

## 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)

---

#### Примечание

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) посредством значения "0" в бите двоичного результата (BR) слова состояния показывает, что при исполнении функции произошла ошибка. Некоторые системные функции предоставляют в распоряжение дополнительный код ошибки в выходном значении, 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 имеет тип 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:

- код общей 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. Таким образом Вы можете препятствовать возникновению последующих ошибок, являющихся результатом первой ошибки.

## Информация об общих и специфических ошибках

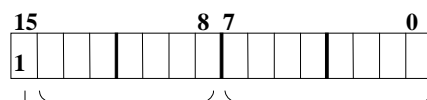
Возвращаемое значение (RET\_VAL) системной функции обеспечивает один из двух следующих типов кодов ошибки:

- Код общей ошибки, относящийся к ошибкам, которые могут возникнуть в любой системной функции.
- Код специфической ошибки, который относится только к конкретной системной функции.

Хотя тип данных выходного параметра RET\_VAL является целым (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 преобразуйте код ошибки в шестнадцатеричный формат.

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

Код ошибки, например, W#16#8081



Номер события или класс ошибки и одиночная ошибка

**x = '0'** указывает, что это **код специфической ошибки SFC**. Коды ошибок перечислены в разделе, озаглавленном "Информация об ошибках", в описаниях отдельных системных функций.

**x > '0'** указывает, что это **код общей ошибки SFC**. В этом случае x – это номер параметра SFC, вызвавшего ошибку. Возможные коды ошибок показаны в следующей таблице.

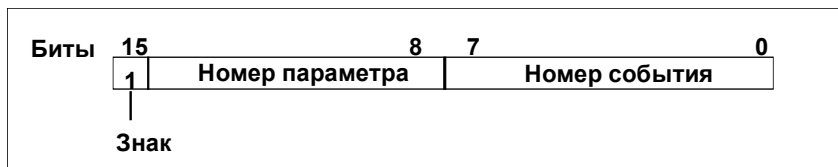
Если знаковый бит = 1, то это указывает, что произошла ошибка.

## Информация об общих ошибках

Код общей ошибки отображает ошибки, которые могут встречаться во всех системных функциях. Код общей 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 0 до 127. Номер события указывает на то, что произошла синхронная ошибка.

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



### Примечание

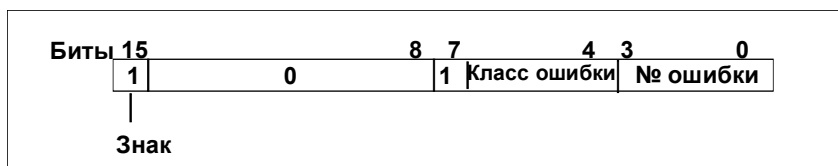
Если в RET\_VAL был записан код общей ошибки, то возможны следующие ситуации:

- 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 может больше отображаться.

## Информация о специфических ошибках

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

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



### Общие коды ошибок

Следующая таблица объясняет коды общих ошибок для 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.
8x01	Неверный синтаксис в ID в параметре ANY
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 для асинхронных SFC

### Асинхронные SFC

Асинхронно работающие SFC – это SFC-функции, которые могут вызваться вновь до того, как завершится выполнение функций предыдущего вызова. Следующие SFC всегда выполняются асинхронно или в определенных ситуациях:

- SFC 7 "DP\_PRAL"
- SFC 11 "DPSYC\_FR"
- SFC 12 "D\_ACT\_DP"
- SFC 13 "DPNRM\_DG"
- SFC 51 "RDSYSST"
- SFC 55 "WR\_PARM"
- SFC 56 "WR\_DPARM"
- SFC 57 "PARM\_MOD"
- SFC 58 "WR\_REC"
- SFC 59 "RD\_REC"
- SFC 65 "X\_SEND"
- SFC 67 "X\_GET"
- SFC 68 "X\_PUT"
- SFC 69 "X\_ABORT"
- SFC 72 "I\_GET"
- SFC 73 "I\_PUT"
- SFC 74 "I\_ABORT"
- SFC 82 "CREA\_DBL"
- SFC 83 "READ\_DBL"
- SFC 84 "WRIT\_DBL"
- SFC 90 "H\_CTRL"
- SFC 102 "RD\_DPARA"
- SFC 103 "DP\_TOPOL"
- SFC 114 "PN\_DP"

## Идентификация задания

Если Вы запускаете аппаратное прерывание, выдаете команды управления для ведомых DP-устройств, запускаете передачу данных или прерываете неконфигурированное соединение с помощью одной из вышеперечисленных SFC, а затем вновь 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 функций SFC. Если эти параметры совпадают с параметрами еще не завершеного задания, то этот вызов SFC считается продолжающимся (follow-on) вызовом.

SFC	Задание идентифицируется посредством...
7 "DP_PRAL"	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
82 "CREA_DBL"	LOW_LIMIT, UP_LIMIT, COUNT, ATTRIB, SRCBLK
83 "READ_DBL"	SRCBLK, DSTBLK
84 "WRIT_DBL"	SRCBLK, DSTBLK
90 "H_CTRL"	MODE, SUBMODE
102 "RD_DPARA"	LADDR, RECNUM
103 "DP_TOPOL"	DP_ID
114 "PN_DP"	-

## 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 вновь, чтобы выполнить то же самое задание (например, в ОБ 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.

Учтите также примечание в пункте Проверка ошибок, касающееся параметра RET\_VAL

- 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 параметр BUSY.

Если требуемые системные в данный момент используются или входные параметры содержат ошибки, то в RET\_VAL вносится соответствующий код ошибки, а BUSY сбрасывается и имеет значение 0.

- 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 SFC 13 "DPNRM\_DG," SFC 67 "X\_GET" и SFC 72 "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.
  - Если не произошло ошибки, то с помощью SFC 59 "RD\_REC" в RET\_VAL вводится размер записи данных в байтах или 0 (обратитесь к пункту Считывание записей с помощью SFC 59 "RD\_REC"). В этом случае BUSY имеет значение 0.  
Если происходит ошибка, то в RET\_VAL вводится код ошибки, а BUSY имеет значение 0.
  - Для 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.

---

### Примечание

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.

---

### Примечание

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

---

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: SFC 59 "RD_REC", если целевая область больше, чем передаваемая запись данных, и SFC 13 "DPNRM_DG," SFC67 "X_GET" и SFC 72 "I_GET"), если не было ошибок	0
			Код ошибки, если были ошибки	0