

DIGSI

CFC

Руководство по эксплуатации

Предисловие, Содержание

Обзор программного продукта

1

Начало работы

2

Практические примеры

3

Блоки CFC

4

Список литературы, Словарь,
Предметный указатель

Версия: 11.06.07

E50417-H1156-C098-A1

Меры безопасности

Данное руководство по эксплуатации не является законченным каталогом всех мер по обеспечению безопасности при работе с рассматриваемым оборудованием (блоком, устройством), так как специальные условия эксплуатации могут потребовать дополнительных мер безопасности. Однако здесь содержатся инструкции, которые должны быть вами соблюдены для обеспечения собственной безопасности и избежания повреждения оборудования. Эти инструкции отмечены треугольником и различными ключевыми словами, указывающими на различные степени опасности:



Предупреждение

Указывает на то, что несоблюдение соответствующих мер предосторожности может привести к смертельному исходу, серьезным телесным повреждениям или значительному материальному ущербу.

Внимание

Указывает на то, что несоблюдение соответствующих мер предосторожности может привести к незначительным телесным повреждениям или материальному ущербу.



Квалифицированный персонал

Вводить в эксплуатацию и обслуживать оборудование (блок, устройство), описанное в данном руководстве, может только квалифицированный персонал. Применительно к данной инструкции квалифицированный персонал - это лица, которые имеют право производить монтаж, вводить устройства в эксплуатацию, заземлять и обслуживать устройства, системы и электрические схемы в соответствии с правилами техники безопасности.

Предписание

Оборудование (устройство, блок) не должно использоваться для каких-либо других целей, кроме тех, которые описаны в Каталоге и Техническом описании. Если оно используется вместе с устройствами и компонентами третьих фирм-производителей, они должны быть рекомендованы или одобрены фирмой Siemens.

Правильная и надежная работа данного программного продукта требует правильной транспортировки, хранения, монтажа и установки, а так же соответствующей эксплуатации и обслуживания.

Во время работы электрооборудования, на некоторых частях неизбежно возникают опасно высокие напряжения. Несоблюдение приведенных ниже мер предосторожности может привести к серьезным телесным повреждениям или значительному материальному ущербу.

- Перед любым соединением устройства, его сначала необходимо заземлить на терминал PE.
- Опасно высокие напряжения могут возникнуть на всех коммутационных компонентах, связанных с источником питания.
- Даже после отключения напряжения питания, опасно высокие напряжения могут все еще присутствовать в оборудовании (заряд конденсатора).
- Устройства с цепями трансформаторов тока не должны использоваться в открытом виде.

Предельные значения, указанные в руководстве или инструкциях по эксплуатации, не должны превышать; это также относится к проверке и вводу в эксплуатацию.

Ограничение ответственности

Содержание данного документа было проверено на предмет соответствия описываемому программному и аппаратному обеспечению. Однако, не исключены расхождения, так что мы не гарантируем полное совпадение.

Текст данного руководства регулярно корректируется и необходимые исправления вносятся в следующие издания. Мы будем признательны за ваши предложения по усовершенствованию данного руководства.

Мы оставляем за собой право вносить технические изменения без дополнительного уведомления.
Версия документа 4.00.04

Copyright

Авторские права Siemens AG 2007 Все права защищены. Передача или тиражирование этого руководства, использование или сообщение его содержания без однозначного разрешения фирмы Siemens запрещено. Нарушение данного условия влечет за собой возмещение ущерба. Все права защищены, в особенности в отношении патентов и регистрации торговых марок.
Зарегистрированные торговые марки
DIGSI® является зарегистрированной торговой маркой SIEMENS AG. Другие обозначения в этом руководстве могут быть марками, использование которых третьими лицами в своих целях может нарушить права владельца.

Предисловие

Цель руководства	Данное руководство содержит: <ul style="list-style-type: none">• информацию о работе с DIGSI CFC,• практические примеры и• подробную информацию о поставленных с программой DIGSI, блоках CFC.
Целевая группа	Данное руководство предназначено для наладчиков, программистов и пользователей программы DIGSI. Необходимы основные навыки работы с DIGSI.
Область применения руководства	Данное руководство предназначено для версии программы DIGSI V4.5 и выше.
Стандарты	Программа DIGSI была разработана в соответствии со стандартом ISO 9001.
Дальнейшая поддержка	Если у вас появились вопросы по DIGSI, пожалуйста, обратитесь в ваше местное представительство фирмы Siemens или свяжитесь с нашей горячей линией поддержки.
Горячая линия	<input type="checkbox"/> Телефон: 01 80 - 5 24 70 00 <input type="checkbox"/> Факс: 01 80 - 5 24 24 71 <input type="checkbox"/> Электронная почта: support@ptd.siemens.de
Учебные курсы	Если вас интересует наша индивидуальная программа обучения, пожалуйста, обратитесь в наш учебный центр по адресу: Siemens AG POWER TRANSMISSION and DISTRIBUTION PTD SE CS Humboldtstr. 59 90459 Nuremberg Телефон: 09 11/4 33-70 05 Факс: 09 11/4 33-79 29 Электронная почта: www.ptd-training.de



Примечание:

Рисунки или снимки экранов, содержащиеся в данном руководстве, были созданы, используя различные операционные системы. Поэтому отображение на вашем компьютере **может** отличаться от рисунков, представленных в данном руководстве.

Данная оговорка **не распространяется** на само содержание этих рисунков.

Содержание

1	Обзор программного продукта	1
2	Начало работы	3
2.1	Программирование устройств SIPROTEC	4
2.2	Классы приоритетов	6
2.3	Рекомендации по программированию	9
2.3.1	Последовательность выполнения при программировании в CFC	9
2.3.2	Разбивка и соединение информационных элементов	9
2.3.3	Максимально допустимое количество блоков	10
2.4	Выбор информационных элементов для CFC	11
2.5	Пример быстрой логики Fast PLC	14
2.5.1	Создание новой схемы CFC	15
2.5.2	Определение класса приоритетов	19
2.5.3	Размещение блока	21
2.5.4	Соединение входного сигнала	24
2.5.5	Соединение выходных сигналов	26
2.5.6	Компиляция схемы CFC	29
2.6	Пример медленной логики Slow PLC	30
2.6.1	Добавление новой информации	31
2.6.2	Ранжирование информации на светодиод	33
2.6.3	Соединение входного сигнала	34
2.6.4	Параметрирование блока	35
2.6.5	Соединение выходного сигнала	36

2.7	Пример обработки измеряемых величин	37
2.7.1	Соединение входных сигналов	38
2.7.2	Соединение блоков	39
2.7.3	Соединение выходных сигналов	40
2.8	Пример блокировки	41
2.8.1	Соединение входных сигналов	43
2.8.2	Увеличение количества входов блока	44
2.8.3	Соединение блоков	46
2.8.4	Соединение выходных сигналов	47
3	Практические примеры	49
3.1	Опция переключения набора уставок	50
3.2	Мигающие светодиоды	62
3.3	Обратная блокировка	68
3.3.1	Заданное действие блокировки при коротком замыкании	68
3.3.2	Варианты выполнения обратной блокировки	69
3.3.3	Обратная блокировка с помощью протокола IEC 61850	70
3.3.4	Схема: обратная блокировка в виде схемы CFC	71
3.4	Подсчёт циклов переключения	72
4	Блоки CFC	73
4.1	Типы данных	74
4.2	Arithmetic (Арифметика)	77
4.2.1	ABSVALUE	78
4.2.2	ADD	79
4.2.3	DIV	80
4.2.4	MUL	81
4.2.5	SQUARE_ROOT	82
4.2.6	SUB	83

4.3	Base Logic (Основная логика)	84
4.3.1	AND	85
4.3.2	CONNECT (Соединение)	87
4.3.3	DYN_OR	89
4.3.4	NAND	91
4.3.5	NEG	93
4.3.6	NOR	94
4.3.7	OR	96
4.3.8	RISE_DETECT	98
4.3.9	X_OR	99
4.4	Status of Information (Состояние информации)	101
4.4.1	CV_GET_STATUS	104
4.4.2	DI_GET_STATUS	106
4.4.3	DI_SET_STATUS	107
4.4.4	MV_GET_STATUS	109
4.4.5	MV_SET_STATUS	110
4.4.6	SI_GET_STATUS	112
4.4.7	SI_SET_STATUS	114
4.4.8	ST_AND	116
4.4.9	ST_NOT	118
4.4.10	ST_OR	119
4.5	Memory (Память)	121
4.5.1	D_FF	122
4.5.2	D_FF_MEMO	125
4.5.3	RS_FF	127
4.5.4	RS_FF_MEMO	129
4.5.5	SR_FF	131
4.5.6	SR_FF_MEMO	133
4.5.7	Memory (Память)	135

4.6	Command Module (Команды управления)	137
4.6.1	BOOL_TO_CO	138
4.6.2	BOOL_TO_IC	143
4.6.3	BOOL_TO_IE	146
4.6.4	CMD_CANCEL	147
4.6.5	CMD_CHAIN	149
4.6.6	CMD_INF	155
4.6.7	LOOP	157
4.7	Type Convertor (Преобразователи типа)	159
4.7.1	BOOL_TO_DI	160
4.7.2	BUILD_DI	162
4.7.3	DI_TO_BOOL	165
4.7.4	DINT_TO_REAL	169
4.7.5	DIST_DECODE	170
4.7.6	DM_DECODE	173
4.7.7	REAL_TO_DINT	175
4.7.8	REAL_TO_INT	177
4.7.9	REAL_TO_UINT	178
4.7.10	INT_TO_REAL	180
4.7.11	UINT_TO_REAL	181
4.8	Comparison (Сравнение)	182
4.8.1	COMPARE (СПРАВНЕНИЕ)	183
4.8.2	LIVE_ZERO	186
4.8.3	LOWER_SETPOINT	189
4.8.4	UPPER_SETPOINT	192
4.8.5	ZERO_POINT	194
4.9	Counter Value (Счетно-импульсная величина)	196
4.9.1	COUNTER (Счетчик)	196

4.10	Time & Clock (Время и тактовые импульсы)	199
4.10.1	ALARM (Аварийная сигнализация)	200
4.10.2	BLINK (Мигание)	202
4.10.3	LONG_TIMER	204
4.10.4	TIMER (Таймер)	206
4.10.5	TIMER_SHORT	211

Список литературы**Словарь****Предметный указатель**

Обзор программного продукта

DIGSI CFC

DIGSI CFC (Continuous Function Chart - непрерывная функциональная схема) является программным компонентом DIGSI 4 для программирования устройств SIPROTEC.

При помощи графического интерфейса пользователя вы соединяете информационные элементы, а также программируете, например, блокировки коммутационного оборудования и последовательности переключений. Кроме того, вы можете изменять измеряемые величины и генерировать сообщения.

Типы информации

Наиболее часто используемыми типами информации являются:

- положение коммутационных устройств и других элементов процесса,
- измеряемые величины,
- двоичная информация о состоянии присоединения и устройств,
- информация защиты,
- общие сообщения и
- аварийные сигналы.

Необходимые вам информационные элементы ранжируются в матрице конфигурации программы DIGSI 4 на DIGSI CFC.

Соединение информационных элементов

Информационные элементы соединяются в DIGSI CFC при помощи имеющихся в программе блоков CFC, таких, например, как:

- Arithmetic (арифметика),
- Base Logic (основная логика),
- Status of Information (состояние информации),
- Memory (память),
- Command Module (команды управления),
- Type Converter (преобразователь типа),
- Comparison (сравнение),
- Counter Value (счетно-импульсная величина),
- Time & Clock (время и тактовые импульсы).

**Работа с
DIGSI CFC**

В DIGSI CFC вы создаете необходимую технологическую цепочку программы в виде схемы CFC.

В схеме CFC вы соединяете, подготовленную в матрице ранжирования DIGSI 4 информацию, с блоками CFC.

В заключении вы компилируете окончательную схему CFC с помощью программы DIGSI CFC в исполняемый модуль программы.

Начало работы

Обзор В этой главе вы познакомитесь с программой DIGSI CFC и научитесь ее использовать.



Примечание:

Описанные ниже главы основаны одна на другой, и каждая из них содержит важную информацию о работе с DIGSI CFC. Проработайте каждую из глав по порядку. Обратите внимание на общую информацию по каждому отдельному примеру.



Примечание:

Для отображения блоков CFC, приведенных в качестве примеров в данной главе, в настройке **Block width (Ширина блока)** была выбрана опция **Wide (широкий)**:

- В редакторе CFC нажмите на **Options (Параметры)** → **Customize (Настроить)** → **Block / Sheet Bar Settings (Панели блока / бланка)**.
- В открывшемся окне отметьте опцию **Wide (широкий)** в настройке **Block width (Ширина блока)** и подтвердите ваш выбор, нажав на **ОК**.

Содержание

2.1	Программирование устройств SIPROTEC	4
2.2	Классы приоритетов	6
2.3	Рекомендации по программированию	9
2.4	Выбор информационных элементов для CFC	11
2.5	Пример быстрой логики Fast PLC	14
2.6	Пример медленной логики Slow PLC	30
2.7	Пример обработки измеряемых величин	37
2.8	Пример блокировки	41

2.1 Программирование устройств SIPROTEC

В устройствах SIPROTEC вы можете программировать ваши собственные функции автоматизации. Эти функции программируются с помощью DIGSI CFC.

Выбор информационных элементов

В матрице ранжирования DIGSI вы выбираете информационные элементы, которые вы желаете использовать для программирования соответствующей функции.

DIGSI CFC

В DIGSI CFC вы создаете необходимую технологическую цепочку функции в виде схемы CFC.

Класс приоритетов

В зависимости от необходимого приоритета обработки (например, функция защиты) и времени обработки (например, циклически), вы определяете тот или иной класс приоритетов для каждой схемы CFC.

Схема CFC

В схеме CFC вы соединяете информационные элементы с помощью блоков CFC.

Набор уставок

После компиляции схемы CFC и выхода из программы DIGSI CFC, вы можете загрузить созданную вами программу с соответствующим набором уставок в устройство SIPROTEC.



Примечание:

В приведенных ниже инструкциях, выполняемая вами работа, описывается, как правило, на основе использования команд меню.

Большинство команд меню могут быть вызваны путем нажатия на соответствующие значки, расположенные на панели инструментов. Для скрытия или отображения панели инструментов, нажмите на пункт меню **View (Вид)** → **Toolbar (Панель инструментов)**.

Кроме того, для многих функций вы можете использовать так называемые горячие клавиши. Для этого обратите внимание на информацию, появляющуюся для выбранных команд меню.

Полезная информация может быть также найдена в строке состояния, которую вы можете скрыть или показать, выбрав пункт меню **View (Вид)** → **Status Bar (Строка состояния)**. Здесь вы найдете следующее:

- информацию о действиях пользователя, производимых в настоящее время,
 - состояние фиксируемых кнопок,
 - номер текущего листа в схеме CFC и
 - текущий класс приоритетов (например, **PLC_BEARB [Fast PLC] (Быстрая логика)**).
-

2.2 Классы приоритетов

Каждая функция, программируемая вами в DIGSI CFC, должна быть назначена определенному классу приоритетов. Классы приоритетов отличаются друг от друга по приоритету обработки и времени выполнения:

- Быстрая логика PLC (**PLC_BEARB / PLC**)
- Медленная логика PLC (**PLC1_BEARB / PLC1**)
- Обработка измеряемых величин (**MW_BEARB / MEASURE**)
- Блокировка (**SFS_BEARB / INTERLOCK**)

Приоритет обработки

Приоритет обработки также определяет тип функций, которые вы можете назначить определенному классу приоритетов:

Таблица 2-1 Классы приоритетов, приоритет обработки и назначенные функции в DIGSI CFC

Класс приоритетов	Приоритет обработки	Назначенные функции
Fast PLC (быстрая логика) (PLC_BEARB)	Функции данного класса приоритетов обрабатываются с самым высоким приоритетом и управляются событием: Каждое изменение логического входного сигнала обрабатывается немедленно. Обработка функции этого класса приоритетов может прервать обработку функции класса приоритетов Slow PLC (медленная логика PLC) (PLC1_BEARB).	Функции защиты, блокировка функций защиты Примечание: В этом классе приоритетов вы можете использовать меньшее количество блоков, чем в классе приоритетов Slow PLC (медленная логика PLC) (PLC1_BEARB). Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.
Slow PLC (медленная логика) (PLC1_BEARB)	Функции данного класса приоритетов управляются событием и обрабатываются с более низким приоритетом, чем функции класса приоритетов Fast PLC (быстрая логика PLC) (PLC_BEARB): Каждое изменение логического входного сигнала обрабатывается немедленно. Однако эта обработка может быть прервана обработкой функции класса приоритетов Fast PLC (быстрая логика PLC) (PLC_BEARB).	Функции управления: управление с использованием синхронизма, управление с функциональных кнопок Примечание: Используйте данный класс приоритетов преимущественно для логических функций, которые не являются функциями защиты.
Обработка измеряемых величин (MW_BEARB)	Функции данного класса приоритетов обрабатываются циклически в фоновом режиме. Примечание: Обработка этих функций не управляется событием.	Обработка измеряемых величин: например, вычисление коэффициента мощности (ANSI 55), вычисление фиксируемой мощности (ANSI 32)
Блокировка (SFS_BEARB)	Функции данного класса приоритетов включаются командой управления и также обрабатываются дополнительно циклически в фоновом режиме. При срабатывании защиты, обработка производится менее часто.	Блокировка коммутационного оборудования:



Примечание:

Некоторые блоки CFC (например, TIMER) могут быть использованы только в предусмотренных для них классах приоритетов. Пожалуйста, обратитесь к соответствующей информации Гл. 4 данного руководства.



Примечание:

Одному классу приоритетов могут соответствовать несколько схем CFC. Системные программно-аппаратные средства устройств SIPROTEC обрабатывают сначала до конца одну схему перед тем, как приступить к обработке следующей схемы того же класса приоритетов.



Примечание:

При использовании циклических уровней, обратите внимание на обработку измеряемых величин (**MW_BEARB**) и блокировку (**SFS_BEARB**):
Для того чтобы обеспечить надежную обработку, входные сигналы должны быть активны по крайней мере, до тех пор, пока не истекло время цикла соответствующего уровня. Изменение входного сигнала **не** запускает процесс обработки схемы.



Примечание:

Порядок обработки схем определенного класса приоритетов во время циклического запуска является случайным и не может быть предсказан заранее.

2.3 Рекомендации по программированию

Программируя функции при помощи DIGSI CFC, вы должны придерживаться ряда рекомендаций во избежание проблем с обработкой программы.

2.3.1 Последовательность выполнения при программировании в CFC

Последовательно выполните следующие восемь шагов при программировании функций с помощью DIGSI CFC:

1. Заранжируйте информацию на CFC.
2. Сохраните матрицу ранжирования.
3. Вставьте новую схему CFC.
4. Определите класс приоритетов.
5. Нарисуйте схему.
6. Проверьте последовательность выполнения задач.
7. Откомпилируйте схему.
8. Сохраните набор уставок.

2.3.2 Разбивка и соединение информационных элементов

Описание

Используя ранжирование DIGSI, вы можете разбить один информационный элемент на несколько сообщений. Если позже вы соедините эти сообщения, работая со схемой и блоком CFC, то это может привести к проблемам при обработке программы (например, блок CFC может быть запущен, хотя он еще не полностью инициализирован).



Примечание:

Проследите за тем, чтобы вы **не разбивали** информационный элемент на несколько сообщений в ранжировании DIGSI, а потом вновь соединяли эти сообщения, работая со схемой и блоком CFC.

Решение

Если для программирования определенных функций вам необходимо сначала разбить информационный элемент на несколько сообщений и позже соединить их, работая со схемой и блоком CFC, то для этого вам следует использовать специальные блоки CFC в схеме CFC.

2.3.3 Максимально допустимое количество блоков

Описание Максимально допустимое количество блоков в отдельных классах приоритетов схем CFC устройства SIPROTEC зависит от производительности вычислительных устройств и контролируется компилятором CFC. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.



Примечание:

Максимально допустимое количество блоков **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** и **COUNTER** зависит от имеющейся энергонезависимой памяти и контролируется компилятором CFC.

Максимально допустимое количество проверяется во время компиляции схемы CFC. При наличии ошибки, система указывает на противоречия. Если расходуется большее количество ресурсов, чем предусмотрено, то это указывается в отображаемом на экране протоколе компиляции.



Примечание:

Следующее относится к устройствам SIPROTEC с версией **ниже, чем V4.5**:

Максимально допустимое количество блоков **CMD_CHAIN** в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**) и медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) составляет **20** блоков.



Примечание:

Максимально допустимое количество блоков **TIMER** и **TIMER_SHORT** ограничено имеющимися в распоряжении системы таймерами и контролируется компилятором CFC.

Максимально допустимое количество проверяется во время компиляции схемы CFC. При наличии ошибки, система указывает на противоречия. Если расходуется большее количество ресурсов, чем предусмотрено, то это указывается в отображаемом на экране протоколе компиляции.

- Информация, заранжированная на CFC, отмечена в столбце **Л** значком **X**.
- Для того чтобы использовать информацию в качестве выходного сигнала из CFC на правой границе схемы, заранжируйте информацию на CFC в столбце **Source (Источник)**:
- Щелкните правой кнопкой мыши по соответствующей ячейке столбца **Л** матрицы ранжирования и выберите **X (Ранжировано)** из появившегося контекстного меню.

	Информация			Источник			Цель								
	Номер	Текст дисплея	НС	Тип	Вх	Ф	С	Л	Вы	СД	Буфер	С	Л	Д	МЧ
									ВЫ	ПР	ПЗ	ПА			
P System Data 2									*	*	*	*			
67 Direct. O/C	02691	67/67N pickedup		OUT							пч	×			
	02696	67/67N TRIP		OUT							п	×			
	02604	>BLK 67/67-TOC		SP									×		
	02615	>BLOCK 67-2		SP										×	
	02621	>BLOCK 67-1		SP											×
	02651	67/67-TOC OFF		OUT							пч				
	02652	67 BLOCKED		OUT							пч		пч	×	
	02653	67 ACTIVE		OUT							пч				×
	02642	67-2 picked up		OUT									пч	×	
	02649	67-2 TRIP		OUT									п	×	
	02660	67-1 picked up		OUT									пч	×	
	02665	67-1 TRIP		OUT									п	×	
	02692	67 A picked up		OUT									пч	×	
	02693	67 B picked up		OUT									пч	×	
	02694	67 C picked up		OUT									пч	×	
	02647	67-2 Time Out		OUT											×
	02664	67-1 Time Out		OUT											×
	02628	Phase A forward		OUT							п				×
	02629	Phase B forward		OUT							п				×
	02630	Phase C forward		OUT							п				×
	02632	Phase A reverse		OUT							п				×
	02633	Phase B reverse		OUT							п				×
	02634	Phase C reverse		OUT							п				×
	02637	67-1 BLOCKED		OUT							пч		пч	×	
	02655	67-2 BLOCKED		OUT							пч		пч	×	
	02614	>BLK 67N/67NTOC		SP											
	02616	>BLOCK 67N-2		SP											
	02623	>BLOCK 67N-1		SP											
	02656	67N OFF		OUT								пч			×
	02657	67N BLOCKED		OUT								пч		пч	×
02658	67N ACTIVE		OUT								пч			×	
02646	67N-2 picked up		OUT									пч	×		
02679	67N-2 TRIP		OUT									п	×		
02681	67N-1 picked up		OUT									пч	×		
02683	67N-1 TRIP		OUT									п	×		
02695	67N picked up		OUT									пч	×		
02648	67N-2 Time Out		OUT											×	
02682	67N-1 Time Out		OUT											×	
02636	Ground reverse		OUT							п				×	
02635	Ground forward		OUT							п				×	
02668	67N-2 BLOCKED		OUT							пч		пч	×		
02659	67N-1 BLOCKED		OUT							пч		пч	×		
Measurerm_Superv										*	*	*	*		

RANGIERUNG_BSP01_02.gif

Рис. 2-2 Ранжирование информации на CFC как выходного сигнала

Информация, заранжированная на CFC, отмечена в столбце **Л** значком **X**.



Примечание:

Если измеряемые величины заранжированы на CFC как входные сигналы для левой границы схемы и соединены в схемах CFC с одним из классов приоритетов PLC_BEARB [Fast PLC] и PLC1_BEARB [Slow PLC], изменение измеряемых величин не приведет к автоматическому запуску обработки этих схем.

Измеряемые величины должны обрабатываться в классе приоритетов MW_BEARB (обработка измеряемых величин).



Примечание:

Информация, заранжированная на CFC как выходной сигнал на правой границе схемы, регистрируется в журналах событий соответствующего устройства только в том случае, если в схеме CFC меняется значение соединенного сигнала.



Примечание:

Информацию переходного процесса нельзя использовать в качестве входного сигнала для логических операций в CFC.

Информацию переходного процесса можно использовать для запуска процесса обработки схем в классах приоритетов PLC_BEARB (Fast PLC (быстрая логика)) и PLC1_BEARB (Slow PLC (медленная логика)). Тогда схемы будут обрабатываться при появлении соответствующего сообщения.

2.5 Пример быстрой логики Fast PLC

Задачи быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**) обрабатываются с наивысшим приоритетом. Обработка управляется событиями и производится в результате изменений входных сигналов.



Примечание:

Для приведенного ниже примера, был выбран набор функций устройства SIPROTEC с кодом заказа MLFB **7SJ63655ER633HH3**. Если вы хотите воспроизвести данный пример, у вас должно быть вставлено подобное устройство из каталога устройств в DIGSI.

В окне **Device Configuration (Конфигурация устройства)** SIPROTEC, выберите опцию Enabled (доступна) для следующих функций: **67, 67-ТОС** и **67N, 67N-ТОС**.

Задача	При коротком замыкании в цепях напряжения должна быть заблокирована направленная МТЗ.
Входной сигнал	<input type="checkbox"/> Группа: Measur. superv (контроль измеряемых величин) Текст дисплея: VT Fuse Failure (повреждение предохранителя ТН)
Выходные сигналы	<input type="checkbox"/> Группа 67 Direct. O/C (67 Направленная МТЗ) Текст дисплея >BLK 67/67-ТОС <input type="checkbox"/> Группа 67 Direct. O/C (67 Направленная МТЗ) Текст дисплея >BLK 67N/67NТОС
Блок CFC	<input type="checkbox"/> CONNECT (соединение)
Порядок выполнения действий	<ul style="list-style-type: none"> • В матрице ранжирования DIGSI заранжируйте входные и выходные сигналы на CFC (см. Главу 2.4). • Создайте новую схему CFC под названием BLK DIR FOR MCB TRIP. • Задайте класс приоритетов PLC_BEARB [быстрая логика Fast PLC]. • Разместите в схеме блок CONNECT. • Соедините входной сигнал. • Соедините выходные сигналы. • Откомпилируйте схему CFC.

2.5.1 Создание новой схемы CFC

Как создать новую схему CFC под названием **BLK DIR FOR MCB TRIP**:

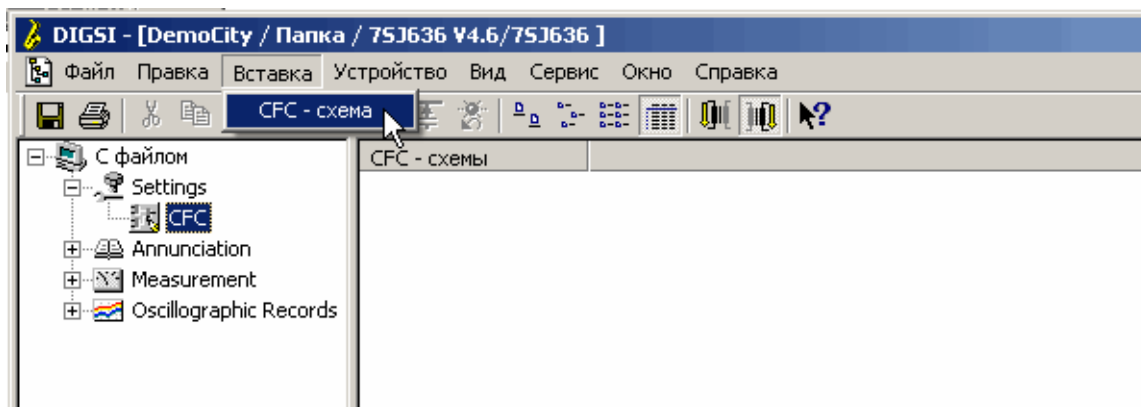
- Перейдите к папке CFC:
 - Выберите папку **Settings (Уставки)** в навигационной области окна DIGSI и
 - щелкните два раза по **CFC** в области данных.
- В окне данных появится список всех имеющихся схем.



Примечание:

Для обеспечения согласованности, выходные сигналы, которые отображаются на правой границе DIGSI CFC, могут быть соединены во всех схемах только один раз (см. Главу 2.5.5).

- Нажмите на **Insert (Вставка) → CFC chart (CFC - схема)**.



CFC_PLAN_EINFUEGENa.gif

Рис. 2-3 Создание новой схемы CFC

- Как переименовать новую схему CFC:
 - Щелкните правой кнопкой мыши по автоматически присвоенному имени схемы CFC и выберите **Object Properties (Свойства объекта...)** из появившегося контекстного меню.
 - В открывшемся диалоговом окне **Properties CFC (Свойства CFC схема)**, введите новое название схемы **BLK DIR FOR MCB TRIP** в поле **Name (Название)**.

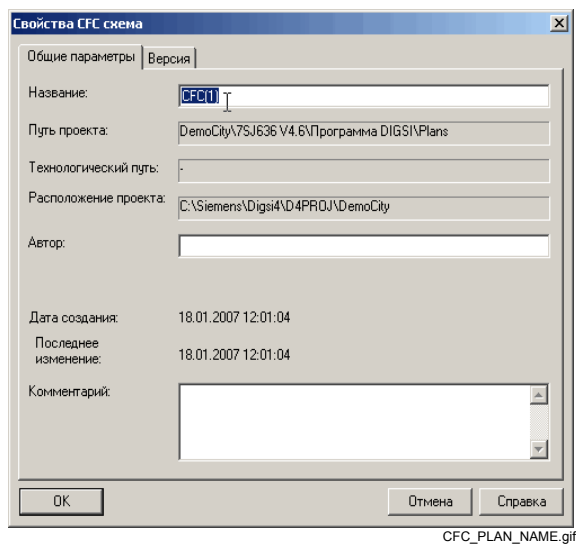


Рис. 2-4 Переименование схемы CFC

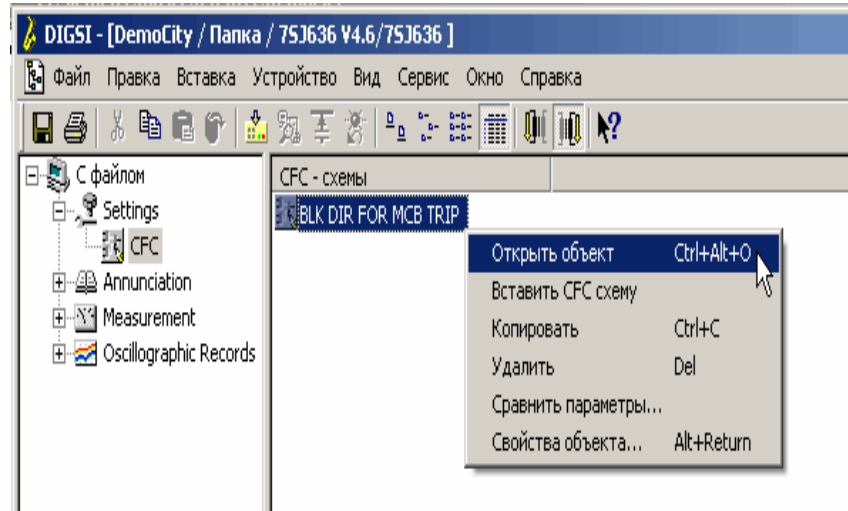


Примечание:

Поля **Author (Автор)** и **Comment (Комментарий)** могут использоваться для ведения документации: так, например, в поле комментария вы можете вести электронный протокол изменений.

- Подтвердите сделанные вами изменения, нажав на кнопку **OK**.

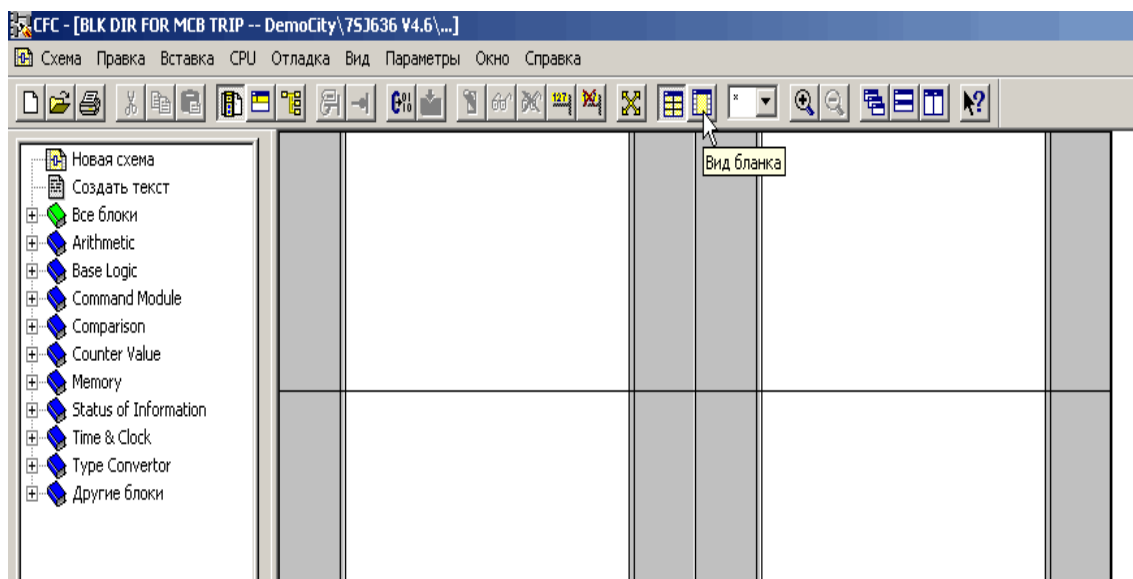
- Как открыть схему CFC.
 - Щелкните правой кнопкой мыши по имени схемы CFC и выберите **Open object (Открыть объект...)** из появившегося контекстного меню.



CFC_BSP01_01a.gif

Рис. 2-5 Открытие схемы CFC

- Если вы находитесь в режиме обзора всей схемы (шесть листов), переключитесь в режим отображения одного листа Sheet View (Вид бланка). Для этого, нажмите на соответствующую кнопку на панели инструментов.



CFC_BSP01_02a.gif

Рис. 2-6 Переключение отображения в вид бланка



Примечание:

Вы можете разбить всю схему CFC на отдельные схемы (каждую с шестью листами) для того, чтобы сделать объемную схему CFC более наглядной.

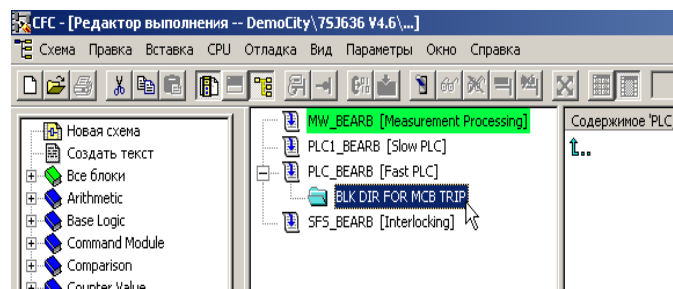
Соединение между отдельными схемами происходит путем соединения сигналов через границы схем.

Дополнительную информацию по данному вопросу вы найдете в диалоговой справке по DIGSI CFC.

2.5.2 Определение класса приоритетов

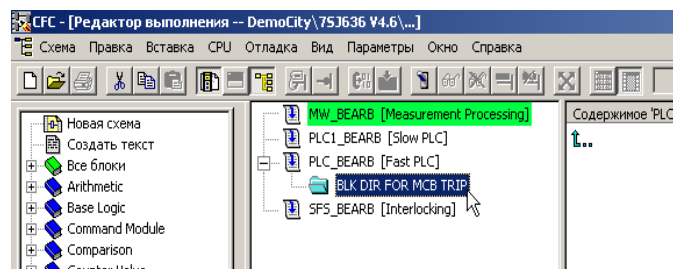
Как задать класс приоритетов **PLC_BEARB**:

- В открытой схеме CFC нажмите на **Edit (Правка) → Run Sequence (Последовательность выполнения...)** для того, чтобы открыть **Run-time editor (Редактор последовательности выполнения задач)**.
- Перейдите к схеме CFC с названием **BLK RICHT WG AUTOFALL** и перетащите схему CFC в класс приоритетов **PLC_BEARB** методом „перетаски и отпусти“.



CFC_Ablaufebene_aendern_01.tif

Рис. 2-7 Изменение класса приоритетов методом „перетаски и отпусти“



CFC_Ablaufebene_aendern_02.tif

Рис. 2-8 Схема CFC в новом классе приоритетов

- Нажмите еще раз на **Edit (Правка) → Run Sequence (Последовательность выполнения...)** в редакторе последовательности выполнения задач для возвращения к схеме CFC.



Примечание:

Проделайте следующие шаги для того, чтобы автоматически поместить новые схемы CFC в класс приоритетов **PLC_BEARB**:

- В навигационной области выделите уровень **PLC_BEARB (быстрая логика Fast PLC)** и нажмите на пункт меню **Edit (Правка) → Predecessor for installation (Предыдущее положение)**.
 - В появившемся на экране сообщении, нажмите на **OK** для подтверждения изменения класса приоритетов.
-

2.5.3 Размещение блока

Как расположить блок **CONNECT** в схеме CFC:

- Щелкните в каталоге по вкладке **Blocks (Блоки)**.



Примечание:

Дополнительную информацию об отдельных вкладках каталога вы найдете в диалоговой справке по DIGSI CFC.

- Щелкните по блоку **CONNECT** и, удерживая нажатой кнопку мыши, перетащите блок на лист схемы CFC.

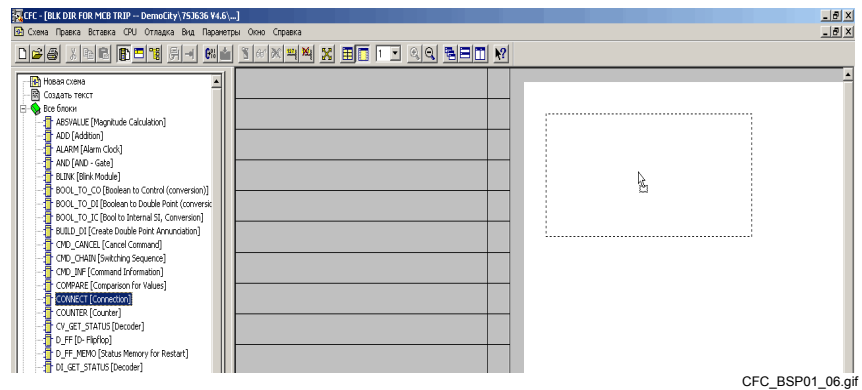
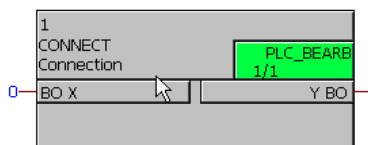


Рис. 2-9 Размещение блока CFC методом „перетаски и отпусти“

- Отпустите кнопку мыши. Блок отображается на листе схемы.



Примечание:



ConnectKopfInfo.tif

Рис. 2-10 Информация в заголовке блока

Заголовок размещенного на листе блока отображает:

- название блока (например, **1**),
- тип блока (например, **CONNECT**),
- класс приоритетов (например, **PLC_BEARB**) и ниже номер в последовательности выполнения (например, **1**).



Примечание:

- Все блоки одной схемы CFC должны находиться в одном и том же классе приоритетов.



Примечание:

Для того чтобы вам было легче сориентироваться в схеме CFC, вы могли бы присвоить названия блокам CFC, в зависимости от цели их применения:

Как переименовать блок:

- Щелкните правой кнопкой мыши по блоку и выберите **Object Properties (Свойства объекта...)** из открывшегося контекстного меню.
- Перейдите во вкладку **General (Общие параметры)** и введите новое название блока в поле **Name (Название)**.
- Нажмите **OK** для подтверждения произведенных изменений.

**Примечание:**

Соединенные в схеме CFC блоки, обрабатываются в указанной последовательности. Последовательность выполнения может быть определена при помощи номеров внутри блоков. Нумерация блоков должна соответствовать логической последовательности.

Если вы помещаете в CFC схему несколько блоков или позже добавляете в нее новые блоки, вы должны проверить класс приоритетов или последовательность выполнения блоков и, при необходимости, исправить несоответствия.

Для изменения класса приоритетов или номера блока в последовательности выполнения, выполните следующие шаги:

- Щелкните правой кнопкой мыши по блоку и выберите из открывшегося контекстного меню **Go to Installation Position (Перейти к положению установки)**. Откроется **Run-time editor (Редактор последовательности выполнения)**. В навигационной области редактора последовательности выполнения текущий блок будет выделен.
- Для удаления блока из класса приоритетов или последовательности выполнения, щелкните правой кнопкой мыши по блоку и выберите **Cut (Вырезать)** из контекстного меню. Блок отображается с приглушенной яркостью.
- Для вставки блока в новом месте в классе приоритетов или последовательности выполнения, щелкните правой кнопкой мыши по новому месту и выберите **Paste (Вставить)** из контекстного меню. Блок вставится на новое место.
- Щелкните в редакторе последовательности выполнения по пункту меню **Edit (Правка) → Run Sequence (Последовательность выполнения...)**. Схема CFC вновь отобразится на экране.

**Примечание:**

Как переместить уже размещенный или соединенный блок в схеме CFC:

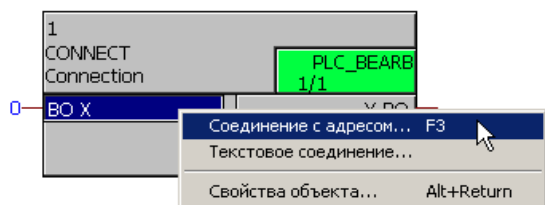
- Щелкните по заголовку блока, не отпуская при этом кнопку мыши, и перетяните блок на новое место в схеме.
- Отпустите кнопку мыши. Блок, со всеми его соединениями, отобразится на новом месте.

Последовательность выполнения **не изменяется** автоматически при перемещении блока по схеме.

2.5.4 Соединение входного сигнала

Как соединить входной сигнал **Measurement Superv Fuse Failure** с блоком **CONNECT**:

- Щелкните правой кнопкой мыши по входу **BO X** блока **CONNECT** и выберите из открывшегося контекстного меню **Interconnection to Address (Соединение с адресом...)**.



CFC_BSP01_07.gif

Рис. 2-11 Соединение входного сигнала

- В открывшемся диалоговом окне **Select Left Border (Соединение с левой границей)**, выберите информацию **Measurement Superv Fuse Failure**. Ориентируйтесь по трем столбцам **Group (Группа)**, **Display Text (Текст дисплея)** и **Type (Тип)**, с которыми вы уже познакомились в матрице ранжирования.



Примечание:

Список диалогового окна **Select Left Border (Соединение с левой границей)** отображает только ту информацию,

- которая отмечена в **Configuration Matrix (Матрица ранжирования)** DIGSI как **Destination (Цель)** CFC и,
- которая может быть соединена с типом данных выбранного блока.

- Нажмите на **ОК** для подтверждения вашего выбора.

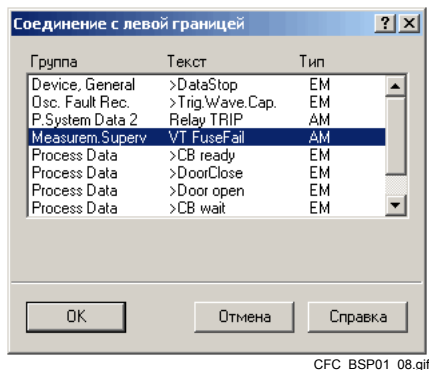


Рис. 2-12 Диалоговое окно выбора левой границы

Новое соединение отображается в схеме CFC. Входной сигнал появляется в левой части схемы и соединяется с блоком линией.

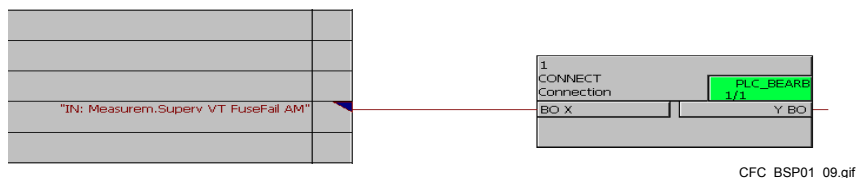
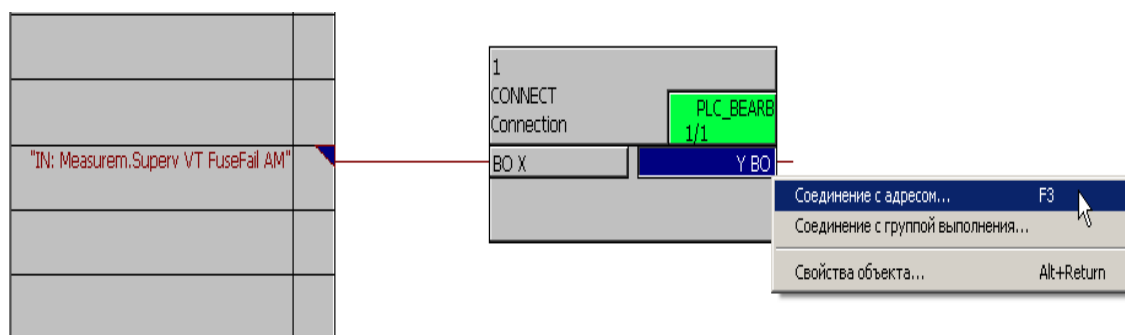


Рис. 2-13 Соединенный входной сигнал в быстрой логике Fast PLC

2.5.5 Соединение выходных сигналов

Как соединить выходные сигналы **67 Direct. O/C (67 Направленная MTЗ) >BLK 67/67-ТОС** и **67 Direct. O/C (67 Направленная MTЗ) >BLK 67N/67NТОС** с блоком **CONNECT**:

- Как соединить информацию **67 Direct. O/C (67 Направленная MTЗ) >BLK 67/67-ТОС** с блоком:
 - Щелкните правой кнопкой мыши по выходу **Y** блока **CONNECT** и выберите из открывшегося контекстного меню **Interconnection to Address (Соединение с адресом...)**.



CFC_BSP01_10.gif

Рис. 2-14 Соединение выходных сигналов



Примечание:

Никогда ни при каких условиях не используйте функцию **Interconnection to Run-time Group (Соединение с группой выполнения...)**.

Этот тип связи не поддерживается DIGSI CFC.



Примечание:

Для обеспечения согласованности, выходной сигнал может быть использован во всех схемах CFC только один раз.

- В открывшемся диалоговом окне **Select Right Border (Соединение с правой границей)**, выберите информацию **67 Направленная MTЗ >BLK 67/67-ТОС**. Ориентируйтесь по трем столбцам **Group (Группа)**, **Display Text (Текст дисплея)** и **Typ (Тип)**, с которыми вы уже познакомились в матрице ранжирования.

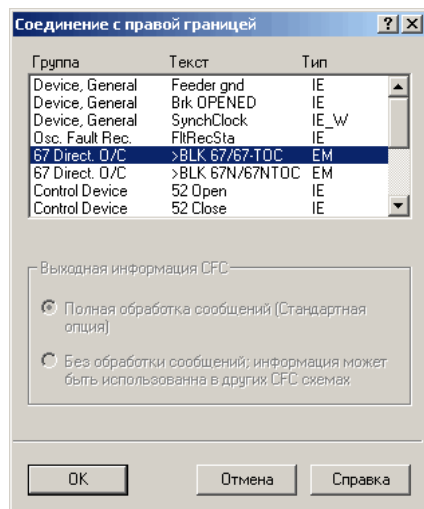


Примечание:

Список диалогового окна **Select Right Border (Соединение с правой границей)** отображает только ту информацию,

- ❑ которая отмечена в **Configuration Matrix (Матрица ранжирования)** DIGSI как **Source (Источник)** CFC и,
 - ❑ которая может быть соединена с типом данных выбранного блока.
-

- Нажмите на **ОК** для подтверждения вашего выбора.



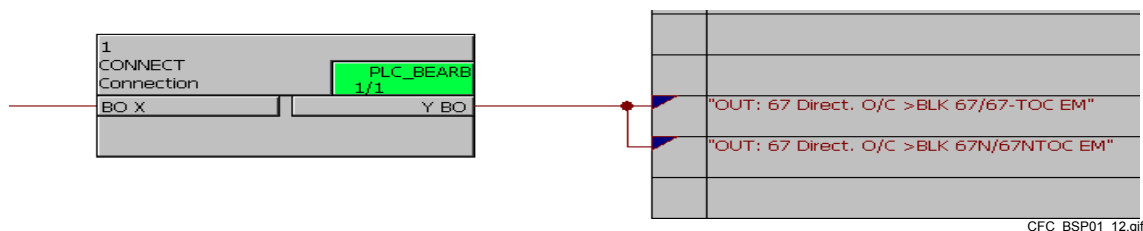
CFC_BSP01_11.gif

Рис. 2-15 Диалоговое окно соединения с правой границей

Новое соединение отображается в схеме CFC. Входной сигнал появляется в правой части схемы и соединяется с блоком линии.

- Как соединить информацию **67 Направленная MT3 >BLK67N/67NTOC** с блоком:
 - Щелкните правой кнопкой мыши по выходу **BO Y** блока **CONNECT** и выберите из открывшегося контекстного меню **Interconnection to Address (Соединение с адресом...)**.
 - В открывшемся диалоговом окне **Select Right Border (Соединение с правой границей)**, выберите информацию **67 Направленная MT3 >BLK 67N/67NTOC**. Ориентируйтесь по трем столбцам **Group (Группа)**, **Display Text (Текст дисплея)** и **Type (Тип)**.
 - Нажмите на **ОК** для подтверждения вашего выбора.

Новое соединение отображается в схеме CFC. Входной сигнал появляется в правой части схемы и соединяется с блоком линии.



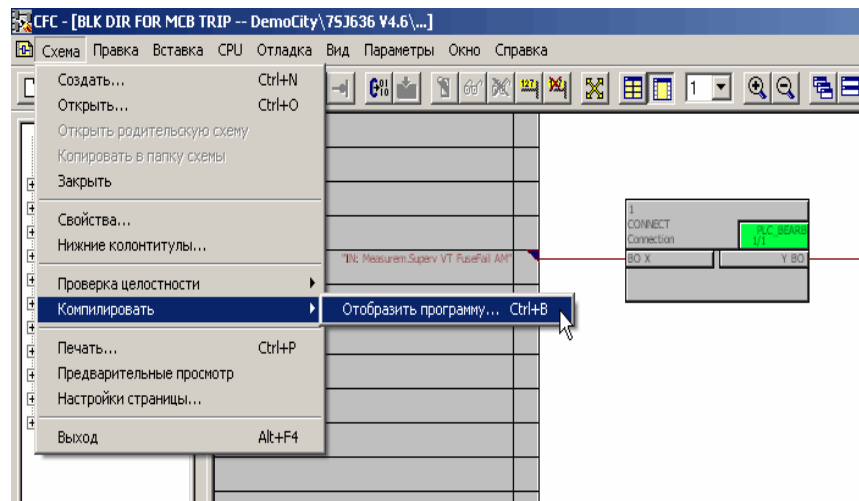
CFC_BSP01_12.gif

Рис. 2-16 Соединенные выходные сигналы в быстрой логике Fast PLC

2.5.6 Компиляция схемы CFC

Для того чтобы использовать созданную схему CFC, а, следовательно, и запрограммированные функции в устройстве SIPROTEC, схему необходимо откомпилировать:

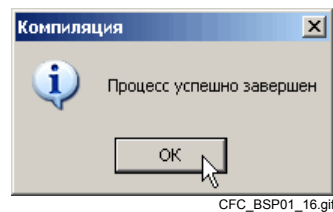
- Выберите **Chart (Схема) → Compile (Компилировать) → Charts as Program (Отобразить программу...)**. Все имеющиеся схемы будут заново откомпилированы.



CFC_BSP01_15.gif

Рис. 2-17 Компиляция схемы CFC

- В появившемся окне сообщений нажмите на кнопку **ОК**.



CFC_BSP01_16.gif

Рис. 2-18 Диалоговое окно компиляции

Для того чтобы использовать запрограммированные функции после компиляции, вы должны сохранить в DIGSI набор уставок и перезагрузить его в устройство SIPROTEC.

2.6 Пример медленной логики Slow PLC

Задачи медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) обрабатываются с низким приоритетом. Обработка управляется событием, в результате изменений входных сигналов.



Примечание:

Для приведенного ниже примера, был выбран набор функций устройства SIPROTEC с кодом заказа MLFB **7SJ63655ER633HH3**. Если вы хотите воспроизвести данный пример, у вас должно быть вставлено подобное устройство из каталога устройств в DIGSI.

Задача	Режим местного управления должен отображаться в устройстве SIPROTEC на светодиодах.
Входной сигнал	<input type="checkbox"/> Группа Cntrl Authority Текст дисплея Cntrl Auth
Выходной сигнал	<input type="checkbox"/> Группа Cntrl Authority Текст дисплея LocalCntrl Этой информации пока еще нет в матрице ранжирования DIGSI.
Блок CFC	<input type="checkbox"/> DI_TO_BOOL (преобразование двухпозиционного сигнала в логическую величину)
Порядок выполнения действий	<ul style="list-style-type: none"> • Вставьте новый информационный элемент Cntrl Authority LocalCntrl в матрицу ранжирования DIGSI. • В матрице ранжирования DIGSI заранжируйте информационный элемент Cntrl Authority LocalCntrl на светодиод. • В матрице ранжирования DIGSI заранжируйте входные и выходные сигналы на CFC (см. Главу 2.4). • Создайте новую схему CFC и назовите ее DEVICE PANEL CTRL MODE (см. Главу 2.5.1) • Задайте класс приоритетов PLC1_BEARB (см. Главу 2.5.2). • Вставьте в схему блок DI_TO_BOOL (см. Главу 2.5.3). • Соедините входной сигнал (см. Главу 2.5.4). • Сконфигурируйте блок DI_TO_BOOL с пороговыми величинами IS_OFF (Выкл.) и IS_ON (Вкл.) для обработки двухпозиционного сообщения Cntrl Authority Cntrl Auth. • Соедините выходной сигнал (см. Главу 2.5.5). • Откомпилируйте схему CFC (см. Главу 2.5.6).

2.6.1 Добавление новой информации

Как добавить новый информационный элемент **Cntrl Authority LocalCntrl** в матрицу ранжирования DIGSI:

- Откройте в DIGSI матрицу ранжирования:
 - Выберите папку **Settings (Уставки)** в навигационной области окна и
 - Щелкните мышью два раза по функции **Masking I/O (Configuration Matrix) (Ранжирование входов / выходов (Матрица ранжирования))**, находящейся в области данных.
- Откройте группу **Cntrl Authority**.
- Нажмите на **Insert (Вставка) → Information (Информация...)**. Откроется окно **Information Catalog (Информационный каталог)**.

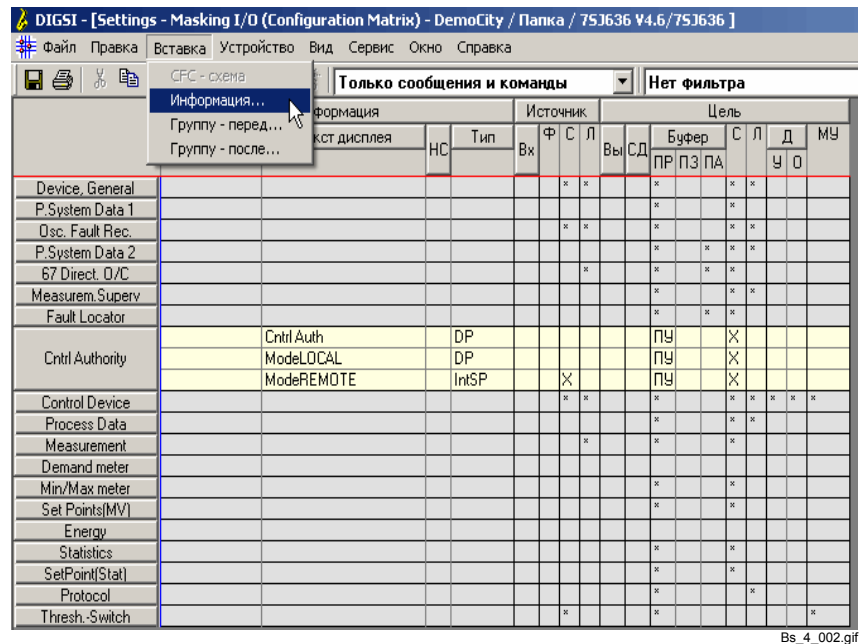


Рис. 2-19 Добавление информации

- Выберите тип информации **Annunciations (Сообщения) \Single Point (Однопозиционные) (SP)\ON / OFF (Вкл / Выкл) (SP)** из окна **Information Catalog (Информационный каталог)**. Удерживая нажатой кнопку мыши, переместите выбранный тип информации в группу **Cntrl Authority**.

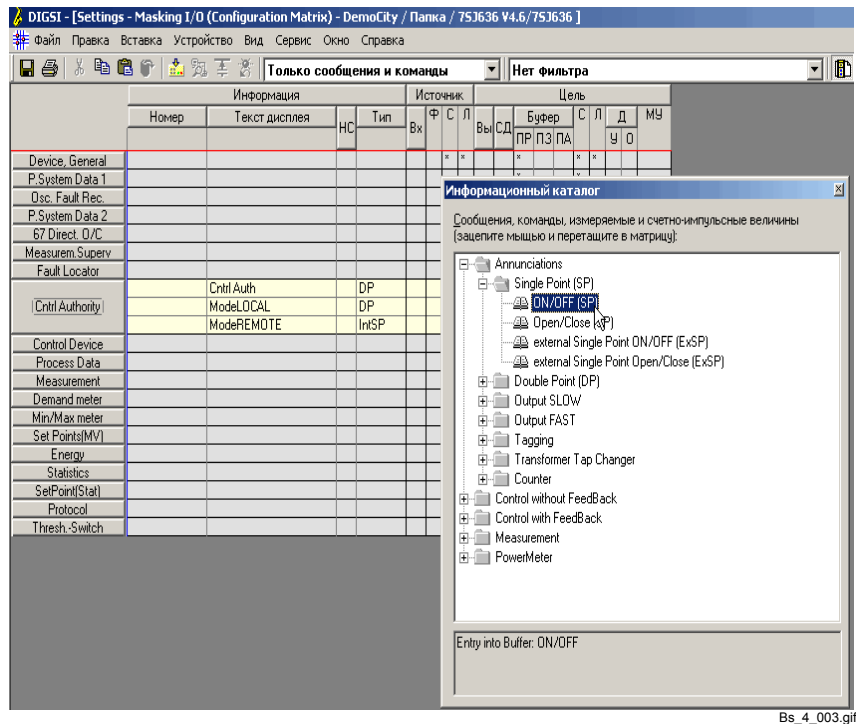


Рис. 2-20 Выбор информационного типа

- Для того чтобы переименовать добавленную информацию, дважды щелкните по названию, заданному по умолчанию в столбце **Display Text (Текст дисплея)**, и введите название **LocalCntrl**.

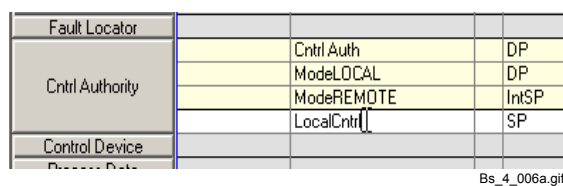


Рис. 2-21 Переименование информации

2.6.2 Ранжирование информации на светодиод

Как заранжировать в матрице DIGSI информацию **Cntrl Authority LocalCntrl** на светодиод.

- Откройте в DIGSI матрицу ранжирования:
 - Выберите папку **Settings (Уставки)** в навигационной области окна и
 - Щелкните мышью два раза по функции **Masking I/O (Configuration Matrix) (Ранжирование входов / выходов (Матрица ранжирования))**, находящейся в области данных.
- Разверните группу **Cntrl Authority** и выберите информацию **LocalCntrl**.
- Щелкните правой кнопкой мыши по ячейке 7 столбца **LED (СД)**, которая находится под разделом матрицы **Destination (Цель)**, и в открывшемся контекстном меню выберите **U (unlatched) (Б (Без запоминания))**.

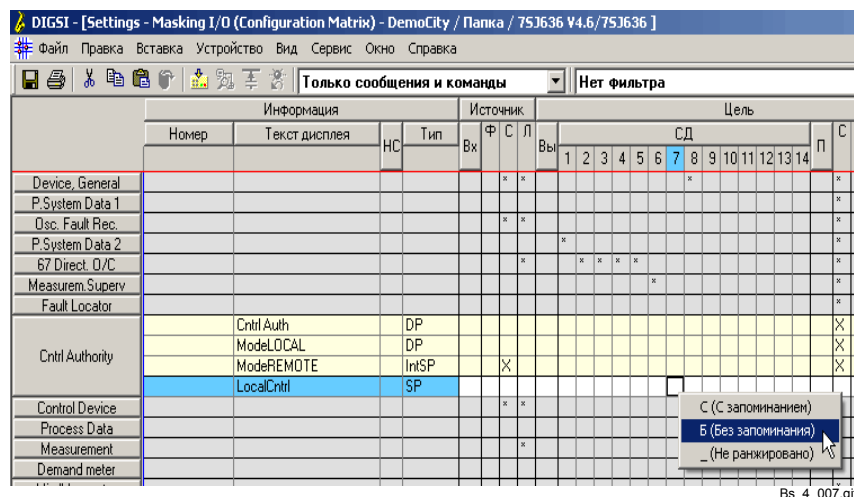


Рис. 2-22 Ранжирование информации на светодиод

2.6.3 Соединение входного сигнала

Для того чтобы соединить входной сигнал **Cntrl Authority Cntrl Auth** с блоком **DI_TO_BOOL**, проделайте шаги, описанные в Гл. 2.5.4.

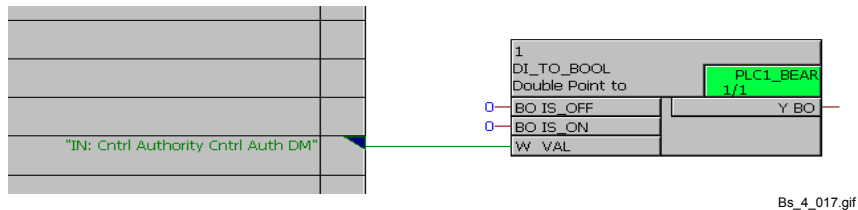
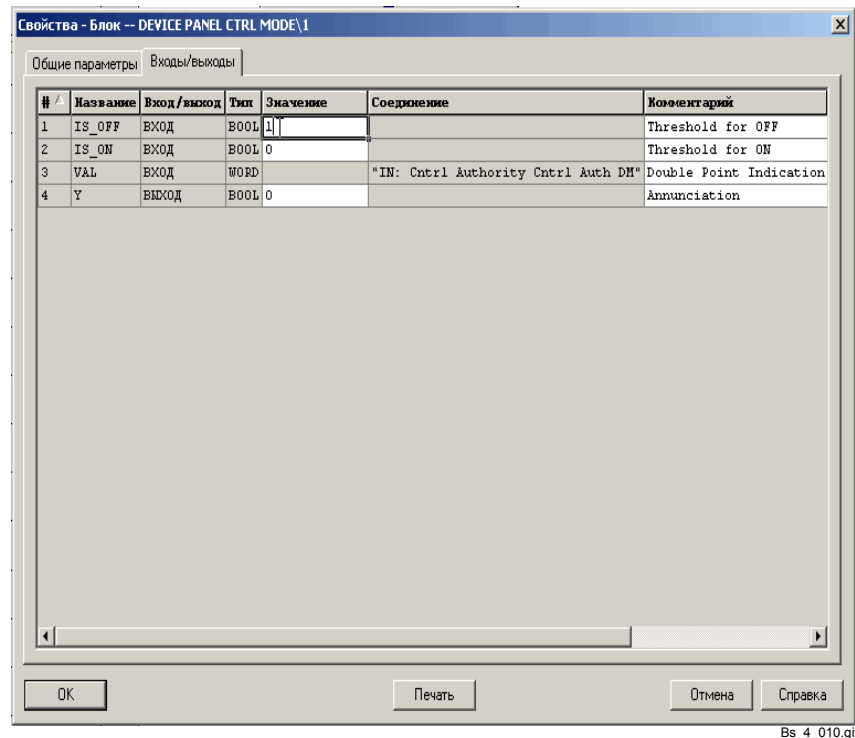


Рис. 2-23 Соединенный входной сигнал, пример логики Slow PLC

2.6.4 Параметрирование блока

Как задать в блоке **DI_TO_BOOL** пороговые величины **IS_OFF** и **IS_ON** для обработки двухпозиционного сообщения **Cntrl Authority Cntrl Auth DP**:

- Щелкните правой кнопкой мыши по блоку и выберите **Object Properties (Свойства объекта...)** из открывшегося контекстного меню.
- Перейдите во вкладку **Inputs / Outputs (Входы / выходы)** и введите в строке **IS_OFF** столбца **Value (Значение)** 1 в качестве пороговой величины.



Bs_4_010.gif

Рис. 2-24 Параметрирование блока

- Подтвердите сделанные вами изменения, нажав на кнопку **ОК**.

2.6.5 Соединение выходного сигнала

Для того чтобы соединить выходной сигнал **Cntrl Authority LocalCntrl** с блоком **DI_TO_BOOL**, проделайте шаги, описанные в Гл. 2.5.5.

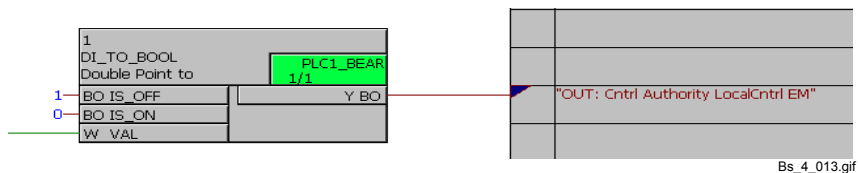


Рис. 2-25 Соединенный выходной сигнал, пример логики Slow PLC

2.7 Пример обработки измеряемых величин

Обработка измеряемых величин (класс приоритетов **MW_BEARB**) имеет средний приоритет и производится циклически.



Примечание:

Для приведенного ниже примера, был выбран набор функций устройства SIPROTEC с кодом заказа MLFB **7SJ63655ER633HH3**. Если вы хотите воспроизвести данный пример, у вас должно быть вставлено подобное устройство из каталога устройств в DIGSI.

Задача	Предполагается, что на вход датчика подается измеряемая величина давления только в пределах линейного диапазона. Соответствующие сообщения должны быть выданы, если величина находится вне этого диапазона, или ниже предела, который может быть установлен в устройстве SIPROTEC.
Входные сигналы	<input type="checkbox"/> Группа Measurement Текст дисплея Td1= <input type="checkbox"/> Группа Set Points (MV) Текст дисплея Press<
Выходные сигналы	<input type="checkbox"/> Группа Measurement Текст дисплея Superv. Pressure <input type="checkbox"/> Группа Set Points (MV) Текст дисплея SP. Pressure<
Блоки CFC	<input type="checkbox"/> LIVE_ZERO (Компаратор) <input type="checkbox"/> LOWER_SETPOINT (Нижний предел)
Порядок выполнения действий	<ul style="list-style-type: none"> • В матрице ранжирования DIGSI заранжируйте входные и выходные сигналы на CFC (см. Главу 2.4). • Создайте новую схему CFC и назовите ее MES VAL PROCESSING (см. Главу 2.5.1). • Задайте класс приоритетов MW_BEARB (см. Главу 2.5.2). • Добавьте в схему блоки LIVE_ZERO и LOWER_SETPOINT (см. Главу 2.5.3). • Соедините входные сигналы (см. Главу 2.5.4). • Сконфигурируйте блок LIVE_ZERO с пороговыми величинами DetectKnee, DispKnee и LiveZero для обработки измеряемой величины measured values Td1= (см. Главу 2.6.4). • Соедините блоки LIVE_ZERO и LOWER_SETPOINT.

- Соедините выходные сигналы (см. Главу 2.5.5).
- Откомпилируйте схему CFC (см. Главу 2.5.6).

2.7.1 Соединение входных сигналов

Для того чтобы соединить входные сигналы **Measurement Td1=** и **Set Points Press<** с блоками **LIVE_ZERO** и **LOWER_SETPOINT**, проделайте шаги, описанные в Гл. 2.5.4.

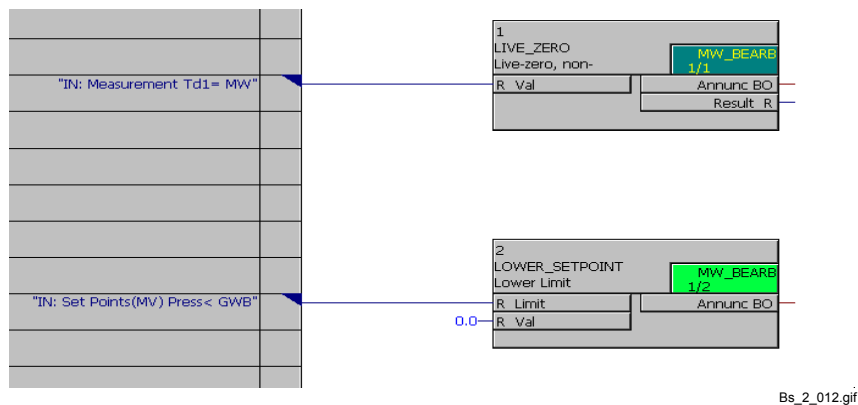


Рис. 2-26 Соединенные входные сигналы, обработка измеряемых величин

2.7.2 Соединение блоков

Как соединить друг с другом блоки **LIVE_ZERO** и **LOWER_SETPOINT**:

- Щелкните по связи **Result (Результат)** блока **LIVE_ZERO**, и, удерживая нажатой кнопку мыши, переместите курсор на связь **Val** блока **LOWER_SETPOINT**.

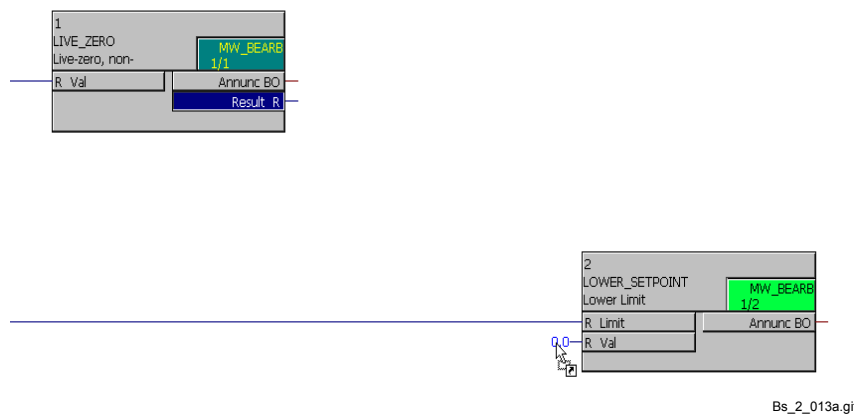


Рис. 2-27 Соединение блоков, пример обработки измеряемых величин

- Отпустите кнопку мыши. На экране появится связь между двумя входами / выходами.

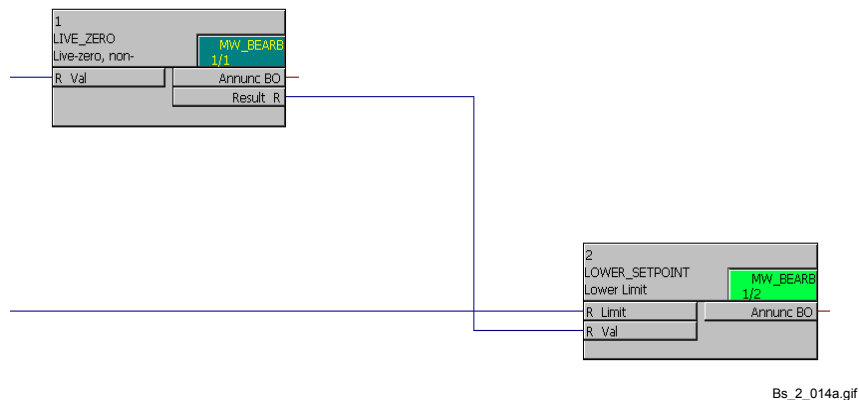


Рис. 2-28 Соединенные блоки, пример обработки измеряемых величин



Примечание:

Два входа / выхода могут быть соединены друг с другом, только если их типы данных идентичны.

2.7.3 Соединение выходных сигналов

Для того чтобы соединить сигналы **Measurement Superv. Pressure** и **Set Points (MV) SP<** с блоками **LIVE_ZERO** и **LOWER_SETPOINT**, проделайте шаги, описанные в Гл. 2.5.5.

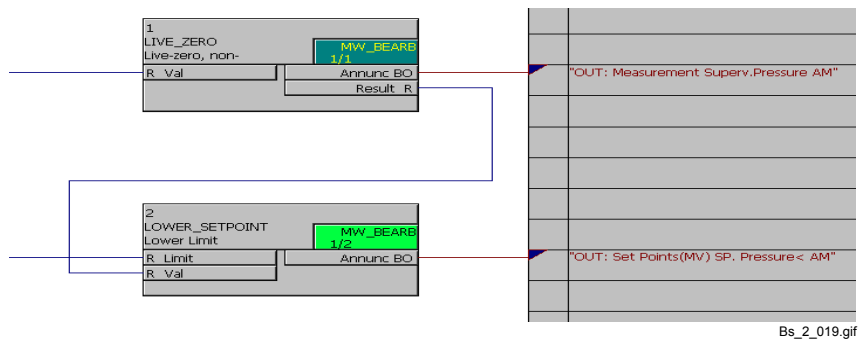


Рис. 2-29 Соединенные выходные сигналы, пример обработки измеряемых величин

2.8 Пример блокировки

Блокировка (класс приоритетов **SFS_BEARB**) используется, если должна быть выдана команда. Кроме того, циклическая обработка производится в фоновом режиме.



Примечание:

Для приведенного ниже примера, был выбран набор функций устройства SIPROTEC с кодом заказа MLFB **7SJ63655ER633HH3**. Если вы хотите воспроизвести данный пример, у вас должно быть вставлено подобное устройство из каталога устройств в DIGSI.

При вставлении устройства SIPROTEC, также вставляются стандартные схемы CFC. В схеме CFC **Interlocking (Блокировка)** уже используются выходные сигналы, показанные в данном примере. Для понимания данного примера, вам необходимо сначала будет удалить схему CFC **Interlocking**.

Задача	Необходимо проверить условие блокировки для включения и отключения заземляющего ножа и сформировать разрешающие сообщения.
Входные сигналы	<ul style="list-style-type: none"> <input type="checkbox"/> Группа Control Device (устройство управления) Текст дисплея 52Breaker Название сигнала 52 Breaker (силовой выключатель) <input type="checkbox"/> Группа Control Device Текст дисплея Disc.Swit. Название сигнала Disconnect Switch (разъединитель) <input type="checkbox"/> Группа Control Device Текст дисплея GndSwit. Название сигнала Ground Switch (заземляющий нож) <input type="checkbox"/> Группа Process Data (данные процесса) Текст дисплея >DoorClose Название сигнала >Door closed (дверь закрыта)
Выходные сигналы	<ul style="list-style-type: none"> <input type="checkbox"/> Группа Control Device Текст дисплея GndSw. Cl. Название сигнала Interlocking: Ground switch Close (блокировка: заземляющий нож включен) <input type="checkbox"/> Группа Control Device Текст дисплея GndSw. Open Название сигнала Interlocking: Ground switch open (блокировка: заземляющий нож отключен)
Блоки CFC	<input type="checkbox"/> DM_DECODE (декодирование двухпозиционного сигнала)

- **X_OR** (блок исключающего ИЛИ)
- **AND** (блок И)

Порядок выполнения действий

- В матрице ранжирования DIGSI заранжируйте входные и выходные сигналы на CFC (см. Главу 2.4).
- Создайте новую схему CFC и назовите ее **INTERL GND SWITCH** (см. Главу 2.5.1).
- Задайте класс приоритетов **SFS_BEARB** (см. Главу 2.5.2).
- Добавьте в схему блоки **DM_DECODE**, **X_OR** и **AND** (см. Главу 2.5.3).
- Соедините входные сигналы (см. Главу 2.5.4).
- Увеличьте количество входов блока **AND** до четырех.
- Соедините блоки **DM_DECODE**, **X_OR** и **AND** (см. Главу 2.7.2).
- Соедините выходные сигналы (см. Главу 2.5.5).
- Откомпилируйте схему CFC (см. Главу 2.5.6).

2.8.1 Соединение входных сигналов

Для того чтобы соединить входные сигналы **Control Device 52Breaker DP**, **Control Device Dis.Swit. DP**, **Control Device GndSwit.** и **Process indications >DoorClose SP** с блоками **DM_DECODE**, **X_OR** и **AND**, проделайте шаги, описанные в Гл. 2.5.4.

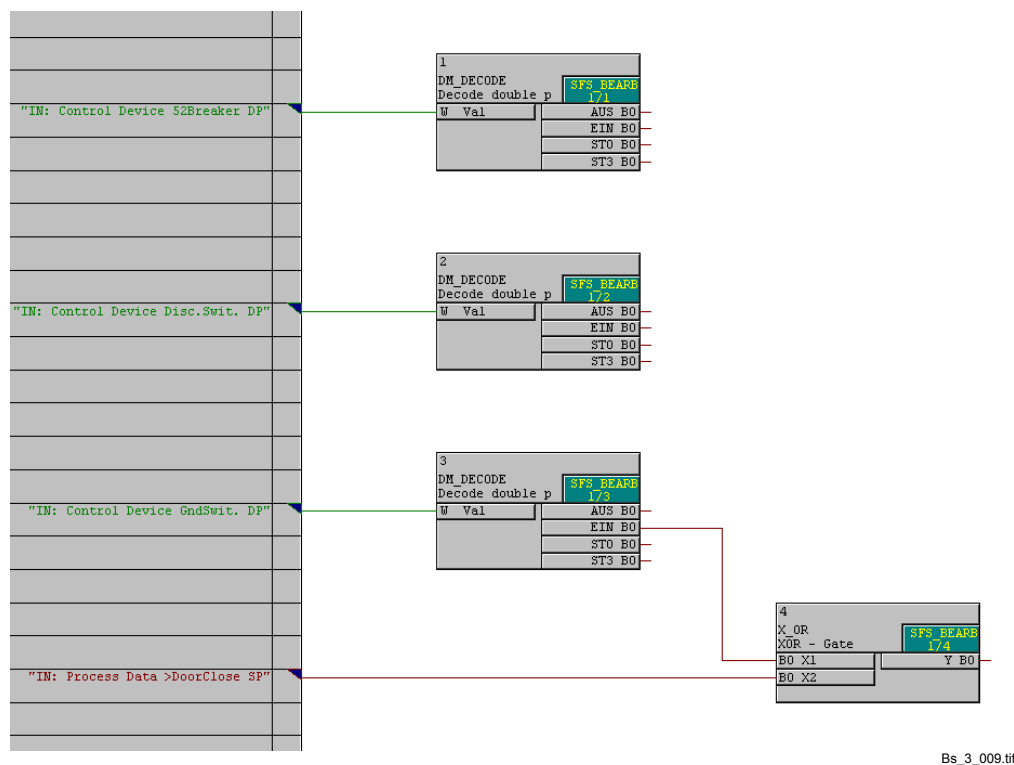


Рис. 2-30 Соединенные входные сигналы, пример блокировки

2.8.2 Увеличение количества входов блока

Как увеличить количество входов блока **AND** до четырех:

- Щелкните правой кнопкой мыши по блоку и выберите из открывшегося контекстного меню **Number of I / Os (Количество входов / выходов...)**.
- Введите 4 в поле **Number (Количество)**, которое находится в появившемся диалоговом окне **Number of I / Os (Количество входов / выходов)** и подтвердите ваш выбор, нажав на **ОК**.

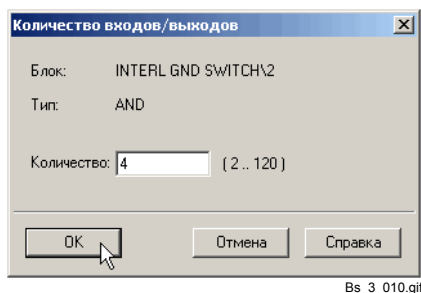
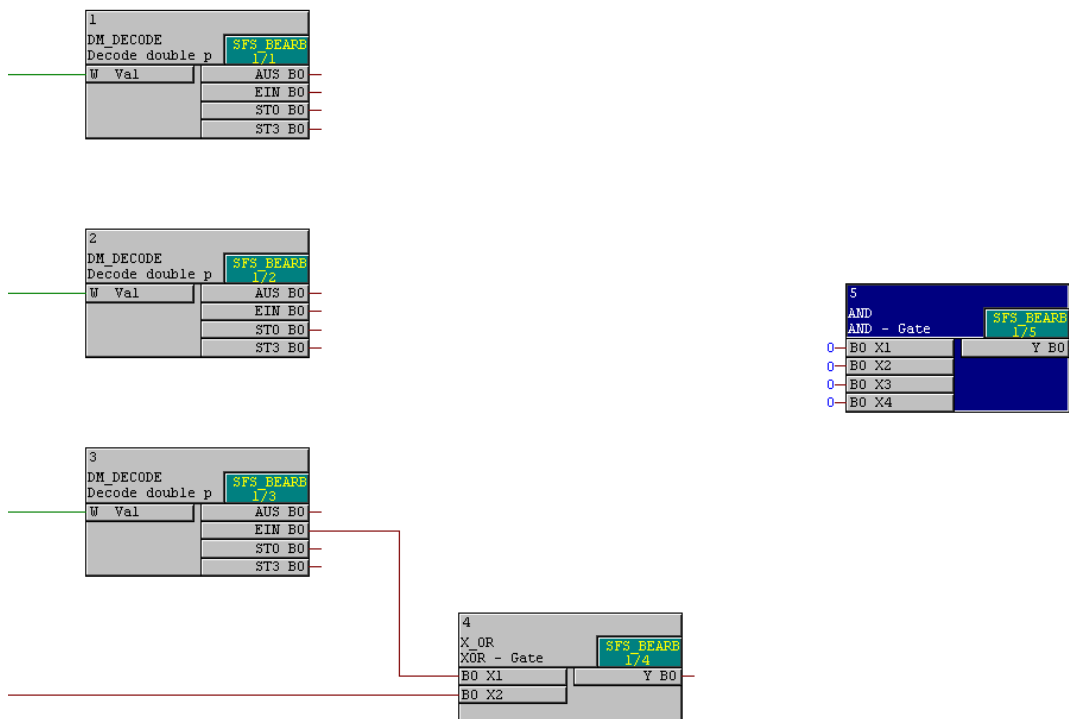


Рис. 2-31 Диалоговое окно количества входов / выходов

Блок отображается на экране с четырьмя входами.

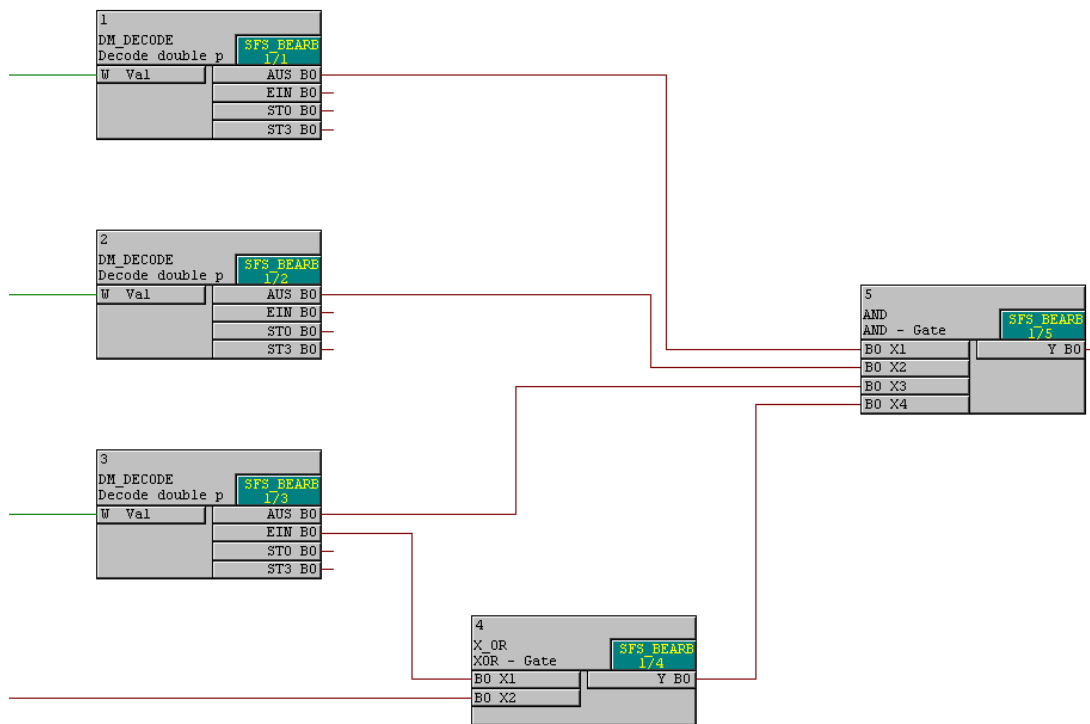


Bs_3_011.tif

Рис. 2-32 Увеличенное количество входов блока AND

2.8.3 Соединение блоков

Для того чтобы соединить между собой блоки **DM_DECODE**, **X_OR** и **AND**, проделайте шаги, описанные в Гл. 2.7.2.



Bs_3_012.tif

Рис. 2-33 Соединенные блоки, пример блокировки

2.8.4 Соединение выходных сигналов

Для того чтобы соединить выходные сигналы **Control Device GndSw Cl.** и **Control Device GndSw Open** с блоком **AND**, проделайте шаги, описанные в Гл. 2.5.5.

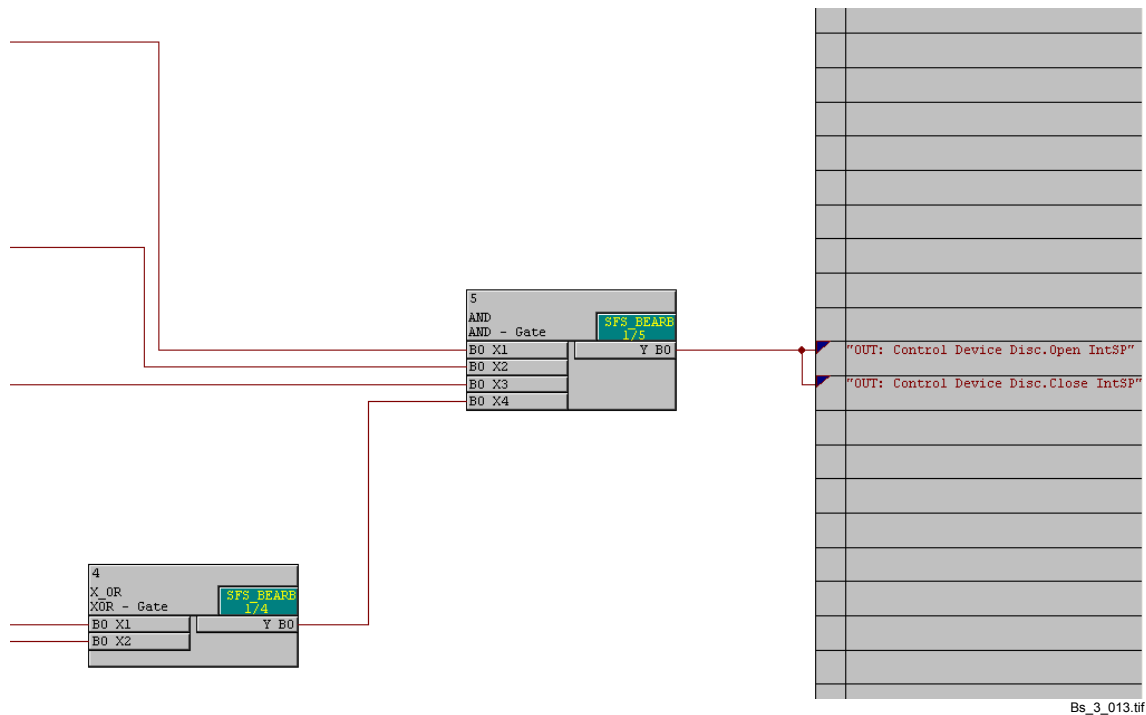


Рис. 2-34 Соединенные выходные сигналы, пример блокировки

Практические примеры

Обзор Данная глава содержит типичные примеры задач и их решений, взятые из практики.



Примечание:

Предполагается, что у вас есть опыт работы с DIGSI V4 или DIGSI CFC.

Глава 2 данного руководства предоставляет вам информацию по основам работы с DIGSI CFC.



Примечание:

Для отображения блоков CFC, приведенных в качестве примеров в данной главе, в настройке **Block width (Ширина блока)** была выбрана опция **wide (широкий)**:

- В редакторе CFC нажмите на **Options (Параметры)** → **Customize (Настроить)** → **Block / Sheet Bar Settings (Панели блока / бланка)**.
- В открывшемся окне отметьте опцию **wide (широкий)** в настройке **Block width (Ширина блока)** и подтвердите ваш выбор, нажав на **ОК**.

Содержание	3.1	Опция переключения набора уставок	50
	3.2	Мигающие светодиоды	61
	3.3	Обратная блокировка	67
	3.4	Подсчёт циклов переключения	71

3.1 Опция переключения набора уставок

Задача

Перед нами стоит следующая задача:

- Двигатель подключен к бинарному выходу ВА1. Двигатель должен запускаться при нажатии на функциональную кнопку F1.
- Набор уставок В должен стать активным, если все фазные токи становятся меньше 5% от номинального значения тока, и в первые 10 секунд после запуска двигателя.
- При нормальной работе, становится активным набор уставок А.

Запуск двигателя кнопкой F1

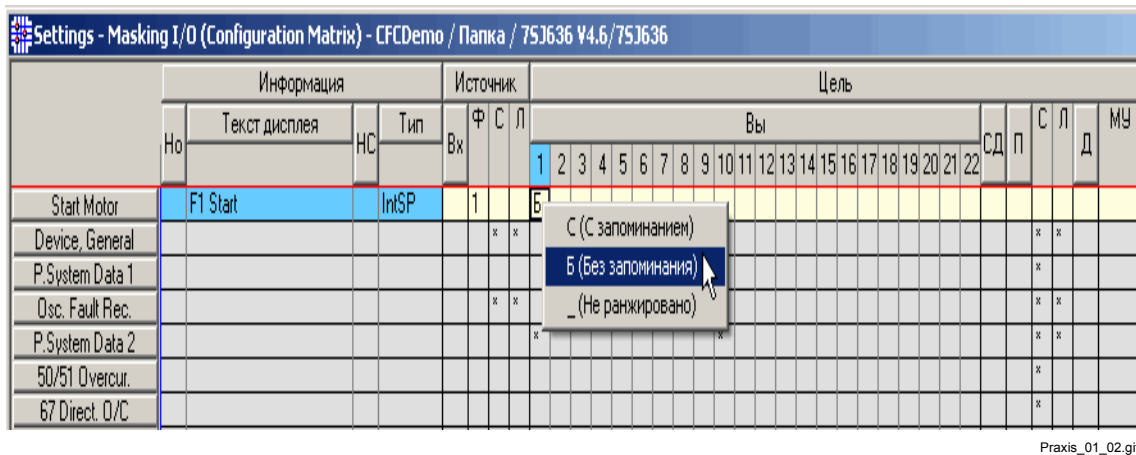
Запуск двигателя функциональной кнопкой F1 производится следующим образом:

- В матрице ранжирования DIGSI создайте новую группу под названием **Start Motor (Запуск двигателя)**.
- Вставьте новый информационный элемент типа **Marking ON / OFF (IE) (Маркирование Вкл. / Выкл.)** в группу **Start Motor (Запуск двигателя)**. Информационный элемент вставляется с именем по умолчанию.
- Переименуйте новую информацию на **F1 Start (F1 Запуск)**.
- Заранжируйте информацию **F1 Запуск** на функциональную кнопку 1.
- Заранжируйте информационный элемент **F1 Запуск** на бинарный выход **В01**. Выход должен быть заранжирован как **Б (Без запоминания)**.

Включение опции переключения набора уставок

Как включить опцию переключения набора уставок:

- В окне **Device Configuration (Конфигурация устройства)** установите функцию **Setting Group Change Option (Опция переключения набора уставок)** на **Enabled (Доступна)**.



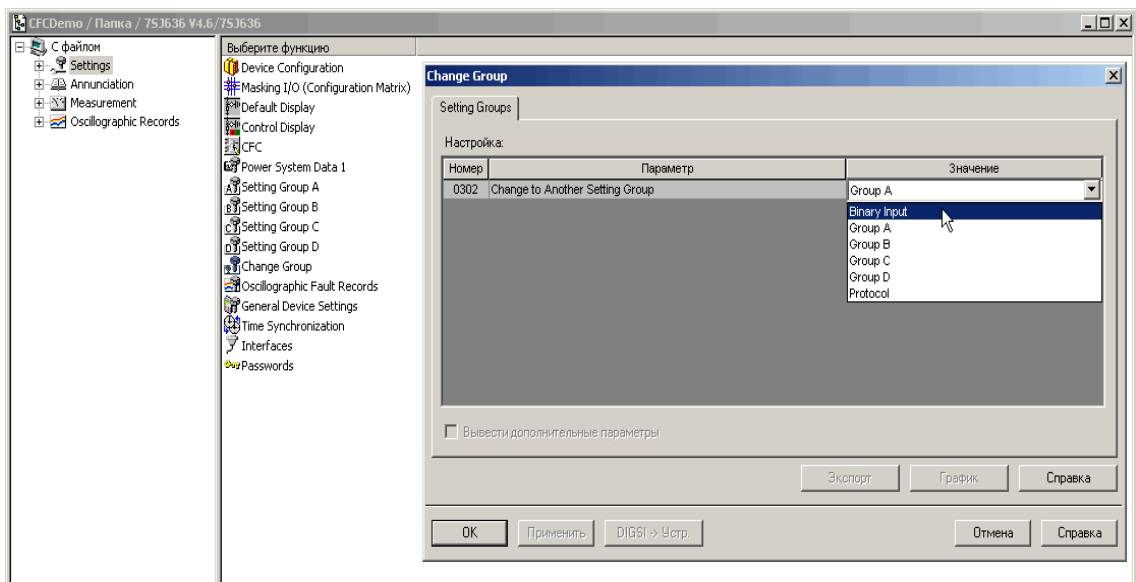
Praxis_01_02.gif

Рис. 3-1 Включение опции переключения набора уставок

Динамическое включение опции переключения набора уставок

Для динамического переключения набора уставок при помощи логики CFC, включите опцию переключения набора уставок через бинарный вход:

- Установите в окне **Change Group (Переключение набора уставок)** в параметре **Change to Another Setting Group (Переключиться на другой набор уставок)** опцию **Binary Input (Через бинарный вход)** или **Protokol (Через протокол)**, которая находится в списке выбора.



Praxis_01_11.gif

Рис. 3-2 Динамическое включение опции переключения набора уставок

Опция переключения набора уставок

Если опция переключения набора уставок активна, в матрице ранжирования DIGSI появляется новая группа **Grp Chge OPTION**.

The screenshot shows a configuration matrix titled "Settings - Masking I/O (Configuration Matrix) - CFC Demo / Папка / 75J636 V4.6/75J636". The matrix is organized into columns for "Информация" (Information), "Источник" (Source), and "Цель" (Target). The "Цель" section includes columns for "Вы" (Output) numbered 1-22, and "СД", "П", "С", "Л", "Д", "МУ". A group of rows under the heading "[Change Group]" includes entries like ">Set Group Bit0", ">Set Group Bit1", "Group A", "Group B", "Group C", and "Group D". The "Group A" row shows a '1' in the "Вх" column and an 'x' in the "С" column. The "Group B" row shows an 'x' in the "С" column. The "Group C" row shows an 'x' in the "С" column. The "Group D" row shows an 'x' in the "С" column. Other rows like "Start Motor" and "Device, General" also show 'x' marks in various columns.

Рис. 3-3 Опция переключения набора уставок в матрице ранжирования DIGSI

Определение активного набора уставок

Активный набор уставок (А, В, С и D на Рис. 3-4) определяется в группе **Grp Chge OPTION (Переключение набора уставок)** двумя информационными элементами **>Set Group Bit0** (P1 на Рис. 3-4) и **>Set Group Bit1** (P2 на Рис. 3-4):

Набор уставок **В** активен, если **>Set Group Bit0** равен **1** и **>Set Group Bit1** равен **0**.

	P 2	0	1
P 1	0	A	C
	1	B	D

Praxis_01_04.gif

Рис. 3-4 Активный набор уставок

Переключение активного набора уставок

Для того чтобы переключиться между наборами уставок А и В, достаточно изменить значение информационного элемента **>Set Group Bit0** (P1 на Рис. 3-4) в программе CFC:

- Заранжируйте информацию **>Set Group Bit0** на CFC как источник.

	Информация				Источники		Цель																													
	Но	Текст дисплея	НС	Тип	Вх	Сл	Вы																													
							1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	СД	П	Л	Д	МЧ			
Start Motor	F1 Start			IntSP	1																															
Device, General																																				
P.System Data 1																																				
Osc. Fault Rec.																																				
Change Group	>Set Group Bit0			SP																																
	>Set Group Bit1			SP																																
	Group A			IntSP																																
	Group B			IntSP																																
	Group C			IntSP																																
P.System Data 2																																				
50/51 Overcur.																																				

Рис. 3-5 Ранжирование информации >Set Group Bit0 на CFC как источник

Распределение опции переключения на различные классы приоритетов

Программа CFC переключается на набор уставок **В**, если

- все фазные токи становятся меньше 5% от номинального значения или
- в первые 10 секунд после запуска двигателя.

Фазные токи контролируются в CFC в классе приоритетов **MW_BEARB** (обработка измеряемых величин).

Для того чтобы переключить набор уставок по истечении 10 секунд, необходимо использовать таймер. Однако, предусмотренный блок **TIMER** (универсальный таймер) не может применяться для класса приоритетов CFC **MW_BEARB** (обработка измеряемых величин). Для этой цели необходим класс приоритетов CFC **PLC1_BEARB** (медленная логика Slow PLC).

В виду того, что требуются разные классы приоритетов, разделите программу CFC на две схемы CFC со следующими классами приоритетов:

- Схема CFC **ParaUmsch**:
Класс приоритетов **MW_BEARB** (обработка измеряемых величин):
Контроль фазных токов
- Схема CFC **ParaUmsch1**:
Класс приоритетов **PLC1_BEARB** (медленная логика Slow PLC):
Контроль времени запуска и переключение набора уставок.

**Связь между
разными
классами
приоритетов**

Разные классы приоритетов (например, MW_BEARB и PLC1_BEARB) можно соединить друг с другом через информационный элемент типа **Tagging ON / OFF (IE)**(Маркирование Вкл. / Выкл.):

В одном классе приоритетов (например, MW_BEARB в схеме CFC **ParaUmsch**) значение ранжируется на информационный элемент, в то время как в другом классе приоритетов (например, PLC1_BEARB в схеме CFC **ParaUmsch1**) эта информация обрабатывается.



Примечание:

Недопустимо использовать блоки в разных классах приоритетов в пределах одной схемы CFC. Используйте отдельную схему CFC для каждого класса приоритетов!

В схеме CFC вы можете использовать информационный элемент типа **Tagging ON / OFF (IE) (Маркирование Вкл. / Выкл.)** исключительно либо на правой границе схемы, либо на левой.

Использование информационного элемента

Как использовать информационный элемент для связи между классами приоритетов MW_BEARB и PLC1_BEARB:

- Вставьте новый информационный элемент типа **Tagging ON / OFF (IE) (Маркирование Вкл. / Выкл.)** в группу **Grp Chge OPTION (Переключение набора уставок)**. Информационный элемент вставляется с именем по умолчанию.
- Переименуйте новый информационный элемент на **I<5%**.
- Заранжируйте информационный элемент **I<5%** как **Источник** и как **Цель** на CFC.

Информация	Источники	Цель																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Start Motor	F1 Start	IntSP	1																				
Device, General																							
P System Data 1																							
Osc. Fault Rec.																							
Change Group	>Set Group Bit0	SP																					
	>Set Group Bit1	SP																					
	Group A	IntSP																					
	Group B	IntSP																					
	Group C	IntSP																					
	Group D	IntSP																					
	I<5%	IntSP																					
P System Data 2																							
50/51 Overcur.																							
67 Direct D/C																							

Рис. 3-6 Ранжирование информации как источник и как цель на CFC

Контроль фазных токов в программе CFC

Как создать функцию контроля фазных токов в программе CFC:

- Заранжируйте фазные токи на CFC как **Цель**.

Информация	Источники	Цель	
		Измеряемая величина	Значения ЭЭ
Osc. Fault Rec.			
Change Group			
P System Data 2			
50/51 Overcur.			
67 Direct D/C			
Measurement Superv			
Fault Locator			
Critl Authority			
Control Device			
Process Data			
	00601 Ia =	MV	
	00602 Ib =	MV	
	00603 Ic =	MV	
	00604 I1 =	MV	
	00605 I2 =	MV	
	00606 I3 =	MV	
	00621 Va =	MV	
	00622 Vb =	MV	
	00623 Vc =	MV	
	00624 Vab =	MV	
	00625 Vbc =	MV	
	00626 Vca =	MV	
	00627 Vb =	MV	
	00628 V1 =	MV	
	00630 V2 =	MV	
	00641 P =	MV	
	00642 Q =	MV	
	00645 S =	MV	
	00644 freq =	MV	
	00622 Vc =	MV	

Рис. 3-7 Ранжирование фазных токов на CFC как цель.

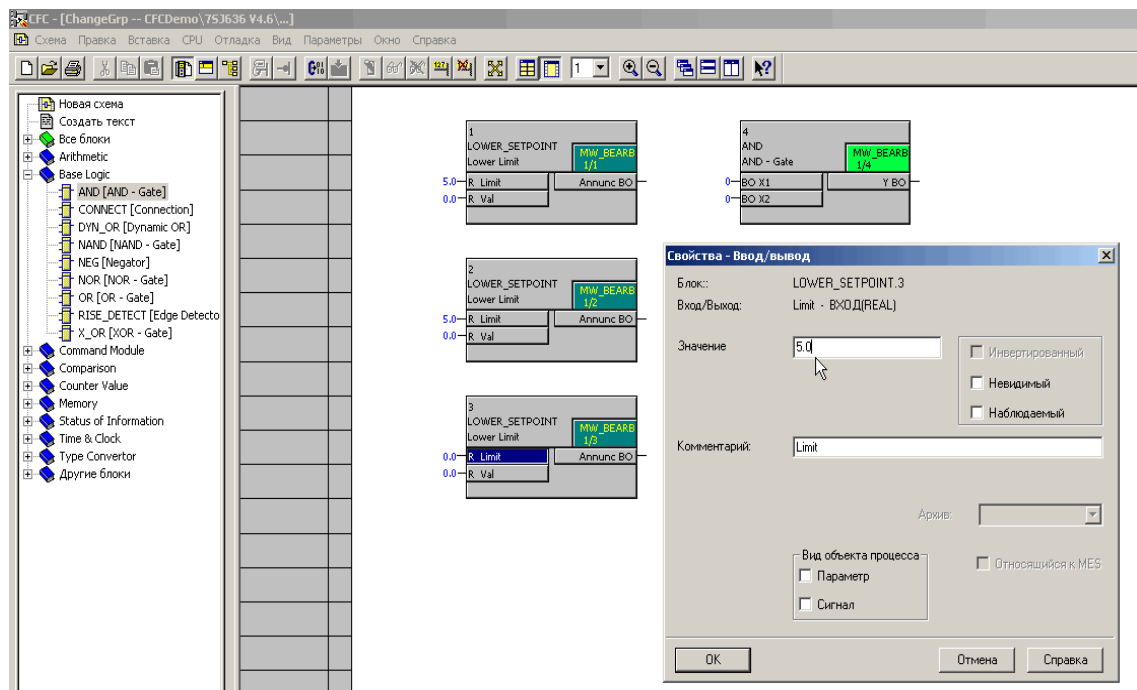
- Создайте в CFC новую схему под названием **ParaUmsch** и откройте эту схему CFC.
- Используйте пункт меню **Edit (Правка) → Run Sequence (Последовательность выполнения)** для определения класса приоритетов **MW_BEARB** в качестве стандартного класса приоритетов. Каждый новый блок вставляется в класс приоритетов **MW_BEARB**.
- Контроль фазного тока осуществляется при помощи блока **LOWER_SETPOINT** (нижний предел). Все три функции контроля соединяются при помощи блока **AND** (логический элемент И) и образуют, таким образом, информацию **I<5%** для дальнейшей связи с классом приоритетов **PLC1_BEARB**.
- Разместите блоки CFC.



Примечание:

При вставлении отдельных блоков CFC обратите внимание на заданную по умолчанию последовательность обработки блоков. Откорректируйте, если это необходимо, последовательность выполнения заданий, используя пункты меню **Edit (Правка) → Run Sequence (Последовательность выполнения)**.

- Установите для блоков **LOWER_SETPOINT** предельное значение на **5,0%** при помощи контекстного меню **Object Properties (Свойства объекта)**.



Praxis_01_08.gif

Рис. 3-8 Параметрирование предельного значения 5,0%

- Используйте контекстное меню **Number of I / Os (Количество входов / выходов)** для того, чтобы увеличить количества входов / выходов блока **AND** до 3.
- Соедините блоки друг с другом и с адресами на границах схемы. Соедините при этом выход блока **AND** с информационным элементом **I<5%** на правой границе схемы.

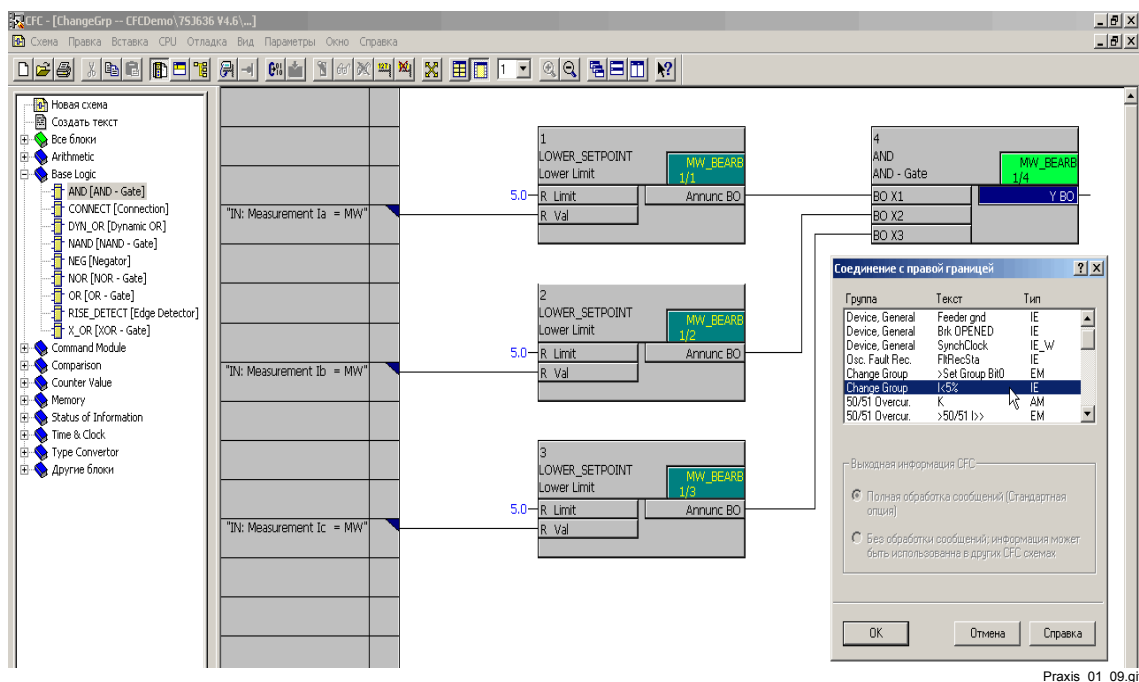


Рис. 3-9 Создание соединений между блоками и границей

Компиляция схемы CFC

После того, как вы создали схему CFC, вы можете откомпилировать ее при помощи пунктов меню **Chart (Схема) → Compile (Компилировать) → Charts as Program (Отобразить программу)**.

Контроль времени запуска и переключение набора уставок в программе CFC

Как контролировать время запуска и переключение набора уставок в программе CFC:

- Заранжируйте информацию **F1 Start (Запуск кнопкой F1)** на CFC как цель.

	Информация				Источник		Цель																															
	Номер	Текст дисплея	НС	Тип	Вх	Ф	С	Л	Вы																						С	Д	МЧ					
								1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	Сд	п	л	д	мч				
Start Motor		F1 Start		IntSP				1																											X (Ранжировано)			
Device, General																																			X (Не ранжировано)			
P System Data 1																																						
Osc. Fault Rec.																																						
Change Group	00007	>Set Group Bit0		SP																																		
	00008	>Set Group Bit1		SP																																		
		Group A		IntSP																																		
		Group B		IntSP																																		
		Group C		IntSP																																		
P System Data 2		I<5%		IntSP																																		

Рис. 3-10 Ранжирование информации Запуск кнопкой F1 на CFC как цель

- Создайте в CFC новую схему под названием **ParaUmsch1** и откройте эту схему.
- Используйте пункт меню **Edit (Правка) → Run Sequence (Последовательность выполнения)** для задания класса приоритетов **PLC1_BEARB** в качестве стандартного класса приоритетов. Каждый новый блок вставляется в класс приоритетов **PLC1_BEARB**.
- Перейдите к пустому листу существующей схемы CFC.
- Функция контроля времени запуска осуществляется с помощью блока **TIMER** (универсальный таймер), который запускается сообщением **Запуск кнопкой F1**. Пока таймер запущен, бинарный сигнал **1** подается на выход **QT1**. Соедините выходной сигнал таймера с сообщением **I<5%** из класса приоритетов **MW BEARB** при помощи блока **OR** (логическое ИЛИ). Сигнал на выходе блока **OR** управляет напрямую информацией **>Set Group Bit0** и переключает, таким образом, группу уставок.
- Разместите блоки CFC.



Примечание:

При использовании отдельных блоков CFC обратите внимание на заданную по умолчанию последовательность обработки блоков. Откорректируйте, если это необходимо, последовательность выполнения заданий, используя пункты меню **Edit (Правка) → Run Sequence (Последовательность выполнения)**.

**Примечание:**

В примере выше, блок **TIMER** (универсальный таймер) используется для контроля времени запуска двигателя. В зависимости от используемой версии устройства SIPROTEC, для осуществления функции контроля вы также можете использовать блоки **TIMER_SHORT** (простой таймер) или **LONG_TIMER** (таймер (макс. 1 193 ч)).

Блок **LONG_TIMER** (таймер (макс. 1 193 ч)) может также обрабатываться в классе приоритетов CFC **MW_BEARB** (обработка измеряемых величин). Используя данный блок, вы можете реализовать полное переключение набора уставок в классе приоритетов CFC **MW_BEARB** (обработка измеряемых величин). При этом вам не надо распределять переключение на различные классы приоритетов.

- Используя контекстное меню **Object Properties (Свойства объекта)**, установите значение для **TIMER T1x1ms** на **10000 мсек.**

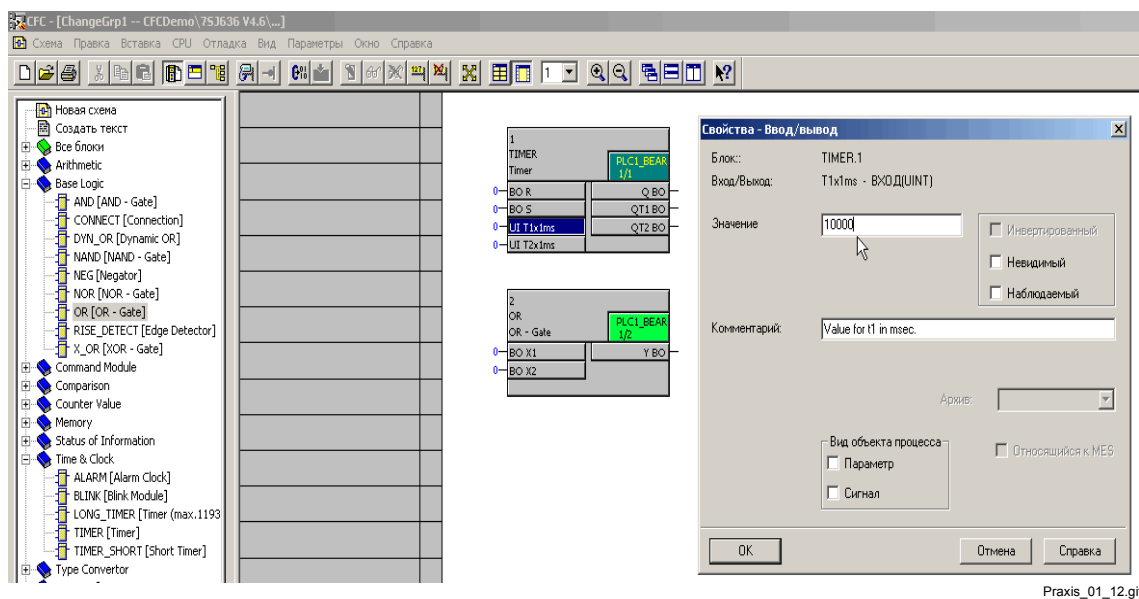


Рис. 3-11 Параметрирование предельного значения 5,0%

- Соедините блоки друг с другом и с адресами на границах схемы. При этом соедините вход блока **OR** с сигналом **I<5%** на левой границе схемы, а выход с сигналом **>Set Group Bit0** на правой границе схемы.

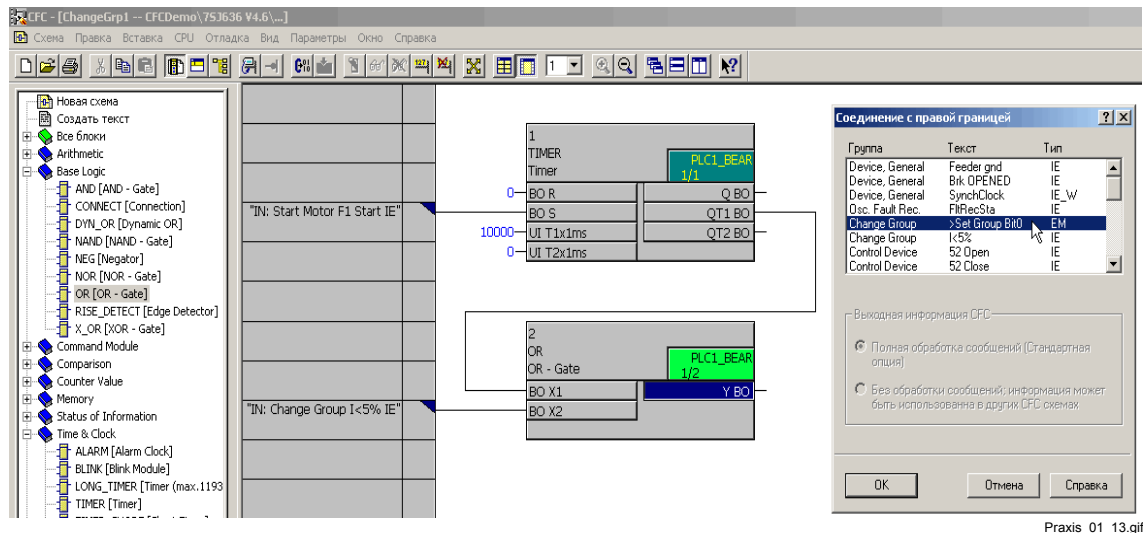


Рис. 3-12 Создание соединений между блоками и границей

Компиляция схемы CFC

После того, как вы создали схему CFC, вы можете откомпилировать ее при помощи пунктов меню **Chart (Схема) → Compile (Компилировать) → Charts as Program (Отобразить программу)**.

3.2 Мигающие светодиоды

Мигающие светодиоды часто используются для привлечения внимания к особому состоянию оборудования (например, критическое состояние с точки зрения безопасности). В устройствах SIPROTEC 4 вы можете использовать программу CFC для имитации механизма мигания.



Примечание:

Для устройств SIPROTEC, начиная с версии V4.5, может использоваться блок **BLINK (Мигание)**. Этот блок может обрабатываться во всех классах приоритетов.

Задача

Перед нами стоит следующая задача:

- Во многих случаях короткий импульс (например, нажатие функциональной кнопки F1) используется для перехода в аварийный режим. Этот режим отображается миганием светодиода.
- Следующий короткий импульс (например, нажатие функциональной кнопки F2) используется для возвращения из этого состояния, и, таким образом, для прекращения мигания светодиода.

Подготовка функциональных кнопок для обработки программой CFC

Как подготовить функциональные кнопки F1 и F2 для обработки, осуществляемой программой CFC:

- В матрице ранжирования DIGSI создайте новую группу под названием **LED Flash (Мигание светодиода)**.
- Вставьте новый информационный элемент типа **Tagging ON / OFF (IE) (Маркирование Вкл. / Выкл.)** в группу **LED Flash (Мигание светодиода)** для каждой функциональной кнопки. Информационные элементы вставляются с именем по умолчанию.
- Переименуйте новые информационные элементы на **F1 Flash (Мигание)** и **F2 NoFlash (Отключение мигания)**.
- Заранжируйте функциональную кнопку **1** на информационный элемент **F1 Flash (Мигание)**.
- Заранжируйте функциональную кнопку **2** на информационный элемент **F2 NoFlash (Отключение мигания)**.
- Заранжируйте информационные элементы **F1 Flash (Мигание)** и **F2 NoFlash (Отключение мигания)** на CFC как **Цель**.

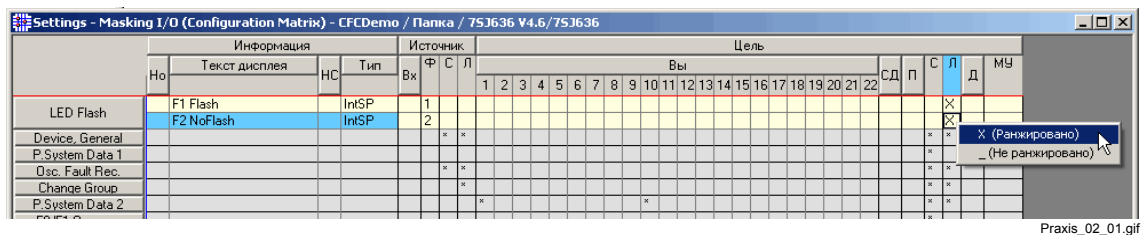


Рис. 3-13 Подготовка функциональных кнопок для обработки программой CFC

Подготовка управления светодиодами через программу CFC

Как подготовить управление светодиодами через программу CFC:

- Вставьте для светодиода новый информационный элемент типа **Tagging ON / OFF (IE) (Маркирование Вкл. / Выкл.)** в группу **LED Flash (Мигание светодиода)**. Информационный элемент вставляется с именем по умолчанию.
- Переименуйте новый информационный элемент на **LED Flash (Мигание светодиода)**.
- Заранжируйте информационный элемент **LED Flash (Мигание светодиода)** на CFC как **Источник**.
- Заранжируйте светодиод **1** на информационный элемент **LED Flash (Мигание светодиода)**. Выход должен быть установлен на **Б (Без запоминания)**.

	Информация			Источник			Цель																														
	Но	Текст дисплея	ИС	Тип	Вх	Ф	С	Л	Вы																												
LED Flash	F1 Flash		IntSP		1				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	СД	П	С	Л	Д	МУ	
	F2 NoFlash		IntSP		2																																
	LED Flash			IntSP																																	
Device, General										*	*																										
P. System Data 1																																					
Osc. Fault Rec.																																					
Change Group																																					
P. System Data 2																																					
50/51 Overv																																					

Рис. 3-14 Подготовка управления светодиодами через программу CFC

Частота мигания

Частота мигания зависит от следующих временных интервалов **Светодиод не горит** (время_{Выкл.}=250 мсек) и **Светодиод горит** (время_{Вкл.}=50 мсек).

Имитация мигания в программе CFC

Как имитировать мигание светодиодов в программе CFC:

- Создайте в CFC новую схему под названием **LEDFlash** и откройте эту схему.
- Используйте пункт меню **Edit (Правка) → Run Sequence (Последовательность выполнения)** для задания класса приоритетов **PLC1_BEARB** в качестве стандартного класса приоритетов. Каждый новый блок вставляется в класс приоритетов **PLC1_BEARB**.



Примечание:

Блок **TIMER** (универсальный таймер) работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**) и медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**).

- Мигание светодиодов осуществляется с помощью двух блоков типа **TIMER** (универсальный таймер):

Первый таймер запускается сигналом **F1 Flash**. Сразу после запуска внутренний таймер **T1** работает 250 мсек (время_{Выкл.}) и устанавливает в течение этого времени выход **QT1** на бинарный сигнал **1**.

Выходной сигнал **QT1** первого таймера служит сигналом запуска второго таймера. Здесь, однако, используется внутренний таймер **T2**, который не запускается до тех пор, пока не появится сигнал запуска на входе блока, т.е. точно по истечении времени $t_{\text{Выкл.}} = 250$ мсек. Таймер **T2** работает 50 мсек (время_{Выкл.}) и устанавливает соответствующий выход **QT2** равный **1**. Этот сигнал соединяется со светодиодом (**LED Flash (Мигание светодиода)**), который соответственно загорается. По истечении заданного времени светодиод опять гаснет.

Для того чтобы светодиод мигал, эта последовательность должна повторяться. Поэтому выход **QT2** второго таймера соединяется с входом запуска первого таймера при помощи блока **OR** (логического элемента ИЛИ). Это разрешенная обратная связь. Предупреждение, которое появляется при компиляции схемы CFC, не имеет значения.

Для того чтобы выключить мигание светодиода, соедините входы сброса этих двух таймеров с сигналом **F2 NoFlash**.

- Разместите блоки CFC.



Примечание:

При добавлении отдельных блоков CFC обратите внимание на заданную по умолчанию последовательность обработки блоков. Порядковый номер выполнения блока **OR** должен быть выше, чем порядковый номер выполнения второго блока **TIMER**. Откорректируйте, если это необходимо, последовательность выполнения заданий, используя пункты меню **Edit (Правка)** → **Run Sequence (Последовательность выполнения)**.



Примечание:

В рассмотренном примере, блок **TIMER** (универсальный таймер) используется для имитации функции мигания. В зависимости от версии устройства SIPROTEC, для осуществления функции контроля вы также можете использовать блоки **TIMER_SHORT** (простой таймер) или **LONG_TIMER** (таймер (макс. 1 193 ч)).

Блок **LONG_TIMER** (таймер (макс. 1 193 ч)) может обрабатываться во всех классах приоритетов CFC.

- Используя контекстное меню **Object Properties (Свойства объекта)**, задайте частоту мигания светодиода, установив для таймера 1 $T1 \times 1ms = 250$ мсек и для таймера 2 $T2 \times 1ms = 50$ мсек.

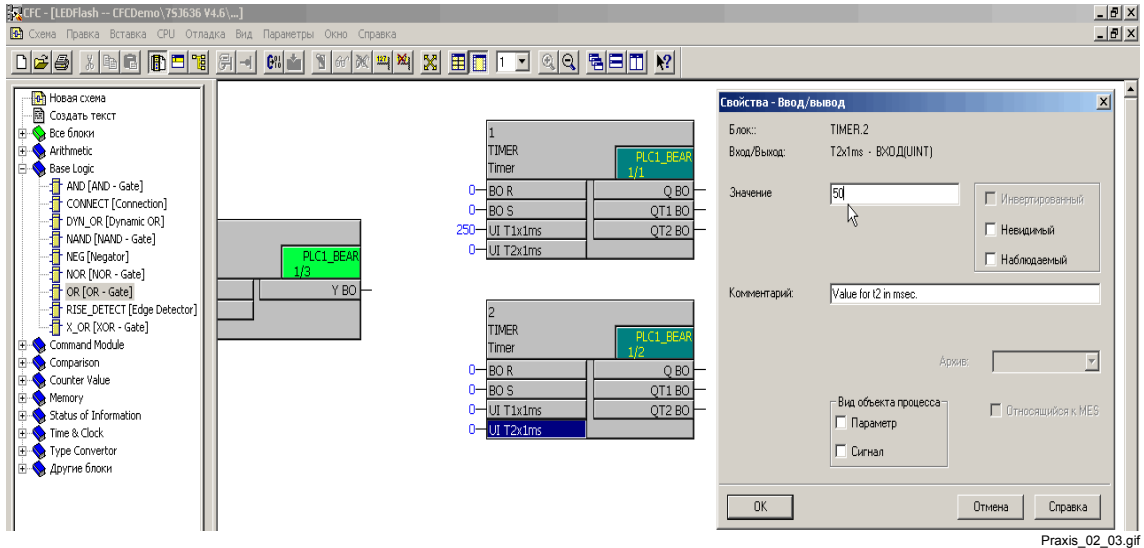


Рис. 3-15 Параметрирование таймера

- Соедините блоки друг с другом и с адресами на границах схемы см. Рис. 3-16.
Соедините вход установки **S** первого блока **TIMER** с информационным элементом **F1 Flash** на левой границе схемы.
Соедините входы сброса **R** двух блоков **TIMER** с информацией **F2 NoFlash** на левой границе схемы.
Соедините выход **QT2** второго блока **TIMER** с информационным элементом **LED Flash (Мигание светодиода)** на правой границе схемы.

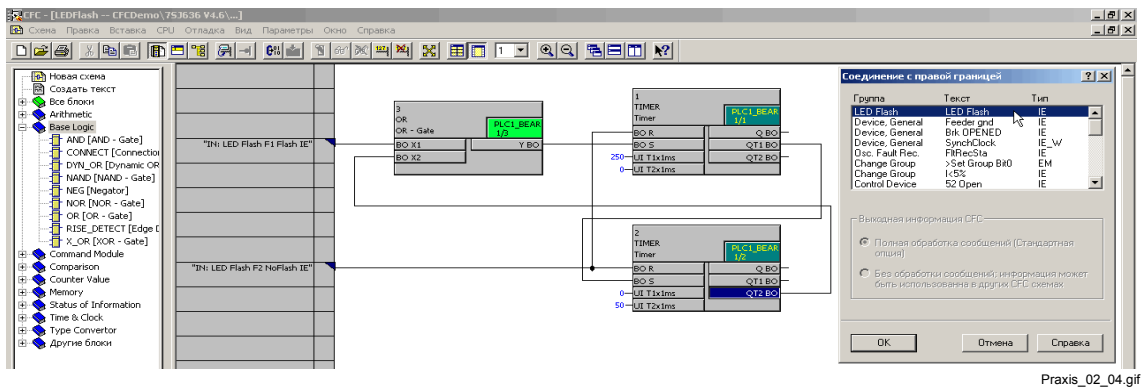


Рис. 3-16 Создание соединений между блоками и границей

Компиляция схемы CFC

После того, как вы создали частичную схему CFC, вы можете откомпилировать всю схему CFC при помощи пунктов меню **Chart (Схема) → Compile (Компилировать) → Charts as Program (Отобразить программу)**.



Примечание:

При компиляции схемы CFC будет выдано предупреждение, так как в схеме CFC существует обратная связь. В данном случае обратная связь разрешается, поэтому предупреждение может быть проигнорировано.

Непрерывный сигнал как сигнал запуска мигания

Если в качестве сигнала запуска мигания, вместо короткого импульса (например, функциональная кнопка F1) должен использоваться непрерывный сигнал (например, ключ управления), то этот непрерывный сигнал должен быть преобразован.

Непрерывный сигнал может быть преобразован при помощи дополнительного блока **TIMER** в два коротких сигнала, соответствующих сигналам **F1 Flash** и **F2 NoFlash**:

Преобразуемый сигнал соединяется с входом запуска этого дополнительного таймера. В параметрах **T1x1ms** и **T2x1ms** устанавливается значение 10 мсек:

Независимо от длительности сигнала, этот сигнал запускает таймер, и на выходе **QT1** появляется импульс на время 10 мсек. При пропадании сигнала дается короткий импульс (10 мсек) на выходе **QT2**. Поэтому выход **QT1** (соответствующий сигналу **F1 Flash**) соединен с входом установки **S** первого таймера. Выход **QT2** (соответствующий сигналу **F2 NoFlash**) соединен с входом сброса **R** обоих таймеров.

3.3 Обратная блокировка

Используя функцию **обратной блокировки**, вы можете создать логическую защиту сборных шин на основе токовой защитой с выдержкой времени.

Предполагается, что сборные шины состоят из вводной ячейки и отходящих присоединений. Типичным применением для сборных шин являются распределительные сети среднего напряжения. В сетях высокого и сверхвысокого напряжения этот принцип применяется редко. Здесь используется отдельная защита сборных шин.

Принцип

Этот принцип довольно прост. МТЗ отходящих присоединений (ступень срабатывания I>) блокирует токовую отсечку (**ступень срабатывания I>>**) в ячейке ввода, через бинарный вход.

Время выдержки токовой отсечки ввода устанавливается так, что гарантируется надежная блокировка от ступеней МТЗ, отходящих присоединений (50-100 мсек).

Токовая отсечка ввода должна быть установлена гораздо выше, чем ступень МТЗ отходящих присоединений, для обеспечения селективности работы защит.

Неисправность при обратной блокировке приводит к отключению сборных шин.

Поэтому, с точки зрения защиты, эта функция имеет большое значение для безопасности.

3.3.1 Заданное действие блокировки при коротком замыкании

Внешнее короткое замыкание на присоединении

Внешнее короткое замыкание приводит к срабатыванию МТЗ отходящего присоединения (ступень I>). Это срабатывание заранжировано на бинарный выход терминала присоединения и блокирует МТО (ступень I>>) в терминале ввода, через бинарный вход.

Таким образом, МТО (ступень I>>) не может производить отключение, даже если ее выдержка времени истекла. Короткое замыкание отключается максимальной токовой защитой присоединения, на котором произошло КЗ. С возвратом срабатывания МТЗ (ступень I>) блокировка снимается, так как короткое замыкание исчезло.

Короткое замыкание на сборных шинах

Токовая отсечка ввода устанавливается так, чтобы она надежно срабатывала при КЗ на сборных шинах. Короткое замыкание на сборных шинах не приводит к срабатыванию МТЗ (ступень I>) других отходящих присоединений. Таким образом, срабатывает только МТО (ступень I>>). По истечении установленной выдержки времени, выдается команда на отключение и питающий силовой выключатель отключается. Таким образом, КЗ на сборных шинах устранено.

3.3.2 Варианты выполнения обратной блокировки

Существуют два варианта выполнения обратной блокировки.

Вариант 1

Блокировка МТО (ступень I>>) терминала ввода осуществляется с помощью подачи напряжения на бинарный вход, на который заранжирована эта блокировка. К этому входу подключаются выходные реле срабатывания МТЗ (ступень I>) отходящих присоединений, соединенных параллельно. Срабатывание одного или нескольких контактов приводит к блокировке бинарного входа, и, таким образом, МТО (ступень I>>).

У этого метода есть следующий недостаток: Если МТЗ присоединения временно неисправна и на этом присоединении возникает КЗ, то блокировка не может быть произведена. В результате этого, произошло бы ошибочное отключение сборных шин. Второй метод блокировки предотвращает этот недостаток.

Вариант 2

Нормально закрытые контакты выходных реле на которые заранжировано срабатывания МТЗ (ступень I>) последовательно подключены к бинарному входу терминала ввода, на который заранжировано сообщение **Разрешение I>>**. В нормальном режиме этот вход находится под напряжением.

Если устройство ввода имеет только сообщение **Блокировка ТО** (ступень I>>), то бинарный вход устанавливается на активный без напряжения. МТО (ступень I>>) ввода имеет разрешение до тех пор, пока не сработала МТЗ (ступень I>) устройства присоединения.

При наличии КЗ на сборных шинах производится отключение. Но если срабатывает МТЗ (ступень I>) на отходящем присоединении, то контакт выходного реле размыкается и рвет последовательную цепь. Таким образом, происходит блокировка МТО (ступень I>>).

Преимуществом данного метода является то, что при неисправном устройстве цепь разрывается и осуществляется блокировка МТО (ступень I>>). Таким образом, избыточное срабатывание устройства при КЗ может быть предотвращено.

3.3.3 Обратная блокировка с помощью протокола IEC 61850

Обратная блокировка при использовании протокола IEC 61850 должна быть реализована **по второму варианту**.

Требования

Следующие требования должны быть соблюдены:

- ❑ Обратная блокировка реализуется при помощи **службы IEC 61850 GOOSE**.
- ❑ Максимальная токовая защита (МТЗ) ввода обладает функцией блокировки ступени I>>.
- ❑ Срабатывание МТЗ присоединения (ступень I>) другой МТЗ передается при помощи IEC 61850.
- ❑ Конфигурируется следующая характеристика передачи GOOSE:
 - First repetition time (время первого повторения) = 4 мсек
 - Last repetition time (время последнего повторения) = 1 сек.

Реализация

Предлагается следующее:

Сообщения о срабатывании МТЗ (ступень I>) от отдельных устройств защиты присоединения, включая соответствующий Quality attribute (атрибут качества) и метку времени, передаются в виде сообщения GOOSE с характеристикой передачи, указанной выше.

Все эти сообщения GOOSE принимаются устройствами ввода. Сообщения о срабатывании и соответствующая информация о состоянии передаются при этом во внутреннюю логику CFC со статусом **достоверно**.

При помощи схемы CFC имитируется описанная выше стандартная процедура, (**вариант 2**):

- ❑ Каждое сообщение о срабатывании МТЗ (ступень I>) приводит к блокировке МТО (ступень I>>).
- ❑ Если максимальная токовая защита присоединения выходит из строя, то по истечении тайм-аута (2 x последнее время повторения), состояние соответствующего сообщения о срабатывании ступени устанавливается в устройстве ввода на **недостоверно**. Это изменение состояния также приводит к блокировке МТО (ступень I>>).

При помощи данной логики имитируется действие, описанное в **варианте 2**: запуск устройства защиты или выход устройства из строя приводит к блокировке ступени I>> и предотвращает избыточное срабатывание ввода МТЗ.

3.3.4 Схема: обратная блокировка в виде схемы CFC

При помощи показанной ниже схемы CFC, вы можете имитировать логику обратной блокировки в устройстве SIPROTEC.

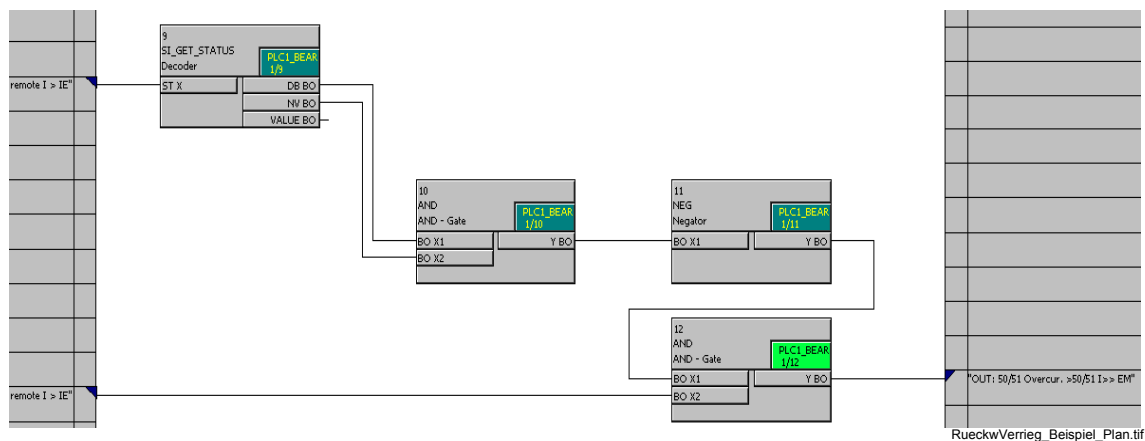
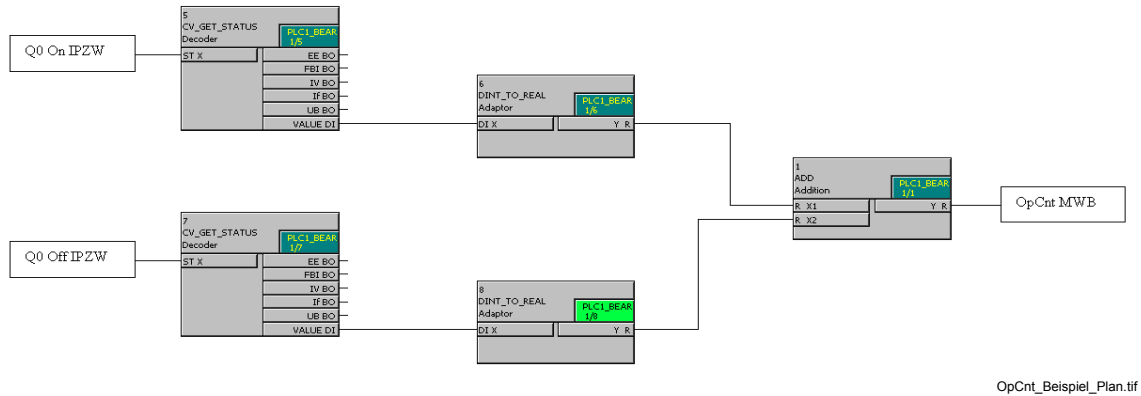


Рис. 3-17 Пример использования обратной блокировки, фрагмент схемы CFC

3.4 Подсчёт циклов переключения

При помощи показанной ниже схемы CFC, в устройстве SIPROTEC могут быть подсчитаны циклы переключения всех управляемых коммутационных объектов.



OpCnt_Beispiel_Plan.tif

Рис. 3-18 Пример подсчёта циклов переключения, фрагмент схемы CFC

Принцип работы

Q0 On PMV и **Q0 Off PMV** представляют собой счетно-импульсные величины, чьими источниками (в ранжировании CFC) является старший / младший байт двухпозиционного сообщения **Q0 ON/OFF**.

Каждая из счетно-импульсных величин отделена от информации о состоянии блоком **CV_GET_STATUS**.

Используя преобразователь типа **DINT_TO_REAL**, счетно-импульсные величины подготавливаются для суммирования при помощи арифметического блока **ADD**.

ADD суммирует количество операций Вкл. / Выкл. и выдает сумму на счетчик переключений **OpCnt MVU**.

Обзор

Данная глава содержит подробное описание блоков программы DIGSI CFC.



Примечание:

Версия устройства SIPROTEC определяет, какие блоки имеются в устройстве и, таким образом, в DIGSI CFC.



Примечание:

Для отображения блоков CFC, приведенных в качестве примеров в данной главе, в настройке **Block width (Ширина блока)**: была выбрана опция **wide (широкий)**:

- В редакторе CFC нажмите на **Options (Параметры)** → **Customize (Настроить)** → **Block / Sheet Bar Settings (Панели блока / бланка)**.
- В открывшемся окне отметьте опцию **wide (широкий)** в настройке **Block width (Ширина блока)** и подтвердите ваш выбор, нажав на **ОК**.

Содержание

4.1	Типы данных	74
4.2	Arithmetic (Арифметика)	77
4.3	Base Logic (Основная логика)	84
4.4	Status of Information (Состояние информации)	101
4.5	Memory (Память)	121
4.6	Command Module (Команды управления)	137
4.7	Type Convertor (Преобразователи типа)	159
4.8	Comparison (Сравнение)	182
4.9	Counter Value (Счетно-импульсная величина)	196
4.10	Time & Clock (Время и тактовые импульсы)	199

4.1 Типы данных

Существуют следующие типы данных в DIGSI CFC:

Таблица 4-1 Типы данных в DIGSI CFC

Тип	Представление в DIGSI	Значение	Диапазон величин
BOOL	BO	Логическая величина (двоичная величина)	0 (ложная), 1 (истинная)
DINT	DI	Целое число со знаком (32 разрядное)	-2147483632 до 2147483632 (в DIGSI -2147483648 до 2147483647) Примечание: Ограниченный в DIGSI диапазон величин вытекает из определения дополнительных величин как информации о состоянии (См. Таблицу 4-2).
INT	I	Адрес (например, адрес команды)	(Диапазон величин задается и вводится DIGSI.)
REAL	R	Число с плавающей запятой	-3.402823466e ⁺³⁸ до 3.402823466e ⁺³⁸
STRUCT	ST	Структура данных	Различный (Содержащаяся информация может быть обработана при помощи блоков состояния информации).
UINT	UI	Целое число без знака (16 разрядное)	0 до 65535
WORD	W	Битовое поле (например, для двухпозиционных сообщений)	16#0000 до 16#FFFF

Информация о состоянии у типа данных DINT

У типа данных DINT, помимо действительного диапазона величин, в DIGSI CFC определяются дополнительные величины в качестве информации о состоянии:

Таблица 4-2 Информация о состоянии у типа данных DINT в DIGSI CFC

Величина DINT	Состояние	Значение
-2147483647	OVERFLOW_NEG	Величина ниже заданного диапазона
-2147483646	NOT_DEFINED	Неопределенная величина
2147483644	LIVE_ZERO	Недостовверная величина, например, в результате обрыва провода
2147483645	NOT_CALCULATED	Вычисление не произведено
2147483646	INVALID	Недостовверная величина
2147483647	OVERFLOW_NEG	Величина ниже заданного диапазона

**Примечание:**

В программе DIGSI, диапазон величин у типа данных DINT **не** ограничен. Используя информацию в DIGSI CFC, обратите внимание на то, что вы **осознанно** устанавливаете информацию о состоянии.

Информация о состоянии у типа данных REAL

У типа данных REAL имеется идентичная информация о состоянии, как и у типа данных DINT. Однако используемые величины состояния, находятся вне действительного диапазона величин и не могут, поэтому быть представлены в виде чисел.

Структуры данных у типа данных STRUCT

Структуры данных типа данных STRUCT состоят из двух элементов: величина (VAL) и состояние (STAT):

- VAL [BOOL] и STAT [WORD]
например, выход Y у блока SI_SET_STATUS
- VAL [WORD] и STAT [WORD]
например, выход Y у блока DI_SET_STATUS
- VAL [DINT] и STAT [WORD]
например, выход Y у блока MV_SET_STATUS



Примечание:

Структуры данных могут быть отображены при помощи контекстного меню **Object Properties (Свойства объекта)** соответствующего соединения блока.

4.2 Arithmetic (Арифметика)

При помощи арифметических блоков типа **Arithmetic** вы можете обрабатывать измеряемые величины типа REAL и производить вычисления.

Существуют следующие арифметические блоки:

- **ABSVALUE** (абсолютная величина)
- **ADD** (сложение)
- **DIV** (деление)
- **MUL** (умножение)
- **SQUARE_ROOT** (извлечение квадратного корня)
- **SUB** (вычитание)

4.2.1 ABSVALUE

Функция

Блок **Absolute Value (Абсолютное значение)** формирует на выходе **AbsVal** абсолютное значение измеряемой величины на входе **Val**.

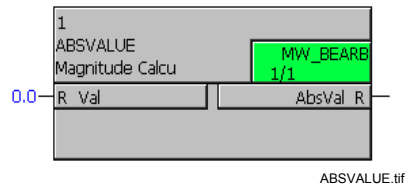


Рис. 4-1 Блок ABSVALUE

Назначение входов / выходов

Блок **ABSVALUE** имеет следующее назначение входов / выходов:

Таблица 4-3 Назначение входов / выходов блока ABSVALUE

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Val	REAL	Значение	0,0
Выходы:	AbsVal	REAL	Абсолютное значение	0,0

4.2.2 ADD

Функция

Блок **Addition (Сложение)** используется для сложения двух величин (например, измеряемых величин) **X1** и **X2**. Результат сложения выводится на выходе **Y**.

При помощи контекстного меню блока вы можете увеличить количество входов до 120:

- Щелкните правой кнопкой мыши по блоку и выберите из открывшегося контекстного меню **количество входов / выходов**.
- Введите необходимое вам **количество** и подтвердите ваш ввод, нажав на **ОК**.

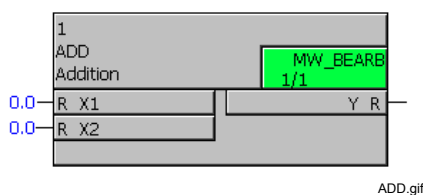


Рис. 4-2 Блок ADD

Назначение входов / выходов

Блок **ADD** имеет следующее назначение входов / выходов:

Таблица 4-4 Назначение входов / выходов блока ADD

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	REAL	Слагаемое 1	0,0
	X2	REAL	Слагаемое 2	0,0
Выходы:	Y	REAL	Результат сложения ($Y = X1 + X2$)	0,0

4.2.3 DIV

Функция

При помощи блока **Division (Деление)** вы можете разделить величину **X1** на величину **X2** (например, измеряемые величины). Результат деления выводится на выходе **Y**.

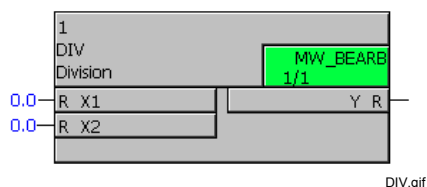


Рис. 4-3 Блок DIV

Назначение входов / выходов

Блок **DIV** имеет следующее назначение входов / выходов:

Таблица 4-5 Назначение входов / выходов блока DIV

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	REAL	Делимое	0,0
	X2	REAL	Делитель	0,0
Выходы:	Y	REAL	Результат деления ($Y = X1 / X2$)	0,0

4.2.4 MUL

Функция

Блок **Multiplication (Умножение)** используется для умножения двух величин **X1** и **X2**. Результат умножения выводится на выходе **Y**.

При помощи контекстного меню блока вы можете увеличить количество входов до 120:

- Щелкните правой кнопкой мыши по блоку и выберите из открывшегося контекстного меню **количество входов / выходов**.
- Введите необходимое вам **количество** и подтвердите ваш ввод, нажав на **ОК**.

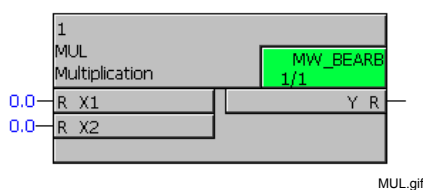


Рис. 4-4 Блок MUL

Назначение входов / выходов

Блок **MUL** имеет следующее назначение входов / выходов:

Таблица 4-6 Назначение входов / выходов блока MUL

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	REAL	Множитель 1	0,0
	X2	REAL	Множитель 2	0,0
Выходы:	Y	REAL	Результат умножения ($Y = X1 \times X2$)	0,0

4.2.5 SQUARE_ROOT

Функция

Блок **Square Root Extractor (Извлечение квадратного корня)** используется для вычисления квадратного корня из подкоренного числа **X**. Если число **X** меньше 0, то на выходе **ERR** появляется 1 и на выходе **Y** выводится 0,0.

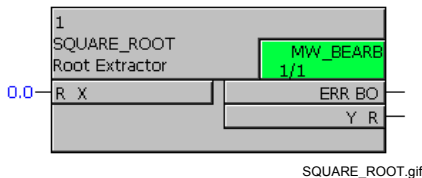


Рис. 4-5 Блок SQUARE_ROOT

Назначение входов / выходов

Блок **SQUARE_ROOT** имеет следующее назначение входов / выходов:

Таблица 4-7 Назначение входов / выходов блока SQUARE_ROOT

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	REAL	Подкоренное число	0,0
Выходы:	ERR	BOOL	Выход ошибки (устанавливается на 1, если $X < 0$)	0
	Y	REAL	Результат извлечения корня ($Y = \text{SQR}(X)$)	0,0

4.2.6 SUB

Функция

При помощи блока **Subtraction (Вычитание)** вы можете вычесть величину **X2** из величины **X1** (например, измеряемые величины). Результат вычитания выводится на выходе **Y**.

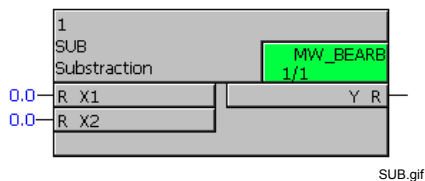


Рис. 4-6 Блок SUB

Назначение входов / выходов

Блок **SUB** имеет следующее назначение входов / выходов:

Таблица 4-8 Назначение входов / выходов блока SUB

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	REAL	Уменьшаемое	0,0
	X2	REAL	Вычитаемое	0,0
Выходы:	Y	REAL	Результат вычитания ($Y = X1 - X2$)	0,0

4.3 Base Logic (Основная логика)

Используя блоки основной логики, вы можете соединять и обрабатывать логические сигналы (булевы).

Существуют следующие блоки основной логики:

- **AND** (логический элемент И)
- **CONNECT** (соединение)
- **DYN_OR** (динамический логический элемент ИЛИ)
- **NAND** (логический элемент И НЕ)
- **NEG** (инвертор)
- **NOR** (логический элемент НЕ - ИЛИ)
- **OR** (логический элемент ИЛИ)
- **RISE_DETECT** (детектор фронта)
- **X_OR** (логический элемент исключающего ИЛИ)

4.3.1 AND

Функция

Блок **AND Gate (Логический элемент И)** соединяет все входные сигналы при помощи логической функции И и формирует из них выходной сигнал.

При помощи контекстного меню блока вы можете увеличить количество входов до 120:

- Щелкните правой кнопкой мыши по блоку и выберите из открывшегося контекстного меню **количество входов / выходов**.
- Введите необходимое вам **количество** и подтвердите ваш ввод, нажав на **ОК**.

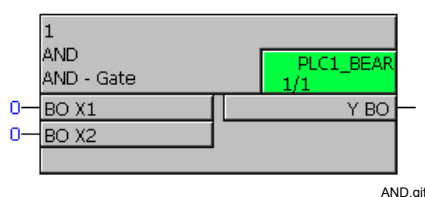


Рис. 4-7 Блок AND



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **AND** имеет следующее назначение входов / выходов:

Таблица 4-9 Назначение входов / выходов блока AND

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	BOOL	Вход	0
	X2	BOOL	Вход	0
Выходы:	Y	BOOL	Выход ($Y = X1 \text{ AND } X2$)	0

Таблица 4-10 Таблица истинности блока AND

X1 ВХОД	X2 ВХОД	Y ВЫХОД
0	0	0
0	1	0
1	0	0
1	1	1

4.3.2 CONNECT (Соединение)

Функция

Блок **Connection (Соединение)** позволяет вам напрямую соединить сигналы правой и левой границы (например, вход CFC с выходом CFC).

Без этого блока прямое соединение между двумя границами невозможно.

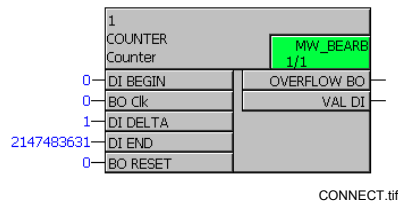


Рис. 4-8 Блок CONNECT



Примечание:

Блок **Connection** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**), медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классе приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

**Назначение
входов / выходов**

Блок **CONNECT** имеет следующее назначение входов / выходов:

Таблица 4-11 Назначение входов / выходов блока CONNECT

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	BOOL	Вход (к левой границе)	0
Выходы:	Y	BOOL	Выход (к правой границе)	0

4.3.3 DYN_OR

Функция

Блок **Dynamic OR Gate (Динамическое ИЛИ)** может использоваться для комбинирования сообщений.

В отличие от логического элемента ИЛИ, динамический элемент ИЛИ показывает на выходе **каждое** изменение поступающего входного сигнала.

Во время запуска устройства, на выходе выдается двухпозиционное сообщение **промежуточное положение (00)**.

Если во время обработки схемы на входах нет никаких сигналов, то выдается двухпозиционное сообщение **OFF (Выкл.) (01)**.

Если же во время обработки схемы на выходе имеется сигнал, то выдается двухпозиционное сообщение **ON (Вкл.) (10)**.

Если помимо уже имеющегося на входе сигнала поступает еще один сигнал на другой вход, то на выходе устанавливается **промежуточное положение (11)**. Изменение сигнала выдается в виде двухпозиционного сообщения **ON (Вкл.) (10)** только при следующей обработке схемы.

Логический элемент **Dynamic OR** имеет 5 входов. Если вам необходимо больше 5 входов, вы можете последовательно соединить несколько динамических элементов OR. Для этого соответствующее двухпозиционное сообщение на выходе предшествующего блока **DYN_OR** необходимо декодировать с помощью блока **DM_DECODE**, а выходной сигнал **ON (Вкл.) (10)** должен быть соединен с входом следующего динамического элемента OR.

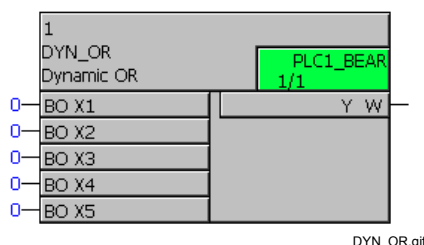


Рис. 4-9 Блок DYN_OR



Примечание:

Блок **Dynamic OR Gate** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**), медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).

**Примечание:**

Для соединения **быстрых сигналов** с левой границей схемы CFC в классе приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **DYN_OR** имеет следующее назначение входов / выходов:

Таблица 4-12 Назначение входов / выходов блока DYN_OR

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	BOOL	Входной сигнал	0
	X2	BOOL	Входной сигнал	0
	X3	BOOL	Входной сигнал	0
	X4	BOOL	Входной сигнал	0
	X5	BOOL	Входной сигнал	0
Выход:	Y	WORD	Выходной сигнал как двухпозиционное сообщение	16#0000

4.3.4 NAND

Функция

Блок **NAND gate (Логический элемент И НЕ)** соединяет все входные сигналы при помощи логической функции И НЕ и формирует из них выходной сигнал.

При помощи контекстного меню блока вы можете увеличить количество входов до 120:

- Щелкните правой кнопкой мыши по блоку и выберите из открывшегося контекстного меню **количество входов / выходов**.
- Введите необходимое вам **количество** и подтвердите ваш ввод, нажав на **ОК**.

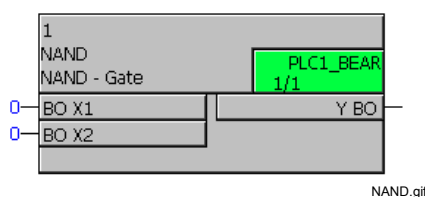


Рис. 4-10 Блок NAND



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатываемого класса приоритетов.

Назначение входов / выходов

Блок **NAND** имеет следующее назначение входов / выходов:

Таблица 4-13 Назначение входов / выходов блока NAND

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	BOOL	Вход	0
	X2	BOOL	Вход	0
Выход:	Y	BOOL	Выход ($Y = \text{NEG}(X1 \text{ AND } X2)$)	0

Таблица 4-14 Таблица истинности блока NAND

X1 ВХОД	X2 ВХОД	Y ВЫХОД
0	0	1
0	1	1
1	0	1
1	1	0

4.3.5 NEG

Функция

Блок **Negator (Логическое НЕ)** инвертирует входной сигнал и формирует из него выходной сигнал.

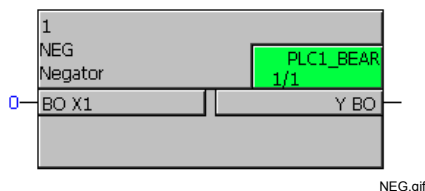


Рис. 4-11 Блок NEG



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **NEG** имеет следующее назначение входов / выходов:

Таблица 4-15 Назначение входов / выходов блока NEG

	Название	Тип данных	Примечание	Значение по умолчанию
Вход:	X1	BOOL	Вход	0
Выход:	Y	BOOL	Выход ($Y = \text{NEG}(X1)$)	1

Таблица 4-16 Таблица истинности блока NEG

X1 вход	Y выход
0	1
1	0

4.3.6 NOR

Функция

Блок **NOR Gate (ИЛИ НЕ)** соединяет все входные сигналы при помощи логической функции ИЛИ НЕ и формирует из них выходной сигнал.

При помощи контекстного меню блока вы можете увеличить количество входов до 120:

- Щелкните правой кнопкой мыши по блоку и выберите из открывшегося контекстного меню **количество входов / выходов**.
- Введите необходимое вам **количество** и подтвердите ваш ввод, нажав на **ОК**.

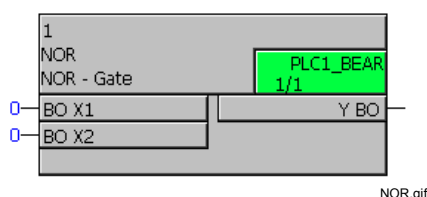


Рис. 4-12 Блок NOR



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **NOR** имеет следующее назначение входов / выходов:

Таблица 4-17 Назначение входов / выходов блока NOR

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	BOOL	Вход	0
	X2	BOOL	Вход	0
Выходы:	Y	BOOL	Выход (Y = NEG(X1 OR X2))	0

Таблица 4-18 Таблица истинности блока NOR

X1 ВХОД	X2 ВХОД	Y ВЫХОД
0	0	1
0	1	0
1	0	0
1	1	0

4.3.7 OR

Функция

Блок **OR Gate (ИЛИ)** соединяет все входные сигналы при помощи логической функции ИЛИ и формирует из них выходной сигнал.

При помощи контекстного меню блока вы можете увеличить количество входов до 120:

- Щелкните правой кнопкой мыши по блоку и выберите из открывшегося контекстного меню **количество входов / выходов**.
- Введите необходимое вам **количество** и подтвердите ваш ввод, нажав на **ОК**.

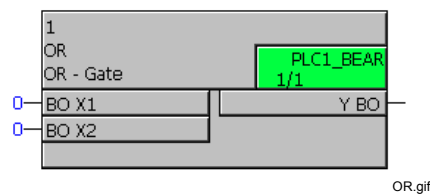


Рис. 4-13 Блок OR



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **OR** имеет следующее назначение входов / выходов:

Таблица 4-19 Назначение входов / выходов блока OR

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	BOOL	Вход	0
	X2	BOOL	Вход	0
Выход:	Y	BOOL	Выход (Y = X1 OR X2)	0

Таблица 4-20 Таблица истинности блока OR

X1 ВХОД	X2 ВХОД	Y ВЫХОД
0	0	0
0	1	1
1	0	1
1	1	1

4.3.8 RISE_DETECT

Функция

Блок **Rise Detector (Детектор фронта)** показывает на выходах **RISING (НАРАСТАЮЩИЙ)** и **FALLING (ПАДАЮЩИЙ)** во время обработки схемы, на входе **D** имеется изменение положительного или отрицательного фронта сигнала.

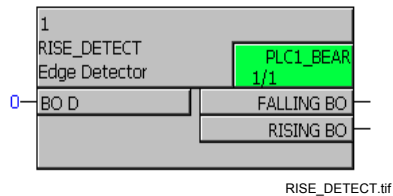


Рис. 4-14 Блок RISE_DETECT



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **RISE_DETECT** имеет следующее назначение входов / выходов:

Таблица 4-21 Назначение входов / выходов блока RISE_DETECT

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	D	BOOL	Вход детектора фронта	0
Выходы:	ПАДАЮЩИЙ	BOOL	Обнаружено отрицательное изменение фронта	0
	НАРАСТАЮЩИЙ	BOOL	Обнаружено положительное изменение фронта	0

4.3.9 X_OR

Функция

Блок **XOR Gate (исключающее ИЛИ)** соединяет все входные сигналы при помощи логической функции исключающего ИЛИ и формирует из них выходной сигнал.

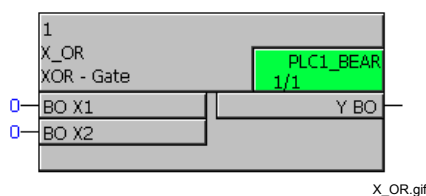


Рис. 4-15 Блок X_OR



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **X_OR** имеет следующее назначение входов / выходов:

Таблица 4-22 Назначение входов / выходов блока X_OR

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	BOOL	Вход	0
	X2	BOOL	Вход	0
Выходы:	Y	BOOL	Выход ($Y = X1 \text{ XOR } X2$)	0

Таблица 4-23 Таблица истинности блока X_OR

X1 ВХОД	X2 ВХОД	Y ВЫХОД
0	0	0
0	1	1
1	0	1
1	1	0

4.4 Status of Information (Состояние информации)

При помощи блоков состояния информации вы можете считывать, устанавливать и обрабатывать состояние сообщения, измеряемой или счетно-импульсной величины в устройстве SIPROTEC.

Существуют следующие возможные состояния:

Таблица 4-24 Обзор состояний информации

Вход / выход CFC	Длинный текст CFC	Журналы сообщений DIGSI	IEC 61850	Величина в устройстве	Значение
Сообщения					
NV	Не обновлена	NA	oldData	0x0040h	Величина неактуальна. Например, после запуска, при нарушении связи, при отсутствии ранжирования (или при ранжировании на несуществующий или неисправный модуль).
DB	Включенная блокировка от дребезга	FS	oscillatory	0x0002h	Блокировка от дребезга включена (бинарный вход заблокирован).
Измеряемые величины					
NV	Не обновлена	NA	oldData	0x0040h	Величина неактуальна. Например, после запуска, при нарушении связи, при отсутствии ранжирования (или при ранжировании на несуществующий или неисправный модуль).
OF	Переполнение	UL	overflow	0x0001h	Величина недостоверна в результате переполнения буфера.
UG	Недостоверна	UG		0x0080h	Величина недостоверна. Например, сбой при обработке измеряемой величины (переполнение).

Таблица 4-24 Обзор состояний информации

Вход / выход CFC	Длинный текст CFC	Журналы сообщений DIGSI	IEC 61850	Величины в устройстве	Значение
Измеряемые величины					
NV	Не обновлена	NA	oldData	0x0040h	Величина неактуальна. Например, после запуска, при нарушении связи, при отсутствии ранжирования (или при ранжировании на несуществующий или неисправный модуль).
IF	Неверна в результате сброса		substituted	0x0020h	Был произведен первоначальный запуск. Счетно-импульсная величина была установлена на 0.
FBI	Переполнение		overflow	0x0100h	Последнее считывание привело к переполнению буфера счетно-импульсных величин.
IV	Недостоверна		invalid	0x0200h	Величина недостоверна. Например, во время запуска. Был произведен сброс. Счетно-импульсные величины возможно утеряны.
EE	Внешнее повреждение	EE	failure	0x0400h	Образ входа внешнего повреждения.
UB	Переключение бита передачи			0x0800h	Состояние изменяется при каждой передаче.

Существуют следующие виды блоков:

- **CV_GET_STATUS** (дешифратор)
- **DI_GET_STATUS** (дешифратор)
- **DI_SET_STATUS** (шифратор)
- **MV_GET_STATUS** (дешифратор)
- **MV_SET_STATUS** (шифратор)
- **SI_GET_STATUS** (дешифратор)
- **SI_SET_STATUS** (шифратор)
- **ST_AND** (логический элемент И) с состоянием
- **ST_NOT** (элемент НЕ) с состоянием
- **ST_OR** (логический элемент ИЛИ) с состоянием

4.4.1 CV_GET_STATUS

Функция

Блок **CV_GET_STATUS** расшифровывает счетно-импульсную величину. Выходы предоставляют два информационных элемента: величину и соответствующее состояние.

Данный блок предназначен для соединения с **левой** границей для считывания заранжированных счетно-импульсных величин и их состояний.

Структура счетно-импульсной величины на входе **X** разбита на счетно-импульсную величину **VALUE (ВЕЛИЧИНА)** и следующую информацию о состоянии:

- EE** (внешнее повреждение)
- FBI** (переполнение)
- IV** (недостоверна)
- IF** (неверна из-за сброса)
- UB** (переключение бита передачи)

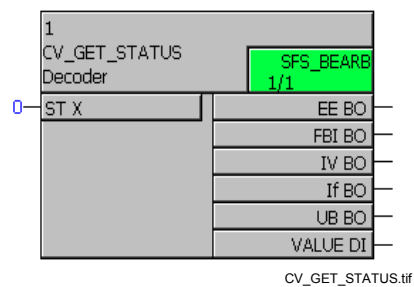


Рис. 4-16 Блок CV_GET_STATUS

Назначение входов / выходов

Блок **CV_GET_STATUS** имеет следующее назначение входов / выходов:

Таблица 4-25 Назначение входов / выходов блока CV_GET_STATUS

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	ST	Счетно-импульсная величина с состоянием	(0)

Таблица 4-25 Назначение входов / выходов блока CV_GET_STATUS

Выходы:	EE	BOOL	Внешнее повреждение	0
	FBI	BOOL	Переполнение	0
	IV	BOOL	Недостоверна	0
	IF	BOOL	Неверна в результате сброса	0
	UB	BOOL	Переключение бита передачи	0
	VALUE	DINT	Счетно-импульсная величина	0

4.4.2 DI_GET_STATUS

Функция

Блок **DI_GET_STATUS** расшифровывает состояние двухпозиционного сообщения. Выходы предоставляют два информационных элемента: величину и соответствующее состояние.

Данный блок предназначен для соединения с **левой** границей для обработки двухпозиционных сообщений и их состояний.

Структура двухпозиционного сообщения на входе **X** разбита на величину двухпозиционного сообщения **VALUE (ВЕЛИЧИНА)** и следующую информацию о состоянии:

- DB** (блокировка от дребезга)
- NV** (неактуальна)

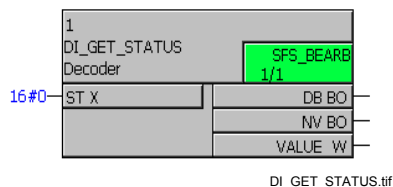


Рис. 4-17 Блок DI_GET_STATUS

Назначение входов / выходов

Блок **DI_GET_STATUS** имеет следующее назначение входов / выходов:

Таблица 4-26 Назначение входов / выходов блока DI_GET_STATUS

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	ST	Двухпозиционное сообщение с состоянием	(0)
Выходы:	DB	BOOL	Включенная блокировка от дребезга	0
	NV	BOOL	Не обновлена	0
	VALUE	WORD	Двухпозиционное сообщение	16#0000

4.4.3 DI_SET_STATUS

Функция

Блок **DI_SET_STATUS** формирует двухпозиционное сообщение с состоянием.

Данный блок предназначен для соединения с **правой** границей для установки двухпозиционных сообщений с их состоянием.

Структура двухпозиционного сообщения на выходе **Y** формируется в зависимости от величины двухпозиционного сообщения **VALUE (ВЕЛИЧИНА)** и следующей информации о состоянии:

- DB** (блокировка от дребезга)
- NV** (неактуальна)

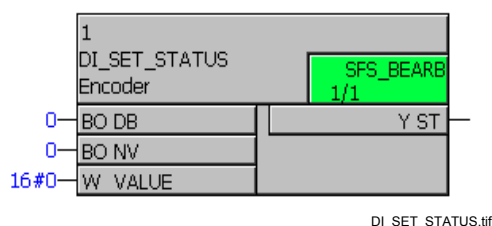


Рис. 4-18 Блок DI_SET_STATUS

Назначение входов / выходов

Блок **DI_SET_STATUS** имеет следующее назначение входов / выходов:

Таблица 4-27 Назначение входов / выходов блока DI_SET_STATUS

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	DB	BOOL	Включенная блокировка от дребезга	0
	NV	BOOL	Не обновлена	0
	VALUE	WORD	Двухпозиционное сообщение	16#0000
Выходы:	Y	ST	Двухпозиционное сообщение с состоянием	(0)



Примечание:

Вход **VALUE**: Передается только величина информации. Состояние этой информации при этом не учитывается.



Примечание:

Выход **Y**: Если выход соединен с правой границей схемы, т.е. данная информация обрабатывается дальше в устройстве, то для обработки в устройстве действительно следующее:

Состояние NV = TRUE

Действует как блокировка. Это значит, что если выход установлен на **неактуальна**, все другие изменения величины и состояния игнорируются.

Состояние NV

Изменение состояния **не обновлена**, которое появляется или пропадает, регистрируется только в журнале спонтанных сообщений, а не в журнале рабочих сообщений.

Состояние DB

Изменение состояния **блокировка от дребезга**, которое появляется или пропадает, используется только для проверки последующей обработки. Это не приводит к включению функции **блокировки от дребезга**.

Все изменения обрабатываются в одной схеме (например, направляются далее дешифратору **DI_SET_STATUS**).

4.4.4 MV_GET_STATUS

Функция

Блок **MV_GET_STATUS** расшифровывает величину (например, измеряемую величину). Выходы предоставляют два информационных элемента: величину и соответствующее состояние.

Измеряемая величина на входе **X** проверяется на наличие следующей информации о состоянии и выдается без изменения на выходе **VALUE**.

- NV** (неактуальна)
- OF** (переполнение)
- UG** (недействительна)

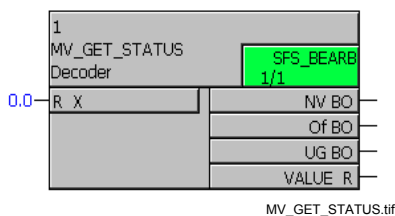


Рис. 4-19 Блок MV_GET_STATUS

Назначение входов / выходов

Блок **MV_GET_STATUS** имеет следующее назначение входов / выходов:

Таблица 4-28 Назначение входов / выходов блока MV_GET_STATUS

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	REAL	Измеряемая величина с состоянием	0,0
Выходы:	NV	BOOL	Не обновлена	0
	OF	BOOL	Переполнение	0
	UG	BOOL	Недостоверна	0
	VALUE	REAL	Измеряемая величина	0,0

4.4.5 MV_SET_STATUS

Функция

Блок **MV_SET_STATUS** устанавливает состояние в измеряемой величине.

Измеряемая величина **VALUE** заменяется соответствующей информацией о состоянии **NV**, **OF**, **UG** и выдается на выходе **Y**:

- NV** (неактуальна)
- OF** (переполнение)
- UG** (недостоверна)

Замена производится со следующим приоритетом: **UG**, **NV**, **OF**.

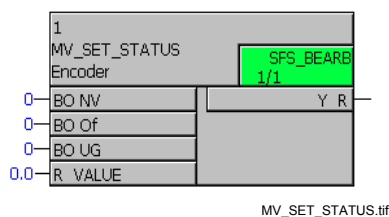


Рис. 4-20 Блок MV_SET_STATUS

Назначение входов / выходов

Блок **MV_SET_STATUS** имеет следующее назначение входов / выходов:

Таблица 4-29 Назначение входов / выходов блока MV_SET_STATUS

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	NV	BOOL	Не обновлена	0
	OF	BOOL	Переполнение	0
	UG	BOOL	Недостоверна	0
	VALUE	REAL	Измеряемая величина	0,0
Выходы:	Y	REAL	Измеряемая величина с состоянием	(0)



Примечание:

Вход **VALUE**: Передается только величина информации. Состояние этой информации при этом не учитывается.



Примечание:

Выход **Y**: Если выход соединен с правой границей схемы, т.е. данная информация обрабатывается дальше в устройстве, то для обработки в устройстве действительно следующее:

Состояние NV = TRUE

Действует как блокировка. Это значит, что если выход установлен на **неактуальна**, все другие изменения величины и состояния игнорируются.

Состояние NV

Изменение состояния **не обновлена**, которое появляется или пропадает, регистрируется только в журнале спонтанных сообщений, а не в журнале рабочих сообщений.

Состояние DB

Изменение состояния **блокировка от дребезга**, которое появляется или пропадает, используется только для проверки последующей обработки. Это не приводит к включению функции **блокировки от дребезга**.

Все изменения обрабатываются в одной схеме (например, направляются далее дешифратору **DI_SET_STATUS**).

4.4.6 SI_GET_STATUS

Функция

Блок **SI_GET_STATUS** расшифровывает однопозиционное сообщение (например, внешнее однопозиционное сообщение). Выходы предоставляют два информационных элемента: величину и соответствующее состояние.

Структура однопозиционного сообщения на входе **X** разбита на величину однопозиционного сообщения **VALUE (ВЕЛИЧИНА)** и следующую информацию о состоянии:

- DB** (блокировка от дребезга)
- NV** (неактуальна)

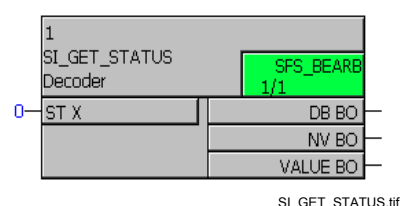


Рис. 4-21 Блок SI_GET_STATUS



Примечание:

Следующее должно быть учтено для того, чтобы соединить **быстрые сигналы** (например, нажимая функциональную кнопку на устройстве SIPROTEC) с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**):

Логические входы не являются запускающим событием для этих классов приоритетов и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки класса приоритетов.

Назначение входов / выходов

Блок **SI_GET_STATUS** имеет следующее назначение входов / выходов:

Таблица 4-30 Назначение входов / выходов блока SI_GET_STATUS

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	ST	Однопозиционное сообщение с состоянием	(0)

Таблица 4-30 Назначение входов / выходов блока SI_GET_STATUS

Выходы:	DB	BOOL	Включенная блокировка от дребезга	0
	NV	BOOL	Не обновлена	0
	VALUE	BOOL	Однопозиционное сообщение	16#0000

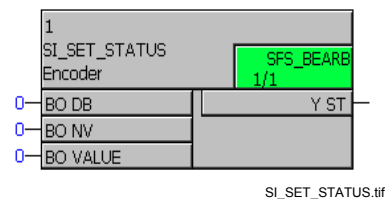
4.4.7 SI_SET_STATUS

Функция

Блок **SI_SET_STATUS** формирует однопозиционное сообщение с состоянием.

Структура однопозиционного сообщения на выходе **Y** формируется в зависимости от величины однопозиционного сообщения **VALUE (ВЕЛИЧИНА)** и следующей информации о состоянии:

- DB** (блокировка от дребезга)
- NV** (не обновлена)



SI_SET_STATUS.tif

Рис. 4-22 Блок SI_SET_STATUS

Назначение входов / выходов

Блок **SI_SET_STATUS** имеет следующее назначение входов / выходов:

Таблица 4-31 Назначение входов / выходов блока SI_SET_STATUS

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	DB	BOOL	Включенная блокировка от дребезга	0
	NV	BOOL	Не обновлена	0
	VALUE	BOOL	Однопозиционное сообщение	0
Выходы:	Y	ST	Однопозиционное сообщение с состоянием	(0)



Примечание:

Следующее должно быть учтено для входа **VALUE**:

Передается только величина информации. Состояние этой информации при этом не учитывается.



Примечание:

Следующее должно быть учтено для выхода Y:

Если выход соединен с правой границей схемы, т.е. данная информация обрабатывается дальше в устройстве, то для обработки в устройстве характерны следующие особенности:

Состояние NV = TRUE

Действует как блокировка. Это значит, что если выход установлен на **неактуальна**, все другие изменения величины и состояния игнорируются.

Состояние NV

Изменение состояния **не обновлена**, которое появляется или пропадает, регистрируется только в журнале спонтанных сообщений, а не в журнале рабочих сообщений.

Состояние DB

Изменение состояния **блокировка от дребезга**, которое появляется или пропадает, используется только для проверки последующей обработки. Это не приводит к включению функции **блокировки от дребезга**.

Все изменения обрабатываются в одной схеме (например, направляются далее дешифратору **DI_SET_STATUS**).

4.4.8 ST_AND

Функция

Блок **ST_AND** соединяет два однопозиционных сообщения с состоянием (бит NV) при помощи логической функции И и формирует из них выходной сигнал.

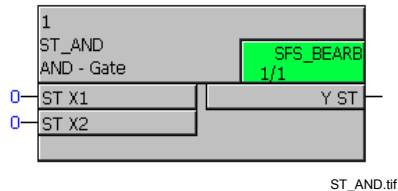


Рис. 4-23 Блок ST_AND



Примечание:

Следующее должно быть учтено для того, чтобы соединить **быстрые сигналы** (например, нажимая функциональную кнопку на устройстве SIPROTEC) с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**):

Логические входы не являются запускающим событием для этих классов приоритетов и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки класса приоритетов.

Назначение входов / выходов

Блок **ST_AND** имеет следующее назначение входов / выходов:

Таблица 4-32 Назначение входов / выходов блока ST_AND

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	ST	Вход с состоянием	(0)
	X2	ST	Вход с состоянием	(0)
Выходы:	Y	ST	Выход с состоянием	(0)

Сообщения и состояние

Если все сигналы соединены со входами, выход устанавливается на **INCOMING (ПРИХОДЯЩЕЕ)**. Состояние выходящего сообщения является результатом связки ИЛИ величин состояния всех сообщений.



Примечание:

Если, как минимум, одно из сообщений **OUTGOING (УХОДЯЩЕЕ)** и одновременно имеет состояние **Current (Актуальное) (= 0)**, то и выходящее сообщение получает состояние **Current (Актуальное) (= 0)**.

Величины состояния

Блок **ST_AND** учитывает только следующие величины состояния:

- NV** (не обновлена)
- DB** (блокировка от дребезга включена)

4.4.9 ST_NOT

Функция

Блок **ST_NOT** инвертирует однопозиционное сообщение с состоянием и формирует из него выходной сигнал с состоянием.

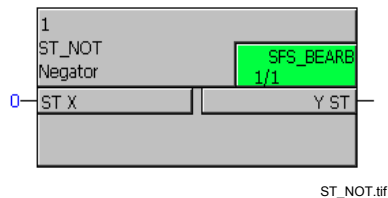


Рис. 4-24 Блок ST_NOT



Примечание:

Следующее должно быть учтено для того, чтобы соединить **быстрые сигналы** (например, нажимая на функциональную кнопку на устройстве SIPROTEC) с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**):

Логические входы не являются запускающим событием для этих классов приоритетов и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки класса приоритетов.

Назначение входов / выходов

Блок **ST_NOT** имеет следующее назначение входов / выходов:

Таблица 4-33 Назначение входов / выходов блока ST_NOT

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	ST	Вход с состоянием	(0)
Выходы:	Y	ST	Выход с состоянием	(0)

Сообщения и состояние

Блок **ST_NOT** отрицает величину сообщения на входе и передает ее далее на выход блока. Информация о состоянии при этом сохраняется.

Величины состояния

Блок **ST_NOT** учитывает только следующие величины состояния:

- NV** (не обновлена)
- DB** (блокировка от дребезга включена)

4.4.10 ST_OR

Функция

Блок **ST_OR** соединяет два однопозиционных сообщения с состоянием (бит NV) при помощи логической функции ИЛИ и формирует из них выходной сигнал.

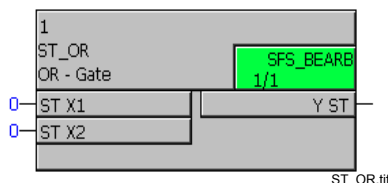


Рис. 4-25 Блок ST_OR



Примечание:

Следующее должно быть учтено для того, чтобы соединить **быстрые сигналы** (например, нажимая на функциональную кнопку на устройстве SIPROTEC) с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**): **Логические входы** не являются запускающим событием для этих классов приоритетов и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки класса приоритетов.

Назначение входов / выходов

Блок **ST_OR** имеет следующее назначение входов / выходов:

Таблица 4-34 Назначение входов / выходов блока ST_OR

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X1	ST	Вход с состоянием	(0)
	X2	ST	Вход с состоянием	(0)
Выходы:	Y	ST	Выход с состоянием	(0)

Сообщения и состояние

Как только первый сигнал соединяется с блоком, выход устанавливается на **INCOMING (ПРИХОДЯЩЕЕ)**. Если последний, соединенный с блоком сигнал **OUTGOING**, то выход также будет установлен на **OUTGOING (УХОДЯЩЕЕ)**.

Состояние выходящего сообщения является результатом связки ИЛИ величин состояния всех сообщений.



Примечание:

Если, как минимум, одно из сообщений **OUTGOING (УХОДЯЩЕЕ)** и одновременно имеет состояние **Current (Актуальное) (= 0)**, то и выходящее сообщение получает состояние **Current (Актуальное) (= 0)**.

**Величины
состояния**

Блок **ST_AND** учитывает только следующие величины состояния:

- NV** (не обновлена)
- DB** (блокировка от дребезга включена)

4.5 Memory (Память)

При помощи блоков памяти вы можете сохранять логические состояния устройства не только во время обработки схемы.

Существуют следующие виды блоков:

- **D_FF** (D триггер)
- **D_FF_MEMO** (D триггер с запоминаемым состоянием)
- **RS_FF** (RS триггер)
- **RS_FF_MEMO** (RS триггер с запоминаемым состоянием)
- **SR_FF** (SR триггер)
- **SR_FF_MEMO** (SR триггер с запоминаемым состоянием)
- **MEMORY** (память данных)

4.5.1 D_FF

Функция

При помощи блока **D Flipflop (D триггер)** сигнал на входе **D** передается на выход **Q** по нарастающему фронту импульса на входе **Clk (Синхровход)**. Величина сохраняется на выходе **Q** до тех пор, пока на входе **Clk** не появится следующий нарастающий фронт импульса.



Примечание:

Сигнал на выходе **Q** может быть установлен при первом запуске как стандартный и сохранен при повторных запусках, а затем восстановлен. Для этого вы должны использовать блок **D_FF_MEMO**.

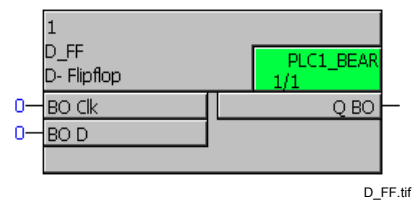


Рис. 4-26 Блок D_FF



Примечание:

Блок **D Flipflop** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**), медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **D_FF** имеет следующее назначение входов / выходов:

Таблица 4-35 Назначение входов / выходов блока D_FF

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Clk	BOOL	Тактовый импульс	0
	D	BOOL	Данные	0
Выход:	Q	BOOL	Выход	0

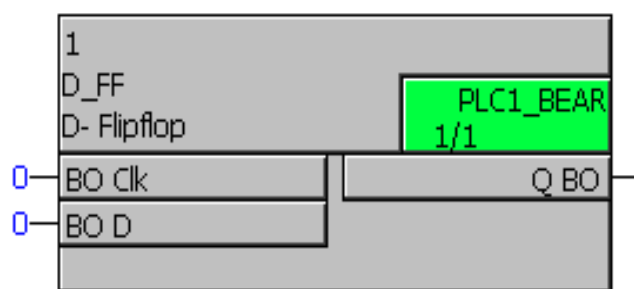
Таблица 4-36 Таблица истинности блока D_FF

Clk вход	D вход	Q выход
0 или 1	0	Q _{n-1}
Изменение с 0 на 1	0	0
0 или 1	1	Q _{n-1}
Изменение с 0 на 1	1	1

Пример использования

Используя показанную ниже схему CFC, вы можете, например, попеременно включать и отключать сирену при помощи одной функциональной кнопки:

- Вставьте в матрице ранжирования внутреннее однопозиционное сообщение **horn on/off (сирена вкл. / выкл.)**, которое вы ранжируете на функциональную кнопку (источник) и на CFC (цель).
- Новое выходное сообщение под названием **horn (сирена)**, сформированное в CFC (источник), соединяется с сиреной при помощи бинарного выхода.



D_FF_Bispiel.tif

Рис. 4-27 Пример использования блока D_FF, фрагмент схемы CFC

4.5.2 D_FF_MEMO

Функция

Блок **D Flipflop (D триггер) с памятью состояния** работает идентично блоку **D_FF**:

Сигнал на входе **D** передается на выход **Q** по нарастающему фронту импульса на входе **Ck** (**Синхровход**). Величина сохраняется на выходе **Q** до тех пор, пока на входе **Ck** не появится следующий нарастающий фронт импульса.

Кроме того, сигнал на выходе **Q** устанавливается по умолчанию при первом запуске на величину входа **INIT**, сохраняется перед каждым повторным запуском, а затем восстанавливается.

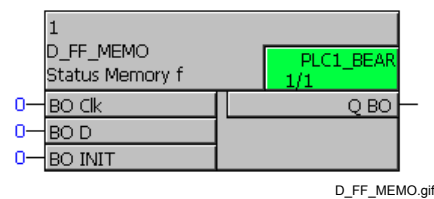


Рис. 4-28 Блок D_FF_MEMO



Примечание:

Максимально допустимое количество блоков **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** и **COUNTER** зависит от имеющейся энергонезависимой памяти и контролируется компилятором CFC. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.

Максимально допустимое количество проверяется во время компиляции схемы CFC. При наличии ошибки, система указывает на противоречия. Если расходуется большее количество ресурсов, чем предусмотрено, то это указывается в отображаемом на экране протоколе компиляции.



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

**Назначение
входов / выходов**

Блок **D_FF_MEMO** имеет следующее назначение входов / выходов:

Таблица 4-37 Назначение входов / выходов блока D_FF_MEMO

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Clk	BOOL	Тактовый импульс	0
	D	BOOL	Данные	0
	INIT	BOOL	Устанавливаемые по умолчанию данные Примечание: Значения по умолчанию относятся только к первому запуску.	0
Выход:	Q	BOOL	Выход	Первый запуск: Q = INIT Перезапуск: Q = Q при отключении устройства

Таблица 4-38 Таблица истинности блока D_FF_MEMO

Clk вход	D вход	Q выход
0 или 1	0	Q _{n-1}
Изменение с 0 на 1	0	0
0 или 1	1	Q _{n-1}
Изменение с 0 на 1	1	1

4.5.3 RS_FF

Функция

У блока **RS Flipflop (RS триггер)** значение 1 на входе **S** устанавливает выход **Q** на 1. Эта величина остается равной 1 до тех пор, пока на входе **R** не появится 1.

В этом блоке доминирует вход **R**, т.е. выход **Q** сбрасывается также в том случае, если на входах **S** и **R** одновременно имеется значение 1.

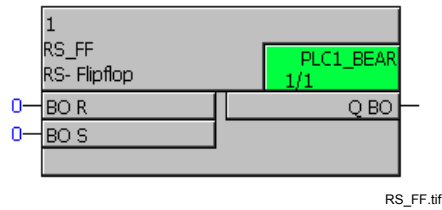


Рис. 4-29 Блок RS_FF



Примечание:

Блок **RS Flipflop** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**), медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **RS_FF** имеет следующее назначение входов / выходов:

Таблица 4-39 Назначение входов / выходов блока RS_FF

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	R	BOOL	Сброс	0
	S	BOOL	Установка	0
Выходы:	Q	BOOL	Выход	0

Таблица 4-40 Таблица истинности блока RS_FF

R ВХОД	S ВХОД	Q_n ВЫХОД
0	0	Q _{n-1}
0	1	1
1	0	0
1	1	0

4.5.4 RS_FF_MEMO

Функция

Блок **RS Flipflop (RS триггер) с памятью состояния** имеет аналогичный принцип работы как и блок **RS_FF**:

Единица на входе **S** устанавливает выход **Q** на 1. Эта величина остается равной 1 до тех пор, пока на входе **R** не появится 1.

В этом блоке доминирует вход **R**, т.е. выход **Q** сбрасывается также в том случае, если на входах **S** и **R** одновременно имеется значение 1.

Кроме того, сигнал на выходе **Q** устанавливается по умолчанию при первом запуске на величину входа **INIT**, сохраняется перед каждым повторным запуском, а затем восстанавливается.

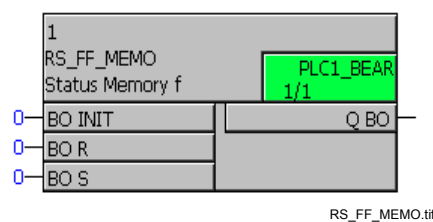


Рис. 4-30 Блок RS_FF_MEMO



Примечание:

Максимально допустимое количество блоков **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** и **COUNTER** зависит от имеющейся энергонезависимой памяти и контролируется компилятором CFC. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.

Максимально допустимое количество проверяется во время компиляции схемы CFC. При наличии ошибки, система указывает на противоречия. Если расходуется большее количество ресурсов, чем предусмотрено, то это указывается в отображаемом на экране протоколе компиляции.



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатываемого класса приоритетов.

**Назначение
входов / выходов**

Блок **RS_FF_MEMO** имеет следующее назначение входов / выходов:

Таблица 4-41 Назначение входов / выходов блока RS_FF_MEMO

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	INIT	BOOL	Устанавливаемые по умолчанию данные Примечание: Значения по умолчанию относятся только к первому запуску.	0
	R	BOOL	Сброс	0
	S	BOOL	Установка	0
Выход:	Q	BOOL	Выход	Первый запуск: Q = INIT Перезапуск: Q = Q при отключении устройства

Таблица 4-42 Таблица истинности блока RS_FF_MEMO

R ВХОД	S ВХОД	Q _n ВЫХОД
0	0	Q _{n-1}
0	1	1
1	0	0
1	1	0

4.5.5 SR_FF

Функция

У блока **SR Flipflop (SR триггер)** единица на входе **S** устанавливает выход **Q** на 1. Эта величина остается равной 1 до тех пор, пока на входе **R** не появится 1.

В этом блоке доминирует вход **S**, т.е. выход **Q = 1** также в том случае, если входы **S** и **R** равны 1.

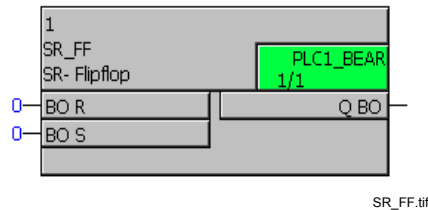


Рис. 4-31 Блок SR_FF



Примечание:

Блок **SR Flipflop** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**), медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **SR_FF** имеет следующее назначение входов / выходов:

Таблица 4-43 Назначение входов / выходов блока SR_FF

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	S	BOOL	Установка	0
	R	BOOL	Сброс	0
Выходы:	Q	BOOL	Выход	0

Таблица 4-44 Таблица истинности блока SR_FF

R ВХОД	S ВХОД	Q_n ВЫХОД
0	0	Q _{n-1}
0	1	1
1	0	0
1	1	1

4.5.6 SR_FF_MEMO

Функция

Блок **SR Flipflop (SR триггер) с памятью состояния** работает аналогично блоку **SR_FF**:

Единица на входе **S** устанавливает выход **Q** на 1. Эта величина остается равной 1 до тех пор, пока на входе **R** не появится 1.

В этом блоке доминирует вход **S**, т.е. выход **Q** = 1 также в том случае, если входы **S** и **R** равны 1.

Кроме того, сигнал на выходе **Q** устанавливается по умолчанию при первом запуске на величину входа **INIT**, сохраняется перед каждым повторным запуском, а затем восстанавливается.

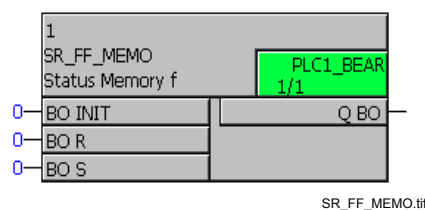


Рис. 4-32 Блок SR_FF_MEMO



Примечание:

Максимально допустимое количество блоков **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** и **COUNTER** зависит от имеющейся энергонезависимой памяти и контролируется компилятором CFC. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.

Максимально допустимое количество проверяется во время компиляции схемы CFC. При наличии ошибки, система указывает на противоречия. Если расходуется большее количество ресурсов, чем предусмотрено, то это указывается в отображаемом на экране протоколе компиляции.



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатываемого класса приоритетов.

Назначение входов / выходов

Блок **SR_FF_MEMO** имеет следующее назначение входов / выходов:

Таблица 4-45 Назначение входов / выходов блока RS_FF_MEMO

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	INIT	BOOL	Устанавливаемые по умолчанию данные Примечание: Значения по умолчанию относятся только к первому запуску.	0
	S	BOOL	Установка	0
	R	BOOL	Сброс	0
Выход:	Q	BOOL	Выход	Первый запуск: Q = INIT Перезапуск: Q = Q при отключении устройства

Таблица 4-46 Таблица истинности блока SR_FF_MEMO

R ВХОД	S ВХОД	Q _n ВЫХОД
0	0	Q _{n-1}
0	1	1
1	0	0
1	1	1

4.5.7 Memory (Память)

Функция Сигнал на входе **D** блока **MEMORY** передается на выход **Q** по нарастающему фронту импульса на входе **Clk (Синхровход)**. Величина сохраняется на выходе **Q** до тех пор, пока на входе **Clk** не появится следующий нарастающий фронт импульса.

Кроме того, сигнал на выходе **Q** устанавливается по умолчанию при первом запуске на величину входа **INIT**, сохраняется перед каждым повторным запуском, а затем восстанавливается.



Примечание:

Максимально допустимое количество блоков **MEMORY**, **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** и **COUNTER** зависит от имеющейся энергонезависимой памяти и контролируется компилятором CFC. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.

Максимально допустимое количество проверяется во время компиляции схемы CFC. При наличии ошибки, система указывает на противоречия. Если расходуется большее количество ресурсов, чем предусмотрено, то это указывается в отображаемом на экране протоколе компиляции.



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

**Назначение
входов / выходов**

Блок **MEMORY** имеет следующее назначение входов / выходов:

Таблица 4-47 Назначение входов / выходов блока MEMORY

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Clk	BOOL	Тактовый импульс	0
	D	REAL	Данные	0,0
	INIT	REAL	Устанавливаемые по умолчанию данные Примечание: Значения по умолчанию относятся только к первому запуску.	0,0
Выходы:	Q	REAL	Выход	Первый запуск: Q = INIT Перезапуск: Q = Q при отключении устройства

4.6 Command Module (Команды управления)

При помощи блоков команд управления вы можете инициировать или отменять команды управления или получать информацию о коммутационных командах.

Существуют следующие виды блоков команд управления:

- **BOOL_TO_CO** (преобразование булевой логики в команду)
- **BOOL_TO_IC** (преобразование булевой логики во внутреннее однопозиционное сообщение)
- **CMD_CANCEL** (отмена команды)
- **CMD_CHAIN** (последовательность переключений)
- **CMD_INF** (информация о команде)
- **LOOP** (обратная связь сигнала)

4.6.1 BOOL_TO_CO

Функция

Блок **Boolean to Command (преобразование булевой логики в команду)** формирует команду на переключение.

Команда на переключение определяется при помощи параметров **ORIGIN, PROP, VAL** и **TIME** и запускается сигналом на входе **TRIG**.

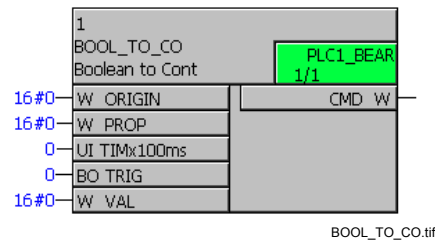


Рис. 4-33 Блок BOOL_TO_CO

**Примечание:**

Блок **Boolean to Command** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**) и медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и при прямом соединении с правой границей схемы.

Назначение входов / выходов

Блок **BOOL_TO_CO** имеет следующее назначение входов / выходов:

Таблица 4-48 Назначение входов / выходов блока BOOL_TO_CO

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	PROP	WORD	Свойства команды	16#0000
	TRIG	BOOL	Переключение	0
	ORIGIN	WORD	Источник команды	16#0000
	VAL	WORD	Направление переключения	16#0000
	TIMx100ms	UINT	Время вывода (разрешение 100 мсек; действительный диапазон величин от 0 до 32767)	0
Выходы:	CMD	WORD	Команда	16#0000

Параметр PROP Следующие значения могут быть установлены в параметре **PROP** (свойства команды):

Таблица 4-49 Значения для блока BOOL_TO_CO, параметр PROP

Значение (шестнадцатеричное)	Значение (десятичное)	Свойства команды
00	0	Без разблокировки
01	1	Положение разблокировки = текущее
02	2	Снятие блокировки подстанции
04	4	Снятие блокировки присоединения
08	8	(без действия)
10	16	(без действия)
20	32	Снятие блокировки от двойной команды
40	64	Снятие блокировки устройства защиты
80	128	Разблокировка при установленной блокировке подстанции или присоединения

Если параметр **PROP** установлен на 00 (шестнадцатеричное), применяются исходные свойства команды, заранжированные в матрице конфигурации DIGSI.



Примечание:

Если вы желаете одновременно задать несколько свойств команды, вы должны отдельно сложить шестнадцатеричные значения.

Параметр ORIGIN

Параметр **ORIGIN** используется для назначения определенного источника команде переключения, которую вы сформировали. Если вы назначаете, например, источнику **ORIGIN** значение 01, то при обработке эта команда будет считаться командой местного управления (в самом устройстве).

Следующие значения могут быть установлены в параметре **ORIGIN** (источник команды):

Таблица 4-50 Значения для блока BOOL_TO_CO, параметр ORIGIN

Значение	Пояснение
00	Автоматически сформированная команда
01	Встроенное управление, панель управления устройства (например, в самом устройстве)
02	DIGSI, SICAM, местное управление (например, DIGSI Remote, DIGSI на шине)
03	Центры телеуправления, дистанционное управление (например, WinCC, станция телеуправления)

Параметр VAL

Следующие значения могут быть установлены в параметре **VAL** (направление переключения):

Таблица 4-51 Значения для блока BOOL_TO_CO, параметр VAL

Значение	Пояснение
01	Выкл. (01)
02	Вкл. (10)

Параметр TIMx100ms

В параметре **TIMx100ms** (время вывода) вы устанавливаете время вывода сигнала с шагом 100 мсек.

Если параметр **TIMx100ms** установлен на 0, применяется исходное время, спараметрированное в матрице ранжирования DIGSI.

Параметр **TIMx100ms** применяется во время запуска устройства SIPROTEC. Данный параметр **не может** быть изменен во время работы устройства.

Устанавливаемое время не должно превышать 3 276,8 секунд. Это максимальный предел регулирования.

Выход IE

Выход **IE** должен быть напрямую соединен с командой на правой границе.

На выходе **IE** блок ведет себя следующим образом:
Если сигнал на входе **TRIG** меняется с 0 на 1, то команда, сформированная параметрами **PROP**, **ORIGIN**, **VAL** и **TIME** передается на выход **CMD**.

Выход CMD

Выход **CMD** должен быть напрямую соединен с командой на правой границе.

На выходе **CMD** блок ведет себя следующим образом:
 Если сигнал на входе **TRIG** меняется с 0 на 1, то команда, сформированная параметрами **PROP**, **ORIGIN**, **VAL** и **TIME** передается на выход **CMD**.

Пример использования

Используя представленную ниже схему CFC, вы можете, например, управлять силовым выключателем при помощи двух функциональных кнопок:

- F1** включает силовой выключатель,
- F2** отключает силовой выключатель.

- Вставьте в матрицу ранжирования необходимую информацию, как показано в следующем снимке экрана:

	Информация				Источник			Цель																			
	Номер	Текст дисплея	НС	Тип	Вх	Ф	С	Л	Вы	СД																	
									1	2	3	4	5	6	7	8	9	10	11	12	13	14	П	С	Л	Д	МУ
Device, General							*	*								*							*	*			
LED Sequence	F1 LED			IntSP					1																		
	LED 1			IntSP						×																	
	LED 2			IntSP							×																

Рис. 4-34 Пример использования блока BOOL_TO_CO, фрагмент матрицы ранжирования

- Создайте следующую схему CFC в классе приоритетов **PLC1_BEARB** (медленная логика Slow PLC):

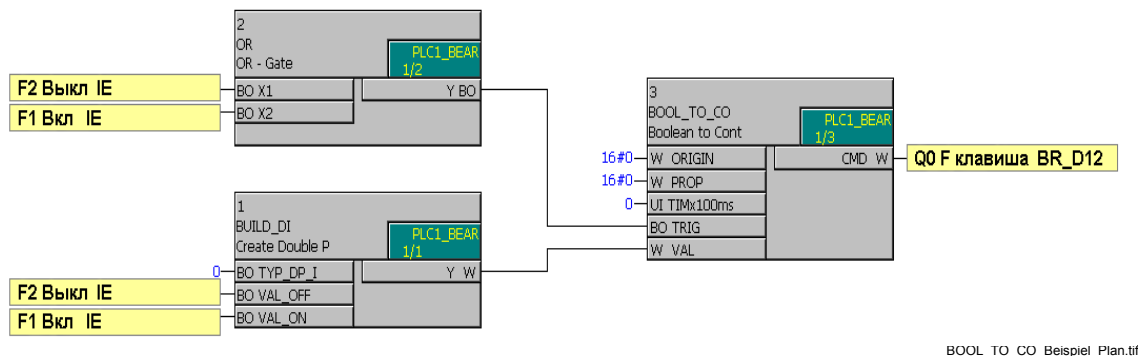


Рис. 4-35 Пример использования блока BOOL_TO_CO, фрагмент схемы CFC

4.6.2 BOOL_TO_IC

**Примечание:**

Структура и функция блоков **BOOL_TO_IC** и **BOOL_TO_IE** идентичны. В устройствах SIPROTEC 4 с более ранней версией программного обеспечения, блок также может еще называться **BOOL_TO_IE**.

Функция

Блок **Boolean to Internal IC (преобразование булевой логики во внутреннее однопозиционное сообщение)** формирует команду как внутреннее однопозиционное сообщение.

Внутреннее однопозиционное сообщение определяется как команда через параметры **ORIGIN** и **VAL** и инициируется при помощи сигнала на входе **TRIG**. Входы **PROP** и **TIM** не имеют значения.

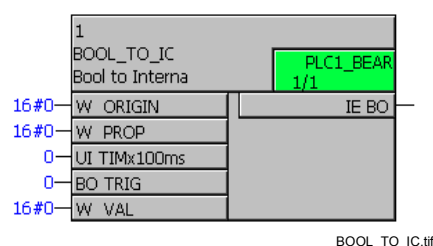


Рис. 4-36 Блок BOOL_TO_IC

**Примечание:**

Блок **Boolean to Internal IC (преобразование булевой логики во внутреннее однопозиционное сообщение)** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**), медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).

Назначение входов / выходов Блок **BOOL_TO_IC** имеет следующее назначение входов / выходов:

Таблица 4-52 Назначение входов / выходов блока BOOL_TO_IC

	Название	Тип данных	Пояснение	Значение по умолчанию
Входы:	PROP	WORD	Без функции	16#0000
	TRIG	BOOL	Переключение	0
	ORIGIN	WORD	Источник	16#0000
	VAL	WORD	Направление переключения	16#0000
	TIMx100ms	UINT	Без функции	0
Выход:	IE	BOOL	Команда (внутреннее однопозиционное сообщение)	0

Параметр ORIGIN

Параметр **ORIGIN** используется для назначения определенного источника команде переключения, которую вы сформировали. Если вы назначаете, например, источнику **ORIGIN** значение 01, то при обработке эта команда будет считаться командой местного управления (в самом устройстве).

Параметр **ORIGIN** (источник команды) может принимать следующие значения:

Таблица 4-53 Значения для блока BOOL_TO_CO, параметр ORIGIN

Значение	Пояснение
00	Автоматически сформированная команда
01	Встроенное управление, панель управления устройства (например, в самом устройстве)
02	DIGSI, SICAM, местное управление (например, DIGSI Remote, DIGSI на шине)
03	Центры телеуправления, дистанционное управление (например, WinCC, станция телеуправления)

Параметр VAL

Параметр **VAL** (направление переключения) может принимать следующие значения:

Таблица 4-54 Значения для блока BOOL_TO_IC, параметр VAL

Значение	Пояснение
01	Выкл. (01)
02	Вкл. (10)

Выход IE

На выходе **IE** блок ведет себя следующим образом: Если сигнал на входе **TRIG** меняется с 0 на 1, то команда, сформированная параметрами **PROP**, **ORIGIN**, **VAL** и **TIME**, передается на выход **CMD** как внутреннее однопозиционное сообщение.

4.6.3 BOOL_TO_IE



Примечание:

Структура и функция блоков **BOOL_TO_IE** и **BOOL_TO_IC** идентичны. В устройствах SIPROTEC 4 с более новой версией ПО этот блок называется **BOOL_TO_IC** (описание блока может быть найдено см. Раздел 4.6.2).

4.6.4 CMD_CANCEL

Функция

Блок **Cancel Command (Отмена команды)** позволяет вам отменить текущую команду.

Если в качестве адреса объекта **DEVICE (Устройство)** назначено значение -1 (уставка по умолчанию), все команды отменяются. Отмена инициируется нарастающим фронтом импульса на входе **TRIG**, не принимая во внимание второстепенные условия (режим управления, условия блокировки и т.д.).

Указание источника **ORIGIN** не влияет на функцию блока. Источник является только дополнительной информацией для журнала сообщений.

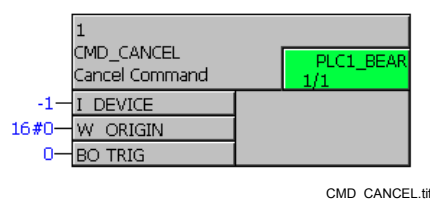


Рис. 4-37 Блок CMD_CANCEL



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение

входов / выходов

Блок **CMD_CANCEL** имеет следующее назначение входов / выходов:

Таблица 4-55 Назначение входов / выходов блока CMD_CANCEL

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	DEVICE	INT	Отменяемая команда Вход соединен с границей. Там отображается название команды.	-1
	ORIGIN	WORD	Источник команды	16#0000
	TRIG	BOOL	Стартовый вход	0

Параметр ORIGIN

При помощи параметра **ORIGIN** вы можете назначить определенный источник отмене команды. Если вы назначаете, например, источнику **ORIGIN** значение 01, то при обработке эта команда будет считаться командой местного управления.

Параметр **ORIGIN** (источник) может принимать следующие значения:

Таблица 4-56 Значения для блока CMD_CANCEL, параметр ORIGIN

Значение	Пояснение
00	Автоматически сформированная команда
01	Встроенное управление, панель управления устройства (например, в самом устройстве)
02	DIGSI, SICAM, местное управление (например, DIGSI Remote, DIGSI на шине)
03	Центры телеуправления, дистанционное управление (например, WinCC, станция телеуправления)

4.6.5 CMD_CHAIN

Функция

Блок **Switching Sequence (Последовательность переключений)** формирует последовательность переключений, состоящую из нескольких отдельных команд.

Основными функциями данного блока являются:

- вывод команды,
- ожидание ответного сигнала успешного выполнения команды,
- вывод сигнала успешного выполнения команды для обработки следующих команд.

Последовательность переключений параметрируется как цепочка команд. Несколько блоков **CMD_CHAIN** переключаются последовательно. Для формирования цепочки блоков используется механизм последовательно-приоритетной цепочки. Последовательность операций переключения в цепочке команд определяется положением блока в этой цепочке.

Блок **LOOP** может использоваться для обнуления всей последовательности переключений после успешного выполнения последней команды.

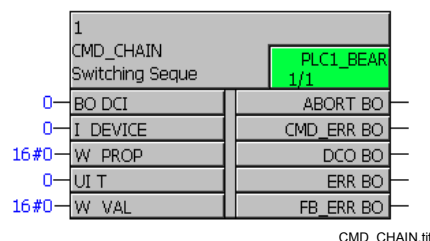


Рис. 4-38 CMD_CHAIN



Примечание:

Блок **CMD_CHAIN** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**) и медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**).



Примечание:

Если блок **CMD_CHAIN** используется в классе приоритетов медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**), то необходимо использовать блок **LOOP** для обнуления всей последовательности переключений после успешного выполнения последней команды.

**Примечание:**

Для устройств SIPROTEC с версией **ниже, чем V4.5** необходимо учесть следующее:

Максимально допустимое количество блоков **CMD_CHAIN** в классах приоритетов

- быстрая логика Fast PLC (класс приоритетов **PLC_BEARB**) и
- медленная логика Slow PLC (класс приоритетов **PLC1_BEARB**) составляет **20** штук.

Назначение входов / выходов Блок **CMD_CHAIN** имеет следующее назначение входов / выходов:

Таблица 4-57 Назначение входов / выходов блока CMD_CHAIN

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	DCI	BOOL	Вход последовательно-приоритетной цепочки блока В цепочке команд, этот вход должен быть соединен с выходом DCO предыдущего блока CMD_CHAIN .	0
	DEVICE	INT	Адресуемая команда Вход должен быть соединен с левой границей. Там указывается название команды.	0
	PROP	WORD	Свойства команды	16#0000
	T	UINT	Время вывода (разрешение 100 мсек; действительный диапазон величин 0 до 32767)	0
	VAL	WORD	Направление переключения	16#0000

Таблица 4-57 Назначение входов / выходов блока CMD_CHAIN

	Название	Тип данных	Примечание	Значение по умолчанию
Выходы:	ABORT	BOOL	Отмена Выход активен, если текущая команда была отменена (AB+).	0
	CMD_ERR	BOOL	Ошибка команды Выход активен, если команда не была выполнена успешно (BF-).	0
	DCO	BOOL	Выход последовательно-приоритетной цепочки блока В цепочке команд, этот выход должен быть соединен с входом DCI следующего блока CMD_CHAIN . Выход активен, если команда переключения была успешно обработана блоком. Выход DCO сразу становится неактивным, как только становится неактивным вход DCI .	0
	ERR	BOOL	Групповая ошибка Выход активен, если ABORT , CMD_ERR или FB_ERR активен, или, если обнаружена внутренняя ошибка.	0
	FB_ERR	BOOL	Ошибка подтверждения Выход активен, если нет подтверждения выполнения команды (RM-).	0

Параметр PROP Параметр **PROP** (свойства команды) может принимать следующие значения:

Таблица 4-58 Значения для блока CMD_CHAIN, параметр PROP

Значение (шестнадцатеричное)	Значение (десятичное)	Свойства команды
00	0	Без разблокировки
01	1	Положение разблокировки = текущее
02	2	Снятие блокировки подстанции
04	4	Снятие блокировки присоединения

Таблица 4-58 Значения для блока CMD_CHAIN, параметр PROP

Значение (шестнадцатеричное)	Значение (десятичное)	Свойства команды
08	8	Без контроля синхронизма (конфигурация SY игнорируется)
10	16	С контролем синхронизма (условия синхронизации проверяются в принудительном порядке)
20	32	Снятие блокировки от двойной команды
40	64	Снятие блокировки защиты
80	128	Разблокировка при установленной блокировке подстанции или присоединения

Если на вход **PROP** подается 00 (шестнадцатеричное), применяются исходные свойства команды, заранжированные в матрице конфигурации DIGSI.

**Примечание:**

Если вы желаете одновременно задать несколько свойств команды, вы должны отдельно сложить шестнадцатеричные значения.

Параметр VAL

Параметр **VAL** (направление переключения) может принимать следующие значения:

Таблица 4-59 Значения для блока CMD_CHAIN, параметр VAL

Значение	Пояснение
01	Выкл. (01)
02	Вкл. (10)

Параметры T

Вход **T** (время вывода) используется для определения времени команды с шагом 100 мсек.

Если параметр **T** установлен на 0, применяется исходное время, спараметрированное в матрице ранжирования DIGSI.

Параметр **T** применяется во время запуска устройства SIPROTEC. Данный параметр **не может** быть изменен во время работы.

Устанавливаемое время не должно превышать 3 276,8 секунд. Это максимальное время, которое можно установить.

**Примечание:**

Текущая последовательность переключений может быть остановлена сбросом сигнала на входе **DCI** первого блока в цепочке команд. Этот блок передает сигнал на все последующие блоки в цепочке через выход **DCO**. Активный блок в цепочке команд обозначен **DCI = 1** и **DCO = 0**.

Если во время выполнения текущей команды на переключение сигнал на входе **DCI** становится неактивным, формируется отмена обработки текущей команды и последовательность переключений прерывается. Затем ожидается результат отмены выполнения задачи:

- Выполнение команды может быть прервано
Операция переключения, которая может быть прервана в настоящее время, немедленно завершается. Последующие операции переключения в цепочке команд не выполняются.
- Выполнение команды не может быть прервано
Операция переключения, которая не может быть прервана в настоящее время, немедленно завершается. Последующие операции переключения в цепочке команд не выполняются.

Возможность или невозможность прерывания выполнения команды зависит от режима выполнения команды (Свойства объекта):

- Выполнение постоянной команды не может быть прервано.
- Выполнение импульсной команды может быть прервано.

4.6.6 CMD_INF

Функция Блок **Command Information (Информация о команде)** предоставляет информацию о запущенной команде на переключение.



Примечание:

Блок **Command Information** предназначен для использования в блокировке (класс приоритетов **SFS_BEARB**) для того, чтобы проверить состояние коммутационного аппарата. В других классах приоритетов данный блок работает только при определенных условиях.

Условия блокировки при этом не учитываются. Здесь только отображается, имеется ли команда на переключение или нет (независимо от выполнения команды на переключение).

Блок **Command Information** должен быть соединен с каждым устройством управления, от которого вы желаете получить информацию о запущенной команде на переключение.

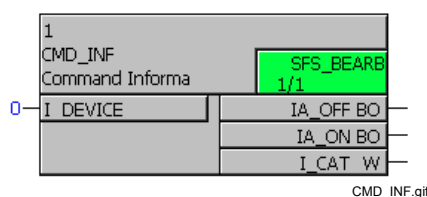


Рис. 4-40 Блок CMD_INF

Назначение входов / выходов Блок **CMD_INF** имеет следующее назначение входов / выходов:

Таблица 4-60 Назначение входов / выходов блока CMD_INF

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	DEVICE	INT	Вход соединен с левой границей. Там отображается название объекта команды.	0

Таблица 4-60 Назначение входов / выходов блока CMD_INF

	Название	Тип данных	Примечание	Значение по умолчанию
Выходы:	IA_ON	BOOL	Информация активна, если была выдана команда на включение.	0
	IA_OFF	BOOL	Информация активна, если была выдана команда на отключение.	0
	I_CAT	WORD	Информацию об инициаторе команды.	16#0000

Параметр I_CAT

Параметр **I_CAT** может принимать следующие значения:

Таблица 4-61 Значения для блока CMD_INF, параметр I_CAT

Значение	Пояснение
00	Автоматически сформированная команда
01	Встроенное управление, панель управления устройства (например, в самом устройстве)
02	DIGSI, SICAM, местное управление (например, DIGSI Remote, DIGSI на шине)
03	Центры телеуправления, дистанционное управление (например, WinCC, станция телеуправления)

4.6.7 LOOP

Функция

Блок **signal feedback (обратная связь сигнала)** позволяет осуществлять обратную связь.

Под обратной связью здесь подразумевается связь выхода блока с входом блока, имеющим более низкий номер последовательности выполнения в схеме CFC.

Схема CFC, содержащая сигнал обратной связи, снова обрабатывается. Новые входные сигналы, которые могут появиться тем временем на входах блоков, игнорируются. Во время повторной обработки рабочего листа, входные сигналы не изменяются, кроме сигнала обратной связи, который возвращен блоком **LOOP**.

Блок **LOOP** используется, например, для последовательностей переключений (см. блок **CMD_CHAIN**): После успешного завершения выполнения последней команды, по сигналу обратной связи обнуляется вся последовательность переключений.

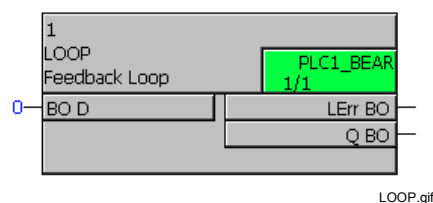


Рис. 4-41 Блок LOOP



Примечание:

Блок **Loop** работает только в классах приоритетов медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**), обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).



Примечание:

Если блок **CMD_CHAIN** используется в классе приоритетов медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**), блок **LOOP** должен использоваться для обнуления всей последовательности переключений после успешного выполнения последней команды.

**Назначение
входов / выходов**

Блок **LOOP** имеет следующее назначение входов / выходов:

Таблица 4-62 Назначение входов / выходов блока LOOP

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	D	BOOL	Сигнал на входе D передается на выход Q .	0
Выходы:	Q	BOOL	Сигнал на выходе Q передается на вход D .	0
	LErr	BOOL	Выход ошибки Выход активен, если сигнал был возвращен блоком LOOP более 5 раз.	0

**Примечание:**

Чтобы избежать бесконечной обратной связи, сигнал не может быть возвращен блоком **LOOP** более 5 раз (вход **D** = выход **Q**). Если это количество превышено, на выходе **LErr** формируется сигнал ошибки и обратная связь прерывается.

4.7 Type Convertor (Преобразователи типа)

При помощи блоков-преобразователей типа, вы можете преобразовывать тип данных информационных элементов.

Существуют следующие виды блоков-преобразователей типа:

- **BOOL_TO_DI** (преобразование булевой логики в двухпозиционное сообщение)
- **BUILD_DI** (формирование двухпозиционного сообщения)
- **DI_TO_BOOL** (преобразование двухпозиционного сообщения в булеву логику)
- **DINT_TO_REAL** (адаптер)
- **DIST_DECODE** (декодирование двухпозиционного сообщения) с состоянием
- **DM_DECODE** (декодирование двухпозиционного сообщения)
- **REAL_TO_DINT** (адаптер)
- **REAL_TO_UINT** (адаптер)
- **REAL_TO_INT** (адаптер)
- **INT_TO_REAL** (адаптер)
- **UINT_TO_REAL** (адаптер)

4.7.1 BOOL_TO_DI

**Примечание:**

Блок **BOOL_TO_DI** входит в исходный комплект поставки устройств SIPROTEC 4. Для создания новых схем CFC мы рекомендуем использовать более простой в обращении блок **BUILD_DI**.

Функция

Блок **Boolean to Double-Point Indication (Преобразование булевой логики в двухпозиционное сообщение)** формирует двухпозиционное сообщение.

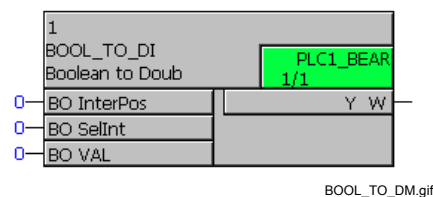


Рис. 4-42 Блок BOOL_TO_DI

**Внимание:**

Для соединения **быстрых сигналов** с левой границей схемы CFC в классе приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

**Примечание:**

Блок **Boolean to Double-Point Indication** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**), медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).

Назначение входов / выходов Блок **BOOL_TO_DI** имеет следующее назначение входов / выходов:

Таблица 4-63 Назначение входов / выходов блока BOOL_TO_DI

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	SelInt	BOOL	Переключатель между входом VAL и входом InterPos (переключает между конечным и промежуточным положением выключателя)	0
	VAL	BOOL	Значение (конечное положение выключателя)	0
	InterPos	BOOL	Промежуточное положение	0
Выходы:	Y	WORD	Созданное двухпозиционное сообщение	16#0000

Созданные двухпозиционные сообщения Сигналы на входах **VAL** и **InterPos** могут формировать на выходе **Y** следующие двухпозиционные сообщения, в зависимости от входа **SelInt** (переключает между конечным и промежуточным положением выключателя):

Используемая в таблице аббревиатура означает следующее:
DP двухпозиционное сообщение и
DP_I двухпозиционное сообщение с распознаванием промежуточного положения.

Таблица 4-64 Созданные двухпозиционные сообщения блока BOOL_TO_DI

SelInt вход	InterPos вход	VAL вход	Y выход	Значение для DP	Значение для DP_I
0	X	0	0001	ВЫКЛ	ВЫКЛ
0	X	1	0002	ВКЛ	ВКЛ
1	0	X	0000	не определено	INTERM (промежуточное положение 00)
1	1	X	0003	ПРОМЕЖУТОЧНОЕ	INTERM (промежуточное положение 11)

4.7.2 BUILD_DI

Функция

Блок **Create Double-Point Indication (Создание двухпозиционного сообщения)** формирует двухпозиционное сообщение из двух входных сигналов.

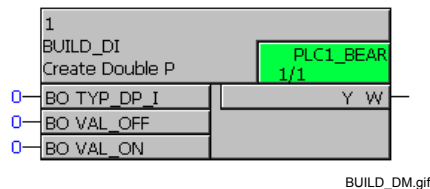


Рис. 4-43 Блок BUILD_DI



Примечание:

Блок **Create Double-Point Indication (Создание двухпозиционного сообщения)** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**), медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **BUILD_DI** имеет следующее назначение входов / выходов:

Таблица 4-65 Назначение входов / выходов блока BUILD_DI

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	VAL_OFF	BOOL	Обозначает состояние Выкл.	0
	VAL_ON	BOOL	Обозначает состояние Вкл.	0
	TYP_DP_I	BOOL	Тип сформированного двухпозиционного сообщения	0
Выходы:	Y	WORD	Созданное двухпозиционное сообщение	16#0000

Параметр TYP_DP_I

Параметр **TYP_DP_I** (тип сформированного двухпозиционного сообщения) может иметь следующие значения:

Таблица 4-66 Значения для блока BUILD_DI, параметр TYP_DP_I

Значение	Пояснение
0	DP (двухпозиционное сообщение)
1	DP_I (двухпозиционное сообщение с распознаванием промежуточного положения)

Созданные двухпозиционные сообщения

Сигналы на входах **VAL_OFF** и **VAL_ON** могут формировать на выходе **Y** следующие двухпозиционные сообщения, в зависимости от входа **TYP_DP_I**:

Таблица 4-67 Созданные двухпозиционные сообщения блока BUILD_DI

TYP_DP_I вход	VAL_ON вход	VAL_OFF вход	Y выход	Пояснение
0	0	0	0003	ПРОМЕЖУТОЧНОЕ
0	0	1	0001	ВЫКЛ
0	1	0	0002	ВКЛ

Таблица 4-67 Созданные двухпозиционные сообщения блока BUILD_DI

ТYP_DP_I ВХОД	VAL_ON ВХОД	VAL_OFF ВХОД	Y ВЫХОД	Пояснение
0	1	1	0003	ПРОМЕЖУТОЧНОЕ
1	0	0	0000	INTERM (промежуточное положение 00)
1	0	1	0001	ВЫКЛ
1	1	0	0002	ВКЛ
1	1	1	0003	INTERM (промежуточное положение 11)

4.7.3 DI_TO_BOOL

**Примечание:**

Блок **DI_TO_BOOL** входит в исходный комплект поставки устройств SIPROTEC 4. Для создания новых схем CFC мы рекомендуем использовать более простой в обращении блок **DM_DECODE**.

Функция

Блок **Double-Point Indication to Boolean (Преобразование двухпозиционного сообщения в булеву логику)** проверяет двухпозиционное сообщение на наличие одного из четырех возможных состояний и формирует сигнал как результат.

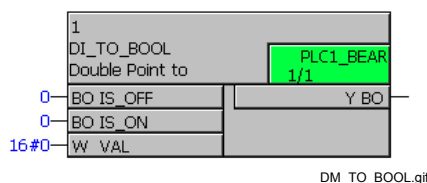


Рис. 4-44 Блок DI_TO_BOOL

**Примечание:**

Блок **Double-Point Indication to Boolean (Преобразование двухпозиционного сообщения в булеву логику)** работает только в классах приоритетов быстрой логики Fast PLC (класс приоритетов **PLC_BEARB**), медленной логики Slow PLC (класс приоритетов **PLC1_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**).

Назначение входов / выходов

Блок **DI_TO_BOOL** имеет следующее назначение входов / выходов:

Таблица 4-68 Назначение входов / выходов блока DI_TO_BOOL

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	IS_OFF	BOOL	Запрос на Выкл.	0
	IS_ON	BOOL	Запрос на Вкл.	0
	VAL	WORD	Двухпозиционное сообщение	16#0000
Выходы:	Y	BOOL	Результат	0

Параметр VAL

Параметр **VAL** (двухпозиционное сообщение) может принимать следующие значения:

Таблица 4-69 Значения для блока DI_TO_BOOL, параметр VAL

Значение	Значение для DP_I (двухпозиционное сообщение с распознаванием промежуточного положения)	Значение для DP (двухпозиционное сообщение)
0000	INTERM (промежуточное положение 00), не определено	не определено (соответствует 00)
0001	ВЫКЛ. (соответствует 01)	ВЫКЛ. (соответствует 01)
0002	ВКЛ. (соответствует 10)	ВКЛ. (соответствует 10)
0003	INTERM (промежуточное положение 11)	INTERM (промежуточное положение) (соответствует 00)

**Сформированный
выходной сигнал**

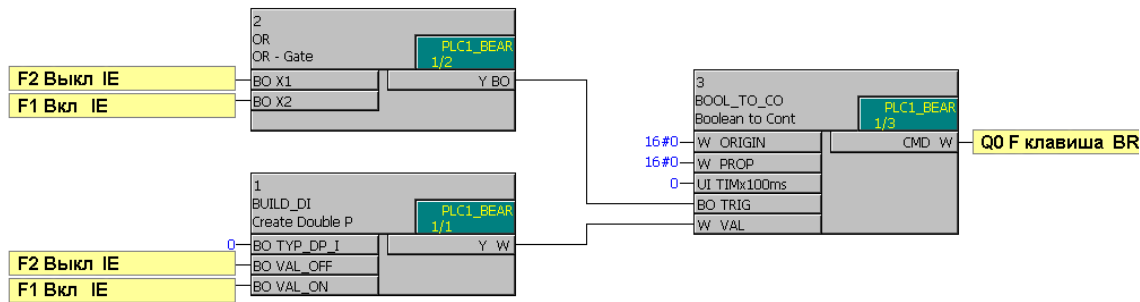
Сигналы на входах **IS_ON** и **IS_OFF** могут формировать на выходе **Y** следующий выходной сигнал, в зависимости от двухпозиционного сообщения **VAL**:

Таблица 4-70 Сформированный выходной сигнал блока DI_TO_BOOL

IS_ON вход	IS_OFF вход	VAL вход для DP	VAL вход для DP_I	Y выход
0	0	не определено	INTERM (промежуточное положение 00), не определено	1
		ВЫКЛ., ВКЛ., ПРОМЕЖУТ	ВЫКЛ., ВКЛ., INTERM (промежуточное положение 11)	0
0	1	ВЫКЛ.	ВЫКЛ.	1
		ВКЛ., ПРОМЕЖУТ не определено	ВКЛ., INTERM (промежуточное положение 00), INTERM (промежуточное положение 11) не определено	0
1	0	ВКЛ.	ВКЛ.	1
		ВЫКЛ., ПРОМЕЖУТ не определено	ВЫКЛ., INTERM (промежуточное положение 00), INTERM (промежуточное положение 11) не определено	0
1	1	ПРОМЕЖУТ, не определено	INTERM (промежуточное положение 11)	1
		ВЫКЛ., ВКЛ.	ВЫКЛ., ВКЛ., INTERM (промежуточное положение 00), не определено	0

Пример использования

Используя представленную ниже схему CFC, вы можете создать следующие условия блокировки в классе обработки **SFS_BEARB** (блокировка):
 Включение **силового выключателя**: Разъединитель = **ON (Вкл.)** и заземляющий нож = **OFF (Выкл.)**,
 Включение **заземляющего ножа**: Силовой выключатель = **OFF (Выкл.)** и заземляющий нож = **OFF (Выкл.)**,
 Включение **разъединителя**: Силовой выключатель = **OFF (Выкл.)** и заземляющий нож = **OFF (Выкл.)**.



DI_TO_BOOL_Beispiel_Plan.tif

Рис. 4-45 Пример использования блока DI_TO_BOOL, фрагмент схемы CFC

4.7.4 DINT_TO_REAL

Функция

Блок **DINT_TO_REAL** преобразовывает двойные целые числа типа **Double Integer** в действительные числа типа **REAL** и является, таким образом, противоположностью блоку **REAL_TO_DINT**.

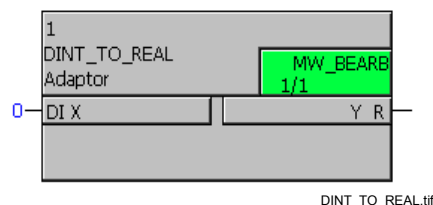


Рис. 4-46 Блок DINT_TO_REAL

Назначение**входов / выходов**

Блок **DINT_TO_REAL** имеет следующее назначение входов / выходов:

Таблица 4-71 Назначение входов / выходов блока DINT_TO_REAL

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	DINT	Double Integer входная величина	0
Выходы:	Y	REAL	Real выходная величина	0,0

**Примечание:**

Информация о состоянии, содержащаяся в типе данных REAL (См. Таблицу 4-2), сохраняется при преобразовании в тип данных DINT и выдается на выходе Y.

4.7.5 DIST_DECODE

Функция

Блок **Decode Double-Point Indication (Декодирование двухпозиционного сообщения)** преобразовывает двухпозиционное сообщение с состоянием в четыре однопозиционных сообщения с состоянием. В противоположность к блоку **DM_DECODE**, состояние передается на выходы в неизменном виде.

Возможными параметрами являются: IS_ON, IS_OFF, INTPOS0 и INTPOS3. Эти параметры доступны как выходы с состоянием.

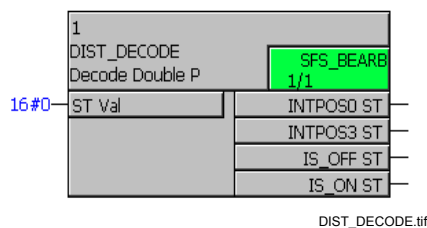


Рис. 4-47 Блок DIST_DECODE

Назначение входов / выходов

Блок **DIST_DECODE** имеет следующее назначение входов / выходов:

Таблица 4-72 Назначение входов / выходов блока DIST_DECODE

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	VAL	STRUCT	Значение двухпозиционного сообщения	(0)

Таблица 4-72 Назначение входов / выходов блока DIST_DECODE

	Название	Тип данных	Примечание	Значение по умолчанию
Выходы:	INTPOS0	SIST	промежуточное положение0 (декодированное двухпозиционное сообщение) с состоянием	0
	INTPOS3	SIST	ВЫКЛ. (декодированное двухпозиционное сообщение) с состоянием	0
	IS_OFF	SIST	ВКЛ. (декодированное двухпозиционное сообщение) с состоянием	0
	IS_ON	SIST	промежуточное положение3 (декодированное двухпозиционное сообщение) с состоянием	0

Параметр VAL

В зависимости от соединения, параметр **VAL** может иметь следующие значения:

Таблица 4-73 Значения для блока DIST_DECODE, параметр VAL

Входное значение ST Val	Выход активен для DP	Выход активен для DP_I
0000 (промежуточное 00)	INTPOS3	INTPOS0
0001 (Выкл.)	IS_OFF	IS_OFF
0002 (Вкл.)	IS_ON	IS_ON
0003 (промежуточное 11)	INTPOS3	INTPOS3

4.7.6 DM_DECODE

Функция

Блок **Decode Double-Point Indication (Декодирование двухпозиционного сообщения)** преобразовывает двухпозиционное сообщение в четыре логические величины. В противоположность блоку **DIST_DECODE**, состояние не передается.

Возможны следующие параметры ON (Вкл.), OFF (Выкл.), INTERM0 (промежуточное положение0) и INTERM3 (промежуточное положение3). Эти параметры доступны как выходы.

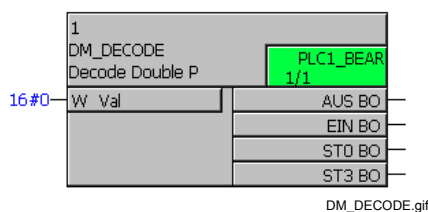


Рис. 4-48 Блок DM_DECODE

Назначение входов / выходов

Блок **DM_DECODE** имеет следующее назначение входов / выходов:

Таблица 4-74 Назначение входов / выходов блока DM_DECODE

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	VAL	WORD	Значение двухпозиционного сообщения	16#0000
Выходы:	ON	BOOL	ВКЛ. (декодированное двухпозиционное сообщение)	0
	OFF	BOOL	ВЫКЛ. (декодированное двухпозиционное сообщение)	0
	ST0	BOOL	Пром. полож.0 (декодированное двухпозиционное сообщение)	0
	ST3	BOOL	Пром. полож.3 (декодированное двухпозиционное сообщение)	0

Параметр VAL В зависимости от соединения, параметр **VAL** может иметь следующие значения:

Таблица 4-75 Значения для блока DM_DECODE, параметр VAL

Значение	Значение для DP (двухпозиционное сообщение)	Значение для DP_I (двухпозиционное сообщение с распознаванием промежуточного положения)	Выход блока DM_DECODE
0000	не определено	INTERM (промежуточное положение 00), не определено	ST0
0001	ВЫКЛ.	ВЫКЛ.	ВЫКЛ.
0002	ВКЛ.	ВКЛ.	ВКЛ.
0003	ПРОМЕЖУТОЧНОЕ	INTERM (промежуточное положение 11)	ST3



Примечание:

Для того чтобы также обработать состояние двухпозиционного сообщения, вместо блока **DM_DECODE**, вы можете использовать блок **DIST_DECODE**.

4.7.7 REAL_TO_DINT

Функция

Блок **REAL_TO_DINT** преобразовывает действительные числа типа **Real** в двойные целые типа **Double Integer**.

Число округляется: в случае с 0,5, число округляется до большего.

При превышении или занижении диапазона выходных величин, на выходе выдается максимальная / минимальная величина. Кроме того, устанавливается выход **ERR**.

Выход **ERR** активен до тех пор, пока входная величина не будет соответствовать диапазону величин выхода **Y**.

Если на входе **X** нет достоверной величины типа Real (действительная), то на выходе **Y** выдается 0 и устанавливается выход **ERR**.

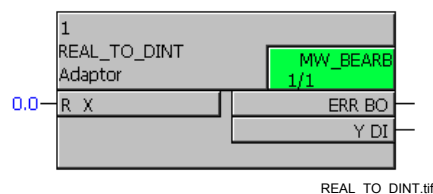


Рис. 4-49 Блок REAL_TO_DINT

Назначение входов / выходов

Блок **REAL_TO_DINT** имеет следующее назначение входов / выходов:

Таблица 4-76 Назначение входов / выходов блока REAL_TO_DINT

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	REAL	Real входная величина	0,0
Выходы:	Y	DINT	Double Integer выходная величина	0
	ERR	BOOL	Превышение / занижение диапазона выходных величин	0



Примечание:

Информация о состоянии, содержащаяся в типе данных REAL (См. Таблицу 4-2), сохраняется при преобразовании в тип данных DINT и выдается на выходе Y.

4.7.8 REAL_TO_INT

Функция

Блок **REAL_TO_INT** преобразовывает действительные числа типа **Real** в целые типа **Integer**. Он позволяет, например, соединить заданную (пороговую) величину с входом времени таймера. Таким образом, таймер может быть установлен во время работы устройства при помощи местного управления.

Число округляется: в случае с 0,5, число округляется до большего.

Если на входе X нет достоверной величины типа Real (действительная), то выход ERR равен 1 и выход Y ведет себя следующим образом:

Таблица 4-77 Поведение выходов у блока REAL_TO_INT

Состояние на входе X	Выход Y	Выход ERR
OVERFLOW	65535	1
OVERFLOW_NEG	0	1
LIVE_ZERO	Y (n-1) без изменения	1
NOT_DEFINED	Y (n-1) без изменения	1
NOT_CALCULATED	Y (n-1) без изменения	1
INVALID	Y (n-1) без изменения	1

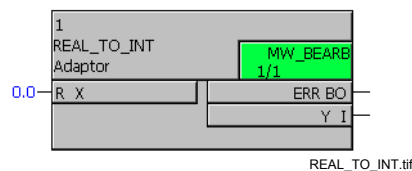


Рис. 4-50 Блок REAL_TO_INT

**Примечание:**

Блок **REAL_TO_INT** входит в исходный комплект поставки устройств SIPROTEC 4. Существующие схемы CFC, использующие данный блок, все еще поддерживаются. Для создания новых схем CFC мы рекомендуем прибегнуть к более универсальному в использовании блоку **REAL_TO_UINT**.

4.7.9 REAL_TO_UINT

Функция

Блок **REAL_TO_UINT** преобразовывает действительные числа типа **Real** в целые числа без знака типа **Unsigned Integer**. Он позволяет, например, соединить заданную (пороговую) величину с входом времени таймера. Таким образом, таймер может быть запущен во время работы устройства при помощи местного управления.

Число округляется: в случае с 0,5, число округляется до большего.

Если на входе **X** нет достоверной величины типа Real (действительная), то выход **ERR** равен 1 и выход **Y** ведет себя следующим образом:

Таблица 4-78 Поведение выходов у блока REAL_TO_UINT

Состояние на входе X	Выход Y	Выход ERR
OVERFLOW	65535	1
OVERFLOW_NEG	0	1
LIVE_ZERO	Y (n-1) без изменения	1
NOT_DEFINED	Y (n-1) без изменения	1
NOT_CALCULATED	Y (n-1) без изменения	1
INVALID	Y (n-1) без изменения	1

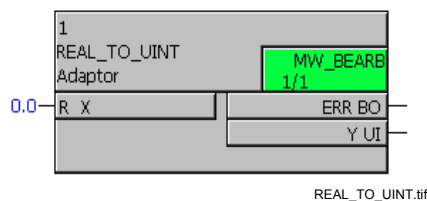


Рис. 4-51 Блок REAL_TO_UINT

**Назначение
входов / выходов**

Блок **REAL_TO_UINT** имеет следующее назначение входов / выходов:

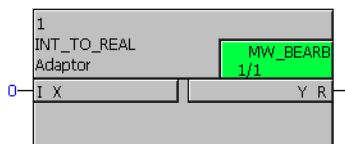
Таблица 4-79 Назначение входов / выходов блока REAL_TO_UINT

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	REAL	Real входная величина	0,0
Выходы:	Y	UINT	Unsigned Integer выходная величина	0
	ERR	BOOL	Превышение / занижение диапазона выходных величин	0

4.7.10 INT_TO_REAL

Функция

Блок **INT_TO_REAL** преобразовывает двойные целые числа типа **Double Integer** в действительные числа типа **Real** и является, таким образом, противоположностью блоку **REAL_TO_INT**.



INT_TO_REAL.tif

Рис. 4-52 Блок INT_TO_REAL

Назначение входов / выходов

Блок **INT_TO_REAL** имеет следующее назначение входов / выходов:

Таблица 4-80 Назначение входов / выходов блока INT_TO_REAL

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	INT	Unsigned Integer входная величина	0
Выходы:	Y	REAL	Real выходная величина	0,0

**Примечание:**

Блок **INT_TO_REAL** входит в исходный комплект поставки устройств SIPROTEC 4. Существующие схемы CFC, использующие данный блок, все еще поддерживаются. Для создания новых схем CFC мы рекомендуем прибегнуть к более универсальному в использовании блоку **UINT_TO_REAL**.

4.7.11 UINT_TO_REAL

Функция

Блок **UINT_TO_REAL** преобразовывает целые числа без знака типа **Unsigned Integer** в действительные числа типа **Real** и является, таким образом, противоположностью блоку **REAL_TO_UINT**.

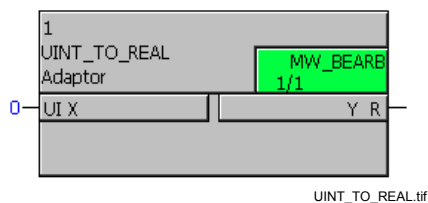


Рис. 4-53 Блок UINT_TO_REAL

Назначение**входов / выходов**

Блок **UINT_TO_REAL** имеет следующее назначение входов / выходов:

Таблица 4-81 Назначение входов / выходов блока UINT_TO_REAL

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	X	UINT	Unsigned Integer входная величина	0
Выходы:	Y	REAL	Real выходная величина	0,0

4.8 Comparison (Сравнение)

При помощи блоков сравнения вы можете сравнивать или корректировать измеряемые величины.

Существуют следующие виды блоков сравнения:

- **COMPARE** (сравнение измеряемых величин)
- **LIVE_ZERO** (компаратор)
- **LOWER_SETPOINT** (нижний предел)
- **LOWER_SETPOINT** (верхний предел)
- **ZERO_POINT** (подавление незначущих нулей)

4.8.1 COMPARE (СРАВНЕНИЕ)

Функция

Блок **Measured value comparison (Сравнение измеряемых величин)** позволяет сравнивать две величины типа Real **VAL1** и **VAL2** по **больше**, **меньше** или **равно** с конфигурируемым параметром **HYSVAL**.

Используя преобразователи типов (например, **DINT_TO_REAL**), вы можете производить универсальные сравнения (например, сравнение счетно-импульсных величин). Результаты сравнения выводятся на выходах **EQUAL (Равно)**, **GREATER (Больше)** и **LESS (Меньше)** в виде логических величин.

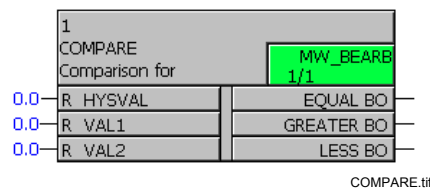


Рис. 4-54 Блок COMPARE

Назначение входов / выходов

Блок **COMPARE** имеет следующее назначение входов / выходов:

Таблица 4-82 Назначение входов / выходов блока COMPARE

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	HYSVAL	REAL	Величина параметра	0,0
	VAL1	REAL	Операнд 1	0,0
	VAL2	REAL	Операнд 2	0,0
Выходы:	EQUAL	BOOL	Величины равны (VAL1 = VAL2)	0
	GREATER	BOOL	Величина 1 больше, чем величина 2 (VAL1 > VAL2)	0
	LESS	BOOL	Величина 1 меньше, чем величина 2 (VAL1 < VAL2)	0

Поведение выходов при HYSVAL = 0

При величине параметра HYSVAL = 0 (простое сравнение двух величин), выходы ведут себя следующим образом:

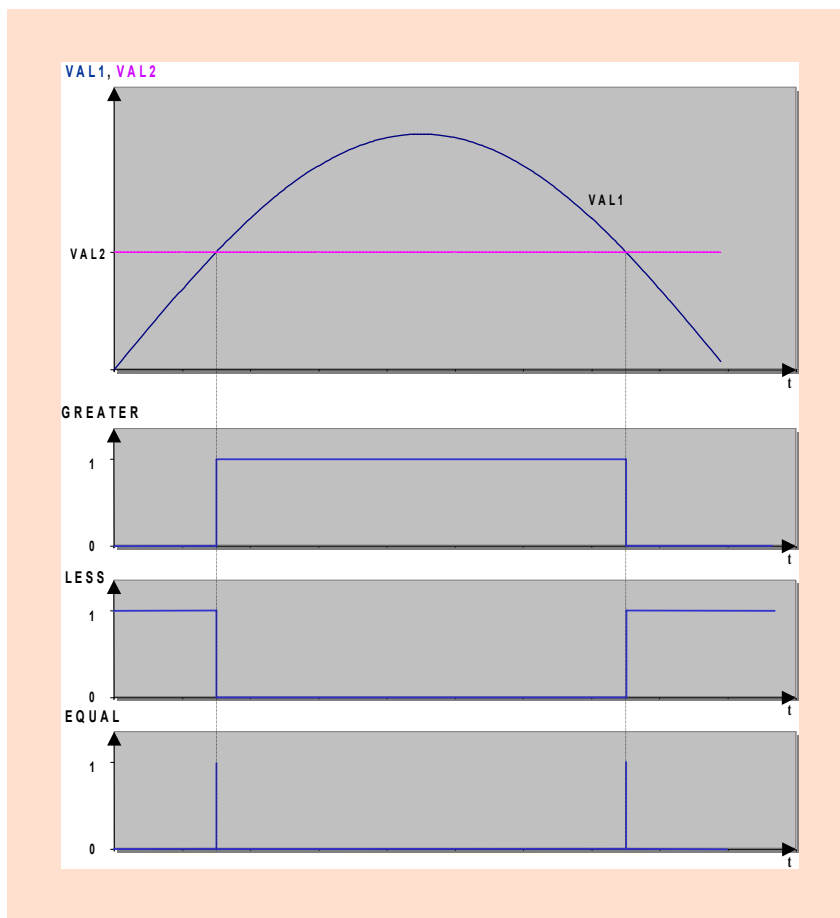


Рис. 4-55 Схема для HYSVAL = 0 (простое сравнение двух величин)

Поведение выходов при HYSVAL > 0

При величине параметра HYSVAL > 0 (параметр рядом с нулевой точкой), выходы ведут себя следующим образом:

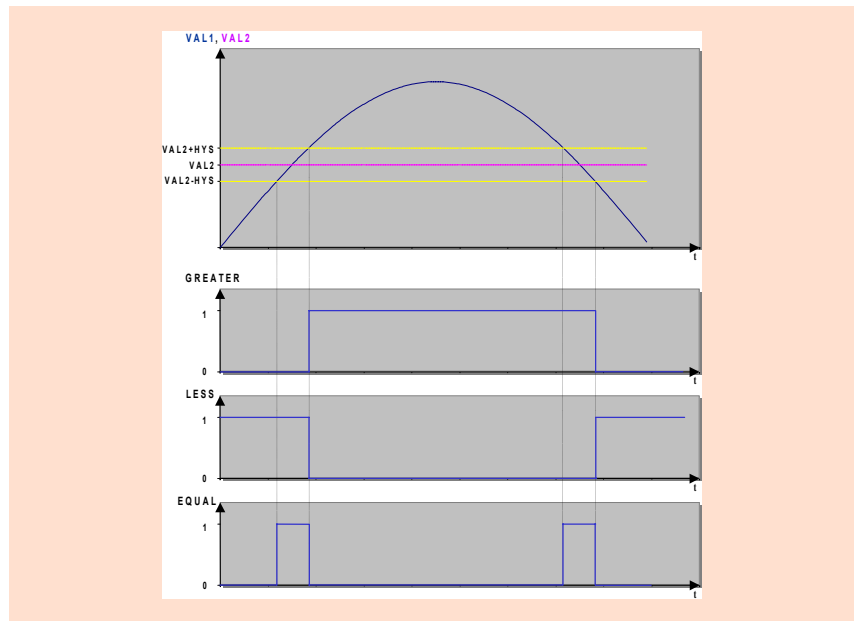


Рис. 4-56 Схема для HYSVAL > 0 (гистерезис рядом с нулевой точкой)

Поведение выходов при HYSVAL < 0

При величине параметра HYSVAL < 0 (сравнение с выдержанной по времени точкой возврата), выходы ведут себя следующим образом:

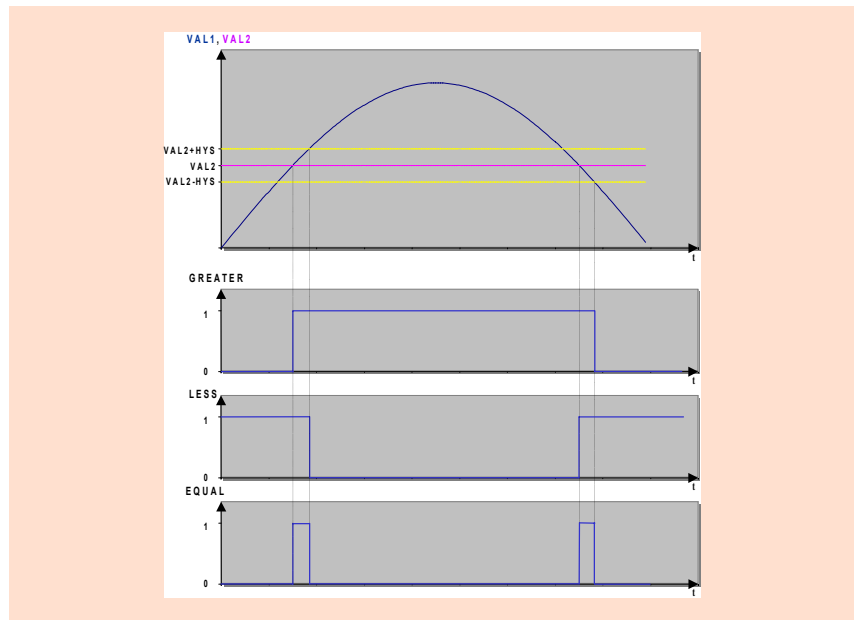


Рис. 4-57 HYSVAL < 0 (сравнение с выдержанной по времени точкой возврата)

4.8.2 LIVE_ZERO



Примечание:

Блоки **LIVE_ZERO**, **LOWER_SETPOINT**, **UPPER_SETPOINT** и **ZERO_POINT** предназначены только для обработки измеряемых величин.

Функция

Для того чтобы обнаружить повреждения датчика или замыкания в проводах между измерительным преобразователем и датчиком, полученным измеряемым величинам не позволяют изменяться от 0 % до 100 %, а устанавливают нижний предел, равный 20 %. Величины ниже нулевой пороговой величины интерпретируются как повреждения, в результате чего блок **Live Zero Monitoring (Контроль нулевой точки)** формирует соответствующее сообщение.

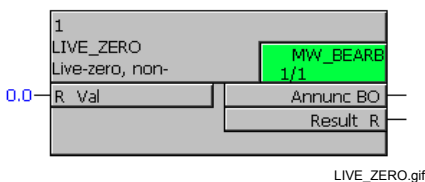


Рис. 4-58 Блок LIVE_ZERO

**Назначение
входов / выходов**

Блок **LIVE_ZERO** имеет следующее назначение входов / выходов:

Таблица 4-83 Назначение входов / выходов блока LIVE_ZERO

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Val	REAL	Измеряемая величина в %	0,0
	LiveZero	REAL	Нулевая пороговая величина в % (0,0 %)	20,0
	DetecKnee	REAL	Обнаружение точки изгиба в % (0,0 % и < 100,0 %)	50,0
	DispKnee	REAL	Отображенная точка изгиба в % (0,0...200,0%)	70,0
Выходы:	Result	REAL	Измеренная величина нулевой пороговой величины в %	0,0
	Annunc	BOOL	Сообщение: Контроль нулевой пороговой величины	0

**Примечание:**

Входы **LiveZero**, **DetecKnee** и **DispKnee** могут быть сконфигурированы при помощи контекстного меню **Object Properties (Свойства объекта)**.

**Примечание:**

Для того чтобы избежать появления сообщения **Live Zero Monitoring**, при формировании сообщения применяется гистерезис с коэффициентом возврата 0,95 (минимум 0,5 %).

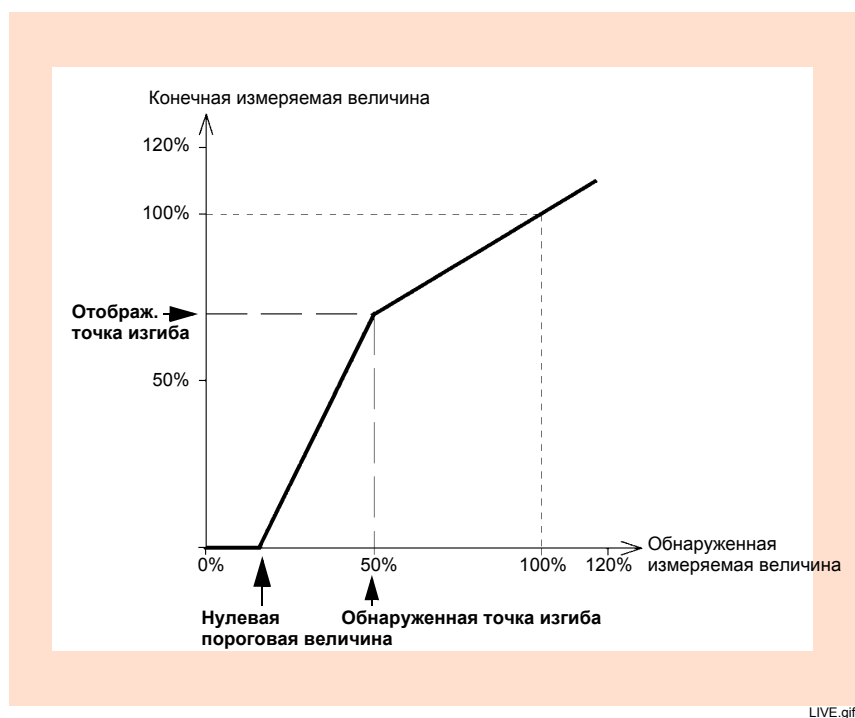


Рис. 4-59 График контроля нулевой пороговой величины

**Внимание:**

Если вы неправильно сконфигурируете блок **LIVE ZERO** и значения на входах **LiveZero**, **DetecKnee** или **DispKnee** будут вне диапазона допустимых величин, выход **Result** будет установлен на **NOT_CALCULATED**. Эта измеряемая величина отображается на дисплее устройства SIPROTEC в виде трех точек....

4.8.3 LOWER_SETPOINT



Примечание:

Блоки **LIVE_ZERO**, **LOWER_SETPOINT**, **UPPER_SETPOINT** и **ZERO_POINT** предназначены только для обработки измеряемых величин.

Функция

Блок **Lower Limit (Нижний предел)** формирует сигнал на выходе **Annunc**, если измеряемая величина на входе **Val** ниже запараметрированной предельной величины **Limit (Предел)**.

Нижний предел может быть соединен с выходным сигналом другого блока или задан как фиксированное значение.

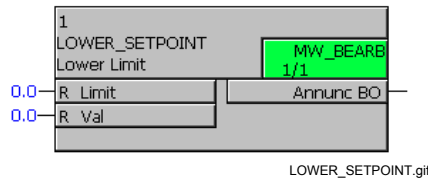


Рис. 4-60 Блок LOWER_SETPOINT

Назначение входов / выходов

Блок **LOWER_SETPOINT** имеет следующее назначение входов / выходов:

Таблица 4-84 Назначение входов / выходов блока LOWER_SETPOINT

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Limit	REAL	Предел в % ($\pm 1,0 \text{ e}^{+38}$)	0,0
	Val	REAL	Измеряемая величина в % ($\pm 1,0 \text{ e}^{+38}$)	0,0
Выходы:	Annunc	BOOL	Сообщение: нижний предел	0



Примечание:

Для того чтобы избежать появления сообщения **Lower Limit**, при формировании сообщения применяется гистерезис с коэффициентом возврата 0,95 (минимум 0,5 %).



Примечание:

Значение на входе **Val** с состоянием **Overflow (Переполнение)** дает значение 0 на выходе **Annunc**.

Пример использования

При помощи представленной ниже схемы CFC, производится контроль трех фазных токов:
Если все три фазных тока ниже 5 % номинального тока, набор уставок переключается.

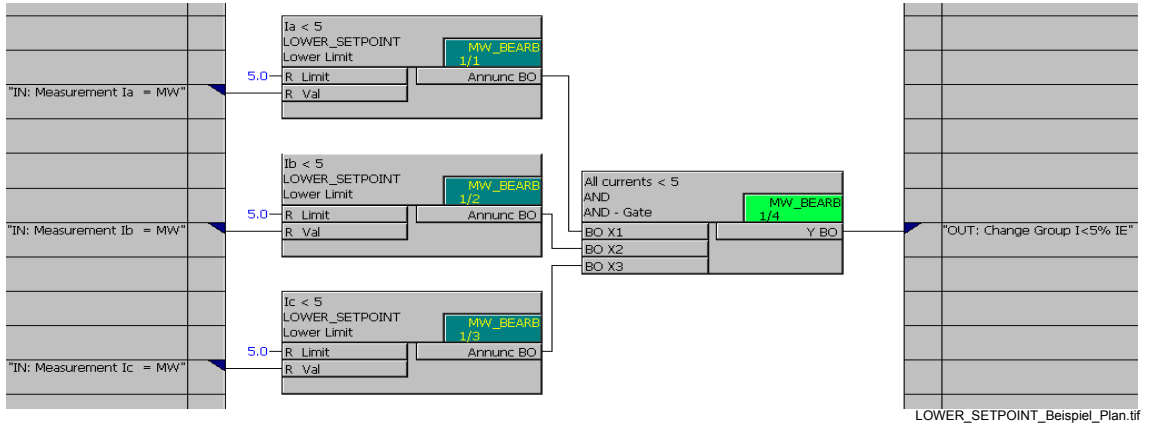


Рис. 4-61 Пример использования блока LOWER_SETPOINT, фрагмент схемы

4.8.4 UPPER_SETPOINT

**Примечание:**

Блоки **LIVE_ZERO**, **LOWER_SETPOINT**, **UPPER_SETPOINT** и **ZERO_POINT** предназначены только для обработки измеряемых величин.

Функция

Блок **Upper Limit (Верхний предел)** формирует сигнал на выходе **Annunc**, если измеряемая величина на входе **Val** превышает запараметрированную предельную величину **Limit (Предел)**.

Верхний предел может быть соединен с выходным сигналом другого блока или задан как фиксированное значение.

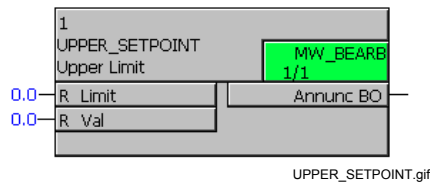


Рис. 4-62 Блок UPPER_SETPOINT

Назначение входов / выходов

Блок **UPPER_SETPOINT** имеет следующее назначение входов / выходов:

Таблица 4-85 Назначение входов / выходов блока UPPER_SETPOINT

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Limit	REAL	Предел в % ($\pm 1,0 \cdot 10^{38}$)	0,0
	Val	REAL	Измеряемая величина в % ($\pm 1,0 \cdot 10^{38}$)	0,0
Выходы:	Annunc	BOOL	Сообщение: превышение верхнего предела	0

**Примечание:**

Для того чтобы избежать появления сообщения **Upper Limit Exceeded (Превышение верхнего предела)**, при формировании сообщения применяется гистерезис с коэффициентом возврата 0,95 (минимум 0,5 %).



Примечание:

Значение на входе **Val** с состоянием **Overflow (Переполнение)** дает значение 1 на выходе **Annunc**.

4.8.5 ZERO_POINT

**Примечание:**

Блоки **LIVE_ZERO**, **LOWER_SETPOINT**, **UPPER_SETPOINT** и **ZERO_POINT** предназначены только для обработки измеряемых величин.

Функция

В результате погрешностей измерений, измеренное нулевое значение может немного отличаться от действительного значения. Для компенсации этого эффекта, измеряемая величина, значение которой находится ниже установленной пороговой величины, устанавливается на ноль. Эту задачу выполняет блок **Zero Supression (Подавление незначущих нулей)**.

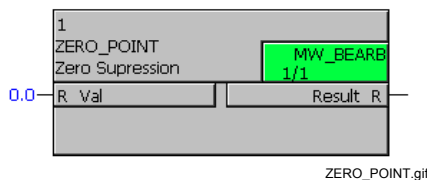


Рис. 4-63 Блок ZERO_POINT

Назначение входов / выходов

Блок **ZERO_POINT** имеет следующее назначение входов / выходов:

Таблица 4-86 Назначение входов / выходов блока ZERO_POINT

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Val	REAL	Измеряемая величина в %	0,0
	ZeroPoint	REAL	Нулевая величина в % (0,0...30,0%)	5,0
Выходы:	Result	REAL	Измеряемая величина с подавлением незначущих нулей в %	0,0

**Примечание:**

Вход **ZeroPoint** может быть сконфигурирован, выбрав пункт контекстного меню **Object Properties (Свойства объекта)**.

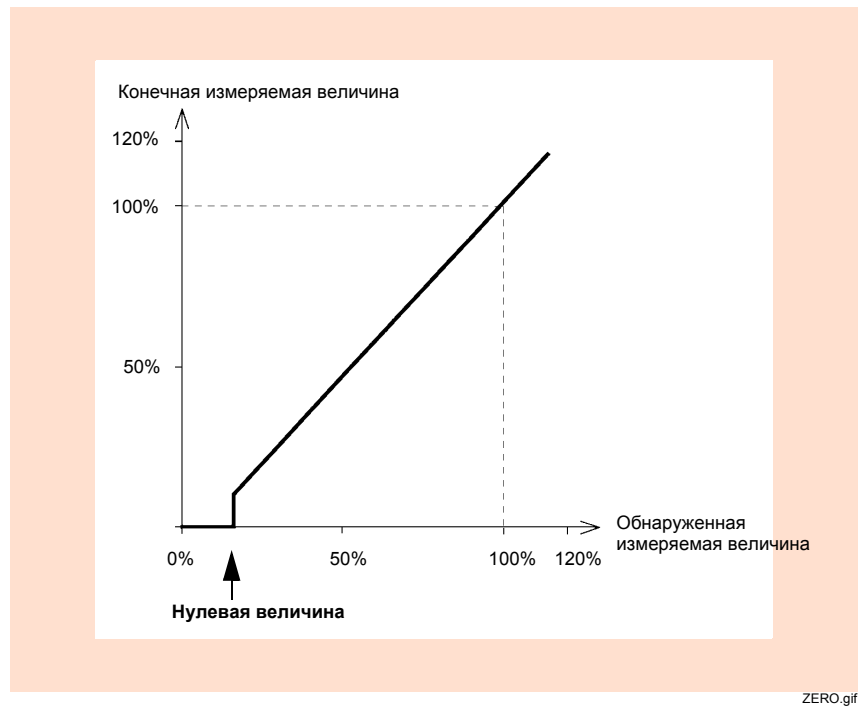


Рис. 4-64 График подавления незначущих нулей

**Внимание:**

Если вы неправильно сконфигурируете блок **ZERO_POINT** и значения на входах **Val** или **ZeroPoint** будут вне диапазона допустимых величин, выход **Result** будет установлен на **NOT_CALCULATED**. Эта измеряемая величина отображается на дисплее устройства SIPROTEC в виде трех точек

4.9 Counter Value (Счетно-импульсная величина)

При помощи блоков счетно-импульсных величин вы можете реализовать функцию счетчика.

Существуют следующие блоки счетно-импульсных величин:

- **COUNTER (Счетчик)**

4.9.1 COUNTER (Счетчик)

Функция

Блок **counter (счетчик)** подсчитывает положительные фронты импульсов на входе **Clk** и добавляет величину **DELTA** к сохраненной величине при каждом положительном фронте.

При первом запуске показание счетчика **VAL** инициализируется величиной, активной на входе **BEGIN**.

При последующем запуске используется старая величина счетчика **VAL**, сохраненная в энергонезависимой памяти.

Если показание счетчика превышает величину **END**, счетчик инициализируется величиной, активной на входе **BEGIN**. При этом выход **OVERFLOW** изменяется с 0 на 1, что является свидетельством переполнения. Сигнал переполнения остается установленным до тех пор, пока значение 0 не появится опять на входе **Clk** или пока устройство не перезапустится.

Счетно-импульсная величина может быть явно установлена на величину входа **BEGIN** при наличии нарастающего фронта импульса на входе **RESET**.

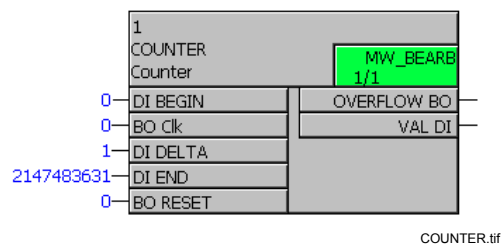


Рис. 4-65 Блок COUNTER

**Примечание:**

Максимально допустимое количество блоков **RS_FF_MEMO**, **SR_FF_MEMO**, **D_FF_MEMO** и **COUNTER** зависит от имеющейся энергонезависимой памяти и контролируется компилятором CFC. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.

Максимально допустимое количество проверяется во время компиляции схемы CFC. При наличии ошибки, система указывает на противоречия. Если расходуется большее количество ресурсов, чем предусмотрено, то это указывается в отображаемом на экране протоколе компиляции.

**Примечание:**

Следующее должно быть учтено для того, чтобы соединить **быстрые сигналы** (например, нажимая функциональную кнопку на устройстве SIPROTEC) с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**):

Логические входы не являются запускающим событием для этих классов приоритетов и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки класса приоритетов.

Назначение входов / выходов Блок **COUNTER** имеет следующее назначение входов / выходов:

Таблица 4-87 Назначение входов / выходов блока COUNTER

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	BEGIN	DINT	Начальное значение счетчика	0
	Clk	BOOL	Подсчитывает положительные фронты импульсов	0
	DELTA	DINT	Изменение счетно-импульсной величины при положительном фронте импульса на входе Clk	0
	END	DINT	Конечное значение счетчика	2147483631
	RESET	BOOL	Устанавливает счетчик на начальное значение	0
Выходы:	VAL	DINT	Текущее показание счетчика	Сохраненное значение
	OVERFLOW	BOOL	Переполнение счетчика Указывает на достижение конечного значения	0

**Примечание:**

Если величина на входах **BEGIN**, **DELTA** и **END** находится вне допустимого диапазона (тип данных DINT: - **2147483631** до **2147483631**), то выход **VAL** устанавливается на **NOT_CALCULATED** (выход **OVERFLOW** = 0). Входы **Clk** и **RESET** не анализируются. Внутренняя счетно-импульсная величина не изменяется.

Если на входах **BEGIN**, **DELTA** и **END** опять появляются новые достоверные величины, то на выходе **VAL** выдается внутренняя счетно-импульсная величина.

4.10 Time & Clock (Время и тактовые импульсы)

При помощи блоков времени и тактовых импульсов вы можете управлять функциями в зависимости от времени.

Существуют следующие блоки времени и тактовых импульсов:

- **ALARM (Аварийная сигнализация)**
- **BLINK** (блок мигания)
- **LONG_TIMER** (таймер (макс. 1 193ч))
- **TIMER** (универсальный таймер)
- **TIMER_SHORT** (простой таймер)

4.10.1 ALARM (Аварийная сигнализация)

Функция

Блок **Alarm (Аварийная сигнализация)** указывает с помощью изменения значения 0 на 1 на выходе **Q** на достижение времени тревоги, заданного входными величинами.

Q остается активным на время совпадения (1 секунда).

При помощи символов-заполнителей вы можете указывать время, позволяющее осуществлять циклический запуск (ежегодно, ежемесячно, ежедневно, один раз в минуту).

Если задано неверное время (например, 31 февраля), это отображается при помощи выхода **ERR**. Выход **ERR** остается активным до тех пор, пока опять не будет задано достоверное время.



Примечание:

Несоответствия во времени (например, при переходе с зимнего на летнее время) при этом не учитываются. Если время находится в пределах этого временного интервала, срабатывание либо не производится, либо производится двойное срабатывание.

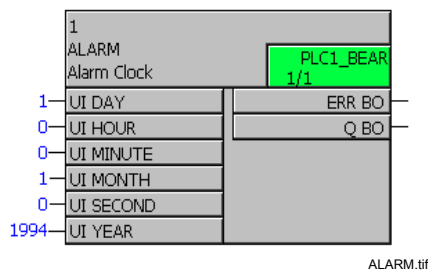


Рис. 4-66 Блок ALARM

Назначение входов / выходов Блок **ALARM** имеет следующее назначение входов / выходов:

Таблица 4-88 Назначение входов / выходов блока ALARM

	Название	Тип данных	Примечание	Символ-заполнитель	Значение по умолчанию
Входы:	DAY	UINT	День	0	1
	HOUR	UINT	Час	24	0
	MINUTE	UINT	Минута	60	0
	MONTH	UINT	Месяц	0	1
	SECOND	UINT	Секунда	без	0
	YEAR	UINT	Год	0	1994
Выходы:	ERR	BOOL	Недостоверная дата		0
	Q	BOOL	Время аварийной сигнализации достигнуто (установлена 1 секунда)		0

4.10.2 BLINK (Мигание)

Функция

Блок мигания был разработан преимущественно для подачи сигналов на осветительные приборы (например, светодиоды). Для этого у блока имеется один вход для задания времени горения и один для затухания.

Блок мигания запускается с изменением состояния сигнала с 0 на 1 на входе **ENABLE**. На выходе **Q** сигнал переключается с 0 на 1 и наоборот, в зависимости от установленного времени. Мигание блока опять прекращается с изменением состояния сигнала с 1 на 0.

Блок мигания запускается всегда с фазы горения **THx100ms** и может быть прерван в любое время. После прерывания, на выходе **Q** блока мигания всегда выдается значение 0.



Примечание:

Разрешение таймера мигания составляет 100 мсек. Таким образом, в зависимости от начального времени таймера, первая фаза горения может быть до 99 мсек выше установленного значения **THx100ms**.

Наименьшее время цикла составляет 100 мсек. Даже для значений, составляющих менее 100 мсек для фазы горения **THx100ms** или фазы затухания **TLx100ms**, это время цикла не может быть меньше нижнего предела. Если же величина ниже минимального предела, то автоматически используется значение 100 мсек.

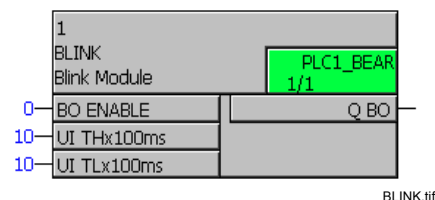


Рис. 4-67 Блок Blink



Примечание:

Следующее применимо для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**):

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов Блок **BLINK** имеет следующее назначение входов / выходов:

Таблица 4-89 Назначение входов / выходов блока BLINK

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	ENABLE	BOOL	Разрешение Запускает и останавливает блок мигания	0
	THx100ms	UINT	Время для фазы горения (разрешение 100 мсек)	10 (= 1 сек)
	TLx100ms	UINT	Время для фазы затухания (разрешение 100 мсек)	10 (= 1 сек)
Выходы:	Q	BOOL	Переключается во время работы в соответствии с фазой горения и затухания между 1 и 0	0

4.10.3 LONG_TIMER

Функция

При помощи блока **Long Timer (Длинный таймер)** вы можете задавать выдержки времени в часах.

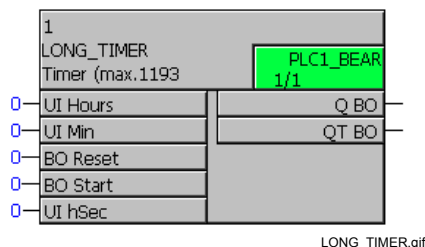


Рис. 4-68 Блок LONG_TIMER



Примечание:

Для соединения **быстрых сигналов** с левой границей схемы CFC в классах приоритетов обработки измеряемых величин (класс приоритетов **MW_BEARB**) и блокировки (класс приоритетов **SFS_BEARB**) необходимо учесть следующее:

Логические входы не являются запускающим событием для этого класса и могут оставаться **незамеченными**, если событие или сигнал короче, чем цикл обработки обрабатывающего класса приоритетов.

Назначение входов / выходов

Блок **LONG_TIMER** имеет следующее назначение входов / выходов:

Таблица 4-90 Назначение входов / выходов блока LONG_TIMER

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Hours	UINT	Значение для целых часов	0
	Min	UINT	Значение для целых минут	0
	Reset	BOOL	Сброс таймера	0
	hSec	UINT	Значение для секунд (разрешение 100 мсек)	0
	Start	BOOL	Запуск или перезапуск таймера (при переходе 0-1)	0
Выходы:	Q	BOOL	Время истекло	0
	QT	BOOL	Таймер все еще запущен	0

Перезапуск длинного таймера

Блок **Long Timer (Длинный таймер)** может быть перезапущен: При повторном изменении сигнала с 0 на 1 на входе **Start (Запуск)**, запущенный таймер прерывается и перезапускается с использованием заданных величин времени.

Вход RESET

Запущенный таймер прерывается по сигналу на входе сброса **Reset**. Выходы **Q** и **QT** устанавливаются на 0.

**Примечание:**

Следующее относится к устройствам SIPROTEC с версией **ниже, чем V4.5**:

Максимальная выдержка времени составляет 1 193 ч. Если в параметрах **Hours (Часы)**, **Min (Минуты)** и **Sec (Секунды)** вы задаете время больше, чем 1 193 ч, то в устройстве SIPROTEC выводится внутреннее сообщение об ошибке. Время завершения установлено на 0.

4.10.4 TIMER (Таймер)

Функция

Блок **Universal Timer (Универсальный таймер)** позволяет вам реализовать различные функции таймера:

- работа обычного таймера,
- работа таймера с повторным пуском,
- работа таймера с выдержкой времени,
- работа таймера с удлинением импульсов.

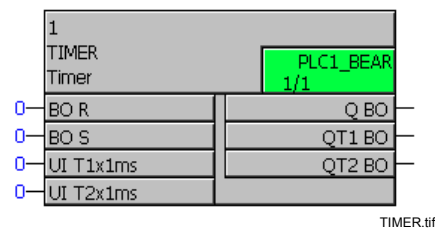


Рис. 4-69 Блок TIMER



Примечание:

Блок **Universal Timer** работает только в классах приоритетов:

- быстрая логика Fast PLC (класс приоритетов **PLC_BEARB**) и
- медленная логика Slow PLC (класс приоритетов **PLC1_BEARB**).



Примечание:

Простой альтернативой блоку **Universal Timer (TIMER)** является блок **простого таймера (TIMER_SHORT)**.

Максимально допустимое количество блоков **TIMER** и **TIMER_SHORT** ограничено имеющимися в распоряжении системы таймерами и контролируется компилятором CFC. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.

Максимально допустимое количество проверяется во время компиляции схемы CFC. При наличии ошибки, система указывает на противоречия. Если расходует больше количество ресурсов, чем предусмотрено, то это указывается в отображаемом на экране протоколе компиляции.

Назначение входов / выходов

Блок **TIMER** имеет следующее назначение входов / выходов:

Таблица 4-91 Назначение входов / выходов блока TIMER

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	S	BOOL	Вход установки	0
	R	BOOL	Вход сброса	0
	T1x1ms	UINT	Значение для T1 (разрешение 1мсек)	0
	T2x1ms	UINT	Значение для T2 (разрешение 1мсек)	0
Выходы:	Q	BOOL	Выход Q	0
	QT1	BOOL	Выход T1	0
	QT2	BOOL	Выход T2	0

**Примечание:**

Максимально допустимые величины времени для **T1x1ms** и **T2x1ms** зависят от разрешения по времени используемого устройства SIPROTEC. Если используемые величины времени меньше, чем разрешение по времени, таймеры не будут запущены при появлении стартового импульса. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.

Работа и соединение отдельных типов таймеров описано ниже:

Вход сброса

Вход сброса (Reset) действует одинаково при всех функциях таймера (на рисунках не показан):

Сигнал на входе сброса **R** прерывает все запущенные таймеры.

Выходы **QT1** и **QT2** устанавливаются на 0.

Сигнал на входе **S** напрямую передается на выход **Q**.

Работа обычного таймера

Следующее относится к работе обычного таймера:

T1 запускается при положительном фронте сигнала на входе **S**.

Если время **T1** истекло при активном сигнале на входе **S**, то сигнал выдается на выходе **Q**.

Когда сигнал поступает на выход **Q**, и при этом пропадает сигнал на входе **S**, начинается отсчет времени **T2**.

Время можно перезапустить.

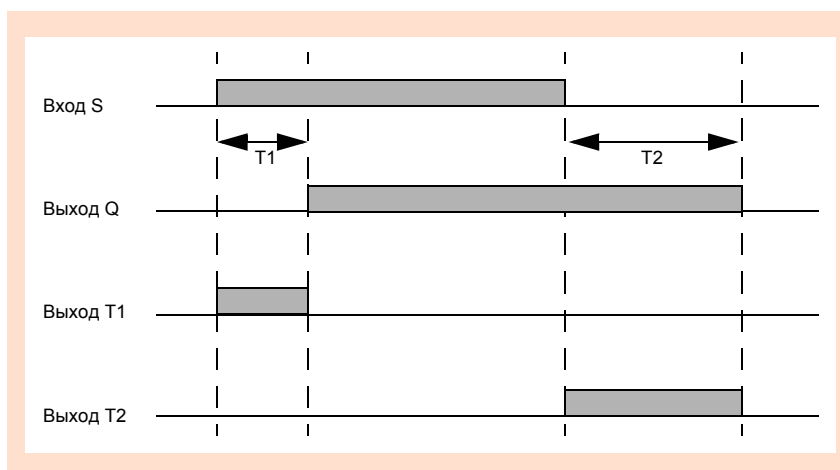


Рис. 4-70 График работы обычного таймера

Работа таймера с повторным пуском

Следующее относится к работе таймера с повторным пуском:

T2 установлен на 0.

Используется только выход **T1**.

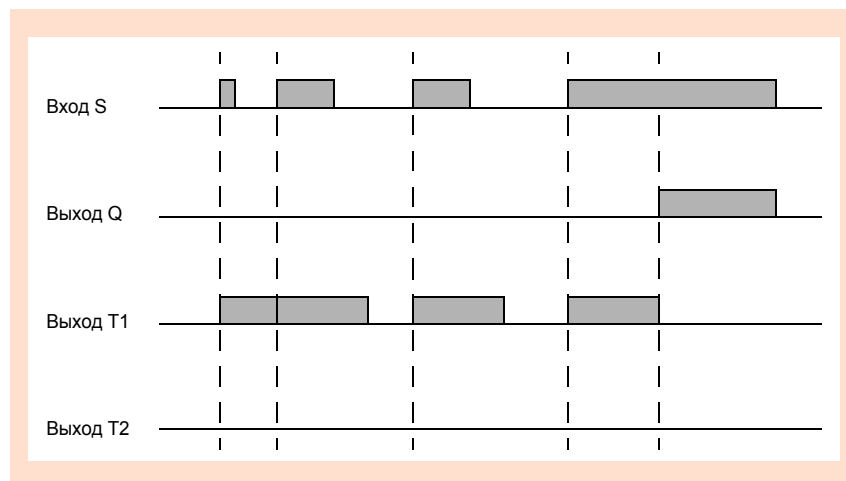


Рис. 4-71 График работы таймера с повторным пуском

Работа таймера с выдержкой времени

Следующее относится к работе таймера с выдержкой времени:

T2 установлен на 0.

Используется только выход **Q**.

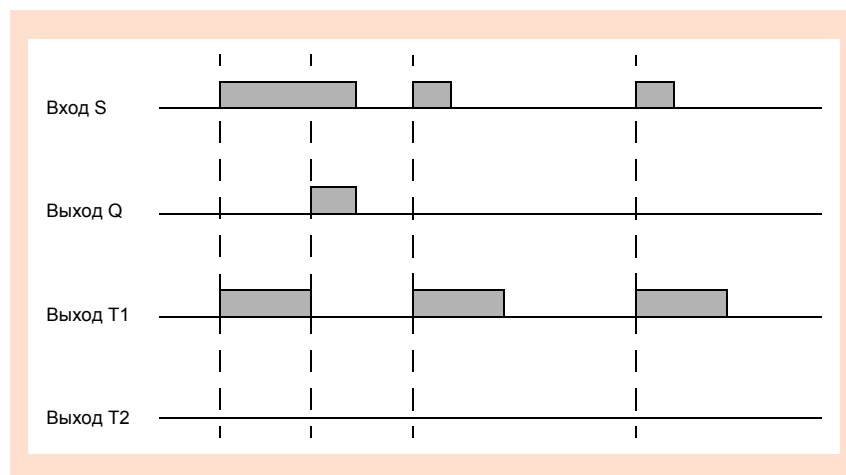


Рис. 4-72 График работы таймера с выдержкой времени

Работа таймера с удлинением импульсов

Следующее относится к работе таймера с удлинением импульсов: **T1** установлен на 0.
Используется только выход **Q**.

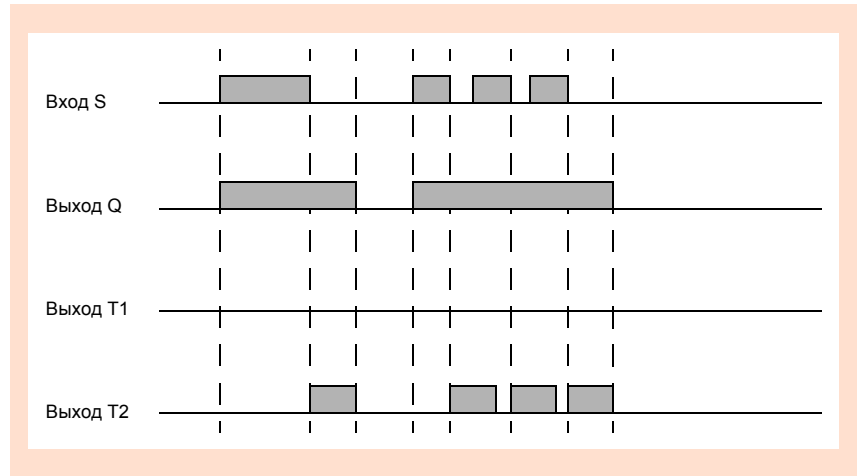


Рис. 4-73 График работы таймера с удлинением импульсов

Пример использования

При помощи показанной внизу схемы CFC, вы можете, например, реализовать простую последовательность переключения: Если нажата **F1**, то светодиод **LED1** загорается на 5 секунд. Затем загорается светодиод **LED2** на 15 секунд.

- Вставьте в матрице ранжирования необходимую информацию, как показано в следующем снимке экрана:

	Информация				Источник			Цель																					
	Номер	Текст дисплея	НС	Тип	Вх	Ф	С	Л	Вы	СД																			
										1	2	3	4	5	6	7	8	9	10	11	12	13	14	П	С	Л	Д	МЧ	
Device, General							x	x																					
LED Sequence		F1 LED		IntSP		1																							
		LED 1		IntSP				x		Б																			
		LED 2		IntSP				x		Б																			

Рис. 4-74 Пример использования блока таймера, фрагмент матрицы ранжирования

- Создайте следующую схему CFC в классе приоритетов **PLC1_BEARB** (медленная логика Slow PLC):

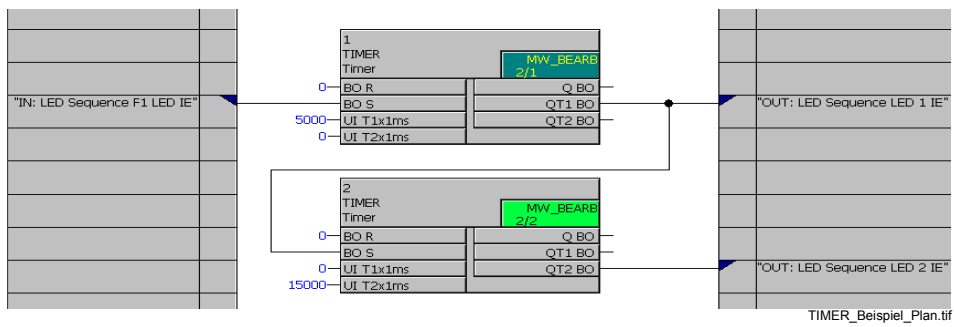


Рис. 4-75 Пример использования блока таймера, фрагмент схемы CFC

4.10.5 TIMER_SHORT

Функция

При помощи блока **simple timer (простой таймер)** могут быть реализованы простые временные задачи (например, выдержки времени). Здесь вы можете установить время до 65,535 секунд с разрешением 1 мсек.

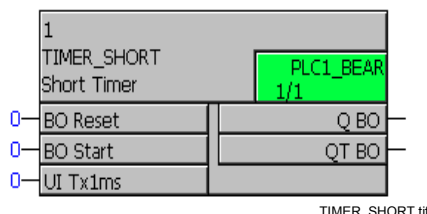


Рис. 4-76 Блок TIMER_SHORT



Примечание:

Блок **simple timer (простой таймер)** работает только в классах приоритетов:

- быстрая логика Fast PLC (класс приоритетов **PLC_BEARB**) и
- медленная логика Slow PLC (класс приоритетов **PLC1_BEARB**).



Примечание:

Блок **простого таймера (TIMER_SHORT)** является простой альтернативой блоку **универсального таймера (TIMER)**.

Максимально допустимое количество блоков **TIMER** и **TIMER_SHORT** ограничено имеющимися в распоряжении системы таймерами и контролируется компилятором CFC. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.

Максимально допустимое количество проверяется во время компиляции схемы CFC. При наличии ошибки, система указывает на противоречия. Если расходуется большее количество ресурсов, чем предусмотрено, то это указывается в отображаемом на экране протоколе компиляции.

Назначение входов / выходов

Блок **TIMER_SHORT** имеет следующее назначение входов / выходов:

Таблица 4-92 Назначение входов / выходов блока **TIMER_SHORT**

	Название	Тип данных	Примечание	Значение по умолчанию
Входы:	Reset	BOOL	Вход сброса	0
	Start	BOOL	Вход установки	0
	Tx1ms	UINT	Значение времени (разрешение 1 мсек)	0
Выходы:	Q	BOOL	Время истекло	0
	QT	BOOL	Время еще идет	0

**Примечание:**

Максимально допустимая величина времени для **T1x1ms** зависит от разрешения по времени используемого устройства SIPROTEC. Если используемые величины времени меньше, чем разрешение по времени, таймеры не будут запущены при появлении стартового импульса. Обратите внимание на технические данные, приведенные в руководстве по эксплуатации устройства SIPROTEC, которое вы хотите использовать.

Перезапуск простого таймера

Блок **simple timer** может быть перезапущен: При повторном изменении сигнала с 0 на 1 на входе **Start (Запуск)**, запущенный таймер прерывается и перезапускается с использованием заданной величины времени.

Вход сброса

Запущенный таймер прерывается по сигналу на входе сброса **Reset**. Выходы **Q** и **QT** устанавливаются на 0.

Функция таймера Следующий рисунок показывает функциональную схему TIMER_SHORT:

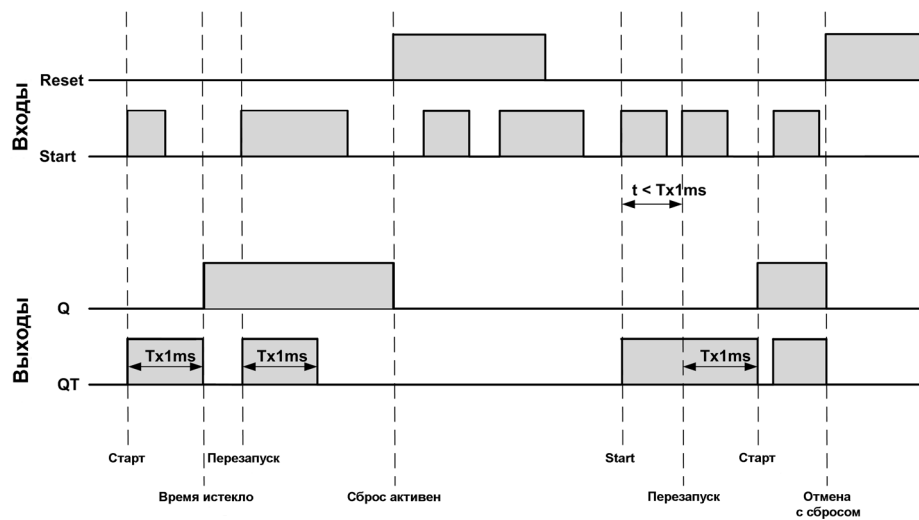


Рис. 4-77 Схема TIMER_SHORT

- /1/ SIPROTEC 4, Описание системы
E50417-H1100-C151
- /2/ SIPROTEC DIGSI 4, Ввод в эксплуатацию
E50417-G1100-C152

Словарь

Устройства управления присоединением	Устройства управления присоединением - это устройства с функциями управления и контроля, не имеющие защитных функций.
Дисплей управления	Изображение, которое отображается на дисплее устройства с большим (графическим) дисплеем после того, как вы нажали клавишу управления, называется дисплеем управления. Дисплей содержит управляемые коммутационные устройства в присоединении, с отображением их состояния. Он используется для выполнения операций переключения. Определение этого дисплея является частью конфигурационных работ.
Очистка памяти	Частое добавление и удаление объектов приводит к увеличению областей памяти, которые не могут больше использоваться. В результате очистки, эти области памяти освобождаются для их последующего использования. Однако, при очистке также производится переназначение VD адресов. Это означает, что все устройства SIPROTEC 4 должны быть вновь инициализированы.
Комбинированные устройства	Комбинированные устройства - это устройства присоединения с функциями защиты и управления.
Матрица связи	До 16 совместимых устройств SIPROTEC 4 могут связываться друг с другом посредством IRC (шина передачи данных между устройствами). Какое устройство обменивается, какой информацией определяется в матрице связи.
Указатель связи CR	Указатель связи описывает тип и версию устройства, которое участвует в связи PROFIBUS.
Детальный вид	В дополнение к топологическому виду Диспетчер SIMATIC предлагает вам детальный вид. Данный вид служит не для предоставления обзора иерархии проекта, а для обзора всех устройств SIPROTEC 4, находящихся в проекте.
Контейнер	Общее понятие, обозначающее объекты, состоящие в свою очередь, из других объектов. Объект типа Папка является примером такого контейнера.

Контейнер устройств	В детальном виде, все устройства SIPROTEC 4 назначены объекту типа контейнер устройств . Данный объект является специальным объектом DIGSI 4 Manager. Однако, так как в DIGSI 4 Manager нет детальное вида, этот объект доступен только при использовании программы STEP 7.
Устройства присоединения	Общее понятие для всех устройств уровня присоединения: устройства защиты, комбинированные устройства, устройства управления присоединением.
Папка	Данный тип объекта используется для создания иерархической структуры проекта.
HV описание присоединения	HV файл описания проекта содержит подробную информацию о присоединениях, имеющихся в проекте ModPara. Текущая информация присоединения для каждого присоединения сохраняется в HV файле описания проекта. В HV файле описания проекта каждое присоединение обращается к такому файлу описания присоединения HV по ссылке на имя файла.
IEC адрес	При использовании шины IEC каждому устройству SIPROTEC 4 необходимо присвоить уникальный IEC адрес. Каждой шине можно присвоить до 254 IEC адресов.
Строка инициализации	Строка инициализации включает в себя набор команд, зависящих от типа модема. Эти команды передаются модему в процессе его инициализации. Такими командами могут быть, например, специальные уставки для модема.
IRC связь	Шина передачи данных, IRC, используется для прямого обмена информацией между устройствами SIPROTEC 4. Для конфигурации шины передачи данных необходим объект типа IRC связь . В этом объекте определяется каждый пользователь связи, а также все необходимые коммуникационные параметры. Тип и объем информации, которой обмениваются пользователи, также хранится в этом объекте.
Адрес связи	Адрес связи указывает адрес устройства версии V3 / V2. Этот адрес может быть изменен при помощи программы DIGSI 3.x.
Списочный вид	В правой части окна проекта отображаются названия и иконки объектов, которые содержатся в контейнере, выбранном в древовидном представлении. Поскольку они отображаются в виде списка, эта часть окна называется списочным видом.

Код заказа MLFB	<p>MLFB - это сокращение от немецкого Maschinenlesbare Fabrikatebezeichnung (машинно-считываемое наименование изделия).</p> <p>Это эквивалент коду заказа. Тип и версия устройства SIPROTEC 4 закодированы в этом коде заказа.</p>
Модемное соединение	<p>Этот тип объекта содержит информацию об обоих участниках модемного соединения, локальном модеме и удаленном модеме.</p>
Профиль модема	<p>Профиль модема состоит из названия профиля и драйвера модема. Он также может включать в себя несколько команд инициализации и адрес пользователя. Для одного модема вы можете создать несколько профилей модема. Для этого вам потребуется привязать разные команды инициализации или адреса пользователей к драйверу модема и его свойствам и сохранить их под разными именами.</p>
Модемы	<p>В этом объекте хранятся профили модемов для модемного соединения.</p>
Объект	<p>Общее понятие, обозначающее каждый элемент в структуре проекта CFC.</p>
Свойства объекта	<p>Каждый объект обладает свойствами. Это могут быть общие свойства, которые являются общими для нескольких объектов, или же объект может также обладать специальными свойствами.</p>
Набор уставок	<p>Набор уставок - это совокупность всех уставок, которые могут быть заданы для устройства SIPROTEC 4.</p>
Телефонная книга	<p>В этом объекте хранятся адреса пользователей для модемного соединения.</p>
PROFIBUS	<p>PROcess Field BUS, немецкий стандарт полевой шины, определенный в международном стандарте EN 50170, Том 2, PROFIBUS. Он задает функциональные, электрические и механические свойства поразрядной полевой шины.</p>
PROFIBUS адрес	<p>В сети PROFIBUS, каждому устройству SIPROTEC 4 необходимо присвоить уникальный адрес PROFIBUS. Каждой сети PROFIBUS можно назначить до 254 адресов PROFIBUS.</p>

Проект	<i>По сути говоря</i> , проект - это отображение реальной энергосистемы. <i>Графически</i> , проект представлен рядом объектов, которые являются частью иерархической структуры. <i>На самом деле</i> , проект состоит из папок и файлов, содержащих данные проекта.
Устройства защиты	Все устройства с функциями защиты.
Сервисный интерфейс	Последовательный интерфейс, расположенный на задней панели устройств, который используется для подключения DIGSI 4 (например, через модем).
Параметрирование	Универсальный термин для всех настроек, производимых в устройстве. Параметрирование производится при помощи программы DIGSI 4 или, в некоторых случаях, прямо в устройстве.
SIPROTEC	Зарегистрированная торговая марка SIPROTEC 4 используется для устройств, построенных на основе системы V4.
Устройство SIPROTEC 4	Этот тип объекта представляет реальное устройство SIPROTEC 4 со всеми уставками и данными процесса, которые относятся к этому устройству.
Варианты SIPROTEC 4	Этот тип объекта представляет вариант объекта типа устройство SIPROTEC 4 . Данные устройства этого варианта могут вполне отличаться от данных устройства исходного объекта. Однако, все варианты, созданные из исходного объекта имеют тот же самый адрес VD, что и исходный объект. По этой причине они всегда соответствуют тому же самому реальному устройству SIPROTEC 4, что и исходный объект. Объекты типа Вариант SIPROTEC 4 могут использоваться для различных целей, например, для документирования различных состояний работы устройства SIPROTEC 4 при задании ему разных уставок.
Системный интерфейс	Последовательный интерфейс, расположенный на задней панели устройств, и используемый для подключения к системе управления через IEC или PROFIBUS.
Топологический вид	DIGSI 4 Manager всегда отображает проект в топологическом виде. Здесь отображается иерархическая структура проекта со всеми доступными объектами.
Древовидное представление	Левая часть окна проекта отображает названия и символы всех контейнеров проекта в виде дерева папок. Эта область называется древовидным представлением.

Адрес пользователя	Адрес пользователя включает в себя название устройства, код выхода на международную линию, код города и номер телефона пользователя.
Пользователи	До 16 совместимых устройств SIPROTEC 4 могут связываться друг с другом посредством связи IRC. Отдельные устройства, участвующие в IRC связи, называются пользователями.
Устройство V3/V2	Этот тип объекта указывает на существование данных устройства версии V3 или V2.
VD адрес устройства	VD адрес устройства автоматически присваивается DIGSI 4 Manager. Этот адрес является уникальным для всего проекта, и используется, таким образом, для однозначной идентификации реального устройства SIPROTEC 4. VD адрес, назначенный DIGSI 4 Manager, должен быть передан в соответствующее устройство SIPROTEC 4 для установления связи с Редактором устройств DIGSI 4.
VFD	VFD (V irtual F ield D evice) - виртуальное устройство присоединения включает в себя все объекты связи, их свойства и состояния, которые используются пользователями связи при помощи соответствующих служб.

Предметный указатель

A

ABSVALUE 78
ADD 79
AND 85

B

BOOL
 Тип данных 74
BOOL_TO_CO 138
BOOL_TO_DI 160
BOOL_TO_IC 143
BUILD_DI 162

C

CMD_CANCEL 147
CMD_CHAIN 149
CMD_INF 155
COMPARE 183
CONNECT 87
Connection 87
CV_GET_STATUS 104

D

D триггер 122
D триггер с памятью состояния 125
D_FF 122
D_FF_MEMO 125
DI_GET_STATUS 106
DI_SET_STATUS 107
DI_TO_BOOL 165
DINT
 Тип данных 74
DINT_TO_REAL 169
DIST_DECODE 170
DIV 80

DM_DECODE 173
DYN_OR 89

H

HV описание присоединения 218

I

IEC адрес 218
INT
 Тип данных 74
INT_TO_REAL 180
IRC комбинация 218

L

LIVE_ZERO 186
LONG_TIMER 204
LOOP 157
LOWER_SETPOINT 189

M

Memory (Память) 135
MUL 81
MV_GET_STATUS 109
MV_SET_STATUS 110

N

NAND 91
NAND gate 91
NEG 93
NOR 94

O

OR 96

P

PROFIBUS 219
PROFIBUS адрес 219

R

REAL
Тип данных 74
REAL_TO_DINT 175
REAL_TO_INT 177
REAL_TO_UINT 178
RISE_DETECT 98
RS триггер с памятью состояния 129
RS_FF_MEMO 129
RS_RS_MEMO 129

S

SHORT_TIMER 211
SI_GET_STATUS 112
SI_SET_STATUS 89
SIPROTEC 220
SQUARE_ROOT 82
SR триггер с памятью состояния 133
SR_FF_MEMO 133
SR_SR_MEMO 133
SUB 83
ST_AND 116
ST_NOT 118
ST_OR 119
STRUCT
Тип данных 74

U

UINT
Тип данных 74
UINT_TO_REAL 181
UPPER_SETPOINT 192

V

VD адрес устройства 221
VFD 221

X

X_OR 99

Z

ZERO_POINT 194

A

Абсолютное значение 78
ABSVALUE 78
Аварийная сигнализация 200
Адрес пользователя 221
Адрес связи 218

B

Вариант SIPROTEC 4 220
Верхний предел 192
Вид
Топологический вид 220
Воссоединение информационных элементов
Рекомендации по программированию 9
Вставка
схемы CFC 15
Входной сигнал
Ранжирование информации в матрице ранжирования 11
Выбор информации в качестве входного сигнала 11
Выбор информации в качестве выходного сигнала 12
Выходной сигнал
Ранжирование информации в матрице ранжирования 12
Вычитание 83

Г

Горячие клавиши 5

Д

Декодирование двухпозиционного сообщения 173

Декодирование двухпозиционного сообщения с состоянием 170

Деление 80
DIV 80

Детальный вид 217, 218

Детектор фронта 98

Диалоговое окно выбора левой границы
Отображенная информация 24Диалоговое окно выбора правой границы
Отображенная информация 27

Диапазон величин типов данных 74

Длинный таймер 204

Добавление новой информации 31

Добавление новой информации в матрицу ранжирования 31

Древовидное представление 220

З

Значение типов данных 74

И

Извлечение квадратного корня 82

Изменение названия блока 22

Изменение последовательности выполнения блока 23

Инвертор 93

Интерфейс SCADA 220

Информация о команде 155

Информация о состоянии

Тип данных DINT 75

Тип данных REAL 75

К

Класс приоритетов 4, 6, 41

Назначенные функции 6

Пример быстрой логики Fast PLC 14

Приоритет обработки 6

Класс приоритетов медленная логика PLC 30

Класс приоритетов обработки измеряемых величин 37

Код заказа MLFB 219

Комбинированные устройства 217

Компиляция
схемы CFC 29

Компиляция схемы CFC 29

Контейнер 217

Контейнер устройств 218

Контроль нулевой точки 186

Л

Левая граница 25

Введенная информация 25

Логический элемент Dynamic OR 89

Логический элемент NOR 94

Логический элемент OR 96

Логический элемент ST_AND 116

Логический элемент ST_OR 119

Логический элемент И 85

Логический элемент И НЕ 91

Логический элемент исключающего ИЛИ 99

М

Матрица ранжирования

Определение информации как входного сигнала 11

Определение информации как выходного сигнала 12

Переименование информации 32

Ранжирование информации на светодиод 33

Матрица связи 221

Модем 219

Модемы 219

Н

Набор уставок 29, 219

Назначение класса приоритетов 6

Нижний предел 189

Новая информация в матрице ранжирования 31

О

Обработка состояния
 CV_GET_STATUS 104
 DI_GET_STATUS 106
 DI_SET_STATUS 107
 DIST_DECODE 170
 MV_GET_STATUS 109
 MV_SET_STATUS 110
 SI_GET_STATUS 112
 SI_SET_STATUS 89
 ST_AND 116
 ST_NOT 118
 ST_OR 119
 Обратная блокировка
 Принцип 68
 Обратная блокировка для защиты шин
 Вариант 1 69
 Вариант 2 69
 Внешнее короткое замыкание на присоединении 68
 Короткое замыкание на сборных шинах 69
 Предпосылки схемы CFC 70
 Реализация схемы CFC 70
 Фрагмент схемы CFC 71
 Обратная связь сигнала 157
 Объекты 219
 Определение класса приоритетов 19
 Открытие
 схемы CFC 17
 Открытие схемы CFC 17
 Отмена команды 147
 Отображение
 схем CFC 15
 Отображение панелей инструментов 5
 Отображение панелей инструментов. 5
 Отображение строки состояния 5
 Отображенная информация
 Диалоговое окно выбора левой границы 24
 Диалоговое окно выбора правой границы 27
 Очистка памяти 220

П

Папка 218
 Параметрирование 220
 Параметрирование блока 35
 Переименование информации 32

Переименование схемы CFC 16
 Переключение отображения в вид бланка 17
 Перемещение
 блока 23
 Перемещение блока 23
 Подавление незначащих нулей 194
 Подсчёт циклов переключения
 Принцип работы 72
 Фрагмент схемы CFC 72
 Пользователь 221
 Последовательность переключений 149
 Правая граница 28
 Введенная информация 28
 Преобразование булевой логики в двухпозиционное сообщение 160
 Преобразование булевой логики в команду 138
 Преобразование булевой логики во внутреннее однопозиционное сообщение 143
 Преобразование двухпозиционного сообщения в булеву логику 165
 Пример
 Включение опции переключения набора уставок посредством двоичного входа 51
 Запуск двигателя кнопкой F1 50
 Имитация мигания в программе CFC 64
 Использование информации для связи 56
 Компиляция схемы CFC 58, 61, 67
 Контроль времени запуска и переключение 59
 Контроль фазных токов в программе CFC 56
 Непрерывный сигнал как сигнал запуска мигания 67
 Обратная блокировка 68
 Определение активного набора уставок 52
 Переключение набора уставок 52
 Подготовка управления светодиодами через программу CFC 64
 Подготовка функциональных кнопок для обработки 62
 Подсчёт циклов переключения 72
 Распределение опции переключения на различные классы приоритетов 54
 связь между разными классами приоритетов 55

Счетчик переключений 72
 Частота мигания 64
 Пример блокировки 41
 Пример быстрой логики Fast PLC
 Пример 14
 Пример использования
 Блок BOOL_TO_CO 142
 Блок CMD_CHAIN 154
 Блок DI_TO_BOOL 168
 Блок LOWER_SETPPOINT 191
 Блок TIMER 210
 Пример медленной логики Slow PLC 30
 Пример обработки измеряемых величин
 37
 Пример применения
 Блок D_FF 82
 Приоритет обработки
 Класс приоритетов 6
 Проверка двухпозиционного сообщения
 165
 Проекты 220
 Простой таймер 211
 Профиль модема 219

Р

Разбивка и соединение информационных
 элементов 9
 Размещение блока 21
 Ранжирование
 информации 11
 Ранжирование информации на светодиод
 33
 Рекомендации по программированию 9
 Решение для воссоединения
 информационных элементов 9
 Решение для разбиения
 информационного элемента 9

С

Свойства объекта 219
 Связь
 IEC адрес 218
 Адрес пользователя 221
 Интерфейс SCADA 220
 Модем 219
 Модемы 219
 Сервисный интерфейс 220
 Телефонная книга 219
 Указатель связи CR 217

Связь выходных сигналов в разных схемах
 26
 Сервисный интерфейс 220
 Сложение 79
 Соединение блоков 39
 Соединение входного сигнала 24
 Соединение входов / выходов блока 39
 Соединение выходных сигналов 26
 Создание двухпозиционного сообщения
 162
 Списочный вид 218
 Сравнение измеряемых величин 183
 Стандартная формула для CFC 9
 Строка инициализации 218
 Строка состояния 5
 Схема CFC
 Изменение последовательности
 выполнения блока 23
 Отображение схем CFC 15
 схема CFC
 Вставка 15
 Счетчик 196
 Счетчик переключений
 Принцип работы 72
 Фрагмент схемы CFC 72

Т

Таймер 206
 Работа обычного таймера 208
 Телефонная книга 219
 Тип данных
 BOOL 74
 DINT 74
 INT 74
 REAL 74
 STRUCT 74
 UINT 74
 WORD 74
 Тип данных DINT
 Информация о состоянии 75
 Тип данных REAL
 Информация о состоянии 75
 Типы данных в DIGSI CFC 74
 Типы объектов
 Вариант SIPROTEC 4 220
 Контейнер устройств 218
 Модем 219
 Модемы 219
 Папка 218
 Проект 220
 Телефонная книга 219

Устройство V3/V2 221
Топологический вид 220

У

Увеличение количества входов блока 44
Указатель связи CR 217
Умножение 81
 MUL 81
Универсальный таймер 206
Устройства защиты 220
Устройства присоединения 218
Устройства управления присоединением
 217
Устройство SIPROTEC 4 220
Устройство V3/V2 221

Ф

Формирование двухпозиционного
 сообщения 160, 162
Формирование команды на переключение
 138

W

WORD
 Тип данных 74

Кому:
Siemens AG
PTD EA D SC 22
Postfach 4806
D-90026 Nuremberg

От:
ФИО:
Должность:
Предприятие:
 Отдел:
 Улица:
 Город:
Телефон:
 Факс:

Пожалуйста, отметьте галочкой вашу отрасль промышленности:

- Автоматизация
- Горная промышленность
- Химическая промышленность
- Производство электроэнергии
- Распределение энергии, управление энергосистемами
- Газо-и водоснабжение, санитарно-технический комплекс
- Строительная техника, энергосистемы, системы воздушного охлаждения
- Тяжелое машиностроение, подъемно-транспортное оборудование
- Системы магистральных трубопроводов
- Кораблестроение, судоходство
- Экологическая техника
- Дорожное дело
- Другие.....

Примечания / предложения

Ваши примечания и предложения помогут нам усовершенствовать нашу документацию. Пожалуйста, заполните данную анкету и пошлите ее на фирму Siemens по почте или по факсу (номер факса: ++49 - 911 / 433-8518).

Название руководства по эксплуатации:CFC

Заказной номер руководства:E50417-H1156-C098-A1

Пожалуйста, дайте вашу личную оценку данной документации, проставляя баллы от 1 = хорошо до 5 = плохо.

1. Соответствует ли содержание вашим требованиям?
2. Смогли ли вы быстро найти интересующую вас информацию?
3. Возникли ли у вас трудности с пониманием текста?
4. Соответствует ли глубина информации вашим требованиям?
5. Довольны ли вы качеством рисунков?

Если вы столкнулись с какой-либо конкретной проблемой, пожалуйста, опишите ее на отведенных для этого строках:

.....

.....

.....

.....

.....

.....

