



SIMATIC S7

Системные и стандартные функции для S7-300 и S7-400

Справочное руководство

Это руководство является частью пакета документации с заказным номером:

6ES7810-4CA05-8BR0

C79000-G7076-C567-01

Важные замечания, содержание	
Организационные блоки	1
Общие параметры для SFC	2
Функции копирования и манипулирования блоками	3
SFC для управления исполнением программы	4
SFC для манипулирования системными часами	5
SFC для манипулирования счетчиками рабочего времени	6
SFC для передачи записей данных	7
SFC для манипулирования прерываниями по времени	8
SFC для манипулирования прерываниями с задержкой	9
SFC для обработки синхронных ошибок	10
SFC для обработки прерываний и асинхронных ошибок	11
SFC для диагностики	12
SFC и SFB для обновления образа процесса и обработки битовых массивов	13
Системные функции для адресации модулей	14
SFC для децентрализованной периферии	15
SFC для связи с помощью глобальных данных	16
Коммуникационные SFB для спроектированных соединений	17
Коммуникационные SFC для непроектируемых соединений S7	18
Создание сообщений, связанных с блоками	19
Таймеры и счетчики IEC	20
Функции IEC	21
SFB для встроенного регулирования	22
SFC для H CPU	23
Встроенные функции (для CPU с встроенными входами/выходами)	24
Гибкая технология	25
Диагностические данные	26
Списки состояний системы (SSL)	27
События	28
Список SFC, SFB и FC	29
Глоссарий	
Литература	
Предметный указатель	

Указания по безопасности

Это руководство содержит указания, которые вы должны соблюдать для обеспечения собственной безопасности, а также защиты продукта и подключенного оборудования. Эти указания выделены в руководстве предупреждающим треугольником и помечены следующим образом в соответствии с уровнем опасности:



Опасность

Указывает, что несоблюдение надлежащих предосторожностей приведет к смерти, тяжким телесным повреждениям или существенному повреждению имущества.



Предупреждение

Указывает, что несоблюдение надлежащих предосторожностей может привести к смерти, тяжким телесным повреждениям или существенному повреждению имущества.



Предостережение

Указывает, что несоблюдение надлежащих предосторожностей может привести к небольшим телесным повреждениям или порче имущества.

Замечание

Привлекает ваше внимание к особенно важной информации о продукте, обращении с продуктом или к определенной части документации.

Квалифицированный персонал

К установке и работе на данном оборудовании должен допускаться только квалифицированный персонал. К квалифицированному персоналу относятся лица, имеющие право пускать в эксплуатацию, заземлять и маркировать электрические цепи, оборудование и системы в соответствии с установленным порядком и стандартами.

Правильное использование

Примите во внимание следующее:



Предупреждение

Это устройство и его компоненты могут быть использованы только для приложений, описанных в каталоге или технических описаниях, и только в соединении с устройствами или компонентами других производителей, которые были одобрены или рекомендованы фирмой Siemens.

Этот продукт может правильно и безопасно функционировать только при правильной транспортировке, хранении, установке и установке, а также эксплуатации и обслуживании в соответствии с рекомендациями.

Торговые марки

SIMATIC®, SIMATIC HMI® и SIMATIC NET® являются зарегистрированными торговыми марками SIEMENS AG.

Некоторые из других обозначений, использованных в этих документах, также являются зарегистрированными торговыми марками; права собственности могут быть нарушены, если эти обозначения используются третьей стороной для своих собственных целей.

Copyright © Siemens AG 1998 Все права сохраняются

Воспроизведение, передача или использование этого документа или его содержания не допускается без специального письменного разрешения. Нарушители будут нести ответственность за нанесенный ущерб. Все права, включая права, создаваемые патентным грантом или регистрацией сервисной модели или проекта, сохраняются.

Siemens AG
Департамент техники автоматизации и приводов
Сфера деятельности: промышленные системы автоматизации
п/я 4848, D- 90327 Нюрнберг

Отказ от ответственности

Мы проверили содержание этого руководства на соответствие с описанной аппаратурой и программным обеспечением. Так как отклонения не могут быть полностью предотвращены, мы не гарантируем полного соответствия. Однако данные, приведенные в этом руководстве, регулярно пересматриваются и необходимые исправления вносятся в последующие издания. Приветствуются предложения по улучшению.

©Siemens AG 1998
Технические данные могут изменяться.

Важные замечания

Назначение

Это руководство дает полный обзор содержащихся в операционных системах CPU S7-300 и S7-400 организационных блоков (OB), системных функций (SFC), системных и стандартных функциональных блоков (SFB), а также функций IEC. В приложении описаны диагностические данные, списки состояний системы (SSL) и события

Замечание

Какие из этих функций и блоков в каких CPU доступны, вы можете узнать из справочной части руководства "S7-300 Programmable Controller, Hardware and Installation [Программируемый контроллер S7-300. Аппаратные средства и монтаж]" [/70/](#), или из справочного руководства "S7-400/M7-400 Programmable Controllers Module Specifications [Программируемые контроллеры S7-400/M7-400, Данные модулей]" [/101/](#), или из "Instruction List: S7-400 Programmable Controller [Список команд: Программируемый контроллер S7-400]" [/102/](#) (в зависимости от того, какая версия относится к вашему CPU). Свойства CFB и функций передачи сигналов S7 вы найдете в [/70/](#) и [/101/](#).

Информацию об операционных системах CPU, о разработке программ и о коммуникационных и диагностических возможностях CPU вы получите в руководстве "Configuring Hardware and Communication Connections STEP 7 V5.0 [Конфигурирование аппаратуры и проектирование соединений с помощью STEP 7 V5.0]" [/234/](#). Как вызвать функции и функциональные блоки в вашей программе, объясняется в описаниях языков. Все названные функции программируются и параметрируются с помощью стандартного программного обеспечения STEP 7. Как использовать это программное обеспечение, описано в руководстве "Programming with STEP 7 V5.0 [Программирование с помощью STEP 7 версии 5.0]" [/231/](#) и в оперативной помощи STEP 7.

Круг читателей

Это руководство предназначено для программистов и инженеров, которые знакомы с процессами управления и которые отвечают за написание программ для программируемых логических контроллеров.

Пакеты документации STEP 7

Это руководство является частью пакета "STEP 7 Basic Information [Базовая информация о STEP 7]".

В следующей таблице представлен обзор документации по STEP 7:

Документация	Назначение	Номер для заказа
Базовая информация о STEP 7, включающая в себя: <ul style="list-style-type: none"> Working with STEP 7 V5.0, Getting Started Manual [Работа со STEP 7 версии 5.0. Введение в STEP 7] Programming with STEP 7 V5.0 [Программирование с помощью STEP 7 версии 5.0] Configuring Hardware and Communication Connections, STEP 7 V5.0 [Конфигурирование аппаратуры и проектирование соединений с помощью STEP 7 v5.0] From S5 to S7, Converter Manual [От S5 к S7. Руководство по конвертированию] 	Базовая информация для технического персонала, описывающая методы реализации задач управления с помощью STEP 7 и программируемых контроллеров S7-300/400.	6ES7810-4CA04-8BA0
Справочники по STEP 7, в том числе <ul style="list-style-type: none"> Руководства Ladder Logic (LAD) /Function Block Diagram (FBD) /Statement List (STL) for S7-300/400 [Контактный план (LAD, KOP) /Функциональный план (FBD, FUP) /Список операторов (STL, AWL) для S7-300/400] Standard and System Functions for S7-300/400 [Стандартные и системные функции для S7-300/400] 	Предоставляется справочная информация и описываются языки программирования LAD (контактный план, KOP), FBD (функциональный план, FUP) и STL (список операторов, AWL) и стандартные и системные функции, расширяя объем базовой информации о STEP 7.	6ES7810-4CA04-8BR0

Оперативные справки	Назначение	Номер для заказа
Помощь по STEP 7	Базовая информация о программировании и конфигурировании аппаратуры с помощью STEP 7 в виде оперативной справки (online).	Часть стандартного программного обеспечения STEP 7.
Справочная информация о STL/LAD/FBD Справочная информация о SFB/SFC Справочная информация об организационных блоках	Контекстно-чувствительная справочная информация	Часть стандартного программного обеспечения STEP 7.

Оперативная помощь

Руководство дополняется оперативной помощью, встроенной в программное обеспечение. Эта оперативная помощь направлена на то, чтобы снабдить вас детальной поддержкой при использовании программного обеспечения. Система помощи встроена в программное обеспечение через ряд интерфейсов:

- Имеется несколько команд меню, которые вы можете выбрать в меню **Help [Помощь]**:
 - Команда **Contents [Содержание]** открывает предметный указатель помощи по Step 7.
 - **Using Help [Использование помощи]** дает подробные инструкции по использованию оперативной помощи.

- Контекстно-чувствительная помощь предлагает информацию о текущем контексте, например, об открытом диалоговом окне или об активном окне. Контекстно-чувствительную помощь можно открыть, щелкнув на кнопке "Help [Помощь]" или нажав F1.
- Строка состояния предлагает еще одну форму контекстно-чувствительной помощи. Она отображает краткое объяснение для каждой команды меню, когда указатель мыши расположен на этой команде.
- Краткое объяснение отображается также для каждой пиктограммы на панели инструментов, когда указатель мыши кратковременно помещается на этой пиктограмме.

Если вы предпочитаете читать информацию, предоставляемую оперативной помощью, в печатном виде, то вы можете распечатать отдельные темы помощи, книги или всю оперативную помощь.

Это руководство является извлечением из помощи по STEP 7, основанной на гипертекстовом языке описания документов HTML. Так как данное руководство и оперативная помощь имеют почти одинаковую структуру, то не составляет труда переключаться между руководством и оперативной помощью.

Другие руководства

Различные CPU S7-300 и S7-400, а также модули S7-300 и S7-400 описаны в следующих руководствах:

- для программируемых логических контроллеров S7-300: „S7-300 Programmable Controller, Hardware and Installation [Программируемый контроллер S7-300. Аппаратные средства и монтаж]“ [/70/](#), „S7-300, M7-300 Programmable Controllers Module Specifications [Спецификации модулей программируемых логических контроллеров S7-300, M7-300]“ [/71/](#) и в списке команд [/72/](#).
- для программируемых логических контроллеров S7-400: „S7-400/M7-400 Programmable Controllers Module Specifications [Программируемые контроллеры S7-400/M7-400, Данные модулей]“ [/101/](#) и в списке команд [/102/](#).

Как пользоваться этим руководством

Это руководство включает в себя следующие темы:

- В главе 1 объясняются функции всех организационных блоков.
- Глава 2 описывает общие параметры RET_VAL, REQ и BUSY.
- Главы с 3 по 22 описывают SFC, SFB и функции IEC.
- Приложения от A до F содержат описание структуры диагностических данных, обзор идентификаторов списков состояний системы (SSL-ID), возможные события, списки SFC, SFB и FC, описанных в этом руководстве, обзор SDB и список литературы, использованной в этом руководстве.
- В глоссарии объяснены важные термины.
- Предметный указатель поможет вам быстро найти нужные разделы текста и темы.

Соглашения

Ссылки на другие руководства и документацию даются с помощью номеров, заключенных между косыми чертами /.../. Эти номера относятся к названиям руководств, перечисленных в приложении F.

Специальное указание

Системные функции могут прерываться. Если имеются какие-либо ограничения, относящиеся к определенным SFC или ситуациям, то они объясняются в описаниях конкретных SFC.

Учебные центры SIMATIC

Фирма Siemens предлагает ряд учебных курсов для ознакомления с системой автоматизации SIMATIC S7. Для получения более подробной информации обращайтесь в свой региональный учебный центр или в центральный учебный центр Нюрнберге, Германия (D-90327 Nuremberg, Germany):

Телефон: +49 (911) 895-3154.

Горячая линия поддержки клиентов SIMATIC

Открыта круглые сутки по всему миру:



Нюрнберг

Основная горячая линия SIMATIC

Мест. время: пн-пт 7:00 – 17:00

Тел.: +49 (911) 895-7000

Факс: +49 (911) 895-7002

E-mail: simatic.support@Nbgm.siemens.de

Время по Гринвичу: +1:00

Платная горячая линия SIMATIC

(Вызовы оплачиваются только с помощью SIMATIC Card)

Время: пн-пт 0:00 – 24:00

Тел.: +49 (911) 895-7777

Факс: +49 (911) 895-7001

Время по Гринвичу: +01:00

Джонсон-Сити

Основная горячая линия SIMATIC

Мест. время: пн-пт 8:00 – 17:00

Тел.: +1 423 461-2522

Факс: +1 423 461-2231

E-mail: simatic.hotline@sea.siemens.com

Время по Гринвичу: -5:00

Горячая линия авторизации SIMATIC

Мест. время: пн-пт 8:00 – 17:00

Тел.: +49 (911) 895-7200

Факс: +49 (911) 895-7201

E-mail: authorizarion@Nbgm.siemens.de

Время по Гринвичу: +01:00

Сингапур

Основная горячая линия SIMATIC

Мест. время: пн-пт 8:30 – 17:30

Тел.: +65 740-7000

Факс: +65 740-7001

E-mail: simatic@singnet.com.sg

Время по Гринвичу: +8:00

Оперативные службы поддержки клиентов SIMATIC

Бригада поддержки клиентов SIMATIC предлагает существенную дополнительную информацию о продуктах SIMATIC через свои оперативные службы:

- Общая текущая информация может быть получена:
 - в **Internet** через <http://www.ad.siemens.de/simatic>
 - через **Fax-Polling** номер 08765-93 02 77 95 00
- Текущие данные о продукте и загрузки, которые вы, возможно, найдете полезными, доступны:
 - в **Internet** через http://www.ad.siemens.de/support/html_00/
 - через **Bulletin Board System** (BBS) в Нюрнберге (*SIMATIC Customer Support Mailbox*) под номером +49 (911) 895-7100.

Для набора почтового ящика используйте модем с протоколом до V.34 (28,8 кБод), параметры которого установите следующим образом: 8, N, 1, ANSI; или через ISDN (x.75, 64 кБод).

Содержание

Важные замечания

iii

Содержание

ix

1	Организационные блоки	1-1
1.1	Обзор организационных блоков (OB)	1-1
1.2	Организационный блок циклического выполнения программы (OB1)	1-4
1.3	Организационные блоки прерываний по времени (OB10 – OB17)	1-6
1.4	Организационные блоки прерываний с задержкой (OB20 – OB23)	1-10
1.5	Организационные блоки циклических прерываний (OB30 – OB38)	1-12
1.6	Организационные блоки аппаратных прерываний (OB40 – OB47)	1-14
1.7	Организационный блок мультипроцессорных прерываний (OB60)	1-16
1.8	ОВ ошибок резервирования входов/выходов (OB70)	1-18
1.9	ОВ ошибок резервирования CPU (OB72)	1-20
1.10	Организационный блок ошибок времени (OB80)	1-22
1.11	Организационный блок неисправностей источника питания (OB81)	1-24
1.12	Организационный блок диагностических прерываний (OB82)	1-26
1.13	Организационный блок снятия/установки модулей (OB83)	1-28
1.14	Организационный блок аппаратных ошибок CPU (OB84)	1-30
1.15	Организационный блок ошибок класса приоритета (OB85)	1-31
1.16	Организационный блок неисправностей стоек (OB86)	1-34
1.17	Организационный блок коммуникационных ошибок (OB87)	1-37
1.18	Организационный блок фонового режима (OB90)	1-39
1.19	Организационные блоки запуска (OB100, OB101 и OB102)	1-41
1.20	Организационный блок ошибок программирования (OB121)	1-45
1.21	Организационный блок ошибок доступа к периферии (OB122)	1-48
2	Общие параметры для SFC	2-1
2.1	Оценка ошибок с помощью выходного параметра RET_VAL	2-2
2.2	Значение параметров REQ, RET_VAL и BUSY у асинхронных SFC	2-6
3	Функции копирования и манипулирования блоками	3-1
3.1	Копирование переменных с помощью SFC20 "BLKMOV"	3-2
3.2	Непрерываемое копирование переменных с помощью SFC81 "UBLKMOV"	3-4
3.3	Инициализация области памяти с помощью SFC21 "FILL "	3-6
3.4	Создание блока данных с помощью SFC22 "CREAT_DB"	3-8
3.5	Удаление блока данных с помощью SFC23 "DEL_DB"	3-10
3.6	Тестирование блока данных с помощью SFC24 "TEST_DB"	3-11
3.7	Сжатие памяти пользователя с помощью SFC25 "COMPRESS"	3-12

3.8	Передача заменяющего значения в аккумулятор 1 с помощью SFC44 "REPL_VAL"	3-14
4	SFC для управления исполнением программы	4-1
4.1	Повторный запуск контроля времени цикла с помощью SFC43 "RE_TRIGR"	4-1
4.2	Перевод CPU в STOP с помощью SFC46 "STP"	4-2
4.3	Задержка исполнения программы пользователя с помощью SFC47 "WAIT"	4-3
4.4	Запуск прерывания многопроцессорной обработки с помощью SFC35 "MP_ALM"	4-4
5	SFC для манипулирования системными часами	5-1
5.1	Установка времени с помощью SFC0 "SET_CLK"	5-2
5.2	Считывание времени с помощью SFC1 "READ_CLK"	5-3
5.3	Синхронизация ведомых часов с помощью SFC48 "SNC_RTCB"	5-4
6	SFC для манипулирования счетчиками рабочего времени	6-1
6.1	Счетчики рабочего времени	6-2
6.2	Установка счетчика рабочего времени с помощью SFC2 "SET_RTM"	6-3
6.3	Запуск и останов счетчика рабочего времени с помощью SFC3 "CTRL_RTM"	6-4
6.4	Считывание счетчика рабочего времени с помощью SFC4 "READ_RTM"	6-5
6.5	Считывание системного времени с помощью SFC64 "TIME_TCK"	6-6
7	SFC для передачи записей данных	7-1
7.1	Запись и чтение записей данных	7-2
7.2	Чтение определенных параметров с помощью SFC54 "RD_DPARM"	7-4
7.3	Запись динамических параметров с помощью SFC55 "WR_PARM"	7-5
7.4	Запись параметров, установленных по умолчанию, с помощью SFC56 "WR_DPARM"	7-7
7.5	Назначение параметров модулю с помощью SFC57 "PARM_MOD"	7-8
7.6	Внесение записи данных с помощью SFC58 "WR_REC"	7-11
7.7	Чтение записи данных с помощью SFC59 "RD_REC"	7-12
7.8	Чтение записи данных с помощью SFC59 "RD_REC" в CPU S7-300	7-17
7.9	Дополнительная информация об ошибках SFC 55–59	7-20
8	SFC для манипулирования прерываниями по времени	8-1
8.1	Манипулирование прерываниями по времени	8-2
8.2	Характеристики SFC 28 – 31	8-3
8.3	Установка прерывания по времени с помощью SFC28 "SET_TINT"	8-5
8.4	Отмена прерывания по времени с помощью SFC29 "CAN_TINT"	8-6
8.5	Активизация прерывания по времени с помощью SFC30 "ACT_TINT"	8-7
8.6	Опрос прерывания по времени с помощью SFC31 "QRY_TINT"	8-8
9	SFC для манипулирования прерываниями с задержкой	9-1
9.1	Манипулирование прерываниями с задержкой	9-2
9.2	Запуск прерывания с задержкой с помощью SFC32 "SRT_DINT"	9-4
9.3	Опрос состояния прерывания с задержкой с помощью SFC34 "QRY_DINT"	9-5
9.4	Отмена прерывания с задержкой с помощью SFC33 "CAN_DINT"	9-6
10	SFC для обработки синхронных ошибок	10-1

10.1	Маскирование синхронных ошибок	10-2
10.2	Маскирование синхронных ошибок с помощью SFC36 "MSK_FLT"	10-12
10.3	Демаскирование синхронных ошибок с помощью SFC37 "DMSK_FLT"	10-13
10.4	Чтение регистра ошибок с помощью SFC38 "READ_ERR"	10-14
11	SFC для обработки прерываний и асинхронных ошибок	11-1
11.1	Задержка и блокировка прерываний и асинхронных ошибок	11-2
11.2	Блокировка обработки новых прерываний и асинхронных ошибок с помощью SFC39 "DIS_IRT"	11-4
11.3	Разблокировка обработки новых прерываний и асинхронных ошибок с помощью SFC40 "EN_IRT"	11-6
11.4	Задержка обработки прерываний более высокого приоритета и асинхронных ошибок с помощью SFC41 "DIS_AIRT"	11-7
11.5	Разблокировка обработки прерываний более высокого приоритета и асинхронных ошибок с помощью SFC42 "EN_AIRT"	11-8
12	SFC для диагностики	12-1
12.1	Системная диагностика	12-2
12.2	Чтение стартовой информации OB с помощью SFC6 "RD_SINFO"	12-3
12.3	Чтение списка состояний системы или подсписка с помощью SFC51 "RDSYSST"	12-6
12.4	Запись диагностического события, определенного пользователем, в диагностический буфер с помощью SFC52 "WR_USMSG"	12-10
13	SFC и SFB для обновления образа процесса и обработки битовых массивов	13-1
13.1	Обновление таблицы входов образа процесса с помощью SFC26 "UPDAT_PI"	13-2
13.2	Обновление таблицы выходов образа процесса с помощью SFC27 "UPDAT_PO"	13-3
13.3	Установка области выходов с помощью SFC79 "SET"	13-4
13.4	Сброс области выходов с помощью SFC80 "RSET"	13-5
13.5	Реализация генератора последовательностей с помощью SFB32 "DRUM"	13-6
14	Системные функции для адресации модулей	14-1
14.1	Определение логического базового адреса модуля с помощью SFC5 "GADR_LGC"	14-2
14.2	Определение слота модуля, соответствующего логическому адресу, с помощью SFC49 "LGC_GADR"	14-4
14.3	Определение всех логических адресов модуля с помощью SFC50 "RD_LGADR"	14-6
15	SFC для децентрализованной периферии	15-1
15.1	Запуск аппаратного прерывания в master-устройстве DP с помощью SFC7 "DP_PRAL"	15-2
15.2	Синхронизация групп slave-устройств DP с помощью SFC11 "DPSYC_FR"	15-4
15.3	Чтение диагностических данных slave-устройства DP (диагностика slave-устройства) с помощью SFC 13 "DPNRM_DG"	15-9
15.4	Чтение непротиворечивых данных стандартного slave-устройства DP с помощью SFC14 "DPRD_DAT"	15-12

15.5	Запись непротиворечивых данных в стандартное slave-устройство DP с помощью SFC15 "DPWR_DAT"	15-14
16	SFC для связи с помощью глобальных данных	16-1
16.1	Передача GD-пакета с помощью SFC60 "GD_SND"	16-2
16.2	Извлечение принятого GD-пакета с помощью SFC61 "GD_RCV"	16-4
17	Коммуникационные SFB для спроектированных соединений	17-1
17.1	Классификация коммуникационных SFB для спроектированных соединений S7	17-2
17.2	Классификация параметров коммуникационных SFB для спроектированных соединений S7	17-4
17.3	Некоординированная передача данных с помощью SFB8 "USEND"	17-8
17.4	Некоординированный прием данных с помощью SFB9 "URCV"	17-10
17.5	Передача сегментированных данных с помощью SFB12 "BSEND"	17-12
17.6	Прием сегментированных данных с помощью SFB13 "BRCV"	17-14
17.7	Чтение данных из удаленного CPU с помощью SFB14 "GET"	17-16
17.8	Запись данных в удаленный CPU с помощью SFB15 "PUT"	17-18
17.9	Передача данных на принтер с помощью SFB16 "PRINT"	17-20
17.10	Инициализация теплого или холодного рестарта в удаленном устройстве с помощью SFB19 "START"	17-26
17.11	Переключение удаленного устройства в состояние STOP с помощью SFB20 "STOP"	17-28
17.12	Инициализация горячего рестарта в удаленном устройстве с помощью SFB21 "RESUME"	17-30
17.13	Запрос состояния удаленного партнера с помощью SFB22 "STATUS"	17-33
17.14	Прием состояния удаленного устройства с помощью SFB23 "USTATUS"	17-35
17.15	Запрос состояния соединения, относящегося к экземпляру коммуникационного SFB, с помощью SFC62 "CONTROL"	17-37
17.16	Подпрограмма запуска коммуникационных SFB для спроектированных соединений S7	17-38
17.17	Как SFB для спроектированных соединений S7 реагируют на проблемы	17-40
18	Коммуникационные SFC для неспроектированных соединений S7	18-1
18.1	Отличия от коммуникационных SFB для спроектированных соединений S7	18-2
18.2	Обзор	18-4
18.3	Общие параметры коммуникационных SFC	18-8
18.4	Непротиворечивость данных в случае SFC типа GET и PUT	18-9
18.5	Передача данных коммуникационному партнеру вне локальной станции S7 с помощью SFC65 "X_SEND"	18-12
18.6	Прием данных от коммуникационного партнера вне локальной станции S7 с помощью SFC66 "X_RCV"	18-14
18.7	Чтение данных из коммуникационного партнера вне локальной станции S7 с помощью SFC67 "X_GET"	18-18
18.8	Запись данных в коммуникационном партнере вне локальной станции S7 с помощью SFC68 "X_PUT"	18-20
18.9	Прерывание существующего соединения с коммуникационным партнером вне локальной станции S7 с помощью SFC69 "X_ABORT"	18-22
18.10	Чтение данных из коммуникационного партнера в пределах локальной станции S7 с помощью SFC72 "I_GET"	18-23

18.11	Запись данных в коммуникационном партнере в пределах локальной станции S7 с помощью SFC73 "I_PUT"	18-25
18.12	Прерывание существующего соединения с коммуникационным партнером в пределах локальной станции S7 с помощью SFC74 "I_ABORT"	18-27
18.13	Информация об ошибках коммуникационных SFC для не спроектированных соединений S7	18-29

19	Создание сообщений, связанных с блоками	19-1
19.1	Введение в создание сообщений, связанных с блоками, с помощью SFB	19-2
19.2	Создание с помощью SFB36 "NOTIFY" сообщений, связанных с блоками, без квитирования	19-5
19.3	Создание с помощью SFB33 "ALARM" сообщений, связанных с блоками, с квитированием	19-8
19.4	Создание сообщений, связанных с блоками, с сопутствующими значениями для восьми сигналов с помощью SFB35 "ALARM_8P"	19-11
19.5	Создание сообщений, связанных с блоками, без сопутствующих значений для восьми сигналов с помощью SFB34 "ALARM_8"	19-14
19.6	Передача архивных данных с помощью SFB37 "AR_SEND"	19-16
19.7	Блокировка сообщений, связанных с блоком, сообщений, связанных с символом, и сообщений о групповом состоянии с помощью SFC10 "DIS_MSG"	19-18
19.8	Разблокировка сообщений, связанных с блоком, сообщений, связанных с символом, и сообщений о групповом состоянии с помощью SFC9 "EN_MSG"	19-20
19.9	Поведение SFB для создания сообщений, связанных с блоками, при пуске	19-21
19.10	Как SFB для создания сообщений, связанных с блоками, реагируют на неисправности	19-23
19.11	Введение в создание сообщений, связанных с блоками, с помощью SFC	19-23
19.12	Создание квитлируемых сообщений, связанных с блоками, с помощью SFC 17 "ALARM_SQ" и всегда квитлируемых сообщений, связанных с блоками, с помощью SFC18 "ALARM_S"	19-26
19.13	Определение состояния квитирования последнего сообщения для наступающего события ALARM_SQ с помощью SFC19 "ALARM_SC"	19-29
20	Таймеры и счетчики IEC	20-1
20.1	Генерирование импульса с помощью SFB3 "TP"	20-2
20.2	Генерирование задержки включения с помощью SFB4 "TON"	20-4
20.3	Генерирование задержки выключения с помощью SFB5 "TOF"	20-6
20.4	Прямой счет с помощью SFB0 "CTU"	20-8
20.5	Обратный счет с помощью SFB1 "CTD"	20-9
20.6	Прямой и обратный счет с помощью SFB2 "CTUD"	20-10
21	Функции IEC	21-1
21.1	Обзор	21-2
21.2	Технические данные функций IEC	21-3
21.3	Дата и время как составные типы данных	21-4
21.4	Функции даты и времени суток: FC3, FC6, FC7, FC8, FC33, FC40, FC1, FC35, FC34	21-5
21.5	Сравнение переменных типа DATE_AND_TIME: FC9, FC12, FC14, FC18, FC23, FC28	21-9
21.6	Сравнение переменных типа STRING: FC10, FC13, FC15, FC19, FC24, FC29	21-11
21.7	Редактирование переменных типа STRING: FC21, FC20, FC32, FC26, FC2, FC17, FC4, FC31, FC11	21-14
21.8	Преобразование форматов типов данных: FC16, FC5, FC30, FC38, FC37, FC39	21-18
21.9	Редактирование числовых значений: FC22, FC25, FC27	21-21
21.10	Двоичный выбор: FC36	21-23

22	SFB для встроенного регулирования	22-1
22.1	Непрерывное регулирование с помощью SFB41 "CONT_C"	22-3
22.2	Ступенчатое регулирование с помощью SFB42 "CONT_S"	22-10
22.3	Формирование импульсов с помощью SFB43 "PULSEGEN"	22-16
22.4	Пример блока PULSEGEN	22-27
23	SFC для H CPU	23-1
23.1	Управление функционированием в H-системах с помощью SFC90 "H_CTRL"	23-1
24	Встроенные функции (для CPU с встроенными входами/выходами)	24-1
24.1	SFB29 (HS_COUNT)	24-2
24.2	SFB30 (FREQ_MES)	24-3
24.3	SFB38 (HSC_A_B)	24-4
24.4	SFB39 (POS)	24-5
25	Гибкая технология	25-1
25.1	SFC63 (AB_CALL)	25-2
26	Диагностические данные	26-1
26.1	Обзор структуры диагностических данных	26-2
26.2	Диагностические данные	26-3
26.3	Структура диагностических данных, относящихся к каналу	26-6
27	Списки состояний системы (SSL)	27-1
27.1	Обзор списков состояний системы (SSL)	27-2
27.2	Структура подписка SSL	27-3
27.3	SSL-ID	27-4
27.4	Возможные подписки состояний системы	27-5
27.5	SSL-ID W#16#xy11 – Идентификация модуля	27-6
27.6	SSL-ID W#16#xy12 – Характеристики CPU	27-7
27.7	SSL-ID W#16#xy13 – Области памяти	27-9
27.8	SSL-ID W#16#xy14 – Системные области	27-10
27.9	SSL-ID W#16#xy15 – Типы блоков	27-11
27.10	SSL-ID W#16#xy19 – Состояние светодиодов модуля	27-12
27.11	SSL-ID W#16#xy22 – Состояние прерывания	27-13
27.12	SSL-ID W#16#xy32 – Данные о состоянии связи	27-15
27.13	Запись данных фрагмента подписка с SSL-ID W#16#0132 и индексом W#16#0005	27-16
27.14	Запись данных фрагмента подписка с SSL-ID W#16#0132 и индексом W#16#0008	27-17
27.15	Запись данных фрагмента подписка с SSL-ID W#16#0232 и индексом W#16#0004	27-18
27.16	SSL-ID W#16#xy71 – Групповая информация H CPU	27-19
27.17	SSL-ID W#16#xy74 – Состояние светодиодов модуля	27-21
27.18	SSL-ID W#16#xy91 – Информация о состоянии модуля	27-23

27.19	SSL-ID W#16#xy92 – Информация о состоянии стойки/станции	27-27
27.20	SSL-ID W#16#xyA0 – Диагностический буфер	27-30
27.21	SSL-ID W#16#00B1 – Диагностическая информация модуля	27-31
27.22	SSL-ID W#16#00B2 – Диагностическая запись данных 1 с географическим адресом	27-32
27.23	SSL-ID W#16#00B3 – Диагностические данные модуля с логическим базовым адресом	27-33
27.24	SSL-ID W#16#00B4 – Диагностические данные Slave-устройства DP	27-34

28 События 28-1

28.1	События и идентификаторы (ID) событий	28-2
28.2	Класс событий 1 – События, связанные со стандартными OB	28-4
28.3	Класс событий 2 – Синхронные ошибки	28-5
28.4	Класс событий 3 – Асинхронные ошибки	28-6
28.5	Класс событий 4 – События, связанные с режимом STOP, и другие изменения режима работы	28-8
28.6	Класс событий 5 – События этапа выполнения	28-11
28.7	Класс событий 6 – Коммуникационные события	28-12
28.8	Класс событий 7 - События H/F	28-13
28.9	Класс событий 8 – Диагностические события для модулей	28-14
28.10	Класс событий 9 – Стандартные события пользователя	28-16
28.11	Классы событий A и B – Свободные пользовательские события	28-18
28.12	Зарезервированные классы событий	28-19

29 Список SFC, SFB и FC 29-1

29.1	Список SFC, упорядоченный по номерам	29-2
29.2	Список SFC, упорядоченный по алфавиту	29-4
29.3	Список SFB, упорядоченный по номерам	29-6
29.4	Список SFB, упорядоченный по алфавиту	29-7
29.5	Функции IEC, упорядоченные по номерам	29-8

Глоссарий Глоссарий-1

Литература Литература -1

Предметный указатель Индекс-1

1 Организационные блоки

1.1 Обзор организационных блоков (ОВ)

Что такое организационные блоки?

Организационные блоки образуют интерфейс между операционной системой CPU и программой пользователя. ОВ используются для исполнения определенных разделов программы:

- при запуске CPU
- при циклическом или зависящем от времени исполнении программы
- при возникновении ошибок
- при возникновении аппаратных прерываний.

Организационные блоки исполняются в соответствии с присвоенными им приоритетами.

Какие ОВ имеются в распоряжении?

Не все CPU могут обрабатывать все ОВ, доступные в STEP 7. Обратитесь к техническому описанию своего CPU, чтобы определить, какие ОВ могут использоваться вашим CPU.

Где найти дополнительную информацию?

Для получения дополнительной информации обращайтесь к оперативной помощи в режиме online или к следующим руководствам:

- **/70/**: Это руководство содержит справочные данные о функциональных возможностях различных CPU S7–300. Там же описаны возможные стартовые события для каждого ОВ.
- **/101/**: Это руководство содержит справочные данные о функциональных возможностях различных CPU S7–400. Там же описаны возможные стартовые события для каждого ОВ.

Следующая таблица содержит стартовое событие и установленный по умолчанию класс приоритета для каждого ОВ.

ОВ	Стартовое событие	Класс приоритета по умолчанию	Объяснение
ОВ1	Конец запуска или конец ОВ1	1	Свободный цикл
ОВ10	Прерывание по времени 0	2	Время по умолчанию не определено
ОВ11	Прерывание по времени 1	2	
ОВ12	Прерывание по времени 2	2	
ОВ13	Прерывание по времени 3	2	
ОВ14	Прерывание по времени 4	2	
ОВ15	Прерывание по времени 5	2	
ОВ16	Прерывание по времени 6	2	
ОВ17	Прерывание по времени 7	2	
ОВ20	Прерывание с задержкой 0	3	Время по умолчанию не определено
ОВ21	Прерывание с задержкой 1	4	
ОВ22	Прерывание с задержкой 2	5	
ОВ23	Прерывание с задержкой 3	6	
ОВ30	Циклическое прерывание 0 (интервал по умолчанию: 5 с)	7	Циклические прерывания
ОВ31	Циклическое прерывание 1 (интервал по умолчанию: 2 с)	8	
ОВ32	Циклическое прерывание 2 (интервал по умолчанию: 1 с)	9	
ОВ33	Циклическое прерывание 3 (интервал по умолчанию: 500 мс)	10	
ОВ34	Циклическое прерывание 4 (интервал по умолчанию: 200 мс)	11	
ОВ35	Циклическое прерывание 5 (интервал по умолчанию: 100 мс)	12	
ОВ36	Циклическое прерывание 6 (интервал по умолчанию: 50 мс)	13	
ОВ37	Циклическое прерывание 7 (интервал по умолчанию: 20 мс)	14	
ОВ38	Циклическое прерывание 8 (интервал по умолчанию: 10 мс)	15	
ОВ40	Аппаратное прерывание 0	16	Аппаратные прерывания
ОВ41	Аппаратное прерывание 1	17	
ОВ42	Аппаратное прерывание 2	18	
ОВ43	Аппаратное прерывание 3	19	
ОВ44	Аппаратное прерывание 4	20	
ОВ45	Аппаратное прерывание 5	21	
ОВ46	Аппаратное прерывание 6	22	
ОВ47	Аппаратное прерывание 7	23	
ОВ60	Вызов SFC35 "MP_ALM"	25	Прерывание мультипроцессорного режима
ОВ70	Ошибка резервирования ввода/вывода	25	Прерывания по ошибке резервирования
ОВ72	Ошибка резервирования CPU	28	

ОВ	Стартовое событие	Класс приоритета по умолчанию	Объяснение
OB80	Ошибка времени	26, 28 ¹⁾	Прерывания в результате асинхронной ошибки
OB81	Неисправность источника питания	26, 28 ¹⁾	
OB82	Диагностическое прерывание	26, 28 ¹⁾	
OB83	Прерывание установки/снятия модуля	26, 28 ¹⁾	
OB84	Аппаратная ошибка CPU	26, 28 ¹⁾	
OB85	Ошибка исполнения программы	26, 28 ¹⁾	
OB86	Выход из строя стойки расширения, master-системы DP или станции для децентрализованной периферии	26, 28 ¹⁾	
OB87	Коммуникационная ошибка	26, 28 ¹⁾	
OB90	Теплый или холодный рестарт, или удаление блока, исполняемого в OB90, или загрузка OB90 в CPU, или завершение OB90	29 ²⁾	Фоновый цикл
OB100	Теплый рестарт	27 ¹⁾	Запуск
OB101	Горячий рестарт	27 ¹⁾	
OB102	Холодный рестарт	27 ¹⁾	
OB121	Ошибка программирования	Приоритет ОВ, вызвавшего ошибку	Прерывания в результате синхронной ошибки
OB122	Ошибка доступа к периферии	Приоритет ОВ, вызвавшего ошибку	

¹⁾ Классы приоритетов 27 и 28 действительны в модели классов приоритетов запуска.

²⁾ Класс приоритета 29 соответствует приоритету 0.29. Фоновый цикл, таким образом, имеет более низкий приоритет, чем свободный цикл.

1.2 Организационный блок циклического выполнения программы (OB1)

Описание

Операционная система CPU S7 исполняет OB1 непрерывно. Когда OB1 исполнен, операционная система начинает его обработку вновь. Циклическая обработка OB начинается по окончании стадии запуска. Вы можете вызывать в OB1 функциональные блоки (FB, SFB) или функции (FC, SFC).

Принцип действия OB1

OB1 имеет самый низкий приоритет среди всех OB, время выполнения которых контролируется, иными словами, все остальные OB, кроме OB90, могут прерывать выполнение OB1. Операционная система вызывает OB1 при следующих событиях:

- Завершение запуска.
- Конец обработки OB 1 (предыдущего цикла).

Когда OB1 исполнен, операционная система отправляет глобальные данные. Перед повторным запуском OB1 операционная система записывает таблицу выходов образа процесса в модули вывода, обновляет таблицу входов образа процесса и получает глобальные данные для CPU.

S7 осуществляет контроль максимальной длительности цикла сканирования, чем гарантируется максимальное время реакции. Значение максимальной длительности цикла сканирования установлено по умолчанию на 150 мс. Вы можете установить новое значение или перезапустить контроль времени в любой точке вашей программы при помощи SFC43 "RE_TRIGR". Если время выполнения программы превышает максимальное время цикла OB 1, то операционная система вызывает OB 80 (OB ошибок времени); если OB 80 не запрограммирован, то CPU переходит в состояние STOP.

Кроме контроля максимального времени цикла может быть гарантировано также минимальное время цикла сканирования. Операционная система задержит следующий запуск нового цикла (запись таблицы выходов образа процесса в модули вывода), пока не будет обеспечено минимальное время сканирования.

Диапазоны параметров задания максимального и минимального времени сканирования приведены в руководствах [/70/](#) и [/101/](#). Вы можете изменить настройку параметров с помощью STEP 7.

Локальные данные OB1

Следующая таблица описывает временные переменные (TEMP) для OB1.
Имена переменных OB1 заданы по умолчанию.

Переменная	Тип	Описание
OB1_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#11: OB1 активен
OB1_SCAN_1	BYTE	В#16#01: завершение теплого рестарта В#16#02: завершение горячего рестарта В#16#03: завершение основного цикла В#16#04: завершение холодного рестарта
OB1_PRIORITY	BYTE	Класс приоритета 1
OB1_OB_NUMBR	BYTE	Номер OB (01)
OB1_RESERVED_1	BYTE	Резерв
OB1_RESERVED_2	BYTE	Резерв
OB1_PREV_CYCLE	INT	Длительность предыдущего цикла (мс)
OB1_MIN_CYCLE	INT	Минимальная длительность цикла (мс) со времени последнего запуска
OB1_MAX_CYCLE	INT	Максимальная длительность цикла (мс) со времени последнего запуска
OB1_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван OB

1.3 Организационные блоки прерываний по времени (OB10 – OB17)

Описание

STEP 7 предоставляет в распоряжение до восьми прерываний по времени (OB 10 - OB 17), которые могут запускаться однократно или периодически. Вы можете так параметризовать Ваше CPU при помощи SFC или STEP 7, что эти OB будут обрабатываться со следующими интервалами:

- Однократно
- Ежеминутно
- Ежечасно
- Ежедневно
- Еженедельно
- Ежемесячно
- В конце каждого месяца

Принцип действия OB прерываний по времени

Чтобы запустить прерывание по времени, его необходимо вначале установить, а потом активизировать. Существует три следующих способа запуска:

- Автоматический запуск прерывания по времени. Он происходит как только вы установили, а затем активизировали прерывание по времени с помощью STEP 7. Следующая таблица показывает основные возможности активизации прерывания по времени с помощью STEP 7.
- Вы устанавливаете прерывания по времени при помощи STEP 7, а затем активизируете их вызовом SFC 30 "ACT_TINT" в своей программе.
- Вы устанавливаете прерывание по времени вызовом SFC 28 "SET_TINT", а затем активизируете его вызовом SFC 30 "ACT_TINT".

Интервал	Описание
Не активизируется	Прерывание по времени не исполняется, даже когда оно загружено в CPU. Оно может быть активизировано вызовом SFC30.
Активизируется только один раз	OB прерывания по времени автоматически отменяется после того, как он исполняется однократно в указанное время. Ваша программа может с помощью SFC 28 заново установить прерывание по времени и вновь его активизировать через SFC 30.
Активизируется периодически	Когда происходит прерывание по времени, CPU рассчитывает момент его следующего старта исходя из текущего времени и периода.

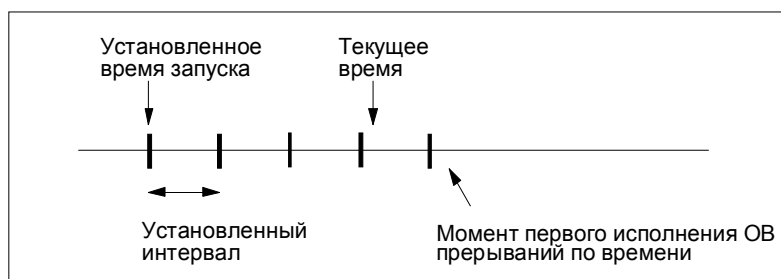
Поведение прерываний по времени при переводе часов вперед или назад описано в [/234/](#).

Замечание

Если вы конфигурируете прерывание по времени на однократную обработку ОВ, то заданные значения даты и времени не могут находиться в прошлом (по отношению к часам реального времени CPU).

Если вы конфигурируете прерывание по времени так, что соответствующий ОВ будет обрабатываться периодически, а дата и время лежат в прошлом, то прерывание по времени будет обработано в следующий надлежащий момент времени. Это иллюстрируется на следующем рисунке.

Прерывания по времени можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42. Более подробная информация содержится в главе 11.



Условия, влияющие на ОВ прерываний по времени

Т.к. прерывание по времени наступает через определенные интервалы времени, то определенные условия могут влиять на работу соответствующего ОВ во время исполнения вашей программы. Следующая таблица представляет некоторые из таких условий и описывает их воздействие на обработку ОВ прерываний по времени.

Условие	Результат
Ваша программа вызывает SFC29 (CAN_TINT) и отменяет прерывание по времени.	Операционная система стирает стартовое событие (дату и время) прерывания по времени. Вам необходимо вновь установить стартовое событие и активизировать его прежде, чем ОВ прерываний по времени снова можно будет вызвать.
Ваша программа пыталась активизировать ОВ прерывания по времени, который ко времени активизации не был загружен в CPU.	Операционная система вызывает ОВ 85. Если ОВ 85 не был запрограммирован (загружен в CPU), то CPU переводится в состояние STOP.
При синхронизации или коррекции системных часов CPU вы переставили время вперед и пропустили стартовое событие, т.е. дату или время для ОВ прерывания по времени.	Операционная система вызывает ОВ 80 и кодирует номер ОВ прерывания по времени и информацию стартового события в ОВ80. Операционная система обрабатывает затем ОВ прерывания по времени один раз независимо от того, сколько раз этот ОВ должен был бы обрабатываться. Информация о стартовом событии ОВ 80 показывает дату и время, когда ОВ прерывания по времени в первый раз был пропущен.
При синхронизации или коррекции системных часов CPU вы перевели время назад так, что стартовое событие, т.е. дата или время для ОВ прерывания по времени повторяется.	Если ОВ прерывания по времени был активизирован до того, как часы были переведены назад, то он не будет вновь вызываться.
CPU выполняет теплый или холодный рестарт.	Каждый ОВ прерывания по времени, который был сконфигурирован через SFC, вновь возвращается к конфигурации, заданной в STEP 7.
ОВ прерывания по времени еще обрабатывается, когда происходит стартовое событие для следующего интервала.	Операционная система вызывает ОВ 80. Если ОВ 80 не был запрограммирован, то CPU переходит в состояние STOP. Если ОВ80 загружен, то сначала исполняются ОВ80 и ОВ прерывания по времени, а затем исполняется второе затребованное прерывание по времени.

Локальные данные ОБ прерываний по времени

Следующая таблица описывает временные переменные (TEMP) одного из ОБ прерываний по времени. В качестве имен переменных выбраны имена по умолчанию ОБ 10.

Переменная	Тип	Описание
OB10_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#11 = прерывание активно
OB10_STRT_INFO	BYTE	В#16#11: запрос на запуск для ОБ10 (В#16#12: запрос на запуск для ОБ11) : : (В#16#18: запрос на запуск для ОБ17)
OB10_PRIORITY	BYTE	Назначенный класс приоритета; по умолчанию 2
OB10_OB_NUMBR	BYTE	Номер ОБ (10 – 17)
OB10_RESERVED_1	BYTE	Резерв
OB10_RESERVED_2	BYTE	Резерв
OB10_PERIOD_EXE	WORD	ОБ обрабатывается с заданными интервалами: W#16#0000: однократно W#16#0201: ежеминутно W#16#0401: ежечасно W#16#1001: ежедневно W#16#1201: еженедельно W#16#1401: ежемесячно W#16#1801: ежегодно
OB10_RESERVED_3	INT	Резерв
OB10_RESERVED_4	INT	Резерв
OB10_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван ОБ.

1.4 Организационные блоки прерываний с задержкой (OB20 – OB23)

Описание

S7 предоставляет в распоряжение до четырех OB (OB 20 – OB 23), которые исполняются после заданной задержки. Каждый OB прерывания с задержкой запускается посредством вызова SFC32 (SRT_DINT). Время задержки является входным параметром SFC.

Когда ваша программа вызывает функцию SFC32 (SRT_DINT), то ей передается номер OB, время задержки и индивидуальный код пользователя. Через заданное время задержки стартует соответствующий OB. Имеется возможность отменить обработку прерывания с задержкой, которое еще не стартовало.

Принцип действия OB прерываний с задержкой

По истечении времени задержки (его значение в миллисекундах передается блоку SFC32 вместе с номером OB) операционная система запускает соответствующий.

Чтобы использовать OB прерываний с задержкой, вы должны выполнить следующие задачи:

- Вы должны вызвать SFC32 (SRT_DINT).
- Вы должны загрузить OB прерываний с задержкой в CPU как часть своей программы.

OB прерываний с задержкой исполняются только в том случае, когда CPU находится в режиме RUN. Теплый или холодный рестарт стирает любые стартовые события для OB прерываний с задержкой. Если прерывание с задержкой еще не запущено, то для отмены его исполнения можно использовать SFC33 (CAN_DINT).

Время задержки измеряется с точностью 1 мс. Время задержки может быть повторно запущено непосредственно после его окончания. С помощью SFC34 (QRY_DINT) вы можете опросить состояние прерывания с задержкой.

Операционная система вызывает OB асинхронных ошибок, если происходит одно из следующих событий:

- Если операционная система пытается запустить OB, который не загружен, а его номер вы задали при вызове SFC32 "SRT_DINT".
- Если следующее стартовое событие для запуска прерывания с задержкой наступает прежде, чем закончилась обработка текущего OB прерывания с задержкой.

Прерывания с задержкой можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42. Более подробная информация содержится в главе 12.

Локальные данные ОБ прерываний с задержкой

Следующая таблица описывает временные переменные (TEMP) ОБ прерывания с задержкой. В качестве имен переменных взяты имена по умолчанию ОБ20.

Переменная	Тип	Описание
OB20_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#11: прерывание активно
OB20_STRT_INF	BYTE	В#16#21: требование запуска для ОБ20 (В#16#22: требование запуска для ОБ21) (В#16#23: требование запуска для ОБ22) (В#16#24: требование запуска для ОБ23)
OB20_PRIORITY	BYTE	Назначенный класс приоритета: значения по умолчанию от 3 (ОБ20) до 6 (ОБ23)
OB20_OB_NUMBR	BYTE	Номер ОБ (20 – 23)
OB20_RESERVED_1	BYTE	Резерв
OB20_RESERVED_2	BYTE	Резерв
OB20_SIGN	WORD	Идентификатор пользователя: входной параметр SIGN из вызова SFC32 (SRT_DINT)
OB20_DTIME	TIME	Истекшая часть времени задержки в мс
OB20_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван ОБ

1.5 Организационные блоки циклических прерываний (ОВ30 – ОВ38)

Описание

S7 представляет в распоряжение до девяти ОВ циклических прерываний (ОВ 30 – ОВ38), которые прерывают вашу программу через фиксированные интервалы времени. Следующая таблица показывает установленные по умолчанию интервалы времени и классы приоритета для ОВ циклических прерываний.

Номер ОВ	Интервал по умолчанию	Класс приоритета по умолчанию
ОВ30	5 с	7
ОВ31	2 с	8
ОВ32	1 с	9
ОВ33	500 мс	10
ОВ34	200 мс	11
ОВ35	100 мс	12
ОВ36	50 мс	13
ОВ37	20 мс	14
ОВ38	10 мс	15

Принцип действия ОВ циклических прерываний

Эквидистантные моменты запуска ОВ циклических прерываний определяются интервалом и фазовым сдвигом. Как связаны друг с другом момент запуска, периодичность и фазовый сдвиг, описано в [/234/](#).

Замечание

Вы должны позаботиться о том, чтобы время работы ОВ циклических прерываний было значительно меньше интервала времени, через который он вызывается. Если ОВ циклических прерываний еще не закончен, а в следствие истечения интервала времени должен обрабатываться вновь, запускается ОВ ошибок времени (ОВ 80). Циклическое прерывание, вызвавшее ошибку, будет отработано позднее.

Циклическое прерывание можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42. **За дополнительной информацией обратитесь к главе 13.**

Диапазон значений для периодичности, классов приоритета и фазового сдвига приводятся в технических данных CPU. Вы можете изменить настройки этих параметров с помощью STEP 7.

Локальные данные ОВ циклических прерываний

Следующая таблица временные (TEMP) переменные одного из ОВ циклических прерываний. В качестве имен переменных по умолчанию взяты имена OB35.

Переменная	Тип	Описание
OB35_EV_CLASS	BYTE	Класс события и идентификаторы: B#16#11: прерывание активно
OB35_STRT_INF	BYTE	(B#16#31 : требование запуска для OB30) : B#16#36 : требование запуска для OB35 : (B#16#39 : требование запуска для OB38)
OB35_PRIORITY	BYTE	Назначенный класс приоритета: по умолчанию от 7 (OB30) до 15 (OB38)
OB35_OB_NUMBR	BYTE	Номер ОВ (30 – 38)
OB35_RESERVED_1	BYTE	Резерв
OB35_RESERVED_2	BYTE	Резерв
OB35_PHASE_OFFSET	WORD	Фазовый сдвиг [мс]
OB35_RESERVED_3	INT	Резерв
OB35_EXC_FREQ	INT	Интервал в миллисекундах
OB35_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван ОВ

1.6 Организационные блоки аппаратных прерываний (OB40 – OB47)

Описание

S7 представляет в распоряжение до восьми независимых друг от друга аппаратных прерываний со своими собственными OB.

Назначая с помощью STEP 7 параметры, вы для каждого сигнального модуля, который будет запускать аппаратные прерывания, указываете следующее:

- Какие каналы при каких условиях запускают аппаратное прерывание.
- Какой OB аппаратных прерываний ставится в соответствие отдельным группам каналов (по умолчанию все аппаратные прерывания обрабатываются OB40).

Для CP и FM эти параметры назначаются с помощью их собственного программного обеспечения.

Классы приоритета для отдельных OB аппаратных прерываний устанавливаются с помощью STEP 7.

Принцип действия OB аппаратных прерываний

После запуска модулем аппаратного прерывания операционная система идентифицирует слот и соответствующий OB аппаратных прерываний. Если этот OB имеет более высокий приоритет, чем активный в данный момент класс приоритета, то он запускается. Соответствующее каналу квитирование выполняется после завершения этого OB аппаратных прерываний.

Если в промежутке времени между идентификацией и квитированием аппаратного прерывания на том же самом модуле возникает еще одно событие, которое вызывает аппаратное прерывание, то:

- Если событие наступает на том же самом канале, который до этого вызвал аппаратное прерывание, то новое прерывание теряется. Это иллюстрируется на следующем рисунке на примере канала модуля цифрового ввода. Запускающим событием является нарастающий фронт. OB аппаратных прерываний является OB40.



- Если событие происходит на другом канале того же самого модуля, то аппаратное прерывание в данный момент не может быть запущено. Однако это прерывание не теряется, а будет запущено после квитирования текущего активного прерывания (только для S7-400). В случае S7-300 аппаратное прерывание теряется, если вызывающее его событие закончилось раньше квитирования.

Если аппаратное прерывание запускается, а его OB в данный момент активен из-за аппаратного прерывания из другого модуля, то новый запрос регистрируется, а OB обрабатывается когда он будет свободен (только на S7-400). В случае S7-300 аппаратное прерывание теряется, если событие, вызывающее прерывание, не сохранилось после квитирования.

Аппаратные прерывания можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42. **За дополнительной информацией обращайтесь к главе 12.**

Вы можете назначать параметры аппаратным прерываниям модуля не только при помощи STEP 7, но и при помощи SFC 55 – 57. **За дополнительной информацией обращайтесь к главе 12.**

Локальные данные OB аппаратных прерываний

Следующая таблица описывает временные (TEMP) переменные одного из OB аппаратных прерывания. В качестве имен переменных выбраны имена по умолчанию OB40.

Переменная	Тип	Описание
OB40_EV_CLASS	BYTE	Класс события и идентификаторы: B#16#11: прерывание активно
OB40_STRT_INF	BYTE	B#16#41: прерывание через канал прерывания 1 B#16#42: прерывание через канал прерывания 2 (только для S7-400) B#16#43: прерывание через канал прерывания 3 (только для S7-400) B#16#44: прерывание через канал прерывания 4 (только для S7-400)
OB40_PRIORITY	BYTE	Назначенный класс приоритета: по умолчанию от 16 (OB40) до 23 (OB47)
OB40_OB_NUMBR	BYTE	Номер OB (40 – 47)
OB40_RESERVED_1	BYTE	Резерв
OB40_IO_FLAG	BYTE	Модуль ввода: B#16#54 Модуль вывода: B#16#55
OB40_MDL_ADDR	WORD	Логический базовый адрес модуля, который запустил прерывание
OB40_POINT_ADDR	DWORD	Для цифровых модулей: битовый массив с состояниями входов на модуле (бит 0 соответствует первому входу) Для аналоговых модулей (CP или FM): состояние прерывания модуля (для пользователя значения не имеет)
OB40_DATE_TIME	DATE_AND_TIME	Дата и время, когда OB был вызван

1.7 Организационный блок мультипроцессорных прерываний (OB60)

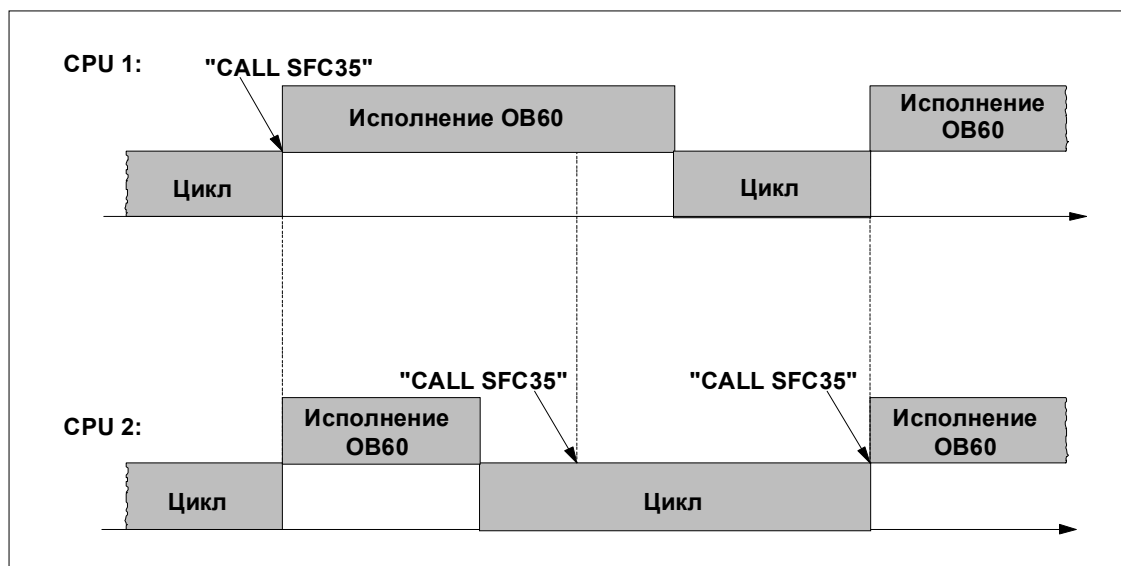
Описание

С помощью мультипроцессорного прерывания вы можете обеспечить синхронную реакцию соответствующих CPU на событие при мультипроцессорной обработке. В отличие от аппаратных прерываний, запускаемых сигнальными модулями, мультипроцессорное прерывание может выдаваться только центральными процессорами.

Принцип действия OB мультипроцессорных прерываний

Мультипроцессорное прерывание инициируется вызовом SFC 35 "MP_ALM". При мультипроцессорной обработке это приводит к синхронизированному старту OB60 на всех установленных CPU данного шинного сегмента, если вы не заблокировали OB60 (с помощью SFC 39 "DIS_IRT") или не задержали его (с помощью SFC 41 "DIS_AIRT"). Если вы не загрузили OB 60 в CPU, то соответствующий CPU немедленно возвращается в последний класс приоритета перед прерыванием и продолжает там обработку программы. В однопроцессорном режиме и при использовании сегментированных стоек OB60 запускается только на том CPU, на котором вы вызвали SFC35 "MP_ALM".

Когда ваша программа вызывает SFC 35 "MP_ALM", вы передаете ей идентификатор задания. Этот идентификатор передается на все. Это позволяет вам реагировать на конкретное событие. Если OB 60 на отдельных CPU запрограммирован по-разному, то время его обработки может оказаться различным. В этом случае CPU возвращаются к прерванному классу приоритета в различные моменты времени. Если следующее мультипроцессорное прерывание выдается одним CPU в то время, как другой еще занят обработкой OB 60 предыдущего мультипроцессорного прерывания, то запуск OB 60 не происходит ни на запрашивающем, ни на каком другом CPU, принадлежащем данному шинному сегменту. Это иллюстрируется на следующем рисунке на примере двух CPU. Вы извещаетесь о результате с помощью значения вызванной функции SFC35.



Локальные данные OB мультипроцессорных прерываний

Следующая таблица описывает временные (TEMP) переменные OB мультипроцессорных прерываний. В качестве имен переменных взяты имена по умолчанию OB60

Переменная	Тип	Описание
OB60_EV_CLASS	BYTE	Класс события и идентификаторы: B#16#11: прерывание активно
OB60_STRT_INF	BYTE	B#16#61: Мультипроцессорное прерывание запущено собственным CPU B#16#62: Мультипроцессорное прерывание запущено другим CPU
OB60_PRIORITY	BYTE	Назначенный класс приоритета: по умолчанию 25
OB60_OB_NUMBR	BYTE	Номер OB: 60
OB60_RESERVED_1	BYTE	Резерв
OB60_RESERVED_2	BYTE	Резерв
OB60_JOB	INT	Идентификатор задания: входная переменная JOB SFC35 "MP_ALM"
OB60_RESERVED_3	INT	Резерв
OB60_RESERVED_4	INT	Резерв
OB60_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван OB.

1.8 ОВ ошибок резервирования входов/выходов (OB70)

Примечание

ОВ ошибок резервирования входов/выходов (OB70) может использоваться только в H CPU.

Описание

Операционная система H CPU вызывает OB70, когда имеет место потеря резервирования в PROFIBUS DP (например, когда происходит отказ шины активного master-устройства DP или ошибка в интерфейсном модуле slave-устройства DP) или когда переключается активное master-устройство DP slave-устройств DP с подключенными входами/выходами.

Если OB70 не запрограммирован и происходит стартовое событие, то CPU не переходит в состояние STOP. Если OB70 загружен и H-система находится в резервном режиме, то OB70 выполняется в обоих CPU. H-система остается в резервном режиме.

Локальные данные ОВ ошибок резервирования входов/выходов

Следующая таблица содержит временные (TEMP) переменные ОВ ошибок резервирования входов/выходов. Выбранные имена переменных являются заданными по умолчанию именами OB70.

Переменная	Тип	Характеристика
OB70_EV_CLASS	BYTE	Класс события и идентификаторы: B#16#78: наступающее событие B#16#79: уходящее событие
OB70_FLT_ID	BYTE	Код ошибки (возможные значения): B#16#A2, B#16#A3
OB70_PRIORITY	BYTE	Назначенный класс приоритета: по умолчанию 25
OB70_OB_NUMBR	BYTE	Номер ОВ (70)
OB70_INFO_1	WORD	Зависит от кода ошибки
OB70_INFO_2	WORD	Зависит от кода ошибки
OB70_INFO_3	WORD	Зависит от кода ошибки
OB70_RESERVED_1	WORD	Резерв
OB80_DATE_TIME	DATE_AND_TIME	Дата и время, когда вызывался ОВ

Следующая таблица показывает, какое событие привело к запуску OB70.

OB70_FLT_ID	Стартовое событие OB70
B#16#A0	Переключился резервный интерфейсный модуль DP вследствие отказа активного slave-устройства DP.
B#16#A1	Переключился резервный интерфейсный модуль DP вследствие отказа активного master-устройства DP.
B#16#A2	Отказ пассивного master-устройства DP.
B#16#A3	Отказ пассивного slave-устройства DP.

Переменные, зависящие от кода ошибки, имеют следующее значение:

- Коды ошибки B#16#A2
 - OB70_INFO_1: Логический базовый адрес неисправного master-устройства DP
 - OB70_INFO_2: Резерв
 - OB70_INFO_3:
 - Биты с 0 по 7: Резерв
 - Биты с 8 по 15: Идентификатор master-системы DP неисправного master-устройства DP
- Коды ошибки B#16#A3
 - OB70_INFO_1: Логический базовый адрес master-устройства DP
 - OB70_INFO_2: Неисправное slave-устройство DP:
 - Биты с 0 по 14: Логический базовый адрес, если используется slave-устройство S7, либо диагностический адрес, если используется стандартное slave-устройство DP.
 - Бит 15: Идентификация входов/выходов
 - OB70_INFO_3: Неисправное slave-устройство DP:
 - Биты с 0 по 7: Номер станции DP
 - Биты с 8 по 15: Идентификатор master-системы DP

1.9 ОВ ошибок резервирования CPU (O B72)

Примечание

ОВ ошибок резервирования CPU (OB72) существует только у H CPU.

Описание

Операционная система H CPU вызывает OB72, когда происходит одно из следующих событий:

- Потеря резервирования CPU
- Переключение на резервное master-устройство
- Ошибка синхронизации
- Ошибка в модуле синхронизации
- Прерывание обновления
- Ошибка сравнения (например, RAM, PIQ)

OB72 выполняется всеми CPU, которые находятся в режиме RUN или STARTUP, после соответствующего стартового события.

Локальные данные ОВ ошибок резервирования CPU

Следующая таблица содержит временные (TEMP) переменные ОВ ошибок резервирования CPU. Заданные по умолчанию имена OB72 были использованы как имена переменных.

Переменная	Тип	Характеристика
OB72_EV_CLASS	BYTE	Класс события и идентификаторы: V#16#73, V#16#75, V#16#79, V#16#78
OB72_FLT_ID	BYTE	Код ошибки (возможные значения: V#16#01, V#16#02, V#16#03, V#16#20, V#16#21, V#16#22, V#16#23, V#16#31, V#16#33, V#16#34, V#16#35, V#16#40, V#16#41, V#16#42, V#16#43, V#16#44, V#16#50, V#16#51, V#16#52, V#16#53, V#16#54, V#16#55, V#16#56, V#16#C1)
OB72_PRIORITY	BYTE	Назначенный класс приоритета: значение по умолчанию 28
OB72_OB_NUMBR	BYTE	Номер ОВ (72)
OB72_INFO_1	WORD	(Не имеет значения для пользователя)
OB72_INFO_2	WORD	(Не имеет значения для пользователя)
OB72_INFO_3	WORD	(Не имеет значения для пользователя)
OB72_RESERVED_1	WORD	Резерв
OB82_DATE_TIME	DATE_AND_TIME	Дата и время, когда вызывался ОВ

Следующая таблица показывает, какое событие привело к запуску OB72.

OB72_FLT_ID	Стартовое событие OB72
B#16#01	Потеря резервирования (1 из 2) вследствие запуска CPU
B#16#02	Потеря резервирования (1 из 2) вследствие перехода в STOP в резерве, запущенном пользователем
B#16#03	Н-система (1 из 2) переключилась в резервный режим
B#16#20	Ошибка при сравнении RAM
B#16#21	Ошибка при сравнении выходного значения образа процесса
B#16#22	Ошибка при сравнении битов памяти, таймеров или счетчиков
B#16#31	Переключение на резервное master-устройство вследствие отказа master-устройства
B#16#33	Переключение на резервное master-устройство вследствие вмешательства оператора
B#16#34	Переключение на резервное master-устройство вследствие проблемы соединения
B#16#35	Переключение на резервное master-устройство, запущенное SFC90 "H_CTRL".
B#16#40	Ошибка синхронизации в программе пользователя из-за истекшего времени ожидания
B#16#41	Ошибка синхронизации в программе пользователя из-за ожидания в различных точках синхронизации
B#16#42	Ошибка синхронизации в операционной системе из-за ожидания в различных точках синхронизации
B#16#43	Ошибка синхронизации в операционной системе из-за истекшего времени ожидания
B#16#50	Нет модуля синхронизации (SYNC)
B#16#51	Изменение в модуле синхронизации без включения питания
B#16#52	Модуль синхронизации удален/вставлен
B#16#53	Изменение в модуле синхронизации без сброса
B#16#54	Модуль синхронизации: номер стойки назначен дважды
B#16#55	Ошибка модуля синхронизации/устранена
B#16#56	В модуле синхронизации установлен запрещенный номер стойки
B#16#C1	Прерывание обновления

1.10 Организационный блок ошибок времени (OB80)

Описание

Операционная система CPU S7-300 вызывает OB80, когда при обработке какого-либо ОБ возникает одна из следующих ошибок: превышение времени цикла, ошибка квитирования при исполнении ОБ, перевод часов вперед, так что пропускается время запуска ОБ. Если, например, стартовое событие для ОБ циклических прерываний возникает до того, как была закончена обработка предыдущего вызова, то операционная система вызывает OB80. Если ОБ 80 не был запрограммирован, то CPU переходит в состояние STOP. ОБ ошибок времени можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42. [За дополнительной информацией обратитесь к главе 12.](#)

Замечание

Если ОБ 80 в одном и том же цикле вызывается дважды из-за превышения времени цикла, то CPU переходит в состояние STOP. Вы можете этому воспрепятствовать вызовом SFC43 "RE_TRIGR" в подходящей точке программы.

Локальные данные ОБ ошибок времени

Следующая таблица описывает временные (TEMP) переменные ОБ ошибок времени. В качестве имен переменных взяты имена по умолчанию OB80.

Переменная	Тип	Описание
OB80_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#35
OB80_FLT_ID	BYTE	Код ошибки (возможные значения В#16#01, В#16#02, В#16#05, В#16#06, В#16#07)
OB80_PRIORITY	BYTE	Класс приоритета: 26 (режим RUN) или 28 (режим STARTUP) На резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты 0 – 2: номер стойки Бит 3: 0=резервный CPU, 1=основной CPU Биты 4 – 7: 1111
OB80_OB_NUMBR	BYTE	Номер ОБ (80)
OB80_RESERVED_1	BYTE	Резерв
OB80_RESERVED_2	BYTE	Резерв
OB80_ERROR_INFO	WORD	Информация об ошибке: зависит от кода ошибки
OB80_ERR_EV_CLASS	BYTE	Класс события, которое вызвало ошибку
OB80_ERR_EV_NUM	BYTE	Номер события, которое вызвало ошибку
OB80_OB_PRIORITY	BYTE	Класс приоритета ОБ, который был активен, когда произошла ошибка
OB80_OB_NUM	BYTE	Номер ОБ, который был активен, когда произошла ошибка
OB80_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван ОБ

Переменные, зависящие от кода ошибки, имеют следующие значения:

• Код ошибки В#16#01		Превышено время цикла.
OB80_ERROR_INFO:		Длительность последнего цикла (мс).
OB80_ERR_EV_CLASS:		Класс события, которое вызвало прерывание.
OB80_ERR_EV_NUM:		Номер события, которое вызвало прерывание.
OB80_OB_PRIORITY:		Класс приоритета, активный на момент прерывания.
• Код ошибки	В#16#02	Затребованный ОБ еще выполняется.
	В#16#07	Переполнение буфера вызовов ОБ для текущего класса приоритета (каждый запрос старта ОБ для класса приоритета заносится в соответствующий буфер вызовов ОБ; по завершении ОБ эта запись удаляется. Если же для некоторого класса приоритета количество вызовов ОБ превысило максимально допустимое число записей соответствующего буфера вызовов, то вызывается ОБ 80 с кодом ошибки В#16#07).
OB80_ERROR_INFO:		Соответствующая временная переменная запрошенного блока, определяемая с помощью OB80_ERR_EV_CLASS и OB80_ERR_EV_NUM.
OB80_ERR_EV_CLASS:		Класс события, вызвавшего прерывание.
OB80_ERR_EV_NUM:		Номер события, вызвавшего прерывание.
OB80_OB_PRIORITY:		Класс приоритета, активный на момент прерывания.
OB80_OB_NUM:		ОБ, активный на момент прерывания.
• Код ошибки	В#16#05	Истекшее прерывание по времени из-за перевода часов вперед.
	В#16#06	Истекшее прерывание по времени после возвращения в RUN после HOLD.
OB80_ERROR_INFO	Бит 0 установлен	Для прерывания по времени №0 стартовая точка лежит в прошлом.
	:	
	:	
	Бит 7 установлен	Для прерывания по времени №7 стартовая точка лежит в прошлом.
	Биты 8 – 15:	не используются
OB80_ERR_EV_CLASS:		не используется
OB80_ERR_EV_NUM:		не используется
OB80_OB_PRIORITY:		не используется
OB80_OB_NUM:		не используется

1.11 Организационный блок неисправностей источника питания (OB81)

Описание

Операционная система CPU S7-300 вызывает OB81, когда происходит событие, вызванное ошибкой или сбоем, связанным с источником питания (только для S7-400) или буферной батареей (при наступающем и при уходящем событии).

В отличие от OB для других асинхронных ошибок CPU в данном случае не переходит в режим STOP, если OB 81 не был запрограммирован.

OB неисправностей источника питания можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42.

За дополнительной информацией обращайтесь к главе 12.

Локальные данные OB неисправностей источника питания

Следующая таблица описывает временные (TEMP) переменные OB неисправностей источника питания. В качестве имен переменных взяты имена по умолчанию OB81.

Переменная	Тип	Описание
OB81_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#38: наступающее событие В#16#39: уходящее событие
OB81_FLT_ID	BYTE	Код ошибки: (возможные значения В#16#21, В#16#22, В#16#23, В#16#25, В#16#26, В#16#27, В#16#31, В#16#32, В#16#33)
OB81_PRIORITY	BYTE	Класс приоритета: 26 (по умолчанию для режима RUN) или 28 (режим STARTUP) На резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты 0 – 2: Номер стойки Бит 3: 0=резервный CPU, 1=основной CPU Биты 4 – 7: 1111
OB81_OB_NUMBR	BYTE	Номер OB (81)
OB81_RESERVED_1	BYTE	Резерв
OB81_RESERVED_2	BYTE	Резерв
OB81_MDL_ADDR	INT	Резерв
OB81_RESERVED_3	BYTE	Имеет значение только для кодов ошибок В#16#31, В#16#32 и В#16#33
OB81_RESERVED_4	BYTE	
OB81_RESERVED_5	BYTE	
OB81_RESERVED_6	BYTE	
OB81_DATE_TIME	DATE_AND_TIME	Дата и время, когда OB был вызван.

Переменные OB81_RESERVED_i, $3 \leq i \leq 6$ указывают на стойки расширения, на которых возникла неисправность или произошло восстановление буферной батареи (код ошибки B#16#31), резервного напряжения (код ошибки B#16#32) или источника питания 24 В (код ошибки B#16#33). Следующая таблица показывает, какой бит какой стойке расширения соответствует в переменных OB81_RESERVED_i, $3 \leq i \leq 6$.

	OB81_RESERVED_6	OB81_RESERVED_5	OB81_RESERVED_4	OB81_RESERVED_3
Бит 0	Резерв	8-я стойка расширения	16-я стойка расширения	Резерв
Бит 1	1-я стойка расширения	9-я стойка расширения	17-я стойка расширения	Резерв
Бит 2	2-я стойка расширения	10-я стойка расширения	18-я стойка расширения	Резерв
Бит 3	3-я стойка расширения	11-я стойка расширения	19-я стойка расширения	Резерв
Бит 4	4-я стойка расширения	12-я стойка расширения	20-я стойка расширения	Резерв
Бит 5	5-я стойка расширения	13-я стойка расширения	21-я стойка расширения	Резерв
Бит 6	6-я стойка расширения	14-я стойка расширения	Резерв	Резерв
Бит 7	7-я стойка расширения	15-я стойка расширения	Резерв	Резерв

Биты в переменной OB81_RESERVED_i имеют следующее значение (для соответствующих стоек расширения):

При наступлении события помечаются стойки расширения (устанавливаются соответствующие биты), у которых вышли из строя хотя бы одна буферная батарея, или резервное питание, или источник питания 24 В. Стойки расширения, у которых до этого хотя бы одна батарея, или резервное питание, или источник питания 24 В уже вышли из строя, более не индицируются.

Когда событие устраняется и резервирование восстанавливается по крайней мере на одной стойке расширения, то об этом сообщается (устанавливаются соответствующие биты).

Переменная OB81_FLT_ID имеет следующее значение:

- B#16#21: Хотя бы одна буферная батарея центральной стойки разрядилась/ проблема устранена (BATTF)
- B#16#22: Резервное напряжение в центральной стойке вышло из строя/ проблема устранена (BAF)
- B#16#23: Неисправность источника питания 24 В в центральной стойке / проблема устранена.
- B#16#25: Хотя бы одна буферная батарея по крайней мере в одной резервной центральной стойке разрядилась / проблема устранена (BATTF)
- B#16#26: Резервное напряжение по крайней мере в одной резервной центральной стойке вышло из строя/ проблема устранена (BAF)
- B#16#27: Неисправность источника питания 24 В по крайней мере в одной резервной центральной стойке
- B#16#31: Хотя бы одна буферная батарея по крайней мере в одной стойке расширения разрядилась/ проблема устранена (BATTF).
- B#16#32: Резервное напряжение по крайней мере в одной стойке расширения вышло из строя/ проблема устранена (BAF)
- B#16#33: Неисправность источника питания 24 В по крайней мере в одной стойке расширения / проблема устранена.

1.12 Организационный блок диагностических прерываний (OB82)

Описание

Если модуль, обладающий диагностическими свойствами, которому вы разрешили диагностические прерывания, распознает ошибку, он выдает на CPU запрос на диагностическое прерывание (как при наступающем, так и при уходящем событии). Затем операционная система вызывает OB82.

OB 82 содержит в своих локальных переменных как логический базовый адрес, так и четырехбайтовую диагностическую информацию неисправного модуля (см. следующую таблицу).

Если OB 82 не был запрограммирован, то CPU переходит в состояние STOP.

OB диагностических прерываний можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42.

За дополнительной информацией обратитесь к главе 13.

Локальные данные OB диагностических прерываний

Следующая таблица описывает временные (TEMP) переменные OB диагностических прерываний. В качестве имен переменных взяты имена по умолчанию OB82.

Переменная	Тип	Описание
OB82_EV_CLASS	BYTE	Класс события и идентификаторы: B#16#38: уходящее событие B#16#39: наступающее событие
OB82_FLT_ID	BYTE	Код ошибки (B#16#42)
OB82_PRIORITY	BYTE	Класс приоритета: 26 (по умолчанию для режима RUN) или 28 (режим STARTUP) На резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты 0 – 2: Номер стойки Бит 3: 0=резервный CPU, 1=основной CPU Биты 4 – 7: 1111
OB82_OB_NUMBR	BYTE	Номер OB (82)
OB82_RESERVED_1	BYTE	Резерв
OB82_IO_FLAG	BYTE	Модуль ввода: B#16#54 Модуль вывода: B#16#55
OB82_MDL_ADDR	INT	Базовый логический адрес модуля, в котором произошел отказ.
OB82_MDL_DEFECT	BOOL	Модуль неисправен
OB82_INT_FAULT	BOOL	Внутренняя ошибка
OB82_EXT_FAULT	BOOL	Внешняя ошибка
OB82_PNT_INFO	BOOL	Ошибка канала
OB82_EXT_VOLTAGE	BOOL	Внешнее питание вышло из строя
OB82_FLD_CONNCTR	BOOL	Не вставлен фронт-штекер
OB82_NO_CONFIG	BOOL	Модуль не сконфигурирован
OB82_CONFIG_ERR	BOOL	Неверные параметры в модуле

Переменная	Тип	Описание
OB82_MDL_TYPE	BYTE	Биты 0 – 3: Класс модуля Бит 4: Имеется информация канала Бит 5: Имеется информация пользователя Бит 6: Диагностическое прерывание из-за замены Бит 7: Резерв
OB82_SUB_MDL_ERR	BOOL	Субмодуль отсутствует или неисправен
OB82_COMM_FAULT	BOOL	Неисправность связи
OB82_MDL_STOP	BOOL	Рабочий режим (0: RUN, 1: STOP)
OB82_WTCH_DOG_FLT	BOOL	Сработал контроль времени
OB82_INT_PS_FLT	BOOL	Отказ внутреннего источника питания
OB82_PRIM_BATT_FLT	BOOL	Разрядилась батарея
OB82_BCKUP_BATT_FLT	BOOL	Выход из строя всего резервирования
OB82_RESERVED_2	BOOL	Резерв
OB82_RACK_FLT	BOOL	Выход из строя стойки расширения
OB82_PROC_FLT	BOOL	Выход из строя процессора
OB82_EPROM_FLT	BOOL	Сбой СППЗУ
OB82_RAM_FLT	BOOL	Сбой ОЗУ
OB82_ADU_FLT	BOOL	Ошибка АЦП/ЦАП
OB82_FUSE_FLT	BOOL	Выход из строя плавкого предохранителя
OB82_HW_INTR_FLT	BOOL	Потеряно аппаратное прерывание
OB82_RESERVED_3	BOOL	Резерв
OB82_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван OB

1.13 Организационный блок снятия/установки модулей (OB83)

Описание

Установка и снятие модулей контролируется внутри системы каждую секунду. Чтобы установка или снятие модуля распознавались CPU, необходимо чтобы между установкой и снятием прошло как минимум две секунды.

Каждая установка или снятие запроецированного модуля в режимах RUN, STOP и STARTUP (не разрешено удаление в этих режимах для блоков питания, CPU, адаптерных модулей и IM) приводит к прерыванию снятия/установки. Это прерывание вызывает у соответствующего CPU запись в диагностический буфер и в список состояний системы. Кроме того, в режиме RUN осуществляется запуск OB снятия/установки. Если этот OB не был запрограммирован, то CPU переходит в состояние STOP.

OB снятия/установки можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42.

За дополнительной информацией обращайтесь к главе 12.

Принцип действия OB83

При снятии запроецированного модуля в режиме RUN запускается OB83. Так как существование модулей контролируется только с интервалом в одну секунду, то сначала может быть обнаружена ошибка доступа при прямом обращении к модулю или обновлении образа процесса.

При установке модуля в запроецированный слот в режиме RUN операционная система проверяет соответствие вставленного модуля зарегистрированной конфигурации. Затем запускается OB 83 и при совпадении типов модулей осуществляется параметризация.

Локальные данные OB83

Следующая таблица описывает временные (TEMP) переменные OB снятия/установки. В качестве имен переменных взяты имена по умолчанию OB83.

Переменная	Тип данных	Описание
OB83_EV_CLASS	BYTE	Класс события и идентификаторы: B#16#38: модуль установлен B#16#39: модуль снят или не реагирует
OB83_FLT_ID	BYTE	Код ошибки (возможные значения B#16#61, B#16#63 или B#16#64)
OB83_PRIORITY	BYTE	Класс приоритета: 26 (по умолчанию для режима RUN) или 28 (режим STARTUP) На резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты 0 – 2: Номер стойки Бит 3: 0=резервный CPU, 1=основной CPU Биты 4 – 7: 1111
OB83_OB_NUMBR	BYTE	Номер OB (83)
OB83_RESERVED_1	BYTE	Резерв
OB83_MDL_TD	BYTE	Область: B#16#54: периферийные входы (PI) B#16#55: периферийные выходы (PQ)
OB83_MDL_ADDR	WORD	Логический базовый адрес соответствующего модуля
OB83_RACK_NUM	WORD	Номер стойки или номер станции DP (младший байт) и идентификатор мастер-системы DP (старший байт)
OB83_MDL_TYPE	WORD	Тип соответствующего модуля (W#16#0005: аналоговый модуль, W#16#0008: функциональный модуль, W#16#000C: CP, W#16#000F: цифровой модуль)
OB83_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван OB

Переменная OB83_MDL_TYPE имеет следующее значение в зависимости от кода ошибки:

- Код ошибки B#16#61: Модуль установлен, тип модуля верен (для класса событий B#16#38)
- OB83_MDL_TYPE: Модуль снят или не реагирует (для класса событий B#16#39)
- Код ошибки B#16#63: Фактический тип модуля
- OB83_MDL_TYPE: Модуль установлен, однако не верен тип модуля
- Код ошибки B#16#64: Фактический тип модуля
- OB83_MDL_TYPE: Модуль установлен, однако неисправен (не читается идентификатор типа)
- OB83_MDL_TYPE: Заданный в конфигурации тип модуля
- Код ошибки B#16#65: Модуль установлен, однако, имеется ошибка в назначении параметров модуля
- OB83_MDL_TYPE: Фактический тип модуля

1.14 Организационный блок аппаратных ошибок CPU (OB84)

Описание

Операционная система CPU вызывает OB 84, если распознается ошибка в интерфейсе с сетью MPI, с внутренней коммуникационной шиной (К-шиной) или в сопряжении с интерфейсным модулем для децентрализованной периферии.

Если возникла такая ошибка, а OB 84 не был запрограммирован, то CPU переходит в состояние STOP.

OB аппаратных ошибок CPU можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42.

За дополнительной информацией обратитесь к главе 12.

Локальные данные OB аппаратных ошибок

Следующая таблица содержит временные (TEMP) переменные OB аппаратных ошибок CPU. В качестве имен переменных взяты имена по умолчанию OB84.

Переменная	Тип	Описание
OB84_EV_CLASS	BYTE	Класс события и идентификатор: B#16#38: уходящее событие B#16#39: наступающее событие
OB84_FLT_ID	BYTE	Код ошибок (B#16#81)
OB84_PRIORITY	BYTE	Класс приоритета: 26 (по умолчанию для режима RUN или 28 (режим STARTUP) На резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты 0 – 2: Номер стойки Бит 3: 0=резервный CPU, 1=основной CPU Биты 4 – 7: 1111
OB84_OB_NUMBR	BYTE	Номер OB (84)
OB84_RESERVED_1	BYTE	Резерв
OB84_RESERVED_2	BYTE	Резерв
OB84_RESERVED_3	WORD	Резерв
OB84_RESERVED_4	DWORD	Резерв
OB84_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван OB

1.15 Организационный блок ошибок класса приоритета (OB85)

Описание

Операционная система CPU вызывает OB85 всякий раз, когда происходит одно из следующих событий:

- Стартовое событие для OB, который не был загружен.
- Ошибка при обращении операционной системы к модулю.
- Ошибка доступа к входам/выходам во время обновления образа процесса системой.

Примечание

Если OB85 не был запрограммирован, то в случае обнаружения одной из этих ошибок CPU переходит в состояние STOP.

Вы можете заблокировать или задержать и вновь разблокировать OB ошибок класса приоритета, используя SFC с номерами от 39 до 42. **За дополнительной информацией обратитесь к главе 12.**

Локальные данные для OB ошибок класса приоритета

Следующая таблица описывает временные (TEMP) переменные для OB ошибок класса приоритета. Имена переменных являются заданными по умолчанию именами OB85.

Переменная	Тип	Характеристика
OB85_EV_CLASS	BYTE	Класс события и идентификаторы: V#16#35 V#16#38 (только с кодами ошибок V#16#B3 и V#16#B4) V#16#39 (только с кодами ошибок V#16#B1 и V#16#B2)
OB85_FLT_ID	BYTE	Код ошибки (возможные значения: V#16#A1, V#16#A2, V#16#A3, V#16#B1, V#16#B2, V#16#B3, V#16#B4)
OB85_PRIORITY	BYTE	Класс приоритета: 26 (значение по умолчанию для режима RUN) или 28 (режим STARTUP) В резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты с 0 по 2: Номер стойки Бит 3: 0= резервный CPU, 1=основной CPU Биты с 4 по 7: 1111
OB85_OB_NUMBR	BYTE	Номер OB (85)
OB85_RESERVED_1	BYTE	Резерв
OB85_RESERVED_2	BYTE	Резерв
OB85_RESERVED_3	INT	Резерв
OB85_ERR_EV_CLASS	BYTE	Класс события, вызвавшего ошибку
OB85_ERR_EV_NUM	BYTE	Номер события, вызвавшего ошибку
OB85_OB_PRIOR	BYTE	Класс приоритета OB, который был активен, когда произошла ошибка
OB85_OB_NUM	BYTE	Номер OB, который был активен, когда произошла ошибка
OB85_DATE_TIME	DATE_AND_TIME	Дата и время дня, когда вызывался OB

Если вы хотите программировать OB85 в зависимости от возможных кодов ошибок, то рекомендуем вам организовать локальные переменные следующим образом:

Переменная	Тип
OB85_EV_CLASS	BYTE
OB85_FLT_ID	BYTE
OB85_PRIORITY	BYTE
OB85_OB_NUMBR	BYTE
OB85_DKZ23	BYTE
OB85_RESERVED_2	BYTE
OB85_Z1	WORD
OB85_Z23	DWORD
OB85_DATE_TIME	DATE_AND_TIME

Переменные, изменяемые по сравнению со значением по умолчанию, имеют в зависимости от кода ошибки следующее значение:

- Код ошибки В#16#A1

В результате вашей конфигурации, созданной с помощью STEP 7, ваша программа или операционная система создает стартовое событие для OB, который не загружен в CPU.

В#16#A2

В результате вашей конфигурации, созданной с помощью STEP 7, ваша программа или операционная система создает стартовое событие для OB, который не загружен в CPU.

OB85_Z1

Соответствующая локальная переменная запрошенного OB, которая определяется OB85_Z23.

OB85_Z23:

старшее слово: Класс и номер события, вызывающего вызов OB.

младшее слово: Программный уровень и OB, активный во время ошибки.
- Код ошибки В#16#A3

Ошибка при обращении операционной системы к модулю.

Идентификатор ошибки операционной системы.

OB85_Z1:

старший байт: 1: встроенная функция

2: таймер IEC

младший байт: 0: разрешающая способность при отсутствии ошибки

1: блок не загружен

2: ошибка длины области

3: ошибка защиты от записи

OB85_Z23

старшее слово: Номер блока

младшее слово: Относительный адрес команды MC7, вызывающей ошибку. Тип блока может быть взят из OB85_DKZ23 (В#16#88: OB, В#16#8C: FC, В#16#8E: FB, В#16#8A: DB).

- Код ошибки V#16#B1: Ошибка доступа к входам/выходам во время обновления всего образа процесса на входах.
- V#16#B2: Ошибка доступа к входам/выходам во время обновления всего образа процесса на выходах.
- OV85_Z1: Зарезервировано для внутреннего использования CPU.
- OV85_Z23: байт 0: Номер раздела образа процесса.
- байты 2,3: Номер байта входов/выходов, вызвавшего ошибку доступа к входам/выходам.

Вы получаете коды ошибки V#16#B1 и V#16#B2, если вы сконфигурировали повторную сигнализацию ошибок доступа к входам/выходам для обновления таблицы образа процесса системой.

- Код ошибки V#16#B3: Ошибка доступа к входам/выходам при обновлении входной таблицы образа процесса (появление/уход).
- V#16#B4: Ошибка доступа к входам/выходам при передаче выходной таблицы образа процесса модулям вывода (появление/уход).
- OV85_Z1: байт 0: Номер раздела образа процесса.
- OV85_Z23: байты 2,3: Номер байта входов/выходов, вызвавшего ошибку доступа к входам/выходам.

Вы получаете коды ошибки V#16#B3 и V#16#B4, если вы сконфигурировали сигнализацию ошибок доступа к входам/выходам при входе и выходе из состояния для обновления таблицы образа процесса системой.

1.16 Организационный блок неисправностей стоек (OB86)

Описание

Операционная система CPU вызывает OB 86 в случаях, когда распознается неисправность стойки расширения, master-системы DP или станции у децентрализованной периферии (как при наступающем, так и при уходящем событии).

Если OB 86 не был запрограммирован, а обнаруживается такая ошибка, то CPU переходит в состояние STOP.

OB 86 можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42.

За дополнительной информацией обращайтесь к главе 12.

Локальные данные OB неисправностей стоек

Следующая таблица описывает временные (TEMP) переменные OB неисправностей стоек. В качестве имен переменных взяты имена по умолчанию OB86.

Переменная	Тип	Описание
OB86_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#38: уходящее событие В#16#39: наступающее событие
OB86_FLT_ID	BYTE	Код ошибки: (возможные значения В#16#C1, В#16#C2, В#16#C3, В#16#C4, В#16#C5, В#16#C6, В#16#C7, В#16#C8)
OB86_PRIORITY	BYTE	Класс приоритета: 26 (значение по умолчанию для режима RUN) или 28 (режим STARTUP) На резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты 0 – 2: Номер стойки Бит 3: 0=резервный CPU, 1=основной CPU Биты 4 – 7: 1111
OB86_OB_NUMBR	BYTE	Номер OB (86)
OB86_RESERVED_1	BYTE	Резерв
OB86_RESERVED_2	BYTE	Резерв
OB86_MDL_ADDR	WORD	Зависит от кода ошибки
OB86_RACKS_FLTD	Array [0 ..31] of BOOL	Зависит от кода ошибки
OB86_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван OB

Если вы желаете запрограммировать OB 86 в зависимости от кодов ошибок, то рекомендуется локальные переменные организовывать следующим образом:

Переменная	Тип
OB86_EV_CLASS	BYTE
OB86_FLT_ID	BYTE
OB86_PRIORITY	BYTE
OB86_OB_NUMBR	BYTE
OB86_RESERVED_1	BYTE
OB86_RESERVED_2	BYTE
OB86_MDL_ADDR	WORD
OB86_Z23	DWORD
OB86_DATE_TIME	DATE_AND_TIME

Переменные, содержимое которых зависит от кодов ошибок, имеют следующее значение:

- Код ошибки В#16#C1: Неисправность стойки расширения
Логический базовый адрес IM

OB86_MDL_ADDR:
OB86_Z23:

Содержит по одному биту для каждой возможной стойки расширения:
Бит:
Бит 0: всегда 0
Бит 1: 1-я стойка расширения
:
:
Бит 21: 21-я стойка расширения
Бит 22 – 29 всегда 0
Бит 30: Неисправность по крайней мере одной стойки расширения в области SIMATIC S5
Бит 31: всегда 0

Значение: когда происходит событие, стойки расширения, обусловившие вызов OB86, индицируются как неисправные (соответствующие им биты устанавливаются). Стойки расширения, вышедшие из строя ранее, больше не индицируются. Когда неисправность устраняется, в коде ошибки сообщается о стойках расширения, снова ставших активными (соответствующие им биты устанавливаются).
- Код ошибки В#16#C2: Восстановление стойки расширения (с расхождением между ожидаемой и фактической конфигурацией)
Логический базовый адрес IM

OB86_MDL_ADDR:
OB86_Z23:

Содержит один бит для каждой возможной стойки расширения, см. код ошибки В#16#C1.
Значение установленного бита (для соответствующей стойки расширения)

 - имеются модули с неправильным идентификатором типа
 - отсутствуют запроектированные модули
 - хотя бы один модуль неисправен.
- Код ошибки В#16#C3: Децентрализованная периферия: неисправность master-системы. (Только наступающее событие ведет к старту OB 86 с кодом ошибки В#16#C3. Уходящее событие запускает OB86 с кодом ошибки В#16#C4 и классом события В#16#38. Восстановление любой slave-станции DP вызывает запуск OB86.)

OB86_MDL_ADDR: Логический базовый адрес master-устройства DP.

OB86_Z23:		Идентификатор master-системы DP Биты 0 – 7: резерв Биты 8 – 15: идентификатор master-системы DP Биты 16 – 31: резерв
• Код ошибки	B#16#C4:	Выход из строя станции DP.
	B#16#C5:	Децентрализованная периферия: сбой станции DP.
OB86_MDL_ADDR:		Логический базовый адрес master-устройства DP.
OB86_Z23:		Адрес неисправного slave-устройства DP: Биты 0 – 7: номер станции DP Биты 8 – 15: идентификатор master-системы DP Биты 16 – 30: логический базовый адрес slave-устройства S7 или диагностический адрес стандартного slave-устройства DP Бит 31: идентификатор ввода/вывода
• Код ошибки	B#16#C6:	Восстановление стойки расширения, но ошибка при назначении параметров модуля
OB86_MDL_ADDR:		Логический базовый адрес IM
OB86_Z23:		Содержит один бит для каждой возможной стойки расширения: Бит 0: всегда 0 Бит 1: 1-я стойка расширения : : Бит 21: 21-я стойка расширения Биты 22 – 30: резерв Бит 31: всегда 0 Значение установленного бита (в соответствующей стойке расширения): Существуют модули с неправильным идентификатором типа Существуют модули с неправильными или незадаанными параметрами
• Код ошибки	B#16#C7:	Восстановление станции DP, но ошибка при назначении параметров модуля
OB86_MDL_ADDR:		Логический базовый адрес master-устройства DP
OB86_Z23:		Адрес неисправного slave-устройства DP: Биты 0 – 7: номер станции DP Биты 8 – 15: идентификатор master-системы DP Биты 16 – 30: логический базовый адрес slave-устройства DP Бит 31: идентификатор ввода/вывода
• Код ошибки	B#16#C8:	Восстановление станции DP, однако имеется расхождение между спроектированной и фактической конфигурацией
OB86_MDL_ADDR:		Логический базовый адрес master-устройства DP
OB86_Z23:		Адрес неисправного slave-устройства DP: Биты 0 – 7: номер станции DP Биты 8 – 15: идентификатор master-системы DP Биты 16 – 30: логический базовый адрес slave-устройства DP Бит 31: идентификатор ввода/вывода

1.17 Организационный блок коммуникационных ошибок (OB87)

Описание

Операционная система CPU вызывает OB 87 в случаях, когда наступает событие, которое было вызвано коммуникационной ошибкой.

Если OB 87 не был запрограммирован, то CPU при обнаружении такого события переходит в состояние STOP.

OB коммуникационных ошибок можно запретить или отложить и вновь разрешить с помощью SFC 39 – 42.

За дополнительной информацией обратитесь к главе 12.

Локальные данные OB87

Следующая таблица описывает временные (TEMP) переменные OB коммуникационных ошибок. В качестве имен переменных взяты имена по умолчанию OB87.

Переменная	Тип	Описание
OB87_EV_CLASS	BYTE	Класс события и идентификаторы: B#16#35
OB87_FLT_ID	BYTE	Код ошибки: (возможные значения 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 или B#16#E6)
OB87_PRIORITY	BYTE	Класс приоритета: 26 (значение по умолчанию для режима RUN) или 28 (режим STARTUP) На резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты 0 – 2: Номер стойки Бит 3: 0=резервный CPU, 1=основной CPU Биты 4 – 7: 1111
OB87_OB_NUMBR	BYTE	Номер OB (87)
OB87_RESERVED_1	BYTE	Резерв
OB87_RESERVED_2	BYTE	Резерв
OB87_RESERVED_3	WORD	Зависит от кода ошибки
OB87_RESERVED_4	DWORD	Зависит от кода ошибки
OB87_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван OB

Переменные, зависящие от кода ошибок, имеют следующие значения:

- Код ошибки

B#16#D2:

B#16#D3:

B#16#D4:

B#16#D5:

Передача диагностических записей в настоящее время невозможна.

Синхронизационные сообщения не могут быть посланы (master).

Недопустимый скачок времени из-за синхронизации часов.

Ошибка при приеме времени синхронизации (slave).
 - OB87_RESERVED_3:

OB87_RESERVED_4:

 - Код ошибки

B#16#E1:

B#16#E3:

B#16#E4:

Не содержит дополнительной информации.

Не содержит дополнительной информации.

Неверный идентификатор кадра при связи с помощью глобальных данных.

Ошибка длины кадра при связи с помощью глобальных данных.

Принят недопустимый номер пакета GD.
 - OB87_RESERVED_3:

OB87_RESERVED_4:

 - Код ошибки

старший байт:

младший байт:

B#16#E2:

идентификатор интерфейса (0: K-шина, 1: MPI)

номер GD-контура

Не содержит дополнительной информации.

Состояние пакета GD не может быть внесено в DB
 - OB87_RESERVED_3:

OB87_RESERVED_4:

 - Код ошибки

старшее слово:

младшее слово:

B#16#E5:

Номер DB

Не содержит дополнительной информации.

номер GD-контура (старший байт), номер пакета GD (младший байт)

Ошибка доступа к DB при обмене данными через коммуникационные функциональные блоки
 - OB87_RESERVED_3:

OB87_RESERVED_4:

старшее слово:

младшее слово:

Зарезервировано для внутреннего использования CPU.

Номер блока с вызвавшей ошибку командой MC7.

Относительный адрес команды MC7, вызвавшей ошибку.
- Тип блока может быть считан из OB_87_RESERVED_1 (B#16#88: OB, B#16#8A: DB, B#16#8C: FC, B#16#8E: FB).
- Код ошибки

B#16#E6:

Групповой статус GD не может быть занесен в DB.
 - OB87_RESERVED_3:

OB87_RESERVED_4:

Номер DB.

Не содержит дополнительной информации.

1.18 Организационный блок фонового режима (OB90)

Описание

С помощью STEP 7 можно контролировать максимальное время цикла и гарантировать минимальное время цикла. Если время исполнения OB1, включая все вложенные прерывания и системные операции, оказывается меньше заданной вами минимальной длительности цикла, то операционная система реагирует следующим образом:

- Она вызывает OB фонового режима (если он имеется в CPU).
- Она задерживает следующий запуск OB1 (если в CPU отсутствует OB 90).

Принцип действия OB90

OB90 имеет самый низкий приоритет среди всех остальных OB. Он прерывается любой системной операцией и любым прерыванием и возобновляется только в том случае, если заданное минимальное время цикла еще не достигнуто. Исключение составляет обработка SFC и SFB, которые были запущены из OB90. Они исполняются с приоритетом OB1 и поэтому не прерываются OB1. Контроля длительности OB 90 не ведется.

Программа пользователя в OB90 обрабатывается, начиная с первой команды, в следующих ситуациях:

- после теплого, холодного или горячего рестарта
- после удаления блока, исполняемого в OB90 (с помощью STEP 7)
- после загрузки OB 90 в CPU в режиме RUN
- по окончании фонового цикла.

Замечание

В случае конфигураций, в которых нет большой разницы между минимальным временем цикла и контрольным временем цикла, вызовы SFC и SFB в фоновом OB могут привести к непредусмотренному превышению времени цикла.

Локальные данные OB90

Следующая таблица описывает временные (TEMP) OB 90. В качестве имен переменных взяты имена по умолчанию OB90.

Переменная	Тип данных	Описание
OB90_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#11: активен
OB90_STRT_INF	BYTE	В#16#91: теплый рестарт/холодный рестарт/ горячий рестарт В#16#92: блок удален В#16#93: загрузка OB 90 в CPU в режиме RUN В#16#95: окончание фонового цикла
OB90_PRIORITY	BYTE	Класс приоритета: 29 (соответствует приоритету 0.29)
OB90_OB_NUMBR	BYTE	Номер OB (90)
OB90_RESERVED_1	BYTE	резерв
OB90_RESERVED_2	BYTE	резерв
OB90_RESERVED_3	INT	резерв
OB90_RESERVED_4	INT	резерв
OB90_RESERVED_5	INT	резерв
OB90_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван OB

1.19 Организационные блоки запуска (ОВ 100, ОВ101 и ОВ102)

Типы запуска

Различают следующие типы запуска:

- Горячий рестарт (нет у S7-300 и S7-400H)
- Теплый рестарт
- Холодный рестарт

В следующей таблице вы можете видеть, какой ОВ вызывается операционной системой во время запуска.

Тип запуска	Соответствующий ОВ
Горячий рестарт	ОВ101
Теплый рестарт	ОВ100
Холодный рестарт	ОВ102

За более подробной информацией о типах запуска обратитесь к руководствам "Programming and Hardware Configuration with STEP 7 [Программирование и конфигурирование аппаратных средств с помощью STEP 7]" и "S7-400H Programmable Controller [Программируемый контроллер S7-400H]".

Стартовые события для запуска

CPU выполняет запуск в следующих случаях:

- после включения питания
- всякий раз, когда вы переводите переключатель режимов работы из STOP в RUN-P
- после запроса, использующего коммуникационную функцию (посредством команды меню из устройства программирования или вызова коммуникационных функциональных блоков 19 "START" или 21 "RESUME" в другом CPU)
- синхронизация в многопроцессорном режиме
- в H-системе после соединения (только в резервном CPU).

В зависимости от стартового события, конкретного CPU и его параметров вызывается соответствующий ОВ запуска (ОВ100, ОВ101 или ОВ102). Посредством подходящего программирования вы можете настраивать определенные параметры вашей циклической программы (исключение: в H-системе, когда подключен резервный CPU, в резервном CPU имеет место запуск, но ОВ запуска не вызывается).

Локальные данные для ОВ запуска

Следующая таблица описывает временные (TEMP) переменные для стартовых ОВ. Имена переменных являются заданными по умолчанию именами ОВ100.

Переменная	Тип	Характеристика
OB10x_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#13: активный
OB10x_STRTUP	BYTE	Запрос запуска: В#16#81: Ручной теплый рестарт В#16#82: Автоматический теплый рестарт В#16#83: Запрос ручного горячего рестарта В#16#84: Запрос автоматического горячего рестарта В#16#85: Запрос ручного холодного рестарта В#16#86: Запрос автоматического холодного рестарта В#16#87: Главный: Запрос ручного холодного рестарта В#16#88: Главный: Запрос автоматического холодного рестарта В#16#8A: Главный: Запрос ручного теплого рестарта В#16#8B: Главный: Запрос автоматического теплого рестарта В#16#8C: Резервный: Запрос ручного рестарта В#16#8D: Резервный: Запрос автоматического рестарта
OB10x_PRIORITY	BYTE	Класс приоритета: 27
OB10x_OB_NUMBR	BYTE	Номер ОВ (100, 101 или 102)
OB10x_RESERVED_1	BYTE	Резерв
OB10x_RESERVED_2	BYTE	Резерв
OB10x_STOP	WORD	Номер события, вызвавшего останов CPU
OB10x_STRT_INFO	DWORD	Дополнительная информация о текущем запуске
OB10x_DATE_TIME	DATE_AND_TIME	Дата и время дня, когда вызывался ОВ

Следующая таблица показывает переменные OB100_STR_INFO и OB101_STR_INFO.

№ бита	Значение	Возможные двоичные значения	Объяснение
31 - 24	Стартовая информация	0000 xxxx 0100 xxxx 1000 xxxx 0001 xxxx 0010 xxxx xxxx xxx0 xxxx xxx1 xxxx xx0x xxxx xx1x xxxx x0xx	Номер стойки 0 (только H CPU) Номер стойки 1 (только H CPU) Номер стойки 2 (только H CPU) Многопроцессорный режим (только S7-400) Работа более одного CPU в сегментированной стойке (только S7-400) Нет различия между ожидаемой и фактической конфигурацией (только S7-300) Есть различие между ожидаемой и фактической конфигурацией (только S7-300) Нет различия между ожидаемой и фактической конфигурацией Есть различие между ожидаемой и фактической конфигурацией Не H CPU

№ бита	Значение	Возможные двоичные значения	Объяснение
		xxxx x1xx xxxx 0xxx xxxx 1xxx	Н CPU Часы для отметки времени при последнем включении питания без резервного батарейного питания Часы для отметки времени при последнем включении питания с резервным батарейным питанием
23 - 16	Запуск только что завершился	0000 0001 0000 0011 0000 0100	Теплый рестарт в многопроцессорной системе обработки без изменения установки в CPU согласно назначению параметров (только S7-400) Рестарт (теплый), запущенный с помощью переключателя режимов работы Рестарт (теплый), запущенный командой через MPI
		0000 0101 0000 0011 0000 1000 0000 1010 0000 1011 0000 1100 0001 0000 0001 0001 0001 0011 0001 0100 0010 0000 0010 0001 0010 0011 0010 0100 1010 0000	Холодный рестарт в многопроцессорной системе обработки без изменения установки в CPU согласно назначению параметров (только S7-400) Холодный рестарт, запущенный с помощью переключателя режимов работы Холодный рестарт, запущенный командой через MPI Горячий рестарт в многопроцессорной системе обработки без изменения установки в CPU согласно назначению параметров (только S7-400) Горячий рестарт, запущенный с помощью переключателя режимов работы (только S7-400) Горячий рестарт, запущенный командой через MPI (только S7-400) Автоматический рестарт (теплый) после включения питания при наличии резервного батарейного питания Автоматический холодный рестарт после включения питания при наличии резервного батарейного питания Рестарт (теплый), запущенный с помощью переключателя режимов работы; последнее включение питания при наличии резервного батарейного питания Рестарт (теплый), запущенный командой через MPI; последнее включение питания при наличии резервного батарейного питания Автоматический рестарт (теплый) после включения питания при наличии резервного батарейного питания (со сбросом памяти системой) Автоматический холодный рестарт после включения питания при наличии резервного батарейного питания (со сбросом памяти системой) Рестарт (теплый), запущенный с помощью переключателя режимов работы; последнее включение питания без резервного батарейного питания Рестарт (теплый), запущенный командой через MPI; последнее включение питания без резервного батарейного питания Автоматический горячий рестарт после включения питания при наличии резервного батарейного питания согласно назначению параметров (только S7-400)

№ бита	Значение	Возможные двоичные значения	Объяснение
15 - 12	Допустимость автоматического запуска	0000	Автоматический запуск запрещен, требуется сброс памяти
		0001	Автоматический запуск запрещен, необходимы изменения параметров и т.д.
		0111	Автоматический запуск (теплый) разрешен
		1111	Автоматический запуск (теплый/горячий) разрешен (только S7-400)
11 - 8	Допустимость ручного запуска	0000	Запуск запрещен, запрошен сброс памяти
		0001	Запуск запрещен, запрошены изменения параметров и т.д.
		0111	Запуск (теплый) разрешен
		1111	Запуск (теплый/горячий) разрешен (только S7-400)
7 - 0	Последнее допустимое вмешательство или установка автоматического запуска при включении питания	0000 0000	Нет запуска
		0000 0001	Теплый рестарт в многопроцессорной системе без изменения установки в CPU согласно назначению параметров (только S7-400)
		0000 0011	Рестарт (теплый), запущенный с помощью переключателя
		0000 0100	Рестарт (теплый), запущенный командой через MPI
		0000 1010	Горячий рестарт в многопроцессорной системе без изменения установки в CPU согласно назначению параметров (только S7-400)
		0000 1011	Горячий рестарт, запущенный с помощью переключателя режимов работы (только S7-400)
		0000 1100	Горячий рестарт, запущенный командой через MPI (только S7-400)
		0001 0000	Автоматический рестарт (теплый) после включения питания при наличии резервного батарейного питания
		0001 0011	Рестарт (теплый), запущенный командой через MPI; последнее включение питания при наличии резервного батарейного питания
		0001 0100	Рестарт (теплый), запущенный командой через MPI; последнее включение питания при наличии резервного батарейного питания
		0010 0000	Автоматический рестарт (теплый) после включения питания при наличии резервного батарейного питания (со сбросом памяти системой)
		0010 0011	Рестарт (теплый), запущенный с помощью переключателя режимов работы; последнее включение питания без резервного батарейного питания
		0010 0100	Рестарт (теплый), запущенный командой через MPI; последнее включение питания без резервного батарейного питания
		1010 0000	Автоматический горячий рестарт после включения питания при наличии резервного батарейного питания согласно назначению параметров (только S7-400)

1.20 Организационный блок ошибок про граммирования (OB121)

Описание

Операционная система CPU вызывает OB 121 в тех случаях, когда происходит событие, вызванное ошибкой, связанной с обработкой программы. Если, например, вы вызвали в своей программе блок, который не был загружен в CPU, то вызывается OB 121.

Принцип работы OB ошибок программирования

OB121 выполняется в том же классе приоритета, что и прерванный блок. Если OB121 не был запрограммирован, то CPU переходит в состояние STOP. S7 предоставляет в распоряжение следующие SFC, с помощью которых вы можете маскировать и демаскировать стартовые события OB 121 во время обработки вашей программы:

- SFC36 (MSK_FLT): маскирует определенные коды ошибок
- SFC37 (DMSK_FLT): демаскирует коды ошибок, которые были замаскированы с помощью SFC36
- SFC38 (READ_ERR): читает регистр ошибок

Локальные данные ОВ ошибок программирования

Следующая таблица описывает временные (TEMP) переменные ОВ ошибок программирования. В качестве имен переменных взяты имена по умолчанию ОВ121.

Переменная	Тип	Описание
OB121_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#25
OB121_SW_FLT	BYTE	Код ошибки : (возможные значения В#16#21, В#16#22, В#16#23, В#16#24, В#16#25, В#16#26, В#16#27, В#16#28, В#16#29, В#16#30, В#16#31, В#16#32, В#16#33, В#16#34, В#16#35, В#16#3A, В#16#3C, В#16#3D, В#16#3E или В#16#3F)
OB121_PRIORITY	BYTE	Класс приоритета: класс приоритета ОВ, в котором произошла ошибка На резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты 0 – 2: Номер стойки Бит 3: 0=резервный CPU, 1=основной CPU Биты 4 – 7: 1111
OB121_OB_NUMBR	BYTE	Номер ОВ (121)
OB121_BLK_TYPE	BYTE	Тип блока, в котором произошла ошибка (для S7-300 сюда не заносится никакого действительного значения): В#16#88: ОВ, В#16#8A: DB, В#16#8C: FC, В#16#8E: FB
OB121_RESERVED_1	BYTE	резерв
OB121_FLT_REG	WORD	Источник ошибки (зависит от кода ошибки). Например: Регистр, в котором возникла ошибка преобразования Неправильный адрес (для ошибки чтения/записи) Неправильный номер таймера, счетчика или блока Неправильный идентификатор области памяти
OB121_BLK_NUM	WORD	Номер блока с командой MC7, вызвавшей ошибку (для S7- 300 сюда не заносится никакого действительного номера)
OB121_PRG_ADDR	WORD	Относительный адрес команды MC7, вызвавшей ошибку (для S7-300 сюда не заносится никакого действительного значения)
OB121_DATE_TIME	DATE_AND_ TIME	Дата и время, когда был вызван ОВ

Переменные, зависящие от кодов ошибок, имеют следующие значения:

• Код ошибки	B#16#21:	Ошибка преобразования в BCD-код.
OB121_FLT_REG:		Идентификатор соответствующего регистра (W#16#0000: аккумулятор 1)
• Код ошибки	B#16#22:	Ошибка длины области при чтении
	B#16#23:	Ошибка длины области при записи
	B#16#28:	Обращение для чтения к байту, слову или двойному слову с указателем, битовый адрес которого не равен нулю.
	B#16#29:	Обращение для записи к байту, слову или двойному слову с указателем, битовый адрес которого не равен нулю.
OB121_FLT_REG:		Ошибочный байтовый адрес. Область данных и тип доступа можно считать из OB121_RESERVED_1.
OB121_RESERVED_1:		Биты 7 – 4 тип доступа. 0: битовый доступ, 1: байтовый доступ, 2: доступ к слову, 3: доступ к двойному слову Биты 3 – 0 область памяти: 0: область периферии, 1: таблица образа процесса на входах, 2: таблица образа процесса на выходах, 3: память с побитовым доступом (меркеры), 4: глобальный DB, 5: экземплярный DB, 6: собственные локальные данные, 7: локальные данные вызывающего блока
• Код ошибки	B#16#24:	Ошибка области при чтении
	B#16#25:	Ошибка области при записи
OB121_FLT_REG:		Содержит в младшем байте идентификатор недопустимой области (B#16#86 из области собственных локальных данных)
• Код ошибки	B#16#26:	Ошибка номера таймера
	B#16#27:	Ошибка номера счетчика
OB121_FLT_REG:		Недопустимый номер
• Код ошибки	B#16#30:	Попытка записи в защищенный от записи глобальный DB
	B#16#31:	Попытка записи в защищенный от записи экземплярный DB
	B#16#32:	Ошибка номера DB при обращении к глобальному DB
	B#16#33:	Ошибка номера DB при обращении к экземпляльному DB
OB121_FLT_REG:		Недопустимый номер DB
• Код ошибки	B#16#34:	Ошибка номера FC при вызове FC
	B#16#35:	Ошибка номера FB при вызове FB
	B#16#3A:	Обращение к незагруженному DB; номер DB находится в допустимом диапазоне
	B#16#3C:	Обращение к незагруженной FC; номер FC находится в допустимом диапазоне
	B#16#3D:	Обращение к незагруженной SFC; номер SFC лежит в допустимом диапазоне
	B#16#3E:	Обращение к незагруженному FB; номер FB лежит в допустимом диапазоне
	B#16#3F:	Обращение к незагруженному SFB; номер SFB лежит в допустимом диапазоне
OB121_FLT_REG:		Недопустимый номер

1.21 Организационный блок ошибок доступа к периферии (OB122)

Описание

Операционная система CPU вызывает OB 122 в случае, если при обращении к данным какого-либо модуля происходит ошибка. Если, например, CPU распознает ошибку чтения при обращении к данным модуля ввода/вывода, то операционная система вызывает OB122.

Принцип работы OB ошибок доступа к периферии

OB122 выполняется в том же классе приоритета, что и прерванный блок. Если OB 122 не запрограммирован, то CPU переходит из RUN в STOP.

S7 предоставляет следующие SFC, с помощью которых можно маскировать и демаскировать стартовые события OB 122 во время исполнения вашей программы:

- SFC36 (MSK_FLT): маскирует определенные коды ошибок
- SFC37 (DMSK_FLT): демаскирует коды ошибок, замаскированные с помощью SFC36
- SFC38 (READ_ERR): читает регистр ошибок

Локальные данные OB ошибок доступа к периферии

Следующая таблица описывает временные (TEMP) переменные OB ошибок доступа к периферии. В качестве имен переменных взяты имена по умолчанию OB122.

Переменная	Тип	Описание
OB122_EV_CLASS	BYTE	Класс события и идентификаторы: В#16#29
OB122_SW_FLT	BYTE	Код ошибки: В#16#42 Для S7-300 и CPU 417: ошибка доступа к периферии при чтении Для всех остальных CPU S7-400: ошибка при первом обращении для чтения после появления ошибки В#16#43 Для S7-300 и CPU 417: ошибка доступа к периферии при записи Для всех остальных CPU S7-400: ошибка при первом обращении для записи после появления ошибки В#16#44 (только для S7-400, кроме CPU 417) ошибка при n-ом обращении для чтения ($n > 1$) после появления ошибки В#16#45 (только для S7-400, кроме CPU 417) ошибка при n-ом обращении для записи ($n > 1$) после появления ошибки

Переменная	Тип	Описание
OB122_PRIORITY	BYTE	Класс приоритета: Класс приоритета OB, в котором произошла ошибка На резервном CPU H-системы в режиме LINK-UP, STARTUP или UPDATE: Биты 0 – 2: Номер стойки Бит 3: 0=резервный CPU, 1=основной CPU Биты 4 – 7: 1111
OB122_OB_NUMBR	BYTE	Номер OB (122)
OB122_BLK_TYPE	BYTE	Тип блока, в котором произошла ошибка (B#16#88: OB, B#16#8A: DB, B#16#8C: FC, B#16#8E: FB) (для S7-300 здесь не вводится никакого действительного значения)
OB122_MEM_AREA	BYTE	Область памяти и тип доступа: Биты 7 – 4 тип доступа: 0: битовый доступ, 1: байтовый доступ, 2: доступ к слову, 3: доступ к двойному слову Биты 3 – 0 область памяти: 0: область периферии, 1: образ процесса на входах, 2: образ процесса на выходах
OB122_MEM_ADDR	WORD	Адрес в памяти, на котором произошла ошибка
OB122_BLK_NUM	WORD	Номер блока с командой MC7, вызвавшей ошибку (для S7-300 здесь не вводится никакого действительного номера)
OB122_PRG_ADDR	WORD	Относительный адрес команды MC7, вызвавшей ошибку (для S7-300 здесь не вводится никакого действительного значения)
OB122_DATE_TIME	DATE_AND_TIME	Дата и время, когда был вызван OB

2 Общие параметры для SFC

2.1 Оценка ошибок с помощью выходного параметра RET_VAL

Виды информации об ошибках

Системная функция (SFC), исполняемая в вашей пользовательской программе показывает, смог ли CPU успешно выполнить функцию SFC.

Вы можете получить информацию о любых произошедших ошибках двумя способами:

- в бите BR слова состояния
- в выходном параметре RET_VAL (возвращаемое значение)

Замечание

Перед анализом выходных параметров, относящихся к SFC, вы всегда должны выполнять следующие шаги:

- Сначала проанализируйте бит BR слова состояния.
- Затем проверьте выходной параметр RET_VAL.

Если бит BR показывает, что произошла ошибка, или если RET_VAL содержит код общей ошибки, то вам нельзя использовать выходной параметр SFC!

Информация об ошибках в возвращаемом значении

Системная функция (SFC) через значение "0" бита двоичного результата (BR) слова состояния показывает, что при исполнении функции произошла ошибка. Некоторые системные функции предоставляют в распоряжение дополнительный код ошибки на выходе, известном как возвращаемое значение (RET_VAL). Если в выходной параметр RET_VAL вводится общая ошибка (объяснение см. ниже), то она отображается только посредством значения "0" бита BR слова состояния.

Возвращаемое значение относится к типу данных "целое число" (INT).

Отношение возвращаемого значения к значению "0" показывает, появилась ли при исполнении функции ошибка.

CPU исполняет SFC	BR	Возвращаемое значение	Знак целого числа
с ошибкой (ами)	0	меньше, чем "0"	отрицательный (знаковый бит равен "1")
без ошибки	1	больше или равно "0"	положительный (знаковый бит равен "0")

Реакция на информацию об ошибках

Имеются следующие два различных типа кодов ошибки в RET_VAL:

- общий код ошибки, который могут выдавать все системные функции, и
- специфический код ошибки, который SFC может выдавать в зависимости от своей конкретной функции.

Вы можете написать свою программу таким образом, что она будет реагировать на ошибки, возникающие при исполнении системной функции. Этим вы препятствуете возникновению последующих ошибок, являющихся результатом первой ошибки.

Общая и специфическая информация об ошибках

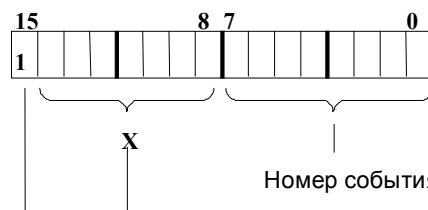
Возвращаемое значение (RET_VAL) системной функции предоставляет в распоряжение один из двух следующих типов кодов ошибки:

- Общий код ошибки, относящийся к ошибкам, которые могут возникнуть в любой системной функции.
- Специфический код ошибки, который относится только к конкретной системной функции.

Хотя тип данных выходного параметра RET_VAL целый (INT), коды ошибок системных функций группируются в соответствии с шестнадцатеричными значениями. Если вы хотите проанализировать возвращаемое значение и сравнить это значение с кодами ошибок, которые приведены в данном руководстве, то отобразите код ошибки в шестнадцатеричном формате.

Следующий рисунок показывает структуру кода ошибки системной функции в шестнадцатеричном формате.

Код ошибки, например, W#16#8081



x = '0' указывает, что это **специфический код ошибки SFC**. Коды ошибок Перечислены в разделе, озаглавленном "Информация об ошибках", в описаниях отдельных системных функций.
x > '0' указывает, что это **общий код ошибки SFC**. В этом случае x – это номер параметра SFC, вызвавшего ошибку. Возможные коды ошибок показаны в следующей таблице.

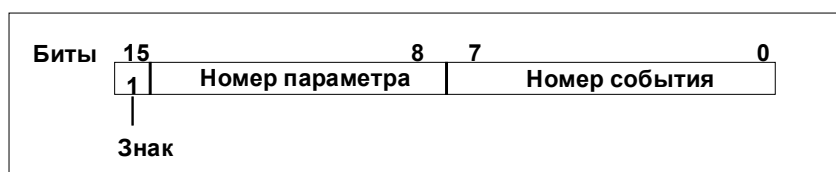
Если знаковый бит = 1, то это указывает, что произошла ошибка.

Общая информация об ошибках

Общий код ошибки показывает ошибки, которые могут встречаться во всех системных функциях. Общий код ошибки состоит из двух следующих номеров:

- Номер параметра в диапазоне от 1 до 111, где 1 указывает на первый параметр, 2 - на второй параметр SFC и т.д.
- Номер события в диапазоне от 1 до 127. Номер события указывает на то, что произошла синхронная ошибка.

В следующей таблице перечислены коды для общих ошибок и объяснение каждой ошибки.



Замечание

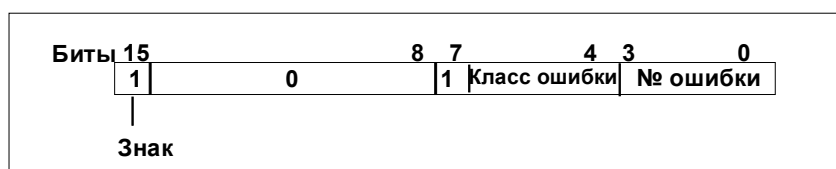
Если в RET_VAL был записан общий код ошибки, то возможны следующие ситуации:

- Возможно, была запущена или уже выполнена операция, относящаяся к SFC.
- Возможно, что при выполнении этой операции произошла специфическая ошибка SFC. Однако в результате общей ошибки, произошедшей позднее, специфическая ошибка не могла больше отображаться.

Специфическая информация об ошибках

Некоторые системные функции (SFC) обладают возвращаемым значением, которое предоставляет в распоряжение специфический код ошибки. Этот код ошибки показывает, что во время обработки функции появилась ошибка, которая относится к определенной системной функции (см. рисунок). Специфический код ошибки состоит из двух следующих чисел:

- Класс ошибки от 0 до 7.
- Номер ошибки от 0 до 15.



Общие коды ошибок

Следующая таблица объясняет общие коды возвращаемого значения. Код ошибки отображается в шестнадцатеричном формате. Буква х в каждом кодовом номере служит только как символ-заполнитель и представляет номер параметра системной функции, вызвавшего ошибку.

Код ошибки (W#16#...)	Объяснение
8x7F	Внутренняя ошибка Этот код ошибки указывает на внутреннюю ошибку в параметре х. Эта ошибка была вызвана не пользователем и не может быть им устранена.
8x22 8x23	Ошибка длины области при чтении параметра. Ошибка длины области при записи параметра. Этот код ошибки показывает, что параметр х полностью или частично находится за пределами диапазона адресов или что длина битового поля для параметра типа ANY не кратна 8.
8x24 8x25	Ошибка области при чтении параметра Ошибка области при записи параметра. Этот код ошибки показывает, что параметр х находится в области, которая недопустима для системной функции. За информацией о недопустимых областях обращайтесь к описаниям отдельных функций.
8x26	Параметр содержит слишком большой номер таймера. Этот код ошибки показывает, что таймер, заданный в параметре х, не существует.
8x27	Параметр содержит слишком большой номер счетчика (ошибка номера счетчика). Этот код ошибки показывает, что счетчик, указанный в параметре х, не существует.
8x28 8x29	Ошибка выравнивания при чтении параметра. Ошибка выравнивания при записи параметра. Этот код ошибки показывает, что ссылка на параметр х является битовым адресом, не равным 0.
8x30 8x31	Параметр находится в защищенном от записи глобальном DB. Параметр находится в защищенном от записи экземплярном DB. Этот код ошибки показывает, что параметр х находится в защищенном от записи блоке данных. Если этот блок данных был открыт самой системной функцией, то системная функция всегда возвращает значение W#16#8x30.
8x32 8x34 8x35	Параметр содержит слишком большой номер DB (ошибка номера DB). Параметр содержит слишком большой номер FC (ошибка номера FC). Параметр содержит слишком большой номер FB (ошибка номера FB). Этот код ошибки показывает, что параметр х содержит номер блока, который больше, чем максимально допустимый номер блока.
8x3A 8x3C 8x3E	Параметр содержит номер незагруженного DB. Параметр содержит номер незагруженного FC. Параметр содержит номер незагруженного FB.
8x42 8x43	Имела место ошибка доступа в то время, когда система пыталась считать параметр из периферийной области входов. Имела место ошибка доступа в то время, когда система пыталась считать параметр из периферийной области выходов.
8x44 8x45	Ошибка при n-ом (n > 1) обращении для чтения после появления ошибки. Ошибка при n-ом (n > 1) обращении для записи после появления ошибки. Этот код ошибки показывает, что получен отказ в доступе к требуемому параметру.

2.2 Значение параметров REQ, RET_VAL и BUSY у асинхронных SFC

Асинхронные SFC

Асинхронно работающие SFC – это SFC, которые вызываются более одного раза до того, как они завершат выполнение своих функций. Следующие SFC всегда выполняются асинхронно или в определенных ситуациях:

- SFC7 "DP_PRAL"
- SFC11 "DPSYC_FR"
- SFC13 "DPNRM_DG"
- SFC51 "RDSYSST"
- SFC55 "WR_PARM"
- SFC56 "WR_DPARM"
- SFC57 "PARM_MOD"
- SFC58 "WR_REC"
- SFC59 "RD_REC"
- SFC65 "X_SEND"
- SFC67 "X_GET"
- SFC68 "X_PUT"
- SFC69 "X_ABORT"
- SFC72 "I_GET"
- SFC73 "I_PUT"
- SFC74 "I_ABORT"
- SFC90 "H_CTRL"

Идентификация задания

Если вы запускаете аппаратное прерывание, выдаете команды управления для slave-устройств DP, запускаете передачу данных или прерываете неконфигурированное соединение с помощью одной из вышеперечисленных SFC, а затем снова вызываете ту же SFC до завершения текущего задания, то реакция SFC будет зависеть от того, включает ли в себя второй вызов то же самое задание.

Следующая таблица объясняет, какие входные параметры определяют задание для каждой из этих SFC. Если эти параметры совпадают с параметрами еще не завершенного задания, то этот вызов SFC считается продолжающимся (follow-on) вызовом.

SFC	Задание идентифицируется через...
7 "DP_PRL"	IOID, LADDR
11"DPSYC_FR"	LADDR, GROUP, MODE
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
90 "H_CTRL"	MODE, SUBMODE

Входной параметр REQ

Входной параметр REQ (request [запрос]) используется исключительно для запуска задания:

- Если вы вызываете SFC для выполнения задания, которое в данный момент не активно, то задание запускается с помощью REQ = 1 (ситуация 1).
- Если определенное задание было запущено и еще не завершено, а вы вызываете SFC снова, чтобы выполнить то же самое задание (например, в ОВ циклических прерываний), то SFC не анализирует REQ (ситуация 2).

Выходные параметры RET_VAL и BUSY

С помощью выходных параметров RET_VAL и BUSY отображается состояние исполнения.

- В ситуации 1 (первый вызов при REQ=1) в RET_VAL вводится W#16#7001, если системные ресурсы свободны и входные параметры правильны. Тогда устанавливается BUSY.
Если требуемые системные в данный момент используются или входные параметры содержат ошибки, то в RET_VAL вносится соответствующий код ошибки, а BUSY имеет значение 0. (См. также замечание в разделе 2.1).

- В ситуации 2 (вызов, когда активно то же самое задание) в RET_VAL вносится W#16#7002 (это предупреждение о том, что это задание еще обрабатывается), а BUSY устанавливается.
- Следующее относится к последнему вызову задания:
 - Если не произошло ошибки, то с помощью SFC13 "DPNRM_DG," SFC67 "X_GET" и SFC72 "I_GET" в RET_VAL в виде положительного числа байтов вводится количество передаваемых данных. Тогда BUSY имеет значение 0.
Если происходит ошибка, то RET_VAL содержит информацию об этой ошибке, а BUSY 0. (Обратитесь также к замечанию в разделе 2.1!)
 - Если не произошло ошибки, то с помощью SFC59 "RD_REC" в RET_VAL вводится размер записи данных в байтах или 0 (см. раздел 2.1!). В этом случае BUSY имеет значение 0.
Если происходит ошибка, то в RET_VAL вводится код ошибки, а BUSY имеет значение 0. (Обратитесь также к замечанию в разделе 2.1!)
 - У всех остальных SFC, если задание было выполнено без ошибок, то в RET_VAL вводится 0, а BUSY имеет значение 0.
Если происходит ошибка, то в RET_VAL вводится код ошибки, а BUSY имеет значение 0. (Обратитесь также к замечанию в разделе 2.1!)

Замечание

Если первый и последний вызов приходят одновременно, то реакция для RET_VAL и BUSY будет такой же, как описано для последнего вызова.

Обзор

Следующая таблица дает обзор описанных выше соотношений. В частности, она показывает возможные значения выходных параметров, если выполнение задания не завершено после вызова SFC.

Замечание

После каждого вызова вы должны проанализировать в своей программе соответствующие выходные параметры.

Таблица 2-1. Связь между вызовом, REQ, RET_VAL и BUSY во время выполнения задания.

Номер вызова	Тип вызова	REQ	RET_VAL	BUSY
1	Первый вызов	1	W#16#7001	1
			Код ошибки	0
от 2 до (n - 1)	Промежуточный вызов	Не имеет значения	W#16#7002	1
n	Последний вызов	Не имеет значения	W#16#0000 (исключения: SFC59 "RD_REC", если целевая область больше, чем передаваемая запись данных, и SFC13 "DPNRM_DG," SFC67 "X_GET" и SFC72 "I_GET", если не было ошибок)	0
			Код ошибки, если были ошибки	0

3 Функции копирования и манипулирования блоками

3.1 Копирование переменных с помощью SFC20 "BLKMOV"

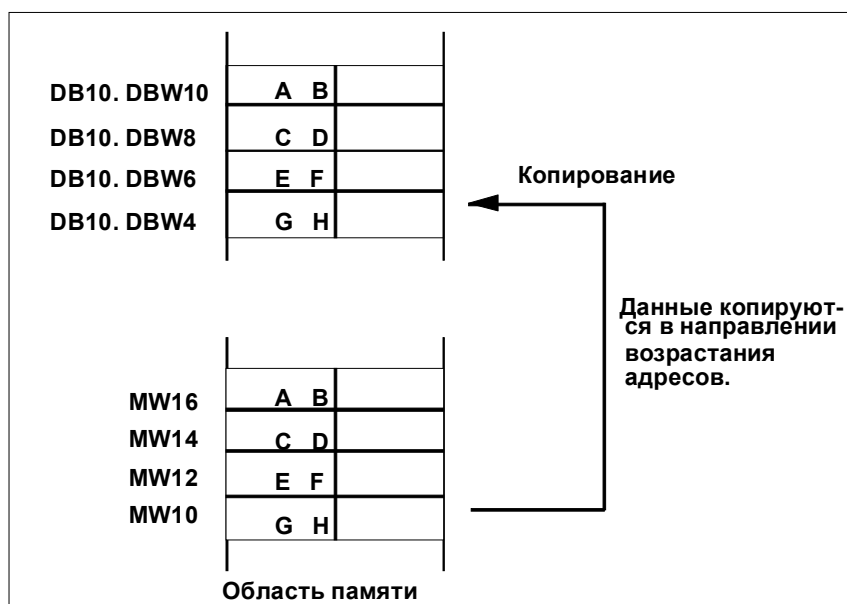
Описание

SFC20 "BLKMOV" (block move [перемещение блока]) используется для копирования содержимого области памяти (= исходной области) в другую область памяти (= целевую область).

С помощью SFC20 "BLKMOV" можно копировать любые области памяти, кроме:

- следующих типов блоков: FB, SFB, FC, SFC, OB, SDB,
- счетчиков,
- таймеров,
- областей памяти периферийных входов/выходов.

Параметр-источник может быть частью блока данных в загрузочной памяти, который не имеет значения для исполнения программы (DB, который компилировался с ключевым словом UNLINKED).



Возможность прерывания

Пока исходная область массив не является частью блока данных, который существует только в загрузочной памяти, нет ограничений на глубину вложения.

Если, однако, SFC20 прерывается при копировании из DB, не существенного для исполнения программы, то исполнение SFC20 более не может быть вложенным.

Параметр	Описание	Тип данных	Область памяти	Характеристика
SRCBLK	INPUT	ANY	I, Q, M, D, L	Указывает области памяти, которая должна копироваться (исходная область). Массивы, относящиеся к типу данных STRING, недопустимы.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время исполнения функции происходит ошибка, то возвращаемое значение содержит код ошибки.
DSTBLK	OUTPUT	ANY	I, Q, M, D, L	Указывает область памяти, в которую будут копироваться данные (целевая область). Массивы, относящиеся к типу данных STRING, недопустимы.

Замечание

Исходная и целевая области не должны перекрываться. Если заданная целевая область больше, чем исходная, то функция копирует лишь столько данных, сколько содержится в исходной области.

Если заданная целевая область меньше, чем исходная, то функция копирует лишь столько данных, сколько может быть записано в целевую область.

Если указатель ANY (источника или адресата) имеет тип данных BOOL, то заданная длина должна делиться на 8; в противном случае SFC не будет выполняться.

Если указатель ANY имеет тип данных STRING, то заданная длина должна быть равна 1.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Нет ошибок
8091	Превышена глубина вложения

3.2 Непрерываемое копирование переменных с помощью SFC81 "UBLKMOV"

Описание

С помощью SFC81 "UBLKMOV" (uninterruptable block move [непрерываемое перемещение блоков]) вы можете непротиворечивым образом копировать содержимое одной области памяти (=исходная область) в другую область памяти (= целевую область). Эта операция копирования не может прерываться другими действиями операционной системы.

Используя SFC81 "UBLKMOV", вы можете копировать все области памяти, кроме:

- следующих типов блоков: FB, SFB, FC, SFC, OB, SDB
- счетчиков
- таймеров
- областей памяти периферийных входов/выходов
- несвязанных блоков данных (скомпилированных с опцией UNLINKED)

Максимальная длина данных, которые вы можете копировать, равна 512 байтам.

Возможность прерывания, время реакция на прерывание

Копирование не может прерываться. Помните, что, если вы используете SFC81 "UBLKMOV", то это может увеличить время реакции вашего CPU на прерывание.

Параметр	Описание	Тип	Область памяти	Характеристика
SRCBLK	INPUT	ANY	I, Q, M, D, L	Определяет копируемую область памяти (исходную область). Массивы типа STRING недопустимы.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при выполнении функции происходит ошибка, то возвращаемое значение содержит код ошибки.
DSTBLK	OUTPUT	ANY	I, Q, M, D, L	Определяет область памяти, в которую будут копироваться данные (целевая область). Массивы типа STRING недопустимы.

Замечание

Исходная и целевая области не должны перекрываться. Если заданная целевая область больше, чем исходная область, то функция копирует в целевую область только такое количество данных, какое содержится в исходной области.

Если указанная целевая область меньше, чем исходная область, то функция копирует только такое количество данных, какое может быть записано в целевой области.

Если указатель ANY (источника или адресата) имеет тип BOOL, то заданная длина должна делиться на 8; иначе SFC не будет выполняться.

Если указатель ANY имеет тип STRING, то заданная длина должна быть равна 1.

Информация об ошибках

Код ошибки (W#16#...)	Характеристика
0000	Ошибки нет
8091	Исходная область находится в несвязанном блоке данных

3.3 Инициализация области памяти с помощью SFC21 "FILL"

Описание

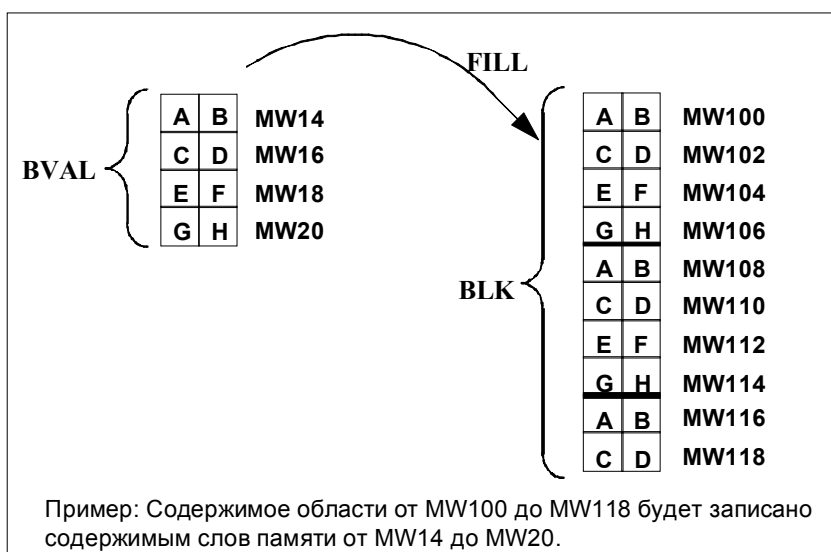
С помощью SFC 21 "FILL" Вы можете предварительно заполнить область памяти (целевую область) содержимым другой области памяти (исходной области). SFC копирует содержимое в заданную целевую область, пока эта область памяти не будет полностью записана.

Замечание

Исходный и целевой массив не должны перекрываться.

Если целевая область, подлежащая инициализации, не является целым кратным длины входного параметра BVAL, то она, тем не менее, записывается до последнего байта.

Если целевая область, подлежащая инициализации, меньше, чем исходная область, то копируется лишь столько данных, сколько может быть записано в целевую область.



Исключения

С помощью SFC 21 нельзя записывать значения в:

- следующие типы блоков: FB, SFB, FC, SFC, SDB,
- счетчики,
- таймеры,
- области памяти периферийных входов/выходов.

Параметр	Описание	Тип данных	Область памяти	Характеристика
BVAL	INPUT	ANY	I, Q, M, D, L	Параметр BVAL содержит значение или описание области, содержимое которой будет использовано для инициализации целевой области(исходная область). Массивы, относящиеся к типу данных STRING, недопустимы.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время обработки функции появляется ошибка, то возвращаемое значение содержит код ошибки.
BLK	OUTPUT	ANY	I, Q, M, D, L	Параметр BLK содержит описание области, подлежащей инициализации (целевой области). Массивы, относящиеся к типу данных STRING, недопустимы.

Входной параметр является структурой

Если в качестве входного параметра передается структура, то помните следующую особенность: STEP 7 всегда определяет длину структуры как четное число байтов. В результате структуре потребуется один байт дополнительного пространства в памяти, если вы опишете структуру с нечетным числом байтов.

Пример

Структура описана следующим образом:

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

Описанная структура "TYP_5_BYTE_STRUCTURE" требует 6 байтов памяти.

Информация об ошибках

Как анализируется информация об ошибках в параметре RET_VAL объяснено в главе 1. Эта глава содержит также общую информацию об ошибках SFC. SFC21 не выдает специфической информации об ошибках через параметр.

3.4 Создание блока данных с помощью SFC22 "CREAT_DB"

Описание

С помощью SFC22 "CREAT_DB" (create data block [создать блок данных]), вы создаете блок данных, который не содержит предварительно установленных значений. Этот SFC создает блок данных выбираемой длины с номером, взятым из заданного диапазона. SFC присваивает DB минимально возможный номер из заданного диапазона. Если вы хотите создать DB с конкретным номером, то просто выберите диапазон, указывая одно и то же значение для верхней и нижней границы. Вы не можете назначить номер, если DB с таким же номером уже существует в программе пользователя. Длина DB должна быть четным числом байтов.

Возможность прерывания

SFC22 "CREAT_DB" может прерываться более приоритетными OB. Если SFC22 "CREAT_DB" снова вызывается в OB более высокого приоритета, то этот вызов отвергается с кодом ошибки W#16#8091.

Параметр	Описание	Тип данных	Область памяти	Характеристика
LOW_LIMIT	INPUT	WORD	I, Q, M, D, L, константа	Нижнее граничное значение является минимальным номером в диапазоне номеров, которые вы можете присвоить своему блоку данных.
UP_LIMIT	INPUT	WORD	I, Q, M, D, L, константа	Верхнее граничное значение является максимальным номером в диапазоне номеров, которые вы можете присвоить своему блоку данных.
COUNT	INPUT	WORD	I, Q, M, D, L, константа	Числовое значение задает количество байтов данных, которое вы хотите зарезервировать для своего блока данных. Здесь вы должны задать четное число байтов (максимум 65534).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время обработки функции появляется ошибка, то возвращаемое значение содержит код ошибки.
DB_NUMBER	OUTPUT	WORD	I, Q, M, D, L	Номер блока данных является номером созданного блока данных. Если происходит ошибка (бит 15 в RET_VAL был установлен), то в DB_NUMBER вводится значение 0.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Нет ошибки.
8091	Вы выполнили вложенный вызов SFC22.
8092	Функция "Создать DB" не может быть в данный момент выполнена, так как: <ul style="list-style-type: none"> • в данный момент активна функция "Сжатие памяти пользователя" • в данный момент активна функция "Сохранение программы пользователя" • основной CPU выполняет функции соединения или обновления.
80A1	Ошибка в номере DB: <ul style="list-style-type: none"> • Номер равен 0. • Номер превышает количество DB для конкретного CPU. • Параметр lower limit [нижняя граница] > upper limit [верхняя граница].
80A2	Ошибка в длине DB: <ul style="list-style-type: none"> • Длина равна 0. • Длина была задана как нечетное число. • Длина больше, чем допускает CPU.
80B1	Нет свободных номеров DB.
80B2	Недостаточно свободной памяти.
80B3	Недостаточно непрерывной памяти (устранение: провести сжатие!)

3.5 Удаление блока данных с помощью SFC23 "DEL_DB"

Описание

С помощью SFC 23 "DEL_DB" (delete data block) вы удаляете блок данных, находящийся в рабочей памяти, и, если имеется, в загрузочной памяти CPU. Удаляемый DB не должен быть открыт ни в текущем, ни в каком-либо более низком классе приоритета, иными словами, он не должен быть внесен ни в один из двух регистров DB, ни в В-стек. В противном случае CPU при вызове SFC 23 переходит в состояние STOP.

Следующая таблица поясняет, когда DB может быть удален с помощью SFC23 "DEL-DB".

Если...	то...
DB был создан путем вызова SFC22 "CREAT_DB",	SFC23 может его удалить.
DB был передан в CPU посредством STEP 7 и был создан не с ключевым словом UNLINKED,	SFC23 может его удалить.
DB находится на флэш-карте,	SFC23 не может его удалить.

Возможность прерывания

SFC23 "DEL_DB" может быть прерван более высокими классами приоритета. Если этот SFC вызывается там снова, то этот второй вызов прерывается и в RET_VAL записывается W#16#8091.

Параметр	Описание	Тип данных	Область памяти	Характеристика
DB_NUMBER	INPUT	WORD	I, Q, M, D, L, константа	Номер удаляемого DB
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибке

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8091	Вызовы SFC23 были вложенными, и была превышена максимальная глубина вложения для используемого CPU.
8092	Функция "Стирание DB" в данный момент не может быть выполнена, так как <ul style="list-style-type: none"> в данный момент активна функция "Сжатие памяти пользователя" в данный момент активна функция "Сохранение программы пользователя". основной CPU выполняет функции соединения или обновления.
80A1	Ошибка во входном параметре DB_NUMBER: выбранный фактический параметр <ul style="list-style-type: none"> имеет значение 0 больше, чем максимально допустимый номер DB для используемого CPU.
80B1	DB с заданным номером в CPU не существует.
80B2	DB с заданным номером был создан с ключевым словом UNLINKED.
80B3	DB находится на флэш-карте.

3.6 Тестирование блока данных с помощью SFC24 "TEST_DB"

Описание

С помощью SFC24 "TEST_DB" (test data block [тестировать блок данных]) вы получаете информацию о блоке данных, находящемся в рабочей памяти CPU. SFC запрашивает сведения о количестве байтов данных в выбранном DB и проверяет, защищен ли DB от записи.

Параметр	Описание	Тип данных	Область памяти	Характеристика
DB_NUMBER	INPUT	WORD	I, Q, M, D, L, константа	Номер тестируемого DB
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках
DB_LENGTH	OUTPUT	WORD	I, Q, M, D, L	Количество байтов данных, которое содержит выбранный DB.
WRITE_PROT	OUTPUT	BOOL	I, Q, M, D, L	Информация о признаке защиты выбранного DB от записи (1 означает, что он защищен от записи).

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
80A1	Ошибка во входном параметре DB_NUMBER: выбранный фактический параметр <ul style="list-style-type: none"> имеет значение 0 больше, чем максимально допустимый номер DB для используемого CPU.
80B1	DB с заданным номером в CPU не существует.
80B2	DB был создан с ключевым словом UNLINKED.

3.7 Сжатие памяти пользователя с помощью SFC25 "COMPRESS"

Пропуски в памяти

Пропуски могут возникать как в рабочей, так и в загрузочной памяти вследствие многократного удаления и повторной загрузки блоков. Эти пропуски сокращают эффективную область памяти.

Описание

С помощью SFC 25 "COMPRESS" вы запускаете сжатие раздела ОЗУ как рабочей, так и загрузочной памяти. Функция сжатия здесь та же самая, что и при внешнем запуске в режиме RUN-P (положение переключателя режимов работы).

Если сжатие было запущено извне и все еще активно, то вызов SFC25 приведет к появлению сообщения об ошибке.

Замечание

Блоки данных, имеющие длину более 1000 байтов, не сдвигаются функцией SFC25 "COMPRESS". Это значит, что после сжатия в рабочей памяти могут остаться пропуски..

Параметр	Описание	Тип данных	Область памяти	Характеристика
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибке
BUSY	OUTPUT	BOOL	I, Q, M, D, L	Показывает, активна ли еще функция сжатия, запущенная вызовом SFC25. (1 означает, что активна)
DONE	OUTPUT	BOOL	I, Q, M, D, L	Показывает, успешно ли завершился процесс сжатия, который был запущен посредством SFC 25 (1 означает "успешно").

Проверка функции сжатия

При однократном вызове SFC 25 "COMPRESS" запускается функция сжатия; однако вы не можете проверить, успешно ли выполнилось сжатие.

Если вы хотите проверить функцию сжатия, выполните описанные ниже шаги:

Вызовите SFC25 циклически. После каждого вызова сначала оценивайте параметр RET_VAL. Если его значение равно 0, можно оценить параметры BUSY и DONE. Если BUSY = 1, а DONE = 0, то это указывает, что функция сжатия еще активна. Когда BUSY примет значение 0, а DONE - значение 1, то это указывает, что функция сжатия успешно завершена. Если потом снова вызвать SFC 25, то снова запустится сжатие.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было. Функция сжатия была запущена SFC 25. Только в этом случае имеет смысл оценивание в программе пользователя выходных параметров BUSY и DONE (см. выше).
8091	Функция сжатия была запущена извне и еще активна.
8092	Функция "Сжатие памяти пользователя" в данный момент не может быть выполнена, так как <ul style="list-style-type: none"> была запущена извне и все еще активна функция "Удаление блока данных" функция тестирования и запуска в настоящее время требует конкретного блока (например, статус) была запущена извне и все еще активна функция "Копирование блоков". основной CPU выполняет функции соединения или обновления.

3.8 Передача заменяющего значения в аккумулятор 1 с помощью SFC44 "REPL_VAL"

Описание

С помощью SFC44 "REPL_VAL" (replace value [заменить значение]) вы передаете значение в аккумулятор 1 класса приоритета, вызвавшего ошибку.

Ограничение: только в ОБ синхронных ошибок

Вы можете вызывать SFC 44 "REPL_VAL" только в ОБ синхронных ошибок (OB121, OB122).

Пример применения

Если модуль ввода поврежден до такой степени, что из него более нельзя считывать значения, то при каждом обращении к этому модулю запускается OB122. В OB122 с помощью SFC44 "REPL_VAL" в аккумулятор 1 прерванного класса приоритета может быть передано подходящее значение, так что программа может продолжать работу с этим заменяющим значением. Информация, нужная для выбора заменяющего значения (например, блок, в котором встретилась ошибка, или поврежденный операнд), находится локальных переменных OB122.

Параметр	Описание	Тип данных	Область памяти	Характеристика
VAL	INPUT	DWORD	I, Q, M, D, L, константа	Заменяющее значение
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время обработки функции встретилась ошибка, то возвращаемое значение содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#....)	Описание
0000	Ошибок не было. Было введено заменяющее значение.
8080	SFC44 не вызывался ни одним из ОБ синхронных ошибок (OB121, OB122).

4 SFC для управления исполнением программы

1.1 Повторный запуск контроля времен и цикла с помощью SFC43 "RE_TRIGR"

Описание

С помощью SFC43 "RE_TRIGR" (retrigger watchdog [перезапустить контрольный таймер]) вы можете повторно запустить контроль времени цикла.

Параметры

SFC43 "RE_TRIGR" не имеет параметров.

Информация об ошибках

SFC43 "RE_TRIGR" не предоставляет информации об ошибках.

4.2 Перевод CPU в STOP с помощью SF C46 "STP"

Описание

С помощью SFC 46 "STP" (stop) вы переводите CPU в состояние STOP.

Параметры

SFC46 "STP" не имеет параметров.

Информация об ошибках

SFC46 "STP" не предоставляет информации об ошибках.

4.3 Задержка исполнения программы пользователя с помощью SFC47 "WAIT"

Описание

С помощью SFC 47 "WAIT" вы программируете задержки или времена ожидания в своей пользовательской программе. Вы можете запрограммировать времена ожидания величиной до 32767 мкс. Наименьшее возможное время ожидания зависит от конкретного CPU и равно времени исполнения SFC 47.

Возможность прерывания

SFC47 "WAIT" может прерываться ОВ более высокого приоритета.

Замечание

(только для S7-300, но не для CPU 318)

Время ожидания, запрограммированное с помощью SFC 47, - это минимальное время. Оно продлевается за счет времени исполнения вложенных классов приоритета и загрузки системы.

Параметр	Описание	Тип данных	Область памяти	Характеристика
WT	INPUT	INT	I, Q, M, D, L, константа	Параметр WT содержит время ожидания в мкс.

Информация об ошибках

SFC47 "WAIT" не предоставляет информации об ошибках.

4.4 Запуск прерывания многопроцессорной обработки с помощью SFC35 "MP_ALM"

Описание

Вызов SFC 35 "MP_ALM" при многопроцессорной обработке запускает прерывание многопроцессорной обработки. Это приводит к синхронизированному запуску OB 60 во всех соответствующих CPU. В однопроцессорном режиме и при работе с сегментированной стойкой OB 60 запускается только в тех CPU, которые вызвали SFC 35.

С помощью входного параметра JOB вы можете отобразить причину прерывания многопроцессорной обработки. Этот идентификатор задания передается на все задействованные CPU и может использоваться вами в OB 60 (см. раздел 1.7 и [/234/](#)).

Вы можете вызывать SFC 35 "MP_ALM" в любом месте своей программы. Однако поскольку вызов имеет смысл только в режиме RUN, то при вызове в режиме STARTUP прерывание многопроцессорной обработки подавляется. Об этом вам сообщается через значение функции.

Параметр	Описание	Тип данных	Область памяти	Характеристика
JOB	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор задания: возможные значения: от 1 до 15
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при исполнении функции возникает ошибка, то возвращаемое значение содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8090	Входной параметр JOB содержит недопустимое значение.
80A0	Еще не завершено исполнение OB60 вслед за последним прерыванием многопроцессорной обработки на локальном или ином CPU.
80A1	Неправильный рабочий режим (STARTUP [запуск] вместо RUN).

5 SFC для манипулирования системными часами

5.1 Установка времени с помощью SFC 0 "SET_CLK"

Описание

С помощью SFC0 "SET_CLK" (set system clock [установить системные часы]) вы устанавливаете время суток и дату в часах CPU. Вызов SFC0 запускает часы. Затем часы ведут отсчет от установленных значений времени и даты. Если часы являются ведущими, то CPU при вызове SFC 0 также начинает синхронизацию времени. Интервалы синхронизации устанавливаются с помощью STEP 7.

Параметр	Описание	Тип данных	Область памяти	Характеристика
PDT	INPUT	DT	D, L	На входе PDT вводятся дата и время суток, которые вы хотите установить.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время обработки функции появляется ошибка, то возвращаемое значение содержит код ошибки.

Дата и время

Дата и время суток вводятся как тип данных DT. Пример: для 15-го января 1995, 10 час. 30 мин. и 30 секунд вам следует ввести: DT#1995-01-15-10:30:30. Время может быть введено только с точностью до секунд. День недели рассчитывается функцией SFC0 "SET_CLK" исходя из даты.

Помните, что тип данных DT вначале должен быть образован с помощью FC 3 "D_TOD_DT", прежде чем вы сможете передать ему входные параметры (см. раздел 21.4).

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Нет ошибок
8080	Ошибка в дате
8081	Ошибка во времени

5.2 Считывание времени с помощью SFC1 "READ_CLK"

Описание

С помощью SFC 1 "READ_CLK" (read system clock [читать системные часы]) считываются текущая дата и время с системных часов CPU.

Параметр	Описание	Тип данных	Область памяти	Характеристика
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при обработке функции появляется ошибка, то возвращаемое значение содержит код ошибки.
CDT	OUTPUT	DT	D,L	На выходе CDT выводятся текущая дата и текущее время суток.

Информация об ошибках

Как оценить информацию об ошибках, содержащуюся в параметре RET_VAL, объяснено в главе 1. Эта глава содержит также общую информацию об ошибках SFC. SFC1 не выводит специфическую информацию об ошибках.

5.3 Синхронизация ведомых часов с помощью SFC48 "SNC_RTCB"

Определение: Синхронизация ведомых часов

Под синхронизацией ведомых часов понимают передачу даты и времени суток от ведущих часов некоторого сегмента шины (например, К- шины S7-400, MPI, задней шины S7) на все ведомые часы этого сегмента шины.

Описание

С помощью SFC 48 "SNC_RTCB" (synchronize real time clocks [синхронизация часов реального времени]) вы синхронизируете все имеющиеся на сегменте шины ведомые часы. Предпосылкой успешной синхронизации является вызов SFC 48 в CPU, часы реального времени которого были назначены ведущими часами по крайней мере для одного сегмента шины. Соответствующие параметры назначаются с помощью STEP 7.

Синхронизация ведомых часов со стороны системы (циклически по истечении выбранного интервала времени) происходит независимо от вызовов SFC 48.

Параметр	Описание	Тип данных	Область памяти	Характеристика
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при исполнении функции происходит ошибка, то возвращаемое значение содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	При синхронизации ошибок не было.
0001	Имеющиеся часы не были назначены ведущими ни для одного сегмента шины.

6 SFC для манипулирования счетчиками рабочего времени

6.1 Счетчики рабочего времени

Введение

CPU имеют в своем распоряжении несколько счетчиков рабочего времени (см. технические данные своего CPU). С помощью SFC 2, 3 и 4 вы можете устанавливать, останавливать и считывать счетчики рабочего времени.

Применение

Вы можете использовать счетчик рабочего времени для самых разных приложений:

- для измерения продолжительности работы CPU
- для измерения продолжительности работы управляемой аппаратуры или подключенных устройств.

Характеристики счетчика рабочего времени

При запуске счетчик рабочего времени начинает считать, начиная с последнего зарегистрированного. Если вы хотите, чтобы он начал с другого начального значения, то вы должны установить это значение с помощью SFC2. Если CPU переходит в состояние STOP или вы останавливаете счетчик рабочего времени, то CPU регистрирует текущее значение счетчика. Когда выполняется теплый или холодный рестарт CPU, нужно снова запустить счетчик рабочего времени с помощью SFC3.

Диапазон значений

Каждый счетчик рабочего времени имеет диапазон значений от 0 до 32 767 часов.

6.2 Установка счетчика рабочего времени с помощью SFC2 "SET_RTM"

Описание

С помощью SFC 2 "SET_RTM" (set run-time meter [установить счетчик рабочего времени]) счетчик рабочего времени CPU устанавливается на выбранное значение. Количество счетчиков рабочего времени, которое вы можете установить, зависит от используемого вами CPU.

Параметр	Описание	Тип данных	Область памяти	Характеристика
NR	INPUT	BYTE	I, Q, M, D, L, константа	Вход NR содержит номер счетчика рабочего времени, который вы хотите установить (возможные значения: от 0 до 7).
PV	INPUT	INT	I, Q, M, D, L, константа	Вход PV содержит установку для счетчика рабочего времени (по умолчанию).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при исполнении функции происходит ошибка, то возвращаемое значение содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Нет ошибок
8080	Неправильный номер счетчика рабочего времени
8081	Параметру PV было передано отрицательное значение.

6.3 Запуск и останов счетчика рабочего времени с помощью SFC3 "CTRL_RTM"

Описание

С помощью SFC 3 "CTRL_RTM" (control run-time meter [управление счетчиком рабочего времени]) вы можете запускать или останавливать счетчик рабочего времени CPU.

Параметр	Описание	Тип данных	Область памяти	Характеристика
NR	INPUT	BYTE	I, Q, M, D, L, константа	Вход NR содержит номер счетчика рабочего времени, который вы хотите запустить или остановить (возможные значения: от 0 до 7).
S	INPUT	BOOL	I, Q, M, D, L, константа	Вход S запускает или останавливает счетчик рабочего времени. Установите состояние сигнала на "0", когда вы хотите остановить счетчик. Установите состояние сигнала на "1", когда вы хотите запустить счетчик.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время обработки функции происходит ошибка, то возвращаемое значение содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Нет ошибок
8080	Неправильный номер счетчика рабочего времени

6.4 Считывание счетчика рабочего времени с помощью SFC4 "READ_RTM"

Описание

С помощью SFC 4 "READ_RTM" (read run-time meter [прочитать счетчик рабочего времени]) вы считываете счетчик рабочего времени. SFC 4 в качестве выходных данных выдает текущее время счета и состояние счетчика, т.е. "остановлен" или "считает".

Если проходит более 32767 часов, то счетчик останавливается на значении 32767 и выводит сообщение об ошибке "overflow [переполнение]".

Параметр	Описание	Тип данных	Область памяти	Характеристика
NR	INPUT	BYTE	I, Q, M, D, L, константа	Вход NR содержит номер счетчика рабочего времени, который вы хотите прочитать (возможные значения: от 0 до 7).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время обработки функции происходит ошибка, то возвращаемое значение содержит код ошибки.
CQ	OUTPUT	BOOL	I, Q, M, D, L	Выход CQ указывает, работает счетчик рабочего времени или остановлен. Состояние сигнала "0" указывает, что счетчик рабочего времени остановлен. Состояние сигнала "1" указывает, что счетчик рабочего времени работает.
CV	OUTPUT	INT	I, Q, M, D, L	Выход CV указывает текущее значение счетчика рабочего времени.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Нет ошибки
8080	Неправильный номер счетчика рабочего времени
8081	Переполнение счетчика рабочего времени

6.5 Считывание системного времени с помощью SFC64 "TIME_TCK"

Описание

С помощью SFC 64 "TIME_TCK" (time tick [такт времени]) вы можете прочитать системное время CPU. Системное время - это "счетчик времени", который считает циклически от 0 до максимума в 2147483647 мс. В случае переполнения системного времени счет ведется опять от нуля. Разрешение и точность системного времени составляют 1 мс для S7-400 и CPU 318 и 10 мс для всех остальных CPU S7-300. Системное время находится под воздействием только рабочих режимов CPU.

Применение

Вы можете использовать системное время, например, для того, чтобы измерять длительность процессов путем сравнения результатов двух вызовов SFC 64.

Системное время и режимы работы

Режим работы	Системное время...
Startup (Запуск)	... обновляется постоянно
RUN	
STOP	... останавливается и сохраняет текущее значение
Горячий рестарт (нет в S7-300 и S7-400 H)	... продолжается от значения, сохраненного при переходе в STOP
Теплый рестарт	... стирается и начинается опять с "0"
Холодный рестарт	

Параметр	Описание	Тип данных	Область памяти	Характеристика
RET_VAL	OUTPUT	TIME	I, Q, M, D, L	Параметр RET_VAL содержит считанное системное время в диапазоне от 0 до $2^{31}-1$ мс.

Информация об ошибках

SFC64 "TIME_TCK" не предоставляет информации об ошибках.

7 SFC для передачи записей данных

7.1 Запись и чтение записей данных

Принцип

Некоторые модули имеют предназначенную только для записи область системных данных, в которую ваша программа может передавать записи данных. Эта область содержит записи данных с номерами от 0 до максимум 240. Не каждый модуль содержит все записи данных (см. следующую таблицу).

Другие модули имеют область системных данных, предназначенную только для чтения, в которой ваша программа может читать записи данных. Эта область содержит записи данных с номерами от 0 до максимум 240. Не каждый модуль содержит все записи данных (см. следующую таблицу).

Замечание

Есть модули, имеющие обе области системных данных.

Это физически отдельные области, у которых общей является только их логическая структура.

Область системных данных, предназначенная только для записи

Следующая таблица показывает структуру области системных данных, предназначенной только для записи. Эта таблица показывает также, какими длинными могут быть отдельные записи данных и с помощью каких SFC они могут записываться.

Номер записи данных	Содержимое	Размер	Ограничение	Может записываться с помощью SFC
0	Параметры	У S7-300: от 2 до 14 байтов	может записываться только контроллером S7-400	56 "WR_DPARM" 57 "PARM_MOD"
1	Параметры	У S7-300: от 2 до 14 байтов Записи данных 0 и 1 вместе имеют ровно 16 байтов.	-	55 "WR_PARM" 56 "WR_DPARM" 57 "PARM_MOD"
от 2 до 127	Данные пользователя	не более, чем по 240 байтов каждая	-	55 "WR_PARM" 56 "WR_DPARM" 57 "PARM_MOD" 58 "WR_REC"
от 128 до 240	Параметры	не более, чем по 240 байтов каждая	-	55 "WR_PARM" 56 "WR_DPARM" 57 "PARM_MOD" 58 "WR_REC"

Область системных данных, предназначенных только для чтения

Следующая таблица показывает структуру области системных данных, предназначенных только для чтения. Эта таблица показывает также, какими длинными могут быть отдельные записи данных и с помощью каких SFC они могут читаться.

Номер записи данных	Содержимое	Размер	Может читаться с помощью SFC
0	Специфические для модулей диагностические данные (устанавливаются, как правило, для всей системы)	4 байта	51 "RDSYSST" (SSL_ID 00B1H) 59 "RD_REC"
1	Специфические для канала диагностические данные (включая запись данных 0)	<ul style="list-style-type: none"> У S7-300: 16 байтов У S7-400: от 7 до 220 байтов 	51 "RDSYSST" (SSL_ID 00B2H и 00B3H) 59 "RD_REC"
от 2 до 127	Данные пользователя	не более, чем по 240 байтов каждая	59 "RD_REC"
от 128 до 240	Диагностические данные	не более, чем по 240 байтов каждая	59 "RD_REC"

Системные ресурсы

Если вы через короткие промежутки времени запускаете подряд несколько асинхронно протекающих процессов передачи записей данных, то распределение системных ресурсов операционной системой гарантирует, что все задания выполнятся и не будут влиять друг на друга.

Если используются все доступные системные ресурсы, то это отображается в RET_VAL. Вы можете устранить эту нерегулярную ошибочную ситуацию простым повторением задания.

Максимальное количество "одновременно" активных заданий SFC зависит от CPU. За более подробной информацией обратитесь к [/70/](#) и [/101/](#).

7.2 Чтение определенных параметров с помощью SFC54 "RD_DPARM"

Описание

С помощью SFC54 "RD_DPARM" (read defined parameter [читать определенный параметр]) вы считываете запись данных адресуемого модуля с номером RECNUM из соответствующего SDB1ху. Считываемая запись данных вводится в целевую область, открытую с помощью параметра RECORD.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор (ID) адресной области: В#16#54 =Периферийный вход (PI) В#16#55 = Периферийный выход (PQ) Если модуль смешанный, то задайте ID области с наименьшим адресом. Если адреса одинаковые, то задайте В#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический базовый адрес модуля. В случае смешанных модулей задайте меньший из двух адресов.
RECNUM	INPUT	BYTE	I, Q, M, D, L, константа	Номер записи данных (разрешенные значения: от 0 до 240)
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если в то время, когда функция активна, происходит ошибка, то возвращаемое значение содержит код ошибки. А также: длина считываемой записи данных в байтах, если считываемая запись данных помещается в область назначения, и при передаче не было ошибок.
RECORD	OUTPUT	ANY	I, Q, M, D, L	Область назначения для считываемой записи данных. Разрешается только тип данных BYTE.

Информация об ошибках

См. Таблица 7-1 Специфическая информация об ошибках для SFC54 "RD_DPARM", SFC55 "WR_PARM", SFC56 "WR_DPARM" и SFC57 "PARM_MOD"

7.3 Запись динамических параметров с помощью SFC55 "WR_PARM"

Описание

С помощью SFC55 "WR_PARM" (write parameter [записать параметр]) вы передаете запись данных RECORD адресуемому модулю. Параметры, которые передаются модулю, **не** заменяют параметры этого модуля в соответствующем SDB, если они там существуют.

Предпосылки

Передаваемая запись данных не должна быть статической:

- Это не должна быть запись данных 0 (запись данных 0 является статической во всей системе).
- Если происходит обращение к записи данных в SDB 100 – 129, то бит статики не должен быть установлен.

За дополнительной информацией о статических записях данных обратитесь к [/71/](#) и [/101/](#).

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	REQ = 1: запрос на запись
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор адресной области: В#16#54 = периферийный вход (PI) В#16#55 = периферийный выход (PQ) Если модуль смешанный, то укажите идентификатор области самого младшего адреса. Если адреса одинаковы, укажите В#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический базовый адрес модуля. У смешанных модулей указывайте младший из двух адресов.
RECNUM	INPUT	BYTE	I, Q, M, D, L, константа	Номер записи данных
RECORD	INPUT	ANY	I, Q, M, D, L	Запись данных
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при исполнении функции появляется ошибка, то возвращаемое значение содержит код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: Процесс записи еще не закончен.

Входной параметр RECORD

Передаваемые данные читаются из параметра RECORD при первом вызове SFC. Если передача записи данных длится дольше, чем вызов, то содержимое параметра RECORD больше не является релевантным для последующих вызовов SFC (для того же самого задания).

Информация об ошибках

Таблица 7-1 Специфическая информация об ошибках для SFC54
"RD_DPARM", SFC55 "WR_PARM", SFC56 "WR_DPARM" и SFC57
"PARM_MOD"

Замечание (только для S7-400)

Если происходит общая ошибка W#16#8544, то это показывает только, что
был заблокирован доступ по крайней мере к одному байту области памяти
входов/выходов, содержащей запись данных. Передача данных была
продолжена

7.4 Запись параметров, установленных по умолчанию, с помощью SFC56 "WR_DPARM"

Описание

С помощью SFC56 "WR_DPARM" (write default parameter [запись параметров, установленных по умолчанию]) вы передаете запись данных с номером RECNUM из соответствующего SDB1ху адресуемому модулю. Для этой функции не имеет значения, является ли набор данных статическим или динамическим.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	REQ = 1: запрос на запись
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор адресной области: B#16#54 = периферийный вход (PI) B#16#55 = периферийный выход (PQ) Если модуль смешанный, то укажите идентификатор области самого младшего адреса. Если адреса одинаковы, укажите B#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический базовый адрес модуля. У смешанных модулей указывайте младший из двух адресов.
RECNUM	INPUT	BYTE	I, Q, M, D, L, константа	Номер записи данных
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при исполнении функции происходит ошибка, то возвращаемое значение содержит код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: процесс записи еще не закончен.

Информация об ошибках

Таблица 7-1 Специфическая информация об ошибках для SFC54 "RD_DPARM", SFC55 "WR_PARM", SFC56 "WR_DPARM" и SFC57 "PARM_MOD"

7.5 Назначение параметров модулю с помощью SFC57 "PARM_MOD"

Описание

С помощью SFC57 "PARM_MOD" (parametrize module [параметризовать модуль]) вы передаете модулю все его записи данных, которые вы спроектировали с помощью STEP 7 в соответствующем SDB. Для этой функции не имеет значения, являются ли эти записи данных статическими или динамическими.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	REQ = 1: запрос на запись
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор адресной области: B#16#54 = периферийный вход (PI) B#16#55 = периферийный выход (PQ) Если модуль смешанный, то укажите идентификатор области самого младшего адреса. Если адреса одинаковы, укажите B#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический базовый адрес модуля. У смешанных модулей указывайте младший из двух адресов.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если ошибка возникает, когда функция активна, то возвращаемое значение содержит код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: процесс записи еще не закончен.

Информация об ошибках

"Реальная" информация об ошибках (коды ошибок W#16#8xyz) (см. Таблица 7-1 Специфическая информация об ошибках для SFC54 "RD_DPARM", SFC55 "WR_PARM", SFC56 "WR_DPARM" и SFC57 "PARM_MOD") может быть разделена на два класса:

- Нерегулярные ошибки (коды ошибок W#16#80A2 – 80A4, 80Cx):
Для этого типа ошибки возможно, что она будет устранена без вашего вмешательства, иными словами, рекомендуется вызвать этот SFC снова (если необходимо, более одного раза).
Примером нерегулярной ошибки является ситуация, когда требуемые ресурсы в данный момент времени используются (W#16#80C3).
- Постоянные ошибки (коды ошибок W#16#809x, 80A1, 80Bx, 80Dx):
Этот тип ошибки не может быть устранен без вашего вмешательства. Повторный вызов SFC будет успешным только после устранения ошибки. Примером постоянной ошибки является ввод неверной длины в параметр RECORD (W#16#80B1).

Таблица 7-1 Специфическая информация об ошибках для SFC54 "RD_DPARM", SFC55 "WR_PARM", SFC56 "WR_DPARM" и SFC57 "PARM_MOD".

Код ошибки (W#16#...)	Объяснение	Ограничение
7000	Первый вызов с REQ=0: нет активных процессов передачи данных; BUSY имеет значение 0.	-
7001	Первый вызов с REQ=1: запущен процесс передачи данных; BUSY имеет значение 1.	Децентрализованная периферия
7002	Промежуточный вызов (REQ не имеет значения): процесс передачи данных активен; BUSY имеет значение 1.	Децентрализованная периферия
8090	Заданный логический базовый адрес недействителен: Нет соответствия в SDB1/SDB2х или нет базового адреса.	-
8092	В ссылке ANY задан тип, отличный от BYTE.	Только в S7-400 для SFC54 "RD_PARM" и SFC55 "WR_PARM"
8093	Этот SFC недопустим для модуля, заданного в LADDR и IOID (разрешены следующие модули: модули S7-300 для S7-300, модули S7-400 для S7-400, модули S7-DP для S7-300 и S7-400).	-
80A1	Отрицательная квитанция при передаче записи данных модулю (во время передачи модуль был вытаскен или стал неисправным).	1)
80A2	Ошибка протокола DP на уровне 2, возможно, неисправность аппаратуры/интерфейса в slave-устройстве DP	Децентрализованная периферия ¹⁾
80A3	Ошибка протокола DP на уровне интерфейс пользователя/пользователь	Децентрализованная периферия ¹⁾
80A4	Ошибка связи на коммуникационной шине.	Ошибка происходит между CPU и внешним интерфейсным модулем DP ¹⁾
80B0	SFC невозможна для этого типа модуля, модуль не распознает эту запись данных.	1)
80B1	Неправильная длина передаваемой записи данных. У SFC54 "RD_PARM": длина целевой области, открытой параметром RECORD, слишком коротка.	-
80B2	Спроектированный слот не занят.	1)
80B3	Фактический тип модуля не совпадает с требуемым типом модуля в SDB1.	1)
80C1	Данные предшествующего задания на запись в модуле для той же самой записи данных еще не обработаны модулем.	1)
80C2	В данный момент времени модуль обрабатывает максимально возможное для CPU количество заданий.	1)
80C3	В данный момент времени требуемые ресурсы (память и т.д.) заняты.	1)

Код ошибки (W#16#...)	Объяснение	Ограничение
80C4	Ошибки связи: <ul style="list-style-type: none"> ошибка четности бит готовности программного обеспечения не установлен ошибка в информации о длине области ошибка контрольной суммы на стороне CPU ошибка контрольной суммы на стороне модуля 	1)
80C5	Децентрализованная периферия недоступна.	Децентрализованная периферия ¹⁾
80C6	Передача записи данных была остановлена из-за прерывания выполнения класса приоритета (горячий рестарт или фоновый режим).	Децентрализованная периферия ¹⁾
80D0	В соответствующем SDB нет записи для модуля.	-
80D1	Номер записи данных для модуля не спроектирован в соответствующем SDB. (Номера записей данных • 241 отклоняются STEP 7).	-
80D2	В соответствии с идентификатором типа модулю не могут быть назначены параметры.	-
80D3	SDB не может быть назначен, так как он не существует.	-
80D4	Ошибка структуры SDB: Внутренний указатель SDB указывает на значение за пределами SDB.	Только у S7-300
80D5	Запись данных является статической.	Только у SFC55 "WR_PARM"

1) Не бывает в SFC54 "RD_DPARM"

7.6 Внесение записи данных с помощью SFC58 "WR_REC"

Описание

С помощью SFC58 "WR_REC" (write record [внести запись]) вы передаете запись данных, содержащуюся в RECORD, адресуемому модулю. Вы запускаете задание на запись, присваивая входному параметру REQ значение 1 при вызове SFC58. Если задание на запись смогло выполниться немедленно, то SFC возвращает в выходном параметре BUSY значение 0. Если BUSY имеет значение 1, то задание на запись еще не завершено.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	REQ = 1: запрос на запись
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор адресной области: B#16#54 = периферийный вход (PI) B#16#55 = периферийный выход (PQ) Если модуль смешанный, то укажите идентификатор области самого младшего адреса. Если адреса одинаковы, укажите B#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический адрес модуля. У смешанных модулей указывайте младший из двух адресов.
RECNUM	INPUT	BYTE	I, Q, M, D, L, константа	Номер записи данных (допустимые значения: от 2 до 240)
RECORD	INPUT	ANY	I, Q, M, D, L	Запись данных. Допустимым является только тип данных BYTE.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если ошибка появляется, когда функция активна, то возвращаемое значение содержит код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: процесс записи еще не закончен.

Входной параметр RECORD

Передаваемые данные читаются из параметра RECORD при первом вызове SFC. Если передача записи данных длится дольше, чем вызов, то содержимое параметра RECORD больше не является релевантным для последующих вызовов SFC (для того же самого задания).

Информация об ошибках

Таблица 7-2 Специфическая информация об ошибках для SFC58 "WR_REC" и SFC59 "RD_REC"

Замечание (только для S7-400)

Если происходит общая ошибка W#16#8544, то это показывает только то, что был заблокирован доступ по крайней мере к одному байту области памяти входов/выходов, содержащей запись данных. Передача данных была продолжена.

7.7 Чтение записи данных с помощью SFC59 "RD_REC"

Описание

С помощью SFC59 "RD_REC" (read record [читать запись]) вы считываете запись данных с номером RECNUM из адресуемого модуля. Задание на чтение запускают, вызывая SFC59 и присваивая входному параметру REQ значение 1. Если задание на чтение может быть выполнено немедленно, то SFC возвращает в выходном параметре BUSY значение 0. Если BUSY имеет значение 1, то задание на чтение еще не завершилось (см. раздел 2.2). Считываемая запись данных вводится в область назначения, указанную параметром RECORD, если передача данных была свободна от ошибок.

Замечание

Если вы считываете запись данных с номером выше 1 из FM или CP, купленного до февраля 1997 (называемого ниже "старым модулем"), то реакция SFC59 отличается от реакции в случае нового модуля. Эта особая ситуация описана в разделе "Использование старых FM и CP S7-300 с номерами записей данных выше 1".

Замечание

Нижеследующее относится к CPU S7-400 более низкой версии, чем показано в таблице ниже: Если область назначения меньше, чем считываемая запись данных, то в RET_VAL вводится W#16#80B1, а область назначения остается неизменной. Если длина области назначения и длина считываемой записи данных одинаковы, то в RET_VAL вместо значения 0 (нет ошибки) вводится длина записи данных в виде положительного значения.

CPU	Заказной номер	Версия (или выше)
CPU 412-1	6ES7412-1XF01-0AB0	03
CPU 413-1	6ES7413-1XG01-0AB0	03
CPU 413-2DP	6ES7413-2XG01-0AB0	03
CPU 414-1	6ES7414-1XG01-0AB0	03
CPU 414-2DP	6ES7414-2XG01-0AB0	03
CPU 414-2DP	6ES7414-2XJ00-0AB0	03
CPU 416-1	6ES7416-1XJ01-0AB0	03
CPU 416-2DP	6ES7416-2XK00-0AB0	03
CPU 416-2DP	6ES7416-2XL00-0AB0	03

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	REQ = 1: Запрос на чтение
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор (ID) адресной области: W#16#54 = Периферийный вход (PI)

				В#16#55 = Периферийный выход (PQ) Если модуль смешанный, то задайте ID области с наименьшим адресом. Если адреса одинаковые, то задайте В#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический адрес модуля. В случае смешанных модулей задайте меньший из двух адресов.
RECNUM	INPUT	BYTE	I, Q, M, D, L, константа	Номер записи данных (разрешенные значения от 0 до 240)
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если в то время, когда функция активна, происходит ошибка, то возвращаемое значение содержит код ошибки. Если область назначения больше, чем переданная запись данных, и при передаче не было ошибок, то вводится также длина фактически переданной записи данных в байтах (возможные значения: от +1 до +240).
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: Чтение еще не закончилось.
RECORD	OUTPUT	ANY	I, Q, M, D, L	Область назначения для считываемой записи данных. В случае асинхронного выполнения SFC59 убедитесь, что фактические параметры RECORD имеют одинаковую информацию о длине во всех вызовах. Разрешен только тип данных BYTE.

Выходной параметр RET_VAL

- Если во время выполнения функции произошла ошибка, то возвращаемое значение содержит код ошибки.
- Если ошибок не было, то RET_VAL содержит:
 - 0: если вся область назначения была заполнена данными из выбранной записи данных (запись данных может быть также неполной).
 - длину фактически переданной записи данных в байтах (возможные значения: от +1 до + 240), если область назначения больше, чем переданная запись данных.

Замечание

(только для S7-400)

Если происходит общая ошибка W#16#8545, то это указывает только на то, что был заблокирован доступ, по крайней мере, к одному байту области памяти входов/выходов, содержащей запись данных. Запись данных была правильно считана модулем и записана в область памяти входов/выходов.

Установка RECORD

Замечание

Если вы хотите гарантировать, чтобы всегда читалась полная запись данных, то выберите область назначения длиной 241 байт. Если передача данных свободна от ошибок, то RET_VAL содержит фактическую длину записи данных.

Использование старых FM и CP S7-300 с номерами записей данных выше 1

Если вы хотите считывать с помощью SFC59 "RD_REC" запись данных с номером выше 1 из старого FM S7-300 или старого CP S7-300, то помните о следующем:

- Если область назначения больше, чем фактическая длина требуемой записи данных, то никакие данные не вводятся в RECORD. RET_VAL имеет значение W#16#80B1.
- Если область назначения меньше, чем фактическая длина требуемой записи данных, то CPU считывает столько байтов, начиная от начала записи, сколько определено в информации о длине RECORD, и вводит это количество байтов в RECORD. RET_VAL имеет значение 0.
- Если длина, заданная в RECORD, равна фактической длине требуемой записи данных, то CPU считывает запись данных и вводит ее в RECORD. RET_VAL имеет значение 0.

Информация об ошибках

"Реальные" данные об ошибках (коды ошибок W#16#8xyz) в следующей таблице могут быть разделены на два класса:

- Нерегулярные ошибки (коды ошибок от W#16#80A2 до 80A4, 80Cх):
При этом типе ошибки возможно, что ошибка будет устранена без принятия вами каких-либо мер, другими словами, целесообразно снова вызвать SFC (в случае необходимости более одного раза).
Примером нерегулярной ошибки является случай, когда требуемые ресурсы в настоящее время используются (W#16#80C3).
- Постоянные ошибки (коды ошибки W#16#809х, 80A1, 80Bх, 80Dх):
Ошибка этого типа не будет устранена без принятия вами мер. Повторный вызов SFC будет успешным только после того, как ошибка устранена.
Примером постоянной ошибки является ввод неправильной длины в RECORD (W#16#80B1).

Таблица 7-2 Специфическая информация об ошибках для SFC58 "WR_REC" и SFC59 "RD_REC".

Код ошибки (W#16#...)	Объяснение	Ограничение
7000	Первый вызов с REQ=0: Передача данных не активна; BUSY имеет значение 0.	-
7001	Первый вызов с REQ=1: Передача данных не активна; BUSY имеет значение 1.	Децентрализованная периферия
7002	Промежуточный вызов (REQ не имеет значения): Передача данных уже активна; BUSY имеет значение 1.	Децентрализованная периферия
8090	Задан недействительный логический базовый адрес: Нет назначения в SDB1/SDB2X или нет базового адреса.	-
8092	Тип, указанный в ссылке ANY, не BYTE.	Только S7-400
8093	Этот SFC не разрешен для модуля, определяемого LADDR и IOID (разрешены следующие модули: модули S7-300 для S7-300, модули S7-400 для S7-400, модули S7-DP для S7-300 и S7-400).	-
80A0	Отрицательное подтверждение при чтении из модуля (модуль был снят во время задания на чтение или неисправен).	Только в случае SFC59 "RD_REC"
80A1	Отрицательное подтверждение при передаче записи данных в модуль (модуль был снят во время передачи или неисправен).	Только в случае SFC58 "WR_REC"
80A2	Ошибка протокола DP на уровне 2, возможно, неисправность аппаратуры.	Децентрализованная периферия
80A3	Ошибка протокола DP в случае преобразователя данных с прямым каналом передачи данных или пользовательского интерфейса/пользователя. Возможно, неисправность аппаратуры.	Децентрализованная периферия
80A4	Ошибка связи в коммуникационной шине	Ошибка происходит между CPU и внешним интерфейсным модулем DP.
80B0	<ul style="list-style-type: none"> SFC недопустим для типа модуля. Модуль не распознает запись данных. Номер записи данных ≥ 241 не разрешен. В случае SFC58 (WR_REC) записи данных 0 и 1 не разрешены. 	-

Код ошибки (W#16#...)	Объяснение	Ограничение
80B1	В параметре RECORD задана неправильная длина.	<ul style="list-style-type: none"> Для SFC58 "WR_REC": неправильная длина Для SFC59 "RD_REC" (только при использовании старых FM S7-300 и CP S7-300): длина > длины записи Для SFC13 "DPNRM_DG": длина < длины записи
80B2	Сконфигурированный слот не занят.	-
80B3	Фактический тип модуля не совпадает с типом модуля, требуемым в SDB1.	-
80C0	В случае SFC59 (RD_REC): Модуль имеет запись данных, но все еще нет данных, которые нужно считывать. В случае SFC13 (DPNRM_DG): Нет доступных диагностических данных.	Только в случае SFC59 (RD_REC) или для SFC13 "DPNRM_DG"
80C1	Данные предыдущего задания на запись для той же самой записи данных в модуле еще не были обработаны модулем.	-
80C2	Модуль в настоящее время обрабатывает максимально возможное для CPU количество заданий.	-
80C3	Требуемые ресурсы (память и т.д.) в настоящее время заняты.	-
80C4	Ошибка связи: <ul style="list-style-type: none"> Ошибка четности Не установлено "Software ready [Программное обеспечение готово]" Ошибка в информации о длине поля Ошибка контрольной суммы на стороне CPU Ошибка контрольной суммы на стороне модуля 	-
80C5	Децентрализованная периферия недоступна.	Децентрализованная периферия
80C6	Передача записи данных была остановлена вследствие прерывания класса приоритета (горячий рестарт или фоновый режим).	Децентрализованная периферия

7.8 Чтение записи данных с помощью SFC59 "RD_REC" в CPU S7-300

Применимость

Следующее описание SFC59 "RD_REC" применимо к CPU, перечисленным ниже:

CPU	Заказной номер
CPU 312 IFM	6ES7312-5AC00-0AB0
CPU 313	6ES7313-1AD00-0AB0
CPU 314	6ES7314-1AE01-0AB0
CPU 314 IFM	6ES7314-5AE00-0AB0
CPU 315	6ES7315-1AF00-0AB0
CPU 315-2DP	6ES7315-2AF00-0AB0
CPU 614	6ES7614-1AH00-0AB3

Описание

С помощью SFC59 "RD_REC" (read data record [читать запись данных]) вы считываете запись данных с номером RECNUM из адресуемого модуля. Считываемая запись данных вводится в область назначения, указанную параметром RECORD, если передача данных была свободна от ошибок.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	REQ = 1: Запрос на чтение.
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор (ID) адресной области: B#16#54 = Периферийный вход (PI) B#16#55 = Периферийный выход (PQ) Если модуль смешанный, то задайте ID области с наименьшим адресом. Если адреса одинаковые, то задайте B#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический адрес модуля. В случае смешанных модулей задайте меньший из двух адресов.
RECNUM	INPUT	BYTE	I, Q, M, D, L, константа	Номер записи данных (разрешенные значения от 0 до 240).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время выполнения функции происходит ошибка, то возвращаемое значение содержит код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: Чтение еще не закончилось.
RECORD	OUTPUT	ANY	I, Q, M, D, L	Область назначения для считываемой записи данных. В случае асинхронного выполнения SFC59 убедитесь, что фактические параметры RECORD имеют одинаковую информацию о длине во всех вызовах. Разрешен только тип данных BYTE.

RECORD

Информация о длине в выходном параметре RECORD интерпретируется следующим образом:

Длина данных, которые нужно считывать из выбранной записи данных. Это означает, что информация о длине в RECORD должна быть не больше, чем фактическая длина записи данных.

Целесообразно выбирать длину для RECORD точно такую же, как фактическая длина записи данных.

Принцип передачи данных

В случае задания на чтение CPU информирует адресованный модуль о длине параметра RECORD. Следующие пункты зависят от того, принадлежит ли модуль станции DP или нет:

- Модуль находится в центральной стойке или стойке расширения.

Если длина, заданная посредством RECORD, короче фактической длины требуемой записи данных, то CPU считывает столько байтов от начала записи данных, сколько задано в информации о длине RECORD, и вводит их в RECORD. RET_VAL имеет значение 0. Если длина в RECORD больше фактической длины требуемой записи данных, то CPU вводит в RET_VAL код ошибки.

Если информация о длине в RECORD такая же, как фактическая длина требуемой записи данных, то CPU считывает требуемую запись данных и вводит ее в RECORD. В RET_VAL вводится значение 0.

- Модуль находится в slave-устройстве DP S7.

Коммуникационный процессор slave-устройства DP S7 оценивает информацию о длине, полученную от CPU.

Если длина в RECORD меньше длины требуемой записи данных, то slave-устройство DP S7-300 возвращает в CPU требуемую часть выбранной записи данных. Если длина в RECORD больше длины требуемой записи данных, то slave-устройство DP S7-300 возвращает в CPU информацию об ошибке.

CPU оценивает ошибку или информацию о длине, принятую от slave-устройства DP S7:

- Если slave-устройство DP S7 предоставляет данные об ошибке, то соответствующий код ошибки вводится в RET_VAL.
- Если slave-устройство DP S7 возвращает длину прочитанных данных, то эта длина сравнивается с информацией о длине в RECORD. В зависимости от результата сравнения в выходных параметрах RET_VAL и RECORD делается запись. (Реакция такая же, как в случае, когда модуль расположен в центральной стойке или стойке расширения.)

Замечание

В случае асинхронной обработки SFC59 убедитесь, что фактические параметры RECORD имеют одинаковую информацию о длине во всех вызовах.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение	Ограничение
7000	Первый вызов с REQ=0: Передача данных не активна; BUSY имеет значение 0.	-
7001	Первый вызов с REQ=1: Передача данных не активна; BUSY имеет значение 1.	Децентрализованная периферия
7002	Промежуточный вызов (REQ не имеет значения): Передача данных уже активна; BUSY имеет значение 1.	Децентрализованная периферия
8090	Задан недействительный логический базовый адрес: Нет назначения в SDB1/SDB2x или нет базового адреса.	-
8093	Этот SFC не разрешен для модуля, определяемого LADDR и IOID (разрешены следующие модули: модули S7-300 и модули S7-300 DP).	-
80A0	Отрицательное подтверждение при чтении из модуля (модуль был снят во время задания на чтение чтения или неисправен).	-
80A2	Ошибка протокола DP на уровне 2	Децентрализованная периферия
80A3	Ошибка протокола DP в случае пользовательского интерфейса/пользователя	Децентрализованная периферия
80A4	Ошибка связи в коммуникационной шине	Ошибка происходит между CPU и внешним интерфейсным модулем DP.
80B0	SFC недопустима для типа модуля. Модуль не распознает запись данных. Номер записи данных ≥ 241 не разрешен.	-
80B1	В параметре RECORD задана неправильная длина	Длина > длины записи
80B2	Сконфигурированный слот не занят.	-
80B3	Фактический тип модуля не соответствует типу модуля, требуемому в SDB1.	-
80C0	Модуль имеет запись данных, но все еще нет данных, которые нужно считывать.	
80C2	Модуль в настоящее время обрабатывает максимально возможное для CPU количество заданий.	-
80C3	Требуемые ресурсы (память и т.д.) в настоящее время заняты.	-
80C4	Ошибка связи: <ul style="list-style-type: none"> • Ошибка четности • Не установлено "Software ready [Программное обеспечение готово]" • Ошибка в информации о длине поля • Ошибка контрольной суммы на стороне CPU • Ошибка контрольной суммы на стороне модуля 	-
80C5	Децентрализованная периферия недоступна.	Децентрализованная периферия
80C6	Передача записи данных была остановлена вследствие прерывания класса приоритета (горячий рестарт или фоновый режим).	Децентрализованная периферия

7.9 Дополнительная информация об ошибках SFC 55–59

Только для S7-400

У S7-400 SFC 55–59 могут возвращать также информацию об ошибке W#16#80Fx. Это значит, что произошла ошибка, которую невозможно было локализовать. В этом случае обращайтесь в отдел обслуживания.

8 SFC для манипулирования прерываниями по времени

8.1 Манипулирование прерываниями по времени

Определение

Прерывание по времени имеет своим результатом вызовы OB прерываний по времени (OB10 – OB17).

Условия для вызова

Прежде чем операционной системой может быть вызван OB прерываний по времени, должны быть выполнены следующие условия:

- OB прерываний по времени должен быть параметрирован (дата и время запуска, выполнение) с помощью
 - STEP 7 или
 - SFC28 "SET_TINT" в программе пользователя.
- OB прерываний по времени должен быть активизирован с помощью
 - STEP 7 или
 - SFC30 "ACT_TINT" в программе пользователя.
- Выбор OB прерываний по времени не должен быть отменен с помощью STEP 7.
- OB прерываний по времени должен существовать в CPU.
- Если вы устанавливаете прерывание по времени с помощью SFC30 "ACT_TINT" и если вы задали исполнение этого OB как **однократное**, то дата и время запуска еще не должны пройти. Если вы выбрали **периодическое** выполнение, то OB прерываний по времени будет вызван, когда будет завершен следующий период (момент времени запуска + кратное длительности периода).

Совет

Вы можете назначать параметры прерыванию по времени с помощью STEP 7, а затем активизировать прерывание в своей пользовательской программе (SFC30 "ACT_TINT").

Назначение SFC28 – SFC31

Системные функции SFC28 – SFC31, описанные в следующих разделах, используются следующим образом:

- для установки прерываний по времени (SFC28 "SET_TINT")
- для отмены прерываний по времени (SFC29 "CAN_TINT")
- для активизации прерываний по времени (SFC30 "ACT_TINT")
- для опроса прерываний по времени (SFC31 "QRY_TINT")

8.2 Характеристики SFC 28 – 31

Что произойдет, если...

Следующая таблица перечисляет ряд различных ситуаций и объясняет, какое воздействие они оказывают на прерывание по времени.

Если ...	то ...
прерывание по времени устанавливается (вызовом SFC28; SET_TINT)	текущее прерывание по времени автоматически отменяется.
прерывание по времени отменяется (вызовом SFC29; CAN_TINT)	дата и время запуска стираются. Затем прерывание по времени должно быть сначала снова установлено, прежде чем оно может быть активизировано.
ОВ прерываний по времени не существует в момент вызова	автоматически генерируется ошибка класса приоритета; т.е. операционная система вызывает ОВ 85. Если ОВ 85 не существует, то CPU переходит в состояние STOP.
синхронизируются часы реального времени или корректируется время <ul style="list-style-type: none"> Часы переводятся вперед 	Если дата/время запуска пропущены из-за перевода часов вперед: <ul style="list-style-type: none"> Операционная система вызывает ОВ80¹. Вслед за ОВ80 вызываются все пропущенные ОВ прерываний по времени (один раз, независимо от количества пропущенных периодов), если они не подвергались обработке в ОВ80². Если ОВ 80 не существует, то CPU переходит в состояние STOP. Если ОВ прерывания по времени уже вызывались на интервале переведенного назад времени, то при новом прохождении этого времени они еще раз не вызываются.
<ul style="list-style-type: none"> Часы переводятся назад 	

1) В информации о событии запуска ОВ 80 кодируется то, какие ОВ прерывания по времени не могут вызываться из-за установки времени вперед. Время в информации о событии запуска соответствует установленному вперед времени.

2) Время в информации о событии запуска прерывания по времени, активизируемого позднее, после того как оно было пропущено, соответствует времени запуска первого пропущенного прерывания по времени.

Теплый рестарт и холодный рестарт

Во время теплого или холодного рестарта все настройки прерываний по времени, сделанные в программе пользователя с помощью SFC, стираются.

После этого действуют установленные посредством STEP 7 параметры из блока параметров "time-of-day interrupts [прерывания по времени]".

Выполнение ОБ прерываний по времени

Следующая таблица показывает различные воздействия параметра "execution [выполнение]". Этот параметр устанавливается с помощью STEP 7 или с помощью SFC 28 "SET_TINT" (входной параметр PERIOD).

Выполнение ОБ прерываний по времени	Реакция
Нет (можно устанавливать только с помощью STEP 7)	ОБ прерываний по времени не выполняется даже тогда, когда он существует в CPU. Параметры могут быть заново установлены в программе пользователя с помощью SFC 28 "SET_TINT" (установить прерывание по времени).
Однократно	Прерывание по времени после вызова ОБ прерываний по времени отменяется. Затем оно может быть снова установлено и активизировано.
Периодически (ежеминутно, ежечасно, ежедневно, еженедельно, ежемесячно, ежегодно)	Если дата и время запуска к моменту активизации уже прошли, то ОБ прерываний по времени прерывает циклическую программу в момент времени "дата/время запуска + кратное установленной длительности периода". В очень редких случаях ОБ прерываний по времени может при следующем вызове оказаться еще в состоянии обработки. Результат: <ul style="list-style-type: none"> ошибка времени (операционная система вызывает ОБ 80; если ОБ 80 не существует, то CPU переходит в состояние STOP). ОБ прерываний по времени выполняется позднее.

8.3 Установка прерывания по времени с помощью SFC28 "SET_TINT"

Описание

С помощью SFC 28 "SET_TINT" (set time-of-day interrupt [установить прерывание по времени]) устанавливаются дата и время запуска организационных блоков прерываний по времени. Секунды и миллисекунды в заданном времени запуска игнорируются и устанавливаются в 0.

Параметр	Описание	Тип данных	Область памяти	Характеристика
OB_NR	INPUT	INT	I, Q, M, D, L, константа	Номер ОБ, который запускается в момент времени SDT + кратное PERIOD (OB10 – OB17).
SDT	INPUT	DT	D, L, константа	Дата и время запуска: секунды и миллисекунды в заданном времени запуска игнорируются и устанавливаются в 0.
PERIOD	INPUT	WORD	I, Q, M, D, L, константа	Периоды от исходной точки SDT: W#16#0000 = однократно W#16#0201 = ежеминутно W#16#0401 = ежечасно W#16#1001 = ежедневно W#16#1202 = еженедельно W#16#1401 = ежемесячно W#16#1801 = ежегодно W#16#2001 = в конце месяца
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при выполнении функции происходит ошибка, то фактический параметр RET_VAL содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Ошибки не было
8090	Ошибочный параметр OB_NR
8091	Ошибочный параметр SDT
8092	Ошибочный параметр PERIOD
80A1	Установленный момент запуска лежит в прошлом.

8.4 Отмена прерывания по времени с помощью SFC29 "CAN_TINT"

Описание

С помощью SFC 29 "CAN_TINT" (cancel time-of-day interrupt [отменить прерывание по времени]) можно отменить активизированный организационный блок прерываний по времени.

Параметр	Описание	Тип данных	Область памяти	Характеристика
OB_NR	INPUT	INT	I, Q, M, D, L, константа	Номер OB, в котором будут отменены дата и время запуска (OB10 – OB17).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при выполнении функции происходит ошибка, то фактический параметр RET_VAL содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Ошибки не было.
8090	Ошибочный параметр OB_NR
80A0	Не установлены дата и время запуска для соответствующего OB прерываний по времени

8.5 Активизация прерывания по времени с помощью SFC30 "ACT_TINT"

Описание

С помощью SFC 30 "ACT_TINT" (activate time-of-day interrupt [активизировать прерывание по времени]) вы можете активизировать организационный блок прерываний по времени.

Параметр	Описание	Тип данных	Область памяти	Характеристика
OB_NR	INPUT	INT	I, Q, M, D, L, константа	Номер ОБ, который должен активизироваться (OB10 – OB17).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при выполнении функции происходит ошибка, то фактический параметр RET_VAL содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Ошибки не было.
8090	Ошибочный параметр OB_NR.
80A0	Не установлены дата и время запуска для ОБ прерываний по времени.
80A1	Активизированное время лежит в прошлом; ошибка имеет место только в случае, когда выбрано однократное выполнение (execution=once).

8.6 Опрос прерывания по времени с помощью SFC31 "QRY_TINT"

Описание

С помощью системной функции SFC 31 "QRY_TINT" (query time-of-day interrupt [опросить прерывание по времени]) можно отобразить состояние организационного блока прерываний по времени в выходном параметре STATUS.

Параметр	Описание	Тип данных	Область памяти	Характеристика
OB_NR	INPUT	INT	I, Q, M, D, L, константа	Номер OB, состояние которого опрашивается (OB10 – OB17).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при выполнении функции происходит ошибка, то фактический параметр RET_VAL содержит код ошибки.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Состояние прерывания по времени; см. следующую таблицу.

Выходной параметр STATUS

Бит	Значение	Смысл
0	0	Прерывание по времени разблокировано операционной системой.
1	0	Новые прерывания по времени принимаются.
2	0	Прерывание по времени не активизировано или прошло.
3	-	-
4	0	ОВ прерываний по времени не загружен.
5	0	Выполнение ОВ прерываний по времени заблокировано работающей функцией тестирования.

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Ошибки не было.
8090	Ошибочный параметр OB_NR

9 SFC для манипулирования прерываниями с задержкой

9.1 Манипулирование прерываниями с задержкой

Определение

После того как вы вызвали SFC 32 "SRT_DINT", операционная система по истечении заданного времени задержки генерирует прерывание, иными словами, вызывается выбранный OB прерываний с задержкой. Это прерывание известно как прерывание с задержкой.

Условия для вызова

Чтобы прерывание с задержкой могло быть вызвано операционной системой, должны быть выполнены следующие условия:

- OB прерываний с задержкой должен быть запущен посредством SFC32 "SRT_DINT."
- Выбор OB прерываний с задержкой не должен быть отменен с помощью STEP 7.
- OB прерываний с задержкой должен существовать в CPU.

Назначение SFC32 – SFC34

Системные функции SFC32 – SFC34, описанные в следующих разделах, используются следующим образом:

- для запуска прерываний с задержкой (SFC32 "SRT_DINT")
- для отмены прерываний с задержкой (SFC33 "CAN_DINT")
- для опроса прерываний с задержкой (SFC34 "QRY_DINT").

Что произойдет, если...

Следующая таблица перечисляет ряд ситуаций и описывает их влияние на прерывание с задержкой.

Если ...	и ...	то ...
запускается прерывание с задержкой (вызовом SFC32 "SRT_DINT").	уже запущено прерывание с задержкой,	время задержки заменяется; прерывание с задержкой запускается снова.
	ОВ прерываний с задержкой не существует к моменту вызова,	операционная система генерирует ошибку класса приоритета (вызывает OB85). Если OB 85 не существует, то CPU переходит в состояние STOP.
	запуск прерывания произошло в ОВ запуска и время задержки истекло прежде, чем CPU оказался в RUN,	вызов ОВ прерывания с задержкой задерживается до тех пор, пока CPU не перейдет в RUN.
время задержки истекло	ранее запущенный ОВ прерываний с задержкой еще выполняется,	операционная система генерирует ошибку времени (вызывает OB80. Если OB 80 не существует, то CPU переходит в состояние STOP.

Теплый рестарт и холодный рестарт

При теплом и холодном рестарте все настройки прерывания с задержкой, сделанные в программе пользователя с помощью SFC, стираются.

Старт в ОВ запуска

Прерывание с задержкой может стартовать в ОВ запуска. Для вызова ОВ прерываний с задержкой должны быть выполнены два условия:

- Истекло время задержки.
- CPU находится в режиме RUN.

Если время задержки истекло, а CPU еще не находится в режиме RUN, то вызов ОВ прерываний с задержкой откладывается до тех пор, пока CPU не перейдет в состояние RUN. Тогда ОВ прерываний с задержкой вызывается еще до первой команды в OB1.

9.2 Запуск прерывания с задержкой с помощью SFC32 "SRT_DINT"

Описание

С помощью SFC 32 "SRT_DINT" (start time–delay interrupt [запустить прерывание с задержкой]) вы запускаете прерывание с задержкой, которое по истечении времени задержки (параметр DTIME) вызывает ОВ прерываний с задержкой.

С помощью параметра SIGN вы можете ввести идентификатор для распознавания запуска прерывания с задержкой. Значения DTIME и SIGN снова появляются в информации о событии запуска указанного ОВ, когда он выполняется.

Параметр	Описание	Тип данных	Область памяти	Характеристика
OB_NR	INPUT	INT	I, Q, M, D, L, константа	Номер ОВ для запуска с задержкой времени (OB20 – OB23).
DTIME	INPUT	TIME	I, Q, M, D, L, константа	Длительность задержки (от 1 до 60000 мс)
SIGN	INPUT	WORD	I, Q, M, D, L, константа	Идентификатор, который вводится в информацию о событии запуска ОВ при вызове ОВ прерываний с задержкой.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при выполнении функции происходит ошибка, то фактический параметр RET_VAL содержит код ошибки.

Точность

Время между вызовом SFC 32 "SRT_DINT" и запуском ОВ прерываний с задержкой максимум на **одну миллисекунду** меньше, чем выбранное время, если только никакие события прерывания не задерживают вызов.

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Ошибки не было.
8090	Ошибочный параметр OB_NR
8091	Ошибочный параметр DTIME

9.3 Опрос состояния прерывания с задержкой с помощью SFC34 "QRY_DINT"

Описание

С помощью SFC 34 "QRY_DINT" (query time–delay interrupt [опросить прерывание с задержкой]) вы можете опрашивать состояние прерывания с задержкой. Прерывания с задержкой управляются организационными блоками OB20 – OB23.

Параметр	Описание	Тип данных	Область памяти	Характеристика
OB_NR	INPUT	INT	I, Q, M, D, L, константа	Номер OB, состояние которого будет опрошено (OB20 – OB23).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при выполнении функции происходит ошибка, то фактический параметр RET_VAL содержит код ошибки.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Состояние прерывания с задержкой, см. следующую таблицу

Выходной параметр STATUS

Бит	Значение	Смысл
0	0	Прерывание с задержкой разблокировано операционной системой.
1	0	Новые прерывания с задержкой не отвергнуты.
2	0	Прерывание с задержкой не активизировано или выполнено.
3	-	-
4	0	ОВ прерываний с задержкой не загружен.
5	0	Выполнение ОВ прерываний с задержкой заблокировано работающей функцией тестирования.

Информация об ошибках

Код ошибки (W#16#..)	Описание
0000	Ошибки не было
8090	Ошибочный параметр OB_NR

9.4 Отмена прерывания с задержкой с помощью SFC33 "CAN_DINT"

Описание

С помощью SFC 33 "CAN_DINT" (cancel time–delay interrupt [отменить прерывание с задержкой]) вы можете отменить уже запущенное прерывание с задержкой (см. раздел 9.2 "Запуск прерывания с задержкой с помощью SFC32 "SRT_DINT"). Тогда ОВ прерываний с задержкой не вызывается.

Параметр	Описание	Тип данных	Область памяти	Характеристика
OB_NR	INPUT	INT	I, Q, M, D, L, константа	Номер ОВ, который должен быть отменен (OB20 – OB23).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при выполнении функции происходит ошибка, то фактический параметр RET_VAL содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Ошибки не было.
8090	Ошибочный параметр OB_NR
80A0	Прерывание с задержкой не запущено.

10 SFC для обработки синхронных ошибок

10.1 Маскирование синхронных ошибок

Введение

Синхронные ошибки – это ошибки программирования и доступа. Такие ошибки происходят в результате программирования с неправильными адресными областями, номерами или неправильными адресами.

Маскирование этих синхронных ошибок означает следующее:

- Маскированные синхронные ошибки не запускают ОВ ошибок и не приводят к запрограммированной альтернативной реакции.
- CPU "регистрирует" произошедшие маскированные ошибки в регистре ошибок.

Демаскирование ошибок означает отмену предварительно установленной маски и очистку соответствующего бита в регистре состояний событий текущего класса приоритета. Маскирование отменяется следующим образом:

- посредством вызова SFC37 "DMSK_FLT"
- когда завершится текущий класс приоритета.

Если ошибка происходит после того, как она была демаскирована, то операционная система запускает соответствующий ОВ ошибок. Вы можете запрограммировать ОВ121 для реагирования на ошибки программирования и ОВ122 для реагирования на ошибки доступа.

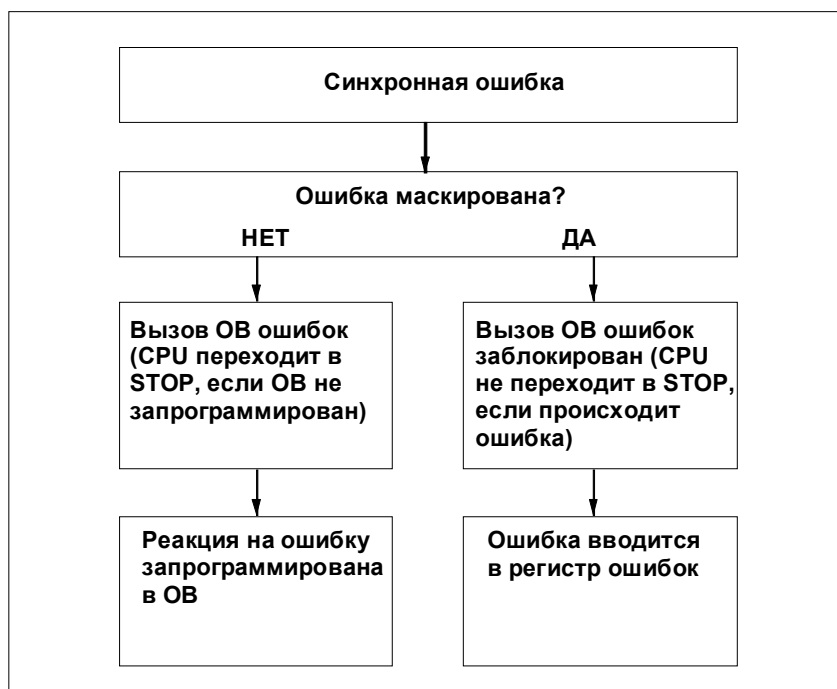
Вы можете использовать SFC38 "READ_ERR", чтобы считывать произошедшие замаскированные ошибки.

Обратите внимание: В S7-300 (кроме CPU 318), независимо от того, замаскирована ошибка или демаскирована, она вводится в диагностический буфер, и светится светодиод групповой ошибки CPU.

Обработка ошибок в общем случае

Если в программе пользователя происходят ошибки программирования и доступа, то вы можете реагировать на них разными способами:

- Вы можете запрограммировать ОВ ошибок, который вызывается операционной системой, когда происходит соответствующая ошибка.
- Вы можете заблокировать вызов ОВ ошибок индивидуально для каждого класса приоритета. В этом случае, когда ошибка данного типа происходит в конкретном классе приоритета, CPU не переходит в STOP. CPU вводит ошибку в регистр ошибок. Однако из этой записи вы не можете узнать, когда или как часто происходила ошибка.



Фильтры

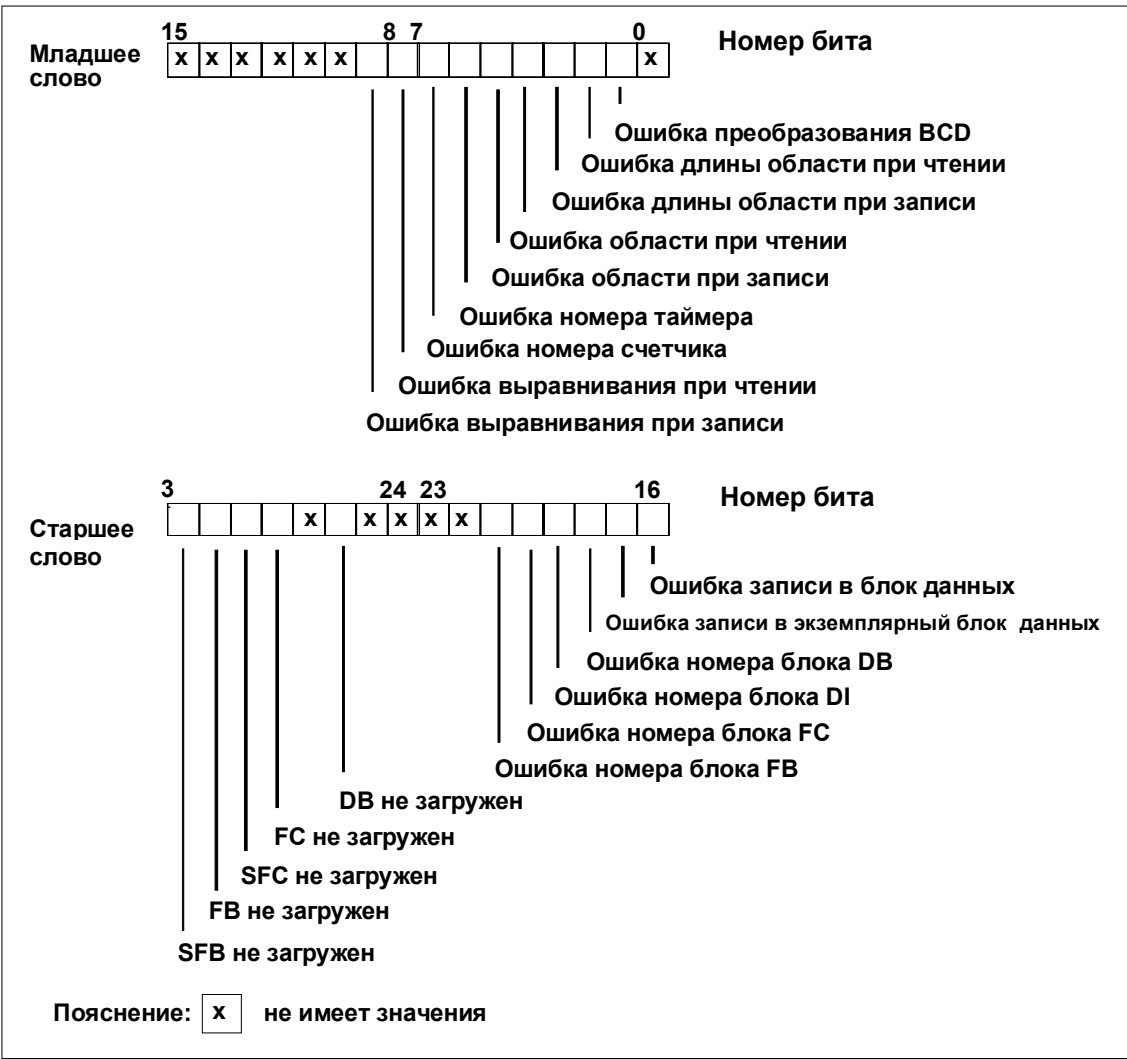
Синхронные ошибки ставятся в соответствие конкретной битовой комбинации, известной как **фильтр ошибок**. Этот фильтр ошибок находится также во входных и выходных параметрах SFC 36, 37 и 38.

Синхронные ошибки делятся на ошибки программирования и ошибки доступа, которые вы можете маскировать, используя два фильтра ошибок. Фильтры ошибок иллюстрируются на следующих рисунках.

Фильтр ошибок программирования

Следующий рисунок показывает битовую комбинацию фильтра ошибок программирования. Фильтр ошибок программирования расположен в параметрах PRGFLT_...

См. также Таблица 7-1 и Таблица 7-2



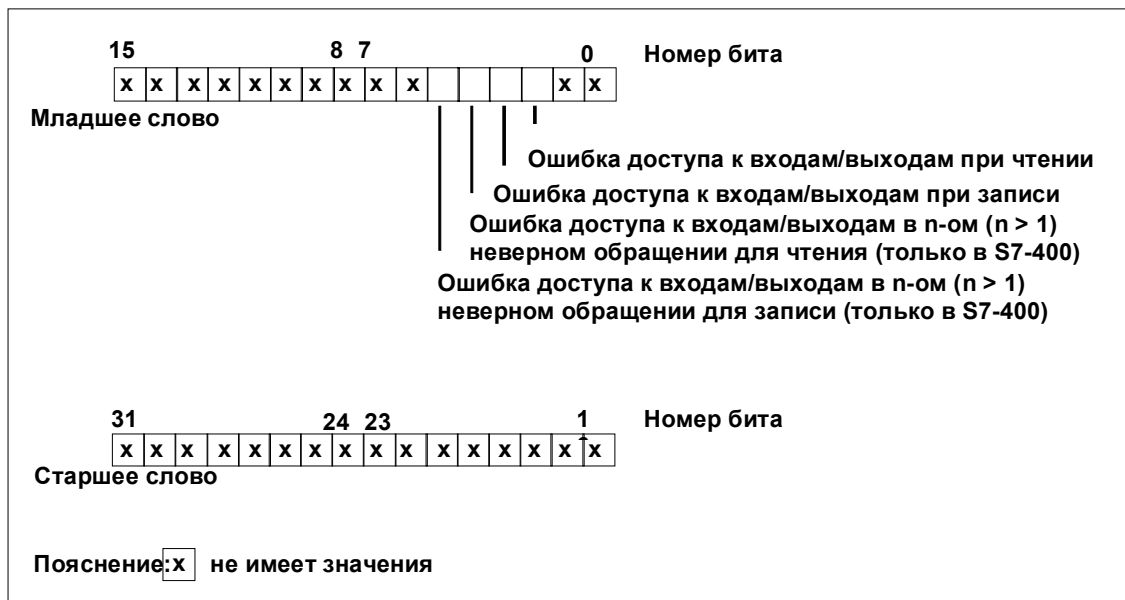
Несущественные биты

На вышеприведенном рисунке **x** означает...

- ... входные параметры для SFC36, 37, 38 = "0"
- ... выходные параметры для SFC36, 37 = "1" для S7-300
= "0" для S7-400
- для SFC38 = "0"

Фильтр ошибок доступа

Следующий рисунок показывает битовую комбинацию фильтра ошибок доступа для всех CPU, кроме CPU417 и CPU 417H. Фильтр ошибок доступа находится в параметрах ACCFLT_ ... За объяснением ошибок доступа обратитесь к таблице "Возможные причины ошибок для всех CPU, кроме CPU 417 и CPU 417H".



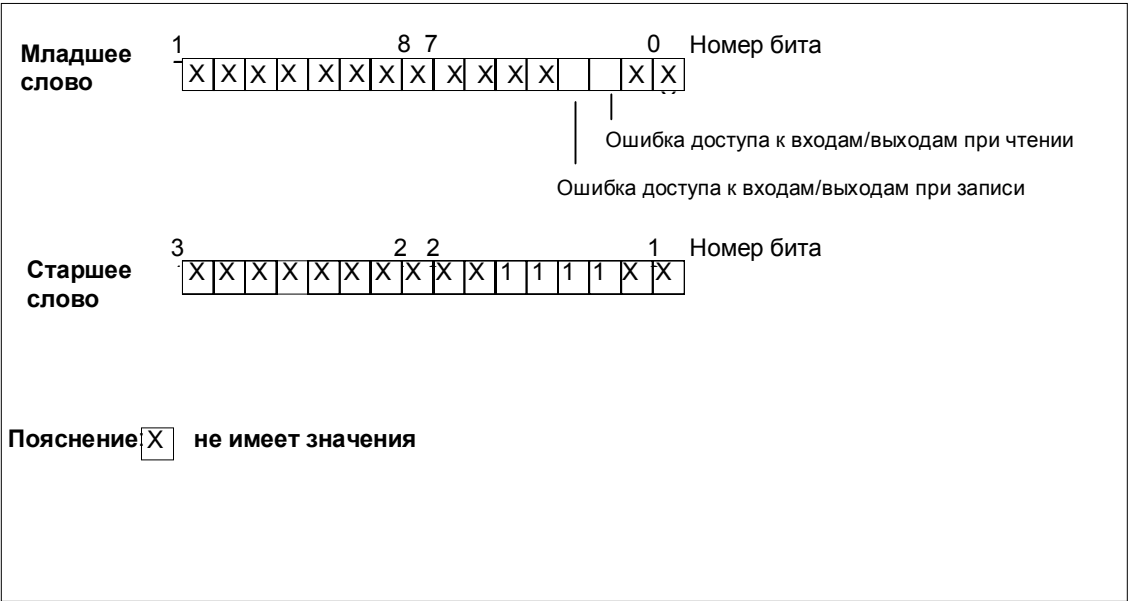
Несущественные биты

На вышеприведенном рисунке **x** означает...

- ... входные параметры для SFC36, 37, 38 = "0"
- ... выходные параметры для SFC36, 37 = "1" для S7-300
= "0" для S7-400
- для SFC38 = "0"

Фильтр ошибок доступа для CPU 417 и CPU 417H

Следующий рисунок показывает битовую комбинацию фильтра ошибок доступа в CPU417 и CPU 417H. Фильтр ошибок доступа находится в параметрах ACCFLT_... За объяснением ошибок доступа обратитесь к таблице Возможные причины ошибок для всех CPU, кроме CPU 417 и CPU 417H".



Несущественные биты

На вышеприведенном рисунке x означает...

- ... входные параметры для SFC36, 37, 38 = "0"
- ... выходные параметры для SFC36, 37 = "0"
- для SFC38 = "0"

Пример

Следующий рисунок показывает младшее слово фильтра ошибок доступа со всеми маскированными ошибками для всех CPU, кроме CPU 417 и CPU 417H:

- в качестве входного параметра для SFC36
- в качестве выходного параметра для SFC36



Пример для CPU 417 и CPU 417H

Следующая схема показывает, как выглядит младшее слово фильтра ошибок доступа со всеми маскированными ошибками для CPU 417 и CPU 417H:

- в качестве входного параметра для SFC36
- в качестве выходного параметра для SFC36.



Младшее слово ошибок программирования

Следующая таблица перечисляет ошибки, поставленные в соответствие младшему слову фильтра ошибок программирования. Таблица показывает также возможные причины ошибок.

Таблица 10-1 Возможные причины ошибок программирования, младшее слово

Ошибка	ID события (W#16#...)	Ошибка происходит ...
Ошибка преобразования BCD	2521	... когда преобразуемое значение не является числом в двоично-десятичном коде (например, 5E8)
Ошибка длины области при чтении	2522	... когда используется адрес, который не находится полностью в пределах возможной адресной области. Пример: нужно прочитать MW 320, хотя область памяти имеет длину только 256 байтов.
Ошибка длины области при записи	2523	... когда используется адрес, который не находится полностью в пределах возможной адресной области. Пример: нужно записать значение в MW 320, хотя область памяти имеет длину только 256 байтов.
Ошибка области при чтении	2524	... когда для адреса задан неправильный идентификатор области при использовании косвенной межсегментной адресации. Пример: правильно: LAR1 P#E 12.0 L W[AR1, P#0.0] неправильно: LAR1 P#12.0 L W[AR1, P#0.0] Для этой операции выводится сообщение об ошибке длины области.
Ошибка области при записи	2525	... когда для адреса задан неправильный идентификатор области при использовании косвенной межсегментной адресации. Пример: правильно: LAR1 P#E 12.0 T W[AR1, P#0.0] неправильно: LAR1 P#12.0 T W[AR1, P#0.0] Для этой операции выводится сообщение об ошибке длины области.
Ошибка номера таймера	2526	... когда происходит обращение к несуществующему таймеру. Пример: SP T [MW 0], где MW 0 = 129; должен запускаться таймер 129, хотя имеются только 128 доступных таймеров.
Ошибка номера счетчика	2527	... когда происходит обращение к несуществующему счетчику. Пример: CU C [MW 0], где MW 0 = 600; должно происходить обращение к счетчику 600, хотя имеются только 512 доступных счетчиков (CPU 416-D).
Ошибка выравнивания при чтении	2528	... когда адрес байта, слова или двойного слова указан с адресом бита $\neq 0$. Пример: правильно: LAR1 P#M12.0 L B[AR1, P#0.0] неправильно: LAR1 P#M12.4 L B[AR1, P#0.0]

Ошибка	ID события (W#16#...)	Ошибка происходит ...
Ошибка выравнивания при записи	2529	... когда адрес байта, слова или двойного слова указан с адресом бита $\neq 0$. Пример: правильно: LAR1 P#M12.0 T B[AR1, P#0.0] неправильно: LAR1 P#M12.4 T B[AR1, P#0.0]

Старшее слово ошибок программирования

Следующая таблица перечисляет ошибки, поставленные в соответствие старшему слову фильтра ошибок программирования. Перечисляются также возможные причины ошибок.

Таблица 10-2 Возможные причины ошибок программирования, старшее слово

Ошибка	ID события (W#16#...)	Ошибка происходит ...
Ошибка записи в блок данных	2530	... когда блок данных, в который должна производиться запись, доступен только для чтения.
Ошибка записи в экземплярный блок данных	2531	... когда экземплярный блок данных, в который должна производиться запись, доступен только для чтения.
Ошибка номера блока DB	2532	... когда должен открываться блок данных с номером, превышающим максимальный разрешенный номер.
Ошибка номера блока DI	2533	... когда должен открываться экземплярный блок данных с номером, превышающим максимальный разрешенный номер.
Ошибка номера блока FC	2534	... когда вызывается функция с номером, превышающим максимальный разрешенный номер.
Ошибка номера блока FB	2535	... когда вызывается функциональный блок с номером, превышающим максимальный разрешенный номер.
DB не загружен	253A	... когда открываемый блок данных не загружен.
FC не загружен	253C	... когда вызываемая функция не загружена.
SFC не существует	253D	... когда вызываемая системная функция не существует.
FB не загружен	253E	... когда вызываемый функциональный блок не загружен.
SFB не существует	253F	... когда вызываемый системный/ стандартный функциональный блок не существует.

Ошибки доступа

Следующая таблица перечисляет ошибки, поставленные в соответствие фильтру ошибок доступа во всех CPU, кроме CPU 417 и CPU 417H. Перечисляются также возможные причины ошибок.

Ошибка	ID события (W#16#...)	для...	Ошибка происходит...
Ошибка доступа к входам/выходам при чтении	2942	S7-300	... когда адресу в области входов/выходов не назначен сигнальный модуль или ... когда обращение к этой области входов/выходов не подтверждается в течение выбранного контрольного времени модуля (таймаута).
		S7-400	... во время первого неправильного доступа для чтения (таймаута)
	2944	S7-400	... когда адресу в области входов/выходов не назначен сигнальный модуль или во время n-го неправильного доступа для чтения ($n > 1$).
Ошибка доступа к входам/выходам при записи	2943	S7-300	... когда адресу в области входов/выходов не назначен сигнальный модуль или ... когда обращение к этой области входов/выходов не подтверждается в течение выбранного контрольного времени модуля (таймаута).
		S7-400	... во время первого неправильного доступа для записи (таймаута)
	2945	S7-400	... когда адресу в области входов/выходов не назначен сигнальный модуль или во время n-го неправильного доступа для записи ($n > 1$).

Ошибки доступа в случае CPU 417 и CPU 417H

Следующая таблица перечисляет ошибки, поставленные в соответствие фильтру ошибок доступа в CPU 417 и CPU 417H. Перечисляются также возможные причины ошибок.

Ошибка	ID события (W#16#...)	Ошибка происходит...
Ошибка доступа к входам/выходам при чтении	2942	... когда адресу в области входов/выходов не назначен сигнальный модуль или ... когда обращение к этой области входов/выходов не подтверждается в течение выбранного контрольного времени модуля (таймаута).
Ошибка доступа к входам/выходам при записи	2943	... когда адресу в области входов/выходов не назначен сигнальный модуль или ... когда обращение к этой области входов/выходов не подтверждается в течение выбранного контрольного времени модуля (таймаута).

10.2 Маскирование синхронных ошибок с помощью SFC36 "MSK_FLT"

Описание

С помощью SFC36 "MSK_FLT" (mask synchronous errors [маскировать синхронные ошибки]) вы можете управлять реакцией CPU на синхронные ошибки. С помощью этой SFC вы можете маскировать синхронные ошибки, используя фильтр ошибок (см. раздел 10.1). При вызове SFC36 вы маскируете синхронные ошибки в текущем классе приоритета.

Если вы устанавливаете в "1" отдельные биты фильтра синхронных ошибок во входных параметрах, то другие ранее установленные биты сохраняют свое значение "1". Следовательно, вы получаете новые фильтры ошибок, которые вы можете считывать через выходные параметры. Маскированные вами синхронные ошибки не вызывают OB, а просто вводятся в регистр ошибок. Вы можете считывать регистр ошибок с помощью SFC38 "READ_ERR" (см. раздел 10.4).

Параметр	Описание	Тип данных	Область памяти	Характеристика
PRGFLT_SET_MASK	INPUT	DWORD	I, Q, M, D, L, константа	Ошибки программирования, подлежащие маскированию.
ACCFLT_SET_MASK	INPUT	DWORD	I, Q, M, D, L, константа	ошибки доступа, подлежащие маскированию
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках
PRGFLT_MASKED	OUTPUT	DWORD	I, Q, M, D, L	Маскированные ошибки программирования
ACCFLT_MASKED	OUTPUT	DWORD	I, Q, M, D, L	Маскированные ошибки доступа

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Никакая из ошибок еще не была маскирована.
0001	По крайней мере, одна из ошибок была уже маскирована. Тем не менее, другие ошибки будут маскироваться.

10.3 Демаскирование синхронных ошибок с помощью SFC37 "DMSK_FLT"

Описание

С помощью SFC37 "DMSK_FLT" (unmask synchronous errors [демаскировать синхронные ошибки]) демаскируются синхронные ошибки, маскированные с помощью SFC36 "MSK_FLT". Для этого вы должны установить в "1" соответствующие биты фильтра ошибок во входных параметрах (см. раздел 10.1). Посредством вызова SFC 37 вы демаскируете соответствующие синхронные ошибки текущего класса приоритета. Одновременно стираются записи в регистре ошибок. Вы можете считывать новые фильтры ошибок через выходные параметры.

Параметр	Описание	Тип данных	Область памяти	Характеристика
PRGFLT_RESET_MASK	INPUT	DWORD	I, Q, M, D, L, константа	Ошибки программирования, подлежащие демаскированию
ACCFLT_RESET_MASK	INPUT	DWORD	I, Q, M, D, L, константа	Ошибки доступа, подлежащие демаскированию
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках
PRGFLT_MASKED	OUTPUT	DWORD	I, Q, M, D, L	Все еще маскированные ошибки программирования
ACCFLT_MASKED	OUTPUT	DWORD	I, Q, M, D, L	Все еще маскированные ошибки доступа

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Все указанные ошибки были демаскированы.
0001	По крайней мере, одна из ошибок не была маскирована. Тем не менее, другие ошибки будут демаскированы.

10.4 Чтение регистра ошибок с помощью SFC38 "READ_ERR"

Описание

С помощью SFC38 "READ_ERR" (read error register [прочитать регистр ошибок]) вы можете считать регистр ошибок. Структура регистра ошибок соответствует структуре фильтров ошибок программирования и доступа, которые вы можете запрограммировать в качестве входных параметров с помощью SFC36 и SFC37 (см. раздел 10.1).

Во входных параметрах вы вводите синхронные ошибки, которые вы хотите считывать из регистра ошибок. Когда вы вызываете SFC38, вы считываете желаемые записи из регистра ошибок и одновременно стираете эти записи.

Регистр ошибок содержит информацию о том, какие из маскированных синхронных ошибок в текущем классе приоритета произошли хотя бы один раз. Если бит установлен, значит, соответствующая маскированная синхронная ошибка появилась, по крайней мере, один раз.

Параметр	Описание	Тип данных	Область памяти	Характеристика
PRGFLT_QUERY	INPUT	DWORD	I, Q, M, D, L, константа	Опрос ошибок программирования
ACCFLT_QUERY	INPUT	DWORD	I, Q, M, D, L, константа	Опрос ошибок доступа
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках
PRGFLT_ESR	OUTPUT	DWORD	I, Q, M, D, L	Появившаяся ошибка программирования
ACCFLT_ESR	OUTPUT	DWORD	I, Q, M, D, L	Появившаяся ошибка доступа

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Все опрошенные ошибки маскированы.
0001	По крайней мере, одна из опрошенных ошибок не маскирована.

11 SFC для обработки прерываний и асинхронных ошибок

11.1 Задержка и блокировка прерываний и асинхронных ошибок

Назначение SFC39 – SFC42

С помощью этих SFC вы можете достичь следующего:

- Заблокировать прерывания и асинхронные ошибки с помощью SFC39 "DIS_IRT" на время всех последующих циклов CPU.
- Задержать более высокие классы приоритета с помощью SFC41 "DIS_AIRT" до конца OB.
- Разблокировать прерывания и асинхронные ошибки с помощью SFC40 "EN_IRT" или SFC42 "EN_AIRT".

Вы программируете обработку прерываний и асинхронных ошибок в программе пользователя. Для этого вы должны запрограммировать также соответствующие OB.

Преимущество SFC41 и SFC42

Задержка высокоприоритетных прерываний и асинхронных ошибок путем их блокировки с помощью SFC 41 "DIS_AIRT" и последующей разблокировки с помощью SFC 42 "EN_AIRT" имеет следующие преимущества:

Количество задержанных прерываний подсчитывается CPU. Если вы задержали прерывания и асинхронные ошибки, то эта задержка не может быть отменена стандартными вызовами FC, если эти прерывания и асинхронные ошибки также блокируются и снова разблокируются в самих этих стандартных FC.

Классы прерываний

Прерывания подразделены на разные классы. Следующая таблица перечисляет все классы прерываний и соответствующие OB.

Класс прерывания	ОВ
Прерывания по времени	OB10 – OB17
Прерывания с задержкой	OB20 – OB23
Циклические прерывания	OB30 – OB38
Аппаратные прерывания	OB40 – OB47
Коммуникационные прерывания	OB50, OB51
Прерывания многопроцессорной обработки	OB60
Прерывания ошибок резервирования	OB70, OB72
Прерывания асинхронных ошибок	OB80 – OB87
Прерывания синхронных ошибок	OB121, OB122 (вы можете маскировать или демаскировать обработку прерываний синхронных ошибок с помощью SFC36 – SFC38)

Асинхронные ошибки

Следующая таблица перечисляет все асинхронные ошибки, на которые вы можете реагировать с помощью вызова ОВ в программе пользователя.

Асинхронные ошибки	ОВ
Ошибка времени (например, превышение времени цикла)	ОВ80
Неисправность источника питания (например, отказ батарейки)	ОВ81
Диагностическое прерывание (например, неисправный предохранитель в сигнальном модуле)	ОВ82
Прерывание из-за удаления/вставки модуля	ОВ83
Аппаратная ошибка CPU (например, удалена плата памяти)	ОВ84
Ошибка исполнения программы	ОВ85
Выход из строя стойки	ОВ86
Коммуникационная ошибка	ОВ87

11.2 Блокировка обработки новых прерываний и асинхронных ошибок с помощью SFC39 "DIS_IRT"

Описание

С помощью SFC 39 "DIS_IRT" (disable interrupt [заблокировать прерывание]) вы блокируете обработку новых прерываний и асинхронных ошибок. Это означает, что при возникновении прерывания операционная система CPU реагирует следующим образом:

- Она **не** вызывает ни ОБ прерываний, ни ОБ асинхронных ошибок и
- **не** запускает стандартную реакцию, если ОБ прерываний или ОБ асинхронных ошибок не запрограммирован.

Блокировка прерываний и асинхронных ошибок остается в силе для всех классов приоритета. Воздействие "DIS_IRT" может быть отменено только вызовом SFC40 "EN_IRT" (см. раздел 11.3) или теплым или холодным рестартом.

Записывает ли операционная система прерывания и асинхронные ошибки, когда они происходят, в диагностический буфер, зависит от выбранной вами настройки входного параметра MODE [режим].

Замечание

Помните, что при программировании использования SFC39 "DIS_IRT" все происходящие прерывания теряются!

Параметр	Описание	Тип данных	Область памяти	Характеристика
MODE	INPUT	BYTE	I, Q, M, D, L, константа	Указывает, какие прерывания и асинхронные ошибки блокируются (см. ниже таблицу MODE).
OB_NR	INPUT	INT	I, Q, M, D, L, константа	Номер ОБ
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если ошибка происходит, когда эта функция активна, то возвращаемое значение содержит код ошибки.

MODE

MODE	Значение
00	Вновь появляющиеся прерывания и асинхронные ошибки блокируются. (синхронные ошибки не блокируются). Назначьте параметру OB_NR значение 0. Записи в диагностический буфер продолжают заноситься.
01	Все вновь появляющиеся события заданного класса прерываний блокируются. Класс прерываний задается указанием номера первого OB, например, OB 40 для аппаратных прерываний (см. таблицу "Классы прерываний"). Записи в диагностический буфер продолжают заноситься.
02	Все новые появления заданного прерывания блокируются. Это прерывание задается с помощью номера OB. Записи в диагностический буфер продолжают заноситься.
80	Все вновь появляющиеся прерывания и асинхронные ошибки блокируются и более не вносятся в диагностический буфер. Операционная система вносит в диагностический буфер событие W#16#5380.
81	Все вновь появляющиеся прерывания, принадлежащие заданному классу прерываний, блокируются и более не вносятся в диагностический буфер. Операционная система вносит в диагностический буфер событие W#16#5380.
82	Все вновь появляющиеся события, относящиеся к заданному прерыванию, блокируются и более не вносятся в диагностический буфер. Операционная система вносит в диагностический буфер событие W#16#5380.

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Ошибки не было.
8090	Входной параметр OB_NR содержит недопустимое значение.
8091	Входной параметр MODE содержит недопустимое значение.

11.3 Разблокировка обработки новых прерываний и асинхронных ошибок с помощью SFC40 "EN_IRT"

Описание

С помощью SFC 40 "EN_IRT"(enable interrupt [разблокировать прерывание]) вы разблокируете обработку новых прерываний и асинхронных ошибок, заблокированную посредством SFC39 "DIS_IRT". Это значит, что при возникновении события прерывания операционная система CPU реагирует одним из следующих способов:

- Вызывает ОВ прерываний или ОВ асинхронных ошибок.
- Запускает стандартную реакцию, если нет запрограммированного ОВ прерываний или ОВ асинхронных ошибок.

Параметр	Описание	Тип данных	Область памяти	Характеристика
MODE	INPUT	BYTE	I, Q, M, D, L, константа	Указывает, какие прерывания и асинхронные ошибки будут разблокированы.
OB_NR	INPUT	INT	I, Q, M, D, L, константа	Номер ОВ
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если ошибка происходит, когда эта функция активна, то возвращаемое значение содержит код ошибки.

MODE

MODE	Значение
0	Все вновь появляющиеся прерывания и асинхронные ошибки разблокируются.
1	Все вновь появляющиеся события заданного класса прерываний разблокируются. Вы задаете класс прерываний с помощью номера первого ОВ, например, ОВ40 для аппаратных прерываний (см. раздел 11.1).
2	Все вновь появляющиеся события заданного прерывания разблокируются. Это прерывание задается номером ОВ.

Информация об ошибках

Код ошибки (W#16#...)	Описание
0000	Ошибки не было.
8090	Входной параметр OB_NR содержит недопустимое значение.
8091	Входной параметр MODE содержит недопустимое значение.

11.4 Задержка обработки прерываний более высокого приоритета и асинхронных ошибок с помощью SFC41 "DIS_AIRT"

Описание

С помощью SFC 41 "DIS_AIRT" (disable alarm interrupts [заблокировать аварийное прерывание]) вы задерживаете обработку ОВ прерываний и ОВ асинхронных ошибок, приоритет которых выше, чем приоритет текущего ОВ. Вы можете вызывать SFC 41 в ОВ многократно. Вызовы SFC 41 подсчитываются операционной системой.

Каждый из этих вызовов остается в силе, пока он не будет снова отменен индивидуально вызовом SFC42 "EN_AIRT" **или** пока текущий ОВ не будет полностью обработан.

Прерывания и асинхронные ошибки, которые произошли во время действия SFC41, обрабатываются, как только они снова разблокируются с помощью SFC42 "EN_AIRT" или как только будет исполнен текущий ОВ.

Параметр	Описание	Тип данных	Область памяти	Характеристика
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Количество задержек (= количеству вызовов SFC 41)

Возвращаемое значение

Следующая таблица показывает возвращаемое значение для SFC 41, которое выводится через параметр RET_VAL.

Возвращаемое значение	Описание
n	"n" показывает, сколько раз обработка была заблокирована, иными словами, число вызовов SFC41 (обработка прерываний опять разблокируется лишь тогда, когда n = 0; см. раздел 11.5).

11.5 Разблокировка обработки прерываний более высокого приоритета и асинхронных ошибок с помощью SFC42 "EN_AIRT"

Описание

С помощью SFC 42 "EN_AIRT" (enable alarm interrupts [разблокировать аварийные прерывания]) вы разблокируете SFC 41 обработку прерываний более высокого приоритета и асинхронных ошибок, заблокированную ранее с помощью "DIS_AIRT". Каждый вызов SFC41 должен быть отменен вызовом SFC42.

Пример

Если вы, например, пять раз заблокировали прерывания с помощью пяти вызовов SFC41, то вы должны отменить эти вызовы пятью вызовами SFC42.

Параметр	Описание	Тип данных	Область памяти	Характеристика
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Количество все еще запрограммированных задержек после выполнения SFC 42 или сообщение об ошибке.

Возвращаемое значение и информация об ошибках

Как оценивать информацию об ошибках из параметра RET_VAL, объяснено в главе 2. В этой главе вы также найдете общую информацию об ошибках SFC. Следующая таблица содержит всю информацию об ошибках, относящуюся к SFC42, которая может быть выведена с помощью параметра RET_VAL.

Возвращаемое значение и информация об ошибках	Описание
n	"n" показывает количество вызовов SFC41, еще не отмененных вызовами SFC42 (обработка прерываний снова разблокируется только при "n" = 0).
W#16#8080	Функция была вызвана снова, хотя обработка прерываний уже была разблокирована.

12 SFC для диагностики

12.1 Системная диагностика

CPU поддерживают внутренние данные о состоянии программируемого логического контроллера. С помощью системных диагностических функций вы можете считывать наиболее важные данные. Некоторые из этих данных можно отображать на устройстве программирования, используя STEP 7.

Вы можете также обращаться к данным, требуемым для диагностики системы в своей программе, используя SFC "RD_SINFO" и "RDSYSST".

12.2 Чтение стартовой информации ОБ с помощью SFC6 "RD_SINFO"

Описание

С помощью SFC6 "RD_SINFO" (read start information [читать стартовую информацию]) вы можете прочитать стартовую информацию, содержащую следующие сведения:

- Последний вызванный ОБ, который еще не полностью выполнен и
- Последний запущенный ОБ запуска.

В обоих случаях метка времени отсутствует. Если вызов происходит в ОБ100, ОБ101 или ОБ102, то возвращаются два идентичных сообщения со стартовой информацией.

Параметр	Описание	Тип данных	Область памяти	Характеристика
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибке
TOP_SI	OUTPUT	STRUCT	D, L	Стартовая информация текущего ОБ
START_UP_SI	OUTPUT	STRUCT	D, L	Стартовая информация запущенного последним ОБ запуска

TOP_SI и START_UP_SI

Выходные параметры TOP_SI и START_UP_SI являются двумя структурами с одинаковыми элементами (см. следующую таблицу).

Элемент структуры	Тип данных	Описание
EV_CLASS	BYTE	<ul style="list-style-type: none"> • Биты с 0 по 3: Идентификатор события • Биты с 4 по 7: Класс события
EV_NUM	BYTE	Номер события
PRIORITY	BYTE	Номер класса приоритета
NUM	BYTE	Номер ОБ
TYP2_3	BYTE	Идентификатор данных 2_3: характеризует информацию, записанную в ZI2_3
TYP1	BYTE	Идентификатор данных 1: характеризует информацию, записанную в ZI1
ZI1	WORD	Дополнительная информация 1
ZI2_3	DWORD	Дополнительная информация 2_3

Замечание

Элементы структуры, перечисленные в таблице, и временные переменные ОБ имеют идентичное содержимое.

Обратите, пожалуйста, внимание на то, что временные переменные отдельных ОБ могут, однако, иметь различные имена и различные типы данных (**см. следующий пример**). Обратите также внимание на то, что интерфейс вызова каждого ОБ включает в себя дополнительную информацию, содержащую дату и время запроса ОБ.

Биты с 4 по 7 элемента структуры EV_CLASS содержат класс события. Здесь возможны следующие значения:

- 1: Стартовые события из стандартных OB
- 2: Стартовые события из OB синхронных ошибок
- 3: Стартовые события из OB асинхронных ошибок

Структурный элемент PRIORITY передает класс приоритета, относящийся к текущему OB (см. главу 2).

Кроме этих двух элементов, имеет значение также NUM. NUM Наряду с этими двумя элементами, важное значение имеет еще NUM. NUM содержит номер текущего OB или запущенного последним OB запуска.

Пример

Пусть последним был вызван и еще не полностью обработан OB 80. Пусть последним запущенным OB запуска будет OB 100.

Следующая таблица показывает соответствие между структурными элементами параметра TOP_SI функции SFC 6 "RD_SINFO" и локальными переменными OB 80.

TOP_SI		OB 80	
Элемент структуры	Тип данных	Локальная переменная	Тип данных
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

Следующая таблица показывает соответствие между структурными элементами параметра START_UP_SI функции SFC 6 "RD_SINFO" и локальными переменными OB 100.

START_UP_SI		OB 100	
Элемент структуры	Тип данных	Локальная переменная	Тип данных
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

Информация об ошибках

SFC6 "RD_SINFO" предоставляет только общую, а не специфическую информацию об ошибках. Коды общих ошибок и способы их оценки описаны в главе 2 "Общие параметры для SFC".

12.3 Чтение списка состояний системы или подписка с помощью SFC51 "RDSYSST"

Описание

С помощью системной функции SFC51 "RDSYSST" (read system status [читать состояние системы]) читают список состояний системы или частичный список состояний системы.

Чтение запускают, присваивая при вызове SFC51 входному параметру REQ значение 1. Если состояние системы можно было прочитать сразу, то SFC возвращает в выходном параметре BUSY значение 0. Если BUSY имеет значение 1, то функция чтения еще не завершилась (см. раздел 2.2).

Замечание

Если вы вызываете SFC51 "RDSYSST" в ОБ диагностического прерывания с SSL-ID W#16#00B1 или W#16#00B2 или W#16#00B3 и обращаетесь к модулю, который инициировал диагностическое прерывание, то состояние системы читается немедленно.

Системные ресурсы

Если вы запускаете несколько асинхронных функций чтения (задания с SSL_ID W#16#00B4, и W#16#4C91, и W#16#4092, и W#16#4292, и W#16#4692, и, возможно, W#16#00B1, и W#16#00B3) одну за другой через короткие интервалы времени, то операционная система гарантирует, что все задания на чтение выполняются и что они не создают помех друг для друга. Если достигаются пределы системных ресурсов, то это отображается в RET_VAL. Вы можете исправить эту ситуацию нерегулярной ошибкой, повторив задание.

Максимальное число "одновременно" активных заданий SFC51 зависит от CPU. Вы найдете эту информацию в [/70/](#) и [/101/](#).

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	REQ=1: запускает обработку.
SSL_ID	INPUT	WORD	I, Q, M, D, L, константа	Идентификатор списка состояний системы или частичного списка, который нужно читать (частичные списки объясняются в главе 27).
INDEX	INPUT	WORD	I, Q, M, D, L, константа	Тип или номер объекта в частичном списке.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если при выполнении SFC происходит ошибка, то параметр RET_VAL содержит код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	TRUE: Чтение еще не завершилось.
SSL_HEADER	OUTPUT	STRUCT	D, L	См. ниже.
DR	OUTPUT	ANY	I, Q, M, L, D	Целевая область для прочитанного списка SSL или для прочитанного частичного списка SSL: <ul style="list-style-type: none"> Если вы считали информацию только заголовка списка SSL, то вы должны оценивать не DR, а только SSL_HEADER. В противном случае, произведение LENGTHDR и N_DR показывает, сколько байтов были введены в DR.

SSL_HEADER

Параметр SSL_HEADER является структурой, определенной следующим образом:

```
SSL_HEADER: STRUCT
    LENGTHHDR: WORD
    N_DR:      WORD
END_STRUCT
```

LENGTHHDR - это длина записи данных списка SSL или частичного списка SSL.

- Если вы считали информацию только заголовка списка SSL, то N_DR содержит количество принадлежащих ему записей данных.
- В противном случае N_DR содержит количество записей данных, переданных в область назначения.

Информация об ошибках

Код ошибки (W#16#...)	Характеристика
0000	Ошибок нет.
0081	Поле результата слишком короткое. (Тем не менее, передается столько записей данных, сколько возможно. Заголовок SLZ указывает это количество).
7000	Первый вызов с REQ=0: Передача данных не активна; BUSY имеет значение 0.
7001	Первый вызов с REQ=1: Передача данных запущена; BUSY имеет значение 1.
7002	Промежуточный вызов (REQ не имеет значения): Передача данных уже активна; BUSY имеет значение 1.
8081	Поле результата слишком короткое (недостаточно места для одной записи данных).
8082	SSL_ID неверен или неизвестен в CPU или SFC.
8083	Неправильный или не разрешенный INDEX.
8085	Из-за проблемы в системе информация в настоящее время недоступна (например, из-за недостатка ресурсов).
8086	Запись данных не может быть прочитана из-за ошибки системы (шина, модули, операционная система).
8087	Запись данных не может быть прочитана, потому что модуль не существует или не подтверждает.
8088	Запись данных не может быть прочитана, потому что фактический идентификатор типа отличается от ожидаемого идентификатора типа.
8089	Запись данных не может быть прочитана, потому что модуль не обладает диагностическими свойствами.
80A2	Ошибка протокола DP (ошибка уровня 2) (нерегулярная ошибка).
80A3	Ошибка протокола DP у пользовательского интерфейса / пользователя (нерегулярная ошибка).
80A4	Ошибка связи в коммуникационной шине (ошибка происходит между CPU и внешним интерфейсным модулем DP).
80C5	Децентрализованная периферия недоступна (нерегулярная ошибка).
80C6	Передача записи данных прекратилась из-за прерывания класса приоритета (рестарт или фоновый режим).

SSL_ID

Замечание

По поводу частичных списков, которые можно считывать с помощью SFC51 "RDSYSST", обратитесь

- к /70/ для S7-300
- к следующей таблице для S7-400.

SSL ID (W#16#...)	Частичный список	INDEX (W#16#...)
	Идентификатор (ID) модуля	
0111	Одна идентифицирующая запись данных	
	Идентификация модуля	0001
	Идентификация основного комплекта оборудования	0006
	Идентификация основного комплекта оборудования	0007
	Характеристики CPU	
0012	Все характеристики	Не имеет значения
0112	Характеристики одной группы	
	Процессор MC7	0000
	Система времени	0100
	Поведение системы	0200
	Описание языка MC7	0300
0F12	Информация только заголовка частичного списка SSL	Не имеет значения
	Области памяти пользователя	
0013	Одна запись данных для указанной области памяти	
0113	Одна запись данных для указанной области памяти	
	Рабочая память	0001
0F13	Информация только заголовка частичного списка SSL	Не имеет значения
	Системные области	
0014	Записи данных всех системных областей	Не имеет значения
0F14	Информация только заголовка частичного списка SSL	Не имеет значения
	Типы модулей	
0015	Записи данных всех типов модулей	Не имеет значения
	Состояние светодиодов модулей (не может считываться из всех CPU, см. /102 /).	
0019	Состояние всех светодиодов	Не имеет значения
0F19	Информация только заголовка частичного списка SSL	Не имеет значения
	Состояние прерывания	
0222	Запись данных указанного прерывания	Номер OB
	Данные о состоянии связи	
0132	Данные о состоянии одного устройства связи	
	Диагностика	0005
	Система времени	0008
0232	Данные о состоянии одного устройства связи	
	Уровень защиты CPU и параметры настройки операторского управления	
	Групповая информация H CPU	
0071	Информация о текущем состоянии H-системы	Не имеет значения
0F71	Информация только заголовка частичного списка SSL	Не имеет значения
	Состояние светодиодов модулей (не может считываться из всех CPU, см. /102 /).	
0174	Состояние светодиода	LED ID
	Информация о состоянии модуля (передается не более 27 записей данных)	
0091	Информация о состоянии всех вставленных модулей / submodule	Не имеет значения
0191	Информация о состоянии всех модулей / стоек с неправильным идентификатором типа	Не имеет значения
0291	Информация о состоянии всех неисправных модулей	Не имеет значения

SSL ID (W#16#...)	Частичный список	INDEX (W#16#...)
0391	Информация о состоянии всех недоступных модулей	Не имеет значения
0591	Информация о состоянии всех субмодулей главного модуля	Не имеет значения
0991	Информация о состоянии всех субмодулей главного модуля в указанной стойке	Идентификатор стойки или master-системы DP
0A91	Информация о состоянии модулей всех master-систем DP	Не имеет значения
0C91	Информация о состоянии модуля в центральной конфигурации или модуля, подключенного к встроенному коммуникационному процессору DP	Логический базовый адрес
4C91	Информация о состоянии модуля, подключенного к внешнему коммуникационному процессору DP	Логический базовый адрес
0D91	Информация о состоянии всех модулей в указанной стойке / станции DP	Идентификатор (ID) стойки или master-системы DP или ID master-системы DP и номер станции
0E91	Информация о состоянии всех размещенных модулей	Не имеет значения
	Информация о состоянии стойки / станции	
0092	Ожидаемое состояние стойки в центральной конфигурации / станций master-системы DP	0 / ID master-системы DP
4092	Ожидаемое состояние станций master-системы DP, подключенной к внешнему интерфейсу DP	ID master-системы DP
0292	Фактическое состояние стойки в центральной конфигурации / станций master-системы DP	0 / ID master-системы DP
4292	Фактическое состояние станций master-системы DP, подключенной через внешний интерфейсный модуль DP	ID master-системы DP
0692	Нормальное состояние стоек расширения в центральной конфигурации / станций master-системы DP, подключенной через встроенный интерфейсный модуль DP	0 / ID master-системы DP
4692	Нормальное состояние станций master-системы DP, подключенной через внешний интерфейсный модуль DP	ID master-системы DP
	Диагностический буфер (максимум 21 запись данных)	
00A0	Все записи, которые могут быть поставлены в активном режиме работы в настоящее время	Не имеет значения
01A0	Самые недавние записи, количество задается в индексе	Количество
0FA0	Информация только заголовка частичного списка SSL	Не имеет значения
	Диагностические данные модулей	
00B1	Первые четыре диагностических байта одного модуля (запись данных 0)	Логический базовый адрес
00B2	Все диагностические данные одного модуля (≤ 220 байтов, запись данных 1) (не модули DP)	Стойка, слот
00B3	Все диагностические данные одного модуля (≤ 220 байтов, запись данных 1)	Логический базовый адрес
00B4	Диагностические данные slave-устройства DP	Конфигурированный диагностический адрес

12.4 Запись диагностического события, определенного пользователем, в диагностический буфер с помощью SFC52 "WR_USMSG"

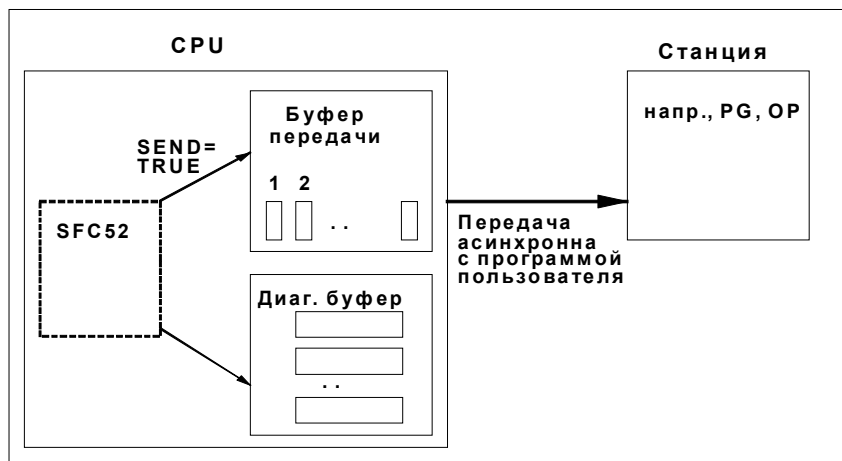
Описание

С помощью SFC 52 "WR_USMSG" (write user element in diagnosis buffer [записать элемент пользователя в диагностический буфер]) определенное пользователем диагностическое событие записывается в диагностический буфер. Кроме того, вы можете передать соответствующее диагностическое сообщение всем зарегистрированным для этой цели станциям. Если появляется ошибка, то выходной параметр RET_VAL предоставляет информацию об ошибке.

Передача определенного пользователем диагностического сообщения

Определенное пользователем диагностическое событие записывается в буфер диагностики с помощью SFC 52. Вы можете также передать соответствующее диагностическое сообщение всем зарегистрированным для этой цели станциям. Тогда определенное пользователем диагностическое сообщение записывается в буфер передачи и оттуда автоматически передается зарегистрированным для этого станциям.

Вы можете проверить, возможна ли в данный момент передача определенных пользователем диагностических сообщений. Для этого вызовите SFC 51 "RDSYSST" с параметрами SSL_ID = W#16#0132 и INDEX = W#16#0005. Четвертое слово полученной записи данных показывает, возможна в данный момент передача (1) или нет (0).



Переполнение передающего буфера

Запись диагностического сообщения в буфер передачи может происходить только тогда, когда буфер передачи неполон. Количество записей, которое может быть сделано в буфере передачи, зависит от типа используемого вами CPU.

Если буфер передачи полон, то:

- диагностическое событие вносится, тем не менее, в диагностический буфер,
- в параметре RET_VAL указывается, что буфер передачи заполнен (RET_VAL = W#16#8092).

Станция не зарегистрирована

Если должно передаваться определенное пользователем диагностическое сообщение (SEND = TRUE) и ни одна станция не зарегистрирована, то

- определенное пользователем диагностическое событие вносится в диагностический буфер,
- параметр RET_VAL указывает, что нет зарегистрированных станций (RET_VAL = W#16#8091 или W#16#8091. Значение W#16#8091 появляется у старых версий CPU).

Общая структура

Элемент в диагностическом буфере имеет следующую структуру:

Байт	Содержимое
1 и 2	Идентификатор (ID) события
3	Класс приоритета
4	Номер ОВ
5 и 6	Резерв
7 и 8	Дополнительная информация 1
9, 10, 11 и 12	Дополнительная информация 2
13 – 20	Отметка времени

Идентификатор (ID) события

Структура ID события объяснена в разделе 26.1. Каждому событию поставлен в соответствие ID события.

Дополнительная информация

Это дополнительная информация о событии. Дополнительная информация может быть различна для каждого события. Когда вы создаете диагностическое событие, то вы можете сами определить содержание этих записей.

Когда вы посылаете определенное пользователем диагностическое сообщение, вы можете встроить дополнительную информацию в текст сообщения (относящийся к идентификатору события) в качестве сопутствующего значения.

Отметка времени

Отметка времени имеет тип Date_and_Time.

Параметр	Описание	Тип данных	Область памяти	Характеристика
SEND	INPUT	BOOL	I, Q, M, D, L, константа	Разрешить передачу определенного пользователем диагностического сообщения всем зарегистрированным станциям
EVENTN	INPUT	WORD	I, Q, M, D, L, константа	Идентификатор (ID) события – Его назначаете вы. Он не назначается сервером сообщений.
INFO1	INPUT	ANY	I, Q, M, D, L	Дополнительная информация длиной в 1 слово
INFO2	INPUT	ANY	I, Q, M, D, L	Дополнительная информация длиной в 2 слова
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках

SEND

Если SEND = TRUE, определенное пользователем диагностическое сообщение передается всем зарегистрированным станциям. Сообщение передается только тогда, когда станция зарегистрирована, и если буфер передачи не полон. Передача элемента асинхронна по отношению к программе пользователя.

EVENTN

Параметр EVENTN содержит идентификатор (ID) определенного пользователем события. Вы можете вводить идентификаторы событий вида W#16#8xyz, W#16#9xyz, W#16#Axyz, W#16#Bxyz.

Идентификаторы в форматах W#16#8xyz и W#16#9xyz принадлежат заранее определенным событиям, идентификаторы в форматах W#16#Axyz и W#16#Bxyz принадлежат свободно определяемым событиям.

Поступающее событие отмечается с помощью x = 1, уходящее событие – с помощью x = 0. У событий в классе A и B "yz" – это номер, назначенный сообщению в конфигурации сообщений, в шестнадцатеричном формате.

Структура ID события объяснена в разделе 26.1.

INFO1

Параметр INFO1 содержит информацию длиной в одно слово. Для INFO1 допустимы следующие типы данных:

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

Вы можете встроить параметр INFO1 в текст сообщения как сопутствующее значение и, таким образом, добавить к сообщению новейшую информацию.

INFO2

Параметр INFO2 содержит информацию длиной в два слова. Для INFO2 допустимы следующие типы данных:

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

Вы можете встроить параметр INFO2 в текст сообщения как сопутствующее значение и, таким образом, добавить к сообщению новейшую информацию.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Нет ошибки
0091	Нет зарегистрированных станций (в диагностический буфер вносится диагностическое событие)
8083	Недопустимый тип данных INFO1
8084	Недопустимый тип данных INFO2
8085	EVENTN недопустим
8086	Недопустимая длина INFO1
8087	Недопустимая длина INFO2
8091	(Этот код ошибки появляется только у старых версий CPU). Нет зарегистрированных станций (в диагностический буфер вносится диагностическое событие).
8092	Передача в данный момент невозможна, буфер передачи полон (в диагностический буфер вносится диагностическое событие).

13 SFC и SFB для обновления образа процесса и обработки битовых массивов

13.1 Обновление таблицы входов образа процесса с помощью SFC26 "UPDAT_PI"

Описание

С помощью SFC 26 "UPDAT_PI" (update process image [обновить образ процесса]) вы обновляете таблицу входов образа процесса блока OB1 (=раздел 0 образа процесса) или раздел входов образа процесса, определенный с помощью STEP 7.

Если вы запроектировали повторную передачу сигнала об ошибках доступа к входам/выходам для обновления таблицы образа процесса системой, то выбранная таблица образа процесса будет обновляться функцией SFC26 постоянно.

В противном случае SFC26 будет обновлять таблицу образа процесса только тогда, когда выбранный раздел образа процесса не обновляется системой, иными словами:

- когда вы не поставили этот раздел образа процесса в соответствие OB прерываний или
- когда вы выбрали раздел 0 образа процесса и заблокировали обновление раздела образа процесса блока OB1 в конфигурации.

Замечание

Каждый логический адрес, который вы ставите в соответствие разделу таблицы входов образа процесса с помощью STEP 7, более не принадлежит таблице входов образа процесса блока OB1.

Вызовы SFC26 не оказывают влияния на обновление таблицы входов образа процесса блока OB1 и разделов входов образа процесса, которые вы назначили OB прерываний.

Параметр	Описание	Тип данных	Область памяти	Характеристика
PART	INPUT	BYTE	I, Q, M, D, L, константа	Номер раздела образа процесса, подлежащего обновлению. Допустимые значения: от 0 до 15 (0 означает образ процесса блока OB1; n, где $1 \leq n \leq 15$, означает раздел n образа процесса)
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках
FLADDR	OUTPUT	WORD	I, Q, M, D, L	Адрес первого байта, вызвавшего ошибку, если произошла ошибка доступа.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8090	Недопустимое значение входного параметра PART.
8091	Указанный раздел образа процесса не был определен или не находится в разрешенной области таблицы образа процесса в CPU.
8092	Раздел образа процесса обновляется системой с помощью OB, а вы не запроектировали повторную сигнализацию обо всех ошибках доступа к входам/выходам. Образ процесса не был обновлен функцией SFC26 "UPDAT_PI"
80A0	При обновлении была обнаружена ошибка доступа.

13.2 Обновление таблицы выходов образа процесса с помощью SFC27 "UPDAT_PO"

Описание

С помощью SFC27 "UPDAT_PO" (update process outputs [обновить выходы процесса]) вы передаете состояния сигналов таблицы выходов образа процесса блока OB1 (= раздел 0 образа процесса) или раздел образа процесса, определенный с помощью STEP 7, в модули вывода.

Замечание

Каждый логический адрес, который вы назначаете разделу таблицы выходов образа процесса с помощью STEP 7, больше не принадлежит таблице выходов образа процесса блока OB1.

Вызовы SFC27 не влияют на передачу таблицы выходов образа процесса блока OB1 и разделов выходов образа процесса, которые вы назначили для OB прерывания.

Параметр	Описание	Тип данных	Область памяти	Характеристика
PART	INPUT	BYTE	I, Q, M, D, L, константа	Номер обновляемого раздела выходов образа процесса. Разрешенные значения: от 0 до 15 (0 обозначает образ процесса OB1, n обозначает раздел n образа процесса, где $1 \leq n \leq 15$).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках.
FLADDR	OUTPUT	WORD	I, Q, M, D, L	Адрес первого байта, который вызвал ошибку, если произошла ошибка доступа.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8090	Недопустимое значение входного параметра PART.
8091	Заданный раздел образа процесса не был определен или не находится в разрешенной области образа процесса в CPU.
80A0	Во время обновления была обнаружена ошибка доступа.

13.3 Установка области выходов с помощью SFC79 "SET"

Описание

Вызов SFC79 "SET" (set range of outputs [установить область выходов]) приводит к следующему результату:

- Устанавливается битовый массив в периферийной области входов/выходов, выбранный с помощью параметров N и SA.
- Соответствующие биты в таблице выходов образа процесса также устанавливаются независимо от того, находятся они в разделе образа процесса или нет.
- Битовый массив должен быть частью периферийной области входов/выходов, назначенной образу процесса.

Если для части выбранного битового массива не подключен модуль, то функция SFC79 все же пытается установить весь битовый массив. Затем она возвращает в RET_VAL соответствующую информацию об ошибках.

Замечание

Во время выполнения SFC79 в область входов/выходов всегда записываются целые байты.

Если битовый массив, выбранный с помощью параметров N и SA, не начинается или не заканчивается на границе байта, то вызов SFC79 приводит к следующему результату:

- Биты в первом и последнем байтах, которые должны передаваться в периферийную область входов/выходов и которые не принадлежат выбранному битовому массиву, содержат значения соответствующих битов в таблице выходов образа процесса. Это может привести к непредусмотренным реакциям, таким как запуск электродвигателя или выключение системы охлаждения.
- Биты, принадлежащие выбранному битовому массиву, устанавливаются так, как объяснено выше.

Если вы присваиваете параметру N значение 0, то вызов SFC79 не оказывает никакого воздействия. Если главное управляющее реле не установлено, то вызов SFC79 не оказывает никакого воздействия.

Параметр	Описание	Тип данных	Область памяти	Характеристика
N	INPUT	INT	I, Q, M, D, L, константа	Количество устанавливаемых битов.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках.
SA	OUTPUT	POINTER	P	Указатель на первый устанавливаемый бит.

Информация об ошибках

Как оценивать информацию об ошибках из параметре RET_VAL, объясняется в главе 2. Эта глава содержит также общую информацию об ошибках SFC. SFC79 не дает в параметре RET_VAL какой-либо информации о конкретных ошибках.

13.4 Сброс области выходов с помощью SFC80 "RSET"

Описание

Вызов SFC80 "RSET" (reset range of outputs [сбросить область выходов]) приводит к следующему результату:

- Сбрасывается битовый массив в периферийной области входов/выходов, выбранной с помощью параметров N и SA.
- Соответствующие биты в таблице выходов образа процесса также сбрасываются независимо от того, находятся они в разделе образа процесса или нет.

Битовый массив должен располагаться в той части периферийной области входов/выходов, которой назначен образ процесса.

Если для части выбранного битового массива не подключен модуль, то SFC80 все же пытается сбросить весь битовый массив. Затем он возвращает в RET_VAL соответствующую информацию об ошибках.

Замечание

Во время выполнения SFC80 в периферийную область входов/выходов записываются целые байты.

Если битовый массив, выбранный с помощью параметров N и SA, не начинается или не заканчивается на границе байта, то вызов SFC80 приводит к следующему результату:

- Биты в первом и последнем байтах, которые должны передаваться в периферийную область входов/выходов и которые не принадлежат выбранному битовому массиву, содержат значения соответствующих битов в таблице выходов образа процесса. Это может привести к непредусмотренным реакциям, таким как запуск электродвигателя или выключение системы охлаждения.
- Биты, принадлежащие выбранному битовому массиву, устанавливаются так, как объяснено выше.

Если вы присваиваете параметру N значение 0, то вызов SFC80 не оказывает никакого воздействия. Если главное управляющее реле не установлено, то вызов SFC80 не оказывает никакого воздействия.

Параметр	Описание	Тип данных	Область памяти	Характеристика
N	INPUT	INT	I, Q, M, D, L, константа	Число сбрасываемых битов.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках.
SA	OUTPUT	POINTER	P	Указатель на первый сбрасываемый бит.

Информация об ошибках

Как оценивать информацию об ошибках из параметра RET_VAL, объясняется в главе 2. Эта глава содержит также общую информацию об ошибках SFC. SFC80 не дает в параметре RET_VAL какой-либо информации о конкретных ошибках.

13.5 Реализация генератора последовательностей с помощью SFB32 "DRUM"

Описание

SFB 32 "DRUM" реализует генератор последовательностей с максимальным числом шагов, равным 16. Номер первого шага задается параметром DSP, номер последнего шага – параметром LST_STEP.

На каждом шаге записываются все 16 выходных битов с OUT0 по OUT15 и выходной параметр OUT_WORD (в который объединены выходные биты). Выходной бит получает значение либо соответствующего бита заданного вами битового массива OUT_VAL, либо соответствующего выходного бита из предыдущего шага. Присваиваемое значение зависит от того, как вы назначите биты маски в параметре S_MASK (см. следующую таблицу).

SFB32 "DRUM" переходит к следующему шагу, когда на входе JOG появляется положительный фронт по отношению к предыдущему вызову SFB. Если этот SFB уже достиг последнего шага, то при появлении положительного фронта на входе JOG переменные Q и EOD устанавливаются, DCC получает значение 0, и SFB остается на последнем шаге до тех пор, пока на входе RESET не будет установлено значение 1.

Вы можете также назначить параметры так, что переход к следующему шагу будет зависеть от времени. Для этого вы должны установить параметр DRUM_EN в 1. Генератор последовательностей переходит к следующему шагу, когда:

- для текущего шага установлен бит события EVENTi и
- истекло запрограммированное для текущего шага время.
Это время задается как произведение базы времени DTBP и коэффициента времени, действующего на текущем шаге (из массива S_PRESET)

Замечание

Остающееся еще на текущем шаге время обработки (DCC) сокращается только тогда, когда установлен соответствующий бит события EVENTi.

Если при вызове SFB на входе RESET установлена 1, то генератор последовательностей переходит к шагу, номер которого вы присвоили входу DSP.

Замечание

Если вы установили 1 для DRUM_EN, то вы можете достичь следующей особой ситуации:

- деблокировки шагов, зависящей только от времени, выбрав $EVENTi = 1$, где $DSP \leq i \leq LST_STEP$.
- деблокировки шагов, зависящей только от событий, используя биты событий EVENTi при установке 0 на DTBP.

Вы можете также перейти в генераторе последовательностей к следующему шагу в любое время (даже если DRUM_EN=1) через вход JOG.

Когда блок вызывается в первый раз, вы должны установить 1 на входе RESET.

Если генератор последовательностей находится на последнем шаге (DSC имеет значение LST_STEP) и для этого шага истекло время исполнения, то выходы Q и EOD устанавливаются, а SFB остается на последнем шаге до тех пор, пока вы не установите 1 на входе RESET.

Таймер DRUM работает только в режимах STARTUP и RUN.

Операционная система сбрасывает SFB32 "DRUM" при холодном, но не при теплом рестарте. Если вы хотите инициализировать SFB32 "DRUM" после теплого рестарта, вызовите его в OB100 с RESET = 1.

Параметр	Описание	Тип данных	Область памяти	Характеристика
RESET	INPUT	BOOL	I, Q, M, D, L, константа	Уровень сигнала 1 сбрасывает генератор последовательностей. При первом вызове блока вы должны установить RESET в 1.
JOG	INPUT	BOOL	I, Q, M, D, L, константа	Нарастающий фронт (по отношению к последнему вызову SFB) переключает генератор последовательностей на следующий шаг, если только он уже не находится на последнем шаге. Следующий шаг разблокируется в зависимости от значения, которое вы назначили параметру DRUM_EN.
DRUM_EN	INPUT	BOOL	I, Q, M, D, L, константа	Управляющий параметр, который определяет, возможен ли переход к следующему шагу в зависимости от времени (1: зависящий от времени переход возможен).
LST_STEP	INPUT	BYTE	I, Q, M, D, L, константа	Номер последнего возможного шага. Возможные значения: 1 – 16
EVENT _i , $1 \leq i \leq 16$	INPUT	BOOL	I, Q, M, D, L, константа	Бит события номер i (относится к шагу i)
OUT _j , $0 \leq j \leq 15$	OUTPUT	BOOL	I, Q, M, D, L	Выходной бит номер j (идентичен биту номер j в OUT_WORD)
Q	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния, который указывает, истекло ли заданное вами время исполнения последнего шага.
OUT_WORD	OUTPUT	WORD	I, Q, M, D, L, P	Выходные биты, объединенные в одну переменную
ERR_CODE	OUTPUT	WORD	I, Q, M, D, L, P	Если во время исполнения SFB появляется ошибка, то ERR_CODE содержит информацию об ошибке.
JOG_HIS	VAR	BOOL	I, Q, M, D, L, константа	(Не имеет значения для пользователя: входной параметр JOG предыдущего вызова SFB)
EOD	VAR	BOOL	I, Q, M, D, L, константа	Идентично выходному параметру Q
DSP	VAR	BYTE	I, Q, M, D, L, P, константа	Номер первого шага. Возможные значения: 1 – 16.
DSC	VAR	BYTE	I, Q, M, D, L, P, константа	Номер текущего шага
DCC	VAR	DWORD	I, Q, M, D, L, P, константа	Остающееся еще на текущем шаге время исполнения в мс (имеет значение только тогда, когда DRUM_EN = 1 и соответствующий бит события = 1)

Параметр	Описание	Тип данных	Область памяти	Характеристика
DTBP	VAR	WORD	I, Q, M, D, L, P, константа	Действительная для всех шагов база времени в мс
PREV_TIME	VAR	DWORD	I, Q, M, D, L, константа	(Не имеет значения для пользователя: системное время предыдущего вызова SFB)
S_PRESET	VAR	ARRAY of WORD	I, Q, M, D, L, константа	Одномерный массив с коэффициентами времени для каждого шага. Рациональный выбор индексов: [1 to 16]. В этом случае S_PRESET [x] содержит коэффициент времени шага x.
OUT_VAL	VAR	ARRAY of BOOL	I, Q, M, D, L, константа	Двумерный массив с выводимыми на каждом шаге значениями, если они не замаскированы с помощью S_MASK. Рациональный выбор индексов: [1 to 16, 0 to 15]. В этом случае OUT_VAL [x, y] содержит значение, которое присваивается выходному биту OUTy на шаге x.
S_MASK	VAR	ARRAY of BOOL	I, Q, M, D, L, константа	Двумерный массив с битами маски для каждого шага. Рациональный выбор индексов: [1 to 16, 0 to 15]. В этом случае S_MASK [x, y] содержит бит маски для y-го выводимого значения на шаге x. Значение битов маски: <ul style="list-style-type: none"> 0: значение из предыдущего шага присваивается соответствующему выходному биту. 1: соответствующее значение из OUT_VAL присваивается соответствующему выходному биту.

Информация об ошибках

Если появляется одно из перечисленных в следующей таблице условий, то SFB 32 "DRUM" остается в текущем состоянии, а выход ERR_CODE устанавливается.

ERR_CODE (W#16#...)	Объяснение
0000	Нет ошибки
8081	Недопустимое значение для LST_STEP
8082	Недопустимое значение для DSC
8083	Недопустимое значение для DSP
8084	Произведение $DCC = DTBP * S_PRESET[DSC]$ превышает величину 2^{**32-1} (примерно 24,86 дня)

14 Системные функции для адресации модулей

14.1 Определение логического базового адреса модуля с помощью SFC5 "GADR_LGC"

Описание

Канал сигнального модуля определяет соответствующий слот модуля и относительный адрес пользовательских данных модуля. С помощью SFC5 "GADR_LGC" (convert geographical address to logical address [преобразовать географический адрес в логический]) вы можете получить соответствующий логический адрес модуля, то есть наименьший адрес входа или выхода.

Параметр	Описание	Тип данных	Область памяти	Характеристика
SUBNETID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор области: <ul style="list-style-type: none"> 0, если слот находится в стойке 0 (центральная стойка) или в стойках с 1 по 21 (стойки расширения). идентификатор master-устройства DP соответствующей системы децентрализованной периферии, если слот находится в устройстве децентрализованной периферии.
RACK	INPUT	WORD	I, Q, M, D, L, константа	<ul style="list-style-type: none"> Номер стойки, если идентификатор области равен 0. номер станции устройства децентрализованной периферии, если идентификатор области > 0.
SLOT	INPUT	WORD	I, Q, M, D, L, константа	Номер слота.
SUBSLOT	INPUT	BYTE	I, Q, M, D, L, константа	Слот субмодуля (если субмодуль не может подключаться, то здесь должен задаваться 0).
SUBADDR	INPUT	WORD	I, Q, M, D, L, константа	Смещение в адресной области пользовательских данных модуля
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках.
IOID	OUTPUT	BYTE	I, Q, M, D, L	Идентификатор области: V#16#54: периферийный вход (PI) V#16#55: периферийный выход (PQ) Если модуль смешанный, то SFC дает идентификатор области V#16#54. Если адреса равны, то SFC дает идентификатор V#16#54.
LADDR	OUTPUT	WORD	I, Q, M, D, L	Логический базовый адрес модуля.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8094	Никакая подсеть не была конфигурирована с указанным SUBNETID.
8095	Недопустимое значение параметра RACK.
8096	Недопустимое значение параметра SLOT.
8097	Недопустимое значение параметра SUBSLOT.
8098	Недопустимое значение параметра SUBADDR.
8099	Слот не конфигурирован.
809A	Субадрес выбранного слота не конфигурирован.

14.2 Определение слота модуля, соответствующего логическому адресу, с помощью SFC49 "LGC_GADR"

Описание

С помощью SFC 49 "LGC_GADR" (convert logical address to geographical address [преобразовать логический адрес в географический]) вы определяете соответствующий логическому адресу слот модуля, а также смещение в пространстве адресов пользовательских данных модуля.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор адресной области: B#16#54 = периферийный вход (PI) B#16#55 = периферийный выход (PQ) Если модуль смешанный, укажите идентификатор области самого младшего адреса. Если адреса одинаковы, укажите B#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический адрес. В случае смешанных модулей укажите младший из двух адресов.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибке
AREA	OUTPUT	BYTE	I, Q, M, D, L	Идентификатор области: он указывает, как должны интерпретироваться остающиеся выходные параметры.
RACK	OUTPUT	WORD	I, Q, M, D, L	Номер стойки.
SLOT	OUTPUT	WORD	I, Q, M, D, L	Номер слота
SUBADDR	OUTPUT	WORD	I, Q, M, D, L	Смещение в пространстве адресов пользовательских данных соответствующего модуля.

Выходной параметр AREA

Выходной параметр AREA указывает, как должны интерпретироваться выходные параметры RACK, SLOT и SUBADDR (см. следующую таблицу).

Значение AREA	Система	Значение RACK, SLOT и SUBADDR
0	S7-400	RACK: номер стойки SLOT: номер слота
1	S7-300	SUBADDR: разность между логическим адресом и логическим базовым адресом.
2	DP	RACK, SLOT и SUBADDR не имеют значения
3	Область S5 P	RACK: номер стойки
4	Область S5 O	SLOT: номер слота
5	Область S5 IM3	SUBADDR: адрес в области S5x
6	Область S5 IM4	

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8090	Заданный логический адрес недопустим или недопустимое значение параметра IOID

14.3 Определение всех логических адресов модуля с помощью SFC50 "RD_LGADR"

Описание

Вы исходите из одного логического адреса модуля. С помощью SFC 50 "RD_LGADR" (read module logical addresses [прочитать логические адреса модуля]) вы получаете все объявленные логические адреса этого модуля. Ранее вы уже присвоили адреса модулям с помощью STEP 7. SFC50 вводит полученные логические адреса в массив PEADDR или в массив PAADDR в порядке возрастания.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор области: В#16#54: периферийный вход (PI) В#16#55: периферийный выход (PQ)
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Один логический адрес
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибке
PEADDR	OUTPUT	ANY	I, Q, M, D, L	Массив для адресов PI, элементы массива должны иметь тип данных WORD.
PECOUNT	OUTPUT	INT	I, Q, M, D, L	Количество возвращенных адресов PI
PAADDR	OUTPUT	ANY	I, Q, M, D, L	Массив для адресов PQ, элементы массива должны иметь тип данных WORD.
PACOUNT	OUTPUT	INT	I, Q, M, D, L	Количество возвращенных адресов PQ

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8090	Заданный логический адрес недействителен или недопустимое значение для параметра IOID.
80A0	Ошибка в выходном параметре PEADDR: Тип данных элементов массива не WORD.
80A1	Ошибка в выходном параметре PAADDR: Тип данных элементов массива не WORD.
80A2	Ошибка в выходном параметре PEADDR: Заданный массив не смог принять все логические адреса.
80A3	Ошибка в выходном параметре PAADDR: Заданный массив не смог принять все логические адреса.

15 SFC для децентрализованной периферии

1.1 Запуск аппаратного прерывания в master-устройстве DP с помощью SFC7 "DP_PRAL"

Применимость

Вы можете использовать описанную в этом разделе SFC только тогда, когда вы в качестве интеллектуального slave-устройства используете CPU 315-2DP.

Описание

С помощью SFC7 "DP_PRAL" вы запускаете аппаратное прерывание в master-устройстве DP из программы пользователя в интеллектуальном slave-устройстве. Это прерывание запускает OB40 в master-устройстве DP.

Вы можете идентифицировать причину аппаратного прерывания, используя входной параметр AL_INFO. Этот идентификатор прерывания передается master-устройству DP, и вы можете оценивать идентификатор в OB40 (переменная OB40_POINT_ADDR) (см. раздел 1.6).

Запрошенное аппаратное прерывание однозначно определяется входными параметрами IOID и LADDR. Для каждой конфигурированной адресной области в памяти передачи вы можете в любое время запустить точно одно аппаратное прерывание.

Как работает SFC

SFC7 "DP_PRAL" работает асинхронно, другими словами, она выполняется на протяжении нескольких вызовов SFC. Запрос аппаратного прерывания запускают, вызывая SFC7 с REQ=1. Состояние задания отображается выходными параметрами RET_VAL и BUSY, см. раздел 2.2. Задание завершается тогда, когда завершается выполнение OB40 в master-устройстве DP.

Замечание

Если вы эксплуатируете slave-устройство DP в качестве стандартного slave-устройства, то задание завершается, как только диагностический кадр будет выбран master-устройством DP.

Идентификация задания

Входные параметры IOID и LADDR однозначно определяют задание.

Если вы вызвали SFC7 "DP_PRAL" в slave-устройстве DP и снова вызываете этот SFC прежде, чем master-устройство подтвердило запрошенное аппаратное прерывание, то способ реагирования SFC в значительной степени зависит от того, включает ли новый вызов то же самое задание; если параметры IOID и LADDR соответствуют заданию, которое еще не завершилось, то вызов SFC интерпретируется как продолженный вызов, независимо от значения параметра AL_INFO, и в RET_VAL вводится значение W#16#7002.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	REQ=1: Запуск аппаратного прерывания в master-устройстве DP, принадлежащем slave-устройству.
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор адресной области в памяти передачи (с точки зрения slave-устройства DP): B#16#54= Периферийный вход (PI) B#16#55= Периферийный выход (PQ) Идентификатором области, принадлежащей смешанному модулю, является младший из двух адресов. Если адреса одинаковые, то задайте B#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Начальный адрес адресной области в памяти передачи (с точки зрения slave-устройства DP). Если это область, принадлежащая смешанному модулю, то задайте младший из двух адресов.
AL_INFO	INPUT	DWORD	I, Q, M, D, L, константа	Идентификатор прерывания Он передается OB40, который запустится в master-устройстве DP (переменная OB40_POINT_ADDR). Если вы используете интеллектуальное slave-устройство с удаленным master-устройством, то вы должны в master-устройстве оценивать диагностический кадр. (См. /70/).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время выполнения функции происходит ошибка, то возвращаемое значение содержит код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Запущенное аппаратное прерывание еще не было подтверждено master-устройством DP.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Задание было выполнено без ошибок.
7000	Первый вызов с REQ=0. Запрос на аппаратное прерывание не активен; BUSY имеет значение 0.
7001	Первый вызов с REQ=1. Запрос на аппаратное прерывание уже передан master-устройству DP; BUSY имеет значение 1.
7002	Промежуточный вызов (REQ не имеет значения): запущенное аппаратное прерывание еще не было подтверждено master-устройством DP; BUSY имеет значение 1.
8090	Неправильный начальный адрес адресной области в памяти передачи.
8091	Прерывание заблокировано (блокировка, конфигурированная пользователем).
8093	Параметры IOID и LADDR адресуют модуль, который не способен к запросу аппаратного прерывания.
80C6	Децентрализованная периферия в настоящее время недоступна.

15.2 Синхронизация групп slave-устройств DP с помощью SFC11 "DPSYC_FR"

Описание

С помощью SFC11 "DPSYC_FR" вы можете синхронизировать одну или более групп slave-устройств DP.

Функция включает передачу соответствующим группам одной из указанных ниже команд управления или их комбинации:

- SYNC (одновременный вывод и «замораживание» состояний выходов в slave-устройствах DP)
- UNSYNC (отменяет команду управления SYNC)
- FREEZE («замораживание» состояний входов в slave-устройствах DP, считывание «замороженных» входов)
- UNFREEZE (отменяет команду управления FREEZE).

Требования

Прежде чем передавать перечисленные выше команды управления, нужно назначить группам slave-устройства DP, используя STEP 7 (см. [/231/](#)). Вы должны знать, какое slave-устройство DP какой группе с каким номером назначено, и знать реакции разных групп на SYNC/FREEZE.

Как работает SFC

SFC11 "DPSYC_FR" является асинхронной SFC, другими словами, ее выполнение занимает время нескольких вызовов SFC. Задание запускают, вызывая SFC11 с REQ=1.

Состояние задания отображается выходными параметрами RET_VAL и BUSY, см. также раздел 2.2.

Идентификация задания

Если вы запустили задание SYNC/FREEZE и снова вызвали SFC11 прежде, чем завершилось первое задание, то реакция SFC зависит от того, предназначен ли новый вызов для того же самого задания. Если входные параметры LADDR, GROUP и MODE согласуются, то вызов SFC интерпретируется как продолжающийся вызов.

Запись выходов модулей DP

Запись выходов модулей DP запускается следующим образом:

- посредством передачи команд на входы/выходы DP,
- посредством записи таблицы выходов образа процесса в модули (операционной системой в конце OB1 или путем вызова SFC27 "UPDAT_PO"),
- посредством вызова SFC15 "DPWR_DAT".

В нормальном режиме master-устройство DP циклически (в пределах цикла шины PROFIBUS DP) передает выходные байты на выходы slave-устройств DP.

Если вы хотите, чтобы определенные выходные данные (возможно, распределенные по нескольким slave-устройствам) точно в одно и то же самое время подавались на выходы к процессу, то вы можете передать соответствующему master-устройству DP команду SYNC, используя SFC11 "DPSYC_FR".

Что происходит в результате применения SYNC?

С помощью команды управления SYNC slave-устройства DP выбранных групп переключаются в режим синхронизации (Sync), другими словами, master-устройство DP передает текущие выходные данные и отдает приказ вовлеченным slave-устройствам DP заморозить свои выходы. С приемом следующих выходных кадров slave-устройства DP вводят выходные данные во внутренний буфер, а состояние выходов остается неизменным.

После каждой команды управления SYNC slave-устройства DP выбранных групп подают выходные данные из своего внутреннего буфера на выходы к процессу.

Выходы снова обновляются циклически только тогда, когда вы передадите команду управления UNSYNC, используя SFC11 "DPSYC_FR".

Чтение входов модулей DP

Входные данные модулей DP читаются следующим образом:

- посредством применения команд загрузки к входам/выходам DP,
- когда обновляется таблица входов образа процесса (операционной системой в начале OB1 или путем вызова SFC26 "UPDAT_PI"),
- посредством вызова SFC14 "DPRD_DAT".

В нормальном режиме master-устройство DP циклически (в пределах цикла шины PROFIBUS DP) принимает эти входные данные от своих slave-устройств DP и делает их доступными для CPU.

Если вы хотите, чтобы определенные входные данные (возможно, распределенные по нескольким slave-устройствам) считывались из процесса точно в одно и то же самое время, то передайте соответствующему master-устройству DP команду управления FREEZE, используя SFC11 "DPSYC_FR".

Что происходит в результате применения FREEZE?

С помощью команды управления FREEZE вовлеченные slave-устройства DP переключаются в режим «замораживания», другими словами, master-устройство DP дает команду slave-устройствам DP заморозить текущее состояние входов. Затем оно передает «замороженные» данные во входную область CPU.

После каждой команды управления FREEZE slave-устройства DP снова «замораживают» состояние своих входов.

Master-устройство DP снова циклически принимает текущее состояние входов только после того, как вы передадите команду управления UNFREEZE с помощью SFC11 "DPSYC_FR".

Непротиворечивость данных

Поскольку SFC11 "DPSYC_FR" функционирует ациклически и может прерываться классами более высокого приоритета, вам нужно убедиться, что при использовании SFC11 "DPSYC_FR" образы процесса совместимы с фактическими входными и выходными данными.

Это гарантируется, если вы соблюдаете следующие правила непротиворечивости:

- Определите подходящие разделы образа процесса для команд "Синхронизировать (SYNC) выходы" и "Заморозить (FREEZE) входы" (возможно только в S7-400). Вызовите SFC27 "UPDAT_PO" непосредственно перед первым вызовом задания SYNC. Вызовите SFC26 "UPDAT_PI" сразу после последнего вызова задания FREEZE.
- В качестве альтернативы: Используйте только прямой доступ к входам/выходам для выходов, включенных в задание SYNC, и для входов, включенных в задание FREEZE. Вы не должны записывать в эти выходы, когда активно задание SYNC, и не должны читать эти входы, когда активно задание FREEZE.

Использование SFC15 и SFC14

Если вы используете SFC15 "DPWR_DAT", то эта SFC должен завершиться прежде, чем вы передадите задание SYNC соответствующим выходам.

Если вы используете SFC14 "DPRD_DAT", то этот SFC должен завершиться прежде, чем вы передадите задание FREEZE соответствующим входам.

SFC11 "DPSYC_FR" и запуск

Только пользователь должен нести ответственность за передачу команд управления SYNC и FREEZE в ОБ запуска.

Если вы хотите, чтобы выходы одной или более групп были в режиме синхронизации, когда запускается программа пользователя, то вы должны инициализировать эти выходы во время запуска и полностью выполнить SFC11 "DPSYC_FR" с командой управления SYNC.

Если вы хотите, чтобы входы одной или более групп были в режиме FREEZE, когда запускается программа пользователя, то вы должны выполнить SFC11 "DPSYC_FR" с командой управления FREEZE полностью для этих входов во время запуска.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	Запускаемый уровнем параметр управления REQ=1: запуск задания SYNC/FREEZE
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический адрес master-устройства DP
GROUP	INPUT	BYTE	I, Q, M, D, L, константа	Выбор группы Бит 0 = 1: выбрана группа 1 Бит 1 = 1: выбрана группа 2 : Бит 7 = 1: выбрана группа 8 Вы можете выбрать несколько групп на задание. Значение B#16#0 не разрешено.

Параметр	Описание	Тип данных	Область памяти	Характеристика
MODE	INPUT	BYTE	I, Q, M, D, L, константа	<p>Идентификатор (ID) задания (кодирование в соответствии с EN 50 170, том 2, PROFIBUS)</p> <p>Бит 0: зарезервирован (значение 0)</p> <p>Бит 1: зарезервирован (значение 0)</p> <p>Бит 2 = 1: выполняется UNFREEZE = 0: нет значения</p> <p>Бит 3 = 1: выполняется FREEZE = 0: нет значения</p> <p>Бит 4 = 1: выполняется UNSYNC = 0: нет значения</p> <p>Бит 5 = 1: выполняется SYNC = 0: нет значения</p> <p>Бит 6: зарезервирован (значение 0)</p> <p>Бит 7: зарезервирован (значение 0)</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> в случае точно одного ID на задание: <ul style="list-style-type: none"> -B#16#04 (UNFREEZE) -B#16#08 (FREEZE) -B#16#10 (UNSYNC) -B#16#20 (SYNC) в случае более одного ID на задание: <ul style="list-style-type: none"> -B#16#14 (UNSYNC, UNFREEZE) -B#16#18 (UNSYNC, FREEZE) -B#16#24 (SYNC, UNFREEZE) -B#16#28 (SYNC, FREEZE)
RET_VAL	OUTPUT	INT	I, Q, M, D, L	<p>Если в то время, когда функция активна, происходит ошибка, то возвращаемое значение содержит код ошибки.</p> <p>Вы должны оценивать RET_VAL каждый раз после того, как блок выполнен.</p>
BUSY	OUTPUT	BOOL	I, Q, M, D, L	<p>BUSY=1:</p> <p>Задание SYNC/FREEZE еще не завершилось.</p>

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Задание завершилось без ошибок.
7000	Первый вызов с REQ =0. Задание, указанное с помощью LADDR, GROUP и MODE, не активно; BUSY имеет значение 0.
7001	Первый вызов с REQ=1. Задание, указанное с помощью LADDR, GROUP и MODE, было запущено; BUSY имеет значение 1.
7002	Промежуточный вызов (REQ не имеет значения). Активизированное задание SYNC/FREEZE все еще активно; BUSY имеет значение 1.
8090	Модуль, выбранный с помощью LADDR, не является master-устройством DP.
8093	Этот SFC не разрешен для модуля, выбранного с помощью LADDR (конфигурация или версия master-устройства DP).
8094	Неверный параметр GROUP.
8095	Неверный параметр MODE.
80B0	Группа, выбранная с помощью GROUP не конфигурирована.
80B1	Группа, выбранная с помощью GROUP не назначена этому CPU.
80B2	Задание SYNC, указанное с помощью MODE, не разрешено в группе, выбранной с помощью GROUP.
80B3	Задание FREEZE, указанное с помощью MODE, не разрешено в группе, выбранной с помощью GROUP.
80C2	Временная нехватка ресурсов в master-устройстве DP: Master-устройство DP в настоящее время обрабатывает максимальное для CPU количество заданий.
80C3	Это задание SYNC/UNSYNC не может быть активизировано в настоящее время, так как в любой момент времени может быть запущено только одно задание SYNC/UNSYNC. Проверьте вашу пользовательскую программу.
80C4	Это задание FREEZE/UNFREEZE не может быть активизировано в настоящее время, так как в любой момент времени может быть запущено только одно задание FREEZE/UNFREEZE. Проверьте вашу пользовательскую программу.
80C5	Децентрализованная периферия недоступна: Отказ подсистемы DP
80C6	Задание прервано из-за отключения входов/выходов посредством центральным процессором.
80C7	Задание прервано из-за теплого или холодного рестарта в master-устройстве DP.

15.3 Чтение диагностических данных slave-устройства DP (диагностика slave-устройства) с помощью SFC 13 "DPNRM_DG"

Диагностика slave-устройства

Каждое slave-устройство DP предоставляет диагностические данные, структурированные в соответствии со стандартом EN 50 170 Volume 2, PROFIBUS. Для считывания этих диагностических данных вам нужна SFC 13 "DPNRM_DG".

Принципиальная структура диагностических данных slave-устройства представлена в следующей таблице, а дополнительную информацию вы найдете в руководствах по slave-устройствам DP.

Байт	Значение
0	Состояние станции 1
1	Состояние станции 2
2	Состояние станции 3
3	Номер master-станции
4	Идентификатор изготовителя (старший байт)
5	Идентификатор изготовителя (младший байт)
6 ...	Дополнительная диагностическая информация, относящаяся к slave-устройству

Описание

С помощью SFC 13 "DPNRM_DG" (read diagnosis data of a DP-slave [прочитать диагностические данные slave-устройства DP]) вы читаете текущие диагностические данные slave-устройства DP в формате, определенном стандартом EN 50 170 Volume 2, PROFIBUS. Прочитанные данные после безошибочной передачи вводятся в целевую область, указанную в параметре RECORD.

Вы запускаете задание на чтение, присваивая 1 параметру REQ в вызове SFC13.

Принцип действия

Задание на чтение выполняется асинхронно, т.е. может распространяться на несколько вызовов SFC 13. Состояние задание отображается выходными параметрами RET_VAL и BUSY, см. раздел 2.2.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	REQ=1: запрос на чтение
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Спроектированный диагностический адрес slave-устройства DP.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если ошибка происходит, когда эта функция активна, то возвращаемое значение содержит код ошибки. Если ошибок нет, то длина фактически переданных данных вводится в RET_VAL.

RECORD	OUTPUT	ANY	I, Q, M, D, L	Целевая область для прочитанных диагностических данных. Допустимым является только тип данных BYTE. Минимальная длина считываемой записи данных или целевой области равна 6. Максимальная длина отправляемой записи данных равна 240. Стандартные slave-устройства могут предоставить более 240 байтов диагностических данных (максимум до 244 байтов). В этом случае первые 240 байтов передаются в целевую область, и в этих данных устанавливается бит переполнения.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Задание на чтение еще не завершено.

Входной параметр RECORD

CPU оценивает фактическую длину прочитанных диагностических данных следующим образом:

- Если длина, указанная для RECORD, меньше, чем количество доставленных байтов данных, то данные отбрасываются, и в RET_VAL записывается соответствующий код ошибки.
- Если длина, указанная для RECORD, больше или равна количеству доставленных байтов данных, то данные принимаются в целевую область, и в RET_VAL записывается фактическая длина в виде положительного значения.

Замечание

Вы должны обеспечить, чтобы все фактические параметры RECORD совпадали во всех вызовах, относящихся к заданию. Задание однозначно идентифицируется входным параметром LADDR.

Стандартные slave-устройства с более чем 240 байтами диагностических данных

В случае стандартных slave-устройств, в которых количество стандартных диагностических данных находится в диапазоне от 241 до 244 байтов примите во внимание следующие моменты:

- Если длина, указанная для RECORD, меньше 240 байтов, то данные отбрасываются, и соответствующая информация об ошибке вносится в RET_VAL.
- Если длина, указанная для RECORD, больше или равна 240 байтам, то первые 240 байтов стандартных диагностических данных передаются в целевую область, а в данных устанавливается бит переполнения.

Выходной параметр RET_VAL

- Если при исполнении функции происходит ошибка, то возвращаемое значение содержит код ошибки.
- Если при передаче данных не происходит ошибок, то RET_VAL содержит длину прочитанных данных в байтах в виде положительного числа.

Информация об ошибках

Как оценивать информацию об ошибках из параметра RET_VAL, объясняется в главе 2. Эта глава содержит также общую информацию об ошибках SFC. Специфическая для SFC 13 информация об ошибках является подмножеством информации об ошибках для SFC 59 "RD_REC", см. Таблица 7-2 Специфическая информация об ошибках для SFC58 "WR_REC" и SFC59 "RD_REC"

Системные ресурсы для S7-400

При вызове SFC 13 "DPNRM_DG" для задания, не обрабатываемого в данный момент времени, в S7-400 занимают ресурсы CPU (место в памяти). Вы можете вызывать SFC 13 через короткие промежутки времени подряд для нескольких slave-устройств DP при условии, что вы не превышаете максимальное для вашего CPU количество "одновременно" активных заданий SFC 13. Это максимальное количество вы найдете в [101](#).

Если вы активизируете несколько заданий "одновременно", то все задания будут выполнены, не влияя друг на друга.

Если вы достигнете ограничения системных ресурсов, то вы получите сообщение об этом в RET_VAL. В этом случае повторите задание.

15.4 Чтение непротиворечивых данных стандартного slave-устройства DP с помощью SFC14 "DPRD_DAT"

Определение: непротиворечивые данные

Данные, которые связаны друг с другом и не допускают разъединения, известны как непротиворечивые данные. Например, значения из аналоговых модулей всегда должны обрабатываться как непротиворечивые данные, иначе говоря, значение аналогового модуля не должно искажаться в результате считывания его в два разных момента времени.

Назначение SFC14

Вам нужен SFC 14 "DPRD_DAT", так как с помощью команд загрузки, которые выполняют доступ к периферии или к таблице входов образа процесса, вы можете считывать максимум четыре последовательных байта.

Описание

С помощью SFC14 "DPRD_DAT" (read consistent data of a DP standard slave) вы считываете непротиворечивые данные стандартного slave-устройства DP. Длина должна составлять три байта или более четырех байтов, причем максимальная длина фиксирована для каждого конкретного CPU. Максимальную длину вы найдете в технических данных своего CPU. Если при передаче данных не было ошибки, то прочитанные данные записываются в целевую область, определяемую параметром RECORD.

Эта целевая область должна иметь ту же самую длину, которую вы с помощью STEP 7 спроектировали для выбранного модуля.

Если вы читаете из стандартного slave-устройства DP, имеющего модульную конструкцию, или с несколькими идентификаторами DP, то вы можете обратиться к данным только одного модуля или идентификатора DP на каждый вызов SFC14, указывая спроектированный начальный адрес.

Параметр	Описание	Тип данных	Область памяти	Характеристика
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Спроектированный начальный адрес из области входов (I) модуля, из которой данные будут считываться.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если ошибка происходит, когда эта функция активна, то возвращаемое значение содержит код ошибки.
RECORD	OUTPUT	ANY	I, Q, M, D, L	Целевая область для прочитанных данных пользователя. Она должна быть точно такой же длины, какую вы спроектировали с помощью STEP 7 для выбранного модуля. Допустимым является только тип данных BYTE.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибки не было.
8090	Вы не спроектировали модуль для заданного логического базового адреса или не учли ограничение на длину непротиворечивых данных.
8092	В ссылке ANY указан тип, отличный от BYTE.
9093	Для заданного в LADDR логического адреса не существует модуля DP, из которого вы можете считывать непротиворечивые данные.
80A0	Выбранный модуль неисправен.
80B0	Отказ slave-устройства на внешнем интерфейсном модуле DP.
80B1	Длина заданной целевой области не равна спроектированной с помощью STEP 7 длине данных пользователя.
80B2	Системная ошибка в случае внешнего интерфейсного модуля DP.
80B3	Системная ошибка в случае внешнего интерфейсного модуля DP.
80C0	Системная ошибка в случае внешнего интерфейсного модуля DP.
80C2	Системная ошибка в случае внешнего интерфейсного модуля DP.
80Fx	Системная ошибка в случае внешнего интерфейсного модуля DP.
87ху	Системная ошибка в случае внешнего интерфейсного модуля DP.
808х	Системная ошибка в случае внешнего интерфейсного модуля DP.

15.5 Запись непротиворечивых данных в стандартное slave-устройство DP с помощью SFC15 "DPWR_DAT"

Определение: непротиворечивые данные

Данные, которые связаны друг с другом и не допускают разъединения, известны как непротиворечивые данные. Например, значения из аналоговых модулей всегда должны обрабатываться как непротиворечивые данные, иначе говоря, значение аналогового модуля не должно искажаться в результате считывания его в два разных момента времени.

Назначение SFC15

Вам нужен SFC15 "DPWR_DAT", так как с помощью команд передачи, которые выполняют доступ к периферии или к таблице выходов образа процесса, вы можете записать максимум четыре последовательных байта

Описание

С помощью SFC15 "DPWR_DAT" вы непротиворечиво передаете указанные в RECORD данные адресованному стандартному slave-устройству DP. Длина передаваемых данных должна составлять три байта или более четырех байтов, причем максимальная длина фиксирована для каждого конкретного CPU. Вы найдете эту информацию в технических данных своего CPU. Данные передаются синхронно, т.е. по окончании SFC задание на запись тоже завершается.

Исходная область должна иметь ту же самую длину, которую вы спроектировали с помощью STEP 7 для выбранного модуля.

Если стандартное slave-устройство DP имеет модульную конструкцию, вы можете получить доступ только к одному модулю slave-устройства DP.

Параметр	Описание	Тип данных	Область памяти	Характеристика
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Спроектированный начальный адрес из области выходов модуля, в который данные будут записываться.
RECORD	INPUT	ANY	I, Q, M, D, L	Исходная область для записываемых данных пользователя. Она должна быть точно такой же длины, какую вы с спроектировали помощью STEP 7 для выбранного модуля. Допустимым является только тип данных BYTE.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если ошибка происходит, когда эта функция активна, то возвращаемое значение содержит код ошибки.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибки не было.
808x	Системная ошибка в случае внешнего интерфейсного модуля DP.
8090	Для заданного логического базового адреса вы не спроектировали модуль или не учли ограничение на длину непротиворечивых данных.
8092	В ссылке ANY указан тип, отличный от BYTE.
9093	Для заданного в LADDR логического адреса не существует модуля DP, в который вы можете записывать непротиворечивые данные.
80A1	Выбранный модуль неисправен.
80B0	Отказ slave-устройства на внешнем интерфейсном модуле DP.
80B1	Длина заданной исходной области не равна спроектированной с помощью STEP 7 длине данных пользователя.
80B2	Системная ошибка в случае внешнего интерфейсного модуля DP.
80B3	Системная ошибка в случае внешнего интерфейсного модуля DP.
80C1	Данные предыдущего задания на запись в модуле еще не обработаны модулем.
80C2	Системная ошибка в случае внешнего интерфейсного модуля DP.
80Fx	Системная ошибка в случае внешнего интерфейсного модуля DP.
85ху	Системная ошибка в случае внешнего интерфейсного модуля DP.

16 SFC для связи с помощью глобальных данных

16.1 Передача GD-пакета с помощью SFC 60 "GD_SND"

Описание

С помощью SFC 60 "GD_SND" (global data send [передача глобальных данных]) данные пакета глобальных данных (GD-пакета) собираются и затем передаются по маршруту, указанному в GD-пакете. Перед этим GD-пакет должен быть спроектирован с помощью STEP 7.

SFC60 "GD_SND" может быть вызван любом месте программы пользователя. Вызовы SFC60 не влияют на скорость сканирования, а также сбор и передачу данных системой в точке контроля цикла.

Возможность прерывания

SFC60 "GD_SND" может прерываться более высокими классами приоритета. Возможно также, что SFC60 снова вызывается для того же GD-пакета в более высоком классе приоритета.

Тогда данные собираются и передаются в более высоком классе приоритета. Когда программа возвращается в прерванную SFC, последняя немедленно завершается, а уже собранные данные отбрасываются.

Эта процедура означает, что при обработке самого высокого класса приоритета передаются непротиворечивые данные (в смысле непротиворечивости, определенной для глобальных данных).

Непротиворечивость данных в GD

Для обеспечения непротиворечивости данных, собранных из различных областей памяти, и посланных данных применяются следующие правила.

Непротиворечивыми являются:

- данные простых типов (бит, байт, слово и двойное слово)
- массив данных типа байт, слово и двойное слово длиной, не превышающей максимума, зависящего от конкретного CPU.

Обеспечение непротиворечивости для всего GD-пакета

GD-пакет в CPU, передающем данные, имеет структуру, которая не гарантирует автоматически, что собранные данные будут непротиворечивыми. Это, например, имеет место, когда пакет состоит из массива байтов, и количество байтов превышает максимальную длину для конкретного CPU.

Если, однако, вам нужна непротиворечивость для всего GD-пакета, то следуйте в своей программе процедуре, описанной ниже:

- Заблокируйте или задержите появление более приоритетных прерываний и асинхронных ошибок, вызвав SFC 39 "DIS_IRT" или SFC 41 "DIS_AIRT".
- Вызовите SFC60 "GD_SND".
- Снова разблокируйте более приоритетные прерывания и асинхронные ошибки, вызвав SFC 40 "EN_IRT" или SFC 42 "EN_AIRT".

Параметр	Описание	Тип данных	Область памяти	Характеристика
CIRCLE_ID	INPUT	BYTE	I, Q, M, D, L, константа	Номер GD-контура, в котором находится передаваемый GD-пакет. Этот номер задается средствами STEP 7 при проектировании глобальных данных. Допустимые значения: от 1 до 16. Максимально возможное количество GD-контуров можно найти в технических данных вашего CPU.
BLOCK_ID	INPUT	BYTE	I, Q, M, D, L, константа	Номер передаваемого GD-пакета в выбранном GD-контуре. Этот номер задается средствами STEP 7 при проектировании глобальных данных. Допустимые значения: от 1 до 3. Максимально возможное количество GD-контуров можно найти в технических данных вашего CPU.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8081	GD-пакет, выбранный с помощью параметров CIRCLE_ID и BLOCK_ID, не спроектирован.
8082	Недопустимое значение для параметра CIRCLE_ID или BLOCK_ID или для обоих параметров.
8083	При выполнении SFC произошла ошибка. Вид ошибки записан в переменную, спроектированную для статусной информации. Он может использоваться вашей программой.
8084	Выполнение SFC было закончено досрочно, так как SFC 60 была вызвана еще раз для того же самого GD-пакета в более высоком классе приоритета (см. "Возможность прерывания").
8085	При записи статусной информации в спроектированную для этого переменную произошла ошибка.

Замечание

Вам следует после каждого вызова SFC 60 "GD_SND" оценивать состояние соответствующего GD-пакета и, при необходимости, сбрасывать его.

16.2 Извлечение принятого GD-пакета с помощью SFC61 "GD_RCV"

Описание

С помощью SFC61 "GD_RCV" (global data receive [прием глобальных данных]) из поступающего кадра GD извлекаются данные точно для одного пакета GD и вводятся в принимаемый пакет GD. Он уже должен быть спроектирован с помощью STEP 7.

SFC61 "GD_SND" может вызываться в любой точке программы пользователя.

Вызовы SFC61 не влияют на скорость сканирования и выборку данных системой в контрольной точке цикла.

Возможность прерывания

SFC61 может прерываться классами более высокого приоритета, но только так, чтобы оставалась гарантированной непротиворечивость данных, определенная для глобальных данных. Если обработка функции прерывается, то возможно, что SFC61 вызывается снова для того же самого GD-пакета в классе более высокого приоритета.

Тогда данные вводятся в принимаемый GD-пакет в классе более высокого приоритета. Когда программа возвращается к прерванной SFC, она сразу завершается.

Непротиворечивость данных в GD

Для обеспечения непротиворечивости данных, вводимых в различные области памяти, применяются следующие правила.

Следующие данные непротиворечивы:

- простые типы данных (бит, байт, слово и двойное слово)
- массивы данных типа байт, слово и двойное слово вплоть до максимальной длины, определенной для принимающего CPU.

Обеспечение непротиворечивости всего пакета GD

GD-пакет в принимающем CPU имеет структуру, которая автоматически не гарантирует, что его данные происходят из одного и того же кадра. Например, это имеет место, когда пакет состоит из трех элементов GD.

Если, однако, вы требуете непротиворечивости для всего GD-пакета, то следуйте в вашей программе процедуре, описанной ниже:

- Запретите или задержите возникновение прерываний более высокого приоритета и асинхронных ошибок, вызывая SFC39 "DIS_IRT" или SFC41 "DIS_AIRT".
- Вызовите SFC60 "GD_SND".
- Разрешите прерывания более высокого приоритета и асинхронные ошибки, вызывая SFC40 "EN_IRT" или SFC42 "EN_AIRT".

Параметр	Описание	Тип данных	Область памяти	Характеристика
CIRCLE_ID	INPUT	BYTE	I, Q, M, D, L, константа	Номер GD-контура, в который будет вводиться поступающий GD-пакет. Этот номер задается во время проектирования глобальных данных с помощью STEP 7. Разрешенные значения: от 1 до 16. Максимальное число возможных GD-контуров можно найти в технических данных вашего CPU.
BLOCK_ID	INPUT	BYTE	I, Q, M, D, L, константа	Номер GD-пакета в выбранном GD-контуре, в который будут вводиться поступающие данные. Этот номер задается во время проектирования глобальных данных с помощью STEP 7. Разрешенные значения: от 1 до 3. Максимальное число возможных GD-контуров можно найти в технических данных вашего CPU.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8081	GD-пакет, выбранный с помощью параметров CIRCLE_ID и BLOCK_ID, не спроектирован.
8082	Недопустимое значение параметров CIRCLE_ID или BLOCK_ID или обоих параметров.
8083	Во время выполнения SFC произошла ошибка. Тип ошибки вводится в переменную, спроектированную для информации о состоянии. Он может оцениваться вашей программой.
8084	Выполнение SFC было преждевременно завершено, потому что SFC61 был вызван снова для того же самого GD-пакета в классе более высокого приоритета (см. "Возможность прерывания").
8085	Ошибка произошла при вводе информации о состоянии в спроектированную переменную.

Примечание

После каждого вызова SFC60 вы должны оценивать состояние соответствующего GD-пакета и в случае необходимости сбрасывать его.

17 Коммуникационные SFB для спроектированных соединений S7

17.1 Классификация коммуникационных SFB для спроектированных соединений S7

Классификация

Коммуникационные SFB для спроектированных соединений S7 можно разбить на следующие категории:

- коммуникационные SFB для обмена данными
- коммуникационные SFB для управления режимом работы
- коммуникационные SFB для запроса режима работы.

Коммуникационные SFB для обмена данными

Коммуникационные SFB для обмена данными используются для обмена данными между двумя коммуникационными партнерами. Если SFB существует только в локальном модуле, то говорят об одностороннем обмене данными. Если SFB существует как в локальном, так и в удаленном модуле, то говорят о двустороннем обмене данными.

Имя SFB	Краткое описание	Обмен данными
USEND/URCV	Некоординированный обмен данными при помощи передающего и принимающего CFB	Двусторонний
BSEND/BRCV	Обмен блоками данных длиной ¹⁾ до 64 килобайтов между передающим CFB и принимающим CFB	Двусторонний
GET	Чтение данных из удаленного устройства (удаленное устройство может быть в режиме RUN или STOP)	Односторонний
PUT	Запись данных в удаленное устройство (удаленное устройство может быть в режиме RUN или STOP)	Односторонний
PRINT	Передача данных, включая команды форматирования для принтера	Односторонний

- 1) В отличие от всех других коммуникационных SFB для спроектированных соединений максимальная длина передаваемых блоков данных не зависит от используемого CPU.

Коммуникационные SFB для управления режимом работы

При помощи коммуникационных SFB для управления режимом работы вы можете управлять режимом работы удаленного устройства.

Обмен данными при помощи коммуникационных SFB для управления режимом работы является односторонним.

Имя SFB	Краткое описание
START	Инициирование теплого или холодного рестарта в удаленном устройстве.
STOP	Инициирование режима STOP в удаленном устройстве.
RESUME	Инициирование горячего рестарта в удаленном устройстве.

Коммуникационные SFB для запроса режима работы

При помощи коммуникационных SFB для запроса режима работы вы можете получать информацию о режиме работы удаленного устройства.

Обмен данными при помощи SFB "STATUS" является односторонним, а при помощи SFB "USTATUS" – двусторонним.

Имя SFB	Краткое описание
STATUS	Активный запрос режима работы удаленного устройства.
USTATUS	Незапрашиваемый отчет о состоянии удаленного устройства, когда его режим работы изменяется.

SFC62 "CONTROL"

При помощи SFC62 "CONTROL" вы получаете состояние соединения, принадлежащего экземпляру коммуникационного SFB.

Типовая программа

Типовая программа для коммуникационных SFB для спроектированных соединений поставляется с пакетом STEP 7. Эта программа находится в папке step7\examples\com_sfb.

17.2 Классификация параметров коммуникационных SFB для спроектированных соединений S7

Классификация

Параметры коммуникационных SFB для спроектированных соединений S7 можно разбить на следующие пять категорий в соответствии с их функциями:

- параметры управления (используются для активизации блока)
- параметры адресации (используются для адресации удаленного коммуникационного партнера)
- параметры передачи (указание областей данных, которые должны передаваться удаленному партнеру)
- параметры приема (указание областей данных, куда будут вводиться данные, принимаемые от удаленных партнеров)
- параметры состояния (используются для проверки того, завершил ли блок свою задачу без ошибки, либо для анализа любых произошедших ошибок).

Параметры управления

Обмен данными будет активизироваться только тогда, когда соответствующие параметры управления имеют определенное состояние сигнала в момент вызова SFB или когда состояние сигнала подверглось определенному изменению, начиная с момента предыдущего вызова SFB. В соответствии с этими двумя методами запуска параметры относятся к параметрам управления, которые запускаются уровнем сигнала и которые запускаются фронтом сигнала.

Параметр	Значение	Отправитель / получатель	Активизация функции при	Характеристика
REQ	Запрос	Отправитель задания	нарастающем фронте сигнала (по сравнению с предыдущим вызовом SFB)	Активизирует обмен данными (если другие условия удовлетворены).
R	Сброс	Отправитель задания	нарастающем фронте сигнала (по сравнению с предыдущим вызовом SFB)	Прерывает обмен данными, активный в текущий момент времени.
EN_R	Разрешение приема	Получатель задания	уровне 1	Передает сообщение "готов к приему".

Параметры адресации

Параметр	Характеристика	Обратить внимание
ID	Ссылка на локальное описание соединения. ID назначается, когда соединение проектируется.	<ul style="list-style-type: none"> ID нужно задавать в форме W#16#wxyz. При двусторонней связи параметр ID существует как в SFB на передающей стороне, так и в SFB на принимающей стороне. При односторонней связи SFB в удаленном модуле не существует. Для описания соединения вы можете выбрать одну из двух следующих процедур: <ul style="list-style-type: none"> -Хранение описания соединения в локальном и удаленном партнерах. Преимущество: При такой конфигурации ресурсы для коммуникационного взаимодействия резервируются постоянно. -Хранение описания соединения только в локальном модуле.
R_ID	При двусторонней связи вы указываете, что передающий и приемный SFB связаны друг с другом, с помощью R_ID: SFB, связанные друг с другом, имеют одинаковое значение R_ID. Это позволяет нескольким парам SFB обмениваться сообщениями по одному логическому соединению.	<ul style="list-style-type: none"> R_ID нужно задавать в форме DW#16#wxyzWXYZ. Пары блоков логического соединения, заданного посредством R_ID, должны быть уникальными для этого соединения. R_ID имеет другое значение при связи «точка-точка», использующей CP 441.
PI_NAME	В блоках SFB19 "START", SFB20 "STOP" и SFB21 "RESUME" указатель на область памяти, в которой расположено имя запускаемой программы (код ASCII).	В S7 разрешаются любые символьные строки, начинающиеся с P_PROGRAM.

Замечание

Параметры адресации ID и R_ID оцениваются только во время первого вызова блока (фактические параметры или предопределенные значения из экземпляра). Поэтому первый вызов задает коммуникационную связь (соединение) с удаленным партнером до следующего теплого или холодного рестарта.

Параметры состояния

С помощью параметров состояния вы контролируете, правильно ли завершил блок свою задачу или он все еще активен. Параметры состояния также отображают ошибки.

Замечание

Параметры состояния имеют силу только для одного цикла, другими словами, с первой команды после вызова SFB до следующего вызова SFB. Вследствие этого вы должны оценивать эти параметры **каждый раз**, когда блок завершается.

Параметр	Тип данных	Отправитель / Получатель	Характеристика
DONE	BOOL	Отправитель	0: Задание еще не было запущено или все еще активно. 1: Задание было выполнено без ошибок.
NDR	BOOL	Получатель	0: Задание еще не было запущено или все еще выполняется. 1: Задание завершилось успешно.
ERROR	BOOL	Отправитель и получатель	Отображение ошибки: ERROR Значение 0 Значением STATUS является - 0000H: ни предупреждения, ни ошибки - не равно 0000H: предупреждение. STATUS дает подробную информацию.
STATUS	WORD	Отправитель и получатель	1 Произошла ошибка. STATUS дает подробную информацию о типе ошибки.

Параметры передачи и приема

Если вы используете не все параметры передачи или приема SFB, то первый неиспользуемый параметр должен быть указателем NIL (см. /232/), а используемые параметры должны располагаться друг за другом без каких-либо промежутков.

Замечание

Во время первого вызова указатель ANY задает максимальное количество данных пользователя, которые могут быть переданы для задания. Чтобы гарантировать непротиворечивость данных, в рабочей памяти CPU создается буфер обмена данными. Он занимает до 480 байтов рабочей памяти. Тогда желательно запрограммировать первый вызов в ОВ теплого или холодного рестарта, если блок с вызовом SFB не загружается позже, когда CPU находится в режиме RUN.

Во время последовательных вызовов вы можете передавать/принимать любое количество данных, однако не больше, чем во время первого вызова. SFB BSEND и BRCV являются исключением из этого правила. С помощью них вы можете передавать до 64 Килобайтов на задание (см. 17.5 и 17.6

Прием сегментированных данных с помощью SFB13 "BRCV").

В случае коммуникационных SFB для двустороннего обмена данными:

- Число используемых параметров SD_i и RD_i должно совпадать на передающей и принимающей сторонах.
- Типы данных для параметров SD_i и RD_i, относящихся друг другу, должны совпадать на передающей и принимающей сторонах.
- Количество данных, передаваемых в соответствии с параметром SD_i, не должно превышать область, сделанную доступной соответствующим параметром RD_i.

ERROR = 1 и STATUS = 4 показывают, что вы нарушили вышеупомянутые правила.

В случае SFB USEND, URCV, GET и PUT количество передаваемых данных, не должно превышать максимальную длину данных пользователя. Эта максимальная длина данных пользователя зависит от того, является ли удаленным партнером S7-300 или S7-400. Это показано в следующей таблице.

SFB	S7-300	S7-400
USEND/URCV	-	440 ¹⁾
PUT/GET	160 ¹⁾	400 ¹⁾

1) Технически возможная длина данных пользователя может отличаться от значений, указанных выше. Она зависит от числа используемых областей. Для получения более подробной информации обратитесь к службе поддержки клиентов (Customer Support).

Непротиворечивость данных

Доступ к данным через операционную систему является пакетно-ориентированным. Все данные такого пакета связаны друг с другом и во время передачи данных не могут изменяться другими разделами программы. Это известно как непротиворечивость данных.

С точки зрения непротиворечивости "более слабый" коммуникационный партнер определяет результирующую длину непротиворечивых данных для передачи данных.

Следующие данные непротиворечивы:

- Простые типы данных (бит, байт, слово, и двойное слово).

Массив байтового типа данных вплоть до максимальной длины для конкретного CPU (см. [/71/](#) , [/101/](#)).

17.3 Некоординированная передача данных с помощью SFB8 "USEND"

Описание

SFB8 "USEND" передает данные блоку SFB типа "URCV" удаленного партнера (параметр R_ID должен быть одинаковым для обоих SFB). Данные передаются при нарастающем фронте сигнала на управляющем входе REQ. Функция выполняется без координации с SFB партнера. Передаваемые данные указываются параметрами SD_1 – SD_4, но не обязательно использовать все четыре параметра передачи. Однако вам нужно убедиться в том, что области, определенные параметрами SD_i и RD_i, $1 \leq i \leq 4$, совпадают друг с другом по номеру, длине и типу данных (RD_i принадлежит соответствующему SFB "URCV" партнера). Успешное завершение передачи отображается значением 1 в параметре состояния DONE.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L,	Параметр управления «запрос».
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
R_ID	INPUT	DWORD	I, Q, M, D, L, константа	Параметр адресации R_ID.
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: данные переданы.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
SD_i, $1 \leq i \leq 4$	IN_OUT	ANY	I, Q, M, D, T, C	Указатель на i-ю область передаваемых данных. Разрешенные типы данных BOOL (не разрешен битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER

Информация об ошибках

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не может вступить в силу, так как предыдущее задание еще не закончено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none"> не загружено описание соединения (локальное или удаленное) разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	4	Ошибка в указателях областей передаваемых данных SD_i, включая длину данных или тип данных.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB8 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	18	R_ID уже существует в идентификаторе (ID) соединения.
1	20	Не хватает доступной рабочей памяти.

Непротиворечивость данных

Чтобы гарантировать непротиворечивость данных, вы можете записывать снова в область передачи SD_i, используемую в текущий момент времени, только после того, как операция передачи завершится. Это имеет место тогда, когда значение параметра состояния DONE станет равным 1.

17.4 Некоординированный прием данных с помощью SFB9 "URCV"

Описание

SFB9 "URCV" асинхронно принимает данные от SFB типа "USEND" удаленного партнера (параметр R_ID должен быть одинаковым в обоих SFB). Если при вызове блока на управляющий вход EN_R подается значение 1, то принимаемые данные копируются в спроектированные области приема. Эти области данных указываются параметрами RD_1 – RD_4. Помните, что области, определенные параметрами RD_i ($1 \leq i \leq 4$), должны по номеру и типу данных совпадать с областями SFB "USEND", определенным параметрами SD_i ($1 \leq i \leq 4$). Когда блок вызывается впервые, создается "почтовый ящик для приема". При всех дальнейших вызовах принимаемые данные должны вестись в этот приемный почтовый ящик. Завершение задания на копирование отображается значением 1 в параметре состояния NDR.

Параметр	Описание	Тип данных	Область памяти	Характеристика
EN_R	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "разрешение на прием".
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
R_ID	INPUT	DWORD	I, Q, M, D, L, константа	Параметр адресации R_ID.
NDR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния NDR: новые данные приняты.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
RD_i, $1 \leq i \leq 4$	IN_OUT	ANY	I, Q, M, D, T, C	Указатель на i-ю область приема. Разрешены типы данных BOOL (не разрешен битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER.

Информация об ошибках

ERROR	STATUS (десятичное число)	Объяснение
0	9	Предупреждение о потере информации: более старые принятые данные заменяются более новыми принятыми данными.
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none"> не загружено описание соединения (локальное или удаленное) разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	4	Ошибки в указателях областей приема RD_i, включая длину данных или тип данных.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB9 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	18	R_ID уже существует в идентификаторе (ID) соединения.
1	19	Соответствующий SFB "USEND" передает данные быстрее, чем они могут копироваться в области приема посредством SFB "URCV".
1	20	Недостаточно доступной рабочей памяти.

Непротиворечивость данных

Данные принимаются согласованно, если вы помните следующие указания:

- После того, как параметр состояния NDR получил значение 1, вы должны снова вызвать SFB9 "URCV" со значением 0 в EN_R. Это гарантирует, что область приема не будет перезаписана снова прежде, чем вы ее оцените.

Полностью оцените использованные в последний раз области приема (RD_i) прежде, чем вы освободите блок для нового приема (вызов со значением 1 на управляющем входе EN_R).

17.5 Передача сегментированных данных с помощью SFB12 "BSEND"

Описание

SFB12 "BSEND" передает данные блоку SFB типа "BRCV" удаленного партнера (параметр R_ID должен быть идентичен тем, что в соответствующих SFB). При этом типе передачи данных между коммуникационными партнерами может транспортироваться большее количество данных, чем возможно в случае всех других коммуникационных SFB для спроектированных соединений S7, а именно до 64 Килобайтов (применительно ко всем CPU). Причина этого заключается в том, что область передаваемых данных, сегментирована. Каждый сегмент передается партнеру отдельно, и отправитель ждет подтверждения приема перед передачей следующего сегмента.

Задание на передачу активизируется после вызова блока, когда имеется нарастающий фронт сигнала на управляющем входе REQ. Начальный адрес передаваемых данных задается параметром SD_1, длина массива данных задается параметром LEN. Длина, заданная параметром SD_1, оценивается только тогда, когда блок вызывается в первый раз после теплого или холодного рестарта. Она определяет максимальную длину области передачи. Передача данных из памяти пользователя асинхронна по отношению к обработке программы пользователя. Успешное завершение передачи отображается значением 1 в параметре состояния DONE. Когда имеется нарастающий фронт сигнала на управляющем входе R, текущая передача данных отменяется.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	Параметр управления «запрос».
R	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления «сброс»: отменяет текущее задание.
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
R_ID	INPUT	DWORD	I, Q, M, D, L, константа	Параметр адресации R_ID. В случае подключения через CP 441 к S5 или другим устройствам R_ID содержит информацию об адресе удаленного устройства. За дополнительной информацией обратитесь к описанию CP 441.
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: передача данных завершена.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
SD_1	IN_OUT	ANY	I, Q, M, D, T, C	Указатель на область передачи. Информация о длине оценивается только во время первого вызова блока после теплого или холодного рестарта. Она определяет максимальную длину области передачи. Разрешены типы данных BOOL (не разрешен битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE AND TIME, COUNTER, TIMER.
LEN	IN_OUT	WORD	I, Q, M, D, L	Длина массива передаваемых данных в байтах.

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB12 информацию об ошибках, которая может выводиться параметрами ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	<ul style="list-style-type: none"> Проблемы связи, например, не загружено описание соединения (локальное или удаленное) Разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	2	Отрицательное подтверждение от SFB партнера. Функция не может быть выполнена.
1	3	R_ID неизвестен в соединении, задаваемом ID, или еще не был вызван блок приема.
1	4	Ошибка в указателе области передачи SD_1, включая длину данных или тип данных, либо с LEN было передано значение 0, либо ошибка в указателе области приема RD_1 соответствующего SFB13 "BRCV".
1	5	Был выполнен запрос на сброс.
1	6	SFB партнера находится в состоянии DISABLED [заблокирован] (EN_R имеет значение 0).
1	7	SFB партнера находится в ошибочном состоянии (блок приема не вызывался снова после последней передачи данных).
1	8	Обращение к удаленному объекту в памяти пользователя было отклонено.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB12 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	18	R_ID уже существует в идентификаторе (ID) соединения.
1	20	Недостаточно доступной рабочей памяти.

Непротиворечивость данных

Чтобы гарантировать непротиворечивость данных, вы можете записывать снова в область передачи SD_i, используемую в текущий момент времени, только после того, как операция передачи завершится. Это имеет место тогда, когда значение параметра состояния DONE станет равным 1.

17.6 Прием сегментированных данных с помощью SFB13 "BRCV"

Описание

SFB13 "BRCV" принимает данные от SFB типа "BSEND" удаленного партнера (параметр R_ID должен быть одинаковым в обоих SFB). После того, как он был вызван и на управляющий вход EN_R подано значение 1, блок готов принимать данные. Начальный адрес области приема определяется RD_1.

После приема каждого сегмента данных в SFB партнера передается подтверждение и обновляется параметр LEN. Если блок вызывается во время асинхронного приема данных, то это приводит к предупреждению, выводимому в параметре состояния STATUS; если вызов выполняется, когда на управляющий вход EN_R подано значение 0, то прием завершается и SFB возвращается в свое исходное состояние. Безошибочный прием всех сегментов данных отображается значением 1 в параметре состояния NDR. Принятые данные остаются неизменными до тех пор, пока SFB13 не будет вызван снова с EN_R=1.

Параметр	Описание	Тип данных	Область памяти	Характеристика
EN_R	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "разрешение на прием".
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
R_ID	INPUT	DWORD	I, Q, M, D, L, константа	Параметр адресации R_ID. В случае подключения через CP 441 к S5 или другим устройствам R_ID содержит информацию об адресе удаленного устройства. За дополнительной информацией, обратитесь к описанию CP 441.
NDR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния NDR: новые данные приняты.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
RD_1	IN_OUT	ANY	I, Q, M, D, T, C	Указатель на область приема. Информация о длине определяет максимальную длину принимаемого массива. Разрешены типы данных BOOL (не разрешен битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER.
LEN	IN_OUT	WORD	I, Q, M, D, L	Длина данных, принятых до сих пор, в байтах.

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB13 информацию об ошибках, которая может выводиться параметрами ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	17	Предупреждение: блок, принимающий данные асинхронно.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	<ul style="list-style-type: none"> Проблемы связи, например, не загружено описание соединения (локальное или удаленное) Разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	2	Функция не может быть выполнена.
1	4	Ошибка в указателе области приема RD_1, включая длину данных или тип данных (переданный блок данных длиннее, чем область приема).
1	5	Принят запрос на сброс, неполная передача.
1	8	Обращение к удаленному объекту в памяти пользователя было отклонено.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB13 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	18	R_ID уже существует в идентификаторе (ID) соединения.
1	20	Недостаточно доступной рабочей памяти.

Непротиворечивость данных

Данные принимаются согласованно, если вы помните следующие указания:

- После того, как параметр состояния NDR стал равным 1, вы должны снова вызвать SFB13 "BRCV" со значением 0 в EN_R. Это гарантирует, что область приема не будет перезаписана снова прежде, чем вы оцените ее.

Полностью оцените использованные в последний раз области приема (RD_1) прежде, чем вы освободите блок для нового приема (вызов со значением 1 на управляющем входе EN_R).

17.7 Чтение данных из удаленного CPU с помощью SFB14 "GET"

Описание

С помощью SFB14 "GET" вы можете читать данные из удаленного CPU. Удаленный CPU может находиться в режиме RUN или STOP.

При нарастающем фронте сигнала на управляющем входе REQ в CPU партнера передаются соответствующие указатели на области (ADDR_i), подлежащие считыванию. Удаленный партнер возвращает данные. Ответ анализируется, чтобы выяснить, были ли проблемы доступа при чтении данных, и выполняется проверка типа данных. Если ошибок не было, то принятые данные копируются в спроектированные области приема (RD_i) при следующем вызове SFB. Завершение задания отображается значением 1 в параметре состояния NDR. Принятые данные остаются неизменными до следующего вызова SFB13 с EN_R=1.

Задание на чтение может быть активизировано снова только после того, как завершится предыдущее задание.

Удостоверьтесь, что области, определенные параметрами ADDR_i и RD_i ($1 \leq i \leq 4$), совпадают по длине и типу данных.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	Параметр управления «запрос».
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
NDR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния NDR: данные из CPU партнера приняты.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
ADDR_1	IN_OUT	ANY	напр., I, Q, M, D	Указатели на области в CPU партнера, которые будут считываться.
ADDR_2	IN_OUT	ANY	напр., I, Q, M, D	
ADDR_3	IN_OUT	ANY	напр., I, Q, M, D	
ADDR_4	IN_OUT	ANY	напр., I, Q, M, D	
RD_i, $1 \leq i \leq 4$	IN_OUT	ANY	I, Q, M, D, T, C	Указатели на области в локальном CPU, в которые будут вводиться прочитанные данные. Разрешены типы данных BOOL (не разрешено: битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER.

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB14 информацию об ошибках, которая может выводиться параметрами ERROR и STATUS

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none"> не загружено описание соединения (локальное или удаленное) разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	2	Отрицательное подтверждение от устройства партнера. Функция не может быть выполнена.
1	4	Ошибки в указателях областей приема RD_i, включая длину данных или тип данных.
1	8	Ошибка доступа в CPU партнера.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB14 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	20	Недостаточно доступной рабочей памяти.

Непротиворечивость данных

В случае SFB14 "GET" переданные данные непротиворечивы, если вы придерживаетесь правил обеспечения непротиворечивости, сформулированных для SFC67 "X_GET" и 72 "I_GET" в разделе 18.4.

17.8 Запись данных в удаленный CPU с помощью SFB15 "PUT"

Описание

С помощью SFB15 "PUT" вы можете записывать данные в удаленный CPU. Удаленный CPU может находиться в режиме RUN или STOP.

При нарастающем фронте сигнала на управляющем входе REQ в CPU партнера передаются указатели на области (ADDR_i), в которые нужно записывать. Удаленный партнер сохраняет требуемые данные по адресам, предоставленным вместе с данными, и возвращает подтверждение выполнения. Подтверждение выполнения оценивается. Если ошибок не было, то это отображается значением 1 в параметре состояния DONE при следующем вызове SFB.

Задание на запись может быть активизировано снова только после того, как завершится последнее задание.

Удостоверьтесь, что области, определенные параметрами ADDR_i и SD_i ($1 \leq i \leq 4$), согласуются по номеру, длине и типу данных.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	Параметр управления «запрос».
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: функция выполнена.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
ADDR_1	IN_OUT	ANY	напр., I, Q, M, D	Указатели на области в CPU партнера, в которые будут записываться данные.
ADDR_2	IN_OUT	ANY	напр., I, Q, M, D	
ADDR_3	IN_OUT	ANY	напр., I, Q, M, D	
ADDR_4	IN_OUT	ANY	напр., I, Q, M, D	
SD_i, $1 \leq i \leq 4$	IN_OUT	ANY	I, Q, M, D, T, C	Указатели на области в локальном CPU, содержащие данные, которые должны передаваться. Разрешены типы данных BOOL (не разрешен битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER.

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB15 информацию об ошибках, которая может выводиться параметрами ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none"> не загружено описание соединения (локальное или удаленное) разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	2	Отрицательное подтверждение от устройства партнера. Функция не может быть выполнена.
1	4	Ошибки в указателях областей передачи SD_i, включая длину данных или тип данных.
1	8	Ошибка доступа в CPU партнера.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB15 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	20	Недостаточно доступной рабочей памяти.

Непротиворечивость данных

В случае SFB15 "PUT" переданные данные непротиворечивы, если вы придерживаетесь правил обеспечения непротиворечивости, сформулированных для SFC 68 "X_PUT" и 73 "I_PUT" в разделе 18.4.

17.9 Передача данных на принтер с помощью SFB16 "PRINT"

Описание

SFB16 "PRINT" передает данные и команду форматирования на удаленный принтер, например, через CP 441.

При нарастающем фронте сигнала на управляющем входе REQ описание формата (FORMAT) и данные (SD_i) передаются на принтер, выбранный посредством ID и PRN_NR.

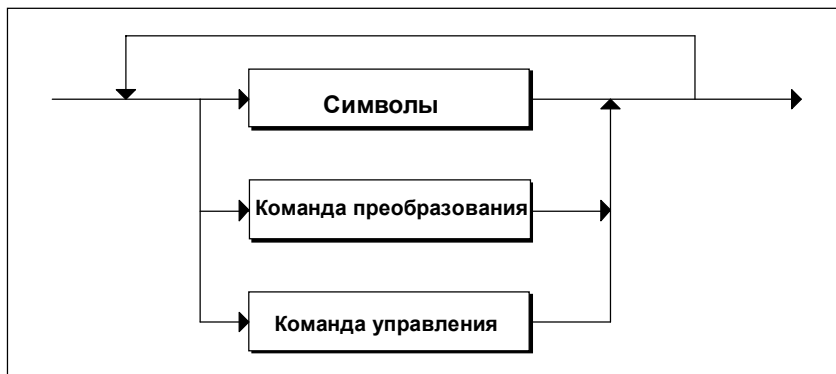
Если вы используете не все четыре области передачи, то вы должны убедиться в том, что первая область описана параметром SD_1, вторая область (если она существует) – параметром SD_2, третья область (если она существует) – параметром SD_3.

Успешное выполнение задания отображается параметром состояния DONE, ошибки отображаются параметрами состояния ERROR и STATUS.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	Параметр управления «запрос».
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: задание на передачу завершено.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
PRN_NR	IN_OUT	BYTE	I, Q, M, D, L	Номер принтера.
FORMAT	IN_OUT	STRING	I, Q, M, D, L	Описание формата.
SD_i, 1 ≤ i ≤ 4	IN_OUT	ANY	I, Q, M, D, T, C	Указатель на i-ю область передаваемых данных. Разрешены типы данных BOOL (не разрешен битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER.

Проходной (in/out) параметр FORMAT

Символьная строка FORMAT содержит печатные символы и элементы формата. Она имеет следующую структуру:

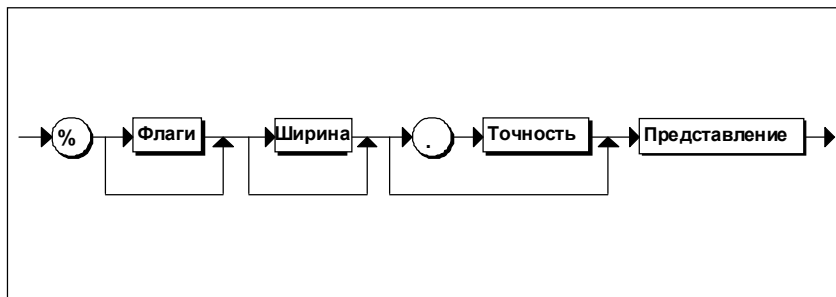


Для каждой области, передаваемой для печати (с SD_1 по SD_4), должна иметься одна команда преобразования в параметре FORMAT. Команды преобразования применяются к областям передачи (SD_i) в том порядке, в котором они формулируются. Символы и команды могут следовать за друг другом в любом порядке.

- Символы

Разрешены следующие символы:

- все печатные символы
- \$\$ (символ доллара), '\$' (отдельный апостроф), \$L и \$I (перевод строки), \$P и \$p (страница), \$R и \$r (возврат каретки), \$T и \$t (табулятор).



Синтаксическая диаграмма команды преобразования

Элемент команды преобразования	Значение
Флаги	<ul style="list-style-type: none"> отсутствуют : выровненный по правому краю вывод. - : выровненный по левому краю вывод .
Ширина	<ul style="list-style-type: none"> отсутствует : вывод в стандартном представлении. n : выводятся точно n символов. Если вывод выровнен по правому краю, то ему могут предшествовать пробелы; если вывод выровнен по левому краю, то пробелы следуют после символов.
Точность	<p>Точность имеет значение только для представлений A, D, F и R (см. следующую таблицу).</p> <ul style="list-style-type: none"> отсутствует : вывод в стандартном представлении. 0 : без вывода десятичной точки или десятичных разрядов в представлениях F и R. n <ul style="list-style-type: none"> в случае F и R: вывод десятичной точки и n десятичных разрядов в случае A и D (дата): количество цифр для года: возможны значения 2 и 4.
Представление	<p>Следующая таблица содержит:</p> <ul style="list-style-type: none"> возможные представления типы данных, возможные для каждого представления стандартный формат для каждого представления (распечатка производится в стандартном представлении, если в параметре FORMAT не заданы ширина и точность) и их максимальная длина.

Следующая таблица показывает возможные виды представления в команде преобразования параметра FORMAT.

Представление	Возможные типы данных	Стандартное представление		Комментарий
		Пример	Длина	
A, a	DATE	25.07.1996	10	-
	DWORD			
C, c	CHAR	K	1	-
	BYTE	M	1	
	WORD	KL	2	
	DWORD	KLMN	4	
	ARRAY of CHAR	KLMNOP	Число символов	
	ARRAY of BYTE			
D, d	DATE	1996-07-25	10	-
	DWORD			
F, f	REAL	0.345678	8	-
	DWORD			
H, h	Все типы данных, включая ARRAY of BYTE (байтовый массив)	В зависимости от типа данных	В зависимости от типа данных	Шестнадцатеричное представление

Представление	Возможные типы данных	Стандартное представление		Комментарий
I, i	INT	- 32 768	макс. 6	-
	WORD	- 2 147 483 648	макс. 11	
N, n	WORD	Вывод текста	-	Соответствующая область передачи SD_i содержит ссылку (в виде номера) на текст, который нужно напечатать. Текст находится в модуле (например, CP 441), который создает печатаемую строку. Если текст под указанным номером не найден, то выводится *****.
R, r	REAL	0.12E-04	8	-
	DWORD			
S, s	STRING	Вывод текста		-
T, t	TIME	2d_3h_10m_5s_250ms	макс. 21	Если появляется ошибка, то выводится *****.
	DWORD			
U, u	BYTE	255	макс. 3	-
	WORD	65 535	макс. 5	
	DWORD	4 294 967 295	макс. 10	
X, x	BOOL	1	1	-
	BYTE	101 ..	8	
	WORD	101 ..	16	
	DWORD	101 ..	32	
Z, z	TIME_OF_DAY	15:38:59.874	12	-

В пунктах этой таблицы, в которых указана максимальная длина для представления, фактическая длина, конечно, может быть короче.

Замечание

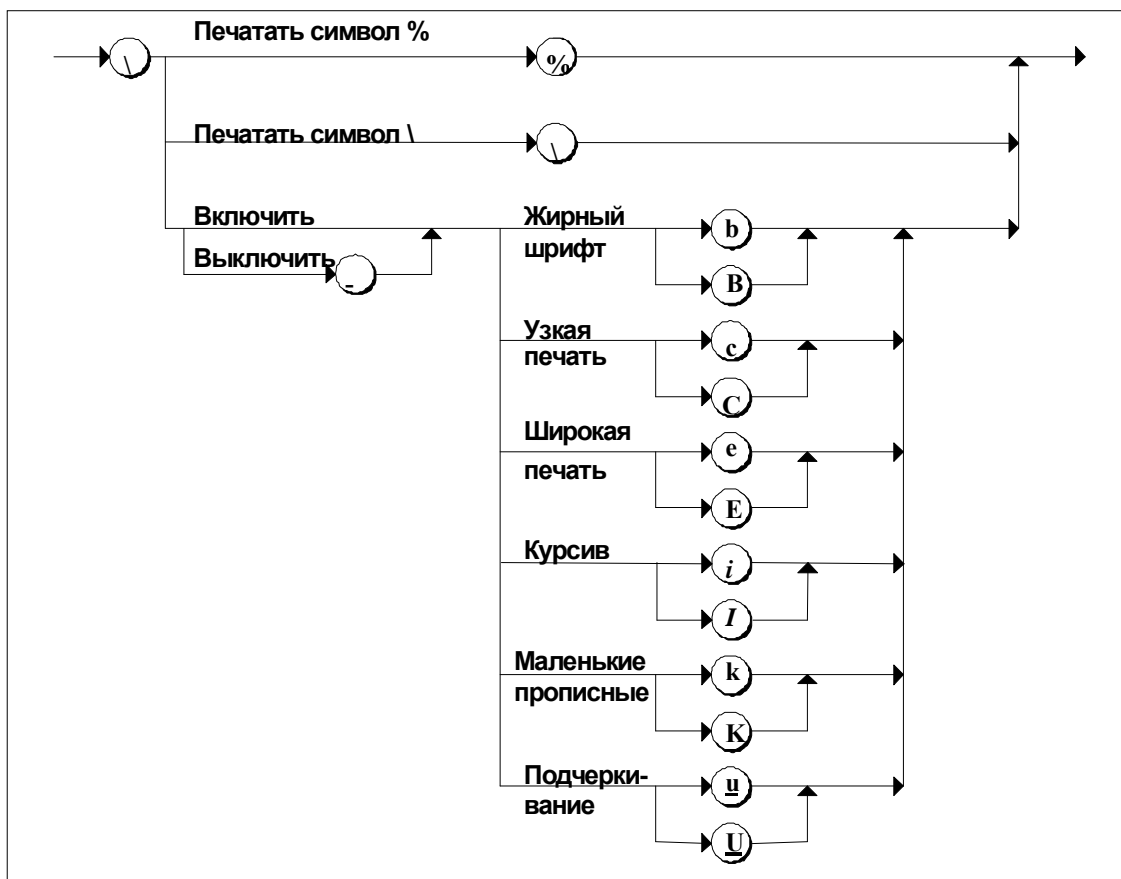
Для типов данных C и S следующие пункты зависят от используемого принтера:

- какие символы могут печататься
- что печатает принтер вместо непечатных символов, если драйвер принтера не имеет таблицы преобразования этих символов.

- Команда управления

С помощью команды управления вы можете делать следующее:

- печатать символы % и \
- изменять параметры настройки принтера.



Синтаксическая диаграмма команды управления

Если вы пытаетесь, например, заблокировать неразрешенный шрифт или выполнить функцию, которую принтер не распознает, то команда управления игнорируется. Следующая таблица содержит ошибки, которые могут происходить в связи с проходным параметром FORMAT.

Ошибка	Вывод принтера
Команда преобразования не может быть выполнена.	Выводятся символы * в соответствии с (максимальной) длиной заданного по умолчанию представления или заданной шириной.
Заданная ширина слишком мала.	В представлениях A, C, D, N, S, T и Z печатается столько символов, сколько задано посредством выбранной ширины. При всех других представлениях на протяжении заданной ширины печатаются символы *.
Слишком много команд преобразования.	Команды преобразования, для которых нет указателя области передачи SD_i, игнорируются.
Слишком мало команд преобразования.	Области передачи, для которых нет команды преобразования, не распечатываются.
Неопределенные или неподдерживаемые команды преобразования.	Печатается *****.
Неполная команда преобразования.	Печатается *****.
Неопределенные или неподдерживаемые команды управления.	Команды управления, не соблюдающие синтаксис, показанный на вышеприведенном рисунке, игнорируются.

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB16 "PRINT" информацию об ошибках, которая может быть распечатана с помощью параметров ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не активно, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none"> не загружено описание соединения (локальное или удаленное) разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	2	Отрицательное подтверждение от принтера. Функция не может быть выполнена.
1	3	PRN_NR неизвестен в канале связи, заданном посредством ID.
1	4	Ошибка в проходном (in/out) параметре FORMAT или в указателях области передачи SD_i относительно длины данных или типа данных.
1	6	Удаленный принтер находится в режиме OFFLINE.
1	7	Удаленный принтер не готов к работе (например, нет бумаги).
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	13	Ошибка в проходном (in/out) параметре FORMAT.
1	20	Недостаточно доступной рабочей памяти.

Количество передаваемых данных

Количество данных, которые можно передать удаленному принтеру, не должно превышать максимальную длину.

Эта максимальная длина данных рассчитывается следующим образом:

максимальная длина = 420 – формат

Формат – это текущая длина параметра FORMAT в байтах. Данные, подлежащие распечатке, могут быть распределены по одной или большему количеству областей передачи.

17.10 Инициализация теплого или холодного рестарта в удаленном устройстве с помощью SFB19 "START"

Описание

SFB19 "START" при нарастающем фронте сигнала на управляющем входе REQ активизирует теплый или холодный рестарт в удаленном устройстве, адресованном посредством ID. Если удаленная система отказоустойчива, то запрос рестарта имеет силу для любого CPU этой системы. Если удаленным устройством является CPU, то должны быть выполнены следующие условия:

- CPU находится в состоянии STOP
- переключатель CPU установлен в положение "RUN" или "RUN-P".

Как только теплый или холодный рестарт завершается, устройство переключается в режим RUN и передает положительное подтверждение выполнения. Когда положительное подтверждение оценено, параметр состояния DONE устанавливается в 1. Если происходят какие-либо ошибки, то они отображаются параметрами состояния ERROR и STATUS.

Дальнейший теплый или холодный рестарт в том же самом удаленном устройстве может быть активизирован только после того, как завершится последний полный рестарт.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	Параметр управления «запрос».
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: функция выполнена.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
PI_NAME	IN_OUT	ANY	I, Q, M, D, T, C	Указатель на область памяти, в которой расположено имя запускаемой программы (код ASCII). Это имя не должно содержать более 32 символов. В ПЛК S7 это должно быть P_PROGRAM.
ARG	IN_OUT	ANY	I, Q, M, D, T, C	Параметр выполнения: <ul style="list-style-type: none"> • Если вы не задаете значение для ARG, то в удаленном устройстве выполняется теплый рестарт. • Если вы задаете значение "C", то в удаленном устройстве выполняется холодный рестарт (если удаленное устройство способно к запуску этого типа).
IO_STATE	IN_OUT	BYTE	I, Q, M, D, L	В настоящее время не имеет значения. Не присваивайте значение этому параметру, если вашим коммуникационным партнером является программируемый контроллер S7.

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB19 информацию об ошибках, которая может выводиться параметрами ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none"> не загружено описание соединения (локальное или удаленное) разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	2	Отрицательное подтверждение от устройства партнера. Функция не может быть выполнена.
1	3	Имя программы, введенное для PI_NAME, неизвестно.
1	4	Ошибка в указателях PI_NAME или ARG, включая длину данных или тип данных.
1	7	Полный рестарт в устройстве партнера невозможен.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB19 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	20	Недостаточно доступной рабочей памяти.

17.11 Переключение удаленного устройства в состояние STOP с помощью SFB20 "STOP"

Описание

SFB20 "STOP" при нарастающем фронте сигнала на управляющем входе REQ активизирует переключение в режим STOP в удаленном устройстве, адресованном посредством ID. Переключение режима возможно, когда устройство находится в режиме RUN, HOLD или STARTUP.

Если удаленная система отказоустойчива, то запрос рестарта имеет силу для любого CPU этой системы.

Успешное выполнение задания отображается значением 1 в параметре состояния DONE. Если происходят какие-либо ошибки, то они отображаются параметрами состояния ERROR и STATUS.

Изменение режима в том же самом удаленном устройстве может быть запущено только тогда, когда предыдущий вызов SFB20 полностью завершен.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	Параметр управления «запрос».
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: функция выполнена.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
PI_NAME	IN_OUT	ANY	I, Q, M, D	Указатель на область памяти, в которой расположено имя запускаемой программы (код ASCII). Это имя не должно содержать более 32 символов. В ПЛК S7 это должно быть P_PROGRAM.
IO_STATE	IN_OUT	BYTE	I, Q, M, D, L	В настоящее время не имеет значения. Не присваивайте значение этому параметру, если вашим коммуникационным партнером является программируемый контроллер S7.

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB20 информацию об ошибках, которая может выводиться параметрами ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none"> не загружено описание соединения (локальное или удаленное) разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	2	Отрицательное подтверждение от устройства партнера. Функция не может быть выполнена.
1	3	Имя программы, введенное для PI_NAME, неизвестно.
1	4	Ошибка в указателе PI_NAME, включая длину данных или тип данных.
1	7	Устройство партнера уже находится в состоянии STOP.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB20 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	20	Недостаточно доступной рабочей памяти.

17.12 Инициализация горячего рестарта в удаленном устройстве с помощью SFB21 "RESUME"

Описание

SFB21 "RESUME" при нарастающем фронте сигнала на управляющем входе REQ активизирует горячий рестарт в удаленном устройстве, выбранном с помощью ID.

Если удаленным устройством является CPU, то должны быть выполнены следующие условия:

- CPU находится в состоянии STOP
- переключатель CPU установлен в положение "RUN" или "RUN-P"
- при создании конфигурации с помощью STEP 7 вы разрешили ручной горячий рестарт
- нет должно быть условий, препятствующих горячему рестарту.

Как только горячий рестарт завершается, устройство переходит в режим RUN и передает положительное подтверждение выполнения. Когда положительное подтверждение оценено, параметр DONE состояния устанавливается в 1. Если происходят какие-либо ошибки, то они отображаются параметрами состояния ERROR и STATUS.

Рестарт в том же самом удаленном устройстве может быть активизирован снова только после того, как завершится предыдущий горячий рестарт.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	Параметр управления «запрос».
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: функция выполнена.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
PI_NAME	IN_OUT	ANY	I, Q, M, D	Указатель на область памяти, в которой расположено имя запускаемой программы (код ASCII). Это имя не должно содержать более 32 символов. В ПЛК S7 это должно быть P_PROGRAM.
ARG	IN_OUT	ANY	I, Q, M, D, T, C	Параметр выполнения. В настоящее время не имеет значения. Не присваивайте значение этому параметру, если вашим коммуникационным партнером является программируемый контроллер S7.
IO_STATE	IN_OUT	BYTE	I, Q, M, D, L	В настоящее время не имеет значения. Не присваивайте значение этому параметру, если вашим коммуникационным партнером является программируемый контроллер S7.

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB21 информацию об ошибках, которая может выводиться параметрами ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none">• не загружено описание соединения (локальное или удаленное)• разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	2	Отрицательное подтверждение от устройства партнера. Функция не может быть выполнена.
1	3	Имя программы, введенное для PI_NAME, неизвестно
1	4	Ошибка в указателях PI_NAME или ARG, включая длину данных или тип данных.
1	7	Горячий рестарт невозможен.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none">• был задан экземплярный DB, который не принадлежит SFB21• был задан не экземплярный DB, а общедоступный DB• экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	20	Недостаточно доступной рабочей памяти.

17.13 Запрос состояния удаленного партнера с помощью SFB22 "STATUS"

Описание

Используя SFB22 "STATUS" вы можете запросить состояние удаленного коммуникационного партнера.

Если имеется нарастающий фронт сигнала на управляющем входе REQ, задание передается удаленному партнеру. Ответ оценивается, чтобы определить, были ли проблемы. Если ошибок не было, то принятое состояние копируется в переменные PHYS, LOG и LOCAL при следующем вызове SFB. Завершение этого задания отображается значением 1 в параметре состояния NDR.

Вы можете запрашивать состояние того же самого коммуникационного партнера снова только после того, как завершится последний запрос.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	Параметр управления «запрос».
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
NDR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния NDR: новое состояние устройства принято.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
PHYS	IN_OUT	ANY	I, Q, M, D	Физическое состояние (минимальная длина: один байт), возможные значения: <ul style="list-style-type: none"> 10H функционирование 13H требуется техническое обслуживание.
LOG	IN_OUT	ANY	I, Q, M, D	Логическое состояние (минимальная длина: один байт), возможное значение: <ul style="list-style-type: none"> 00H разрешено изменение состояния.
LOCAL	IN_OUT	ANY	I, Q, M, D	Состояние, если устройством партнера является CPU S7 (минимальная длина: два байта).

Проходной (in/out) параметр LOCAL

Если коммуникационным партнером является CPU S7, то проходной (in/out) параметр LOCAL содержит его текущее состояние. Первый байт зарезервирован, второй байт содержит ID состояния.

Режим работы	Соответствующий идентификатор
STOP	00H
Теплый рестарт	01H
RUN	02H
Горячий рестарт	03H
HOLD	04H
Холодный рестарт	06H
RUN_R	09H
LINK-UP	0BH
UPDATE	0CH

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB22 информацию об ошибках, которая может выводиться параметрами ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none"> не загружено описание соединения (локальное или удаленное) разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	2	Отрицательное подтверждение от устройства партнера. Функция не может быть выполнена.
1	4	Ошибка в PHYS, LOG или LOCAL, включая длину данных или тип данных.
1	8	Обращение к удаленному объекту было отклонено.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB22 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	20	Недостаточно доступной рабочей памяти.

17.14 Прием состояния удаленного устройства с помощью SFB23 "USTATUS"

Описание

SFB23 "USTATUS" принимает состояние устройства удаленного коммуникационного партнера. Партнер передает свое состояние без запроса, когда происходит его изменение, если это запроецировано в STEP 7.

Если при вызове SFB на управляющий вход EN_R подается значение 1 и имеется кадр от партнера, то информация о состоянии вводится в переменные PHYS, LOG и LOCAL при следующем вызове SFB. Завершение этого задания отображается значением 1 в параметре состояния NDR.

В соединении, используемом USTATUS, должна быть разблокирована передача сообщений о рабочем состоянии.

Замечание

Вы можете использовать только один экземпляр SFB23 на соединение.

Параметр	Описание	Тип данных	Область памяти	Характеристика
EN_R	INPUT	BOOL	I, Q, M, D, L	Параметр управления "разрешение на прием".
ID	INPUT	WORD	I, Q, M, D, константа	Ссылка на соединение.
NDR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния NDR: новое состояние устройства принято.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
PHYS	IN_OUT	ANY	I, Q, M, D	Физическое состояние (минимальная длина: один байт), возможные значения: <ul style="list-style-type: none"> • 10H функционирование • 13H требуется техническое обслуживание.
LOG	IN_OUT	ANY	I, Q, M, D	Логическое состояние (минимальная длина: один байт), возможное значение: <ul style="list-style-type: none"> • 00H разрешено изменение состояния.
LOCAL	IN_OUT	ANY	I, Q, M, D	Состояние, если устройством партнера является CPU S7 (минимальная длина: один байт).

Проходной (in/out) параметр LOCAL

Если коммуникационным партнером является CPU S7, то проходной (in/out) параметр LOCAL содержит его текущее состояние. Первый байт зарезервирован, второй байт содержит ID состояния.

Режим работы	Соответствующий идентификатор
STOP	00H
Теплый рестарт	01H
RUN	02H
- Горячий рестарт	03H
HOLD	04H
Холодный рестарт	06H
RUN_R	09H
LINK-UP	0BH
UPDATE	0CH

Информация об ошибках

Следующая таблица содержит всю относящуюся к SFB23 информацию об ошибках, которая может выводиться параметрами ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	9	Предупреждение о потере информации: Более старое состояние устройства было заменено более новым состоянием устройства.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи, например: <ul style="list-style-type: none"> не загружено описание соединения (локальное или удаленное) разрыв соединения (например, кабель, CPU выключен, CP в режиме STOP).
1	4	Ошибка в PHYS, LOG или LOCAL, включая длину данных или тип данных.
1	10	Обращение к локальной памяти пользователя невозможно (например, обращение к DB, который был удален).
1	12	При вызове SFB <ul style="list-style-type: none"> был задан экземплярный DB, который не принадлежит SFB23 был задан не экземплярный DB, а общедоступный DB экземплярный DB не был найден (загрузка нового экземплярного DB из PG).
1	18	Для соединения, идентифицируемого ID, уже имеется экземпляр SFB23 "USTATUS".
1	19	Удаленный CPU передает данные быстрее, чем они могут приниматься SFB в программе пользователя.
1	20	Недостаточно доступной рабочей памяти

17.15 Запрос состояния соединения, относящегося к экземпляру коммуникационного SFB, с помощью SFC62 "CONTROL"

Описание

С помощью SFC62 "CONTROL" вы можете запрашивать состояние соединения, относящегося к экземпляру локального коммуникационного SFB.

После вызова этой системной функции со значением 1 на управляющем входе EN_R запрашивается текущее состояние соединения, относящегося к экземпляру коммуникационного SFB, выбранному с помощью I_DB.

Параметр	Описание	Тип данных	Область памяти	Характеристика
EN_R	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "разрешение на прием".
I_DB	INPUT	BLOCK_DB	I, Q, M, D, L, константа	Номер экземплярного DB.
OFFSET	INPUT	WORD	I, Q, M, D, L, константа	Номер записи данных в мультитемпларном DB (если мультитемпларный DB не существует, то здесь нужно ввести 0).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках.
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR.
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS.
I_TYP	OUTPUT	BYTE	I, Q, M, D, L	Идентификатор типа блока, принадлежащего выбранному экземпляру.
I_STATE	OUTPUT	BYTE	I, Q, M, D, L	Не имеет значения для пользователя. Зарезервирован для внутренней диагностики.
I_CONN	OUTPUT	BOOL	I, Q, M, D, L	Состояние соответствующего соединения; возможные значения: <ul style="list-style-type: none"> 0: соединение прервано или не установлено 1: соединение существует.
I_STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS запрашиваемого экземпляра коммуникационного SFB.

Выходной параметр I_TYP

Следующая таблица перечисляет различные типы SFB и соответствующие идентификаторы.

Тип SFB	Идентификатор (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
(SFB не существует; неправильный I_DB или OFFSET)	FF

Информация об ошибках

В SFC62 "CONTROL" выходной параметр RET_VAL может иметь два следующих значения:

- 0000H: во время выполнения SFC ошибок не было
- 8000H: во время выполнения SFC возникла ошибка.

Замечание

Выходные параметры ERROR и STATUS следует оценивать, даже если в выходном параметре RET_VAL отображается значение 0000H.

ERROR	STATUS (десятичное число)	Объяснение
1	10	Обращение к локальной памяти пользователя невозможно (например, для I_TYP в качестве фактического параметра был указан байт памяти, и этот байт памяти не существует в используемом CPU).
1	12	Для номера, заданного посредством I_DB <ul style="list-style-type: none"> • имеется не экземплярный DB, а общедоступный DB • нет DB, либо экземпляр разрушен.

17.16 Подпрограмма запуска коммуникационных SFB для спроектированных соединений S7

Требования

В следующем описании предполагается, что

- в модулях существуют описания соединений (SDB)
- установлены спроектированные соединения
- фактический параметр для ID совпадает с ID спроектированного соединения для каждого SFB.

Теплый рестарт и холодный рестарт

Во время теплого и холодного рестарта все коммуникационные SFB устанавливаются в состояние NO_INIT. Фактические параметры, хранимые в экземплярных DB, не изменяются.

Теплый рестарт и холодный рестарт с помощью SFB для двустороннего обмена данными

В общем случае два модуля с коммуникационными SFB для двустороннего обмена данными не проходят теплый или холодный рестарт одновременно. Реакцией SFB управляют правила, приведенные ниже.

Блоки приема (SFB URCV, BRCV) реагируют следующим образом:

- Если SFB принял задание, но не подтвердил это задание во время теплого или холодного рестарта, то он генерирует кадр прерывания последовательности (CFB, BRCV) и затем сразу переходит в состояние NO_INIT.
- В случае SFB BRCV, возможно, будет принят другой сегмент данных, несмотря на передачу кадра прерывания последовательности. Он будет отброшен на месте.
- SFB URCV переключается в состояние NO_INIT сразу.

Блоки передачи (SFB USEND, BSEND) реагируют следующим образом:

- Если SFB BSEND запустил последовательность задания, которая еще не была завершена, то он передает прерывание последовательности, когда инициируется теплый или холодный рестарт. Тогда он сразу после этого переходит в состояние NO_INIT. Поступающее позже подтверждение отбрасывается на месте.
- Если SFB BSEND уже передал или принял прерывание последовательности к моменту, когда запрашивается теплый или холодный рестарт, то он сразу переходит в состояние NO_INIT.
- Во всех других случаях и всякий раз, когда SFB передает только сообщения (например, SFB USEND), локальная обработка прерывается, и SFB сразу переходит в состояние NO_INIT.

Теплый рестарт и холодный рестарт с помощью коммуникационных SFB для одностороннего обмена данными

Предполагается, что после того, как соединения установлены, сервер коммуникационного партнера работает, иными словами, он может в любое время обрабатывать задания или выводить сообщения.

Коммуникационные SFB, которые передают задания и ожидают подтверждений, реагируют на полный рестарт следующим образом:

Текущая обработка прерывается, и SFB сразу после этого переходит в состояние NO_INIT. Если подтверждение для задания, переданного до теплого или холодного рестарта, поступает позже, то оно отбрасывается на месте.

Новое задание, возможно, было передано прежде, чем принято подтверждение на более раннее задание.

Коммуникационные SFB, которые выводят или принимают сообщения, реагируют следующим образом:

- Текущая обработка прерывается, и SFB сразу после этого переходит в состояние NO_INIT.
- В случае SFB USTATUS сообщения, поступающие во время состояний NO_INIT и DISABLED, отбрасываются на месте.

Реакция на горячий рестарт

Коммуникационные SFB для спроектированных соединений S7 устанавливаются в состояние NO_INIT только во время теплого или холодного рестарта. Это означает, что они реагируют подобно функциональным блокам пользователя, выполнение которых может возобновляться после горячего рестарта.

Реакция на сброс памяти

Сброс памяти всегда вызывает прерывание всех соединений. Так как теплый или холодный рестарт – это единственный возможный тип запуска программы пользователя после сброса памяти, все коммуникационные SFB для спроектированных соединений S7 (если они все еще существуют) устанавливаются в состояние NO_INIT и инициализируются. Блоки партнера в модуле, память которого не сбрасывалась, в качестве реакции на прерывание соединения переключаются в состояние IDLE, ENABLED или DISABLED.

17.17 Как SFB для спроектированных соединений S7 реагируют на проблемы

Прерванное соединение

Соединения, выделенные экземплярам коммуникационных SFB, проверяются.

Если соединение прерывается, то реакция коммуникационного SFB зависит от его внутреннего состояния.

Если разрыв соединения обнаруживается, когда блок находится в состоянии IDLE или ENABLED, то SFB реагирует следующим образом:

- Он переходит в состояние ERROR и выводит ID ошибки "Communication problems [Ошибки связи]" в выходных параметрах ERROR и STATUS.
- Когда блок вызывается в следующий раз, он возвращается в свое первоначальное состояние и снова проверяет соединение.

Коммуникационный SFB, который не находится в состоянии IDLE или ENABLED, реагирует следующим образом:

- Он прерывает обработку, переходит в состояние ERROR сразу или при следующем вызове блока и выводит ID ошибки "Communication problems [Ошибки связи]" в выходных параметрах ERROR и STATUS.
- Когда блок вызывается в следующий раз, он переходит в состояние IDLE, DISABLED или ENABLED. В состояниях IDLE и ENABLED соединение проверяется снова.

Эта процедура будет выполнена также, если соединение тем временем снова было установлено.

Выключение электропитания

Выключение электропитания при резервном батарейном питании, сопровождаемое последующим рестартом вызывает прерывание всех установленных соединений. Поэтому ко всем вовлеченным блокам применяются вышеуказанные пункты.

Если имеют место выключение электропитания при резервном батарейном питании с последующим автоматическим теплым или холодным рестартом, то применяются указания, относящиеся к прерванным соединениям и тепловому или холодному рестарту.

В частном случае автоматического теплого или холодного рестарта без резервного батарейного питания, где после восстановления питания автоматически выполняется сброс памяти, коммуникационные SFB для спроектированных соединений S7 реагируют так, как описано в разделе *"Подпрограмма запуска коммуникационных SFB для спроектированных соединений S7"*.

Реакция на изменения режима работы

Если режим работы переключается между состояниями STOP, START, RUN и HOLD, то коммуникационный SFB остается в своем текущем состоянии (исключение: во время теплого или холодного рестарта он переходит в состояние NO_INIT). Это относится как к SFB для одностороннего обмена данными, так и к SFB для двустороннего обмена данными.

Интерфейс ошибок с программой пользователя

Если во время обработки коммуникационного SFB происходит ошибка, то он всегда переходит в состояние ERROR. В то же самое время выходной параметр ERROR устанавливается в 1, и соответствующий ID ошибки вводится в выходной параметр STATUS. Вы можете оценивать эту информацию об ошибках в своей программе.

Примеры возможных ошибок:

- Ошибка при сборе передаваемых данных.
- Ошибка при копировании принимаемых данных в области приема (например, попытка обращения к DB, который не существует).
- Длина области переданных данных не соответствует длине области приема, заданной в SFB партнера.

18 Коммуникационные SFC для неспроектированных соединений S7

18.1 Отличия от коммуникационных SFB для спроектированных соединений S7

Критерии выбора

Кроме обмена глобальными данными, имеются два других метода обмена данными между CPU/FM программируемых контроллеров SIMATIC S7:

- Обмен данными с использованием коммуникационных SFC для не спроектированных соединений S7.
- Обмен данными с использованием коммуникационных SFB для спроектированных соединений S7.

Выбор метода зависит от используемого вами программируемого контроллера SIMATIC S7 (S7-300, S7-400) и других параметров обмена данными. Следующая таблица содержит список критериев, на которых вы можете основывать свой выбор.

Критерий	Коммуникационные SFC для не спроектированных соединений S7	Коммуникационные SFB для спроектированных соединений S7
Возможность использования блоков	S7-300 и S7-400.	Только S7-400.
Соединения	Соединение не проектируется, но устанавливается во время выполнения SFC. В зависимости от того, как назначены параметры, соединение после завершения задания или остается установленным, или прекращается. Если соединение временно не может быть установлено, то соответствующее задание не может быть передано.	Соединения проектируются во время конфигурирования системы.
Переключение в состояние STOP	Если CPU, который инициировал задание (и поэтому установил соединение), переходит в STOP во время передачи данных, то все установленные им соединения прерываются.	Соединение поддерживается в состоянии STOP.
Более одного соединения с партнером	В любой момент времени возможно не более одного соединения с коммуникационным партнером.	Вы можете устанавливать несколько соединений с одним и тем же партнером.
Диапазон адресов	Можно адресовать модули в локальной станции или в подсети MPI.	Можно адресовать модули в подсети MPI, в PROFIBUS или в Industrial Ethernet.

Критерий	Коммуникационные SFC для не спроектированных соединений S7	Коммуникационные SFB для спроектированных соединений S7
Количество коммуникационных партнеров	Количество коммуникационных партнеров, с которыми можно связываться поочередно, не ограничивается доступными ресурсами соединения (см. <u>/70/</u> , <u>/101/</u>). (Соединения можно устанавливать и прерывать во время выполнения программы.)	Количество одновременно доступных коммуникационных партнеров ограничено количеством доступных ресурсов соединения. Оно зависит также от используемого CPU (см. <u>/70/</u> , <u>/101/</u>).
Максимальная длина данных пользователя	Гарантируется длина данных пользователя 76 байтов.	Максимальная длина передаваемых данных пользователя зависит от типа блока (USER / URCV, GET и т.д.) и от коммуникационного партнера (S7-300 или S7-400).
Число переменных, передаваемых за вызов	Вы можете передавать только одну переменную.	Вы можете передавать не более четырех переменных.
Классификация блоков	Коммуникационные SFC для не спроектированных соединений S7 являются системными функциями и не требуют памяти пользователя.	Коммуникационные SFB для спроектированных соединений S7 являются системными функциональными блоками и требуют экземплярного DB для текущих параметров и статических данных.
Динамическая модификация адресных параметров	Возможна динамическая модификация адресных параметров: при завершении активного задания вы можете обратиться к другим коммуникационным партнерам.	Динамическая модификация адресных параметров невозможна: соединение определяется и фиксируется первым вызовом блока и остается неизменным до следующего теплого или холодного рестарта.

18.2 Обзор

Классификация SFC

В случае коммуникационных SFC для не спроектированных соединений S7 вы можете производить обмен данными между CPU S7 и другими модулями с коммуникационными функциональными возможностями, если коммуникационные партнеры подключены к одной и той же подсети MPI или если они принадлежат одной и той же станции S7. Связь со станциями в других подсетях с помощью коммуникационных SFC для не спроектированных соединений S7 невозможна.

Коммуникационные SFC для не спроектированных соединений S7 могут выполняться на всех CPU S7-300 и S7-400. С помощью этих CPU вы можете также записывать переменные в CPU S7-200 и считывать переменные из них.

Если коммуникационный партнер не принадлежит той же самой станции S7, но присоединен к той же самой подсети MPI, то обмен данными производится с использованием одного из блоков, перечисленных в следующей таблице.

Блок	Краткая характеристика
SFC65 "X_SEND"/SFC66 "X_RCV"	Обмен данными между коммуникационными партнерами с использованием передающей и принимающей SFC.
SFC67 "X_GET"	Чтение переменной из коммуникационного партнера.
SFC68 "X_PUT"	Запись переменной в коммуникационном партнере.
SFC69 "X_ABORT"	Прерывание существующего соединения с коммуникационным партнером. Ресурсы на обоих концах освобождаются.

Если оба коммуникационных партнера принадлежат одной и той же станции S7, то обмен данными производится с использованием одного из блоков, перечисленных в следующей таблице.

Блок	Краткая характеристика
SFC72 "I_GET"	Чтение переменной из коммуникационного партнера (например, из FM).
SFC73 "I_PUT"	Запись переменной в коммуникационном партнере (например, в FM).
SFC74 "I_ABORT"	Прерывание существующего соединения с коммуникационным партнером. Ресурсы на обоих концах освобождаются.

Максимальная длина данных пользователя

Коммуникационные SFC для не спроектированных соединений S7 встраиваются во все CPU S7-300 и S7-400.

Гарантируется, что с помощью любого SFC могут передаваться 76 байтов данных пользователя (параметр SD или RD).

Соединение с коммуникационным партнером

В случае коммуникационных SFC для не спроектированных соединений S7 соединение устанавливается во время выполнения SFC. В зависимости от значения, присваиваемого вами входному параметру CONT, соединение при завершении обмена данными или остается установленным, или прерывается. Это означает, что связь имеет следующие характеристики:

- Число коммуникационных партнеров, с которыми можно связаться поочередно, выше, чем число коммуникационных партнеров, с которыми можно связаться одновременно (это число зависит от конкретного CPU, см. [/70/](#), [/101/](#)).
- Если соединение с коммуникационным партнером в текущий момент времени не может быть установлено из-за того, что используются все ресурсы соединения (в локальном CPU или в коммуникационном партнере), то это отображается в RET_VAL. Тогда вы должны снова запустить задание позднее в подходящий момент времени. Однако нет гарантии того, что позже установление соединения будет успешным. В случае необходимости проверьте использование ресурсов соединения в вашей программе и используйте CPU с большим количеством ресурсов.

Существующие соединения коммуникационных SFB для спроектированных соединений S7 не могут использоваться коммуникационными SFC для не спроектированных соединений S7.

Как только вы запустили задание, соединение, установленное для задания, может использоваться только для этого конкретного задания. Тогда другие задания, связанные с тем же самым коммуникационным партнером, могут выполняться только после того, как завершено текущее задание.

Замечание

Если ваша программа включает несколько заданий, связанных с одним и тем же коммуникационным партнером, то убедитесь, что вы вызываете SFC, для которых в RET_VAL позже в подходящий момент времени снова вводится W#16#80C0.

Идентификация задания

Если вы запустили передачу данных или прерывание соединения с помощью одного из коммуникационных SFC для не спроектированных соединений S7 и снова вызываете этот SFC прежде, чем завершена текущая передача, то реакция SFC зависит от того, связан ли новый вызов с тем же самым заданием. Следующая таблица объясняет, какие входные параметры определяют задание для каждого SFC. Если параметры соответствуют параметрам задания, которое еще не завершено, то вызов SFC считается продолжающимся вызовом.

SFC	Задание, идентифицируется с помощью
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

Реакция на прерывания

Коммуникационные SFC для не спроектированных соединений S7 могут прерываться ОВ более высокого приоритета. Если прерывающим ОВ снова вызывается тот же самый SFC с идентичным заданием, то этот второй вызов отменяется, и делается соответствующая запись в RET_VAL. Затем продолжается выполнение прерванного SFC.

Обращение к рабочей памяти CPU

Независимо от количества передаваемых данных пользователя коммуникационные функции операционной системы обращаются к рабочей памяти CPU, используя массивы максимальной длины, чтобы время реакции на прерывание не продлевалось за счет использования коммуникационных функций.

В зависимости от того, как вы устанавливаете максимальную загрузку цикла, вытекающую из коммуникаций с помощью STEP 7, к рабочей памяти можно обращаться несколько раз за время выполнения задания с помощью коммуникационных функций операционной системы.

Клиент переключается в STOP

Если CPU, который инициировал задание (и поэтому установил соединение), во время передачи данных переходит в состояние STOP, то все установленные им соединения прерываются.

Выполнение изменений программы

Все части вашей программы, которые непосредственно влияют на вызовы коммуникационных SFC для не спроектированных соединений S7, должны изменяться только в состоянии STOP. В частности, это относится к удалению FC, FB или OB, содержащих вызовы коммуникационных SFC для не спроектированных соединений S7.

После изменения программы вы должны выполнить теплый или холодный рестарт. Несоблюдение этих правил может привести к тому, что ресурсы остаются назначенными, и программируемый контроллер впоследствии будет находиться в неопределенном состоянии.

Типовая программа

С пакетом STEP 7 поставляются две типовые программы для коммуникационных SFB для не спроектированных соединений. Эти программы находятся в папках `step7\examples\com_sfc1` и `step7\examples\com_sfc2`.

18.3 Общие параметры коммуникационных SFC

Входной параметр REQ

Входной параметр REQ (request to activate [запрос, подлежащий активизации]) – это параметр управления, запускаемый уровнем сигнала. Он используется для запуска задания (передачи данных или прерывания соединения).

- При вызове SFC для задания, которое в текущий момент времени неактивно, вы запускаете задание с REQ=1. Если при первом вызове SFC соединение с коммуникационным партнером не существует, то прежде чем начнется передача данных, устанавливается соединение.
- Если вы запускаете задание и оно еще не завершено, когда вы вызываете SFC снова для того же самого задания, то SFC не анализирует REQ.

Выходные параметры RET_VAL и BUSY

Коммуникационные SFC выполняются асинхронно; это означает, что выполнение задания распространяется более, чем на один вызов SFC. Выходные параметры RET_VAL и BUSY показывают состояние задания, см. раздел 2.2.

Входной параметр CONT

Входной параметр CONT (continue [продолжить]) – это параметр управления. Используя этот параметр, вы решаете, остается установленным или нет соединение с коммуникационным партнером после того, как задание завершено.

- Если при первом вызове вы выбираете CONT=0, то после завершения передачи данных соединение прерывается. Затем соединение снова доступно для обмена данными с новым коммуникационным партнером. Этот метод гарантирует, что ресурсы соединения являются занятыми только тогда, когда они используются фактически.
- Если при первом вызове вы выбираете CONT=1, то после завершения передачи данных соединение остается установленным. Например, этот метод полезен тогда, когда вы выполняете циклический обмен данными между двумя станциями.

Замечание

Соединение, установленное при CONT=1, может быть прервано явно с помощью SFC69 "X_ABORT" или SFC74 "I_ABORT".

18.4 Непротиворечивость данных в случае SFC типа GET и PUT

Обращение к рабочей памяти CPU

Коммуникационные функции операционной системы обращаются к рабочей памяти CPU, используя массивы фиксированного размера. Размер массива зависит от заданного CPU и указан в технических данных вашего CPU.

Это гарантирует, что время реакции на прерывание не продлевается за счет использования коммуникационных функций. Так как этот доступ является асинхронным по отношению к программе пользователя, вы не можете передавать любое число байтов данных и быть уверенным в поддержании непротиворечивости данных.

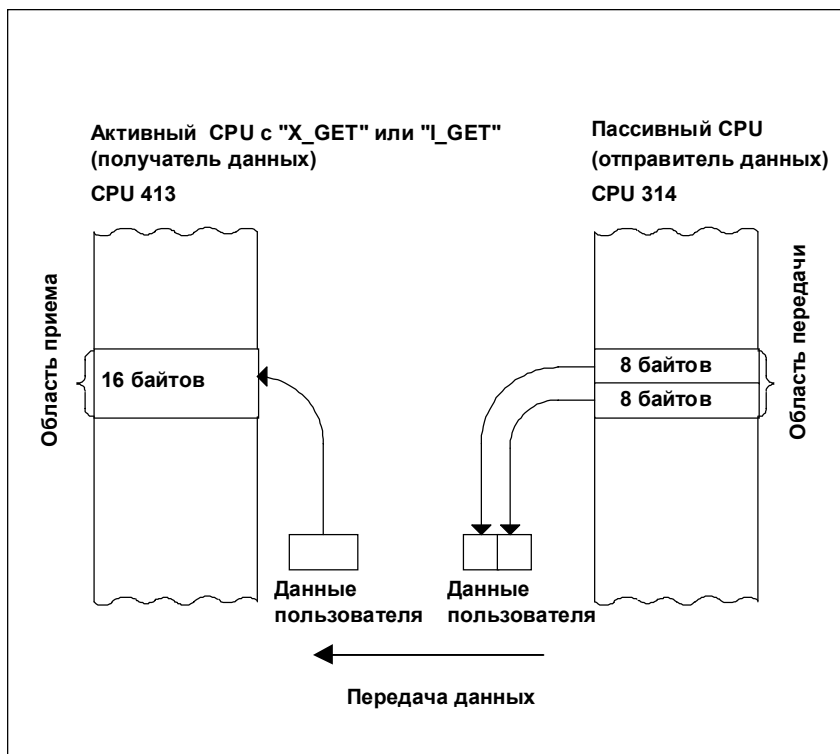
Правила обеспечения непротиворечивости объясняются ниже.

Правила обеспечения непротиворечивости для SFC типа GET

С помощью SFC 67 "X_GET" и 72 "I_GET" данные передаются согласованно, если вы придерживаетесь всех следующих правил обеспечения непротиворечивости:

- **Активный CPU (получатель данных):**
Считывайте область приема в OB, в котором вы вызываете соответствующий SFC. Если вы не можете соблюдать это правило, то считывайте область приема тогда, когда выполнение соответствующего SFC завершено.
- **Пассивный CPU (отправитель данных):**
Не записывайте в область передачи большее количество данных, чем размер массива пассивного CPU (отправителя данных).
- **Пассивный CPU (отправитель данных):**
Записывайте передаваемые данные в область передачи в то время, когда прерывания заблокированы.

Следующая схема показывает пример передачи данных с помощью SFC67 "X_GET" или SFC72 "I_GET", при которой не может гарантироваться непротиворечивость данных, потому что не было соблюдено второе правило обеспечения непротиворечивости: передаются 16 байтов, хотя размер массива пассивного CPU (отправитель данных) составляет только 8 байтов.

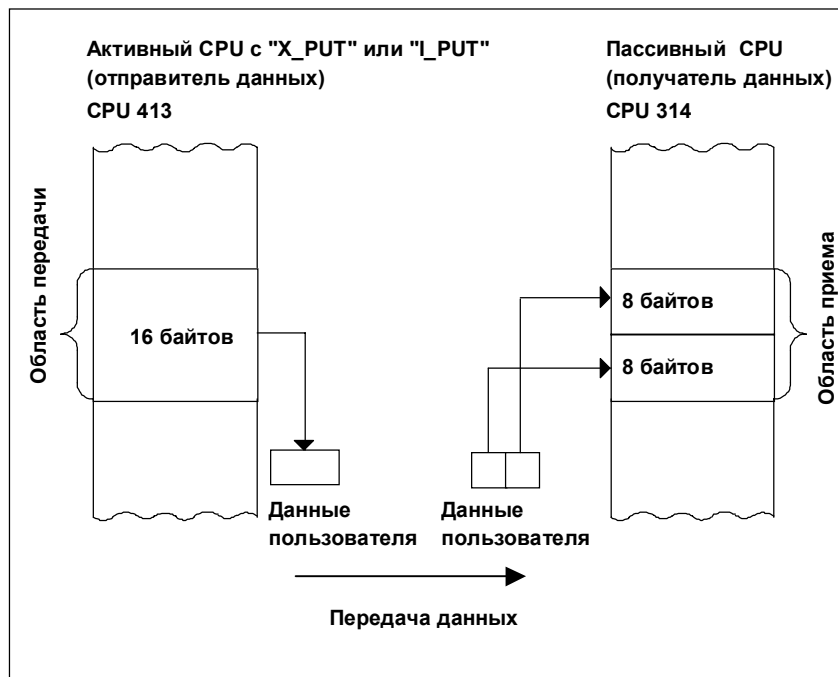


Правила непротиворечивости для SFC типа PUT

С помощью SFC 68 "X_PUT" и 73 "I_PUT" данные передаются согласованно, если вы придерживаетесь всех следующих правил обеспечения непротиворечивости:

- Активный CPU (отправитель данных):
Записывайте в область передачи изнутри OB, в котором вы вызываете соответствующий SFC.
Если вы не можете соблюдать это правило, то записывайте в область передачи только после того, как первый вызов соответствующего SFC завершен.
- Активный CPU (отправитель данных):
Не записывайте в область передачи большее количество данных, чем размер массива пассивного CPU (отправителя данных).
- Пассивный CPU (получатель данных):
Считывайте принимаемые данные из области приема в то время, когда прерывания заблокированы.

Следующая схема показывает пример передачи данных с помощью SFC68 "X_PUT" или SFC73 "I_PUT", при которой не может гарантироваться непротиворечивость данных, потому что не было соблюдено второе правило непротиворечивости: передаются 16 байтов, хотя размер массива пассивного CPU (получателя данных) составляет только 8 байтов.



18.5 Передача данных коммуникационному партнеру вне локальной станции S7 с помощью SFC65 "X_SEND"

Описание

С помощью SFC65 "X_SEND" вы передаете данные коммуникационному партнеру вне локальной станции S7. Данные принимаются коммуникационным партнером с помощью SFC66 "X_RCV".

Вы можете идентифицировать ваши данные с помощью входного параметра REQ_ID. Этот идентификатор (ID) задания передается с данными. Вы можете оценивать этот параметр в коммуникационном партнере, чтобы определить источник данных.

Данные передаются после вызова SFC с REQ=1.

Убедитесь, что область передачи, определенная параметром SD (в передающем CPU) имеет меньший или точно такой же размер, что и область приема, определенная параметром RD (в коммуникационном партнере). Если SD имеет тип данных BOOL, то RD тоже должен иметь тип данных BOOL.

Параметр	Описание	Тип данных	Область данных	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "запрос, подлежащий активизации", см. 18.3
CONT	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "продолжить", см. 18.3
DEST_ID	INPUT	WORD	I, Q, M, D, L, константа	Адресный параметр "ID адресата". Он содержит MPI-адрес коммуникационного партнера. Вы конфигурировали его с помощью STEP 7.
REQ_ID	INPUT	DWORD	I, Q, M, D, L, константа	Идентификатор задания. Он используется для идентификации данных в коммуникационном партнере.
SD	INPUT	ANY	I, Q, M, D	Ссылка на область передачи. Разрешены следующие типы данных: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME и массивы этих типов данных, кроме BOOL.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если в то время, когда функция выполняется, происходит ошибка, то возвращаемое значение содержит соответствующий код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Передача еще не завершена. BUSY=0: Передача завершена или функция передачи не активна.

Входной параметр REQ_ID

Входной параметр REQ_ID используется для идентификации ваших данных. Он передается операционной системой передающего CPU функции SFC66 "X_RCV" в CPU коммуникационного партнера.

Параметр REQ_ID нужен на приемном конце в следующих ситуациях:

- Когда вы вызываете более одного блока SFC65 "X_SEND" с различными параметрами REQ_ID в передающем CPU и передаете данные одному коммуникационному партнеру.
- Когда вы передаете данные одному коммуникационному партнеру из более, чем одного передающего CPU, используя SFC65 "X_SEND".

Оценивая REQ_ID, вы можете сохранять принимаемые данные в различных областях памяти.

Непротиворечивость данных

Когда SFC вызывается впервые, передаваемые данные копируются во внутренний буфер операционной системы. Вы должны удостовериться, что запись в область передачи не происходит прежде, чем завершится первый вызов. Иначе могут быть переданы противоречивые данные.

Запись в область передачи после первого вызова не влияет на непротиворечивость данных.

Информация об ошибках

См. Таблица 18-1 Конкретная информация об ошибках для SFC с 65 по 74.

18.6 Прием данных от коммуникационного партнера вне локальной станции S7 с помощью SFC66 "X_RCV"

Описание

С помощью SFC66 "X_RCV" вы принимаете данные, передаваемые одним или несколькими коммуникационными партнерами вне локальной станции S7 с использованием SFC65 "X_SEND".

С помощью SFC66 "X_RCV"

- Вы можете проверить, были ли данные переданы и ожидают копирования. Данные были введены во внутреннюю очередь операционной системой.
- Вы можете копировать самый старый блок данных из очереди в выбранную область приема.

Параметр	Описание	Тип данных	Область данных	Характеристика
EN_DT	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "разблокировать передачу данных". С помощью значения 0 вы можете проверить, ожидает ли, по крайней мере, один блок данных ввода в область приема. Значение 1 копирует самый старый блок данных из очереди в область рабочей памяти, заданную в RD.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время выполнения функции, происходит ошибка, то возвращаемое значение содержит соответствующий код ошибки. Если ошибки не происходит, то RET_VAL содержит следующее: <ul style="list-style-type: none"> • W#16#7000, если EN_DT=0/1 и NDA=0. В этом случае в очереди нет блоков данных. • длину самого старого блока данных, введенного в очередь, в форме положительного числа в байтах, если EN_DT=0 и NDA=1. • длину блока данных, скопированного в область приема RD, в форме положительного числа в байтах, если EN_DT=1 и NDA=1.
REQ_ID	OUTPUT	DWORD	I, Q, M, D, L	Идентификатор задания SFC "X_SEND", чьи данные стоят первыми в очереди, иными словами, самые старые данные в очереди. Если в очереди нет блоков данных, то REQ_ID имеет значение 0.

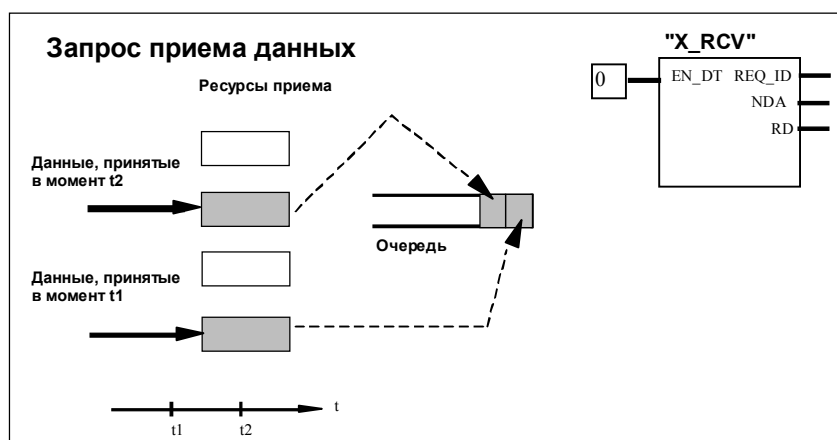
Параметр	Описание	Тип данных	Область данных	Характеристика
NDA	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния "поступили новые данные". NDA=0: <ul style="list-style-type: none"> В очереди нет блоков данных. NDA=1: <ul style="list-style-type: none"> Очередь содержит, по крайней мере, один блок данных. (Вызов SFC66 с EN_DT=0). Самый старый блок данных в очереди был скопирован в программу пользователя. (Вызов SFC66 с EN_DT=1).
RD	OUTPUT	ANY	I, Q, M, D	Ссылка на область принимаемых данных. Разрешены следующие типы данных: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME и массивы этих типов данных, кроме BOOL. Если вы хотите отбросить самый старый блок данных в очереди, то присвойте RD значение NIL.

Индикация приема данных с помощью EN_DT=0

Как только данные поступают от коммуникационного партнера, они вводятся операционной системой в очередь в том порядке, в котором они приняты. Если вы хотите проверить, находится ли в очереди хотя бы один блок данных, то вызовите SFC66 с EN_DT=0 и оцените выходной параметр NDA следующим образом:

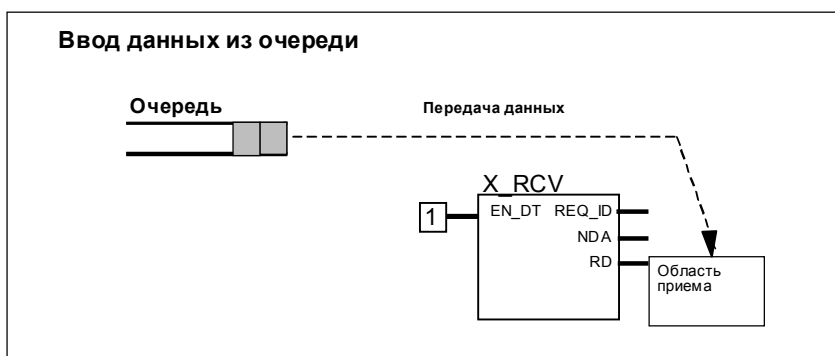
- NDA=0 означает, что очередь не содержит блоков данных. Параметр REQ_ID является несущественным, RET_VAL имеет значение W#16#7000.
- NDA=1 означает, что в очереди имеется, по крайней мере, один блок данных, который может быть извлечен.

В этом случае вы должны также оценить выходной параметр RET_VAL и, если применимо, REQ_ID. RET_VAL содержит длину блока данных в байтах, REQ_ID содержит идентификатор задания передающего блока. Если в очереди имеются несколько блоков данных, то REQ_ID и RET_VAL принадлежат самому старому блоку данных в очереди.



Ввод данных в область приема с помощью EN_DT=1

Когда вы вызываете SFC66 "X_RCV" с EN_DT=1, самый старый блок данных из очереди копируется в область рабочей памяти, заданную посредством RD. RD должен быть по размеру больше или равен области передачи соответствующего SFC65 "X_SEND", определенной параметром SD. Если входной параметр SD имеет тип данных BOOL, то RD также должен иметь тип данных BOOL. Если вы хотите вводить принимаемые данные в разные области, то вы можете запрашивать REQ_ID (вызов SFC с EN_DT = 0) и выбирать подходящий RD в продолженном вызове (при EN_DT=1). Если во время копирования данных ошибки не происходит, то RET_VAL содержит длину скопированного блока данных в байтах, и отправителю передается положительное подтверждение.



Отказ от данных

Если вы не хотите вводить данные из очереди, то присвойте RD значение NIL (см. [/232/](#)). В этом случае отправитель принимает отрицательное подтверждение (RET_VAL в соответствующем SFC65 "X_SEND" имеет значение W#1680B8). RET_VAL в SFC66 "X_RCV" имеет значение 0.

Непротиворечивость данных

Вы не должны считывать область приема, пока задание не завершено. Иначе вы могли бы считать данные, которые не связаны друг с другом (противоречивые данные).

Переключение в режим STOP

Если CPU переключается в режим STOP, то

- все вновь поступающие задания получают отрицательное подтверждение
 - все задания, которые поступили и находятся в очереди, получают отрицательное подтверждение.
 - Если после STOP следует теплый или холодный рестарт, то все блоки данных отбрасываются.
 - Если после STOP следует горячий рестарт (невозможный в S7-300 и S7-400H), то блок данных, принадлежащий самому старому заданию, вводится в программу пользователя при условии, что перед переключением в режим STOP очередь запрашивалась (посредством вызова SFC66 "X_RCV" с EN_DT=0). В противном случае он отбрасывается.
- Все другие блоки данных отбрасываются.

Прерывание соединения

Если соединение прерывается, то принадлежащее соединению задание, которое уже находится в очереди, отбрасывается.

Исключение: Если это задание самое старое в очереди, и вы уже обнаружили его присутствие, вызвав SFC66 "X_RCV" с EN_DT=0, то вы можете ввести его в область приема с помощью EN_DT=1.

Информация об ошибках

См. Таблица 18-1. Конкретная информация об ошибках для SFC с 65 по 74

18.7 Чтение данных из коммуникационного партнера вне локальной станции S7 с помощью SFC67 "X_GET"

Описание

С помощью SFC67 "X_GET" вы можете читать данные из коммуникационного партнера, который находится вне локальной станции S7. В коммуникационном партнере нет соответствующего SFC.

Задание на чтение активизируется после вызова SFC с REQ=1. После этого вы продолжаете вызывать SFC до тех пор, пока посредством BUSY=0 не отобразится прием данных. Тогда RET_VAL содержит длину принятого блока данных в байтах.

Убедитесь, что определенная параметром RD область приема (в приемном CPU) имеет, по крайней мере, такую же длину, как подлежащая чтению область (в коммуникационном партнере), определенная параметром VAR_ADDR. Типы данных RD и VAR_ADDR также должны совпадать.

Параметр	Описание	Тип данных	Область данных	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "запрос, подлежащий активизации", см. 18.3 Общие параметры коммуникационных SFC
CONT	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "продолжать" см. 18.3 Общие параметры коммуникационных SFC.
DEST_ID	INPUT	WORD	I, Q, M, D, L, константа	Адресный параметр "ID адресата". Он содержит MPI-адрес коммуникационного партнера. Вы спроектировали его с помощью STEP 7.
VAR_ADDR	INPUT	ANY	I, Q, M, D	Ссылка на область в CPU партнера, из которой будут читаться данные. Вы должны выбрать тип данных, который поддерживается коммуникационным партнером.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если в то время, когда функция выполняется, происходит ошибка, то возвращаемое значение содержит соответствующий код ошибки. Если ошибки не происходит, то RET_VAL содержит длину блока данных, скопированного в область приема RD, в форме положительного числа байтов.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Прием еще не завершен. BUSY=0: Прием завершен или нет активного задания на прием.

Параметр	Описание	Тип данных	Область данных	Характеристика
RD	OUTPUT	ANY	I, Q, M, D	Ссылка на область приема (область принимаемых данных). Разрешены следующие типы данных: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME и массивы этих типов данных, кроме BOOL. Область приема RD должна иметь, по крайней мере, такую же длину, как определенная параметром VAR_ADDR область, из которой читаются данные. Типы данных RD и VAR_ADDR также должны совпадать

Непротиворечивость данных

См. раздел 18.4.

Переключение в режим STOP

Если CPU переключается в режим STOP, то установленное SFC67 "X_GET" соединение прерывается. От типа выполняемого рестарта зависит, будут ли потеряны принятые данные, расположенные в буфере операционной системы:

- После горячего рестарта (нет в S7-300 и S7-400H) данные копируются в область, определяемую RD.
- После теплого или холодного рестарта данные отбрасываются.

Партнер переключается в режим STOP

Если CPU коммуникационного партнера переключается в режим STOP, то это не влияет на передачу данных с помощью SFC67 "X_GET". Данные могут читаться также и тогда, когда партнер находится в состоянии STOP.

Информация об ошибках

См. Таблица 18-1. Конкретная информация об ошибках для SFC с 65 по 74

18.8 Запись данных в коммуникационном партнере вне локальной станции S7 с помощью SFC68 "X_PUT"

Описание

С помощью SFC68 "X_PUT" вы записываете данные в коммуникационном партнере, который не находится в той же самой локальной станции S7. В коммуникационном партнере нет соответствующего SFC. Задание на запись активизируется после вызова SFC с REQ=1. После этого вы продолжаете вызывать SFC до тех пор, пока не будет принято подтверждение в виде BUSY=0. Убедитесь, что определенная параметром SD область передачи (в передающем CPU) имеет такую же длину, как определенная параметром VAR_ADDR область приема (в коммуникационном партнере). Типы данных SD и VAR_ADDR также должны совпадать.

Параметр	Описание	Тип данных	Область данных	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "запрос, подлежащий активизации", см. 18.3 Общие параметры коммуникационных SFC.
CONT	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "продолжать" см. 18.3 Общие параметры коммуникационных SFC.
DEST_ID	INPUT	WORD	I, Q, M, D, L, константа	Адресный параметр "ID адресата". Он содержит MPI-адрес коммуникационного партнера. Вы спроектировали его с помощью STEP 7.
VAR_ADDR	INPUT	ANY	I, Q, M, D	Ссылка на область в CPU партнера, в которую будут записываться данные. Вы должны выбрать тип данных, который поддерживается коммуникационным партнером.
SD	INPUT	ANY	I, Q, M, D	Ссылка на область локального CPU, содержащую передаваемые данные. Разрешены следующие типы данных: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME и массивы этих типов данных, кроме BOOL. SD должен иметь такую же длину, как параметр VAR_ADDR коммуникационного партнера. Типы данных SD и VAR_ADDR также должны совпадать.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если в то время, когда функция выполняется, происходит ошибка, то возвращаемое значение содержит соответствующий код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Передача еще не завершена. BUSY=0: Передача завершена или функция передачи не активна.

Непротиворечивость данных

См. раздел 18.4 Непротиворечивость данных в случае SFC типа GET и PUT

Переключение в режим STOP

Если CPU переключается в режим STOP, то установленное SFC68 "X_PUT" соединение прерывается. Данные больше не могут передаваться. Если переданные данные уже были скопированы во внутренний буфер до смены режима CPU, то содержимое буфера отбрасывается.

Партнер переключается в режим STOP

Если CPU коммуникационного партнера переключается в режим STOP, то это не влияет на передачу данных с помощью SFC68 "X_PUT". Данные могут записываться также и тогда, когда партнер находится в состоянии STOP.

Информация об ошибках

См. Таблица 18-1. Конкретная информация об ошибках для SFC с 65 по 74

18.9 Прерывание существующего соединения с коммуникационным партнером вне локальной станции S7 с помощью SFC69 "X_ABORT"

Описание

С помощью SFC69 "X_ABORT" вы прерываете соединение, установленное SFC X_SEND, X_GET или X_PUT с коммуникационным партнером, который не находится в той же самой локальной станции S7. Если задание, принадлежащее X_SEND, X_GET или X_PUT, завершено (BUSY = 0), то после вызова SFC69 "X_ABORT" ресурсы соединения, используемые на обоих концах, освобождаются. Если задание, принадлежащее X_SEND, X_GET или X_PUT, еще не завершено (BUSY = 1), то после того, как соединение будет прервано, снова вызовите соответствующий SFC с REQ = 0 и CONT = 0 и затем ожидайте BUSY = 0. Только тогда все ресурсы соединения снова освобождаются. Вы можете вызывать SFC69 "X_ABORT" только на том конце, где расположены SFC "X_SEND", "X_PUT" или "X_GET". Прерывание соединения активизируется посредством вызова SFC с REQ=1.

Параметр	Описание	Тип данных	Область данных	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "запрос, подлежащий активизации", см. 18.3 Общие параметры коммуникационных SFC.
DEST_ID	INPUT	WORD	I, Q, M, D, L, константа	Адресный параметр "ID адресата". Он содержит MPI-адрес коммуникационного партнера. Вы спроектировали его с помощью STEP 7.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если в то время, когда функция выполняется, происходит ошибка, то возвращаемое значение содержит соответствующий код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Прерывание соединения еще не завершено. BUSY=0: Прерывание соединения завершено.

Переключение в режим STOP

Если CPU переключается в режим STOP, то прерывание соединения, запущенное посредством SFC69 "X_ABORT", завершается.

Партнер переключается в режим STOP

Если CPU коммуникационного партнера переключается в режим STOP, то это не влияет на прерывание соединения с помощью SFC69 "X_ABORT". Соединение прерывается.

Информация об ошибках

См. Таблица 18-1. Конкретная информация об ошибках для SFC с 65 по 74

18.10 Чтение данных из коммуникационного партнера в пределах локальной станции S7 с помощью SFC72 "I_GET"

Описание

С помощью SFC72 "I_GET" вы можете читать данные из коммуникационного партнера в той же самой локальной станции S7. Коммуникационный партнер может находиться в центральной стойке, в стойке расширения или быть децентрализованным. Убедитесь, что вы назначили децентрализованных коммуникационных партнеров локальному CPU с помощью STEP 7. В коммуникационном партнере нет соответствующего SFC.

Задание на прием активизируется после вызова SFC с REQ=1. После этого вы продолжаете вызывать SFC до тех пор, пока посредством BUSY=0 не отобразится прием данных. Тогда RET_VAL содержит длину принятого блока данных в байтах.

Убедитесь, что определенная параметром RD область приема (в принимающем CPU) имеет, по крайней мере, такую же длину, как подлежащая чтению область (в коммуникационном партнере), определенная параметром VAR_ADDR. Типы данных RD и VAR_ADDR также должны совпадать.

Параметр	Описание	Тип данных	Область данных	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "запрос, подлежащий активизации", см. 18.3 Общие параметры коммуникационных SFC.
CONT	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "продолжать" см. 18.3 Общие параметры коммуникационных SFC.
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор адресного диапазона в модуле партнера: B#16#54= Периферийный вход (PI) B#16#55= Периферийный выход (PQ) Идентификатором диапазона, принадлежащего смешанному модулю, является меньший из двух адресов. Если адреса одинаковые, то задайте B#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический адрес модуля партнера. Если это смешанный модуль, то задайте меньший из двух адресов.
VAR_ADDR	INPUT	ANY	I, Q, M, D	Ссылка на область в CPU партнера, из которой будут читаться данные. Выберите тип данных, поддерживаемый коммуникационным партнером.

Параметр	Описание	Тип данных	Область данных	Характеристика
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если в то время, когда функция выполняется, происходит ошибка, то возвращаемое значение содержит соответствующий код ошибки. Если ошибки не происходит, то RET_VAL содержит длину блока данных, скопированного в область приема RD, в форме положительного числа байтов.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Прием еще не завершен. BUSY=0: Прием завершен или нет активного задания на прием.
RD	OUTPUT	ANY	I, Q, M, D	Ссылка на область приема (область принимаемых данных). Разрешены следующие типы данных: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME и массивы этих типов данных, кроме BOOL. Область приема RD должна иметь, по крайней мере, такую же длину, как определенная параметром VAR_ADDR область, из которой читаются данные. Типы данных RD и VAR_ADDR также должны совпадать.

Непротиворечивость данных

См. раздел 18.4 Непротиворечивость данных в случае SFC типа GET и PUT.

Переключение в режим STOP

Если CPU переключается в режим STOP, то установленное SFC72 "I_GET" соединение прерывается. От типа выполняемого рестарта зависит, будут ли потеряны принятые данные, расположенные в буфере операционной системы:

- После горячего рестарта (нет в S7-300 и S7-400H) данные копируются в область, определяемую RD.
- После теплого или холодного рестарта данные отбрасываются.

Партнер переключается в режим STOP

Если CPU коммуникационного партнера переключается в режим STOP, то это не влияет на передачу данных с помощью SFC72 "I_GET". Данные могут читаться также и тогда, когда партнер находится в состоянии STOP.

Информация об ошибках

См. Таблица 18-1. Конкретная информация об ошибках для SFC с 65 по 74.

18.11 Запись данных в коммуникационном партнере в пределах локальной станции S7 с помощью SFC73 "I_PUT"

Описание

С помощью SFC73 "I_PUT" вы записываете данные в коммуникационном партнере, который находится в той же самой локальной станции S7. Коммуникационный партнер может быть в центральной стойке, в стойке расширения или быть децентрализованным. Убедитесь, что вы назначили децентрализованных коммуникационных партнеров локальному CPU с помощью STEP 7. В коммуникационном партнере нет соответствующего SFC. Задание на передачу активизируется после вызова SFC при сигнальном уровне 1 на управляющем входе REQ.

Убедитесь, что определенная параметром SD область передачи (в передающем CPU) имеет такую же длину, как определенная параметром VAR_ADDR область приема (в коммуникационном партнере). Типы данных SD и VAR_ADDR также должны совпадать.

Параметр	Описание	Тип данных	Область данных	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "запрос, подлежащий активизации", см. 18.3 Общие параметры коммуникационных SFC.
CONT	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "продолжать" см. 18.3 Общие параметры коммуникационных SFC.
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор адресного диапазона в модуле партнера: V#16#54= Периферийный вход (PI) V#16#55= Периферийный выход (PQ) Идентификатором диапазона, принадлежащего смешанному модулю, является меньший из двух адресов. Если адреса одинаковые, то задайте V#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический адрес модуля партнера. Если это смешанный модуль, то задайте меньший из двух адресов.
VAR_ADDR	INPUT	ANY	I, Q, M, D, L	Ссылка на область в коммуникационном партнере, в которую будут записываться данные. Выберите тип данных, который поддерживается коммуникационным партнером.

Параметр	Описание	Тип данных	Область данных	Характеристика
SD	INPUT	ANY	I, Q, M, D	Ссылка на область локального CPU, содержащую передаваемые данные. Разрешены следующие типы данных: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME и массивы этих типов данных, кроме BOOL. SD должен иметь такую же длину, как параметр VAR_ADDR коммуникационного партнера. Типы данных SD и VAR_ADDR также должны совпадать.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если в то время, когда функция выполняется, происходит ошибка, то возвращаемое значение содержит соответствующий код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Передача еще не завершена. BUSY=0: Передача завершена или функция передачи не активна.

Непротиворечивость данных

См. раздел 18.4 Непротиворечивость данных в случае SFC типа GET и PUT.

Переключение в режим STOP

Если CPU переключается в режим STOP, то установленное SFC73 "I_PUT" соединение прерывается. Данные больше не могут передаваться. Если переданные данные уже были скопированы во внутренний буфер до смены режима CPU, то содержимое буфера отбрасывается.

Партнер переключается в режим STOP

Если CPU коммуникационного партнера переключается в режим STOP, то это не влияет на передачу данных с помощью SFC73 "I_PUT". Данные могут записываться также и тогда, когда партнер находится в состоянии STOP.

Информация об ошибках

См. Таблица 18-1. Конкретная информация об ошибках для SFC с 65 по 74.

18.12 Прерывание существующего соединения с коммуникационным партнером в пределах локальной станции S7 с помощью SFC74 "I_ABORT"

Описание

С помощью SFC74 "I_ABORT" вы прерываете соединение, установленное SFC72 "I_GET" или SFC73 "I_PUT" с коммуникационным партнером в той же самой локальной станции S7. Если задание, принадлежащее I_GET или I_PUT, завершено (BUSY = 0), то после вызова SFC74 "I_ABORT" ресурсы соединения, используемые на обоих концах, освобождаются.

Если задание, принадлежащее I_GET или I_PUT, еще не завершено (BUSY = 1), то после того, как соединение будет прервано, вызовите снова соответствующий SFC с REQ = 0 и CONT = 0 и затем ожидайте BUSY = 0. Только тогда все ресурсы соединения снова освобождаются.

Вы можете вызывать SFC74 "I_ABORT" только на том конце, где расположен SFC "I_PUT" или "I_GET" (иными словами на стороне клиента).

Прерывание соединения активизируется посредством вызова SFC с REQ=1.

Параметр	Описание	Тип данных	Область данных	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "запрос, подлежащий активизации", см. 18.3 Общие параметры коммуникационных SFC.
IOID	INPUT	BYTE	I, Q, M, D, L, константа	Идентификатор адресного диапазона в модуле партнера: V#16#54= Периферийный вход (PI) V#16#55= Периферийный выход (PQ) Идентификатором диапазона, принадлежащего смешанному модулю, является меньший из двух адресов. Если адреса одинаковые, то задайте V#16#54.
LADDR	INPUT	WORD	I, Q, M, D, L, константа	Логический адрес модуля партнера. Если это смешанный модуль, то задайте меньший из двух адресов.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если в то время, когда функция выполняется, происходит ошибка, то возвращаемое значение содержит соответствующий код ошибки.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Прерывание соединения еще не завершено. BUSY=0: Прерывание соединения завершено.

Переключение в режим STOP

Если CPU переключается в режим STOP, то прерывание соединения, запущенное посредством SFC74 "I_ABORT", завершается.

Партнер переключается в режим STOP

Если CPU коммуникационного партнера переключается в режим STOP, то это не влияет на прерывание соединения с помощью SFC74 "I_ABORT". Соединение прерывается.

Информация об ошибках

См. Таблица 18-1. Конкретная информация об ошибках для SFC с 65 по 74.

18.13 Информация об ошибках коммуникационных SFC для не спроектированных соединений S7

Информация об ошибках

"Реальная" информация об ошибках для SFC с 65 по 74, как показано в таблице "Конкретная информация об ошибках для SFC с 65 по 74", может быть классифицирована следующим образом:

Код ошибки (W#16#...)	Объяснение
809х	Ошибка в CPU, в котором выполняется SFC.
80Ах	Постоянная ошибка связи.
80Вх	Ошибка в коммуникационном партнере.
80Сх	Нерегулярная ошибка.

Таблица 18-1. Конкретная информация об ошибках для SFC с 65 по 74.

Код ошибки (W#16# ...)	Объяснение (общее)	Объяснение (для конкретного SFC)
0000	Выполнение завершено без ошибок.	SFC69 "X_ABORT" и SFC74 "I_ABORT": REQ=1 и указанное соединение не установлено. SFC66 "X_RCV": EN_DT=1 и RD=NIL
00ху	-	SFC66 "X_RCV" при NDA=1 и RD<>NIL: RET_VAL содержит длину принятых данных (при EN_DT=0) или длину данных, скопированных в RD (при EN_DT=1). SFC67 "X_GET": RET_VAL содержит длину принятого блока данных. SFC72 "I_GET": RET_VAL содержит длину принятого блока данных.
7000	-	SFC65 "X_SEND", SFC67 "X_GET", SFC68 "X_PUT", SFC69 "X_ABORT", SFC72 "I_GET", SFC73 "I_PUT" и SFC74 "I_ABORT": Вызов с REQ = 0 (вызов без выполнения), BUSY имеет значение 0, передача данных не активна. SFC66 "X_RCV": EN_DT=0/1 и NDA=0
7001	Первый вызов с REQ=1: была запущена передача данных; BUSY имеет значение 1.	-
7002	Промежуточный вызов (REQ не имеет значения): передача данных уже активна; BUSY имеет значение 1.	SFC69 "X_ABORT" и SFC74 "I_ABORT": промежуточный вызов с REQ=1
8090	Указанный адрес назначения коммуникационного партнера недействителен, например: <ul style="list-style-type: none"> • неправильный IOID • неправильный базовый адрес • неправильный адрес MPI (> 126) 	-

Код ошибки (W#16# ...)	Объяснение (общее)	Объяснение (для конкретного SFC)
8092	Ошибка в SD или RD, например: адресация локальной области данных не разрешена.	<p>SFC65 "X_SEND", например:</p> <ul style="list-style-type: none"> • недопустимая длина для SD • SD=NIL недопустимо. <p>SFC66 "X_RCV", например:</p> <ul style="list-style-type: none"> • было принято большее количество данных, чем, может поместиться в области, заданной через RD • RD имеет тип данных BOOL, но принятые данные длиннее байта. <p>SFC67 "X_GET" и SFC72 "I_GET", например:</p> <ul style="list-style-type: none"> • недопустимая длина для RD • длина или тип данных RD не соответствует принятым данным • RD=NIL не разрешено. <p>SFC68 "X_PUT" и SFC73 "I_PUT", например:</p> <ul style="list-style-type: none"> • недопустимая длина для SD • SD=NIL недопустимо.
8095	Блок уже выполняется в более низком классе приоритета.	-
80A0	Ошибка в принятом подтверждении.	SFC68 "X_PUT" и SFC73 "I_PUT": тип данных, указанный в SD передающего CPU, не поддерживается коммуникационным партнером.
80A1	Проблемы связи: вызов SFC после прерывания существующего соединения.	-
80B0	Объект недоступен, например, незагруженный DB.	Возможно в случае SFC67 "X_GET" и SFC68 "X_PUT" и SFC72 "I_GET" и SFC73 "I_PUT".
80B1	Ошибка в указателе ANY. Неправильная длина области передаваемых данных.	-
80B2	Аппаратная ошибка: модуль не существует. <ul style="list-style-type: none"> • Конфигурированный слот не занят. • Фактический тип модуля не соответствует ожидаемому типу. • Децентрализованная периферия недоступна.. • Нет записи для модуля в соответствующем SDB. 	Возможно в случае SFC67 "X_GET" и SFC68 "X_PUT" и SFC72 "I_GET" и SFC73 "I_PUT".
80B3	Данные могут или только читаться, или только записываться, например, DB, защищенный от записи.	Возможно в случае SFC67 "X_GET" и SFC68 "X_PUT" и SFC72 "I_GET" и SFC73 "I_PUT".
80B4	Ошибка типа данных в указателе ANY, или массив заданного типа не разрешен.	SFC67 "X_GET" и SFC68 "X_PUT" и SFC72 "I_GET" и SFC73 "I_PUT": тип данных, заданный в VAR_ADDR, не поддерживается коммуникационным партнером.
80B5	Выполнение отклонено из-за недопустимого режима.	Возможно в случае SFC65 "X_SEND".

Код ошибки (W#16# ...)	Объяснение (общее)	Объяснение (для конкретного SFC)
80B6	Принятое подтверждение содержит неизвестный код ошибки.	-
80B7	Тип данных и/или длина переданных данных не соответствуют области в CPU партнера, в которой они должны записываться.	Возможно в случае SFC68 "X_PUT" и SFC73 "I_PUT".
80B8	-	SFC65 "X_SEND": SFC66 "X_RCV" коммуникационного партнера не позволил принять данные (RD=NIL).
80B9	-	SFC65 "X_SEND": Блок данных был идентифицирован коммуникационным партнером (вызов SFC66 "X_RCV" с EN_DT=0), он еще не был введен в программу пользователя, потому что партнер находится в режиме STOP.
80BA	Ответ коммуникационного партнера не помещается в кадре связи.	-
80C0	Указанное соединение используется другим заданием.	-
80C1	Недостаток ресурсов в CPU, в котором выполняется SFC, например: В модуле уже выполняется максимальное количество разных заданий на передачу. Ресурс соединения использован, например, для приема данных.	-
80C2	Временный недостаток ресурсов в коммуникационном партнере, например: <ul style="list-style-type: none"> • Коммуникационный партнер в настоящее время обрабатывает максимальное число заданий. • Требуемые ресурсы, память, и т.д. используются. • Недостаточно рабочей памяти (сжать память). 	-
80C3	Ошибка в установлении соединения, например: <ul style="list-style-type: none"> • Локальная станция S7 не подключена к подсети MPI. • Вы обратились в подсети MPI к своей собственной станции. • Коммуникационный партнер больше не доступен. • Временный недостаток ресурсов в коммуникационном партнере. 	-

19 Создание сообщений, связанных с блоками

19.1 Введение в создание сообщений, связанных с блоками, с помощью SFB

SFB для создания сообщений, связанных с блоками

Вы можете создать сообщение, связанное с блоком, вызывая в своей программе один из следующих SFB:

- SFB36 "NOTIFY"
- SFB33 "ALARM"
- SFB35 "ALARM_8P"
- SFB34 "ALARM_8"

Эти SFB имеют следующие свойства:

- Каждая обнаруженная смена фронта приводит к передаче сообщения.
- После выполнения блока сопутствующие значения (входы SD_i) полностью считываются и ставятся в соответствие сообщению (см. "Параметры передачи и приема" в разделе 17.2).
С точки зрения непротиворечивости относительно высокоприоритетных классов следующие сопутствующие значения непротиворечивы:
простые типы данных (бит, байт, слово и двойное слово)
массив байтового типа с длиной, не превышающей максимального значения, зависящего от конкретного CPU (см. **171**, **101**).
- С помощью параметров состояния DONE, ERROR и STATUS вы наблюдаете обработку состояния блока (см. "Параметры состояния" в разделе 17.2).

Замечание

Параметры ID и EV_ID оцениваются только при первом вызове блока (фактические параметры или предварительно определенные значения из экземпляра). Для обработки сообщений требуется до 480 байтов рабочей памяти, в зависимости от длины и количества параметров SD_i.

Регистрация устройств отображения

Прежде чем SFB для создания сообщений, связанных с блоками, смогут передать сообщение при обнаружении смены фронта, для сообщений, связанных с блоками должно быть зарегистрировано хотя бы одно устройство отображения. Вы можете это проверить с помощью SFC62 "CONTROL".

Хранение сообщений

Во избежание потери сообщений при большой рабочей нагрузке на систему связи, каждый SFB, генерирующий сообщение, может запомнить в буфере два сообщения.

Если сообщения все же теряются, то Вам об этом сообщается через выходные параметры ERROR и STATUS (ERROR = 0, STATUS = 11). В следующий раз, когда сообщение может быть послано, об этой потере информируются и зарегистрированные устройства отображения.

Квитирование сообщений

Используется концепция централизованного квитирования. Когда вы квитировали сообщение на устройстве отображения, информация о квитировании сначала посылается в CPU, генерировавшее это сообщение. Отсюда информация о квитировании распределяется всем станциям, зарегистрированным для этой цели.

Вы квитируете сигнал, а не отдельное сообщение. Например, если было сообщено о нескольких нарастающих фронтах сигнала, и вы квитируете поступившее событие, то все предшествующие события с таким же номером сообщения считаются квитированными.

Отображение квитирования

SFB36 "NOTIFY" не отображает квитирование. Вы можете проверить выходные параметры ACK_UP и ACK_DN блока SFB33 "ALARM" и выходные параметры ACK_STATE блоков SFB 35 "ALARM_8P" и "ALARM_8". Эти выходы обновляются в момент вызова блока, если параметр управления EN_R имеет значение 1.

Блокировка и разблокировка сообщений

В некоторых ситуациях может быть полезно подавлять сообщения, например, когда сигнал "пульсирует" или когда вы запускает свою систему. Поэтому у вас есть возможность блокировать или разблокировать сообщения на устройстве отображения или в своей программе. Блокировка/ разблокировка относится ко всем станциям, зарегистрированным для конкретного сообщения. Сообщение остается заблокированным, пока оно снова не будет разблокировано.

Информацию о заблокированных сообщениях вы получаете через выходные параметры ERROR и STATUS (ERROR = 1, STATUS = 21).

Обновление сообщений

С помощью обновления сообщения вы можете считывать текущие состояния сигнала и квитирования на устройстве отображения. Во время обновления все зарегистрированные станции продолжают получать сообщения, для которых они зарегистрированы.

Количество передаваемых данных

Данные, передаваемые с помощью сопутствующих значений SD_i системных функциональных блоков NOTIFY, ALARM и ALARM_8P, не должны превышать максимальную длину. Максимальная длина данных рассчитывается следующим образом:

Максимальная длина = $\min(\text{pdu_local}, \text{pdu_remote}) - 44 - 4 * \text{Количество используемых параметров SD_i}$,

где:

- pdu_local – это максимальная длина массивов данных местного CPU (SSL_ID W#16#0131, INDEX 1, переменная pdu)
- pdu_remote – это максимальная длина массивов данных устройств отображения.

Пример:

CPU 414-1 передает сообщения на PG 760 (через MPI).

Используются сопутствующие значения SD_1, SD_2 и SD_3.

pdu_local = 480 байтов, pdu_remote = 480 байтов

Количество используемых параметров SD_i: 3

Тогда:

Максимальная длина = $\min(480, 480) - 44 - 4 * 3 = 480 - 44 - 12 = 424$

Максимальная длина данных, которые могут быть переданы, составляет 424 байта на SFB.

19.2 Создание с помощью SFB36 "NOTIFY" сообщений, связанных с блоками, без квитирования

Описание

SFB36 "NOTIFY" контролирует сигнал. Он генерирует сообщение как при нарастающем фронте (приходящее событие), так и при падающем фронте (уходящее событие). Вы можете послать с этим сообщением до десяти сопутствующих значений. Это сообщение передается всем зарегистрированным для него станциям. При первом вызове SFB сообщение передается с текущим состоянием сигнала.

Сопутствующие значения опрашиваются при обнаружении фронта и ставятся в соответствие сообщению. Если вы квитируете сообщение на зарегистрированном устройстве отображения, то об этом сообщается всем остальным зарегистрированным устройствам отображения. Блок NOTIFY не информируется об этом квитировании.

SFB36 "NOTIFY" может временно хранить один нарастающий и один падающий фронт сигнала. Любые последующие изменения сигнала игнорируются. Эта потеря сообщений отображается с помощью выходных параметров ERROR и STATUS (ERROR = 0, STATUS = 11); зарегистрированные устройства отображения также информируются об этой потере. SFB36 "NOTIFY" соответствует стандарту IEC 1131-5.

Параметр	Описание	Тип данных	Область памяти	Характеристика
SIG	INPUT	BOOL	I, Q, M, D, L	Контролируемый сигнал
ID	INPUT	WORD	I, Q, M, D, L, константа	Канал данных для сообщений: W#16#EEEE. ID оценивается только при первом вызове.
EV_ID	INPUT	DWORD	I, Q, M, D, L, константа	Номер сообщения (0 не разрешен). EV_ID оценивается только при первом вызове. После этого при каждом вызове SFB36 с соответствующим экземпляром DB используется номер сообщения из первого вызова. При назначении номеров сообщений используйте функции проектирования сообщений. Это гарантирует непротиворечивость номеров сообщений.
SEVERITY	INPUT	WORD	I, Q, M, D, L, константа	Весовой коэффициент события: возможные значения: от 0 до 127 (0 означает наивысший вес)
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE. Генерирование сообщения завершено
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS
SD_i, 1 ≤ i ≤ 10	IN_OUT	ANY	I, Q, M, D, T, C	Сопутствующее значение: разрешены следующие типы данных: BOOL (не разрешен битовый массив) BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER.

Информация об ошибках

Следующая таблица содержит всю информацию об ошибках, относящуюся к SFB36, которая может быть выведена через параметры ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	22	<ul style="list-style-type: none"> Ошибка в указателе на сопутствующие значения SD_i: <ul style="list-style-type: none"> -относительно длины или типа данных -сопутствующие значения в памяти пользователя недоступны, например, из-за DB, который был удален, или из-за ошибки длины области Активизированное сообщение передается без сопутствующих значений Выбранный вами параметр для SEVERITY больше допустимого диапазона. Активизированное сообщение будет послано с SEVERITY=127.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи: соединение прервано или не зарегистрировано
1	4	При первом вызове: <ul style="list-style-type: none"> заданный EV_ID находится вне допустимого диапазона или указатель ANY для SD_i имеет формальную ошибку превышен максимально возможный размер памяти, который может быть послан для CPU посредством SFB36
1	10	Доступ к локальной памяти пользователя невозможен (например, обращение к DB, который был удален)
1	12	При вызове SFB был задан: <ul style="list-style-type: none"> неинициализированный экземплярный DB экземплярный DB, который не принадлежит SFB 36 совместно используемый, а не экземплярный DB
1	18	EV_ID уже используется одним из SFB 33 – 36.
1	20	Недостаточно памяти.
1	21	Сообщение с заданным EV_ID заблокировано.

19.3 Создание с помощью SFB33 "ALARM" сообщений, связанных с блоками, с квитированием

Описание

SFB33 "ALARM" контролирует сигнал. Он генерирует сообщение как при нарастающем фронте (приходящее событие), так и при падающем фронте (уходящее событие). С сообщением вы можете посылать до десяти сопутствующих значений. Сообщение посылается всем зарегистрированным для этого станциям.

При первом вызове SFB сообщение передается с текущим состоянием сигнала.

Выход ACK_UP сбрасывается, если имеет место нарастающий фронт и генерация сообщения завершена (DONE = 1). Он устанавливается при квитировании вами наступающего события из зарегистрированного устройства отображения. Ситуация для выхода ACK_DN аналогична: он сбрасывается, когда имеет место падающий фронт и генерирование сообщения завершено (DONE = 1). Он устанавливается при квитировании вами уходящего события из зарегистрированного устройства отображения. Как только ваше квитирование получено с зарегистрированного устройства отображения, информация о квитировании передается всем зарегистрированным для этого устройствам отображения.

SFB33 "ALARM" может временно хранить один нарастающий и один падающий фронт сигнала. Любые следующие изменения сигнала игнорируются. Эта потеря сообщений отображается с помощью выходных параметров ERROR и STATUS (ERROR = 0, STATUS = 11); зарегистрированные устройства отображения также информируются об этой потере.

SFB33 "ALARM" соответствует стандарту IEC 1131–5.

Параметр	Описание	Тип данных	Область памяти	Характеристика
EN_R	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "разблокировано для приема", определяющий, будут ли выходы ACK_UP и ACK_DN обновляться при первом вызове блока (EN_R=1) или нет (EN_R=0). При EN_R=0 SFB33 "ALARM" ведет себя так же, как SFB36 "NOTIFY". Выходные параметры ACK_UP и ACK_DN в этом случае остаются неизменными.
SIG	INPUT	BOOL	I, Q, M, D, L	Контролируемый сигнал
ID	INPUT	WORD	I, Q, M, D, L, константа	Канал данных для сообщений: W#16#EEEE. ID оценивается только при первом вызове.

Параметр	Описание	Тип данных	Область памяти	Характеристика
EV_ID	INPUT	DWORD	I, Q, M, D, L, константа	Номер сообщения (0 не разрешен) EV_ID оценивается только при первом вызове. После этого при каждом вызове SFB 33 с соответствующим экземпляром DB используется номер сообщения из первого вызова. При назначении номера сообщения используйте функции проектирования сообщений. Это гарантирует непротиворечивость номеров сообщений.
SEVERITY	INPUT	WORD	I, Q, M, D, L, константа	Весовой коэффициент события. Возможные значения: от 0 до 127 (0 означает наибольший вес)
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: генерирование сообщения завершено
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS
ACK_DN	OUTPUT	BOOL	I, Q, M, D, L	Уходящее сообщение было квитировано на устройстве отображения
ACK_UP	OUTPUT	BOOL	I, Q, M, D, L	Поступающее сообщение было квитировано на устройстве отображения
SD_i, 1 ≤ i ≤ 10	IN_OUT	ANY	I, Q, M, D, T, C	Сопутствующее значение. Разрешены следующие типы данных: BOOL (не разрешен: битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER.

Информация об ошибках

Следующая таблица содержит всю информацию об ошибках, относящуюся к SFB33, которая может быть выведена с помощью параметров ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	22	<ul style="list-style-type: none"> Ошибка в указателе на сопутствующие значения SD_i: <ul style="list-style-type: none"> - Относительно длины или типа данных - Сопутствующие значения в памяти пользователя недоступны, например, из-за DB, который был удален, или из-за ошибки длины области - Активизированное сообщение послано без сопутствующих значений

ERROR	STATUS (десятичное число)	Объяснение
		<ul style="list-style-type: none"> Выбранный вами фактический параметр для SEVERITY выше допустимого диапазона. Активизированное сообщение будет послано с SEVERITY=127.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи: соединение прервано или нет регистрации
1	4	<ul style="list-style-type: none"> При первом вызове: <ul style="list-style-type: none"> - заданный EV_ID находится вне допустимого диапазона или - указатель ANY для SD_i содержит формальную ошибку - превышен максимально возможный размер памяти, который может быть послан для CPU посредством SFB33
1	10	Доступ к локальной памяти пользователя невозможен (например, обращение к DB, который был удален)
1	12	При вызове SFB был задан: <ul style="list-style-type: none"> - неинициализированный экземплярный DB - экземплярный DB, не принадлежащий SFB33 - совместно используемый, а не экземплярный DB
1	18	EV_ID уже используется одним из SFB 33 – 36.
1	20	Недостаточно памяти
1	21	Сообщение с заданным EV_ID заблокировано.

Замечание

После первого вызова блока выходы ACK_UP и ACK_DN имеют значение 1, и принимается, что предыдущее значение входа SIG было равно 0.

19.4 Создание сообщений, связанных с блоками, с сопутствующими значениями для восьми сигналов с помощью SFB35 "ALARM_8P"

Описание

SFB35 "ALARM_8P" является прямым расширением SFB33 "ALARM", допускающим восемь сигналов.

Сообщение генерируется, когда обнаруживается смена фронта у одного или более сигналов (исключение: при первом вызове блока сообщение передается всегда). Для всех восьми сигналов существует общий номер сообщения, которое на устройстве отображения разбивается на восемь отдельных сообщений. Вы можете квитировать каждое отдельное сообщение независимо друг от друга или все восемь отдельных сообщений сразу.

Для включения в свою программу состояния квитирования для отдельных сообщений вы можете использовать выходной параметр ACK_STATE. Если вы блокируете или разблокируете сообщение блока ALARM_8P, то это всегда влияет на весь блок ALARM_8P. Блокировка и разблокировка отдельных сигналов невозможна.

SFB35 "ALARM_8P" может временно запоминать два сигнала. Любое следующее затем изменение сигнала игнорируется. Эта потеря сообщений отображается выходными параметрами ERROR и STATUS (ERROR = 0, STATUS = 11); зарегистрированные устройства отображения также информируются об этой потере.

Параметр	Описание	Тип данных	Область памяти	Характеристика
EN_R	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "разблокировано для приема", который определяет, обновляется ли выход ACK_STATE при вызове блока (EN_R=1) или нет (EN_R=0).
SIG_i, 1 • i • 8	INPUT	BOOL	I, Q, M, D, L	i-й контролируемый сигнал
ID	INPUT	WORD	I, Q, M, D, L, константа	Канал данных для сообщений: W#16#EEEE. ID оценивается только при первом вызове.
EV_ID	INPUT	DWORD	I, Q, M, D, L, константа	Номер сообщения (0 не разрешен). EV_ID оценивается только при первом вызове. После этого при каждом вызове SFB 35 с соответствующим экземплярным DB используется номер сообщения из первого вызова. При назначении номера сообщения используйте функции проектирования сообщений. Это гарантирует непротиворечивость номеров сообщений.

Параметр	Описание	Тип данных	Область памяти	Характеристика
SEVERITY	INPUT	WORD	I, Q, M, D, L, константа	Весовой коэффициент события. Возможные значения: от 0 до 127 (0 означает наивысший вес)
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: генерирование сообщения завершено
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS
ACK_STATE	OUTPUT	WORD	I, Q, M, D, L	Битовый массив с текущим состоянием квитирования всех восьми сообщений: <ul style="list-style-type: none"> бит 2^0 : поступающее событие в SIG_1 было квитировано бит 2^7 : поступающее событие в SIG_8 было квитировано бит 2^8 : уходящее событие в SIG_1 было квитировано бит 2^{15} : уходящее событие в SIG_8 было квитировано
SD_j, $1 \leq j \leq 10$	IN_OUT	ANY	I, Q, M, D, T, C	j-е сопутствующее значение Эти значения относятся ко всем сообщениям. Разрешены следующие типы данных: BOOL (не разрешен: битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER.

Информация об ошибках

Следующая таблица содержит всю информацию об ошибках, относящуюся к SFB35, которая может быть выведена с помощью параметров ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	22	<ul style="list-style-type: none"> Ошибка в указателе на сопутствующие значения SD_i: <ul style="list-style-type: none"> - Относительно длины или типа данных - Сопутствующие значения в памяти пользователя недоступны, например, из-за DB, который был удален, или из-за ошибки длины области - Активизированное сообщение послано без сопутствующих значений Выбранный вами фактический параметр для SEVERITY выше допустимого диапазона. Активизированное сообщение будет послано с SEVERITY=127.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи: соединение прервано или нет регистрации

ERROR	STATUS (десятичное число)	Объяснение
1	4	При первом вызове: <ul style="list-style-type: none"> заданный EV_ID находится вне допустимого диапазона или указатель ANY для SD_i содержит формальную ошибку превышен максимально возможный размер памяти, который может быть послан для CPU посредством SFB35
1	10	Доступ к локальной памяти пользователя невозможен (например, обращение к DB, который был удален)
1	12	При вызове SFB был задан: <ul style="list-style-type: none"> неинициализированный экземплярный DB экземплярный DB, не принадлежащий SFB35 совместно используемый, а не экземплярный DB
1	18	EV_ID уже используется одним из SFB 33 – 36.
1	20	Недостаточно памяти.
1	21	Сообщение с заданным EV_ID заблокировано.

Замечание

После первого вызова блока все биты выхода ACK_STATE установлены, и принимается, что предыдущие значения входов SIG_i, $1 \leq i \leq 8$ были равны 0.

19.5 Создание сообщений, связанных с блоками, без сопутствующих значений для восьми сигналов с помощью SFB34 "ALARM_8"

Описание

SFB34 "ALARM_8" идентичен SFB35 "ALARM_8P" за исключением того, что у него нет сопутствующих значений SD_1 – SD_10.

Параметр	Описание	Тип данных	Область памяти	Характеристика
EN_R	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "разблокировано для приема", который определяет, обновляется ли выход ACK_STATE при вызове блока (EN_R=1) или нет (EN_R=0).
SIG_i, 1≤i≤8	INPUT	BOOL	I, Q, M, D, L	i-й контролируемый сигнал
ID	INPUT	WORD	I, Q, M, D, L, константа	Канал данных для сообщений: W#16#EEEE. ID оценивается только при первом вызове.
EV_ID	INPUT	DWORD	I, Q, M, D, L, константа	Номер сообщения (0 не разрешен). EV_ID оценивается только при первом вызове. После этого при каждом вызове SFB34 35 с соответствующим экземплярным DB используется номер сообщения из первого вызова. При назначении номера сообщения используйте функции проектирования сообщений. Это гарантирует непротиворечивость номеров сообщений.
SEVERITY	INPUT	WORD	I, Q, M, D, L, константа	Весовой коэффициент события. Возможные значения: от 0 до 127 (0 означает наивысший вес)
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: генерирование сообщения завершено
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS
ACK_STAT E	OUTPUT	WORD	I, Q, M, D, L	Битовый массив с текущим состоянием квитирования всех восьми сообщений: <ul style="list-style-type: none"> бит 2⁰ : поступающее событие в SIG_1 было квитировано бит 2⁷ : поступающее событие в SIG_8 было квитировано бит 2⁸ : уходящее событие в SIG_1 было квитировано бит 2¹⁵: уходящее событие в SIG_8 было квитировано

Информация об ошибках

Следующая таблица содержит всю информацию об ошибках, относящуюся к SFB34, которая может быть выведена с помощью параметров ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	22	Выбранный вами фактический параметр для SEVERITY выше допустимого диапазона. Активизированное сообщение будет послано с SEVERITY = 127.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи: соединение прервано или нет регистрации
1	4	При первом вызове: заданный EV_ID находится вне допустимого диапазона.
1	10	Доступ к локальной памяти пользователя невозможен (например, обращение к DB, который был удален)
1	12	При вызове SFB был задан: <ul style="list-style-type: none"> • неинициализированный экземплярный DB • экземплярный DB, не принадлежащий SFB34 • совместно используемый, а не экземплярный DB
1	18	EV_ID уже используется одним из SFB 33 – 36.
1	20	Недостаточно памяти.
1	21	Сообщение с заданным EV_ID заблокировано

Замечание

После первого вызова блока все биты выхода ACK_STATE установлены, и принимается, что предыдущие значения входов SIG_i, $1 \leq i \leq 8$ были равны 0.

19.6 Передача архивных данных с помощью SFB37 "AR_SEND"

Описание

SFB37 "AR_SEND" передает архивные данные системам взаимодействия с оператором, зарегистрированным для этой цели. Эти системы сообщают CPU о номере соответствующего архива в зарегистрированном сообщении. В зависимости от доступной памяти в CPU и используемой адресной области архивные данные могут иметь длину до 65534 байтов. В структуре архивных данных могут быть приняты во внимание умолчания используемой вами системы взаимодействия с оператором.

Передача данных активизируется положительным фронтом на входе управления REQ после вызова блока. Начальный адрес передаваемых архивных данных задается через SD_1, длина массива данных – через LEN. Передача данных происходит асинхронно по отношению к исполнению программы пользователя. Успешное завершение передачи отображается параметром состояния DONE, имеющим значение 1. Нарастающий фронт на входе управления R прекращает передачу данных.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	Параметр управления "запрос"
R	INPUT	BOOL	I, Q, M, D, L, константа	Параметр управления "сброс": прерывание текущего задания
ID	INPUT	WORD	I, Q, M, D, L, константа	Канал данных для сообщений: W#16#EEEE. ID оценивается только при первом вызове.
AR_ID	INPUT	DWORD	I, Q, M, D, L, константа	Номер архива (0 не разрешен). AR_ID оценивается только при первом вызове. После этого при каждом вызове SFB 37 с соответствующим экземплярным DB используется номер архива из первого вызова. При назначении номера архива используйте функции проектирования сообщений. Это гарантирует непротиворечивость номеров архивов.
DONE	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния DONE: передача завершена
ERROR	OUTPUT	BOOL	I, Q, M, D, L	Параметр состояния ERROR
STATUS	OUTPUT	WORD	I, Q, M, D, L	Параметр состояния STATUS
SD_1	IN_OUT	ANY	I, Q, M, D, T, C	Указатель на архивные данные. Данные о длине не оцениваются. Разрешены следующие типы данных: BOOL (не разрешен: битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER. Структура архивных данных должна соответствовать целевой системе.
LEN	IN_OUT	WORD	I, Q, M, D, L	Длина передаваемого массива данных в байтах

Информация об ошибках

Следующая таблица содержит всю информацию об ошибках, относящуюся к SFB37, которая может быть выведена с помощью параметров ERROR и STATUS.

ERROR	STATUS (десятичное число)	Объяснение
0	11	Предупреждение: новое задание не действует, так как предыдущее задание еще не завершено.
0	25	Обмен данными начался. Задание обрабатывается.
1	1	Проблемы связи
1	2	Отрицательное подтверждение, функция не может быть выполнена
1	3	Нет регистрации для заданного AR_ID.
1	4	<ul style="list-style-type: none"> Ошибка в указателе архивных данных SD_1 в отношении длины данных или типа данных. При первом вызове заданный AR_ID находится вне допустимого диапазона.
1	5	Было выполнено требование сброса.
1	7	Задание RESET [сброс] не имеет смысла, так как текущая функция завершена или не активизирована (блок в неправильном состоянии).
1	10	Доступ к локальной памяти приложения невозможен (например, обращение к DB, который был удален).
1	12	При вызове SFB был задан: <ul style="list-style-type: none"> неинициализированный экземплярный DB экземплярный DB, который не принадлежит SFB37 совместно используемый, а не экземплярный DB
1	18	AR_ID уже использовался блоком SFB37.
1	20	Недостаточно памяти.

19.7 Блокировка сообщений, связанных с блоком, сообщений, связанных с символом, и сообщений о групповом состоянии с помощью SFC10 "DIS_MSG"

Описание

С помощью SFC10 "DIS_MSG" (disable message [блокировать сообщение]) вы можете блокировать сообщения, связанные с блоком, генерируемые посредством SFB, сообщения, связанные с символом (SCAN), и сообщения о групповом состоянии. Блокируемые сообщения выбирают, используя входные параметры MODE и MESGN. Вызов SFC10 "DIS_MSG" и успешная блокировка сообщения возможны только тогда, когда блокировка сообщения еще не активизирована с помощью SFC10.

Сообщения, готовые к передаче на момент вызова SFC10, но все еще находящиеся во внутреннем буфере, больше не могут блокироваться и передаются. Блокированное сообщение отображается на выходах ERROR и STATUS в SFB "NOTIFY", "ALARM", "ALARM_8P" и "ALARM_8".

Блокировку сообщения запускают, присваивая при вызове SFC10 входному параметру REQ значение 1.

Как функционирует SFC10

Блокировка выполняется асинхронно, другими словами, она может быть активной на протяжении нескольких вызовов SFC10:

- При первом вызове (REQ =1) SFC10 проверяет входные параметры и пытается занять требуемые системные ресурсы. В случае успеха в RET_VAL вводится значение W#16#7001, устанавливается BUSY и запускается блокировка сообщения.
В случае неудачи в RET_VAL вводится информация об ошибке, и задание завершается. В этом случае BUSY не должен оцениваться.
- Если тем временем имеют место дальнейшие вызовы, то в RET_VAL вводится значение W#16#7002 (задание все еще выполняется CPU) и устанавливается BUSY. Дальнейшие вызовы не влияют на текущее задание.
- При последнем вызове SFB в RET_VAL вводится значение W#16#0000, если не было ошибок. Тогда BUSY имеет значение 0. Если произошла ошибка, то в RET_VAL вводится информация об ошибке и BUSY не должен оцениваться.

См. Таблица 2-1. Связь между вызовом, REQ, RET_VAL и BUSY во время выполнения задания.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	REQ = 1: запустить блокировку.
MODE	INPUT	BYTE	I, Q, M, D, L, константа	Параметр для выбора блокируемых сообщений, см. следующую таблицу.
MESGN	INPUT	DWORD	I, Q, M, D, L, константа	Номер сообщения, существенный только тогда, когда MODE установлен в 5, 6, 7. Это позволяет блокировать отдельное сообщение.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках, см. таблицу "Информация об ошибках".
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: блокировка еще не была отменена.

Входной параметр MODE

Следующая таблица показывает допустимые значения входного параметра MODE.

Значение	Характеристика
0	Все сообщения CPU, связанные с блоком, связанные с символом, и все сообщения о групповом состоянии, генерируемые с помощью SFB
1	Все связанные с блоком сообщения CPU, генерируемые с помощью SFB, иными словами, все сообщения, генерируемые SFB "NOTIFY", "ALARM", "ALARM_8P" и "ALARM_8"
2	Все сообщения CPU о групповом состоянии
3	Все связанные с символом сообщения CPU (SCAN)
5	Отдельное сообщение класса "сообщения, связанные с символом"
6	Отдельное сообщение класса "сообщения, связанные с блоком"
7	Отдельное сообщение класса "сообщения о групповом состоянии"

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Блокировка завершилась без ошибки.
7000	REQ = 0 при первом вызове: блокировка не была активизирована.
7001	REQ = 1 при первом вызове: блокировка была запущена.
7002	Дальнейший вызов: блокировка уже активна.
8081	Ошибка при обращении к параметру.
8082	MODE имеет недопустимое значение.
8083	Номер сообщения находится вне допустимого диапазона значений.
8084	Отсутствует регистрация для сообщения (сообщений), заданного с помощью MODE и, возможно, MESGN.
80C3	Сообщение (сообщения), блокируемое в MODE и, возможно, MESGN, не может блокироваться в настоящее время, так как SFC10 уже блокирует сообщения.

19.8 Разблокировка сообщений, связанных с блоком, сообщений, связанных с символом, и сообщений о групповом состоянии с помощью SFC9 "EN_MSG"

Описание

С помощью SFC9 "EN_MSG" (enable message [разблокировать сообщение]) вы можете разблокировать сообщения, связанные с блоком, сообщения, связанные с символом, и сообщения о групповом состоянии, которые ранее были заблокированы. Вы блокировали эти сообщения или в устройстве отображения, или с помощью SFC10 "DIS_MSG". Сообщения, подлежащие разблокированию, задают, используя входные параметры MODE и MESGN. Успешная разблокировка сообщений с помощью SFC9 "EN_MSG" возможна только тогда, когда SFC9 уже активно не разблокирует сообщения. Функцию разблокировки запускают, присваивая при вызове SFC9 входному параметру REQ значение 1.

Как функционирует SFC9

Разблокировка выполняется асинхронно, другими словами, она может быть активной на протяжении нескольких вызовов SFC9:

- При первом вызове (REQ = 1) SFC9 проверяет входные параметры и пытается занять требуемые системные ресурсы. В случае успеха в RET_VAL вводится значение W#16#7001, устанавливается BUSY и запускается разблокировка сообщения. В случае неудачи в RET_VAL вводится информация об ошибке, и задание завершается. В этом случае BUSY не должен оцениваться.
- Если тем временем имеют место дальнейшие вызовы, то в RET_VAL вводится значение W#16#7002 (задание все еще выполняется CPU) и устанавливается BUSY. Дальнейшие вызовы не влияют на текущее задание.
- При последнем вызове SFB в RET_VAL вводится значение W#16#0000, если не было ошибок. Тогда BUSY имеет значение 0. Если произошла ошибка, то в RET_VAL вводится информация об ошибке и BUSY не должен оцениваться.

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L	REQ = 1: запустить разблокировку.
MODE	INPUT	BYTE	I, Q, M, D, L, константа	Параметр для выбора разблокируемых сообщений, см. следующую таблицу.
MESGN	INPUT	DWORD	I, Q, M, D, L, константа	Номер сообщения, существенный только тогда, когда MODE установлен в 5, 6, 7. Это позволяет разблокировать отдельное сообщение.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибках, см. таблицу.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY = 1: разблокировка еще не была отменена.

Входной параметр MODE

Следующая таблица показывает допустимые значения входного параметра MODE.

Значение	Характеристика
0	Все сообщения CPU, связанные с блоком, связанные с символом, и все сообщения CPU о групповом состоянии, генерируемые с помощью SFB
1	Все связанные с блоком сообщения CPU, генерируемые с помощью SFB, иными словами, все сообщения, генерируемые SFB "NOTIFY", "ALARM", "ALARM_8P" и "ALARM_8"
2	Все сообщения CPU о групповом состоянии
3	Все связанные с символом сообщения CPU (SCAN)
5	Отдельное сообщение класса "сообщения, связанные с символом"
6	Отдельное сообщение класса "сообщения, связанные с блоком"
7	Отдельное сообщение класса "сообщения о групповом состоянии"

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Разблокировка завершилась без ошибки.
7000	REQ = 0 при первом вызове: разблокировка не была активизирована.
7001	REQ = 1 при первом вызове: разблокировка была запущена.
7002	Дальнейший вызов: разблокировка уже активна.
8081	Ошибка при обращении к параметру.
8082	MODE имеет недопустимое значение.
8083	Номер сообщения находится вне допустимого диапазона значений.
8084	Нет регистрации для сообщения (сообщений), заданного с помощью MODE и, возможно, MESGN.
80C3	Сообщение (сообщения), разблокируемое в MODE и, возможно, MESGN, не может быть разблокировано в настоящее время, так как SFC9 уже разблокирует сообщения.

19.9 Поведение SFB для создания сообщений, связанных с блоками, при пуске

Теплый и холодный рестарт

При теплом и холодном рестарте SFB для создания сообщений, связанных с блоками, переводятся в состояние NO_INIT. Фактические параметры, хранящиеся в экземплярных DB, остаются неизменными.

Горячий рестарт

При горячем рестарте SFB для создания сообщений, связанных с блоками, ведут себя как функциональные блоки пользователя, способные возобновлять исполнение. Они продолжают выполнение с точки прерывания.

Сброс памяти

Сброс всегда приводит к разрыву всех соединений, так что станции, зарегистрированные для сообщений, отсутствуют. Программа пользователя удаляется. Если вы вставили флэш-карту, то существенные для исполнения части программы загружаются оттуда в CPU повторно, и CPU выполняет теплый или холодный рестарт (неявно это всегда холодный рестарт, так как после сброса памяти все данные пользователя инициализируются).

19.10 Как SFB для создания сообщений, связанных с блоками, реагируют на неисправности

Обрыв соединения

Соединения, поставленные в соответствие экземплярам SFB, контролируются на обрыв. При обрыве соединения, затронутые этим станции удаляются в CPU из внутреннего списка станций, зарегистрированных для сообщений, связанных с блоками. Все еще стоящие в очереди сообщения для этих станций удаляются.

Если другие станции все еще зарегистрированы после обрыва соединения, то они продолжают получать сообщения. SFB прекращают передачу сообщений только тогда, когда больше нет соединений ни с одной зарегистрированной станцией. Эту ситуацию отображают выходные параметры ERROR и STATUS (ERROR = 1, STATUS = 1).

Интерфейс ошибок с программой пользователя

Если при исполнении SFB для создания сообщений, связанных с блоками, появляется ошибка, то SFB переходит в состояние ERROR или ERROR_E. Одновременно выходной параметр ERROR устанавливается в 1, а в выходной параметр STATUS записывается соответствующий идентификатор ошибки. Эту информацию об ошибке вы можете оценивать в своей программе.

Примеры возможных ошибок:

- Передача невозможна из-за недостатка ресурсов
- Ошибка при обращении к контролируемому сигналам.

19.11 Введение в создание сообщений, связанных с блоками, с помощью SFC

SFC для создания сообщений, связанных с блоками

Вы можете создать сообщение, связанное с блоком, с помощью следующих SFC:

- SFC17 "ALARM_SQ"
- SFC18 "ALARM_S"

Эти SFC имеют следующие свойства:

- Сообщения, переданные функцией SFC17 "ALARM_SQ", когда состояние сигнала равно 1, могут быть квитированы на зарегистрированном устройстве отображения. Сообщения SFC18 "ALARM_S" всегда неявно квитируются.
- Сообщение генерируется не обнаруженным изменением фронта, а каждым вызовом SFC. За более подробной информацией обратитесь к разделу 19.2.
- После исполнения блока сопутствующее значение SD_1 считывается целиком и ставится в соответствие сообщению. С точки зрения непротиворечивости относительно классов более высокого приоритета непротиворечивыми являются следующие сопутствующие значения:
 - простые типы данных (бит, байт, слово и двойное слово)
 - массив байтового типа, длина которого не превышает максимальной длины, зависящей от типа CPU (см. 171, 101).

SFC19 "ALARM_SC"

С помощью SFC 19 "ALARM_SC" вы можете запросить:

- состояние квитирования последнего "поступившего сообщения" и состояние сигнала при последнем вызове SFC17 или
- состояние сигнала при последнем вызове SFC18.

Регистрация устройств отображения

SFC для создания сообщений, связанных с блоками, при вызове передают сообщение только тогда, когда по крайней мере одно устройство отображения зарегистрировано для сообщений, связанных с блоками.

Хранение сообщений

Во избежание потери сообщений при большой загрузке системы связи каждая из SFC 17 и 18 может сохранить в буфере по два сообщения. Если, однако, сообщения теряются, то вы информируетесь об этом через RET_VAL. Зарегистрированные устройства отображения информируются об этом в следующий раз, когда сообщение может быть передано.

Квитирование сообщений в случае SFC17 "ALARM_SQ"

Если вы квитировали в устройстве отображения "поступившее сообщение", то информация об этом квитировании сначала посылается в CPU, породившее это сообщение. Он затем распределяет информацию о квитировании по всем станциям, зарегистрированным для этой цели.

Блокировка и разблокировка сообщений

Зависимые от блоков сообщения, созданные с помощью SFC 17 "ALARM_SQ" или SFC 18 "ALARM_S", не могут быть заблокированы, а затем снова разблокированы.

Обновление сообщений

Вы можете использовать обновление сообщений на устройстве отображения, чтобы считывать текущее состояние сигнала и квитирования. Во время обновления все зарегистрированные станции продолжают получать сообщения. для которых они зарегистрированы.

Изменения в вашей программе

Замечание

Когда вы загружаете блок, уже находящийся в CPU, с помощью вызовов SFC17/SFC18, может оказаться так, что предыдущий блок послал сообщение со статусом "входящее", а новый блок не посылает соответствующего сообщения со статусом "уходящее". Это значит, что сообщение остается во внутренней памяти сообщений CPU. Эта ситуация может возникнуть также, когда вы удаляете блоки с помощью SFC17/SFC18. Вы можете удалить такие сообщения из внутренней памяти сообщений CPU, переведя CPU в STOP, а затем пройдя через теплый или холодный рестарт.

19.12 Создание квитируемых сообщений, связанных с блоками, с помощью SFC 17 "ALARM_SQ" и всегда квитируемых сообщений, связанных с блоками, с помощью SFC18 "ALARM_S"

Описание

SFC 17 "ALARM_SQ" и SFC 18 "ALARM_S" при каждом вызове генерируют сообщение, к которому вы можете добавить сопутствующие значения. Это сообщение передается всем зарегистрированным для станциям. SFC17 и SFC18 предоставляют в ваше распоряжение простой механизм для передачи сообщений. Вы должны обеспечить вызов SFC17 или SFC18 только тогда, когда значение запускающего сигнала SIG инвертируется по сравнению с последним вызовом. Если это не так, то это отображается в RET_VAL, и сообщения не посылаются. При самом первом вызове SFC17 или SFC18 вы должны позаботиться о том, чтобы на входе SIG была 1. В противном случае RET_VAL содержит информацию об ошибке, и сообщение не будет передано.

Замечание

Вызывайте SFC 17 и SFC 18 из FB, которому вы предварительно назначили соответствующие системные атрибуты! Более подробную информацию по назначению блокам системных атрибутов вы найдете в [/232/](#) и [/233/](#).

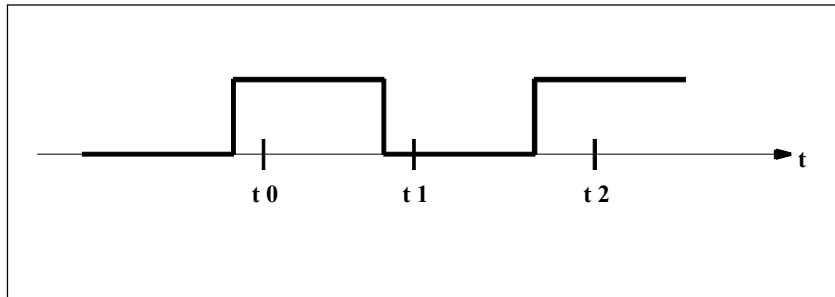
Квितिруемые сообщения

Вы можете квитиловать сообщения, передаваемые SFC17 "ALARM_SQ", когда состояние контролируемого сигнала равно 1. Вы можете опросить состояние квитирувания последнего "поступившего" сообщения и состояние сигнала при последнем вызове SFC с помощью SFC19 "ALARM_SC". Сообщения, посланные с помощью SFC18 "ALARM_S", всегда неявно квитируются. Вы можете опросить состояние сигнала при последнем вызове SFC18 с помощью SFC19 "ALARM_SC".

Временное запоминание состояний сигналов

SFC17 "ALARM_SQ" и SFC18 "ALARM_S" временно занимают память. Сюда они вводят, среди прочего, последние два состояния сигнала, включая отметку времени и сопутствующее значение. Если SFC17 или SFC18 вызывается в момент времени, когда состояния сигнала двух последних "имеющих силу" вызовов SFC еще не переданы (переполнение буфера сигналов), то текущее и последнее состояние сигнала отбрасываются, и в буферной памяти устанавливается признак переполнения. В ближайший возможный момент времени передается предпоследний сигнал вместе с признаком переполнения.

Пример:



t_0 , t_1 и t_2 – это моменты времени, когда вызываются SFC17 или SFC18. Если состояния сигнала в моменты t_0 и t_1 еще не переданы к моменту времени t_2 , то состояния сигнала, относящиеся к моментам t_1 и t_2 , отбрасываются, а для состояния сигнала, соответствующего моменту t_0 устанавливается признак переполнения.

Переполнение экземпляров

Если количество вызовов SFC17 или SFC18 больше, чем максимальное количество динамических экземпляров, то результатом этого может быть недостаток ресурсов (переполнение экземпляров). Об этом сообщается как посредством информации об ошибке в RET_VAL, так и индикацией на зарегистрированных устройствах отображения.

Максимальное количество вызовов SFC17 или SFC18 зависит от CPU. Эту информацию вы можете найти в [/70/](#) и [/101/](#).

Параметр	Описание	Тип данных	Область памяти	Характеристика
SIG	INPUT	BOOL	I, Q, M, D, L	Сигнал для запуска сообщения
ID	INPUT	WORD	I, Q, M, D, L, константа	Канал данных для сообщений: W#16#EEEE
EV_ID	INPUT	DWORD	I, Q, M, D, L, константа	Номер сообщения (0 не разрешен). При назначении номера сообщения используйте функции проектирования сообщений. Это гарантирует непротиворечивость номеров сообщений.
SD_1	IN_OUT	ANY	I, Q, M, D, T, C	Сопутствующее значение. Максимальная длина: 12 байтов Разрешены следующие типы данных: BOOL (не разрешен: битовый массив), BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5TIME, DATE_AND_TIME, COUNTER, TIMER.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибке

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
0001	<ul style="list-style-type: none"> • Сопутствующее значение длиннее, чем максимально допустимая длина, или • Доступ к памяти пользователя невозможен (например, обращение к ранее удаленному DB). Сообщение передается.
0002	Предупреждение: была использована последняя свободная память для квитирования сообщений.
8081	Заданный EV_ID находится вне области допустимых значений.
8082	Потеря сообщений, так как ваш CPU не имеет больше ресурсов для создания сообщений, связанных с блоками, с помощью SFC.
8083	Потеря сообщения, так как та же самая смена сигнала уже имеется, но еще не смогла быть передана (переполнение буфера сигналов).
8084	Сигнал SIG, запустивший сообщение, при текущем и при предыдущем вызове SFC17 или SFC18 имеет одно и то же значение.
8085	Отсутствует регистрация для указанного EV_ID
8086	Вызов SFC для заданного EV_ID уже выполняется классе более низкого приоритета.
8087	При первом вызове SFC17 или SFC18 сигнал запуска сообщения имел значение 0.
8088	Заданный EV_ID уже используется SFC другого типа, который в данный момент времени (все еще) занимает память Заданный EV_ID уже используется SFC другого типа, который в данный момент времени (все еще) занимает память.

19.13 Определение состояния квитирования последнего сообщения для наступающего события ALARM_SQ с помощью SFC19 "ALARM_SC"

Описание

С помощью SFC 19 "ALARM_SC" вы можете опросить:

- состояние квитирования последнего сообщения для наступающего события ALARM_SQ и состояние сигнала, запустившего это сообщение при последнем вызове SFC17 "ALARM_SQ".
- состояние сигнала, запустившего сообщение при последнем вызове SFC18 "ALARM_S".

Если вы назначили номера сообщений при их проектировании, то обращение к сообщению или сигналу происходит с помощью уникального номера сообщения. SFC19 "ALARM_SC" получает доступ к временно занятой памяти функции SFC17 или SFC18.

Параметр	Описание	Тип данных	Область памяти	Характеристика
EV_ID	INPUT	DWORD	I, Q, M, D, L, константа	Номер сообщения, для которого вы хотите определить состояние сигнала при последнем вызове SFC или состояние квитирования последнего "поступившего" сообщения (только для SFC17).
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Информация об ошибке
STATE	OUTPUT	BOOL	I, Q, M, D, L	Состояние сигнала, запустившего сообщение, при последнем вызове SFC
Q_STATE	OUTPUT	BOOL	I, Q, M, D, L	Если заданный параметр EV_ID относится к вызову SFC18: 1 Если заданный параметр EV_ID относится к вызову SFC 17: состояние квитирования последнего "поступившего" сообщения: 0: не квитировано; 1: квитировано

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Ошибок не было.
8081	Заданный EV_ID находится вне допустимой области.
8082	Для этого EV_ID в данный момент времени не зарезервировано место в памяти (возможная причина: соответствующий сигнал еще не имел состояния 1 или он уже опять принял состояние 0).

20 Таймеры и счетчики IEC

20.1 Генерирование импульса с помощью SFB3 "TP"

Описание

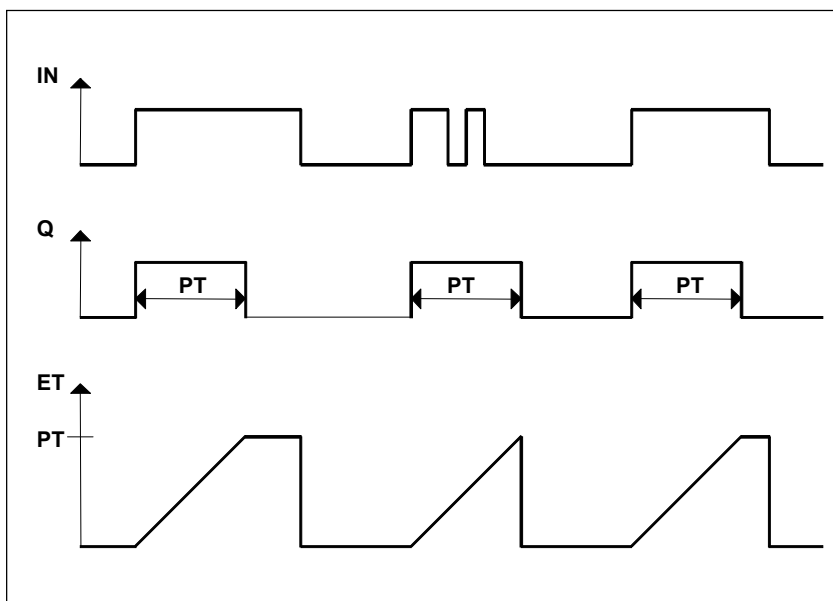
SFB3 "TP" генерирует импульс длительностью PT. Таймер работает только в режимах STARTUP и RUN.

Нарастающий фронт сигнала на входе IN запускает импульс. Выход Q остается установленным в течение промежутка времени PT, независимо от изменений во входном сигнале (иными словами, даже тогда, когда вход IN снова меняет состояние с 0 на 1 до истечения времени PT). Выход ET дает информацию о времени, в течение которого выход Q уже является установленным. Максимальное значение выхода ET равно значению входа PT. Выход ET сбрасывается, когда вход IN изменяет свое состояние на 0, но не прежде, чем истечет время PT.

SFB3 "TP" соответствует стандарту IEC 1131-3.

Операционная система сбрасывает экземпляры SFB3 "TP" во время холодного рестарта. Если вы хотите, чтобы экземпляры этого SFB были инициализированы после теплого рестарта, вы должны вызвать SFB3 с PT = 0 мс в OB100. Если экземпляры этого SFB расположены внутри другого блока, вы можете сбросить эти экземпляры, например, инициализацией блока более высокого уровня.

Временная диаграмма



Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	BOOL	I, Q, M, D, L, константа	Вход запуска
PT	INPUT	TIME	I, Q, M, D, L, константа	Длительность импульса. PT должен быть положительным. (Примечание: диапазон значений фиксирован типом данных TIME)
Q	OUTPUT	BOOL	I, Q, M, D, L	Состояние времени
ET	OUTPUT	TIME	I, Q, M, D, L	Истекшее время

20.2 Генерирование задержки включения с помощью SFB4 "TON"

Описание

SFB4 "TON" задерживает нарастающий фронт сигнала на время PT. Таймер работает только в режимах STARTUP и RUN.

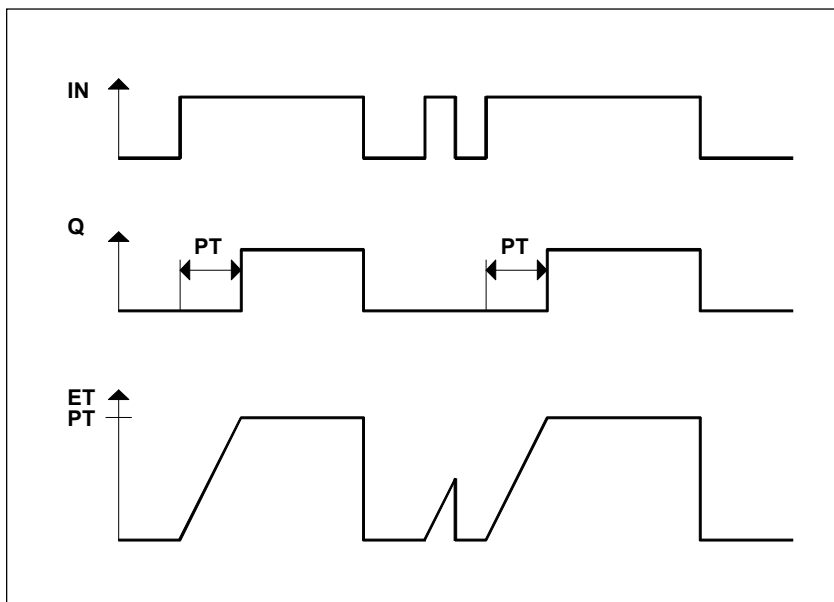
Нарастающий фронт на входе IN вызывает нарастающий фронт на выходе Q по истечении промежутка времени PT. Потом Q остается установленным до тех пор, пока вход IN не сменит состояние на 0. Если вход IN изменяет состояние на 0 до того, как истекло время PT, то выход Q остается установленным на 0.

Выход ET дает информацию о времени, которое прошло с момента последнего нарастающего фронта на входе IN. Его максимальное значение равно значению входа PT. Выход ET сбрасывается, когда вход IN изменяет состояние на 0.

SFB4 "TON" соответствует стандарту IEC 1131-3.

Операционная система сбрасывает экземпляры SFB4 "TON" во время холодного рестарта. Если вы хотите, чтобы экземпляры этого SFB были инициализированы после теплого рестарта, вы должны вызвать SFB4 с PT = 0 мс в OB100. Если экземпляры этого SFB расположены внутри другого блока, то вы можете сбросить эти экземпляры, например, инициализацией блока более высокого уровня.

Временная диаграмма



Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	BOOL	I, Q, M, D, L, константа	Вход запуска
PT	INPUT	TIME	I, Q, M, D, L, константа	Промежуток времени, на который задерживается нарастающий фронт на входе IN. PT должен быть положительным. (Примечание: диапазон значений фиксирован типом данных TIME)
Q	OUTPUT	BOOL	I, Q, M, D, L	Состояние времени
ET	OUTPUT	TIME	I, Q, M, D, L	Истекшее время

20.3 Генерирование задержки выключения с помощью SFB5 "TOF"

Описание

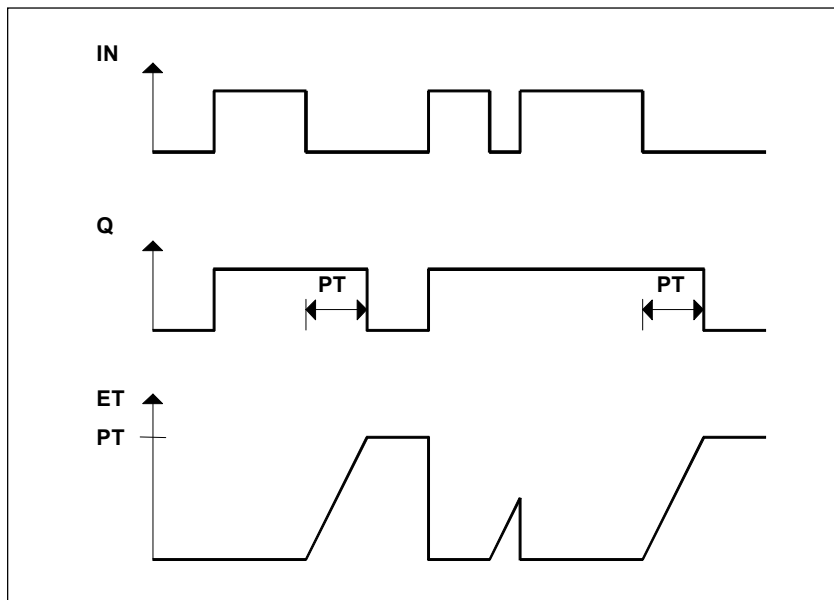
SFB5 "TOF" задерживает падающий фронт на время PT. Таймер работает только в режимах STARTUP и RUN.

Нарастающий фронт на входе IN вызывает нарастающий фронт на выходе Q. Падающий фронт на входе IN вызывает падающий фронт на выходе Q по истечении промежутка времени PT. Если вход IN снова изменяет состояние на 1 до того, как истекло время PT, то выход Q остается установленным в 1. Выход ET дает информацию о времени, истекшем после последнего падающего фронта на входе IN. Однако его максимальное значение равно значению входа PT. ET сбрасывается, когда вход IN изменяет свое состояние на 1.

SFB5 "TOF" соответствует стандарту IEC 1131-3.

Операционная система сбрасывает экземпляры SFB5 "TOF" во время холодного рестарта. Если вы хотите, чтобы экземпляры этого SFB были инициализированы после теплого рестарта, вы должны вызвать SFB5 с $PT = 0$ мс в OB100. Если экземпляры этого SFB расположены внутри другого блока, то вы можете сбросить эти экземпляры, например, инициализацией блока более высокого уровня

Временная диаграмма



Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	BOOL	I, Q, M, D, L, константа	Вход запуска
PT	INPUT	TIME	I, Q, M, D, L, константа	Промежуток времени, на который задерживается падающий фронт на входе IN. PT должен быть положительным. (Примечание: диапазон значений фиксирован типом данных TIME)
Q	OUTPUT	BOOL	I, Q, M, D, L	Состояние времени
ET	OUTPUT	TIME	I, Q, M, D, L	Истекшее время

20.4 Прямой счет с помощью SFB0 "CTU"

Описание

С помощью SFB 0 "CTU" вы можете вести прямой счет. Счетчик увеличивается на 1 при нарастающем фронте на входе CU (по сравнению с последним вызовом SFB). Если значение счетчика достигает верхней границы 32 767, то оно больше не увеличивается. После этого каждый следующий нарастающий фронт на входе CU остается без последствий.

Уровень сигнала 1 на входе R вызывает сброс счетчика в 0, независимо от того, какое значение имеет место на входе CU.

На выходе Q отображается, является ли текущее значение счетчика большим или равным предварительно заданному значению PV.

SFB0 "CTU" соответствует стандарту IEC 1131-3.

Операционная система сбрасывает экземпляры SFB0 "CTU" во время холодного рестарта. Если вы хотите, чтобы экземпляры этого SFB были инициализированы после теплого рестарта, вы должны вызвать SFB0 с $R = 1$ в OB100. Если экземпляры этого SFB расположены внутри другого блока, то вы можете сбросить эти экземпляры, например, инициализацией блока более высокого уровня

Параметр	Описание	Тип данных	Область памяти	Характеристика
CU	INPUT	BOOL	I, Q, M, D, L, константа	Вход счетчика.
R	INPUT	BOOL	I, Q, M, D, L, константа	Вход сброса R имеет преимущество перед CU.
PV	INPUT	INT	I, Q, M, D, L, константа	Предварительно заданное значение. О воздействии PV см. параметр Q.
Q	OUTPUT	BOOL	I, Q, M, D, L	Состояние счетчика: Q имеет следующее значение <ul style="list-style-type: none"> • 1, если $CV \geq PV$ • 0 в противном случае
CV	OUTPUT	INT	I, Q, M, D, L	Текущее значение счетчика. (Возможные значения: от 0 до 32 767).

20.5 Обратный счет с помощью SFB1 "CTD"

Описание

С помощью SFB 1 "CTD" вы можете выполнять обратный счет. Счетчик уменьшается на 1 при нарастающем фронте на входе CD (по сравнению с последним вызовом SFB). Если значение счетчика достигает нижней границы -32 767, то оно больше не уменьшается. После этого каждый следующий нарастающий фронт на входе CD остается без последствий.

Уровень сигнала 1 на входе LOAD загружает в счетчик предварительно установленное значение независимо от значения на входе CD.

На выходе Q отображается, является ли текущее счетное значение меньшим или равным нулю.

SFB1 "CTD" соответствует стандарту IEC 1131-3.

Операционная система сбрасывает экземпляры SFB1 "CTD" во время холодного рестарта. Если вы хотите, чтобы экземпляры этого SFB были инициализированы после теплого рестарта, вы должны вызвать SFB1 с $LOAD = 1$ и PV = требуемому начальному значению для CV в OB100. Если экземпляры этого SFB расположены внутри другого блока, то вы можете сбросить эти экземпляры, например, инициализацией блока более высокого уровня.

Параметр	Описание	Тип данных	Область памяти	Характеристика
CD	INPUT	BOOL	I, Q, M, D, L, константа	Счетный вход
LOAD	INPUT	BOOL	I, Q, M, D, L, константа	Вход загрузки. LOAD имеет преимущество над CD.
PV	INPUT	INT	I, Q, M, D, L, константа	Предварительно заданное значение. Счетчик устанавливается на PV, когда уровень сигнала на входе LOAD равен 1.
Q	OUTPUT	BOOL	I, Q, M, D, L	Состояние счетчика: Q имеет значение: <ul style="list-style-type: none"> 1, если $CV \leq 0$ 0 в противном случае
CV	OUTPUT	INT	I, Q, M, D, L	Текущее значение счетчика (возможные значения: от -32 768 до 32 767)

20.6 Прямой и обратный счет с помощью SFB2 "CTUD"

Описание

С помощью SFB 2 "CTUD" вы можете вести прямой и обратный счет. Значение счетчика при нарастающем фронте изменяется по сравнению с последним вызовом SFB следующим образом:

- на CU он увеличивается на 1
- на CD он уменьшается на 1.

Если счетное значение достигает границ, то счетчик реагирует следующим образом:

- нижняя граница -32768, он больше не уменьшается
- верхняя граница 32767, он больше не увеличивается.

Если в одном цикле имеет место нарастающий фронт как на входе CU, так и на входе CD, то счетчик сохраняет свое текущее значение. Эта реакция не соответствует стандарту IEC 1131-3. В соответствии со стандартом вход CU имеет преимущество, если оба сигнала активны одновременно. Это изменение предложено IEC (International Electrotechnical Commission, Международной электротехнической комиссии).

Уровень сигнала 1 на входе LOAD предварительно устанавливает счетчик на значение PV независимо от значений на входах CU и CD.

Уровень сигнала 1 на входе R сбрасывает счетчик на значение 0 независимо от значений на входах CU, CD и LOAD. Выход QU показывает, является ли текущее счетное значение большим или равным предварительно заданному значению PV; выход QD показывает, является ли это значение меньшим или равным 0.

Операционная система сбрасывает SFB2 "CTUD" во время холодного рестарта. Если вы хотите, чтобы SFB2 "CTUD" инициализировался после теплого рестарта, вы должны вызвать SFB2 в OB100 следующим образом:

- с R = 1 при использовании блока для прямого счета
- с R = 0 и LOAD = 1 и PV = требуемому начальному значению для CV при использовании блока для обратного счета

Если экземпляры этого SFB расположены внутри другого блока, то вы можете сбросить эти экземпляры, например, инициализацией блока более высокого уровня.

Параметр	Описание	Тип данных	Область памяти	Характеристика
CU	INPUT	BOOL	I, Q, M, D, L, константа	Вход прямого счета.
CD	INPUT	BOOL	I, Q, M, D, L, константа	Вход обратного счета
R	INPUT	BOOL	I, Q, M, D, L, константа	Вход сброса. R имеет преимущество над LOAD.
LOAD	INPUT	BOOL	I, Q, M, D, L, константа	Вход загрузки. LOAD имеет преимущество над CU и CD.
PV	INPUT	INT	I, Q, M, D, L, константа	Предварительно заданное значение. Счетчик устанавливается на предварительно заданное значение PV, когда уровень сигнала на входе LOAD равен 1.
QU	OUTPUT	BOOL	I, Q, M, D, L	Состояние прямого счетчика: QU имеет следующее значение <ul style="list-style-type: none"> • 1, если $CV \geq PV$ • 0 в противном случае
QD	OUTPUT	BOOL	I, Q, M, D, L	Состояние обратного счетчика: QD имеет следующее значение <ul style="list-style-type: none"> • 1, если $CV \leq 0$ • 0 в противном случае
CV	OUTPUT	INT	I, Q, M, D, L	Текущее значение счетчика (возможные значения: от -32 768 до 32 767)

21 Функции IEC

21.1 Обзор

Имеющиеся в распоряжении функции

Вы можете копировать следующие функции Международной электротехнической комиссии (International Electrotechnical Commission, IEC) из библиотеки STDLIBS\IEC пакета STEP 7 в свою папку с программами.

Функция	Имя		Семейство блоков IEC
Объединение DATE и TIME_OF_DAY в DT	D_TOD_DT	FC 3	Convert [Преобразование]
Извлечение DATE из DT	DT_DATE	FC 6	Convert
Извлечение дня недели из DT	DT_DAY	FC 7	Convert
Извлечение TIME_OF_DAY из DT	DT_TOD	FC 8	Convert
Преобразование типа данных S5TIME в TIME	S5TI_TIM	FC 33	Convert
Преобразование типа данных TIME в S5TIME	TIM_S5TI	FC 40	Convert
Преобразование типа данных INT в STRING	I_STRNG	FC 16	Convert
Преобразование типа данных DINT в STRING	DI_STRNG	FC 5	Convert
Преобразование типа данных REAL в STRING	R_STRNG	FC 30	Convert
Преобразование типа данных STRING в INT	STRNG_I	FC 38	Convert
Преобразование типа данных STRING в DINT	STRNG_DI	FC 37	Convert
Преобразование типа данных STRING в REAL	STRNG_R	FC 39	Convert
Сравнение DT на равно	EQ_DT	FC 9	DT
Сравнение DT на больше или равно	GE_DT	FC 12	DT
Сравнение DT на больше	GT_DT	FC 14	DT
Сравнение DT на меньше или равно	LE_DT	FC 18	DT
Сравнение DT на меньше	LT_DT	FC 23	DT
Сравнение DT на не равно	NE_DT	FC 28	DT
Сравнение STRING на равно	EQ_STRNG	FC 10	String [строка]
Сравнение STRING на больше или равно	GE_STRNG	FC 13	String
Сравнение STRING на больше	GT_STRNG	FC 15	String
Сравнение STRING на меньше или равно	LE_STRNG	FC 19	String
Сравнение STRING на меньше	LT_STRNG	FC 24	String
Сравнение STRING на не равно	NE_STRNG	FC 29	String
Длина переменной STRING	LEN	FC 21	String
Левая часть переменной STRING	LEFT	FC 20	String
Правая часть переменной STRING	RIGHT	FC 32	String
Средняя часть переменной STRING	MID	FC 26	String
Объединение двух переменных STRING	CONCAT	FC 2	String
Вставка в переменной STRING	INSERT	FC 17	String
Удаление в переменной STRING	DELETE	FC 4	String
Замена в переменной STRING	REPLACE	FC 31	String
Поиск в переменной STRING	FIND	FC 11	String
Прибавление промежутка времени к моменту времени	AD_DT_TM	FC 1	Floating-Point Math [математика с плавающей точкой]
Вычитание промежутка времени из момента времени	SB_DT_TM	FC 35	Floating-Point Math
Вычитание двух значений времени	SB_DT_DT	FC 34	Floating-Point Math
Ограничение	LIMIT	FC 22	Floating-Point Math
Выбор максимума	MAX	FC 25	Floating-Point Math
Выбор минимума	MIN	FC 27	Floating-Point Math
Двоичный выбор	SEL	FC 36	Floating-Point Math

21.2 Технические данные функций IEC

Требования к памяти

Следующая таблица показывает, сколько требуется рабочей и загрузочной памяти для каждой функции IEC, а также количество байтов локальных данных, требуемых для каждой функции IEC.

Имя		Байты рабочей памяти	Байты загрузочной памяти	Байты локальных данных
D_TOD_DT	FC3	634	810	12
DT_DATE	FC6	340	466	10
DT_DAY	FC7	346	472	10
DT_TOD	FC8	114	210	6
S5TI_TIM	FC33	94	208	2
TIM_S5TI	FC40	104	208	6
I_STRNG	FC16	226	340	10
DI_STRNG	FC5	314	440	18
R_STRNG	FC30	528	684	28
STRNG_I	FC38	292	420	12
STRNG_DI	FC37	310	442	12
STRNG_R	FC39	828	1038	30
EQ_DT	FC9	96	194	2
GE_DT	FC12	174	288	4
GT_DT	FC14	192	310	4
LE_DT	FC18	168	280	4
LT_DT	FC23	192	310	4
NE_DT	FC28	96	194	2
EQ_STRNG	FC10	114	220	4
GE_STRNG	FC13	162	282	8
GT_STRNG	FC15	158	278	8
LE_STRNG	FC19	162	282	8
LT_STRNG	FC24	158	278	8
NE_STRNG	FC29	150	266	8
LEN	FC21	38	132	2
LEFT	FC20	200	320	8
RIGHT	FC32	230	350	8
MID	FC26	264	390	8
CONCAT	FC2	320	452	14
INSERT	FC17	488	644	20
DELETE	FC4	376	512	8
REPLACE	FC31	562	726	20
FIND	FC11	236	360	14
AD_DT_TM	FC1	1350	1590	22
SB_DT_TM	FC35	1356	1596	22
SB_DT_DT	FC34	992	1178	30
LIMIT	FC22	426	600	12
MAX	FC25	374	532	8
MIN	FC27	374	532	8
SEL	FC36	374	560	8

21.3 Дата и время как составные типы данных

Фактические параметры для DATE_AND_TIME

Тип данных DATE_AND_TIME относится к категории составных типов данных, наряду с ARRAY, STRING и STRUCT. Допустимыми областями памяти для составных типов данных являются блоки данных (DB) и области локальных данных (L-стек).

Так как DATE_AND_TIME – это составной тип данных, то когда вы используете его в качестве формального параметра в команде, вы должны определять фактический параметр в одной из следующих форм:

- как локальный символ блока из таблицы описания переменных конкретного блока
- как символическое имя для блока данных, например, "DB_sys_info.System_Time", составленное из следующих двух частей:
 - имя, определенное в таблице символов для номера блока данных (например, "DB_sys_info" для DB5)
 - имя, определенное внутри блока данных для элемента DATE_AND_TIME (например, "System_Time" для переменной типа данных DATE_AND_TIME, которая содержится в DB5)

Вы не можете использовать константы в качестве фактических параметров для формальных параметров, принадлежащих к составным типам данных, включая DATE_AND_TIME. Вы также не можете передавать абсолютные адреса в качестве фактических параметров для DATE_AND_TIME.

21.4 Функции даты и времени суток: FC3, FC6, FC7, FC8, FC33, FC40, FC1, FC35, FC34

Описание FC3

Функция FC 3 объединяет форматы данных DATE и TIME_OF_DAY (TOD) и преобразует эти форматы в формат данных DATE_AND_TIME (DT). Входное значение IN1 должно лежать между границами DATE#1990-01-01 и DATE#2089-12-31 (это значение не проверяется). Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN1	INPUT	DATE	I, Q, M, D, L, константа	Входная переменная в формате DATE
IN2	INPUT	TIME_OF_DAY	I, Q, M, D, L, константа	Входная переменная в формате TOD
RET_VAL	OUTPUT	DATE_AND_TIME	D, L	Возвращаемое значение в формате DT

Возвращаемому значению можно ставить в соответствие только символически определенную переменную.

Описание FC6

Функция FC6 извлекает формат данных DATE из формата данных DATE_AND_TIME. Значение DATE должно находиться между граничными значениями DATE#1990-1-1 и DATE#2089-12-31. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
RET_VAL	OUTPUT	DATE	I, Q, M, D, L	Возвращаемое значение в формате DATE

Входному значению можно ставить в соответствие только символически определенную переменную.

Описание FC7

Функция IEC FC7 извлекает день недели из формата DATE_AND_TIME. недели представляется в формате данных INT (от 1 до 7):

- 1 Воскресенье
- 2 Понедельник
- 3 Вторник
- 4 Среда
- 5 Четверг
- 6 Пятница
- 7 Суббота

Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Возвращаемое значение в формате INT

Входному значению можно ставить в соответствие только символически определенную переменную.

Описание FC8

Функция IEC FC8 извлекает формат данных TIME_OF_DAY из формата DATE_AND_TIME. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
RET_VAL	OUTPUT	TIME_OF_DAY	I, Q, M, D, L	Возвращаемое значение в формате TOD

Входному значению можно ставить в соответствие только символически определенную переменную.

Описание FC33

Функция IEC FC33 преобразует формат данных S5TIME в формат TIME. Если результат преобразования лежит вне диапазона TIME, то результат ограничивается соответствующим значением и бит двоичного результата (BR) слова состояний устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	S5TIME	I, Q, M, D, L, константа	Входная переменная в формате S5TIME
RET_VAL	OUTPUT	TIME	I, Q, M, D, L	Возвращаемое значение в формате TIME

Описание FC40

Функция FC 40 преобразует формат данных TIME в формат S5TIME. При преобразовании происходит округление. Если входной параметр больше, чем допускает формат S5TIME (больше, чем TIME#02:46:30.000), то выводится результат S5TIME#999.3 и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	TIME	I, Q, M, D, L, константа	Входная переменная в формате TIME
RET_VAL	OUTPUT	S5TIME	I, Q, M, D, L	Возвращаемое значение в формате S5TIME

Описание FC1

Функция FC 1 прибавляет промежуток времени (формат TIME) к моменту времени (формат DT) и поставляет в качестве результата новый момент времени (формат DT). Момент времени (параметр T) должен лежать в диапазоне от DT#1990-01-01-00:00:00.000 до DT#2089-12-31-23:59:59.999. Функция не выполняет входной проверки. Если результат сложения не лежит внутри допустимого диапазона, то результат ограничивается соответствующим значением и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
T	INPUT	DT	I, Q, M, D, L	Момент времени в формате DT
D	INPUT	TIME	I, Q, M, D, L, константа	Промежуток времени в формате TIME
RET_VAL	OUTPUT	DT	I, Q, M, D, L	Сумма в формате DT

Для входного параметра T и выходного параметра можно ставить в соответствие только символически определенную переменную.

Описание FC35

Функция FC 35 вычитает промежуток времени (формат TIME) из момента времени (формат DT) и поставляет в качестве результата новый момент времени (формат DT). Момент времени (параметр T) должен лежать в диапазоне от DT#1990-01-01-00:00:00.000 до DT#2089-12-31-23:59:59.999. Функция не выполняет входной проверки. Если результат вычитания не лежит внутри допустимого диапазона, то результат ограничивается соответствующим значением и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
T	INPUT	DT	I, Q, M, D, L	Момент времени в формате DT
D	INPUT	TIME	I, Q, M, D, L, константа	Промежуток времени в формате TIME
RET_VAL	OUTPUT	DT	I, Q, M, D, L	Разность в формате DT

Для входного параметра T и выходного параметра можно ставить в соответствие только символически определенную переменную.

Описание FC34

Функция FC34 вычитает два значения времени (формат DT) и поставляет в качестве результата интервал времени (формат TIME). Значения времени должны лежать в диапазоне от DT#1990-01-01-00:00:00.000 до DT#2089-12-31-23:59:59.999. Функция не выполняет входной проверки. Если первое время (параметр T1) больше (младше), чем второе (параметр T2), то результат положителен; если первое время меньше (старше), чем второе, то результат отрицателен. Если результат вычитания лежит вне диапазона TIME, то результат ограничивается соответствующим значением и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
DT1	INPUT	DT	I, Q, M, D, L	Первый момент времени в формате DT
DT2	INPUT	DT	I, Q, M, D, L	Второй момент времени в формате DT
RET_VAL	OUTPUT	TIME	I, Q, M, D, L	Разность в формате TIME

Входным параметрам вы можете ставить в соответствие только символически определенную переменную.

21.5 Сравнение переменных типа DATE_AND_TIME: FC9, FC12, FC14, FC18, FC23, FC28

Описание FC9

Функция FC 9 сравнивает содержимое двух переменных в формате DATE_AND_TIME, чтобы выяснить, равны ли они, и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала "1", если момент времени в параметре DT1 равен моменту времени в параметре DT2. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
DT1	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
DT2	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC12

Функция FC 12 сравнивает содержимое двух переменных в формате DATE_AND_TIME на "больше или равно" и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала "1", если момент времени в параметре DT1 больше (младше), чем момент времени в параметре DT2, или оба момента времени равны. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
DT1	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
DT2	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC14

Функция FC14 сравнивает содержимое двух переменных в формате DATE_AND_TIME на "больше" и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала "1", если момент времени в параметре DT1 больше (младше), чем момент времени в параметре DT2. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
DT1	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
DT2	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC18

Функция FC 18 сравнивает содержимое двух переменных в формате DATE_AND_TIME на “меньше или равно” и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала “1”, если момент времени в параметре DT1 меньше (старше), чем момент времени в параметре DT2, или оба момента времени равны. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
DT1	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
DT2	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC23

Функция FC 23 сравнивает содержимое двух переменных в формате DATE_AND_TIME на “меньше” и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала “1”, если момент времени в параметре DT1 меньше (старше), чем момент времени в параметре DT2. Функция не сообщает об ошибке.

Параметр	Описание	Тип данных	Область памяти	Характеристика
DT1	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
DT2	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC28

Функция FC 28 сравнивает значения двух переменных в формате DATE_AND_TIME на “не равно” и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала “1”, если момент времени в параметре DT1 не равен моменту времени в параметре DT2. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
DT1	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
DT2	INPUT	DATE_AND_TIME	D, L	Входная переменная в формате DT
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

21.6 Сравнение переменных типа STRING: FC10, FC13, FC15, FC19, FC24, FC29

Замечание

Если вы назначаете фактический параметр, определенный в области локальных данных вызывающего блока, формальному параметру типа данных STRING при вызове одной из следующих FC, помните о следующем:

Перед вызовом FC вы должны записать определенную длину строки в управляющий байт "maximum length of the string [максимальная длина строки]" во временных переменных типа данных STRING.

Описание FC10

Функция FC 10 сравнивает значения двух переменных в формате STRING на "равно" и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала "1", если строка символов в параметре S1 равна строке символов в параметре S2. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
S1	INPUT	STRING	D, L	Входная переменная в формате STRING
S2	INPUT	STRING	D, L	Входная переменная в формате STRING
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC13

Функция FC 13 сравнивает содержимое двух переменных в формате STRING на "больше или равно" и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала "1", если строка символов в параметре S1 больше, чем строка символов в параметре S2, или равна ей. Начиная слева, символы сравниваются по их ASCII-кодам (например, 'a' больше, чем 'A'). Первый отличный символ определяет результат сравнения. Если левая часть более длинной строки символов идентична более короткой строке символов, то более длинная строка символов считается большей. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
S1	INPUT	STRING	D, L	Входная переменная в формате STRING
S2	INPUT	STRING	D, L	Входная переменная в формате STRING
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC15

Функция FC 15 сравнивает содержимое двух переменных в формате STRING на “больше” и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала “1”, если строка символов в параметре S1 больше, чем строка символов в параметре S2. Начиная слева, символы сравниваются по их ASCII-кодам (например, ‘a’ больше, чем ‘A’). Первый отличный символ определяет результат сравнения. Если левая часть более длинной строки символов идентична более короткой строке символов, то более длинная строка символов считается большей. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
S1	INPUT	STRING	D, L	Входная переменная в формате STRING
S2	INPUT	STRING	D, L	Входная переменная в формате STRING
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC19

Функция FC 19 сравнивает содержимое двух переменных в формате STRING на “меньше или равно” и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала “1”, если строка символов в параметре S1 меньше, чем строка символов в параметре S2, или равна ей. Начиная слева, символы сравниваются по их ASCII-кодам (например, ‘A’ меньше, чем ‘a’). Первый отличный символ определяет результат сравнения. Если левая часть более длинной строки символов идентична более короткой строке символов, то более короткая строка символов считается меньшей. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
S1	INPUT	STRING	D, L	Входная переменная в формате STRING
S2	INPUT	STRING	D, L	Входная переменная в формате STRING
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC24

Функция FC 24 сравнивает содержимое двух переменных в формате STRING на “меньше” и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала “1”, если строка символов в параметре S1 меньше, чем строка символов в параметре S2. Начиная слева, символы сравниваются по их ASCII-кодам (например, ‘A’ меньше, чем ‘a’). Первый отличный символ определяет результат сравнения. Функция FC 24 сравнивает содержимое двух переменных в формате STRING на “меньше” и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала “1”, если строка символов в параметре S1 меньше, чем строка символов в параметре S2. Начиная слева, символы сравниваются по их ASCII-кодам (например, ‘A’ меньше, чем ‘a’). Первый отличный символ определяет результат сравнения. Если левая часть более длинной строки символов идентична более короткой строке символов, то более короткая строка символов считается меньшей. Функция не сообщает об ошибке.

Параметр	Описание	Тип данных	Область памяти	Характеристика
S1	INPUT	STRING	D, L	Входная переменная в формате STRING
S2	INPUT	STRING	D, L	Входная переменная в формате STRING
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC29

Функция FC 29 сравнивает содержимое двух переменных в формате STRING на “не равно” и выводит результат сравнения в качестве возвращаемого значения. Возвращаемое значение имеет состояние сигнала “1”, если строка символов в параметре S1 не равна строке символов в параметре S2. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
S1	INPUT	STRING	D, L	Входная переменная в формате STRING
S2	INPUT	STRING	D, L	Входная переменная в формате STRING
RET_VAL	OUTPUT	BOOL	I, Q, M, D, L	Результат сравнения

Входным параметрам можно ставить в соответствие только символически определенную переменную.

21.7 Редактирование переменных типа STRING: FC21, FC20, FC32, FC26, FC2, FC17, FC4, FC31, FC11

Замечание

Если вы назначаете фактический параметр, определенный в области локальных данных вызывающего блока, формальному параметру типа данных STRING при вызове одной из следующих FC, помните о следующем:

Перед вызовом FC вы должны записать определенную длину строки в управляющий байт "maximum length of the string [максимальная длина строки]" во временных переменных типа данных STRING.

Описание FC21

Переменная типа STRING содержит две длины: максимальную длину (она задается при определении переменных в квадратных скобках) и текущую длину (это количество символов, действующих в данный момент времени). Текущая длина должна быть меньше, чем максимальная длина, или равна ей. Количество байтов, занимаемых строкой символов, больше максимальной длины на 2. Переменная типа STRING содержит две длины: максимальную длину (она задается при определении переменных в квадратных скобках) и текущую длину (это количество символов, действующих в данный момент времени). Текущая длина должна быть меньше, чем максимальная длина, или равна ей. Количество байтов, занимаемых строкой символов, больше максимальной длины на 2.

Функция FC 21 выводит текущую длину строки (количество действующих символов) в качестве возвращаемого значения. Пустая строка (") имеет длину нуль. Максимальная длина равна 254. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
S	INPUT	STRING	D, L	Переменная типа STRING
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Количество действующих символов

Входному параметру можно ставить в соответствие только символически определенную переменную.

Описание FC20

Функция FC 20 предоставляет в распоряжение первые L символов строки. Если L больше, чем текущая длина переменной типа STRING, то возвращается входное значение. При L = 0 и при пустой строке в качестве входного значения возвращается пустая строка. Если число L отрицательно, то выводится пустая строка и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	STRING	D, L	Входная переменная в формате STRING
L	INPUT	INT	I, Q, M, D, L, константа	Длина левой части строки
RET_VAL	OUTPUT	STRING	D, L	Выходная переменная в формате STRING

Параметру IN и возвращаемому значению можно ставить в соответствие только символически определенную переменную.

Описание FC32

Функция FC 32 предоставляет в распоряжение последние L символов строки (где L означает число). Если L больше, чем текущая длина переменной STRING, то возвращается входное значение. При L = 0 и при пустой строке в качестве входного значения возвращается пустая строка. Если число L отрицательно, то выводится пустая строка и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	STRING	D, L	Входная переменная в формате STRING
L	INPUT	INT	I, Q, M, D, L, константа	Длина правой части строки
RET_VAL	OUTPUT	STRING	D, L	Выходная переменная в формате STRING

Параметру IN и возвращаемому значению можно ставить в соответствие только символически определенную переменную.

Описание FC26

Функция FC 26 предоставляет в распоряжение среднюю часть строки символов (L символов, начиная с P-го символа включительно). Если сумма L и (P-1) превосходит текущую длину переменной типа STRING, то возвращается строка символов, начиная с P-го символа входной строки до ее конца. Во всех остальных случаях (P находится вне текущей длины, P и/или L равны нулю или отрицательны) выводится пустая строка и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	STRING	D, L	Входная переменная в формате STRING
L	INPUT	INT	I, Q, M, D, L, константа	Длина средней части строки
P	INPUT	INT	I, Q, M, D, L, константа	Позиция первого символа
RET_VAL	OUTPUT	STRING	D, L	Выходная переменная в формате STRING

Параметру IN и возвращаемому значению можно ставить в соответствие только символически определенную переменную.

Описание FC2

Функция FC 2 объединяет две переменные типа STRING в одну строку символов. Если результирующая строка символов длиннее, чем переменная, заданная в выходном параметре, то результирующая строка ограничивается максимально установленной длиной и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN1	INPUT	STRING	D, L	Входная переменная в формате STRING
IN2	INPUT	STRING	D, L	Входная переменная в формате STRING
RET_VAL	OUTPUT	STRING	D, L	Объединенная строка

Параметрам можно ставить в соответствие только символически определенную переменную.

Описание FC17

Функция FC 17 вставляет строку символов из параметра IN2 в строку символов в параметре IN1 после P-го символа. Если P равно нулю, то вторая строка символов вставляется перед первой строкой символов. Если P больше, чем текущая длина первой строки символов, то вторая строка символов присоединяется к первой. Если P отрицательно, то выводится пустая строка и бит двоичного результата (BR) слова состояния устанавливается в "0". Бит двоичного результата также устанавливается в "0", если 0" результирующая строка длиннее, чем переменная, заданная в выходном параметре; в этом случае результирующая строка ограничивается максимально установленной длиной.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN1	INPUT	STRING	D, L	Переменная типа STRING, в которую происходит вставка
IN2	INPUT	STRING	D, L	Вставляемая переменная типа STRING
P	INPUT	INT	I, Q, M, D, L, константа	Позиция вставки
RET_VAL	OUTPUT	STRING	D, L	Результирующая строка

Входным параметрам IN1 и IN2 и выходному параметру можно ставить в соответствие только символически определенную переменную.

Описание FC4

Функция FC 4 удаляет L символов в строке, начиная с P-го символа (включительно). Если L и/или P равны нулю или P больше, чем текущая длина входной строки, то возвращается входная строка. Если сумма L и P больше, чем входная строка символов, то строка символов удаляется до конца. Если L и/или P отрицательны, то выводится пустая строка и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN	INPUT	STRING	D, L	Переменная типа STRING, в которой происходит удаление
L	INPUT	INT	I, Q, M, D, L, константа	Количество удаляемых символов
P	INPUT	INT	I, Q, M, D, L, константа	Позиция 1-го удаляемого символа
RET_VAL	OUTPUT	STRING	D, L	Результирующая строка

Входному параметру IN и выходному параметру можно ставить в соответствие только символически определенную переменную.

Описание FC31

Функция FC 31 заменяет L символов первой строки (IN1), начиная с P-го символа (включительно) второй строкой символов (IN2). Если L равно нулю, то возвращается первая строка символов. Если P равно нулю или единице, то замена происходит, начиная с 1-го символа (включительно). Если P лежит вне первой строки символов, то вторая строка присоединяется к первой строке. Если L и/или P отрицательны, то возвращается пустая строка и бит двоичного результата (BR) слова состояния устанавливается в "0". Бит двоичного результата также устанавливается в "0", результирующая строка длиннее, чем переменная, заданная в выходном параметре; в этом случае результирующая строка ограничивается максимально установленной длиной.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN1	INPUT	STRING	D, L	Переменная типа STRING, в которой происходит замена
IN2	INPUT	STRING	D, L	Заменяющая переменная типа STRING
L	INPUT	INT	I, Q, M, D, L, константа	Количество заменяемых символов
P	INPUT	INT	I, Q, M, D, L, константа	Позиция 1-го заменяемого символа
RET_VAL	OUTPUT	STRING	D, L	Результирующая строка

Входным параметрам IN1 и IN2 и выходному параметру можно ставить в соответствие только символически определенную переменную.

Описание FC11

Функция FC 11 сообщает позицию второй строки символов (IN2) внутри первой строки символов (IN1). Поиск начинается слева; сообщается о первом появлении строки символов. Если вторая строка символов не содержится в первой, то возвращается нуль. Функция не сообщает об ошибках.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN1	INPUT	STRING	D, L	Переменная типа STRING, в которой происходит поиск
IN2	INPUT	STRING	D, L	Искомая переменная типа STRING
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Позиция найденной строки символов

Входным параметрам IN1 и IN2 можно ставить в соответствие только символически определенную переменную.

21.8 Преобразование форматов типов данных: FC16, FC5, FC30, FC38, FC37, FC39

Замечание

Если вы назначаете фактический параметр, определенный в области локальных данных вызывающего блока, формальному параметру типа данных STRING при вызове одной из следующих FC, помните о следующем:

Перед вызовом FC вы должны записать определенную длину строки в управляющий байт "maximum length of the string [максимальная длина строки]" во временных переменных типа данных STRING.

Описание FC16

Функция FC 16 преобразует переменную в формате INT в строку символов. Строка символов представляется с предшествующим знаком. Если переменная, заданная для возвращаемого параметра, слишком коротка, то преобразование не происходит и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
I	INPUT	INT	I, Q, M, D, L, константа	Входное значение
RET_VAL	OUTPUT	STRING	D, L	Результирующая строка

Для выходного параметра можно ставить в соответствие только символически определенную переменную.

Описание FC5

Функция FC 5 преобразует переменную в формате DINT в строку символов. Строка символов представляется с предшествующим знаком. Если переменная, заданная для возвращаемого параметра, слишком коротка, то преобразование не происходит и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
I	INPUT	INT	I, Q, M, D, L, константа	Входное значение
RET_VAL	OUTPUT	STRING	D, L	Результирующая строка

Для выходного параметра можно ставить в соответствие только символически определенную переменную.

Описание FC30

Функция FC 30 преобразует переменную в формате REAL в строку символов. Строка символов представляется 14 разрядами:

±v.nnnnnnnE±xx ± знак
 v 1 разряд перед десятичной точкой
 n 7 разрядов после десятичной точки
 x 2 разряда показателя степени

Если заданная для возвращаемого параметра переменная слишком коротка или если параметру IN задано недопустимое число с плавающей точкой, то преобразование не происходит и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
I	INPUT	INT	I, Q, M, D, L, константа	Входное значение
RET_VAL	OUTPUT	STRING	D, L	Результирующая строка

Выходному параметру можно ставить в соответствие только символически определенную переменную.

Описание FC38

Функция FC 38 преобразует строку символов в переменную в формате INT. Первый символ в строке символов может быть знаком или цифрой, следующие после этого символы должны состоять из цифр. Если длина строки символов равна нулю или больше 6 или в строке символов находятся неразрешенные символы, то преобразование не происходит и бит двоичного результата (BR) слова состояния устанавливается в "0". Если результат преобразования лежит вне области INT, то результат ограничивается соответствующим значением и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
S	INPUT	STRING	D, L	Входная строка
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Результат

Входному параметру можно ставить в соответствие только символически определенную переменную.

Описание FC37

Функция FC 37 преобразует строку символов в переменную в формате DINT. Первый символ в строке символов может быть знаком или цифрой, следующие после этого символы должны состоять из цифр. Если длина строки символов равна нулю или больше 11 или в строке символов находятся неразрешенные символы, то преобразование не происходит и бит двоичного результата (BR) слова состояния устанавливается в "0". Если результат преобразования лежит вне области DINT, то результат ограничивается соответствующим значением и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
S	INPUT	STRING	D, L	Входная строка
RET_VAL	OUTPUT	DINT	I, Q, M, D, L	Результат

Входному параметру можно ставить в соответствие только символически определенную переменную.

Описание FC39

Функция FC 39 преобразует строку символов в переменную в формате REAL. Строка символов должна быть представлена в следующем формате:

±v.nnnnnnnE±xx ± знак
 v 1 разряд перед десятичной точкой
 n 7 разрядов после десятичной точки
 x 2 разряда показателя степени

Если длина строки символов меньше 14 или если она построена не так, как показано выше, то преобразование не происходит и бит двоичного результата (BR) слова состояния устанавливается в "0". Если результат преобразования лежит вне диапазона REAL, то результат ограничивается соответствующим значением и бит двоичного результата (BR) слова состояния устанавливается в "0".

Параметр	Описание	Тип данных	Область памяти	Характеристика
S	INPUT	STRING	D, L	Входная строка
RET_VAL	OUTPUT	REAL	I, Q, M, D, L	Результат

Входному параметру можно ставить в соответствие только символически определенную переменную.

21.9 Редактирование числовых значений: FC22, FC25, FC27

Описание FC22

Функция FC 22 ограничивает числовое значение переменной выбираемыми граничными значениями. В качестве входных значений допускаются переменные типов INT, DINT и REAL. Все переменные с назначенными параметрами должны иметь тот же тип данных. Нижнее граничное значение (параметр MN) не должен быть больше, чем верхнее граничное значение (параметр MX).

Выходное значение остается неизменным и бит двоичного результата (BR) слова состояния устанавливается в "0", если выполняется любое из следующих условий:

- переменная с назначенными параметрами имеет недопустимый тип данных
- не все переменные с назначенными параметрами имеют один и тот же тип данных
- нижнее граничное значение больше, чем верхнее граничное значение
- переменная типа REAL не представляет допустимое число с плавающей точкой.

Параметр	Описание	Тип данных	Область памяти	Характеристика
MN	INPUT	ANY	I, Q, M, D, L	Нижняя граница
IN	INPUT	ANY	I, Q, M, D, L	Входная переменная
MX	INPUT	ANY	I, Q, M, D, L	Верхняя граница
RET_VAL	OUTPUT	ANY	I, Q, M, D, L	Ограниченная выходная переменная

Описание FC25

Функция FC 25 выбирает наибольшее из трех числовых значений переменных. В качестве входных значений допускаются значения переменных, принадлежащих к типам данных INT, DINT и REAL. Все переменные с назначенными параметрами должны принадлежать к одному типу данных.

Выходное значение остается неизменным и бит двоичного результата (BR) слова состояния устанавливается в "0", если выполняется любое из следующих условий:

- переменная с назначенными параметрами имеет недопустимый тип данных
- не все переменные с назначенными параметрами имеют один и тот же тип данных
- переменная типа REAL не представляет допустимое число с плавающей точкой.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN1	INPUT	ANY	I, Q, M, D, L	Первое входное значение
IN2	INPUT	ANY	I, Q, M, D, L	Второе входное значение
IN3	INPUT	ANY	I, Q, M, D, L	Третье входное значение
RET_VAL	OUTPUT	ANY	I, Q, M, D, L	Наибольшее из входных значений

Описание FC27

Функция FC 27 выбирает наименьшее из трех числовых значений переменных. В качестве входных значений допускаются значения переменных, принадлежащих к типам данных INT, DINT и REAL. Все переменные с назначенными параметрами должны принадлежать к одному типу данных.

Выходное значение остается неизменным и бит двоичного результата (BR) слова состояния устанавливается в "0", если выполняется любое из следующих условий:

- переменная с назначенными параметрами имеет недопустимый тип данных
- не все переменные с назначенными параметрами имеют один и тот же тип данных
- переменная типа REAL не представляет допустимое число с плавающей точкой.

Параметр	Описание	Тип данных	Область памяти	Характеристика
IN1	INPUT	ANY	I, Q, M, D, L	Первое входное значение
IN2	INPUT	ANY	I, Q, M, D, L	Второе входное значение
IN3	INPUT	ANY	I, Q, M, D, L	Третье входное значение
RET_VAL	OUTPUT	ANY	I, Q, M, D, L	Наименьшее из входных значений

21.10 Двоичный выбор: FC36

Описание FC36

Функция FC 36 выбирает, в зависимости от переключателя (параметр G), одно из двух значений переменных. В качестве входных значений в параметрах IN0 и IN1 допускаются значения переменных со всеми типами данных, которые соответствуют размеру данных бит, байт, слово и двойное слово (но не типы данных DT и STRING). Как входные переменные, так и выходная переменная должны принадлежать к одному типу данных.

Выходное значение остается неизменным и бит двоичного результата (BR) слова состояния устанавливается в "0", если выполняется любое из следующих условий:

- переменная с назначенными параметрами имеет недопустимый тип данных
- не все переменные с назначенными параметрами имеют один и тот же тип данных
- переменная типа REAL не представляет допустимое число с плавающей точкой.

Параметр	Описание	Тип данных	Область памяти	Характеристика
G	INPUT	BOOL	I, Q, M, D, L, константа	Переключатель выбора
IN0	INPUT	ANY	I, Q, M, D, L	Первое входное значение
IN1	INPUT	ANY	I, Q, M, D, L	Второе входное значение
RET_VAL	OUTPUT	ANY	I, Q, M, D, L	Выбранное входное значение

22 SFB для встроенного регулирования

Замечание

Описанные в этой главе SFB имеются только в CPU 314 IFM.

Концепция встроенного регулирования

Описанные в этой главе SFB называются SFB для встроенного регулирования, так как они встроены в операционную систему CPU 314 IFM. Они состоят из блоков для непрерывного (CONT_C) и ступенчатого (CONT_S) регулирования, а также SFB для модуляции длины импульса (PULSEGEN).

Блоки регулирования реализуют на программном уровне регулятор, в котором блок содержит все функции регулятора. Нужные для циклического вычисления данные располагаются в соответствующих блоках данных (экземплярных DB). Поэтому SFB могут вызывать их многократно.

SFB PULSEGEN используется в соединении с SFB CONT_C для того, чтобы получить регулятор с импульсным выходом для пропорциональных исполнительных устройств.

Основные функции

Регулятор, построенный с помощью SFB, складывается из ряда частных функций, которые вы можете активизировать и деактивизировать. В дополнение к собственно регулятору с его PID-алгоритмом, имеются в распоряжении также функции для воздействия на задаваемые и фактические значения, а также для воздействия на рассчитанную управляемую переменную.

Применения

Реализованный регулятор, использующий эти блоки, не ограничен с точки зрения возможных применений. Эффективность регулирования и скорость обработки зависят только от производительности используемого CPU.

В случае любого заданного CPU должен определяться компромисс между количеством регуляторов и частотой, с которой должны обрабатываться отдельные регуляторы. Чем быстрее подключаемые контуры регулирования, т.е. чем чаще должны рассчитываться управляемые переменные в единицу времени, тем меньше количество регуляторов, которые могут быть установлены.

Ограничений в отношении вида регулируемых процессов не существует. Могут регулироваться как медленные процессы (температуры, уровни наполнения и т.д.), так и очень быстрые процессы (потоки, скорости вращения и т.д.).

Анализ процесса

Статистическая характеристика (усиление) и динамические свойства (задержка, время запаздывания, постоянная интегрирования и т.д.) процесса оказывают решающее воздействие на разработку и реализацию регулятора и на расчет величин его статистических (пропорциональное воздействие) и динамических (интегрирующее воздействие и воздействие по производной) параметров.

Поэтому очень существенно точное знание типа и характеристик процесса.

Выбор регулятора

Характеристики контуров управления определяются физическими свойствами процесса или оборудования и едва ли поддаются влиянию. Следовательно, хороший результат регулирования может достигаться только за счет выбора типа регулятора, наилучшим образом подходящего к типу объекта регулирования, а также его подгонки к временным характеристикам объекта регулирования.

Реализация регулятора

Создание регулятора от начальной структуризации и назначения параметров до его вызова в системной программе возможно в значительной степени без программирования. Однако необходимо знание STEP 7.

Оперативная помощь

За дополнительной информацией об SFB обращайтесь к оперативной помощи STEP 7.

Дополнительная информация

Встроенное регулирование является подмножеством стандартных функций регулирования. Дополнительную информацию о стандартных регуляторах вы найдете в [/350/](#).

22.1 Непрерывное регулирование с помощью SFB41 "CONT_C"

Введение

SFB "CONT_C" используется в программируемых логических контроллерах SIMATIC S7 для управления техническими процессами с непрерывными входными и выходными переменными. При назначении параметров вы можете активизировать и деактивизировать частные функции PID-регулятора, чтобы адаптировать его к процессу.

Применение

Этот регулятор можно использовать как PID-регулятор с постоянными уставками или в многоконтурных системах регулирования в качестве каскадного регулятора, регулятора состава смеси или регулятора соотношения. Функции регулятора основаны на PID-алгоритме регулирования дискретного регулятора с аналоговым сигналом, дополненным, в необходимых случаях, ступенью формирования импульсов в целях формирования выходных сигналов с широтно-импульсной модуляцией для двух- или трехпозиционного регулирования с пропорциональными исполнительными звеньями.

Описание

Наряду с функциями в ветвях заданного и фактического значений, SFB реализует готовый PID-регулятор с непрерывным выходом регулирующего воздействия и возможностью влияния на управляющее воздействие вручную. Ниже вы найдете подробное описание частных функций:

Ветвь заданных значений

Заданное значение вводится на входе **SP_INT** в формате с плавающей точкой.

Ветвь фактических значений

Фактическое значение может вводиться в формате периферии или в формате с плавающей точкой. Функция **CRP_IN** преобразует периферийное значение **PV_PER** в формат с плавающей точкой в диапазоне -100 +100 % в соответствии со следующей формулой:

$$\text{Выход CRP_IN} = \text{PV_PER} * \frac{100}{27648}$$

Функция **PV_NORM** нормирует выход **CRP_IN** в соответствии со следующей формулой:

$$\text{Выход PV_NORM} = (\text{выход CRP_IN}) * \text{PV_FAC} + \text{PV_OFF}$$

PV_FAC имеет значение по умолчанию, равное 1, а **PV_OFF** значение по умолчанию, равное 0.

Сигнал ошибки

Разность между заданным и фактическим значениями дает сигнал ошибки. Для подавления малых незатухающих колебаний из-за квантования регулирующего воздействия (например, в случае широтно-импульсной модуляции с помощью **PULSEGEN**) к сигналу ошибки применяется мертвая зона (**DEADBAND**). Если **DEADB_W** = 0, то мертвая зона выключена.

PID-алгоритм

PID-алгоритм работает как позиционный алгоритм. Пропорциональное, интегрирующее (INT) и дифференцирующее (DIF) воздействия включены параллельно и могут активизироваться и деактивизироваться по отдельности. Это дает возможность проектировать P-, PI-, PD- и PID-регуляторы. Возможны также чистые I- и D-регуляторы.

Значение, вводимое вручную

Имеется возможность переключения между ручным и автоматическим режимом. В ручном режиме управляющее воздействие корректируется в соответствии со значением, выбранным вручную. Интегратор (INT) внутренне устанавливается на LMN - LMN_P - DISV, а дифференцирующее устройство (DIF) устанавливается на 0, и производится внутреннее согласование. Это значит, что переключение в автоматический режим не вызывает внезапного изменения управляющего воздействия.

Управляющее воздействие

Управляющая величина может быть ограничена выбранным значением с помощью функции LMNLIMIT. Пересечение входной величиной границ отображается сигнальными битами. Функция LMN_NORM нормирует выход функции LMNLIMIT в соответствии со следующей формулой:

$$LMN = (\text{выход LMNLIMIT}) * LMN_FAC + LMN_OFF$$

LMN_FAC по умолчанию равно 1, а LMN_OFF по умолчанию равно 0.

Управляющее значение доступно также в периферийном формате. Функция CRP_OUT преобразует значение с плавающей точкой LMN в периферийное значение в соответствии со следующей формулой:

$$LMN_PER = LMN * \frac{27648}{100}$$

Возмущающее воздействие

Возмущающее воздействие может быть подано на вход DISV.

Инициализация

SFB41 "CONT_C" имеет подпрограмму инициализации, которая прогоняется, когда установлен входной параметр COM_RST = TRUE.

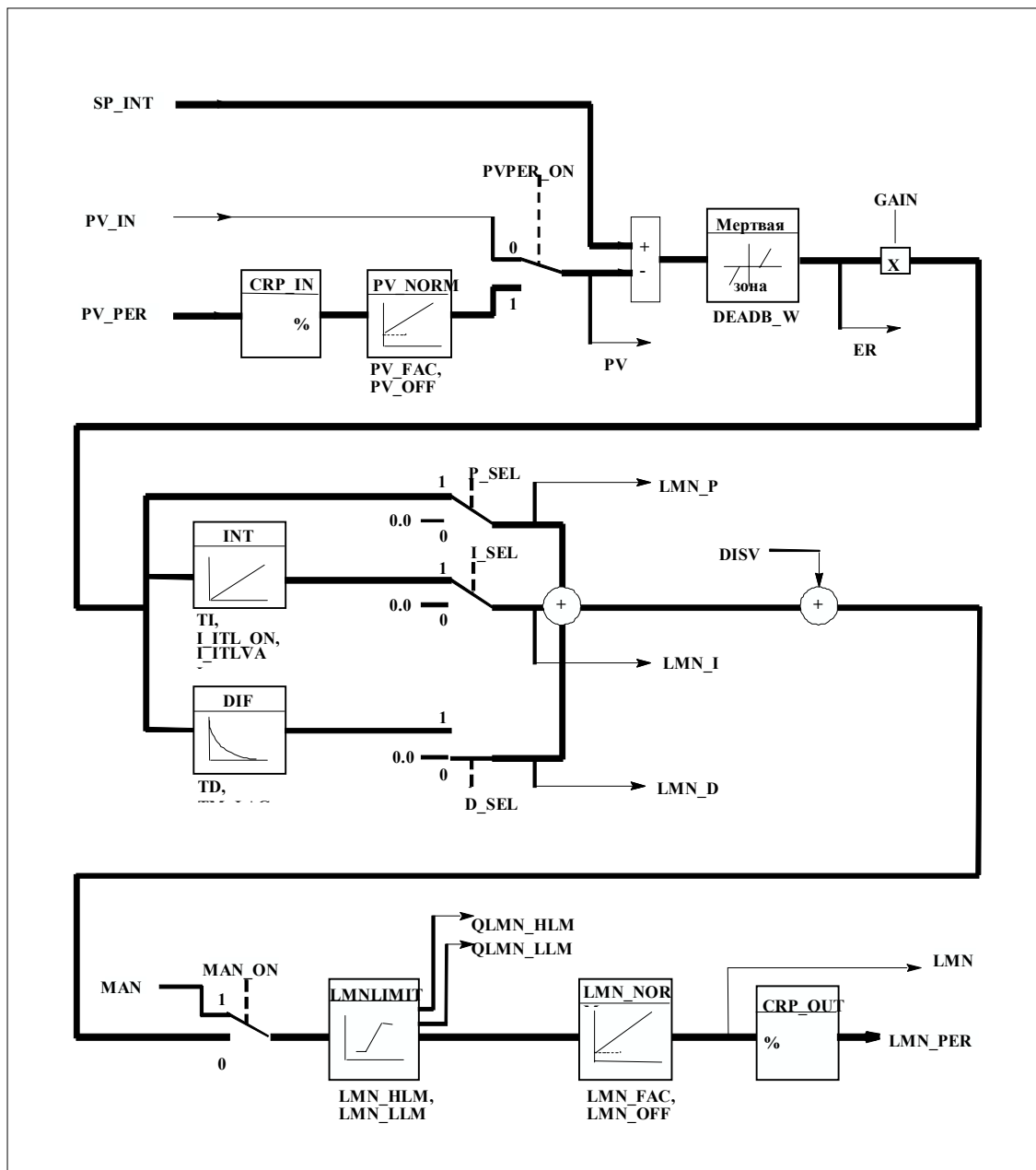
Во время инициализации интегратор устанавливается внутренне на значение инициализации I_ITVAL. Когда он вызывается в классе приоритета циклических прерываний, он затем продолжает работать, начиная с этого значения.

Все остальные выходы устанавливаются на их значения по умолчанию.

Информация об ошибках

Параметр вывода ошибок RET_VAL не используется.

Блок-схема CONT_C



Входные параметры

Следующая таблица содержит описание входных параметров SFB41 "CONT_C".

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
COM_RST	BOOL		FALSE	COMPLETE RESTART/ Полный рестарт Блок имеет подпрограмму инициализации, которая обрабатывается, когда установлен вход "Полный рестарт".
MAN_ON	BOOL		TRUE	MANUAL VALUE ON / Включение ручного режима Если этот вход установлен, то контур управления разрывается. Значение, заданное вручную, устанавливается в качестве управляющего.
PVPER_ON	BOOL		FALSE	PROCESS VARIABLE PERIPHERAL ON / Включение фактического значения периферии Если фактическое значение считывается из периферии, то вход PV_PER должен быть соединен с периферией, а вход "Включение фактического значения периферии" должен быть установлен.
P_SEL	BOOL		TRUE	PROPORTIONAL ACTION ON / Включение пропорционального воздействия В PID-алгоритме отдельные воздействия можно активизировать или деактивизировать индивидуально. Пропорциональное воздействие (P-воздействие) включено, когда установлен вход "Включение пропорционального воздействия".
I_SEL	BOOL		TRUE	INTEGRAL ACTION ON / Включение интегрирующего воздействия В PID-алгоритме отдельные воздействия можно активизировать или деактивизировать индивидуально. Интегрирующее воздействие (I-воздействие) включено, когда установлен вход "Включение интегрирующего воздействия".
INT_HOLD	BOOL		FALSE	INTEGRAL ACTION HOLD / Интегрирующее воздействие заморожено Выход интегратора может быть "заморожен" установкой входа "Интегрирующее воздействие заморожено".
I_ITL_ON	BOOL		FALSE	INITIALIZATION OF THE INTEGRAL ACTION ON / Включение инициализации I-воздействия Выход интегратора может быть подключен к входу I_ITL_VAL установкой входа "Включение инициализации I-воздействия".
D_SEL	BOOL		FALSE	DERIVATIVE ACTION ON / Включение воздействия по производной В PID-алгоритме отдельные воздействия можно активизировать или деактивизировать индивидуально. Воздействие по производной (D-воздействие) включено, когда установлен вход "Включение воздействия по производной".

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
CYCLE	TIME	>= 1 мс	T#1s	SAMPLING TIME / Время опроса Время между вызовами блока должно быть постоянным. Вход "Время опроса" задает время между вызовами блока.
SP_INT	REAL	От -100.0 до +100.0 (%) или физическая величина ¹⁾	0.0	INTERNAL SETPOINT / Внутреннее заданное значение Вход "Внутреннее заданное значение" служит для установления заданного значения.
PV_IN	REAL	От -100.0 до +100.0 (%) или физическая величина ¹⁾	0.0	PROCESS VARIABLE IN / Ввод фактического значения На входе "Вход фактического значения" может быть установлено значение инициализации или подключено внешнее фактическое значение в формате с плавающей точкой.
PV_PER	WORD		W#16#0000	PROCESS VARIABLE PERIPHERAL/ Фактическое значение периферии Фактическое значение в формате периферии подключается к регулятору на входе "Фактическое значение периферии".
MAN	REAL	От -100.0 до +100.0 (%) или физическая величина ²⁾	0.0	MANUAL VALUE / Ручное значение Вход "Ручное значение" используется для установки ручного значения с помощью функций взаимодействия с оператором.
GAIN	REAL		2.0	PROPORTIONAL GAIN / Коэффициент пропорциональности Вход "Коэффициент пропорциональности" задает коэффициент усиления регулятора.
TI	TIME	>= CYCLE	T#20s	RESET TIME / Время интегрирования Вход "Время интегрирования" определяет временную характеристику интегратора.
TD	TIME	>= CYCLE	T#10s	DERIVATIVE TIME / Время дифференцирования Вход "Время дифференцирования" определяет временную характеристику дифференцирующего звена.
TM_LAG	TIME	>= CYCLE/2	T#2s	TIME LAG OF THE DERIVATIVE ACTION / Время запаздывания D-воздействия Алгоритм D-компоненты содержит запаздывание, которое может быть назначено входу "Время запаздывания D-воздействия".
DEADB_W	REAL	>= 0.0 (%) или физическая величина ¹⁾	0.0	DEAD BAND WIDTH / Ширина мертвой зоны Мертвая зона относится к ошибке. Вход "Ширина мертвой зоны" определяет размер мертвой зоны.
LMN_HLM	REAL	LMN_LLM ... 100.0 (%) или физическая величина ²⁾	100.0	MANIPULATED VALUE HIGH LIMIT/ Верхняя граница управляющего воздействия Управляющее воздействие всегда ограничивается верхней и нижней границами. Вход "Верхняя граница управляющего воздействия" задает верхнюю границу.

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
LMN_LLM	REAL	-100.0... LMN_HLM (%) или физическая ²⁾ величина	0.0	MANIPULATED VALUE LOW LIMIT / Нижняя граница управляющего воздействия Управляющее воздействие всегда ограничивается верхней и нижней границами. Вход "Нижняя граница управляющего воздействия" задает нижнюю границу.
PV_FAC	REAL		1.0	PROCESS VARIABLE FACTOR/ Коэффициент фактического значения Вход "Коэффициент фактического значения" умножается на фактическое значение. Вход служит для настройки диапазона фактических значений.
PV_OFF	REAL		0.0	PROCESS VARIABLE OFFSET/ Смещение фактического значения Вход "Смещение фактического значения" складывается с фактическим значением. Вход служит для настройки диапазона фактических значений.
LMN_FAC	REAL		1.0	MANIPULATED VALUE FACTOR Коэффициент управляющего воздействия Вход "Коэффициент управляющего воздействия" умножается на управляющее воздействие. Вход служит для настройки диапазона управляющих воздействий.
LMN_OFF	REAL		0.0	MANIPULATED VALUE OFFSET/ Смещение управляющего воздействия Вход "Смещение управляющего воздействия" складывается с управляющим воздействием. Вход служит для настройки диапазона управляющих воздействий.
I_ITLVAL	REAL	От -100.0 до +100.0 (%) или физическая ²⁾ величина	0.0	INITIALIZATION VALUE OF THE INTEGRAL ACTION / Инициализирующее значение интегрирующего воздействия Выход интегратора может быть установлен на входе I_ITL_ON. Инициализирующее значение прикладывается к входу "Инициализирующее значение интегрирующего воздействия".
DISV	REAL	От -100.0 до +100.0 (%) или физическая ²⁾ величина	0.0	DISTURBANCE VARIABLE / Возмущающее воздействие Возмущающее воздействие подключается к входу "Возмущающее воздействие" для упреждающего регулирования.

1) Параметры в ветвях заданного и фактического значений с одинаковой единицей измерения

2) Параметры в ветви задающего воздействия с одинаковой единицей измерения

Выходные параметры

Следующая таблица содержит описание выходных параметров SFB41 "CONT_C".

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
LMN	REAL		0.0	MANIPULATED VALUE/ Управляющее воздействие На выходе "Управляющее воздействие" выводится эффективное управляющее воздействие в формате с плавающей точкой.
LMN_PER	WORD		W#16#0000	MANIPULATED VALUE PERIPHERAL / Управляющее воздействие в формате периферии К регулятору на этом выходе подключается управляющее воздействие в формате периферии.
QLMN_HLM	BOOL		FALSE	HIGH LIMIT OF MANIPULATED VALUE REACHED / Достигнута верхняя граница управляющего воздействия Управляющее воздействие всегда ограничено сверху и снизу. Выход "Достигнута верхняя граница управляющего воздействия" указывает на переход через верхнюю границу.
QLMN_LLM	BOOL		FALSE	LOW LIMIT OF MANIPULATED VALUE REACHED/ Достигнута нижняя граница управляющего воздействия Управляющее воздействие всегда ограничено сверху и снизу. Выход "Достигнута нижняя граница управляющего воздействия" указывает на переход через нижнюю границу.
LMN_P	REAL		0.0	PROPORTIONAL COMPONENT / Пропорциональная составляющая Выход "Пропорциональная составляющая" содержит пропорциональную составляющую управляющего воздействия.
LMN_I	REAL		0.0	INTEGRAL COMPONENT / Интегральная составляющая Выход "Интегральная составляющая" содержит интегральную составляющую управляющего воздействия.
LMN_D	REAL		0.0	DERIVATIVE COMPONENT/ Дифференциальная составляющая Выход "Дифференциальная составляющая" содержит дифференциальную составляющую управляющего воздействия.
PV	REAL		0.0	PROCESS VARIABLE / Фактическое значение На выходе "Фактическое значение" выводится эффективное фактическое значение.
ER	REAL		0.0	ERROR SIGNAL / Сигнал ошибки На выходе "Сигнал ошибки" выводится эффективное значение сигнала ошибки.

22.2 Ступенчатое регулирование с помощью SFB42 "CONT_S"

Введение

SFB42 "CONT_S" используется в программируемых логических контроллерах SIMATIC S7 для управления техническими процессами с помощью дискретных выходных сигналов управляющего воздействия для интегрирующих исполнительных звеньев. При назначении параметров вы можете активизировать или деактивизировать частные функции ступенчатого PI-регулятора, чтобы настроить его на процесс.

Применение

Этот регулятор можно использовать как PI-регулятор с постоянными уставками или во вторичных контурах регулирования в многоконтурных системах регулирования в качестве каскадного регулятора, регулятора состава смеси или регулятора соотношения, но не в качестве следящего регулятора. Принцип работы регулятора основан на PI-алгоритме управления дискретного регулятора, дополненном функциями для генерирования двоичного выходного сигнала из аналогового управляющего сигнала.

Описание

Наряду с функциями в ветви фактических значений, SFB реализует готовый PI-регулятор с двоичным выходом управляющего воздействия и возможностью влияния на управляющее воздействие вручную. Ступенчатый регулятор действует без сигнала обратной связи по положению. Ниже вы найдете описание частных функций:

Ветвь заданных значений

Заданное значение вводится на входе **SP_INT** в формате с плавающей точкой.

Ветвь фактических значений

Фактическое значение может вводиться в формате периферии или в формате с плавающей точкой. Функция **CRP_IN** преобразует периферийное значение **PV_PER** в формат с плавающей точкой в диапазоне от -100 до +100 % в соответствии со следующей формулой:

$$\text{Выход CRP_IN} = \text{PV_PER} * \frac{100}{27648}$$

Функция **PV_NORM** нормирует выход **CRP_IN** в соответствии со следующей формулой:

$$\text{Выход PV_NORM} = (\text{выход CRP_IN}) * \text{PV_FAC} + \text{PV_OFF}$$

PV_FAC по умолчанию равно 1, а **PV_OFF** по умолчанию равно 0.

Сигнал ошибки

Разность между заданным и фактическим значениями дает сигнал ошибки. Для подавления малых незатухающих колебаний из-за квантования регулирующего воздействия (например, из-за ограниченной разрешающей способности управляющего воздействия из-за регулирующего клапана) к сигналу ошибки применяется мертвая зона (**DEADBAND**). Если **DEADB_W** = 0, то мертвая зона выключена.

Ступенчатый PI-алгоритм

SFB работает без сигнала обратной связи по положению. I-воздействие PI-алгоритма и воображаемый сигнал обратной связи по положению рассчитываются в **одном** интеграторе (INT) и сравниваются с остающимся P-воздействием в качестве значения обратной связи. Разность поступает на трехпозиционное звено (THREE_ST) и генератор импульсов (PULSEOUT), который вырабатывает импульсы для исполнительного устройства. Частота переключения регулятора может быть уменьшена настройкой порога в трехпозиционном звене.

Возмущающее воздействие

Возмущающее воздействие может быть подано на вход **DISV**.

Инициализация

SFB42 "CONT_S" имеет подпрограмму инициализации, которая прогоняется, когда установлен входной параметр COM_RST = TRUE.

Все остальные выходы устанавливаются на их значения по умолчанию.

Информация об ошибках

Параметр вывода ошибок RET_VAL не используется.

Входные параметры

Следующая таблица содержит описание входных параметров SFB42 "CONT_S".

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
COM_RST	BOOL		FALSE	COMPLETE RESTART/ Полный рестарт Блок имеет подпрограмму инициализации, которая обрабатывается, когда установлен вход "Полный рестарт".
LMNR_HS	BOOL		FALSE	HIGH LIMIT OF POSITION FEEDBACK SIGNAL/ Верхняя граница сигнала обратной связи по положению Сигнал "Исполнительное устройство у упора верхней границы" подключен к входу "Верхняя граница сигнала обратной связи по положению". LMNR_HS=TRUE означает, что исполнительное устройство находится у упора верхней границы.
LMNR_LS	BOOL		FALSE	LOW LIMIT OF POSITION FEEDBACK SIGNAL/ Нижняя граница сигнала обратной связи по положению Сигнал "Исполнительное устройство у упора нижней границы" подключен к входу "Нижняя граница сигнала обратной связи по положению". LMNR_LS=TRUE означает, что исполнительное устройство находится у упора нижней границы.
LMNS_ON	BOOL		TRUE	MANUAL ACTUATING SIGNALS ON /Включение сигналов ручного воздействия Обработка управляющего сигнала включена на ручной режим на входе "Включение сигналов ручного воздействия".
LMNUP	BOOL		FALSE	ACTUATING SIGNALS UP / Увеличение управляющих сигналов В случае сигналов управляющего воздействия, вводимых вручную, выходной сигнал QLMNUP устанавливается на входе "Увеличение управляющих сигналов".
LMNDN	BOOL		FALSE	ACTUATING SIGNALS DOWN/ Уменьшение управляющих сигналов В случае сигналов управляющего воздействия, вводимых вручную, выходной сигнал QLMNDN устанавливается на входе "Уменьшение управляющих сигналов".
PVPER_ON	BOOL		FALSE	PROCESS VARIABLE PERIPHERAL ON/ Включение фактического значения периферии Если фактическое значение считывается из периферии, то вход PV_PER должен быть соединен с периферией, а вход "Включение фактического значения периферии" должен быть установлен.
CYCLE	TIME	>= 1мс	T#1s	SAMPLING TIME/ Время опроса Время между вызовами блока должно быть постоянным. Вход "Время опроса" задает время между вызовами блока.

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
SP_INT	REAL	От -100.0 до +100.0 (%) или физическая ¹⁾ величина	0.0	INTERNAL SETPOINT/ Внутреннее заданное значение Вход "Внутреннее заданное значение" служит для установления заданного значения.
PV_IN	REAL	От -100.0 до +100.0 (%) или физическая ¹⁾ величина	0.0	PROCESS VARIABLE IN/ Ввод фактического значения На входе "Вход фактического значения" может быть установлено значение инициализации или подключено внешнее фактическое значение в формате с плавающей точкой.
PV_PER	WORD		W#16#0000	PROCESS VARIABLE PERIPHERAL/ Фактическое значение периферии Фактическое значение в формате периферии подключается к регулятору на входе "Фактическое значение периферии".
GAIN	REAL		2.0	PROPORTIONAL GAIN/ Коэффициент пропорциональности Вход "Коэффициент пропорциональности" задает коэффициент усиления регулятора.
TI	TIME	>= CYCLE	T#20s	RESET TIME/ Время интегрирования Вход "Время интегрирования" определяет временную характеристику интегратора.
DEADB_W	REAL	0.0...100.0 (%) или физическая ¹⁾ величина	1.0	DEAD BAND WIDTH/ Ширина мертвой зоны Мертвая зона относится к ошибке. Вход "Ширина мертвой зоны" определяет размер мертвой зоны.
PV_FAC	REAL		1.0	PROCESS VARIABLE FACTOR / Коэффициент фактического значения Вход "Коэффициент фактического значения" умножается на фактическое значение. Вход служит для настройки диапазона фактических значений.
PV_OFF	REAL		0.0	PROCESS VARIABLE OFFSET/ Смещение фактического значения Вход "Смещение фактического значения" складывается с фактическим значением. Вход служит для настройки диапазона фактических значений.
PULSE_TM	TIME	>= CYCLE	T#3s	MINIMUM PULSE TIME /Минимальная длительность импульса Минимальная продолжительность импульса может быть установлена с помощью параметра "Минимальная длительность импульса".
BREAK_TM	TIME	>= CYCLE	T#3s	MINIMUM BREAK TIME / Минимальное время паузы Минимальная продолжительность паузы может быть установлена с помощью параметра "Минимальное время паузы".

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
MTR_TM	TIME	>= CYCLE	T#30s	MOTOR ACTUATING TIME / Время перестановки двигателя Время, требуемое исполнительному устройству для перемещения от упора до упора, вводится в параметре "Время перестановки двигателя".
DISV	REAL	От -100.0 до +100.0 (%) или физическая величина ²⁾	0.0	DISTURBANCE VARIABLE/ Возмущающее воздействие Возмущающее воздействие подключается к входу "Возмущающее воздействие" для упреждающего регулирования.

1) Параметры в ветвях заданного и фактического значений с одинаковой единицей измерения

2) Параметры в ветви управляющего воздействия с одинаковой единицей измерения

Выходные параметры

Следующая таблица содержит описание выходных параметров SFB42 "CONT_S".

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
QLMNUP	BOOL		FALSE	ACTUATING SIGNAL UP / Увеличение управляющего сигнала Если выход "Увеличение управляющего сигнала" установлен, то регулирующий клапан открыт.
QLMNDN	BOOL		FALSE	ACTUATING SIGNAL DOWN Уменьшение управляющего сигнала Если выход "Уменьшение управляющего сигнала" установлен, то регулирующий клапан закрыт.
PV	REAL		0.0	PROCESS VARIABLE / Фактическое значение На выходе "Фактическое значение" выводится эффективное фактическое значение.
ER	REAL		0.0	ERROR SIGNAL / Сигнал ошибки На выходе "Сигнал ошибки" выводится эффективное значение сигнала ошибки.

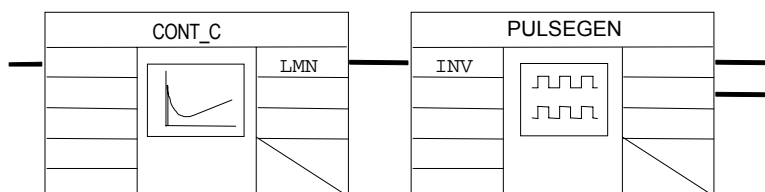
22.3 Формирование импульсов с помощью SFB43 "PULSEGEN"

Введение

SFB43 "PULSEGEN" служит для построения PID-регулятора с импульсным выходом для пропорциональных исполнительных звеньев

Применение

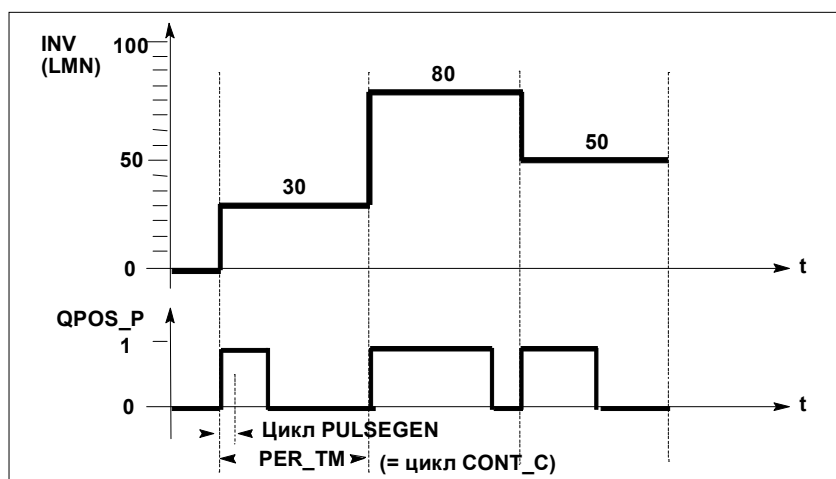
С помощью SFB "PULSEGEN" можно проектировать двух- или трехпозиционные PID-регуляторы с широтно-импульсной модуляцией. Эта функция обычно используется в соединении с непрерывным регулятором "CONT_C".



Описание

Функция PULSEGEN преобразует входную переменную INV (= управляющее воздействие PID-регулятора) путем модуляции длительности импульса в последовательность импульсов с постоянным периодом следования, который соответствует времени цикла обновления входной переменной и должен быть назначен в PER_TM.

Длительность импульса на протяжении периода пропорциональна входной величине. Цикл, назначенный PER_TM, не идентичен циклу обработки SFB "PULSEGEN." Цикл PER_TM складывается из нескольких циклов обработки SFB "PULSEGEN," причем количество вызовов SFB "PULSEGEN" за цикл PER_TM представляет собой меру точности широтно-импульсной модуляции.

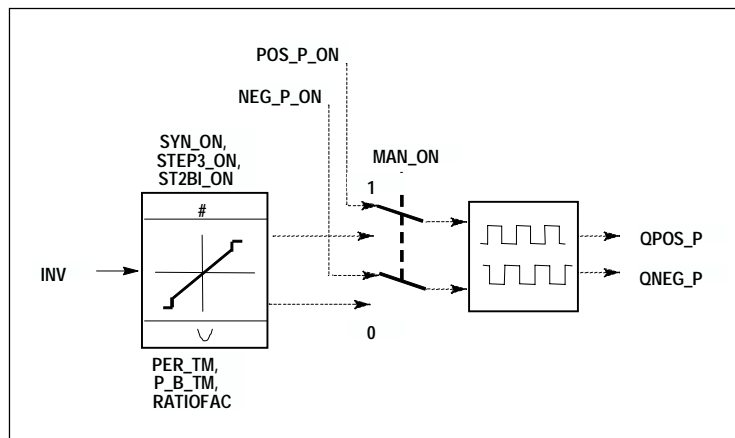


Широтно-импульсная модуляция

Входная величина 30% и 10 вызовов SFB "PULSEGEN" на PER_TM означает следующее:

- "единицу" на выходе QPOS для первых трех вызовов SFB "PULSEGEN" (30% от 10 вызовов)
- "нуль" на выходе QPOS для семи последующих вызовов SFB "PULSEGEN" (70% от 10 вызовов)

Блок-схема



Точность управляющего воздействия

В этом примере за счет "соотношения опроса" 1:10 (отношение количества вызовов CONT_C к количеству вызовов PULSEGEN) точность управляющего воздействия ограничена 10%, то есть заданные входные значения INV могут быть имитированы длительностью импульсов на выходе QPOS только шагами в 10 %.

С увеличением количества вызовов SFB "PULSEGEN" на вызов CONT_C точность повышается.

Например, если PULSEGEN вызывается в 100 раз чаще, чем CONT_C, то достигается разрешение в 1% от диапазона управляющего воздействия.

Замечание

Частота вызовов должна программироваться пользователем.

Автоматическая синхронизация

Существует возможность синхронизировать импульсный вывод с блоком, который обновляет входную переменную INV (например, CONT_C). Это обеспечивает максимально быстрый вывод изменения входной переменной в виде импульса.

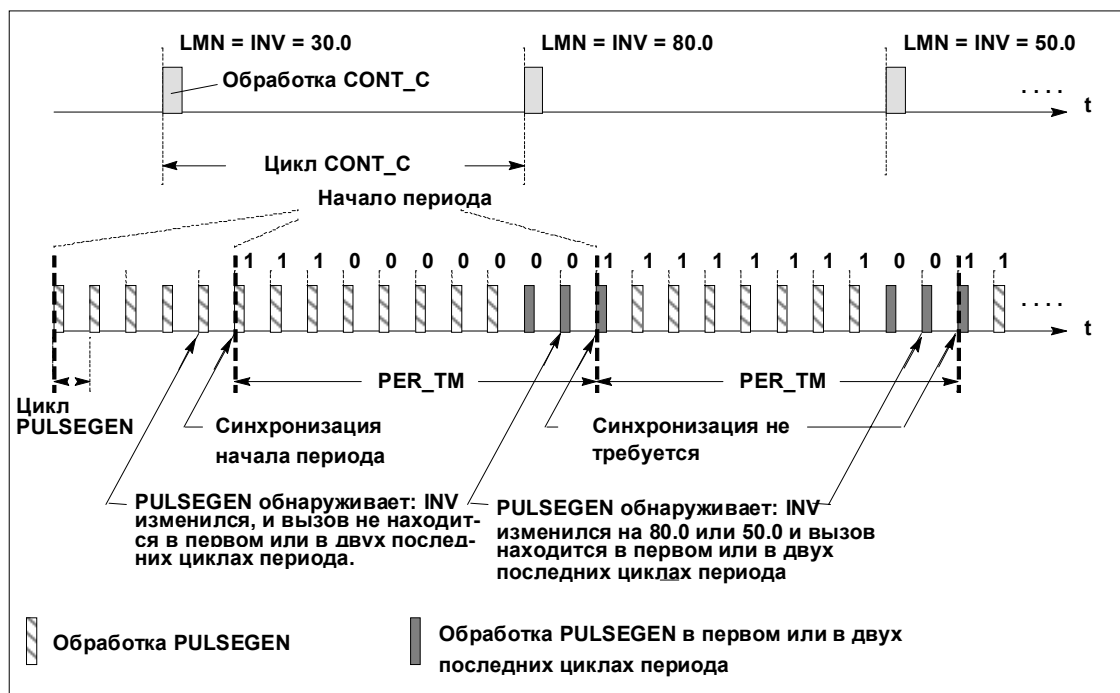
Формирователь импульсов оценивает входную величину на интервалах времени длиной в период PER_TM и преобразует это значение в импульсный сигнал соответствующей длительности.

Однако, так как INV обычно рассчитывается в более медленном классе циклических прерываний, то формирователь импульсов должен начинать преобразование дискретного значения в импульсный сигнал как можно скорее после обновления INV.

Этот блок может синхронизировать начало периода с помощью следующей процедуры:

Если INV меняется и вызов блока не находится в первом или в двух последних циклах вызова периода, то проводится синхронизация.

Длительность импульса вычисляется снова, и в следующем цикле вывод происходит с новым периодом.



Автоматическая синхронизация может быть заблокирована на входе "SYN_ON" (= FALSE).

Замечание

В начале нового периода старое значение INV (то есть LMN) имитируется импульсным сигналом после синхронизации более или менее точно.

Режимы работы

В зависимости от параметров, назначенных генератору импульсов, могут быть спроектированы PID-регуляторы с трехпозиционным выходом или с биполярным или униполярным двухпозиционным выходом. Следующая таблица иллюстрирует установку комбинаций выключателей для возможных режимов.

Режим	Ключ		
	MAN_ON	STEP3_ON	ST2BI_ON
Трехпозиционное регулирование	FALSE	TRUE	Любое
Двухпозиционное регулирование с биполярным диапазоном (от -100 % до +100 %)	FALSE	FALSE	TRUE
Двухпозиционное регулирование с униполярным диапазоном (0 % ... 100 %)	FALSE	FALSE	FALSE
Ручной режим	TRUE	Любое	Любое

Трехпозиционное регулирование

В режиме “Трехпозиционное регулирование” управляющий сигнал может принимать три состояния. Значения двоичных выходных сигналов QPOS_P и QNEG_P ставятся в соответствие состояниям исполнительного устройства. В таблице показан пример регулирования температуры:

Выходные сигналы	Исполнительное устройство		
	Нагрев	Выключено	Охлаждение
QPOS_P	TRUE	FALSE	FALSE
QNEG_P	FALSE	FALSE	TRUE

Исходя из входной величины, по характеристической кривой рассчитывается длительность импульса. Форма этой характеристической кривой определяется минимальной длительностью импульса или минимальной длительностью паузы и коэффициентом соотношения.

Нормальное значение коэффициента соотношения равно 1.

Точки излома на кривых определяются минимальной длительностью импульса или паузы.

Минимальная длительность импульса или паузы

Правильно назначенная минимальная длительность импульса или паузы P_B_TM может предотвратить короткие времена включения или выключения, которые могут отрицательно влиять на срок службы переключательных звеньев и исполнительного оборудования.

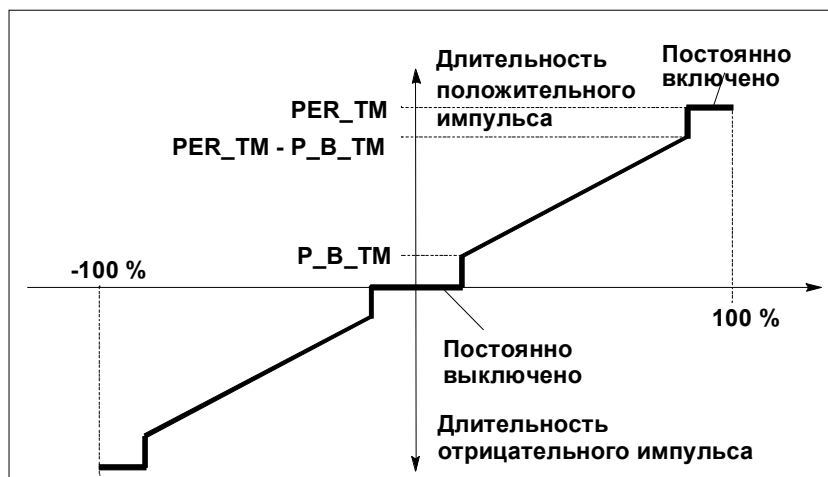
Замечание

Малые абсолютные значения входной величины LMN, которые создавали бы длительность импульса меньше, чем P_B_TM, подавляются. Большие входные значения, которые создавали бы длительность импульса больше, чем (PER_TM - P_B_TM), устанавливаются на 100 % или -100 %.

Длительность положительных или отрицательных импульсов рассчитывается из входной величины (в %), умноженной на длительность периода.

$$\text{Длительность импульса} = \frac{\text{INV}}{100} * \text{PER_TM}$$

На следующем рисунке показана симметричная характеристика трехпозиционного регулятора (коэффициент соотношения = 1).



Несимметричное трехпозиционное регулирование

С помощью коэффициента соотношения **RATIOFAC** можно изменять отношение длительности положительных импульсов к длительности отрицательных импульсов. Благодаря этому, например, в случае термического процесса можно учитывать разные постоянные времени объекта регулирования для нагревания и охлаждения.

Коэффициент соотношения влияет также на минимальную длительность импульса или паузы. Коэффициент соотношения < 1 означает, что пороговое значение для отрицательных импульсов умножается на этот коэффициент.

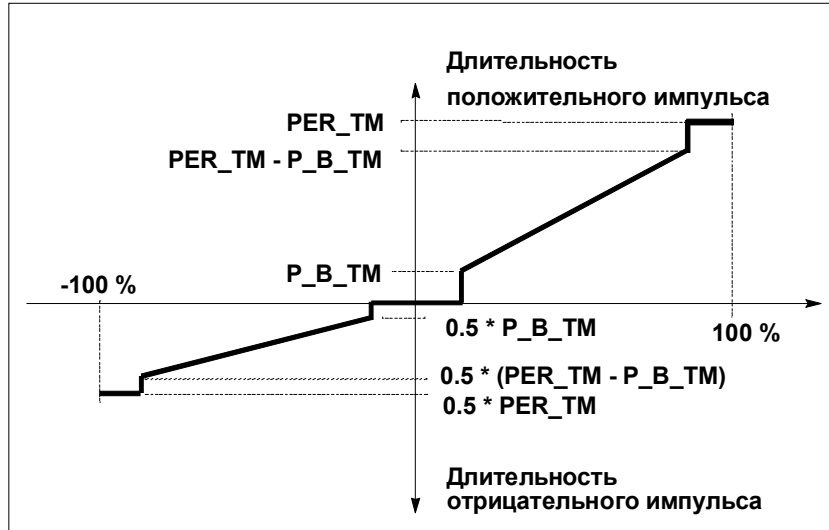
Коэффициент соотношения < 1

Длительность импульса на выходе отрицательных импульсов, рассчитанная из входной величины, умноженной на длительность периода, сокращается пропорционально коэффициенту соотношения (см. рисунок).

$$\text{Длительность положительного импульса} = \frac{\text{INV}}{100} * \text{PER_TM}$$

$$\text{Длительность отрицательного импульса} = \frac{\text{INV}}{100} * \text{PER_TM} * \text{RATIOFAC}$$

На следующем рисунке показана асимметричная характеристика трехступенчатого регулятора (коэффициент соотношения = 0.5):



Коэффициент соотношения > 1

Длительность импульса на выходе положительных импульсов, рассчитанная из входной величины, умноженной на длительность периода, изменяется обратно пропорционально коэффициенту соотношения.

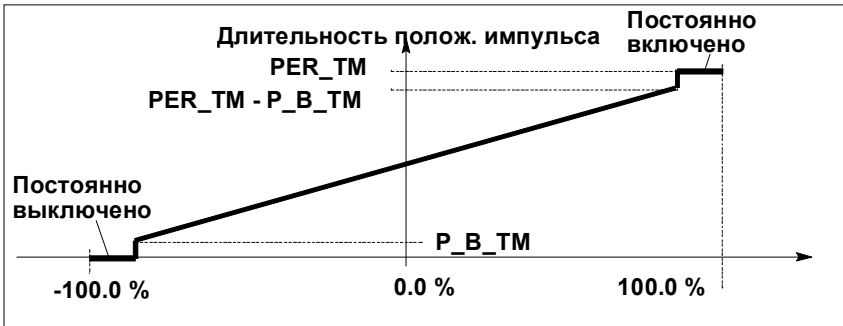
$$\text{Длительность отрицательного импульса} = -\frac{\text{INV}}{100} * \text{PER_TM}$$

$$\text{Длительность положительного импульса} = \frac{\text{INV}}{100} * \frac{\text{PER_TM}}{\text{RATIOFAC}}$$

Двухпозиционное регулирование

При двухпозиционном регулировании к исполнительному устройству типа "включено-выключено" подключается только выход положительных импульсов QPOS_P блока PULSEGEN. При двухпозиционном регулировании с соответствующим исполнительным звеном типа "включено-выключено" связывается только положительный импульсный выход QPOS_P блока PULSEGEN. В зависимости от используемого диапазона управляющего воздействия двухпозиционный регулятор имеет биполярный или униполярный диапазон управляющего воздействия.

Двухпозиционное регулирование с биполярным диапазоном управляющего воздействия (от -100% до 100%)



Двухпозиционное регулирование с униполярным диапазоном управляющего воздействия (от 0% до 100%)



На QNEG_P имеется в распоряжении инвертированный выходной сигнал на тот случай, когда включение двухпозиционного регулятора в контуре регулирования требует логически инвертированного двоичного сигнала для управляющих импульсов.

Импульс	Исполнительное устройство	
	Включено	Выключено
QPOS_P	TRUE	FALSE
QNEG_P	FALSE	TRUE

Ручной режим при двух- или трехпозиционном регулировании

В ручном режиме (MAN_ON = TRUE) двоичные выходы трех- или двухпозиционного регулятора могут устанавливаться сигналами POS_P_ON и NEG_P_ON независимо от INV.

	POS_P_ON	NEG_P_ON	QPOS_P	QNEG_P
Трехпозиционное регулирование	FALSE	FALSE	FALSE	FALSE
	TRUE	FALSE	TRUE	FALSE
	FALSE	TRUE	FALSE	TRUE
	TRUE	TRUE	FALSE	FALSE
Двухпозиционное регулирование	FALSE	Any	FALSE	TRUE
	TRUE	Any	TRUE	FALSE

Инициализация

SFB42 "PULSGEN" имеет подпрограмму инициализации, которая прогоняется, когда установлен входной параметр COM_RST = TRUE.

Все выходы сигналов устанавливаются в 0.

Информация об ошибках

Параметр вывода ошибок RET_VAL не используется.

Входные параметры

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
INV	REAL	-100.0...100.0 (%)	0.0	INPUT VARIABLE/ Входная переменная К входному параметру "Входная переменная" подключается аналоговое управляющее воздействие
PER_TM	TIME	>=20*CYCLE	T#1s	PERIOD TIME / Длительность периода С помощью входного параметра "Длительность периода" вводится постоянный период широтно-импульсной модуляции. Это соответствует времени опроса регулятора. Соотношение между временем опроса генератора импульсов и временем опроса регулятора определяет точность широтно-импульсной модуляции.
P_B_TM	TIME	>= CYCLE	T#0ms	MINIMUM PULSE/BREAK TIME / Минимальная длительность импульса/паузы Минимальная длительность импульса или паузы, которая может быть назначена во входных параметрах "Минимальная длительность импульса или паузы"
RATIOFAC	REAL	0.1 ...10.0	1.0	RATIO FACTOR / Коэффициент соотношения Входной параметр "Коэффициент соотношения" может быть использован для изменения отношения длительности отрицательного импульса к положительному. В термическом процессе это позволило бы, например, компенсировать разность постоянных времени нагрева и охлаждения (например, в процессе с электрическим нагревом и водяным охлаждением).

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
STEP3_ON	BOOL		TRUE	THREE STEP CONTROL ON / Включение трехпозиционного регулирования Входной параметр "Включение трехпозиционного регулирования" активизирует этот режим. При трехпозиционном регулировании активны оба выходных сигнала.
ST2BI_ON	BOOL		FALSE	TWO STEP CONTROL FOR BIPOLAR MANIPULATED VALUE RANGE ON / Включение двухпозиционного регулирования для биполярного управляющего воздействия С помощью входного параметра "Включение двухпозиционного регулирования для биполярного управляющего воздействия" вы можете выбирать между режимами "Двухпозиционное регулирование для биполярного управляющего воздействия" и "Двухпозиционное регулирование для униполярного управляющего воздействия". Должен быть установлен параметр STEP3_ON = FALSE.
MAN_ON	BOOL		FALSE	MANUAL MODE ON / Включение ручного режима При установке входного параметра "Включение ручного режима" выходные сигналы могут устанавливаться вручную.
POS_P_ON	BOOL		FALSE	POSITIVE PULSE ON / Включение положительного импульса В ручном режиме при трехпозиционном регулировании выходной сигнал QPOS_P может быть установлен через входной параметр "Включение положительного импульса". В ручном режиме при двухпозиционном регулировании, QNEG_P всегда устанавливается инверсно по отношению к QPOS_P.

Параметр	Тип данных	Диапазон значений	Значение по умолчанию	Описание
NEG_P_ON	BOOL		FALSE	NEGATIVE PULSE ON / Включение отрицательного импульса В ручном режиме при трехпозиционном регулировании выходной сигнал QNEG_P может быть установлен через входной параметр "Включение отрицательного импульса". В ручном режиме при двухпозиционном регулировании, QNEG_P всегда устанавливается инверсно по отношению к QPOS_P.
SYN_ON	BOOL		TRUE	SYNCHRONIZATION ON / Включение синхронизации При установке входного параметра "Включение синхронизации" имеется возможность автоматической синхронизации с блоком, обновляющим входную переменную INV. Это гарантирует максимально быстрый вывод изменения входной переменной в виде импульса.
COM_RST	BOOL		FALSE	COMPLETE RESTART / Полный рестарт Блок имеет подпрограмму инициализации, которая обрабатывается, когда установлен вход COM_RST.
CYCLE	TIME	>= 1мс	T#10ms	SAMPLING TIME / Длительность цикла опроса Время между вызовами блока должно быть постоянным. Вход "Длительность цикла опроса" определяет интервал между вызовами блока.

Замечание

Значения входных параметров в блоке не ограничиваются; проверка параметров не происходит.

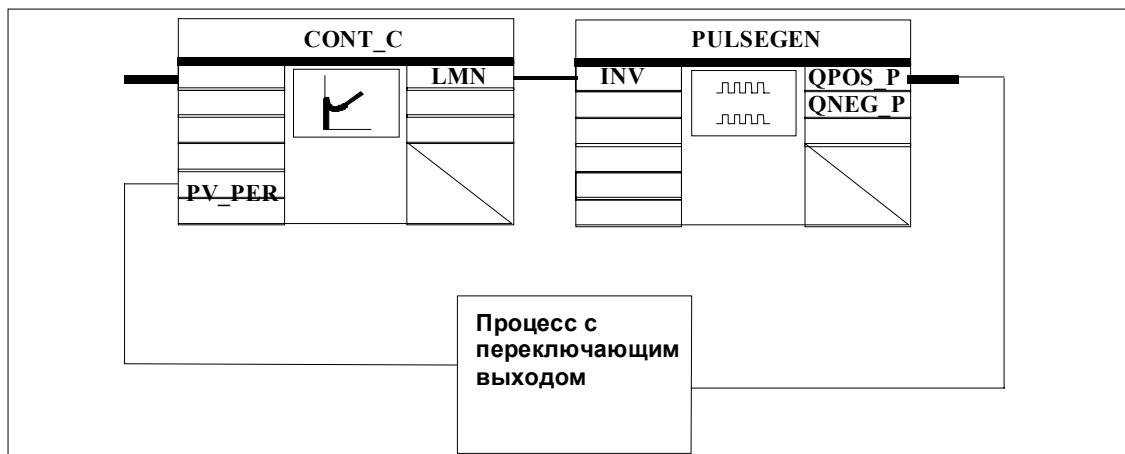
Выходные параметры

Параметр	Тип данных	Значения	Значения по умолчанию	Описание
QPOS_P	BOOL		FALSE	OUTPUT POSITIVE PULSE / Выходной сигнал - положительный импульс Выходной параметр "Выходной сигнал - положительный импульс" устанавливается, когда должен выводиться импульс. При трехпозиционном регулировании это всегда положительный импульс. При двухпозиционном регулировании QNEG_P всегда устанавливается инверсно по отношению к QPOS_P.
QNEG_P	BOOL		FALSE	OUTPUT NEGATIVE PULSE / Выходной сигнал - отрицательный импульс Выходной параметр "Выходной сигнал - отрицательный импульс" устанавливается, когда должен выводиться импульс. При трехпозиционном регулировании это всегда отрицательный импульс. При двухпозиционном регулировании QNEG_P всегда устанавливается инверсно по отношению к QPOS_P.

22.4 Пример блока PULSEGEN

Контур регулирования

С помощью регулятора непрерывного действия CONT_C и импульсного генератора PULSEGEN вы можете реализовать регулятор с фиксированной уставкой и переключающим выходом для пропорциональных исполнительных устройств. Следующий рисунок показывает поток сигналов в контуре регулирования.

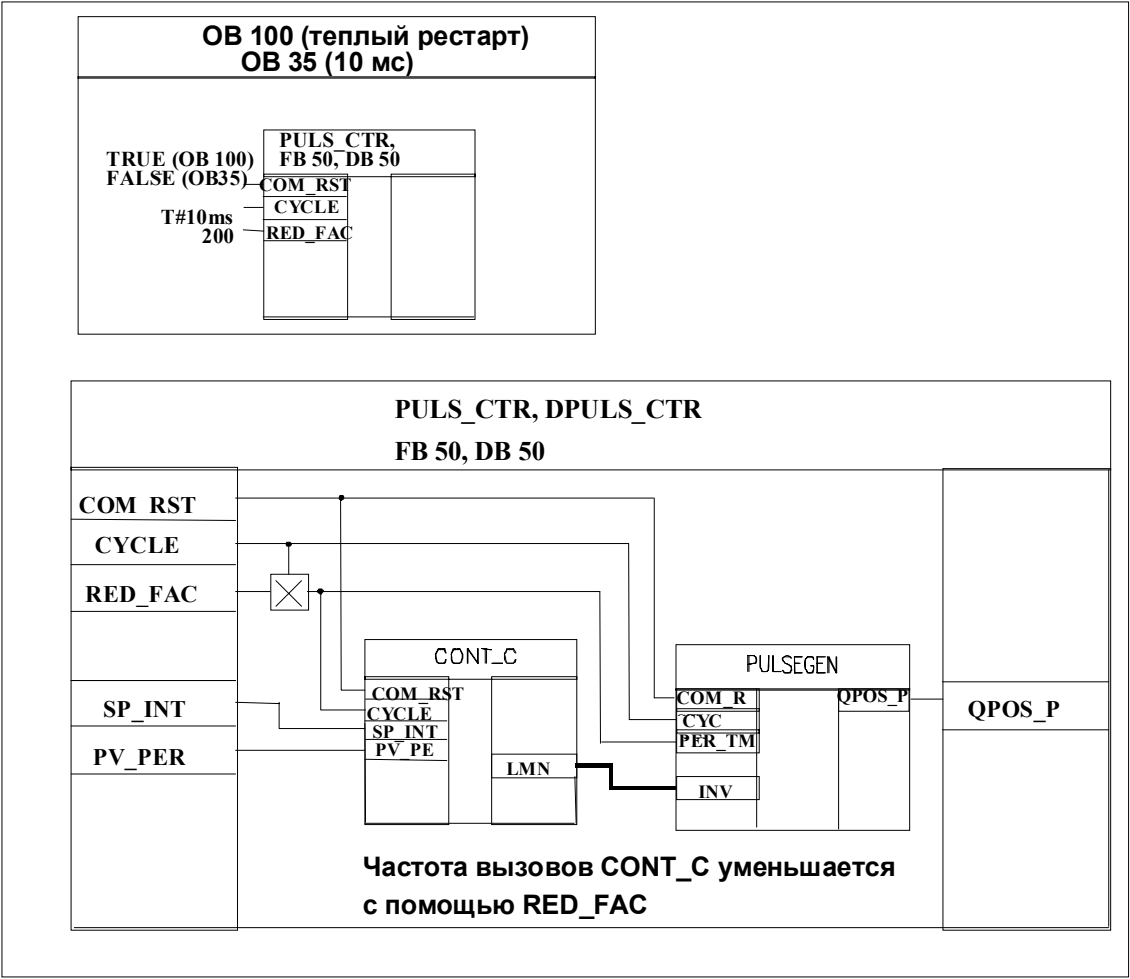


Регулятор непрерывного действия CONT_C формирует управляющее воздействие LMN, которое преобразуется импульсным генератором PULSEGEN в импульсно-прерывистый сигнал QPOS_P или QNEG_P.

Вызов блока и его подключение

Регулятор с фиксированной уставкой и переключающим выходом для пропорциональных исполнительных устройств PULS_CTR состоит из блоков CONT_C и PULSEGEN. Вызов блока осуществляется так, чтобы CONT_C вызывался каждые 2 секунды ($=\text{CYCLE} \cdot \text{RED_FAC}$), а PULSEGEN каждые 10 мс ($=\text{CYCLE}$). Время цикла OB35 устанавливается равным 10 мс. Взаимосвязь можно видеть на следующем рисунке.

Во время теплого рестарта в OB100 вызывается блок PULS_CTR, и вход COM_RST устанавливается в TRUE.



Программа на STL для FB PULS_CTR

Адрес	Описание	Имя	Тип	Комментарий
0.0	in	SP_INT	REAL	Уставка
4.0	in	PV_PER	WORD	Периферийное фактическое значение
6.0	in	RED_FAC	INT	Коэффициент уменьшения частоты вызовов
8.0	in	COM_RST	BOOL	Полный рестарт
10.0	in	CYCLE	TIME	Время опроса
14.0	out	QPOS_P	BOOL	Управляющий сигнал
16.0	stat	DI_CONT_C	FB-CONT_C	Счетчик
142.0	stat	DI_PULSEGEN	FB-PULSEGEN	Счетчик
176.0	stat	sCount	INT	Счетчик
0.0	temp	tCycCtr	TIME	Время опроса регулятора

STL	Описание
A #COM_RST	// Подпрограмма инициализации
JCN M001	
L 0	
T #sCount	
M001: L #CYCLE	//Вычисление цикла опроса регулятора
L #RED_FAC	
*D	
T #tCycCtr	
L #sCount	// Уменьшение значения счетчика и сравнение с нулем
L 1	
-I	
T #sCount	
L 0	
<=I	
JCN M002	//Условный вызов блока и установка счетчика
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	

23 SFC для H CPU

1.1 Управление функционированием в H-системах с помощью SFC90 "H_CTRL"

Описание

С помощью SFC90 "H_CTRL" вы можете влиять на H-системы следующим образом:

- Вы можете предотвращать подключение резерва в главном CPU. Тогда оно блокируется до тех пор, пока вы не отмените эту установку с помощью SFC90 "H_CTRL".

Любой запрос из резервного CPU на соединение с главным сохраняется.

- Вы можете отключить обновление в главном CPU. Тогда оно блокируется до тех пор, пока вы не отмените эту установку с помощью SFC90 "H_CTRL".

Любой запрос из резервного CPU на обновление сохраняется.

Следующая таблица объясняет разрешенные комбинации входных параметров MODE и SUBMODE.

Задание	Вход MODE	Вход SUBMODE
Запретить соединение	3	не имеет значения
Вновь разрешить соединение	4	не имеет значения
Запретить обновление	1	не имеет значения
Вновь разрешить обновление	2	не имеет значения

Как работает SFC

SFC90 "H_CTRL" является асинхронным SFC, иными словами, его выполнение может растягиваться на время нескольких вызовов SFC.

Задание запускают, вызывая SFC90 с REQ=1.

Если задание можно было выполнить сразу, то SFC возвращает в выходном параметре BUSY значение 0. Если BUSY имеет значение 1, то задание все еще активно (см. также раздел 2.2).

Идентификация задания

Входные параметры MODE и SUBMODE определяют задание. Если они соответствуют заданию, которое еще не закончено, то вызов SFC является продолженным вызовом.

Параметры

Параметр	Описание	Тип данных	Область памяти	Характеристика
REQ	INPUT	BOOL	I, Q, M, D, L,	Запускаемый уровнем параметр управления REQ=1: запускает задание.
MODE	INPUT	BYTE	I, Q, M, D, L, константа	Задание.
SUBMODE	INPUT	BYTE	I, Q, M, D, L, константа	Подзадание.
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Если во время выполнения функции происходит ошибка, то возвращаемое значение содержит код ошибки. Убедитесь, что вы оцениваете RET_VAL каждый раз, когда блок выполнен.
BUSY	OUTPUT	BOOL	I, Q, M, D, L	BUSY=1: Задание еще не закончено.

Информация об ошибках

Код ошибки (W#16#...)	Объяснение
0000	Задание выполнено без ошибки.
7000	REQ = 0 при первом вызове: задание не было активизировано; BUSY имеет значение 0.
7001	REQ = 1 при первом вызове: задание было запущено; BUSY имеет значение 1.
7002	Продолженный вызов (REQ не имеет значения). Активизированное задание все еще выполняется; BUSY имеет значение 1.
0001	<ul style="list-style-type: none"> Когда MODE=1: обновление уже было заблокировано. Когда MODE=3: соединение уже было заблокировано.
8082	<ul style="list-style-type: none"> Когда MODE=1: обновление уже активно и больше не может блокироваться. Когда MODE=3: соединение уже активно и больше не может блокироваться.
8090	Входной параметр MODE имеет недопустимое значение.

Пример использования SFC 90

С помощью SFC90 "H_CTRL" вы можете добиться того, чтобы соединение и обновление не начинались в моменты времени, когда требуются максимальные ресурсы CPU.

Вы можете достичь этого включением следующих разделов в главном CPU перед периодом повышенной активности процесса:

- вызов SFC 90 с MODE = 3 и SUBMODE = 0 (запретить соединение)
- вызов SFC 90 с MODE = 1 и SUBMODE = 0 (запретить обновление).

В конце периода повышенной активности включите следующие разделы программы в главном CPU:

- вызов SFC 90 с MODE = 4 и SUBMODE = 0 (вновь разрешить соединение)
- вызов SFC 90 с MODE = 2 и SUBMODE = 0 (вновь разрешить обновление).

24 Встроенные функции (для CPU с встроенными входами/выходами)

24.1 SFB29 (HS_COUNT)

Описание

С помощью SFB29 "HS_COUNT" (счетчик) вы можете воздействовать на встроенную функцию счетчика CPU со встроенными входами/выходами следующим образом:

- устанавливать и вводить начальное значение
- выбирать и устанавливать значения для сравнения
- разблокировать счетчики
- разблокировать цифровые выходы
- считывать текущие счетные значения и текущие значения для сравнения
- запрашивать соотношение между счетным значением и значением для сравнения.

Дополнительная информация

Значение отдельных параметров SFB29 вместе с параметрами встроенной функции счетчика и аппаратных входов и выходов CPU подробно описано в руководстве *S7-300 Programmable Controller, Integrated Functions* [Программируемый контроллер S7-300, встроенные функции].

Параметры

Параметр	Описание	Тип данных	Область памяти	Характеристика
PRES_COUNT	INPUT	DINT	I, Q, M, D, L, константа	Начальное значение счетчика
PRES_COMP_A	INPUT	DINT	I, Q, M, D, L, константа	Новое значение для сравнения COMP_A
PRES_COMP_B	INPUT	DINT	I, Q, M, D, L, константа	Новое значение для сравнения COMP_B
EN_COUNT	INPUT		I, Q, M, D, L	Разблокировка счетчика
EN_DO	INPUT	BOOL	I, Q, M, D, L, константа	Разблокировка цифровых выходов
SET_COUNT	INPUT	BOOL	I, Q, M, D, L, константа	Вход установки начального значения PRES_COUNT
SET_COMP_A	INPUT	BOOL	I, Q, M, D, L, константа	Вход установки значения для сравнения COMP_A
SET_COMP_B	INPUT	BOOL	I, Q, M, D, L, константа	Вход установки значения для сравнения COMP_B
COUNT	OUTPUT	DINT	I, Q, M, D, L	Фактическое значение счетчика
COMP_A	OUTPUT	DINT	I, Q, M, D, L	Текущее значение для сравнения COMP_A
COMP_B	OUTPUT	DINT	I, Q, M, D, L	Текущее значение для сравнения COMP_B
STATUS_A	OUTPUT	BOOL	I, Q, M, D, L	Бит состояния STATUS_A 1: COUNT ≥ COMP_A 0: COUNT < COMP_A
STATUS_B	OUTPUT	BOOL	I, Q, M, D, L	Бит состояния STATUS_B 1: COUNT ≥ COMP_B 0: COUNT < COMP_B

24.2 SFB30 (FREQ_MES)

Описание

С помощью SFB30 "FREQ_MES" (частотомер) вы можете воздействовать на встроенную функцию частотомера CPU со встроенными входами/выходами следующим образом:

- выбирать и устанавливать значения для сравнения
- выводить измеренную частоту
- считывать текущие значения для сравнения
- запрашивать отношение измеренной частоты к значению для сравнения.

Дополнительная информация

Значение отдельных параметров SFB30 вместе с параметрами встроенной функции частотомера и аппаратных входов и выходов CPU подробно описано в руководстве *S7-300 Programmable Controller, Integrated Functions* [Программируемый контроллер S7-300, встроенные функции].

Параметры

Параметр	Описание	Тип данных	Область памяти	Характеристика
PRES_U_LIMIT	INPUT	DINT	I, Q, M, D, L, константа	Новое (верхнее) значение для сравнения U_LIMIT
PRES_L_LIMIT	INPUT	DINT	I, Q, M, D, L, константа	Новое (нижнее) значение для сравнения L_LIMIT
SET_U_LIMIT	INPUT	BOOL	I, Q, M, D, L, константа	Вход установки нового значения для сравнения U_LIMIT
SET_L_LIMIT	INPUT	BOOL	I, Q, M, D, L, константа	Вход установки нового значения для сравнения L_LIMIT
FREQ	OUTPUT	DINT	I, Q, M, D, L	Измеренная частота в МГц
U_LIMIT	OUTPUT	DINT	I, Q, M, D, L	Текущее значение для сравнения (верхний предел)
L_LIMIT	OUTPUT	DINT	I, Q, M, D, L	Текущее значение для сравнения (нижний предел)
STATUS_U	OUTPUT	BOOL	I, Q, M, D, L	Бит состояния "1": FREQ > U_LIMIT "0": FREQ ≤ U_LIMIT
STATUS_L	OUTPUT	BOOL	I, Q, M, D, L	Бит состояния "1": FREQ < L_LIMIT "0": FREQ ≥ U_LIMIT

24.3 SFB38 (HSC_A_B)

Описание

С помощью SFB38 (HSC_A_B) вы можете воздействовать на встроенную функцию счетчика A/B в CPU со встроенными входами/выходами, следующим образом:

- задавать и принимать начальное значение
- задавать и устанавливать значения для сравнения
- разблокировать счетчики
- разблокировать цифровые выходы
- считывать текущие счетные значения и текущие значения для сравнения.
- запрашивать соотношение между счетным значением и значением для сравнения

SFB38 (HSC_A_B) читает или записывает данные из программы пользователя в экземплярном DB встроенной функции. Счетчик A/B состоит из двух счетчиков A и B, которые могут вести счет одновременно и независимы друг от друга (возможен прямой и обратный счет). Счетчики функционируют одинаково; могут регистрироваться счетные импульсы частотой до 10 кГц.

Дополнительная информация

Точное значение параметров SFB38 вместе с параметрами встроенной функции счетчика A/B и аппаратных входов и выходов CPU подробно описано в руководстве *S7-300 Programmable Controller, Integrated Functions CPU 312 IFM/314 IFM* [Программируемый контроллер S7-300, встроенные функции CPU 312 IFM/314 IFM].

Параметры

Параметр	Описание	Тип данных	Область памяти	Характеристика
PRES_COMP	INPUT	DINT	I, Q, M, D, L, константа	Новое значение для сравнения COMP
EN_COUNT	INPUT	BOOL	I, Q, M, D, L константа	Разблокировка счетчика
EN	INPUT	BOOL	I, Q, M, D, L, константа	Разблокировка цифрового выхода
SET_COMP	INPUT	BOOL	I, Q, M, D, L, константа	Вход установки значения для сравнения COMP
COUNT	OUTPUT	DINT	I, Q, M, D, L	Фактическое значение счетчика
COMP	OUTPUT	DINT	I, Q, M, D, L	Текущее значение для сравнения COMP
ENO	OUTPUT	BOOL	I, Q, M, D, L	Обработка ошибок: 1 : нет ошибки при выполнении 0 : ошибка при выполнении

24.4 SFB39 (POS)

Описание

С помощью SFB39 (POS) вы можете воздействовать на встроенную функцию позиционирования CPU со встроенными входами/выходами. SFB39 (POS) предоставляет в распоряжение следующие функции:

- Синхронизация
- Выполнение замедленного режима
- Позиционирование

SFB39 (POS) для встроенной функции позиционирования читает или записывает данные из программы пользователя в экземплярный DB встроенной функции. Встроенная функция позиционирования собирает сигналы от асимметричных инкрементальных шифраторов с напряжением 24 В с частотой до 10 кГц. Она управляет режимом «быстро/медленно» или преобразователем частоты через заданные встроенные выходы CPU 314 IFM (управляемое позиционирование).

Дополнительная информация

Точное значение параметров SFB39 вместе с параметрами встроенной функции счетчика A/B и аппаратных входов и выходов CPU подробно описано в руководстве *S7-300 Programmable Controller, Integrated Functions CPU 312 IFM/314 IFM* [Программируемый контроллер S7-300, встроенные функции CPU 312 IFM/314 IFM].

Параметры

Параметр	Описание	Тип данных	Область памяти	Характеристика
EN	INPUT	BOOL	I, Q, M, D, L, константа	Разблокировка цифровых выходов
DEST_VAL	INPUT	DINT	I, Q, M, D, L, константа	Целевая позиция для интегрированной функции позиционирования
REF_VAL	INPUT	DINT	I, Q, M, D, L, константа	Опорная точка синхронизации
SWITCH_OFF_DIFF	INPUT	WORD	I, Q, M, D, L, константа	Упреждение выключения (разность между точкой выключения и целевой позицией) в шагах перемещения
PRES_COMP	INPUT	DINT	I, Q, M, D, L, константа	Новое значение для сравнения COMP
BREAK	INPUT	BYTE	I, Q, M, D, L, константа	Максимальное аналоговое значение, которым регулируется поперечное движение

POS_MODE1, POS_MODE2	INPUT	BOOL	I, Q, M, D, L, константа	Запуск и выполнение замедленного режима
POS_STRT	INPUT	BOOL	I, Q, M, D, L, константа	Запуск операции позиционирования при нарастающем фронте
SET_POS	INPUT	BOOL	I, Q, M, D, L, константа	При нарастающем фронте значение входного параметра REF_VAL принимается как новое фактическое значение
ENO	OUTPUT	BOOL	I, Q, M, D, L	Обработка ошибок: 1 : нет ошибки при выполнении 0 : ошибка при выполнении
ACTUAL_POS	OUTPUT	DINT	I, Q, M, D, L	Текущее фактическое значение
POS_READY (сообщение о состоянии)	OUTPUT	BOOL	I, Q, M, D, L	Позиционирование/ замедленное перемещение завершено, если POS_READY=1
REF_VALID (сообщение о состоянии)	OUTPUT	BOOL	I, Q, M, D, L	Достигнут или нет переключатель опорной точки
POS_VALID (сообщение о состоянии)	OUTPUT	BOOL	I, Q, M, D, L	Фактическое положение оси синхронизировано с фактическим положением встроенной функции

25 Гибкая технология

25.1 SFC63 (AB_CALL)

Описание

SFC63 (AB_CALL) вызывает блок скомпонованного кода.

Блоки скомпонованного кода – это логические блоки, которые были написаны на языке программирования С или на Ассемблере, а затем скомпилированы.

Применение

Вы можете использовать блоки скомпонованного кода только для CPU 614.

Дополнительная информация

Значение отдельных параметров SFC63 подробно объясняется в документации для CPU 614. Имеется отдельное руководство по программированию блоков скомпонованного кода.

Параметры

Параметр	Описание	Тип данных	Область памяти	Характеристика
AB_NUMBER	INPUT	WORD	I, Q, M, D, L, константа	Биты для вызываемых блоков скомпонованного кода
CALL_REASON	INPUT	WORD	I, Q, M, D, L, константа	Организационный блок, в котором вызывался SFC, или оценка указателя DB (параметр DB_NUMBER), или активизация отладчика
DB_NUMBER	INPUT	WORD	I, Q, M, D, L, константа	Номер указателя DB
RET_VAL	OUTPUT	WORD	I, Q, M, D, L	Возвращаемое значение SFC

26 Диагностические данные

26.1 Обзор структуры диагностических данных

Записи данных 0 и 1 области системных данных

Диагностические данные модуля находятся в записях данных 0 и 1 области системных данных (см. раздел 7.1).

- Запись данных 0 содержит 4 байта диагностических данных, которые описывают текущее состояние сигнального модуля.
- Запись данных 1 содержит
 - 4 байта диагностических данных, которые находятся также в записи данных 0, и
 - диагностические данные, специфические для модуля.

Структура и содержание диагностических данных

Этот раздел описывает структуру и содержание отдельных байтов диагностических данных.

Если появляется ошибка, то соответствующий бит устанавливается в "1".

26.2 Диагностические данные

Структура и содержание диагностических данных:

Байт	Бит	Значение	Примечание
0	0	Неисправность модуля	
	1	Внутренняя ошибка	
	2	Внешняя ошибка	
	3	Ошибка канала	
	4	Отсутствует внешнее вспомогательное напряжение	
	5	Отсутствует передний штепсельный разъем	
	6	Отсутствует параметризация	
	7	Неправильные параметры в модуле	
1	от 0 до 3	Класс модуля	0101 Аналоговый модуль
			0000 CPU
			1000 Функциональный модуль
			1100 CP
			1111 Цифровой модуль
			0011 Стандартное slave-устройство DP
			1011 I-slave
			0100 IM
	4	Имеется информация канала	
	5	Имеется информация пользователя	
	6	Диагностическое прерывание из-за замены	
	7	0	
2	0	Модуль памяти неисправен или отсутствует	
	1	Нарушение связи	
	2	Рабочий режим	0 RUN 1 STOP
	3	Сработал контроль времени цикла	
	4	Отказ внутреннего напряжения питания модуля	
	5	Разрядилась батарея	
	6	Отказ всей батарейной буферизации	
	7	0	
3	0	Отказ стойки расширения	
	1	Отказ процессора	
	2	Ошибка EPROM	
	3	Ошибка RAM	
	4	Ошибка АЦП/ЦАП	
	5	Сгорел предохранитель	
	6	Потеря аппаратного прерывания	
	7	0	

Байт	Бит	Значение	Примечание	
4	от 0 до 6	Тип канала	B#16#70 Цифровой ввод B#16#72 Цифровой вывод B#16#71 Аналоговый ввод B#16#73 Аналоговый вывод B#16#74 FM-POS B#16#75 FM-REG B#16#76 FM-ZAENL B#16#77 FM-TECHNO B#16#78 FM-NCU от B#16#79 до B#16#7D резерв B#16#7E US300 B#16#7F резерв	
	7	Есть дополнительный канал?	0 нет 1 да	
5	от 0 до 7	Количество битов диагностики, выводимых модулем на один канал.	Количество битов диагностики на один канал округляется до границ байта	
6	от 0 до 7	Количество каналов одного типа на модуль	Если в модуле существуют разные типы каналов, то структура для каждого типа канала повторяется, начиная с байта 4 в записи данных 1.	
7	0	Ошибка канала - канал 0/ группа каналов 0	Первый байт вектора ошибок канала (длина вектора ошибок канала определяется числом каналов и округляется до границ байта).	
	1	Ошибка канала - канал 1/ группа каналов 1		
	2	Ошибка канала - канал 2/ группа каналов 2		
	3	Ошибка канала - канал 3/ группа каналов 3		
	4	Ошибка канала - канал 4/ группа каналов 4		
	5	Ошибка канала - канал 5/ группа каналов 5		
	6	Ошибка канала - канал 6/ группа каналов 6		
	7	Ошибка канала - канал 7/ группа каналов 7		
...	-	Специфические для канала ошибки (см. раздел Ошибка! Источник ссылки не найден.)		

26.3 Структура диагностических данных, относящихся к каналу

Ошибки, относящиеся к каналу

Начиная с байта, который находится непосредственно за вектором ошибок каналов, для каждого канала модуля отображаются относящиеся к этому каналу ошибки. Следующая таблица показывает структуру относящихся к каналу диагностических данных для разных типов каналов. Биты имеют следующие значения:

- 1 = ошибка
- 0 = нет ошибки

Канал ввода аналоговых сигналов

Диагностический байт канала аналогового ввода:

Бит	Значение	Примечание
0	Ошибка проектирования/ параметризации	Может быть передана функцией SFC52 и EVENTN = W#16#8x50
1	Синфазная ошибка	Может быть передана функцией SFC52 и EVENTN = W#16#8x51
2	Короткое замыкание на Р	Может быть передана функцией SFC52 и EVENTN = W#16#8x52
3	Короткое замыкание на М	Может быть передана функцией SFC52 и EVENTN = W#16#8x53
4	Обрыв провода	Может быть передана функцией SFC52 и EVENTN = W#16#8x54
5	Ошибка опорного канала	Может быть передана функцией SFC52 и EVENTN = W#16#8x55
6	Ток ниже границы диапазона измерения	Может быть передана функцией SFC52 и EVENTN = W#16#8x56
7	Ток выше границы диапазона измерения	Может быть передана функцией SFC52 и EVENTN = W#16#8x57

Канал аналогового вывода

Диагностический байт для канала аналогового вывода:

Бит	Значение	Примечание
0	Ошибка проектирования/ параметризации	Может быть передана функцией SFC52 и EVENTN = W#16#8x60
1	Синфазная ошибка	Может быть передана функцией SFC52 и EVENTN = W#16#8x61
2	Короткое замыкание на Р	Может быть передана функцией SFC52 и EVENTN = W#16#8x62
3	Короткое замыкание на М	Может быть передана функцией SFC52 и EVENTN = W#16#8x63
4	Обрыв провода	Может быть передана функцией SFC52 и EVENTN = W#16#8x64
5	0	резерв
6	Отсутствует напряжение нагрузки	Может быть передана функцией SFC52 и EVENTN = W#16#8x66
7	0	резерв

Канал цифрового ввода

Диагностический байт для канала цифрового ввода:

Бит	Значение	Примечание
0	Ошибка проектирования/ параметризации	Может быть передана функцией SFC52 и EVENTN = W#16#8x70
1	Ошибка заземления на корпус	Может быть передана функцией SFC52 и EVENTN = W#16#8x71
2	Короткое замыкание на Р (датчик)	Может быть передана функцией SFC52 и EVENTN = W#16#8x72
3	Короткое замыкание на М	Может быть передана функцией SFC52 и EVENTN = W#16#8x73
4	Обрыв провода	Может быть передана функцией SFC52 и EVENTN = W#16#8x74
5	Отсутствует питание датчика	Может быть передана функцией SFC52 и EVENTN = W#16#8x75
6	0	резерв
7	0	резерв

Канал цифрового вывода

Диагностический байт для канала цифрового вывода:

Бит	Значение	Примечание
0	Ошибка проектирования/ параметризации	Может быть передана функцией SFC52 и EVENTN = W#16#8x80
1	Ошибка заземления на корпус	Может быть передана функцией SFC52 и EVENTN = W#16#8x81
2	Короткое замыкание на Р	Может быть передана функцией SFC52 и EVENTN = W#16#8x82
3	Короткое замыкание на М	Может быть передана функцией SFC52 и EVENTN = W#16#8x83
4	Обрыв провода	Может быть передана функцией SFC52 и EVENTN = W#16#8x84
5	Сгорел предохранитель	Может быть передана функцией SFC52 и EVENTN = W#16#8x86
6	Отсутствует напряжение нагрузки	Может быть передана функцией SFC52 и EVENTN = W#16#8x86
7	Перегрев	Может быть передана функцией SFC52 и EVENTN = W#16#8x87

27 Списки состояний системы (SSL)

27.1 Обзор списков состояний системы (SSL)

Обзор этой главы

Эта глава описывает все подписки списка состояний системы, которые относятся:

- к CPU
- к модулям, подписки которых не являются специфическими для модулей (например, SSL-ID W#16#00B1, W#16#00B2, W#16#00B3).

Специфические для модулей подписки, в частности, для CP и FM включены в описания конкретных модулей.

Определение: Список состояний системы

Список состояний системы (system status list, SSL) описывает текущее состояние программируемого логического контроллера. Содержание SSL может быть только прочитано с помощью информационных функций, но не изменено. Подписки являются виртуальными списками, то есть они создаются операционной системой центральных модулей только конкретному по запросу.

С помощью SFC 51 "RDSYSST" вы можете прочитать только один список состояний системы.

Содержание

Список состояний системы содержит информацию о:

- системных данных
- данных состояния диагностики в CPU
- диагностических данных модулей
- диагностическом буфере

Системные данные

Системные данные представляют собой фиксированные или назначенные характеристики CPU. Они предоставляют информацию:

- о конфигурации CPU
- о состоянии классов приоритета
- о связи

Данные состояния диагностики

Данные состояния диагностики описывают текущее состояние компонентов, контролируемых системными диагностическими функциями.

Диагностические данные модулей

Поставленные в соответствие CPU модули с диагностическими способностями обладают диагностическими данными, которые хранятся непосредственно в модуле.

Диагностический буфер

Диагностический буфер содержит диагностические записи в порядке их появления.

27.2 Структура подписка SSL

Основы

Вы можете прочитать подписок или фрагменты подписка с помощью SFC51 "RDSYSST". Что вы хотите прочитать, вы указываете с помощью параметров SSL_ID и INDEX.

Структура

Подписок состоит из:

- заголовка и
- записей данных.

Заголовок

Заголовок подписка состоит из:

- SSL-ID (идентификатора SSL)
- индекса
- длины записи данных этого подписка в байтах
- количества записей данных, содержащихся в этом подписке.

Индекс

С конкретными подписками или фрагментами подписков должен быть задан идентификатор типа объекта или номер объекта. Для этого используется индекс. Если он не нужен для информации, то его содержание не имеет значения.

Записи данных

Запись данных подписка имеет определенную длину. Она зависит от информации, содержащейся в подписке. Как используются слова данных в записи данных, также зависит от конкретного подписка.

27.3 SSL-ID

SSL-ID

Каждый подписок состояний системы имеет номер. Вы можете вывести полный подписок или фрагмент из него. Возможные фрагменты подписков predeterminedены и идентифицируются с помощью номера. SSL-ID (идентификатор SSL) состоит из номера подписка, номера фрагмента подписка и класса модуля.

Структура

Длина SSL-ID равна одному слову. Биты SSL-ID имеют следующие значения:



Структура *SSL_ID*

Класс модуля

Примеры классов модулей:

Класс модуля	Код (двоичный)
CPU	0000
IM	0100
FM	1000
CP	1100

Номер фрагмента подписка

Номера фрагментов подписка и их значение зависят от конкретного списка состояний системы, которому они принадлежат. С помощью номера фрагмента подписка вы задаете, какое подмножество подписка вы хотите прочитать.

Номер подписка

С помощью номера подписка вы задаете, какой подписок SSL вы хотите прочитать.

27.4 Возможные подписки состояний системы

Подмножество

Любой модуль содержит только подмножество всех возможных подписков.
Какие подписки имеются в распоряжении, зависит от конкретного модуля.

Возможные подписки SSL

В следующей таблице представлены все возможные подписки с соответствующими номерами в SSL-ID.

Подписок	SSL-ID
Идентификация модуля	W#16#xy11
Характеристики CPU	W#16#xy12
Области памяти пользователя	W#16#xy13
Системные области	W#16#xy14
Типы блоков	W#16#xy15
Состояние светодиодов модуля	W#16#xy19
Состояние прерываний	W#16#xy22
Данные о состоянии связи	W#16#xy32
Групповая информация H CPU	W#16#xy71
Состояние светодиодов модуля	W#16#xy74
Информация о состоянии модуля	W#16#xy91
Информация о состоянии стойки/ станции	W#16#xy92
Диагностический буфер CPU	W#16#xyA0
Диагностическая информация модуля (запись данных 0)	W#16#00B1
Диагностическая информация модуля (запись данных 1), географический адрес	W#16#00B2
Диагностическая информация модуля (запись данных 1), логический адрес	W#16#00B3
Диагностические данные slave-устройства DP	W#16#00B4

27.5 SSL-ID W#16#xy11 – Идентификация модуля

Цель

Если вы читаете список состояний системы с SSL-ID W#16#xy11, то вы получаете идентификацию этого модуля.

Заголовок

Заголовок списка состояний системы с SSL-ID W#16#xy11 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID фрагмента подписка W#16#0111: отдельная идентификационная запись данных
INDEX	Номер конкретной записи данных W#16#0001: идентификация модуля W#16#0006: идентификация базового аппаратного обеспечения W#16#0007: идентификация базового микропрограммного обеспечения
LENGTHDR	W#16#001C: одна запись данных имеет длину 14 слов (28 байтов)
N_DR	Количество записей данных

Запись данных

Запись данных списка состояний системы с SSL-ID W#16#xy11 имеет следующую структуру:

Имя	Длина	Значение
Index	1 слово	Индекс идентификационной записи данных
MlfB	20 байтов	Заказной номер модуля: строка из 19 символов и пробела (20H); например, для CPU 314: "6ES7 314-0AE01-0AB0 "
BGTyp	1 слово	резерв
Ausbg	1 слово	С индексом W#16#0001: версия модуля С индексом W#16#0006 и W#16#0007: "V" и первая цифра идентификатора версии
Ausbe	1 слово	С индексом W#16#0001: резерв С индексом W#16#0006 и W#16#0007: остальные цифры идентификатора версии

27.6 SSL-ID W#16#xy12 – Характеристики CPU

Цель

Модули CPU обладают различными характеристиками в зависимости от используемого аппаратного обеспечения. Каждой характеристике поставлен в соответствие идентификатор. Если вы читаете подписание с SSL-ID W#16#xy12, то вы получаете характеристики модуля.

Заголовок

Заголовок подписка SSL-ID W#16#xy12 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID фрагмента подписки: W#16#0012: все характеристики W#16#0112: характеристики группы Группу вы задаете в параметре INDEX. W#16#0F12: информация заголовка подписки
INDEX	Группа W#16#0000: блок обработки MC7 W#16#0100: система времени W#16#0200: реакция системы W#16#0300: описание языка CPU
LENGTHDR	W#16#0002: одна запись данных имеет длину 1 слово (2 байта)
N_DR	Количество записей данных

Запись данных

Запись данных подписка с SSL-ID W#16#xy12 имеет длину одно слово. Идентификатор введен для каждой характеристики. Идентификатор характеристики имеет длину одно слово.

Идентификатор характеристики

Следующая таблица перечисляет все идентификаторы характеристик.

Идентификатор	Значение
W#16#0000 - 00FF	Блок обработки MC7 (группа с индексом 0000)
W#16#0001	Обработка MC7, генерирующая код
W#16#0002	Интерпретатор MC7
W#16#0100 - 01FF	Система времени (группа с индексом 0100)
W#16#0101	Разрешающая способность 1 мс
W#16#0102	Разрешающая способность 10 мс
W#16#0103	Нет часов реального времени
W#16#0104	BCD-формат времени суток
W#16#0200 - 02FF	Реакция системы (группа с индексом 0200)
W#16#0201	Способность к многопроцессорной обработке
W#16#0300 - 03FF	Описание языка MC7 CPU (группа с индексом 0300)
W#16#0301	резерв
W#16#0302	Все 32-битовые команды вычислений с фиксированной точкой
W#16#0303	Все команды вычислений с плавающей точкой
W#16#0304	sin, asin, cos, acos, tan, atan, sqr, sqrt, ln, exp
W#16#0305	Аккумулятор 3/аккумулятор 4 с соответствующими командами

	(ENT,PUSH,POP,LEAVE)
W#16#0306	Команды главного управляющего реле (Master Control Relay)
W#16#0307	Имеется адресный регистр 1 с соответствующими командами
W#16#0308	Имеется адресный регистр 2 с соответствующими командами
W#16#0309	Операции для адресации, выходящей за пределы области
W#16#030A	Операции для адресации внутри области
W#16#030B	Все команды косвенной адресации через память для памяти с побитовым доступом (меркеры) (M)
W#16#030C	Все команды косвенной адресации через память для блоков данных (DB)
W#16#030D	Все команды косвенной адресации через память для блоков данных (DI)
W#16#030E	Все команды косвенной адресации через память для локальных данных (L)
W#16#030F	Все команды передачи параметров в FC
W#16#0310	Команды, связанные с фронтом бита памяти (меркера), для входа образа процесса (I)
W#16#0311	Команды, связанные с фронтом бита памяти, для выхода образа процесса (Q)
W#16#0312	Команды, связанные с фронтом бита памяти, для бита памяти (M)
W#16#0313	Команды, связанные с фронтом бита памяти, для блоков данных (DB)
W#16#0314	Команды, связанные с фронтом бита памяти, для блоков данных (DI)
W#16#0315	Команды, связанные с фронтом бита памяти, для локальных данных (L)
W#16#0316	Динамическая оценка бита FC
W#16#0317	Динамическая область локальных данных с соответствующими командами
W#16#0318	Резерв
W#16#0319	Резерв

27.7 SSL-ID W#16#xy13 – Области памяти

Цель

Считывая подписание с SSL-ID W#16#xy13, вы получаете информацию об областях памяти модуля.

Заголовок

Заголовок подписка SSL-ID W#16#xy13 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID фрагмента подписка W#16#0113: запись данных для одной области памяти Вы задаете область памяти через параметр INDEX.
INDEX	Задаёт область памяти (только с SSL-ID W#16#0113) W#16#0001: рабочая память
LENGTHDR	W#16#0024: одна запись данных имеет длину 18 слов (36 байтов)
N_DR	Количество записей данных

Запись данных

Запись данных подписка с SSL-ID W#16#xy13 имеет следующую структуру:

Имя	Длина	Значение
Index	1 слово	Индекс области памяти W#16#0001: рабочая память
Code	1 слово	Тип памяти: W#16#0001: энергозависимая память (RAM) W#16#0002: энергонезависимая память (FEPR0M) W#16#0003: смешанная память (RAM + FEPR0M)
Size	2 слова	Общий размер выбранной памяти (сумма области 1 и области 2)
Mode	1 слово	Логический тип памяти Бит 0: энергозависимая область памяти Бит 1: энергонезависимая область памяти Бит 2: смешанная область памяти Для рабочей памяти: Бит 3: код и данные раздельно Бит 4: код и данные раздельно
Granu	1 слово	Всегда имеет значение 0
Ber1	2 слова	Размер энергозависимой области памяти в байтах.
Belegt1	2 слова	Размер используемой энергозависимой области памяти в байтах
Block1	2 слова	Наибольший свободный блок в энергозависимой области памяти Если 0: информация отсутствует или не может быть найдена.
Ber2	2 слова	Размер энергонезависимой области памяти в байтах
Belegt2	2 слова	Размер занятой энергонезависимой области памяти
Block2	2 слова	Наибольший свободный блок в энергонезависимой области памяти Если 0: информация отсутствует или не может быть найдена.

27.8 SSL-ID W#16#xy14 – Системные области

Цель

Считывая подписание с SSL-ID W#16#xy14, вы получаете информацию о системных областях модуля.

Заголовок

Заголовок подписка с SSL-ID W#16#xy14 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID фрагмента подписка W#16#0014: все системные области модуля W#16#0F14: только для информации заголовка подписка
INDEX	Не имеет значения
LENGTHDR	W#16#0008: на запись данных имеет длину 4 слова (8 байтов)
N_DR	Количество записей данных

Запись данных

Запись данных подписка с SSL-ID W#16#xy14 имеет следующую структуру:

Имя	Длина	Значение
Index	1 слово	Индекс системной области W#16#0001: PII (количество в байтах) W#16#0002: PIQ (количество в байтах) W#16#0003: память (меркеры) (количество в битах) W#16#0004: таймеры (количество) W#16#0005: счетчики (количество) W#16#0006: количество байтов в логическом адресном пространстве W#16#0007: локальные данные (вся область локальных данных CPU в байтах) W#16#0008: память (меркеры) (количество в байтах) W#16#0009: локальные данные (вся область локальных данных CPU в килобайтах)
code	1 слово	Тип памяти W#16#0001: энергозависимая память (RAM) W#16#0002: энергонезависимая память (FEPROM) W#16#0003: смешанная память (RAM и FEPROM)
anzahl	1 слово	Количество элементов системной области
reman	1 слово	Количество сохраняемых (реманентных) элементов

27.9 SSL-ID W#16#ху15 – Типы блоков

Цель

Считывая подписание с SSL-ID W#16#ху15, вы получаете типы блоков, которые имеются в модуле.

Заголовок

Заголовок подписка SSL-ID W#16#ху15 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID фрагмента подписка W#16#0015: Записи данных всех типов блоков модуля
INDEX	Не имеет значения
LENGTHDR	W#16#0006: одна запись данных имеет длину 5 слов (10 байтов)
N_DR	Количество записей данных

Запись данных

Запись данных подписка с SSL-ID W#16#ху15 имеет следующую структуру:

Имя	Длина	Значение
Index	1 слово	Номер типа блока W#16#0800: OB W#16#0A00: DB W#16#0B00: SDB W#16#0C00: FC W#16#0E00: FB
MaxAnz	1 слово	Максимальное количество блоков этого типа OB: максимально возможное количество OB в CPU DB: максимально возможное количество DB, включая DB0 SDB: максимально возможное количество SDB, включая SDB2 FC и FB: максимально возможное количество загружаемых блоков
MaxLng	1 слово	Максимальный общий размер загружаемых объектов в Кбайтах
Maxabl	2 слова	Максимальная длина раздела рабочей памяти блока в байтах

27.10 SSL-ID W#16#xy19 – Состояние светодиодов модуля

Цель

Считывая подпосписок с SSL-ID W#16#xy19, вы получаете состояние светодиодов модуля.

Примечание

Если вы хотите считать подпосписок W#16#16#xy19 для H CPU, помните, что это возможно только в не резервируемых режимах работы H.

Заголовок

Заголовок подпосписка W#16#xy19 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID подпосписка списка состояний системы W#16#0019 состояние всех светодиодов W#16#0119 состояние одного светодиода
INDEX	LED ID (имеет значение только для SSL-ID W#16#0119) W#16#0001: SF (групповая ошибка) W#16#0002: INTF (внутренняя ошибка) W#16#0003: EXTf (внешняя ошибка) W#16#0004: RUN W#16#0005: STOP W#16#0006: FRCE (принудительное задание) W#16#0007: CRST (рестарт) W#16#0008: BAF (неисправность батареи/перегрузка, короткое замыкание напряжения батареи на шину) W#16#0009: USR (определено пользователем) W#16#000A: USR1 (определено пользователем) W#16#000B: BUS1F (ошибка шины, интерфейс 1) W#16#000C: BUS2F (ошибка шины, интерфейс 2) W#16#000D: REDF (ошибка резервирования) W#16#000E: MSTR (мастер) W#16#000F: RACK0 (стойка номер 0) W#16#0010: RACK1 (стойка номер 1) W#16#0011: RACK2 (стойка номер 2) W#16#0012: IFM1F (ошибка интерфейса интерфейсного модуля 1) W#16#0013: IFM2F (ошибка интерфейса интерфейсного модуля 2)
LENGTHDR	W#16#0004: одна запись данных имеет длину 2 слова (4 байта)
N_DR	Количество записей данных

Запись данных

Запись данных подпосписка с SSL-ID W#16#xy19 имеет следующую структуру:

Имя	Длина	Значение
led_on	1 байт	Состояние светодиода: 0 : выключено 1 : включено
led_blink	1 байт	Состояние мигания светодиода: 0: не мигает 1: мигает нормально (2 Гц) 2: мигает медленно (0,5 Гц)

27.11 SSL-ID W#16#xy22 – Состояние прерывания

Цель

Считывая подписание с SSL-ID W#16#xy22, вы получаете информацию о текущем состоянии обработки прерывания и о прерываниях, генерируемых модулем.

Заголовок

Заголовок подписка с SSL-ID W#16#xy22 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID подписка списка состояний системы W#16#0222 запись данных для заданного прерывания. Вы задаете прерывание (номер OB) с помощью параметра INDEX.
INDEX	Класс прерывания или номер OB (при SSL-ID W#16#0222) W#16#0000: свободный цикл W#16#0A0A: прерывание по времени W#16#1414: прерывание с задержкой W#16#1E23: циклическое прерывание W#16#2828: аппаратное прерывание W#16#5050: прерывание по асинхронной ошибке W#16#005A: фоновое W#16#0064: запуск W#16#7878: прерывание по синхронной ошибке
LENGTHDR	W#16#001C: одна запись данных имеет длину 14 слов (28 байтов)
N DR	Количество записей данных

Запись данных

Запись данных подписка с SSL-ID W#16#xy22 имеет следующую структуру:

Имя	Длина	Значение
info	10 слов	<p>Стартовая информация для соответствующего OB со следующими исключениями:</p> <ul style="list-style-type: none"> Для OB1 имеются в распоряжении текущее минимальное (в байтах 8 и 9) и максимальное (в байтах 10 и 11) время цикла (база времени: мс, счет байтов начинается с 0). Для OB80 спроектированные минимальное и максимальное время цикла могут быть считаны. Для прерываний по ошибке без текущей информации Для прерываний информация о состоянии содержит текущее назначение параметров источника прерывания. Для синхронных ошибок в качестве класса приоритета записывается W#16#7F, если OB еще не исполнены, в противном случае - класс приоритета последнего вызова. Если OB имеет несколько стартовых событий и они на момент извлечения информации еще не наступили, то в качестве номера события возвращается W#16#xyzz, где x: класс события, y: не определено, zz: наименьший определенный номер группы. В противном случае используется номер последнего появившегося стартового события.

Имя	Длина	Значение
	1 слово	<p>Идентификатор обработки</p> <p>Бит 0: событие прерывания заблокировано/разблокировано назначением параметра</p> <p>= 0: разблокировано</p> <p>= 1: заблокировано</p> <p>Бит 1: событие прерывания заблокировано с помощью SFC39 "DIS_IRT"</p> <p>= 0: не заблокировано</p> <p>= 1: заблокировано</p> <p>Бит 2 = 1: источник прерывания активен (существует задание на генерирование для прерываний по времени, запущен ОВ прерываний по времени, запущен ОВ прерываний с задержкой, ОВ циклических прерываний: время пошло)</p> <p>Бит 4: ОВ прерываний</p> <p>= 0: не загружен</p> <p>= 1: загружен</p> <p>Бит 5: ОВ прерываний блокирован функцией тестирования и инсталляции</p> <p>= 1: блокирован</p> <p>Бит 6: запись в диагностический буфер</p> <p>= 1: блокирована</p>
al 2	1 слово	<p>Реакция, если ОВ не загружен/ заблокирован</p> <p>Бит 0 = 1: блокировать источник прерывания</p> <p>Бит 1 = 1: генерировать ошибку события прерывания</p> <p>Бит 2 = 1: CPU переходит в состояние STOP</p> <p>Бит 3 = 1: только отклонить прерывание</p>
al 3	2 слова	<p>Отброшено функциями тестирования и инсталляции:</p> <p>Установленный бит № x означает: номер события, который на x больше, чем наименьший номер события соответствующего ОВ, отбрасывается функцией тестирования и инсталляции.</p>

27.12 SSL-ID W#16#ху32 – Данные о состоянии связи

Цель

Считывая подписание с SSL-ID W#16#ху32, то вы получаете данные о состоянии связи блока.

Заголовок

Заголовок подписка SSL-ID W#16#ху32 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID фрагмента подписка W#16#0132 Данные о состоянии для одного компонента связи CPU (всегда одна запись данных). Компонент связи CPU задается с помощью параметра INDEX. W#16#0232 Данные о состоянии для одного компонента связи (в H-системе в режиме RUN-REDUNDANT [работа с резервированием], возвращается n записей данных, где n – количество резервных CPU в H-системе). Компонент связи CPU задается с помощью параметра INDEX. W#160F32 Только информация заголовка подписка SSL
INDEX	Компонент связи CPU: <ul style="list-style-type: none"> для SSL-ID W#16#0132: <ul style="list-style-type: none"> W#16#0005 диагностика W#16#0008 система времени для SSL-ID W#16#0232: <ul style="list-style-type: none"> W#16#0004 уровень защиты CPU и параметры настройки управления со стороны оператора
LENGTHDR	W#16#00028: одна запись данных имеет длину 20 слов (40 байтов)
N_DR	Количество записей данных

Запись данных

Запись данных подписка с SSL-ID W#16#0132 всегда имеет длину 20 слов. Записи данных имеют различное содержание. Содержание зависит от параметра INDEX, иными словами, от компонента связи CPU, которому принадлежит эта запись данных.

27.13 Запись данных фрагмента подписка с SSL-ID W#16#0132 и индексом W#16#0005

Содержание

Фрагмент подписка с SSL-ID W#16#0132 и индексом W#16#0005 содержит информацию о состоянии диагностики модуля.

Запись данных

Запись данных фрагмента подписка с SSL-ID W#16#0132 и индексом W#16#0005 имеет следующую структуру:

Имя	Длина	Значение
Index	1 слово	W#16#0005: диагностика
erw	1 слово	Расширенный набор функций 0: нет 1: да
send	1 слово	Автоматическая передача 0: нет 1: да
моег	1 слово	Передача диагностических сообщений, определенных пользователем, в настоящее время возможна 0: нет 1: да
res	16 слов	резерв

27.14 Запись данных фрагмента подписка с SSL-ID W#16#0132 и индексом W#16#0008

Содержание

Фрагмент подписка с SSL-ID W#16#0132 и индексом W#16#0008 содержит информацию о состоянии системы времени модуля.

Запись данных

Запись данных фрагмента подписка с SSL-ID W#16#0132 и индексом W#16#0008 имеет следующую структуру:

Имя	Длина	Значение
Index	1 слово	W#16#0008: состояние системы времени
zykl	1 слово	Время цикла кадров синхронизации
korr	1 слово	Коэффициент коррекции для времени
clock 0	1 слово	Счетчик рабочего времени 0: время в часах
clock 1	1 слово	Счетчик рабочего времени 1: время в часах
clock 2	1 слово	Счетчик рабочего времени 2: время в часах
clock 3	1 слово	Счетчик рабочего времени 3: время в часах
clock 4	1 слово	Счетчик рабочего времени 4: время в часах
clock 5	1 слово	Счетчик рабочего времени 5: время в часах
clock 6	1 слово	Счетчик рабочего времени 6: время в часах
clock 7	1 слово	Счетчик рабочего времени 7: время в часах
time	4 слова	Текущая дата и время (формат: DATE_AND_TIME)
bszl_0 – bszl_1	2 байта	Счетчик рабочего времени активен (бит =1: счетчик рабочего времени активен)
bszl_0	1 байт	Бит x: счетчик рабочего времени x, $0 \leq x \leq 7$
bszl_1	1 байт	Резерв
bszu_0 – bszu_1	2 байта	Переполнение счетчика рабочего времени (бит = 1: переполнение)
bszu_0	1 байт	Бит x: счетчик рабочего времени x, $0 \leq x \leq 7$
bszu_1	1 байт	Резерв
Res	3 слова	Резерв

27.15 Запись данных фрагмента подписка с SSL-ID W#16#0232 и индексом W#16#0004

Содержание

Фрагмент подписка с SSL-ID W#16#0232 и индексом W#16#0004 содержит информацию об уровне защиты CPU и параметрах настройки элементов управления со стороны оператора.

В H-системе в режиме RUN-REDUNDANT [работа с резервированием] возвращается одна запись данных на резервный CPU.

Запись данных

Запись данных фрагмента подписка с SSL-ID W#16#0232 и индексом W#16#0004 имеет следующую структуру:

Имя	Длина	Значение
Index	1 слово	<ul style="list-style-type: none"> Байт 1: W#16#04: уровень защиты CPU и параметры настройки управления со стороны оператора Байт 0: Стандартный CPU: W#16#00 H CPU: биты 0 – 2: номер стойки бит 3: 0 = резервный CPU, 1 = главный CPU биты 4 – 7: 1111
sch_schal	1 слово	Уровень защиты установлен с помощью переключателя режимов работы
sch_par	1 слово	Уровень защиты установлен в параметрах
sch_rel	1 слово	Действительный уровень защиты CPU
bart_sch	1 слово	Положение переключателя режимов (1: RUN, 2: RUN-P, 3: STOP, 4: MRES, 0: неопределенное или не может быть определено)
anl_sch	1 слово	Положение переключателя запуска (1: CRST, 2: WRST, 0: неопределенное, не может быть определено, или переключатель отсутствует)
res	14 слов	Резерв

27.16 SSL-ID W#16#xy71 – Групповая информация Н CPU

Цель

Фрагмент частичного списка с SSL- ID W#16#xy71 содержит информацию о текущем состоянии Н-системы.

Заголовок

Заголовок подсписка SSL-ID W#16#xy71 построен следующим образом:

Содержание	Значение
SSL-ID	SSL- ID фрагмента частичного списка: W#16#0071: Информация о текущем состоянии Н-системы W#16#0F71: Информация только заголовка частичного списка SSL
INDEX	W#16#0000
LENGTHDR	W#16#0010: Запись данных имеет длину 8 слов (16 байтов)
N_DR	W#16#0001: Количество записей данных

Запись данных

Запись данных фрагмента подсписка с ID W#16#xy71 имеет следующую структуру:

Содержание	Длина	Значение
redinf	2 байта	Информация о резервировании W#16#0011: отдельный Н CPU W#16#0012: 1 из 2 Н-систем
mwstat1	1 байт	Байт состояния 1 Бит 0: зарезервирован Бит 1: зарезервирован Бит 2: зарезервирован Бит 3: зарезервирован Бит 4: Н-состояние CPU в стойке 0 =0: резервный CPU =1: главный CPU Бит 5: Н-состояние CPU в стойке 1 =0: резервный CPU =1: главный CPU Бит 6: зарезервирован Бит 7: зарезервирован
mwstat2	1 байт	Байт состояния 2 Бит 0: Состояние соединения синхронизации 01: Синхронизация между CPU 0 и CPU 1 =0: невозможна =1: возможна Бит 1: 0 Бит 2: 0 Бит 3: зарезервирован Бит 4: =0: CPU не вставлен в стойку 0 =1: CPU вставлен в стойку 0 (в режиме резервирования : бит 4 = 0) Бит 5: =0: CPU не вставлен в стойку 1 =1: CPU вставлен в стойку 1 (в режиме резервирования : бит 5 = 0) Бит 6: зарезервирован Бит 7: переключение «резервный – главный» после последней депассивации =0: нет =1: да

Содержание	Длина	Значение
hsfcinfo	2 байта	Слово информации для SFC90 "H_CTRL" Бит 0: =0: депассивация не активна =1: депассивация активна Бит 1: =0: обновление резерва разблокировано =1: обновление резерва заблокировано Бит 2: =0: соединение с резервом разблокировано =1: соединение с резервом заблокировано Бит 3: зарезервирован Бит 4: =1: переключение «резервный – главный» запущено с помощью SFC90 Бит 5: зарезервирован Бит 6: =1: резервный CPU переключен в STOP посредством SFC90 Бит 7: =1: затребовано модифицирование с обновлением Бит 8: =1: затребовано модифицирование без обновления
samfehl	2 байта	Слово групповой ошибки Бит 0 = 1: ошибка CPU в стойке 0 Бит 1 = 1: ошибка CPU в стойке 1 Бит 2: зарезервирован Бит 3: зарезервирован Бит 4 = 1: ошибка подключения кабеля FO Биты с 5 по 15: зарезервированы
bz_cpu_0	2 байта	Режим CPU в стойке 0 W#16#0001: STOP (обновление) W#16#0002: STOP (сброс памяти) W#16#0003: STOP (самостоятельная инициализация) W#16#0004: STOP (внутренний) W#16#0005: STARTUP (холодный рестарт) W#16#0006: STARTUP (теплый рестарт) W#16#0007: STARTUP (горячий рестарт) W#16#0008: RUN (режим соло) W#16#0009: RUN-R (избыточный режим) W#16#000A: HOLD (приостановка) W#16#000B: LINK-UP (соединение) W#16#000C: UPDATE (обновление) W#16#000D: DEFECTIVE (дефектный) W#16#000E: SELFTEST (самотестирование) W#16#000F: NO POWER (нет питания)
bz_cpu_1	2 байта	Режим CPU в стойке 1 (значения как для bz_cpu_0)
bz_cpu_2	2 байта	Зарезервирован
cpu_valid	1 байт	Допустимость переменных bz_cpu_0 и bz_cpu_1 В#16#01: bz_cpu_0 допустима В#16#02: bz_cpu_1 допустима В#16#03: bz_cpu_0 и bz_cpu_1 допустимы
reserve	1 байт	Зарезервирован

27.17 SSL-ID W#16#xy74 – Состояние светодиодов модуля

Цель

Если вы читаете частичный список SSL-ID W#16#xy74, то в случае стандартных CPU (если присутствуют) и в случае H CPU вы получаете состояние светодиодов модулей.

Если H CPU находятся в режиме H без резервирования, то вы получаете состояние светодиода адресованного CPU. Если H CPU находятся в режиме RUN-REDUNDANT (работа с резервированием), то возвращается состояние светодиодов всех резервных H CPU.

Заголовок

Заголовок подписка SSL-ID W#16#xy74 построен следующим образом:

Содержание	Значение
SSL-ID	SSL- ID фрагмента частичного списка W#16#0174 состояние светодиода. Светодиод выбирают с помощью параметра INDEX.
INDEX	ID светодиода (имеет значение только для SSL-ID W#16#0174) W#16#0001: SF (групповая ошибка) W#16#0002: INTF (внутренняя ошибка) W#16#0003: EXTf (внешняя ошибка) W#16#0004: RUN W#16#0005: STOP W#16#0006: FRCE (принудительный) W#16#0007: CRST (холодный рестарт) W#16#0008: BAF (отказ батареи/перегрузка, короткое замыкание напряжения батареи на шину) W#16#0009: USR (определяемый пользователем) W#16#000A: USR1 (определяемый пользователем) W#16#000B: BUS1F (интерфейс ошибки шины 1) W#16#000C: BUS2F (интерфейс ошибки шины 2) W#16#000D: REDF (ошибка резервирования) W#16#000E: MSTR (главный) W#16#000F: RACK0 (стойка номер 0) W#16#0010: RACK1 (стойка номер 1) W#16#0011: RACK2 (стойка номер 2) W#16#0012: IFM1F (интерфейсный модуль 1, ошибка интерфейса) W#16#0013: IFM2F (интерфейсный модуль 2, ошибка интерфейса)
LENGTHDR	W#16#0004: одна запись данных имеет длину 2 слова (4 байта)
N_DR	Число записей данных

Запись данных

Запись данных фрагмента подписка с SSL-ID W#16#0074 имеет следующую структуру:

Имя	Длина	Значение
index	1 слово	<ul style="list-style-type: none"> Байт 0 <ul style="list-style-type: none"> Стандартный CPU: В#16#00 Н-CPU: Биты с 0 по 2: номер стойки Бит 3: 0=резервный CPU, 1=главный CPU Биты с 4 по 7: 1111 Байт 1: идентификатор светодиода
led_on	1 байт	Состояние светодиода: 0: выключен 1: включен
led_blink	1 байт	Мигающее состояние светодиода: 0: нет мигания 1: нормальное мигание (2 Гц) 2: медленное мигание (0.5 Гц)

Имя	Длина	Значение
index	1 слово	<ul style="list-style-type: none"> Байт 0 <ul style="list-style-type: none"> Стандартный CPU: В#16#00 Н-CPU: Биты с 0 по 2: номер стойки Бит 3: 0=резервный CPU, 1=главный CPU Биты с 4 по 7: 1111 Байт 1: идентификатор светодиода
led_on	1 байт	Состояние светодиода: 0: выключен 1: включен
led_blink	1 байт	Мигающее состояние светодиода: 0: нет мигания 1: нормальное мигание (2 Гц) 2: медленное мигание (0,5 Гц)

27.18 SSL-ID W#16#xy91 – Информация о состоянии модуля

Цель

Считывая подписание с SSL-ID W#16#xy91, вы получаете информацию о состоянии модуля.

Заголовок

Заголовок подписка SSL-ID W#16#xy91 построен следующим образом:

Содержание	Значение
SSL-ID	<p>SSL-ID фрагмента подписка</p> <p>W#16#0091 информация о состоянии модуля для всех вставленных модулей и субмодулей</p> <p>W#16#0191 информация о состоянии всех модулей/ стоек с неправильными идентификатором типа</p> <p>W#16#0291 информация о состоянии всех неисправных модулей</p> <p>W#16#0391 информация о состоянии всех недоступных модулей</p> <p>W#16#0591 информация о состоянии всех субмодулей основного модуля</p> <p>W#16#0991 информация о состоянии модулей мастер-системы DP</p> <p>W#16#0A91 информация о состоянии модулей всех мастер-систем DP</p> <p>W#16#0C91 информация о состоянии модуля в центральной стойке, подключенного к встроенному интерфейсу DP через логический базовый адрес</p> <p>W#16#4C91 информация о состоянии модуля, подключенного к внешнему интерфейсу DP через логический базовый адрес</p> <p>W#16#0D91 информация о состоянии всех модулей в указанной стойке/ станции (DP)</p> <p>W#16#0E91 информация о состоянии всех спроектированных модулей</p>
INDEX	<ul style="list-style-type: none"> Для фрагмента подписка с SSL-ID W#16#0C91: <ul style="list-style-type: none"> -S7-400: биты 0 – 14: логический базовый адрес модуля бит 15: 0 = вход, 1 = выход -S7-300: начальный адрес модуля Для фрагмента подписка с SSL-ID W#16#4C91 (только S7-400): <ul style="list-style-type: none"> биты 0 – 14: логический базовый адрес модуля бит 15: 0 = вход, 1 = выход Для фрагмента подписка с SSL-ID 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: INDEX не имеет значения, все модули (в стойке и в децентрализованной периферии) Для фрагмента подписка с SSL-ID W#16#0991 и W#16#0D91: <ul style="list-style-type: none"> W#16#00xx все модули и субмодули стойки (xx содержит номер стойки) W#16#xx00 все модули мастер-системы DP (xx содержит идентификатор мастер-системы DP) W#16#xxуу все модули DP-станции (xx содержит идентификатор мастер-системы, уу номер станции)
LENGTHDR	W#16#0010: одна запись данных имеет длину 8 слов (16 байтов)
N_DR	Количество записей данных

Запись данных

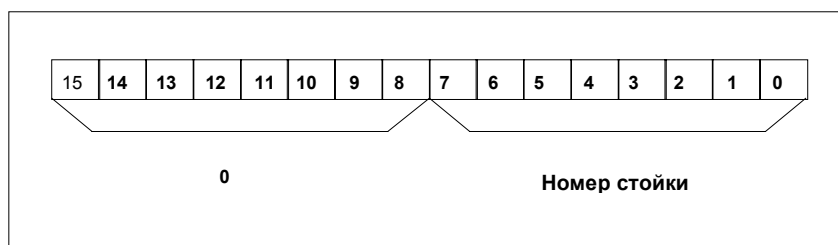
Запись данных подписка с ID W#16#ху91 имеет следующую структуру:

Имя	Длина	Значение
adr1	1 слово	Номер стойки (идентификатор мастер-системы DP с DP) (географический адрес)
adr2	1 слово	Слот модуля и слот субмодуля
logadr	1 слово	Первый назначенный логический адрес входа/выхода (базовый адрес)
solltyp	1 слово	Ожидаемый тип модуля
isttyp	1 слово	Фактический тип модуля
alarm	1 слово	Резерв
eastat	1 слово	Состояние ввода/вывода Бит 0 = 1: модуль неисправен (обнаружено через диагностическое прерывание) Бит 1 = 1: модуль существует Бит 2 = 1: модуля нет (обнаружено по ошибке доступа) Бит 5 = 1: модуль может быть основным модулем для субмодуля Бит 6 = 1: зарезервирован для S7-400 Бит 7 = 1: модуль в локальном сегменте шины Биты 8 – 15: идентификатор данных для логического адреса (вход: B#16#B4, выход: B#16#B5, внешний интерфейс DP: B#16#FF)
ber_bgbr	1 слово	Идентификатор области/ширина модуля Биты 0 – 2 : ширина модуля Биты 4 – 6 : идентификатор области 0 = S7-400 1 = S7-300 2 = область ET 3 = область P 4 = область Q 5 = область IM3 6 = область IM4

adr1

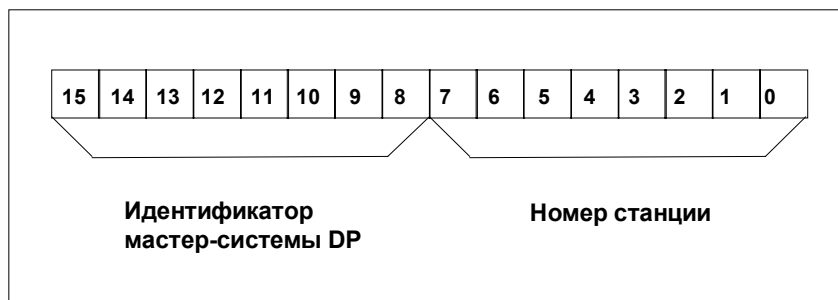
Параметр adr1 содержит следующую информацию:

- при центральной установке – номер стойки.



Биты параметра adr1 при центральной установке

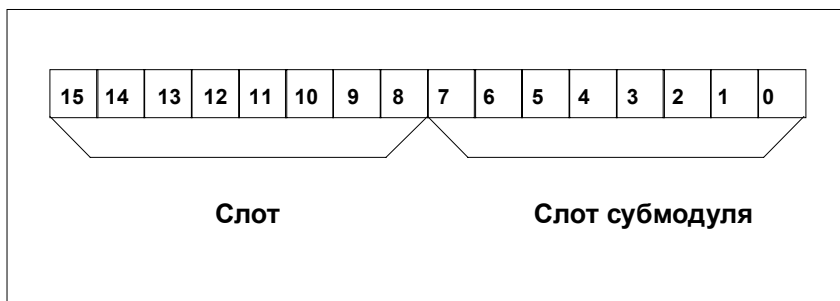
- в случае децентрализованной конфигурации
идентификатор мастер-системы DP
номер станции.



Биты параметра adr1 в децентрализованной конфигурации.

adr2

Параметр adr2 содержит слот модуля и слот submodule.



Биты параметра adr2.

27.19 SSL-ID W#16#xy92 – Информация о состоянии стойки/станции

Цель

Считывая подписание с SSL-ID W#16#xy92, вы получаете информацию об ожидаемой и текущей конфигурации аппаратуры стоек централизованной структуры и станций мастер-системы DP.

Заголовок

Заголовок подписка SSL-ID W#16#xy92 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID фрагмента подписка: W#16#0092: ожидаемое состояние центральных стоек/станций мастер-системы DP, подключенных через встроенный интерфейс DP W#16#4092: ожидаемое состояние станций мастер-системы DP, подключенных через внешний интерфейс DP W#16#0292: фактическое состояние центральных стоек/станций мастер-системы DP, подключенных через встроенный интерфейс DP W#16#4292: фактическое состояние мастер-системы DP, подключенных через внешний интерфейс DP W#16#0692: нормальное состояние стоек расширения в центральной конфигурации/ станций мастер-системы DP, подключенных через встроенный интерфейс DP W#16#4692: нормальное состояние станций мастер-системы DP, подключенных через внешний интерфейс DP
INDEX	0/ идентификатор мастер-системы DP
LENGTHDR	W#16#0010: одна запись данных имеет длину 8 слов (16 байтов)
N_DR	Количество записей данных

Запись данных

Запись данных подписка с ID W#16#ху92 имеет следующую структуру:

Содержание	Длина	Значение
status_0 – status_15	16 байтов	Состояние стойки/станции или состояние буферной батареи. W#16#0092: Бит=0: стойка/станция не запроекирована Бит=1: стойка/станция запроекирована W#16#4092 Бит=0: станция не запроекирована Бит=1: станция запроекирована W#16#0292: Бит=0: стойка/станция неисправна или не запроекирована Бит=1: стойка/станция существует и исправна W#16#4292: Бит=0: станция неисправна или не запроекирована Бит=1: станция существует и исправна W#16#0692: Бит=0: все модули стойки расширения /станции существуют, доступны, ошибок нет Бит=1: по крайней мере, 1 модуль стойки расширения/ станции не в порядке W#16#4692: Бит=0: все модули станции существуют, доступны, ошибок нет Бит=1: по крайней мере, 1 модуль станции не в порядке
status_0	1 байт	Бит 0: центральная стойка (INDEX = 0) или станция 1 (INDEX ≠ 0) Бит 1: 1-я стойка расширения или станция 2 : : Бит 7: 7-я стойка расширения или станция 8
status_1	1 байт	Бит 0: 8-я стойка расширения или станция 9 : : Бит 7: 15-я стойка расширения или станция 16
status_2	1 байт	Бит 0: 16-я стойка расширения или станция 17 : : Бит 5: 21-я стойка расширения или станция 22 Бит 6: 0 или станция 23 Бит 7: 0 или станция 24
status_3	1 байт	Бит 0: 0 или станция 25 : : Бит 5: 0 или станция 30 Бит 6: стойка расширения (область SIMATIC S5) или станция 31 Бит 7: 0 или станция 32

Содержание	Длина	Значение		
status_4	1 байт	Бит 0:	0	или станция 33
			:	
			:	
		Бит 7:	0	или станция 40
:				
:				
status_15	1 байт	Бит 0:	0	или станция 121
			:	
			:	
		Бит 7:	0	или станция 128

27.20 SSL-ID W#16#xyA0 – Диагностический буфер

Цель

Считывая подписание с SSL-ID W#16#xyA0, вы получаете записи в диагностическом буфере модуля.

Заголовок

Заголовок подписка SSL-ID W#16#xyA0 построен следующим образом:

Содержание	Значение
SSL-ID	SSL-ID фрагмента подписка: W#16#00A0: все записи, возможные в текущем режиме W#16#01A0: самые новые записи; количество самых новых задается через параметр INDEX W#16#0FA0: только информация заголовка подписка
INDEX	Только для SSL-ID W#16#01A0: Количество самых новых записей
LENGTHDR	W#16#0014: запись данных имеет длину 10 слов (20 байтов)
N_DR	Количество записей данных

Запись данных

Запись данных подписка с SSL-ID W#16#xyA0 имеет следующую структуру:

Имя	Длина	Значение
ID	1 слово	Идентификатор события
info	5 слов	Информация о событии и его последствиях
time	4 слова	Отметка времени события

Диагностический буфер

Дополнительную информацию к событиям из диагностического буфера вы получите с помощью STEP 7.

27.21 SSL-ID W#16#00B1 – Диагностическая информация модуля

Цель

Считывая подсписок с SSL-ID W#16#00B1, вы получаете первые 4 диагностических байта модуля, обладающего диагностическими свойствами.

Заголовок

Заголовок подсписка SSL-ID W#16#00B1 построен следующим образом:

Содержание	Значение
SSL-ID	W#16#00B1
INDEX	Бит 0 – бит 14: логический базовый адрес Бит 15: 0 = вход, 1 = выход
LENGTHDR	W#16#0004: запись данных имеет длину 2 слова (4 байта)
N_DR	1

Запись данных

Запись данных подсписка с SSL-ID W#16#00B1 имеет следующую структуру:

Имя	Длина	Значение
byte1	1 байт	Бит 0: модуль неисправен/ОК (идентификатор групповой ошибки) Бит 1: внутренняя ошибка Бит 2: внешняя ошибка Бит 3: имеется ошибка канала Бит 4: отсутствует внешнее вспомогательное напряжение Бит 5: отсутствует передний штепсельный разъем Бит 6: модулю не назначены параметры Бит 7: неправильные параметры в модуле
byte2	1 байт	Бит 0 – Бит 3: класс модуля (CPU, FM, CP, IM, SM, ...) Бит 4: имеется информация канала Бит 5: имеется информация пользователя Бит 6: диагностическое прерывание из-за замены Бит 7: резерв (инициализирован 0)
byte3	1 байт	Бит 0: модуль пользователя неправильный/отсутствует Бит 1: нарушение связи Бит 2: рабочий режим RUN/STOP (0 = RUN, 1 = STOP) Бит 3: сработал контроль времени Бит 4: отказал внутренний источник питания модуля Бит 5: разрядилась батарея (BFS) Бит 6: отказ всей буферизации Бит 7: резерв (инициализирован 0)
byte4	1 байт	Бит 0: отказ стойки расширения (обнаружен IM) Бит 1: отказ процессора Бит 2: ошибка СППЗУ Бит 3: ошибка ОЗУ Бит 4: ошибка АЦП/ЦПУ Бит 5: сгорел предохранитель Бит 6: потеряно аппаратное прерывание Бит 7: резерв (инициализирован 0)

27.22 SSL-ID W#16#00B2 – Диагностическая запись данных 1 с географическим адресом

Цель

Считывая подписание с SSL-ID W#16#00B2, вы получаете диагностическую запись данных 1 модуля в центральной стойке (не для DP и субмодулей).
Модуль задается номером стойки и слота.

Заголовок

Заголовок подписка SSL-ID W#16#00B2 построен следующим образом:

Содержание	Значение
SSL-ID	W#16#00B2
INDEX	W#16#ххуу: хх содержит номер стойки уу содержит номер слота
LENGTHDR	Длина записи данных зависит от модуля.
N_DR	1

Запись данных

Размер записи данных подписка с SSL-ID W#16#00B2 и ее содержание зависят от конкретного модуля. За дополнительной информацией обратитесь к /70/, /101/ и к руководству, описывающему соответствующий модуль.

27.23 SSL-ID W#16#00B3 – Диагностические данные модуля с логическим базовым адресом

Цель

Считывая подсписок с SSL-ID W#16#00B3, вы получаете все диагностические данные модуля. Вы можете получить эту информацию также для DP и субмодулей. Модуль выбирается с помощью его логического базового адреса.

Заголовок

Заголовок подсписка SSL-ID W#16#00B3 построен следующим образом:

Содержание	Значение
SSL-ID	W#16#00B3
INDEX	Бит 0 – Бит 14: логический базовый адрес Бит 15: 0 =вход, 1 = выход
LENGTHDR	Длина записи данных зависит от модуля.
N_DR	1

Запись данных

Размер записи данных подсписка с SSL-ID W#16#00B3 и ее содержание зависит от конкретного модуля. За дополнительной информацией обратитесь к [/70/](#), [/101/](#) и к руководству, описывающему соответствующий модуль.

27.24 SSL-ID W#16#00B4 – Диагностические данные Slave-устройства DP

Цель

Считывая подписание с SSL-ID W#16#00B4, вы получаете диагностические данные slave-устройства DP. Эти диагностические данные структурированы в соответствии с EN 50 170 Volume 2, PROFIBUS. Модуль выбирается с помощью спроектированного вами диагностического адреса.

Заголовок

Заголовок подписка SSL-ID W#16#00B4 построен следующим образом:

Содержание	Значение
SSL-ID	W#16#00B4
INDEX	Спроектированный диагностический адрес slave-устройства DP
LENGTHDR	Длина записи данных. Максимальная длина составляет 240 байтов. Для стандартных slave-устройств, имеющих длину диагностических данных больше 240 байтов, но не более 244 байтов, считываются первые 240 байтов, и в данных устанавливается бит переполнения.
N_DR	1

Запись данных

Запись данных подписка с SSL-ID W#16#00B4 имеет следующую структуру:

Имя	Длина	Значение
status1	1 байт	Состояние станции 1
status2	1 байт	Состояние станции 2
status3	1 байт	Состояние станции 3
stat_nr	1 байт	Номер главной станции
ken_hi	1 байт	Идентификатор изготовителя (старший байт)
ken_lo	1 байт	Идентификатор изготовителя (младший байт)
....	Дополнительная диагностика, специфическая для конкретного slave-устройства

28 События

28.1 События и идентификаторы (ID) событий

Событие

Все события в программируемых логических контроллерах SIMATIC S7 нумеруются. Это позволяет поставить в соответствие событию текст сообщения.

Идентификатор (ID) события

Каждому событию соответствует идентификатор (ID) события. ID построен следующим образом:



Структура ID события.

Класс события

События разделены на классы следующим образом:

Номер	Класс события
1	События, связанные со стандартными ОВ
2	Синхронные ошибки
3	Асинхронные ошибки
4	Переходы из одного рабочего режима в другой
5	События этапа выполнения
6	Коммуникационные события
7	События в отказоустойчивых системах
8	Стандартизованные диагностические данные в модулях
9	События, определяемые пользователем
A, B	Свободно определяемые события
C, D, E	Резерв
F	События для модулей, отличных от CPU (например, CP, FM)

Идентификатор

Идентификатор используется, чтобы различать типы событий. Эти четыре бита имеют следующие значения:

№ бита в ID события	Значение	
8	= 0	уходящее событие
	= 1	наступающее событие
9	= 1	запись в диагностический буфер
10	= 1	внутренняя ошибка
11	= 1	внешняя ошибка

28.2 Класс событий 1 – События, связанные со стандартными ОВ

ID события	Событие
W#16#1381	Запрос на ручной теплый рестарт
W#16#1382	Запрос на автоматический теплый рестарт
W#16#1383	Запрос на ручной горячий рестарт
W#16#1384	Запрос на автоматический горячий рестарт
W#16#1385	Запрос на ручной холодный рестарт
W#16#1386	Запрос на автоматический холодный рестарт
W#16#1387	Основной CPU: запрос на ручной холодный рестарт
W#16#1388	Основной CPU: запрос на автоматический холодный рестарт
W#16#138A	Основной CPU: запрос на ручной теплый рестарт
W#16#138B	Основной CPU: запрос на автоматический теплый рестарт
W#16#138C	Резервный CPU: запрос на ручной горячий рестарт
W#16#138D	Резервный CPU: запрос на автоматический горячий рестарт

28.3 Класс событий 2 – Синхронные ошибки

ID события	Событие	ОВ
W#16#2521	Ошибка BCD - преобразования	ОВ121
W#16#2522	Ошибка длины области при чтении	
W#16#2523	Ошибка длины области при записи	
W#16#2524	Ошибка области при чтении	
W#16#2525	Ошибка области при записи	
W#16#2526	Ошибка номера таймера	
W#16#2527	Ошибка номера счетчика	
W#16#2528	Ошибка выравнивания при чтении	
W#16#2529	Ошибка выравнивания при записи	
W#16#2530	Ошибка записи при обращении к DB	
W#16#2531	Ошибка записи при обращении к DI	
W#16#2532	Ошибка номера блока при открытии DB	
W#16#2533	Ошибка номера блока при открытии DI	
W#16#2534	Ошибка номера блока при вызове FC	
W#16#2535	Ошибка номера блока при вызове FB	
W#16#253A	DB не загружен	
W#16#253C	FC не загружен	
W#16#253D	SFC не загружен	
W#16#253E	FB не загружен	
W#16#253F	SFB не загружен	ОВ122
W#16#2942	Ошибка доступа к периферии, чтение	
W#16#2943	Ошибка доступа к периферии, запись	
W#16#2944	Ошибка доступа к периферии при n-ном обращении для чтения (n>1)	
W#16#2945	Ошибка доступа к периферии при n-ном обращении для записи (n>1)	

28.4 Класс событий 3 – Асинхронные ошибки

ID события	Событие	ОВ
W#16#3501	Превышение длительности цикла.	ОВ80
W#16#3502	Ошибка запроса пользовательского интерфейса (ОВ или FRB)	
W#16#3503	Слишком большая задержка при обработке класса приоритета	-
W#16#3505	Прерывание(я) по времени, пропущенное(ые) из-за новой установки часов	ОВ80
W#16#3506	Прерывание(я) по времени, пропущенное(ые) при переходе в RUN после HOLD	
W#16#3507	Многочисленные ошибки при запросе ОВ, вызванные переполнением буфера стартовой информации	
W#16#3921/3821	BATTF: выход из строя по крайней мере одной буферной батареи в центральной стойке/ проблема устранена	ОВ81
W#16#3922/3822	BAF: буферное напряжение в центральной стойке отсутствует / проблема устранена	
W#16#3923/3823	Выход из строя блока питания 24 В в центральной стойке / проблема устранена	
W#16#3925/3825	BATTF: выход из строя по крайней мере одной буферной батареи резервной центральной стойки/ проблема устранена	
W#16#3926/3826	BAF: отсутствует буферное напряжение в резервной центральной стойке / проблема устранена	
W#16#3917/3827	Выход из строя блока питания 24 В в резервной центральной стойке / проблема устранена	
W#16#3931/3831	BATTF: выход из строя по крайней мере одной буферной батареи в стойке расширения/ проблема устранена	
W#16#3932/3832	BAF: отсутствует буферное напряжение в стойке расширения/ проблема устранена	
W#16#3933/3833	Выход из строя блока питания 24 В по крайней мере в одной стойке расширения / проблема устранена	
W#16#3942	Модуль неисправен	ОВ82
W#16#3842	Модуль исправен	
W#16#3861	Модуль установлен, тип модуля верен	ОВ83
W#16#3961	Модуль удален, недоступен	
W#16#3863	Модуль установлен, но неверен тип модуля	
W#16#3864	Модуль установлен, но поврежден (не читается идентификатор типа)	
W#16#3865	Модуль установлен, но неверно назначены параметры модуля	
W#16#3981	Ошибка интерфейса, наступающее событие	ОВ84
W#16#3881	Ошибка интерфейса, уходящее событие	
W#16#35A1	Пользовательский интерфейс (ОВ или FRB) не найден	ОВ85
W#16#35A2	ОВ не загружен (запущен посредством SFC или операционной системы на основе проектирования)	
W#16#35A3	Ошибка при обращении к блоку операционной системы	
W#16#39B1	Ошибка доступа к периферии при обновлении таблицы входов образа процесса	ОВ85
W#16#39B2	Ошибка доступа к периферии при передаче образа процесса в модули вывода	
W#16#39B3/38B3	Ошибка доступа к периферии при обновлении таблицы входов образа процесса	

ID события	Событие	ОВ
W#16#39B4/38B4	Ошибка доступа к периферии при передаче образа процесса в модули вывода	
W#16#38C1	Выход из строя стойки расширения (1 - 21), уходящее событие	ОВ86
W#16#39C1	Выход из строя стойки расширения (1 - 21), наступающее событие	
W#16#38C2	Стойка расширения восстановлена, но с расхождением между заданной и фактической конфигурацией	
W#16#39C3	Децентрализованная периферия: отказ мастер-системы, наступающее событие	
W#16#39C4	Децентрализованная периферия: отказ станции, наступающее событие	
W#16#38C4	Децентрализованная периферия: отказ станции, уходящее событие	
W#16#39C5	Децентрализованная периферия: отказ станции, наступающее событие	
W#16#38C5	Децентрализованная периферия: отказ станции, уходящее событие	
W#16#38C6	Стойка расширения восстановлена, но ошибка(и) при назначении параметров модулю	
W#16#38C7	DP: станция восстановлена, но ошибка(и) при назначении параметров модулю	
W#16#38C8	DP: станция восстановлена, но несовпадение между заданной и фактической конфигурацией	
W#16#35D2	Передача диагностических записей в данное время невозможна	ОВ87
W#16#35D3	Кадры синхронизации не могут быть посланы	
W#16#35D4	Недопустимый скачок времени в результате синхронизации	
W#16#35D5	Ошибка при приеме времени синхронизации	
W#16#35E1	Неверный идентификатор кадра в GD	
W#16#35E2	Статус пакета GD не может быть занесен в DB	
W#16#35E3	Ошибка длины кадра в GD	
W#16#35E4	Принят недопустимый номер пакета GD	
W#16#35E5	Ошибка при обращении к DB в коммуникационных SFB для проектирования соединений S7	
W#16#35E6	Общий статус GD не может быть занесен в DB	

28.5 Класс событий 4 – События, связанные с режимом STOP, и другие изменения режима работы

ID события	Событие
W#16#4300	Включение резервного питания
W#16#4301	Переход из режима STOP в режим STARTUP (запуск)
W#16#4302	Переход из режима STARTUP в RUN
W#16#4303	STOP в результате перевода в STOP переключателя режимов работы
W#16#4304	STOP в результате команды STOP от PG или через SFB20 "STOP"
W#16#4305	HOLD: достигнута точка останова
W#16#4306	HOLD: точка останова покинута
W#16#4307	Запуск сброса памяти командой из PG
W#16#4308	Запуск сброса памяти от переключателя режимов
W#16#4309	Автоматический запуск сброса памяти (небуферизованное включение питания)
W#16#430A	HOLD покинут, переход в STOP
W#16#430D	STOP, вызванный другим CPU в мультипроцессорном режиме
W#16#430E	Сброс памяти выполнен
W#16#430F	STOP в модуле из-за перехода в STOP в CPU
W#16#4510	STOP, вызванный нарушением диапазона данных CPU
W#16#4520	DEFFECTIVE: STOP невозможен
W#16#4521	DEFFECTIVE: : отказ процессора обработки команд
W#16#4522	DEFFECTIVE: отказ микросхемы часов
W#16#4523	DEFFECTIVE: отказ генератора тактовых импульсов
W#16#4524	DEFFECTIVE: отказ функции обновления таймера
W#16#4525	DEFFECTIVE: отказ многопроцессорной синхронизации
W#16#4926	DEFFECTIVE: отказ контроля времени при обращении к периферии
W#16#4527	DEFFECTIVE: отказ контроля обращения к периферии
W#16#4528	DEFFECTIVE: отказ контроля длительности цикла
W#16#4530	DEFFECTIVE: ошибка тестирования внутренней памяти
W#16#4931	DEFFECTIVE: ошибка тестирования памяти субмодуля
W#16#4532	DEFFECTIVE: отказ ресурсов ядра
W#16#4933	DEFFECTIVE: ошибка контрольной суммы
W#16#4934	DEFFECTIVE: память недоступна
W#16#4935	DEFFECTIVE: отменено сторожевой схемой/ ненормальные состояния процессора
W#16#4536	DEFFECTIVE: неисправен переключатель режимов работы
W#16#4540	STOP: расширение внутренней рабочей памяти имеет пробелы. Первое расширение памяти слишком мало или отсутствует.
W#16#4541	STOP, вызванный системой классов приоритета
W#16#4542	STOP, вызванный системой управления объектом
W#16#4543	STOP, вызванный функциями тестирования
W#16#4544	STOP, вызванный диагностической системой
W#16#4545	STOP, вызванный системой связи
W#16#4546	STOP, вызванный управлением памятью CPU
W#16#4547	STOP, вызванный управлением образом процесса
W#16#4548	STOP, вызванный управлением периферией
W#16#4949	STOP, вызванный непрерывным аппаратным прерыванием
W#16#454A	STOP, вызванный проектированием, отмененный OB был загружен при полном рестарте
W#16#494D	STOP, вызванный ошибкой ввода/вывода
W#16#494E	STOP, вызванный сбоем питания
W#16#494F	STOP, вызванный ошибкой конфигурации

ID события	Событие
W#16#4550	DEFECTIVE: внутренняя системная ошибка
W#16#4555	Повторный запуск невозможен, истекло время контроля
W#16#4556	STOP: запрос на сброс памяти от системы связи
W#16#4357	Запущено время контроля модуля
W#16#4358	Все модули готовы к работе
W#16#4959	STOP: не все модули готовы к работе
W#16#4562	STOP, вызванный ошибкой программирования (ОВ не загружен или его запуск невозможен или отсутствует FRB)
W#16#4563	STOP, вызванный ошибкой доступа к периферии (ОВ не загружен или его запуск невозможен или отсутствует FRB)
W#16#4567	STOP, вызванный H-событием
W#16#4568	STOP, вызванный ошибкой времени (ОВ не загружен или его запуск невозможен или отсутствует FRB)
W#16#456A	STOP, вызванный диагностическим прерыванием (ОВ не загружен или его запуск невозможен или отсутствует FRB)
W#16#456B	STOP, вызванный снятием/установкой модуля (ОВ не загружен или его запуск невозможен или отсутствует FRB)
W#16#456C	STOP, вызванный аппаратной ошибкой CPU (ОВ не загружен или его запуск невозможен или отсутствует FRB)
W#16#456D	STOP, вызванный ошибкой исполнения программы (ОВ не загружен или его запуск невозможен или отсутствует FRB)
W#16#456E	STOP, вызванный ошибкой связи (ОВ не загружен или его запуск невозможен или отсутствует FRB)
W#16#456F	STOP, вызванный неисправностью стойки (ОВ не загружен или его запуск невозможен или отсутствует FRB)
W#16#4571	STOP, вызванный ошибкой скобочного стека
W#16#4572	STOP, вызванный ошибкой стека главного управляющего реле
W#16#4573	STOP из-за превышения глубины вложения при синхронных ошибках
W#16#4574	STOP, вызванный превышением глубины вложения стека прерываний в стеке классов приоритета
W#16#4575	STOP, вызванный превышением глубины вложения стека блоков в стеке классов приоритета
W#16#4576	STOP, вызванный ошибкой при размещении локальных данных
W#16#4578	STOP, вызванный неизвестным кодом операции
W#16#457A	STOP, вызванный ошибкой длины кода
W#16#457B	STOP, вызванный незагруженным DB у встроенной периферии
W#16#457F	STOP, вызванный командой STOP
W#16#4580	STOP: содержимое резервного буфера противоречиво (нет перехода в RUN)
W#16#4590	STOP, вызванный перегрузкой внутренних функций
W#16#49A0	STOP, вызванный ошибкой назначения параметров или недопустимым расхождением заданного и фактического расширения: запуск заблокирован.
W#16#49A1	STOP, вызванный ошибкой назначения параметров: запрос на сброс памяти
W#16#49A2	STOP, вызванный ошибкой при модификации параметров: запуск заблокирован
W#16#49A3	STOP, вызванный ошибкой при модификации параметров: запрос на сброс памяти
W#16#49A4	STOP: противоречивость данных проектирования
W#16#49A5	STOP: децентрализованная периферия: противоречия в загруженной проектной информации
W#16#49A6	STOP: децентрализованная периферия: недопустимая проектная информация
W#16#49A7	STOP: децентрализованная периферия: отсутствует проектная информация

ID события	Событие
W#16#49A8	STOP: ошибка, отображаемая интерфейсным модулем для децентрализованной периферии
W#16#43B0	Обновление программы ПЗУ было успешным
W#16#49B1	Ошибочные данные при обновлении программы ПЗУ
W#16#49B2	Обновление программы ПЗУ: версия аппаратуры не соответствует программе ПЗУ
W#16#49B3	Обновление программы ПЗУ: тип модуля не соответствует программе ПЗУ
W#16#49D0	LINK-UP (соединение) прервано из-за нарушения правил координации
W#16#49D1	LINK-UP/UPDATE (соединение/обновление) прервано
W#16#49D2	Резервный CPU перешел в STOP из-за перехода в STOP в основном CPU при соединении
W#16#43D3	STOP в резервном CPU, запрошенный пользователем
W#16#49D4	STOP в главном CPU, так как CPU-партнер тоже является главным (ошибка соединения)
W#16#45D5	LINK-UP/UPDATE (соединение/обновление) прервано из-за недопустимой конфигурации памяти CPU подустройства
W#16#45D6	LINK-UP (соединение/обновление) прервано из-за противоречий в системной программе подустройства
W#16#49D7	LINK-UP (соединение/обновление) прервано из-за противоречий в коде программы пользователя на платах памяти
W#16#45D8	DEFFECTIVE: неисправность аппаратуры, обнаруженная благодаря другим ошибкам
W#16#45D9	STOP из-за ошибки модуля синхронизации (SYNC)
W#16#45DA	STOP из-за ошибки синхронизации между H CPU
W#16#43E0	Выход из режима самостоятельной работы после соединения
W#16#43E1	Выход из режима соединения после обновления
W#16#43E2	Переход из режима обновления в режим резервирования
W#16#43E3	Основной CPU: переход из режима резервирования в режим самостоятельной работы
W#16#43E4	Резервный CPU: выход из режима резервирования после режима поиска ошибок
W#16#43E5	Резервный CPU: выход из режима поиска ошибок после соединения или состояния STOP
W#16#43E6	Соединение аварийно завершено на резервном CPU
W#16#43E7	Обновление аварийно завершено на резервном CPU
W#16#43E8	Резервный CPU: выход из соединения после запуска
W#16#43E9	Резервный CPU: выход из запуска после обновления
W#16#43F1	Переключение "резервный-основной"

28.6 Класс событий 5 – События этапа выполнения

ID события	Событие
W#16#530D	Новая информация о запуске в состоянии STOP
W#16#5311	Запуск несмотря на отсутствие сообщения о готовности от модуля(ей)
W#16#5961	Ошибка назначения параметров
W#16#5962	Ошибка назначения параметров, препятствующая запуску
W#16#5963	Ошибка назначения параметров с запросом на сброс памяти
W#16#5371	Децентрализованная периферия: конец синхронизации с master-устройством DP
W#16#5979/5879	Диагностическое сообщение от интерфейса DP: светодиод EXTf вкл/выкл
W#16#5380	Заблокированы записи в диагностический буфер прерываний и асинхронных ошибок
W#16#5395	Децентрализованная периферия: сброс master-устройства DP

28.7 Класс событий 6 – Коммуникационные события

ID события	Событие
W#16#6500	Идентификатор соединения содержится в модуле дважды
W#16#6501	Ресурсов связи недостаточно
W#16#6502	Ошибка в описании соединения
W#16#6905/6805	Проблема с ресурсами в спроектированных соединениях/устранена
W#16#6510	Ошибка структуры CFB в экземплярном DB при анализе СППЗУ
W#16#6514	Номер пакета GD содержится в модуле дважды
W#16#6515	Некорректное задание длины в проектной информации GD
W#16#6316	Ошибка интерфейса при запуске программируемого контроллера
W#16#6521	Ни субмодуль памяти, ни внутренняя память недоступны
W#16#6522	Недопустимый субмодуль памяти: замените субмодуль и сбросьте память
W#16#6523	Запрос на сброс памяти из-за ошибки при обращении к субмодулю
W#16#6524	Запрос на сброс памяти из-за ошибки в заголовке блока
W#16#6526	Запрос на сброс памяти из-за замены памяти
W#16#6527	Память заменена, поэтому рестарт невозможен
W#16#6528	Функция обработки объекта в режиме STOP/HOLD, рестарт невозможен
W#16#6529	Запуск невозможен при выполнении команды "Загрузить программу пользователя"
W#16#652A	Нет запуска, т.к. блок содержится дважды в памяти пользователя
W#16#652B	Нет запуска, т.к. длина блока слишком велика для субмодуля – замените субмодуль
W#16#652C	Нет запуска из-за недопустимого ОВ в субмодуле
W#16#6532	Нет запуска из-за недопустимой информации о конфигурации в субмодуле
W#16#6533	Запрос на сброс памяти из-за недопустимого содержимого субмодуля
W#16#6534	Нет запуска: блок содержится в субмодуле многократно
W#16#6535	Нет запуска: недостаточен объем памяти, чтобы передать блок из субмодуля
W#16#6536	Нет запуска: субмодуль содержит недопустимый номер блока
W#16#6537	Нет запуска: субмодуль содержит блок недопустимой длины
W#16#6538	Локальные данные или идентификатор защиты от записи (для DB) блока недопустимы для CPU
W#16#6539	В блоке недопустимая команда (выявлена компилятором)
W#16#653A	Запрос на сброс памяти, т.к. локальные данные ОВ в субмодуле слишком кратки
W#16#6543	Нет запуска: недопустимый тип блока
W#16#6544	Нет запуска: атрибут "существенно для обработки" недопустим
W#16#6545	Исходный язык недопустим
W#16#6546	Достигнуто максимальное количество проектной информации
W#16#6547	Ошибка при назначении параметров модулям (не на Р-шине, отмените загрузку)
W#16#6548	Ошибка достоверности при проверке блока
W#16#6549	Структурная ошибка в блоке
W#16#6550	Блок имеет ошибку в CRC
W#16#6551	У блока нет CRC
W#16#6560	Переполнение SCAN
W#16#6981	Приходящая ошибка интерфейса
W#16#6881	Уходящая ошибка интерфейса

28.8 Класс событий 7 - События Н/Ф

ID события	Событие	ОВ
W#16#7301	Потеря резервирования (1 из 2) вследствие отказа CPU	OB72
W#16#7302	Потеря резервирования (1 из 2) из-за запущенного пользователем перехода в STOP в резерве	
W#16#7303	Н-система (1 из 2) перешла в режим резервирования	
W#16#9710/7810	Потеря резервных входов/выходов	OB70
W#16#9711/7811	Частичная потеря резервных входов/выходов	
W#16#7520	Ошибка при сравнении RAM	OB 72
W#16#7521	Ошибка при сравнении значения выходов образа процесса	
W#16#7522	Ошибка при сравнении битов памяти, таймеров или счетчиков	
W#16#7323	Обнаружено противоречие в данных операционной системы	
W#16#7331	Переключение на резервное master-устройство из-за отказа master-устройства	
W#16#7333	Переключение на резервное master-устройство из-за вмешательства оператора	
W#16#7934	Переключение на резервное master-устройство из-за проблемы подключения в модуле синхронизации	
W#16#7335	Переключение на резервное master-устройство, запущенное SFC90 "H_CTRL"	
W#16#7340	Ошибка синхронизации в программе пользователя из-за истекшего времени ожидания	
W#16#7341	Ошибка синхронизации в программе пользователя из-за ожидания в различных точках синхронизации	
W#16#7342	Ошибка синхронизации в операционной системе из-за ожидания в различных точках синхронизации	
W#16#7343	Ошибка синхронизации в операционной системе из-за истекшего времени ожидания	
W#16#7344	Ошибка синхронизации в операционной системе из-за неправильных данных	
W#16#734A	Было выполнено задание на "депассивацию", запущенное SFC90 "H_CTRL"	-
W#16#734B	Было выполнено задание на "депассивацию", запущенное операционной системой	
W#16#7950	Модуль синхронизации не существует	OB 72
W#16#7951	Изменение в модуле синхронизации без включения питания	
W#16#7952/7852	Модуль синхронизации удален/вставлен	
W#16#7953	Изменение в модуле синхронизации без сброса	
W#16#7954	Модуль синхронизации: номер стойки назначен дважды	
W#16#7955/7855	Ошибка модуля синхронизации/устранена	
W#16#7956	В модуле синхронизации установлен запрещенный номер стойки	
W#16#79A2	Отказ master-устройства DP или master-систем DP	OB70
W#16#79A3	Потеря резервирования в slave-устройстве DP	
W#16#78A3	Возврат резервирования в slave-устройстве DP	
W#16#79C1	Прерывание обновления	OB72

28.9 Класс событий 8 – Диагностические события для модулей

ID события	Событие	Тип модуля
W#16#8x00	Модуль неисправен/ исправен	Любой
W#16#8x01	Внутренняя ошибка	
W#16#8x02	Внешняя ошибка	
W#16#8x03	Ошибка канала	
W#16#8x04	Отсутствует внешнее вспомогательное напряжение	
W#16#8x05	Отсутствует фронт-штекер	
W#16#8x06	Параметры не назначены	
W#16#8x07	Неверные параметры в модуле	
W#16#8x30	Субмодуль пользователя неисправен/ не найден	
W#16#8x31	Неисправность связи	
W#16#8x32	Режим работы: RUN/STOP (STOP: наступающий, RUN: уходящий)	
W#16#8x33	Сработал контроль времени (сторожевая схема)	
W#16#8x34	Вышло из строя внутренний источник питания модуля	
W#16#8x35	BATTF: батарея разряжена	
W#16#8x36	Вышла из строя вся буферизация	
W#16#8x37	Резерв	
W#16#8x40	Вышла из строя стойка расширения	
W#16#8x41	Вышел из строя процессор	
W#16#8x42	Ошибка СППЗУ	
W#16#8x43	Ошибка ОЗУ	
W#16#8x44	Ошибка АЦП/ЦАП	
W#16#8x45	Сгорел предохранитель	
W#16#8x46	Потеряно аппаратное прерывание	
W#16#8x47	Резерв	
W#16#8x50	Ошибка проектирования/ параметризации	Аналоговый ввод
W#16#8x51	Синфазная ошибка	
W#16#8x52	Короткое замыкание на фазу	
W#16#8x53	Короткое замыкание на землю	
W#16#8x54	Обрыв провода	
W#16#8x55	Ошибка опорного канала	
W#16#8x56	Переход нижней границы измерения	
W#16#8x57	Переход верхней границы измерения	Аналоговый выход
W#16#8x60	Ошибка проектирования/ параметризации	
W#16#8x61	Синфазная ошибка	
W#16#8x62	Короткое замыкание на фазу	
W#16#8x63	Короткое замыкание на землю	
W#16#8x64	Обрыв провода	
W#16#8x65	Резерв	
W#16#8x66	Отсутствует напряжение на зажимах	Цифровой ввод
W#16#8x70	Ошибка проектирования/ параметризации	
W#16#8x71	Неисправность заземления на корпус	
W#16#8x72	Короткое замыкание на фазу (датчик)	
W#16#8x73	Короткое замыкание на землю (датчик)	
W#16#8x74	Обрыв провода	

ID события	Событие	Тип модуля
W#16#8x75	Отсутствует питание датчика	Цифровой вывод
W#16#8x80	Ошибка проектирования/ параметризации	
W#16#8x81	Неисправность заземления на корпус	
W#16#8x82	Короткое замыкание на фазу	
W#16#8x83	Короткое замыкание на землю	
W#16#8x84	Обрыв провода	
W#16#8x85	Обрыв цепи плавкого предохранителя	
W#16#8x86	Нет напряжения на зажимах	
W#16#8x87	Превышение температуры	
W#16#8xB0	Счетчик, ошибка в сигнале A	FM
W#16#8xB1	Счетчик, ошибка в сигнале B	
W#16#8xB2	Счетчик, ошибка в сигнале N	
W#16#8xB3	Счетчик, между каналами передано ошибочное значение	
W#16#8xB4	Счетчик, неисправно питание датчика 5,2 В	
W#16#8xB5	Счетчик, неисправно питание датчика 24 В	

28.10 Класс событий 9 – Стандартные события пользователя

ID события	Событие
W#16#9001	Автоматический режим
W#16#9101	Ручной режим
W#16#9x02	ОТКРЫТО/ЗАКРЫТО, ВКЛ/ВЫКЛ
W#16#9x03	Разрешение ручного управления
W#16#9x04	Команда защиты агрегата (ОТКРЫТО/ЗАКРЫТО)
W#16#9x05	Деблокировка процесса
W#16#9x06	Команда защиты системы
W#16#9x07	Сработал контроль фактического значения
W#16#9x08	Сработал контроль заданного значения
W#16#9x09	Ошибка регулирования больше допустимой
W#16#9x0A	Ошибка граничного состояния
W#16#9x0B	Динамическая ошибка
W#16#9x0C	Ошибка исполнения команды (генератор последовательности)
W#16#9x0D	Рабочий режим выполняется > ОТКРЫТО
W#16#9x0E	Рабочий режим выполняется > ЗАКРЫТО
W#16#9x0F	Блокирование команды
W#16#9x11	Состояние процесса ОТКРЫТО /ВКЛЮЧЕНО
W#16#9x12	Состояние процесса ЗАКРЫТО / ВЫКЛЮЧЕНО
W#16#9x13	Состояние процесса промежуточное
W#16#9x14	Состояние процесса ВКЛЮЧЕНО через АВТОМАТИЧЕСКИЙ РЕЖИМ
W#16#9x15	Состояние процесса ВКЛЮЧЕНО через РУЧНОЙ РЕЖИМ
W#16#9x16	Состояние процесса ВКЛЮЧЕНО через команду защиты
W#16#9x17	Состояние процесса ВЫКЛЮЧЕНО через АВТОМАТИЧЕСКИЙ РЕЖИМ
W#16#9x18	Состояние процесса ВЫКЛЮЧЕНО через РУЧНОЙ РЕЖИМ
W#16#9x19	Состояние процесса ВЫКЛЮЧЕНО через команду защиты
W#16#9x21	Ошибка функционирования при приближении
W#16#9x22	Ошибка функционирования при удалении
W#16#9x31	Исполнительный орган (DE/WE) граничное положение ОТКРЫТО
W#16#9x32	Исполнительный орган (DE/WE) граничное положение не ОТКРЫТО
W#16#9x33	Исполнительный орган (DE/WE) граничное положение ЗАКРЫТО
W#16#9x34	Исполнительный орган (DE/WE) граничное положение не ЗАКРЫТО
W#16#9x41	Неразрешенное состояние, допустимое время истекло
W#16#9x42	Неразрешенное состояние, допустимое время не истекло
W#16#9x43	Ошибка блокировки, допустимое время = 0
W#16#9x44	Ошибка блокировки, допустимое время > 0
W#16#9x45	Реакция отсутствует
W#16#9x46	Неразрешенный выход из конечного состояния, допустимое время = 0
W#16#9x47	Неразрешенный выход из конечного состояния, допустимое время > 0
W#16#9x50	Верхняя граница диапазона сигнала USR
W#16#9x51	Верхняя граница диапазона измерения UMR
W#16#9x52	Нижняя граница диапазона сигнала LSR
W#16#9x53	Нижняя граница диапазона измерения LMR
W#16#9x54	Верхняя граница аварийного сигнала UAL
W#16#9x55	Верхняя граница предупреждения UWL
W#16#9x56	Верхняя допустимая граница UTL
W#16#9x57	Нижняя допустимая граница LTL
W#16#9x58	Нижняя граница предупреждения LWL

ID события	Событие
W#16#9x59	Нижняя граница аварийного сигнала LAL
W#16#9x60	Шаг GRAPH7 наступающий/уходящий
W#16#9x61	Ошибка блокировки GRAPH7
W#16#9x62	Ошибка исполнения GRAPH7
W#16#9x63	Ошибка GRAPH7 принята к сведению
W#16#9x64	Ошибка GRAPH7 квитируется
W#16#9x70	Среднее значение перейдено в положительном направлении
W#16#9x71	Среднее значение перейдено в отрицательном направлении
W#16#9x72	Реакция отсутствует
W#16#9x73	Недопустимый выход из конечного состояния
W#16#9x80	Выход за верхнюю границу, допустимое время = 0
W#16#9x81	Выход за верхнюю границу, допустимое время > 0
W#16#9x82	Выход за нижнюю границу, допустимое время = 0
W#16#9x83	Выход за нижнюю границу, допустимое время > 0
W#16#9x84	Выход за верхнюю границу градиента, допустимое время = 0
W#16#9x85	Выход за верхнюю границу градиента, допустимое время > 0
W#16#9x86	Выход за нижнюю границу градиента, допустимое время = 0
W#16#9x87	Выход за нижнюю границу градиента, допустимое время > 0
W#16#9190/9090	Ошибка назначения параметров пользователя наступающая/ уходящая
W#16#91F0	Перепополнение
W#16#91F1	Потеря значимости
W#16#91F2	Деление на ноль
W#16#91F3	Недопустимая математическая операция

28.11 Классы событий А и В – Свободные пользовательские события

ID события	Событие
W#16#Axyz	События, доступные для пользователя
W#16#Bxyz	

28.12 Зарезервированные классы событий

Зарезервировано

Следующие классы событий зарезервированы для последующих расширений:

- C
- D
- E
- F Зарезервировано для модулей, не находящихся в центральной стойке (например, CP или FM)

29 Список SFC, SFB и FC

29.1 Список SFC, упорядоченный по номерам

№	Краткое название	Функция	Раздел
SFC0	SET_CLK	Установка системных часов	5.1
SFC1	READ_CLK	Чтение системных часов	5.2
SFC2	SET_RTM	Установка счетчика рабочего времени	6.2
SFC3	CTRL_RTM	Запуск/останов счетчика рабочего времени	6.3
SFC4	READ_RTM	Чтение счетчика рабочего времени	6.4
SFC5	GADR_LGC	Выяснение логического адреса канала	14.1
SFC6	RD_SINFO	Считывание стартовой информации OB	12.2
SFC7	DP_PRAL	Запуск аппаратного прерывания на master-устройстве DP	15.1
SFC9	EN_MSG	Деблокирование сообщений, связанных с блоком, сообщений, связанных с символом, и сообщений о групповом состоянии	19.8
SFC10	DIS_MSG	Блокирование сообщений, связанных с блоком, сообщений, связанных с символом, и сообщений о групповом состоянии	19.7
SFC11	DPSYC_FR	Синхронизация групп slave-устройств DP	15.2
SFC13	DPNRM_DG	Чтение диагностических данных slave-устройства DP (Slave-диагностика)	15.3
SFC14	DPRD_DAT	Чтение непротиворечивых данных стандартного slave-устройства DP	15.4
SFC15	DPWR_DAT	Запись непротиворечивых данных в стандартное slave-устройство DP	15.5
SFC17	ALARM_SQ	Генерирование квитуемых сообщений, связанных с блоками	19.12
SFC18	ALARM_S	Генерирование постоянно квитуемых сообщений, связанных с блоками	19.12
SFC19	ALARM_SC	Определение состояния квитирования последнего пришедшего сообщения ALARM_SQ	19.13
SFC20	BLKMOV	Копирование переменных	3.1
SFC21	FILL	Инициализация области памяти	3.2
SFC22	CREAT_DB	Создание блока данных	3.3
SFC23	DEL_DB	Удаление блока данных	3.4
SFC24	TEST_DB	Тестирование блока данных	3.5
SFC25	COMPRESS	Сжатие памяти пользователя	3.6
SFC26	UPDAT_PI	Обновление таблицы входов образа процесса	13.1
SFC27	UPDAT_PO	Обновление таблицы выходов образа процесса	13.2
SFC28	SET_TINT	Установка прерывания по времени	8.3
SFC29	CAN_TINT	Отмена прерывания по времени	8.4
SFC30	ACT_TINT	Активизация прерывания по времени	8.5
SFC31	QRY_TINT	Опрос прерывания по времени	8.6
SFC32	SRT_DINT	Запуск прерывания с задержкой	9.2
SFC33	CAN_DINT	Отмена прерывания с задержкой	9.3
SFC34	QRY_DINT	Опрос прерывания с задержкой	9.4
SFC35	MP_ALM	Запуск мультипроцессорного прерывания	4.4
SFC36	MSK_FLT	Маскирование синхронных ошибок	10.2
SFC37	DMSK_FLT	Демаскирование синхронных ошибок	10.3
SFC38	READ_ERR	Чтение регистра ошибок	10.4
SFC39	DIS_IRT	Блокирование новых прерываний и асинхронных ошибок	11.2
SFC40	EN_IRT	Деблокирование новых прерываний и асинхронных ошибок	13.3

№	Краткое название	Функция	Раздел
SFC41	DIS_AIRT	Задержка более приоритетных прерываний и асинхронных ошибок	11.4
SFC42	EN_AIRT	Деблокирование более приоритетных прерываний и асинхронных ошибок	11.5
SFC43	RE_TRIGR	Повторный запуск контроля времени цикла	4.1
SFC44	REPL_VAL	Передача заменяющего значения в аккумулятор 1	3.7
SFC46	STP	Перевод CPU в состояние STOP	4.2
SFC47	WAIT	Задержка исполнения программы пользователя	4.3
SFC48	SNC_RTCB	Синхронизация ведомых часов	5.3
SFC49	LGC_GADR	Выявление слота модуля, соответствующего логическому адресу	14.2
SFC50	RD_LGADR	Выявление всех логических адресов модуля	14.3
SFC51	RDSYSST	Считывание списка и подписка состояний системы	12.3
SFC52	WR_USMSG	Запись в диагностический буфер диагностического события, определенного пользователем	12.4
SFC54	RD_PARM	Считывание определенных параметров	7.2
SFC55	WR_PARM	Запись динамических параметров	7.3
SFC56	WR_DPARM	Запись параметров по умолчанию	7.4
SFC57	PARM_MOD	Назначение параметров модулю	7.5
SFC58	WR_REC	Внесение записи данных	7.6
SFC59	RD_REC	Чтение записи данных	7.7
SFC60	GD_SND	Передача GD-пакета	16.1
SFC61	GD_RCV	Извлечение принятого GD-пакета	16.2
SFC62	CONTROL	Опрос состояния соединения, принадлежащего экземпляру коммуникационного SFB	17.5
SFC63	AB_CALL	Ассемблирование кодового блока	*
SFC64	TIME_TCK	Чтение системного времени	6.5
SFC65	X_SEND	Передача данных коммуникационному партнеру вне локальной станции S7	18.5
SFC66	X_RCV	Прием данных от коммуникационного партнера вне локальной станции S7	18.6
SFC67	X_GET	Чтение данных от коммуникационного партнера вне локальной станции S7	18.7
SFC68	X_PUT	Запись данных коммуникационному партнеру вне локальной станции S7	18.8
SFC69	X_ABORT	Прерывание существующего соединения с коммуникационным партнером вне локальной станции S7	18.9
SFC72	I_GET	Чтение данных от коммуникационного партнера внутри локальной станции S7	18.10
SFC73	I_PUT	Запись данных коммуникационному партнеру внутри локальной станции S7	18.11
SFC74	I_ABORT	Прерывание существующего соединения с коммуникационным партнером внутри локальной станции S7	18.12
SFC79	SET	Установка области выходов	13.3
SFC80	RSET	Сброс области выходов	13.4
SFC81	UBLKMOV	Непрерываемое копирование переменных	3.2
SFC90	H_CTRL	Управление функционированием в H-системах	23.1

* SFC63 "AB_CALL" существует только для CPU 614. За подробным описанием обратитесь к соответствующему руководству.

29.2 Список SFC, упорядоченный по алфавиту

Краткое название	№	Функция	Раздел
AB_CALL	SFC63	Ассемблирование кодового блока	*
ACT_TINT	SFC30	Активизация прерывания по времени	8.5
ALARM_SQ	SFC17	Генерирование квитуемых сообщений, связанных с блоками	19.12
ALARM_S	SFC18	Генерирование постоянно квитуемых сообщений, связанных с блоками	19.13
ALARM_SC	SFC19	Определение состояния квитирования последнего пришедшего сообщения ALARM_SQ	19.12
BLKMOV	SFC20	Копирование переменных	3.1
CAN_DINT	SFC33	Отмена прерывания с задержкой	9.4
CAN_TINT	SFC29	Отмена прерывания по времени	8.4
COMPRESS	SFC25	Сжатие памяти пользователя	3.6
CONTROL	SFC62	Опрос состояния соединения, принадлежащего экземпляру коммуникационного SFB	17.5
CREAT_DB	SFC22	Создание блока данных	3.3
CTRL_RTM	SFC3	Запуск/останов счетчика рабочего времени	6.3
DEL_DB	SFC23	Удаление блока данных	3.4
DIS_AIRT	SFC41	Задержка более приоритетных прерываний и асинхронных ошибок	11.4
DIS_IRT	SFC39	Блокирование новых прерываний и асинхронных ошибок	11.2
DIS_MSG	SFC10	Блокирование сообщений, связанных с блоком, сообщений, связанных с символом, и сообщений о групповом состоянии	19.7
DMSK_FLT	SFC37	Демаскирование синхронных ошибок	10.3
DP_PRAL	SFC7	Запуск аппаратного прерывания на master-устройстве DP	15.1
DPNRM_DG	SFC13	Чтение диагностических данных slave-устройства DP (Slave-диагностика)	15.3
DPRD_DAT	SFC14	Чтение непротиворечивых данных стандартного slave-устройства DP	15.4
DPSYC_FR	SFC11	Синхронизация групп slave-устройств DP	15.2
DPWR_DAT	SFC15	Запись непротиворечивых данных в стандартное slave-устройство DP	15.5
EN_AIRT	SFC42	Деблокирование более приоритетных прерываний и асинхронных ошибок	11.5
EN_IRT	SFC40	Деблокирование новых прерываний и асинхронных ошибок	13.3
EN_MSG	SFC9	Деблокирование сообщений, связанных с блоком, сообщений, связанных с символом, и сообщений о групповом состоянии	19.8
FILL	SFC21	Инициализация области памяти	3.2
GADR_LGC	SFC5	Выяснение логического адреса канала	14.1
GD_RCV	SFC61	Извлечение принятого GD-пакета	16.2
GD_SND	SFC60	Передача GD-пакета	16.1
H_CTRL	SFC90	Управление функционированием в H-системах	23.1
I_ABORT	SFC74	Прерывание существующего соединения с коммуникационным партнером внутри локальной станции S7	18.12
I_GET	SFC72	Чтение данных от коммуникационного партнера внутри локальной станции S7	18.10
I_PUT	SFC73	Запись данных коммуникационному партнеру внутри локальной станции S7	18.11

Краткое название	№	Функция	Раздел
LGC_GADR	SFC49	Выявление слота модуля, соответствующего логическому адресу	14.1
MP_ALM	SFC35	Запуск мультипроцессорного прерывания	4.4
MSK_FLT	SFC36	Маскирование синхронных ошибок	10.2
PARM_MOD	SFC57	Назначение параметров модулю	7.5
QRY_DINT	SFC34	Опрос прерывания с задержкой	9.3
QRY_TINT	SFC31	Опрос прерывания по времени	8.6
RD_PARM	SFC54	Считывание определенных параметров	7.2
RD_LGADR	SFC50	Выявление всех логических адресов модуля	14.3
RD_REC	SFC59	Чтение записи данных	7.7
RD_SINFO	SFC6	Считывание стартовой информации ОВ	12.2
RDSYST	SFC51	Считывание списка и подсписка состояний системы	12.3
READ_CLK	SFC1	Чтение системных часов	4.1
READ_RTM	SFC4	Чтение счетчика рабочего времени	5.2
READ_ERR	SFC38	Чтение регистра ошибок	10.4
REPL_VAL	SFC44	Передача заменяющего значения в аккумулятор 1	6.4
RE_TRIGR	SFC43	Повторный запуск контроля времени цикла	3.7
RSET	SFC80	Сброс области выходов	13.4
SET	SFC79	Установка области выходов	13.3
SET_CLK	SFC0	Установка системных часов	5.1
SET_RTM	SFC2	Установка счетчика рабочего времени	6.2
SET_TINT	SFC28	Установка прерывания по времени	8.3
SNC_RTCB	SFC48	Синхронизация ведомых часов	5.3
SRT_DINT	SFC32	Запуск прерывания с задержкой	9.2
STP	SFC46	Перевод CPU в состояние STOP	4.2
TEST_DB	SFC24	Тестирование блока данных	3.5
TIME_TCK	SFC64	Чтение системного времени	6.5
UBKLMOV	SFC81	Непрерываемое копирование переменных	3.2
UPDAT_PI	SFC26	Обновление таблицы входов образа процесса	13.1
UPDAT_PO	SFC27	Обновление таблицы выходов образа процесса	13.2
WAIT	SFC47	Задержка исполнения программы пользователя	4.3
WR_DPARM	SFC56	Запись параметров по умолчанию	7.4
WR_PARM	SFC55	Запись динамических параметров	7.3
WR_REC	SFC58	Внесение записи данных	7.6
WR_USMSG	SFC52	Запись в диагностический буфер диагностического события, определенного пользователем	12.4
X_ABORT	SFC69	Прерывание существующего соединения с коммуникационным партнером вне локальной станции S7	18.9
X_GET	SFC67	Чтение данных от коммуникационного партнера вне локальной станции S7	18.7
X_PUT	SFC68	Запись данных коммуникационному партнеру вне локальной станции S7	18.8
X_RCV	SFC66	Прием данных от коммуникационного партнера вне локальной станции S7	18.6
X_SEND	SFC65	Передача данных коммуникационному партнеру вне локальной станции S7	18.5

* SFC63 "AB_CALL" существует только для CPU 614. За подробным описанием обратитесь к соответствующему руководству.

29.3 Список SFB, упорядоченный по номерам

№	Краткое название	Функция	Раздел
SFB0	CTU	Прямой счет	20.4
SFB1	CTD	Обратный счет	20.5
SFB2	CTUD	Прямой и обратный счет	20.6
SFB3	TP	Генерирование импульса	20.1
SFB4	TON	Генерирование задержки включения	
SFB5	TOF	Генерирование задержки выключения	20.3
SFB8	USEND	Некоординированная передача данных	17.3
SFB9	URCV	Некоординированный прием данных	17.4
SFB12	BSEND	Передача сегментированных данных	17.5
SFB13	BRCV	Прием сегментированных данных	17.6
SFB14	GET	Чтение данных из удаленного CPU	17.7
SFB15	PUT	Запись данных в удаленный CPU	17.8
SFB16	PRINT	Передача данных на принтер	17.9
SFB19	START	Инициализация полного рестарта в удаленном устройстве	17.10
SFB20	STOP	Перевод удаленного устройства в состояние STOP	17.11
SFB21	RESUME	Инициализация рестарта в удаленном устройстве	17.12
SFB22	STATUS	Опрос состояния удаленного партнера	17.13
SFB23	USTATUS	Получение состояния удаленного устройства	17.14
SFB29	HS_COUNT	Счетчик (высокоскоростной счетчик, встроенная функция)	*
SFB30	FREQ_MES	Частотомер (измеритель частоты, встроенная функция)	*
SFB32	DRUM	Реализация генератора последовательностей	13.5
SFB33	ALARM	Генерирование сообщений, связанных с блоком, с индикацией квитирования	19.3
SFB34	ALARM_8	Генерирование сообщений, связанных с блоком, без сопутствующих значений для восьми сигналов	19.5
SFB35	ALARM_8P	Генерирование сообщений, связанных с блоком, с сопутствующими значениями для восьми сигналов	19.4
SFB36	NOTIFY	Генерирование сообщений, связанных с блоком, без индикации квитирования	19.2
SFB37	AR_SEND	Передача архивных данных	19.6
SFB38	HSC_A_B	Счетчик A/B (встроенная функция)	*
SFB39	POS	Позиционирование (встроенная функция)	*
SFB41 ¹⁾	CONT_C	Непрерывное регулирование	22.1
SFB42 ¹⁾	CONT_S	Ступенчатое регулирование	22.2
SFB43 ¹⁾	PULSEGEN	Генерирование импульсов	22.3

* SFB29 "HS_COUNT" и SFB30 "FREQ_MES" имеются только в CPU 312 IFM и CPU 314 IFM. SFB 38 "HSC_A_B" и 39 "POS" имеются только в CPU 314 IFM. За подробным описанием обратитесь к /73/.

1) SFB 41 "CONT_C," 42 "CONT_S" и 43 "PULSEGEN" имеются только в CPU 314 IFM.

29.4 Список SFB, упорядоченный по алфавиту

Краткое название	№	Функция	Раздел
ALARM	SFB33	Генерирование сообщений, связанных с блоком, с индикацией квитирования	19.3
ALARM_8	SFB34	Генерирование сообщений, связанных с блоком, без сопутствующих значений для восьми сигналов	19.5
ALARM_8P	SFB35	Генерирование сообщений, связанных с блоком, с сопутствующими значениями для восьми сигналов	19.4
AR_SEND	SFB37	Передача архивных данных	19.6
BRCV	SFB13	Прием сегментированных данных	17.6
BSEND	SFB12	Передача сегментированных данных	17.5
CONT_C ¹⁾	SFB41	Непрерывное регулирование	22.1
CONT_S ¹⁾	SFB42	Ступенчатое регулирование	22.2
CTD	SFB1	Обратный счет	20.5
CTU	SFB0	Прямой счет	20.4
CTUD	SFB2	Прямой и обратный счет	20.6
DRUM	SFB32	Реализация генератора последовательностей	13.5
FREQ_MES	SFB30	Частотомер (измеритель частоты, встроенная функция)	*
GET	SFB14	Чтение данных из удаленного CPU	17.7
HSC_A_B	SFB38	Счетчик A/B (встроенная функция)	*
HS_COUNT	SFB29	Счетчик (высокоскоростной счетчик, встроенная функция)	*
NOTIFY	SFB36	Генерирование сообщений, связанных с блоком, без индикации квитирования	19.2
POS	SFB39	Позиционирование (встроенная функция))	*
PRINT	SFB16	Передача данных на принтер	17.9
PULSEGEN ¹⁾	SFB43	Генерирование импульсов	22.3
PUT	SFB15	Запись данных в удаленный CPU	17.8
RESUME	SFB21	Инициализация рестарта в удаленном устройстве	17.12
START	SFB19	Инициализация полного рестарта в удаленном устройстве	17.10
STATUS	SFB22	Опрос состояния удаленного партнера	17.13
STOP	SFB20	Перевод удаленного устройства в состояние STOP	20.3
TOF	SFB5	Генерирование задержки выключения	
TON	SFB4	Генерирование задержки включения	20.1
TP	SFB3	Генерирование импульса	17.4
URCV	SFB9	Некоординированный прием данных	17.4
USEND	SFB8	Некоординированная передача данных	17.3
USTATUS	SFB23	Получение состояния удаленного устройства	17.14

* SFB29 "HS_COUNT" и SFB30 "FREQ_MES" имеются только в CPU 312 IFM и CPU 314 IFM. SFB 38 "HSC_A_B" и 39 "POS" имеются только в CPU 314 IFM. За подробным описанием обратитесь к [/73/](#).

1) SFB 41 "CONT_C," 42 "CONT_S" и 43 "PULSEGEN" имеются только в CPU 314 IFM.

29.5 Функции IEC, упорядоченные по номерам

№	Краткое название	Функция	Раздел
FC1	AD_DT_TM	Прибавление интервала к моменту времени	21.4
FC2	CONCAT	Слияние двух переменных типа STRING	21.7
FC3	D_TOD_DT	Объединение DATE и TIME_OF_DAY в DT	21.4
FC4	DELETE	Удаление в переменной типа STRING	21.7
FC5	DI_STRNG	Преобразование типа данных DINT в STRING	21.8
FC6	DT_DATE	Извлечение DATE из DT	21.4
FC7	DT_DAY	Извлечение дня недели из DT	21.4
FC8	DT_TOD	Извлечение TIME_OF_DAY из DT	21.4
FC9	EQ_DT	Сравнение DT на равенство	21.5
FC10	EQ_STRNG	Сравнение STRING на равенство	21.6
FC11	FIND	Поиск в переменной типа STRING	21.7
FC12	GE_DT	Сравнение DT на больше или равно	21.5
FC13	GE_STRNG	Сравнение STRING на больше или равно	21.6
FC14	GT_DT	Сравнение DT на больше	21.5
FC15	GT_STRNG	Сравнение STRING на больше	21.6
FC16	I_STRNG	Преобразование типа данных INT в STRING	21.8
FC17	INSERT	Вставка в переменную типа STRING	21.7
FC18	LE_DT	Сравнение DT на меньше или равно	21.5
FC19	LE_STRNG	Сравнение STRING на меньше или равно	21.6
FC20	LEFT	Левая часть переменной типа STRING	21.7
FC21	LEN	Длина переменной типа STRING	21.7
FC22	LIMIT	Ограничитель	21.9
FC23	LT_DT	Сравнение DT на меньше	21.5
FC24	LT_STRNG	Сравнение STRING на меньше	21.6
FC25	MAX	Выбор максимума	21.9
FC26	MID	Средняя часть переменной типа STRING	21.7
FC27	MIN	Выбор минимума	21.9
FC28	NE_DT	Сравнение DT на не равно	21.5
FC29	NE_STRNG	Сравнение STRING на не равно	21.6
FC30	R_STRNG	Преобразование типа данных REAL в STRING	21.8
FC31	REPLACE	Замена в переменной типа STRING	21.7
FC32	RIGHT	Правая часть переменной типа STRING	21.7
FC33	S5TI_TIM	Преобразование типа данных S5TIME в TIME	21.4
FC34	SB_DT_DT	Вычитание двух значений времени	21.4
FC35	SB_DT_TM	Вычитание интервала из момента времени	21.4
FC36	SEL	Двоичный выбор	21.10
FC37	STRNG_DI	Преобразование типа данных STRING в DINT	21.8
FC38	STRNG_I	Преобразование типа данных STRING в INT	21.8
FC39	STRNG_R	Преобразование типа данных STRING в REAL	21.8
FC40	TIM_S5TI	Преобразование типа данных TIME в S5TIME	21.4

Глоссарий

А

Адрес

Адрес - это идентификатор, присваиваемый ячейке памяти или некоторой области ячеек памяти, например: вход I 12.1; меркерное слово MW25; блок данных DB3.

Адресация

Назначение адреса в программе пользователя. Адреса могут быть назначены ячейкам памяти или областям ячеек памяти (например: вход I 12.1; меркерное слово MW25).

Аккумулятор

Аккумуляторы - это регистры в →CPU, которые служат в качестве промежуточной памяти при операциях загрузки, передачи, а также сравнения, преобразования и в арифметических операциях.

Аппаратное прерывание

Аппаратное прерывание вызывается модулями, выполняющими прерывание на основе определенных событий в процессе. Аппаратное прерывание передается в CPU. Затем, в соответствии с приоритетом этого прерывания, обрабатывается соответствующий → организационный блок.

Б

Битовая память

Битовая память - это память, допускающая побитовый доступ. Основные операции STEP 7 имеют доступ к битовой памяти на запись и чтение (с адресацией побитно, побайтно, пословно и двойными словами). Область битовой памяти может использоваться для сохранения промежуточных результатов.

Блок данных (DB)

Блоки данных - это области в программе пользователя, содержащие данные пользователя. Имеются совместно используемые (глобальные) блоки данных, к которым можно обратиться из любого кодового блока, и экземплярные блоки данных, которые ставятся в соответствие отдельному вызову функционального блока (FB).

В

Встроенный регулятор

Встроенный регулятор - это готовый запрограммированный блок регулирования, хранящийся в операционной системе и содержащий важнейшие функции управления по замкнутому контуру. Пользователь с помощью программного переключателя может включать или выключать эти функции.

Входные параметры

Входные параметры имеются только у → функций и → функциональных блоков. С помощью входных параметров данные передаются для обработки в вызываемый блок.

Г

Групповая ошибка

Сообщение об ошибке, отображаемое светодиодами на передней панели модулей (только) в S7-300. Светодиод загорается при любой ошибке (→ внутренней или → внешней) в соответствующем модуле.

Д

Двухпозиционный регулятор

Двухпозиционным регулятором называется регулятор, в котором управляющее воздействие может принимать только два состояния (напр., включено – выключено).

Диагностика

Диагностические функции охватывают всю системную диагностику и включают в себя распознавание, интерпретацию и сообщение об ошибках внутри системы автоматизации.

Диагностическая запись

Диагностическое событие описывается в диагностическом буфере с помощью диагностической записи.

Диагностический буфер

Диагностический буфер - это область памяти в CPU, в которой хранятся все диагностические события в порядке их возникновения.

Диагностические данные

Диагностические данные - это информация, содержащаяся в сообщении об ошибке (диагностическое событие, метка времени).

Диагностическое прерывание

Модули, обладающие диагностическими свойствами, сообщают в → CPU о распознанных системных ошибках с помощью диагностических прерываний.

Диагностическое сообщение

Диагностическое сообщение состоит из обработанного диагностического события и посылается из CPU на устройство отображения.

И

Интегрирующее звено

Интегрирующее звено регулятора. При скачкообразном изменении регулируемой величины (или сигнала ошибки) выходная величина растёт линейно с течением времени со скоростью, пропорциональной $K_I (=1/T_I)$. Интегрирующее звено в замкнутом контуре регулирования действует так, что выходное значение регулируемой величины изменяется до тех пор, пока сигнал ошибки не станет равным нулю.

Информация о стартовом событии

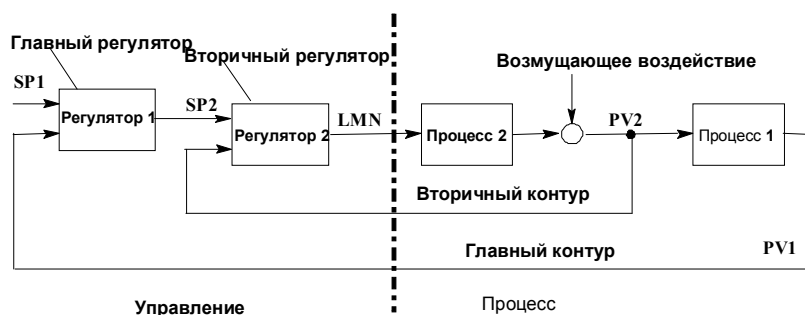
Информация о стартовом событии - это составная часть → организационного блока (ОВ). Она предоставляет пользователю S7 подробную информацию о событии, вызвавшем запуск этого ОВ. Информация о стартовом событии содержит номер события (состоящий из класса и идентификатора события), метку времени события, а также дополнительные данные (например, адрес сигнального модуля, вызвавшего прерывание).

К

Каскадное регулирование

Каскадное регулирование включает в себя ряд взаимосвязанных регуляторов, в котором главный регулятор (master) устанавливает задающее воздействие для вторичных (slave) регуляторов в соответствии с текущим сигналом ошибки главной регулируемой величины.

Благодаря включению дополнительных переменных процесса результат регулирования с помощью каскадного регулятора может быть улучшен. Для этого в подходящей точке снимается значение вспомогательной регулируемой величины PV2, которая регулируется в соответствии эталонным задающим воздействием (выход главного регулятора SP2). Главный регулятор стабилизирует фактическое значение PV1 в соответствии с жестко заданным значением SP1 и для этого устанавливает SP2 таким образом, чтобы эта цель могла быть достигнута возможно быстрее и без перерегулирования.



Класс приоритета

Операционная система CPU имеет до 28 классов приоритета, которым ставятся в соответствие различные организационные блоки. Класс приоритета определяет, какие ОВ могут прерывать исполнение других ОВ. Если класс приоритета включает в себя более одного ОВ, то они не прерывают друг друга, а выполняются последовательно.

Команда

Команда (STEP 5 или STEP 7) - это самая малая самостоятельная единица программы, созданной на текстовом языке. Она представляет рабочее предписание для процессора.

Коммуникационные SFB для спроектированных соединений

Коммуникационные SFB - это системные функциональные блоки для обмена данными и управления программами.

Примеры обмена данными: SEND, RECEIVE, GET.

Примеры управления программами: перевод CPU коммуникационного партнера в состояние STOP, опрос состояния CPU коммуникационного партнера.

Коммуникационные SFC для неспроектированных соединений

Коммуникационные SFC – это системные функции для обмена данными и прерывания существующих соединений, установленных с помощью коммуникационных SFC.

Константа

“Константы” - это маркеры для постоянных величин в → логических блоках. Константы применяются для повышения удобочитаемости программы. Например, вместо прямого ввода значения (напр., 10) в → функциональном блоке указывается маркер "Max_iteration_count" ["Макс_количество_итераций"]. Значение константы (например, 10) затем вводится при вызове блока.

Контур регулирования

Под контуром регулирования понимается соединение выхода процесса (регулируемой величины) с входом регулятора и выхода регулятора (управляющего воздействия) с входом процесса, так что регулятор и процесс образуют замкнутый контур.

Л

Логический блок

Логический блок в SIMATIC S7 - это блок, содержащий часть программы пользователя STEP 7. Другим типом блока является → блок данных, который содержит только данные. Следующий список перечисляет типы логических блоков:

- организационный блок (OB)
- функциональный блок (FB)
- функция (FC)
- системный функциональный блок (SFB)
- системная функция (SFC).

Н

Непрерывный регулятор

У непрерывного регулятора любое изменение сигнала ошибки вызывает изменение управляющего воздействия. Последнее может принимать любое значение внутри своей области изменения.

О

Обработка ошибок с помощью OB

Если системная программа распознает определенную ошибку (например, → ошибку доступа в S7), то она вызывает предусмотренный для этого случая организационный блок, в котором программой пользователя может быть установлена реакция CPU на ошибку.

Исполнение программы, управляемое событиями

При исполнении программы, управляемой событиями, циклическая программа пользователя прерывается стартовыми событиями (→ классы приоритетов). Если происходит стартовое событие, то исполняемый в данный момент блок прерывается перед следующей командой, и вызывается и выполняется соответствующий организационный блок. Затем циклическая обработка программы продолжается с точки прерывания.

Объект регулирования

Объектом регулирования (процессом) называется часть установки, в которой регулируемая величина находится под влиянием управляющего воздействия (благодаря изменению энергии, необходимой для перемещения исполнительного органа, или потока материала). Его можно разделить на исполнительное устройство и управляемый процесс.



Операционная система CPU

Операционная система CPU организует все функции и процессы CPU, не связанные со специальной задачей управления.

Описание переменной

Описание переменной включает в себя ввод символического имени, типа данных и, возможно, начального значения, адреса и комментария.

Организационный блок (ОВ)

Организационные блоки образуют интерфейс между операционной системой CPU и программой пользователя. В организационных блоках определяется последовательность обработки программы пользователя.

Ошибка, асинхронная

Асинхронные ошибки – это → ошибки этапа исполнения программы, не привязанные к определенному месту в программе пользователя (например, неисправность блока питания, превышение времени цикла). При возникновении такой ошибки операционная система вызывает соответствующий → организационный блок, в котором пользователь может запрограммировать реакцию на эту ошибку.

Ошибка, синхронная

Синхронные ошибки – это ошибки → этапа выполнения программы, которые можно поставить в соответствие определенному месту в программе пользователя (например, ошибка доступа к модулю ввода / ввода). При возникновении такой ошибки операционная система вызывает соответствующий организационный блок, в котором пользователь может запрограммировать реакцию на эту ошибку.

Ошибки этапа исполнения

Ошибки, возникающие во время обработки программы пользователя в программируемом логическом контроллере (т.е. не в процессе).

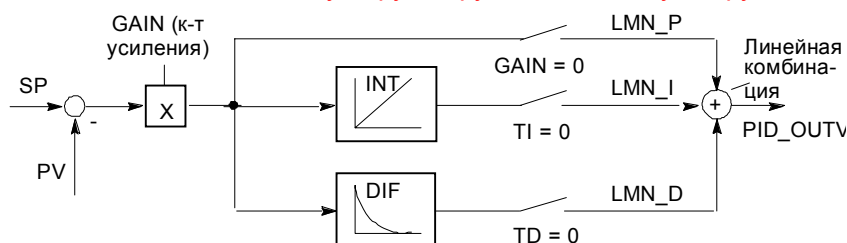
Ошибки программы пользователя

Ошибки, которые могут возникнуть при обработке программы пользователя в микроконтроллере SIMATIC S7 (в противоположность → ошибкам процесса). Обработка ошибок операционной системой производится с помощью ОВ ошибок (система → классов приоритета), → слова состояния и выходных параметров → системных функций.

П

Параллельная структура

Параллельная структура - это специальный вид обработки сигналов в регуляторе (математическая обработка). Пропорциональная, интегральная и дифференциальная компоненты рассчитываются параллельно, не взаимодействуя друг с другом, а затем суммируются.



Параметр

1. Параметр – это переменная логического блока S7 (см. → параметр блока → фактический параметр → формальный параметр).
2. Переменная для настройки поведения модуля (одна или несколько на модуль)

Каждый конфигурируемый модуль при поставке с завода имеет некоторую рациональную базовую установку, которая может быть изменена с помощью STEP 7.

Имеется два вида параметров: статические и динамические
→ Параметр, статический / →Параметр, динамический).

Параметр, динамический

Динамические параметры модулей, в отличие от статических, могут быть изменены программой пользователя во время работы модуля с помощью вызова SFC, например, граничные значения аналогового модуля.

Параметр, статический

Статические параметры модулей, в отличие от динамических, не могут быть изменены программой пользователя, а только через STEP 7, например, задержка входа цифрового модуля ввода.

Параметры модуля

Параметры модуля – это величины, с помощью которых можно установить режим работы модуля. В зависимости от конкретного модуля некоторые из этих параметров могут быть изменены в программе пользователя.

Параметры регулятора

Параметры регулятора – это характеристические значения для согласования статических и динамических свойств регулятора с заданными свойствами объекта регулирования или процесса.

Переменная

Переменная – это элемент данных с переменным содержимым, который может быть применен в программе пользователя STEP 7. Переменная состоит из адреса (например, M 3.1) и типа данных (например, BOOL) и представляется символом (напр., MOTOR_ON).

Полный рестарт

При запуске CPU (например, при переводе переключателя режимов работы из STOP в RUN или при включении питающего напряжения) перед циклической обработкой программы (OB1) сначала выполняется организационный блок OB101 (рестарт; только у S7-400) или OB100 (полный рестарт). При полном рестарте считывается таблица входов образа процесса и программа пользователя STEP 7 обрабатывается начиная с первой команды в OB1.

Прерывание

Система классов приоритета → SIMATIC S7 распознает 10 различных классов приоритета, управляющих обработкой программы пользователя. К этим классам приоритета относятся, в частности, прерывания, например, аппаратные прерывания. При возникновении прерывания операционная система автоматически вызывает соответствующий организационный блок, в котором пользователь может запрограммировать желаемую реакцию (например, в функциональном блоке).

Прерывание по времени

Прерывание по времени относится к одному из классов приоритета при исполнении программы SIMATIC S7. Оно генерируется в определенный день и момент времени (например, 9:50 или ежечасно, ежеминутно). Затем выполняется соответствующий организационный блок.

Прерывание с задержкой

Прерывание с задержкой относится к одному из классов приоритета при исполнении программы SIMATIC S7. Оно генерируется при завершении работы запущенного в программе пользователя таймера. Затем выполняется соответствующий организационный блок.

Приоритет

Назначая приоритет организационному блоку, вы определяете возможность прерывания активной в данный момент программы пользователя так, что события с высоким приоритетом прерывают события с более низким приоритетом.

Приоритет ОБ

→ Операционная система CPU различает классы приоритета, например, циклическую обработку программы, обработку программы, управляемую аппаратными прерываниями. Каждому классу приоритета ставится в соответствие → организационный блок (ОБ), в котором пользователь S7 может запрограммировать реакцию на событие. ОБ имеют различные приоритеты, позволяющие обрабатывать их в правильной последовательности при одновременном вызове и дающие возможность ОБ с более высоким приоритетом прерывать ОБ с более низким приоритетом. Стандартные приоритеты могут изменяться пользователем.

Программа пользователя

Программа пользователя содержит все операторы и описания, а также данные для обработки сигналов, с помощью которых можно управлять системой или процессом. Она ставится в соответствие программируемому модулю (→ Модуль, программируемый) и может быть структурирована делением на более мелкие единицы, известные как → блоки.

Пропорциональное исполнительное звено

→ Широтно-импульсная модуляция

Процесс

→ Объект регулирования

Р

Реакция на ошибку

Реакция на ошибку → этапа исполнения программы. Операционная система может реагировать следующим образом: переводом контроллера в состояние STOP, вызовом организационного блока, в котором пользователь может запрограммировать реакцию на ошибку, или отображением ошибки.

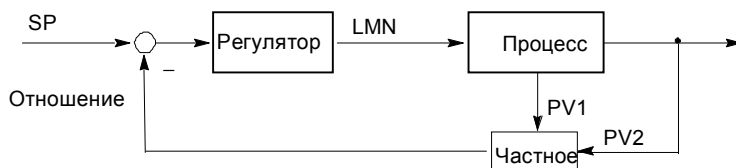
Результат логической операции (RLO, VKE)

Результат логической операции (англ. RLO, нем. VKE) – это текущее состояние сигнала в процессоре, применяемое для дальнейшей двоичной обработки сигнала. Определенные операции выполняются или не выполняются в зависимости от предшествующего состояния RLO.

Регулирование отношения

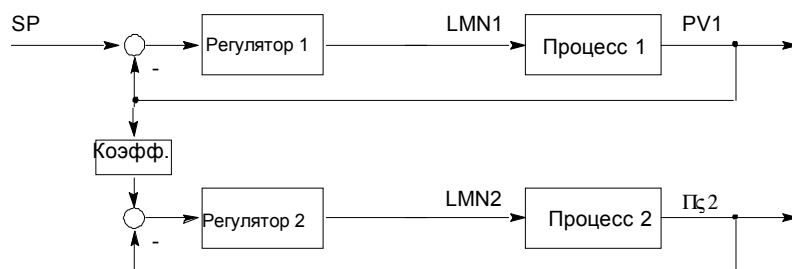
- Одноконтурный регулятор отношения

Одноконтурный регулятор отношения используется тогда, когда отношение двух регулируемых величин важнее, чем их абсолютные значения.



- • Многоконтурный регулятор отношения.

В двухконтурном регуляторе отношения выдерживается постоянным отношение двух регулируемых величин PV1 и PV2. Для этого задающее значение 2-го контура регулирования вычисляется из регулируемой величины 1-го контура. Это отношение поддерживается даже при динамическом изменении регулируемой величины PV1.

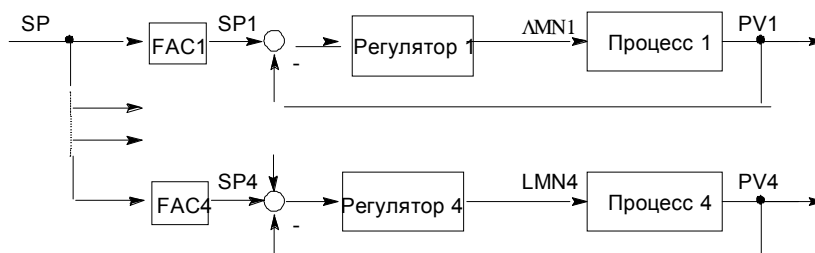


Регулятор

Регулятор – это устройство, в котором непрерывно рассчитывается сигнал ошибки и генерируется управляющее воздействие с целью быстро и без перерегулирования устранить сигнал ошибки.

Регулирование смешивания

Регулирование смешивания - это регулирующая структура, в которой задающее воздействие для общего количества SP преобразуется в процентное содержание отдельных компонентов. Сумма коэффициентов смешивания FAC должна быть равна 1 (=100 %).



Рестарт

При запуске CPU (например, при переводе переключателя режимов работы из STOP в RUN или при включении напряжения сети) перед циклической обработкой программы (OB1) сначала выполняется организационный блок OB100 (полный рестарт) или организационный блок OB101 (рестарт; только у S7-400). При рестарте считывается таблица входов образа процесса, и обработка программы пользователя продолжается с того места, на котором она была прервана при последнем останове (STOP, отключение питания).

С

Связь, двусторонняя

При использовании для обмена данными коммуникационных SFB различают одностороннюю и двустороннюю связь. Связь является двусторонней, если SFB имеется как на локальном, так и на удаленном модуле, например, коммуникационные SFB "USEND" и "URCV".

Связь, односторонняя

При использовании для обмена данными коммуникационных SFB различают одностороннюю и двустороннюю связь. Связь является односторонней, если SFB имеется только на локальном модуле, например, SFB "GET".

Символическое программирование

Язык программирования STEP 7 дает возможность применять символические обозначения вместо адресов STEP 7. Это значит, например, что адрес "Q 1.1" может быть заменен символическим именем "Valve 17".

Так называемый список символов в STEP 7 создает при этом связь между адресом и соответствующим символическим именем.

Системная ошибка

Системные ошибки – это ошибки, которые могут возникать внутри контроллера (т.е. не в процессе). Системными ошибками могут быть, например, программные ошибки в CPU и неисправности в модулях.

Системная функция (SFC)

Системная функция (SFC) – это встроенная в операционную систему CPU → функция, которая при необходимости может быть вызвана в программе пользователя STEP 7.

Системный функциональный блок (SFB)

Системный функциональный блок (SFB) – это встроенный в операционную систему CPU → функциональный блок, который может быть при необходимости вызван в программе пользователя STEP 7.

Список операторов

Список операторов (английское сокращение STL, немецкое сокращение AWL) является языком ассемблера → STEP 7. Если программа написана на STL, то отдельные команды соответствуют рабочим шагам, выполняемым CPU при обработке программы.

Стандартная функция

Стандартные функции – это поставляемые фирмой SIEMENS функции для решения сложных задач.

Стандартный функциональный блок

Стандартные функциональные блоки – это поставляемые фирмой SIEMENS функциональные блоки для решения сложных задач.

Стартовая информация

Когда операционная система вызывает организационный блок, она передает стартовую информацию, которая может быть проанализирована в программе пользователя.

Стартовое событие

Стартовые события – это определенные события, например, ошибки или прерывания, побуждающие операционную систему вызывать подходящий организационный блок.

Ступенчатый регулятор

Ступенчатый регулятор - это квазинепрерывный регулятор с дискретным выходом (и двигательным исполнительным устройством с интегрирующим действием). Исполнительное устройство имеет трехпозиционную характеристику, например, вперед–стоп–назад или открыть –удерживать– закрыть (→ Трехпозиционный регулятор).

Т**Трехпозиционный регулятор**

Регулятор, который может принимать только три дискретных состояния; например, “нагревать - выключено - охлаждать” или “направо - остановка - налево” (→ Ступенчатый регулятор).

У**Удаленное устройство**

Удаленные устройства – это устройства, например, принтеры или компьютеры, доступ к которым осуществляется через сеть. В отличие от локальных устройств, при их установке им должен быть присвоен сетевой адрес.

Ф**Фактические параметры**

Фактические параметры заменяют формальные параметры при вызове функционального блока (FB) или функции (FC), например, формальный параметр “REQ” заменяется фактическим параметром “I 3.6”.

Формальные параметры

Формальный параметр – это метка-заполнитель для фактического параметра в параметризуемых логических блоках. В FB и FC формальные параметры описываются пользователем, в SFB и SFC они уже имеются. При вызове блока формальному параметру ставится в соответствие фактический параметр, так что вызванный блок работает с этим фактическим значением. Формальные параметры относятся к
→ локальным данным блока и делятся на входные, выходные и проходные (in/out) параметры.

Ш**Широтно-импульсная модуляция**

Широтно-импульсная модуляция – это способ влияния на управляющее воздействие при дискретном выходе. Расчетное управляющее воздействие в процентах преобразуется в пропорциональную сигналу длительность импульса (ED), на выходе управляющего воздействия, например, 100 %
 $T_p = T_A$ или = CYCLE.

Я

Язык программирования STEP 7

Язык программирования для контроллеров SIMATIC S7. Программист может применять STEP 7 в различных формах представления: а) список операторов, б) функциональный план, с) контактный план.

О

ОВ запуска

В зависимости от положения переключателя видов запуска (только в S7-400), причины запуска (восстановление исчезнувшего напряжения питания; ручное переключение из STOP в RUN переключателем режимов работы или командой с устройства программирования) операционной системой вызывается организационный блок запуска “Полный рестарт” или “Рестарт” (имеется только в S7-400). В ОВ запуска пользователь SIMATIC S7 может запрограммировать, например, как система будет запускаться снова после исчезновения напряжения.

ОВ ошибок

ОВ ошибок – это организационные блоки, с помощью которых пользователь может запрограммировать реакцию на ошибку. Правда, запрограммированная реакция возможна только тогда, когда ошибка не приводит к остановке контроллера. Соответствующие ОВ ошибок имеются для различных типов ошибок (например, ОВ ошибок адресации, ОВ ошибок доступа в S7).

ОВ1

Организационный блок ОВ1 – это интерфейс пользователя с системной программой для циклической обработки программы.

Р

Р–алгоритм

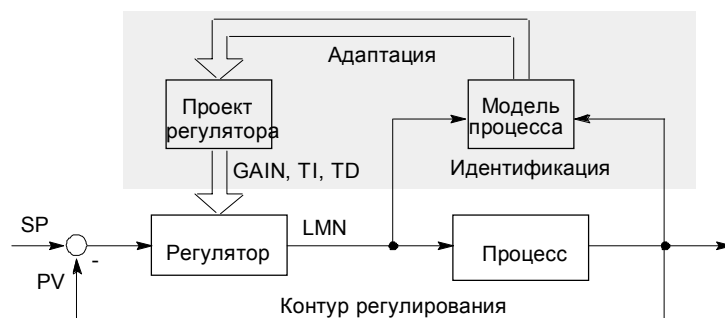
Алгоритм расчета выходного сигнала, в котором имеет место пропорциональная связь между сигналом ошибки и изменением задающего воздействия. Характеристики: установившийся сигнал ошибки, не следует применять на объектах с транспортным запаздыванием.

PI–алгоритм

Алгоритм расчета выходного сигнала, в котором изменение управляющего воздействия составляется из компонента, пропорционального сигналу ошибки, и интегрального компонента, значение которого пропорционально сигналу ошибки и времени. Характеристики: отсутствие установившегося сигнала ошибки, более быстрая компенсация, чем в случае I-алгоритма, пригодность для любых процессов.

PID-алгоритм

Алгоритм расчета выходного сигнала, который образуется путем умножения, интегрирования и дифференцирования из сигнала ошибки. PID-алгоритм – это параллельная структура. Характеристика: высокое качество регулирования достигается, если транспортное запаздывание объекта регулирования не превышает остальных постоянных времени.



S

STEP 7

Программное обеспечение для разработки программ пользователя для контроллеров SIMATIC S7.

STL

→ Список операторов

V

VKE

см. Результат логической операции.

Литература

/30/

Getting Started: Working with STEP 7 V5.0
[Введение: Работа со STEP 7 V5.0]

/70/

Руководство: *S7-300 Programmable Controller*
Hardware and Installation
[Программируемый контроллер S7-300.
Аппаратура и установка]

/71/

Справочное руководство: *S7-300, M7-300 Programmable Controllers*
Module Specifications
[Программируемые контроллеры S7-300, M7-300
Описания модулей]

/72/

Instructions List: S7-300 Programmable Controller
[Список команд: Программируемый контроллер S7-300]

/101/

Справочное руководство: *S7-400, M7-400 Programmable controllers*
Module Specifications
[Программируемые контроллеры S7-400, M7-400
Описания модулей]

/102/

Instructions List: S7-400 Programmable Controller
[Список команд: Программируемый контроллер S7-400]

/231/

Руководство: Configuring Hardware and Communication Connections,
STEP 7 V5.0
[Конфигурирование аппаратуры и коммуникационных связей, STEP
7 V5.0]

/234/

Руководство: Programming with STEP 7 V5.0
[Программирование с помощью STEP 7 V5.0]

/232/

Справочное руководство: Statement List (STL) for S7-300 and S7-400
[Список операторов (STL) для S7-300 и S7-400]

- /233/ Справочное руководство: Ladder Logic (LAD) for S7-300 and S7-400
[Контактный план (LAD) для S7-300 и S7-400]
- /236/ Справочное руководство: Function Block Diagram (FBD) for S7-300
and S7-400 [Функциональный план (FBD) для S7-300 и S7-400]
- /250/ Руководство: Structured Control Language (SCL) for S7-300 and S7-
400 Programming
[Язык структурного управления (SCL) для программирования
S7-300 и S7-400]
- /251/ Руководство: *S7-GRAPH for S7-300 and S7-400*,
Programming Sequential Control Systems
[*S7-GRAPH для S7-300 и S7-400*,
Программирование систем последовательного управления]
- /252/ Руководство: *S7-HiGraph for S7-300 and S7-400*,
Programming State Graphs
[*S7-HiGraph для S7-300 и S7-400*,
Программирование графов состояния]
- /254/ Руководство: *CFC Continuous Functions Charts*
[CFC, *Схемы последовательных функций*]
- /270/ Руководство: *S7-PDIAG for S7-300 and S7-400*
"Configuring Process Diagnostics for LAD, STL, and FBD"
[*S7-PDIAG для S7-300 и S7-400*
"Проектирование диагностики процесса для контактного плана,
списка операторов и функционального плана"]
- /350/ Руководство пользователя: SIMATIC 7, Standard Controller
[SIMATIC7, Стандартный контроллер]

Предметный указатель

А

Аппаратное прерывание 11-2
 потеряно 26-4
Асинхронная ошибка 11-2, 28-6
 блокирование с помощью SFC39
 DIS_IRT 11-4
 задержка с помощью SFC41
 DIS_AIRT 11-7
 разблокирование с помощью SFC40
 EN_IRT 11-6
 разблокирование с помощью SFC42
 EN_AIRT 11-8
Асинхронные ошибки,
 OB80 1-22

Б

Батарея разряжена 26-3
Битовый массив области
 входов/выходов
 сброс с помощью SFC80 13-5
 установка с помощью SFC79 13-4
Блок данных
 создание с помощью SFC22
 CREAT_DB 3-8
 удаление с помощью SFC23 3-10
Блок скомпонованного кода, вызов 25-2

В

Ведомые часы
 синхронизация 5-4
Ведущие часы 5-2
Внешняя ошибка 26-3
Внутренняя ошибка 26-3
Возвращаемое значение
 SFC41 DIS_AIRT 11-7
 SFC42 EN_AIRT 11-8
Временные переменные (TEMP)
 требуемые для OB 1-5, 1-48
Время задержки 9-3
Время суток (TOD) 5-2

Системные и стандартные функции для S7-300 и S7-400
C79000-G7076-C567-01

Встроенная функция измерения частоты
24-3

Встроенная функция счетчика 24-2

Встроенное регулирование

 анализ процесса 22-1

 выбор регулятора 22-2

 применения 22-1

Вызов блока скомпонованного кода 25-2

Выход за пределы диапазона измерений
 аналоговый модуль ввода 26-6

Г

Генератор последовательностей
 реализация 13-6

Главный программный цикл (OB1) 1-4

Группы slave-устройств DP

 синхронизация 15-4

Д

Дата 5-2

Демаскирование

 событий, связанных с ошибками 10-2

Диагностическая информация модуля
 27-33

Диагностические данные 26-2

 сигнальных модулей 7-2, 27-2

 содержимое 26-2

 структура 26-2

 CPU 27-2

Диагностические данные модуля 26-2,
 27-34, 27-35

Диагностические события 28-14

Диагностический буфер 10-2, 27-2, 27-32

Диагностическое прерывание 11-3

 из-за замены 26-3

З

Задержка включения

 генерирование 20-4

Задержка выключения

 генерирование 20-6

Задержка программы пользователя с
 помощью SFC47 WAIT 4-3

Заменяющее значение

 запись в ACCU 1 с помощью SFC44

 REPL_VAL 3-14

Запись данных

 запись 7-2

 запись с помощью SFC58 WR_REC

 7-11

чтение 7-2
чтение с помощью SFC59 RD_REC 7-12
Запись
 непротиворечивых данных в
 стандартное slave-устройство DP
 15-14
Запись данных в удаленный CPU с
 помощью SFB15 17-18
Запуск 1-41

И

Идентификатор (ID) типа модуля 27-4
Изменения режимов работы 28-8
Импульс
 генерирование 20-2
Инициализация области памяти с помощью SFC21 FILL 3-6
Информация об ошибках 2-2
 общая 2-4
 специфическая 2-4
 SFC22 CREAT_DB 3-9
 SFC23 DEL_DB 3-10
 SFC34 QRY_DINT 9-5
 SFC40 EN_IRT 11-6
 SFC42 EN_AIRT 11-8
Информация о состоянии модуля 27-25
Информация пользователя 26-3

К

Канал
 информация 26-3
 ошибка 26-3, 26-4
Класс приоритета 1-1, 1-5, 1-9, 1-11, 1-13,
 1-14, 1-15, 1-17, 1-18, 1-20, 1-22, 1-
 23, 1-24, 1-26, 1-29, 1-30, 1-31, 1-34,
 1-37, 1-40, 1-42, 1-46, 1-48, **10-14**,
 12-4, **27-14, 28-8, 28-9**
типы OB
 OB1 1-4
 OB121 1-45
 OB122 1-48
 OB20 1-10
 OB35 1-12
 OB80 1-22
 OB81 1-24
 OB82 1-26
 OB85 1-31
Коммуникации
 данные о состоянии 27-16

ошибка 11-3
прерывание 11-2
Коммуникационные события 28-12
Коммуникационные SFB См. SFB
Коммуникационные SFC 18-4
Коммуникационные SFC для неспроектированных соединений S7
 классификация 18-4
Контроль времени цикла 26-3
Конфигурирование
 ошибка
 аналоговый модуль ввода 26-6
 аналоговый модуль вывода 26-7
 цифровой модуль ввода 26-8
 цифровой модуль вывода 26-8
Копирование переменных с помощью SFC20 BLKMOV 3-2
Короткое замыкание на землю
 аналоговый модуль ввода 26-6
 аналоговый модуль вывода 26-7
 цифровой модуль ввода 26-8
 цифровой модуль вывода 26-8
Короткое замыкание на фазу
 аналоговый модуль ввода 26-6
 аналоговый модуль вывода 26-7
 цифровой модуль ввода 26-8
 цифровой модуль вывода 26-8

Л

Логический адрес
 канала
 определение 14-2
 модуля
 определение всех адресов 14-6
Локальные данные OB 27-23

М

Маскирование
 ошибок 10-2
Модуль
 идентификатор (ID) типа 27-4
 идентификация 27-6
 неисправность 26-3
Мультипроцессорное прерывание 11-2

Н

Неисправность источника питания 11-3
Неисправность процессора 26-4
Неисправность стойки 1-34, 11-3

Некоординированная передача данных
с помощью SFB8 17-8
Некоординированный прием данных
с помощью SFB9 17-10

О

Области памяти 27-9

Обнаружение ошибок

типы OV

OV1 1-4
OV10 1-6
OV121 1-45
OV122 1-48
OV20 1-10
OV35 1-12
OV80 1-22

Обновление таблицы входов образа
процесса 13-2

Обновление таблицы выходов образа
процесса 13-3

Обработка ошибок 10-2

Обратный счет 20-9

Обрыв провода

аналоговый модуль ввода 26-6
аналоговый модуль вывода 26-7
цифровой модуль ввода 26-8
цифровой модуль вывода 26-8

Оперативная помощь iv

Организационные блоки (OV)

обзор 1-1

OV аппаратных ошибок CPU (OV84)
1-30

OV диагностических прерываний
(OV82) 1-26

OV запуска (OV 100 и 101) 1-41

OV коммуникационных ошибок (OV87)
1-37

OV неисправности стойки OV (OV86)
1-34

OV ошибок класса приоритета (OV85)
1-31

OV полного рестарта (OV100) 1-41

OV прерываний по времени
(OV 10 – 17) 1-6

OV прерываний при установке/
удалении модуля (OV83) 1-28

OV рестарта (OV101) 1-41

Организационный блок (OV) 1-1

фоновый OV (OV90) 1-39

OV мультипроцессорных прерываний
(OV60) 1-16

OV1 1-4

OV121 1-45

OV122 1-48

типы OV

OV20 1-10

OV35 1-12

OV80 1-22

OV81 1-24

OV82 1-26

OV85 1-31

Отказ буферной батареи 26-3

Отказ источника питания 26-3

Отказ стойки расширения 26-4

Отсутствует вспомогательное
напряжение 26-3

Отсутствует напряжение на зажимах
аналоговый модуль вывода 26-7

цифровой модуль вывода 26-8

Отсутствует параметризация 26-3

Отсутствует питание датчика

цифровой модуль ввода 26-8

Отсутствует фронт-штекер 26-3

Ошибка

асинхронная 11-2

АЦП/ЦАП 26-4

доступа 10-2

маскирование 10-2

ОЗУ 26-4

синхронная 10-2

программирования 10-2

СППЗУ 26-4

Ошибка АЦП/ЦАП 26-4

Ошибка времени 11-3

Ошибка выравнивания

при записи 10-9

при чтении 10-9

Ошибка длины области

при записи 10-9

при чтении 10-9

Ошибка доступа 10-2, 10-11

Ошибка доступа к входам/выходам

при записи 10-11

при чтении 10-11

Ошибка заземления на корпус

цифровой модуль ввода 26-7

цифровой модуль вывода 26-7

Ошибка записи

блок данных 10-10

экземплярный блок 10-10

Ошибка назначения параметров

аналоговый модуль ввода 26-6

аналоговый модуль вывода 26-7

цифровой модуль ввода 26-8

цифровой модуль вывода 26-8

Системные и стандартные функции для S7-300 и S7-400

C79000-G7076-C567-01

Индекс-3

Ошибка номера блока 10-10
 Ошибка номера таймера 10-9
 Ошибка области
 при записи 10-9
 при чтении 10-9
 Ошибка ОЗУ 26-4
 Ошибка опорного канала
 аналоговый модуль ввода 26-6
 Ошибка программирования 10-2, **10-9**
 Ошибка программы 11-3
 Ошибка BCD-преобразования 10-9
Ошибки доступа в случае CPU 417 и
CPU 417H 10-11Ф

П

Пакет GD
 извлечение с помощью SFC61 16-4
 передача с помощью SFC60 16-2
 Память пользователя
 сжатие с помощью SFC25 3-12
 Параметр
 запись параметра по умолчанию 7-7
 ACCFLT_ESR 10-14
 ACCFLT_Masked 10-13
 ACCFLT_MASKED 10-12
 ACCFLT_QUERY 10-14
 ACCFLT_RESET_MASK 10-13
 ACCFLT_SET_MASK 10-12
 BLK 3-7
 BUSY с SFC 51 и 55 – 59 2-7
 BVAL 3-7
 CDT 5-3
 COUNT 3-8
 CQ 6-5
 CV 6-5
 DB_NUMBER 3-8
 DSTBLK 3-3
 DTIME 9-4
 EVENTN 12-11
 INFO1 12-11
 INFO2 12-11
 IOID 7-5, 7-7, 7-8, 7-11, 7-13
 LADDR 7-5, 7-7, 7-8, 7-11, 7-13
 LOW_LIMIT 3-8
 MODE 11-4, **11-6**
 NR 6-3, **6-4, 6-5**
 OB_NR 8-5, **8-6, 8-7, 8-8, 9-4, 9-5, 9-6,**
 11-4, 11-6
 PDT 5-2
 PERIOD 8-5
 PRGFLT_ESR 10-14
 PRGFLT_MASKED 10-12, 10-13

PRGFLT_QUERY 10-14
 PRGFLT_RESET_MASK 10-13
 PRGFLT_SET_MASK 10-12
 PV 6-3
 RECNUM 7-5, 7-7, 7-11, 7-13
 RECORD 7-5, 7-11
 REQ с асинхронными SFC 2-7
 RET_VAL с SFC 51 и 55 – 59 2-7
 SDT 8-5
 SEND 12-11
 SRCBLK 3-3
 STATUS 8-8, **9-5**
 UP_LIMIT 3-8
 WT 4-3

Параметры

сигнальных модулей 7-2
 SFC1 READ_CLK 5-3
 SFC58 WR_REC 7-11
SFC64 TIME_TICK 6-6

Передача параметров

с помощью SFC55 WR_PARM 7-5
с помощью SFC56 WR_DPARM 7-7

Передача сегментированных данных
 с помощью SFB12 17-12

Плата памяти 26-3

Повторный запуск контроля времени
цикла с помощью SFC43
RE_TRIGR 4-1

Полный рестарт 1-41
 инициирование на удаленном
 устройстве 17-26

Превышение температуры
 цифровой модуль вывода 26-8

Предохранитель сгорел 26-4

Прерывание 11-2

блокирование с помощью SFC39
DIS_IRT 11-4

деблокирование с помощью SFC40
EN_IRT 11-6

деблокирование с помощью SFC42
EN_AIRT 11-8

задержка с помощью SFC41
DIS_AIRT 11-7

классы 11-2

с задержкой (OB20) 1-10
 циклическое (OB35) 1-12

Прерывание по времени 8-2, 11-2

активизация с помощью SFC30
ACT_TINT 8-7

влияющие ситуации 8-3

исполнение и реакция 8-4

опрос с помощью SFC31 QRY_TINT
8-8

- отмена с помощью SFC29 CAN_TINT 8-6
 - теплый рестарт 8-3
 - условия для вызова 8-2
 - установка с помощью SFC28
 - холодный рестарт 8-3
 - ОВ 8-2
 - SET_TINT 8-5
 - Прерывание по ошибке
 - асинхронной 11-2
 - синхронной 11-2
 - Прерывание при удалении/ вставке модуля 11-3
 - Прерывание с задержкой 9-2, 11-2
 - влияющие ситуации 9-3
 - вызов в ОВ запуска 9-3
 - запрос с помощью SFC34 QRY_DINT 9-5
 - запуск с помощью SFC32 SRT_DINT 9-4
 - отмена с помощью SFC33 CAN_DINT 9-6
 - условия для вызова 9-2
 - Прерывания
 - ОВ аппаратных прерываний 1-14
 - Прерывания по асинхронным ошибкам 1-3
 - Прерывания по синхронным ошибкам 1-3
 - Прерывания с задержкой
 - ОВ20 1-10
 - Прием сегментированных данных с помощью SFB13 17-14
 - Программа
 - циклическая 1-4
 - Программирование
 - типы ОВ
 - ОВ1 1-4
 - ОВ121 1-45
 - ОВ122 1-48
 - ОВ20 1-10
 - ОВ35 1-12
 - ОВ80 1-22
 - ОВ81 1-24
 - ОВ82 1-26
 - ОВ85 1-31
 - Прямой счет 20-8
 - Прямой и обратный счет 20-10
- Р**
- Регистр ошибок 10-2
 - чтение с помощью SFC38 READ_ERR 10-14
- Системные и стандартные функции для S7-300 и S7-400
C79000-G7076-C567-01
- Регулирование
 - непрерывное с помощью SFB41 22-3
 - ступенчатое с помощью SFB42 22-10
 - Рестарт 1-41
 - инициирование на удаленном устройстве 17-30
- С**
- Свободные события пользователя 28-18
 - Синфазная ошибка
 - аналоговый модуль ввода 26-6
 - аналоговый модуль вывода 26-7
 - Синхронизация
 - часов 5-2
 - Синхронные ошибки 10-2, 28-5
 - демаскирование с помощью SFC37 DMSK_FLT 10-13
 - маскирование с помощью SFC36 MSK_FLT 10-12
 - ОВ121 1-45
 - ОВ122 1-48
 - Системные данные 27-2
 - Системные области 27-10
 - Слот модуля
 - опрос логического адреса 14-4
 - Событие 28-2
 - идентификатор (ID) 12-11, 28-2
 - класс 28-2
 - События, связанные со стандартными ОВ 28-4
 - События, связанные с состоянием STOP 28-8
 - События Н/В 28-13
 - Состояние прерывания 27-14
 - Состояние удаленного партнера
 - запрос 17-32
 - получение 17-34
 - Список состояний системы 27-2
 - подписки 27-5
 - Стандартные события пользователя 28-16
 - Счетчик (CPU 312) 24-2
 - Счетчики
 - ошибка номера 10-9
 - Счетчик рабочего времени 6-2
 - диапазон значений 6-2
 - запуск с помощью SFC3 CTRL_RTM 6-4
 - останов с помощью SFC3 CTRL_RTM 6-4

**считывание с помощью SFC4
READ_RTM 6-5**
**установка с помощью SFC2
SET_RTM 6-3**
характеристики 6-2
Считывание
 диагностических данных slave-
 устройства DP 15-9
 непротиворечивых данных slave-
 устройства DP 15-12
 с помощью SFC51 RDSYSST 12-6
**Считывание времени
с помощью SFC1 READ_CLK 5-3**
Считывание данных из удаленного CPU
 с помощью SFB14 17-16
**Считывание системного времени
с помощью SFC64 TIME_TCK 6-6**
**Считывание состояния системы с
помощью SFC51 RDSYSST 12-6**
Считывание стартовой информации OB
 с помощью SFC6 12-3

Т

Таблица описания переменных
 для OB1 1-4
 для OB10 1-6
 для OB121 1-45
 для OB122 1-48
 для OB20 1-10
 для OB35 1-12
 для OB80 1-22
Типы блоков 27-11
Ток ниже нижней границы диапазона
 модуль аналогового ввода 26-6

У

**Управление функционированием в Н-
системах с помощью SFC90 23-1**
**Установка времени
с помощью SFC0 SET_CLK 5-2**

Ф

**Фильтрация
ошибок 10-3**
**Фильтр ошибок
ошибки доступа 10-5, 10-6**
 ошибки программирования 10-4
Фильтр ошибок доступа 10-5
**Фильтр ошибок доступа для CPU 417
и CPU 417H 10-6**

Фильтр ошибок программирования 10-4

Функции IEC
 обзор 21-2
 требования к рабочей и загрузочной
 памяти 21-3

Ц

Циклические прерывания **11-2**
 OB35 1-12

Ч

Частотомер (CPU 312) 24-3
Часы
 ведущие 5-2
 синхронизация 5-2

Ш

Широтно-импульсная модуляция 22-15

А

ACCFLT_ESR 10-14
ACCFLT_MASKED 10-12, 10-13
ACCFLT_QUERY 10-14
ACCFLT_RESET_MASK 10-13
ACCFLT_SET_MASK 10-12
ACT_TINT 8-7
ALARM 19-7
ALARM_8 19-13
ALARM_8P 19-10
ALARM_S 19-25
ALARM_SC 19-28
ALARM_SQ 19-25
AR_SEND 19-15

В

BLK 3-7
BLKMOV 3-2
BRCV 17-14
BSEND 17-12
BVAL 3-7

С

CAN_DINT 9-6
CAN_TINT 8-6
CDT 5-3
COMPRESS 3-12
CONT_C 22-3

Системные и стандартные функции для S7-300 и S7-400
C79000-G7076-C567-01

CONT_S 22-10

CONTROL 17-36

COUNT 3-8

CPU

отказ аппаратуры 11-3

переход в состояние STOP с

помощью SFC46 STP 4-2

характеристики 27-7

CQ 6-5

CREATE_DB 3-8**CTD 20-9****CTRL_RTM 6-4****CTU 20-8****CTUD 20-10**

CV 6-5

D

DB_NUMBER 3-8

DEL_DB 3-10

DIS_AIRT 11-7**DIS_IRT 11-4****DIS_MSG 19-17****DMSK_FLT 10-13****DP_PRAL 15-2**

DPNRM_DG 15-9

DPRD_DAT 15-12

DPSYC_FR 15-4

DPWR_DAT 15-14

DRUM 13-6

DSTBLK 3-3

DTIME 9-4

E**EN_AIRT 11-8****EN_IRT 11-6****EN_MSG 19-19**

EVENTN 12-11

F

FC1 21-7

FC10 21-11

FC11 21-17

FC12 21-9

FC13 21-11

FC14 21-9

FC15 21-12

FC16 21-18

FC17 21-16

FC18 21-10

FC19 21-12

FC2 21-15

FC20 21-14

FC21 21-14

FC22 21-21

FC23 21-10

FC24 21-13

FC25 21-21

FC26 21-15

FC27 21-22

FC28 21-10

FC29 21-13

FC3 21-5

FC30 21-19

FC31 21-17

FC32 21-15

FC33 21-6

FC34 21-8

FC35 21-7

FC36 21-23

FC37 21-20

FC38 21-19

FC39 21-20

FC4 21-16

FC40 21-6

FC5 21-18

FC6 21-5

FC7 21-5

FC8 21-6

FC9 21-9

FILL 3-6**G**

GADR_LGC 14-2

GD_RCV 16-4

GD_SND 16-2

GET 17-16

H**H_CTRL 23-1****I****I_ABORT 18-27****I_GET 18-23****I_PUT 18-25**

INFO1 12-11

INFO2 12-11

I OID 7-5, 7-7, 7-8, 7-11, 7-13

Системные и стандартные функции для S7-300 и S7-400

C79000-G7076-C567-01

Индекс-7

L

LADDR 7-5, 7-7, 7-8, 7-11, 7-13
LGC_GADR 14-4
LOW_LIMIT 3-8

M

MODE 11-4, **11-6**
MP_ALM 4-4
MSK_FLT 10-12

N

NOTIFY 19-5
NR 6-3, **6-4**, **6-5**

O

ОВ аппаратных ошибок CPU 1-30
ОВ аппаратных прерываний 1-14
ОВ ошибок **10-2**
 типы ОВ
 OB1 1-4
 OB20 1-10
 OB35 1-12
 OB80 1-22
 OB81 1-24
 OB82 1-26
 OB85 1-31
ОВ ошибок класса приоритета 1-31
ОВ прерываний
 диагностические прерывания 1-26
 прерывание при установке/снятии модуля 1-28
 прерывания по времени 1-6
OB_NR 8-5, **8-6**, **8-7**, **8-8**, **9-4**, **9-5**, **9-6**,
 11-4, **11-6**
OB1 1-4
OB100, OB101 и OB102 1-41
OB121 1-45, 1-48
OB20 – OB23) 1-10
OB30 – OB38 1-12
OB40 – OB47 1-14
OB60 1-16, **4-4**
OB70 1-18
OB72 1-20
OB80 1-22
OB81 1-24
OB82 1-26
OB83 1-28
OB84 1-30
OB85 1-31
OB86 1-34

OB87 1-37

P

PARM_MOD 7-8
PDT 5-2
PERIOD 8-5
PRGFLT_ESR 10-14
PRGFLT_MASKED 10-12, 10-13
PRGFLT_QUERY 10-14
PRGFLT_RESET_MASK 10-13
PRGFLT_SET_MASK 10-12
PRINT 17-20
PULSEGEN 22-15
PUT 17-18
PV 6-3

Q

QRY_DINT 9-5
QRY_TINT 8-8

R

RD_LGADR 14-6
RD_REC 7-12
RD_SINFO 12-3
RDSYSST 12-6, **27-2**
RE_TRIGR 4-1
READ_CLK 5-3
READ_ERR 10-14
READ_RTM 6-5
RECNUM 7-5, 7-7, 7-11, 7-13
RECORD 7-5, 7-11
REPL_VAL 3-14
RESUME 17-30

S

S 6-4
SDT 8-5
SEND 12-11
SET 13-4, 13-5
SET_CLK 5-2
SET_RTM 6-3
SET_TINT 8-5
SFB
 классификация 17-2
 классификация параметров 17-4
 опрос состояния соединения,
 относящегося к экземпляру SFB
 17-36
 реакция на запуск 17-38

- реакции на ошибки и неисправности 17-40
- SFB0 CTU 20-8**
- SFB1 CTD 20-9**
- SFB12 BSEND 17-12
- SFB13 BRCV 17-14
- SFB14 GET 17-16
- SFB15 PUT 17-18
- SFB16 PRINT 17-20**
- SFB19 START 17-26
- SFB2 CTUD 20-10**
- SFB20 STOP 17-28
- SFB21 RESUME 17-30
- SFB22 STATUS 17-32
- SFB23 USTATUS 17-34
- SFB3 TP 20-2**
- SFB32 DRUM 13-6**
- SFB33 ALARM 19-7**
- SFB34 ALARM_8 19-13**
- SFB35 ALARM_8P 19-10**
- SFB36 NOTIFY 19-5**
- SFB37 AR_SEND 19-15**
- SFB4 TON 20-4**
- SFB41 CONT_C 22-3
- SFB42 CONT_S 22-10
- SFB43 PULSEGEN 22-15
- автоматическая синхронизация 22-17
- двухпозиционное регулирование 22-20
- трехпозиционное регулирование 22-18
- трехпозиционное регулирование, асимметричное 22-19
- SFB5 TOF 20-6**
- SFB8 USEND 17-8
- SFB9 URCV 17-10
- SFC0 SET_CLK 5-2**
- SFC1 READ_CLK 5-3**
- параметры 5-3
- SFC10 DIS_MSG 19-17**
- SFC11 DPSYC_FR 15-4
- SFC13 DPNRM_DG 15-9
- SFC14 DPRD_DAT 15-12
- SFC15 DPWR_DAT 15-14
- SFC17 ALARM_SQ 19-25**
- SFC18 ALARM_S 19-25**
- SFC19 ALARM_SC 19-28**
- SFC2 SET_RTM 6-3**
- SFC20 BLKMOV 3-2**
- SFC21 FILL 3-6**
- SFC22 CREAT_DB**
- информация об ошибках 3-9
- SFC22 CREATE_DB 3-8**
- SFC23 DEL_DB 3-10**
- информация об ошибках 3-10
- SFC25 COMPRESS 3-12
- SFC26 UPDAT_PI 13-2
- SFC27 UPDAT_PO 13-3
- SFC28 SET_TINT 8-5**
- SFC29 CAN_TINT 8-6**
- SFC3 CTRL_RTM 6-4**
- SFC30 ACT_TINT 8-7**
- SFC31 QRY_TINT 8-8**
- SFC32 SRT_DINT 9-4**
- SFC33 CAN_DINT 9-6**
- SFC34 QRY_DINT 9-5**
- информация об ошибках 9-5
- SFC35 MP_ALM 4-4**
- SFC36 MSK_FLT 10-12**
- SFC37 DMSK_FLT 10-13**
- SFC38 READ_ERR 10-14**
- SFC39 DIS_IRT 11-4**
- SFC4 READ_RTM 6-5**
- SFC40 EN_IRT 11-6**
- информация об ошибках 11-6
- SFC41 DIS_AIRT 11-7**
- возвращаемое значение 11-7
- SFC42 EN_AIRT 11-8**
- возвращаемое значение 11-8
- информация об ошибках 11-8
- SFC43 RE_TRIGR 4-1**
- SFC44 REPL_VAL 3-14**
- SFC46 STP 4-2**
- SFC46 TIME_TCK 6-6**
- SFC47 WAIT 4-3**
- SFC48 SNC_RTCB 5-4
- SFC49 LGC_GADR 14-4
- SFC5 GADR_LGC 14-2
- SFC50 RD_LGADR 14-6
- SFC51 RDSYSST 12-6, 27-2**
- SFC52 WR_USRMSG 12-10**
- SFC55 WR_PARM 7-5**
- SFC56 WR_DPARM 7-7**
- SFC57 PARM_MOD 7-8
- SFC58 WR_REC 7-11**
- параметры 7-11
- SFC59 RD_REC 7-12
- SFC6 RD_SINFO 12-3
- SFC60 GD_SND 16-2
- SFC61 GD_RCV 16-4
- SFC62 CONTROL 17-36
- SFC64 TIME_TICK**
- параметры 6-6
- SFC65 X_SEND 18-12**
- SFC66 X_RCV 18-14**
- SFC67 X_GET 18-18**

SFC68 X_PUT 18-20
SFC69 X_ABORT 18-22
SFC7 DP_PRL 15-2
SFC72 I_GET 18-23
SFC73 I_PUT 18-25
SFC74 I_ABORT 18-27
SFC79 SET 13-4, 13-5
SFC9 EN_MSG 19-19
SFC90 H_CTRL 23-1
SIGN 9-4
SNC_RTCB 5-4
SRCBLK 3-3
SRT_DINT 9-4
SSL_HEADER 12-7
SSL-ID 27-4
START 17-26
STATUS 8-8, **9-5**, 17-32
STEP 7
 типы OB
 OB1 1-4
 OB10 1-6
 OB121 1-45
 OB122 1-48
 OB20 1-10
 OB35 1-12
 OB80 1-22
 OB81 1-24
 OB82 1-26
 OB85 1-31
STOP 17-28
 перевод удаленного устройства 17-28
STP 4-2

T
TIME_TCK 6-6
TOF 20-6
TON 20-4
TP 20-2

U
UP_LIMIT 3-8
UPDAT_PI 13-2
UPDAT_PO 13-3
URCV 17-10
USEND 17-8
USTATUS 17-34

W
WAIT 4-3
WR_DPARM 7-7
WR_PARM 7-5
WR_REC 7-11
WR_USRMSG 12-10
WT 4-3

X
X_ABORT 18-22
X_GET 18-18
X_PUT 18-20
X_RCV 18-14
X_SEND 18-12