



**TREI GMBH**

---

**УСТРОЙСТВО ПРОГРАММНОГО УПРАВЛЕНИЯ  
TREI-5B**

**ТЕХНОЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ  
В СИСТЕМЕ ISaGRAF**

**Библиотека функций,  
функциональных блоков**

---

TREI1.421457.002-02.ТП-БФ  
вторая редакция (second edition)

Information in this document is subject to change without notice and does not represent a commitment on the part of TREI GmbH. The software, which includes information contained in any databases, described in this document is furnished under a license agreement or nondisclosure agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the license or nondisclosure agreement. No part of this manual may be reproduced in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of TREI GmbH.

© 1990-2006 TREI GmbH. All rights reserved.  
Printed in Russia by TREI GmbH.  
ООО «ТРЭИ ГМБХ»  
Россия,  
440028, Пенза,  
ул. Титова, 1Г  
Телефон (fax): +7 (8412) 55-58-90, 49-95-39  
fax: +7 (8412) 49-85-13  
e-mail: [trei@trei-gmbh.ru](mailto:trei@trei-gmbh.ru)

ISaGRAF® is a registered trademark of CJ International.

All other brand or product names are trademarks or registered trademarks of their respective holders.

## СОДЕРЖАНИЕ

1.	ВВЕДЕНИЕ.....	6
2.	СОСТАВ И КЛАССИФИКАЦИЯ БИБЛИОТЕКИ .....	7
3.	БИБЛИОТЕКА РЕГУЛИРОВАНИЯ И УПРАВЛЕНИЯ .....	10
3.1	АВТОМАТИЧЕСКОЕ РЕГУЛИРОВАНИЕ .....	10
3.1.1	RAN - Регулирование аналоговое.....	10
3.1.2	RIM - Регулирование импульсное .....	12
3.1.3	SET – Управление заданием.....	14
3.1.4	LCTRL - Управление локальное.....	15
3.1.5	CCTRL - Управление каскадное .....	16
3.1.6	PWM – Широтно-импульсная модуляция .....	18
3.2	ДИНАМИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ.....	20
3.2.1	INTEG_H – Интегрирование .....	20
3.2.2	DERIV_H – Дифференцирование .....	21
3.2.3	DDERIV - Вторая производная .....	22
3.2.4	FILTER – Экспоненциальное сглаживание .....	23
3.2.5	PFILTER – Фильтрация пиков.....	24
3.2.6	DFILTER - Фильтр дискретный.....	25
3.2.7	APERT - Апертура.....	26
3.2.8	DCNV - Динамическое преобразование.....	27
3.2.9	LRATE - Ограничение скорости .....	28
3.2.10	DELAY – Запоздывание .....	29
3.2.11	AVRGM – Скользящее среднее .....	30
3.2.12	AVRGD - Дискретное среднее .....	31
3.3	СТАТИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ .....	32
3.3.1	DZONE - Зона нечувствительности .....	32
3.3.2	LIMITR - Ограничение.....	33
3.3.3	INS0 - Вставка нуля .....	34
3.3.4	MINR - Минимум .....	35
3.3.5	MAXR - Максимум .....	36
3.3.6	MAX_MIN – Максимум и минимум .....	37
3.3.7	EXTRM - Экстремум .....	38
3.3.8	ADDM - Суммирование с выделением модуля и знака .....	40
3.4	АНАЛОГО-ДИСКРЕТНЫЕ ПРЕОБРАЗОВАНИЯ .....	41
3.4.1	SELR – Переключатель на 2 входа .....	41
3.4.2	MUXD – Дискретный переключатель .....	42
3.4.3	MUXR - Переключатель по номеру .....	43
3.4.4	HYSSTER_H - Пороговый элемент по верхнему пределу .....	44
3.4.5	LIMALR_H - Гистерезис по верхнему и нижнему пределам .....	45
3.4.6	DCHNG - Запрет изменения .....	46
3.4.7	DSIGN - Запрет знака .....	47
3.4.8	FLW_HOLD - Слежение-запоминание.....	48
3.4.9	LATCH - Запоминание .....	49
3.5	ДИСКРЕТНОЕ УПРАВЛЕНИЕ.....	50
3.5.1	MAJOR - Мажорирование .....	50
3.5.2	RS_H – RS-триггер.....	51
3.5.3	SR_H – SR-триггер.....	52
3.5.4	DTRIG – D-триггер .....	53
3.5.5	RF_TRIG – R-триггер .....	54
3.5.6	TON_H – Таймер по фронту .....	55
3.5.7	TOF_H – Таймер по спаду.....	56
3.5.8	TP_H – Таймер по уставке .....	57
3.5.9	CTD_H – Счетчик вниз .....	58
3.5.10	CTU_H – Счетчик вверх .....	59
3.5.11	CTUD_H – Счетчик вверх-вниз .....	60
3.6	ПРОГРАММНОЕ УПРАВЛЕНИЕ .....	61
3.6.1	PULSE – Одновибратор .....	61
3.6.2	BLINKE – Мультивибратор .....	62
3.6.3	SIN_GEN – Генератор синусоидального сигнала .....	63
3.6.4	PSET - Программный задатчик .....	64

3.6.5	<i>SECTION</i> - Кусочно-линейная функция.....	66
3.7	ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ .....	67
3.7.1	<i>Регулирование с каскадным и локальным управлением</i> .....	67
3.7.2	<i>Каскадное регулирование с локальным заданием</i> .....	71
<b>4.</b>	<b>БИБЛИОТЕКА ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ</b> .....	<b>73</b>
4.1	ОПЕРАЦИИ С МАССИВАМИ.....	73
4.1.1	<i>AR_OPEN</i> - Создание массива элементов .....	73
4.1.2	<i>AR_CLOSE</i> - Закрытие массива .....	74
4.1.3	<i>AR_FILL</i> - Заполнение массива данными из файла.....	75
4.1.4	<i>AR_SAVE</i> - Сохранение массива данных в файле.....	76
4.1.5	<i>AR_WR_</i> - Запись элемента массива .....	77
4.1.6	<i>AR_RD_</i> - Чтение элемента массива .....	78
4.1.7	<i>ER_TAKE</i> - Получение кода ошибки последней операции .....	79
4.1.8	<i>INTERP2</i> - Интерполяция .....	80
4.1.9	<i>INDEX</i> – Поиск индекса.....	81
4.2	ФАЙЛОВЫЕ ОПЕРАЦИИ.....	82
4.2.1	<i>FL_OPEN</i> - Открытие файла .....	82
4.2.2	<i>FL_CLOSE</i> - Закрытие файла .....	83
4.2.3	<i>FL_SPOS</i> - Установка текущей позиции в файле.....	84
4.2.4	<i>FL_GPOS</i> - Получение текущей позиции в файле.....	85
4.2.5	<i>FL_WR_</i> - Запись в файл .....	86
4.2.6	<i>FL_RD_</i> - Чтение из файла .....	87
<b>5.</b>	<b>ФУНКЦИИ ВРЕМЕНИ</b> .....	<b>88</b>
5.1	УСТАНОВКА ВРЕМЕНИ.....	88
5.1.1	<i>SETTIME_</i> - Установка текущего времени .....	90
5.1.2	<i>SETTIME</i> - Установка текущего времени (специальный формат) .....	91
5.1.3	<i>TM_SYNC</i> - Синхронизация времени по ИВЧ-1.....	92
5.2	ПОЛУЧЕНИЕ ВРЕМЕНИ.....	93
5.2.1	<i>GETTIME_</i> - Получение текущего времени .....	93
5.2.2	<i>GETTIME</i> - Получение текущего времени (специальный формат) .....	94
5.2.3	<i>T5B_TIME</i> – Получение временных параметров ISaGRAF.....	95
<b>6.</b>	<b>ФУНКЦИИ ПРЕОБРАЗОВАНИЯ</b> .....	<b>96</b>
6.1	ПРЕОБРАЗОВАНИЯ ДЛЯ ТЕМПЕРАТУРЫ .....	96
6.1.1	<i>TERMOENV</i> – Преобразование физической величины в температуру.....	97
6.1.2	<i>TERMOINV</i> – Преобразование температуры в физическую величину.....	98
6.2	ПРЕОБРАЗОВАНИЯ ДЛЯ ПРИВЕДЕННОГО КОДА .....	99
6.2.1	<i>VALUEENV</i> – Преобразование приведенного кода в физическую величину.....	100
6.2.2	<i>VALUEINV</i> – Преобразование физической величины в приведенный код .....	101
6.3	ПРЕОБРАЗОВАНИЯ ДЛЯ ВРЕМЕНИ .....	102
6.3.1	<i>TIME_CNV</i> – Преобразование времени в строку символов .....	102
6.3.2	<i>TIME_SEC</i> – Преобразование времени в формат секунд с 1970г .....	103
6.4	ЛОГИЧЕСКИЕ ОПЕРАЦИИ.....	104
6.4.1	<i>GETBYTE</i> - Выделение байта в переменной.....	104
6.4.2	<i>SETBYTE</i> - Установка байта в переменной .....	105
6.4.3	<i>GETBIT</i> - Выделение бита в переменной .....	106
6.4.4	<i>SETBIT</i> - Установка бита в переменной .....	107
<b>7.</b>	<b>СПЕЦИАЛЬНЫЕ РАССЧЕТНЫЕ ФУНКЦИИ</b> .....	<b>108</b>
7.1	УЧЕТ ТЕПЛА И ГАЗА.....	108
7.1.1	<i>AGA8_92</i> – Расчет свойств природного газа (метод УС АГА8-92DC).....	108
7.1.2	<i>GERG_91</i> – Расчет свойств природного газа (метод GERG-91 мод.) .....	109
7.1.3	<i>NX_19</i> – Расчет свойств природного газа (метод NX_19) .....	110
7.1.4	<i>VNIC_SMV</i> – Расчет свойств природного газа (метод ВНИЦ СМВ).....	111
7.1.5	<i>DS_CALC</i> – Расчет плотности природного газа .....	112
7.1.6	<i>FLOW_R</i> – Расчет расхода природного газа .....	113
<b>8.</b>	<b>MODBUS ФУНКЦИИ</b> .....	<b>114</b>
8.1	MODBUS ФУНКЦИИ ЧТЕНИЯ .....	114
8.1.1	<i>MB_R_C</i> - Чтение состояния 8-ми бинарных ячеек памяти .....	114
8.1.2	<i>MB_R_V</i> - Чтение состояния 8-ми дискретных входов .....	115

8.1.3	<i>MB_R_H</i> - Чтение целочисленного регистра .....	116
8.1.4	<i>MB_R_I</i> - Чтение целочисленного входного регистра .....	117
8.1.5	<i>MB_R_F</i> - Чтение пары регистров в формате с плавающей точкой .....	118
8.1.6	<i>MB_R_R16</i> - Чтение 16 целочисленных регистров .....	119
8.1.7	<i>MB_R_F16</i> - Чтение 16 пар регистров в формате с плавающей точкой .....	120
8.2	MODBUS ФУНКЦИИ ЗАПИСИ .....	121
8.2.1	<i>MB_W_C</i> - Запись в бинарную ячейку памяти .....	121
8.2.2	<i>MB_W_H</i> - Запись целочисленного регистра .....	122
8.2.3	<i>MB_W_F</i> - Запись пары регистров в формате с плавающей точкой .....	123
8.3	MODBUS ФУНКЦИИ ДИАГНОСТИКИ .....	124
8.3.1	<i>MB_DIAG</i> - Диагностика связи с подчиненным устройством .....	124
8.3.2	<i>MB_STATE</i> - Получение ID подчиненного устройства .....	125
8.3.3	<i>MB_PARAM</i> - Установка параметров задачи связи <i>modbus</i> .....	126
<b>9.</b>	<b>БИБЛИОТЕКА ФУНКЦИЙ HART</b> .....	<b>127</b>
9.1	HART ФУНКЦИИ ЧТЕНИЯ ДАННЫХ .....	127
9.1.1	<i>HART_1</i> - Считать первичную переменную .....	127
9.1.2	<i>HART_2</i> - Считать ток и процент диапазона .....	128
9.1.3	<i>HART_3</i> - Считать значения четырех динамических переменных .....	129
9.2	HART ФУНКЦИИ ЧТЕНИЯ/ЗАПИСИ ДИАПАЗОНА .....	130
9.2.1	<i>HART_15</i> - Считать информацию о выходе первичной переменной .....	130
9.2.2	<i>HART_35</i> - Записать границы диапазона первичной переменной .....	131
9.3	HART ФУНКЦИИ ДИАГНОСТИКИ .....	132
9.3.1	<i>HART_0</i> - Читать уникальный идентификатор .....	132
9.3.2	<i>HART_41</i> - Выполнить самотестирование устройства .....	133
9.3.3	<i>HART_48</i> - Считать дополнительный статус устройства .....	134
9.4	КОДЫ ОТКЛИКА УСТРОЙСТВА (СТАТУС КОМАНДЫ) .....	135

## 1. ВВЕДЕНИЕ

Устройство программного управления **TREI-5B** (далее по тексту **КОНТРОЛЛЕР**) предназначено для локальных и распределенных систем автоматического контроля и управления технологическими процессами на промышленных предприятиях с нормальным и взрывоопасным производством.

Контроллер принимает, обрабатывает и нормирует дискретные, аналоговые и частотно-импульсные входные сигналы с первичных преобразователей и приборов, выполняет операции управляющей логики, осуществляет сложную обработку аналоговых сигналов, в том числе фильтрацию, интегрирование, ПИ- и ПИД- регулирование, а также формирует и выдает управляющие воздействия на объект управления.

TREI-5B это программируемый контроллер, работать с которым может проектировщик системы управления и эксплуатационный персонал, связанный с обслуживанием традиционной релейной и аналоговой аппаратуры и не знакомый с вычислительной техникой и методами программирования для ЭВМ.

Разработка, ввод в контроллер технологической программы контроля и управления объектом и отладка этой программы осуществляется с помощью инструментальной системы ISaGRAF и языков программирования PLC стандарта IEC 1131-3.

Документ предназначен для проектировщиков и эксплуатационного персонала АСУ ТП.

При работе используйте документацию следующих фирм:

### **TREI GmbH**

- «Устройство программного управления TREI-5B-02 Руководство по эксплуатации»
- «Устройство программного управления TREI-5B-00 Руководство по эксплуатации»
- «Устройство программного управления TREI-5B Программа метрологической поверки»
- «Модули измерительные TREI-5B-M Руководство по эксплуатации»
- «Модули измерительные TREI-5B-M Методика поверки»
- «Устройство программного управления TREI-5B Технологическое программирование в системе ISaGRAF Руководство пользователя»

### **CJ International**

- «ISaGRAF версия 3.3 Руководство пользователя» Перевод на русский язык Науцилус. (электронная версия документа поставляется в составе дистрибутива ISaGRAF).

## 2. СОСТАВ И КЛАССИФИКАЦИЯ БИБЛИОТЕКИ

Библиотека разработана с помощью системы программирования функций и функциональных блоков "С" и встроена в целевую задачу ISaGRAF для контроллера TREI-5B. Библиотека содержит набор алгоритмов, необходимых для решения сложных задач автоматического регулирования, логико-программного управления, статического, математического, логического и аналого-дискретного преобразования сигналов.

Внутреннее состояние функциональных блоков сохраняется в энергонезависимом ОЗУ контроллера с целью восстановления состояния технологической программы при повторном перезапуске.

Типы входов и выходов функций и функциональных блоков соответствуют стандартным типам переменных языков программирования ISaGRAF.

Состав и классификация библиотеки приведены в таблице.

Код	Наименование	Аналог ISaGRAF	Сохранение / Восстановление.
<b>БИБЛИОТЕКА РЕГУЛИРОВАНИЯ И УПРАВЛЕНИЯ</b>			
<i>Автоматическое регулирование</i>			
RAN	Регулирование аналоговое		√
RIM	Регулирование импульсное		√
SET	Управление заданием		√
LCTRL	Управление локальное		√
CCTRL	Управление каскадное		√
PWM	Широтно-импульсная модуляция		√
<i>Динамические преобразования</i>			
INTEG_H	Интегрирование	INTEGRAL	√
DERIV_H	Дифференцирование	DERIVATE	√
DDERIV	Вторая производная		√
FILTER	Экспоненциальное сглаживание		√
PFILTER	Фильтр высокочастотной помехи		√
DFILTER	Фильтр дискретный		√
APERT	Апертура		√
DCNV	Динамическое преобразование		√
LRATE	Ограничение скорости		√
DELAY	Запаздывание		√
AVRGM	Скользящее среднее	AVERAGE	√
AVRGD	Дискретное среднее		√
<i>Статические преобразования</i>			
DZONE	Зона нечувствительности		
LIMITR	Ограничение		
INS0	Вставка нуля		√
MINR	Минимум		
MAXR	Максимум		
MAX_MIN	Максимум и минимум		
EXTRM	Экстремум		√
ADDM	Суммирование с выделением модуля и знака		
<i>Аналого-дискретные преобразования</i>			
SELR	Переключатель на 2 входа	SEL	
MUXD	Дискретный переключатель		
MUXR	Переключатель по номеру	MUX	
HYSTER_H	Пороговый элемент по верхнему пределу	HYSTER	√
LIMALR_H	Гистерезис по верхнему и нижнему пределам	LIM_ALARM	√
DCHNG	Запрет изменения		√
DSIGN	Запрет знака		
FLW_HOLD	Слежение-запоминание		√
LATCH	Запоминание		√
<i>Дискретное управление</i>			
MAJOR	Мажорирование		
RS_H	RS-триггер	RS	√

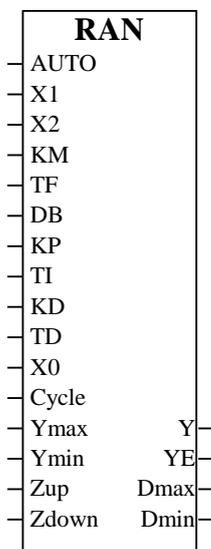
SR_H	SR-триггер	SR	√
DTRIG	D триггер		√
RF_TRIG	R/F-триггер	R_TRIG	√
TON_H	Таймер по фронту	TON	√
TOF_H	Таймер по спаду	TOF	√
TP_H	Таймер по уставке	TP	√
CTD_H	Счетчик вниз	CTD	√
CTU_H	Счетчик вниз	CTU	√
CTUD_H	Счетчик вверх-вниз	CTUD	√
<b>Программное управление</b>			
PULSE	Одновибратор		√
BLINKE	Мультивибратор	BLINK	√
SIN_GEN	Генератор синусоидального сигнала		√
PSET	Программный задатчик		√
SECTION	Кусочно-линейная функция		
<b>БИБЛИОТЕКА ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ</b>			
<b>Операции с массивами</b>			
AR_OPEN	Создание массива элементов	ARCREATE	
AR_CLOSE	Закрытие массива		
AR_FILL	Заполнение массива данными из файла		
AR_SAVE	Сохранение массива данных в файле		
AR_WR_A	Запись элемента массива целых чисел	ARWRITE	
AR_WR_R	Запись элемента массива вещественных чисел		
AR_WR_M	Запись элемента массива сообщений		
AR_RD_A	Чтение элемента массива целых чисел	ARREAD	
AR_RD_R	Чтение элемента массива вещественных чисел		
AR_RD_M	Чтение элемента массива сообщений		
ER_TAKE	Получение кода ошибки последней операции		
INTERP2	Интерполяция		
INDEX	Поиск индекса		
<b>Файловые операции</b>			
FL_OPEN	Открытие файла	F_ROPEN	
FL_CLOSE	Закрытие файла	F_CLOSE	
FL_SPOS	Установка текущей позиции в файле		
FL_GPOS	Получение текущей позиции в файле		
FL_WR_A	Запись в файл переменной целого типа	FA_WRITE	
FL_WR_R	Запись в файл переменной вещественного типа		
FL_WR_M	Запись в файл переменной типа сообщение	FM_WRITE	
FL_RD_A	Чтение из файла переменной целого типа	FA_READ	
FL_RD_R	Чтение из файла переменной вещественного типа		
FL_RD_M	Чтение из файла переменной типа сообщение	FM_READ	
<b>ФУНКЦИИ ВРЕМЕНИ</b>			
<b>Установка времени</b>			
SETTIME_	Установка текущего времени		
SETTIME	Установка текущего времени (специальный формат)		
TM_SYNC	Синхронизация времени по ИВЧ-1		
<b>Получение времени</b>			
GETTIME_	Получение текущего времени		
GETTIME	Получение текущего времени (специальный формат)		
T5B_TIME	Получение временных параметров ISaGRAF		
<b>ФУНКЦИИ ПРЕОБРАЗОВАНИЯ</b>			
<b>Преобразования для температуры</b>			
TERMOCNV	Преобразование физической величины в температуру		
TERMOINV	Преобразование температуры в физическую величину		
<b>Преобразования для приведенного кода</b>			
VALUECNV	Преобразование кода в физическую величину		
VALUEINV	Преобразование физической величины в код		

<b>Преобразования для времени</b>			
TIME_CNV	Преобразование времени в строку символов		
TIME_SEC	Преобразование времени в формат секунд с 1970г		
<b>Логические операции</b>			
GETBYTE	Выделение байта в переменной		
SETBYTE	Установка байта в переменной		
GETBIT	Выделение бита в переменной		
SETBIT	Установка бита в переменной		
<b>СПЕЦИАЛЬНЫЕ РАССЧЕТНЫЕ ФУНКЦИИ</b>			
<b>Учет тепла и газа</b>			
AGA8_92	Расчет свойств природного газа (метод UC AGA8_92DC)		
GERG_91	Расчет свойств природного газа (метод GERG_91)		
NX_19	Расчет свойств природного газа (метод NX_19)		
VNIC_SMV	Расчет свойств природного газа (метод ВНИЦ СМВ)		
DS_CALC	Расчет плотности природного газа		
FLOW_R	Расчет расхода природного газа		
<b>MODBUS ФУНКЦИИ</b>			
<b>Modbus функции чтения</b>			
MB_R_C	Чтение состояния 8-ми бинарных ячеек памяти		
MB_R_B	Чтение состояния 8-ми дискретных входов		
MB_R_H	Чтение целочисленного регистра		
MB_R_I	Чтение целочисленного входного регистра		
MB_R_F	Чтение пары регистров в формате с плавающей точкой		
MB_R_R16	Чтение 16 целочисленных регистров		
MB_R_F16	Чтение 16 пар регистров в формате с плавающей точкой		
<b>Modbus функции записи</b>			
MB_W_C	Запись в бинарную ячейку памяти		
MB_W_H	Запись целочисленного регистра		
MB_W_F	Запись пары регистров в формате с плавающей точкой		
<b>Modbus функции диагностики</b>			
MB_DIAG	Диагностика связи с подчиненным устройством		
MB_STATE	Получение ID подчиненного устройства		
MB_PARAM	Установка параметров задачи связи modbus		
<b>HART ФУНКЦИИ</b>			
<b>HART функции чтения данных</b>			
HART_1	Считать первичную переменную		
HART_2	Считать ток и процент диапазона		
HART_3	Считать значения четырех динамических переменных		
<b>HART функции чтения/записи диапазона</b>			
HART_15	Считать информацию о выходе первичной переменной		
HART_35	Записать границы диапазона первичной переменной		
<b>HART функции диагностики</b>			
HART_0	Читать уникальный идентификатор устройства		
HART_41	Выполнить самотестирование устройства		
HART_48	Считать дополнительный статус устройства		

### 3. БИБЛИОТЕКА РЕГУЛИРОВАНИЯ И УПРАВЛЕНИЯ

#### 3.1 АВТОМАТИЧЕСКОЕ РЕГУЛИРОВАНИЕ

##### 3.1.1 RAN - Регулирование аналоговое



Вх/Вых	Обозначение	Тип	Назначение
Вход	AUTO	Boolean	Режим работы: TRUE – автоматический, FALSE - ручной
	X1	Real	Не масштабируемый вход (задание)
	X2	Real	Масштабируемый вход (регулируемый параметр)
	KM	Real	Масштабный коэффициент
	TF	Real	Постоянная времени фильтра (сек.)
	DB	Real	Зона нечувствительности
	KP	Real	Коэффициент пропорциональности
	TI	Real	Постоянная времени интегрирования (сек.)
	KD	Real	Коэффициент дифференцирования
	TD	Real	Постоянная времени дифференцирования (сек.)
	X0	Real	Значение выхода в ручном режиме
	Cycle	Integer	Период работы блока (мсек.)
	Ymax	Real	Ограничение по максимуму
	Ymin	Real	Ограничение по минимуму
	Zup	Boolean	Сигнал запрета в направлении “Больше”
	Zdown	Boolean	Сигнал запрета в направлении “Меньше”
Выход	Y	Real	Основной выход алгоритма (регулирующее воздействие)
	YE	Real	Сигнал рассогласования
	Dmax	Boolean	Признак ограничения по максимуму
	Dmin	Boolean	Признак ограничения по минимуму

#### Назначение

Функциональный блок RAN используется для построения системы ПИД регулирования, имеющей аналоговый выход. Блок, как правило, сочетается с пропорциональным исполнительным механизмом (позиционером), либо используется в качестве ведущего в схеме каскадного регулирования. Помимо формирования ПИД закона в алгоритме вычисляется сигнал рассогласования. Выходной сигнал ограничивается по максимуму и минимуму.

#### Описание алгоритма

##### *Автоматический режим работы регулятора*

В автоматическом режиме работы (AUTO=TRUE) функциональный блок RAN формирует регулирующее воздействие по ПИД закону регулирования. Преобразования реализуются в следующей последовательности. Входной сигнал X2 (регулируемый параметр) фильтруется (т.е. проходит через апериодическое звено 1-го порядка) в соответствии с передаточной функцией:

$$W(p) = \frac{1}{TF \cdot p + 1},$$

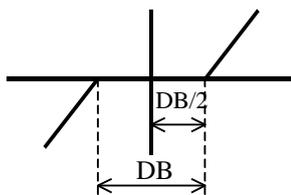
где TF – постоянная времени фильтра.

Отфильтрованное значение  $X_{2f}$  масштабируется (умножается на коэффициент KM). От значения задания  $X_1$  вычитается получившееся после фильтрации и масштабирования значение регулируемого параметра, в результате чего определяется значение рассогласования:  $X_e = X_1 - KM \cdot X_{2f}$ . Далее сигнал рассогласования проходит через зону нечувствительности в соответствии со следующим законом:

$$\begin{aligned} X_z &= 0, \text{ если } |X_e| \leq DB/2 \\ X_z &= X_e - DB/2, \text{ если } X_e > DB/2 \\ X_z &= X_e + DB/2, \text{ если } X_e < -DB/2 \end{aligned}$$

где DB – заданная зона нечувствительности.

Прохождение сигнала через зону нечувствительности поясняется следующим рисунком.



Сигнал с выхода зоны нечувствительности  $X_z$  преобразуется по ПИД закону регулирования в соответствии с передаточной функцией:

$$W(p) = KP + \frac{KP}{TI \cdot p} + \frac{KD \cdot TD \cdot p}{TD \cdot p + 1},$$

где KP – коэффициент пропорциональности, TI – постоянная интегрирования, KD – коэффициент дифференцирования, TD – постоянная дифференцирования.

Блок RAN может использоваться в качестве ПД-, ПИ- или П-регулятора. В ПД-регуляторе устанавливается  $TI=0$ , при этом интегральная часть обнуляется ( $KP/TI \cdot p=0$ ). В ПИ-регуляторе устанавливается  $KD=0$ , в этом случае приравняется к нулю дифференциальная часть регулятора. В П-регуляторе устанавливается  $TI=0$  и  $KD=0$ .

Интервал между расчетами регулирующего воздействия не меньше параметра Cycle и кратен времени цикла контроллера.

В алгоритме предусмотрено ограничение выходного сигнала  $Y$  по нижнему и верхнему пределам. Верхний и нижний пределы задаются входами  $Y_{max}$  и  $Y_{min}$ .

Также предусмотрены дискретные входы  $Z_{up}$  (запрет больше) и  $Z_{down}$  (запрет меньше). При  $Z_{up}=TRUE$  значения выхода  $Y$  и интегральной составляющей не увеличиваются, при  $Z_{down}=TRUE$  не уменьшаются.

Блок имеет 4 выхода. Выход  $Y$  – основной выход алгоритма (регулирующее воздействие). На выходе  $Y_E$  формируется сигнал рассогласования, т.е.  $Y_E = X_e$ . Два дискретных выхода  $D_{max}$  и  $D_{min}$  фиксируют момент наступления ограничения выходного сигнала  $Y$ . Логика формирования выходных дискретных сигналов определяется следующей таблицей (здесь  $Y_1$  – сигнал на входе звена ограничения):

$Y_1$	$Y$	$D_{max}$	$D_{min}$
$Y_{min} < Y_1 < Y_{max}$	$Y = Y_1$	0	0
$Y_1 \geq Y_{max}$	$Y = Y_{max}$	1	0
$Y_1 \leq Y_{min}$	$Y = Y_{min}$	0	1

Алгоритм будет правильно работать, только если  $Y_{max} > Y_{min}$ . Если действует ограничение по максимуму ( $D_{max}=TRUE$ ), значение интегральной части регулятора не увеличивается. Если действует ограничение по минимуму ( $D_{min}=TRUE$ ), значение интеграла не уменьшается.

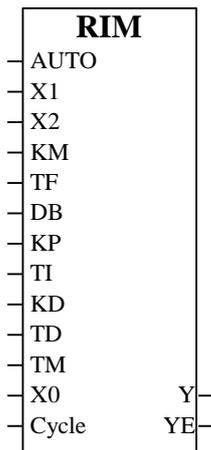
#### Ручной режим работы регулятора

При ручном режиме работы ( $AUTO=FALSE$ ) основной выход блока равен входу  $X_0$ , т.е.  $Y=X_0$ . Входной сигнал  $X_2$  при ручном режиме работы не фильтруется, поэтому выход  $Y_E = X_e = X_1 - KM \cdot X_2$ . Ограничения и запреты на выход  $Y$  действуют также как и при автоматическом режиме работы.

В момент переключения регулятора на автоматический режим работы (переход входного сигнала AUTO из состояния FALSE в состояние TRUE) интегральная часть регулятора приводится к значению  $IV = X_0 - KP \cdot X_z$ , дифференциальная часть регулятора обнуляется (т.е. производная приравняется к нулю), значение выхода  $Y$  не изменяется. Таким образом, переход на автоматический режим выполняется безударно.

Примечания: При значении  $Cycle \leq 0$  период работы блока равен циклу контроллера. Отрицательные значения параметров KP, TI, KD, TD воспринимаются как нулевые значения. При значении  $TF \leq 0$  входной сигнал  $X_2$  не фильтруется. При значении  $DB \leq 0$   $X_z = X_e$ . Параметры настройки регулятора (DB, KP, TI, KD, TD) рекомендуется изменять при ручном режиме работы регулятора.

3.1.2 RIM - Регулирование импульсное



Вх/Вых	Обозначение	Тип	Назначение
Вход	AUTO	Boolean	Режим работы: TRUE – автоматический, FALSE - ручной
	X1	Real	Не масштабируемый вход (задание)
	X2	Real	Масштабируемый вход (регулируемый параметр)
	KM	Real	Масштабный коэффициент
	TF	Real	Постоянная времени фильтра (сек.)
	DB	Real	Зона нечувствительности
	KP	Real	Коэффициент пропорциональности
	TI	Real	Постоянная времени интегрирования (сек.)
	KD	Real	Коэффициент дифференцирования
	TD	Real	Постоянная времени дифференцирования (сек.)
	TM	Real	Полное время хода ИМ (сек.)
	X0	Real	Значение выхода регулятора в ручном режиме
	Cycle	Integer	Период работы блока (мсек.)
Выход	Y	Real	Основной выход алгоритма
	YE	Real	Сигнал рассогласования

**Назначение**

Функциональный блок RIM используется при построении системы ПИД регулирования, работающей совместно с исполнительным механизмом постоянной скорости. Блок, как правило, применяется в сочетании с алгоритмом широтно-импульсной модуляции (PWM), который преобразует выходной аналоговый сигнал алгоритма RIM в последовательность импульсов для дискретного или импульсного вывода. Помимо формирования закона регулирования в алгоритме вычисляется сигнал рассогласования.

**Описание алгоритма**

*Автоматический режим работы регулятора*

В автоматическом режиме работы (AUTO=TRUE) функциональный блок RIM формирует регулирующее воздействие по ПДД<sup>2</sup> закону регулирования. Преобразования входных сигналов в блоке RIM аналогичны преобразованиям в блоке RAN, т.е. входной сигнал X2 (регулируемый параметр) фильтруется и умножается на коэффициент масштабирования KM, получившееся значение вычитается из задания X1, в результате чего определяется сигнал рассогласования Xe, который проходит через зону нечувствительности. Сигнал с выхода зоны нечувствительности Xz преобразуется по ПДД<sup>2</sup> закону в соответствии с передаточной функцией:

$$W(p) = TM \cdot \left( KP \cdot p + \frac{KP}{TI} + \frac{KD \cdot TD \cdot p^2}{(TD \cdot p + 1)^2} \right),$$

что позволяет совместно с исполнительным механизмом (ИМ) постоянной скорости имеющим передаточную функцию  $W_{им}(p) = 1/(TM \cdot p)$  приблизительно реализовать ПИД преобразование сигнала Xz в соответствии с передаточной функцией:

$$W(p) = KP + \frac{KP}{TI \cdot p} + \frac{KD \cdot TD \cdot p}{(TD \cdot p + 1)^2},$$

где  $T_M$  – полное время хода ИМ,  $K_P$  – коэффициент пропорциональности,  $T_I$  – постоянная интегрирования,  $K_D$  – коэффициент дифференцирования,  $T_D$  – постоянная дифференцирования.

Интервал между расчетами регулирующего воздействия не менее параметра  $Cycle$  и кратен времени цикла контроллера.

Блок имеет 2 выхода. Выход  $Y$  - основной выход алгоритма (регулирующее воздействие). На выходе  $YE$  формируется сигнал рассогласования, т.е.  $YE=Xe$ .

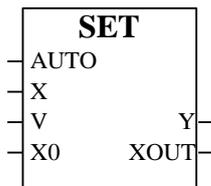
Ручной режим работы регулятора

При ручном режиме работы ( $AUTO=FALSE$ ) выход блока  $Y=X0$ . Входной сигнал  $X2$  не фильтруется, поэтому  $YE=Xe=X1-KM \cdot X2$ .

В момент переключения регулятора на автоматический режим работы (переход входного сигнала  $AUTO$  из состояния  $FALSE$  в состояние  $TRUE$ ) первая и вторая производная приравниваются к нулю, выход регулятора равен нулю. Таким образом, переход на автоматический режим выполняется безударно.

Примечания: При значении  $Cycle \leq 0$  период работы блока равен циклу контроллера. Отрицательные значения параметров  $K_P$ ,  $K_D$ ,  $T_D$  воспринимаются как нулевые. При значении  $T_I \leq 0$   $T_I=1$ . При значении  $T_F \leq 0$  входной сигнал  $X2$  не фильтруется. При значении  $DB \leq 0$   $Xz=Xe$ . Параметры настройки регулятора ( $DB$ ,  $K_P$ ,  $T_I$ ,  $K_D$ ,  $T_D$ ) рекомендуется изменять при ручном режиме работы регулятора.

3.1.3 SET – Управление заданием



Вх/Вых	Обозначение	Тип	Назначение
Вход	AUTO	Boolean	Режим работы: TRUE-автоматический, FALSE-ручной
	X	Real	Задание
	V	Real	Скорость динамической балансировки (1/сек.)
	X0	Real	Значение выхода блока в ручном режиме
Выход	Y	Real	Задание для регулятора с динамической балансировкой
	XOUT	Real	Выход без динамической балансировки

Назначение

Функциональный блок SET предназначен для формирования задания для регулятора.

Описание алгоритма

В автоматическом режиме работы блока (AUTO=TRUE) входной сигнал X проходит через звено динамической балансировки. Для этого определяется скорость изменения значения входа X относительно выхода:

$$V1 = \frac{X - Y_{i-1}}{Tc},$$

где  $Y_{i-1}$  – значение выхода в предыдущем цикле,  $Tc$  – время цикла контроллера.

Если  $|V1| \leq V$ , то сигнал X проходит на выход без изменений. Если  $|V1| > V$ , то выход Y будет изменяться в соответствии со следующей зависимостью:

$$Y_i = Y_{i-1} + sign(V1) \cdot V \cdot Tc,$$

где  $Y_i$  – значение выхода в текущем цикле,  $V$  – скорость динамической балансировки.

Выход Y будет изменяться с ограниченной скоростью V до тех пор, пока не достигнет значения X.

При отключенной динамической балансировке ( $V \leq 0$ ) входной сигнал X проходит на выход Y без изменений. На выход XOUT в автоматическом режиме работы проходит без изменений сигнал с входа X, минуя звено динамической балансировки. В ручном режиме работы (AUTO=FALSE) выходы блока равны входу X0.

На Рисунок 3.1 приведен пример соединения блоков SET и RAN. Сигнал с выхода Y блока SET подается на вход X1 (задание) регулятора. На вход X0 блока SET подается тот же сигнал, что и на вход X2 регулятора (регулируемый параметр).

При ручном режиме работы данной схемы ввод нового значения задания (ustavca) игнорируется, т.е. задание отслеживает регулируемый параметр. При переключении на автоматический режим, задание перестает отслеживать регулируемый параметр и равно последнему значению регулируемого параметра в ручном режиме, изменить значение задания можно в следующем цикле работы контроллера. Если выход XOUT не задействовать (т.е. не подключать переменную ustavca к выходу XOUT), значение задания можно изменять при любом режиме работы.

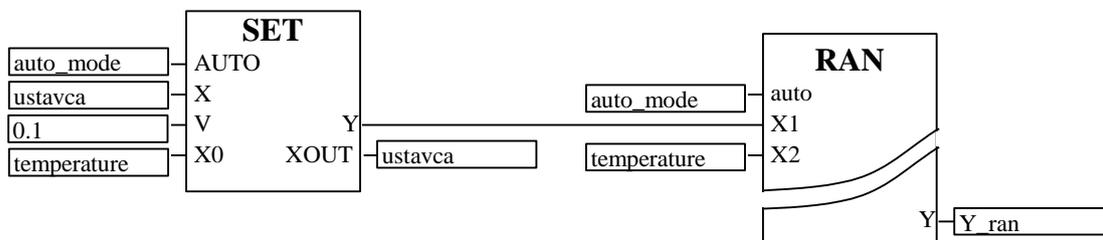
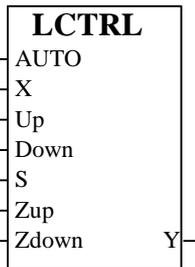


Рисунок 3.1 Пример соединения блоков SET и RAN.

## 3.1.4 LCTRL - Управление локальное



Вх/Вых	Обозначение	Тип	Назначение
Вход	AUTO	Boolean	Режим работы: TRUE-автоматический, FALSE-ручной
	X	Real	Значение выхода в автоматическом режиме
	Up	Boolean	Увеличить управляющее воздействие
	Down	Boolean	Уменьшить управляющее воздействие
	S	Real	Приращение выхода в ручном режиме
	Zup	Boolean	Сигнал запрета в направлении "Больше"
	Zdown	Boolean	Сигнал запрета в направлении "Меньше"
Выход	Y	Real	Управляющее воздействие

**Назначение**

Функциональный блок LCTRL используется в составе каскадного регулятора. Он необходим, если предусматриваться ручное управление импульсным регулятором RIM.

**Описание алгоритма**

При автоматическом режиме работы AUTO=TRUE выход блока Y равен входу X. При ручном режиме выход Y (управляющее воздействие) увеличивается по абсолютному значению на величину S в каждом цикле контроллера в соответствии со следующей зависимостью:

$$Y_i = Y_{i-1} + S,$$

где  $Y_i$  – значение Y в текущем цикле,  $Y_{i-1}$  – значение Y в предыдущем цикле, S – величина приращения.

Знак управляющего воздействия Y зависит от состояния дискретных входов Up и Down. Если Up=TRUE управляющее воздействие Y положительно, если Down=TRUE управляющее воздействие Y отрицательно. В ситуации, когда Up=Down (оба входа равны логическому 0 или логической 1) выход Y обнуляется. Если Zup=TRUE и Up=TRUE или Zdown=TRUE и Down=TRUE, то управляющее воздействие Y=0.

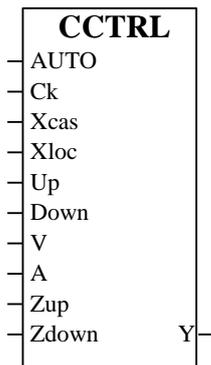
Сигнал с выхода блока Y подается на вход X0 импульсного регулятора RIM. На вход X подается сигнал с выхода RIM. Следовательно, в автоматическом режиме выход блока равен выходу регулятора.

В момент переключения на ручной режим управляющее воздействие Y равно 0, следовательно, переход на ручной режим выполняется безударно. Поскольку сигнал с выхода RIM поступает на вход блока широтно-импульсной модуляции PWM, увеличение абсолютного значения управляющего воздействия Y приведет к увеличению длительности импульсов. Таким образом, длительность импульсов будет увеличиваться в каждом цикле до тех пор, пока временное расстояние между ними больше минимально допустимой паузы Tmpс, установленной для блока PWM. Как только расстояние между импульсами станет меньше, чем Tmpс, импульсы слипнутся и выход, соответствующий знаку управляющего воздействия Y будет постоянно включен. Направление перемещения ИМ будет зависеть от состояния входных сигналов Up и Down блока LCTRL. При состоянии входа Up=TRUE ИМ перемещается по направлению "больше", если Down=TRUE по направлению "меньше". Если Up=Down ИМ не перемещается.

Дискретные входы Zup и Zdown блока LCTRL предназначены для запрещения перемещения ИМ по одному из направлений. Если Zup=TRUE или Zdown=TRUE ИМ не будет перемещаться по направлению "больше" или "меньше" соответственно.

Примечание: значение  $S \leq 0$  воспринимается алгоритмом как  $S=0$ .

3.1.5 CCTRL - Управление каскадное



Вх/Вых	Обозначение	Тип	Назначение
Вход	AUTO	Boolean	Режим работы: TRUE – автоматический, FALSE – ручной
	Ck	Boolean	Ручной режим работы: TRUE – каскадный, FALSE – локальный
	Xcas	Real	Значение выхода блока в автоматическом режиме
	Xloc	Real	Значение выхода блока в ручном локальном режиме
	Up	Boolean	TRUE – увеличение значение выхода в ручном каскадном режиме
	Down	Boolean	TUE – уменьшение значения выхода в ручном каскадном режиме
	V	Real	Скорость изменения выхода в ручном каскадном режиме (1/сек)
	A	Real	Ускорение выходного сигнала в ручном каскадном режиме (1/сек <sup>2</sup> )
	Zup	Boolean	Сигнал запрета в направлении “Больше”
	Zdown	Boolean	Сигнал запрета в направлении “Меньше”
Выход	Y	Real	Основной выход

**Назначение**

Функциональный блок CCTRL используется в составе каскадного регулятора, если необходимо ручное управление аналоговым регулятором.

**Описание алгоритма.**

Блок работает в одном из трех режимов: автоматическом, ручном каскадном и ручном локальном. Автоматический режим определяется состоянием входа AUTO=TRUE, при этом состояние входа Ck игнорируется. Ручной каскадный режим определяется состоянием входов AUTO=FALSE и Ck=TRUE. Ручной локальный режим определяется состоянием входов AUTO=FALSE и Ck=FALSE.

Выход блока Y в автоматическом режиме (AUTO=TRUE) равен входному сигналу Xcas. В ручном локальном режиме (AUTO=FALSE и Ck=FALSE) выход блока Y равен входному сигналу Xloc.

В ручном каскадном режиме выход блока изменяется в соответствии с состоянием дискретных входов Up и Down. Если Up=Down (оба входа в состоянии логического нуля или логической единицы), то выход Y не изменяется. Если Up=TRUE выходной сигнал Y увеличивается:

$$Y_i = Y_{i-1} + h \cdot Tc ,$$

где  $Y_{i-1}$  – значение выхода в предыдущем цикле,  $Y_i$  – обновленное значение выхода в текущем цикле,  $h$  – скорость изменения значения выходного сигнала,  $Tc$  – длительность цикла.

Если Down=TRUE выходной сигнал Y уменьшается:

$$Y_i = Y_{i-1} - h \cdot Tc ,$$

Скорость изменения значения выходного сигнала зависит от значения входных параметров V и A. Если  $A \leq 0$ , скорость  $h=V$ . Если  $A > 0$ , то при наличии сигнала на входах Up или Down скорость  $h$  изменяется в каждом цикле контроллера в соответствии со следующей зависимостью:

$$h_i = h_{i-1} + A \cdot Tc ,$$

где  $h_{i-1}$  – значение скорости в предыдущем цикле,  $h_i$  – значение скорости в текущем цикле.

В момент изменения состояния входов Up или Down скорость  $h=V$ .

В блоке предусмотрены дискретные входы Zup и Zdown. При Zup=TRUE выходной сигнал Y не будет увеличиваться независимо от состояния входа Up. При Zdown=TRUE выходной сигнал не будет уменьшаться независимо от состояния входа Down. Состояние сигналов запрета анализируется алгоритмом только в ручном каскадном режиме.

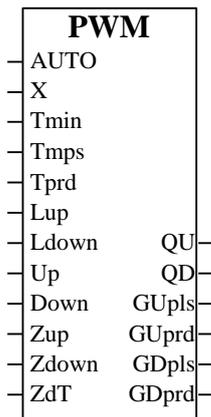
На вход Xcas подается сигнал с основного выхода ведущего аналогового регулятора RAN, на вход Xloc подается значение регулируемого параметра (вход X2) ведомого регулятора. Сигнал с выхода блока CCTRL подается на вход X0 ведущего регулятора RAN. Таким образом, в ручном каскадном режиме с по-

мощью дискретных сигналов Up и Down блока CTRL можно изменять значение выходного сигнала Y блока RAN. В ручном локальном режиме выход ведущего регулятора RAN будет равен регулируемому параметру ведомого регулятора, т.е. входные сигналы X1 и X2 ведомого регулятора будут равны. Следовательно, рассогласование для ведомого регулятора будет равно нулю.

В момент переключения на ручной каскадный режим выход блока CTRL равен выходу RAN. Таким образом, в момент переключения выход ведущего регулятора RAN не изменится и удара не произойдет.

Примечание: значение  $V < 0$  воспринимается алгоритмом как  $V = 0$ .

3.1.6 PWM – Широтно-импульсная модуляция



Вх/Вых	Обозначение	Тип	Назначение
Вход	AUTO	Boolean	Режим работы TRUE-автоматический, FALSE-ручной
	X	Real	Основной вход, пропорциональный скважности импульсов
	Tmin	Integer	Минимальная длительность импульса (мсек.)
	Tmps	Integer	Минимальная длительность паузы (мсек.)
	Tprd	Integer	Период следования импульсов (мсек.)
	Lup	Integer	Люфт в направлении “больше” (мсек.)
	Ldown	Integer	Люфт в направлении “меньше” (мсек.)
	Up	Boolean	Команда “больше” в ручном режиме
	Down	Boolean	Команда “меньше” в ручном режиме
	Zup	Boolean	Сигнал запрета в направлении “больше”
	Zdown	Boolean	Сигнал запрета в направлении “меньше”
ZdT	Boolean	Запрет накопления минимального импульса	
Выход	QU	Boolean	Дискретный выход “больше”
	QD	Boolean	Дискретный выход “меньше”
	GUpls	Integer	Длительность импульсов выхода “больше”
	GUprd	Integer	Период следования импульсов выхода “больше”
	GDpls	Integer	Длительность импульсов выхода “меньше”
	GDprd	Integer	Период следования импульсов выхода “меньше”

**Назначение**

Функциональный блок PWM используется для периодического включения и выключения нагрузки в том случае, когда скважность включения должна быть пропорциональна непрерывному управляющему сигналу.

**Описание алгоритма.**

Блок широтно-импульсной модуляции PWM преобразует значение входного сигнала X в длительность импульса. В автоматическом режиме работы блока (AUTO=TRUE) входной сигнал X преобразуется в последовательность импульсов со скважностью Q, пропорциональной входному сигналу:

$$Q=X/100\%.$$

Период следования импульсов задается на входе Tprd. Реальный период следования импульсов T кратен времени цикла контроллера и отличается от заданного Tprd на время  $t=0...T_c$ , где  $T_c$  – время одного цикла контроллера:

$$T=Tprd+t.$$

Длительность импульса определяется как произведение периода T и скважности Q:

$$P=T*|Q|,$$

При положительном значении Q импульсы будут формироваться с выхода “больше”, при отрицательном с выхода “меньше”. При абсолютном значении скважности  $|Q| \geq 1$  выход, соответствующий знаку скважности будет постоянно включен. При значении  $Q=0$  оба выхода находятся в отключенном состоянии.

Если определенная в блоке длительность импульса оказалась меньше Tmin, то длительность устанавливается равной нулю. Причем это значение длительности импульса, который не был выдан в текущем периоде  $T_i$ , запоминается и будет добавлено (в соответствии со знаком) к значению длительности импульса в следующем периоде  $T_{i+1}$ . При состоянии входа  $ZdT=TRUE$  импульс меньше минимального по времени не выдается и далее никак не учитывается, т.е. не добавляется к следующему импульсу.

Если временное расстояние между импульсами меньше заданного параметра Tmps или минимальной длительности импульса Tmin, импульсы слипаются.

Параметры  $L_{up}$  и  $L_{down}$  задают время дополнительного импульса для выборки люфта исполнительного механизма в направлении соответственно “больше” или “меньше” при изменении направления его движения.

Дискретные входы  $Z_{up}$  и  $Z_{down}$  предназначены для блокировки включения исполнительного механизма (ИМ) при достижении крайнего положения. На эти входы подаются либо дискретные сигналы с концевых выключателей ИМ, либо сигналы, сформированные программной логикой. При состоянии  $Z_{up}=TRUE$  импульсы с выхода “больше” не формируются, при  $Z_{down}=TRUE$  не формируются импульсы с выхода “меньше”.

При состоянии входа  $AUTO=FALSE$  блок работает в ручном режиме, при этом анализируется состояние входов  $Up$  и  $Down$ . Дискретные входы  $Up$  и  $Down$  предназначены для ручного управления исполнительным механизмом. При состоянии входа  $Up=TRUE$  ИМ перемещается по направлению “больше” ( $QU=TRUE$ ,  $GUpls=1000$ ,  $Guprd=1000$ ), если  $Down=TRUE$  по направлению “меньше” ( $QD=TRUE$ ,  $GDpls=1000$ ,  $GDprd=1000$ ). Если  $Up=Down$  ИМ не перемещается (все выходы блока равны нулю).

Блок является универсальным в связи с тем, что он может использоваться как для дискретного, так и для импульсного вывода. Дискретные сигналы с выходов блока  $QU$  и  $QD$  подаются на дискретный вывод, аналоговые сигналы с выходов блока  $GUpls$ ,  $Guprd$ ,  $GDpls$  и  $GDprd$  подаются на импульсный вывод. На дискретных выходах блока  $Up$  и  $Down$  формируются импульсы, длительность и период следования которых кратны циклу контроллера.

Импульсный вывод работает независимо от цикла работы контроллера, поэтому длительность импульсов с этого вывода не кратна циклу контроллера.

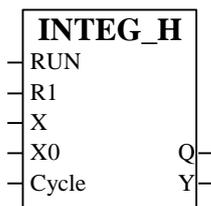
В связи с особенностью реализации импульсного вывода значение длительности импульса ( $GUpls$  или  $GDpls$ ) обновляется один раз в начале каждого периода  $T$ . При этом значение на выходе  $GUpd$  или  $GDprd$  устанавливается равным нулю. Далее до истечения периода  $T$ , значение длительности импульса не изменяется, а значение  $GUpd$  или  $GDprd$  устанавливается равным  $100c$ . Таким образом, реальный период следования импульсов с импульсного вывода будет больше заданного  $Tprd$  и кратен циклу контроллера.

Блок PWM обычно работает совместно с импульсным регулятором RIM. Сигнал с выхода импульсного регулятора необходимо подавать на вход  $X$  блока PWM. Поскольку длительность импульса определяется один раз за период, блоки RIM и PWM должны работать синхронно, т.е. с одинаковым периодом. Поэтому на вход  $Cycle$  блока RIM и на вход  $Tprd$  блока PWM необходимо подавать одно и то же значение.

**Примечания:** блок будет корректно работать только в том случае, если  $Tmin < Tprd$ . При переключении на ручной режим работы блока накопленное значение длительности импульса обнуляется вне зависимости от состояния входа  $ZdT$ . Если на входе  $Tprd$  задать значение  $Tprd=0$ , то реальный период следования импульсов  $T$  будет равен циклу контроллера. При значении  $Tprd < 0$  все выходы блока равны нулю.

## 3.2 ДИНАМИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ

### 3.2.1 INTEG\_H – Интегрирование



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы: TRUE – интегрировать, FALSE – держать
	R1	Boolean	Сигнал сброса
	X	Real	Основной вход
	X0	Real	Начальное значение
	Cycle	Integer	Период работы блока (мсек.)
Выход	Q	Boolean	Инверсия от R1
	Y	Real	Основной выход

#### Назначение

Функциональный блок INTEG\_H используется для интегрирования и (или) запоминания сигнала.

#### Описание алгоритма

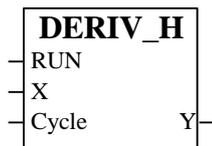
В режиме работы, когда RUN=TRUE и R1=FALSE, сигнал с входа X проходит через интегрирующее звено и поступает на выход Y. Передаточная функция интегрирующего звена:

$$W(p) = \frac{Y(p)}{X(p)} = \frac{1}{p},$$

При состоянии входа RUN=FALSE интегрирования не происходит, значение выхода  $Y_i$  не изменяется и равно предыдущему значению  $Y_{i-1}$ , если R1=FALSE.  $Y_i$  – значение выхода в текущем цикле,  $Y_{i-1}$  – значение выхода в предыдущем цикле. Если R1=TRUE выход блока Y равен значению входа X0. На выходе Q формируется инверсия от входного сигнала R1.

Параметр Cycle является периодом работы блока и периодом интегрирования. При значении  $Cycle \leq 0$  период работы блока равен циклу контроллера.

## 3.2.2 DERIV\_H – Дифференцирование



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы
	X	Real	Основной вход
	Cycle	Integer	Период работы блока (мсек.)
Выход	Y	Real	Основной выход

**Назначение**

Функциональный блок DERIV\_H применяется в схемах динамической коррекции для получения сигналов, связанных со скоростью изменения параметра.

**Описание алгоритма**

Алгоритм представляет собой идеальное дифференцирующее звено первого порядка.

Передаточная функция алгоритма имеет вид:

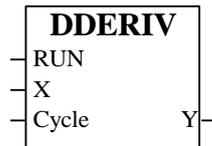
$$W(p) = \frac{Y(p)}{X(p)} = p,$$

В ручном режиме (RUN=TRUE) и в момент переключения на автоматический режим выход блока Y равен нулю.

Параметр Cycle является периодом работы блока и периодом дифференцирования (шагом дискретизации).

Примечание: при значении Cycle ≤ 0 период работы блока равен циклу контроллера.

## 3.2.3 DDERIV - Вторая производная



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы
	X	Real	Основной вход
	Cycle	Integer	Период работы блока (мсек.)
Выход	Y	Real	Основной выход

**Назначение**

Функциональный блок DDERIV применяется в схемах динамической коррекции для получения сигналов, связанных со скоростью изменения параметра.

**Описание алгоритма**

Алгоритм представляет собой идеальное дифференцирующее звено второго порядка.

Передаточная функция алгоритма имеет вид:

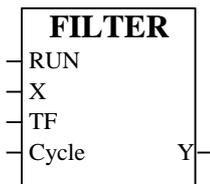
$$W(p) = \frac{Y(p)}{X(p)} = p^2,$$

В ручном режиме (RUN=FALSE) и в момент переключения на автоматический режим (RUN=TRUE) выход блока Y равен нулю.

Параметр Cycle является периодом работы блока и периодом дифференцирования (шагом дискретизации).

Примечание: при значении  $Cycle \leq 0$  период работы блока равен циклу контроллера.

## 3.2.4 FILTER – Экспоненциальное сглаживание



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы
	X	Real	Основной вход
	TF	Real	Постоянная времени фильтра (сек.)
	Cycle	Integer	Период работы блока (мсек.)
Выход	Y	Real	Отфильтрованный выход

**Назначение**

Функциональный блок FILTER используется для фильтрации высокочастотных помех, а также для динамической коррекции. Фильтр, имеющий порядок выше первого, можно получить путем последовательного включения нескольких блоков FILTER.

**Описание алгоритма**

Алгоритм является фильтром нижних частот первого порядка. При состоянии входа RUN=TRUE выходной сигнал Y равен отфильтрованному значению входного сигнала X. Передаточная функция фильтра:

$$W(p) = \frac{Y(p)}{X(p)} = \frac{1}{TF \cdot p + 1},$$

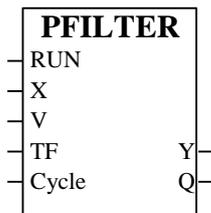
где TF - постоянная времени фильтра.

Параметр Cycle является периодом работы блока.

При состоянии входа RUN=FALSE выход блока Y равен входу X, т.е. входной сигнал не фильтруется.

Примечание: при значении cycle ≤ 0 период работы блока равен циклу контроллера.

## 3.2.5 PFILTER – Фильтрация пиков



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы
	X	Real	Входной сигнал
	V	Real	Максимальная скорость изменения сигнала (1/сек.)
	TF	Real	Максимальная длительность помехи (сек.)
	Cycle	Integer	Период работы бока (мсек.)
Выход	Y	Real	Основной выход
	Q	Boolean	Признак превышения скорости

**Назначение**

Функциональный блок PFILTER осуществляет фильтрацию входных аналоговых сигналов от помех с заданными параметрами.

**Описание алгоритма**

При состоянии входа RUN=TRUE в каждом i-ом цикле выполнения алгоритма определяется реальная скорость изменения входного сигнала X относительно выходного Y:

$$VX_i = \frac{X_i - Y_{i-1}}{T_c},$$

где  $X_i$  - значение входного сигнала в текущем цикле;  $Y_{i-1}$  - значение выходного сигнала в предыдущем цикле.

Значение  $T_c$  приближенно равно входному параметру Cycle и кратно времени цикла контроллера.

Если скорость изменения входного сигнала не больше V, т.е.  $|VX_i| \leq V$ , то входной сигнал передается на выход без изменения, при этом  $Y=X$ ,  $Q=FALSE$ . Если в каком-либо цикле обнаружено, что  $|VX_i| \geq V$ , то предполагается возможность случайного выброса и выходной сигнал  $Y_i=Y_{i-1}$ , при этом запускается измерение длительности  $T_r$  этой ситуации. Выходной сигнал не изменяется до тех пор, пока одновременно выполняются отношения  $|VX_i| \geq V$  и  $T_r < TF$ , при этом  $Q=TRUE$ . В противном случае, при  $|VX_i| \leq V$  выходной сигнал Y начинает опять отслеживать входной сигнал, т.е.  $Y_i=X_i$ , при этом  $Q=FALSE$ . Если время  $T_r$  истекло, т.е.  $T_r \geq TF$ , то выходной сигнал также начинает отслеживать входной сигнал, при этом значение  $T_r$  обнуляется  $T_r=0$ .

Таким образом, изменение входного сигнала со скоростью больше допустимой, рассматривается алгоритмом как случайный выброс, который "вырезается" из входного сигнала, если длительность этого изменения меньше заданного TF. Если длительность такого изменения сигнала больше заданного TF, то это изменение рассматривается алгоритмом как естественное и передается на выход блока с задержкой времени TF.

При состоянии входа RUN=FALSE выход блока Y равен входу X, т.е. фильтрации не происходит. При переходе входного сигнала RUN из состояния FALSE в состояние TRUE  $VX=0$ .

Примечание: при значении Cycle=0 период работы блока равен циклу контроллера. При значении  $V \leq 0$  или  $TF \leq 0$  входной сигнал X фильтроваться не будет.

## 3.2.6 DFILTER - Фильтр дискретный



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы
	X	Boolean	Основной вход
	TF	Integer	Минимальная длительность импульса (мсек.)
Выход	Y	Boolean	Отфильтрованный выход

**Назначение**

Функциональный блок DFILTER используется для фильтрации входных дискретных сигналов.

**Описание алгоритма**

Алгоритм анализирует длительность дискретных значений входного сигнала X. При состоянии входа RUN=FALSE входные импульсы с длительностью дискретных значений (1 или 0) меньше заданной не пропускаются на выход Y. Т.е. выполняются условия:

если  $t_x < TF$ , то  $t_y = TF$ ;

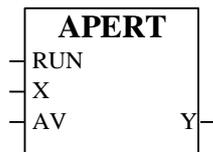
если  $t_x \geq TF$ , то  $t_y = t_x$ ;

где  $t_x$  – длительность импульсов на входе X;  $t_y$  – длительность импульсов на выходе Y; TF – заданная длительность импульсов.

Длительность импульсов с выхода Y кратна времени цикла контроллера.

При состоянии входа RUN=FALSE входной сигнал X не фильтруется, выход Y равен входу X.

## 3.2.7 APERT - Апертура



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы: TRUE – нормальный, FALSE – отключенный
	X	Real	Основной вход
	AV	Real	Величина апертуры
Выход	Y	Real	Основной выход

**Назначение**

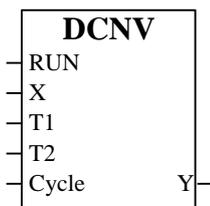
Функциональный блок APERT используется для фильтрации небольших изменений значения параметра (например, дребезг последнего бита в АЦП).

**Описание алгоритма**

При состоянии входа RUN=TRUE блок не пропускает на выход малые изменения сигнала. Если выполняется соотношение:  $|X-Y| \geq AV$ , то значение выхода изменится и будет равно входу  $Y=X$ . Если  $|X-Y| < AV$  значение выхода не изменится и останется прежним.

При состоянии входа RUN=FALSE выход всегда равен входу ( $Y=X$ ) независимо от значений X и AV.

## 3.2.8 DCNV - Динамическое преобразование



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы
	X	Real	Основной вход
	T1	Real	Интегральная постоянная (сек.)
	T2	Real	Дифференциальная постоянная (сек.)
	Cycle	Integer	Период работы блока (мсек.)
Выход	Y	Real	Основной выход

**Назначение**

Функциональный блок DCNV применяется для динамической коррекции систем управления в тех случаях, когда требуется интегрально - дифференцирующее преобразование сигнала.

**Описание алгоритма**

При состоянии входа RUN=TRUE входной сигнал X преобразуется в выходной сигнал Y в соответствии с передаточной функцией интегрально-дифференцирующего звена:

$$W(p) = \frac{T2 \cdot p + 1}{T1 \cdot p + 1},$$

где T1 и T2 - постоянные времени.

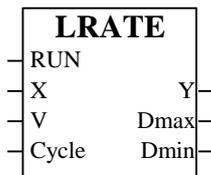
При T1=T2 выход алгоритма равен текущему значению входа. Если T1>T2 преобладает интегральная составляющая звена, если T2>T1 преобладает дифференциальная составляющая звена. Если T1≤0 или T2≤0 выход алгоритма равен нулю.

Параметр Cycle является периодом работы блока и шагом дискретизации при преобразовании входного сигнала.

При состоянии входа RUN=FALSE выход блока Y равен входному сигналу X. При этом период работы блока равен циклу контроллера.

Примечание: при значении Cycle≤0 период работы блока равен циклу контроллера.

3.2.9 LRATE - Ограничение скорости



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы
	X	Real	Основной вход алгоритма
	V	Real	Максимальная скорость (1/сек.)
	Cycle	Integer	Период работы блока (мсек.)
Выход	Y	Real	Основной выход алгоритма
	Dmax	Boolean	Признак увеличения с превышенной скоростью
	Dmin	Boolean	Признак уменьшения с превышенной скоростью

**Назначение**

Функциональный блок LRATE используется в тех случаях, когда необходимо ограничить скорость изменения сигнала.

**Описание алгоритма**

При состоянии входа RUN=TRUE сигнал с входа X проходит на выход Y в соответствии со следующей последовательностью преобразований. Определяется скорость изменения значения входа X относительно выхода:

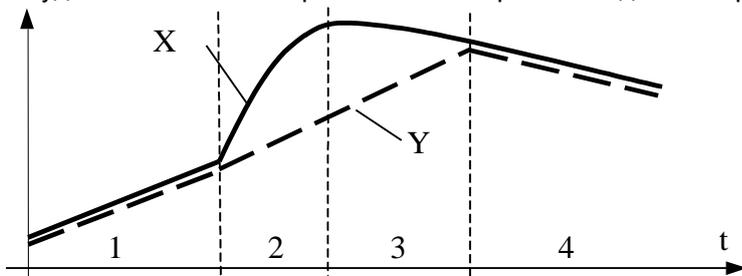
$$V_x = \frac{X - Y}{T_c}, \text{ где } T_c \text{ – время цикла.}$$

Если  $|V_x| \leq V$ , то сигнал X проходит на выход без изменений. Если  $|V_x| > V$  (при этом если  $V > 0$ ), то выход Y будет изменяться в соответствии со следующей зависимостью:

$$Y_i = Y_{i-1} + \text{sign}(V_x) \cdot V \cdot T_c,$$

где  $Y_{i-1}$  – значение выхода в предыдущем цикле;  $Y_i$  – обновленное значение выхода в текущем цикле; V – заданное значение ограничения скорости.

Выход Y будет изменяться с ограниченной скоростью V до тех пор, пока не достигнет значения X.



- 1 -  $|V_x| < V; V_y = V_x$
- 2 -  $|V_x| > V; V_y = V$
- 3 -  $|V_x| < V; V_y = V$
- 4 -  $|V_x| < V; V_y = V_x$

В любой момент времени выходной сигнал Y стремится сравниться с входным сигналом X. Если скорость изменения входного сигнала  $V_x$  меньше заданного ограничения V (т.е.  $V_x < V$ ), то выходной сигнал Y изменяется со скоростью  $V_y = V_x$ , оставаясь в каждый момент времени равным сигналу X. Если  $V_x > V$ , сигнал Y начинает изменяться с ограниченной скоростью V до тех пор, пока не сравняется с сигналом X.

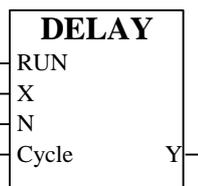
Блок имеет два дискретных выхода Dmax и Dmin. Если скорость  $V_y$  меньше заданного ограничения V, сигналы на обоих выходах равны нулю. В противном случае появляется сигнал на выходе Dmax или Dmin, в зависимости от того, увеличивается (изменяется в направлении "больше") или уменьшается (изменяется в направлении "меньше") выходной сигнал Y.

$V_y$	Dmax	Dmin
$ V_y  < V$	0	0
$V_y = V$	1	0
$V_y = -V$	0	1

При состоянии входа RUN=FALSE выход Y равен входу X, выходы Dmax и Dmin в состоянии логического нуля.

Примечание: при значении Cycle ≤ 0 период работы блока равен циклу контроллера.

## 3.2.10 DELAY – Запоздывание



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы: TRUE – нормальный, FALSE – отключенный
	X	Real	Основной вход
	N	Integer	Количество циклов запаздывания от 1 до 128
	Cycle	Integer	Длительность цикла (мсек.)
Выход	Y	Real	Основной выход

**Назначение**

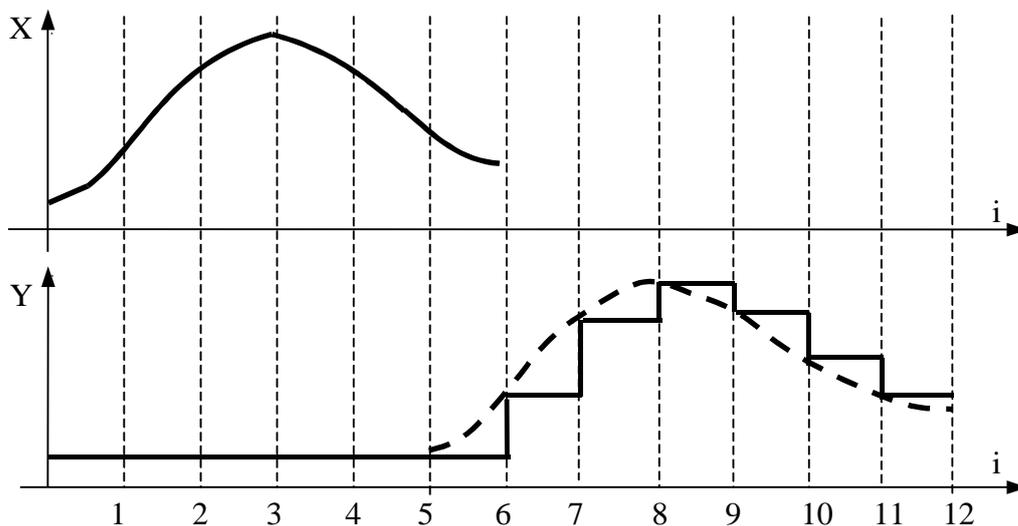
Функциональный блок DELAY предназначен для моделирования звена чистого запаздывания и используется для введения динамической коррекции или для моделирования динамических свойств объекта управления.

**Описание алгоритма**

При состоянии входа RUN=TRUE текущее значение входного сигнала X заносится в массив и будет прочитано из этого массива через N циклов. Таким образом, текущее значение X появится на выходе Y через N циклов. При этом выходной сигнал Y будет запаздывать относительно сигнала X на время  $N \cdot \text{Cycle}$ , где Cycle – период работы блока (длительность одного цикла запаздывания). Рисунок поясняет работу алгоритма при  $N=5$ .

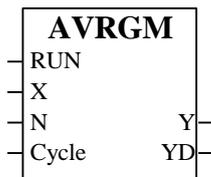
При состоянии входа RUN=FALSE или при  $N \leq 0$  выход Y равен текущему значению входа X. При переходе входного сигнала RUN из состояния FALSE в состояние TRUE все N элементов массива инициализируются текущим значением X.

Примечание: Изменение значения N при состоянии входа RUN=TRUE алгоритмом не воспринимается, поэтому значение N можно изменять только при состоянии RUN=FALSE. Если значение параметра Cycle меньше времени цикла контроллера (или  $\text{Cycle} \leq 0$ ), то период работы блока равен циклу контроллера. Значение  $N > 128$  алгоритм воспринимает как  $N=128$ .



$i$  – количество циклов, прошедших с момента запуска алгоритма

## 3.2.11 AVRGM – Скользящее среднее



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы: TRUE – нормальный, FALSE – отключенный
	X	Real	Основной вход
	N	Integer	Количество циклов запаздывания от 1 до 128
	Cycle	Integer	Период работы блока (мсек.)
Выход	Y	Real	Основной выход
	YD	Real	Запаздывающее значение входа X

**Назначение**

Функциональный блок AVRGM применяется для вычисления среднего из нескольких (до 128) последних значений аналогового сигнала.

**Описание алгоритма**

Структура и работа алгоритма AVRGM соответствует алгоритму запаздывания DELAY, т.е. значение входного сигнала X заносится в массив и будет прочитано из этого массива через N циклов. Отличие алгоритма AVRGM заключается в наличии дополнительного сумматора, который усредняет значение сигнала на выходе. При состоянии входа RUN=TRUE выходной сигнал Y равен :

$$Y = \frac{\sum_{i=1}^N X_i}{N},$$

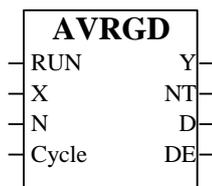
где  $1 \leq N \leq 128$  – количество циклов запаздывания;  $X_i$  - значение сигнала X в i-ом цикле запаздывания.

Выход YD равен запаздывающему значению входа. Время запаздывания равно N·Cycle, где Cycle – период работы блока (длительность одного цикла запаздывания).

Если значение входа  $N \leq 0$  или RUN=FALSE выходы алгоритма равны текущему значению входа  $Y=X$  и  $YD=X$ . В момент перехода входного сигнала RUN из состояния FALSE в состояние TRUE все N элементов массива инициализируются текущим значением входа X.

**Примечание:** Изменение значения N при состоянии входа RUN=TRUE алгоритмом не воспринимается, поэтому значение N можно изменять только при состоянии RUN=FALSE. Если значение параметра Cycle меньше времени цикла контроллера (или  $\text{Cycle} \leq 0$ ), то период работы блока равен циклу контроллера. Значение  $N > 128$  алгоритм воспринимает как  $N=128$ .

## 3.2.12 AVRGD - Дискретное среднее



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы: TRUE – нормальный , FALSE – отключенный
	X	Real	Основной вход
	N	Integer	Заданное число отсчетов
	Cycle	Integer	Период отсчета (мсек.)
Выход	Y	Real	Основной выход
	NT	Integer	Номер текущего отсчета
	D	Boolean	Признак отсчета
	DE	Boolean	Признак окончания отсчетов

**Назначение**

Функциональный блок AVRGD применяется для усреднения аналогового сигнала на фиксированном отрезке времени за заданное число отсчетов.

**Описание алгоритма**

В алгоритме суммируются значения входного сигнала, полученные за N отсчетов, после чего полученная сумма делится на N. Число отсчетов, за которое усредняется выходной сигнал, задается входным параметром N.

Счетчик внутри блока считает текущее число отсчетов NT и в момент, когда NT=N среднее значение, вычисленное сумматором, запоминается и обновляется сигнал на выходе Y. Тем самым заканчивается очередной интервал усреднения, после чего вплоть до окончания нового интервала сигнал Y не изменяется.

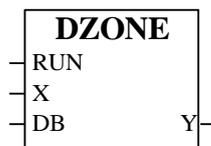
На выходе NT формируется текущий (внутри интервала) номер отсчета. Отсчеты производятся через период времени Cycle (входной параметр блока). На выходе D в начале каждого отсчета в течение одного цикла работы контроллера устанавливается D=TRUE, в остальное время D=FALSE. По окончании интервала усреднения устанавливается сигнал в течение времени Cycle на выходе DE=TRUE (до окончания этого интервала DE=FALSE).

При состоянии входа RUN=FALSE или при значении  $N \leq 0$  выходной сигнал  $Y=X$ , т.е. выходной сигнал равен текущему значению входного сигнала. Сумматор и счетчик циклов обнуляются. Выходы D, DE, NT равны нулю.

Примечание: При значении  $Cycle \leq 0$  период отсчета равен циклу контроллера.

### 3.3 СТАТИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ

#### 3.3.1 DZONE - Зона нечувствительности



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы: TRUE-нормальный, FALSE-отключенный
	X	Real	Основной вход
	DB	Real	Зона нечувствительности
Выход	Y	Real	Основной выход

#### Назначение

Функция DZONE используется при создании схем регулирования, чтобы оградить исполнительные органы от лишних операций при малом отличии регулируемой величины от значения задания.

#### Описание алгоритма

При состоянии входа RUN=TRUE алгоритм работает в соответствии со следующим законом:

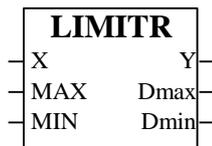
$$\begin{aligned}
 Y &= 0, \text{ если } |X| \leq DB/2, \\
 Y &= X - DB/2, \text{ если } X > DB/2, \\
 Y &= X + DB/2, \text{ если } X < DB/2,
 \end{aligned}$$

где DB – заданная зона нечувствительности, X – основной вход алгоритма, Y – основной выход алгоритма

Т.е. при абсолютном значении входа  $|X| \leq DB/2$  выход алгоритма равен нулю, в остальных случаях из значения X вычитается значение DB/2 в соответствии со знаком. Значение входа  $DB < 0$  алгоритм воспринимает как  $DB=0$ .

При состоянии входа RUN=FALSE выход алгоритма равен входу  $Y=X$ .

## 3.3.2 LIMITR - Ограничение



Вх/Вых	Обозначение	Тип	Назначение
Вход	X	Real	Основной вход алгоритма
	MAX	Real	Верхняя граница
	MIN	Real	Нижняя граница
Выход	Y	Real	Основной выход алгоритма
	Dmax	Boolean	Достижение верхней границы
	Dmin	Boolean	Достижение нижней границы

**Назначение**

Функция LIMITR используется для ограничения верхней и (или) нижней границы диапазона изменения сигнала.

**Описание алгоритма**

Алгоритм содержит ограничитель верхнего и нижнего значения сигнала. На двух дискретных выходах Dmax и Dmin фиксируется достижение сигналом верхней и нижней границы ограничения. Работа алгоритма определяется следующей таблицей:

X	Y	Dmax	Dmin
$MAX > X > MIN$	X	0	0
$X \geq MAX$	MAX	1	0
$X \leq MIN$	MIN	0	1

Уровни ограничений задаются входами MAX, MIN. Если  $MAX \leq MIN$ , то выход  $Y=X$ .

## 3.3.3 INS0 - Вставка нуля

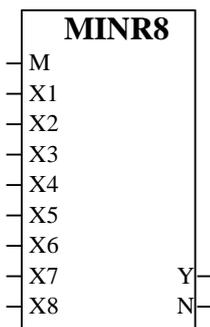


Вх/Вых	Обозначение	Тип	Назначение
Вход	X	Real	Основной вход блока
Выход	Y	Real	Основной выход блока

**Описание алгоритма**

Этот блок воспроизводит на выходе входное значение. Однако при каждом изменении входного значения выход обнуляется на один цикл работы контроллера и только в следующем цикле становится равным входу.

## 3.3.4 MINR - Минимум



Вх/Вых	Обозначение	Тип	Назначение
Вход	M	Integer	Количество используемых входов
	X1	Real	1-й вход
	X2	Real	2-ой вход
	...	...	...
	X8	Real	8-ой вход
Выход	Y	Real	Минимум из входов
	N	Integer	Номер входа с минимальным значением

**Назначение**

Функция MINR используется для выделения минимального из нескольких (до 8) сигналов.

**Описание алгоритма**

На вход алгоритма поступают сигналы, число которых  $0 < M < 8$  и задается на входе M. Выходной сигнал Y равен минимальному из этих сигналов:

$$Y = \min \{X1; X2; \dots; XM\}$$

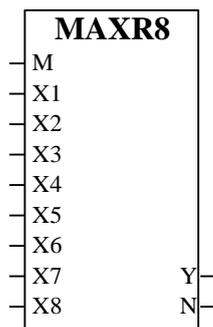
При этом входные сигналы, номер которых больше M не учитываются.

Алгоритм имеет дополнительный выход N, на котором формируется число, равное номеру входного сигнала, прошедшего на выход (т.е. являющегося минимальным). Если имеется группа равных между собой сигналов, причем эти сигналы являются минимальными, то номер N равен минимальному номеру сигналов в этой группе.

При  $M < 1$  выходы Y и N равны нулю. Значение  $M > 8$  алгоритм воспринимает как  $M = 8$ .

**Примечания:** На неиспользуемые входы можно подавать любые значения, т.к. эти значения не будут учитываться при определении минимума. Для блока MINR8 максимальное количество используемых входов – 8, для MINR4 – 4 входа

## 3.3.5 MAXR - Максимум



Вх/Вых	Обозначение	Тип	Назначение
Вход	M	Integer	Количество используемых входов
	X1	Real	1-ый вход
	X2	Real	2-ой вход
	...	...	...
	X8	Real	8-ой вход
Выход	Y	Real	Максимум из входов
	N	Integer	Номер входа с максимальным значением

**Назначение.**

Функция MAXR используется для выделения максимального из нескольких (до 8) сигналов.

**Описание алгоритма.**

На вход алгоритма поступают сигналы, число которых  $0 < M < 8$  и задается на входе M. Выходной сигнал равен максимальному из этих сигналов:

$$Y = \max \{X1; X2; \dots; XM\}$$

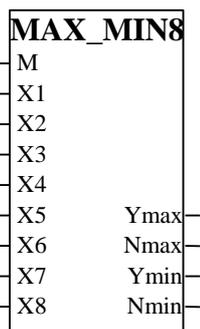
При этом входные сигналы, номер которых больше M не учитываются.

Алгоритм имеет дополнительный выход N, на котором формируется число, равное номеру входного сигнала, прошедшего на выход (т.е. являющегося максимальным). Если имеется группа равных между собой сигналов, причем эти сигналы являются максимальными, то номер N равен минимальному номеру сигналов в этой группе.

При  $M < 1$  выходы Y и N равны нулю. Значение  $M > 8$  алгоритм воспринимает как  $M = 8$ .

**Примечания:** На неиспользуемые входы можно подавать любые значения, т.к. эти значения не будут учитываться при определении максимума. Для блока MAXR8 максимальное количество используемых входов – 8, для MAXR4 – 4 входы.

## 3.3.6 MAX\_MIN – Максимум и минимум



Вх/Вых	Обозначение	Тип	Назначение
Вход	M	Integer	Количество используемых входов
	X1	Real	1-ый вход
	X2	Real	2-ой вход
	...	...	...
	X8	Real	8-ой вход
Выход	Ymax	Real	Максимум из входов
	Nmax	Integer	Номер входа с максимальным значением
	Ymin	Real	Минимум из входов
	Nmin	Integer	Номер входа с минимальным значением

**Назначение.**

Функция MAX\_MIN используется для выделения максимального и минимального из нескольких (до 8) сигналов.

**Описание алгоритма.**

На вход алгоритма поступают сигналы, число которых  $0 < M < 8$  и задается на входе M. Выходной сигнал Ymax равен максимальному из этих сигналов, выходной сигнал Ymin равен минимальному из этих сигналов:

$$Y_{max} = \min \{X_1; X_2; \dots; X_M\};$$

$$Y_{min} = \min \{X_1; X_2; \dots; X_M\}$$

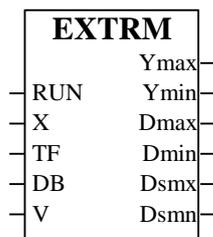
При этом входные сигналы, номер которых больше M не учитываются.

Алгоритм имеет дополнительные выходы Nmax и Nmin. На выходе Nmax формируется число, равное номеру входного сигнала, прошедшего на выход Ymax (т.е. являющегося максимальным). На выходе Nmin формируется число, равное номеру входного сигнала, прошедшего на выход Ymin (т.е. являющегося минимальным). Если имеется группа равных между собой сигналов и эти сигналы являются максимальными, то номер Nmax равен минимальному номеру сигналов в этой группе. Если имеется группа равных между собой сигналов и эти сигналы являются минимальными, то номер Nmin равен минимальному номеру сигналов в этой группе.

При  $M < 1$  все выходы алгоритма равны нулю. Значение  $M > 8$  алгоритм воспринимает как  $M = 8$ .

**Примечания:** На неиспользуемые входы можно подавать любые значения, т.к. эти значения не будут учитываться при определении максимума и минимума. Для блока MAX\_MIN8 максимальное количество используемых входов – 8, для MAX\_MIN4 – 4 входа.

## 3.3.7 EXTRM - Экстремум



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы: FALSE – сброс, TRUE – запуск
	X	Real	Входной сигнал
	TF	Real	Постоянная времени фильтра (сек.)
	DB	Real	Зона нечувствительности
	V	Real	Допустимая скорость изменения входного сигнала (1/сек.)
Выход	Ymax	Real	Максимальное значение входного сигнала
	Ymin	Real	Минимальное значение входного сигнала
	Dmax	Boolean	Фиксация максимума
	Dmin	Boolean	Фиксация минимума
	Dsmx	Boolean	Признак поиска максимума
	Dsmn	Boolean	Признак поиска минимума

**Назначение**

Функциональный блок EXTRM применяется для поиска и фиксации максимального и/или минимального значения меняющегося во времени сигнала. В частности, алгоритм используется в задачах оптимизации.

**Описание алгоритма**

Входной сигнал X проходит через фильтр нижних частот. Фильтр нижних частот имеет передаточную функцию:

$$W(p) = \frac{1}{TF \cdot p + 1}, \text{ где } TF - \text{ постоянная времени фильтра.}$$

Если  $TF \leq 0$ , то входной сигнал X не фильтруется.

На выходах блока Ymax, Ymin фиксируется последнее соответственно максимальное и минимальное значение отфильтрованного сигнала Xf.

Алгоритм работает следующим образом (Рисунок 3.2). До тех пор, пока блок находится в состоянии сброса (RUN=FALSE) поиск экстремума не ведется и выходные сигналы Ymax = Ymin = X. При запуске блока (RUN=TRUE) выходные сигналы Ymax и Ymin замораживаются, и начинается поиск экстремума.

Как только найден максимум, выходной сигнал Ymax становится равным максимальному значению Xf. Когда алгоритм фиксирует минимум, выходной сигнал Ymin принимает значение, равное минимальному значению Xf. Выходы Ymax и Ymin остаются неизменными до обнаружения очередного экстремума – соответственно максимума или минимума.

В момент обнаружения максимума или минимума на время Tc, равное времени цикла работы контроллера, на дискретных выходах соответственно Dmax или Dmin формируются дискретные сигналы Dmax = TRUE или Dmin = TRUE.

На дискретных выходах Dsmx (поиск максимума) и Dsmn (поиск минимума) формируются сигналы, свидетельствующие о направлении поиска. В состоянии сброса поиск не ведется и Dsmx = Dsmn = FALSE. Если в текущем цикле ведется поиск максимума, то Dsmx = TRUE, Dsmn = FALSE, если ведется поиск минимума, то Dsmx = FALSE и Dsmn = TRUE.

В алгоритме предусмотрены меры по повышению помехозащищенности процесса поиска. Как отмечалось выше, входной сигнал проходит через фильтр нижних частот. Также имеется возможность введения зоны нечувствительности и предусмотрена возможность анализа на допустимую скорость изменения сигнала в районе экстремума. Эти меры могут применяться по отдельности, а также в любых сочетаниях.

Влияние зоны нечувствительности и допустимой скорости показано на Рисунок 3.3.

Если на входе DB установлено значение зоны нечувствительности  $DB \leq 0$ , то экстремум фиксируется сразу же, как только знак производной сменился на противоположный (именно такая ситуация представлена на рисунке 4.1). Если введена зона нечувствительности ( $DB > 0$ ), то экстремум фиксируется не сразу, а лишь после того, как, пройдя экстремум, сигнал изменится на величину, большую DB. Если же сигнал изменится на меньшую величину и затем вновь начнет изменяться в исходном направлении, экстремум не фиксируется (Рисунок 3.3).

На входе  $V$  устанавливается допустимая скорость изменения сигнала, при которой фиксируется экстремум. Если на входе  $V$  установлено значение  $V \leq 0$ , то ограничения на скорость не накладываются. В противном случае экстремум фиксируется лишь тогда, когда после прохождения экстремума (или, при наличии зоны нечувствительности – после прохождения этой зоны) скорость изменения сигнала  $V_x \leq V$  (Рисунок 3.3).

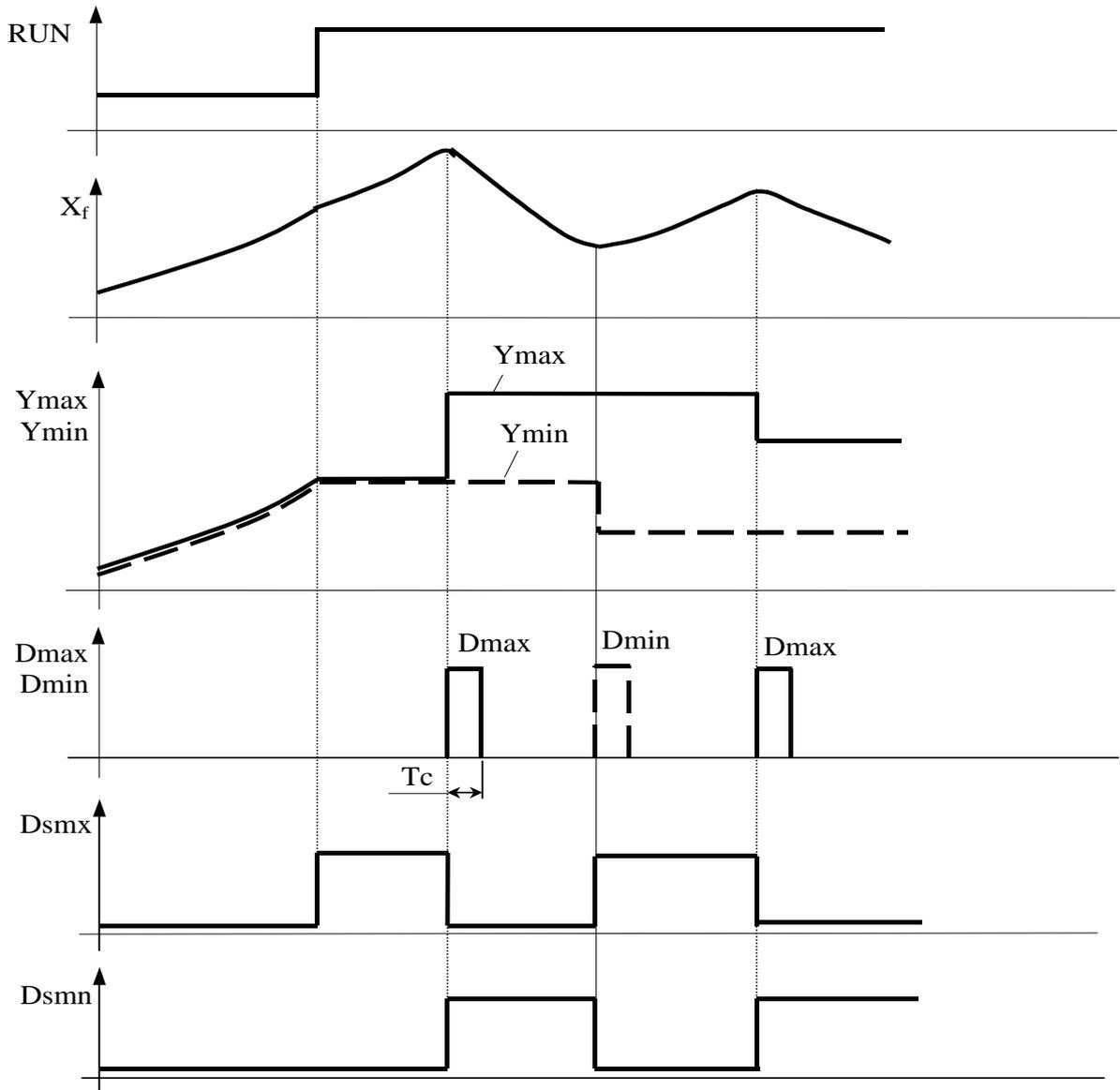


Рисунок 3.2 Диаграмма работы функционального блока EXTRM.

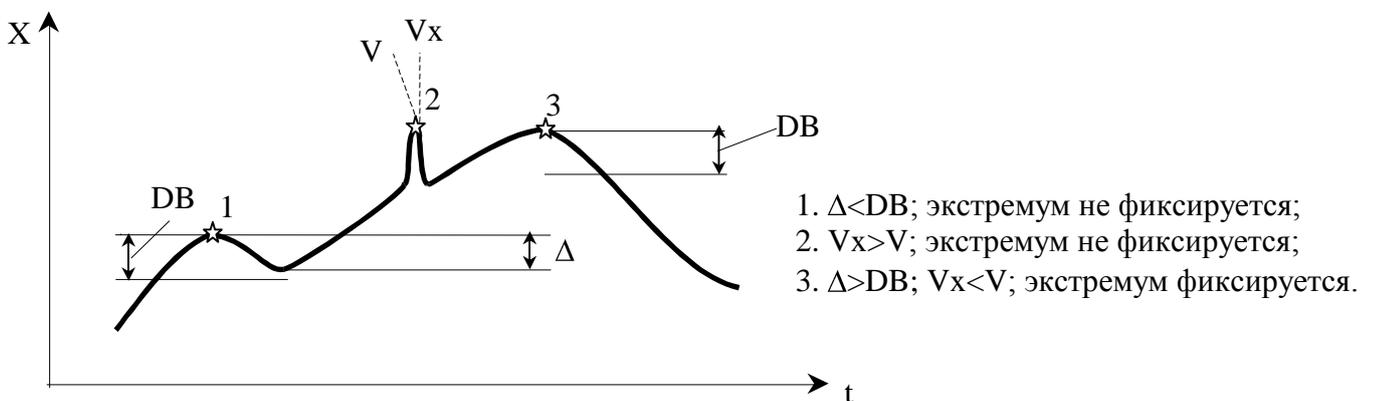
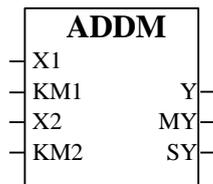


Рисунок 3.3 Влияние зоны нечувствительности и допустимой скорости.

## 3.3.8 ADDM - Суммирование с выделением модуля и знака



Вх/Вых	Обозначение	Тип	Назначение
Вход	X1	Real	1-й масштабируемый вход
	KM1	Real	1-й масштабный коэффициент
	X2	Real	2-й масштабируемый вход
	KM2	Real	2-й масштабный коэффициент
Выход	Y	Real	Основной выход
	MY	Real	Модуль суммы
	SY	Boolean	Знак суммы: FALSE – положительная, TRUE - отрицательная

**Назначение**

Функция ADDM применяется для суммирования двух чисел с выделением знака и модуля числа на отдельных выходах.

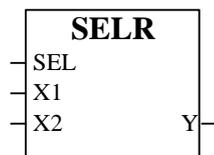
**Описание алгоритма**

Функциональный блок осуществляет масштабирование и суммирование двух входных сигналов X1 и X2. На выходах блока формируется три сигнала:

1. Сигнал  $Y=KM1*X1+KM2*X2$ .
2. Сигнал MY, равный модулю сигнала Y,  $MY=|Y|$ .
3. Сигнал SY, равный знаку сигнала Y (при  $Y \geq 0$  SY=FALSE, иначе SY=TRUE).

### 3.4 АНАЛОГО-ДИСКРЕТНЫЕ ПРЕОБРАЗОВАНИЯ

#### 3.4.1 SELR – Переключатель на 2 входа



Вх/Вых	Обозначение	Тип	Назначение
Вход	SEL	Boolean	Выбор входа: FALSE – X1, TRUE – X2
	X1	Real	1-ый переключаемый вход
	X2	Real	2-ой переключаемый вход
Выход	Y	Real	Основной выход

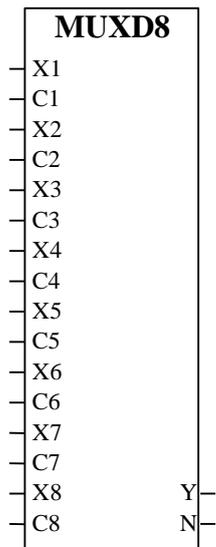
#### Назначение

Функция SELR предназначена для переключения двух аналоговых сигналов.

#### Описание алгоритма

Переключение сигналов производится по состоянию дискретного входа SEL. При состоянии входа SEL=FALSE выход Y равен входу X1. При состоянии SEL=TRUE выход Y равен входу X2.

3.4.2 MUXD – Дискретный переключатель



Вх/Вых	Обозначение	Тип	Назначение
Вход	X1	Real	1-й переключаемый вход
	C1	Boolean	Команда выбора входа 1
	X2	Real	2-й переключаемый вход
	C2	Boolean	Команда выбора входа 2
	....	....	....
	X8	Real	8-й переключаемый вход
	C8	Boolean	Команда выбора входа 8
	Выход	Y	Real
N		Integer	Порядковый номер выбранного входа

**Назначение**

Функция MUXD представляет собой многополюсный переключатель аналоговых сигналов, положение которого определяется дискретными сигналами, поступающими на вход алгоритма. Алгоритм используется для выбора одного из нескольких (до 8) сигналов. Если на аналоговых входах алгоритма заданы константы, то алгоритм может использоваться для дискретной установки требуемой константы.

**Описание алгоритма**

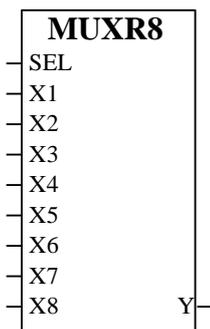
Функциональный блок является переключателем аналоговых сигналов. Если на всех дискретных входах C, управляющих положением переключателя, сигнал отсутствует (лог. 0), выходной сигнал Y=0. Если на какой-либо из дискретных входов подается дискретный сигнал Ci =1, выход алгоритма Y подключается к одноименному (по номеру индекса) аналоговому входу Xi.

Если дискретные сигналы подаются одновременно на несколько входов, приоритетен вход с младшим номером.

Сигнал на выходе Y равен сигналу на выбранном входе. Число на выходе N указывает номер выбранного входа. Если ни один вход не выбран, оба выхода равны нулю Y=0, N=0.

Примечание: функциональный блок MUXD8 рассчитан на 8 переключаемых входов, MUXD4 – на 4 входа.

## 3.4.3 MUXR - Переключатель по номеру



Вх/Вых	Обозначение	Тип	Назначение
Вход	SEL	Integer	Номер переключаемого входного сигнала от 0 до 7
	X1	Real	1-ый переключаемый вход
	X2	Real	2-ой переключаемый вход
	...	...	...
	X8	Real	8-ой переключаемый вход
Выход	Y	Real	Основной выход

**Назначение**

Функция MUXR используется для переключения нескольких (до 8) сигналов. Положение переключателя определяется значением числа, поступающего на специальный вход алгоритма.

**Описание алгоритма**

Функция MUXR реализует переключатель аналоговых сигналов. На вход N подается число, определяющее номер подключенного входа.

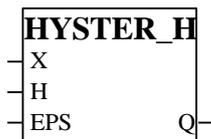
Работа переключателя описывается таблицей :

SEL	SEL<0	SEL=0	SEL=1	...	SEL=7	SEL>7
Y	0	X1	X2	...	X8	0

Сигнал на выходе Y равен сигналу на входе в соответствии со значением SEL.

Примечание: функция MUXR8 рассчитана на 8 переключаемых входов, MUXR4 – на 4 входа.

3.4.4 HYSTER\_H - Пороговый элемент по верхнему пределу



Вх/Вых	Обозначение	Тип	Назначение
Вход	X	Real	Основной вход
	H	Real	Порог срабатывания
	EPS	Real	Гистерезис
Выход	Q	Boolean	Тревога: TRUE, если пороговый элемент сработал

**Назначение**

Функциональный блок HYSTER\_H применяется для контроля за выходом сигнала из ограниченной сверху области допустимых значений.

**Описание алгоритма**

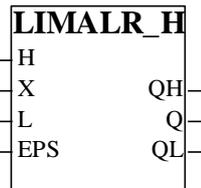
Входной сигнал X подается на звено порогового элемента с порогом срабатывания H и гистерезисом EPS. Звено порогового элемента срабатывает, когда  $X > H + EPS$ , при этом появляется дискретный сигнал на выходе  $Q = TRUE$  до тех пор, пока входной сигнал не уменьшится до значения  $X < H - EPS$ . Логика работы блока описывается таблицей:

X	Q
$X < H - EPS$	0
$X > H + EPS$	1
$H - EPS \leq X \leq EPS + H$	$Q_{i-1}$

Здесь  $Q_{i-1}$  – предыдущее значение выходного сигнала.

На входах H, EPS задается соответственно порог срабатывания и гистерезис. Значение параметра EPS необходимо задавать только положительным  $EPS \geq 0$ .

## 3.4.5 LIMLR\_H - Гистерезис по верхнему и нижнему пределам



Вх/Вых	Обозначение	Тип	Назначение
Вход	H	Real	Верхний порог срабатывания
	X	Real	Основной вход
	L	Real	Нижний порог срабатывания
	EPS	Real	Гистерезис
Выход	QH	Boolean	Верхняя тревога: TRUE, если X выше границы H
	Q	Boolean	Тревога: TRUE, если X вне границ
	QL	Boolean	Нижняя тревога: TRUE, если X ниже границы L

**Назначение**

Функциональный блок LIMLR\_H или «нуль-орган» используется для контроля за выходом сигнала из области допустимых значений, ограниченной верхним и нижним пределами.

**Описание алгоритма**

Функциональный блок LIMLR\_H реализует контроль входного аналогового сигнала X на достоверность. Значение сигнала считается достоверным, если выполняется условие:

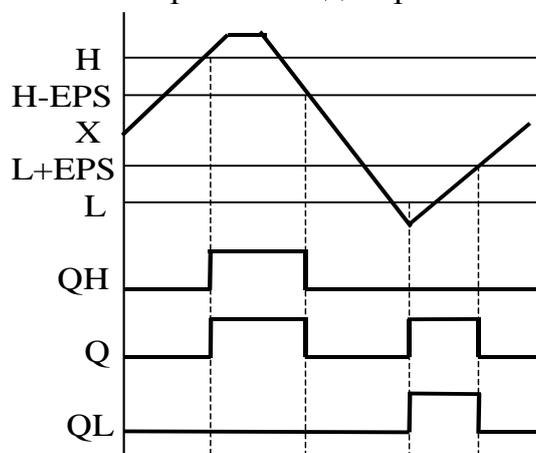
$$L \leq X \leq H,$$

где H, L – верхняя и нижняя границы достоверности. Если  $H \leq L$  блок работать не будет.

При выходе аналогового сигнала за любую из предварительно заданных границ логический признак QH или QL, сигнализирующий о выходе за конкретную границу, переходит в состояние TRUE. Также в состоянии TRUE переходит другой логический признак Q, сигнализирующий о том, что сигнал находится вне границ. При нахождении в пределах границ все выходы функционального блока находятся в состоянии FALSE. Логика работы блока поясняется таблицей и временной диаграммой:

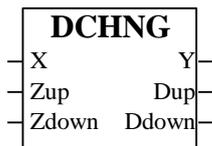
X	QH	QL
$X < H - EPS$	0	*
$X \geq H$	1	*
$H - EPS \leq X < H$	$QH_{i-1}$	*
$X > L + EPS$	*	0
$X \leq L$	*	1
$L + EPS \geq X > L$	*	$QL_{i-1}$

\* - выходной сигнал не зависит от данного условия

**Временная диаграмма**

Примечание: Значение параметра EPS задается только положительное ( $EPS \geq 0$ ).

3.4.6 DCHNG - Запрет изменения



Вх/Вых	Обозначение	Тип	Назначение
Вход	X	Real	Основной вход
	Zup	Boolean	Команда запрета вверх
	Zdown	Boolean	Команда запрета вниз
Выход	Y	Real	Основной выход
	Dup	Boolean	Признак запрета вверх
	Ddown	Boolean	Признак запрета вниз

**Назначение**

Функция DCHNG применяется для запрета увеличения или уменьшения сигнала, например, при действии технологических защит. В частности, запрет может блокировать изменение сигнала задания или выходного сигнала аналогового регулятора.

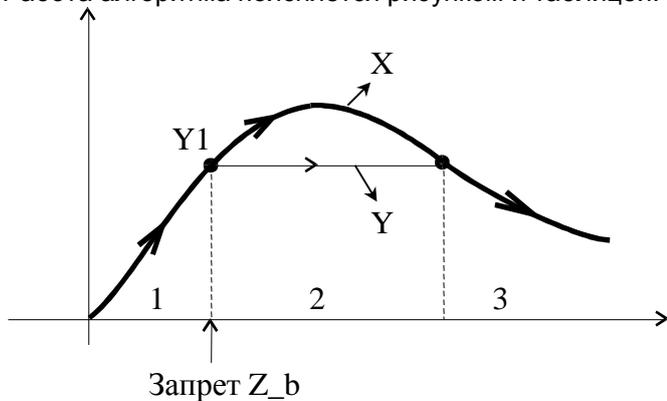
**Описание алгоритма**

Алгоритм содержит узел запрета, который реагирует на дискретные сигналы запрета в направлении увеличения (Zup - вверх) или уменьшения (Zdown - вниз). Помимо основного выхода Y алгоритм содержит два дискретных выхода Dup и Ddown, сигнализирующих о том, что алгоритм работает в режиме запрета.

Если команды запрета отсутствуют, то выходной сигнал равен входному  $Y = X$ . Если поступила команда Zup и  $X > Y$ , то алгоритм переходит в режим "запрета вверх", при этом  $Y = Y1 = \text{const}$  и  $Dup = 1$ , где Y1 - значение Y в момент поступления команды запрета. Этот режим сохраняется до тех пор, пока входной сигнал не уменьшится до значения  $X < Y1$ , после чего режим запрета снимается и вновь  $Y = X$ .

Аналогично, но с реакцией на противоположную тенденцию изменения X, работает алгоритм в режиме "запрет вниз".

Работа алгоритма поясняется рисунком и таблицей.

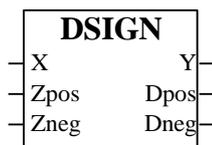


- 1. Zup=0; Y=X; Dup=0;
- 2. Zup=1; Y=Y1=const; Dup=1
- 3. Zup=1; Y=X; Dup=0

Zup	Zdown	X	Y	Dup	Ddown	Режим запрета
0	0	*	$Y = X$	0	0	-
1	0	$X \geq Y$	$Y = Y1 = \text{const}$	1	0	+
1	0	$X < Y$	$Y = X$	0	0	-
0	1	$X \leq Y$	$Y = Y1 = \text{const}$	0	1	+
0	1	$X > Y$	$Y = X$	0	0	-
1	1	$X = Y$	$Y = Y1 = \text{const}$	1	1	+
1	1	$X > Y$	$Y = Y1 = \text{const}$	1	0	+
1	1	$X < Y1$	$Y = Y1 = \text{const}$	0	1	+

\* - значение сигнала безразлично;  
Y1 - значение Y в момент прихода команды запрета.

## 3.4.7 DSIGN - Запрет знака



Вх/Вых	Обозначение	Тип	Назначение
Вход	X	Real	Основной вход
	Zpos	Boolean	Команда запрета "плюс"
	Zneg	Boolean	Команда запрета "минус"
Выход	Y	Real	Основной выход
	Dpos	Boolean	Режим запрета "плюс"
	Dneg	Boolean	Режим запрета "минус"

**Назначение**

Функция DSIGN применяется для запрета перехода сигнала в область положительных или отрицательных значений.

**Описание алгоритма**

Алгоритм содержит узел запрета, управляемый входными дискретными сигналами Cplus и Cmin. При действии команды Cmin запрещается переход выходного сигнала Y в область положительных значений, команда Cplus запрещает изменение Y в область отрицательных значений. Выход Dplus устанавливается в 1 при режиме запрета "плюс", выход Dmin устанавливается в 1 при режиме запрета "минус". Работа алгоритма описывается следующей таблицей.

Zpos	Zneg	X	Y	Dpos	Dneg	Режим запрета
0	0	*	$Y=X$	0	0	-
1	0	$X>0$	$Y=0$	1	0	+
1	0	$X \leq 0$	$Y=X$	0	0	-
0	1	$X<0$	$Y=0$	0	1	+
0	1	$X \geq 0$	$Y=X$	0	0	-
1	1	*	$Y=0$	1	1	+

\* - значение сигнала безразлично.

3.4.8 FLW\_HOLD - Слежение-запоминание



Вх/Вых	Обозначение	Тип	Назначение
Вход	FLOW	Boolean	Команда слежения
	X	Real	Основной вход
Выход	DF	Boolean	Признак слежения
	Y	Real	Основной выход

**Назначение**

Функциональный блок FLW\_HOLD используется для слежения и запоминания аналогового сигнала.

**Описание алгоритма**

На вход блока X подается аналоговый сигнал, который нужно запомнить. На вход FLOW подается команда слежения.

До тех пор, пока FLOW=TRUE, входной сигнал X передается на выход Y, т.е. выход "следит" за входом. При FLOW=FALSE слежение блокируется, текущее значение Y запоминается и перестает зависеть от входного сигнала X.

Выходной сигнал DF=FLOW, т.е. повторяет входной управляющий сигнал.

Работа алгоритма описывается следующей таблицей.

FLOW	Y	DF
TRUE	Y=X	TRUE
FALSE	Y=X0	FALSE

X0 - значение входного сигнала X в момент снятия команды FLOW.

## 3.4.9 LATCH - Запоминание



Вх/Вых	Обозначение	Тип	Назначение
Вход	CLK	Boolean	Команда запоминания
	X	Real	Основной вход
Выход	Q	Boolean	Признак запоминания
	Y	Real	Основной выход

**Назначение**

Функциональный блок LATCH используется для запоминания аналогового сигнала.

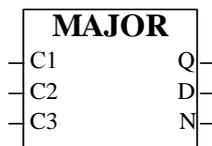
**Описание алгоритма**

На вход блока X подается аналоговый сигнал, который нужно запомнить. На вход CLK подается команда запоминания.

Запись осуществляется по переднему фронту дискретного сигнала CLK, т.е. в момент перехода CLK из состояния логического 0 в состояние логической 1. В этот момент текущее значение входного сигнала X запоминается и выход Q принимает состояние TRUE на 1 цикл контроллера. Запомненное значение остается неизменным вплоть до прихода нового фронта сигнала CLK. Сигнал на выходе Y соответствует запомненному значению сигнала X.

### 3.5 ДИСКРЕТНОЕ УПРАВЛЕНИЕ

#### 3.5.1 MAJOR - Мажорирование



Вх/Вых	Обозначение	Тип	Назначение
Вход	C1	Boolean	1-й вход
	C2	Boolean	2-й вход
	C3	Boolean	3-й вход
Выход	Q	Boolean	Основной выход
	D	Boolean	Признак несовпадения
	N	Integer	Номер входного сигнала, состояние которого не совпадает с состоянием двух других сигналов

#### Назначение

Функциональный блок MAJOR используется для повышения достоверности дискретных сигналов, поступающих, например, от модулей дискретного ввода контроллера. Алгоритм работает по правилу "два из трех".

#### Описание алгоритма

Алгоритм содержит узел мажорирования, а также логику, определяющую номер сигнала, состояние которого не совпадает с состоянием двух других сигналов. Работа алгоритма описывается следующей таблицей.

Состояние входных сигналов	Q	D	N
$C1 = C2 = C3 = C$	C	0	0
$C2 = C3 = C; C1 \neq C$	C	1	1
$C1 = C3 = C; C2 \neq C$	C	1	2
$C1 = C2 = C; C3 \neq C$	C	1	3

Выход D устанавливается в 1, если состояние одного из входных сигналов не совпадает с состоянием двух других сигналов. На выходе N формируется номер несовпадающего входного сигнала.

## 3.5.2 RS\_H – RS-триггер



Вх/Вых	Обозначение	Тип	Назначение
Вход	SET	Boolean	Если TRUE, устанавливает выход в TRUE
	RESET	Boolean	Если TRUE, сбрасывает выход в FALSE (доминанта)
Выход	Q	Boolean	Основной выход

**Назначение**

Функциональный блок RS\_H применяется для запоминания дискретных сигналов.

**Описание алгоритма**

Функциональный блок RS\_H реализует доминанту сброса. Выходной текущий дискретный сигнал  $Q_i$  в зависимости от состояния входных дискретных сигналов SET и RESET, и с учетом доминанты сброса устанавливается в следующее состояние:

SET	RESET	$Q_{i-1}$	$Q_i$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

$Q_{i-1}$  – состояние выхода в предыдущем цикле;

$Q_i$  – состояние выхода в текущем цикле.

## 3.5.3 SR\_H – SR-триггер



Вх/Вых	Обозначение	Тип	Назначение
Вход	SET	Boolean	Если TRUE, устанавливает выход в TRUE (доминанта)
	RESET	Boolean	Если TRUE, сбрасывает выход в FALSE
Выход	Q	Boolean	Основной выход

**Назначение**

Функциональный блок SR\_H применяется для запоминания дискретных сигналов.

**Описание алгоритма**

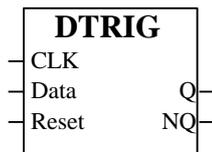
Функциональный блок SR\_H реализует доминанту установки. Выходной текущий дискретный сигнал  $Q_i$  в зависимости от состояния входных дискретных сигналов SET и RESET, и с учетом доминанты установки устанавливается в следующее состояние:

SET	RESET	$Q_{i-1}$	$Q_i$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$Q_{i-1}$  – состояние выхода в предыдущем цикле;

$Q_i$  – состояние выхода в текущем цикле.

## 3.5.4 DTRIG – D-триггер



Вх/Вых	Обозначение	Тип	Назначение
Входы	CLK	Boolean	Команда записи
	Data	Boolean	Входной сигнал
	Reset	Boolean	Команда сброса
Выходы	Q	Boolean	Основной выход
	NQ	Boolean	Инверсный выход

**Назначение**

Функциональный блок DTRIG используется для запоминания дискретного сигнала. Информация записывается по переднему фронту входного дискретного управляющего сигнала.

**Описание алгоритма**

По переднему фронту дискретного сигнала CLK (т.е. в момент перехода сигнала CLK из состояния логического 0 в состояние логической 1) информация на входе Data записывается, после чего алгоритм не реагирует на изменение сигнала на входе Data. Записанная информация передается на выход Q. По команде сброса (Reset=TRUE) триггер сбрасывается, выход Q=FALSE.

Инверсный выход NQ равен инвертированному значению основного выхода Q.

## 3.5.5 RF\_TRIG – R-триггер



Вх/Вых	Обозначение	Тип	Назначение
Вход	CLK	Boolean	Основной вход
Выход	Q	Boolean	Основной выход

**Назначение**

Функциональный блок RF\_TRIG применяется для выделения переднего или заднего фронта дискретного сигнала.

**Описание алгоритма**

Если на входе блока дискретный сигнал CLK изменяет свое состояние с логического 0 на логическую 1 (передний фронт), то на выходе алгоритма формируется сигнал Q=1 на время одного цикла работы контроллера. Остальное время Q=0.

Для выделения заднего фронта на входе блока устанавливается инверсия.

## 3.5.6 TON\_H – Таймер по фронту



Вх/Вых	Обозначение	Тип	Назначение
Входы	IN	Boolean	TRUE – запустить таймер. FALSE – сбросить таймер
	PT	Integer	Максимальное время отсчета (мсек.)
Выходы	Q	Boolean	TRUE – если время PT истекло
	ET	Integer	Текущее время, отсчитываемое таймером (мсек.)

**Назначение**

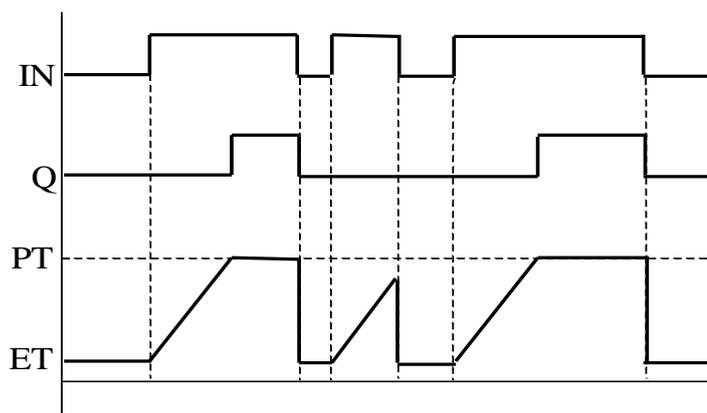
Функциональный блок TON\_H используется для задания временных задержек.

**Описание алгоритма**

Функциональный блок содержит внутренний таймер, который запускается по переднему фронту входного дискретного сигнала IN. На выходе ET формируется текущее время, отсчитываемое таймером от момента пуска. Если текущее время, отсчитываемое таймером, сравнивается со значением входа PT, отсчет времени прекращается. При этом устанавливается в 1 выход Q.

Текущее время	Выход Q
$ET < PT$	$Q = 0$
$ET \geq PT$	$Q = 1$

По заднему фронту сигнала IN отсчет времени прекращается и таймер сбрасывается, выход ET обнуляется. Работу блока поясняет временная диаграмма.



## 3.5.7 TOF\_H – Таймер по спаду



Вх/Вых	Обозначение	Тип	Назначение
Входы	IN	Boolean	TRUE – запустить таймер. FALSE – сбросить таймер
	PT	Integer	Максимальное время отсчета (мсек.)
Выходы	Q	Boolean	TRUE – если время PT не истекло
	ET	Integer	Текущее время, отсчитываемое таймером (мсек.)

**Назначение**

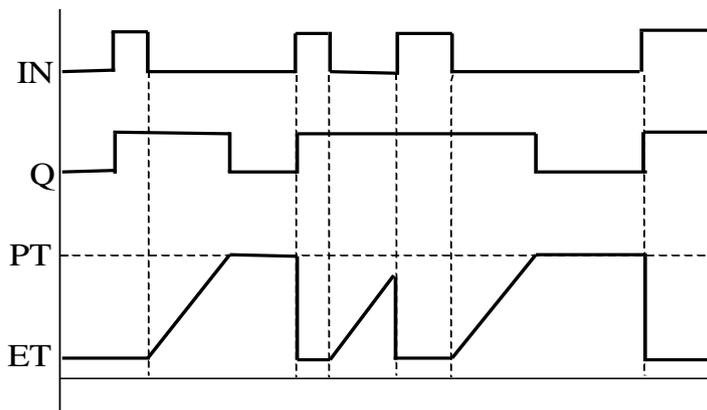
Функциональный блок TOF\_H используется для задания временных задержек.

**Описание алгоритма**

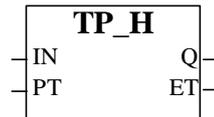
Функциональный блок содержит внутренний таймер, который запускается по заднему фронту входного дискретного сигнала IN. На выходе ET формируется текущее время, отсчитываемое таймером от момента пуска. Если текущее время, отсчитываемое таймером, сравняется со значением входа PT, отсчет времени прекращается. При этом выход Q=0.

Текущее время	Выход Q
$ET < PT$	Q=1
$ET \geq PT$	Q=0

По переднему фронту сигнала IN отсчет времени прекращается и таймер сбрасывается, выход ET обнуляется, выход Q=1. Работу блока поясняет временная диаграмма.



## 3.5.8 TP\_H – Таймер по уставке



Вх/Вых	Обозначение	Тип	Назначение
Входы	IN	Boolean	TRUE – запустить таймер. FALSE – сбросить таймер, если время PT истекло
	PT	Integer	Максимальное время отсчета (мсек.)
Выходы	Q	Boolean	TRUE – если таймер считает
	ET	Integer	Текущее время, отсчитываемое таймером

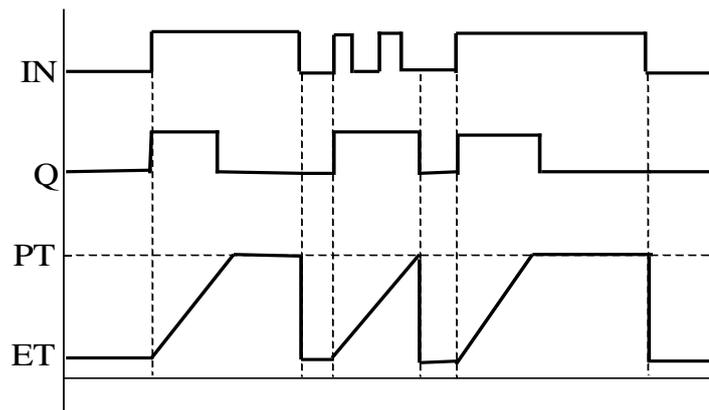
**Назначение**

Функциональный блок TP\_H используется для задания временных задержек.

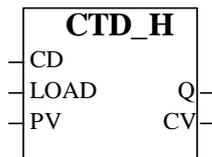
**Описание алгоритма**

Функциональный блок содержит внутренний таймер, который запускается по переднему фронту входного дискретного сигнала IN. На выходе ET формируется текущее время, отсчитываемое таймером от момента пуска. Во время счета алгоритм не реагирует на изменения сигнала IN. Если текущее время, отсчитываемое таймером, сравнивается со значением входа PT, отсчет времени прекращается, таймер останавливается. Сброс таймера происходит по заднему фронту сигнала IN, но только в том случае, если отсчитываемое таймером время достигло значения PT.

Выход Q устанавливается в 1, когда таймер начинает считать. Если таймер остановлен, выход Q равен логическому нулю. Работу блока поясняет временная диаграмма.



## 3.5.9 CTD\_H – Счетчик вниз



Вх/Вых	Обозначение	Тип	Назначение
Входы	CD	Boolean	Команда “считать”
	LOAD	Boolean	Команда “загрузить”
	PV	Integer	Начальное значение
Выходы	Q	Boolean	Переполнение счетчика
	CV	Integer	Результат счета

**Назначение**

Функциональный блок CTD\_H представляет собой счетчик, содержимое которого можно уменьшать на 1 в каждом цикле работы контроллера. Совместно с блоком RF\_TRIG блок CTD\_H используется для подсчета числа дискретных событий (переход из состояния логического 0 в состояние логической 1 или обратных переходов).

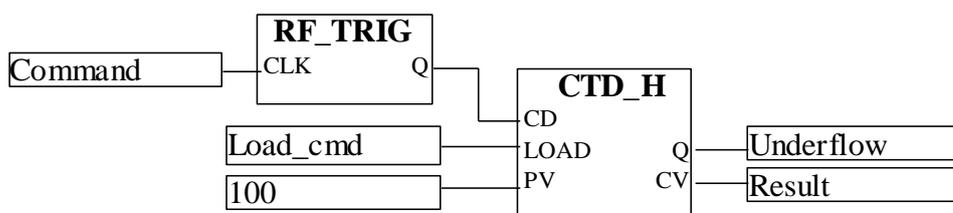
**Описание алгоритма**

При состоянии входа CD=TRUE содержимое счетчика уменьшается до 0. Содержимое счетчика уменьшается на 1 в каждом цикле работы контроллера. Выход CV равен результату счета в текущем цикле. Если CD=FALSE содержимое счетчика не изменяется.

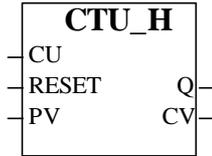
По команде “загрузить” (LOAD=TRUE) в счетчик записывается начальное значение PV. До тех пор, пока LOAD=TRUE, выход CV=PВ.

Выход Q принимает значение TRUE в тот момент, когда произошло переполнение счетчика, т.е. результат счета равен  $CV \leq 0$ .

Примечание: блок CTD\_H не определяет передний и задний фронты входа (CD). Для того чтобы создать импульсный счетчик, на вход CD необходимо подавать сигнал с выхода блока RF\_TRIG, который выделяет передний и задний фронты дискретного сигнала. На рисунке приведен пример соединения блоков. Содержимое счетчика будет уменьшаться на 1 по переднему фронту сигнала Command. Если установить инверсию на входе CLK блока RF\_TRIG, содержимое счетчика будет уменьшаться по заднему фронту сигнала Command.



3.5.10 CTU\_H – Счетчик вверх



Вх/Вых	Обозначение	Тип	Назначение
Входы	CU	Boolean	Команда “считать”
	RESET	Boolean	Команда “сброс”
	PV	Integer	Максимальное значение
Выходы	Q	Boolean	Переполнение счетчика
	CV	Integer	Результат счета

**Назначение**

Функциональный блок CTU\_H представляет собой счетчик, содержимое которого можно увеличивать на 1 в каждом цикле работы контроллера. Совместно с блоком RF\_TRIG блок CTD\_H используется для подсчета числа дискретных событий (переход из состояния логического 0 в состояние логической 1 или обратных переходов).

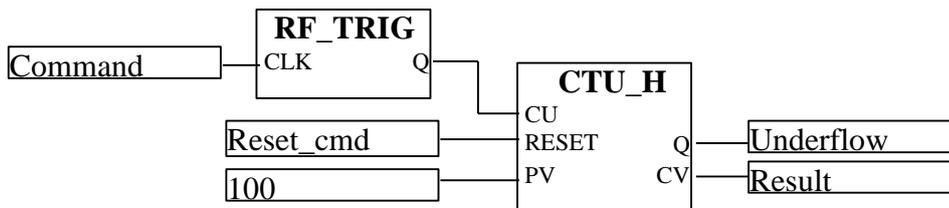
**Описание алгоритма**

При состоянии входа CU=TRUE содержимое счетчика увеличивается до максимального значения PV. Содержимое счетчика увеличивается на 1 в каждом цикле работы контроллера. Выход CV равен результату счета в текущем цикле. Если CU=TRUE содержимое счетчика не изменяется.

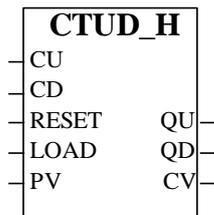
По команде “сброс” (RESET=TRUE) содержимое счетчика обнуляется. До тех пор, пока RESET=TRUE, выход CV=0.

Выход Q принимает значение TRUE в тот момент, когда произошло переполнение счетчика, т.е. результат счета равен максимальному значению PV или  $PV \leq 0$ .

Примечание: блок CTU\_H не определяет передний и задний фронты входа (CU). Для того чтобы создать импульсный счетчик, на вход CU необходимо подавать сигнал с выхода блока RF\_TRIG, который выделяет передний и задний фронты дискретного сигнала. На рисунке приведен пример соединения блоков. Содержимое счетчика будет увеличиваться на 1 по переднему фронту сигнала Command. Если установить инверсию на входе CLK блока RF\_TRIG, содержимое счетчика будет увеличиваться по заднему фронту сигнала Command.



## 3.5.11 CTUD\_H – Счетчик вверх-вниз



Вх/Вых	Обозначение	Тип	Назначение
Вход	CU	Boolean	Команда на увеличение содержимого счетчика
	CD	Boolean	Команда на уменьшение содержимого счетчика
	RESET	Boolean	Команда "сброс"
	LOAD	Boolean	Команда "загрузить"
	PV	Integer	Максимальное значение
Выход	QU	Boolean	Переполнение счетчика. CV=PV
	QD	Boolean	Переполнение счетчика. CV=0
	CV	Integer	Результат счета

**Назначение**

Функциональный блок CTUD\_H представляет собой счетчик, содержимое которого можно уменьшать или увеличивать на 1 в каждом цикле работы контроллера. Совместно с блоком RF\_TRIG блок CTUD\_H используется для подсчета числа дискретных событий (переход из состояния логического 0 в состояние логической 1 или обратных переходов).

**Описание алгоритма**

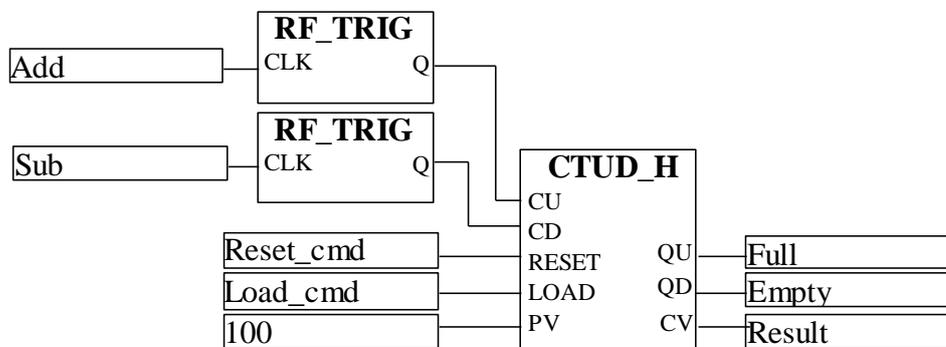
При состоянии входа CU=TRUE содержимое счетчика увеличивается до максимального значения PV, при состоянии входа CD=TRUE уменьшается до 0. Содержимое счетчика увеличивается или уменьшается на 1 в каждом цикле работы контроллера. Выход CV равен результату счета в текущем цикле. Если CU=CD содержимое счетчика не изменяется.

Работа счетчика разрешается, если отсутствуют сигналы "загрузить" и "сброс". По команде "загрузить" (LOAD=TRUE) в счетчик записывается значение PV. По команде "сброс" (RESET=TRUE) содержимое счетчика обнуляется, если нет команды "загрузить".

Выход QU принимает значение TRUE в тот момент, когда произошло переполнение счетчика и результат счета равен максимальному значению PV или  $PV \leq 0$ .

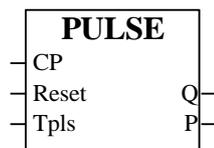
Выход QD принимает значение TRUE в тот момент, когда произошло переполнение счетчика и результат счета равен  $CV \leq 0$ .

Примечание: блок CTUD\_H не определяет передний и задний фронты входов CU и CD. Для того чтобы создать импульсный счетчик, на входы CU и CD необходимо подавать сигнал с выходов двух блоков RF\_TRIG, которые будут выделять передний и задний фронты дискретного сигнала. На рисунке приведен пример соединения блоков. Содержимое счетчика будет увеличиваться или уменьшаться на 1 по переднему фронту сигналов Add или Sub соответственно. Если установить инверсию на входы CLK блоков RF\_TRIG, содержимое счетчика будет изменяться по заднему фронту сигналов Add или Sub.



### 3.6 ПРОГРАММНОЕ УПРАВЛЕНИЕ

#### 3.6.1 PULSE – Одновибратор



Вх/Вых	Обозначение	Тип	Назначение
Вход	CP	Boolean	Команда "пуск" по фронту
	Reset	Boolean	Команда "сброс"
	Tpls	Integer	Длительность импульса (мсек.)
Выход	Q	Boolean	Основной выход
	P	Integer	Текущее время импульса (мсек.)

#### Назначение

Функциональный блок PULSE применяется в тех случаях, когда необходимо сформировать одиночный импульс заданной длительности.

#### Описание алгоритма

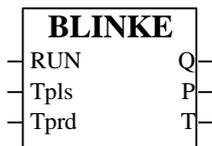
Алгоритм запускается по переднему фронту сигнала на входе CP (пуск), т.е. когда на входе CP дискретный сигнал переходит из состояния лог.0 в состояние лог.1. Перед пуском выходной дискретный сигнал Q отсутствует. После пуска появляется сигнал на выходе Q, причем этот сигнал находится в состоянии лог.1 в течение времени  $P=Tpls$ , где Tpls - параметр настройки. По истечении времени Tpls сигнал на выходе вновь переходит в нулевое состояние, после чего алгоритм можно вновь запустить.

На выходе P формируется текущее время, отсчитываемое от момента пуска. После отработки импульса  $P=0$ .

Сигнал на входе Reset (сброс) в любой момент времени обнуляет оба выхода. При наличии команды "сброс" алгоритм не может быть запущен а также не может быть повторно запущен командой "пуск" до тех пор, пока не закончится формирование выходного импульса.

Примечание: Для запуска по заднему фронту, сигнал на входе CP инвертируется.

3.6.2 BLINKE – Мультивибратор



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы: TRUE – пуск, FALSE – стоп
	Tpls	Integer	Длительность импульса (мсек.)
	Tprd	Integer	Период следования импульсов (мсек.)
Выход	Q	Boolean	Основной выход
	P	Integer	Текущее время импульса (мсек.)
	T	Integer	Текущее время периода (мсек.)

**Назначение**

Функциональный блок BLINKE применяется для периодического включения оборудования (двигателя, нагревателя, обеспечения мигающей сигнализации и т.п.).

**Описание алгоритма**

Алгоритм формирует последовательность импульсов с заданной длительностью и периодом следования. При состоянии входа RUN=TRUE с выхода блока Q формируются импульсы с длительностью Tpls и периодом следования Tprd. Фактическая длительность импульсов, также как и период следования кратны циклу контроллера.

Состояние основного выхода Q в режиме пуска (RUN=TRUE) при различных значениях Tpls и Tprd определяется таблицей:

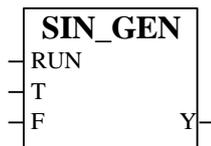
Tpls	Tprd	Выход Q
$Tpls > 0$	$Tprd > T_c$	<u>  </u> <u>  </u> <u>  </u> <u>  </u> <u>  </u> <u>  </u>
$Tpls \leq 0$	Безразлично	0
$Tpls > 0$	$Tprd \leq T_c$	1

$T_c$  – время цикла контроллера.

На выходе P формируется время, отсчитываемое от переднего фронта импульса. В следующем такте после отработки импульса P=0. На выходе T формируется текущее время, отсчитываемое от начала нового периода. Если заданная длительность периода меньше времени цикла контроллера  $Tprd < T_c$ , то выход T равен нулю.

При переходе входного сигнала RUN в состояние FALSE последовательность импульсов прерывается, выход Q равен логическому нулю. Выходы P и T обнуляются.

## 3.6.3 SIN\_GEN – Генератор синусоидального сигнала



Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Boolean	Режим работы
	T	Real	Период колебаний (сек.)
	F	Real	Фаза колебаний (рад.)
Выход	Y	Real	Основной выход

**Назначение**

Функциональный блок SIN\_GEN применяется для генерации синусоидальных сигналов при моделировании и настройке контуров регулирования.

**Описание алгоритма**

Функциональный блок содержит генератор синусоидального сигнала с периодом T и фазой F. Фаза F задается в радианах,  $F=0 - 2\pi$ .

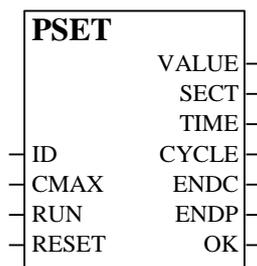
На выходе Y формируется сигнал по закону

$$Y = \sin(\omega \cdot t + \varphi) = \sin\left(\frac{2 \cdot \pi \cdot t}{T} + F\right),$$

Текущее время t изменяется дискретно по закону  $t=n \cdot T_c$ , где  $T_c$ -время цикла контроллера. Период выходного сигнала T приводится к значению, кратному  $T_c$ .

Генерация выходного сигнала начинается при установке входного сигнала RUN в состояние TRUE. При состоянии RUN=FALSE генерация синусоидального сигнала прекращается, выходной сигнал принимает значение Y=0.

## 3.6.4 PSET - Программный задатчик



Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор массива
	CMAX	Integer	Число повторений программы (устанавливается при сбросе)
	RUN	Boolean	Пуск задатчика
	RESET	Boolean	Сброс задатчика
Выход	VALUE	Real	Основной выход задатчика
	SECT	Integer	Номер текущего участка
	TIME	Integer	Время, оставшееся до конца текущего участка (мсек)
	CYCLE	Integer	Оставшееся число повторений
	ENDC	Boolean	Конец текущего повторения программы
	ENDP	Boolean	Конец программы
	OK	Integer	Результат операции

**Назначение**

Программный задатчик формирует кусочно-линейную функцию времени, состоящую из отрезков. Конечная ордината и продолжительность во времени каждого отрезка задаются в двумерном массиве. Идентификатор массива подается на вход ID. Начальная ордината задается нулевым элементом массива.

Массив должен быть предварительно открыт функцией AR\_OPEN.

**Описание алгоритма.**

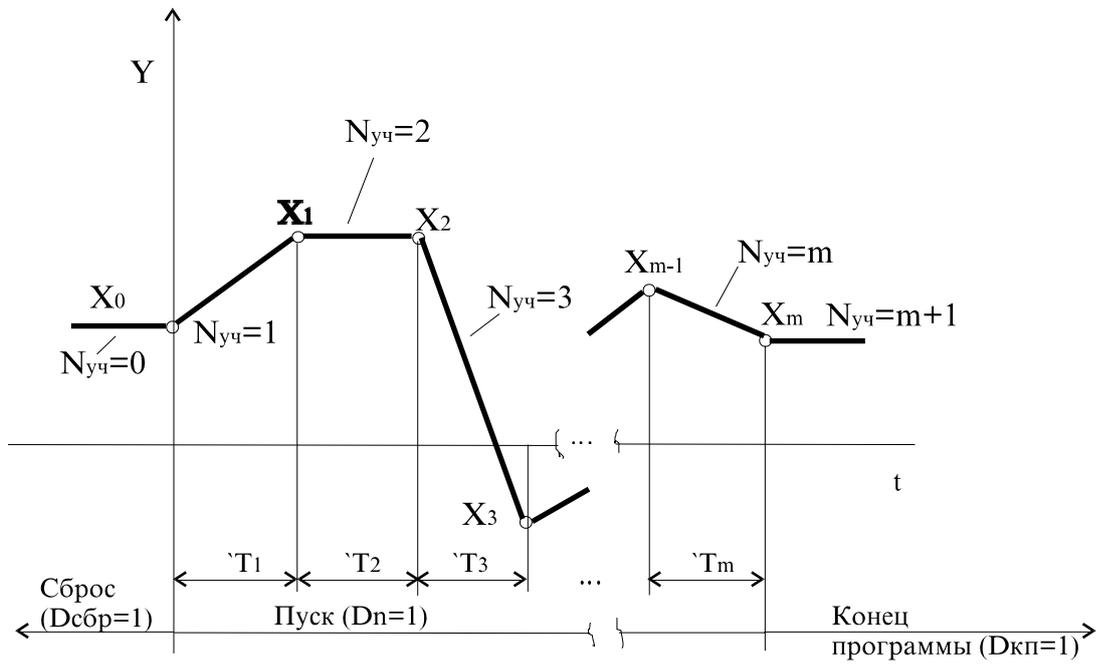
В состоянии сброса значение на выходе VALUE равно начальной ординате. После пуска значение на выходе VALUE начинает изменяться в соответствии с заданной программой.

Вход RESET является приоритетным по отношению к RUN.

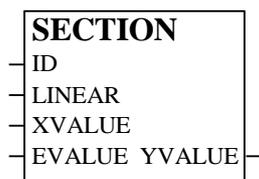
После выполнения последнего участка программа переходит в состояние "конец программы", при этом выходное значение VALUE замораживается. Если задано количество повторов CMAX =1, программа выполняется один раз, после чего переходит в состояние "конец программы".

При CMAX >1 программа, дойдя до конца, автоматически переходит в начало, оставаясь в состоянии пуска до тех пор, пока не закончится заданное число повторений. После каждого окончания программы выход ENDC (конец повторения) на один цикл переходит в состояние TRUE и вновь возвращается в состояние лог.0. После того, как все повторения будут выполнены, программа переходит в состояние "конец программы". При этом выход ENDP становится равным TRUE и остается в этом состоянии до тех пор, пока программа не будет сброшена.

Сигнал ,формируемый программным задатчиком



## 3.6.5 SECTION - Кусочно-линейная функция



Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор массива
	LINEAR	Boolean	Признак кусочно-линейной функции
	XVALUE	Real	Входное значение
	EVALUE	Real	Выходное значение в случае ошибки
Выход	YVALUE	Real	Выходное значение

**Назначение**

Функция SECTION вычисляет линейаризованное значение функции  $Y=f(X)$  методом кусочно-линейной аппроксимации с использованием узловых точек. Координаты концов отрезков определяются парами  $X_i, Y_i$  (абсцисса и ордината конца отрезка), которые задаются в массиве. Массив должен быть предварительно открыт функцией AR\_OPEN.

Линейаризованное значение  $Y$  вычисляется по следующей формуле:

$$Y = Y_{i-1} + ((Y_i - Y_{i-1}) / (X_i - X_{i-1})) * (X - X_{i-1})$$

где:  $X_i, Y_i$  – значения в узлах аппроксимации.

При нулевом значении на входе LINEAR формируется кусочно-ступенчатая функция.

### 3.7 ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ БИБЛИОТЕКИ

#### 3.7.1 Регулирование с каскадным и локальным управлением

На рисунке приведен пример программы на языке FBD. В программе используются алгоритмы регулирования RAN, RIM, SET, CCTRL, LCTRL, PWM. Функциональные блоки RAN – аналоговый регулятор, блок RIM – импульсный регулятор. Аналоговый и импульсный регуляторы включены по каскадной схеме, где ведущим является аналоговый регулятор, а ведомым импульсный регулятор.

##### Назначение переменных.

Название	Тип	Назначение
A_cc	Real	Скорость изменения выхода блока CCTRL (при ручном каскадном режиме)
AM_max	Real	Крайнее “правое” положение ИМ
AM_min	Real	Крайнее “левое” положение ИМ
Auto_mode	Boolean	TRUE – автоматический режим работы каскада
Auto_pwm	Boolean	FALSE – ручной режим работы блока PWM
Cascad	Boolean	TRUE – ручное управление аналоговым регулятором FALSE – ручное управление импульсным регулятором
Cycle_av	Integer	Период отсчетов (для блока AVRGD)
Cycle_ran	Integer	Цикл работы аналогового регулятора
Cycle_rim	Integer	Цикл работы импульсного регулятора. Период следования импульсов.
DB_ran	Real	Зона нечувствительности аналогового регулятора
DB_rim	Real	Зона нечувствительности импульсного регулятора
Dmax_ran	Boolean	TRUE – признак ограничения по максимуму выхода аналогового регулятора
Dmin_ran	Boolean	TRUE – признак ограничения по минимуму выхода аналогового регулятора
Down_cc	Boolean	TRUE – уменьшить значение выхода блока CCTRL (при ручном каскадном режиме)
Down_lc	Boolean	TRUE – уменьшить значение выхода блока LCTRL (при ручном локальном режиме)
Down_pwm	Boolean	Переместить ИМ в сторону “меньше” (при ручном режиме работы блока PWM)
KD_ran	Real	Коэффициент дифференцирования аналогового регулятора
KD_rim	Real	Коэффициент дифференцирования импульсного регулятора
KM_ran	Real	Коэффициент масштабирования аналогового регулятора
KM_rim	Real	Коэффициент масштабирования импульсного регулятора
KP_ran	Real	Коэффициент пропорциональности аналогового регулятора
KP_rim	Real	Коэффициент пропорциональности импульсного регулятора
Ldown_pwm	Integer	Люфт “меньше”
Local	Boolean	TRUE – автоматический режим работы импульсного регулятора
Lup_pwm	Integer	Люфт “больше”
Manual	Boolean	TRUE – ручное управление ИМ
N_av	Integer	Количество отсчетов (для блока AVRGD)
S_lc	Real	Приращение выхода блока LCTRL (при ручном локальном режиме)
TD_ran	Real	Постоянная дифференцирования аналогового регулятора
TD_rim	Real	Постоянная дифференцирования импульсного регулятора
TF_ran	Real	Постоянная фильтра аналогового регулятора
TF_rim	Real	Постоянная фильтра импульсного регулятора
TI_ran	Real	Постоянная интегрирования аналогового регулятора
TI_rim	Real	Постоянная интегрирования импульсного регулятора
TM_rim	Real	Полное время хода ИМ
Tmin_pwm	Integer	Минимальная длительность импульса
Tmps_pwm	Integer	Минимальная длительность паузы между импульсами
Up_cc	Boolean	TRUE – увеличить значение выхода блока CCTRL (при ручном каскадном режиме)
Up_lc	Boolean	TRUE – увеличить значение выхода блока LCTRL (при ручном локальном режиме)
Up_pwm	Boolean	Переместить ИМ в сторону “больше” (при ручном режиме работы блока PWM)
V_cc	Real	Ускорение выходного сигнала блока CCTRL (при ручном каскадном режиме)

V_set	Real	Ограничение скорости изменения задания для аналогового регулятора
X_set	Real	Каскадное задание
XC	Real	Каскадный регулируемый параметр (показания датчика)
XL	Real	Локальный регулируемый параметр (показания датчика)
XL_max	Real	Максимальное допустимое значение локального регулируемого параметра
XL_min	Real	Минимальное допустимое значение локального регулируемого параметра
Y_am	Real	Указатель положения ИМ (показания датчика)
Y_ran	Real	Выход аналогового регулятора
Y_rf	Real	Усредненное значение выхода аналогового регулятора. Задание для импульсного регулятора.
Y_rim	Real	Выход импульсного регулятора. Сквозность импульсов (%).
Y_set	Real	Выход блока SET. Задание для аналогового регулятора.
YDpls	Integer	Длительность импульсов выхода “меньше” (для импульсного вывода)
YDprd	Integer	Период следования импульсов выхода “меньше” (для импульсного вывода)
Ye_ran	Real	Ошибка аналогового регулятора
Ye_rim	Real	Ошибка импульсного регулятора
YUpls	Integer	Длительность импульсов выхода “больше” (для импульсного вывода)
YUprd	Integer	Период следования импульсов выхода “больше” (для импульсного вывода)
ZD_cc	Boolean	TRUE – запретить изменение значения выхода блока CCTRL в меньшую сторону (при ручном каскадном режиме работы)
ZD_lc	Boolean	TRUE – запретить отрицательное значение выхода блока LCTRL (при ручном локальном режиме работы)
ZD_pwm	Boolean	TRUE – запретить формирование импульсов с выхода “меньше” (при любом режиме работы)
ZD_ran	Boolean	TRUE – запретить изменение значения выхода аналогового регулятора в меньшую сторону (при любом режиме работы)
ZdT_pwm	Boolean	TRUE – запретить накопление минимального импульса
ZU_cc	Boolean	TRUE – запретить изменение значения выхода блока CCTRL в большую сторону (при ручном каскадном режиме работы)
ZU_lc	Boolean	TRUE – запретить положительное значение выхода блока LCTRL (при ручном локальном режиме работы)
ZU_pwm	Boolean	TRUE – запретить формирование импульсов с выхода “больше” (при любом режиме работы)
ZU_ran	Boolean	TRUE – запретить изменение значения выхода аналогового регулятора в большую сторону (при любом режиме работы)

**Режимы работы каскада.** Для рассматриваемой схемы включения регуляторов предусмотрено четыре режима работы:

1. Автоматический режим работы каскада. Этот режим характеризуется состоянием Auto\_mode=TRUE. Все блоки работают в автоматическом режиме.
2. Режим ручного управления аналоговым (ведущим) регулятором. При этом режиме Auto\_mode=FALSE, Cascad=TRUE. Аналоговый регулятор работает в ручном режиме, импульсный регулятор продолжает работать в автоматическом режиме.
3. Режим ручного управления импульсным (ведомым) регулятором. При этом режиме Auto\_mode=FALSE, Cascad=FALSE, Manual=FALSE. Оба регулятора работают в ручном режиме.
4. Режим ручного управления исполнительным механизмом. При этом режиме Auto\_mode=FALSE, Cascad=FALSE, Manual=TRUE. Оба регулятора и блок PWM работают в ручном режиме. При первых трех режимах работы каскада блок PWM работает в автоматическом режиме.

Ниже приведена таблица значений дискретных переменных для переключения на различные режимы работы каскада.

Режим работы каскада	1	2	3	4
Auto_mode	TRUE	FALSE	FALSE	FALSE
Cascad	*	TRUE	FALSE	FALSE
Manual	*	*	FALSE	TRUE

\* - значение безразлично.

**Автоматический режим.** В момент переключения на автоматический режим выход блока SET (Y\_set) равен регулируемому параметру XC. В последующих циклах работы контроллера значение Y\_set изменяется в соответствии с заданным ограничением скорости V\_set до тех пор, пока не достигнет значения X\_set. Входы аналогового регулятора X1=Y\_set, X2=XC. Блок AVRGD усредняет значение выхода аналогового

регулятора  $Y_{ran}$  за 10 циклов длительностью по 1 секунде каждый. Вход импульсного регулятора  $X1=Y_{rf}$ , т.е. усредненное значение выхода аналогового регулятора является заданием для импульсного регулятора. Вход  $X2$  импульсного регулятора равен локальному регулируемому параметру  $XL$ . Блок PWM преобразует выход импульсного регулятора  $Y_{rim}$  в длительность импульсов. В соответствии со знаком  $Y_{rim}$  импульсы формируются с выхода “больше” или “меньше” и ИМ перемещается в соответствующем направлении.

Выход аналогового регулятора ограничивается по нижнему пределу  $XL_{min}$  и верхнему пределу  $XL_{max}$  при любом режиме работы. Также для аналогового регулятора предусмотрен запрет изменения выхода по одному из направлений. Если  $ZU_{ran}=TRUE$  выход аналогового регулятора не увеличивается, если  $ZD_{ran}=TRUE$  не уменьшается. Запреты  $ZU_{ran}$  и  $ZD_{ran}$  действуют при любом режиме работы.

**Ручное управление аналоговым регулятором** ( $Local=TRUE$ ). Этот режим предусмотрен для ручного изменения локального задания (задания для импульсного регулятора). В зависимости от состояния дискретных переменных  $Up_{cc}$  и  $Down_{cc}$  значение выхода блока  $CCTRL$  увеличивается или уменьшается соответственно со скоростью  $V_{cc}$  и ускорением  $A_{cc}$ .

Выход аналогового регулятора при ручном режиме работы равен входу  $X0$ . Т.к.  $Auto\_mode=FALSE$ , то  $Y_{rf}=Y_{ran}$ . Таким образом, в режиме ручного управления аналоговым регулятором задание для импульсного регулятора  $Y_{rf}$  равно выходу блока  $CCTRL$  и изменяется в большую или меньшую сторону в зависимости от состояния дискретных переменных  $Up_{cc}$  и  $Down_{cc}$ . Если  $Up_{cc}=TRUE$  значение локального задания увеличивается, если  $Down_{cc}=TRUE$  уменьшается. Переменные  $ZU_{cc}$  и  $ZD_{cc}$  предназначены для запрета уменьшения или увеличения значения выхода блока  $CCTRL$ . Если  $ZU_{cc}=TRUE$  значение выхода не будет увеличиваться независимо от состояния  $Up_{cc}$ . Если  $ZD_{cc}=TRUE$  значение выхода уменьшаться не будет независимо от состояния  $Down_{cc}$ . Запреты  $ZU_{cc}$  и  $ZD_{cc}$  действуют только в режиме ручного управления аналоговым регулятором.

Выход блока  $SET$  равен  $Y_{set}=XC$ . Следовательно, входы  $X1$  и  $X2$  аналогового регулятора равны и ошибка  $Ye_{ran}=0$ .

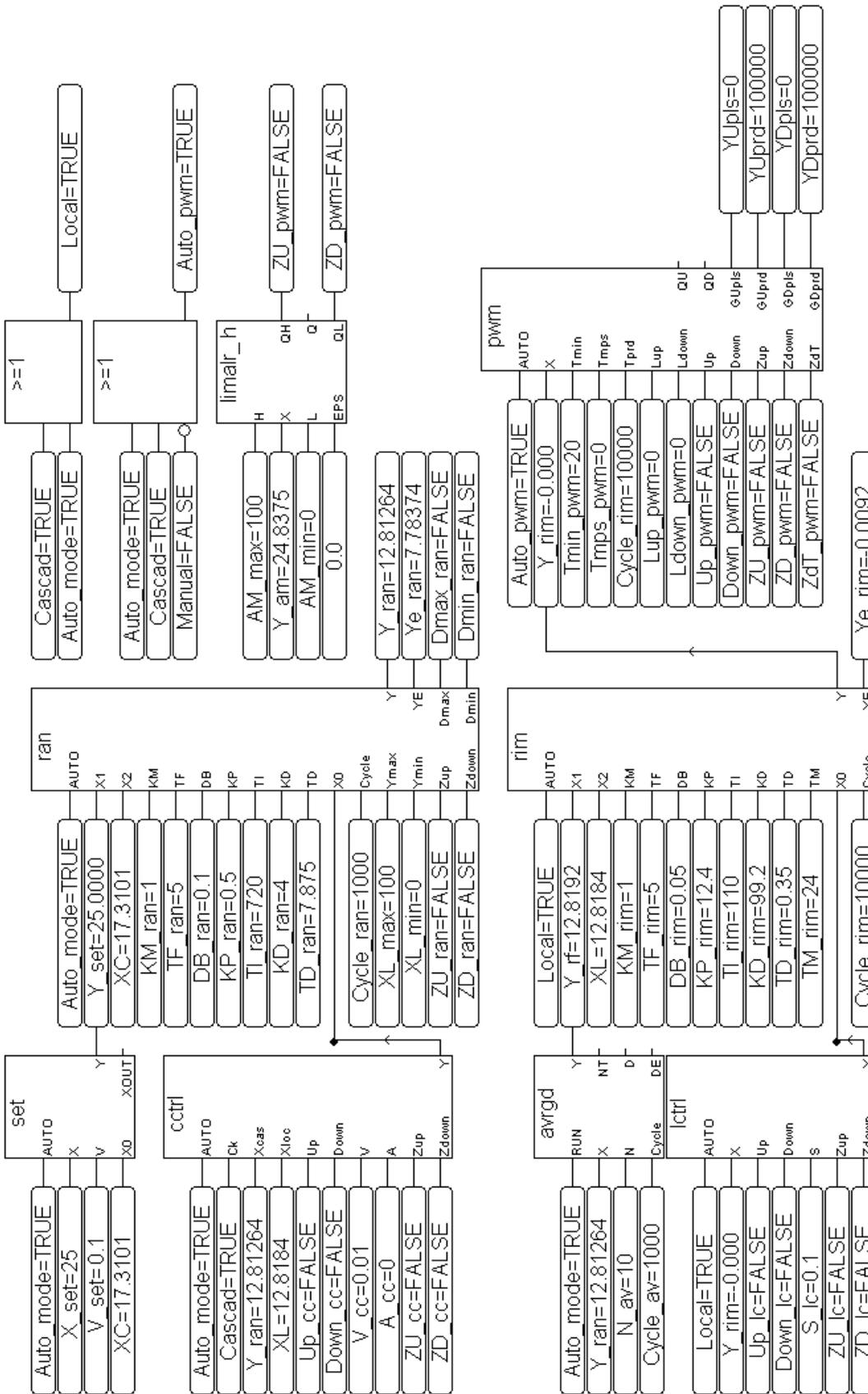
**Ручное управление импульсным регулятором** ( $Local=FALSE$ ). В этом режиме значение аналоговой переменной  $Y_{rim}$  (скважность импульсов в процентах) изменяется в зависимости от состояния дискретных переменных  $Up_{lc}$  и  $Down_{lc}$ . Абсолютное значение выхода блока  $LCTRL$  увеличивается на величину  $S_{lc}$  в каждом цикле работы контроллера. Знак выхода зависит от состояния дискретных переменных  $Up_{lc}$  и  $Down_{lc}$  в соответствии с таблицей. Если  $Up_{lc}=Down_{lc}$ , выход равен нулю.

$Up_{cc}$	TRUE	FALSE	TRUE	FALSE
$Down_{cc}$	FALSE	TRUE	TRUE	FALSE
Значение выхода	+	-	0	0

Выход импульсного регулятора при ручном режиме работы равен входу  $X0$ . Следовательно, увеличение абсолютного значения выхода блока  $LCTRL$  приводит к увеличению длительности импульсов в каждом цикле работы контроллера до тех пор, пока импульсы не слипнутся. Переменная  $ZU_{lc}$  предназначена для запрета положительного значения выхода блока  $LCTRL$ . Если  $ZU_{lc}=TRUE$  и  $Up_{lc}=TRUE$ , выход  $YUpls=0$ . Переменная  $ZD_{lc}$  предназначена для запрета отрицательного значения выхода. Если  $ZD_{lc}=TRUE$  и  $Down_{lc}=TRUE$ , выход  $YDpls=0$ . Запреты  $ZU_{lc}$  и  $ZD_{lc}$  действуют только при ручном локальном режиме.

Выход аналогового регулятора равен локальному регулируемому параметру  $XL$ . Следовательно, входы  $X1$  и  $X2$  импульсного регулятора равны и ошибка  $Ye_{rim}=0$ . Выход блока  $SET$  равен  $Y_{set}=XC$ . Ошибка  $Ye_{ran}=0$ .

**Ручное управление исполнительным механизмом** ( $Auto\_pwm=FALSE$ ). Этот режим работы предусмотрен для ручного управления исполнительным механизмом. В зависимости от состояния переменных  $Up_{pwm}$  и  $Down_{pwm}$  исполнительный механизм перемещается по направлению “больше” или “меньше” соответственно. Переменные  $ZU_{pwm}$  и  $ZD_{pwm}$  предназначены для запрета перемещения ИМ по одному из направлений. Если  $ZU_{pwm}=TRUE$ , выход “больше” постоянно равен нулю ( $YUpls=0$ ), следовательно, ИМ по направлению “больше” перемещаться не будет. Если  $ZD_{pwm}=TRUE$ , выход “меньше” постоянно равен нулю ( $YDpls=0$ ) и ИМ по направлению “меньше” перемещаться не будет. Поскольку дискретные переменные  $ZU_{pwm}$  и  $ZD_{pwm}$  равны соответствующим выходам блока  $LIMALR\_H$ , то при достижении ИМ положения  $Y_{am}=AM_{max}$  переменная  $ZU_{pwm}=TRUE$ , при достижении ИМ положения  $Y_{am}=AM_{min}$  переменная  $ZD_{pwm}=TRUE$ . Запреты  $ZU_{pwm}$  и  $ZD_{pwm}$  действуют при любом режиме работы.



### 3.7.2 Каскадное регулирование с локальным заданием

На рисунке приведен пример программы на языке FBD. В программе используется 2 блока SET для каскадного и локального задания.

**Назначение переменных.**

Название	Тип	Назначение
Auto_mode	Boolean	TRUE – автоматический режим работы каскада
Manual	Boolean	TRUE – ручное управление ИМ
Local	Boolean	FALSE – ручной режим работы блоков RIM и PWM
Lset	Boolean	TRUE – режим локального задания
ZU	Boolean	TRUE – запрет “больше”
ZD	Boolean	TRUE – запрет “меньше”
XL_set	Real	Локальное задание
VL_set	Real	Ограничение скорости изменения задания для импульсного регулятора
YL_set	Real	Задание для импульсного регулятора

Назначение остальных переменных см. в разделе 9.1.1.

**Режимы работы каскада.** Для рассматриваемой схемы предусмотрено 3 режима работы:

1. Автоматический режим.
2. Режим локального задания.
3. Режим ручного управления ИМ.

Ниже приведена таблица значений дискретных переменных для переключения на различные режимы работы каскада.

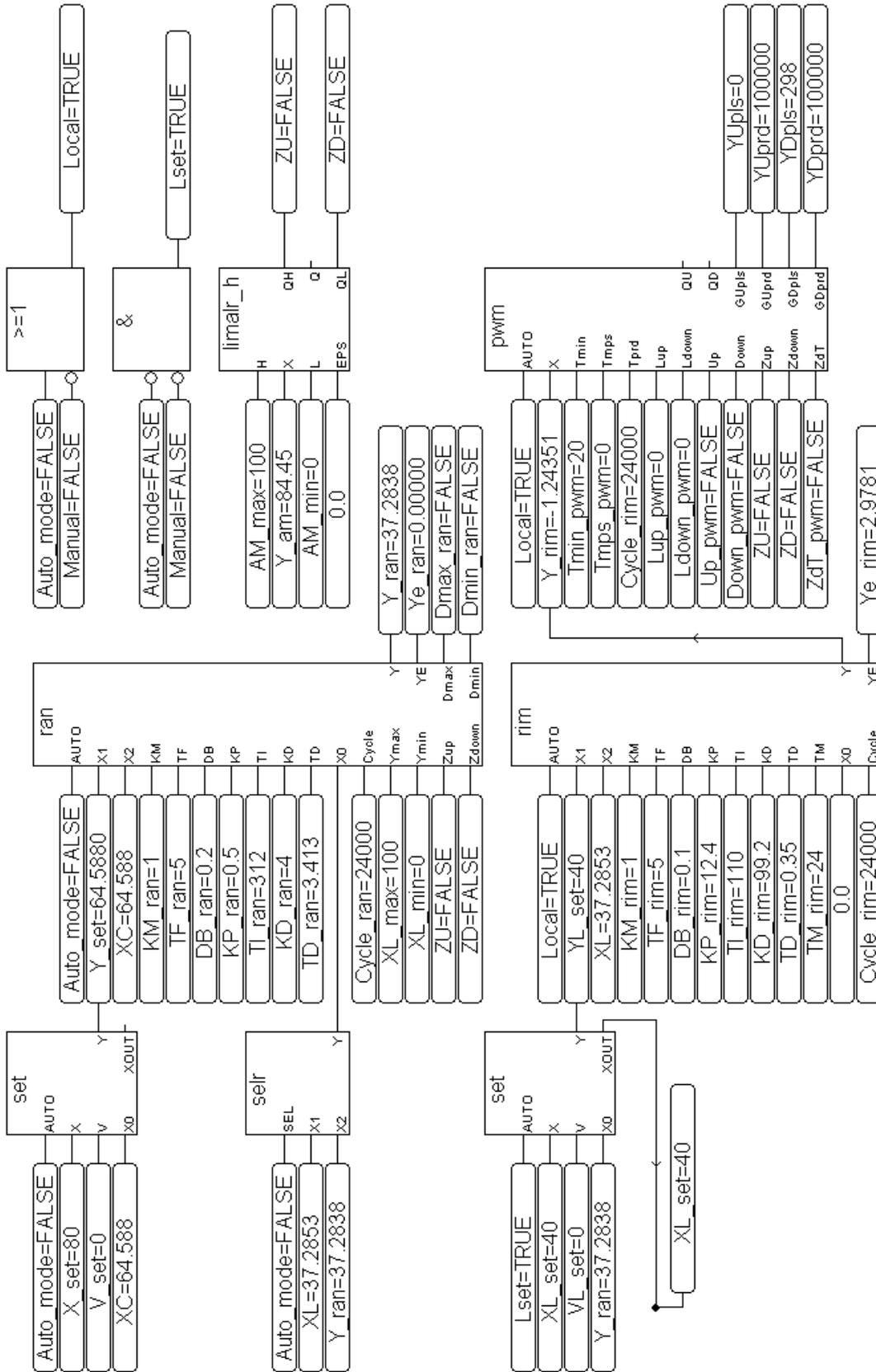
Режим работы каскада	1	2	3
Auto_mode	TRUE	FALSE	FALSE
Manual	*	FALSE	TRUE

\* - значение безразлично.

**Автоматический режим.** Этот режим аналогичен автоматическому режиму, рассмотренному в разделе 9.1.1. Задание для импульсного регулятора  $Y_{L\_set}=Y_{ran}$ . Запреты ZU и ZD действуют как для аналогового регулятора, так и для блока PWM. При достижении ИМ положения  $Y_{am}=AM_{min}$  дискретная переменная  $ZD=TRUE$ . При этом импульсы с выхода “меньше” не формируются, и выход аналогового регулятора не уменьшается. При достижении ИМ положения  $Y_{am}=AM_{max}$  дискретная переменная  $ZU=TRUE$ . При этом импульсы с выхода “больше” не формируются, и выход аналогового регулятора не увеличивается. Запреты ZU и ZD действуют при любом режиме работы. Выход аналогового регулятора ограничивается по нижнему пределу  $XL_{min}$  и верхнему пределу  $XL_{max}$  также при любом режиме работы.

**Локальное задание** ( $Lset=TRUE$ ). Этот режим предусмотрен для изменения локального задания. Для локального задания ( $XL_{set}$ ) предусмотрена статическая балансировка. При 1-м и 3-м режимах работы каскада локальное задание отслеживает выход аналогового регулятора, значение  $XL_{set}$  нельзя изменить вручную. В момент переключения на 2-ой режим  $XL_{set}$  запоминает последнее значение выхода аналогового регулятора  $Y_{ran}$  и не изменится, если его не изменить вручную. Импульсный регулятор и блок PWM работают в автоматическом режиме.

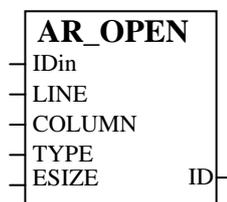
**Ручное управление исполнительным механизмом** ( $Auto\_pwm=FALSE$ ). Этот режим работы предусмотрен для ручного управления исполнительным механизмом. В зависимости от состояния переменных  $Up\_pwm$  и  $Down\_pwm$  исполнительный механизм перемещается по направлению “больше” или “меньше” соответственно.



## 4. БИБЛИОТЕКА ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ

### 4.1 ОПЕРАЦИИ С МАССИВАМИ

#### 4.1.1 AR\_OPEN - Создание массива элементов



Вх/Вых	Обозначение	Тип	Назначение
Вход	IDin	Integer	Идентификатор массива (проверяется наличие массива)
	LINE	Integer	Количество строк
	COLUMN	Integer	Количество столбцов
	TYPE	Integer	Тип элементов: 1 – целое число 2 – вещественное число 3 – строка фиксированной длины (по умолчанию - 16 символов)
	ESIZE	Integer	Размер элемента (символов). Только для массива строк
Выход	ID	Integer	Идентификатор массива

#### Назначение

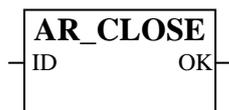
Функция AR\_OPEN используется для создания массива элементов. Максимальная размерность массива – двумерный. Для одномерного массива один из параметров LINE-COLUMN должен быть равен 1.

При значении входа IDin равно нулю – создается новый массив с уникальным идентификатором. В противном случае проверяется наличие открытого массива с идентификатором IDin.

Идентификатор созданного массива возвращается в выходном параметре ID. В случае ошибки – идентификатор ID равен нулю. Код ошибки уточняется вызовом функции ER\_TAKE.

При резервировании процессорной части контроллера имеется возможность зеркализации массивов, созданных через функцию ar\_open. Для включения этого механизма в поле Parameters платы m701e должны быть заданы следующие данные: ARn,s, где: n – количество зеркализируемых массивов, s – максимальный суммарный размер в килобайтах (правила задания параметров плат приводятся в документе “Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя”). При создании массива, предполагаемого для зеркализации, вход IDin должен быть установлен в ненулевое уникальное значение.

## 4.1.2 AR\_CLOSE - Закрытие массива

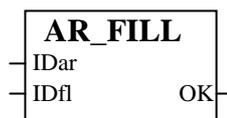


Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор массива
Выход	OK	Integer	Результат операции: 0 – выполнено успешно 1 – ошибка при операции с памятью 14 – массив не существует

**Назначение**

Функция AR\_CLOSE используется для закрытия массива элементов, созданного функцией AR\_OPEN.

## 4.1.3 AR\_FILL - Заполнение массива данными из файла



Вх/Вых	Обозначение	Тип	Назначение
Вход	IDar	Integer	Идентификатор массива
	IDfl	Integer	Идентификатор файла
Выход	OK	Integer	Результат операции: 0 – выполнено успешно 3 – недопустимые входные данные 5 – ошибка при копировании данных 7 – ошибка при установке текущей позиции в файле 8 – ошибка при чтении текущей позиции в файле 12 – файл не открыт 14 – массив не открыт 15 – ошибка при заполнении массива из файла 16 – размер прочитанных данных меньше размера массива

**Назначение**

Функция AR\_FILL используется для заполнения массива данными из файла.

Массив должен быть предварительно открыт функцией AR\_OPEN.

Файл должен быть предварительно открыт функцией FL\_OPEN.

Файл может иметь двоичный, либо текстовый формат. Количество и тип элементов, содержащихся в файле должно соответствовать параметрам массива. В случае текстового файла применяется преобразование. В качестве разделителей могут использоваться следующие символы: пробел, табуляция. Количество разделителей не ограничивается. Начало новой строки отмечают следующие символы: перевод каретки, перевод формата.

Операция с файлом, имеющим идентификатор IDfl, начинается с текущей позиции. По завершению операции чтения, текущая позиция в файле устанавливается на следующий байт после прочитанного блока данных.

## 4.1.4 AR\_SAVE - Сохранение массива данных в файле



Вх/Вых	Обозначение	Тип	Назначение
Вход	IDar	Integer	Идентификатор массива
	IDfl	Integer	Идентификатор файла
Выход	OK	Integer	Результат операции: 0 – выполнено успешно 3 – недопустимые входные данные 5 – ошибка при копировании данных 7 – ошибка при установке текущей позиции в файле 8 – ошибка при чтении текущей позиции в файле 12 – файл не открыт 14 – массив не открыт

**Назначение**

Функция AR\_SAVE используется для сохранения массива данных в файле.

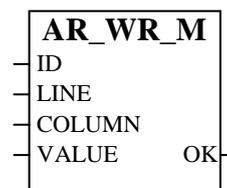
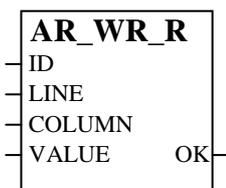
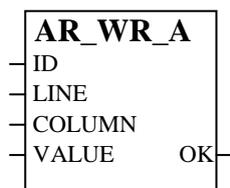
Массив должен быть предварительно открыт функцией AR\_OPEN.

Файл должен быть предварительно открыт функцией FL\_OPEN.

Файл может иметь двоичный, либо текстовый формат. В случае текстового файла применяется преобразование.

Операция с файлом, имеющим идентификатор IDfl, начинается с текущей позиции. По завершению операции записи, текущая позиция в файле устанавливается на следующий байт после записанного блока данных.

## 4.1.5 AR\_WR\_ - Запись элемента массива



Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор массива
	LINE	Integer	Номер строки элемента (нумерация строк начинается с 0)
	COLUMN	Integer	Номер столбца элемента (нумерация столбцов начинается с 0)
	VALUE	Integer Real Message	Значение элемента
Выход	OK	Integer	Результат операции: 0 – выполнено успешно 3 – недопустимые входные данные 5 – ошибка при копировании данных 14 – массив не открыт

**Назначение**

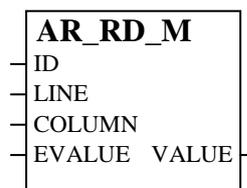
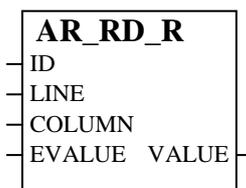
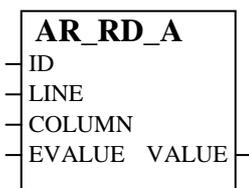
Функция AR\_WR\_A используется для записи значения в заданный элемент массива целых чисел.

Функция AR\_WR\_R используется для записи значения в заданный элемент массива вещественных чисел.

Функция AR\_WR\_M используется для записи значения в заданный элемент массива сообщений.

Массив должен быть предварительно открыт функцией AR\_OPEN.

4.1.6 AR\_RD\_ - Чтение элемента массива



Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор массива
	LINE	Integer	Номер строки элемента (нумерация строк начинается с 0)
	COLUMN	Integer	Номер столбца элемента (нумерация столбцов начинается с 0)
	EVALUE	Integer Real Message	Значение элемента в случае ошибки
Выход	VALUE	Integer Real Message	Значение элемента

**Назначение**

Функция AR\_RD\_A используется для чтения элемента массива целых чисел.

Функция AR\_RD\_R используется для чтения элемента массива вещественных чисел.

Функция AR\_RD\_M используется для чтения элемента массива сообщений.

Массив должен быть предварительно открыт функцией AR\_OPEN.

На вход EVALUE подается значение, которое будет на выходе функции при обнаружении ошибки. Код ошибки уточняется вызовом функции ER\_TAKE.

## 4.1.7 ER\_TAKE - Получение кода ошибки последней операции

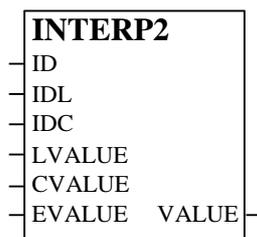


Вх/Вых	Обозначение	Тип	Назначение
Выход	OK	Integer	Код ошибки: 1 – ошибка при работе с памятью 2 – несоответствие атрибутов объекта 3 – недопустимые входные данные 4 – объект уже существует 5 – ошибка при копировании данных 6 – ошибка при открытии файла 7 – ошибка при установке текущей позиции в файле 8 – ошибка при чтении текущей позиции в файле 9 – ошибка при закрытии файла 10 – позиция чтения опережает позицию записи 11 – имя файла превышает 127 символов 12 – файл не открыт 14 – массив не открыт 15 – ошибка заполнения массива из файла 16 – обнаружен конец файла, но операция не завершена

**Назначение**

Функция ER\_TAKE используется для получения кода ошибки последней операции с массивом, открытым функцией AR\_OPEN, либо с файлом, открытым функцией FL\_OPEN.

## 4.1.8 INTERP2 - Интерполяция



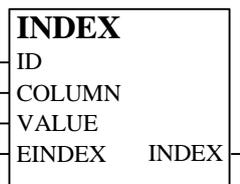
Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор массива-таблицы
	IDL	Integer	Идентификатор массива-вектора коэффициентов по строкам
	IDC	Integer	Идентификатор массива-вектора коэффициентов по столбцам
	LVALUE	Real	Входное значение для поиска индекса по строкам
	CVALUE	Real	Входное значение для поиска индекса по столбцам
	EVALUE	Real	Выходное значение интерполяции в случае ошибки
Выход	VALUE	Real	Выходное значение интерполяции

**Назначение**

Функция INTERP2 вычисляет интерполированное значение в таблице по 4-м точкам. Для поиска индексов по строкам и столбцам используются переменные, подаваемые на входы LVALUE и CVALUE. Если массивы-векторы IDC и IDL не заданы (индексы равны нулю), то поиск индексов осуществляется в нулевом столбце и нулевой строке таблицы, заданной индексом ID.

Массивы должны быть предварительно открыты функцией AR\_OPEN.

## 4.1.9 INDEX – Поиск индекса



Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор массива
	COLUMN	Boolean	Признак поиска в нулевом столбце таблицы
	VALUE	Real	Значение для поиска
	EINDEX	Integer	Значение индекса в случае ошибки
Выход	INDEX	Integer	Значение индекса

**Назначение**

Функция INDEX выполняет поиск индекса в массиве. Для поиска индекса используются переменная, подаваемая на вход VALUE. Если на вход COLUMN подано ненулевое значение, поиск выполняется в нулевом столбце массива-таблицы. Условие поиска:  $(X \geq X_{i-1}) \& (X < X_i)$

Массив должен быть предварительно открыт функцией AR\_OPEN.

## 4.2 ФАЙЛОВЫЕ ОПЕРАЦИИ

### 4.2.1 FL\_OPEN - Открытие файла



Вх/Вых	Обозначение	Тип	Назначение
Вход	IDin	Integer	Идентификатор файла (проверяется наличие открытого файла)
	PATH	Message	Полное имя файла
	RDONLY	Boolean	Файл открывается только для чтения
	BIN	Boolean	Файл имеет двоичный формат
	SIZE	Integer	Размер файла
Выход	ID	Integer	Идентификатор файла.

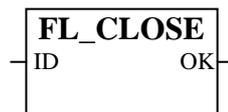
#### Назначение

Функция FL\_OPEN используется для открытия файла. При значении входа *IDin* равном нулю – открывается файл с уникальным идентификатором. В противном случае проверяется наличие открытого файла с идентификатором *IDin*. При задании полного имени файла используются символы слэш “/”.

Идентификатор открытого файла возвращается в выходном параметре *ID*. В случае ошибки – идентификатор *ID* равен нулю. Код ошибки уточняется вызовом функции ER\_TAKE.

После открытия файла текущая позиция устанавливается в начало файла. Позиция может быть явно установлена через вызов функции *fl\_spos*. В случае, если размер файла не задан (-1 на входе *SIZE*), существующий файл открывается для чтения и записи. В противном случае файл будет усечен до заданного размера. Если файл не существует – он будет создан.

## 4.2.2 FL\_CLOSE - Закрытие файла

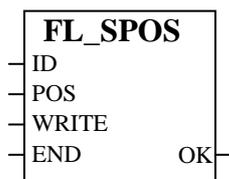


Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор файла
Выход	OK	Integer	Результат операции: 0 – выполнено успешно 9 – ошибка при закрытии файла 12 – файл не открыт

**Назначение**

Функция FL\_CLOSE используется для закрытия файла, открытого функцией FL\_OPEN.

4.2.3 FL\_SPOS - Установка текущей позиции в файле



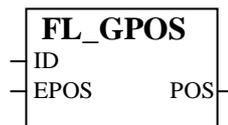
Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор файла
	POS	Integer	Текущая позиция в файле
	WRITE	Boolean	Признак позиции для записи в файл
	END	Boolean	Признак установки текущей позиции в конец файла
Выход	OK	Integer	Результат операции: 0 – выполнено успешно 7 – ошибка при установке текущей позиции в файле 12 – файл не открыт

**Назначение**

Функция FL\_SPOS используется для установки текущей позиции в файле, который был открыт функцией FL\_OPEN.

Позиции для чтения и записи различаются. При установке позиции необходимо указывать, какая именно позиция устанавливается.

#### 4.2.4 FL\_GPOS - Получение текущей позиции в файле



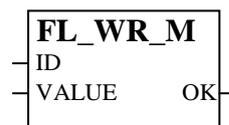
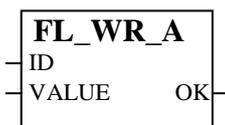
Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор файла
	EPOS	Integer	Значение позиции в случае ошибки
Выход	POS	Integer	Текущая позиция в файле

##### Назначение

Функция FL\_GPOS используется для получения текущей позиции в файле, который был открыт функцией FL\_OPEN.

На вход EPOS подается значение, которое будет на выходе функции при обнаружении ошибки. Код ошибки уточняется вызовом функции ER\_TAKE.

4.2.5 FL\_WR\_ - Запись в файл



Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор файла
	VALUE	Integer Real Message	Значение переменной
Выход	OK	Integer	Результат операции: 0 – выполнено успешно 3 – недопустимые входные данные 5 – ошибка при копировании данных 7 – ошибка при установке текущей позиции в файле 8 – ошибка при чтении текущей позиции в файле 12 – файл не открыт

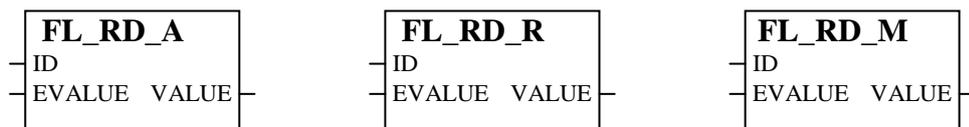
**Назначение**

Функция FL\_WR\_A используется для записи в файл переменной целого типа.  
 Функция FL\_WR\_R используется для записи в файл переменной вещественного типа.  
 Функция FL\_WR\_M используется для записи в файл строковой переменной.

Файл должен быть предварительно открыт функцией FL\_OPEN.

Запись в файл производится с текущей позиции для записи. По завершению операции текущая позиция для записи увеличивается на размер записанного блока данных.

## 4.2.6 FL\_RD\_ - Чтение из файла



Вх/Вых	Обозначение	Тип	Назначение
Вход	ID	Integer	Идентификатор файла
	EVALUE	Integer Real Message	Значение переменной в случае ошибки
Выход	VALUE	Integer Real Message	Значение переменной

**Назначение**

Функция FL\_RD\_A используется для чтения из файла переменной целого типа.

Функция FL\_RD\_R используется для чтения из файла переменной вещественного типа.

Функция FL\_RD\_M используется для чтения из файла строковой переменной.

Файл должен быть предварительно открыт функцией AR\_OPEN.

На вход EVALUE подается значение, которое будет на выходе функции при обнаружении ошибки. Код ошибки уточняется вызовом функции ER\_TAKE.

Чтение из файла производится с текущей позиции для чтения. По завершению операции текущая позиция для чтения увеличивается на размер прочитанного блока данных.

## 5. ФУНКЦИИ ВРЕМЕНИ

Целевая задача ISaGRAF поддерживает специальную систему времени на контроллере. Основные составляющие системы следующие:

- часы реального времени CMOS;
- система времени ОС QNX;
- поправка времени ISaGRAF, функции установки/получения системного времени.

*Поправка времени* представляет собой разницу между системным временем CMOS и временем, которое было установлено через функции ISaGRAF. При чтении системного времени контроллера через функции "C" ISaGRAF данная поправка учитывается. Поправка времени сохраняется в энергонезависимой памяти SRAM, и восстанавливается после перезагрузки контроллера. При загрузке контроллера до запуска приложения ISaGRAF производится корректировка времени CMOS с учетом данной поправки.

Необходимость в подобном механизме обусловлена тем, что корректировка времени непосредственно в CMOS приводит к сбою в работе всех программных объектов, которые используют ресурс времени: функциональные блоки, таймеры и т.п.

Механизм поправки времени ISaGRAF задействуется только при использовании следующих функциональных блоков "C": `tm_sync`, `settime(settime_)`, `gettime(gettime_)`.

### 5.1 УСТАНОВКА ВРЕМЕНИ

Порядок установки времени с использованием ФБ `settime (settime_)` следующий:

- на контроллере в определенный момент фиксируется текущее системное время CMOS;
- вычисляется разница между зафиксированным временем и тем, которое устанавливается;

На вход **KEEP** функционального блока подается 1 (переход 0 → 1), когда нужно зафиксировать текущее системное время CMOS. Предполагается использование аппаратной схемы, в которой задействован ПК и группа контроллеров. ПК подает дискретный сигнал, заведенный на входы **KEEP** всех контроллеров, а затем передает по каналу связи время, которое должно соответствовать времени подачи сигнала. Контроллеры вычисляют поправку времени. Если аппаратная схема отсутствует, на вход **KEEP** сигнал должен подаваться одновременно со входом **SET**.

На вход **SET** функционального блока подается 1 (переход 0 → 1), когда нужно вычислить поправку времени. На входах времени функционального блока должны быть корректные значения. После этого поправка времени вступает в силу.

На вход **RESET** функционального блока подается 1 (переход 0 → 1) для сброса поправки времени

**Важно!** Вход **CMOS** должен устанавливаться в значение 0. Если есть необходимость в записи времени непосредственно в CMOS следует в приложении ISaGRAF учитывать возможность сбоя в работе таймеров.

Система единого времени контроллеров TREI-5B-02 может быть также организована с использованием источника точного времени ИВЧ-1/СП. К общей шине на базе RS-422/RS-485 также могут подключаться ПК и прочие устройства, поддерживающие данный интерфейс с возможностью выключения передатчика. Предполагается, что только одно устройство(резервированное) на общей шине может запрашивать данные о времени от ИВЧ. Остальные устройства находятся в режиме готовности к приему данных от ИВЧ. Кроме того, ПК может по Ethernet(протокол IP UDP) запрашивать у контроллеров значение скорректированного системного времени. Формат ответа контроллера приводится в нотации языка "C":

```
struct T5B_DATE {unsigned long YY,MM,DD,hh,mm,ss,ms,er,dT;} T5B_DATE;
```

Назначение полей структуры следующее:

- **YY** - год;
- **MM** - месяц с начала года;
- **DD** - день с начала месяца;
- **hh** - час с начала суток;
- **mm** - минута с начала часа;
- **ss** - секунда с начала минуты;
- **ms** - миллисекунда с начала секунды;
- **er** - признак наличия ошибки (0 – нет ошибки);
- **dT** - поправка к системному времени контроллера (мсек).

Порядок установки единого времени контроллеров с использованием ИВЧ следующий:

- ПК по UDP запрашивает и анализирует текущее время контроллеров;
- ПК выдает по общей шине для ИВЧ команду-запрос "s";
- ИВЧ по запросу выдает по общей шине информацию о дате, текущем времени и состоянии часов;
- ПК и контроллеры принимают пакет от ИВЧ и вычисляют поправку к системному времени;
- далее поправка учитывается при запросе системного времени на контроллерах и ПК;

## 5.1.1 SETTIME\_ - Установка текущего времени

```

SETTIME_
- DAY
- MONTH
- YEAR
- HOUR
- MINUTE
- SECOND
- MSEC
- RESET
- KEEP
- SET
- CMOS      OK

```

Вх/Вых	Обозначение	Тип	Назначение
Вход	DAY	Integer	День с начала месяца
	MONTH	Integer	Месяц с начала года
	YEAR	Integer	Год
	HOUR	Integer	Час с начала суток
	MINUTE	Integer	Минута с начала часа
	SECOND	Integer	Секунда с начала минуты
	MSEC	Integer	Миллисекунда с начала секунды
	RESET	Integer	Сброс поправки времени
	KEEP	Integer	Фиксация текущего системного времени CMOS
	SET	Integer	Вычисление поправки времени
	CMOS	Integer	Запись времени в CMOS
Выход	OK	Integer	Результат операции

**Назначение**

Функциональный блок SETIME\_ используется для установки текущего времени и даты. Порядок использования функционального блока приводится в подразделе 5.1.

В случае ошибки при выполнении операции – выход OK не равен нулю.

## 5.1.2 SETTIME - Установка текущего времени (специальный формат)

SETTIME	
SEC	
MSEC	
RESET	
KEEP	
SET	
CMOS	OK

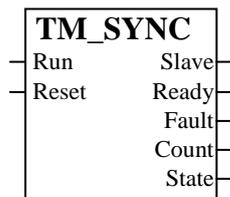
Вх/Вых	Обозначение	Тип	Назначение
Вход	SEC	Integer	Секунда с 1970г
	MSEC	Integer	Миллисекунда с начала секунды
	RESET	Integer	Сброс поправки времени
	KEEP	Integer	Фиксация текущего времени CMOS
	SET	Integer	Вычисление поправки времени
	CMOS	Integer	Запись времени в CMOS
Выход	OK	Integer	Результат операции

**Назначение**

Функциональный блок SETTIME используется для установки текущего времени и даты. Порядок использования функционального блока приводится в подразделе 5.1.

В случае ошибки при выполнении операции – выход OK не равен нулю.

## 5.1.3 TM\_SYNC - Синхронизация времени по ИВЧ-1



Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Синхронизация: TRUE – выполнить, FALSE – не выполнять
	Reset	Boolean	Сброс константы: TRUE – выполнить, FALSE – не выполнять
Выход	Slave	Boolean	Режим работы: TRUE – прием от ИВЧ, FALSE – запрос ИВЧ
	Ready	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 4 – данные не достоверны 138 – задача связи <i>tm_sync</i> не запущена на контроллере
	Count	Integer	Количество успешных тактов синхронизации времени
	State	Integer	Значение байта состояния ИВЧ

**Назначение**

Функциональный блок *TM\_SYNC* предназначен для управления режимом синхронизации времени по данным ИВЧ. При этом на контроллере должна быть запущена задача связи *tm\_sync* (правила запуска приводятся в документе “Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя”).

Выход *Slave* отображает текущий режим работы задачи связи.

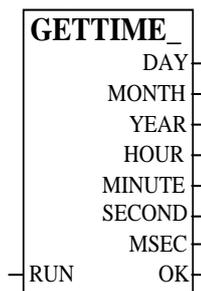
При запуске задачи связи с параметром *-master*, выход *Slave* имеет значение *False*. При этом вход *Run* управляет посылкой команды-запроса информации от ИВЧ. По завершению операции выход *Ready* устанавливается в состояние *True*. Перед очередным запросом значение на входе *Run* должно быть сброшено в *False*.

При запуске задачи связи без параметра *-master*, выход *Slave* имеет значение *True*. Контроллер может только принимать данные от ИВЧ. При этом вход *Run* разрешает/запрещает коррекцию времени по принимаемым по общей шине данным.

Поправка к системному времени контроллера может быть сброшена при установке входа *Reset* в состояние *True*.

## 5.2 ПОЛУЧЕНИЕ ВРЕМЕНИ

### 5.2.1 GETTIME\_ - Получение текущего времени



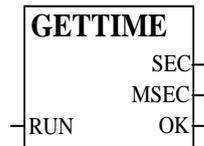
Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Integer	Признак необходимости обновления данных
Выход	DAY	Integer	День с начала месяца
	MONTH	Integer	Месяц с начала года
	YEAR	Integer	Год
	HOUR	Integer	Час с начала суток
	MINUTE	Integer	Минута с начала часа
	SECOND	Integer	Секунда с начала минуты
	MSEC	Integer	Миллисекунда с начала секунды
	OK	Integer	Результат операции

#### Назначение

Функциональный блок GETTIME\_ используется для получения текущего времени и даты. Системное время возвращается с учетом поправки, введенной через функциональные блоки *tm\_sync*, *settime* (*settime\_*). Операция выполняется при значении входа RUN не равном нулю.

В случае ошибки при выполнении операции – выход OK не равен нулю.

## 5.2.2 GETTIME - Получение текущего времени (специальный формат)



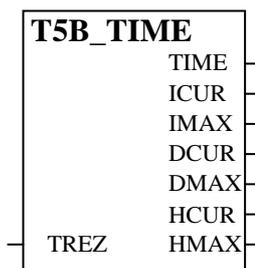
Вх/Вых	Обозначение	Тип	Назначение
Вход	RUN	Integer	Признак необходимости обновления данных
Выход	SEC	Integer	Секунда с 1970г
	MSEC	Integer	Миллисекунда с начала секунды
	OK	Integer	Результат операции

**Назначение**

Функциональный блок GETTIME используется для получения текущего времени. Системное время возвращается с учетом поправки, введенной через функциональные блоки *tm\_sync*, *settime* (*settime\_*). Операция выполняется при значении входа RUN не равном нулю.

В случае ошибки при выполнении операции – выход OK не равен нулю.

## 5.2.3 T5B\_TIME – Получение временных параметров ISaGRAF



Вх/Вых	Обозначение	Тип	Назначение
Вход	TREZ	Boolean	TRUE – сброс накоплений
Выход	TIME	Integer	Текущее время (сек) с 1970 года
	ICUR	Integer	Текущий цикл ISaGRAF (мсек)
	IMAX	Integer	Максимальный цикл ISaGRAF (мсек)
	DCUR	Integer	Текущий цикл драйвера I/O (мсек)
	DMAX	Integer	Максимальный цикл драйвера I/O (мсек)
	HCUR	Integer	Текущий цикл горячего резервирования (мсек)
	HMAX	Integer	Максимальный цикл горячего резервирования (мсек)

**Назначение**

Функциональный блок T5B\_TIME используется для чтения временных параметров ISaGRAF и драйвера ввода-вывода.

## 6. ФУНКЦИИ ПРЕОБРАЗОВАНИЯ

### 6.1 ПРЕОБРАЗОВАНИЯ ДЛЯ ТЕМПЕРАТУРЫ

Значение электрического сигнала, измеренное мезонином ввода терморпары, термосопротивления, преобразуется в значение температуры методом линеаризации с использованием таблиц. Также может быть выполнено обратное преобразование значения температуры в физическое значение. Данные преобразования выполняются посредством вызова функций *termosnv* и *termoinv*. Номер таблицы преобразования температуры, которая соответствует конкретному типу датчика, задается через параметр *NT*.

Далее приводится перечень таблиц преобразования. Применяются следующие обозначения:

- ТП - Терморпара;
- ТС - Термосопротивление.
- NT - Номер таблицы преобразования температуры;
- RANG - Диапазон измерения сигнала;
- POLN - Полярность сигнала (0 – биполярный, 1– униполярный).

Тип ТП/ТС	Электрический сигнал	NT	RANG	POLN
ТП S (ПП)	0÷19 mV	1	19	1
ТП B (ПР)	0÷19 mV	2	19	1
ТП J (ЖК)	±78 mV	3	78	0
ТП T (МК)	±78 mV	4	78	0
ТП E (ЖК)	±78 mV	5	78	0
ТП K (ХА)	±78 mV	6	78	0
ТП N (НН)	±78 mV	7	78	0
ТП L (ХК)	±78 mV	8	78	0
ТП А-1 (ВР)	0÷78 mV	9	78	1
ТП А-2 (ВР)	0÷78 mV	10	78	1
ТП А-3 (ВР)	0÷78 mV	11	78	1
ТС 50 П ( $W_{100}=1.3910$ )	250 Ом	12	250	1
ТС 100 П ( $W_{100}=1.3910$ )	500 Ом	13	500	1
ТС 50 М ( $W_{100}=1.4280$ )	100 Ом	14	100	1
ТС 100 М ( $W_{100}=1.4280$ )	200 Ом	15	200	1
ТС 21	250 Ом	16	250	1
ТС 23	100 Ом	17	100	1
ТС 50 ПТ ( $W_{100}=1.3910$ )	65 Ом	18	65	1
ТС 100 ПТ ( $W_{100}=1.3910$ )	130 Ом	19	130	1
ТС 100 Н	250 Ом	20	250	1
ТС 50 ПВ ( $W_{100}=1.3910$ )	125 Ом	21	125	1
ТС 100 ПВ ( $W_{100}=1.3910$ )	250 Ом	22	250	1
ТС 50 ПА ( $W_{100}=1.3850$ )	250 Ом	23	250	1
ТС 100 ПА ( $W_{100}=1.3850$ )	500 Ом	24	500	1
ТС 50 МА ( $W_{100}=1.4260$ )	100 Ом	25	100	1
ТС 100 МА ( $W_{100}=1.4260$ )	200 Ом	26	200	1
ТС 50 ПТА ( $W_{100}=1.3850$ )	65 Ом	27	65	1
ТС 100 ПТА ( $W_{100}=1.3850$ )	130 Ом	28	130	1
ТС 50 ПВА ( $W_{100}=1.3850$ )	125 Ом	29	125	1
ТС 100 ПВА ( $W_{100}=1.3850$ )	250 Ом	30	250	1

## 6.1.1 TERMOCNV – Преобразование физической величины в температуру



Вх/Вых	Обозначение	Тип	Назначение
Вход	NT	Integer	Номер таблицы преобразования температуры
	X	Real	Значение физической величины (mV/Om)
Выход	Y	Real	Значение температуры (ГрадС)

**Назначение**

Функция используется для преобразования электрической величины в температуру для различных типов термодатчиков (ТП) и термосопротивлений (ТС) TREI-5B.

Перечень таблиц преобразования приводится в подразделе 6.1.

6.1.2 TERMOINV – Преобразование температуры в физическую величину



Вх/Вых	Обозначение	Тип	Назначение
Вход	NT	Integer	Номер таблицы преобразования температуры
	X	Real	Значение температуры (ГрадС)
Выход	Y	Real	Значение физической величины (mV/Om)

**Назначение**

Функция используется для обратного преобразования температуры в электрическую величину для различных типов термпар (ТП) и термосопротивлений (ТС) TREI-5B. Данная операция может быть использована для компенсации температуры холодного спая термпары. При этом температура холодного спая, измеренная термосопротивлением, должна быть преобразована в физическую величину по таблице, которая соответствует типу датчика термпары. Значение физической величины, измеренное термпарой, поправляется на величину компенсации.

Перечень таблиц преобразования приводится в подразделе 6.1.

## 6.2 ПРЕОБРАЗОВАНИЯ ДЛЯ ПРИВЕДЕННОГО КОДА

Значение аналогового сигнала в формате приведенного кода (16 разрядов) используется в интеллектуальном модуле M732C контроллера TREI-5B-02. При этом существует следующая зависимость:

- Для униполярного сигнала:

$$\text{Физическая Величина} = \text{Диапазон Измерения} \times \frac{\text{Приведенный Код}}{65535}$$

- Для биполярного сигнала:

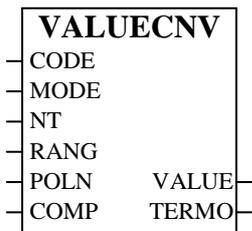
$$\text{Физическая Величина} = \text{Диапазон Измерения} \times \frac{\text{Приведенный Код}}{32767}$$

Для преобразования переменных ввода/вывода модуля M732C из формата приведенного кода в физическую величину и обратно используются следующие функциональные блоки (функции): *valuecny*, *valueinv*. Одновременно может быть выполнено преобразование приведенного кода в значение температуры и обратно. Для возможности преобразования должны быть заданы следующие параметры: диапазон измерения сигнала, полярность сигнала, номер таблицы преобразования температуры. В подразделе 6.1 приведены параметры для каналов ввода термпар и термосопротивлений. Параметры для каналов аналогового ввода приводятся в следующей таблице. Применяются обозначения:

- NT - Номер таблицы преобразования температуры;
- RANG - Диапазон измерения сигнала;
- POLN - Полярность сигнала (0 – биполярный, 1– униполярный).

Канал	Тип	NT	RANG	POLN
Сопротивление	100 Ом	0	100	1
	200 Ом	0	200	1
	500 Ом	0	500	1
Токовый вход	0÷5 mA	0	5	1
	0÷20 mA	0	20	1
	4÷20 mA	0	20	1
	±5 mA	0	5	0
	±10 mA	0	10	0
Милливольтовый вход	±19 mV	0	19	0
	±78 mV	0	78	0
	0÷19 mV	0	19	1
	0÷78 mV	0	78	1
Вольтовый вход	±5 V	0	5	0
	±10 V	0	10	0
	0÷5 V	0	5	1
	0÷10 V	0	10	1
	0÷1 V	0	1	1

6.2.1 VALUECNV – Преобразование приведенного кода в физическую величину



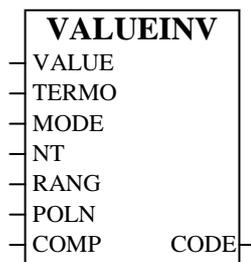
Вх/Вых	Обозначение	Тип	Назначение
Вход	CODE	Integer	Приведенный код
	MODE	Integer	Вид преобразования приведенного кода: 1 – в физическую величину; 2 – в физическую величину и температуру; 3 – в физическую величину и температуру с компенсацией
	NT	Integer	Номер таблицы преобразования температуры
	RANG	Integer	Диапазон измерения сигнала
	POLN	Boolean	Полярность сигнала
	COMP	Real	Температура холодного спая термопары
Выход	VALUE	Real	Значение физической величины (Om / mV / mA / V)
	TERMO	Real	Значение температуры (ГрадС)

**Назначение**

Функциональный блок используется для преобразования приведенного кода в физическую величину. Значение на выходе *TERMO* будет ненулевым при задании соответствующего вида преобразования, и ненулевым значении номера таблицы преобразования. На вход *COMP* подается значение температуры холодного спая термопары, измеренное термосопротивлением. Компенсация выполняется автоматически.

Перечень таблиц преобразования *NT* и параметров *RANG* и *POLN* приводится в подразделах 6.1 и 6.2.

## 6.2.2 VALUEINV – Преобразование физической величины в приведенный код



Вх/Вых	Обозначение	Тип	Назначение
Вход	VALUE	Real	Значение физической величины (Ом / mV / mA / V)
	TERMO	Real	Значение температуры (ГрадС)
	MODE	Integer	Вид преобразования в приведенный код: 1 – из физической величины; 2 – из температуры; 3 – из температуры с компенсацией
	NT	Integer	Номер таблицы преобразования температуры
	RANG	Integer	Диапазон измерения сигнала
	POLN	Boolean	Полярность сигнала
	COMP	Real	Температура холодного спая термопары
Выход	CODE	Integer	Приведенный код

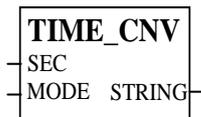
**Назначение**

Функция используется для преобразования физической величины либо температуры в приведенный код. Выбор исходного значения для преобразования (*VALUE* - физическая величина, *TERMO* - температура) осуществляется по значению на входе *MODE* – вид преобразования. При преобразовании температуры должен быть задан номер таблицы преобразования *NT*. Также может подано значение на вход *COMP*, если должна быть автоматически выполнена компенсация на температуру холодного спая термопары.

Перечень таблиц преобразования *NT* и параметров *RANG* и *POLN* приводится в подразделах 6.1 и 6.2.

6.3 ПРЕОБРАЗОВАНИЯ ДЛЯ ВРЕМЕНИ

6.3.1 TIME\_CNV – Преобразование времени в строку символов



Вх/Вых	Обозначение	Тип	Назначение
Вход	SEC	Integer	Секунда с начала 1970г
	MODE	Integer	Режим преобразования: 1 – дата и время 2 – дата 3 – время
Выход	STRING	Message	Строка символов

**Назначение**

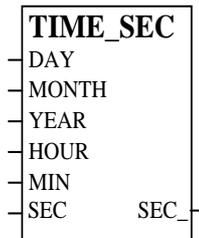
Функция TIME\_CNV используется для преобразования значения времени из формата секунд с 1970г в формат строки символов. Формат строкового представления времени определяется режимом преобразования (вход *MODE*):

- 1 - “mm/dd/yy HH:MM:SS”
- 2 - “mm/dd/yy”
- 3 - “HH:MM:SS”

Каждая составляющая времени представляется в виде двух цифр:

- mm - месяц;
- dd - день;
- yy - год;
- HH - час;
- MM - минута;
- SS - секунда.

## 6.3.2 TIME\_SEC – Преобразование времени в формат секунд с 1970г



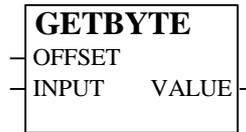
Вх/Вых	Обозначение	Тип	Назначение
Вход	DAY	Integer	День с начала месяца
	MONTH	Integer	Месяц с начала года
	YEAR	Integer	Год
	HOUR	Integer	Час с начала суток
	MIN	Integer	Минута с начала часа
	SEC	Integer	Секунда с начала минуты
Выход	SEC_	Integer	Секунда с начала 1970г

**Назначение**

Функция TIME\_SEC используется для преобразования времени из развернутого формата в формат секунд с 1970г.

## 6.4 ЛОГИЧЕСКИЕ ОПЕРАЦИИ

## 6.4.1 GETBYTE - Выделение байта в переменной

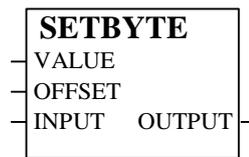


Вх/Вых	Обозначение	Тип	Назначение
Вход	OFFSET	Integer	Смещение байта в переменной [0÷3]
	INPUT	Integer	Значение переменной
Выход	VALUE	Integer	Значение байта

**Назначение**

Функция GETBYTE используется для выделения байта в переменной целого типа. Значение входной переменной не изменяется.

## 6.4.2 SETBYTE - Установка байта в переменной

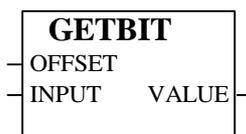


Вх/Вых	Обозначение	Тип	Назначение
Вход	VALUE	Integer	Значение байта
	OFFSET	Integer	Смещение байта в переменной [0÷3]
	INPUT	Integer	Значение переменной до операции
Выход	OUTPUT	Integer	Значение переменной после операции

**Назначение**

Функция SETBYTE используется для установки байта в переменной целого типа. Значение остальных байтов переменной не изменяется.

## 6.4.3 GETBIT - Выделение бита в переменной

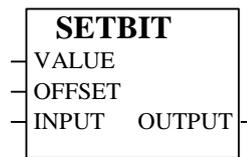


Вх/Вых	Обозначение	Тип	Назначение
Вход	OFFSET	Integer	Смещение бита в переменной [0÷3]
	INPUT	Integer	Значение переменной
Выход	VALUE	Integer	Значение бита

**Назначение**

Функция GETBIT используется для выделения бита в переменной целого типа. Значение входной переменной не изменяется.

## 6.4.4 SETBIT - Установка бита в переменной



Вх/Вых	Обозначение	Тип	Назначение
Вход	VALUE	Integer	Значение бита
	OFFSET	Integer	Смещение бита в переменной (0-3)
	INPUT	Integer	Значение переменной до операции
Выход	OUTPUT	Integer	Значение переменной после операции

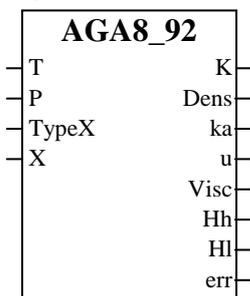
**Назначение**

Функция SETBIT используется для установки бита в переменной целого типа. Значение остальных бит переменной не изменяется.

## 7. СПЕЦИАЛЬНЫЕ РАССЧЕТНЫЕ ФУНКЦИИ

## 7.1 УЧЕТ ТЕПЛА И ГАЗА

## 7.1.1 AGA8\_92 – Расчет свойств природного газа (метод УС AGA8-92DC)



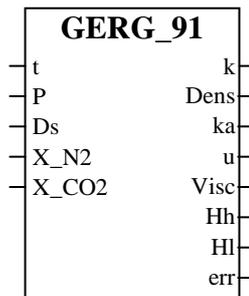
Вх/Вых	Обозначение	Тип	Назначение
Вход	Т	Real	Температура газа (град.С)
	Р	Real	Давление газа (МПа)
	TypeX	Boolean	Тип компонентного состава газа: FALSE – в молярных % TRUE – в объемных %
	X	Integer	Идентификатор массива с компонентным составом газа
Выход	К	Real	Коэффициент сжимаемости
	Dens	Real	Плотность газа при рабочих условиях (кг / куб.м)
	ka	Real	Показатель адиабаты
	u	Real	Скорость звука в газе (м / с)
	Visc	Real	Динамическая вязкость (мкПа * с )
	Hh	Real	Высшая удельная теплота сгорания (МДж / куб.м )
	HI	Real	Низшая удельная теплота сгорания (МДж / куб.м )
	err	Integer	Код ошибки: 0 – нет ошибки; 1 – недопустимое значение температуры; 2 – недопустимое значение давления; 8 – недопустимое значение доли азота; 16 – недопустимое значение доли углекислого газа; 32 – недопустимое значение доли метана; 64 – недопустимое значение доли этана; 128 – недопустимое значение доли пропана; 256 – недопустимое значение доли бутана; 512 – недопустимое значение доли сероводорода; 1024 – недопустимое значение доли остальных компонентов

**Назначение**

Функциональный блок используется для расчета свойств природного газа по методу УС AGA8\_92DC (ГОСТ 30319.2-96).

При использовании данного функционального блока на контроллере должна быть запущена задача *aga8\_92* (правила запуска приводятся в документе “Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя”).

## 7.1.2 GERG\_91 – Расчет свойств природного газа (метод GERG-91 мод.)



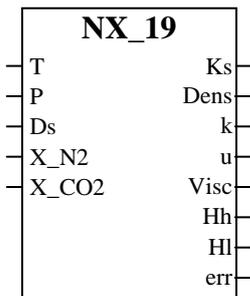
Вх/Вых	Обозначение	Тип	Назначение
Вход	t	Real	Температура газа (град.С)
	P	Real	Давление газа (кПа)
	Ds	Real	Плотность газа при стандартных условиях (кг / куб.м)
	X_N2	Real	Концентрация азота (мол.%)
	X_CO2	Real	Концентрация углекислого газа (мол.%)
Выход	k	Real	Коэффициент сжимаемости
	Dens	Real	Плотность газа при рабочих условиях (кг / куб.м)
	ka	Real	Показатель адиабаты
	u	Real	Скорость звука в газе (м / с)
	Visc	Real	Динамическая вязкость (мкПа * с)
	Hh	Real	Высшая удельная теплота сгорания (МДж / куб.м)
	HI	Real	Низшая удельная теплота сгорания (МДж / куб.м)
	err	Integer	Код ошибки: 0 – нет ошибки; 1 – недопустимое значение температуры; 2 – недопустимое значение давления; 4 – недопустимое значение плотности; 8 – недопустимая концентрация азота; 16 – недопустимая концентрация диоксида углерода

**Назначение**

Функциональный блок используется для расчета свойств природного газа по методу GERG-91 мод. (ГОСТ 30319.2-96).

При использовании данного функционального блока на контроллере должна быть запущена задача *gerg\_91* (правила запуска приводятся в документе “Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя”).

## 7.1.3 NX\_19 – Расчет свойств природного газа (метод NX\_19)



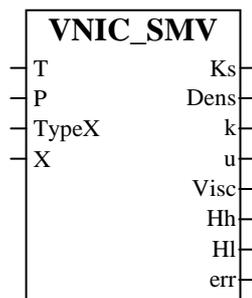
Вх/Вых	Обозначение	Тип	Назначение
Вход	T	Real	Температура газа (град.С)
	P	Real	Давление газа (МПа)
	Ds	Real	Плотность газа при стандартных условиях (кг / куб.м)
	X_N2	Real	Доля азота в газе (в молярных или объемных %)
	X_CO2	Real	Доля углекислоты (в молярных или объемных %)
Выход	Ks	Real	Коэффициент сжимаемости
	Dens	Real	Плотность газа при рабочих условиях (кг / куб.м)
	k	Real	Показатель адиабаты
	u	Real	Скорость звука в газе (м / с)
	Visc	Real	Динамическая вязкость (мкПа * с)
	Hh	Real	Высшая удельная теплота сгорания (МДж / куб.м)
	HI	Real	Низшая удельная теплота сгорания (МДж / куб.м)
err	Integer	Код ошибки: 0 – нет ошибки; 1 – недопустимое значение температуры; 2 – недопустимое значение давления; 4 – недопустимое значение плотности; 8 – недопустимая концентрация азота; 16 – недопустимая концентрация диоксида углерода	

**Назначение**

Функциональный блок используется для расчета свойств природного газа по методу NX\_19 (ГОСТ 30319.2-96).

При использовании данного функционального блока на контроллере должна быть запущена задача *nx\_19* (правила запуска приводятся в документе "Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя").

## 7.1.4 VNIC\_SMV – Расчет свойств природного газа (метод ВНИЦ СМВ)



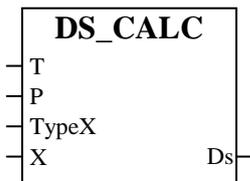
Вх/Вых	Обозначение	Тип	Назначение
Вход	T	Real	Температура газа (град.С)
	P	Real	Давление газа (МПа)
	TypeX	Boolean	Тип компонентного состава газа: FALSE – в молярных % TRUE – в объемных %
	X	Integer	Идентификатор массива с компонентным составом газа
Выход	Ks	Real	Коэффициент сжимаемости
	Dens	Real	Плотность газа при рабочих условиях (кг / куб.м)
	k	Real	Показатель адиабаты
	u	Real	Скорость звука в газе (м / с)
	Visc	Real	Динамическая вязкость (мкПа * с)
	Hh	Real	Высшая удельная теплота сгорания (МДж / куб.м)
	HI	Real	Низшая удельная теплота сгорания (МДж / куб.м)
	err	Integer	Код ошибки: 0 – нет ошибки; 1 – недопустимое значение температуры; 2 – недопустимое значение давления; 8 – недопустимое значение доли азота; 16 – недопустимое значение доли углекислого газа; 32 – недопустимое значение доли метана; 64 – недопустимое значение доли этана; 128 – недопустимое значение доли пропана; 256 – недопустимое значение доли бутана; 512 – недопустимое значение доли сероводорода; 1024 – недопустимое значение доли остальных компонентов

**Назначение**

Функциональный блок используется для расчета свойств природного газа по методу ВНИЦ СМВ (ГОСТ 30319.2-96).

При использовании данного функционального блока на контроллере должна быть запущена задача *vnic\_smv* (правила запуска приводятся в документе “Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя”).

## 7.1.5 DS\_CALC – Расчет плотности природного газа



Вх/Вых	Обозначение	Тип	Назначение
Вход	T	Real	Температура газа (град.С)
	P	Real	Давление газа (МПа)
	TypeX	Boolean	Тип компонентного состава газа: FALSE – в молярных долях TRUE – в объемных долях
	X	Integer	Идентификатор массива с компонентным составом газа
Выход	Ds	Real	Плотность газа при стандартных условиях (кг / куб.м)

**Назначение**

Функциональный блок используется для расчета плотности природного газа при стандартных условиях по компонентному составу (ГОСТ 30319.1-96).

При использовании данного функционального блока на контроллере должна быть запущена задача *ds\_calc* (правила запуска приводятся в документе “Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя”).

## 7.1.6 FLOW\_R – Расчет расхода природного газа

FLOW_R	
- N	qm
- Dens	qv
- Ds	qs
- ka	ok
- Visc	err
- t	
- P	
- dP	

Вх/Вых	Обозначение	Тип	Назначение
Вход	N	Integer	Номер расходомера
	Dens	Real	Плотность газа при рабочих условиях (кг / куб.м)
	Ds	Real	Плотность газа при стандартных условиях (кг / куб.м)
	ka	Real	Показатель адиабаты
	Visc	Real	Динамическая вязкость (мкПа * с)
	t	Real	Температура (град.С)
	P	Real	Давление (кПа)
	dP	Real	Перепад давления (кПа)
Выход	qm	Real	Массовый расход (кг / с)
	qv	Real	Объемный расход (куб.м / с)
	qs	Real	Объемный расход, приведенный к станд. условиям (куб.м / с)
	ok	Integer	Код завершения
	err	Integer	Код ошибки: 0 – нет ошибки; 1 – недопустимое значение температуры; 2 – недопустимое значение давления; 4 – недопустимое значение перепада давления на СУ; 8 – ошибка в параметрах СУ; 16 – недопустимое значение плотности; 32 – недопустимое значение плотности при нормальных усл-ях; 64 – недопустимое значение показателя адиабаты; 128 – недопустимое значение вязкости

**Назначение**

Функциональный блок используется для расчета расхода природного газа (ГОСТ 8.563.1-97, ГОСТ 8.563.2-97).

При использовании данного функционального блока на контроллере должна быть запущена задача *flow\_r* (правила запуска приводятся в документе “Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя”).

## 8. MODBUS ФУНКЦИИ

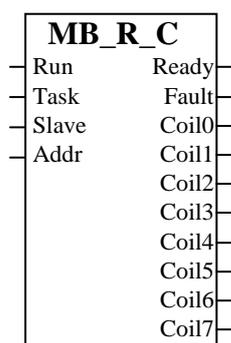
Функциональные блоки, реализующие функции протокола Modbus-RTU, предназначены для организации информационного обмена с подчиненными устройствами, поддерживающими данный протокол.

При использовании данных функциональных блоков на контроллере должна быть запущена задача *modbus* (правила запуска приводятся в документе "Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя").

Правила задания адреса подчиненного устройства и блока данных определяются документацией на конкретный тип устройства.

### 8.1 MODBUS ФУНКЦИИ ЧТЕНИЯ

#### 8.1.1 MB\_R\_C - Чтение состояния 8-ми бинарных ячеек памяти



Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Addr	Integer	Начальный адрес логической ячейки
Выход	Ready	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере
	Coil0	Boolean	Состояние ячейки с адресом <i>Addr+0</i>
	...	...	...
	Coil7	Boolean	Состояние ячейки с адресом <i>Addr+7</i>

#### Назначение

Чтение состояния 8-ми последовательных бинарных ячеек памяти. Используется функция Modbus с кодом 1 (Read Coil Status).

## 8.1.2 MB\_R\_B - Чтение состояния 8-ми дискретных входов

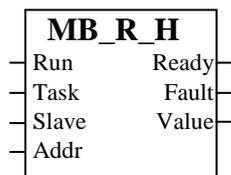
MB_R_B	
Run	Ready
Task	Fault
Slave	Inp_0
Addr	Inp_1
	Inp_2
	Inp_3
	Inp_4
	Inp_5
	Inp_6
	Inp_7

Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Addr	Integer	Начальный адрес дискретного входа
Выход	Ready	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере
	Inp_0	Boolean	Состояние входа с адресом $Addr+0$
	...	...	...
	Inp_7	Boolean	Состояние входа с адресом $Addr+7$

**Назначение**

Чтение состояния 8-ми последовательных дискретных входов. Используется функция Modbus с кодом 2 (Read Input Status).

8.1.3 MB\_R\_H- Чтение целочисленного регистра

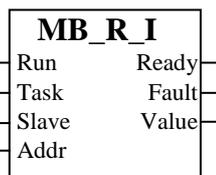


Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Addr	Integer	Адрес регистра
Выход	Ready	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере
	Value	Integer	Значение целочисленного регистра

**Назначение**

Чтение целочисленного регистра. Используется функция Modbus с кодом 3 (Read Holding Registers).

## 8.1.4 MB\_R\_I - Чтение целочисленного входного регистра

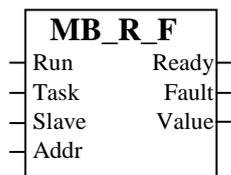


Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Addr	Integer	Адрес входного регистра
Выход	Ready	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере
	Value	Integer	Значение целочисленного входного регистра

**Назначение**

Чтение целочисленного входного регистра. Используется функция Modbus с кодом 4 (Read Input Registers).

8.1.5 MB\_R\_F - Чтение пары регистров в формате с плавающей точкой

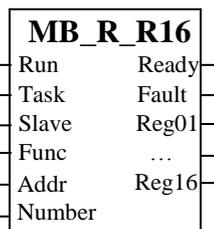


Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Addr	Integer	Начальный адрес пары регистров
Выход	Ready	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере
	Value	Real	Значение пары регистров в формате с плавающей точкой

**Назначение**

Чтение пары последовательных регистров. Значение возвращается в формате числа с плавающей точкой. Используется функция Modbus с кодом 3 (Read Holding Registers).

## 8.1.6 MB\_R\_R16 - Чтение 16 целочисленных регистров

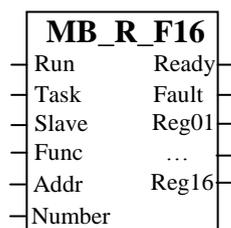


Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Func	Integer	Код Modbus функции (3/4)
	Addr	Integer	Начальный адрес группы регистров
	Number	Integer	Количество регистров
Выход	Ready	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 4 – данные не достоверны 138 – задача связи modbus не запущена на контроллере
	Reg01	Integer	Значение целочисленного регистра с адресом Addr+0
	...	...	...
	Reg16	Integer	Значение целочисленного регистра с адресом Addr+15

**Назначение**

Чтение 16-ти целочисленных регистров подряд начиная с адреса *Addr*. Вход *Func* задает код функции Modbus для запроса данных от устройства. Допускаются следующие функции: 3 (Read Holding Registers) и 4 (Read Input Registers).

8.1.7 MB\_R\_F16 - Чтение 16 пар регистров в формате с плавающей точкой



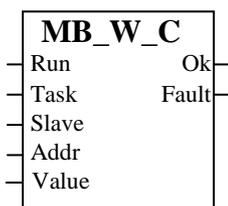
Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Func	Integer	Код Modbus функции (3/4)
	Addr	Integer	Начальный адрес группы регистров
	Number	Integer	Количество пар регистров
Выход	Ready	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 4 – данные не достоверны 138 – задача связи modbus не запущена на контроллере
	Reg01	Real	Значение пары регистров с адреса Addr+0
	Reg02	Real	Значение пары регистров с адреса Addr+2
	...	...	...
	Reg16	Real	Значение пары регистров с адреса Addr+30

**Назначение**

Чтение 16-ти пар регистров подряд начиная с адреса *Addr*. Вход *Func* задает код функции Modbus для запроса данных от устройства. Допускаются следующие функции: 3 (Read Holding Registers) и 4 (Read Input Registers). Значение пары регистров представляется в формате с плавающей точкой.

## 8.2 MODBUS ФУНКЦИИ ЗАПИСИ

### 8.2.1 MB\_W\_C - Запись в бинарную ячейку памяти

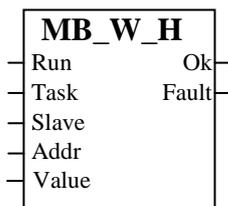


Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Addr	Integer	Адрес ячейки памяти
	Value	Boolean	Состояние ячейки памяти
Выход	Ok	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере

#### Назначение

Запись в бинарную ячейку памяти. Используется функция Modbus с кодом 5 (Force Single Coil). Запись инициируется при переходе значения на входе *Run* из 0 в 1.

8.2.2 MB\_W\_H - Запись целочисленного регистра

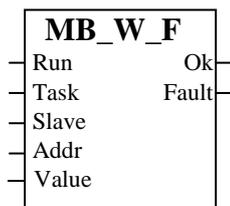


Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Addr	Integer	Адрес регистра
	Value	Integer	Значение целочисленного регистра
Выход	Ok	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере

**Назначение**

Запись целочисленного регистра. Используется функция Modbus с кодом 6 (Preset Single Register). Запись инициируется при переходе значения на входе *Run* из 0 в 1.

## 8.2.3 MB\_W\_F- Запись пары регистров в формате с плавающей точкой



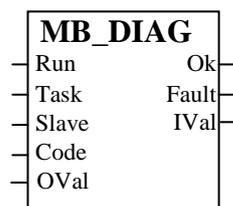
Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Addr	Integer	Начальный адрес пары регистров
	Value	Real	Значение пары регистров в формате с плавающей точкой
Выход	Ok	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере

**Назначение**

Запись пары последовательных регистров. Значение представляется в формате с плавающей точкой. Используется функция Modbus с кодом 16 (Preset Multiple Registers). Запись инициируется при переходе значения на входе *Run* из 0 в 1.

### 8.3 MODBUS ФУНКЦИИ ДИАГНОСТИКИ

#### 8.3.1 MB\_DIAG - Диагностика связи с подчиненным устройством

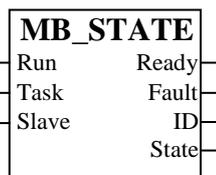


Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
	Code	Integer	Диагностический код
	OVal	Integer	Отправляемые данные
Выход	Ok	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере
	IVal	Integer	Принятые данные

#### Назначение

Проверяется система коммуникации с подчиненным устройством. Используется функция Modbus с кодом 8 (Loopback Test). Диагностика инициируется при переходе значения на входе Run из 0 в 1. Возможные значения диагностического кода определяются документацией на конкретное устройство. В общем случае, при нулевом значении диагностического кода, подчиненное устройство возвращает значение (*IVal*), отправленное через вход *OVal*.

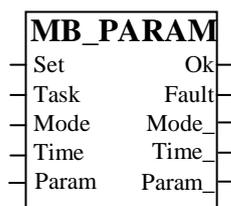
## 8.3.2 MB\_STATE - Получение ID подчиненного устройства



Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи modbus на контроллере
	Slave	Integer	Адрес подчиненного устройства
Выход	Ready	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере
	ID	Integer	ID подчиненного устройства
	State	Integer	Рабочее состояние подчиненного устройства

**Назначение**

Получение ID подчиненного устройства и его рабочего состояния. Используется функция Modbus с кодом 17 (Report Slave ID). Возможные значения ID и рабочего состояния определяются документацией на конкретное устройство.

**8.3.3 MB\_PARAM - Установка параметров задачи связи modbus**


Вх/Вых	Обозначение	Тип	Назначение
Вход	Set	Boolean	Операция: TRUE – установить параметры, FALSE – прочитать
	Task	Integer	Номер задачи связи modbus на контроллере
	Mode	Integer	Режим обмена с устройством: 1 – синхронный, 0 – асинхронный
	Time	Integer	Таймаут на ответ от устройства (мсек)
	Param	Boolean	0 (зарезервировано)
Выход	Ok	Boolean	Операция завершена – TRUE
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 2 – устройство не отвечает 3 – системная ошибка при работе с COM портом 138 – задача связи modbus не запущена на контроллере
	Mode_	Integer	Режим обмена с устройством: 1 – синхронный, 0 – асинхронный
	Time_	Integer	Таймаут на ответ от устройства (мсек)
	Param_	Integer	0 (зарезервировано)

**Назначение**

Установка параметров работы задачи связи modbus. Установка инициируется при переходе значения на входе Set из 0 в 1. При значении 0 на входе Set производится чтение текущих параметров.

## 9. БИБЛИОТЕКА ФУНКЦИЙ HART

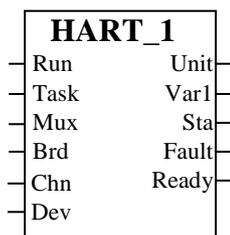
Библиотека содержит набор функциональных блоков, реализующих функции протокола HART, и предназначена для организации информационного обмена с подчиненными устройствами, поддерживающими данный протокол.

При использовании данных функциональных блоков на контроллере должна быть запущена задача *hart* (правила запуска приводятся в документе “Устройство программного управления TREI-5B. Технологическое программирование в системе ISaGRAF. Руководство пользователя”).

Правила задания адреса устройства определяются документацией на конкретный тип устройства.

### 9.1 HART ФУНКЦИИ ЧТЕНИЯ ДАННЫХ

#### 9.1.1 HART\_1 - Считать первичную переменную

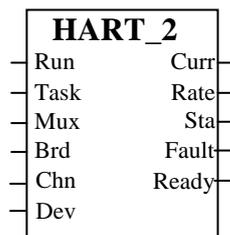


Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи hart на контроллере [1÷4]
	Mux	Integer	Адрес мультиплексора [1÷15]
	Brd	Integer	Номер платы в/в мультиплексора [0÷(N-1)]
	Chn	Integer	Номер канала мультиплексора [0÷31]
	Dev	Integer	Адрес устройства [1÷15]
Выход	Unit	Integer	Код единиц измерения первичной переменной
	Var1	Real	Первичная переменная
	Sta	Integer	Код отклика устройства
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 3 – системная ошибка при работе с COM портом 4 – данные не достоверны(устройство не отвечает) 138 – задача связи hart не запущена на контроллере
	Ready	Boolean	Операция завершена – TRUE

#### Назначение

Используется HART команда #1 – считать первичную переменную. Первичная переменная возвращается в плавающем формате.

## 9.1.2 HART\_2 - Считать ток и процент диапазона



Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи hart на контроллере [1÷4]
	Mux	Integer	Адрес мультиплексора [1÷15]
	Brd	Integer	Номер платы в/в мультиплексора [0÷(N-1)]
	Chn	Integer	Номер канала мультиплексора [0÷31]
	Dev	Integer	Адрес устройства [1÷15]
Выход	Curr	Real	Ток (мА)
	Rate	Real	Процент диапазона (%)
	Sta	Integer	Код отклика устройства
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 3 – системная ошибка при работе с COM портом 4 – данные не достоверны(устройство не отвечает) 138 – задача связи hart не запущена на контроллере
	Ready	Boolean	Операция завершена – TRUE

**Назначение**

Используется HART команда #2 – считать ток и процент диапазона. Чтение первичной переменной как тока и процента диапазона соответствует выходному току прибора, включая аварийные состояния и установленные величины. Процент диапазона не ограничен величинами между 0% и 100%, но и отслеживается за границами диапазона первичной переменной до границ диапазона сенсора (если они определены).

## 9.1.3 HART\_3 - Считать значения четырех динамических переменных

HART_3	
Run	Curr
Task	Unit1
Mux	Var1
Brd	Unit2
Chn	Var2
Dev	Unit3
	Var3
	Unit4
	Var4
	Sta
	Fault
	Ready

Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи hart на контроллере [1÷4]
	Mux	Integer	Адрес мультиплексора [1÷15]
	Brd	Integer	Номер платы в/в мультиплексора [0÷(N-1)]
	Chn	Integer	Номер канала мультиплексора [0÷31]
	Dev	Integer	Адрес устройства [1÷15]
Выход	Curr	Real	Ток первичной переменной (мА)
	Unit1	Integer	Код единиц измерения первичной переменной
	Var1	Real	Первичная переменная
	Unit2	Integer	Код единиц измерения второй переменной
	Var2	Real	Вторая переменная
	Unit3	Integer	Код единиц измерения третьей переменной
	Var3	Real	Третья переменная
	Unit4	Integer	Код единиц измерения четвертой переменной
	Var4	Real	Четвертая переменная
	Sta	Integer	Код отклика устройства
Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 3 – системная ошибка при работе с COM портом 4 – данные не достоверны(устройство не отвечает) 138 – задача связи hart не запущена на контроллере	
Ready	Boolean	Операция завершена – TRUE	

**Назначение**

Используется HART команда #3 – считать значения четырех (предопределенных) динамических переменных и ток первичной переменной. Первичная переменная соответствует выходному току прибора, включая аварийные состояния и установленные величины. Вторая, Третья и Четвертая Переменные определяются для каждого типа прибора.

## 9.2 HART ФУНКЦИИ ЧТЕНИЯ/ЗАПИСИ ДИАПАЗОНА

## 9.2.1 HART\_15 - Считать информацию о выходе первичной переменной

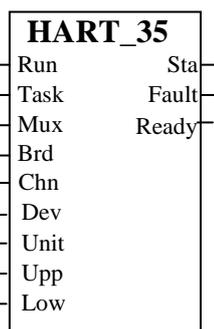
HART_15	
Run	Alarm
Task	Conv
Mux	Unit
Brd	Upp
Chn	Low
Dev	Damp
	Prot
	Label
	Sta
	Fault
	Ready

Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи hart на контроллере [1÷4]
	Mux	Integer	Адрес мультиплексора [1÷15]
	Brd	Integer	Номер платы в/в мультиплексора [0÷(N-1)]
	Chn	Integer	Номер канала мультиплексора [0÷31]
	Dev	Integer	Адрес устройства [1÷15]
Выход	Alarm	Integer	Код выбора тревоги
	Conv	Integer	Код функции преобразования
	Unit	Integer	Код единиц измерения диапазона первичной переменной
	Upp	Real	Верхнее значение диапазона
	Low	Real	Нижнее значение диапазона
	Damp	Real	Величина демпфирования
	Prot	Integer	Код защиты от записи
	Label	Integer	Код метки дистрибьютера
	Sta	Integer	Код отклика устройства
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 3 – системная ошибка при работе с COM портом 4 – данные не достоверны(устройство не отвечает) 138 – задача связи hart не запущена на контроллере
Ready	Boolean	Операция завершена – TRUE	

**Назначение**

Используется HART команда #15 – считать информацию о выходе первичной переменной. Считывается выборочный код алармов первичной переменной, код передаточной функции, единицы измерения, верхний и нижний диапазоны первичной переменной, величина демпфирования первичной переменной, код защиты от записи, код индивидуальной метки дистрибьютера, связанной с прибором или Первичной переменной.

## 9.2.2 HART\_35 - Записать границы диапазона первичной переменной



Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи hart на контроллере [1÷4]
	Mux	Integer	Адрес мультиплексора [1÷15]
	Brd	Integer	Номер платы в/в мультиплексора [0÷(N-1)]
	Chn	Integer	Номер канала мультиплексора [0÷31]
	Dev	Integer	Адрес устройства [1÷15]
	Unit	Integer	Код единиц измерения диапазона
	Upp	Real	Верхнее значение диапазона
	Low	Real	Нижнее значение диапазона
Выход	Sta	Integer	Код отклика устройства
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 3 – системная ошибка при работе с COM портом 4 – данные не достоверны(устройство не отвечает) 138 – задача связи hart не запущена на контроллере
	Ready	Boolean	Операция завершена – TRUE

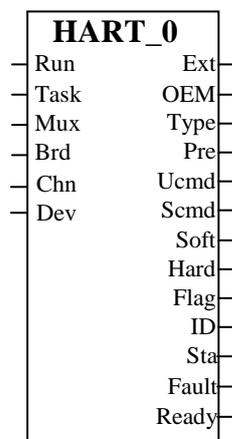
**Назначение**

Используется HART команда #35 – записать верхнюю и нижнюю границы диапазона первичной переменной. Верхняя граница диапазона первичной переменной независима от нижней.

Единицы измерения диапазона первичной переменной, посылаемые с этой командой, не влияют на единицы измерения первичной переменной устройства.

## 9.3 HART ФУНКЦИИ ДИАГНОСТИКИ

## 9.3.1 HART\_0 - Читать уникальный идентификатор

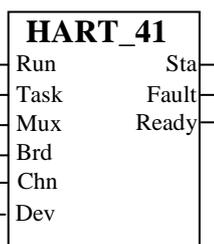


Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи hart на контроллере
	Mux	Integer	Адрес мультиплексора [1÷15]
	Brd	Integer	Номер платы в/в мультиплексора [0÷(N-1)]
	Chn	Integer	Номер канала мультиплексора [0÷31]
	Dev	Integer	Адрес устройства [1÷15]
Выход	Ext	Integer	"254" (расширение)
	OEM	Integer	Код изготовителя
	Type	Integer	Код типа устройства
	Pre	Integer	Число преамбул
	Ucmd	Integer	Версия универсальных команд
	Scmd	Integer	Версия специфических команд
	Soft	Integer	Версия программного обеспечения
	Hard	Integer	Версия аппаратной части
	Flag	Integer	Флаги функций устройства
	ID	Integer	Идентификационный номер устройства
	Sta	Integer	Код отклика устройства
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 3 – системная ошибка при работе с COM портом 4 – данные не достоверны (устройство не отвечает) 138 – задача связи hart не запущена на контроллере
	Ready	Boolean	Операция завершена – TRUE

**Назначение**

Используется HART команда #0 – считать уникальный идентификатор. Возвращает расширенный код типа устройства, версию, идентификационный номер.

## 9.3.2 HART\_41 - Выполнить самотестирование устройства



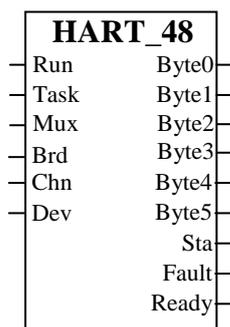
Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи hart на контроллере [1÷4]
	Mux	Integer	Адрес мультиплексора [1÷15]
	Brd	Integer	Номер платы в/в мультиплексора [0÷(N-1)]
	Chn	Integer	Номер канала мультиплексора [0÷31]
	Dev	Integer	Адрес устройства [1÷15]
Выход	Sta	Integer	Код отклика устройства
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 3 – системная ошибка при работе с COM портом 4 – данные не достоверны(устройство не отвечает) 138 – задача связи hart не запущена на контроллере
	Ready	Boolean	Операция завершена – TRUE

**Назначение**

Используется HART команда #41 – выполнение самотестирования устройства. Иницирует функцию самотестирования в устройстве. Устройство немедленно отвечает на команду, затем выполняет самотестирование. Обратитесь к документации на устройство, чтобы узнать о диагностике, которую оно выполняет и информации, доступной через команду #48(считать дополнительный статус устройства).

Выполнение этой команды может занять достаточно большой промежуток времени. Если устройство не может выполнять команды во время самотестирования, оно может не ответить. Иначе, для обнаружения завершения используется команда 48. За детальной информацией обращайтесь к документации на устройство.

## 9.3.3 HART\_48 - Считать дополнительный статус устройства



Вх/Вых	Обозначение	Тип	Назначение
Вход	Run	Boolean	Режим работы: TRUE - выполнить, FALSE не выполнять
	Task	Integer	Номер задачи связи hart на контроллере [1÷4]
	Mux	Integer	Адрес мультиплексора [1÷15]
	Brd	Integer	Номер платы в/в мультиплексора [0÷(N-1)]
	Chn	Integer	Номер канала мультиплексора [0÷31]
	Dev	Integer	Адрес устройства [1÷15]
Выход	Byte0	Integer	Байт 0 в ответе (см. описание на устройство)
	Byte1	Integer	Байт 1 в ответе (см. описание на устройство)
	Byte2	Integer	Байт 2 в ответе (см. описание на устройство)
	Byte3	Integer	Байт 3 в ответе (см. описание на устройство)
	Byte4	Integer	Байт 4 в ответе (см. описание на устройство)
	Byte5	Integer	Байт 5 в ответе (см. описание на устройство)
	Sta	Integer	Код отклика устройства
	Fault	Integer	Код завершения операции: 0 – выполнено успешно 1 – ошибка выделения памяти (некорректные параметры) 3 – системная ошибка при работе с COM портом 4 – данные не достоверны(устройство не отвечает) 138 – задача связи hart не запущена на контроллере
	Ready	Boolean	Операция завершена – TRUE

**Назначение**

Используется HART команда #48 – считать дополнительный статус устройства. Возвращает информацию о статусе устройства, не включенную в код отклика. Эта команда так же возвращает результаты само-тестирования устройства (команда #41). За информацией, содержащейся в каждом байте статуса, обращайтесь к документации на устройство.

Код отклика #8 (предупреждение: идет обновление данных) будет возвращен всегда, когда может быть сделан ответ и информация о статусе ожидает завершения команды, требующий для выполнения достаточно большого промежутка времени.

## 9.4 КОДЫ ОТКЛИКА УСТРОЙСТВА (СТАТУС КОМАНДЫ)

Первый байт:			
ошибки передачи данных (бит7=1)		ошибки команд (бит7=0)	
бит 6	ошибка по четности	0	ошибка, не характерная для команд
бит 5	ошибка переполнения	1	(не определено)
бит 4	ошибка формирования фрейма	2	неверный выбор
бит 3	ошибка контрольной суммы	3	переданный параметр слишком велик
бит 2	(зарезервирован)	4	переданный параметр слишком мал
бит 1	переполнен буфер приемника	5	получено слишком мало байт данных
бит 0	(не определен)	6	ошибка команды, специфической для датчика
		7	в режиме, защищенном от записи
		8	Неудача при обновлении. Обновление в процессе работы. Значение установлено равным ближайшему.
		9	Параметр процесса слишком велик. Нижнее значение диапазона слишком велико. Не в режиме фиксированного тока.
		10	Параметр процесса слишком мал. Нижнее значение диапазона слишком мало. Не поддерживается режим моноканала.
		11	В режиме моноканала. Неверный код переменной датчика. Верхнее значение диапазона слишком велико.
		12	Неверный код единиц измерения. Верхнее значение диапазона слишком мало.
		13	оба значения диапазона выходят за пределы
		14	интервал слишком мал.
		16	доступ ограничен
		32	устройство занято
		64	команда не задействована
Второй байт:			
бит 7	все 0	бит 7	неисправность прибора
бит 6		бит 6	изменена конфигурация
бит 5		бит 5	холодный старт
бит 4		бит 4	
бит 3		бит 3	фиксирован выходной ток
бит 2		бит 2	насыщение аналогового выходного сигнала
бит 1		бит 1	переменная вышла за ограничения
бит 0		бит 0	первичная переменная вышла за ограничения

