

## **ЧАСТЬ 2**

# **ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ**

Руководство Пользователя

Модульная интегрированная SCADA КРУГ-2000™. Кругол™. Библиотека функций.

Технологические функции. Часть 2.

Руководство Пользователя/2-е изд.

© 1992-2020. ООО НПФ «КРУГ». Все права защищены.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотографирование, магнитную запись или иные средства копирования или сохранения информации, без письменного разрешения владельцев авторских прав.

Все упомянутые в данном издании товарные знаки и зарегистрированные товарные знаки принадлежат своим законным владельцам.

---

---

#### ООО НПФ «КРУГ»

440028, г. Пенза, ул. Титова 1

Тел. +7 (8412) 49-97-75, 49-72-24, 49-75-34, 49-94-14

E-mail: [support@krug2000.ru](mailto:support@krug2000.ru)

http:// [www.krug2000.ru](http://www.krug2000.ru)

## ОБ ЭТОЙ КНИГЕ

Библиотечные функции технологического языка КРУГОЛ обеспечивают эффективное решение многих задач автоматизации производства и учета ресурсов, таких как противоаварийные защиты и блокировки, управление задвижками, насосами и электрооборудованием, учет наработки оборудования, технический и коммерческий учет энергоносителей и тепловой энергии.

Данная книга содержит описание библиотеки функций КРУГОЛ. Описание структурировано по области применения функций и платформе среды исполнения КРУГОЛ.

Структура книги приведена в таблице 1.

Таблица 1. Классификация функций ИСР КРУГОЛ

Название книги	Название части	Содержание
Модульная интегрированная SCADA <b>КРУГ-2000™</b>  <b>КРУГОЛ™. БИБЛИОТЕКА ФУНКЦИЙ</b>  Обозначение документа: КР01.20111W-03.00-И2.3.1	<u>Часть 1</u>  <b>ЛОГИЧЕСКИЕ И МАТЕМАТИЧЕСКИЕ ФУНКЦИИ</b> Руководство Пользователя Обозначение документа: КР01.20111W-03.10-И2.3.2	1. Функции логические 2. Функции логического преобразования 3. Функции математические 4. Функции преобразований переменной 5. Функции сравнения переменных
	<u>Часть 2</u>  <b>ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ</b> Руководство Пользователя КР01.20111W-03.10-И2.3.3	6. Функции архивирования 7. Функции работы с датой и временем 8. Функции чтения значений из таблиц нелинейности 9. Функции работы с признаками сигнализации 10. Функции передачи паспортов переменных 11. Функции динамического преобразования 12. Функции-таймеры 13. Функции работы с печатными документами 14. Функции общесистемные 15. Функции коммутации данных 16. Протокол HART
	<u>Часть 3</u>  <b>ФУНКЦИИ УЧЕТА РЕСУРСОВ</b> Руководство Пользователя КР01.20111W-03.10-И2.3.4	17. Учет теплоресурсов (ГОСТ серии 8.563.1-3) 18. Учет природного газа и его компонентов (ГОСТ серии 8.563.1-3) 19. Учет теплоресурсов (ГОСТ серии 8.586.1-5) 20. Учет природного газа и его компонентов (ГОСТ серии 8.586.1-5) 21. Расчёт количества нефтепродуктов в резервуарах (ГОСТ серии 8.595)

Название книги	Название части	Содержание
	<u>Часть 4</u> <b>«УСТАРЕВШИЕ» ФУНКЦИИ</b> Руководство Пользователя КР01.20111W-03.10-И2.3.5	22. Устаревшие функции

В данной книге термины «SCADA КРУГ-2000», «Система КРУГ-2000» и «КРУГ-2000» – синонимы. Информация, содержащаяся в данной книге, не может гарантировать абсолютную точность и полноту приводимых сведений. Это связано с возможными человеческими или техническими ошибками, допущенными в процессе подготовки информации, а также с политикой совершенствования и развития SCADA КРУГ-2000.

НПФ «КРУГ» не несет ответственности за возможные ошибки, связанные с использованием данной информации.

Надеемся, что SCADA КРУГ-2000 позволит Вам успешно разрабатывать и эксплуатировать системы контроля и управления.

С уважением,  
НПФ «КРУГ».

**СОДЕРЖАНИЕ**

Стр.

<b>ОБ ЭТОЙ КНИГЕ</b>	<b>3</b>
<b>ВВЕДЕНИЕ</b>	<b>1</b>
В.1. Инсталляция Библиотеки функций языка КРУГОЛ	1
В.1.1. Указания по инсталляции Библиотеки функций языка КРУГОЛ	1
В.1.2. Порядок инсталляции	1
В.2. «Перегрузка» вызова функций	7
В.3. Функции с переменным числом параметров и «устаревшие» функции	9
В.4. Привязка функций к платформам	10
<b>6 АРХИВИРОВАНИЕ</b>	<b>6-1</b>
6.1 Ускорение доступа к трендам	6-1
6.2 ЗаписьВ, WriteR32	6-3
6.3 ЗаписьЛ, WriteB	6-4
6.4 ЗаписьЦ, Writel16	6-5
6.5 ЧтениеВ, ReadR32	6-6
6.6 ЧтениеЛ, ReadB	6-7
6.7 ЧтениеЦ, Readl16	6-8
6.8 ЗпТр, WTrend	6-9
6.9 ПгТр1, PTrend1	6-10
6.10 ПгТр2, PTrend2	6-11
6.11 ЧтТр, RTrend	6-12
6.12 ЧтТрТ, RTrendC	6-14
6.13 Статистика, Statistics	6-15
6.14 ЗпСбТр, WETrend	6-25
6.15 ЗпСбТр2, WETrend2	6-28
<b>7 ДАТА И ВРЕМЯ</b>	<b>7-1</b>
7.1 СДата, SDate	7-1
7.2 СВремя, STime	7-2
<b>8 ЧТЕНИЕ ИЗ ТАБЛИЦ НЕЛИНЕЙНОСТИ</b>	<b>8-1</b>
8.1 НЛн2, N	8-1
8.2 НЛн3, NT	8-2
8.3 НЛн3Инт, NInt	8-3
<b>9 РАБОТА С ПРИЗНАКАМИ СИГНАЛИЗАЦИИ</b>	<b>9-1</b>
9.1 САГ	9-1
9.2 СДГ	9-2
9.3 СЗД	9-3
9.4 СИМ	9-4

**СОДЕРЖАНИЕ**

9.5	СПГ _____	Стр. 9-5
9.6	ССК _____	9-6
<b>10</b>	<b>ПЕРЕДАЧА ПАСПОРТА ПЕРЕМЕННОЙ _____</b>	<b>10-1</b>
10.1	Ограничения на использование функций посХХ _____	10-1
10.2	ПосАВ, SendАО _____	10-4
10.3	ПосВА, SendАІ _____	10-5
10.4	ПосВД, SendDІ _____	10-6
10.5	ПосДВ, SendDO _____	10-7
10.6	ПосПР, SendVar _____	10-8
10.7	ПосРВ, SendНІ _____	10-9
<b>11</b>	<b>ДИНАМИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ _____</b>	<b>11-1</b>
11.1	Зап, Delay _____	11-1
11.2	Зап_ц16, Delay_l16 _____	11-2
11.3	Зап_ц32, Delay_l32 _____	11-3
11.4	Зап_в32, Delay_r32 _____	11-4
11.5	Зап_в64, Delay_r64 _____	11-5
11.6	Диф, Dif _____	11-6
11.7	Диф_в32, Dif_r32 _____	11-7
11.8	Диф_в64, Dif_r64 _____	11-8
11.9	Фил, Fil _____	11-9
11.10	Фил_в32, Fil_r32 _____	11-10
11.11	Фил_в64, Fil_r64 _____	11-11
11.12	Инт, Int _____	11-12
11.13	Инт_в32, Int_r32 _____	11-14
11.14	Инт_в64, Int_r64, _____	11-16
<b>12</b>	<b>ТАЙМЕРЫ _____</b>	<b>12-1</b>
12.1	ТаймМ, ТМ _____	12-1
12.2	ТаймС, TS _____	12-2
12.3	ТаймЧ, TH _____	12-3
12.4	ТБ, TOff _____	12-4
12.5	ТЗ, TOn _____	12-5
12.6	ТИЗ, TP _____	12-7
12.7	ТМИ, TMI _____	12-9
12.8	ТМП, TMP _____	12-11
12.9	ТСИ, TSI _____	12-13
12.10	ТСП, TSP _____	12-15

**СОДЕРЖАНИЕ**

	Стр.
12.11 ТЧИ, THI _____	12-17
12.12 ТЧП, THP _____	12-19
<b>13 ПЕЧАТНЫЕ ДОКУМЕНТЫ _____</b>	<b>13-1</b>
13.1 ППД, PDoc _____	13-1
13.2 ППД1, PDoc1 _____	13-2
13.3 ПчПД, PrDoc _____	13-3
13.4 ПчПД1, PrDoc1 _____	13-4
13.5 ППД_ФВ, PDoc_FT _____	13-5
<b>14 ОБЩЕСИСТЕМНЫЕ _____</b>	<b>14-1</b>
14.1 AVR_C2 _____	14-1
14.2 AVR_U2 _____	14-3
14.3 BUD_2 _____	14-5
14.4 БУД_3, BUD_3 _____	14-9
14.5 ВУД_4, BUD_4 _____	14-27
14.6 BUZ_3K _____	14-44
14.7 ФЗадв, Fzadv _____	14-49
14.8 ФЗадв2, FZadv2 _____	14-52
14.9 Сообщение, Message _____	14-58
14.10 Сообщение2, Message2 _____	14-59
14.11 Сообщение3, Message3 _____	14-61
14.12 Пауза, Pause _____	14-62
14.13 ПрЗвук, PlaySound _____	14-63
14.14 ПрЗвукФайл, PlaySoundFile _____	14-65
14.15 ЧтБ_ц16, RBC_i16 _____	14-66
14.16 ЧтБ_ц32, RBC_i32 _____	14-67
14.17 СервСтат, ServState _____	14-68
14.18 ЗапБ_ц16, WBC_i16 _____	14-69
14.19 ЗапБ_ц32, WBC_i32 _____	14-70
14.20 ВНО1, DOT1 _____	14-71
14.21 Вызов, Call _____	14-74
14.22 Квит, Confirm _____	14-75
14.23 НайтиРВ, FindHI _____	14-76
14.24 ОбрД, ProcD _____	14-77
14.25 СММ, ResMM _____	14-84
14.26 ССч, ResC _____	14-85

**СОДЕРЖАНИЕ**

	Стр.
14.27 Счетчик, Counter _____	14-86
14.28 ТПУ2, TPU2 _____	14-87
14.29 Филд, FiID _____	14-89
14.30 ЧтБ, RBC _____	14-90
14.31 ЗапБ, WBC _____	14-91
14.32 ЧтПарам, RParam _____	14-92
<b>15 ФУНКЦИИ КОММУТАЦИИ ДАННЫХ _____</b>	<b>15-1</b>
15.1 DMUL_D8 _____	15-1
15.2 MUL_D8 _____	15-3
15.3 MUL_A8 _____	15-5
15.4 Мул_л, MUX_b _____	15-7
15.5 Мул_ц16, MUX_i16 _____	15-8
15.6 Мул_ц32, MUX_i32 _____	15-9
15.7 Мул_в32, MUX_r32 _____	15-10
15.8 Мул_в64, MUX_r64 _____	15-11
15.9 ДМул8_л, DMUX8_b _____	15-12
15.10 ДМул8_ц16, DMUX8_i16 _____	15-14
15.11 ДМул8_ц32, DMUX8_i32 _____	15-16
15.12 ДМул8_в32, DMUX8_r32 _____	15-18
15.13 ДМул8_в64, DMUX8_r64 _____	15-20
15.14 Выбор_л, Select_b _____	15-22
15.15 Выбор_ц16, Select_i16 _____	15-23
15.16 Выбор_ц32, Select_i32 _____	15-24
15.17 Выбор_в32, Select_r32 _____	15-25
15.18 Выбор_в64, Select_r64 _____	15-26
15.19 Шаг, STEP _____	15-27
15.20 КД, CD _____	15-31
15.21 КВ _____	15-33
15.22 КЛ _____	15-34
15.23 КЦ _____	15-35
<b>16 ПРОТОКОЛ HART _____</b>	<b>16-36</b>
16.1 Hart0_Run, Харт0_Запуск _____	16-36
16.2 Hart0_Result, Харт0_Рез _____	16-38
16.3 HartAdr_Run, ХартАдр_Запуск _____	16-41
16.4 HartAdr_Result, ХартАдр_Рез _____	16-43
16.5 Hart1_Run, Харт1_Запуск _____	16-45



**СОДЕРЖАНИЕ**

	Стр.
16.6 Hart1_Result, Харт1_Рез_____	16-47
16.7 Hart2_Run, Харт2_Запуск_____	16-49
16.8 Hart2_Result, Харт2_Рез_____	16-51
16.9 Hart3_Run, Харт3_Запуск_____	16-53
16.10 Hart3_Result, Харт3_Рез_____	16-55
16.11 Коды ошибок HART_____	16-57
16.12 Пример использования функций опроса датчиков по протоколу HART _____	16-58



## **ВВЕДЕНИЕ**

### **В.1. Установка Библиотеки функций языка КРУГОЛ**

#### **В.1.1 Указания по установке Библиотеки функций языка КРУГОЛ**

- 1 Библиотека функций языка КРУГОЛ (далее **Библиотека**) поставляется вместе со SCADA КРУГ-2000 и для полнофункциональной работы требует **электронный ключ**.
- 2 Для установки Библиотеки Пользователь должен иметь права администратора.
- 3 Перед установкой Библиотеки закройте все работающие программы.
- 4 Если Вы устанавливали предыдущую версию Библиотеки, то удалите ее.
- 5 Если в процессе установки Вы хотите прервать установку, нажмите кнопку «**Отмена**». В появившемся окне инсталлятор спросит Вас, действительно ли Вы хотите прервать установку. Вы можете нажать кнопку «**Да**» и выйти из инсталлятора или нажать кнопку «**Нет**» и вернуться к процессу установки.
- 6 В окнах инсталлятора может присутствовать кнопка «**<<Назад**», при нажатии на которую Вы можете вернуться в предыдущее окно. Данная возможность может пригодиться, если Вы захотите изменить настройки, выбранные в предыдущих окнах.

**Примечание :** Для корректной работы Библиотеки функций и всех её компонентов необходимо, чтобы её установка выполнялась всегда после завершения установки интегрированной среды разработки КРУГОЛ (ИСР). Также если выполнена переустановка ИСР, то и необходимо переустановить (удалить ранее установленную версию Библиотеки и установить новую) Библиотеку функций.



#### **Внимание!**

Данная версия Библиотеки функций КРУГОЛ предназначена для установки и совместной работы с интегрированной средой разработки КРУГОЛ версии 2.4 и выше и SCADA КРУГ-2000 версии 4.x.

Установка Библиотеки функций совместно с работающим ядром КРУГОЛ не допускается.

#### **В.1.2. Порядок установки**

##### **Шаг 1. Начало установки**



#### **Внимание!**

Для работы инсталлятора необходим установленный на компьютер Microsoft Windows Installer версии 3.0 и выше.

Если на компьютере включен контроль учетных записей (UAC), необходимо запускать инсталлятор Библиотеки функции от имени администратора.

Для установки Библиотеки функций языка КРУГОЛ запустите на выполнение инсталлятор (рисунк В.1) и следуйте его указаниям.

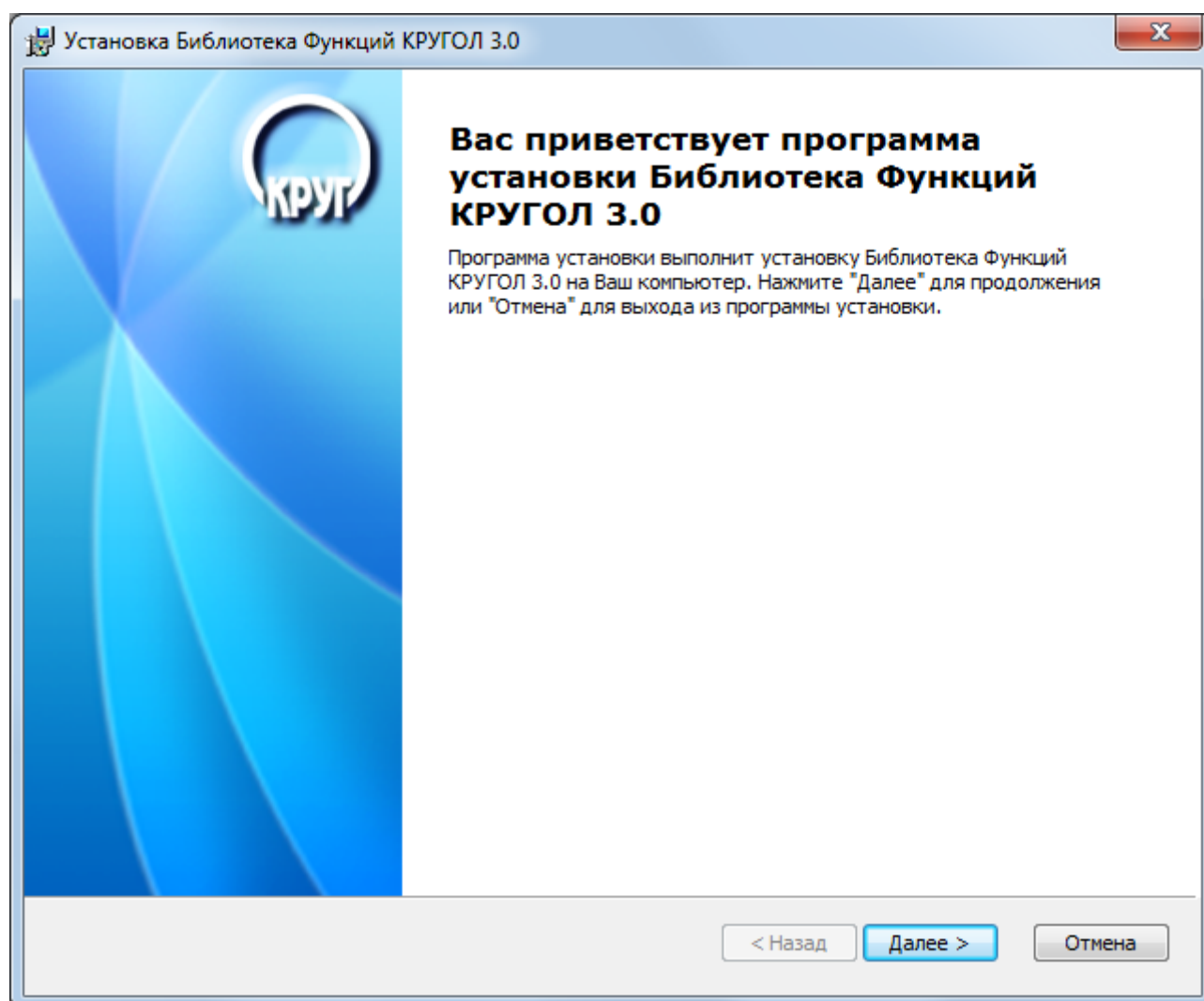


Рисунок В.1 - Окно инсталлятора Библиотеки

### Шаг 2. Лицензионное соглашение

Для продолжения установки внимательно прочитайте лицензионное соглашение. Если Вы согласны с условиями соглашения, нажмите кнопку «**Я согласен**» (рисунок В.2).

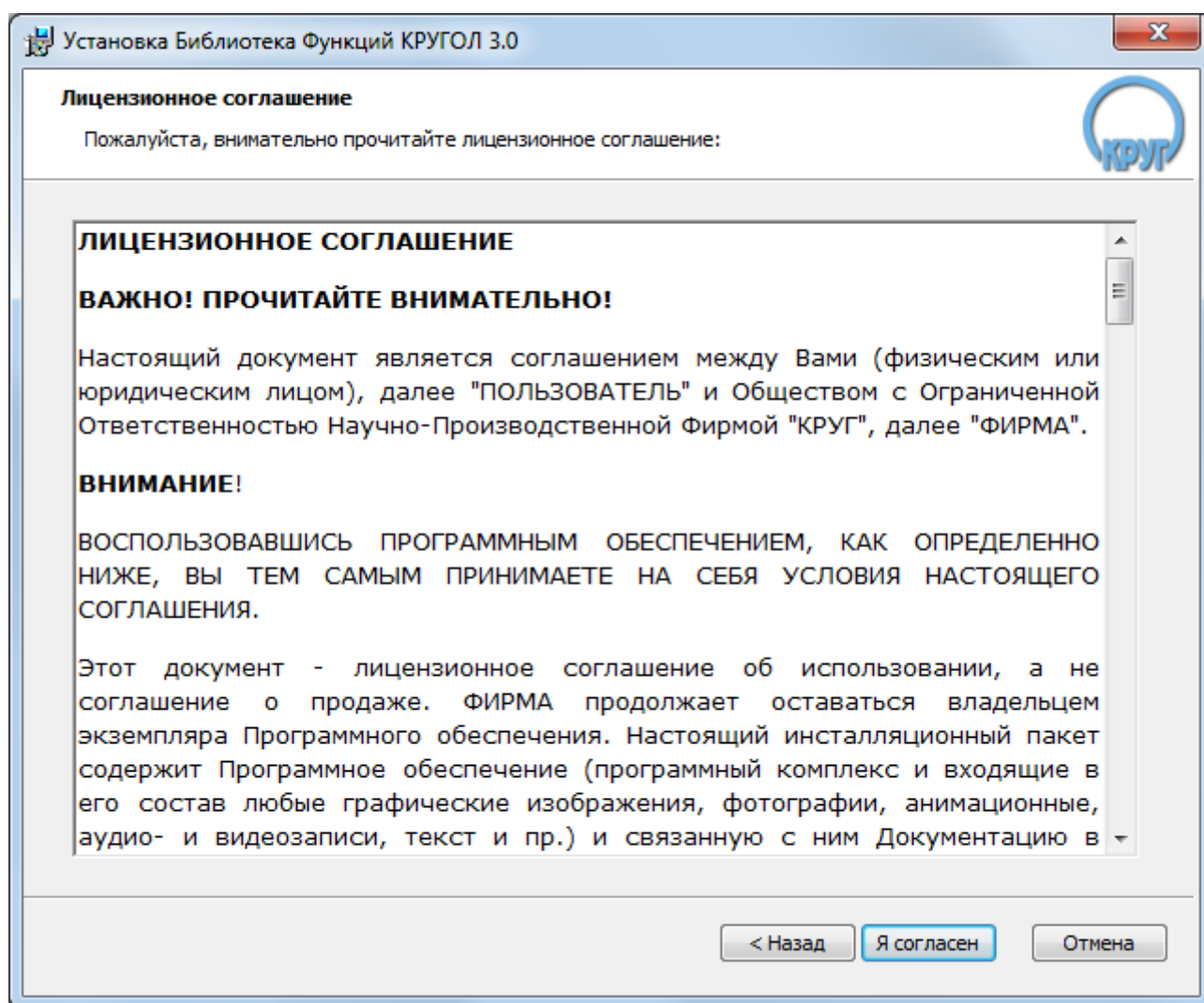


Рисунок В.2 – Окно «Лицензионное соглашение»

**Шаг 4. Выбор пути установки и устанавливаемых компонентов**

В окне «**Выбор пути установки**» указан путь установки по умолчанию (рисунок В.3).

Для продолжения нажмите на кнопку «**Далее>>**».

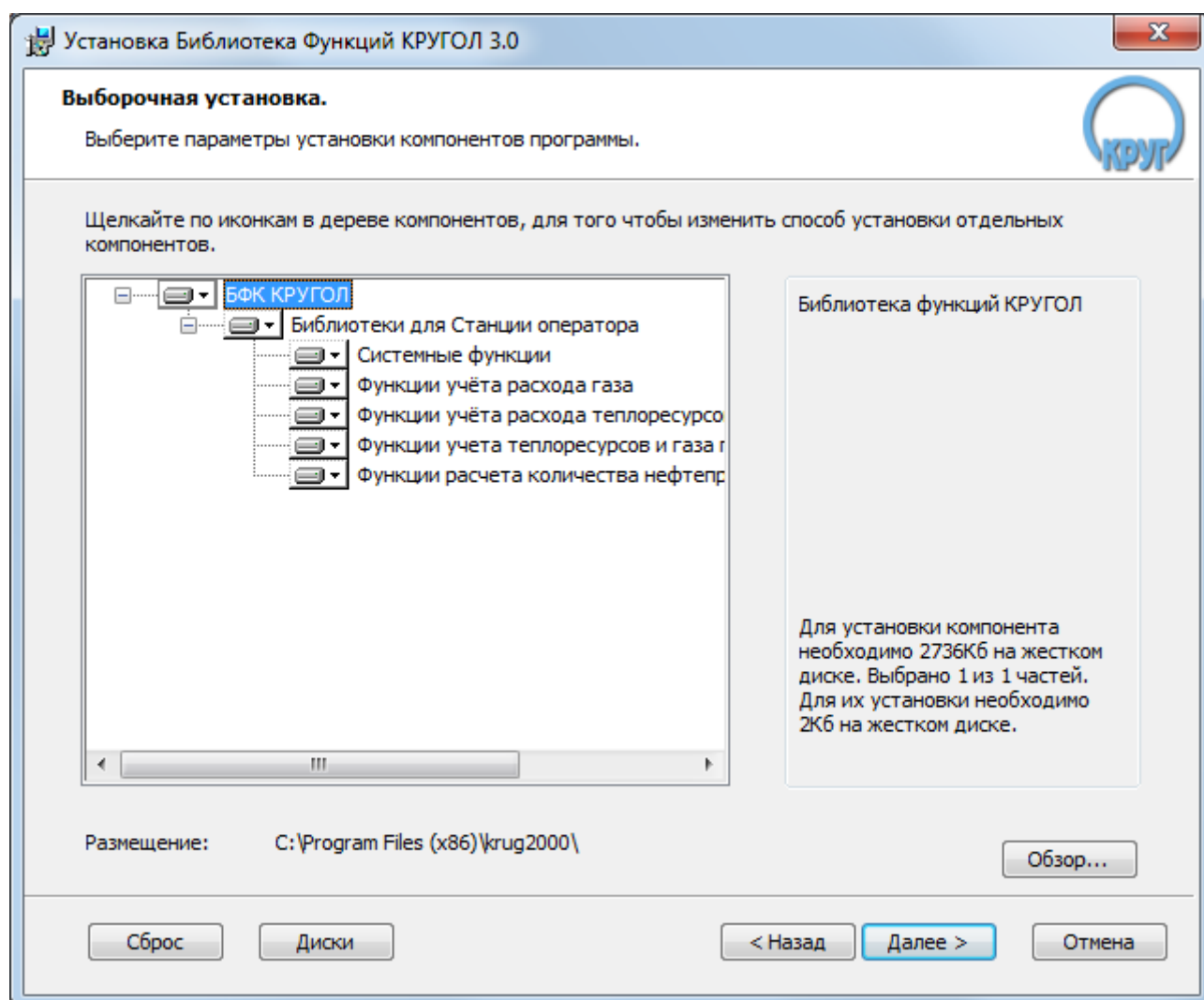


Рисунок В.3 - Окно выбора пути и компонентов установки

При выборе одного из компонентов для установки в правой части диалога отображается описание и назначение выбранного компонента.

Описание и назначение компонентов приведено в таблице В.1.

Таблица В.1

Наименование компонента	Описание
<b>Библиотеки для Станции оператора</b>	Библиотеки функций языка КРУГОЛ для выполнения на Станции оператора. Используются так же при отладке ПрП в ИСР КРУГОЛ и для трансляции ПрП под платформы СРВК.
Системные функции	Системные функции для Станции оператора
Функции учёта расхода газа	Функции учёта расхода газа для Станции оператора
Функции учёта расхода теплоресурсов	Функции учёта расхода теплоресурсов для Станции оператора
Функции учета теплоресурсов и газа по ГОСТ серии 8.586.X-2005	Функции учета теплоресурсов и газа по ГОСТ серии 8.586.X-2005 для Станции оператора
Функции расчёта количества нефтепродуктов	Функции расчёта количества нефтепродуктов в резервуарах

#### Шаг 6. Подтверждение готовности к установке

После выбора компонент для продолжения нажмите на кнопку «**Далее>>**» и следуйте указаниям инсталлятора (рисунок В.4).

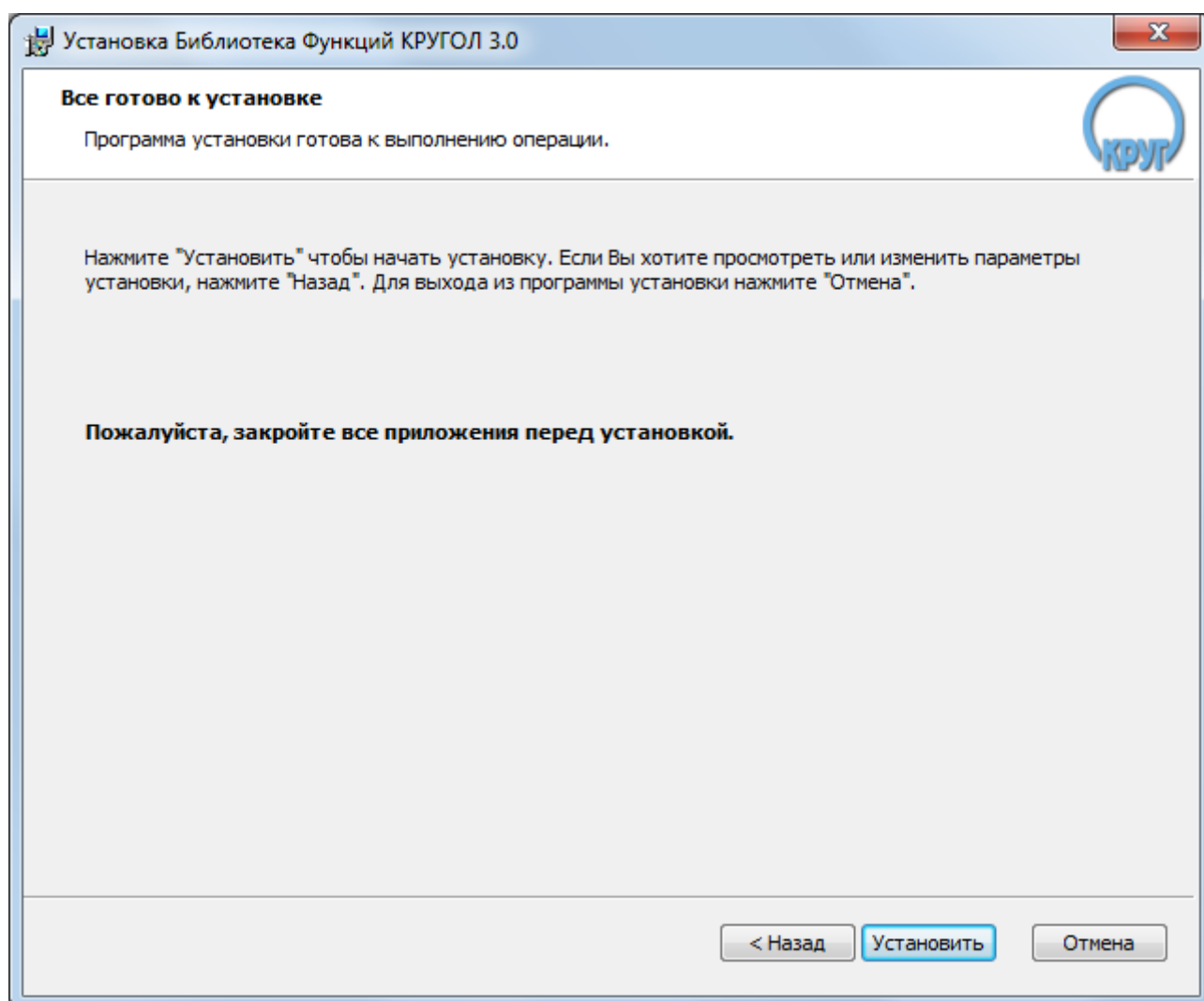


Рисунок В.4 – Всё готово для установки приложения

### Шаг 7. Завершение установки

После нажатия на кнопку «**Готово**» (рисунок В.5) установка завершается.



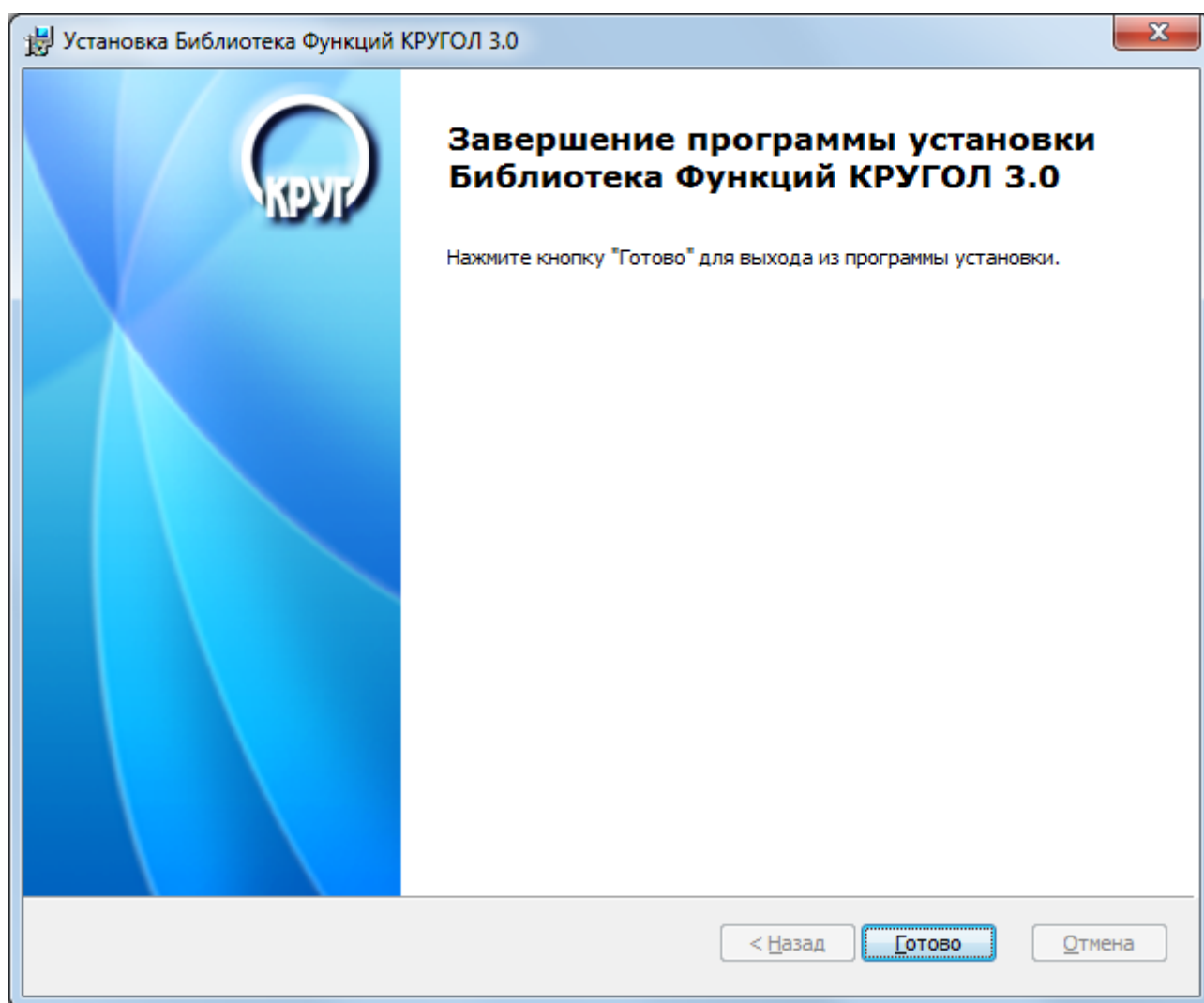


Рисунок В.5 - Завершение установки Библиотеки

## В.2. «Перегрузка» вызова функций

Начиная с Библиотеки версии 2.0, в языке КРУГОЛ появилось множество функций, которые реализуют одну и ту же операцию, но с различными типами данных. Например: Add\_i8 – сложение целых 8битовых чисел, Add\_i16 – сложение целых 16- битовых чисел, Add\_r32 – сложение вещественных 32- битовых чисел и т. д.

Функцию, которая реализует оператор вычисления выражений для конкретных типов данных, будем называть *типизированной* (например, Add\_i8).

Для соблюдения уникальности имён функций в языке КРУГОЛ принято соглашение, что имена типизированных функции будут иметь суффикс, соответствующий типу данных, с которым работает функция. Суффиксу соответствует синтаксис:

\_<тип данных> <размер данных>

Таблица соответствия типов данных и суффиксов приведена ниже.

Тип данных	Суффикс1	Суффикс2
Вещ32	_r32	_в32
Вещ64	_r64	_в64
Цел8	_i8	_ц8
Цел16	_i16	_ц16
Цел32	_i32	_ц32

## ВНИМАНИЕ!!!

**Использование типизированных функций разрешено для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.**

Для удобства пользователя в интегрированной среде языка КРУГОЛ разработан механизм «перегрузки» вызова типизированных функций. Суть этого механизма заключается в том, что Пользователь может для вызова оператора вычисления выражений указать *обобщенное имя* (без суффикса) функции и операнды требуемого типа. Например, для сложения целых чисел – **Add (5, 2)**, для сложения вещественных – **Add (2.5, 1.2)**.

В процессе трансляции программы на основании имени функции, типов входных параметров и их количества транслятор решает, какую функцию конкретно вызвать, и подставляет в исполняемый код вызов соответствующей типизированной функции.

В случае, если транслятор не сможет найти соответствующую типизированную функцию, будет выдано сообщение об ошибке и исполняемый код не будет сформирован.

## ВНИМАНИЕ!!!

**Механизм перегрузки вызова работает только для типизированных функций, которые имеют:**

- **Одинаковые обобщенные имена**
- **Одинаковое количество входных параметров**
- **Одинаковый тип входных параметров.**

Примеры вызова функции на языке СТ:

пц1 = Add(1,1) :с использованием механизма перегрузки вызова  
 пв1 = Слож(1.1, 2.2) :с использованием механизма перегрузки вызова  
 пц1 = Add\_ц16(1,1) :без использованием механизма перегрузки вызова  
 пв1 = Слож\_в32(1.1, 2.2) :без использованием механизма перегрузки вызова

В языке ФБД входные контакты «перегружаемой» функции будут отображаться, как контакты, отмеченные звездочкой – контакты неопределённого типа (рисунок В.6):

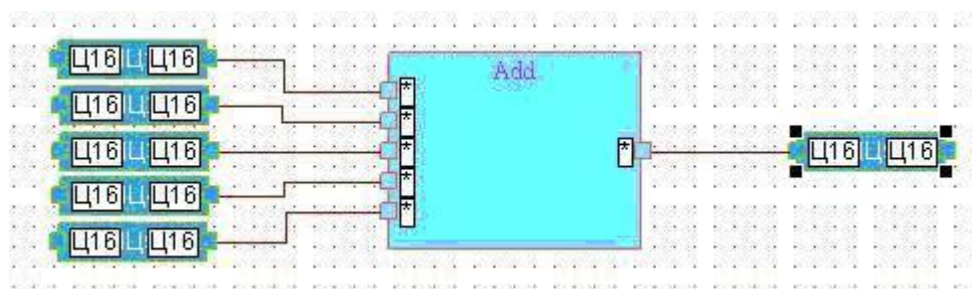


Рисунок В.6 – Пример вызова «перегружаемой» функции на языке ФБД

### В.3. Функции с переменным числом параметров и «устаревшие» функции

Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, некоторые функции библиотеки получили возможность вызова с переменным числом параметров (входов). Например, логическая функция «И» может вызываться с 2, 3, 4 и другим количеством входов. Синтаксис вызова функции следующий:

$$Y = И(X1, X2[, X3, \dots, Xn])$$

Для платформ СРВК, ниже версии 8.0, и среды исполнения КРУГОЛ, ниже версии 2.2, такой возможности нет. Для выполнения вычисления используются функции с фиксированным числом параметров. Например, «И2», «И3» и другие.

Для сохранения приемственности с проектами ранних версий ИСР КРУГОЛ для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, поддерживает выполнение функций с фиксированным числом параметров, но «считает их устаревшими».

Поэтому при выборе для разработки программы платформы СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, функции с фиксированным числом параметром будут располагаться в разделе библиотеки «Устаревшие функции» (рисунок В.7).

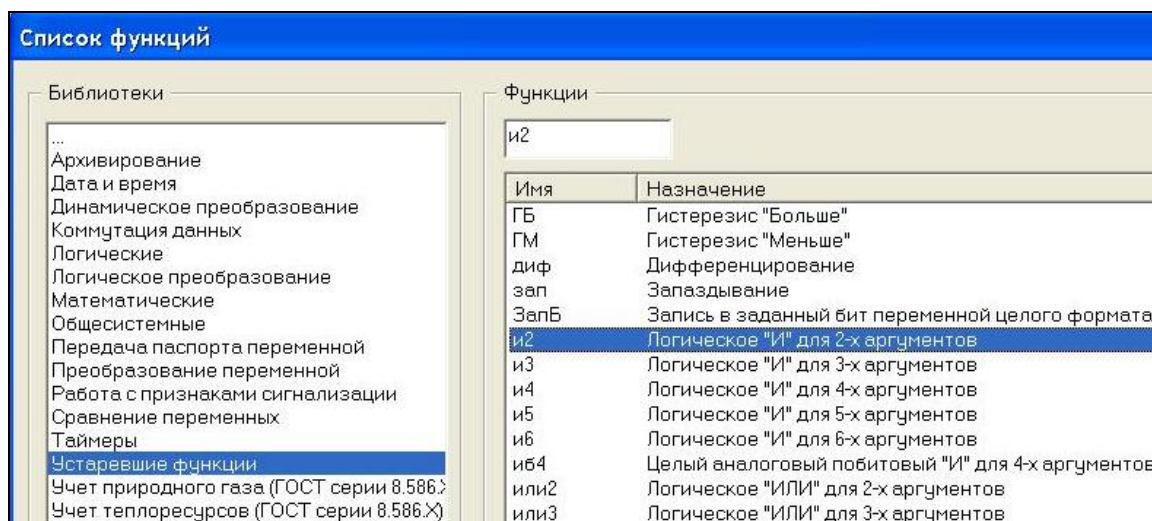


Рисунок В.7 – Раздел библиотеки «Устаревшие функции»

При выборе для разработки программы платформы СРВК, ниже версии 8.0, и среды исполнения КРУГОЛ, ниже версии 2.2, функции с фиксированным числом параметром будут располагаться в разделе библиотеки, соответствующему назначению функции. Например, функция «И2» будет располагаться в разделе «Логические» (рисунок В.8).

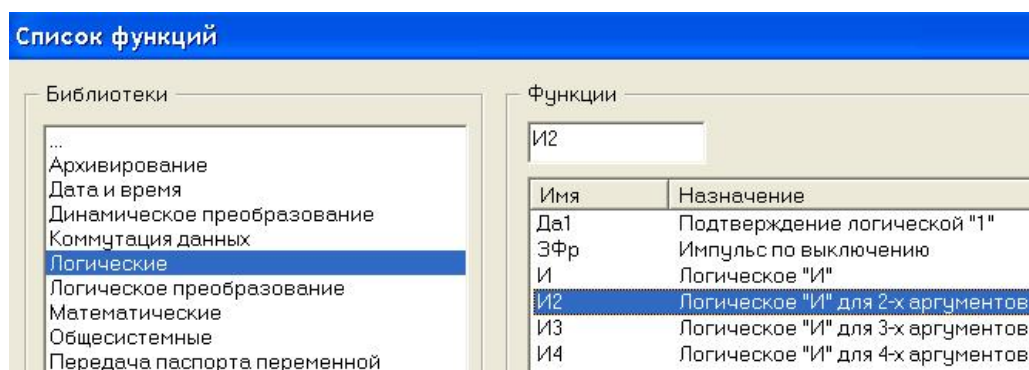


Рисунок В.8 – Раздел библиотеки «Логические»



## 6 АРХИВИРОВАНИЕ

### 6.1 Ускорение доступа к трендам

#### Назначение

Ускоряет чтение значений из трендов.

#### Область применения

Функции ускорения доступа к трендам работают только при чтении из трендов. Ускорить доступ к трендам можно только при обработке группы точек из одного пера. Например, при обработке группы точек при формировании суточной или сменной ведомости.

Рассмотрим на примерах случаи, в которых функции ускорения доступа к трендам могут дать существенный выигрыш в скорости.

**Пример 1.** Обработка группы точек одного пера.

- : Самописец 1 – часовой самописец.
- : Перо 1 – по этому перу требуется провести расчет
- : Требуется обработать точки за истекшие сутки.

**Процедура** пример\_1

**Начало**

```

: Подготавливаем тренды к считыванию
: Читаем точки за предыдущие сутки
ПгТр1(1,1,1,1)

пц1=1
пв1=0
Для к1[1...24]
{
    Если пц1=1
    {
        пв1=ЧтТр(1,1,1,пц1):Читаем первую точку в прошедших сутках
    }
    иначе
    {
        пв4=ЧтТр(1,1,1,пц1)
        пв1=пв1+пв4      : Суммируем точки
    }
    пц1=пц1+1
}

```

**Выход**

**Конец**

В этом примере перед проведением расчета точки пера считываются во внутреннюю область памяти функцией **ПгТр1**, после чего функцией **ЧтТр** считывают необходимые точки. Так как физически с сервера точки пера считываются за один запрос вместо

нескольких, то получаем следующие преимущества от использования функций ускорения доступа к трендам:

- Ускоряется доступ к трендам. Каждое обращение к серверу базы данных требует времени, в этом примере для каждого пера выполняется одно обращение на чтение всех точек за истекшие сутки вместо одного обращения для чтения каждой точки.
- Уменьшается нагрузка на сервер базы данных. Уменьшение количества запросов способствует уменьшению нагрузки на сервер базы данных.

**Пример 2.** Обработка точек нескольких перьев.

: Самописец 1 – часовой самописец  
: Каждое перо показывает расход газа. Есть несколько труб, по которым подается газ.  
: Необходимо подсчитать суммарный расход газа за истекший час.  
: Требуется, чтобы перья имели по возможности последовательно расположенные  
: номера, например 1,2,3,...

**Процедура** пример\_2  
**Начало**

: Читаем точки из первого самописца из перьев с 1 по 5 за текущие сутки  
**ПгТр2(1,1,5,0,0)**

**(пц1,пц2,пц3)=stime()**

**пв1=ЧтТр(1,1,0,пц1)** : Расход газа по первой трубе за прошедший час

**пв2=ЧтТр(1,2,0,пц1)** : Расход газа по второй трубе за прошедший час

**пв3=ЧтТр(1,3,0,пц1)** : Расход газа по третьей трубе за прошедший час

**пв4=ЧтТр(1,4,0,пц1)** : Расход газа по четвертой трубе за прошедший час

**пв5=ЧтТр(1,5,0,пц1)** : Расход газа по пятой трубе за прошедший час

**пв6=пв1+пв2+пв3+пв4+пв5:** Суммарный расход газа за прошедший час

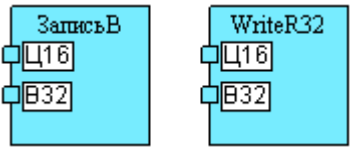
**Выход**  
**Конец**

В этом примере из каждого пера читается по одной точке. Благодаря тому, что функция **ПгТр2** умеет запрашивать точки сразу из нескольких перьев, все необходимые для расчета точки были считаны за одно обращение к серверу базы данных вместо пяти.

## 6.2 ЗаписьВ, WriteR32

### Назначение

Запись значения вещественной переменной в файл данных *fdat.dat*.

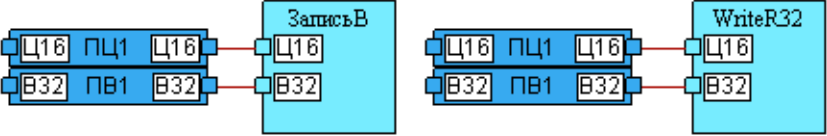
Отображение	
СТ:	ФБД:
ЗаписьВ(X1, X2)  WriteR32(X1,X2)	
Входные параметры: X1(ц16), X2(в32)	

### Логика работы функции

X1 - адрес (от 1 до 1024 для контроллера), в файле данных вещественного формата.

X2 - записываемое значение.

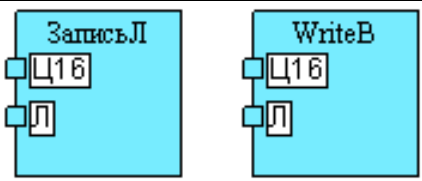
Файл с данными располагается в каталоге базы данных (для станции оператора) или в каталоге dat (для контроллера).

Пример вызова функции	
СТ:	ФБД:
ЗаписьВ(пц1,пв1)  WriteR32(пц1,пв1)	

### 6.3 ЗаписьЛ, WriteB

#### Назначение

Запись значения логической переменной в файл данных ldat.dat.

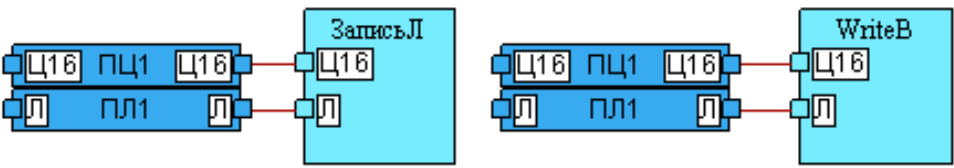
Отображение	
СТ:	ФБД:
ЗаписьЛ(X1,X2) WriteB(X1,X2)	
Входные параметры: X1(ц16), X2(л)	

#### Логика работы функции

X1 - адрес (от 1 до 1024 для контроллера), в файле данных логического формата.

X2 - записываемое значение.

Файл с данными располагается в каталоге базы данных (для станции оператора) или в каталоге dat (для контроллера).

Пример вызова функции	
СТ:	ФБД:
ЗаписьЛ(пц1,пл1) WriteB(пц1,пл1)	



## 6.4 ЗаписьЦ, WriteI16

### Назначение

Запись значения целой переменной в файл данных idat.dat

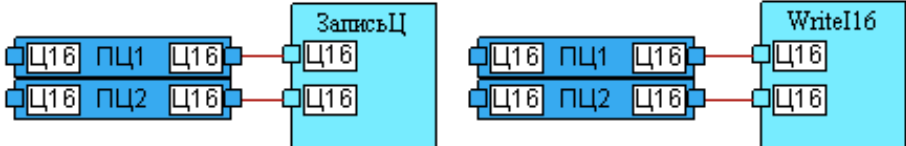
Отображение	
СТ:	ФБД:
ЗаписьЦ(X1, X2)	<div><div>ЗаписьЦ</div><div><div>Ц16</div><div>Ц16</div></div></div> <div><div>WriteI16</div><div><div>Ц16</div><div>Ц16</div></div></div>
WriteI16(X1,X2)	
Входные параметры: X1(ц16), X2(ц16)	

### Логика работы функции

X1 - адрес (от 1 до 1024 для контроллера), в файле данных целого формата.

X2 - записываемое значение.

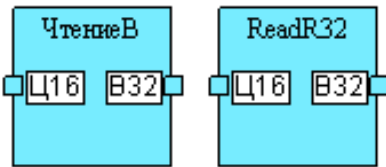
Файл с данными располагается в каталоге базы данных (для станции оператора) или в каталоге dat (для контроллера).

Пример вызова функции	
СТ:	ФБД:
ЗаписьЦ(пц1, пц2)	
WriteI16(пц1,пц2)	

## 6.5 ЧтениеВ, ReadR32

### Назначение


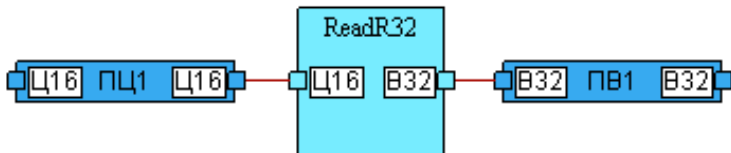
Чтение значения вещественной переменной из файла данных fdat.dat.

Отображение	
СТ:	ФБД:
$Y = \text{ЧтениеВ}(X)$ $Y = \text{ReadR32}(X)$	
Входные параметры: X(ц16) Выходные параметры: Y(в32)	

### Логика работы функции

Выходному параметру Y присваивается вещественное значение переменной, находящейся в файле данных вещественного формата по адресу (от 1 до 1024 для контроллера), заданному входным параметром X.

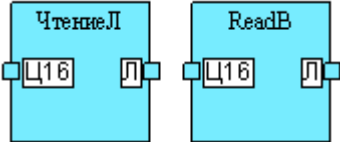
Файл с данными должен располагаться в каталоге базы данных (для станции оператора) или в каталоге dat (для контроллера).

Пример вызова функции	
СТ:	ФБД:
$\text{пв1} = \text{ЧтениеВ}(\text{пц1})$	
$\text{пв1} = \text{ReadR32}(\text{пц1})$	

## 6.6 ЧтениеЛ, ReadB

### Назначение

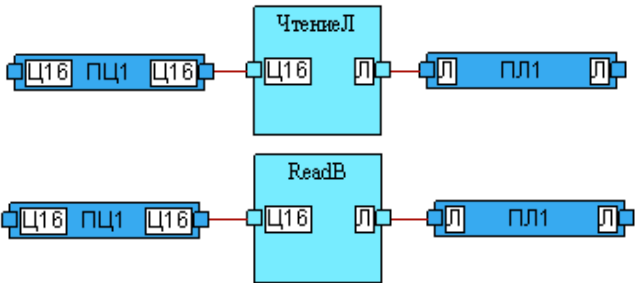
Чтение значения логической переменной из файла данных ldat.dat.

Отображение	
СТ:	ФБД:
$Y = \text{ЧтениеЛ}(X)$ $Y = \text{ReadB}(X)$	
Входные параметры: X(ц16) Выходные параметры: Y(л)	

### Логика работы функции

Выходному параметру Y присваивается логическое значение переменной, находящейся в файле данных логического формата по адресу (от 1 до 1024 для контроллера), заданному входным параметром X.

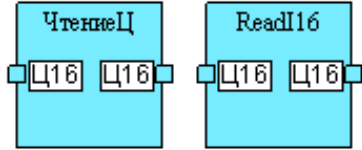
Файл с данными должен располагаться в каталоге базы данных (для станции оператора) или в каталоге dat (для контроллера).

Пример вызова функции	
СТ:	ФБД:
$\text{пл1} = \text{ЧтениеЛ}(\text{пц1})$ $\text{пл1} = \text{ReadB}(\text{пц1})$	

6.7 ЧтениеЦ, ReadI16

Назначение

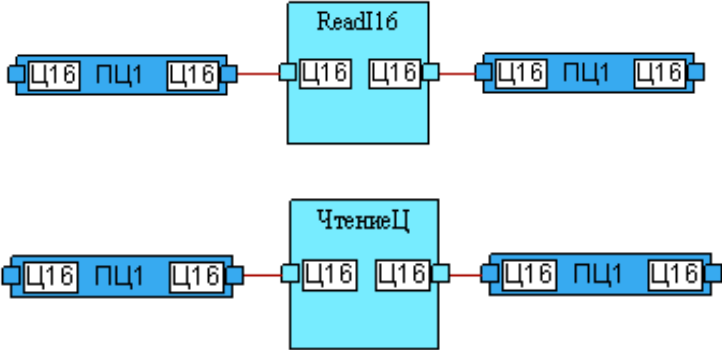
Чтение значения целой переменной из файла данных idat.dat

Отображение	
СТ:	ФБД:
$Y = \text{ЧтениеЦ}(X)$ $Y = \text{ReadI16}(X)$	
Входные параметры: X(ц16) Выходные параметры: Y(ц16)	

Логика работы функции

Выходному параметру Y присваивается целое значение переменной, находящейся в файле данных целого формата по адресу (от 1 до 1024 для контроллера), заданному входным параметром X.

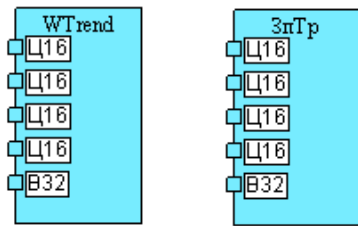
Файл с данными должен располагаться в каталоге базы данных (для станции оператора) или в каталоге dat (для контроллера).

Пример вызова функции	
СТ:	ФБД:
$\text{пц1} = \text{ЧтениеЦ}(\text{пц1})$ $\text{пц1} = \text{ReadI16}(\text{пц1})$	

## 6.8 3пТр, WTrend

### Назначение

Запись тренда.

Отображение	
СТ:	ФБД:
3пТр(X1, X2, X3, X4, X5) WTrend(X1, X2, X3, X4, X5)	
Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16), X5(в32)	

### Описание

Для платформ СРВК любых версий данная функция не применяется.

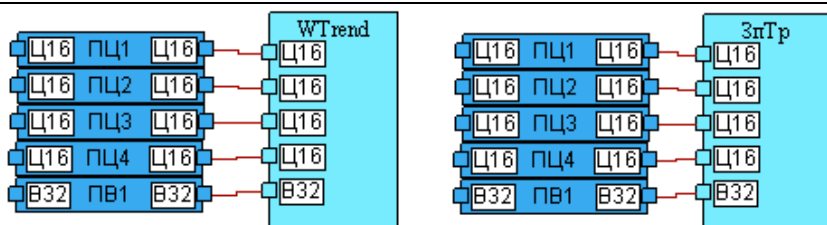
### Логика работы функции

X1 – порядковый номер самописца;  
 X2 - порядковый номер пера в указанном самописце;  
 X3 – номер интервала хранения от текущего назад;  
 X4 – порядковый номер точки записи в интервале хранения;  
 X5 – записываемое значение

Для самописцев с базовым интервалом записи “секунда” интервалом хранения является “минута”, с интервалом записи “минута” интервал хранения “час”, с интервалом записи “час” интервалом хранения является “сутки”, с интервалом записи “сутки” интервалом хранения является “месяц”.

Если X1 =3 (пусть, это будет, например, часовой самописец), X2=1, X3 =2, X4 =4, X5=0.345, то точке, взятой из четвертого часа от начала суток два дня назад из тренда часового самописца 3 пера 1, присваивается значение 0.345.



Если порядковый номер самописца соответствует суточному самописцу, то значение X4 будет соответствовать: «0» - 1-й день месяца, «1» - 2-й день месяца и т.д. Например, 3пТр(1,1,0,4,2.34) означает, что точке, взятой из пятых суток текущего месяца из суточного тренда 1 пера 1, будет присвоено значение 2.34. Или 3пТр(1,1,1,4,56) означает, что точке, взятой из пятых суток прошедшего месяца из суточного тренда 1 пера 1, будет присвоено значение 56.

Пример вызова функции	
СТ:	ФБД:
3пТр(пц1, пц2, пц3, пц4, пв1) WTrend(пц1, пц2, пц3, пц4, пв1)	

## 6.9 ПгТр1, PTrend1

### Назначение

Подготовка тренда.

Отображение		
СТ:	ФБД:	
ПгТр1 (X1, X2, X3, X4) PTrend1(X1, X2, X3, X4)		
Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16)		

### Описание

Для платформ СРВК любых версий данная функция не применяется.

### Логика работы

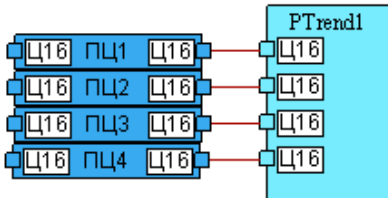
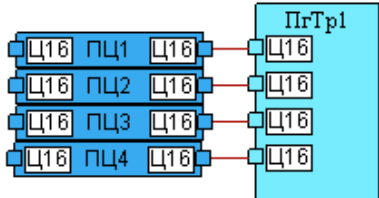
X1 – порядковый номер самописца.

X2 – порядковый номер пера в этом самописце.

X3 - номер интервала хранения от текущего назад, с которого начать считывание. Для самописцев с базовым интервалом записи “секунда” интервалом хранения является “минута“, с интервалом записи “минута” интервал хранения “час” и так далее.

X4 - номер интервала хранения от текущего назад, на котором закончить считывание.



Функция осуществляет считывание точек тренда группой во внутреннюю область памяти. Использование этой функции описано в разделе 6.1 «Ускорение доступа к трендам».

Пример вызова функции		
СТ:	ФБД:	
ПгТр1 (пц1, пц2, пц3, пц4) PTrend1(пц1, пц2, пц3, пц4)		

## 6.10 ПгТр2, PTrend2

**Назначение**

Подготовка группы трендов.

Отображение		
СТ:	ФБД:	
ПгТр2(X1, X2, X3, X4, X5) PTrend2(X1, X2, X3, X4, X5)		
Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16), X5(ц16)		

**Описание**

Для платформ СВВК любых версий данная функция не применяется.

**Логика работы**

X1 – порядковый номер самописца.

X2 – порядковый номер пера в этом самописце, с которого начать считывание.

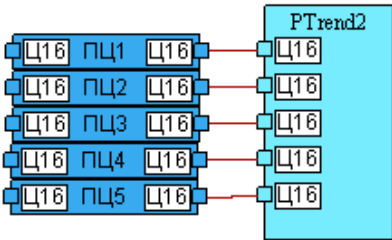
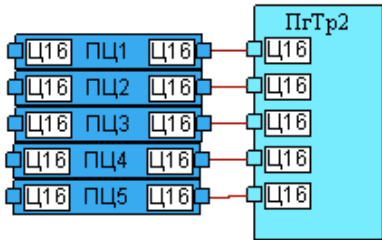
X3 – порядковый номер пера в этом самописце, на котором закончить считывание.

X4 – номер интервала хранения от текущего назад, с которого начать считывание. Для самописцев с базовым интервалом записи “секунда” интервалом хранения является “минута”, с интервалом записи “минута” интервал хранения “час” и так далее.

X5 – номер интервала хранения от текущего назад, на котором закончить считывание.

Функция осуществляет считывание точек тренда группой во внутреннюю область памяти.

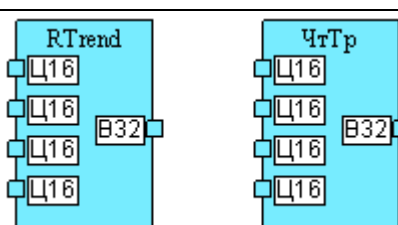
Использование этой функции описано в разделе 6.1 «Ускорение доступа к трендам».

Пример вызова функции		
СТ:	ФБД:	
PTrend2(пц1, пц2, пц3, пц4, пц5)  ПгТр2 (пц1, пц2, пц3, пц4, пц5)		

## 6.11 ЧтТр, RTrend

### Назначение

Чтение тренда.

Отображение	
СТ:	ФБД:
$Y = \text{ЧтТр}(X1, X2, X2, X4)$ $Y = \text{RTrend}(X1, X2, X2, X4)$	
Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16) Выходные параметры: Y(в32)	

### Описание

Для платформ СРВК всех версий данная функция не применяется.

### Логика работы функции

X1 – порядковый номер самописца;

X2 – порядковый номер пера в указанном самописце;

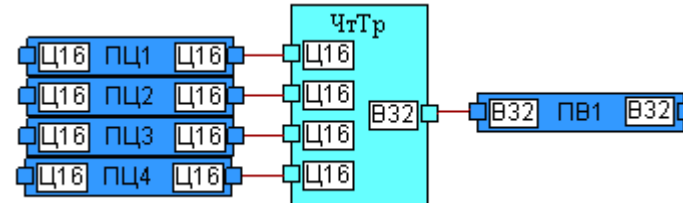
X3 – номер интервала хранения от текущего назад.

Для самописцев с базовым интервалом записи "секунда" интервалом хранения является "минута", с интервалом записи "минута" интервал хранения "час", интервалом записи "час" интервалом хранения является "сутки", с интервалом записи "сутки" интервалом хранения является "месяц".

X4 – порядковый номер точки в интервале хранения;

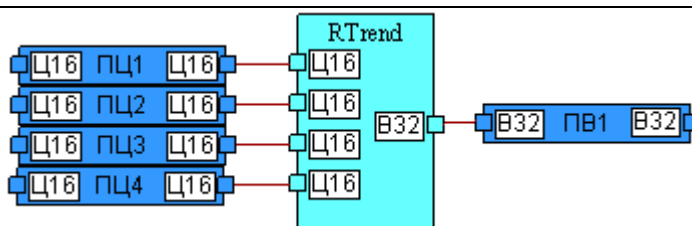
Если X1 =3 (пусть, это будет, например, часовой самописец), X2=1, X3 =2, X4 =4, то переменной Y присваивается значение точки, взятой из четвертого часа от начала суток два дня назад из тренда часового самописца 3 пера 1.

Если порядковый номер самописца соответствует суточному самописцу, то значение X4 будет соответствовать: «0» - 1-й день месяца, «1» - 2-й день месяца и т.д. Например, пв1=ЧтТр(1,1,0,4) означает, что в переменную пв1 будет записано значение точки, взятой из пятых суток текущего месяца из суточного тренда 1 пера 1. Или пв1=ЧтТр(1,1,1,4) означает, что в переменную пв1 будет записано значение точки, взятой из пятых суток прошедшего месяца из суточного тренда 1 пера 1.

Пример вызова функции	
СТ:	ФБД:
$\text{пв1} = \text{ЧтТр}(\text{пц1}, \text{пц2}, \text{пц3}, \text{пц4})$	



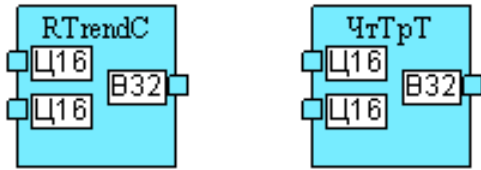
пв1 =RTrend(пц1, пц2, пц3, пц4)



6.12 ЧтТрТ, RTrendC

Назначение

Чтение текущего значения тренда.

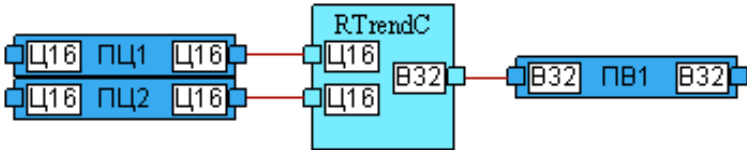
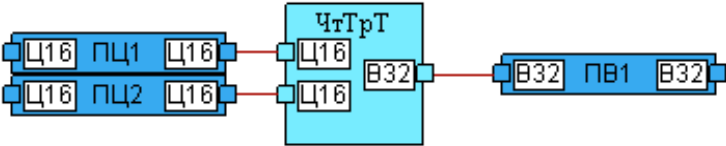
Отображение	
СТ:	ФБД:
$Y = \text{ЧТТрТ}(X1, X2)$ $Y = \text{RTrendC}(X1, X2)$	
Входные параметры: X1(ц16), X2(ц16) Выходные параметры: Y(в32)	

Описание

Для платформ СРВК любой версии данная функция не применяется.

Логика работы функции

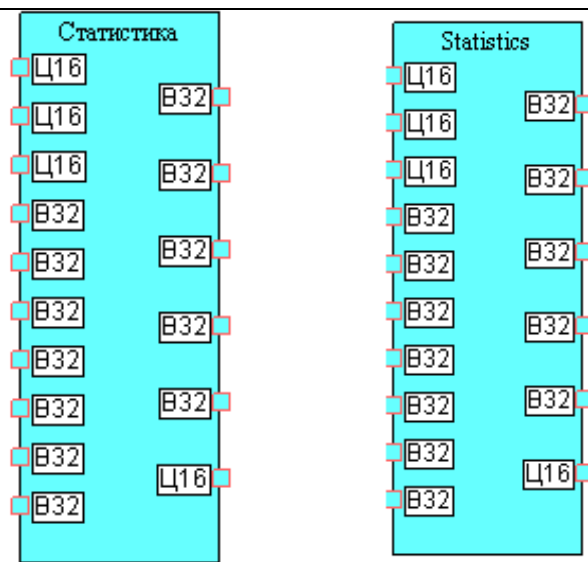
Выходному параметру Y присваивается текущее значение тренда самописца номер X1, пера номер X2.

Пример вызова функции	
СТ:	ФБД:
$\text{пв1} = \text{RTrendC}(\text{пц1}, \text{пц2})$	
$\text{пв1} = \text{ЧТТрТ}(\text{пц1}, \text{пц2})$	

## 6.13 Статистика, Statistics

### Назначение

Статистический анализ данных базового, производного или событийного тренда.

Отображение	
СТ:	ФБД:
<p>(Y1, Y2, Y3, Y4, Y5, Y6) = Статистика (X1, X2, X3, X4, X5, X6, X7, X8, X9, X10)</p> <p>(Y1, Y2, Y3, Y4, Y5, Y6) = Statistics (X1, X2, X3, X4, X5, X6, X7, X8, X9, X10)</p>	 <p>Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(в32), X5(в32), X6(в32), X7(в32), X8(в32), X9(в32), X10(в32) Выходные параметры: Y1(в32), Y2(в32), Y3(в32), Y4(в32), Y5(в32), Y6(ц16)</p>

### Логика работы функции:

Функция предназначена для статистического анализа данных выбранного тренда.

**X1** – порядковый номер самописца

**X2** – порядковый номер пера в указанном самописце

**X3** – код типа статистической обработки. Возможные значения данного параметра:

- 1 – нахождение среднего значения
- 2 – нахождение суммарного значения
- 3 – нахождение текущего значения
- 4 – нахождение минимального значения
- 5 – нахождение максимального значения
- 6 – нахождение разности значений

**X4** – дата начала диапазона выборки, в формате ДДММГГ

**X5** – время начала диапазона выборки, в формате ЧЧММСС

**X6** – дополнительный вход к входу X5, миллисекунды

**X7** – дата конца диапазона выборки, в формате ДДММГГ

**X8** – время конца диапазона выборки, в формате ЧЧММСС

**X9** – дополнительный вход к входу X8, миллисекунды

**X10** – коэффициент масштабирования по времени (задаётся в случае суммирования величин, зависящих от времени). Коэффициент масштабирования равен количеству секунд во временном интервале измеряемой величины. Возможные значения данного параметра:

- 0 – суммарное значение рассчитывается без учёта коэффициента
- 1 – для «секундных» параметров (м3/с, т/с, м/с)

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

- 60 – для «минутных» параметров (об/мин, м3/мин)
- 3600 – для «часовых» параметров (м3/ч, т/ч, км/ч)
- и т.д.

Y1 – результирующее значение

Y2 – качество результирующего значения (только для событийных трендов). Может принимать следующие значения:

- 0 – «неопределенное»
- 1 – «хорошее»
- 2 – «плохое»

Y3 – дата, соответствующая результату, в формате ДДММГГ

Y4 – время, соответствующее результату, в формате ЧЧММСС

Y5 – дополнительный выход к выходу Y4, миллисекунды

Y6 – достоверность входных данных. Может принимать следующие значения:

- 0 – входные данные достоверны
- 1 – самописца или пера с заданным номером в Базе Данных не существует
- 2 – тип статистической обработки не удовлетворяет диапазону значений [1...6]
- 3 – начало диапазона выборки больше конца диапазона выборки
- 4 – отсутствуют точки, попадающие в заданный диапазон.
- 5 – данные в процессе подготовки

Код ошибки **5** функция может вернуть только в случае работы с контроллерными платформами. В силу повышенных требований к стабильности цикла контроллера, а также в силу того, что вычитка данных тренда может занимать довольно продолжительное (до нескольких секунд) время, схема использования данной функции для контроллерной платформы отличается от схемы использования аналогичной функции для платформ станции оператора. Принципиальное отличие заключается в том, что данная функция работает асинхронно от основного цикла ПрП, то есть «в фоновом режиме». Первый вызов функции функции инициирует процесс запроса данных к модулю ведения трендов СРВК. Функция завершает работу сразу же с выставлением на выходе **«Код ошибки»** значения **5**, что означает *«данные в процессе подготовки»*. При последующих вызовах функции производится анализ готовности данных. Если данные подготовлены, то на выходе **«Код ошибки»** появляется значение, отличное от **5**, а на других выходах – значения соответствующие алгоритмам работы функции.

Работа с функцией может, например, производиться по следующему алгоритму:

**Программа stat**

**Начало**

**Если** (пл1=1)

{  
(пв8,пв9,пв10,пв11,пв12,пц4)=

Статистика(пц1,пц2,пц3,пв1,пв2,пв3,пв4,пв5,пв6,пв7)

**Если** (пц4#5)

{  
пл1=0  
}

}

**Конец**

В данном примере при выставлении в «1» переменной пл1 происходит сбор статистики, по окончании которого пл1 сбрасывается в «0».

Пример вызова функции	
СТ:	ФБД:
<p>(пв1, пв2, рв3, пв4, пв5, пц20) = Статистика (пц1, пц2, пц3, рв1, рв2, рв3, рв4, рв5, рв6, рв7)</p>	
<p>(пв1, пв2, пв3, пв4, пв5, пц20) = Statistics (пц1, пц2, пц3, рв1, рв2, рв3, рв4, рв5, рв6, рв7)</p>	

### Нахождение среднего значения (X3 = 1)

#### Для базового/производного самописца

Для нахождения среднего значения Ядро Кругола получает все точки Пера Самописца, которые попадают в заданный интервал времени, кроме «нулевой» точки (у которой время формирования совпадает с началом заданного интервала Дата/Время1).

Среднее значение за интервал рассчитывается по следующей формуле:

$$AVERAGE = \frac{\sum_{i=1}^N Val[i]}{N}$$

где

*AVERAGE* – среднее за интервал,

*Val[i]* – значение точки,

*N* – количество точек, которые попадают в заданный интервал времени, кроме «нулевой» точки.

Примечание: при определении среднего значения «нулевая» точка (у которой время формирования совпадает с началом заданного интервала) не используется.

Результат вычисления подаётся на выход Y1 «Результат».

Выходы Y2 «Качество», Y3 «Дата», Y4 «Время» и Y5 «мСек» сбрасываются в ноль.

**Пример:**

Дата	Время	Расход (м3/ч)
01.06.2009	00:00:00	50
01.06.2009	01:00:00	100
01.06.2009	02:00:00	100
...	...	...
01.06.2009	22:00:00	100
01.06.2009	23:00:00	100
02.06.2009	00:00:00	100

Для статистической обработки суточной ведомости указывают Начало и Конец диапазона выборки:

X4 «Дата1» = 010609            X5 «Время1» = 000000

X7 «Дата2» = 020609            X8 «Время2» = 000000

Для определения средней температуры за сутки из самописца выбираются 24 точки, которые попадают в заданный диапазон (кроме «нулевой»).

#### **Для событийного самописца**

Для нахождения среднего значения Ядро Кругола получает все точки Пера Самописца, которые попадают в заданный интервал времени, кроме «последней» точки (у которой время формирования совпадает с концом заданного интервала Дата/Время2). Кроме этого, еще необходима точка, у которой время изменения меньше или равно начала заданного интервала (для того чтобы знать, какое было значение пера от начала заданного интервала до первой точки).

Среднее значение рассчитывается по следующей формуле:

$$AVERAGE = \frac{\sum_{i=1}^N (Val[i] * (t_{i+1} - t_i))}{T}$$

где :

*AVERAGE* – среднее за интервал,

*Val[i]* – значение точки,

*N* – количество точек,

$(t_{i+1} - t_i)$  – интервал времени между точками *i* и *i+1*, сек,

*T* – интервал времени между первой и последней точкой в сек.

Примечания для формулы:

– время формирования первой точки  $t_1$  равно времени начала диапазона выборки;

– время формирования следующей после последней точки  $t_{N+1}$  равно времени конца диапазона выборки.

Результат вычисления подаётся на выход Y1 «Результат».

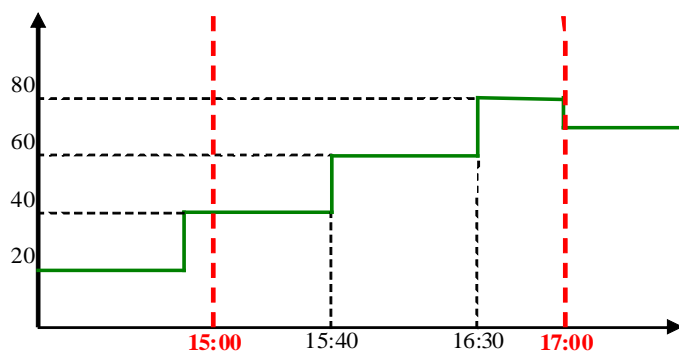
Качество среднего значения за интервал рассчитывается следующим образом:

- если качество всех точек «хорошее», то качество среднего значения тоже принимается «хорошим» (Y2 = 1);
- если качество хотя бы одной точки «плохое», то качество среднего значения тоже принимается «плохим» (Y2 = 2);
- - в остальных случаях качество среднего значения принимается «неопределенным» (Y2 = 0).

Выходы Y3 «Дата», Y4 «Время» и Y5 «мСек» сбрасываются в ноль.

**Пример:**

Время	T	Качество
14:00	20	хор
14:50	40	хор
15:40	60	хор
16:30	80	хор
17:00	70	хор



В данном примере, в заданный интервал с 15 до 17 часов попадают следующие значения:

- значение с 15:00 до 15:40;
- значение с 15:40 до 16:30;
- значение с 16:30 до 17:00.

Для определения средней температуры из самописца выбираются 2 точки, которые попадают в заданный диапазон, а также точка, у которой время изменения меньше или равно начала заданного интервала:

- 15:00:00 – 40
- 15:40:00 – 60
- 16:30:00 – 80.

**Нахождение суммарного значения (X3 = 2)**

**Для базового/производного самописца**

Для нахождения суммарного значения Ядро Кругола получает все точки Пера Самописца, которые попадают в заданный интервал времени, кроме «нулевой» точки (у которой время формирования совпадает с началом заданного интервала Дата/Время1).

Суммарное значение за интервал рассчитывается по следующей формуле:

$$SUM = \sum_{i=1}^N Val[i]$$

где  $SUM$  – суммарное за интервал,

$Val[i]$  – значение точки,

$N$  – количество точек, которые попадают в заданный интервал времени, кроме «нулевой» точки.

**Примечание:** при определении суммарного значения «нулевая» точка (у которой время формирования совпадает с началом заданного интервала) не используется.

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Результат вычисления подаётся на выход Y1 «Результат».

Выходы Y2 «Качество», Y3 «Дата», Y4 «Время» и Y5 «мСек» сбрасываются в ноль.

**Пример:**

Дата	Время	Расход (м3/ч)
01.06.2009	00:00:00	50
01.06.2009	01:00:00	100
01.06.2009	02:00:00	100
...	...	...
01.06.2009	22:00:00	100
01.06.2009	23:00:00	100
02.06.2009	00:00:00	100

Для статистической обработки суточной ведомости указывают Начало и Конец диапазона выборки:

X4 «Дата1» = 010609      X5 «Время1» = 000000

X7 «Дата2» = 020609      X8 «Время2» = 000000

Для определения суммарного расхода за сутки из самописца выбираются 24 точки, которые попадают в заданный диапазон (кроме «нулевой»).

### Для событий ийного самописца

Для нахождения суммарного значения Ядро Кругола получает все точки Пера Самописца, которые попадают в заданный интервал времени, кроме «последней» точки (у которой время формирования совпадает с концом заданного интервала Дата/Время2).

Кроме этого, также еще необходима точка, у которой время изменения меньше или равно начала заданного интервала (для того чтобы знать, какое было значение пера от начала заданного интервала до первой точки).

Суммарное значение рассчитывается по следующей формуле:

$$SUM = \sum_{i=1}^N \frac{Val[i] * (t_{i+1} - t_i)}{Kt}$$

где  $SUM$  – суммарное за интервал,

$Val[i]$  – значение точки,

$N$  – количество точек,

$(t_{i+1} - t_i)$  – интервал времени между точками  $i$  и  $i+1$ , сек,

$Kt$  – коэффициент масштабирования по времени, который необходим в случае суммирования величин, зависящих от времени (например, расходов (м<sup>3</sup>/ч, т/ч), скоростей (км/ч, об/мин) и т.д), для соблюдения размерностей.

Например, расход воды в трубе первые 30 минут был равен 40 м<sup>3</sup>/ч, следующие 30 минут – 80 м<sup>3</sup>/ч. Необходимо узнать расход воды за час.

Находим сумму по формуле:

$$40 \text{ м}^3/\text{ч} * 1800 \text{ сек} + 80 \text{ м}^3/\text{ч} * 1800 \text{ сек} = 60 \text{ м}^3.$$

$$Kt = 3600$$



Коэффициент Kt равен количеству секунд во временном интервале измеряемой величины, так:

- для «часовых» параметров (м<sup>3</sup>/ч, т/ч, км/ч) – Kt = 3600,
- для «минутных» параметров (об/мин, м<sup>3</sup>/мин) – Kt = 60,
- для «секундных» параметров (м<sup>3</sup>/с, т/с, м/с) – Kt = 1 и т.д.

Примечание для формулы: если коэффициент Kt равен нулю, то суммарное значение рассчитывается без учёта времени по следующей формуле:

$$SUM = \sum_{i=1}^N Val[i]$$

Результат вычисления подаётся на выход Y1 «Результат».

Качество суммарного значения за интервал рассчитывается следующим образом:

- если качество всех точек «хорошее», то качество суммарного значения тоже принимается «хорошим» (Y2=1);
- если качество хотя бы одной точки «плохое», то качество суммарного значения тоже принимается «плохим» (Y2=2);
- в остальных случаях качество суммарного значения принимается «неопределённым» (Y2=0).

Выходы Y3 «Дата», Y4 «Время» и Y5 «мСек» сбрасываются в ноль.

**Нахождение текущего значения (X3 = 3)**

**Для базового/производного самописца**

Для нахождения текущего значения Ядро Кругола получает последнюю точку Пера Самописца из точек, которые попадают в заданный интервал времени.

Текущее значение из полученной точки подаётся на выход Y1 «Результат».

Выход Y2 «Качество» сбрасывается в ноль.

Дата формирования полученной точки подаётся на выход Y3 «Дата».

Время формирования полученной точки подаётся на выходы Y4 «Время» и Y5 «мСек».

**Для событийного самописца**

Для нахождения текущего значения Ядро Кругола получает последнюю точку Пера Самописца из точек, которые попадают в заданный интервал времени. Если такой точки нет, то используется точка, у которой время изменения меньше или равно началу заданного интервала Дата/Время1.

Текущее значение из полученной точки подаётся на выход Y1 «Результат».

Качество результирующего значения определяется по качеству полученной точки («хорошее» - 1, «плохое» - 2, «неопределённое» - 0), и подаётся на выход Y2 «Качество».

Дата формирования полученной точки подаётся на выход Y3 «Дата».

Время формирования полученной точки подаётся на выходы Y4 «Время» и Y5 «мСек».

**Нахождение минимального значения (X3 = 4)**

**Для базового/производного самописца**

Для нахождения минимального значения Ядро Кругола получает все точки Пера Самописца, которые попадают в заданный интервал времени, кроме «нулевой» точки (у которой время формирования совпадает с началом заданного интервала Дата/Время1).

Минимальное значение из полученных точек подаётся на выход Y1 «Результат».

Выход Y2 «Качество» сбрасывается в ноль.

Дата формирования минимальной точки подаётся на выход Y3 «Дата».

Время формирования минимальной точки подаётся на выходы Y4 «Время» и Y5 «мСек».

### Для событий ийного самописца

Для нахождения минимального значения Ядро Кругола получает все точки Пера Самописца, которые попадают в заданный интервал времени, кроме «последней» точки (у которой время формирования совпадает с концом заданного интервала Дата/Время2).

Кроме этого, также еще необходима точка, у которой время изменения меньше или равно начала заданного интервала (для того чтобы знать, какое было значение пера от начала заданного интервала до первой точки).

Минимальное значение из полученных точек подаётся на выход Y1 «Результат».

Качество минимального значения определяется по качеству минимальной точки («хорошее» - 1, «плохое» - 2, «неопределенное» - 0), и подаётся на выход Y2 «Качество».

Дата формирования минимальной точки подаётся на выход Y3 «Дата».

Время формирования минимальной точки подаётся на выходы Y4 «Время» и Y5 «мСек».

### Нахождение максимального значения (X3 = 5)

#### Для базового/производного самописца

Для нахождения максимального значения Ядро Кругола получает все точки Пера Самописца, которые попадают в заданный интервал времени, кроме «нулевой» точки (у которой время формирования совпадает с началом заданного интервала Дата/Время1).

Максимальное значение из полученных точек подаётся на выход Y1 «Результат».

Выход Y2 «Качество» сбрасывается в ноль.

Дата формирования максимальной точки подаётся на выход Y3 «Дата».

Время формирования максимальной точки подаётся на выходы Y4 «Время» и Y5 «мСек».

#### Для событий ийного самописца

Для нахождения максимального значения Ядро Кругола получает все точки Пера Самописца, которые попадают в заданный интервал времени, кроме «последней» точки (у которой время формирования совпадает с концом заданного интервала Дата/Время2).

Кроме этого, также еще необходима точка, у которой время изменения меньше или равно начала заданного интервала (для того чтобы знать, какое было значение пера от начала заданного интервала до первой точки).

Максимальное значение из полученных точек подаётся на выход Y1 «Результат».

Качество максимального значения определяется по качеству максимальной точки («хорошее» - 1, «плохое» - 2, «неопределенное» - 0), и подаётся на выход Y2 «Качество».

Дата формирования максимальной точки подаётся на выход Y3 «Дата».

Время формирования максимальной точки подаётся на выходы Y4 «Время» и Y5 «мСек».

### Нахождение разности значений (X3 = 6)

#### Для базового/производного самописца

Для нахождения разности значений Ядро Кругола получает первую и последнюю точку Пера Самописца из точек, которые попадают в заданный интервал времени.

Разность значений за интервал рассчитывается по следующей формуле:

$$\Delta = Val[N] - Val[1]$$

где

$\Delta$  – разность значений за интервал,

$Val[N]$  – значение последней точки,

$Val[1]$  – значение первой точки.

Результат вычисления подаётся на выход Y1 «Результат».

Выходы Y2 «Качество», Y3 «Дата», Y4 «Время» и Y5 «мСек» сбрасываются в ноль.

**Пример:**

Дата	Время	Счетчик объема
01.06.2009	00:00:00	11000
01.06.2009	01:00:00	11100
01.06.2009	02:00:00	11200
...	...	...
01.06.2009	22:00:00	13200
01.06.2009	23:00:00	13300
02.06.2009	00:00:00	13400

Выборки:

X4 «Дата1» = 010609      X5 «Время1» = 000000

X7 «Дата2» = 020609      X8 «Время2» = 000000

Для определения приращения счетчика объема за сутки из самописца выбираются первая и последняя точки из точек, которые попадают в заданный диапазон.

$$D = 13400 - 11000 = 2400 \text{ М}^3$$

### Для событий ийного самописца

Для нахождения разности значений Ядро Кругола получает последнюю точку Пера Самописца из точек, которые попадают в заданный интервал времени. А также точку, у которой время изменения меньше или равно начала заданного интервала.

Разность значений за интервал рассчитывается по следующей формуле:

$$\Delta = Val[N] - Val[1]$$

где  $\Delta$  – разность значений за интервал,

$Val[N]$  – значение последней точки,

$Val[1]$  – значение точки, у которой время изменения меньше или равно начала заданного интервала.

Результат вычисления подаётся на выход Y1 «Результат».

Качество разности значений за интервал рассчитывается следующим образом:

- если качество последней и первой точки «хорошее», то качество разности значений тоже принимается «хорошим» (Y2 «Качество» = 1);
- если качество последней или первой точки «плохое», то качество разности значений тоже принимается «плохим» (Y2 «Качество» = 2);

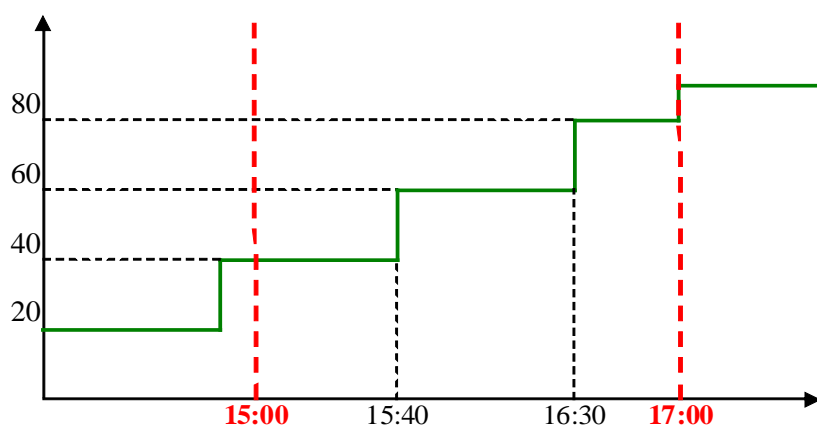
## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

- в остальных случаях качество разности значений принимается «неопределенным» (Y2 «Качество» = 0).

Выходы Y3 «Дата», Y4 «Время» и Y5 «мСек» сбрасываются в ноль.

*Пример:*

Время	qV	Качество
14:00	20	хор
14:50	40	хор
15:40	60	хор
16:30	80	хор
17:00	90	хор



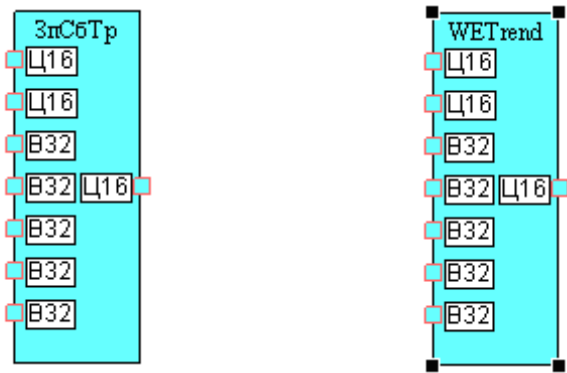
В данном примере, для определения приращения счетчика объема в интервале с 15 до 17 часов из самописца выбирается последняя точка из точек, которые попадают в заданный диапазон, а также точка, у которой время изменения меньше или равно началу заданного интервала.

$$D=90-40=50 \text{ м}^3$$

## 6.14 3пСбТр, WETrend

### Назначение

Функция записи точки событийного тренда с указанием значения, метки времени и качества

Отображение	
СТ:	ФБД:
$Y1=3пСбТр(X1,X2,X3,X4,X5,X6,X7)$  $Y1= WETrend (X1,X2,X3,X4,X5,X6,X7)$	
Входные параметры: X1 (Ц16), X2 (Ц16), X3 (В32), X4 (В32), X5 (В32), X6 (В32), X7 (В32)	
Выходные параметры: Y1 (Ц16)	

### Входные параметры:

X1 (Ц16) - Порядковый номер событийного самописца. Является обязательным входом, принимающим целые значения в диапазоне от 1 до 65535

X2 (Ц16) - Номер (идентификатор) пера в указанном событийном самописце для платформы для платформ "Среда исполнения КРУГОЛ Windows", порядковый номер пера в указанном событийном самописце для платформ "СРВК/Имитатор СРВК". Является обязательным входом, принимающим целые значения в диапазоне от 1 до 1000

X3 (В32) - Значение, записываемое в точку тренда. Является обязательным входом, принимающим вещественные значения

X4 (В32) – Дата для формирования метки времени точки пера в формате ДДММГГ. Является необязательным входом, принимающим вещественные значения в диапазоне от 010101 до 311299

X5 (В32) – Время для формирования метки времени точки пера в формате ЧЧММСС. Является необязательным входом, принимающим вещественные значения в диапазоне от 0 до 235959

X6 (В32) – Миллисекунды для формирования метки времени точки пера. Является необязательным входом, принимающим вещественные значения в диапазоне от 0 до 999

X7 (В32) - Качество значения. Является необязательным входом, принимающим вещественные значения в диапазоне от 0 до 255. Основными значениями качества являются: «хорошее» - 192, «плохое» - 0, «неопределенное» - 64.

### Выходные параметры:

Y1 (Ц16) - Код достоверности (ошибки) входных параметров. Является обязательным выходом, принимающим целые значения в диапазоне от 0 до 5:

0 - Если во входных данных функции ошибок нет

1 - Если заданного номера событийного самописца (вход X1) или номера пера самописца (вход X2) в базе данных не существует, или значения входов X1, X2 не соответствуют формату данных

2 - Если записываемое в точку пера значение (вход X3) не соответствует формату данных

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

- 3 - Если любой из параметров задаваемой даты (вход X4), времени (вход X5) или миллисекунд (вход X6) при их явном описании не соответствуют формату
- 4 - Если задаваемое значение качества (вход X7) не соответствует формату данных.
- 5 - Если возникают ошибки, возвращаемые функциями API при выполнении записи точки в событийный самописец.

### Логика работы функции

Функция осуществляет запись в событийный самописец точку пера с параметрами: значение, дата, время и качество.

Запись осуществляется вставкой новой точки пера самописца в соответствии с заданной датой и временем.

Задание даты и времени может осуществляться явным образом путём указания на соответствующих входах функции констант или ссылок на переменные (локальные и переменные базы данных) и их атрибуты.

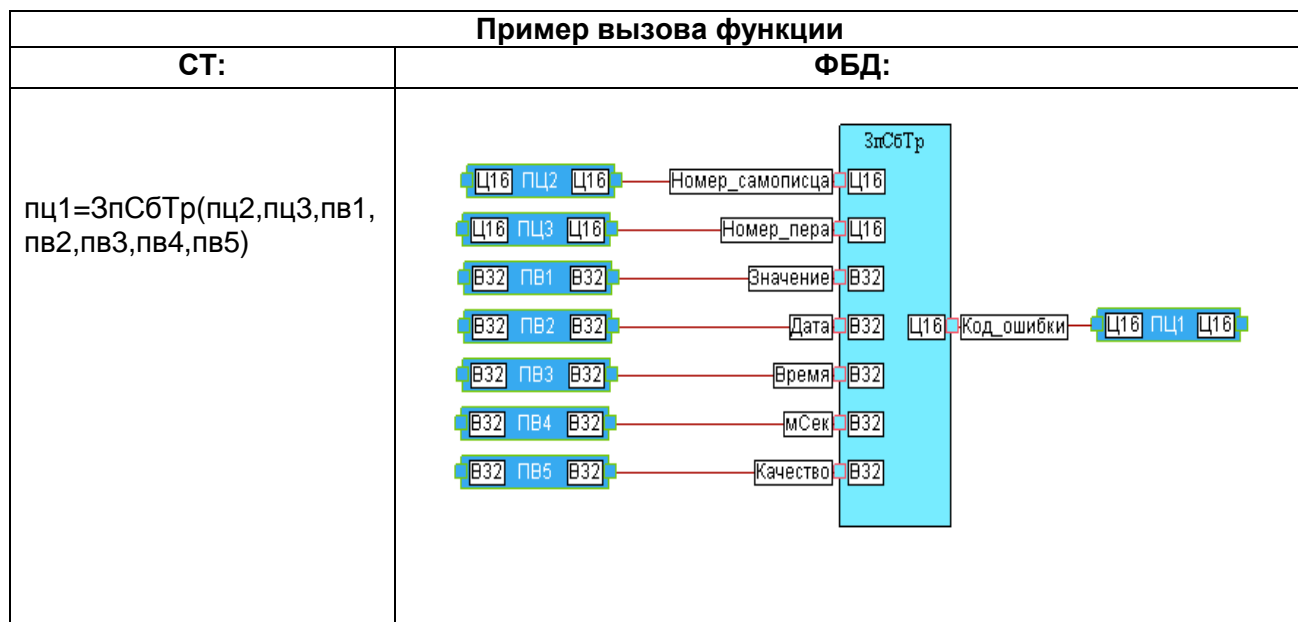
Кроме того, запись точки в событийный самописец может осуществляться без указания явным образом даты и времени (случай, когда вход X4 «Дата» пропущен при объявлении функции). При этом в качестве даты и времени записываемой точки будет автоматически использоваться текущие дата и время контроллера.

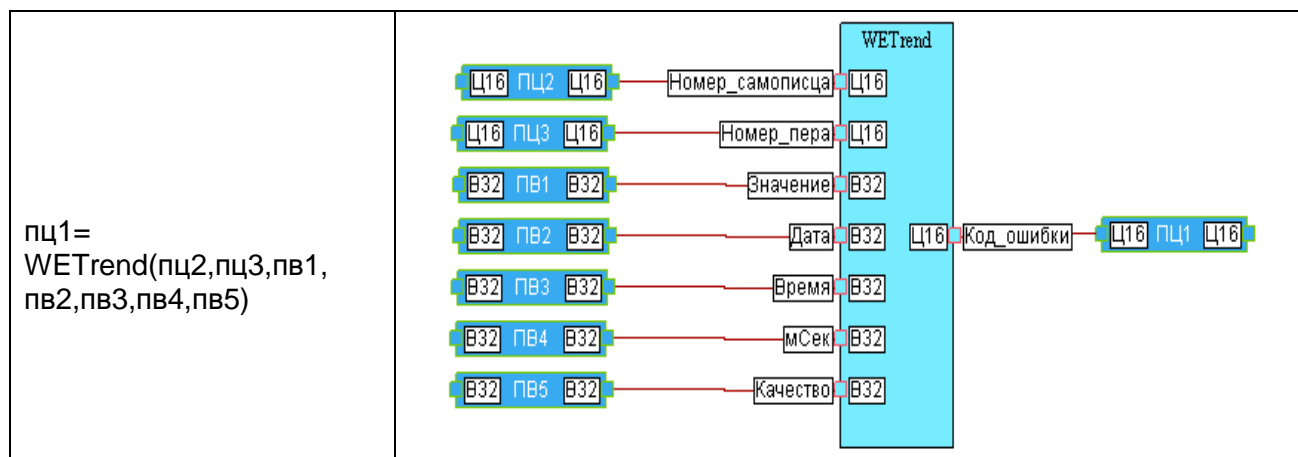
В зависимости от результата выполнения функции возвращает код достоверности (ошибки).



### ВНИМАНИЕ!!!

- 1) При записи точки пера с меткой времени меньше времени последней опрошенной точки пера сервером БД, передача данной точки пера на верхний уровень не осуществляется.
- 2) При записи точки пера с меткой времени большей, чем текущее время контроллера или с меткой времени меньше чем глубина событийного самописца, точка пера самописца не формируется, а код достоверности (ошибки) выставляется в нулевое значение.
- 3) Существующие точки пера самописца не перезаписываются. При этом, при попытке перезаписи существующей точки пера в код достоверности (ошибки) записывается нулевое значение.

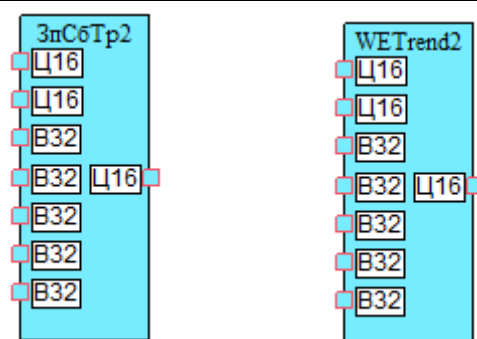




## 6.15 3пС6Тр2, WETrend2

### Назначение

Функция записи точки событийного тренда с указанием значения, метки времени и качества

Отображение	
СТ:	ФБД:
$Y1=3пС6Тр2(X1,X2,X3,X4,X5,X6,X7)$  $Y1= WETrend2(X1,X2,X3,X4,X5,X6,X7)$	
<p>Входные параметры: X1 (Ц16), X2 (Ц16), X3 (B32), X4 (B32), X5 (B32), X6 (B32), X7 (B32)</p> <p>Выходные параметры: Y1 (Ц16)</p>	

### Входные параметры:

- X1 (Ц16) - Порядковый номер событийного самописца. Является обязательным входом, принимающим целые значения в диапазоне от 1 до 65535
- X2 (Ц16) - Номер пера в указанном событийном самописце, выводимый в интерфейсе Генератора базы данных SCADA КРУГ-2000. Является обязательным входом, принимающим целые значения в диапазоне от 1 до 1000
- X3 (B32) - Значение, записываемое в точку тренда. Является обязательным входом, принимающим вещественные значения
- X4 (B32) – Дата для формирования метки времени точки пера в формате ДДММГГ. Является необязательным входом, принимающим вещественные значения в диапазоне от 010101 до 311299
- X5 (B32) – Время для формирования метки времени точки пера в формате ЧЧММСС. Является необязательным входом, принимающим вещественные значения в диапазоне от 0 до 235959
- X6 (B32) – Миллисекунды для формирования метки времени точки пера. Является необязательным входом, принимающим вещественные значения в диапазоне от 0 до 999
- X7 (B32) - Качество значения. Является необязательным входом, принимающим вещественные значения в диапазоне от 0 до 255. Основными значениями качества являются: «хорошее» - 192, «плохое» - 0, «неопределенное» - 64.

### Выходные параметры:

- Y1 (Ц16) - Код достоверности (ошибки) входных параметров. Является обязательным выходом, принимающим целые значения в диапазоне от 0 до 5:
- 0 - Если во входных данных функции ошибок нет
  - 1 - Если заданного номера событийного самописца (вход X1) или номера пера самописца (вход X2) в базе данных не существует, или значения входов X1, X2 не соответствуют формату данных
  - 2 - Если записываемое в точку пера значение (вход X3) не соответствует формату данных
  - 3 - Если любой из параметров задаваемой даты (вход X4), времени (вход X5) или миллисекунд (вход X6) при их явном описании не соответствуют формату



4 - Если задаваемое значение качества (вход X7) не соответствует формату данных.

5 - Если возникают ошибки, возвращаемые функциями API при выполнении записи точки в событийный самописец.

### Логика работы функции

Функция осуществляет запись в событийный самописец точку пера с параметрами: значение, дата, время и качество.

Запись осуществляется вставкой новой точки пера самописца в соответствии с заданной датой и временем.

Задание даты и времени может осуществляться явным образом путём указания на соответствующих входах функции констант или ссылок на переменные (локальные и переменные базы данных) и их атрибуты.

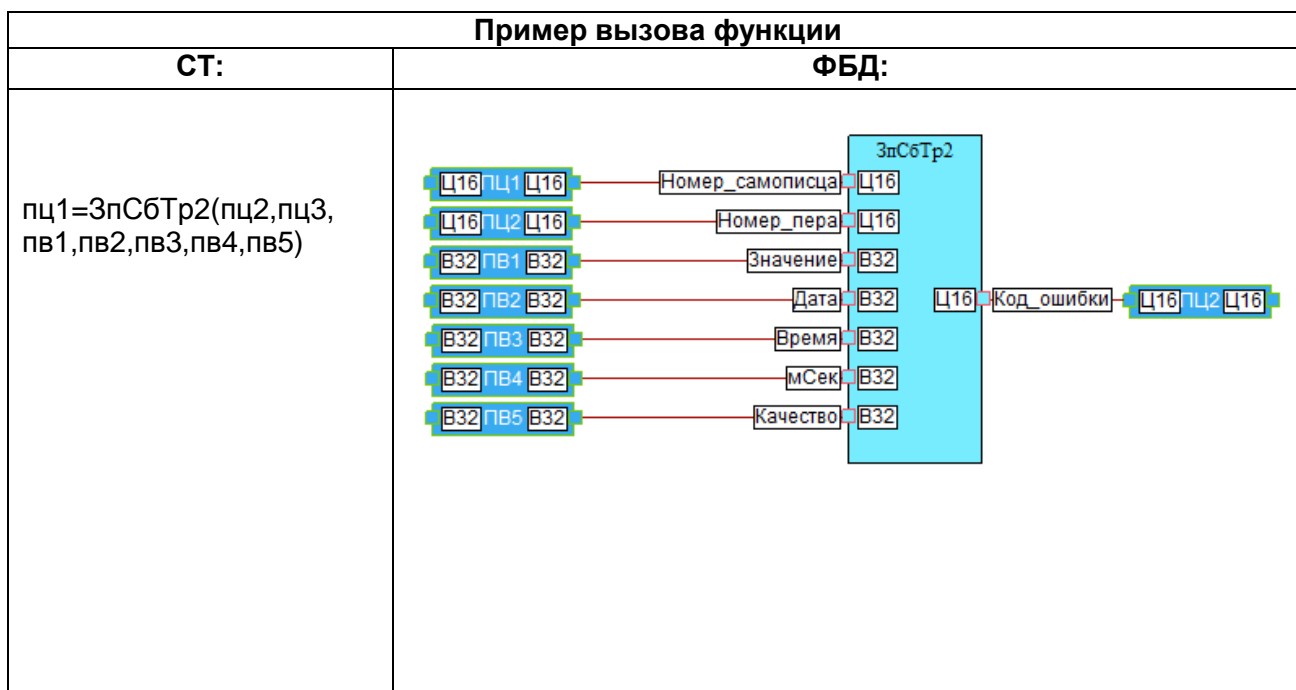
Кроме того, запись точки в событийный самописец может осуществляться без указания явным образом даты и времени (случай, когда вход X4 «Дата» пропущен при объявлении функции). При этом в качестве даты и времени записываемой точки будет автоматически использоваться текущие дата и время контроллера.

В зависимости от результата выполнения функции возвращает код достоверности (ошибки).

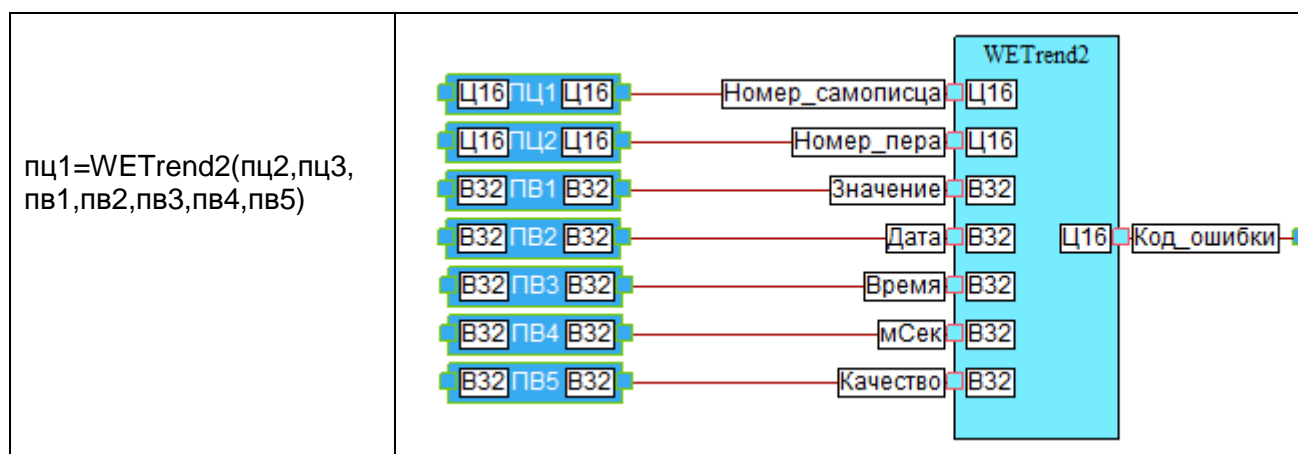


### ВНИМАНИЕ!!!

- 1) При записи точки пера с меткой времени меньше времени последней опрошенной точки пера сервером БД, передача данной точки пера на верхний уровень не осуществляется.
- 2) При записи точки пера с меткой времени большей, чем текущее время контроллера или с меткой времени меньше чем глубина событийного самописца, точка пера самописца не формируется, а код достоверности (ошибки) выставляется в нулевое значение.
- 3) Существующие точки пера самописца не перезаписываются. При этом, при попытке перезаписи существующей точки пера в код достоверности (ошибки) записывается нулевое значение.



## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ





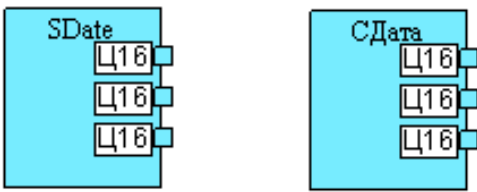


## 7 ДАТА И ВРЕМЯ

### 7.1 СДата, SDate

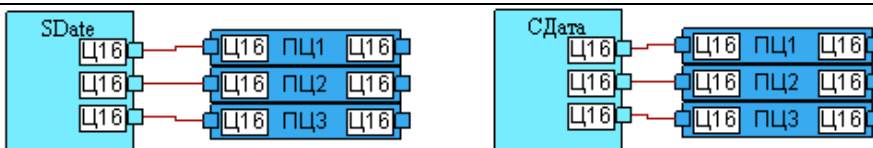
#### Назначение

Чтение текущего года, месяца и дня.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3) = \text{СДата}()$ $(Y1, Y2, Y3) = \text{SDate}()$	
Выходные параметры: Y1(ц16), Y2(ц16), Y3(ц16)	

#### Логика работы функции

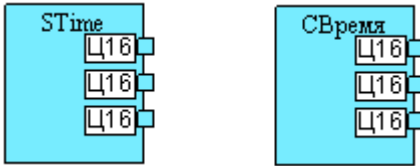
Выходным параметрам Y1, Y2, Y3 присваиваются значения:  
Y1-текущего года (номер), Y2- номер месяца, Y3 - номер дня.

Пример вызова функции	
СТ:	ФБД:
$(\text{пц1}, \text{пц2}, \text{пц3}) = \text{SDate}()$ $(\text{пц1}, \text{пц2}, \text{пц3}) = \text{СДата}()$	

7.2 СВремя, STime

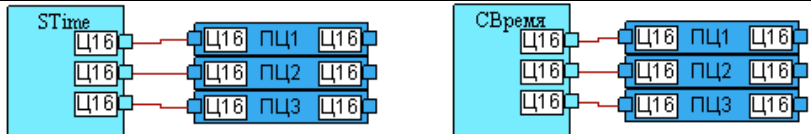
Назначение

Чтение текущего часа, минуты и секунды.

Отображение	
СТ:	ФБД:
(Y1, Y2, Y3) = STime()  (Y1, Y2, Y3) = СВремя()	
Выходные параметры: Y1(ц16), Y2(ц16), Y3(ц16)	

Логика работы функции

Выходным параметрам Y1, Y2, Y3 присваиваются значения:  
Y1- астрономического часа, Y2 - количества минут, Y3 - количества секунд.

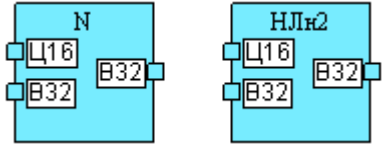
Пример вызова функции	
СТ:	ФБД:
(пц1, пц2, пц3) = СВремя( )  (пц1, пц2, пц3) = STime( )	

## 8 ЧТЕНИЕ ИЗ ТАБЛИЦ НЕЛИНЕЙНОСТИ

### 8.1 НЛн2, N

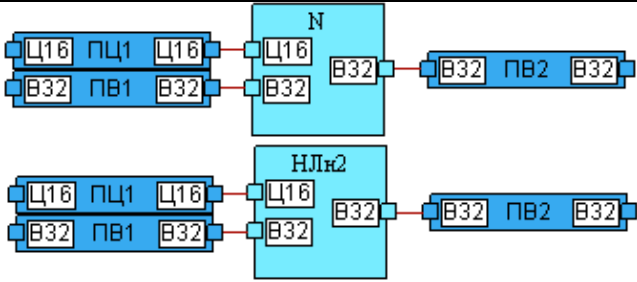
#### Назначение

Обращение к двумерным таблицам нелинейности.

Отображение	
СТ:	ФБД:
$Y = \text{НЛн2}(X1, X2)$ $Y = N(X1, X2)$	
Входные параметры: X1(ц16), X2(в32) Выходные параметры: Y(в32)	

#### Логика работы функции

Переменной Y присваивается значение точки двумерной таблицы с номером X1, точка по координате X определена значением X2.

Пример вызова функции	
СТ:	ФБД:
$\text{пв2} = N(\text{пц1}, \text{пв1})$ $\text{пв2} = \text{НЛн2}(\text{пц1}, \text{пв1})$	

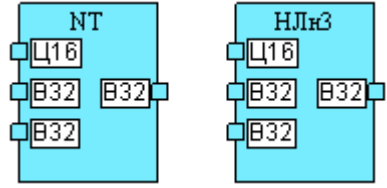


ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

8.2 НЛнЗ, NT

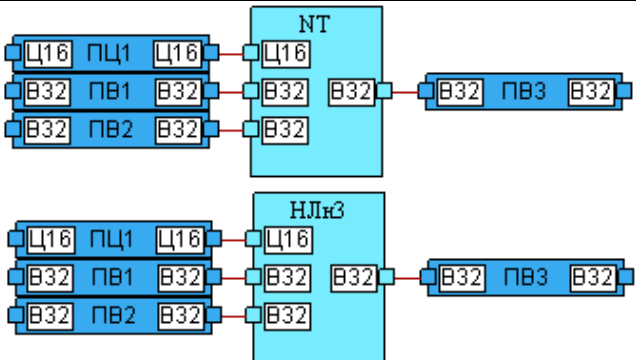
Назначение

Обращение к трехмерным таблицам нелинейности.

Отображение	
СТ:	ФБД:
$Y = \text{НЛнЗ}(X1, X2, X3)$ $Y = \text{NT}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(в32), X3(в32) Выходные параметры: Y(в32)	

Логика работы функции

Переменной Y присваивается значение точки трехмерной таблицы с номером X1, точка по координате X определена значением X2, а по координате Y определена значением X3.

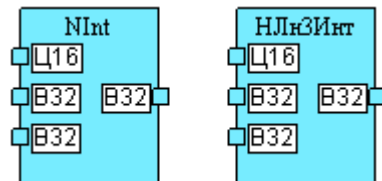
Пример вызова функции	
СТ:	ФБД:
$\text{пвЗ} = \text{NT}(\text{пц1}, \text{пв1}, \text{пв2})$ $\text{пвЗ} = \text{НЛнЗ}(\text{пц1}, \text{пв1}, \text{пв2})$	



### 8.3 НЛн3Инт, NInt

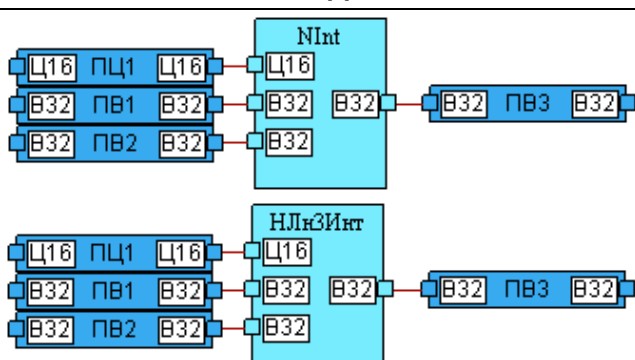
#### Назначение

Обращение к трехмерным интервальным таблицам нелинейности.

Отображение	
СТ:	ФБД:
$Y = NInt(X1, X2, X3)$ $Y = НЛн3Инт(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(в32), X3(в32) Выходные параметры: Y(в32)	

#### Логика работы функции

Переменной Y присваивается значение точки трехмерной интервальной таблицы с номером X1, точка по координате X определена значением X2, а по координате Y определена значением X3.

Пример вызова функции	
СТ:	ФБД:
$пв3 = NInt(пц1, пв1, пв2)$ $пв3 = НЛн3Инт(пц1, пв1, пв2)$	

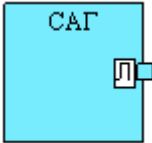


## 9 РАБОТА С ПРИЗНАКАМИ СИГНАЛИЗАЦИИ

### 9.1 САГ

#### Назначение

Функция признака аварийной сигнализации.

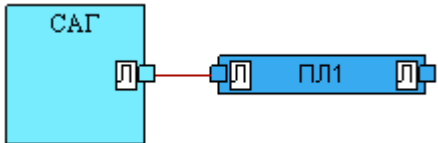
Отображение	
СТ:	ФБД:
$Y = \text{САГ}()$	
Выходные параметры: Y(л)	

#### Описание

Данная функция не применяется для платформ среды исполнения КРУГОЛ любой версии.

#### Логика работы функции

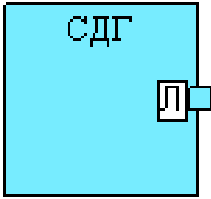
Функция возвращает значение логической «1» при наличии признаков «Новое нарушение по НАГ (нижняя предаварийная граница сигнализации)» или «Новое нарушение по ВАГ (верхняя предаварийная граница сигнализации)» хотя бы у одной переменной в базе данных. После квитирования данных признаков функция возвращает значение логического «0».

Пример вызова функции	
СТ:	ФБД:
пл1 = САГ()	

## 9.2 СДГ

### Назначение

Функция признака сигнализации по недостоверности.

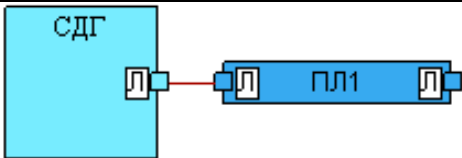
Отображение	
СТ:	ФБД:
$Y = \text{СДГ}()$	
Выходные параметры: Y(л)	

### Описание

Данная функция не применяется для платформ среды исполнения КРУГОЛ любой версии.

### Логика работы функции

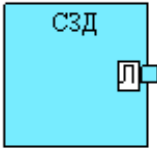
Функция возвращает значение логической «1» при наличии признаков «Новое нарушение по недостоверности» хотя бы у одной переменной в базе данных. После квитирования данного признака функция возвращает значение логического «0».

Пример вызова функции	
СТ:	ФБД:
$\text{пл1} = \text{СДГ}()$	

## 9.3 СЗД

**Назначение**

Функция признака сигнализации по отклонению от задания.

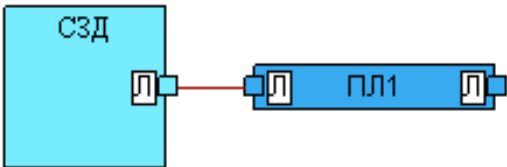
Отображение	
СТ:	ФБД:
$Y = \text{СЗД}()$	
Выходные параметры: Y(л)	

**Описание**

Данная функция не применяется для платформ среды исполнения КРУГОЛ любой версии.

**Логика работы функции**

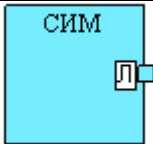
Функция возвращает значение логической «1» при наличии признаков «Новое нарушение положительного отклонения от задания» или «Новое нарушение отрицательного отклонения от задания» хотя бы у одной аналоговой выходной переменной в базе данных. После квитирования данных признаков функция возвращает значение логического «0».

Пример вызова функции	
СТ:	ФБД:
$\text{пл1} = \text{СЗД}()$	

## 9.4 СИМ

### Назначение

Функция признака сигнализации по ИМ.

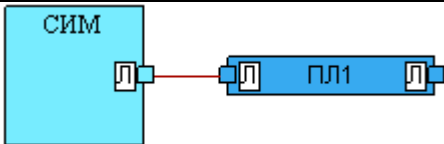
Отображение	
СТ:	ФБД:
$Y = СИМ()$	
Выходные параметры: Y(л)	

### Описание

Данная функция не применяется для платформ среды исполнения КРУГОЛ любой версии.

### Логика работы функции

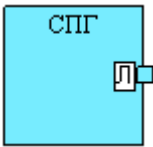
Функция возвращает значение логической «1» при наличии признаков «Новое нарушение верхней границы сигнализации по ИМ» или «Новое нарушение нижней границы сигнализации по ИМ» хотя бы у одной аналоговой выходной переменной в базе данных. После квитирования данных признаков функция возвращает значение логического «0».

Пример вызова функции	
СТ:	ФБД:
пл1 = СИМ()	

## 9.5 СПГ

### Назначение

Функция признака предупредительной сигнализации.

Отображение	
СТ:	ФБД:
$Y = \text{СПГ}()$	
Выходные параметры: Y(л)	

### Описание

Данная функция не применяется для платформ среды исполнения КРУГОЛ любой версии.

### Логика работы функции

Функция возвращает значение логической «1» при наличии признаков «Новое нарушение по НПГ (нижняя предупредительная граница сигнализации)» или «Новое нарушение по ВПГ (верхняя предупредительная граница сигнализации)» хотя бы у одной переменной в базе данных. После квитирования данных признаков функция возвращает значение логического «0».

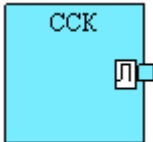
Пример вызова функции	
СТ:	ФБД:
пл1 = СПГ()	

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

### 9.6 ССК

#### Назначение

Функция признака сигнализации по скорости роста / падения параметра.

Отображение	
СТ:	ФБД:
$Y = \text{ССК}()$	
Выходные параметры: Y(л)	

#### Описание

Данная функция не применяется для платформ среды исполнения КРУГОЛ любой версии.

#### Логика работы функции

Функция возвращает значение логической «1» при наличии признаков «Новое нарушение границы сигнализации по скорости роста параметра» или «Новое нарушение границы сигнализации по скорости падения параметра» хотя бы у одной аналоговой переменной в базе данных. После квитирования данных признаков функция возвращает значение логического «0».

Пример вызова функции	
СТ:	ФБД:
пл1 = ССК()	



## 10 ПЕРЕДАЧА ПАСПОРТА ПЕРЕМЕННОЙ

### 10.1 Ограничения на использование функций посХХ

#### Назначение

В данном разделе описываются ограничения на использование семейства функций посВА, посВД, посДВ, посАВ, посРВ и посПР.

Под функцией посХХ подразумевается любая из функций посВА, посВД, посДВ, посАВ, посРВ, посПР.

#### Ограничение на использование в программах для Станции оператора.

Выполнение функции посХХ зависит от номера канала, на который назначена переменная, для которой выполняется функция посХХ. Если номер канала равен нулю, то никаких действий функция не производит. Далее будем считать, что функция посХХ выполняется для переменной с ненулевым номером канала.

Функции посХХ выполняются не в момент вызова функции в программе на КРУГОЛ, а после выполнения всех программ, в конце цикла. Функция посХХ выполняется следующим образом:

- Ядро КРУГОЛ дает команду Серверу базы данных на формирование команды для контроллера.
- Сформированная команда помещается в буфер команд. Размер буфера – 100 команд. Если буфер заполнен, то новая команда записывается на место самой старой команды.
- Команда из буфера посылается в контроллер при очередном сеансе обмена с контроллером. Т.е. вместо запроса текущих значений из контроллера, Сервер базы данных отправляет в УСО одну команду из буфера. Пока в буфере есть хотя бы одна команда, Сервер базы данных будет посылать в контроллер команды. Вызов функций посХХ замедляет получение новых текущих значений из контроллера.

Как часто можно вызывать функцию посХХ? Необходимо знать период опроса Сервером базы данных канала (выделено красным цветом на рисунке).

IP адрес (Имя)	192.9.200.155
№ канала резервного 1	0
№ канала резервного 2	0
IP порт	2100
Период опроса (мсек)	100
Передача групп паспортов	<input type="checkbox"/>
Поддержка 24-х символьной позиции	<input type="checkbox"/>
Кол-во переменных типа ВА в УСО	8
Кол-во переменных типа АВ в УСО	8
Кол-во переменных типа ВД в УСО	8
Кол-во переменных типа ДВ в УСО	8
Кол-во переменных типа РВ в УСО	8
Состояние канала	
<input type="radio"/> Вкл.	<input checked="" type="radio"/> Выкл.

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Функции посХХ не должны вызываться чаще, чем столько раз, сколько циклов опроса укладывается в один цикл КРУГОЛ для «Станции оператора» (1 сек). Для приведенного примера на рис. 1 функции посХХ можно вызывать не более 10 раз за один цикл КРУГОЛ – при таком количестве вызовов функции посХХ **обмен текущими значениями с контроллером остановится**.

### Ограничение на использование в программах для Контроллера.

Функция посХХ выполняется в момент ее вызова в программе для КРУГОЛ следующим образом:

- 1 Ядро КРУГОЛ формирует команду для Сервера базы данных.
- 2 Команда помещается в буфер команд. Размер буфера – 10 команд. Если буфер команд заполнен, то новая команда помещается на место самой старой команды.
- 3 При очередном запросе Сервером базы данных текущих значений, ему вместо текущих значений возвращается команда.

Примечание - если выполнялась, например, функция посВА, то команда будет отдана Серверу базы данных только при запросе им текущих значений входных аналоговых переменных. Т.е. вызов функции посВА будет тормозить только обмен текущими значениями входных аналоговых переменных. Соответственно, посВД – только обмен текущими значениями входных дискретных переменных.

Как часто может вызываться функция посХХ? Необходимо знать, какое время требуется Серверу базы данных, чтобы опросить контроллер. Время опроса канала Сервером базы данных показывается на кадре диагностики каналов связи (выделено красным цветом на рисунке).

Диагностика каналов связи

порт	Управление каналом связи		Статус канала	Статус УСО	№ рез. канала	Состояние схемы АВР	Цикл опроса (мсек)	Кол-во пакетов	Время опроса (сек)
200	<input type="button" value="Включить"/>	<input type="button" value="Отключить"/>	Отключен	Основной	2	Автомат	100	0	0.00
200	<input type="button" value="Включить"/>	<input type="button" value="Отключить"/>	Отключен	Резервный	1	Автомат	100	0	0.00
200	<input type="button" value="Включить"/>	<input type="button" value="Отключить"/>	Отключен	Основной	4	Автомат	100	0	0.00
200	<input type="button" value="Включить"/>	<input type="button" value="Отключить"/>	Отключен	Резервный	3	Автомат	100	0	0.00
<div><input type="button" value="Все вкл."/><input type="button" value="Все откл."/></div>									

состояние

Диагностика сети

Функция посХХ не должна выполняться чаще, чем один раз за время опроса «Сервером базы данных». Например, пусть Сервер базы данных опрашивает контроллер за 1200 мс, цикл контроллера составляет 400 мс, следовательно, функцию посХХ можно вызывать не чаще, чем один раз за 3 цикла КРУГОЛ – при таком периодическом вызове функции посХХ **обмен текущими значениями с сервером БД остановится**.

Способы уменьшения количества вызовов функций посХХ.

Дискретные значения необходимо передавать по изменению.

```
: Было
: посВД(1)
: Стало
Если пл1#вд1.а32
{
    посВД(1)
    пл1=вд1.а32
}
```

Аналоговые значения необходимо передавать при их изменении на некоторую величину

```
: Было
: посВА(1)
: Стало
Если 0.01<fabs(пв1-ва1.а13)
{
    : Здесь при изменении значения более чем на 0.01 выполняется
    : функция посВА
    посВА(1)
    пв1=ва1.а13
}
```

Если есть большая необходимость передавать большое количество переменных, то их необходимо разбить на группы, и в каждом цикле контроллера передавать по одной группе.

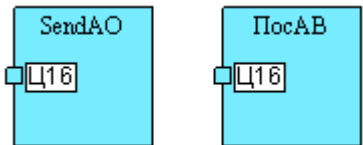
```
: Было
: посВА(10)
: посВА(11)
: посВА(12)
: посВА(34)
: посВА(100)
: посВА(13)
: Стало
Если пц1=0 {
    посВА(10)
    посВА(11)
    пц1=1
}
Иначе_Если пц1=1 {
    посВА(12)
    посВА(34)
    пц1=2
}
Иначе {
    посВА(100)
    посВА(13)
    пц1=0
}
}
```

## 10.2 ПосАВ, SendAO

### Назначение

Передача в УСО / из УСО паспорта переменной типа АВ.

Ограничения на использование функции посАВ описаны в разделе 10.1 «Ограничения на использование функций посХХ».

Отображение	
СТ:	ФБД:
ПосАВ (X) SendAO (X)	
Входные параметры: X(ц16)	

### Логика работы функции

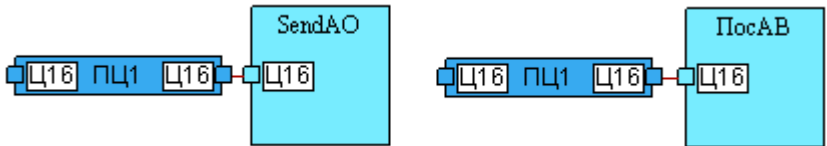
X - номер переменной типа АВ, паспорт которой необходимо передать в УСО / из УСО.

Для станции оператора:

1. Паспорт переменной передается в контроллер или драйвер (зависит от привязки переменной).
2. Паспорт переменной будет передан в конце цикла выполнения программ пользователя.

Для контроллера:

1. Паспорт переменной будет передан в Сервер БД.
2. Паспорт переменной будет передан в ближайшем цикле опроса Сервером БД контроллера.

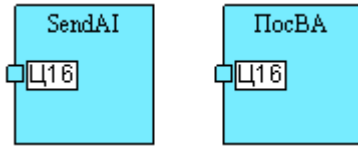
Пример вызова функции	
СТ:	ФБД:
ПосАВ (пц1) SendAO (пц1)	

### 10.3 ПосВА, SendAI

#### Назначение

Передача в УСО / из УСО паспорта переменной типа ВА.

Ограничения на использование функции посВА описаны в разделе 10.1 «Ограничения на использование функций посХХ».

Отображение	
СТ:	ФБД:
ПосВА (X) SendAI (X)	
Входные параметры: X(ц16)	

#### Логика работы функции

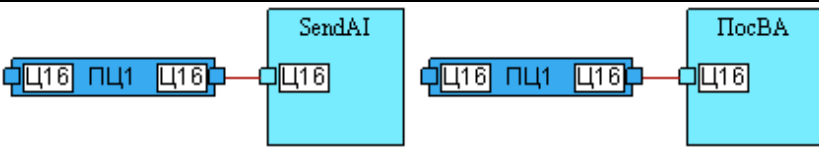
X - номер переменной типа ВА, паспорт которой необходимо передать в УСО / из УСО.

Для станции оператора:

1. Паспорт переменной передается в контроллер или драйвер (зависит от привязки переменной).
2. Паспорт переменной будет передан в конце цикла выполнения программ пользователя.

Для контроллера:

1. Паспорт переменной будет передан в Сервер БД.
2. Паспорт переменной будет передан в ближайшем цикле опроса Сервером БД контроллера.

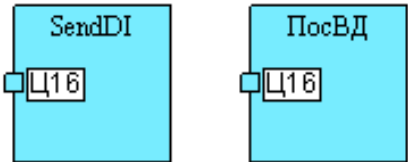
Пример вызова функции	
СТ:	ФБД:
ПосВА(пц1) SendAI(пц1)	

## 10.4 ПосВД, SendDI

### Назначение

Передача в УСО / из УСО паспорта переменной типа ВД.

Ограничения на использование функции посВД описаны в разделе 10.1 «Ограничения на использование функций посХХ».

Отображение	
СТ:	ФБД:
ПосВД(Х) SendDI(Х)	
Входные параметры: Х(ц16)	

### Логика работы функции

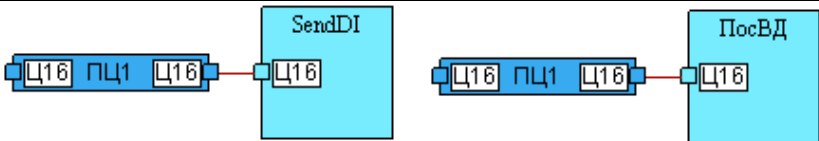
Х - номер переменной тип ВД, паспорт которой необходимо передать в УСО / из УСО.

Для станции оператора:

1. Паспорт переменной передается в контроллер или драйвер (зависит от привязки переменной).
2. Паспорт переменной будет передан в конце цикла выполнения программ пользователя.

Для контроллера:

1. Паспорт переменной будет передан в Сервер БД.
2. Паспорт переменной будет передан в ближайшем цикле опроса Сервером БД контроллера.

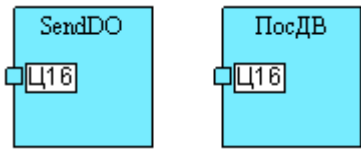
Пример вызова функции	
СТ:	ФБД:
ПосВД (пц1) SendDI (пц1)	

## 10.5 ПосДВ, SendDO

### Назначение

Передача в УСО / из УСО паспорта переменной типа ДВ.

Ограничения на использование функции посДВ описаны в разделе 10.1 «Ограничения на использование функций посХХ».

Отображение	
СТ:	ФБД:
ПосДВ(Х) SendDO(Х)	
Входные параметры: Х(ц16)	

### Логика работы функции

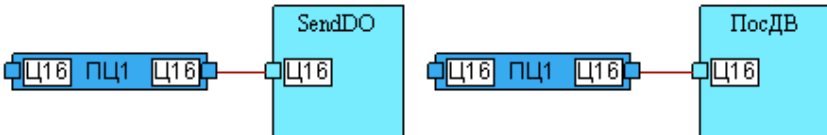
Х - номер переменной типа ДВ, паспорт которой необходимо передать в УСО / из УСО.

Для станции оператора:

1. Паспорт переменной передается в контроллер или драйвер (зависит от привязки переменной).
2. Паспорт переменной будет передан в конце цикла выполнения программ пользователя.

Для контроллера:

1. Паспорт переменной будет передан в Сервер БД.
2. Паспорт переменной будет передан в ближайшем цикле опроса Сервером БД контроллера.

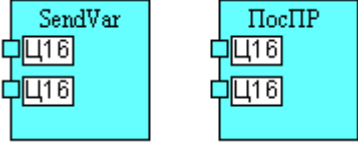
Пример вызова функции	
СТ:	ФБД:
ПосДВ(пц1) SendDO(пц1)	

## 10.6 ПосПР, SendVar

### Назначение

Функция передачи паспорта переменной.

Ограничения на использование функции посПР описаны в разделе 10.1 «Ограничения на использование функций посХХ».

Отображение	
СТ:	ФБД:
<p>ПосПР(X1, X2)</p> <p>SendVar(X1, X2)</p>	
Входные параметры: X(ц16)	

### Логика работы функции

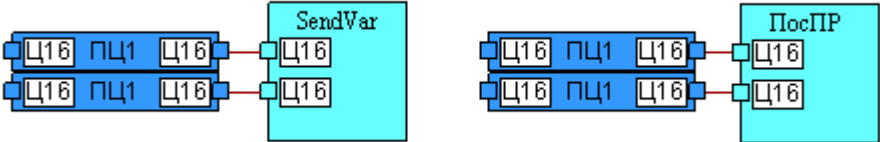
Универсальная функция, позволяющая переслать паспорт любой переменной любого типа.

X1 – тип переменной: 1 – ВА; 2 – ВД; 3 – РВ; 6 – ДВ; 8 – АВ.

X2 – номер переменной.

Для контроллера:

1. Паспорт переменной будет передан в Сервер БД.
2. Паспорт переменной будет передан в ближайшем цикле опроса Сервером БД контроллера.

Пример вызова функции	
СТ:	ФБД:
<p>ПосПР (пц1, пц2)</p> <p>SendVar (пц1, пц2)</p>	

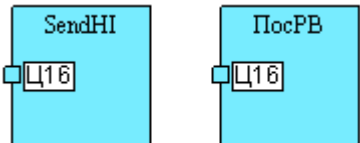


## 10.7 ПосРВ, SendHI

### Назначение

Передача в УСО / из УСО паспорта переменной типа РВ.

Ограничения на использование функции посРВ описаны в разделе 10.1 «Ограничения на использование функций посХХ».

Отображение	
СТ:	ФБД:
SendHI (X) ПосРВ (X)	
Входные параметры: X(ц16)	

### Логика работы функции

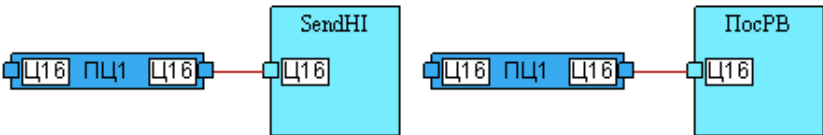
X - номер переменной типа РВ, паспорт которой необходимо передать в УСО / из УСО.

Для станции оператора:

1. Паспорт переменной передается в контроллер или драйвер (зависит от привязки переменной).
2. Паспорт переменной будет передан в конце цикла выполнения программ пользователя.

Для контроллера:

1. Паспорт переменной будет передан в Сервер БД.
2. Паспорт переменной будет передан в ближайшем цикле опроса Сервером БД контроллера.

Пример вызова функции	
СТ:	ФБД:
SendHI (пц1) посРВ (пц1)	

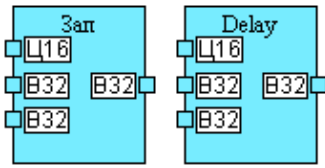


## 11 ДИНАМИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ

### 11.1 Зап, Delay

#### Назначение

Запаздывание

Отображение	
СТ:	ФБД:
$Y = \text{Зап}(X1, X2, X3)$ $Y = \text{Delay}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(в32), X3(в32) Выходные параметры: Y(в32)	

#### Описание

Применяется для платформ СРВК версии ниже 8.0, и среды исполнения КРУГОЛ версии ниже 2.2.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

#### Логика работы функции

Алгоритм моделирует звено чистого запаздывания.

Время запаздывания должно быть не более  $100 \cdot T_{\text{ц}}$ , где  $T_{\text{ц}}$  – цикл работы контроллера (станции оператора) в секундах.

Алгоритм содержит 100 ячеек памяти.

X1 - порядковый номер (от 1 до 20 000). Для платформ СРВК версии 7.1 и среды исполнения КРУГОЛ версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

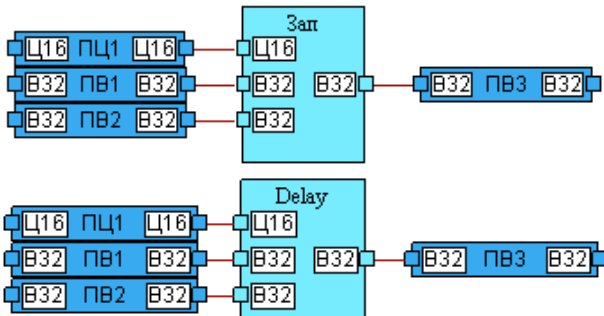
X2 - входное значение.

X3 - время запаздывания в секундах,  $T_3$ .



**Внимание!!!**

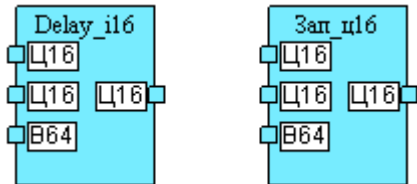
Если время запаздывания равно нулю ( $X3=0$ ), то входной сигнал сразу подается на выход  $Y = X2$ .

Пример вызова функции	
СТ:	ФБД:
$\text{пв3} = \text{Зап}(\text{пц1}, \text{пв1}, \text{пв2})$ $\text{пв3} = \text{Delay}(\text{пц1}, \text{пв1}, \text{пв2})$	

## 11.2 Зап\_ц16, Delay\_I16

### Назначение

Запаздывание.

Отображение	
СТ:	ФБД:
$Y = \text{Зап\_ц16}(X1, X2, X3)$ $Y = \text{Delay\_i16}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(ц16), X3(в64) Выходные параметры: Y(ц16)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

### Логика работы функции

Алгоритм моделирует звено чистого запаздывания.

Время запаздывания должно быть не более  $100 \cdot T_{\text{ц}}$ , где  $T_{\text{ц}}$  – цикл работы контроллера (станции оператора) в секундах.

Алгоритм содержит 100 ячеек памяти.

X1 - Порядковый номер (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

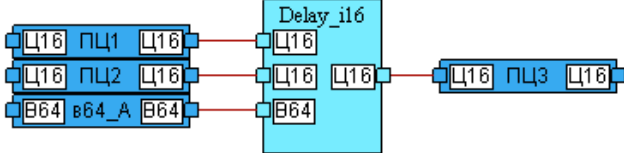
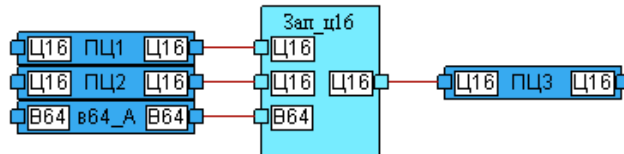
X2 - Входное значение.

X3 - Время запаздывания в секундах,  $T_3$ .



**Внимание!!!**

Если время запаздывания равно нулю ( $X3=0$ ), то входной сигнал сразу подается на выход  $Y = X2$ .

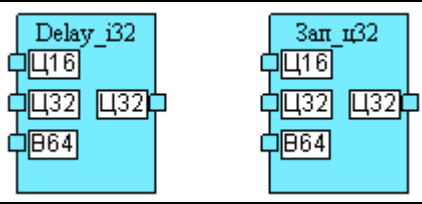
Пример вызова функции	
СТ:	ФБД:
$\text{пц3} = \text{Delay\_i16}(\text{пц1}, \text{пц2}, \text{в64\_A})$	
$\text{пц3} = \text{Зап\_ц16}(\text{пц1}, \text{пц2}, \text{в64\_A})$	

Здесь в64\_A – глобальная переменная вещественного 64-битного типа.

### 11.3 Зап\_ц32, Delay\_i32

#### Назначение

Запаздывание.

Отображение	
СТ:	ФБД:
$Y = \text{Зап\_ц32}(X1, X2, X3)$ $Y = \text{Delay\_i32}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(ц32), X3(в64) Выходные параметры: Y(ц32)	

#### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

#### Логика работы функции

Алгоритм моделирует звено чистого запаздывания.

Время запаздывания должно быть не более  $100 \cdot T_{\text{ц}}$ , где  $T_{\text{ц}}$  – цикл работы контроллера (станции оператора) в секундах.

Алгоритм содержит 100 ячеек памяти.

X1 - Порядковый номер (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

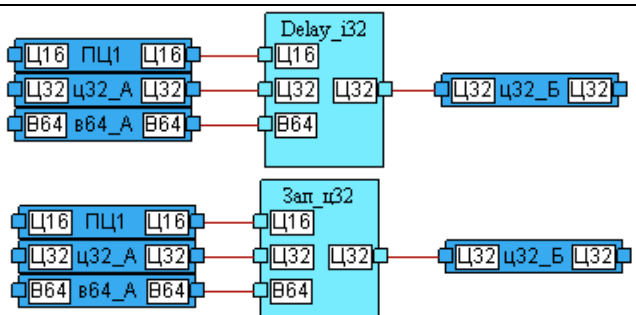
X2 - Входное значение.

X3 - Время запаздывания в секундах,  $T_{\text{з}}$ .



**Внимание!!!**

**Если время запаздывания равно нулю ( $X3=0$ ), то входной сигнал сразу подается на выход  $Y = X2$ .**

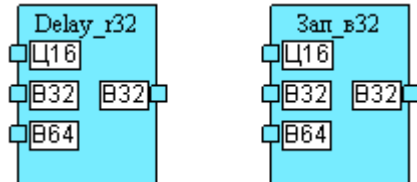
Пример вызова функции	
СТ:	ФБД:
$\text{ц32\_Б} = \text{Delay\_i32}(\text{пц1}, \text{ц32\_А}, \text{в64\_А})$ $\text{ц32\_Б} = \text{Зап\_ц32}(\text{пц1}, \text{ц32\_А}, \text{в64\_А})$	

Здесь в64\_А – глобальная переменная вещественного 64-битного типа и ц32\_А, ц32\_Б – глобальные переменные целого 32-битного типа.

## 11.4 Зап\_в32, Delay\_r32

### Назначение

Запаздывание.

Отображение	
СТ:	ФБД:
$Y = \text{Зап\_в32} (X1, X2, X3)$ $Y = \text{Delay\_r32} (X1, X2, X3)$	
Входные параметры: X1(ц16), X2(в32), X3(в64) Выходные параметры: Y(в32)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

### Логика работы функции

Алгоритм моделирует звено чистого запаздывания.

Время запаздывания должно быть не более  $100 \cdot T_{\text{ц}}$ , где  $T_{\text{ц}}$  – цикл работы контроллера (станции оператора) в секундах.

Алгоритм содержит 100 ячеек памяти.

X1 - Порядковый номер (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

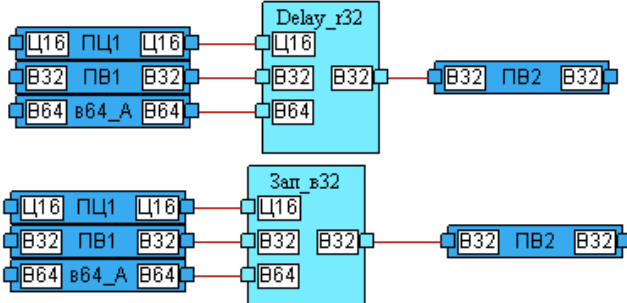
X2 - Входное значение.

X3 - Время запаздывания в секундах,  $T_{\text{з}}$ .



**Внимание!!!**

Если время запаздывания равно нулю ( $X3=0$ ), то входной сигнал сразу подается на выход  $Y = X2$ .

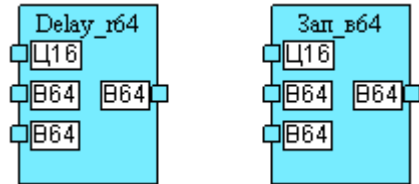
Пример вызова функции	
СТ:	ФБД:
$\text{пв2} = \text{Delay\_r32}(\text{пц1}, \text{пв1}, \text{в64\_A})$ $\text{пв2} = \text{Зап\_в32}(\text{пц1}, \text{пв1}, \text{в64\_A})$	

Здесь  $\text{в64\_A}$  – глобальная переменная вещественного 64-битного типа.

## 11.5 Зап\_в64, Delay\_r64

**Назначение**

Запаздывание.

Отображение	
СТ:	ФБД:
$Y = \text{Зап\_в64}(X1, X2, X3)$ $Y = \text{Delay\_r64}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(в64), X3(в64) Выходные параметры: Y(в64)	

**Описание**

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

**Логика работы функции**

Алгоритм моделирует звено чистого запаздывания.

Время запаздывания должно быть не более  $100 \cdot T_{\text{ц}}$ , где  $T_{\text{ц}}$  – цикл работы контроллера (станции оператора) в секундах.

Алгоритм содержит 100 ячеек памяти.

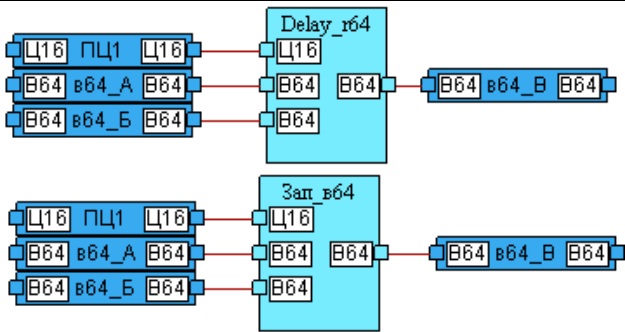
X1 - Порядковый номер (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - Входное значение.

X3 - Время запаздывания в секундах,  $T_{\text{з}}$ .

**Внимание!!!**

Если время запаздывания равно нулю ( $X3=0$ ), то входной сигнал сразу подается на выход  $Y = X2$ .

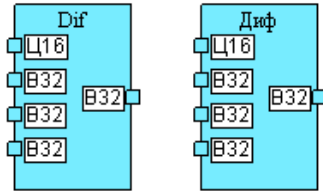
Пример вызова функции	
СТ:	ФБД:
$\text{в64\_В} = \text{Delay\_r64}(\text{пц1}, \text{в64\_А}, \text{в64\_Б})$ $\text{в64\_В} = \text{Зап\_в64}(\text{пц1}, \text{в64\_А}, \text{в64\_Б})$	

Здесь в64\_А, в64\_Б, в64\_В – глобальные переменные вещественного 64-битного типа.

## 11.6 Диф, Dif

### Назначение

Дифференцирование

Отображение	
СТ:	ФБД:
$Y = \text{Диф}(X1, X2, X3, X4)$ $Y = \text{Dif}(X1, X2, X3, X4)$	
Входные параметры: X1(ц16), X2(в32), X3(в32), X4(в32) Выходные параметры: Y(в32)	

### Описание

Применяется для платформ СРВК версии ниже 8.0, и среды исполнения КРУГОЛ версии ниже 2.2.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

### Логика работы функции

Передаточная функция алгоритма:

$$W_{(P)} = \frac{Y_{(P)}}{X_{(P)}} = K \times \frac{T_D^P}{T_D^{P+1}}$$

X1 - порядковый номер дифференциатора (от 1 до 20 000). Для платформ СРВК версии 7.1 и среды исполнения КРУГОЛ версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - входное значение.

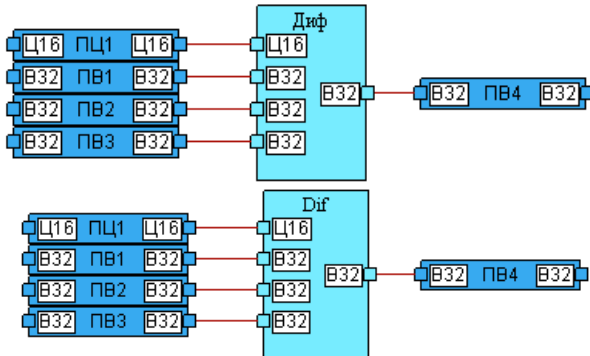
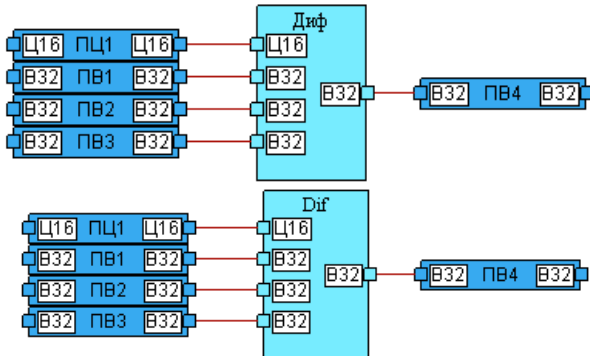
X3 - коэффициент предварения, Кд.

X4 - постоянная времени дифференцирования в секундах, Тд.



**Внимание!!!**

Если постоянная времени дифференцирования равна нулю (X4=0), то выходное значение также будет равно нулю Y1=0.

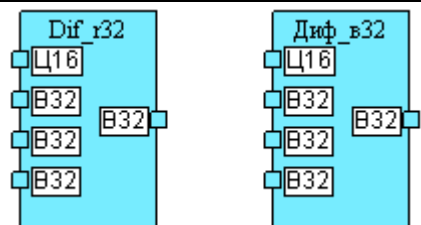
Пример вызова функции	
СТ:	ФБД:
$пв4 = \text{Диф}(пц1, пв1, пв2, пв3)$	
$пв4 = \text{Dif}(пц1, пв1, пв2, пв3)$	



## 11.7 Диф\_v32, Dif\_r32

**Назначение**

Дифференцирование.

Отображение	
СТ:	ФБД:
$Y = \text{Dif\_r32}(X1, X2, X3, X4)$ $Y = \text{Диф\_в32}(X1, X2, X3, X4)$	
Входные параметры: X1(ц16), X2(в32), X3(в32), X4(в32) Выходные параметры: Y(в32)	

**Описание**

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

**Логика работы функции**

Передаточная функция алгоритма:

$$W_{(P)} = \frac{Y_{(P)}}{X_{(P)}} = K_D \times \frac{T_D P}{T_D P + 1}$$

X1 - Порядковый номер дифференциатора (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

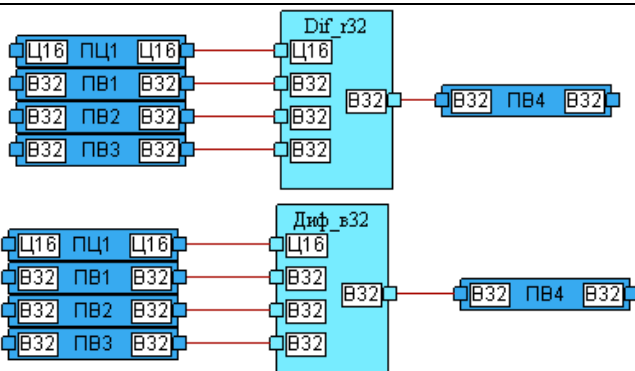
X2 - Входное значение.

X3 - Коэффициент дифференцирования, КД.

X4 - Постоянная времени дифференцирования в секундах, ТД.

**Внимание!!!**

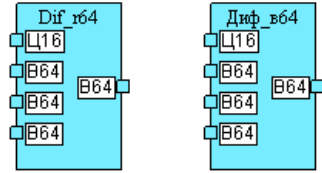
Если постоянная времени дифференцирования равна нулю (X4=0), то выходное значение также будет равно нулю Y1=0.

Пример вызова функции	
СТ:	ФБД:
$\text{пв4} = \text{Dif\_r32}(\text{пц1}, \text{пв1}, \text{пв2}, \text{пв3})$ $\text{пв4} = \text{Диф\_в32}(\text{пц1}, \text{пв1}, \text{пв2}, \text{пв3})$	

11.8 Диф\_в64, Dif\_r64

Назначение

Дифференцирование.

Отображение	
СТ:	ФБД:
$Y = \text{Dif\_r64}(X1, X2, X3, X4)$ $Y = \text{Диф\_в64}(X1, X2, X3, X4)$	
Входные параметры: X1(ц16), X2(в64), X3(в64), X4(в64) Выходные параметры: Y(в64)	

Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

Логика работы функции

Передаточная функция алгоритма:

$$W_{(P)} = \frac{Y_{(P)}}{X_{(P)}} = K \times \frac{T_D^P}{T_D^{P+1}}$$

X1 - Порядковый номер дифференциатора (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

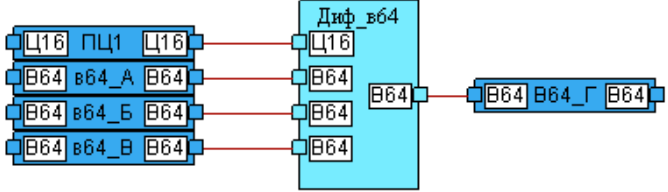
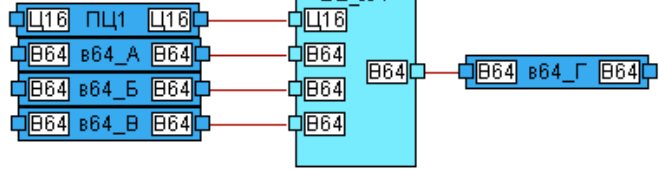
X2 - Входное значение.

X3 - Коэффициент предварения, КД.

X4 - Постоянная времени дифференцирования в секундах, ТД.

 **Внимание!!!**

**Если постоянная времени дифференцирования равна нулю (X4=0), то выходное значение также будет равно нулю Y1=0.**

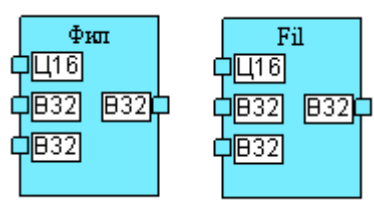
Пример вызова функции	
СТ:	ФБД:
$\text{в64\_Г} = \text{Диф\_в64}(\text{пц1}, \text{в64\_А}, \text{в64\_Б}, \text{в64\_В})$	
$\text{в64\_Г} = \text{Dif\_r64}(\text{пц1}, \text{в64\_А}, \text{в64\_Б}, \text{в64\_В})$	

Здесь в64\_А, в64\_Б, в64\_В, в64\_Г – глобальные переменные вещественного 64-битного типа.

## 11.9 Фил, Fil

**Назначение**

Фильтр входной аналоговой переменной.

Отображение	
СТ:	ФБД:
$Y = \text{Фил}(X1, X2, X3)$ $Y = \text{Fil}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(в32), X3(в32) Выходные параметры: Y(в32)	

**Описание**

Применяется для платформ СРВК версии ниже 8.0, и среды исполнения КРУГОЛ версии ниже 2.2.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

**Логика работы функции**

Передаточная функция алгоритма равна

$$W_{(P)} = \frac{Y_{(P)}}{X_{(P)}} = \frac{1}{T_{\phi}^{P+1}}$$

X1 - порядковый номер фильтра (от 1 до 20 000). Для платформ СРВК версии 7.1 и среды исполнения КРУГОЛ версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

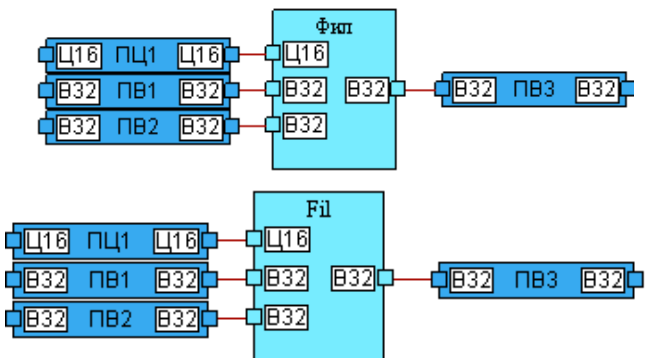
X2 - входное значение.

X3 - постоянная времени фильтра в секундах, Тф.



**Внимание!!!**

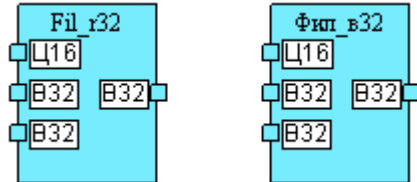
Если постоянная фильтра равна нулю (X3=0), выходное значение равно входному Y=X2.

Пример вызова функции	
СТ:	ФБД:
$пв3 = \text{Фил}(пц1, пв1, пв2)$ $пв3 = \text{Fil}(пц1, пв1, пв2)$	

11.10 Фил\_в32, Fil\_r32

Назначение

Наложение фильтра первого порядка (инерционное звено) на аналоговый параметр.

Отображение	
СТ:	ФБД:
$Y = \text{Fil\_r32}(X1, X2, X3)$ $Y = \text{Фил\_в32}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(в32), X3(в32) Выходные параметры: Y(в32)	

Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

Логика работы функции

Передаточная функция алгоритма равна:

$$W_{(P)} = \frac{Y_{(P)}}{X_{(P)}} = \frac{1}{T_{\phi} P + 1}$$

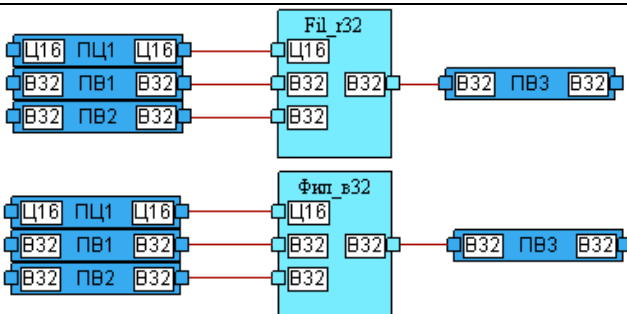
X1 - Порядковый номер фильтра (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - Входное значение.

X3 - Постоянная времени фильтра в секундах, Тф.

 **Внимание!!!**

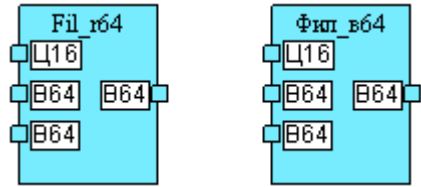
Если постоянная фильтра равна нулю (X3=0), выходное значение равно входному Y=X2.

Пример вызова функции	
СТ:	ФБД:
$\text{пв3} = \text{Fil\_r32}(\text{пц1}, \text{пв1}, \text{пв2})$  $\text{пв3} = \text{Фил\_в32}(\text{пц1}, \text{пв1}, \text{пв2})$	

## 11.11 Фил\_в64, Fil\_r64

**Назначение**

Наложение фильтра первого порядка (инерционное звено) на аналоговый параметр.

Отображение	
СТ:	ФБД:
$Y = \text{Fil\_r64}(X1, X2, X3)$ $Y = \text{Фил\_в64}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(в64), X3(в64) Выходные параметры: Y(в64)	

**Описание**

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

**Логика работы функции**

Передаточная функция алгоритма равна:

$$W_{(P)} = \frac{Y_{(P)}}{X_{(P)}} = \frac{1}{T_{\phi} P + 1}$$

X1 - Порядковый номер фильтра (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

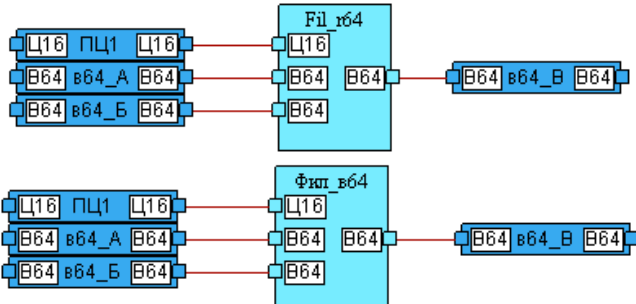
X2 - Входное значение.

X3 - Постоянная времени фильтра в секундах, Тф.



**Внимание!!!**

Если постоянная фильтра равна нулю (X3=0), выходное значение равно входному Y=X2.

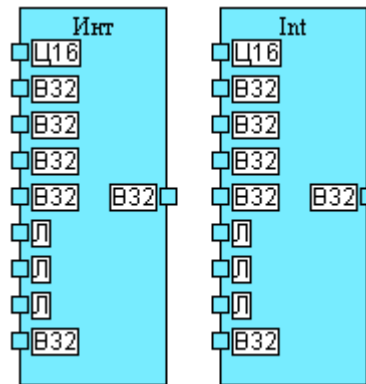
Пример вызова функции	
СТ:	ФБД:
$v64\_B = \text{Fil\_r64}(\text{пц1}, v64\_A, v64\_B)$  $v64\_B = \text{фил\_в64}(\text{пц1}, v64\_A, v64\_B)$	

Здесь v64\_A, v64\_B, v64\_B – глобальные переменные вещественного 64-битного типа.

## 11.12 Инт, Int

### Назначение

Интегрирование.

Отображение	
СТ:	ФБД:
$Y = \text{Инт}(X1, X2, X3, X4, X5, X6, X7, X8, X9)$ $Y = \text{Int}(X1, X2, X3, X4, X5, X6, X7, X8, X9)$	
Входные параметры: X1(ц16), X2(в32), X3(в32), X4(в32), X5(в32), X6(л), X7(л), X8(л), X9(в32) Выходные параметры: Y(в32)	

### Описание

Применяется для платформ СРВК версии ниже 8.0, и среды исполнения КРУГОЛ версии ниже 2.2.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

### Логика работы функции

Передаточная функция алгоритма:

$$W_{(P)} = \frac{Y_{(P)}}{X_{(P)}} = \frac{1}{T_u P}$$

X1 - порядковый номер интегратора (от 1 до 20 000). Для платформ СРВК версии 7.1 и среды исполнения КРУГОЛ версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - входное значение.

X3 - время интегрирования, Ти



**Внимание!!!**

Если время интегрирования равно нулю (X3=0), то значение выхода будет рассчитываться  $Y1 = Y1_{\text{предыдущее}} + X2$ .

X4 - минимальное значение интегрирования.

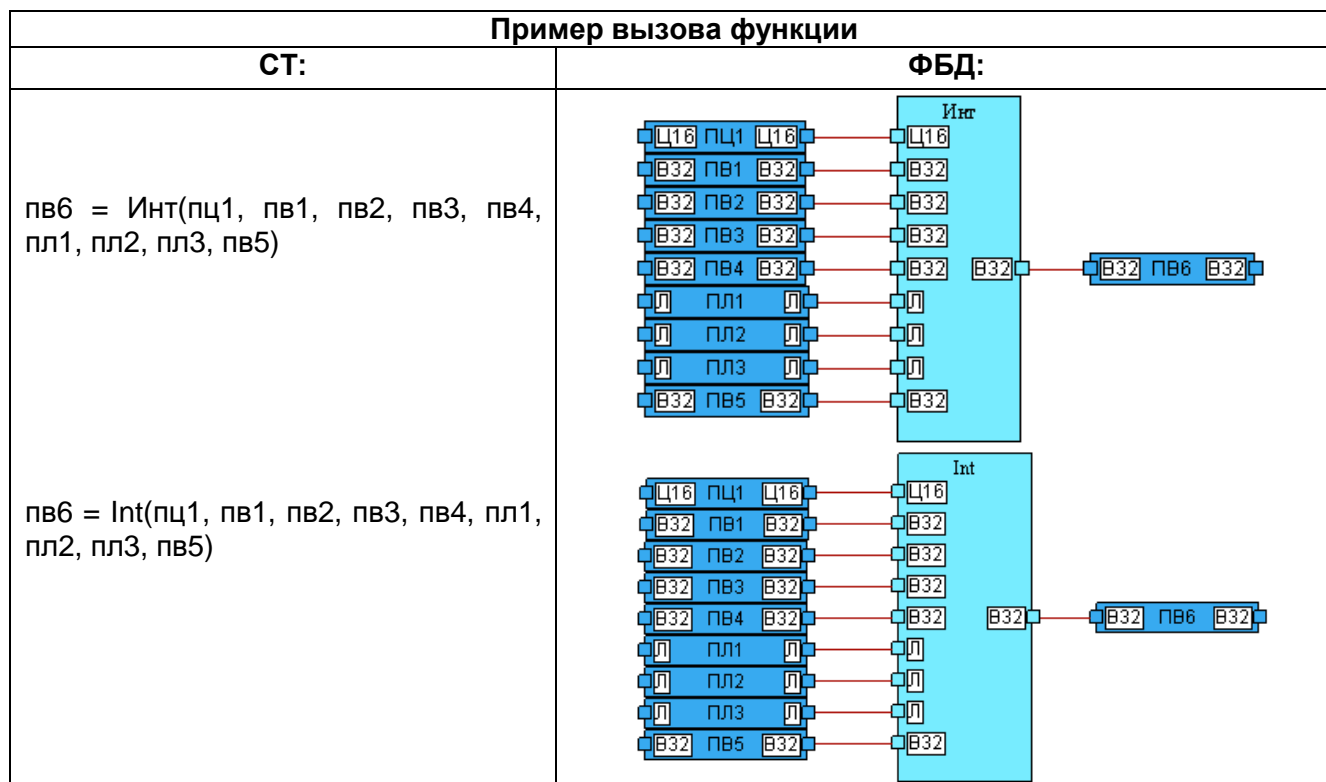
X5 - максимальное значение интегрирования.

X6 - флаг запрета интегрирования в сторону уменьшения.

X7 - флаг запрета интегрирования в сторону увеличения.

X8 - флаг сброса выходного значения в начальное значение.

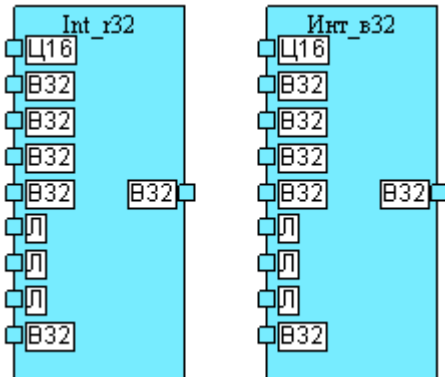
X9 - начальное значение.



### 11.13 Инт\_в32, Int\_r32

#### Назначение

Интегрирование.

Отображение	
СТ:	ФБД:
$Y = \text{Int\_r32}(X1, X2, X3, X4, X5, X6, X7, X8, X9)$  $Y = \text{Инт\_в32}(X1, X2, X3, X4, X5, X6, X7, X8, X9)$	
<p>Входные параметры: X1(ц16), X2(в32), X3(в32), X4(в32), X5(в32), X6(л), X7(л), X8(л), X9(в32)</p> <p>Выходные параметры: Y(в32)</p>	

#### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

#### Логика работы функции

Передаточная функция алгоритма:

$$W_{(P)} = \frac{Y_{(P)}}{X_{(P)}} = \frac{1}{T_u P}$$

X1 - Порядковый номер интегратора (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - Входное значение.

X3 - Постоянная времени интегрирования, Ти.



**Внимание!!!**

Если время интегрирования равно нулю (X3=0), то значение выхода будет рассчитываться  $Y1(i) = Y1(i-1) + X2(i)$ .

X4 - Минимальное значение интегрирования.

X5 - Максимальное значение интегрирования.

X6 - Флаг запрета интегрирования в сторону уменьшения.

X7 - Флаг запрета интегрирования в сторону увеличения.

X8 - Флаг сброса выходного значения в начальное значение.

X9 - Начальное значение.

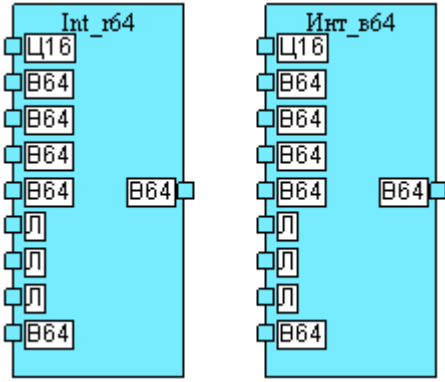


Пример вызова функции	
СТ:	ФБД:
<p>пв6 = Int_r32(пц1, пв1, пв2, пв3, пв4, пл1, пл2, пл3, пв5)</p>	
<p>пв6 = Инт_в32(пц1, пв1, пв2, пв3, пв4, пл1, пл2, пл3, пв5)</p>	

## 11.14 Инт\_в64, Int\_r64,

### Назначение

Интегрирование

Отображение	
СТ:	ФБД:
$Y = \text{Int\_r64}(X1, X2, X3, X4, X5, X6, X7, X8, X9)$  $Y = \text{Инт\_в64}(X1, X2, X3, X4, X5, X6, X7, X8, X9)$	
Входные параметры: X1(ц16), X2(в64), X3(в64), X4(в64), X5(в64), X6(л), X7(л), X8(л), X9(в64) Выходные параметры: Y(в64)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

### Логика работы функции

Передаточная функция алгоритма:

$$W_{(P)} = \frac{Y_{(P)}}{X_{(P)}} = \frac{1}{T_u P}$$

X1 - Порядковый номер интегратора (от 1 до 20 000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - Входное значение.

X3 - Время интегрирования, TИ.



**Внимание!!!**

Если время интегрирования равно нулю (X3=0), то значение выхода будет рассчитываться  $Y1(i) = Y1(i-1) + X2(i)$ .

X4 - Минимальное значение интегрирования.

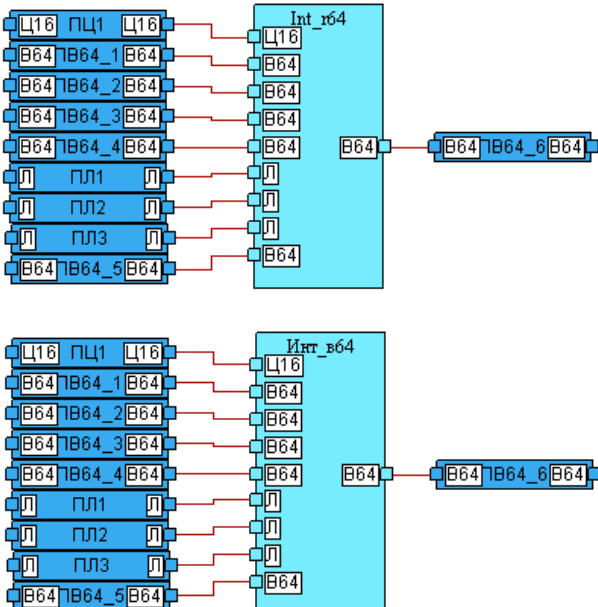
X5 - Максимальное значение интегрирования.

X6 - Флаг запрета интегрирования в сторону уменьшения.

X7 - Флаг запрета интегрирования в сторону увеличения.

X8 - Флаг сброса выходного значения в начальное значение.

X9 - Начальное значение

Пример вызова функции	
СТ:	ФБД:
<p>пв64_6=Int_r64(пц1, пв64_1, пв64_2, пв64_3, пв64_4, пл1, пл2, пл3, пв64_5)</p> <p>пв64_6 = Инт_в64(пц1, пв64_1, пв64_2, пв64_3, пв64_4, пл1, пл2, пл3, пв64_5)</p>	

Здесь пв64\_1, пв64\_2, пв64\_3, пв64\_4, пв64\_5, пв64\_6 – глобальные переменные вещественного 64-битного типа.

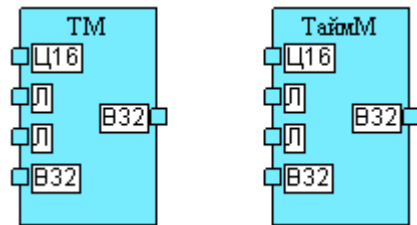


## 12 ТАЙМЕРЫ

### 12.1 ТаймМ, ТМ

#### Назначение

Счетчик минутный

Отображение	
СТ:	ФБД:
$Y = TM(X1, X2, X3, X4)$ $Y = ТаймМ(X1, X2, X3, X4)$	
Входные параметры: X1(ц16), X2(л), X3(л), X4(в32) Выходные параметры: Y(в32)	

#### Логика работы функции

X1 - порядковый номер счетчика (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - вход включения/выключения счетчика.

X3 - вход сброса счетчика.

На вход X4 подается начальное значение при сбросе счетчика.

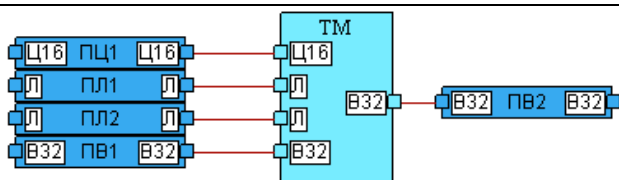
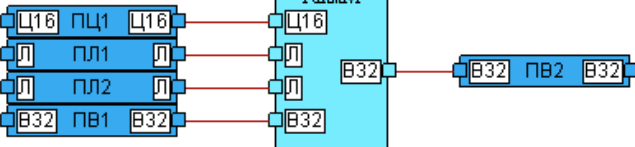
С выхода Y снимается значение счетчика.

Если X3 = 1 то Y = X4.

Если X3 = 0, X2 = 1 то Y = Y0 + Δ.

Если X3 = X2 = 0 то Y = Y0.

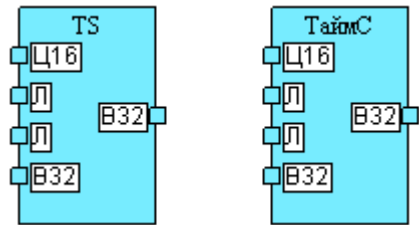
Δ – приращение времени в минутах (время от последнего вызова функции с этим номером).  
Y0 – предыдущее значение счетчика.

Пример вызова функции	
СТ:	ФБД:
$пв2 = TM(пц1, пл1, пл2, пв1)$	
$пв2 = ТаймМ(пц1, пл1, пл2, пв1)$	

12.2 ТаймС, TS

Назначение

Счетчик секундный.

Отображение	
СТ:	ФБД:
$Y = TS(X1, X2, X3, X4)$ $Y = \text{ТаймС}(X1, X2, X3, X4)$	
Входные параметры: X1(ц16), X2(л), X3(л), X4(в32) Выходные параметры: Y(в32)	

Логика работы функции

X1 - порядковый номер счетчика (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - вход включения/выключения счетчика.

X3 - вход сброса счетчика.

X4 - начальное значение при сбросе счетчика.

С выхода Y снимается значение счетчика.

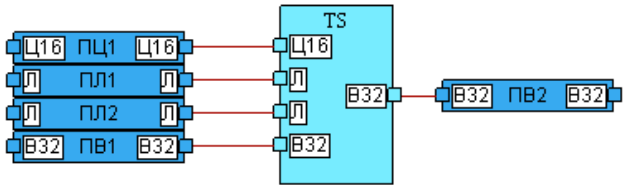
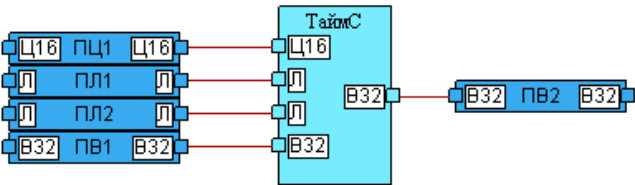
Если X3 = 1 то Y = X4.

Если X3 = 0, X2 = 1 то Y = Y0 + Δ.

Если X3 = X2 = 0 то Y = Y0.

Δ – приращение времени в секундах (время от последнего вызова функции с этим номером).

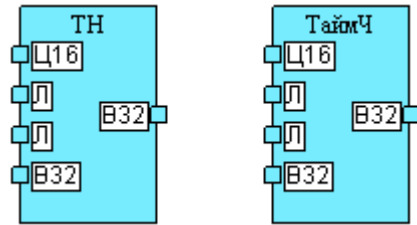
Y0 – предыдущее значение счетчика.

Пример вызова функции	
СТ:	ФБД:
$пв2 = TS(пц1, пл1, пл2, пв1)$	
$пв2 = \text{ТаймС}(пц1, пл1, пл2, пв1)$	

### 12.3 ТаймЧ, ТН

#### Назначение

Счетчик часовой

Отображение	
СТ:	ФБД:
$Y = TH(X1, X2, X3, X4)$  $Y = \text{ТаймЧ} (X1, X2, X3, X4)$	
Входные параметры: X1(ц16), X2(л), X3(л), X4(в32) Выходные параметры: Y(в32)	

#### Логика работы функции

X1 - порядковый номер счетчика (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - вход включения/выключения счетчика.

X3 - вход сброса счетчика.

На вход X4 подается начальное значение при сбросе счетчика.

С выхода Y снимается значение счетчика.

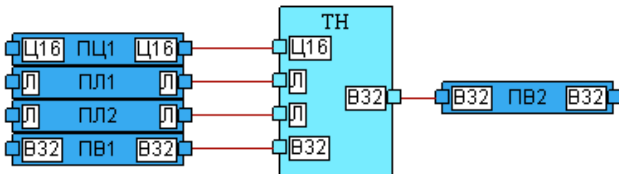
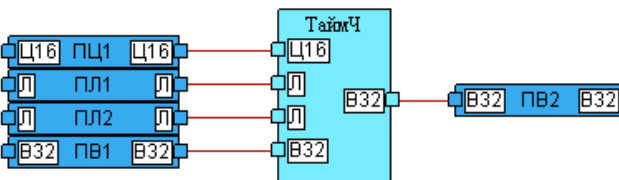
Если X3 = 1 то Y = X4.

Если X3 = 0, X2 = 1 то Y = Y0 + Δ.

Если X3 = X2 = 0 то Y = Y0.

Δ – приращение времени в часах (время от последнего вызова функции с этим номером).

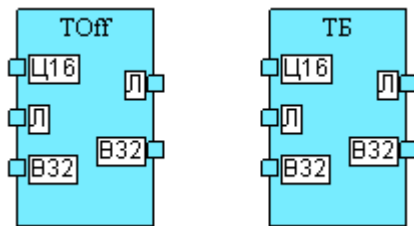
Y0 – предыдущее значение счетчика.

Пример вызова функции	
СТ:	ФБД:
$пв2 = TH(пц1, пл1, пл2, пв1)$	
$пв2 = \text{ТаймЧ}(пц1, пл1, пл2, пв1)$	

12.4 ТБ, TOff

Назначение

Таймер задержки на выключение.

Отображение	
СТ:	ФБД:
$(Y1, Y2) = TOff(X1, X2, X3)$ $(Y1, Y2) = ТБ(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(л), X3(в32) Выходные параметры: Y1(л), Y2(в32)	

Логика работы функции

Таймер задержки на выключение.

X1 - порядковый номер таймера (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - вход включения/выключения таймера.

X3 - вход задания времени задержки на выключение таймера, переменная T<sub>3</sub> (в интервале 0,01 – 42 949 671с).

Y1 – выходной сигнал.

Y2 - текущее значение счетчика (в интервале 0-T<sub>3</sub>).

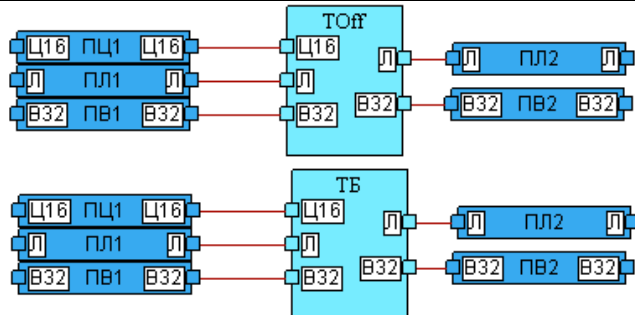
Если X2 = 0, то Y1 = 0. сброс счетчика времени.

Если X2 = 1, Y1 = 1.

По заднему фронту X2 - начало задержки.

Y1 = 1 до истечения времени T<sub>3</sub>, установленного на входе X3.

Если t = <T<sub>3</sub> то Y1 = 1. Y2 = t

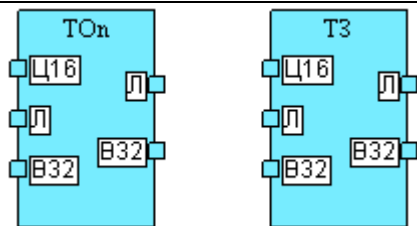
Пример вызова функции	
СТ:	ФБД:
$(пл2, пв2) = TOff(пц1, пл1, пв1)$ $(пл2, пв2) = ТБ(пц1, пл1, пв1)$	



## 12.5 T3, TOn

**Назначение**

Таймер задержки на включение.

Отображение	
СТ:	ФБД:
$(Y1, Y2) = TOn (X1, X2, X3)$ $(Y1, Y2) = T3 (X1, X2, X3)$	
Входные параметры: X1(ц16), X2(л), X3(в32) Выходные параметры: Y1(л), Y2(в32)	

**Логика работы функции**

X1 - порядковый номер таймера (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - вход включения/выключения таймера.

X3 - вход задания времени задержки на включение таймера, переменная T<sub>3</sub> (в интервале 0,01 – 42 949 671с).

Y1 – выходной сигнал.

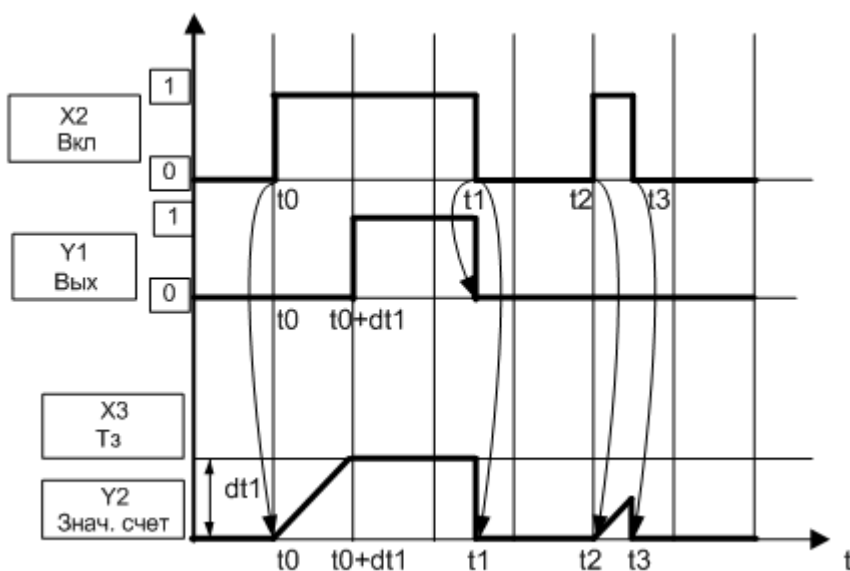
Y2 - текущее значение счетчика (в интервале 0-T<sub>3</sub>).

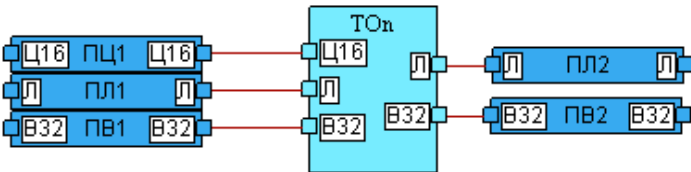
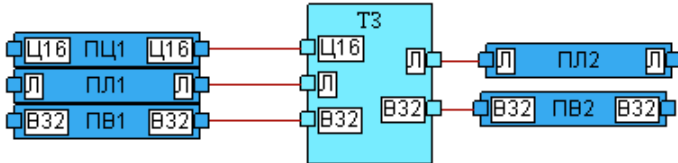
Если X2 = 0 то Y1 = 0 сброс счетчика времени. По переднему фронту X2 -начало задержки.

Y1 = 0 до истечения времени T<sub>3</sub>, установленного на входе X3. Если по истечении времени T<sub>3</sub> X2=1 то Y1 станет равным 1.

Если  $t = < T_3$  то Y1 = 0, Y2 = t.

Диаграмма работы таймера:



Пример вызова функции	
СТ:	ФБД:
(пл2, пв2) = ТОн(пц1, пл1, пв1)	
(пл2, пв2) = ТЗ(пц1, пл1, пв1)	

## 12.6 ТИЗ, ТР

### Назначение

Таймер импульса с задержкой.

### Отображение

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3) = TP(X1, X2, X3, X4)$	
$(Y1, Y2, Y3) = ТИЗ(X1, X2, X3, X4)$	
Входные параметры: X1(ц16), X2(л), X3(в32), X4(в32) Выходные параметры: Y1(л), Y2(в32), Y3(в32)	

### Логика работы функции

X1 – Порядковый номер таймера (от 1 до 20000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – «Вкл» - признак включения таймера.

X3 – «Длит\_Пауз» - длительность паузы.

X4 – «Длит\_имп» - длительность импульса.

Y1 – «Вых» - выходной сигнал таймера.

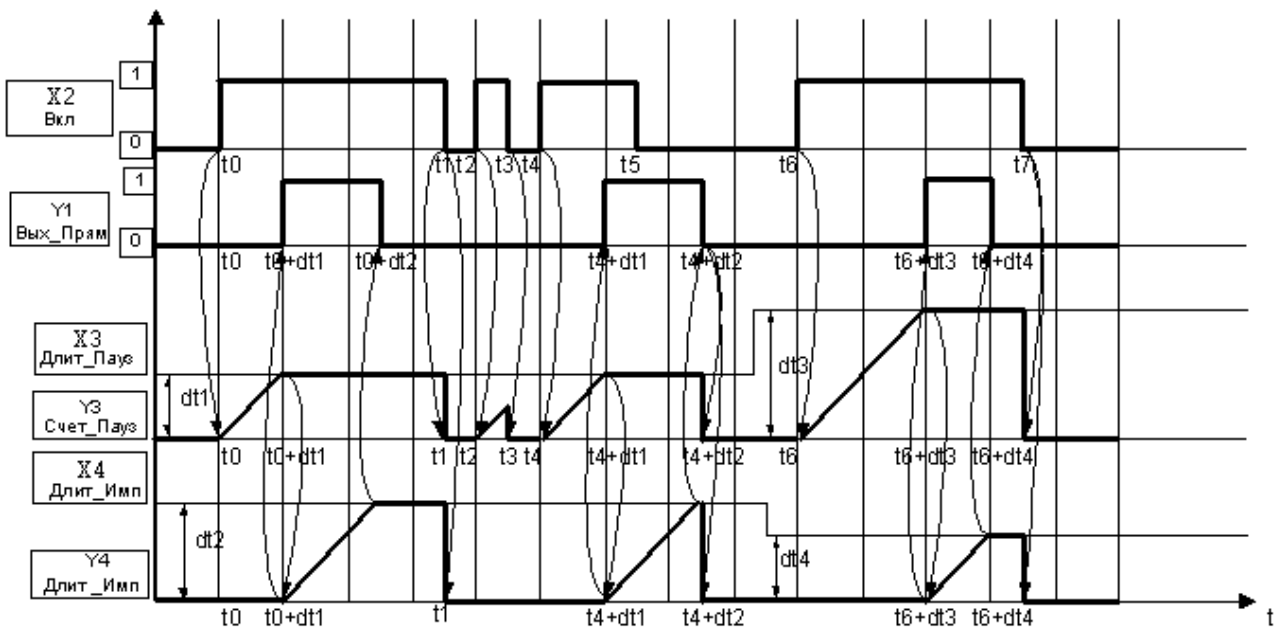
Y2 – «Счет\_Пауз» - счетчик времени задержки.

Y3 – «Счет\_Имп» - счетчик времени импульса.

При подаче на вход «Вкл» функции логической 1, через время задержки (заданное на входе «Длит\_Пауз») на выходе таймера «Вых», появляется логическая 1 и выдерживается в течении времени импульса (заданное на входе «Длит\_Имп»). Если вход «Вкл» сбросится в 0 во время задержки, то таймер сбрасывается (входной сигнал считается ложным). Если вход «Вкл» сбросится в 0 во время выдачи импульса, то таймер, независимо от этого выдаст заданный импульс. При подаче на вход «Вкл» функции логической 1 на выходе «Счет\_Пауз» отсчитывается время паузы и, после того, как значение на выходе «Счет\_Пауз» станет равно значению входа «Длит\_Пауз» отсчет паузы прекращается. После окончания отсчета паузы начинается отсчет импульса (выход «Счет\_Имп») и, после того, как значение на выходе «Счет\_Имп» станет равно значению входа «Длит\_Имп» отсчет импульса прекращается.

Если значение времени задержки, заданное на входе «Длит\_Пауз» = 0, то, при включении таймера, сигнал на выходе «Вых» появляется в текущем цикле. Если значение время импульса, заданное на входе «Длит\_Имп» = 0, то выдача импульса осуществляется на 1 цикл работы.

Диаграмма работы таймера.

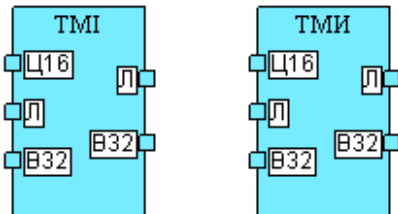


Пример вызова функции	
СТ:	ФБД:
(пл2, пв3, пв4) = ТР (пц1, пл1, пв1, пв 2)	
(пл2, пв3, пв4) = ТИЗ (пц1, пл1, пв1, пв 2)	

## 12.7 ТМИ, ТМИ

## Назначение

Таймер минутный импульса.

Отображение	
СТ:	ФБД:
$(Y1, Y2) = \text{ТМИ} (X1, X2, X3)$  $(Y1, Y2) = \text{ТМИ} (X1, X2, X3)$	
Входные параметры: X1(ц16), X2(л), X3(в32) Выходные параметры: Y1(л), Y2(в32)	

## Логика работы функции

X1 – Порядковый номер таймера (от 1 до 20000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – «Вкл» - признак включения таймера.

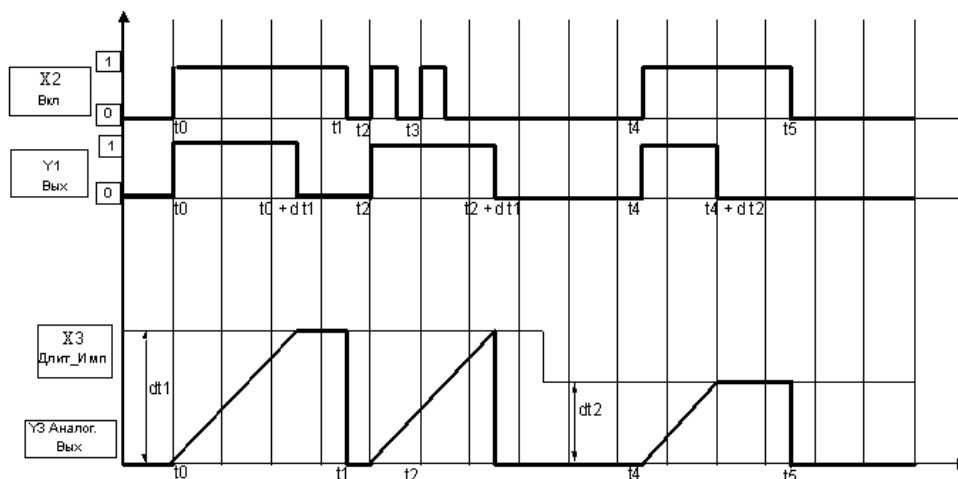
X3 – «Длит\_имп» - длительность импульса в минутах.

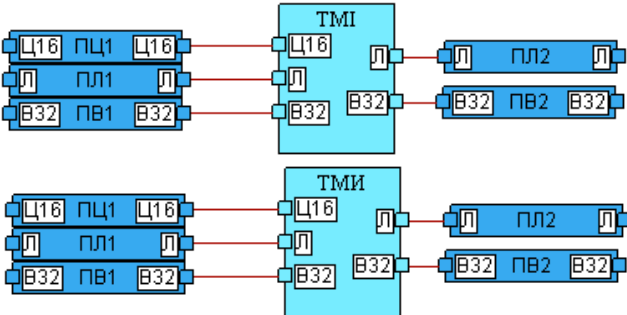
Y1 – «Вых» - логический выход таймера.

Y2 – «Счетчик» - счетчик таймера.

При подаче на вход «Вкл» логической "1", на выходе таймера «Вых», появляется "1". Таймер начинает отсчет времени и выдает его на выход «Счетчик». Пока не вышло время, установленное на входе «Длит\_Имп», выходной сигнал «Вых» не сбросится в 0, не зависимо от состояния входного сигнала «Вкл». Т.е. при включении таймера, он в любом случае выдаст импульс определенной длительности. При этом, после того, как таймер насчитает время, заданное на входе «Длит\_Имп», насчитанное значение будет сохранено на выходе «Счетчик» до момента, когда и входной сигнал «Вкл» станет равен 0. После этого выход «Счетчик» станет равен 0.

Диаграмма работы таймера.



Пример вызова функции	
СТ:	ФБД:
(пл2, пв2) = ТМИ (пц1, пл1, пв1)	
(пл2, пв2) = ТМИ (пц1, пл1, пв1)	

## 12.8 ТМП, ТМП

### Назначение

Таймер минутный программируемый.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3) = TMP(X1, X2, X3, X4, X5, X6, X7, X8)$  $(Y1, Y2, Y3) = ТМП(X1, X2, X3, X4, X5, X6, X7, X8)$	
Входные параметры: X1(ц16), X2(в32), X3(в32), X4(л), X5(л), X6(л), X7(в32), X8(л) Выходные параметры: Y1(л), Y2(л), Y3(в32)	

### Логика работы функции:

X1 - порядковый номер таймера (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – уставка на время задержки появления выходных сигналов таймера.

X3 – длительность импульса на выходах таймера, переменная  $T_3$  (в интервале 0,01 – 42 949 671с).

X4 – вход «ВКЛ»; приоритет низший.

X5 – вход «Выкл»; приоритет средний.

X6 – вход «Сброс»; приоритет высший.

X7 – значение внутреннего счетчика при сбросе.

X8 – «0/1» настройка X4, X5 на работу по «уровню / фронту».

Y1 – выходной сигнал.

Y2 – выходной сигнал. Инверсия Y1.

Y3 - текущее значение счетчика (в интервале 0- $T_3$ ).

Если X4=1, X5=0, X6=0, - включение счета.

Если X5=1, X6=0 – останов счетчика времени.

Если X6=1, установка внутреннего счетчика в состояние, соответствующее значению на входе X7.

При X2=0, задержки на включение нет.

Исходное состояние: Y1=0, Y2=1.

При  $t < t_1$ , Y1=0, Y2=1.

При  $t \geq t_1$ , Y1=1, Y2=0.

При  $t > t_2$ , Y1=0, Y2=1.

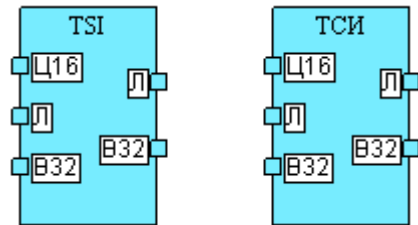
Пример вызова функции	
СТ:	ФБД:
<p>(пл5, пл6, пв4) = ТМР(пц1, пв1, пв2, пл1, пл2, пл3, пв3, пл4)</p> <p>(пл5, пл6, пв4) = ТМП(пц1, пв1, пв2, пл1, пл2, пл3, пв3, пл4)</p>	<p>The diagram illustrates the connection of two function blocks, ТМР and ТМП, to a set of logic elements. On the left, there are two columns of logic elements: the first column contains Ц16, В32, В32, Л, Л, Л, В32, Л; the second column contains ПЦ1, ПВ1, ПВ2, ПЛ1, ПЛ2, ПЛ3, ПВ3, ПЛ4. On the right, there are three logic elements: ПЛ5, ПЛ6, and В32 ПВ4 В32. The ТМР block has inputs connected to Ц16, В32, В32, Л, Л, Л, В32, Л and outputs connected to ПЛ5, ПЛ6, and В32 ПВ4 В32. The ТМП block has inputs connected to Ц16, В32, В32, Л, Л, Л, В32, Л and outputs connected to ПЛ5, ПЛ6, and В32 ПВ4 В32.</p>



## 12.9 ТСИ, TSI

### Назначение

Таймер секундный импульса.

Отображение	
СТ:	ФБД:
$(Y1, Y2) = TSI(X1, X2, X3)$	
$(Y1, Y2) = ТСИ(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(л), X3(в32) Выходные параметры: Y1(л), Y2(в32)	

### Логика работы функции

X1 – Порядковый номер таймера (от 1 до 20000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – «Вкл» - признак включения таймера.

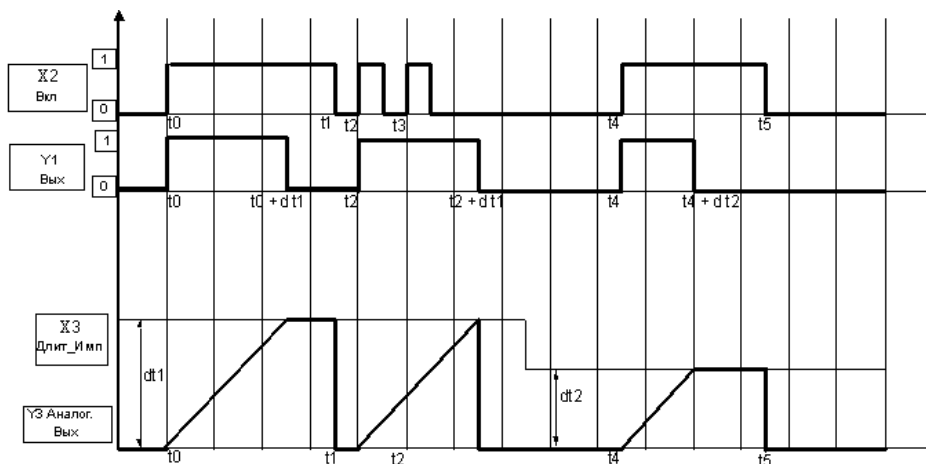
X3 – «Длит\_имп» - длительность импульса в секундах.

Y1 – «Вых» - логический выход таймера.

Y2 – «Счетчик» - счетчик таймера.

При подаче на вход «Вкл» логической "1", на выходе таймера «Вых», появляется "1". Таймер начинает отсчет времени и выдает его на выход «Счетчик». Пока не вышло время, установленное на входе «Длит\_Имп», выходной сигнал «Вых» не сбросится в 0, не зависимо от состояния входного сигнала «Вкл». Т.е. при включении таймера, он в любом случае выдаст импульс определенной длительности. При этом, после того, как таймер насчитает время, заданное на входе «Длит\_Имп», насчитанное значение будет сохранено на выходе «Счетчик» до момента, когда и входной сигнал «Вкл» станет равен 0. После этого выход «Счетчик» станет равен 0.

Диаграмма работы таймера.



Пример вызова функции	
СТ:	ФБД:
(пл2, пв2) = TSI (пц1, пл1, пв1)	
(пл2, пв2) = ТСИ (пц1, пл1, пв1)	

## 12.10 ТСП, TSP

**Назначение**

Таймер секундный программируемый.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3) = TSP(X1, X2, X3, X4, X5, X6, X7, X8)$  $(Y1, Y2, Y3) = ТСП(X1, X2, X3, X4, X5, X6, X7, X8)$	
Входные параметры: X1(ц16), X2(в32), X3(в32), X4(л), X5(л), X6(л), X7(в32), X8(л) Выходные параметры: Y1(л), Y2(л), Y3(в32)	

**Логика работы функции**

X1 – порядковый номер таймера (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – уставка времени задержки появления выходных сигналов таймера ( $t_1$ ).

X3 – длительность импульса на выходах таймера ( $t_2$ ).

X4 – вход «ВКЛ»; приоритет низший.

X5 – вход «Выкл»; приоритет средний.

X6 – вход «Сброс»; приоритет высший.

X7 – значение внутреннего счетчика при сбросе.

X8 – «0/1» настройка входов X4-X5 на работу по «уровню/фронту».

Y1 – выходной сигнал.

Y2 – выходной сигнал. Инверсия Y1.

Y3 - текущее значение счетчика.

Если X4 = 1, X5=0, X6=0, – включение счета.

Если X5 = 1, X6=0 – останов счетчика времени.

Если X6 = 1, установка внутреннего счетчика в состояние, соответствующее значению на входе X7.

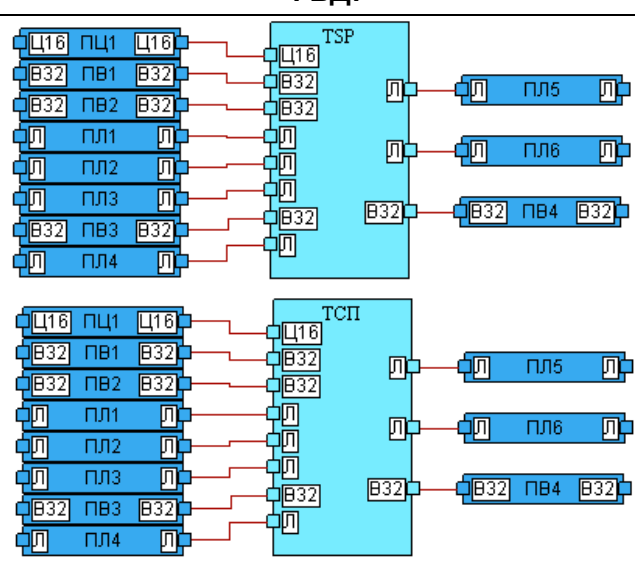
При X2 = 0 задержки на включение нет.

Исходное состояние: Y1=0, Y2=1.

При  $t < t_1$ , Y1=0, Y2=1.

При  $t \geq t_1$ , Y1=1, Y2=0.

При  $t > t_2$ , Y1=0, Y2=1.

Пример вызова функции	
СТ:	ФБД:
<p>(пл5, пл6, пв4) = TSP(пц1, пв1, пв2, пл1, пл2, пл3, пв3, пл4)</p> <p>(пл5, пл6, пв4) = ТСП(пц1, пв1, пв2, пл1, пл2, пл3, пв3, пл4)</p>	

## 12.11 ТЧИ, ТНІ

**Назначение**

Таймер часовой импульса.

Отображение	
СТ:	ФБД:
$(Y1, Y2) = \text{ТНІ} (X1, X2, X3)$ $(Y1, Y2) = \text{ТЧИ} (X1, X2, X3)$	
Входные параметры: X1(ц16), X2(л), X3(в32) Выходные параметры: Y1(л), Y2(в32)	

**Логика работы функции**

X1 – Порядковый номер таймера (от 1 до 20000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – «Вкл» - признак включения таймера.

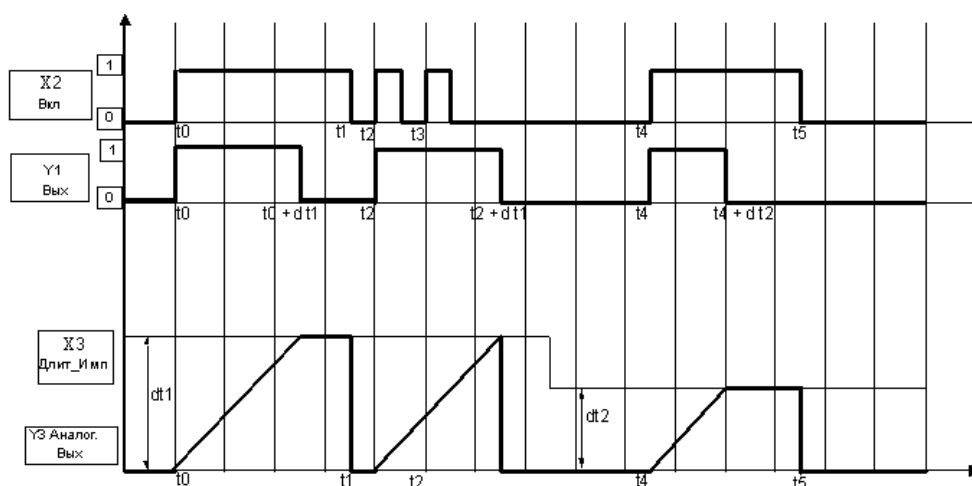
X3 – «Длит\_имп» - длительность импульса в часах.

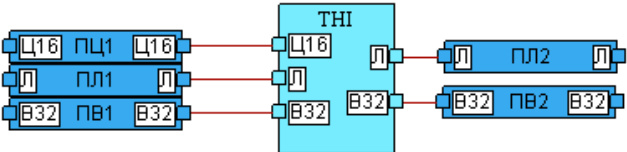
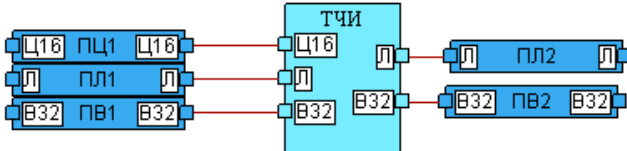
Y1 – «Вых» - логический выход таймера.

Y2 – «Счетчик» - счетчик таймера.

При подаче на вход «Вкл» логической "1", на выходе таймера «Вых», появляется "1". Таймер начинает отсчет времени и выдает его на выход «Счетчик». Пока не вышло время, установленное на входе «Длит\_Имп», выходной сигнал «Вых» не сбросится в 0, не зависимо от состояния входного сигнала «Вкл». Т.е. при включении таймера, он в любом случае выдаст импульс определенной длительности. При этом, после того, как таймер насчитает время, заданное на входе «Длит\_Имп», насчитанное значение будет сохранено на выходе «Счетчик» до момента, когда и входной сигнал «Вкл» станет равен 0. После этого выход «Счетчик» станет равен 0.

Диаграмма работы таймера.

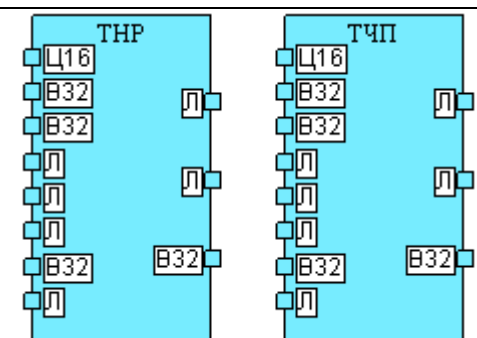


Пример вызова функции	
СТ:	ФБД:
(пл2, пв2) = ТНІ (пц1,пл1, пв1)	 <p>The diagram shows the SFC for the TNI function. On the left, three input blocks are stacked: the top one has Ц16, ПЦ1, Ц16; the middle one has Л, ПЛ1, Л; the bottom one has В32, ПВ1, В32. Red lines connect these to the TNI block's inputs: Ц16, Л, and В32. The TNI block has two outputs: Л and В32. These connect to two output blocks on the right: the top one has Л, ПЛ2, Л; the bottom one has В32, ПВ2, В32.</p>
(пл2, пв2) = ТЧИ (пц1,пл1, пв1)	 <p>The diagram shows the SFC for the TCHI function. It is identical to the TNI diagram, with the central block labeled ТЧИ instead of ТНІ.</p>

## 12.12 ТЧП, ТНР

### Назначение

Таймер часовой программируемый.

Отображение	
СТ:	ФБД:
<p>(Y1, Y2, Y3) = ТНР (X1, X2, X3, X4, X5, X6, X7, X8)</p> <p>(Y1, Y2, Y3) = ТЧП (X1, X2, X3, X4, X5, X6, X7, X8)</p>	
<p>Входные параметры: X1(ц16), X2(в32), X3(в32), X4(л), X5(л), X6(л), X7(в32), X8(л)</p> <p>Выходные параметры: Y1(л), Y2(л), Y3(в32)</p>	

### Логика работы функции

X1 - порядковый номер таймера (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – уставка на время задержки появления выходных сигналов таймера.

X3 – длительность импульса на выходах таймера, переменная  $T_3$  (в интервале 0,01 – 42 949 671с).

X4 – вход «ВКЛ»; приоритет низший.

X5 – вход «Выкл»; приоритет средний.

X6 – вход «Сброс»; приоритет высший.

X7 – значение внутреннего счетчика при сбросе.

X8 – «0/1» настройка X4-X5 на работу по «уровню/фронту».

Y1 – выходной сигнал.

Y2 – выходной сигнал. Инверсия Y1.

Y3 - текущее значение счетчика (в интервале 0- $T_3$ ).

Если X4 = 1, X5=0, X6=0 - включение счета.

Если X5=1, X6=0 – останов счетчика времени.

Если X6=1, установка внутреннего счетчика в состояние, соответствующее значению на входе X7.

При X2 =0 задержки на включение нет.

Исходное состояние: Y1=0, Y2=1.

При  $t < t_1$ , Y1=0, Y2=1.

При  $t \geq t_1$ , Y1=1, Y2=0.

При  $t > t_2$ , Y1=0, Y2=1.

Пример вызова функции	
СТ:	ФБД:
(пл5, пл6, пв4) = ТНР (пц1, пв1, пв2, пл1, пл2, пл3, пв3, пл4)	
(пл5, пл6, пв4) = ТЧП(пц1, пв1, пв2, пл1, пл2, пл3, пв3, пл4)	

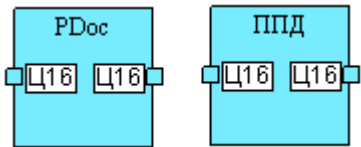


## 13 ПЕЧАТНЫЕ ДОКУМЕНТЫ

### 13.1 ППД, PDoc

#### Назначение

Функция подготовки печатного документа.

Отображение	
СТ:	ФБД:
$Y = PDoc(X)$ $Y = ППД(X)$	
Входные параметры: X1(ц16) Выходные параметры: Y(ц16)	

#### Описание

Данная функция не применяется на платформах СРВК.

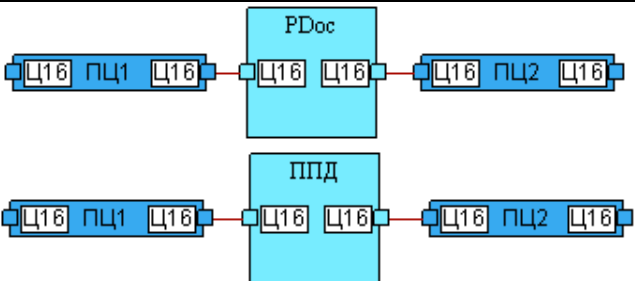
#### Логика работы функции

X – целое число, указывающее на номер подготавливаемого документа в БД.

Y – результат выполнения документа.

Если Y = 0 - успешно.

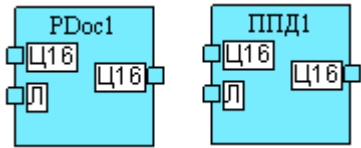
Если Y отлично от 0 - ошибка.

Пример вызова функции	
СТ:	ФБД:
$пц2 = PDoc(пц1)$ $пц2 = ППД(пц1)$	

## 13.2 ППД1, PDoc1

### Назначение

Функция подготовки печатного документа с условием.

Отображение	
СТ:	ФБД:
$Y = PDoc1(X1, X2)$ $Y = ППД1(X1, X2)$	
Входные параметры: X1(ц16), X2(л) Выходные параметры: Y(ц16)	

### Описание

Данная функция не применяется на платформах СРВК.

### Логика работы функции

X1 –целое число, указывающее на номер подготавливаемого документа в БД.

X2 – исполнение функции (логическая переменная).

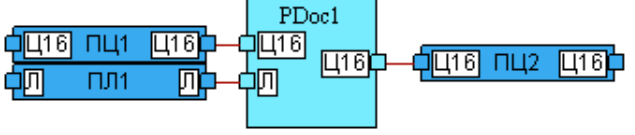
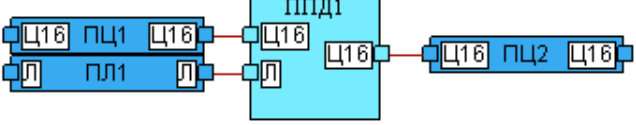
Если X2 =0, нет подготовки печатного документа.

Если X2 =1, подготовка печатного документа при каждом проходе программы пользователя.

Y – результат выполнения документа.

Если Y =0 - успешно.

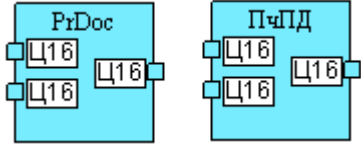
Если Y отлично от 0 - ошибка.

Пример вызова функции	
СТ:	ФБД:
$пц2 = PDoc1(пц1, пл1)$	
$пц2 = ППД1(пц1, пл1)$	

### 13.3 ПчПД, PrDoc

#### Назначение

Функция печати печатного документа.

Отображение	
СТ:	ФБД:
$Y = \text{PrDoc}(X1, X2)$ $Y = \text{ПчПД}(X1, X2)$	
Входные параметры: X1(ц16), X2(ц16) Выходные параметры: Y(ц16)	

#### Описание

Данная функция не применяется на платформах СРВК.

#### Логика работы функции

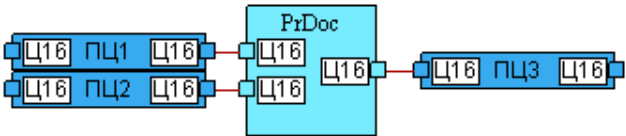
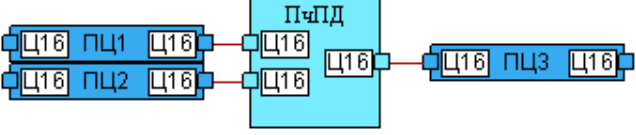
X1 – целое число, указывающее на номер печатаемого документа в БД.

X2 – целое число, указывающее смещение в истории для данного документа.

Y – результат выполнения документа.

Если Y = 0 – успешно.

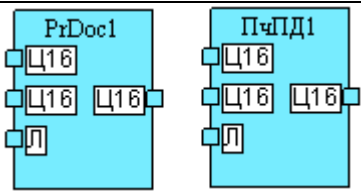
Если Y отлично от 0 - ошибка.

Пример вызова функции	
СТ:	ФБД:
$\text{пц3} = \text{PrDoc}(\text{пц1}, \text{пц2})$	
$\text{пц3} = \text{ПчПД}(\text{пц1}, \text{пц2})$	

### 13.4 ПчПД1, PrDoc1

#### Назначение

Функция печати печатного документа с условием.

Отображение	
СТ:	ФБД:
$Y = PrDoc1(X1, X2, X3)$ $Y = ПчПД1 (X1, X2, X3)$	
Входные параметры: X1(ц16), X2(ц16), X3(л) Выходные параметры: Y(ц16)	

#### Описание

Данная функция не применяется на платформах СРВК.

#### Логика работы функции

X1 – целое число, указывающее на номер печатаемого документа в БД.

X2 – целое число, указывающее смещение в истории для данного документа.

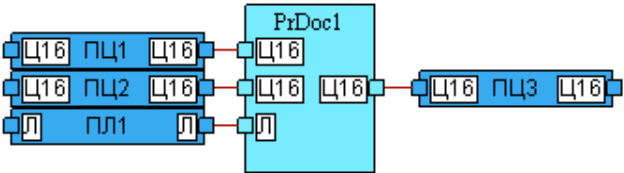
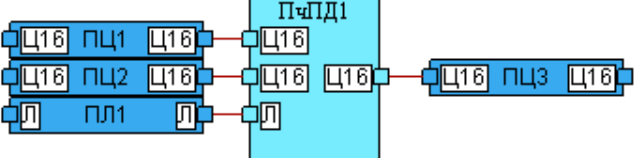
X3 - разрешение выдачи.

При X3=1 формируется команда выдачи на печать при каждом проходе программы пользователя.

Y – результат выполнения документа.

Если Y =0 – успешно.

Если Y отлично от 0 - ошибка.

Пример вызова функции	
СТ:	ФБД:
$пц3 = PrDoc1 (пц1, пц2, пл1)$	
$пц3 = ПчПД1 (пц1, пц2, пл1)$	

**13.5 ППД ФВ, PDoc\_FT****Назначение**

Функция обеспечивает подготовку печатных документов с определенной датой и временем.

Отображение	
СТ:	ФБД:
$Y = \text{pdoc\_ft}(X1, X2, X3, X4)$ $Y = \text{ппд\_фв}(X1, X2, X3, X4)$	
Входные параметры: X1(ц32), X2(в32), X3(в32), X4(л) Выходные параметры: Y(ц16)	

**Описание**

Данная функция не применяется на платформах СРВК.

**Логика работы функции**

X1 – целое число, указывающее на номер подготавливаемого документа в БД.

X2 – вещественное число, содержащее в себе дату ДДММГГ.

X3 – вещественное число, содержащее в себе время ЧЧММСС.

X4 – исполнение функции (логическая переменная).

Если X4 = 0, нет подготовки печатного документа.

Если X4 = 1, подготовка печатного документа при каждом проходе программы пользователя.

Y – результат выполнения документа.

Если Y = 0 - успешно.

Если Y отлично от 0 - ошибка.

Пример вызова функции	
СТ:	ФБД:
$\text{пц2} = \text{PPD\_FT}(\text{IdDoc}, \text{date}, \text{time}, \text{flag})$ $\text{пц2} = \text{ППД\_ФВ}(\text{IdDoc}, \text{date}, \text{time}, \text{flag})$	

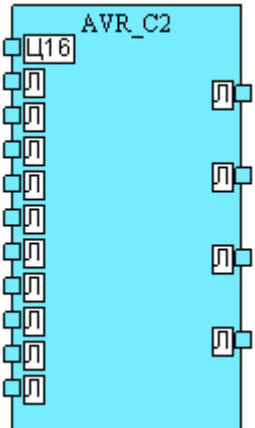


## 14 ОБЩЕСИСТЕМНЫЕ

### 14.1 AVR\_C2

#### Назначение

Автоматическое включение резерва двух сетевых насосов.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4) = AVR\_C2(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11)$	
<p>Входные параметры: X1(ц16), X2(л), X3(л), X4(л), X5(л), X6(л), X7(л), X8(л), X9(л), X10(л), X11(л)</p> <p>Выходные параметры: Y1(л), Y2(л), Y3(л), Y4(л)</p>	

#### Описание

**AVR\_C2** – функция предназначена для автоматического включения резервного сетевого насоса, находящегося в рабочем состоянии, при снижении давления напора работающего насоса или аварийном отключении другого насоса.

Функция не применяется для платформ CPBK версии 6.5 для ОС QNX и LINUX и среда исполнения КРУГОЛ любой версии.

#### Логика работы функции

X1 – порядковый номер функции (от 1 до 20 000). Для платформ CPBK, начиная с версии 7.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – сигнал "перевод 1-го насоса в рабочее положение"

X3 – сигнал "снижение давления на 1-ом насосе"

X4 – сигнал "состояние 1-го насоса"

X5 – сигнал "аварийное отключение 1-го насоса"

X6 – сигнал "перевод 1-го насоса в ремонтное положение"

X7 – сигнал "перевод 2-го насоса в рабочее положение"

X8 – сигнал "снижение давления на 2-ом насосе"

X9 – сигнал "состояние 2-го насоса"

X10 – сигнал "аварийное отключение 2-го насоса"

X11 – сигнал "перевод 2-го насоса в ремонтное положение"

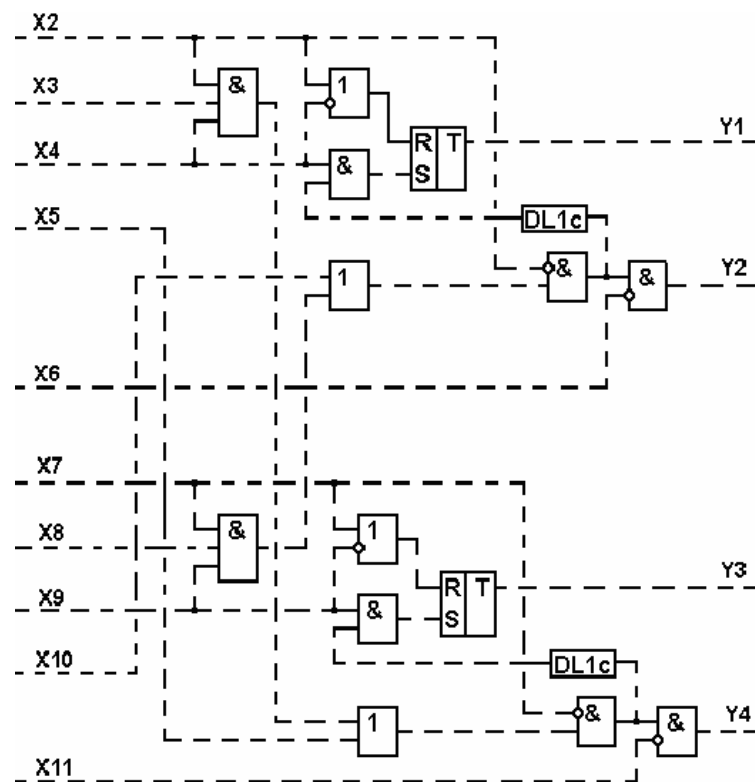
Y1 – сигнал "сигнализация включения 1-го насоса"

Y2 – сигнал "включить 1-ый насос"

ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Y3 – сигнал "сигнализация включения 2-го насоса"  
Y4 – сигнал "включить 2-ой насос"

Выполняемая функция алгоритма представлена на следующей структурной схеме:



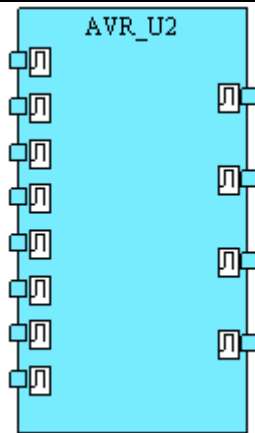
Пример вызова функции	
СТ:	ФБД:
<p>(пл11, пл12, пл13, пл14) = AVR_C2(пц1, пл1, пл2, пл3, пл4, пл5, пл6, пл7, пл8, пл9, пл10)</p>	



## 14.2 AVR\_U2

### Назначение

Автоматическое включение резерва двух насосов по уровню.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4) = AVR\_U2(X1, X2, X3, X4, X5, X6, X7, X8)$	
Входные параметры: X1(л), X2(л), X3(л), X4(л), X5(л), X6(л), X7(л), X8(л) Выходные параметры: Y1(л), Y2(л), Y3(л), Y4(л)	

### Описание

**AVR\_U2** – функция предназначена для переключения двух насосов, находящихся в исправном состоянии, в автоматическом режиме работы по дискретным сигналам уровня.

Функция не применяется для платформ СРВК версии 6.5 для ОС QNX и LINUX и среда исполнения КРУГОЛ любой версии.

### Логика работы функции

X1 – сигнал "перевод 1-го насоса в рабочее положение"  
X2 – сигнал "перевод 1-го насоса в ремонтное положение"  
X3 – сигнал "перевод 2-го насоса в рабочее положение"  
X4 – сигнал "перевод 2-го насоса в ремонтное положение"  
X5 – сигнал "перевод насосов в автоматический режим работы"  
X6 – сигнал "достижение нижнего уровня"  
X7 – сигнал "достижение первого уровня"  
X8 – сигнал "достижение второго уровня"

Y1 – сигнал "включить 1-ый насос"  
Y2 – сигнал "отключить 1-ый насос"  
Y3 – сигнал "включить 2-ой насос"  
Y4 – сигнал "отключить 2-ой насос"

При получении сигнала "перевод насосов в автоматический режим работы" (X5 = TRUE) функция предусматривает:

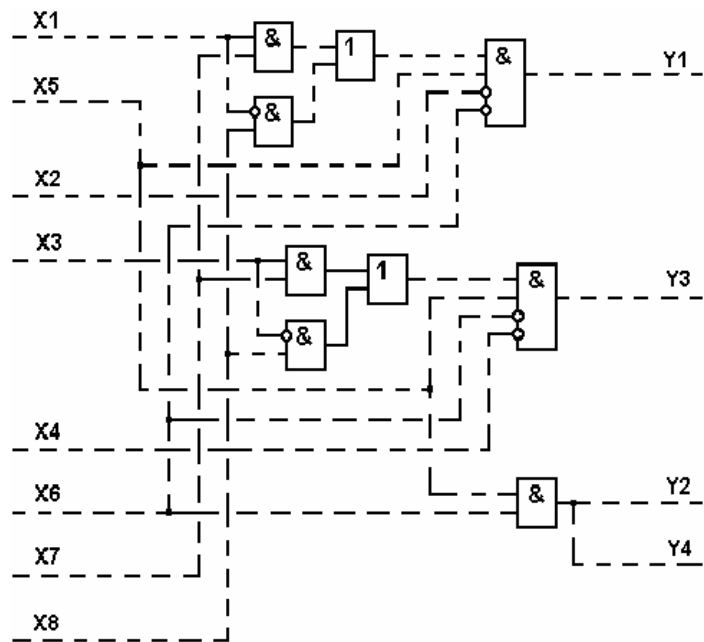
- включение насоса, выбранного рабочим (X1 или X3 = TRUE), по достижению первого уровня дренажа в баке (X7 = TRUE);

ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

- включение насоса, выбранного резервным (X1 или X3 = FALSE), по достижению второго уровня дренажа в баке (X8 = TRUE);
- отключение всех насосов при получении сигнала нижнего уровня дренажа в баке (X6 = TRUE).

При снятии сигнала "перевод насосов в автоматический режим работы" (X5 = FALSE) управление насосами переводится в режим дистанционного управления, при этом функция устанавливает все выходные сигналы в состояние FALSE, независимо от состояния других входных сигналов.

Логическая схема алгоритма представлена на следующем рисунке:

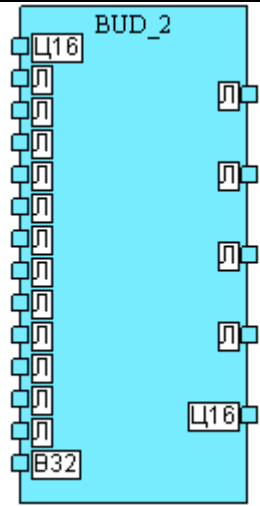


Пример вызова функции		
СТ:	ФБД:	
(пл9, пл10, пл11, пл12) = AVR_U2(пл1, пл2, пл3, пл4, пл5, пл6, пл7, пл8)		

### 14.3 BUD\_2

#### Назначение

Управление электродвигателем.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4, Y5) = BUD\_2(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14)$	
<p>Входные параметры: X1(ц16), X2(л), X3(л), X4(л), X5(л), X6(л), X7(л), X8(л), X9(л), X10(л), X11(л), X12(л), X13(л), X14(в32)</p> <p>Выходные параметры: Y1(л), Y2(л), Y3(л), Y4(л), Y5(ц16)</p>	

#### Описание

**BUD\_2** – функция предназначена для управления электродвигателем. Функция формирует коды состояния двигателя, передаваемые на операторский уровень.

Функция не применяется для «старых» платформ (СРВК версии 6.5 для ОС QNX и LINUX и среда исполнения версии 1.0).

#### Логика работы функции

X1 – порядковый номер функции (от 1 до 20 000). Для платформ, начиная с "СРВК 7.1" и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – команда "ВКЛЮЧИТЬ" (ДУ)

X3 – команда "ОТКЛЮЧИТЬ" (ДУ)

X4 – команда "ВКЛЮЧИТЬ" (АУ)

X5 – команда "ОТКЛЮЧИТЬ" (АУ)

X6 – команда "ВКЛЮЧИТЬ" (БЛ)

X7 – команда "ОТКЛЮЧИТЬ" (БЛ)

X8 – разрешение включения ДУ и АУ

X9 – разрешение отключения ДУ и АУ

X10 – разрешение выдачи выходных команд

X11 – разрешение включения АУ

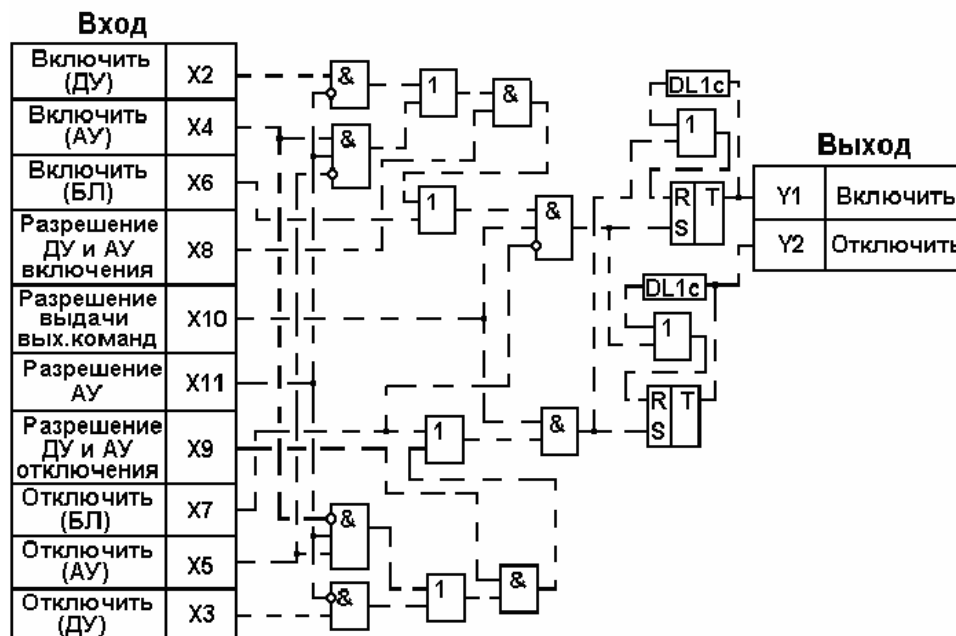
X12 – сигнал "ДВИГАТЕЛЬ ВКЛЮЧЕН"

X13 – сигнал "ДВИГАТЕЛЬ ОТКЛЮЧЕН"

X14 – время выдержки на срабатывание схем контроля выполнения команды управления двигателем

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Y1 – команда "ВКЛЮЧИТЬ ДВИГАТЕЛЬ"  
Y2 – команда "ОТКЛЮЧИТЬ ДВИГАТЕЛЬ"  
Y3 – сигнал "РЕМОНТ"  
Y4 – сигнал "ДВИГАТЕЛЬ АВАРИЙНО ОТКЛЮЧИЛСЯ"  
Y5 – состояние двигателя



Функция BUD\_2 формирует выходные команды "ВКЛЮЧИТЬ ДВИГАТЕЛЬ" (Y1) и "ОТКЛЮЧИТЬ ДВИГАТЕЛЬ" (Y2) длительностью не менее 1 секунды (зависит от длительности входных команд). Алгоритм формирования выходных команд представлен на рисунке. Длительность входных команд может быть различной, но не меньше задаваемого цикла решения задачи управления.

Команда "ОТКЛЮЧИТЬ" (БЛ) имеет более высокий приоритет перед командами на включение, поэтому при поступлении команды "ОТКЛЮЧИТЬ" (БЛ) блокируются любые команды на включение. В случае поступления на вход одного из сигналов дистанционного управления (X2, X3), он в течение одного цикла работы функции участвует в формировании выходных команд, а затем сбрасывается в нулевое состояние. Длительность соответствующей выходной команды равна 1 секунде.

Функция одновременно формирует только одну из команд "ВКЛЮЧИТЬ ДВИГАТЕЛЬ" (Y1) или "ОТКЛЮЧИТЬ ДВИГАТЕЛЬ" (Y2). При поступлении на вход функции сигналов на формирование обеих команд, они взаимно блокируются и ни одна команда не выдается.

Выходному сигналу "РЕМОНТ" (Y3) присваивается значение TRUE и соответствующий код состояния двигателя XB07 (см. таблицу "Содержимое кода состояния двигателя"), если разрешение выдачи выходных команд (X10) равен FALSE и сигнал "ДВИГАТЕЛЬ ОТКЛЮЧЕН" (X13) равен TRUE.

Логика работы алгоритма определения команды на включение и отключение двигателя следующая: первая пришедшая команда «ВКЛЮЧИТЬ» или «ОТКЛЮЧИТЬ» считается первопричиной и запоминается. После подтверждения исполнения команды (появления соответствующего сигнала состояния двигателя) формируется код состояния двигателя (Y5) – двигатель включен или отключен по такой-то причине. Если двигатель включен, а команд

на включение двигателя не поступало, формируется состояние "Двигатель включен по месту". Если двигатель отключен, а команд на отключение и включение двигателя не поступало, формируется состояние "Двигатель отключен по месту". Алгоритм приводится в исходное состояние после смены состояния двигателя.

Логика работы алгоритма контроля выполнения команды на включение двигателя следующая: после выдачи команды "ВКЛЮЧИТЬ" команда запоминается.

Если в течение контрольного интервала времени (X14) не придет подтверждение выполнения команды, т.е. X13 = TRUE, то запрещается выполнение алгоритма формирования сигнала аварийного отключения и формируется код состояния двигателя (Y5) - "Команда включить не выполнена".

Если в течение контрольного интервала времени (X14) приходит подтверждение выполнения команды, т.е. X13 = FALSE, то разрешается выполнение алгоритма формирование аварийного отключения.

Логика работы алгоритма контроля выполнения команды на отключение двигателя следующая: после выдачи команды "ОТКЛЮЧИТЬ" команда запоминается.

Если в течение контрольного интервала времени (X14) не придет подтверждение выполнения команды, т.е. X12 = TRUE, то формируется код состояния двигателя (Y5) - "Команда отключить не выполнена".

Если в течение контрольного интервала времени (X14) приходит подтверждение выполнения команды, т.е. X12 = FALSE, то алгоритм контроля выполнения команды на отключение возвращается в исходное состояние.

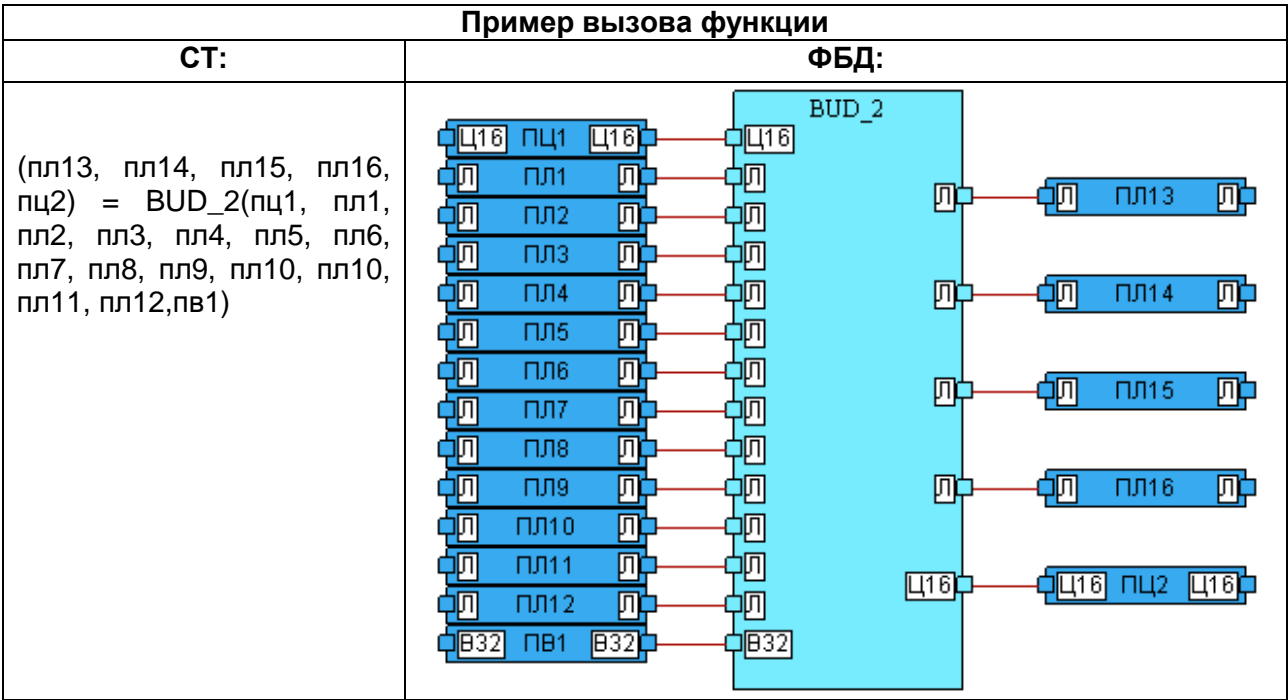
Логика работы алгоритма формирования сигнала аварийного отключения следующая: после получения признака разрешения от алгоритма контроля выполнения команды на включение, алгоритм запускается в работу. В случае появления сигнала состояния двигателя X13 = TRUE и при отсутствии команды на выходе функции "ОТКЛЮЧИТЬ ДВИГАТЕЛЬ" (Y2) - формируется выходной сигнала – Y4 = TRUE и код состояния двигателя (Y5) – "Двигатель аварийно отключился".

Алгоритм возвращается в исходное состояние после выдачи команды "ОТКЛЮЧИТЬ ДВИГАТЕЛЬ" (Y2).

Контроль достоверности: если в течение интервала времени большего 1с X13 = X12, то формируется код недостоверного состояния двигателя.

Кодировка состояний двигателя (Y5):

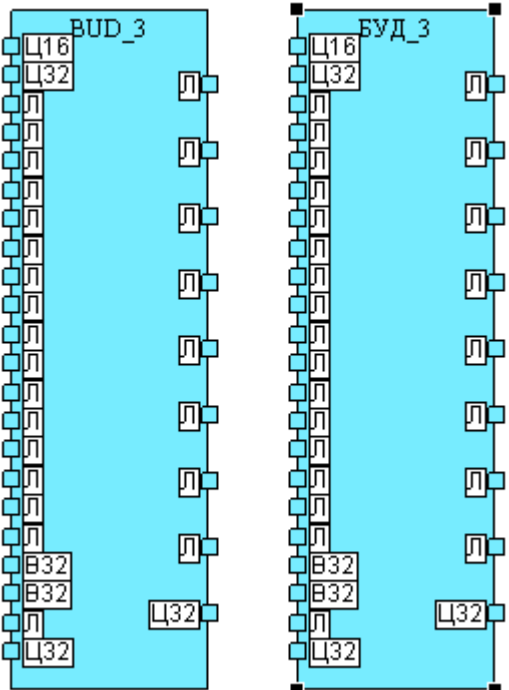
Состояния двигателя передаваемые на операторский уровень	Код
Двигатель включен оператором	0x1
Двигатель включен по автоматике	0x2
Двигатель включен по блокировке с запретом на отключение	0x4
Двигатель включен по месту	0x8
Двигатель отключен оператором	0x10
Двигатель отключен по автоматике	0x20
Двигатель отключен по блокировке с запретом на включение	0x40
Двигатель отключен по месту	0x80
Двигатель аварийно отключился	0x100
Двигатель в ремонте	0x200
Команда на включение не выполнена	0x400
Команда на отключение не выполнена	0x800
Недостоверность состояния двигателя	0x1000



## 14.4 БУД\_3, BUD\_3

**Назначение**

Функция управления электрифицированным двигателем.

Отображение	
СТ:	ФБД:
<p>(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9) = BUD_3(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20, X21, X22)</p> <p>(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9) =БУД_3 (X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20, X21, X22)</p>	
<p><b>Входные параметры:</b> X1(Ц16), X2 (Ц32), X3(л), X4(л), X5(л), X6(л), X7(л), X8(л), X9(л), X10(л), X11(л), X12(л), X13(л), X14(л), X15(л), X16(л), X17(л), X18(л), X19(В32), X20(В32), X21 (л), X22(Ц32).</p> <p><b>Выходные параметры:</b> Y1(л), Y2(л), Y3(л), Y4(л), Y5(л), Y6(л), Y7(л), Y8(л), Y9(Ц32).</p>	

**Описание**

Применяется только для платформ СРВК версии 8.0 и выше.

**Логика работы функции**

Для корректной работы функции необходимо, чтобы:

- На входе X12 и выходе Y5 (КО «Ремонт») была одна и та же переменная;
- На входе X13 и выходе Y6 (КО «Резерв») была одна и та же переменная;
- На входе X14 и выходе Y7 (КО «Работа») была одна и та же переменная;
- На входе X15 и выходе Y2 (КО «Включить») была одна и та же переменная;
- На входе X16 и выходе Y4 (КО «Отключить») была одна и та же переменная;
- На входе X22 и выходе Y9 (Состояние двигателя) была одна и та же переменная;

Если сброс сигнализации происходит по команде оператора (КО) «Сброс сигнализации», то на входе X21 и выходе Y8 (Сброс сигнализации) должна быть одна и та же переменная. Если сброс осуществляется по условию, то выход Y8 может быть не привязан к переменной.

**ВАЖНО.** Значение переменной "Состояние двигателя" (вход X22 и выход Y9) должно храниться между циклами контроллера. Для этого с помощью функций побитового копирования (BCIF,BCFI) необходимо записать выход Y9 в переменную Базы Данных контроллера вещественного типа (например, PB).

### Приоритет входных команд управления

Для команд, имеющих одинаковый приоритет, команда на отключение имеет преимущество (например, при одновременной подаче команд KO\_ON и KO\_OFF будет выполняться KO\_OFF).

1. SMA (двигатель аварийно отключен) – наивысший приоритет.
2. OFFLZ (Отключить ЛЗ) и ONLZ (Включить ЛЗ).
3. MU\_OFF (Отключить МУ) и MU\_ON (Включить МУ).
4. KO\_OFF (Отключить командой оператора) и KO\_ON (Включить командой оператора).

**ВАЖНО.** Все команды на включение игнорируются при наличии признака «Нет питания».

**ВАЖНО.** КО «Включить» игнорируется при наличии признака «Неисправность KB».

### Входные параметры:

**X1 – Номер.** Порядковый номер функции (от 1 до 20000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

**X2 – OPS.** Описание признаков настройки:

**0<sup>ой</sup> бит OPS** – указывает способ определения состояния двигателя по 2 концевым выключателям (KB) или по 1 KB и обработке входа X7 «Двигатель включен» и входа X8 «Двигатель отключен».

При **единичном** значении бита обрабатываются оба входа и состояние двигателя «Включен/Отключен» определяется по 2 концевым выключателям.

При **нулевом** значении бита обрабатывается только вход X7 «Двигатель включен» и состояние двигателя определяется только по состоянию данного входа. При этом значение входа X8 игнорируется.

**1<sup>ый</sup> бит OPS** – указывает способ формирования признаков «Нет питания» и «Неисправность KB» и обработку входа X9 «Нет питания».

**ВАЖНО.** При формировании признаков «Нет питания» и «Неисправность KB» учитываются значения **2<sup>ого</sup>** и **17<sup>ого</sup>** бита OPS и входа X4 «Режим «МУ».

При **единичном** значении бита вход X9 «Нет питания» обрабатывается функцией и признак «Нет питания» равен сигналу «Нет питания». При этом признак «Неисправность KB» формируется, если значение входов X7 «Двигатель включен» и X8 «Двигатель отключен» равны.



**ВАЖНО.** Формирование признака «Неисправность КВ» осуществляется, если 0<sup>ой</sup> бит OPS равен 1, иначе признак «Неисправность КВ» не выставляется.

При нулевом значении бита вход Х9 «Нет питания» игнорируется и признак «Нет питания» определяется по состоянию входов Х7 «Двигатель включен» и 8 «Двигатель отключен» (если оба входа равны 0, иначе не формируется).

**ВАЖНО.** Формирование признака «Нет питания» осуществляется, если 0<sup>ой</sup> бит OPS равен 1, иначе признак «Нет питания» не формируется.

**ВАЖНО.** Формирование признака «Неисправность КВ» осуществляется, если 0<sup>ой</sup> бит OPS равен 1 (значение входов Х7 «Двигатель включен» и Х8 «Двигатель отключен» равны 1), иначе признак «Неисправность КВ» не выставляется.

2<sup>ой</sup> бит OPS – указывает способ формирования признаков «Нет питания» и «Неисправность КВ» и реакцию функции на совместное появление единичных значений на входах Х4 «Режим МУ» и Х5 «МУ «Включить» или входах Х4 «Режим МУ» и Х6 «МУ «Отключить».

**ВАЖНО.** При формировании признаков «Нет питания» и «Неисправность КВ» учитывается значение 17<sup>ого</sup> бита OPS.

При единичном значении бита управление при МУ (вход Х4 равен 1) осуществляется через функцию и обрабатываются входы Х5 «МУ «Включить» и Х6 «МУ «Отключить». При этом, прохождение управляющих команд от оператора блокируется (игнорируются входы Х15 «КО Включить» и Х16 «КО Отключить»).

**ВАЖНО.** При поступлении сигнала МУ (вход Х4) физические команды, выдаваемые из функции и запущенные до появления сигнала МУ, остаются до истечения времени выдачи выходных команд (вход Х20) и только по завершении всех управляющих воздействий выставляется признак МУ. После выставления признака МУ состояние двигателя меняется на «Работа».

При нулевом значении бита управление при МУ осуществляется помимо контроллера. При этом, при появлении единичного значения на входе Х5 «МУ «Включить» или Х6 «МУ «Отключить» управляющее воздействие из функции не выдается, а запускается контроль за выполнением команды. При наличии сигнала МУ (вход Х4 равен 1) прохождение управляющих команд от оператора блокируется (игнорируются входы Х15 «КО Включить» и Х16 «КО Отключить»).

Также при наличии сигнала МУ не производится формирование признака «Неисправность КВ». Формирование признака «Нет питания» зависит от 1<sup>ого</sup> бита OPS: если он равен 1, то признак «Нет питания» равен сигналу «Нет питания», иначе не формируется.

**ВАЖНО.** При поступлении сигнала МУ (вход Х4) физические команды, выдаваемые из функции и запущенные по КО до появления сигнала МУ, сбрасываются и выставляется признак МУ. После выставления признака МУ состояние двигателя меняется на «Работа».

**3<sup>ий</sup> бит OPS** – указывает способ формирования выходного физического сигнала на включение ИМ при возникновении **сигнала** «Включить ЛЗ».

При **единичном** значении бита из функции выдается выходной физический сигнал на включение при поступлении **сигнала** «Включить ЛЗ» только если двигатель в состоянии «Резерв».

При **нулевом** значении бита при поступлении **сигнала** «Включить ЛЗ» из функции выдается выходной физический сигнал на включение двигателя.

**4<sup>ый</sup> бит OPS** – указывает способ формирования выходного физического сигнала на отключение ИМ при одновременном возникновении **сигналов** «Отключить ЛЗ» (вход X11) или «КО Отключить» (вход X16) и **сигнала** «Запрет отключения» (вход X18).

При **единичном** значении бита при поступлении **сигнала** «Запрет отключения» блокируется только вход X16 «КО Отключить».

При **нулевом** значении бита при поступлении **сигнала** «Запрет отключения» блокируются вход X16 «КО Отключить» и вход X11 «Отключить ЛЗ».

**5<sup>ый</sup> бит OPS** – указывает способ формирования выходного физического сигнала на включение ИМ при возникновении **сигналов** «Включить ЛЗ» (вход X10) или «КО Включить» (вход X15) и **сигнала** «Запрет включения» (вход X17).

При **единичном** значении бита при поступлении **сигнала** «Запрет включения» блокируется только вход X15 «КО Включить».

При **нулевом** значении бита при поступлении **сигнала** «Запрет включения» блокируются вход X15 «КО Включить» и вход X10 «Включить ЛЗ».

**6<sup>ой</sup> бит OPS** – указывает способ формирования выходных физических сигналов на включение и отключение ИМ (выход X1 «Физическая команда Включить» и Выход 3 «Физическая команда Отключить»).

При **единичном** значении бита используются два сигнала управления (выход X1 «Физическая команда Включить» и выход X3 «Физическая команда Отключить»).

При **нулевом** значении бита выход X3 «Физическая команда Отключить» игнорируется (всегда равен 0), а выход X1 «Физическая команда Включить» меняет смысл: 1 – Включить, 0 – Отключить.

**7<sup>ой</sup> бит OPS** – указывает способ формирования выходных физических сигналов на включение и отключение двигателя при получении заданного состояния ИМ.

При **единичном** значении бита физический сигнал сбрасывается в ноль как только ИМ перейдет в требуемое состояние. Это справедливо только для импульсных команд управления.

**ВАЖНО.** Условием формирования импульсного выходного физического сигнала управления является постановка **6<sup>ого</sup> бита OPS** в 1 (2 команды управления) и время выхода физического сигнала **Time\_Out\_CMD>0**.

При **нулевом** значении бита физический сигнал сбросится в ноль только по истечении времени выдачи физического сигнала (вход X20 «Уставка времени выдачи управляющих команд»).

**ВАЖНО.** Вне зависимости от состояния **7<sup>ого</sup> бита OPS**, если вход X20 «Уставка времени выдачи управляющих команд» равен 0 или **6<sup>ой</sup> бит OPS** равен 0, то физические сигналы не сбрасываются!

**8<sup>ой</sup> бит OPS** – указывает способ формирования состояния «Резерв» при появлении **сигнала** «Включить ЛЗ».

При **единичном** значении бита при включении двигателя ЛЗ сбрасывается состояние «Резерв» и выставляется состояние «Работа».

При **нулевом** значении бита при включении двигателя ЛЗ состояние двигателя остается неизменным.

**9<sup>ый</sup> бит OPS** – указывает способ формирования состояния «Резерв» при появлении **сигнала** «Отключить ЛЗ».

При **единичном** значении бита при отключении двигателя ЛЗ сбрасывается состояние «Резерв» и выставляется состояние «Работа».

При **нулевом** значении бита при отключении двигателя ЛЗ состояние двигателя остается неизменным.

**10<sup>ый</sup> бит OPS** – указывает способ формирования состояния «Резерв» при появлении «КО Включить».

При **единичном** значении бита при включении двигателя по КО сбрасывается состояние «Резерв» и выставляется состояние «Работа».

При **нулевом** значении бита при включении двигателя по КО состояние двигателя остается неизменным.

**11<sup>ый</sup> бит OPS** – указывает способ формирования состояния «Резерв» при появлении «КО Отключить».

При **единичном** значении бита при отключении двигателя по КО сбрасывается состояние «Резерв» и выставляется состояние «Работа».

При **нулевом** значении бита при отключении двигателя по КО состояние двигателя остается неизменным.

**12<sup>ый</sup> бит OPS** – указывает способ формирования выходного физического сигнала на отключение ИМ при появлении **сигнала** «Двигатель аварийно отключен».

При **единичном** значении бита при возникновении **сигнала** «Двигатель аварийно отключен» из функции будет выдана дублирующая команда на отключение ИМ.

При **нулевом** значении бита дублирующая команда на отключение ИМ при наличии **сигнала** Двигатель аварийно отключен» не выдается.

**ВАЖНО.** Вне зависимости от состояния **12<sup>ого</sup> бита OPS** производится контроль за отключением двигателя. Если по истечении времени контроля смены состояния (вход X19 «Уставка времени контроля смены состояния») двигатель не отключился, выставляется **признак** «Просрочено время на отключение от **SMA**».

**13<sup>ый</sup> бит OPS** – указывает способ формирования состояние «Резерв» при наличии **признака** «Неисправность KB».

При **единичном** значении бита при наличии **признака** «Неисправность KB» разрешается смена состояния двигателя на «Резерв» по «КО Резерв» (вход X13).

При **нулевом** значении бита при наличии **признака** «Неисправность KB» смена состояния двигателя на «Резерв» не возможна.

**14<sup>ый</sup> бит OPS** – указывает способ формирования состояние «Резерв» при наличии **признака** «Нет питания».

При **единичном** значении бита при наличии **признака** «Нет питания» разрешается смена состояния двигателя на «Резерв» по «КО Резерв» (вход X13).

При **нулевом** значении бита при наличии **признака** «Нет питания» смена состояния двигателя на «Резерв» не возможна.

**15<sup>ый</sup> бит OPS** – указывает способ сброса (квитации) **признаков** «Отключен ЛЗ» и «Отключен SMA (аварийными защитами)» по «КО Отключить».

При **единичном** значении бита разрешается квитация **признаков** «Отключен ЛЗ» и «Отключен SMA (аварийными защитами)» по «КО Отключить» (вход X16). При этом повторная команда на отключение двигателя из функции не выдается.

При **нулевом** значении бита квитация **признаков** «Отключен ЛЗ» и «Отключен SMA (аварийными защитами)» возможна по **сигналу** «Сброс сигнализации» (вход X21) или при выдачи команде на включение.

**16<sup>ый</sup> бит OPS** – указывает способ квитации **признака** «Включен ЛЗ» по «КО Включить».

При **единичном** значении бита разрешается квитация **признака** «Включен ЛЗ» «КО Включить» (вход X15). При этом, повторная команда на включение двигателя из функции не выдается.

При **нулевом** значении бита квитация **признака** «Включен ЛЗ» возможна по **сигналу** «Сброс сигнализации» (вход X21) или при выдаче команды на отключение.

**17<sup>ый</sup> бит OPS** – указывает способ формирования **признаков** «Нет питания» и «Неисправность KB».

При **единичном** значении бита формирование **признаков** «Нет питания» и «Неисправность KB» осуществляется всегда (вне зависимости от состояния входа X4 (Режим МУ) и **2<sup>ого</sup> бита OPS**).

При **нулевом** значении бита формирование **признаков** «Нет питания» и «Неисправность KB» осуществляется как указано выше.

**18<sup>ый</sup>, 19<sup>ый</sup>, 20<sup>ый</sup> бит OPS** – указывают способ формирования **признаков** «Двигатель включен» и «Двигатель отключен».

При **нулевом** значении **18<sup>ого</sup> бита OPS** формирование **признаков** «Двигатель включен» и «Двигатель отключен» осуществляется по состоянию KB. В случае, если по значению KB нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен», то их значения сбрасываются в ноль. При этом, значение **19<sup>ого</sup> и 20<sup>ого</sup> бита OPS** игнорируются.

При **единичном** значении **18<sup>ого</sup> бита OPS** **признаки** «Двигатель включен» и «Двигатель отключен» формируются в зависимости от **19<sup>ого</sup> бита OPS**.

При **нулевом** значении **19<sup>ого</sup> бита OPS** формирование **признаков** «Двигатель включен» и «Двигатель отключен» осуществляется по состоянию KB. В случае, если по значению KB нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен», то они остаются равными последнему достоверному состоянию. При этом, значение **20<sup>ого</sup> бита OPS** игнорируется.

При **единичном** значении **19<sup>ого</sup> бита OPS** **признаки** «Двигатель включен» и «Двигатель отключен» формируются в зависимости от **20<sup>ого</sup> бита OPS**.

При **нулевом** значении **20<sup>ого</sup> бита OPS** формирование **признаков** «Двигатель включен» и «Двигатель отключен» осуществляется по состоянию КВ. В случае, если по значению КВ нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен», то считается, что двигатель отключен.

При **единичном** значении **20<sup>ого</sup> бита OPS** формирование **признаков** «Двигатель включен» и «Двигатель отключен» осуществляется по состоянию КВ. В случае, если по значению КВ нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен», то считается, что двигатель включен.

**ВАЖНО.** Нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен» в том случае, если функцией обрабатываются значения от обоих КВ (**0<sup>ой</sup> бит OPS** равен 1) и значения входов X7 «Двигатель включен» и X8 «Двигатель отключен» равны 0 или 1.

**X3 - Двигатель аварийно отключен.** Сигнал «Двигатель аварийно отключен» (SMA).

Реакция функции на этот сигнал определена в **12<sup>ом</sup> бите OPS**;

**X4 – МУ.** Режим «Местное управление» (Rg\_MU). При единичном значении Rg\_MU игнорируются КО. При Rg\_MU = 1 будут приняты в обработку вход X5 «МУ «Включить» и вход X6 «МУ «Отключить». Способ обработки определяет **2<sup>ой</sup> бит OPS**;

**X5 - МУ "Включить".** Команда «Включить» при местном управлении (MU\_ON);

**X6 - МУ "Отключить".** Команда «Отключить» при местном управлении (MU\_OFF);

**X7 - Двигатель включен.** Двигатель включен (On). Используется совместно с параметром Off для идентификации состояния двигателя.

Реакция функции на этот сигнал определена в **0<sup>ом</sup> бите OPS**;

**X8 - Двигатель отключен.** Двигатель отключен (Off). Используется совместно с параметром On для идентификации состояния двигателя.

Реакция функции на этот сигнал определена в **0<sup>ом</sup> бите OPS**;

**X9 - Нет питания.** Нет питания. Реакция функции на этот сигнал определена в **1<sup>ом</sup> бите OPS**;

**X10 - Включить ЛЗ.** Включить логической задачей (ONLZ). На данный вход могут подаваться команды алгоритмического управления (если двигатель надо включать через определенное время), команда «Включить по АВР» и команды включения двигателя по блокировкам и защита;

**X11 - Отключить ЛЗ .** Отключить логической задачей (OFFLZ). На данный вход могут подаваться команды алгоритмического управления (если двигатель надо отключать через определенное время), и команды отключения двигателя по блокировкам и защита;

**X12 - КО "Ремонт".** Команда оператора «Ремонт». Устанавливает состояние двигателя «Ремонт»;

**ВАЖНО.** При установке состояния «Ремонт» ВСЕ управляющие команды из функции блокируются.

**X13 - КО "Резерв".** Команда оператора «Резерв». Устанавливает состояние двигателя «Резерв» с учетом значений **13<sup>ого</sup>** и **14<sup>ого</sup>** битов **OPS**;

**X14 - КО "Работа".** Команда оператора «Работа». Устанавливает состояние двигателя «Работа»;

**ВАЖНО.** Смена состояний двигателя оператором возможна только после завершения всех управляющих воздействий на ИМ.

**X15 - КО "Включить".** Команда оператора «Включить» (KO\_ON);

**X16 - КО "Отключить".** Команда оператора «Отключить» (KO\_OFF);

**X17 - Запрет включения.** Признак запрета включения двигателя. Реакция функции на этот сигнал определена в **5<sup>ом</sup>** бите **OPS**;

**X18 - Запрет отключения.** Признак запрета отключения двигателя. Реакция функции на этот сигнал определена в **4<sup>ом</sup>** бите **OPS**;

**X19 - Задержка на изменение состояния.** Уставка максимального времени на изменение состояния двигателя (Time\_Wait).

**ВАЖНО.** Значение уставки Time\_Wait должно быть больше 0, иначе будет считаться, что смена состояний КВ происходит мгновенно (в течение одного цикла контроллера);

**X20 - Длительность выдачи команд.** Уставка времени выдачи управляющих команд (Time\_Out\_CMD).

**ВАЖНО.** Если уставка Time\_Out\_CMD равна 0, то выходы Y1 «Физическая команда Включить» и Y3 «Физическая команда Отключить» **ПОТЕНЦИАЛЬНЫЕ**, если больше 0 – **ИМПУЛЬСНЫЕ** с временем импульса, равным Time\_Out\_CMD.

**ВАЖНО.** Значение уставки Time\_Out\_CMD не должно превышать значение уставки Time\_Wait. В случае, если Time\_Out\_CMD > Time\_Wait и будет превышено время на изменение состояния ИМ, выходные команды сбросятся до истечения Time\_Out\_CMD;

**X21 - Сброс сигнализации.** Сброс сигнализации. При **единичном** значении входа сбрасываются признаки невыполнения команд и признаки несанкционированного включения/отключения (например, превышено время включения ЛЗ, превышено время отключения при МУ, двигатель несанкционированно включился);

**X22 - Состояние двигателя.** Состояние двигателя (**StatusIm**).

**0<sup>ый</sup>** бит **StatusIm** – состояние «Ремонт». Состояние «Ремонт» может выставить только оператор командой «Ремонт» при условии, что ИМ отключен, нет команд оператора перевода в состояние «Резерв» или «Работа» и нет ни одной управляющей команды.

**1<sup>ый</sup>** бит **StatusIm** – состояние «Резерв». Состояние «Резерв» может выставить только оператор командой «Резерв» при условии, что ИМ отключен, нет команд оператора перевода в состояние «Ремонт» или «Работа» и нет ни одной управляющей команды, а также если нет ошибки КВ или разрешен перевод в состояние «Резерв» при наличии ошибки КВ (**13<sup>ый</sup>** бит **OPS**

равен 1) и нет признака «Нет питания» или разрешен перевод в состояние «Резерв» при наличии признака «Нет питания» (14<sup>ый</sup> бит OPS равен 1).

2<sup>ой</sup> бит StatusIm – состояние «Работа».

3<sup>ий</sup> бит StatusIm – запущен процесс отключения ИМ аварийными защитами ИМ. Данный бит выставляется в 1, если выполняется контроль отключения или выдается дублирующая команда на отключение при наличии сигнала «Двигатель аварийно отключен» (SMA).

4<sup>ый</sup> бит StatusIm – запущен процесс отключения ИМ командой местного управления (MU\_OFF). Данный бит выставляется в 1, если выполняется отключение ИМ по команде МУ «Отключить».

5<sup>ый</sup> бит StatusIm – запущен процесс включения ИМ командой местного управления (MU\_ON). Данный бит выставляется в 1, если выполняется включение ИМ по команде МУ «Включить».

6<sup>ой</sup> бит StatusIm – запущен процесс отключения ИМ логической задачей (OFFLZ). Данный бит выставляется в 1, если выполняется отключение ИМ по команде «Отключить ЛЗ».

7<sup>ой</sup> бит StatusIm – запущен процесс включения ИМ логической задачей (ONLZ). Данный бит выставляется в 1, если выполняется включение ИМ по команде «Включить ЛЗ».

8<sup>ой</sup> бит StatusIm – запущен процесс отключения ИМ командой оператора (KO\_OFF). Данный бит выставляется в 1, если выполняется отключение ИМ по «КО Отключить».

9<sup>ый</sup> бит StatusIm – запущен процесс включения ИМ командой оператора (KO\_ON). Данный бит выставляется в 1, если выполняется включение ИМ по «КО Включить».

10<sup>ый</sup> бит StatusIm – признак «Нет питания».

11<sup>ый</sup> бит StatusIm – признак «Неисправность КВ».

12<sup>ый</sup> бит StatusIm – сохраненный сигнал – Отключение аварийными защитами ИМ. Данный бит выставляется в 1, если произошло отключение ИМ при наличии сигнала «Двигатель аварийно отключен» (SMA) (вне зависимости, выдавалась ли дублирующая команда или выполнялся только контроль отключения).

13<sup>ый</sup> бит StatusIm – сохраненный сигнал – Отключение ИМ командой местного управления. Данный бит выставляется в 1, если произошло отключение ИМ по команде MU\_OFF.



- 14<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Включение ИМ командой местного управления. Данный бит выставляется в 1, если произошло включение ИМ по команде MU\_ON.
- 15<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Отключение ИМ командой ЛЗ. Данный бит выставляется в 1, если произошло отключение ИМ по команде OFFLZ.
- 16<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Включение ИМ командой ЛЗ. Данный бит выставляется в 1, если произошло включение ИМ по команде ONLZ.
- 17<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Отключение ИМ КО. Данный бит выставляется в 1, если произошло отключение ИМ по «КО Отключить».
- 18<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Включение ИМ КО. Данный бит выставляется в 1, если произошло Включение ИМ по «КО Включить».
- 19<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Несанкционированное отключение ИМ. Данный бит выставляется в 1, если произошло отключение ИМ без команд управления. Контроль осуществляется, если ИМ запитан, нет ошибки КВ, **17<sup>ый</sup> бит OPS** равен 1. В случае, если **17<sup>ый</sup> бит OPS** равен 0, то не должно быть признака МУ или управление при МУ через контроллер (**2<sup>ый</sup> бит OPS** равен 1).
- 20<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Несанкционированное включение ИМ. Данный бит выставляется в 1, если произошло включение ИМ без команд управления. Контроль осуществляется, если ИМ запитан, нет ошибки КВ, **17<sup>ый</sup> бит OPS** равен 1. В случае, если **17<sup>ый</sup> бит OPS** равен 0, то не должно быть признака МУ или управление при МУ через контроллер (**2<sup>ый</sup> бит OPS** равен 1).
- 21<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Просрочено время на отключение ИМ от SMA. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Отключен».
- 22<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Просрочено время на отключение ИМ от команде MU\_OFF. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Отключен».
- 23<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Просрочено время на включение ИМ от команде MU\_ON. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Включен».
- 24<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Просрочено время на отключение ИМ от команде OFFLZ. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Отключен».

- 25<sup>ый</sup> бит StatusIm** – сохраненный сигнал – Просрочено время на включение ИМ от команде ONLZ. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Включен».
- 26<sup>ой</sup> бит StatusIm** – сохраненный сигнал – Просрочено время на отключение ИМ от «КО Отключить». Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Отключен».
- 27<sup>ой</sup> бит StatusIm** – сохраненный сигнал – Просрочено время на включение ИМ от «КО Включить». Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Включен».
- 28<sup>ой</sup> бит StatusIm** – Выдать команду «Включить».
- 29<sup>ый</sup> бит StatusIm** – Выдать команду «Отключить».
- 30<sup>ый</sup> бит StatusIm** – **признак** «Двигатель включен». Данный бит выставляется в 1, если двигатель включен. Данный **признак** определяется в зависимости от значений **18<sup>ого</sup>**, **19<sup>ого</sup>**, **20<sup>ого</sup>** бита **OPS** и является информационным. Он не участвует в определении признаков «Нет питания», «Неисправность KB», «Просрочено время на ...», «Несанкционированное включение/отключение».
- 31<sup>ый</sup> бит StatusIm** – **признак** «Двигатель отключен». Данный бит выставляется в 1, если двигатель отключен. Данный **признак** определяется в зависимости от значений **18<sup>ого</sup>**, **19<sup>ого</sup>**, **20<sup>ого</sup>** бита **OPS** и является информационным. Он не участвует в определении признаков «Нет питания», «Неисправность KB», «Просрочено время на ...», «Несанкционированное включение/отключение».

### Выходные данные:

- Y1 - Физ команда "Включить".** Физическая команда «Включить»;
- Y2 - КО "Включить".** Команда оператора «Включить». По выходу из функции всегда равна 0;
- Y3 - Физ команда "Отключить".** Физическая команда «Отключить»;
- Y4 - КО "Включить".** Команда оператора «Отключить». По выходу из функции всегда равна 0;
- Y5 - КО "Ремонт".** Команда оператора «Ремонт». По выходу из функции всегда равна 0;
- Y6 - КО "Резерв".** Команда оператора «Резерв». По выходу из функции всегда равна 0;
- Y7 - КО "Работа".** Команда оператора «Работа». По выходу из функции всегда равна 0;
- Y8 - Сброс сигнализации.** Сброс сигнализации. По выходу из функции всегда равен 0;
- Y9 - Состояние двигателя.** Состояние двигателя.

## Пример вызова функции

СТ:

ФБД:

перем\_ц32 =  
BCFI(PB1)

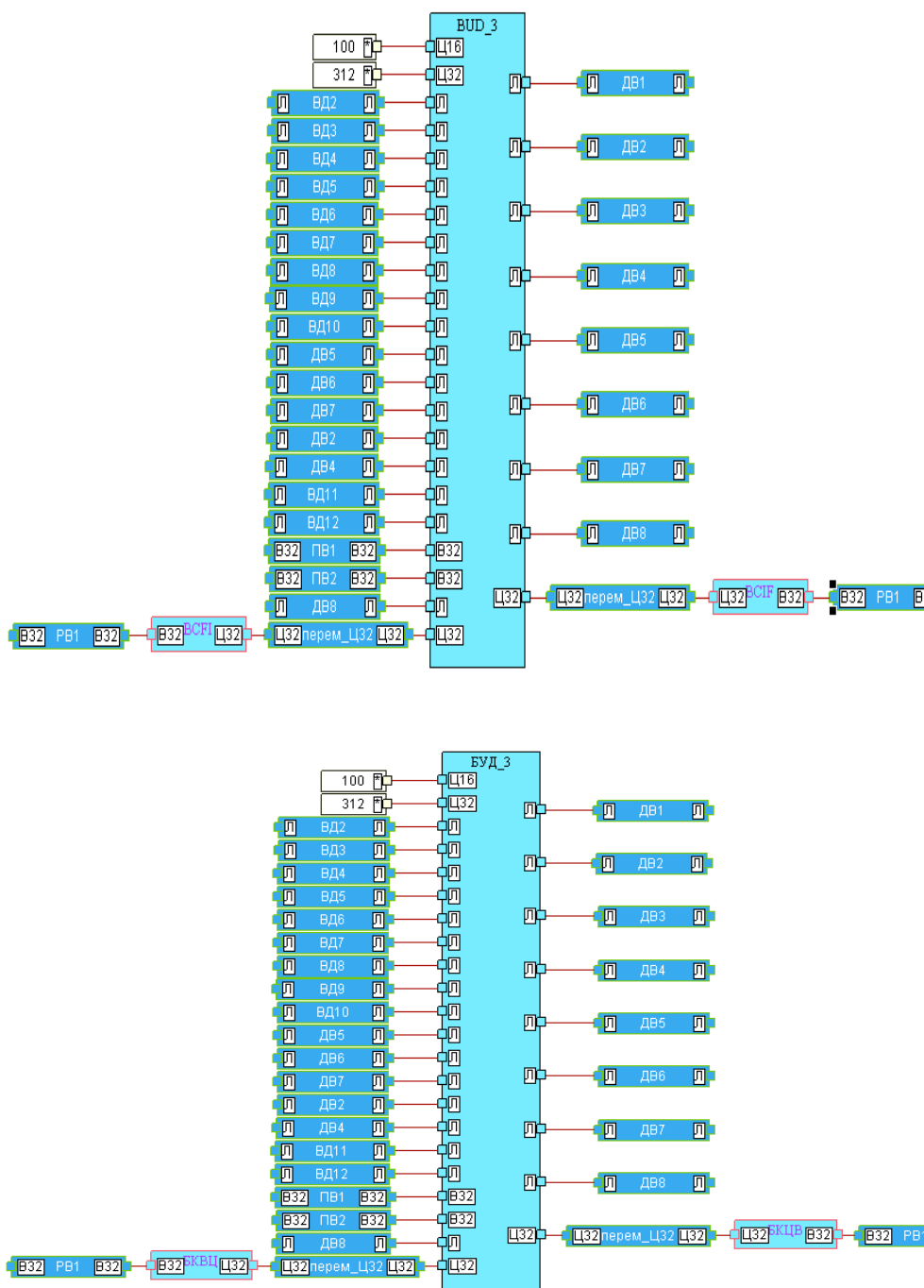
(ДВ1, ДВ2, ДВ3, ДВ4,  
ДВ5, ДВ6, ДВ7, ДВ8,  
перем\_ц32) =  
БУД\_3(100, 312, ВД2,  
ВД3, ВД4, ВД5, ВД6,  
ВД7, ВД8, ВД9, ВД10,  
ДВ5, ДВ6, ДВ7, ДВ2,  
ДВ4, ВД11, ВД12, пв1,  
пв2, ДВ8, перем\_ц32)

PB1 =  
BCIF(перем\_ц32)

перем\_ц32 =  
БКВЦ(PB1)

(ДВ1, ДВ2, ДВ3, ДВ4,  
ДВ5, ДВ6, ДВ7, ДВ8,  
перем\_ц32) =  
БУД\_3(100, 312, ВД2,  
ВД3, ВД4, ВД5, ВД6,  
ВД7, ВД8, ВД9, ВД10,  
ДВ5, ДВ6, ДВ7, ДВ2,  
ДВ4, ВД11, ВД12, пв1,  
пв2, ДВ8, перем\_ц32)

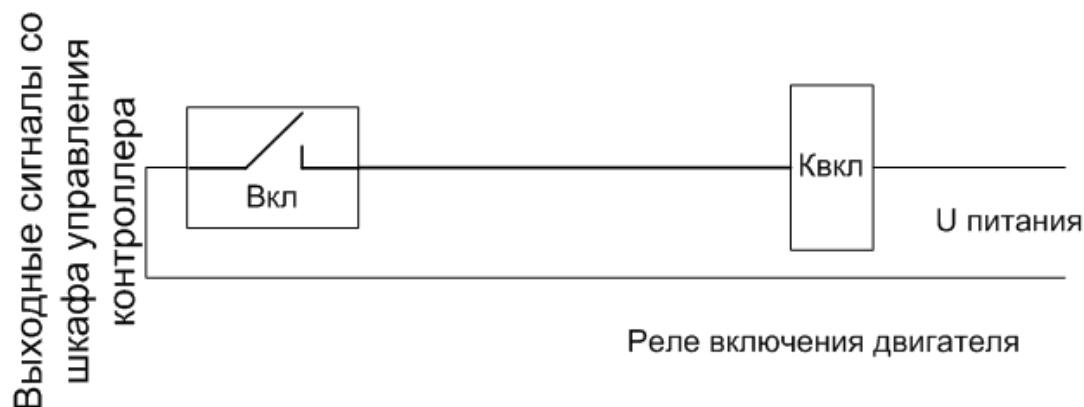
PB1 =  
БКЦВ(перем\_ц32)



где перем\_ц32 – локальная 32-битная переменная целого типа

Даная функция предназначена для схем управления двигателем, приведенных ниже:

### Для схемы с одной (потенциальной) управляющей командой



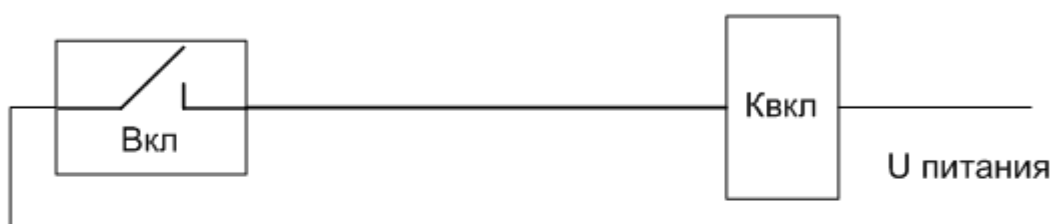
Некоторые настройки кода OPS для схемы с одной управляющей командой:

- OPS = 0:
  - Определение состояния двигателя по 1 KB;
  - Сигнал «Нет питания» не заведен в функцию;
  - МУ не через функцию;
  - Физическая команда «Включить» по сигналу ONLZ всегда;
  - Запрет отключения для всех команд;
  - Запрет включения для всех команд;
  - Количество команд управления – 1;
  - Не сбрасывать команду управления по получению требуемого состояния ИМ **(Для схем с одной командой управления данный бит всегда 0)**;
  - При включении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При отключении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При включении ИМ КО «Включить» не сбрасывать состояние «Резерв»;
  - При отключении ИМ КО «Отключить» не сбрасывать состояние «Резерв»;
  - Не выдавать дублирующую команду при SMA;
  - Не разрешать установку состояния «Резерв» при ошибке KB;
  - Не разрешать установку состояния «Резерв» при признаке «Нет питания»;
  - Не разрешать квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
  - Не разрешать квитацию сигнала «Включен ЛЗ» по КО «Включить»;
  - Отключить диагностику при МУ, когда МУ не через контроллер;
  - Определять состояние двигателя по KB;
  - Определять состояние двигателя по последнему достоверному значению;
  - При неисправности KB считать, что двигатель отключен.
- OPS = 126842/130938:
  - Определение состояния двигателя по 1 KB;
  - Сигнал «Нет питания» заведен в функцию;
  - МУ не через функцию;
  - Физическая команда «Включить» по сигналу ONLZ только в резерве;
  - Запрет отключения для КО;
  - Запрет включения для КО;

- Количество команд управления – 1;
- Не сбрасывать команду управления по получению требуемого состояния ИМ (**Для схем с одной командой управления данный бит всегда 0**);
- При включении ИМ ЛЗ сбрасывать состояние «Резерв»;
- При отключении ИМ ЛЗ сбрасывать состояние «Резерв»;
- При включении ИМ КО «Включить» сбрасывать состояние «Резерв»;
- При отключении ИМ КО «Отключить» сбрасывать состояние «Резерв»;
- Не выдавать дублирующую команду при SMA/Выдавать дублирующую команду при SMA;
- Разрешить установку состояния «Резерв» при ошибке КВ;
- Разрешить установку состояния «Резерв» при признаке «Нет питания»;
- Разрешить квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
- Разрешить квитацию сигнала «Включен ЛЗ» по КО «Включить»;
- Отключить диагностику при МУ, когда МУ не через контроллер;
- Определять состояние двигателя по КВ;
- Определять состояние двигателя по последнему достоверному значению;
- При неисправности КВ считать, что двигатель отключен.

### Для схемы с двумя (потенциальными) управляющими командами

Выходные сигналы со шкафа  
управления контроллера



Реле включения двигателя



Реле отключения двигателя

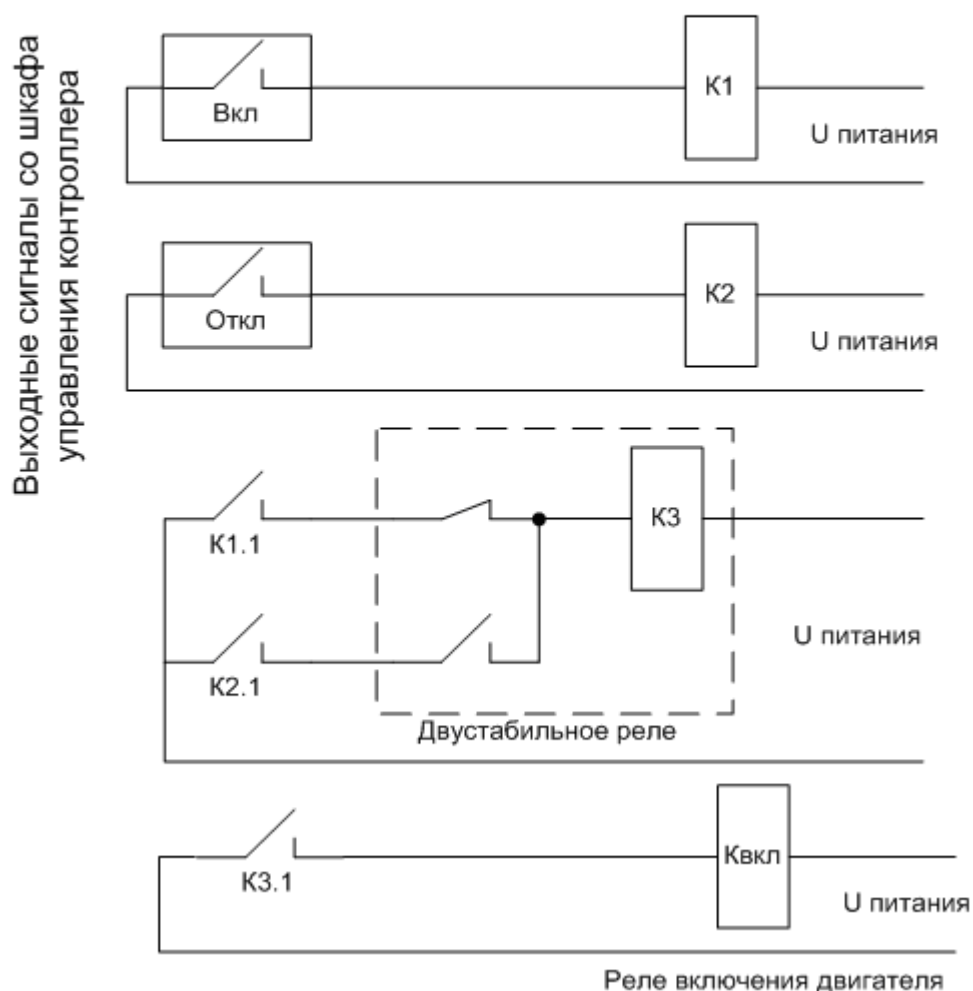
**ВАЖНО:** для данной схемы обязательно нужно, чтоб Time\_Out\_CMD (Уставка максимального времени подачи команды) была равна 0.

Некоторые настройки кода OPS для схемы с двумя потенциальными управляющими командами:

- OPS = 69:
  - Определение состояния двигателя по 2 KB;
  - Сигнал «Нет питания» не заведен в функцию;
  - МУ через функцию;
  - Физическая команда «Включить» по сигналу ONLZ только в резерве;
  - Запрет отключения для всех команд;
  - Запрет включения для всех команд;
  - Количество команд управления – 2;
  - Не сбрасывать команду управления по получению требуемого состояния ИМ **(Для схем с двумя потенциальными командами управления данный бит всегда 0)**;
  - При включении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При отключении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При включении ИМ КО «Включить» не сбрасывать состояние «Резерв»;
  - При отключении ИМ КО «Отключить» не сбрасывать состояние «Резерв»;
  - Не выдавать дублирующую команду при SMA;
  - Не разрешать установку состояния «Резерв» при ошибке KB;
  - Не разрешать установку состояния «Резерв» при признаке «Нет питания»;
  - Не разрешать квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
  - Не разрешать квитацию сигнала «Включен ЛЗ» по КО «Включить»;
  - Отключить диагностику при МУ, когда МУ не через контроллер;
  - Определять состояние двигателя по KB;
  - Определять состояние двигателя по последнему достоверному значению;
  - При неисправности KB считать, что двигатель отключен.
  
- OPS = 130943/130941:
  - Определение состояния двигателя по 2 KB;
  - Сигнал «Нет питания» заведен в функцию/Сигнал «Нет питания» не заведен в функцию;
  - МУ через функцию;
  - Физическая команда «Включить» по сигналу ONLZ только в резерве;
  - Запрет отключения для КО;
  - Запрет включения для КО;
  - Количество команд управления – 2;
  - Не сбрасывать команду управления по получению требуемого состояния ИМ **(Для схем с двумя потенциальными командами управления данный бит всегда 0)**;
  - При включении ИМ ЛЗ сбрасывать состояние «Резерв»;
  - При отключении ИМ ЛЗ сбрасывать состояние «Резерв»;
  - При включении ИМ КО «Включить» сбрасывать состояние «Резерв»;
  - При отключении ИМ КО «Отключить» сбрасывать состояние «Резерв»;
  - Выдавать дублирующую команду при SMA;
  - Разрешить установку состояния «Резерв» при ошибке KB;
  - Разрешить установку состояния «Резерв» при признаке «Нет питания»;
  - Разрешить квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
  - Разрешить квитацию сигнала «Включен ЛЗ» по КО «Включить»;
  - Отключить диагностику при МУ, когда МУ не через контроллер;

- Определять состояние двигателя по КВ;
- Определять состояние двигателя по последнему достоверному значению;
- При неисправность КВ считать, что двигатель отключен.

### Для схемы с двумя (импульсными) управляющими командами



**ВАЖНО:** для данной схемы обязательно нужно, чтоб Time\_Out\_CMD (Уставка максимального времени подачи команды) была больше 0.

Некоторые настройки кода OPS для схемы с двумя управляющими командами:

- OPS = 64:
  - Определение состояния двигателя по 1 КВ;
  - Сигнал «Нет питания» не заведен в функцию;
  - МУ не через функцию;
  - Физическая команда «Включить» по сигналу ONLZ только в резерве;
  - Запрет отключения для всех команд;

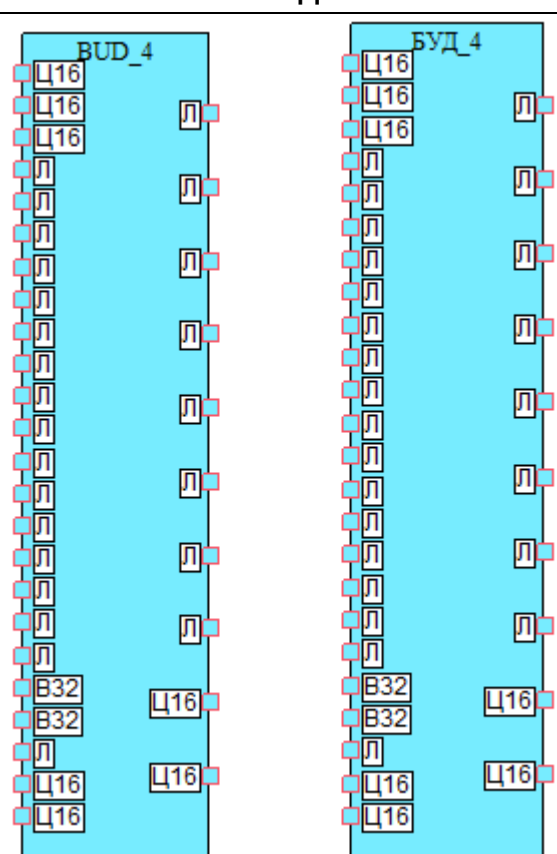
- Запрет включения для всех команд;
  - Количество команд управления – 2;
  - Не сбрасывать команду управления по получению требуемого состояния ИМ;
  - При включении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При отключении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При включении ИМ КО «Включить» не сбрасывать состояние «Резерв»;
  - При отключении ИМ КО «Отключить» не сбрасывать состояние «Резерв»;
  - Не выдавать дублирующую команду при SMA;
  - Не разрешать установку состояния «Резерв» при ошибке KB;
  - Не разрешать установку состояния «Резерв» при признаке «Нет питания»;
  - Не разрешать квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
  - Не разрешать квитацию сигнала «Включен ЛЗ» по КО «Включить»;
  - Отключить диагностику при МУ, когда МУ не через контроллер;
  - Определять состояние двигателя по KB;
  - Определять состояние двигателя по последнему достоверному значению;
  - При неисправности KB считать, что двигатель отключен.
- OPS = 262143/262015:
    - Определение состояния двигателя по 2 KB;
    - Сигнал «Нет питания» заведен в функцию;
    - МУ через функцию;
    - Физическая команда «Включить» по сигналу ONLZ только в резерве;
    - Запрет отключения для КО;
    - Запрет включения для КО;
    - Количество команд управления – 2;
    - Не сбрасывать команду управления по получению требуемого состояния ИМ/сбрасывать команду управления по получению требуемого состояния ИМ;
    - При включении ИМ ЛЗ сбрасывать состояние «Резерв»;
    - При отключении ИМ ЛЗ сбрасывать состояние «Резерв»;
    - При включении ИМ КО «Включить» сбрасывать состояние «Резерв»;
    - При отключении ИМ КО «Отключить» сбрасывать состояние «Резерв»;
    - Выдавать дублирующую команду при SMA;
    - Разрешить установку состояния «Резерв» при ошибке KB;
    - Разрешить установку состояния «Резерв» при признаке «Нет питания»;
    - Разрешить квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
    - Разрешить квитацию сигнала «Включен ЛЗ» по КО «Включить»;
    - Включить диагностику при МУ, когда МУ не через контроллер;
    - Определять состояние двигателя по KB;
    - Определять состояние двигателя по последнему достоверному значению;
    - При неисправности KB считать, что двигатель отключен.



## 14.5 ВУД\_4, ВУД\_4

**Назначение**

Функция управления электрифицированным двигателем.

Отображение	
СТ:	ФБД:
<p>(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10) = ВУД_4(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20, X21, X22, X23, X24)</p> <p>(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10) = БУД_4(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20, X21, X22, X23, X24)</p>	
<p><b>Входные параметры:</b>  X1(Ц16), X2 (Ц16), X3 (Ц16), X4(л), X5(л), X6(л), X7(л), X8(л), X9(л), X10(л), X11(л), X12(л), X13(л), X14(л), X15(л), X16(л), X17(л), X18(л), X19(л), X20(В32), X21(В32), X22(л), X23(Ц16), X24(Ц16).</p> <p><b>Выходные параметры:</b>  Y1(л), Y2(л), Y3(л), Y4(л), Y5(л), Y6(л), Y7(л), Y8(л), Y9(Ц16), Y10(Ц16).</p>	

**Логика работы функции**

Для корректной работы функции необходимо, чтобы:

- На входе X13 и выходе Y5 (КО «Ремонт») была одна и та же переменная;
- На входе X14 и выходе Y6 (КО «Резерв») была одна и та же переменная;
- На входе X15 и выходе Y7 (КО «Работа») была одна и та же переменная;
- На входе X16 и выходе Y2 (КО «Включить») была одна и та же переменная;
- На входе X17 и выходе Y4 (КО «Отключить») была одна и та же переменная;

- На входах X23, X24 и выходах Y9, Y10 (Состояние двигателя) были одни и те же переменные соответственно;

Если сброс сигнализации происходит по команде оператора (КО) «Сброс сигнализации», то на входе X22 и выходе Y8 (Сброс сигнализации) должна быть одна и та же переменная. Если сброс осуществляется по условию, то выход Y8 может быть не привязан к переменной.

**ВАЖНО.** Значение переменной "Состояние двигателя" (вход X23, X24 и выход Y9, Y10) должно храниться между циклами контроллера.

### Приоритет входных команд управления

Для команд, имеющих одинаковый приоритет, команда на отключение имеет преимущество (например, при одновременной подаче команд KO\_ON и KO\_OFF будет выполняться KO\_OFF).

1. SMA (двигатель аварийно отключен) – наивысший приоритет.
2. OFFLZ (Отключить ЛЗ) и ONLZ (Включить ЛЗ).
3. MU\_OFF (Отключить МУ) и MU\_ON (Включить МУ).
4. KO\_OFF (Отключить командой оператора) и KO\_ON (Включить командой оператора).

**ВАЖНО.** Все команды на включение игнорируются при наличии признака «Нет питания».

**ВАЖНО.** КО «Включить» игнорируется при наличии признака «Неисправность КВ».

### Входные параметры:

**X1 – Номер.** Порядковый номер функции (от 1 до 20000). Необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

**X2 – OPS\_1.** Описание признаков настройки:

**0<sup>ый</sup> бит OPS\_1** – указывает способ определения состояния двигателя по 2 концевым выключателям (КВ) или по 1 КВ и обработке входа X7 «Двигатель включен» и входа X8 «Двигатель отключен».

При **единичном** значении бита обрабатываются оба входа и состояние двигателя «Включен/Отключен» определяется по 2 концевым выключателям.

При **нулевом** значении бита обрабатывается только вход X7 «Двигатель включен» и состояние двигателя определяется только по состоянию данного входа. При этом значение входа X8 игнорируется.

**1<sup>ый</sup> бит OPS\_1** – указывает способ формирования признаков «Нет питания» и «Неисправность КВ» и обработку входа X9 «Нет питания».

**ВАЖНО.** При формировании признаков «Нет питания» и «Неисправность КВ» учитываются значения **2<sup>ого</sup>** и **17<sup>ого</sup>** бита OPS и входа X4 «Режим «МУ».

При **единичном** значении бита вход X9 «Нет питания» обрабатывается функцией и **признак** «Нет питания» равен **сигналу** «Нет питания». При этом **признак** «Неисправность КВ» формируется, если значение входов X7 «Двигатель включен» и X8 «Двигатель отключен» равны.

**ВАЖНО.** Формирование признака «Неисправность КВ» осуществляется, если **0<sup>ый</sup> бит OPS** равен 1, иначе **признак** «Неисправность КВ» не выставляется.

При **нулевом** значении бита вход Х9 «Нет питания» игнорируется и **признак** «Нет питания» определяется по состоянию входов Х7 «Двигатель включен» и Х8 «Двигатель отключен» (если оба входа равны 0, иначе не формируется).

**ВАЖНО.** Формирование **признака** «Нет питания» осуществляется, если **0<sup>ой</sup> бит OPS** равен 1, иначе **признак** «Нет питания» не формируется.

**ВАЖНО.** Формирование **признака** «Неисправность КВ» осуществляется, если **0<sup>ой</sup> бит OPS** равен 1 (значение входов Х7 «Двигатель включен» и Х8 «Двигатель отключен» равны 1), иначе **признак** «Неисправность КВ» не выставляется.

**2<sup>ой</sup> бит OPS\_1** – указывает способ формирования **признаков** «Нет питания» и «Неисправность КВ» и реакцию функции на совместное появление единичных значений на входах Х4 «Режим «МУ» и Х5 «МУ «Включить» или входах Х4 «Режим «МУ» и Х6 «МУ «Отключить».

**ВАЖНО.** При формировании **признаков** «Нет питания» и «Неисправность КВ» учитывается значение **1<sup>ого</sup> бита OPS\_1**.

При **единичном** значении бита управление при МУ (вход Х4 равен 1) осуществляется через функцию и обрабатываются входы Х6 «МУ «Включить» и Х7 «МУ «Отключить». При этом, прохождение управляющих команд от оператора блокируется (игнорируются входы Х15 «КО Включить» и Х16 «КО Отключить»).

**ВАЖНО.** При поступлении **сигнала** МУ (вход Х5) физические команды, выдаваемые из функции и запущенные до появления **сигнала** МУ, остаются до истечения времени выдачи выходных команд (вход Х21) и только по завершении всех управляющих воздействий выставляется **признак** МУ. После выставления **признака** МУ состояние двигателя меняется на «Работа».

При **нулевом** значении бита управление при МУ осуществляется помимо контроллера. При этом, при появлении единичного значения на входе Х5 «МУ «Включить» или Х6 «МУ «Отключить» управляющее воздействие из функции не выдается, а запускается контроль за выполнением команды. При наличии **сигнала** МУ (вход Х4 равен 1) прохождение управляющих команд от оператора блокируется (игнорируются входы Х15 «КО Включить» и Х16 «КО Отключить»). Также при наличии **сигнала** МУ не производится формирование **признака** «Неисправность КВ». Формирование **признака** «Нет питания» зависит от **1<sup>ого</sup> бита OPS**: если он равен 1, то **признак** «Нет питания» равен **сигналу** «Нет питания», иначе не формируется.

**ВАЖНО.** При поступлении **сигнала** МУ (вход Х4) физические команды, выдаваемые из функции и запущенные по КО до появления **сигнала** МУ, сбрасываются и выставляется **признак** МУ. После выставления **признака** МУ состояние двигателя меняется на «Работа».

**3<sup>ий</sup> бит OPS\_1** – указывает способ формирования выходного физического сигнала на включение ИМ при возникновении **сигнала** «Включить ЛЗ».

При **единичном** значении бита из функции выдается выходной физический сигнал на включение при поступлении **сигнала** «Включить ЛЗ» только если двигатель в состоянии «Резерв».

При **нулевом** значении бита при поступлении **сигнала** «Включить ЛЗ» из функции выдается выходной физический сигнал на включение двигателя.

**4<sup>ый</sup> бит OPS\_1** – указывает способ формирования выходного физического сигнала на отключение ИМ при одновременном возникновении **сигналов** «Отключить ЛЗ» (вход X11) или «КО Отключить» (вход X16) и **сигнала** «Запрет отключения» (вход X18).

При **единичном** значении бита при поступлении **сигнала** «Запрет отключения» блокируется только вход X17 «КО Отключить».

При **нулевом** значении бита при поступлении **сигнала** «Запрет отключения» блокируются вход X17 «КО Отключить» и вход X12 «Отключить ЛЗ».

**5<sup>ый</sup> бит OPS\_1** – указывает способ формирования выходного физического сигнала на включение ИМ при возникновении **сигналов** «Включить ЛЗ» (вход X11) или «КО Включить» (вход X16) и **сигнала** «Запрет включения» (вход X18).

При **единичном** значении бита при поступлении **сигнала** «Запрет включения» блокируется только вход X16 «КО Включить».

При **нулевом** значении бита при поступлении **сигнала** «Запрет включения» блокируются вход X16 «КО Включить» и вход X11 «Включить ЛЗ».

**6<sup>ой</sup> бит OPS\_1** – указывает способ формирования выходных физических сигналов на включение и отключение ИМ (выход X1 «Физическая команда Включить» и Выход 3 «Физическая команда Отключить»).

При **единичном** значении бита используются два сигнала управления (выход X1 «Физическая команда Включить» и выход X3 «Физическая команда Отключить»).

При **нулевом** значении бита выход X3 «Физическая команда Отключить» игнорируется (всегда равен 0), а выход X1 «Физическая команда Включить» меняет смысл: 1 – Включить, 0 – Отключить.

**7<sup>ой</sup> бит OPS\_1** – указывает способ формирования выходных физических сигналов на включение и отключение двигателя при получении заданного состояния ИМ.

При **единичном** значении бита физический сигнал сбрасывается в ноль как только ИМ перейдет в требуемое состояние. Это справедливо только для импульсных команд управления.

**ВАЖНО.** Условием формирования импульсного выходного физического сигнала управления является постановка **6<sup>ого</sup> бита OPS** в 1 (2 команды управления) и время выхода физического сигнала **Time\_Out\_CMD** > 0.

При **нулевом** значении бита физический сигнал сбросится в ноль только по истечении времени выдачи физического сигнала (вход X21 «Уставка времени выдачи управляющих команд»).

**ВАЖНО.** Вне зависимости от состояния **7<sup>ого</sup> бита OPS\_1**, если вход X21 «Уставка времени выдачи управляющих команд» равен 0 или **6<sup>ой</sup> бит OPS\_1** равен 0, то физические сигналы не сбрасываются!

**8<sup>ой</sup> бит OPS\_1** – указывает способ формирования состояния «Резерв» при появлении **сигнала** «Включить ЛЗ».

При **единичном** значении бита при включении двигателя ЛЗ сбрасывается состояние «Резерв» и выставляется состояние «Работа».

При **нулевом** значении бита при включении двигателя ЛЗ состояние двигателя остается неизменным.

**9<sup>ый</sup> бит OPS\_1** – указывает способ формирования состояния «Резерв» при появлении **сигнала** «Отключить ЛЗ».

При **единичном** значении бита при отключении двигателя ЛЗ сбрасывается состояние «Резерв» и выставляется состояние «Работа».

При **нулевом** значении бита при отключении двигателя ЛЗ состояние двигателя остается неизменным.

**10<sup>ый</sup> бит OPS\_1** – указывает способ формирования состояния «Резерв» при появлении «КО Включить».

При **единичном** значении бита при включении двигателя по КО сбрасывается состояние «Резерв» и выставляется состояние «Работа».

При **нулевом** значении бита при включении двигателя по КО состояние двигателя остается неизменным.

**11<sup>ый</sup> бит OPS\_1** – указывает способ формирования состояния «Резерв» при появлении «КО Отключить».

При **единичном** значении бита при отключении двигателя по КО сбрасывается состояние «Резерв» и выставляется состояние «Работа».

При **нулевом** значении бита при отключении двигателя по КО состояние двигателя остается неизменным.

**12<sup>ый</sup> бит OPS\_1** – указывает способ формирования выходного физического сигнала на отключение ИМ при появлении **сигнала** «Двигатель аварийно отключен».

При **единичном** значении бита при возникновении **сигнала** «Двигатель аварийно отключен» из функции будет выдана дублирующая команда на отключение ИМ.

При **нулевом** значении бита дублирующая команда на отключение ИМ при наличии **сигнала** Двигатель аварийно отключен» не выдается.

**ВАЖНО.** Вне зависимости от состояния **12<sup>ого</sup> бита OPS** производится контроль за отключением двигателя. Если по истечении времени контроля смены состояния (вход X19 «Уставка времени контроля смены состояния») двигатель не отключился, выставляется **признак** «Просрочено время на отключение от SMA».

**13<sup>ый</sup> бит OPS\_1** – указывает способ формирования состояние «Резерв» при наличии **признака** «Неисправность KB».

При **единичном** значении бита при наличии **признака** «Неисправность КВ» разрешается смена состояния двигателя на «Резерв» по «КО Резерв» (вход Х14).

При **нулевом** значении бита при наличии **признака** «Неисправность КВ» смена состояния двигателя на «Резерв» не возможна.

**14<sup>ый</sup> бит OPS\_1** – указывает способ формирования состояние «Резерв» при наличии **признака** «Нет питания».

При **единичном** значении бита при наличии **признака** «Нет питания» разрешается смена состояния двигателя на «Резерв» по «КО Резерв» (вход Х14).

При **нулевом** значении бита при наличии **признака** «Нет питания» смена состояния двигателя на «Резерв» не возможна.

**15<sup>ый</sup> бит OPS\_1** – указывает способ сброса (квитации) **признаков** «Отключен ЛЗ» и «Отключен SMA (аварийными защитами)» по «КО Отключить».

При **единичном** значении бита разрешается квитация **признаков** «Отключен ЛЗ» и «Отключен SMA (аварийными защитами)» по «КО Отключить» (вход Х17). При этом повторная команда на отключение двигателя из функции не выдается.

При **нулевом** значении бита квитация **признаков** «Отключен ЛЗ» и «Отключен SMA (аварийными защитами)» возможна по **сигналу** «Сброс сигнализации» (вход Х22) или при выдачи команде на включение.

### **Х3 – OPS\_2.** Описание признаков настройки:

**0<sup>ый</sup> бит OPS\_2** – указывает способ квитации **признака** «Включен ЛЗ» по «КО Включить».

При **единичном** значении бита разрешается квитация **признака** «Включен ЛЗ» «КО Включить» (вход Х16). При этом, повторная команда на включение двигателя из функции не выдается.

При **нулевом** значении бита квитация **признака** «Включен ЛЗ» возможна по **сигналу** «Сброс сигнализации» (вход Х22) или при выдаче команды на отключение.

**1<sup>ый</sup> бит OPS\_2** – указывает способ формирования **признаков** «Нет питания» и «Неисправность КВ».

При **единичном** значении бита формирование **признаков** «Нет питания» и «Неисправность КВ» осуществляется всегда (вне зависимости от состояния входа Х5 (Режим МУ) и **2<sup>ого</sup> бита OPS\_1**).

При **нулевом** значении бита формирование **признаков** «Нет питания» и «Неисправность КВ» осуществляется как указано выше.

**2<sup>ый</sup>, 3<sup>ый</sup>, 4<sup>ый</sup> бит OPS\_2** – указывают способ формирования **признаков** «Двигатель включен» и «Двигатель отключен».

При **нулевом** значении **2<sup>ого</sup> бита OPS\_2** формирование **признаков** «Двигатель включен» и «Двигатель отключен» осуществляется по состоянию КВ. В случае, если по значению КВ нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен», то их значения сбрасываются в ноль. При этом, значение **3<sup>ого</sup> и 4<sup>ого</sup> бита OPS\_2** игнорируются.

При **единичном** значении **2<sup>ого</sup> бита OPS\_2** **признаки** «Двигатель включен» и «Двигатель отключен» формируются в зависимости от **3<sup>ого</sup> бита OPS**.

При **нулевом** значении **3<sup>ого</sup> бита OPS\_2** формирование **признаков** «Двигатель включен» и «Двигатель отключен» осуществляется по состоянию КВ. В случае, если по значению КВ нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен», то они остаются равными последнему достоверному состоянию. При этом, значение **20<sup>ого</sup> бита OPS\_2** игнорируется.

При **единичном** значении **3<sup>ого</sup> бита OPS\_2** **признаки** «Двигатель включен» и «Двигатель отключен» формируются в зависимости от **4<sup>ого</sup> бита OPS\_2**.

При **нулевом** значении **4<sup>ого</sup> бита OPS\_2** формирование **признаков** «Двигатель включен» и «Двигатель отключен» осуществляется по состоянию КВ. В случае, если по значению КВ нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен», то считается, что двигатель отключен.

При **единичном** значении **4<sup>ого</sup> бита OPS\_2** формирование **признаков** «Двигатель включен» и «Двигатель отключен» осуществляется по состоянию КВ. В случае, если по значению КВ нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен», то считается, что двигатель включен.

**ВАЖНО.** Нельзя достоверно сформировать **признаки** «Двигатель включен» и «Двигатель отключен» в том случае, если функцией обрабатываются значения от обоих КВ (**0<sup>ой</sup> бит OPS\_2** равен 1) и значения входов X7 «Двигатель включен» и X8 «Двигатель отключен» равны 0 или 1.

**X4 - Двигатель аварийно отключен.** Сигнал «Двигатель аварийно отключен» (SMA).

Реакция функции на этот сигнал определена в **12<sup>ом</sup> бите OPS\_1**;

**X5 – МУ.** Режим «Местное управление» (Rg\_MU). При единичном значении Rg\_MU

игнорируются КО. При Rg\_MU = 1 будут приняты в обработку вход X6 «МУ «Включить» и вход X7 «МУ «Отключить». Способ обработки определяет **2<sup>ой</sup> бит OPS\_1**;

**X6 - МУ "Включить".** Команда «Включить» при местном управлении (MU\_ON);

**X7 - МУ "Отключить".** Команда «Отключить» при местном управлении (MU\_OFF);

**X8 - Двигатель включен.** Двигатель включен (On). Используется совместно с параметром Off для идентификации состояния двигателя. Реакция функции на этот сигнал определена в **0<sup>ом</sup> бите OPS\_1**;

**X9 - Двигатель отключен.** Двигатель отключен (Off). Используется совместно с параметром On для идентификации состояния двигателя. Реакция функции на этот сигнал определена в **0<sup>ом</sup> бите OPS\_1**;

**X10 - Нет питания.** Нет питания. Реакция функции на этот сигнал определена в **1<sup>ом</sup> бите OPS\_1**;

**X11 - Включить ЛЗ.** Включить логической задачей (ONLZ). На данный вход могут подаваться команды алгоритмического управления (если двигатель

надо включать через определенное время), команда «Включить по АВР» и команды включения двигателя по блокировкам и защита; **X12 - Отключить ЛЗ**. Отключить логической задачей (OFFLZ). На данный вход могут подаваться команды алгоритмического управления (если двигатель надо отключать через определенное время), и команды отключения двигателя по блокировкам и защита; **X13 - КО "Ремонт"**. Команда оператора «Ремонт». Устанавливает состояние двигателя «Ремонт»;

**ВАЖНО.** При установке состояния «Ремонт» ВСЕ управляющие команды из функции блокируются.

**X14 - КО "Резерв"**. Команда оператора «Резерв». Устанавливает состояние двигателя «Резерв» с учетом значений **13<sup>ого</sup> и 14<sup>ого</sup> битов OPS\_1**;

**X15 - КО "Работа"**. Команда оператора «Работа». Устанавливает состояние двигателя «Работа»;

**ВАЖНО.** Смена состояний двигателя оператором возможна только после завершения всех управляющих воздействий на ИМ.

**X16 - КО "Включить"**. Команда оператора «Включить» (KO\_ON);

**X17 - КО "Отключить"**. Команда оператора «Отключить» (KO\_OFF);

**X18 - Запрет включения.** Признак запрета включения двигателя. Реакция функции на этот сигнал определена в **5<sup>ом</sup> бите OPS\_1**;

**X19 - Запрет отключения.** Признак запрета отключения двигателя. Реакция функции на этот сигнал определена в **4<sup>ом</sup> бите OPS\_1**;

**X20 - Задержка на изменение состояния.** Уставка максимального времени на изменение состояния двигателя (Time\_Wait).

**ВАЖНО.** Значение уставки Time\_Wait должно быть больше 0, иначе будет считаться, что смена состояний КВ происходит мгновенно (в течение одного цикла контроллера);

**X21 - Длительность выдачи команд.** Уставка времени выдачи управляющих команд (Time\_Out\_CMD).

**ВАЖНО.** Если уставка Time\_Out\_CMD равна 0, то выходы Y1 «Физическая команда Включить» и Y3 «Физическая команда Отключить» **ПОТЕНЦИАЛЬНЫЕ**, если больше 0 – **ИМПУЛЬСНЫЕ** с временем импульса, равным Time\_Out\_CMD.

**ВАЖНО.** Значение уставки Time\_Out\_CMD не должно превышать значение уставки Time\_Wait. В случае, если Time\_Out\_CMD > Time\_Wait и будет превышено время на изменение состояния ИМ, выходные команды сбросятся до истечения Time\_Out\_CMD;

**X22 - Сброс сигнализации.** Сброс сигнализации. При **единичном** значении входа сбрасываются признаки невыполнения команд и признаки несанкционированного включения/отключения (например, превышено время включения ЛЗ, превышено время отключения при МУ, двигатель несанкционированно включился);

**X23 - Состояние двигателя.** Состояние двигателя (**StatusIm 1**).

**0<sup>ой</sup> бит StatusIm 1** – состояние «Ремонт». Состояние «Ремонт» может выставить только оператор командой «Ремонт» при условии, что ИМ отключен, нет команд оператора перевода в состояние «Резерв» или «Работа» и нет ни одной управляющей команды.



- 1<sup>ый</sup> бит StatusIm 1** – состояние «Резерв». Состояние «Резерв» может выставить только оператор командой «Резерв» при условии, что ИМ отключен, нет команд оператора перевода в состояние «Ремонт» или «Работа» и нет ни одной управляющей команды, а также если нет ошибки КВ или разрешен перевод в состояние «Резерв» при наличии ошибки КВ (**13<sup>ый</sup> бит OPS\_1** равен 1) и нет признака «Нет питания» или разрешен перевод в состояние «Резерв» при наличии признака «Нет питания» (**14<sup>ый</sup> бит OPS\_1** равен 1).
- 2<sup>ой</sup> бит StatusIm 1** – состояние «Работа».
- 3<sup>ий</sup> бит StatusIm 1** – запущен процесс отключения ИМ аварийными защитами ИМ. Данный бит выставляется в 1, если выполняется контроль отключения или выдается дублирующая команда на отключение при наличии **сигнала** «Двигатель аварийно отключен» (SMA).
- 4<sup>ый</sup> бит StatusIm 1** – запущен процесс отключения ИМ командой местного управления (MU\_OFF). Данный бит выставляется в 1, если выполняется отключение ИМ по команде МУ «Отключить».
- 5<sup>ый</sup> бит StatusIm 1** – запущен процесс включения ИМ командой местного управления (MU\_ON). Данный бит выставляется в 1, если выполняется включение ИМ по команде МУ «Включить».
- 6<sup>ой</sup> бит StatusIm 1** – запущен процесс отключения ИМ логической задачей (OFFLZ). Данный бит выставляется в 1, если выполняется отключение ИМ по команде «Отключить ЛЗ».
- 7<sup>ой</sup> бит StatusIm 1** – запущен процесс включения ИМ логической задачей (ONLZ). Данный бит выставляется в 1, если выполняется включение ИМ по команде «Включить ЛЗ».
- 8<sup>ой</sup> бит StatusIm 1** – запущен процесс отключения ИМ командой оператора (KO\_OFF). Данный бит выставляется в 1, если выполняется отключение ИМ по «КО Отключить».
- 9<sup>ый</sup> бит StatusIm 1** – запущен процесс включения ИМ командой оператора (KO\_ON). Данный бит выставляется в 1, если выполняется включение ИМ по «КО Включить».
- 10<sup>ый</sup> бит StatusIm 1** – признак «Нет питания».
- 11<sup>ый</sup> бит StatusIm 1** – признак «Неисправность КВ».
- 12<sup>ый</sup> бит StatusIm 1** – сохраненный сигнал – Отключение аварийными защитами ИМ. Данный бит выставляется в 1, если произошло отключение ИМ при наличии сигнала «Двигатель аварийно отключен» (SMA) (вне зависимости, выдавалась ли дублирующая команда или выполнялся только контроль отключения).
- 13<sup>ый</sup> бит StatusIm 1** – сохраненный сигнал – Отключение ИМ командой местного управления. Данный бит выставляется в 1, если произошло отключение ИМ по команде MU\_OFF.

**14<sup>ый</sup> бит StatusIm 1** – сохраненный сигнал – Включение ИМ командой местного управления. Данный бит выставляется в 1, если произошло включение ИМ по команде MU\_ON.

**15<sup>ый</sup> бит StatusIm 1** – сохраненный сигнал – Отключение ИМ командой ЛЗ. Данный бит выставляется в 1, если произошло отключение ИМ по команде OFFLZ.

### **X24 - Состояние двигателя 2. Состояние двигателя (StatusIm 2).**

**0<sup>ый</sup> бит StatusIm 2** – сохраненный сигнал – Включение ИМ командой ЛЗ. Данный бит выставляется в 1, если произошло включение ИМ по команде ONLZ.

**1<sup>ый</sup> бит StatusIm 2** – сохраненный сигнал – Отключение ИМ КО. Данный бит выставляется в 1, если произошло отключение ИМ по «КО Отключить».

**2<sup>ый</sup> бит StatusIm 2** – сохраненный сигнал – Включение ИМ КО. Данный бит выставляется в 1, если произошло Включение ИМ по «КО Включить».

**3<sup>ый</sup> бит StatusIm 2** – сохраненный сигнал – Несанкционированное отключение ИМ. Данный бит выставляется в 1, если произошло отключение ИМ без команд управления. Контроль осуществляется, если ИМ запитан, нет ошибки KB, **1<sup>ый</sup> бит OPS\_2** равен 1. В случае, если **1<sup>ый</sup> бит OPS\_2** равен 0, то не должно быть признака МУ или управление при МУ через контроллер (**2<sup>ый</sup> бит OPS\_1** равен 1).

**4<sup>ый</sup> бит StatusIm 2** – сохраненный сигнал – Несанкционированное включение ИМ. Данный бит выставляется в 1, если произошло включение ИМ без команд управления. Контроль осуществляется, если ИМ запитан, нет ошибки KB, **1<sup>ый</sup> бит OPS\_2** равен 1. В случае, если **1<sup>ый</sup> бит OPS\_2** равен 0, то не должно быть признака МУ или управление при МУ через контроллер (**2<sup>ый</sup> бит OPS\_1** равен 1).

**5<sup>ый</sup> бит StatusIm 2** – сохраненный сигнал – Просрочено время на отключение ИМ от SMA. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Отключен».

**6<sup>ой</sup> бит StatusIm 2** – сохраненный сигнал – Просрочено время на отключение ИМ от команде MU\_OFF. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Отключен».

**7<sup>ий</sup> бит StatusIm 2** – сохраненный сигнал – Просрочено время на включение ИМ от команде MU\_ON. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Включен».

**8<sup>ый</sup> бит StatusIm 2** – сохраненный сигнал – Просрочено время на отключение ИМ от команде OFFLZ. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Отключен».

**9<sup>ый</sup> бит StatusIm 2** – сохраненный сигнал – Просрочено время на включение ИМ от команде ONLZ. Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Включен».

- 10<sup>ой</sup> бит StatusIm 2** – сохраненный сигнал – Просрочено время на отключение ИМ от «КО Отключить». Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Отключен».
- 11<sup>ой</sup> бит StatusIm 2** – сохраненный сигнал – Просрочено время на включение ИМ от «КО Включить». Данный бит выставляется в 1, если за время **Time\_Wait** ИМ не сменил свое состояние на «Включен».
- 12<sup>ой</sup> бит StatusIm 2** – Выдать команду «Включить».
- 13<sup>ый</sup> бит StatusIm 2** – Выдать команду «Отключить».
- 14<sup>ый</sup> бит StatusIm 2** – признак «Двигатель включен». Данный бит выставляется в 1, если двигатель включен. Данный **признак** определяется в зависимости от значений **2<sup>ого</sup>, 3<sup>ого</sup>, 4<sup>ого</sup> бита OPS\_1** и является информационным. Он не участвует в определении признаков «Нет питания», «Неисправность КВ», «Просрочено время на ...», «Несанкционированное включение/отключение».
- 15<sup>ый</sup> бит StatusIm 2** – признак «Двигатель отключен». Данный бит выставляется в 1, если двигатель отключен. Данный **признак** определяется в зависимости от значений **2<sup>ого</sup>, 3<sup>ого</sup>, 4<sup>ого</sup> бита OPS\_1** и является информационным. Он не участвует в определении признаков «Нет питания», «Неисправность КВ», «Просрочено время на ...», «Несанкционированное включение/отключение».

#### Выходные данные:

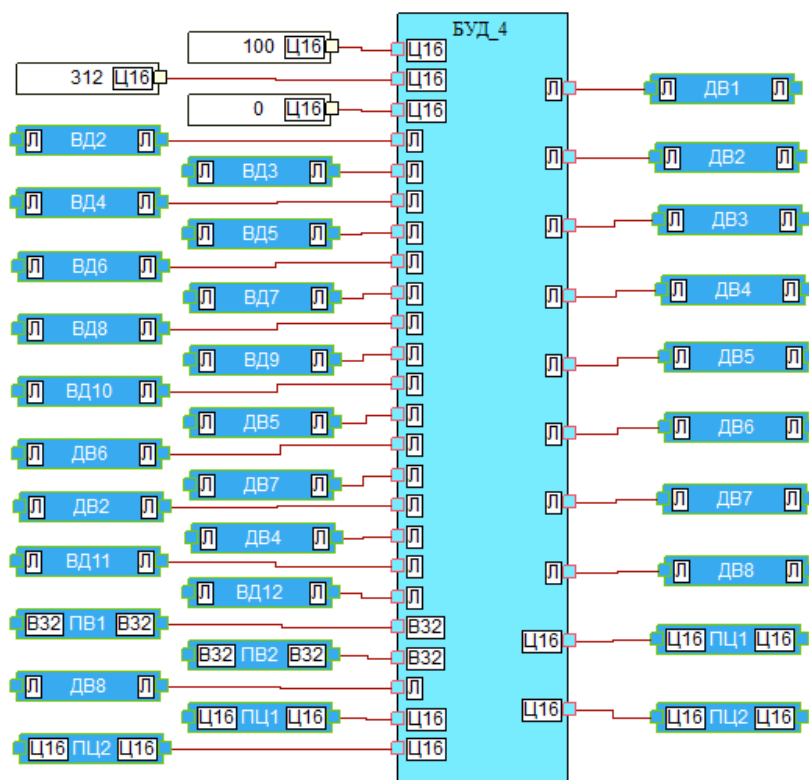
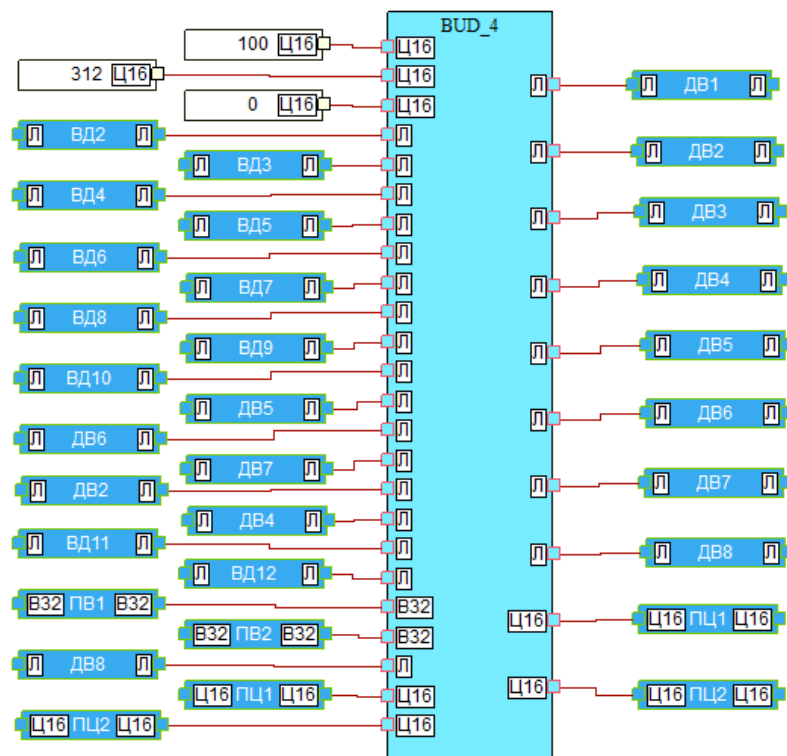
- Y1 - Физ команда "Включить".** Физическая команда «Включить»;
- Y2 - КО "Включить".** Команда оператора «Включить». По выходу из функции всегда равна 0;
- Y3 - Физ команда "Отключить".** Физическая команда «Отключить»;
- Y4 - КО "Включить".** Команда оператора «Отключить». По выходу из функции всегда равна 0;
- Y5 - КО "Ремонт".** Команда оператора «Ремонт». По выходу из функции всегда равна 0;
- Y6 - КО "Резерв".** Команда оператора «Резерв». По выходу из функции всегда равна 0;
- Y7 - КО "Работа".** Команда оператора «Работа». По выходу из функции всегда равна 0;
- Y8 - Сброс сигнализации.** Сброс сигнализации. По выходу из функции всегда равен 0;
- Y9 - Состояние двигателя 1.** Состояние двигателя 1 (первые 16 бит).
- Y10 - Состояние двигателя 2.** Состояние двигателя 2 (вторые 16 бит).

Пример вызова функции	
СТ:	ФБД:
(ДВ1, ДВ2, ДВ3, ДВ4, ДВ5, ДВ6, ДВ7, ДВ8, ПЦ1, ПЦ2) = BUD_3(100, 312, ВД2, ВД3, ВД4, ВД5, ВД6, ВД7, ВД8, ВД9, ВД10, ДВ5, ДВ6, ДВ7,	

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

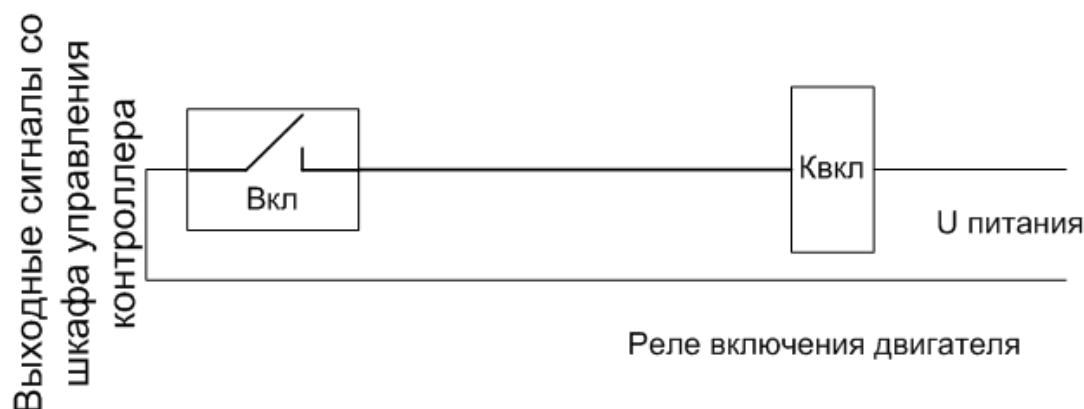
ДВ2, ДВ4, ВД11,  
ВД12, пв1, пв2,  
ДВ8, ПЦ1, ПЦ2)

(ДВ1, ДВ2, ДВ3,  
ДВ4, ДВ5, ДВ6,  
ДВ7, ДВ8,  
ПЦ1, ПЦ2) =  
БУД\_3(100, 312,  
ВД2, ВД3, ВД4,  
ВД5, ВД6, ВД7,  
ВД8, ВД9, ВД10,  
ДВ5, ДВ6, ДВ7,  
ДВ2, ДВ4, ВД11,  
ВД12, пв1, пв2,  
ДВ8, ПЦ1, ПЦ2)



Даная функция предназначена для схем управления двигателем, приведенных ниже:

### Для схемы с одной (потенциальной) управляющей командой



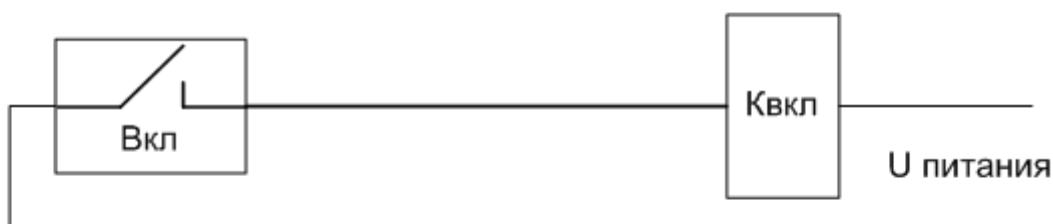
Некоторые настройки кода OPS для схемы с одной управляющей командой:

- OPS\_1 = 0, OPS\_2 = 0:
  - Определение состояния двигателя по 1 KB;
  - Сигнал «Нет питания» не заведен в функцию;
  - МУ не через функцию;
  - Физическая команда «Включить» по сигналу ONLZ всегда;
  - Запрет отключения для всех команд;
  - Запрет включения для всех команд;
  - Количество команд управления – 1;
  - Не сбрасывать команду управления по получению требуемого состояния ИМ (**Для схем с одной командой управления данный бит всегда 0**);
  - При включении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При отключении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При включении ИМ КО «Включить» не сбрасывать состояние «Резерв»;
  - При отключении ИМ КО «Отключить» не сбрасывать состояние «Резерв»;
  - Не выдавать дублирующую команду при SMA;
  - Не разрешать установку состояния «Резерв» при ошибке KB;
  - Не разрешать установку состояния «Резерв» при признаке «Нет питания»;
  - Не разрешать квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
  - Не разрешать квитацию сигнала «Включен ЛЗ» по КО «Включить»;
  - Отключить диагностику при МУ, когда МУ не через контроллер;
  - Определять состояние двигателя по KB;
  - Определять состояние двигателя по последнему достоверному значению;
  - При неисправность KB считать, что двигатель отключен.
- OPS\_1 = 61306 , OPS\_2 = 1 / OPS\_1 = 65402, OPS\_2 = 1:
  - Определение состояния двигателя по 1 KB;
  - Сигнал «Нет питания» заведен в функцию;
  - МУ не через функцию;
  - Физическая команда «Включить» по сигналу ONLZ только в резерве;
  - Запрет отключения для КО;
  - Запрет включения для КО;

- Количество команд управления – 1;
- Не сбрасывать команду управления по получению требуемого состояния ИМ (**Для схем с одной командой управления данный бит всегда 0**);
- При включении ИМ ЛЗ сбрасывать состояние «Резерв»;
- При отключении ИМ ЛЗ сбрасывать состояние «Резерв»;
- При включении ИМ КО «Включить» сбрасывать состояние «Резерв»;
- При отключении ИМ КО «Отключить» сбрасывать состояние «Резерв»;
- Не выдавать дублирующую команду при SMA/Выдавать дублирующую команду при SMA;
- Разрешить установку состояния «Резерв» при ошибке КВ;
- Разрешить установку состояния «Резерв» при признаке «Нет питания»;
- Разрешить квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
- Разрешить квитацию сигнала «Включен ЛЗ» по КО «Включить»;
- Отключить диагностику при МУ, когда МУ не через контроллер;
- Определять состояние двигателя по КВ;
- Определять состояние двигателя по последнему достоверному значению;
- При неисправности КВ считать, что двигатель отключен.

## Для схемы с двумя (потенциальными) управляющими командами

Выходные сигналы со шкафа управления контроллера



Реле включения двигателя



Реле отключения двигателя

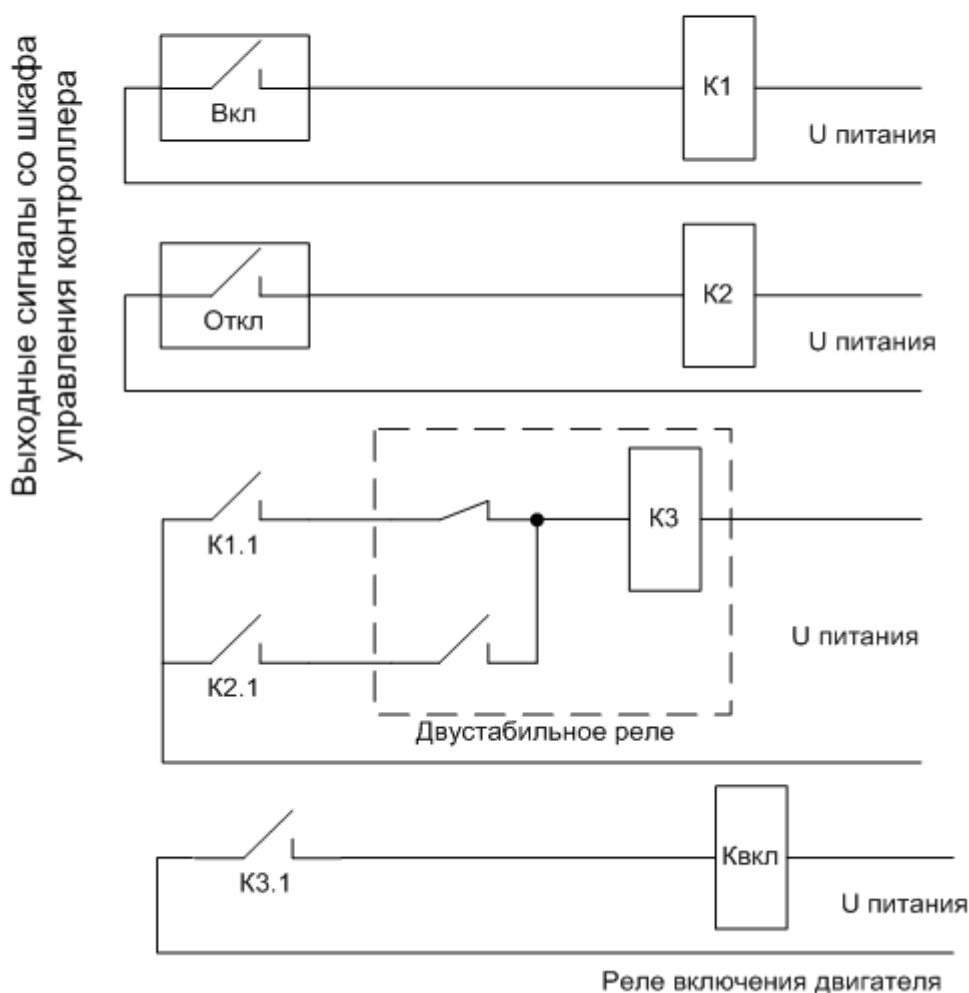
**ВАЖНО:** для данной схемы обязательно нужно, чтоб Time\_Out\_CMD (Уставка максимального времени подачи команды) была равна 0.

Некоторые настройки кода OPS для схемы с двумя потенциальными управляющими командами:

- OPS\_1 = 69, OPS\_2 = 0:
  - Определение состояния двигателя по 2 KB;
  - Сигнал «Нет питания» не заведен в функцию;
  - МУ через функцию;
  - Физическая команда «Включить» по сигналу ONLZ только в резерве;
  - Запрет отключения для всех команд;
  - Запрет включения для всех команд;
  - Количество команд управления – 2;
  - Не сбрасывать команду управления по получению требуемого состояния ИМ (**Для схем с двумя потенциальными командами управления данный бит всегда 0**);
  - При включении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При отключении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При включении ИМ КО «Включить» не сбрасывать состояние «Резерв»;
  - При отключении ИМ КО «Отключить» не сбрасывать состояние «Резерв»;
  - Не выдавать дублирующую команду при SMA;
  - Не разрешать установку состояния «Резерв» при ошибке KB;
  - Не разрешать установку состояния «Резерв» при признаке «Нет питания»;
  - Не разрешать квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
  - Не разрешать квитацию сигнала «Включен ЛЗ» по КО «Включить»;
  - Отключить диагностику при МУ, когда МУ не через контроллер;
  - Определять состояние двигателя по KB;
  - Определять состояние двигателя по последнему достоверному значению;
  - При неисправности KB считать, что двигатель отключен.
- OPS\_1 = 65407, OPS\_2 = 1/ OPS\_1 = 65405, OPS\_2 = 1:
  - Определение состояния двигателя по 2 KB;
  - Сигнал «Нет питания» заведен в функцию/Сигнал «Нет питания» не заведен в функцию;
  - МУ через функцию;
  - Физическая команда «Включить» по сигналу ONLZ только в резерве;
  - Запрет отключения для КО;
  - Запрет включения для КО;
  - Количество команд управления – 2;
  - Не сбрасывать команду управления по получению требуемого состояния ИМ (**Для схем с двумя потенциальными командами управления данный бит всегда 0**);
  - При включении ИМ ЛЗ сбрасывать состояние «Резерв»;
  - При отключении ИМ ЛЗ сбрасывать состояние «Резерв»;
  - При включении ИМ КО «Включить» сбрасывать состояние «Резерв»;
  - При отключении ИМ КО «Отключить» сбрасывать состояние «Резерв»;
  - Выдавать дублирующую команду при SMA;
  - Разрешить установку состояния «Резерв» при ошибке KB;
  - Разрешить установку состояния «Резерв» при признаке «Нет питания»;
  - Разрешить квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;

- Разрешить квитацию сигнала «Включен ЛЗ» по КО «Включить»;
- Отключить диагностику при МУ, когда МУ не через контроллер;
- Определять состояние двигателя по КВ;
- Определять состояние двигателя по последнему достоверному значению;
- При неисправность КВ считать, что двигатель отключен.

### Для схемы с двумя (импульсными) управляющими командами



**ВАЖНО:** для данной схемы обязательно нужно, чтоб Time\_Out\_CMD (Уставка максимального времени подачи команды) была больше 0.

Некоторые настройки кода OPS для схемы с двумя управляющими командами:

- OPS\_1 = 64, OPS\_2 = 0:
  - Определение состояния двигателя по 1 КВ;
  - Сигнал «Нет питания» не заведен в функцию;
  - МУ не через функцию;

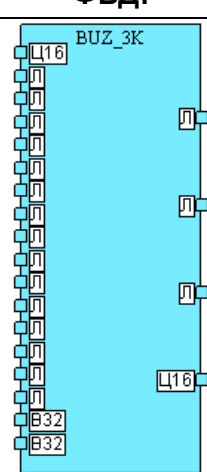


- Физическая команда «Включить» по сигналу ONLZ только в резерве;
  - Запрет отключения для всех команд;
  - Запрет включения для всех команд;
  - Количество команд управления – 2;
  - Не сбрасывать команду управления по получению требуемого состояния ИМ;
  - При включении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При отключении ИМ ЛЗ не сбрасывать состояние «Резерв»;
  - При включении ИМ КО «Включить» не сбрасывать состояние «Резерв»;
  - При отключении ИМ КО «Отключить» не сбрасывать состояние «Резерв»;
  - Не выдавать дублирующую команду при SMA;
  - Не разрешать установку состояния «Резерв» при ошибке KB;
  - Не разрешать установку состояния «Резерв» при признаке «Нет питания»;
  - Не разрешать квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
  - Не разрешать квитацию сигнала «Включен ЛЗ» по КО «Включить»;
  - Отключить диагностику при МУ, когда МУ не через контроллер;
  - Определять состояние двигателя по KB;
  - Определять состояние двигателя по последнему достоверному значению;
  - При неисправность KB считать, что двигатель отключен.
- OPS\_1 = 3, OPS\_2 = 65535 / OPS\_1 = 3, OPS\_2 = 65407:
    - Определение состояния двигателя по 2 KB;
    - Сигнал «Нет питания» заведен в функцию;
    - МУ через функцию;
    - Физическая команда «Включить» по сигналу ONLZ только в резерве;
    - Запрет отключения для КО;
    - Запрет включения для КО;
    - Количество команд управления – 2;
    - Не сбрасывать команду управления по получению требуемого состояния ИМ/ сбрасывать команду управления по получению требуемого состояния ИМ;
    - При включении ИМ ЛЗ сбрасывать состояние «Резерв»;
    - При отключении ИМ ЛЗ сбрасывать состояние «Резерв»;
    - При включении ИМ КО «Включить» сбрасывать состояние «Резерв»;
    - При отключении ИМ КО «Отключить» сбрасывать состояние «Резерв»;
    - Выдавать дублирующую команду при SMA;
    - Разрешить установку состояния «Резерв» при ошибке KB;
    - Разрешить установку состояния «Резерв» при признаке «Нет питания»;
    - Разрешить квитацию сигналов «Отключен ЛЗ» и «Отключен аварийными защитами» по КО «Отключить»;
    - Разрешить квитацию сигнала «Включен ЛЗ» по КО «Включить»;
    - Включить диагностику при МУ, когда МУ не через контроллер;
    - Определять состояние двигателя по KB;
    - Определять состояние двигателя по последнему достоверному значению;
    - При неисправность KB считать, что двигатель отключен.

## 14.6 BUZ\_3K

### Назначение

Управление электроприводом запорной арматуры при возможности дистанционного, автоматического управления и управления по сигналам технологических защит и блокировок, а также контролем времени движения.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4) = BUZ\_3K(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18)$	
<p>Входные параметры: X1(ц16), X2(л), X3(л), X4(л), X5(л), X6(л), X7(л), X8(л), X9(л), X10(л), X11(л), X12(л), X13(л), X14(л), X15(л), X16(л), X17(в32), X18(в32)</p> <p>Выходные параметры: Y1(л), Y2(л), Y3(л), Y4(ц16)</p>	

### Описание

**BUZ\_3K** – функция управления электроприводом запорной арматуры предназначена для использования с имеющими "самоподхват" команды исполнительными механизмами. Функция контролирует наличие начала и окончания движения исполнительного механизма. Электротехническая часть управления этими механизмами должна обеспечивать:

- блокировку (запоминание) команд "Открыть" и "Закрыть";
- "развал" цепей управления исполнительным механизмом при получении команды "Стоп";
- "развал" цепи открытия (закрытия) при обжатии соответствующего конечного выключателя или возникновении перегрузки.

Функция не применяется для «старых» платформ (СРВК версии 6.5 для ОС QNX и LINUX и среда исполнения версии 1.0).

### Логика работы функции

X1 – порядковый номер функции (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1 и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – команда "ОТКРЫТЬ ДУ"

X3 – команда "ЗАКРЫТЬ ДУ"

X4 – команда "СТОП ДУ"

X5 – команда "ОТКРЫТЬ АУ"

X6 – команда "ЗАКРЫТЬ АУ"

X7 – команда "ОТКРЫТЬ ПО БЛОКИРОВКЕ"

X8 – команда "ЗАКРЫТЬ ПО БЛОКИРОВКЕ"

X9 – команда "ОТКРЫТЬ И ЗАПРЕТИТЬ ЗАКРЫТИЕ"

X10 – команда "ЗАКРЫТЬ И ЗАПРЕТИТЬ ОТКРЫТИЕ"  
X11 – команда "РАЗРЕШИТЬ ОТКРЫТИЕ"  
X12 – команда "РАЗРЕШИТЬ ЗАКРЫТИЕ"  
X13 – команда "РАЗРЕШИТЬ АУ"  
X14 – сигнал "НЕ ЗАКРЫТ"  
X15 – сигнал "НЕ ОТКРЫТ"  
X16 – команда "СБРОС КОНТРОЛЯ"  
X17 – параметр настройки "МАКСИМАЛЬНОЕ ВРЕМЯ ДВИЖЕНИЯ"  
X18 – параметр настройки "МАКСИМАЛЬНОЕ ВРЕМЯ СХОДА"

Y1 – команда "ОТКРЫТЬ"  
Y2 – команда "ЗАКРЫТЬ"  
Y3 – команда "СТОП"  
Y4 – состояние задвижки

Функция BUZ\_3K формирует выходные команды "ОТКРЫТЬ" (Y1), "СТОП" (Y2), "ЗАКРЫТЬ" (Y3) длительностью не менее 1 секунды (зависит от длительности входных команд). Длительность выходной команды может быть менее 1с в случае поступления сигнала с более высоким приоритетом, равного по приоритету сигнала на формирование команды противоположного направления. Длительность входных команд может быть различной, но не меньше задаваемого цикла решения задачи управления. Текущее состояние выходных команд Y1, Y2, Y3 определяется в соответствии с приведенным на рисунке алгоритмом исхода из текущего состояния входных сигналов и внутренней памяти алгоритма (RS-триггеров).

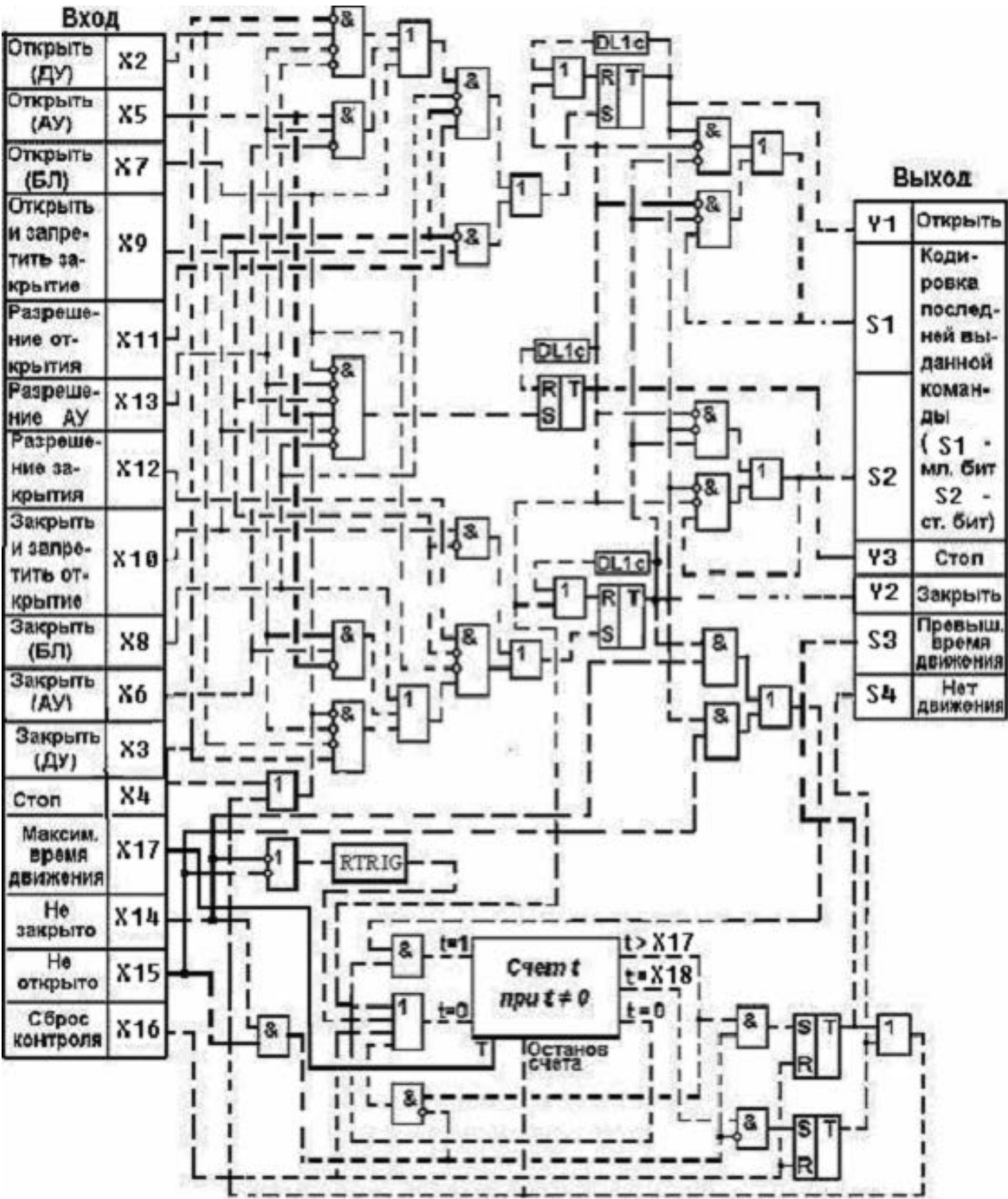
Логика работы схемы определения команды на открытие и закрытие задвижки должна быть следующей: первая пришедшая команда "ОТКРЫТЬ" или "ЗАКРЫТЬ" считается первопричиной и запоминается. После подтверждения исполнения команды (появления соответствующего сигнала "НЕ ОТКРЫТ" X14 или "НЕ ЗАКРЫТ" X15) формируется код состояния задвижки (Y4) (на рисунке не показан).

Контроль времени движения начинает работать по переднему фронту выходных команд "ОТКРЫТЬ" (Y1) или "ЗАКРЫТЬ" (Y3). При этом начинается счет интервала времени движения исполнительного органа. Исполнительный механизм должен сойти с одного из двух концевых выключателей (одновременно должны присутствовать оба входных сигнала "НЕ ОТКРЫТ" X14 и "НЕ ЗАКРЫТ" X15) до истечения времени заданного в параметре настройки "МАКСИМАЛЬНОЕ ВРЕМЯ СХОДА" (X18).

После превышения заданного интервала времени "МАКСИМАЛЬНОЕ ВРЕМЯ ДВИЖЕНИЯ" (X17), должен обжаться другой концевой выключатель – при этом один из входных сигналов сигнала "НЕ ОТКРЫТ" (X15) или "НЕ ЗАКРЫТ" (X14) должен перейти в состояние FALSE.

Процесс контроля прерывается, если на вход функционального блока поступит команда "СТОП" (X4), при этом сформировавшиеся выходные команды контроля сохраняют свое состояние.

Если условия безотказного движения не выполняются, то формируется сигнал "Останов счета", который инициирует выходную команду управления "СТОП" (Y2), то есть движение исполнительного механизма прекращается.

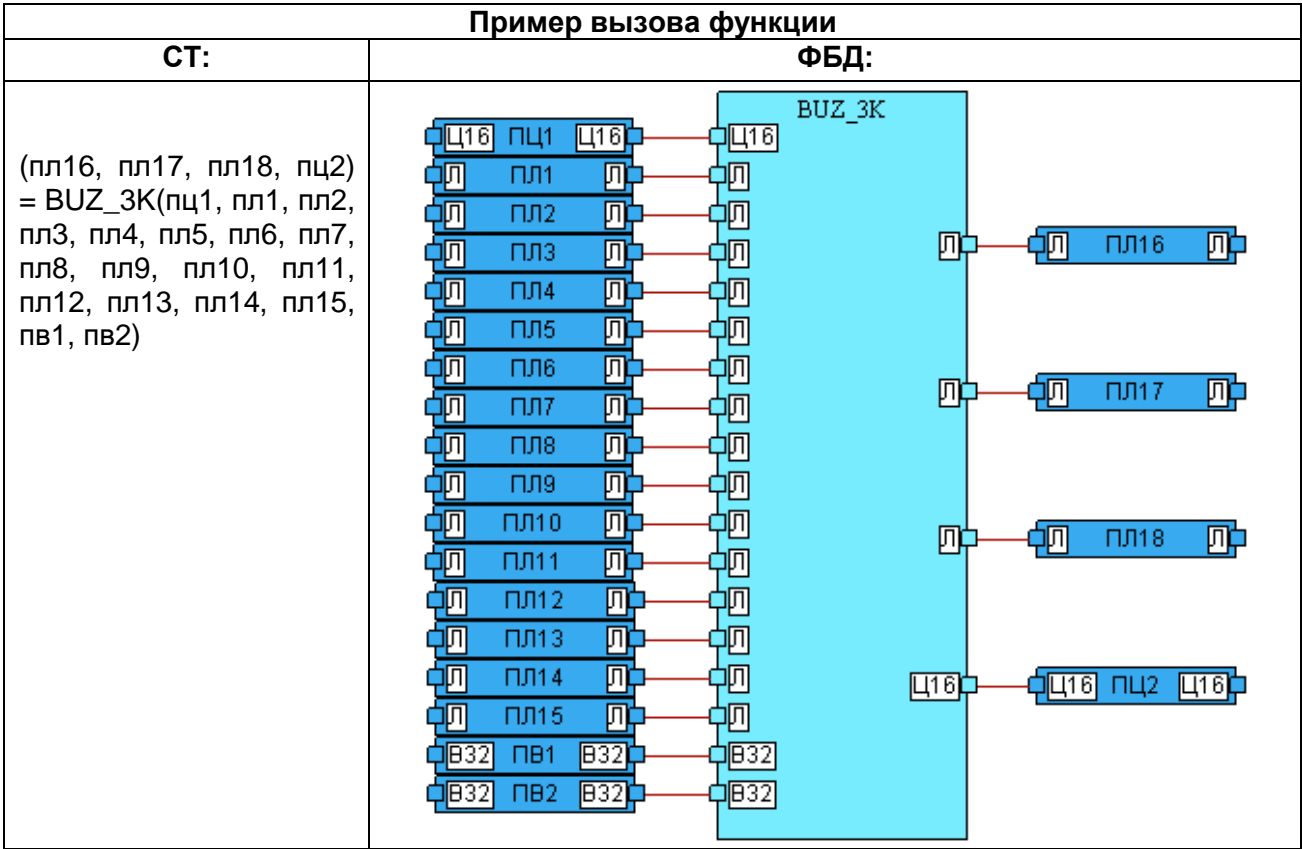


Коды состояния задвижки, передаваемые на операторский уровень (Y4) приведены в таблице:

Состояние задвижки	Код (Y4)
Открыта оператором	0x1
Открыта по автоматике	0x2
Открыта по блокировке	0x4
Открыта	0x7
Закрыта оператором	0x8
Закрыта по автоматике	0x10
Закрыта по блокировке	0x20
Закрыта	0x38
Открывается	0x40
Закрывается	0x80
Нет движения на открытие	0x100
Нет движения на закрытие	0x200
Остановлена оператором	0x400
Остановлена в промежуточном положении	0x800
Промежуточное положение	0xC00
Превышено время открытия	0x1000
Превышено время закрытия	0x2000
Отсутствие питания	0x4000

Кроме того, нужно иметь в виду следующее:

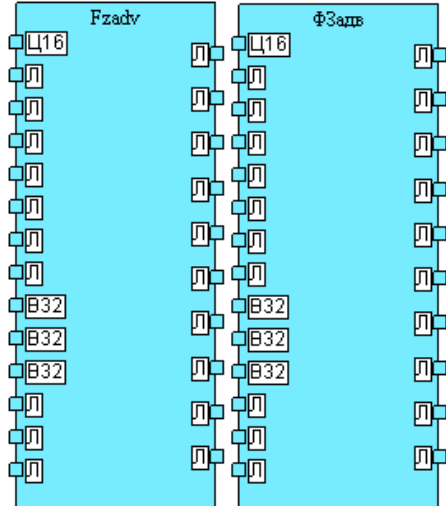
- после срабатывания схемы контроля и присвоения выходу Y4 соответствующего кода состояния задвижки (0x100, 0x200, 0x1000, 0x2000) значение данного выхода не будет изменено, пока не выполнится команда «СБРОС КОНТРОЛЯ» (X16), вне зависимости от изменения сигналов «НЕ ЗАКРЫТ» (X14), «НЕ ОТКРЫТ» (X15), за исключением случая отсутствия питания на исполнительном механизме (X14=X15=0);
- если был осуществлен сброс (YB63) схемы контроля, сработавшей из-за отсутствия схода с концевика по прошествии контрольного времени (X18), выходу Y4 будет присвоен код, который был до выдачи невыполненной команды на открытие/закрытие (возможные варианты: 0x1, 0x2, 0x4, 0x7, 0x8, 0x10, 0x20, 0x38).



## 14.7 ФЗадв, Fzadv

### Назначение

Функция управления электрифицированной задвижкой.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10) = Fzadv(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14)$  $(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10) = \Phi Задв(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14)$	
<p>Входные параметры: X1(ц16), X2(л), X3(л), X4(л), X5(л), X6(л), X7(л), X8(л), X9(в32), X10(в32), X11(в32), X12(л), X13(л), X14(л)</p> <p>Выходные параметры: Y1(л), Y2(л), Y3(л), Y4(л), Y5(л), Y6(л), Y7(л), Y8(л), Y9(л), Y10(л)</p>	

### Описание

X1 – номер задвижки. Диапазон изменения от 1 до 9999.

X2 – команда оператора «Открыть». Значение активного сигнала равно 1.

X3 – команда оператора «Закрыть». Значение активного сигнала равно 1.

X4 – сигнал от концевого выключателя «Не открыто». Значение сигнала при открытом положении задвижки равно 0.

X5 – сигнал от концевого выключателя «Не закрыто». Значение сигнала при закрытом положении задвижки равно 0.

X6 – сигнал подтверждения хода задвижки в сторону открытия. При значении сигнала = 1, задвижка в движении.

X7 – сигнал подтверждения хода задвижки в сторону закрытия. При значении сигнала = 1, задвижка в движении.

X8 – тип выходного сигнала управления. Импульсный - значение = 1. Потенциальный - значение = 0.

X9 – уставка максимального времени хода задвижки. Диапазон изменения 5-1200 с.

X10 – уставка выходного импульса. Имеет смысл только при импульсном виде выходного сигнала управления. Диапазон изменения 100-9000 мс.

X11 – уставка времени ожидания подтверждения хода задвижки. Диапазон изменения 1-480 с.

X12 – сигнал непосредственного воздействия на задвижку в сторону открытия от технологической защиты. В данном алгоритме используется для блокировки закрытия задвижки оператором. Значение = 1 сигнал открытия от технологической защиты.

X13 – сигнал непосредственного воздействия на задвижку в сторону закрытия от технологической защиты. В данном алгоритме используется для блокировки открытия задвижки оператором. Значение = 1 сигнал закрытия от технологической защиты.

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

X14 – тип контроля выполнения команды. Контроль от пускателя – 0. Контроль от времени схода с концевого выключателя – 1.

Y1 – выходная команда "Открыть с контроллера".

Y2 – команда оператора «Открыть».

Y3 – выходная команда "Закреть с контроллера".

Y4 – команда оператора «Закреть».

Y5 – сигнал ошибки при управлении задвижкой.

Y6 – превышение времени хода задвижки.

Y7 – нет подтверждения хода.

Y8 – самоход.

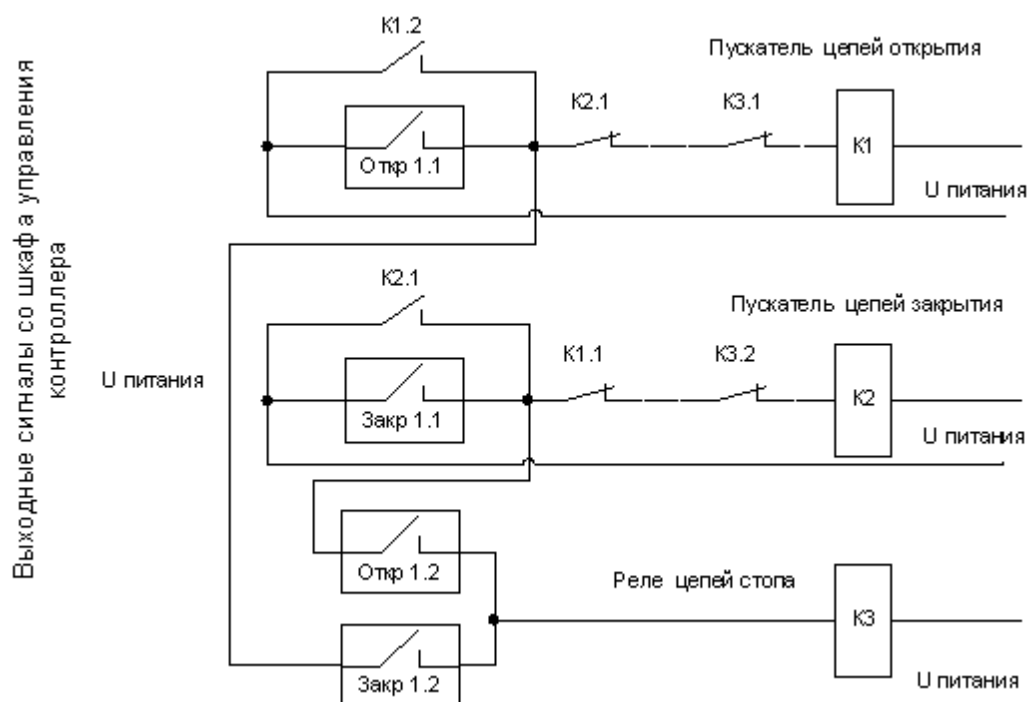
Y9 – запрет кнопки «Закреть».

Y10 – запрет кнопки «Открыть».

Функция выдает в роллинг сообщений следующие сообщения:

1. Задвижка № Разобрана
2. Задвижка № Перейти на ручное управление
3. Задвижка № Запрет открытия
4. Задвижка № Запрет закрытия
5. Задвижка № Превышено время хода
6. Задвижка № Нет подтверждения хода
7. Задвижка № не контроллерное управление
8. № неверный номер задвижки
9. Задвижка № ОШ ТЗ\_О и ТЗ\_3=1

Функция реализована для схемы с самоподхватом в сборке РТЗО с косвенным выходом на стоп





Функция применяется только для платформ СРВК любой версии.

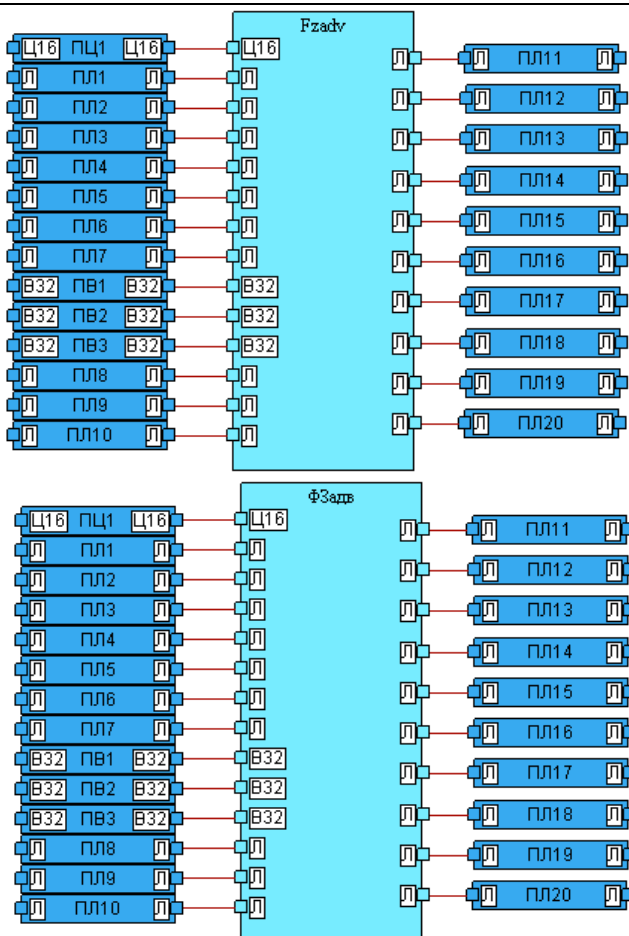
### Пример вызова функции

СТ:

(пл11,пл12,пл13,пл14,пл15,пл16,пл17,пл18,  
пл19,пл20)=Fzadv(пц1,пл1,пл2,пл3,пл4,пл5,  
пл6,пл7,пв1,пв2,пв3,пл8,пл9,пл10)

(пл11,пл12,пл13,пл14,пл15,пл16,пл17,пл18,  
пл19,пл20)=ФЗадв(пц1,пл1,пл2,пл3,пл4,пл5,  
,пл6,пл7,пв1,пв2,пв3,пл8,пл9,пл10)

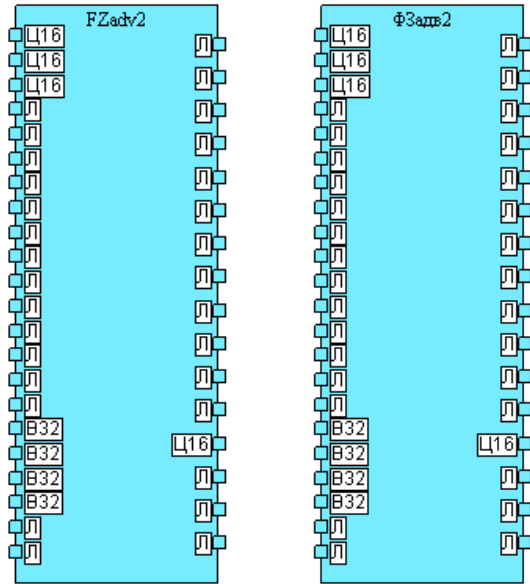
ФБД:



## 14.8 ФЗадв2, FZadv2

### Назначение

Функция управления электрифицированной задвижкой.

Отображение	
СТ:	ФБД:
<p>(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13, Y14, Y15, Y16) = FZadv2(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20, X21, X22)</p> <p>(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13, Y14, Y15, Y16) = ФЗадв2(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14, X15, X16, X17, X18, X19, X20, X21, X22)</p>	
<p>Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(л), X5(л), X6(л), X7(л), X8(л), X9(л), X10(л), X11(л), X12(л), X13(л), X14(л), X15(л), X16(л), X17(в32), X18(в32), X19(в32), X20(в32), X21(л), X22(л)</p> <p>Выходные параметры: Y1(л), Y2(л), Y3(л), Y4(л), Y5(л), Y6(л), Y7(л), Y8(л), Y9(л), Y10(л), Y11(л), Y12(л), Y13(ц16), Y14(л), Y15(л), Y16(л)</p>	

### Описание

X1 – номер алгоблока. Для вывода номера 4<sup>ый</sup> и 8<sup>ый</sup> биты X3 должны быть установлены в 1.

X2 – идентификатор задвижки. Используется для идентификации задвижки по присвоенному ей номеру. Выводится в сообщении, формируемом функцией.

X3 – описание признаков настройки.

- ☐ 0<sup>ый</sup> бит X3 - сигнал от X11 в функцию заведен - да/нет (1/0). При установке этого бита в 1 функцией будет обрабатываться аргумент X11. При установке этого бита в 0 безразличное значение аргумента X11.
- ☐ 1<sup>ый</sup> бит X3 - крайнее состояние концевого выключателя (КВ) разомкнуто – да/нет (1/0). Возможность настройки функции для арматуры, у которой крайнее состояние КВ замкнуто.
- ☐ 2<sup>ой</sup> бит X3 - задвижка с самоподхватом в РТЗО - да/нет (1/0). Возможность настройки функции для конкретной схемной реализации сборок РТЗО. При установке этого бита в 0 «подхват» организуется в функции.
- ☐ 3<sup>ий</sup> бит X3 - управляющее действие на арматуру при ТЗ - из функции/самостоятельный сигнал (1/0). При установке этого бита в 1 функцией будет выдан управляющий сигнал на движение, в зависимости от типа ТЗ. При установке этого бита в 0 функцией будет производиться только контроль

перемещения задвижки. Управляющий сигнал в этом случае должен быть сформирован другими управляющими цепями.

- ☐ 4<sup>ый</sup> бит X3 - вывод в сообщении - номер/идентификатор (1/0). Определяет выводимый в роллинг аргумент функции X1 или X2. При установке этого бита в 1 в роллинг выводится номер, при 0 – идентификатор. Более предпочтительно устанавливать этот бит в 0, т.к. реальный номер алгоблока может не совпадать с заданным в параметре X1.
- ☐ 5<sup>ый</sup> бит X3 - контроль движения от КВ - да/нет (1/0). При установке этого бита в 1 контроль движения организуется от параметров X7 и X8.
- ☐ 6<sup>ой</sup> бит X3 - контроль движения от пускателя - да/нет (1/0). При установке этого бита в 1 контроль движения организуется от параметров X9 и X10.



**Внимание!!!**

**5<sup>ый</sup> и 6<sup>ый</sup> биты X3 одновременно не должны быть равны 0.**

- ☐ 7<sup>ой</sup> бит X3 - ввод сигнала от пускателя - индивидуальный/обобщенный (1/0). При установке этого бита в 1 аргумент функции X9 - сигнал от пускателя «заккрыть», X10 - сигнал от пускателя «открыть». При установке этого бита в 0 аргумент функции X9 - обобщенный сигнал от пускателей. Значение аргумента X10 безразличное.
- ☐ 8<sup>ой</sup> бит X3 – вывод сообщения из функции - да/нет (1/0). При установке этого бита в 0 сообщения из функции не выводятся и для формирования сообщения необходимо использовать функцию message.
- ☐ 9<sup>ый</sup> бит X3 – местное управление через контроллер - да/нет (1/0). Указывает, как будет выдаваться управляющее воздействие от кнопки с местного щита управления. При установке этого бита в 1 сигнал от кнопки должен быть заведен в контроллер и дальнейшее управляющее воздействие будет выдано функцией. При установке этого бита в 0 команда от кнопки поступает непосредственно в схему сборки РТЗО.
- ☐ 10<sup>ый</sup> бит X3 – непосредственный выход на стоп - да/нет (1/0). При установке этого бита в 1 выходной сигнал стопа выдается непосредственно в схему стопового реле. При установке этого бита в 0 выходной сигнал стопа выдается в цепь противоположную текущей выполняемой команды.

X4 – команда оператора "Открыть" (виртуальная);

X5 – команда оператора "Заккрыть" (виртуальная);

X6 – команда оператора "Стоп" (виртуальная);

X7 – сигнал от концевого выключателя "Открыто". Используется совместно с параметром X8 для идентификации положения задвижки.

X8 – сигнал от концевого выключателя "Заккрыто". Используется совместно с параметром X7 для идентификации положения задвижки.

X9 – сигнал от пускателя №1 (обобщенный/заккрыть). Обрабатывается функцией в зависимости от значения 7<sup>го</sup> бита X3.

X10 – сигнал от пускателя №2 (0/открыть). Обрабатывается функцией в зависимости от значения 7<sup>го</sup> бита X3.

X11 – сигнал от автомата питания Вкл/Выкл. Используется для определения состояния задвижки. Обрабатывается функцией в зависимости от значения 0<sup>го</sup> бита X3.

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

---

X12 – признак местного управления задвижкой. При установке X12 блокируется управление задвижкой со станции оператора. При X12 = 1 будут приняты в обработку параметры X21 и

X22. Способ обработки определяет 9<sup>ый</sup> бит X3.

X13 – команда технологической защиты "ОТКРЫТЬ";

X14 – команда технологической защиты "ЗАКРЫТЬ";

X15 – признак наличия технологической блокировки ОТКРЫТИЯ;

X16 – признак наличия технологической блокировки ЗАКРЫТИЯ;

X17 – уставка максимального времени движения задвижки;

X18 – уставка максимального времени ожидания начала движения задвижки;

X19 – уставка максимального времени затяга задвижки (только на закрытие);

X20 – уставка времени действия выходного сигнала в цепи управления пускателем;

X21 – признак открытия от местного управления;

X22 – признак закрытия от местного управления.

Установка 9<sup>го</sup> бита X3 в 0, определяет для функции признак управление задвижкой с местного щита управления без выдачи управляющих воздействий из функции. Для корректной работы функции (контроль хода, выдача состояния и т.д.) на эти входа требуется подавать сигналы, либо от кнопок управления местного щита управления либо сигналы от каждого из пускателя. Если местное управление существует, а требуемые параметры в контроллер не заводят, то для отмены контроля движения необходимо аргументам X21 и X22 присвоить значение 1.

Приоритет команд управления:

1. X13 и X14.
2. X15 и X16.
3. X4, X5, X6, X21, X22.

Y1 – реальный сигнал «Открыть». Y1 = 1 после анализа возможности выполнения команды;

Y2 – команда оператора открыть (виртуальная);

Y3 – реальный сигнал «Закрыть». Y3 = 1 после анализа возможности выполнения команды;

Y4 – команда оператора закрыть (виртуальная);

Y5 – реальный сигнал «Стоп». Y5 = 1 после анализа возможности выполнения команды;

Y6 – команда оператора стоп (виртуальная);

Y7 – признак ошибки «задвижка разобрана»;

Y8 – признак ошибки «превышение T хода задвижкой»;

Y9 – признак ошибки «нет подтверждения хода задвижки»;

Y10 – признак ошибки «не контроллерный ход задвижки»;

Y11 – признак открытия задвижки;

Y12 – признак закрытия задвижки;

Y13 – код сообщения согласно номеру из таблицы 8. В Y13 может быть записано максимум 3 кода сообщений. На каждое сообщение 5 бит;

Y14 – признак положение задвижки открыта;

Y15 – признак положение задвижки закрыта;

Y16 – признак положение задвижки среднее.

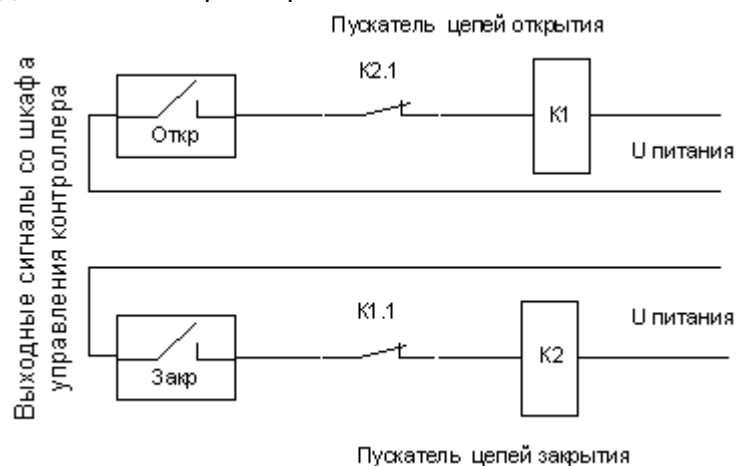
Функция выдает в роллинг сообщений следующие сообщения:

1. Здв № ОШ ТЗ\_О и ТЗ\_З=1
2. Здв № ОШ ТБ\_О и ТБ\_З=1
3. Задвижка № Разобрана
4. Здв № ОШ от КВ.
5. Здв № Отмена команды по ОШ1.
6. Здв № Запрет открытия.

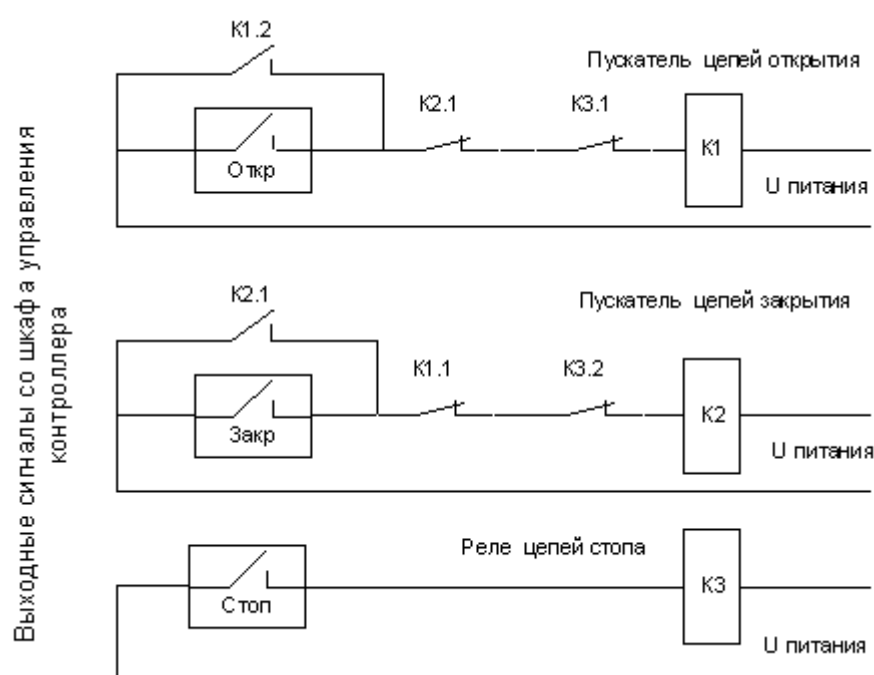
7. Здв №Запрет закрытия.
8. Здв № Запрет останова.
9. Здв № Кмд.Закр уже выполняется.
10. Здв № Кмд.Откр уже выполняется.
11. Здв № Отмена команды ОТКРЫТЬ.
12. Здв № Отмена команды ЗАКРЫТЬ.
13. Здв № МУ. Отмена команды.
14. Здв № Закрыта. Отмена команды.
15. Здв № Открыта. Отмена команды.
16. Здв № нет сигн.от пускателя.
17. Здв № нет смены состояния от КВ.
18. Здв № Нет подтверждения хода.
19. Здв № нет параметров контроля.
20. Здв № Превышено время хода.
21. Здв № не контроллерное управл.
22. Здв № Стоп по ОШ1.
23. № неверный номер задвижки.
24. Здв № Настрочный параметр неверен.
25. Здв № Перейти на руч. управл

Функция реализована:

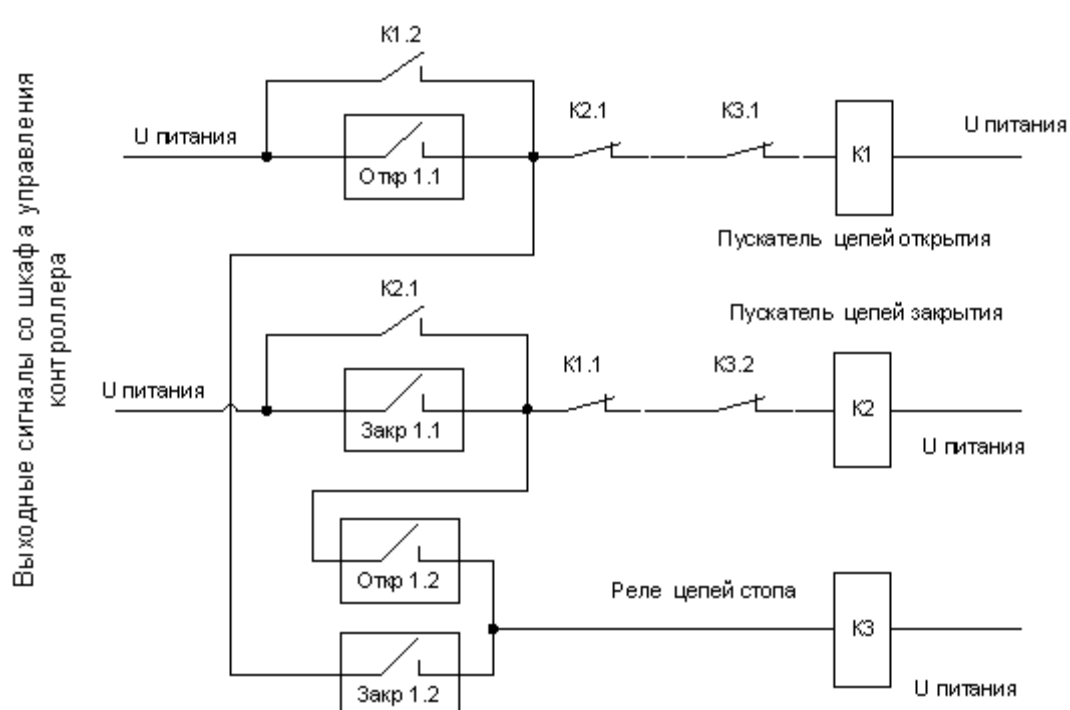
для схемы с самоподхватом в контроллере



для схемы с самоподхватом в сборке РТЗО с непосредственным выходом на стоп



для схемы с самоподхватом в сборке РТ30 с косвенным выходом на стоп



Функция применяется только для платформ СРВК любой версии.

Пример вызова функции	
СТ:	ФБД:
<p>(пл16, пл17, пл18, пл19, пл20, пл21, пл22, пл23, пл24, пл25, пл26, пл27, пц4, пл25, пл26, пл27) = FZadv2(пц1, пц2, пц3, пл1, пл2, пл3, пл4, пл5, пл6, пл7, пл8, пл9, пл10, пл11, пл12, пл13, пв1, пв2, пв3, пв4, пл14, пл15)</p>	
<p>(пл16, пл17, пл18, пл19, пл20, пл21, пл22, пл23, пл24, пл25, пл26, пл27, пц4, пл25, пл26, пл27) = ФЗadv2(пц1, пц2, пц3, пл1, пл2, пл3, пл4, пл5, пл6, пл7, пл8, пл9, пл10, пл11, пл12, пл13, пв1, пв2, пв3, пв4, пл14, пл15)</p>	

## 14.9 Сообщение, Message

### Назначение

Вывод сообщений в протокол событий

Отображение	
СТ:	ФБД:
<p>Message (X1 [,X2, ... ,Xn])</p> <p>Сообщение (X1 [,X2, ... ,Xn])</p>	 <p>В качестве примера рассмотрена функция с двумя входами.</p>
<p>Входные параметры: X1(*), X2(*)</p> <p><math>X_i</math> – входной параметр, который может быть:</p> <ul style="list-style-type: none"> <li>• строковой константой, записанной в кавычках (например, message("текст сообщения"));</li> <li>• именем переменной БД, записанным в виде &lt;Тип&gt;&lt;Номер&gt; (например, message(va11));</li> <li>• именем локальной или глобальной переменной (например, message(v32_A));</li> <li>• обозначением атрибута переменной (например, message(va1.a1));</li> <li>• числовой явной константой, локальной или глобальной константой.</li> </ul>	

### Описание

Каждый параметр преобразуется в строку. Длина результирующей строки не должна превышать 59 символов.

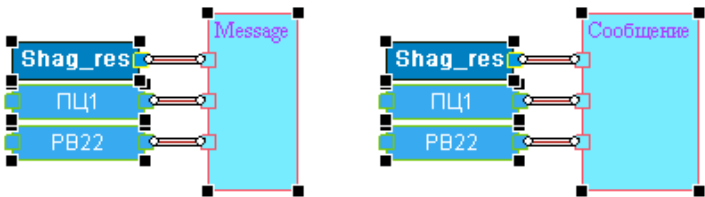
В ФБД у данной функции можно задать произвольное количество входных параметров неопределенного типа (см. раздел 5.4.2.1 «Вставка элементов» книги «Модульная интегрированная SCADA КРУГ-2000. КРУГОЛ. Интегрированная среда разработки»).

Выводит сообщение в протокол событий.

### Логика работы функции

Инструкция для записи сообщения в роллинг в виде текста и значений переменных  
Результат выполнения инструкции, приведенной в примере, на видеокадре "ПРОТОКОЛ СОБЫТИЙ" отображается в следующем виде:

Дата Время Шаг, результат: пц1,рв22= 6 764.984

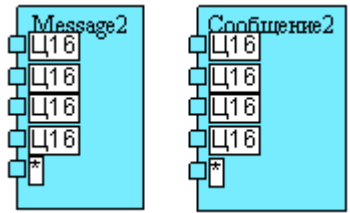
Пример вызова функции	
СТ:	ФБД:
<p>Message ("Шаг, результат: пц1,рв22=",пц1,рв22)</p> <p>Сообщение ("Шаг, результат: пц1,рв22=",пц1,рв22)</p>	 <p>где Shag_res – глобальная строковая константа со значением «Шаг, результат: пц1, рв22=».</p>



## 14.10 Сообщение2, Message2

### Назначение

Функция Message2 предназначена для вывода сообщений в протокол событий. Текст сообщения берется из словаря.

Отображение	
СТ:	ФБД:
Message2 (X1,X2,X3,X4[,X5, ..., Xn])  Сообщение2 (X1,X2,X3,X4[,X5, ..., Xn])	
Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16)[,X5(*),..., Xn(*)]	

### Описание

X1 - по коду словаря определяется словарь, из которого необходимо взять событие.

X2 - по коду события определяется событие в словаре.

X3 - тип переменной.

Значение	Тип переменной
1	Входная аналоговая
2	Входная дискретная
3	Ручной ввод
6	Дискретная выходная
8	Аналоговая выходная

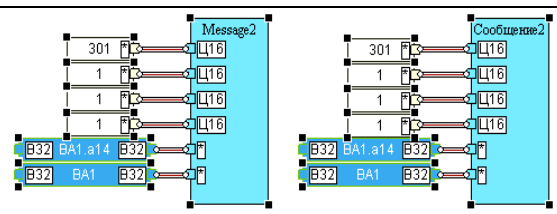
X4 - номер переменной.

X3, X4 задают переменную, с которой связано событие. Если событие не связано с переменной, то эти параметры должны быть равны 0.

X5 - Xn(\*) - список параметров, которые будут подставляться в текст сообщения.

Количество этих дополнительных входов должно совпадать с количеством параметров события, заданного параметром X2, из словаря, заданного параметром X1. Использование дополнительных входов описано в разделе 5.4.2.1 «Вставка элементов» книги «Модульная интегрированная SCADA КРУГ-2000. КРУГОЛ. Интегрированная среда разработки.».

Данная функция не применяется для «старых» платформ (СРВК верии 6.5 для ОС QNX и LINUX и среда исполнения версии 1.0).

Пример вызова функции	
СТ:	ФБД:
Message2(301,1,1,1,Ba1.a14, Ba1)  Сообщение2(301,1,1,1,Ba1.a14,Ba1)  где: <ul style="list-style-type: none"> <li>301 – код словаря (Системные события).</li> <li>1 – код события в словаре (ВПГ=%-.4f значение ПРМ=%-.4f).</li> </ul>	

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

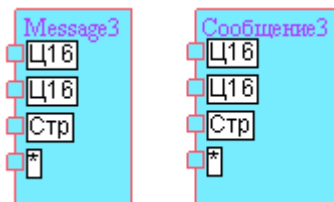
---

<ul style="list-style-type: none"><li>• 1 – тип переменной (ВА).</li><li>• 1 – номер переменной.</li><li>• Ba1.a14 – первый параметр события (значение ВПГ)</li><li>• Ba1 – второй параметр события (текущее значение параметра)</li></ul>	
--	--

### 14.11 Сообщение3, Message3

#### Назначение

Функция message3 предназначена для вывода сообщений в протокол событий. Текст сообщения берется из словаря.

Отображение	
СТ:	ФБД:
Сообщение3 (X1,X2,X3[,X4, ..., Xn])	
Message3 (X1,X2,X3[,X4, ..., Xn])	
Входные параметры: X1(ц16), X2(ц16), X3(стр)[, X4(*),.....Xn(*)]	

#### Описание

X1 - по коду словаря определяется словарь, из которого необходимо взять событие.

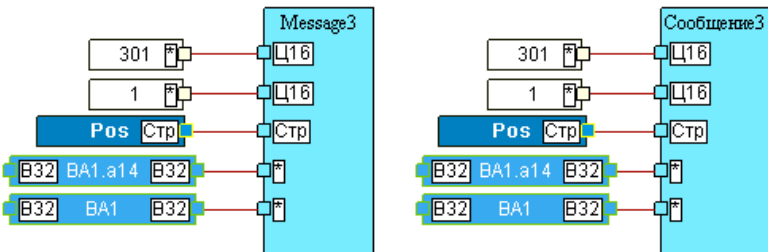
X2 - по коду события определяется событие в словаре.

X3 - позиция переменной

X4 - Xn(\*) - список параметров, которые будут подставляться в текст сообщения.

Количество этих дополнительных входов должно совпадать с количеством параметров события, заданного параметром X2, из словаря, заданного параметром X1. Использование дополнительных входов описано в разделе "5.4.2.1 «Вставка элементов» книги «Модульная интегрированная SCADA КРУГ-2000. КРУГОЛ. Интегрированная среда разработки».

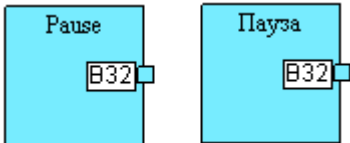
Данная функция не применяется для «старых» платформ (СРВК версии 6.5 для ОС QNX и LINUX и среда исполнения версии 1.0).

Пример вызова функции	
СТ:	ФБД:
<p>Message3(301, 1, "ТКТ13-876", ва1.а14, ва1)</p> <p>Сообщение3(301, 1, "ТКТ13-876", ва1.а14, ва1)</p> <p>где:</p> <ul style="list-style-type: none"> <li>• 301 – код словаря (Системные события).</li> <li>• 1 – код события в словаре (ВПГ=%-.4f значение ПРМ=%-.4f).</li> <li>• "ТКТ13-876" – позиция переменной.</li> </ul> <p>ВА1.а14 – первый параметр сообщения (значение ВПГ).</p> <p>Ва1 - второй параметр события (текущее значение параметра).</p>	 <p>Где: Pos – глобальная строковая константа со значением "ТКТ13-876".</p>

## 14.12 Пауза, Pause

### Назначение

Функция чтения времени простоя системы.

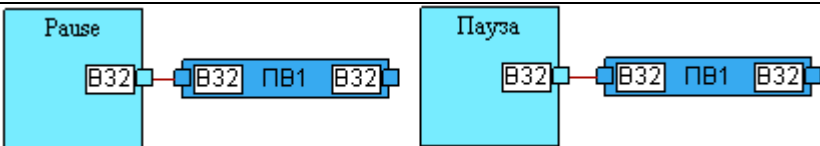
Отображение	
СТ:	ФБД:
Y = Пауза() Y = Pause()	
Выходные параметры: Y(В32)	

### Логика работы функции

Функция выдает результат простоя системы, как число срабатываний системного счетчика минут в вещественном формате.

Y может иметь три варианта значений:

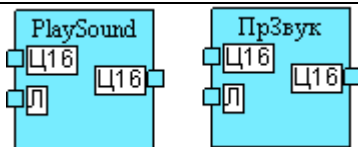
1. положительное число - число срабатываний счетчика минут.
2. ноль - простой в течение одной и той же минуты;
3. отрицательное число - текущее время запуска системы меньше того, которое было в момент последнего останова (при переводе времени).

Пример вызова функции	
СТ:	ФБД:
пв1= Пауза( ) пв1= Pause( )	

## 14.13 ПрЗвук, PlaySound

### Назначение

Функция воспроизведения звука.

Отображение	
СТ:	ФБД:
$Y = \text{ПрЗвук}(X1, X2)$ $Y = \text{PlaySound}(X1, X2)$	
Входные параметры: X1(ц16), X2(л) Выходные параметры: Y(ц16)	

### Описание

Функция обеспечивает воспроизведение wav-файла (waveform-audio), имя которого ассоциировано с целочисленным идентификатором, заданным в качестве первого входного параметра. Формирование ассоциации между целочисленным идентификатором и именем воспроизводимого файла обеспечивается с помощью файла PlaySound.ini. Данный файл должен находиться поддиректории Bin директории установки системы «КРУГ-2000». Файл PlaySound.ini должен иметь следующий формат:

"<Полный путь к файлу 1>" = <Целочисленный идентификатор файла 1>  
 "<Полный путь к файлу 2>" = <Целочисленный идентификатор файла 2>  
 "<Полный путь к файлу N>" = <Целочисленный идентификатор файла N>

Второй входной параметр инициирует проверку наличия воспроизведения звука данной функцией в текущий момент времени. Если он равен 0, проверка не выполняется, функция начинает воспроизводить заданный файл в любом случае. Если параметр равен 1 и функция еще не завершила воспроизведение звука – воспроизведение не прерывается, заданный файл не воспроизводится.

В результате своей работы функция возвращает единственный выходной параметр, который может быть использован для анализа возможных ошибочных ситуаций. Выходной параметр может принимать следующие значения:

- 0 – успешное начало воспроизведения файла;
- 1 – в текущий момент уже воспроизводится звук (в результате предыдущих вызовов функций PlaySound или PlaySoundFile);
- 2 – отсутствует файл, ассоциированный с соответствующим целочисленным идентификатором;
- 3 – заданный целочисленный идентификатор не является уникальным в пределах ini-файла;
- 4 – идентификатор принимает недопустимые значения;
- 5 – ошибка формата ini-файла;
- 6 – ошибка воспроизведения файла.

Данная функция не применяется для платформ СРВК любой версии.

ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Пример вызова функции	
СТ:	ФБД:
пц2 = PlaySound(пц1, пл1)	
пц2 = ПрЗвук(пц1, пл1)	

## 14.14 ПрЗвукФайл, PlaySoundFile

**Назначение**

Функция воспроизведения звукового файла.

Отображение	
СТ:	ФБД:
$Y = \text{ПрЗвукФайл}(X1, X2)$ $Y = \text{PlaySoundFile}(X1, X2)$	
Входные параметры: X1(стр), X2(л) Выходные параметры: Y(ц16)	

**Описание**

Функция обеспечивает воспроизведение wav-файла (waveform-audio).

Первый входной параметр задает имя воспроизводимого звукового файла.

Второй входной параметр инициирует проверку наличия воспроизведения звука данной функцией в текущий момент времени. Если он равен 0, проверка не выполняется, функция начинает воспроизводить заданный файл в любом случае. Если параметр равен 1 и функция еще не завершила воспроизведение звука – воспроизведение не прерывается, заданный файл не воспроизводится.

В результате своей работы функция возвращает единственный выходной параметр, который может быть использован для анализа возможных ошибочных ситуаций. Выходной параметр может принимать следующие значения:

0 – успешное начало воспроизведения файла;

1 – в текущий момент уже воспроизводится звук (в результате предыдущих вызовов функций PlaySound или PlaySoundFile);

6 – ошибка воспроизведения файла.

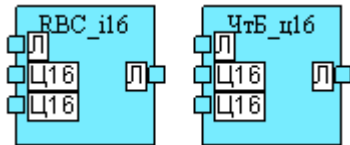
Данная функция не применяется для платформ CPVK любой версии и среды исполнения КРУГОЛ версии 1.0.

Пример вызова функции	
СТ:	ФБД:
$\text{Пц2} = \text{PlaySoundFile}(\text{"D:\Temp\SoundFileName.wav"}, \text{пл1})$  $\text{Пц2} = \text{ПрЗвукФайл}(\text{"D:\Temp\SoundFileName.wav"}, \text{пл1})$	<p>Где: FileName – глобальная строковая константа со значением " D:\Temp\SoundFileName.wav ".</p>

## 14.15 ЧтБ\_ц16, RBC\_i16

### Назначение

Считать значение из указанного бита переменной целого 16-битного формата.

Отображение	
СТ:	ФБД:
$Y1 = RBC\_i16(X1, X2, X3)$ $Y1 = ЧтБ\_ц16(X1, X2, X3)$	
Входные параметры: X1(л), X2(ц16), X3(ц16) Выходные параметры: Y1(л)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

### Логика работы функции

Функция считывает значение из указанного бита переменной целого формата.

X1 – значение переменной до работы функции. Это значение присваивается выходу Y1 при некорректном значении входа X3.

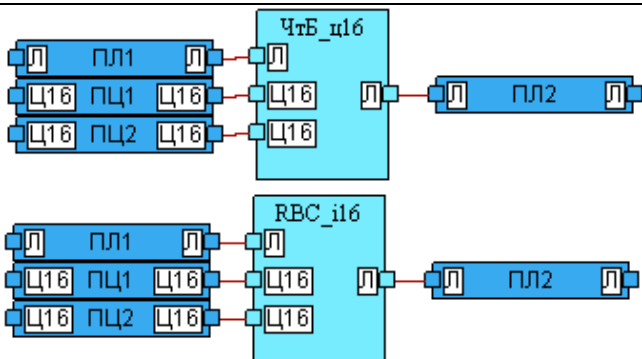
X2 – переменная целого формата, из которой требуется считать значение указанного бита. Номер бита определяет следующий аргумент функции.

X3 – номер бита, определяет, из какого бита второго аргумента необходимо считать значение. Значение аргумента X3 должно лежать в диапазоне от 1 до 16. Здесь 1 – обращение к младшему биту, 16 – к старшему.

Y1 – значение заданного бита.

### Обработка ошибок

Если значение, переданное на вход X3, меньше 1 или больше 16, то на выход Y1 подается значение с входа X1 и в роллинг выводится сообщение «КРУГОЛ:RBC Ош.значен.аргум.X3».

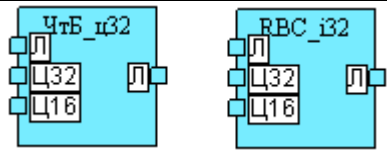
Пример вызова функции	
СТ:	ФБД:
$пл2 = ЧтБ\_ц16(пл1, пц1, пц2)$ $пл2 = RBC\_i16(пл1, пц1, пц2)$	



## 14.16 ЧтБ\_ц32, RBC\_ц32

**Назначение**

Считать значение из указанного бита переменной целого 32-битного формата.

Отображение	
СТ:	ФБД:
$Y1 = \text{ЧтБ\_ц32}(X1, X2, X3)$ $Y1 = \text{RBC\_ц32}(X1, X2, X3)$	
Входные параметры: X1(л), X2(ц32), X3(ц16) Выходные параметры: Y1(л)	

**Описание**

Применяется только для платформ CPBK, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

**Логика работы функции**

Функция считывает значение из указанного бита переменной целого формата.

X1 – значение переменной до работы функции. Это значение присваивается выходу Y1 при некорректном значении входа X3.

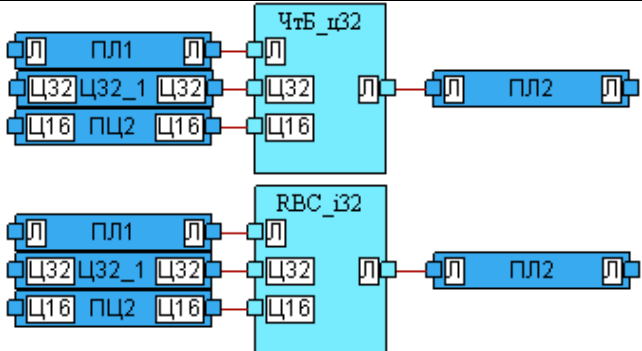
X2 – переменная целого 32-битного формата, из которой требуется считать значение указанного бита. Номер бита определяет следующий аргумент функции.

X3 – номер бита, определяет, из какого бита второго аргумента необходимо считать значение. Значение аргумента X3 должно лежать в диапазоне от 1 до 32. Здесь 1 – обращение к младшему биту, 32 – к старшему.

Y1 – значение заданного бита.

**Обработка ошибок**

Если значение, переданное на вход X3, меньше 1 или больше 32, то на выход Y1 подается значение с входа X1 и в роллинг выводится сообщение «КРУГОЛ:RBC Ош.значен.аргум.X3».

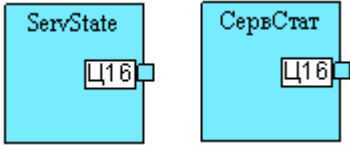
Пример вызова функции	
СТ:	ФБД:
$\text{пл2} = \text{ЧтБ\_ц32}(\text{пл1}, \text{ц32\_1}, \text{пц2})$ $\text{пл2} = \text{RBC\_ц32}(\text{пл1}, \text{ц32\_1}, \text{пц2})$	

Здесь ц32\_1 – глобальная переменная целого 32-битного типа.

## 14.17 СервСтат, ServState

### Назначение

Чтение состояния сервера.

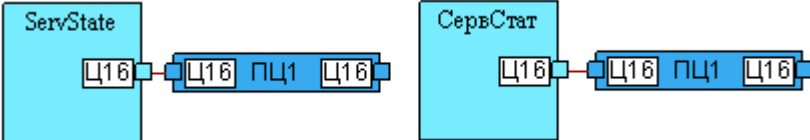
Отображение	
СТ:	ФБД:
$Y = \text{ServState}()$ $Y = \text{СервСтат}()$	
Выходные параметры: Y(ц16)	

### Описание

Функция производит считывание состояния сервера. Не применяется для платформ СРВК любой версии.

### Логика работы функции

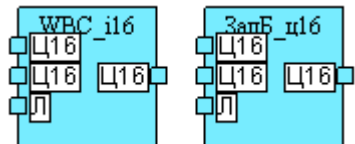
Функция возвращает значение текущего статуса сервера БД в целом формате. Текущий статус сервера БД может принимать значения: «0» - резервный, «1» - основной. Сервер, не описанный в группе резервируемых и зеркализуемых серверов (в форме «Настройка зеркализации» Генератора Базы Данных), всегда имеет статус основного сервера.

Пример вызова функции	
СТ:	ФБД:
$\text{пц1} = \text{СервСтат}()$ $\text{пц1} = \text{ServState}()$	

## 14.18 ЗапБ\_ц16, WBC\_i16

**Назначение**

Записать логическое значение в указанный бит переменной целого формата.

Отображение	
СТ:	ФБД:
$Y1 = \text{ЗапБ\_ц16}(X1, X2, X3)$ $Y1 = \text{WBC\_i16}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(ц16), X3(л) Выходные параметры: Y1(ц16)	

**Описание**

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

**Логика работы функции**

Функция записывает логическое значение в указанный бит переменной целого формата.

X1 – переменная целого формата, в которую требуется записать данные в указанный бит. Номер бита определяет следующий аргумент функции.

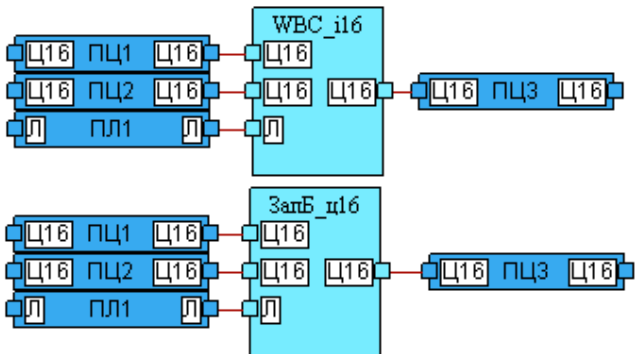
X2 – номер бита, определяет, в какой бит первого аргумента необходимо записать значение. Значение аргумента X2 должно лежать в диапазоне от 1 до 16. Здесь 1 – обращение к младшему биту, 16 – к старшему.

X3 – значение, записываемое в указанный бит первого аргумента.

Y1 – измененное значение первого аргумента функции.

**Обработка ошибок**

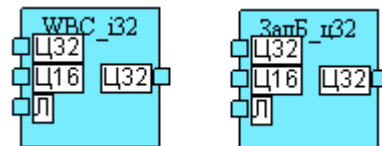
Если значение, переданное на вход X2, меньше 1 или больше 16, то в роллинг выводится сообщение «КРУГОЛ:WBC Ош.значен.аргум.X2».

Пример вызова функции	
СТ:	ФБД:
$\text{пц3} = \text{WBC\_i16}(\text{пц1}, \text{пц2}, \text{пл1})$ $\text{пц3} = \text{ЗапБ\_ц16}(\text{пц1}, \text{пц2}, \text{пл1})$	

## 14.19 ЗапБ\_ц32, WBC\_ц32

### Назначение

Записать логическое значение в указанный бит переменной целого 32-битного формата.

Отображение	
СТ:	ФБД:
$Y1 = WBC\_ц32(X1, X2, X3)$ $Y1 = ЗапБ\_ц32(X1, X2, X3)$	
Входные параметры: X1(ц32), X2(ц16), X3(л) Выходные параметры: Y1(ц32)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2,.

### Логика работы функции

Функция записывает логическое значение в указанный бит переменной целого формата.

X1 – переменная целого 32-битного формата, в которую требуется записать данные в указанный бит. Номер бита определяет следующий аргумент функции.

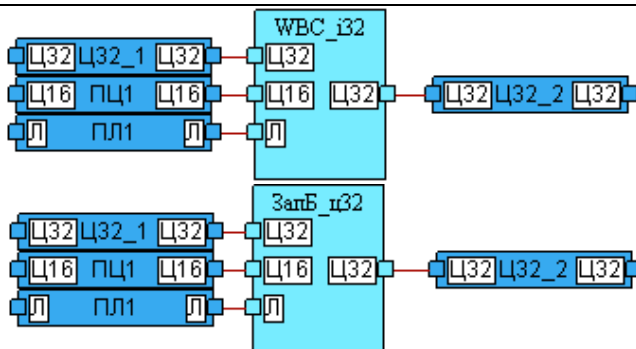
X2 – номер бита, определяет, в какой бит первого аргумента необходимо записать значение. Значение аргумента X2 должно лежать в диапазоне от 1 до 32. Здесь 1 – обращение к младшему биту, 32 – к старшему.

X3 – значение, записываемое в указанный бит первого аргумента.

Y1 – измененное значение первого аргумента функции.

### Обработка ошибок

Если значение, переданное на вход X2, меньше 1 или больше 32, то в роллинг выводится сообщение «КРУГОЛ:WBC Ош.значен.аргум.X2».

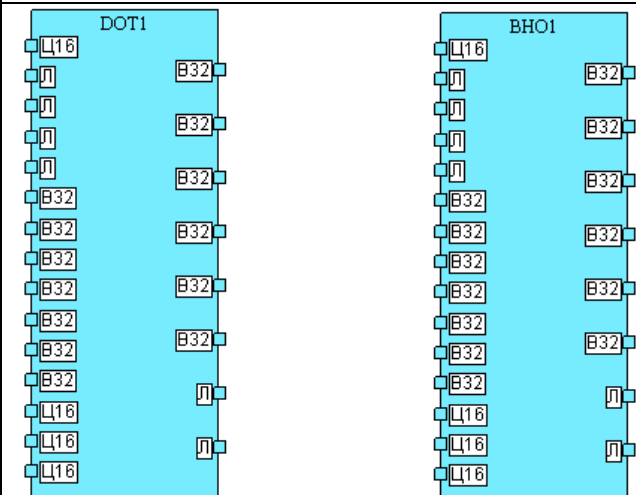
Пример вызова функции	
СТ:	ФБД:
$ц32\_2 = WBC\_ц32(ц32\_1, пц1, пл1)$ $ц32\_2 = ЗапБ\_ц32(ц32\_1, пц1, пл1)$	

Здесь ц32\_1, ц32\_2 – глобальные переменные целого 32-битного типа.

## 14.20 ВНО1, DOT1

### Назначение

Время наработки оборудования.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) =$ $ВНО1(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11,$ $X12, X13, X14, X15)$  $(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8)$ $=DOT1(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11,$ $X12, X13, X14, X15)$	
<p>Входные параметры: X1(ц16), X2(л), X3(л), X4(л), X5(л), X6(в32), X7(в32), X8(в32), X9(в32), X10(в32), X11(в32), X12(ц16), X13(ц16), X14(ц16)</p> <p>Выходные параметры: Y1(в32), Y2(в32), Y3(в32), Y4(в32), Y5(в32), Y6(в32), Y7(л), Y8(л)</p>	

### Описание

Примечание:

Некоторые входные переменные одновременно являются и входными и выходными, например: X4 и Y7, X6 и Y1.

Функция «Время наработки оборудования», реализует стандартный алгоритм расчёта времени наработки оборудования.

Применяется как для платформ СРВК любой версии, так и среды исполнения КРУГОЛ любой версии.

Функция выполняет следующие действия:

- Расчёт времени наработки оборудования с момента обнуления счётчика;
- Автоматическое обнуление счётчика (при установке автоматического сброса счётчика);
- Определение состояния оборудования (включено / выключено);
- Запоминание времени и даты обнуления счётчика;
- Запоминание даты и времени последнего включения \ выключения оборудования.

### Логика работы функции

X1 – порядковый номер функции (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – сигнал «Оборудование включено». X2=1 – оборудование включено, X2=0 – оборудование выключено.

X3 – сигнал «Оборудование выключено». X3=1 – оборудование выключено. Используется при X14=1.

X4 – сигнал «Сброс счётчика наработки» ( $X4=1$ ). Используется при ручном сбросе счётчика наработки.

X5 – команда «Провести расчёт». Используется при внешнем режиме задания периода расчёта времени наработки (если параметр настройки «Режим работы с внешним таймером»  $X15=1$ ).  $X5=1$  – провести расчёт.

X6 – время наработки оборудования с момента последнего сброса счётчика (считывается из переменной при запуске). Такая же переменная, как и Y1.

X7 – дата последнего изменения состояния оборудования (вкл/выкл) (считывается из переменной при запуске). Такая же переменная, как и Y2.

X8 – время последнего изменения состояния оборудования (вкл/выкл) (считывается из переменной при запуске). Такая же переменная, как и Y3.

X9 – дата последнего сброса счётчика (считывается из переменной при запуске). Такая же переменная, как и Y4.

X10 – время последнего сброса счётчика (считывается из переменной при запуске). Такая же переменная, как и Y5.

X11 – состояние оборудования (вкл/выкл). Считывается из переменной при запуске. Такая же переменная, как и Y6.

X12 – период расчёта (выходных данных). Учитывается в режиме работы с внутренним таймером ( $X15=0$ ). Выходные параметры функции ВНО1 будут меняться с периодом X12 (в сек).

X13 – режим обнуления счётчика наработки. Задаёт условия обнуления счётчика наработки.

0 – счётчик сбрасывается только вручную сигналом X4 («Сброс счётчика наработки»).

1 – счётчик сбрасывается автоматически каждый понедельник в 00ч. 02мин.

2 – счётчик сбрасывается автоматически первого числа каждого нового месяца в 00ч. 02мин.

3 – счётчик сбрасывается автоматически первого числа каждого нового года в 00ч. 02мин.

2 минуты даются для формирования печатного документа.

В режимах автоматического сброса счётчика сохраняется возможность обнулить счётчик вручную.

X14 – параметр «Режим». При  $X14=0$  состояние оборудование (вкл/выкл) определяется по одной входной переменной X2 – сигнал «Оборудование включено». При  $X14=1$  состояние оборудование (вкл/выкл) определяется по двум сигналам: X2 и X3.

X15 – параметр, позволяющий использовать внешний таймер для выдачи команды «Провести расчёт» (X5). При  $X15=0$ , расчёт выходных параметров (Y) функции ВНО1 проводится с периодом, определённым параметром X12, с использованием внутреннего таймера. При  $X15=1$  параметр X12 не используется, а расчёт производится при  $X5=1$ . После проведения расчёта переменной X5 присваивается значение 0.

Y1 – время наработки оборудования (с момента последнего сброса счётчика).

Y2 – дата последнего изменения состояния оборудования (вкл/выкл).

Y3 – время последнего изменения состояния оборудования (вкл/выкл).

Y4 – дата последнего сброса счётчика.

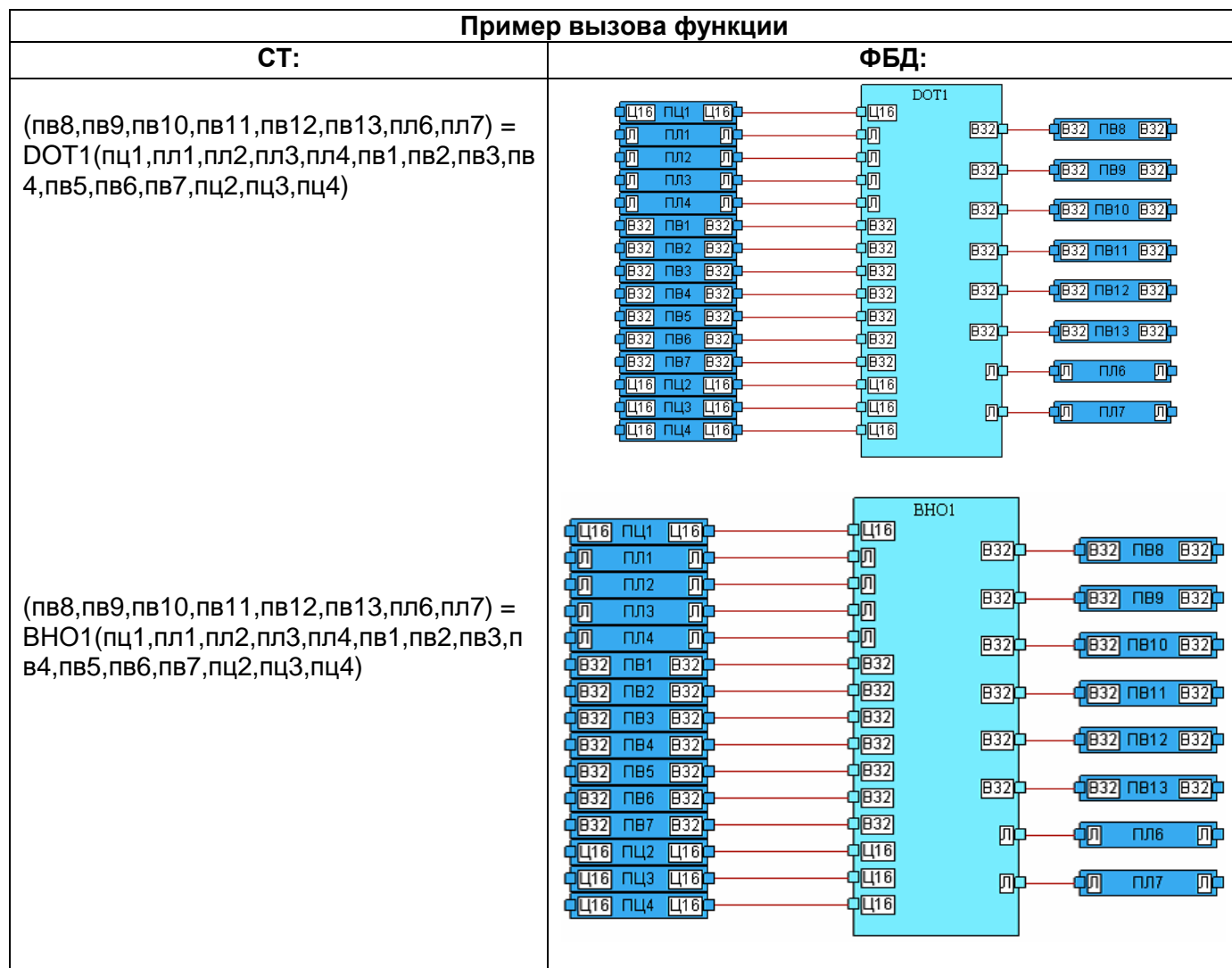
Y5 – время последнего сброса счётчика.

Y6 – состояние оборудования (вкл/выкл). Определяется по двум переменным (если настроенный параметр  $X14=1$ ): X2 (сигнал «Оборудование включено») и X3 (сигнал «Оборудование выключено»); и по одной переменной X2 (сигнал «Оборудование включено»), если настроенный параметр  $X14=0$ . В первом случае состоянию оборудование «ВКЛ» ( $Y6=1$ ) соответствует выполнение двух условий  $X2=1$  и  $X3=0$ . Во всех остальных сочетаниях – состояние оборудования – «ВЫКЛ» ( $Y6=0$ ). Во втором случае состоянию

оборудование «ВКЛ» ( $Y6=1$ ) соответствует выполнение условия  $X2=1$ . При  $X2=0$  состояние оборудования – «ВЫКЛ» ( $Y6=0$ ).

$Y7$  – сигнал «Сброс счётчика наработки». Используется при ручном сбросе счётчика наработки. Такая же переменная, как и  $X4$ .

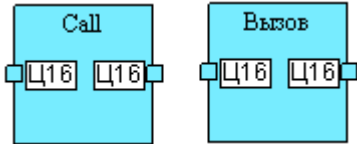
$Y8$  – команда «Провести расчёт». Такая же переменная, как и  $X5$ .



14.21      Вызов, Call

**Назначение**

Вызов ПрП из списка программ в файле programs.lst контроллера.

Отображение	
СТ:	ФБД:
$Y = \text{Call}(X)$ $Y = \text{Вызов}(X)$	
Входные параметры: X(ц16) Выходные параметры: Y(ц16)	

**Описание**

Применяется только для платформ СРВК любой версии.

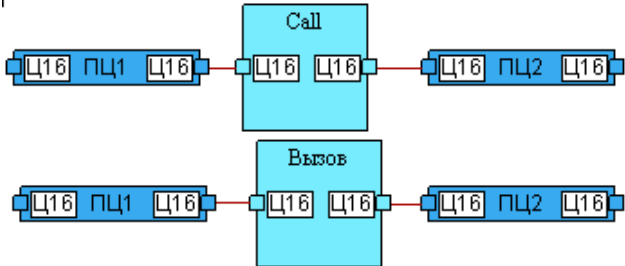
**Логика работы функции**

Вызов ПрП из списка программ в файле programs.lst контроллера.

X –целое число, определяющее номер вызываемой программы из списка файла programs.lst.

Первая ПрП, имеющая условный номер "0", вызывается на выполнения по умолчанию. Из нее, с помощью функции Вызов, организуется вызов на выполнение любой другой ПрП, содержащейся в файле programs.lst.

Y – любая переменная целого 16-битного типа типа .

Пример вызова функции	
СТ:	ФБД:
$\text{пц1} = \text{Call}(\text{пц2})$ $\text{пц1} = \text{Вызов}(\text{пц2})$	



## 14.22 Квит, Confirm

### Назначение

Функция предназначена для квитирования всех событий, связанных с переменной.

Отображение	
СТ:	ФБД:
Confirm (X1,X2)	<div><div>Confirm</div><div><div>Ц16</div><div>Ц16</div></div></div> <div><div>Квит</div><div><div>Ц16</div><div>Ц16</div></div></div>
Квит (X1,X2)	
Входные параметры: X1(ц16), X2(ц16)	

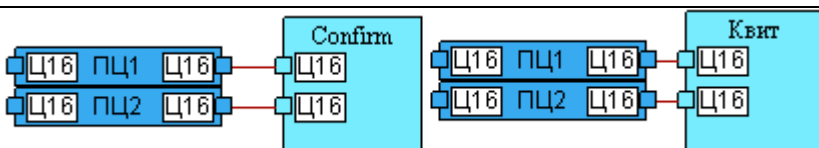
### Описание

X1 - тип переменной:

Значение	Тип переменной
1	Входная аналоговая
2	Входная дискретная
3	Ручной ввод
6	Дискретная выходная
8	Аналоговая выходная

X2 - номер переменной.

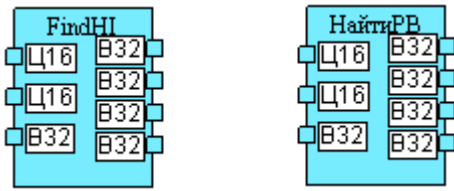
Данная функция не применяется для платформ СРВК любой версии и среда исполнения КРУГОЛ версии 1.0.

Пример вызова функции	
СТ:	ФБД:
Confirm(1, 13)	
Квит(1, 13)	
где: 1 – тип переменной (ВА) 13 – номер переменной.	

## 14.23 НайтиРВ, FindHI

### Назначение

Функция ищет переменную ручного ввода.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4) = \text{FindHI}(X1, X2, X3)$ $(Y1, Y2, Y3, Y4) = \text{НайтиРВ}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(ц16), X3(в32)	
Выходные параметры: Y1(в32), Y2(в32), Y3(в32), Y4(в32)	

### Описание

X1 - начальный номер.

X2 - кол-во.

X3 - значение.

Y1 - значение\_рв.

Y2 - значение\_рв2.

Y3 - значение\_рв\_max.

Y4 - значение\_рв\_min.

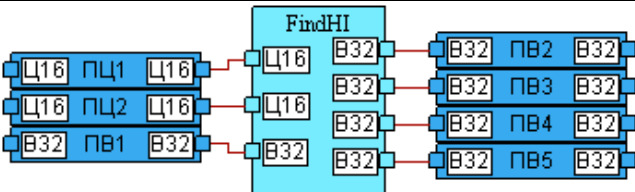
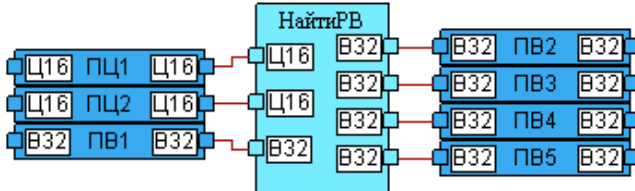
Данная функция применяется только для платформ «СРВК...».

### Логика работы функции

Функция НайтиРВ – ищет переменную ручного ввода, начиная с номера переменной **"Начальный номер + кол-во"** в диапазоне **"кол-во"-1 включительно..**

В случае нахождения переменной РВ, имеющей текущее значение меньше или равное параметру **"значение"**, возвращает значения атрибутов переменной РВ вещественного типа.

Если в указанном диапазоне поиска будут переменные РВ не вещественного типа, результат поиска будет неопределен.

Пример вызова функции	
СТ:	ФБД:
$(\text{пв2}, \text{пв3}, \text{пв4}, \text{пв5}) = \text{FindHI}(\text{пц1}, \text{пц2}, \text{пв1})$	
$(\text{пв2}, \text{пв3}, \text{пв4}, \text{пв5}) = \text{НайтиРВ}(\text{пц1}, \text{пц2}, \text{пв1})$	

## 14.24 ОбрД, ProcD

**Назначение**

Обработка сигналов дублированных и троированных датчиков, формирование по ним обобщенных параметров.

Отображение	
СТ:	ФБД:
<p>(Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,Y11,Y12,Y13,Y14,Y15) = ProcD(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12,X13)</p> <p>(Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,Y11,Y12,Y13,Y14,Y15) = ОбрД(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10,X11,X12,X13)</p>	
<p>Входные параметры: X1(ц16), X2(в32), X3(в32), X4(в32), X5(л), X6(л), X7(л), X8(л), X9(л), X10(л), X11(л), X12(в32), X13(в32)</p> <p>Выходные параметры: Y1(л), Y2(л), Y3(л), Y4(л), Y5(л), Y6(л), Y7(л), Y8(л), Y9(л), Y10(л), Y11(л), Y12(л), Y13(в32), Y14(в32), Y15(ц16)</p>	

**Описание**

Для корректной работы функции необходимо, чтобы:

- параметры X8 и Y9 были одной и той же переменной, если 2<sup>0й</sup> бит параметра X1 равен 1 (кнопки выбора привязаны к дискретным переменным);
- параметры X9 и Y10 были одной и той же переменной, если если 2<sup>0й</sup> бит параметра X1 равен 1 (кнопки выбора привязаны к дискретным переменным);
- параметры X10 и Y11 были одной и той же переменной, если если 2<sup>0й</sup> бит параметра X1 равен 1 (кнопки выбора привязаны к дискретным переменным);
- параметры X11 и Y12 были одной и той же переменной, если если 2<sup>0й</sup> бит параметра X1 равен 1 (кнопки выбора привязаны к дискретным переменным);
- параметры X12 и Y13 были одной и той же переменной при любых условиях;

Данная функция применяется только для платформ СРВК любой версии.

**Логика работы функции**

Функция предназначена для обработки сигналов дублированных и троированных датчиков, формирования по ним обобщенных параметров. Данные обработанные обобщенные параметры могут быть использованы для вывода информации на отображение оператору и ввод информации в регулятор.

X1 – Описание признаков настройки.

**0 бит OPS** – Сброс ошибок рассогласования по конкретным датчикам, при выставлении общей ошибки рассогласования (да-1/нет-0).

- 1 бит OPS** – Переход на автоматический расчет результирующего значения при недостоверности выбранного датчика (да-1/нет-0).
- 2 бит OPS** – Кнопки выбора привязаны к дискретным переменным / к аналоговой переменной (да-1/нет-0).
- 3 бит OPS** – Есть возможность выбора недостоверного датчика (да-1/нет-0).
- 4 бит OPS** – Обрабатываем троированные / дублированные датчики (да-1/нет-0).
- 5 бит OPS** – Расчетному значению присваивается (с учетом достоверности каждого датчика) - среднее арифметическое значение датчиков / медианное значение датчиков (для троированных датчиков) (да-1/нет-0).
- 6 бит OPS** – При недостоверном значении одного из датчиков результирующим значением будет - значение первого / второго датчика из оставшихся достоверных (да-1/нет-0). (6 бит анализируется при значении 5 и 8 битов OPS = 0).
- 7 бит OPS** – Обработка расчетного значения внутри функции / вне функции (да-1/нет-0).
- 8 бит OPS** - При недостоверном значении одного из датчиков результирующим значением будет – среднее арифметическое значение показаний двух оставшихся достоверных датчиков / результирующее значение вычисляется в зависимости от состояния 6 бита OPS (да-1/нет-0). (8 бит анализируется при значении 5 бита OPS = 0).

X2 – показания датчика №1;

X3 – показания датчика №2;

X4 – показания датчика №3;

X5 – признак недостоверности датчика №1 (достоверен – 0/ недостоверен -1);

X6 – признак недостоверности датчика №2 (достоверен – 0/ недостоверен -1);

X7 – признак недостоверности датчика №3 (достоверен – 0/ недостоверен -1);

## ПРИМЕЧАНИЕ.

Признаки недостоверности датчиков формируются в программе пользователя. При этом обычно учитывается:

- признак снятия переменной с опроса. Если переменная снята с опроса, она считается недостоверной;
- атрибут «тип замены недостоверного значения». Если значение данного атрибута не равно 0, переменная считается недостоверной;
- признак недостоверности самой переменной (входной сигнал вне диапазона измерения или отказ канала измерения).

X8 – признак выбора датчика №1 (не выбран – 0/ выбран - 1);

X9 – признак выбора датчика №2 (не выбран – 0/ выбран - 1);

X10 – признак выбора датчика №3 (не выбран – 0/ выбран - 1);

X11 – признак выбора автоматического режима расчета результирующего значения (ручной режим – 0/ автоматический режим -1);

X12 – код состояния признаков выбора датчиков, признака автоматического расчета. Если 2<sup>0й</sup> бит OPS (т.е. параметр X1) равен 0, т.е. кнопки выбора привязаны к аналоговым параметрам, то в зависимости от комбинации кнопок выбора датчиков и автоматического режима расчета в X12 должно записываться следующее значение:

Состояние кнопки автоматического режима	Состояние кнопки выбора датчика №3	Состояние кнопки выбора датчика №2	Состояние кнопки выбора датчика №1	Значение параметра X12 двоичное	Значение параметра X12 десятичное
0	0	0	0	0000	0

Состояние кнопки автоматического режима	Состояние кнопки выбора датчика №3	Состояние кнопки выбора датчика №2	Состояние кнопки выбора датчика №1	Значение параметра X12 двоичное	Значение параметра X12 десятичное
0	0	0	1	0001	1
0	0	1	0	0010	2
0	1	0	0	0100	4
1	0	0	0	1000	8

X13 – уставка рассогласования;

Y1 – ошибка расчетного параметра. Выставляется в 1, когда два или более датчиков недостоверны;

Y2 – ошибка рассогласования. Выставляется в 1, когда как минимум 2 датчика достоверны и рассогласование между ними больше уставки рассогласования;

Y3 – ошибка рассогласования между датчиком №1 и средним значением. Выставляется в 1, если рассогласование между датчиком №1 и средним значением больше уставки рассогласования и все датчики достоверны. Если Y2=1 и 0<sup>ой</sup> бит OPS выставлен в 1, то сбрасывается в 0;

Y4 – ошибка рассогласования между датчиком №2 и средним значением. Выставляется в 1, если рассогласование между датчиком №2 и средним значением больше уставки рассогласования и все датчики достоверны. Если Y2=1 и 0<sup>ой</sup> бит OPS выставлен в 1, то сбрасывается в 0;

Y5 – ошибка рассогласования между датчиком №3 и средним значением. Выставляется в 1, если рассогласование между датчиком №3 и средним значением больше уставки рассогласования и все датчики достоверны. Если Y2=1 и 0<sup>ой</sup> бит OPS выставлен в 1, то сбрасывается в 0;

Y6 – блокировка выбора датчика №1 (нет блокировки – 0/ есть блокировка - 1);

Y7 – блокировка выбора датчика №2 (нет блокировки – 0/ есть блокировка - 1);

Y8 – блокировка выбора датчика №3 (нет блокировки – 0/ есть блокировка - 1);

Y9 – признак выбора датчика №1 (не выбран – 0/ выбран - 1);

Y10 – признак выбора датчика №2 (не выбран – 0/ выбран - 1);

Y11 – признак выбора датчика №3 (не выбран – 0/ выбран - 1);

Y12 – признак выбора автоматического расчета результирующего значения (ручной режим – 0/ автоматический режим -1);

Y13 – код состояния признаков выбора датчиков, признака автоматического расчета. В зависимости от Y13 признаки выбора имеют следующие значения:

Значение параметра Y13 двоичное	Значение параметра Y13 десятичное	Состояние кнопки автоматического режима	Состояние кнопки выбора датчика №3	Состояние кнопки выбора датчика №2	Состояние кнопки выбора датчика №1
0000	0	0	0	0	0
0001	1	0	0	0	1
0010	2	0	0	1	0
0100	4	0	1	0	0
1000	8	1	0	0	0

Y14 – результирующее значение;

Y15 – код сообщения, формируемый функцией при возникновении ошибочной ситуации.

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Код сообщения	Ситуация
1	Настрочный параметр неверен
2	Отрицательная уставка рассогласования

### ПРИМЕЧАНИЕ.

Данная функция не выдает сообщений в роллинг. Но выходной параметр Y15 можно анализировать в программе пользователя после вызова данной функции, и если его значение равно 1, то выдавать сообщение «Настрочный параметр неверен», а если 2- то «Отрицательная уставка рассогласования».

Функция может быть настроена для двух режимов работы:

- С троированными датчиками
- С дублированными датчиками

Возможны два режима расчета результирующего значения:

- Автоматический расчет результирующего значения параметра;
- Ручной выбор конкретного значения параметра.

При настройке функции для работы с троированными датчиками она работает следующим образом:

1. Автоматический расчет результирующего значения параметра:

1.1. при достоверных значениях поступающих сигналов определяются минимальное, среднее и максимальное значение параметра:  $D_{min}$ ,  $D_{среднее}$ ,  $D_{max}$ .

1.1.1. определяется разница между  $D_{max}$ - $D_{среднее}$  и  $D_{среднее}$ - $D_{min}$ . Полученные разницы сравниваются с предельно допустимыми значениями рассогласования между датчиками.

1.1.2. при значениях, не превышающих допустимое рассогласование, результирующее значение вычисляется как среднее арифметическое значений 3-х параметров, если 5<sup>ый</sup> бит OPS равен 1 или как  $D_{среднее}$ , если 5<sup>ый</sup> бит OPS равен 0.

1.1.3. при выходе одного параметра за допустимые границы рассогласования данный параметр исключается из расчета, формируется признак «Ошибка рассогласования датчика» и результирующее значение вычисляется как среднее арифметическое двух оставшихся значений.

1.2. при выходе двух параметров за допустимые границы рассогласования формируется признак «Ошибка рассогласования». . Если в OPS 0<sup>ой</sup> бит равен 1, то признаки "Ошибка рассогласования датчика" по каждому конкретному датчику не формируются, если же равен 0, то формируются. Если 7<sup>ой</sup> бит OPS равен 0, то результирующее значение равно значению  $D_{среднее}$ . Если же этот бит равен 1, то в результирующее значение записывается признак недостоверности (значение вне диапазоне измерения).

1.3. при недостоверности одного из параметров результирующее значение вычисляется как среднее арифметическое двух оставшихся значений, если 5ый бит OPS равен 1, либо если 5ый бит OPS равен 0 и 8ой бит OPS равен 1. Если 5ый и 8ой биты OPS равны 0 результирующее значение будет вычисляться в зависимости от состояния бита 6 OPS.

Если 6ой бит OPS равен 1, то:

- при недостоверности датчика №1 результирующее значение равно значению датчика №2

- при недостоверности датчика №2 результирующее значение равно значению датчика №1
- при недостоверности датчика №3 результирующее значение равно значению датчика №1.

Если 6-й бит OPS равен 0, то:

- при недостоверности датчика №1 результирующее значение равно значению датчика №3;
- при недостоверности датчика №2 результирующее значение равно значению датчика №3;
- при недостоверности датчика №3 результирующее значение равно значению датчика №2.

- 1.4. при недостоверности двух параметров выставляется признак «Ошибка расчетного параметра». Если 7-й бит OPS равен 0, то результирующее значение равно значению оставшегося достоверного параметра. Если же этот бит равен 1, то в результирующее значение записывается признак недостоверности (значение вне диапазоне измерения).
- 1.5. при недостоверности трех параметров выставляется признак «Ошибка расчетного параметра». Если 7-й бит OPS равен 0, то результирующее значение вычисляется как среднее арифметическое значений трех датчиков. Если же этот бит равен 1, то в результирующее значение записывается признак недостоверности (значение вне диапазоне измерения).
- 1.6. В зависимости от описания признаков настройки в автоматическом режиме разрешен либо ручной выбор датчика при его достоверности, либо ручной выбор любого датчика не зависимо от его состояния.

## 2. Ручной режим выбора датчика :

- 2.1. Результирующим значением при выборе конкретного датчика являются показания этого датчика.
- 2.2. Если 3-й бит OPS равен 0, то нельзя выбрать недостоверный датчик, если равен 1, то можно.
- 2.3. Может быть выбран только один датчик.
- 2.4. Если 1-й бит OPS равен 1, то если выбранный датчик вдруг становится недостоверным, происходит переход в автоматический режим расчета результирующего параметра, даже если 3-й бит равен 1. Иначе переход в автоматический режим не происходит.
- 2.5. В ручном режиме ошибки по рассогласованию для всех датчиков не формируются.
- 2.6. В ручном режиме при недостоверности выбранного параметра формируется ошибка по недостоверности результирующего параметра.
- 2.7. В ручном режиме при достоверности выбранного параметра ошибки по недостоверности результирующего параметра не формируются.

При настройке функции для работы с дублированными датчиками она работает следующим образом:

1. Автоматический расчет результирующего значения параметра
  - 1.1. При достоверных значениях обоих параметров:
    - 1.1.1. Определяется разница между показаниями датчиков. Полученный результат сравнивается с предельно допустимым значениям рассогласования.
    - 1.1.2. При превышении разницей предельно допустимого значения рассогласования формируется признак «Ошибка рассогласования» .

- 1.1.3. Если 7<sup>ой</sup> бит OPS равен 0, то результирующее значение вычисляется как среднее арифметическое значений двух датчиков. Если же этот бит равен 1, то в результирующее значение записывается признак недоверности (значение вне диапазоне измерения).
- 1.2. При недоверности одного из параметров формируется признак «Ошибка расчетного параметра». , . Если 7<sup>ой</sup> бит OPS равен 0, то результирующее значение равно значению оставшегося достоверного датчика. Если же этот бит равен 1, то в результирующее значение записывается признак недоверности (значение вне диапазоне измерения).
- 1.3. При недоверности обоих параметров формируется признак «Ошибка расчетного параметра» , . Если 7<sup>ой</sup> бит OPS равен 0, то результирующее значение вычисляется как среднее арифметическое значений двух датчиков. Если же этот бит равен 1, то в результирующее значение записывается признак недоверности (значение вне диапазоне измерения).
- 1.4. В зависимости от описания признаков настройки в автоматическом режиме разрешен либо ручной выбор датчика при его достоверности, либо ручной выбор любого датчика не зависимо от его состояния
2. Ручной режим выбора датчика
  - 2.1. Результирующим значением при выборе конкретного датчика являются показания этого датчика.
  - 2.2. Если 3<sup>й</sup> бит OPS равен 0, то нельзя выбрать недоверный датчик, если равен 1, то можно.
  - 2.3. Может быть выбран только один датчик.
  - 2.4. Если 1<sup>ый</sup> бит OPS равен 1, то если выбранный датчик вдруг становится недоверным, происходит переход в автоматический режим расчета результирующего параметра, даже если 3<sup>й</sup> бит равен 1. Иначе переход в автоматический режим не происходит.
  - 2.5. В ручном режиме ошибки по рассогласованию для всех датчиков не формируются.
  - 2.6. В ручном режиме при недоверности выбранного параметра формируется ошибка по недоверности результирующего параметра.
  - 2.7. В ручном режиме при достоверности выбранного параметра ошибки по недоверности результирующего параметра не формируются.

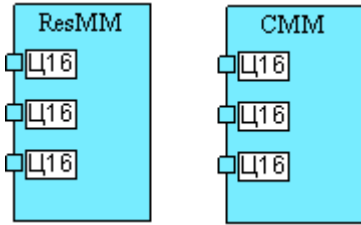


Пример вызова функции	
СТ:	ФБД:
<p>(пл8, пл9, пл10, пл11, пл12, пл13, пл14, пл15, пл16, пл17, пл18, пл19, пв6, пв7, пц2) = ProcD(пц1, пв1, пв2, пв3, пл1, пл2, пл3, пл4, пл5, пл6, пл7, пв4, пл5)</p>	
<p>(пл8, пл9, пл10, пл11, пл12, пл13, пл14, пл15, пл16, пл17, пл18, пл19, пв6, пв7, пц2) = ОбрД(пц1, пв1, пв2, пв3, пл1, пл2, пл3, пл4, пл5, пл6, пл7, пв4, пл5)</p>	

## 14.25 CMM, ResMM

### Назначение

Сброс мезонин-модуля.

Отображение	
СТ:	ФБД:
ResMM(X1,X2,X3)  CMM(X1,X2,X3)	
Входные параметры: X1(ц16), X2(ц16), X3(ц16)	

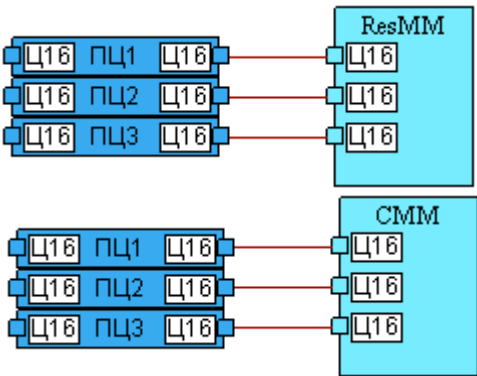
### Описание

- X1 - номер\_канала\_ЭИУСО – номер канала, к которому привязан модуль номер\_входа\_на\_плате с мезонином номер\_входа\_на\_плате;
- X2 - номер\_платы – данный параметр указывает на номер модуля контроллера TREI-5B-02, мезонин которого необходимо сбросить. Данный параметр эквивалентен значению поля «номер платы» базы данных системы КРУГ- 2000;
- X3 - номер\_входа\_на\_плате - данный параметр указывает на номер мезонина на модуле контроллера TREI-5B-02, который необходимо сбросить. Данный параметр эквивалентен значению поля «номер входа/выхода» базы данных системы КРУГ-2000.

Данная функция применяется только для платформ СРБК TREI-5B-02 любой версии.

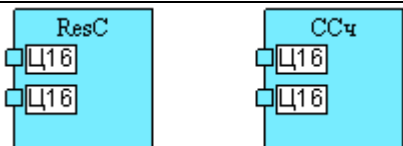
### Логика работы функции

Функция CMM предназначена для сброса (или переинициализации) какого-либо мезонина на модуле контроллера TREI-5B-02. Необходимость вызова данной функции возникает в том случае, когда требуется сбросить накопленное значение счетчика, либо когда необходимо запустить мезонин в работу после его сбоя.

Пример вызова функции	
СТ:	ФБД:
ResMM(пц1, пц2, пц3)  CMM(пц1, пц2, пц3)	

**14.26      ССч, ResC****Назначение**

Функция сброса значения канала счетчика.

Отображение	
СТ:	ФБД:
ResC(X1, X2) ССч(X1, X2)	
Входные параметры: X1(ц16), X2(ц16)	

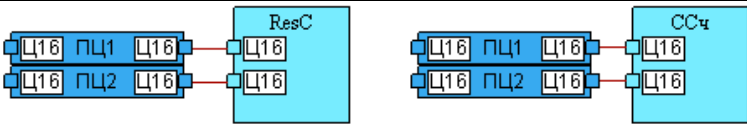
**Описание**

Функция применяется только для платформ СРВК TREI-5B-02 любой версии, СРВК МФК3000 любой версии, СРВК МФК любой версии.

**Логика работы функции**

X1 - номер платы в контроллере.

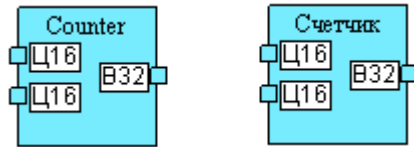
X2 - номер места (с 1 по 16) на плате, где установлен канал счетчика, который нужно сбросить.

Пример вызова функции	
СТ:	ФБД:
ResC(пц1, пц2) ССч(пц1, пц2)	

14.27 Счетчик, Counter

Назначение

Непосредственное чтение значения узла ICNT.

Отображение	
СТ:	ФБД:
$Y = \text{Counter}(X1, X2)$ $Y = \text{Счетчик}(X1, X2)$	
Входные параметры: X1(ц16), X2(ц16) Выходные параметры: Y(в32)	

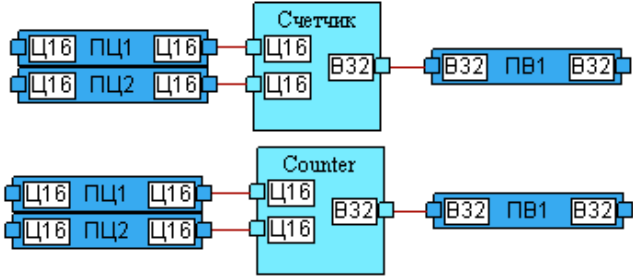
Описание

Функция применяется только для платформ СРВК TREI-5В-02 любой версии , «СРВК МФК3000 любой версии, СРВК МФК любой версии.

X1 – целое число, указывающее на номер платы, где находится узел ICNT.

X2 – целое число, указывающее номер места на плате, где установлен узел ICNT.

Y – результат считывания значения узла ICNT.

Пример вызова функции	
СТ:	ФБД:
$пв1 = \text{Счетчик}(пц1, пц2)$ $пв1 = \text{Counter}(пц1, пц2)$	

## 14.28 ТПУ2, TPU2

**Назначение**

Программа КРУГОЛ использует функцию ТПУ2 для старта опроса узла импульсного ввода для проверки расходомеров (узел ITPU), и далее получает результаты измерений через переменные БД, назначенные через аргументы функции.

**Отображение**

Отображение	
СТ:	ФБД:
<p>TPU2(X1, X2, X3, X4, X5, X6, X7, X8, X9)</p> <p>ТПУ2(X1, X2, X3, X4, X5, X6, X7, X8, X9)</p>	<div><div><p>TPU2</p><div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div></div></div><div><p>ТПУ2</p><div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div><div>Ц16</div></div></div></div>
<p>Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16), X5(ц16), X6(ц16), X7(ц16), X8(ц16), X9(ц16)</p>	

**Описание**

X1 - МП – номер платы, где установлены мезонины (узлы) ввода сигналов от ТПУ.

X2 - ММ – номер входа платы, где установлен мезонин ввода сигналов от ТПУ.

X3 - ВХд1 – номер входа платы подключения стартового детектора первой пары по направлению движения шара.

X4 - ВХд3 – номер входа платы подключения детектора финиша первой пары по направлению движения шара.

X5 - ВХд2 – номер входа платы подключения стартового детектора второй пары по направлению движения шара.

X6 - ВХд4 – номер входа платы подключения детектора финиша второй пары по направлению движения шара.

X7 - иРВ – начальный номер переменной ручного ввода для буфера (4 переменных) вывода полученных результатов измерений.

X8 - иВД – начальный номер входной дискретной переменной для буфера (4 переменных) вывода последовательности срабатывания детекторов ТПУ.

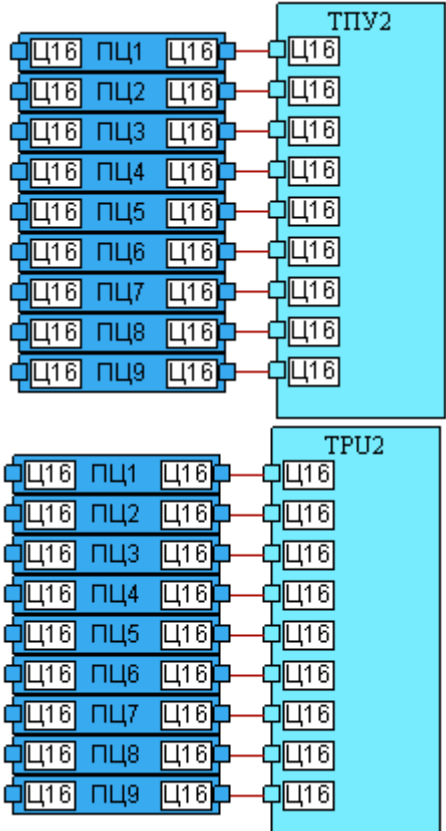
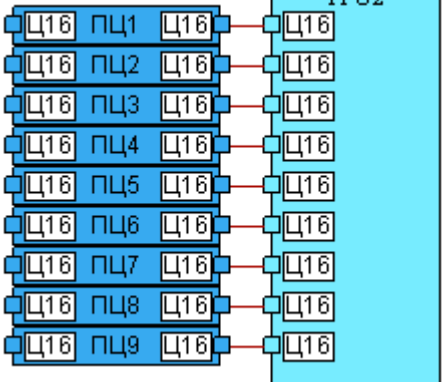
X9 - НП – направление последовательности записи последовательности срабатывания детекторов, при 0 с первой переменной буфера, при 1 с последней переменной буфера.

При 0 значениях ВХд1,3 или ВХд2,4 пара детекторов исключается из работы.

Данная функция применяется только для платформ CPBK TREI-5B-02 любой версии.

Логика работы функции

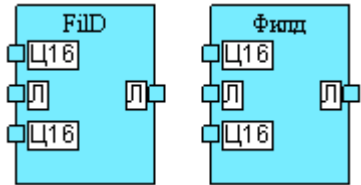
Программа КРУГОЛ использует функцию ТПУ2 для старта опроса узла импульсного ввода для поверки расходомеров (узел ITRU), и далее получает результаты измерений через переменные БД, назначенные через аргументы функции.

Пример вызова функции	
СТ:	ФБД:
ТПУ2(пц1, пц2, пц3, пц4, пц5, пц6, пц7, пц8, пц9)	
TRU2(пц1, пц2, пц3, пц4, пц5, пц6, пц7, пц8, пц9)	

## 14.29 Филд, FiID

**Назначение**

Фильтр входной дискретной переменной.

Отображение	
СТ:	ФБД:
$Y = FiID(X1, X2, X3)$ $Y = \text{Филд}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(л), X3(ц16) Выходные параметры: Y(л)	

**Описание**

Применяется только для платформ СРВК любой версии.

**Логика работы функции**

X1 - порядковый номер блока (от 1 до 20000). Для платформ, начиная с СРВК версии 7.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 - входное значение.

X3 - число подтверждений "1" при проходах.

С выхода Y снимается результат коммутации логического формата.

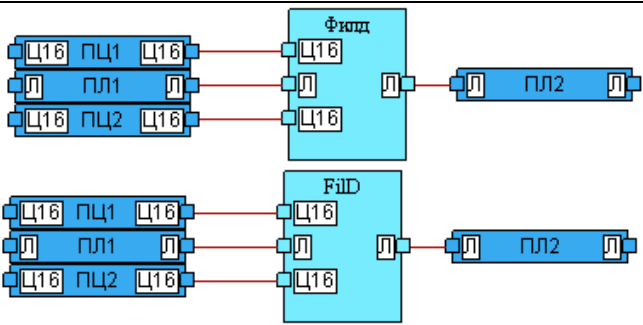
Если параметр X2 равен "1" в течение X3 циклов, то выход становится равным "1". Если параметр X2 равен "0" в течение X3 циклов, то выход становится равным "0". В остальных случаях выход не изменяется от цикла к циклу.

Если X2 = 1 то  $i = i + 1$ , и если  $i > X3$  то Y = 1.

Если X2 = 0 то  $i = i + 1$ , и если  $i > X3$  то Y = 0.

Иначе Y=Yпредыдущее.

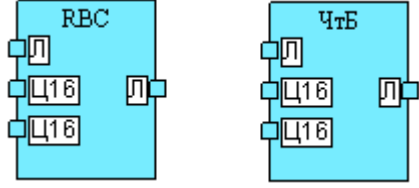
При изменении значения X2 внутренний счетчик i сбрасывается в нуль.

Пример вызова функции	
СТ:	ФБД:
$пл2 = \text{Филд}(пц1, пл1, пц2)$  $пл2 = FiID(пц1, пл1, пц2)$	

14.30 ЧтБ, RBC

Назначение

Считать значение из указанного бита переменной 16-битного формата.

Отображение	
СТ:	ФБД:
$Y1 = RBC(X1, X2, X3)$ $Y1 = ЧтБ(X1, X2, X3)$	
Входные параметры: X1(л), X2(ц16), X3(ц16) Выходные параметры: Y1(л)	

Описание

Применяется для платформ СРВК версии ниже 8.0, и среды исполнения КРУГОЛ версии ниже 2.2.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

Логика работы функции

Функция считывает значение из указанного бита переменной целого формата.

X1 - значение переменной до работы функции. Это значение присваивается выходу Y1 при некорректном значении входа X3.

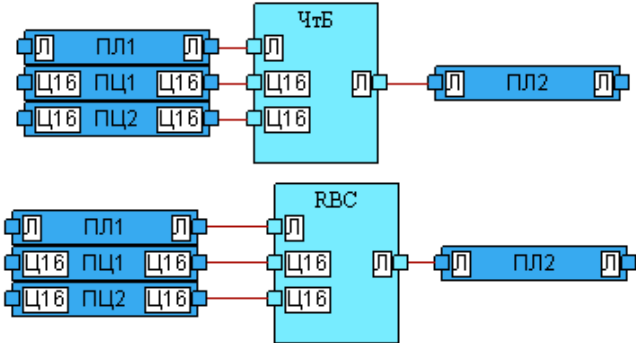
X2 - переменная целого 16-битного формата, из которой требуется считать значение указанного бита. Номер бита определяет следующий аргумент функции.

X3 - номер бита, определяет, из какого бита второго аргумента необходимо считать значение. Значение аргумента X3 должно лежать в диапазоне от 1 до 16. Здесь 1 - обращение к младшему биту, 16 - к старшему.

Y1 - значение заданного бита.

Обработка ошибок

Если значение, переданное на вход X3, меньше 1 или больше 16, то на выход Y1 подается значение с входа X1 и в роллинг выводится сообщение «КРУГОЛ:RBC Ош.значен.аргум.X3».


Пример вызова функции	
СТ:	ФБД:
$пл2 = ЧтБ(пл1, пц1, пц2)$ $пл2 = RBC(пл1, пц1, пц2)$	



## 14.31 ЗапБ, WBC

**Назначение**

Записать логическое значение в указанный бит переменной целого 16-битного формата.

Отображение	
СТ:	ФБД:
$Y1 = \text{ЗапБ}(X1, X2, X3)$ $Y1 = \text{WBC}(X1, X2, X3)$	
Входные параметры: X1(ц16), X2(ц16), X3(л) Выходные параметры: Y(ц16)	

**Описание**

Применяется для платформ СРВК версии ниже 8.0, и среды исполнения КРУГОЛ версии ниже 2.2.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

X1 - переменная целого формата, в которую требуется записать данные в указанный бит. Номер бита определяет следующий аргумент функции.

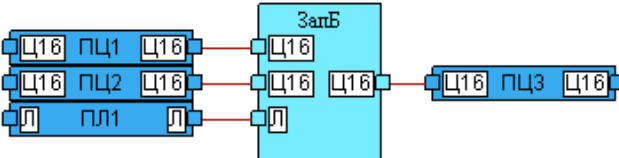
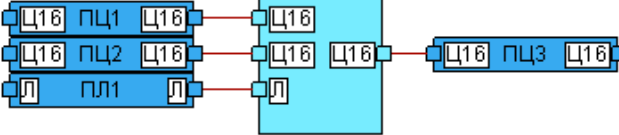
X2 - номер бита, определяет, в какой бит первого аргумента необходимо записать значение. Значение аргумента X2 должно лежать в диапазоне от 1 до 16. Здесь 1 - обращение к младшему биту, 16 - к старшему.

X3 - значение, записываемое в указанный бит первого аргумента.

Y1 - измененное значение первого аргумента функции.

**Обработка ошибок**

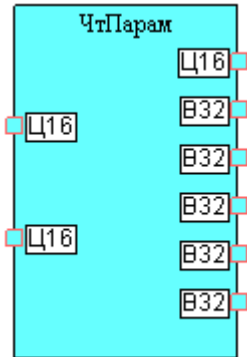
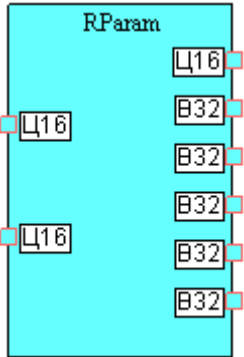
Если значение, переданное на вход X2, меньше 1 или больше 16, то в роллинг выводится сообщение «КРУГОЛ:WBC Ош.значен.аргум.X2».

Пример вызова функции	
СТ:	ФБД:
$\text{пц3} = \text{ЗапБ}(\text{пц1}, \text{пц2}, \text{пл1})$	
$\text{пц3} = \text{WBC}(\text{пц1}, \text{пц2}, \text{пл1})$	

### 14.32 ЧтПарам, RParam

#### Назначение

Функция чтения параметров переменной базы данных (текущее значение, метка времени, качество)

Отображение	
СТ:	ФБД:
<p>(Y1, Y2, Y3, Y4, Y5, Y6)=ЧтПарам(X1, X2)</p> <p>(Y1, Y2, Y3, Y4, Y5, Y6)=RПарам(X1, X2)</p>	<div><div><p>ЧтПарам</p></div><div><p>RParam</p></div></div>
<p>Входные параметры: X1 (Ц16)</p> <p>Выходные параметры: Y1 (Ц16), Y2 (В32), Y3 (В32), Y4 (В32), Y5 (В32), Y6 (В32)</p>	

#### Входные параметры:

X1 (Ц16) - Тип переменной базы данных. Является обязательным входом, принимающим следующие целые значения:

- 1 – входная аналоговая,
- 2 – входная дискретная,
- 3 – ручной ввод,
- 6 – дискретная выходная,
- 8 – аналоговая выходная.

X2 (Ц16) - Номер переменной базы данных типа, указанного на входе X1. Является обязательным входом, принимающим целые значения в диапазоне от 1 до 10000

#### Выходные параметры:

Y1 (Ц16) - Код достоверности (ошибки) входных параметров. Является необязательным выходом, принимающим целые значения в диапазоне от 0 до 1:

- 0 - Если во входных данных ошибок нет
- 1 - Если в базе данных не существует переменной с заданным типом и номером.

Y2 (В32) - Значение текущего значения переменной (используется атрибут «текущее значение» в СРВК для соответствующего типа переменной). Является обязательным выходом, принимающим вещественные значения

Y3 (В32) – Дата из метки времени переменной в формате ДДММГГ. Является обязательным выходом, принимающим вещественные значения в диапазоне от 010101 до 311299

Y4 (В32) – Время из метки времени переменной в формате ЧЧММСС. Является обязательным выходом, принимающим вещественные значения в диапазоне от 0 до 235959

Y5 (В32) – Значение миллисекунд из метки времени переменной. Является необязательным выходом, принимающим вещественные значения в диапазоне от 0 до 999

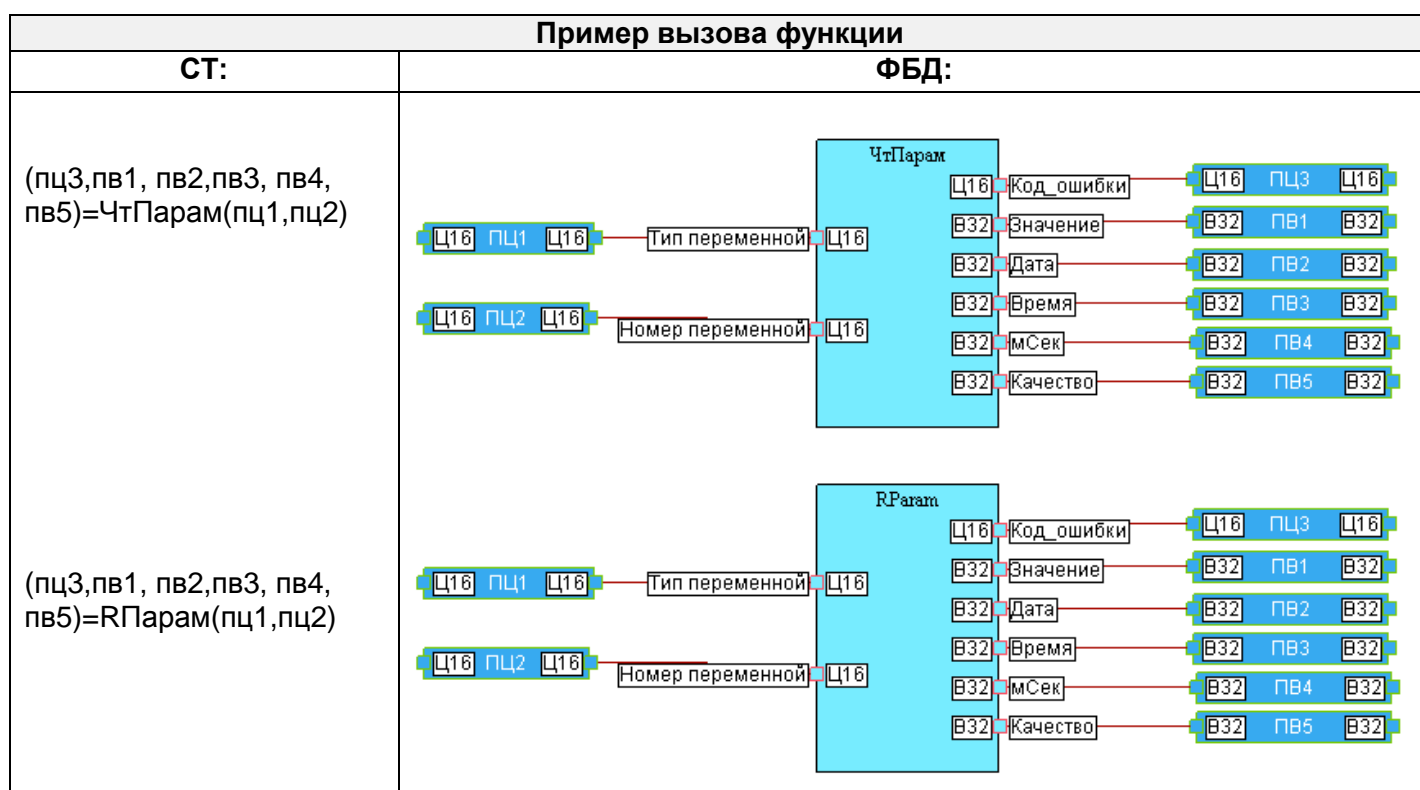
Y6 (В32) - Качество значения из атрибута качества переменной базы данных. Является необязательным выходом, принимающим вещественные значения в диапазоне от 0 до

255. Основными значениями качества являются: «хорошее» - 192, «плохое» - 0, «неопределенное» - 64.

### Логика работы функции

При обращении к заданной переменной базы данных по типу и номеру, функция возвращает:

- текущее значение заданной переменной,
- дату из метки времени заданной переменной в формате ДДММГГ (день-месяц-год),
- время из метки времени заданной переменной в формате ЧЧММСС (час-минута-секунда),
- целое значение миллисекунд из метки времени заданной переменной,
- качество в виде младшего байта параметра качества заданной переменной.



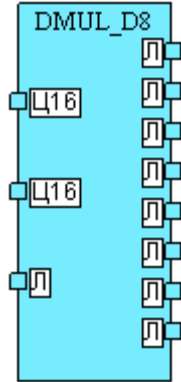


## 15 ФУНКЦИИ КОММУТАЦИИ ДАННЫХ

### 15.1 DMUL\_D8

#### Назначение

Демультимплексор на 8 дискретных выходов.

Отображение	
СТ:	ФБД:
(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = DMUL_D8(X1, X2, X3)	
<p>Входные параметры: X1 (ц16), X2 (ц16), X3 (л)</p> <p>Выходные параметры: Y1 (л), Y2 (л), Y3 (л), Y4 (л), Y5 (л), Y6 (л), Y7 (л), Y8 (л)</p>	

#### Описание

**DMUL\_D8** – функция демультимплексора на 8 дискретных входах производит подключение входного дискретного сигнала к одному из восьми выходных дискретных сигналов.

Применяется для платформ СРВК версий 7.0 и 7.1, и среды исполнения КРУГОЛ версий 2.0 и 2.1.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

#### Логика работы функции

X1 – номер группы входных сигналов.

X2 – селектор входных сигналов.

X3 – входной сигнал.

Y1 – первый выходной сигнал.

Y2 – второй выходной сигнал.

Y3 – третий выходной сигнал.

Y4 – четвертый выходной сигнал.

Y5 – пятый выходной сигнал.

Y6 – шестой выходной сигнал.

Y7 – седьмой выходной сигнал.

Y8 – восьмой выходной сигнал.

ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

В зависимости от значения селектора X2 одному из выходов функции Y1 ... Y8 присваивается входной сигналов X3, если эти выходные сигналы соответствуют заданному значению номера группы X1. Если ни один из выходных сигналов не соответствует заданному значению номера группы X1, всем выходам функции Y1 ... Y8 присваивается значение FALSE.

Номер выходного сигнала N из значений селектора и номера группы вычисляется по следующей формуле:

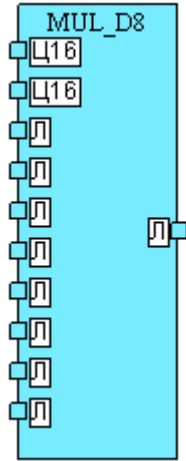
Если  $(X1 = (X2-1)/8)$ ,  
то  $N = (X2-1) \& 7 + 1$ ;  
иначе Y1 ... Y8 = FALSE.

Пример вызова функции	
СТ:	ФБД:
(пл2, пл3, пл4, пл5, пл6, пл7, пл8, пл9) = DMUL_D8(пц1, пц2, пл1)	

## 15.2 MUL\_D8

### Назначение

Мультиплексор на 8 дискретных входов.

Отображение	
СТ:	ФБД:
$Y1 = \text{MUL\_D8}(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10)$	
Входные параметры: X1(ц16), X2(ц16), X3(л), X4(л), X5(л), X6(л), X7(л), X8(л), X9(л), X10(л) Выходные параметры: Y(л)	

### Описание

**MUL\_D8** – функция мультиплексора на 8 дискретных входов производит выбор одного из восьми входных дискретных сигналов.

Применяется для платформ СРВК версий 7.0 и 7.1, и среды исполнения КРУГОЛ версий 2.0 и 2.1.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

### Логика работы функции

X1 – номер группы входных сигналов.  
X2 – селектор входных сигналов.  
X3 – первый входной сигнал.  
X4 – второй входной сигнал.  
X5 – третий входной сигнал.  
X6 – четвертый входной сигнал.  
X7 – пятый входной сигнал.  
X8 – шестой входной сигнал.  
X9 – седьмой входной сигнал.  
X10 – восьмой входной сигнал.  
Y1 – выходной сигнал.

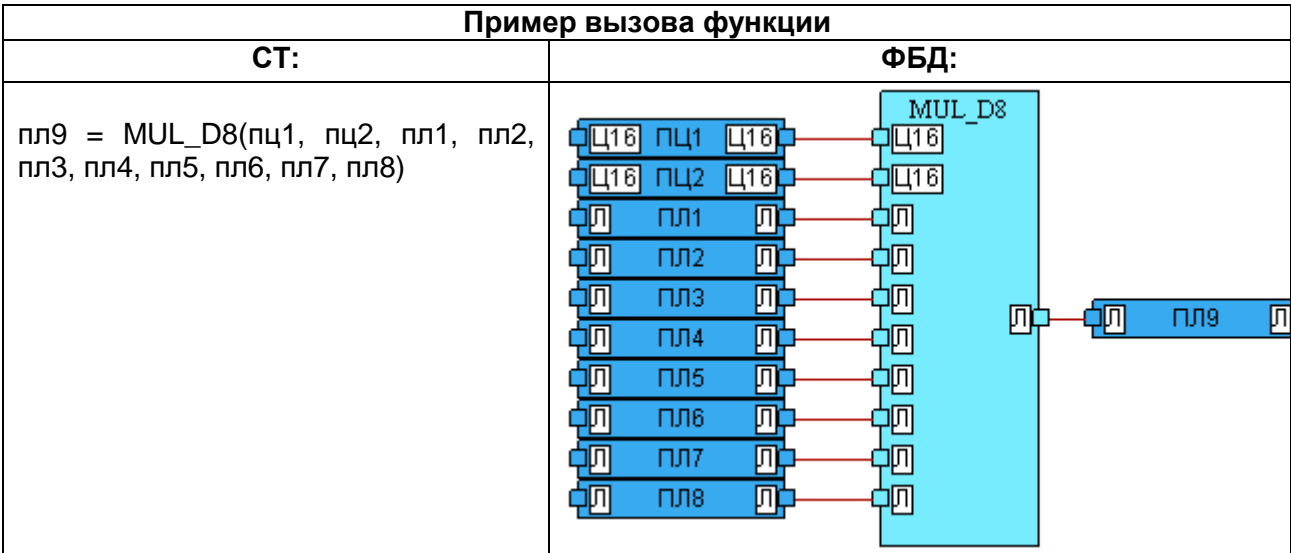
В зависимости от значения селектора X2 выходу функции Y1 присваиваются один из входных сигналов X3 - X10, если эти сигналы соответствуют заданному значению номера

ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

группы X1. Если ни один из входных сигналов не соответствует заданному значению номера группы X1, выходу функции Y1 присваивается значение FALSE.

Номер входного сигнала N из значений селектора и номера группы вычисляется по следующей формуле:

Если  $(X1 = (X2-1)/8)$ ,  
то  $N = (X2-1) \& 7 + 1$ ;  
иначе Y1 = FALSE.

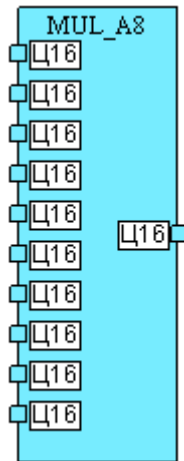




### 15.3 MUL\_A8

#### Назначение

Мультиплексор на 8 аналоговых входов.

Отображение	
СТ:	ФБД:
$Y1 = MUL\_A8(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10)$	
<p>Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16), X5(ц16), X6(ц16), X7(ц16), X8(ц16), X9(ц16), X10(ц16)          Выходные параметры: Y(ц16)</p>	

#### Описание

**MUL\_A8** – функция мультиплексора на 8 аналоговых входов производит выбор одного из восьми входных аналоговых сигналов.

Применяется для платформ СРВК версий 7.0 и 7.1, и среды исполнения КРУГОЛ версий 2.0 и 2.1.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

#### Логика работы функции

X1 – номер группы входных сигналов.  
 X2 – селектор входных сигналов.  
 X3 – первый входной сигнал.  
 X4 – второй входной сигнал.  
 X5 – третий входной сигнал.  
 X6 – четвертый входной сигнал.  
 X7 – пятый входной сигнал.  
 X8 – шестой входной сигнал.  
 X9 – седьмой входной сигнал.  
 X10 – восьмой входной сигнал.  
 Y1 – выходной сигнал.

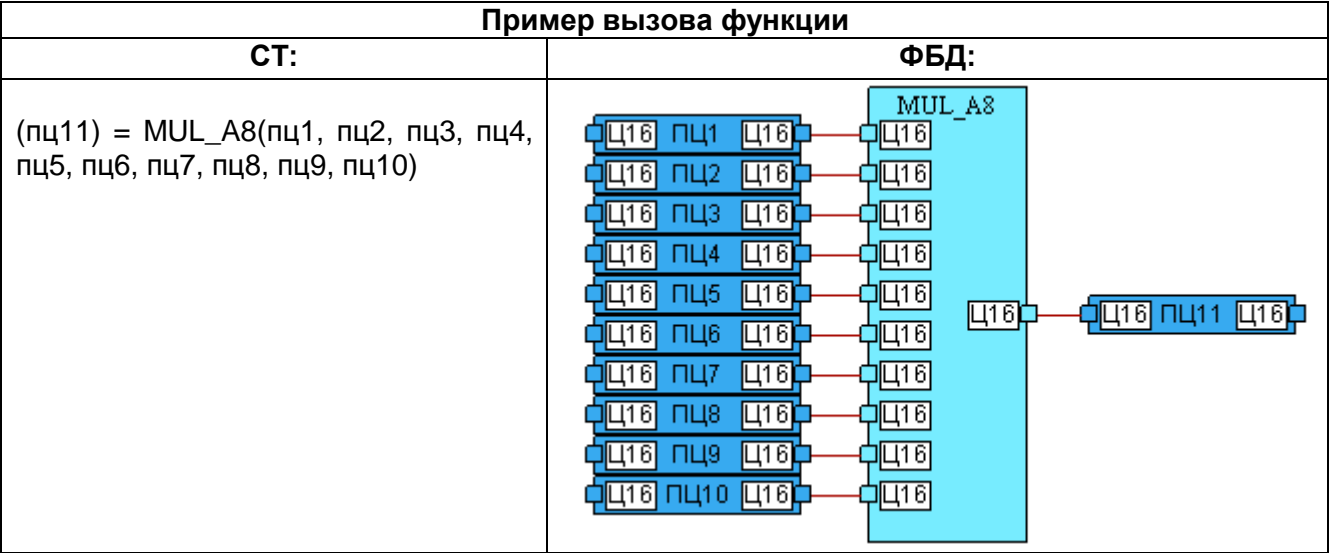
В зависимости от значения селектора X2 выходу функции Y1 присваиваются один из входных сигналов X3 - X10, если эти сигналы соответствуют заданному значению номера

ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

группы X1. Если ни один из входных сигналов не соответствует заданному значению номера группы X1, выходу функции Y1 присваивается значение 0.

Номер входного сигнала N из значений селектора и номера группы вычисляется по следующей формуле:

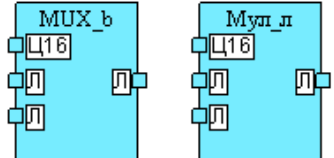
Если  $(X1 = (X2-1)/8)$ ,  
то  $N = (X2-1) \& 7 + 1$ ;  
иначе  $Y1 = FALSE$ .



## 15.4 Мул\_л, MUX\_b

### Назначение

Мультиплексор на неопределённое количество логических входов.

Отображение	
СТ:	ФБД:
$Y1 = \text{MUX\_b}(X1, X2, X3 [, X4, ..., Xn])$ $Y1 = \text{Мул\_л}(X1, X2, X3 [, X4, ..., Xn])$	
Входные параметры (по умолчанию - три): X1(ц16), X2(л), X3(л) Выходные параметры: Y1(л)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

Функция может иметь произвольное количество входных параметров логического формата: X2, X3 [, X4, ..., Xn] (по умолчанию -2). Использование дополнительных входов описано в разделе «Вставка элементов» книги «Модульная интегрированная SCADA КРУГ-2000. КРУГОЛ. Интегрированная среда разработки».

### Логика работы функции

**MUX\_b** – функция мультиплексора на произвольное количество входов производит выбор одного из входных логических сигналов.

X1 – селектор входных сигналов.

X2 – первый входной сигнал.

X3 – второй входной сигнал.

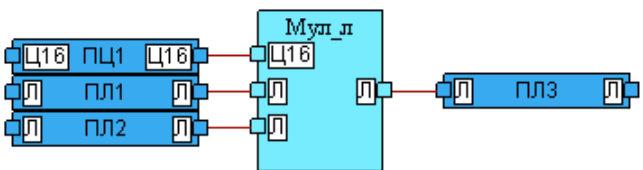
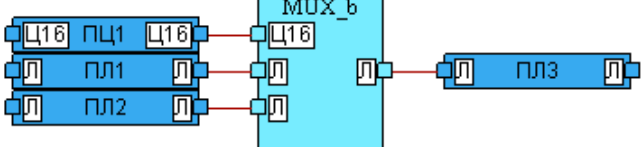
Xn – n-1-ый входной сигнал.

Y1 – выходной сигнал.

В зависимости от значения селектора X1 выходу функции Y1 присваиваются один из входных сигналов X2- Xn.

Если X1=1, то Y1=X2;

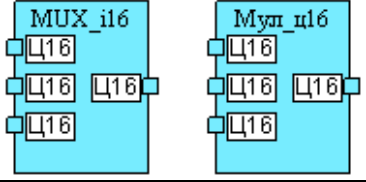
Если X1=n-1, то Y1=Xn.

Пример вызова функции	
СТ:	ФБД:
пл3 = Мул_л(плц1, пл1, пл2)	
пл3 = MUX_b(плц1, пл1, пл2)	

## 15.5 Мул\_ц16, MUX\_i16

### Назначение

Мультиплексор на неопределённое количество входов целого 16-битного формата.

Отображение	
СТ:	ФБД:
$Y1 = \text{MUX\_i16}(X1, X2, X3 [, X4, ..., Xn])$ $Y1 = \text{Мул\_ц16}(X1, X2, X3 [, X4, ..., Xn])$	
Входные параметры (по умолчанию - три): X1(ц16), X2(ц16), X3(ц16) Выходные параметры: Y1(ц16)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

Функция может иметь произвольное количество входных параметров целого формата: X1, X2, X3 [, X4, ..., Xn] (по умолчанию -3). Использование дополнительных входов описано в разделе «Вставка элементов» книги «Модульная интегрированная SCADA КРУГ-2000. КРУГОЛ. Интегрированная среда разработки».

### Логика работы функции

**MUX\_i16** – функция мультиплексора на произвольное количество входов производит выбор одного из входных сигналов целого типа.

X1 – селектор входных сигналов.

X2 – первый входной сигнал.

X3 – второй входной сигнал.

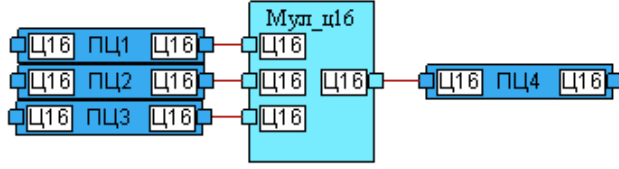
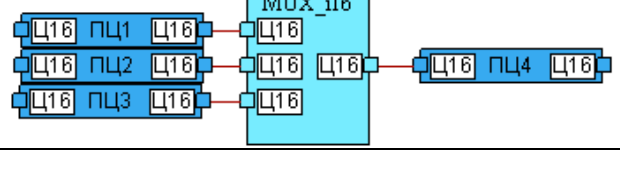
Xn – n-1ый входной сигнал.

Y1 – выходной сигнал.

В зависимости от значения селектора X1 выходу функции Y1 присваиваются один из входных сигналов X2 - Xn.

Если X1=1, то Y1=X2;

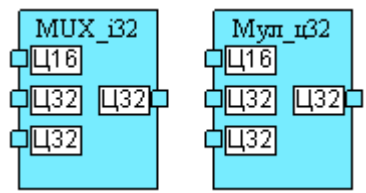
Если X1=n-1, то Y1=Xn.

Пример вызова функции	
СТ:	ФБД:
пц4 = Мул_ц16(пц1, пц2, пц3)	
пц4 = MUX_i16(пц1, пц2, пц3)	

## 15.6 Мул\_ц32, MUX\_i32

### Назначение

Мультиплексор на неопределённое количество входов целого 32-битного формата.

Отображение	
СТ:	ФБД:
$Y1 = \text{MUX\_i32}(X1, X2, X3 [, X4, \dots, Xn])$ $Y1 = \text{Мул\_ц32}(X1, X2, X3 [, X4, \dots, Xn])$	
Входные параметры (по умолчанию - три): X1(ц16), X2(ц32), X3(ц32) Выходные параметры: Y1(ц32)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

Функция может иметь произвольное количество входных параметров целого 32-битного формата: X2, X3 [, X4, ..., Xn] (по умолчанию -2). Использование дополнительных входов описано в 5.4.2.1 «Вставка элементов» книги «Модульная интегрированная SCADA КРУГ-2000. КРУГОЛ. Интегрированная среда разработки».

### Логика работы функции

**MUX\_i32** – функция мультиплексора на произвольное количество входов производит выбор одного из входных сигналов целого 32-битного типа.

X1 – селектор входных сигналов.

X2 – первый входной сигнал.

X3 – второй входной сигнал.

Xn – n-1ый входной сигнал.

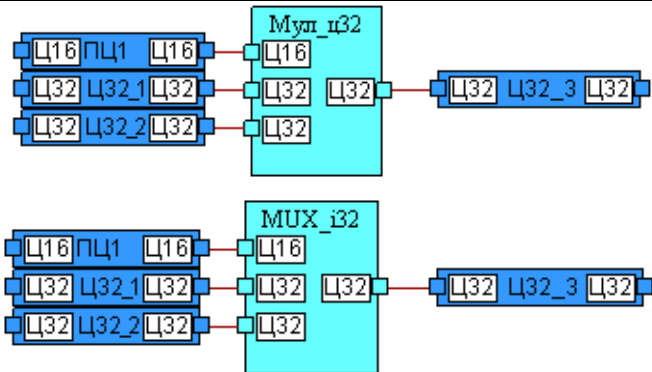
Y1 – выходной сигнал.

В зависимости от значения селектора X1 выходу функции Y1 присваиваются один из входных сигналов X2 - Xn.

Если X1=1, то Y1=X2;

...

Если X1=n-1, то Y1=Xn.

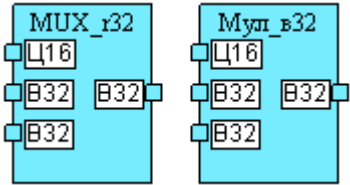
Пример вызова функции	
СТ:	ФБД:
$\text{ц32\_3} = \text{Мул\_ц32}(\text{пц1}, \text{ц32\_1}, \text{ц32\_2})$  $\text{ц32\_3} = \text{MUX\_i32}(\text{пц1}, \text{ц32\_1}, \text{ц32\_2})$	

Здесь: ц32\_1, ц32\_2, ц32\_3 - локальные переменные целого 32-битного типа.

## 15.7 Мул\_в32, MUX\_r32

### Назначение

Мультиплексор на неопределённое количество входов вещественного 32-битного формата.

Отображение	
СТ:	ФБД:
$Y1 = \text{MUX\_r32}(X1, X2, X3 [, X4, ..., Xn])$ $Y1 = \text{Мул\_в32}(X1, X2, X3 [, X4, ..., Xn])$	
Входные параметры (по умолчанию - три): X1(ц16), X2(в32), X3(в32) Выходные параметры: Y1(в32)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

Функция может иметь произвольное количество входных параметров вещественного 32-битного формата: X2, X3 [, X4, ..., Xn] (по умолчанию -2). Использование дополнительных входов описано в разделе 5.4.2.1 «Вставка элементов» книги «Модульная интегрированная SCADA КРУГ-2000. КРУГОЛ. Интегрированная среда разработки».

### Логика работы функции

**MUX\_r32** – функция мультиплексора на произвольное количество входов производит выбор одного из входных сигналов вещественного 32-битного типа.

X1 – селектор входных сигналов.

X2 – первый входной сигнал.

X3 – второй входной сигнал.

Xn – n-1ый входной сигнал.

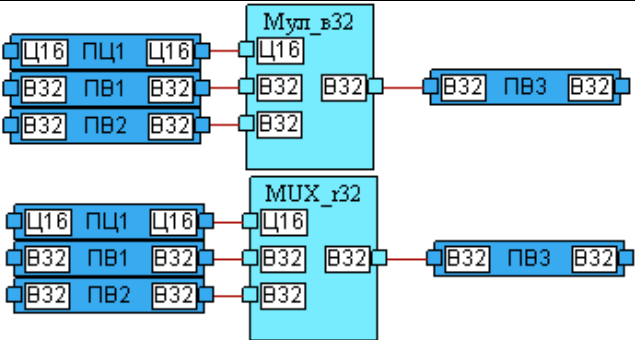
Y1 – выходной сигнал.

В зависимости от значения селектора X1 выходу функции Y1 присваиваются один из входных сигналов X2 - Xn.

Если X1=1, то Y1=X2;

...

Если X1=n-1, то Y1=Xn.

Пример вызова функции	
СТ:	ФБД:
$пв3 = \text{Мул\_в32}(пц1, пв1, пв2)$ $пв3 = \text{MUX\_r32}(пц1, пв1, пв2)$	

## 15.8 Мул\_в64, MUX\_r64

### Назначение

Мультиплексор на неопределённое количество входов вещественного 64-битного формата.

Отображение	
СТ:	ФБД:
$Y1 = \text{MUX\_r64}(X1, X2, X3 [, X4, ..., Xn])$ $Y1 = \text{Мул\_в64}(X1, X2, X3 [, X4, ..., Xn])$	
Входные параметры (по умолчанию - три): X1(ц16), X2(в64), X3(в64) Выходные параметры: Y1(в64)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

Функция может иметь произвольное количество входных параметров вещественного 64-битного формата: X2, X3 [, X4, ..., Xn] (по умолчанию -2). Использование дополнительных входов описано в разделе 5.4.2.1 «Вставка элементов» книги «Модульная интегрированная SCADA КРУГ-2000. КРУГОЛ. Интегрированная среда разработки».

### Логика работы функции

**MUX\_r64** – функция мультиплексора на произвольное количество входов производит выбор одного из входных сигналов вещественного 64-битного типа.

X1 – селектор входных сигналов.

X2 – первый входной сигнал.

X3 – второй входной сигнал.

Xn – n-1ый входной сигнал.

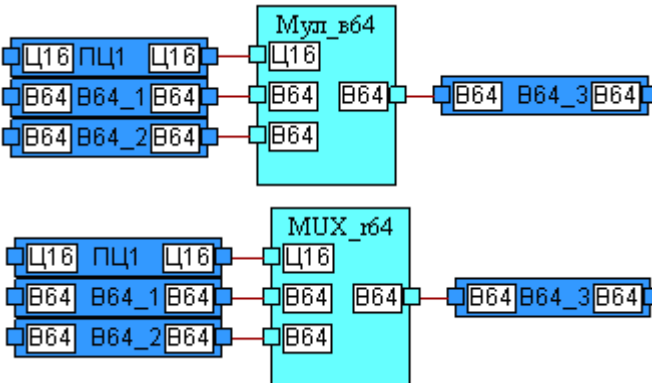
Y1 – выходной сигнал.

В зависимости от значения селектора X1 выходу функции Y1 присваиваются один из входных сигналов X2 - Xn.

Если X1=1, то Y1=X2;

...

Если X1=n-1, то Y1=Xn.

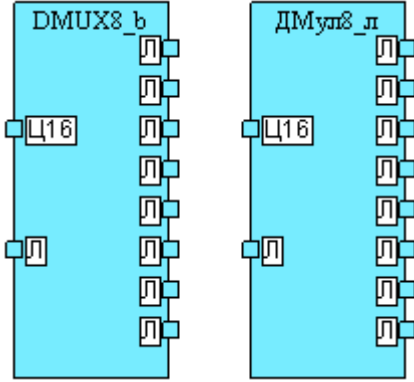
Пример вызова функции	
СТ:	ФБД:
$\text{в64\_3} = \text{Мул\_в64}(\text{пц1}, \text{в64\_1}, \text{в64\_2})$  $\text{в64\_3} = \text{MUX\_r64}(\text{пц1}, \text{в64\_1}, \text{в64\_2})$	

Здесь в64\_1, в64\_2, в64\_3 - локальные переменные вещественного 64 -битного типа.

## 15.9 ДМул8\_л, DMUX8\_b

### Назначение

Демультимплексор на 8 выходов логического формата.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = DMUX8\_b(X1, X2)$ $(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = ДМул8\_л(X1, X2)$	
Входные параметры: X1(ц16), X2(л) Выходные параметры: Y1(л), Y2(л), Y3(л), Y4(л), Y5(л), Y6(л), Y7(л), Y8(л)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

### Логика работы функции

**DMUX8\_B** – функция демультимплексора на 8 выходов производит подключение входного дискретного сигнала к одному из восьми выходных дискретных сигналов.

X1 – селектор выходных сигналов.

X2 – входной сигнал.

Y1 – первый выходной сигнал.

Y2 – второй выходной сигнал.

Y3 – третий выходной сигнал.

Y4 – четвертый выходной сигнал.

Y5 – пятый выходной сигнал.

Y6 – шестой выходной сигнал.

Y7 – седьмой выходной сигнал.

Y8 – восьмой выходной сигнал.

В зависимости от значения селектора X1 одному из выходов функции Y1 - Y8 присваивается значение входного сигнала X2. Всем остальным выходам функции присваивается значение "0".

Если X1=1, то Y1=X2;

...

Если X1=8, то Y8=X2.



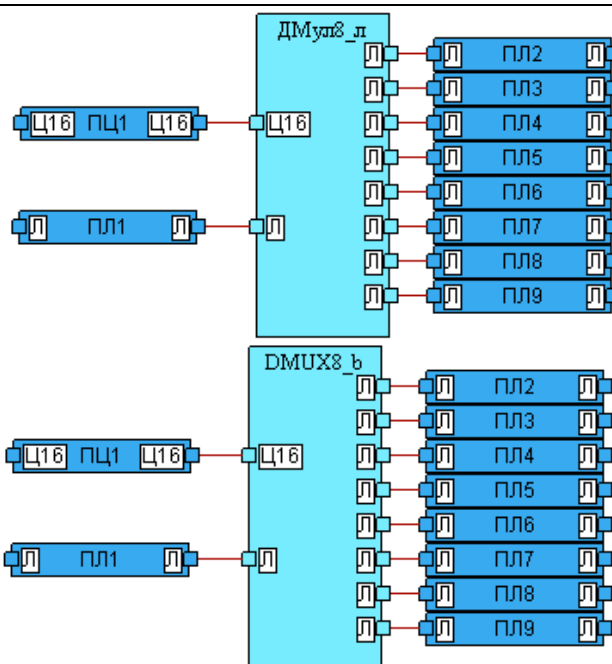
## Пример вызова функции

СТ:

(пл2, пл3, пл4, пл5, пл6, пл7, пл8, пл9) = ДМул8\_л(пц1, пл1)

(пл2, пл3, пл4, пл5, пл6, пл7, пл8, пл9) = DMUX8\_B(пц1, пл1)

ФБД:



15.10 ДМул8\_ц16, DMUX8\_i16

Назначение

Демультимплексор на 8 выходов целого 16-битного формата.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = DMUX8\_i16(X1, X2)$  $(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = ДМул8\_ц16(X1, X2)$	
Входные параметры: X1(ц16), X2(ц16) Выходные параметры: Y1(ц16), Y2(ц16), Y3(ц16), Y4(ц16), Y5(ц16), Y6(ц16), Y7(ц16), Y8(ц16)	

Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

Логика работы функции

**DMUX8\_i16** – функция демультимплексора на 8 выходов производит подключение входного целочисленного 16-битного сигнала к одному из восьми выходов.

X1 – селектор выходных сигналов.  
X2 – входной сигнал.

Y1 – первый выходной сигнал.  
Y2 – второй выходной сигнал.  
Y3 – третий выходной сигнал.  
Y4 – четвертый выходной сигнал.  
Y5 – пятый выходной сигнал.  
Y6 – шестой выходной сигнал.  
Y7 – седьмой выходной сигнал.  
Y8 – восьмой выходной сигнал.

В зависимости от значения селектора X1 одному из выходов функции Y1 - Y8 присваивается значение входного сигнала X2. Всем остальным выходам функции присваивается значение "0".

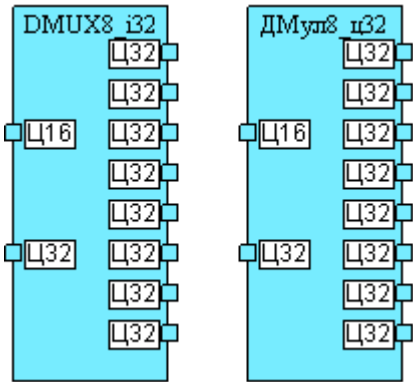
Если X1=1, то Y1=X2;  
...  
Если X1=8, то Y8=X2.

Пример вызова функции	
СТ:	ФБД:
<p>(пц3, пц4, пц5, пц6, пц7, пц8, пц9, пц10) = ДМул8_ц16(пц1, пц2)</p>	
<p>(пц3, пц4, пц5, пц6, пц7, пц8, пц9, пц10) = DMUX8_i16(пц1, пц2)</p>	

## 15.11 ДМул8\_ц32, DMUX8\_i32

### Назначение

Демультимплексор на 8 выходов целого 32-битного формата.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = DMUX8\_i32(X1, X2)$ $(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = ДМул8\_ц32(X1, X2)$	
Входные параметры: X1(ц16), X2(ц32) Выходные параметры: Y1(ц32), Y2(ц32), Y3(ц32), Y4(ц32), Y5(ц32), Y6(ц32), Y7(ц32), Y8(ц32)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

### Логика работы функции

**DMUX8\_i32** – функция демультимплексора на 8 выходов производит подключение входного целочисленного 32-битного сигнала к одному из восьми выходов.

X1 – селектор выходных сигналов.

X2 – входной сигнал.

Y1 – первый выходной сигнал.

Y2 – второй выходной сигнал.

Y3 – третий выходной сигнал.

Y4 – четвертый выходной сигнал.

Y5 – пятый выходной сигнал.

Y6 – шестой выходной сигнал.

Y7 – седьмой выходной сигнал.

Y8 – восьмой выходной сигнал.

В зависимости от значения селектора X1 одному из выходов функции Y1 - Y8 присваивается значение входного сигнала X2. Всем остальным выходам функции присваивается значение "0".

Если X1=1, то Y1=X2;

...

Если X1=8, то Y8=X2.

Пример вызова функции	
СТ:	ФБД:
<p>(ц32_2, ц32_3, ц32_4, ц32_5, ц32_6, ц32_7, ц32_8, ц32_9) = ДМул8_ц32(пц1, ц32_1)</p>	
<p>(ц32_2, ц32_3, ц32_4, ц32_5, ц32_6, ц32_7, ц32_8, ц32_9) = DMUX8_i32(пц1, ц32_1)</p>	

Здесь ц32\_1, ц32\_2, ц32\_3, ц32\_4, ц32\_5, ц32\_6, ц32\_7, ц32\_8, ц32\_9 - локальные переменные целого 32-битного типа.

## 15.12 ДМул8\_в32, DMUX8\_r32

### Назначение

Демультимплексор на 8 выходов вещественного 32-битного формата.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = \text{Дмул8\_в32}(X1, X2)$  $(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = \text{DMUX8\_r32}(X1, X2)$	
Входные параметры: X1(ц16), X2(в32) Выходные параметры: Y1(в32), Y2(в32), Y3(в32), Y4(в32), Y5(в32), Y6(в32), Y7(в32) Y8(в32)	

### Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

### Логика работы функции

**DMUX8\_r32** – функция демультимплексора на 8 выходов производит подключение входного вещественного 32-битного сигнала к одному из восьми выходов.

X1 – селектор выходных сигналов.

X2 – входной сигнал.

Y1 – первый выходной сигнал.

Y2 – второй выходной сигнал.

Y3 – третий выходной сигнал.

Y4 – четвертый выходной сигнал.

Y5 – пятый выходной сигнал.

Y6 – шестой выходной сигнал.

Y7 – седьмой выходной сигнал.

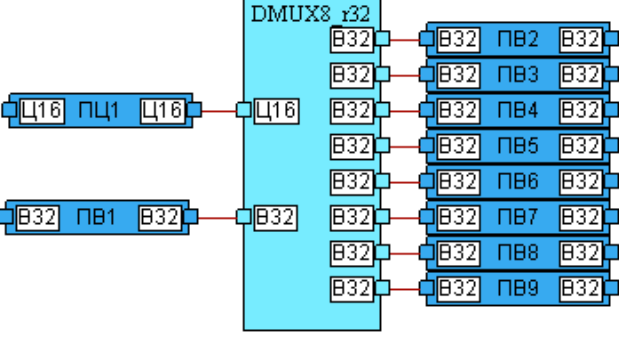
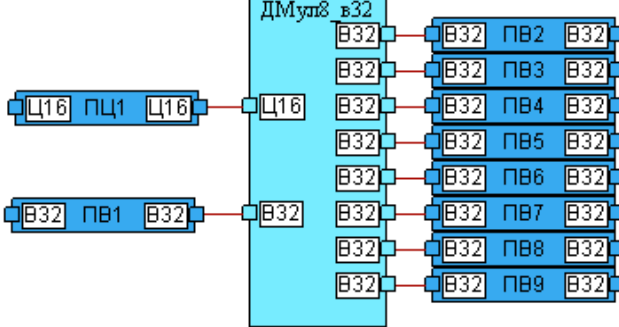
Y8 – восьмой выходной сигнал.

В зависимости от значения селектора X1 одному из выходов функции Y1 - Y8 присваивается значение входного сигнала X2. Всем остальным выходам функции присваивается значение "0".

Если X1=1, то Y1=X2;

...

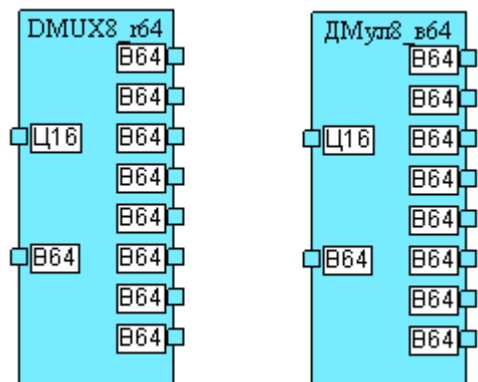
Если X1=8, то Y8=X2.

Пример вызова функции	
СТ:	ФБД:
<p>(пв2, пв3, пв4, пв5, пв6, пв7, пв8, пв9) = DMUX8_r32(пц1, пв1)</p>	
<p>(пв2, пв3, пв4, пв5, пв6, пв7, пв8, пв9) = ДМул8_в32(пц1, пв1)</p>	

15.13 ДМул8\_в64, DMUX8\_r64

Назначение

Демультимплексор на 8 выходов вещественного 64-битного формата.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = DMUX8\_r64(X1, X2)$  $(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8) = ДМул8\_в64(X1, X2)$	
Входные параметры: X1(ц16), X2(в64) Выходные параметры: Y1(в64), Y2(в64), Y3(в64), Y4(в64), Y5(в64), Y6(в64), Y7(в64), Y8(в64)	

Описание

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

Логика работы функции

**DMUX8\_r64** – функция демультимплексора на 8 выходов производит подключение входного вещественного 64-битного сигнала к одному из восьми выходов.

X1 – селектор выходных сигналов  
X2 – входной сигнал

Y1 – первый выходной сигнал  
Y2 – второй выходной сигнал  
Y3 – третий выходной сигнал  
Y4 – четвертый выходной сигнал  
Y5 – пятый выходной сигнал  
Y6 – шестой выходной сигнал  
Y7 – седьмой выходной сигнал  
Y8 – восьмой выходной сигнал

В зависимости от значения селектора X1 одному из выходов функции Y1 - Y8 присваивается значение входного сигнала X2. Всем остальным выходам функции присваивается значение "0".

Если X1=1, то Y1=X2;  
...  
Если X1=8, то Y8=X2.



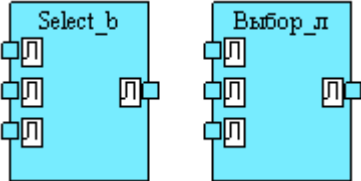
Пример вызова функции	
СТ:	ФБД:
<p>(в64_2, в64_3, в64_4, в64_5, в64_6, в64_7, в64_8, в64_9) = ДМул8_в64(пц1, в64_1)</p>	
<p>(в64_2, в64_3, в64_4, в64_5, пв64_6, в64_7, в64_8, в64_9) = DMUX_r64(пц1, в64_1)</p>	

Здесь в64\_1, в64\_2, в64\_3, в64\_4, в64\_5, в64\_6, в64\_7, в64\_8, в64\_9 - локальные переменные вещественного 64 -битного типа.

15.14      **Выбор\_л, Select\_b**

**Назначение**

Коммутатор логических переменных.

Отображение	
СТ:	ФБД:
$Y = \text{Select\_b}(X1, X2, X3)$ $Y = \text{Выбор\_л}(X1, X2, X3)$	
Входные параметры: X1(л), X2(л), X3(л) Выходные параметры: Y(л)	

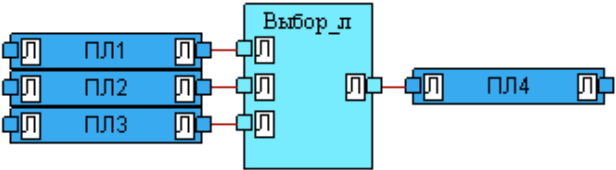
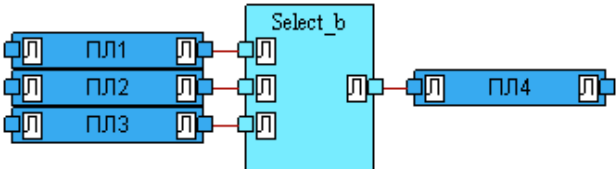
**Описание**

Функция коммутирует две переменных логического формата.

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

**Логика работы функции**

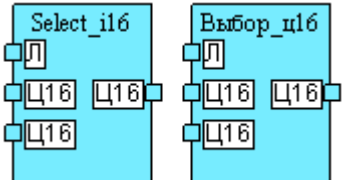
Если X1 = 0 то Y = X2,  
Если X1 = 1 то Y = X3.

Пример вызова функции	
СТ:	ФБД:
$\text{пл4} = \text{Выбор\_л}(\text{пл1}, \text{пл2}, \text{пл3})$	
$\text{пл4} = \text{Select\_b}(\text{пл1}, \text{пл2}, \text{пл3})$	

### 15.15 Выбор\_ц16, Select\_i16

#### Назначение

Коммутатор целых переменных.

Отображение	
СТ:	ФБД:
$Y = \text{Select\_i16}(X1, X2, X3)$ $Y = \text{Выбор\_ц16}(X1, X2, X3)$	
Входные параметры: X1(л), X2(ц16), X3(ц16) Выходные параметры: Y(ц16)	

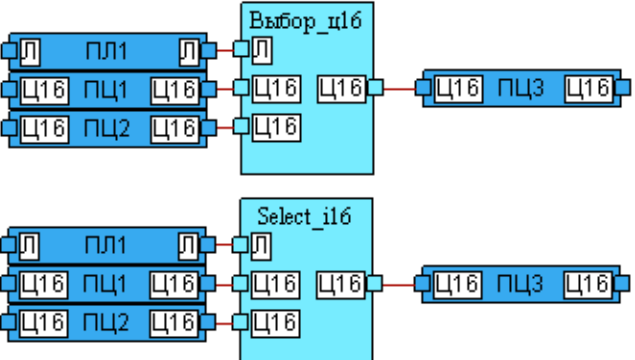
#### Описание

Функция коммутирует две переменных целого 16-битного формата.

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

#### Логика работы функции

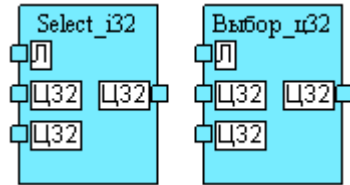
Если X1 = 0 то Y = X2,  
 Если X1 = 1 то Y = X3.

Пример вызова функции	
СТ:	ФБД:
$\text{пц3} = \text{Выбор\_ц16}(\text{пл1}, \text{пц1}, \text{пц2})$  $\text{пц3} = \text{Select\_i16}(\text{пл1}, \text{пц1}, \text{пц2})$	

15.16      **Выбор\_ц32, Select\_i32**

**Назначение**

Коммутатор целых переменных 32-битного формата.

Отображение	
СТ:	ФБД:
$Y = \text{Выбор\_ц32}(X1, X2, X3)$ $Y = \text{Select\_i32}(X1, X2, X3)$	
Входные параметры: X1(л), X2(ц32), X3(ц32) Выходные параметры: Y(ц32)	

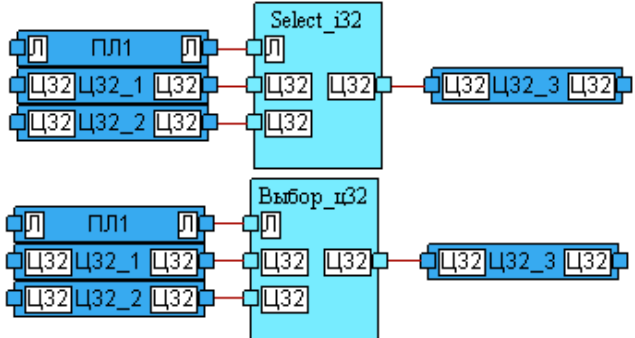
**Описание**

Функция коммутирует две переменных целого 32-битного формата.

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

**Логика работы функции**

Если X1 = 0 то Y = X2,  
Если X1 = 1 то Y = X3.

Пример вызова функции	
СТ:	ФБД:
$\text{ц32\_3} = \text{Select\_i32}(\text{пл1}, \text{ц32\_1}, \text{ц32\_2})$ $\text{ц32\_3} = \text{Выбор\_ц32}(\text{пл1}, \text{ц32\_1}, \text{ц32\_2})$	

Здесь ц32\_1, ц32\_2, ц32\_3 - локальные переменные целого 32-битного типа.

## 15.17 Выбор\_v32, Select\_r32

**Назначение**

Коммутатор вещественных переменных.

Отображение	
СТ:	ФБД:
$Y = \text{Select\_r32}(X1, X2, X3)$ $Y = \text{Выбор\_в32}(X1, X2, X3)$	
Входные параметры: X1(л), X2(в32), X3(в32) Выходные параметры: Y(в32)	

**Описание**

Функция коммутирует две переменных вещественного 32-битного формата.

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

**Логика работы функции**

Если X1 = 0 то Y = X2,

Если X1 = 1 то Y = X3.

Пример вызова функции	
СТ:	ФБД:
$пв3 = \text{Select\_r32}(пл1, пв1, пв2)$ $пв3 = \text{Выбор\_в32}(пл1, пв1, пв2)$	

15.18      **Выбор\_v64, Select\_r64**

**Назначение**

Коммутатор вещественных 64-битных переменных.

Отображение	
СТ:	ФБД:
$Y = \text{Select\_r64}(X1, X2, X3)$ $Y = \text{Выбор\_v64}(X1, X2, X3)$	
Входные параметры: X1(л), X2(в64), X3(в64) Выходные параметры: Y(в64)	

**Описание**

Функция коммутирует две переменных вещественного 64-битного формата.

Применяется только для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2.

**Логика работы функции**

Если X1 = 0 то Y = X2,  
Если X1 = 1 то Y = X3.

Пример вызова функции	
СТ:	ФБД:
$\text{в64\_3} = \text{Select\_r64}(\text{пл1}, \text{в64\_1}, \text{в64\_2})$	
$\text{в64\_3} = \text{Выбор\_v64}(\text{пл1}, \text{в64\_1}, \text{в64\_2})$	

Здесь в64\_1, в64\_2, в64\_3 - локальные переменные вещественного 64 -битного типа.

## 15.19 Шаг, STEP

## Назначение

Шаговое управление.

Отображение	
СТ:	ФБД:
$(Y1, Y2, Y3, Y4, Y5) = \text{Шаг}(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10)$  $(Y1, Y2, Y3, Y4, Y5) = \text{Step}(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10)$	
Входные параметры: X1(ц16), X2(л), X3(л), 4(л), X5(ц16), X6(ц16), X7(л), X8(л), X9(ц16), X10(ц16) Выходные параметры: Y1(ц16), Y2(ц16), Y3(л), Y4(л), Y5(л)	

## Описание

Функция для «старых» платформ (СРВК версии 6.5 для ОС QNX и LINUX и среда исполнения КРУГОЛ версии 1.0). не применяется.

## Логика работы функции

**STEP** – функция шагового управления формирует логическую последовательность во времени выходных дискретных сигналов в соответствии с заданными для каждого выходного сигнала условиями формирования и интервалом времени. Логическая последовательность выходных сигналов, а также условий формирования и интервалов времени должны разворачиваться с помощью соответствующих мультиплексоров на входе и демультиплексора на выходе.

X1 – порядковый номер функции (от 1 до 20 000). Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – сигнал перевода алгоритма в автоматический режим работы (AUTO).

X3 – сигнал перехода к следующему шагу в ручном режиме работы (PASS).

X4 – сигнал установки алгоритма в исходное состояние в ручном режиме работы (RESET).

X5 – константа, определяющая количество шагов в алгоритме (MOUN).

X6 – сигнал, определяющий количество выполняемых циклов (цикл - от первого до последнего шага) (NUM).

X7 – сигнал, разрешающий запуск алгоритма из исходного состояния (COND).

X8 – сигнал текущего шага, определяющий условие его выполнения (CON\_1).

X9 – заданное время выполнения текущего шага, сек (TIM\_1).

X10 – номер шага перехода (JMP\_1).

Y1 – номер выполняемого шага (STEP\_S).

Y2 – текущее время выполнения шага (TIME\_S).

Y3 – окончание выполнения последнего шага и переход в исходное состояние (END\_S).

Y4 – останов выполнения текущего шага (STOP\_S).

Y5 – сигнал управления на текущем шаге (OUT\_1).

Обеспечение заданных временных характеристик шагов алгоритма достигается тем, что алгоритм шагового управления выполняется периодически. Длительность одного периода соответствует периоду просчета алгоритмов управления контроллера. На каждом периоде алгоритма формируется (устанавливается в состояние TRUE или FALSE) только один выходной сигнал, таким образом, от сброса одного выходного сигнала до установки другого происходит не менее одного периода просчета алгоритмов управления.

После выполнения последнего шага в алгоритме анализируется задание на число проходов (циклов) программы NUM. Если NUM = 1 или 0, то программа считается завершенной, формируется сигнал окончания END\_S = TRUE и алгоритм приводится в исходное состояние.

Если NUM > 1 и число выполненных циклов не достигло заданного, то алгоритм организует переход на выполнение следующего цикла, начиная с первого шага, (сигнал END\_S при этом остается равным FALSE), при этом проверка наличия сигнала запуска алгоритма COND не производится. Операции завершения алгоритма (формирование END\_S = TRUE и установка алгоритма в исходное состояние) в этом случае производится после того, как будет выполнено заданное количество циклов программы.

Если же задано отрицательное число циклов NUM, то организуется бесконечный повтор программы до тех пор, пока не последует перевод алгоритма в ручной режим работы.

Сигнал END\_S сбрасывается в значение FALSE при новом запуске алгоритма из исходного состояния.

Если на каком-либо шаге признак ветвления JMP\_1 имеет значение больше заданного количества шагов алгоритма, то по завершению выполнения данного шага производится переход на конец программы.

### Режим ручного управления

При поступлении сигнала запуска COND алгоритм формирует контрольный сигнал STOP\_S = TRUE и переходит в режим ожидания команды на запуск первого шага алгоритма.

Запуск первого шага в ручном режиме осуществляется по значению сигнала PASS = TRUE. Получив сигнал запуска шага, алгоритм при наличии условия CON\_1 = TRUE, формирует выходной сигнал OUT\_1 = TRUE и номеру шага присваивается значение STEP\_S = 1. Состояние выходного сигнала OUT\_1=TRUE существует в течение данного шага все время, пока выполняется условие CON\_1=TRUE. На время невыполнения условия CON\_1 (CON\_1 = FALSE) выходной сигнал OUT\_1 сбрасывается (OUT\_1 = FALSE). Завершив формирование выходных сигналов OUT\_1 и STEP\_S, алгоритм переходит в режим ожидания сигнала окончания шага PASS. При PASS = TRUE переход к выполнению следующего шага (запуск следующего шага) производится при наличии условия CON\_1 = TRUE. При CON\_1 = FALSE алгоритм снова переходит в режим ожидания сигнала PASS и так до тех пор, пока не будет сформировано условие CON\_1 = TRUE.

Номер следующего шага для перехода определяется в соответствии с признаком JMP\_1:

- при JMP\_1 = 0 выбирается следующий по порядку номер шага;
- при JMP\_1 > 0 выбирается номер шага равный значению JMP\_1;
- при JMP\_1 < 0 выбирается номер шага равный значению |JMP\_1|.



Все последующие шаги выполняются аналогично первому с той только разницей, что запуск текущего шага осуществляется предыдущим шагом (см. выше), а параметру STEP\_S присваивается значение текущего номера шага.

В режиме ручного управления предусмотрена возможность установки алгоритма в исходное состояние на любом шаге по сигналу RESET = TRUE. При этом выполнение текущего цикла алгоритма прекращается и алгоритм начинает работать с начала (с анализа сигнала запуска алгоритма COND и с первого шага).

#### **Режим автоматического управления**

При поступлении сигнала запуска COND = TRUE алгоритм из начального состояния переходит на выполнения первого шага безусловно (сигнал COND в данном случае выполняет так же функцию сигнала запуска шага).

Получив сигнал запуска шага при наличии условия CON\_1 = TRUE, алгоритм формирует выходной сигнал OUT\_1 = TRUE. При этом номер шага STEP\_S = 1 и запускается таймер интервала времени. Состояние выходного сигнала OUT\_1 = TRUE существует в течение данного шага все время, пока выполняется условие CON\_1 = TRUE. На время невыполнения условия CON\_1=FALSE выходной сигнал сбрасывается OUT\_1=FALSE.

По истечении заданного интервала времени и наличии условия CON\_1 производится переход к следующему шагу. Выбор следующего шага производится по признаку JMP\_1.

При JMP\_1 >= 0 условие CON\_1 контролируется только после истечения заданного интервала времени и при отсутствии условия CON\_1 = FALSE производится останов программы с формированием сигнала останова STOP\_S=TRUE. После останова, запуск алгоритма на продолжение осуществляется по входному сигналу PASS = TRUE. При этом вновь производится контроль наличия условия CON\_1 (интервал времени, соответствующий данному шагу, при этом считается исчерпанным), при CON\_1 = TRUE производится переход к заданному в JMP\_1 шагу: при JMP\_1 = 0 к следующему по порядку, при JMP\_1 > 0 к номеру равному значению параметра JMP\_1. Если при перезапуске алгоритма на продолжение фиксируется отсутствие условия CON\_1 = FALSE, то алгоритм вновь останавливается в ожидании сигнала на перезапуск и т.д.

Если JMP\_1 < 0, то ведется контроль наличия условия CON\_1 в течение всего, заданного для данного шага, интервала времени. В том случае, если наличие условия перехода CON\_1 присутствует в течение всего заданного интервала времени то, по окончании интервала, производится переход к следующему по порядку шагу. В том случае, если будет зафиксировано отсутствие условия перехода CON\_1, то, не дожидаясь окончания заданного интервала времени, производится останов программы с формированием сигнала останова STOP\_S, и при последующем пуске алгоритма по входному сигналу PASS производится переход к шагу, номер которого записан в JMP\_1 как отрицательное число. Наличие условия перехода CON\_1 при запуске алгоритма на продолжение не контролируется. Если модуль признака ветвления |JMP\_1| остановленного шага равен номеру остановленного шага STEP\_S, то алгоритм организует продолжение данного шага с продолжением счета времени шага со значения, зафиксированного в момент останова.

Все последующие шаги выполняются аналогично первому с той только разницей, что запуск текущего шага осуществляется предыдущим шагом (см. выше), а параметру STEP\_S присваивается значение текущего номера шага.

### Переключение режимов

Любое переключение режимов из автоматического в ручное и обратно не изменяет состояние алгоритма. Алгоритм продолжает выполнять свои функции в момент перехода из того состояния, в котором он находился.

Сигнал останова  $STOP\_S = TRUE$  формируется всякий раз, когда производится останов алгоритма. При работе в режиме ручного управления сигнал  $STOP\_S$  всегда находится в состоянии  $TRUE$ .

При переводе режима из ручного в автоматический из произвольного состояния алгоритма интервал времени, соответствующий данному шагу, считается исчерпанным. При  $JMP\_1 \geq 0$  алгоритм проверяет наличие условия  $CON\_1$  и при  $CON\_1 = TRUE$  переходит к заданному в параметре  $JMP\_1$  шагу. При  $JMP\_1 < 0$  наличие условия  $CON\_1$  не проверяется, а алгоритм переходит к шагу, номер которого записан в  $JMP\_1$  как отрицательное число.

### Первое обращение к функциональному блоку

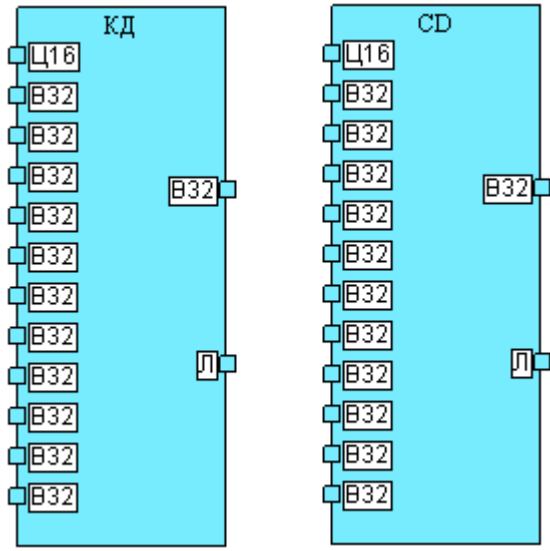
При первом обращении функциональный блок устанавливается в исходное состояние: выходным сигналам  $STEP\_S$  и  $TIME\_S$  присваивается значение 0, а сигналам  $END\_S$ ,  $STOP\_S$  и  $OUT\_1$  – значение  $FALSE$ .

Пример вызова функции	
СТ:	ФБД:
<p>(пц6, пц7, пл6, пл7, пл8) = Step(пц1, пл1, пл2, пл3, пц2, пц3, пл4, пл5, пц4, пц5)</p>	
<p>(пц6, пц7, пл6, пл7, пл8) = Шаг(пц1, пл1, пл2, пл3, пц2, пц3, пл4, пл5, пц4, пц5)</p>	

## 15.20 КД, CD

## Назначение

Коммутатор диапазонов. В зависимости от диапазона значений, в который попадает входной параметр, функция выдает на выходе значение, заданное на соответствующем данному диапазону входе.

Отображение	
СТ:	ФБД:
$(Y1, Y2) = \text{КД}(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12)$  $(Y1, Y2) = \text{CD}(X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12)$	
Входные параметры: X1 (ц16), X2 (в32), X3 (в32), X4 (в32), X5 (в32), X6 (в32), X7 (в32), X8 (в32), X9 (в32), X10 (в32), X11 (в32), X12 (в32) Выходные параметры Y1 (в32), Y2 (л)	

## Описание

Функция не применяется для «старых» платформ (СРВК версии 6.5 для ОС QNX и LINUX и среда исполнения КРУГОЛ версии 1.0).

## Логика работы функции

X1 – номер алгоблока. Для платформ СРВК, начиная с версии 7.1, и среды исполнения КРУГОЛ, начиная с версии 2.1 - необязательный вход (если вход не привязан явно, то номер будет подставлен автоматически).

X2 – входной параметр.

X3 – значение для первого диапазона.

X4 – конец первого диапазона.

X5 – значение для второго диапазона.

X6 – конец второго диапазона.

X7 – значение для третьего диапазона.

X8 – конец третьего диапазона.

X9 – значение для четвертого диапазона.

X10 – конец четвертого диапазона.

X11 – значение для пятого диапазона.

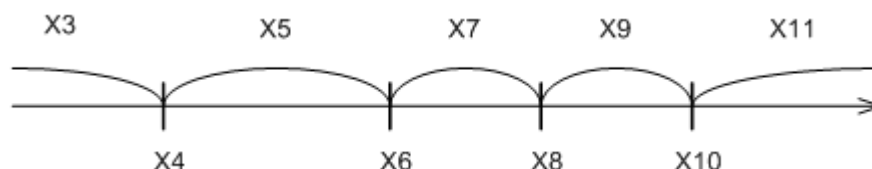
X12 – гистерезис.

Y1 – значение выходного параметра.

Y2 – импульсный выход, сигнализирующий о смене диапазона.

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

В зависимости от диапазона значений, в который попадает входной параметр, функция выдает на выходе значение, заданное на соответствующем данному диапазону входе. Т.е. если значение параметра меньше X4 (попадает в первый диапазон), то на выходе выдается значение для первого диапазона (вход X3). Если параметр больше X4, но меньше X6, то на выходе выдается значение для второго диапазона (вход X5). И так далее.



Распределение значений, выдаваемых на выход, в зависимости от значения входа.

Входной параметр считается принадлежащим диапазону «i» (при нулевом гистерезисе), если верно условие:

«Конец диапазона (i-1)» < «входной параметр» < «конец диапазона «i»

Вход X12 определяет, с каким гистерезисом будет происходить переход из одного диапазона в другой. Гистерезис (половина ширины зоны нечувствительности) задается в единицах измерения параметра.

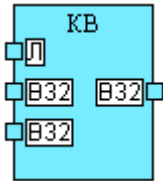
Для отключения лишнего диапазона надо, чтобы конец этого диапазона был численно меньше конца предыдущего диапазона.

Пример вызова функции	
СТ:	ФБД:
(пв12, пл1) = CD (пц1, пв1, пв2, пв3, пв4, пв5, пв6, пв7, пв8, пв9, пв10, пв11)	
(пв12, пл1) = КД (пц1, пв1, пв2, пв3, пв4, пв5, пв6, пв7, пв8, пв9, пв10, пв11)	

## 15.21 КВ

**Назначение**

Коммутатор вещественных 32-битных переменных.

Отображение	
СТ:	ФБД:
$Y = KB(X1, X2, X3)$	
Входные параметры: X1(л), X2(в32), X3(в32) Выходные параметры: Y(в32)	

**Описание**

Функция коммутирует две переменные вещественного 32-битного формата.

Применяется для платформ СРВК версии ниже 8.0, и среды исполнения КРУГОЛ версии ниже 2.2.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

**Логика работы функции**

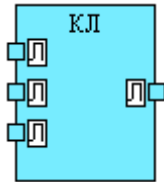
Если  $X1 = 0$  то  $Y = X2$ ,  
 Если  $X1 = 1$  то  $Y = X3$ .

Пример вызова функции	
СТ:	ФБД:
$пв3 = KB(пл1, пв1, пв2)$	

## 15.22 КЛ

### Назначение

Коммутатор логических переменных.

Отображение	
СТ:	ФБД:
$Y = \text{КЛ}(X1, X2, X3)$	
Входные параметры: X1(л), X2(л), X3(л) Выходные параметры: Y(л)	

### Описание

Функция коммутирует две переменные логического формата.

Применяется для платформ СРВК версии ниже 8.0, и среды исполнения КРУГОЛ версии ниже 2.2.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

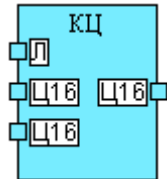
### Логика работы функции

Если X1 = 0 то Y = X2,  
 Если X1 = 1 то Y = X3.

Пример вызова функции	
СТ:	ФБД:
$\text{пл4} = \text{КЛ}(\text{пл1}, \text{пл2}, \text{пл3})$	

**15.23 КЦ****Назначение**

Коммутатор целых 16-битных переменных.

Отображение	
СТ:	ФБД:
$Y = \text{КЦ}(X1, X2, X3)$	
Входные параметры: X1(л), X2(ц16), X3(ц16) Выходные параметры: Y(ц16)	

**Описание**

Функция коммутирует две переменные целого 16-битного формата.

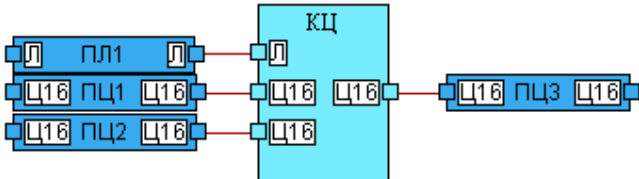
Применяется для платформ СРВК версии ниже 8.0, и среды исполнения КРУГОЛ версии ниже 2.2.

**Примечание.** Для платформ СРВК, начиная с версии 8.0, и среды исполнения КРУГОЛ, начиная с версии 2.2, одноименная функция имеется в группе «Устаревшие функции».

**Логика работы функции**

Если X1 = 0 то Y = X2,

Если X1 = 1 то Y = X3.

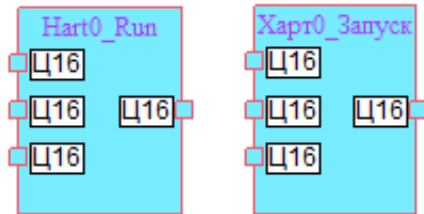
Пример вызова функции	
СТ:	ФБД:
$\text{пц3} = \text{КЦ}(\text{пл1}, \text{пц1}, \text{пц2})$	

## 16 ПРОТОКОЛ HART

### 16.1 Hart0\_Run, Харт0\_Запуск

#### Назначение

Протокол HART, команда 0. Считать уникальный идентификатор. Запуск выполнения команды.

Отображение	
СТ:	ФБД:
Y1 = Hart0_Run (X1, X2, X3) Y1 = Харт0_Запуск (X1, X2, X3)	
Входные параметры: X1(ц16), X2(ц16), X3(ц16) Выходные параметры: Y1(ц16)	

#### Описание

**Hart0\_Run, Харт0\_Запуск** – функция получения уникального идентификатора HART-датчика.

Применяется для платформы «СРБК TREI-5B-04/05 8.1 Linux».

#### Логика работы функции

X1 – Аппаратный адрес модуля M941A.

X2 – Номер канала на модуле M941A, к которому подключен HART-датчик.

X3 – Короткий адрес HART-устройства.

Y1 – Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Команда 0 HART-протокола используется для получения длинного адреса HART-датчика и разнообразной информации по нему.

Данная функция запускает выполнение команды HART-модемом, который находится на плате M941A. Результат выполнения команды нужно проверять функцией Hart0\_Result (Харт0\_Рез).

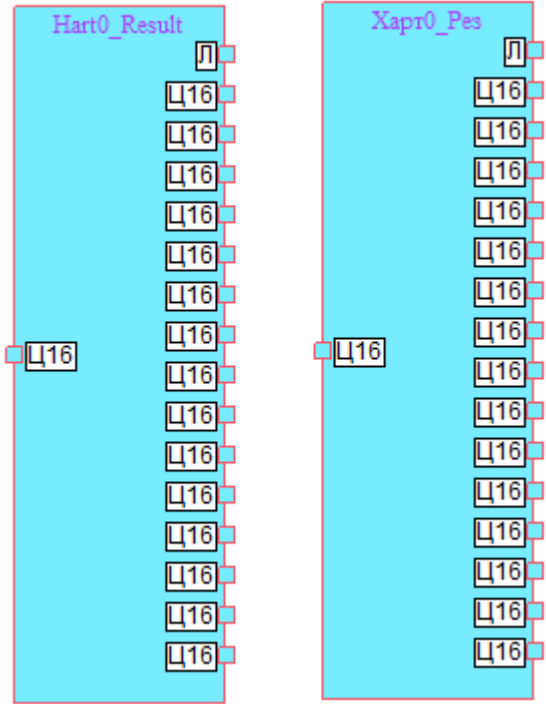


Пример вызова функции	
СТ:	ФБД:
<p>пц4= Hart0_Run (пц1, пц2, пц3) пц4= Харт0_Запуск(пц1, пц2, пц3)</p>	<p>The diagram illustrates two function blocks, Hart0_Run and Харт0_Запуск, each represented by a light blue rectangle. For Hart0_Run, three blue input blocks labeled пц1, пц2, and пц3 are connected to the left side of the function block. A single blue output block labeled пц4 is connected to the right side. The same structure is repeated for the Харт0_Запуск block, with inputs пц1, пц2, пц3 and output пц4.</p>

## 16.2 Hart0\_Result, Харт0\_Рез

### Назначение

Протокол HART, команда 0. Проверка завершения выполнения команды.

Отображение	
СТ:	ФБД:
<p>(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13, Y14, Y15, Y16) = Hart0_Result (X1)</p> <p>(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12, Y13, Y14, Y15, Y16) = Харт0_Рез(X1)</p>	
<p>Входные параметры: X1(ц16))</p> <p>Выходные параметры: Y1(л), Y2(ц16), Y3(ц16), Y4(ц16), Y5(ц16), Y6(ц16), Y7(ц16), Y8(ц16), Y9(ц16), Y10(ц16), Y11(ц16), Y12(ц16), Y13(ц16), Y14(ц16), Y15(ц16), Y16(ц16), Y1(ц16)</p>	

### Описание

**Hart0\_Result, Харт0\_Рез** – функция получения уникального идентификатора HART-датчика. Проверка завершения выполнения команды.

Применяется для платформы «СРБК TREI-5B-04/05 8.1 Linux».

### Логика работы функции

X1 - Аппаратный адрес модуля M941A.

Y1 - Признак успешного выполнения команды (=1 - команда успешно выполнена)

Y2 - Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Y3 - "Статус" от HART-устройства.

Y4 - Длинный адрес HART-устройства (часть 1).

Y5 - Длинный адрес HART-устройства (часть 2).

Y6 - Длинный адрес HART-устройства (часть 3).

Y7 - Код изготовителя устройства HART  
Y8 - Код тип устройства HART  
Y9 - Количество преамбул устройства HART  
Y10 - Версия универсальных команд устройства HART  
Y11 - Версия специальных команд устройства HART  
Y12 - Версия программного обеспечения устройства HART  
Y13 - Версия аппаратной части устройства HART  
Y14 - Флаги функций устройства HART  
Y15 - Идентификационный номер устройства HART. Младшие 2 байта  
Y16 - Идентификационный номер устройства HART. Старший байт

Команда 0 HART-протокола используется для получения длинного адреса HART-датчика и разнообразной информации по нему.

Данная функция ожидает выполнение команды 0 HART-модемом, который находится на плате M941A. Запуск команды стартует при вызове функции Hart0\_Run (Харт0\_Запуск).

Параметр Y1 будет возвращать значение 0 пока выполняется команда. Если команда успешно выполнена, то параметр Y1 примет значение 1, а параметр Y2 – значение 0, и будут возвращены значения параметров Y3-Y16. Если произошла ошибка при выполнении команды, Y1 останется в значении 0, а параметр Y2 примет значение кода ошибки, и значения параметров Y3-Y16 обнулятся.

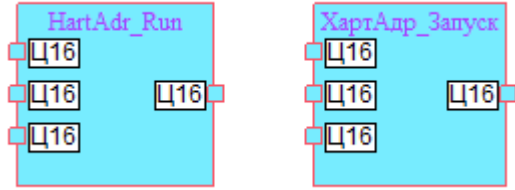
Выходные аргументы Y4-Y6 в дальнейшем должны использоваться в качестве адреса HART-устройства в остальных функциях работы с HART-протоколом. Выходные аргументы Y3, Y7-Y16 имеют информационный характер.

Пример вызова функции	
СТ:	ФБД:
<p>(пл1, пц2, пц3, пц4, пц5, пц6, пц7, пц8, пц9, пц10, пц11, пц12, пц13, пц14, пц15, пц16) = Hart0_Result (пц1)</p> <p>(пл1, пц2, пц3, пц4, пц5, пц6, пц7, пц8, пц9, пц10, пц11, пц12, пц13, пц14, пц15, пц16) = Харт0_Рез(пц1)</p>	<p>The diagram illustrates two function blocks, <b>Hart0_Result</b> and <b>Харт0_Рез</b>, each represented by a light blue rectangle. Each block has 16 outputs on its right side, labeled <b>пл1</b> through <b>пл16</b>, and one input on its left side, labeled <b>пц1</b>. The inputs are connected to the blocks via red lines. The outputs are connected to the blocks via green lines. The two blocks are stacked vertically, with <b>Hart0_Result</b> on top and <b>Харт0_Рез</b> on the bottom.</p>

### 16.3 HartAdr\_Run, ХартАдр\_Запуск

#### Назначение

Преобразовать короткий адрес HART-устройства в длинный (укороченный вариант функции Hart0\_Run (Харт0\_Запуск)). Запуск выполнения команды 0.

Отображение	
СТ:	ФБД:
Y1 = HartAdr_Run (X1, X2, X3) Y1 = ХартАдр_Запуск (X1, X2, X3)	
Входные параметры: X1(ц16), X2(ц16), X3(ц16) Выходные параметры: Y1(ц16)	

#### Описание

**HartAdr\_Run, ХартАдр\_Запуск** – функция преобразования короткого адрес HART-устройства в длинный.

Применяется для платформы «СРБК TREI-5B-04/05 8.1 Linux».

#### Логика работы функции

X1 – Аппаратный адрес модуля M941A.

X2 – Номер канала на модуле M941A, к которому подключен HART-датчик.

X3 – Короткий адрес HART-устройства.

Y1 – Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Аналог функции Hart0\_Run (Харт0\_Запуск). Отличие в том, что «парная» функция HartAdr\_Result (ХартАдр\_Рез) вернет только длинный адрес HART-устройства, прочая информация будет опущена.

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Пример вызова функции	
СТ:	ФБД:
<p>пц4= HartAdr_Run (пц1, пц2, пц3)</p> <p>пц4= ХартАдр_Запуск(пц1, пц2, пц3)</p>	<pre> graph LR     subgraph HartAdr_Run         direction TB         I1[пц1]         I2[пц2]         I3[пц3]         O1[пц4]         I1 --&gt; F1         I2 --&gt; F1         I3 --&gt; F1         F1 --&gt; O1     end     subgraph ХартАдр_Запуск         direction TB         I4[пц1]         I5[пц2]         I6[пц3]         O2[пц4]         I4 --&gt; F2         I5 --&gt; F2         I6 --&gt; F2         F2 --&gt; O2     end             </pre>

## 16.4 HartAdr\_Result, ХартАдр\_Рез

### Назначение

Преобразовать короткий адрес HART-устройства в длинный (укороченный вариант функции Hart0\_Result (Харт0\_Рез)). Проверка завершения выполнения команды.

Отображение	
СТ:	ФБД:
(Y1, Y2, Y3, Y4, Y5, Y6) = HartAdr_Result (X1) (Y1, Y2, Y3, Y4, Y5, Y6) = ХартАдр_Рез(X1)	
Входные параметры: X1(ц16) Выходные параметры: Y1(л), Y2(ц16), Y3(ц16), Y4(ц16), Y5(ц16), Y6(ц16)	

### Описание

**HartAdr\_Result, ХартАдр\_Рез** – функция преобразования короткого адрес HART-устройства в длинный, проверка завершения выполнения команды.

Применяется для платформы «СРБК TREI-5B-04/05 8.1 Linux».

### Логика работы функции

X1 - Аппаратный адрес модуля M941A.

Y1 - Признак успешного выполнения команды (=1 - команда успешно выполнена)

Y2 - Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Y3 - "Статус" от HART-устройства.

Y4 - Длинный адрес HART-устройства (часть 1).

Y5 - Длинный адрес HART-устройства (часть 2).

Аналог функции Hart0\_Result(Харт0\_Рез). Отличие в том, что возвращается только длинный адрес HART-устройства, прочая информация будет опущена.

Данная функция ожидает выполнение команды, которая стартует при вызове функции HartAdr\_Run (ХартАдр\_Запуск). В остальном логика работы функции идентична функции Hart0\_Result(Харт0\_Рез).

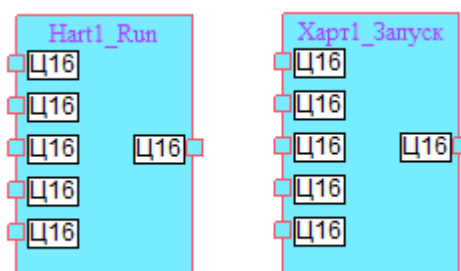
Пример вызова функции	
СТ:	ФБД:
<p>(пл1, пц2, пц3, пц4, пц5, пц6) = HartAdr_Result (пц1)</p> <p>(пл1, пц2, пц3, пц4, пц5, пц6) = ХартАдр_Рез(пц1)</p>	<p>The diagram illustrates two function blocks used for data routing. The top block, labeled 'HartAdr_Result', receives an input from 'пц1' and outputs data to six channels: 'пл1', 'пц2', 'пц3', 'пц4', 'пц5', and 'пц6'. The bottom block, labeled 'ХартАдр_Рез', also receives an input from 'пц1' and outputs data to the same six channels. The outputs are represented by blue boxes with green outlines, and the connections are shown as red lines.</p>



## 16.5 Hart1\_Run, Харт1\_Запуск

### Назначение

Протокол HART, команда 1. Считать первичную переменную. Запуск выполнения команды.

Отображение	
СТ:	ФБД:
Y1 = Hart1_Run (X1, X2, X3, X4, X5) Y1 = Hart1_Запуск(X1, X2, X3, X4, X5)	
Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16), X5(ц16) Выходные параметры: Y1(ц16)	

### Описание

**Hart1\_Run, Харт1\_Запуск** – функция получения первичной переменной.

Применяется для платформы «СРБК TREI-5B-04/05 8.1 Linux».

### Логика работы функции

X1 – Аппаратный адрес модуля M941A.

X2 – Номер канала на модуле M941A, к которому подключен HART-датчик.

X3 - Длинный адрес HART-устройства (часть 1).

X4 - Длинный адрес HART-устройства (часть 2).

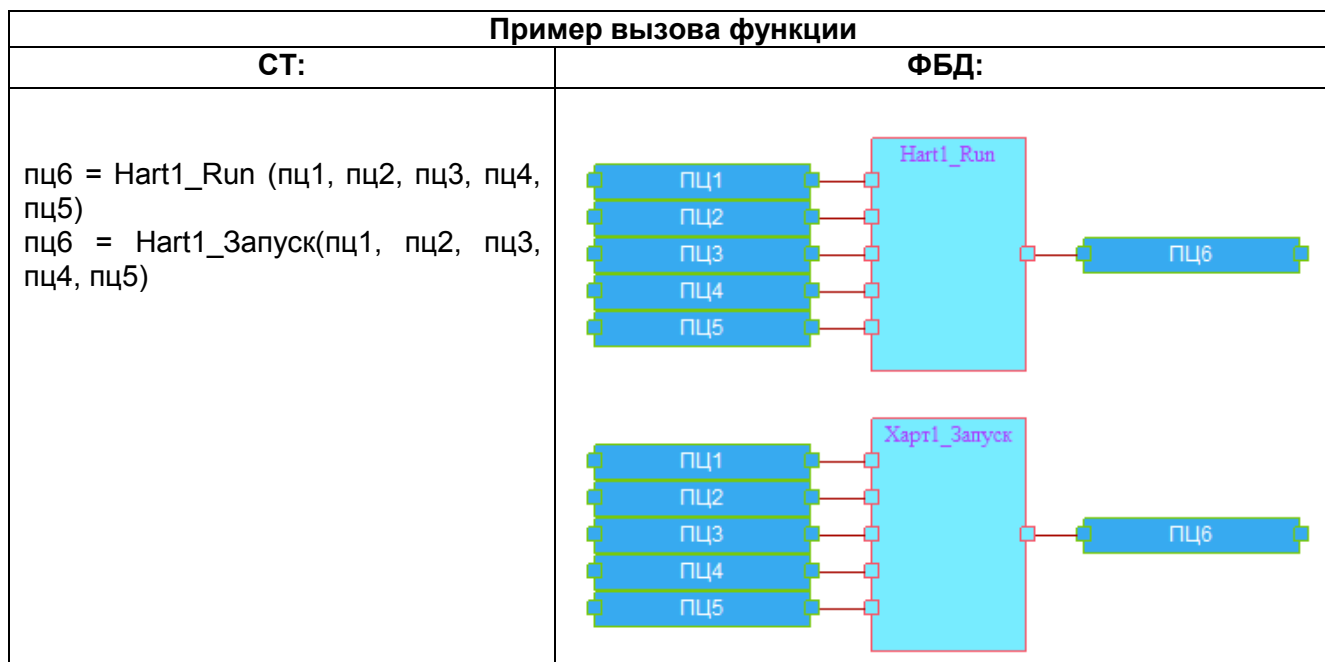
X5 - Длинный адрес HART-устройства (часть 3).

Y1 – Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Команда 1 HART-протокола используется для считывания значения первичной переменной с HART-датчика.

Данная функция запускает выполнение команды HART-модемом, который находится на плате M941A. Результат выполнения команды нужно проверять функцией Hart1\_Result (Харт1\_Рез).

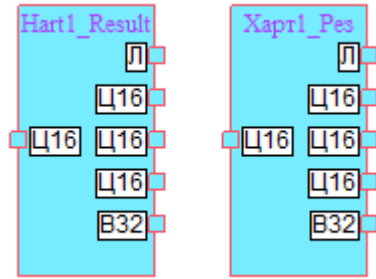
Длинный адрес устройства (параметры X3-X5) получается от функций HartAdr\_Result (ХартАдр\_Рез) или Hart0\_Result (Харт0\_Рез). Адрес HART-устройства достаточно получить только один раз при старте, затем он не меняется.



## 16.6 Hart1\_Result, Харт1\_Рез

### Назначение

Протокол HART, команда 1. Считать первичную переменную. Проверка завершения выполнения команды.

Отображение	
СТ:	ФБД:
(Y1, Y2, Y3, Y4, Y5) = Hart1_Result (X1) (Y1, Y2, Y3, Y4, Y5) = Харт1_Рез(X1)	
Входные параметры: X1(ц16) Выходные параметры: Y1(л), Y2(ц16), Y3(ц16), Y4(ц16), Y5(в32)	

### Описание

**Hart1\_Result, Харт1\_Рез** – функция получения первичной переменной, проверка завершения выполнения команды.

Применяется для платформы «СРБК TREI-5B-04/05 8.1 Linux».

### Логика работы функции

X1 - Аппаратный адрес модуля М941А.

Y1 - Признак успешного выполнения команды (=1 - команда успешно выполнена)

Y2 - Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Y3 - "Статус" от HART-устройства.

Y4 – Код единиц измерения первичной переменной устройства HART.

Y5 – Первичная переменная устройства HART.

Команда 1 HART-протокола используется для считывания значения первичной переменной с HART-датчика.

Данная функция ожидает выполнение команды 1 HART-модемом, который находится на плате М941А. Запуск команды стартует при вызове функции Hart1\_Run (Харт1\_Запуск).

Параметр Y1 будет возвращать значение 0 пока выполняется команда. Если команда успешно выполнена, то параметр Y1 примет значение 1, а параметр Y2 – значение 0, и будут возвращены значения параметров Y3-Y5. Если произошла ошибка при выполнении

ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

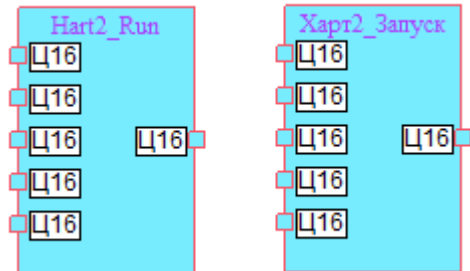
команды, Y1 останется в значении 0, а параметр Y2 примет значение кода ошибки, и значения параметров Y3-Y5 обнулятся.

Пример вызова функции	
СТ:	ФБД:
<div>(пл1, пц2, пц3, пц4, пв1) = Hart1_Result (пц1) (пл1, пц2, пц3, пц4, пв1) = Харт1_Рез(пц1)</div>	<p>The diagram illustrates two function blocks, 'Hart1_Result' and 'Харт1_Рез', which are functionally identical. Each block is represented by a light blue rectangle. The 'Hart1_Result' block has a single input on its left side labeled 'ПЦ1' and five outputs on its right side labeled 'ПЛ1', 'ПЦ2', 'ПЦ3', 'ПЦ4', and 'ПВ1'. The 'Харт1_Рез' block also has a single input on its left side labeled 'ПЦ1' and five outputs on its right side labeled 'ПЛ1', 'ПЦ2', 'ПЦ3', 'ПЦ4', and 'ПВ1'. Red lines connect the input and output ports of the blocks to their respective labels.</p>

## 16.7 Hart2\_Run, Харт2\_Запуск

### Назначение

Протокол HART, команда 2. Считать ток и процент диапазона. Запуск выполнения команды.

Отображение	
СТ:	ФБД:
Y1 = Hart2_Run (X1, X2, X3, X4, X5) Y1 = Hart2_Запуск(X1, X2, X3, X4, X5)	
Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16), X5(ц16) Выходные параметры: Y1(ц16)	

### Описание

**Hart2\_Run, Харт2\_Запуск** – функция получения тока и процента диапазона.

Применяется для платформы «СРБК TREI-5B-04/05 8.1 Linux».

### Логика работы функции

X1 – Аппаратный адрес модуля M941A.

X2 – Номер канала на модуле M941A, к которому подключен HART-датчик.

X3 - Длинный адрес HART-устройства (часть 1).

X4 - Длинный адрес HART-устройства (часть 2).

X5 - Длинный адрес HART-устройства (часть 3).

Y1 – Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Команда 2 HART-протокола используется для считывания значения первичной переменной с HART-датчика.

Данная функция запускает выполнение команды HART-модемом, который находится на плате M941A. Результат выполнения команды нужно проверять функцией Hart2\_Result (Харт2\_Рез).

Длинный адрес устройства (параметры X3-X5) получается от функций HartAdr\_Result (ХартАдр\_Рез) или Hart0\_Result (Харт0\_Рез). Адрес HART-устройства достаточно получить только один раз при старте, затем он не меняется.

Пример вызова функции	
СТ:	ФБД:
<p>пц6 = Hart2_Run (пц1, пц2, пц3, пц4, пц5)</p> <p>пц6 = Hart2_Запуск(пц1, пц2, пц3, пц4, пц5)</p>	<p>The diagram illustrates two function blocks, <b>Hart2_Run</b> and <b>Hart2_Запуск</b>, each represented by a light blue rectangle. For each block, five input blocks labeled <b>пц1</b>, <b>пц2</b>, <b>пц3</b>, <b>пц4</b>, and <b>пц5</b> (blue rectangles) are connected to the left side of the function block via red lines. A single output block labeled <b>пц6</b> (blue rectangle) is connected to the right side of each function block via a red line.</p>

## 16.8 Hart2\_Result, Харт2\_Рез

### Назначение

Протокол HART, команда 2. Считать ток и процент диапазона. Проверка завершения выполнения команды.

Отображение	
СТ:	ФБД:
(Y1, Y2, Y3, Y4, Y5) = Hart2_Result (X1) (Y1, Y2, Y3, Y4, Y5) = Харт2_Рез(X1)	
Входные параметры: X1(ц16) Выходные параметры: Y1(л), Y2(ц16), Y3(ц16), Y4(ц16), Y5(в32)	

### Описание

**Hart2\_Result, Харт2\_Рез** – функция получения тока и процента диапазона, проверка завершения выполнения команды.

Применяется для платформы «СРБК TREI-5B-04/05 8.1 Linux».

### Логика работы функции

X1 - Аппаратный адрес модуля M941A.

Y1 - Признак успешного выполнения команды (=1 - команда успешно выполнена)

Y2 - Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Y3 - "Статус" от HART-устройства.

Y4 - Ток первичной переменной(мА)

Y5 - Процент диапазона (%)

Команда 2 HART-протокола используется для считывания значения тока и процента диапазона с HART-датчика.

Данная функция ожидает выполнение команды 2 HART-модемом, который находится на плате M941A. Запуск команды стартует при вызове функции Hart2\_Run (Харт2\_Запуск).

Параметр Y1 будет возвращать значение 0 пока выполняется команда. Если команда успешно выполнялась, то параметр Y1 примет значение 1, а параметр Y2 – значение 0, и будут возвращены значения параметров Y3-Y5. Если произошла ошибка при выполнении команды, Y1 останется в значении 0, а параметр Y2 примет значение кода ошибки, и значения параметров Y3-Y5 обнулятся.

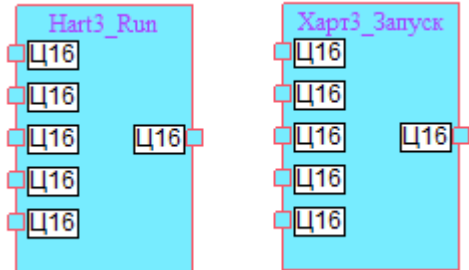
Пример вызова функции	
СТ:	ФБД:
<p>(пл1, пц2, пц3, пц4, пв1) = Hart2_Result (пц1) (пл1, пц2, пц3, пц4, пв1) = Харт2_Рез(пц1)</p>	<p>The diagram illustrates two function blocks, Hart2_Result and Харт2_Рез, each receiving an input пц1 and producing four outputs: пл1, пц2, пц3, and пв1. The outputs are connected to corresponding blocks on the right.</p>



## 16.9 Hart3\_Run, Харт3\_Запуск

### Назначение

Протокол HART, команда 3. Считать ток и значения четырех (предопределенных) динамических переменных. Запуск выполнения команды.

Отображение	
СТ:	ФБД:
Y1 = Hart3_Run (X1, X2, X3, X4, X5) Y1 = Hart3_Запуск(X1, X2, X3, X4, X5)	
Входные параметры: X1(ц16), X2(ц16), X3(ц16), X4(ц16), X5(ц16) Выходные параметры: Y1(ц16)	

### Описание

**Hart3\_Run, Харт3\_Запуск** – функция получения тока и значения четырех (предопределенных) динамических переменных.

Применяется для платформы «CPBK TREI-5B-04/05 8.1 Linux».

### Логика работы функции

X1 – Аппаратный адрес модуля M941A.

X2 – Номер канала на модуле M941A, к которому подключен HART-датчик.

X3 - Длинный адрес HART-устройства (часть 1).

X4 - Длинный адрес HART-устройства (часть 2).

X5 - Длинный адрес HART-устройства (часть 3).

Y1 – Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Команда 3 HART-протокола используется для считывания значения тока и четырех (предопределенных) динамических переменных с HART-датчика.

Данная функция запускает выполнение команды HART-модемом, который находится на плате M941A. Результат выполнения команды нужно проверять функцией Hart3\_Result (Харт3\_Рез).

Длинный адрес устройства (параметры X3-X5) получается от функций HartAdr\_Result (ХартАдр\_Рез) или Hart0\_Result (Харт0\_Рез). Адрес HART-устройства достаточно получить только один раз при старте, затем он не меняется.



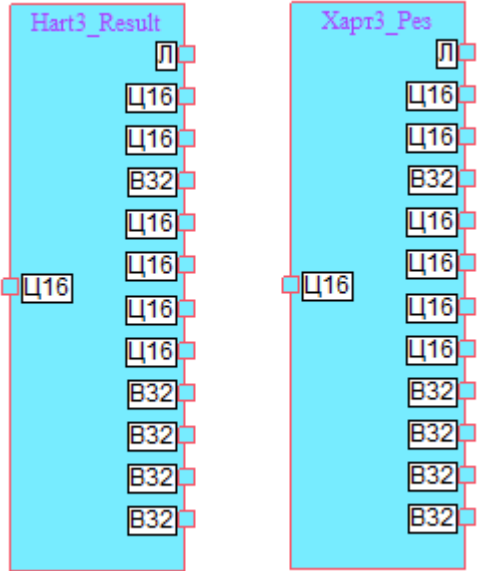
ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Пример вызова функции	
СТ:	ФБД:
<p>пц6 = Hart3_Run (пц1, пц2, пц3, пц4, пц5) пц6 = Hart3_Запуск(пц1, пц2, пц3, пц4, пц5)</p>	<p>The diagram illustrates two function blocks, Hart3_Run and Hart3_Запуск, each represented by a light blue rectangle. Each block has five inputs on the left, labeled пц1, пц2, пц3, пц4, and пц5, and one output on the right, labeled пц6. The inputs are represented by blue rectangles with small square terminals. Red lines connect each input terminal to the corresponding input on the function block. A single red line connects the output terminal of each function block to the пц6 output terminal, which is also represented by a blue rectangle with small square terminals.</p>

## 16.10 Hart3\_Result, Харт3\_Рез

### Назначение

Протокол HART, команда 3. Считать ток и значения четырех (предопределенных) динамических переменных. Проверка завершения выполнения команды.

Отображение	
СТ:	ФБД:
(Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12) = Hart3_Result (X1) (Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y8, Y9, Y10, Y11, Y12) = Харт3_Рез(X1)	
Входные параметры: X1(ц16)  Выходные параметры: Y1(л), Y2(ц16), Y3(ц16), Y4(в32), Y5(в32), Y6(ц16), Y7(ц16), Y8(ц16), Y9(ц16), Y10(в32), Y11(в32), Y12(в32)	

### Описание

**Hart3\_Result, Харт3\_Рез** – функция получения тока и значения четырех (предопределенных) динамических переменных, проверка завершения выполнения команды.

Применяется для платформы «СРБК TREI-5B-04/05 8.1 Linux».

### Логика работы функции

X1 - Аппаратный адрес модуля M941A.

Y1 - Признак успешного выполнения команды (=1 - команда успешно выполнена)

Y2 - Код ошибки при выполнении команды (0 – нет ошибки, остальные коды ошибок см. в Коды ошибок HART).

Y3 - "Статус" от HART-устройства.

Y4 - Ток первичной переменной(мА)

Y5 - Код единиц измерения первичной переменной устройства HART

Y6 - Код единиц измерения второй переменной устройства HART

Y7 - Код единиц измерения третьей переменной устройства HART

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Y8 - Код единиц измерения четвертой переменной устройства HART

Y9 - Первичная переменная устройства HART

Y10 - Вторая переменная устройства HART

Y11 - Третья переменная устройства HART

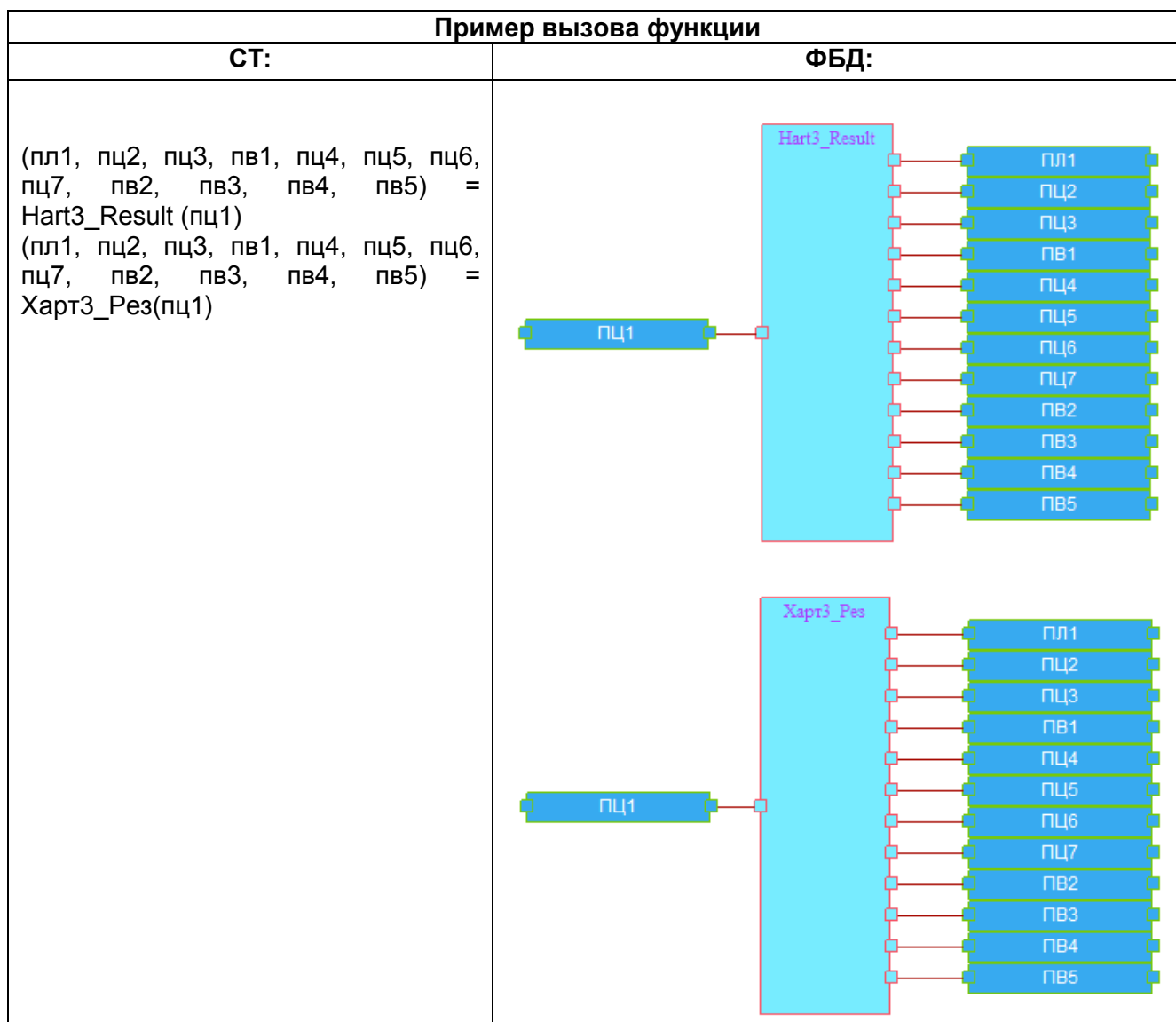
Y12 - Четвертая переменная устройства HART

Логика работы функции

Команда 3 HART-протокола используется для считывания значения тока и четырех (предопределенных) динамических переменных с HART-датчика.

Данная функция ожидает выполнение команды 3 HART-модемом, который находится на плате M941A. Запуск команды стартует при вызове функции Hart3\_Run (Харт3\_Запуск).

Параметр Y1 будет возвращать значение 0 пока выполняется команда. Если команда успешно выполнена, то параметр Y1 примет значение 1, а параметр Y2 – значение 0, и будут возвращены значения параметров Y3-Y12. Если произошла ошибка при выполнении команды, Y1 останется в значении 0, а параметр Y12 примет значение кода ошибки, и значения параметров Y3-Y7 обнулятся.



### 16.11 Коды ошибок HART

Данные коды ошибок возвращаются функциями работы с HART-модемом платы M941A. Полный список ошибок:

- 0 - нет ошибки
- 1 - устройство не отвечает (количество повторов превысило допустимое значение)
- 2 - ошибка интерфейса (ошибка фрейма)
- 3 - ошибка контрольной суммы
- 4 - ошибка ответного пакета
- 5 - мезонин не найден
- 6 - аппаратная ошибка мезонина
- 7 - обрыв линии
- 100 - ошибка выполнения запроса на уровне доступа к мастеру (ПО КРУГ)
- 101 - ошибка выполнения запроса на уровне доступа к модулю (ПО КРУГ)
- 102 - ошибка выполнения запроса на уровне доступа к словарию обмена платы (ПО КРУГ)
- 103 - таймаут при ожидании выполнения команды HART
- 104 - Неопределенная ошибка (ПО КРУГ)
- 105 - На модуле не прошивка КРУГ
- 106 - внутренняя ошибка ПО КРУГ
- 107 - контроллер резервный (разрешена работа только на основном)

### 16.12 Пример использования функций опроса датчиков по протоколу HART

Пример программы пользователя на технологическом языке СТ для Интегрированной среды разработки КРУГОЛ с использованием функций HART.

(\* Программа опрашивает HART-датчик, подключенный к одному из входов платы M941A контроллера TREI-5B-04/05

Библиотека для опроса HART-устройств устроена следующим образом:

- для каждой HART-команды существует пара функций вида HARTxxxRun/HARTxxxResult. Первая функция запускает выполнение команды, вторая ожидает её завершения (либо удачное, либо ошибку). Когда функция HARTxxxResult возвращает "команда выполнена", она заполняет значения выходных параметров, до этого момента их значения не изменяются при вызове функции.
- Для работы функций HART1xxx, HART2xxx, HART3xxx необходим "длинный адрес" устройства HART. Этот адрес можно получить вызовом функций HARTAdr\_Run/HARTAdr\_Result (или функциями HART0\_Run/HART0\_Result).

Данная программа сначала получает адрес HART-датчика, подключенного к входу 3 платы M941A с физическим адресом 1. Далее использует этот адрес для поочередного выполнения команд 2 и 3.

\*)

Программа Test\_HART  
Начало

: Задаем привязку к плате опрашиваемого HART-датчика  
пц5101=1 : Адрес модуля M941A (№ платы)  
пц5102=3 : Номер канала на модуле M941A (№ входа)  
пц5103=0 : Короткий адрес HART устройства, обычно "0"

если пл5100=0 : Первый запуск  
{  
пл5101=1 : Вызвать функцию получения идентификатора HART устройства  
пл5100=1  
}

Если ПЛ5101=1 : Получить идентификатор HART устройства (длинный адрес, разбитый на три части, используется затем для : получения результатов стандартных функций HART1, HART2, HART3).  
{

(пц5104, пц5105, пц5106, пл5101, пл5102)=HARTADR (пл5101, пл5102, пц5101, пц5102, пц5103)

: пц5104 Длинный адрес HART-устройства (часть 1)  
: пц5105 Длинный адрес HART-устройства (часть 2)  
: пц5106 Длинный адрес HART-устройства (часть 3)

если пл5101=0 : Функция получения идентификатора HART устройства отработала  
{  
пл5121=1 : Вызов функции HART, команда 2

```

        сообщение ("Адрес", пц5104, пц5105, пц5106) : выдать в роллинг
сообщение с адресом
    }
}

```

```

Если ПЛ5121=1 : HART, команда 2: Считать ток и процент диапазона.
{

```

```

(пв5121, пв5122, пл5121, пл5122)=HART2 (пл5121, пл5122, пц5101, пц5102, пц5104, пц5105, пц5106)

```

```

    : пв5121 Ток первичной переменной (мА)
    : пв5122 Процент диапазона (%) (команда №2)
    если пл5121=0 : Функция HART, команда 2 отработала
    {
        пл5131=1 : Вызов функции HART, команда 3
    }
}

```

```

Если ПЛ5131=1 : HART, команда 3: Считать ток и значения четырех
(предопределенных) динамических переменных.
{

```

```

(пв5131, пц5131, пц5132, пц5133, пц5134, пв5132, пв5133, пв5134, пв5135, пл5131, пл5132)=HART3 (пл5131, пл5132, пц5101, пц5102, пц5104, пц5105, пц5106)

```

```

    : пв5131 Ток первичной переменной (мА)
    : пц5131 Код единиц измерения первичной переменной устройства HART
    : пц5132 Код единиц измерения второй переменной устройства HART
    : пц5133 Код единиц измерения третьей переменной устройства HART
    : пц5134 Код единиц измерения четвертой переменной устройства HART
    : пв5132 Первичная переменная устройства HART
    : пв5133 Вторая переменная устройства HART
    : пв5134 Третья переменная устройства HART
    : пв5135 Четвертая переменная устройства HART
    если пл5131=0 : Функция HART, команда 3 отработала
    {
        пл5121=1 : Вызов функции HART, команда 2
    }
}

```

```

: Выкладывание значений использованных промежуточных переменных в БД для
передачи в Станцию Оператора SCADA КРУГ-2000

```

```

ва25=пв5132 : Первичная переменная устройства HART
ва25.а26=пц5131 : Код единиц измерения первичной переменной устройства
HART
ва26=пв5121 : Ток первичной переменной (мА)
ва27=пв5122 : Процент диапазона (%) (команда №2)
ва28=пв5133 : Вторая переменная устройства HART
ва28.а26=пц5132 : Код единиц измерения второй переменной устройства HART
ва29=пв5134 : Третья переменная устройства HART
ва29.а26=пц5133 : Третья переменная устройства HART
ва30=пв5135 : Четвертая переменная устройства HART
ва30.а29=пц5134 : Код единиц измерения четвертой переменной устройства
HART
Конец

```

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

**Функция HARTADR** : функция получения идентификатора через команду "0" HART без лишних аргументов

Входные\_Переменные

Начало

лог флаг1 : Работаем, пока не получены данные или ошибка выполнения функций запроса/получения данных

лог флаг2 : Флаг первого прохода, функцию HartAdr\_Run можем вызывать один раз для каждой платы, затем вызываем функцию HartAdr\_Result

Цел16 плата

Цел16 вход

Цел16 адр : короткий адрес HART устройства

Конец

Выходные\_Переменные

Начало

Цел16 адр1 : Длинный адрес HART-устройства (часть 1)

Цел16 адр2 : Длинный адрес HART-устройства (часть 2)

Цел16 адр3 : Длинный адрес HART-устройства (часть 3)

лог флаг1\_вых : флаг работы, для передачи в основное тело ПрП

лог флаг2\_вых : флаг первого прохода, для передачи в основное тело ПрП

Конец

Начало

Переменные

Начало

Цел16 ош\_запр

Лог успех

Цел16 код\_ош\_отв

Цел16 статус

Конец

флаг1\_вых=флаг1

флаг2\_вых=флаг2

если флаг2=0

{

флаг2\_вых=1

(ош\_запр)=HartAdr\_Run (плата, вход, адр)

если ош\_запр#0 : Если запрос выполнен с ошибкой, то сбрасываем флаг работы

{

флаг1\_вых=0

флаг2\_вых=0

}

}

иначе

{

(успех, код\_ош\_отв, статус, адр1, адр2, адр3)=HartAdr\_Result (Плата)

если успех=1 | код\_ош\_отв#0 : если данные подготовлены или выставлена ошибка функции чтения результатов, то сбрасываем флаг работы

{

флаг1\_вых=0

флаг2\_вых=0

}

}

Конец

**Функция HART2** : HART, команда 2: Считать ток и процент диапазона

Входные\_Переменные

Начало



```

    лог флаг1 : Работаем, пока не получены данные или ошибка выполнения
функций запроса/получения данных
    лог флаг2 : Флаг первого прохода, функцию Hart2_Run можем вызвать
один раз для каждой платы, затем вызываем функцию Hart2_Result
    Цел16 плата
    Цел16 вход
    Цел16 адр1 : Длинный адрес HART-устройства (часть 1), получено
заранене вызовом функции HART0 или HARTADR
    Цел16 адр2 : Длинный адрес HART-устройства (часть 2)
    Цел16 адр3 : Длинный адрес HART-устройства (часть 3)
    Конеч
Выходные_Переменные
Начало
    Вещ32 Ток_первич_перем
    Вещ32 Процент_диапазона
    лог флаг1_вых : Флаг работы, для передачи в основное тело ПрП
    лог флаг2_вых : Флаг первого прохода, для передачи в основное тело ПрП
    Конеч
Начало
Переменные
Начало
    Цел16 ош_запр
    Лог успех
    Цел16 код_ош_отв
    Цел16 статус
    Конеч
флаг1_вых=флаг1
флаг2_вых=флаг2
если флаг2=0
{
    флаг2_вых=1
    (ош_запр)=Hart2_Run(плата,вход,адр1, адр2,адр3)
    если ош_запр#0 : Если запрос выполнен с ошибкой, то сбрасываем флаг
работы
    {
        флаг1_вых=0
        флаг2_вых=0
    }
}
иначе
{
    (успех,код_ош_отв,статус,Ток_первич_перем,Процент_диапазона)=Hart2_Result(Пл
ата)
    если успех=1 | код_ош_отв#0 : Если данные подготовлены или
выставлена ошибка функции чтения результатов, то сбрасываем флаг работы
    {
        флаг1_вых=0
        флаг2_вых=0
    }
}
Конеч

Функция HART3 : HART, команда 3: Считать ток и значения четырех
(предопределенных) динамических переменных
Входные_Переменные

```

## ТЕХНОЛОГИЧЕСКИЕ ФУНКЦИИ

Начало

лог флаг1 : Работаем, пока не получены данные или ошибка выполнения функций запроса/получения данных

лог флаг2 : Флаг первого прохода, функцию Hart3\_Run можем вызвать один раз для каждой платы, затем вызываем функцию Hart3\_Result

Цел16 плата

Цел16 вход

Цел16 адр1 : Длинный адрес HART-устройства (часть 1), получено заранее вызовом функции HART0 или HARTADR

Цел16 адр2 : Длинный адрес HART-устройства (часть 2)

Цел16 адр3 : Длинный адрес HART-устройства (часть 3)

Конец

Выходные\_Переменные

Начало

Вещ32 ток\_первич\_перем

Цел16 код\_ед\_изм\_первич

Цел16 код\_ед\_изм\_втор

Цел16 код\_ед\_изм\_трет

Цел16 код\_ед\_изм\_четв

Вещ32 первичная\_переменная

Вещ32 вторая\_переменная

Вещ32 третья\_переменная

Вещ32 четвертая\_переменная

лог флаг1\_вых : Флаг работы, для передачи в основное тело ПрП

лог флаг2\_вых : Флаг первого прохода, для передачи в основное тело ПрП

Конец

Начало

Переменные

Начало

Цел16 ош\_запр

Лог успех

Цел16 код\_ош\_отв

Цел16 статус

Конец

флаг1\_вых=флаг1

флаг2\_вых=флаг2

если флаг2=0

{

флаг2\_вых=1

(ош\_запр)=Hart3\_Run(плата,вход,адр1, адр2,адр3)

если ош\_запр#0 : Если запрос выполнен с ошибкой, то сбрасываем флаг работы

{

флаг1\_вых=0

флаг2\_вых=0

}

}

иначе

{

(успех, код\_ош\_отв, статус, ток\_первич\_перем, код\_ед\_изм\_первич, код\_ед\_изм\_втор, код\_ед\_изм\_трет, код\_ед\_изм\_четв, первичная\_переменная, вторая\_переменная, третья\_переменная, четвертая\_переменная)=Hart3\_Result(Плата)

если успех=1 | код\_ош\_отв#0 : Если данные подготовлены или выставлена ошибка функции чтения результатов, то сбрасываем флаг работы

{

флаг1\_вых=0

```
        флаг2_вых=0
    }
Конец
```