若需要调用一次以上）。  
暂时故障的一个例子是需要的资源当前被占用（W#16#80C3）。
- 永久故障（故障代码 W#16#809x, 80A1, 80Bx, 80Dx）：  
这类故障若不采取措施便不会消除。只有消除错误，再次调用 SFC 才能成功。  
永久故障的一个例子是在 RECORD 输入错误的长度（W#16#80B1）。

### 注意

如果利用 SFC55、SFC56 或 SFC57 向 DPV1 从站传送数据记录且该从站运行于 DPV1 模式，DP 主站会按如下原则评估所收到的从站故障信息：

若故障信息代码处于 W#16#8000 至 W#16#80FF 或 W#16#F000 至 W#16#FFFF 区间则 DP 主站会将故障信息转移给 SFC。若故障信息代码不在上述区间，CPU 将数值 W#16#80A2 传递给 SFC 并把从站挂起。

针对 SFC 54 “RD\_DPARM”、SFC 55 “WR\_PARM”、SFC 56 “WR\_DPARM” 和 SFC 57 “PARM\_MOD” 的特定故障信息。

故障代码 (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=0: 无数据传送已开始; 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	指定的逻辑基本地址无效: 在 SDB1/SDB2x 中未分配或无基本地址。	-
8092	在 ANY 参考中指定的类型不是 BYTE。	仅适用于 S7-400 的 SFC 54 “RD_PARM” 和 SFC 55 “WR_PARM”
8093	本 SFC 不允许用于由 LADDR 和 IOID 所指定的模块(下列模块是允许的: S7-300 模块用于 7-300, S7-400 模块用于 S7-400, S7-DP 模块用于 S7-300 和 S7-400)。	-
80A1	当向模块送数据时, 消极的应答 (模块被摘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	通信总线上的通信问题。	故障发生在 CPU 与外部 DP 接口模块之间
80B0	SFC 不可能用于此类型模块, 模块不认可数据记录。	1)
80B1	传送的数据度不正确, SFC 54 “RD_PARM”: 由 RECORD 打开的目标区域太短。	-
80B2	组态的槽位没占用。	1)
80B3	实际模块的类型与 SDB1 中要求的类型不匹配。	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。	1)
80C2	模块正在处理对一个 CPU 所可能的最大数量的任务、	1)
80C3	需要的资源 (存储器, 等) 当前被占用	1)
80C4	内部暂时错误, 任务不能被处理。 • 重复任务。若本故障频繁出现, 要针对系统检查电干扰源。	1)
80C5	分布式 I/O 不存在。	分布式 I/O 1)
80C6	数据记录传送停止, 由于优先级的中止 (热启动或后台)	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 (数据记录编号=241 被 STEP 7 拒绝)	-
80D2	根据模块类型识别符, 该模块不能被赋予参数。	-
80D3	SDB 不能被访问因为其不存在。	-
80D4	SDB 结构错误: SDB 内部指针指向 SDB 以外的一个值。	仅对 S7-300
80D5	数据记录是静态的	仅对 SFC 55 “WR_PARM”

1) 在 SFC 54 “RD\_DPARM” 中不会发生。

## 7.7 用 SFC 58 “WR\_REC” 写数据记录

### 说明

利用 SFC 58 “WR\_REC”（写记录），你可将包含在 RECORD 中的数据记录传送到指定的地址的模块。

调用 SFC 58 时，给输入参数 REQ 赋值 1，则启动写任务，若写任务可以立即被执行，SFC 在输出参数 BUSY 中返回 0 值。若 BUSY 中值为 1，表明写任务尚未完成。

参数	声明	数据类型	存储区	说明
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	地址区的 ID: B#16#54 = 外设输入 (PI) B#16#55 = 外设输出 (PQ) 若是混合型模块，指定最低地址的 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。对混合型模块，指定低端两个地址
RECNUM	INPUT	BYTE	I, Q, M, D, L, 常数	数据记录号（允许值 0 至 240）
RECORD	INPUT	ANY	I, Q, M, D, L	数据记录。只允许 BYTE 数据类型。
RET_VAT	OUTPUT	INT	I, Q, M, D, L	若在功能激活时发生故障，返回值是故障代码。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 写尚为完成

### 输入参数 RECORD

被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次 SFC 调用的持续时间，则参数 RECORD 的内容不再与后续的 SFC 调用相关（对于相同的任务）。

### 故障信息

参见用 SFC 59 “RD\_REC” 读取数据纪录。

### 注意

（只对 S7-400）

如出现常规错误 W#16#8544，只是表明对于至少一个字节包含数据记录的 I/O 存储区的访问拒绝。数据传送继续进行。



## 7.8 用 SFC 59 “RD\_REC” 读数据记录

### 说明

利用 SFC 59 “RD\_REC”（读记录），你可以从指定地址的模块中读取编号为 RECNUM 的数据记录。读任务的启动通过调用 SFC59 且将输入参数 REQ 置为 1。若读任务可以马上被执行，SFC 在输出参数 BUSY 中返回 0 值。若 BUSY 中值为 1，表明读任务尚未完成（参见 0 章）。

若数据传送中未出错，读取的数据记录被传送至由参数 RECORD 指定的目标区域。

### 注

当你从 1997 年 2 月以前购买的 FM 和 CP（以下称“旧式模块”）中读取编号 > 1 的数据记录时，SFC59 的响应会有所不同。这种特殊情况在章节“使用旧的 S7-300 FM 和 CP 数据记录编号 > 1”中有说明。

参数	声明	数据类型	存储区	说明
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	地址区的 ID: B#16#54 = 外设输入 (PI) B#16#55 = 外设输出 (PQ) 若是混合型模块, 指定最低地址的 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。对混合型模块, 指定低端两个地址
RECNUM	INPUT	BYTE	I, Q, M, D, L, 常数	数据记录号 (允许值 0 至 240)
RECORD	OUTPUT	ANY	I, Q, M, D, L	数据记录。只允许 BYTE 数据类型。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	若在功能激活时发生故障, 返回值是故障代码。 若目标区长度大于被传送的数据记录长度且传送无错误, 返回值是被传送的数据记录实际字节数 (可能值: +1 至 +240_)
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 读尚为完成
RECORD	OUTPUT	ANY	I, Q, M, D, L	被读取数据的目标区, 当非同步执行 SFC 59 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, 只允许字节数据类型。

### 输出参数 RET\_VAL

- 若在功能执行时故障，返回值中包含故障代码。
- 若未发生错误，RET\_VAL 中内容如下：
  - 0: 整个目标区都被选定的数据记录所充填（数据记录也有可能不完整）。
  - 若目标区长度大于被传送数据记录的长度。RET\_VAL 为被传数据记录的实际字节数（可能值：+1 至+240）。

---

#### 注意

如果出现常规错误 W#16#8545，只是表明对于至少一个字节包含数据记录的 I/O 存储区的访问被封锁。数据记录已经正确地被模块读取且写入了 I/O 存储区。

---

### 设定 RECORD

---

#### 注意

如果你想确保每次都读取到完整的数据记录，可将目标区长度选公平为 241 字节。若数据传送无错误，则 RET\_VAL 中包含被传送数据记录的实际字节数。

---

使用旧的 S7-300 FM 和 CP 数据记录编号 > 1

如果你打算利用 SFC 59 “RD\_REC” 从旧的 S7-300 FM 和 S7-300 CP 中读取编号 > 1 的数据记录，需记住如下各点：

- 若目标区域的长度大于数据记录所需的实际长度，数据不传入 RECORD。  
RET\_VAL 的值为 W#16#80B1
- 若目标区域的长度小于数据记录所需的实际长度，CPU 将从数据记录的起始按 RECORD 的中长信息所定义的字节并把这些字节传送到 RECORD。  
RET\_VAL 的值为 0
- 若目标区域的长度等于数据记录所需的实际长度，CPU 数据记录并把传送到 RECORD。  
RET\_VAL 的值为 0

### 故障信息

“真实的”故障信息（故障代码 W#16#8xyz）可以分为两个级别：

- 暂时故障（故障代码 W#16#80A2 至 80A4, 80Cx）：  
这类故障有可能在不采取任何措施的情况下便被消除，换言之建议再次调用 SFC（若需要调用一次以上）。  
暂时故障的一个例子是需要的资源当前被占用（W#16#80C3）。
- 恒定故障（故障代码 W#16#809x, 80A1, 80Bx, 80Dx）：  
这类故障若不采取措施便不会消除。只有消除故障，再次调用 SFC 才能成功。  
恒定故障的一个例子是在 RECORD 输入故障的长度（W#16#80B1）。

---

#### 注

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 从站读取数据记录且该从站运行 DPV1 模式，DP 主站会按如下原则评估所收到的来自从站故障信息：

若故障信息代码处于 W#16#8000 至 W#16#80FF 或 W#16#F000 至 W#16#FFFF 区间则 DP 主站会将故障移给 SFC。若故障信息代码不在上述区间，CPU 将数值 W#16#80A2 传递补给 SFC 并把从站挂起。

对于来自 DPV1 从站故障信息的说明，参见利用 SFB 54 “RALRM” STATUS[3]接收来自 DP 从站的中断。

针对 SFC 58 “WR\_REC” SFC 59 “RD\_REC” 的特定出错信息。

故障代码 (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=0: 无数据传送已激活; 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	指定的逻辑基本地址无效: 在 SDB1/SDB2x 中未分配或无基本地址	-
8092	在 ANY 参考中指定的类型不是 BYTE。	仅适用于 S7-400
8093	本 SFC 不允许用于由 LADDR 和 IOID 所指定的模块 (下列模块是允许的: S7-300 模块用于 7-300, S7-400 模块用于 S7-400, S7-DP 模块用于 S7-300 和 S7-400)。	-
80A0	当从模块读数据时, 消极的应答 <ul style="list-style-type: none"> <li>模块在读任务期间被摘除或损坏</li> <li>对 H-系统: 备用 CPU 一侧的 I/O 不可用 (例如, CPU 在 STOP 状态)</li> </ul>	仅适用于 SFC 59 “RD_REC”
80A1	当从模块读数据时, 消极的应答 <ul style="list-style-type: none"> <li>模块在读任务期间被摘除或损坏</li> <li>对 H-系统: 备用 CPU 一侧的 I/O 不可用 (例如, CPU 在 STOP 状态)</li> </ul>	仅适用于 SFC 58 “WR_REC”
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	通信总线上的通信问题。	错误发生在 CPU 与外部 DP 接口模块之间
80B0	SFC 不可能用于此类型模块, 模块不认可数据记录。 数据记录编号=241 不允许。 对 SFC 58 “WR_REC” 数据记录编号 0 和 1 不允许。	-
80B1	由参数 RECORD 定义的长度不正确	<ul style="list-style-type: none"> <li>SFC 58 “WR_REC”: 长度不正确</li> <li>SFC 59 “RD_REC” (仅在使用旧版 S7-300 FM 和 S7-300 CP 时): 指定长度 &gt; 记录长度</li> <li>SFC 13</li> </ul>
80B2	组态的槽位没占用。	-
80B3	实际模块类型与在 SDB1 中要求的模块类型不匹配。	-

故障代码 (W#16#...)	说明	限制
80C0	对 SFC 59 “RD_REC”：模块有数据记录但无数据被读 对 SFC 13 “DPNRM_DG”：诊断数据不存在	仅适用于 SFC 59 “RD_REC”或 SFC 13 “DPNRM_DG”
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。	-
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（存储器，等）当前被占用	-
80C4	内部暂时错误，任务不能被处理。 • 重复任务。若本错误频繁出现，要针对系统检查电干扰源。	-
80C5	分布式 I/O 不存在。	分布式 I/O 1)
80C6	数据记录传送停止，由于优先级的中止（热启动或后台）	分布式 I/O 1)

## 7.9 关于 SFC 55 至 59 进一步的故障信息

仅对于 S7-400

使用 S7-400 时 SFC 55 至 59 也可返回故障信息 W#16#80Fx。在此情况下一个故障的发生可以是局部化的。在此时请与维护部门联系。

## 8 符合 PNO AK1131 的 DPV1 SFB

### 8.1 用 SFB 52 “RDREC” 读来自 DP 从站的数据记录

---

#### 注意

SFB52 “RDREC” 的接口与 FB “RDREC” 是一致的，在标准“PROFIBUS 指导方针-依照 IEC 61131-3 的 PROFIBUS 通信和代理服务器功能模块”中作了定义。

---

#### 说明

利用 SFB52 “RDREC”（读记录）你可以从由参数 ID 指定地址的 DP 从站部件（模块或子模块）读取编号为 INDEX 的一个数据记录。

要读取的最大字节数在参数 MLEN 中指定。目标区域 RECORD 应至少与 MLEN 等长。输出参数 VALID 中得到“真”，证明数据记录已成功传送到目标区域 RECORD 中，此时，输出参数 LEN 的内容是被读取数据的字节数。

输出参数 ERROR 指出在数据记录传送过程中是否出现错误。在出错的情况下，输出参数 STATUS 中包含了故障信息。

#### 工作原理

SFB52 “RDREC” 是非同步工作的。即，处理多次 SFB 调用。任务的启动通过调用 SFB52 且置 REQ=1 而实现。

任务的状态通过输出参数 BUSY 和输出参数 STATUS 的第 2 和第 3 位显示出来。这里，参数 STATUS 的第 2 和第 3 位对应与非同步工作的 SFC 的输出参数 RET\_VAL（参见非同步工作的 SFC 的参数 REQ, RET\_VAL 和 BUSY 的含义）。

当输出参数 BUSY=“假”时，数据记录传送完成。

参数	声明	数据类型	存储区	说明
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	REQ = 1: 传送数据
ID	INPUT	DWORD	I, Q, M, D, L, 常数	DP 从站部件（模块或子模块）的逻辑地址。 对于输出模块量，必须设置高位的值。 对于混合模块应指定较小的两个地址。
INDEX	INPUT	INT	I, Q, M, D, L, 常数	数据记录编号
MLEN	INPUT	INT	I, Q, M, D, L, 常数	要读取数据记录的最大字节数
VALID	OUTPUT	BOOL	I, Q, M, D, L	新数据记录接收到且有效
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 读尚未完成
ERROR	OUTPUT	BOOL	I, Q, M, D, L	ERROR = 1: 读出错。

参数	声明	数据类型	存储区	说明
STATUS	OUTPUT	DWORD	I, Q, M, D, L	调用的 ID (位 2 和 3) 或故障代码
LEN	OUTPUT	INT	I, Q, M, D, L	被读取数据记录的字节数。
RECORD	IN_OUT	ANY	I, Q, M, D, L	被读取数据记录的目标区域。

#### 出错信息

参见利用 SFB54 “RALRM” 接收来自 DP 从站的中断。

## 8.2 用 SFB53 “WRREC” 向 DP 从站写数据记录

#### 注意

SFB53 “WRREC” 的接口与 FB “WRREC” 是一致的，在标准 “PROFIBUS 指导方针-依照 IEC 61131-3 的 PROFIBUS 通信和代理服务器功能模块” 中作了定义。

#### 说明

利用 SFB53 “WRREC” (写记录) 你可以从由参数 ID 指定地址的 DP 从站部件 (模块或子模块) 读取编号为 INDEX 的一个数据记录。

编号为 INDEX 的数据记录。

要指定被传送数据记录的字节数。源区域 RECORD 应至少与 LEN 等长。

输出参数 DONE 中得到 “真”，证明数据记录中成功地传送到 DP 从站中。

输出参数 ERROR 指出在数据记录传送过程中是否出现错误。在出错的情况下，输出参数 STATUS 中包含了出错信息。

#### 工作原理

SFB53 “WRREC” 是非同步工作的。即，处理多次 SFB 调用。任务的启动通过调用 SFB53 且置 REQ=1 而实现。

任务的状态通过输出参数 BUSY 和输出参数 STATUS 的第 2 和第 3 位显示出来。这里，参数 STATUS 的第 2 和第 3 位对应与非同步工作的 SFC 的输出参数 RET\_VAL (参见非同步工作的 SFC 的参数 REQ, RET\_VAL 和 BUSY 的含义)。

请注意，对属于同一任务的 SPB53 的所有调用，必须给参数 RECORD 赋予相同的值，此规定同样适用于参数 LEN。

当输出参数 BUSY= “假” 时，数据记录传送完成。

参数	声明	数据类型	存储区	说明
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	REQ = 1: 传送数据
ID	INPUT	DWORD	I, Q, M, D, L, 常数	DP 从站部件（模块或子模块）的逻辑地址。 对于输出模块量高位的值必须设置。 对于混合模块应指定较小的两个地址。
INDEX	INPUT	INT	I, Q, M, D, L, 常数	数据记录编号
LEN	INPUT	INT	I, Q, M, D, L, 常数	要读取数据记录的最大字节数
DONE	OUTPUT	BOOL	I, Q, M, D, L	数据记录传送完毕
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 写过程尚未完成
ERROR	OUTPUT	BOOL	I, Q, M, D, L	ERROR = 1: 写出错。
STATUS	OUTPUT	DWORD	I, Q, M, D, L	调用的 ID（位 2 和 3）或故障代码
RECORD	IN_OUT	ANY	I, Q, M, D, L	数据记录

### 8.3 用 SFB 54 “RALRM” 接收来自 DP 从站的中断

#### 注意

SFB 54 “RALRM” 的接口与 FB “RALRM” 是一致的，在标准“PROFIBUS 指导方针-依照 IEC 61131-3 的 PROFIBUS 通信和代理服务器功能模块”中作了定义。

#### 说明

SFB “RALRM” 接收来自外设模块（集中结构）或 DP 从站部件的中断及其全部相关信息。它把这些信息提供给它的输出参数。

输出参数中的信息包括被调用 OB 的启动信息以及中断源的信息。

SFB54 的调用只是作为要被检查的外设中断的结果这一过程由 CPU 操作系统启动中断 OB 中进行。

#### 注意

如果你在一个 OB 中调用 SFB 54 “RALRM”，而此 OB 的启动事件不是由外设触发的，则 SFB 会在它的输出内提供相应简化的信息。

确保在不同 OB 中调用 SFB 54 “RALRM” 时使用不同的背景 DB。若打算评价在相关中断 OB 以外 SFB54 的调用结果，你应该进一步注意每次 OB 启动事件，要使用单独的背景 DB。

## SFB 54 的调用

你可以在 3 种运行模式（MODE）下调用 SFB 54 “RALRM”。说明见下表。

模式	SFB54 ...
0	... 在输出参数 ID 中指出 触发中断的部件并把输出参数 NEW 置为“真”
1	... 记述所有的输出参数而不受中断触发部件的约束
2	... 检查在输入参数 F_ID 中指定的部件是否已触发中断。 <ul style="list-style-type: none"> <li>• 若否, NEW = 假</li> <li>• 若是, NEW = 真</li> </ul>

参数	声明	数据类型	存储区	说明
MODE	INPUT	INT	I, Q, M, D, L, 常数	运行模式
F_ID	INPUT	DWORD	I, Q, M, D, L, 常数	准备接收产生中断的部件（模块）的逻辑起始地址
MLEN	INPUT	INT	I, Q, M, D, L, 常数	准备接收中断信息数据最大字节数
NEW	OUTPUT	BOOL	I, Q, M, D, L	一个新的中断被接收
STATUS	OUTPUT	DWORD	I, Q, M, D, L	SFB 或 DP 主站的故障代码
ID	OUTPUT	DWORD	I, Q, M, D, L	接收到的中断产生部件（模块）的逻辑起始地址。最高位的值包含 I/O 的识别标志： 0 表示输入地址 1 表示输出地址
LEN	OUTPUT	INT	I, Q, M, D, L	收到的中断信息长度
TINFO	IN_OUT	ANY	I, Q, M, D, L	（任务信息） 用于 OB 启动和管理的目标范围
AINFO	IN_OUT	ANY	I, Q, M, D, L	（中断信息） 用于标题信息和附加中断信息的目标区域 AINFO 应至少与 MLEN 等长度。

## 注意

若选择的目标区域 TINFO 或 AINFO 太短，SFB54 就不能输入完整的信息。

## 目标区域 TINFO 的数据结构

字节	含义
0 - 19	调用 SFB54 的 OB 的起始信息
20 - 27	管理信息



## 管理信息的数据结构

用于 TINFO 的字节号	数据类型	含义				
20	BYTE	集中的:	0			
		分布的:	DP 主站的系统 ID (可能值 1-255)			
21	BYTE	集中的:	模块机架号 (可能值 0 - 31)			
		分布的:	DP 从站号 (可能值 0-27)			
22	BYTE	集中的:	0			
		分布的:	• 位 0 至 3:	从站类型	0000: 0001: 0010: 0011: 0100: 之后	DP DPS7 DPS7 V1 DPV1 保留
			• 位 4 至 7:	特征类型	0000: 0001 之后:	DP 保留
23	BYTE	集中的:	0			
		分布的:	• 位 0 至 3:	中断信息类型	0000:	透明的 (中断源来自一个组态的分布式模块)
					0001:	典型的 (中断源来自一个非 DPV1 从站或一个未组态的槽位)
					0010:	生成的 (中断在 CPU 中生成)
					as of 0011:	保留
			• 位 4 至 7:	结构版本	0000: as of 0001:	初始的 保留
24	BYTE	集中的:	0			
		分布的:	DP 主站的标志			
			• 位 = 0:	中断源来自一个集成的 DP 接口		
			• 位 0 = 1:	中断源来自一个外部的 DP 接口		

用于 TINFO 的字节号	数据类型	含义	
			<ul style="list-style-type: none"> <li>• 位 1-7: 保留</li> </ul>
25	BYTE	集中的:	0
		分布的:	DP 从站的标志
			<ul style="list-style-type: none"> <li>• 位 = 0: 诊断信息构架的 EXT_DIAG 位, 或若此位在中断不存在则为 0</li> </ul>
			<ul style="list-style-type: none"> <li>• 位 1-7: 保留</li> </ul>
26 to 27	WORD	集中的:	0
		分布的:	PROFIBUS 的 ID 号

## 目标区域 AINFO 的数据结构

字节	含义	
0 to 3	部首信息	
4 to 223	附加的中断信息: 对于不同中断的模块特定数据	
	• 集中的:	矩阵[0]—矩阵[220]
	• 分布的:	矩阵[0]—矩阵[59]

## 部首信息的结构

字节	数据类型	含义	
0	BYTE	不同中断信息的字节数	
		• 局部的:	1 - 224
		• 分布的	4 - 63
1	BYTE	局部的:	保留
		分布的	针对中断类型的 ID
		1:	诊断中断
		2:	硬件中断
		3:	除去的中断
		4:	插入的中断
		5:	状态中断
		6:	刷新的中断
		31:	扩展装置, DP 主系统或 DP 站故障
		32 -126	制造商特定中断

字节	数据类型	含义	
2	BYTE	中断触发部件的槽位号	
3	BYTE	局部的:	保留
		分布的:	标识
		位 0 和 1:	0: 无进一步的信息 1: 到来的事件, 槽位已混乱 2: 离去的事件, 槽位不再混乱 3: 离去的事件, 槽位仍混乱
		位 2:	Add_Ack
		位 3-7:	顺序号

### 目标区域 TINFO 和 AINFO

根据调用 SFB54 和 OB 的不同, 目标区域 TINFO 和 AINFO 只是部分的被写入信息。参考下表可知有关的不同目标区域被写的情况。

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	有	有	有	集中的:	无
					分布的:	由 DP 从站提供
状态中断	55	有	有	有	有	
刷新的中断	56	有	有	有	有	
制造间特定中断	57	有	有	有	有	
外设冗余错误	70	有	有	无	无	
诊断中断	82	有	有	有	集中的:	数据记录 1
					分布的:	由 DP 从站提供
除去的/插入的中断	83	有	有	有	集中的:	无
					分布的:	由 DP 从站提供
模块机架/站故障	86	有	有	无	无	
...	所有其它 OB	有	无	无	无	

### 故障信息

输出参数 STATUS 包含有信息。它被解译为 ARRAY [1..4] OF BYTE 的格式。故障信息结构如下:

域元素	名称	含义
STATUS[1]	功能号	<ul style="list-style-type: none"> <li>若无错, B#16#00</li> <li>来自 DPV1-PDU 的功能 ID: 在出错时为 B#16#80。若没有使用 DPV1 协议元素则为 B#16#C0</li> </ul>
STATUS[2]	故障解码	故障 ID 的位置
STATUS[3]	故障代码 1	故障 ID
STATUS[4]	故障代码 2	制造商特定故障 ID 解释

STATUS[2] 可能有如下值:

故障解码(B#16#...)	源	含义
00 - 7F	CPU	无故障或无告警
80	DPV1	根据 IEC 61158-6 的故障
81 - 8F	CPU	B#16#8x 表示在第 n 次调用 SFB 参数中的一个故障
FE, FF	DP 特征数据区	特征数据区专有故障

STATUS[3]可能有如下值:

故障解码 (B#16#...)	故障代码 1 (B#16#...)	根据 DVP1 的解释	含义
00	00		无故障, 无告警
70	00	保留, 拒绝	初始调用; 未激活数据记录传送
	01	保留, 拒绝	初始调用; 数据记录传送已经开始
	02	保留, 拒绝	中间调用; 数据记录传送已激活
80	90	保留, 通过	无效逻辑起始地址
	92	保留, 通过	对 ANY 指针的非法类型
	93	保留, 通过	由 ID 或 F_ID 指定的 DP 部件没有组态
	95		当获取附加中断信息时 H 系统中的故障 (当从通过外部 DP 接口的局部或分布的 I/O 获取附加中断信息时, 该故障作为“组故障”输出)。注: 当连接或刷新时, 附加中断信息可能暂时不可用。
	96		H 系统中发生了主保留切换, 且先前的主 CPU 已处于 STOP 模式。在那时刻一个 OB 被处理。SFB54 无法提供该 OB 的起始信息, 管理信息, 部首信息或附加中断信息。 你可利用 SFC 6 “RD_SINFO” 读取 OB 的起始信息。另外, 对于 OB 4x, 55, 56, 57, 82 和 83 你还可利用 SFC 13 “DPNRM_DG” 同步地读取受影响 DP 从站的当前诊断结构 (地址信息来自

故障解码 (B#16#...)	故障代码 1 (B#16#...)	根据 DVP1 的解 释	含义
			OB 的起始信息)。
	A0	读故障	读模块时的否定应答
	A1	写故障	写模块时的否定应答
	A2	模块故障	第 2 层 DP 协议出错, 可能硬件故障
	A3	保留, 通过	带直接数据连接图样器或用户接口/用户的 DP 协议出错, 可能硬件故障
	A4	保留, 通过	总线通信混乱
	A5	保留, 通过	-
	A7	保留, 通过	DP 从站或模块被占用 (暂时错误)
	A8	版本冲突	DP 从站或模块报告版本不兼容
	A9	特征不被支持	特征不被 DP 从站或模板所支持
	AA to AF	用户特定	DP 从站或模块报告在其应用中的一个制造商特定错误。请检查来自制造商的 DP 从站或模块的文献
	B0	无效指针	数据记录不可知, 模块中非法数据记录, 编号 $\geq$ 256
	B1	写长度错误	在参数 RECORD 中指定的错误长度; SFB54: 长度错误在 AINFO 中
	B2	无效槽位	组态的槽位未占用
	B3	类型冲突	实际模块类型与指定模块类型不相等
	B4	无效区域	DP 从站或模块报告访问一个无效的区域
	B5	状态冲突	DP 从站或模块未准备好
	B6	访问被拒绝	DP 从站或模块拒绝访问
	B7	无效范围	DP 从站或模块报告对于参数或数值的一个无效范围
	B8	无效参数	DP 从站或模块报告一个无效参数
	B9	无效类型	DP 从站或模块报告一个无效类型
	BA to BF	用户特定	在访问时 DP 从站或模块报告一个制造商特定错误。请检查来自制造商的 DP 从站或模块的文献
	C0	读抑制冲突	模块有数据记录, 但还没有读数据
	C1	写抑制冲突	先前对该模块的确同一数据记录的确定请求还没有被模块处理。
	C2	源忙	对某一 CPU 而言模块正处理最大可能的任务。
	C3	源不可用	需要处理的源当前被占用。
	C4		内部暂时错位。任务不能执行。请重复任务。若此错误频繁发生, 对现场进行电气干扰检查。
	C5		DP 从站或模块不可用
	C6		由于优先级的取消使数据记录的传送被取消。

故障解码 (B#16#...)	故障代码 1 (B#16#...)	根据 DVP1 的解 释	含义
	C7		由于 DP 主站的暖启动或冷启动使任务被取消。
	C8 to CF		DP 从站或模块报告一个制造商特定故障。请检查来自制造商的 DP 从站或模块的文献
	Dx	用户特定	DP 从站专用，参考 DP 从站的说明
81	00 to FF		第 1 个调用参数出错 (SFB54: MODE)
	00		非法运行模式
82	00 to FF		第 2 个调用参数出错
:	:		:
88	00 to FF		第 8 个调用参数出错 (SFB54: TINFO)
	01		错误语法 ID
	23		数量构架超出或目标区域太小
	24		错误范围 ID
	32		DB/DI 号超出用户范围
	3A		对区域 ID DB/DI, DB/DI 号为 NULL 或指定的 DB/DI 不存在。
89	00 to FF		第 9 个调用参数出错 (SFB54: AINFO)
	01		错误语法 ID
	23		数量构架超出或目标区域太小
	24		错误范围 ID
	32		DB/DI 号超出用户范围
	3A		对区域 ID DB/DI, DB/DI 号为 NULL 或指定的 DB/DI 不存在。
8A	00 - FF		第 10 个调用参数出错
:	:		:
8F	00 - FF		第 15 个调用参数出错
FE, FF	00 - FF		特征专用故障

有 DPV1 故障时，DP 主站将 STATUS[4]转送给 CPU 和 SFB。无 DPV1 故障时，该值被置 0，对于 SFB52 有如下例外：

- 若  $MLEN > RECORD$  指定的目标区域长度，STATUS[4]内容为 RECORD 指定的目标区域长度
- 若实际数据记录长度  $< MLEN < RECORD$  指定的目标区域长度，STATUS[4]=MLEN

## 8.4 用 SFB 75 “SALRM” 向 DP 从站发送中断

### 注意

SFB57 “SALRM” 的接口与 FB “SALRM” 是一致的，在标准“PROFIBUS 指导方针-依照 IEC 61131-3 的 PROFIBUS 通信和代理服务器功能模块”中作了定义。

### 说明

利用 SFB75 “SALRM” 你可以将一个智能从站用户程序的一个槽位中断传送给相关的 DP 主站，该主站在转换区访问这个中断（虚拟槽位）。在 DP 主站上相关的 OB 会启动。

你可以把附加的中断特定信息与中断一起发送。在 DP 主站上，可利用 SFB54 “RALARM” 读取全部附加信息。

SFB 75 只被用于 S7 兼容模式。

- DP：主站用 GSD 文件以结合 I-从站
- S7 兼容模式：I-从站通过 STEP7 HW Config 连接 STEP 7 中的主站。

### 工作原理

SFB75 “SALRM” 是非同步工作的 SFB。即，处理扩展至多次 SFB 调用。中断传送通过调用 SFB75 且置 REQ=1 而实现。

传送过程一直维护有效，直到中断处理被 DP 从站应答或取消。

任务的状态通过输出参数 BUSY 和输出参数 STATUS 的第 2 和第 3 位显示出来。这里，参数 STATUS 的第 2 和第 3 位对应于非同步工作的 SFC 的输出参数 RET\_VAL（参见非同步工作的 SFC 的参数 REQ、RET\_VAL 和 BUSY 的含义）。

当输出参数 BUSY=“假”时，数据记录传送完成。

### 任务识别

当你初始调用 SFB 75 向 DP 主站传送一个中断，而在当前任务完成之前再次调用该 SFB 时，SFB 的反应将取决于新的调用与前一次调用是否一致。

如果参数 ID 和 ATYPE 与未完成的任务中一致，新的 SFB 调用将接续处理。

参数	声明	数据类型	存储区	说明
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	以 DP 从站角度看的 DP 主站转换区（虚拟槽位）的逻辑起始地址。 相关的信息在低字部分，在高字部分输入 0。 第 15 位包含 I/O 的识别标志： 0=输入地址， 1=输出地址。 注意：可以使用槽口 2 的诊断地址和逻辑地址。

参数	声明	数据类型	存储区	说明
ATYPE	INPUT	INT	I, Q, M, D, L, 常数	中断类型 用于中断类型的 ID, 可能值: • 1: 诊断中断 • 2: 硬件中断
ASPEC	INPUT	INT	I, Q, M, D, L, 常数	中断识别符 • 0: 无进一步的信息 • 1: 到来的事件, 槽位有错 • 2: 离去的事件, 槽位不再有错 • 3: 离去的事件, 槽位依然有错
LEN	INPUT	INT	I, Q, M, D, L, 常数	准备发送的附加中断信息长度 (字节数)。 最大值: 16
DONE	OUTPUT	BOOL	I, Q, M, D, L,	DONE=1: 中断被传送完
BUSY	OUTPUT	BOOL	I, Q, M, D, L,	BUSY = 1: 中断被传仍忙。
ERROR	OUTPUT	BOOL	I, Q, M, D, L,	ERROR = 1: 有一个故障发生
STATUS	OUTPUT	DWORD	I, Q, M, D, L,	故障信息
AINFO	IN_OUT	ANY	I, Q, M, D, L,	中断信息 附加中断信息的源区域

### 参数 ATYPE

下表指出了所有 ATYPE 的允许值, 在相关的主站中哪个 OB 要被启动以及在何种 DP 模式下给定的中断被允许。

ATYPE	在 DPV1 标准中含义	在 S7 主站中相关的 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 - 47)	-	是

### 注意

除了上表给出的信息之外, 中断类型所适应的范围也可被 DP 主站限制。

### 涉及 S7 主站的运行模式的中断类型的一致性

对处于 S7-兼容模式的从站 (被 S7 主站操作), 硬件中断和诊断中断只能在相关的 DP 主站处于 RUN 模式时 (DP: 运行) 才能被发送。若 DP 主站处于 STOP 模式 (DP: 清除), 则中断被抑制, 而且 SFB75 会返回故障信息 W#16#80C8。在这种情况下, 用户自己要负责再次发送中断。

### 参数 ASPEC

根据可应用的标准, 本参数指明了虚拟槽位的诊断状态。因此, 只有当发送诊断中断时你才可除 0 以外的数值赋予 ASPEC。



由于用于 S7 诊断中断的附加中断信息（数据记录 0）包括到来/离去的信息（见诊断数据 0 字节 0 位），你必须按下表来描述附加中断信息的 0 字节第 0 位（模块故障）：

ASPEC	在 AINFO 的“模块故障”位
0	0
1	1
2	0
3	1

### 参数 LEN

参数 LEN 指定要被发送的附加中断信息的字节数，数值的最大允许范围是 0 至 16。  
对所有可能的中断类型，下表指明了对于智能从站的不同模式 LEN 可能取的值。

中断类型	DP	S7-兼容
诊断中断	-	4 - 16
硬件中断	-	4

下表指明了当你赋给 LEN 的值在字节类型上与 AINFO 不同时 SFB75 的反应

LEN 的值	SFB 75 的反应
<=为 AINFO 指定的长度	SFB 75 发送一个中断到 DP 主站。被发送的附加中断信息的字节数按 LEN 指定
超出允许范围(< 0 或 > 16)	SFB 75 不断发送中断。 故障信息：W#16#80B1，STATUS[4]=B#16#FF
> 为 AINFO 指定的长度	SFB 75 发送一个中断到 DP 主站。被发送的附加中断信息的字节数取决于 AINFO 中的长度信息。 故障信息：W#16#00B1，STATUS[4]=AINFO 中的长度信息

### 参数 AINFO

AINFO 是附加中断信息的源区域，对于智能从站，你可向此区域充填任意你希望的值。但如果你正使用 S7 系列中的 DP 主站，则与中断一起发送的附加信息必须符合 S7 的协定。如果你发送一个诊断中断（ASPEC=1），你就要负责在数据记录 0 必要时也在数据记录 1 中输入符合要求的相关数值。

下表为你指出了对于 S7-兼容建议的设定。在此情况下，“模块故障”位（见上文）已经被设定。对于实际命名的位，本建议对应于默认设定（即接通电源，智能从站从 STOP 到 RUN 转换之后或站返回所存在的值）。

数据记录号...	设定
0	B#16#01, 0B, 00, 00
1	对于 S7-兼容模式：数据记录 0 + 12 个带 0 的字节

（可参考诊断数据的结构概览）

### SFB75 调用对模块状态信息和组故障 LED (SF) 的影响

对于任意的 CPU，一个智能从站的槽位属性储存在模块状态信息中（参见 SSL-ID W#16#xy91 – 模块状态信息）。

当你利用 SFB75 发送一个诊断中断，智能从站的操作系统会影响从站的局部模块状态信息和基于 SINFO 中 0 号字节第 0 位的 SF LED（这一位被加入到状态信息作为“模块故障”），但是，在智能从站上不生成诊断缓冲区条目，诊断中断 OB 也不启动。

### DP 主站与智能从站之间模块状态信息的一致性

下文介绍了不同的情况以及它们对模块状态信息的影响：

- 站返回（导致在 DP 主站和智能从站上 OB86 启动）

这对智能从站和 S7 主站的模块状态信息有影响（“模块故障”被复位）。若从用户的角度看在站返回后在智能从站上有故障，这就必须通过 SFB75 的调用向 DP 主站报告。

- DP 主站上 STOP-RUN 运行模式转换（导致在智能从站上 OB82 启动）

智能从站的模块状态信息维持不变。DP 主站复位相关模块状态信息中的“模块故障”位。

为了保持 DP 主站与智能从站之间模块状态信息在 S7-兼容模式的一致性，你必须在智能从站上采取如下行动：

- 对于每个无错的虚拟槽位，用 SFB75 向 DP 主站发送一个离去的诊断中断。
- 对于每个出错的虚拟槽位，用 SFB75 向 DP 主站发送一个到来的诊断中断。

- 智能从站上 STOP-RUN 运行模式转换（导致在 DP 主站上 OB82 启动）

DP 主站的模块状态信息维持不变。智能从站上模块状态信息中的“模块故障”位被复位。

为了保持 DP 主站与智能从站之间模块状态信息在 S7-兼容模式的一致性，你必须在智能从站上采取如下行动：

- 对于每个无错的虚拟槽位，用 SFB75 向 DP 主站发送一个离去的诊断中断。
- 对于每个出错的虚拟槽位，用 SFB75 向 DP 主站发送一个到来的诊断中断。

---

#### 注意

由于 SFB75 是非同步运行的，所以 SFB75 的调用不可能在启动 OB 中结束。换言之，该 SFB 只能在循环程序中运行直至完成。

---

---

#### 注意

所有上述提及的主站与智能从站之间在模块状态信息上的区别，仅会发生在那些通过 SFB75 接收诊断中断的槽位。也就是说，上文关于补救行动的讨论仅适用与这些槽位。

---

### 出错信息

输出参数 STATUS 包含有信息。它被解译为 ARRAY [1...4] OF BYTE 的格式。故障信息结构如下：

域元素	含义
STATUS[1]	<ul style="list-style-type: none"> <li>• B#16#00: 无错</li> <li>• B#16#C0: 错误被智能从站检测到</li> </ul>
STATUS[2], STATUS[3]	与 SFC 中的输出参数 RET_VAL 对应
STATUS[4]	B#16#00, 当 LEN 和 AINFO 之间有长度冲突时例外。这些例外在下一个表中有说明

下表指出了对于 SFB75 的全部故障信息。

故障	STATUS[2, 3] (W#16# ...)	解释
0	0000	任务完成未出错。若 $LEN < AINFO$ 的长度, 则只发送附加中断信息中按 LEN 指定的字节数
0	00B1	$LEN > AINFO$ 的长度。任务完成, AINFO 中的附加中断信息被发送。STATUS[4] 接收到 AINFO 的长度。
0	7000	初始调用但 REQ=0 (空循环)。无中断被传送。BUSY 的值为 0。
0	7001	初始调用且 REQ=1。任务已经开始。BUSY 的值为 1。
0	7002	中间调用 (REQ 无关)。发送的中断还未被 DP 主站应答。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未组态。
1	8091	<ul style="list-style-type: none"> <li>• 在组态中封锁了中断</li> <li>• 中断在此类型的从站上不允许</li> </ul>
1	8092	AINFO 中无效的数据类型 (BYTE 和 BLOCK-DB 是有效的类型)
1	8093	参数 ID 属于一个虚拟槽位。它不可能发出中断请求。
1	80B0	ASPEC <ul style="list-style-type: none"> <li>• 与 AINFO 的 0 字节第 0 位不匹配</li> <li>• 对于使用的中断类型必须为 0 值</li> <li>• 超出允许的数值范围</li> </ul>
1	80B1	LEN 超出允许的数值范围。STATUS[4] 接收到 B#16#FF。
1	80B5	在 DP 主站上调用 SFB75 不允许
1	80C5	分布式 I/O 装置此时不可用 (例如, 站故障)
1	80C8	该功能在当前 DP 主站的运行模式下不允许



## 9 用于日期时间中断操作的 SFC

### 9.1 日期时间中断

#### 定义

日期时间中断会导致系统调用日期时间中断 OB 块（OB10~OB17）。

#### 调用环境

在系统调用日期时钟中断 OB 之前，必须适合以下情况：

- 日期时间中断 OB 已经利用以下两种方法设定好了相关参数（启动和运行的日期、时间）
  - STEP7 或
  - 用户程序中的 SFC28 “SET\_TINT”
- 日期时间中断必须用以下方法激活
  - STEP7 或
  - 用户程序中的 SFC30 “ACT\_TINT”
- 用户不能用 STEP7 取消日期时间中断 OB
- CPU 中必须有日期时间中断 OB
- 如果你利用 SFC30 “ACT\_TINT” 设定了日期时间中断并且该中断 OB 只执行一次，那么你所设置的日期和时间必须保证还没有进去。如果你设置的是循环执行，那么日期时间中断 OB 会在下一个周期（启动时间+设定时间的倍数）被调用。

#### 提示

你可以利用 STEP7 设定日期时间中断参数，然后在用户程序中用 SFC30 “ACT\_TINT” 激活中断。

#### SFC28~SFC31 的用途

系统功能 SFC28~SFC30 用法说明如下：

- 设置日期时间中断（SFC28 “SET\_TINT”）
- 取消日期时间中断（SFC29 “CAN\_TINT”）
- 激活日期时间中断（SFC30 “ACT\_TINT”）
- 查询日期时间中断（SFC31 “QRY\_TINT”）

## 9.2 SFC28~SFC31 特性

### 如果...会发生什么

下表列出了在不同的情况下，日期时间中断的不同效果

如果...	那么...
设置日期时间中断 (调用 SFC28, SET_TINT)	当前日期时间中断被取消
取消日期时间中断 (调用 SFC 29, CAN_TINT)	启动日期和时间被取消，日期时间中断在使用前必须重复赋值
在调用时日期时间中断 OB 不存在	当操作系统调用 OB85，而 OB85 不存在，则 CPU 进入 STOP 状态，并且有优先等级错误报警产生。
调整的时钟超前	由于时钟超前而跳过了启动日期/时间 <ul style="list-style-type: none"> <li>• 系统调用 OB80</li> <li>• 调用 OB80 后的每一个日期时间中断都被跳过 OB 被调用（一次，不管已经跳过多少次）在 OB80 中不能使用该操作</li> </ul> 如果 OB80 不存在，CPU 进入 STOP 状态
时钟向回调整	如果在时钟被调回的时间段内，日期时间中断 OB 已经被调用过，则时间再次到达时，不再被调用。

- 1) 在 OB80 的启动信息中，指明了由于时钟向前调整，哪些日期时间中断 OB 未被调用，启动信息中的时间对应向前调整的时间
- 2) 被跳过后又激活的日期时间中断 OB 启始信息中的时间对应第一次被跳过的日期时间中断的启始时间

### 暖启动或冷启动

在暖启动或者冷启动期间，用户程序中的所有通过 SFC 设定的日期时间中断设置全部被清空。通过 STEP7 进行的设定还生效。

### 执行日期时间中断 OB

下表列出了通过 STEP7 或利用 SFC28 “SET\_TINT”（输入参数 PER10D）设定“执行”参数所起到的不同效果。

执行日期时间中断 OB	效果
不执行 (只能利用 STEP 下进行设定)	即使 CPU 中有日期时间中断 OB，该中断也不执行。 其参数可以通过用户程序中的 SFC28 “SET_TINT” 重新进行设置。
执行一次	当日期时间中断 OB 被执行过后，该中断即被取消。它可以被重新设定并且再次调用。
周期运行 (每分，每小时，每天，每周，每月，每年)	当日期时间中断被激活时，它所设定的时间已经过去，那么该中断 OB 会在下一个循环周期的中断时间点（启动日期/时间+所选时间的倍数）执行中断操作。在极少数的情况下，当再次调用中断 OB 时，该时钟中断 OB 不执行。 原因： <ul style="list-style-type: none"> <li>• 时间错误，（系统调用了 OB80，而 CPU 中没有 OB80，CPU 进入 STOP 状态）</li> <li>• 日期时间中断 OB 延时</li> </ul>

## 9.3 用 SFC28 “SET\_TINT” 设置日期时间中断

### 描述:

利用 SFC28 “SET\_TINT”（设置日期时间中断），你可以设置日期时间中断 OB 的启动日期和时间，在设定启动时间时，秒和毫秒是被忽略的，且用 0 代替。

参数	声明	数据类型	存储区	描述
OB_NR	INPUT	INT	I, Q, M, D, L, 常数	中断 OB 号 (OB10—OB17) 启动时间 SDT+PER10D 倍数
SDT	INPUT	DT	D, L, 常数	启动时间：所定义的启动时间中秒和毫秒省略，用 0 代替
PERIOD	INPUT	WORD	I, Q, M, D, L, 常数	SDT 启动周期 W#16#0000 =一次 W#16#0201 =每分钟 W#16#0401 =每小时 W#16#1001 =每天 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	当中断被激活后有故障时，故障代码会显示在 RET_VAL 中

### 故障信息

故障代码 (W#16#...)	说明
0000	没有故障产生
8090	参数 OB_NR 不正确
8091	参数 SDT 不正确
8092	参数 PERIOD 不正确
80A1	所设置的时间已经过去了

## 9.4 用 SFC29 “CAN\_TINT” 除日期时间中断

### 描述

利用 SFC29 “CAN\_TINT”（取消日期时间中断），你可以取消日期时间中断组织块。

参数	声明	数据类型	存储区	描述
OB_NR	INPUT	INT	I, Q, M, D, L, 常数	中断 OB 号 (OB10-OB17) 此块的启动日期和时间将被取消。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	当此功能被激活存有错误产生相应的故障代码会显示在 RET_VAL 中。

### 故障信息

故障代码 (W#16#...)	说明
0000	无故障产生
8090	参数 OB_NR 不正确
80A0	在日期时间中断 OB 中没有启动日期/时间

## 9.5 用 SFC30 “ACT\_TINT” 启动日期时间中断

### 描述

利用 SFC30 “ACT\_TINT”（启动日期时间中断），你可以启动日期时间中断组织块

参数	声明	数据类型	存储区	描述
OB_NR	INPUT	INT	I, Q, M, D, L, 常数	被启动的 OB 号 (OB10—OB17)
RET_VAL	OUTPUT	INT	I, Q, M, D, L	当此功能被激活后有故障产生。相应的故障代码会显示在 RET_VAL 中

### 故障信息

故障代码 (W#16#...)	说明
0000	无故障产生。
8090	参数 OB_NR 不正确。
80A0	有的日期时间中断 OB 没有设置启动日期/时间。
80A1	启动时间已经过去了，如果你设置的中断状态为只执行一次，那么该故障代码才出现。



## 9.6 用 SFC 31 “QRY\_TINT” 查询日期时间中断

### 描述

利用 SFC 31 “QRY\_TINT”（查询日期时间中断），你可以通过输出参数 STATUS 查看日期时间中断组织块的状态

参数	声明	数据类型	存储区	描述
OB_NR	INPUT	INT	I, Q, M, D, L, 常数	需要查询的中断 OB 号 (OB10-OB17)。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如此功能被激活时有错误产生则相应的故障代码会显示在 RET_VAL 中。
STATUS	OUTPUT	WORD	I, Q, M, D, L	日期时间中断状态； 下见表。

### 输出参数 STATUS

位	值	含义
0	0	操作系统已激活日期时间中断
1	0	允许新的日期时间中断
2	0	日期时间中断未被激活或所设时间已过去
3	-	-
4	0	没有装载日期时间中断 OB
5	0	在测试状态下正在进行的日期时间中断 OB 不能够被取消
6	0	日期时间中断以基准时间为准
	1	日期时间中断以当地时间为准

### 故障信息

故障代码 (W#16#...)	说明
0000	无故障产生
8090	参数 OB_NR 不正确



# 10 处理延时中断的 SFC

## 10.1 处理延时中断

### 定义

调用了 SFC 32 “SRT\_DINT” 之后，在指定的延时时间到达后操作系统会产生一个中断，换言之，即选定的延时中断 OB 被调用。这个中断称为延时中断。

### 调用的前提条件

在一个延时中断 OB 被操作系统调用之前，必须满足下列条件：

- 延时中断 OB 必须由 SFC 32 “SRT\_DINT” 启动
- 延时中断 OB 必须是在 STEP7 中被选中的
- 延时中断 OB 必须在 CPU 中存在

### SFC32 到 SFC34 的功能

SFC32 到 SFC34 的系统功能见下列描述：

- 启动延时中断（SFC 32 “SRT\_DINT”）
- 删除延时中断（SFC 33 “CAN\_DINT”）
- 查询延时中断（SFC 34 “QRY\_DINT”）

### 如果...会...

如果...	并且...	那么...
一个延时中断启动 (SFC 32 “SRT_DINT”)	这个延时中断已经启动	延时时间被重写；再次启动延时中断
	调用时延时中断 OB 不存在	操作系统产生一个优先级故障（调用 OB85） 若 OB85 不存在，CPU 停机
	这个中断已由启动 OB 启动并且在 CPU 转到运行状态之前延时时间到	直到 CPU 进入运行状态延时中断 OB 方被调用
延时时间到	先前的启动延时中断 OB 仍在执行	操作系统产生一个时间故障（调用 OB80）。 若 OB80 不存在，CPU 停机

### 暖启动和冷启动

在暖或冷启动过程中，所有用户程序中通过 SFC 设置的延时中断被清除。

### 在启动 OB 中启动

一个延时中断可在启动 OB 中启动。调用延时中断 OB，必须满足下面两个条件：

- 延时时间已到
- CPU 必须在运行状态

如果延时时间到而 CPU 仍未到运行状态，则延时中断的调用将被延迟直到 CPU 进入运行状态。时间中断 OB 将在第一次执行 OB1 前被调用。

## 10.2 用 SFC32 “SRT\_DINT” 启动延时中断

### 描述

通过 SFC32 “SRT\_DINT”（启动延时中断）可以启动一个延时中断，一旦延时时间到（参数 DTIME），调用延时中断组织块。

由参数 SIGN，可以确认延时中断是否启动。DTIME 与 SIGN 的值在被指定的组织块执行时会再次出现在起始事件信息中。

参数	声明	数据类型	存储区域	说明
OB_NR	INPUT	INT	I, Q, M, D, L, 常数	组织块的编号 (OB20 到 OB23)，延时后被启动
DTIME	INPUT	TIME	I, Q, M, D, L, 常数	延时值 (1 到 60000ms) 可以设一个更长的时间，例如，通过在延时中断 OB 中使用计数器。
SIGN	INPUT	WORD	I, Q, M, D, L, 常数	当延时中断 OB 被调用时在起始事件信息中出现的开始标志
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如果在系统功能执行时出现故障，则参数 RET_VAL 中包含一个故障代码

### 准确性

假如没有其它中断事件延迟调用的话，那么用 SFC 32 “SRT\_DINT” 调用和启动延时中断 OB 的时间最大为 1ms，小于选定的时间。

### 故障信息

故障代码 (W#16#...)	说明
0000	无故障发生
8090	参数 OB_NR 不正确
8091	参数 DTIME 不正确

## 10.3 用 SFC 34 “QRY\_DINT” 查询一个延时中断

### 描述

通过 SFC 34 “QRY\_DINT”（查询延时中断）可以查询延时中断 OB 的状态。

### 组织块 OB20 到 OB23 控制延时中断

参数	声明	数据类型	存储区域	说明
OB_NR	INPUT	INT	I, Q, M, D, L, 常数	被查询状态的组织块编号（OB20 到 OB23）
RET_VAL	OUTPUT	INT	I, Q, M, D, L	在功能执行时或发生故障，参数 RET_VAL 中包含一个故障代码
STATUS	OUTPUT	WORD	I, Q, M, D, L	时间中断的状态，见下表

### 输出参数 STATUS

位	值	含义
0	0	延时中断已被操作系统使能
1	0	不接受新的延时中断
2	0	延时中断未被激活或已完成
3	-	-
4	0	延时中断组织块未被装载
5	0	当测试功能激活时延时中断组织块也可以执行

### 故障信息

故障代码（W#16#...）	说明
0000	无故障
8090	参数 OB_NR 不正确

## 10.4 用 SFC33 “CAN\_DINT” 取消一个延时中断

### 描述

通过 SFC33“CAN\_DINT”（取消延时中断），可以取消一个已启动的延时中断（见表 SFC32 “SRT\_DINT” 启动延时中断一节）。这样，延时中断 OB 就不被调用了。

参数	声明	数据类型	存储区域	说明
OB_NR	INPUT	INT	I, Q, M, D, L, 常数	被取消的 OB 编号（OB20 到 OB23）
RET_VAL	OUTPUT	INT	I, Q, M, D, L	在功能执行时若发生故障，则参数 RET_VAL 中包含故障代码

### 故障信息

故障代码（W#16#...）	说明
0000	无故障发生
8090	参数 OB_NR 不正确
80A0	延时中断还未启动

# 11 处理同步故障的 SFC

## 11.1 屏蔽同步故障

### 简介

同步故障是程序和访问故障。这些故障发生是因为程序中存在不正确的地址区域、编号，或不正确的地址。屏蔽这些故障则意味着：

- 被屏蔽的同步故障不会触发故障 OB 的调用，不会发生程序选择的响应。
- 在故障寄存器中 CPU 记录已发生的但被屏蔽掉的故障。

通过调用 SFC 36 “MSK\_FLT” 执行屏蔽功能。

去除故障屏蔽即消除一个先前设置为屏蔽的故障，并且在当前的优先级中清除事件状态寄存器中相应的位。下面介绍去除故障屏蔽：

- 通过调用 SFC 37 “DMSK\_FLT”。
- 在当前的优先级结束之后。

如果去除故障屏蔽后产生故障，操作系统会启动相应的故障 OB。可以编程 OB121 作为程序故障的响应，OB122 作为访问故障的响应。

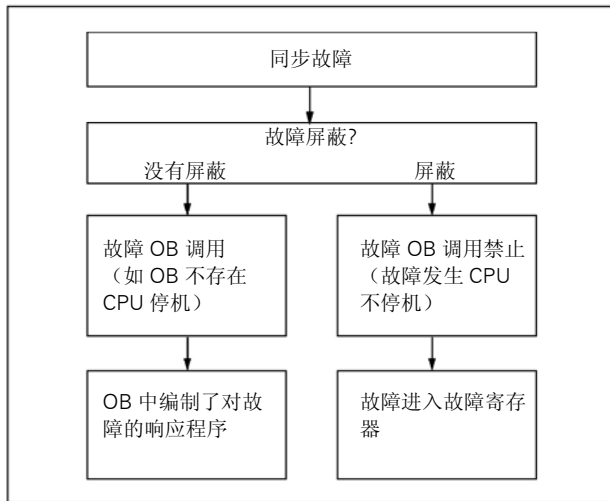
可以通过 SFC 38 “READ\_ERR” 读出已经发生的被屏蔽掉的故障。

注意：对于 S7-300（CPU 318 除外），不管故障是否被屏蔽，都会进入诊断缓冲区，CPU 上的“组故障”指示灯亮。

### 故障处理的一般程序

如果用户程序中发生程序和访问故障，可以不同的方式处理：

- 编程一个故障 OB，在相应的故障发生时由操作系统调用。
- 可以在不同优先级中单独地禁止故障 OB，这种情况下，在特定的优先级中发生此类故障 CPU 不会停机，CPU 把此故障放到故障寄存器。但从这儿不能看出什么时间或多长时间发生一次故障。



### 过滤器

同步故障被送到一个特殊的位模型,称为**故障过滤器(屏蔽)**,故障过滤器也存在于 SFC 36、37 和 38 的输入输出参数中。

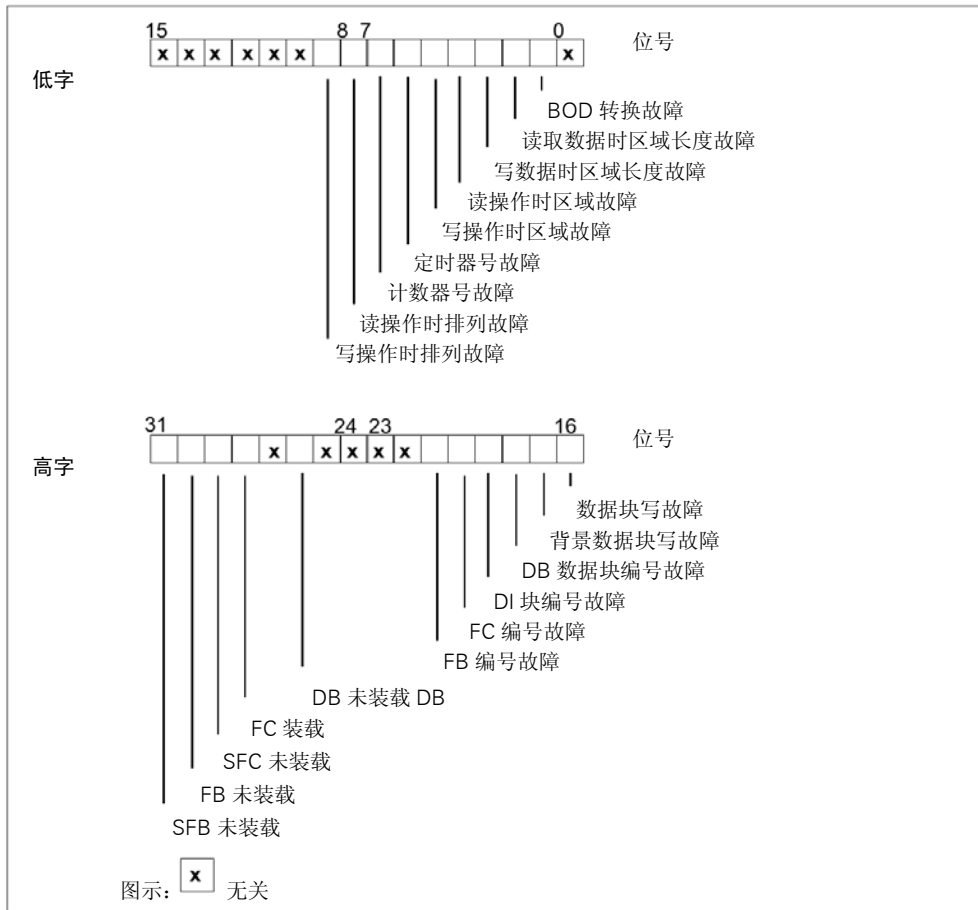
同步故障分为程序故障和访问故障,可用两个故障过滤器屏蔽,故障过滤器的说明见下图。

### 程序故障过滤器

下图为程序故障过滤器的位模型,程序故障过滤器位于参数 PRGFLT\_...中。

参见程序故障低字,可能的故障原因或者程序故障高字,可能的故障原因:





**注意:**

程序故障过滤中高字中位 29 (“SFC 未装载”)和位 31 (“SFB 未装载”)只是与 S7-400 和 CPU 318 有关。

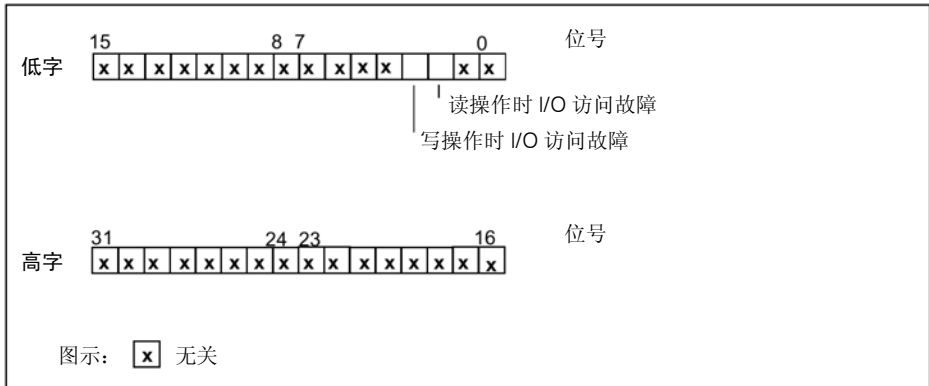
**无关的位**

在上图中，x 表示…

- ...输入参数      对于 SFC 36、37、38      = “0”
- ...输出参数      对于 SFC 36、37      = “1” 对于 S7-300  
   = “0” 对于 S7-400
- 对于 SFC 38      = “0”

### 所有 CPU 的访问故障过滤器

下图为除 CPU 417 和 CPU 417H 之外的所有 CPU 的访问故障过滤器的位模型，访问故障过滤器位于参数 ACCFLT\_...。关于访问故障过滤器的说明，参见表“除 CPU 318 外所有 CPU 31x 的可能故障原因”，或者“除 CPU 41x 和 CPU 318 外所有 CPU 的可能故障原因”。



### 无关的位

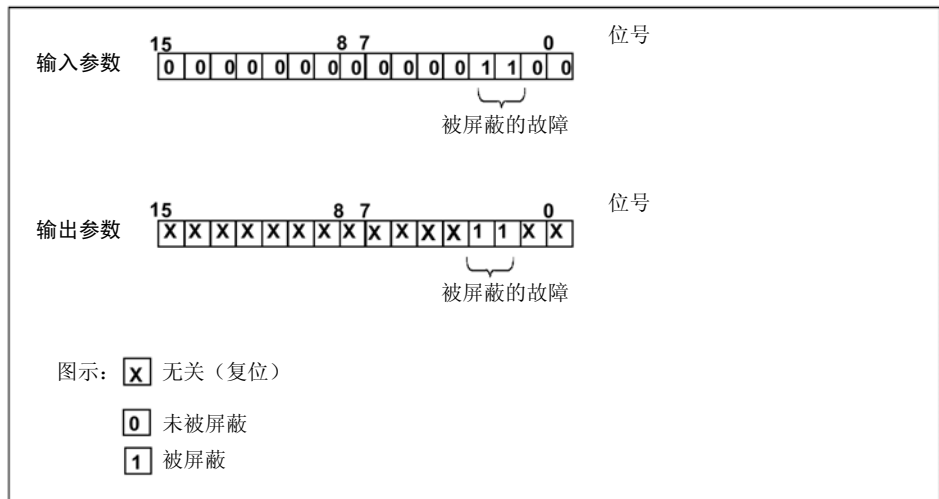
在上图中，x 表示...

- ...输入参数 对于 SFC 36、37、38 = “0”
- ...输出参数 对于 SFC 36、37 = “1”，对于 S7-300 = “0”，S7-400 与 CPU 318 = “0”
- 对于 SFC 38 = “0”

### 举例说明

下图表明了带有屏蔽故障的访问故障过滤器的低字在 CPU 417 和 CPU 417H 中是如何出现的。

- 作为 SFC 36 的输入参数
- 作为 SFC 36 的输出参数。



### 程序故障低字

下表列出了分配给程序故障过滤器低字的故障以及可能的故障原因。

程序故障可能的原因，低字

故障	代码 (W#16#...)	故障出现...
BCD 转换故障	2521	...当被转换的数值不为 BCD 码时（例如，5E8）。
读操作时区间长度故障	2522	...当所用到的地址完全不在可能的地址区域中。 例如：存储区只有 256 字节长，读取 MW 320。
写操作时区间长度故障	2523	...当所用到的地址完全不在可能的地址区域中。 例如：存储区只有 256 字节长，写数值到 MW 320。
读操作时区间故障	2524	...当使用间接、交叉寻址时指定给地址一个不正确的区域标识。 例如： 正确： LAR1 P#E 12.0 LW[AR1, P#0.0] 例如： 不正确： LAR1 P# 12.0 LW[AR1, P#0.0] 这种操作，会出现区间长度故障。
写操作时区间故障	2525	...当使用间接、交叉寻址时指定给地址一个不正确的区域标识。 例如： 正确： LAR1 P#E 12.0 TW[AR1, P#0.0] 例如： 不正确： LAR1 P# 12.0 TW[AR1, P#0.0] 这种操作，会出现区间长度故障。

故障	代码 (W#16#...)	故障出现...
定时器编号故障	2526	...当访问一个不存在的定时器时。 例如: SPT[MW0], 且 MW0=129, 只有 128 个定时器, 却启用定时器 129。
计数器编号故障	2527	...当访问一个不存在的计数器时。 例如: CUC[MW0], 且 MW0=600, 只有 512 个计数器 (CPU 416-D), 却要访问计数器 600。
读操作时排列故障	2528	...当一个字节、字或双字的地址用一个不等于 0 的位地址赋值时。 例如: 正确: LAR1 P#M12.0 LB[AR1, P#0.0] 例如: 不正确: LAR1 P#M12.4 LB[AR1, P#0.0]
写操作时排列故障	2529	...当一个字节、字或双字的地址用一个不等于 0 的位地址赋值时, 例如: 正确: LAR1 P#M12.0 TB[AR1, P#0.0] 例如: 不正确: LAR1 P#M12.4 TB[AR1, P#0.0]

### 程序故障高字

下表列出了分配给程序故障过滤器高字的故障, 以及可能的故障原因。

程序故障高字, 可能的故障原因

故障	代码 (W#16#...)	故障出现...
DB 写故障	2530	...当要写入的数据块是只读的
DI 写故障	2531	...当要写入的背景数据块是只读的
DB 编号故障	2532	...当被打开的数据块编号超出最大允许值时
DI 编号故障	2533	...当被打开的背景数据块编号超出最大允许值时
FC 编号故障	2534	...调用的功能编号超出最大允许值
FB 编号故障	2535	...调用的功能块编号超出最大值
DB 未装载	253A	...打开的数据块未被装载
FC 未装载	253C	...调用的功能未被装载
SFC 未装载	253D	...调用的系统功能不存在
FB 未装载	253E	...调用的功能块未被装载
SFB 不存在	253F	...调用的系统/标准功能块不存在

## 访问故障

下表列出了分配给访问故障过滤器的故障以及可能的故障原因。

故障	代码(W#16#...)	故障出现...
读操作时 I/O 访问故障	2942	...当在 I/O 区没有分配给所用地址的信号模板。 或者 ...在选定模板的监控时限（时间到）内对此 I/O 区的访问不被承认。
写操作时 I/O 访问故障	2943	...当在 I/O 区没有分配给所用地址的信号模板。 或者 ...在选定模板的监控时限（时间到）内对此 I/O 区的访问不被承认。

## 11.2 用 SFC 36 “MSK\_FLT” 屏蔽同步故障

## 描述

通过 SFC 36 “MSK\_FLT”（屏蔽同步故障），可以控制 CPU 对同步故障的响应。通过此 SFC 可以用故障过滤器屏蔽同步故障（参见“屏蔽同步故障”）。调用 SFC 36 的同时就在当前的优先级中屏蔽掉了同步故障。

如果把同步故障过滤中输入参数中的个别位设为 1，其它先前设定的位仍保持为 1，那么就得到一个新的故障过滤器。并且可以通过输出参数读出，屏蔽掉的同步故障发生时不用 OB，只被写入故障寄存器中，可以通过 SFC 38 “READ\_ERR” 读故障寄存器。

参数	声明	数据类型	存储区域	说明
PRGFLT_SET_MASK	INPUT	DWORD	I, Q, M, D, L, 常数	屏蔽的程序故障
ACCFLT_SET_MASK	INPUT	DWORD	I, Q, M, D, L, 常数	屏蔽的访问故障
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
PRGFLT_MASKED	OUTPUT	DWORD	I, Q, M, D, L	被屏蔽的程序故障
ACCFLT_MASKED	OUTPUT	DWORD	I, Q, M, D, L	被屏蔽的访问故障

## 故障信息

故障代码 (W#16#...)	说明
0000	没有已经被屏蔽的故障。
0001	至少已屏蔽了一个故障。 其它故障永远不能被屏蔽掉。

## 11.3 用 SFC 37 “DMSK\_FLT” 解除同步故障的屏蔽

### 描述

通过 SFC 37 “DMSK\_FLT”（解除同步故障屏蔽），可以对通过 SFC 36 “MSK\_FLT” 屏蔽的同步故障解除屏蔽。通过调用 SFC 37，可以在当前的优先级中解除相应同步故障的屏蔽。同时，故障寄存器清除记录，可以由输出参数读出新的故障过滤器。

参数	声明	数据类型	存储区域	说明
PRAFLT_RESET_MASK	INPUT	DWORD	I, Q, M, D, L, 常数	解除屏蔽的程序故障
ACCFLT_RESET_MASK	INPUT	DWORD	I, Q, M, D, L, 常数	解除屏蔽的访问故障
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
PRGFLT_MASKED	OUTPUT	DWORD	I, Q, M, D, L	仍被屏蔽的程序故障
ACCFLT_MASKED	OUTPUT	DWORD	I, Q, M, D, L	仍被屏蔽的访问故障

### 故障信息

故障代码 (W#16#...)	说明
0000	所有指定的故障被解除屏蔽。
0001	至少有一个故障没被屏蔽。 其它故障永远不能解除屏蔽。

## 11.4 用 SFC 38 “READ\_ERR” 读故障寄存器中信息

### 描述

通过 SFC 38 “READ\_ERR”（读故障寄存器），可以读取故障寄存器中信息。

故障寄存器的结构同程序和访问故障过滤器相对应。故障过滤器是可以作为输入参数通过 SFC 36 和 SFC 37 编程的。

在输入参数中，输入一个欲从故障寄存器读出的同步故障，调用 SFC 38，就可在故障寄存器中读取所要求的记录，同时，清除记录。

从故障寄存器中还可以看出哪些被屏蔽的同步故障在当前的优先级中至少发生了一次。如果一个位为 1，表示相应被屏蔽的故障至少发生了一次。

参数	声明	数据类型	存储区域	说明
PRGFLT_QUERY	INPUT	DWORD	I, Q, M, D, L, 常数	查询程序故障
ACCFLT_QUERY	INPUT	DWORD	I, Q, M, D, L, 常数	查询访问故障
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
PRGFLT_CLR	OUTPUT	DWORD	I, Q, M, D, L	发生的程序故障
ACCFLT_CLR	OUTPUT	DWORD	I, Q, M, D, L	发生的访问故障

### 故障信息

故障代码 (W#16#...)	说明
0000	所有查询的故障都被屏蔽
0001	至少有一个查询的故障未被屏蔽





## 12 处理中断和异步故障的 SFC

### 12.1 延迟和禁止中断和异步故障

#### SFC 39 到 SFC 42 的功能

通过这些 SFC，可以做到：

- 用 SFC 39 “DIS\_IIRT” 禁止随后的所有 CPU 循环过程中的中断和异步故障
- 用 SFC 41 “DIS\_AIRT” 延迟高优先级的执行直到本 OB 结束。
- 用 SFC 40 “EN\_IIRT” 或 SFC 42 “EN\_AIRT” 使能中断和异步故障。

需要在用户程序中编制对中断和异步故障的处理程序，同时还必须编程相应的 OB。

#### SFC 41 到 SFC 42 的功用

通过 SFC 41 “DIS\_AIRT” 来延迟高优先权中断和异步故障程序，接下来再由 SFC 42 “EN\_AIRT” 重新激活中断和异步故障。这样做的好处在于：

被中断的数量依 CPU 类型而定，如果中断和异步故障已被延迟，这个延迟不能用标准 FC 取消。只能用标准 FC 将其激活。

#### 中断种类

中断分为多种，下表列出了所有的中断种类及相应的 OB。

中断种类	OB
时间日期中断	OB10~OB17
延时中断	OB20~OB23
循环中断	OB30~OB38
硬件中断	OB40~OB47
DPV1 中断	OB55~OB57
多处理器中断	OB60
冗余故障中断	OB70, OB72
异步故障中断	OB80~OB87（见下表）
同步故障中断	OB121, OB122 （可以用 SFC 36~SFC 38 屏蔽同步故障中断，也可解除对其的屏蔽）

### 异步故障

下表列出了可以通过用户程序的 OB 来响应的异步故障。

异步故障	OB
时间故障（例如，循环扫描时间超长）	OB80
电源故障（例如，电池故障）	OB81
诊断故障（例如，信号模板上的熔断器故障）	OB82
插拔模板故障	OB83
CPU 硬件故障（例如，存储卡被取下）	OB84
程序故障	OB85
机架故障	OB86
通讯故障	OB87

## 12.2 用 SFC 39 “DIS\_IRT” 禁止新的中断和异步故障的处理

### 描述

通过 SFC 39 “DIS\_IRT”（禁止中断），可以禁止新中断的异步故障的处理。这意味着如果一个中断出现，CPU 的操作系统做出如下响应：

- 既不调用中断 OB 也不调用异步故障 OB。
- 如果没有编制中断 OB 和异步故障 OB 的程序不会引发通常的响应。

如果禁止了中断和异步故障，会影响所有的优先级。“DIS\_IRT”影响的消除只能通过调用 SFC 40 “EN\_IRT”（见 0 节）或者暖启动或者冷启动。

当中断和异步故障发生时操作系统是否将他们写入诊断缓冲冲击取决于在输入参数 MODE 中的设置。

### 注意

如果程序中调用了 SFC 39 “DIS\_IRT”，那么所有的中断都不起作用！

参数	声明	数据类型	存储区域	说明
MODE	INPUT	BYTE	I, Q, M, D, L, 常数	定义被禁止的中断和异步故障
OB_NR	INPUT	INT	I, Q, M, D, L, 常数	OB 编号
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能执行时若发生故障，此参数中包含一个故障代码

## MODE

MODE	含义
00	所有新发生的中断和异步故障被禁止（同频故障不被禁止），参数 OB_NR 置 0。在诊断缓冲区里连续记录所发生的故障。
01	所有新发生的，且指定类别的中断故障被禁止。通过指定最小可能的 OB 号来指定中断种类，即使指定的 OB 在 CPU 中不存在也没有关系。 例如： <ul style="list-style-type: none"> <li>如欲禁止监控中断，在 OB_NR 中输入 30（即便 OB30 不是 CPU 中第一个监控 OB）。</li> <li>如欲禁止所有的硬件中断，在 OB_NR 输入 40。</li> </ul> 故障会在缓冲区内被连续记录。
01	所有新发生的且指定类别的中断故障被禁止，通过下面值来指定中断种类： <ul style="list-style-type: none"> <li>时间日期中断：10</li> <li>延时中断：20</li> <li>循环中断：30</li> <li>硬件中断：40</li> <li>DPV1 中断：50</li> <li>多处理器中断：60</li> <li>冗余故障中断：70</li> <li>异步故障中断：80</li> </ul> 故障会在缓冲区内被连续记录。
02	所有新发生的指定的中断被禁止，使用 OB 号来指定中断种类，故障会在缓冲区内被连续记录。
80	所有新发生的中断和异步故障被禁止，并且故障不会在缓冲区内被记录。操作系统在诊断缓冲区里输入事件 W#16#5380。
81	所有新发生的且指定类别的中断被禁止，并且故障不会在缓冲区内被记录，操作系统在缓冲区内输入事件 W#16#5380。
82	所有新发生的且指定的中断被禁止，并且故障不会在缓冲区内被记录，操作系统在缓冲区里输入事件 W#16#5380。

## 故障信息

故障代码 (W#16#...)	说明
0000	无故障发生。
8090	输入参数 OB_NR 中包含一个非法字符。
8091	输入参数 MODE 中包含一个非法字符。

## 12.3 用 SFC 40 “EN\_IRT” 激活新中断和异步故障的处理

### 描述

通过 SFC 40 “EN\_IRT”（激活中断），可以激活先前由 SFC 39 “DIS\_IRT” 禁止的新中断和异步故障。这意味着，如果一个中断事件发生，操作系统会以下面的方式响应：

- 调用中断 OB 或者异步故障 OB。
- 如果中断 OB 或者异步故障 OB 未被编程，则触发标准的响应。

参数	声明	数据类型	存储区域	说明
MODE	INPUT	BYTE	I, Q, M, D, L, 常数	定义被激活的中断和异步故障中断
OB_NR	INPUT	INT	I, Q, M, D, L, 常数	OB 编号
RET_VAL	OUTPUT	INT	I, Q, M, D, L,	如果功能执行时发生故障，此参数中包含一个故障代码

### MODE

MODE (B#16#...)	含义
0	所有新发生的中断和异步故障被激活。
1	所有新发生的且指定类别的中断被激活。通过下面数值指定中断种类： <ul style="list-style-type: none"> <li>• 时间日期中断：10</li> <li>• 延时中断：20</li> <li>• 循环中断：30</li> <li>• 硬件中断：40</li> <li>• DPV1 中断：50</li> <li>• 多处理器中断：60</li> <li>• 冗余故障中断：70</li> <li>• 异步故障中断：80</li> </ul>
2	所有新发生的指定的中断被激活，通过 OB 号定义中断。

### 故障信息

故障代码 (W#16#...)	说明
0000	无故障发生。
8090	输入参数 OB_NR 中包含一个非法字符。
8091	输入参数 MODE 中包含一个非法字符。

## 12.4 用 SFC 41 “DIS\_AIRT” 延迟一个高优先级的中断和异步故障的处理

### 描述

通过 SFC 41 “DIS\_AIRT”（去活报警中断），可以延迟比当前 OB 优先级高的中断 OB 和异步故障 OB 的执行。可在 OB 中多次调用 SFC 41。操作系统记录 SFC 41 的调用次数。一次调用会一直生效直到被 SFC 42 “EN\_AIRT” 再次取消或者当前的 OB 执行完毕。

一旦中断和异步故障被 SFC 42 “EN\_AIRT” 重新激活，在 SFC 41 起作用时被推迟的中断和异步故障会马上被处理，或者，当前的 OB 一被执行就得到处理。

参数	声明	数据类型	存储区域	说明
RET_VAL	OUTPUT	INT	I, Q, M, D, L	延迟的编号（等于 SFC 41 调用的编号）

### 返回值

下面是 SFC 41 中输出参数 RET\_VAL 即返回值的说明。

返回值	说明
n	“n” 为被禁止的次数，换言之即调用 SFC 41 的次数。 （仅当 n=0 时中断才可以被重新激活；参见用 SFC 42 “EN_AIRT” 激活高优先级中断和异步故障的处理）。

## 12.5 用 SFC 42 “EN\_AIRT” 激活具有高优先级的中断和异步故障的处理

### 描述

通过 SFC 42 “EN\_AIRT”（激活报警中断），可以激活先前被 SFC 41 “DIS\_AIRT” 禁止的具有高优先级的中断和异步故障的处理，每次调用 SFC 41 都必须通过调用 SFC 42 才能取消。

### 举例说明

如果禁止中断调用了 5 次 SFC 41，那么必须调用 SFC 42 五次才能激活中断。

参数	声明	数据类型	存储区域	说明
RET_VAL	OUTPUT	INT	I, Q, M, D, L	激活报警中断需调用 SFC 42 的次数或者故障信息。

### 返回值与故障信息

参见输出参数 RET\_VAL 中的故障说明。

在 0 章中描述了怎样通过参数 RET\_VAL 了解故障信息，本章中是否包了 SFC 的一般的故障信息。下面特别给出 SFC 42 中可以通过参数 RET\_VAL 输出的所有故障信息。

返回值和故障信息	说明
n	“n” 为仍未被 SFC 42 消掉的 SFC 41 的调用次数。 (只当 n=0 时中断才可被重新激活)。
W#16#8080	激活功能再次被调用，虽然中断已被激活。

## 13 用于诊断的 SFC

### 13.1 系统诊断

CPU 保留可编程控制器状态的内部数据。通过系统诊断功能，可以读取非常重要的数据，一些数据可以在装有 STEP 7 的编程器上显示出来。

通过 SFC “RD\_SINFO” 与 “RDSYSST”，也可在程序中访问系统诊断需要的数据。

### 13.2 用 SFC 6 “RD\_SINFO” 读取 OB 启动信息

#### 描述

通过 SFC 6 “RD\_SINFO”（读启动信息），可以读取下列 OB 的启动信息。

- 最近调用的还未被执行完毕的 OB。
- 最近启动的启动 OB。

两种情况下都没有时间标识。如果在 OB100、OB101 或 OB102 中调用 SFC 6，会返回两条相同的启动信息。

参数	声明	数据类型	存储区域	说明
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
TOP_SI	OUTPUT	STRUCT	D, L	当前 OB 的启动信息
START_UP_SI	OUTPUT	STRUCT	D, L	最近启动的启动 OB 的信息

#### TOP\_SI 与 START\_UP\_SI

结构元素	数据类型	说明
EV_CLASS	BYTE	<ul style="list-style-type: none"><li>• 位 0~3: 事件标识</li><li>• 位 4~7: 事件种类</li></ul>
EV_NUM	BYTE	事件号
PRIORITY	BYTE	优先级号
NUM	BYTE	OB 号
TYP2_3	BYTE	数据 ID2-3: 定义进入 ZI2-3 的信息
TYP1	BYTE	数据 ID1: 定义进入 ZI1 的信息
ZI1	WORD	附加信息 1
ZI2_3	DWORD	附加信息 2-3

**注意**

上表中列出的结构元素与 OB 中的临时变量内容相同。

请注意个别 OB 的临时变量可能有不同的名称与数据类型，还要注意每个 OB 的调用接口包括 OB 需要的日期与时间的附加信息。

结构元素 EV\_CLASS 的位 4~7 包含事件种类的信息。下面的值是可能的：

- 1：标准 OB 中的启动事件。
- 2：同步故障 OB 中的启动事件。
- 3：异步故障 OB 中的启动事件。

结构元素 PRIORITY 给出了当前 OB 的优先级（见 0 章）。

除了前两个元素，NUM 也是个重要元素，NUM 包含当前 OB 或最近启动的启动 OB 的编号。

**示例**

最近调用的还未被执行完毕的 OB 假设是 OB80，最近启动的启动 OB 假设是 OB100。

下表给出了 SFC 6 “RD\_SINFO” 中参数 TOP\_SI 的结构元素的配置以及 OB80 的局部变量。

TOP_SI		OB80	
结构元素	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



下表给出了 SFC 6 “RD\_SINFO” 中参数 START\_UP\_SI 的结构元素的配置以及 OB100 的局部变量。

START_UP_SI		OB100	
结构元素	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

#### 故障信息

SFC 6 “RD\_SINFO” 只提供一般故障的信息，没有特殊故障的信息。一般故障的代码及详细说明参见标题为“SFC 的通用参数”的章节中通过输出参数 RET\_VAL 分析故障部分。

## 13.3 用 SFC 51 “RDSYSST” 读取系统状态信息表或部分状态信息表

#### 描述

通过 SFC 51 “RDSYSST”（读系统状态），可以读取系统状态信息表或部分系统状态信息表。

调用 SFC 51，把 1 赋值给参数 REQ 即开始读取数据。如果系统状态信息可被马上读出，此 SFC 输出参数 BUSY 的值为 0。如果 BUSY 为 1，表示读操作还未结束（见 0 节）。

#### 注意

如果在诊断 OB 中调用 SFC 51“RDSYSST”，在 SSL\_ID 中输入 W#16#00B1、W#16#00B2 或 W#16#00B3，同时访问引起诊断中断的模板，那么立即就可读出系统状态信息。

#### 系统资源

如果启用了几个非同步的读任务（令 SSL\_ID 为 W#16#00B4、W#16#4C91、W#16#4092、W#16#4292、W#16#4692，还可能有 W#16#00B1、W#16#00B3）以一定的时间间隔一个接一个地执行，操作系统会确保所有任务的执行并且互不干扰。如果达到系统资源的极限，会在 RET\_VAL 中显示，由于任务的重复导致这类故障。

SFC 51 可以同时激活的任务的最大数目取决于 CPU 的能力，可以在/70/与/101/中找到相关信息。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	REQ=1: 正在启动
SSL_ID	INPUT	WORD	I, Q, M, D, L, 常数	被读的系统状态信息表或部分列表的 SSL_ID (部分信息列表在 0 章介绍)。
INDEX	INPUT	WORD	I, Q, M, D, L, 常数	部分信息表中元素的类型与编号。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	在 SFC 执行时若出现故障, 此参数包含一个故障代码。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	TRUE: 读任务还未结束。
SSL_HEADER	OUTPUT	STRUCT	D, L	见下页
DR	OUTPUT	ANY	I, Q, M, L, D	读出的 SSL 或部分 SSL 的目标区域: <ul style="list-style-type: none"> <li>• 如果只读取了 SSL 中的标题信息, 就不用查看 DR 而应查看 SSL_HEADER。</li> <li>• 若不是上面的情况, LENTHDR 与 N_DR 的乘积会指出有多少字节被写入 DR。</li> </ul>

### SSL\_HEADER

参数 SSL\_HEADER 是一种结构, 定义如下:

SSL\_HEADER: STRUCT

LENTHDR: WORD

N\_DR: WORD

END\_STRUCT

LENTHDR 是 SSL 或部分 SSL 中数据记录的长度。

- 如果仅读取了 SSL 中的标题信息, N\_DR 包含数据记录的个数。
- 否则, N\_DR 包含被输送到目标区域的数据记录的个数。

### 故障信息

故障代码 (W#16#...)	说明
0000	无故障
0081	目标区域太短 (仍然提供尽可能多的数据记录, SSL 中标题部分指出了数目)
7000	REQ=0 首次调用: 无数据传输; BUSY 为 0。
7001	REQ=1 首次调用: 数据传输开始; BUSY 为 1
7002	中间调用 (与 REQ 无关): 数据传输功能已激活; BUSY 为 1

故障代码 (W#16#...)	说明
8081	目标区域太短（一条数据记录都容纳不下）
8082	SSL_ID 错误，CPU 或 SFC 不能识别
8083	INDEX 错误或不允许
8085	因为系统总题，现在没有可用的信息（例如，因为资源缺乏）
8086	由于系统故障（总线、模板、操作系统）不能读取数据记录
8087	因为模板不存在或不对无法读取数据记录
8088	因为实际模板与设置的模板不同而无法读取数据记录
8089	因为模板不具有诊断能力不能读取数据记录
80A2	DP 协议故障（第 2 层故障）（临时故障）
80A3	用户接口/用户 DP 协议故障（临时故障）
80A4	通讯总线上故障（在 CPU 与外部 DP 接口板间发生故障）
80C5	没有分布式 I/O（临时故障）
80C6	由于优先级退出数据传输中止（重启或转入幕后）

## SSL\_ID

## 注意

可通过 SFC 51 “RDSYSST” 读取的部分信息表指：

- /70/对应于 S7-300
- 下表是对应于 S7-400 的。

SSL_ID (W#16#...)	部分信息表	INDEX (W#16#...)
	模板 ID	
0111	一条指定数据记录	
	模板标识	0001
	基础硬件标识	0006
	基础硬件标识	0007
	CPU 特性	
0012	所有特性	无关
0112	一组特性	
	MC7 处理单元	0001
	时间系统	0100
	系统行为	0200
	MC7 语言说明	0300
	可用的 SFC	0400

SSL_ID (W#16#...)	部分信息表	INDEX (W#16#...)
0F12	只读取 SSL 部分表标题信息	无关
	<b>用户存储区</b>	
0113	定义存储区的一条数据记录	
	工作存储器	0001
	<b>系统区域</b>	
0F14	所有系统区域的数据	无关
0F14	只有 SSL 部分表的标题信息	无关
	<b>模板类型</b>	
0015	所有模板的数据记录	无关
	<b>模板 LED 的状态</b> (所有的 CPU 中都读不出, 见 /102/)。	
0019	所有 LED 的状态	无关
0F19	只是 SSL 部分表标题信息	无关
	<b>组件标识</b>	
001C	所有组件标识	无关
011C	一个组件的标识	
	自动化系统的名称	0001
	CPU 名称	0002
	CPU 的系统 ID	0003
	版权记录	0004
	为操作系统保留的标题	0006
0F1C	只是 SSL 部分表的标题信息	无关
	<b>中断状态</b>	
0222	指定中断的数据记录	OB 号
	<b>CPU 与过程映象区的配置</b>	
0025	给 OB 配置所有的过程映象区	无关
0125	给相应的 OB 位置过程映象区	过程映象区号
0225	给相应的过程映象区配置 OB	OB 号
0F25	只是 SSL 部分表的标题信息	无关
	<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

SSL_ID (W#16#...)	部分信息表	INDEX (W#16#...)
0232	一个通讯单元的状态数据	
	CPU 的保护等级与操作员的权限设置	0004
	<b>H CPU 组信息</b>	
0071	H 系统的当前状态信息	无关
0F71	只是 SSL 部分表的标题信息	无关
	<b>模板 LED 的状态 (所有的 CPU 上都读不出, 见 /102/)</b>	
0174	一个 LED 的状态	LED ID
	<b>H 系统中可切换的 DP 从站</b>	
0075	H 系统与可切换的 DP 从站间的通讯状态	DP 从站接的诊断地址
	<b>DP 主站的系统信息</b>	
0090	CPU 承认的 DP 主站系统的信息	0000
0190	DP 主站系统的信息	DP 主站系统 ID
0F90	只是 SSL 部分表的标题信息	0000
	<b>模板状态信息 (提供最多 27 条数据记录)</b>	
0091	所有插入的模板/子模板的状态信息	无关
0191	所有具有错误 ID 的模板/机架的状态信息	无关
0291	所有故障模板的状态信息	无关
0391	所有没有用到的模板的状态信息	无关
0591	主模板上所有与模板的状态信息	无关
0991	指定机架上位于主模板上的所有子模板的状态信息	机架或 DP 主站 ID
0A91	所有 DP 主站系统模板的状态信息	无关
0C91	在中央机架上或连接到集成的 DP 通讯处理器的模板的状态信息	逻辑还始地址
0D91	在指定的机架/DP 站上的所有模板的状态信息	机架或 DP 主站 ID 和站号
0E91	所有配置模板的状态信息	无关
	<b>机架/站状态信息</b>	
0092	中央/DP 主站系统机架的期望状态	0/DP 主站系统 ID
4092	连接到外部 DP 口的 DP 主站的期望状态	DP 主站系统 ID
0192	通过集成的 DP 口连接的 DP 主站的工作状态	DP 主站系统 ID
0292	中央/DP 主站系统机架的当前状态	0/DP 主站系统 ID
0392	机架/模板机架上 CPU 的一块电池失效后后备电池的状态	0

SSL_ID (W#16#...)	部分信息表	INDEX (W#16#...)
0492	在所有机架/模板机架上的 CPU 的全部电池的状态	0
0592	所有机架/模板机架上的 CPU 的 24V 电源的状态	0
4292	通过外部接口板连接的 DP 主站系统中站的当前状态	DP 主站系统 ID
0692	扩展机架的 OK 状态/通过集成的 DP 接口板连接的 DP 主站系统的扩展机架的 OK 状态	0/DP 主站 ID
4692	通过外部 DP 接口板连接的 DP 主站系统的 OK 状态	DP 主站系统 ID
	<b>更多 DP 主站信息</b>	
0195	关于 DP 主站系统的更多信息	DP 主站系统 ID
0F95	只是 SSL 部分表的标题信息	0000
	<b>诊断缓冲区 (提供最多 21 条数据记录)</b>	
00A0	当前激活的操作模式下可提供的所有记录	无关
01A0	最近的记录, 在索引中指定数目	数量
0FA0	只是 SSL 部分表的标题信息	无关
	<b>模板诊断数据</b>	
00B1	一个模板的诊断数据的前 4 个字节(数据记录 0)	逻辑逻辑起始地址
00B2	一个模板的所有诊断数据 (≤220 字节, 数据记录 1) (无 DP 模板)	机架, 插槽
00B3	一个模板的所有诊断数据 (≤220 字节, 数据记录 1)	逻辑起始地址
00B4	DP 从站的诊断数据	组态的诊断地址

## 13.4 用 SFC 52 “WR\_USMSG” 在诊断缓冲器中写入一个用户定义的诊断事件

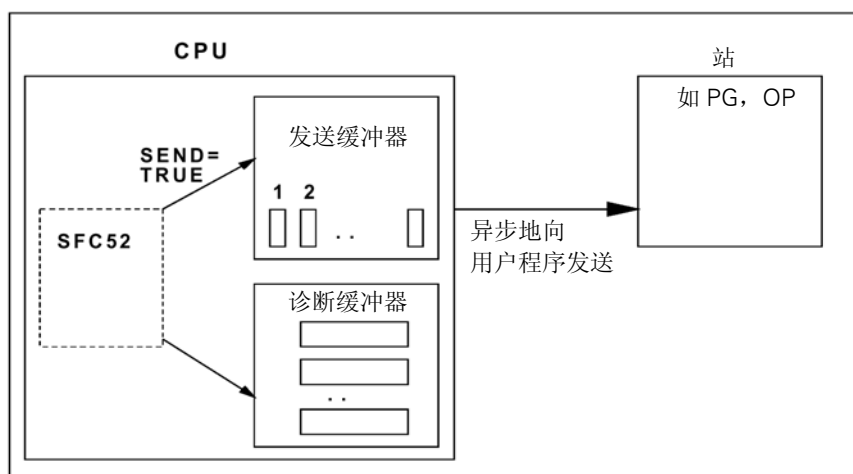
### 描述

通过 SFC 52 “WR\_USMSG”（在诊断缓冲器中写入用户信息），可以把一个用户定义的诊断事件写入诊断缓冲器，还可以把相应的诊断信息发送给所有为此而登录的站（通过设置输入参数 SEND=TRUE）。如发生故障，输出参数 RET\_VAL 会提供故障信息。

### 发送一条用户定义的诊断信息

SFC 52 把一个用户定义的诊断事件写入诊断缓冲器，还可以把相应的诊断信息发送给所有为此而登录的站（通过设置输入参数 SEND=TRUE），这样用户定义的诊断信息就写入了诊断缓冲器，并且自动送到登录的各站。

可以检查当前是否可能发送用户诊断信息。为此，需调用 SFC 51 “RDSYSST” 并且设置参数 SSL\_ID=W#16#0132 和 INDEX=WX16#0005，得到的数据记录的第四个字指出当前是否可能发送用户信息，（可能为 1，不可能为 0）。



### 发送缓冲区满

如果发送缓冲区未满，诊断信息只能进入发送缓冲区，发送缓冲区可以容纳的诊断信息的数目取决于所用 CPU 的型号。

如果诊断缓冲区已满，则：

- 诊断事件可进入诊断缓冲区。
- 参数 RET\_VAL 指示发送缓冲区已满（RET\_VAL =W#16#8092）

### 站未登录

如果一条用户定义的诊断信息要发送（SEND=TRUE）但没有站登录，

- 用户定义的诊断信息进入诊断缓冲区，

- 参数 RET\_VAL 指示没有站登录 (RET\_VAL = W#16#8091 或 W#16#8091。W#16#8091 出现于老版本的 CPU 中)。

### 通用结构

下表所示为诊断缓冲区内一个元素的内部结构：

字节	内容
1 和 2	事件 ID
3	优先级
4	OB 号
5 和 6	备用
7 和 8	附加信息 1
9、10、11 和 12	附加信息 2
13 到 20	时间标识

### 事件 ID

每个事件都有一个事件 ID。

### 附加信息

这里指事件的附加信息。每个事件的附加信息可以不同，当你生成一个诊断事件时，你可以自己决定这些信息的内容。

当发送一条用户定义的诊断信息时，可以在（指定的事件 ID）信息文本中集成这些附加信息为相关的数值。

### 时间标识

时间标识的型式为 Date\_and\_Time。

参数	声明	数据类型	存储区域	说明
SEND	INPUT	BOOL	I, Q, M, D, L, 常数	激活发送用户定的诊断信息到所有登录的站
EVENTN	INPUT	WORD	I, Q, M, D, L, 常数	事件 ID—分配给事件的标识号,它不是由信息服务器分配的。
INF01	INPUT	ANY	I, Q, M, D, L	1 个字长的附加信息
INF02	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	故障信息

### SEND

如果 SEND=TURE，用户定义的诊断信息发送到所有登录的站，只有在此站登录并且发送缓冲区未满时信息被发送。信息异步地发送到用户程序。



**EVENTN**

参数 EVENTN 包含用户事件的 ID，可以给事件 ID 赋值为 W#16#8xyz、W#16#9xyz、W#16#Axyz、W#16#Bxyz。

预定义的事件 ID 格式为 W#16#8xyz 与 W#16#9xyz。自由定义的事件 ID 格式为 W#16#16xyz 与 W#16#Bxyz。

一个事件进入通过 x=1 指示出来，一个事件离开通过 x=0 指示出来。对于 A 种与 B 种事件，“yz”是以 16 进制格式进行信息组态时分配给信息的编号。

事件 ID 的结构说明见 26.1 节。

**INFO1**

参数 INFO1 包含 1 个字长的信息，允许使用下列数据类型：

- WORD
- INT
- ARRAY [0 到 1] OF CHAR

可以在信息文本中将参数 INFO1 集成为一个相关的数值，这样就可以加入一些新信息。

**INFO2**

参数 INFO2 包含 2 个字长的信息，允许使用下列数据类型：

- WORD
- DINT
- REAL
- TIME
- ARRAY [0 到 3] OF CHAR

可以在信息文本中将参数 INFO2 集成为一个相关的数值这样就可以加入一些新信息。

**故障信息**

故障代码 (W#16#...)	说明
0000	无故障
0091	没有站登录 (诊断事件进入诊断缓冲冲击)
8083	INFO1 不允许的数据类型
8084	INFO2 不允许的数据类型
8085	EVENTV 不允许
8086	INFO1 长度不允许
8087	INFO2 长度不允许
8091	(这个故障代码仅在老版本的 CPU 中出现) 没有站登录 (诊断事件进入诊断缓冲区)
8092	当前不可能发送信息，发送缓冲区已满 (诊断事件进入诊断缓冲区)

## 13.5 用 SFC 78 “OB\_RT” 确定 OB 程序运行时间

### 描述

通过 SFC 78 “OB\_RT” 可以确定单个 OB 在不同时间段内的运行时间。

### 注意

SFC 78 返回所需 OB 的最近一次运行时间记录值，而不管此 OB 当前是否被装载。SFC 78 的数据不能被删除和覆盖，但在重启（暖启动）后会被复位。

### 参数

参数	声明	数据类型	存储区域	说明
OB_NR	INPUT	INT	I, Q, M, D, L	OB 号，查询此 OB 最近一次程序运行的时间，CPU 中所有组态的 OB 号都是有效的，但 OB121 与 OB122 除外。同步故障的处理时间包含在处理引起故障的 OB 所需的时间中，此处如使用 OB121 与 OB122，或者 CPU 中有未组态的 OB，会产生一条故障信息。 若 OB_NR=0 调用 SFC 78 的 OB 的数据会被送出，如 SFC 78 在 OB121 或 OB122 中调用，并且 OB_NR=0，中断触发的 OB 的所有时间数据会被送出，包括 OB12x 的时间数据。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如果功能执行时有故障发生，此参数包含一个故障代码。否则，RET_VAL 包含的是所需的 OB 号。
PRIO	OUTPUT	INT	I, Q, M, D, L	PRIO 中输出所查询 OB 的优先级
LAST_RT	OUTPUT	DINT	I, Q, M, D, L	指定 OB 最近一次执行所用的运行时间（用 ms 表示）。 如果欲查询运行时间的 OB 正在执行：在本 OB 执行过程中首次调用 SFC 78 后，LAST_RT 输出上次总的执行时间，后面再陆续调用 SFC 78，则 LAST_RT <ul style="list-style-type: none"> <li>• 指示 DW#16#FFFF FFFF，如果此 OB 已经调用了 SFC 78，并且设 OB_NR=0。</li> <li>• 指示上次执行时间，如果此 OB 没有调用 SFC 78，并且设 OB_NR=0。</li> </ul> 注意：由高优先级 OB 引起的中断时间未包含在 LAST_RT 中。

参数	声明	数据类型	存储区域	说明
LAST_ET	OUTPUT	DINT	I, Q, M, D, L	<p>时间间隔（用 ms 表示），指的是对于指定的上次完全执行的 OB，调用 SFC 78 与此 OB 执行完毕之间的时间间隔。</p> <p>如果此 OB 当前正在执行： 在此 OB 处理过程中首次调用 SFC 78，LAST_ET 指示上次调用 SFC 78 与 OB 执行完毕之间的时间间隔，以后再陆续调用 SFC 78，LAST_ET</p> <ul style="list-style-type: none"> <li>指示 DW#16#FFFF FFFF，如果此 OB 已调用了 SFC 78，并且 OB_NR=0。</li> <li>指示上次调用 SFC 78 与 OB 执行时完毕之间的时间间隔，如果此 OB 还未调用 SFC 78，且 OB_NR=0。</li> </ul> <p>注意：由高优先级 OB 引起的中断时间未包含在 LAST_ET 中。</p>
CUR_T	OUTPUT	DINT	I, Q, M, D, L	<p>指定的、当前运行的 OB 所需的时间（用 ms 表示的相对时间，如果指定的 OB 当前没有被执行 CUR_T=0）</p> <p>注意：系统时钟是一个计数器，计数值从 0 到 2.147.483.647（ms），计数值满后回 0。</p>
CUR_RT	INPUT	DINT	I, Q, M, D, L	<p>指定 OB：已执行的时间（ms）。如果此 OB 没有或还没有执行，则 CUR_RT=0。执行结束后，运行时间数据会在 LAST_RT 中显示，CUR_RT 被置为 0。</p> <p>注意：由于高优先级 OB 引起的中断时间不包括在 CUR_RT 中。</p>
CUR_ET	OUTPUT	DINT	I, Q, M, D, L	<p>从被指定的当前运行的 OB 开始执行到结束的时间（用 ms 表示的运行时间）。如此 OB 还未被执行 CUR_ET=0。执行后，运行时间数据会在 LAST_ET 中指示，CUR_ET 置为 0。</p> <p>注意：由高优先级 OB 引起的中断时间不包括在 CUR_ET 中。</p>
NEXT_ET	OUTPUT	DINT	I, Q, M, D, L	<p>如果当前的 OB 还没有执行完毕又出现新的调用，NEXT_ET 指出从现在开始到下次 OB 执行中间的时间间隔。（用 ms 表示）。如果除了当前排列的或处理的 OB 外不存在其它调用，则 NEXT_ET=0。</p> <p>WinLC RTX 没有使用此参数。</p> <p>注意：由高优先级 OB 引起的中断时间不包括在 NEXT_ET 中。</p>

所述的时间包括一套处理同步故障中断（OB121，OB122）的时间。

---

**注意**

如果在 OB\_NR 输入了一个 OB 号，此 OB 号是动态任务数据，OS 还没有调用相应的 OB，或者说此 OB 还未下载到 CPU，RET\_VAL 包含这个指定的 OB 号，PRIO 包含组态了的（缺省值，如果需要）指定 OB 的优先级，LAST\_RT 为 DW#16#FFFF FFFF。

---

**故障信息**

事件类别故障代码	说明
1 到 102	传送信息的 OB 号
W#16#8080	参数 OB_NR 包含一个非法值

## 13.6 用 SFC 87 “C\_DIAG” 诊断当前的连接状态

**描述**

可通过 SFC 87 “C\_DIAG” 确定所有的 S7 连接与所有容错的 S7 连接（或部分连接）的当前状态。

正确运用这些连接数据可以看出 S7 连接与容错 S7 连接是否失败，进一步讲还可以生成报告，送到操作与可视化系统。监视这些连接可以成为自动化系统和可视化系统的连接纽带，如同自动化系统与操作系统之间的连接一样。

---

**注意**

CPU 操作状态的改变：RUN→STOP→RUN，不影响组态的连接的状态。除非一个 H 站从一个冗余系统状态变到停止系统状态，备用 CPU 的部分容错连接会断开。

另一方面，电源掉电后，所有组态的连接将被重建，这样会改变连接状态。

在启动中或启动后首次调用 SFC 87，连接信息会不同，这取决于上次 CPU 的操作模式是 STOP 还是 POWER OFF。

---

### 操作模式

SFC 87 “C\_DIAG” 为异步 SFC 运行，也就是说，程序运行过程中可多次调用。

通过调用 SFC 87，设 REQ=1 来启用任务。

如果能立即执行此任务，SFC 87 中输出参数 BUSY 为 0。如果 BUSY 为 1，表示此任务仍在执行。

### 何时调用 SFC 87?

为了识别 S7 连接与容错 S7 连接是否失败，在开始的循环中断 OB 中调用 SFC 87。例如，每 10 秒钟操作系统调用一次。

因为连接状态通常是不会改变的，所以只于上次调用后状态改变时将连接数据拷贝到用户程序中是可以的（MODE=B#16#02 时调用，如下所述）。

### 怎样调用 SFC 87?

SFC 87 “C\_DIAG” 提供 4 种可能的操作模式，见下表说明。

MODE (B#16#...)	SFC 拷贝连接数据到用户程序	SFC 把确认的信息传送到操作系统
00	否	是
01	是	是
02	<ul style="list-style-type: none"> <li>• 是，如果连接数据改变</li> <li>• 不，如果连接数据不改变</li> </ul>	是
03	是	否

自上次调用 SFC 87（MODE=B#16#00，01 或者 02）后连接数据的状态改变，通过将确认的信息传送到操作系统得到确认。

### 注意

如果在循环中断 OB 中调用 SFC 87 使用“条件拷贝”模式（MODE=B#16#02），必须保证 CPU 冷启动后目标区域内不包含初始化数据。这可以通过下面方法来实现：在 OB102 中调用一次 SFC 87，采用“带确认的无条件拷贝”模式（MODE=B#16#01）。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	请求激活的控制参数。 REQ=1: 如果还未启动, 则任务启动。
RET_VAL	INPUT	BYTE	I, Q, M, D, L, 常数	任务设计 可能值: <ul style="list-style-type: none"> <li>• B#16#00: 此 SFC 不拷贝连接数据, 只传给操作系统确认信息。</li> <li>• B#16#01: 不管状态是否改变, 此 SFC 都将所有的连接数据拷贝到用户程序, 并且传给操作系统确认信息。</li> <li>• B#16#02: 如果连接数据改变, SFC 将其拷贝到用户程序, 如没有改变, 则不拷贝, 两种情况下均传给操作系统确认信息。</li> <li>• B#16#03: 此 SFC 拷贝连接数据到用户程序, 与状态是否改变无关, 不向操作系统传送确认信息。</li> </ul>
RET_VAL	OUTPUT	INT	I, Q, M, D, L	返回值 (故障代码或者工作状态)
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: 任务还未完成
N_CON	OUTPUT	INT	I, Q, M, D, L	当 DIS_PCON 或 DIS_CON 为 TRUE 时 参数 CON_ARR 上次结构的标签号在用户程序中只有 CON_ARR 中的第一个 N_CON 参数需要检查。 注: CON_ARR 中第一个结构标签号为 1。
CON_ARR	OUTPUT	ANY	I, Q, M, D, L	接收连接数据的目标区域。 允许使用的数据类型只有“BYTE”。 每个连接都分配一个结构。 为 CPU 选择一个目标区域长度, 以便可以接收最大数目的可能连接的所有结构事件。

### 组织目标区域 CON\_ARR

读目标区域为一个结构域, 每个连接都分配一个结构。

开始时, 域没有被有效数据占满, 在两个有效的数据之间可能包含一个无效数据。

连接不被连接参照排序。

#### 注意

从操作系统拷贝连接数据到一个指定的目标区域时, 确保数据的一致性。

## 结构组织

参数	数据类型	说明
CON_ID	WORD	在 NETPRO 中分配给本次连接的连接参照。 W#16#FFFF: 无效设计, 也就是说, 此连接未被组态。如果 CON_ARR[i].DIS_PCON 或 CON_ARR[i].DIS_CON (如下所述) 被置位, 表示此连接已被重新组态或者自上次调用 SFC 87 后被删除。
STAT_CON	BYTE	S7 连接或者冗余 S7 连接的当前状态。 可能值: <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: S7 连接正在建立</li> <li>• B#16#11: 冗余 S7 连接正在建立</li> <li>• B#16#02: S7 连接已经建立</li> <li>• B#16#12: 冗余 S7 连接已建立, 但不冗余</li> <li>• B#16#13: 冗余 S7 连接已建立</li> </ul>
PROD_CON	BYTE	运行时间连接的部分连接号。 可能值: 0, 1, 2, 4
STBY_CON	BYTE	冗余连接的部分连接号 (B#16#FF: 无备用连接) 可能值: 0, 1, 2, 3 注意: 只有冗余 S7 连接可以有备用连接。
DIS_PCON	BOOL	CON_ARR[i].STAT_CON 由 W#16#12 变为 W#16#13, W#16#13 变为 W#16#12, 在最近一次调用 SFC 87 把 CON_ARR[i].DIS_PCON 置位 1 后。所有其它连接状态的改变都不影响 CON_ARR[i].DIS_PCON。 注意: <ul style="list-style-type: none"> <li>• 若 MODE=B#16#01 与 02, 响应 DIS_PCON 的操作系统位在连接数据拷贝到目标区域时被复位。</li> <li>• 若 MODE=B#16#03, 响应 DIS_PCON 的操作系统位保持不变。</li> </ul>
RES0	BYTE	备用 (B#16#00)
RES1	BYTE	备用 (B#16#00)

故障信息

故障代码 (W#16#...)	说明
0000	<ul style="list-style-type: none"> <li>MODE=B#16#00, 01 或 02: 自上次调用后连接状态没有改变 (结构元素 STAT_CON) 调用无故障运行。</li> <li>MODE=B#16#03: 拷贝任务无故障执行。</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状态改变 (结构元素 STAT_CON) 任务无故障执行。</li> <li>MODE=B#16#03: 不允许 RET_VAL 为 W#16#0001。</li> </ul>
7000	REQ=0, 首次调用, MODE 中指定的任务不能执行, BUSY 为 0。
7001	REQ=1, 首次调用, MODE 中指定的任务开始执行, BUSY 为 1。
7002	中间调用 (与 REQ 无关), 任务仍在执行, BUSY 为 1。
8000	参数 MODE 中非法数值。
8001	参数 CON_ARR 中包含非法数据类型。
8002	参数 CON_ARR 声明的长度太短。SFC 无法拷贝数据到目标区域。

## 13.7 用 SFC 103 “DP\_TOPOL” 识别 DP 主站系统的总线拓扑结构

### 描述

调用 SFC 103 “DP\_TOPOL” 用它的诊断重复器可以开始识别选定的 DP 主站系统的总线拓扑结构。所有连接到 DP 主站系统的诊断重复器均需在 SFC 103 被调用时指定地址。

### 注意

在一个给定的时间只可以识别一个 DP 主站系统的拓扑结构。

拓扑结构的识别是当一个线路故障事件发生时, 进行详细故障显示的前提。当一个 DP 主站中的物理结构发生安装上的或其它任何改变后, 都需用 SFC 103 重新识别拓扑结构一次。

物理结构的改变包括:

- 电缆长度改变
- 通过重复器功能增加或减少站点或元件
- 改变站点地址

每个诊断重复器都把结果 (所有站点的 PROFIBUS 地址和到诊断重复器的绝对距离) 送入拓扑结构表中。可以在 STEP 7 中读出这些值。

推荐在 OB1 或一个监控中断 OB 中调用 SFC 103。S7-300 不允许在一个启动 OB 中调用 SFC 103。



此 SFC 把诊断中继器返回的故障信息写入输出参数 DPR 与 DPRI。如果在选定的 DP 主站系统中不止一个诊断中继器发出故障信息，SFC 只把与第一次生成故障信息的诊断重复器有关的信息写入 DPR 与 DPRI。调用 SFC 13 “DPNRM\_DG” 或者 STEP 7 读取完整的诊断信息，如果诊断中继器没有报故障则 DPR 与 DPRI 的值为 NULL。

### 运行方式

SFC 103 “DP\_TOPOL” 异步执行，即它的执行需多次调用。调用 SFC 103，令 REQ=1，则开始识别 DP 总线拓扑结构。调用 SFC 103，令 R=1，则退出识别。

任务的状态由参数 RET\_VAL 与 BUSY 指示，参见异步执行的 SFC 中 REQ、REQ\_VAL 与 BUSY 的说明。

### 注意

识别拓扑结构可能需要 n 分钟的时间。

### 任务标识

输入参数 DP\_ID 确定一个请求。

如果在拓扑结构识别完成之前再次调用 SFC 103 “DP\_TOPOL”。SFC 的响应取决于新的请求与原来的是否相同；如果参数 DP\_ID 响应的任务还未结束，此 SFC 的调用被看作是一个调用顺序的一部分，并在 RET\_VAL 中写入 W#16#7002。如果是一个新任务，CPU 不予理会。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	REQ=1: 开始识别总线拓扑
R	INPUT	BOOL	I, Q, M, D, L, 常数	R=1: 退出识别
DP_ID	INPUT	INT	I, Q, M, D, L, 常数	进行拓扑结构识别的 DP 主站系统的 ID
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能执行时若发生故障，此返回值中包含一个故障代码
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: 总线拓扑结构识别仍在进行
DPR	OUTPUT	BYTE	I, Q, M, D, L	已生成故障信息的诊断重复器的 PROFIBUS 地址。
DPRI	OUTPUT	BYTE	I, Q, M, D, L	已生成故障信息的诊断重复器的区段： <ul style="list-style-type: none"> <li>• 位 0=1: 故障在 DP2 段</li> <li>• 位 1=1: 故障在 DP3 段</li> <li>• 位 4=1: DP3 段暂时崩溃</li> <li>• 位 5=1: DP3 段永久崩溃</li> </ul>
<p><b>永久故障:</b> 系统检测到阻止拓扑结构识别顺利进行的永久故障。可以通过调用 SFC 13 “DPNRM_DG” 或者 STEP 7 获得详细的诊断信息。</p> <p><b>临时故障:</b> 系统检测到阻止拓扑结构识别顺利进行的临时故障。可能是一个连接故障或一个重复出现的故障。这些故障的本质不允许故障源精确定位。</p>				

### 故障信息

观察下表中的“真实的”故障信息（故障代码 W#16#8xyz），可以分为两类故障：

- 临时故障（故障代码 W#16#80A2~80A4, 80C3, 80C5）：  
这些故障可能不需用户清除，即重新调用此 SFC（多次调用，如果需要）可能会有帮助。  
例如一个临时故障：系统资源正被占用（W#16#80C3）。
- 永久故障（故障代码 W#16#8082, 80B0, 80B2）：  
这些故障不会自己消失，只能在解决故障后调用此 SFC。例如一个永久故障：DP 主站/CPU 不支持此任务（W#16#80B0）。

故障代码 W#16#...	说明
0000	任务顺利完成
7000	REQ=0 开始调用，总线拓扑结构识别未开始，BUSY=0。
7001	REQ=1 开始调用，总线拓扑结构识别开始，BUSY=1。
7002	中间调用（与 REQ 无关）：总线拓扑结构识别仍在开始，BUSY=1。
7010	已请求取消识别，没有进行指定 DP_ID 的识别任务，BUSY=1。
7011	开始调用，令 R=1。取消拓扑结构识别开始，BUSY=1。
7012	最后调用：拓扑结构识别已取消。BUSY=0。
7013	中间调用：拓扑结构识别取消还未完成。BUSY=1。
8082	指定 DP_ID 的 DP 主站系统未被组态。
80A2	拓扑结构识别过程中发生故障：参考输出参数 DPR 和 DPRI 获取详细信息。
80A3	拓扑结构识别过程中发生故障：监控时间到。
80A4	K 总线通讯故障。
80B0	DP 主站/CPU 不支持此任务。
80B2	拓扑结构识别过程中发生故障：在选定的 DP 主站系统中找不到诊断重复器
80C3	系统资源正被占用。可能原因：开始了第 2 个识别任务（在给定的时间只允许进行一个总线拓扑结构识别）或者 H CPU 正在执行链接或升级功能。
80C5	DP 主站系统当前不能用。

## 14 用于刷新过程映象和处理位区域的 SFC 与 SFB

### 14.1 用 SFC 26 “UPDAT\_PI” 刷新过程映象输入表

#### 描述

通过 SFC 26 “UPDAT\_PI” (刷新过程映象), 可以刷新 OB1 的过程映象输入表 (= 过程映象区 0) 或者一个在 STEP 7 中定义的过程映象输入区。

若用 I/O 访问故障重复发出的信号组态系统过程映象表刷新, 则选定的过程映象表被 SFC 26 连续刷新。

否则, SF 626 将只刷新选定的但没被系统刷新的过程映象表, 也就是:

- 如果没有把此过程映象区分配给中断 OB, 或者
- 选定了过程映象区 0 但组态为禁止 OB1 过程映象区刷新。

---

#### 注意

通过 STEP 7 分配给一个输入过程映象区的逻辑地址不再属于 OB1 的过程映象输入表。

如果采用了 SFC 26 刷新一个输入过程映象区, 就不能再用 SFC 126 “SYNC\_PI” 去刷新它。

---

SFC 26 的调用不影响 OB1 过程映象输入表和分配给中断 OB 的过程映象输入区的刷新。

参数	声明	数据类型	存储区域	说明
PART	INPUT	BYTE	I, Q, M, D, L, 常数	被刷新的过程映象区号, 最大值范围 (取决于 CPU): 0~15 (0 表示 OB1 的过程映象, n 当 1<n<15 时表示过程映象区 n)。
RET_VAL	OUTPUT	INT	I, Q, M, D, L, 常数	故障信息
FLADDR	OUTPUT	WORD	I, Q, M, D, L	发生 I/O 访问故障时引起故障的地址的第一个字节

#### 故障信息

故障代码 (W#16#...)	说明
0000	无故障发生。
8090	输入参数 PART 的值非法。
8091	指定的过程映象区未被定义或者不在 CPU 允许的过程映象区域内。
8092	过程映象区在系统执行 OB 时被刷新, 没有组态所有的 I/O 访故障重复发信号, 此过程映象区不能被 SFC 26 “UPDAT_PI” 刷新。
80A0	在刷新过程中检测到一个访问故障。

**注意**

如果采用了 SFC 26 “UPDAT\_PI” 刷新 DP 标准从站的过程映象区，而给这个 DP 从站定义了一个超过 32 字节的区域，就可能从 SFC 14 “DPRD\_DAT” 中产生一个故障代码。

## 14.2 用 SFC 27 刷新过程映象输出表

**描述**

通过 SFC 27 “UPDAT\_PI” (刷新过程输出)，可以把 OB1 的过程映象输出表 (=过程映象区 0) 或者一个通过 STEP 7 中定义的过程映象输入区的信号状态传送到输出模板。如果给部分过程映象指定了一个相应的范围，那么相应的数据就会传送到相应的外设模板。

**注意**

通过 STEP 7 分配给一个输出过程映象区的逻辑地址不再属于 OB1 的过程映象输出表。如果采用 SFC 27 刷新一个输出过程映象区，就不能再用 SFC 127 “SYNC\_PO” 刷新它。

SFC 27 的调用不影响 OB1 过程映象输出表与分配给中断 OB 的过程映象输出区的信号传输。

参数	声明	数据类型	存储区域	说明
PART	INPUT	BYTE	I, Q, M, D, L 常数	被刷新的过程映象区号，最大值范围 (取决于 CPU): 0~15 (0 表示 OB1 的过程映象, n 当 1≤n≤15 时表示过程映象区 n)。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
FLADDR	OUTPUT	WORD	I, Q, M, D, L	发生 I/O 访问故障时引起故障的地址的第一个字节

**故障信息**

故障代码 (W#16#...)	说明
0000	无故障发生。
8090	输入参数 PART 的值非法。
8091	指定的过程映象区未被定义或者不在 CPU 允许的过程映象区域内。
80A0	在刷新过程中检测到一个访问故障。

**注意**

如果采用了 SFC 27 “UPDAT\_PO” 刷新 DP 标准从站的过程映象区，而给这个 DP 从站定义了一个超过 32 字节的区域，就可能从 SFC 15 “DPWR\_DAT” 中产生一个故障代码。

## 14.3 用 SFC 126 “SYNC\_PI” 同步刷新过程映象区输入表

### 描述

通过 SFC 126 “SYNC\_PI” 可以在一个同步周期内刷新过程映象输入表，链接到一个 DP 周期的用户程序可以通过此 SFC 连续一致地同步地刷新一个过程映象区的输入数据。

SFC 26 可以被中断并且只能在 OB 61、62、63 和 64 中调用。

为保证 SFC 126 可靠执行，下列条件应该满足：

- DP 时钟周期 > 约 5.0ms
- 从站刷新时间 < DP 时钟周期 - 4.0ms

### 注意

只有在硬件组态中给相关 OB 分配了相应过程映象区才允许在 OB 61~64 中调用 SFC 126 “SYNC\_PI”。

若采用 SFC 126 刷新一个过程映象输入表，就不能再用 SFC 26 “UPDAT\_PI” 刷新它。



### 告诫

避免直接访问（L PEB 命令）正在被 SFC 126 刷新的过程映象区域。

如果违反此规定，会收不到当前的数据。

### 参数

参数	声明	数据类型	存储区域	缺省	说明
PART	INPUT	BYTE	1~30		在一个同步周期内被刷新的过程映象区输入表号
RET_VAL	OUTPUT	INT			故障信息
FLADDR	OUTPUT	WORD			访问故障发生时引起故障的地址的第一个字节

### 故障信息

事件种类 故障代码	说明
W#16#8090	参数 PART 的值非法或者指定了 OB 不允许的过程映象区输入表。此过程映象输入表没有被刷新。
W#16#8091	指定的过程映象区还未被定义或者不在 CPU 可访问的过程映象区域内。此过程映象输入表未被刷新。
W#16#80A0	在刷新过程中检测到一个访问故障，受到影响的输入被设为“0”。
W#16#80A1	刷新时间在允许访问的窗口之后。此过程映象区输入表未被刷新。
W#16#80C1	刷新时间在允许访问的窗口前。过程映象区输入表未被刷新。

**注意**

如果用 SFC 126 “SYNC\_PI” 刷新 DP 标准从站的过程映象区，并且定义了一个大于 32 字节的区域，在 SFC 14 “DPRD\_DAT” 中同样得到一个故障代码。

## 14.4 用 SFC 127 “SYNC\_PO” 同步刷新过程映象区输出表

**描述**

通过 SFC 127 “SYNC\_PO” 可以在一个同步周期内刷新一个过程映象输出表，链接到一个 DP 周期的用户程序可以通过此 SFC 同步地刷新位于过程映象区内的输出数据和连续地把输出数据发送的 I/O 装置。

SFC 127 可被中断并且只能在 OB 61、62、63 与 64 中调用。

为保证 SFC 126 可靠执行。下列条件应该满足：

- DP 时钟周期 > 约 5.0ms
- 从站刷新时间 < DP 时钟周期 - 4.0ms

**注意**

只有在硬件组态中给相关 OB 分配了相应过程映象区才允许在 OB 61~64 中调用 SFC 127 “SYNC\_PO”。

若采用 SFC 127 刷新过程映象区，就不能再用 SFC 27 刷新它。



**告诫**

避免直接访问（L PEB 命令）正在被 SFC 127 刷新的过程映象区。

如果违反此规定，写操作会失败。

**参数**

参数	声明	数据类型	存储区域	缺省	说明
PART	INPUT	BYTE	1~30		在一个同步周期内被刷新的过程映象区输出表号
RET_VAL	OUTPUT	INT			在功能执行时若发生故障，此返回值中包含一个故障代码。
FLADDR	OUTPUT	WORD			引起访问故障的地址的第一个字节

## 故障信息

事件种类 故障代码	说明
W#16#0001	连续报警，过程映象区表的刷新分配在 2 个 DP 周期。一个从站的数据被连续发送。
W#16#8090	参数 PART 的值非法或者指定了 OB 不允许的过程映象区输出表。输出没被传送到 I/O 装置。过程映象区输出表没有改变。
W#16#8091	指定的过程映象区还未被定义或者不在 CPU 可访问的过程映象区域内。输出没被传送到 I/O 装置。过程映象区输出表没有改变。
W#16#80A0	在刷新过程中检测到一个访问故障，输出没被传送到 I/O 装置。过程映象区输出表没有改变。
W#16#80A1	刷新时间在允许访问窗口之后或者输出数据没有被 DP 主站刷新。输出没被传送到 I/O 装置。过程映象区输出表没有改变。
W#16#80C1	刷新时间在允许访问的窗口前，输出没被传送到 I/O 装置。过程映象区输出表没有改变。

## 注意

如果用 SFC 127 “SYNC\_PO” 刷新 DP 标准从站的过程映象区，并且定义了一个大于 32 字节的区域，在 SFC 15 “DPWR\_DAT” 中同样得到一个故障代码。

## 14.5 用 SFC 79 “SET” 置位 I/O 区域中的位区域

## 描述

调用 SFC 79 “SET”（置位输出范围）会发生下列影响：

- 由参数 N 与 SA 选定的外设 I/O 区域内的位区域被置位。
- 过程映象输出表中相应的位也被置位，而不管这些位是否位于过程映象区，位区域必须是分配给过程映象的外设 I/O 区域的一部分。

如果选定的位区域的一部分没有插模板，SFC 79 仍企图把整个位区域置位。

这时会在 RET\_VAL 中返回一个相应的故障信息。

## 注意

SFC 79 执行时总是把整个字节写入 I/O 区域。

如果参数 N 与 SA 选定的位区域不是以一个字节开始或结束，SFC 79 的调用会产生下列影响：

- 传送到外设 I/O 区域的第一个与最后一个字节中不属于选定的位区域的位包含过程映像输出表中相应位的数值，这会导致不希望的动作发生。例如电机启动或冷却系统关断。
- 属于选定位区域中的位如前面所述被置位。

如果把参数 N 设为 0，SFC 79 的调用不起作用，如果主控继电器未合，SFC 79 的调用不起作用。

参数	声明	数据类型	存储区域	说明
N	INPUT	INT	I, Q, M, D, L, 常数	被置位的位的数目
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
SA	OUTPUT	POINTER	P	被置位的第一个位的指针

#### 故障信息

通过输出参数 RET\_VAL 分析故障一节说明了怎样分析 RET\_VAL 中的故障信息。本章亦包含 SFC 一般的故障信息。SFC 79 的参数 RET\_VAL 不提供特殊故障的信息。

## 14.6 用 SFC 80 “REST” 复位 I/O 区域中的位区域

### 描述

调用 SFC 80 “RSET”（复位输出范围）会产生下列影响：

- 由参数 N 与 SA 选定的外设 I/O 区域内的位区域被复位。
- 过程映像输出表中相应的位也被置位，而不管这些位是否位于过程映像位区域必须是分配给过程映像的外设 I/O 区域的一部分。

如果选定的位区域的一部分没有插模板，SFC 80 仍企图复位整个位区域，这时会在 RET\_VAL 中返回一个相应的故障信息。

---

#### 注意

SFC 80 执行时，总是把整个字节写入外设 I/O 区域。

---

如果参数 N 与 SA 选定的位区域不是以一个字节开始或结束，SFC 80 的调用会产生下列影响：

- 传送到外设 I/O 区域的第一个与最后一个字节中不属于选定的位区域的位包含过程映像输出表中相应位的数值，这会导致不希望的动作发生例如电机启动或冷却系统关断。



- 属于选定位区域中的位复位，如前面所述。

如果把参数 N 设为 0，SFC 80 的调用不起作用，如果主控继电器未合，SFC 80 的调用不起作用。

参数	声明	数据类型	存储区域	说明
N	INPUT	INT	I, Q, M, D, L, 常数	被复位的位的数目
RET_VAL	OUTPUT	INT	I, Q, M, D, L,	故障信息
SA	OUTPUT	POINTER	P	被复位的第一个位的指针

#### 故障信息

通过输出参数 RET\_VAL 分析故障一节说明了怎样分析 RET\_VAL 中的故障信息。本章亦包含 SFC 的一般故障信息。SFC 80 的参数 RET\_VAL 不提供特殊故障的信息。

## 14.7 用 SFB 32 “DRUM” 执行一段顺序程序

#### 描述

SFB 32 “DRUM” 可执行一段多达 16 步的顺序程序。在参数 DSP 中指定第一步的编号，在参数 LST\_STEP 中指定最后一步的编号。

在每一步中，所有的 16 个输出位 OUT0~OUT15 以及输出参数 OUT\_WORD（这个参数把各输出位收集在一起）被写。一个输出位不是被指定的 OUT\_VAL 数组中相应的位赋值，就是被前面步骤中相应的输出位赋值。到底赋予其哪个值取决于在参数 S\_MASK（见下表）如何设置屏蔽位。

当与先前 SFB 调用相比在输入 JOG 中出现一个上升沿时，SFB 32 “DRUM” 切换到下一步。如果此 SFB 执行到最后一步，JOG 中出现上升沿则变量 Q 与 EOD 被置位，DCL 为 0；此 FB 一直保持在最后一步，直到 RESET 输入被置位。

也可以配置参数以使下一步的由时间决定。为了做到这一点，必须把参数 DRUM\_EN 设为 1。程度切换到下一步，当：

- 事件位 EVENT 在当前步中置位并且
- 当前步的执行时间已到

此处的时间是 DTBP 时基与当前步有效的的时间系数（来自 S\_PRESET 数组）的乘积。

---

#### 注意

当前步骤（DCC）的剩余时间只有在相应的事件位 EVENT 置位的情况下才会减少。

---

如果在 SFB 调用时 RESET 设为 1，则顺序程序会去执行 DSP 中输入步骤。

**注意**

如把 DRUM\_EN 设为 1，会出现下面的特殊情况：

- 选定 EVENT=1，在  $DSP \leq i \leq LST\_STEP$  时成为一个单纯的时间控制程序。
- 使用时间控制位 EVENT，并设 DTBP 为 0 则成为一个单纯的事件控制程序。

还可以通过 JOG 的输入在任意时间（甚至在 DRUM\_EN=1 时）转到程序中的下一步。

此块被第一次调用时，RESET 必须为 1。

如果顺序程序已执行到最后一步（DSC=LST\_STEP），并且最后一步的执行时间已到，则输出 Q 与 EOD 被置位，直到 RESET 为 1，SFB 都将保持在最后一步。

DRUM 定时器只在 STARTUP 与“RUN”模式下运行。

操作系统在冷启动时，复位 SFB 32“DRUM”，在暖启动时则不然。如果在暖启动后初始化 SFB“DRUM”，应在 OB100 中调用它并使 RESET 为 1。

参数	声明	数据类型	存储区域	说明
RESET	INPUT	BOOL	I, Q, M, D, L, 常数	为 1 时复位顺序程序，第一次调用此块，必须令 RESET 为 1。
JOG	INPUT	BOOL	I, Q, M, D, L, 常数	上升沿（参照最后一次 SFB 调用）出现时如果程序还没到最后一步，则会切换到下一步。下一步的使能由参数 DRUM_EN 的值决定。
DRUM_EN	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数：取决于时间的切换是否允许（1：取决于时间的切换可能）。
LST_STEP	INPUT	BYTE	I, Q, M, D, L, 常数	最后步的编号，可能值：1~16
EVENT <sub>i</sub> , 1 ≤ i ≤ 16	OUTPUT	BOOL	I, Q, M, D, L, 常数	事件位号 i（属于步 i）
OUT <sub>j</sub> , 0 ≤ j ≤ 15	OUTPUT	BOOL	I, Q, M, D, L, 常数	输出位号 j（与 OUT_WORD 中位号 j 相同）
Q	OUTPUT	BOOL	I, Q, M, D, L	状态参数：选定的最后一步的执行时间是否已到。
OUT_WORD	OUTPUT	WORD	I, Q, M, D, L, P	把所有的输出位集中在一起的变量
ERR_CODE	OUTPUT	WORD	I, Q, M, D, L, P	在执行此 SFB 时若发生故障，ERR_CODE 中包含故障信息。
JOG_HIS	VAR	BOOL	I, Q, M, D, L, 常数	（与用户无关，先前的 SFB 调用的 JOG 输入参数）
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, 常数	第一步的编号，可能值：1~16
DSC	VAR	BYTE	I, Q, M, D, L, 常数	当前步的编号

参数	声明	数据类型	存储区域	说明
DCC	VAR	BOOL	I, Q, M, D, L, 常数	当前步剩余的执行时间, 用 ms 表示 (只在 DRUM_EN=1 和相应的事件位为 1 时有意义)
DTBP	VAR	WORD	I, Q, M, D, L, P, 常数	对于所有步都有效的时间基值, 用 ms 表示
PREV_TIME	VAR	DWORD	I, Q, M, D, L, 常数	(与用户无关: 先前 SFB 调用的系统时间)
S_PRESET	VAR	ARRAY of WORD	I, Q, M, D, L, 常数	包含每一步的时间系数的一维数组、标签的可能选择为: [1~16]。这种情况下, S_PRESET[x]中包含第 x 步的时间系数。
OUT_VAL	VAR	ARRAY of BOOL	I, Q, M, D, L, 常数	包含每一步的输出值的二维数组, 如果这些输出没有用 S_MASK 屏蔽掉, 标签的可能选择为 [1~16, 0~15]。这种情况下, OUT_VAL[x, y]中包含第 x 步分配给输出位 OUTY 的数值。
S_MASK	VAR	ARRAY of BOOL	I, Q, M, D, L, 常数	包含每一步屏蔽位的二维数组、标签的可能选择为: [1~16, 0~15]。这种情况下, S_MASK[x, y]中包含第 x 步, 第 y 个输出值的屏蔽位。 屏蔽位的含义: <ul style="list-style-type: none"> <li>• 0: 先前步的输出值赋值给相应的输出位</li> <li>• 1: OUT_VAL 中相应的值赋值给相应的输出位</li> </ul>

## 故障信息

如果下表所列的任一种情况发生, SFC 32 “DRUM” 将保持在当前状态, ERR\_CODE 输出被置位。

故障代码 (W#16#...)	说明
0000	无故障
8081	LST_STEP 值非法
8082	DSC 值非法
8083	DSP 值非法
8084	乘积 $DCC=DTBP*S\_PRESET[DSC]$ 超过了 $2^{32-1}$ 约为 24.86 天



## 15 用于模板寻址的系统功能

### 15.1 用 SFC 5 “GADR\_LGC” 查询模板的逻辑起始地址

#### 描述

依据信号模板的通道，相应模板的槽号以及模板的偏置用户数据地址区域就被确定了。用 SFC 5 “GADR\_LGC”（把地理地址转换为逻辑地址），可以得到模板的逻辑地址，例如，最小的输入输出地址。

参数	声明	数据类型	存储区域	说明
SUBNETID	INPUT	BYTE	I, Q, M, D, L, 常数	区域标识： <ul style="list-style-type: none"> <li>0, 槽位于 0 号机架（中央机架）或者 1~21（扩展机架）</li> <li>插槽位于分布式 I/O 设备上时相应分布式 I/O 系统的 DP 主站 ID。</li> </ul>
RACK	INPUT	WORD	I, Q, M, D, L, 常数	<ul style="list-style-type: none"> <li>若区域标识为 0, 指机架号</li> <li>若区域标识&gt;0, 指分布式 I/O 设备的站号</li> </ul>
SLOT	INPUT	WORD	I, Q, M, D, L, 常数	槽号
SUBSLOT	INPUT	BYTE	I, Q, M, D, L, 常数	子模板槽（如果没有子模板插入，此值必须指定为 0）
SUBADDR	INPUT	BYTE	I, Q, M, D, L, 常数	模板的用户数据地址区域的偏置
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
IOID	OUTPUT	BYTE	I, Q, M, D, L	区域标识 B#16#54: 外设输入（PI） B#16#55: 外设输出（PQ） 对于混合模板，此 SFC 给出较低地址的区域标识，如果地址相同，此 SFC 给出标识 B#16#54。
LADDR	OUTPUT	WORD	I, Q, M, D, L	模板的逻辑起始地址

故障信息

故障代码 (W#16#...)	说明
0000	无故障发生
8094	指定的 SUBNETID 没有组态子网络
8095	参数 RACK 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	槽未组态
809A	选定槽的写地址未组态

## 15.2 用 SFC 49 “LGC\_GADR” 查询逻辑地址所属的插槽

描述

通过 SFC 49 “LGC\_GADR”（把逻辑地址转换为地理地址），可以得到逻辑地址所属的插槽与在模板用户数据地址区域中的偏置。

参数	声明	数据类型	存储区域	说明
IOID	INPUT	BYTE	I, Q, M, D, L, 常数	地址区域的 ID: <ul style="list-style-type: none"> <li>W#16#00: LADDR 的位 15, 指出是一个输入地址 (位 15=0) 还是输出地址 (位 15=1) 存在</li> <li>W#16#54=外设输入 (PI)</li> <li>W#16#55=外设输出 (PQ)</li> </ul> 如果模板为混合模板, 指出最低地址的区域 ID。如地址相同, 指示 B#16#54。
LADDR	INPUT	WORD	I, Q, M, D, L, 常数	逻辑地址, 对于混合模板, 指两个地址中较低的一个。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
AREA	OUTPUT	BYTE	I, Q, M, D, L	区域 ID: 指出后面输出参数的不同含义
RACK	OUTPUT	WORD	I, Q, M, D, L	机架号
SLOT	OUTPUT	WORD	I, Q, M, D, L	槽号
SUBADDR	OUTPUT	WORD	I, Q, M, D, L	相应模板用户数据地址区域内的偏置

## 输出参数 AREA

输出参数 AREA 指出了如何解释输出参数 RACK, SLOT 与 SUBADDR (见下表)。

AREA 的值	系统	RACK, SLOT 与 SUBADDR 的含义
0	S7-400	RACK: 模板所在的机架号 SLOT: 槽号 SUBADDR: 逻辑地址与逻辑起始地址的区别
1	S7-300	RACK: 模板所在的机架号 SLOT: 槽号 SUBADDR: 逻辑地址与逻辑起始地址的区别
2	DP	RACK: (低字节) 站号 RACK: (高字节) DP 主站 ID SLOT: 在此站内的槽号 SUBADDR: 相应模板用户数址区域内的偏置
3	S5 P 区域	RACK: 模板所在的机架号 SLOT: 适配板的槽号 SUBADDR: S5 中的地址 x 范围
4	S5 O 区域	RACK: 模板所在的机架号 SLOT: 适配板的槽号 SUBADDR: S5 中的地址 x 范围
5	S5 IM3 区域	RACK: 模板所在的机架号 SLOT: 适配板的槽号 SUBADDR: S5 中的地址 x 范围
6	S5 IM4 区域	RACK: 模板所在的机架号 SLOT: 适配板的槽号 SUBADDR: S5 中的地址 x 范围

## 故障信息

故障代码 (W#16#...)	说明
0000	无故障发生
8090	指定的地址无效或参数 IOID 的值非法

## 15.3 用 SFC 50 “RD\_LGADR” 查询一个模板所有的逻辑地址

### 描述

从模板的一个逻辑地址开始，通过 SFC 50 “RD\_LGADR”（读模板逻辑地址），可以得到这块模板所有声明的逻辑地址。先前已通过 STEP 7 给这块模板分配了地址，SFC 50 把得到的逻辑地址升序排列存入域 PEADDR 或 PAADDR 中。

参数	声明	数据类型	存储区域	说明
IOID	INPUT	BYTE	I, Q, M, D, L, 常数	区域标识： <ul style="list-style-type: none"> <li>W#16#00: LADDR 的位 15 指出是一个输入（位 15=0）还是输出（位 15=1）地址存在</li> <li>W#16#54=外设输入（PI）</li> <li>W#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	故障信息
PEADDR	OUTPUT	ANY	I, Q, M, D, L	PI 地址区域，区域元素的数据类型必须是 WORD
PECOUNT	OUTPUT	INT	I, Q, M, D, L	返回的 PI 地址数目
PAADDR	OUTPUT	ANY	I, Q, M, D, L	PQ 地址区域，区域元素的数据类型必须是 WORD
PACOUNT	OUTPUT	INT	I, Q, M, D, L	返回的 PQ 地址数目

### 故障信息

故障代码 (W#16#...)	说明
0000	无故障发生
8090	指定的逻辑地址无效或 IOID 参数的值非法
80A0	输出参数 PEADDR 错误：区域元素的数据类型不是 WORD
80A1	输出参数 PAADDR 错误：区域元素的数据类型不是 WORD
80A2	输出参数 PEADDR 错误：指定的区域容纳不下所有的逻辑地址
80A3	输出参数 PAADDR 错误：指定的区域容纳不下所有的逻辑地址



## 16 分布式 I/O 使用的 SFC

### 16.1 用 SFC 7 “DP\_PRAL” 在 DP 主站上触发硬件中断

#### 描述

在智能从站用户程序中调用 SFC 7 “DP\_PAAL”，可以触发 DP 主站上的硬件中断，硬件中断意味着 DP 主站将执行一次 OB40 中的程序。

通过输入类型参数 “AL\_IVFO”，可以标识触发硬件中断的原因，这个中断标识符号将被传送到 DP 主站，在 OB40 中用户可以访问这个标识符号（变量 OB40\_POINT\_ADDR）。

通过参数 “IOID” 和 “LADDR” 保证了硬件中断请求的唯一性——对于每一个已配置的传输区域地址，任何时候反能触发一个硬件中断。

#### SFC 执行过程

SFC 7 “DP\_PRAL” 是非同步执行的，也就是说，这个块要执行多个 SFC 调用周期。调用 SFC 7 时通过设定参数 “REQ=1” 就可以启动硬件中断申请。

SFC 的执行状态通过输出类型参数 “RET\_VAL” 和 “BUSY” 给出。参数 “REQ”、“RET\_VAL” 和 “BUSY” 具体含义见后面内容，当主站硬件中断标识块 OB40 执行结束时，表示智能从站触发的硬件中断作业完成。

---

#### 注意

如果 DP 从站是作为标准从站，只需 DP 主站得到诊断信息，则从站触发的硬件中断作业完成。

---

#### 标识硬件中断作业

输入类型参数 “IOID” 和 “LADDR” 决定了唯一的硬件中断作业。当 DP 从站中用户程序已经调用了 SFC 7 “DP\_PRAL”，然而在 DP 主站确认这个硬件中断请求之前，用户程序却再次调用了 SFC 7 “DP\_PRAL”，此时 SFC 7 的反应主要看新调用的是否为同一个作业。若新的调用中参数 “IOID” 和 “LADDR” 与未完成的作业相同。则新的调用被视作上一个调用的继续，参数 “AL\_INFO” 的值无关紧要，SFC 7 返回值 “RET\_VAL” =W#16#7002。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	REQ=1: 在 DP 主站上触发的从站的硬件中断
IOID	INPUT	BYTE	I, Q, M, D, L, 常数	在传输区域内标识地址区域 (从 DP 从站的角度看): B#16#54= 外设输入 (PI) B#16#55= 外设输出 (PQ) 如果传输区域是一个混合模板 (即有输入又有输出) 则标识符为两个地址中较低的那个, 若两地址相同, 则指定为 B#16#54。
LADDR	INPUT	WORD	I, Q, M, D, L, 常数	传输区域的起始地址 (从 DP 从站的角度看)。 如果传输区域为一个混合模板 (即有输入又有输出), 则此参数为两地址中较低的那一个
AL_INFO	INPUT	DWORD	I, Q, M, D, L, 常数	中断 ID 此参数值将传送到在 DP 主站上运行的 OB40 中, (变量 OB40_POINT_ADDR)。 如果从站为智能从站, 则必须在重站程序中访问诊断信息。(见/70/)
RET_VAL	OUTPUT	INT	I, Q, M, D, L	当功能块执行过程中出错时, 返回的故障代码在此参数输出。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: 从站触发的硬件中断尚未被 DP 重站确认

故障信息

故障代码 (W#16#...)	说明
0000	作业无故障执行结束
7000	第一次调用时 REQ=0, 未激活硬件中断请求; 参数 BUSY=0
7001	第一次调用时 REQ=1, 硬件中断申请已发送到 DP 主站; 参数 BUSY=1
7002	调用间歇 (REQ 无关)。
8090	传输区域的起始地址不正确
8091	中断禁止 (用户配置)
8093	参数 “IOID” 和 “LADDR” 指定了一个不具有硬件中断请求能力的模板
80B5	DP 主站中的调用不允许
80C6	分布 I/O 当前不能访问

## 16.2 用 SFC 11 “DPSYC\_FR” 同步 DP 从主站组

### 描述

通过 SFC 11 “DPSYC\_FR”，实现同步一个或多个组 DP 从站。

此功能包括发送以下一个命令或多个命令的组合到相关的组：

- SYNC（同时输出，并且保持 DP 从站上输出的状态）
- UNSYNC（取消 SYNC 控制命令）
- FREEZE（保持 DP 从站的输入状态，读入到输入区）
- UNFREEZE（取消 FREEZE 控制命令）

---

### 注意

注意，当执行重新启动和暖启动时，控制命令 FREEZE 和 SYNC 依然保持有效。

另外应注意在同一时间，只能初始化一个 SYNC/UNSYNC 或者一个 FREEZE/UNFREEZE，控制命令。

---

### 前提条件

在发送以上的控制命令之前，必须通过 STEP 7（见/231/）将 DP 从站分配到一个组内，必须清楚哪一个 DP 从站属于哪一个组号，并且知道不同组对于 SYNC 和 FREEZE 控制命令的响应。

### SFC 的执行过程

SFC 11 “DPSYC\_FR” 是一个非同步 SFC，也就是说，SFC 的执行需要几个 SFC 调用周期，通过设定参数 REQ=1，启动 SFC 11 作业。

作业执行的状态通过输出类型参数 “RET\_VAL” 和 “BUSY” 反映，参见参数 “REQ”、“RET\_VAL” 和 “BUSY” 的具体含义。

### 作业识别

在第一个工作完成以前，如果触发了 SYNC/FREEZE 并且再次调用了 SFC 11，则 SFC 的响应与新的一次调用的作业是否相同有关。如果输入参数 LADDR、GROUP 和 MODE 相匹配，这个 SFC 可以理解为上一个调用的继续。

### 写 DP 模板的输出

触发写 DP 模板的输出的方法如下：

- 通过传送指令
- 写输出过程映像表值到模板（在 OB1 执行结束或调用 SFC 27 “UPDAT\_PO” 实现）
- 调用 SFC 15 “DPWR\_DAT”

通常情况下，DP 主站周期性地将输出字节发送到 DP 从站的输出模板上。（在 PROFIBUS DP 总线的周期内）。

如果需要同时将一组输出发送到从站模板上（可能是分布在不同的从站上），可以通过 SFC 11 “DPSYC\_FR” 标准块发送 SYNC 命令到相关的 DP 主站上实现。

### 控制命令 SYNC 的作用

通过 SYNC 控制命令，所选择的组内的 DP 从站切换到 Sync 模式。即，DP 主站传送当前的输出数据和结构到 DP 从站，并保持这些输出状态不变。接下来的扫描周期中，DP 从站将新的输出数据存储在内缓冲区中，但实际模板输出状态不变。

每执行一个 SYNC 控制命令，DP 从站（所选择的组内的从站）将存储在内部缓冲区中的输出数据发送到输出模板上。

当 SFC 11 “DPSYC\_FR” 执行 UNSYNC 命令后，输出模板状态才能周期性刷新。

---

#### 注意

若当发送控制命令时，所选择的组内的 DP 从站当时不在网络上或者有故障，它将不会切换到 SYNC 模式。这种情况也不会有反馈信息在 SFC 执行后。

---

### 读入 DP 模板的输入数据

DP 模板的输入数据读入方法如下：

- 通过装载指令
- 刷新输入过程映像（当 OB1 开始时系统自动刷新或通过调用 SFC 26 “UPDAT\_PI” 实现）
- 调用 SFC 14 “DPRD\_DAT”

通常情况下，DP 主站周期性地刷新输入状态，供 CPU 使用（在 PROFIBUS DP 总线周期内）。

如果需要同时得到某一组输入状态，这些输入可能分布在不同的 DP 从站上的，可以通过 SFC 11 “DPSYC\_FR” 发送 FREEZE 控制命令到相关 DP 主站来实现。

### 控制命令 FREEZE 的作用

通过 FREEZE 控制命令，涉及到的 DP 从站将切换到 Freeze 模式。即，DP 主站指示相关从站锁定当前的输入信号状态，并将锁定了的输入信号状态传送给 CPU。

每执行一次 FREEZE 控制命令，DP 从站将再一次锁定他的输入信号状态，只有当通过 SFC 11 “DPSYC\_FR” 发送了 UNFREEZE 命令后，DP 主站才能接收到周期性刷新的 DP 从站的输入信号状态。

---

#### 注意

当发送控制命令时，若所选择组的 DP 从站未连接到网络上或者是有故障，DP 从站将不会切换到 FREEZE 模式，此信号也不会反馈到 SFC 11 执行后的返回参数中。

---

### 数据一致性

由于 SFC 11 “DPSYC\_FR” 的执行是异步的，并且在执行过程中有可被更高级别的 OB 块中断，因此在调用 SFC 11 “DPSYC\_FR” 时，必须确定，过程映像区与实际的输入输出是一致的。

如果你遵守以下一致性原则，才可以保证数据的一致性。

- 定义合适的“SYNC 输出”和“FREEZE 输入”过程映象子区（S7-400 有此功能）在第一次调用 SYNC 作业之前调用 SFC 27 “UPDAT\_PO”；在最后一次调用 FREEZE 作业之后调用 SFC 26 “UPDAT\_PI”

- 使用直接 I/O 访问的方式存取 SYNC 作业和 FREEZE 作业所涉及到的输入和输出。当 SYNC 作业激活时，不允许写此作业所涉及的输出；当 FREEZE 作业激活，不允许读该作业所涉及的输入。

#### 使用 SFC 15 和 SFC 14

若使用了 SFC 15 “DPWR\_DAT”，则在发送 SYNC 作业之前，SFC 15 必须执行完毕。

若使用了 SFC 14 “DPRD\_DAT”，则在发送 FREEZE 作业之前，SFC 14 必须执行完毕。

#### SFC 11 “DPSYNC\_FR” 和启动组织块

要使用 SFC 11 的功能，用户必须在启动组织块内编写发送 SYNC 和 FREEZE 控制命令。

当用户程序启动时，若需要在 Sync 模式下对一组或多组 DP 从站进行输出，则必须在启动过程中初始化这些输出，并且调用带有 SYNC 控制命令的 SFC 11。

当用户程序启动时，若需要在 FREEZE 模式下读入一组或多组 DP 从站的信号状态，则必须对这些输入信号在启动过程中，调用带有 FREEZE 控制命令的 SFC 11 “DPSYC\_FR”。

参数	声明	数据类型	存储区域	说明
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: 选择第 1 组 位 1=1: 选择第 2 组 : 位 7=1: 选择第 8 组 一个作业可以选择多个组，常量 B#16#0 是非法的。
MODE	INPUT	BYLE	I, Q, M, D, L, 常数	作业 ID 号（按 EN50170 卷 2，PROFIBUS 编码） 位 0: 保留（值为 0） 位 1: 保留（值为 0） 位 2: =1 执行 UNFREEZE =0 无意义 位 3: =1 执行 FREEZE =0 无意义 位 4: =1 执行 UNSYNC =0 无意义 位 5: =1 执行 SYNC =0 无意义 位 6: 保留（值为 0） 位 7: 保留（值为 0） 可能的值： • 一个作业一个 ID 号 - B#16#04 (UNFREEZE)

参数	声明	数据类型	存储区域	说明
				<ul style="list-style-type: none"> <li>- B#16#08 (FREEZE)</li> <li>- B#16#10 (UNSYNC)</li> <li>- B#16#20 (SYNC)</li> <li>• 一个作业多个 ID 号</li> <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>
RET_VAL	OUTPUT	INT	I, Q, M, D, L	当 SFC 执行过程中, 若发生错误, 则此参数中返回故障代码。 每次 SFC 执行后, 用户程序应访问此参数。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: SYNC/FREEZE 作业尚未完成。

故障信息

故障代码 (W#16#...)	说明
0000	作业无错误完成
7000	第一次调用时 REQ=0, LADDR、GROUP 和 MODE 所指定的作业未激活: BUSY=0
7001	第一次调用时 REQ=1, LADDR、GROUP 和 MODE 所指定的作业已触发: BUSY=1
7002	调用过程中 (REQ 值不相干), 激活的 SYNC/FREEZE 作业进行中。BUSY=
8090	LADDR 所指定的地址不是 DP 主站
8093	SFC 不能用于 LADDR 参数所指定的模板 (DP 主站的配置或版本)
8094	GROUP 参数非法
8095	MODE 参数非法
80B0	GROUP 参数所选择的组未配置
80B1	GROUP 参数所选择的组号未分配到此 CPU
80B2	MODE 参数所定义的 SYNC 作业不允许在 GROUP 参数所选择的组内
80B3	MODE 参数所定义的 FREEZE 作业不允许在 GROUP 参数所选择的组内
80C2	DP 主站临时存储资源: DP 主站当前正处理最大数量的作业
80C3	当前 SYNC/UNSYNC 作业不能被激活, 因为在一个时间内只能有一个 SYNC/UNSYNC 作业激活, 请检查用户程序
80C4	当前 FREEZE/UNFREEZE 作业不能被激活, 因为在一个时间内只能有一个 FREEZE/UNFREEZE 作业激活, 请检查用户程序
80C5	不能访问分布式 I/Os: DP 子系统故障
80C6	由于 CPU 断开与 I/O 的联接, 作业被拒绝
80C7	由于 DP 主站在冷/热启动, 作业被拒绝

## 16.3 用 SFC 12 “D\_ACT\_DP” 取消和激活 DP 从站

### 描述

通过 SFC 12“D\_ACT\_DP”，可以专门取消和重新激活配置的 DP 从站，另外，通过 SFC 12，可以决定是否激活或取消配置的 DP 从站。

SFC 12 “D\_ACT\_DP” 不能作用于通过 DP/PA 连接到 DP 主站上的 PROFIBUS PA 现场设备。

---

### 注意

当 SFC 12 作业正在执行时，不能通过 PG 将修改后的配置下载到 CPU（即处于 CiR 处理过程中）

当 CPU 处于 CiR 处理过程中，即正在接收修改后的配置时，CPU 将拒绝初始化 SFC 12 的请求。

---

### 目的

当我们配置了 DP 从站后，而此从站当前不存在或者暂时不需要时，但是 CPU 还是会有规律地去访问这些 DP 从站。若我们取消了这些从站，则 CPU 对这些站的访问就会停止。通过这种形式，我们会获得尽可能快的 DP 总线循环时间，并且相应的报警也再不会出现。

### 应用举例

对于机器设备生产者来说，在一机器设备系列中可能选择很多种装置。然而，每一个售出的机器设备只包含了其中一些装置。

对于制造商而言，他将每一个可能的装置选择看作是一个 DP 从站，这样他可以开发和保持一个完整的通用的程序，它包括所有可能的选择装置，针对于不同的机械设备，可以利用 SFC 12 取消掉那些不同的选择装置。

这种情况同样发生在机床上，机床有很多可用的工具选项，但实际上，在一定时期内我们只用其中一部分，这些工具可以看作是 DP 从站。利用 SFC 12 可以激活那些我们当前需要的 DP 从站，取消那些以后会需要的 DP 从站。

### SFC 的执行

SFC 12 “D\_ACT\_DP” 是非同步执行的，即他的执行需要多个 SFC 的调用周期。调用 SFC 12 将参数 REQ=1，则启动这个块的功能。

块执行作业的状态通过参数“RET\_VAL”和“BUSY”反映，另见 2.2 章节。参见“S7-300/400 系统的系统软件/—标准功能块”。

### 识别作业

若启动了一个取消或激活作业后，在 SFC 12 执行完之前，用户程序中再次调用了 SFC，此时 SFC 的反应主要看新的调用是否与上次调用指的是同一个作业，即两次调用参数“LADDR”参数相同，若相同，则新的调用被视作上一个调用的继续。

### 取消 DP 从站

当利用 SFC 12 取消了一个 DP 从站后，此从站的输出将被置成配置的替换值或者“0”值（安全状态），DP 主站将不在对这个 DP 从站进行寻址。此时 CPU 或者 DP 主站也不认为这个从站有故障或丢失，因此没有相应的错误 LED 显示。

取消的 DP 从站的输入过程映像区被刷新成“0”，即此时这个从站的状态如同一个故障的 DP 从站。

如果用户程序中直接访问了已经被取消了的 DP 从站的数据，则会出现 I/O 访问错误，系统会自动调用 OB122，并且相应的事件信息会存储在 CPU 的故障缓冲区内；如果试图用 SFC（即 SFC 59 “RD\_REC”）去访问一个已经被取消的从站，则会在参数“RET\_VAL”中返回错误信息，提示所访问的是一个不可用的 DP 从站。

取消 DP 从站不会启动程序错误 OB 块（OB 85），即使此从站的输入输出是要被系统过程映像刷新的，在诊断缓冲区内也不会有记录。

取消 DP 从站也不会启动机架错误 OB 块（OB86），在诊断缓冲区内也不会有故障记录。

若在已经将从站通过 SFC 12 取消掉之后，此从站出现故障，则操作系统检测不到这种故障。因此也不会接下来启动 OB 86，或者将信息写到诊断缓冲区中，此从站的故障只有当使用重新激活功能时，在返回变量“RET\_VAL”中有显示。

若希望取消在交叉通用中作为发送方的 DP 从站，我们建议首先取消掉接收方 DP 从站（收听者）。即接收从站检测发送方输送到 DP 主站上的输入数据，只有进行完以上的步骤才能取消操作为发送者的 DP 从站。

### 激活 DP 从站

当使用 SFC 12 重新激活 DP 从站时，DP 主站将对该从站进行配置和参数分配（如同故障站的恢复）。当从站能够重新传送数据时，激活作业结束。

激活 DP 从站时，不触发程序故障 OB（OB85），即使从站的输入或输出属于系统刷新的过程映像区。同时在诊断缓冲区中也没有记录。

激活 DP 从站时，不触发机架故障 OB（OB86），操作系统也不在诊断缓冲区中做记录。

如果使用 SFC 12 激活一个已经被取消的从站，且此从站已经从 DP 总线摘除，此时 CPU 上的“DP\_BUSF”指示灯将闪烁的一分钟。过了这段监控时间，指示灯熄灭并且 SFC 返回故障代码 W#16#80A2，从站保持取消状态。如果从站以后重新接入总线，必须用 SFC 12 重新激活。

---

#### 注意

激活一个 DP 从站是一项耗时的工作。因此，如果希望取消激活工作，可将 MODE=2，LADDR 值不变，重新启动 SFC 12。重复调用 SFC 12 直到 RET\_VAL=0，则激活作业被取消。

---

如果想激活参与交叉通讯的 DP 从站，建议先激活发送方，然后激活接收方。

### CPU 启动

根据启动模式，对于 DP 从站，CPU 操作系统有如下表现：

- 冷启或暖启模式下，DP 从站自动被激活。
- 在热启模式下，从站激活状态不变。即激活的从站还是激活的，取消的从站还是取消的。



CPU 启动后，该 CPU 周期性地访问所有配置的且未被取消的从站，即使这些从站不存在或无响应。

### 注意

启动 OB 中不能调用 SFC 12。

### 参数

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	电平触发控制参数 REQ=1: 执行激活或取消
MODE	INPUT	BYTE	I, Q, M, D, L, 常数	作业 ID。允许值 <ul style="list-style-type: none"> <li>• 0: 查询 DP 从站状态为激活或取消</li> <li>• 1: 激活 DP 从站</li> <li>• 2: 取消 DP 从站</li> </ul>
LADDR	INPUT	WORD	I, Q, M, D, L, 常数	DP 从站的逻辑地址
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如果块执行有误，返回故障代码
BUSY	OUTPUT	BOOL	I, Q, M, D, L	状态码： <ul style="list-style-type: none"> <li>• BUSY=1: 作业激活</li> <li>• BUSY=0: 作业结束</li> </ul>

### 故障代码

故障代码 (W#16#...)	说明
0000	作业无错误完成
0001	DP 从站激活状态 (仅当 MODE=0 时)
0002	DP 从站取消状态 (仅当 MODE=0 时)
7000	REQ=0 首次调用。LADDR 指定的作业未激活，BUSY=0
7001	REQ=1 首次调用。LADDR 指定的作业已触发，BUSY=1
7002	调用过程中 (REQ 无关)，激活的作业正在进行中，BUSY=1
8090	<ul style="list-style-type: none"> <li>• LADDR 指定的地址未做配置</li> <li>• LADDR 指定的是一个 CPU 类型智能从站的地址</li> </ul>
8092	所寻址的 DP 从站当前未执行激活作业 (仅当 MODE=1)
8093	LADDR 指定的 DP 从站不存在，或参数 MODE 无效
80A1	所寻址的 DP 从站不能被参数化 (仅当 MODE=1) 注意： 当被激活的从站在被参数化过程中重新出现故障，则 CPU 仅提供这条信息。若在参数化单个模板时故障，则 SFC 返回信息 W#16#0000。
80A2	寻址的 DP 从站没有回确认
80A3	DP 主站不支持 OB 功能

故障代码 (W#16#...)	说明
80A4	CPU 不支持此功能，对于外部 DP 主站
80A6	DP 从站中槽错误，不能访问用户数据（当 MODE=1 时） 注意 只要从站在参数化结束后，但在 SFC 结束前出现故障，则 SFC 返回此故障信息，如果仅是单个模板不能访问，SFC 返回值为 W#16#0000。
80C1	SFC 12 在启动和执行另外一个逻辑地址的作业（仅当 MODE=1 时有些代码）
80C3	<ul style="list-style-type: none"> <li>临时资源错误：CPU 在处理最多的激活和取消作业（仅当 MODE=1 或 2 时有此信息）。</li> <li>CPU 正在接收修改后的配置数据，当前不能进行激活/取消 DP 从站的作业。</li> </ul>

## 16.4 用 SFC 13 “DPNRM\_DG” 读 DP 从站诊断数据（从站诊断）

### 从站诊断

每一个 DP 从站都提供诊断数据，数据构成遵照 EN 50170 第二卷，PROFIBUS。你需要使用 SFC 13 “DPNRM\_DG” 来读取这些诊断数据。

下表中列出了从站诊断数据的基本构成，如需更多信息请参阅 DP 从站手册。

### 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	主站站号
4	供应商 ID（高字节）
5	供应商 ID（低字节）
6...	更多的从站诊断信息

### 说明

使用 SFC 13 “DPNRM\_DG”（读 DP 从站诊断数据），你可以读取 DP 从站当前的诊断数据，其格式遵照 EN 50170 第二卷 PROFIBUS 被读去的数据将无误地传送到 RECORD 所指定的目的区。

将 SFC 13 的输入参数 REQ 置 1 来启动读取工作。

## 功能

读取工作是异步执行的,也就是说需要多次调用 SFC 13。工作的状态由输出参数 RET\_VAL 和 BUSY 来指示,请参阅异步 SFC 中参数 REQ、RET\_VAL 和 BUSY 的含义。

参数	声明	数据类型	存储区域	说明
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 从站的诊断地址 <b>注意:</b> 地址必须以十六进制格式输入例如: 诊断地址 1022 意味着 LADDR: = W#16#3FE
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能执行时如果出现故障,返回值中存放故障代码,如果没有故障,其中存放实际传送的数据长度。
RECORD	OUTPUT	ANY	I, Q, M, D, L	读取的诊断数据存放的目的还只允许 BYTE 数据类型。读取的数据记录或目的区的最小长度 6。数据记录的最大长度 240。标准从站能够提供的诊断数据可多于 240 字节,最多到 244 字节。在这种情况下,前 240 字节被传送到目的区并且数据中的溢出位被置位。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: 读取作业未完成。

### 输入参数 RECORD

CPU 按如下情况处理不同长度的诊断数据:

- 如果 RECORD 指定的长度小于提供的数据字节数,则数据被放弃并在 RET\_VAL 中返回故障代码。
- 如果 RECORD 指定的长度大于提供的数据字节数,则数据被接收到目的区并将实际长度写入 RET\_VAL,该值为正数。

#### 注意

你必须确保 RECORD 的实参适用于一个工作的所有作业。

一个作业由输入参数 LADDR 来唯一指定。

### 诊断数据多于 240 字节的标准从站

当标准从站的标准诊断数据在 241 和 244 字节之间时,请注意以下几点:

- 如果 RECORD 指定的长度小于 240 字节,则数据被放弃并在 RET\_VAL 中返回故障代码。
- 如果 RECORD 指定的长度大于或等于 240 字节,则前 240 字节的标准诊断数据被传送到目的区并且数据中的溢出位被置位。

#### 输出参数 RET\_VAL

- 如果功能执行时出现故障，则返回值中存放故障代码。
- 如果数据传送没有出现故障，则返回值中存放读取数据的字节数，该值为正数。

---

#### 注意

从 DP 从站读取的数据量取决于诊断状态。

---

#### 故障信息

RET\_VAL 参数中的故障代码说明在第二章，这章中还包括 SFC 的一般故障代码。SFC 13 所指明的故障信息附属于 SFC 59 “RD\_REC” 的故障信息参见使用 SFC 59 “RED\_REC” 读取数据记录。

#### S7-400 系统资源

当调用 SFC 13 “DPNRM\_DG” 执行一个当前未被处理的工作时，在 S7-400 中，CPU 的资源（内存空间）被占用。你可以快速连续地调用 SFC 13，只需不超过 CPU 允许的同时激活的 SFC 13 的最多作业个数。你可在/101/中查到这种作业的最多个数。

如果你同时激活多个作业，所有工作将被执行且不互相干扰。

如果你达到了系统资源的极限，将在 RET\_VAL 中指示。在这种情况下，重复作业。

## 16.5 用 SFC 14 “DPRD\_DAT” 读取 DP 标准从站的连续数据

#### 数据连续性

参阅 S7 通讯和 S7 基本通讯概述部分——数据连续性。

#### SFC 14 的作用

你需要 SFC 14 “DPRD\_DAT”，因为使用装载指令访问 I/O 或输入映象区最多只能读出 4 个连续的字节。

---

#### 注意

如果需要，你也可以通过输入过程映象区来读取连续数据。查阅技术数据，以确认 CPU 是否支持该功能。

---

#### 说明

使用 SFC 14 “DPRD\_DAT”（读取 DP 标准从站的连续数据），你可读取 DP 标准从站的连续数据，最大长度由 CPU 决定。你可从 CPU 的技术数据中查到该最大长度。如果在数据传送中没有出现错误，则读出的数据被存放在由 RECORD 指定的目的区。

目的区的长度应与 STEP 7 中对所选模板的配置一致。

如果你需读取的 DP 从站是模块化设计或只有多个 DP 识别符，那么通过指定起始地址，每次 SFC 14 的调用只能访问一个模板/DP 身份的数据。

参数	声明	数据类型	存储区域	说明
LADDR	INPUT	WORD	I, Q, M, D, L, 常数	模板输入区中将被读取的数据的起始地址。 注意： 地址必须以十六进制格式输入。例如： 诊断地址 100 意味着 LADDR： =16#16#64
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如果功能执行出现错误，返回故障代码
RECORD	OUTPUT	ANY	I, Q, M, D, L	读取数据存放的目的区。其长度必须与 STEP 7 中对所选模板的配置一致。只能使用 BYTE 数据类型。

## 故障信息

## 注意

如果访问 DPV1 从站，来自这类从站的故障信息可从 DP 主站传至 SFC。故障信息详情，请见使用 SFB 54 “RALRM” STATUS [3] 从 DP 从站接收中断。

故障代码 (W#16#...)	说明
0000	没有发生错误
8090	<ul style="list-style-type: none"> <li>没有指定逻辑基本地址的模板或</li> <li>忽视了连续数据长度的严格要求或</li> <li>LADDR 参数中起始地址不是十六进制格式</li> </ul>
8092	ANY 中指定的数据类型不是 BYTE
8093	要读取连续数据的 DP 从站中没有 LADDR 所指定的逻辑地址。
80A0	I/O 设备访问错误
80B0	外部 DP 接口模板引起从站故障
80B1	目的已指定的长度与 STEP 7 配置的用户数据长度不一致。
80B2	系统故障，外部 DP 接口模板
80B3	系统故障，外部 DP 接口模板
80C0	数据未被模板读取
80C2	系统故障，外部 DP 接口模板
80Fx	系统故障，外部 DP 接口模板
87xy	系统故障，外部 DP 接口模板
808x	系统故障，外部 DP 接口模板

## 16.6 用 SFC 15 “DPWR\_DAT” 向 DP 标准从站写连续数据

### 数据连续性

参阅 S7 通讯和 S7 基本通讯概述部分——数据连续性。

### SFC 15 的目的

你需要 SFC 15 “DPWR\_DAT”，因为使用传送指令访问 I/O 或输出映象区最多只能写 4 个连续字节。

#### 注意

如果需要，你也可以通过输入过程映象区来写连续数据。查阅技术数据，以确认 CPU 是否支持该功能。

### 说明

使用 SFC 15 “DPWR\_DAT”（向 DP 标准从站写连续数据），你可将 RECORD 指定的数据连续地传送到寻址的 DP 标准从站，如果需要，可传到过程映象区（如果你已为 DP 从站的相应地址区在过程映象区中配置了连续区域）传送的数据长度取决于 CPU。你可从 CPU 的技术数据中查到该最大长度。数据传送是同步的，也就是说 SFC 结束时，写工作也结束。该数据的长度应与 STEP 7 中对所选模板的配置一致。

如果 DP 标准从站是模块化设计，你只能访问一个 DP 从站模板。

参数	声明	数据类型	存储区域	说明
LADDR	INPUT	WORD	I, Q, M, D, L, 常数	模板输出区中将被写入数据的起始地址。 <b>注意：</b> 地址必须以十六进制格式输入。例如： 诊断地址 100 意味着 LADDR： =16#16#64
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如果功能执行出现错误，返回故障代码
RECORD	INPUT	ANY	I, Q, M, D, L	待写数据存放的源数据区。其长度必须与 STEP 7 中对所选模板的配置一致。只能使用 BYTE 数据类型。

### 故障信息

#### 注意

如果访问 DPV1 从站，来自这类从站的故障信息可从 DP 主站传至 SFC。故障信息详情，请见使用 SFB54 “RALRM” STATUS [3] 从 DP 从站接收中断。

故障代码 (W#16#...)	说明
0000	没有发生错误
808x	系统故障，外部 DP 接口模板
8090	<ul style="list-style-type: none"> <li>• 没有指定逻辑基本地址的模板或</li> <li>• 忽视了连续数据长度的严格要求或</li> <li>• LADDR 参数中起始地址不是十六进制格式</li> </ul>
8092	ANY 中指定的数据类型不是 BYTE
8093	要写入连续数据的 DP 从站中没有 LADDR 所指定的逻辑地址。
80A1	I/O 设备访问错误
80B0	外部 DP 接口模板引起从站故障
80B1	目的已指定的长度与 STEP 7 配置的用户数据长度不一致。
80B2	系统故障，外部 DP 接口模板
80B3	系统故障，外部 DP 接口模板
80C1	模板上原先写工作的数据还未被模板处理
80C2	系统故障，外部 DP 接口模板
80FX	系统故障，外部 DP 接口模板
85xy	系统故障，外部 DP 接口模板





## 17 用于全局数据通讯的 SFC

### 17.1 利用 SFC 60 “GD\_SND” 传送一个全局数据包

#### 描述

利用 SFC 60 “GD\_SND”（全局数据发送），全局数据包收集数据并且将它们按照全局数据包中指定的路径发送出去。这些全局数据包必须已经在 STEP 7 中设置好。

SFC 60 “GD\_SND”可以在用户程序的任意位置被调用。

系统的扫描速率、数据的采集和发送的周期检测不会受到 SFC 60 调用的影响。

#### 中断能力

SFC 60 “GD\_SND”可以被高优先级的块中断。在高优先级的块中再次使用 SFC 60 发送相同的全局数据包也是允许的。

数据可以被高优先级的块采集并发送，但当程序回到被中断的 SFC 时，该动作立即停止，并且已经被采集的数据也被放弃。

这个过程表明在高优先级的块运行当中，传送连续数据（全局数据的连续性）。

#### 全局数据的连续性

以下规则保证了从不同的存储区采集和发送数据的连续性。

规则如下：

- 简单的数据类型（位、字节、字、双字）
- 字节、字、双字类型，矩阵的最大长度取决于相应的 CPU。

#### 保证一个完整全局数据包的连续性

CPU 所发送的全局数据包数据结构不能够保证所采集数据的连续性。例如，有的情况当数据包由一个字节型的矩阵构成，而有的字节数量却超出了 CPU 所允许的最大长度范围。

如果你要保证一个完整全局数据包的连贯性，在你的程序中请遵照以下步骤：

- 利用 SFC 39 “DIS\_IRT” 或者 SFC 41 “DIS\_AIRT” 取消或延迟高优先级的中断和同步故障。
- 调用 SFC 60 “GD\_SND”
- 利用 SFC 40 “EN\_IRT” 或者 SFC 42 “EN\_AIRT” 再次确认高优先级的中断和同步故障。

参数	声明	数据类型	存储区域	说明
CIRCLE_ID	INPUT	BYTE	I, Q, M, D, L, 常数	要发送的全局数据包所在的全局数据组号, 利用 STEP 7 配置全局数据时需要指明这个号码。 允许值: 1~16 允许最大全局数据组号, 见 CPU 的技术数据。
BLOCK_ID	INPUT	BYTE	I, Q, M, D, L, 常数	在被选定的全局数据组中要被发送的全局数据包号码。该号码在 STEP 7 配置全局数据时设置。 数据包时定义 允许值: 1~3 允许最大全局数据组号, 见 CPU 的技术数据。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息

#### 故障信息

故障代码 (W#16#...)	说明
0000	没有故障发生。
8081	全局数据包选择参数 CIRCLE_ID 和 BLOCK_ID 没有设置。
8082	参数 CIRCLE_ID 或 BLOCK_ID 的值不合法。
8083	在执行 SFC 时有故障产生。错误类型被写入了状态信息中, 可由程序读取该故障。
8084	由于在高优先级的块中再次调用了含有相同全局数据包的全局数据包导致 SFC 中断。
8085	一个故障被写入了状态信息中。

#### 注意

在每次调用 SFC 60 后, 你应该检查全局数据包的连续性。如果需要, 进行重新设置。

## 17.2 用 SFC 61 “GD\_RCV” 接收全局数据包

### 描述

利用 SFC 61 “GD\_RCV”（全局数据接收），根据全局数据表中定义的指定的数据从一个数据包释放，然后被写入接收的数据包，但是这些设置必须已经在 STEP 7 中设置好了。

SFC 61 “GD\_RCV” 可以在程序的任意位置被调用。

系统的扫描速率、数据的采集和发送周期不会影响 SFC 61 调用的影响。

### 中断能力

SFC 61 可以被高优先级的块中断，这只是在全局数据设置一致的情况下才可以确定。如果 SFC 61 在执行的过程中被中断，可能是由于传输相同全局数据块的 SFC 61 在高优先级的块被再次调用了。

相应的数据被写入了高优先级的全局数据块中。当程序返回到被中断的 SFC 块时，该操作立即停止。

### 全局数据的连贯性

以下的规则保证了在不同的存储区中数据的连贯性。

规则如下：

- 简单的数据类型（位、字节、字、双字）
- 字节、字、双字类型，矩阵的最大长度由相应的 CPU 定义。

### 保证一个完整全局数据包的连贯性

接收端 CPU 中的全局数据包不能够自己识别其原始数据见来由同一个或者其它的数据表，例如，当数据包由 3 个元素组成。

如果你要保证一个完整全局数据包的连贯性，在你的程序中请遵照以下步骤：

- 利用 SFC 39 “DIS\_IRT” 或者 SFC 41 “DIS\_AIRT” 取消或延迟高优先级的中断和同步错误。
- 调用 SFC 60 “GD\_SND”
- 利用 SFC 40 “EN\_IRT” 或者 SFC 42 “EN\_AIRT” 再次确认高优先级的中断和同步错误。

参数	声明	数据类型	存储区域	说明
CIRCLE_ID	INPUT	BYTE	I, Q, M, D, L, 常数	接收数据的全局数据包所在全局数据组号，这个号码在配置全局数据时设定。 允许值：1~16。允许最大全局数据组号，见 CPU 的技术数据。
BLOCK_ID	INPUT	BYTE	I, Q, M, D, L, 常数	在被选定的全局数据组中要存放所接收数据的全局数据号码，这个号码在配置全局数据时定义。 允许值：1~3。允许最大全局数据组号，见 CPU 的技术数据。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息

故障信息

故障代码 (W#16#...)	说明
0000	无故障产生。
8081	全局数据包选择参数 CIRCLE_ID 和 BLOCK_ID 没有设置。
8082	参数 CIRCLE_ID 或 BLOCK_ID 的值不合法。
8083	在执行 SFC 时有错误产生，错误类型被写入了状态信息中，可由程序检查该错误。
8084	由于在高优先级的块中再次调用了含有相同全局数据包的块，导致了 SFC 中断。
8085	一个错误被写入了状态信息中。

注意

在每次调用 SFC 61 后，你应该检查全局数据包的连贯性。如果需要，可重新设置。

特性标识

以下表格列出了所有特性标识

标识	含义
W#16#0000—00FF	MC7 控制单元（组索引号为 0000）
W#16#0001	MC7 控制生成代码
W#16#0002	MC7 解码
W#16#0100—01FF	时钟系统（组索引号为 0100）
W#16#0101	分辨率为 1ms
W#16#0102	分辨率为 10ms
W#16#0103	无实际时钟
W#16#0104	BCD 控制的日时钟
W#16#0105	所有日时钟功能（设置日时钟，设置并读取日时钟） 日时钟同步：日时钟从属或以日时钟为主
W#16#0200—02FF	系统应答（组索引号为 0200）
W#16#0201	多处理器模拟能力
W#16#0202	可以冷启动、暖启动、热启动
W#16#0203	可以冷启动、热启动
W#16#0204	可以暖启动、热启动
W#16#0205	只允许暖启动
W#16#0206	使用预设资源可以在运行模式下对分布式 I/O 进行设置。
W#16#0207	冗余 CPU 在单一运行模式下：使用预设资源可以在运行模式下对分布式 I/O 进行设置。
W#16#0300—03FF	MC7 语言描述（组索引号 0B00）
W#16#0301	保留
W#16#0302	所有 32 位定点指令
W#16#0303	所有浮点指令
W#16#0304	Sin, asin cos, acos tan, atan, sqr, sqrt, ln, exp。

## 18 S7 通讯及 S7 基本通讯概述

### 18.1 S7 通讯及 S7 基本通讯所使用块的区别

#### 选择标准

除了全局数据通讯，在 S7 可编程控制器中 CPU 和 FM 之间还有两种交换数据的方式：

- 利用 SFC 进行非定义 S7 连接的数据交换通讯方式
- 利用 SFB 进行需定义 S7 连接的数据交换通讯方式

你选择哪种，取决于你所使用的 SIMATIC S7 可编程控制器类型（S7-300，S7-400）和其它用于数据交换的参数。下面表格可作为你在选择时的尺度。

标准	非定义 S7 连接的 SFC 通讯	需定义 S7 连接的 SFB 通讯
可用的块	S7-300 和 S7-400 用 SFC	S7-300 用 FB 和 FC S7-400 用 SFB 和 SFC
通讯连接	连接不必被组态。连接在 SFC 被激活时建立。通过参数控制可决定数据传输后该连接继续保持还是终止。如临时连接不能被建立，则相应发送工作不能进行。	连接在系统组态时被永久建立
进入停止时的变化	初始化了数据传输的 CPU 进入停止时，所有已建连接均被终止。	在停止时连接保持
在一个对象上的几条连接	在任何同一时间，在同一对象上只能有一条连接被建立。	可在同一对象上建立多条连接
寻址范围	可寻址本站和同一 MPI 网络的模板	可寻址在 MPI，PROFIBUS 和工业以太网上的模板。
通讯对象的数量	通讯对象的数量在一个接一个被建立时不会受到连接资源限制（见 /70/101）。（连接在程序运行时建立又在结束时被终止）。	同时可用的通讯对象连接数量受到连接资源的限制，决定于所用 CPU 的不同。
最大用户数据长度	最大 76 个用户数据长度被允许	最大传输的用户数据长度取决块的类型（USEND/BRCV，GET 等等）和通讯的对象（S7-300，S7-400 或者 M7）。
每次块调用可传输的变量数量	只可一个变量	S7-300：一个变量 S7-400：最多四个变量
块的分类	用于 S7 基础通讯的 SFC 是系统功能，因此，不占用用户存储器。	用于 S7 通讯的 SFB/FB 是系统功能，因此，它们需要背景数据块用于存储实际参数和静态变量

标准	非定义 S7 连接的 SFC 通讯	需定义 S7 连接的 SFB 通讯
地址参数的动态修改	地址参数的动态修改是可能的：在完成激活工作后，你可以寻址其它通讯对象。	S7-300：当即将激活时你可以识别寻址参数，当前面工作结束后新的参数可被确认。 S7-400：地址参数的动态修改是不可能的：连接在块首次被调用时固定和确认，直到暖启动和冷启动前都保持不变。

#### IEC 61131-5 在 S7-400 中的实现

IEC 61131-5 准则通过下列块来实现：

- USEND (SFB 8) /URCV (SFB 9)
- BSEND (SFB 12) /BRCV (SFB 13)
- PUT (SFB 15) /GET (SFB 14) 及相应的 READ/WRITE
- STATUS (SFB 22) /USTATUS (SFB 23)
- ALARM (SFB 33)
- NOTIFY (SFB 36)

START (SFB 19)、STOP (SFB 20) 和 RESUME (SFB 21) 实现用于程序控制功能的调用接口。

#### IEC 61131-5 在 S7-300 中的实现

IEC 61131-5 准则通过下列块来实现：

- USEND (SFB 8) /URCV (SFB 9)
- BSEND (SFB 12) /BRCV (SFB 13)
- PUT (SFB 15) /GET (SFB 14) 及相应的 READ/WRITE

## 18.2 数据连续性

### 定义

通过同时处理被同步修改的数据区域的大小被称作连续的数据区域。数据区大于连续数据区将被整体当作伪造数据区。

这意味着属于一起的数据区和大于连续性数据区可在同一时间内划分为几个新的数据区和旧的连续数据区。

### 举例

如果通讯块被中断将会生产不连续性，例如，被一个高优先级的 OB 块所中断。如果在本 OB 块的用户程序部分改变了已经被通讯块处理的数据，被传输的数据会产生：

- 部分为硬件中断处理前的
- 部分为硬件中断处理后的
- 这意味着这些数据不连续（不连贯）。

### 效果

如果较大的数据包以连续的形式被传送，传送方尽量不要被中断。这可能，例如，增加 CPU 的中断反应时间。

这意味着：较高质量的数据应该被绝对连续的传送，同时在系统中需要较长的中断反应时间。

### 在 SIMATIC 中的数据连续性

如果用户程序含有通讯功能，例如，BSEND/BRCV，访问普通的数据区，这些数据区可以，例如，借助于参数“DONE”来协调。在用户程序中通过本地通讯块传送的数据区的连续性因而可以得到保证。

然而，在 S7 通讯方式下，例如借助 OP 通讯 PUT/GET 或者 WRITE/READ，连续数据的大小必须在编程和组态阶段被考虑在内，因为在发送同步通讯数据时目的地设备（服务器）的用户程序中没有通讯块可用。

在 S7-300 和 C7-300（CPU 318-2 例外）通讯数据在程序的循环监测点以 32 字节被连续拷贝到用户程序中。大数据连续性不能得到保证，如要求一个定义的数据连续，在用户程序中的通讯数据不能超过 32 字节（最大可大 8 字节，以 CUP 型号而定）。

相反 S7-400 中的通讯数据不在程序的循环监测点，而是在程序循环的固定间隔被处理。因而系统中的数据连续性得以保证。

这些通讯区可用 GET/PUT 和 READ/WRITE 功能被连续访问，例如通过一个 OS 站的 OP 来实现。

### 推荐

关于数据连续性的深入信息，请参阅单独的块帮助信息或者 SIMATIC 通讯手册。

## 18.3 S7 通讯块概述

### 分类

对于 S7 通讯，组态连接是必须的。从应用程序中，使用 SFB/FB 或 SFC/FC 可调用集成的通讯功能。

块分为以下几类：

- 用于数据交换的 SFB/FB
- 用于改变操作状态的 SFB
- 用于查询操作状态的 SFB
- 用于要求连接的 SFC/FC

### S7-300/S7-400 通讯块的可用性

- 用于 S7-400 的块在“标准库”中
- 用于 S7-300 的可装载的块在“SIMATIC\_NET\_CP”库中。

要运行 S7-300 块，你需要一个用于 S7-300 的 SIMATIC NET CP，详细内容见相关资料。

### 用于数据交换的 SFB/FB

用于数据交换的 SFB/FB 用来在两个通讯对象间交换数据。如 SFB 只存在于本方的模板中，这种叫做单边通讯。如 SFB/FB 存在于本方和对方的模板中，这种叫做双边通讯。

S7-400 的块	S7-300 的块	缩写	简述
SFB 8 SFB 9	FB 8 FB 9	USEND URCV	和通讯对方通讯功能（URCV）执行序列无关的快速无需确认的数据交换（例如，操作和维护信息）这意味着在通讯对方数据可能被新数据所覆盖。
SFB 12 SFB 13	FB 12 FB 13	BSEND BRCV	安全的数据块传输到通讯对方。这意味着数据传输直到通讯对方的接收功能（BRCV）完成数据接受才结束。
SFB 14	FB 14	GET	在通讯对方无需额外的通讯功能的可控程序读数据
SFB 15	FB 15	PUT	在通讯对方无需额外的通讯功能的可控程序写数据
SFB 16	FB 16	PRINT	发送数据给打印机（S7-400 适用）

### 用于改变操作状态的 SFB（S7-400 适用）

通过 SFB 改变操作系统状态，你可以控制远端设备的运行方式。

用 SFB 进行数据交换以改变操作状态是单边的编程方式。



S7-400 的块		简述
SFB 19	START	当操作系统为 STOP 模式时触发 S7/M7-300/400 或 C7-300 的 CPU 的 RESTART 行为
SFB 20	STOP	当操作系统为 RUN 模式时触发 S7/M7-300/400 或 C7-300 的 CPU 的 STOP 行为或者终止操作模式或者重新启动
SFB 21	RESUME	当 S7-400 操作系统为 STOP 模式时触发 CPU 的 RESUME 行为

#### 用于查询操作状态的 SFB

通过 SFB 查询操作系统状态，你可以获得远端设备的运行信息。

用 SFB “STATUS” 是单边的编程方式，用 SFB “USTATUS” 是双边的编程方式。

S7-400 的块		简述
SFB 22	STATUS	按用户要求提供通讯对方（S7-400 CPU/M7-300/400）的操作状态
SFB 23	USTATUS	如相应连接的属性（发送运行状态消息）被定义，当运行方式改变时接收 S7-400 CPU 的运行状态。

#### 用于查询连接的 SFC/FC

S7-400 的块	S7-300 的块	简述
SFC 62 CONTROL		查询属于 SFB/FB 的、背景数据块连接状态
	FC 62 C CNTRL	通过连接 ID 来查询状态

#### 提示

你可用 SFC 87 C\_DIAG 来实现一个实际连接状态的诊断行为（仅适用于 S7-400）。

#### 程序示例

在 STEP 7 中提供了一个用于 S7-400 的示例程序将向你显示如何应用 SFB。该示例存储在 step7examples\com\_sfb 中，你可在示例程序的..\STEP7\Examples\ZDT01\_10 路径中找到。

## 18.4 S7 基础通讯块概述

### 用于 S7 基础通讯的 SFC 的分类

对于 S7 基础通讯，不必进行连接组态。在用户程序中通过 SFC 可调用集成的通讯功能。

SFC 被划分为两类：

- 用于在 S7 CPU 和其它模板交换数据的 SFC，此时通讯对象属于一个 S7 站内（对于内部通讯，标有“I”）。
- 用于在 S7 CPU 和其它模板交换数据的 SFC，此时通讯对象属于一个 MPI 网络内（对于外部通讯，标有“X”）。

通过 SFC 基础通讯在不同网络内的站间通讯是不可能的。

SFC 基础通讯可运行在 S7-300/400 的 CPU。你还可以用这些 CPU 去读写 S7-200 CPU 中的变量。

### 用于外部通讯的 SFC

块		简述
SFC 65	X_SEND	安全的传送数据块到通讯对象，这意味着数据传输直到通讯对方的接收功能（X_RCV）完成数据接受才结束。
SFC 66	X_RCV	
SFC 67	X_GET	读取通讯对象的变量而无需在对方编写程序，该功能在通讯对方中执行。
SFC 68	X_PUT	发送取通讯对象的变量而无需在对方编写程序，该功能在通讯对方中执行。
SFC 69	X_ABORT	结束一个已存在并无数据传输的连接，相应的连接资源在双方都被释放。

### 用于内部通讯的 SFC

块		简述
SFC 72	I_GET	读取通讯对象的变量而无需在对方编写程序，该功能在通讯对方中执行。
SFC 73	I_PUT	发送取通讯对象的变量而无需在对方编写程序，该功能在通讯对方中执行。
SFC 74	I_ABORT	结束一个已存在并无数据传输的连接，相应的连接资源在双方都被释放。

### 程序范例

在 STEP 7 中提供了两个用于 SFC 基础通讯的程序范例。你可在下列路径中找到 step7\examples\com\_SFC1 和 step7\examples\com\_SFC2。

### 最大用户数据长度

集成在 S7-300/400 CPU 的 SFC 通讯用于非组态 S7 连接。

通过 SFC 通讯可确保的用户数据长度为 76 字节（参数 SD 或者 RD）。确切数字在每个块中都可找到。

### 连接到通讯对象

对于用于非组态 S7 连接的通讯 SFC，连接在执行 SFC 时建立。取决于你所设定的 CONT 输入参数，连接将在通讯结束后保持或者终止。这意味着通讯有下列特性：

- 一个连接接一个连接的执行时候的通讯对象的数量可多于同时执行连接时候的数量。（具体见 CPU 规范/70/101/）。
- 因连接资源被占而目前无连接被建立（在本方或对方 CPU），RET\_VAL 将会显示。你应该在适合的时间点合理设置触发工作。因为不能保证之后的连接是否成功，如必要，应检查在 CPU 中是否有更多的连接资源。

用于组态 S7 连接的通讯 SFB 的现有连接不能用于非组态 S7 连接的通讯 SFC

一旦你触发了一个作业，为该作业建立的通讯连接只能用于该类作业。其它具有同一对象的作业可在一个作业执行后继续执行。

### 注释

如你的程序对同一对象有许多作业，请确保你调用的 SFC 的 RET\_VAL 得到 16#80C0 之后再继续进行下个作业。

### 作业识别

如你触发了一个传输工作或者一个非组态 S7 连接在任务完成后被终止，SFC 将根据工作结果得到不同的反映。下面表格解释了以 SFC 什么参数表示什么含义，如参数匹配，则 SFC 开始执行：

块		工作识别
SFC 65	X_SEND	DEST_ID, REQ_ID
SFC 67	X_GET	DEST_ID, VAR_ADDR
SFC 68	X_PUT	DEST_ID, VAR_ADDR
SFC 69	X_ABORT	DEST_ID
SFC 72	X_GET	IOID_LADDR, VAR_ADDR
SFC 73	X_PUT	IOID_LADDR, VAR_ADDR
SFC 74	X_ABORT	IOID, ADDR

### 中断的反应

非组态 S7 连接的 SFC 可以被高优先级的 OB 中断。如相同的 SFC 同样工作在那里被调用，第二个调用将失败，相应的记录进入 RET\_VAL。被中断的 SFC 接着执行。

### 访问 CPU 的工作存储器

与传输的用户数据的大小无关，操作系统的通讯功能访问 CPU 的最大的工作存储器，所以由于通讯的中断反应时间不会延长。

取决于你在 STEP 7 中所设定的最大通讯循环时间，在系统通讯功能执行一个工作时工作存储器可被访问多次。

### 客户端改变为 STOP

如果 CPU 初始化一个工作（并因此建立一个连接），在传输数据时改变为 STOP 状态，所有连接都被终止。

### 改变程序

在停机模式下对于用于非组态 S7 连接的 SFC 你所作的所有改变都会立即生效。这包括，特别的，删除包含通讯 SFC 非组态 S7 连接的 FC，FB，OB。

当你改变程序后，你必须执行一个暖启动或冷启动

无以上规则会导致连接资源被保留，因而可编程控制器处于无定义状态。

# 19 S7 通讯

## 19.1 用于 S7 通讯的 SFC/FC 和 SFB/FB 常见参数

### 分类

用于需组态连接 S7 通讯的 SFB/FB 的常见参数可按功能分为下面 5 类：

1. 用于激活块的控制参数
2. 用于寻址远端通讯对象的寻址参数
3. 发送到远端的数据接收点
4. 接受来自远端的数据接收点
5. 用于分析的是否块被正确执行的状态监视参数

### 控制参数

当 SFB/FB 被调用时或者信号状态与前面比发生一个变化时（例如一个上升沿）适合的控制参数状态（例如被置位）被设定，数据交换就发生。

---

#### 对于 S7-300 的注释

REQ 在第一次被调用时设定为 FALSE。

---

参数	说明
ID	参照本地连接的设定（在 STEP 7 连接组态中规范的）
R_ID	此参数设定一对发送和接受 SFB 块于一起，参数 R_ID 在 SFB/FB 的发送端和 SFB/FB 的接收端必须相同。 允许几个 SFB/FB 通讯对使用统一逻辑连接。 <ul style="list-style-type: none"><li>• R_ID 必须设定为 DW#16#wxyzWXYZ。</li><li>• 一个逻辑连接的一对通讯块时用的参数 R_ID 必须为唯一的。</li></ul>

参数 PI\_NAME 在相关的 SFB 中有描述（仅适用于 S7-400）。

---

#### 注释

**S7-300:** 当块激活时你可改变寻址参数 ID 和 R\_ID。新参数将在新任务开始前被确认有效。因此，你可以给多个 FB 使用一个背景数据块。

**提示技巧:** 为节省背景数据块的工作存储器有下列可能：

1. 对于 ID 不同的连接可用一个背景数据块。
2. 一个接收工作可用不同的匹配的 R\_ID 来实现不同的连接。
3. 1、2 两项可结合使用。

当前面工作结束时请注意新参数的有效性。当你激活发送块，发送块的 R\_ID 一定要和接收的匹配。

**S7-400:** 寻址参数 ID 和 R\_ID 仅在第一次调用时被确认（实际参数或背景预设值）。因此首次调用规范了与远端的通讯联系（连接），并保持直到新的暖或冷启动出现。

---

## 状态参数

借助状态参数你可监视块的任务是否正确执行或仍在工作中，也用来指示错误。

### 注释

状态参数仅在一个循环内有效，即从 SFB/FB 调用的第一条指令到下次调用，因此你需在每个块结束后确认。

## 发送和接收参数

如你不使用所有的 SFB/FB 发送和接收参数，首次未用参数应为 NIL 指针（参见/232/），并且参数应为无任何间隙。

### 对于 S7-400 应注意：

在第一次调用时，ANY 指针规范了最大的用户传输数据大小。这就是说，为确保数据连续性在 CPU 的工作存储器创建一个通讯缓冲区，该缓冲区占用 480 字节。推荐在暖、冷启动 OB 块中执行首次调用，否则 CPU 运行时涉及 SFB 调用的块还未被装载。

接下来你可使用发送和接受块来实现任何数据的通讯，然而，却不能超出首次调用的数量。SFB 的 B\_SEND 和 B\_RCV 是此规则的例外。借助它们你可每次传输高达 64K 字节（详见相关 SFB 12 B\_SEND 和 SFB 13 B\_RCV）

### SFB/FB 的双边通讯：

- 在发送和接收端的 SD\_I 和 RD\_I 数量必须相同。
- 在发送和接收端的 SD\_I 和 RD\_I 数据类型必须匹配。
- 在发送端 SD\_I 的数据量绝不能超出接收端 RD\_I 规定的数量（B\_SEND 和 B\_RCV 例外）。

ERROR=1 和 STATUS=4 表明你违反了以上规则。

最大数据量取决于你的通讯远是 S7-300 或 S7-400。

## 用户数据量

对于 SFB/FB、U\_SEND、U\_RCV、PUT、GET，传输的数据量都不得超出最大限度，最大用户数据量取决于下列：

- 块的类型
- 通讯对象

块	对象： S7-300/C7-300	对象： S7-400/M7-400
PUT/GET	160 字节	400 字节
U_SEND/U_RCV	160 字节	440 字节
B_SEND/B_RCV	32K 字节	64K 字节

有关详细最大数据量可参见相关 CPU 的技术数据。

## 实际用户数据量

如果以上数据量对你是不够的，你可按下列来决定最大用户数据量。

1. 与下列比表列出的对象通讯的首次读取有效的数据块大小。

本方 CPU	远端 CPU	数据量 (字节)
S7-300	任意	240 (S7-300)
S7-400	S7-300/C7-300	240 (S7-400)
S7-400	S7-400/CPU 318	480
S7-400	M7 模板	480
M7 模板	M7 模板	960

使用下列表中读取最大数据 (以字节为单位) 的可能, 在 SD\_I, RD\_I 和 ADDR\_I 区使用偶数长度。

		SD_I, RD_I, ADDR_I 的参数数量			
数据块大小	SFB / FB	1	2	3	4
240 (S7-300)	PUT/GET/ USEND	160	-	-	-
240 (S7-400)	PUT	212	196	180	164
	GET	222	218	214	210
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 用于需组态连接的 SFB 的启动路线

### 要求

下列描述用于 S7-400, 假设:

- 连接描述 (SDB) 已存在于模板中
- 组态连接已经建立
- 实际参数 ID 与 SFB 组态连接的 ID 相匹配

### 暖/冷启动

在暖/冷启动期间, 所有 SFB 都被设定为 NO\_INIT 状态, 实际参数被存储于背景数据块中不发生改变。

### SFB 双边数据传输时的暖/冷启动

通常在进行 SFB 双边数据通讯的模板不会同时进入暖/冷启动，此时 SFB 的反映按下列规范：

- 当出现暖/冷启动时，SFB 完成接收工作，但没有应答将产生一个错误序列帧（CFB，B\_RCV）并立即进入 NO\_INIT 状态。
- 对于 SFB B\_RCV，可能另外的数据被接收的同时发送了错误序列帧，这将在本地当作无效处理。
- SFB U\_RCV 立即进入 NO\_INIT 状态。

发送块 SFB、USEND、BSEND 反映如下：

- 如发送块 SFB BSEND 已进行工作但未完成，当暖/冷启动时产生一个错误序列帧，之后就进入到 NO\_INIT 状态，之后到达的应答被废弃。
- 当暖/冷启动时，假如 SFB\_SEND 发送或接收到错误序列帧，它进入 NO\_INIT 状态。
- 其它什么情况和什么时候 SFB 仅发送消息（例如 SFB BSEND），本地处理失败后，SFB 都进入 NO\_INIT 状态。

### SFB 单边数据传输时的暖/冷启动

假设在连接建立后在通讯对象服务器端处理，也就是说由它来在任何时间处理工作或输出消息。

SFB 发送工作、等待应答时发生完全启动时的反映：

当前处理失败，并且之后 CFB 进入 NO\_INIT 状态。如果一个应答在暖/冷启动前到来将被废弃。

一个新的工作在前面的应答被接收之前可能被发送。

用于输出或接收消息的 SFB 反映如下：

- 当前处理失败，并且之后 CFB 进入 NO\_INIT 状态。
- 对于 SFB USTATUS，在 NO\_INIT 期间和 DISABLED 状态到来的消息将被废弃。

### 对于热启动的响应

仅在暖/冷启动时 SFB 需组态连接进入 NO\_INIT 状态，这意味在热启动时它们向用户功能一样响应可继续运行。

### 存储器清除时的响应

一个存储器清除总是引起所有的连接终止，之后的暖/冷启动只能器定用户程序类型，所有 SFB 需组态 S7 连接（即使仍存在）都进入到 NO\_INIT 状态。作为连接终止的响应对方模板存储区的块不能复位，并且转换至 IDLE，ENABLED 或 DISABLED 状态。



## 19.3 SFB 对于故障的响应

下列描述适用于 S7-400 的 SFB 需组态连接对于故障的响应。

如连接被终止，SFB 的反映依内部状态而定。

当连接终止被检测到，相关块进入空闲，使能或非使能状态，SFB 响应如下：

- 在 ERROR 和 STATUS 端显示通讯故障的代码。
- 在下次调用时，块返回到初始状态并再次检查连接。

下列情况 SFB 通讯不进入空闲，使能或非使能状态：

- 终止处理，立即进入错误状态在下次调用时在 ERROR 和 STATUS 参数端输出通讯故障的代码。
- 在下次调用时，块进入空闲，使能或非使能状态，在空闲，使能状态再次检查连接。

假如与此同时连接被设定好，处理将再次进行。

### 关机

有备用电池的之后的重新启动将把已建立的连接终止。这应用于所有的涉及的块。

有备用电池的关机之后的自动暖冷启动，在启动时终止连接。

在没有备用电池的自动暖冷启动的特殊情况，在恢复供电后执行一个存储器清除，SFB 需组态连接反映如下面所述“SFB 需组态连接的启动路线”。

### 操作模式改变时的响应

如果操作模式在运行、停止、启动和保持之间变化，通讯的 SFB 保持它的当前状态（暖/冷启动例外，此时其为 NO\_INIT 状态）。这适用于无论 SFB 双边或单边通讯。

### 用于程序的错误接口

如在处理 SFB 通讯时出现故障，总是在 ERROR 端显示故障状态。除了在 ERROR 端被设定为 1 状态外，与此同时还在 STATUS 端显示相应故障的代码。你可在程序中确定这类信息。

可能错误举例：

- 收集发送数据时的错误。
- 拷贝接收数据到接收区域时的错误（例如访问不存在的数据块）。
- 发送数据长度与对方 SFB 所规定的接收数据长度不匹配。

## 19.4 用 SFB 8/FB 8 “U\_SEND” 非协调发送数据

### 简述

SFB/FB 8 “U\_SEND” 发送数据到远端 SFB/FB “U\_RCV”。这种过程是非协调发送的发送到远端对象的。这意味着这种数据传输的执行无须对方 SFB/FB 应答。

S7-300: 数据在 REQ 的上升沿发送数据。参数 R\_ID, ID 和 SD\_1 在每个 REQ 的上升沿被传输。在工作结束后, 你可以给 R\_ID, ID 和 SD\_1 分配新值。

S7-400: 数据在 REQ 的上升沿发送数据。参数于相应的 SD\_1 到 SD\_4 来发送数据, 但不一定 4 个参数段都要使用。

然而, 必须确保你定义的 SD\_1 到 SD\_4 区域与 RD\_1 到 RD\_4 (相应对方 SFB/FB “U\_RCV”) 相一致。

- 数量
- 长度
- 数据类型

在两边的 R\_ID 也必须相同。

传输完成后状态参数 DONE 被设定为值 1。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
ID	INPUT	WORD	D, L, 常数	ID 寻址参数, 参见 S7 通讯的 SFC 和 SFC 参数有关章节
R_ID	INPUT	DWORD	I, Q, M, D, L, 常数	R_ID 寻址参数, 参见 S7 通讯的 SFC 和 SFC 参数有关章节
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数: 0: 作业未开始或仍在运行 1: 作业被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数, 错误显示: ERROR=0 STATUS 有下列值: 0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息

参数	声明	数据类型	存储区域	说明
S7-300: RD_1 S7-400: RD_i (1= $\leq$ i= <4)	输入_ 输出	ANY	M, D, T, Z I, Q, M, D, T, C	指向发送数据区的指针, 不仅有下列数据类型被允许: BOOL (位区域不允许), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER。 <b>注释:</b> 如用 ANY 指针访问一个数据块, DB 的号码必须被首先指明 (例如: P#DB10.DBX5.0 BYTE 10)。

## 故障信息

故障	STATUS(十进制)	解释
0	11	警告: 前面作业未完成, 新作业不能生效。
0	25	通讯开始, 作业正处理中。
1	1	通讯故障, 例如: <ul style="list-style-type: none"> <li>• 连接简述未装载 (本地或远端)</li> <li>• 连接中断 (例如: 电缆, 电源失灵, CP 处于 STOP 模式)</li> <li>• 与对方的连接没有建立</li> </ul>
1	4	涉及发送数据段 SD_I 的数据长度或类型错误。
1	10	访问本地用户存储故障 (例如访问已删除数据块)
1	12	当 SFB 调用时: <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB “U_SEND” 不匹配</li> <li>• 无背景数据块而是全局数据块</li> <li>• 无背景数据块 (没有从编程器下载新的数据块)</li> </ul>
1	18	R_ID 在连接中已存在
1	20	存储器不足。 <ul style="list-style-type: none"> <li>• H 系统: 在更新过程中的 SFB 首次调用</li> <li>• S7-300 <ul style="list-style-type: none"> <li>- 最大并行工作数超出</li> <li>- 在 CPU 运行时背景被重复装载</li> <li>- 可能在首次调用时</li> </ul> </li> </ul>

## 数据连续性

S7-300: 未确保连续性, 你可在每次工作完成前只发送数据 SD\_1。这种情况在每次参数 DONE 的状态值为 1 时候。

S7-400: 当发送作业被激活时 (REQ 为上升沿) 来自程序的 RD\_I 指向的数据被发送。在块调用后你可写这些区域而无需打断当前发送数据。

## 注释

发送操作仅在 DONE 参数段被设定为 1 时结束。

## 19.5 用 SFB/FB 9 “U\_RECV” 非协调接收数据

### 简述

SFB/FB U\_RECV 异步接收远端对象 U\_RECV 的数据，并将其拷贝到组态的接收区。

当输入 EN\_R 有 1 信号时，块准备接受当 EN\_R 为 0 时可取消该工作。

S7-300: 参数 R\_ID, ID, RD\_1 在每次 EN\_R 有 1 信号时可得到应用。在块结束后，你可以给 R\_ID, ID, RD\_1 赋新值。

S7-400: 接收数据区依照参数 RD\_1 到 RD\_4 而定。

你必须，然而，确保你定义的 RD\_1 到 RD\_4 区域与 SD\_1 到 SD\_4（相应对方 SFB/FB “U\_SEND”）相一致：

- 数量
- 长度
- 数据类型

成功完成数据传输后，NDR 状态参数被设定为 1。

在两边的 R\_ID 也必须相同。

参数	声明	数据类型	存储区域	说明
EN_R	INPUT	BOOL	I, Q, M, D, L, 常数	使能的接收控制参数，输入被置位时准备接受。
ID	INPUT	WORD	M, D, 常数	寻址参数 ID，参见 SFC 和 SFC S7 通讯常见参数章
R_ID	INPUT	DWORD	I, Q, M, D, L, 常数	R_ID 寻址参数，参见 SFC 和 SFC S7 通讯常见参数章
NDR	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数： 0: 作业未开始或仍在运行 1: 作业被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数，错误显示： ERROR=0 STATUS 有下列值： 0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息

参数	声明	数据类型	存储区域	说明
S7-300: RD_1 S7-400: RD_i (1≤i≤4)	IN_ OUT	ANY	M, D, T, Z I, Q, M, D, T, C	指向发送数据区的指针，不仅有下列数据类型被允许： BOOL（位区域不允许）， BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER。 <b>注释：</b> 如用 ANY 指针访问一个数据块，DB 的号码必须被首先指明（例如：P#DB10.DBX5.0 BYTE 10）。

## 故障信息

故障	STATUS(十进制)	解释
0	9	覆盖警告：新的数据覆盖旧的数据。
0	11	警告：前面作业未完成，新作业不能生效。
0	25	通讯开始，作业正处理中。
1	1	通讯错误，例如： <ul style="list-style-type: none"> <li>• 连接简述未装载（本地或远端）</li> <li>• 连接中断（例如：电缆，电源失败，CP 处于 STOP 模式）</li> <li>• 与对方的连接没有建立</li> </ul>
1	4	涉及发送数据段 SD_I 的数据长度或类型故障。
1	10	访问本地用户存储故障（例如访问已删除数据块）
1	12	当 SFB 调用时： <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB “U_RCV” 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>
1	18	R_ID 在连接中已存在
1	19	相应的 SFB/FB U_SEND 发送速度大于 SFB/FB “U_RCV” 拷贝到接收区的速度。
1	20	存储器不足。 <ul style="list-style-type: none"> <li>• H 系统：在更新过程中的 SFB 首次调用</li> <li>• S7-300 <ul style="list-style-type: none"> <li>– 最大并行工作数超出</li> <li>– 在 CPU 运行时背景被重复装载</li> <li>– 可能在首次调用时</li> </ul> </li> </ul>

## 数据连续性

记住下列要点将确保接收数据的连续性

- S7-300：在每次 NDR 改变为 1 后，你再将 FB 9 的 EN\_R 段设定为 1，可确保新的数据不会覆盖旧的数据
- 在你将 EN\_R 设定为 1 时，确定 RD\_1 的接收已完成。

S7-400: 当 NDR 参数改变为 1 时, 会有新的数据到来 (RD\_I), 新的数据块被调用会导致新的数据区被填写。如你想防止此类事件, 你应调用 SFB 9 U\_RCV (在每个块的处理周期) 保持 EN\_R 段为 0 直到接收数据被处理完毕。

## 19.6 用 SFB/FB 12 “B\_SEND” 发送段数据

### 简述

SFB/FB 发送数据给远端对象 SFB/FB “B\_RCV”。通过这种数据传输类型, 可以有比其它 SFB/FB 需组态连接更多的数据在通讯对象之间被传递, 即对 S7-300 有 32K 字节, 对 S7-400 有 64K 字节。

数据区按段被传送。每个段被单独发送到对象。最后一个段被接收对象所应答, 无论是否调用的是 SFB/FB “B\_RCV”。

S7-300: 当 REQ 有上升沿时发送工作被激活。参数 R\_ID, ID, SD\_1 和 LEN 在 REQ 有上升沿时被传送。当工作结束后, 你可以给参数 R\_ID, ID, SD\_1 和 LEN 赋新值。数据块的段数据的传输必须在程序中被周期的调用。

最大数据长度的起始地址在 SD\_1 中指定。然后根据 LEN 所制定的数据长度来决定。

S7-400: 当输入控制 REQ 有上升沿时发送工作被激活。来自程序指定的发送数据被异步的处理发送数据。

最大数据长度的起始地址在 SD\_1 中指定。你可根据 LEN 所制定的数据长度来决定。在这种情况下 LEN 代替了 SD\_1 指定的长度。

在两边的相应 SFB/FB 的 R\_ID 必须相同。

如在控制输入 R 上得到一个上升沿, 当前的数据传输则被取消。

传输被成功的完成后将在 DONE 上得到一个值 1。

一个新的工作不会被处理直到前面的工作通过参数 DONE, ERROR 得到 1 表示结束。

由于异步的传输方式, 一个新的传输仅在前面的工作被完成之后才能初始化, 当 SFB/FB B\_SEND 被调用时, 显示状态值 7 (见下列表格)。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
R	INPUT	BOOL	I, Q, M, D, L, 常数	复位控制参数, 参见 SFC 和 SFC S7 通讯常见参数章
ID	INPUT	WORD	M, D, 常数	寻址参数 ID, 参见 SFC 和 SFC S7 通讯常见参数章
R_ID	INPUT	DWORD	I, Q, M, D, L, 常数	R_ID 寻址参数, 参见 SFC 和 SFC S7 通讯常见参数章 通过 CP 441 或 S5 或其它设备的连接, 其包含了于远端设备地址信息, 详见 CP 441 的信息

参数	声明	数据类型	存储区域	说明
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数: 0: 工作未开始或仍在运行 1: 工作被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数, 错误显示: ERROR=0 STATUS 有下列值: 0000H: 无警告和故障 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个故障。STATUS 提供详细信息
SD_1	IN_ OUT	ANY	S7-300: M, D S7-400: I, Q, M, D, T, Z	指向发送数据区的指针, 仅有下类数据类型被允许: BOOL (位区域不允许), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER。 <b>注释:</b> 如用 ANY 指针访问一个数据块, DB 的号码必须 被首先指明 (例如: P#DB10.DBX5.0 BYTE 10)。
LEN	IN_ OUT	WORD	I, Q, M, D, L	数据区域长度按字节来指明

### 故障信息

下表包含了使用 SFB/FB 12 时在 ERROR 和 STATUS 可能出现的错误信息。

ERROR	STATUS (十进制)	解释
0	11	警告: 前面工作未完成, 新工作不能生效。
0	25	通讯开始, 工作正处理中。
1	1	通讯故障, 例如: <ul style="list-style-type: none"> <li>• 连接简述未装载 (本地或远端)</li> <li>• 连接中断 (例如: 电缆, 电源失败, CP 处于 STOP 模式)</li> <li>• 与对方的连接没有建立</li> </ul>
1	2	收到来自对方 SFB/FB 的否定应答, 功能不能被执行
1	3	连接上的 R_ID 不可识别或接收块未被调用
1	4	在发送数据点 SD_1 上的数据长度或者 LEN 指定为 0 出现故障
1	5	复位请求被执行
1	6	对方 SFB/FB 处于 DISABLED 模式 (EN_R 设定为 0), 检查 BSEND 块的输入参数是否连续。

ERROR	STATUS (十进制)	解释
1	7	对方 SFB/FB 处于故障状态 最后数据传输时接收块没有被调用。
1	8	访问远端目标的用户存储器被拒绝：相应 SFB/FB BRCV 指定的目标区太小 相应的 SFB/FB BRCV 报告 ERROR=1，STATUS=4 或者 ERROR=1，STATUS=10。
1	10	访问本地用户存储器不可能（例如，访问已删除的数据块）
1	12	当 SFB 调用时： <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB 12 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>
1	18	R_ID 在连接中已存在
1	20	存储器不足。 <ul style="list-style-type: none"> <li>• H 系统：在更新过程中的 SFB 首次调用</li> <li>• S7-300 <ul style="list-style-type: none"> <li>- 最大并行工作数超出</li> <li>- 在 CPU 运行时块被重复装载</li> <li>- 可能是第一次调用</li> </ul> </li> </ul>

### 数据连续性

未确保连续性，你可在每次工作完成后发送当前数据 SD\_1。这种情况在每次参数 DONE 的状态值为 1 时候。

## 19.7 用 SFB/FB 13 “B\_RCV” 接收段数据

### 简述

SFB/FB “B\_RCV” 接收远端对象 SFB/FB “B\_SEND” 的数据。待每次接收段数据后发送一个应答给对方 SFB/FB 及刷新 LEN 参数。

当控制输入 EN\_R 得到 1 时被调用，该块接收数据。EN\_R 得到 0 时，工作被取消。

最大数据长度的起始地址在 RD\_1 中指定。然后根据 LEN 所制定的数据长度来决定。

S7-300：参数 R\_ID，ID，SD\_1 和 LEN 在 REQ 有上升沿时被接收。当工作结束后，你可以给参数 R\_ID，ID，SD\_1 和 LEN 赋新值。数据块的段数据的传输必须在程序中被周期的调用。

S7-400：来自用户存储器的数据配方被程序异步的执行。

在两边的相应 SFB/FB 的 R\_ID 必须相同。

所有接收的数据端无误则在参数 NDR 上显示 1，接受的数据保持不变直到 SFB/FB 13 的 EN\_R 上得到 1。



如在异步接收数据时调用块，将导致警告病在输出 STATUS 上显示，如果控制输入 EN\_R 值为 0，则接收被终止 SFB/FB 回到初始状态。

参数	声明	数据类型	存储区域	说明
EN_R	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
ID	INPUT	WORD	M, D, 常数	寻址参数 ID, 参见 SFC 和 SFC S7 通讯常见参数章
R_ID	INPUT	DWORD	I, Q, M, D, L, 常数	R_ID 寻址参数, 参见 SFC 和 SFC S7 通讯常见参数章 通过 CP 441 或 S5 或其它设备的连接, 其包含了于远端设备地址信息, 详见 CP 441 的信息
NDR	OUTPUT	BOOL	I, Q, M, D, L	NDR 状态参数: 0: 工作未开始或仍在运行 1: 工作被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数, 错误显示: ERROR=0 STATUS 有下列值: 0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
RD_1	IN_ OUT	ANY	S7-300: M, D S7-400: I, Q, M, D, T, C	指向发送数据区的指针, 仅有下类数据类型被允许: BOOL (位区域不允许), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER。 <b>注释:</b> 如用 ANY 指针访问一个数据块, DB 的号码必须被首先指明 (例如: P#DB10.DBX5.0 BYTE 10)。
LEN	IN_ OUT	WORD	I, Q, M, D, L	数据区域长度按字节来指明

#### 故障信息

下表包含了使用 SFB/FB 13 时在 ERROR 和 STATUS 可能出现的错误信息。

ERROR	STATUS (十进制)	解释
0	11	警告：前面工作未完成，新工作不能生效。
0	17	警告：异步接收块
0	25	数据已被接收。LEN 显示已收集数量（字节位单位）
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>• 连接简述未装载（本地或远端）</li> <li>• 连接中断（例如：电缆，电源失败，CP 处于 STOP 模式）</li> <li>• 与对方的连接没有建立</li> </ul>
1	2	功能不能被执行（协议故障）
1	4	接收区域点 RD_1 的数据长度或类型故障，发送区长于接收区。
1	5	复位请求接收，传送不能完成
1	8	错误访问目标 SFB/FB “BRCV”，在最后一个发送数据断后报告 ERROR=1，STATUS=4 或者 ERROR=1，STATUS=10。
1	10	访问本地用户存储器不可能（例如，访问已删除的数据块）
1	12	当 SFB 调用时： <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB 13 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>
1	18	R_ID 在连接中已存在
1	20	存储器不足。 <ul style="list-style-type: none"> <li>• H 系统：在更新过程中的 SFB 首次调用</li> <li>• S7-300 <ul style="list-style-type: none"> <li>- 最大并行工作数超出</li> <li>- 在 CPU 运行时块被重复装载</li> <li>- 可能是第一次调用</li> </ul> </li> </ul>

### 数据连续性

记住下列要点将确保接收数据的连续性：

再你将 EN\_R 设定为 1 时，确定 RD\_1 的接收已完成。

### 接收数据的特殊情况（仅 S7-400 适用）

如一个用 BRCV 块接收的 CPU 准备好接收数据（就是说，EN\_R=1 已经被设定），在接收相应发送块的数据完毕之前进入到 CPU，此前首次发送数据段已完成，将会发生下列情况：

- 接收 CPU 进入到 STOP 时首次工作的数据将完整进入接收区。
- 对象 SFB B\_SEND 接收一个肯定的应答。
- 任何此外的 BSEND 工作将不再能被进入 STOP 的 CPU 所接受。
- 只要 CPU 保持 STOP 模式，NDR 和 LEN 都将得到值 0。

为防止接收信息丢失，你应当给接收 CPU 热启动并调用 SFB 13 “B\_RCV” 时 EN\_R=1。

## 19.8 用 SFB/FB 15 “PUT” 写数据到远端 CPU

### 简述

使用 SFB/FB 15 “PUT”，你可以写数据到远端 CPU。

S7-300：REQ 为上升沿数据被发送。参数 ID，ADDR\_1 和 SD\_1 在 REQ 有上升沿时被发送。当工作结束后，你可以给参数 ID、ADDR\_1 和 SD\_1 赋新值。

S7-400：控制输入 REQ 得到 1 时 SFB 开始工作。将写区域（ADDR\_I）和数据 SD\_I 在此过程中被发送到远端 CPU。

远端对象保留这些请求数据在相应地址里并发送一个执行应答。

确保参数 ADDR\_I 和 SD\_I 定义的区域在数量、长度和数据类型上匹配。

如无任何错误发生，会在下次 SFB/FB 调用时以状态参数 DONE 为 1 来指示。

写作业只能在最后一次工作完成后被激活。

远端 CPU 可处于 RUN 或 STOP 模式。

在数据被写时如有访问错误和警告将输出在 ERROR 和 STATUS，可访问此得到相应结果。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
ID	INPUT	WORD	I, M, D, 常数	寻址参数 ID，参见 SFC 和 SFC S7 通讯常见参数一章
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数： 0：作业未开始或仍在运行 1：作业被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数，错误显示： ERROR=0 STATUS 有下列值： 0000H：无警告和错误 <>0000H：警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
S7-300: ADDR_1 S7-400: ADDR_i (1≤i≤4)	IN_ OUT	ANY	M, D	指向远端 CPU 被写数据的区域指针

参数	声明	数据类型	存储区域	说明
S7-300: SD_1 S7-400: SD_i (1≤i≤4)	IN_ OUT	ANY	M, D, T, Z I, Q, M, D, T, C	指向发送数据区的指针，仅有下列数据类型被允许： BOOL（位区域不允许）， BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER。 <b>注释：</b> 如用 ANY 指针访问一个数据块，DB 的号码必须被首先指明（例如：P#DB10.DBX5.0 BYTE 10）。

## 故障信息

下表包含了使用 SFB/FB 15 时在 ERROR 和 STATUS 可能出现的故障信息。

ERROR	STATUS（十进制）	解释
0	11	警告：前面工作未完成，新工作不能生效。
0	25	数据已开始。工作进行中
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>• 连接简述未装载（本地或远端）</li> <li>• 连接中断（例如：电缆，CPU 关闭，CP 处于 STOP 模式）</li> <li>• 与对方的连接没有建立</li> </ul>
1	2	收到对象设备否定应答，通讯功能不能执行
1	4	接收区域点 SD_1 的数据长度或类型错误，发送区长于接收区。
1	8	访问远端 CPU 故障。
1	10	访问本地用户存储器不可能（例如，访问已删除的数据块）
1	12	当 SFB 调用时： <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB 15 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>
1	20	存储器不足。 <ul style="list-style-type: none"> <li>• H 系统：在更新过程中的 SFB 首次调用</li> <li>• S7-300 <ul style="list-style-type: none"> <li>- 最大并行工作数超出</li> <li>- 在 CPU 运行时块被重复装载</li> <li>- 可能是第一次调用</li> </ul> </li> </ul>

## 对于 S7-300 的数据连续性

未确保连续性，你可在每次工作完成后发送当前数据 SD\_I。这种情况在每次参数 DONE 的状态值为 1 时候。

### 对于 S7-400 的数据连续性

控制输入 REQ 得到上升沿时发送作业开始作业。来自程序将被发送的数据 SD\_I 在此过程中被发送到远端 CPU。在块调用后，你可写这些数据区而不必中断当前发送数据。

#### 注释

发送作业仅在 DONE 状态参数得到值 1 时完成。

## 19.9 用 SFB/FB 14 “GET” 读远端 CPU 数据

### 简述

你可使用 SFB/FB 14 “GET”，读取来自远端 CPU 的数据。

S7-300: REQ 为上升沿数据被接收。参数 ID, ADDR\_1 和 RD\_1 在 REQ 有上升沿时被接收。当工作结束后，你可以给参数 ID, ADDR\_1 和 RD\_1 赋新值。

S7-400: 控制输入 REQ 得到 1 时 SFB 开始工作。将来自远端 CPU 的读区域 (ADDR\_I) 和数据 RD\_I 在此过程中被接收。

远端对象回应数据。

在下此调用时接收数据被存在组态接收区域 (RD\_I)。

确保参数 ADDR\_I 和 SD\_I 定义的区域在数量、长度和数据类型上匹配。

成功完成数据传输后，NDR 状态参数被设定为 1。

读工作只能在最后一次工作完成后被激活。

远端 CPU 可处于 RUN 或 STOP 模式。

在数据被读取时如有访问错误和警告将输出在 ERROR 和 STATUS，可访问此得到相应结果。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
ID	INPUT	WORD	I, M, D, 常数	寻址参数 ID, 参见 SFC 和 SFC S7 通讯常见参数一章
NDR	OUTPUT	BOOL	I, Q, M, D, L	NDR 状态参数: 0: 作业未开始或仍在运行 1: 作业被无误执行

参数	声明	数据类型	存储区域	说明
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数，错误显示： ERROR=0 STATUS 有下列值： 0000H：无警告和错误 <>0000H：警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
S7-300: ADDR_1 S7-400: ADDR_i (1≤i≤4)	IN_ OUT	ANY	S7-300: M, D S7-400: I, Q, M, D, T, C	指向远端 CPU 被读取数据的区域指针
S7-300: SD_1 S7-400: SD_i (1≤i≤4)	IN_ OUT	ANY	S7-300: M, D, T, Z S7-400: I, Q, M, D, T, C	指向读取数据区的指针，仅有下类数据类型被允许： BOOL（位区域不允许）， BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER。 注释： 如用 ANY 指针访问一个数据块，DB 的号码必须被首先指明（例如： P#DB10.DBX5.0 BYTE 10）。

## 故障信息

ERROR	STATUS（十进制）	解释
0	11	警告：前面作业未完成，新作业不能生效。
0	25	数据已开始。作业进行中
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>• 连接简述未装载（本地或远端）</li> <li>• 连接中断（例如：电缆，电源失败，CP 处于 STOP 模式）</li> <li>• 与对方的连接没有建立</li> </ul>
1	2	收到对象设备否定应答，通讯功能不能执行
1	4	接收区域点 SD_1 的数据长度或类型故障，发送区长于接收区。
1	8	访问远端 CPU 故障。
1	10	访问本地用户存储器不可能（例如，访问已删除的数据块）
1	12	当 SFB 调用时： <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB 14 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>

ERROR	STATUS (十进制)	解释
1	20	存储器不足。 <ul style="list-style-type: none"> <li>• H 系统：在更新过程中的 SFB 首次调用</li> <li>• S7-300               <ul style="list-style-type: none"> <li>- 最大并行工作数超出</li> <li>- 在 CPU 运行时块被重复装载</li> <li>- 可能是第一次调用</li> </ul> </li> </ul>

### 数据连续性

为确保数据连续性，应遵守以下原则：

在初始化另一项作业时，应确保接收区域 RD\_I 当前可用。

## 19.10 用 SFB 16 “PRINT” 发送数据到打印机

### 简述

SFB 16 “PRINT” 发送数据和指令格式到一个远端打印机。例如借助于 CP 441。当控制输入 REQ 为上升沿，格式描述 (FORMAT) 和数据 (SD\_I) 被发送到 ID 和 PRN\_BR 指定的打印机。

如你没有使用四个发送数据区，需确保首个区域 SD\_1 参数被赋值，然后依次第二个（如存在）SD\_2、第三个 SD\_3...

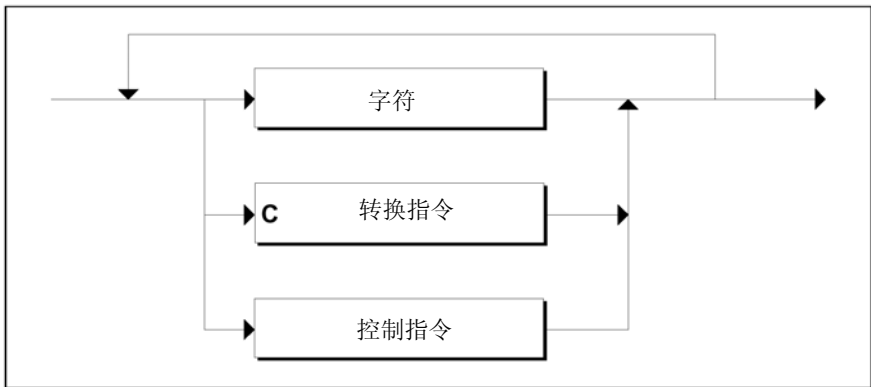
工作成功的执行后将在 DONE 状态参数指示，错误在 ERROR 和 STATUS 参数中指明。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
ID	INPUT	WORD	M, D, 常数	寻址参数 ID，参见 SFC 和 SFC S7 通讯常见参数章
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数： 0：工作未开始或仍在运行 1：工作被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数，错误显示： ERROR=0 STATUS 有下列值： 0000H：无警告和错误 <>0000H：警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
PRN_NR	IN_ OUT	BYTE	I, Q, M, D, L	打印机号码

参数	声明	数据类型	存储区域	说明
FORMAT	IN_ OUT	STRING	I, Q, M, D, L	格式简述
SD <sub>i</sub> (1≤i≤4)	IN_ OUT	ANY	M, D, T, C	指向发送数据区的指针，仅有下列数据类型被允许： BOOL（位区域不允许）， BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER。 注释： 如用 ANY 指针访问一个数据块，DB 的号码必须被首先指明（例如： P#DB10.DBX5.0 BYTE 10）。

**IN\_OUT 参数格式**

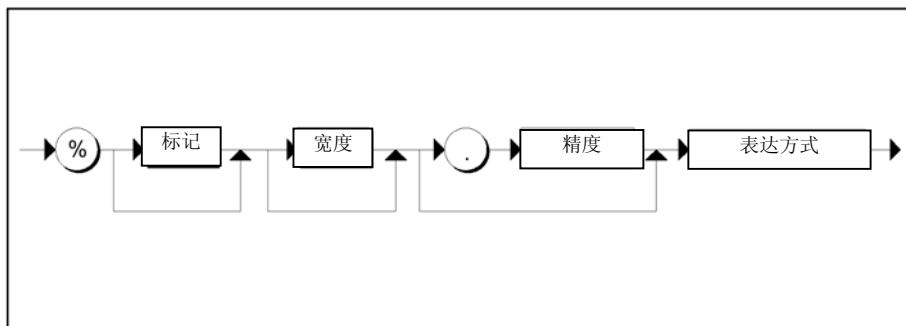
FORMAT 字符串包含可打印的字符和格式元素。如下列格式：



对于每个要打印的发送区域（SD<sub>1</sub> 到 SD<sub>4</sub>）必须一个为转换为 FORMAT 格式的指令。被应用的转换为 FORMAT 格式的指令（SD<sub>i</sub>）有如下。公式字符和指令可按下列任何顺序。

- 字符
  - 允许字符
    - 所有可打印字符
    - \$\$ (美元字符)、\$' (单引号)、\$L 和\$I (换行)、\$P 和\$p (页)、\$R 和\$r (回车)、\$T 和\$t (跳格)





转换指令语法表

转换指令元素	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 标准显示输出</li> <li>• n 精确 N 个字符输出。如输出为右对齐输出，可能导致输出空白，左对齐输出可使空白之后出现字符。</li> </ul>
精度	精度与 A, D, F 和 R 表达方式（见下列表格）相关 <ul style="list-style-type: none"> <li>• 无 标准表达方式输出</li> <li>• 0 在 F 和 R 表达方式时无十进制点的输出</li> <li>• n               <ul style="list-style-type: none"> <li>- F 和 R: 输出十进制 N 位</li> <li>- A 和 D (日期): 年份的位数, 可使用 2 或 4</li> </ul> </li> </ul>
表现	下表包含: <ul style="list-style-type: none"> <li>• 可能的表达方式方式</li> <li>• 每个表达方式的数据类型</li> <li>• 每个表达方式的标准格式（如在 FORMAT 参数无宽度和精度要求的标准表达方式）和他们的最大长度。</li> </ul>

下表包含可能的 FORMAT 参数规定的表达方式方式。

表现	可能数据类型	例子	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	
	CHAR 的 ARRAY	KLMNOP	字符的数量	
	BYTE 的 ARRAY			
D, d	DATE	1996-07-25	10	-
	DWORD			

表现	可能数据类型	例子	长度	注释
F, f	REAL	0.345678	8	-
	DWORD			
H, h	包括 BYTE 的 ARRAY 的所有数据类型	取决于数据类型	取决于数据类型	16 进制表示
I, i	INT	- 32768	最大 6	-
	WORD	- 2147483648	最大 11	
N, n	WORD	文本输出	-	相应发送数据区 SD_1 含有一个将被打印文本的参考（号码），该文本在模板（例如 CP 441）中创建可打印字符串，如在相应参考中无文本发现，则打印*****。
R, r	REAL	0.12E-04	8	-
	DWORD			
S, s	STRING	文本输出		-
T, t	TIME	2d_3h_10m_5s_250ms	MAX.21	如错误发生。输出*****
	DWORD			
U, u	BYTE	255	MAX.3	-
	WORD	65535	MAX.5	
	DWORD	4294967295	MAX.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	-

表中所列为表达方式的<sup>最大长度</sup>，实际可能会短一些。

#### 注释

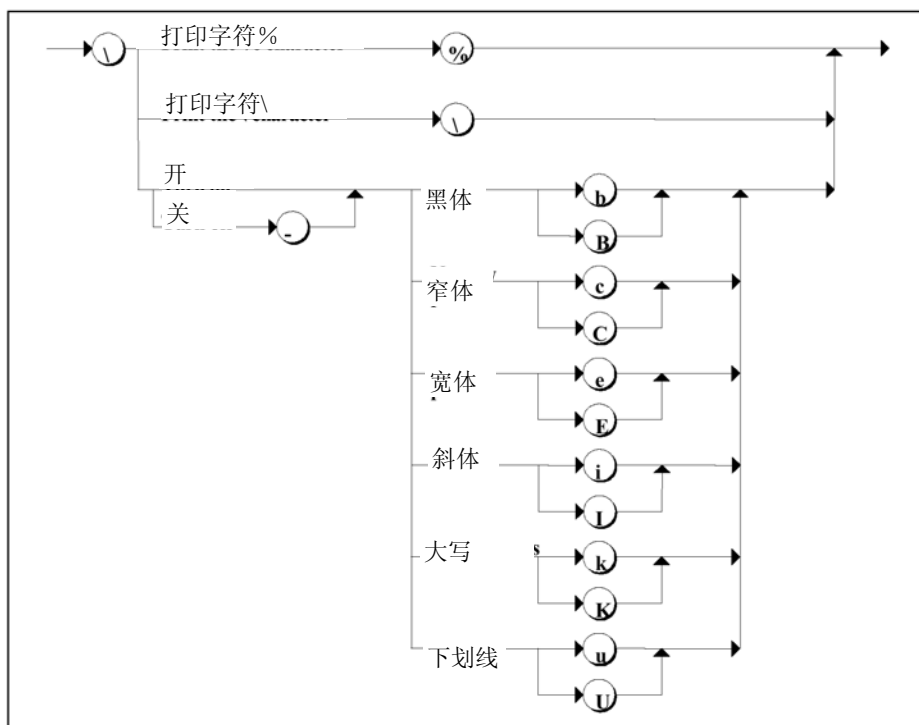
在数据类型 C 和 S，取决于所使用打印机有下列要点：

- 可打什么字符
- 什么是打印机不可打印字符，除非打印机驱动含有这些字符。

- 控制指令

使用控制指令你应做到下点：

- 打印字符%和\
- 改变打印机设定



### 控制指令语法表

如你想关闭，例如，一个不能使用的字体或者一个打印机不能执行的功能，控制指令将被忽略。下列表格包含 FORMAT 参数可能导致的错误。

故障	打印输出
转换指令不能被执行	*字符被输出根据默认表现所定义的长度（最大）或宽度。
规定宽度太小	在表现 A.C.D.N.S.T 和 Z，许多规定字符按选定宽度输出，对于其它表现，超出字符宽度*字符被输出。
过多转换指令	无发送点 SD_I 的转换指令将被忽略
过少转换指令	发送区无转换指令不能输出
未定义或不支持转换指令	*****被输出
未完成转换指令	*****被输出
未定义或不支持控制指令	控制指令与语法不符合，忽略。

### 故障信息

下表包含了使用 ERROR 和 STATUS 参数可打印的 SFB 16 “PRINT” 的所有可能出现的故障信息。

ERROR	STATUS (十进制)	解释
0	11	警告：前面作业未完成，新作业不能生效。
0	25	数据已开始。作业进行中
1	1	通讯错误，例如： <ul style="list-style-type: none"> <li>• 连接简述未装载（本地或远端）</li> <li>• 连接中断（例如：电缆，CPU 关闭，CP 处于 STOP 模式）</li> <li>• 与对方的连接没有建立</li> </ul>
1	2	收到对象设备否定应答，通讯功能不能执行
1	3	PRN_NR 不可被 ID 规定的打印机识别
1	4	参数 FORMAT 错误或者发送区 SD_I 的数据类型或长度错误
1	6	远端打印处于离线状态
1	7	远端打印机不在正确状态
1	10	不能访问本地用户存储器（例如访问已删除的数据块）
1	13	参数 FORMAT 错误
1	20	存储器空间不足 H-系统：过程更新时 SFB 被调用

#### 传送数据的数量

可被传送到远端打印机的传送数量不可超过最大数量。

最大数量计算如下：

最大量 = 420 - 格式

格式为参数 FORMAT 的字节数。被打印数据可以是一个或多个数据区。

## 19.11 用 SFB 19 “START” 在远端设备上初始化一个暖或冷启动

### 简述

如在控制输入 REQ 上得到一个上升沿, SFB 19 “START”, 将对根据 ID 所标地址的远端设备进行一个暖或冷启动。如远端设备为容错系统, 再启动效果由参数 PI\_NAME 的设定来决定: 启动请求对该系统的一个或所有 CPU 执行。远端设备必须符合下面所列的条件:

- CPU 应处于 STOP 模式
- CPU 的钥匙开关应被设定为 RUN 或者 RUN-P

一旦暖或冷启动完成, 设被改变为 RUN 模式而且发送一个肯定的执行应答。当肯定的执行应答被确认, 状态参数 DONE 被设定为 1。如任何错误发生, 将在 ERROR 和 STATUS 上指示出来。

一个再次的暖或冷启动可以在完成了暖或冷启动的同一个设备上再次被执行。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
ID	INPUT	WORD	I, Q, M, D, 常数	寻址参数 ID, 参见 SFC 和 SFC S7 通讯常见参数章
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数: 0: 工作未开始或仍在运行 1: 工作被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数, 错误显示: ERROR=0 STATUS 有下列值: 0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
PI_NAME	IN_ OUT	ANY	I, Q, M, D, T, C	指向将被启动程序名称 (ASCII 码) 的区域指针。名称应不得超出 32 个字符, 对于 S7 设备, 为 P_PROGRAM。对于 H 系统, 可以为下列参数: <ul style="list-style-type: none"> <li>• P_PROGRAM(启动对所有系统中的 CPU 有效)</li> <li>• P_PROG_0 (启动对机架 0 的 CPU 有效)</li> <li>• P_PROG_1 (启动对机架 1 的 CPU 有效)</li> </ul>
ARG	IN_ OUT	ANY	I, Q, M, D, T, C	执行条款 <ul style="list-style-type: none"> <li>• 如不在 ARG 上设定值, 则在远端设备上进行一次暖启动。</li> <li>• 如赋值 C, 一个冷启动在远端设备上执行 (该设备应支持此功能)。</li> </ul>

参数	声明	数据类型	存储区域	说明
IO_STATE	IN_ OUT	BYTE	I, Q, M, D, L	暂时未使用。 如你通讯的对象为 S7 可编程控制设备, 则此项不用设定。

## 故障信息

下表包含了使用 ERROR 和 STATUS 参数可以输出得 SFB 19 的所有可能出现的故障信息。

ERROR	STATUS (十进制)	解释
0	11	警告: 前面作业未完成, 新作业不能生效。
0	25	数据已开始。作业进行中
1	1	通讯故障, 例如: <ul style="list-style-type: none"> <li>• 连接简述未装载 (本地或远端)</li> <li>• 连接中断 (例如: 电缆, 电源失败, CP 处于 STOP 模式)</li> <li>• 与对方的连接没有建立</li> </ul>
1	2	收到对象设备否定应答, 通讯功能不能执行
1	3	不能识别的 PI_NAME
1	4	PI_NAME 或 ARG 涉及的数据长度或类型故障
1	7	远端设备不能完成启动
1	10	访问本地用户存储器不可能 (例如, 访问已删除的数据块)
1	12	当 SFB 调用时: <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB 19 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>
1	20	存储器不足。 <ul style="list-style-type: none"> <li>• H 系统: 在更新过程中的 SFB 首次调用</li> <li>• S7-300 <ul style="list-style-type: none"> <li>- 最大并行工作数超出</li> <li>- 在 CPU 运行时块被重复装载</li> <li>- 可能是第一次调用</li> </ul> </li> </ul>

## 19.12 用 SFB 20 “STOP” 改变远端设备到停止状态

### 简述

如在控制输入 REQ 上得到一个上升沿，SFB 20 将把 ID 所表明的远端设备改变到 STOP 状态。当远端设备被 RUN、HOLD、STARTUP 时这种改变可以执行。

如远端设备为容错系统，再启动效果由参数 PI\_NAME 的设定来决定：启动请求对该系统的一个或所有 CPU 执行。

一旦暖或冷启动完成，设被改变为 RUN 模式而且发送一个肯定的执行应答。当肯定的执行应答被确认，状态参数 DONE 被设定为 1。如任何错误发生，将在 ERROR 和 STATUS 上指示出来。

只有之前被 SFB 20 停止的远端设备才可被再次启动。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
ID	INPUT	WORD	I, Q, M, D, 常数	寻址参数 ID, 参见 SFC 和 SFC S7 通讯常见参数章
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数: 0: 工作未开始或仍在运行 1: 工作被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数, 错误显示: ERROR=0 STATUS 有下列值: 0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
PI_NAME	IN_ OUT	ANY	I, Q, M, D	指向将被启动程序名称 (ASCII 码) 的区域指针。名称应不得超出 32 个字符, 对于 S7 设备, 为 P_PROGRAM。对于 H 系统, 可以为下列参数: <ul style="list-style-type: none"> <li>• P_PROGRAM (对所有系统中的 CPU 有效)</li> <li>• P_PROG_0 (对机架 0 的 CPU 有效)</li> <li>• P_PROG_1 (对机架 1 的 CPU 有效)</li> </ul>
IO_STATE	IN_ OUT	BYTE	I, Q, M, D, L	暂时未使用。 如你通讯的对象为 S7 可编程控制设备, 则此项不用设定。

## 故障信息

下表包含了使用 SFB/FB 20 时在 ERROR 和 STATUS 可能出现的故障信息。

ERROR	STATUS (十进制)	解释
0	11	警告：前面作业未完成，新作业不能生效。
0	25	数据已开始。作业进行中
1	1	通讯错误，例如： <ul style="list-style-type: none"> <li>• 连接简述未装载（本地或远端）</li> <li>• 连接中断（例如：电缆，电源失败，CP 处于 STOP 模式）</li> </ul>
1	2	收到对象设备否定应答，通讯功能不能执行
1	3	不能识别的 PI_NAME
1	4	指针 PI_NAME 涉及的数据长度或类型故障
1	7	远端设备不能完成启动
1	10	访问本地用户存储器不可能（例如，访问已删除的数据块）
1	12	当 SFB 调用时： <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB 20 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>
1	20	存储器不足。 H 系统：在更新过程中的 SFB 首次调用

## 19.13 用 SFB 21 “RESUME” 在远端设备上初始化一个热启动

### 简述

当控制输入 REQ 为上升沿，SFB 21 “RESUME” 激活在 ID 所指明的远端设备上的热启动。

如远端设备为 CPU，下列条件必须满足：

- CPU 必须处于 STOP 模式
- CPU 的钥匙开关应处于 RUN 或 RUN-P 位置
- 在你使用 STEP 7 建立组态时，设定手动启动模式
- 应没有任何阻止热启动的条件

当热启动完成后，设备将改变到运行模式并且发送肯定的执行应答。肯定应答被确认后，参数 DONE 被设定为 1，任何发生的错误在参数 ERROR 和 STATUS 中显示出来。

当前面的热启动完成后，一个重启动可在该远端设备上再次被激活。



参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
ID	INPUT	WORD	I, Q, M, D, 常数	寻址参数 ID, 参见 SFC 和 SFC S7 通讯常见参数章
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数: 0: 工作未开始或仍在运行 1: 工作被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数, 错误显示: ERROR=0 STATUS 有下列值: 0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
PI_NAME	IN_ OUT	ANY	I, Q, M, D	指向将被启动程序名称 (ASCII 码) 的区域指针。名称应不得超出 32 个字符, 对于 S7 设备, 为 P_PROGRAM。
ARG	IN_ OUT	ANY	I, Q, M, D, T, C	执行条款。 暂时未使用。 如你通讯的对象为 S7 可编程控制设备, 则此项不用设定。
IO_STATE	IN_ OUT	BYTE	I, Q, M, D, L,	暂时未使用。 如你通讯的对象为 S7 可编程控制设备, 则此项不用设定。

### 故障信息

下表包含了 ERROR 和 STATUS 参数可输出得 SFB 21 的所有可能出现的故障信息。

ERROR	STATUS (十进制)	解释
0	11	警告: 前面作业未完成, 新作业不能生效。
0	25	数据已开始。作业进行中
1	1	通讯错误, 例如: <ul style="list-style-type: none"> <li>• 连接简述未装载 (本地或远端)</li> <li>• 连接中断 (例如: 电缆, CPU 关闭, CP 处于 STOP 模式)</li> <li>• 与对方的连接没有建立</li> </ul>
1	2	收到对象设备否定应答, 通讯功能不能执行
1	3	不能识别的 PI_NAME
1	4	PI_NAME 或 ARG 涉及的数据长度或类型故障
1	7	远端设备不能完成启动
1	10	访问本地用户存储器不可能 (例如, 访问已删除的数据块)

ERROR	STATUS (十进制)	解释
1	12	当 SFB 调用时： <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB 19 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>
1	20	存储器不足。 <ul style="list-style-type: none"> <li>• H 系统：在更新过程中的 SFB 首次调用</li> <li>• S7-300 <ul style="list-style-type: none"> <li>- 最大并行工作数超出</li> <li>- 在 CPU 运行时块被重复装载</li> <li>- 可能是第一次调用</li> </ul> </li> </ul>

## 19.14 用 SFB 22 “STATUS” 块查询远端对象的状态

### 简述

使用 SFB 22 “STATUS”，你可以问询一个远端通讯对象的状态。

如在控制输入 REQ 上得到一个上升沿，一个工作被发送到远端对象。通过确定回应来确定是否其中有错误发生。如无错误发生，接收的状态被拷贝到 SFB 调用时的变量 PHYS、LOG 和 LOCAL 中。该工作的完成通过状态参数 NDR 得到 1 来表示。

你可以在同一个通讯对象上反复查询状态。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	请求控制参数。在上升沿时激活数据交换
ID	INPUT	WORD	I, M, D, 常数	寻址参数 ID, 参见 SFC 和 SFC S7 通讯常见参数章
NDR	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数： 0: 作业未开始或仍在运行 1: 作业被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数，错误显示： ERROR=0 STATUS 有下列值： 0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息

参数	声明	数据类型	存储区域	说明
PHYS	IN_ OUT	ANY	I, Q, M, D	物理状态（最小长度：1 字节） 可能值： <ul style="list-style-type: none"> <li>• 10H 功能</li> <li>• 13H 服务请求</li> </ul>
LOG	IN_ OUT	ANY	I, Q, M, D	本地状态（最小长度：1 字节） 可能值： <ul style="list-style-type: none"> <li>• 00H 状态改变被允许</li> </ul>
LOCAL	IN_ OUT	ANY	I, Q, M, D	如远端对象为 S7 的 CPU 的状态 （最小长度：2 字节）

### 输入/输出参数 LOCAL

如远端通讯对象为 S7 CPU，该输入/输出参数 LOCAL 包含它的当前状态。首个字节被保留，第二个字节以下列 ID 代表含义。

操作模式	相应标识
STOP	00H
暖启动	01H
RUN	02H
热启动	03H
HOLD	04H
冷启动	06H
RUN_R	09H
LINK_UP	0BH
UPDATE	0CH

### 故障信息

下表包含了使用 SFB/FB 22 时在 ERROR 和 STATUS 可能出现的故障信息。

ERROR	STATUS (十进制)	解释
0	11	警告：前面工作未完成，新工作不能生效。
0	25	数据已开始。工作进行中
1	1	通讯故障，例如： <ul style="list-style-type: none"> <li>• 连接简述未装载（本地或远端）</li> <li>• 连接中断（例如：电缆，CPU 关闭，CP 处于 STOP 模式）</li> <li>• 与对方的连接没有建立</li> </ul>
1	2	收到对象设备否定应答，通讯功能不能执行
1	4	PHYS, LOG 或 OCAL 涉及的数据长度或类型故障
1	8	远端目标访问被拒绝
1	10	访问本地用户存储器不可能（例如，访问已删除的数据块）

ERROR	STATUS (十进制)	解释
1	12	当 SFB 调用时： <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB 22 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>
1	20	存储器空间不足。 H 系统：SFB 调用更新中

## 19.15 用 SFB 23 “USTATUS” 接收远端设备的状态

### 简述

使用 SFB 23 “USTATUS”，接收一个远端通讯设备的状态。如在对方 STEP 7 中对其组态定义状态改变时其无条件发送状态。

在控制输入 EN\_R 上得到 1 时 SFB 被调用，得到来自远端的一个数据帧，状态信息进入变量 PHYS, LOG 和 LOCAL 中。该工作完成后在参数 NDR 上得到 1。

使用 USTATUS 时连接中的传送操作状态信息必须被使能。

注释：每个连接你只能给 SFB 23 定义一个背景数据块。

参数	声明	数据类型	存储区域	说明
EN_R	INPUT	BOOL	I, Q, M, D, L	被置位将接受远端通讯设备的状态
ID	INPUT	WORD	I, M, D, 常数	寻址参数 ID, 参见 SFC 和 SFC S7 通讯常见参数章
NDR	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数： 0: 作业未开始或仍在运行 1: 作业被无误执行
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L	ERROR 和 STATUS 状态参数，错误显示： ERROR=0 STATUS 有下列值：0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
PHYS	IN_OUT	ANY	I, Q, M, D	物理状态（最小长度：1 字节） 可能值： • 10H 功能 • 13H 服务请求
LOG	IN_OUT	ANY	I, Q, M, D	本地状态（最小长度：1 字节） • 可能值：00H 状态改变被允许

参数	声明	数据类型	存储区域	说明
LOCAL	IN_OUT	ANY	I, Q, M, D	如远端对象为 S7 的 CPU 的状态 (最小长度: 2 字节)

### 输入/输出参数 LOCAL

如远端通讯对象为 S7 的 CPU，该输入/输出参数 LOCAL 包含它的当前状态。首个字节被保留，第二个字节以下列 ID 代表含义。

操作模式	相应标识
STOP	00H
暖启动	01H
RUN	02H
热启动	03H
HOLD	04H
冷启动	06H
RUN (H 系统状态: 冗余)	09H
LINK_UP	0BH
UPDATE	0CH

### 故障信息

下表包含了使用 SFB/FB 22 时在 ERROR 和 STATUS 可能出现的故障信息。

ERROR	STATUS (十进制)	解释
0	9	覆盖警告: 一个旧的设备状态被新的状态所取代
0	25	数据已开始。工作处理中。
1	1	通讯故障, 例如: <ul style="list-style-type: none"> <li>• 连接简述未装载 (本地或远端)</li> <li>• 连接中断 (例如: 电缆, CPU 关闭, CP 处于 STOP 模式)</li> <li>• 与对方的连接没有建立</li> </ul>
1	4	PHYS, LOG 或 LOCAL 涉及的数据长度或类型故障
1	10	访问本地用户存储器不可能 (例如, 访问已删除的数据块)
1	12	当 SFB 调用时: <ul style="list-style-type: none"> <li>• 一个背景数据块与 SFB 23 不匹配</li> <li>• 非背景数据块而是全局数据块</li> <li>• 无背景数据块</li> </ul>
1	18	ID 所指示的连接中已存在一个背景数据块给 SFB 23 “USTATUS”
1	19	远端 CPU 发送数据块于用户程序中 SFB 接收
1	20	存储器空间不足 SFB 调用更新中

## 19.16 用 SFB 62 “CONTROL” 查询所属连接的状态

### 简述

使用 SFC 62 “CONTROL”，你可以查询属于 S7-400 的一个本地通讯 SFB 背景的连接的状态。

控制输入 ER\_R 得到 1 之后调用系统功能，属于通讯 SFB 背景的连接的状态由 I\_DB 来决定。

参数	声明	数据类型	存储区域	说明
EN_R	INPUT	BOOL	I, Q, M, D, L, 常数	被置位时将接受远端通讯设备的状态
I_DB	INPUT	WORD	I, Q, M, D, L, 常数	背景数据块的号码
OFFSET	INPUT	WORD	I, Q, M, D, L, 常数	在多重背景数据块下的数据记录的号码（如没有，则必须填 0）
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	错误信息
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数，错误显示： ERROR=0 STATUS 有下列值： 0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
I_TYP	OUTPUT	BYTE	I, Q, M, D, L	所属背景的块类型的标示符
I_STATE	OUTPUT	BYTE	I, Q, M, D, L	=0: 自上次暖/冷启动后无相应 SFB 背景数据块被装 <>0: 自上次暖/冷启动后至少已调用一次 SFB 背景数据块
I_CONN	OUTPUT	BOOL	I, Q, M, D, L	相应连接的状态 可能值： 0: 连接断开或未建立 1: 连接存在
I_STATU S	OUTPUT	WORD	I, Q, M, D, L	查询通讯 SFB 背景数据块的状态参数 STATUS

### 输出参数 I\_TYP

下表列出不同 SFB 类型和相应的标识。

SFB 类型	标示 (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
(无 SFB 存在; L_DB 或 OFFSET 错误)	FF

#### 故障信息

当使用 SFC 62 “CONTROL” 时输出参数 RET\_VAL 可有下例值

- 0000H: 在执行 SFC 时无错误发生
- 8000H: 在执行 SFC 时发生错误

ERROR	STATUS (十进制)	解释
1	10	不能访问本地用户存储器 (例如, 在实际参数 L_TYP 定义一个存储字节, 而定 CPU 正用的存储器中该存储字节不存在)
1	12	L_DB 所定义的号码, <ul style="list-style-type: none"> <li>• 无背景数据块, 而是全局数据块</li> <li>• 无数据块, 或背景数据块已毁坏</li> </ul>

## 19.17 用 FC 62 “C\_CNTRL” 查询连接状态

### 简述

使用 FC 62 “C\_CNTRL”，查询 S7-300 的连接状态。

当控制输入 EN\_R 得到 1 时 ID 所查询的通讯的当前状态可在系统功能调用之后确定。

参数	声明	数据类型	存储区域	简述
EN_R	INPUT	BOOL	I, Q, M, D, L, 常数	被置位时将接受远端通讯设备的状态
ID	INPUT	WORD	M, D, 常数	寻址参数 ID, 参见 SFC 和 SFC S7 通通常见参数章
RET_VAL	OUTPUT	INT	I, Q, M, D, L	错误信息
ERROR STATUS	OUTPUT OUTPUT	BOOL WORD	I, Q, M, D, L I, Q, M, D, L	ERROR 和 STATUS 状态参数, 错误显示: ERROR=0 STATUS 有下列值: 0000H: 无警告和错误 <>0000H: 警告。STATUS 提供详细信息 ERROR=1 有一个错误。STATUS 提供详细信息
C_CONN	OUTPUT	BOOL	I, Q, M, D, L	当前连接状态 可能值 <ul style="list-style-type: none"> <li>• 0: 连接中断</li> <li>• 1: 连接建立</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 连接未建立</li> <li>• W#16#0001 连接正建立</li> <li>• W#16#0001 连接已建立</li> <li>• W#16#000F 连接上无数据状态 (如 CP 的启动)</li> <li>• W#16#00FF 连接未组态</li> </ul>

### 故障信息

当使用 FC 62 “C\_CNTRL” 时输出参数 RET\_VAL 可有下列值:

0000H: FC 执行无错误; 8000H: FC 执行错误

#### 注释:

即使输出参数 RET\_VAL 指示为此 0000H, 输出参数 ERROR 和 STATUS 也应被确认。

ERROR	STATUS (十进制)	解释
1	10	CP 访问错误。另一工作进行中。迟后再试。



## 19.18 S7 通讯的 SFB/FB 工作存储器要求

为顺利工作，S7 SFB/FB 通讯要求在 CPU 的工作存储器里建立一个临时的存储器，存放用户数据（代码区）。所占用存储大小在下列表中：

在 S7-300 中的块		对于工作存储器的要求（单位：字节）
FB 8	USEND	块：4583 字节；背景：368 字节
FB 9	URCV	块：4880 字节；背景：370 字节
FB 12	BSEND	块：5284 字节；背景：372 字节
FB 13	BRCV	块：5258 字节；背景：374 字节
FB 14	GET	块：4888 字节；背景：336 字节
FB 15	PUT	块：4736 字节；背景：384 字节
FC 62	C_CNTRL	块：546 字节

关于中断的注释：

在 S7-300 中，SIMATIC\_NET 通讯块只能在一个优先级中调用。

在 S7-400 中的块		对于工作存储器的需求（单位：字节）
SFB 8/SFB 9	USEND/URCV	68+首次调用时从 SD_1, ...SD_4/RD_1, ...RD_4 指明的用户数据长度
SFB 12/SFB 13	BSEND/BRCV	54
SFB 14	GET	88+首次调用时从 RD_1, ...RD_4 指明的用户数据长度
SFB 15	PUT	108+首次调用时从 SD_1, ...SD_4 指明的用户数据长度
SFB 16	PRINT	78+首次调用时从 SD_1, ...SD_4/RD_1, ...RD_4 指明的用户数据长度+FORMAT 的长度
SFB 19	START	52+首次调用时从 PI_NAME 和 ARG 指明的用户数据长度
SFB 20	STOP	48+首次调用时从 PI_NAME 指明的用户数据长度
SFB 21	RESUME	52+首次调用时从 PI_NAME 和 ARG 指明的用户数据长度
SFB 22	STATUS	50
SFB 23	USTATUS	50





## 20 用于非组态 S7 连接的通讯 SFC

### 20.1 SFC 通讯的通用参数

#### 输入参数 REQ

输入参数 REQ（请求激活）是水平触发控制端参数。其被用来触发工作（数据传输或连接中断）。

- 如你调用一个当前未激活的 SFC 工作，通过 REQ=1 来触发工作。如首次 SFC 调用通讯时在通讯对象无连接，先建立连接之后再发送数据。
- 如你触发一个工作，在还未完成工作时你又调用一次，REQ 不能被 SFC 所确定。

#### 输入参数 REQ\_ID（仅在 SFC65 和 SFC66 中使用）

输入参数 REQ\_ID 用来确定你的发送数据。其被发送系统的 CPU 传递给通讯对象 CPU 的 SFC66 “X\_RCV” 端。

在下列情况下在接收端你也需要输入参数 REQ\_ID：

- 在发送 CPU 端有几个 SFC65 “X\_SEND”，借助不同的 REQ\_ID 来传送数据到通讯对象。
- 如你在不同的 CPU 使用 SFC65 “X\_SEND” 发送数据到通讯对象  
通过确定 REQ\_ID，你可保存不同的接收数据到存储区。

#### 输出参数 RET\_VAL 和 BUSY

通讯 SFC 被异步执行，这意味一个工作的执行可能跨越几个 SFC 调用。输出参数 RET\_VAL 和 BUSY 指明工作状态，参见异步 SFC 的参数含义 REQ，RET\_VAL 和 BUSY。

#### 输入参数 CONT

输入参数 CONT（继续）是一个控制参数。使用该参数，你决定属于一个通讯对象的连接在工作完成后是否还建立。

- 如你在第一次调用时选择（CONT=0，连接在数据传输完成后被终止。连接与一个新的通讯对象交换数据时又可获得。）  
这种方式可确保连接在实际被应用时资源被占据。
- 如你在第一次调用时选择 CONT=1，连接在数据传输完成后继续保持。这种方式，例如，当你在两个站之间周期性交换数据时有用。

---

#### 注释

CONT=1 时一个建立的连接可以明确的被 SFC69 “X\_ABORT” 或者 SFC74 “I\_ABORT” 终止。

---

## 20.2 用于非组态 S7 连接的通讯 SFC 的故障信息

### 故障信息

对于 SFC65 到 SFC74 真正的故障信息在“SFC65 到 SFC74 的详细信息”表中，其可被分类如下：

故障代码 (W#16# ...)	解释
809x	CPU 执行 SFC 时的故障
80Ax	参数通讯故障
80Bx	通讯对象错误
80Cx	临时故障

### SFC65 到 SFC74 的详细信息

故障代码 (W#16# ...)	解释 (通用的)	解释 (用于特殊的 SFC)
0000	完全执行无错误	SFC 69“X_ABORT”和 SFC 74“L_ABORT”： REQ=1，且该联结未建立。
		SFC 66 “X_RCV”：EN_DT=1 且 RD=NIL
00xy	-	SFC 66 “X_RCV”，NDA=1 且 RD<>NIL： RET_VAL 包含接收数据的长度 (EN_DT=0) 或拷贝到 RD 的数据长度 (EN_DT=1)
		SFC 67 “X_GET”：RET_VAL 包含接收数据的长度
		SFC 72 “L_GET”：RET_VAL 包含接收数据的长度
7000	-	SFC 65 “X_SEND,” SFC 67 “X_GET,” SFC 68 “X_PUT,” SFC 69 “X_ABORT,” SFC 72 “L_GET,” SFC 73 “L_PUT”, SFC 74 “L_ABORT”：REQ = 0 的调用 (调用未执行)，BUSY 的值为 1，无数据激活。
		SFC 66 “X_RCV”：EN_DT=0/1 和 NDA=0
7001	REQ=1 的首次调用：数据传输被触发；BUSY 值为 1。	-
7002	中间调用 (与 REQ 无关)：数据传输已激活，BUSY 值为 1。	SFC 69“X_ABORT”和 SFC 74“L_ABORT”： 使用 REQ 的中间调用。
8090	通讯对象的详细目的地地址无效， 例如： <ul style="list-style-type: none"> <li>• 错误的 IOID</li> <li>• 错误的起始地址</li> <li>• 错误的 MPI 地址 (&gt; 126)</li> </ul>	-

故障代码 (W#16# ...)	解释 (通用的)	解释 (用于特殊的 SFC)
8092	错误的 SD 或 RD, 例如: 局部数据区的地址不允许。	SFC 65 “X_SEND,” 例如 <ul style="list-style-type: none"> <li>非法的 SD 长度</li> <li>SD=NIL 为非法</li> </ul>
		SFC 66 “X_RCV,” 例如. <ul style="list-style-type: none"> <li>过多的数据被接受因而找不到 RD 规定的区域</li> <li>RD 规定的为 BOOL, 接收数据区然而超出一个字节。</li> </ul>
		SFC 67 “X_GET” 和 SFC 72 “I_GET,” 例如 <ul style="list-style-type: none"> <li>RD 长度非法</li> <li>RD 的数据长度与接收的数据不匹配</li> <li>RD=NIL 不允许</li> </ul>
		SFC 68 “X_PUT” 和 SFC 73 “I_PUT”, 例如 <ul style="list-style-type: none"> <li>SD 长度非法</li> <li>SD=NIL 非法</li> </ul>
8095	块已经在低优先级运行	-
80A0	应答错误	SFC 68 “X_PUT” 和 SFC 73 “I_PUT”: 发送 CPU 的 SD 规定的数据类型不被通讯对象支持
80A1	通讯问题: 断开一个现有的连接后调用 SFC	-
80B0	没有目标, 例如 DB 没有装载	可能因为 SFC 67“X_GET”, SFC 68“X_PUT” 和 SFC 72 “I_GET”, SFC 73 “I_PUT”
80B1	ANY 指针错误, 数据区的长度不匹配	-
80B2	硬件错误: 模板不存在 <ul style="list-style-type: none"> <li>组态的插槽被占用</li> <li>实际模板类型与预期不同</li> <li>分布式外设 I/Q 不存在</li> <li>无相应模板的 SDB 纪录</li> </ul>	可能因为 SFC 67“X_GET”, SFC 68“X_PUT” 和 SFC 72 “I_GET”, SFC 73 “I_PUT”
80B3	数据可能只读, 例如被保护的数据块	可能因为 SFC 67“X_GET”, SFC 68“X_PUT” 和 SFC 72 “I_GET”, SFC 73 “I_PUT”
80B4	ANY 指针数据类型错误, 或 ARRAY 规定的类型不允许	SFC 67 “X_GET, SFC 68 “X_PUT” 和 SFC 72 “I_GET”, SFC 73 “I_PUT”: VAR_ADDR 所规定数据类型不被通讯对象所支持
80B5	非法模式执行被拒绝	可能因为 SFC 65 “X_SEND”
80B6	接受的应答含有非法代码	-
80B7	传送的数据类型或长度不适合对方 CPU 被写区域。	可能因为 SFC 68“X_PUT”, SFC 73“I_PUT”

故障代码 (W#16# ...)	解释 (通用的)	解释 (用于特殊的 SFC)
80B8	-	SFC 65 “X_SEND”: 该 SFC 66 “X_RCV” 的通讯对象不允许接收 (RD=NIL)
80B9	-	SFC 65 “X_SEND”: 通讯对象所规定的数据块 (SFC66 “X_RCV”, EN_DT=0 调用), 因对方进入 STOP 模式而无法进入用户程序
80BA	通讯对象的反映不适合通讯帧	-
80C0	规定连接正被其他工作占用	-
80C1	SFC 执行时 CPU 缺乏资源, 例如: 模板的不同工作已达最大 连接资源正被使用, 例如: 接收数据	-
80C2	<ul style="list-style-type: none"> <li>• 通讯对象暂时缺乏资源, 例如通讯对象正在处理最大的工作</li> <li>• 所需的资源, 存储器等正被占用</li> <li>• 不足的工作存储器 (压缩的存储器)</li> </ul>	-
80C3	连接建立时错误, 例如: <ul style="list-style-type: none"> <li>• 本方 S7 站为接如 MPI 网</li> <li>• 所寻地址为自己的 MPI 地址</li> <li>• 通讯对象不再存在</li> <li>• 通讯对象临时缺乏资源</li> </ul>	-

## 20.3 使用 SFC65 “X\_SEND” 发送数据到外部通讯对象 S7 站

### 简述

使用 SFC65 “X\_SEND”，发送数据到一个外部 S7 站的通讯对象。

数据在通讯对象的 SFC “X\_RCV” 接收

REQ=1 时调用 SFC 数据被传输

确保（在发送 CPU 端）参数 SD 所定义的发送数据小于或等于（在接收 CPU 端）参数 RD 所定义的接收区。如 SD 为 BOOL 类型，RD 也应为 BOOL 类型。

参数	声明	数据类型	存储区域	简述
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“请求激活”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
CONT	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“继续”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
DEST_ID	INPUT	WORD	I, Q, M, D, L, 常数	地址参数“目标 ID”包含了用 STEP 7 组态的对方的 MPI 地址
REQ_ID	INPUT	DWORD	I, Q, M, D, L, 常数	工作标识符，用于标识通讯读象的工作
SD	INPUT	ANY	I, Q, M, D	发送数据参考，允许下面类型： BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME 及 ARRAY 除了 BOOL
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能被调用时如故障发生，返回值包含故障代码
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1:发送未完成 BUSY=0:发送完成或无法送功能被激活

### 数据连续性

数据以连续状态被发送

### 故障信息

见用于非组态 S7 连接的通讯 SFC 的故障信息



## 20.4 使用 SFC66 “X\_RCV” 接收来自外部通讯对象 S7 站的数据

### 简述

使用 SFC66 “X\_RCV”，你可以接收来自外部通讯对象 S7 站的 SFC65 “X\_SEND” 的数据

使用 SFC66 “X\_RCV”，

- 你可以检查是否数据被发送或处于等待。数据进入操作系统的队列中
- 你可以从一个选定的接收区里拷贝数据

参数	声明	数据类型	存储区域	简述
EN_DT	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“使能数据传输”。值为 0 时，你可用来检查是否有一个数据等待接收，值为 1 将拷贝最近的数据到 RD 定义的工作存储器区域
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如功能执行时出现错误，返回值包括错误代码。如无错误，RET_VAL 返回如下值： <ul style="list-style-type: none"> <li>• W#16#7000 如 EN_DT=0/1 且 NDA=0，这种情况下，无数据处于队列中</li> <li>• 如 EN_DT=0 且 NDA=1，最近的数据进入队列，并有一个正数被填充</li> <li>• EN_DT=1 且 NDA=1 数据被拷贝到接收区域，并有一个正数被填充</li> </ul>
REQ_ID	OUTPUT	DWORD	I, Q, M, D, L	SFC “X_SEND” 的标识符，那个数据在队列，换句话说最近的数据进入队列，如无数据在队列中，REQ_ID 值为 0
NDA	OUTPUT	BOOL	I, Q, M, D, L	状态参数“新数据到达” <p>NDA=0:</p> <ul style="list-style-type: none"> <li>• 无数据在队列中</li> </ul> <p>NDA=1:</p> <ul style="list-style-type: none"> <li>• 队列中至少包含一个数据 (EN_DT=0 时的 SFC 66 调用)</li> <li>• 队列中最近的数据背拷贝到用户程序 (EN_DT=1 时的 SFC 66 调用)</li> </ul>

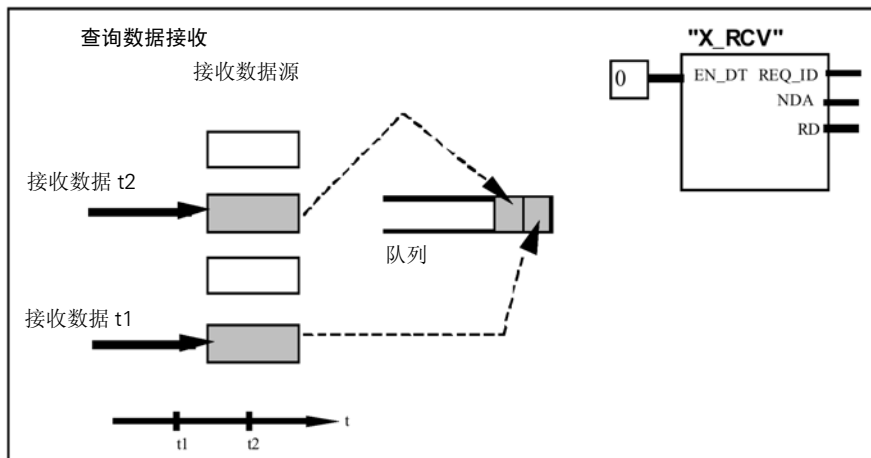
参数	声明	数据类型	存储区域	简述
RD	OUTPUT	ANY	I, Q, M, D	接收数据区参考。允许下面类型： BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT REAL, DATE, TOD, TIME, S5_TIME.DATE_AND_TIME 及 ARRAY 除了 BOOL 如你放弃在队列中最近的数据，给 RD 赋 NIL。 最大接收区域长度为 76 字节。

### EN\_DT=0 表明数据的接收

一旦来自通讯对象的数据到达，他们将按到来的顺序进入操作系统的队列。

如你想检查是否至少有一个块在队列中，EN\_DT=0 调用 SFC66 并且确定输出参数 NDA 如下：

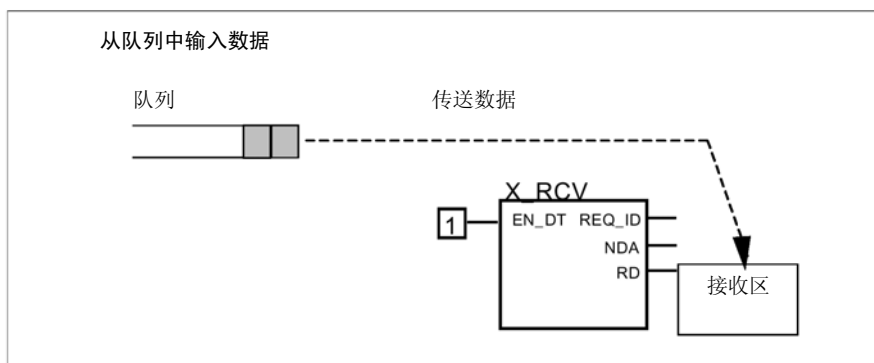
- NDA=0 表明队列中不含数据。与 REQ\_ID 无关，RET\_VAL 返回 W#16#7000。
- NDA=1 表明至少有一个数据在队列中可取。  
这种情况，你应该总是确定输出 RET\_VAL，而且如必要也包括 REQ\_ID。RET\_VAL 包含以字节为单位的数据长度，REQ\_ID 包含工作标识符。如有几个数据在队列中，REQ\_ID 和 RET\_VAL 属于队列中最原始的数据。



数据接收

### 当 EN\_DT=1 接收数据到接收区

当你设定 EN\_DT=1 调用 SFC66 “X\_RCV”，队列中的最原始数据被拷贝到 RD 所指定的工作存储器中。RD 必须大于或等于相应 SFC65 “X\_SEND” 中的 SD 参数发送区。如输入参数 SD 为 BOOL 数据类型，RD 必须也为 BOOL 类型。如你想接收位于不同区域的数据，你可以查询 REQ\_ID (EN\_DT=0 时调用 SFC) 并在接下来的调用选择一个适合的 RD (此时 EN\_RT=1)。如数据在拷贝时无错误发生，RET\_VAL 以字节为单位返回拷贝的数据长度并给发送端一个肯定的应答。



数据接收

### 放弃数据

如你不想取队列中的数据，给 RD 赋值 NIL（见/232/）。这种情况，发送端接收一个否定的应答（相应的 SFC65 “X\_SEND” RET\_VAL 得到值 W#16#80B8）。SFC66 “X\_RCV” 的 RET\_VAL 得到值 0。

### 数据连续性

在 EN\_DT=1 调用后，RET\_VAL=W#16#00XY，接受区域 RD 包含新数据。这些数据可以被后续的块调用所覆盖。为防止这种情况，直到你确定已完成接收以后，才能在同一接收区域 RD 调用 SFC66 “X\_RCV”。

### 改变到 STOP 模式

如果 CPU 变换到 STOP 模式

- 所有新到来的工作被否定应答
  - 所有已到达和进入队列的工作被否定应答
    - 如 STOP 之后执行一个暖/冷启动，块的数据全部被放弃。
    - 如改变到 STOP 之前队列被闻讯，之后一个重启动（在 S7-300 和 S7-400H 不可行）最原始的工作进入用户程序（通过 EN\_DT=0 调用 SFC66 “X\_RCV”）。否则数据被放弃。
- 所有其它的数据被放弃。

### 连接中断

如连接中断，属于该连接已进入队列的工作将被放弃。

例外：如在队列中最原始的工作，你已通过设定 EN\_DT=0 调用 SFC66 “X\_RCV” 检测到，可通过 EN\_DT=1 取该数据。

### 故障信息

见用于非组态 S7 连接的通讯 SFC 的故障信息。

## 20.5 使用 SFC68 “X\_PUT” 写数据到外部通讯对象 S7

### 简述

使用 SFC68 “X\_PUT”，你写数据到一个 S7 站通讯对象。在通讯对方不用相应的 SFC 块。当设定 REQ=1 调用该 SFC 时写工作被激活。此后，你可继续调用直到接收到 BUST=1 的应答。

确保（在发送 CPU 端）参数 SD 所定义的发送数据小于或等于参数 VAR\_ADDR 所定义接收区域。SD 和 VAR\_ADDR 的数据类型也必须匹配。

参数	声明	数据类型	存储区域	简述
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“请求激活”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
CONT	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“继续”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
DEST_ID	INPUT	WORD	I, Q, M, D, L, 常数	地址参数“目标 ID”，包含了对方的 MPI 地址
VAR_ADDR	INPUT	ANY	I, Q, M, D	发送到对方 CPU 的参考区域。必须选择通讯对象支持的数据类型
SD	INPUT	ANY	I, Q, M, D	发送数据参考，允许下面类型：BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME.S5_TIME, DATE_AND_TIME 及 ARRAY 除了 BOOL SD 必须和 VAR_ADDR 相同长度，数据类型也必须匹配。最大发送数据长度为 84 字节
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能被调用时如错误发生，返回值包含错误代码
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: 发送未完成 BUSY=0: 发送完成或无法送功能被激活

### 改变到 STOP 模式

如 CPU 改变到 STOP 模式，通过 SFC68 “X\_PUT” 建立的连接被终止。数据不再被发送。如 CPU 模式改变时发送数据已被拷贝到内部缓存区，缓存区的内容被放弃。

通讯对象改变到 STOP 模式

如通讯对方的 CPU 改变到 STOP 模式，并不影响 SFC68 “X\_PUT” 的数据传送。在 STOP 模式时数据也可以被写到对方。

### 数据连续性

数据以连续的状态被发送。

### 故障信息

见用于非组态 S7 连接的通讯 SFC 的故障信息。

## 20.6 使用 SFC67 “X\_GET” 读来自外部通讯对象 S7 站的数据

### 简述

使用 SFC67 “X\_GET”，你可以读取来自外部通讯对象 S7 站的数据，这种通讯方式你不需要在通讯对方编写相应程序块。

当 REQ=1 时调用 SFC 将激活工作。之后，到 BUSY 为 0 时还可以调用这个块。RET\_VAL 以字节数量返回接收数据的长度。

确保参数 RD（在接收 CPU 方）所定义接收区长度与参数 VAR\_ADDR（在发送区）相同，他们的数据类型也必须匹配。

参数	声明	数据类型	存储区域	简述
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“请求激活”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
CONT	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“继续”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
DEST_ID	INPUT	WORD	I, Q, M, D, L, 常数	地址参数“目标 ID”，包含了用 STEP 7 组态的对方的 MPI 地址
VAR_ADDR	INPUT	ANY	I, Q, M, D	发送到对方 CPU 的参考区域。必须选择通讯对象支持的数据类型
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能被调用时如错误发生，返回值包含错误代码如没有错误发生，RET_VAL 将返回所接受的数据长度（单位字节）
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: 发送未完成 BUSY=0: 发送完成或无法送功能被激活
RD	OUTPUT	ANY	I, Q, M, D	接收数据区参考。允许下面类型： BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME.S5_TIME.DATE_AND_TIME 以及除 BOOL 以外的 ARRAY RD 接收区域长度与通讯对象的 VAR_ADDR 的规定一致，数据类型也必须匹配。 最大接收区域长度为 96 字节。

### 改变到 STOP 模式

如 CPU 改变到 STOP 模式，SFC67 “X\_GET” 所建立的连接被终止。重启类型决定进入到操作系统缓冲区的届欧数据是否丢失：

- 热启动（不包括 S7-300 或 S7-400H），数据被拷贝到 RD 所定义的区域。
- 暖/冷启动，数据被放弃。

**通讯对方改变到 STOP 模式**

如通讯对方改变到 STOP 模式，这并不影响 SFC67“X\_GET”传送数据。仍然读取对方 STOP 模式的数据。

**数据连续性**

数据以连续的状态被读取

**故障信息**

见用于非组态 S7 连接的通讯 SFC 的故障信息。

## 20.7 使用 SFC69 “X\_ABORT” 中断一个连到外部 S7 站的已有连接

**简述**

使用 SFC69 “X\_ABORT”，你可以中断一个由 SFC X\_SEND，X\_GET 或 X\_PUT 建立的已存在连接。如果属于 X\_SEND，X\_GET 或 X\_PUT 的工作已完成（BUSY=0），连接资源会在调用 SFC69 “X\_ABORT” 后在双方被释放。如果属于 X\_SEND，X\_GET 或 X\_PUT 的工作仍未完成（BUSY=1），在连接中断后再次使用 REQ=0 和 CONT=0 调用 SFC，等待 BUSY=0，之后所用的连接才被释放。你可以调用 SFC69 “X\_ABORT” 在使用 SFCs “—X\_SEND”，“X\_PUT”或“X\_GET”的端点。中断的连接可在之后调用 SFC 时设定 REQ=1 再激活。

参数	声明	数据类型	存储区域	简述
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“请求激活”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
DEST_ID	INPUT	WORD	I, Q, M, D, L, 常数	地址参数“目标 ID”，包含了对方的 MPI 地址
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能被调用时如错误发生，返回值包含错误代码
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: 发送未完成 BUSY=0: 发送完成或无法送功能被激活

**改变到 STOP 模式**

如 CPU 改变到 STOP 模式，连接中断再开始调用 SFC69 “X\_ABORT” 后完成。

**通讯对方改变到 STOP 模式**

如通讯对方改变到 STOP 模式，这并不影响 SFC69 “X\_ABORT” 中断连接。

**故障信息**

见用于非组态 S7 连接的通讯 SFC 的故障信息。

## 20.8 使用 SFC73 “I\_PUT” 写数据到内部通讯对象 S7 站

### 简述

使用 SFC73 “I\_PUT” 块，你可以写数据到内部通讯对象 S7 站中的通讯对象，其可在中央机架，扩展机架或者分布式机架。一定确保你在 STEP7 所赋的对象属于本方 CPU。这种方式不必在对方编程。

发送工作在控制参数 REQ=1 时调用 SFC 时激活。

确保（在发送 CPU 端）参数 SD 所定义的发送数据等于参数 VAR\_ADDR 所定义的接收区（在通讯对象方），数据类型也应匹配。

参数	声明	数据类型	存储区域	简述
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“请求激活”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
CONT	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“继续”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
IOID	INPUT	BYTE	I, Q, M, D, L, 常数	对方模板的地址标示符： B#16#54=外设输入（PI） B#16#55=外设输出（PQ） 如一个混合模板地址标示符取两个中较小的地址；如地址相同，写 B#16#54。
LADDR	INPUT	WORD	I, Q, M, D, L, 常数	对方模板的逻辑地址，如一个混合模板地址取两个中较小的。
VAR_ADDR	INPUT	ANY	I, Q, M, D, L	发送到对方 CPU 的参考区域。必须选择通讯对象支持的数据类型
SD	INPUT	ANY	I, Q, M, D	发送数据参考，允许下面类型： BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME 及 ARRAY 除了 BOOL。 SD 必须和 VAR_ADDR 相同长度，数据类型也必须匹配。 最大发送数据长度为 94 字节
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能被调用时如错误发生，返回值包含错误代码
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: 发送未完成 BUSY=0: 发送完成或无法送功能被激活

### 改变到 STOP 模式

如 CPU 改变至 STOP 模式，通过 SFC73 “I\_PUT” 建立的连接被终止。数据不再被发送。

如 CPU 模式改变时发送数据已被拷贝到内部缓存区，缓存区的内容被放弃。

**通讯对象改变到 STOP 模式**

如通讯对方的 CPU 改变到 STOP 模式，并不影响 SFC73 “I\_PUT”的数据传送。在 STOP 模式时数据也可以被写到对方。

**数据连续性**

数据以连续的状态被发送

**故障信息**

见用于非组态 S7 连接的通讯 SFC 的故障信息。

## 20.9 使用 SFC72 “I\_GET” 读来自内部通讯对象 S7 站的数据

**简述**

使用 SFC72 “I\_GET”，你可以读来自内部通讯对象 S7 站的数据。通讯对象可在中央机架，扩展机架或者分布式机架。一定确保你在 STEP7 所赋的对象属于本方 CPU。这种方式不必在对方编程。

接收工作在 REQ=1 调用 SFC 时激活。之后你可继续接收数据直到 BUSY=0.RET\_VAL 以字节数量返回接收数据的长度。

确保参数 RD（在接收 CPU 方）所定义接收区长度与参数 VAR\_ADDR（在通讯对象）相同，他们的数据类型也必须匹配。

参数	声明	数据类型	存储区域	简述
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“请求激活”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
CONT	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“继续”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
IOID	INPUT	BYTE	I, Q, M, D, L, 常数	对方模板的地址标示符： B#16#54=外设输入（PI） B#16#55=外设输出（PQ） 如一个混合模板地址标示符取两个中较小的地址；如地址相同，写 B#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, 常数	对方模板的逻辑地址，如一个混合模板地址取两个中较小的。
VAR_ADDR	INPUT	ANY	I, Q, M, D	发送到对方 CPU 的参考区域。必须选择通讯对象支持的数据类型。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能被调用时如错误发生，返回值包含错误代码。如无错误，返回接收数据长度值（单位字节）。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1：发送未完成 BUSY=0：发送完成或无法送功能被激活



参数	声明	数据类型	存储区域	简述
RD	OUTPUT	ANY	I, Q, M, D	接收数据区参考。允许下面类型： BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT REAL, DATE, TOD, TIME.S5_TIME. DATE_AND_TIME 及 ARRAY 除了 BOOL。 RD 接收区域长度与通讯对象的 VAR_ADDR 的规定一致，数据类型也必须匹配。 最大接收区域长度为 94 字节。

#### 改变到 STOP 模式

如 CPU 改变到 STOP 模式，通过 SFC72 “I\_PUT” 建立的连接被终止。数据不再被发送。如 CPU 模式改变时发送数据已被拷贝到内部缓存区，缓存区的内容是否丢失取决于重新启动的方式：

- 热启动（不包括 S7-300 或 S7-400H），数据被拷贝到 RD 所定义的区域。
- 暖/冷启动，数据被放弃。

#### 通讯对象改变到 STOP 模式

如通讯对方的 CPU 改变到 STOP 模式，并不影响 SFC72 “I\_GET” 的数据传送。在 STOP 模式时数据也可以被读到对方。

#### 数据连续性

数据以连续的状态被发送。

#### 故障信息

见用于非组态 S7 连接的通讯 SFC 的故障信息。

## 20.10 使用 SFC74 “I\_ABORT” 中断一个已存在到本方 S7 站的连接

### 简述

使用 SFC74 “I\_ABORT”，你可以终止一个由 SFC72 “I\_GET” 或 SFC73 “I\_PUT” 建立到通讯对象的连接。如属于 SFC72“I\_GET”或 SFC73“I\_PUT”的工作已经完成（BUSY=0），连接资源在调用 SFC74 “I\_ABORT” 后在双方都被释放。

如属于 SFC72 “I\_GET” 或 SFC73 “I\_PUT” 的工作还未完成（BUSY=1），连接被中断后的相关 SFC 调用（REQ=0 且 CONT=0）等待 BUSY=0，这之后连接资源被释放。

你在调用 SFC “I\_PUT” 或 SFC “I\_GET” 的设备中调用 SFC74 “I\_ABORT”（或者换句话说在客户端）。

当以 REQ=1 时调用 SFC 将激活连接中断功能。

参数	声明	数据类型	存储区域	简述
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数“请求激活”，参见基于 S7 基础通讯的 SFC 通讯的通用参数
IOID	INPUT	BYTE	I, Q, M, D, L, 常数	对方模板的地址标示符： B#16#54=外设输入（PI） B#16#55=外设输出（PQ） 如一个混合模板地址标示符取两个中较小的地址；如地址相同，写 B#16#54。
LADDR	INPUT	WORD	I, Q, M, D, L, 常数	对方模板的逻辑地址，如一个混合模板地址取两个中较小的。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	功能被调用时如错误发生，返回值包含错误代码。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1：发送未完成 BUSY=0：发送完成或无法送功能被激活

### 改变到 STOP 模式

如 CPU 改变到 STOP 模式，通过 SFC74 “I\_ABORT” 开始的连接中断被完成。

### 通讯对方改变到 STOP 模式

如通讯对方的 CPU 改变到 STOP 模式，这并不影响 SFC74 “I\_ABORT” 的连接中断的执行。连接将被终止。

### 故障信息

见用于非组态 S7 连接的通讯 SFC 的故障信息。



## 21 生成块相关的信息

### 21.1 用 SFB 生成块相关信息简介

#### SFB 用于生成块相关的信息

可以通过在程序中调用下列其中一个 SFB 生成块相关的信息：

- SFB 36 “NOTIFY”
- SFB 31 “NOTIFY\_8P”
- SFB 33 “ALARM”
- SFB 34 “ALARM\_8”
- SFB 35 “ALARM\_8P”

这些 SFB 具有下列属性：

- 用 SFB 36 “NOTIFY” 与 31 “NOTIFY\_8P” 保存块调用时所有检测到的 0 -> 1 或 1 -> 0 的信号变化都将引发信息的传送。
- 如果设置为缺省报告方式（确认触发报告禁止），当块被调用时任何检测到的信号变化也将在 SFB 33 “ALARM”，34 “ALARM\_8” 与 35 “ALARM\_8P” 中生成一条故障信息。  
另外，如果确认触发报告已经使能，并不是所有的信号变化都会触发信息的传送（详细说明见下面）。
- 随着块的执行，相关的数值（输入 SD\_i）被全部读出并赋值给信息（参见 SFB/FB 与用于 S7 通讯的 SFC/FC 的公共参数中“发送与接收参数”）  
由于与高优先级相比较的一致性，下列相关数据是一致的：每个相关数据 SD\_I 都是一致的。
- 通过状态参数 DONE, ERROR 与 STATUS，可以监视块的运行状态（参见 SFB/FB 与用于 S7 通讯的 SFC/FC 的公共参数中“状态参数”）

---

#### 注意

只有在块开始调用时分析参数 ID 与 EV\_ID 才有意义（实际参数或调用时定义的数值）。

---

#### 登录显示设备

在检测到信号变化用于生成块相关的信息的 SFB 发出信号之前，至少要登录一个显示设备（如，WinCC）。如没有登录参数 STATUS = 1

#### 信号变化检测

信息块的每一次调用都配有一个带有两个存储块的信息存储器。

这个信息存储器开始时清零。在输入 SIG 或者输入 SIG\_1, ...SIG\_8 中的一个中发生的信号变化进入第一个存储块。这个存储块一直被占用着直到相应的信息被送出去。

下一个在输入 SIG 或输入 SIG\_1, ...SIG\_8 中的一个中检测到的信号变化则进入第二个存储块。如果第一个存储块一直被占用，第二个存储块将总是被覆盖。

输出参数 ERROR 与 STATUS 会指示信息丢失。这个信息连同下一个传送的信息也送到在线的显示设备。

第一个存储块清零后，第二个存储块的数据移到第一个块。这样，第二个存储块也被清零。

### 确认触发报告

可以通过操作 SFB33 “ALARM”，34 “ALARM\_8” 采用确认触发报告来减少系统的信息阻塞。

也就是说，在一个进入的信号信息被生成（信号变化 0 > 1）之后，直到在显示系统上确认了第一条信息，后面的信息才能被生成。下一条信息在显示设备上显示直到确认此信号消失（信号变化 1 至 0）。下一个信息循环开始，又进入一条信息（信号变化 0 至 1），亦必须被确认，通过这种方式可以通过显示设为控制信号变化信息（存储消掉的信息）。

设置报告模式（使能或禁止确认触发报告）在 STEP 7 CPU 组态中为 SFB 33~35。确认触发报告缺省为禁止。

为保证系统中信息分析的一致性，应该确认所有的显示设为都能处理确认信息报告。

---

#### 注意不能处理确认信息报告的操作显示设备。

一个确认触发报告使能的 CPU 只把信息发布到能够处理这个报告的显示设备上。如果没有能够处理确认触发报告的显示设备，CPU 不发送任何信息。这种状况由 ERROR=1 与 STATUS=1 指示一次。

---

### 用 SFB33 “ALARM”、34 “ALARM\_8” 与 35 “ALARM\_8P” 确认信息

使用集中确认的概念。在一台显示设备上对信息进行确认后，这个确认信息首先送到产生此信息的 CPU。从这儿，确认信息送到所有为此目的登录的站。

确认的是一个信号而不是一个单独的信息。例如，如果显示一个信号的多个上升沿，确认了事件的进入状态，前面所有具有同一个信息号的事件都被看作是已经确认。

### 确认显示

SFB 36 “NOTIFY” 与 SFB 31 “NOTIFY\_8P” 不具有确认指示。可以检查 SFB 33 “ALARM” 中的输出参数 ACK\_UP 和 SFB 35 “ALARM\_8P” 与 34 “ALARM\_8” 中的输出参数 ACK\_STATE。这些输出在调用块时被刷新，前提是控制参数 EN\_R 为 1。

### 通过 SFC 或显示设备（WinCC）禁止和使能信息

一些情况下，禁止信息是有必要的（比如，系统重新组态时）。为此可以在显示设备或程序中禁止或使能信息。禁止/使能适用于所有登录使用此信息的站。一个禁止的信息保持在禁止状态直到重新使能。

禁止信息可以在输出参数 ERROR 与 STATUS 中看出（ERROR = 1，STATUS = 21）。

### SFB 生成块相关的信息需要的存储资源

为正常运行，生成块相关信息的 SFB 需要使用 CPU 工作存储器中的中间存储空间，大小取决于相关数据的长度。参见下表，使用的存储器空间的信息。

块类型	需要的 CPU 工作存储器空间 (字节)
NOTIFY	2 x (190 + 首次调用时在 SD_1, ...SD_10 中指定的相关数据的长度)
NOTIFY_8P	2 x (190 + 首次调用时在 SD_1, ...SD_10 中指定的相关数据的长度)
ALARM	2 x (190 + 首次调用时在 SD_1, ...SD_10 中指定的相关数据的长度)
ALARM_8	180
ALARM_8P	2 x (190 + 首次调用时在 SD_1, ...SD_10 中指定的相关数据的长度)
AR_SEND	108

### 可传送数据的数目

通过 NOTIFY, NOTIFY\_8P, ALARM 与 ALARM\_8P SFB 的相关参数 SD\_i 传送的数据一定不能超出一个最大长度。最大数据长度由下式计算:

$$\text{maxleng} = \min(\text{pdu\_lokal}, \text{pdu\_remote}) - \text{diff} - 4 * \text{所用 SD\_i 参数的数目}$$

说明:

- $\min(\text{pdu\_lokal}, \text{pdu\_remote})$  指数据区域 pdu\_lokal 与 pdu\_remote 的最小值。
- pdu\_lokal 指本地 CPU 数据区域的最大长度 (参见 CPU 的技术数据)
- pdu\_remote 指显示设备数据区域的最大长度。
- 如果确认触发报告使能 diff = 48, 如果禁止使能 diff = 44。

例如:

CPU 414-2 通过以太网向 WinCC 发送信息。确认触发报告禁止。

相关参数 SD\_1, SD\_2 与 SD\_3 使用

pdu\_lokal = 480 字节, pdu\_remote = 480 字节

使用的 SD\_i 参数数目: 3。

计算结果:

$$\text{maxleng} = \min(480, 480) - 44 - 4 \times 3 = 480 - 44 - 12 = 424$$

每个 SFB 能传送的数据的最大长度为 424 字节

## 21.2 用 SFB 36 “NOTIFY” 生成无需确认的块相关的信息

### 描述

SFB 36 “NOTIFY” 监视信号。上升沿（事件进入）和下降沿（事件离开）都会生成信息。可以通过这条信息最多发送 10 个相关数据。信息发送到为此而登录的站。首次调用 SFB 时，当前信号状态信息被发送。

相关的数据在检测到沿时产生并赋给信息。

SFB 36 “NOTIFY” 的每次调用都可以使用一个带有两个存储块的存储器。关于保存信号变化到中间存储器可参阅用 SFB 生成块相关的信息简介中“信号变化检测”一节。

SFB 36 “NOTIFY” 符合标准 IEC 1131-5。

参数	声明	数据类型	存储区域	说明
SIG	INPUT	BOOL	I, Q, M, D, L	监视的信号
ID	INPUT	WORD	I, Q, M, D, L, 常数	信息的数据通道 W#16#EEEE ID 只在首次调用时有意义。
EV_ID	INPUT	DWORD	I, Q, M, D, L, 常数	信息号（不允许为 0）只在首次调用时有意义。以后每次 SFB 36 调用时，使用首次调用的信息号与相应的背景数据块 STEP 7 自动地分配信息号以保证信息号一致。
SEVERITY	INPUT	WORD	I, Q, M, D, L, 常数	事件的重要性：可能值： 0~127（0 表示最重要） 此参数与信息处理无关。
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数。信息生成完毕。
ERROR	OUTPUT	BOOL	I, Q, M, D, L	状态参数。 ERROR=TRUE 表示运行中有故障产生。详细信息参考参数 STATUS。
STATUS	OUTPUT	WORD	I, Q, M, D, L	STATUS 参数：显示故障信息。
SD_I, 1≤i≤10	IN_OUT	ANY	I, Q, M, D, T, C	第 i 个相关数据。 只允许使用下列数据类型：BOOL（不允许用：位域），BYTE，CHAR，WORD，INT，DWORD，DINT，REAL，DATE，TOD，TIME，S5TIME，DATE_AND_TIME <b>注意：</b> 如果 ANY 指针访问一个 DB，这个 DB 必须一直被指定（例如：P# DB10.DBX5.0 字节 10）。

## 故障信息

下表包含了所有定义给 SFB 36 可以通过参数 ERROR 与 STATUS 输出的故障信息。

ERROR	(十进制) STATUS	说明
0	11	信息丢失，信号变化/信息不能发送。
0	22	<ul style="list-style-type: none"> <li>• 相关数据 SD_I 指针错误：               <ul style="list-style-type: none"> <li>- 数据长度或数据类型有误</li> <li>- 相关数据在用户存储器中找不到，例如：由于 DB 删除或区域长度故障</li> </ul> </li> <li>生成的信息发出，但相关的数据可能没发出，也可能发出。</li> <li>• 为 SEVERITY 选定的实参数超出允许范围。生成的信息发出设置 SEVERITY=127。</li> </ul>
0	25	通讯开始。正在处理信息。
1	1	通讯故障：连接失败或没有登录
1	4	首次调用时： <ul style="list-style-type: none"> <li>• 指定的 EV_ID 超出范围或者</li> <li>• ANY 指针 SD_I 有一个格式错误</li> <li>• 每个 SFB CPU 可以发送的最大存储区域超出。</li> </ul>
1	10	不能访问本地用户存储器（例如，访问一个删除的 DB）
1	12	SFB 调用时： <ul style="list-style-type: none"> <li>• 指定了一个不属于 SFB 36 的背景 DB</li> <li>• 指定了一个共享 DB 而不是背景 DB</li> </ul>
1	18	EV_ID 已经被 SFB 31 或 33~36 中的一个使用了。
1	20	工作存储器不够用。 H-系统：刷新时调用了 SFB
1	21	指定 EV_ID 的信息被禁止。



## 21.3 用 SFB 31 “NOTIFY\_8P” 生成无需确认显示的块相关的信息

### 描述

SFB 31 “NOTIFY\_8P” 扩展了 SFB 36 “NOTIFY” 的功能，它可以生成 8 个信号的信息。至少检测到一个信号变换信息才能生成。总是在开始调用 SFB31 时生成信息。8 个信号具有一个公共的信息号，可以在显示设备上分为 8 个子信息。

每次调用 SFB 31 “NOTIFY\_8P” 都可以使用一个带有两个存储块的存储器。关于把信号变化存入中间存储器参见用 SFB 生成块相关的信息简介中“信号变化检测”一节。

### 注意

显示设备列出最近的两个信号变化，这不表示信号丢失。



### 警告

在 CPU 中调用 SFB 31 “NOTIFY\_8P” 之前，必须保证所有连接的显示设备都能识别此块。也就是说，至少应是下列版本之一：STEP 7, V5.1 SP3, WinCC V5.1 Hot Fix 1, PCS7 V5.2 Service Pack 2, SIMATIC Device Driver V5.6。

否则，PLC 与连接的显示设备之间的通讯会关断。这时就不能再通过连接的显示设备访问系统了。

参数	声明	数据类型	存储区域	说明
SIG_i, 1 ≤ i ≤ 8	INPUT	BOOL	I, Q, M, D, L	第 i 个被监视的信号。
ID	INPUT	WORD	常数 (I, Q, M, D, L)	信息的数据通道: W#16#EEEE ID 只在开始调用时有意义。
EV_ID	INPUT	DWORD	常数 (I, Q, M, D, L)	信息号 (不允许为: 0) EV_ID 只在首次调用时有意义。以后每次 SFB 31 与相应的背景数据块调用时都使用第一次调用的信息号。 STEP 7 自动地分配信息号以保证信息号一致。
SEVERITY	INPUT	WORD	常数 (I, Q, M, D, L)	事件的重要性: 0~127 (0 为最重要); 缺省值: 64 这个参数与信息处理无关。
DONE	OUTPUT	BOOL	I, Q, M, D, L	状态参数 DONE: 信息生成完毕。
ERROR	OUTPUT	BOOL	I, Q, M, D, L	状态参数 ERROR ERROR=TRUE 表示运行中有故障产生。详细信息参考参数 STATUS。
STATUS	OUTPUT	WORD	I, Q, M, D, L	状态参数 STATUS: 显示故障信息

参数	声明	数据类型	存储区域	说明
SD_i, 1 ≤ i ≤ 10	IN_OUT	ANY	I, Q, M, D, T, Z	<p>第 i 个相关数据</p> <p>只允许使用下列数据类型 BOOL（不允许用：位域），BYTE，CHAR，WORD，INT，DWORD，DINT，REAL，DATE，TOD，TIME，S5TIME，DATE_AND_TIME。</p> <p><b>注意：</b></p> <p>如果 ANY 指针访问一个 DB，这个 DB 必须一直被指定。</p> <p>（例如：P# DB10.DBX5.0 字节 10）。</p>

## 故障信息

下表包含了所有定义给 SFB 31 可以通过参数 ERROR 与 STATUS 输出的故障信息。

ERROR	STATUS（十进制）	说明
0	11	信息丢失
0	11	信息丢失，至少有一个信号变化/信息不能发送。
0	22	<ul style="list-style-type: none"> <li>相关数据 SD_i 指针错误： <ul style="list-style-type: none"> <li>与数据长度/类型有关</li> <li>相关数据在用户存储器中找不到，例如：由于 DB 删除或区域长度故障</li> </ul> </li> </ul> <p>生成的信息发出没有与或者，要求，与最大可能数目的相关数据一起发送。</p> <ul style="list-style-type: none"> <li>选定的 SEVERITY 的实际参数超出上限。生成的信息将发出，其中 SEVERITY=127。</li> </ul>
0	25	通讯开始。正在处理信息。
1	1	通讯故障：通讯关闭或没有登录
1	4	<p>首次调用时：</p> <ul style="list-style-type: none"> <li>指定的 EV_ID 超出范围</li> <li>ANY 指针 SD_i 格式错误</li> <li>每个 SFB 31 CPU 可以发送的最大存储区域超出。</li> </ul>
1	10	不能访问本地用户存储器（例如，试图访问一个删掉的 DB）
1	12	<p>SFB 调用时</p> <ul style="list-style-type: none"> <li>指定了一个不属于 SFB 31 的背景 DB</li> <li>指定了一个全局 DB 而不是背景 DB</li> </ul>
1	18	EV_ID 已经被 SFB 31 或 33~36 使用。
1	20	<p>工作存储器不够用。</p> <p>H-系统：刷新时调用 SFB</p>
1	21	指定 EV_ID 的信息被禁止

## 21.4 用 SFB 33 “ALARM” 生成需确认的块相关的信息

### 描述

SFB 33 “ALARM” 监视一个信号。

缺省模式（即确认触发报告禁止）：此块在正沿（事件进入）和负沿（事件离开）都生成信息。可以通过此信息发送多发 10 个相关数据。

确认触发报告使能：在生成了一个信号的信息之后，这个块将不再产生信息直到在显示设备上确认这条信息。（亦可参见用 SFB 生成块相关的信息简介）

这条信息被发送到所有为此中登录的站。

首次调用此 SFB 时，当前信号状态的信息被发送。

ACK\_UP 输出在上升沿复位。当来自登录显示设备的确认事件进入状态信号到时它被置位。

ACK\_DN 输出的情况与上面类似：它在下降沿复位。当来自登录显示设备的确认事件离去状态信号到来时被置位。一旦收到来自登录显示设备的确认信号，确认信号就传送到所有为此目的而登录的站。

每次调用 SFB33 “ALARM” 都可使用一个带有两个存储块的信息存储器。关于保存信号变化到中间存储器可参阅用 SFB 生成块相关的信息简介中“信号变化检测”一节。

SFB 33 “ALARM” 符合 IEC 1131-5 标准。

参数	声明	数据类型	存储区域	说明
EN_R	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数，使能接收在块首次调用时 ACK_UP 与 ACK_DN 是被刷新（EN_R=1）还是不被刷新（EN_R=0）的决定。如果 EN_R=0，输出参数 ACK_UP 与 ACK_DN 保持不变。
SIG	INPUT	BOOL	I, Q, M, D, L	监视的信号
ID	INPUT	WORD	I, Q, M, D, L, 常数	信息的数据通道： W#16#EEEE ID 只在首次调用时使用。
EV_ID	INPUT	DWORD	I, Q, M, D, L, 常数	信息号（不允许为 0）只在首次调用时有意义。以后每次 SFB 33 调用时，使用首次调用的信息号与相应的背景数据块 STEP 7 自动地分配信息号以保证信息号一致性。
SEVERITY	INPUT	WORD	I, Q, M, D, L, 常数	事件的重要性：可能值：0~127（0 表示最重要），此参数与信息处理无关。
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数。信息生成完毕。
ERROR	OUTPUT	BOOL	I, Q, M, D, L	ERROR 状态参数。ERROR=TRUE 表示在运行中出现一个故障。详细信息参考参数 STATUS
STATUS	OUTPUT	WORD	I, Q, M, D, L	STATUS 参数：显示故障信息
ACK_DN	OUTPUT	BOOL	I, Q, M, D, L	事件离去状态在显示设备上被确认。初始值：1

参数	声明	数据类型	存储区域	说明
ACK_UP	OUTPUT	BOOL	I, Q, M, D, L	事件进入状态在显示设备上被确认。初始值: 1
SD_i, 1≤i≤10	IN_OUT	ANY	I, Q, M, D, T, C	第 i 个相关数据 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。 注意: 如果 ANY 指针访问一个 DB, 则此 DB 必须一直被指定(例如: P# DB10.DBX5.0 字节 10)

## 故障信息

下表包含了所有定义给 SFB 33 可以通过参数 ERROR 与 STATUS 输出的故障信息。

ERROR	STATUS (十进制)	说明
0	11	警告: 信息丢失, 信号变化/信息不能发送。
0	22	<ul style="list-style-type: none"> <li>相关数据 SD_i 指针错误: <ul style="list-style-type: none"> <li>与数据长度/类型有关</li> <li>相关数据在用户存储器中找不到, 例如: 由于 DB 删除或区间长度故障</li> <li>生成的信息发出, 但不包括相关数据</li> </ul> </li> <li>选定的 SEVERITY 的实际参数退出上限。生成的信息将发出 SEVERITY=127。</li> </ul>
0	25	通讯开始。正在处理信息。
1	1	通讯故障: 连接失败或没有登录
1	4	首次调用时: <ul style="list-style-type: none"> <li>指定的 EV_ID 超出范围或者</li> <li>ANY 指针 SD_i 格式错误</li> <li>一个 SFB 33 CPU 可以发送的最大存储区域超出</li> </ul>
1	10	不能访问本地用户存储器(例如, 试图访问一个删掉的 DB)
1	12	SFB 调用时: <ul style="list-style-type: none"> <li>指定了一个不属于 SFB 36 的背景 DB</li> <li>指定了一个共享 DB 而不是背景 DB</li> </ul>
1	18	EV_ID 已被 SFB 31 或 33~36 使用
1	20	工作存储器不够用。 H-系统: 刷新时调用 SFB
1	21	指定 EV_ID 的信息被禁止。

## 注意

首次调用后, ACK\_UP 与 ACK\_DN 输出为 1, 先前的 SIG 输入值假定为 0。

## 21.5 用 SFB 35 “ALARM\_8P” 生成 8 个信号的带相关数据的块 相关的信息

### 描述

SFB 35 “ALARM\_8P” 线性地扩展了 SFB 33 “NOTIFY” 的功能，它可以生成 8 个信号的信息。

只要已经使能确认触发报告，当检测到一个或多个信号变化时信息就被生成（除非：首次调用时信息总是被发送）。8 个信号具有一个公共的信息 ID，在显示设备上分为 8 个子信息。你可以确认每一个单独的信息或一组信息。

可以在程序中使用 ACK\_STATE 输出参数处理每条信息的确认状态。如果禁止或使能了一个 ALARM\_8P 块信息，这会影响整个 ALARM\_8P 块，不可能禁止或使能单个信号。

每次调用 SFB35 “ALARM\_8P” 都可以使用一个带有两个存储块的存储器。

关于把信号变化入中间存储器参见用 SFB 生成块相关的信息简介中“信号变化检测”一节。

参数	声明	数据类型	存储区域	说明
EN_R	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数，使能接收在块调用时输出 ACK_STATE 是被更新（EN_R=1）还是不被更新（EN_R=0）的决定。
SIG_i, 1 = i = 8	INPUT	BOOL	I, Q, M, D, L	第 i 个被监视的信号。
ID	INPUT	WORD	I, Q, M, D, L, 常数	信息的数据通道：W#16#EEEE ID 只在首次调用时使用。
EV_ID	INPUT	DWORD	I, Q, M, D, L, 常数	信息号（不允许为 0）EV_ID 只在首次调用时使用。以后每次 SFB 35 与相应的 DB，都使用首次调用的信息号。STEP 7 自动分配信息号以保证信息号的一致性。
SEVERITY	INPUT	WORD	I, Q, M, D, L, 常数	事件的重要性：0~127（0 表示最重要），此参数与信息处理无关。
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数：信息生成完毕。
ERROR	OUTPUT	BOOL	I, Q, M, D, L	ERROR 状态参数。 ERROR=TRUE 表示在运行过程中出现故障。详细信息参考参数 STATUS。
STATUS	OUTPUT	WORD	I, Q, M, D, L	STATUS 参数：显示故障信息

参数	声明	数据类型	存储区域	说明
ACK_STATE	OUTPUT	WORD	I, Q, M, D, L	表示8条信息的当前确认状态的位区域：（1：事件已确认，0：未确认）： <ul style="list-style-type: none"> <li>位 0~7 分配给进入事件 SIG_1~SIG_7</li> <li>位 8~15 分配给离开事件 SIG_1~SIG_7</li> </ul> 初始状态：W#16#FFFF，即所有进入的离开的事件都已确认。
SD_i, 1≤j≤10	IN_OUT	ANY	I, Q, M, D, T, C	第 i 个相关数据 这些相关数据适用于所有信息。只允许使用数据类型：BOOL（不允许：位域），BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME <b>注意：</b> 当 ANY 指针访问一个 DB 时，此 DB 必须一直被指定（例如：P#DB10.DBX5.0 字节 10）。

## 故障信息

下表包含了所有定义给 SFB 35 可以通过参数 ERROR 与 STATUS 参数输出的故障信息。

ERROR	STATUS (十进制)	说明
0	11	警告：信息丢失，至少有一个信号变化/信息不能发送。
0	22	<ul style="list-style-type: none"> <li>相关数据 SD_i 指针错误：               <ul style="list-style-type: none"> <li>与数据长度/类型有关</li> <li>相关数据在用户存储器中找不到，例如：由于 DB 删除或区间长度故障</li> </ul> </li> </ul> 生成的信息发出但不包含相关数据 <ul style="list-style-type: none"> <li>选定的 SEVERITY 的实际参数超出上限。生成的信息将被发送，其中 SEVERITY=127。</li> </ul>
0	25	通讯开始。正在处理信息。
1	1	通讯故障：连接失败或没有登录。
1	4	首次调用时： <ul style="list-style-type: none"> <li>指定的 EV_ID 超出范围或者</li> <li>ANY 指针 SD_i 格式错误</li> <li>一个 SFB 35 CPU 可以发送的最大存储区域超出</li> </ul>
1	10	不能访问本地用户存储器（例如，访问一个删除的 DB）
1	12	SFB 调用时： <ul style="list-style-type: none"> <li>指定了一个不属于 SFB 35 的背景 DB</li> <li>指定了一个共享 DB 而不是背景 DB</li> </ul>
1	18	EV_ID 已被 SFB 31 或 33~36 使用。

ERROR	STATUS (十进制)	说明
1	20	工作存储器不够用。H-系统：刷新时调用 SFB
1	21	指定 EV_ID 的信息被禁止。

**注意**

首次调用块后，所有 ACK\_STATE 输出位被置位。先前的 SIG<sub>i</sub>，1≤i≤8 输入值假定为 0。

## 21.6 用 SFB 34 “ALARM\_8” 生成 8 个信号的不带相关数据的块 相关的信息

**描述**

除了不含有相关数据 SD<sub>1</sub>~SD<sub>10</sub> 外 SFB 34 “ALARM\_8” 与 SFB 35 “ALARM\_8P” 相同。

参数	声明	数据类型	存储区域	说明
EN-R	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数，使能接收在块调用时输出 ACK_STATE 是被更新 (EN_R=1) 还是不被更新 (EN_R=0) 的决定。
SIG <sub>i</sub> , 1≤i≤8	INPUT	BOOL	I, Q, M, D, L	第 i 个被监视的信号。
ID	INPUT	WORD	I, Q, M, D, L, 常数	信息的数据通道：W#16#EEEE ID 只在首次调用时使用。
EV_ID	INPUT	DWORD	I, Q, M, D, L, 常数	信息号（不允许为 0）EV_ID 只在首次调用时使用。以后每次 SFB 34 与相应的 DB，都使用首次调用的信息号。STEP 7 自动分配信息号以保证信息号的一致性。
SEVERITY	INPUT	WORD	I, Q, M, D, L, 常数	事件的重要性： 可能值：0~127（0 表示最重要） 此参数与信息处理无关。
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数：信息生成完毕。
ERROR	OUTPUT	BOOL	I, Q, M, D, L	ERROR 状态参数。 ERROR=TRUE 表示在运行过程中出现故障。详细信息参考参数 STATUS。
STATUS	OUTPUT	WORD	I, Q, M, D, L	STATUS 参数：显示故障信息

参数	声明	数据类型	存储区域	说明
ACK_STATE	OUTPUT	WORD	I, Q, M, D, L	表示 8 条信息的当前确认状态的位区域：（1：事件已确认，0：未确认）： <ul style="list-style-type: none"> <li>• 位 0~7 分配给进入事件 SIG_1~SIG_7</li> <li>• 位 8~15 分配给离开事件 SIG_1~SIG_7</li> </ul> 初始状态：W#16#FFFF，即所有进入的离开的事件都已确认。

## 故障信息

下表包含了所有定义给 SFB 34 可以通过参数 ERROR 与 STATUS 参数输出的故障信息。

ERROR	STATUS（十进制）	说明
0	11	警告：信息丢失，至少有一个信号变化/信息不能发送。
0	22	选定的 SEVERITY 的实际参数超出上限。生成的信息将被发送其中 SEVERITY =127。
0	25	通讯开始。正在处理信息。
1	1	通讯故障：连接失败或未登录
1	4	首次调用时，指定的 EV_ID 超出范围。
1	10	不能访问本地用户存储器（例如，访问一个删除的 DB）
1	12	<ul style="list-style-type: none"> <li>• 此 SFB 调用时：</li> <li>• 指定了一个不属于 SFB 34 的背景 DB</li> <li>• 指定了一个共享 DB 而不是背景 DB</li> </ul>
1	18	EV_ID 已被 SFB 31 或 33~36 使用。
1	20	工作存储器不够用。 H-系统：刷新进程中调用 SFB
1	21	指定 EV_ID 的信息被禁止。

## 注意

首次调用块后，由ACK\_STATE的别扭 有输出位被置位，先前的SIG\_I,  $1 \leq i \leq 8$  的输入值假定为 0。



## 21.7 用 SFB 37 “AR\_SEND” 发送存档数据

### 描述

SFB 37 “AR\_SEND” 把存档数据发送到为此而登录泊操作员接口系统。操作员接口系统把注册信息中的相关的存档号告知 CPU。存档数据最长可达 65534 个字节，这取决于 CPU 中可用的存储器以及使用的地址区域。正使用的操作员接口系统的缺省设置必须考虑到存档数据的结构中。

在块被调用后控制参数 REQ 的一个上升沿激活数据发送。存档数据的起始地址由 SD\_1 指定，数据区域的长度由 LEN 指定。数据的传输与用户程序的执行不是同步的。状态参数 DONE 为 1 表示数据传输顺利完成，输入参数 R 的一个上升沿取消数据传输。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	控制参数：请求
R	INPUT	BOOL	I, Q, M, D, L, 常数	控制参数复位：当前任务取消
ID	INPUT	WORD	I, Q, M, D, L, 常数	信息的数据通道：W#16#EEEE ID 只在首次调用时使用。
AR_ID	INPUT	DWORD	I, Q, M, D, L, 常数	存档号（不允许为 0） AR_ID 只在首次调用时使用。以后每次 SFB 37 与相应的 DB 调用时都使用首次调用的存档号。STEP 7 自动分配存档号以保证存档号的一致性。
DONE	OUTPUT	BOOL	I, Q, M, D, L	DONE 状态参数：发送结束
ERROR	OUTPUT	BOOL	I, Q, M, D, L	ERROR 状态参数 ERROR=TRUE 表示运行过程中出现故障。 详细信息参考参数 STATUS。
STATUS	OUTPUT	WORD	I, Q, M, D, L	STATUS 参数：显示故障信息。
SD_1	IN_OUT	ANY	I, Q, M, D, T, C	存档数据指针。此处不考虑长度。 只有下列数据类型允许使用：BOOL（不允许：位域），CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME 存档数据只能是 PLC 指定的结构。 <b>注意：</b> 如果 ANY 指针访问一个 DB，则此 DB 必须一直被指定（例如：P# DB10.DBX5.0 字节 10）。
LEN	IN_OUT	WORD	I, Q, M, D, L	发送的数据区域的长度以字节表示。

## 故障信息

下表包含所有定义给 SFB 37 可以通过参数 ERROR 与 STATUS 参数输出的故障信息。

ERROR	STATUS (十进制)	说明
0	11	警告：因为前面的任务正在执行新任务不能激活。
0	25	通讯开始。任务正在进行中。
1	1	通讯故障
1	2	确认无效，功能不被执行
1	3	指定 AR_ID 的信息未注册。
1	4	<ul style="list-style-type: none"> <li>存档数据指针 SD_i 错误，与数据长度或类型有关。</li> <li>首次调用时，指定的 AR_ID 超出范围。</li> </ul>
1	5	执行了请求复位。
1	7	因为当前的功能已执行完毕或还未激活（块处于不正确的状态）RESET 任务无效。
1	10	不能访问本地用户存储器（例如，访问一个删掉的 DB）。
1	12	SFB 调用时： <ul style="list-style-type: none"> <li>指定了一个不属于 SFB 37 的背景 DB</li> <li>指定了一个共享 DB 而不是背景 DB</li> </ul>
1	18	AR_ID 已被 SFB 37 使用。
1	20	工作存储器不够用。 H-系统：刷新过程中调用 SFB。

## 21.8 用 SFC 10 “DIS\_MSG” 禁止使能块相关的、符号相关的以及组状态信息

### 描述

用 SFC 10 “DIS\_MSG”（禁止信息）可以禁止使能由 SFB 产生的块相关的信息，符号相关的信息（SCAN）以及组状态信息。用输入参数 MODE 与 MESGN 选择禁止使能的信息。调用 SFC 10 “DIS\_MSG”，只有在 SFC 10 中信息的禁止使能功能还没激活的情况下才可能顺利地禁止信息的使能。

在 SFC 10 调用时正准备发送仍在内部缓冲区中的信息不再能被禁止使能和发送。一条禁止使能的信息可由 “NOTIFY”，“NOTIFY\_8P”，“ALARM”，“ALARM\_8P” 以及 “ALARM\_8”。SFB 的输出参数 ERROR 与 STATUS 指示出来。

当 SFC10 调用时，设置输入参数 REQ 为 1 就开始禁止使能功能。

### 如何执行 SFC 10 的功能

禁止使能异步执行，也就是说它可以通过多次 SFC 10 调用来实现。

- 首次调用（REQ =1）时，SFC 10 检查输入参数，调图占用需要的系统资源。如果成功，W#16#7001 进入 RET\_VAL，BUSY 置位，禁止信息使能开始。  
如果不成功，故障信息进入 RET\_VAL，任务终止。这种情况下 BUSY 无意义。
- 如果同时还有更多的调用，W#16#7002 进入 RET\_VAL（任务仍被 CPU 执行）。BUSY 置位。更多调用不影响当前任务。
- 最后一次 SFB 调用，若无故障产生 W#16#0000 进入 RET\_VAL。BUSY 为 0。如果故障产生，故障信息进入 RET\_VAL，并且 BUSY 无意义。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	REQ = 1:触发禁止使能
MODE	INPUT	BYTE	I, Q, M, D, L, 常数	选择禁止使能信息的参数，见下表
MESGN	INPUT	DWORD	I, Q, M, D, L, 常数	信息号，只在 MODE 设为 5, 6, 7 时有意义，表示允许单条信息被禁止使能。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息，见表“故障信息”
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 禁止使能还未被完成。

### MODE 输入参数

下表列出了 MODE 输入参数的允许值。

值	含义
0	所有 CPU 中由 SFB 产生的块相关的，符号相关的以及组状态信息。
1	所有 CPU 中由 SFB 产生的块相关的信息，即由“NOTIFY”，“NOTIFY_8P”，“ALARM”，“ALARM_8P”以及“ALARM_8” SFB 产生的所有信息。
2	所有 CPU 中的组状态信息
3	所有 CPU 中符号相关的信息（SCAN）
5	“符号相关的信息”类中的一条信息。
6	“用 SFB 产生的块相关的信息”类中的一条信息
7	“组状态信息”类中的一条信息

## 故障信息

故障代码（W#16#...）	说明
0000	禁止使能完成，没故障
7000	REQ = 0 首次调用：禁止使能没有激活。
7001	REQ = 1 首次调用：禁止使能触发。
7002	更多调用：禁止使能已经执行。
8081	参数访问故障
8082	MODE 数值非法。
8083	信息号超出允许值范围。
8084	由 MODE 也可能是由 MESGN 指定的信息没有注册。
80C3	由 MODE 也可能是由 MESGN 指定的信息禁止使能当前不能执行- SFC 10 已经禁止了信息的使能。

## 21.9 用 SFC 9 “EN\_MSG” 使能块相关的，符号相关的以及组状态信息。

### 描述

用 SFC 9 “EN\_MSG”（使能信息），可以使能先前禁止使能的块相关的，符号相关的以及组状态信息。禁止信息使能或者是在显示设备上或者是通过 SFC 10 “DIS\_MSG”。

用 MODE 与 MESGN 输入参数指定使能的信息。只有在 SFC 9 “EN\_MSG” 中信息的使能功能还没激活的情况下方可能顺利地使能信息。

把 SFC 9 中的输入参数 REQ 配置为 1 就可以开始使能功能。

### 如何执行 SFC 9 的功能

使能异步执行，即可以通过多次 SFC 9 调用来实现。

- 首次调用 (REQ = 1) 时，SFC 9 检查输入参数，调图占用需要的系统资源。如果成功，W#16#7001 进入 RET\_VAL，BUSY 置位，信息使能开始。  
如果不成功，故障信息进入参数 RET\_VAL，任务终止。这种情况下 BUSY 无意义。
- 如果同时还有更多的调用，W#16#7002 进入 RET\_VAL（任务仍被 CPU 执行）。BUSY 置位。更多调用不影响当前任务。
- 最后一次 SFB 调用，若无故障产生 W#16#0000 进入 RET\_VAL。BUSY 为 0。如有故障产生，故障信息进入 RET\_VAL，并且 BUSY 无意义。

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L	REQ = 1: 触发使能
MODE	INPUT	BYTE	I, Q, M, D, L, 常数	选择被使能的信息的参数
MESGN	INPUT	DWORD	I, Q, M, D, L, 常数	信息号，只在 MODE 设为 5, 6, 7 时有意义，表示允许一条信息使能。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息，见下表
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: 使能还未完成。

### MODE 输入参数

下表列出了 MODE 输入参数的允许值。

值	含义
0	所有 CPU 中由 SFB 产生的块相关的，符号相关的以及组状态信息。
1	所有 CPU 中由 SFB 产生的块相关的信息，即由“NOTIFY”，“NOTIFY_8P”，“ALARM”，“ALARM_8P”以及“ALARM_8”SFB 产生的所有信息。
2	所有 CPU 中的组状态信息
3	所有 CPU 中符号相关的信息（SCAN）
5	“与符号相关的信息”类中的一条信息。

值	含义
6	“由 SFB 产生的块相关的信息”类中的一条信息
7	“组状态信息”类中的一条信息

## 故障信息

故障代码 (W#16#...)	说明
0000	使能完成，没有故障。
7000	REQ = 0 首次调用：使能没有激活。
7001	REQ = 1 首次调用：使能触发。
7002	更多调用：使能已经执行。
8081	参数访问故障
8082	MODE 数值非法。
8083	信息号超出允许值范围。
8084	由 MODE 也可能是由 MESSN 指定的信息没有注册。
80C3	由 MODE 也可能是由 MESSN 指定的信息使能当前不能执行- SFC 9 已经使能了信息。

## 21.10 生成块相关的信息的 SFB 的启动响应

### 重启（暖启动）

在 NO\_INIT 状态。存储在背景 DB 中的实际参数保持不变。参数 ID 和 EV\_ID 将在下次块调用时使用。

### 完全重启

完全重启时，生成块相关的信息的 SFB 的背景 DB 的内容复位到初始值。

### 热重启

热重启时，SFB 的行为就像继续执行的功能块。它们在中断点继续执行。

### 内存复位

内存复位总是导致所有连接中止，因此没有登录站的信息，用户程序被删除。如果插有存储卡，和程序执行相关的部分被再次从卡上装入 CPU，CPU 执行暖启动或冷启动（实际上总为冷启动，因为清内存后所有用户数据被初始化）。

## 21.11 生成块相关的信息的 SFB 对故障的响应

### 连接中断

发现与 SFB 相关的连接中断。如果连接中断，所有的站从 CPU 内部登录块相关信息的列表中移出。这些站的任何悬而未决的信息将被删除。

如果其他站在连接中断后仍然在登录状态，它们将继续接收信息。只有不再有任何连接到登录的站，SFB 才停止发送信息。输出参数 ERROR 与 STATUS 中提示这些状态（ERROR = 1，STATUS = 1）。

### 到用户程序的故障接口

如果在执行生成块相关的信息的 SFB 时出现故障，输出参数 ERROR 置 1，STATUS 输出相应的故障标识。可以在程序中使用这些故障信息。

可能的故障例如：

- 由于缺少资源发送不可能
- 一个监视信号的访问故障。

## 21.12 用 SFC 生成块相关的信息简介

### 生成块相关的信息的 SFC

可以用下列 SFC 生成与块相关的信息。

- SFC 17 “ALARM\_SQ”
- SFC 18 “ALARM\_S”
- SFC 107 “ALARM\_DQ”
- SFC 108 “ALARM\_D”

这些 SFC 具有下列特征：

- 当信号状态为 1 时，SFC17 “ALARM\_SQ” 和 “SFC 107” “ALARM\_DQ” 发出的信息可以被登录的显示设备确认。SFC 18 与 SFC108 的信息总是被毫无疑问地确认。它不是在检测到信号变化时生成信息，而是在每次 SFC 调用时。详细的信息参见用 SFC 17 “ALARM\_SQ” 生成与块相关的可确认的信息和用 SFC 18 “ALARM\_S” 生成永久确认的与块相关的信息。
- “ALARM\_S” 和用 SFC 107 “ALARM\_DQ” 与 108 “ALARM\_D” 生成可确认的与块相关的信息一节。
- 随着块的执行，相关的数据 SD\_1 被完全读出。与更高优先级的一致性相比较，下列相关的数据是一致的：
  - 简单的数据类型（位，字节，字与双字）
  - 指定给特定 CPU 的达最大长度的一个数据类型为字节的数组（见/71/，/101/）

## SFC 19 “ALARM\_SC”

用 SFC19，可以查询：

- 最后一个“进入状态信息”的确认状态，最后一个 SFC17/SFC107 的信号状态或者
- 最后一次调用 SFC 18/SFC 108 时的信号状态。

## 登录显示设备

只有至少一个显示设备注册了块相关的信息，生成块相关的信息的 SFC 才会在调用时发出信息。

## 信息存储

为避免通讯系统堵塞时信号丢失，SFC17，18，107 与 108 能缓冲两条信息。

如果信息丢失，可在 RET\_VAL 中被告知。下次信息能被发送时，登录的显示设备将得到通知。

## 用 SFC17 “ALARM\_SQ” 和 107 “ALARM\_DQ” 信息确认生成确认信息

如果在显示设备上确认了一条输入 Q 事件信息，这条确认信息首先发送到信息源 CPU。接着确认的信息被分送到所有为此而登录的站。

## 禁止和使能信息

用 SFC 17 “ALARM\_SQ”，SFC 18 “ALARM\_S”，SFC 107 “ALARM\_DQ” 或者 SFC 108 “ALARM\_D” 生成的块相关的信息不能被禁止然后再次使能。

包含 SFC 17/SFC 18 调用的用户程序的变化。

---

### 注意

当下载块到一个已经使用了 SFC 17/SFC 18 的 CPU 上，可能前面的块已经发送了一个输入状态信息，但新块没有发送相应的离开状态信息。这意味着信息保持在 CPU 的内部信息存储区。

当你用 SFC 17/SFC 18 删除块时也会发生这种情况，可以通过将 CPU 切换到 STOP 状态然后进行暖启动或冷启动来删去 CPU 内部信息存储器中的信息。

---

## 包含 SFC 17/SFC 18 调用的用户程序的变化。

即使你的程序可能包含 SFC107 和/或 SFC108 的调用，描述的程序修改可能引起信息驻留在内部信息存储器。

与 SFC 17/SFC 18 调用所占有的系统资源相反，在不把 CPU 打到停止状态的条件下也可以释放 SFC 107/SFC 108 对系统资源的占用，通过调用 SFC 106 “DEL\_SI” 来实现，参见用 SFC 106 释放动态占用的系统资源。在调用 SFC 106 “DEL\_SI” 释放动态占用的系统资源前，可以用 SFC 105 “READ\_SI” 来提取当前占用 CPU 系统资源的信息，参见用 SFC 105 “READ\_SI” 读动态占用的系统资源。

## 可传输的数据量

对于 SFC ALARM\_S, ALARM\_SQ, ALARM\_D 以及 ALARM\_DQ，使用相关数据 SD 可传输的数据总量不能超过一个最大长度。长度计算如下：

$$\text{maxleng} = \min(\text{pdu\_local}, \text{pdu\_remote}) - 48$$



定义:

- pdu\_local: CPU 数据块的最大长度 (SZL\_ID W#16#0131, INDEX 1, 变量 pdu)
- pdu\_remote: 显示设备数据块的最大长度

例如:

一个 CPU 414-1 发送一条信息到编程器 PG 760 (通过 MPI)

pdu\_local = 480 字节, pdu\_remote = 480 字节

结果:

$\text{maxleng} = \min(480, 480) - 48 = 480 - 48 = 432$

最大可传输的数据长度每个 SFC 为 432 字节。

## 21.13 用 SFC 17 “ALARM\_SQ” 生成可确认的与块相关的信息和用 SFC 18 “ALARM\_S” 生成永久确认的块相关的信息

---

**注意**

新建程序只应用 SFC 107/108, 因为它们提供管理系统资源的改进选项。

---

### 描述

每次调用时, SFC 17 “ALARM\_SQ” 与 SFC 18 “ALARM\_S” 生成一条可增加相关数据的信息。该信息将发给所有注册了该信息的站。SFC 17 和 SFC 18 提供了发送信息的简单机制, 必须确保只在触发信号 SIG 和上次调用相比较有变化时再调用 SFC 17 或 SFC 18。否则将在 RET\_VAL 中提示, 并且无信息发出。初次调用 SFC 17 或 SFC 18 时, 必须确认 SIG 输入为 1, 否则 RET\_VAL 中会包含一个故障信息, 并且无信息发出。

---

**注意**

在分配了合适的系统属性的 FB 中调用 SFC 17 与 SFC 18。关于分配块系统属性的详细信息参见/232/与/233/。

---

### 系统资源的使用

当用 SFC 17 “ALARM\_SQ” 与 18 “ALARM\_S” 生成信息时, 操作系统使用持续一个信号周期的一个系统资源。

对 SFC 18 “ALARM\_S”, 信号周期从 SFC 调用 SIG=1 到另一次调用 SIG=0。对于 SFC 17 “ALARM\_SQ”, 如果需要这个时间段还包括直到输入信号被一个报告显示设备确认。

如果在信号周期内, 信息生成块过载或被删除, 相关的系统资源保持占用直到下一次重启 (暖启动)。

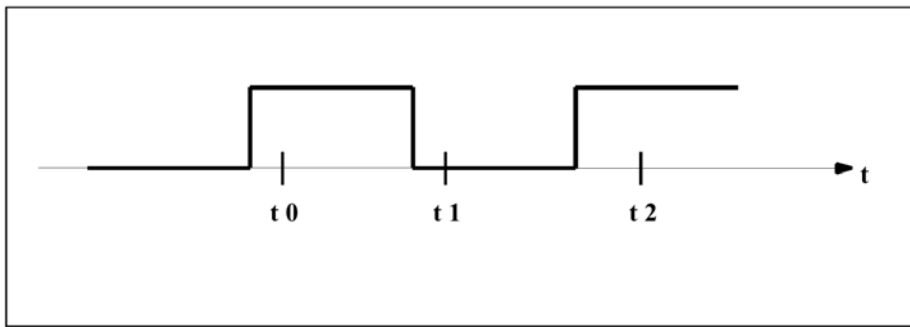
## 确认信息

当监视信号为 1 时，可确认由 SFC17 发送的信息。可以用 SFC 19 “ALARM\_SC” 查询最后调用 SFC 的信号状态和最后“进入事件信息”的确认状态。用 SFC 18 发送的信息总是毫不怀疑地确认。可用 SFC 19 “ALARM\_SC” 查询最后调用 SFC 18 时的信号状态。

## 信号状态的临时存储

SFC 17 “ALARM\_SQ” 与 SFC 18 “ALARM\_S” 临时占用系统资源。它们记录最后两个信号状态，包括时间标识和相关数据。如果在最后两个有效的 SFC 调用的信号状态没有被发送（信号溢出）时调用 SFC 17/18，则当前的和最后的信号状态被忽略，缓冲区的溢出 ID 被置位。在下一个可能的机会里，倒数第二个信号溢出 ID 被发送出去。

例如：



$t_0$ ,  $t_1$  和  $t_2$  是 SFC 17 或 SFC 18 被调用的点。如果  $t_0$ ,  $t_1$  的信号状态在  $t_2$  时间发送， $t_0$  与  $t_1$  的信号状态被忽略， $t_0$  的溢出 ID 被置位。

## 调用溢出

如果调用 SFC 17 或 SFC 18 的次数超出了 CPU 系统资源的最大数，将导致资源短缺（调用溢出）。这将同时被 RET\_VAL 和登录的显示设备提示。

调用 SFC 17 或 SFC 18 的最多次数取决于特定的 CPU 类型。这些信息在 /70/ 与 /101/ 中。

参数	声明	数据类型	存储区域	说明
SIG	INPUT	BOOL	I, Q, M, D, L	触发信息的信号
ID	INPUT	WORD	I, Q, M, D, L, 常数	信息的数据通道: W#16#EEEE
EV_ID	INPUT	DWORD	I, Q, M, D, L, 常数	信息号（不允许为 0），STEP 7 自动分配信息号以确保信息号的一致性。
SD	INPUT	ANY	I, Q, M, D, T, C	相关数据 最大长度：12 字节 允许使用下列数据类型 BOOL（不允许：位区域），BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息

## 故障信息

故障代码 (W#16#...)	说明
0000	无故障产生
0001	<ul style="list-style-type: none"> <li>相关数据超出最大允许长度或者</li> <li>无法访问用户存储区（例如，访问一个删除的 DB）。信息发出。</li> </ul>
0002	警告：最后一个可用的信息确认存储器已被使用。
8081	定义的 EV_ID 超出允许的范围。
8082	由于 CPU 没有更多的资源给 SFC 生成与块相关的信息导致信息丢失。
8083	信息丢失，因为相同的信号改变已经存在但不能发出（信号溢出）。
8084	当前的 SFC 17 或 SFC 18 与上次调用相比触发信号 SIG 值相同。
8085	指定 EV_ID 的信息没有注册。
8086	指定 EV_ID 的 SFC 已在较低优先级中运行。
8087	SFC 17 或 SFC 18 首次调用时，信息触发信号为 0。
8088	指定的 EV_ID 已被另一个系统资源（SFC 17, 18, 107, 108）使用

## 21.14 用 SFC 19 “ALARM\_SC” 查询最后 ALARM\_SQ/ALARM\_DQ 进入事件信息的确认状态

## 描述

用 SFC19 “ALARM\_SC” 可以查询：

- 最后 ALARM\_SQ/ALARM\_DQ 进入事状态信息的确认状态与最后调用 SFC 17 “ALARM\_SQ” /SFC 107 “ALARM\_DQ” 触发信息的信号状态
- 最后调用 “SFC 18 “ALARM\_S” /SFC 108 “ALARM\_D” 触发信息的信号状态。

假设在信息配置时配置了信息号，则这个唯一的信息号就代表了信息或信号，SFC 19 “ALARM\_SC” 依此访问 SFC17 或 SFC18/SFC107/SFC108 临时占用的内存。

参数	声明	数据类型	存储区域	说明
EV_ID	INPUT	DWORD	I, Q, M, D, L, 常数	查询的信息号最后一次 SFC 调用时的信号状态或者最后进入状态信息的确认状态（只对于 SFC 17 与 SFC 107）。
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息
STATE	OUTPUT	BOOL	I, Q, M, D, L	最后一次 SFC 调用时触发信息的信号状态。
Q_STATE	OUTPUT	BOOL	I, Q, M, D, L	如果指定的 EV_ID 参数属于一个 SFC 18/SFC 108 调用，此值为 1。

参数	声明	数据类型	存储区域	说明
				如果指定的 EV_ID 参数属于一个 SFC 17/SFC 107 调用,则为最后进入状态信息的确认状态: 0: 未确认 1: 已确认

## 故障信息

故障代码 (W#16#...)	说明
0000	无故障产生
8081	指定的 EV_ID 超出允许范围。
8082	没有这个 EV_ID 当前占用的内存 (可能的原因: 相应的信号状态仍不为 1, 或者此信号状态已返回 0)

## 21.15 用 SFC 107 “ALARM\_DQ” 与 108 “ALARM\_D” 生成可确认的与永久确认的块相关的信息

## 描述

每次调用 SFC 107 “ALARM\_DQ” 与 108 “ALARM\_D” 都生成一条可附加相关数据的信息, 与 SFC 17 “ALARM\_SQ” 与 18 “ALARM\_S” 相应。

用 SFC 107 “ALARM\_DQ” 与 108 “ALARM\_D” 生成信息, 操作系统临时占用持续一个信号周期的系统资源。

SFC 108 “ALARM\_D” 的信号周期始于 SIG=1 时的 SFC 调用, 终止于 SIG=0 的新一次调用。SFC 107 “ALARM\_DQ” 的信号周期将延长到输入信号在一个登录的显示设备上被确认。

对 SFC 108 来说, 信号循环从 SIG=1 调用 SFC 开始, 直到 SIG=0。而对 SFC 107 除上述时间外, 还包括按需要任一显示设备确认新来的信息。

如果在信号周期中, 信息生成块过载或被删除, 相关的系统资源将保持占用直到下一次重启 (暖启动)。

与 SFC 107 和 SFC 18 相比, SFC 107 “ALARM\_DQ” 与 SFC 108 “ALARM\_D” 的附加功能是你可管理被占用的系统资源:

- 借助 SFC 105 “READ\_SI” 可提取被占用的系统资源的信息。
- 用 SFC 106 “DEL\_SI” 能释放被占用的系统资源。这对被永久占用的系统资源有特殊意义。如果在程序变更时删除了一个包含 SFC107 或 SFC108 调用的 FB, 当前占用的系统资源将一直被占用直到下一次暖启动。当改变程序, 重新下载包含 SFC 107 或 SFC 108 的 FB, 可能发生 SFC 107 和 108 不再生成更多信息的情况。

SFC 107 和 108 比 SFC 17 和 18 多一个参数，名为输入 CMP\_ID。使用此输入把 SFC 107 和 108 生成的信息赋值给逻辑区域，例如系统的部分。如在一个 FB 中调用 SFC 107/SFC 108，显然是要把相应背景 DB 号赋给 CMP\_ID。

参数	声明	数据类型	存储区域	说明
SIG	INPUT	BOOL	I, Q, M, D, L	信息触发信号
ID	INPUT	WORD	I, Q, M, D, L, 常数	信息的数据通道: W#16#EEEE
EV_ID	INPUT	DWORD	I, Q, M, D, L, 常数	信息号（不允许为 0）。STEP 7 自动分配信息号以保证信息号的一致性。
CMP_ID	INPUT	DWORD	I, Q, M, D, L, 常数	元件标识（不允许为 0）ID 为相应信息赋给的部分系统的标识： <ul style="list-style-type: none"> <li>低字：1~65535；</li> <li>高字：0</li> </ul> 如果应用这些推荐值，将不会和 SIEMENS 的程序会发生任何冲突。
SD	INPUT	ANY	I, Q, M, D, T, C	相关数据 最大长度：12 字节 只允许使用下列数据类型：BOOL（不允许：位区域），BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息

## 故障信息

故障代码 (W#16#...):	说明
0000	无故障产生
0001	<ul style="list-style-type: none"> <li>相关数据的长度超出最大允许值，或者</li> <li>不能访问用户存储器（例如，访问一个删除的 DB）生成的信息发出。</li> </ul>
0002	警告：最后一个可用的信息确认存储器被占用。
8081	指定的 EV_ID 不在有效范围内。
8082	由于 CPU 没有生成与块相关的信息的 SFC 所用的资源导致信息丢失。
8083	信息丢失，相同的信号变化已经存在但不能发出（信号溢出）。
8084	对于当前与先前的 SFC 107/SFC 108 调用，信息触发信号 SIG 具有相同的值。
8085	指定的 EV_ID 的信息没有注册。
8086	已经在较低的优先级中调用了处理指定 EV_ID 信息的 SFC。
8087	SFC 107/SFC 108 开始调用时，信息触发信号为 0。
8088	指定的 EV_ID 已被另外的系统资源使用（SFC 17, 18, 107, 108）。
8089	把 0 赋值给了 CMP_ID。

## 21.16 用 SFC 105 “READ\_SI” 读取动态系统资源

### 用 SFC 107 与 108 生成信息时系统资源的动态开发情况

用 SFC 107 “ALARM\_DQ” 与 108 “ALARM\_D” 生成信息时，操作系统临时占用系统存储空间。

例如，如果没有删除 CPU 中包含 SFC107 或 108 调用的 FB，可能导致相应的系统资源被永久占用。如果重装带 SFC107 或 SFC108 调用的 FB。可能导致 SFC 107 或 108 不能再被正确处理。

### 描述

用 SFC 105，可读出被 SFC 107 和 108 在信息生成时当前占用的系统资源。通过取这里的 EV\_ID 和 CMP\_ID 的数值来完成。数值传送到 SFC105 的参数 SI\_ID。

SFC 105 “READ\_SI” 有 4 种可能的操作方式，说明见下表，通过 MODE 参数设置期望的操作方式。

MODE	读取哪些被 SFC 107/SFC 108 占用的系统资源？
1	全部（调用 SFC 105，且 SI_ID: =0）
2	调用 SFC 107/SFC 108，指定 EV_ID: =ev_id 占用的系统资源（调用 SFC 105，令 SI_ID: =ev_id）
3	调用 SFC 107/SFC 108，CMP_ID: =cmp_id 情况下占用的系统资源（调用 SFC 105 令 SI_ID: =ev_id）
0	先前调用在 MODE=1 或 MODE=3 时无法读出的附加系统资源，因为指定的目标区域 SYS_INST 大小。

### 工作原理

如果在调用 SFC 105，且 MODE=1 或 MODE=3 时没有选定足够大的 SYS\_INST 目标区域，它将包含通过 MODE 参数选择的所有当前占用的系统资源的内容。

高系统资源将导致相应高的 SFC 运行时间。

资源上的高系统负载将导致配置的最大周期监视时间超调。

可如下处理运行时间问题：选择一个相对小的 SYS\_INST 目标区域。如果 SFC 不能把全部系统资源读入 SYS\_INST，RET\_VAL=W#16#0001 会提示。这时设 MODE=0 调用 SFC 105，用和上次相同的 SI\_ID，直到 RET\_VAL 为 W#16#0000。

### 注意

因为操作系统不协调属于读任务的 SFC 105 调用，所以应当把 SFC 105 调用置于相同的优先级。

### 目标区域 SYS\_INST 的结构

存放占用系统资源的目标区域必须位于一个 DB 中。应该正确定义这个目标区域为一个结构区域，构建的结构如下：

结构元素	数据类型	说明
SFC_NO	WORD	占用系统资源的 SFC 号
LEN	BYTE	用字节表示的结构长度，包括 SFC_NO 与 LEN: B#16#0C
SIG_STAT	BOOL	信号状态
ACK_STAT	BOOL	进入事件（上升沿）的确认状态
EV_ID	DWORD	报文号
CMP_ID	DWORD	部分系统 ID

参数	声明	数据类型	存储区域	说明
MODE	INPUT	INT	I, Q, M, D, L, 常数	任务标识 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 SFC 107/SFC 108 指定 EV_ID = ev_id 占用的系统资源</li> <li>3: 读取调用 SFC 107/SFC 108, CMP_ID = cmp_id 时占用的系统资源</li> <li>0: 后续调用</li> </ul>
SI_ID	INPUT	DWORD	I, Q, M, D, L, 常数	被读系统资源的 ID。可能值 <ul style="list-style-type: none"> <li>0, 如果 MODE=1</li> <li>报文号 ev_id。如果 MODE=2</li> <li>系统部分 ID cmp_id, 如果 MODE=3</li> </ul>
RET_VAL	OUTPUT	INT	I, Q, M, D, L	返回值（故障信息或任务状态）。
N_SI	OUTPUT	INT	I, Q, M, D, L	用 SYS_INT 输出的系统资源号。
SYS_INT	OUTPUT	ANY	D	存放系统资源的目标区域。

## 故障信息

故障代码 (W#16#...)	说明
0000	无故障产生
0001	因为选定的 SYS_INT 目标区域太小无法读取全部系统资源。
8081	（只对于 MODE=2 或 3）把 0 赋给了 SI_ID。
8082	（只对于 MODE=1）把一个不为 0 的值赋给了 SI_ID。
8083	（只对于 MODE=0），给 SI_ID 分配了一个与先前 SFC 调用，MODE=1 或 3 时相比不同的值。
8084	把一个非法数值赋给 MODE。
8085	SFC 105 已在另外一个 OB 中被处理。
8086	目标区域 SYS_INST 相对于系统资源太小。
8087	目标区域 SYS_INST 在 DB 中不存在。

## 21.17 用 SFC 106 “DEL\_SI” 删除动态系统资源

### 用 SFC 107 与 SFC 108 生成信息时系统资源的动态开发情况

用 SFC 107 “ALARM\_DQ” 与 108 “ALARM\_D” 生成信息时，操作系统临时占用系统存储空间。

例如，如果没有删除 CPU 中包含 SFC107 或 108 调用的 FB，可能导致相应的系统资源被永久占用。如果重装带 SFC107 或 SFC108 调用的 FB。可能导致 SFC 107 或 108 不能再被正确处理。

### 描述

用 SFC 106 “DEL\_SI” 可以删除当前占用的系统资源。SFC 106 “DEL\_SI” 有三种可能的操作方式，见下表说明，通过 MODE 参数设置期望的操作方式。

MODE	删除哪些被 SFC 107/SFC 108 占用的系统资源？
1	全部（调用 SFC 106，设 SI_ID: =0）
2	调用 SFC 107/SFC 108，指定 EV_ID: =ev_id 占用的系统资源（调用 SFC 106，令 SI_ID: =ev_id）
3	调用 SFC 107/SFC 108，CMP_ID: =cmp_id 情况下占用的系统资源（调用 SFC 106 令 SI_ID: =ev_id）

参数	声明	数据类型	存储区域	说明
MODE	INPUT	INT	I, Q, M, D, L, 常数	任务标识 可能值 <ul style="list-style-type: none"> <li>1: 删除所有系统资源</li> <li>2: 删除 SFC 107/SFC 108 调用时指定 EV_ID = ev_id 占用的系统资源</li> <li>3: 删除调用 SFC 107/SFC 108, CMP_ID = cmp_id 情况下占用的系统资源</li> </ul>
SI_ID	INPUT	DWORD	I, Q, M, D, L, 常数	删除的系统资源的 ID。可能值 <ul style="list-style-type: none"> <li>0, 如果 MODE=1</li> <li>报文号 ev_id, 如果 MODE=2</li> <li>系统部分标识 ID cmp_id, 如果 MODE=3</li> </ul>
RET_VAL	OUTPUT	INT	I, Q, M, D, L	故障信息



故障信息

故障代码 (W#16#...)	说明
0000	无故障产生
8081	(只对于 MODE=2 或 3) 把 0 赋给了 SI_ID。
8082	(只对于 MODE=1) 把一个不为 0 的值赋给了 SI_ID。
8084	把一个非法数值赋给了 MODE。
8085	SFC 106 正在执行。
8086	不是所有选定的系统资源都能删除。因为至少有一种资源在 SFC 106 调用时被使用着。

## 22 IEC 定时器和 IEC 计数器

### 22.1 用 SFB 3 “TP” 生成一个脉冲信号

#### 描述

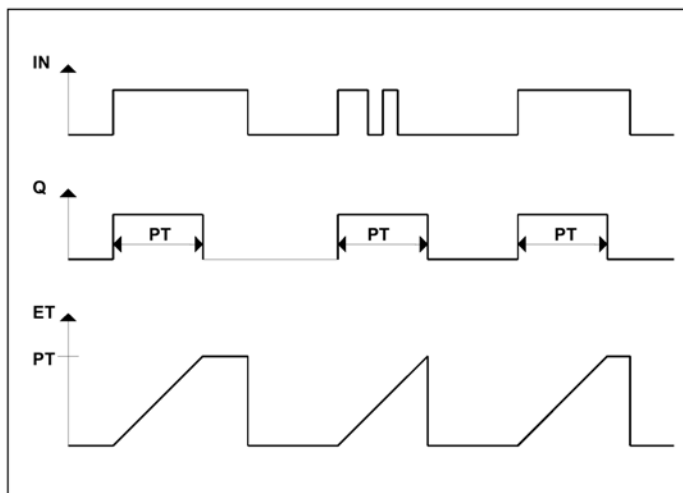
SFB 3 “TP” 利用 PT 长度可以产生一个脉冲，此定时器只能运行在起动的或运行模式。

在输入端 IN 的上升沿可以击活脉冲。输出端 Q 保持置位状态不管输入端 PT 是否变化（换句话说，甚至 IN 输入端在 PT 完成前再次由 0 到 1 变化）ET 输出端记录了输出 Q 已经被置位的时间。ET 的最大值等于 PT 输入端的值，当 IN 输入端变为 0 时，ET 输出端被复位，但是必须在 PT 完成后才可以。

SFB 3 “TP” 遵照 IEC 1131-3 标准。

系统在冷启动时对 SFB 3 “TP” 进行复位。如果你想在系统暖启后对此 SFB 进行初始化，你必须在 OB100 中调用当 PT = 0 时的 SFB3。如果该 SFB 也被其它块调用了你可以重新设置这些参数，例如，对优先级高的块进行初始化。

#### 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	脉冲持续时间, PT 必须为正值。(注意: 所取值不能超过 TIME 数据类型所规定的范围)
Q	OUTPUT	BOOL	I, Q, M, D, L	时间状态
ET	OUTPUT	TIME	I, Q, M, D, L	完成时间

## 22.2 用 SFB 4 “TON” 生成一个延时接通信号

### 描述

SFB 4 “TON” 利用 PT 时间可以产生一个延时接通的信号，此定时器只能运行在起动的运行模式。

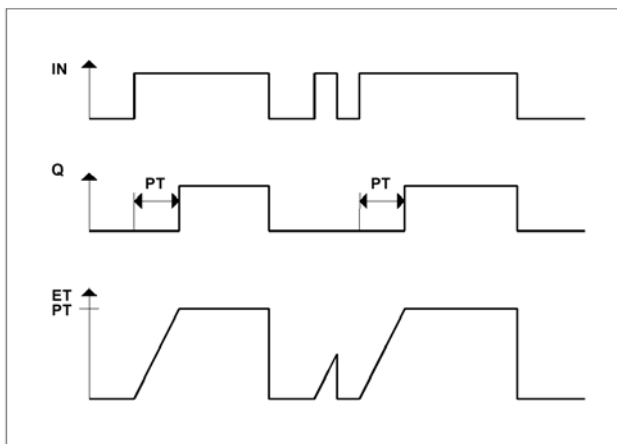
在输入端 IN 的上升沿可以使输出端 Q 在经过定时时间 PT 后产生一个上升沿。Q 端会一直保持为 1，直到输入端 IN 再次变为 0。如果在定时时间 PT 完成前输入端变为 0，则 Q 端保持为 0。

输出端 ET 记录了从输入端 IN 最后一次产生的上升沿开始到现在的时间，ET 端的最大值等于 PT 端输入值，当输入端 IN 变为 0 时，ET 被复位。

SFB 4 “TON” 遵照 IEC 1131-3 标准。

系统在冷启动时对 SFB 4 “TON” 进行复位。如果你想在系统暖启动以后对此 SFB 进行初始化，你必须在 OB100 中调用当 PT = 0 时的 SFB4。如果该 SFB 也被其它块调用了，你可以重新设置这些参数，例如，对优先级高的块进行初始化。

### 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	输入端 IN 上升沿延时时间，PT4 必须为正数。 (注意：所取值不能超过 TIME 数据类型所规定的范围)
Q	OUTPUT	BOOL	I, Q, M, D, L	时间状态
ET	OUTPUT	TIME	I, Q, M, D, L	完成时间

## 22.3 用 SFB 5 “TOF” 产生一个延时断开信号

### 描述

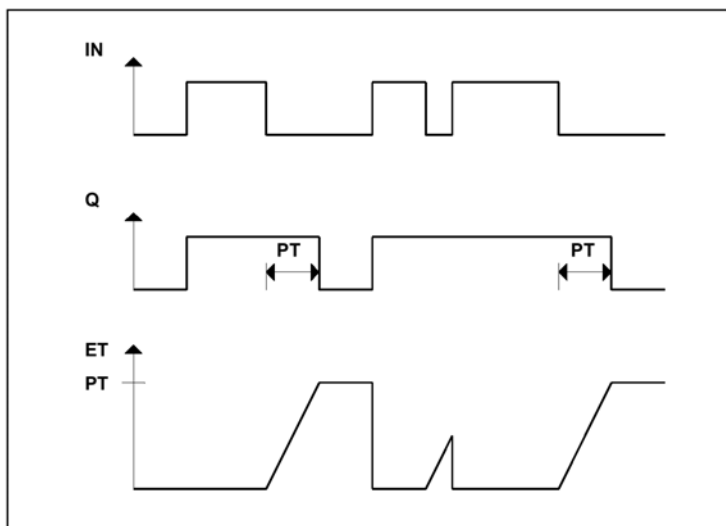
SFB 5 “TOF” 利用 PT 时间可以产生一个延时断开的信号，此定时器只能运行在起动的运行模式。

在输入端 IN 的上升沿可以使输出端 Q 产生一个上升沿。在输入端 IN 的下降沿会使输出端 Q 在经过定时时间 PT 后产生一个下降沿。如果输入端在 PT 时间完成之前变为 1，输出端 Q 保持为 1。输出端 ET 记录了输入端 IN 从最后一个下降沿开始已经过去的时间。ET 端的最大值等于 PT 端输入值，当输入端 IN 变为 1 时，ET 被复位。

SFB 5 “TOF” 遵照 IEC 1131-3 标准。

系统在冷启动时对 SFB 5 “TOF” 进行复位。如果你想在系统暖启动以后对此 SFB 进行初始化，你必须在 OB100 中调用当 PT = 0 时的 SFB5。如果该 SFB 也被其它块调用了，你可以重新设置这些参数，例如，对优先级高的块进行初始化。

### 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	输入端 IN 的下降沿延时时间, PT 必须为正值。(注意: 所取值不能超过 TIME 类型数据所规定的范围)
Q	OUTPUT	BOOL	I, Q, M, D, L	时间状态
ET	OUTPUT	TIME	I, Q, M, D, L	完成时间

## 22.4 用 SFB 0 “CTU” 实现加计数功能

### 描述

你可以利用 SFB 0 “CTU” 加计数。当输入端 CU（与上一次 SFB 调用比较）产生一个上升沿则计数器加 1。如果计数值到达了上限值 32767，就不再增加了。输入端 CU 随后的上升沿将不再起作用。

输入端 R 的 1 信号可以对计数器进行复位，不管此时输入端 CU 为何值。

输出端 Q 表明了是否当前定时器的值大于或者等于预置位 PV。

SFB 0 “CTU” 遵照 IEC 1131-3 标准。

系统在冷启动时对 SFB 0 “CTU” 进行复位。如果你想在系统暖启后对此 SFB 进行初始化，你必须在 OB100 中调用当 R = 1 时的 SFB0。如果该 SFB 也被其它块调用了，你可以重新设置这些参数，例如，对优先级高的块进行初始化。

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	复位端 R 优先于 CU 端。
PV	INPUT	INT	I, Q, M, D, L, 常数	预置值。会影响 Q 的值。
Q	OUTPUT	BOOL	I, Q, M, D, L	计数器状态：Q 有以下几种值 <ul style="list-style-type: none"> <li>• 1, 如果 <math>CV \geq PV</math></li> <li>• 0, 其它</li> </ul>
CV	OUTPUT	INT	I, Q, M, D, L	当前计数值（允许值：0~32767）

## 22.5 用 SFB 1 “CTD” 实现减计数功能

### 描述

你可以利用 SFB 1 “CTD” 实现减计数。当输入端 CD（与上一次 SFB 调用比较）产生一个上升沿则计数器减 1。如果计数值达到了下限值-32768，就不再减了。在输入端 CD 随后产生的任何上升沿都不再起作用。

输入端 LOAD 的 1 信号将计数器设置为预置值 PV 的值，不管此时输入端 CD 为何值。

输出端 Q 表明了是否当前定时器的值小于或者等于 0。

SFB 1 “CTD” 遵照 IEC 1131-3 标准。

系统在冷启动时对 SFB 1 “CTD” 进行复位。如果你想在系统暖启后对此 SFB 进行初始化，你必须在 OB100 中调用当 LOAD = 1 并且 PV = 预置值时的 SFB1。如果该 SFB 也被其它块调用了，你可以重新设置这些值，例如，对优先级高的块进行初始化。

参数	声明	数据类型	存储区域	描述
CD	INPUT	BOOL	I, Q, M, D, L, 常数	计数器输入
LOAD	INPUT	BOOL	I, Q, M, D, L, 常数	装载输入端，LOAD 优先级高于 CD。
PV	INPUT	INT	I, Q, M, D, L, 常数	预置值。当输入端 LOAD 为 1 时，计数器被设置为预置值 PV。
Q	OUTPUT	BOOL	I, Q, M, D, L	计数器状态：Q 有以下几种值： <ul style="list-style-type: none"> <li>• 1, 如果 <math>CV \leq 0</math></li> <li>• 0 其它</li> </ul>
CV	OUTPUT	INT	I, Q, M, D, L	当前值 (允许值：-32768~32767)

## 22.6 用 SFB 2 “CTUD” 实现加/减计数功能

### 描述

你可以利用 SFB 2 “CTUD” 实现加/减计数。当有上升沿时计数值变化，与上一次 SFB 调用进行比较：

- 当 CU 端有上升沿变化时，计数器加 1
- 当 CD 端有上升沿变化时，计数器减 1

如果计数值达到了极限，计数器有以下反应：

- 当达到下限值-32768，计数值不再减小
- 当达到上限值 32767，计数值不再增加。

如果在一个周期中，在 CU 和 CD 端同时产生了一个上升沿，计数器保持当前值。但这不符合 IEC 1131-3 标准，标准规定，如果两个输入端同时有上升沿产生，CU 端的优先级高。这个变化在 IEC 中已提出。

当输入端 LOAD 有 1 信号产生时，计时器被设置为 PV 端定义的预置值，不管此时 CU 和 CD 输入端为何值。

当输入端 R 为 1 时，计时器被置为 0，不管此时 CU，CD 和 LOAD 输入端为何值。输出端 QU 表明了计数器的当前值是否大于或等于预置值 PV；输出端 QD 表明了计数器的当前值是否小于或等于 0。

系统在冷启动时对 SFB 2 “CTUD” 进行复位。如果你想在系统暖启后对 SFB 2 “CTUD” 进行初始化，你必须在 OB100 中在以下情况下调用 SFB 2。

- 当用此块进行加计数时，R = 1
- 当用此块进行减计数时，R = 0，LOAD = 1，PV = 预置值

如果 SFB 被其它的块调用时，你可以重新设置这些参数，例如，对优先级较高的块进行初始化。

参数	声明	数据类型	存储区域	描述
CU	INPUT	BOOL	I, Q, M, D, L, 常数	加计数输入端
CD	INPUT	BOOL	I, Q, M, D, L, 常数	减计数输入端
R	INPUT	BOOL	I, Q, M, D, L, 常数	复位输入端，R 优先于 LOAD。
LOAD	INPUT	BOOL	I, Q, M, D, L, 常数	装载输入端，LOAD 优先于 CU 和 CD。
PV	INPUT	INT	I, Q, M, D, L, 常数	预置值。当输入端 LOAD 为 1 时，计数器被设置为预置值 PV。
QU	OUTPUT	BOOL	I, Q, M, D, L	加计数器状态： QU 有以下几种值 <ul style="list-style-type: none"> <li>• 1，当 <math>CV \geq PV</math></li> <li>• 0，其它</li> </ul>
QD	OUTPUT	BOOL	I, Q, M, D, L	减计数器状态： QD 有以下几种值 <ul style="list-style-type: none"> <li>• 1，当 <math>CV \leq 0</math></li> <li>• 0，其它</li> </ul>
CV	OUTPUT	INT	I, Q, M, D, L	计数器当前值 (允许值：-32768~32767)

## 23 IEC 功能集

### 23.1 概述

用户可将下列国际电工委员会（IEC）功能从 STEP 7 库“S7libs\Stdlib30”中拷贝到用户程序目录中。

名称	IEC 块家族	功能
FC3 D_TOD_DT	转换	将 DATE 和 TIME_OF_DAY 合并为 DT
FC6 DT_DATE	转换	从 DT 中提取 DATE
FC7 DT_DAY	转换	从 DT 中提取星期日期
FC8 DT_TOD	转换	从 DT 中提取 TIME_OF_DAY
FC33 S5TI_TIM	转换	数据类型转换 S5TIME 到 TIME
FC40 TIM_S5TI	转换	数据类型转换 TIME 到 S5TIME
FC16 I_STRNG	转换	数据类型转换 INT 到 STRING
FC5 DI_STRNG	转换	数据类型转换 DINT 到 STRING
FC30 R_STRNG	转换	数据类型转换 REAL 到 STRING
FC38 STRNG_I	转换	数据类型转换 STRING 到 INT
FC37 STRNG_DI	转换	数据类型转换 STRING 到 DINT
FC39 STRNG_R	转换	数据类型转换 STRING 到 REAL
FC9 EQ_DT	DT	DT 相等比较
FC12 GE_DT	DT	DT 大于等于比较
FC14 GT_DT	DT	DT 大于比较
FC18 LE_DT	DT	DT 小于等于比较
FC23 LT_DT	DT	DT 小于比较
FC28 NE_DT	DT	DT 不相等比较
FC10 EQ_STRNG	字符串	字符串相等比较
FC13 GE_STRNG	字符串	字符串大于等于比较
FC15 GT_STRNG	字符串	字符串大于比较
FC19 LE_STRNG	字符串	字符串小于等于比较
FC24 LT_STRNG	字符串	字符串小于比较
FC29 NE_STRNG	字符串	字符串不相等比较
FC21 LEN	字符串	字符串变量的长度
FC20 LEFT	字符串	字符串变量的左部分



名称	IEC 块家族	功能
FC32 RIGHT	字符串	字符串变量的右部分
FC26 MID	字符串	字符串变量的中部
FC2 CONCAT	字符串	合并两个字符串变量
FC17 INSERT	字符串	插入一个字符串变量。
FC4 DELETE	字符串	删除一个字符串变量
FC31 REPLACE	字符串	替换一个字符串变量
FC11 FIND	字符串	查找一个字符串变量
FC1 AD_DT_TM	浮点运算	时间加持续值
FC35 SB_DT_TM	浮点运算	时间减持续值
FC34 SB_DT_DT	浮点运算	两个时间相减
FC22 LIMIT	浮点运算	极限值
FC25 MAX	浮点运算	选取最大值
FC27 MIN	浮点运算	选取最小值
FC36 SEL	浮点运算	开关选取

有关 IEC 相符的通讯块的信息，参见系统功能块（SFB/SFC）帮助内容中的通讯功能（见：*S7 通讯与 S7 基本通讯的区别*）。

## 23.2 IEC 功能的技术数据

### 内存需求

下表中列出了每个 IEC 功能所需的工作存储器，装载存储器的大小以及局部数据的字节数。

FC 块号	名称	存储器（字节数）		局部数据（字节）
		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	L_STRNG	226	340	10
FC5	DI_STRNG	314	440	18
FC30	R_STRNG	528	684	28

FC 块号	名称	存储器（字节数）		局部数据（字节）
		工作存储器	装载存储器	
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	RIGHT	230	350	8
FC26	MID	302	390	8
FC2	CONCAT	358	452	14
FC17	INSERT	488	644	20
FC4	DELETE	376	512	8
FC31	REPLACE	562	726	20
FC11	FIND	236	360	14
FC1	AD_DT_TM	1350	1590	22
FC35	SB_DT_TM	1356	1596	22
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

## 23.3 复杂数据类型日期和时间

### DATE\_AND\_TIME 的实参

DATE\_AND\_TIME 与 ARRAY, STRING 和 STRUCT 都属于复杂数据类型。复杂数据类型允许使用的存储区为数据块 (D) 和局部数据区 (L)。

由于 DATE\_AND\_TIME 是复杂数据类型, 当你在语句中使用 DATE\_AND\_TIME 类型的形参时, 你只能以下列方式之一提供实参。

- 在块声明表中定义一个符号名称
- 定义一个数据单元的符号名称, 例如 “DB\_sys\_info.System\_Time”。由以下两部分构成:
  - 在符号表中定义数据块的名称 (例如 DB5 符号名称为 “DB\_sys\_info”)
  - 在该数据块中为 DATE\_AND\_TIME 变量定义一个符号名称, (例如在 DB5 中定义一个 DATE\_AND\_TIME 类型的变量, 名称为 “System\_Time”)

你不能将一个常数做为实参传递给一个复杂数据类型的形参, 包括 DATE\_AND\_TIME。你也不能将绝对地址做为实参传递给 DATE\_AND\_TIME。

## 23.4 Time-of-Day 功能

### FC1 AD\_DT\_TM 说明

功能 FC1 将一个持续时间 (TIME 格式) 与一个时间 (DT 格式) 相加, 结果产生一个新的时间 (DT 格式)。时间值 (参数 T) 必须在 DT#1990-01-01-00:00:00.000 到 DT#2089-12-31-23:59:59.999 范围内。该功能不进行输入检查。如果相加后的结果不在有效范围内, 则该结果被限制为一个相应的数值, 并且状态字的二进制结果位 (BR) 被置 “0”。

参数	声明	数据类型	存储区域	说明
T	INPUT	DATE_AND_TIME	D, L	DT 格式时间
D	INPUT	TIME	I, Q, M, D, L, 常数	TIME 格式持续时间
RET_VAL	OUTPUT	DATE_AND_TIME	D, L	DT 格式时间和

输入参数 T 与输出参数只能用符号地址赋值。

### FC3 D\_TOD\_DT 说明

功能 FC3 将 DATE 格式的数据与 TIME\_OF\_DAY (TOD) 格式的数据合并。并将格式转换为 DATE\_AND\_TIME 格式 (DT)。IN1 的输入值必须限制在 DATE#1990-01-01 到 DATE#2089-12-31。(该值不被检查) 此功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
IN1	INPUT	DATE	I, Q, M, D, L, 常数	DATE 格式输入变量
IN2	INPUT	TIME_OF_DAY	I, Q, M, D, L, 常数	TOD 格式输入变量
RET_VAL	OUTPUT	DATE_AND_TIME	D, L	DT 格式返回值

返回值只能用符号地址赋值。

#### FC6 DT\_DATE 说明

功能 FC6 从 DATE\_AND\_TIME 格式数据中提取 DATE 格式的数据 DATE 取值必须限制在 DATE#1990-1-1 到 DATE#2089-12-31。该功能不报告任务错误。

参数	声明	数据类型	存储区域	说明
IN	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
RET_VAL	OUTPUT	DATE	I, Q, M, D, L	DATE 格式返回值

输入参数只能用符号地址赋值。

#### FC7 DT\_DAY 说明

功能 FC7 从 DATE\_AND\_TIME 格式数据中提取星期日期。

星期日期用 INTEGER 格式表达：

- 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

该功能不报告任务错误：

参数	声明	数据类型	存储区域	说明
IN	INPUT	DATE_AND_TIME	D	DT 格式输入变量
RET_VAL	OUTPUT	INT	I, Q, M, D	INT 格式返回值

输入参数只能用符号地址赋值。

#### FC8 DT\_TOD 说明

功能 FC8 从 DATE\_AND\_TIME 格式数据中提取 DATE\_AND\_TIME 数据。该功能不报告任何错误：

参数	声明	数据类型	存储区域	说明
IN	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
RET_VAL	OUTPUT	TIME_OF_DAY	I, Q, M, D, L	TOD 格式返回值

输入参数只能用符号地址赋值。

## FC33 S5TI\_TIM 说明

功能 FC33 将数据从 S5TIME 格式转换为 TIME 格式。如果转换的结果超出了 TIME 表达范围，则该结果被限制为一个相应的数值，并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
IN	INPUT	S5TIME	I, Q, M, D, L, 常数	S5TIME 格式输入变量
RET_VAL	OUTPUT	TIME	I, Q, M, D, L	TIME 格式返回值

## FC34 SB\_DT\_DT 说明

功能 FC34 将两个时间相减（DT 格式），结果产生一个持续时间（TIME 格式）。时间值必须在 DT#1990-01-01-00:00:00.000 到 DT#2089-12-31-23:59:59.999 范围内。该功能不进行输入检查。如果第一个时间（参数 DT1）大于（近期）第二个时间（参数 DT2），则结果为正；如果第一个时间小于（远期）第二个时间，则结果为负。如果相减后的结果超出了 TIME 表达范围，则该结果被限制为一个相应的数值，并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
DT1	INPUT	DATE_AND_TIME	D, L	DT 格式第一个时间
DT2	INPUT	DATE_AND_TIME	D, L	DT 格式第二个时间
RET_VAL	OUTPUT	TIME	I, Q, M, D, L	TIME 格式时间差

输入参数只能用符号地址赋值。

## FC35 SB\_DT\_TM 说明

功能 FC35 将一个时间（DT 格式）与一个持续时间（TIME 格式）相减，结果产生一个新的时间（DT 格式）。时间值（参数 T）必须在 DT#1990-01-01-00:00:00.000 到 DT#2089-12-31-23:59:59.999 范围内。该功能不进行输入检查。如果相减后的结果不在有效范围内，则该结果被限制为一个相应的数值，并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
T	INPUT	DATE_AND_TIME	D, L	DT 格式时间
D	INPUT	TIME	I, Q, M, D, L, 常数	TIME 格式持续时间
RET_VAL	OUTPUT	DATE_AND_TIME	D, L	DT 格式时间差

输入参数 T 与输出参数只能用符号地址赋值。

## 23.5 DATE\_AND\_TIME 变量比较

### FC9 EQ\_DT 说明

功能 FC9 将两个 DATE\_AND\_TIME 数据类型的变量内容进行等于比较，并将比较结果作为返回值输出。如果参数 DT1 与参数 DT2 数值相等，则返回值信号状态为“1”。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
DT1	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
DT2	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

### FC12 GE\_DT 说明

功能 FC12 将两个 DATE\_AND\_TIME 数据类型的变量内容进行大于等于比较，并将比较结果作为返回值输出。如果参数 DT1 大于（近期）或等于参数 DT2，则返回值信号状态为“1”。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
DT1	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
DT2	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

### FC14 GT\_DT 说明

功能 FC14 将两个 DATE\_AND\_TIME 数据类型的变量内容进行大于比较，并将比较结果作为返回值输出。如果参数 DT1 大于（近期）参数 DT2，则返回值信号状态为“1”。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
DT1	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
DT2	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

### FC18 LE\_DT 说明

功能 FC18 将两个 DATE\_AND\_TIME 数据类型的变量内容进行小于等于比较，并将比较结果作为返回值输出。如果参数 DT1 小于（远期）或等于参数 DT2，则返回值信号状态为“1”。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
DT1	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
DT2	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

#### FC23 LT\_DT 说明

功能 FC23 将两个 DATE\_AND\_TIME 数据类型的变量内容进行小于比较，并将比较结果作为返回值输出。如果参数 DT1 小于（远期）参数 DT2，则返回值信号状态为“1”。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
DT1	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
DT2	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

#### FC28 NE\_DT 说明

功能 FC28 将两个 DATE\_AND\_TIME 数据类型的变量内容进行不等于比较，并将比较结果作为返回值输出。如果参数 DT1 不等于参数 DT2，则返回值信号状态为“1”。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
DT1	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
DT2	INPUT	DATE_AND_TIME	D, L	DT 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

## 23.6 STRING 变量比较

### FC10 EQ\_STRNG 说明

功能 FC10 将两个 STRING 数据类型的变量内容进行等于比较，并将比较结果作为返回值输出。如果参数 S1 与参数 S2 相等，则返回值信号状态为“1”。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
S1	INPUT	STRING	D, L	以 STRING 格式输入变量
S2	INPUT	STRING	D, L	以 STRING 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

### FC13 GE\_STRNG 说明

功能 FC13 将两个 STRING 数据类型的变量内容进行大于等于比较，并将比较结果作为返回值输出。如果参数 S1 大于或等于参数 S2，则返回值信号状态为“1”。字符通过它们的 ASCII 码进行比较（例如：“a”大于“A”），从左边开始。第一位的不同就可以产生比较的结果。如果第一位字符相同，字符串长的比较大。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
S1	INPUT	STRING	D, L	以 STRING 格式输入变量
S2	INPUT	STRING	D, L	以 STRING 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

### FC15 GT\_STRNG 说明

功能 FC15 将两个 STRING 数据类型的变量内容进行大于比较，并将比较结果作为返回值输出。如果参数 S1 大于参数 S2，则返回值信号状态为“1”。

字符通过它们的 ASCII 码进行比较（例如：“a”大于“A”），从左边开始。第一位的不同就可以产生比较的结果。如果第一位字符相同，字符串长的比较大。

该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
S1	INPUT	STRING	D, L	STRING 格式输入变量
S2	INPUT	STRING	D, L	STRING 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

### FC19 LE\_STRNG 说明

功能 FC19 将两个 STRING 数据类型的变量内容进行小于等于比较，并将比较结果作为返回值输出。如果参数 S1 小于或等于参数 S2，则返回值信号状态为“1”。



字符通过它们的 ASCII 码进行比较（例如：“a”大于“A”），从左边开始。第一位的不同就可以产生比较的结果。如果第一位字符相同，字符串短的比较小。

该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
S1	INPUT	STRING	D, L	以 STRING 格式输入变量
S2	INPUT	STRING	D, L	以 STRING 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

#### FC24 LT\_STRNG 说明

功能 FC24 将两个 STRING 数据类型的变量内容进行小于比较，并将比较结果作为返回值输出。如果参数 S1 小于参数 S2，则返回值信号状态为“1”。

字符通过它们的 ASCII 码进行比较（例如：“a”大于“A”），从左边开始。第一位的不同就可以产生比较的结果。如果第一位字符相同，字符串短的比较小。

该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
S1	INPUT	STRING	D, L	以 STRING 格式输入变量
S2	INPUT	STRING	D, L	以 STRING 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址赋值。

#### FC29 NE\_STRNG 说明

功能 FC29 将两个 STRING 数据类型的变量内容进行不等于比较，并将比较结果作为返回值输出。如果参数 S1 不等于参数 S2，则返回值信号状态为“1”。

该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
S1	INPUT	STRING	D, L	以 STRING 格式输入变量
S2	INPUT	STRING	D, L	以 STRING 格式输入变量
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	比较结果

输入参数只能用符号地址。

## 23.7 数值编辑

### FC22 LIMIT 说明

功能 FC22 将变量的数值限制在由参数给定的极限内。变量的数据类型可以是 INT, DINT, 和 REAL。参数中所有变量的数据类型必须一致。变量的类型由 ANY 指针指定。下限值（参数 MN）必须小于或等于上限值（参数 MX）。

如果下列情况之一发生，则输出保持不变状态字的二进制结果位（BR）被置“0”：

- 变量数据类型无效
- 变量数据类型不一致
- 下限值大于上限值
- REAL 变量不是有效浮点数

参数	声明	数据类型	存储区域	说明
MN	INPUT	ANY	I, Q, M, D, L	下限值
IN	INPUT	ANY	I, Q, M, D, L	输入变量
MX	INPUT	ANY	I, Q, M, D, L	上限值
RET_VAL	OUTPUT	ANY	I, Q, M, D, L	限幅输出变量

### FC25 MAX 说明

功能 FC25 在三个变量中选取最大值。变量的数据类型可以是 INT, DINT, 和 REAL。参数中所有变量的数据类型必须一致。变量的类型由 ANY 指针指定。

如果下列情况之一发生，则输出保持不变状态字的二进制结果位（BR）被置“0”：

- 变量数据类型无效
- 变量数据类型不一致
- REAL 变量不是有效浮点数

参数	声明	数据类型	存储区域	说明
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	最大输入值

## 23.8 STL 举例

```

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：

允许的数据类型 INT, DINT, 和 REAL 必须在 ANY 指针中指定。象“MD20”这样的参数也是允许的，但是必须在符号表中定义“MD20”相应的数据类型。

### FC27 MIN 说明

功能 FC27 在三个变量中选取最小值。变量的数据类型可以是 INT, DINT, 和 REAL。参数中所有变量的数据类型必须一致。变量的类型由 ANY 指针指定。

如果下列情况之一发生，则输出保持不变状态字的二进制结果位（BR）被置“0”：

- 变量数据类型无效
- 变量数据类型不一致
- REAL 变量不是有效浮点数

参数	声明	数据类型	存储区域	说明
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	最小输入值

## 23.9 STL 举例

```
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:

允许的数据类型 INT, DINT, 和 REAL 必须在 ANY 指针中指定。象“MD20”这样的参数也是允许的, 但是必须在符号表中定义“MD20”相应的数据类型。

### FC36 SEL 说明

功能 FC36 根据开关 (参数 G) 在两个变量值中做选择。参数 IN0 和 IN1 的数据类型可以是对应位、字节、字、双字长度的任何数据类型 (非 DT 和 STRING 类型), 但输入变量与输出变量数据类型必须一致。

如果下列情况之一发生, 则输出保持不变状态字的二进制结果位 (BR) 被置“0”:

- 变量数据类型无效
- 变量数据类型不一致
- REAL 变量不是有效浮点数

参数	声明	数据类型	存储区域	说明
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	被选取的输入值

## 23.10 STRING 变量编辑

### FC2 CONCAT 说明

功能 FC2 将两个 STRING 变量串接成一个字符串。如果该字符串的长度超过输出参数规定的长度，则被限制在最大给定长度并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
IN1	INPUT	STRING	D, L	STRING 格式输入变量
IN2	INPUT	STRING	D, L	STRING 格式输入变量
RET_VAL	OUTPUT	STRING	D, L	合并结果字符串

所有参数只能用符号地址赋值。

### FC4 DELETE 说明

功能 FC4 将一个字符串从字符位置 P 开始（包含）删除若干字符（L）。如果 L 和/或 P 等于 0，或者 P 大于当前输入字符串的长度，则该输入字符串被返回。如果 L 与 P 的和大于输入字符串的长度，则字符串被删至最后一个字符。如果 L 和/或 P 为负值，返回一个空字符串并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
IN	INPUT	STRING	D, L	待删除的 STRING 变量
L	INPUT	INT	I, Q, M, D, L, 常数	删除字符个数
P	INPUT	INT	I, Q, M, D, L, Const.	第一个被删除字符的位置
RET_VAL	OUTPUT	STRING	D, L	结果字符串

输入参数 IN 与输出参数只能用符号地址赋值。

### FC11 FIND 说明

功能 FC11 提供第二个字符串（IN2）在第一个字符串（IN1）中的位置。搜索从左开始；字符串首次出现的位置被报告。如果没有在第一个字符串中找到第二个字符串，则返回值 为 0。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
IN1	INPUT	STRING	D, L	被搜索的 STRING 变量
IN2	INPUT	STRING	D, L	待查找的 STRING 变量
RET_VAL	OUTPUT	STRING	I, Q, M, D, L	找到字符串的位置

输入参数 IN1 和 IN2 只能用符号地址赋值。

**FC17 INSERT 说明**

功能 FC17 在参数 IN1 字符串中从字符位置 P 后插入参数 IN2 字符串。如果 P 等于 0，则第二个字符串在第一个字符串前插入，如果 P 大于第一个字符串的当前长度，则第二个字符串在第一个字符串后插入。如果 P 为负值，则返回一个空字符串，并且状态字的二进制结果位（BR）被置“0”。当结果字符串的长度超过输出参数规定的长度，二进制结果位也被置“0”，在这种情况下结果字符串被限制在最大给定长度。

参数	声明	数据类型	存储区域	说明
IN1	INPUT	STRING	D, L	被插入的 STRING 变量
IN2	INPUT	STRING	D, L	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

输入参数 IN1 和 IN2 以及输出参数只能用符号地址赋值。

**FC20 LEFT 说明**

功能 FC20 提供一个字符串的左边 L 个字符（L 代表一个数字）。如果 L 大于 STRING 变量的当前长度，则输入值被返回。如果 L = 0 或输入值为一个空字符串，则返回一个空字符串。如果 L 为负值，则返回一个空字符串，并且二进制结果位（BR）“0”。

参数	声明	数据类型	存储区域	说明
IN	INPUT	STRING	D, L	以 STRING 格式输入变量
L	INPUT	INT	I, Q, M, D, L, 常数	左边字符串长度
RET_VAL	OUTPUT	STRING	D, L	以 STRING 格式输出变量

输入参数 IN 和返回值只能用符号地址赋值。

**FC21 LEN 说明**

一个 STRING 变量包含两个长度：最大长度（当定义变量时在方括号中给出）和当前长度（当前的有效字符数）。当前长度必须小于或等于最大长度。字符串所占字节数比最大长度大 2。

功能 FC21 输出字符串的当前长度作为返回值。一个空字符串（"）长度为 0。字符串最大长度为 254。该功能不报告任何错误。

参数	声明	数据类型	存储区域	说明
S	INPUT	STRING	D, L	以 STRING 格式输入变量
RET_VAL	OUTPUT	INT	I, Q, M, D, L, 常数	有效字符数

输入参数只能用符号地址赋值。

**FC26 MID 说明**

FC26 提供一个字符串的中间部分（从字符位置 P 开始，包含 P 的 L 个字符）。如果 L 和 P 的和大于 STRING 变量的当前长度，则返回的字符串从输入值的 P 处开始直到末尾。其它情况（P 超出当前长度，P 和/或 L 等于零或负值）则返回一个空字符串并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
IN	INPUT	STRING	D, L	以 STRING 格式输入变量
L	INPUT	INT	I, Q, M, D, L, 常数	中间字符串长度
P	INPUT	INT	I, Q, M, D, L, 常数	第一个字符位置
RET_VAL	OUTPUT	STRING	D, L	以 STRING 格式输出变量

输入参数 IN 和返回值只能用符号地址赋值。

### FC31 REPLACE 说明

功能 FC31 用第二个字符串 (IN2) 替换第一个字符串中从字符位置 P 开始 (包含) 的若干字符 (L)。如果 L=0, 则第一个字符串被返回。如果 P 等于零或 1, 字符串从第一个字符 (包含) 开始被替换。如果 P 超出第一个字符串, 第二个字符串加在第一个字符串后面。如果 L 和/或 P 为负值, 则返回一个空字符串并且状态字的二进制结果位 (BR) 被置“0”。当结果字符串的长度超过输出参数规定的长度, 二进制结果位也被置“0”, 在这种情况下结果字符串被限制在最大给定长度。

参数	声明	数据类型	存储区域	说明
IN1	INPUT	STRING	D, L	被替换的 STRING 变量
IN2	INPUT	STRING	D, L	插入的 STRING 变量
L	INPUT	INT	I, Q, M, D, L, 常数	被替换的字符个数
P	INPUT	INT	I, Q, M, D, L, 常数	第一个被替换的字符位置
RET_VAL	OUTPUT	STRING	D, L	结果字符串

输入参数 IN1 和 IN2 以及输出参数只能用符号地址赋值。

### FC32 RIGHT 说明

功能 FC32 提供一个字符串的右边 L 个字符 (L 代表一个数字)。如果 L 大于 STRING 变量的当前长度, 则输入值被返回。如果 L = 0 或输入值为一个空字符串, 则返回一个空字符串。如果 L 为负值, 则返回一个空字符串, 并且二进制结果位 (BR) 被置“0”。

参数	声明	数据类型	存储区域	说明
IN	INPUT	STRING	D, L	以 STRING 格式输入变量
L	INPUT	INT	I, Q, M, D, L, 常数	右边字符串长度
RET_VAL	OUTPUT	STRING	D, L	以 STRING 格式输出变量

输入参数 IN 和返回值只能用符号地址赋值。

## 23.11 数据类型格式转换

### FC5 DI\_STRNG 说明

功能 FC5 将 DINT 数据类型格式的变量转换为字符串。该字符串从符号位开始。如果赋予返回参数的变量太短，则不进行转换，并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
I	INPUT	DINT	I, Q, M, D, L, 常数	输入值
RET_VAL	OUTPUT	STRING	D, L	结果字符串

输出参数只能用符号地址赋值。

### FC16 I\_STRNG 说明

功能 FC16 将 INT 数据类型格式的变量转换为字符串。该字符串从符号位开始。如果赋予返回参数的变量太短，则不进行转换，并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
I	INPUT	INT	I, Q, M, D, L, 常数	输入值
RET_VAL	OUTPUT	STRING	D, L	结果字符串

输出参数只能用符号地址赋值。

### FC30 R\_STRNG 说明

功能 FC30 将 REAL 数据类型格式的变量转换为字符串。该字符串表达 14 个数字：

±v.nnnnnnnE±xx	±	符号
	v	小数点前 1 位
	n	小数点后 7 位
	x	指数 2 位

如果赋予返回参数的变量太短或参数 IN 未被赋予一个有效浮点数，则不进行转换，并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
IN	INPUT	REAL	I, Q, M, D, L, 常数	输入值
RET_VAL	OUTPUT	STRING	D, L	结果字符串

输出参数只能用符号地址赋值。

### FC37 STRNG\_DI 说明

功能 FC37 将一个字符串转换为 DINT 数据类型格式的变量。字符串的第一个字符可以是符号或数字，以后的必须是数字。如果字符串的长度等于 0 或大于 11，或者字符串中存在无效字符，则不进行转换，并且状态字的二进制结果位（BR）被置“0”。如果转换后的结果超出 DINT 的范围，则结果被限制为一个相应的数值，并且状态字的二进制结果位（BR）被置“0”。



参数	声明	数据类型	存储区域	说明
S	INPUT	STRING	D, L	输入字符串
RET_VAL	OUTPUT	D INT	I, Q, M, D, L	结果

输入参数只能用符号地址赋值。

#### FC38 STRNG\_I 说明

功能 FC38 将一个字符串转换为 INT 数据类型格式的变量。字符串的第一个字符可以是符号或数字，以后的必须是数字。如果字符串的长度等于 0 或大于 6，或者字符串中存在无效字符，则不进行转换，并且状态字的二进制结果位（BR）被置“0”。如果转换后的结果超出 INT 的范围，则结果被限制为一个相应的数值，并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
S	INPUT	STRING	D, L	输入字符串
RET_VAL	OUTPUT	INT	I, Q, M, D, L	结果

输入参数只能用符号地址赋值。

#### FC39 STRNG\_R 说明

功能 FC39 将一个字符串转换为 REAL 数据类型格式的变量。字符串必须具有下列格式：

$\pm v.nnnnnnnE\pm xx$	$\pm$	符号
	v	小数点前 1 位
	n	小数点后 7 位
	x	指数 2 位

如果字符串的长度小于 14，或者字符串结构与上面不符，则不进行转换，并且状态字的二进制结果位（BR）被置“0”。如果转换后的结果超出 REAL 的范围，则结果被限制为一个相应的数值，并且状态字的二进制结果位（BR）被置“0”。

参数	声明	数据类型	存储区域	说明
S	INPUT	STRING	D, L	输入字符串
RET_VAL	OUTPUT	REAL	I, Q, M, D, L	结果

输入参数只能用符号地址赋值。

## 24 用于集成控制功能的 SFB

### 24.1 连续调节功能 SFB 41/FB 41 “CONT\_C”

#### 介绍

SFB/FB “CONT\_C”（连续调节器）用在 SIMATIC S7 可编程逻辑控制器上，用于调节带有连续输入和输出变量的技术过程。赋参数时可以激活或取消 PID 调节器的子功能，以使调节器与过程匹配。赋参数可以简单地通过参数赋值工具进行（菜单路径：Start > Simatic > Step7 > Assign PID Control parameters）。在线电子手册位于：Start > Simatic > Step7 > Assign PID Control English。

#### 应用

该调节器可以作为固定设定值 PID 调节器，或者在多闭环控制中作为串级调节器，混合调节器或比例调节器。调节器的功能基于带模拟信号的采样调节器的 PID 控制算法，如果需要，还可以扩展出一个脉冲发生器级，以产生脉宽调制的输出信号，用于两个或三个带比例执行器的步进调节器。

#### 说明

除了设定值通道和过程数据通道的功能外，SFB/FB 能实现带连续被控量输出的完整的 PID 调节器，并可选择手动控制被控量，下面是这些子功能的详细说明：

##### 设定值通道

设定值以浮点数格式在 SP\_INT 端输入。

##### 过程变量通道

过程变量可以以外设（I/O）或浮点数格式输入。CRP\_IN 功能按下面公式将 PV\_PER 外设值转换成 -100% 到 +100 %之间的浮点数格式：

$$\text{CPR\_IN 的输出} = \text{PV\_PER} * \frac{100}{2764}$$

PV\_NORM 功能按下面公式规范 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 缺省值为 1，PV\_OFF 缺省值为 0。

##### 误差信号

设定值与过程变量之间的差值叫误差信号。为了抑制由于被控量量子化引起的小的、恒定的振荡（例如，用 PULSEGEN 进行脉宽调制时），为误差信号设置了一个死区（DEADBAND）。若 DEADB\_W = 0，则死区被关闭。

##### PID 算法

PID 算法作为位置算法运行。比例，积分（INT）和微分（DIF）分量并联连接，可以分别激活和取消。这就允许组成成 P，PI，PD，和 PID 调节器。也可以是纯 I 和 D 调节器。

### 手动值

可以在手动模式和自动模式之间切换。在手动模式下，被控量被修改成手动选定的值。积分器（INT）被内部设置成 LMN - LMN\_P - DISV，微分单元（DIF）设置成 0，并内部匹配。这就是说切换到自动模式时不会引起被控量的突变。

### 被控量

被控量可以被限制在一个用 LMNLIMIT 功能选定的值。如果输入变量超出限幅值，信号位指示。LMN\_NORM 功能按下面公式规范 LMNLIMIT 的输出：

$$LMN = (\text{LMNLIMIT 的输出}) * LMN\_FAC + LMN\_OFF$$

LMN\_FAC 缺省值为 1，LMN\_OFF 缺省值为 0。

被控量也可以是外设格式。CPR\_OUT 功能按下面公式将浮点数 LMN 转换成外设格式：

$$LMN\_PER = LMN = \frac{100}{2764}$$

### 前馈控制

干扰量可以在 DISV 端输入。

### 初始化

SFB 41 “CONT\_C” 有一个初始化过程，当输入参数 COM\_RST = TRUE 时执行该过程。

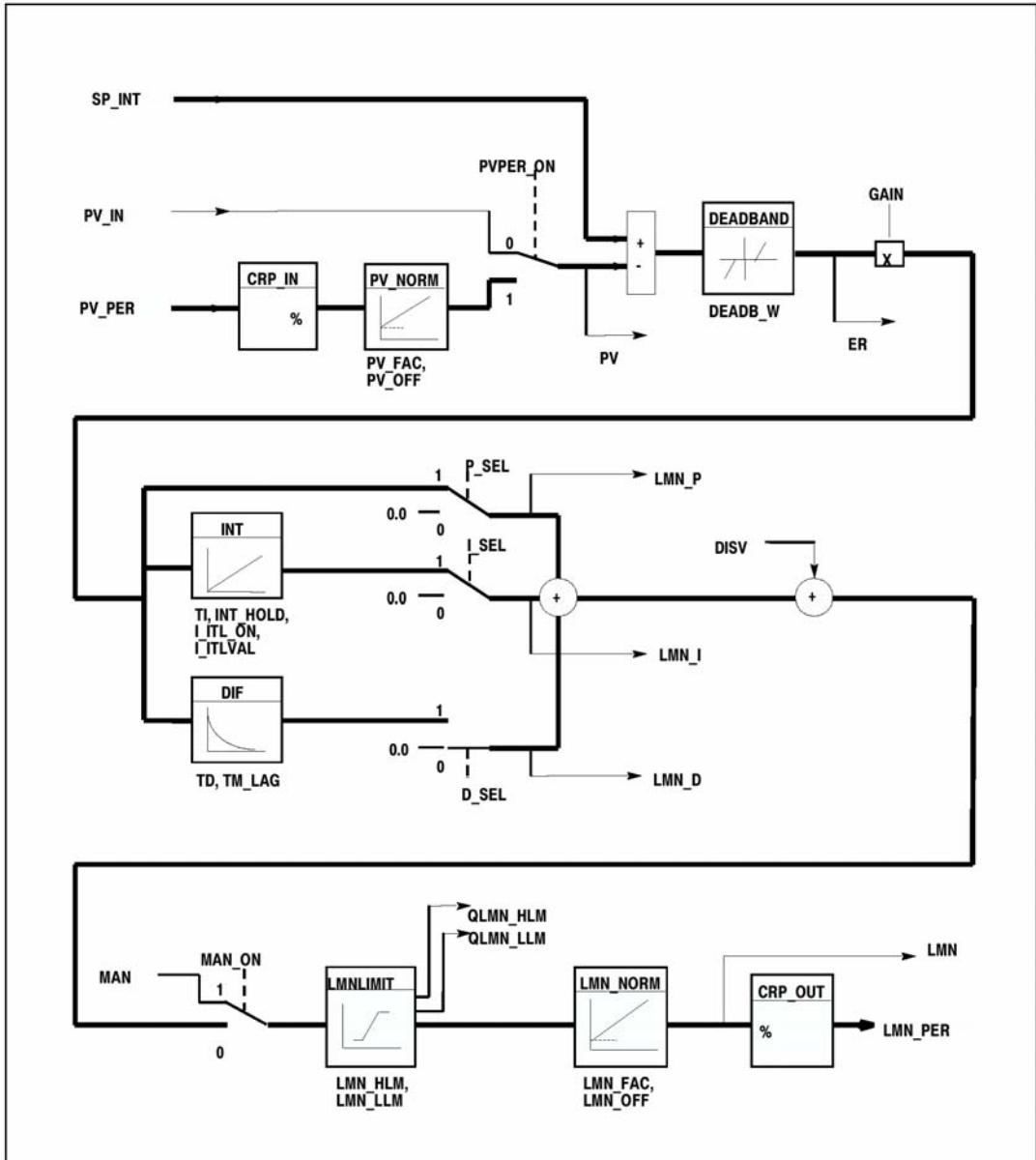
在初始化过程中，积分器被内部置成初始化值 I\_ITVAL。如果是在循环中断优先级中调用的，那么它从该值开始继续工作。

所有其它输出端被置成缺省值。

### 故障信息

故障输出参数 RET\_VAL 不使用。

CONT\_C 框图



## 输入参数

下表列出 SFB 41/FB 41 “CONT\_C” 输入参数的说明。

参数	数据类型	数值范围	缺省	说明
COM_RST	BOOL		FALSE	COMPLETE RESTART（完全再启动） 当输入端 COM_RST 置位时，该块执行初始化过程。
MAN_ON	BOOL		TRUE	MANUAL VALUE ON（手动数值接通） 如果输入端“手动数值接通”置位，那么闭环控制中断。手动值被置成被控值。
PVPER_ON	BOOL		FALSE	PROCESS VARIABLE PERIPHERAL ON（过程变量外设接通）如果过程变量从 I/O 读入，那么输入端 PV_PER 必须接到 I/O 上且输入端“过程变量外设接通”必须被置位。
P_SEL	BOOL		TRUE	PROPORTIONAL ACTION ON（比例分量接通） PID 各分量在 PID 算法中可以分别激活和取消。当输入端“比例分量接通”被置位时，P 分量被接通。
I_SEL	BOOL		TRUE	INTEGRAL ACTION ON（积分分量接通） PID 各部分在 PID 算法中可以分别激活和取消。当输入端“积分分量接通”被置位时，I 分量被接通。
INT_HOLD	BOOL		FALSE	INTEGRAL ACTION HOLD（积分分量保持） 积分器的输出可以通过设置“积分分量保持”被“冻结”。
I_ITL_ON	BOOL		FALSE	INITIALIZATION OF THE INTEGRAL（积分分量初始化接通） 积分器的输出可以通过设置输入端“积分分量初始化接通”连接到输入端 I_ITL_VAL 上。
D_SEL	BOOL		FALSE	DERIVATIVE ACTION ON（微分分量接通） PID 各分量在 PID 算法中可以分别激活和取消。当输入端“微分分量接通”被置位时，D 分量被接通。

参数	数据类型	数值范围	缺省	说明
CYCLE	TIME	$\geq 1$ ms	T#1s	SAMPLING TIME (采样时间) 块调用之间的时间必须恒定。“采样时间”输入端确定块调用之间的时间。
SP_INT	REAL	-100.0 至+100.0(%)或物理量 1)	0.0	INTERNAL SETPOINT (内部设定值) “内部设定值”输入端用于确定设定值。
PV_IN	REAL	-100.0 至+100.0 (%)或物理量 1)	0.0	PROCESS VARIABLE IN (过程变量输入) 可以设置一个初始值到“过程变量输入”输入端或者连接一个浮点数格式的外部过程变量。
PV_PER	WORD		W#16#0000	PROCESS VARIABLE PERIPHERAL (过程变量外设) I/O 格式的过程变量通过“过程变量外设”输入端连接到调节器上。
MAN	REAL	-100.0 至+100.0 (%)或物理量 2)	0.0	MANUAL VALUE (手动值) “手动值”输入端用于通过操作员接口功能设置一个手动值。
GAIN	REAL		2.0	PROPORTIONAL GAIN (比例增益) “比例增益”输入端确定调节器比例系数。
TI	TIME	$\geq$ CYCLE	T#20s	RESET TIME (复位时间) “复位时间”输入端确定积分器的时间响应。
TD	TIME	$\geq$ CYCLE	T#10s	DERIVATIVE TIME (微分时间) “复位时间”输入端确定微分单元的时间响应。
TM_LAG	TIME	$\geq$ CYCLE/2	T#2s	TIME LAG OF THE DERIVATIVE ACTION (微分分量的滞后时间) D 分量算法包括一个滞后时间,可以赋到“微分分量滞后时间”输入端上。
DEADB_W	REAL	$\geq 0.0$ (%)或物理量	0.0	DEAD BAND WIDTH (死区宽度) 死区用于误差。“死区宽度”输入端确定死区的大小。
LMN_HLM	REAL	LMN_LLM ...100.0 (%)或物理量 2)	100.0	MANIPULATED VALUE HIGH LIMIT (被控量上限) 被控量总是被上、下限限制。“被控量上限”输入端确定上限值。

参数	数据类型	数值范围	缺省	说明
LMN_LLM	REAL	-100.0...LMN_HLM(% 或物理量 2)	0.0	MANIPULATED VALUE LOW LIMIT (被控量下限) 被控量总是被上、下限限制。“被控 量下限”输入端确定下限值。
PV_FAC	REAL		1.0	PROCESS VARIABLE FACTOR (过 程变量系数) “过程变量系统”输入端与过程变量 相乘。该输入端用于匹配过程变量的 范围。
PV_OFF	REAL		0.0	PROCESS VARIABLE OFFSET (过 程变量偏移量) “过程变量偏移量”输入端与过程变 量相加。该输入端用于匹配过程变量 的范围。
LMN_FAC	REAL		1.0	MANIPULATED VALUE FACTOR (被控量系数) “被控量系数”输入端与被控量相 乘。该输入端用于匹配被控量范围。
LMN_OFF	REAL		0.0	MANIPULATED VALUE OFFSET (被控量偏移量) “被控量偏移量”与被控量相加。该 输入端用于匹配被控量范围。
I_ITLVAL	REAL	-100.0 至+100.0 (%)或 物理量 2)	0.0	INITIALIZATION VALUE OF THE INTEGRAL ACTION (积分分量初始 值) 积分器的输出可以用输入端 I_ITL_ON 设置。初始值被设成“积 分分量初始值”。
DISV	REAL	-100.0 至+100.0 (%)或 物理量 2)	0.0	DISTURBANCE VARIABLE (干扰变 量) 对于前馈控制, 干扰变量连接到输入 端“干扰变量”。

1) 设定值通道和过程变量通道中的参数有相同的单位。

2) 被控量通道中的参数有相同的单位。

## 输出参数

下表列出 SFB 41/FB41 “CONT\_C” 输出参数的说明。

参数	数据类型	数值范围	缺省	说明
LMN	REAL		0.0	MANIPULATED VALUE (被控量) 有效的被控量以浮点数格式在“被控量”输出端上输出。
LMN_PER	WORD		W#16#0000	MANIPULATED VALUE PERIPHERAL (被控量外设) I/O 格式的被控量在“被控量外设”输出端上输出。
QLMN_HLM	BOOL		FALSE	HIGH LIMIT OF MANIPULATED VALUE REACHED (被控量上限值到达) 被控量总是被限制在上、下限之间,“被控量上限值到达”表示上限值已被超出。
QLMN_LLM	BOOL		FALSE	LOW LIMIT OF MANIPULATED VALUE REACHED (被控量下限值到达) 被控量总是被限制在上、下限之间。“被控量下限值到达”表示下限值已被超出。
LMN_P	REAL		0.0	PROPORTIONAL COMPONENT (比例分量) “比例分量”输出端输出被控量的比例分量。
LMN_I	REAL		0.0	INTEGRAL COMPONENT (积分分量) “积分分量”输出端输出被控量的积分分量。
LMN_D	REAL		0.0	DERIVATIVE COMPONENT (微分分量) “微分分量”输出端输出被控量的微分分量。
PV	REAL		0.0	PROCESS VARIABLE (过程变量) 有效的过程变量在“过程变量”输出端上输出。
ER	REAL		0.0	ERROR SIGNAL (误差信号) 有效的误差在“误差信号”输出端输出。



## 24.2 步进调节功能 SFB 42/FB 42 “CONT\_S”

### 介绍

SFB/FB “CONT\_S”（步进调节器）用在 SIMATIC S7 可编程逻辑控制器上，用于调节带有数字被控量输出的技术过程。赋参数时可以激活或取消 PI 步进调节器的子功能，以使调节器与过程匹配。赋参数可以简单地通过参数赋值工具进行（菜单路径：**Start > Simatic > Step7 > Assign PID Control parameters**）。在线电子手册位于：**Start > Simatic > Step7 > Assign PID Control English**。

### 应用

该调节器可以作为固定设定值 PI 调节器，或者在第二级闭环中作为串级调节器，混合调节器或比例调节器。但不能作为第一级调节器。调节器的功能基于采样调节器的 PI 控制算法，外加由模拟执行信号生成开关量输出信号的功能。

下列功能适合 CPU 314 IFM 的 FB V1.5 或 V1.1.0 以上版本：

利用  $T_I = T\#0ms$ ，可以封锁调节器的积分分量，因此允许功能块用作比例调节器。

由于调节器没有位置反馈信号，因此内部计算的被控量不会精确适合信号控制基本位置。如果被控量（ $ER * GAIN$ ）为负要进行调整。然后调节器置位输出端 QLMNDN（被控量信号低），直到 LMNR\_LS（位置反馈信号下限）被置位。

调节器也可以用作调节器级联中的第二个执行器。设定值输入端 SP\_INT 用于设置控制基本位置。在这种情况下实际值输入和参数  $T_I$ （积分时间）必须置成 0。一个应用实例是借助脉 - 空控制来控制热量输出的温度调节和利用阀门控制制冷容量。在这种情况下，为了完全关闭阀门，被控量（ $ER * GAIN$ ）应该有一个负值。

### 说明

除了过程数据通道的功能外，SFB 能实现带数字被控量输出的完整的 PI 调节器，并可选择手动控制被控量。步进调节器的运行不带位置反馈信号。下面是部分功能的说明：

#### 设定值通道

设定值以浮点数格式在 SP\_INT 输入端上输入。

#### 过程变量通道

过程变量可以以外设（I/O）或浮点数格式输入。CRP\_IN 功能按下面公式将 PV\_PER 外设定值转换成 -100% 到 +100 % 之间的浮点数格式：

$$CPR\_IN \text{ 的输出} = PV\_PER = \frac{100}{27648}$$

PV\_NORM 功能按下面公式规范 CRP\_IN 的输出：

$$PV\_NORM \text{ 的输出} = (CPR\_IN \text{ 的输出}) * PV\_FAC + PV\_OFF$$

PV\_FAC 缺省值为 1，PV\_OFF 缺省值为 0。

#### 误差信号

设定值与过程变量之间的差值叫误差信号。为了抑制由于被控量量子化引起的小的、恒定的振荡（例如，由于执行器阀的被控量精度有限），为误差信号设置了一个死区（DEADBAND）。若 DEADB\_W = 0，则死区被关闭。

### PI 步进算法

SFB/FB 没有位置反馈信号。PI 算法的 I 分量和假设的位置反馈信号在一个积分器 (INT) 中计算，并作为反馈值与剩余的 P 分量比较。其差值送给一个三级单元 (THREE\_ST) 和一个脉冲发生器 (PULSEOUT)，再电脉冲发生器产生用于执行器的脉冲。调节器的开关频率可以通过使其阈值匹配三级单元的阈值而降低。

### 前馈控制

干扰变量可以前馈到 DISV 输入端。

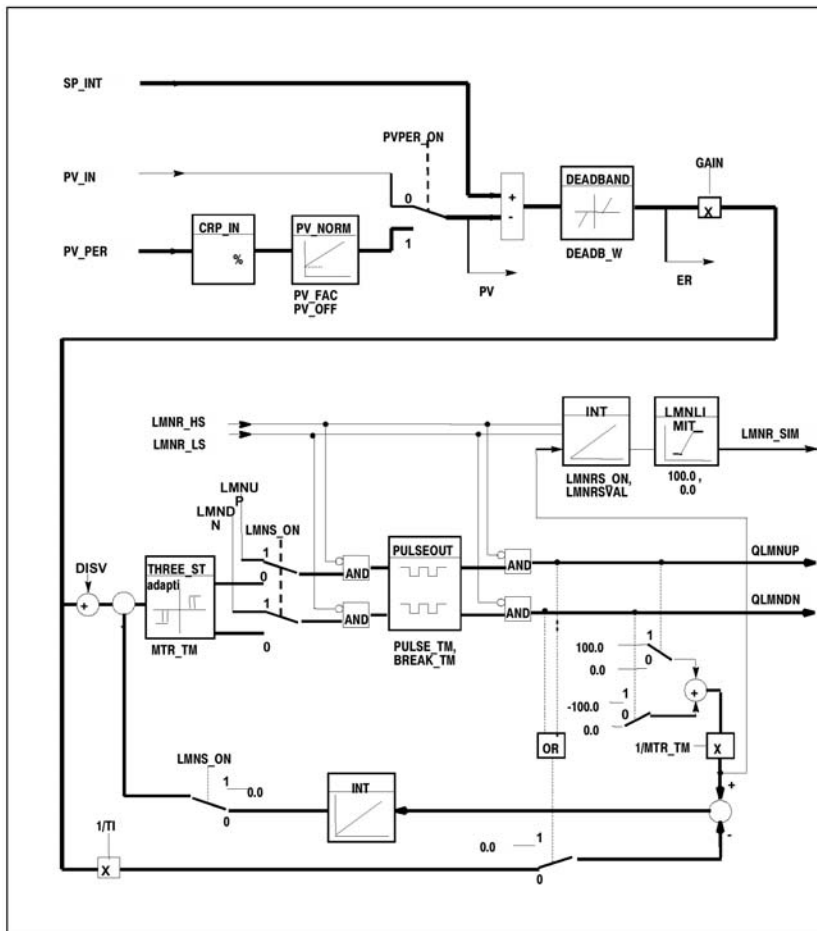
### 初始化

SFB/FB42 “CONT\_S” 有一个初始化过程，当输入参数 COM\_RST = TRUE 时执行该过程。所有其它输出端被置成缺省值。

### 故障信息

故障输出参数 RET\_VAL 没有使用。

### 框图



## 输入参数

下表列出 SFB 42/FB 42 “CONT\_S” 输入参数的说明。

参数	数据类	数值	缺省	说明
COM_RST	BOOL		FALSE	COMPLETE RESTART（完全再启动） 当输入端 COM_RST 置位时，该块执行初始化过程。
LMNR_HS	BOOL		FALSE	HIGH LIMIT OF POSITION FEEDBACK SIGNAL（位置反馈信号上限） “执行器在上限停”信号连接到“位置反馈信号上限”输入端。 LMNR_HS=TRUE 表示执行器在上限停。
LMNR_LS	BOOL		FALSE	LOW LIMIT OF POSITION FEEDBACK SIGNAL（位置反馈信号下限） “执行器在下限停”信号连接到“位置反馈信号下限”输入端。 LMNR_LS=TRUE 表示执行器在下限停。
LMNS_ON	BOOL		TRUE	MANUAL ACTUATING SIGNALS ON（手动执行信号接通） 通过“手动执行信号接通”执行信号处理切换到手动。
LMNUP	BOOL		FALSE	ACTUATING SIGNALS UP（执行信号上升） 通过手动执行信号，输出信号 QLMNUP 被置成输入端“执行信号上升”。
LMNDN	BOOL		FALSE	ACTUATING SIGNALS DOWN（执行信号下降） 通过手动执行信号，输出信号 QLMNDN 被置成输入端“执行信号下降”。
PVPER_ON	BOOL		FALSE	PROCESS VARIABLE PERIPHERAL ON（过程变量外设接通） 如果过程变量从 I/O 读入，输入端 PV_PER 必须连接到 I/O 上且输入端“过程变量外设接通”必须置位。

参数	数据类	数值	缺省	说明
CYCLE	TIME	$\geq 1\text{ms}$	T#1s	SAMPLING TIME (采样时间) 块调用之间的时间必须恒定。“采样时间”输入端确定块调用之间的时间。
SP_INT	REAL	-100.0...+100.0 (%) 或物理值 1)	0.0	INTERNAL SETPOINT(内部设定值) “内部设定值”输入端用于确定设定值。
PV_IN	REAL	-100.0...+100.0 (%) 或物理值 1)	0.0	PROCESS VARIABLE IN (过程变量输入) 可以设置一个初始值到“过程变量输入”输入端或连接一个浮点数值格式的外部过程变量。
PV_PER	WORD		W#16#0000	PROCESS VARIABLE PERIPHERAL (过程变量外设) I/O 格式的过程变量连接到调节器的“过程变量外设”输入端。
GAIN	REAL		2.0	PROPORTIONAL GAIN (比例增益) “比例增益”输入端设置调节器增益。
TI	TIME	$\geq \text{CYCLE}$	T#20s	RESET TIME (复位时间) “复位时间”输入端确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 (死区宽度) 死区用于误差。“死区宽度”确定死区的大小。
PV_FAC	REAL		1.0	PROCESS VARIABLE FACTOR (过程变量系数) “过程变量系统”与过程变量相乘。该输入端用于匹配过程变量的范围。
PV_OFF	REAL	0.0	PROCESS	VARIABLE OFFSET (过程变量偏移量) “过程变量偏移量”输入端与过程变量相加。该输入端用于匹配过程变量的范围。
PULSE_TM	TIME	$\geq \text{CYCLE}$	T#3s	MINIMUM PULSE TIME (最小脉冲时间) 最小脉冲持续时间可以用参数“最小脉冲时间”设置。
BREAK_TM	TIME	$\geq \text{CYCLE}$	T#3s	MINIMUM BREAK TIME (最小间隔时间) 最小脉冲间隔时间可以用参数“最小间隔时间”设置。

参数	数据类	数值	缺省	说明
MTR_TM	TIME	>= CYCLE	T#30s	MOTOR ACTUATING TIME (电动执行时间) 执行器从限幅停到限幅停所需要的时间用“电动执行时间”参数输入。
DISV	REAL	-100.0...100.0 (%) or phys. value 2)	0.0	DISTURBANCE VARIABLE (干扰变量) 对于前馈控制, 干扰变量连接到输入端“干扰变量”。

- 1) 设定值通道和过程变量通道中的参数有相同的单位。
- 2) 被控量通道中的参数有相同的单位。

#### 输出参数

下表列出 SFB 42/FB 42 “CONT\_S” 输出参数的说明。

参数	数据类型	数值	缺省	说明
QLMNUP	BOOL		FALSE	ACTUATING SIGNAL UP (执行信号上升) 如果输出端“执行信号上升”被置位, 那么执行阀是打开的。
QLMNDN	BOOL		FALSE	ACTUATING SIGNAL DOWN (执行信号下降) 如果输出端“执行信号下降”被置位, 那么执行阀是打开的。
PV	REAL		0.0	PROCESS VARIABLE (过程变量) 有效的过程变量在“过程变量”输出端输出。
ER	REAL		0.0	ERROR SIGNAL (误差信号) 有效的误差在“误差信号”输出端上输出。

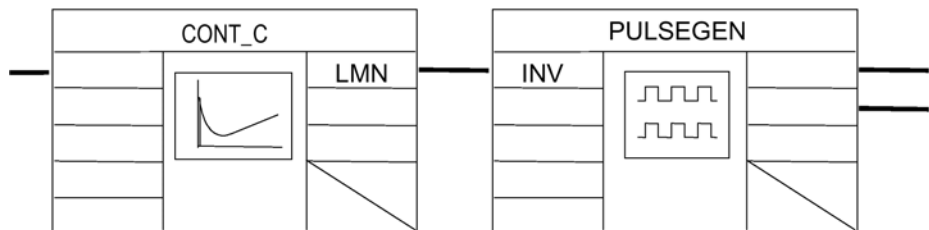
## 24.3 脉冲发生功能 SFB 43/FB 43 “PULSEGEN”

### 介绍

SFB 43 “PULSEGEN”（脉冲发生器）用于为 PID 调节器建造一个用于比例执行器的脉冲输出。电子版手册位于 Documentation > English > STEP7 PID Control。

### 应用

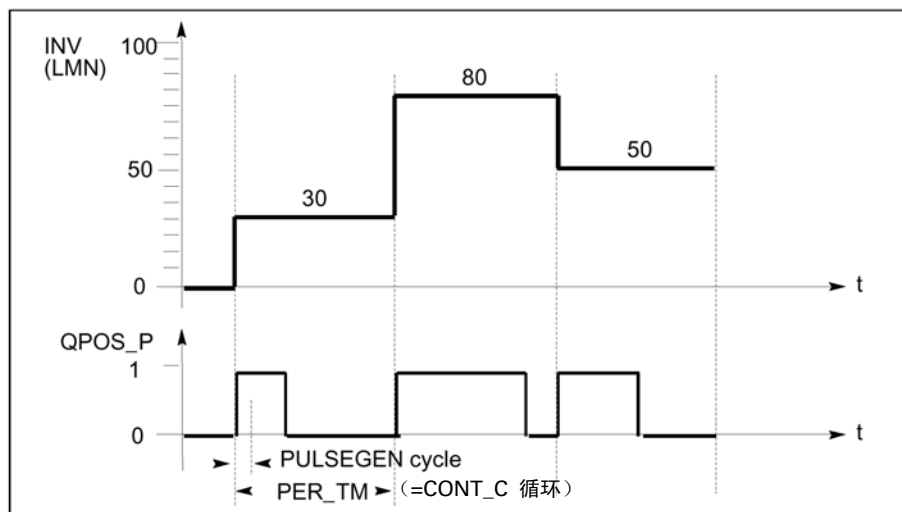
利用 SFB/FB “PULSEGEN”，可以组成带脉宽调制的 PID 两级或三级调节器。该功能通常与连续调节器 “CONT\_C” 一起使用。



### 说明

PULSEGEN 功能通过调制脉冲宽度将输入变量 INV (=PID 调节器的被控量) 转换成带有恒定周期的脉冲序列，周期取决于输入变量更新的循环时间，由 PER\_TM 输入。

每个周期脉冲的宽度与输入变量成正比。赋到 PER\_TM 上的循环时间与 SFB/FB “PULSEGEN” 的处理时间不一致。PER\_TM 循环时间由多个 SFB/FB “PULSEGEN” 处理时间组成，每个 PER\_TM 循环时间内调用 SFB/FB “PULSEGEN” 的次数是衡量脉宽调制精度的标准。

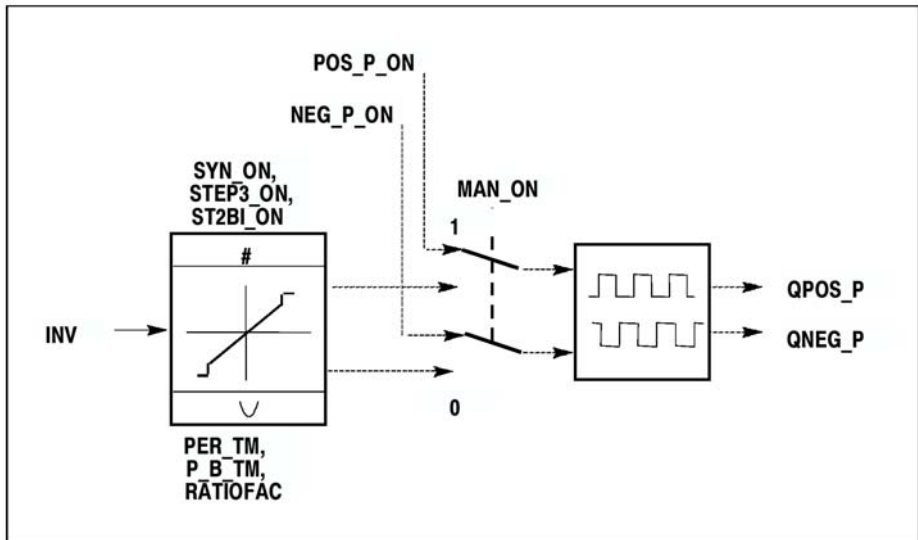


### 脉宽调制

输入变量 30%以及每个 PER\_TM 循环时间调用 SFB/FB “PULSEGEN” 10 次含义如下：

- 前三次调用 SFB/FB “PULSEGEN” 时 QPOS 输出端上输出 “1”（10 次调用的 30%）
- 其余七次调用 SFB/FB“PULSEGEN”时 QPOS 输出端上输出“0”（10 次调用的 70%）

### 框图



### 被控量的精度

如果“采样比例”为 1:10（CONT\_C 调用比 PULSEGEN 调用），那么在这个例子中被控量的精度被限定在 10%。换句话说，设定的输入值 INV 只能在 QPOS 输出端上以 10 % 的步长转换成脉冲宽度。

只有当每次 CONT\_C 调用中 SFB/FB “PULSEGEN” 调用的次数增加时才能提高精度。

例如，如果 PULSEGEN 调用的次数比 CONT\_C 多 100 次，那么被控量的精度达到 1 %。

### 注意

调用频率必须由用户编写。

### 自动同步

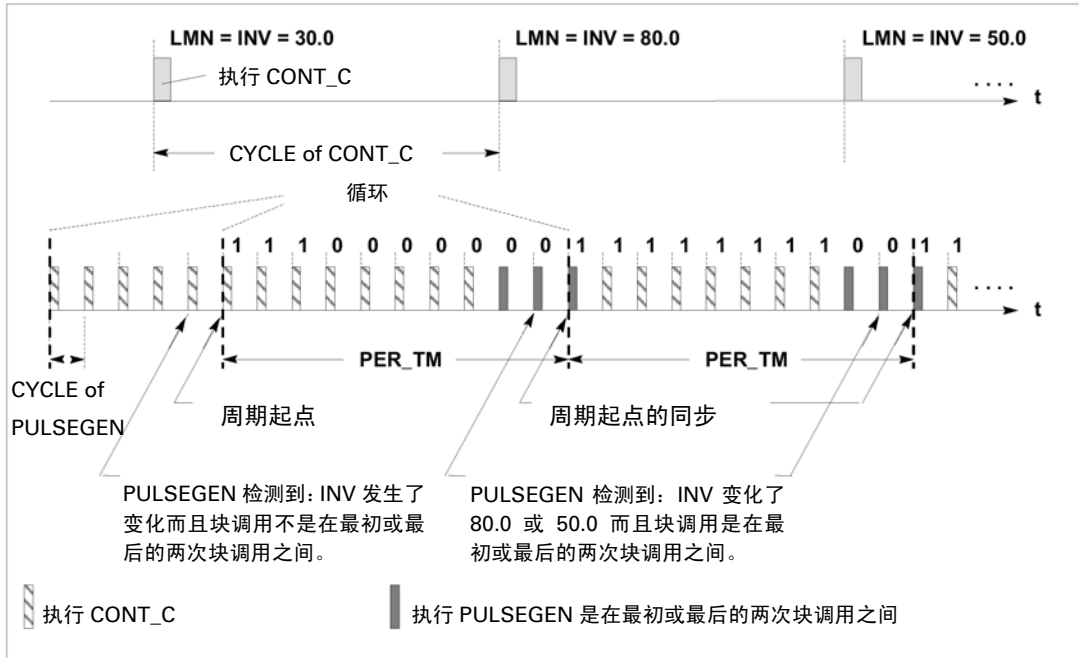
可以使脉冲输出与刷新输入变量 INV 的块（例如 CONT\_C）同步。这就保证了输入变量的变化尽可能快地作为脉冲输出。

脉冲发生器以与周期 PER\_TM 相应的时间间隔处理输入变量 INV，并将变量值转换成相应长度的脉冲信号。

然而由于通常 INV 都以一个较慢的循环中断级计算，因此脉冲发生器应该在 INV 刷新后尽可能快地将离散值转换成脉冲信号。

为实现这一功能，可以按下列过程对块循环周期的起点进行同步：

如果 INV 变化而且块调用不是在最初或最后的两次块调用之间，那么进行同步。这时重新计算脉冲宽度并在下一次循环中输出一个新的周期。



自动同步可以用“SYN\_ON”输入端封锁 (= FALSE)。

### 注意

同步后新的周期开始时，INV（或者说 LMN）原来的值转换成脉冲信号，其精确与同步相关。

### 模式

根据赋给脉冲发生器的参数，可以将 PID 调节器组态成具有三级输出或者双向或单向的两级输出。下表列出开关组合的设置以实现各种模式。

模式	MAN_ON	开关 STEP3_ON	ST2BI_ON
三级调节	FALSE	TRUE	ANY
两级调节，带双向调节区 (-100% 或+100%)	FALSE	FALSE	TRUE
两级调节，带单向调节区 (0%或...100%)	FALSE	FALSE	FALSE
手动模式	TRUE	ANY	ANY



### 三级调节

在“三级调节”模式下，执行信号可以采取三种状态。开关量输出信号 QPOS\_P 和 QNEG\_P 的值对应执行器的各种状态。

下表给出温度调节的例子：

输出信号	加热	执行器关闭	制冷
QPOS_P	TRUE	FALSE	FALSE
QNEG_P	FALSE	FALSE	TRUE

根据输入变量，利用特性曲线来计算脉冲宽度。特性曲线的形状由最小脉冲或最小间隔时间与比例系数定义。

比例系数的一般值是 1。

曲线中的“拐点”由最小脉冲或最小间隔时间引起。

#### 最小脉冲或最小间隔时间

正确分配最小脉冲或最小间隔时间 P\_B\_TM 可以防止过短的开/关时间以延长开关元件和执行器的工作寿命。

#### 注意

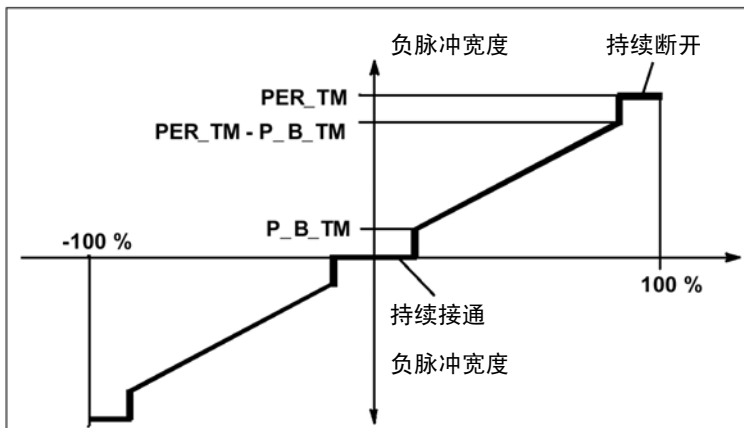
若没有最小脉冲限制，输入变量 LMN 上的小的绝对值会生成过短的脉冲宽度。大的输入值会产生比 PER\_TM - P\_B\_TM 设置的 100 % 或 -100 % 更大的脉冲宽度。

正的或负的脉冲的宽度由输入变量（%）乘以周期计算。

$$\text{周期} = \frac{INV}{100} * \text{PER\_TM}$$

下图给出一个三级调节器（比例系数= 1）的对称曲线。

$$\text{周期} = \frac{INV}{100} * \text{PER\_TM}$$



### 非对称的三级调节

利用比例系数 RATIOFAC，可以改变正脉冲到负脉冲宽度的比例。例如在一个供热系统中，可以允许供热和制冷有水同的系统时间常数。

比例系数也可以影响最小脉冲或最小间隔时间。比例系数 < 1 表示负脉冲的阈值与比例系数相乘。

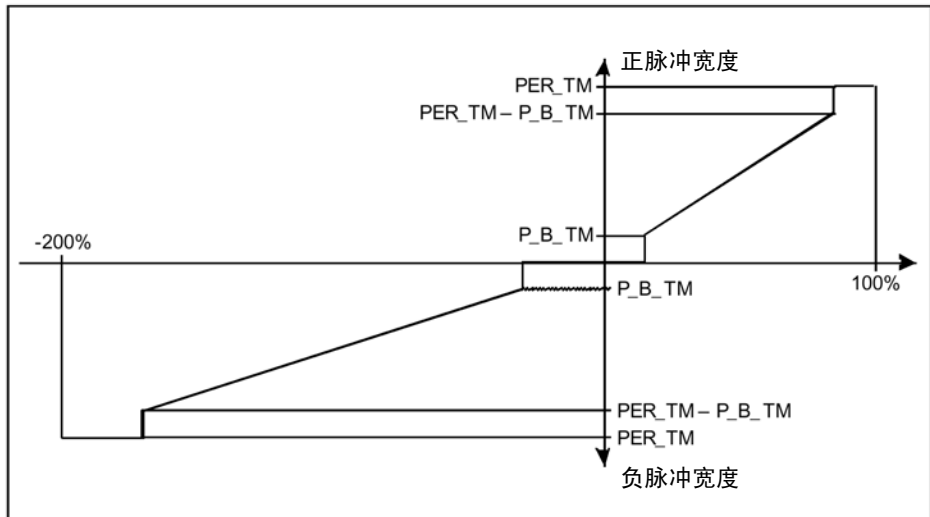
#### 比例系数 < 1

由输入变量乘以周期算出的负脉冲输出的脉冲宽度被比例系数降低。

$$\text{正脉冲宽度} = \frac{INV}{100} * PER\_TM$$

$$\text{负脉冲宽度} = \frac{INV}{100} * PER\_TM * RATIOFAC$$

下图给出三次调节器的非对称曲线（比例系数 = 0.5）：



#### 比例系数 > 1

由输入变量乘以周期算出的负脉冲输出的脉冲宽度被比例系数降低。

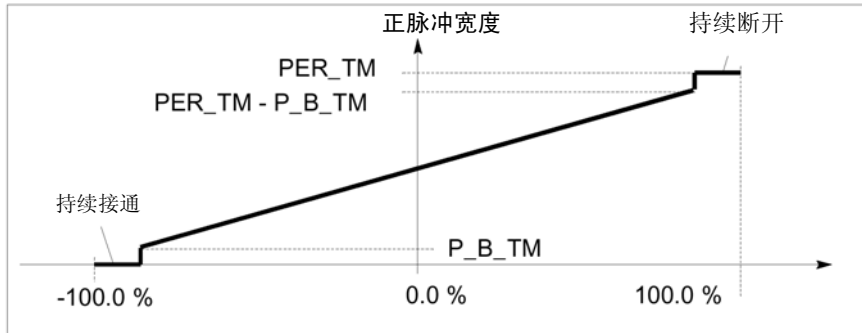
$$\text{负脉冲宽度} = \frac{INV}{100} * PER\_TM$$

$$\text{正脉冲宽度} = \frac{INV}{100} * \frac{PER\_TM}{RATIOFAC}$$

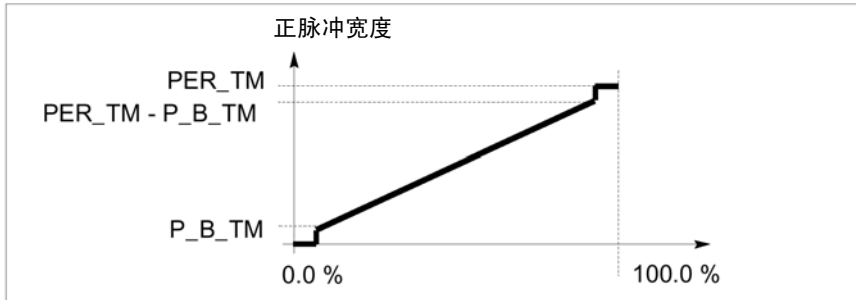
### 两级调节

在两级调节中，只有 PULSEGEN 的正脉冲输出 QPOS\_P 连接到接通/断开执行器上。根据所使用的被控量的范围，两级调节器有双向或单向被控量范围。

两级调节器带双向被控量范围（-100% 至 100%）



两级调节器带单向被控量范围（0% 至 100%）



如果两级调节器在闭环控制中要求连接逻辑反向的二进制信号，那么可以从 QNEG\_P 端输出取反的输出信号。

脉冲	执行器接通	断开
QPOS_P	TRUE	FALSE
QNEG_P	FALSE	TRUE

两级/三级调节中的手动模式如果两级调节器在闭环控制中要求连接逻辑反向的二进制信号，那么可以从 QNEG\_P 端输出取反的输出信号。

调节	POS_P_ON	NEG_P_ON	QPOS_P	QNEG_P
三级调节	FALSE	FALSE	FALSE	FALSE
	TRUE	FALSE	TRUE	FALSE
	FALSE	TRUE	FALSE	TRUE
	TRUE	TRUE	FALSE	FALSE
两级调节	FALSE	Any	FALSE	TRUE
	TRUE	Any	TRUE	FALSE

### 初始化

SFB/FB42 “PULSGEN” 有一个初始化过程，当输入参数 COM\_RST = TRUE 时执行该过程。

所有信号输出端被置成。

### 故障信息

故障输出参数 RET\_VAL 没有使用。

### 输入参数

参数	数据类型	数值范围	缺省	说明
INV	REAL	-100.0...100.0 (%)	0.0	INPUT VARIABLE (输入变量) 模拟被控量连接到输入参数“输入变量”。
PER_TM	TIME	>=20*循环	T#1s	RESET TIME (最小脉冲/间隔时间) 脉宽调制恒定的周期用输入参数“周期时间”输入。它与调节器的采样时间相关。脉冲发生器的采样时间与调节器采样时间的比例决定脉宽调制的精度。
P_B_TM	TIME	>= CYCLE	T#0ms	MINIMUM BREAK TIME (最小脉冲/间隔时间) 最小脉冲或最小间隔时间可以由输入参数“最小脉冲或最小间隔时间”赋值。
RATIOFAC	REAL	0.1 ...10.0	1.0	RAT10 FACTOR (比例系数) 输入参数“比例系数”可以用于更改负脉冲到正脉冲宽度的比例。例如应供热系统中，它允许对供热和制冷不同的热时间常数进行补偿。(例如，系统中用电加热水冷却)
STEP3_ON,	BOOL		TRUE	THREE STEP CONTROL ON (三级调节接通) 该输入参数激活“三级调节”。在三级调节中，两路输出信号都被激活。
ST2BI_ON	BOOL		FALSE	TWO STEP CONTROL FOR BIPOLAR MANIPULATED VALUE RANGE ON (两级调节，双向被控量范围接通) 利用“两级调节，双向被控量范围接通”可以选择“两级调节，双向被控量范围”模式或“两级调节，单向被控量范围”模式。参数 STEP3_ON = FALSE 必须设置。
MAN_ON	BOOL		FALSE	MANUAL VALUE ON(手动模式接通) 通过设备“手动模式接通”参数，输出信号可以手动设置。

参数	数据类型	数值范围	缺省	说明
POS_P_ON	BOOL		FALSE	POSITIVE PULSE ON (正脉冲接通) 在三级调节手动模式下, 输出信号 QPOS_P 可以在输入参数“正脉冲接通”上设置。在两级调节手动模式下, QNEG_P 总是被置成与 QPOS_P 反向。
NEG_P_ON	BOOL		FALSE	NEGATIVE PULSE ON (负脉冲接通) 在三级调节手动模式下, 输出信号 QNEG_P 可以在输入参数“负脉冲接通”上设置。在两级调节手动模式下, QNEG_P 总是被置成与 QPOS_P 反向。
SYN_ON	BOOL		TRUE	SYNCHRONIZATION ON (同步接通) 通过设置输入参数“同步接通”, 可以自动与刷新输入变量 INV 的块同步。这就保证了更改输入变量时可以尽可能地以脉冲形式输出。
COM_RST	BOOL		FALSE	COMPLETE RESTART (完全再起动) 如果输入端“完全再起动”置位, 该块执行初始化过程。
CYCLE	TIME	>= 1ms	T#10ms	SAMPLING TIME (采样时间) 块调用之间的时间必须恒定。“采样时间”输入端确定块调用之间的时间。

**注意**

输入参数的数值在块中没有限制。没有参数检查。

**输出参数**

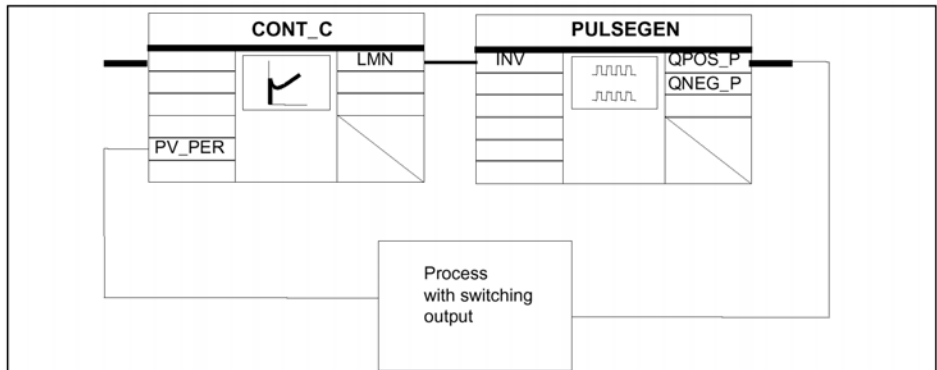
参数	数据类型	数值范围	缺省	说明
QPOS_P	BOOL		FALSE	OUTPUT POSITIVE PULSE (输出正脉冲) 如果有脉冲输出, 输出参数“输出正脉冲”被置位。在三级调节中总是正脉冲输出。在两级调节中, QNEG_P 总是与 QPOS_P 反向。
QNEG_P	BOOL		FALSE	OUTPUT NEGATIVE PULSE (输出负脉冲) 如果有脉冲输出, 输出参数“输出负脉冲”被置位。在三级调节中总是负脉冲输出, 在两级调节中, QNEG_P 总是与 QPOS_P 反向。

## 24.4 PULSEGEN 功能块举例

### 闭环控制

利用连续调节器 CONT\_C 和脉冲发生器 PULSEGEN，可以实现一个带有固定设定值，有用于比例执行器的开关量输出的调节器。

下图给出闭环控制的信号流程。

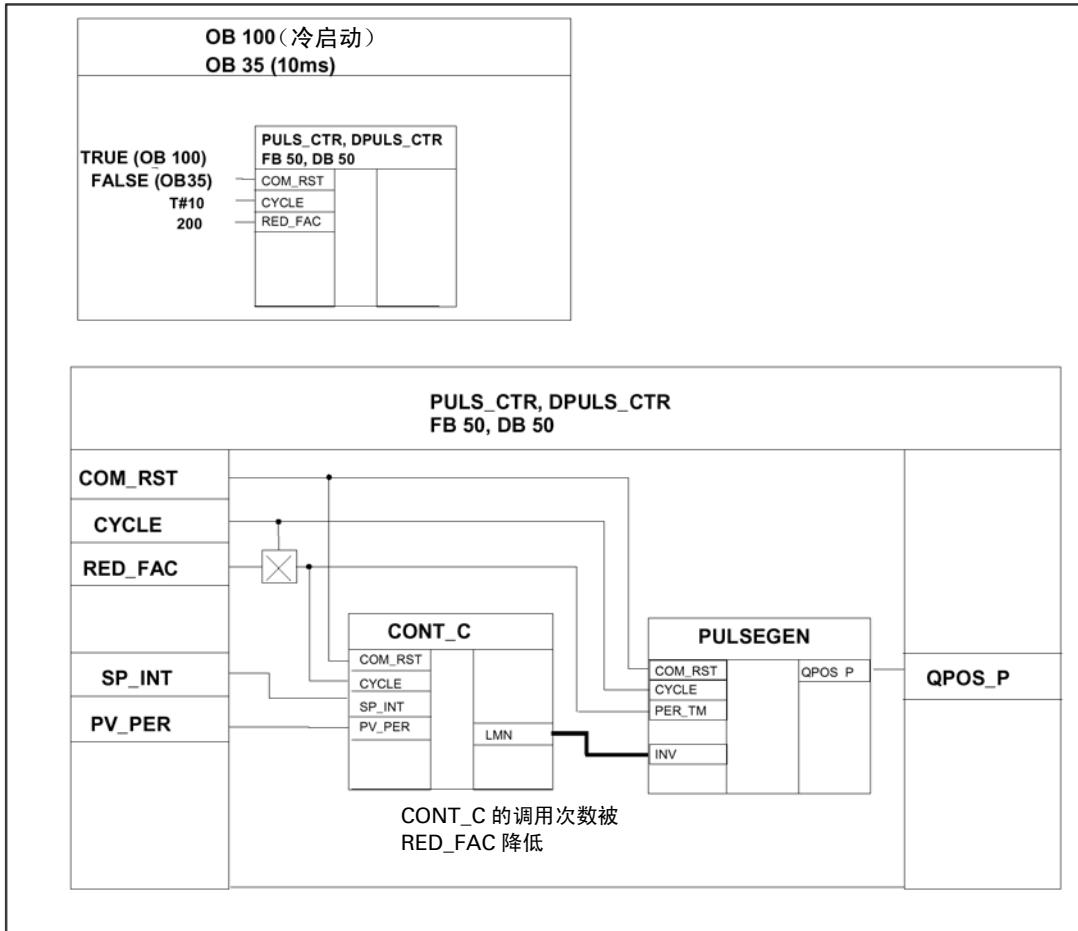


连续调节器 CONT\_C 形成被控量 LMN，然后被控量被脉冲发生器 PULSEGEN 转换成脉冲/间隔信号 QPOS\_P 或 QNEG\_P。

### 块的调用和连接

带有开关量输出（用于比例执行器）的固定设定值调节器 PULS\_CTRL 由功能块 CONT\_C 和 PULSEGEN 组成。执行块调用，使 CONT\_C 每 2 秒（=CYCLE\*RED\_FAC）调用一次，PULSEGEN 每 10 ms（=CYCLE）调用一次。OB35 的循环时间设置为 10 ms。其连接见下图。

在冷起动过程中，功能块 PULS\_CTRL 在 OB100 中调用，且输入端 COM\_RST 被置成 TRUE。



FB PULS\_CTRL 的 STL 程序

地址	声明	名称	数据类型	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	调用次数降低系数
8.0	in	COM_RST	BOOL	完全再启动
10.0	in	CYCLE	TIME	采样时间
14.0	out	QPOS_P	BOOL	执行信号
16.0	stat	DI_CONT_C	FB-CONT_C	计数器
142.0	stat	DI_PULSEGEN	FB-PULSEGEN	计数器
176.0	stat	Scount	INT	计数器
0.0	temp	TcycCtr	TIME	调节器采样时间

STL		说明
A	#COM_RST	//初始化过程
JCN	M001	
L	0	
T	#sCount	
M001:L	#CYCLE	//计算调节器采样时间
L	#RED_FAC	
*D		
T	#tCycCtr	
L	#sCount	//减计数并与零比较
L	1	
-I		
T	#sCount	
L	0	
<=I		
JCN	M002	有条件块调用并起动计数器。
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		





## 25 用于紧凑型 CPU 的 SFB

### 25.1 使用 SFB 44 “Analog” 实现模拟量输出定位

#### 说明

通过用户程序控制定位功能，可使用 SFB ANALOG（SFB 44）

由一个固定分配的模拟量输出通过 $\pm 10\text{ V}$ （电压信号）或 $\pm 20\text{ mA}$ （电流信号）来控制给定。

- 在加速阶段（RAM\_UP）后，驱动按设定速度（V\_设定）接近目标点。
- 在制动点，由 CPU 计算开始减速（RAMP\_DN）至转换点。
- 一旦到达转换点，则以爬行速度（V\_爬行）运行。
- 驱动在关断点被关断。
- 每次步进的转换点和关断点由参数转换距离和关断距离中的取值决定。正向移动和负向移动的转换距离和关断距离可以不同。
- 当到达关断点时，运行结束（WORKING = FALSE）。可启动新的运行。
- 当实际值进入目标范围内时，指定的目标点被到达（POS\_RCD = TRUE）。如果没有启动新的运行，而实际值发生漂移，“位置到达”信号不复位。

当转换距离小于关断距离时，驱动在制动点减速，设定速度为 0。

#### 基本参数:

这里我们对各种工作模式下 SFB 的参数进行说明。与工作模式有关的参数说明在每种工作模式中。

#### 参数:

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
LADDR	INPUT	WORD	0	CPU 指定	W#16#0310	在“HW Config”中指定的模板 I/O 地址。 如果 I 和 O 的地址不同，按较小的地址指定。
CHANNEL	INPUT	INT	2	0	0	通道号
STOP	INPUT	BOOL	4.4	TRUE/FALSE	FALSE	停止运行 使 STOP = TRUE 可提前停止/断运行
ERR_A	INPUT	BOOL	4.5	TRUE/FALSE	FALSE	外部错误确认如果 ERR_A = TRUE，外部错误被清除。

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
SPEED	INPUT	DINT	12	爬行速度最大 1,000,000 脉冲 /s, 不能高于参数 规定的最大速度	1000	轴加速到设定速度 V 设定在运行 期间不能改变速度。
WORKING	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
ACT_POS	OUTPUT	DINT	18	$-5 \times 10^8$ to $+5 \times 10^8$ pulses	0	实际位置值
MODE_OUT	OUTPUT	INT	22	0, 1, 3, 4, 5	0	激活/配置的工作模式
ERR	OUTPUT	WORD	24	每位“0”至“1”:	0	外部错误。 位 2: 零脉监控 位 11: 移动范围监控 (常“1”) 位 12: 工作范围监控 位 13: 实际值监控 位 14: 目标置位监控 位 15: 目标范围监控 剩余位含义保留
ST_ENBLD	OUTPUT	BOOL	26.0	TRUE/FALSE	TRUE	如果下列条件全部满足, CPU 置 位启动使能 <ul style="list-style-type: none"> <li>• 无 STOP (STOP = FALSE)</li> <li>• 无外部错误 (ERR = 0)</li> <li>• 驱动使能置位 (DRV_EN = TRUE)</li> <li>• 无定位运行被激活 (WORKING = FALSE)</li> </ul>
ERROR	OUTPUT	BOOL	26.1	TRUE/FALSE	FALSE	启动/继续运行错误
STATUS	OUTPUT	WORD	28.0	W#16#0000 to W#16#FFFF	W#16#0000	故障代码

无需分配的参数（静态局部数据）

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
ACCEL	STATIC	DINT	30	1 到 100, 000 脉冲/s <sup>2</sup>	100	加速度运行中不能改变。
DECEL	STATIC	DINT	34	1 到 100, 000 脉冲/s <sup>2</sup>	100	减速度运行中不能改变。
CHGDIFF_P	STATIC	DINT	38	0 到+10 <sup>8</sup> 脉冲	1000	正向转换距离： “正向转换距离”定义了转换点， 从该点处，驱动继续以爬行速度向前运行
CUTOFF-DIFF_P	STATIC	DINT	42	0 到+10 <sup>8</sup> 脉冲	100	正向关断距离： “正向关断距离”定义了转换点， 从该点处，驱动停止的爬行速度向前运行
CHGDIFF_M	STATIC	DINT	46	0 到+10 <sup>8</sup> 脉冲	1000	负向转换距离： “负向转换距离”定义了转换点， 从该点处驱动继续以爬行速度反向运行
CUTOFF-DIFF_P	STATIC	DINT	50	0 到+10 <sup>8</sup> 脉冲	100	负向关断距离： “负向关断距离”定义了关断点， 从该点处，驱动停止以爬行速度反向运行
PARA	STATIC	BOOL	54.0	TRUE/FALSE	FALSE	该轴已被分配参数
DIR	STATIC	BOOL	54.1	TRUE/FALSE	FALSE	实际/最近的方向检测 FALSE = 向前（正向） TRUE = 反向（负向）
CUTOFF	STATIC	BOOL	54.2	TRUE/FALSE	FALSE	驱动在关断范围内（从关断点到下次启动点）
CHGOVER	STATIC	BOOL	54.3	TRUE/FALSE	FALSE	驱动在转换范围内（从达到爬行速度到下次运行启动点之间）
RAMP_DN	STATIC	BOOL	54.4	TRUE/FALSE	FALSE	驱动减速（从制动点到转换点）
RAMP_UP	STATIC	BOOL	54.5	TRUE/FALSE	FALSE	驱动加速（从启动直至达到 SPEED（V 指定）速度）
DIST_TO_GO	STATIC	DINT	56	-5x10 <sup>8</sup> to +5x10 <sup>8</sup> 脉冲	0	实际剩余距离
LAST_TRG	STATIC	DINT	60	-5x10 <sup>8</sup> to +5x10 <sup>8</sup> 脉冲	0	最近/当前目标点 <ul style="list-style-type: none"> <li>绝对值步进：运行起点 LST_TRG = 当前目标点绝对值 (TARGET)</li> <li>相对值步进：运行起点 LST_TRG = LAST_TRG 指定为前次运行点 (TARGET) +/- 的距离。</li> </ul>

## 用于“点动”模式的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动使能
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	工作模式， 1 = 点动
WORKING	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
ACT_POS	OUTPUT	DINT	18	-5×10 <sup>8</sup> 至 +5×10 <sup>8</sup> 脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	22	0, 1, 3, 4, 5	0	激活/配置的工作模式

## 用于“回参考点”模式的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	正向回参考点（上升沿）
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	负向回参考点（上升沿）
MODE_IN	INPUT	INT	6	0, 1, 3, 4, 5	1	工作模式 3 = 回参考点
WORKING	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
SYNC	OUTPUT	BOOLT	16.3	-5×10 <sup>8</sup> 至 +5×10 <sup>8</sup> 脉冲	FALSE	SYNC = TRUE: 轴同步
ACT_POS	OUTPUT	DINT	18	-5×10 <sup>8</sup> 至 +5×10 <sup>8</sup> 脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	22	0, 1, 3, 4, 5	0	激活/配置的工作模式。

## 用于“相对值步进”模式的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	正向运行（上升沿）
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	负向运行（上升沿）
MODE_IN	INPUT	INT	6	0, 1, 3, 4, 5	1	工作模式 4 = 相对值步进
TARGET	INPUT	DINT	8	0 to 10 <sup>9</sup> 脉冲	1000	距离脉冲数（只允许正值）
WORKING	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
POS_RCD	OUTPUT	BOOLT	16. 1	-5×10 <sup>8</sup> 至 +5×10 <sup>8</sup> 脉冲	FALSE	位置到达
ACT_POS	OUTPUT	DINT	18	-5×10 <sup>8</sup> 至 +5×10 <sup>8</sup> 脉冲	0	实际位置值
MODE_OUT	OUTPUT	_INT	22	0, 1, 3, 4, 5	0	激活/配置的工作模式

## 用于“绝对值步进”模式的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动使能
START	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	启动运行（上升沿）
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	正向运行（上升沿）
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	负向运行（上升沿）
MODE_IN	INPUT	INT	6	0, 1, 3, 4, 5	1	工作模式, 5=绝对值步进
TARGET	INPUT	DINT	8	线性轴 $-5 \times 10^8$ 至 $+5 \times 10^8$ 转动轴: 0 转动轴结束 -1	1000	目标点脉冲数
WORKING	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	正在运行
POS_RCD	OUTPUT	BOOL	16.1	TRUE/FALSE	FALSE	位置到达
ACT_POS	OUTPUT	DINT	18	$-5 \times 10^8$ 至 $+5 \times 10^8$ 脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	22	1, 3, 4, 5, 0	0	激活/配置的工作模式

## 用于“设置参考点”工作的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
SYNC	OUTPUT	BOOL	16.3	TRUE/FALSE	FALSE	轴同步

## 无需分配的参数（静态局部数据）

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
JOB_REQ	STATIC	BOOL	76.0	TRUE/FALSE	FALSE	作业启动（上升沿）
JOB_DONE	STATIC	BOOL	76.1	TRUE/FALSE	TRUE	可以启动新工作
JOB_ERR	STATIC	BOOL	76.2	TRUE/FALSE	FALSE	作业错误
JOB_ID	STATIC	INT	78	1, 2	0	作业号 1=设置参考点
JOB_STAT	STATIC	WORD	80	W#16#0000 to W#16#FFFF	W#16#0000	作业错误代码
JOB_VAL	STATIC	DINT	82	$-5 \times 10^8$ 到 $+5 \times 10^8$ 脉冲	0	作业参数: 参考点坐标值

## 用于“删除余程”工作的参数

## 无需分配的参数（静态局部数据）

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
JOB_REQ	STATIC	BOOL	76.0	TRUE/FALSE	FALSE	作业启动（上升沿）
JOB_DONE	STATIC	BOOL	76.1	TRUE/FALSE	TRUE	可以启动新工作
JOB_ERR	STATIC	BOOL	76.2	TRUE/FALSE	FALSE	作业错误
JOB_ID	STATIC	INT	78	1, 2	0	作业号 2=删除余程
JOB_STAT	STATIC	WORD	80	W#16#0000 至 W#16#FFFF	W#16#0000	作业错误代码
JOB_VAL	STATIC	DINT	82	-	0	任何设定

用于“长度测量”作业的参数

该作业由数字输入端的上升沿启动。无输入参数。

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
MSR_DONE	OUTPUT	BOOL	16.2	TRUE/FALSE	FALSE	长度测量结束

无需分配的参数（静态局部数据）

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
BEG_VAL	STATIC	DINT	64	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 脉冲	0	测量开始时实际位置值
END_VAL	STATIC	DINT	68	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 脉冲	0	测量结束时实际位置值
LEN_VAL	STATIC	DINT	72	0 到 10 <sup>9</sup> 脉冲	0	测量的长度值

故障信息

运行模式错误（ERROR = TRUE）

如果检查到错误，输出参数 ERROR=TRUE。参数 STATUS 指明故障原因。

事件类别 故障代码	说明
W#16#2002	错误的 SFB，使用 SFB 44
W#16#2004	错误的通道号（CHANNEL）。设置通道“0”
W#16#3001	在相同 SFB 调用中工作错误使得运行工作被拒绝。修改相关的 JOB 参数
W#16#3002	驱动工作时不能修改 MODE_IN。等待当前定位运行结束。
W#16#3003	不识别的工作模式（MODE_IN）。允许值为 1（手动），3（回参考点），4（相对值步进），5（绝对值步进）。
W#16#3004	同一时刻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	START 只允许在“绝对值步进”工作模式下。用 DIR_P 或 DIR_M 启动运行。
W#16#3006	在“绝对值步进”工作模式下，DIR_P 或 DIR_M 不允许用于线性轴，用 START 启动运行。
W#16#3007	轴未同步。“绝对值步进”只能用于已同步的轴。
W#16#3008	清除的工作范围。返回到工作位置只允许在手动模式下。
W#16#3101	轴未参数化不能启动。通过硬件组态对“定位”模板进行参数化
W#16#3102	驱动未使能不能启动。在 SFB 中置位驱动使能（DRV_EN=TRUE）
W#16#3103	STOP 被置位不能启动。在 SFB 中复位 STOP（STOP=FALSE）
W#16#3104	驱动当前正在执行定位运行（WORKING=TRUE）不能启动。等待当前运行结束。
W#16#3105	至少有一个错误未清除，不能启动。消除并复位所有外部错误后再启动运行。
W#16#3202	SPEED 中速度设定错误。速度设定值尽管低于参数化的最大值，但超过了爬行速度允许的范围，最大 1000000 脉冲/s。
W#16#3203	ACCEL 中加速度设定值超过了 1 到 100, 000 脉冲/s <sup>2</sup> 的范围。

事件类别 故障代码	说明
W#16#3204	DECEL中减速度设定值超过了 1 到 100, 000 脉冲/s <sup>2</sup> 的范围。
W#16#3206	SPEED 中速度设定值必须大于或等于参数化的基准频率。
W#16#3301	转换/关断距离太大。最大转换/关断距离为 10 <sup>8</sup>
W#16#3304	关断距离太小。关断距离至少应为目标范围的一半。
W#16#3305	转换距离太小。转换距离至少应为目标范围的一半。
W#16#3401	目标设置超出工作范围。对线性轴在步进模式下,目标必须设定在软限位之内(包含)。
W#16#3402	目标设定错误。对旋转轴目标必须设定在 0 和旋转轴终点之间。
W#16#3403	距离设定错误。在相对值步进模式下,移动距离必须设为正值。
W#16#3404	距离设定错误。目标的绝对坐标必须大于-5x10 <sup>8</sup>
W#16#3405	距离设定错误。目标的绝对坐标必须小于 5x10 <sup>8</sup>
W#16#3406	距离设定错误。目标的绝对坐标必须在工作范围内 (+/-目标范围的一半)
W#16#3501	移动距离太大。目标坐标+当前余程必须大于/等于-5x10 <sup>8</sup>
W#16#3502	移动距离太大。目标坐标+当前余程必须小于/等于 5x10 <sup>8</sup>
W#16#3503	移动距离太小。正向移动距离必须大于指定的正向关断距离
W#16#3504	移动距离太小。负向移动距离必须大于指定的负向关断距离
W#16#3505	移动距离太短或正限位已作用。最近一次的正向目标点(工作范围或位移极限)太靠近实际位置。
W#16#3506	移动距离太短或负限位已作用。最近一次的负向目标点(工作范围或位移极限)太靠近实际位置。

#### 作业故障 (JOB\_ERR = TRUE)

如果检查到故障,则输出参数 JOB\_ERROR=TRUE。参数 JOB\_STAT 中指明故障原因。

事件类别 错误代码	说明
W#16#4001	轴未参数化。通过硬件组态对“定位”模板进行参数化
W#16#4002	定位进行中工作不允许。等待直至 WORKING = FALSE 再重复工作。
W#16#4004	不识别的工作号。检查工作 ID 再重复工作。
W#16#4101	对线性轴参考点坐标不能超出工作范围极限。
W#16#4102	对线性轴参考点坐标+实际余程必须大于/等于-5x10 <sup>8</sup>
W#16#4103	对线性轴参考点坐标+实际余程必须小于/等于 5x10 <sup>8</sup>
W#16#4104	对线性轴参考点坐标+到起始点的余程必须大于/等于-5x10 <sup>8</sup>
W#16#4105	对线性轴参考点坐标+到起始点的余程必须小于/等于 5x10 <sup>8</sup>
W#16#4106	对旋转轴参考点坐标不能小于 0 和大于/等于旋转轴终点。



### 外部故障（ERR）

技术电路将监控运行，移动距离和连接的外设。前提是你已经在“驱动”，“轴”和“编码器”参数屏幕表中启动了监控。

当监控单元被触发时报告外部故障。外部故障可独立于已启动的工作。你必须使用 `ERR_A = TRUE` 来清除外部故障。

SFB 参数 ERR（WORD）中的设置位。

监控	故障代码	ERR-WORD 中的位
零脉冲（零标记）	W#16#0004	2
移动距离	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

### 系统故障

系统故障发生时 `BIE = FALSE`。当读/写背景 DB 或多次调用 SFB 错误时会引起系统故障。

## 25.2 使用 SFB 46 “DIGITAL” 实现数字量输出定位

### 说明

通过用户程序控制定位功能，可使用 SFB DIGITAL（SFB 46）

四个 24-V 数字量输出固定地分配给驱动，来控制给定。根据控制方式的配置。数字量输出控制方向和速度（快速/爬行）。

距离测量通过不对称的 24-V 增量编码器，两个相位差 90 度。

- 首先，快速（V 快速）接近目标点
- 在转换点，速度转换成爬行速度（V 爬行）
- 驱动在关断点被关断。
- 每次步进转换点和关断点由参数转换距离和关断距离中的取值决定。正向移动和负向移动的转换距离和关断距离可以不同。
- 当到达关断点时，运行结束（`WORKING = 0`）。可启动新的运行。
- 当实际值进入目标范围内时，指定的目标点被到达（`POS_RCD = 1`）如果没有启动新的运行，而实际值发生漂移，“位置到达”信号不复位。

### 基本参数：

这里我们对各种工作模式下 SFB 的参数进行说明。与工作模式有关的参数说明在每种工作模式中。

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
LADDR	INPUT	WORD	0	CPU 指定	W#16#0310	在“HW Config”中指定的模板 I/O 地址。如果 I 和 O 的地址不同，按较小的地址指定。
CHANNEL	INPUT	INT	2	0	0	通道号
STOP	INPUT	BOOL	4.4	TRUE/FALSE	FALSE	停止运行 使 STOP = TRUE 可提前停止/ 断运行
ERR_A	INPUT	BOOL	4.5	TRUE/FALSE	FALSE	外部错误确认如果 ERR_A = TRUE，外部错误被清除。
SPEED	INPUT	BOOL	12.0	TRUE/FALSE	FALSE	用于快速/爬行模式的速度级 TRUE =快速模式 FALSE =爬 行模式
WORKING	OUTPUT	BOOL	14.0	TRUE/FALSE	FALSE	正在运行
ACT_POS	OUTPUT	DINT	16	-5x10 <sup>8</sup> 至 5x10 <sup>8</sup> 脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0, 1, 3, 4, 5	0	激活/配置的工作模式
ERR	OUTPUT	WORD	22	每一位 "0" 或 "1":	0	外部错误。 位 2: 零脉监控 位 11: 移动范围监控 (常“1”) 位 12: 工作范围监控 Bit13:实际值监控 Bit14:目标本位监控 位 15: 目标范围监控 剩余位含义保留
ST_ENBLD	OUTPUT	BOOL	24.0	TRUE/FALSE	TRUE	如果下列条件全部满足，CPU 置位启动使能 无STOP (STOP = FALSE) 无外部错误 (ERR = 0) 驱动使能置位 (DRV_EN = TRUE) 无定位运行被激活 (WORKING = FALSE)
ERROR	OUTPUT	BOOL	24.1	TRUE/FALSE	FALSE	启动/继续运行错误
STATUS	OUTPUT	WORD	26.0	W#16#0000 至 W#16#FFFF	W#16#0000	错误代码

无需分配的参数（静态局部数据）

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
CHGDIFF_P	STATIC	DINT	28	0 到+10 <sup>8</sup> 脉冲	1000	正向转换距离： “正向转换距离”定义了转换点，在该点处驱动继续以爬行速度向前运行
CUTOFF- DIFF_P	STATIC	DINT	32	0 到+10 <sup>8</sup> 脉冲	100	正向关断距离： “正向关断距离”定义了转换点，从该点处驱动停止的爬行速度向前运行
CHGDIFF_M	STATIC	DINT	36	0 到+10 <sup>8</sup> 脉冲	1000	负向转换距离： “负向转换距离”定义了转换点，在该点处驱动继续以爬行速度反向运行
CUTOFF- DIFF_P	STATIC	DINT	40	0 到+10 <sup>8</sup> 脉冲	100	负向关断距离：“负向关断距离”定义了关断点，在该点处驱动停止以爬行速度反向运行
PARA	STATIC	BOOL	44.0	TRUE/FALSE	FALSE	该轴已被分配参数
DIR	STATIC	BOOL	44.1	TRUE/FALSE	FALSE	实际/最近的方向检测 FALSE = 向前(正向) TRUE = 反向(负向)
CUTOFF	STATIC	BOOL	44.2	TRUE/FALSE	FALSE	驱动在关断范围内（从关断点到下次启动点）
CHGOVER	STATIC	BOOL	44.3	TRUE/FALSE	FALSE	驱动在转换范围内（从达到爬行速度到下次运行启动点之间）
DIST_TO_GO	STATIC	DINT	46	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 脉冲	0	实际剩余距离
LAST_TRG	STATIC	DINT	50	-5x10 <sup>8</sup> 到 +5x10 <sup>8</sup> 脉冲	0	最近/当前目标点 <ul style="list-style-type: none"> <li>绝对值步进：</li> <li>运行起点 LST_TRG = 当前目标点绝对值 (TARGET)</li> <li>相对值步进：运行起点 LST_TRG = LAST_TRG 指定为前次运行点 (TARGET) +I 的距离。</li> </ul>

## 用于“点动”模式的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动使能
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	工作模式， 1 = 点动
WORKING	OUTPUT	BOOL	14.0	TRUE/FALSE	FALSE	正在运行
ACT_POS	OUTPUT	DINT	16	-5x10 <sup>8</sup> 至 +5x10 <sup>8</sup> 脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0, 1, 3, 4, 5	0	激活/配置的工作模式

## 用于“回参考点”模式的参数

## 用于“点动”模式的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	正向回参考点（上升沿）
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	负向回参考点（上升沿）
MODE_IN	INPUT	INT	6	0, 1, 3, 4, 5	1	工作模式 3 = 回参考点
WORKING	OUTPUT	BOOL	14.0	TRUE/FALSE	FALSE	正在运行
SYNC	OUTPUT	BOOL	14.3	TRUE/FALSE	FALSE	SYNC = TRUE: 轴同步
ACT_POS	OUTPUT	DINT	16	-5x10 <sup>8</sup> 至 +5x10 <sup>8</sup> 脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0, 1, 3, 4, 5	0	激活/配置的工作模式

## 用于“相对值步进”模式的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动使能
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	正向运行（上升沿）
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	负向运行（上升沿）
MODE_IN	INPUT	INT	6	0, 1, 3, 4, 5	1	工作模式 4 = 相对值步进
TARGET	INPUT	DINT	8	0 to 10 <sup>9</sup> 脉冲	1, 000	距离脉冲数（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	位置到达
ACT_POS	OUTPUT	DINT	16	-5x10 <sup>8</sup> 至 +5x10 <sup>8</sup> 脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0, 1, 3, 4, 5	0	激活/配置的工作模式

用于“绝对值步进”模式的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
DRV_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	驱动使能
START	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	启动运行（上升沿）
DIR_P	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	正向运行（上升沿）
DIR_M	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	负向运行（上升沿）
MODE_IN	INPUT	INT	6	0, 1, 3, 4, 5	1	工作模式, 5=绝对值步进
TARGET	INPUT	DINT	8	线性轴 $-5 \times 10^8$ 至 $+5 \times 10^8$ 转动轴: 0 至转动轴结束 -1	1, 000	目标点脉冲数
WORKING	OUTPUT	BOOL	14.0	TRUE/FALSE	FALSE	正在运行
POS_RCD	OUTPUT	BOOL	14.1	TRUE/FALSE	FALSE	位置到达
ACT_POS	OUTPUT	DINT	16	$-5 \times 10^8$ 至 $+5 \times 10^8$ 脉冲	0	实际位置值
MODE_OUT	OUTPUT	INT	20	0, 1, 3, 4, 5	0	激活/配置的工作模式

用于“设置参考点”工作的参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
SYNC	OUTPUT	BOOL	14.3	TRUE/FALSE	FALSE	轴同步

无需分配的参数（静态局部变量）

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
JOB_REQ	STATIC	BOOL	66.0	TRUE/FALSE	FALSE	作业启动（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	作业错误
JOB_ID	STATIC	INT	68	1, 2	0	作业号 1=设置参考点
JOB_STAT	STATIC	WORD	70	W#16#0000 至 W#16#FFFF	W#16# 0000	作业错误代码
JOB_VAL	STATIC	DINT	72	$-5 \times 10^8$ 到 $+5 \times 10^8$ 脉冲	0	作业参数：参考点坐标值

## 用于“删除余程”作业的参数

无需分配的参数（静态局部变量）

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
JOB_REQ	STATIC	BOOL	66.0	TRUE/FALSE	FALSE	作业启动（上升沿）
JOB_DONE	STATIC	BOOL	66.1	TRUE/FALSE	TRUE	可以启动新工作
JOB_ERR	STATIC	BOOL	66.2	TRUE/FALSE	FALSE	作业错误
JOB_ID	STATIC	INT	68	1, 2	0	作业号 2=删除余程
JOB_STAT	STATIC	WORD	70	0 to FFFF hex	0	作业错误代码
JOB_VAL	STATIC	DINT	72	-	0	任何设定

## 用于“长度测量”作业的参数

该作业由数字输入端的上升沿启动。无输入参数。

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
MSR_DONE	OUTPUT	BOOL	14.2	TRUE/FALSE	FALSE	长度测量结束

无需分配的参数（静态局部变量）

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
BEG_VAL	STATIC	DINT	54	-5x10 <sup>8</sup> 至 +5x10 <sup>8</sup> 脉冲	0	测量开始时实际位置值
END_VAL	STATIC	DINT	58	-5x10 <sup>8</sup> 至 +5x10 <sup>8</sup> 脉冲	0	测量结束时实际位置值
LEN_VAL	STATIC	DINT	62	0 至 10 <sup>9</sup> 脉冲	0	测量的长度值

## 故障信息

运行模式故障（ERROR = TRUE）

如果检查到故障，输出参数 ERROR=TRUE。参数 STATUS 指明故障原因。

事件类别 故障代码	说明
W#16#2001	错误的 SFB，使用 SFB 46
W#16#2004	错误的通道号（CHANNEL）。设置通道“0”
W#16#3001	在相同 SFB 调用中工作错误使得运行工作被拒绝。修改相关的 JOB 参数
W#16#3002	驱动工作时不能修改 MODE_IN。等待当前定位运行结束。
W#16#3003	不识别的工作模式（MODE_IN）。允许值为 1（手动），3（回参考点），4（相对值步进），5（绝对值步进）。
W#16#3004	同一时刻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	START 只允许在“绝对值步进”工作模式下。用 DIR_P 或 DIR_M 启动运行。
W#16#3006	在“绝对值步进”工作模式下，DIR_P 或 DIR_M 不允许用于线性轴。用 START 启动运行。

事件类别 故障代码	说明
W#16#3007	轴未同步。“绝对值步进”只能用于已同步的轴。
W#16#3008	清除的工作范围。返回到工作位置只允许在手动模式下。
W#16#3101	轴未参数化不能启动。通过硬件组态对“定位”模板进行参数化
W#16#3102	驱动未使能不能启动。在 SFB 中置位驱动使能 (DRV_EN=TRUE)
W#16#3103	STOP 被置位不能启动。在 SFB 中复位 STOP (STOP=FALSE)
W#16#3104	驱动当前正在执行定位运行 (WORKING=TRUE) 不能启动。等待当前定位运行结束。
W#16#3105	至少有一个错误未清除, 不能启动。消除并复位所有外部错误后再启动运行。
W#16#3201	SPEED 中速度设定错误。用数字量输出实现定位只允许“爬行”(0)和快速(1)。
W#16#3301	转换/关断距离太大。最大转换/关断距离为 $10^8$
W#16#3303	转换距离太小。转换距离必须大于/等于关断距离。
W#16#3304	关断距离太小。关断距离至少应为目标范围的一半。
W#16#3401	目标设置超出工作范围。对线性轴在步进模式下, 目标必须设定在软限位之内(包含)。
W#16#3402	目标设定错误。对旋转轴目标必须设定在 0 和旋转轴终点之间。
W#16#3403	距离设定错误。在相对值步进模式下, 移动距离必须设为正值。
W#16#3404	距离设定错误。目标的绝对坐标必须大于 $-5 \times 10^8$
W#16#3405	距离设定错误。目标的绝对坐标必须小于 $5 \times 10^8$
W#16#3406	距离设定错误。目标的绝对坐标必须在工作范围内 (+/-目标范围的一半)
W#16#3501	移动距离太大。目标坐标+当前余程必须大于/等于 $-5 \times 10^8$
W#16#3502	移动距离太大。目标坐标+当前余程必须小于/等于 $5 \times 10^8$
W#16#3503	移动距离太小。正向移动距离必须大于指定的正向关断距离
W#16#3504	移动距离太小。负向移动距离必须大于指定的负向关断距离
W#16#3505	移动距离太短或正限位已作用。最近一次的正向目标点(工作范围或位移极限)太靠近实际位置。
W#16#3506	移动距离太短或负限位已作用。最近一次的负向目标点(工作范围或位移极限)太靠近实际位置。

#### 作业故障 (JOB\_ERR = TRUE)

如果检查到故障, 则输出参数 JOB\_ERROR=TRUE。参数 JOB\_STAT 中指明故障原因。

事件类别 故障代码	说明
W#16#4001	轴未参数化。通过硬件组态对“定位”模板进行参数化
W#16#4002	定位进行中工作不允许。等待直至 WORKING = FALSE 再重复工作。
W#16#4004	不识别的工作号。检查工作 ID 再重复工作。
W#16#4101	对线性轴参考点坐标不能超出工作范围极限。

事件类别 故障代码	说明
W#16#4102 For	对线性轴参考点坐标+实际余程必须大于/等于 $-5 \times 10^8$
W#16#4103	对线性轴参考点坐标+实际余程必须小于/等于 $5 \times 10^8$
W#16#4104	对线性轴参考点坐标+到起始点的余程必须大于/等于 $-5 \times 10^8$
W#16#4105	对线性轴参考点坐标+到起始点的余程必须小于/等于 $5 \times 10^8$
W#16#4106	对旋转轴参考点坐标不能小于 0 和大于/等于旋转轴终点。

#### 外部故障（ERR）

技术电路将监控运行，移动距离和连接的外设。前提是你已经在“驱动”，“轴”和“编码器”参数屏幕表中启动了监控。

当监控单元被触发时报告外部故障。外部故障可独立于已启动的工作。你必须使用 ERR\_A = TRUE 来清除外部故障。SFB 参数 ERR（WORD）中的设置位。

监控	故障代码	ERR-WORD 中的位
零脉冲（零标记）	W#16#0004	2
移动距离	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

#### 系统故障

系统故障发生时 BIE = FALSE。当读/写背景 DB 或多次调用 SFB 错误时会引起系统故障。



## 25.3 使用 SFB 47 “COUNT” 控制计数器

### 说明

通过用户程序控制计数功能，可使用 SFB COUNT（SFB 47）

可实现如下操作：

- 通过软件门控 SW\_GATE 启动/停止计数器
- 使能/控制输出 DO
- 获取状态位 STS\_CMP, STS\_OFLW, STS\_UFLW, STS\_ZP
- 获取计数器实际值 COUNTVAL
- 读/写内部计数寄存器

### 参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
LADDR	INPUT	WORD	0	CPU 指定	W#16#0300	在“HW Config”中指定的模板 I/O 地址。 如果 I 和 O 的地址不同，按较小的地址指定。
CHANNEL	INPUT	INT	2	CPU 312C: 0 到 1 CPU 313C: 0 到 2 CPU 314C: 0 到 3	0	通道号
SW_GATE	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	启动/停止计数器的软件门控
CTRL_DO	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	使能输出
SET_DO	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	控制输出
JOB_REQ	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	工作启动（上升沿）
JOB_ID	INPUT	WORD	6	W#16#0000 作业无功能 W#16#0001 写计数值 W#16#0002 写装入值 W#16#0004 写比较值 W#16#0008 写滞后量 W#16#0010 写脉冲周期 W#16#0082 读装入值 W#16#0084	#16#0000	工作号

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
				读比较值 W#16#0088 读滞后量 W#16#0090 读脉冲周期		
JOB_VAL	INPUT	DINT	8	$-2^{31}$ 到 $+2^{31}-1$	0	写工作值
STS_GATE	OUTPUT	BOOL	12.0	TRUE/FALSE	FALSE	内部门控状态
STS_STRT	OUTPUT	BOOL	12.1	TRUE/FALSE	FALSE	硬件门控状态（启动输入）
STS_LTCH	OUTPUT	BOOL	12.2	TRUE/FALSE	FALSE	锁存输入的状态
STS_DO	OUTPUT	BOOL	12.3	TRUE/FALSE	FALSE	输出状态
STS_C_DN	OUTPUT	BOOL	12.4	TRUE/FALSE	FALSE	反向状态。 指示最近一次的计数方向。在 SFB 第一次调用后 STS_C_DN=FALSE
STS_C_UP	OUTPUT	BOOL	12.5	TRUE/FALSE	FALSE	正向状态。指示最近一次的计数方向。在 SFB 第一次调用后 STS_C_DN=TRUE
COUNTVAL	OUTPUT	DINT	14	$-2^{31}$ 到 $+2^{31}-1$	0	实际计数值
LATCHVAL	OUTPUT	DINT	18	$-2^{31}$ 到 $+2^{31}-1$	0	实际锁存值
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	作业故障
JOB_STAT	OUTPUT	WORD	24	0 到 W#16#FFFF	0	故障代码

### 注意

如果在配置中将参数“输出反应”设为“无比较”，允许下列情况：

- 输出按一般输出方式控制。
- SFB 中的输入参数 CTRL\_DO 和 SET\_DO 不激活。
- 状态位 STS\_DO 和 STS\_CMP（IDB 中的比较状态位）保持复位。

无需分配的参数（静态局部变量）：

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
STS_CMP	STATIC	BOOL	26.3	TRUE/FALSE	FALSE	比较状态位。由 RES_STS 复位。状态位 STS_CMP 指示比较的条件满足或已经满足。STS_CMP 还指示输出被置位（STS_DO = TRUE）
STS_OFLOW	STATIC	BOOL	26.5	TRUE/FALSE	FALSE	上溢状态位，由 RES_STS 复位。
STS_UFLOW	STATIC	BOOL	26.6	TRUE/FALSE	FALSE	下溢状态位，由 RES_STS 复位。
STS_ZP	STATIC	BOOL	26.7	TRUE/FALSE	FALSE	过零状态位，由 RES_STS 复位。仅用于无主计数方向的计数器。过零指示。当计数器被置 0 或装入的初始值 =0 时，该参数也被置位。

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
JOB_OVAL	STATIC	DINT	28	$-2^{31}$ up to $+2^{31}-1$	0	读作业输出值。
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 和 STS_ZP。需两次调用 SFB 来复位状态位。

### 故障信息

#### 作业故障

如果工作发生故障，则 **JOB\_ERR = TRUE**。准确的故障原因在 **JOB\_STAT** 中说明请参阅用户手册，以获取每个参数的有效值信息。

事件类别 故障代码	说明
W#16#0121	比较值大小。
W#16#0122	比较值太大。
W#16#0131	滞后量太小。
W#16#0132	滞后量太大。
W#16#0141	脉冲周期太小。
W#16#0142	脉冲周期太大。
W#16#0151	装入值太小。
W#16#0152	装入值太大。
W#16#0161	计数值太小。
W#16#0162	计数值太大。
W#16#01FF	非法工作号。

#### 系统故障

当出现系统故障时 **BIE =0**。

事件类别 故障代码	说明
W#16#8001	错误的工作模式或错误的参数。在“硬件组态中”设置正确的工作模式或使用与所设工作模式匹配的 SFB
W#16#8009	非法通道号。设定通道号 $\leq 3$ (CPU 指定值)。

## 25.4 使用 SFB 48 “FREQUENC” 控制频率测量

### 说明

通过用户程序操作频率计数器，必须使用 SFB FREQUENC（SFB48）。

可实现如下操作：

- 通过软件门控 SW\_GATE 启动/停止
- 使能/控制输出 DO
- 获取状态位 STS\_CMP，STS\_OFLOW，STS\_UFLOW
- 获取实际频率值 MEAS\_VAL
- 读/写内部频率计数寄存器

### 参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
LADDR	INPUT	WORD	0	CPU 规定	W#16#0300	在“HW Config”中指定的模板 I/O 地址。如果 I 和 O 的地址不同，按较小的地址指定。
CHANNEL	INPUT	INT	2	CPU 312C: 0 至 1 CPU 313C: 0 至 2 CPU 314C: 0 to 3	0	通道号
SW_GATE	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	启动/停止频率测量的软件门控
MAN_DO	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	使能手动输出控制
SET_DO	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	控制输出
JOB_REQ	INPUT	BOOL	4.3	TRUE/FALSE	FALSE	启动工作（上升沿）
JOB_ID	INPUT	WORD	6	W#16#0000 = 工作无功能 W#16#0001 = 写下限值 W#16#0001 = 写上限值 W#16#0004 = 写积分时间 W#16#0081 = 读下限值 W#16#0081 = 读上限值 W#16#0084 = 读积分时间	0	工作号
JOB_VAL	INPUT	DINT	8	-2 <sup>31</sup> 到+2 <sup>31</sup> -1	0	写工作值
STS_GATE	OUTPUT	BOOL	12.0	TRUE/FALSE	FALSE	内部门控状态
STS_STRT	OUTPUT	BOOL	12.1	TRUE/FALSE	FALSE	硬件门控状态（启动输入）
STS_DO	OUTPUT	BOOL	12.2	TRUE/FALSE	FALSE	输出状态
STS_C_DN	OUTPUT	BOOL	12.3	TRUE/FALSE	FALSE	反向状态。指示最近一次的计数方向。在 SFB 第一次调用后，STS_C_DN=FALSE

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
STS_C_UP	OUTPUT	BOOL	12.4	TRUE/FALSE	FALSE	正向状态。指示最近一次的计数方向。在 SFB 第一次调用后，STS_C_UP=TRUE
MEAS_VAL	OUTPUT	DINT	14	0 至 $+2^{31}-1$	0	实际频率值
COUNTVAL	OUTPUT	DINT	18	$-2^{31}$ 至 $+2^{31}-1$	0	实际计数值（每次门控打开时从 0 开始）
JOB_DONE	OUTPUT	BOOL	22.0	TRUE/FALSE	TRUE	可以启动新工作
JOB_ERR	OUTPUT	BOOL	22.1	TRUE/FALSE	FALSE	作业故障
JOB_STAT	OUTPUT	WORD	24	W#16#0000 至 W#16#FFFF	W#16#0000	故障代码

### 注意

如果在配置中将参数“输出反应”设为“无比较”，允许下列情况：

- 输出按一般输出方式控制。
- SFB 中输入参数 MAN\_DO 和 SET\_DO 不激活。
- 状态位 STS\_DO 保持复位。

### 无需分配的参数（静态局部变量）：

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
STS_CMP	STATIC	BOOL	26.3	TRUE/FALSE	FALSE	测量结束状态位。由 RES_STS 复位。每个时间周期期满后输出测量值。测量结束由状态位 STS_CMP 报告
STS_OFLW	STATIC	BOOL	26.5	TRUE/FALSE	FALSE	上溢状态位。由 RES_STS 复位。
STS_UFLW	STATIC	BOOL	26.6	TRUE/FALSE	FALSE	下溢状态位，由 RES_STS 复位。
JOB_OVAL	STATIC	DINT	28	$-2^{31}$ 至 $2^{31}-1$	0	读工作输出值。
RES_STS	STATIC	BOOL	32.2	TRUE/FALSE	FALSE	复位状态位。 复位 STS_CMP，STS_OFLW，STS_UFLW 复位状态位需两次调用 SFB。

### 作业故障

作业出现故障时 JOB\_ERR =TRUE。准确的故障原因在 JOB\_STAT 中指明。

请参阅用户手册，以获取每个参数的有效值信息。

事件类别 故障代码	说明
W#16#0221	积分时间太小。
W#16#0222	积分时间太大。
W#16#0231	频率下限太小。
W#16#0232	频率上限太大。
W#16#0241	频率上限太小。
W#16#0242	频率上限太大。
W#16#02FF	非法工作号。

## 系统故障

当出现系统故障时 BIE =0。

事件类别 故障代码	说明
W#16#8001	错误的工作模式或错误的参数。在“硬件组态中”设置正确的工作模式或使用与所设工作模式匹配的 SFB
W#16#8009	设定通道号 ≤ 3 (CPU 指定值)。

## 25.5 使用 SFB 49 “PULSE” 控制脉宽调制

### 说明

通过用户程序控制脉宽调制，可使用 SFB ANALOG（SFB 49）

可实现如下操作：

- 通过软件门控 SW\_EN 启动/停止
- 使能/控制输出 DO
- 获取状态位 STS\_EN, STS\_STRT, STS\_DO
- 输入输出值
- 读/写寄存器

### 参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
LADDR	INPUT	WORD	0	CPU 设定	W#16#0300	在“HW Config”中指定的模板 I/O 地址。如果 I 和 O 的地址不同，按较小的地址指定。
CHANNEL	INPUT	INT	2	CPU 312C: 0 至 1 CPU 313C: 0 至 2 CPU 314C: 0 至 3	0	通道号
SW_EN	INPUT	BOOL	4.0	TRUE/FALSE	FALSE	启动/停止输出的软件门控
MAN_DO	INPUT	BOOL	4.1	TRUE/FALSE	FALSE	使能手动输出控制
SET_DO	INPUT	BOOL	4.2	TRUE/FALSE	FALSE	控制输出
OUTP_VAL	INPUT	INT	6.0	以 ppm 表示: 0 至 1,000 作为 S7 模拟值: 0 至 27,648	0	缺省输出值。如果输出值 >1000 或 27648。CPU 将它限制在 1000 或 27648
JOB_REQ	INPUT	BOOL	8.0	TRUE/FALSE	FALSE	工作启动（上升沿）
JOB_ID	INPUT	WORD	10	W#16#0000=工作无功能 W#16#0001=写积分时间 W#16#0001=写接通延迟 W#16#0004=写最小脉冲周期 W#16#0081=写积分时间 W#16#0081=读接通延迟 W#16#0084=读最小脉冲周期	W#16#0000	工作号
JOB_VAL	INPUT	DINT	12	$-2^{31}$ 到 $+2^{31}-1$	0	写工作值

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
STS_EN	OUTPUT	BOOL	16.0	TRUE/FALSE	FALSE	使能状态
STS_STRT	OUTPUT	BOOL	16.1	TRUE/FALSE	FALSE	硬件门控状态（启动输入）
STS_DO	OUTPUT	BOOL	16.2	TRUE/FALSE	FALSE	输出状态
JOB_DONE	OUTPUT	BOOL	16.3	TRUE/FALSE	TRUE	可以启动新工作
JOB_ERR	OUTPUT	BOOL	16.4	TRUE/FALSE	FALSE	工作错误
JOB_STAT	OUTPUT	WORD	18	W#16#0000 to W#16#FFFF	W#16#0000	错误代码

无需分配的参数（静态局部变量）：

参数	声明	数据类型	地址 (背景 DB)	取值范围	缺省值	说明
JOB_OVAL	OUTPUT	DINT	20	$-2^{31}$ up to $2^{31}-1$	0	读工作输出值。

#### 作业故障

如果作业发生故障则 **JOB\_ERR = TRUE**。准确的故障原因在 **JOB\_STAT** 中指明。  
请参阅用户手册，以获取每个参数的有效值信息。

事件类别 故障代码	说明
W#16#0411	周期太小。
W#16#0412	周期太大。
W#16#0421	接通延迟太小。
W#16#0422	接通延迟太大。
W#16#0431	最小脉冲周期太小。
W#16#0432	最小脉冲周期太大。
W#16#04FF	非法工作号

#### 系统故障

当出现系统故障时 **BIE = 0**。

事件类别 故障代码	说明
W#16#8001	错误的工作模式或错误的参数。在“硬件组态中”设置正确的工作模式或使用与所设工作模式匹配的 SFB
W#16#8009	非法通道号。设定通道号 $\leq 3$ （CPU 指定值）。



## 25.6 使用 SFB 60 “SEND\_PTP” 发送数据 (ASCII, 3964 (R))

### 说明

你可以通过 SFB SEND\_PTP (SFB 60) 传送一个数据块。

当该块被调用并且控制输入 REQ 有上升沿时，发送操作被执行。

被传送的数据范围由 SD\_1 (数据块号和起始地址) 决定。数据块的长度由 LEN 决定。

为了使 SFB 工作，调用时必须保证 R (复位) =0。当控制输入 R 有上升沿时，当前的传送被放弃并且 SFB 被复位到初始状态。被放弃的工作以一个错误信息为结束 (STATUS 输出)。

在 LADDR 中，为子模块按照 “HW Config” 中的定义作 I/O 地址声明。

如果工作执行结束没有错误，则 DONE=1，如果有错误则 ERROR=1。

如果工作被执行且 DONE=1，这意味着：

- 当使用 ASCII 驱动器：数据被传送到通讯伙伴。不能确定所有的数据正被对方接收。
- 当使用协议 3964 (R)：数据被传送到通讯伙伴并得到对方肯定的确认。不能确定数据被传送到对方 CPU。

在 STATUS 中 CPU 用不同的事件 ID，指明一个错误或警告。

当 SFB 被复位 (R=1) 时，DONE 或 ERROR/STATUS 也有输出。

一旦错误发生二进制结果位 BR 被复位。如果块执行无误则二进制结果位状态为 “1”。

---

### 注意

SFB 不进行参数检查。如果参数化错误可导致 CPU 进入 STOP 状态。

---

### 背景 DB

SFB SEND\_PTP 工作需连接背景 DB。DB 块号通过调用传递。不允许直接对背景 DB 进行访问。

### 参数

参数	声明	数据类型	地址 (背景 DB)	取值范围	说明
REQ	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数 “请求”。由上升沿触发数据交换。
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数 “复位”。工作被取消。传送被封锁。
LADDR	INPUT	WORD	CPU 设定	W#16#03FF	在 “HW Config” 中指定的模板 I/O 地址。
DONE	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数 (该参数只在调用期间被设定) 1. FALSE 工作未启动或正在执行。 2. TRUE 工作执行没有错误。
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数 (该参数只在调用期间被设定) 工作结束无错误。

参数	声明	数据类型	地址 (背景 DB)	取值范围	说明
STATUS	OUTPUT	WORD	W#16#0000 到 W#16#FFFF	W#16#0000	状态参数（该参数只在调用期间被设定）你必须将 STATUS 拷贝到空闲数据区以显示状态。根据 ERROR 位，STATUS 有下列含义： ERROR=FALSE， STATUS=W#16#0000：无错误和警告， STATUS≠W#16#0000：提供详细信息。 ERROR=TRUE：发生错误，STATUS 提供详细的错误信息。
SD_1:	INPUT/ OUTPUT	ANY	CPU 指定	0	发送参数：输入以下数值： 存放待传送的数据的DB块号 从何处开始传送的数据字节号 例如：DB 10的字节2处-> DB10.DBB2
LEN	INPUT/ OUTPUT	INT	1 至 1024	1	要传送的数据块长度（长度在此间接设定。）

#### 数据连续性

数据连续性被限制在 206 个字节。如果你要传送的连续数据超过 206 个字节，你必须注意下列情况：

不能写发送范围 SD\_1 当前被使用的部分，除非传送已经结束，也就是当状态参数 DONE=1 时。

## 25.7 使用 SFB 61 “RCV\_PTP” 接收数据（ASCII，3964（R））

### 说明

使用 SFB RCV\_PTP（SFB 61）你可以接收数据并将它们存入数据块。

当该块被调用并且控制输入 EN\_R=1 时，接收数据就绪。EN\_R=0 可取消当前的传送。被取消的工作以一个错误信息为结束（STATUS 输出）一旦 EN\_R=0，输入立即关断。

在 RD\_1 中指定接收区（数据块号和起始地址）。在 LEN 中指定数据块长度。

为使 SFB 工作，调用时必须保证 R（复位）=0。当控制输入 R 有上升沿时，当前的传送被放弃并且 SFB 被复位到初始状态。被放弃的工作以一个错误信息为结束（STATUS 输出）。

在 LADDR 中，为子模块按照“HW Config”中的定义作 I/O 地址声明。

如果工作执行结束没有错误则 NDR=1，如果有错误则 ERROR=1。

在 STATUS 中 CPU 用不同的事件 ID，指明一个错误或警告。

当 SFB 被复位（R=1）时，NDR 或 ERROR/STATUS 也有输出（参数 LEN = 16#00）

一旦错误发生则二进制结果位 BR 被复位。如果块执行无误则二进制结果位状态为“1”。

### 注意

SFB 不进行参数检查。如果参数化错误可导致 CPU 进入 STOP 状态。

### 背景 DB

SFB RCV\_PTP 工作需连接背景 DB。DB 块号通过调用传递。不允许直接对背景 DB 进行访问。

### 参数

参数	声明	数据类型	取值范围	缺省值	说明
EN_R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“允许接收”： 接收使能
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”。工作取消。
LADDR	INPUT	WORD	CPU 设定	W#16#03FF	在“HW Config”中指定的模板 I/O 地址。
NDR	OUTPUT	BOOL	TRUE/FALSE	FALSE	工作完成无错误，数据被接收。 <ul style="list-style-type: none"> <li>FALSE 工作未启动或正在执行</li> <li>TRUE 工作执行没有错误</li> </ul>
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数（该参数只在调用期间被设定）工作结束无错误

参数	声明	数据类型	取值范围	缺省值	说明
STATUS	OUTPUT	WORD	W#16#0000 至 W#16#FFFF	W#16#0000	状态参数（该参数只在调用期间被设定） 你必须将 STATUS 拷贝到空闲数据区以显示状态。根据 ERROR 位，STATUS 有下列含义： <ul style="list-style-type: none"> <li>• ERROR=FALSE: STATUS=W#16#0000: 无错误和警告，STATUS≠W#16#0000: 提供详细信息。</li> <li>• ERROR=TRUE: 发生错误，STATUS 提供详细的错误信息。</li> </ul>
RD_1	INPUT/ OUTPUT	ANY	CPU 设定	0	接收参数：输入以下数值： 存放接收数据的DB块号。 从何处开始存放的数据字节号 例如:DB 20 的字节 5 处-> DB10.DBB2
LEN	INPUT/ OUTPUT	INT	0 至 1024	0	输出数据长度（字节数）

#### 数据连续性

数据连续性被限制在 206 个字节。如果你要传送的连续数据超过 206 个字节，你必须注意下列情况：

不能访问接收 DB 直到数据全部被接收 (NDR=TRUE)。然后封锁接收 DB (EN\_R =FALSE) 直到处理完数据。

## 25.8 使用 SFB 62“RES\_RCVB”删除接收缓冲区(ASCII, 3964(R))

### 说明

使用 SFB RES\_RECV (SFB 62) 你可以清除模板的全部输入缓冲区。所有存储的报文全部丢失。当 SFB RES\_RCVB 被调用时，行到信息帧被存储。

当该块被调用并且控制输入 REQ 有一个上升沿时，工作被激活。该工作可执行多个循环周期。

为使 SFB 工作，调用时必须保证 R (复位) =0。当控制输入 R 有上升沿时，当前的传送被放弃并且 SFB 被复位到初始状态。被放弃的工作以一个错误信息为结束 (STATUS 输出) 在 LADDR 中，为子模块按照“HW Config”中的定义作 I/O 地址声明。

如果工作执行结束没有错误，则 DONE=1，如果有错误则 ERROR=1。

在 STATUS 中 CPU 用不同的事件 ID，指明一个错误或警告。

当 SFB 被复位 (R=1) 时，DONE 或 ERROR/STATUS 也有输出。一旦错误发生则二进制结果位 BR 被复位。如果块执行无误则二进制结果位状态为“1”。

### 注意

SFB 不进行参数检查。如果参数化错误可导致 CPU 进入 STOP 状态。

### 背景 DB

SFB RES\_RCVB 工作需连接背景 DB。DB 块号通过调用传递。不允许直接对背景 DB 进行访问。

### 参数

参数	声明	数据类型	取值范围	缺省值	说明
REQ	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“请求”： 由上升沿触发工作
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”： 工作被取消
LADDR	INPUT	WORD	CPU 设定	W#16#03FF	在“HW Config”中指定的模板 I/O 地址。
DONE	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数 (该参数只在调用期间被设定) <ul style="list-style-type: none"> <li>• FALSE 工作未启动或正在执行。</li> <li>• TRUE 工作执行没有错误。</li> </ul>
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数 (该参数只在调用期间被设定) 工作结束无错误

参数	声明	数据类型	取值范围	缺省值	说明
STATUS	OUTPUT	WORD	W#16#0000 至 W#16#FFFF	W#16#0000	<p>状态参数（该参数只在调用期间被设定）</p> <p>你必须将 STATUS 拷贝到空闲数据区以显示状态。根据 ERROR 位，STATUS 有下列含义：</p> <ul style="list-style-type: none"> <li>• ERROR=FALSE: STATUS=W#16#0000: 无错误或警告，STATUS≠W#16#0000: 提供详细信息。</li> <li>• ERROR=TRUE: 发生错误，STATUS 提供详细的错误信息。</li> </ul>

## 25.9 使用 SFB 63 “SEND\_RK” 发送数据（512（R））

### 说明

你可以通过 SFB SEND\_PTP（SFB 63）传送一个数据块中的数据。

当该块被调用并且控制输入 REQ 有上升沿时，发送操作被执行。

被传送的数据范围由 SD\_1（数据块号和起始地址）决定。数据块的长度由 LEN 决定。

在 SFB 中你也要指定对方站的接收范围。CPU 将这一信息写入信息帧的头部（参见附录 F）并传送到对方。

目的地由以下参数决定，CPU 编号 R\_CPU（只与多处理器通讯有关），R\_TYPE 中的数据类型（数据块 DB 和扩展数据块（DX）），R\_DBNO 中的数据块号以及 R\_OFFSET 中决定第一个字节写到何处的偏移量。

在 R\_CF\_BYT 和 R\_CF\_BIT 中指定对方 CPU 中连接存储字节和位。

在参数 SYNC\_DB 中指定一个 DB，其中存放所有 SFB 将使用的数据，用于启动和同步时的初始化。用户程序中用于所有 SFB 的 DB 块号必须相同。

为使 SFB 工作，调用时必须保证 R（复位）=0。当控制输入 R 有上升沿时，当前的传送被放弃并且 SFB 被复位到初始状态。被放弃的工作以一个错误信息为结束（STATUS 输出）

在 LADDR 中，为子模块按照“HW Config”中的定义作 I/O 地址声明。

如果工作执行结束没有错误，则 DONE=1，如果有错误则 ERROR=1。

一旦工作被处理且 DONE =1，数据被传送到给以肯定确认的通讯伙伴并传递到对方的 CPU。

在 STATUS 中 CPU 用不同的事件 ID，指明一个错误或警告。

当 SFB 被复位（R=1）时，DONE 或 ERROR/STATUS 也有输出。

一旦错误发生则二进制结果位 BR 被复位。如果块执行无误则二进制结果位状态为“1”。

### 注意

SFB 不进行参数检查。如果参数化错误可导致 CPU 进入 STOP 状态。

### 背景 DB

SFB SEND\_RK 工作需要连接背景 DB。DB 块号通过调用传递。不允许直接对背景 DB 进行访问。

### 发送特性

当“发送数据”时，请注意以下特性：

- 使用 RK512 只能传送偶数个数据。如果指定的长度（LEN）为奇数，则在传送的数据后面将附加一个值为“0”的填充字节。
- 在 RK512 中只能使用偶数的偏移量。如果指定了一个奇数偏移量。数据在对方站中存放时将采用相邻的较小的偶数偏移量。

例如：偏移量为 7，数据存放在第 6 个字节。

### 参数

参数	声明	数据类型	取值范围	缺省值	说明
SYNC_DB	INPUT	INT	CPU 指定，不允许为 0	0	用于 RK-SFB 同步的指令数据存放的 DB 块号（最小长度=240 字节）
REQ	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“请求”：由上升沿触发工作
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”。工作被取消
LADDR	INPUT	WORD	CPU 指定	W#16#03FF	在“HW Config”中指定的模板 I/O 地址。
R_CPU	INPUT	INT	0 到 4	1	对方 CPU 的编号（仅用于多处理器操作）
R_TYPE	INPUT	CHAR	‘D’，‘X’	‘D’	对方 CPU 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 到 255	0	对方 CPU 中的数据块号
R_OFFSET	INPUT	INT	0 到 510（值允许偶数值）	0	对方 CPU 中的数据字节号
R_CF_BYT	INPUT	INT	0 到 255	255	对方 CPU 中的连接存储字节（255：意为无连接存储位）
R_CF_BIT	INPUT	INT	0 to 7	0	对方 CPU 中的连接存储位。
DONE	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数（该参数只在调用期间被设定） • FALSE 工作未启动或正在执行。 • TRUE 工作执行没有错误。
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数（该参数只在调用期间被设定） 工作结束无错误

参数	声明	数据类型	取值范围	缺省值	说明
STATUS	OUTPUT	WORD	W#16#0000 到 W#16#FFFF	W#16#0000	状态参数（该参数只在调用期间被设定） 你必须将 STATUS 拷贝到空闲数据区以显示状态。根据 ERROR 位, STATUS 有下列含义： <ul style="list-style-type: none"> <li>• ERROR=FALSE: STATUS=W#16#0000: 无错误或警告, STATUS≠W#16#0000: 提供详细信息。</li> <li>• ERROR=TRUE: 发生错误, STATUS 提供详细的错误信息。</li> </ul>
SD_1:	INPUT/ OUTPUT	ANY	CPU 指定	0	发送参数: 输入以下数值: 1. 存放待传送的数据的 DB 块号 2. 从何处开始传送的数据字节号 例如: DB 10 的字节 2 处-> DB10.DBB2
LEN	INPUT/ OUTPUT	INT	1 到 1024	1	要传送的数据块长度(长度在此间接设定。)

#### 报文帧中的声明

下表列出了 RK 512 报文帧帧头中的声明

S7 自动化系统中的源 (本地 CPU)	目的地对方 CPU	报文帧头, 字节		
		3/4 指令字节	5/6 D-DBNR/D 偏移量	7/8 数字以 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

缩写的含义:

D-DBNR	目的数据块号
D Offset	目的起始地址
DW	按字的偏移量

#### 数据连续性

数据连续性被限制在 128 个字节。如果你要传送的数据超过 128 个字节, 你必须注意下列情况:

不能写发送范围 SD\_1 当前被使用的部分, 除非传送已经结束, 也就是当状态参数 DONE=TRUE 时。



## 25.10 使用 SFB 64 “FETCH RK” 获取数据 (RK 512)

### 说明

使用 SFB FETCH\_RK (SFB 64) 从对方获取数据，并将它们存入数据块。

当该块被调用并且控制输入 REQ 有上升沿时，发送操作被执行。

获取数据的存放区由 RD\_1 (DB 块号和起始地址) 决定。数据块的长度由 LEN 块定。

在 SFB 中你也要指定对方站中要获取的数据区域。CPU 将这一信息写入信息帧头部并传送到对方。

对方数据区由以下参数决定。CPU 编号 R\_CPU (只与多处理器通讯有关)，R\_TYPE 中的数据类型 (数据块，扩展数据块，存储位，输入，输出，计数器和定时器)，R\_DBNO 中的数据块号 (只与数据块和扩展数据块有关) 以及 R\_OFFSET 中决定第一个字节从何处取偏移量在 R\_CF\_BYT 和 R\_CF\_BIT 中指定对方 CPU 中连接存储字节和连接存储位。

在参数 SYNC\_DB 中指定一个 DB，其中存放所有 SFB 将使用的数据，用于启动和同步时的初始化。用户程序中用于所有 SFB 的 DB 块号必须相同。

为使 SFB 工作，调用时必须保证 R (复位) =0。当控制输入 R 有上升沿时，当前的传送被放弃并且 SFB 被复位到初始状态。被放弃的工作以一个错误信息为结束 (STATUS 输出)。

在 LADDR 中，为子模块按照 “HW Config” 中的定义作 I/O 地址声明。

如果工作执行结束没有错误，则 DONE=1，如果有错误则 ERROR=1。

在 STATUS 中 CPU 用不同的事件 ID，指明一个错误或警告。

当 SFB 被复位 (R=1) 时，DONE 或 ERROR/STATUS 也有输出。一旦错误发生则二进制结果位 BR 被复位。如果块执行无误则二进制结果位状态为 “1”。

### 注意

SFB 不进行参数检查。如果参数化错误可导致 CPU 进入 STOP 状态。

### 注意

如果从你的 CPU 中获取数据，你必须在 CPU 中调用 SFB “SERVE\_RK”。

### 背景 DB

SFB FETCH\_RK 工作需连接背景 DB。DB 块号通过调用传递。不允许直接对背景 DB 进行访问。

### (扩展) 数据块特性

当从数据块或扩展数据块中获取数据时，请注意以下特性：

- 使用 RK512 只能传送偶数个数据。如果指定的长度 (LEN) 为奇数，则传送一个附加字节。不过在目的 DB 中总是存入正确个数的数据。
- 在 RK512 中只能使用偶数的偏移量。如果指定了一个奇数偏移量。数据在对方站中存储位置将采用相邻的较小的偶数偏移量。  
例如：偏移量为 7，数据存放在第 6 个字节。

### 定时器和计数器特性

如果从通讯伙伴中获取定时器或计数器，每个定时器或计数器都需要 2 个字节。例如：你要获取 10 个计数器，必须指定长度为 20。

### 参数

注意：在这个 SFB 中参数取值范围按德国惯例表达。

参数	声明	数据类型	取值范围	缺省值	说明
SYNC_DB	INPUT	INT	CPU 设定, 不允许位 0	0	用于“RK-SFB 同步的指令数据存放的 DB 块号（最小长度=240 字节）”
REQ	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“请求”： 由上升沿触发工作
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”。工作被取消
LADDR	INPUT	WORD	CPU 设定	W#16#03FF	在“HW Config”中指定的模板 I/O 地址。
R_CPU	INPUT	INT	0 to 4	1	对方 CPU 编号（仅用于多处理器操作）
R_TYPE	INPUT	CHAR	‘D’，‘X’， ‘M’，‘E’， ‘A’，‘Z’， ‘T’	‘D’	对方 CPU 地址类型 ‘D’：数据块 ‘X’：扩展数据块 ‘M’：存储位 ‘E’：输入 ‘A’：输出 ‘Z’：计数器 ‘T’：定时器
R_DBNO	INPUT	INT	0 到 255	0	对方 CPU 中的数据块号
R_OFFSET	INPUT	INT	请参照：数据源 （对方 CPU）FB 的参数	0	对方 CPU 中的数据字节号
R_CF_BYT	INPUT	INT	0 到 255	255	对方 CPU 中的连接存储字节（255：意为无连接存储位）
R_CF_BIT	INPUT	INT	0 到 7	0	对方 CPU 中的连接存储字节
DONE	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数（该参数只在调用期间被设定） <ul style="list-style-type: none"> <li>FALSE 工作未启动或正在执行。</li> <li>TRUE 工作执行没有错误。</li> </ul>
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数（该参数只在调用期间被设定）工作结束无错误

参数	声明	数据类型	取值范围	缺省值	说明
STATUS	OUTPUT	WORD	W#16#0000 到 W#16#FFFF	W#16#0000	状态参数（该参数只在调用期间被设定） 你必须将 STATUS 拷贝到空闲数据区以显示状态。根据 ERROR 位，STATUS 有下列含义： <ul style="list-style-type: none"> <li>• ERROR=FALSE： STATUS=W#16#0000：无错误和警告，STATUS ≠ W#16#0000：提供详细信息。</li> <li>• ERROR=TRUE： 发生错误，STATUS 提供详细的错误信息。</li> </ul>
RD_1	INPUT/ OUTPUT	ANY	CPU 指定	0	接收参数： 输入以下数值： <ul style="list-style-type: none"> <li>• 存放接收数据的 DB 块号</li> <li>• 从何处开始存放的数据字节号 例如：DB 10 的字节 2 处-&gt; DB10.DBB2</li> </ul>
LEN	INPUT/ OUTPUT	INT	1 到 1024	1	要获取的数据块长度。每个定时器和每个计数器必须占 2 个字节（长度在此被间接设定）

用于数据源（对方 CPU）的 SFB 参数

下表列出了可传送的数据类型。

R\_OFFSET 取值由对方 CPU 决定。

对方 CPU 中的源	R_TYP	R_NO	R_OFFSET（字节）
数据块	‘D’	0 – 255	0 – 510（只能为偶数值）
扩展数据块	‘X’：	0 – 255	0 – 510（只能为偶数值）
存储位	‘M’	无关	0 – 255
输入	‘E’	无关	0 – 255
输出	‘A’	无关	0– 255
计数器	‘Z’	无关	0 – 255
定时器	‘T’	无关	0 – 255.

## 信息帧中的说明

下表列出了 RK512 信息帧帧头中的声明

对方 CPU 中的源	目的地, 你的 S7 自动化系统 (本地 CPU)	信息帧头, 字节		
		3/4 指令字节	5/6 S-DBNR/S 偏移量	7/8 数字以
数据块	数据块	ED	DB/DW	字
扩展数据块	数据块	EX	DB/DW	字
存储位	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	计数器号	字
定时器	数据块	ET	定时器号	字

缩写的含义:

S-DBNO 源数据块号

S Offset 源起始地址

## 数据连续性

数据连续性被限制在 128 个字节。如果你要传送的数据超过 128 个字节, 你必须注意下列情况:

不能写获取范围 SD\_1 当前被使用的部分, 除非传送已经结束, 也许是当状态参数 DONE=1 时。

## 25.11 使用 SFB 65 “SERVE\_RK” 接收和提供数据（RK 512）

### 说明

使用 SFB SERVE\_RK（SFB 65）接收和提供数据

- 接收数据：数据存放的区域由对方在 RK512 信息帧头中指定。当通讯伙伴执行“发送数据”（SEND 工作）时，需调用该 SFB。
- 提供数据：提供的数据区由对方在 RK512 信息帧头中指定。当通讯伙伴执行“获取数据”（FETCH 工作）时，需调用该 SFB。

当该块被调用并且输入控制 EN\_R=1 时，块已准备好。当 EN\_R=0 时可取消当前的传送。被取消的工作以错误信息为结束（STATUS 输出）一旦 EN\_R=0 输入立即关断。

在参数 SYNC\_DB 中指定一个 DB，其中存放所有 SFB 将使用的数据，用于启动和同步时的初始化。用户程序中用于所有 SFB 的 DB 块号必须相同。

为使 SFB 工作，调用时必须保证 R（复位）=0。当控制输入 R 有上升沿时，当前的传送被放弃并且 SFB 被复位到初始状态。被放弃的工作以一个错误信息为结束（STATUS 输出）在 LADDR 中，为子模块按照“HW Config”中的定义作 I/O 地址声明。

如果工作执行结束没有错误则 NDR=1，如果有错误则 ERROR=1。

若 SFB 调用且 NDR=1，那么 CPU 在参数 L\_TYPE，L\_DBNO 和 L\_OFFSET 中指明数据存放位置或从何处获取。相关参数还有 L\_CF\_BYT，L\_CF\_BIT 和长度 LEN。

在 STATUS 中 CPU 用不同的事件 ID，指明一个错误或警告。

当 SFB 被复位（R=1）时，NDR 或 ERROR/STATUS 也有输出（参数 LEN = 16#00）

一旦错误发生则二进制结果位 BR 被复位。如果块执行无误则二进制结果位状态为“1”。

### 注意

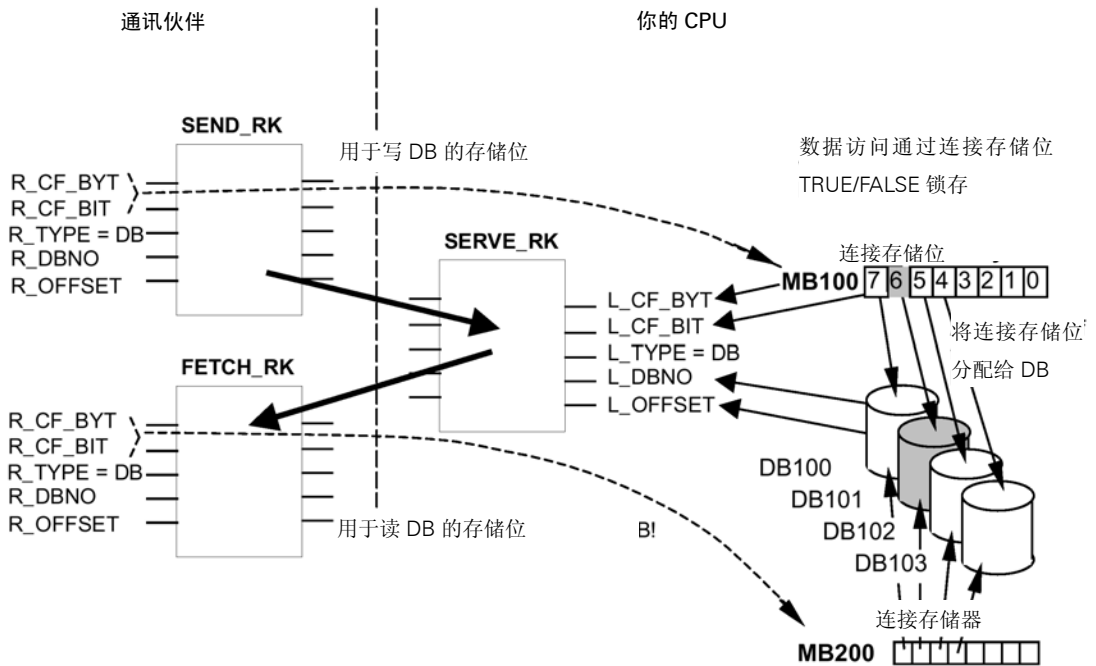
SFB 不进行参数检查。如果参数化错误可导致 CPU 进入 STOP 状态。

### 背景 DB

SFB SERVE\_RK 工作需连接背景 DB，DB 块号通过调用传递。DB 块号通过调用传递。不允许直接对背景 DB 进行访问。

### 如何使用连接存储位

你可通过连接存储位封锁或使能通讯伙伴的 SEND 和 FETCH 工作。这样你就可以防止写或读还未被处理的数据。你可以为每个工作指定一个连接存储位。



**举例：使用连接存储位的 SEND\_RK**

在本例中通讯伙伴将数据传送到你的 CPU DB 101 中

1. 在你的 CPU 中，设连接存储位 100.6=0
2. 在通讯伙伴中，为 SEND 工作指定连接存储位 100.6 (R\_CF\_BYT, R\_CF\_BIT)  
连接存储位通过 RK 512 信息帧头传到你的 CPU。

在处理工作前，CPU 检查 RK512 信息帧头中指定的连接存储位，只有当连接存储位置“0”时工作才被处理。如果连接存储位被置“1”，错误信息“32 hex”从响应信息帧中传到通讯伙伴。

数据传送到 DB101 后，你的 CPU 中的 SFB SERVE 使连接存储位=1。同时，连接存储位和字节在 SFB SERVE 上保持一个调用周期（如果 NDR =1）

3. 如果在用户程序中查看连接存储位（连接存储位 100.6 =1），可确定工作是否完成以及传送的数据是否被处理。
4. 当你在用户程序中处理完数据后，你必须复位连接存储位 100.6=0，以使你的伙伴能够再次执行工作。

## 参数

注意：在这个 SFB 中参数取值范围按德国惯例表达。

参数	声明	数据类型	取值范围	缺省值	说明
SYNC_DB	INPUT	INT	CPU 设定	0	用于存放 RK-SFB 同步指令数据的 DB 块号（最小长度=240 字节）。
EN_R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“允许接收”工作使能
R	INPUT	BOOL	TRUE/FALSE	FALSE	控制参数“复位”。工作终止
LADDR	INPUT	WORD	CPU 设定	W#16#03FF	在“HW Config”中指定的模板 I/O 地址。
NDR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数“新数据准备好”（只能在调用期间设定此参数） <ul style="list-style-type: none"> <li>FALSE：作业没有开始或正在进行。</li> <li>TRUE：作业顺利完成。</li> </ul>
ERROR	OUTPUT	BOOL	TRUE/FALSE	FALSE	状态参数（该参数只在调用期间被设定）工作结束无错误
STATUS	OUTPUT	WORD	W#16#0000 至 W#16#FFFF	W#16#0000	状态参数（该参数只在调用期间被设定）：你必须将 STATUS 拷贝到空闲数据区以显示状态。根据 ERROR 位，STATUS 有下列含义： <ul style="list-style-type: none"> <li>ERROR=FALSE： STATUS=W#16#0000：无错误和警告，STATUS ≠ W#16#0000：提供详细信息。</li> <li>ERROR=TRUE： 发生错误，STATUS 提供详细的错误信息。</li> </ul>
L_TYPE	OUTPUT	CHAR	‘D’，‘D’，‘X’，‘M’，‘E’，‘A’，‘Z’，‘T’	‘D’	接收数据： 本地 CPU 接收区的类型（只允许大写字母）： ‘D’：数据类型提供数据： 本地 CPU 源区的类型（只允许大写字母） ‘D’：数据块 ‘M’：存储位 ‘E’：输入 ‘A’：输出 ‘Z’：计数器 ‘T’：定时器 该参数只在调用期间被设定。
L_DBNO	OUTPUT	INT	CPU 设定	0	本地 CPU 中的 DB 块号。该参数只在调用期间被设定。
L_OFFSET	OUTPUT	INT	0 – 510	0	本地 CPU 的数据字节数。该参数只在调用期间被设定。

参数	声明	数据类型	取值范围	缺省值	说明
L_CF_BYT	OUTPUT	INT	0 到 255	0	本地 CPU 中的连接存储字节。该参数只在调用期间被设定（255：意为无连接存储区）
L_CF_BIT	OUTPUT	INT	0 到 7	0	本地 CPU 中的连接存储位。该参数只在调用期间被设定。
LEN	INPUT/ OUTPUT	INT	0 到 1024	0	信息帧长度，字节数（该参数只在调用期间被设定）。

### 数据连续性

数据连续性被限制在 128 个字节。如果你要传送的数据超过 128 个字节，你必须注意下列情况：

使用连接存储区功能。不要访问数据直到它们区被完全传送。（检查为该工作指定的连接存储区，如果 NDR = TRUE，则用于 SFB 调用的连接存储区被激活）。不要复位连接存储位为 FALSE 除非你已处理完数据。



## 25.12 SFB60 到 SFB65 的附加故障信息

### 故障信息

下表列出了各种事件类别的事件 ID。

SFB 参数配置错误		
事件类别 故障代码	事件	修正
W#16#0301	源/目的数据类型非法或不存在。 范围非法（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。 无效的连接存储字节或位 ID。	核对参数，必要时加以修改。 对方在信息帧头中传送了非法参数。 核对参数，必要时创建块。 从工作表中查出允许的数据类型。 对方在信息帧头中传送了非法参数
W#16#0303	无法访问某个范围	核对参数。从工作表中查出允许的起始地址和长度，或对方在信息帧头中传送了非法参数。
“工作处理错误”		
事件类别 故障代码	事件	修正
W#16#0501	因再启动或复位结束当前工作。	重复该工作。当使用 PG 再分配参数时，确保在向接口写数据前无其它工作被处理。
W#16#0502	在当前操作状态下，工作不允许（如：设备接口未分配参数）	给设备接口分配参数
W#16#050E	无效的信息帧长度 在最大允许长度内，参数中未出现信息结束 ID	信息帧长 > 1024 字节。选择较小的长度 在发送缓冲区中，在需要的位置加入信息结束 ID。
W#16#0513	数据类型错误（DB ...） 不认识的数据类型或非法数据类型（如：DE） 在 SFB 中指定的源和目的的数据类型不匹配。	从工作表中查出允许的数据类型以及它们的组合。
W#16#0515	协调存储区位编号错误。	允许的位编号 0 到 7
W#16#0516	指定的 CPU 编号太大。	允许的 CPU 编号 0, 1, 2, 3, 4
W#16#0517	传送 > 1024 字节	将工作分为几个小长度的工作。
W#16#051D	发送/接收工作被取消，由于 复位通讯块 再分配参数	重复调用通讯块。
W#16#0522	新的发送工作被启动，尽管原先的工作尚未完成。	不要启动新的发送工作，除非原先的工作以 DONE 或 ERROR 为结束。

“处理对方工作的错误” 仅用于 RK512		
事件类别 故障代码	事件	修正
W#16#0601	第一个指令字节错误（非 00 或 FFH）	对方基本的帧头结构错误如有必要可在数据链接中接入接口测试器，以证实对方错误。
W#16#0602	第三个指令字节错误（非 A, 0 或 E）	对方基本的帧头结构错误。如有必要可在数据链接中接入接口测试器，以证实对方错误。
W#16#0603	在后续信息帧中第三个指令字节错误（指令与第一个信息帧不同）	对方基本的帧头结构错误如有必要可在数据链接中接入接口测试器，以证实对方错误。
W#16#0604	第四个指令字节错误（指令字符错）	对方基本的帧头结构错误或非法的指令组合。核对允许的指令。如有必要可在数据链接中接入接口测试器，以证实对方错误。
W#16#0606	第五个指令字节错误（非法 DB 号）	从工作表中查出允许的 DB 块号，起始地址和长度
W#16#0607	Error in 5th or 6th instruction byte (start address too high)	Refer to the job tables for info on permitted DB numbers, start addresses or lengths.
W#16#0609	第九或第十个指令字节错误（协调存储位对该数据类型不允许或位编号太大）。	对方基本的帧头结构错误从工作表中查出何时协调存储位允许。
W#16#060A	第十个指令字节错误（非法 CPU 编号）	对方基本的帧头结构错误。
“发送错误”		
事件类别 故障代码	事件	修正
W#16#0701	仅用于 3964 (R): 第一次重发: 发送信息帧时检测到错误 对方以否定的确认字符 (NCC) 请求重发。	重发并不代表错误。然而它表明数据链接或对方出现故障。如果信息帧没有在最大的重发次数内被发送，则错误报告中将记录第一次出现的错误。
W#16#0702	仅用于 3964 (R): 在 STX 被传送并收到 NCC 或任何字符（除 DLE 或 STX）后，建立连接时发生错误	如有必要，在数据链接中接入接口测试器，以证实对方错误。
W#16#0703	仅用于 3964 (R): 确认超时 (QVZ): 在传送 STX 后，对方在确认延迟时间内未作出响应	对方设备太慢，或接收未就绪，或数据链接断开。如有必要，在数据链接中接入接口测试器，以证实对方错误。

“发送错误”		
事件类别 故障代码	事件	修正
W#16#0704	仅用于 3964 (R): 对方终止: 在发送过程中收到对方一个或多个字符	检查对方是否也有错误指示。因为它很可能没有收到全部发送的数据 (如由于数据链接的中断) 或对方设备错误动作。如有必要, 在数据链接中接入接口测试器, 以证实对方错误。
W#16#0705	仅用于 3964 (R): 传送过程中收到否定的确认	检查对方是否也有错误指示。因为它很可能没有收到全部发送的数据 (如由于数据链接的中断) 或对方设备错误动作。如有必要, 在数据链接中接入接口测试器, 以证实对方错误。
W#16#0706	仅用于 3964 (R): 传送结束错误: 对方最终以 NCC 或任何字符 (除 DLE) 拒绝信息帧 确认字符 (DLE) 收到太早	检查对方是否也有错误指示。因为它很可能没有收到全部发送的数据 (如由于数据链接的中断) 或对方设备错误动作。如有必要, 在数据链接中接入接口测试器, 以证实对方错误。
W#16#0707	仅用于 3964 (R): 传送结束延迟时间/监控狗响应时间超时: 在连接通过 DLE ETX 延续后, 对方未在 QVZ 内作出响应。	对方设备太慢或毁坏。如有必要, 可在数据链接中接入接口测试器, 以证实对方错误。
W#16#0708	仅用于 ASCII 驱动器: XON 等待超时。	通讯伙伴毁坏或太慢或脱机。检查通讯伙伴或必要时更改参数。
W#16#0709	仅用于 3964 (R): 无法建立连接, 允许的启动重试次数被超出	检查接口线或传送参数。同时检查对方用于 CPU 与 CP 间接收功能的参数已正确分配。
W#16#070A	仅用于 3964 (R): 无法传送数据, 允许的传送重试次数被超出。	检查接口线或传送参数。
W#16#070B	仅用于 3964 (R): 初始化冲突无法解决, 由于双方都被设为高优先级。	更改参数。
W#16#070C	仅用于 3964 (R): 初始化冲突无法解决, 由于双方都被设为低优先级。	更改参数。

“接收错误”		
事件类别 故障代码	事件	修正
W#16#0801	<ul style="list-style-type: none"> <li>仅用于 3964 (R):</li> </ul> 要求第一次重发: 收到信息帧时检查到错误, 对方 CPU 以否定确认 (NCC) 请求重发。	重发并不代表错误。然而它表明数据链接或对方出现故障。如果信息帧没有在最大的重发次数内被发送, 则错误报告中将记录第一次出现的错误。
W#16#0802	仅用于 3964 (R): 建立连接时错误 <ul style="list-style-type: none"> <li>空闲时收到一个或多个字符 (除 NCC 或 STX)</li> <li>在收到 STX 后, 对方未等待响应 DLE 就传送多个字符。</li> </ul> 在对方上电后: <ul style="list-style-type: none"> <li>在对方开机时, CPU 收到一个不明确的字符。</li> </ul>	如有必要, 可在数据链接中接入接口测试器, 以证实对方错误。
W#16#0805	仅用于 3964 (R): 逻辑接收错误: 接收到 DLE 后又收到其它字符 (除 DLE, ETX)	检查对方是否在信息帧头或数据串中重复 DLE。或者是否通过 DLE ETX 建立连接。如有必要, 可在数据链接中接入接口测试器, 以证实对方错误。
W#16#0806	字符延迟时间 (CDT) 超时: <ul style="list-style-type: none"> <li>在 CDT 内两个后续字符未被接收。</li> </ul> 仅用于 3964 (R): <ul style="list-style-type: none"> <li>1. 连接已建立, 发送 DLE 后, 在 CDT 内未收到字符。</li> </ul>	对方设备太慢或毁坏。如有必要, 可在数据链接中接入接口测试器, 以证实对方错误。
W#16#0807	非法的信息帧长度: 收到的信息帧长度为 0。	收到信息帧长度为 0 并不代表错误。检查为何对方传送的信息帧不含用户数据。
W#16#0808	仅用于 3964 (R): 块检查字符错误 BCC: 内部产生的 BCC 值与通讯链接末尾对方收到的 BCC 值不一致。	检查通讯是否严重受损。你也可查偶发错误代码。如有必要, 可在数据链接中接入接口测试器, 以证实对方错误。
W#16#0809	仅用于 3964 (R): 块重发延迟时间超时	在通讯伙伴和你的模板参数中重新声明同样的块延迟时间。如有必要, 可在数据链接中接入接口测试器, 以证实对方错误。
W#16#080A	无空闲的输入缓冲区: 无空闲的输入缓冲区用来接收。	必须更频繁地调用 SFB RCV。

“接收错误”		
事件类别 故障代码	事件	修正
W#16#080C	<ul style="list-style-type: none"> <li>• 传送错误</li> <li>• 检测到传送错误（奇偶/停止位溢出错误）</li> <li>• 仅用于 3964（R）：</li> <li>• 如果在空闲模式下收到坏字符，立即报告错误以便尽可能快地识别数据链接中的干扰。</li> <li>• 仅用于 3964（R）：</li> <li>• 如果这发生在发送或接收时，立刻启动重发。</li> </ul>	数据链接中的干扰引起信息帧重发，并因此降低用户数据通量。不检查错误的风险增加，改变你的系统结构或接线。检查通讯双方的数据链接，或波特率是否相同。奇偶性和停止位是否都已设定。
W#16#080D	BREAK： 到对方的接收链接断开。	再建立链接或对方开机。
W#16#080E	输入缓冲区溢出而流控制未使能。	在用户程序中必须更频繁地调用接收 SFB 或必须在通讯中分配流控制参数。
W#16#0810	奇偶性	检查通讯双方的数据链接，或波特率是否相同，奇偶性和停止位是否都已设定。
W#16#0811	字符帧错误	检查通讯双方的数据链接，或波特率是否相同，奇偶性和停止位是否都已设定。改变你的系统结构或接线。
W#16#0812	仅用于 ASCII 驱动器： CPU 传送 XOFF 后又收到字符。	在通讯伙伴中重新分配参数或更快地处理数据。
W#16#0814	仅用于 ASCII 驱动器： 由于操作无流控制，一个或多个信息帧丢失。	尽可能在操作中使用流控制。使用全新输入缓冲区。在基本参数中将“继续操作”设为“CPU STOP 响应”。
W#16#0816	接收到的信息帧长度超过了最大规定长度。	在对方站中做更改。

“接收响应报文帧错误或来自通讯伙伴的错误信息帧”		
事件类别 故障代码	事件	修正
W#16#0902	<ul style="list-style-type: none"> <li>• 仅用于 RK 512:</li> <li>• 对方站内存访问错误(内存不存在)。</li> <li>对方站为 SIMATIC S5:</li> <li>• 显示字范围错误</li> <li>• 数据范围不存在 (除 DB/DX)</li> <li>• 数据范围不足 (除 DB/DX)</li> </ul>	检查对方是否具备所需的数据范围。是否有足够的空间或检查调用的 SFB 参数。检查 SFB 指定的长度。
W#16#0903	<p>仅用于 RK 512: 对方站 DB/DX 访问错误 (DB/DX 不存在或太短)。</p> <p>对方站为 SIMATIC S5:</p> <ul style="list-style-type: none"> <li>• DB/DX 不存在</li> <li>• DB/DX 太短</li> <li>• 非法 DB/DX 块号</li> </ul> <p>FETCH 工作超出了允许的源范围。</p>	检查对方是否具备所需的数据范围。是否有足够的空间或检查调用的 SFB 参数。检查 SFB 指定的长度。
W#16#0904	仅用于 RK 512: 对方报告“工作类型不允许”。	对方错误动作因为 CPU 没有输出系统指令。
	<p>仅用于 RK 512: 对方错误或 SIMATIC S5 错误:</p> <ul style="list-style-type: none"> <li>• 源/目的类型不允许</li> <li>• 对方设备内存错误</li> <li>• 对方 CP/CPU 通讯错误</li> <li>• 对方 PLC 处于 STOP 模式</li> </ul>	检查对方是否能够传送所需数据。检查对方硬件结构。将对方 PLC 置于 RUN 模式。
W#16#0908	仅用于 RK 512: 对方发现同步错误: 信息帧序列被破坏。	当启动你自己或对方 PLC 时发生错误这是系统的正常启动行动。不需修正。在操作过程中也可能表现为前面错误的结果, 否则你认为是对方错误。
W#16#0909	仅用于 RK 512: 对方中的 DB/DX 被协调存储位封锁。	在对方程序中: 在处理了最后传送的数据后复位协调存储位。 程序: 重复工作!
W#16#090A	仅用于 RK 512: 对方发现信息帧头错误: 第三个指令字节错	检查错误原因是否由于干扰或对方错误动作。可在数据链接中接入接口测试器以证实这一点。
W#16#090C	仅用于 RK 512: 对方发现信息帧长度错误。	检查错误原因是否由于干扰或对方错误动作。可在数据链接中接入接口测试器以证实这一点。
W#16#090D	• 仅用于 RK 512: 对方至今未再启动。	再启动对方 PLC 或将模式选择开关置于 RUN。
W#16#090E	仅用于 RK 512: 在响应信息帧中收到不认识的错误号。	检查错误原因是否由于干扰或对方错误动作。可在数据链接中接入接口测试器以证实这一点。

“CPU 在对方的响应信息帧中检查到错误”		
事件类别 故障代码	事件	修正
W#16#0A02	仅用于 RK 512: 接收到的响应信息帧结构错误 (第一个字节非 00 或 FF)	如有必要, 可在数据链接中接入接口测试器, 以证实对方错误。
W#16#0A03	仅用于 RK 512: 接收到响应信息帧包含数据太多或不足。	如有必要, 可在数据链接中接入接口测试器, 以证实对方错误。
W#16#0A05	仅用于 RK 512: 在监控时间内未收到对方的响应信息帧。	对方设备是否太慢? 这一错的经常发生也会表现为前面错误的结果。例如在发送了一个 FETCH 信息帧后, 可表现为接收过程错误 (事件类别 8) 原因: 由于干扰或监控狗超时会收不到响应信息帧。如果对方在对最后收到的 FETCH 信息帧做出响应前再启动也会造成错误发生。
“警告”		
事件类别 故障代码	事件	修正
W#16#0B01	输入缓冲区超过 2/3 的容量已被装满	为防止输入缓冲区溢出, 更频繁地调用接收块。

## 26 用于 H CPU 的 SFC

### 26.1 使用 SFC 90 “H\_CTRL” 在 H 系统中控制操作

#### 说明

使用 SFC 90 “H\_CTRL”，你可对 H 系统实现以下影响：

- 禁止备用设备接入主 CPU。该禁止保持有效直到在 SFC 90 “H\_CTRL” 中取消设定。任何来自备用 CPU 的与主站连接请求都被锁存。
- 禁止主 CPU 上的更新。该禁止保持有效直到在 SFC 90 “H\_CTRL” 中取消设定。任何来自备用 CPU 的更新请求都被锁存。



#### 告诫

如果你已经禁止更新但未连接，硬件系统仍能象以前一样决定连接状态。请注意当主 CPU 处于连接时，它不处理任何取消/插入的中断，站故障/恢复的中断或机架故障/恢复的中断。

- 从循环自检中取消某个测试部分，再加入或立即执行。

#### 注意：

如果在冗余系统中使用 CPU 414-4H 或 417-4H，请注意：如果禁止某部分超过 24 小时，CPU 会进入 STOP 模式。在冗余系统中，应用规则指明某些测试必须在 24 小时内完成。

下表列出了输入参数 MODE 和 SUBMODE 允许的组合。

工作	MODE 输入	SUBMODE 输入
禁止接入	3	0
允许接入	4	0
禁止更新	1	0
允许更新	2	0
从循环自检中取消 SUBMODE 指定的测试部分。 一个测试部分只能被取消一次。	20	0.1, ...5
在循环自检中加入 SUBMODE 指定的测试部分。 只有被取消了的测试部分才能再被加入。	21	0.1, ...5
立即执行 SUBMODE 指定的测试部分。 该测试部分不能已被取消。	22	0.1, ...5

下表中列出了循环自检中测试部分与 SUBMODE 输入值的分配。（只有当输入 MODE 取值 20，21 和 22 时）



SUBMODE 取值	对应的测试部分
0	SP7 – ASIC – Test
1	编码区测试
2	数据区测试
3	操作系统编码校验和测试
4	编码块校验和测试
5	在冗余操作中比较数字，时间，标志和数据块。

### SFC 如何工作

SFC 90 “H\_CTRL” 是一个异步 SFC，也就是说它的执行需要多次调用。

当 REQ=1 时执行 SFC 90 的调用，从而启动工作。

如果工作被立即执行，则 SFC 在输出参数 BUSY 中返回 0 值。一个长期的惯例测试的初始化以 SFC 的首次调用为结束（BUSY=0），即使该测试要经历多个循环周期（RET\_VAL=W#16#0001，MODE=22）如果 BUSY=1，该工作仍在进行（又见参数 REQ，RET\_VAL 的含义以及异步 SFC 的 BUSY）

### 作业标识

输入参数 MODE 和 SUBMODE 指定某个工作。如果该工作还未完成，则 SFC 调用将顺延。

### 参数

参数	声明	数据类型	存储区域	说明
REQ	INPUT	BOOL	I, Q, M, D, L, 常数	电平触发控制参数 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	子作业
RET_VAL	OUTPUT	INT	I, Q, M, D, L	如果功能执行时发生故障，返回值中存放故障代码。确保每当块执行完毕检查 RET_VAL。
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1：作业未完成。

## 故障信息

故障代码 (W#16#...)	说明
0000	作业执行无误
7000	REQ = 0 首次调用：作业未激活；BUSY=0。
7001	REQ = 1 首次调用：作业已启动；BUSY=1。
7002	顺延调用（REQ 无关）。已激活的作业仍在运行；BUSY=1。
0001	<ul style="list-style-type: none"> <li>当 MODE=1：更新已被禁止。</li> <li>当 MODE=3：接入已被禁止。</li> <li>当 MODE=22：测试部分已在运行，不能再启动。</li> </ul>
8082	<ul style="list-style-type: none"> <li>当 MODE=1：更新已激活，不能被禁止。</li> <li>当 MODE=3：接入已激活，不能被禁止。</li> <li>当 MODE=20：指定的测试部分已从循环自检中取消。</li> <li>当 MODE=21：指定的测试部分还未从循环自检中取消。</li> <li>当 MODE=22：指定的测试部分不能执行，因为它已被从循环自检中取消。</li> </ul>
8090	输入参数 MODE 中数值无效。
8091	输入参数 SUBMODE 中数值无效。

## SFC 90 使用举例

使用 SFC 90，当 CPU 需要最大资源时，确保接入和更新不要频繁启动。

为达此目的，你可以在大量处理工作前，在主 CPU 中加入下列程序：

- Call SFC 90, MODE = 3 并且 SUBMODE =0（禁止接入）
- Call SFC 90, MODE = 1 并且 SUBMODE =0（禁止更新）

在大量处理工作之后，在主 CPU 中加入下列程序：

- Call SFC 90, MODE = 4 并且 SUBMODE =0（允许接入）
- Call SFC 90, MODE = 2 并且 SUBMODE =0（允许更新）



## 27 集成功能（用于带集成 I/O 的 CPU）

### 27.1 SFB 29（HS\_COUNT）

#### 说明

使用 SFB 29 “HS\_COUNT”（计数器），你对带集成 I/O 的 CPU 内部集成的计数器功能产生以下影响：

- 设置和输入初始值。
- 选择和设置比较值。
- 使能计数器。
- 使能数字量输出。
- 读当前计数值和比较值。
- 查询当前计数值和比较值的关系。

#### 更多信息

与集成计数功能有关的每个参数的含义以及 CPU 的硬件输入和输出详情，可在 *S7-300 可编程控制器集成功能手册* 中查阅。

#### 参数

参数	声明	数据类型	存储区域	说明
PRES_COUNT	INPUT	DINT	I, Q, M, D, L, 常数	计数器初始值
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
COMP_B	EN_COUNT	INPUT	I, Q, M, D, L	计数器使能
EN_DO	INPUT	BOOL	I, Q, M, D, L, 常数	数字量输出使能
SET_COUNT	INPUT	BOOL	I, Q, M, D, L, 常数	设置初始值 PRES_COUNT
SET_COMP_A	INPUT	BOOL	I, Q, M, D, L, 常数	设置比较值 COMP_A
SET_COMP_B	INPUT	BOOL	I, Q, M, D, L, 常数	设置比较值 COMP_B
COUNT	OUTPUT	DINT	I, Q, M, D, L	计数器实际值
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

## 27.2 SFB 30（FREQ\_MES）

### 说明

使用 SFB 30 “FREQ\_MES”（频率测量计），你对带集成 I/O 的 CPU 内部集成的频率测量计功能产生以下影响：

- 选择和设置比较值。
- 输出测量的频率。
- 读当前的比较值
- 查询测量的频率与比较值的关系。

### 更多信息

与集成的频率测量计功能有关的每个参数的含义以及 CPU 的硬件输入和输出详情，可在 *S7-300 可编程控制器集成功能手册* 中查阅。

参数	声明	数据类	存储区域	说明
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	设置新比较值 U_LIMIT
SET_L_LIMIT	INPUT	BOOL	I, Q, M, D, L, 常数	设置新比较值 L_LIMIT
FREQ	OUTPUT	DINT	I, Q, M, D, L	测量的频率 MHz
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 >= U_LIMIT

## 27.3 SFB 38（HSC\_A\_B）

### 说明

使用 SFB 38（HSC\_A\_B），你对带集成 I/O 的 CPU 内部集成的 A/B 计数器功能产生以下影响：

- 指定和选用初始值
- 指定和设置比较值
- 使能计数器
- 使能数字量输出
- 读当前计数值和比较值
- 查询当前计数值和比较值的关系

SFB 38（HSC\_A\_B）从用户程序中的集成功能背景 DB 中读或写数据。A/B 计数器包括 A 和 B 两个计数器，它们可同时计数并且相互独立（可向上和向下计数）。

计数器功能相同；计数脉冲频率可达 10 kHz。

### 更多信息

与集成的 A/B 计数器功能有关的每个参数的含义以及 CPU 的硬件输入和输出详情，可在 *S7-300 可编程控制器集成功能 CPU 312 IFM/314 IFM* 手册中查阅。

### 参数

参数	声明	数据类型	存储区域	说明
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	计数器使能
EN	INPUT	BOOL	I, Q, M, D, L, 常数	数字量输出使能
SET_COMP	INPUT	BOOL	I, Q, M, D, L, 常数	设置比较值 COMP
COUNT	OUTPUT	DINT	I, Q, M, D, L	计数器实际值
COMP	OUTPUT	DINT	I, Q, M, D, L	当前比较值 COMP
ENO	OUTPUT	BOOL	I, Q, M, D, L	错误监控： 1: 执行无错误 0: 执行有错误

## 27.4 SFB 39（POS）

### 说明

使用 SFB 39（POS），你可对带集成 I/O 的 CPU 内部集成的定位功能产生以下影响：

- 同步
- 点动
- 定位

用于集成的定位功能的 SFB 39（POS）从用户程序中的集成功能背景 DB 中读或写数据。集成的定位功能获取不对称 24 V 增量编码器信号。最大频率 10 kHz。它控制高速/爬行运行或通过 CPU 314 IFM 特定的集成输出口实现变频输出（控制定位）。

### 更多信息

与集成的定位功能有关的每个参数的含义以及 CPU 的硬件输入和输出详情，可在 *S7-300 可编程控制器集成功能 CPU 312 IFM/314 IFM* 手册中查阅。

### 参数

参数	声明	数据类	存储区域	说明
EN	INPUT	BOOL	I, Q, M, D, L, 常数	数字量输入使能
DEST_VAL	INPUT	DINT	I, Q, M, D, L, 常数	集成定位功能的终点位置
REF_VAL	INPUT	DINT	I, Q, M, D, L, 常数	用于同步的参考点
SWITCH_OFF_DIFF	INPUT	WORD	I, Q, M, D, L, 常数	关断距离（关断点与终点位置的差距）
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	控制移动的最大模拟量值
POS_MODE1, POS_MODE2	INPUT	BOOL	I, Q, M, D, L, 常数	启动和执行爬行
POS_STRT	INPUT	BOOL	I, Q, M, D, L, 常数	上升沿启动定位功能
SET_POS	INPUT	BOOL	I, Q, M, D, L, 常数	上升沿触发输入参数 REF_VAL 取新的实际值
ENO	OUTPUT	BOOL	I, Q, M, D, L	错误监控： 1：执行无错误 0：执行有错误
ACTUAL_POS	OUTPUT	DINT	I, Q, M, D, L	当前实际值
POS_READY （状态信息）	OUTPUT	BOOL	I, Q, M, D, L	POS_READY=1，定位/爬行结束
REF_VALID （状态信息）	OUTPUT	BOOL	I, Q, M, D, L	参考点开关是否到达
POS_VALID （状态信息）	OUTPUT	BOOL	I, Q, M, D, L	轴的实际位置与集成功能的实际位置同步

## 28 汇编技术

### 28.1 SFB 63 (AB\_CALL)

#### 说明

SFC 63 (AB\_调用) 调用一个汇编代码块。

汇编代码块是由 C 语言编写或由汇编器编写然后编译的逻辑块。

#### 应用

汇编代码块只能用于 CPU 614。

#### 进一步的信息

每个 SFC 63 的参数的含义在 CPU 614 的有条文献中有详细解释。另外有单独的用于汇编代码块的编程指南。

#### 参数

参数	声明	数据类型	存储区域	说明
AB_NUMBER	INPUT	WORD	I, Q, M, D, L, 常数	用于要调用的汇编代码块的若干位
CALL_REASON	INPUT	WORD	I, Q, M, D, L, 常数	调用 SFC 的 OB 或对 DB 指针的评估 (参数 DB_NUMBER) 或对调试器的激活。
DB_NUMBER	INPUT	WORD	I, Q, M, D, L, 常数	DP 指针号
RET_VAL	OUTPUT	WORD	I, Q, M, D, L	SFC 的返回值。





## 29 诊断数据

### 29.1 诊断数据的结构概览

#### 系统数据的 0 号和 1 号数据纪录

一个模块的诊断数据位于系统数据区的 0 号和 1 号数据纪录中（参见数据纪录的读和写）。

- 0 号数据纪录包含诊断数据的 4 个字节，记述信号模块的电流状态。
- 1 号数据纪录包含
  - 诊断数据的 4 个字节以及
  - 模块特定的诊断数据。

#### 诊断数据的结构和内容

本节对诊断数据的结构和每个字节的内容进行说明。每当有一个错误发生时，相应的位便被置为“1”。

### 29.2 诊断数据

#### 诊断数据的结构和内容

字节	位	含义	注释
0	0	模块故障	
	1	内部错误	
	2	外部错误	
	3	通道错误	
	4	无外部辅助电压	
	5	无前连接器	
	6	无参数分配	
	7	模块内错误参数	
1	0 至 3	模块类别	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

字节	位	含义	注释
	4	存在通道信息	
	5	存在用户信息	
	6	来自替代值的诊断中断	
	7	保留	
2	0	无存储卡或存储卡错误	
	1	通讯问题	
	2	模式	0 运行 1 停止
	3	循环监控已响应	
	4	内部模块电源故障	
	5	电池耗尽	
	6	整个电池后备故障	
	7	保留	
3	0	扩展机架故障	
	1	处理器故障	
	2	EPROM 故障	
	3	RAM 故障	
	4	ADC/DAC 故障	
	5	保险断	
	6	硬件中断丢失	
	7	保留	
4	0 至 6	通道类型	B#16#70 数字输入 B#16#72 数字输出 B#16#71 模拟输入 B#16#73 模拟输出 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 保留 B#16#7D 保留 B#16#7E US300 B#16#7F 保留
	7	存在进一步的通道类型吗?	0 无 1 有
5	0 至 7	一个模块上每通道的诊断位输出数	每通道的诊断位数依字节范围 (8 位) 而舍入

字节	位	含义	注释
6	0 至 7	一个模块上某种通道的数量	若一个模块上存在不同类型的通道,则此结构在数据纪录 1 中从第 4 字节为每个通道而重复。
7	0	通道错误, 通道 0/通道组 0	
	1	通道错误, 通道 1/通道组 1	
	2	通道错误, 通道 2/通道组 2	
	3	通道错误, 通道 3/通道组 3	通道错误矢量(矢量长度取决于通道数量且依字节范围而舍入)
	4	通道错误, 通道 4/通道组 4	
	5	通道错误, 通道 5/通道组 5	
	6	通道错误, 通道 6/通道组 6	
	7	通道错误, 通道 7/通道组 7	
...	-	通道特定错误(见通道特定的诊断数据的结构)	

## 29.3 通道特定的诊断数据的结构

### 通道特定错误

从紧接着通道错误矢量之后的字节开始, 模块上每一通道的通道特定错误都被指明。下列各表显示了对于不同类型通道的通道特定错误的结构。这些位含义为:

- 1 = 有错
- 0 = 无错

### 模拟输入通道

用于模拟输入通道的诊断字节:

位	含义	注释
0	组态/参数分配错误	可由 SFC 52 发信号通知且 EVENTN = W#16#8x50
1	共模错误	可由 SFC 52 发信号通知且 EVENTN = W#16#8x51
2	P 短路	可由 SFC 52 发信号通知且 EVENTN = W#16#8x52
3	M 短路	可由 SFC 52 发信号通知且 EVENTN = W#16#8x53
4	断线	可由 SFC 52 发信号通知且 EVENTN = W#16#8x54
5	参考通道错误	可由 SFC 52 发信号通知且 EVENTN = W#16#8x55
6	电流低于测量范围	可由 SFC 52 发信号通知且 EVENTN = W#16#8x56
7	电流高于测量范围	可由 SFC 52 发信号通知且 EVENTN = W#16#8x57

模拟输出通道

用于模拟输出通道的诊断字节:

位	含义	注释
0	组态/参数分配错误	可由 SFC 52 发信号通知且 EVENTN = W#16#8x60
1	共模错误	可由 SFC 52 发信号通知且 EVENTN = W#16#8x61
2	P 短路	可由 SFC 52 发信号通知且 EVENTN = W#16#8x62
3	M 短路	可由 SFC 52 发信号通知且 EVENTN = W#16#8x63
4	断线	可由 SFC 52 发信号通知且 EVENTN = W#16#8x64
5	0	保留
6	无负载电压	可由 SFC 52 发信号通知且 EVENTN = W#16#8x66
7	0	保留

数字输入通道

用于数字输入通道的诊断字节:

位	含义	注释
0	组态/参数分配错误	可由 SFC 52 发信号通知且 EVENTN = W#16#8x70
1	对地短路	可由 SFC 52 发信号通知且 EVENTN = W#16#8x71
2	P 短路 (传感器)	可由 SFC 52 发信号通知且 EVENTN = W#16#8x72
3	M 短路	可由 SFC 52 发信号通知且 EVENTN = W#16#8x73
4	断线	可由 SFC 52 发信号通知且 EVENTN = W#16#8x74
5	无传感器电源	可由 SFC 52 发信号通知且 EVENTN = W#16#8x75
6	0	保留
7	0	保留

数字输出通道

用于数字输出通道的诊断字节:

位	含义	注释
0	组态/参数分配错误	可由 SFC 52 发信号通知且 EVENTN = W#16#8x80
1	对地短路	可由 SFC 52 发信号通知且 EVENTN = W#16#8x81
2	P 短路	可由 SFC 52 发信号通知且 EVENTN = W#16#8x82
3	M 短路	可由 SFC 52 发信号通知且 EVENTN = W#16#8x83
4	断线	可由 SFC 52 发信号通知且 EVENTN = W#16#8x84
5	保险断	可由 SFC 52 发信号通知且 EVENTN = W#16#8x86
6	无负载电压	可由 SFC 52 发信号通知且 EVENTN = W#16#8x86
7	超温	可由 SFC 52 发信号通知且 EVENTN = W#16#8x87

## 30 系统状态列表（SSL）

### 30.1 系统状态列表（SSL）概述

这一节只对与以下情况有关的局部系统状态列表进行说明：

- 中央处理器（CPU）
  - 具有非特定模式列表的模块（例如，SSL-IDs, W#16#00B1, W#16#00B2, W#16#00B3）
- 特殊模块的部分列表，例如，在特殊模块说明中的 CP 和 FM。

#### 定义：系统状态列表

系统状态列表（SSL）描述了可编程逻辑控制器的当前状态。SSL 的内容只能使用信息功能读取，但是不能被修改。局部列表是虚拟的列表。换句话说，它们只能在 CPU 有特殊要求的情况下由系统生成。

你只能使用 SFC 51 “RDSYSST” 读取一个系统状态列表。

#### 目录

系统状态列表包含有以下信息：

- 系统数据
- CPU 中的模块状态信息
- 模块诊断信息
- 诊断缓冲区

#### 系统数据

系统数据定义或分配了 CPU 的特有的数据。它们提供了以下信息：

- CPU 的配置信息
- 优先级的状态
- 通讯

#### 模块的状态信息

模块的状态信息描述了系统诊断功能当前的监控状态。

#### 模块的诊断信息

CPU 将诊断信息直接保存在具有诊断能力的模块中。

#### 诊断缓冲区

诊断缓冲区中包含了所有发生过的诊断信息。

## 30.2 局部系统状态列表的结构

### 基础

你可以使用 SFC 51 “RDSYSST” 读取局部列表和局部列表摘要。你可以通过使用参数 SSL\_ID 和 INDEX 来定义你要读取的信息。

### 结构

一个局部系统状态列表包括：

- 一个标题
- 记录的数据

### 标题

一个局部系统状态的标题包括：

- SSL-ID
- 索引
- 局部系统状态列表的记录数据字节长度
- 局部系统状态列表所记录数据的个数

### 索引

利用一些局部系统状态列表或局部列表摘要可以选取一个目标类型 ID 或一个目标代码索引用于此目的。如果不需要此类信息，其内容无关紧要。

### 数据记录

在局部列表中的数据记录有其特定长度。这取决于局部列表中的信息。在数据记录中的数据消息的具体内容还要看详细的局部列表。

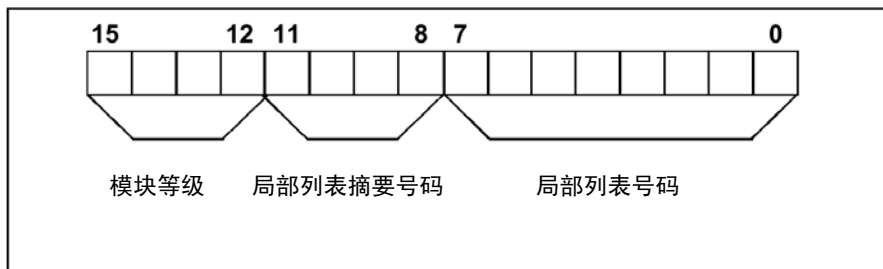
## 30.3 SSL - ID

### SSL-ID

每一个局部系统状态列表都有一个号码。你可以从中输出一个完整的局部列表或者一个摘要。通过该号码可能出现的局部列表摘要被预先确定和识别。SSL-ID 包含有局部列表号码，局部列表摘要号码，模块等级。

### 结构

SSL-ID 为一字长，以下是 SSL-ID 中每一位的含义：



*SCL\_ID 结构*

### 模块等级

模块等级举例：

模块等级	代码（二进制）
CPU	0000
IM	0100
FM	1000
CP	1100

### 局部列表摘要号码

局部列表摘要号码和它们的含义要取决于它的所在的系统状态列表的详细内容。利用局部列表摘要号码，你可以指明你要读取局部列表的哪一个部分。

### 局部列表号码

利用局部列表号码，你可以指定你要读取系统状态列表中的哪一个局部列表。



## 30.4 可能出现的局部系统状态列表

### 子集

任意一个模块只能有一个包括所有可能出现的局部列表的子集，哪一个局部列表可用要看具体的模块。

### 可能出现的 SSL 局部列表

下表列出了所有可能出现的局部列表及其在 SSL-ID 中的号码。

局部列表	SSL-ID
模块标识	W#16#xy11
CPU 特性	W#16#xy12
用户存储区	W#16#xy13
系统存储区	W#16#xy14
块类型	W#16#xy15
模块 LED 灯状态	W#16#xy19
中断状态	W#16#xy22
过程映像区和 OB 块间的参数分配	W#16#xy25
通讯状态数据	W#16#xy32
冗余 CPU 组信息	W#16#xy71
模块 LED 灯状态	W#16#xy74
在冗余系统中控制 DP 从站	W#16#xy75
模块状态信息	W#16#xy91
机架/站状态信息	W#16#xy92
扩展 DP 主站系统信息	W#16#xy95
CPU 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) 物理地址	W#16#00B2
模块诊断信息 (数据记录 1) 逻辑地址	W#16#00B3
DP 从站诊断数据	W#16#00B4

## 30.5 SSL - ID W#16#xy11 - 模块标识

### 目的

如果你利用 SSL-ID W#16#xy11 读取系统状态列表，你可以得到该模块的标识。

### 首部

系统状态列表 SSL-ID W#16#xy11 首部结构如下：

内容	含义
SSLID	局部列表摘要 SSL-ID W#16#0001: 一个单独的标识数据记录
INDEX	详细数据记录编号 W#16#0001: 模块标识 W#16#0006:基本硬件标识 W#16#0007: 基本固件标识
LENTHDR	W#16#001C: 一个数据记录为 14 个字长 (28 个字节)
N_DR	数据记录号码

### 数据记录

一个系统状态列表数据记录 SSL-ID W#16#xy11 有以下结构：

名称	长度	含义
Index	1 个字	标识数据记录索引号
MIFB	20 个字节	当 INDEX W#16#0007: 保留 当 INDEX W#16#0001, W#16#0006: 模块定货号 字符串包含 19 个字符和 1 个空格 (20H); 例如 CPU 314 “6ES7 314-0AE01-0AB0”
BGTYP	1 个字	保留
Ausbg1	1 个字	当 Index W#16#0001: 模块的版本 当 Index W#16#0006, W#16#0007: “V” 和版本号中的第 1 个数字。
Ausbg2	1 个字	当 Index W#16#0001: 保留 当 Index W#16#0006, W#16#0007: 剩下的版本号。

## 30.6 SSL - ID W#16#xy12 – CPU 特性

### 目的

根据所使用的硬件 CPU 模块具有不同的特性，每一种特性用一个 ID 表示。如果你利用 SSL-ID W#16#xy12 读取局部列表，你可以得到该模块的特性信息。

### 首部

局部列表 SSL-ID W#16#xy12 首部结构如下：

内容	含义
SSLID	局部列表摘要 SSL-ID W#16#0012: 所有特性 W#16#0112: 组特性 你可以在 INDEX 参数中定义组。 W#16#0F12: 局部列表头文件信息
INDEX	组 W#16#0000: MC7 控制单元 W#16#0100: 时钟系统 W#16#0200: 系统应答 W#16#0300: MC7 语言描述 W#16#0400: SFC 块的利用率
LENTHDR	W#16#0002: 一个数据记录为 1 个字长 (2 个字节)
N_DR	数据记录号码

### 数据记录

一个局部列表数据记录 SSLID W#16#xy12 为一个字长。每一种特性都有一个标识，一个特性标识为一个字长。

### 特性标识

下表列出了所有特性标识

标识	含义
W#16#0000 - 00FF	MC7 程序单元
W#16#0001	MC7 程序代码
W#16#0002	MC7 编译器
W#16#0100 - 01FF	时间系统
W#16#0101	1 ms 分辨率
W#16#0102	10 ms 分辨率
W#16#0103	无实时时钟
W#16#0104	BCD 时间日期格式
W#16#0105	所有的时间日期功能 (设定时间日期、设定并读取时间日期、时间日期同步: 主站和从站的时间日期)

标识	含义
W#16#0200 - 02FF	系统响应
W#16#0201	多处理器模式
W#16#202	允许冷启动、暖启动和热启动
W#16#203	允许冷启动和热启动
W#16#204	允许暖启动和热启动
W#16#205 Only	允许暖启动
W#16#0206	RUN 状态下允许重新组态新的分布式 I/O
W#16#0207	H-CPU 在单一操作模式: RUN 状态下允许重新组态新的分布式 I/O
W#16#0300 - 03FF	MC7 CPU 的语言说明
W#16#0301	保留
W#16#0302	所有 32 位定点数指令
W#16#0303	所有浮点数指令
W#16#0304	sin, asin, cos, acos, tan, atan, sqr, sqrt, ln, exp
W#16#0305	(ENT, PUSH, POP, LEAVE) 指令中的累加器 3/累加器 4
W#16#0306	主控继电器指令
W#16#0307	利用相应指令保留地址寄存器 1
W#16#0308	利用相应指令保留地址寄存器 2
W#16#0309	交叉寻址操作
W#16#030A	内部寻址操作
W#16#030B	所有用于位存储 (M) 的间接寻址指令
W#16#030C	所有用于数据块存储 (DB) 的间接寻址指令
W#16#030D	所有用于数据块存储 (DI) 的间接寻址指令
W#16#030E	所有用于局部数据 (L) 的间接寻址指令
W#16#030F	所有用于 FC 参数传递的指令
W#16#0310	用于过程映像输入 (I) 的存储位边沿指令
W#16#0311	用于过程映像输出 (Q) 的存储位边沿指令
W#16#0312	用于位存储器 (M) 的存储位边沿指令
W#16#0313	用于数据块 (DB) 的存储位边沿指令
W#16#0314	用于数据块 (DI) 的存储位边沿指令
W#16#0315	用于局部数据 (L) 的存储位边沿指令
W#16#0316	FC 位的动态赋值
W#16#0317	相应指令的动态局部数据区
W#16#0318	保留
W#16#0319	保留
W#16#0401	SFC 87 “C_DIAG” 可用
W#16#0402	SFC 88 “C_CNTRL” 可用

## 30.7 SL - ID W#16#xy13 – 存储区域

### 目的

如果读取了部分表 SSLID W#16#xy13，就会得到模板存储区域的信息。

### 首部

部分表 SSL-ID W#16#xy13 的首部结构如下：

内容	含义
SSLID	部分表的 SSLID 摘要 W#16#0113: 一个存储区域的数据记录 用 INDEX 参数指定存储区域
INDEX	指定存储区域（只对 SSL-ID W#16#0113） W#16#0001: 工作存储器
LENTHDR	W#16#0024:一条数据记录有 18 个字（36 个字节）
N_DR	数据记录的数目

### 数据记录

部分表 SSL-ID W#16#xy13 的数据记录结构如下：

名称	长度	含义
Index	1 个字	存储区域标签 W#16#0001:工作存储器
Code	1 个字	存储器类型: W#16#0001: 易失性存储器 (RAM) W#16#0002: 永久性存储器 (FEPRM) W#16#0003: 混合存储器 (RAM + FEPRM)
Size	2 个字	选定存储器的总容量 (存储区 1 与存储区 2 的总和)
Mode	1 个字	存储器的逻辑模式 位 0: 易失性存储区域 位 1: 永久性存储区域 位 2: 混合存储区域 对于工作存储器 位 3: 代码和数据分离 位 4: 代码和数据集中
Granu	1 个字	总为 0
Ber1	2 个字	易失性存储区域的字节数。
Belegt1	2 个字	正在使用的易失性存储区域的容量
Block1	2 个字	易失性存储区域中最大的自由块 如果为 0: 没有可得到的信息或者不能确定。
Ber2	2 个字	永久性存储区域的字节数
Belegt2	2 个字	正在使用的永久性存储器的容量
Block2	2 个字	永久性存储区域中最大的自由块 如果为 0: 没有可得到的信息或者不能确定。

## 30.8 SSL - ID W#16#xy14 – 系统区域

### 目的

如果读取了部分表 SSL-ID W#16#xy14，就会得到模板系统区域的信息。

### 首部

部分表 SSL-ID W#16#xy14 的首部结构如下：

内容	含义
SSLID	部分表的 SSL-ID 摘要 W#16#0014: 模板的所有系统区域 W#16#0F14: 只有部分表首部的信息
INDEX	无关
LENTHDR	W#16#0008: 一条数据记录有 4 个字 (8 个字节)
N_DR	数据记录的数目 最少指定 9 条数据记录。如果选择的目标区域太小，SFC51 将不提供数据记录。

### 数据记录

部分表 SSL-ID W#16#xy14 的数据记录结构如下：

名称	长度	含义
Index	1 个字	系统区域标签： W#16#0001: PII (字节数) W#16#0002: PIQ (字节数) W#16#0003: 存储器 (位数) <b>注意：</b> 此标签只能由 CPU 提供，特征数可以以一个字表示。 如果你的 CPU 不提供此数据，你必须估测标签 W#16#0008。 W#16#0004: 定时器 (数目) W#16#0005: 计数器 (数目) W#16#0006: 位于逻辑地址区域中的字节数 W#16#0007: 本地数据 (用字节表示的 CPU 的整个本地数据区域) <b>注意：</b> 此标签只能由 CPU 提供，特征数可以以一个字表示。 如果你的 CPU 不提供此数据，你就必须估测标签 W#16#0009。 W#16#0008: 存储器 (字节数) W#16#0009: 本地数据 (用 K 字节表示的 CPU 整个本地数据区域)
Code	1 个字	存储器类型 W#16#0001: 易失性存储器 (RAM) W#16#0002: 永久性存储器 (FEPROM) W#16#0003: 混合存储器 (RAM 和 FEPROM)

名称	长度	含义
Quantity	1 个字	系统区域元素的数目
Reman	1 个字	记忆元素的数目

## 30.9 SSL - ID W#16#xy15 – 块类型

### 目的

如果读取了部分表 SSL-ID W#16#xy15，就可得到模板上块类型的信息。

### 首部

部分表 SSL-ID W#16#xy15 的首部结构如下：

内容	含义
SSLID	部分表 SSL-ID 摘要 W#16#0015: 模板上所有块类型的数据记录
INDEX	无关
LENTHDR	W#16#0006: 一条数据记录有 5 个字 (10 个字节)
N_DR	数据记录的数目

### 数据记录

部分表 SSL-ID W#16#xy15 的数据记录结构如下：

名称	长度	含义
Index	1 个字	块类型 W#16#0800: OB W#16#0A00: DB W#16#0B00: SDB W#16#0C00: FC W#16#0E00: FB
MaxAnz	1 个字	同类型块的最大数目 OB: 一个 CPU OB 可能的最大数 DB: DB 可能的最大数 (包括 DB0) SDB: SDB 可能的最大数 (包括 SDB2) FC 和 FB: 可装载的块可能的最大数
MaxLng	1 个字	装载总容量, 以 K 字节表示。
Maxabl	2 个字	工作存储器中块的最大长度, 以字节表示。

## 30.10 SSL - ID W#16#xy19 – 模板 LED 的状态

### 目的

如果读取了部分表 SSLID W#16#xy19，就可得到模板 LED 状态的信息。

### 注意

如果想读取 H CPU 中的部分表 W#16#16#xy19，注意只有在非冗余 H 操作模下方有可能。

### 首部

部分表 W#16#xy19 的首部结构如下：

内容	含义
SSLID	部分表 SSLID W#16#0019 所有 LED 的状态 W#16#0119 一个 LED 的状态
LENTHDR	W#16#0004:一条数据记录有 2 个字 (4 个字节)
N_DR	数据记录的数目

### 数据记录

部分表 SSLID W#16#xy19 的数据记录结构如下：

名称	长度	含义
Index	1 个字	LED ID (只与 SSLID W#16#0119 有关) W#16#0001: SF (组故障) W#16#0002: INTF (内部故障) W#16#0003: EXTF (外部故障) 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 (电池故障/母线电压过载, 短路) W#16#0009: USR (用户定义) W#16#000A: USR1 (用户定义) W#16#000B: BUS1F (接口 1 总线故障) W#16#000C: BUS2F (接口 2 总线故障) W#16#000D: REDF (冗余故障) W#16#000E: MSTR (主站) W#16#000F: RACK0 (机架 0) W#16#0010: RACK1 (机架 1) W#16#0011: RACK2 (机架 2) W#16#0012: IFM1F (接口模板 1 接口故障) W#16#0013: IFM2F (接口模板 2 接口故障)



名称	长度	含义
led_on	1 个字节	LED 的状态： 0: 灭 1: 亮
led_blink	1 个字节	LED 的闪烁状态： 0: 不闪烁 1: 正常闪烁 (2 Hz) 2: 慢闪 (0.5 Hz)

### 30.11 SSL - ID W#16#xy1C – 组件识别

#### 目的

如果读取了部分表 SSL-ID W#16#xy1C，就可以识别 CPU 或者 PLC。

#### 首部

部分表 W#16#xy1C 的首部结构如下：

内容	含义	
SSLID	部分表 SSL-ID 摘要	
	W#16#001C:	所有组件识别
	W#16#011C:	一个组件识别
	W#16#0F1C:	只是 SSL 部分表首部信息
INDEX	对于部分系统信息表 SSL ID W#16#011C 的组件识别	
	W#16#0001:	PLC 的名称
	W#16#0002:	CPU 的名称
	W#16#0003:	CPU 的出厂标记
	W#16#0004:	版权
	W#16#0005:	CPU 的系列号
	W#16#0006:	为操作系统保留
	W#16#0007:	CPU 类型名称
	W#16#0008:	存储卡的系列号 提供不能插存储卡的模板，没有数据记录
LENTHDR	W#16#0022:	一条数据记录有 17 个字 (34 个字节)
N_DR	数据记录的数目	

## 数据记录

部分表 SSL-ID W#16#xy1C 的数据记录结构如下：

- INDEX = W#16#0001

名称	长度	含义
Index	1 个字	组件识别：W#16#0001
Name	12 个字	PLC 的名称 (最多 24 个字符；用短名时，其余空位用 B#16#00 填充)
Res	4 个字	保留

- INDEX = W#16#0002

名称	长度	含义
Index	1 个字	组件识别：W#16#0002
Name	12 个字	CPU 的名称 (最多 24 个字符；用短名时，其余空位用 B#16#00 填充)
Res	4 个字	保留

- INDEX = W#16#0003

名称	长度	含义
Index	1 个字	组件识别：W#16#0003
Tag	16 个字	(最多 32 个字符；使用一个短的出厂标志时，其余空位用 B#16#00 填充)

- INDEX = W#16#0004

名称	长度	含义
Index	1 个字	元件识别：W#16#0004
Copyright	13 个字	连续字符序列“纯正的西门子设备”
Res	3 个字	保留

- INDEX = W#16#0005

名称	长度	含义
Index	1 个字	元件 ID：W#16#0005
Serialn	12 个字	CPU 的列号；最多 24 个字符的字符串。不足 24 个字符由 B#16#00 填充。 注意：序列号是世界范围内 SIMATIC 元件的唯一标志，与 CPU 硬件永远联系在一起，即固件升级时序列号不改变。
Res	4 个字	保留

## 系统状态列表 (SSL)

---

- INDEX = W#16#0006  
相应的数据记录为操作系统保留。
- INDEX = W#16#0007

名称	长度	含义
Index	1 个字	元件 ID: W#16#0007
Cputypname	16 个字	CPU 类型名称, 最多 32 个字符。不足 32 个字符由 B#16#00 填空。

- INDEX = W#16#0008

名称	长度	含义
Index	1 个字	元件 ID: W#16#0008
sn_mc/mmc	16 个字	存储卡/微存储卡的系列号; 最多 32 个字符。不足 32 个字符由 B#16#00 填空。 <ul style="list-style-type: none"> <li>• 西门子系列号: 只有系列号, 没有标签</li> <li>• S7 微存储器卡的产品系列号 (PSN): “MMC” 加系列号 (PSN)</li> <li>• S7 存储器卡的系列号: “MC” 机系列号</li> </ul> 如果没有安装存储器卡, “MMC” 或 “MC” 后没有字符。

## 30.12 SSL - ID W#16#xy22 – 中断状态

### 目的

如果读部分表 SSL-ID W#16#xy22，就会得到正在执行的中断与由模板产生的中断的状态信息。

### 首部

部分表 SSL-ID W#16#xy22 的首部结构如下：

内容	含义
SSL-ID	部分系统状态表 SSL-ID W#16#0222 指定中断的数据记录。 用 INDEX 参数指定中断 (OB 号)。
INDEX	OB 号或者优先级 (SSL ID W#16#0222) W#16#0000: 正常循环 W#16#000A: 时间日期中断 W#16#0014: 延时中断 W#16#001E: 监控中断 W#16#0028: 硬件中断 W#16#0032 DP 中断 W#16#003C 多处理器或时钟同步中断 W#16#0048: 冗余故障中断 (只对于 S7-400H 系统) W#16#0050: 异步故障中断 W#16#005A: 后台 W#16#0064 启动 W#16#0078: 同步故障中断
LENTHDR	W#16#001C: 一条数据记录有 14 个字 (28 个字节)
N_DR	数据记录的数目

数据记录

部分表 SSL-ID W#16#xy22 的数据记录结构如下:

名称	长度	含义
Info	10 个字	<p>相应 OB 的启动信息，下列除外：</p> <ul style="list-style-type: none"> <li>对于 OB1，当前最小（字节 8 和 9）和最大（字节 10 和 11）的循环时间（时基：ms，字节计数由 0 开始）。</li> <li>当延时中断请求有效时，字节 8 ... 11（字节计数由 0 开始）包含距中断执行完毕的时间（ms）。</li> <li>最小（字节 8 和 9）和最大（字节 10 和 11）循环时间可以设置的在 OB80 中读出（时基：ms，字节计数由 0 开始）。</li> <li>对于没有当前信息的故障中断</li> <li>对于中断，状态信息包含中断源的当前参数配置。</li> <li>对于同步故障，如果中断 OB 还未执行，B#16#7F 作为优先级进入，否则，取上次优先级。</li> <li>如果一个 OB 对应多个启动事件而这些事件当读取信息时都还没有发生，则事件号 W#16#xyzz 与 x 一起返回：事件种类：zz：当读取信息时定义的最低的组合号，y：未定义。否则使用最近的启动事件号。</li> </ul>
al 1	1 个字	<p>执行标识</p> <p>位 0： 中断事件禁止/使能由参数设置 = 0： 使能 = 1： 禁止</p> <p>位 1： 中断事件由 SFC 39 “DIS_IRT” 禁止。 = 0： 使能 = 1： 禁止</p> <p>位 2 = 1： 中断源激活（时间中断任务生成，时间日期中断 OB 已开始执行，延时中断 OB 已开始执行，循环中断 OB 计时开始）</p> <p>位 4： 中断 OB = 0： 未装载 = 1： 已装载</p> <p>位 5： 中断 OB 被测试与安装功能禁止 = 1： 禁止</p> <p>位 6： 进入诊断缓存区 = 1： 禁止</p>
al 2	1 个字	<p>OB 未装载/禁止时的响应</p> <p>位 0 = 1： 禁止中断源</p> <p>位 1 = 1： 产生中断事件故障</p> <p>位 2 = 1： CPU 转换到 STOP 模式</p> <p>位 3 = 1： 只是放弃中断</p>
al 3	2 个字	<p>被测试与安装功能放弃：</p> <p>位号 x 置位表示：事件号 x，如果大于最低的事件号，此事件相应的 OB 被测试与安装功能放弃。</p>

## 30.13 SSL ID W#16#xy25 – 给 OB 分配过程映像区

### 目的

由部分表 SSL ID W#16#xy25 可以了解 OB 过程映像区的配置。

它提供关于下面过程映像区的信息

- 分配给指定 OB 由系统刷新的过程映像区
- 分配给指定时钟同步中断 OB (OB61~64) 的过程映像区, 这些过程映像区通过调用 SFC 126 “SYNC\_PI” 和 127 “SYNC\_PO” 刷新。  
DP 主系统和时钟同步中断 OB 的配置在 SSL W#16#xy95 中。

### 首部

部分表 SSL ID W#16#xy25 的首部结构如下:

内容	含义
SSL ID	部分表 SSL ID 摘要 <ul style="list-style-type: none"> <li>• W#16#0025: CPU 中 OB 的过程映像区的配置</li> <li>• W#16#0125: 相应 OB 过程映像区的配置 在参数 INDEX 中指定过程映像区 ID。</li> <li>• W#16#0225: 对应过程映像区的 OB 的配置 在参数 INDEX 中指定 OB 号 注意: 时钟同步中断 OB (OB61~64) 是唯一一类可以分配多个过程映像区的 OB。</li> <li>• W#16#0F25: 只对于 SSL 部分表的首部信息</li> </ul>
INDEX	<ul style="list-style-type: none"> <li>• 对于 SSL ID W#16#0025: 无关</li> <li>• 对于 SSL ID W#16#0125: 过程映像区 ID</li> <li>• 对于 SSL ID W#16#0225: OB 号</li> <li>• 对于 SSL ID W#16#0F25: 无关</li> </ul>
LENTHDR	W#16#0004: 一条数据记录有 2 个字 (4 个字节)
N_DR	数据记录的数目

### 数据记录

SSL ID W#16#xy25 局部列表摘要数据结构如下:

名称	长度	含义
tpa_nr	1 个字节	局部过程映像 ID
tpa_use	1 个字节	过程映像区和 OB 间的类型分配: <ul style="list-style-type: none"> <li>• 位 0 = 1: 输入过程映像区被分配到特定的 OB 中用于刷新系统。</li> <li>• 位 1 = 1: 输出过程映像区被分配到特定的 OB 中用于刷新系统。</li> <li>• 位 2 = 1: 输入过程映像区被分配到特定的时钟同步中断 OB 中, 它可以通过调用 SFC 126 “SYNC_PI” 进行刷新。</li> <li>• Bit 3 = 1: 输出过程映像区被分配到特定的时钟同步中断 OB 中, 它可以通过调用 SFC 126 “SYNC_PO” 进行刷新。</li> <li>• 位 4 到 7: 0</li> </ul>
ob_nr	1 个字节	OB 号
res	1 个字节	保留

局部列表摘要

- SSL ID = W#16#0025: 你在配置时在 OB 中分配的所有过程映像区的数据记录都会按升序返回, 如果不将过程映像区分配给 OB。则 ob\_nr 参数为 0, 在这种情况下, 不返回过程映像区数据记录。
- SSL ID = W#16#0125: 如果你在配置时已在 OB 中分配了地址过程映像区将有一个数据记录返回。如果没有分配 OB, 将没有返回的数据记录。

注意

OB 永远是被分配给过程映像区 0, 这样, 当你需要过程映像区 0 的信息时, 你就可以得到相应的数据记录。

- SSL ID = W#16#0225 局部列表摘要:  
当你将所有过程映像区分配到被调用的 OB 中会有数据记录返回。如果没有将过程映像区分配到被调用的 OB 中就没有数据记录返回。

注意

时钟同步中断 OB 可以被分配多重过程映像区。如果这样的话, 会有多个数据记录被返回。

- SSL ID = W#16#0F25 局部列表摘要:  
数据记录的最大号码被返回。

数据记录含义举例

SFC 51 调用参数	返回变量	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	返回的数据记录。 输入/输出过程映像区 8 被分配到 OB21 用于系统升级程序映像区。
SZL_ID = W#16#0125, INDEX = W#16#0009	-	无数据记录项目, 因而过程映像区 9 没有被分配到任何 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#3	返回两个数据记录。 输入/输出映像区 10 和 16 被分配给 OB61 他们可以通过在 OB61 中调用 SFC126, SFC127 进行升级。
SZL_ID = W#16#0225, INDEX = W#16#0001	tpa_nr = B#16#00, tpa_use = B#16#03, ob_nr = B#16#01	一个数据记录被返回。 输入/输出映像区 0 被分配给 OB1。可通过系统刷新。

## 30.14 SSL - ID W#16#xy32 – 通讯状态数据

### 目的

如果你读取 SSL-ID W#16#xy32 你可以获得模块通讯状态数据。

### 首部

SSL-ID W#16#xy32 局部列表首部结构如下：

内容	含义
SSLID	局部列表 SSL ID W#16#0132 CPU 通讯单元的状态数据（总是一个数据记录）。你可以通过 INDEX 参数定义 CPU 通讯单元。 W#16#0232 一个通讯单元状态数据（在冗余运行模式下，n 条数据记录被返回，其中 n 是指冗余系统中冗余 CPU 的个数）。你可以通过 INDEX 参数定义两个 CPU 通讯单元。
INDEX	通讯单元 <ul style="list-style-type: none"> <li>• 对于 SSL ID W#16#0132:                W#16#0005 诊断                W#16#0008 时钟系统                W#16#000B 时钟系统                W#16#000C 时钟系统</li> <li>• 对于 SSL ID W#16#0232:                W#16#0004 CPU 保护等级，操作设定和 ID 版本</li> </ul>
LENTHDR	W#16#0028:一个数据记录有 20 个字长（40 个字节）
N_DR	数据记录的个数

### 数据记录

一个局部列表 SSL-ID W#16#0132 总是 20 个字长，数据记录有不同的内容。内容依靠 INDEX 参数，换句话说，它是属于 CPU 通讯数据记录中的哪一个部分。



## 30.15 SSL - ID W#16#0132 索引号为 W#16#0005 的局部列表摘要数据记录

### 内容

SSL-ID W#16#0132 索引号为 W#16#0005 的局部列表摘要数据记录包含有模块诊断状态信息。

### 数据记录

SSL-ID W#16#0132 索引号为 W#16#0005 的局部列表摘要数据记录结构如下：

名称	长度	含义
Index	1 个字	W#16#0005:诊断
Erw	1 个字	扩展功能 0: 无 1: 有
Send	1 个字	自动发送 0: 无 1: 有
Moeg	1 个字	发送当前允许的用户自定义诊断信息 0: 无 1: 有
Res	16 个字	保留

## 30.16 SSL - ID W#16#0132 索引号为 W#16#0008 的局部列表摘要数据记录

### 内容

SSL-ID W#16#0132 索引号为 W#16#0008 的局部列表摘要包含有关于模块时间系统状态的信息。

### 数据记录

SSL-ID W#16#01032 索引号为 W#16#0008 的局部列表摘要数据记录的结构如下：

名称	长度	含义
Index	1 个字	W#16#0008:时间系统状态
Zykl	1 个字	同步结构的循环扫描时间
Korr	1 个字	时间修正因子
clock 0	1 个字	运行时间计时器 0: 时间为小时
clock 1	1 个字	运行时间计时器 1: 时间为小时
clock 2	1 个字	运行时间计时器 2: 时间为小时
clock 3	1 个字	运行时间计时器 3: 时间为小时
clock 4	1 个字	运行时间计时器 4: 时间为小时
clock 5	1 个字	运行时间计时器 5: 时间为小时
clock 6	1 个字	运行时间计时器 6: 时间为小时
clock 7	1 个字	运行时间计时器 7: 时间为小时
Time	4 个字	当前的日期和时间 (格式 DATE_AND_TIME)
bszl_0 to bszl_1	2 个字节	运行时间计量的启动 (位=1: 运行时间计量器启动)
bszl_0	1 个字节	位 x: 运行时间计时器 X, $0 \leq x \leq 7$
bszl_1	1 个字节	保留
bszü_0 to bszü_1	2 个字节	运行时间计时器溢出 (位=1: 溢出)
bszü_0	1 个字节	位 x: 运行时间计时器 X, $0 \leq x \leq 7$
bszü_1	1 个字节	保留
Status	1 个字	时间状态 (参照位的分配, 看下面)
Res	3 个字节	保留
status_valid	1 个字节	变量状态的合法性: B#16#01: 状态有效

系统状态列表 (SSL)

状态

位	默认值	描述
15	0	修正值符号 (0: 正值, 1: 负值)
14 至 10	00000	修正值 此参数允许将结构中的基本时间修正为本地时间: 本地时间=基本时间±修正值* 0.5 h 这个修正值要考虑时区和夏令时与冬令时的区别。
9	0	保留
8	0	保留
7	0	时间声明 这个参数指明了是否下次的调整也包括从夏令时到冬令时的转换。 (0: 不做调整, 1: 做调整)。
6	0	夏令时/冬令时指示器此参数指出当前的时间使用的是夏令时还是冬令时。 (0: 冬令时, 1: 夏令时)
5	0	此参数不能用于 S7
4 至 3	00	时基 这个参数指出了传输时基。 (00: 0.001 s, 01: 0.01 s, 10: 0.1 s, 11: 1 s)
2	0	此参数不能用于 S7。
1	0	此参数不能用于 S7
0	0	同步失败 此参数指出是否传输时间在有效范围内。 (0: 同步失败, 1: 同步产生)  注意: 如果有外部时钟同步在 CPU 中为该位赋值也是有意义的。

## 30.17 SSL - ID W#16#0132 索引号为 W#16#000B 的局部列表摘要数据记录

### 内容

SSL ID W#16#0132 索引号为 W#16#000B 局部列表摘要包含有 32 位运行时间计时器 0 ... 7 模块的状态信息。

### 注意

SSL ID W#16#0132 索引号为 W#16#0008 局部列表摘要以 16 位模式显示运行时间。

SSL ID W#16#0132 索引号 W#16#0008 局部列表摘要开发 CPU 的 16 位运行时间计时器。

### 数据记录

SSL ID W#16#0132 索引号为 W#16#000B 局部列表摘要数据记录结构如下：

名称	长度	含义
Index	1 个字	W#16#000C: 时间系统状态
Bszl_0	1 个字节	位 x: 运行时间计时器 x 状态, $0 \leq x \leq 7$ (位=1: 运行时间计时器忙)
Bszl_1	1 个字节	保留
Bszü_0	1 个字节	位 x: 运行时间计时器 x 溢出 $0 \leq x \leq 7$ (位=1: 溢出)
Bszü_1	1 个字节	保留
Clock 8	2 个字	运行时间计时器 8: 时间为小时
Clock 9	2 个字	运行时间计时器 9: 时间为小时
Clock 10	2 个字	运行时间计时器 10: 时间为小时
Clock 11	2 个字	运行时间计时器 11: 时间为小时
Clock 12	2 个字	运行时间计时器 12: 时间为小时
Clock 13	2 个字	运行时间计时器 13: 时间为小时
Clock 14	2 个字	运行时间计时器 14: 时间为小时
clock 15	2 个字	运行时间计时器 15: 时间为小时
Res	1 个字	保留

## 30.18 SSL - ID W#16#0132 索引号为 W#16#000C 的局部列表摘要数据记录

### 内容

SSL ID W#16#0132 索引号为 W#16#000C 的局部列表摘要包含有 32 位运行时间计时器 8 ... 15 的模块信息。

### 数据记录

SSL ID W#16#0132 索引号为 W#16#000C 的局部列表摘要数据记录的结构如下：

名称	长度	含义
Index	1 个字	<b>W#16#000C</b> : 时间系统状态
Bszl_0	1 个字节	位 x: 运行时间计时器 (8+x) 状态, $0 \leq x \leq 7$ (位=1: 运行时间计时器忙)
Bszl_1	1 个字节	保留
Bszü_0	1 个字节	位 x: 运行时间计时器 (8+x) 溢出 $0 \leq x \leq 7$ (位=1: 溢出)
Bszü_1	1 个字节	保留
Clock 8	2 个字	运行时间计时器 8: 时间为小时
Clock 9	2 个字	运行时间计时器 9: 时间为小时
Clock 10	2 个字	运行时间计时器 10: 时间为小时
Clock 11	2 个字节	运行时间计时器 11: 时间为小时
Clock 12	2 个字	运行时间计时器 12: 时间为小时
Clock 13	2 个字	运行时间计时器 13: 时间为小时
Clock 14	2 个字	运行时间计时器 14: 时间为小时
clock 15	2 个字	运行时间计时器 15: 时间为小时
Res	1 个字	保留

## 30.19 SSL - ID W#16#0232 索引号为 W#16#0004 的局部列表摘要数据记录

### 内容

SSL-ID W#16#0232 索引号为 W#16#0004 的局部列表摘要数据记录包含有关于 CPU 保护等级和运行模式开关设置，以及硬件设置和用户程序的版本识别。

### 数据记录

SSL-ID W#16#0232 索引号为 W#16#0004 的局部列表摘要数据记录结构如下：

名称	长度	含义
Index	1 个字	<ul style="list-style-type: none"> <li>字节 1:</li> <li>B#16#04:CPU 保护等级和运行模式开关设置以及版本识别。</li> <li>字节 0:</li> <li>标准 CPU: B#16#00</li> <li>冗余 CPU: 位 0 到 2: 机架号</li> <li>位 3: 0 = 备用 CPU, 1=主 CPU</li> <li>位 4 到 7: 1111</li> </ul>
sch_schal	1 个字	利用模式选择器 (1, 2, 3) 设置保护等级
sch_par	1 个字	设置保护等级为以下参数值 (0, 1, 2, 3; 0: 无密码, 保护等级无效)
sch_rel	1 个字	CPU 有效的保护等级
bart_sch	1 个字	模式选择设置 (1: 运行, 2: 运行停止, 3: 停机, 4: 复位, 0: 尚未定义)
anl_sch	1 个字	启动开关设置 (1: 冷启 2: 暖启, 0: 尚未定义, 不存在或未定义)
ken_rel	1 个字	允许版本的 ID 号 (0: 不允许)
ken_ver1_hw	1 个字	硬件组态版本 ID1
ken_ver2_hw	1 个字	硬件组态版本 ID2
ken_ver1_awp	1 个字	用户程序版本 ID1
ken_ver2_awp	1 个字	用户程序版本 ID2
Res	8 个字	保留

## 30.20 SSL - ID W#16#xy71 – 冗余 CPU 组信息

### 目的

SSL-ID W#16#xy71 局部列表摘要包含有冗余系统当前的状态信息。

### 首部

SSL-ID W#16#xy71 局部列表首部结构如下：

内容	含义
SSLID	局部列表摘要的 SSLID: W#16#0071: 冗余系统当前的状态信息 W#16#0F71: 只有 SSL 局部列表首部信息
INDEX	W#16#0000
LENTHDR	W#16#0010: 一个数据记录为 8 个字长 (16 个字节)
N_DR	W#16#0001: 数据记录编号

### 数据记录

一个局部列表摘要数据记录 ID W#16#xy71 有以下结构：

内容	长度	含义
Redinf	2 个字节	冗余信息 W#16#0011: 单个冗余 CPU W#16#0012: 1/2 冗余系统
Mwstat1	1 个字节	状态字节 1 位 0: 保留 位 1: 保留 位 2: 保留 位 3: 保留 位 4: 位于机架 0 上的 CPU 的冗余状态 =0: 备用 CPU =1: 主 CPU 位 5: 位于机架 1 上的 CPU 的冗余状态 =0: 备用 CPU =1: 主 CPU 位 6: 保留 位 7: 保留
Mwstat2	1 个字节	状态字节 2 位 0: 同步连接 01 状态: CPU 与 CPU1 之间的同步 0: 不能 1: 能

内容	长度	含义
		位 1: 0 位 2: 0 位 3: 保留 位 4: =0: 机架 0 上没有插入 CPU =1: 机架 1 上没有插有 CPU (在冗余模式下: 位 4=0) 位 5: =0: 机架 0 上没有插入 CPU =1: 机架 1 上插有 CPU (在冗余模式下: 位 5=0) 位 6: 保留 位 7: 从最后一次重新使能开始进行主-备 转换 =0: 不转换 =1: 转换
Hsfinfo	2 个字节	SFC 90 “H_CTRL” 信息字 位 0: =0: 不允许重新使能 =1: 激活重新使能 位 1: =0: 刷新被激活的备份 =1: 刷新被取消的备份 位 2: =0: 整理为被激活的备份 =1: 整理为被取消的备份 位 3: 保留 位 4: 保留 位 5: 保留 位 6: 保留 位 7: =1: 请求刷新升级 位 8: =1: 不请求刷新升级
Samfehl	2 个字节	保留
Bz_cpu_0	2 个字节	机架 0 上的 CPU 模式 W#16#0001: 停机 (升级) W#16#0002: 停机 (存储器复位) W#16#0003: 停机 (自动初始化) W#16#0004: 停机 (内部) W#16#0005: 启动 (冷启) W#16#0006: 启动 (暖启) W#16#0007: 启动 (热启) W#16#0008: 运行 (单独模式) W#16#0009: RUN-R (冗余模式) W#16#000A: 保持 W#16#000B: 连接 W#16#000C: 升级 W#16#000D: 有效的



内容	长度	含义
		W#16#000E: 自动测试 W#16#000F: 没有上电
Bz_cpu_1	2 个字节	在机架 1 上的 CPU 模式 (关于 bz_cpu_0 的值)
Bz_cpu_2	2 个字节	保留
Cpu_valid	1 个字节	bz_cpu_0 和 bz_cpu_1 的有效变量 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 都有的
Reserve	1 个字节	保留

## 30.21 SSL-ID W#16#xy74 - 模板 LED 灯状态

### 目的

如果你读取局部列表 SSL-ID W#16#xy74, 利用标准 CPU 和冗余 CPU, 你可以获得模块 LED 灯状态。

如果冗余 CPU 处于非冗余模式, 你可以得到 CPU 地址的 LED 灯状态。如果冗余 CPU 处于冗余模式, LED 灯状态用于显示所有冗余 CPU。

### 首部

SSL-ID W#16#xy74 局部列表首部结构如下:

内容	含义
SSLID	局部状态列表摘要 SSL-ID W#16#0174 LED 灯状态 利用 INDEX 参数选择 LED INDEX LED ID 号 (与 SSL-ID W#16#0174 有关) W#16#0001: SF (组错误) W#16#0002: INTF (内部错误) W#16#0003:EXTF (外部错误) W#16#0004:RUN W#16#0005: STOP W#16#0006: FRCE (强制) W#16#0007: CRST (冷启动) W#16#0008:BAF (电池负压/过压, 总线上电池短路) W#16#0009: USR (用户自定义)

内容	含义
	W#16#000A: USR1 (用户自定义) W#16#000B: BUS1F (接口 1 总线错误) W#16#000C: BUS2F (接口 2 总线错误) W#16#000D: REDF (冗余错误) W#16#000E: MSTR (主) W#16#000F: RACK0 (机架 0) W#16#0010: RACK1 (机架 1) W#16#0011: RACK2 (机架 2) W#16#0012: IFM1F (接口模板 1 接口错误) W#16#0013: IFM2F (接口模板 2 接口错误)
LENTHDR	W#16#0004: 一个 2 字长 (4 个字节) 数据记录
N_DR	数据记录代码

## 数据记录

SSL-ID W#16#0074 局部列表摘要数据记录结构如下:

名称	长度	含义
cpu_led_ID	1 个字	<ul style="list-style-type: none"> <li>• 字节 0</li> <li>• 标准 CPU: B#16#00</li> <li>• H-CPU: 位 0 到 2: 机架号</li> </ul> 位 3: 0=备用 CPU, 1=主 CPU 位 4 到 7: 1111 字节 1: LED 灯 ID
led_on	1 个字节	LED 灯状态: 0: 关闭 1: 开启
led_blink	1 个字节	LED 灯闪烁状态: 0: 不闪烁 1: 一般速度闪烁 (2 Hz) 2: 慢闪 (0.5 Hz)

## 30.22 SSL - ID W#16#xy75 – H 系统中切换的 DP 从站

### 目的

当 H 系统中的 CPU 处于冗余 H 运行模式时，你读取局部列表 SSL-ID W#16#xy75，你便可获得 H 系统与切换的 DP 从站之间通讯的状态信息。

局部列表会告诉你当前正用于与 DP 从站通讯的 DP 主站系统接口模块插入了哪一个机架。

### 首部

局部列表 SSL-ID W#16#xy75 的首部结构如下：

内容	含义
SSLID	局部列表摘要的 SSLID W#16#0C75: H 系统与一个切换的 DP 从站之间通讯的状态。你必须通过参数 INDEX 选择 DP 从站。
INDEX	DP 从站接口模块的诊断地址
LENTHDR	W#16#0010: 8 字长的一个数据纪录
N_DR	W#16#0001: 数据纪录的个数

### 数据纪录

局部列表 SSL-ID W#16#xy75 的数据纪录结构如下：

名称	长度	含义
adr1_bgt0	1 个字	其 DP 主站接口模块插于机架 0 的 DP 从站接口模块的第一部分地址： DP 主站的系统 ID 和站号
adr2_bgt0	1 个字	其 DP 主站接口模块插于机架 0 的 DP 从站接口模块的第二部分地址： 槽位和子模块槽位
adr1_bgt1	1 个字	其 DP 主站接口模块插于机架 1 的 DP 从站接口模块的第一部分地址： DP 主站的系统 ID 和站号
adr2_bgt1	1 个字	其 DP 主站接口模块插于机架 1 的 DP 从站接口模块的第二部分地址： 槽位和子模块槽位
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—14: 逻辑基本地址</li> <li>• 位 15: I/O 识别符 (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: 无访问其 DP 主站接口模块插于机架 0 的 DP 从站接口模块</li> <li>• 位 1 = 1: 无访问其 DP 主站接口模块插于机架 1 的 DP 从站接口模块</li> <li>• 位 2–7: 保留 (都为 0)</li> <li>• 位 8 = 1: 两个通讯通道功能都正常; 通讯正在通过机架 0 上 DP 主站接口模块进行。</li> <li>• 位 9 = 1: 两个通讯通道功能都正常; 通讯正在通过机架 1 上 DP 主站接口模块进行。</li> <li>• 位 10–15: 保留 (都为 0)</li> </ul>

### 30.23 SSL - ID W#16#xy90 – DP 主站系统信息

#### 目的

如果你读取局部列表 SSL-ID W#16#xy90, 你便可获得对 CPU 的已知的全部 DP 主站系统状态信息。

#### 首部

局部列表 SSL-ID W#16#xy90 的首部结构如下:

内容	Meaning
SSL-ID	局部列表摘要的 SSL-ID
	W#16#0090: 对 CPU 的已知的全部 DP 主站系统的信息
	W#16#0190: 一个 DP 主站系统的信息
	W#16#0F90: 仅为 SSL 局部列表首部信息
INDEX	<ul style="list-style-type: none"> <li>• 对于 SSL-ID W#16#0190 的局部列表摘要:               <ul style="list-style-type: none"> <li>低字节: B#16#00</li> <li>高字节: DP 主站系统 ID</li> </ul> </li> <li>• 对于 SSL-IDs W#16#0090 和 W#16#0F90 的局部列表摘要: W#16#0000</li> </ul>
LENTHDR	W#16#000E: 一个数据纪录长度为 7 个字 (14 个字节)
N_DR	数据纪录的个数 <ul style="list-style-type: none"> <li>• 对于 SSL-ID W#16#0190 的局部列表摘要: 0-1</li> <li>• 对于 SSL-ID W#16#0090 的局部列表摘要:               <ul style="list-style-type: none"> <li>- 标准 CPU: 0-14</li> <li>- H 系统: 0-12 (在除冗余以外的所有系统状态)</li> <li>0-2 x 12 (在除冗余系统状态)</li> </ul> </li> </ul>

数据纪录

局部列表 SSL-ID W#16#xy90 的数据纪录结构如下：

名称	长度	含义	
dp_m_id	1 个字节	DP 主站系统 ID	
rack_dp_m	1 个字节	DP 主站的机架号 <ul style="list-style-type: none"> <li>对标准 CPU: 0</li> <li>对 H 系统: 0 或 1</li> </ul>	
Steckpl_dp_m	1 个字节	DP 主站的槽位或 CPU (带集成 DP 接口) 的槽位	
Subm_dp_m	1 个字节	<ul style="list-style-type: none"> <li>带集成 DP 接口: DP 主站接口号:                             <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>带外部 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 个字节	DP 主站的进一步属性	
		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: 不相等</li> <li>1: 相等</li> </ul>
		位 2-6:	保留
		位 7:	DP 主站类型 <ul style="list-style-type: none"> <li>0: 集成 DP 主站</li> <li>1: 外部 DP 主站</li> </ul>
Reserve	3 个字节	保留	

关于多 CPU 系统的说明 (仅对 S7-400)

所有的局部列表都只提供已分配给一个 CPU 的模块的信息。在多 CPU 系统模式下, 你必须对所有 CPU 采样以便获得全部相关模块的数据。

## 30.24 SSL - ID W#16#xy91 – 模块状态信息

### 目的

如果你读取局部列表 SSL-ID W#16#xy91, 你便可获得....的信息。

### 首部

局部列表 SSL-ID W#16#xy91 的首部结构如下:

内容	含义
SSLID	<p>局部列表摘要的 SSL-ID</p> <p>W#16#0091 所有插入的模块和子模块的状态信息 (仅对 S7-400)</p> <p>W#16#0191 所有模块的状态信息/有错模块的识别符 (仅对 S7-400)</p> <p>W#16#0291 所有故障模块的状态信息 (仅对 S7-400)</p> <p>W#16#0391 所有不可用模块的状态信息 (仅对 S7-400)</p> <p>W#16#0591 所有主模块的子模块的状态信息</p> <p>W#16#0991 一个 DP 主系统的状态信息</p> <p>W#16#0A91 所有 DP 子模块和 DP 主系统的状态信息 (仅对 S7-300, 除 CPU 318-2 DP)</p> <p>W#16#0C91 一个位于中央机架的模块或通过逻辑基本地址的集成 DP 接口的状态信息</p> <p>W#16#4C91 一个通过逻辑基本地址连接到外部 DP 接口的模块的状态信息 若使用多于 4 个外部 DP 接口由于错误可能使 RET_VAL 为 W#16#80A4</p> <p>W#16#0D91 所有指定机架/指定站 (DP) 中模块的状态信息</p> <p>W#16#0E91 所有组态的模块的状态信息</p> <p>W#16#0E91 所有组态的模块的状态信息</p> <p>W#16#0F91 所有组态的模块的状态信息</p> <p>W#16#0F91 带有所有插入的模块/子模块的状态信息的数据纪录数量</p>
INDEX	<ul style="list-style-type: none"> <li>• 对于 SSLID W#16#0C91 的局部列表摘要: <ul style="list-style-type: none"> <li>- S7-400: 位 0—14: 模块的逻辑基本地址 位 15: 0 =输入, 1=输出</li> <li>- - S7-300: 模板起始地址</li> </ul> </li> <li>• 对于 SSLID W#16#4C91 的局部列表摘要 (仅对 S7-400): <ul style="list-style-type: none"> <li>位 0-14: 模块的逻辑基本地址</li> <li>位 15: 0 =输入, 1=输出</li> </ul> </li> <li>• 对于 SSLID 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 的局部列表摘要: INDEX 是无关的, 所有模块 (在机架中或在分布的 I/O 中)</li> </ul> <p>对于 SSLIDs W#16#0991 和 W#16#0D91 的局部列表摘要:</p> <p>W#16#00xx 一个机架的所有模块和子模块 (xx 包含机架的号码)</p> <p>W#16#xx00 一个 DP 主系统的所有模块 (xx 包含 DP 主系统的 ID)</p>

系统状态列表 (SSL)

内容	含义
	W#16#xyyy 一个 DP 站的所有模块 (xx 包含 DP 主系统的 ID, yy 为站号)
LENTHDR	W#16#0010:8 字长的一个数据纪录 (16 字节)
N_DR	数据记录的个数。与产品有关, SFC51 传送的纪录数可能减少

在 W#16#0091, W#16#0191 和 W#16#0F91 情况下, 为每个机架提供 2 个附加数据纪录:

- 一个纪录只对电源, 只要它存在并规划过, 而
- 一个纪录只对机架
- 在中央结构情况下的纪录顺序是: PS, 槽位 1, 槽位 2, ..., 槽位 18, 机架。.

局部列表 SSL\_ID W#16#xy91 的数据纪录结构如下:

名称	长度	含义
Adr1	1 个字	物理地址的机架号 (DP 主系统 ID 和带 DP 的站号)
Adr2	1 个字	槽位和子槽位
Logadr	1 个字	初始分配逻辑 I/O 的地质
Solltyp	1 个字	保留
Isttyp	1 个字	保留
Alarm	1 个字	保留 (00xx=CPU No. 1-4)
Eastat	1 个字	I/O 状态 位 0 = 1: 模块错误 (被诊断中断检测到) 位 1 = 1: 模块存在 位 2 = 1: 模块不存在 位 3 = 1: 模块被封锁 位 4 = 1: (仅对典型槽位) 位 5 = 1: M7: 模块可以作子模块的主 S7: 该模块/站上的一个 CiR 事件忙或尚未完成 为 S7-400 保留 位 6 = 1: 模块在局部总线段中 位 7 = 1: 用于逻辑地址的 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 0-2: 模块宽度 位 3: 保留 位 4-6: 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

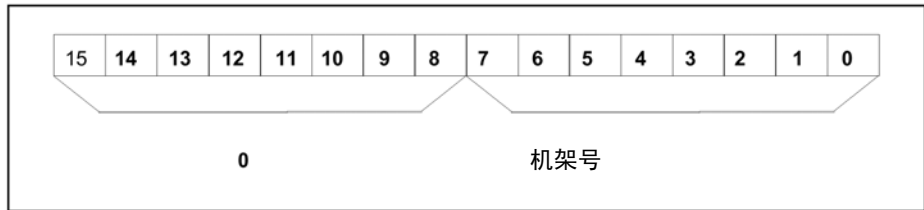
某些模块的记录值在下表中指明:

名称	PS (仅 S7-400)	CPU	IFM-CPU (仅 S7-300)	机架 (仅 S7-400)
Adr1	机架号	上文描述的标准信息	上文描述的标准信息	机架号
Adr2	W#16#01FF	W#16#0200 或 W#16#0200 至 W#16#1800	W#16#0200	W#16#00FF
Logadr	W#16#0000	W#16#7FFF	W#16#007C	W#16#0000
Solltyp	上文描述的标准信息	W#16#00C0 或 W#16#0081 或 W#16#0082	W#16#00C0	上文描述的标准信息
Eastat	W#16#0000	上文描述的标准信息	上文描述的标准信息	W#16#0000
Ber_b gbr	W#16#0000	W#16#0011 或 W#16#0001 或 W#16#0002	W#16#0011	W#16#0000

adr1

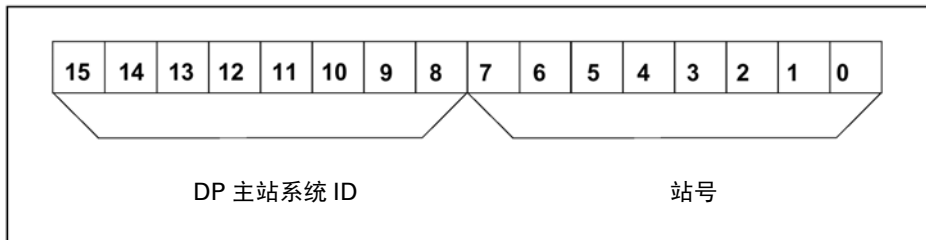
参数 adr1 的内容如下:

- 机架号, 当集中配置时



集中配置时参数 adr1 的各位

- 当分布配置时, DP 主系统的 ID 及站号。

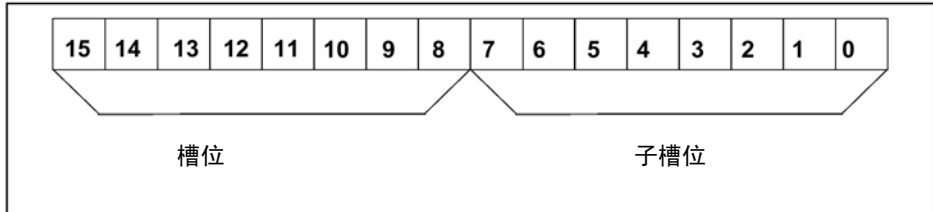


分布配置时参数 adr1 的各位



adr2

参数 adr2 包含槽位和子槽位。



参数 adr2 的各位。

---

关于多 CPU 系统的说明 (仅对 S7-400)

所有的局部列表都只提供已分配给一个 CPU 的模块的信息。在多 CPU 系统模式下, 你必须对所有 CPU 采样以便获得全部相关模块的数据。

---

## 30.25 SSL - ID W#16#xy92 – 机架/站状态信息

目的

如果你读取局部列表 SSL-ID W#16#xy92, 你便可获得关于集中配置的机架及 DP 主系统的预期和当前实际硬件组态的信息。

使用 S7-400 利用 SFC51 „RDSYSST“ 读出 SSL

如果你利用 SFC51 来读出局部列表, 你必须注意 SFC51 的参数 SSL\_ID 和 INDEX 要相互匹配。

SSL_ID	INDEX
W#16#0092 或 W#16#0292 或 W#16#0692 或	通过集成 DP 开关连接的 DP 主系统的 ID。
W#16#4092 或 W#16#4292 或 W#16#4692 或	通过外部 DP 开关连接的 DP 主系统的 ID。

## 首部

局部列表 SSL\_ID W#16#xy92 的首部结构如下：

内容	含义
SSLID	局部列表摘要的 SSL-ID: W#16#0092: 通过集成 DP 接口连接的一个 DP 主系统的中央机架/站的预期状态 W#16#4092: 通过外部 DP 接口连接的一个 DP 主系统站的预期状态 W#16#0292: 通过集成 DP 接口连接的一个 DP 主系统的中央机架/站的实际状态 W#16#0392: CPU 机架/站的电池供电状态, 若至少有一个电池失灵 W#16#0492: CPU 机架/站的整个电池供电状态 W#16#0592: CPU 机架/站的 24V 电源状态 W#16#4292: 通过外部 DP 接口连接的一个 DP 主系统站的实际状态 W#16#0692: 在一个通过集成 DP 接口连接的 DP 主系统站中央组态中的扩展机架的 OK 状态 W#16#4692: 一个通过外部 DP 接口连接的 DP 主系统站的 OK 状态
INDEX	0/ DP 主系统 ID
LENTHDR	W#16#0010: 一个 8 字长的数据记录 (16 字节)
N_DR	数据纪录的个数

## 数据纪录

局部列表 SSL\_ID W#16#xy92 的数据纪录结构如下：

名称	长度	含义
status_0 至 status_15	16 个字节	机架状态/站状态或后备状态。(后备状态只与 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: 站未组态或已组态但未激活 位=1: 站已组态并已激活 W#16#0292: 位=0: 机架/站失灵, 无效或未组态 位=1: 机架/站存在, 已激活并且没有失灵 W#16#4292: 位=0: 站失灵, 无效或未组态 位=1: 站存在, 已激活并且没有失灵 W#16#0692: 位=0: 一个站/扩展机架上所有模块都存在, 可用并无问题且站已激活 位=1: 一个站/扩展机架上至少有一个模块有问题或站无效 W#16#4692: 位=0: 一个站上所有模块都存在, 可用并无问题且站已激活 位=1: 一个站上至少有一个模块有问题或站无效
status_0	1 个字节	位 0: 中央机架 (INDEX = 0) 或站 1 (INDEX tu0)

系统状态列表 (SSL)

名称	长度	含义
		位 1: 1 扩展机架或站 2 : : 位 7: 7 扩展机架或站 8
status_1	1 个字节	位 0: 8 扩展机架或站 9 : : 位 7: 15 扩展机架或站 16
status_2	1 个字节	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 个字节	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
系统状态列表 (SSL)		
status_4	1 个字节	位 0: 0 或站 33 : : 位 7: 0 或站 40
status_15	1 个字节	位: 0 0 或站 121 : : 位 7: 0 或站 128

关于多 CPU 系统的说明 (仅对 S7-400)

所有的局部列表都只提供已分配给一个 CPU 的模块的信息。在多 CPU 系统模式下, 你必须对所有 CPU 采样以便获得全部相关模块的数据。

## 30.26 SSL - ID W#16#xy95 – 扩展的 DP 主系统信息

### 目的

局部列表 SSL ID W#16#xy95 为你提供对 CPU 的已知全部 DP 主站系统的扩展状态信息。与局部列表 SSL ID W#16#xy90 相比，这个列表包含有关一个 DP 主系统时钟同步的信息。

### 首部

局部列表 SSL ID W#16#xy95 的首部结构如下：

内容	含义	
SZLID	局部列表摘要的 SSL ID	
	W#16#0195:	一个 DP 主站系统的扩展信息
	W#16#0F95:	仅为 SSL 局部列表首部信息
INDEX	<ul style="list-style-type: none"> <li>对于 SSL ID W#16#0195 的局部列表摘要：               <ul style="list-style-type: none"> <li>低字节：B#16#00</li> <li>高字节：DP 主站系统 ID</li> </ul> </li> <li>对于 SSL ID W#16#0F95 的局部列表摘要：               <ul style="list-style-type: none"> <li>W#16#0000</li> </ul> </li> </ul>	
LENTHDR	W#16#0028:	一个数据纪录长度为 20 个字（40 个字节）
N_DR	数据纪录的个数： 对于 SSL ID W#16#0195 的局部列表摘要：0–1	

### 数据记录

局部列表 SSL\_ID W#16#xy95 的数据纪录结构如下：

名称	长度	含义
dp_m_id	1 个字节	DP 主站系统 ID
rack_dp_m	1 个字节	DP 主站的模块机架号 <ul style="list-style-type: none"> <li>对标准 CPU：0</li> <li>对 H 系统：0 或 1</li> </ul>
steckpl_dp_m	1 个字节	DP 主站的槽位或 CPU（带集成 DP 接口）的槽位
subm_dp_m	1 个字节	<ul style="list-style-type: none"> <li>带集成 DP 接口：DP 主站接口号：               <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>带外部 DP 接口：0</li> </ul>
logadr	2 个字节	DP 主站的逻辑起始地址
dp_m_sys_cpu	2 个字节	保留

系统状态列表 (SSL)

名称	长度	含义	
dp_m_sys_dpm	2 个字节	保留	
dp_m_state	1 个字节	DP 主站的进一步属性	
		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: 不相等</li> <li>• 1: 相等</li> </ul>
		位 2—6:	保留
		位 7:	DP 主站类型 <ul style="list-style-type: none"> <li>• 0: 集成 DP 主站</li> <li>• 1: 外部 DP 主站</li> </ul>
reserve	3 个字节	保留	
tsal_ob	1 个字节	已赋值的时钟同步中断 OB (仅在 DP 循环相等时有关)	
reserve	1 个字节	保留	
baudrate	4 个字节	DP 主站系统的传输速率 (16 进值)	
dp_iso_takt	4 个字节	相等的 DP 循环周期, 单位: $\mu\text{s}$	
reserve	16 个字节	保留	

关于多 CPU 系统的说明 (仅对 S7-400)

所有的局部列表都只提供已分配给一个 CPU 的模块的信息。在多 CPU 系统模式下, 你必须对所有 CPU 采样以便获得全部相关模块的数据。

## 30.27 SSL - ID W#16#xyA0 – 诊断缓冲区

### 目的

如果你读取局部列表 SSL-ID W#16#xyA0，你便可获得模块诊断缓冲区的条目。

### 首部

局部列表 SSL-ID W#16#xyA0 的首部结构如下：

内容	含义
SSLID	局部列表摘要的 SSL-ID: W#16#00A0: 当前模式下所有可能的条目 W#16#01A0: 最近的条目；你用参数 INDEX 指定最近的条目数。若诊断缓冲区中的消息数少于组态的最大的消息数，SFC51 可能会利用这个局部列表摘要提供无效的数值。因此要避免在无备份情况下断电。 W#16#0FA0: 仅为局部列表的首部信息
INDEX	仅对 SSL-ID W#16#01A0: 最近的条目数
LENTHDR	W#16#0014: 一个 10 字长的数据记录 (20 字节)
N_DR	数据纪录的个数

### 数据纪录

局部列表 SSL-ID W#16#xyA0 的数据纪录结构如下：

名称	长度	含义
ID	1 个字	事件 ID
Info	5 个字	关于事件及其结果的信息
Time	4 个字	事件的时间标记

### 诊断缓冲区

你可以用 STEP 7 来获得更多有关诊断缓冲区中事件的信息。

## 30.28 SSL - ID W#16#00B1 – 模块诊断信息

### 目的

如果你读取局部列表 SSL-ID W#16#00B1，你便可获得带有诊断能力的模块的前 4 个诊断字节。

### 首部

局部列表 SSL-ID W#16#00B1 的首部结构如下：

	含义
SSLID	W#16#00B1
INDEX	位 0—14：逻辑基本地址 位 15：0 = 输入，1 = 输出
LENTHDR	W#16#0004：一个 2 字长的数据记录（4 字节）
N_DR	1

### 数据纪录

局部列表 SSL-ID W#16#00B1 的数据纪录结构如下：

名称	长度	含义
byte1	1 个字节	位 0： 模块故障/OK（组故障 ID） 位 1： 内部故障 位 2： 外部故障 位 3： 通道错误存在 位 4： 无外部辅助电压 位 5： 无前连接器 位 6： 模块无参数分配 位 7： 模块错误参数
byte2	1 个字节	位 0—3： 模块类型（CPU，FM，CP，IM，SM，...） 位 4： 存在通道信息 位 5： 存在用户信息 位 6： 来自替代值的诊断中断 位 7： 保留（初始为 0）
byte3	1 个字节	位 0： 用户模块不正确/不存在 位 1： 通讯故障 位 2： 模式 RUN/STOP (0 = RUN, 1 = STOP) 位 3： 看门狗监控已响应 位 4： 内部模块电源故障 位 5： 电池耗尽（BFS） 位 6： 整个电池后备故障 位 7： 保留（初始为 0）

名称	长度	含义
byte4	1 个字节	位 0: 扩展机架失灵 (被 IM 检测到) 位 1: 处理器故障 位 2: EPROM 故障 位 3: RAM 故障 位 4: ADC/DAC 故障 位 5: 保险断 位 6: 硬件错误丢失 位 7: 保留 (初始为 0)

---

#### 关于多 CPU 系统的说明 (仅对 S7-400)

所有的局部列表都只提供已分配给一个 CPU 的模块的信息。在多 CPU 系统模式下, 你必须对所有 CPU 采样以便获得全部相关模块的数据。

---

## 30.29 SSL - ID W#16#00B2 – 带物理地址的诊断数据纪录 1

### 目的

如果你读取局部列表 SSL-ID W#16#00B2, 你便可获得中央机架中一个模块 (不适用 DP 或子模块) 的诊断数据纪录 1。你可以用机架和槽位号来指定编号。

### 首部

局部列表 SSL-ID W#16#00B2 的首部结构如下:

内容	含义
SSL-ID	W#16#00B2
INDEX	W#16#xyyy:   xx 包含机架号 yy 包含槽位号
LENTHDR	数据纪录的长度取决于模块
N_DR	1

### 数据纪录

局部列表 SSL-ID W#16#00B2 的数据纪录的大小和内容与模块有关。进一步的信息可参考 /70/, /101/ 和有关模块手册的说明。

---

#### 关于多 CPU 系统的说明 (仅对 S7-400)

所有的局部列表都只提供已分配给一个 CPU 的模块的信息。在多 CPU 系统模式下, 你必须对所有 CPU 采样以便获得全部相关模块的数据。

---



## 30.30 SSL - ID W#16#00B3 – 带逻辑基本地址的模块诊断数据

### 目的

如果你读取局部列表 SSL-ID W#16#xyB3，你便可获得一个模块的全部诊断数据。你还可获得用于 DP 和子模块的信息。你可以通过逻辑基本地址选择模块。

### 首部

局部列表 SSL-ID W#16#00B3 的首部结构如下：

内容	含义
SSLID	W#16#00B3
INDEX	位 0—14：逻辑基本地址 位 15：0 =输入，1=输出
LENTHDR	数据纪录的长度取决于模块
N_DR	1

### 数据纪录

局部列表 SSL-ID W#16#00B3 的数据纪录的大小和内容与特定的模块有关。进一步的信息可参考/70/，/101/和有关模块手册的说明。

---

#### 关于多 CPU 系统的说明（仅对 S7-400）

所有的局部列表都只提供已分配给一个 CPU 的模块的信息。在多 CPU 系统模式下，你必须对所有 CPU 采样以便获得全部相关模块的数据。

---

---

#### 注意

你只能在 OB82 之外用 SFC51 读取局部列表 SSL-ID W#16#00B3。

---

### 30.31 SSL - ID W#16#00B4 – 一个 DP 从站的诊断数据

#### 目的

如果你读取局部列表 SSL-ID W#16#xyB4, 你便可获得一个 DP 从站的诊断数据。这个诊断数据的结构遵从 EN 50 170 第 2 卷, PROFIBUS。你利用你组态的诊断地址选择模块。

#### 首部

局部列表 SSL-ID W#16#00B4 的首部结构如下:

内容	含义
SSLID	W#16#00B4
INDEX	DP 从站组态的诊断地址
LENTHDR	一个数据纪录的长度。最大长度为 240 个字节。对于其诊断数据长度大于 240 字节上至最多 244 字节的标准从站, 则读取前 240 个字节并在数据中将溢出位置位。
N_DR	1

#### 数据纪录

局部列表 SSL-ID W#16#00B4 的数据纪录结构如下:

名称	长度	含义
status1	1 个字节	站状态 1
status2	1 个字节	站状态 2
status3	1 个字节	站状态 3
stat_nr	1 个字节	主站号
ken_hi	1 个字节	供货商 ID (高字节)
ken_lo	1 个字节	供货商 ID (低字节)
....	....	对于特殊从站的专用的进一步诊断数据

#### 关于多 CPU 系统的说明 (仅对 S7-400)

所有的局部列表都只提供已分配给一个 CPU 的模块的信息。在多 CPU 系统模式下, 你必须对所有 CPU 采样以便获得全部相关模块的数据。



# 31 事件

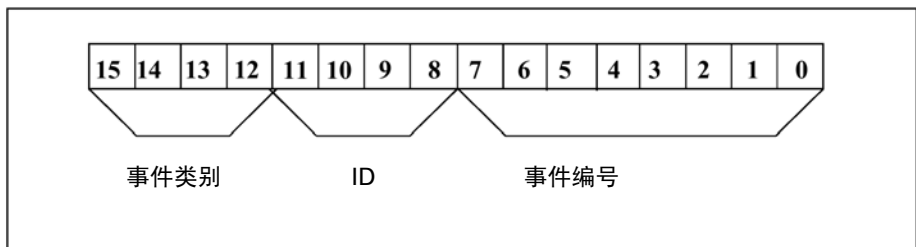
## 31.1 事件和事件 ID

### 事件

所有 SIMATIC S7 可编程控制器中的事件都进行了编号。  
这使得可以将信息文本与事件联系起来。

### 事件 ID

每个事件都有指定的事件 ID。事件 ID 的结构如下：



事件 ID 的结构

### 事件类别

事件类别如下：

编号	事件类别
1	标准 OB 事件
2	同步错误
3	非同步错误
4	模式转换
5	运行时间事件
6	通讯事件
7	针对自动防故障冗余系统的事件
8	模块的诊断事件
9	标准用户事件
A, B	自由用户事件
C, D, E	保留
F	针对除 CPU 之外模块（例如，CP 和 FM）的事件

## 标识

标识被用来区分事件的类型。其四个位具有如下含义：

事件 ID	含义
8	= 0 离去的事件 = 1 到来的事件
9	= 1 条目在诊断缓冲区内
10	= 1 内部错误
11	= 1 外部错误

## 31.2 事件类别 1 – 标准 OB 事件

事件 ID	事件
W#16#1381	要求手动暖起动
W#16#1382	要求自动暖起动
W#16#1383	要求手动热起动
W#16#1384	要求自动热起动
W#16#1385	要求手动冷起动
W#16#1386	要求自动冷起动
W#16#1387	主 CPU：要求手动冷起动
W#16#1388	主 CPU：要求自动冷起动
W#16#138A	主 CPU：要求手动暖起动
W#16#138B	主 CPU：要求自动暖起动
W#16#138C	备用 CPU：要求手动热起动
W#16#138D	备用 CPU：要求自动热起动

### 31.3 事件类别 2 – 同步错误

事件 ID	事件	OB
W#16#2521	BCD 转换错误	OB 121
W#16#2522	读操作时区域长度错误	OB 121
W#16#2523	当写操作时区域长度错误	OB 121
W#16#2524	读操作时区域错误	OB 121
W#16#2525	写操作时区域错误	OB 121
W#16#2526	计时器错误	OB 121
W#16#2527	计数器错误	OB 121
W#16#2528	读操作时校正错误	OB 121
W#16#2529	写操作时校正错误	OB 121
W#16#2530	访问 DB 时写错误	OB 121
W#16#2531	访问 DI 时写错误	OB 121
W#16#2532	打开 DB 时块号错误	OB 121
W#16#2533	打开 DI 时块号错误	OB 121
W#16#2534	调用 FC 时块号错误	OB 121
W#16#2535	调用 FB 时块号错误	OB 121
W#16#253A	DB 未装载	OB 121
W#16#253C	FC 未装载	OB 121
W#16#253D	SFC 未装载	OB 121
W#16#253E	FB 未装载	OB 121
W#16#253F	SFB 未装载	OB 121
W#16#2942	I/O 访问错误, 读	OB 122
W#16#2943	I/O 访问错误, 写	OB 122

## 31.4 事件类别 3 – 非同步错误

事件 ID	事件	OB
W#16#3501	循环时间超出。	OB 80
W#16#3502	用户接口 (OB 或 FRB) 请求错误	OB 80
W#16#3503	处理一个优先级延时太长	-
W#16#3505	日期时间中断被跳过, 由于设定了新时钟	OB 80
W#16#3506	日期时间中断被跳过, 当 HOLD 之后变为 RUN	OB 80
W#16#3507	内部缓冲区溢出引起的多 OB 错误	OB 80
W#16#3508	同步循环中断计时错误	OB 80
W#16#3509	由于中断负载超额引起中断丢失	OB 80
W#16#350A	在 CiR 之后恢复 RUN	OB 80
W#16#3921/3821	BATTF: 至少一个中央机架的后备电池失灵/问题被排除。注: 事件的到来仅发生在后备电池这一失灵时 (若有冗余的后备电池)。如果另一个后备电池碰巧也失灵事件回再次发生。	OB 81
W#16#3922/3822	BAF: 中央机架的后备电压失灵/问题被排除。	OB 81
W#16#3923/3823	中央机架的 24V 电压失灵/问题被排除	OB 81
W#16#3925/3825	BATTF: 至少一个冗余中央机架的后备电池失灵/问题被排除	OB 81
W#16#3926/3826	BAF: 冗余中央机架的后备电压失灵/问题被排除。	OB 81
W#16#3917/3827	冗余中央机架的 24V 电压失灵/问题被排除	OB 81
W#16#3931/3831	BATTF: 至少一个扩展机架的后备电池失灵/问题被排除	OB 81
W#16#3932/3832	BAF: 扩展机架的后备电压失灵/问题被排除	OB 81
W#16#3933/3833	至少一个扩展机架的 24V 电压失灵/问题被排除	OB 81
W#16#3942	模块错误	OB 82
W#16#3842	模块 OK	OB 82
W#16#3861	模块/接口模块已插入, 模块类型 OK	OB 83
W#16#3961	模块/接口模块被摘除, 无法编址	OB 83
W#16#3863	模块/接口模块插入, 但模块类型错	OB 83
W#16#3864	模块/接口模块插入, 但引起问题 (类型 ID 不可读)	OB 83
W#16#3865	模块插入, 但在模块参数分配中出错	OB 83
W#16#3866	模块可重新被编址, 负载电压错误排除	OB 83
W#16#3966	模块不能被编址, 负载电压错误	OB 83
W#16#3367	开始模块重新组态	OB 83
W#16#3267	结束模块重新组态	OB 83
W#16#3968	模板重新组态出错结束	OB 83
W#16#3571	嵌套深度太大	OB 88
W#16#3572	嵌套深度太大, 对主控继电器	OB 88
W#16#3573	嵌套深度太大, 同步错误之后	OB 88

事件 ID	事件	OB
W#16#3574	对于块调用（U 堆栈）嵌套深度太大	OB 88
W#16#3575	对于块调用（B 堆栈）嵌套深度太大	OB 88
W#16#3576	局部数据分配错误	OB 88
W#16#3578	未知指令	OB 88
W#16#357A	跳转指令的目标在块之外	OB 88
W#16#3884	接口模块插入	OB 83
W#16#3984	接口模块被摘除	OB 83
W#16#3981	接口错误到来状态	OB 84
W#16#3881	接口错误离去状态	OB 84
W#16#35A1	用户接口（OB 或 FRB）未找到	OB 85
W#16#35A2	OB 未装载（由 SFC 启动或操作系统组态）	OB 85
W#16#35A3	操作系统访问一个块时出错	OB 85
W#16#39B1	I/O 访问错误，刷新 PII 表时	OB 85
W#16#39B2	I/O 访问错误，传送 PIQ 到输出模块时	OB 85
W#16#39B3/38B3	I/O 访问错误，刷新 PII 表时	OB 85
W#16#39B4/38B4	I/O 访问错误，传送 PIQ 到输出模块时	OB 85
W#16#38C1	（1 至 21），离去状态	OB 86
W#16#39C1	扩展机架失灵（1 至 21），到来状态	OB 86
W#16#38C2	扩展机架重新运行但给定值与实际值组态错误匹配	OB 86
W#16#39C3	分布的 I/O：主系统失灵，到来状态	OB 86
W#16#39C4	分布的 I/O：站失灵，到来状态	OB 86
W#16#38C4	分布的 I/O：站失灵，离去状态	OB 86
W#16#39C5	分布的 I/O：站失灵，到来状态	OB 86
W#16#38C5	分布的 I/O：站失灵，离去状态	OB 86
W#16#38C6	扩展机架重新运行，但在模块参数分配中出错	OB 86
W#16#38C7	DP：站重新运行，但在模块参数分配中出错	OB 86
W#16#38C8	DP：站重新运行，但给定值与实际值组态错误匹配	OB 86
W#16#35D2	诊断条目当前不能发送	OB 87
W#16#35D3	同步构架不能发送	OB 87
W#16#35D4	由同步引起的非法时间跳转	OB 87
W#16#35D5	错误采用同步时间	OB 87
W#16#35E1	GD 中不正确的构架 ID	OB 87
W#16#35E2	GD 包的状态不能进入 DB	OB 87
W#16#35E3	GD 中构架长度错误	OB 87
W#16#35E4	接收了非法的 GD 包号	OB 87
W#16#35E5	访问用于“组态的 S7 连接”的 SFB 通讯中的 DB 时出错	OB 87
W#16#35E6	GD 的整体状态不能进入 DB	OB 87



## 31.5 事件类别 4 – 停机事件和其他模式转换

事件 ID	事件
W#16#4300	后备电源接通
W#16#4301	模式由转 STOP 换至 STARTUP
W#16#4302	模式由 STARTUP 转换至 RUN
W#16#4303	停机，由于 STOP 开关作用
W#16#4304	停机，由编程器设 STOP 或由 SFB20 引起
W#16#4305	HOLD：断点到达
W#16#4306	HOLD：断点退出
W#16#4307	存储器清除，由 PG 实行
W#16#4308	存储器清除，由开关进行
W#16#4309	存储器清除，自动进行（电源接通未备份）
W#16#430A	HOLD 退出，转换至 STOP
W#16#430D	停机，由多 CPU 系统中其它 CPU 引起
W#16#430E	存储器清除已执行
W#16#430F	模块停止，由于 CPU 停止造成
W#16#4510	停机，由于侵害 CPU 的数据范围
W#16#4318	CiR 开始
W#16#4319	CiR 完成
W#16#4520	故障：停机不可能
W#16#4521	故障：指令处理器故障
W#16#4522	故障：时钟芯片故障
W#16#4523	故障：时钟脉冲发生器故障
W#16#4524	故障：计时器刷新功能故障
W#16#4525	故障：多 CPU 系统同步故障
W#16#4926	故障：用于 I/O 访问的看门狗故障
W#16#4527	故障：I/O 访问监控故障
W#16#4528	故障：扫描时间监控故障
W#16#4530	故障：内部存储器检测错误
W#16#4931	STOP 或故障：存储器子模块检测错误
W#16#4532	故障：内核资源故障
W#16#4933	故障：总和检查错误
W#16#4934	故障：存储器不存在
W#16#4935	故障：由看门狗取消/处理器除外

事件 ID	事件
W#16#4536	故障：开关损坏
W#16#4540	停机：内部工作内存的存储器扩展有间隙。第一存储器扩展太小或丢失。
W#16#4541	停机，优先级系统引起
W#16#4542	停机，目标管理系统引起
W#16#4543	停机，检测功能引起
W#16#4544	停机，诊断系统引起
W#16#4545	停机，通讯系统引起
W#16#4546	停机，CPU 内存管理引起
W#16#4547	停机，处理缓冲管理引起
W#16#4548	停机，I/O 管理引起
W#16#4949	停机，连续硬件中断引起
W#16#454A	停机，由组态引起：通过 STEP7 取消的 OB 在 STARTUP 时被装入 CPU
W#16#494D	停机，由 I/O 错误引起
W#16#494E	停机，由电源故障引起
W#16#494F	停机，由组态错误引起
W#16#4550	故障：内部系统错误
W#16#4555	再起动不可能，监控时间已过
W#16#4556	停机：需要从通讯系统实施内存清除
W#16#4357	模块看门狗启动
W#16#4358	所有模块做好运行准备
W#16#4959	一个或几个模块未做好运行准备
W#16#4562	停机，由编程错误引起（OB 未装载或不可能）
W#16#4563	停机，由 I/O 访问错误引起（OB 未装载或不可能）
W#16#4567	停机，由 H 事件引起
W#16#4568	停机，由时间错误引起（OB 未装载或不可能）
W#16#456A	停机，由诊断中断引起（OB 未装载或不可能）
W#16#456B	停机，由模块摘除/插入引起（OB 未装载或不可能）
W#16#456C	停机，由 CPU 硬件错误引起（OB 未装载或不可能）
W#16#456D	停机，由程序顺序错误引起（OB 未装载或不可能）
W#16#456E	停机，由通讯错误引起（OB 未装载或不可能）
W#16#456F	停机，由机架故障引起（OB 未装载或不可能）
W#16#4570	停机，由处理中断引起（OB 未装载或不可能）
W#16#4571	停机，由嵌套堆栈错误引起
W#16#4572	停机，由主控继电器错误引起
W#16#4573	停机，同步错误嵌套层深超出引起
W#16#4574	停机，由优先级堆栈内中断堆栈层深超出引起

事件

事件 ID	事件
W#16#4575	停机, 由优先级堆栈内块堆栈层深超出引起
W#16#4576	停机, 由分配局部数据时出错引起
W#16#4578	停机, 由未知代码引起
W#16#457A	停机, 由代码长度错误引起
W#16#457B	停机, 由 on-board I/O 的 DB 未装载引起
W#16#457F	停机, 由停机指令引起
W#16#4580	停机: 备份缓冲内容矛盾 (不能转换到 RUN)
W#16#4590	停机, 由内部功能过载引起
W#16#49A0	停机, 由参数分配错误或不允许的给定值和实际值变化引起: STARTUP 被封锁
W#16#49A1	停机, 由参数分配错误引起: 需要清楚内存
W#16#49A2	停机, 由参数修改错误引起: STARTUP 被封锁
W#16#49A3	停机, 由参数修改错误引起: 需要清楚内存
W#16#49A4	停机: 组态数据矛盾
W#16#49A5	停机: 分布 I/O: 装载的组态信息矛盾
W#16#49A6	停机: 分布 I/O: 组态信息无效
W#16#49A7	停机: 分布 I/O: 无组态信息
W#16#49A8	停机: 由用于分布 I/O 的接口模块指明的错误
W#16#43B0	固件升级成功
W#16#49B1	固件升级数据不正确
W#16#49B2	固件升级: 硬件版本与固件不匹配
W#16#49B3	固件升级: 模块类型与固件不匹配
W#16#49D0	LINK-UP 失败, 由于违反协调规则造成
W#16#49D1	LINK-UP/UPDATE 过程失败
W#16#49D2	由于主 CPU 在连通过程中处于 STOP 造成备用 CPU 转为 STOP
W#16#43D3	备用 CPU 处于 STOP
W#16#49D4	主 CPU 处于 STOP, 由于对组 CPU 也是主 (连通错误)
W#16#45D5	LINK-UP 被拒绝, 由于子 PLC 错误匹配 CPU 内存组态
W#16#45D6	LINK-UP 被拒绝, 由于子 PLC 系统程序错误匹配
W#16#49D7	LINK-UP 被拒绝, 由于用户程序或组态的改变
W#16#45D8	DEFECTIVE: 由于其它错误引起硬件故障
W#16#45D9	停机, 由 SYNC 模块错误引起
W#16#45DA	停机, 由 H CPU 之间的同步错误引起
W#16#43DC	切换连通失败
W#16#45DD	LINK-UP 被拒绝, 由运行试验或在线功能引起
W#16#43DE	刷新失败, 由于 n 次尝试中监控时间超出, 新的刷新已开始
W#16#43DF	最终刷新失败, 由于完成了最大尝试次数后监控时间超出。需要用户干涉。

事件 ID	事件
W#16#43E0	连通后转出单独模式
W#16#43E1	刷新后转出连通模式
W#16#43E2	从刷新转至冗余模式
W#16#43E3	主 CPU：从冗余模式转至单独模式
W#16#43E4	备用 CPU：故障诊断模式后转出冗余模式
W#16#43E5	备用 CPU：连通或停机后转出故障诊断模式
W#16#43E6	备用 CPU 上连通失败
W#16#43E7	备用 CPU 上刷新失败
W#16#43E8	备用 CPU：STARTUP 之后转出连通模式
W#16#43E9	备用 CPU：刷新后转出 STARTUP
W#16#43F1	保留主切换
W#16#43F2	不兼容的 H-CPU 耦合被系统程序封锁
W#16#42F3	发现总和检查错误且被系统程序纠正
W#16#42F4	备用 CPU：通过 SFC90 的连接和刷新在主 CPU 中被封锁

## 31.6 事件类别 5 – 模式运行时间事件

事件 ID	事件
W#16#530D	停机模式下新的启动信息
W#16#5311	启动而不顾来自模块的“未准备好”信息
W#16#5545	在 RUN 模式下开始系统重新组态
W#16#5445	在 RUN 模式下开始系统重新组态
W#16#5371	分布式 I/O：带一个 DP 的主站同步结束
W#16#5380	诊断缓冲区关于中断和非同步错误的条目被封锁
W#16#5395	分布式 I/O：DP 的主站复位
W#16#53FF	复位恢复工厂设定
W#16#5481	所有运行时间软件的许可再次完成
W#16#5498	由于 CiR，DP 的主系统不再有矛盾
W#16#5581	一个或几个运行时间软件的许可丢失
W#16#558A	MLFB 的组态与插入的 CPU 之间的不同
W#16#558B	固件版本与插入的 CPU 之间的不同
W#16#5598	由于 CiR，可能与带 DP 的主系统矛盾
W#16#5960	转换时参数分配错误

事件 ID	事件
W#16#5961	参数分配错误
W#16#5962	参数分配错误防止启动
W#16#5963	参数分配错误，需要内存清除
W#16#5966	切换时参数分配错误
W#16#5969	参数分配错误，启动被封锁
W#16#5979/5879	来自 DP 接口的诊断信息：EXTF 发光二极管亮/灭
W#16#597C	DP 全局控制指令失灵或被移动
W#16#597C	DP 全局控制指令失灵或被移动
W#16#59A0	中断在 CPU 中不能联系

### 31.7 事件类别 6 – 通讯事件

事件 ID	事件
W#16#6316	启动 PLC 时接口出错
W#16#6390	微存储卡格式化完毕
W#16#6500	连接的 ID 在模块中存在两次
W#16#6501	连接的资源不充分
W#16#6502	连接描述中出错
W#16#6510	连接描述中出错
W#16#6514	当评测 EPROM 时在背景 DB 中检测到 CFB 结构出错
W#16#6515	GD 包编号在模块中存在两次
W#16#6521	GD 组态信息中矛盾的长度规范
W#16#6522	无存储子模块且内部存储不存在
W#16#6523	非法存储子模块：替换子模块并清除存储器
W#16#6524	需要清除存储器，由于访问子模块时出错
W#16#6526	需要清除存储器，由于存储器的替换
W#16#6527	由于存储器被替换再起动不可能
W#16#6528	在 STOP/HOLD 模式中的目标处理功能，再起动不可能
W#16#6529	在“加载用户程序”功能时再启动不可能
W#16#652A	启动不可能，因为块在用户内存中存在两次
W#16#652B	启动不可能，因为块对于存储子模块太长-替换子模块
W#16#652C	启动不可能，由于在子模块中非法的 OB
W#16#6532	启动不可能，由于在子模块中非法的组态信息
W#16#6533	需要清除存储器，由于无效的子模块内容

事件 ID	事件
W#16#6534	启动不可能：因为块在存储器中存在一次以上
W#16#6535	启动不可能：存储容量不够接收从子模块传来的块
W#16#6536	启动不可能：子模块中含有非法块号
W#16#6537	启动不可能：子模块中含有非法长度的块
W#16#6538	一个块的局部数据或写保护 ID（对于 DB）对于 CPU 非法
W#16#6539	块内非法命令（编译器检测到）
W#16#653A	需要清除存储器，由于子模块中 OB 局部数据太短
W#16#6543	启动不可能：非法的块类型
W#16#6544	启动不可能：属性“相关处理”非法
W#16#6545	源语言非法
W#16#6546	达到组态信息的最大量
W#16#6547	给模块分配参数时参数分配出错（不适应 P 总线，取消下装）
W#16#6548	块检查时似是而非的错误
W#16#6549	块内结构出错
W#16#6550	块在循环冗余码校验中有一个错误
W#16#6551	块无循环冗余码校验
W#16#6560	扫描溢出
W#16#6881	接口出错，离去状态
W#16#6905/6805	组态的连接有资源问题/被排除
W#16#6981	接口出错，到来状态

## 31.8 事件类别 7 – H/F 事件

事件 ID	事件	OB
W#16#72A2	一个 DP 的主站或 DP 的主站系统故障	OB 70
W#16#72A3	DP 从站的冗余恢复	OB 70
W#16#7310/7210	冗余丢失 I/O	OB 70
W#16#7311/7211	部分冗余丢失 I/O	OB 70
W#16#73A3	冗余丢失, DP 从站	OB 70
W#16#7301	冗余丢失 (1 of 2), 由于用户触发使备份机处于 STOP	OB 72
W#16#7302	冗余丢失 (1 of 2) 转变为冗余模式	OB 72
W#16#7303	H 系统 (二中之) 变为冗余模式	OB 72
W#16#7320	RAM 比较故障	OB 72
W#16#7321	处理映象区输出值比较中出错	OB 72
W#16#7322	存储器位, 计时器或计数器比较中出错	OB 72
W#16#7323	操作系统数据中发现矛盾	OB 72
W#16#7331	备份主站切换, 由于主站失灵	OB 72
W#16#7333	备份主站切换, 由于在运行时系统修正	OB 72
W#16#7334	备份主站切换, 由于同步模块上的通讯错误	OB 72
W#16#7335	备份主站切换, 由于被 90 “H_CTRL” 触发	OB 72
W#16#7340	用户程序中同步错误, 由于超过等待时间	OB 72
W#16#7341	用户程序中同步错误, 由于在不同的同步点等待	OB 72
W#16#7342	操作系统中同步错误, 由于在不同的同步点等待	OB 72
W#16#7343	操作系统中同步错误, 由于超过等待时间	OB 72
W#16#7344	操作系统中同步错误, 由于不正确的数据	OB 72
W#16#73C1	刷新处理被取消	OB 72
W#16#73C2	刷新失败, 由于 n 次尝试过程中监控时间超出 (1 n 监控超时时最大可能的刷新尝试次数)	OB 72
W#16#7950	同步模块丢失	OB 72
W#16#7951	在同步模块上作改动不实行上电	OB 72
W#16#7952/7852	SYNC 模块移出/插入	OB 72
W#16#7953	在同步模块上作改动不实行复位	OB 72
W#16#7954	SYNC 模块: 机架号被指定两次	OB 72
W#16#7955/7855	SYNC 模块故障/排除	OB 72
W#16#7956	在同步模块上设了非法的机架号	OB 72
W#16#734A	由 SFC 90 “H_CTRL” 触发的“再使能”任务执行完	-
W#16#734B	由操作系统触发的“再使能”任务执行完	-

W#16#7960	冗余 I/O: 数字输入上时间失调超时, 错误未局部化	-
W#16#7961	冗余 I/O, 数字输入错误: 时间失调结束后信号变化	-
W#16#7962	冗余 I/O: 数字输入错误	-
W#16#796F	冗余 I/O: I/O 被实行全局封锁	-
W#16#7970	冗余 I/O: 数字输出错误	-
W#16#7980	冗余 I/O: 模拟输入上时间失调	-
W#16#7981	冗余 I/O: 模拟输入错误	-
W#16#7990	冗余 I/O: 模拟输出错误	-
W#16#75D1	安全程序: 内部 CPU 出错	-
W#16#75D2	安全程序错误: 循环时间超时	-
W#16#79D3/78D3	PROFIsafe 与 F I/O 之间通讯出错	-
W#16#79D4/78D4	F CPU 之间安全关联的通讯错误	-
W#16#79D5/78D5	F CPU 之间安全关联的通讯错误	-
W#16#75D6	安全程序中数据在输出到 F I/O 之前被毁坏	-
W#16#75D7	安全程序中数据在输出到组对 F CPU 之前被毁坏	-
W#16#73D8	安全模式被封锁	-
W#16#75D9	在一个 DB 中有无效的 REAL 数	-
W#16#75DA	安全程序: 安全数据格式出错	-
W#16#75DC	运行时间组, 内部协议出错	-
W#16#75DD/74DD	安全程序: 失败保存运行时间组关断使能/封锁	-
W#16#75DE/74DE	安全程序: F 程序关断使能/封锁	-
W#16#75DF/74DF	F 程序初始化开始/结束	-
W#16#75E1	安全程序: 在 FB “F_PLK” 或 “F_PLK_O” 或 “F_CYC_CO” 或 “F_TEST” 或 “F_TESTC”	-



## 31.9 事件类别 8 – 模块的诊断事件

事件 ID	事件	模块类型
W#16#8x00	模块故障/OK	任意
W#16#8x01	内部错误	
W#16#8x02	外部错误	
W#16#8x03	通道错误	
W#16#8x04	无外部辅助电压	
W#16#8x05	无前连接器	
W#16#8x06	无参数分配	
W#16#8x07	模块中参数不正确	
W#16#8x00	用户子模块不正确/未发现	
W#16#8x31	通讯问题	
W#16#8x32	运行模式：RUN/STOP（STOP：到来状态，RUN：离去状态）	
W#16#8x33	时间监控响应（看门狗）	
W#16#8x34	内部模块电源失灵	
W#16#8x35	BATTF：电池耗尽	
W#16#8x36	整体备份失败	
W#16#8x37	保留	
W#16#8x40	扩展机架失灵	
W#16#8x41	处理器故障	
W#16#8x42	EPROM 故障	
W#16#8x43	RAM 故障	
W#16#8x44	ADC/DAC 故障	
W#16#8x45	保险断	
W#16#8x46	硬件中断丢失	
W#16#8x47	保留	
W#16#8x50	组态/参数分配错误	模拟输入
W#16#8x51	共模错误	
W#16#8x52	相短路	
W#16#8x53	对地短路	
W#16#8x54	断线	
W#16#8x55	参考通道错误	
W#16#8x56	下测量范围	
W#16#8x57	上测量范围	

事件 ID	事件	模块类型
W#16#8x60	组态/参数分配错误	模拟输出
W#16#8x61	共模错误	
W#16#8x62	相短路	
W#16#8x63	对地短路	
W#16#8x64	断线	
W#16#8x65	保留	
W#16#8x66	无负载电压	
W#16#8x70	组态/参数分配错误	数字输入
W#16#8x71	底盘接地故障	
W#16#8x72	相短路（传感器）	
W#16#8x73	对地短路（传感器）	
W#16#8x74	断线	
W#16#8x75	无传感器电源	
W#16#8x80	组态/参数分配错误	数字输出
W#16#8x81	底盘接地故障	
W#16#8x82	相短路	
W#16#8x83	对地短路	
W#16#8x84	断线	
W#16#8x85	保险断	
W#16#8x86	无负载电压	
W#16#8x87	超温	
W#16#8xB0	计数模块，信号 A 故障	FM
立 W#16#8xB1	计数模块，信号 B 故障	
W#16#8xB2	计数模块，信号 N 故障	
W#16#8xB3	计数模块，通道间传递了不正确的值	
W#16#8xB4	计数模块，5.2 V 传感器电源故障	
W#16#8xB5	计数模块，24 V 传感器电源故障	

## 31.10 事件类别 9 – 标准用户事件

事件 ID	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	手动命令使能
W#16#9x04	单位保护命令 (OPEN/CLOSED)
W#16#9x05	处理使能
W#16#9x06	系统保护命令
W#16#9x07	处理值监控响应
W#16#9x08	操纵量监控响应
W#16#9x09	系统偏差大于允许值
W#16#9x0A	限位错误
W#16#9x0B	运行时间错误
W#16#9x0C	命令执行错误 (顺序器)
W#16#9x0D	操作状态运行 > OPEN
W#16#9x0E	操作状态运行 > CLOSED
W#16#9x0F	命令封锁
W#16#9x11	处理状态 OPEN/ON
W#16#9x12	处理状态 CLOSED/OFF
W#16#9x13	处理状态中间位置
W#16#9x14	处理状态通过 AUTO 到 ON
W#16#9x15	处理状态通过手动到 ON
W#16#9x16	处理状态通过保护命令到 ON
W#16#9x17	处理状态通过 AUTO 到 OFF
W#16#9x18	处理状态通过手动到 OFF
W#16#9x19	处理状态通过保护命令到 OFF
W#16#9x21	接近时功能出错
W#16#9x22	离去时功能出错
W#16#9x31	执行器 (DE/WE) 限位 OPEN
W#16#9x32	执行器 (DE/WE) 限位没有 OPEN
W#16#9x33	执行器 (DE/WE) 限位 CLOSED
W#16#9x34	执行器 (DE/WE) 限位没有 CLOSED
W#16#9x41	非法状态, 允差时间超出
W#16#9x42	非法状态, 允差时间未超出

事件 ID	事件
W#16#9x43	互锁错误, 允差时间=0
W#16#9x44	互锁错误, 允差时间> 0
W#16#9x45	无反应
W#16#9x46	最终状态非法退出, 允差时间=0
W#16#9x47	最终状态非法退出, 允差时间> 0
W#16#9x50	信号范围上限 USR
W#16#9x51	测量范围上限 UMR
W#16#9x52	信号范围下限 LSR
W#16#9x53	测量范围上限 LMR
W#16#9x54	报警上限 UAL
W#16#9x55	警告上限 UWL
W#16#9x56	允差上限 UTL
W#16#9x57	允差下限 LTL
W#16#9x58	报警下限 LWL
W#16#9x59	报警下限 LAL
W#16#9x60	GRAPH7 步到来/离去
W#16#9x61	GRAPH7 互锁错误
W#16#9x62	GRAPH7 执行错误
W#16#9x63	GRAPH7 错误注释
W#16#9x64	GRAPH7 错误应答
W#16#9x70	趋势超出正方向
W#16#9x71	趋势超出负方向
W#16#9x72	无反应
W#16#9x73	最终状态非法退出
W#16#9x80	极限值超出, 允差时间=0
W#16#9x81	极限值超出, 允差时间> 0
W#16#9x82	低限值, 允差时间=0
W#16#9x83	低限值, 允差时间> 0
W#16#9x84	梯度超出, 允差时间=0
W#16#9x85	梯度超出, 允差时间> 0
W#16#9x86	低梯度, 允差时间=0
W#16#9x87	低梯度, 允差时间> 0
W#16#9190/9090	用户参数分配错误到来/离去
W#16#91F0	溢出
W#16#91F1	下溢出
W#16#91F2	除 0 除
W#16#91F3	非法运算操作

### 31.11 事件类别 A 和 B – 自由用户事件

事件ID	事件
W#16#Axyz	用户可利用的事件
W#16#Bxyz	

### 31.12 保留的事件类别

#### 保留

下列事件类别保留做为以后的扩展使用：

- C
- D
- E
- F 保留用于不在中央机架上的模块（例如，CP 和 FM）

## 32 SFC 和 SFB 表

### 32.1 SFC 表，按编号排序

编号	短名	功能
SFC 0	SET_CLK	设系统时钟
SFC 1	READ_CLK	读系统时钟
SFC 2	SET_RTM	运行时间计时器设定
SFC 3	CTRL_RTM	运行时间计时器启/停
SFC 4	READ_RTM	运行时间计时器读取
SFC 5	GADR_LGC	查询模板的逻辑起始地址
SFC 6	RD_SINFO	读 OB 起动信息
SFC 7	DP_PRAL	在 DP 主站上触发硬件中断
SFC 9	EN_MSG	使能块相关，符号相关和组状态的信息
SFC 10	DIS_MSG	封锁块相关，符号相关和组状态的信息
SFC 11	DPSYC_FR	同步 DP 从站组
SFC 12	D_ACT_DP	取消和激活 DP 从站
SFC 13	DPNRM_DG	读 DP 从站的诊断数据（从站诊断）
SFC 14	DPRD_DAT	读标准 DP 从站的连续数据
SFC 15	DPWR_DAT	写标准 DP 从站的连续数据
SFC 17	ALARM_SQ	生成可应答的块相关信息
SFC 18	ALARM_S	生成恒定可应答的块相关信息
SFC 19	ALARM_SC	查询最后的 ALARM_SQ 到来状态信息的应答状态
SFC 20	BLKMOV	拷贝变量
SFC 21	FILL	初始化存储区
SFC 22	CREAT_DB	生成 DB
SFC 23	DEL_DB	删除 DB
SFC 24	TEST_DB	测试 DB
SFC 25	COMPRESS	压缩用户内存
SFC 26	UPDAT_PI	刷新过程映像更新表
SFC 27	UPDAT_PO	刷新过程映像输出表
SFC 28	SET_TINT	设置日时钟中断
SFC 29	CAN_TINT	取消日时钟中断
SFC 30	ACT_TINT	激活日时钟中断
SFC 31	QRY_TINT	查询日时钟中断

编号	短名	功能
SFC 32	SRT_DINT	启动延时中断
SFC 33	CAN_DINT	取消延时中断
SFC 34	QRY_DINT	询查延时中断
SFC 35	MP_ALM	触发多 CPU 中断
SFC 36	MSK_FLT	屏蔽同步故障
SFC 37	DMSK_FLT	解除同步故障屏蔽
SFC 38	READ_ERR	读故障寄存器
SFC 39	DIS_IRT	封锁新中断和非同步故障
SFC 40	EN_IRT	使能新中断和非同步故障
SFC 41	DIS_AIRT	延迟高优先级中断和非同步故障
SFC 42	EN_AIRT	使能高优先级中断和非同步故障
SFC 43	RE_TRIGR	再触发循环时间监控
SFC 44	REPL_VAL	传送替代值到累加器 1
SFC 46	STP	使 CPU 进入停机状态
SFC 47	WAIT	延迟用户程序的执行
SFC 48	SNC_RTCB	同步子时钟
SFC 49	LGC_GADR	查询一个逻辑地址的模块槽位的属性
SFC 50	RD_LGADR	查询一个模块的全部逻辑地址
SFC 51	RDSYSST	读系统状态表或部分表
SFC 52	WR_USMSG	向诊断缓冲区写用户定义的诊断事件
SFC 54	RD_PARM	读取定义参数
SFC 55	WR_PARM	写动态参数
SFC 56	WR_DPARM	写默认参数
SFC 57	PARM_MOD	为模块指派参数
SFC 58	WR_REC	写数据纪录
SFC 59	RD_REC	读数据纪录
SFC 60	GD_SND	全局数据包发送
SFC 61	GD_RCV	全局数据包接收
SFC 62	CONTROL	查询通讯 SFB 所属的确连接状态
SFC 63 *	AB_CALL	汇编代码块
SFC 64	TIME_TCK	读系统时间
SFC 65	X_SEND	向局域 S7 站之外的通讯伙伴发送数据
SFC 66	X_RCV	接收局域 S7 站之外的通讯伙伴发来的数据
SFC 67	X_GET	读取局域 S7 站之外的通讯伙伴的数据
SFC 68	X_PUT	写数据到局域 S7 站之外的通讯伙伴
SFC 69	X_ABORT	中止现存的与局域 S7 站之外的通讯伙伴的连接

编号	短名	功能
SFC 72	I_GET	读取局域 S7 站内的通讯伙伴
SFC 73	I_PUT	写数据到局域 S7 站内的通讯伙伴
SFC 74	I_ABORT	中止现存的与局域 S7 站内通讯伙伴的连接
SFC 78	OB_RT	决定 OB 的程序运行时间
SFC 79	SET	置位输出范围
SFC 80	RSET	复位输出范围
SFC 81	UBLKMOV	不可中断拷贝变量
SFC 82	CREA_DBL	在装载存储器中生成 DB 块
SFC 83	READ_DBL	读装载存储器中的 DB 块
SFC 84	WRIT_DBL	写装载存储器中的 DB 块
SFC 87	C_DIAG	实际连接状态的诊断
SFC 90	H_CTRL	H 系统中的控制操作
SFC 100	SET_CLKS	设日期时间和日期时间状态
SFC 101	RTM	处理时间计时器
SFC 102	RD_DPARA	读取预定义参数（重新定义参数）
SFC 103	DP_TOPO	识别 DP 主系统中总线的拓扑
SFC 104	CiR	控制 CiR
SFC 105	READ_SI	读动态系统资源
SFC 106	DEL_SI	删除动态系统资源
SFC 107	ALARM_DQ	生成可应答的块相关信息
SFC 108	ALARM_D	生成恒定可应答的块相关信息
SFC 126	SYNC_PI	同步刷新过程映象区输入表
SFC 127	SYNC_PO	同步刷新过程映象区输出表

\* SFC 63 “AB\_CALL” 仅在 CPU614 中存在。详细说明可参考相应的手册。



## 32.2 SFC 表，按字母排序

短名	编号	功能
AB_CALL *	SFC 63	汇编代码块
ACT_TINT	SFC 30	激活日时钟中断
ALARM_D	SFC 108	生成恒定可应答的块相关信息
ALARM_DQ	SFC 107	生成可应答的块相关信息
ALARM_S	SFC 18	生成恒定可应答的块相关信息
ALARM_SC	SFC 19	查询最后的 ALARM_SQ 到来状态信息的应答状态
ALARM_SQ	SFC 17	生成可应答的块相关信息
BLKMOV	SFC 20	拷贝变量
C_DIAG	SFC 87	实际连接状态的诊断
CAN_DINT	SFC 33	取消延时中断
CAN_TINT	SFC 29	取消日时钟中断
CiR	SFC 104	控制 CiR
COMPRESS	SFC 25	压缩用户内存
CONTROL	SFC 62	查询通讯 SFB 所属的确连接状态
CREA_DBL	SFC 82	在装载存储器中生成 DB 块
CREAT_DB	SFC 22	生成 DB
CTRL_RTM	SFC 3	运行时间计时器起/停
D_ACT_DP	SFC 12	取消和激活 DP 从站
DEL_DB	SFC 23	删除 DB
DEL_SI	SFC 106	删除动态系统资源
DIS_AIRT	SFC 41	延迟高优先级中断和非同步故障
DIS_IRT	SFC 39	封锁新中断和非同步故障
DIS_MSG	SFC 10	封锁块相关，符号相关和组状态的信息
DMSK_FLT	SFC 37	解除同步故障屏蔽
DP_PRAL	SFC 7	在 DP 主站上触发硬件中断
DP_TOPOL	SFC 103	识别 DP 主系统中总线的拓扑
DPNRM_DG	SFC 13	读 DP 从站的诊断数据（从站诊断）
DPRD_DAT	SFC 14	读标准 DP 从站的连续数据
DPSYC_FR	SFC 11	同步 DP 从站组
DPWR_DAT	SFC 15	写标准 DP 从站的连续数据
EN_AIRT	SFC 42	使能高优先级中断和非同步故障
EN_IRT	SFC 40	使能新中断和非同步故障

短名	编号	功能
EN_MSG	SFC 9	使能块相关, 符号相关和组状态的信息
FILL	SFC 21	初始化存储区
GADR_LGC	SFC 5	查询模板的逻辑起始地址
GD_RCV	SFC 61	全局数据包接收
GD_SND	SFC 60	全局数据包发送
H_CTRL	SFC 90	H 系统中的控制操作
I_ABORT	SFC 74	中止现存的与局域 S7 站内通讯伙伴的连接
I_GET	SFC 72	读取局域 S7 站内的通讯伙伴
I_PUT	SFC 73	写数据到局域 S7 站内的通讯伙伴
LGC_GADR	SFC 49	查询一个逻辑地址的模块槽位的属性
MP_ALM	SFC 35	触发多 CPU 中断
MSK_FLT	SFC 36	屏蔽同步故障
OB_RT	SFC 78	决定 OB 的程序运行时间
PARM_MOD	SFC 57	为模块指派参数
QRY_DINT	SFC 34	查询延时中断
QRY_TINT	SFC 31	查询日时钟中断
RD_DPARA	SFC 102	读取预定义参数 (重新定义参数)
RD_DPARM	SFC 54	读取定义参数
RD_LGADR	SFC 50	查询一个模块的全部逻辑地址
RD_REC	SFC 59	读数据纪录
RD_SINFO	SFC 6	读 OB 起动信息
RDSYSST	SFC 51	读系统状态表或部分表
RE_TRIGR	SFC 43	再触发循环时间监控
READ_CLK	SFC 1	读系统时钟
READ_DBL	SFC 83	SFC 83
READ_ERR	SFC 38	读故障寄存器
READ_RTM	SFC 4	运行时间计时器读取
READ_SI	SFC 105	读动态系统资源
REPL_VAL	SFC 44	传送替代值到累加器 1
RSET	SFC 80	复位输出范围
RTM	SFC 101	处理时间计时器
SET	SFC 79	置位输出范围
SET_CLK	SFC 0	设系统时钟
SET_CLKS	SFC 100	设日期时间和日期时间状态
SET_RTM	SFC 2	运行时间计时器设定
SET_TINT	SFC 28	设置日时钟中断

短名	编号	功能
SNC_RTCB	SFC 48	同步子时钟
SRT_DINT	SFC 32	起动延时中断
STP	SFC 46	使 CPU 进入停机状态
SYNC_PI	SFC 126	同步刷新过程映象区输入表
SYNC_PO	SFC 127	同步刷新过程映象区输出表
TEST_DB	SFC 24	测试 DB
TIME_TCK	SFC 64	读系统时间
UBLKMOV	SFC 81	不可中断拷贝变量
UPDAT_PI	SFC 26	刷新过程映象更新表
UPDAT_PO	SFC 27	刷新过程映象输出表
WAIT	SFC 47	延迟用户程序的执行
WR_DPARAM	SFC 56	写默认参数
WR_PARM	SFC 55	写动态参数
WR_REC	SFC 58	写数据纪录
WR_USMSG	SFC 52	向诊断缓冲区写用户定义的诊断事件
WRIT_DBL	SFC 84	写装载存储器中的 DB 块
X_ABORT	SFC 69	中止现存的与局域 S7 站之外的通讯伙伴的连接
X_GET	SFC 67	读取局域 S7 站之外的通讯伙伴的数据
X_PUT	SFC 68	写数据到局域 S7 站之外的通讯伙伴
X_RCV	SFC 66	接收局域 S7 站之外的通讯伙伴发来的数据
X_SEND	SFC 65	向局域 S7 站之外的通讯伙伴发送数据

\* SFC 63 “AB\_CALL” 仅在 CPU614 中存在。详细说明可参考相应的手册。

### 32.3 SFB 表，按编号排序

编号	短名	功能
SFB 0	CTU	增计数
SFB 1	CTD	减计数
SFB 2	CTUD	增/减计数
SFB 3	TP	脉冲定时
SFB 4	TON	延时接通
SFB 5	TOF	延时断开
SFB 8	USEND	非协调数据发送
SFB 9	URCV	非协调数据接收
SFB 12	BSEND	段数据发送
SFB 13	BRCV	段数据接收
SFB 14	GET	向远程 CPU 写数据
SFB 15	PUT	从远程 CPU 读数据
SFB 16	PRINT	向打印机发送数据
SFB 19	START	在远程装置上实施暖启动或冷启动
SFB 20	STOP	将远程装置变为停止状态
SFB 21	RESUME	在远程装置上实施热启动
SFB 22	STATUS	查询远程装置的状态
SFB 23	USTATUS	接收远程装置的状态
SFB 29 *	HS_COUNT	计数器（高速计数器，集成功能）
SFB 30 *	FREQ_MES	频率计（频率计，集成功能）
SFB 31	NOTIFY_8P	生成不带应答指示的块相关信息
SFB 32	DRUM	执行顺序器
SFB 33	ALARM	生成带应答显示的块相关信息
SFB 34	ALARM_8	生成不带 8 个信号值的块相关信息
SFB 35	ALARM_8P	生成带 8 个信号值的块相关信息
SFB 36	NOTIFY	生成不带应答显示的块相关信息
SFB 37	AR_SEND	发送归档数据
SFB 38 *	HSC_A_B	计数器 A/B（集成功能）
SFB 39 *	POS	定位（集成功能）
SFB 41	CONT_C <sup>1)</sup>	连续调节器
SFB 42	CONT_S <sup>1)</sup>	步进调节器
SFB 43	PULSEGEN <sup>1)</sup>	脉冲发生器

编号	短名	功能
SFB 44	ANALOG <sup>2)</sup>	带模拟输出的定位
SFB 46	DIGITAL <sup>2)</sup>	带数字输出的定位
SFB 47	COUNT <sup>2)</sup>	计数器控制
SFB 48	FREQUENC <sup>2)</sup>	频率计控制
SFB 49	PULSE <sup>2)</sup>	脉冲宽度控制
SFB 52	RDREC	读来自 DP 从站的数据纪录
SFB 53	WRREC	向 DP 从站写数据纪录
SFB 54	RALRM	接收来自 DP 从站的中断
SFB 60	SEND_PTP <sup>2)</sup>	发送数据 (ASCII, 3964(R))
SFB 61	RCV_PTP <sup>2)</sup>	接收数据 (ASCII, 3964(R))
SFB 62	RES_RECV <sup>2)</sup>	清除接收缓冲区 (ASCII, 3964(R))
SFB 63	SEND_RK <sup>2)</sup>	发送数据 (RK 512)
SFB 64	FETCH_RK <sup>2)</sup>	获取数据 (RK 512)
SFB 65	SERVE_RK <sup>2)</sup>	接收和提供数据 (RK 512)
SFB 75	SALRM	向 DP 从站发送中断

\* SFB 29 “HS\_COUNT” 和 SFB 30 “FREQ\_MES” 仅在 CPU 312 IFM 和 CPU 314 IFM 中存在。SFB 38 “HSC\_A\_B” 和 39 “POS” 仅在 CPU 314 IFM 中存在。  
详细说明可参考/73/。

- 1) SFB 41 “CONT\_C”，42 “CONT\_S” 和 43 “PULSEGEN” 仅在 CPU 314 IFM 中存在。
- 2) SFB 44 至 49 和 60 至 65 仅在 S7-300C CPU 中存在。

## 32.4 SFB 表，按字母排序

短名	编号	功能
ALARM	SFB 33	生成带应答显示的块相关信息
ALARM_8	SFB 34	生成不带 8 个信号值的块相关信息
ALARM_8P	SFB 35	生成带 8 个信号值的块相关信息
ANALOG <sup>2)</sup>	SFB 44	带模拟输出的定位
AR_SEND	SFB 37	发送归档数据
BRCV	SFB 13	段数据接收
BSEND	SFB 12	段数据发送
CONT_C <sup>1)</sup>	SFB 41	连续调节器
CONT_S <sup>1)</sup>	SFB 42	步进调节器
COUNT <sup>2)</sup>	SFB 47	计数器控制
CTD	SFB 1	减计数
CTU	SFB 0	增计数
CTUD	SFB 2	增/减计数
DIGITAL <sup>2)</sup>	SFB 46	带数字输出的定位
DRUM	SFB 32	执行顺序器
FETCH_RK <sup>2)</sup>	SFB 64	获取数据 (RK 512)
FREQ_MES *	SFB 30	频率计 (频率计, 集成功能)
FREQUENC <sup>2)</sup>	SFB 48	频率计控制
GET	SFB 14	向远程 CPU 写数据
HSC_A_B *	SFB 38	计数器 A/B (集成功能)
HS_COUNT *	SFB 29	计数器 (高速计数器, 集成功能)
NOTIFY	SFB 36	生成不带应答显示的块相关信息
NOTIFY_8P	SFB 31	生成不带应答指示的块的相关信息
POS *	SFB 39	定位 (集成功能)
PRINT	SFB 16	向打印机发送数据
PULSE <sup>2)</sup>	SFB 49	脉冲宽度控制
PULSEGEN <sup>1)</sup>	SFB 43	脉冲发生器
PUT	SFB 15	从远程 CPU 读数据
RALRM	SFB 54	接收来自 DP 从站的中断
RDREC	SFB 52	读来自 DP 从站的数据纪录
RCV_PTP <sup>2)</sup>	SFB 61	接收数据 (ASCII, 3964(R))
RES_RECV <sup>2)</sup>	SFB 62	清除接收缓冲区 (ASCII, 3964(R))

短名	编号	功能
RESUME	SFB 21	在远程装置上实施热启动
SALRM	SFB 75	向 DP 从站发送中断
SEND_PTP <sup>2)</sup>	SFB 60	发送数据 (ASCII, 3964(R))
SEND_RK <sup>2)</sup>	SFB 63	发送数据 (RK 512)
SERVE_RK <sup>2)</sup>	SFB 65	接收和提供数据 (RK 512)
START	SFB 19	在远程装置上实施暖启动或冷启动
STATUS	SFB 22	查询远程装置的状态
STOP	SFB 20	将远程装置变为停止状态
TOF	SFB 5	延时断开
TON	SFB 4	延时接通
TP	SFB 3	脉冲定时
URCV	SFB 9	非协调数据接收
USEND	SFB 8	非协调数据发送
USTATUS	SFB 23	接收远程装置的状态
WRREC	SFB 53	向 DP 从站写数据纪录

\* SFB 29 “HS\_COUNT” 和 SFB 30 “FREQ\_MES” 仅在 CPU 312 IFM 和 CPU 314 IFM 中存在。

SFB 38 “HSC\_A\_B” 和 39 “POS” 仅在 CPU 314 IFM 中存在。详细说明可参考/73/。

1) SFB 41 “CONT\_C”，42 “CONT\_S” 和 43 “PULSEGEN” 仅在 CPU 314 IFM 中存在。List of FCs.

2) SFB 44 至 49 和 60 至 65 仅在 S7-300C CPU 中存在。

# 术语

## 伴随值

可以随报文一起输出的值。它提供报文生成时的变量状态或/和地址的信息。

## ACCU（累加器）

累加器是 CPU 中的寄存器，用作装载、运输、比较、算术运算和转换操作的中间缓冲区。

## 实际参数

当调用功能块（FB）与功能（FC）时用实际参数代替形式参数，例如，用实际参数“13.6”代替形式参数“REQ”。

## 地址

地址是存储区域或存储区间的标识，例如：输入 I 12.1；位存储器 MW25；数据块 DB3。

## 寻址

在用户程序中分配地址。地址可分配给一个存储区域或存储区间（例如：输入 I 12.1，位存储器 MW25）。

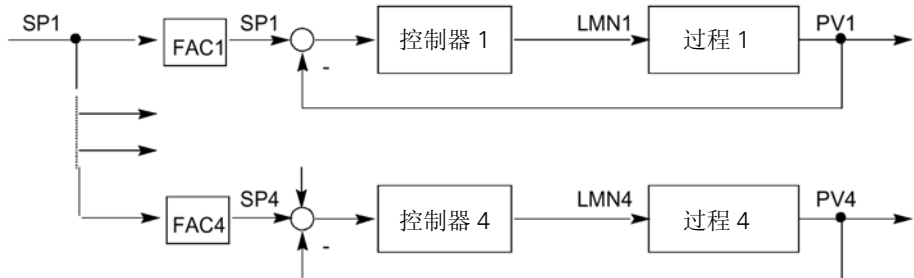
## 位存储器

为一个位存储单元。位存储器允许用 STEP 7 基本操作（寻址位、字节、字和双字）进行读写访问。

用户可以使用位存储器的地址区域存储中间结果。

## 混合控制

混合控制涉及一种控制结构，在这种控制结构中，设定值总和 SP 被转换为几个子设定值的百分数。混和系数 FAL 的总和必须为 1 (=100%)。



## 块相关的信息

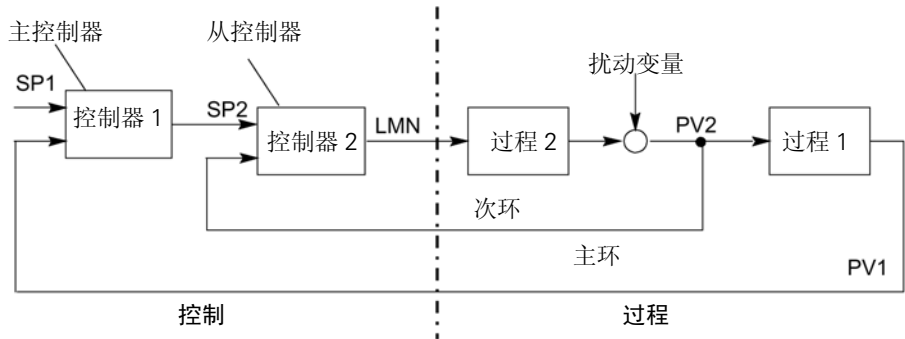
为具有信息能力的块（FB 或 DB）组态的信息。



### 串级控制

串级控制涉及一系列内部连接的控制器，其中主控制器依据主过程变量的瞬态偏差信号调节次（从）控制器的设定值。

一个串级控制系统还可以包含其他过程变量。从过程变量 PV2 的测量值与参考设定值（主控制器的输出 SP2）共同控制。主控制器控制过程变量 PV1 到达固定设定值 SP1，并且设置 SP2，以使控制过程尽可能快且没有超调。



### 闭环控制器

在闭环控制器中，连续地计算偏差信号，以使实际值快速且没有超调地跟随设定值，消除偏差。

### 通讯，双工

用通讯 SFB 进行数据交换时，需要区分双工与单工通讯。当本地和远程模板上都有 SFB 时为双工通讯，例如，通讯 SFB “USEND” 和 “URCV”。

### 用于组态的连接通讯 SFB

通讯 SFB 是用于数据交换和程序管理的系统功能块。

数据交换的例子：SEND，RECEIVE，GET。

程序管理的例子：把通讯伙伴的 CPU 设为 STOP 状态，查询通讯伙伴 CPU 的 STATUS。

### 用于取消组态的连接通讯 SFC

通讯 SFC 是用于数据交换和取消由通讯 SFC 建立起的已经存在的连接的系统功能。

### 通讯，单工

用通讯 SFB 进行数据交换时，需要区分双工与单工通讯。当 SFB 只位于本地模板上时为单工通讯，例如，SFB “GET”。

### 完全重新启动

CPU 启动（例如，模式选择由 STOP 转换到 RUN 或者电源开关合上），在循环程序（OB1）启动之前，首先执行组织块 OB101（重新启动：只在 S7-400 中）或者 OB100（完全再启动），之后，读入过程映象输入表，OB1 开始执行。

## 常数

“常数”在逻辑块中表示固定值。常数用于提高程序的易读性。例如，在系统功能块中输入符号“Max\_iteration\_count”代替直接输入值（例如 10）。这样当块调用时常数（例如，10）即被读入。

## 连续控制器

在连续控制器中，偏差信号的每次改变都会引起控制变量的变化。可以采用控制变量范围内的任何数值。

## 控制设备

用于控制过程变量的全部设备。包括一个控制器，一个受控装置（例如，执行器）和一个传感器（测量装置）。

## 控制环

控制环是过程输出（过程变量）与控制器输入之间，控制器输出（控制变量）与过程输入之间的连接，由此控制器与过程形成一个闭环。

## 控制系统组信息

当一个标明诊断事件进入诊断缓存区时 CPU 操作系统生成组信息。

## 控制器参数

控制器参数是控制器静态和动态响应给定控制环或者过程特性的特征值。

## CPU 操作系统

CPU 操作系统组织所有功能和没有与特殊控制任务链接的 CPU 程序。

## 数据块（DB）

数据块是用户程序中存储用户数据的存储区。共享数据块可以被所有逻辑块访问，背景数据块可以被与其有联系的特殊功能块（FB）调用。

## 诊断缓存区

诊断缓存区是指 CPU 中可以按诊断事件发生的顺序进行诊断事件保存的一个缓冲存储区。

## 诊断数据

诊断数据是故障消息（诊断事件，时间标识）中包含的信息。

## 诊断进入

诊断事件在诊断缓存区中以诊断进入来描述。

## 诊断中断

使用诊断中断功能，具有诊断能力的模块可以将检测到的系统故障事件报告给 CPU。

### 诊断信息

诊断信息由处理过的诊断事件构成，由 CPU 传送到显示设备。

### 诊断

诊断功能涵盖了所有的系统诊断，包括 PLC 中故障的识别、编译、报告。

### 显示设备

显示处理结果的设备。

### 故障，异步

异步故障为不占用用户程序中空间运行时间故障（例如，电源故障，扫描时间过长）。当这些故障发生时，操作系统调用相应的组织块，用户可在相应的组织块中编程对故障的响应，参见用 OB 进行故障处理。

如果系统识别出一个特殊故障（例如，S7 访问故障），操作系统会调用 CPU 响应此故障的特定的组织块。

### 故障 OB

故障 OB 是用户可以用来编制故障响应程序的组织。但是，只有当故障不引起 PLC 停机的情况下才可能发生编程的响应。每类故障都有一个故障 OB。（例如，用于寻址故障的故障 OB，用于 S7 访问故障的故障 OB）。

### 故障响应

对运行时间故障的响应。操作系统可以下面的方式响应：将 PLC 转换到 STOP 状态，调用用户编程的故障组织块，或者显示故障。

### 故障，同步

同步故障是占用用户程序空间的运行时间故障（例如，I/O 模板访问故障）。故障发生时，操作系统调用相应的组织块，用户可以在相应的组织块中编程对故障的响应。

### 故障，系统故障

系统故障为可能在 PLC 中发生的故障（不是在处理过程中发生的故障）。系统故障可以是，例如，CPU 中的程序故障和模板故障。

### 形式参数

形式参数与逻辑块中实际参数相对应。在功能块（FB）和功能（FC）中，由用户声明形式参数，在系统功能块 SFB 和系统功能 SFC 中，形式参数亦可以使用。当调用块时，会把实际参数赋给形式参数以使调用块采用最后的值，形式参数属于局部数据，可以是输入，输出或输入/输出参数。

### 组故障

在（只在）S7-300 模板的面板上由 LED 显示的故障信息，当有关模板中出现故障（内部故障和外部故障）时 LED 灯亮。

**硬件中断**

硬件中断由具有诊断能力的模板触发，作为过程中特殊事件的结果，硬件中断报告给 CPU。根据此中断的优先级执行相应的组织块。

**输入参数**

输入参数只存在于功能和功能块中，借助于输入参数，数据传送到调用执行的块。

**指令**

指令（STEP5 或者 STEP7）是用源语言生成的最小程序单元，它代表一个处理命令。

**积分元件**

控制器的积分元件

在过程变量（或偏差信号）发生一个阶跃变化后，输出变量依据积分系数  $K1 (=1/T1)$  成比例地变换比率斜坡上升。闭环控制中积分元件可明确控制输出变量消除静差。

**集成控制器**

集成控制器是操作系统中已编程的可用的控制器块，它环控制应用最重要的功能，用户可以通过软件开关选择此功能。

**完全再启动**

CPU 启动模式。完全再启动时，由 SFC 生成的数据块将从工作存储器中删除，其他数据块恢复缺省值同装载存储器中的相同。过程映像定时器，计数器和存储位都被复位—而与其余的存储器组态无关。在完全再启动过程中，CPU 执行 OB102，读取过程映像输入表，然后连续地执行用户程序，由 OB1 的第一条指令开始。

**中断**

SIMATIC S7 优先级系统可以识别 10 种不同的优先级，从控制用户程序的执行，中断属于这些优先级，例如，硬件中断，当有一个中断触发时，操作系统会自动调用一个特定的组织块，在该组织块中，用户可以编程所需的响应（例如在一个 FB 中）。

**中断，时间日期**

时间日期中断属于 SIMATIC S7 中程序执行的优先级之一。它根据特定的日期和时间（例如，9: 50 或每小时或每分钟）生成。然后，执行相应的组织块。

**中断，延时**

延时中断属于 SIMATIC S7 中程序执行的优先级之一。在达到用户程序中的时间时生成。然后执行相应的组织块。

## 逻辑块

逻辑块是 SIMATIC S7 中的块，它包含 STEP7 用户程序的一部分。另一种是数据块，它只包含数据，有以下几种类型的逻辑块。

- 组织块（OB）
- 功能块（FB）
- 功能（FC）
- 系统功能块（SFB）
- 系统功能（SFC）

## 报文

事件发生的报告。报文可以输送到一个正确组态了的显示设备，它包含优先级，位置，报文事件的时间以及状态改变（进入状态/离开状态）等信息。

## 报文组态

报文组态指以其文本和属性生成和编辑报文和报文模板，与下列信息有关：块相关的信息、符号相关的信息与诊断信息。

## 报文号

分配给报文的唯一号，用于识别，比如确认操作。

## 模板参数

模板参数是用于设定模板响应的参数，依据模板的不同，一些参数可在用户程序中修改。

## OB1

组织块 OB 是循环执行的程序，是系统程序的用户接口。

## OB 优先级

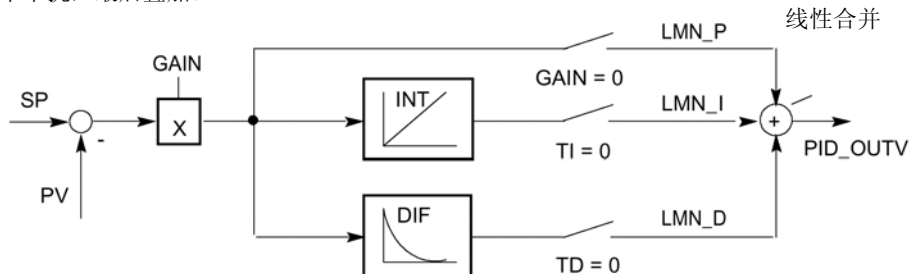
CPU 操作系统可以识别不同的优先级，例如循环程序执行，由硬件中断控制的程序执行等，每个组织块（OB）都赋给一个优先级，在组织块中，S7 用户可以编程一个响应。OB 具有不同的优先级，以确定在两个事件同时发生时执行的正确顺序，并且允许高优先级中断低优先级。C7 用户可以改变正常的优先级。

## 组织块（OB）

组织块是 CPU 的操作系统与用户程序之间的接口。用户程序的执行顺序在组织块中指定。

## 并行结构

并行结构是控制器中信号处理（算术处理）的一种特殊型式，P、I 或 D 元件单独计算，互不干扰，最后叠加。



## 参数

1. 一个 S7 逻辑块的变量。  
用于声明模块响应的变量（每个模板一个或几个变量）
2. 所有模块在断时都进行了设置，使用 STEP7，在组态中可以修改。  
（每个模板一个或几个变量）。  
有两种类型的参数。  
静态参数和动态参数。

## 参数，动态

与静态参数不同。模块的动态参数可以在调用 SFC 过程中通过用户程序改变。例如，一个模拟量输入模板的极限值。

## 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修改，但是可以改变 STEP7 中的组态，例如，一个输字量输入模板的输入延迟。

## P 算法

计算输出信号的算法，其中偏差信号与控制变量变化呈比例关系。

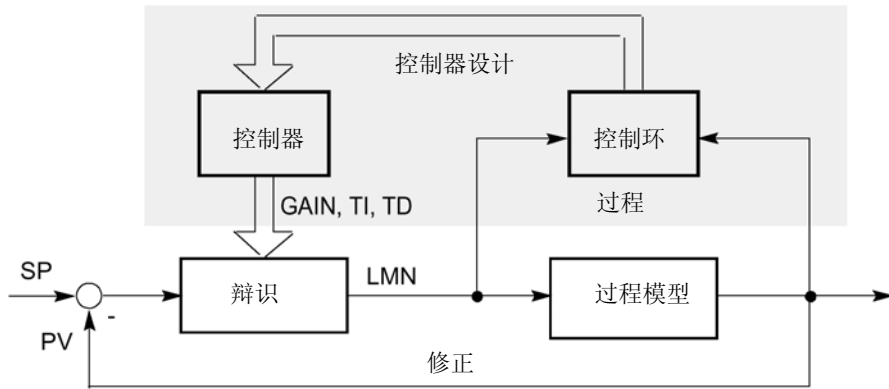
特点：不能消除稳态偏差，不能用于包含死区的控制系统。

## PI 算法

计算输出信号的算法，其中 P 元件使控制变量变化与偏误差信号呈比例关系，I 元件使控制变量与偏差信号和时间成比例。特点：无静差，与 I 算法相比有快速补偿的作用，适用于所有过程控制。

### PID 算法

计算输出信号的算法，由偏差信号的比例，积分，微分组成。PID 算法为并行结构。  
 特点：可达到较高的控制精度，死区不大于其他时间常数。



### 优先权

给一个组织块分配优先权时，就决定了当前用户程序的中断能力，高优先级事件可中断低优先级事件。

### 优先级

CPU 操作系统提供最多 28 个优先级，这些优先级可以分配给组织块。  
 优先级可以决定哪一个 OB 有权中断其他 OB。如果一个优先级中包含几个 OB。他们之间不能相互中断，但是，可以顺序执行。

### 过程

过程是系统的一部分，控制变量影响过程变量（通过改变能量或者质量）。过程可以分为执行器和实际受控对象两部分。



### 程序执行，事件控制

由于事件控制程序的执行，循环执行的用户程序被启动事件（优先级）中断。如果一个启动事件发生，当前执行的块在下一条指令前中断，一个指定的组织块调用并执行，之后循环程序在中断点继续执行。

## 比例执行器

### 脉冲持续调制

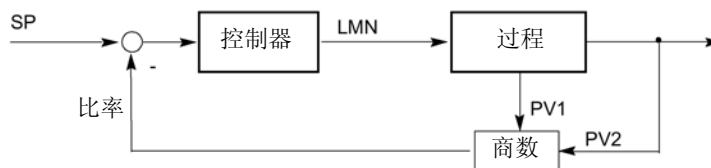
## 脉冲持续调制

脉冲持续调制是一种以不连续的输出来调节控制变量的方法。计算的控制值的百分数比例地转换为信号脉冲时间  $TP$ ，调节控制变量输出。例如， $100\%TP=TA$  或者  $=CYCLE$

## 比率控制

- 单环比率控制器

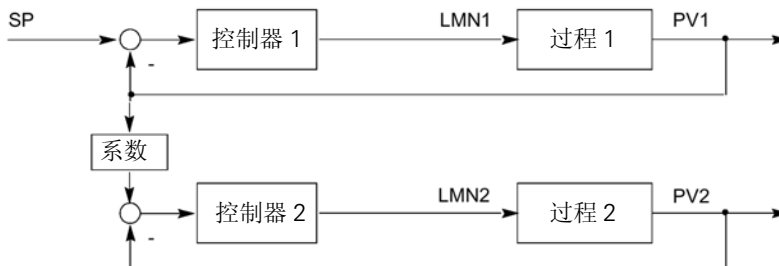
当控制两个过程变量的比率比绝对值更重要时采用单一环比率控制器。



- 多环比率控制器

在多环比率控制器中，过程变量  $PV1$  与  $PV2$  的比率必须保持为一个常数。

所以第二个控制环的设定值是由第一个控制环的过程变量计算得来。即使是在过程变量  $PV1$  动态变化时，此比率仍不变。



## 远程设备

远程设备是可以在网络上得到的设备，比如召唤，打印机或电脑。与本地设备不同，它列在安装时必须分配一个网络地址。

## 重新启动

CPU 启动（例如，模式选择由 STOP 转换到 RUN 或者电源开关合上），在循环程序（OB1）启动之前，首先执行组织块 OB100（重新启动）或组织块 OB101（重新启动；只在 S7-400 中）或者组织块 OB102（完全再启动）。然后读入过程映像输入表，STEP 用户程序从上次停机（STOP，掉电）时的中断点开始执行。

## 逻辑操作结果（RLO）

逻辑操作结果（RLO）是处理器中的当前信号状态，用于进一步处理二进制信号，上次 RLO 的信号状态决定一些操作是否执行。



### 运行时间故障

在 PLC 中（不是在过程中）用户程序的执行过程中产生的故障。

### 扫描

集成在 CPU 中的操作系统功能，用于在设定的时间间隔扫描和检测信号，以确定是否发生了信号改变。

### 标准功能

标准功能是 SIEMENS 用于执行复杂任务的功能块。

### 标准功能块

标准功能块是 SIEMENS 用于执行复杂任务的功能块。

### 启动事件

启动事件是定义的事件，诸如故障和中断等，它令操作系统立即调用相应的组织块。

### 启动事件信息

启动事件信息是组织块的一部分。启动事件信息提供给 S7 用户关于触发 OB 调用的事件的详细信息。启动事件信息包括事件号（由事件种类和事件 ID 组成），事件时间标识，和一些附加信息（例如，中断激活的信号模板的地址）。

### 启动信息

操作系统调用组织块的同时传送启动信息，这些信息可在用户程序中得到。

### 启动 OB

取决于启动模式的不同选择（只 S7-400），在启动时（停电后又送电，手动将模式选择由 STOP 切换到 RUN 或者在编程器上发出运行命令）操作系统或者调用启动组织块“完全再启动”或者调用启动组织块“再启动”（只存在于 S7-400）。在启动 OB 中 SIMATIC S7 用户可以编程停电后系统再启动的方式。

### 语句表

语句表是 STEP7 的汇编语言。程序在 STL 中处理时，不同的指令依照 CPU 执行的顺序依次响应程序。

### STEP7

对于 SIMATIC S7 控制器生成用户程序的编程软件。

### STEP7 编程语言

SIMATIC S7 控制器的编程语言。S7 程序员可以使用三种典型型式：a) 语句表，b) 控制系统流程图，c) 梯形逻辑

### 步控器

步控器类似于连续控制器，但它的输出是不连续的（是带有积分作用的电机驱动执行器），执行器发生三步响应，例如：上升—停止—下降（或者打开—保持—关闭）。（三步控制器）

**STL**

语句表

**子号码**

如果一个信息块可以监视一个以上信号，子号码表示监视信号的号。

**符号相关的信息**

符号表中符号（输入，输出，位存储器，数据块）的组态信息，在组态过程中，必须设置监视信号的扫描时间。

**符号编程**

STEP7 编程语言允许用户使用符号名代替 STEP7 地址。例如，可用“ValVe 17”STEP7 代替地址“Q1.1”。

STEP7 中的符号表把实际地址与符号名联系到一起。

**系统诊断**

系统诊断事件的检测与分析。

**系统诊断事件**

CPU 诊断缓存区进入的事件，用于操作系统初始化。

**系统功能（SFC）**

系统功能（SFC）是集成在 CPU 操作系统中的功能，如果需要，STEP7 用户程序可以调用它。

**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，如果需要，STEP7 用户程序可以调用它。

**3 步控制器**

只允许有 3 种离散状态的控制器；例如，“热—停—冷”或者“右—停—左”（步控器）。

**延时中断**

延时中断属于 SIMATIC S7 中程序执行的优先级之一，当指定的时间到时，用户程序中中断生成，之后，执行相应的组织块。

**工具**

用于组态和编程的软件功能

**两步控制器**

只允许设置两种控制变量状态的控制器（例如，启动—停止）

#### 用户定义诊断

用户定义的诊断事件的检测和分析。

#### 用户定义的诊断事件

由用户检测的诊断事件，可以装入诊断缓存区（通过 SFC52）

#### 用户定义的诊断信息

报告用户定义的诊断事件发生的信息。

#### 用户程序

用户程序包含用于信号处理的所有语句，声明和数据，这些信号可以用来控制一个系统或过程，它分配给可编程模块（模块，可编程的）并且可以是最小单元（块）结构。

#### 用户程序故障

SIMATIC S7PLC（与过程故障对照）中在用户程序处理过程中产生的故障。  
操作系统调用故障 OB（优先级系统）。状态字和系统功能中的输出参数来处理故障。

#### 变量

变量用于定义数据，可在 STEP7 用户程序中使用。变量由地址（例如，M3.1）和数据类型（例如，BOOL）组成，并可以用符号表示（例如，MOTOR—ON）。

变量声明