

***NaftaProcess***



## **Библиотека функциональных блоков**

2023

# Содержание

<b>1. Условные обозначения и термины.....</b>	<b>11</b>
1.1. Условные обозначения.....	11
1.2. Перечень терминов и сокращений.....	11
<b>2. Введение.....</b>	<b>15</b>
<b>3. Встроенные функциональные блоки.....</b>	<b>17</b>
3.1. Функциональный блок System.....	17
3.1.1. Атрибуты System.....	17
3.1.2. Режимы System.....	18
<b>4. Функциональные блоки для создания удаленного соединения с контроллером.....</b>	<b>19</b>
4.1. Функциональный блок RemoteConnection.....	20
4.1.1. Атрибуты RemoteConnection.....	20
4.1.2. Режимы RemoteConnection.....	20
4.1.3. Алгоритм RemoteConnection.....	20
4.2. Функциональный блок RemoteRealPoint.....	21
4.2.1. Атрибуты RemoteRealPoint.....	21
4.2.2. Режимы RemoteRealPoint.....	22
4.2.3. Алгоритм RemoteRealPoint.....	22
4.2.3.1. Чтение данных из контроллера-источника.....	23
4.2.3.2. Запись.....	23
4.2.3.3. Диагностика.....	23
4.3. Функциональный блок RemoteBooleanPoint.....	24
4.3.1. Атрибуты RemoteBooleanPoint.....	24
4.3.2. Режимы RemoteBooleanPoint.....	25
4.3.3. Алгоритм RemoteBooleanPoint.....	25
4.3.3.1. Чтение данных из контроллера-источника.....	26
4.3.3.2. Запись.....	26
4.3.3.3. Диагностика.....	26
4.4. Функциональный блок RemoteIntegerPoint.....	27
4.4.1. Атрибуты RemoteIntegerPoint.....	27
4.4.2. Режимы RemoteIntegerPoint.....	28
4.4.3. Алгоритм RemoteIntegerPoint.....	28
4.4.3.1. Чтение данных из контроллера-источника.....	29
4.4.3.2. Запись.....	29

4.4.3.3. Диагностика.....	29
<b>5. Функциональные блоки для настройки ввода/вывода.....</b>	<b>30</b>
5.1. Корзина MKRack.....	30
5.1.1. Атрибуты MKRack.....	30
5.1.2. Режимы MKRack.....	31
5.2. Модуль питания МК550024PSU.....	31
5.2.1. Атрибуты МК550024PSU.....	31
5.2.2. Режимы МК550024PSU.....	32
5.3. Модуль ЦПУ МК502142CPU.....	32
5.3.1. Атрибуты МК502142CPU.....	32
5.3.2. Режимы МК502142CPU.....	33
5.3.3. Алгоритм МК502142CPU.....	34
5.3.3.1. Самоидентификация.....	34
5.3.3.2. Запуск.....	35
5.3.3.3. Диагностика.....	35
5.3.3.4. Формирование сигнализаций.....	36
5.4. Модуль аналогового ввода МК513016AI.....	37
5.4.1. Атрибуты МК513016AI.....	37
5.4.2. Режимы МК513016AI.....	38
5.4.3. Алгоритм МК513016AI.....	38
5.4.3.1. Диагностика.....	39
5.4.3.2. Формирование сигнализаций.....	40
5.4.3.3. Преобразование и запись.....	40
5.5. Модуль аналогового ввода МК516008AI.....	40
5.5.1. Атрибуты МК516008AI.....	40
5.5.2. Режимы МК516008AI.....	41
5.5.3. Алгоритм МК516008AI.....	42
5.5.3.1. Диагностика.....	42
5.5.3.2. Формирование сигнализаций.....	43
5.5.3.3. Преобразование и запись.....	43
5.6. Модули дискретного ввода МК521032DI и МК521032ADI.....	43
5.6.1. Атрибуты МК521032DI.....	44
5.6.2. Режимы МК521032DI.....	45
5.6.3. Алгоритм МК521032DI.....	45
5.6.3.1. Диагностика.....	46
5.6.3.2. Формирование сигнализаций.....	46
5.6.3.3. Запись данных модуля ввода/вывода.....	47
5.7. Модуль дискретного ввода МК523032ADI.....	47
5.7.1. Атрибуты МК523032ADI.....	47
5.7.2. Режимы МК523032ADI.....	49
5.7.3. Алгоритм МК523032ADI.....	49
5.7.3.1. Диагностика.....	50

5.7.3.2.	Формирование сигнализаций.....	51
5.7.3.3.	Запись данных модуля ввода/вывода.....	51
5.7.3.4.	Поддержка сигналов NAMUR.....	51
5.7.3.5.	Работа в составе резервной пары.....	52
5.8.	Модули аналогового вывода МК514008АО и МК514008ААО.....	53
5.8.1.	Атрибуты МК514008АО.....	53
5.8.2.	Режимы МК514008АО.....	55
5.8.3.	Алгоритм МК514008АО.....	55
5.8.3.1.	Диагностика.....	56
5.8.3.2.	Формирование сигнализаций.....	56
5.8.3.3.	Преобразование.....	56
5.8.3.4.	Запись данных в модуль ввода/вывода.....	56
5.9.	Модули дискретного вывода МК531032DO и МК531032ADO.....	57
5.9.1.	Атрибуты МК531032DO.....	57
5.9.2.	Режимы МК531032DO.....	58
5.9.3.	Алгоритм МК531032DO.....	58
5.9.3.1.	Диагностика.....	59
5.9.3.2.	Формирование сигнализаций.....	60
5.9.3.3.	Запись данных в модуль ввода/вывода.....	60
5.10.	Модуль аналогового ввода МК576008ААHART.....	60
5.10.1.	Атрибуты МК576008ААHART.....	60
5.10.2.	Режимы МК576008ААHART.....	63
5.10.3.	Алгоритм МК576008ААHART.....	63
5.10.3.1.	Диагностика.....	64
5.10.3.2.	Формирование сигнализаций.....	64
5.10.3.3.	Преобразование и запись.....	65
5.10.3.4.	Работа HART модулей в составе PDM систем.....	65
5.11.	Модуль аналогового ввода МК576016ААHART.....	66
5.11.1.	Атрибуты МК576016ААHART.....	66
5.11.2.	Режимы МК576016ААHART.....	69
5.11.3.	Алгоритм МК576016ААHART.....	69
5.11.3.1.	Диагностика.....	71
5.11.3.2.	Формирование сигнализаций.....	71
5.11.3.3.	Преобразование и запись.....	71
5.11.3.4.	Работа HART в режиме резервирования.....	72
5.12.	Модуль аналогового вывода МК574008ААО с поддержкой HART... ..	73
5.12.1.	Атрибуты МК574008ААОHART.....	74
5.12.2.	Режимы МК574008ААОHART.....	76
5.12.3.	Алгоритм МК574008ААОHART.....	76
5.12.3.1.	Диагностика.....	78
5.12.3.2.	Формирование сигнализаций.....	78
5.12.3.3.	Преобразование.....	78
5.12.3.4.	Обработка данных в модуле ввода/вывода.....	79

5.12.3.5. Работа HART в режиме резервирования.....	79
5.13. Модуль коммуникации МК541002 с поддержкой Modbus.....	80
5.13.1. Атрибуты МК541002.....	80
5.13.2. Режимы МК541002.....	82
5.13.3. Алгоритм МК541002.....	82
5.13.3.1. Конфигурация.....	83
5.13.3.2. Коммуникация.....	83
5.13.3.3. Диагностика.....	84
5.13.3.4. Формирование сигнализаций.....	84
<b>6. Функциональные блоки коммуникации по ModBus.....</b>	<b>85</b>
6.1. Устройство ModBusTCPDevice.....	87
6.1.1. Атрибуты ModBusTCPDevice.....	87
6.1.2. Режимы ModBusTCPDevice.....	87
6.1.3. Алгоритм ModBusTCPDevice.....	87
6.1.3.1. Коммуникация с устройством ModBusTCP.....	88
6.1.3.2. Диагностика коммуникации.....	88
6.2. Устройство ModBusRTUDevice.....	88
6.2.1. Атрибуты ModBusRTUDevice.....	88
6.2.2. Режимы ModBusRTUDevice.....	89
6.2.3. Алгоритм ModBusRTUDevice.....	89
6.2.3.1. Коммуникация с устройством Modbus.....	89
6.2.3.2. Диагностика коммуникации.....	90
6.3. Карта ModBusRealInput32.....	90
6.3.1. Атрибуты ModBusRealInput32.....	90
6.3.2. Режимы ModBusRealInput32.....	92
6.3.3. Алгоритм ModBusRealInput32.....	92
6.3.3.1. Чтение данных устройства Modbus.....	92
6.3.3.2. Диагностика коммуникации.....	93
6.4. Карта ModBusRealOutput32.....	93
6.4.1. Атрибуты ModBusRealOutput32.....	93
6.4.2. Режимы ModBusRealOutput32.....	94
6.4.3. Алгоритм ModBusRealOutput32.....	95
6.4.3.1. Запись данных в устройство Modbus.....	95
6.4.3.2. Диагностика коммуникации.....	96
6.5. Карта ModBusIntegerInput32.....	96
6.5.1. Атрибуты ModBusIntegerInput32.....	96
6.5.2. Режимы ModBusIntegerInput32.....	97
6.5.3. Алгоритм ModBusIntegerInput32.....	97
6.5.3.1. Чтение данных устройства Modbus.....	98
6.5.3.2. Диагностика коммуникации.....	98
6.6. Карта ModBusIntegerOutput32.....	99
6.6.1. Атрибуты ModBusIntegerOutput32.....	99

6.6.2.	Режимы ModBusIntegerOutput32.....	100
6.6.3.	Алгоритм ModBusIntegerOutput32.....	100
6.6.3.1.	Запись данных в Modbus устройство.....	101
6.6.3.2.	Диагностика коммуникации.....	101
6.7.	Карта ModBusUIntegerInput32.....	102
6.7.1.	Атрибуты ModBusUIntegerInput32.....	102
6.7.2.	Режимы ModBusUIntegerInput32.....	103
6.7.3.	Алгоритм ModBusUIntegerInput32.....	103
6.7.3.1.	Чтение данных устройства Modbus.....	104
6.7.3.2.	Диагностика коммуникации.....	104
6.8.	Карта ModBusUIntegerOutput32.....	105
6.8.1.	Атрибуты ModBusUIntegerOutput32.....	105
6.8.2.	Режимы ModBusUIntegerOutput32.....	106
6.8.3.	Алгоритм ModBusUIntegerOutput32.....	106
6.8.3.1.	Запись данных в Modbus устройство.....	107
6.8.3.2.	Диагностика коммуникации.....	107
6.9.	Карта ModBusBooleanInput32.....	108
6.9.1.	Атрибуты ModBusBooleanInput32.....	108
6.9.2.	Режимы ModBusBooleanInput32.....	109
6.9.3.	Алгоритм ModBusBooleanInput32.....	109
6.9.3.1.	Чтение данных устройства Modbus.....	110
6.9.3.2.	Диагностика коммуникации.....	110
6.10.	Карта ModBusBooleanOutput32.....	111
6.10.1.	Атрибуты ModBusBooleanOutput32.....	111
6.10.2.	Режимы ModBusBooleanOutput32.....	112
6.10.3.	Алгоритм ModBusBooleanOutput32.....	112
6.10.3.1.	Запись данных в Modbus устройство.....	113
6.10.3.2.	Диагностика коммуникации.....	113
6.11.	Карта ModBusShortInput32.....	114
6.11.1.	Атрибуты ModBusShortInput32.....	114
6.11.2.	Режимы ModBusShortInput32.....	115
6.11.3.	Алгоритм ModBusShortInput32.....	115
6.11.3.1.	Чтение данных устройства Modbus.....	116
6.11.3.2.	Диагностика коммуникации.....	116
6.12.	Карта ModBusShortOutput32.....	117
6.12.1.	Атрибуты ModBusShortOutput32.....	117
6.12.2.	Режимы ModBusShortOutput32.....	118
6.12.3.	Алгоритм ModBusShortOutput32.....	118
6.12.3.1.	Запись данных в Modbus устройство.....	119
6.12.3.2.	Диагностика коммуникации.....	119
6.13.	Карта ModBusUshortInput32.....	120
6.13.1.	Атрибуты ModBusUshortInput32.....	120
6.13.2.	Режимы ModBusUshortInput32.....	121

6.13.3.	Алгоритм ModBusUshortInput32.....	121
6.13.3.1.	Чтение данных устройства Modbus.....	122
6.13.3.2.	Диагностика коммуникации.....	122
6.14.	Карта ModBusUshortOutput32.....	123
6.14.1.	Атрибуты ModBusUshortOutput32.....	123
6.14.2.	Режимы ModBusUshortOutput32.....	124
6.14.3.	Алгоритм ModBusUshortOutput32.....	124
6.14.3.1.	Запись данных в Modbus устройство.....	125
6.14.3.2.	Диагностика коммуникации.....	125
<b>7.</b>	<b>Технологические функциональные блоки.....</b>	<b>126</b>
7.1.	Функциональный блок Control Button.....	126
7.1.1.	Атрибуты Control Button.....	126
7.1.2.	Режимы Control Button.....	126
7.1.3.	Алгоритм Control Button.....	126
7.2.	Входной аналоговый блок AnalogInputPoint.....	127
7.2.1.	Атрибуты AnalogInputPoint.....	127
7.2.2.	Режимы AnalogInputPoint.....	130
7.2.3.	Алгоритм AnalogInputPoint.....	130
7.2.3.1.	Контроль достоверности.....	131
7.2.3.2.	Масштабирование.....	132
7.2.3.3.	Контроль ручного значения.....	132
7.2.3.4.	Формирование сигнализаций.....	132
7.3.	Выходной аналоговый блок AnalogOutputPoint.....	132
7.3.1.	Атрибуты AnalogOutputPoint.....	133
7.3.2.	Диагностика состояния физического канала.....	134
7.3.3.	Режимы AnalogOutputPoint.....	134
7.3.4.	Алгоритм AnalogOutputPoint.....	134
7.3.4.1.	Контроль достоверности.....	135
7.3.4.2.	Масштабирование.....	135
7.4.	Дискретный блок DiscretePoint.....	136
7.4.1.	Атрибуты DiscretePoint.....	136
7.4.2.	Режимы DiscretePoint.....	138
7.4.3.	Алгоритм DiscretePoint.....	138
7.4.3.1.	Инверсия.....	138
7.4.3.2.	Фильтрация дребезга.....	139
7.4.3.3.	Формирование сигнализаций.....	139
7.5.	ПИД регулятор PIDController.....	139
7.5.1.	Атрибуты PIDController.....	139
7.5.2.	Режимы PIDController.....	142
7.5.3.	Алгоритм PIDController.....	144
7.5.3.1.	Регулирование по ПИД-закону.....	145
7.5.3.2.	Контроль выходного значения.....	146

7.6.	Функциональный блок Motor.....	146
7.6.1.	Атрибуты Motor.....	146
7.6.2.	Режимы Motor.....	149
7.6.3.	Алгоритм Motor.....	149
7.6.3.1.	Диагностика и управление.....	149
7.6.3.2.	Формирование сигнализаций.....	151
7.7.	Функциональный блок Program.....	152
7.7.1.	Атрибуты Program.....	152
7.7.2.	Режимы Program.....	153
7.7.3.	Алгоритм блока Program.....	153
7.7.3.1.	Чтение.....	154
7.7.3.2.	Интерпретация и запись.....	154
7.7.3.3.	Диагностика.....	155
7.8.	Резервируемый дискретный функциональный блок RedundantDiscretePoint.....	155
7.8.1.	Атрибуты RedundantDiscretePoint.....	155
7.8.2.	Режимы RedundantDiscretePoint.....	157
7.8.3.	Алгоритм RedundantDiscretePoint.....	157
7.8.3.1.	Диагностика и резервирование.....	158
7.8.3.2.	Обработка данных.....	158
7.8.3.3.	Формирование сигнализаций.....	159
7.9.	Резервируемый входной аналоговый функциональный блок RedundantAnalogInputPoint.....	159
7.9.1.	Атрибуты RedundantAnalogInputPoint.....	159
7.9.2.	Режимы RedundantAnalogInputPoint.....	163
7.9.3.	Алгоритм RedundantAnalogInputPoint.....	163
7.9.3.1.	Диагностика и резервирование.....	164
7.9.3.2.	Масштабирование и запись данных.....	165
7.9.3.3.	Контроль ручного значения.....	165
7.9.3.4.	Формирование сигнализаций.....	165
7.10.	Функциональный блок Valve.....	166
7.10.1.	Атрибуты Valve.....	166
7.10.2.	Режимы Valve.....	169
7.10.3.	Алгоритм Valve.....	169
7.10.3.1.	Диагностика и управление Valve.....	170
7.10.3.2.	Формирование сигнализаций.....	172
7.11.	Функциональный блок Logical block.....	173
7.11.1.	Атрибуты Logical block.....	173
7.11.2.	Режимы Logical block.....	174
7.11.3.	Алгоритм Logical block.....	174
7.11.3.1.	Чтение.....	174
7.11.3.2.	Обработка данных.....	175
7.11.3.3.	Диагностика.....	175



7.12.	Функциональный блок MinMax Block.....	176
7.12.1.	Атрибуты MinMax Block.....	176
7.12.2.	Режимы MinMax Block.....	177
7.12.3.	Алгоритм MinMax Block.....	178
7.12.3.1.	Чтение.....	178
7.12.3.2.	Обработка данных.....	178
7.12.3.3.	Диагностика.....	179
7.13.	Функциональный блок Accumulator Block.....	180
7.13.1.	Атрибуты Accumulator Block.....	180
7.13.2.	Режимы Accumulator Block.....	181
7.13.3.	Алгоритм Accumulator Block.....	181
7.13.3.1.	Чтение.....	182
7.13.3.2.	Обработка данных.....	182
7.13.3.3.	Диагностика.....	182
7.14.	Резервируемый ПИД регулятор RedundantPIDController.....	182
7.14.1.	Атрибуты RedundantPIDController.....	183
7.14.2.	Режимы RedundantPIDController.....	189
7.14.3.	Алгоритм RedundantPIDController.....	191
7.14.3.1.	Диагностика и резервирование.....	192
7.14.3.2.	Масштабирование и запись данных.....	193
7.14.3.3.	Контроль ручного значения.....	193
7.14.3.4.	Формирование сигнализаций.....	193
7.14.3.5.	Регулирование по ПИД-закону.....	194
7.14.3.6.	Контроль выходного значения.....	194
7.15.	Функциональный блок DiscretePacker.....	194
7.15.1.	Атрибуты DiscretePacker.....	194
7.15.2.	Режимы DiscretePacker.....	195
7.15.3.	Алгоритм DiscretePacker.....	195
7.15.3.1.	Чтение.....	196
7.15.3.2.	Обработка данных.....	196
7.16.	Функциональный блок DigitalUnpacker.....	196
7.16.1.	Атрибуты DigitalUnpacker.....	196
7.16.2.	Режимы DigitalUnpacker.....	197
7.16.3.	Алгоритм DigitalUnpacker.....	197
7.16.3.1.	Чтение.....	198
7.16.3.2.	Обработка данных.....	198
7.17.	Функциональный блок TimeDelay.....	198
7.17.1.	Атрибуты TimeDelay.....	199
7.17.2.	Режимы TimeDelay.....	199
7.17.3.	Алгоритм TimeDelay.....	199
7.17.3.1.	Чтение.....	200
7.17.3.2.	Обработка данных.....	200
7.18.	Функциональный блок PulseDuration.....	201

7.18.1. Атрибуты PulseDuration.....	201
7.18.2. Режимы PulseDuration.....	202
7.18.3. Алгоритм PulseDuration.....	202
7.18.3.1. Чтение.....	203
7.18.3.2. Обработка данных.....	203
7.19. Функциональный блок RSTrigger.....	204
7.19.1. Атрибуты RSTrigger.....	204
7.19.2. Режимы RSTrigger.....	205
7.19.3. Алгоритм RSTrigger.....	205
7.19.3.1. Чтение.....	206
7.19.3.2. Обработка данных.....	206
7.20. Функциональный блок ToWords.....	208
7.20.1. Атрибуты ToWords.....	208
7.20.2. Режимы ToWords.....	209
7.20.3. Алгоритм ToWords.....	209
7.21. Функциональный блок WordsTo.....	210
7.21.1. Атрибуты WordsTo.....	210
7.21.2. Режимы WordsTo.....	211
7.21.3. Алгоритм WordsTo.....	211
7.22. Функциональный блок Not.....	212
7.22.1. Атрибуты Not.....	212
7.22.2. Режимы Not.....	212
7.22.3. Алгоритм Not.....	213
7.23. Функциональный блок Change Analyzer.....	213
7.23.1. Атрибуты Change Analyzer.....	213
7.23.2. Режимы Change Analyzer.....	213
7.23.3. Алгоритм Change Analyzer.....	213
<b>8. Функциональные блоки противоаварийной защиты.....</b>	<b>215</b>
8.1. Функциональный блок DiscreteVoter4/16.....	215
8.1.1. Атрибуты DiscreteVoter4/16.....	215
8.1.2. Режимы DiscreteVoter4/16.....	218
8.1.3. Алгоритм DiscreteVoter4/16.....	218
<b>9. Кастомные атрибуты для функциональных блоков.....</b>	<b>221</b>
<b>10. Приложение. Базовый функционал.....</b>	<b>223</b>
10.1. Формирование сигнализаций.....	223

# 1. Условные обозначения и термины

## 1.1. Условные обозначения



### **Внимание:**

Помечает информацию, с которой необходимо ознакомиться, чтобы учесть особенности работы какого-либо элемента программного обеспечения.



### **ОСТОРОЖНО:**

Помечает информацию, с которой необходимо ознакомиться, чтобы предотвратить нарушения в работе программного обеспечения либо предотвратить потерю данных.



### **ОПАСНО:**

Помечает информацию, с которой необходимо ознакомиться, чтобы избежать потери контроля над технологическим процессом.

## 1.2. Перечень терминов и сокращений

### **Атрибут**

Атрибут<sup>1</sup> функционального блока - одно из значений, характеризующих функциональный блок, позволяющее задавать его свойства<sup>2</sup>.

### **Контроллер РСУ**

Контроллер распределенной системы управления - узел РСУ, выполняющий технологическую программу. Контроллер РСУ обеспечивает связь технологической программы с объектом управления и вышестоящими узлами РСУ.

---

<sup>1</sup> Перечень типов данных атрибутов функционального блока описан в разделе **4. Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока** документа "Концепция технологического программного обеспечения".

<sup>2</sup> Базовый набор атрибутов описан в разделе **4.1 Базовый набор атрибутов функционального блока** документа "Концепция технологического программного обеспечения".

## **Лицевая панель функционального блока**

Лицевая панель функционального блока представляет собой программный компонент (окно), вызываемый в режиме исполнения и содержащий параметры функционального блока. Привязка шаблона лицевой панели к типу функционального блока производится при настройке станции оператора<sup>3</sup>.

## **Модуль ввода/вывода**

Составная аппаратная часть контроллера РСУ, предназначенная для сопряжения с объектом управления, которая обладает интерфейсами для считывания показаний со средств измерения и/или выдачи управляющего воздействия на исполнительный механизм.

## **Модуль ЦПУ**

Модуль центрального процессорного устройства - составная аппаратная часть контроллера РСУ, предназначенная для выполнения технологической программы, а также коммуникации с другими узлами РСУ.

## **Окно конфигурации функционального блока**

Визуальный компонент конфигурации функционального блока, посредством которого задаются его настройки.

## **ПО**

Программное обеспечение.

## **Прикладная программа**

Программа с пользовательским интерфейсом, предназначенная для выполнения задач пользователя станции инженера и станции оператора.

## **Проект**

Набор данных, который представляет конфигурацию РСУ. Проект хранится на станции инженера в единственном экземпляре.

## **Распределенная система управления - РСУ**

Программно-аппаратный комплекс управления технологическим процессом, характеризующийся распределенной системой ввода-вывода и децентрализацией обработки данных.

---

<sup>3</sup> "Руководство по созданию технологического программного обеспечения станции оператора" п. Добавление станции оператора в проект.

## **Сигнализация**

Сообщение о технологическом или системном событии в системе, которое требует внимания оператора или инженера.

## **Системная программа**

Программа, которая обеспечивает исполнение технологической программы и обмен информацией между узлами РСУ.

## **Системное программное обеспечение - системное ПО**

Набор программ, которые обеспечивают функционирование технологических программ в узлах РСУ, а также обмен информацией между этими узлами.

## **Системный функциональный блок**

Функциональный блок, обеспечивающий доступ к аппаратным функциям контроллера РСУ.

## **Системный цикл**

Однократное выполнение технологической программы в среде исполнения контроллера РСУ.

## **Станция инженера**

Узел РСУ, предоставляющий функции конфигурации и диагностики. Включает в себя персональный компьютер и программное обеспечение станции инженера.

## **Станция интеграции**

Узел РСУ, представляющий собой сервер, программное обеспечение которого предназначено для интеграции РСУ в вышестоящие автоматизированные системы предприятия.

## **Станция оператора**

Узел РСУ, который представляет собой программно-аппаратный комплекс системы, включающий в себя персональный компьютер и программное обеспечение станции оператора (ПО станции оператора). Станция оператора выполняет функции визуализации и дистанционного управления технологическим процессом.

## **Технологическое программное обеспечение**

Программное обеспечение, которое выполняется в среде исполнения контроллера РСУ и состоит из связанных между собой в контуры функциональных блоков.

## **Технологический функциональный блок**

Функциональный блок, выполняющий функцию автоматизации.

## **Транспортная сеть распределенной системы управления - транспортная сеть РСУ**

Локальная вычислительная сеть, которая связывает все узлы РСУ между собой.

## **Узел распределенной системы управления - узел РСУ**

Программно-аппаратная составная часть РСУ (контроллер РСУ, станция инженера, станция оператора, станция интеграции, транспортная сеть РСУ), соединенная с другими составными частями РСУ посредством транспортной сети РСУ, и выполняющая конкретные функции.

## **Функциональный блок**

Составная программная часть технологической программы РСУ, которая используется при построении программного обеспечения РСУ. Функциональные блоки подразделяются на технологические и системные.

## **Функция автоматизации**

Алгоритм, применяемый для автоматизированного управления и контроля над производственным технологическим процессом (например, обработка аналогового сигнала, управление задвижкой и т. д.).

## **Модуль ЦПУ**

Модуль центрального процессорного устройства - составная аппаратная часть контроллера РСУ, предназначенная для выполнения технологической программы, а также коммуникации с другими узлами РСУ.

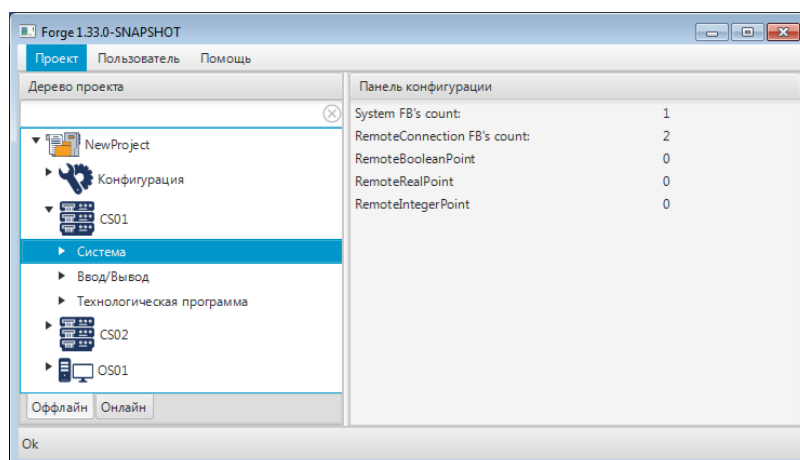
## 2. Введение

Документ "Библиотека функциональных блоков" (далее Библиотека) относится к комплексу эксплуатационных документов программного обеспечения распределенной системы управления (ПО РСУ).

Технологическое программное обеспечение состоит из экземпляров функциональных блоков, которые выполняют функции автоматизации. Каждый экземпляр уникален и имеет собственное состояние и настройки. Функциональные блоки являются частью системного программного обеспечения РСУ и разделены на группы в соответствии с их функциональным назначением:

- встроенные системные функциональные блоки (блок System);
- функциональные блоки для создания удаленного соединения с контроллером;
- функциональные блоки-модули ввода/вывода;
- функциональные блоки коммуникации по ModBus;
- технологические функциональные блоки.

В панели конфигурации группы блоков производится подсчет функциональных блоков.



**Рисунок 1. Счетчик функциональных блоков**

Каждый экземпляр функционального блока принадлежит к определенному классу. Класс задает назначение и алгоритм работы функционального блока. Функциональный блок имеет такие параметры как атрибуты и режим. Атрибуты предоставляют доступ к данным функционального блока. Режим определяет работу алгоритма функционального блока.

Дополнительная информация о функциональных блоках представлена в документе "Концепция технологического программного обеспечения" п. Базовый набор атрибутов функционального блока, п. Режим функционального блока.



**Внимание:** Справочная информация доступна:

- из главного меню командой **Помощь > Справка**;
- по клавише **“F1”**;
- выбором пункта **Справка** из контекстного меню дерева проекта.



## 3. Встроенные функциональные блоки

Встроенные функциональные блоки являются частью системного программного обеспечения PCY.

К данным функциональным блокам относятся:

- Функциональный блок System;

Встроенные функциональные блоки и их имена создаются в контроллере автоматически и присутствует в одном экземпляре.

### 3.1. Функциональный блок System

Функциональный блок System (далее - System) предоставляет доступ к настройкам контроллера и системной информации.

Для System автоматически создается имя вида **System##**, где ## - номер контроллера.

System используется для настройки синхронизации времени контроллера с сервером времени<sup>4</sup>.

#### 3.1.1. Атрибуты System

System имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 1. Дополнительные атрибуты System**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Диагностические атрибуты</i>			
Version	STRING	чтение	Версия конфигурации контроллера
SystemLoad	REAL	чтение	Заполнение системного цикла, %
FBCount	INT	чтение	Количество блоков
RawTime	DATE_AND_	чтение/ запись	Текущее время

<sup>4</sup> "Руководство по инсталляции", Настройка синхронизации внутреннего времени узлов PCY.

Имя атрибута	Тип данных	Доступ	Описание атрибута
Year	INT	чтение	Текущий год
Month	INT	чтение	Текущий месяц
Day	INT	чтение	Текущий день
Hour	INT	чтение	Текущий час
Minute	INT	чтение	Текущая минута
Second	INT	чтение	Текущая секунда
InterfaceConfiguration	STRING	чтение	Конфигурация Ethernet порта в ТС PCY
<b><i>Конфигурационные атрибуты</i></b>			
TimeServerIP	STRING	чтение/ запись	IP сервера времени

### 3.1.2. Режимы System

Режимы System:

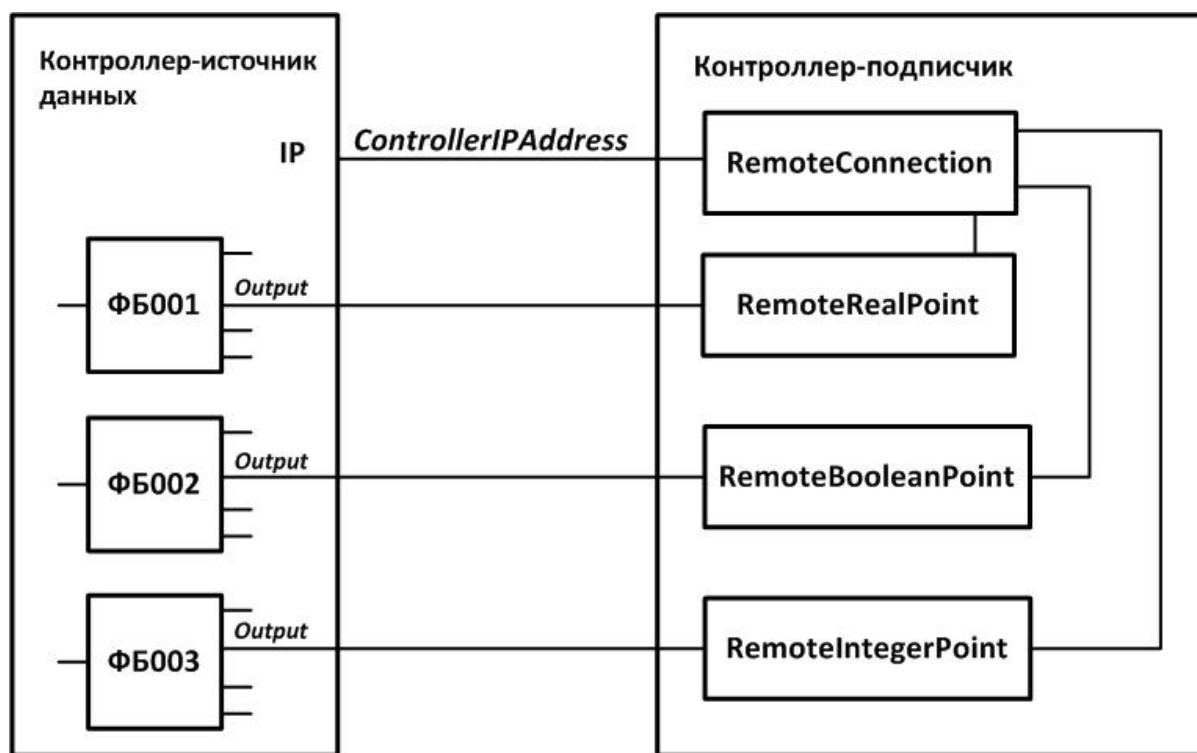
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

## 4. Функциональные блоки для создания удаленного соединения с контроллером

Функциональные блоки для создания удаленного соединения с контроллером являются частью системного программного обеспечения PCY.

К данным функциональным блокам относятся:

- Функциональный блок RemoteConnection;
- Функциональный блок RemoteRealPoint;
- Функциональный блок RemoteBooleanPoint;
- Функциональный блок RemoteIntegerPoint.



**Рисунок 2. Межконтроллерное соединение с контроллером-источником данных**

Этапы настройки межконтроллерного соединения:

1. Добавьте RemoteConnection в контроллер-подписчик на данные.
2. Запишите IP адрес контроллера-источника в атрибут RemoteConnection.
3. Добавьте удаленную точку в контроллер-подписчик (RemoteRealPoint либо RemoteDiscretePoint).
4. Привяжите RemoteConnection к удаленной точке.

5. Привяжите удаленную точку к функциональному блоку контроллера-источника.

Настройку межконтроллерного соединения в среде разработки см. "Руководство по созданию технологического программного обеспечения контроллера РСУ".

## 4.1. Функциональный блок RemoteConnection

RemoteConnection применяется при создании межконтроллерного соединения.

RemoteConnection добавляется в контроллер-подписчик на данные и содержит IP адрес контроллера-источника.

### 4.1.1. Атрибуты RemoteConnection

RemoteConnection имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 2. Дополнительные атрибуты RemoteConnection**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Конфигурационные атрибуты</i>			
ControllerIPAddress	STRING	чтение/запись	IP адрес контроллера-источника данных

### 4.1.2. Режимы RemoteConnection

Режимы RemoteConnection:

- AUTO ("Автоматический");
- MASKING ("Маскирование").

### 4.1.3. Алгоритм RemoteConnection

После добавления RemoteConnection в проект, в атрибут **ControllerIPAddress** записывается IP адрес контроллера-источника данных.

Функциональные блоки RemoteRealPoint, RemoteIntegerPoint и RemoteBooleanPoint используют имя RemoteConnection для привязки к контроллеру-источнику.

## 4.2. Функциональный блок RemoteRealPoint

RemoteRealPoint применяется при создании межконтроллерного соединения. ФБ этого типа подписываются на вещественные числа удаленного контроллера.

### 4.2.1. Атрибуты RemoteRealPoint

RemoteRealPoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 3. Дополнительные атрибуты RemoteRealPoint

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
Input	REAL	чтение	Вход
Output	REAL	чтение	Выход
<i>Конфигурационные атрибуты</i>			
Source	STRING	чтение/ запись	Источник
ManualValue	REAL	чтение/ запись	Ручное значение
RemoteConnection	STRING	чтение/ запись	Удаленное соединение

Некоторые значения атрибута *Status* для RemoteRealPoint имеют дополнительное описание:

Таблица 4. Дополнительное описание значений атрибута *Status* для RemoteRealPoint

Имя	Значение	Описание
Uninitialized	0x01010000	Функциональный блок добавлен в проект, но еще не участвовал ни в одном системном цикле

Имя	Значение	Описание
ResolvingReference	0x01030000	Функциональный блок находится в поиске контроллера-источника данных
UncertainSourceStatus	0x01040000	Функциональный блок обнаружил контроллер-источник, но источник данных еще не инициализирован
BadSourceStatus	0x02040000	Атрибут <b>Status</b> функционального блока контроллера-источника имеет значение не Ok
CommunicationError	0x02070000	Отсутствует коммуникация с контроллером-источником

#### 4.2.2. Режимы RemoteRealPoint

Режимы RemoteRealPoint:

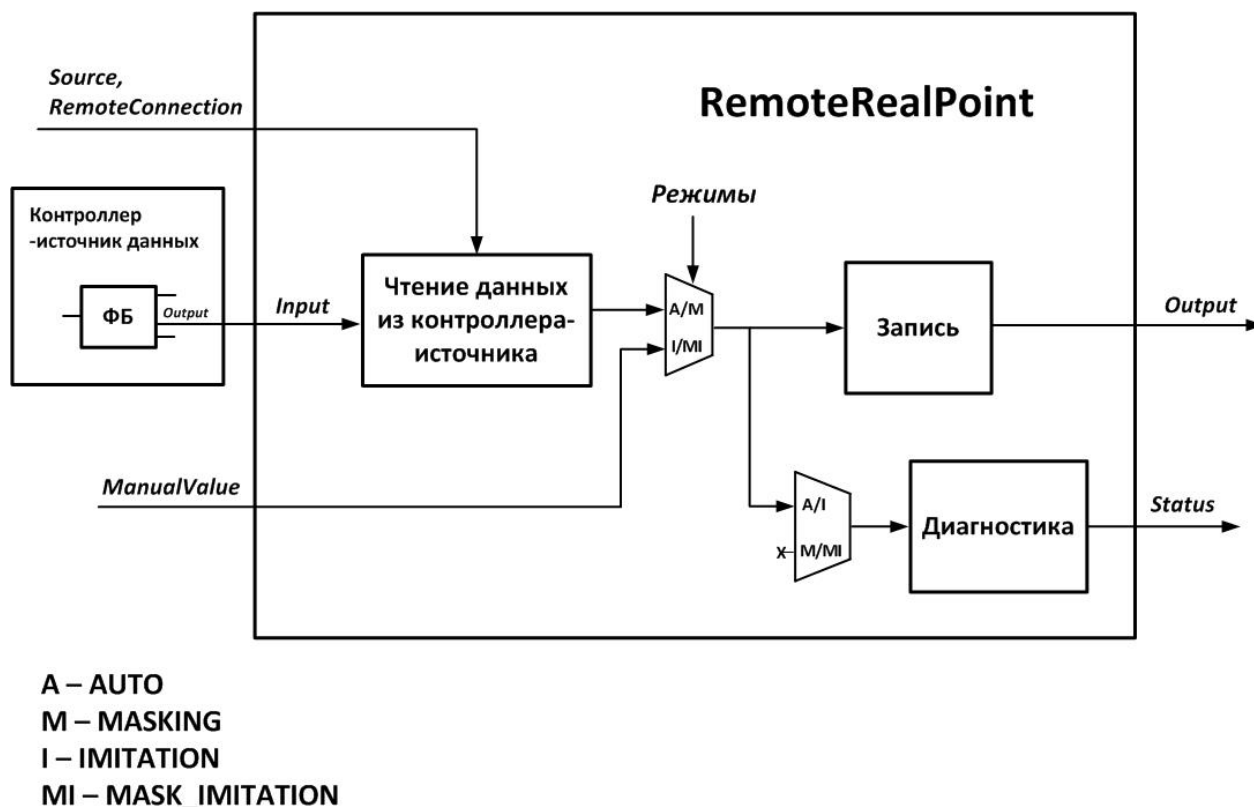
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

#### 4.2.3. Алгоритм RemoteRealPoint

Выполнение алгоритма RemoteRealPoint зависит от установленного режима.

Алгоритм включает в себя следующие этапы:

- чтение данных из контроллера-источника;
- диагностика;
- запись.



**Рисунок 3. Схема процесса обработки данных RemoteRealPoint**

#### 4.2.3.1. Чтение данных из контроллера-источника

Чтение данных из функционального блока-источника происходит в режимах AUTO и MASKING.

В атрибут **Source** записывается ссылка на выход функционального блока контроллера-источника.

В атрибут **RemoteConnection** записывается ссылка на функциональный блок RemoteConnection.

В режимах IMITATION и MASK\_IMITATION входные данные вводятся вручную и записываются в атрибут **ManualValue**.

#### 4.2.3.2. Запись

Полученные данные записываются в выходной атрибут **Output**.

#### 4.2.3.3. Диагностика

В процессе диагностики осуществляется проверка коммуникации с контроллером-источником.

В режимах MASKING и MASK\_IMITATION этап диагностики коммуникации отключен.

### 4.3. Функциональный блок RemoteBooleanPoint

RemoteBooleanPoint применяется при создании межконтроллерного соединения. ФБ этого типа подписываются на логические переменные удаленного контроллера.

#### 4.3.1. Атрибуты RemoteBooleanPoint

RemoteBooleanPoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 5. Дополнительные атрибуты RemoteBooleanPoint**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b><i>Входные/выходные атрибуты</i></b>			
Input	BOOL	чтение	Вход
Output	BOOL	чтение	Выход
<b><i>Конфигурационные атрибуты</i></b>			
Source	STRING	чтение/ запись	Источник
ManualValue	BOOL	чтение/ запись	Ручное значение
RemoteConnection	STRING	чтение/ запись	Удаленное соединение

Некоторые значения атрибута *Status* для RemoteBooleanPoint имеют дополнительное описание:

**Таблица 6. Дополнительное описание значений атрибута *Status* для RemoteBooleanPoint**

Имя	Значение	Описание
Uninitialized	0x01010000	Функциональный блок добавлен в проект, но еще не участвовал ни в одном системном цикле



Имя	Значение	Описание
ResolvingReference	0x01030000	Функциональный блок находится в поиске контроллера-источника данных
UncertainSourceStatus	0x01040000	Функциональный блок обнаружил контроллер-источник, но источник данных еще не инициализирован
BadSourceStatus	0x02040000	Атрибут <b>Status</b> функционального блока контроллера-источника имеет значение не Ok
CommunicationError	0x02070000	Отсутствует коммуникация с контроллером-источником

### 4.3.2. Режимы RemoteBooleanPoint

Режимы RemoteBooleanPoint:

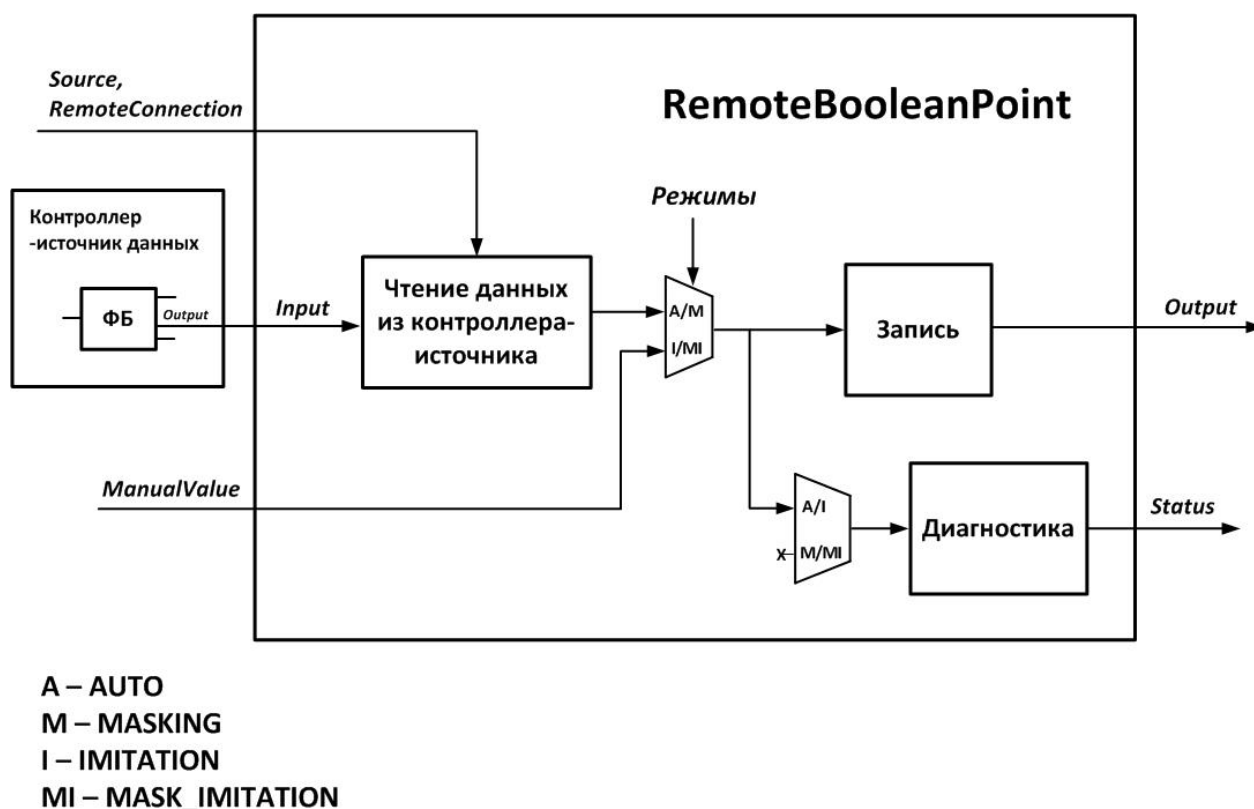
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 4.3.3. Алгоритм RemoteBooleanPoint

Выполнение алгоритма RemoteBooleanPoint зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение данных из контроллера-источника;
- диагностика;
- запись.



**Рисунок 4. Схема процесса обработки данных RemoteBooleanPoint**

#### 4.3.3.1. Чтение данных из контроллера-источника

Чтение данных из функционального блока-источника происходит в режимах AUTO и MASKING.

В атрибут **Source** записывается ссылка на выход функционального блока контроллера-источника.

В атрибут **RemoteConnection** записывается ссылка на функциональный блок RemoteConnection.

В режимах IMITATION и MASK\_IMITATION входные данные вводятся вручную и записываются в атрибут **ManualValue**.

#### 4.3.3.2. Запись

Полученные данные записываются в выходной атрибут **Output**.

#### 4.3.3.3. Диагностика

В процессе диагностики осуществляется проверка коммуникации с контроллером-источником.

В режимах MASKING и MASK\_IMITATION этап диагностики коммуникации отключен.

## 4.4. Функциональный блок RemoteIntegerPoint

RemoteIntegerPoint применяется при создании межконтроллерного соединения. ФБ этого типа подписываются на целочисленные числа удаленного контроллера.

### 4.4.1. Атрибуты RemoteIntegerPoint

RemoteIntegerPoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 7. Дополнительные атрибуты RemoteIntegerPoint

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Входные/выходные атрибуты</b>			
Input	INT	чтение	Вход
Output	INT	чтение	Выход
<b>Конфигурационные атрибуты</b>			
Source	STRING	чтение/ запись	Источник
ManualValue	INT	чтение/ запись	Ручное значение
RemoteConnection	STRING	чтение/ запись	Удаленное соединение

Некоторые значения атрибута *Status* для RemoteIntegerPoint имеют дополнительное описание:

Таблица 8. Дополнительное описание значений атрибута *Status* для RemoteIntegerPoint

Имя	Значение	Описание
Uninitialized	0x01010000	Функциональный блок добавлен в проект, но еще не участвовал ни в одном системном цикле

Имя	Значение	Описание
ResolvingReference	0x01030000	Функциональный блок находится в поиске контроллера-источника данных
UncertainSourceStatus	0x01040000	Функциональный блок обнаружил контроллер-источник, но источник данных еще не инициализирован
BadSourceStatus	0x02040000	Атрибут <b>Status</b> функционального блока контроллера-источника имеет значение не Ok
CommunicationError	0x02070000	Отсутствует коммуникация с контроллером-источником

#### 4.4.2. Режимы RemoteIntegerPoint

Режимы RemoteIntegerPoint:

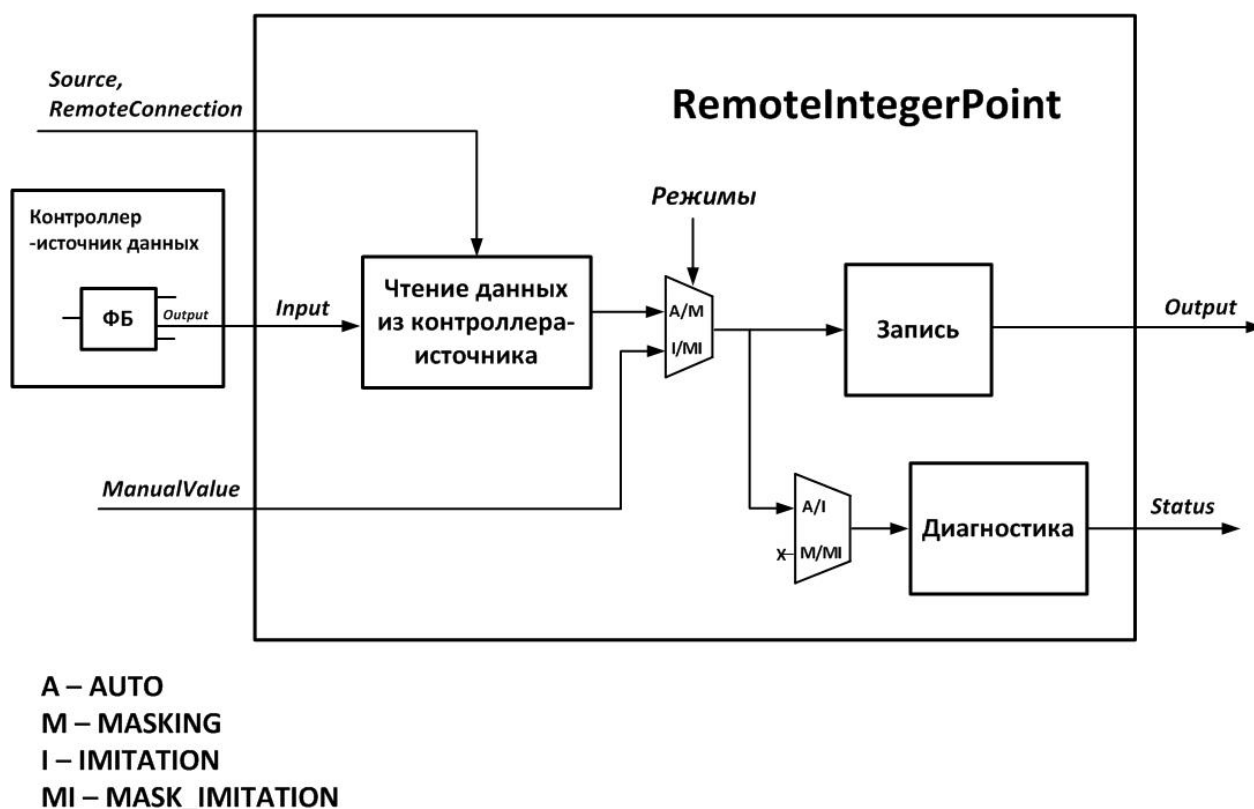
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

#### 4.4.3. Алгоритм RemoteIntegerPoint

Выполнение алгоритма RemoteIntegerPoint зависит от установленного режима.

Алгоритм включает в себя следующие этапы:

- чтение данных из контроллера-источника;
- диагностика;
- запись.



**Рисунок 5. Схема процесса обработки данных RemoteIntegerPoint**

#### 4.4.3.1. Чтение данных из контроллера-источника

Чтение данных из функционального блока-источника происходит в режимах AUTO и MASKING.

В атрибут **Source** записывается ссылка на выход функционального блока контроллера-источника.

В атрибут **RemoteConnection** записывается ссылка на функциональный блок RemoteConnection.

В режимах IMITATION и MASK\_IMITATION входные данные вводятся вручную и записываются в атрибут **ManualValue**.

#### 4.4.3.2. Запись

Полученные данные записываются в выходной атрибут **Output**.

#### 4.4.3.3. Диагностика

В процессе диагностики осуществляется проверка коммуникации с контроллером-источником.

В режимах MASKING и MASK\_IMITATION этап диагностики коммуникации отключен.

## 5. Функциональные блоки для настройки ввода/вывода

Функциональные блоки для настройки ввода/вывода являются частью системного программного обеспечения РСУ.

К данным функциональным блокам относятся:

- Корзина MKRack;
- Модуль питания MK550024PSU;
- Модуль ЦПУ MK502142CPU;
- Модуль аналогового ввода MK513016AI;
- Модуль аналогового ввода MK516008AI;
- Модули дискретного ввода MK521032DI, MK521032ADI;
- Модуль дискретного ввода MK523032ADI;
- Модули аналогового вывода MK514008AO, MK514008AAO;
- Модули дискретного вывода MK531032DO, MK531032ADO;
- Модуль аналогового ввода MK576008AAIHART;
- Модуль аналогового ввода MK576016AAIHART;
- Модуль аналогового вывода MK574008AAO с поддержкой HART;
- Модуль коммуникации MK541002 с поддержкой Modbus.

Применение данных функциональных блоков в среде разработки см. в Руководстве по настройке контроллера РСУ, п. Настройка ввода/вывода данных.

### 5.1. Корзина MKRack

Корзина MKRack (далее - MKRack) является системным функциональным блоком, связанным с аппаратной корзиной ввода/вывода.

Позиции модулей в MKRack должны быть заданы в соответствии с правилами, см. Руководство по настройке контроллера РСУ, п. Позиции модулей в корзине.

#### 5.1.1. Атрибуты MKRack

MKRack имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 9. Дополнительные атрибуты MKRack**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Конфигурационные атрибуты</i>			

Имя атрибута	Тип данных	Доступ	Описание атрибута
Address	INT	чтение/запись	Адрес корзины
<i>Сигнализационные атрибуты</i>			
NotAvailable	BOOL	чтение	Доступность. Значение атрибута true показывает, что ни один модуль корзины не доступен PCY

### 5.1.2. Режимы MKRack

Режимы MKRack:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

## 5.2. Модуль питания MK550024PSU

Модуль питания MK550024PSU (далее MK550024PSU) обеспечивает выполнение следующих функций:

- считывание и запись выходного напряжения аппаратного модуля;
- формирование сигнализаций;
- формирование данных для состояния сигнализаций в планшете сигнализаций.

### 5.2.1. Атрибуты MK550024PSU

MK550024PSU имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 10. Дополнительные атрибуты MK550024PSU**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Конфигурационные атрибуты</i>			
Rack	STRING	чтение/ запись	Корзина
ModulePosition	INT	чтение/ запись	Позиция модуля питания
<i>Диагностические атрибуты</i>			

Имя атрибута	Тип данных	Доступ	Описание атрибута
OutputVoltage	REAL	чтение	Выходное напряжение
<i>Сигнализационные атрибуты</i>			
BusError1	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BusError2	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)
LowVoltage	BOOL	чтение	Низкое напряжение

Значение атрибута **Status** шин ввода/вывода при любом состоянии имеет значение 0x00000000 (ОК).

### 5.2.2. Режимы МК550024PSU

Режимы МК550024PSU:

- OFF ("Выключен") - алгоритм не выполняется;
- AUTO ("Автоматический") - алгоритм выполняется в полном объеме;
- MASKING ("Маскирование") - алгоритм выполняется за исключением этапа формирования сигнализаций.

## 5.3. Модуль ЦПУ МК502142CPU

Модуль ЦПУ МК502142CPU (далее МК502142CPU) обеспечивает выполнение следующих функций:

- запуск технологической программы;
- диагностика коммуникации с сетевыми интерфейсами;
- резервирование и диагностика синхронизации;
- формирование сигнализаций.

Максимальная возможная загрузка ЦП для МК502142CPU — 85%.

### 5.3.1. Атрибуты МК502142CPU

МК502142CPU имеет базовый набор атрибутов, а также дополнительные атрибуты:



Таблица 11. Дополнительные атрибуты МК502142CPU

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Конфигурационные атрибуты</i>			
RACK	STRING	чтение/ запись	Корзина
SLOT	DINT	чтение/ запись	Позиция модуля ЦПУ
<i>Диагностические атрибуты</i>			
CPU_LOAD	REAL	чтение	Загрузка модуля ЦПУ, %
SYNC_ROLE	DINT	чтение	Роль модуля ЦПУ (primary(1)/secondary(2))
SYNC_STS	DINT	чтение	Текущее состояние модуля ЦПУ (stop(0)/active(1)/standby(2)/error(3)/disabled(4))
UTIL_MEM	DINT	чтение	Память
<i>Сигнализационные атрибуты</i>			
ETH1_ERR... ETH4_ERR	BOOL	чтение	Ошибка каналов Ethernet1...Ethernet4
FO_ERR	BOOL	чтение	Ошибка оптического сетевого интерфейса
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)
SYNC_ERR	BOOL	чтение	Ошибка синхронизации
SW_TO_STBY	BOOL	чтение	Переход в пассивный режим
SW_TO_ACTV	BOOL	чтение	Переход в активный режим

### 5.3.2. Режимы МК502142CPU

Режимы МК502142CPU:

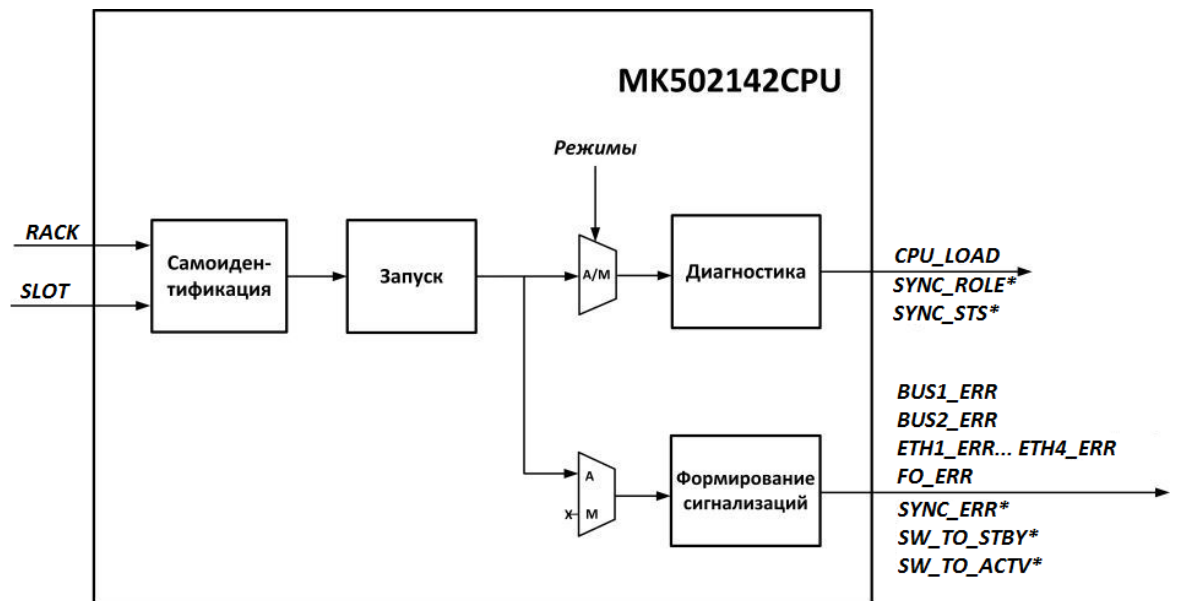
- OFF ("Выключен") - алгоритм не выполняется;
- AUTO ("Автоматический") - алгоритм выполняется в полном объеме;
- MASKING ("Маскирование") - алгоритм выполняется за исключением этапа формирования сигнализаций.

### 5.3.3. Алгоритм МК502142CPU

Выполнение алгоритма МК502142CPU зависит от установленного режима.

Алгоритм МК502142CPU включает в себя следующие элементы:

- самоидентификация;
- коммуникация;
- диагностика;
- формирование сигнализаций.



A – AUTO  
M – MASKING

\*Для контроллера РСУ с резервированием

Рисунок 6. Схема процесса обработки данных МК502142CPU

#### 5.3.3.1. Самоидентификация

На данном этапе происходит вычисление адреса аппаратного модуля ЦПУ на шине ввода/вывода.

Вычисление производится по атрибутам **RACK** и **SLOT**.

В результате вычислений определяется роль МК502142CPU:

- "current" - совпадение адреса аппаратного модуля ЦПУ на шине ввода/вывода с атрибутами **RACK** и **SLOT**.
- "redundant" - при несовпадении адреса аппаратного модуля ЦПУ на шине ввода/вывода с атрибутами **RACK** и **SLOT** и наличии резервирования в контроллере.

В случае несовпадения вычисленного адреса и отсутствии резервирования формируется сообщение об ошибке конфигурации.

### 5.3.3.2. Запуск

Элементы этапа запуска различаются в зависимости от наличия резервирования и роли МК502142CPU:

**Таблица 12. Элементы этапа запуска МК502142CPU**

Резервирование		
<i>Есть</i>		<i>Нет</i>
Роль модуля ЦПУ		<ul style="list-style-type: none"> <li>• осуществляет физическую настройку шин ввода/вывода;</li> <li>• запускает драйвер ввода/вывода;</li> <li>• запускает технологическую программу.</li> </ul>
<i>current</i>	<i>redundant</i>	
<ul style="list-style-type: none"> <li>• осуществляет физическую настройку шин ввода/вывода;</li> <li>• запускает драйвер ввода/вывода;</li> <li>• настраивает драйвер ввода/вывода на работу в активном либо пассивном режиме в зависимости от значения <b><i>SYNC_STS</i></b>;</li> <li>• запускает технологическую программу.</li> </ul>	<ul style="list-style-type: none"> <li>• получает и записывает данные о коммуникации от соседнего аппаратного модуля.</li> </ul>	

### 5.3.3.3. Диагностика

Элементы этапа диагностики различаются в зависимости от наличия резервирования и роли МК502142CPU:

**Таблица 13. Элементы этапа диагностики МК502142CPU**

Резервирование	
<i>Есть</i>	<i>Нет</i>

Роль модуля ЦПУ		
<i>current</i>	<i>redundant</i>	
<ul style="list-style-type: none"> <li>• диагностирует коммуникацию сетевых интерфейсов;</li> <li>• диагностирует синхронизацию с соседним аппаратным модулем ЦПУ;</li> <li>• формирует и записывает данные для диагностических атрибутов (Таблица 11. Дополнительные атрибуты МК502142CPU).</li> </ul>	<ul style="list-style-type: none"> <li>• получает данные о текущем состоянии сетевых интерфейсов от соседнего аппаратного модуля ЦПУ;</li> <li>• получает данные о значении диагностических атрибутов соседнего аппаратного модуля ЦПУ;</li> <li>• получает данные от основного модуля ЦПУ о текущей синхронизации с соседним аппаратным модулем ЦПУ.</li> </ul>	<ul style="list-style-type: none"> <li>• диагностирует коммуникацию сетевых интерфейсов;</li> <li>• формирует и записывает данные для диагностических атрибутов (Таблица 11. Дополнительные атрибуты МК502142CPU).</li> </ul>

#### 5.3.3.4. Формирование сигнализаций

Формирование сигнализаций зависит от наличия резервирования.

МК502142CPU в контроллере с резервированием формирует следующие сигнализации:

1. Сигнализации, вызванные ошибками коммуникации сетевых интерфейсов и ошибками коммуникации между соседними модулями:

- ETH1\_ERR... ETH4\_ERR;
- FO\_ERR;
- BUS1\_ERR;
- BUS2\_ERR.

2. Сигнализации, вызванные событиями в подсистеме синхронизации:

- SYNC\_ERR;
- SW\_TO\_STBY;
- SW\_TO\_ACTV.

МК502142CPU в контроллере без резервирования формирует сигнализации только из группы 1.

## 5.4. Модуль аналогового ввода МК513016AI

Модуль аналогового ввода МК513016AI (далее МК513016AI) обеспечивает выполнение следующих функций:

- преобразование аналоговых сигналов в значения физической величины для записи в выходные атрибуты;
- формирование сигнализаций.

### 5.4.1. Атрибуты МК513016AI

Кроме базового набора атрибутов МК513016AI имеет дополнительные атрибуты:

**Таблица 14. Дополнительные атрибуты МК513016AI**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b><i>Выходные атрибуты</i></b>			
Output1... Output16	REAL	чтение	Выход 1...Выход 16 - значение соответствующего канала в процентах
<b><i>Конфигурационные атрибуты</i></b>			
Rack	STRING	чтение/ запись	Имя корзины, в которую установлен аппаратный модуль
ModulePosition	INT	чтение/ запись	Позиция модуля в корзине
Type1...Type16	INT	чтение/ запись	Тип 1...Тип 16 - тип аналогового входа (0...20 мА либо 4...20 мА)
ManualValue1... ManualValue16	REAL	чтение/ запись	Ручное значение 1...Ручное значение 16 - значение для записи в выходной атрибут в режимах IMITATION и MASK_IMITATION в процентах
<b><i>Диагностические атрибуты</i></b>			
Raw1...Raw16	INT	чтение	Код АЦП1...Код АЦП 16

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Сигнализационные атрибуты</b>			
BusError1	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BusError2	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)

Некоторые значения атрибута *Status* для МК513016AI имеют дополнительное описание:

**Таблица 15. Дополнительное описание значений атрибута *Status* МК513016AI**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружена корзина
BadConfiguration	0x02030000	Ошибка конфигурации модуля
CommunicationError	0x02070000	Отсутствует коммуникация с модулем ввода/вывода
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

#### 5.4.2. Режимы МК513016AI

Режимы МК513016AI:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

#### 5.4.3. Алгоритм МК513016AI

Выполнение алгоритма МК513016AI зависит от установленного режима.

Алгоритм МК513016AI включает в себя следующие элементы:

- диагностика;

- формирование сигнализаций;
- преобразование и запись.



A – AUTO  
M – MASKING  
I – IMITATION  
MI – MASK\_IMITATION

Рисунок 7. Схема процесса обработки данных MK513016AI

### 5.4.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных MK513016AI данным модуля ввода/вывода.

Элементы диагностики:

1. Проверка рабочего состояния модуля ввода/вывода и наличие связи с шиной ввода/вывода.
2. Самоидентификация.

Вычисляется номер корзины по атрибуту *Rack*.

Вычисляется уникальный номер MK513016AI. При вычислении используется номер корзины и атрибут *ModulePosition*.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

### 5.4.3.2. Формирование сигнализаций

МК513016AI формирует сигнализации в следующих случаях:

- неисправность аппаратного модуля или отсутствие связи с шиной ввода/вывода (атрибут сигнализации **BusError1** либо **BusError2** принимает значение true);
- ошибка вычисления или соответствия идентификационных данных МК513016AI (атрибуты сигнализации **BusError1** и **BusError2** принимают значение true).

### 5.4.3.3. Преобразование и запись

Производится при положительном результате диагностики.

В режимах AUTO и MASKING данный этап включает в себя следующие элементы:

- преобразование значений аналоговых сигналов из атрибутов **Raw1...Raw16** в процентном отношении от физической величины. При этом для каждого из указанных атрибутов учитывается тип аналогового входа **Type1...Type16**. Тип аналогового входа может быть выбран 0...20 мА либо 4...20 мА.
- запись преобразованных данных в выходные атрибуты **Output1...Output16**.

В режимах IMITATION и MASK\_IMITATION преобразование не происходит, данные в выходные атрибуты записываются из значений **ManualValue1...ManualValue16** в процентах.

## 5.5. Модуль аналогового ввода МК516008AI

Модуль аналогового ввода МК516008AI (далее МК516008AI) обеспечивает выполнение следующих функций:

- преобразование аналоговых сигналов в значения физической величины для записи в выходные атрибуты;
- формирование сигнализаций.

### 5.5.1. Атрибуты МК516008AI

Кроме базового набора атрибутов МК516008AI имеет дополнительные атрибуты:

**Таблица 16. Дополнительные атрибуты МК516008AI**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Выходные атрибуты</b>			



Имя атрибута	Тип данных	Доступ	Описание атрибута
Output 1... Output 8	REAL	чтение	Выход 1...Выход 8 - значение соответствующего канала в процентах
<b>Конфигурационные атрибуты</b>			
Rack	STRING	чтение/ запись	Имя корзины, в которую установлен аппаратный модуль
Module Position	INT	чтение/ запись	Позиция модуля в корзине
Type 1...Type 8	INT	чтение/ запись	Тип 1...Тип 8 - тип аналогового входа (0...20 мА либо 4...20 мА)
ManualValue1... ManualValue8	REAL	чтение/ запись	Ручное значение 1...Ручное значение 8 - значение для записи в выходной атрибут в режимах IMITATION и MASK_IMITATION в процентах
<b>Диагностические атрибуты</b>			
Raw 1...Raw 8	INT	чтение	Код АЦП 1...Код АЦП 8
<b>Сигнализационные атрибуты</b>			
Bus Error 1	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
Bus Error 2	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)

### 5.5.2. Режимы МК516008AI

Режимы МК516008AI:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");

- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 5.5.3. Алгоритм МК516008AI

Выполнение алгоритма МК516008AI зависит от установленного режима.

Алгоритм МК516008AI включает в себя следующие элементы:

- диагностика;
- формирование сигнализаций;
- преобразование и запись.



A – AUTO  
M – MASKING  
I – IMITATION  
MI – MASK\_IMITATION

Рисунок 8. Схема процесса обработки данных МК516008AI

#### 5.5.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных МК516008AI данным модуля ввода/вывода.

Элементы диагностики:

1. Проверка рабочего состояния модуля ввода/вывода и наличие связи с шиной ввода/вывода.
2. Самоидентификация.

Вычисляется номер корзины по атрибуту **Rack**.

Вычисляется уникальный номер МК516008АІ. При вычислении используется номер корзины и атрибут **ModulePosition**.

### 3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

#### 5.5.3.2. Формирование сигнализаций

МК516008АІ формирует сигнализации в следующих случаях:

- неисправность аппаратного модуля или отсутствие связи с шиной ввода/вывода (атрибут сигнализации **BusError1** либо **BusError2** принимает значение true);
- ошибка вычисления или соответствия идентификационных данных МК516008АІ (атрибуты сигнализации **BusError1** и **BusError2** принимают значение true)

#### 5.5.3.3. Преобразование и запись

Производится при положительном результате диагностики.

В режимах AUTO и MASKING данный этап включает в себя следующие элементы:

- преобразование значений аналоговых сигналов из атрибутов **Raw1...Raw8** в процентном отношении от физической величины. При этом для каждого из указанных атрибутов учитывается тип аналогового входа **Type1...Type8**. Тип аналогового входа может быть выбран 0...20 мА либо 4...20 мА.
- запись преобразованных данных в выходные атрибуты **Output1...Output8**.

В режимах IMITATION и MASK\_IMITATION преобразование не происходит, данные в выходные атрибуты записываются из значений **ManualValue1...ManualValue8** в процентах.

## 5.6. Модули дискретного ввода МК521032DІ и МК521032ADІ

Модули дискретного ввода МК521032DІ и МК521032ADІ (далее МК521032DІ) обеспечивает выполнение следующих функций:

- запись дискретных сигналов модуля ввода/вывода в выходные атрибуты;
- формирование сигнализаций.



**Внимание:** Модуль MK521032ADI является модификацией модуля MK521032DI.

### 5.6.1. Атрибуты MK521032DI

Кроме базового набора атрибутов MK521032DI имеет дополнительные атрибуты:

**Таблица 17. Дополнительные атрибуты MK521032DI**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Выходные атрибуты</i>			
OUT1... OUT32	BOOL	чтение	Выход 1...Выход 32 - значение соответствующего канала
<i>Конфигурационные атрибуты</i>			
RACK	STRING	чтение/ запись	Имя корзины, в которую установлен аппаратный модуль
SLOT	DINT	чтение/ запись	Позиция модуля в корзине
MV1... MV32	BOOL	чтение/ запись	Ручное значение 1...Ручное значение 32 - значение, вводимое в режимах IMITATION и MASK_IMITATION
<i>Сигнализационные атрибуты</i>			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)

Некоторые значения атрибута *Status* для MK521032DI имеют дополнительное описание:

**Таблица 18. Дополнительное описание значений атрибута *Status* MK521032DI**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружена корзина
BadConfiguration	0x02030000	Ошибка конфигурации модуля
CommunicationError	0x02070000	Отсутствует коммуникация с модулем ввода/вывода
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

### 5.6.2. Режимы MK521032DI

Режимы MK521032DI:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 5.6.3. Алгоритм MK521032DI

Выполнение алгоритма MK521032DI зависит от установленного режима.

Алгоритм MK521032DI включает в себя следующие элементы:

- диагностика;
- формирование сигнализаций;
- запись данных модуля ввода/вывода.

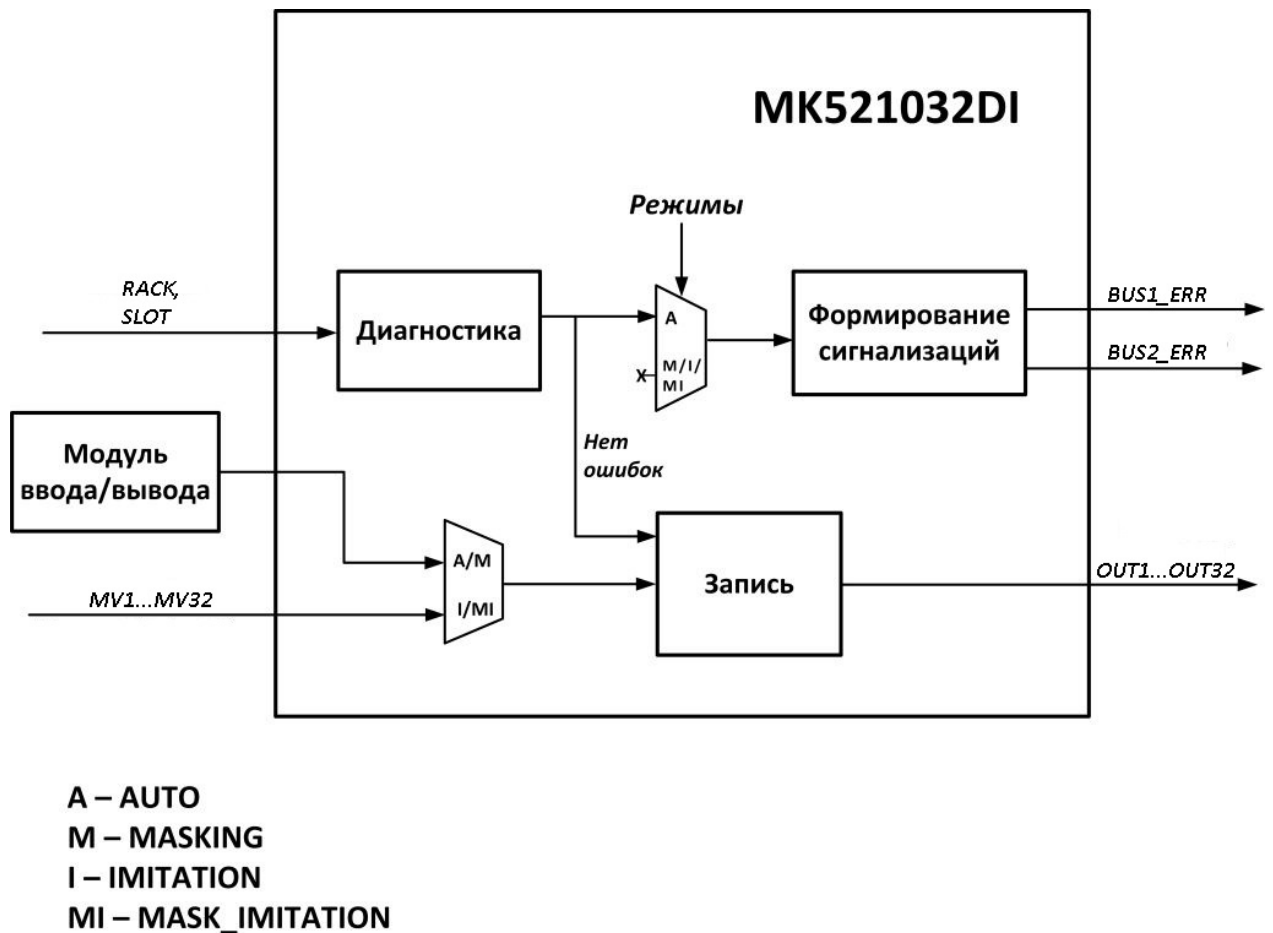


Рисунок 9. Схема процесса обработки данных MK521032DI

### 5.6.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных MK521032DI данным модуля ввода/вывода.

Элементы диагностики:

1. Проверка рабочего состояния модуля ввода/вывода. Проверка наличия связи с шиной ввода/вывода.
2. Самоидентификация.

Вычисляется номер корзины по атрибуту **RACK**.

Вычисляется уникальный номер MK521032DI. При вычислении используется номер корзины и атрибут **SLOT**.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

### 5.6.3.2. Формирование сигнализаций

MK521032DI формирует сигнализации в следующих случаях:

- неисправность модуля ввода/вывода или отсутствие связи с шиной ввода/вывода (атрибут сигнализации **BUS1\_ERR** либо **BUS2\_ERR** принимает значение true);
- ошибка вычисления или соответствия идентификационных данных МК521032DI (атрибуты сигнализации **BUS1\_ERR** и **BUS2\_ERR** принимают значение true).

### 5.6.3.3. Запись данных модуля ввода/вывода

Производится при положительном результате диагностики.

В режимах AUTO и MASKING данные в выходные атрибуты записываются из значений, полученных от модуля ввода/вывода.

В режимах IMITATION и MASK\_IMITATION данные в выходные атрибуты записываются из ручных значений *MV1...MV32*.

## 5.7. Модуль дискретного ввода МК523032ADI

Модуль дискретного ввода МК523032ADI (далее МК523032ADI) обеспечивает выполнение следующих функций:

- запись дискретных сигналов модуля ввода/вывода в выходные атрибуты;
- формирование сигнализаций;
- поддержка сигналов NAMUR;
- работа в составе резервной пары.

### 5.7.1. Атрибуты МК523032ADI

Кроме базового набора атрибутов МК523032ADI имеет дополнительные атрибуты:

**Таблица 19. Дополнительные атрибуты МК523032ADI**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Выходные атрибуты</b>			
OUT1...OUT32	BOOL	чтение	Выход 1...Выход 32 - значение соответствующего канала
<b>Конфигурационные атрибуты</b>			

Имя атрибута	Тип данных	Доступ	Описание атрибута
RACK	STRING	чтение/ запись	Имя корзины, в которую установлен аппаратный модуль
SLOT	DINT	чтение/ запись	Позиция модуля в корзине
MV1...MV32	BOOL	чтение/ запись	Ручное значение 1...Ручное значение 32 - значение, вводимое в режимах IMITATION и MASK_IMITATION
STS1_ALM_EN... STS32_ALM_EN	BOOL	чтение/ запись	Включение сигнализации 1...Включение сигнализации 32
RDNCY_ENA	BOOL	чтение/ запись	Включение режима резервирования
RDNCY_MODE	USINT	чтение/ запись	Резервный режим
NMATCH_RETR	UDINT	чтение/ запись	Нет повторных попыток совпадения каналов
<b><i>Сигнализационные атрибуты</i></b>			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода (основной) 1
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода (резервной) 2
STS1_ALM... STS32_ALM	BOOL	чтение	Состояние сигнализации 1...Состояние сигнализации 32
<b><i>Диагностические атрибуты</i></b>			



Имя атрибута	Тип данных	Доступ	Описание атрибута
STS1...STS32	USINT	чтение	Статус входа 1...Статус входа 32

Некоторые значения атрибута *Status* для МК523032АДІ имеют дополнительное описание:

**Таблица 20. Дополнительное описание значений атрибута *Status* МК523032АДІ**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не загружена корзина
BadConfiguration	0x02030000	Ошибка конфигурации модуля
CommunicationError	0x02070000	Отсутствует коммуникация с модулем ввода/вывода
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

### 5.7.2. Режимы МК523032АДІ

Режимы МК523032АДІ:

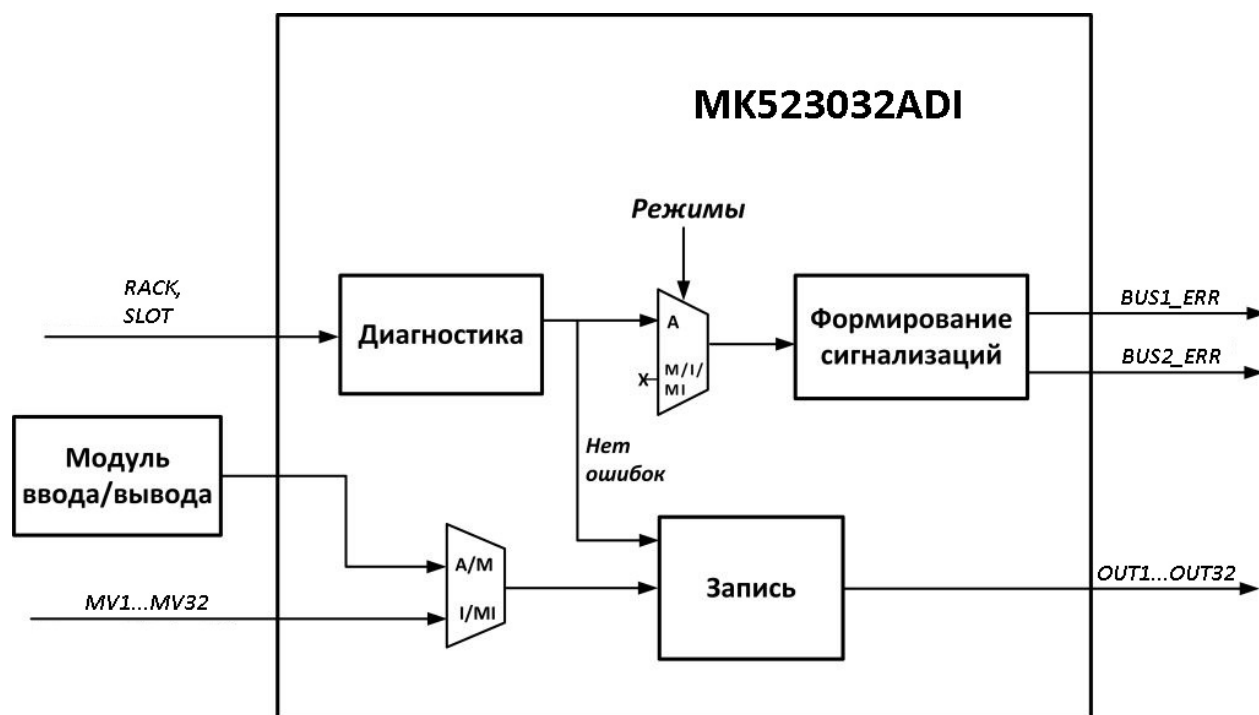
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 5.7.3. Алгоритм МК523032АДІ

Выполнение алгоритма МК523032АДІ зависит от установленного режима.

Алгоритм МК523032АДІ включает в себя следующие элементы:

- диагностика;
- формирование сигнализаций;
- запись данных модуля ввода/вывода.



**A – AUTO**  
**M – MASKING**  
**I – IMITATION**  
**MI – MASK\_IMITATION**

**Рисунок 10. Схема процесса обработки данных MK523032ADI**

1.

### 5.7.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных MK523032ADI данным модуля ввода/вывода.

Элементы диагностики:

1. Проверка рабочего состояния модуля ввода/вывода. Проверка наличия связи с шиной ввода/вывода.
2. Самоидентификация.

Вычисляется номер корзины по атрибуту **RACK**.

Вычисляется уникальный номер MK521032DI. При вычислении используется номер корзины и атрибут **SLOT**.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

### 5.7.3.2. Формирование сигнализаций

MK523032ADI формирует сигнализации в следующих случаях:

- неисправность модуля ввода/вывода или отсутствие связи с шиной ввода/вывода (атрибут сигнализации **BUS1\_ERR** либо **BUS2\_ERR** принимает значение true);
- ошибка вычисления или соответствия идентификационных данных MK523032ADI (атрибуты сигнализации **BUS1\_ERR** и **BUS2\_ERR** принимают значение true).

### 5.7.3.3. Запись данных модуля ввода/вывода

Производится при положительном результате диагностики.

В режимах AUTO и MASKING данные в выходные атрибуты записываются из значений, полученных от модуля ввода/вывода.

В режимах IMITATION и MASK\_IMITATION данные в выходные атрибуты записываются из ручных значений **MV1...MV32**.

### 5.7.3.4. Поддержка сигналов NAMUR

MK523032ADI предназначен для ввода сигналов от датчиков стандарта *NAMUR*.

Помимо логического уровня вход имеет следующие статусные состояния:

- входная линия в "обрыве";
- входная линия в "коротком замыкании".

Статусные состояния позволяют определить качество входного сигнала и выдать аварию на верхний уровень.

В MK523032ADI предусмотрены выходы:

- **OUT1... OUT32** - логическое состояние входа от датчика;
- **STS1...STS32** - статус входа;
- **STS1\_ALM...STS32\_ALM** - авария, которая может быть заблокирована индивидуально атрибутом **STS1\_ALM\_EN...STS32\_ALM\_EN** или переводом модуля в режим MASKING.

MK523032ADI формирует следующие выходные статусы каналов:

- "ActiveFirst": 1 - канал первого модуля пары в состоянии логической 1;
- "OpenCircuitFirst": 2 - канал первого модуля в состоянии "обрыв цепи";
- "ShortCircuitFirst": 4 - канал первого модуля в состоянии "короткое замыкание";
- "NotRespFirst": 8 - первый модуль не отвечает;
- "ActiveSecond": 16 - канал второго модуля пары в состоянии логической 1;

- "OpenCircuitSecond": 32 - канал второго модуля в состоянии "обрыв цепи";
- "ShortCircuitSecond": 64 - канал второго модуля в состоянии "короткое замыкание";
- "NotRespSecond": 128 - второй модуль не отвечает;
- "RedundDataNMatch": 256 - логические состояния соответствующего канала обоих модулей не совпадают.

### 5.7.3.5. Работа в составе резервной пары

MK523032ADI поддерживает работу в составе резервной пары.

Режим резервирования включается с помощью атрибута *RDNCY\_ENA*.

Типы резервирования MK523032ADI:

1. AllSlots - позволяет автоматически выбирать какой из каналов пары модулей сформирует выходной сигнал функционального блока;
2. OnlyFirstSlot - данные берутся из первого модуля, но инициализируются оба модуля из резервной пары;
3. OnlySecondSlot - данные берутся из второго модуля, но инициализируются оба модуля из резервной пары.

В режиме резервирования тип AllSlots осуществляет выбор канала следующим образом:

Производится поканальный обход статусов обоих модулей. Если статус соответствующего канала обоих модулей без "обрыва" или "короткого замыкания", то логическое состояние каналов сравнивается. Если оно - одинаковое, то формируется выходное состояние атрибута *OUT1...OUT32*. Если логическое состояние каналов различается, и канал одного из модулей в 1, а второго в 0, то функциональный блок производит проверку на число несовпадений подряд с помощью атрибута *NMATCH\_RETR*, и если оно превышено, выставляется дополнительный статус канала "RedundDataNMatch" и вводится атрибут *STS1\_ALM...STS32\_ALM*.



#### **Внимание:**

Если статус канала одного из модулей в режиме "обрыв" или "короткое замыкание", берется выход другого, но авария не выставляется. При этом в выходном статусе канала функционального блока отобразится канал какого из модулей вышел в ошибку. Если статусы каналов обоих модулей в состоянии "обрыв цепи" или "короткое замыкание", *OUT1...OUT32* функционального блока сохранит предыдущее состояние, будет выставлена авария и соответствующий статус.

При включении режима резервирования системой занимается дополнительная позиция в корзине, соответствующая номеру **SLOT**+1 для установки резервного модуля. Таким образом, нумерация для следующих модулей в корзине продолжится с адреса **SLOT**+2.

## 5.8. Модули аналогового вывода МК514008АО и МК514008ААО

Модули аналогового вывода МК514008АО и МК514008ААО (далее МК514008АО) обеспечивают выполнение следующих функций:

- формирование выходных аналоговых сигналов;
- формирование сигнализаций.



**Внимание:** Модуль МК514008ААО является модификацией модуля МК514008АО. Отличие состоит в том, что исполнение МК514008ААО выполнено с быстросъемным разъемом в 40 контактов, тогда как у МК514008АО разъем в 20 контактов.

### 5.8.1. Атрибуты МК514008АО

Кроме базового набора атрибутов МК514008АО имеет дополнительные атрибуты:

**Таблица 21. Дополнительные атрибуты МК514008АО**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b><i>Входные атрибуты</i></b>			
Input1... Input8	REAL	чтение	Вход 1...Вход 8 - значение соответствующего канала в процентах
<b><i>Выходные атрибуты</i></b>			
ChannelError1... ChannelError8	BOOL	чтение	Ошибка канала 1...Ошибка канала 8
<b><i>Конфигурационные атрибуты</i></b>			

Имя атрибута	Тип данных	Доступ	Описание атрибута
Rack	STRING	чтение/ запись	Имя корзины, в которую установлен модуль ввода/вывода
ModulePosition	INT	чтение/ запись	Позиция модуля в корзине
Source1...Source8	STRING	чтение/ запись	Источник 1...Источник 8
Type1...Type8	INT	чтение/ запись	Тип 1...Тип 8 - тип входов (0...20 мА либо 4...20 мА)
ManualValue1... ManualValue8	REAL	чтение/ запись	Ручное значение 1...Ручное значение 8 - значение в режиме MANUAL в процентах
<b>Диагностические атрибуты</b>			
Raw1...Raw8	INT	чтение	Код ЦАП 1...Код ЦАП 8
<b>Сигнализационные атрибуты</b>			
BusError1	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BusError2	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)

Некоторые значения атрибута *Status* для МК514008АО имеют дополнительное описание:

**Таблица 22. Дополнительное описание значений атрибута *Status* МК514008АО**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружена корзина
BadConfiguration	0x02030000	Ошибка конфигурации модуля
CommunicationError	0x02070000	Отсутствует коммуникация с модулем ввода/вывода

Имя	Значение	Описание
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

### 5.8.2. Режимы МК514008АО

Режимы МК514008АО:

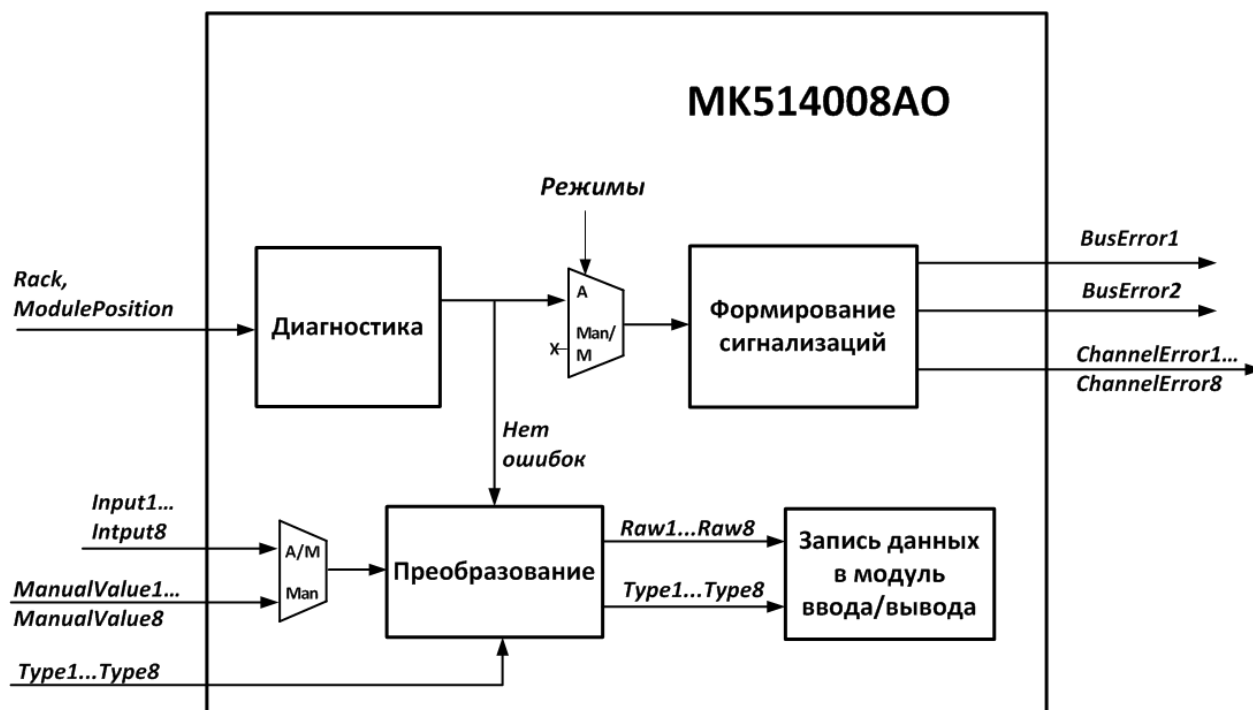
- OFF ("Выключен");
- AUTO ("Автоматический");
- MANUAL ("Ручной")
- MASKING ("Маскирование").

### 5.8.3. Алгоритм МК514008АО

Выполнение алгоритма МК514008АО зависит от установленного режима.

Алгоритм МК514008АО включает в себя следующие элементы:

- диагностика;
- формирование сигнализаций;
- преобразование;
- запись данных в модуль ввода/вывода.



A – AUTO  
 Man – MANUAL  
 M – MASKING

Рисунок 11. Схема процесса обработки данных МК514008АО

### 5.8.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных МК514008АО данным модуля ввода/вывода.

Элементы диагностики:

1. Проверка рабочего состояния модуля ввода/вывода и наличие связи с шиной ввода/вывода.
2. Самоидентификация.

Вычисляется номер корзины по атрибуту *Rack*.

Вычисляется уникальный номер МК514008АО. При вычислении используется номер корзины и атрибут *ModulePosition*.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

### 5.8.3.2. Формирование сигнализаций

МК514008АО формирует сигнализации в следующих случаях:

- неисправность модуля ввода/вывода или отсутствие связи с шиной ввода/вывода (атрибут сигнализации *BusError1* либо *BusError2* принимает значение true);
- ошибка вычисления или соответствия идентификационных данных МК514008АО (атрибуты сигнализации *BusError1* и *BusError2* принимают значение true).

На этапе формирования сигнализаций также производится запись в выходные атрибуты *ChannelError1...ChannelError8*. Указанные атрибуты содержат данные о наличии ошибки каналов. Данные о наличии ошибки предоставляются модулем ввода/вывода.

### 5.8.3.3. Преобразование

Этап преобразования включает в себя следующие элементы:

- преобразование входных значений в физических единицах в аналоговые сигналы. Тип входных значений может быть выбран 0...20 мА либо 4...20 мА и записывается в атрибуты *Type1...Type8*.
- запись преобразованных данных в выходные атрибуты *Raw1...Raw8*.

### 5.8.3.4. Запись данных в модуль ввода/вывода

Преобразованные данные записываются в модуль ввода/вывода из атрибутов *Raw1...Raw8*.



## 5.9. Модули дискретного вывода МК531032DO и МК531032ADO

Модули дискретного вывода МК531032DO и МК531032ADO (далее МК531032DO) обеспечивает выполнение следующих функций:

- отображение выходных дискретных сигналов модуля ввода/вывода;
- формирование сигнализаций.



**Внимание:** Модуль МК531032ADO является модификацией модуля МК531032DO.

### 5.9.1. Атрибуты МК531032DO

МК531032DO имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 23. Дополнительные атрибуты МК531032DO**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b><i>Входные атрибуты</i></b>			
IN1... IN32	BOOL	чтение	Вход 1...Вход 32 (значение соответствующего канала)
<b><i>Конфигурационные атрибуты</i></b>			
RACK	STRING	чтение/ запись	Имя корзины модуля ввода/вывода
SLOT	DINT	чтение/ запись	Позиция модуля в корзине
IN1_SRC... IN32_SRC	STRING	чтение/ запись	Источник 1...Источник 32
MV1...MV32	BOOL	чтение/ запись	Ручное значение 1...Ручное значение 32 - значение, вводимое в режиме MANUAL

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Сигнализационные атрибуты</b>			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)

Некоторые значения атрибута *Status* для МК531032DO имеют дополнительное описание:

**Таблица 24. Дополнительное описание значений атрибута *Status* МК531032DO**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не загружена корзина
BadConfiguration	0x02030000	Ошибка конфигурации модуля
CommunicationError	0x02070000	Отсутствует коммуникация с модулем ввода/вывода
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

### 5.9.2. Режимы МК531032DO

Режимы МК531032DO:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MANUAL ("Ручной");
- MASKING ("Маскирование").

### 5.9.3. Алгоритм МК531032DO

Выполнение алгоритма МК531032DO зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- диагностика;
- формирование сигнализаций;
- запись данных в модуль ввода/вывода.



A – AUTO  
 Man – MANUAL  
 M – MASKING

Рисунок 12. Схема процесса обработки данных MK531032DO

### 5.9.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных MK531032DO данным модуля ввода/вывода.

Элементы диагностики:

1. Проверка рабочего состояния модуля ввода/вывода. Проверка наличия связи с шиной ввода/вывода.
2. Самоидентификация.

Вычисляется номер корзины по атрибуту **RACK**.

Вычисляется уникальный номер MK531032DO. При вычислении используется номер корзины и атрибут **SLOT**.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке **Bad**.

### 5.9.3.2. Формирование сигнализаций

МК531032DO формирует сигнализации в следующих случаях:

- неисправность аппаратного модуля или отсутствие связи с шиной ввода/вывода (атрибут сигнализации **BUS1\_ERR** либо **BUS2\_ERR** принимает значение true);
- ошибка вычисления или соответствия идентификационных данных МК531032DO (атрибуты сигнализации **BUS1\_ERR** и **BUS2\_ERR** принимают значение true).

### 5.9.3.3. Запись данных в модуль ввода/вывода

В режимах AUTO и MASKING данные в модуль ввода/вывода записываются из атрибутов **IN1...IN32**.

В режиме MANUAL данные в модуль ввода/вывода записываются из атрибутов **MV1...MV32**.

## 5.10. Модуль аналогового ввода МК576008AAIHART

Модуль аналогового ввода МК576008AAIHART (далее МК576008AAIHART) обеспечивает выполнение следующих функций:

- преобразование аналоговых сигналов в значения физической величины для записи в выходные атрибуты;
- преобразование полученных данных из HART каналов в значения физической величины для записи в соответствующие выходные атрибуты;
- формирование сигнализаций.

### 5.10.1. Атрибуты МК576008AAIHART

Кроме базового набора атрибутов МК576008AAIHART имеет дополнительные атрибуты:

**Таблица 25. Дополнительные атрибуты МК576008AAIHART**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Выходные атрибуты</b>			
Output1... Output8	REAL	чтение	Выход 1...Выход 8 - значение соответствующего канала в процентах

Имя атрибута	Тип данных	Доступ	Описание атрибута
HARTPrimaryVariable1... HARTPrimaryVariable8	REAL	чтение	Значение первой переменной HART канала 1...8
HARTSecondaryVariable1... HARTSecondaryVariable8	REAL	чтение	Значение второй переменной HART канала 1...8
HARTTertiaryVariable1... HARTTertiaryVariable8	REAL	чтение	Значение третьей переменной HART канала 1...8
HARTQuaternaryVariable1... HARTQuaternaryVariable8	REAL	чтение	Значение четвертой переменной HART канала 1...8
HARTError1... HARTError8	BOOL	чтение	Соответствующий HART канал в состоянии ошибки
<b><i>Конфигурационные атрибуты</i></b>			
Rack	STRING	чтение/ запись	Имя корзины, в которую установлен аппаратный модуль
ModulePosition	INT	чтение/ запись	Позиция модуля в корзине
Type1...Type8	INT	чтение/ запись	Тип 1...Тип 8 - тип аналогового входа (0...20 мА либо 4...20 мА)
ManualValue1... ManualValue8	REAL	чтение/ запись	Ручное значение 1...Ручное значение 8 - значение для записи в выходной атрибут в режимах IMITATION и MASK_IMITATION

Имя атрибута	Тип данных	Доступ	Описание атрибута
HARTVariableNumber1... HARTVariableNumber8	INT	чтение/ запись	Количество параметров HART канала 1...8 – количество параметров, передаваемых через выбранный HART канал
HARTID1... HARTID8	INT	чтение/ запись	Идентификатор HART-устройства, установленного на канале 1...8
<i>Диагностические атрибуты</i>			
Raw1...Raw8	INT	чтение	Код АЦП 1...Код АЦП 8
<i>Сигнализационные атрибуты</i>			
BusError1	BOOL	чтение	Ошибка шины ввода/вывода (основной) 1
BusError2	BOOL	чтение	Ошибка шины ввода/вывода (резервной) 2

Некоторые значения атрибута **Status** для МК576008AAIHART имеют дополнительное описание:

**Таблица 26. Дополнительное описание значений атрибута Status МК576008AAIHART**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружена корзина
BadConfiguration	0x02030000	Ошибка конфигурации модуля
CommunicationError	0x02070000	Отсутствует коммуникация с модулем ввода/вывода

Имя	Значение	Описание
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

### 5.10.2. Режимы МК576008ААИНАРТ

Режимы МК576008ААИНАРТ:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 5.10.3. Алгоритм МК576008ААИНАРТ

Выполнение алгоритма МК576008ААИНАРТ зависит от установленного режима.

Алгоритм МК576008ААИНАРТ включает в себя следующие элементы:

- диагностика;
- формирование сигнализаций;
- преобразование и запись;
- работа НАРТ модулей в составе PDM систем.

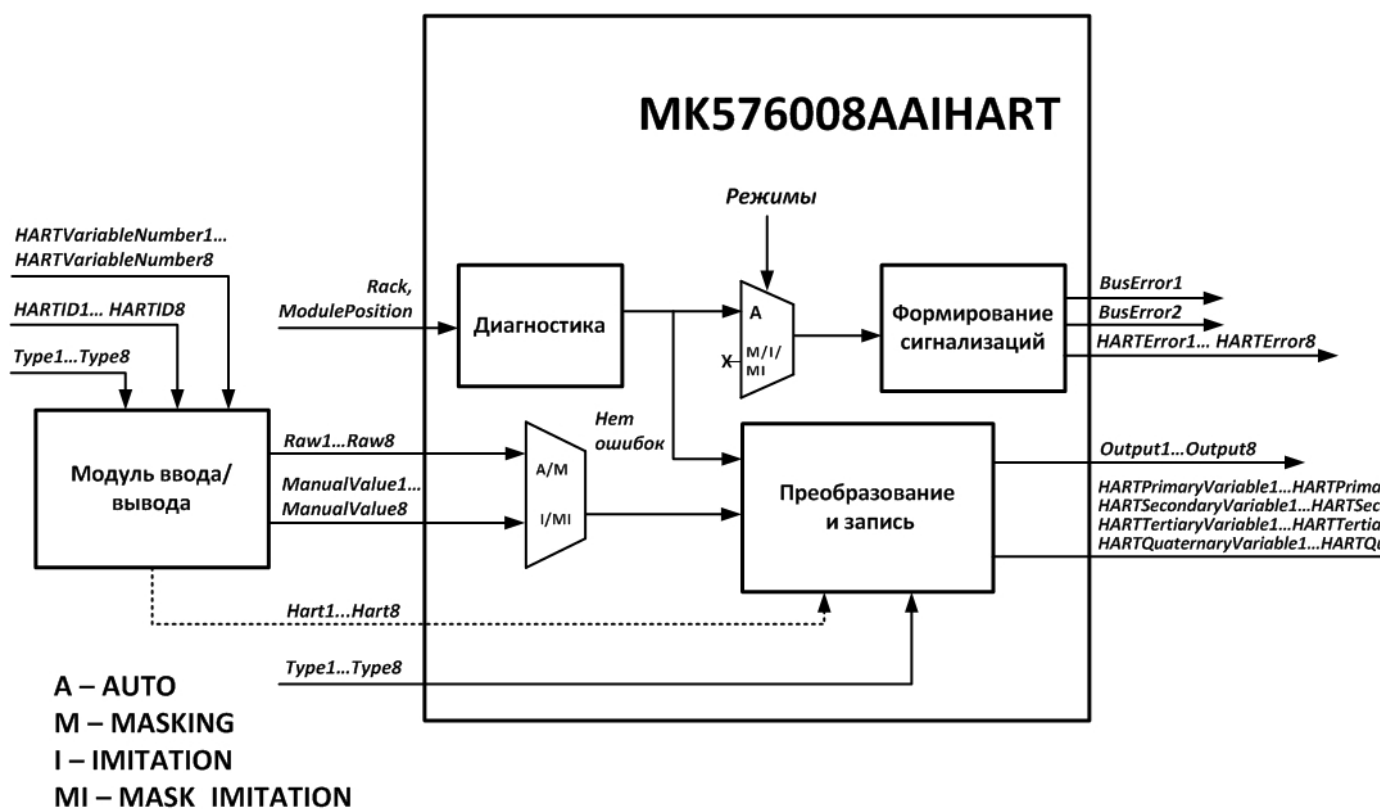


Рисунок 13. Схема процесса обработки данных MK576008AAIHART

### 5.10.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных MK576008AAIHART данным модуля ввода/вывода.

Элементы диагностики:

1. Проверка рабочего состояния модуля ввода/вывода и наличие связи с шиной ввода/вывода.
2. Самоидентификация.

Вычисляется номер корзины по атрибуту *Rack*.

Вычисляется уникальный номер MK576008AAIHART. При вычислении используется номер корзины и атрибут *ModulePosition*.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

4. Анализ диагностических сообщений из HART каналов и запись в соответствующие атрибуты *HARTError1...HARTError8*.

### 5.10.3.2. Формирование сигнализаций

MK576008AAIHART формирует сигнализации в следующих случаях:



- неисправность аппаратного модуля или отсутствие связи с шиной ввода/вывода (атрибут сигнализации ***BusError1*** либо ***BusError2*** принимает значение true);
- ошибка вычисления или соответствия идентификационных данных МК576008ААIНART (атрибуты сигнализации ***BusError1*** и ***BusError2*** принимают значение true).

### 5.10.3.3. Преобразование и запись

Производится при положительном результате диагностики.

В режимах AUTO и MASKING данный этап включает в себя следующие элементы:

- преобразование значений аналоговых сигналов из атрибутов ***Raw1...Raw8*** в процентном отношении от физической величины. При этом для каждого из указанных атрибутов учитывается тип аналогового входа ***Type1...Type8***. Тип аналогового входа может быть выбран 0...20 мА либо 4...20 мА.
- запись преобразованных данных в выходные атрибуты ***Output1...Output8***.
- запись преобразованных данных из HART каналов:
  - ***HARTPrimaryVariable1...HARTPrimaryVariable8***;
  - ***HARTSecondaryVariable1...HARTSecondaryVariable8***;
  - ***HARTTertiaryVariable1...HARTTertiaryVariable8***;
  - ***HARTQuaternaryVariable1...HARTQuaternaryVariable8***.

В режимах IMITATION и MASK\_IMITATION преобразование не происходит, данные в выходные атрибуты записываются из значений ***ManualValue1...ManualValue8*** в процентах.

В режимах IMITATION и MASK\_IMITATION переменные:

- ***HARTPrimaryVariable1...HARTPrimaryVariable8***;
- ***HARTSecondaryVariable1...HARTSecondaryVariable8***;
- ***HARTTertiaryVariable1...HARTTertiaryVariable8***;
- ***HARTQuaternaryVariable1...HARTQuaternaryVariable8***.

не опрашиваются, а принимаются их последние значения.

### 5.10.3.4. Работа HART модулей в составе PDM систем

Модули ввода-вывода МК576008ААIНART, МК576016ААIНART и МК574008ААOНART могут интегрироваться в PDM-системы, поддерживающие формат DTM (PactWare, FDT Container, PRM Yokogawa).

Модули МК574008ААOНART и МК576008ААIНART являются 8-ми канальными HART-модемами, модуль МК576016ААIНART – 16-ти канальным HART-модемом.

Для того чтобы подключить несколько модулей в одну шину, модули необходимо предварительно сконфигурировать.

Конфигурация заключается в задании базового смещения адресации (0...7) каналов HART.

Конфигурация осуществляется через Среду разработки Forge заданием параметра *ADDR\_OFFS* устройства от 0 до 7, по умолчанию смещение равно 0.

Базовое смещение модуля соответствует следующим адресам:

- 0 – адреса 0, 1, 2, 3, 4, 5, 6, 7
- 1 – адреса 8, 9, 10, 11, 12, 13, 14, 15
- 2 – адреса 16, 17, 18, 19, 20, 21, 22, 23
- 3 – адреса 24, 25, 26, 27, 28, 29, 30, 31
- 4 – адреса 32, 33, 34, 35, 36, 37, 38, 39
- 5 – адреса 40, 41, 42, 43, 44, 45, 46, 47
- 6 – адреса 48, 49, 50, 51, 52, 53, 54, 55
- 7 – адреса 56, 57, 58, 59, 60, 61, 62, 63

Для работы необходимо установить DTM с поддержкой многоканального HART-modem, например, бесплатный HART Communication фирмы CodeWrights. Также необходимо установить DTM подключенного датчика (на сайте производителя датчика). После чего необходимо обновить библиотеку PDM системы. Все подключенные датчики должны иметь базовый адрес 0.

## 5.11. Модуль аналогового ввода MK576016AAIHART

Модуль аналогового ввода MK576016AAIHART (далее MK576016AAIHART) обеспечивает выполнение следующих функций:

- преобразование аналоговых сигналов в значения физической величины для записи в выходные атрибуты;
- преобразование полученных данных из HART каналов в значения физической величины для записи в соответствующие выходные атрибуты;
- формирование сигнализаций.

### 5.11.1. Атрибуты MK576016AAIHART

Кроме базового набора атрибутов MK576016AAIHART имеет дополнительные атрибуты:

Таблица 27. Дополнительные атрибуты МК576016ААHART

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Выходные атрибуты</b>			
OUT1... OUT16	REAL	чтение	Выход 1...Выход 16 - значение соответствующего канала в процентах
HART_PV1... HART_PV16	REAL	чтение	Значение первой переменной HART канала 1...16
HART_SV1... HART_SV16	REAL	чтение	Значение второй переменной HART канала 1...16
HART_TV1... HART_TV16	REAL	чтение	Значение третьей переменной HART канала 1...16
HART_QV1... HART_QV16	REAL	чтение	Значение четвертой переменной HART канала 1...16
HART_ERR1... HART_ERR16	BOOL	чтение	Соответствующий HART канал в состоянии ошибки
<b>Конфигурационные атрибуты</b>			
RACK	STRING	чтение/ запись	Имя корзины, в которую установлен аппаратный модуль
SLOT	DINT	чтение/ запись	Позиция модуля в корзине
TYPE1...TYPE16	DINT	чтение/ запись	Тип 1...Тип 16 - тип аналогового входа (0...20 мА либо 4...20 мА)
MV1... MV16	REAL	чтение/ запись	Ручное значение 1...Ручное значение 16 - значение для записи в выходной атрибут в режимах IMITATION и MASK_IMITATION

Имя атрибута	Тип данных	Доступ	Описание атрибута
HART_VAR_NUM1... HART_VAR_NUM16	DINT	чтение/ запись	Количество параметров HART канала 1...16 – количество параметров, передаваемых через выбранный HART канал
HART_ID1... HART_ID16	DINT	чтение/ запись	Идентификатор HART-устройства, установленного на канале 1...16
RDNCY_EN	BOOL	чтение/ запись	Включение режима резервирования
RDNCY_MODE	USINT	чтение/ запись	Резервный режим (Тип резервирования)
ADDR_OFFS	USINT	чтение/ запись	Смещение каналов HART
NMATCH_RETR	UDINT	чтение/ запись	Максимальное число несовпадений подряд входной величины по одноименным каналам в режиме резервирования
NMATCH_DELTA	REAL	чтение/ запись	Максимальный разброс входной величины по одноименным каналам в режиме резервирования
<b>Диагностические атрибуты</b>			
RAW1...RAW16	DINT	чтение	Код АЦП 1...Код АЦП 16
OUTI1...OUTI16	REAL	чтение	Входное значение тока, мА
STS1...STS16	UINT	чтение	Статус входа 1...16
<b>Сигнализационные атрибуты</b>			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)

Некоторые значения атрибута *Status* для МК576016ААИНАРТ имеют дополнительное описание:

**Таблица 28. Дополнительное описание значений атрибута *Status* МК576016ААИНАРТ**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не загружена корзина
BadConfiguration	0x02030000	Ошибка конфигурации модуля
CommunicationError	0x02070000	Отсутствует коммуникация с модулем ввода/вывода
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

### 5.11.2. Режимы МК576016ААИНАРТ

Режимы МК576016ААИНАРТ:

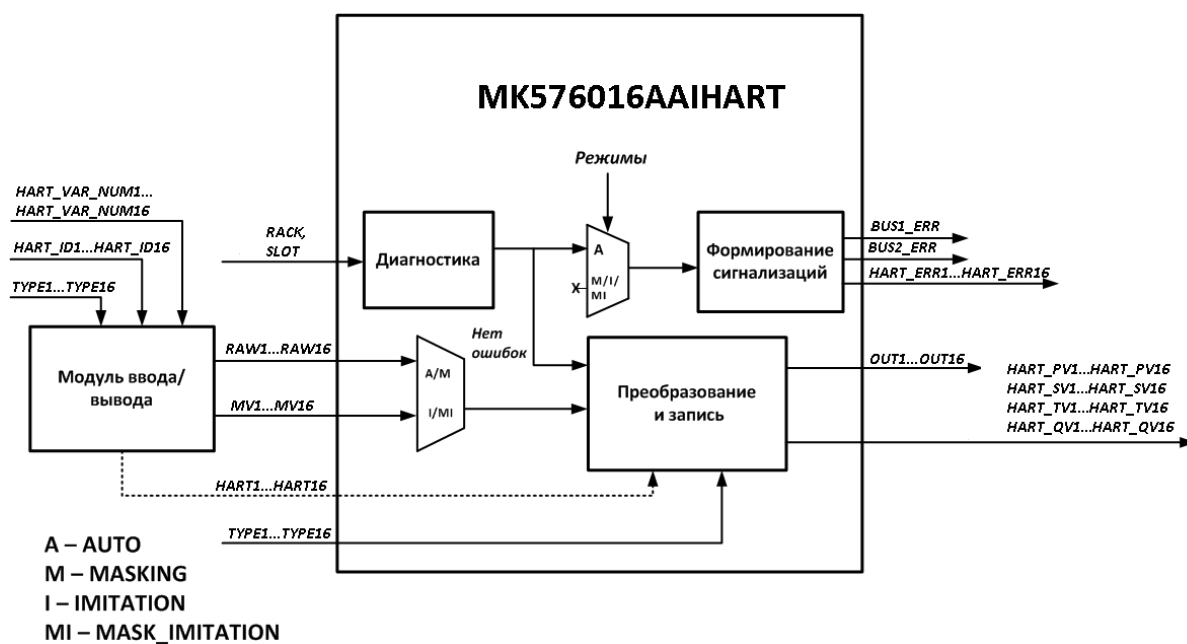
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 5.11.3. Алгоритм МК576016ААИНАРТ

Выполнение алгоритма МК576016ААИНАРТ зависит от установленного режима.

Алгоритм МК576016ААИНАРТ включает в себя следующие элементы:

- диагностика;
- формирование сигнализаций;
- преобразование и запись;
- работа НАРТ в режиме резервирования.



**Рисунок 14. Схема процесса обработки данных MK576016AAIHART**

MK576016AAIHART предназначен для ввода аналоговых сигналов от датчиков стандарта 0-20мА/4-20мА. Кроме аналоговой величины вход имеет следующие статусные состояния:

- уровень сигнала во входном канале превысил допустимый диапазон;
- модуль на данном канале не отвечает (для режима резервирования);
- входные величины аналогового сигнала основного и резервного модулей превысили допустимы диапазон разброса значений.

Статусные состояния позволяют определить качество входного сигнала и выдать аварию на верхний уровень.

В MK576016AAIHART предусмотрены выходы:

- **RAW(1...16)** - выходная величина в кодах АЦП модуля;
- **OUT(1...16)** - выходная величина в процентах от выходного диапазона канала (0-20мА / 4-20мА);
- **OUTI(1...16)** - выходная величина в миллиамперах (мА);
- **STS(1...16)** - статус входа.

MK576016AAIHART может формировать следующие выходные статусы каналов:

- ActiveFirst - канал первого модуля пары является активным (0b00000001);
- RawLimitOverFirst - уровень сигнала канала первого модуля пары вне диапазона значений (0b00000010);
- NotRespFirst - первый модуль данного канала не отвечает (0b00000100);
- ActiveSecond - канал второго модуля пары является активным (0b00001000);

- RawLimitOverSecond - уровень сигнала канала второго модуля пары вне диапазона значений (0b00010000);
- NotRespSecond - второй модуль данного канала не отвечает (0b00100000);
- RedundDataNMatch - разброс величины аналогового сигнала получаемого с 1 и 2 модулей превысил установленный предел *NMATCH\_DELTA* более чем *NMATCH\_RETR* раз подряд (0b01000000).

### 5.11.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных МК576016AAIHART данным модуля ввода/вывода.

Элементы диагностики:

1. Проверка рабочего состояния модуля ввода/вывода и наличие связи с шиной ввода/вывода.
2. Самоидентификация.

Вычисляется номер корзины по атрибуту *RACK*.

Вычисляется уникальный номер МК576016AAIHART. При вычислении используется номер корзины и атрибут *SLOT*.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

4. Анализ диагностических сообщений из HART каналов и запись в соответствующие атрибуты *HART\_ERR1...HART\_ERR16*.

### 5.11.3.2. Формирование сигнализаций

МК576016AAIHART формирует сигнализации в следующих случаях:

- неисправность аппаратного модуля или отсутствие связи с шиной ввода/вывода (атрибут сигнализации *BUS1\_ERR* либо *BUS2\_ERR* принимает значение true);
- ошибка вычисления или соответствия идентификационных данных МК576016AAIHART (атрибуты сигнализации *BUS1\_ERR* и *BUS2\_ERR* принимают значение true).

### 5.11.3.3. Преобразование и запись

Производится при положительном результате диагностики.

В режимах AUTO и MASKING данный этап включает в себя следующие элементы:

- преобразование значений аналоговых сигналов из атрибутов *RAW1...RAW16* в процентном отношении от физической величины. При этом для каждого из

указанных атрибутов учитывается тип аналогового входа **TYPE1...TYPE16**. Тип аналогового входа может быть выбран 0...20 мА либо 4...20 мА.

- запись преобразованных данных в выходные атрибуты **OUT1...OUT16**.
- запись преобразованных данных из HART каналов:
  - **HART\_PV1... HART\_PV16**;
  - **HART\_SV1... HART\_SV16**;
  - **HART\_TV1...HART\_TV16**;
  - **HART\_QV1...HART\_QV16**.

В режимах IMITATION и MASK\_IMITATION преобразование не происходит, данные в выходные атрибуты записываются из значений **MV1...MV16** в процентах.

В режимах IMITATION и MASK\_IMITATION переменные:

- **HART\_PV1... HART\_PV16**;
- **HART\_SV1... HART\_SV16**;
- **HART\_TV1...HART\_TV16**;
- **HART\_QV1...HART\_QV16**.

не опрашиваются, а принимаются их последние значения.

#### 5.11.3.4. Работа HART в режиме резервирования

Модуль MK576016AAIHART поддерживает работу в составе резервной пары. Данный режим включается установкой атрибута **RDNCY\_ENA** в 1. Кроме того, имеется возможность включить оба модуля. Выключить первый или второй модуль из работы, установив тип резервирования в атрибуте **RDNCY\_MODE** (OnlyFirstSlot - 1, OnlySecondSlot - 2, AllSlots - 3).

- В режиме резервирования тип AllSlots является адаптивным, поскольку позволяет автоматически выбирать какой из каналов пары модулей сформирует выходной сигнал функционального блока. Выбор осуществляется следующим образом:

Производится поканальный обход статусов обоих модулей:

1. Если статус соответствующего канала обоих в пределах установленного диапазона, входная величина каналов сравнивается, и если в пределах диапазона **NMATCH\_DELTA** - формируются выходные величины атрибутов **OUT(1...16)**, **RAW(1...16)**, **OUTI(1...16)** как среднее от показаний соответствующих каналов.
2. Если разность уровней сигналов от одноименных каналов модулей превышает установленный максимальный разброс, функциональный блок производит проверку на число несовпадений подряд (атрибут **NMATCH\_RETR**), и если оно превышено, выставляется дополнительный статус канала RedundDataNMatch.



3. Если статус канала одного из модулей плохой (превышен предел входной величины 22мА), берется выход другого, но авария не выставляется. При этом в выходном статусе канала функционального блока будет отображено, канал какого из модулей вышел в ошибку.
  4. Если статусы каналов обоих модулей плохие **OUT(1..16)**, функциональный блок сохранит предыдущее состояние, будет выставлен плохой статус канала.
- Если выставлен тип резервирования OnlyXXXSlot (XXX - First/Second), будут инициализированы оба модуля из резервной пары, но данные будут браться из первого или второго модуля, соответственно. Этот режим практически идентичен работе функционального блока без резервирования, но позволяет быстро переключиться при необходимости на соседний модуль из пары.

МК576016ААHART имеет возможность опроса HART-переменных с присоединенных датчиков. Для включения HART на канале необходимо указать количество опрашиваемых HART-переменных и HART-ID датчиков.

В режиме резервирования только один из модулей будет производить опросы HART. Переключение на опрос с резервного или основного модулей происходит автоматически при возникновении ошибок HART канала.

Для доступа к HART переменным или конфигурационным данным датчиков от стороннего оборудования на модуле предусмотрен соответствующий вход RS-485 конвертера HART-протокола. Если опрос от мастер устройства производится параллельно через несколько модулей поддерживающих HART, то для каждого модуля необходимо задать атрибут **ADDR\_OFFS** ( для подробного описания перейдите по ссылке [Работа HART модулей в составе PDM систем](#)).

При включении режима резервирования системой занимается дополнительная позиция в корзине, соответствующая номеру **SLOT+1** для установки резервного модуля. Таким образом, нумерация для следующих модулей в корзине продолжится с адреса **SLOT+2**.

## 5.12. Модуль аналогового вывода МК574008ААО с поддержкой HART

Модуль аналогового ввода МК574008ААО с поддержкой HART (далее МК574008ААОHART) обеспечивает выполнение следующих функций:

- преобразование значений входных атрибутов в соответствующие данные для записи в модуль ввода/вывода;
- преобразование полученных данных из HART каналов в значения физической величины для записи в соответствующие выходные атрибуты;
- формирование сигнализаций.

### 5.12.1. Атрибуты МК574008ААОНАRT

Кроме базового набора атрибутов МК574008ААОНАRT имеет дополнительные атрибуты:

**Таблица 29. Дополнительные атрибуты МК576008ААИНАRT**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b><i>Входные атрибуты</i></b>			
IN1... IN8	REAL	чтение	Вход 1...Вход 8 - значение соответствующего канала в процентах
<b><i>Выходные атрибуты</i></b>			
HART_PV1... HART_PV8	REAL	чтение	Значение первой переменной HART канала 1...8
HART_SV1... HART_SV8	REAL	чтение	Значение второй переменной HART канала 1...8
HART_TV1... HART_TV8	REAL	чтение	Значение третьей переменной HART канала 1...8
HART_QV1... HART_QV8	REAL	чтение	Значение четвертой переменной HART канала 1...8
HART_ERR1... HART_ERR8	BOOL	чтение	Соответствующий HART канал в состоянии ошибки
CH1_ERR...CH8_ERR	BOOL	чтение	Ошибка канала 1...8
<b><i>Конфигурационные атрибуты</i></b>			
RACK	STRING	чтение/ запись	Имя корзины, в которую установлен аппаратный модуль
SLOT	DINT	чтение/ запись	Позиция модуля в корзине
TYPE1...TYPE8	DINT	чтение/ запись	Тип 1...Тип 8 - тип аналогового входа (0...20 мА либо 4...20 мА)

Имя атрибута	Тип данных	Доступ	Описание атрибута
MV1... MV8	REAL	чтение/ запись	Ручное значение 1...Ручное значение 8 - значение соответствующего канала в процентах
HART_VAR_NUM1... HART_VAR_NUM8	DINT	чтение/ запись	Количество параметров HART канала 1...8 – количество параметров, передаваемых через выбранный HART канал
HART_ID1... HART_ID8	DINT	чтение/ запись	Идентификатор HART-устройства, установленного на канале 1...8
IN1_SRC...IN8_SRC	STRING	чтение/ запись	Источник 1...8
RDNCY_EN	BOOL	чтение/ запись	Включение режима резервирования
RDNCY_MODE	USINT	чтение/ запись	Резервный режим (Тип резервирования)
ADDR_OFFS	USINT	чтение/ запись	Смещение каналов HART
<b>Диагностические атрибуты</b>			
RAW1...RAW8	DINT	чтение	Код АЦП 1...Код АЦП 8
STS1...STS8	UINT	чтение	Статус входа 1...8
<b>Сигнализационные атрибуты</b>			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)

Некоторые значения атрибута **Status** для МК574008ААОHART имеют дополнительное описание:

**Таблица 30. Дополнительное описание значений атрибута *Status* МК574008ААОНART**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружена корзина
BadConfiguration	0x02030000	Ошибка конфигурации модуля
CommunicationError	0x02070000	Отсутствует коммуникация с модулем ввода/вывода
WaitForService	0x01020000	Ожидание коммуникации с драйвером ввода/вывода

### 5.12.2. Режимы МК574008ААОНART

Режимы МК574008ААОНART:

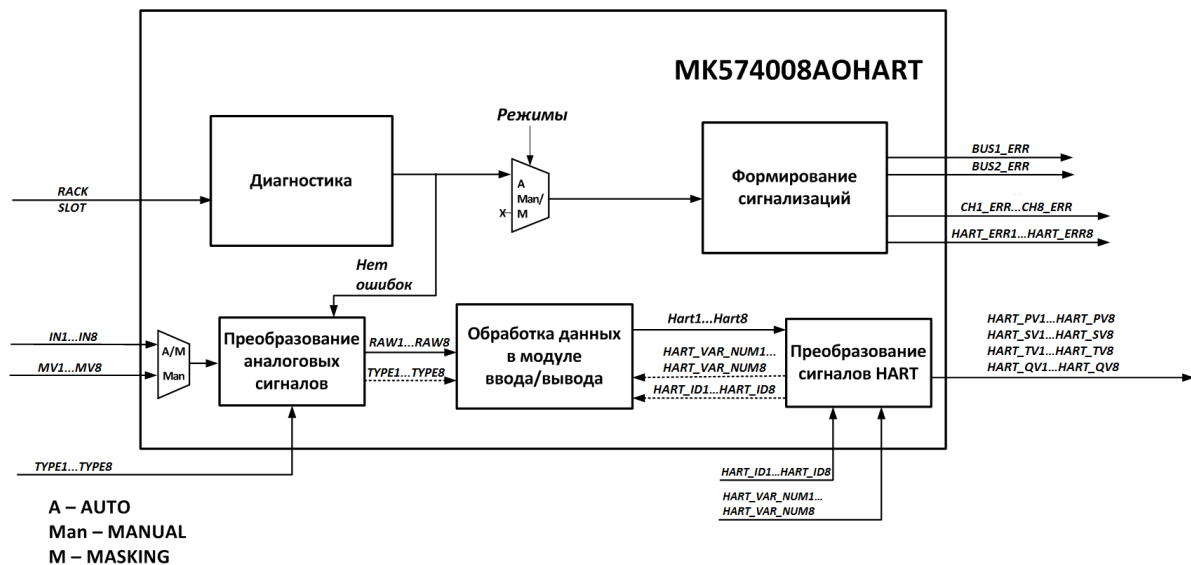
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- MANUAL ("Ручной").

### 5.12.3. Алгоритм МК574008ААОНART

Выполнение алгоритма МК574008ААОНART зависит от установленного режима.

Алгоритм МК574008ААОНART включает в себя следующие элементы:

- диагностика;
- формирование сигнализаций;
- преобразование аналоговых сигналов;
- преобразование сигналов HART;
- обработка данных в модуле ввода/вывода;
- работа HART в режиме резервирования.



**Рисунок 15. Схема процесса обработки данных МК574008AAOHART**

МК574008AAOHART предназначен для вывода аналоговых сигналов в приемное оборудование стандарта 0-20мА/4-20мА. Кроме выходной аналоговой величины вход имеет следующие статусные состояния:

- входная линия в обрыве;
- отсутствие питания в выходном каскаде модуля;
- модуль на данном канале не отвечает (для режима резервирования);
- модуль дисквалифицирован.

Статусные состояния позволяют определить качество входного сигнала и выдать аварию на верхний уровень.

Для установки уровня выходного сигнала в функциональный блок предусмотрены атрибуты:

- **IN(1...8)** - входная величина в процентах от диапазона канала модуля;
- **RAW(1...8)** - выходная величина в кодах АЦП модуля - предусмотрена для контроля на этапе настройки модуля и рассчитывается автоматически;
- **STS(1...8)** - статус выхода.

МК574008AAOHART может формировать следующие выходные статусы каналов:

- **ActiveFirst** - канал первого модуля пары является активным (0b00000001);
- **NotRespFirst** - первый модуль канала не отвечает (0b00000010);
- **OpenCircuitFirst** - данный канал первого модуля в состоянии обрыв цепи (0b00000100);
- **PowerDownFirst** - данный канал первого модуля в состоянии отсутствия питания (0b00001000);

- RedundModFaultFirst - первый модуль на канале дисквалифицирован (0b00010000);
- ActiveSecond - канал второго модуля пары является активным (0b00100000);
- NotRespSecond - второй модуль канала не отвечает (0b01000000);
- OpenCircuitSecond - данный канал второго модуля в состоянии обрыв цепи (0b10000000);
- PowerDownSecond - данный канал второго модуля в состоянии отсутствия питания (0b000100000000);
- RedundModFaultSecond - второй модуль на канале дисквалифицирован (0b001000000000).

### 5.12.3.1. Диагностика

Диагностика определяет соответствие идентификационных данных МК574008ААОНART данным модуля ввода/вывода.

Элементы диагностики:

1. Проверка рабочего состояния модуля ввода/вывода и наличие связи с шиной ввода/вывода.
2. Самоидентификация.

Вычисляется номер корзины по атрибуту **RACK**.

Вычисляется уникальный номер МК574008ААОНART. При вычислении используется номер корзины и атрибут **SLOT**.

3. Определение соответствия.

Происходит сравнение идентификационных данных с данными модуля ввода/вывода.

4. Анализ диагностических сообщений из HART каналов и запись в соответствующие атрибуты **HART\_ERR1...HART\_ERR8**.

### 5.12.3.2. Формирование сигнализаций

МК574008ААОНART формирует сигнализации в следующих случаях:

- неисправность аппаратного модуля или отсутствие связи с шиной ввода/вывода (атрибут сигнализации **BUS1\_ERR** либо **BUS2\_ERR** принимает значение true);
- ошибка вычисления или соответствия идентификационных данных МК574008ААОНART (атрибуты сигнализации **BUS1\_ERR** и **BUS2\_ERR** принимают значение true).

### 5.12.3.3. Преобразование

Преобразование производится при положительном результате диагностики.

В режиме AUTO и MASKING данный этап включает в себя следующие элементы:

- преобразование значений аналоговых сигналов из атрибутов *IN1... IN8* в соответствующие коды АЦП аппаратного модуля (модуля ввода/вывода). При этом для каждого из указанных атрибутов учитывается тип аналогового входа *TYPE1... TYPE8*. Тип аналогового входа может быть выбран 0...20 мА либо 4...20 мА.
- запись преобразованных данных из HART каналов:
  - *HART\_PV1... HART\_PV8*;
  - *HART\_SV1... HART\_SV8*;
  - *HART\_TV1...HART\_TV8*;
  - *HART\_QV1...HART\_QV8*.

В режиме MASKING этап диагностики коммуникации отключен.

В режиме MANUAL запись данных в атрибуты *RAW1... RAW8* осуществляется из атрибутов *MV1... MV8*.

В режимах MANUAL переменные:

- *HART\_PV1... HART\_PV8*;
- *HART\_SV1... HART\_SV8*;
- *HART\_TV1...HART\_TV8*;
- *HART\_QV1...HART\_QV8*.

не опрашиваются, а принимаются их последние значения.

В режиме OFF логика блока не обрабатывается, и принимаются последние значения переменных.

#### 5.12.3.4. Обработка данных в модуле ввода/вывода

В модуле ввода/вывода производится прием кодовых посылок HART, от подсоединенных устройств. HART-посылки расшифровываются и передаются в блок "Преобразование сигналов HART".

В модуле ввода/вывода формируются выходные аналоговые сигналы для каналов АЦП *AO1... AO8*.

#### 5.12.3.5. Работа HART в режиме резервирования

Модуль MK574008AAOHART поддерживает работу в составе резервной пары. Данный режим включается установкой атрибута *RDNCY\_ENA* в 1. Также для корректной работы модулей должна быть установлена межмодульная перемычка (подробнее см. документацию на модуль).

В режиме резервирования поддерживается только автоматический режим работы резервной пары AllSlots. Данный режим реализуется полностью на аппаратном

уровне. Попытка изменить конфигурацию, например, выставить ***RDNCY\_ENA*** в 0 при установленной перемычке, приведет к сбоям в работе модулей.

В резервируемом режиме каждый из пары модулей устанавливает половину выходной величины тока, что в сумме дает полный заданный уровень. В случае сбоев в работе одного из модулей или его дисквалификации, переключение выходного сигнала происходит автоматически с выставлением соответствующего статуса канала.

Модуль MK574008AAOHART имеет возможность опроса HART-переменных с присоединенных датчиков. Для включения HART на канале необходимо указать количество опрашиваемых HART-переменных и HART-ID датчиков.

В режиме резервирования только один из модулей будет производить опросы HART. Переключение на опрос с резервного или основного модулей происходит автоматически при возникновении ошибок HART канала.

Для доступа к HART переменным или конфигурационным данным датчиков от стороннего оборудования на модуле предусмотрен соответствующий вход RS-485 конвертера HART-протокола. Если опрос от мастер устройства производится параллельно через несколько модулей, поддерживающих HART, то для каждого модуля необходимо задать атрибут ***ADDR\_OFFS*** (для подробного описания перейдите по ссылке [Работа HART модулей в составе PDM систем](#)).

При включении режима резервирования системой занимается дополнительная позиция в корзине, соответствующая номеру ***SLOT***+1 для установки резервного модуля. Таким образом, нумерация для следующих модулей в корзине продолжится с адреса ***SLOT***+2.

## 5.13. Модуль коммуникации MK541002 с поддержкой Modbus

Модуль коммуникации MK541002 с поддержкой Modbus (далее MK541002) обеспечивает приём и передачу данных по интерфейсу RS-485 протокола ModbusRTU и выполняет следующие функции:

- конфигурация;
- коммуникация;
- диагностика;
- формирование сигнализаций.

Модуль коммуникации MK541002 используется только в качестве ведущего устройства (master) протокола ModbusRTU.

### 5.13.1. Атрибуты MK541002

MK541002 имеет базовый набор атрибутов, а также дополнительные атрибуты:



Таблица 31. Дополнительные атрибуты МК541002

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Конфигурационные атрибуты</b>			
RACK	STRING	чтение/ запись	Корзина
SLOT	INT	чтение/ запись	Позиция модуля
P1_SPD	DINT	чтение/ запись	Скорость порта 1
P1_SP_BITS	DINT	чтение/ запись	Число стоп-битов порта 1
P1_PRTY	DINT	чтение/ запись	Четность порта 1
P1_TO	DINT	чтение/ запись	Таймаут ожидания овета на запрос (мс) порта 1
P1_DTA_BITS	DINT	чтение/ запись	Число бит в символе порта 1
P2_SPD	DINT	чтение/ запись	Скорость порта 2
P2_SP_BITS	DINT	чтение/ запись	Число стоп-битов порта 1
P2_PRTY	DINT	чтение/ запись	Четность порта 2
P2_TO	DINT	чтение/ запись	Таймаут ожидания овета на запрос (мс) порта 2
P2_DTA_BITS	DINT	чтение/ запись	Число бит в символе порта 2
SupportedParity	DINT	чтение/ запись	Контроль чётности: None - 78, Odd - 79, Even - 69
WRK_PAUSE_SE	BOOL	чтение	Установить рабочую паузу
WRK_PAUSE_RS	BOOL	чтение	Сбросить рабочую паузу
<b>Сигнализационные атрибуты</b>			
BUS1_ERR	BOOL	чтение	Ошибка шины ввода/вывода 1 (основной)

Имя атрибута	Тип данных	Доступ	Описание атрибута
BUS2_ERR	BOOL	чтение	Ошибка шины ввода/вывода 2 (резервной)
<i>Диагностические атрибуты</i>			
WRK_PAUSE_ST	BOOL	чтение	Статус рабочей паузы

### 5.13.2. Режимы МК541002

Режимы МК541002:

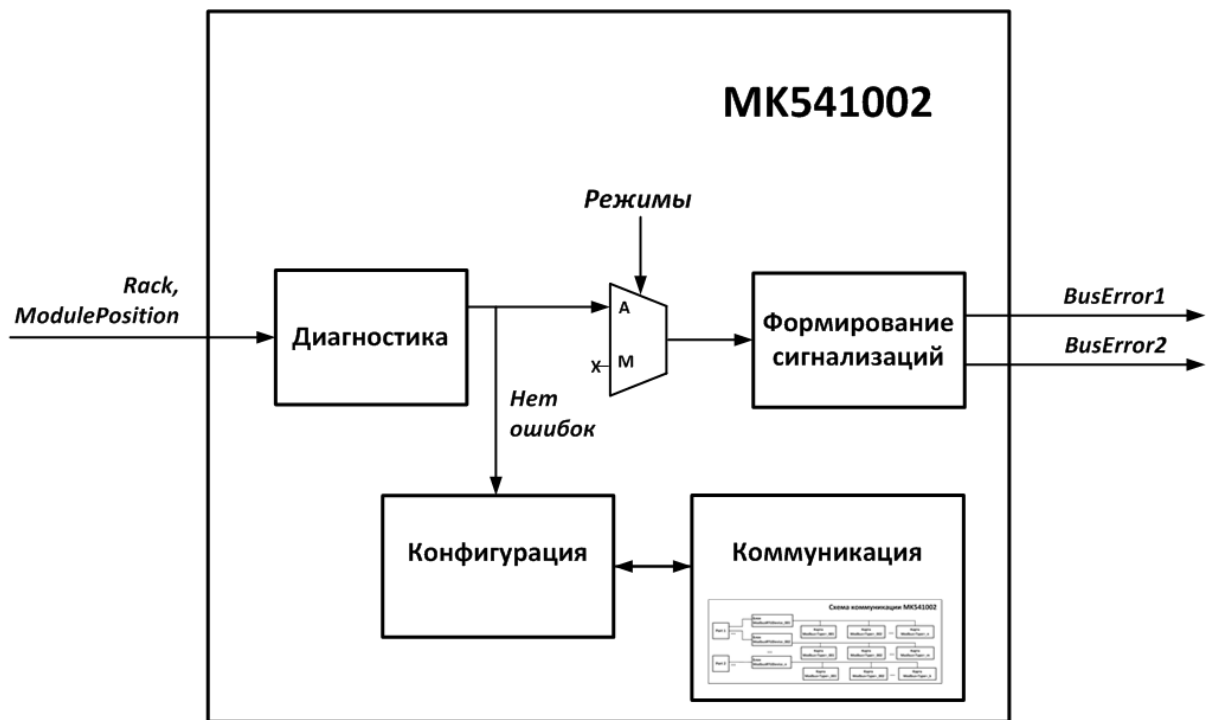
- OFF ("Выключен") - алгоритм не выполняется;
- AUTO ("Автоматический") - алгоритм выполняется в полном объеме;
- MASKING ("Маскирование") - алгоритм выполняется за исключением этапа формирования сигнализаций.

### 5.13.3. Алгоритм МК541002

Выполнение алгоритма МК541002 зависит от установленного режима.

Алгоритм МК541002 включает в себя следующие элементы:

- конфигурация;
- коммуникация;
- диагностика;
- формирование сигнализаций.



A – AUTO  
M – MASKING

Рисунок 16. Схема процесса обработки данных MK541002

### 5.13.3.1. Конфигурация

На данном этапе производится настройка коммуникации модуля MK541002 с устройствами Modbus.

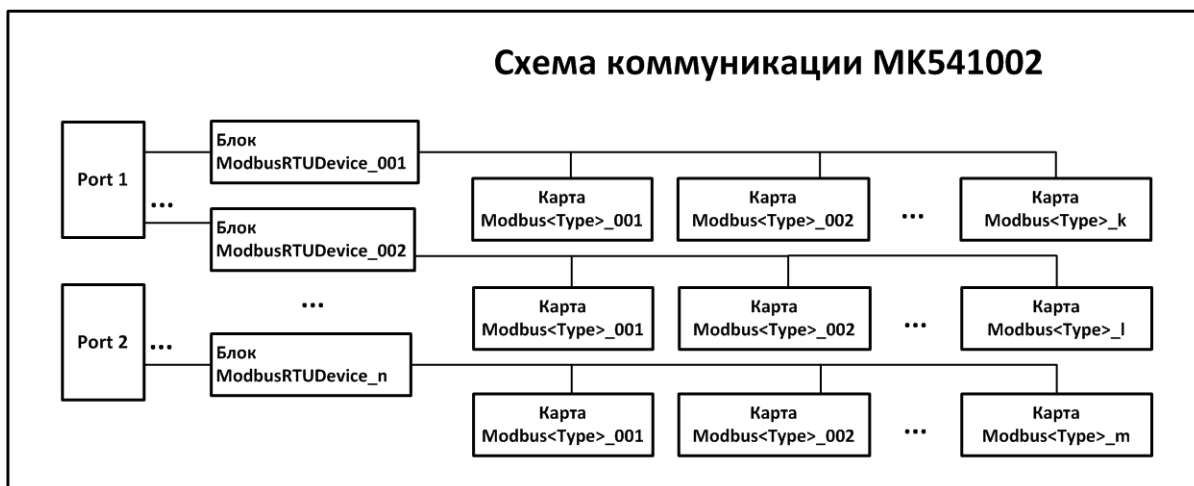
Параметры настройки записываются в конфигурационные атрибуты MK541002.

### 5.13.3.2. Коммуникация

На данном этапе модулем MK541002 производится последовательный опрос подключаемых к его картам устройств по протоколу ModbusRTU в соответствии с заданными настройками. Формирование типа Modbus-запросов осуществляется в процессе конфигурации модуля.

Собранные данные передаются в MK541002 и далее распределяются по блокам карт Modbus. Данная работа осуществляется в соответствии с алгоритмом работы Modbus карты.<sup>5</sup>

<sup>5</sup> Подробное описание алгоритмов работы карт Modbus описано в разделе Функциональные блоки коммуникации по ModBus.



**Рисунок 17. Структурная схема связи МК541002**

### 5.13.3.3. Диагностика

Диагностика коммуникации производится в режиме AUTO. В режиме MASKING диагностика отключена.

В процессе диагностики осуществляется проверка коммуникации МК541002 с Modbus устройством. При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

Также в процессе диагностики осуществляется проверка исправности модуля ввода/вывода.

В случае если Modbus устройство отсутствует или находится в состоянии ошибки, формируется сигнализация карты/карт соответствующего устройства .

В случае если все внешние устройства Modbus, подключенные к порту МК541002, находятся в состоянии ошибки, то весь модуль переводится в состояние *Communication Error* (атрибут *Status* принимает значение 0x02070000).

### 5.13.3.4. Формирование сигнализаций

МК541002 формирует сигнализации в следующих случаях:

- неисправность модуля ввода/вывода или отсутствие связи с шиной ввода/вывода (атрибут сигнализации *BusError1* либо *BusError2* принимает значение true);
- ошибка вычисления или соответствия идентификационных данных МК541002 (атрибуты сигнализации *BusError1* и *BusError2* принимают значение true).

## 6. Функциональные блоки коммуникации по ModBus

Функциональные блоки коммуникации по ModBus являются частью системного программного обеспечения РСУ.

К данным функциональным блокам относятся:

- Устройство ModBusTCPDevice;
- Устройство ModBusRTUDevice;
- Карта ModBusRealInput32;
- Карта ModBusRealOutput32;
- Карта ModBusIntegerInput32;
- Карта ModBusIntegerOutput32;
- Карта ModBusUIntegerInput32;
- Карта ModBusUIntegerOutput32;
- Карта ModBusBooleanInput32;
- Карта ModBusBooleanOutput32;
- Карта ModBusShortInput32;
- Карта ModBusShortOutput32;
- Карта ModBusUshortInput32;
- Карта ModBusUshortOutput32.

При настройке чтения/записи применяются следующие параметры:

***StartRegister*** - начальный регистр. В качестве начального регистра выбирается первый адрес из нужного диапазона регистров устройства Modbus. Например, для диапазона регистров устройства Modbus с адресами от 180 до 500, начальный адрес в карте ModBusRealInput32 рекомендуется задать 180.

***RegistersType*** - тип регистров. Показывает, к данным какого типа регистров необходимо обращаться для чтения из Modbus устройства. Адреса в картах с разным типом регистров для одного устройства Modbus могут совпадать. Для выбора доступны следующие типы регистра:

- HoldingRegisters (регистры хранения, значение 3);
- InputRegisters (регистры ввода, значение 4).

***StartBit*** - начальный бит. В качестве начального бита выбирается первый адрес из нужного диапазона битов устройства Modbus. Например, для диапазона битов устройства Modbus с адресами от 180 до 500, начальный адрес в карте рекомендуется задать 180.

**BitType** - тип битов. Показывает, к данным какого типа битов необходимо обращаться для чтения из устройства Modbus. Адреса в картах с разным типом битов для одного устройства Modbus могут совпадать. В карте ModBusBooleanInput32 для выбора доступны следующие типы битов:

- DiscreteInputs (доступный для чтения, значение 0);
- Coils (доступный для чтения и записи, значение 1).

**WordEndian** и **ByteEndian** - порядок слов и порядок байтов. Порядок слов регистров в карте задается при помощи указания вида последовательности:

- WordLittleEndian (от младшего к старшему, значение 0);
- WordBigEndian (от старшего к младшему, значение 1).

Аналогично задается порядок байтов в регистрах в карте при помощи указания вида последовательности:

- ByteLittleEndian (от младшего к старшему, значение 0);
- ByteBigEndian (от старшего к младшему, значение 1).

**Таблица 32. Определение порядка слов и байтов в зависимости от их расположения**

Расположение байтов	Порядок слов ( <i>WordEndian</i> )/значение	Порядок байтов ( <i>ByteEndian</i> )/значение
1 2   3 4	WordLittleEndian/0	ByteLittleEndian/0
2 1   4 3	WordLittleEndian/0	ByteBigEndian/1
3 4   1 2	WordBigEndian/1	ByteLittleEndian/0
4 3   2 1	WordBigEndian/1	ByteBigEndian/1

**Offset1...Offset32** - смещения для каждого из 32 выходов устройства Modbus, считываемых одной картой ModBusBooleanInput32. Первое смещение задается относительно начального бита. Последующие смещения задаются относительно предыдущего. Максимальная величина смещения - 2000 единиц. Если смещению какого-либо выхода задано значение -1, то данный выход исключается из опроса.

**ScanRate** - период опроса устройства Modbus, с которым производится считывание данных. Величина измерения - миллисекунды.

## 6.1. Устройство ModBusTCPDevice

Устройство ModBusTCPDevice (далее ModBusTCPDevice) выполняет настройку и диагностику коммуникации с устройством ModBusTCP через сетевой интерфейс контроллера.

### 6.1.1. Атрибуты ModBusTCPDevice

ModBusTCPDevice имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 33. Дополнительные атрибуты ModBusTCPDevice**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Конфигурационные атрибуты</i>			
DeviceId	INT	чтение/ запись	ID устройства
DeviceIPAddress	INT	чтение/ запись	IP адрес устройства
DeviceIPPort	INT	чтение/ запись	IP адрес порта устройства
ResponseTimeout	INT	чтение/ запись	Максимальное время ожидания ответа, мс

Перечень типов данных атрибутов функционального блока описан в разделе 4. *Атрибуты функциональных блоков* Таблица 2. *Типы данных атрибутов функционального блока* документа "Концепция технологического программного обеспечения".

### 6.1.2. Режимы ModBusTCPDevice

Режимы ModBusTCPDevice:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

### 6.1.3. Алгоритм ModBusTCPDevice

Выполнение алгоритма ModBusTCPDevice зависит от установленного режима.

Алгоритм ModBusTCPDevice включает в себя элементы:

- коммуникация контроллера с устройством ModBusTCP;

- диагностика коммуникации.

### 6.1.3.1. Коммуникация с устройством ModBusTCP

На данном этапе производится настройка коммуникации контроллера с устройством ModBusTCP через сетевой интерфейс контроллера.

Параметры настройки записываются в конфигурационные атрибуты ModBusTCPDevice.

### 6.1.3.2. Диагностика коммуникации

Диагностика коммуникации производится в режиме AUTO. В режиме MASKING диагностика отключена.

ModBusTCPDevice диагностирует коммуникацию контроллера с устройством ModBusTCP.

Диагностический атрибут *Status* предоставляет информацию о состоянии ModBusTCPDevice.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

**Примечание:** если коммуникация отсутствует, атрибут *Status* принимает значение 0x02070000 (*CommunicationError*).

## 6.2. Устройство ModBusRTUDevice

Устройство ModBusRTUDevice (далее ModBusRTUDevice) выполняет настройку и диагностику соединения контроллера с устройством ModbusRTU по последовательному порту.

### 6.2.1. Атрибуты ModBusRTUDevice

ModBusRTUDevice имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 34. Дополнительные атрибуты ModBusRTUDevice**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Конфигурационные атрибуты</i>			
DeviceId	INT	чтение/ запись	ID устройства
SerialPortNumber	INT	чтение/ запись	Номер последовательного порта (1 либо 2)



Имя атрибута	Тип данных	Доступ	Описание атрибута
Speed	INT	чтение/ запись	Скорость, (9600 бит/с)
StopBits	INT	чтение/ запись	Стоповый бит (1 либо 2)
Parity	INT	чтение/ запись	Четность
Timeout	INT	чтение/ запись	Допустимая задержка на ответ, мс

Перечень типов данных атрибутов функционального блока описан в разделе 4. Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока документа "Концепция технологического программного обеспечения".

### 6.2.2. Режимы ModBusRTUDevice

Режимы ModBusRTUDevice:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");

### 6.2.3. Алгоритм ModBusRTUDevice

Выполнение алгоритма ModBusRTUDevice зависит от установленного режима.

Алгоритм ModBusRTUDevice включает в себя элементы:

- коммуникация контроллера с устройством ModbusRTU по последовательному порту;
- диагностика коммуникации.

#### 6.2.3.1. Коммуникация с устройством Modbus

На данном этапе производится настройка соединения контроллера с устройством ModbusRTU по последовательному порту.

Параметры настройки записываются в конфигурационные атрибуты ModbusRTUDevice.

При этом в атрибут *Parity* может быть записано одно из следующих значений:

- None (значение 78) - нет контроля четности;
- Odd (значение 79) - нечетный бит;

- Even (значение 69) - четный бит;
- Mark (значение 77) - бит четности всегда установлен в единицу.

### 6.2.3.2. Диагностика коммуникации

Диагностика коммуникации производится в режиме AUTO. В режиме MASKING диагностика отключена.

ModbusRTUDevice диагностирует коммуникацию контроллера с устройством Modbus.

Диагностический атрибут *Status* предоставляет информацию о состоянии ModbusRTUDevice.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

**Примечание:** если коммуникация отсутствует, атрибут *Status* принимает значение 0x02070000 (*CommunicationError*).

## 6.3. Карта ModBusRealInput32

Карта ModBusRealInput32 (далее ModBusRealInput32) обеспечивает выполнение следующих функций:

- организация чтения данных типа REAL из устройства Modbus;
- предоставление доступа к прочитанным данным в технологической программе;
- диагностика коммуникации с другими функциональными блоками.

Один ModBusRealInput32 может считать данные из 32 выходов устройства Modbus, содержащих переменные типа REAL.

### 6.3.1. Атрибуты ModBusRealInput32

ModBusRealInput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 35. Дополнительные атрибуты ModBusRealInput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Выходные атрибуты</i>			
Output1...Output32	REAL	чтение	Выход 1...Выход 32
<i>Конфигурационные атрибуты</i>			

Имя атрибута	Тип данных	Доступ	Описание атрибута
StartRegister	INT	чтение/ запись	Начальный регистр
RegistersType	INT	чтение/ запись	Тип регистра (Input или Holding)
WordEndian	INT	чтение/ запись	Порядок слов ( <b>WordEndian</b> )
ByteEndian	INT	чтение/ запись	Порядок байтов ( <b>ByteEndian</b> )
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Имя устройства Modbus, к которому привязана карта

Перечень типов данных атрибутов функционального блока описан в разделе 4. *Атрибуты функциональных блоков* Таблица 2. *Типы данных атрибутов функционального блока* документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута **Status** для ModBusRealInput32 имеют дополнительное описание:

**Таблица 36. Дополнительное описание значений атрибута *Status* ModBusRealInput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.3.2. Режимы ModBusRealInput32

Режимы ModBusRealInput32:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

### 6.3.3. Алгоритм ModBusRealInput32

Выполнение алгоритма ModBusRealInput32 зависит от установленного режима.

Алгоритм ModBusRealInput32 включает в себя следующие элементы:

- чтение данных Modbus устройства;
- диагностика коммуникации.

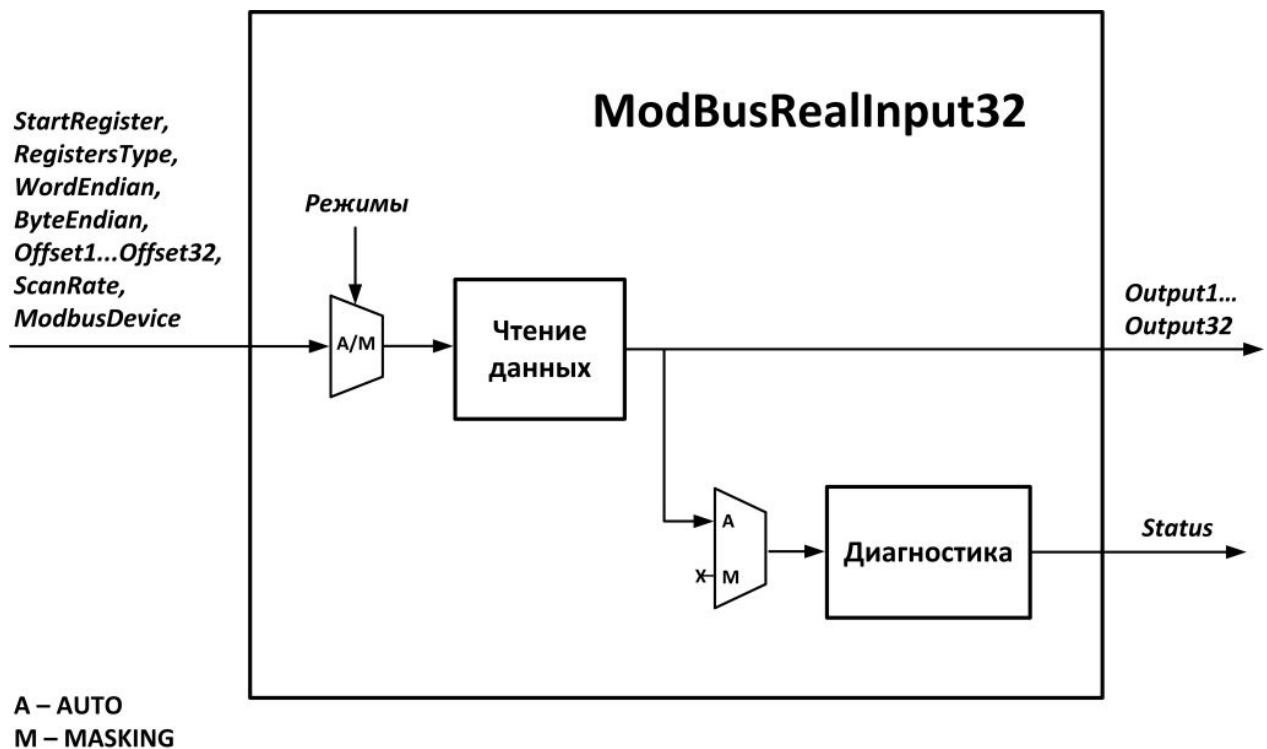


Рисунок 18. Схема процесса обработки данных ModBusRealInput32

#### 6.3.3.1. Чтение данных устройства Modbus

На данном этапе производится чтение данных из регистров устройства Modbus и запись в выходные атрибуты *Output1 ... Output32* ModBusRealInput32.

Для чтения данных должны быть заданы конфигурационные атрибуты:

*StartRegister*;

*RegistersType*;

*WordEndian* и *ByteEndian*;

*Offset1...Offset32;*

*ScanRate;*

*ModbusDevice.*

### 6.3.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты с Modbus устройством.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad.*

## 6.4. Карта ModBusRealOutput32

Карта ModBusRealOutput32 (далее ModBusRealOutput32) обеспечивает выполнение следующих функций:

- организация записи данных типа REAL в устройство ModBus;
- диагностика коммуникации с другими функциональными блоками.

В один ModBusRealOutput32 могут быть записаны данные типа REAL для 32 входов устройства Modbus.

### 6.4.1. Атрибуты ModBusRealOutput32

ModBusRealOutput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 37. Дополнительные атрибуты ModBusRealOutput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
Input1...Input32	REAL	чтение	Вход 1...Вход 32
<i>Конфигурационные атрибуты</i>			
StartRegister	INT	чтение/ запись	Начальный регистр
WordEndian	INT	чтение/ запись	Порядок слов ( <b>WordEndian</b> )
ByteEndian	INT	чтение/ запись	Порядок байтов ( <b>ByteEndian</b> )
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32

Имя атрибута	Тип данных	Доступ	Описание атрибута
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)
Source1...Source32	STRING	чтение/ запись	Имя источника входных данных

Перечень типов данных атрибутов функционального блока описан в разделе 4. Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута *Status* для ModBusRealOutput32 имеют дополнительное описание:

**Таблица 38. Дополнительное описание значений атрибута *Status* ModBusRealOutput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

#### 6.4.2. Режимы ModBusRealOutput32

Режимы ModBusRealOutput32:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");

В режиме MASKING этап диагностики коммуникации отключен.

### 6.4.3. Алгоритм ModBusRealOutput32

Выполнение алгоритма ModBusRealOutput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- запись данных в Modbus устройство;
- диагностика коммуникации.

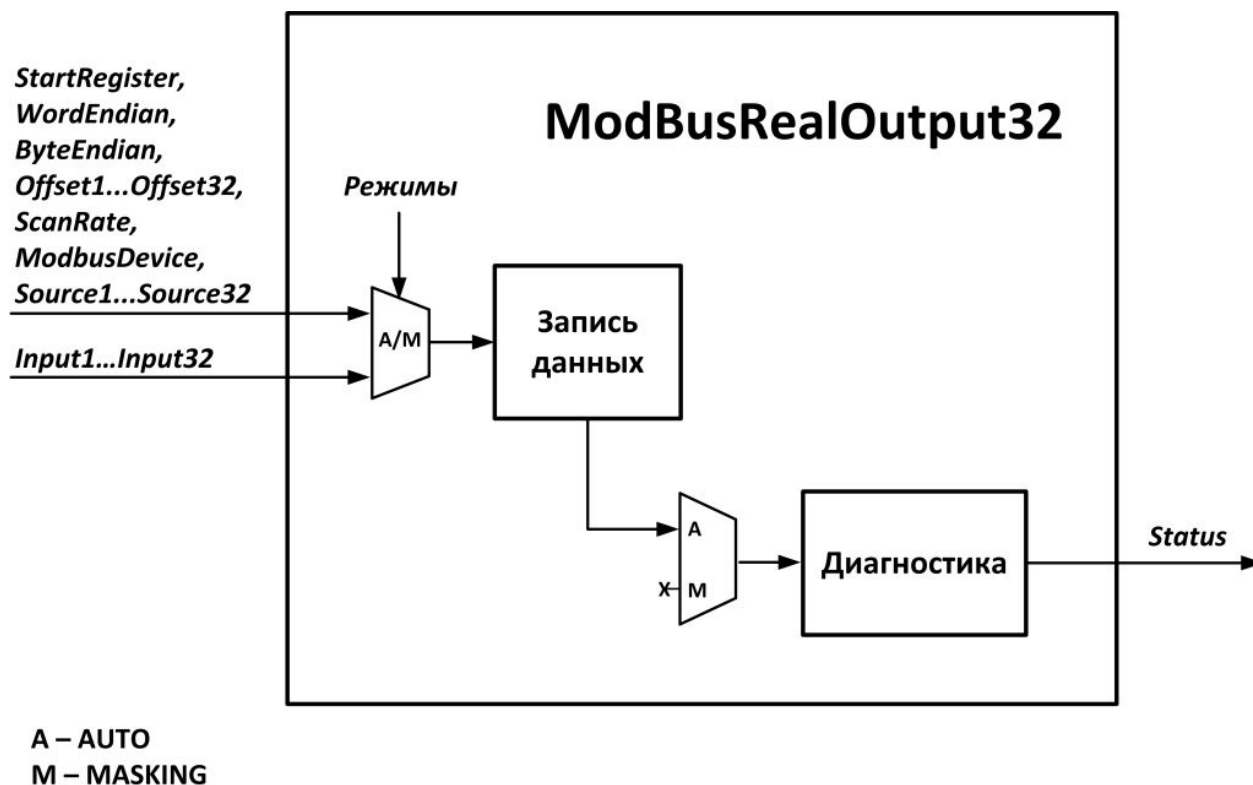


Рисунок 19. Схема процесса обработки данных ModBusRealInput32

#### 6.4.3.1. Запись данных в устройство Modbus

На данном этапе производится запись данных из входных атрибутов *Input1...Input32* ModBusRealOutput32.

Для записи данных должны быть заданы конфигурационные атрибуты:

*StartRegister*;

*WordEndian* и *ByteEndian*;

*Offset1...Offset32*;

*ScanRate*;

*ModbusDevice*;

*Source1...Source32*.

### 6.4.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusRealOutput32 с Modbus устройством.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

## 6.5. Карта ModBusIntegerInput32

Карта ModBusIntegerInput32 (далее ModBusIntegerInput32) обеспечивает выполнение следующих функций:

- организация чтения данных типа INT из устройства Modbus;
- предоставление доступа к прочитанным данным в технологической программе;
- диагностика коммуникации с другими функциональными блоками.

Один ModBusIntegerInput32 может считать данные из 32 выходов устройства Modbus, содержащих переменные типа INT.

### 6.5.1. Атрибуты ModBusIntegerInput32

ModBusIntegerInput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 39. Дополнительные атрибуты ModBusIntegerInput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Выходные атрибуты</i>			
Output1...Output32	INT	чтение	Выход 1...Выход 32
<i>Конфигурационные атрибуты</i>			
StartRegister	INT	чтение/ запись	Начальный регистр
RegistersType	INT	чтение/ запись	Тип регистра (Input или Holding)
WordEndian	INT	чтение/ запись	Порядок слов ( <b>WordEndian</b> )
ByteEndian	INT	чтение/ запись	Порядок байтов ( <b>ByteEndian</b> )
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32



Имя атрибута	Тип данных	Доступ	Описание атрибута
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Имя устройства Modbus, к которому привязана карта

Перечень типов данных атрибутов функционального блока описан в разделе 4. *Атрибуты функциональных блоков* Таблица 2. *Типы данных атрибутов функционального блока* документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута *Status* для ModBusIntegerInput32 имеют дополнительное описание:

**Таблица 40. Дополнительное описание значений атрибута *Status* ModBusIntegerInput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.5.2. Режимы ModBusIntegerInput32

Режимы ModBusIntegerInput32:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

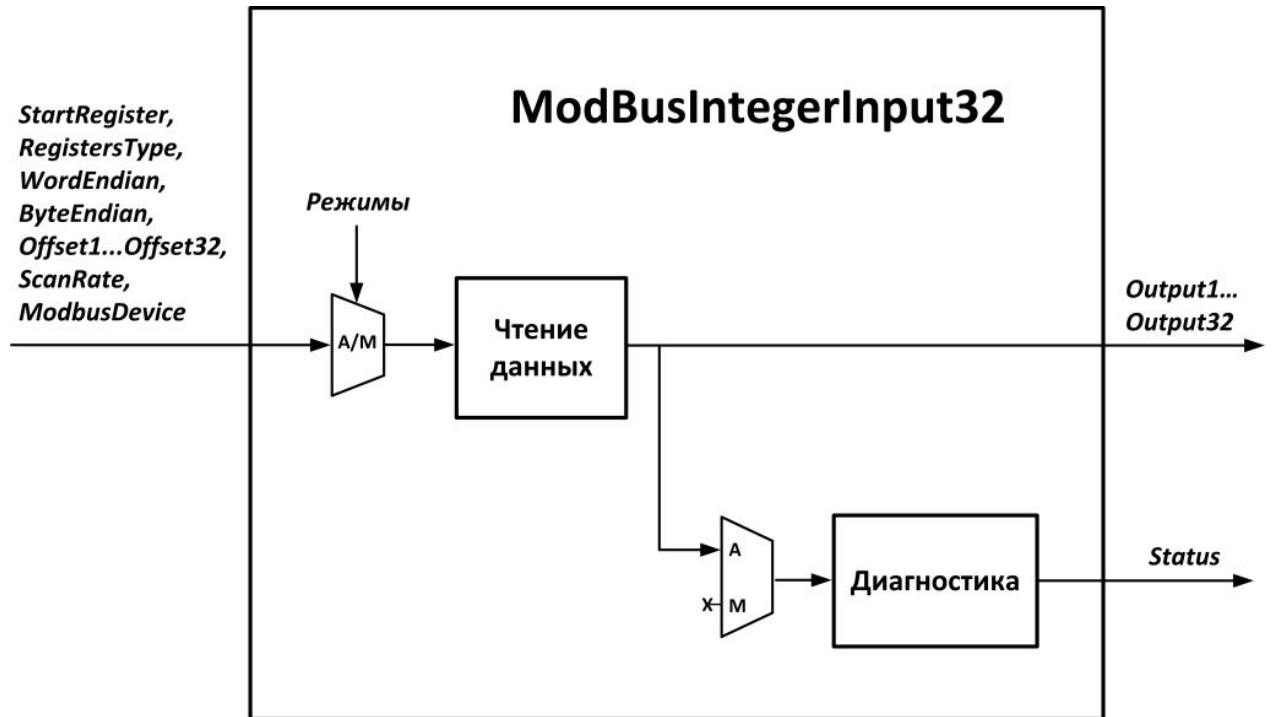
### 6.5.3. Алгоритм ModBusIntegerInput32

Выполнение алгоритма ModBusIntegerInput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение данных устройства Modbus;

- диагностика коммуникации.



A – AUTO  
M – MASKING

Рисунок 20. Схема процесса обработки данных ModBusIntegerInput32

### 6.5.3.1. Чтение данных устройства Modbus

На данном этапе производится чтение данных из регистров устройства Modbus и запись в выходные атрибуты *Output1...Output32* ModBusIntegerInput32.

Для чтения данных должны быть заданы конфигурационные атрибуты:

*StartRegister*;

*RegistersType*;

*WordEndian* и *ByteEndian*;

*Offset1...Offset32*;

*ScanRate*;

*ModbusDevice*.

### 6.5.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusIntegerInput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

## 6.6. Карта ModBusIntegerOutput32

Карта ModBusIntegerOutput32 (далее ModBusIntegerOutput32) обеспечивает выполнение следующих функций:

- организация записи данных типа INT в устройство ModBus;
- диагностика коммуникации с другими функциональными блоками.

В один ModBusIntegerOutput32 могут быть записаны данные типа INT для 32 входов устройства Modbus.

### 6.6.1. Атрибуты ModBusIntegerOutput32

ModBusIntegerOutput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 41. Дополнительные атрибуты ModBusIntegerOutput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
Input1...Input32	INT	чтение	Вход 1...Вход 32
<i>Конфигурационные атрибуты</i>			
StartRegister	INT	чтение/ запись	Начальный регистр
WordEndian	INT	чтение/ запись	Порядок слов ( <b>WordEndian</b> )
ByteEndian	INT	чтение/ запись	Порядок байтов ( <b>ByteEndian</b> )
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта ( <b>Устройство ModBusTCPDevice / Устройство ModBusRTUDevice</b> )
Source1...Source32	STRING	чтение/ запись	Имя источника входных данных

Перечень типов данных атрибутов функционального блока описан в разделе 4. *Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока* документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута **Status** для ModBusIntegerOutput32 имеют дополнительное описание:

**Таблица 42. Дополнительное описание значений атрибута Status ModBusIntegerOutput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.6.2. Режимы ModBusIntegerOutput32

Режимы ModBusIntegerOutput32:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");

В режиме MASKING этап диагностики коммуникации отключен.

### 6.6.3. Алгоритм ModBusIntegerOutput32

Выполнение алгоритма ModBusIntegerOutput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- запись данных в устройство Modbus;
- диагностика коммуникации.

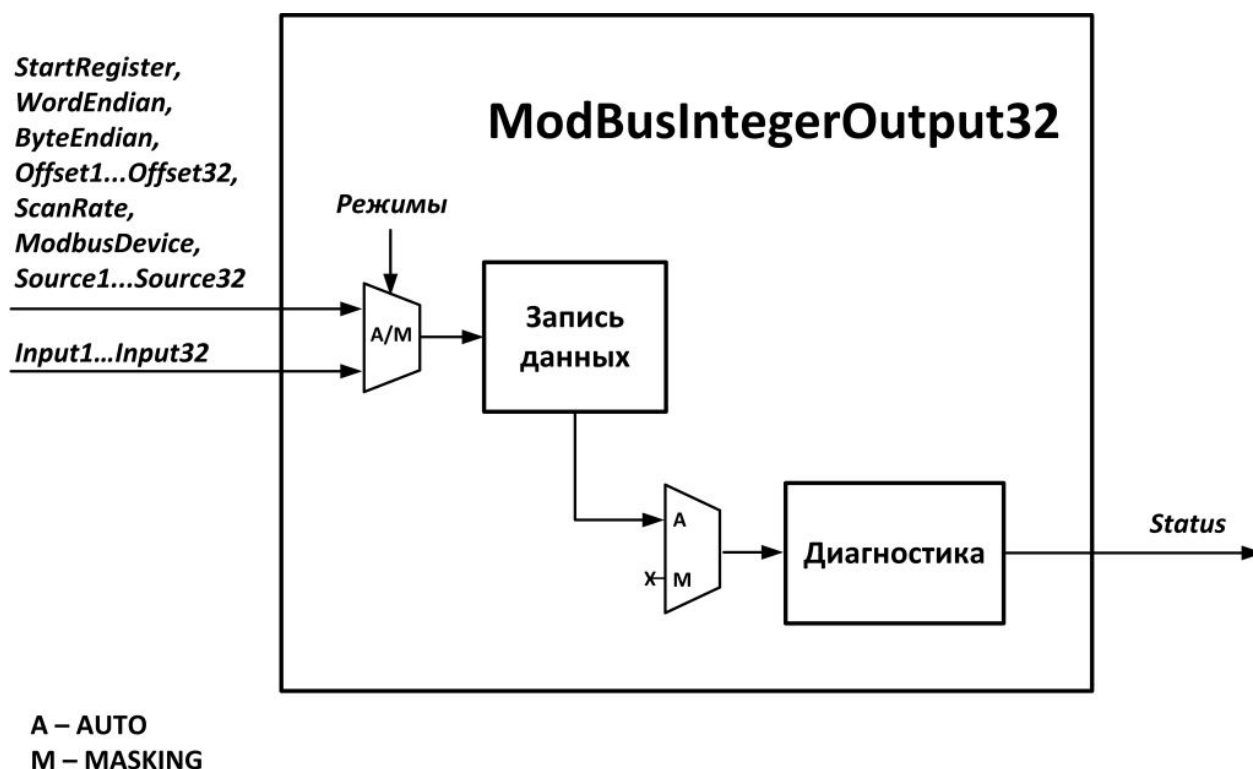


Рисунок 21. Схема процесса обработки данных ModBusIntegerOutput32

#### 6.6.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *Input1...Input32* ModBusIntegerOutput32.

Для записи данных должны быть заданы конфигурационные атрибуты:

*StartRegister*;

*WordEndian* и *ByteEndian*;

*Offset1...Offset32*;

*ScanRate*;

*ModbusDevice*;

*Source1...Source32*.

#### 6.6.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusIntegerOutput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

## 6.7. Карта ModBusUIntegerInput32

Карта ModBusUIntegerInput32 (далее ModBusUIntegerInput32) обеспечивает выполнение следующих функций:

- организация чтения данных типа UINT из устройства Modbus;
- предоставление доступа к прочитанным данным в технологической программе;
- диагностика коммуникации с другими функциональными блоками.

Один ModBusUIntegerInput32 может считать данные из 32 выходов устройства Modbus, содержащих переменные типа UINT.

### 6.7.1. Атрибуты ModBusUIntegerInput32

ModBusUIntegerInput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 43. Дополнительные атрибуты ModBusIntegerInput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Выходные атрибуты</i>			
Output1...Output32	UINT	чтение	Выход 1...Выход 32
<i>Конфигурационные атрибуты</i>			
StartRegister	INT	чтение/ запись	Начальный регистр
RegistersType	INT	чтение/ запись	Тип регистра (Input или Holding)
WordEndian	INT	чтение/ запись	Порядок слов ( <b>WordEndian</b> )
ByteEndian	INT	чтение/ запись	Порядок байтов ( <b>ByteEndian</b> )
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Имя устройства Modbus, к которому привязана карта

Перечень типов данных атрибутов функционального блока описан в разделе 4. *Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока* документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута **Status** для ModBusUIntegerInput32 имеют дополнительное описание:

**Таблица 44. Дополнительное описание значений атрибута Status ModBusUIntegerInput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.7.2. Режимы ModBusUIntegerInput32

Режимы ModBusUIntegerInput32:

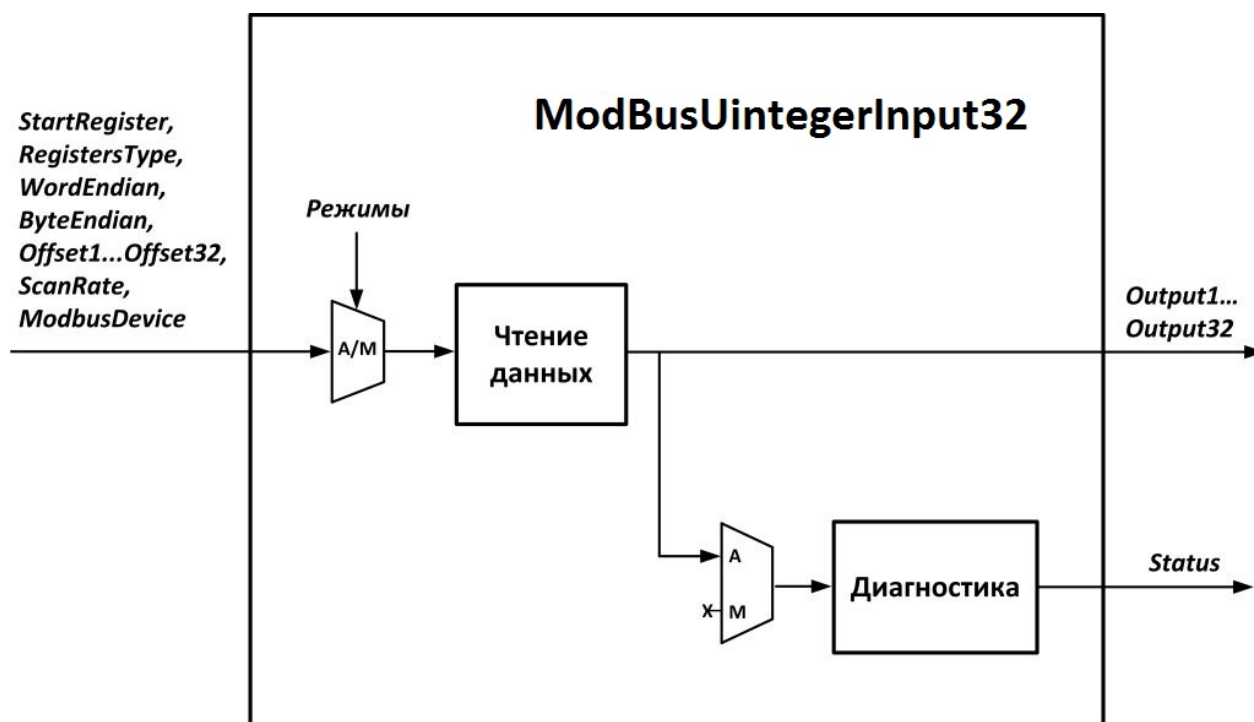
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

### 6.7.3. Алгоритм ModBusUIntegerInput32

Выполнение алгоритма ModBusUIntegerInput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение данных устройства Modbus;
- диагностика коммуникации.



A – AUTO  
M – MASKING

**Рисунок 22. Схема процесса обработки данных ModBusUIntegerInput32**

### 6.7.3.1. Чтение данных устройства Modbus

На данном этапе производится чтение данных из регистров устройства Modbus и запись в выходные атрибуты *Output1...Output32* ModBusUIntegerInput32.

Для чтения данных должны быть заданы конфигурационные атрибуты:

*StartRegister*;

*RegistersType*;

*WordEndian* и *ByteEndian*;

*Offset1...Offset32*;

*ScanRate*;

*ModbusDevice*

### 6.7.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusUIntegerInput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.



## 6.8. Карта ModBusUIntegerOutput32

Карта ModBusUIntegerOutput32 (далее ModBusUIntegerOutput32) обеспечивает выполнение следующих функций:

- организация записи данных типа UINT в устройство ModBus;
- диагностика коммуникации с другими функциональными блоками.

В один ModBusUIntegerOutput32 могут быть записаны данные типа UINT для 32 входов устройства Modbus.

### 6.8.1. Атрибуты ModBusUIntegerOutput32

ModBusUIntegerOutput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 45. Дополнительные атрибуты ModBusUIntegerOutput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
Input1...Input32	UINT	чтение	Вход 1...Вход 32
<i>Конфигурационные атрибуты</i>			
StartRegister	INT	чтение/ запись	Начальный регистр
WordEndian	INT	чтение/ запись	Порядок слов ( <b>WordEndian</b> )
ByteEndian	INT	чтение/ запись	Порядок байтов ( <b>ByteEndian</b> )
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта ( <b>Устройство ModBusTCPDevice / Устройство ModBusRTUDevice</b> )
Source1...Source32	STRING	чтение/ запись	Имя источника входных данных

Перечень типов данных атрибутов функционального блока описан в разделе 4. *Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока* документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута **Status** для ModBusUIntegerOutput32 имеют дополнительное описание:

**Таблица 46. Дополнительное описание значений атрибута Status ModBusUIntegerOutput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.8.2. Режимы ModBusUIntegerOutput32

Режимы ModBusUIntegerOutput32:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");

В режиме MASKING этап диагностики коммуникации отключен

### 6.8.3. Алгоритм ModBusUIntegerOutput32

Выполнение алгоритма ModBusUIntegerOutput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- запись данных в устройство Modbus;
- диагностика коммуникации.

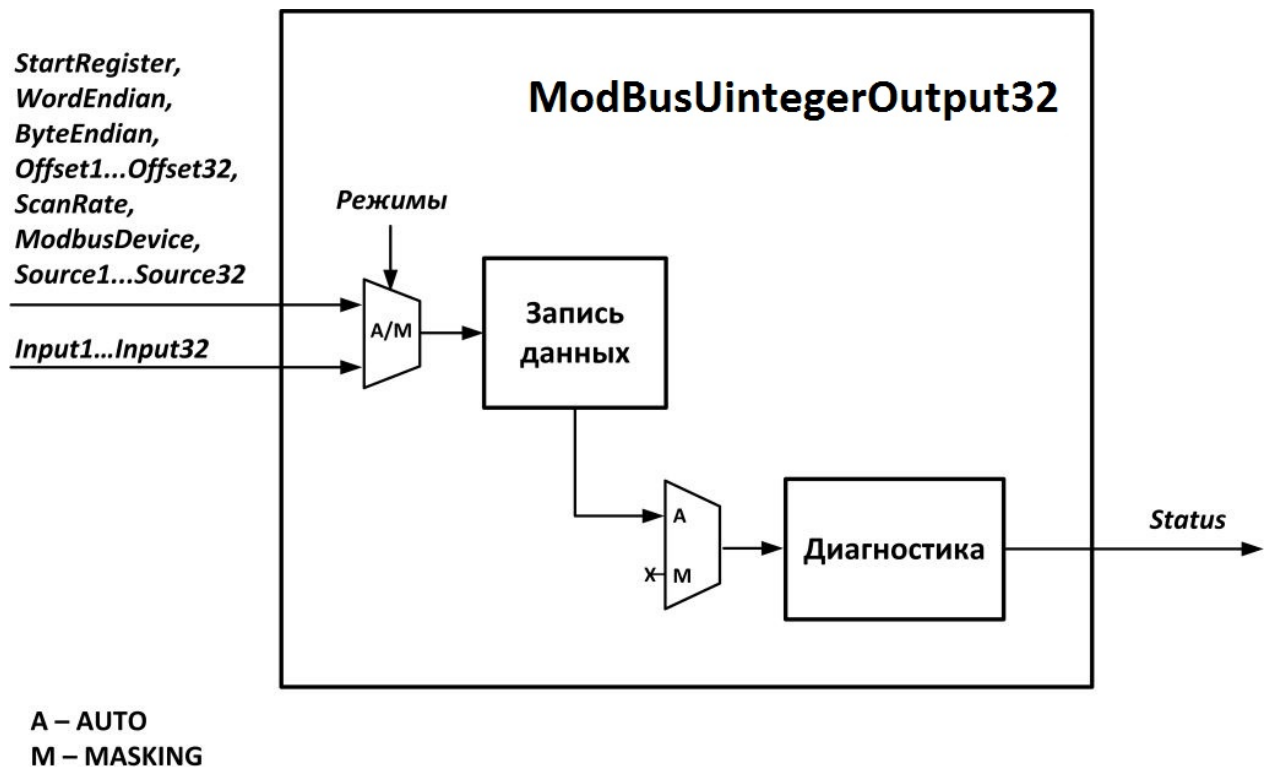


Рисунок 23. Схема процесса обработки данных ModBusUIntegerOutput32

#### 6.8.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *Input1...Input32* ModBusUIntegerOutput32.

Для записи данных должны быть заданы конфигурационные атрибуты:

*StartRegister*;

*WordEndian* и *ByteEndian*;

*Offset1...Offset32*;

*ScanRate*;

*ModbusDevice*;

*Source1...Source32*.

#### 6.8.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusUIntegerOutput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

## 6.9. Карта ModBusBooleanInput32

Карта ModBusBooleanInput32 (далее ModBusBooleanInput32) обеспечивает выполнение следующих функций:

- организация чтения данных типа BOOL из устройства Modbus;
- предоставление доступа к прочитанным данным в технологической программе;
- диагностика коммуникации с другими функциональными блоками.

Один ModBusBooleanInput32 может считать данные из 32 выходов устройства Modbus, содержащих переменные типа BOOL.

### 6.9.1. Атрибуты ModBusBooleanInput32

ModBusBooleanInput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 47. Дополнительные атрибуты ModBusBooleanInput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Выходные атрибуты</i>			
Output1...Output32	BOOL	чтение	Выход 1...Выход 32
<i>Конфигурационные атрибуты</i>			
StartBit	INT	чтение/ запись	Начальный бит
BitType	INT	чтение/ запись	Тип бита
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Имя устройства Modbus, к которому привязана карта

Перечень типов данных атрибутов функционального блока описан в разделе 4. Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута *Status* для ModBusBooleanInput32 имеют дополнительное описание:

**Таблица 48. Дополнительное описание значений атрибута *Status* ModBusBooleanInput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.9.2. Режимы ModBusBooleanInput32

Режимы ModBusBooleanInput32:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

### 6.9.3. Алгоритм ModBusBooleanInput32

Выполнение алгоритма ModBusBooleanInput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение данных устройства Modbus;
- диагностика коммуникации.

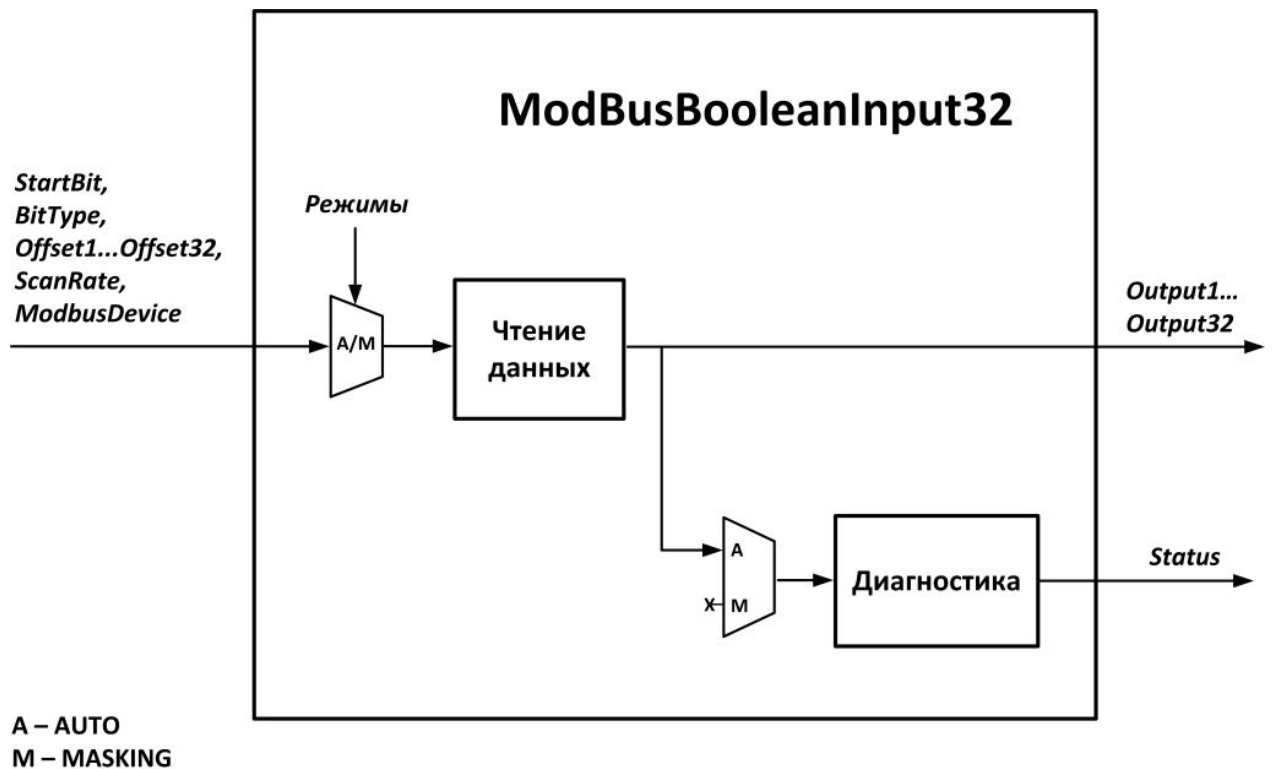


Рисунок 24. Схема процесса обработки данных ModBusBooleanInput32

### 6.9.3.1. Чтение данных устройства Modbus

На данном этапе производится чтение данных из битов устройства Modbus и запись в выходные атрибуты **Output1...Output32** ModBusBooleanInput32.

Для чтения данных должны быть заданы конфигурационные атрибуты:

**StartBit;**

**BitType;**

**Offset1...Offset32;**

**ScanRate;**

**ModbusDevice.**

### 6.9.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusBooleanInput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке **Bad.**

## 6.10. Карта ModBusBooleanOutput32

Карта ModBusBooleanOutput32 (далее ModBusBooleanOutput32) обеспечивает выполнение следующих функций:

- организация записи данных типа BOOL в устройство ModBus;
- диагностика коммуникации с другими функциональными блоками.

В один ModBusBooleanOutput32 могут быть записаны данные типа BOOL для 32 входов устройства Modbus.

### 6.10.1. Атрибуты ModBusBooleanOutput32

ModBusBooleanOutput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 49. Дополнительные атрибуты ModBusBooleanOutput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
Input1...Input32	BOOL	чтение	Вход 1...Вход 32
<i>Конфигурационные атрибуты</i>			
StartBit	INT	чтение/ запись	Начальный бит
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)
Source1...Source32	STRING	чтение/ запись	Имя источника входных данных

Перечень типов данных атрибутов функционального блока описан в разделе 4. Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута *Status* для ModBusBooleanOutput32 имеют дополнительное описание:

**Таблица 50. Дополнительное описание значений атрибута *Status* ModBusBooleanOutput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.10.2. Режимы ModBusBooleanOutput32

Режимы ModBusBooleanOutput32:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

В режиме MASKING этап диагностики коммуникации отключен.

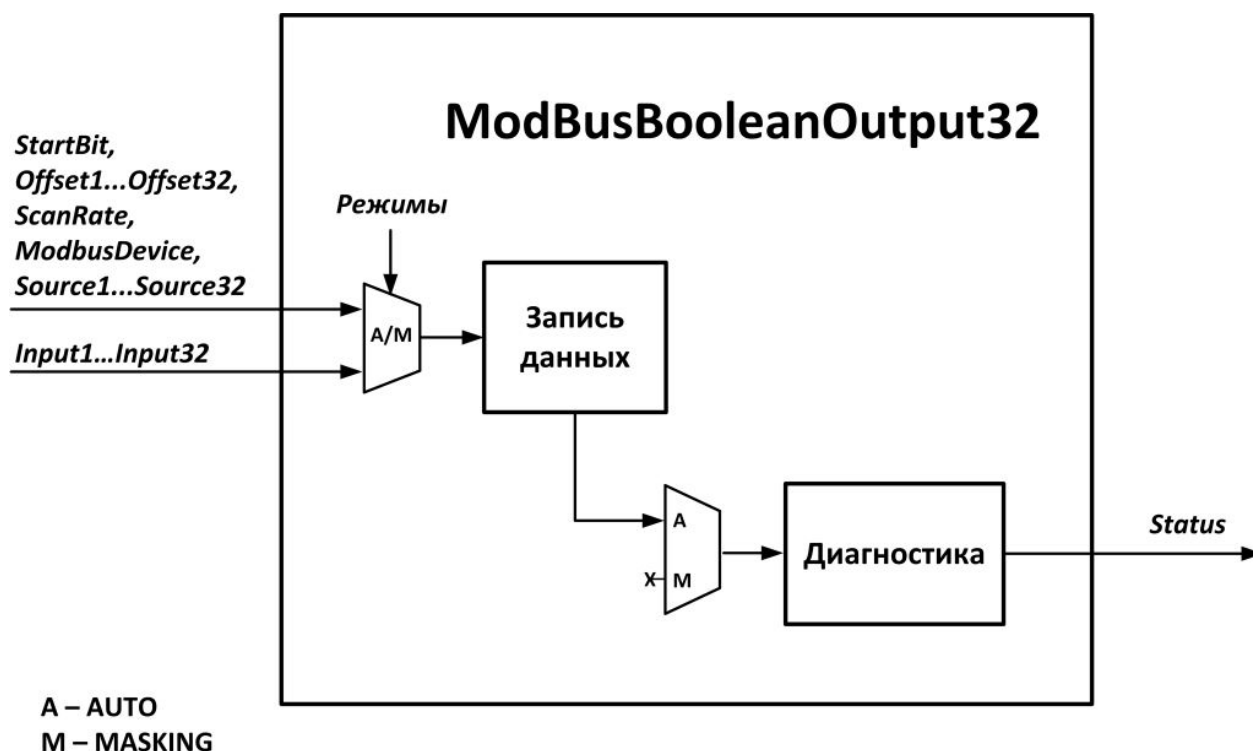
### 6.10.3. Алгоритм ModBusBooleanOutput32

Выполнение алгоритма ModBusBooleanOutput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- запись данных в устройство Modbus;
- диагностика коммуникации.





**Рисунок 25. Схема процесса обработки данных ModBusBooleanOutput32**

#### 6.10.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *Input1...Input32* ModBusBooleanOutput32 в биты устройства Modbus.

Для записи данных должны быть заданы конфигурационные атрибуты:

*StartBit*;

*Offset1...Offset32*;

*ScanRate*;

*ModbusDevice*;

*Source1...Source32*.

#### 6.10.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusBooleanOutput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

## 6.11. Карта ModBusShortInput32

Карта ModBusShortInput32 (далее ModBusShortInput32) обеспечивает выполнение следующих функций:

- организация чтения данных типа SHORT из устройства Modbus;
- предоставление доступа к прочитанным данным в технологической программе;
- диагностика коммуникации с другими функциональными блоками.

Один ModBusShortInput32 может считать данные из 32 выходов устройства Modbus, содержащих переменные типа SHORT.

### 6.11.1. Атрибуты ModBusShortInput32

ModBusShortInput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 51. Дополнительные атрибуты ModBusShortInput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Выходные атрибуты</i>			
Output1...Output32	INT	чтение	Выход 1...Выход 32
<i>Конфигурационные атрибуты</i>			
StartRegister	INT	чтение/ запись	Начальный регистр
RegistersType	INT	чтение/ запись	Тип регистра (Input или Holding)
ByteEndian	INT	чтение/ запись	Порядок байтов (ByteEndian)
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Имя устройства Modbus, к которому привязана карта

Перечень типов данных атрибутов функционального блока описан в разделе 4. Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов

функционального блока документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута **Status** для ModBusShortInput32 имеют дополнительное описание:

**Таблица 52. Дополнительное описание значений атрибута Status ModBusShortInput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.11.2. Режимы ModBusShortInput32

Режимы ModBusShortInput32:

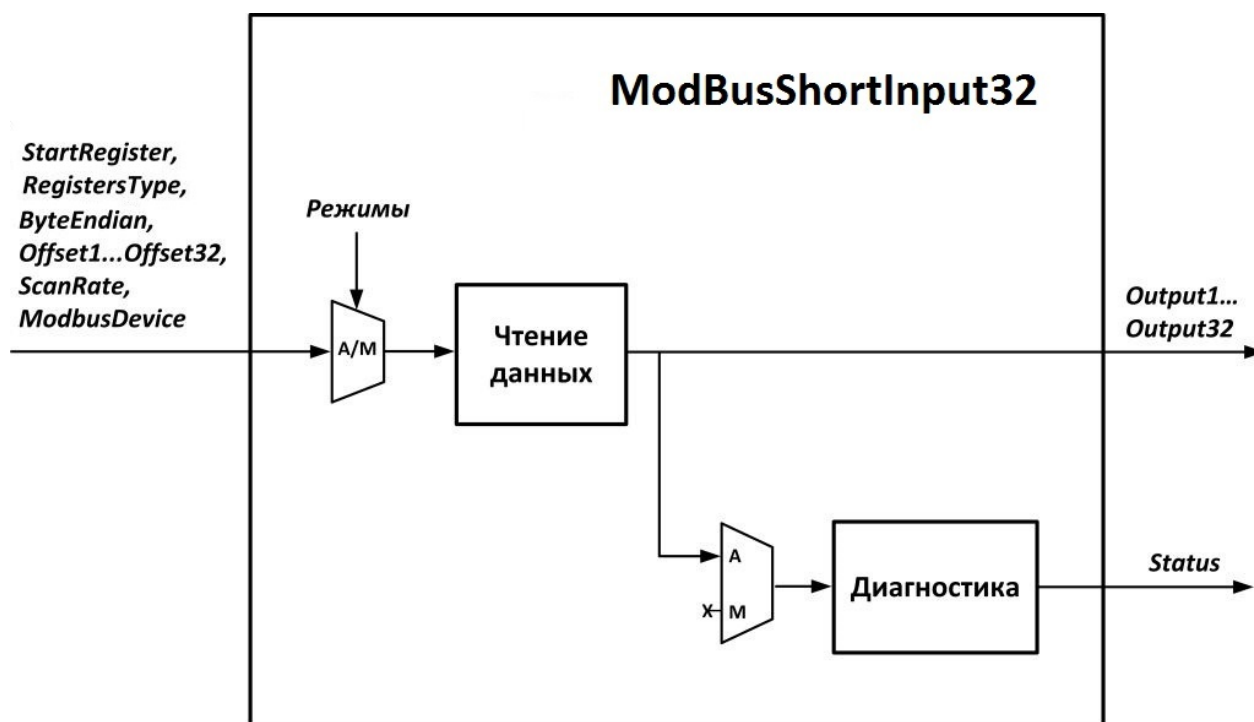
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

### 6.11.3. Алгоритм ModBusShortInput32

Выполнение алгоритма ModBusShortInput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение данных устройства Modbus;
- диагностика коммуникации.



A – AUTO  
M – MASKING

Рисунок 26. Схема процесса обработки данных ModBusShortInput32

### 6.11.3.1. Чтение данных устройства Modbus

На данном этапе производится чтение данных из регистров устройства Modbus и запись в выходные атрибуты **Output1...Output32** ModBusShortInput32. Каждый считываемый регистр устройства Modbus будет расширен до типа данных INT путем добавления двух старших значащих байтов.

Для чтения данных должны быть заданы конфигурационные атрибуты:

**StartRegister**;

**RegistersType**;

**ByteEndian**;

**Offset1...Offset32**;

**ScanRate**;

**ModbusDevice**

### 6.11.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusShortInput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке **Bad**.

## 6.12. Карта ModBusShortOutput32

Карта ModBusShortOutput32 (далее ModBusShortOutput32) обеспечивает выполнение следующих функций:

- организация записи данных типа SHORT в устройство ModBus;
- диагностика коммуникации с другими функциональными блоками.

В один ModBusShortOutput32 могут быть записаны данные типа SHORT для 32 входов устройства Modbus.

### 6.12.1. Атрибуты ModBusShortOutput32

ModBusShortOutput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 53. Дополнительные атрибуты ModBusShortOutput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
Input1...Input32	INT	чтение	Вход 1...Вход 32
<i>Конфигурационные атрибуты</i>			
StartRegister	INT	чтение/ запись	Начальный регистр
ByteEndian	INT	чтение/ запись	Порядок байтов (ByteEndian)
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ModbusDevice	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)
Source1...Source32	STRING	чтение/ запись	Имя источника входных данных

Перечень типов данных атрибутов функционального блока описан в разделе 4. Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута *Status* для ModBusShortOutput32 имеют дополнительное описание:

**Таблица 54. Дополнительное описание значений атрибута *Status* ModBusShortOutput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.12.2. Режимы ModBusShortOutput32

Режимы ModBusShortOutput32:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");

В режиме MASKING этап диагностики коммуникации отключен.

### 6.12.3. Алгоритм ModBusShortOutput32

Выполнение алгоритма ModBusShortOutput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- запись данных в устройство Modbus;
- диагностика коммуникации.

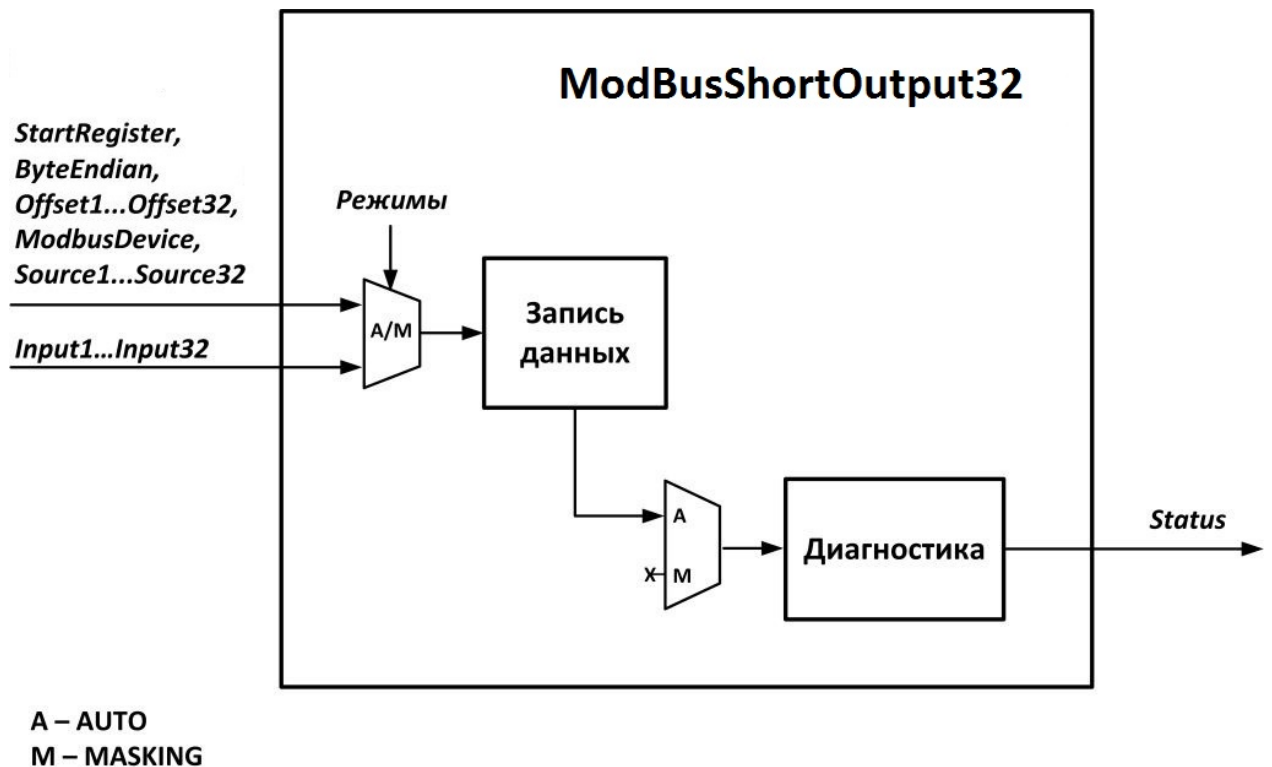


Рисунок 27. Схема процесса обработки данных ModBusShortOutput32

#### 6.12.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *Input1...Input32* ModBusShortOutput32. Каждый входной атрибут *Input1...Input32* усекается до типа SHORT путем отбрасывания двух старших значащих байтов.

Для записи данных должны быть заданы конфигурационные атрибуты:

*StartRegister*;

*ByteEndian*;

*Offset1...Offset32*;

*ModbusDevice*;

*Source1...Source32*.

#### 6.12.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusShortOutput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

## 6.13. Карта ModBusUshortInput32

Карта ModBusUshortInput32 (далее ModBusUshortInput32) обеспечивает выполнение следующих функций:

- организация чтения данных типа USHORT из устройства Modbus;
- предоставление доступа к прочитанным данным в технологической программе;
- диагностика коммуникации с другими функциональными блоками.

Один ModBusUshortInput32 может считать данные из 32 выходов устройства Modbus, содержащих переменные типа USHORT.

### 6.13.1. Атрибуты ModBusUshortInput32

ModBusUshortInput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 55. Дополнительные атрибуты ModBusUshortInput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Выходные атрибуты</i>			
Output1...Output32	INT	чтение	Выход 1...Выход 32
<i>Конфигурационные атрибуты</i>			
StartRegister	INT	чтение/ запись	Начальный регистр
RegistersType	INT	чтение/ запись	Тип регистра (Input или Holding)
ByteEndian	INT	чтение/ запись	Порядок байтов (ByteEndian)
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ScanRate	INT	чтение/ запись	Период опроса, мс
ModbusDevice	STRING	чтение/ запись	Имя устройства Modbus, к которому привязана карта

Перечень типов данных атрибутов функционального блока описан в разделе 4. Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов



функционального блока документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута **Status** для ModBusUshortInput32 имеют дополнительное описание:

**Таблица 56. Дополнительное описание значений атрибута Status ModBusUshortInput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

### 6.13.2. Режимы ModBusUshortInput32

Режимы ModBusUshortInput32:

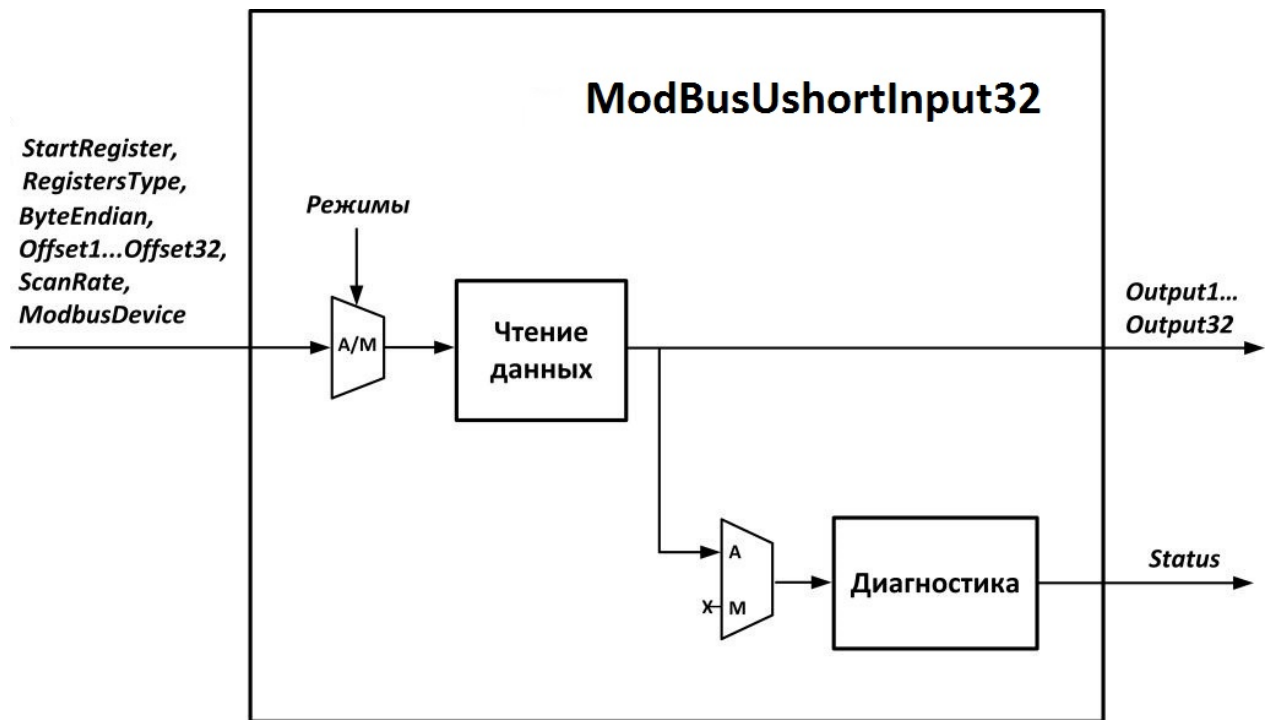
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

### 6.13.3. Алгоритм ModBusUshortInput32

Выполнение алгоритма ModBusUshortInput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение данных устройства Modbus;
- диагностика коммуникации.



A – AUTO  
M – MASKING

Рисунок 28. Схема процесса обработки данных ModBusUshortInput32

### 6.13.3.1. Чтение данных устройства Modbus

На данном этапе производится чтение данных из регистров устройства Modbus и запись в выходные атрибуты **Output1...Output32** ModBusUshortInput32. Каждый считываемый регистр устройства Modbus будет расширен до типа данных INT путем добавления двух нулевых старших значащих байтов.

Для чтения данных должны быть заданы конфигурационные атрибуты:

**StartRegister**;

**RegistersType**;

**ByteEndian**;

**Offset1...Offset32**;

**ScanRate**;

**ModbusDevice**

### 6.13.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusUshortInput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке **Bad**.

## 6.14. Карта ModBusUshortOutput32

Карта ModBusUshortOutput32 (далее ModBusUshortOutput32) обеспечивает выполнение следующих функций:

- организация записи данных типа USHORT в устройство ModBus;
- диагностика коммуникации с другими функциональными блоками.

В один ModBusUshortOutput32 могут быть записаны данные типа USHORT для 32 входов устройства Modbus.

### 6.14.1. Атрибуты ModBusUshortOutput32

ModBusUshortOutput32 имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 57. Дополнительные атрибуты ModBusUshortOutput32**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
Input1...Input32	INT	чтение	Вход 1...Вход 32
<i>Конфигурационные атрибуты</i>			
StartRegister	INT	чтение/ запись	Начальный регистр
ByteEndian	INT	чтение/ запись	Порядок байтов (ByteEndian)
Offset1...Offset32	INT	чтение/ запись	Смещение 1...Смещение 32
ModbusDevice	STRING	чтение/ запись	Устройство Modbus, к которому привязана карта (Устройство ModBusTCPDevice / Устройство ModBusRTUDevice)
Source1...Source32	STRING	чтение/ запись	Имя источника входных данных

Перечень типов данных атрибутов функционального блока описан в разделе 4. Атрибуты функциональных блоков Таблица 2. Типы данных атрибутов функционального блока документа "Концепция технологического программного обеспечения".

Некоторые значения атрибута *Status* для ModBusUshortOutput32 имеют дополнительное описание:

**Таблица 58. Дополнительное описание значений атрибута *Status* ModBusUshortOutput32**

Имя	Значение	Описание
BadSourceReference	0x02020000	Не прогружено Modbus устройство.
BadConfiguration	0x02030000	Ошибка в конфигурации карты (например, неверно заданы смещения).
CommunicationError	0x02070000	Отсутствует коммуникация с устройством.
WaitForService	0x01020000	Ожидание коммуникации с драйвером Modbus.

#### 6.14.2. Режимы ModBusUshortOutput32

Режимы ModBusUshortOutput32:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");

В режиме MASKING этап диагностики коммуникации отключен.

#### 6.14.3. Алгоритм ModBusUshortOutput32

Выполнение алгоритма ModBusUshortOutput32 зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- запись данных в устройство Modbus;
- диагностика коммуникации.

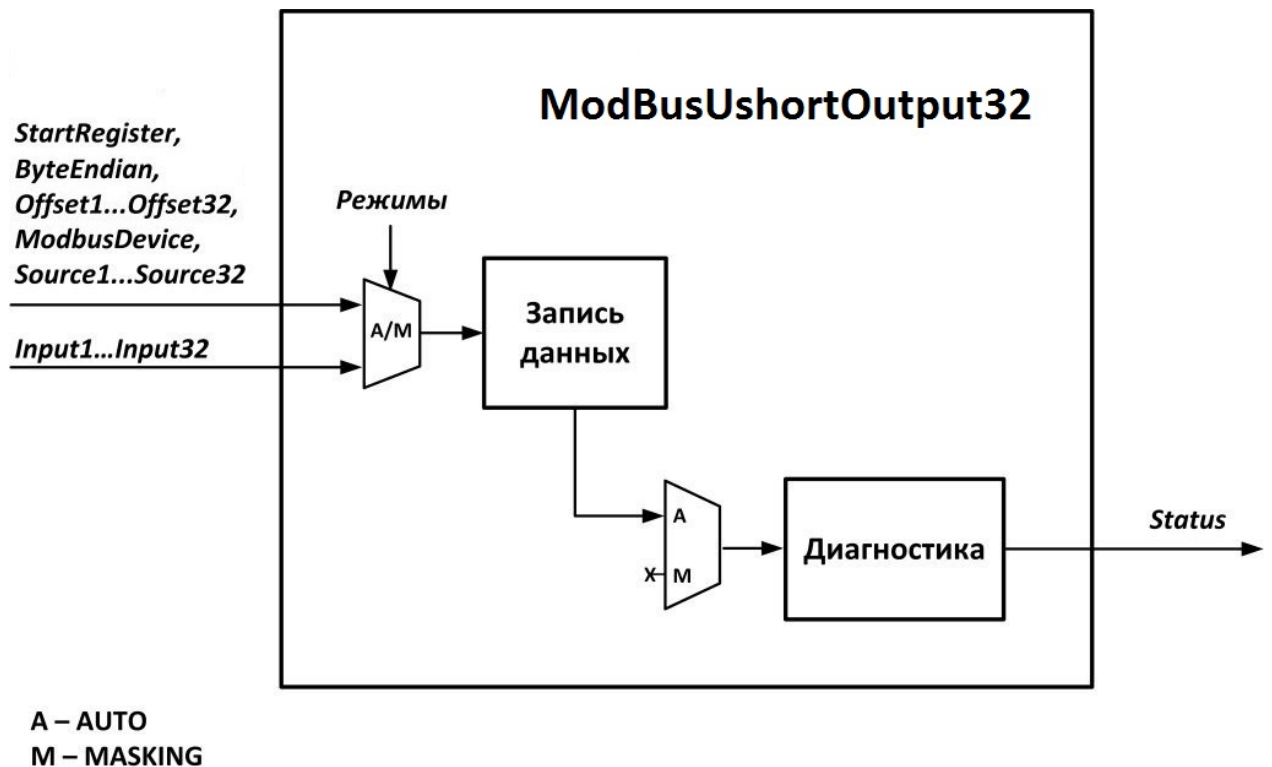


Рисунок 29. Схема процесса обработки данных ModBusUshortOutput32

#### 6.14.3.1. Запись данных в Modbus устройство

На данном этапе производится запись данных из входных атрибутов *Input1...Input32* ModBusUshortOutput32. Каждый входной атрибут *Input1...Input32* усекается до типа SHORT путем отбрасывания двух старших значащих байтов.

Для записи данных должны быть заданы конфигурационные атрибуты:

*StartRegister*;

*ByteEndian*;

*Offset1...Offset32*;

*ModbusDevice*;

*Source1...Source32*.

#### 6.14.3.2. Диагностика коммуникации

В процессе диагностики осуществляется проверка коммуникации карты ModBusUshortOutput32 с устройством Modbus.

При отсутствии коммуникации формируется сигнализационный флаг об ошибке *Bad*.

# 7. Технологические функциональные блоки

## 7.1. Функциональный блок Control Button

Функциональный блок Control Button (далее Control Button) передает в контроллер дискретные управляющие импульсы.

### 7.1.1. Атрибуты Control Button

Control Button имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 59. Дополнительные атрибуты Control Button

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
EN	BOOL	чтение	Разрешение команды
<i>Выходные атрибуты</i>			
OUT	BOOL	чтение	Выход
<i>Конфигурационные атрибуты</i>			
EN_SRC	STRING	чтение/ запись	Включение источника
CMD	BOOL	чтение/запись	Команда
CTRL_TYPE	USINT	чтение/запись	Тип управления

### 7.1.2. Режимы Control Button

Режимы Control Button:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.1.3. Алгоритм Control Button

Control Button имеет два типа управления

- импульс;
- удержание.

Когда на Control Button установлен тип данных "импульс", функциональный блок выдает импульс на один скан, сбрасывает команду и готов для следующей команды, а при типе данных "удержание" выходное значение равно команде.

## 7.2. Входной аналоговый блок AnalogInputPoint

Входной аналоговый блок AnalogInputPoint (далее AnalogInputPoint) обеспечивает выполнение следующих функций:

- считывание и контроль достоверности данных аналогового датчика;
- преобразование данных аналогового датчика в инженерные единицы;
- сравнение полученных данных с величиной предупредительных и аварийных уставок и формирование сигнализаций.

### 7.2.1. Атрибуты AnalogInputPoint

AnalogInputPoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 60. Дополнительные атрибуты AnalogInputPoint**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
IN	REAL	чтение	Входное значение в процентах
HH_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1.
HI_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1.
LO_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1.
LL_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1.
PV	REAL	чтение	Выходное значение
HH_ACT	BOOL	чтение	Срабатка аларма до разрешающего бита HH_EN. Учитывает гистерезис

Имя атрибута	Тип данных	Доступ	Описание атрибута
HI_ACT	BOOL	чтение	Сработка аларма до разрешающего бита HI_EN. Учитывает гистерезис
LO_ACT	BOOL	чтение	Сработка аларма до разрешающего бита LO_EN. Учитывает гистерезис
LL_ACT	BOOL	чтение	Сработка аларма до разрешающего бита LL_EN. Учитывает гистерезис
HH_NACT	BOOL	чтение	Инвертированный HH_ACT, имеет статус PV
HI_NACT	BOOL	чтение	Инвертированный HI_ACT, имеет статус PV
LO_NACT	BOOL	чтение	Инвертированный LO_ACT, имеет статус PV
LL_NACT	BOOL	чтение	Инвертированный LL_ACT, имеет статус PV
<b>Конфигурационные атрибуты</b>			
IN_SRC	STRING	чтение/ запись	Имя функционального блока-источника данных
IV	REAL	чтение/ запись	Ручное значение, вводимое в режимах имитация и имитация маскирования
IN_SL	REAL	чтение/ запись	Нижняя граница шкалы датчика в процентах
IN_SH	REAL	чтение/ запись	Верхняя граница шкалы датчика в процентах
PV_SL	REAL	чтение/ запись	Нижняя граница шкалы датчика в инженерных единицах
PV_SH	REAL	чтение/ запись	Верхняя граница шкалы датчика в инженерных единицах
HH_LIM	REAL	чтение/ запись	Верхняя аварийная уставка в инженерных единицах
HI_LIM	REAL	чтение/ запись	Верхняя предупредительная уставка в инженерных единицах



Имя атрибута	Тип данных	Доступ	Описание атрибута
LO_LIM	REAL	чтение/ запись	Нижняя предупредительная уставка в инженерных единицах
LL_LIM	REAL	чтение/ запись	Нижняя аварийная уставка в инженерных единицах
ALM_HYS	REAL	чтение/ запись	Уставка гистерезиса в процентах
IN_SHX	REAL	чтение/ запись	Верхнее расширение необработанного значения
IN_SLX	REAL	чтение/ запись	Нижнее расширение необработанного значения
HH_EN	BOOL	чтение/ запись	Включение/выключение сигнализации HiHiAlarm
HI_EN	BOOL	чтение/ запись	Включение/выключение сигнализации HiAlarm
LO_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LoAlarm
LL_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LoLoAlarm
CONV_TYPE	INT	чтение/ запись	Преобразование входного значения: Linear - линейное, SQRT - квадратичное
FV	REAL	чтение	Значение поля
HH_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала HH
HI_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала HI
LO_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала LO
LL_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала LL
HH	BOOL	чтение	Переход выходного значения в диапазон верхних аварийных значений. Аларм HH == HH_ACT && HH_CND_EN && HH_EN

Имя атрибута	Тип данных	Доступ	Описание атрибута
HI	BOOL	чтение	Переход выходного значения в диапазон верхних предупредительных значений. Аларм HI == HI_ACT && HI_CND_EN && HI_EN
LO	BOOL	чтение	Переход выходного значения в диапазон нижних предупредительных значений. Аларм LO == LO_ACT && LO_CND_EN && LO_EN
LL	BOOL	чтение	Переход выходного значения в диапазон нижних аварийных значений. Аларм LL == LL_ACT && LL_CND_EN && LL_EN

### 7.2.2. Режимы AnalogInputPoint

Режимы AnalogInputPoint:

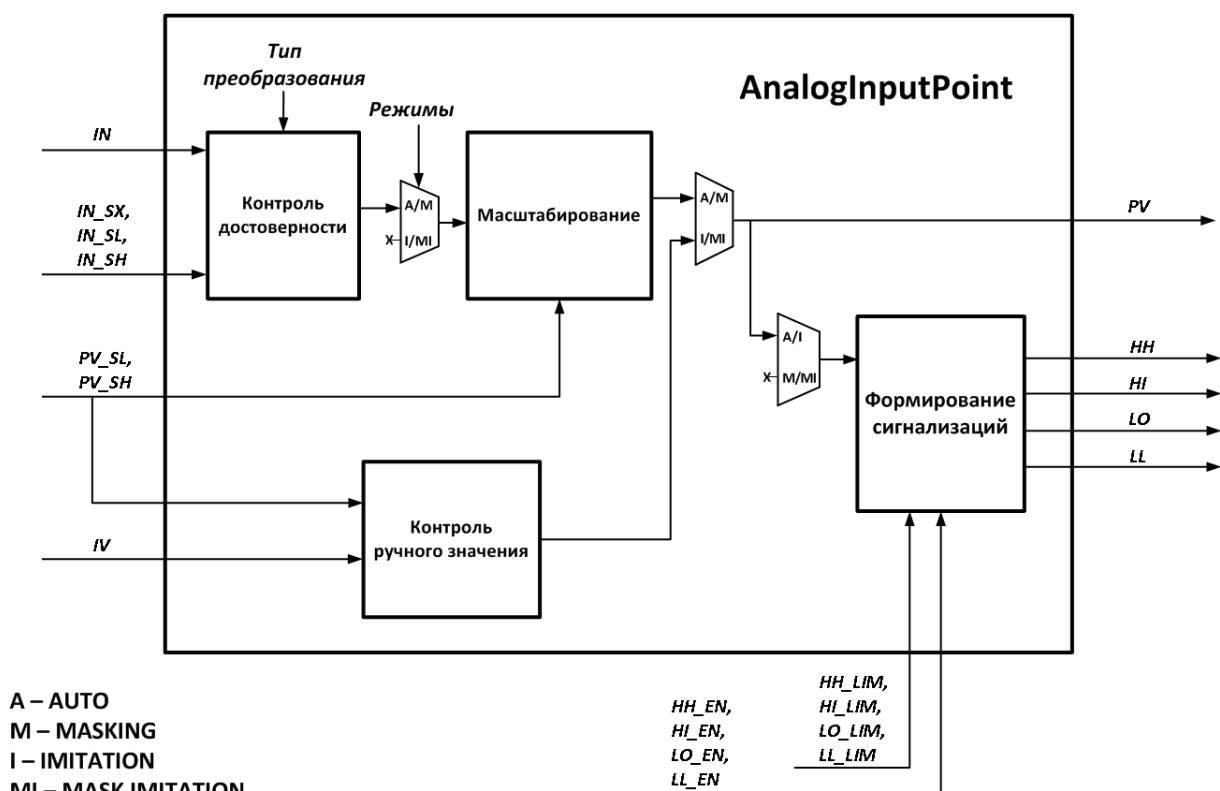
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 7.2.3. Алгоритм AnalogInputPoint

Выполнение алгоритма AnalogInputPoint зависит от установленного режима.

Алгоритм содержит следующие элементы:

- контроль достоверности;
- масштабирование;
- контроль ручного значения;
- формирование сигнализаций.



**Рисунок 30. Схема процесса обработки данных AnalogInputPoint**

### 7.2.3.1. Контроль достоверности

В процессе контроля достоверности осуществляется проверка входного значения на соответствие диапазону шкалы датчика.

Если входное значение выходит за пределы допустимого диапазона, то атрибут **Status** AnalogInputPoint примет значение 0x02030001 (BadInput).

Расширение необработанного значения нижнего диапазона **IN\_SLX** и расширение необработанного значения верхнего диапазона входного значения **IN\_SHX** позволяют диагностировать неисправность датчика при отсутствии обрыва или короткого замыкания.

Если диапазон шкалы датчика задан некорректно (например, верхний предел шкалы меньше нижнего предела), атрибут **Status** блока AnalogInputPoint примет значение 0x02030000 (BadConfiguration).

Различают типы преобразования входных сигналов (**CONV\_TYPE**): Linear - линейное, SQRT - квадратичное. Линейное преобразование позволяет использовать текущее значение сигнала, квадратичное - квадратный корень, извлеченный из значения сигнала.

### 7.2.3.2. Масштабирование

На данном этапе осуществляется масштабирование входного значения в инженерные единицы и запись в атрибут *PV*.

Масштабирование производится согласно формуле:

$$PV = \frac{PV_{SH} - PV_{SL}}{IN_{SH} - IN_{SL}} \cdot (IN - IN_{SL}) + PV_{SL},$$

где *IN\_SH* и *IN\_SL* - верхняя и нижняя границы шкалы датчика в процентах, *PV\_SH* и *PV\_SL* - верхняя и нижняя границы шкалы датчика в инженерных единицах.

При квадратичном преобразовании входного значения (установлен флаг *CONV\_TYPE*) масштабирование производится согласно формуле:

$$PV = \frac{PV_{SH} - PV_{SL}}{IN_{SH} - IN_{SL}} \cdot (\sqrt{IN} - IN_{SL}) + PV_{SL}$$

### 7.2.3.3. Контроль ручного значения

Контроль ручного значения позволяет сравнить значение атрибута *IV* со шкалой датчика в инженерных единицах.

В случае выхода значения *IV* за границы *PV\_SL* либо *PV\_SH* значение будет приравнено к величине соответствующей границы.

### 7.2.3.4. Формирование сигнализаций

Этап формирования сигнализаций функционального блока описан в разделе 8.1. Формирование сигнализаций Библиотеки.

## 7.3. Выходной аналоговый блок AnalogOutputPoint

Выходной аналоговый блок AnalogOutputPoint (далее AnalogOutputPoint) обеспечивает выполнение следующих функций:

- считывание и контроль достоверности аналогового сигнала;
- обработка аналогового сигнала;
- контроль достоверности и обработка ручного значения в режиме MANUAL;
- диагностика состояния управляющего канала.

### 7.3.1. Атрибуты AnalogOutputPoint

AnalogOutputPoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 61. Дополнительные атрибуты AnalogOutputPoint**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i><b>Входные/выходные атрибуты</b></i>			
IN	REAL	чтение	Входное значение в инженерных единицах
FLT	BOOL	чтение	Значение, отображающее состояние физического канала приема данных
OUT	REAL	чтение	Выходное значение
CAS0	BOOL	чтение	Готовность к управлению выход
<i><b>Конфигурационные атрибуты</b></i>			
IN_SRC	STRING	чтение/ запись	Имя функционального блока-источника данных
FLT_SRC	STRING	чтение/ запись	Имя функционального блока-источника состояния физического канала
MV	REAL	чтение/ запись	Ручное значение
IN_SL	REAL	чтение/ запись	Нижняя граница диапазона управляющего сигнала исполнительного механизма в процентах
IN_SH	REAL	чтение/ запись	Верхняя граница диапазона управляющего сигнала исполнительного механизма в процентах
OUT_SL	REAL	чтение/ запись	Нижняя граница диапазона управляющего сигнала исполнительного механизма в инженерных единицах

Имя атрибута	Тип данных	Доступ	Описание атрибута
OUT_SH	REAL	чтение/ запись	Верхняя граница диапазона управляющего сигнала исполнительного механизма в инженерных единицах
FV	REAL	чтение	Значение поля
INVERT	BOOL	чтение/ запись	Инвертирование выхода

### 7.3.2. Диагностика состояния физического канала

Для диагностики состояния физического канала применяются атрибуты *FLT* и *FLT\_SRC*.

В конфигурационный атрибут *FLT\_SRC* записывается имя источника диагностических данных.

Значение входного атрибута *FLT* показывает состояние физического канала. Если данный атрибут принимает значение *true*, атрибут *Status* блока AnalogOutputPoint принимает значение 0x02030001 (BadInput).

### 7.3.3. Режимы AnalogOutputPoint

Режимы AnalogOutputPoint:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MANUAL ("Ручной");
- MASKING ("Маскирование").

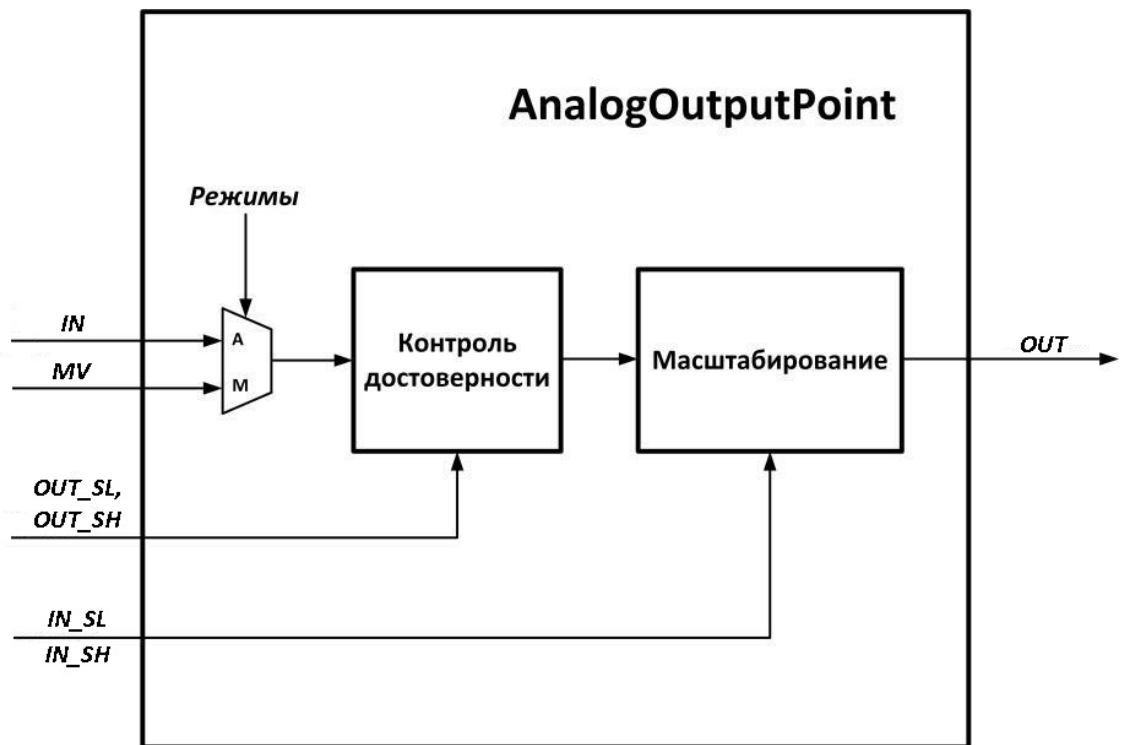
В режиме MASKING сигнализационный флаг об ошибке *Bad* не формируется.

### 7.3.4. Алгоритм AnalogOutputPoint

Выполнение алгоритма AnalogOutputPoint зависит от установленного режима.

Алгоритм содержит следующие элементы:

- контроль достоверности входных данных;
- преобразование;
- диагностика состояния физического канала.



**A – AUTO**  
**M – MANUAL**

**Рисунок 31. Схема процесса обработки данных AnalogOutputPoint**

#### 7.3.4.1. Контроль достоверности

В процессе контроля достоверности осуществляется проверка входного значения на соответствие диапазону шкалы исполнительного механизма в инженерных единицах.

Если входное значение выходит за пределы допустимого диапазона, то атрибут **Status** AnalogOutputPoint примет значение 0x02030001 (BadInput), а само входное значение приравнивается к соответствующему пределу диапазона.

Если диапазон шкалы исполнительного механизма задан некорректно (верхний предел шкалы равен нижнему пределу), атрибут **Status** примет значение 0x02030000 (BadConfiguration).

#### 7.3.4.2. Масштабирование

На данном этапе осуществляется масштабирование входного значения из инженерных единиц в единицы шкалы исполнительного механизма и запись в атрибут **OUT**.

Преобразование производится согласно формуле:

если **INVERT** принимает значение **false**

$$OUT = IN\_SL + k \cdot (IN - OUT\_SL),$$

где

$$k = \frac{IN\_SH - IN\_SL}{OUT\_SH - OUT\_SL}$$

или

если *INVERT* принимает значение *true*

$$OUT = IN\_SH - k \cdot (IN - OUT\_SL),$$

где

$$k = \frac{IN\_SH - IN\_SL}{OUT\_SH - OUT\_SL}$$

где *IN\_SH* и *IN\_SL* - верхняя и нижняя границы диапазона управляющего сигнала исполнительного механизма в процентах,

*OUT\_SH* и *OUT\_SL* - верхняя и нижняя границы диапазона управляющего сигнала исполнительного механизма в инженерных единицах.

## 7.4. Дискретный блок DiscretePoint

Дискретный блок DiscretePoint (далее DiscretePoint) обеспечивает выполнение следующих функций:

- контроль и управление дискретными значениями датчика;
- фильтрация дребезга;
- формирование сигнализаций.

### 7.4.1. Атрибуты DiscretePoint

DiscretePoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 62. Дополнительные атрибуты DiscretePoint

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
IN	BOOL	чтение	Входное значение



Имя атрибута	Тип данных	Доступ	Описание атрибута
DISC_CND_EN	BOOL	чтение	Если не имеет сорса, то == 1. Бит контекстного управления алармом со стороны алгоритмов.
OUT	BOOL	чтение	Выходное значение
DISC_ACT	BOOL	чтение	Сработка аларма до разрешающего бита DISC_EN. Учитывает время фильтрации
DISC_NACT	BOOL	чтение	Инвертированный DISC_ACT, имеет статус OUT
<b>Конфигурационные атрибуты</b>			
IN_SRC	STRING	чтение/ запись	Имя функционального блока-источника данных
IV	BOOL	чтение/ запись	Входное значение, вводимое в режимах имитация и имитация маскирования
INVERT	BOOL	чтение/ запись	Переключение на противоположное дискретное значение
ALM_EN	BOOL	чтение/ запись	Включение/выключение сигнализации
DISC_LIM	BOOL	чтение/ запись	Значение выходного атрибута, при котором формируется сигнализация
ALM_PRI	DINT	чтение/ запись	Приоритет сигнализации
PV_FTIME	DINT	чтение/ запись	Уставка временного промежутка (в миллисекундах), в котором изменения входного атрибута игнорируются
FV	BOOL	чтение	Значение поля
DISC_CND_EN_SR	STRING	чтение/ запись	Включение источника входного сигнала
<b>Сигнализационные атрибуты</b>			
DISC_ALM	BOOL	чтение	Сигнализация

## 7.4.2. Режимы DiscretePoint

Режимы DiscretePoint:

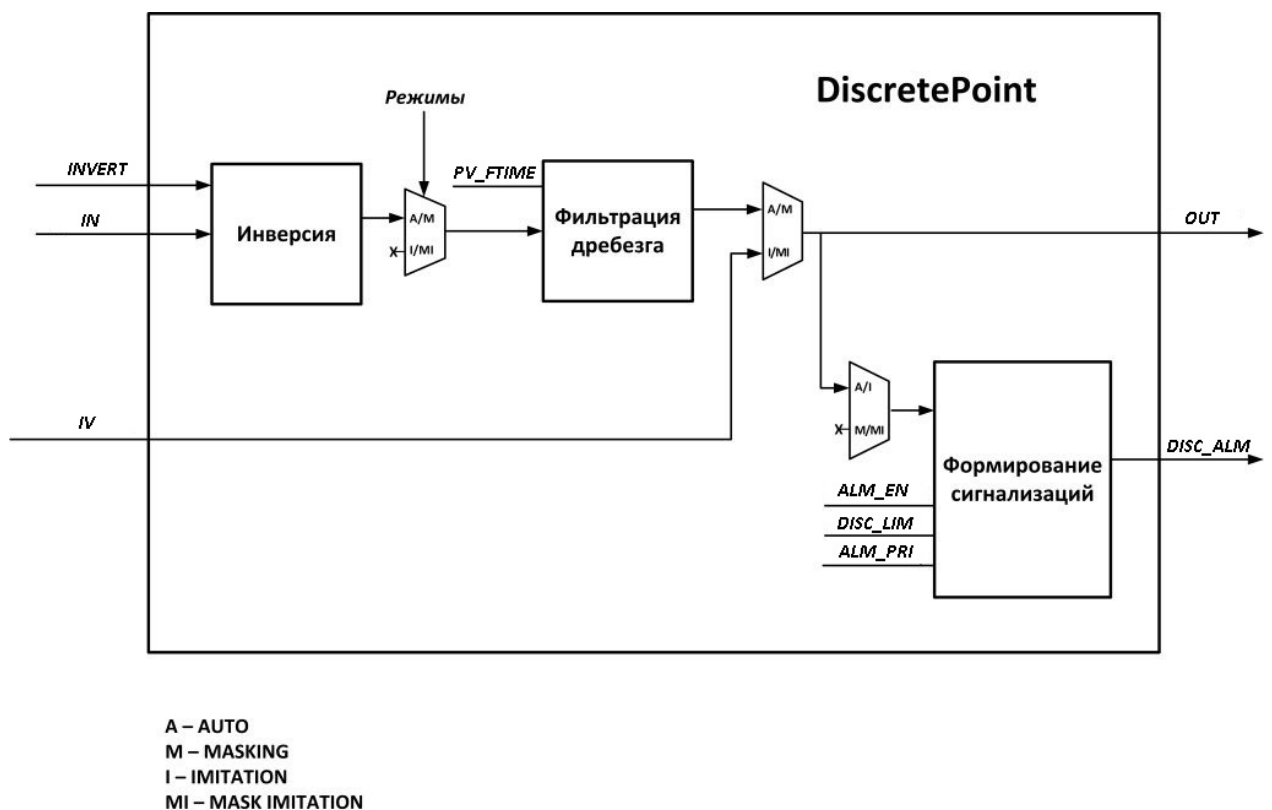
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

## 7.4.3. Алгоритм DiscretePoint

Выполнение алгоритма DiscretePoint зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- инверсия;
- фильтрация дребезга;
- формирование сигнализаций.



**Рисунок 32. Схема процесса обработки данных DiscretePoint**

### 7.4.3.1. Инверсия

При инверсии происходит переключение входного атрибута на противоположное значение.

Переключение выполняется, если атрибуту *INVERT* задано значение 1 (true).

### 7.4.3.2. Фильтрация дребезга

При фильтрации дребезга учитывается возможный дребезг контактов датчиков.

Каждый раз, когда входной атрибут меняет свое значение, осуществляется сравнение уставки *PV\_FTIME* с периодом, в котором данное значение остается неизменным.

Если указанный период не превышает уставку, изменение значения игнорируется.

### 7.4.3.3. Формирование сигнализаций

При формировании сигнализаций атрибут *DISC\_ALM* принимает значение 1 (true).

Сигнализации формируются при одновременном выполнении следующих условий:

- функция формирования сигнализаций активирована. С этой целью атрибуту *ALM\_EN* задается значение 1 (true).
- Выходное значение атрибута *OUT* совпадает с заданным значением атрибута *DISC\_LIM*.

## 7.5. ПИД регулятор PIDController

ПИД регулятор PIDController (далее PIDController) обеспечивает выполнение следующих функций:

- автоматическое регулирование параметров исполнительных механизмов по ПИД-закону;
- управление исполнительным механизмом в режиме MANUAL;
- автоматическое регулирование параметров исполнительных механизмов в режиме CASCADE совместно с другим ПИД-регулятором.

### 7.5.1. Атрибуты PIDController

PIDController имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 63. Дополнительные атрибуты PIDController**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN	REAL	чтение	Входное значение

Имя атрибута	Тип данных	Доступ	Описание атрибута
REM_SP	REAL	чтение	Входное значение, заменяющее уставку SetPoint в режиме CASCADE, ед. изм. №2
STB	BOOL	чтение	Стробирование
CASI	BOOL	чтение	Готовность к управлению вход
TRK_IN	BOOL	чтение	Режим отслеживания вход
TRK_VAL	REAL	чтение	Значение в режиме отслеживания
RDBK_IN	REAL	чтение	Обратная связь от исполнительного механизма
<b>Выходные атрибуты</b>			
SP	REAL	чтение/запись	Управляющее значение
OUT	REAL	чтение	Выход
ERR	REAL	чтение	Разница между управляющим значением и входным значением аналогового датчика, ед.изм. №1
MOUТ	REAL	чтение/запись	Ручное значение
INTEGRAL	REAL	чтение/запись	Накопленный интегральный компонент
PV	REAL	чтение	Выходное значение
CAS0	BOOL	чтение	Готовность к управлению выход
<b>Конфигурационные атрибуты</b>			
IN_SRC	STRING	чтение/запись	Имя функционального блока-источника данных
OUT_SL	REAL	чтение/запись	Нижний предел шкалы исполнительного механизма
OUT_SH	REAL	чтение/запись	Верхний предел шкалы исполнительного механизма

Имя атрибута	Тип данных	Доступ	Описание атрибута
SP_MIN	REAL	чтение/запись	Нижний предел допустимого диапазона управляющих значений, инженерные единицы
SP_MAX	REAL	чтение/запись	Верхний предел допустимого диапазона управляющих значений, инженерные единицы
INVERT	BOOL	чтение/запись	Разрешение инверсии
KP	REAL	чтение/запись	Коэффициент пропорционального вычисления выходного воздействия
TI	REAL	чтение/запись	Время интегрирования, с
TD	REAL	чтение/запись	Время дифференцирования, с <sup>-1</sup>
REM_SP_SRC	STRING	чтение/запись	Имя ПИД регулятора-источника для режима CASCADE
SP_TRK	BOOL	чтение/запись	Отслеживание выхода
SP_TARG	REAL	чтение/запись	Конечная уставка для безударного режима
SP_RAMP_PCT	REAL	чтение/запись	Приращение уставки в безударном режиме, %
SP_RAMP_EN	BOOL	чтение/запись	Включить безударный режим перехода на новую уставку
SP_INIT	REAL	чтение/запись	Стартовое значение уставки при загрузке блока
OUT_INIT	REAL	чтение/запись	Стартовое значение выхода при загрузке блока

Имя атрибута	Тип данных	Доступ	Описание атрибута
PV_DB	REAL	чтение/запись	Зона нечувствительности регулятора <sup>6</sup>
PV_DB_HYS	REAL	чтение/запись	Гистерезис зоны нечувствительности <sup>7</sup>
STB_SRC	STRING	чтение/запись	Источник стробирования
STB_EN	BOOL	чтение/запись	Включение стробирования
MAN_ON_START	BOOL	чтение/запись	Переключаться в ручной режим при перезагрузке контроллера
MAN_ON_BAD	BOOL	чтение/запись	Переключаться в ручной режим при статусе BAD
CASI_SRC	STRING	чтение/запись	Готовность к управлению источник
TRK_IN_SRC	STRING	чтение/запись	Режим отслеживания источник
TRK_VAL_SRC	STRING	чтение/запись	Значение в режиме отслеживания источник
RDBK_IN_SRC	STRING	чтение/запись	Обратная связь от исполнительного механизма источник
BAD_REM_SP	USINT	чтение/запись	Режим при плохом статусе удаленной уставки
<b>Диагностические атрибуты</b>			
RDBK	REAL	чтение	Обратная связь
TRK	BOOL	чтение	Режим отслеживания

### 7.5.2. Режимы PIDController

Режимы PIDController:

<sup>6</sup>  $\frac{Max\ Set\ Point\ Limit - Min\ Set\ Point\ Limit}{100\%} \times Process\ Value\ Deadband\ Percent$

<sup>7</sup>  $\frac{Max\ Set\ Point\ Limit - Min\ Set\ Point\ Limit}{100\%} \times PV\ Deadband\ Hys\ Percent$

1. AUTO ("Автоматический").

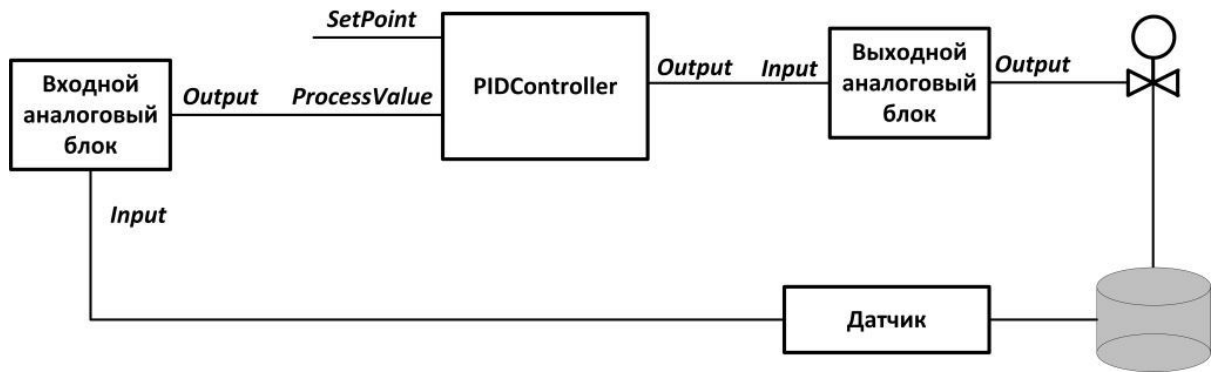


Рисунок 33. PIDController в режиме AUTO

В режиме входное значение записывается в атрибут *IN*.

PIDController обрабатывает входное значение в соответствии с заданными значениями конфигурационных атрибутов и уставкой *SP*.

Выходное управляющее значение записывается в атрибут *OUT*.

2. CASCADE ("Каскадный").

В данном режиме обеспечивается возможность работы PIDController в качестве ведомого по отношению к другому PIDController.

На рисунке приведена схема, в которой PIDController для расхода является ведомым по отношению к PIDController для уровня:

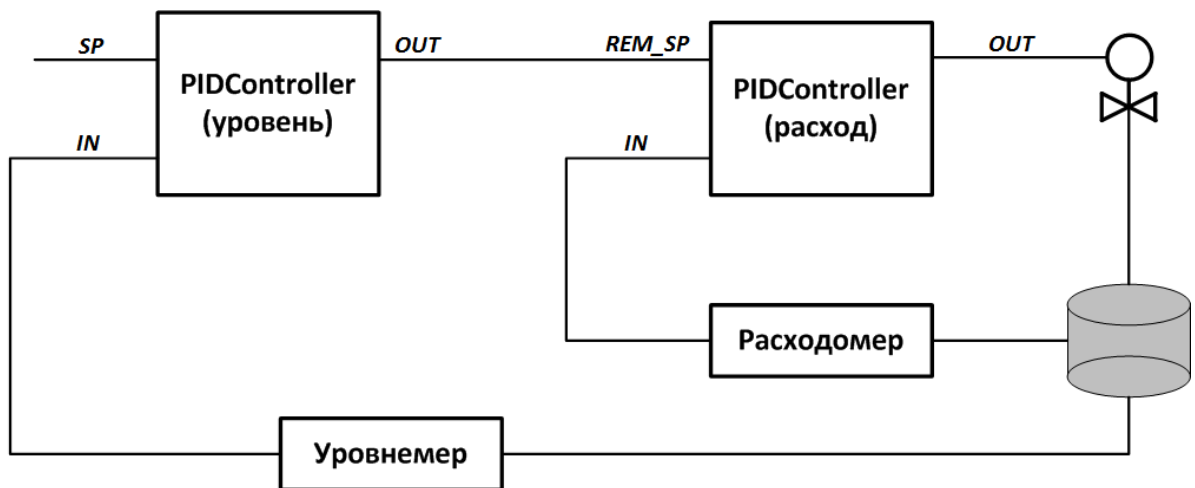


Рисунок 34. PIDController в режиме CASCADE

3. MANUAL ("Ручной").

В данном режиме производится ручное управление PIDController через запись ручного значения *MOUT* в атрибут *OUT*.

4. MASKING ("Маскирование") - алгоритм выполняется за исключением этапа формирования сигнализаций.

## 5. OFF ("Выключен").

### 7.5.3. Алгоритм PIDController

Выполнение алгоритма PIDController зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- регулирование по ПИД-закону;
- контроль выходного значения.

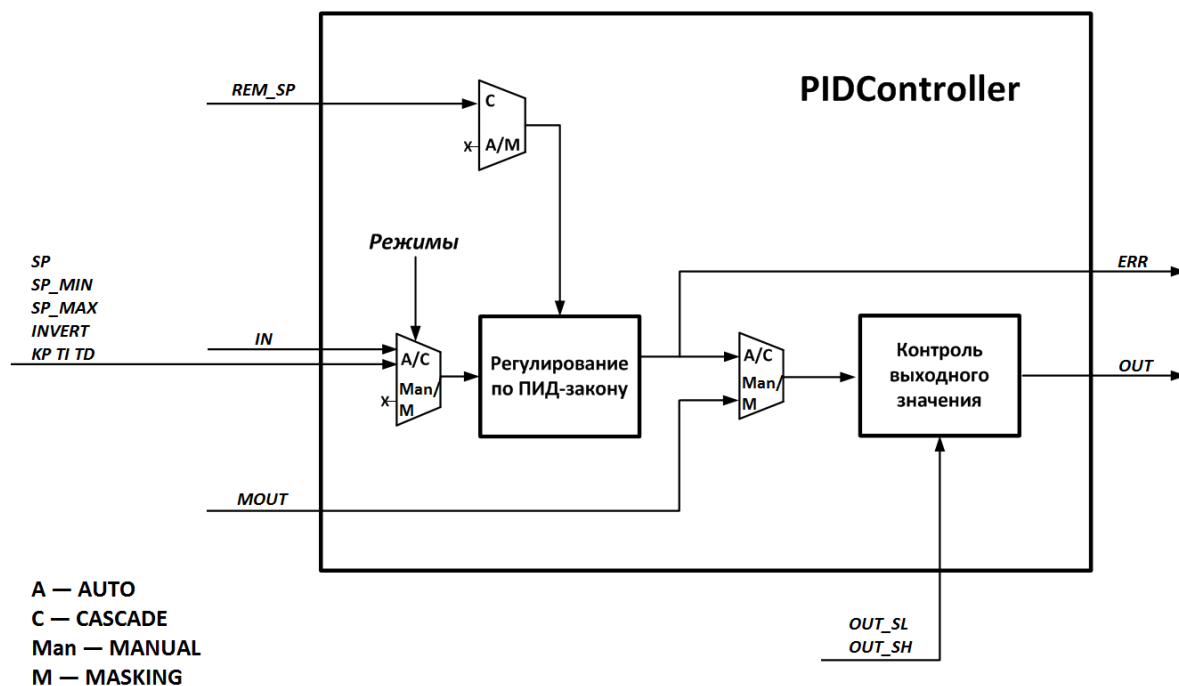


Рисунок 35. Схема процесса обработки данных PIDController

В режиме MANUAL при установке  $SP\_TRK = true$   $SP$  отслеживает  $IN$  и не дает себя менять вручную.

Если включен режим безударного перехода на новую уставку ( $SP\_RAMP\_EN = true$ ), то  $SP$  будет стремиться сравняться с  $SP\_TARG$ . В этом режиме атрибут  $SP$  можно изменить только через  $SP\_TARG$ .

При переключении в режим AUTO, если был включен  $SP\_RAMP\_EN$ ,  $SP$  будет двигаться до  $SP\_TARG$  в соответствии с заданным приращением  $SP\_RAMP\_PCT$  (процент от  $SP\_MAX - SP\_MIN$  добавляется или отнимается каждый рабочий цикл).

Если установлен флаг  $MAN\_ON\_START$  и ранее был установлен режим AUTO или CASCADE, то при старте системы функциональный блок автоматически переходит в режим MANUAL. При старте системы  $SP$  и  $OUT$  будут установлены



из своих стартовых значений  $SP\_INIT$  и  $OUT\_INIT$  соответственно для любого режима,  $MOUT$  будет приравнен к  $OUT$ .

Если блок установлен в режим MANUAL и старт прошел успешно, каждый цикл в  $OUT$  будет записываться  $MOUT$ .

Установка параметра  $PV\_DB$  позволяет осуществлять функции регулирования только в моменты, когда величина ошибки превышает данный порог нечувствительности. Установка  $PV\_DB = false$  отключает данный функционал. Величина зоны нечувствительности задается в процентах от входного диапазона  $SP$  (процент от  $SP\_MAX - SP\_MIN$ ). Для этого параметра можно установить  $PV\_DB\_HYS$ , который также отсчитывается в процентах диапазона  $SP$ .

Стробирование может быть включено в режиме AUTO. Для его настройки необходимо задать источник стробирования и установить флаг включения  $STB\_EN$ . Данная опция позволяет включать регулирование в определенные интервалы времени путем задания флага  $STB = true$ .

### 7.5.3.1. Регулирование по ПИД-закону

На данном этапе алгоритма производится обработка данных для последующей записи в выходные атрибуты.

Значение уставки  $SP$  должно соответствовать заданному диапазону  $SP\_MIN$  и  $SP\_MAX$ . В случае выхода из указанного диапазона значение  $SP$  приравнивается к ближайшему граничному значению.

Обработка данных производится в соответствии с ПИД-законом:

$$OUT = \frac{OUT\_SH - OUT\_SL}{2} + Inv \cdot (KP \cdot ERR + intsum_n + TD \cdot \Delta ERR)$$

где

$Inv$  – переменная инверсии;  $Inv = 1$ , если атрибут  $INVERT$  принимает значение false;  $Inv = -1$ , если значение  $INVERT$  true.

$$ERR = SP - IN;$$

$$\Delta ERR = ERR_n - ERR_{n-1};$$

$n$  – номер системного цикла;

$$intsum_n = \frac{\Delta t}{TI} \cdot ERR_n + intsum_{n-1} - \text{интегральная сумма};$$

$\Delta t$  – длительность системного цикла,  $\Delta t = 0.1$  с

### 7.5.3.2. Контроль выходного значения

Контроль выходного значения осуществляет проверку обработанного значения на соответствие границам диапазона *OUT\_SL* и *OUT\_SH* и запись в атрибут *OUT*.

При выходе обработанного значения из указанного диапазона происходит приравнивание этого значения к ближайшей границе диапазона.

## 7.6. Функциональный блок Motor

Функциональный блок Motor (далее Motor) предназначен для управления электроприводом.

Motor обеспечивает выполнение следующих функций:

- отправляет команды на запуск и останов электродвигателя;
- диагностирует состояние запуска электродвигателя;
- формирует сигнализации.

### 7.6.1. Атрибуты Motor

Motor имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 64. Дополнительные атрибуты Motor

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
ANS2	BOOL	чтение	Вход включен
IL	BOOL	чтение	Сигнал блокировки
CMD2_A	BOOL	чтение	Программная команда запуска в режимах AUTO и MASKING
CMD0_A	BOOL	чтение	Программная команда останова в режимах AUTO и MASKING
TRK	BOOL	чтение	Команда перевода в состояние Tracking
ANS0	BOOL	чтение	Вход выключен
CMD2_ESD	BOOL	чтение	Безопасный запуск
CMD0_ESD	BOOL	чтение	Безопасный останов
<i>Выходные атрибуты</i>			
CMD2	BOOL	чтение	Команда "Пуск"

Имя атрибута	Тип данных	Доступ	Описание атрибута
CMD0	BOOL	чтение	Команда "Стоп"
<i>Конфигурационные атрибуты</i>			
ANS2_SRC	STRING	чтение/ запись	Источник входа включен
IL_SRC	STRING	чтение/ запись	Источник блокировки
RST	BOOL	чтение/ запись	Команда "Деблокировать"
CMD2M	BOOL	чтение/ запись	Ручная команда запуска в режиме MANUAL
CMD0M	BOOL	чтение/ запись	Ручная команда останова в режиме MANUAL
CMD2_A_SRC	STRING	чтение/ запись	Источник программной команды запуска в режимах AUTO и MASKING
CMD0_A_SRC	STRING	чтение/ запись	Источник программной команды останова в режимах AUTO и MASKING
MTM2	DINT	чтение/ запись	Время на запуск, с
MTM0	DINT	чтение/ запись	Время на останов, с
TRK_SRC	STRING	чтение/ запись	Источник команды перевода в состояние Tracking
CTRL_TYPE	INT	чтение/ запись	Тип управления
PTM	INT	чтение/ запись	Длительность импульса
MAN_ON_ERR	BOOL	чтение/ запись	Переход в режим Manual при ошибке
ANS0_SRC	STRING	чтение/ запись	Источник входа включен
ESD_PRI	DINT	чтение/ запись	Приоритетная безопасная команда

Имя атрибута	Тип данных	Доступ	Описание атрибута
CMD2_ESD_SRC	STRING	чтение/ запись	Выполнение безопасного источника
CMD0_ESD_SRC	STRING	чтение/ запись	Останов безопасного источника
CTRL_TYPE_ESD	DINT	чтение/ запись	Тип управляющего сигнала при защите
<i>Диагностические атрибуты</i>			
STATE	INT	чтение	Состояние блока
MV	USINT	чтение	Требуемое состояние, код команды <sup>8</sup>
PV	USINT	чтение	Переменная <sup>9</sup>
<i>Сигнализационные атрибуты</i>			
FLT	BOOL	чтение	Сигнал "Заблокирован"
CMD2_ERR	BOOL	чтение	Сигнал "Не запустился"
CMD0_ERR	BOOL	чтение	Сигнал "Не остановился"
CMD_ERR	BOOL	чтение	Сигнал "Неверная программная команда"
CMD2_TRP	BOOL	чтение	Выполнение безопасной сигнализации
CMD0_TRP	BOOL	чтение	Останов безопасной сигнализации

<sup>8</sup> Атрибут *MV* - требуемое состояние, оно же код команды:

- 0 - выключить
- 2 - включить

<sup>9</sup> Атрибут *PV* - переменная процесса:

- 0 - выключено
- 2 - включено
- 4 - неопределенное состояние

### 7.6.2. Режимы Motor

Режимы Motor:

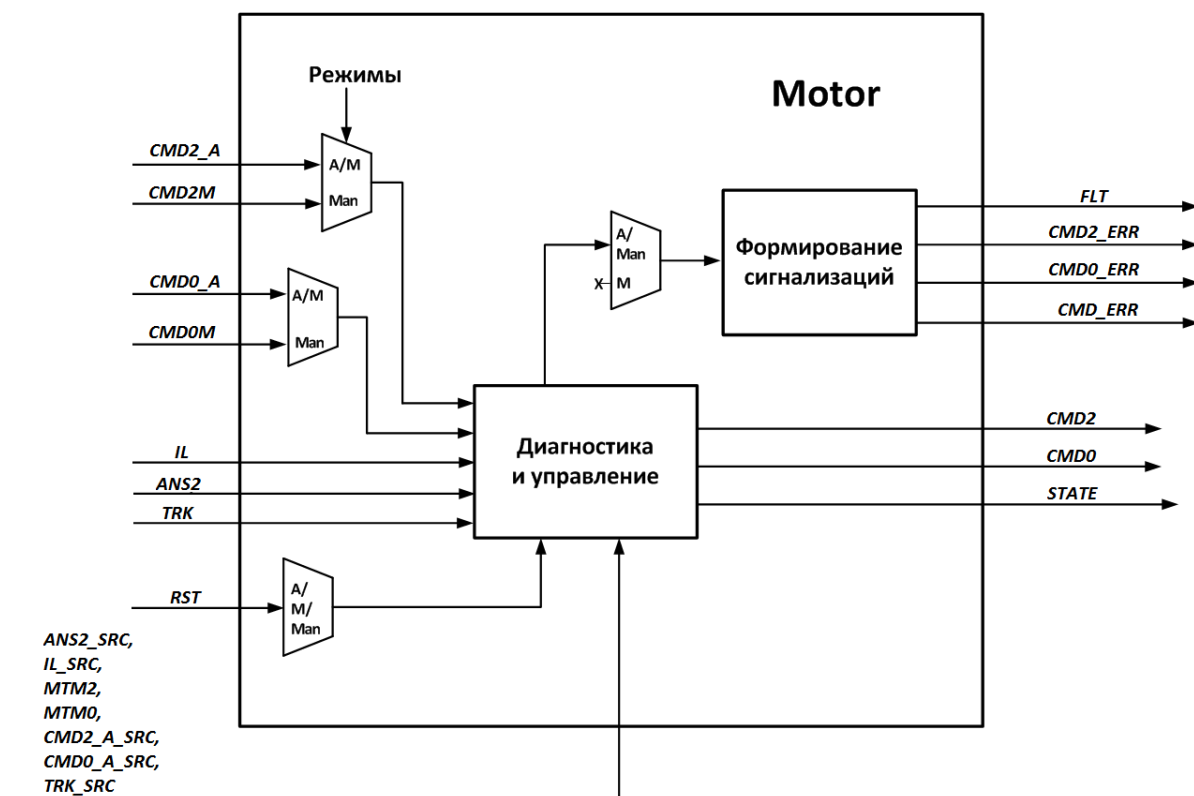
- OFF ("Выключен");
- AUTO ("Автоматический");
- MANUAL ("Ручной");
- MASKING ("Маскирование").

### 7.6.3. Алгоритм Motor

Алгоритм Motor зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- диагностика и управление;
- формирование сигнализаций.



A – AUTO  
 Man – MANUAL  
 M – MASKING

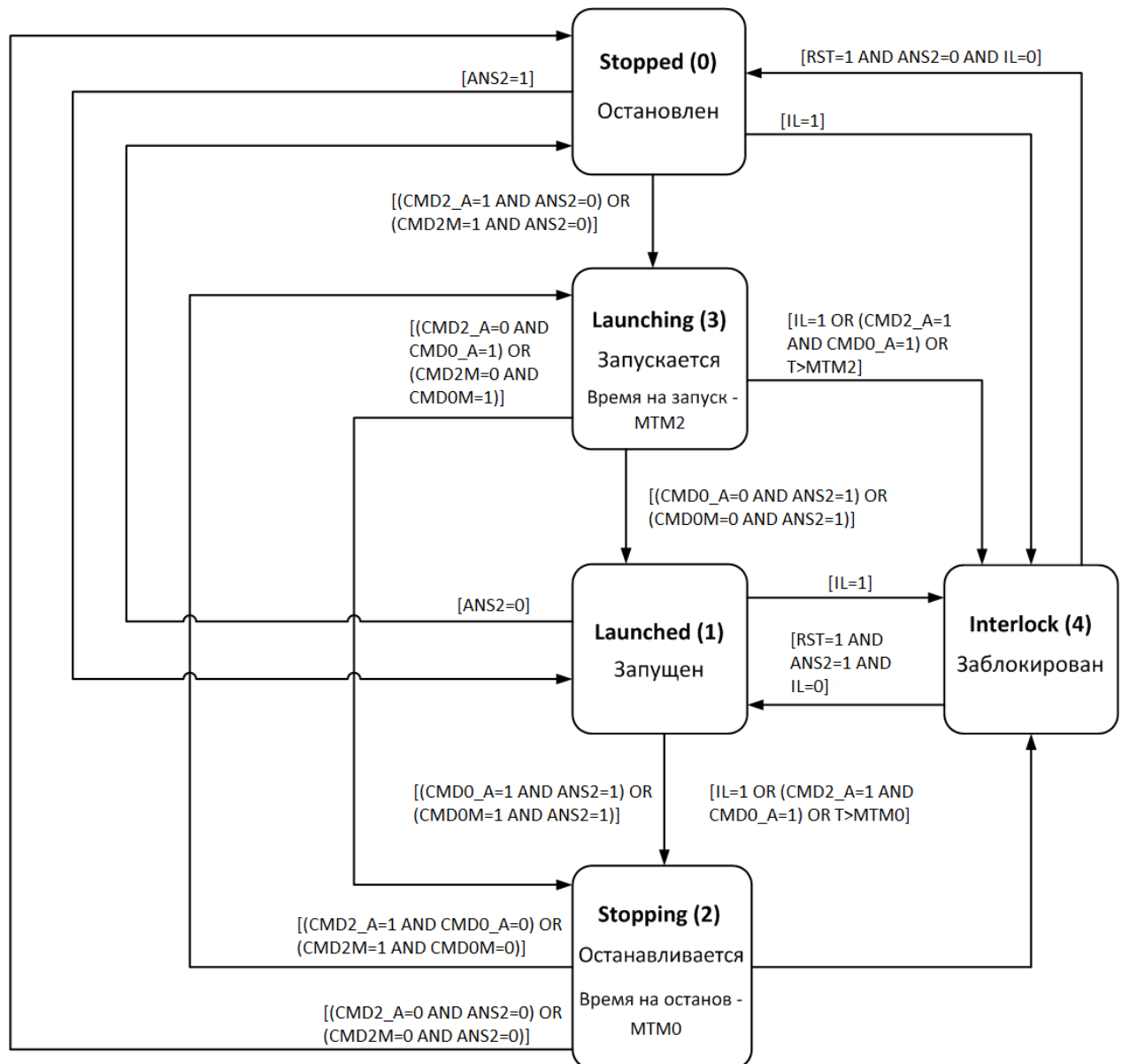
Рисунок 36. Процесс обработки данных Motor

#### 7.6.3.1. Диагностика и управление

Во время диагностики определяется состояние Motor. Состоянию присваивается значение и записывается в атрибут **STATE**.

Значения состояний:

- 0 - Stopped (остановлен);
- 1 - Launched (запущен);
- 2 - Stopping (останавливается);
- 3 - Launching (запускается);
- 4 - Interlock (заблокирован);
- 5 - Tracking (отслеживает).



**Рисунок 37. Диаграмма перехода состояний Motor**

В результате выполнения этапа диагностики и управления электродвигателю передается управляющая команда:

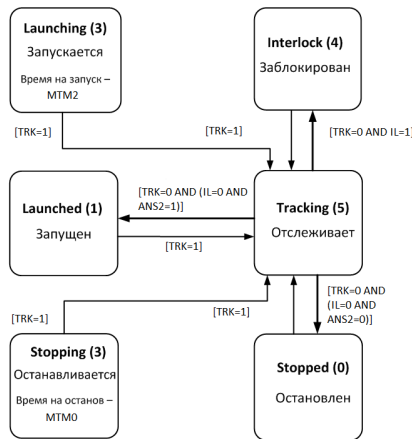
- для команды "Пуск"  $CMD2=1$ ;
- для команды "Стоп"  $CMD0=1$ .

Если TRK = 1, то блок из любого состояния переходит в состояние Tracking.

Если TRK = 0, то из состояния Tracking блок переходит в одно из трех состояний:

- Interlock;
- Launched;
- Stopped.

Критерии перехода представлены на диаграмме Рисунок 38. Диаграмма перехода из/в состояние Tracking.



**Рисунок 38. Диаграмма перехода из/в состояние Tracking**

### 7.6.3.2. Формирование сигнализаций

Сигнализации формируются в режимах AUTO и MANUAL.

В режиме MASKING сигнализации не формируются.

Сигнализационные атрибуты получают значение 1 (true) в случае наличия сигнализации.

**Таблица 65. Условия наличия сигнализации**

Атрибут	Обозначение	Условие
FLT	"Авария"	Входной атрибут IL показывает наличие блокировки: IL = 1
CMD2_ERR	"Не запустился"	Motor заблокирован при попытке запуска из-за превышения времени на запуск: T > MTM2

Атрибут	Обозначение	Условие
CMD0_ERR	"Не остановился"	Motor заблокирован при попытке останова из-за превышения времени на останов: $T > MTM0$
CMD_ERR	"Неверная программная команда"	Подаются одновременные команды на различные действия: $CMD2\_A = 1 \text{ AND } CMD0\_A = 1$

## 7.7. Функциональный блок Program

Функциональный блок Program (далее Program) запускает программу на языке Calculon<sup>10</sup>.

Программа позволяет реализовать алгоритмы, не входящие в набор стандартных функциональных блоков.

### 7.7.1. Атрибуты Program

Program имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 66. Дополнительные атрибуты Program

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i><b>Входные атрибуты</b></i>			
BI1...BI8	BOOL	чтение	Входы BI1...BI8 (Boolean input)
RI1...RI4	REAL	чтение	Входы RI1...RI4 (Real input)
II1...II2	INT	чтение	Входы II1...II2 (Integer input)
<i><b>Выходные атрибуты</b></i>			
BO1...BO4	BOOL	чтение	Выходы BO1...BO4
RO1...RO2	REAL	чтение	Выходы RO1...RO2
IO1	INT	чтение	Выход IO1
<i><b>Конфигурационные атрибуты</b></i>			

<sup>10</sup> "Справочное руководство по языку Calculon".



Имя атрибута	Тип данных	Доступ	Описание атрибута
Text	STRING	чтение/ запись	Текст программы
SourceVI1...SourceVI8	BOOL	чтение/ запись	Источник VI1...Источник VI8
SourceRI1...SourceRI4	STRING	чтение/ запись	Источник RI1...Источник RI4
SourceII1...SourceII2	STRING	чтение/ запись	Источник II1...Источник II2

Некоторые значения атрибута *Status* для Program имеют дополнительное описание:

**Таблица 67. Дополнительное описание значений атрибута *Status* Program**

Имя	Значение	Описание
ProgramError	0x02050000	Ошибка в программе Calculon

### 7.7.2. Режимы Program

Режимы Program:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.7.3. Алгоритм блока Program

Выполнение алгоритма Program зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- чтение;
- интерпретация и запись;
- диагностика.



**Рисунок 39. Процесс обработки данных Program**

### 7.7.3.1. Чтение

При чтении данные с выходов функциональных блоков контроллера записываются во входные атрибуты.

Функциональные блоки привязываются к Program по номерованным атрибутам источника.

Тип входного атрибута и атрибута источника должен соответствовать типу считываемых данных (см. Таблица 66. Дополнительные атрибуты Program, Рисунок 39. Процесс обработки данных Program).

### 7.7.3.2. Интерпретация и запись

На данном этапе интерпретатор языка Calculon получает входные значения и выполняет программу.

Текст программы записывается в конфигурационный атрибут Text. Максимальное количество символов в тексте программы 10000.

**Пример 1.** Можно добавлять комментарии следующего вида:

```
static a = 0;
static b = false;
static c = 0;

c=0;
a = a + 1;
//комментарий

if (a > 100) {
```

```
a = 0;

c = c + 1;
if (b) { /*КОММЕНТАРИЙ*/
    b = false;
} else {
    b = true;
```

**Примечание:** Пробелы и комментарии не учитываются в подсчёте символов.

Рассчитанные значения записываются в выходные атрибуты в зависимости от типа данных. В контроллер программа прогружается без пробелов и комментариев.

**Пример 2.** Обработка данных программой

```
x = RI1;    //считываются входные данные
x = x + 1;  //производится расчет
RO1 = x;    //обновляются выходные данные
```

### 7.7.3.3. Диагностика

В процессе диагностики осуществляется проверка:

- коммуникации Program с функциональным блоком-источником;
- синтаксиса программы.

## 7.8. Резервируемый дискретный функциональный блок RedundantDiscretePoint

Резервируемый дискретный функциональный блок RedundantDiscretePoint (далее RedundantDiscretePoint) обеспечивает выполнение следующих функций:

- резервирование канала приема данных;
- контроль и управление дискретными значениями датчика;
- фильтрация дребезга;
- формирование сигнализаций.

### 7.8.1. Атрибуты RedundantDiscretePoint

RedundantDiscretePoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 68. Дополнительные атрибуты RedundantDiscretePoint

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
IN1	BOOL	чтение	Входное значение от канала 1
IN2	BOOL	чтение	Входное значение от канала 2
DISC_CND_EN	BOOL	чтение	Если не имеет сорса, то == 1. Бит контекстного управления алармом со стороны алгоритмов
OUT	BOOL	чтение	Выходное значение
DISC_ACT	BOOL	чтение	Сработка аларма до разрешающего бита DISC_EN. Учитывает время фильтрации
DISC_NACT	BOOL	чтение	Инвертированный DISC_ACT, имеет статус OUT
<i>Конфигурационные атрибуты</i>			
IN1_SRC	STRING	чтение/ запись	Имя источника данных 1
IN2_SRC	STRING	чтение/ запись	Имя источника данных 2
IV	BOOL	чтение/ запись	Ручное значение в режимах IMITATION и MASK_IMITATION
INVERT	BOOL	чтение/ запись	Переключение на противоположное дискретное значение
ALM_EN	BOOL	чтение/ запись	Включение/выключение сигнализации
DISC_LIM	BOOL	чтение/ запись	Значение выходного атрибута, при котором формируется сигнализация
ALM_PRI	DINT	чтение/ запись	Приоритет сигнализации

Имя атрибута	Тип данных	Доступ	Описание атрибута
PV_FTIME	DINT	чтение/ запись	Уставка временного промежутка (в миллисекундах), в котором изменения входного атрибута игнорируются
FV	BOOL	чтение	Значение поля
DISC_CND_EN_SR	STRING	чтение/ запись	Включение источника входного сигнала
<i>Сигнализационные атрибуты</i>			
DISC_ALM	BOOL	чтение	Сигнализация
CH1_ERR	BOOL	чтение	Ошибка канала номер 1
CH2_ERR	BOOL	чтение	Ошибка канала номер 2

### 7.8.2. Режимы RedundantDiscretePoint

Режимы RedundantDiscretePoint:

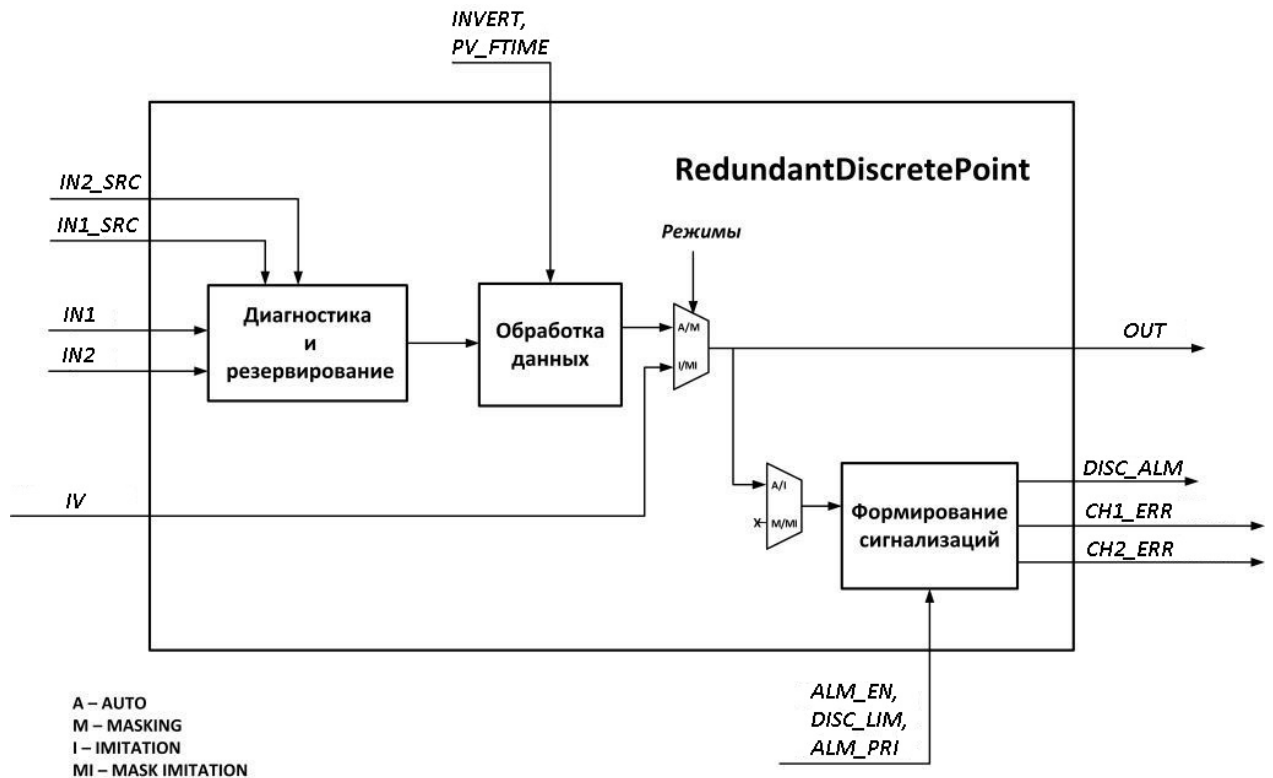
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 7.8.3. Алгоритм RedundantDiscretePoint

Выполнение алгоритма RedundantDiscretePoint зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- диагностика и резервирование;
- обработка данных;
- формирование сигнализаций.



**Рисунок 40. Процесс обработки данных RedundantDiscretePoint**

### 7.8.3.1. Диагностика и резервирование

На данном этапе производится диагностика и резервирование каналов приема данных.

При этом учитываются следующие правила:

- Для приема данных используется канал номер 1.
- Если канал номер 1 не доступен, используется канал номер 2.
- Если оба канала не доступны, формируется статус **BadInput**. Формируется сигнализационный флаг об ошибке **Bad**.
- Если оба канала доступны, но выдают разные значения в течение трех рабочих циклов, то формируется статус **BadInput**. Формируется сигнализационный флаг об ошибке **Bad**.

### 7.8.3.2. Обработка данных

Обработка данных включает следующие этапы:

- Инверсия. При инверсии входной атрибут переключается на противоположное значение. Для того чтобы включить функцию инверсии, задайте атрибуту **INVERT** значение 1 (true).
- Фильтрация дребезга. Учитывается возможный дребезг контактов датчиков.

Фильтрация производится, когда входной атрибут меняет значение. Уставка **PV\_FTIME** сравнивается с периодом, в котором данное значение остается

неизменным. Если указанный период не превышает уставку, изменение значения игнорируется.

- Запись данных в выходной атрибут **OUT**.

В режиме MANUAL данные в **OUT** записываются из атрибута **IV**.

### 7.8.3.3. Формирование сигнализаций

При формировании сигнализаций атрибуты **DISC\_ALM**, **CH1\_ERR**, **CH2\_ERR** принимают значение 1 (true).

Сигнализации формируются при одновременном выполнении условий:

- функция формирования сигнализаций активирована. Для активации задайте атрибуту **ALM\_EN** значение 1 (true).
- Выходное значение атрибута **OUT** совпадает с заданным значением атрибута **DISC\_LIM**.

В режимах MASKING и MASK\_IMITATION сигнализации не формируются.

## 7.9. Резервируемый входной аналоговый функциональный блок RedundantAnalogInputPoint

Резервируемый входной аналоговый функциональный блок RedundantAnalogInputPoint (далее RedundantAnalogInputPoint) обеспечивает выполнение следующих функций:

- резервирование канала приема данных;
- считывание и контроль достоверности данных аналогового датчика;
- преобразование данных из единиц датчика в инженерные единицы;
- сравнение выходного значения с величиной предупредительных и аварийных уставок;
- формирование сигнализаций.

### 7.9.1. Атрибуты RedundantAnalogInputPoint

RedundantAnalogInputPoint имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 69. Дополнительные атрибуты RedundantAnalogInputPoint**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
IN1	REAL	чтение	Входное значение, полученное от канала 1 в процентах

Имя атрибута	Тип данных	Доступ	Описание атрибута
IN2	REAL	чтение	Входное значение, полученное от канала 2 в процентах
HH_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1
HI_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1
LO_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1
LL_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1
PV	REAL	чтение	Выходное значение
HH_ACT	BOOL	чтение	Сработка аларма до разрешающего бита HH_EN. Учитывает гистерезис
HI_ACT	BOOL	чтение	Сработка аларма до разрешающего бита HI_EN. Учитывает гистерезис
LO_ACT	BOOL	чтение	Сработка аларма до разрешающего бита LO_EN. Учитывает гистерезис
LL_ACT	BOOL	чтение	Сработка аларма до разрешающего бита LL_EN. Учитывает гистерезис
HH_NACT	BOOL	чтение	Инвертированный HH_ACT, имеет статус PV
HI_NACT	BOOL	чтение	Инвертированный HI_ACT, имеет статус PV
LO_NACT	BOOL	чтение	Инвертированный LO_ACT, имеет статус PV
LL_NACT	BOOL	чтение	Инвертированный LL_ACT, имеет статус PV



Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Конфигурационные атрибуты</b>			
IN1_SRC	STRING	чтение/ запись	Имя источника данных 1
IN2_SRC	STRING	чтение/ запись	Имя источника данных 2
IV	REAL	чтение/ запись	Ручное значение, вводимое в режимах имитация и имитация маскирования
IN_SL	REAL	чтение/ запись	Нижняя граница шкалы датчика в процентах
IN_SH	REAL	чтение/ запись	Верхняя граница шкалы датчика в процентах
PV_SL	REAL	чтение/ запись	Нижняя граница шкалы датчика в инженерных единицах
PV_SH	REAL	чтение/ запись	Верхняя граница шкалы датчика в инженерных единицах
HH_LIM	REAL	чтение/ запись	Верхняя аварийная уставка в инженерных единицах
HI_LIM	REAL	чтение/ запись	Верхняя предупредительная уставка в инженерных единицах
LO_LIM	REAL	чтение/ запись	Нижняя предупредительная уставка в инженерных единицах
LL_LIM	REAL	чтение/ запись	Нижняя аварийная уставка в инженерных единицах
ALM_HYS	REAL	чтение/ запись	Уставка гистерезиса в процентах
IN_SLX	REAL	чтение/ запись	Нижнее расширение необработанного значения
IN_SHX	REAL	чтение/ запись	Верхнее расширение необработанного значения
HH_EN	BOOL	чтение/ запись	Включение/выключение сигнализации HiHiAlarm
HI_EN	BOOL	чтение/ запись	Включение/выключение сигнализации HiAlarm

Имя атрибута	Тип данных	Доступ	Описание атрибута
LO_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LoAlarm
LL_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LoLoAlarm
CONV_TYPE	DINT	чтение/ запись	Преобразование входного значения: Linear - линейное, SQRT - квадратичное
IN_CMP_DIS	BOOL	чтение/ запись	Отключение диагностики входных сигналов
FV	REAL	чтение	Значение поля
HH_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала HH
HI_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала HI
LO_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала LO
LL_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала LL
<b><i>Сигнализационные атрибуты</i></b>			
HH	BOOL	чтение	Переход выходного значения в диапазон верхних аварийных значений. Аларм HH == HH_ACT && HH_CND_EN && HH_EN
HI	BOOL	чтение	Переход выходного значения в диапазон верхних предупредительных значений. Аларм HI == HI_ACT && HI_CND_EN && HI_EN
LO	BOOL	чтение	Переход выходного значения в диапазон нижних предупредительных значений. Аларм LO == LO_ACT && LO_CND_EN && LO_EN

Имя атрибута	Тип данных	Доступ	Описание атрибута
LL	BOOL	чтение	Переход выходного значения в диапазон нижних аварийных значений. Аларм LL == LL_ACT && LL_CND_EN && LL_EN
CH1_ERR	BOOL	чтение	Ошибка канала 1
CH2_ERR	BOOL	чтение	Ошибка канала 2

### 7.9.2. Режимы RedundantAnalogInputPoint

Режимы RedundantAnalogInputPoint:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 7.9.3. Алгоритм RedundantAnalogInputPoint

Алгоритм RedundantAnalogInputPoint зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- диагностика и резервирование;
- масштабирование и запись данных;
- контроль ручного значения и запись данных;
- формирование сигнализаций.

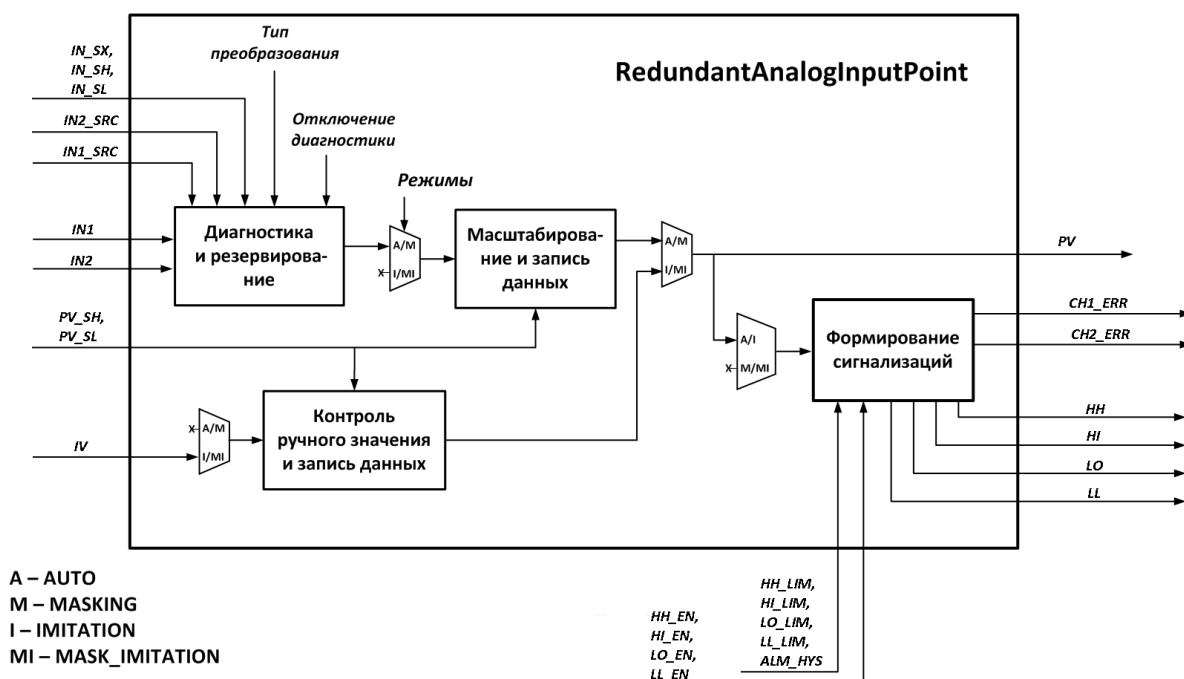


Рисунок 41. Процесс обработки данных RedundantAnalogInputPoint

### 7.9.3.1. Диагностика и резервирование

На данном этапе производится диагностика и резервирование каналов приема данных.

При этом учитываются следующие правила:

- Для приема данных используется канал 1.
- Если канал 1 не доступен, используется канал 2.
- Если оба канала не доступны, формируется статус **BadInput** и сигнализационный флаг об ошибке **Bad**.
- Если оба канала доступны, но выдают разные значения, то производится сравнение абсолютной разности значений с допустимым пределом 2 % от разности границ шкалы. Если абсолютная разность значений превышает допустимый предел в течение трех рабочих циклов, атрибут **Status** принимает значение 0x02030001 (**BadInput**) и выставляется сигнализационный флаг об ошибке **Bad**. Данное правило может быть отключено установкой флага в атрибуте **IN\_CMP\_DIS**.

Входное значение проверяется на соответствие диапазону шкалы.

Если входное значение выходит за пределы диапазона шкалы, то атрибут **Status** RedundantAnalogInputPoint принимает значение 0x02030001 (**BadInput**).

Если неверно задан диапазон шкалы (например, верхний предел меньше нижнего предела), то атрибут **Status** блока RedundantAnalogInputPoint примет значение 0x02030000 (**BadConfiguration**).

Расширение необработанного значения нижнего диапазона  $IN\_SLX$  и расширение необработанного значения верхнего диапазона входного значения  $IN\_SHX$  позволяют диагностировать неисправность датчика при отсутствии обрыва или короткого замыкания.

Различают типы преобразования входных сигналов ( $CONV\_TYPE$ ): Linear - линейное, SQRT - квадратичное. Линейное преобразование позволяет использовать текущее значение сигнала, квадратичное - квадратный корень, извлеченный из значения сигнала.

### 7.9.3.2. Масштабирование и запись данных

Данный этап выполняется в режимах AUTO и MASKING.

Входное значение переводится в инженерные единицы и записывается в атрибут  $PV$ .

Формула для перевода входного значения в инженерные единицы:

$$PV = \frac{PV\_SH - PV\_SL}{IN\_SH - IN\_SL} \cdot (IN - IN\_SL) + PV\_SL,$$

где  $IN\_SH$  и  $IN\_SL$  - верхняя и нижняя границы шкалы в процентах,

$PV\_SH$  и  $PV\_SL$  - верхняя и нижняя границы шкалы в инженерных единицах.

При квадратичном преобразовании входного значения (установлен флаг  $CONV\_TYPE$ ) масштабирование производится согласно формуле:

$$PV = \frac{PV\_SH - PV\_SL}{IN\_SH - IN\_SL} \cdot (\sqrt{IN} - IN\_SL) + PV\_SL$$

### 7.9.3.3. Контроль ручного значения

Данный этап выполняется в режимах MASKING и MASK\_IMITATION.

Значение атрибута  $IV$  сравнивается со шкалой датчика в инженерных единицах.

Если значение  $IV$  выходит за одну из границ шкалы, выходное значение будет приравнено к величине соответствующей границы.

### 7.9.3.4. Формирование сигнализаций

Этап формирования сигнализаций функционального блока описан в разделе 8.1. Формирование сигнализаций Библиотеки.

## 7.10. Функциональный блок Valve

Функциональный блок Valve (далее Valve) предназначен для управления электродвигателем задвижки.

Valve обеспечивает выполнение следующих функций:

- отправка команд на управление электродвигателем;
- диагностика состояния электродвигателя;
- формирование сигнализаций.

### 7.10.1. Атрибуты Valve

Valve имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 70. Дополнительные атрибуты Valve**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i><b>Входные атрибуты</b></i>			
ANS2	BOOL	чтение	Открыт
ANS0	BOOL	чтение	Закрыт
IL	BOOL	чтение	Сигнал блокировки
CMD2_A	BOOL	чтение	Программная команда открытия
CMD1_A	BOOL	чтение	Программная команда останова
CMD0_A	BOOL	чтение	Программная команда закрытия
TRK	BOOL	чтение	Команда перевода в состояние Tracking
CMD2_ESD	BOOL	чтение	Безопасный запуск
CMD0_ESD	BOOL	чтение	Безопасный останов
<i><b>Выходные атрибуты</b></i>			
CMD2	BOOL	чтение	Открыть
CMD1	BOOL	чтение	Остановить
CMD0	BOOL	чтение	Закрыть
<i><b>Конфигурационные атрибуты</b></i>			
ANS2_SRC	STRING	чтение/ запись	Источник состояния открыт

Имя атрибута	Тип данных	Доступ	Описание атрибута
ANS0_SRC	STRING	чтение/ запись	Источник состояния закрыт
IL_SRC	STRING	чтение/ запись	Источник блокировки
RST	BOOL	чтение/ запись	Деблокировать
CMD2M	BOOL	чтение/ запись	Ручная команда открытия
CMD1M	BOOL	чтение/ запись	Ручная команда останова
CMD0M	BOOL	чтение/ запись	Ручная команда закрытия
CMD2_A_SRC	STRING	чтение/ запись	Источник программной команды открытия
CMD1_A_SRC	STRING	чтение/ запись	Источник программной команды останова
CMD0_A_SRC	STRING	чтение/ запись	Источник программной команды закрытия
MTM2	DINT	чтение/ запись	Время на открытие, с
MTM0	DINT	чтение/ запись	Время на закрытие, с
TRK_SRC	STRING	чтение/ запись	Источник команды перевода в состояние Tracking
CTRL_TYPE	DINT	чтение/ запись	Тип управления
PTM	DINT	чтение/ запись	Длительность импульса
MAN_ON_ERR	BOOL	чтение/ запись	Переход в режим Manual при ошибке
CTM	DINT	чтение/ запись	Время ожидания начала перемещения из крайнего положения (Сход с концевика)

Имя атрибута	Тип данных	Доступ	Описание атрибута
ESD_PRI	DINT	чтение/ запись	Приоритетная безопасная команда
CMD2_ESD_SRC	STRING	чтение/ запись	Выполнение безопасного источника
CMD0_ESD_SRC	STRING	чтение/ запись	Останов безопасного источника
CTRL_TYPE_ESD	DINT	чтение/ запись	Тип управляющего сигнала при защите
<i>Диагностические атрибуты</i>			
STATE	DINT	чтение	Состояние блока
MV	USINT	чтение	Требуемое состояние, код команды <sup>11</sup>
PV	USINT	чтение	Переменная <sup>12</sup>
<i>Сигнализационные атрибуты</i>			
FLT	BOOL	чтение	Сигнал "Заблокирован"
CMD2_ERR	BOOL	чтение	Сигнал "Не открылся"
CMD1_ERR	BOOL	чтение	Сигнал "Не остановился"
CMD0_ERR	BOOL	чтение	Сигнал "Не закрылся"
CMD_ERR	BOOL	чтение	Сигнал "Неверная программная команда"
JAMMED_ERR	BOOL	чтение	Сигнал "Заклинило"

<sup>11</sup> Атрибут *MV* - требуемое состояние, оно же код команды:

- 0 - закрыть
- 1 - стоп
- 2 - открыть

<sup>12</sup> Атрибут *PV* - переменная процесса:

- 0 - закрыто
- 1 - промежуточное состояние
- 2 - открыто
- 4 - ошибка концевиков



Имя атрибута	Тип данных	Доступ	Описание атрибута
ANS2_ERR	BOOL	чтение	Превышено время ожидания начала перемещения из крайнего положения "Закрыто"
ANS0_ERR	BOOL	чтение	Превышено время ожидания начала перемещения из крайнего положения "Открыто"
CMD2_TRP	BOOL	чтение	Выполнение безопасной сигнализации
CMD0_TRP	BOOL	чтение	Останов безопасной сигнализации

### 7.10.2. Режимы Valve

Режимы Valve:

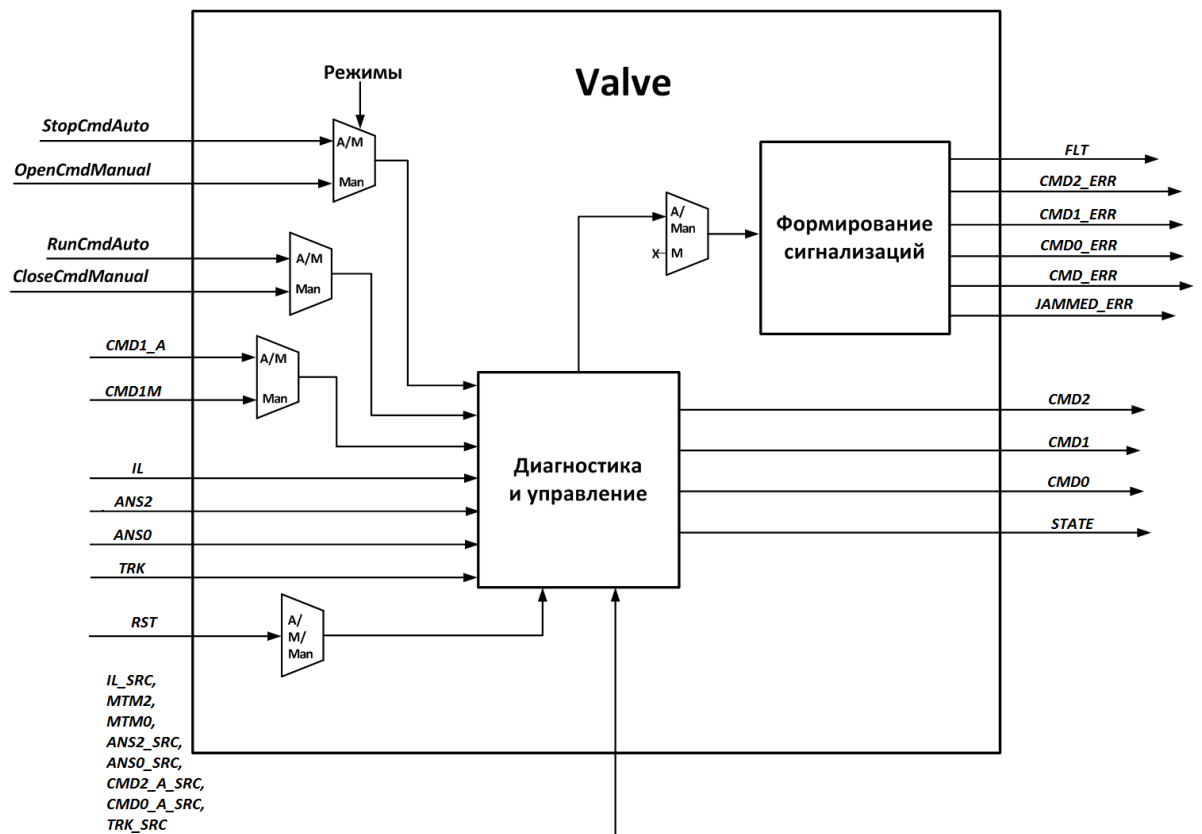
- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- MANUAL ("Ручной").

### 7.10.3. Алгоритм Valve

Алгоритм блока Valve зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- диагностика и управление;
- формирование сигнализаций.



A – AUTO  
 Man – MANUAL  
 M – MASKING

**Рисунок 42. Схема процесса обработки данных Valve**

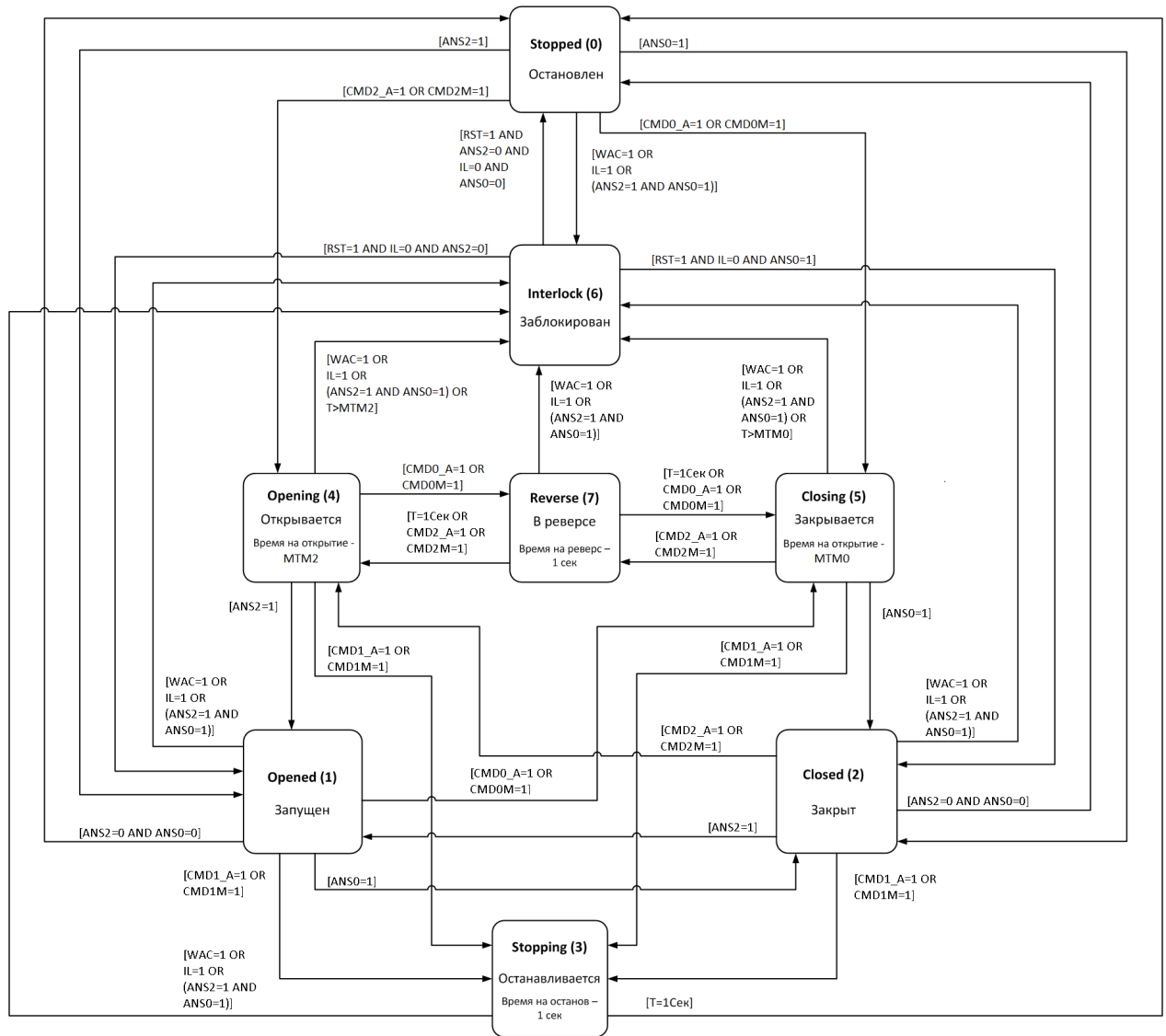
### 7.10.3.1. Диагностика и управление Valve

Во время диагностики определяется состояние блока Valve. Состоянию присваивается значение и записывается в атрибут **STATE**.

Значения состояний:

- 0 - Stopped (остановлен);
- 1 - Opened (открыт);
- 2 - Closed (закрыт);
- 3 - Stopping (останавливается);
- 4 - Opening (открывается);
- 5 - Closing (закрывается);
- 6 - Interlock (заблокирован);
- 7 - Reverse (в реверсе);
- 8 - Tracking (отслеживает).

Одно состояние переходит в другое при выполнении заданных условий.



**Рисунок 43. Диаграмма перехода состояний блока Valve**

*WAC* на диаграмме обозначает ошибку, при которой хотя бы двое из атрибутов *CMD2\_A*, *CMD1\_A* или *CMD0\_A* равны единице.

По результатам диагностики Valve передает управляющую команду:

CMD2 = 1 - "Открыть";

CMD1 = 1 - "Остановить";

CMD0 = 1 - "Закреть".

Если TRK = 1, то блок из любого состояния переходит в состояние Tracking.

Если TRK = 0, то из состояния Tracking блок переходит в одно из четырех состояний:

- Interlock;

- Opened;
- Closed;
- Stopped.

Критерии перехода представлены на диаграмме Рисунок 44. Диаграмма перехода из/в состояние Tracking.

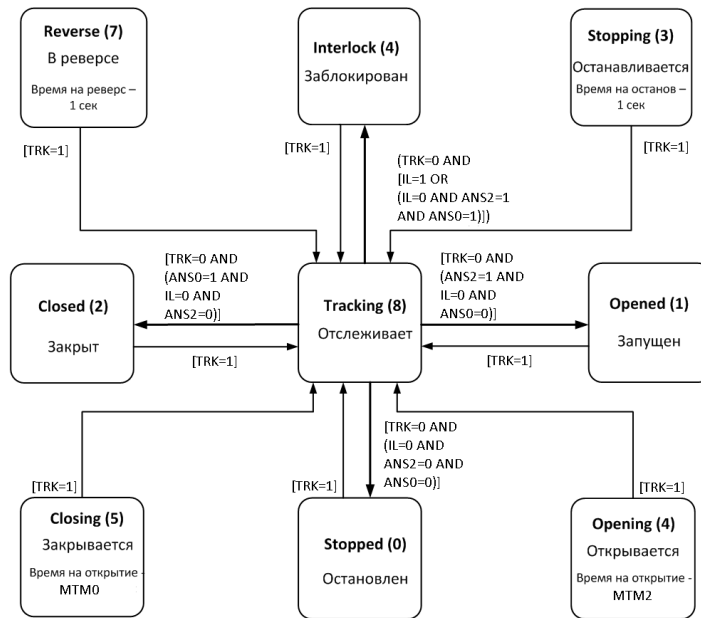


Рисунок 44. Диаграмма перехода из/в состояние Tracking

### 7.10.3.2. Формирование сигнализаций

Сигнализации формируются в режимах AUTO и MANUAL. Сигнализационные атрибуты получают значение 1 (true).

Таблица 71. Условия формирования сигнализации

Атрибут	Обозначение	Условие
IL	"Авария"	Входной атрибут FaultInput показывает наличие блокировки: IL = 1
CMD2_ERR	"Не открылся"	Valve заблокирован при попытке запуска из-за превышения времени на запуск: T > MTM2

Атрибут	Обозначение	Условие
CMD0_ERR	"Не закрылся"	Valve заблокирован при попытке останова из-за превышения времени на останов: $T > MTM0$
CMD_ERR	"Неверная программная команда"	Подаются одновременные команды на различные действия. Хотя бы двое из атрибутов CMD2_A, CMD1_A или CMD0_A равны единице
JAMMED_ERR	"Заклинило"	Входные атрибуты показывают одновременно открытое и закрытое состояние: $ANS2 = 1$ AND $ANS0 = 1$

В режиме MASKING сигнализации не формируются.

## 7.11. Функциональный блок Logical block

Функциональный блок Logical block (далее Logical block) обеспечивает выполнение логических операций OR, AND или XOR.

### 7.11.1. Атрибуты Logical block

Logical block имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 72. Дополнительные атрибуты Logical block**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/Выходные атрибуты</i>			
IN1...IN8	BOOL	чтение	Входное значение от каналов 1...8
OUT, NOUT	BOOL	чтение	Выходное значение: 0...1
<i>Конфигурационные атрибуты</i>			
IN1_SRC...IN8_SRC	STRING	чтение/запись	Имя источника данных 1...8

Имя атрибута	Тип данных	Доступ	Описание атрибута
FUNC	DINT	чтение/запись	Функция

### 7.11.2. Режимы Logical block

Режимы Logical block:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование").

### 7.11.3. Алгоритм Logical block

Алгоритм Logical block включает в себя следующие элементы:

- чтение;
- обработка данных;
- диагностика.

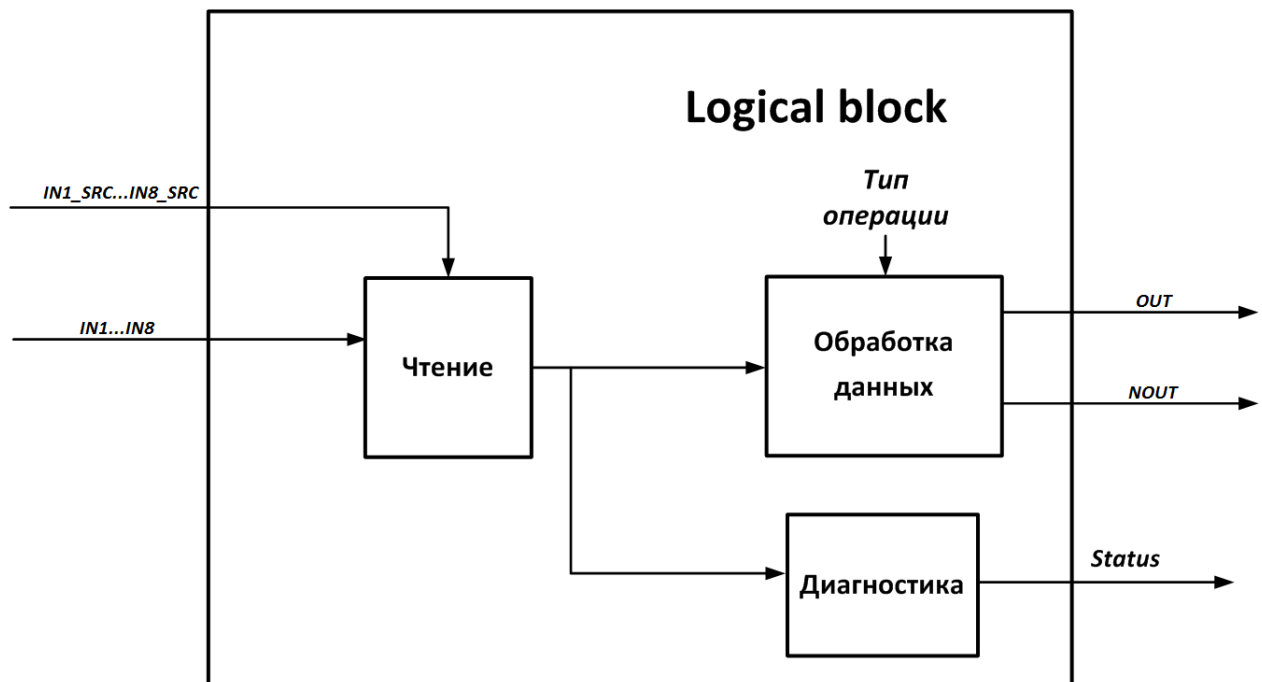


Рисунок 45. Процесс обработки данных Logical block

#### 7.11.3.1. Чтение

При чтении данные с выходов функциональных блоков-источников записываются во входные атрибуты.

### 7.11.3.2. Обработка данных

Обработка данных заключается в выполнении логических операций OR / AND / XOR над атрибутами входного значения *IN1...IN8* и запись в выходные атрибуты *OUT, NOUT*.

Обработка данных выполняется по следующим правилам:

- OR — это операция, действие которой эквивалентно применению логического «ИЛИ» к входным атрибутам *IN1...IN8*.
- AND — операция, действие которой эквивалентно применению логического «И» к входным атрибутам *IN1...IN8*.
- XOR — это операция, действие которой эквивалентно применению логического исключающего «ИЛИ», результат действия которой равен 1 (true), если число входных атрибутов в состоянии 1 (true) нечетно, и равен 0 (false), если четно.

В таблице продемонстрировано применение логических операций к входным атрибутам (в примере используются значения *IN1\_SRC, IN2\_SRC*).

**Таблица 73. Пример применения логических операций**

Входные атрибуты		Результат OR		Результат AND		Результат XOR	
IN1_SRC	IN2_SRC	OUT	NOUT	OUT	NOUT	OUT	NOUT
0	0	0	1	0	1	0	1
0	1	1	0	0	1	1	0
1	0	1	0	0	1	1	0
1	1	1	0	1	0	0	1

### 7.11.3.3. Диагностика

В процессе диагностики осуществляется проверка коммуникации входных атрибутов с соответствующими функциональными блоками-источниками.

При этом учитываются следующие правила<sup>13</sup>:

- Если источник *IN1\_SRC...IN8\_SRC* отсутствует, он исключается из обработки данных.
- Если все источники *IN1\_SRC...IN8\_SRC* отсутствуют, атрибут *Status* функционального блока Logical block принимает значение ResolvingReference - обработка данных не выполняется.

<sup>13</sup> Подробнее в документе "Концепция технологического ПО", п. 4.2. Атрибут Status, Таблица 6. Значения атрибута Status.

- Если один из источников *IN1\_SRC...IN8\_SRC* имеет статус отличный от **OK**, атрибут *Status* блока принимает значение статуса этого источника, и обработка данных в текущем рабочем цикле прекращается.

По результатам диагностики Logical block формирует значения выходных атрибутов *OUT*, *NOUT*.

## 7.12. Функциональный блок MinMax Block

Функциональный блок MinMax Block (далее MinMax Block) предназначен для математической и логической обработки входов. Блок поддерживает до 6 функций управления подключения входов.

### 7.12.1. Атрибуты MinMax Block

MinMax Block имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 74. Дополнительные атрибуты MinMax block**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i><b>Входные/Выходные атрибуты</b></i>			
Input1...Input8	REAL	чтение	Входное значение от каналов 1...8
Output	REAL	чтение	Выходное значение
Disable1...Disable8	BOOL	чтение	Входы отключения Input1...Input8
Selected	INT	чтение	Выход, указывающий выбранный вход или количество пригодных для использования входов
<i><b>Конфигурационные атрибуты</b></i>			
Source1...Source8	STRING	чтение/ запись	Имя источника данных 1...8



Имя атрибута	Тип данных	Доступ	Описание атрибута
FunctionType	USINT	чтение/ запись	Тип функции - значение аналогового входа: <ul style="list-style-type: none"> <li>• Минимум (min);</li> <li>• Максимум (max);</li> <li>• Средний (mid);</li> <li>• Первый Нормальный (firstgood);</li> <li>• Усредненный (avg);</li> <li>• Горячий Резерв (hotbackup).</li> </ul>
OpSelect	USINT	чтение/ запись	Если имеет ненулевое значение, выбирает соответствующий вход (т.е. 2 выбирает Input2), если значение нулевое, выходное значение Output определяется параметром FunctionType
DisableSource1... DisableSource8	STRING	чтение/ запись	Источник данных для входа в Disable 1...8
DisableNv1... DisableNv8	BOOL	чтение/ запись	Выключает входы Input1...Input8 при неподключенном источнике входа Disable
MinGood	USINT	чтение/ запись	Минимальное количество пригодных для использования входов, при котором результат не имеет статуса <i>Bad</i>

### 7.12.2. Режимы MinMax Block

Режимы MinMax Block:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.12.3. Алгоритм MinMax Block

Алгоритм MinMax Block включает в себя следующие элементы:

- чтение;
- обработка данных;
- диагностика.

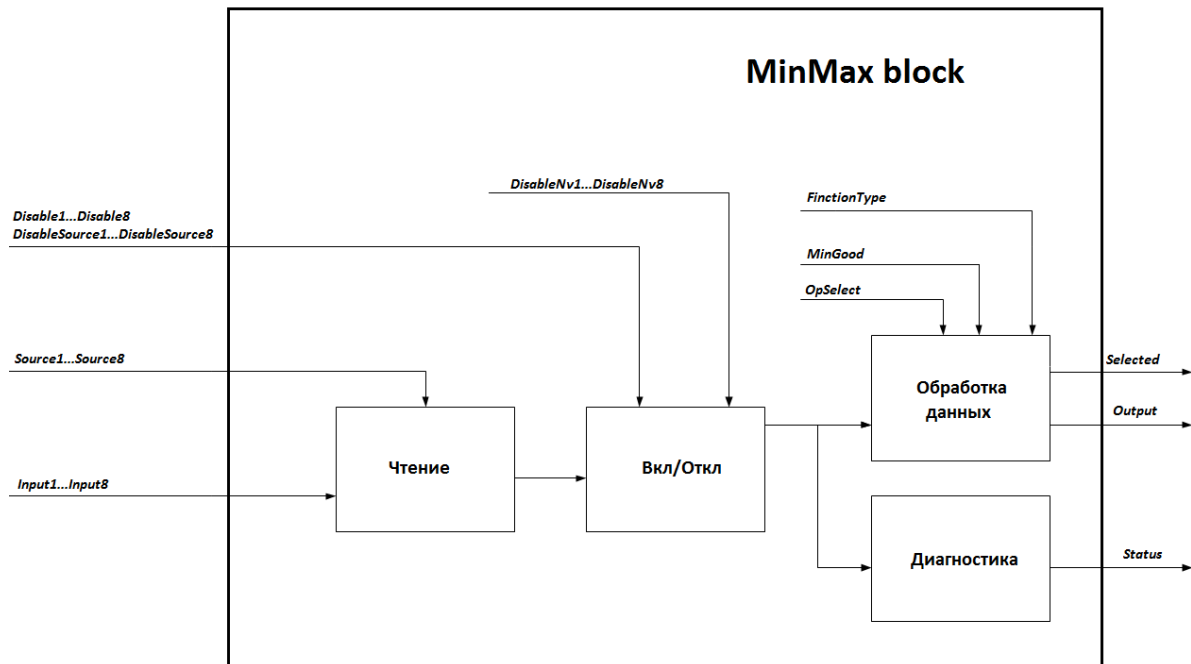


Рисунок 46. Процесс обработки данных MinMax block

#### 7.12.3.1. Чтение

При чтении данные с выходов функциональных блоков-источников записываются во входные атрибуты.

#### 7.12.3.2. Обработка данных

*OpSelect* – если имеет ненулевое значение, выбирает соответствующий вход (т.е. 2 выбирает *Input2*). Если значение равно 0, то вход выбирает *FunctionType*.

Тип функции - значение аналогового входа:

- Минимум (min) - выбирает вход с минимальным значением из тех входов, которые подключены к источникам, не выключены<sup>14</sup> и не имеют статус источника *Bad*.
- Максимум (max) - выбирает вход с максимальным значением из тех входов, которые подключены к источникам, не выключены и не имеют статус *Bad*;
- Средний (mid) - выбирает вход со средним значением из тех входов, которые подключены к источникам, не выключены и не имеют статус *Bad*;
- Первый Нормальный (firstgood) - выбирает первый вход, который подключен к источникам, не выключен и не имеет статуса *Bad*, начиная с Input1;
- Усредненный (avg) - усредняет те входы которые подключены к источникам, не выключены и не имеют статус *Bad*;
- Горячий Резерв (hotbackup) - обслуживает вход, выбранный на предыдущем цикле сканирования, если он пригоден для использования. Если нет, переходит к следующему пригодному входу, считая от номера входа, выбранного на предыдущем цикле сканирования.

***Disable1...8*** имея истинное значение, выключает соответствующий вход (т.е. ***Disable1*** выключает ***Input1***, ***Disable2*** выключает ***Input2***). Выключенный вход не используется алгоритмом, сконфигурированным в ***FunctionType***.

Если соответствующий вход ***Disable1...8*** не подключен к ***DisableSource1...8***, то его состояние берется из параметра ***DisableNv1...8*** соответственно, если подключен, то ***DisableNv1...8*** игнорируется и состояние берется из ***DisableSource1...8***.

Если включенных входов, имеющих статус ***Ok***, меньше ***MinGood***, то блок переходит в состояние ***Bad***.

### 7.12.3.3. Диагностика

В процессе диагностики осуществляется проверка коммуникации входных атрибутов с соответствующими функциональными блоками-источниками.

Во время диагностики определяется состояние блока MinMax Block.

---

<sup>14</sup> Выключенным считается вход, который:

- имеет подключенным соответствующий вход ***Disable***, источник которого в состоянии true (1);
- при неподключенном входе ***Disable***, вход ***Input*** может быть отключен соответствующим конфигурационным атрибутом ***DisabledNv***;
- вход ***Disable***, если он подключен к источнику, имеет приоритет над конфигурационным атрибутом ***DisabledNv***.

При этом учитываются следующие правила<sup>15</sup>:

- Если источник *Source1...Source8* отсутствует, он исключается из обработки данных.
- Если все источники *Source1...Source8* отсутствуют, атрибут *Status* функционального блока MinMax block принимает значение ResolvingReference - обработка данных не выполняется.
- Если один из источников *Source1...Source8* имеет статус отличный от ОК, атрибут *Status* блока принимает значение статуса этого источника, и обработка данных в текущем рабочем цикле прекращается.

## 7.13. Функциональный блок Accumulator Block

Функциональный блок Accumulator Block (далее AccumulatorBlock) предназначен для суммирования входного значения и записи итоговой суммы в выходной атрибут.

### 7.13.1. Атрибуты Accumulator Block

Таблица 75. Дополнительные атрибуты Accumulator Block

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/Выходные атрибуты</i>			
Input	REAL	чтение	Входное значение
Reset	BOOL	чтение	Сброс по импульсу
Output	REAL	чтение	Выходное значение
IntegerOutput	INT	чтение	Целочисленная часть выходного значения
FractionalOutput	REAL	чтение	Дробная часть выходного значения
<i>Конфигурационные атрибуты</i>			
InputSource	STRING	чтение/ запись	Имя источника входных данных
ResetSource	STRING	чтение/ запись	Имя источника данных для сброса по импульсу
EngineerUnit	STRING	чтение/ запись	Единица измерения

<sup>15</sup> Подробнее в документе "Концепция технологического ПО", п. 4.2. Атрибут Status, Таблица 6. Значения атрибута Status.

Имя атрибута	Тип данных	Доступ	Описание атрибута
TimePeriod	INT	чтение/ запись	Единица измерения времени
SumType	INT	чтение/ запись	Тип суммирования: Module - по модулю, OneSide - однонаправленное, Bidirectional - двунаправленное

### 7.13.2. Режимы Accumulator Block

Режимы Accumulator Block:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.13.3. Алгоритм Accumulator Block

Алгоритм Accumulator Block включает в себя следующие элементы:

- чтение;
- обработка данных;
- диагностика.

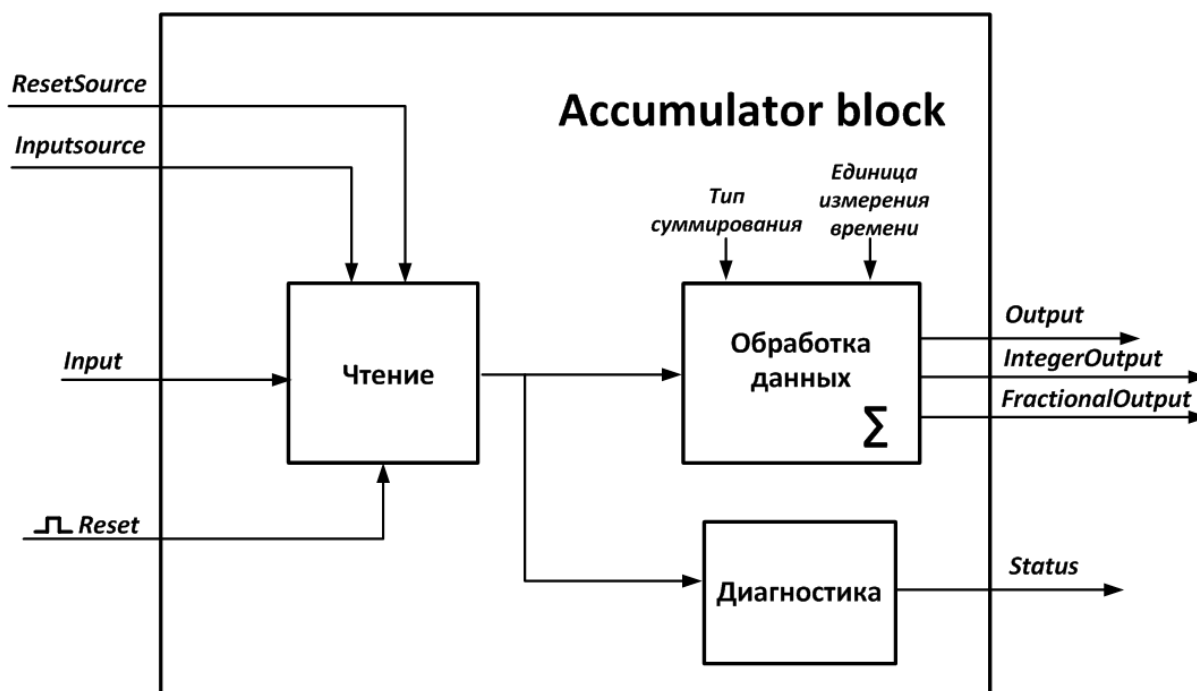


Рисунок 47. Процесс обработки данных Accumulator Block

### 7.13.3.1. Чтение

При чтении данные с выхода функционального блока-источника записываются во входной атрибут *Input*.

### 7.13.3.2. Обработка данных

Обработка данных заключается в суммировании по времени входного значения *Input* и запись итоговой суммы в выходные атрибуты *Output*, *IntegerOutput*, *FractionalOutput*.

Различают типы суммирования (*SumType*):

- *Module* - при суммировании по модулю учитывается абсолютная величина входного значения;
- *OneSide* - при однонаправленном суммировании отрицательное входное значение игнорируется блоком;
- *Bidirectional* - при двунаправленном суммировании учитываются и положительное, и отрицательное входное значение.

Существует возможность сброса выходных значений по импульсу на входе *Reset*, в результате которого отсчет суммы начинается заново.

### 7.13.3.3. Диагностика

В процессе диагностики осуществляется проверка коммуникации входных атрибутов с соответствующими функциональными блоками-источниками.

Во время диагностики определяется состояние блока Accumulator Block.

При этом если источник *InputSource* отсутствует, атрибут *Status* функционального блока принимает значение *ResolvingReference* - обработка данных не выполняется.<sup>16</sup>

## 7.14. Резервируемый ПИД регулятор RedundantPIDController

Резервируемый ПИД регулятор RedundantPIDController (далее RedundantPIDController) обеспечивает выполнение следующих функций:

- резервирование канала приема данных;
- считывание и контроль достоверности данных аналогового датчика;
- преобразование данных из процентов в инженерные единицы;
- сравнение выходного значения с величиной предупредительных и аварийных уставок;

---

<sup>16</sup> Подробнее в документе "Концепция технологического ПО", п. 4.2. Атрибут Status, Таблица 6. Значения атрибута Status.

- формирование сигнализаций;
- формирование управляющего сигнала на выходе по ПИД-закону;
- управление исполнительным механизмом в режиме MANUAL;
- формирование управляющего сигнала на выходе в режиме CASCADE с изменяющейся уставкой.

### 7.14.1. Атрибуты RedundantPIDController

RedundantPIDController имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 76. Дополнительные атрибуты RedundantPIDController**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
IN1	REAL	чтение	Входное значение, полученное от канала 1, ед.изм. №1
IN2	REAL	чтение	Входное значение, полученное от канала 2
REM_SP	REAL	чтение	Входное значение, заменяющее уставку SP в режиме CASCADE, ед. изм. №2
STB	BOOL	чтение	Стробирование
CASI	BOOL	чтение	Готовность к управлению вход
TRK_IN	BOOL	чтение	Режим отслеживания вход
TRK_VAL	REAL	чтение	Значение в режиме отслеживания
RDBK_IN	REAL	чтение	Обратная связь от исполнительного механизма
HH_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1
HI_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1
LO_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1

Имя атрибута	Тип данных	Доступ	Описание атрибута
LL_CND_EN	BOOL	чтение	Бит контекстного управления со стороны алгоритмов. Если не имеет сорса, то == 1
OUT	REAL	чтение	Выходное значение
ERR	REAL	чтение	Разница между управляющим значением и входным значением аналогового датчика, ед.изм. №1
SP	REAL	чтение/ запись	Управляющее значение, инженерные единицы
MOUT	REAL	чтение/ запись	Ручное значение в режиме MANUAL, инженерные единицы
HH_ACT	BOOL	чтение	Сработка аларма до разрешающего бита HH_EN. Учитывает гистерезис
HI_ACT	BOOL	чтение	Сработка аларма до разрешающего бита HI_EN. Учитывает гистерезис
LO_ACT	BOOL	чтение	Сработка аларма до разрешающего бита LO_EN. Учитывает гистерезис
LL_ACT	BOOL	чтение	Сработка аларма до разрешающего бита LL_EN. Учитывает гистерезис
HH_NACT	BOOL	чтение	Инвертированный HH_ACT, имеет статус PV
HI_NACT	BOOL	чтение	Инвертированный HI_ACT, имеет статус PV
LO_NACT	BOOL	чтение	Инвертированный LO_ACT, имеет статус PV
LL_NACT	BOOL	чтение	Инвертированный LL_ACT, имеет статус PV
INTEGRAL	REAL	чтение/зап	Накопленный интегральный компонент
PV	REAL	чтение	Выходное значение
CAS0	BOOL	чтение	Готовность к управлению выход



Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Конфигурационные атрибуты</b>			
IN1_SRC	STRING	чтение/ запись	Имя источника данных 1
IN2_SRC	STRING	чтение/ запись	Имя источника данных 2
IN_SL	REAL	чтение/ запись	Нижняя граница шкалы, %
IN_SH	REAL	чтение/ запись	Верхняя граница шкалы, %
PV_SL	REAL	чтение/ запись	Нижняя граница шкалы, инженерные единицы
PV_SH	REAL	чтение/ запись	Верхняя граница шкалы, инженерные единицы
HH_LIM	REAL	чтение/ запись	Верхняя аварийная уставка, инженерные единицы
HI_LIM	REAL	чтение/ запись	Верхняя предупредительная уставка, инженерные единицы
LO_LIM	REAL	чтение/ запись	Нижняя предупредительная уставка, инженерные единицы
LL_LIM	REAL	чтение/ запись	Нижняя аварийная уставка, инженерные единицы
ALM_HYS	REAL	чтение/ запись	Уставка гистерезиса, %
IN_SHX	REAL	чтение/ запись	Верхнее расширение необработанного значения
IN_SLX	REAL	чтение/ запись	Нижнее расширение необработанного значения
HH_EN	BOOL	чтение/ запись	Включение/выключение сигнализации HiHiAlarm
HI_EN	BOOL	чтение/ запись	Включение/выключение сигнализации HiAlarm
LO_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LoAlarm

Имя атрибута	Тип данных	Доступ	Описание атрибута
LL_EN	BOOL	чтение/ запись	Включение/выключение сигнализации LoLoAlarm
OUT_SH	REAL	чтение/ запись	Верхний предел шкалы исполнительного механизма, инженерные единицы
OUT_SL	REAL	чтение/ запись	Нижний предел шкалы исполнительного механизма, инженерные единицы
SP_MIN	REAL	чтение/ запись	Нижний предел допустимого диапазона управляющих значений, инженерные единицы
SP_MAX	REAL	чтение/ запись	Верхний предел допустимого диапазона управляющих значений, инженерные единицы
INVERT	BOOL	чтение/ запись	Разрешение инверсии
KP	REAL	чтение/ запись	Коэффициент пропорционального вычисления выходного воздействия
TI	REAL	чтение/ запись	Время интегрирования, с
TD	REAL	чтение/ запись	Время дифференцирования, с <sup>-1</sup>
REM_SP_SRC	STRING	чтение/ запись	Имя ПИД регулятора-источника для режима CASCADE
SP_TRK	BOOL	чтение/ запись	Отслеживание выхода
SP_TARG	REAL	чтение/ запись	Конечная уставка для безударного режима
SP_RAMP_PCT	REAL	чтение/ запись	Приращение уставки в безударном режиме, %
SP_RAMP_EN	BOOL	чтение/ запись	Включить безударный режим перехода на новую уставку
SP_INIT	REAL	чтение/ запись	Стартовое значение уставки при загрузке блока

Имя атрибута	Тип данных	Доступ	Описание атрибута
OUT_INIT	REAL	чтение/ запись	Стартовое значение выхода при загрузке блока
PV_DB	REAL	чтение/ запись	Зона нечувствительности регулятора <sup>17</sup>
PV_DB_HYS	REAL	чтение/ запись	Гистерезис зоны нечувствительности <sup>18</sup>
STB_SRC	STRING	чтение/ запись	Источник стробирования
STB_EN	BOOL	чтение/ запись	Включение стробирования
MAN_ON_START	BOOL	чтение/ запись	Переключаться в ручной режим при перезагрузке контроллера
CONV_TYPE	INT	чтение/ запись	Преобразование входного значения: Linear - линейное, SQRT - квадратичное
IN_CMP_DIS	BOOL	чтение/ запись	Отключение диагностики входных сигналов
HH_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала HH
HI_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала HI
LO_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала LO
LL_CND_EN_SRC	STRING	чтение/ запись	Включение источника входного сигнала LL
MAN_ON_BAD	BOOL	чтение/зап	Переключаться в ручной режим при статусе BAD
CASI_SRC	STRING	чтение/зап	Готовность к управлению источник

<sup>17</sup> 
$$\frac{\text{Max Set Point Limit} - \text{Min Set Point Limit}}{100\%} \times \text{Process Value Deadband Percent}$$

<sup>18</sup> 
$$\frac{\text{Max Set Point Limit} - \text{Min Set Point Limit}}{100\%} \times \text{PV Deadband Hys Percent}$$

Имя атрибута	Тип данных	Доступ	Описание атрибута
TRK_IN_SRC	STRING	чтение/зап	Режим отслеживания источник
TRK_VAL_SRC	STRING	чтение/зап	Значение в режиме отслеживания источник
RDBK_IN_SRC	STRING	чтение/зап	Обратная связь от исполнительного механизма источник
BAD_REM_SP	USINT	чтение/зап	Режим при плохом статусе удаленной уставки
<b>Сигнализационные атрибуты</b>			
HH	BOOL	чтение	Сигнал ВАУ - выходное значение перешло в диапазон верхних аварийных значений. Аларм HH == HH_ACT && HH_CND_EN && HH_EN
HI	BOOL	чтение	Сигнал ВПУ - выходное значение перешло в диапазон верхних предупредительных значений. Аларм HI == HI_ACT && HI_CND_EN && HI_EN
LO	BOOL	чтение	Сигнал НПУ - выходное значение перешло в диапазон нижних предупредительных значений. Аларм LO == LO_ACT && LO_CND_EN && LO_EN
LL	BOOL	чтение	Сигнал НАУ - выходное значение перешло в диапазон нижних аварийных значений. Аларм LL == LL_ACT && LL_CND_EN && LL_EN
CH1_ERR	BOOL	чтение	Ошибка канала 1
CH2_ERR	BOOL	чтение	Ошибка канала 2
<b>Диагностические атрибуты</b>			
RDBK	REAL	чтение	Обратная связь
TRK	BOOL	чтение	Режим отслеживания

### 7.14.2. Режимы RedundantPIDController

Режимы RedundantPIDController:

1. OFF ("Выключен") - алгоритм не выполняется.
2. AUTO ("Автоматический"). В режиме AUTO входное значение записывается в атрибуты *Input1* и *Input2*. RedundantPIDController обрабатывает входное значение в соответствии с заданными значениями конфигурационных атрибутов и уставкой *Set Point*. Выходное управляющее значение записывается в атрибут *Output*.

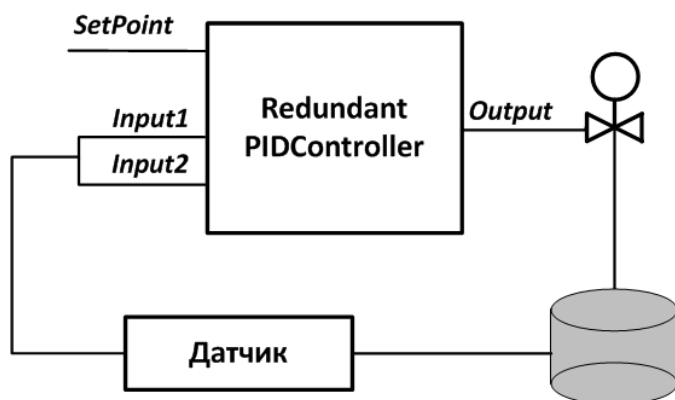


Рисунок 48. RedundantPIDController в режиме AUTO с блокировкой схемы резервирования

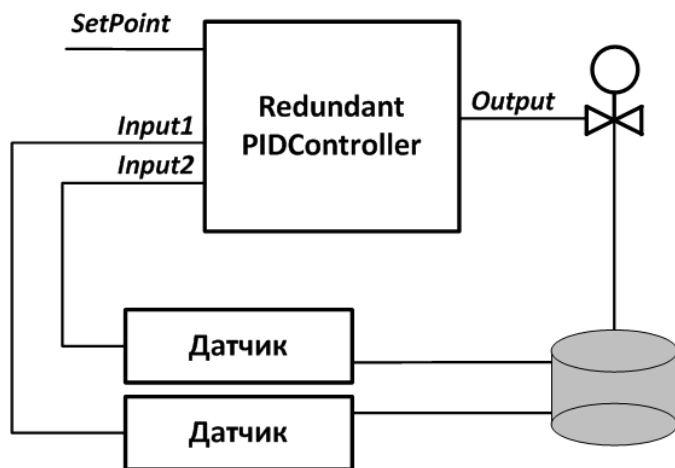
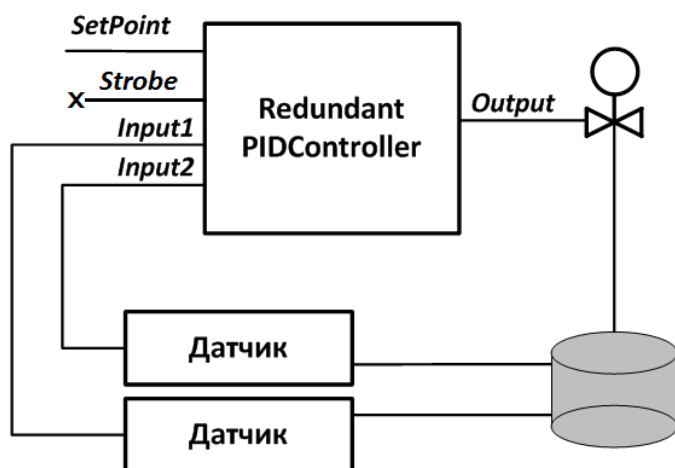


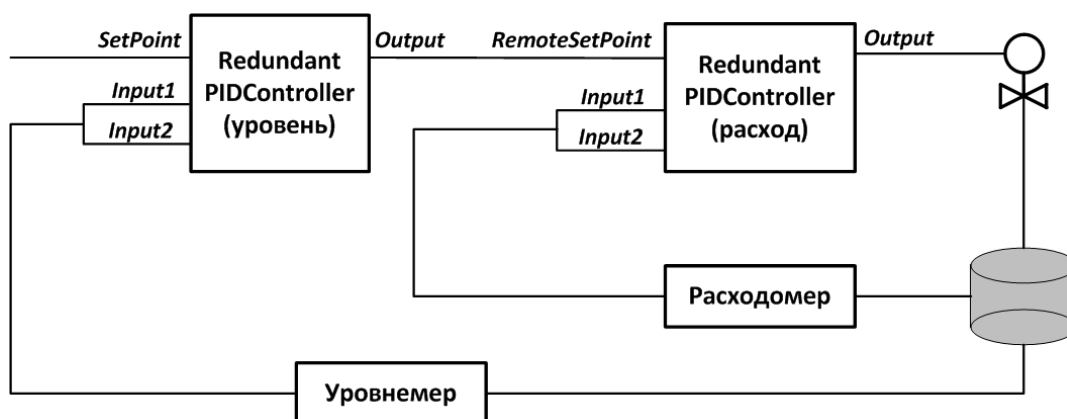
Рисунок 49. RedundantPIDController в режиме AUTO с использованием схемы резервирования



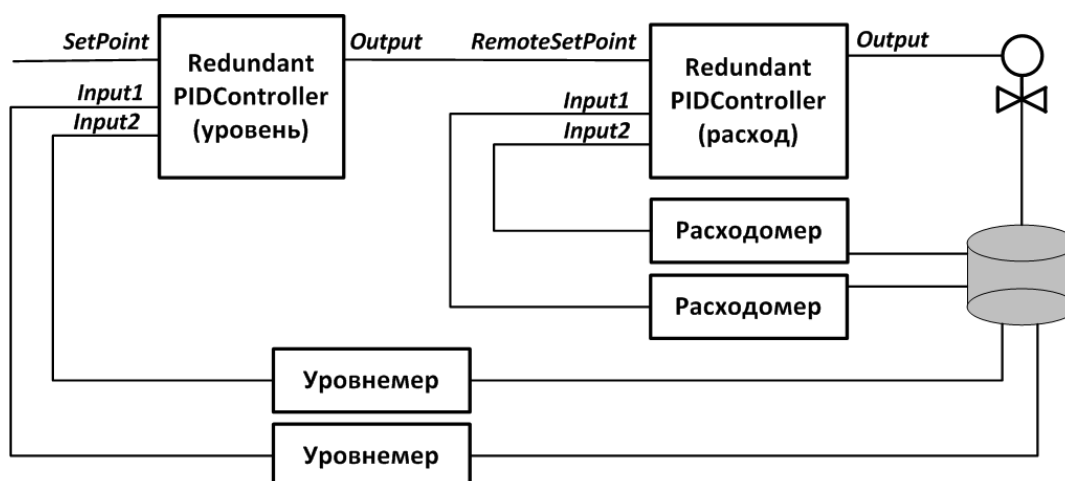
**Рисунок 50. RedundantPIDController в режиме AUTO с использованием входа стробирования**

**Прим.:** Здесь и далее на рисунках приведены схемы с использованием/без использования схемы резервирования.

3. MASKING ("Маскирование") - алгоритм выполняется за исключением этапа формирования сигнализаций.
4. MANUAL ("Ручной"). В данном режиме производится ручное управление RedundantPIDController через запись ручного значения *Manual Value* в атрибут *Output*, при этом входы *Input1* и *Input2* отключаются.
5. CASCADE ("Каскадный"). В данном режиме обеспечивается возможность работы RedundantPIDController в качестве ведомого по отношению к другому RedundantPIDController. На рисунках приведены схемы, в которых RedundantPIDController для расхода является ведомым по отношению к RedundantPIDController для уровня.



**Рисунок 51. RedundantPIDController в режиме CASCADE с блокировкой схемы резервирования**



**Рисунок 52. RedundantPIDController в режиме CASCADE с использованием схемы резервирования**

Дополнительные управляющие функции:

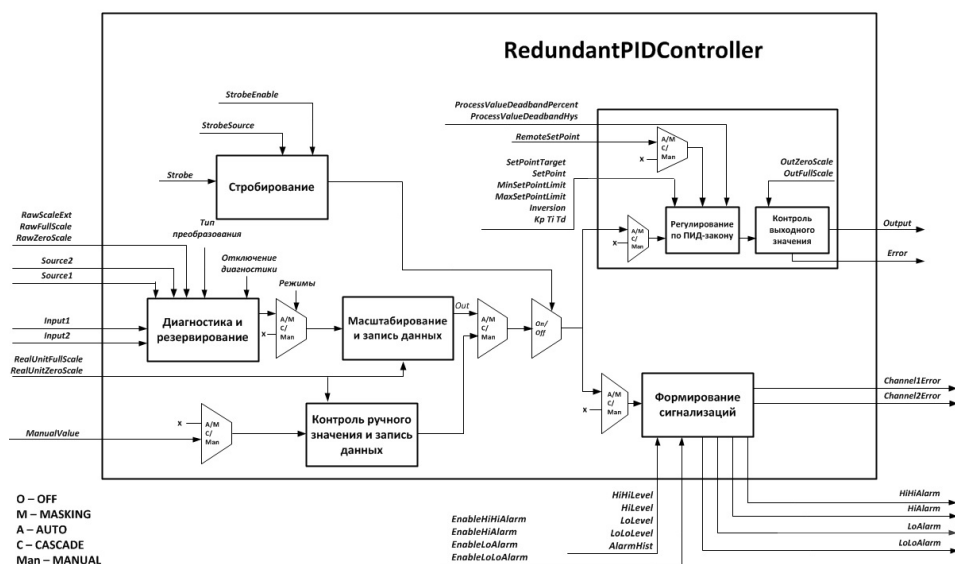
- Стробирование - блок ПИД регулятора со стробированием выполняет обработку входа, управляющих расчётов, выхода и аварийной сигнализации. ПИД регулятор со стробированием останавливается после выполнения каждого действия управления и ждёт выхода результата. Этот блок может использоваться для управления процессами с длительным периодом нечувствительности или для управления на основе данных от анализаторов.
- Безударный пуск - переключение режима функционального блока или меняет значение управляющего выхода вторичного блока в каскадном соединении без резкого изменения значения.

### 7.14.3. Алгоритм RedundantPIDController

Алгоритм RedundantPIDController зависит от установленного режима.

Алгоритм включает в себя следующие элементы:

- диагностика и резервирование;
- масштабирование и запись данных;
- контроль ручного значения и запись данных;
- формирование сигнализаций;
- регулирование по ПИД-закону;
- контроль выходного значения.



**Рисунок 53. Процесс обработки данных RedundantPIDController**

### 7.14.3.1. Диагностика и резервирование

На данном этапе производится диагностика и резервирование каналов приема данных.

При этом учитываются следующие правила:

- Для приема данных используется канал 1;
- Если канал 1 не доступен, используется канал 2;
- Если оба канала не доступны, формируется статус **BadInput** и сигнализационный флаг об ошибке **Bad**;
- Если оба канала доступны, но выдают разные значения, то производится сравнение абсолютной разности значений с допустимым пределом 2 % от разности границ шкалы. Если абсолютная разность значений превышает допустимый предел в течение трех рабочих циклов, атрибут Status принимает значение 0x02030001 (BadInput) и выставляется сигнализационный флаг об ошибке **Bad**. Данное правило может быть отключено установкой флага в атрибуте **DisableInputSignalsComparison**.

Входное значение проверяется на соответствие диапазону шкалы.

Если входное значение выходит за пределы диапазона шкалы, то атрибут **Status** RedundantPIDController принимает значение 0x02030001 (BadInput).

Если неверно задан диапазон шкалы (например, верхний предел меньше нижнего предела), то атрибут **Status** блока RedundantPIDController примет значение 0x02030000 (BadConfiguration).

Расширение диапазона входного значения **RawScaleExt** позволяет диагностировать неисправность датчика при отсутствии обрыва или короткого замыкания.



Различают типы преобразования входных сигналов (*InputConversionType*): Linear - линейное, Sqrt - квадратичное. Линейное преобразование позволяет использовать текущее значение сигнала, квадратичное - квадратный корень, извлеченный из значения сигнала.

#### 7.14.3.2. Масштабирование и запись данных

Данный этап выполняется в режимах AUTO и MASKING.

Входное значение переводится в инженерные единицы, и полученное значение переменной *Out* подается на вход элемента алгоритма "Регулирование по ПИД-закону".

Формула для перевода входного значения в инженерные единицы:

$$Out = \frac{RealUnitFullScale - RealUnitZeroScale}{RawFullScale - RawZeroScale} \cdot (Input - RawZeroScale) + RealUnitZeroScale,$$

где *RawFullScale* и *RawZeroScale* - верхняя и нижняя границы шкалы в процентах, *RealUnitFullScale* и *RealUnitZeroScale* - верхняя и нижняя границы шкалы в инженерных единицах.

При квадратичном преобразовании входного значения (установлен флаг *InputConversionType*) масштабирование производится согласно формуле:

$$PV = \frac{PV_{SH} - PV_{SL}}{IN_{SH} - IN_{SL}} \cdot (\sqrt{IN} - IN_{SL}) + PV_{SL}$$

#### 7.14.3.3. Контроль ручного значения

Данный этап выполняется в режимах MASKING и MANUAL.

Значение атрибута *ManualValue* сравнивается со шкалой датчика в инженерных единицах.

Если значение *ManualValue* выходит за одну из границ шкалы, выходное значение будет приравнено к величине соответствующей границы.

#### 7.14.3.4. Формирование сигнализаций

Этап формирования сигнализаций функционального блока описан в разделе 8.1. Формирование сигнализаций Библиотеки.

### 7.14.3.5. Регулирование по ПИД-закону

На данном этапе алгоритма производится обработка данных для последующей записи в выходные атрибуты.

Значение уставки **SetPoint** должно соответствовать заданному диапазону **MinSetPointLimit** и **MaxSetPointLimit**. В случае выхода из указанного диапазона значение **SetPoint** приравнивается к ближайшему граничному значению.

Обработка данных производится в соответствии с ПИД-законом:

$$Output = \frac{OutFullScale - OutZeroScale}{2} + Inv \cdot (Kp \cdot Error + intsum_n + Td \cdot \Delta Error),$$

где

*Inv* – переменная инверсии; *Inv* = 1, если атрибут *Inversion* принимает значение false ; *Inv* = -1, если значение *Inversion* true.

$$Error = SetPoint - Out$$

$$\Delta Error = Error_n - Error_{n-1};$$

*n* – номер системного цикла;

$$intsum_n = \frac{\Delta t}{Ti} \cdot Error_n + intsum_{n-1} - \text{интегральная сумма};$$

$\Delta t$  – длительность системного цикла,  $\Delta t = 0.1$  с

### 7.14.3.6. Контроль выходного значения

Контроль выходного значения осуществляет проверку обработанного значения на соответствие границам диапазона **OutZeroScale** и **OutFullScale** и запись в атрибут **Output**.

При выходе обработанного значения из указанного диапазона происходит приравнивание этого значения к ближайшей границе диапазона.

## 7.15. Функциональный блок DiscretePacker

Функциональный блок Discrete Packer (далее DiscretePacker) предназначен для упаковки данных из атрибутов входного значения. На входе производится упаковка данных типа BOOL, а на выходе производится запись в тип INT.

### 7.15.1. Атрибуты DiscretePacker

DiscretePacker имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 77. Дополнительные атрибуты DiscretePacker

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные/выходные атрибуты</i>			
Input1... Input16	BOOL	чтение	Входное значение от каналов 1...16 (Рисунок 54. Битовый формат выходного значения)
Output	INT	чтение	Выходное значение
<i>Конфигурационные атрибуты</i>			
Source1...Source16	STRING	чтение/ запись	Имя источника данных 1...16
Manual Value	INT	чтение/ запись	Ручное значение

Флаги **Bad** формируются из статусов источников входного сигнала.

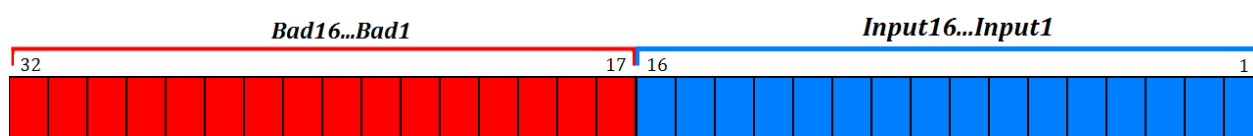


Рисунок 54. Битовый формат выходного значения

### 7.15.2. Режимы DiscretePacker

Режимы DiscretePacker:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.15.3. Алгоритм DiscretePacker

Алгоритм DiscretePacker включает в себя следующие элементы:

- чтение;
- обработка данных.

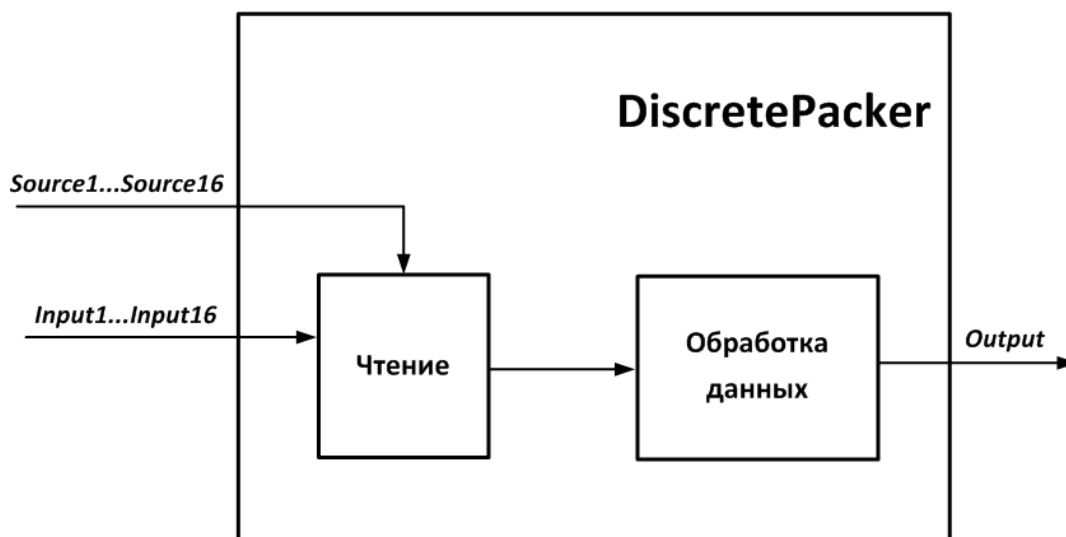


Рисунок 55. Процесс обработки данных DiscretePacker

#### 7.15.3.1. Чтение

При чтении данные с выхода функционального блока-источника атрибуты *Source1...Source16* записываются во входной атрибут *Input*.

#### 7.15.3.2. Обработка данных

Обработка данных заключается в упаковке данных из атрибутов входного значения *Input1...Input16* в тип INT и запись в выходной атрибут *Output*.

## 7.16. Функциональный блок DigitalUnpacker

Функциональный блок Digital Unpacker (далее DigitalUnpacker) выполняет обратную по отношению к DiscretePacker операцию - распаковку данных. На входе производится распаковка данных типа INT, а на выходе производится запись в тип BOOL.

### 7.16.1. Атрибуты DigitalUnpacker

DigitalUnpacker имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 78. Дополнительные атрибуты DigitalUnpacker

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN	DINT	чтение	Входное значение (Рисунок 56. Битовый формат входного значения)

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Выходные атрибуты</b>			
OUT1... OUT16	BOOL	чтение	Выход1... Выход16
BAD1... BAD16	BOOL	чтение	Качество сигнала
<b>Конфигурационные атрибуты</b>			
IN_SRC	STRING	чтение/ запись	Имя источника данных
MV	DINT	чтение/ запись	Входное значение, вводимое в режимах имитация и имитация маскирования
FV	DINT	чтение	Значение поля

Флаг **Bad** формируется из статусов источников сигналов проединенного блока упаковки (DiscretePacker).

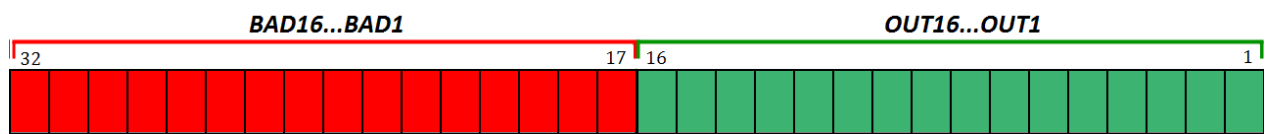


Рисунок 56. Битовый формат входного значения

### 7.16.2. Режимы DigitalUnpacker

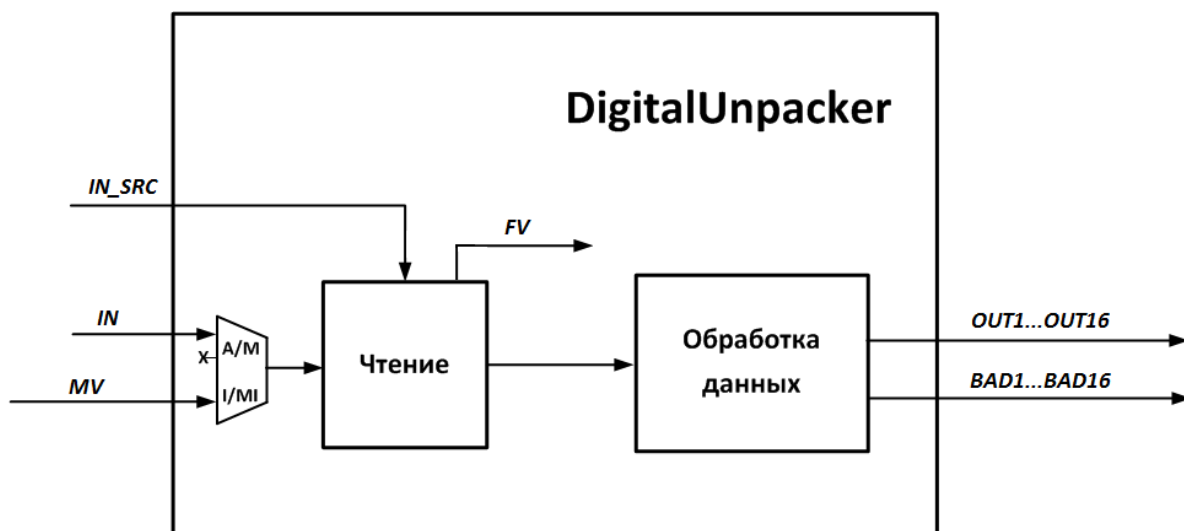
Режимы DigitalUnpacker:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 7.16.3. Алгоритм DigitalUnpacker

Алгоритм DigitalUnpacker включает в себя следующие элементы:

- чтение;
- обработка данных.



**A** – AUTO  
**M** – MASKING  
**I** – IMITATION  
**MI** – MASK\_IMITATION

**Рисунок 57. Процесс обработки данных DigitalUnpacker**

### 7.16.3.1. Чтение

Чтение данных из функционального блока-источника происходит в режимах AUTO и MASKING.

При чтении данные с выхода функционального блока-источника атрибута *IN\_SRC* записываются во входной атрибут *IN*.

В режимах IMITATION и MASK\_IMITATION входные данные вводятся вручную и записываются в атрибут *MV*.

### 7.16.3.2. Обработка данных

Обработка данных заключается в выполнении распаковки данных типа INT из атрибута входного значения *IN* и запись в выходные атрибуты *OUT1...OUT16* данных типа BOOL.

В атрибуты *BAD1...BAD16* записывается качество выходных сигналов *OUT1...OUT16* соответственно.

В режиме MANUAL данные в *OUT1...OUT16* и *BAD1...BAD16* записываются из атрибута *MV*.

## 7.17. Функциональный блок TimeDelay

Функциональный блок Time Delay (далее TimeDelay) обеспечивает исполнение таймера с задержкой включения/ выключения.

### 7.17.1. Атрибуты TimeDelay

TimeDelay имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 79. Дополнительные атрибуты TimeDelay**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i><b>Входные атрибуты</b></i>			
Input	BOOL	чтение	Входное значение
Reset	BOOL	чтение	Сброс
TimeLeft	INT	чтение	Оставшееся время
<i><b>Выходные атрибуты</b></i>			
Output	BOOL	чтение	Выходное значение
Output Inverse	BOOL	чтение	Выходное значение, инвертированное
<i><b>Конфигурационные атрибуты</b></i>			
Source1	STRING	чтение/ запись	Источник данных для Input
Source2	STRING	чтение/ запись	Источник данных для Reset
SignalEdge	INT	чтение/ запись	Тип таймера: Rising - с задержкой включения, Falling - с задержкой выключения
TimeDelay	INT	чтение/ запись	Длительность задержки

### 7.17.2. Режимы TimeDelay

Режимы TimeDelay:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.17.3. Алгоритм TimeDelay

Алгоритм включает в себя следующие элементы:

- чтение;
- обработка данных.

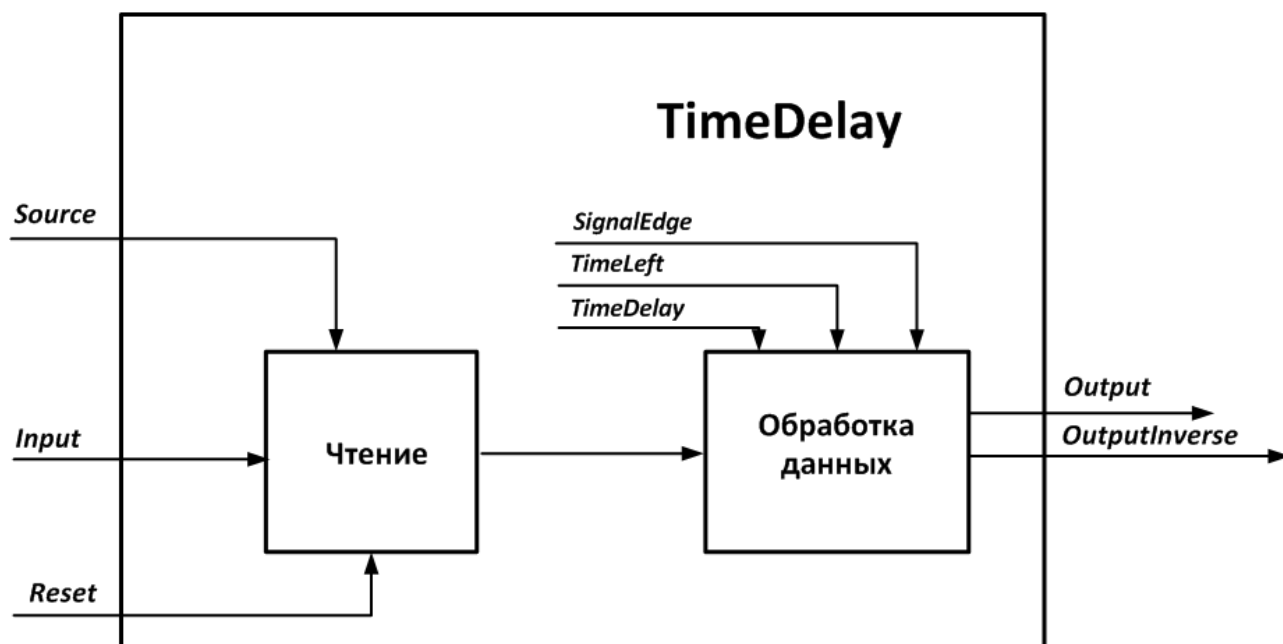


Рисунок 58. Процесс обработки данных TimeDelay

### 7.17.3.1. Чтение

При чтении данные с выходов функциональных блоков-источников записываются во входные атрибуты.

### 7.17.3.2. Обработка данных

Различают типы таймера (атрибут *SignalEdge*):

- Rising - 0, таймер с задержкой включения (TON);
- Falling - 1, таймер с задержкой отключения (TOF).

### Таймер с задержкой включения (TON)

Блок используется для операции задержки передачи сигнала. Работу поясняет приведенная на рисунке диаграмма.

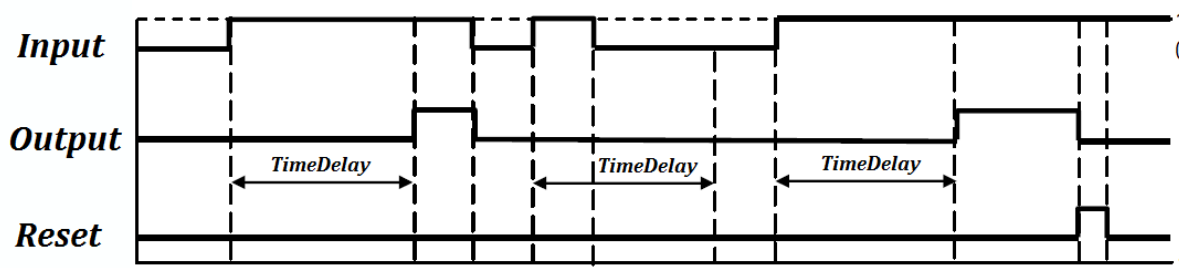


Рисунок 59. Таймер с задержкой включения

На выходе *Output* блока появится логическая «1» с задержкой относительно фронта входного сигнала *Input*. Выход включается логической «1» на входе продолжительностью не менее длительности *TimeDelay*, а выключается по спаду входного сигнала либо когда команда сброса *Reset* изменяется на "1".



### Таймер с задержкой отключения (TOF)

Блок используется для задержки отключения выхода. Работу поясняет приведенная на рисунке диаграмма.

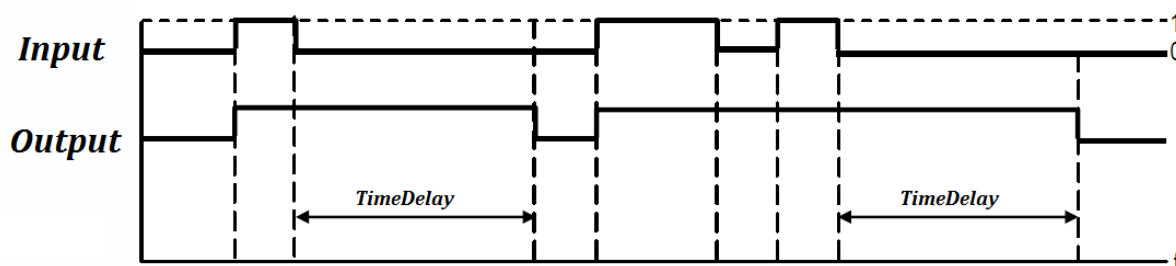


Рисунок 60. Таймер с задержкой отключения

На выходе блока **Output** появится логическая «1» по фронту сигнала на входе **Input**, а начало отсчета времени задержки отключения (*TimeDelay*) происходит по каждому спаду входного сигнала. После отключения входного сигнала на выходе появится логический «0» с задержкой *TimeDelay*. В данном блоке вход **Reset** не используется

## 7.18. Функциональный блок PulseDuration

Функциональный блок Pulse Duration (далее PulseDuration) используется для генерации импульсов заданной/переменной длительности.

### 7.18.1. Атрибуты PulseDuration

PulseDuration имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 80. Дополнительные атрибуты PulseDuration

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
Input	BOOL	чтение	Входное значение
Reset	BOOL	чтение	Сброс
TimeLeft	INT	чтение	Оставшееся время
<i>Выходные атрибуты</i>			
Output	BOOL	чтение	Выходное значение
Output Inverse	BOOL	чтение	Выходное значение, инвертированное
<i>Конфигурационные атрибуты</i>			

Имя атрибута	Тип данных	Доступ	Описание атрибута
Source2	STRING	чтение/ запись	Источник данных для Reset
Source1	STRING	чтение/ запись	Источник данных для Input
PulseType	INT	чтение/ запись	Тип импульса: Fixed - с заданной длительностью, Variable - с переменной длительностью
PulseDuration	INT	чтение/ запись	Длительность импульса

### 7.18.2. Режимы PulseDuration

Режимы PulseDuration:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.18.3. Алгоритм PulseDuration

Алгоритм включает в себя следующие элементы:

- чтение;
- обработка данных.

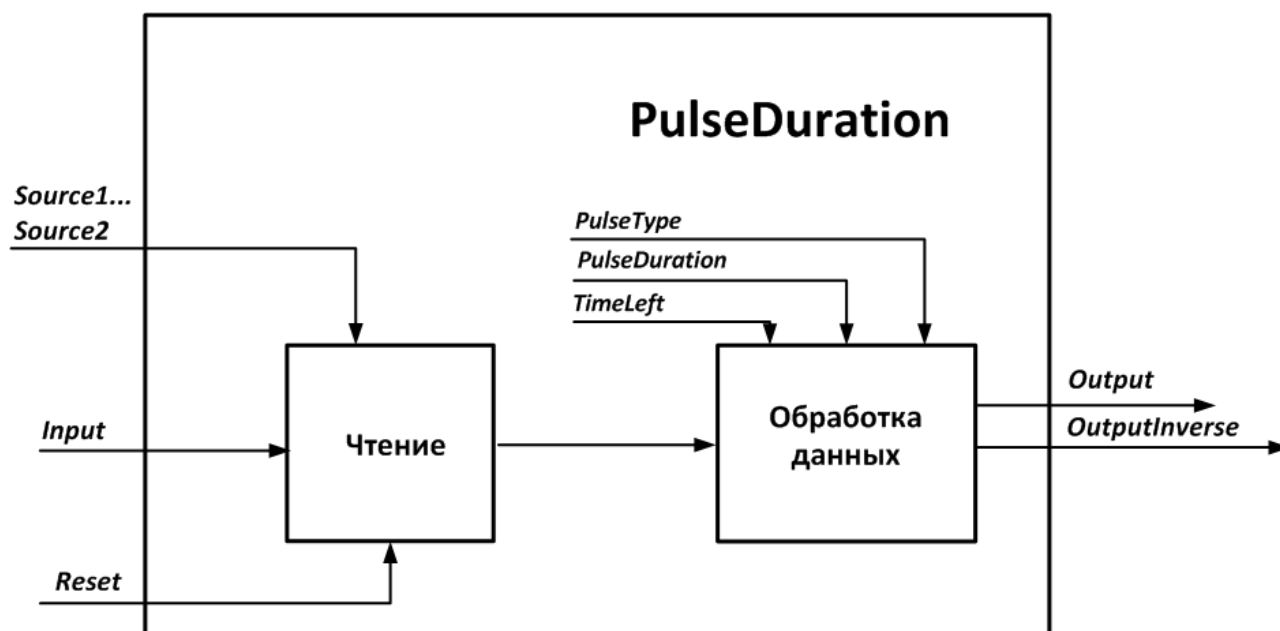


Рисунок 61. Процесс обработки данных PulseDuration

### 7.18.3.1. Чтение

При чтении данные с выходов функциональных блоков-источников записываются во входные атрибуты.

### 7.18.3.2. Обработка данных

Различают типы импульса (атрибут *PulseType*):

- Fixed - 0, импульс с заданной длительностью;
- Variable - 1, импульс с переменной длительностью.

#### Импульс включения заданной длительности

Блок используется для формирования импульса включения выхода на заданный интервал времени. Работу поясняет приведенная на рисунке диаграмма.

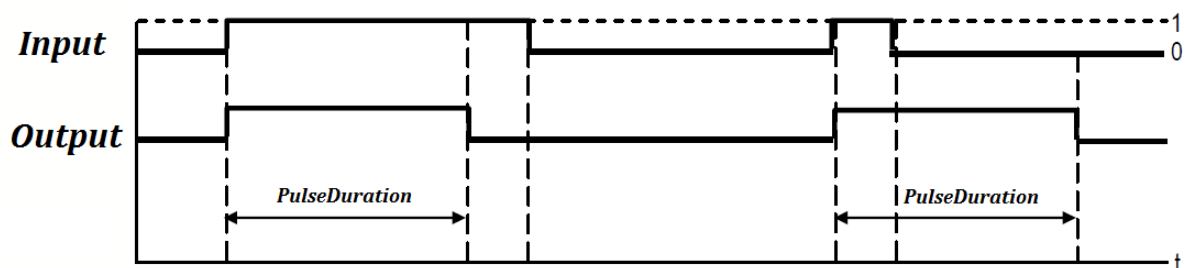


Рисунок 62. Импульс включения заданной длительности

На выходе **Output** блока появится логическая «1» по фронту входного сигнала **Input**.

После запуска выход **Output** не реагирует на изменение значения входного сигнала на интервале *PulseDuration*, а по истечении этого интервала сбрасывается в «0». В данном блоке вход **Reset** не используется.

#### Импульс включения переменной длительности

Блок используется для формирования импульса включения выхода на переменный интервал времени. Работу поясняет приведенная на рисунке диаграмма.

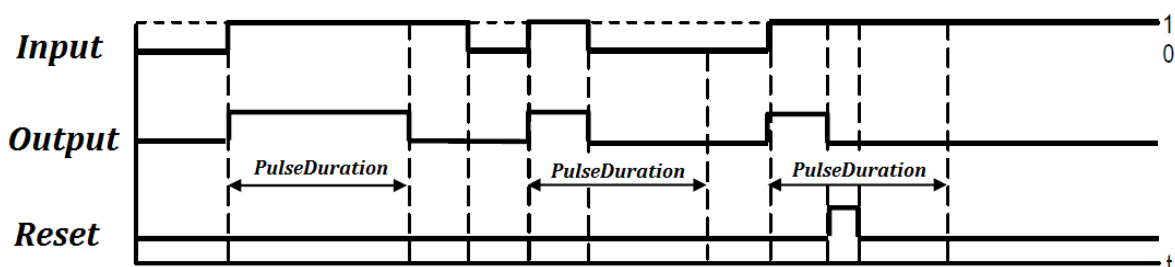


Рисунок 63. Импульс включения переменной длительности

На выходе **Output** блока появится логическая «1» по фронту входного сигнала **Input**.

Выход включается логической «1» на входе продолжительностью не менее длительности **PulseDuration**, а выключается по спаду входного сигнала либо когда команда сброса **Reset** изменяется на "1".

## 7.19. Функциональный блок RSTrigger

Функциональный блок RSTrigger (далее RSTrigger) представляет собой асинхронный триггер с поддержкой тактирования.

Возможность тактирования обеспечивается подключением входа **InputEnable** к источнику сигнала.

### 7.19.1. Атрибуты RSTrigger

RSTrigger имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 81. Дополнительные атрибуты RSTrigger

Имя атрибута	Тип данных	Доступ	Описание атрибута
<b>Входные атрибуты</b>			
InputSet	BOOL	чтение	Входное значение установки
InputReset	BOOL	чтение	Входное значение сброса
InputEnable <sup>19</sup>	BOOL	чтение	Входное значение активации
Reset	BOOL	чтение	Сброс
<b>Выходные атрибуты</b>			
Output	BOOL	чтение	Выходное значение
OutputInverse	BOOL	чтение	Выходное значение, инвертированное
<b>Конфигурационные атрибуты</b>			
Source1	STRING	чтение/ запись	Источник данных для InputSet

<sup>19</sup> Атрибут **InputEnable** предназначен для запуска триггера. При установке значения "1" включается стандартный тактируемый RS-триггер. При установке значения "0" включается состояние хранения, т.е. независимо от выходного значения атрибут **Output** сохраняет предыдущее состояние.

Имя атрибута	Тип данных	Доступ	Описание атрибута
Source2	STRING	чтение/ запись	Источник данных для InputReset
Source3	STRING	чтение/ запись	Источник данных для InputEnable
OutputInitial <sup>20</sup>	BOOL	чтение/ запись	Первоначальное значение выхода в нулевой момент времени
TriggerType	INT	чтение/ запись	Тип триггера: RS - с приоритетом сброса, SR - с приоритетом установки

### 7.19.2. Режимы RSTrigger

Режимы RSTrigger:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.19.3. Алгоритм RSTrigger

Алгоритм включает в себя следующие элементы:

- чтение;
- обработка данных.

<sup>20</sup> Атрибуту *Output* присваивается состояние *OutputInitial*, если в нулевой момент времени управляющими входами был сформирован состояние хранения.

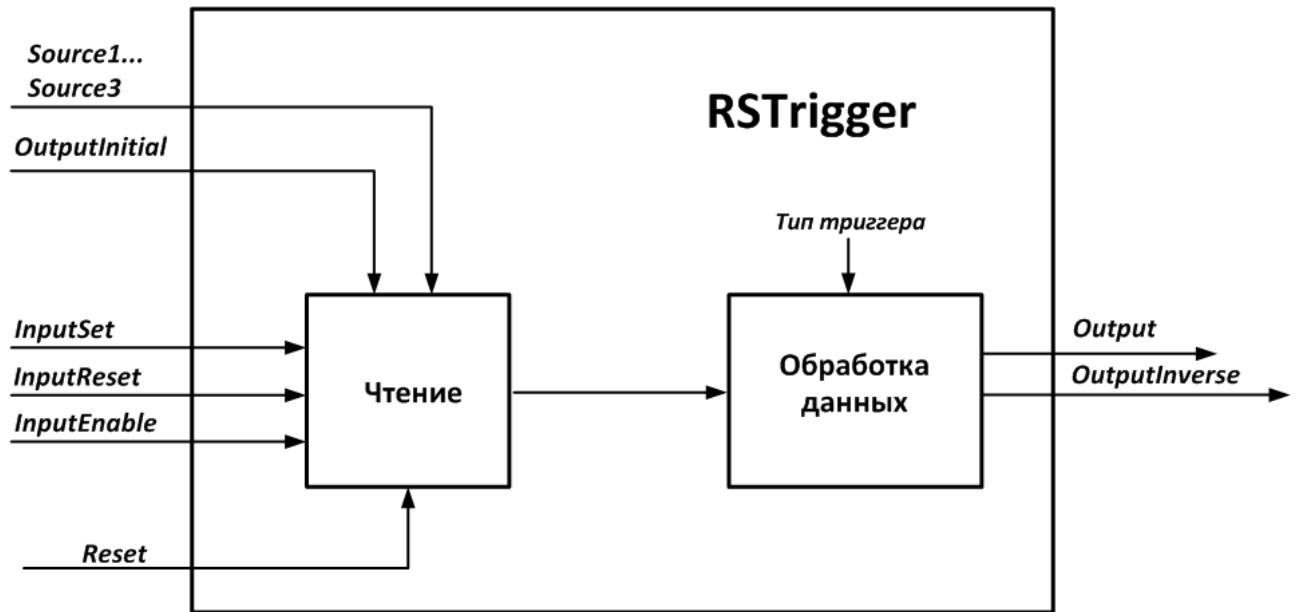


Рисунок 64. Процесс обработки данных RSTrigger

7.19.3.1. Чтение

При чтении данные с выходов функциональных блоков-источников записываются в соответствующие входные атрибуты.

7.19.3.2. Обработка данных

Блок RSTrigger поддерживает следующие типы триггеров (атрибут *TriggerType*):

- RSTrigger с приоритетом сброса - RS;
- RSTrigger с приоритетом установки - SR.

Таблица 82. Таблица перехода состояний блока RSTrigger с приоритетом сброса

R	S	Q(t)	Q(t+1)	Пояснения
0	0	0	0	Состояние хранения информации R=S=0
0	0	1	1	
0	1	0	1	Состояние установки
0	1	1	1	
1	0	0	0	Состояние сброса
1	0	1	0	
1	1	0	0	R=S=1
1	1	1	0	

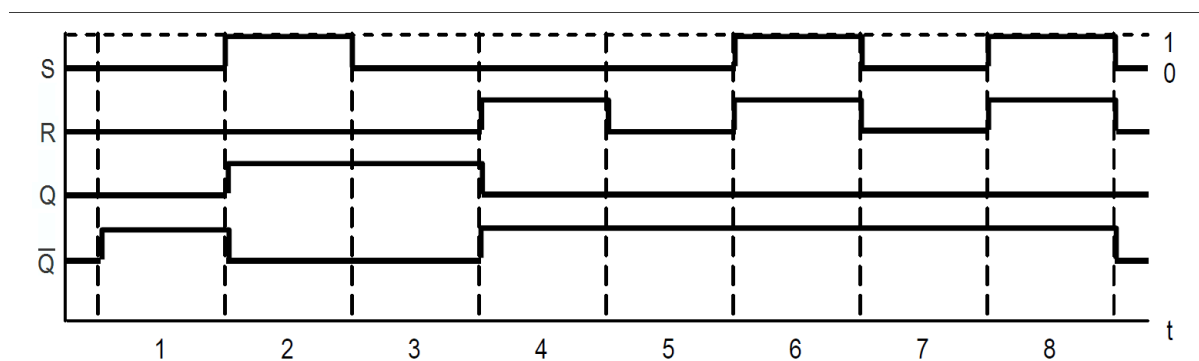
где  $Q(t)$  - текущее состояние,  $Q(t+1)$  - последующее состояние.

**Таблица 83. Таблица перехода состояний блока RSTrigger с приоритетом установки**

R	S	Q(t)	Q(t+1)	Пояснения
0	0	0	0	Состояние хранения информации R=S=0
0	0	1	1	
0	1	0	1	Состояние установки
0	1	1	1	
1	0	0	0	Состояние сброса
1	0	1	0	
1	1	0	1	R=S=1
1	1	1	1	

**Функциональный блок RSTrigger с приоритетом сброса**

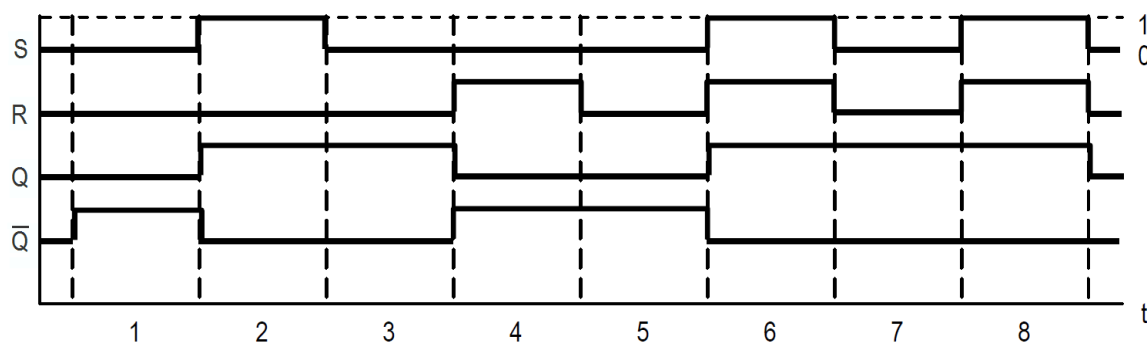
Работу блока RSTrigger с приоритетом сброса (RS) поясняет приведенная на рисунке диаграмма.



**Рисунок 65. Временная диаграмма перехода состояний блока RSTrigger с приоритетом сброса**

**Функциональный блок RSTrigger с приоритетом установки**

Работу блока RSTrigger с приоритетом установки (SR) поясняет приведенная на рисунке диаграмма.



**Рисунок 66. Временная диаграмма перехода состояний блока RSTrigger с приоритетом установки**

## 7.20. Функциональный блок ToWords

Функциональный блок ToWords (далее ToWords) предназначен для разбиения входной величины на две части в соответствии с заданной конфигурацией.

### 7.20.1. Атрибуты ToWords

ToWords имеет базовый набор атрибутов, а также дополнительные атрибуты:

**Таблица 84. Дополнительные атрибуты ToWords**

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
Input 1	REAL	чтение	Вход 1
Input 2	INT	чтение	Вход 2
<i>Выходные атрибуты</i>			
Output 1	INT	чтение	Выход 1
Output 2	INT	чтение	Выход 2
<i>Конфигурационные атрибуты</i>			
Source 1	STRING	чтение/ запись	Имя источника входных данных 1
Source 2	STRING	чтение/ запись	Имя источника входных данных 2
Manual Value 1	INT	чтение/ запись	Ручное значение 1



Имя атрибута	Тип данных	Доступ	Описание атрибута
Manual Value 2	INT	чтение/ запись	Ручное значение 2
Word Endian	INT	чтение/ запись	Порядок слов (WordEndian)
Byte Endian	INT	чтение/ запись	Порядок байтов (ByteEndian)

### 7.20.2. Режимы ToWords

Режимы ToWords:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 7.20.3. Алгоритм ToWords

ToWords разбивает входную величину на две части: старшую и младшую. В качестве входной величины может быть использован тип данных REAL или INT. Порядок следования слов или байтов в выходных величинах задается с помощью конфигурационных атрибутов, согласно *Таблице 79. Дополнительные атрибуты ToWords*.

Правила приоритета входных величин:

1. Если подключен источник *Input 1* или *Input 2*, блок берет входные данные с подключенного входа.
2. Если подключены оба источника, приоритетным является *Input 1*.

В режимах IMITATION и MASK\_IMITATION входы блока отключаются, на выходы *Output 1*, *Output 2* подается значение атрибутов *Manual Value 1*, *Manual Value 2* соответственно.

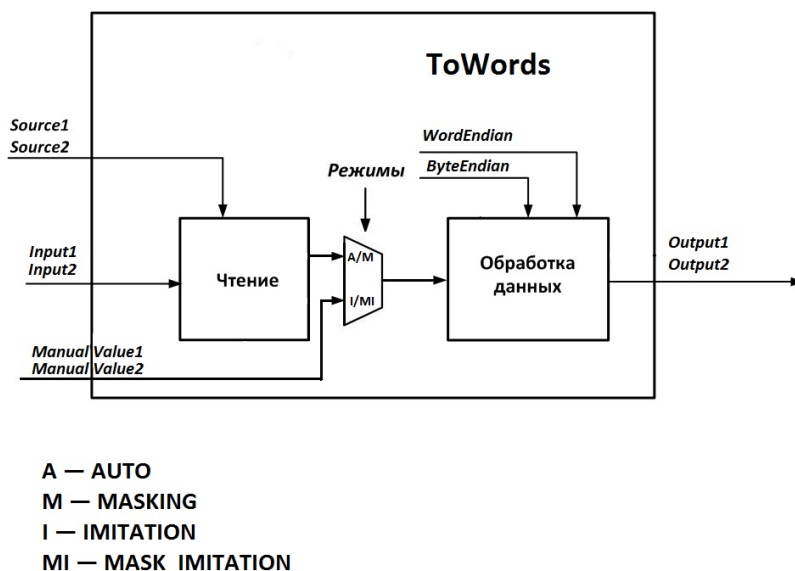


Рисунок 67. Процесс обработки данных ToWords

## 7.21. Функциональный блок WordsTo

Функциональный блок WordsTo (далее WordsTo) предназначен для комбинирования двух входных значений в одно выходное в соответствии с заданной конфигурацией.

### 7.21.1. Атрибуты WordsTo

WordsTo имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 85. Дополнительные атрибуты WordsTo

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
Input 1	INT	чтение	Вход 1
Input 2	INT	чтение	Вход 2
<i>Выходные атрибуты</i>			
Output 1	REAL	чтение	Выход 1
Output 2	INT	чтение	Выход 2
<i>Конфигурационные атрибуты</i>			
Source 1	STRING	чтение/ запись	Имя источника входных данных 1
Source 2	STRING	чтение/ запись	Имя источника входных данных 2

Имя атрибута	Тип данных	Доступ	Описание атрибута
Manual Value 1	REAL	чтение/ запись	Ручное значение 1
Manual Value 2	INT	чтение/ запись	Ручное значение 2
Word Endian	INT	чтение/ запись	Порядок слов ( <b>WordEndian</b> )
Byte Endian	INT	чтение/ запись	Порядок байтов ( <b>ByteEndian</b> )

### 7.21.2. Режимы WordsTo

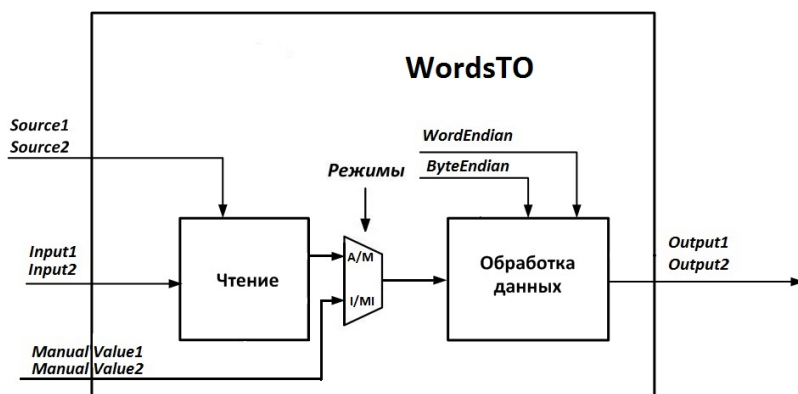
Режимы WordsTo:

- OFF ("Выключен");
- AUTO ("Автоматический");
- MASKING ("Маскирование");
- IMITATION ("Имитация");
- MASK\_IMITATION ("Имитация и маскирование").

### 7.21.3. Алгоритм WordsTo

Входные величины *Input 1*, *Input 2* преобразуются согласно конфигурационным атрибутам *WordEndian*, *ByteEndian* и выдаются в *Output 1* в виде фиксированного типа данных REAL, а в *Output 2* в виде типа данных INT.

В режимах IMITATION и MASK\_IMITATION входы блока отключаются, на выходы *Output 1*, *Output 2* подается значение атрибутов *Manual Value 1*, *Manual Value 2* соответственно.



A — AUTO  
M — MASKING  
I — IMITATION  
MI — MASK\_IMITATION

Рисунок 68. Процесс обработки данных WordsTo

## 7.22. Функциональный блок Not

Функциональный блок Not (далее Not) инвертирует дискретный входной сигнал и генерирует дискретное выходное значение.

### 7.22.1. Атрибуты Not

Not имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 86. Дополнительные атрибуты Not

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN	BOOL	чтение	Вход
<i>Выходные атрибуты</i>			
OUT	BOOL	чтение	Выход
<i>Конфигурационные атрибуты</i>			
IN_SRC	STRING	чтение/ запись	Имя источника входных данных

### 7.22.2. Режимы Not

Режимы Not:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.22.3. Алгоритм Not

Not формирует выходное значение, которое является отрицанием входа: если значение входа - истинно, то выход имеет ложное значение, и наоборот.

## 7.23. Функциональный блок Change Analyzer

Функциональный блок Change Analyzer (далее Change Analyzer) генерирует на выходе дискретный импульс, при позитивном и (или) негативном переходе на дискретном входе с момента последнего выполнения этого блока. Если перехода нет, то дискретный выход имеет состояние "ложно".

### 7.23.1. Атрибуты Change Analyzer

Change Analyzer имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 87. Дополнительные атрибуты Change Analyzer

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN	BOOL	чтение	Вход
<i>Выходные атрибуты</i>			
OUT	BOOL	чтение	Выход
<i>Конфигурационные атрибуты</i>			
IN_SRC	STRING	чтение/ запись	Имя источника входных данных
FR_SEL	USINT	чтение/запись	Передний выбор

### 7.23.2. Режимы Change Analyzer

Режимы Change Analyzer:

- OFF ("Выключен");
- AUTO ("Автоматический").

### 7.23.3. Алгоритм Change Analyzer

1. Если **FR\_SEL** = Both, то ФБ генерирует на выходе дискретный импульс "истинно" на один скан, когда на дискретном входе возникает позитивный переход или негативный переход с момента последнего выполнения этого

блока. Если перехода не было, то дискретный выход блока имеет состояние "ложно".

2. Если  $FR\_SEL = \text{Positive}$ , то, в случае, когда значение входа с момента последнего выполнения блока изменилось из "ложно" в "истинно", то на выходе блока устанавливается состояние "истинно". Если значение входа не менялось, то выход блока имеет состояние "ложно".
3. Если  $FR\_SEL = \text{Negative}$ , то в случае, когда значение входа с момента последнего выполнения блока изменилось из "истинно" в "ложно", то на выходе блока устанавливается состояние "истинно". Если значение входа не менялось, то выход блока имеет состояние "ложно".

## 8. Функциональные блоки противоаварийной защиты

### 8.1. Функциональный блок DiscreteVoter4/16

Функциональные блоки Discrete Voter 4 и Discrete Voter 16 (далее DiscreteVoter4/16) обеспечивают голосование по дискретным входам. DiscreteVoter4/16 может контролировать до 4 или 16 входов соответственно, определяя, достаточно ли голосов для срабатывания защиты. Нормальным значением на выходе является 1, значением срабатывания защиты - 0.

#### 8.1.1. Атрибуты DiscreteVoter4/16

DiscreteVoter4/16 имеет базовый набор атрибутов, а также дополнительные атрибуты:

Таблица 88. Дополнительные атрибуты DiscreteVoter4/16

Имя атрибута	Тип данных	Доступ	Описание атрибута
<i>Входные атрибуты</i>			
IN1...IN16	BOOL	чтение	Входное значение от каналов 1...16
BPS_IN	BOOL	чтение	Байпас всех входов
BPS1_IN...BPS16_IN	BOOL	чтение	Байпас входа 1...16
RST_IN	BOOL	чтение	Сброс входного значения
<i>Выходные атрибуты</i>			
OUT	BOOL	чтение	Выходное значение
OUT_NO_BPS	BOOL	чтение	Выходное значение без байпасов
AOS_LAST_ALIVE	DINT	чтение	Последнее рабочее входное значение перед AOS <sup>21</sup> (первые 4 бит)

<sup>21</sup> AOS — 32-битное число.

Имя атрибута	Тип данных	Доступ	Описание атрибута
NFOUT1...NFOUT16	BOOL	чтение	Отражение состояния входов на момент срабатывания воутера 1...16
RST_RDY	BOOL	чтение	Готовность к сбросу
<b>Конфигурационные атрибуты</b>			
IN1_SRC...IN16_SRC	STRING	чтение/ запись	Имя источника данных 1...16
BPS1...BPS16	BOOL	чтение/ запись	Включение байпаса 1...16
BPS	BOOL	чтение/ запись	Включение байпаса всех входов
BPS_OPTS	USINT	чтение/ запись	Опции байпасов
STS_OPTS	DINT	чтение/ запись	Опции обработки статуса входа
NUM_TO_TRP	DINT	чтение/ запись	Количество голосов, необходимое для перевода блока в защиту
NUM_TOT	DINT	чтение/ запись	Общее количество голосов
TRP_DLY	DINT	чтение/ запись	Задержка переключения в защиту, мс
NR_DLY	DINT	чтение/ запись	Задержка переключения в норму, мс
RQ_RST	BOOL	чтение/ запись	Выход требует сброса после защиты
RST	BOOL	чтение/ запись	Сброс выходного значения
BPS_PERM	BOOL	чтение/ запись	Опция, позволяющая установить Вурасс, если включена
AOS_TRP_DLY	DINT	чтение/ запись	Задержка переключения в защиту при неисправности всех сигналов



Имя атрибута	Тип данных	Доступ	Описание атрибута
AOS_NR_DLY	DINT	чтение/ запись	Задержка переключения в норму при неисправности всех сигналов
BPS_IN_SRC	STRING	чтение/ запись	Байпас всех источников данных
BPS1_IN_SRC... BPS16_IN_SRC	STRING	чтение/ запись	Байпас источника входа 1...16
RST_IN_SRC	STRING	чтение/ запись	Сброс источника данных
<i>Диагностические атрибуты</i>			
TRP_STS	DINT	чтение	Статус воутера
AOS_TRP_ET	DINT	чтение	Таймер переключения в защиту при неисправности всех сигналов
AOS_NR_ET	DINT	чтение	Таймер переключения в норму при неисправности всех сигналов
AOS	BOOL	чтение	Неисправность всех сигналов
ET1...ET16	DINT	чтение	Таймер 1...16
A_NUM_TO_TRP	USINT	чтение	Общее количество голосов для перевода блока в защиту
VOTE1...VOTE16	BOOL	чтение	Голосование 1...16
IN1_STS...IN16_STS	DINT	чтение	Статус входного значения 1...16
VOTE1_NO_BPS... VOTE16_NO_BPS	BOOL	чтение	Голосование без байпасов 1...16

### 8.1.2. Режимы DiscreteVoter4/16

Режимы DiscreteVoter4/16:

- AlwaysUseValue 2003
- WillNotVoteIfBad 2003
- VoteToTripIfBad 2003
- DegradationBySignalMalfunction 2003
- AutomaticOverrideSwitch(AOS) 2003
- DegradationWithAOS 2003

### 8.1.3. Алгоритм DiscreteVoter4/16

**BypassOpts** - опции байпасов:

- BypassReduceNumVoteToTrip - байпас уменьшает количество голосов, необходимых для перехода в защиту.

**StatusOpts** - опции обработки статуса входа:

- AlwaysUseValue – всегда использовать значение входа;
- WillNotVoteIfBad – не будет голосовать, если статус входа плохой (bad);
- VoteToTripIfBad – голосовать за защиту, если статус входа плохой (bad);
- DegradationBySignalMalfunction - деградация по неисправности сигнала (все неисправные каналы не приводят к срабатыванию);
- AutomaticOverrideSwitch - AOS - все неисправные каналы приводят к срабатыванию с задержкой;
- DegradationWithAOS – деградация по неисправности сигнала + AOS.
- AlwaysUseValue 2003

Значение 1	Значение 2	Значение 3	Выход	<u>TripStatus</u>
1	1	1	Норма	<u>Normal</u>
1	1	0	Норма	<u>Normal</u>
1	0	0	Защита	<u>Tripped</u>
1	1	1	Норма	<u>Normal</u>
1	1	0	Норма	<u>Normal</u>
1	0	0	Защита	<u>Tripped</u>





— статус канала ОК

— статус канала BAD


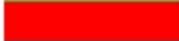
• WillNotVoteIfBad 2003

Значение 1	Значение 2	Значение 3	Выход	<u>TripStatus</u>
1	1	1	Норма	<u>Normal</u>
1	1	0	Защита	<u>TripInhibited</u>
1	0	0	Защита	<u>TripInhibited</u>

 — статус канала ОК  
 — статус канала BAD



• VoteToTripIfBad 2003

Значение 1	Значение 2	Значение 3	Выход	<u>TripStatus</u>
1	1	1	Норма	<u>Normal</u>
1	1	1	Защита	<u>Tripped</u>
1	0	0	Защита	<u>Tripped</u>

 — статус канала ОК  
 — статус канала BAD



• DegradationBySignalMalfunction 2003

Значение 1	Значение 2	Значение 3	Архитектура	Выход	<u>TripStatus</u>
1	1	1	2003	Норма	<u>Normal</u>
1	1	1	2002	Норма	<u>Normal</u>
1	1	1	1001	Норма	<u>Normal</u>
1	1	1		Защита	<u>TripInhibited</u>

 — статус канала ОК  
 — статус канала BAD

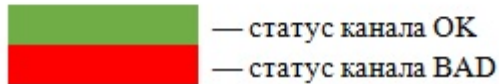
• AutomaticOverrideSwitch(AOS) 2003

Значение 1	Значение 2	Значение 3	Выход	<u>TripStatus</u>
1	1	1	Норма	<u>Normal</u>
1	1	0	Защита	<u>TripInhibited</u>
1	0	0	Защита	<u>TrippedByAOS</u>

 — статус канала ОК  
 — статус канала BAD

- DegrodationWithAOS 2003

Значение 1	Значение 2	Значение 3	Архитектура	Выход	TripStatus
1	1	1	2003	Норма	<u>Normal</u>
1	1	1	2002	Норма	<u>Normal</u>
1	1	1	1001	Норма	<u>Normal</u>
1	1	1		Защита	<u>TrippedByAOS</u>



*TripStatus* - статусы воутера:

- Normal – нормальное состояние;
- Tripped – воутер переключен по защите;
- TrippedByAOS - воутер переключен по защите после AOS;
- TripInhibited – защита подавлена;
- VotedToTripDelayed – голосование за защиту, задержка, если атрибут TripDelay > 0;
- VotedNormalDelayed – голосование за норму, задержка, если атрибут NormalDelay > 0;
- VotedToTripDelayedAOS – голосование за защиту после срабатывания AOS, задержка, если атрибут AoSTripDelay > 0;
- ResetReady – Воутер готов к сбросу (если установлен флаг RequireReset).

Поведение блока при разных статусах входов:

При статусах:

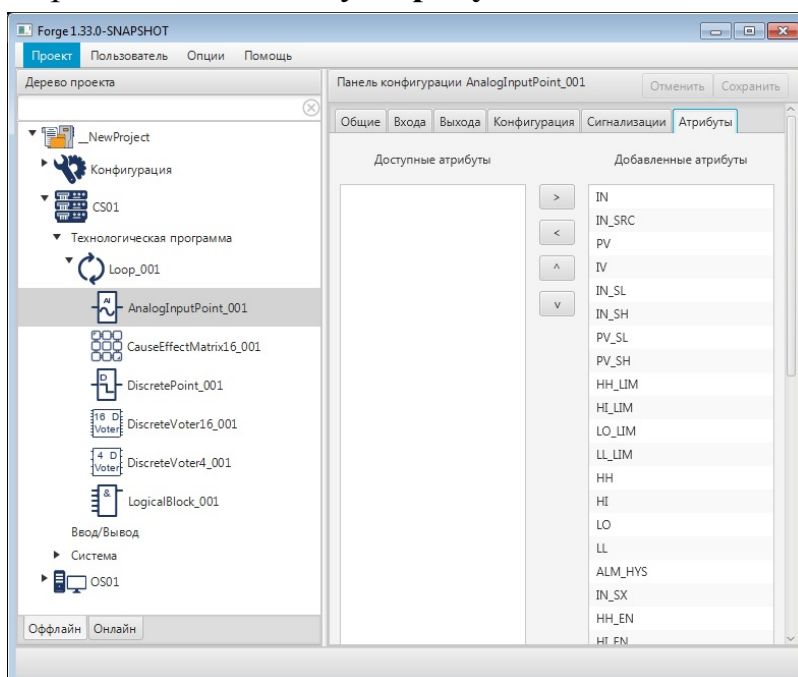
1. **kUninitialised**, **kWaitForService**, **kResolvingReference**, **kUncertainSourceStatus** - статус воутера становится **kWaitForService**;
2. **kOk** - записать во входной атрибут данные из блока источника, продолжать выполнять блок;
3. **kBadInput**, **kHardwareError**, **kCommunicationError**, **kBadSourceStatus** - если канал не в байпасе, то записать ошибку в лог и продолжать выполнять блок с ошибкой на канале;
4. **default** (другие статусы) - переводит воутер в статус входа и выходит из цикла выполнения блока например, **kBadTypeReference/kBadConfiguration/kBadSourceStatus/kBadSourceReference**).

## 9. Кастомные атрибуты для функциональных блоков

Функциональные блоки AnalogInputPoint, Cause Effect Matrix16, Discrete Point, Discrete Voter 4, DiscreteVoter 16, Logical Block имеют кастомные атрибуты - в редакторе диаграмм будут отображаться атрибуты, выбранные Вами.

Настраиваются атрибуты в **Forge > Контроллер > Технологическая программа > Loop > Функциональный блок:**

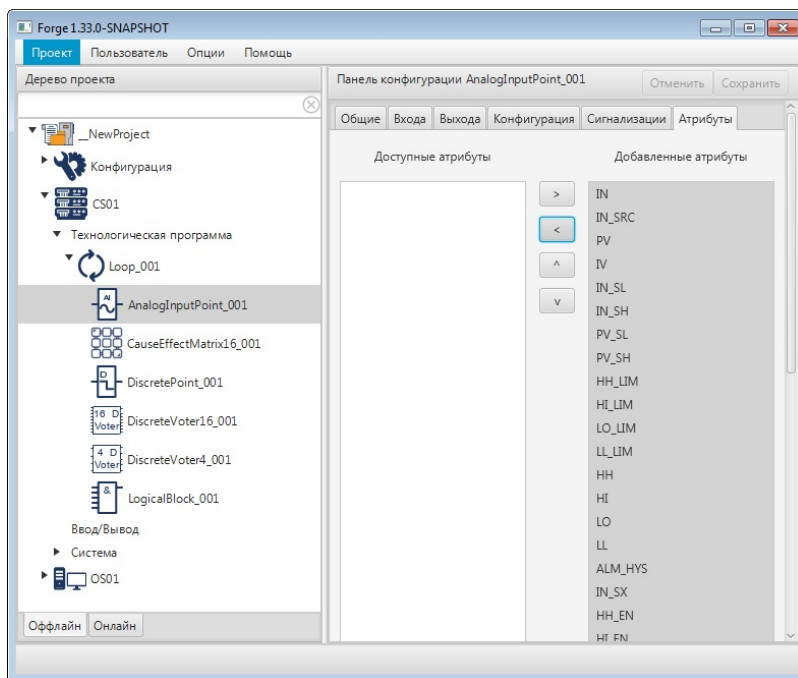
**1. Перейдите во вкладку Атрибуты.**



**Рисунок 69. Вкладка атрибуты**

**2. Выделите атрибуты, которые не нужны для отображения, затем переместите их в окно **Доступные атрибуты** при помощи кнопки **<** в центре панели конфигурации.**

**Прим.:** Для выделения нескольких атрибутов сразу, нажмите на клавиатуре кнопку **Ctrl**, для выделения всех атрибутов нажмите на клавиатуре **Ctrl + A**.



**Рисунок 70. Выделение атрибутов**

3. Переместите атрибуты, которые должны отображаться на контуре в окно **Добавленные атрибуты** аналогичным способом при помощи кнопки **>**.
4. Сохраните изменения.
5. Результат настройки будет отображён в редакторе диаграмм.

**Прим.:** Если редактор диаграмм был запущен ранее, его необходимо перезапустить.

# 10. Приложение. Базовый функционал

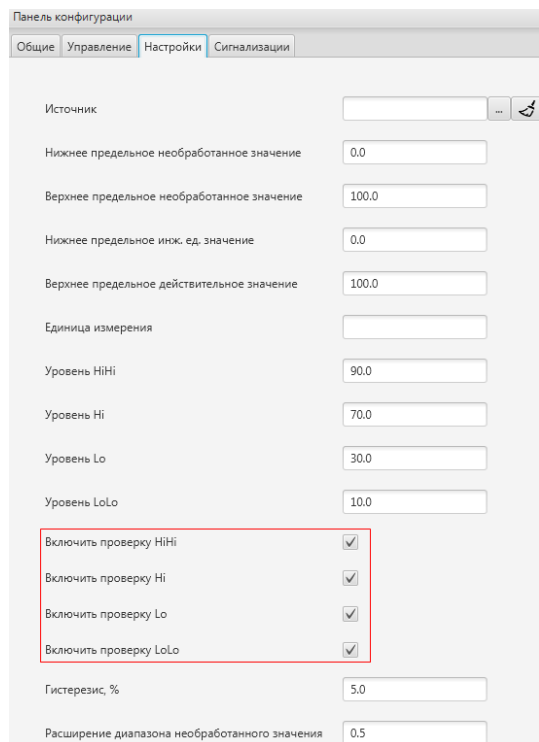
## 10.1. Формирование сигнализаций

В системе реализованы блоки со встроенной сигнализацией выхода уровня сигнала за допустимый диапазон. Данный функционал предоставляют следующие блоки:

- AnalogInputPoint;
- RedundantAnalogInputPoint;
- RedundantPIDController.

Для активации проверки сигнализации на вкладке **Настройки** панели конфигурации блока установите флаги в соответствующих полях (по умолчанию проверка отключена):

- Включить проверку HiHi (аварийного верхнего предела);
- Включить проверку LoLo (аварийного нижнего предела);
- Включить проверку Hi (предупредительного верхнего предела);
- Включить проверку Lo (предупредительного нижнего предела).



Панель конфигурации

Общие | Управление | **Настройки** | Сигнализации

Источник:  ...

Нижнее предельное необработанное значение:

Верхнее предельное необработанное значение:

Нижнее предельное инж. ед. значение:

Верхнее предельное действительное значение:

Единица измерения:

Уровень HiHi:

Уровень Hi:

Уровень Lo:

Уровень LoLo:

Включить проверку HiHi

Включить проверку Hi

Включить проверку Lo

Включить проверку LoLo

Гистерезис, %:

Расширение диапазона необработанного значения:

**Рисунок 71. Настройка проверки сигнализации**

При проверке выполняется сравнение выходного значения **Output** с аварийными и предупредительными уставками.

Сравнение производится с учетом приведенного значения уставки гистерезиса **AlarmHist'** в инженерных единицах.

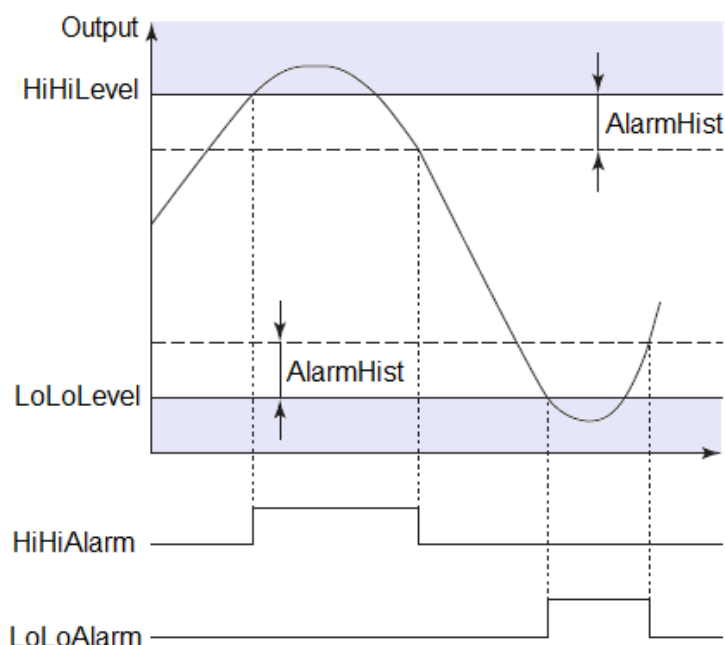
Сброс сформированной сигнализации производится с учетом зоны гистерезиса. Зона гистерезиса задается соответствующей уставкой. В пределах обозначенной зоны сброс сформированной сигнализации блокируется.

В таблице представлены условия формирования сигнализаций для атрибутов:

**Таблица 89. Условия срабатывания и сброса сигнализаций**

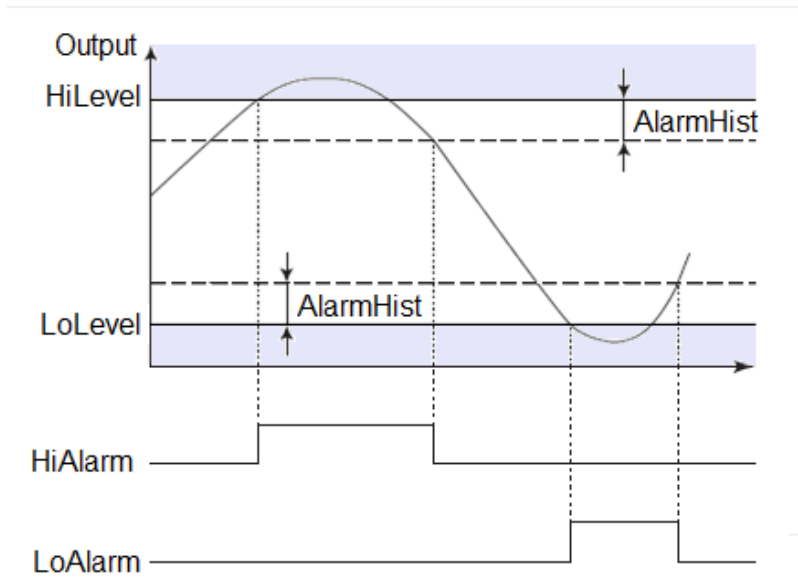
Атрибут	Приор.	Условия срабатывания	Условия сброса
<i>HiHiAlarm</i>	750	$Output > HiHiLevel$	$Output \leq HiHiLevel - AlarmHist'$
<i>HiAlarm</i>	500	$Output > HiLevel$	$Output \leq HiLevel - AlarmHist'$
<i>LoAlarm</i>	500	$Output < LoLevel$	$Output \geq LoLevel + AlarmHist'$
<i>LoLoAlarm</i>	750	$Output < LoLoLevel$	$Output \geq LoLoLevel + AlarmHist'$

Проверка сигнализаций проиллюстрирована на графиках:



**Рисунок 72. Проверка сигнализации аварийного верхнего/аварийного нижнего предела**





**Рисунок 73. Проверка сигнализации предупредительного верхнего/предупредительного нижнего предела**